A dynamical analysis of infinitely wide neural networks

Karl Hajjar

université PARIS-SACLAY

A dynamical analysis of infinitely wide neural networks

Analyse dynamique des réseaux de neurones de largeur infinie

Thèse de doctorat de l'université Paris-Saclay

École doctorale de Mathématiques Hadamard n° 574 (EDMH)
Spécialité de doctorat: Mathématiques appliquées
Graduate School : Mathématiques, Référent : Faculté des sciences d'Orsay

Thèse préparée au **Laboratoire de mathématiques d'Orsay** (Université Paris-Saclay, CNRS), sous la direction de **Christophe GIRAUD**, Professeur, et le co-encadrement de **Lénaïc CHIZAT**, Professeur.

Thèse soutenue à Paris-Saclay, le 12 janvier 2024, par

Karl HAJJAR

Composition du jury

Membres du jury avec voix délibérative

Gilles BLANCHARD Professeur, Laboratoire de Mathématiques d'Orsay, Université Paris-Saclay	Président
Qin LI Professeure Associée (équivalent-HDR), University of Wisconsin–Madison	Rapportrice & Examinatrice
Claire BOYER Maître de Conférences (HDR), Sorbonne Université	Rapportrice & Examinatrice
Aymeric DIEULEVEUT Professeur, CMAP, École Polytechnique	Examinateur

THESE DE DOCTORAT
NNT : 2024UPASM001

université PARIS-SACLAY | ÉCOLE DOCTORALE de mathématiques Hadamard (EDMH)

Titre: Analyse dynamique des réseaux de neurones de largeur infinie
Mots clés: Apprentissage, réseaux de neurones, descente de gradient, largeur infinie

Résumé: Durant la dernière décénnie, les réseaux de neurones ont eu un succès retentissant dans de nombreuses tâches en pratique, cependendant les arguments théoriques derrière ce succès restent insuffisants et une théorie mathématique appropriée pour étudier rigoureusement ces objets fait toujours défaut. Les limites des réseaux de neurones à largeur infinie sont récemment apparues comme une façon d'éclaircir certains aspects du problème. Dans cette thèse, nous étudions la limite des réseaux de neurones de largeur infinie avec une renormalisation particulière souvent dénomée "*champ moyen*" dans la littérature. La difficulté d'analyser les réseaux de neurones d'un point de vue théorique réside en partie dans la nature hautement non-linéaire de ces objets et dans l'énorme quantité de paramètres, ou "*poids*" (pouvant aller jusqu'à la centaine de milliards en pratique) qui interagissent lorsqu'ils sont mis à jour durant la descente de gradient. Nous examinons les trajectoires durant l'optimisation des réseaux de neurones de largeur infinie pendant la phase d'entraînement afin d'exhiber des propriétés de ces modèles dans certains cadres simples tels que les réseaux de neurones entièrement connectés avec une ou plusieurs couches cachées. Cette thèse traite de différents aspects de la dynamique d'optimisation des réseaux de neurones de largeur infinie: des méthodes pour rendre possible l'entraînement de ces modèles aux symétries qui peuvent émerger dans cette limite en passant par de nouveaux algorithmes d'optimisation qui adaptent le nombre de neurones à la volée durant la phase d'entraînement.

Title: A dynamical analysis of infinitely wide neural networks
Keywords: Machine learning, neural networks, gradient descent, infinite-width limit

Abstract: Neural networks have had tremendous success in many practical tasks over the last decade, yet the theoretical reasons behind their performance are poorly understood and we lack a proper mathematical theory to rigorously study the properties of those objects. Infinite-width limits of neural networks have recently emerged as a way to shed light on some of the aspects of the problem. In this thesis, we study the infinite-width limit of networks of different depths under a particular scaling often referred to as the "*mean-field*" scaling in the literature. Part of the reason why neural networks are difficult to analyze from a theoretical standpoint is because they are highly non-linear and involve a huge amount of parameters, or weights, (up to hundreds of billions in practice) which interact as they are updated during gradient descent. We investigate the optimization trajectories of the infinite-width limit of neural networks during training in order to exhibit properties of those models in simple settings such as fully-connected networks with one or more hidden layers. This thesis focuses on different aspects of the optimization dynamics of networks in the infinite-width limit: from methods to enable training those models at arbitrary depths to the symmetry properties that can emerge in that limit as well as novel optimization algorithms which adapt the number of neurons in an on-line fashion during training.

Contents

Remerciements

Il est impossible d'énumérer toutes les personnes à qui je dois d'avoir pu réaliser ce projet de thèse dans de si bonnes conditions durant les trois dernières années, mais je tiens quand même à en remercier les figures les plus importantes.

En premier lieu, je souhaiterais remercier chaleureusement mes deux directeurs de thèse Lénaïc et Christophe. Merci de m'avoir accordé votre confiance pour m'encadrer sur un sujet difficile alors que je ne suivais pas le parcours "classique" du doctorant. Lénaïc, j'étais ton premier thésard et tu as su me guider dans les méandres du monde de la recherche, toujours avec bienveillance et sagacité. Merci aussi pour ta patience, je garderai avec moi par-delà cette thèse ton approche intuitive des problèmes que tu as su me transmettre, ainsi que ta passion communicative pour les sciences. Christophe, merci pour ta bonne humeur que rien ne semble ébranler, ainsi que pour ta curiosité qui t'a poussé à t'aventurer bien au-delà de ton jardin fleuri de stateux pour explorer les contrées encore sombres de l'optimisation des réseaux de neurones.

I would also like to thank Qin Li and Claire Boyer for agreeing to review this PhD thesis and for taking the time to read it, as well as the members of jury for attending my PhD defense.

Ensuite, je voudrais remercier mon amie, mon amour, Louise, sans qui rien de tout ça n'aurait été possible. Toi qui m'as poussé pendant plus d'un an à m'engager dans une thèse malgré mes réserves, toi qui as eu une foi inébranlable en moi, et qui a même partiellement financé cette thèse, je te dois plus que tu ne le sauras jamais. Merci de m'avoir soutenu dans les moments difficiles, et merci de partager ma vie, à tes côtés tout devient plus doux. Cette thèse t'es en partie dédiée.

Je tiens également à remercier mes parents et mes frères, le support sur lequel je peux toujours m'appuyer, qui sont ravis d'accueillir le premier doctorant de la famille bien que cela soit venu interrompre (au grand dam de certains) une assise financière et une carrière solidement établies...

À ceux qui m'ont encouragé à faire de la recherche et à quitter mon confort financier, cette thèse vous est aussi dédiée. Jérémy, ton avidité pour les sciences et ta curiosité intellectuelle ont été et resteront une grande source d'inspiration pour moi. Léonard, toi qui as vécu ta thèse en parallèle de la mienne, merci pour toutes nos discussions, merci de m'avoir aiguillé par moments et de m'avoir remis les pieds sur terres quand il le fallait. Martin (même si tu ne t'en souviens sans doute pas), merci de m'avoir incité à démarrer une thèse au détour d'une conversation avant un foot, et merci d'avoir contribué à ma stabilité financière en quelques occasions.

À tous mes amis qui partagent ma passion du ballon rond, ma famille d'élection, Hippolyte, Matthieu[2], Ruben, Thibaut, Joseph, Goga, Omar, Léonard, Martin, Jérémy, merci de m'avoir apporté un équilibre indispensable en me soustrayant

régulièrement à mes activités scientifiques, et merci pour tous les moments que l'on partage, la vie n'aurait pas la même saveur sans vous.

À Steven et Laurent, mes amis de toujours, vous qui suivez mes pérégrinations (intellectuelles et autres) depuis quasiment 20 ans, merci de me rappeler aux plaisirs simples de la vie, cette thèse est un jalon de plus que l'on franchit ensemble, et je souhaite qu'il y en ait encore de nombreux autres !

À Romain, Mehdi, Emile, Hadrien (et Jean), merci d'accepter mon caractère parfois difficile et de faire dégonfler mon ego quand cela est nécessaire. Vous comptez énormément pour moi, et nos interactions, mêmes rares, me sont extrêmement précieuses. Je promets de continuer à aller vous rendre visite dans vos provinces reculées (dans le 15ème ou même plus loin).

Antoine, Patrick et Ferdinand, nul besoin de vous dire ce que vous représentez pour moi, nos années de prépa ont tissé un lien indéfectible que même un océan ne saurait amenuiser. Merci pour votre soutien, pour votre amitié inconditionnelle et merci de toujours tirer le meilleur de moi-même.

À Carole et Pascal, merci de m'avoir montré que l'on peut rester de bonne humeur même quand ça va mal, et merci de vous vous être intéressés à ma recheche quand bien même le sujet vous était grandement étranger.

À Nathalie et Olivier, avec qui Louise et moi avons cohabité pendant ma première année de thèse, merci de nous avoir accueillis les bras ouverts et d'avoir rendu cette première année aussi facile à vivre.

1 - Introduction

"I think, therefore I am", wrote 17^{th} century French philosopher and mathematician Descartes, hinting at the fact that knowledge of one's own consciousness is a key element of intelligent beings. Will machines ever be able to produce a similar reasoning ? The quest to develop machines able to think, reason, compute and solve problems has occupied scientists of different eras, at least dating back to the efforts of Pascal and Leibniz to produce an arithmetic machine capable of doing various algebraic operations.

Since then, technology has evolved to the point that software installed on one's computer can help solve math problems, translate text into different languages or play the game of chess at a super-human level. A surge of interest in the topic of *artificial intelligence* (AI) has occurred after the second world war with the work of Turing on computing machinery and intelligence (Turing, 1950). Numerous endeavours (such as the Logic Theorist, the Dartmouth Research Project, or Cybernetics) started to appear with the goal of developing expert systems capable of mimicking the problem-solving and reasoning skills of humans. At that time, research was mainly theoretical focusing on the ideas on how to build an AI and how to test for intelligence in machines.

Based on the observation that brain activity simply amounts to electrical impulses that might be reproduced in a computer, research around artificial neural networks and mathematical models of neurons rapidly emerged. One such example is Rosenblatt's perceptron (Rosenblatt, 1958), where a set of potentiometers implementing adaptive weights are able to recognize letters provided as input to the system through an array of 400 photocells. However, the interest in such models quickly faded away as a number of reservations began to surface around artificial neural networks. For instance, the perceptron was criticized for its incapacity to correctly classify data which are not linearly separable (even in simple settings such as the XOR problem), thereby implying a need for deeper networks which came with many practical and theoretical difficulties. Furthermore, practical advances were limited by the compute resources and the expensiveness of computers.

Neural networks thus went out of fashion roughly until the end of 1980s, and research in artificial intelligence focused mainly on expert systems doing symbolic reasoning: that is following a set of handcrafted rules to solve a specific task. Nevertheless, some groups still studied artificial neural networks and new empirical as well as theoretical results revived the interest in such models. Rumelhart et al. (1985) derive the rules of *backpropagation* to compute algorithmically the partial derivatives of the cost function *w.r.t.* the weights of a network using the chain rule. LeCun et al. (1989, 1998) show that neural networks can successfully be applied to handwritten digit, zip code and document recognition, and Barron (1993) and Pinkus (1999) show the universal approximation property of neural networks

with two-layers.

Yet, progress is slow because compute resources are still limited, and a lot of practical experience is needed to design neural networks. Theory and practice of machine learning systems of different kinds are developed in the late 1990s and competitions are even organized to designate the best algorithms on tasks such as image recognition or Natural Language Processing (NLP). These systems often focus their efforts on the crucial problem of feature extraction: transformations of the input data are hand-designed by human experts before being fed to a linear layer whose parameters are algorithmically learned. A seminal moment in the history of artificial neural networks is when the neural network AlexNet (Krizhevsky et al., 2012) won the first place in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) contest in 2012 with an error of only 15.3% on the test set. In this system, all the features are learned automatically by the network along with the last layer.

Since then, neural networks have had many successes in practice, learning to play Atari games (Mnih et al., 2013), achieving less than 5% error on the ImageNet dataset (He et al., 2016), understanding and translating text as well as answering questions on a document (Vaswani et al., 2017), playing go, chess and shogi at a super-human level (Silver et al., 2017), and generating text and images without supervision (Goodfellow et al., 2014; Devlin et al., 2018; Rombach et al., 2022).

Fast progress has been enabled by an ever-growing computational power, allowing networks to grow deeper and wider with a huge amount of parameters (up to *hundreds of billions* for systems like ChatGPT), as well as practical recipes to train modern neural networks such as residual connections (He et al., 2016), batch or layer normalization (Ioffe and Szegedy, 2015; Ba et al., 2016), adaptive gradient-based algorithms that use momentum and learning rate schedules, or attention layers in transformer architectures (Vaswani et al., 2017). Despite the numerous achievements of modern neural networks, theoretical advances are lagging behind and our understanding of the reasons behind such prowesses remains limited.

1.1 . General background

The goal of this thesis is to further our theoretical understanding of the dynamics of the training algorithm of neural networks in the limit where the number of neurons in a layer grows unbounded. Infinite-width asymptotics have recently emerged as a way of providing insights into the training dynamics of neural networks from a mathematical standpoint (see Section 1.2), and the research we present here is part of this line of work.

Many variants of neural network architectures (fully-connected, convolutional, recurrent, transformers, etc) and training algorithms (momentum, learning rate scheduling, batch or full gradient) exist. In this thesis, since we seek to study these objects rigorously, we focus on arguably the simplest form to reduce the complex-

ity: fully-connected neural networks (sometimes with only two layers, sometimes more) trained with plain (stochastic) gradient descent ((S)GD). While our work is theoretical in nature, this thesis does not develop a new mathematical theory of artificial neural networks, rather it uses the available mathematical tools in order to shed light on the behaviour of neural networks used in practice by studying idealized versions of the training dynamics of real-world finite-width networks. We thus make a number of simplifying assumptions (on top of studying the limit where the network becomes infinitely wide) whose nature depends on the setting we consider, varying from the knowledge of the full data distribution (population risk objective), using infinitesimally small step-size (gradient flow dynamics), to studying smooth or positively homogeneous activation functions.

1.1.1 . The risk minimization problem

In machine learning, different types of paradigms exist such as unsupervised learning, reinforcement learning, probabilistic graphical models, and perhaps most ubiquitous the problem of supervised learning. In supervised learning, one is given a data distribution $\mathcal{D}$ of pairs $(x, y) \in \mathbb{R}^d \times \mathbb{R}^K$, and a loss function $\ell : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$, and the goal is to minimize

$$\mathbb{E}_{x,y\sim\mathcal{D}} \left[\ell(y, f(x))\right]$$

over some class $f \in \mathcal{F}$ of functions, called *predictors*, from $\mathbb{R}^d$ to $\mathbb{R}^K$. Usually, y, called the *target*, is modeled by $y = f^*(x)$ or $y = f^*(x)+\epsilon$ where $f^* : \mathbb{R}^d \to \mathbb{R}^K$ is the *target function* and ϵ is some random variable representing potential noise in the data. K is the number of categories or classes, y can either be a continuous variable as in the *regression problem* or an integer in $\{1, \dots, K\}$ as in the *classification problem*.

In this thesis, to simplify, we focus on the regression problem with a single real-valued target y (*i.e.*, $K = 1$), although extensions to multi-dimensional regression should be straightforward. We also consider targets $y = f^*(x)$, $f^* : \mathbb{R}^d \to \mathbb{R}$ without noise as we are mostly concerned with the **optimization trajectory** rather than the statistical properties of the models we consider. In this setting, we thus consider a distribution ρ on the input data and try to solve:

$$\min_{\theta\in\Theta} \Big\{ F(\theta) := \mathbb{E}_{x\sim\rho} \left[\ell(f^*(x), f(\theta; x))\right] \Big\}$$

where F is called the *risk* and the minimization is over a class of *parametric functions* $f(\theta; \cdot)$ with a parameter domain given by some set Θ. The distribution ρ can either be the *empirical distribution* $\hat{\rho}_n := \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, leading to the *Empirical Risk Minimization* (ERM) problem

$$\min_{\theta\in\Theta} \Big\{ F(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(f^*(x_i), f(\theta; x_i)) \Big\},$$

or ρ can be the full theoretical data distribution (*e.g.*, Gaussian, or uniform on some manifold), leading to the *population risk minimization* problem. The task of finding a good value θ^* for the parameter (one that minimizes the objective F, or is close to minimizing it) is called *learning*.

When doing ERM, the end goal is not to be able to learn a good value θ^* for the empirical risk, but for the population risk, often evaluated through a dataset, called the test set (since the true data distribution is unknown), different from the one used to learn θ^*, which is called the training set. A measure of the soundness of the learned value θ^* is the *generalization error*, that is the difference between the empirical risk and the population risk of the optimal parameter θ^* learned on the empirical risk. Sometimes, a *penalization term* is added to the data-fitting term in order to induce good generalization properties and the objective F becomes $F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f^*(x_i), f(\theta; x_i)) + \lambda H(\theta)$ for some penalty H which is often convex (*e.g.*, L^1 or L^2 penalty). When trying to learn θ^*, one turns to the **optimization landscape** of F, while the value of the generalization error is more related the statistical properties of θ^*. This thesis focuses on the first aspect of the problem related to optimization and we mainly consider the risk without penalty.

Linear models

The most studied example of parametric functions is probably that of *linear models* $f(\theta; x) = \theta^\top \Phi(x)$ where $\theta \in \mathbb{R}^p$ is the parameter and $\Phi : \mathbb{R}^d \to \mathbb{R}^p$ is a *feature extractor*. Here, the objective F of the risk minimization problem is convex as soon as the loss ℓ is convex in its second argument, and thus global minimizers exist is most cases, or at least (S)GD is guaranteed to approach a global minimum under mild assumptions. For a long time, this has been the standard way of proceeding: the feature extractor Φ is manually designed depending on the task at hand and the parameter θ of the linear model is learned using (S)GD.

In this setting, the value of the optimal θ^* can often be computed explicitly, or at least with theoretical guarantees leading to a precise analysis of the generalization error and enabling to quantify rigorously the statistical soundness of the learning method.

1.1.2 . Neural networks

Neural networks are parametric functions defined by a succession of linear (or affine) operations followed by a non-linearity, defined through the following expression: $f(\theta; x) = W^L \sigma(W^{L-1}\sigma(\dots\sigma(W^1 x + b^1)) + b^{L-1}) + b^L$, which can also be written recursively as

$$\begin{aligned} f(\theta; x) &= W^L x^{L-1} + b^L, \\ x^l &= \sigma(h^l), \quad h^l = W^l x^{l-1} + b^l, \quad l \in [1, L-1] \\ x^0 &= x, \end{aligned}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is a real-valued function called the *activation function*, and is applied *element-wise* to vectors, and the integer L is called the depth of the network (that is, the total number of layers). For $l \in [1, L]$, the matrices $W^l \in \mathbb{R}^{m_l \times m_{l-1}}$ are called the *weights* and their rows are referred to as *neurons*, and the vectors $b^l \in \mathbb{R}^{m_l}$ are called the *intercepts* or bias terms. The integer m_l is called the width of the l-th layer (that is, the number of neurons in layer l), with $m_L = 1$ in our setting, and $m_0 = d$ the dimension of the inputs x. The vectors $h^l \in \mathbb{R}^{m_l}$ are referred to as the *pre-activations* of layer l, and x^l the *activations*, or intermediate features, while the penultimate layer activations x^{L-1} are generically referred to as the "features". Here the parameter θ is the concatenation of all the weight matrices and intercepts of all layers.

For simplicity of presentation, in this thesis we often omit the intercepts b^l, but this should not have a huge impact on the generality of our work (the extension to networks with intercepts is rather straightforward, and the transformation : $x \mapsto w^\top x + b$ can always be re-written : $x \mapsto \tilde{w}^\top \tilde{x}$ where $\tilde{w} := (w, b)$ and $\tilde{x} := (x, 1)$). Figure 1.1 depicts a typical fully-connected neural network architecture with 6 layers in total. In a neural network, the first layer, also called the *input layer*

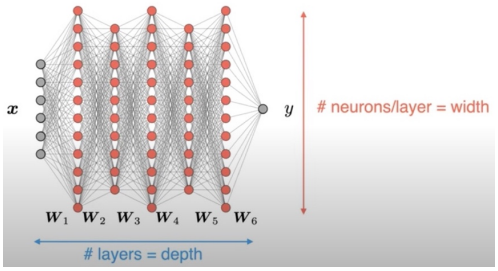


Figure 1.1

($l = 1$) and the last layer, also called the *output or prediction layer*, often behave somewhat differently from the intermediate layers, and for practical reasons, we always consider networks with $L + 1$ layers, with $l = 1$ corresponding to the input layer, $l = L + 1$ to the output layer and $l \in [2, L]$ to the intermediate layers. We also consider networks where all layers except the last one (which is of width 1 in our setting) to have a common width $m \in \mathbb{N}$ for simplicity, and we are interested in describing the limit $m \to \infty$. In this thesis, we always consider $L \geq 1$, so that there are at least 2 layers.

1.1.3 . Learning with neural networks

One of the main modeling differences between neural networks and more traditional machine learning models (such as linear models or kernel methods) is that the features are not manually designed but actually learned automatically by the network itself. For instance one can still write the prediction function, also called the network function, $f(\theta; x) = (\theta^{L+1})^\top \Phi(\theta^L; x)$ but now the feature extractor Φ

has parameters of its own (θ^L is the concatenation first L layer weights) which can be learned simultaneously with the prediction layer's parameters $\theta^{L+1} = W^{L+1}$.

While this sounds appealing, it brings on many complications. First and foremost, even when the loss ℓ is convex, the objective function F corresponding to the risk minimization problem is **not convex** as soon as $L \geq 1$, *i.e.*, when there are two layers or more. Therefore, no guarantees in terms of optimization can be expected a priori, and worse than that, as the number of layers and parameters grow larger, the problem becomes highly non-convex and one can expect many plateaux and/or saddle points where gradient descent can get stuck. There is no knowledge of the value θ^* the learning algorithm will converge to (if it does), which makes studying its statistical properties difficult, and with the sheer number of parameters involved being much larger than the number of training data points, deep networks can be expected to overfit and perform poorly on the test data (generalization error). Hence the need to study the whole training trajectory along the optimization path, but the non-linear nature of the dynamics and the large number of parameters to track make it a difficult task.

Gradient descent for neural networks

While neural networks are complex parametric functions, the algorithm to train them is surprisingly simple. One initializes the parameters at random: weights of different layer are initialized *independently*, and in a given layer, all the entries W^l_{ij} of W^l are initialized i.i.d. following a given law, *e.g.*, Gaussian or uniform. In this thesis, we typically consider the case of a Gaussian initialization $W^l_{ij}(0) \sim \mathcal{N}(0, \sigma_l^2)$ i.i.d. over i, j, with a variance σ_l^2 which depends on the width m. From this initialization, one follows the negative gradient of the objective function: for any $t \geq 0$

$$W^l(t+1) = W^l(t) - \eta_l \nabla_{W^l} F(\theta),$$

where $\eta_l > 0$ is the *learning rate* associated with layer l, which might vary from layer to layer and often depends on m as well. Many different variants of plain (S)GD exist where the learning rate can also depend on the time step t, different coordinates of the gradients are allowed to scaled differently depending on the directions of faster growth. While no guarantee exists a priori for gradient-based algorithms in a non-convex setting, this relatively simply recipe has had huge success in practice, but it is difficult to analyze mathematically due to the highly non-linear and compositional structure of neural networks. It is common in theoretical studies to find the version of gradient descent where $\eta_l \to 0^+$, which is known as *gradient flow* (GF). It is the continuous-time equivalent of discrete gradient descent, and is described by the ordinary differential equation (ODE) $\frac{d}{dt} W^l(t) = -\nabla_{W^l} F(\theta(t))$.

Stochastic gradient descent algorithm. When the number n of training samples is large, it is common to compute the gradient on a subset, called

a *batch* or *mini-batch*, rather than on the whole training set. Calling $F_i(\theta) := \ell(f^*(x_i), f(\theta; x_i))$ for $i \in [1, n]$, the *full gradient* is $\nabla_{W^l} F(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_{W^l} F_i(\theta)$, whereas for $\mathcal{B} \subset [1, n]$, the batch-gradient (approximate gradient) is $\hat{\nabla}_{W^l} F(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_{W^l} F_i(\theta)$. We call **gradient descent** (GD) optimizing F directly by computing the full gradients at each time steps, and we call **stochastic gradient descent** (SGD) optimizing F by sub-sampling batches among the training set at each time step, to the limit where there might be a single sample $x_t \in \{x_1, \dots, x_n\}$ at every time step.

Forward, backward passes and backpropagation. The computations of the gradients involved in the weight updates is called the **backward pass**, and they are computed recursively using the equations of backpropagation. The gradient *w.r.t.* any variable z can always be decomposed in the following way: $\nabla_z F_i(\theta) = \partial_2 \ell(f^*(x_i), f(\theta; x_i)) \nabla_z f(\theta; x_i)$, and the recursive equations read as:

$$\begin{aligned}
\nabla_{x^L} f(\theta; x_i) &= W^{L+1}, \quad \nabla_{h^L} f(\theta; x_i) = W^{L+1} \odot \sigma'(h^L), \\
\nabla_{W^{L+1}} f(\theta; x_i) &= x^L, \\
\nabla_{x^l} f(\theta; x_i) &= (W^{l+1})^\top \nabla_{h^{l+1}} f(\theta; x_i), \quad l \in [1, L-1] \\
\nabla_{h^l} f(\theta; x_i) &= (\nabla_{x^l} f(\theta; x_i)) \odot \sigma'(h^l), \quad l \in [1, L-1] \\
\nabla_{W^l} f(\theta; x_i) &= \nabla_{h^l} f(\theta; x_i)(x^{l-1})^\top, \quad l \in [1, L]
\end{aligned}$$

In contrast, the computation of the (pre-)activations h^l, x^l and the output $f(\theta; x)$ is called the **forward pass**.

Initial variance and learning rates. The choice of the initial variance and learning rates can have a huge impact on the behaviour of the network during and after training, especially at infinite-width (see Section 1.2 and Chapter 2). At finite width, a way to scale the variances and learning rates at initialization is discussed in (Glorot and Bengio, 2010; He et al., 2015) but an analysis beyond the first forward and backward pass is still lacking as the recursive equations and the presence of imbricated non-linearities quickly hinder any theoretical analysis. Yet, at infinite-width, the Tensor Program (Yang and Hu, 2021), which we introduce briefly in Section 1.2.5, allows to precisely analyze the magnitudes of the forward and backward passes at any time step for neural networks with i.i.d. initialization, and this type of analysis plays an important role in this thesis.

1.1.4 . Open questions and research directions

Modern neural networks require many ingredients to achieve a high level of performance on difficult tasks, such as large width and depth, normalization layers, residual connections, or adaptive gradient methods. Those ingredients are instrumental to the practical success of neural networks but are difficult to analyze mathematically, and it is unclear what the exact benefit of each of them is from a

theoretical standpoint. Moreover, many questions around neural networks remain unsolved: how is gradient-based learning able to find good values for the weights and converge given the sheer number of parameters ? How are they able to achieve (close to) zero loss when the objective is non-convex? Why do those models generalize so well when they could easily be over-fitting, with many paremeter values leading to zero loss and no control over the learned parameter θ^* at the end of training ? What do these models actually learn and what do the values of the learned weights mean ?

Organisation of the thesis

The rest of the thesis is organized as follows: Section 1.2 is dedicated to presenting the literature and mathematical tools around infinite-width limits, Section 1.3 highlights the main contributions of this thesis, Chapter 2 studies the infinite-width limit of deep networks in the integrable parameterization (see Section 1.2.3 for a definition) and Chapter 3 is devoted to studying the symmetries which emerge in the dynamics of infinitely wide two-layer networks. Finally, Chapter 4 studies the properties of optimization algorithms over the space of measures where neurons can dynamically be added or removed within the iterations of the algorithms.

1.2 . Infinite-width limits, a promising path to study the problem rigorously

1.2.1 . General context and motivation

A long line of research around infinitely wide neural networks

Infinite-width limits of neural networks have a long history tracing back to Barron (1993) and Neal (1995). The former shows that any function with sufficient regularity can be approximated uniformly on closed balls by two-layer neural networks with sigmoid-like activation functions (*i.e.*, bounded measurable functions satisfying $\lim_{-\infty} \sigma = 0$ and $\lim_{+\infty} \sigma = 1$) and the level of approximation achieved can be arbitrarily small provided the number of neurons in the first layer is allowed to *grow unbounded*. Pinkus (1999) goes even further and shows that uniform approximation on any compact set holds if and only if the activation function is not polynomial provided it is continuous. The caveat here is that although functions can be approximated with arbitrary precision on compact sets by neural networks, actually finding good parameter values that realize that approximation from finite data is difficult a priori. In a separate line of work, Neal (1995) adopts a Bayesian point of view and proves that the neural network function converges to a Gaussian process as the number of parameters tends to infinity when their distribution is Gaussian.

More recently, Bengio et al. (2006) demonstrate that the training objective of two-layer neural networks can be convexified (in a potentially infinite-dimensional

space) as soon as the loss function is convex if one considers an infinite number of neurons, which leads to algorithms potentially achieving the global minimum. Following this idea, Bach (2017) shows that infinitely wide two-layer networks with positively homogeneous activations form a class of functions which has favorable statistical properties: namely that in the presence of a lower-dimensional structure, the generalization error depends only on the dimension of the sub-space and not that of the ambient space. However, it is highlighted that minimizing the empirical risk in this setting (or its expected version) is a hard problem computationally.

Why study infinitely wide networks ?

Deep neural networks (even in their simplest form) are highly non-linear objects and their training dynamics correspond to the optimization of a complex, non-convex functions which makes them difficult to analyze theoretically. However, as presented above, important theoretical results have been obtained by considering limits where the number of neurons in a layer can go unbounded. As we discuss throughout the rest of this section, there has been a recent surge of interest in large width asymptotics due to a number of results which shed light on the behaviour of neural networks and help grasp why they work so well in practice. Among those results are **global convergence** of gradient descent (*e.g.*, Mei et al., 2018; Chizat and Bach, 2018; Wojtowytsch, 2020; Jacot et al., 2018), insights into their training dynamics by revealing a form of **implicit bias** (Chizat and Bach, 2020), as well as statistical results on the generalization properties of such models (Bach, 2017; Chizat and Bach, 2020).

In addition, with the acceleration made possible by advances in modern hardware, state-of-the-art neural networks have a huge number of parameters (up to *several hundreds of billions*) which makes studying the limit where the number of parameters tends to infinity not unreasonable. More than that, it is shown in (Nguyen and Pham, 2020) that the dynamics of infinitely wide networks track closely that of networks with sufficiently many neurons, and Yang and Hu (2021); Yang et al. (2022) demonstrate that theoretical results on infinitely wide networks can translate into practical insights on real-world finite-width networks whose behaviour is sometimes well described by the theory on their infinite-width counterpart.

In summary, infinite-width limits of neural networks appear as a neat way to adopt a theoretical standpoint while still leading to practical insights: they lend themselves nicely to theoretical analysis and represent a mathematically grounded approach that has borne fruit in furthering our understanding of certain questions around optimization and generalization.

Intuitive approach to the infinite-width limit

We present here informal ideas and calculations which allow to understand what the infinite-width limit of two-layer neural networks (sometime also called one-hidden-layer networks) consists in and how one might think of the limiting object. Rigorously taking this limit is often subtle and requires a lot of technical work, which is why many papers in the literature (which we review below) are concerned with dealing with these issues. Our work, in contrast, is not so much focused on the mathematical soundness of the limit but rather on exploiting the tools available to produce new results and ideas on infinitely wide networks.

Recall that a width-m two-layer neural network with real-valued output is a parametric function $f(\theta;\cdot) : \mathbb{R}^d \to \mathbb{R}$ defined as:

$$f(\theta; x) = \sum_{j=1}^{m} w_j^2 \sigma(x^\top w_j^1),$$

where $\theta = ((w_j^2, w_j^1))_{j\in[1,m]} \in (\mathbb{R} \times \mathbb{R}^d)^m$ is the list parameters consisting of the *input* weight matrix $w^1 = (w_1^1, \dots, w_m^1)$ and *output* weights $(w_1^2, \dots, w_m^2)$, and $\sigma : \mathbb{R} \to \mathbb{R}$ is the activation function. We stress that the w_j^1 represents the j-th neuron of the first layer ($l = 1$) and w_j^2 represents the j-th entry of the only neuron of the second layer ($l = 2$), so that the superscripts **do not** indicate exponents.

Taking the infinite-width limit is understood as taking the limit $m \to \infty$, which entails an infinite sum and with it issues related to convergence. A natural way to ensure that the sum remains finite as $m \to \infty$ is to add a scale factor in front of the sum that is a fixed negative power of m, *i.e.*, considering the new parameterization $f(\theta; x) = m^{-a} \sum_{j=1}^m w_j^2 \sigma(x^\top w_j^1)$ of the neural network with $a > 0$. Note that this does not change the class of functions we consider since the factor m^{-a} can be incorporated in the output weights w_j^2. Not all values of a guarantee convergence of the sum but large enough values ensure there are no issues.

One must think of the parameters $(w_j^2, w_j^1)_{j\in[1,m]}$ as random variables, this is the case at initialization and remains true during the course of training. As such, there are different modes of convergence for the sum. Typically, with i.i.d. parameters $((w_j^2, w_j^1))_{j\in[1,m]}$ (as is the case at initialization), one has a convergence in law for $a = 1/2$ (by the central limit theorem) and an almost-sure convergence for $a = 1$ (by the law of large numbers) as $m \to \infty$. It turns out that the scales $a = 1/2$ and $a = 1$ are widely studied in the infinite-width literature, the former is referred to as the Neural Tangent Kernel (NTK) parameterization and the latter is often referred to as the mean-field parameterization although we find that this denomination is somewhat ambiguous for networks with more than two-layers as the proper generalization to deeper layers comes with some difficulties in the "mean-field" setting (further discussed in Section 1.2.3), and we thus prefer the term *Integrable Parameterization* (IP) in reference to the fact that the re-normalized sum is absolutely convergent. As we will see shortly (see Sections 1.2.2, 1.2.3

and 1.2.5), these different scales lead to very different types of behaviour for the limiting model, and this thesis focuses on the integrable parameterization.

Parameterizations of networks of any depth

Generalizing to deeper networks the intuition presented above for the infinite-width limit of two-layer networks is not always straightforward. Indeed, one can still introduce factors m^{-a_l} for each layer l (or at least layers $l \geq 2$) with $a_l \geq 0$, leading to the parameterization of an L-hidden-layer network as

$$\begin{aligned} f(\theta; x) &= m^{-a_{L+1}} (w^{L+1})^\top x^L \\ x^l &= \sigma(h^l), \quad h^l = m^{-a_l} w^l x^{l-1}, \quad l \in [2, L], \\ x^1 &= \sigma(h^1), \quad h^1 = w^1 x \end{aligned}$$

where $w^{L+1} \in \mathbb{R}^m$, $w^l \in \mathbb{R}^{m \times m}$ for $l \in [2, L]$, $w^1 \in \mathbb{R}^{m \times d}$, and h^l denotes the pre-activations at layer l, and x^l the activations at layer l, and by default x^0 simply denotes the input x fed to the first layer of the network. We do not need to rescale the first layer since the sums in the inner products occurring there are always finite comprising as many terms as the input dimension d.

Taking the limit $m \to \infty$ when $L \geq 2$ makes things more difficult (even with $\sigma = \mathrm{id}$) as one has to handle imbricated infinite sums. As reviewed in the following sections, there are diverse mathematical frameworks and tools to take this limit depending on the parameterization under consideration, but the Tensor Program (described in Section 1.2.5) provides a comprehensive point of view for rigorously deriving the limit of any parameterization with techniques and ideas which emerged in the statistical physics literature in order to deal with random matrices whose size tends to infinity.

It turns out that to understand the training dynamics of such models in the limit $m \to \infty$, a more complete description of a ***parameterization*** of the network is given by adding scale factors m^{-b_l} ($b_l \geq 0$) to the standard deviation of the initial distribution of weights in layer l and scale factors m^{-c_l} for the learning rate of layer l applied to the weight updates. That is, the matrices w^l are initialized i.i.d. with a law such that $\tilde{W}^l_{ij}(0) := m^{b_l} w^l_{ij}(0)$ has variance one (or at least independent of m), and the update rule for the weights of layer l is given by $w^l(t+1) = w^l(t) - \eta_l m^{-c_l} \nabla_{w^l} F(\theta(t))$. This is called the abc-parameterization of a neural network in (Yang and Hu, 2021). There is a *redundancy* between the three scales a_l, b_l and c_l as only two of them suffice to provide a complete picture: one can for example always choose to initialize the matrices with unit variance (that is, $b_l = 0$) or alternatively use a unit learning rate ($c_l = 0$) without restricting the class of parameterizations considered. Indeed, considering the effective weight matrices $W^l(t) = m^{-a_l} w^l(t)$ that are actually used in the computation, one has

that $\nabla_{W^l} F(\theta(t)) = m^{-a_l} \nabla_{w^l} F(\theta(t))$ and thus

$$W^l(t) = m^{-(a_l+b_l)} \tilde{W}^l(0) - \eta m^{-(2a_l+c_l)} \sum_{s=0}^{t-1} \nabla_{W^l} F(\theta(s)). \tag{1.1}$$

It is then clear that starting from the same initialization $\tilde{W}^l(0)$, any parameterizations for which a_l+b_l and $2a_l+c_l$ have the same value will lead to the same effective weights and thus the same function. Therefore, any parameterization can be expressed with $b_l = 0$ or $c_l = 0$ (but not both at the same time). While Yang and Hu (2021) decide to drop the learning rate values (they consider mostly $c_l = 0$), we choose to consider ac-parameterizations where the initial weight matrices are always initialized with unit variance ($b_l = 0$). The name of a parameterization mostly refers to the choice of scale for the weights (a_l), *e.g.*, NTK ($a_l = 1/2$) or IP ($a_l = 1$) although the choice of learning rate (c_l) does have its importance.

1.2.2 . The NTK parameterization

When using i.i.d. Gaussian initialization for the weights of a neural network, scaling the initial standard deviations as $m^{-1/2}$ has appeared as a natural way to preserve the signal in first forward and backward passes (Glorot and Bengio, 2010; He et al., 2015). As detailed above this scaling of the initial standard deviation can equivalently be understood as a scale pre-factor of $m^{-1/2}$ in front of the weights which characterizes the NTK parameterization. With this scale factor, it is already understood since Neal (1995) that this results in a Gaussian process for the output of shallow networks at initialization as the width $m \to \infty$. Jacot et al. (2018), who first coined the term "Neural Tangent Kernel", go even further and prove that the training dynamics of fully-connected networks of any depth in this parameterization can be described as a kernel method with a specific kernel which we detail below. Yang (2020a) derives rigorously the generalization of this kernel description to any architecture.

The kernel description of the dynamics of the NTK parameterization in the infinite-width limit amounts to saying that the prediction function evolves as

$$f(\theta(t+1); x) = f(\theta(t); x) - \frac{\eta}{n} \sum_{i=1}^{n} \chi_{t,i} K(x, x_i)$$

for some kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, where $(x_i)_{i\in[1,n]}$ are the n samples in the training data set and $\chi_{t,i} := \partial F(\theta(t))/\partial f(\theta(t); x_i) = \partial_2 \ell(f^*(x_i), f(\theta(t); x_i))$ is the derivative of the loss on sample x_i at time t. As demonstrated in (Chizat et al., 2019), this kernel descent property in the NTK parameterization is the natural consequence of the fact that the scale (or magnitude) of weight updates is much smaller than that of the initial weights. First we explain how this intuitively leads to the kernel behaviour described above, and then we detail why this property holds in the infinite-width limit.

Linearization around the initialization for the NTK

Assume that at initialization, the gradients of the weights in the NTK parameterization are such that the updates are of much smaller magnitude then the initial weight values $W^l(0)$ for large m. The first parameter update on a batch of n samples reads as $\Delta\theta = -\frac{\eta}{n}\sum_{i=1}^{n}\nabla_\theta F_i(\theta(0))$, and the gradient *w.r.t.* the parameters can be decomposed as $\nabla_\theta F_i(\theta(0)) = \chi_{0,i}\nabla_\theta f(\theta(0); x_i)$. Assuming that $||\Delta\theta||$ is mall compared to $||\theta(0)||$, one can **linearize** the predictor around its initial parameters:

$$\begin{aligned} f(\theta(1); x) &= f(\theta(0) + \Delta\theta; x) \\ &\approx f(\theta(0); x) + \Delta\theta^\top \nabla_\theta f(\theta(0); x) \\ &= f(\theta(0); x) - \frac{\eta}{n}\sum_{i=1}^{n}\chi_{0,i}\nabla_\theta f(\theta(0); x_i)^\top \nabla_\theta f(\theta(0); x) \end{aligned}$$

which is exactly kernel descent with a kernel called the neural tangent kernel (Jacot et al., 2018), defined by $K_m(x, y) := \nabla_\theta f(\theta(0); x)^\top \nabla_\theta f(\theta(0); y)$ at width m. This is an inner product kernel, albeit in a space whose dimension goes to infinity as m becomes large. Two facts are noteworthy about this kernel: (i) it converges (almost surely) to a *deterministic* limiting kernel K_∞ as $m \to \infty$; and (ii) it actually stays constant in time in the infinite-width limit, that is $\lim_{m\to\infty} \nabla_\theta f(\theta(t); x)^\top \nabla_\theta f(\theta(t); y) = \lim_{m\to\infty} \nabla_\theta f(\theta(0); x)^\top \nabla_\theta f(\theta(0); y)$ for any $t \geq 0$. The second point is also a consequence of the fact that the weight updates in the NTK parameterization are much smaller in magnitude than the initial weights. This phenomenon, coined *"lazy training"* in (Chizat et al., 2019), cannot explain the feature learning as well as the transfer learning abilities of neural networks used in practice (in computer vision systems or in Large Language Models).

Features move infinitesimally in the NTK

Let us now explain why the weight updates have much smaller magnitude than the initial weights. To fix ideas, let us consider that the gradients are computed using a single sample and that the initialization is Gaussian, so that the initial (effective) weights read as $W^l(0) = m^{-1/2}\tilde{W}^l(0)$ for $l \in [2, L+1]$ and $W^1(0) = \tilde{W}^1(0)$ where $\tilde{W}^l(0)$ has i.i.d. entries following $\mathcal{N}(0, 1)$ for any l. Recall that the updates of the weights for any parameterization are given in Equation (1.1). The gradient *w.r.t.* the weights W^l are given by the backpropagation equations: $\nabla_{W^l} F(\theta(t)) = \chi_t \nabla_{h^l} f(\theta(t); x_t)(x_t^{l-1})^\top$, where $\chi_t = \partial F(\theta(t))/\partial f(\theta(t), x_t)$ is the loss derivative on the training sample x_t at time t. For the NTK parameterization, the first weight

updates are given by

$$\begin{aligned} W^1(1) &= \tilde{W}^1(0) - \eta\chi_0 \nabla_{h^1} f(\theta(0); x_0) x_0^\top \\ W^l(1) &= m^{-1/2}\tilde{W}^l(0) - \eta m^{-1}\chi_0 \nabla_{h^l} f(\theta(0); x_0)(x_0^{l-1})^\top, \quad l \in [2, L] \\ W^{L+1}(1) &= m^{-1/2}\tilde{W}^{L+1}(0) - \eta m^{-1}\chi_0 x_0^L. \end{aligned}$$

For layers $l \geq 2$, the factor m^{-1} in the weight update compared to the factor $m^{-1/2}$ present in the initial weight already hints towards the difference in magnitudes between those two contributions to the weight $W^l(1)$ for large m. The only thing left to analyze is the actual magnitude of the entries of the term $\nabla_{h^l} f(\theta(0); x_0)(x_0^{l-1})^\top$ as m grows large. The magnitude of the activations x_0^l for the NTK parameterization are well understood since Neal (1995): the factors $m^{-1/2}$ along with the i.i.d. Gaussian initialization guarantee that the initial forward pass is of order 1 (see Section 1.2.5) for more details. The scale of the gradients $\nabla_{h^l} f(\theta(0); x_0)$ however is not as straightforward and has to be derived recursively. Essentially, it follows from the equations of backpropagation that the coordinates of those gradients are of order $m^{-1/2}$. One thus deduces the relative magnitude of the updates $\Delta W^l = W^l(1) - W^l(0)$ compared to the initialization: $||\Delta W^l||/||W^l(0)||$ is of order $m^{-1/2}$ for the first and last layers $l \in \{1, L+1\}$ and of order m^{-1} for the intermediate layers $l \in [2, L]$.

The weights thus move away from their initialization only by an infinitesimal amount in the NTK parameterization in the large width limit. But how come then that the output function still evolves during training and does not stay at its initial value ? That is because although all the entries are all individually small compared to the initialization, they *collectively* induce a non-zero result in the inner products involved in the matrix multiplications of the forward pass. The Tensor Program precisely enables to derive the scales of the updates and the inner products rigorously as $m \to \infty$ and justifies the informal calculations presented above.

In the NTK parameterization, it thus appears that the evolution is only described in function space: the parameters of the network do not appear to move away significantly from their initialization. **Crucially**, it is even proved in (Yang and Hu, 2021) that as $m \to \infty$, the features x_t^l of any layer l at time t do not move significantly away from their initialization either, in the sense that for the same input $x \in \mathbb{R}^d$, $||x_t^l - x_0^l||^2/m$ converges towards 0 as $m \to \infty$ where the $1/m$ is simply here to re-normalize a sum that becomes infinite and which might explode otherwise. This is not surprising since the NTK dynamics are akin to learning with a kernel method which amounts to learning a linear predictor on top of fixed (albeit infinite-dimensional) features. Therefore, although the NTK parameterization and its infinite-width limit are appealing for their theoretical properties, they are not enough to capture the richness of the dynamics of real-world deep neural networks.

Global convergence of the NTK

Despite its drawbacks in terms of feature learning, the infinite-width limit of the NTK parameterization still yields interesting theoretical results such as the convergence of the objective to a global minimum. Indeed, it is shown in (Jacot et al., 2018) that if the loss is convex, the NTK dynamics lead to a convergence towards the global minimum. For example, for the squared loss objective $F(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(\theta; x_i) - y_i)^2$, the NTK dynamics lead to an exponentially fast convergence of $f(\theta(t); x_i)$ towards y_i in the infinite-width limit. This results from the fact that when considering the gradient flow of the objective (the limit of gradient descent when the step-size $\eta \to 0^+$), the prediction vector $\bar{y}_t = (f(\theta(t); x_i))_{i\in[1,n]}$ satisfies $\frac{d}{dt}(\bar{y}_t - y^*) = -\bar{K}_\infty(\bar{y}_t - y^*)$ where $y^* = (y_i)_{i\in[1,n]}$ is the vector of targets and $\bar{K}_\infty$ is the NTK matrix defined by $\bar{K}_{\infty,ij} = K_\infty(x_i, x_j)$. This leads to $\bar{y}_t = y^* + e^{-t\bar{K}_\infty}(\bar{y}_0 - y^*)$ which guarantees convergence of $\bar{y}_t$ towards y^* as $t \to \infty$ provided that $\bar{K}_\infty$ is positive definite.

1.2.3 . Integrable parameterization

Integrable parameterizations are characterized by the scale factor m^{-1} in front of the weights and have properties that are quite different from the NTK parameterization. This thesis focuses on IPs, therefore the literature and results around these types of models—which we review in this section—are of particular relevance to our work.

Infinite-width limit

With two-layers, the integrable parameterization has the form

$$f(\theta; x) = \frac{1}{m}\sum_{j=1}^{m} w_j^2 \sigma(x^\top w_j^1), \tag{1.2}$$

and is often referred to as a "mean-field" model as averages of this type are frequent in statistical physics where the study of systems with a growing number of particles interacting is common. The intuition is that when m (here the number of neurons, but it can be thought of as the number of particles of a system) is large, because of the $1/m$ term in front of the sum, the function will behave as a mean over some measure. Indeed, as $m \to \infty$, it is natural to replace the sum with an integral (one can think of the law of large numbers) against some probability measure $\mu \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R})$, leading to the parameterization

$$f(\mu; x) = \int_{(w^1,w^2)\in\mathbb{R}^{d+1}} w^2 \sigma(x^\top w^1) \mathrm{d}\mu(w^1, w^2) \tag{1.3}$$

in the infinite-width limit. To prevent the integral from diverging, we can restrict the class to functions parameterized by probability measures $\mu \in \mathcal{P}_2(\mathbb{R}^d \times \mathbb{R})$ which have finite second moment if σ has at most linear growth. In this setting,

any width-m two-layer network as in Equation (1.2) can be recovered with an atomic measure $\mu_m = \frac{1}{m}\sum_{j=1}^{m} \delta_{(mw_j^2, w_j^1)}$, where δ_w is the Dirac measure at w.

Dynamics over measures

The objective to minimize is now a functional F over the space of measures. In all generality, we seek to minimize the risk of a new class of functions:

$$\min_{\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})} \Big\{ F(\mu) := \mathbb{E}_{x\sim\rho}\left[\ell(f^*(x), f(\mu; x))\right] \Big\}. \tag{1.4}$$

At finite width, one uses (stochastic) gradient descent to decrease the value of the objective, but how does one proceed in the space of probability measures ? Tools originating from optimal transport theory have been developed to this end, and the answer is Wasserstein Gradient Flows (WGF), which are the equivalent of gradient descent on the space of measures with infinitesimal step-size. The corresponding dynamics are described by the partial differential equation (PDE) known as the **continuity equation** (see Ambrosio et al., 2005):

$$\begin{aligned} \partial_t \mu_t &= -\mathrm{div}(v_t \mu_t), \\ v_t &= -\nabla F'_{\mu_t} \end{aligned} \tag{1.5}$$

which is to be understood in the distributional sense. In Equation (1.5), the initial measure $\mu_0 \in \mathcal{P}_2(\mathbb{R}^{d+1})$ evolves according to a vector field $v_t = -\nabla F'_{\mu_t}$ given by the gradient of the first variation, or Fréchet derivative, F'_{μ_t} (a function from $\mathbb{R}^{d+1}$ to $\mathbb{R}$) of the functional F at μ_t. More details and mathematical background on Wasserstein gradient flows, the first variation of functionals over probability measures and the continuity equation are provided in Section 1.2.4.

The natural interpretation of that equation is that at any given time t, mass is *displaced* (or advected) according to some vector field v_t, thereby changing the distribution of mass μ_t at time t. In fact, an alternative description of the continuity equation can be provided using the point of view of a system of infinitely many interacting particles: consider an initial distribution μ_0 of particles $w \in \mathbb{R}^{d+1}$, and consider the flow $X_t(w)$ defined for any $w \in \mathbb{R}^{d+1}$ by

$$\begin{aligned} X_0(w) &= w, \\ \frac{d}{dt} X_t(w) &= v_t(X_t(w)). \end{aligned} \tag{1.6}$$

$X_t(w) \in \mathbb{R}^{d+1}$ represents the position at time t of a particle initially located at $w \in \mathbb{R}^{d+1}$ and which interacts with all the other particles (at other locations) through the velocity field v_t. Then, given the flow X_t, the solution to the continuity equation (1.5) starting from μ_0 is given by the push-forward $\mu_t = X_{t\#}\mu_0$ of the measure μ_0 by the map $X_t(\cdot)$. Said differently, the measure μ_t is simply the distribution of particles at time t, initially distributed according to μ_0, and which

have evolved according to the system (1.6). Note that from this point of view, the distribution of particles at time t determines the measure μ_t and thus also the velocity field v_t, which in turn will determine in which direction particles evolve, so that particles actually interact since the velocity of a particle at any given time is determined by the position of all the other particles.

Importantly, the WGF (1.5) **recovers gradient flow** on the objective of finite-width networks. Indeed, for an atomic initial measure $\mu_0 = \frac{1}{m}\sum_{j=1}^m \delta_{(w^1(0),w^2(0))}$, the WGF (1.5) is exactly continuous-time gradient descent on the parameters of a finite-width network. In other words, the WGF $(\mu_{m,t})_{t\geq 0}$ starting from an initial atomic measure $\mu_{m,0} = \frac{1}{m}\sum_{j=1}^m \delta_{(w^1(0),w^2(0))}$ is of the form $\mu_{m,t} = \frac{1}{m}\sum_{j=1}^m \delta_{(w_j^1(t),w_j^2(t))}$, and the parameters $\theta(t) = ((w_j^1(t), w_j^2(t)))_{j\in[1,m]}$ are in fact given by the gradient flow $\theta'(t) = -m\nabla F_m(\theta(t))$ on the finite-width objective defined by $F_m(\theta) = \mathbb{E}_{x\sim\rho}\left[\ell(f^*(x), f_m(\theta; x))\right]$ with $f_m(\theta; x) = \frac{1}{m}\sum_{j=1}^m w_j^2\sigma(x^\top w_j^1)$. The factor m in the gradient flow is to compensate for the $1/m$ in the definition of f_m which downscales the gradients. Conversely, if $\theta(t)$ is the gradient flow of the finite-width objective F_m, that is $\theta'(t) = -m\nabla F_m(\theta(t))$, then the atomic measure $\mu_{m,t} = \frac{1}{m}\sum_{j=1}^m \delta_{(w_j^1(t),w_j^2(t))}$ is the WGF of the functional F starting from $\mu_{m,0}$. Furthermore, if $\mu_{m,0}$ converges (in Wasserstein distance) to μ_0 as m $to\infty$, then $\mu_{m,t}$ converges, as $m\to\infty$, to the WGF of the functional F starting from μ_0 on any bounded time interval. For more details on the equivalence between the WGF for atomic measures and the finite-width gradient flow, see Section 1.2.4.

Literature review

Mean-field models are ubiquitous in mathematical physics but IPs have only recently been studied as models for infinitely wide neural networks. They have rapidly caught on as an interesting approach to studying first two-layer networks and then deeper ones. The questions that arise when studying the infinite-width limit of networks in the integrable parameterization are of diverse nature: is there existence and/or uniqueness of the solution to the continuity equation (1.5) in the typical setting of neural networks ? How far are the dynamics of finite-width networks *w.r.t.* the infinite-width description ? Can one give quantitative bounds depending on m ? Is there convergence of the dynamics as $t\to\infty$? How do these models behave numerically ?

The case of two-layer networks. A flurry of works study those questions from a mathematical perspective for two-layer networks (Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Chizat and Bach, 2018; Araújo et al., 2019; Wojtowytsch, 2020; Sirignano and Spiliopoulos, 2020), and establish the well-posedness of Equation (1.5) in the context of two-layer neural networks under mild assumptions on the loss function and the activation function, as well as the convergence, as the number of neurons m goes to infinity, of the finite-width gradient flow dy-

namics to the dynamics in the space of measures given by Equation (1.5). What is more, the convergence of μ_t towards a *global minimizer* of the objective F as $t \to \infty$ is also proved in (Mei et al., 2018; Chizat and Bach, 2018; Wojtowytsch, 2020) when the loss ℓ is convex under mild assumptions on the initialization μ_0.

The global convergence result requires technical proofs, but it is noteworthy that although the finite-width objective is non-convex for two-layer networks, if the loss ℓ is convex, since the parameterization of an infinite-width network by a probability measure as in Equation (1.3) is *linear* in the measure μ, the objective F is now convex in μ. This is a good property for optimization but is not enough to guarantee global convergence of the WGF in general, the good property is that of *displacement convexity* (convexity along geodesics) but it does not always hold in the context of neural networks. It is important to observe that the global convergence results for two-layer networks in the integrable parameterizations are of a *different nature* than those discussed for the NTK parameterization. Indeed, in the integrable parameterization, the dynamics given by the WGF (1.7) are truly *non-linear* and imply that the weights evolve non-trivially away from their initialization: features are actually learned by the network as training progresses.

Statistical results. For two-layer networks, the class of functions represented by the infinite-width limit of the integrable parameterization also has interesting statistical properties. Bach (2017) studies their statistical and approximation properties and shows that when the target function only depends on the projection on a (unknown) low-dimensional sub-space, these networks circumvent the curse of dimensionality with approximation and generalization bounds which depend exponentially on the dimension of the sub-space only.

In the context of binary classification, Chizat and Bach (2020) show that for exponentially tailed losses, the WGF (1.5) leads to a predictor which is a max-margin classifier as $t \to \infty$. This is a form of *implicit bias* of the gradient descent dynamics: the WGF does not converge to any global minimizer, but to one that realizes the maximum margin, and thus has favorable generalization properties. Indeed, when there is a low-dimensional sub-space for which the projection of the data has sufficiently large inter-class distance, the margin is independent of the ambient dimension, leading to an upper bound on the probability of misclassification which only depends on the dimension of the sub-space.

The strong results discussed above for two-layer networks along with the fact that weights do actually move away from their initialization demonstrate that the infinite-width limit of integrable parameterizations is a promising research avenue to further our understanding of neural networks and justify the growing body of work around those models.

Multi-layer networks. Generalizing the result obtained for two-layer networks to deeper networks is not easy (see Nguyen and Pham, 2020). Indeed, the

particularity of two-layer networks is that their is an *exchangeability* of neurons due to the invariance by permutation in the sum in Equation (1.2). For three layers or more, some weights will appear in all the terms of the sum, leading to complications (we are not summing over independent parts of the parameter set) and the basic exchangeability is lost. However, there is still a number of works which study deeper networks in the integrable parameterization (Nguyen and Pham, 2020; Fang et al., 2020; Sirignano and Spiliopoulos, 2021; Araújo et al., 2019). Yet, they all point out the difficulty of describing properly the dynamics of the infinite-width limit when the network has more than three layers, and they all present different descriptions of the dynamics of the infinite-width limit, either requiring specific assumptions or inducing undesirable properties. Among the difficulties that arise in deeper versions of the integrable parameterization, of particular interest are the questions of how to take the limit (sequentially or all layers at once), how to scale properly the layers and their learning rates to obtain non-degenerate dynamics, and how to describe the resulting dynamics. We shall see in Section 1.2.5 that the Tensor Program (Yang, 2019, 2020a,b; Yang and Hu, 2021) allows to answer these questions rigorously.

Despite the difficulties that arise for deep networks, Nguyen and Pham (2020) and Sirignano and Spiliopoulos (2021) are still able to prove global convergence results for networks with three layers or more under specific sets of assumptions. In addition, aside from the convergence results of the finite-width dynamic to an idealized limit dynamic as the width m goes to infinity, Fang et al. (2020), Nguyen and Pham (2020) and Araújo et al. (2019) provide quantitative bounds on the distance between the finite-width dynamic and its idealized counterpart at infinite-width *w.r.t.* the number of neurons m roughly scaling as $m^{-1/2}$.

It is however clear from the literature that the behaviour of integrable parameterizations with more than four layers and i.i.d. initialization is degenerate and that gradients of different layers are of different magnitudes *w.r.t.* the width m. For example, it is stated in (Araújo et al., 2019) and (Nguyen and Pham, 2020) that under i.i.d. initialization with more than four layers, weights of different layers evolve independently of other layers in the infinite-width limit, and furthermore all the weights in the same layer evolve by the same deterministic quantity which depends only on time. Although these pitfalls are clearly identified, the setting and/or the assumptions are tweaked (*e.g.*, non-i.i.d. initialization, only training certain layers, restricted number of layers) in order to circumvent them and establish a theory of the infinite-width limit for deep networks. It is the object of Chapter 2 of this thesis to tackle these issues in the standard setting used in practice taking an alternative approach using the Tensor Program.

1.2.4 . Evolution equations in the space of measures

In this section, we review the mathematical tools around functionals on spaces of measures and Wasserstein gradient flows, as this is a core part of the work presented in this thesis, especially in Chapter 3. The notions we discuss here are

presented in great detail in (Ambrosio et al., 2005) and (Santambrogio, 2017, 2015).

Spaces of probability measures and Wasserstein distances

Let $q \geq 1$ be a scalar, and consider the space $\mathcal{P}_q(\mathbb{R}^p)$ of probability measures on $\mathbb{R}^p$ satisfying $\int ||x||^q \mathrm{d}\mu(x) < \infty$. One can define a distance on $\mathcal{P}_q(\mathbb{R}^p)$, called the q-Wasserstein distance, by

$$W_q(\mu, \nu) := \left(\min_{\gamma \in \Gamma(\mu,\nu)} \int ||x - y||^q \mathrm{d}\gamma(x, y) \right)^{1/q}$$

where for any $\mu, \nu \in \mathcal{P}_q(\mathbb{R}^p)$, $\Gamma(\mu, \nu)$ is the set of *transport plans* from μ to ν, *i.e.*, the set of probability measures over $\mathbb{R}^p \times \mathbb{R}^p$ whose marginals are equal to μ and ν. Formally

$$\Gamma(\mu, \nu) = \{\gamma \in \mathcal{P}(\mathbb{R}^p \times \mathbb{R}^p) \, : \, \gamma_{\#}\pi_x = \mu, \gamma_{\#}\pi_y = \nu\}$$

where $\pi_x : (x, y) \in \mathbb{R}^p \times \mathbb{R}^p \mapsto x$ and $\pi_y : (x, y) \in \mathbb{R}^p \times \mathbb{R}^p \mapsto y$ are the canonical projections onto the first and second component respectively. The Wasserstein distance (also know as the Monge-Kantorovich or Kantorovich-Rubinstein distance) comes from optimal transport theory whose objective is to understand how to move mass from one distribution to another optimally according to some cost.

The space $\mathcal{P}_q(\mathbb{R}^p)$ endowed with the distance W_q forms a complete and convex metric space $(\mathcal{P}_q(\mathbb{R}^p), W_q)$ for which convergence according to the distance W_q is roughly equivalent to the weak convergence of measures (sometimes also called narrow convergence) as the following holds: for any sequence $(\mu_n)_{n\in\mathbb{N}}$ and μ in $\mathcal{P}_q(\mathbb{R}^p)$, one has that

$$W_q(\mu_n, \mu) \to 0 \text{ if and only if } \mu_n \rightharpoonup \mu \text{ and } \int ||x||^q \mathrm{d}\mu_n(x) \to \int ||x||^q \mathrm{d}\mu(x),$$

where $\mu_n \rightharpoonup \mu$ denotes the weak convergence of measures, that is

$$\int \varphi \mathrm{d}\mu_n \to \int \varphi \mathrm{d}\mu$$

for every φ in the space $\mathcal{C}_b(\mathbb{R}^p)$ of continuous and bounded functions over $\mathbb{R}^p$. If one replaces the whole domain $\mathbb{R}^p$ by a compact subset $\Omega \subset \mathbb{R}^p$, the equivalence statement above holds true without the condition on the convergence of the integral of the norm.

Note that for any $q_2 \geq q_1 \geq 1$, Jensen's inequality ensures that for any $\gamma \in \mathcal{P}(\mathbb{R}^p \times \mathbb{R}^p)$, $\left(\int ||x - y||^{q_1} \mathrm{d}\gamma(x, y)\right)^{1/q_1} \leq \left(\int ||x - y||^{q_2}\right)^{1/q_2} \mathrm{d}\gamma(x, y)$, which implies that $W_{q_1}(\mu, \nu) \leq W_{q_2}(\mu, \nu)$. In this thesis, we will focus only on the space $(\mathcal{P}_2(\mathbb{R}^p), W_2)$.

Functionals of probability measures and first variation

A functional F over $\mathcal{P}_2(\mathbb{R}^p)$ is a function $F : \mathcal{P}_2(\mathbb{R}^p) \to \mathbb{R}$. One would like to define a notion of derivative over $\mathcal{P}_2(\mathbb{R}^p)$ similarly to the notion of derivative or gradient in finite dimension, but the issue is that $\mathcal{P}_2(\mathbb{R}^p)$ is an infinite-dimensional convex space and not a euclidean space, so that care has to be taken when defining that notion which is a bit more subtle in this context. Given $\mu \in \mathcal{P}_2(\mathbb{R}^p)$, the *first variation* or *Fréchet derivative* of F at μ, if it exists, is a measurable function from $\mathbb{R}^p$ to $\mathbb{R}$, denoted by $\frac{\delta F}{\delta \mu}(\mu)$ or simply F'_μ , satisfying, for any suitable perturbation ν,

$$\frac{d}{dt} F(\mu + t\nu)\Big|_{t=0} = \int \frac{\delta F}{\delta \mu}(\mu) \mathrm{d}\nu.$$

Admissible perturbations ν have to satisfy $\mu + t\nu \in \mathcal{P}_2(\mathbb{R}^p)$ for sufficiently small t, and are therefore chosen of the form $\nu = \tilde{\nu} - \mu$ where $\tilde{\nu}$ is a probability measure with bounded density and compact support. Note that ν is not a probability measure but rather lies in the set $\mathcal{M}(\mathbb{R}^p)$ of signed measures, and as the difference of two probability measures (of total mass 1), satisfies $\int \mathrm{d}\nu = 0$, so that the first variation is defined up to additive constants, but is unique modulo that invariance.

Note that the definition of the first variation is akin to the equality satisfied by the gradient in finite dimension: $\frac{d}{dt} f(x + ty)\Big|_{t=0} = \langle \nabla f(x), y \rangle$, and as such the integral $\int \frac{\delta F}{\delta \mu}(\mu) \mathrm{d}\nu$ can be interpreted as some kind of inner product (or rather a duality bracket) $\langle \frac{\delta F}{\delta \mu}(\mu), \nu \rangle$ which represents the "action" of the measure ν on the measurable function $\frac{\delta F}{\delta \mu}(\mu)$.

Classical examples of functionals and their first variations. Given $V : \mathbb{R}^p \to \mathbb{R}$, $W : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$, one can define the following functionals,

$$\begin{aligned}
\mathcal{V}(\mu) &= \int V(x) \mathrm{d}\mu(x), \\
\mathcal{W}(\mu) &= \int W(x, y) \mathrm{d}\mu(x) \mathrm{d}\mu(y), \\
\mathcal{F}(\mu) &= \begin{cases} \int f\left(\frac{d\mu}{d\lambda}\right) \mathrm{d}\lambda(x) \text{ if } \mu \in L^1(\lambda) \\ +\infty \text{ otherwise} \end{cases},
\end{aligned}$$

where λ denotes the Lebesgue measure over $\mathbb{R}^p$, and it is easily checked that their respective first variations are

$$\begin{aligned}
\frac{\delta \mathcal{V}}{\delta \mu}(\mu)(x) &= V(x), \\
\frac{\delta \mathcal{W}}{\delta \mu}(\mu)(x) &= \int W(x, y) \mathrm{d}\mu(y) + \int W(y, x) \mathrm{d}\mu(y), \\
\frac{\delta \mathcal{F}}{\delta \mu}(\mu)(x) &= f'\left(\frac{d\mu}{d\lambda}(x)\right) \text{ for } \mu \in L^1(\lambda).
\end{aligned}$$

First variation for infinitely wide two-layer networks. In the case of the objective functional F defined in Equation (1.4), it is straightforward to derive that for any $w = (w^1, w^2) \in \mathbb{R}^{d+1}$,

$$F'_\mu(w) = \frac{\delta F}{\delta \mu}(\mu)(w) = \int_x \partial_2 \ell(f^*(x), f(\mu; x)) w^2 \sigma(x^\top w^1) \mathrm{d}\rho(x).$$

First variation of the 2-Wasserstein distance. The definition of the 2-Wasserstein distance can be seen as a constrained minimization problem over an infinite-dimensional space, where the constraint is that the probability measure $\gamma \in \mathcal{P}(\mathbb{R}^p \times \mathbb{R}^p)$ must have marginals equal to μ and ν. As Lagrange duality allows to handle constrained optimization problems in finite dimension, Kantorovich developed a theory, called Kantorovich duality, which allows to deal with constrained optimization problems in the space of measures, and in particular optimal transport problems. The dual problem for the optimal transport problem associated with Wasserstein distances reads as:

$$\max_{\varphi, \psi \in \Lambda} \int \varphi \mathrm{d}\mu + \int \psi \mathrm{d}\nu - \int \Big(\varphi(x) + \psi(y)\Big) \mathrm{d}\gamma(x, y),$$
$$\Lambda := \Big\{ \varphi, \psi \in \mathcal{C}_b(\mathbb{R}^p) \ : \ \varphi(x) + \psi(y) \leq ||x - y||^q \Big\}.$$

Alternative descriptions of the dual problem which allow to study it in more depth are quite involved and would require introducing new notations and concepts. This is not the object of this thesis and we refer to (Santambrogio, 2017)[Section 4.1] for a detailed presentation. However, we note that this duality allows to derive the form of the solution to the optimal transport problem defining the Wasserstein distances. In particular, it can be shown (see Santambrogio, 2017[Theorem 4.2]) that if μ is absolutely continuous, there exists a map $T : \mathbb{R}^p \to \mathbb{R}^p$ called an *optimal transport map*, and a function $\varphi : \mathbb{R}^p \to \mathbb{R}$, called a *Kantorovich potential* (coming from Kantorovich duality), satisfying the three following conditions:

(i) $\frac{\delta W_2}{\delta \mu}(\mu, \nu) = \varphi$,

(ii) the push-forward measure $\gamma^* := (\mathrm{id}, T)_\# \mu$ realizes the minimum in the definition of W_2,

(iii) $\nabla \varphi = \mathrm{id} - T$.

Wasserstein gradient flows and optimality conditions

We now turn our attention to Wasserstein gradient flows which are the main theoretical tool to optimize functionals over $\mathcal{P}_2(\mathbb{R}^p)$. Let us consider a functional $F : \mathcal{P}_2(\mathbb{R}^p) \to \mathbb{R}$ which admits a first variation at every $\mu \in \mathcal{P}_2(\mathbb{R}^p)$ that is differentiable almost everywhere. Starting from an initial measure $\mu_0 \in \mathcal{P}_2(\mathbb{R}^p)$,

the Wasserstein gradient flow of the functional F is a path $(\mu_t)_{t\geq 0}$ in the space $\mathcal{P}_2(\mathbb{R}^p)$ satisfying, *in the sense of distributions*, the continuity equation

$$\partial_t \mu_t = -\text{div}\left(-\nabla\left(\frac{\delta F}{\delta \mu}(\mu_t)\right)\mu_t\right). \tag{1.7}$$

A pair $(\mu_t, v_t)_{t\geq 0}$ consisting of a path in $\mathcal{P}_2(\mathbb{R}^p)$ and a time-dependent vector field $v_t : \mathbb{R}^p \to \mathbb{R}^p$ satisfies the continuity equation $\partial_t \mu_t = -\text{div}(v_t \mu_t)$ in the sense of distributions if for any test function φ in the space $\mathcal{C}_c^1(\mathbb{R}^p)$ of continuously differentiable and compactly supported functions, it holds:

$$\frac{d}{dt}\int \varphi \mathrm{d}\mu_t = \int \nabla\varphi^\top v_t \mathrm{d}\mu_t.$$

In particular, setting φ to be the constant function equal to 1 shows that the total mass is preserved in the Wasserstein gradient flow: mass is neither injected nor lost along the flow, but simply displaced.

Minimizing movement scheme. Where does this equation come from ? In metric spaces such as the space of probability measures, it is difficult to define a notion of derivative due to the lack of a linear structure, and one typically resorts to *minimizing movement schemes*. In $\mathbb{R}^p$ (or any Hilbert space), let $\tau > 0$ be a parameter and consider a differentiable function $f : \mathbb{R}^p \to \mathbb{R}$, as well as a sequence $(x_k)_{k\geq 0}$ in $\mathbb{R}^p$ satisfying for any k,

$$x_{k+1} \in \operatorname*{argmin}_y f(y) + \frac{1}{2\tau}||y - x_k||^2.$$

This is called a minimizing movement scheme: indeed, one tries to minimize f while staying close to the current estimate x_k. The sequence of estimates satisfies $\nabla f(x_{x+1}) = -\frac{x_{k+1}-x_k}{\tau}$, and considering a function $\tilde{x} : \mathbb{R}_+ \to \mathbb{R}^p$ interpolating the x_k (*i.e.*, $\tilde{x}(k\tau) = x_k$), one has $\frac{\tilde{x}(k\tau+\tau)-\tilde{x}(k\tau)}{\tau} = -\nabla f(\tilde{x}((k+1)\tau)$ and as $\tau \to 0^+$ one gets a curve satisfying $\tilde{x}'(t) = -\nabla f(\tilde{x}(t))$, which is exactly the gradient flow of the function f (the minimizing movement scheme is in fact the implicit Euler scheme for discretizing the gradient flow). The convergence of the interpolating function to a gradient flow as $\tau \to 0^+$ can be made rigorous if f is continuously differentiable or if it is convex (see Santambrogio, 2017[Proposition 2.3]).

Going back to the space of measures, one can try to derive a similar minimizing movement scheme as

$$\mu_{k+1} \in \operatorname*{argmin}_\mu F(\mu) + \frac{1}{2\tau} W_2(\mu, \mu_k)^2. \tag{1.8}$$

To understand the conditions that μ_{k+1} should satisfy, we first need to explain what the optimality conditions are for functionals on $\mathcal{P}_2(\mathbb{R}^p)$. Similarly to the finite dimensional case, if one wishes to minimize $G(\mu)$ for some functional G

admitting a first variation at every μ, then the optimality of a certain minimizer μ^* is related to the value of the first variation $\frac{\delta G}{\delta \mu}(\mu^*)$ of G at μ^*. It is stated in (Santambrogio, 2015)[Proposition 7.20] that under the appropriate regularity assumptions, for a minimizer μ^* of G, the first variation $\frac{\delta G}{\delta \mu}(\mu^*)$ must be constant on the support of μ^*. For the minimizing movement scheme (1.8) above, the first variation of the squared 2-Wasserstein distance is related to the optimal transport from μ_{k+1} to μ_k, and the optimality condition translates to

$$\frac{\delta F}{\delta \mu}(\mu_{k+1}) + \frac{1}{\tau}\varphi_\tau = C$$

for some constant C, where φ_τ is the Kantorovich potential associated with the transport from μ_{k+1} to μ_k. Optimal transport theory tells us that $\nabla\varphi_\tau = x - T_\tau(x)$ for the optimal transport map T_τ from μ_{k+1} to μ_k, so that, by taking the gradient of the equation above, the following holds:

$$\nabla\left(\frac{\delta F}{\delta \mu}(\mu_{k+1})\right)(x) = -\frac{T_\tau(x) - x}{\tau}.$$

Therefore, if we wish to transport mass from μ_k to minimize the quantity in (1.8), the displacement of mass $\frac{T_\tau(x)-x}{\tau}$ at any point x must be equal to the vector field $v_{k+1}(x) := -\nabla\left(\frac{\delta F}{\delta \mu}(\mu_{k+1})\right)(x)$. As $\tau \to 0^+$, it follows that the change in mass induced by the minimizing movement scheme must satisfy the Wasserstein gradient flow equation (1.7): at any time t, mass located at x is displaced with velocity $v_t(x) = -\nabla\left(\frac{\delta F}{\delta \mu}(\mu_t)\right)(x)$. The proof for the convergence of the iterated minimization scheme above is technical in general metric spaces and described in (Santambrogio, 2015, 2017).

Properties of the Wasserstein gradient flow. The existence of a solution to the gradient flow problem (or the continuity equation) is guaranteed when enough regularity is assumed on the functional F (and the gradient of its first variation $\nabla\frac{\delta F}{\delta \mu}$ which is the negative velocity in the continuity equation). As for the uniqueness, it often requires some notion of convexity (*e.g.*, geodesic semi-convexity) on F or some assumptions on the initial measure (*e.g.*, that it has a density *w.r.t.* the Lebesgue measure) to be guaranteed in general. For infinitely wide two-layer networks, (Chizat and Bach, 2018; Mei et al., 2018; Wojtowytsch, 2020) show the existence and uniqueness of the the WGF (1.7) under mild assumptions. However, the the regularity assumptions are not met by ReLU which must be dealt with separately. We discuss this in the next paragraph.

As for gradient flows in finite dimension, it can be shown that the Wasserstein gradient flow always decreases the functional one is trying to minimize, as the following holds:

$$\frac{d}{dt}F(\mu_t) = -\int \left|\left|\nabla\left(\frac{\delta F}{\delta \mu}(\mu_t)\right)\right|\right|^2 \mathrm{d}\mu_t \leq 0.$$

Equivalence between WGF for atomic measures and finite-width GF

We detail here the derivation of the relationship between the WGF (1.7) of the infinite-dimensional objective F over the space $\mathcal{P}_2(\mathbb{R}^{d+1})$ and the gradient flow of the objective F_m on the weights of a width-m two-layer network. First we derive the flow description of the continuity equation and then we proceed to show the equivalence between the WGF for atomic measures and the finite-width gradient flow.

Flow description of the continuity equation. Let $(\mu_t, v_t)_{t\geq 0}$ be a pair formed by a path in the space $\mathcal{P}_2(\mathbb{R}^p)$ and a time-dependent vector field $v_t : \mathbb{R}^p \to \mathbb{R}^p$, satisfying, in the sense of distributions, the continuity equation

$$\partial_t \mu_t = -\mathsf{div}(v_t \mu_t),$$

and consider the flow defined, for any $w \in \mathbb{R}^d$, by the ODE

$$\begin{aligned} X_0(w) &= w, \\ \frac{d}{dt} X_t(w) &= v_t\left(X_t(w)\right). \end{aligned}$$

Then, it holds that for any $t \geq 0$, $\mu_t = (X_t)_{\#}\mu_0$. Indeed, defining $\nu_t := (X_t)_{\#}\mu_0$, since X_0 is the identity map of $\mathbb{R}^d$, it holds $\nu_0 = \mu_0$, and the uniqueness of the solution to the continuity equation will suffice to conclude to equality. Let $t \geq 0$, and $\varphi \in \mathcal{C}^1_c(\mathbb{R}^p)$. We have

$$\begin{aligned} \frac{d}{dt}\int \varphi \mathrm{d}\nu_t &= \frac{d}{dt}\int \varphi \circ X_t \mathrm{d}\mu_0 \\ &= \int \left\langle (\nabla\varphi)\circ X_t, \frac{d}{dt}X_t \right\rangle \mathrm{d}\mu_0 \\ &= \int \langle (\nabla\varphi)\circ X_t, v_t \circ X_t\rangle \,\mathrm{d}\mu_0 \\ &= \int \nabla\varphi^\top v_t \mathrm{d}\nu_t, \end{aligned}$$

which means ν_t satisfies the continuity equation in the sense of distributions with initial condition $\nu_0 = \mu_0$. The uniqueness of such a solution allows to conclude that $\nu_t = \mu_t$ for any $t \geq 0$.

WGF - GF equivalence. Call $\phi : \mathbb{R}^{d+1} \times \mathbb{R}^d \to \mathbb{R}$ defined, for any $w = (w^1, w^2) \in \mathbb{R}^d \times \mathbb{R}$ by $\phi(w; x) = w^2 \sigma(x^\top w^1)$. For any $m \in \mathbb{N}$, let f_m denote the integrable parameterization of a width-m two-layer network, defined by $f_m(\theta; x) = \frac{1}{m}\sum_{j=1}^m \phi(\theta_j; x)$ where $\theta_j = (w_j^1, w_j^2) \in \mathbb{R}^{d+1}$. In addition, let F_m be the finite-width objective defined, for any $\theta \in (\mathbb{R}^{d+1})^m$, by $F_m(\theta) = \mathbb{E}_\rho\left[\ell(f^*(x), f_m(\theta; x))\right]$,

and let F be the infinite-width objective over measures, defined, for any $\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})$, by $F(\mu) = \mathbb{E}_\rho\left[\ell(f^*(x), f(\mu; x))\right]$ where $f(\mu; x) = \int \phi(w; x)\mathrm{d}\mu(w)$.

First, observe that defining the atomic measure $\mu_m = \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j}$, it holds that $f(\mu_m; \cdot) = f_m(\theta; \cdot)$, and consequently $F(\mu_m) = F_m(\theta)$. Next, notice that because the first variation of F at μ is given, for any $w \in \mathbb{R}^{d+1}$, by $F'_\mu(w) = \int_x \partial_2\ell(f^*(x), f(\mu; x))\phi(w; x)\mathrm{d}\rho(x)$, it then follows that its gradient is given by $\nabla F'_\mu(w) = \int_x \partial_2\ell(f^*(x), f(\mu; x))\nabla_w\phi(w; x)\mathrm{d}\rho(x)$. Hence the following equality: $m\nabla_{\theta_j} F_m(\theta) = \nabla F'_{\mu_m}(\theta_j)$.

Consider the initial weights $((w_1^1(0), w_1^2(0)), \dots, (w_m^1(0), w_m^2(0)))$, and the initial atomic measure $\mu_{m,0} = \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j(0)}$ where $\theta_j(0) = (w_j^1(0), w_j^2(0))$. Let $(\mu_{m,t})_{t\geq 0}$ be the WGF of the objective F starting from $\mu_{m,0}$, and let X_t be the flow associated to the continuity equation with vector field $v_t = -\nabla F'_{\mu_{m,t}}$ as in the previous paragraph. It holds that $\mu_{m,t} = (X_t)_{\#}\mu_{m,0}$ and as a push-forward of the atomic measure $\mu_{m,0}$, $\mu_{m,t}$ is also an atomic measure and its masses are located at the images of the masses of $\mu_{m,0}$ by the push-forward map, *i.e.*, $\mu_{m,t} = \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j(t)}$ with $\theta_j(t) := X_t(\theta_j(0))$. Showing that $\theta(t) = (\theta_j(t))_{j\in[1,m]}$ is a gradient flow for F_m easily follows from the ODE satisfied by the flow X_t:

$$\begin{aligned}\frac{d}{dt}\theta_j(t) &= \frac{d}{dt}X_t(\theta_j(0)) \\ &= -\nabla F'_{\mu_{m,t}}\Big(X_t(\theta_j(0))\Big) \\ &= -m\nabla_{\theta_j} F_m(\theta_j(t)).\end{aligned}$$

Conversely, define $(\theta(t))_{t\geq 0}$ as the gradient flow of F_m starting from $\theta(0) = (\theta_j(0))_{j\in[1,m]}$, that is $\frac{d}{dt}\theta(t) = -m\nabla F_m(\theta(t))$, and define $\mu_{m,t} := \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j(t)}$. Then, showing that $\mu_{m,t}$ is a WGF for F easily follows from the ODE satisfied by $\theta(t)$. Indeed, let $\varphi \in \mathcal{C}_c^1(\mathbb{R}^{d+1})$. It holds:

$$\begin{aligned}\frac{d}{dt}\int \varphi\mathrm{d}\mu_{m,t} &= \frac{d}{dt}\left(\frac{1}{m}\sum_{j=1}^m \varphi(\theta_j(t))\right) \\ &= \frac{1}{m}\sum_{j=1}^m \nabla\varphi(\theta_j(t))^\top\left(\frac{d}{dt}\theta_j(t)\right) \\ &= \frac{1}{m}\sum_{j=1}^m \nabla\varphi(\theta_j(t))^\top\left(-m\nabla_{\theta_j} F_m(\theta_j(t))\right) \\ &= \frac{1}{m}\sum_{j=1}^m \nabla\varphi(\theta_j(t))^\top\left(-\nabla F'_{\mu_{m,t}}(\theta_j(t))\right) \\ &= \int \nabla\varphi^\top\left(-\nabla F'_{\mu_{m,t}}\right)\mathrm{d}\mu_{m,t}\end{aligned}$$

which shows that $\mu_{m,t}$ satisfies the continuity equation in the sense of distributions with the vector field $v_t = -\nabla F'_{\mu_{m,t}}$, *i.e.*, $(\mu_{m,t})_{t\geq 0}$ is the WGF of F starting from $\mu_{m,0}$.

Homogeneity and reduction to measures on the sphere

Positively homogeneous activations are very common in the literature around neural networks and especially for theoretical studies as they often lead to some simplifications. The ReLU (rectified linear unit) activation $\sigma(z) = \max(0, z)$ is one such example which is ubiquitous both in theory and in practice. However, it can also lead to technical difficulties due to its non-differentiability and the non-continuity of its derivative $\mathbb{1}_{z>0}$ at 0. In particular, the existence of the WGF (1.5) cannot be guaranteed in general when using ReLU as the activation function. Nevertheless, it is possible to circumvent that technical issue thanks to the positive homogeneity of ReLU and to specific assumptions on the initial distribution μ_0. It is shown in (Wojtowytsch, 2020) and (Chizat and Bach, 2020) that when the initial measure $\mu_0 \in \mathcal{P}_2(\mathbb{R}^{d+1})$ is supported on the cone $\{(w^1, w^2) \in \mathbb{R}^{d+1} : ||w^1|| = |w^2|\}$, the WGF (1.5) is well-defined with a ReLU activation. In addition, it is shown that with this initialization, the measure μ_t stays supported on the cone at any time t.

Reduction to signed measures on the sphere. The positive homogeneity property of ReLU also enables taking an alternative point of view for the WGF (1.5). Indeed, for any measure $\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})$, one can define a pair of non-negative measures supported on the sphere $\nu_+, \nu_- \in \mathcal{M}_+(\mathbb{S}^{d-1})$ via the following characterization, which is particularly suited to the homogeneity of two-layer networks with a ReLU activation: for any continuous test function $\varphi : \mathbb{S}^{d-1} \to \mathbb{R}$, it must hold that

$$\int \varphi \mathrm{d}\nu^{\pm} = \int_{\pm w^2 \geq 0, w^1} |w^2| \, ||w^1|| \varphi\left(\frac{w^1}{||w^1||}\right) \mathrm{d}\mu(w).$$

Essentially, this can be understood as a form of projection which factors out the redundancy—induced by the homogeneity property—between the norm of the first layer weights and the magnitude of the output layer weights . With this definition, the network function can be expressed as

$$f(\mu; x) = \int w^1 \sigma(x^\top w^2) \mathrm{d}\mu(w) = \int \sigma(x^\top u) \mathrm{d}\nu(u),$$

with $\nu = \nu^+ - \nu^- \in \mathcal{M}(\mathbb{S}^{d-1})$ a signed measure on the sphere. From this perspective, neurons of the first layer are seen as directions on the sphere, while the weights of the second layer are seen as (signed) mass weighing those directions. The mass in this parameterization takes into account both the second layer weights and the norm of the first layer weights in the original parameterization of Equation (1.3). In this point of view, the problem of learning an infinitely-wide two-layer network is viewed as learning the positions and masses of neurons of the first layer. The total mass of ν measured by the total variation norm is given by $|\nu|(\mathbb{S}^{d-1}) = \int \mathrm{d}(\nu^+ + \nu^-) = \int ||w^1|| \, |w^2| \mathrm{d}\mu(w)$.

Consider the WGF (1.5) with a ReLU activation, with μ_0 supported on the cone. Then, defining $\nu_t^{\pm}$ from μ_t as above, the fact that μ_t is supported on the cone at any time allows to derive evolution equations for the measures $\nu_t^{\pm}$. We stress that *this is not possible* in general (even if we assume homogeneity only). The pair (ν_t^+, ν_t^-) satisfies the following equations, known as advection-reaction equations (or Wasserstein-Fisher-Rao gradient flow Gallouët et al., 2019), in the sense of distributions:

$$\partial_t \nu_t^{\pm} = -\mathrm{div}(\pm \tilde{v}_t \nu_t^{\pm}) \pm 2 g_t \nu_t^{\pm}, \tag{1.9}$$

with $g_t(u) = F'_{\mu_t}(u, 1)$ and $\tilde{v}_t(u) = -\mathrm{proj}_{\{u\}^\perp}(\nabla g_t(u))$ for $u \in \mathbb{S}^{d-1}$. That is, for any test function $\varphi \in \mathcal{C}_c^1(\mathbb{R}^d)$, it holds

$$\frac{d}{dt} \int \varphi \mathrm{d}\nu_t^{\pm} = \pm \int \tilde{v}_t^\top \nabla \varphi \mathrm{d}\nu_t^{\pm} \pm 2 \int \varphi g_t \mathrm{d}\nu_t^{\pm}.$$

The derivation of this equation follows from the continuity equation satisfied by μ_t, the definition of $\nu_t^{\pm}$ from μ_t using homogeneity, and the the fact that $|w^2| = ||w^1||$ on the support of μ_t. See Appendix B.5.1 for more details on the derivation of the Wasserstein-Fisher-Rao equation. Put differently, the signed measure $\nu_t = \nu_t^+ - \nu_t^-$ satisfies the equation

$$\partial_t \nu_t = -\mathrm{div}(\tilde{v}_t(\nu_t^+ + \nu_t^-)) + 2 g_t (\nu_t^+ + \nu_t^-)$$

where ν_t^+ and ν_t^- represent the positive and negative part of ν_t respectively. This point of view is used extensively in Chapter 3. From this standpoint, mass is not preserved and the change in mass is governed by the reaction term g_t while the advection (displacement) of the mass is governed by the vector field $\tilde{v}_t$ which is tangential to the sphere. In particular, the total mass $|\nu_t|(\mathbb{S}^{d-1})$ evolves according to $\frac{d}{dt}|\nu_t|(\mathbb{S}^{d-1}) = \int g_t \mathrm{d}\nu_t$.

1.2.5 . Tensor programs and infinite-width limits of any parameterization

The Tensor Program is a framework developed in a series of works (Yang, 2019, 2020a,b; Yang and Hu, 2021; Yang et al., 2022) in order to better understand and describe rigorously the infinite-width limit of various parameterizations (as introduced in Section 1.2.1) of neural networks. The goal is to understand precisely the magnitude of the quantities involved in the forward and backward passes of a neural network as $m \to \infty$. In doing so, one particular obstacle is to understand how the different quantities correlate to each other.

The ideas and techniques developed in the Tensor Program series originate in the statistical physics literature (Bayati and Montanari, 2011; Bolthausen, 2014) where they have emerged in order to describe the behavior of algorithms (such as message passing) involving large random matrices and non-linearities using the Gaussian conditioning technique. The added benefit of the Tensor Program is to

provide a formalism to systemically apply those techniques in the context of neural networks. We use the Tensor Program extensively in the proofs of most of the results presented in Chapter 2.

The first work in the Tensor Program series (Yang, 2019) is devoted to understanding what kind of function is computed by deep neural networks *at initialization* with i.i.d. Gaussian matrices with a standard deviation scaling as $m^{-1/2}$. While the answer is known for shallow fully-connected networks since Neal (1995), several recent works (Lee et al., 2017; Matthews et al., 2018; Novak et al., 2018; Garriga-Alonso et al., 2018) have generalized that result to deeper networks or convolutional architecture. The first version of the Tensor Program in (Yang, 2019) provides mathematical tools to systematically prove that neural networks of *any architecture* behave as Gaussian processes at initialization in the infinite-width limit.

The second version of the Tensor Program (Yang, 2020a) extends the analysis to the first backward pass (the gradients at initialization) and proves that the neural tangent kernel $\nabla f(\theta; x)^\top \nabla f(\theta; y)$ (see Section 1.2.2) converges almost surely, as $m \to \infty$, to a deterministic limit at initialization for any architecture in the NTK parameterization.

The third version of the Tensor Program (Yang, 2020b) is focused on extending the mathematical tools previously developed to cover the forward and backward passes at any time step. One crucial step is the ability to describe the limit of quantities where both a weight matrix W^l and its transpose $(W^l)^\top$ are involved, and to handle the potential correlations that might result from this.

Finally, Yang and Hu (2021) use the framework of the Tensor Program to categorize different types of parameterization in the infinite-width limit. This categorization specifies whether an abc-parameterization (see Section 1.2.1 and Yang and Hu, 2021) is in the *kernel regime* or in the *feature learning regime* based on the values of the exponents a_l, b_l and c_l. Moreover, a new parameterization called μP is proposed, corresponding to the following values for the exponents: $a_1 = -1/2$, $a_l = 0$ for $l \in [2, L]$ and $a_{L+1} = 1/2$, $b_l = 1/2$, for $l \in [1, L+1]$, and $c_l = 0$ for $l \in [1, L+1]$. Equivalently, the exponents for that parameterization can also be given by: $a_1 = 0$, $a_l = 1/2$ for $l \in [2, L]$ and $a_{L+1} = 1$, $b_l = 0$ for $l \in [1, L+1]$, $c_l = -1$ for $l \in [1, L+1]$. It is the proper extension of "mean-field" models for more than two layers (they are identical for two-layer networks), and it "maximizes" learning in all layers (in a sense made precise in Yang and Hu, 2021). However, the analysis of Yang and Hu (2021) leaves out any parameterization for which the (pre-)activations might vanish at initialization as $m \to \infty$, which is the case of IPs with three layers or more. Hence the need for a special treatment which we present in Chapter 2.

Intuition behind the technique

We present briefly here the intuition behind the Tensor Program as well as its formalism and the main results associated with it. We start by describing the situation in the forward pass at initialization, where things are easier to understand, then move to describe the calculations involved in the first backward pass, and finally explain how to handle general computations in subsequent forward and backward passes.

First forward pass. The main point to study here is the behaviour of sums of the type $m^{-1/2}\sum_{j=1}^{m} w_j x_j$ for large m when $w \in \mathbb{R}^m$ is a Gaussian vector with i.i.d. entries following $\mathcal{N}(0,1)$ and $x \in \mathbb{R}^m$ is a random vector independent from w. When x has i.i.d. entries, the central limit theorem ensures that the latter quantity converges in law to a Gaussian variable as $m \to \infty$. In fact, this result also holds as soon as $||x||^2/m$ converges almost surely to some limit σ_∞^2 (see Yang, 2019[Proposition G.4]). The situation is more difficult when x and w are correlated and we discuss that case later on (it is handled in the third version of the Tensor Program Yang, 2020b). It easily follows that the entries of the pre-activations $h^l = m^{-1/2} w^l x^{l-1}$ of a network in the NTK parameterization become Gaussian as $m \to \infty$. The convergence of $||x^{l-1}||^2/m$ is due to the fact that entries tend to be roughly independent in the limit because different rows (which are i.i.d.) of the Gaussian matrix w^{l-1} are used to compute the different entries. It thus appears clear that independently of the activation function σ and of the topology of the network architecture, the output of the network tends to be Gaussian in the large-m limit as soon as the Gaussian initialization is scaled appropriately.

In the setting of (Yang and Hu, 2021), any other choice of scaling factors for the weights (the a in abc-parameterizations) will lead to the forward pass either vanishing or exploding.

First backward pass. When studying the backward pass at initialization, the key quantity is the gradients of the output of the network *w.r.t.* the activations x^l, that is $\nabla_{x^l} f(\theta; x) = (m^{-1/2} w^{l+1})^\top \nabla_{h^{l+1}} f(\theta; x)$. Essentially, the situation is the same as in the first forward pass: since $\nabla_{h^{l+1}} f(\theta; x)$ is computed using matrices w^k for $k \geq l+2$, it is independent from w^{l+1}. The difference is simply that we use the transpose of the initial matrices in the multiplications, but since those have entries initialized i.i.d. the same logic as in the forward pass applies. Since $\nabla_{x^L} f(\theta; x) = m^{-1/2} w^{L+1}$, its entries are of order $m^{-1/2}$ and this factor propagates to the gradients of all layers by the equations of backpropagation. This results in $\nabla_{w^l} f(\theta; x)$ being of order m^{-1}, which is significantly smaller than the initial magnitude, leading to a linearization, as discussed in Section 1.2.2, if the learning rates are not scaled appropriately. The magnitude of the gradients in the first backward pass is thus well understood.

General computation in subsequent steps. Correcting the gradient scales by using a learning rate of $m^{1/2}$ for the intermediate layers $l \in [2, L]$, solves the issue described above for the gradients at initialization. With that correction, the entries of the weight updates $\Delta W^l = W^l(1) - W^l(0)$ are of order m^{-1}, while that of $W^l(0)$ are of order $m^{-1/2}$. This is essentially what μP does, except it also corrects the scale of the output layer so that the weights are of order m^{-1} in order to prevent the output of the network from diverging after initialization.

This begs the question of how that is any different from the NTK behaviour since the magnitude of the updates are still much smaller than that of the initialization. The answer is subtle, and one has study the following forward pass at time step $t = 1$ to understand why this is the correct magnitudes. In short, the reason is that although $W^l(0)$ and ΔW^l have different magnitudes *w.r.t.* m, $W^l(0)x_1^{l-1}$ and $\Delta W^l x_1^{l-1}$ are both of the same order (namely of order 1) *w.r.t.* m because of the non-linearities and correlations involved in the second term. Thus, there is no linearization effect here.

Indeed, with the scale correction induced by the learning rates described above, the contribution of the weight update to the pre-activations h_1^l at layer l and time $t = 1$ reads as $\frac{(x_1^{l-1})^\top x_0^{l-1}}{m} \nabla_{h^l} f(\theta(0); x_0)$. Although the scale is not in $m^{-1/2}$ here as it is at initialization, it is clear that the computations involved are of a different nature: the two vectors multiplied in the inner product have no reason to have Gaussian coordinates, and furthermore they are *not independent* since the Gaussian matrix w^{l-1} is used to compute both terms. It is one of the most important result of the Tensor Program series, summarized in a Master theorem (see Yang, 2020b[Theorem 2.10], Yang and Hu, 2021[Theorem 7.4]), to prove that inner products of that type, rescaled by m^{-1}, converge to an almost sure limit as $m \to \infty$, justifying the scale in m^{-1} compared to the scale in $m^{-1/2}$ at initialization.

The intuitive idea for the convergence of the inner-products rescaled by m^{-1} is that the coordinates of (pre-)activations remain roughly i.i.d. throughout training for large m. Understanding how inner products and multiplications with i.i.d. Gaussian matrices scale with m and taking into account the potential correlation between different quantities is precisely what enables to understand how one should scale the initialization and the learning rates in the infinite-width limit to get weight updates which contribute maximally without leading to an explosion.

Tracking the scales and correlations as training progresses quickly becomes cumbersome for time steps $t \geq 1$, and the Tensor Program offers a way to make the computations systematic in the infinite-width limit. There are three types of objects in the tensor program framework: (i) i.i.d. Gaussian matrices $\hat{W}$ of size $m \times m$ with standard deviation $m^{-1/2}$, (ii) vectors $z \in \mathbb{R}^m$ with roughly i.i.d. coordinates, and (iii) scalars $\omega \in \mathbb{R}$. While the Gaussian matrices essentially represent the matrices of a neural network at initialization (or rescaled versions thereof), the vectors can be obtained in two ways: either as a matrix vector computation $z = \hat{W}x$ with

some other vector x, or through a non-linearity $z = \psi(z^1, \dots, z^p; \omega_1, \dots, \omega_q)$, where $\psi : \mathbb{R}^{p+q} \to \mathbb{R}$ is a parametric function applied *element-wise*, that is for any $j \in [1, m]$, $z_j = \psi(z_j^1, \dots, z_j^p; \omega_1, \dots, \omega_q)$. Those vectors represent the (pre-)activations or the gradients *w.r.t.* to the (pre-)activations, and scalars are obtained through rescaled inner products $x^\top y/m$ for some vectors x and y.

The rules of the Tensor Program, detailed in (Yang, 2020b; Yang and Hu, 2021), describe a system of computations that allow to derive the infinite-width limit of any series of computations (called a Tensor Program) using the three operations presented above. These rules state that any scalar $\omega = x^\top y/m$ converges almost surely to some finite value as $m \to \infty$ (that is the result from the Master theorem). In addition, in the infinite-width limit, the coordinates of a vector z all have the same distribution described by the law of a single random variable $Z \in \mathbb{R}$. In fact, the entries of the vector z converge in law to the random variable Z. If $z = \hat{W}x$, the law of Z is given by $Z = \hat{Z} + \dot{Z}$ where $\hat{Z}$ is Gaussian, centered and independent of x, and $\dot{Z}$ is a random variable accounting for the potential correlation between x and $\hat{W}$, with the most crucial example being $x = \hat{W}^\top y$. If $z = \psi(z^1, \dots, z^p; \omega_1, \dots, \omega_q)$, as $m \to \infty$, the law of Z is given by $Z = \psi(Z^{z^1}, \dots, Z^{z^p}; \bar{\omega}_1, \dots, \bar{\omega}_q)$ where Z^{z^r} is the limiting law of the entries of the vector z^r and $\bar{\omega}_s$ is the almost sure limit of ω_s. Finally, the almost sure limit of $\omega = x^\top y/m$ is equal to $\bar{\omega} = \mathbb{E}[Z^x Z^y]$. The precise description of the rules of a Tensor Program are provided in (Yang, 2020b; Yang and Hu, 2021), along with the proofs that any finite-width system of computations using the three operations described above (such as a neural network with practically any architecture) can be described in the infinite-width limit $m \to \infty$ by a corresponding system of computations on the limiting random variables Z.

Limitations of the tensor program. We discuss briefly here some of the limitations of the Tensor Program framework, which we elaborate further on in Chapter 2.

One obvious limitation is that the definition of the limiting Z variables are recursive, with formulas that quickly become intractable for deep neural networks trained for more than one step of (S)GD, so that although the description of the limit is clear, using the Tensor Program to study the properties of the training dynamics beyond the first couple of steps of training can be impractical, except in some specific cases (such as integrable parameterizations, and the reason for that is discussed in Chapter 2). Another drawback of the Tensor Program is that only non-linearities ψ with a certain regularity are allowed for the results to hold, which prevents from studying neural networks with a ReLU activation directly, although the framework could possibly be extended to handle non-smooth activations, but at the cost of tedious technical proofs.

In addition, the Tensor Program in its initial form (Yang and Hu, 2021) only allows Gaussian initialization for the weight matrices. However the universality of

the Tensor Program computations and its master theorem has recently been proved in (Golikov and Yang, 2022), allowing for general i.i.d. initializations.

1.3 . Contributions

In this section, we highlight the goals pursued in this thesis as well as the main contributions. IPs, at least with two layers, shift away from the kernel behaviour observed in the NTK parameterization and actually produce dynamics where features evolve with time, a fact that seems crucial for the empirical success of neural networks. The purpose of this thesis is to study the dynamics of infinitely-wide neural networks in the integrable parameterization, sometimes deep, and sometimes shallow. We seek to study different scenarios where we can uncover interesting properties of the training dynamics of integrable parameterizations in the infinite-width limit.

The first part of this thesis (Chapter 2) is devoted to better understanding the degeneracies which arise for deep networks in the integrable parameterization, and how one can train them in the infinite-width limit in a setting as close as possible to what is done in practice. The second part (Chapter 3) focuses on how the dynamics of infinitely-wide two-layer networks adapt to the symmetries and structure of a given task, and in particular studies the problem of learning low-dimensional sub-spaces. Finally, the third part (Chapter 4) studies different optimization algorithms over the space of measures which provide either theoretical results of global convergence with an explicit rate, or practically relevant methods where neurons can dynamically be added or removed during training.

1.3.1 . Infinite-width dynamics of integrable parameterizations

Integrable parameterizations with two-layers seem to have favorable properties compared to the NTK parameterization, but it appears that they have a degenerate behaviour with more than four layers in the standard setting where the weights of a given layer are initialized i.i.d. Our goal in Chapter 2 is to connect different lines of work around infinitely wide neural networks such as "mean-fiedl" limits and the Tensor Program. In particular, we wish to better understand the nature of this degeneracy, propose a solution to the issue while staying in a setting as close as possible to practically relevant methods, and study the properties of the resulting model in the infinite-width limit.

Degeneracy of integrable parameterizations

Araújo et al. (2019); Nguyen and Pham (2020) study the idealized gradient flow dynamics of IPs with i.i.d. initializations and more than four layers, and observe that in the infinite-width limit, the weights in a given intermediate layer all translate by the same deterministic quantity depending only on time. We go a step further and prove that this quantity is zero even with SGD, so that the weights do not

move at all as $m \to \infty$, causing the prediction function to be the same as at initialization at any time step. The following result appears in Proposition 2.3.1 of Chapter 2: for any $t \geq 0$ and any x,

$$\lim_{m\to\infty} f(\theta(t); x) = \lim_{m\to\infty} f(\theta(0); x) = 0,$$

where the convergence is almost sure.

Dynamics with large initial Large learning rates

The natural question that ensues is whether there is a fix to that issue in the case of i.i.d. initializations. We answer positively to that question. By studying precisely the magnitude of the gradients at initialization for deep IPs thanks to the Tensor Program and to the positive homogeneity assumption we consider on the activation σ, we notice that large learning rates allow the prediction function to evolve *non-trivially* after the first training step. However, it is important to note that the issue is much more subtle than "speeding up" the dynamics (using larger learning rates as m grows larger) to enable learning in the infinite-width limit (as is done for two-layers where (S)GD for IPs needs learning rates of order m, see Section 1.2.3). The subtlety lies in the fact that for deep IPs, the learning rates at initialization and at subsequent time steps cannot have the same value to enable stable training in the limit $m \to \infty$: if the learning rates' growth with m is too fast, the (pre-)activations will diverge after the first step, and if it is too small, they stay bounded but the weights do not move.

The correct point of view is that random fluctuations need to be amplified at initialization via **large initial learning rates** (LLR) before reverting to the "standard" learning rates found in the literature on IPs. We show that the correct magnitudes for the learning rates are, at $t = 0$ (initialization), $\eta_1 = \eta_{L+1} = m^{(L+1)/2}$ and $\eta_l = m^{(L+2)/2}$ for $l \in [2, L]$, and for $t \geq 1$, $\eta_1 = \eta_{L+1} = m$ and $\eta_l = m^2$ for $l \in [2, L]$. Under mild assumptions on the initial loss value and on the input data, we prove in Theorem 2.4.1 that when using those learning rates, the following holds:

$$\begin{aligned}
&f(\theta(0); x) \xrightarrow[m\to\infty]{a.s.} 0, \\
&f(\theta(1); x) \xrightarrow[m\to\infty]{a.s.} f_1^\infty, \quad 0 < |f_1^\infty| < \infty \ a.s., \\
&f(\theta(2); x) \xrightarrow[m\to\infty]{a.s.} f_2^\infty, \quad |f_2^\infty| < \infty \ a.s.
\end{aligned}$$

The homogeneity assumption is crucial here, although the magnitudes of the first forward and backward passes can also be well understood when $\sigma'(0) \neq 0$, but the full study would require a separate analysis.

Connection with μP

We now wish to understand the properties of a network trained with the learning rate schedule proposed above, which we call IP-LLR. We establish a connection between IP-LLR and the recently proposed μP (Yang and Hu, 2021): we show that, in the infinite-width limit, IP-LLR is in fact a modified version of μP where the weight matrices at $t = 0$ are initialized with the first weight updates of μP instead of the usual random Gaussian initialization. That is, we "forget" the random initialization of μP after the first gradient step.

Numerical results and other alternatives

We also explore in Chapter 2 other alternatives which enable training for deep i.i.d. IPs and show (theoretically as well as empirically) that the two other natural options we consider lead to degenerate behaviours. We complement our theoretical results with thorough numerical experiments to corroborate our findings and demonstrate that our mathematical statements seem to hold with much more general assumptions (non-homogeneous or non-smooth activation functions). Future directions include extending our theoretical results to non-homogeneous or non-smooth functions as well as analyzing more precisely the qualitative differences in the training dynamics of IP-LLR and μP.

1.3.2 . Symmetries in the dynamics of infinitely wide two-layer networks

In the theoretical quest to better understand how neural networks learn representations of the input data to solve the task they are presented with, it is natural to consider the problem of how networks adapt to the symmetries of the function they are trying to learn. Symmetries can be of various nature but we focus in Chapter 3 on **orthogonal symmetries**, and in particular on the setting where the target function f^* depends only on the orthogonal projection to a low-dimensional sub-space of $\mathbb{R}^d$. We study the symmetries induced by that of f^* on the gradient flow dynamics of infinitely wide two-layer ReLU networks. In this context, infinitely wide networks have the benefit that they allow the emergence of symmetries in the training dynamics which are only approximate at finite-width.

As mentioned in Section 1.2.3, the setting where f^* depends only on the projection to a lower-dimensional sub-space has already been studied from the statistical point of view in (Bach, 2017; Chizat and Bach, 2020), focusing on the favorable properties in terms of generalization of infinitely wide two-layer networks with positively homogeneous activations. Yet, the question of whether or not (S)GD is actually able to learn this sub-space is not addressed. Similarly, Cloninger and Klock (2021) and Damian et al. (2022) study how a single step of SGD on the input layer weights is already able to induce favorable statistical properties with bounds depending only on the dimension of the sub-space and not that of the

ambient space. Closer to our approach, Mousavi-Hosseini et al. (2022) show that doing (S)GD on the first layer only aligns the weights with the low-dimensional sub-space when sufficient L^2 regularization is used. Abbe et al. (2022) are able to prove that the gradient flow dynamics are able to learn the low-dimensional structure in a setting similar to ours due to their strong assumption that the data are Rademacher variables (*i.e.*, their entries belong to $\{-1, 1\}$).

In studying symmetries, another objective is to assess whether quantitative convergence results can be obtained with the added symmetry assumptions. While many global convergence results exist in the literature for two-layer networks (Chizat and Bach, 2018; Nguyen and Pham, 2020; Sirignano and Spiliopoulos, 2020; Wojtowytsch, 2020), no convergence rate is available in general. We demonstrate that for particular instances, exponential convergence can be proved.

In Chapter 3, we study the training dynamics of infinitely wide two-layer networks where both layers are trained and we focus on the WGF dynamics rather than statistical properties. We work under the added assumption that the input data distribution has spherical symmetry, and we optimize the population risk objective to allow the emergence of exact symmetries.

General results for orthogonal symmetries

We first show that in our setting, if f^* is invariant by some orthogonal transformation, then the measure μ_t and the predictor $f(\mu_t; \cdot)$ inherit this invariance (see more details in Proposition 3.2.1). We then apply this result to specific instances in which f^* is invariant by some sub-group of orthogonal transformations.

Exponential convergence for odd target functions

A consequence of the result discussed above is that if f^* is an odd function, then $f(\mu_t; \cdot)$ is also odd. It then follows from the identity $\sigma(z) - \sigma(-z) = z$ satisfied by ReLU that the predictor is actually linear: $f(\mu_t; x) = w(t)^\top x$ with $w(t) = \frac{1}{2} \int w^1 \, w^2 \mathrm{d}\mu_t(w^1, w^2)$. This linearization is different from the behaviour of NTK: both layer weights evolve non-trivially but the symmetries of the problem imply a degeneracy to linear predictors. In fact, this degeneracy is not surprising as the risk minimizer must be linear in this context. We show in Theorem 3.3.2 that the WGF dynamics converge **exponentially fast** towards this global minimizer of the training objective: given the global minimum F^*, we show that there exists a positive constant $c > 0$ and a time $t_0 \geq 0$, such that for any $t \geq t_0$, it holds

$$F(\mu_t) - F^* \leq e^{-c(t-t_0)} \left(F(\mu_{t_0}) - F^*\right).$$

Note that in this setting, although the predictor is linear, the WGF dynamics are still non-linear as the optimization path is different from optimizing the linear parameterization $f(w; x) = w^\top x$: defining $w(t) = \frac{1}{2} \int w^1 \, w^2 \mathrm{d}\mu_t(w^1, w^2)$ or $w(t)$ as the gradient flow of the objective $\tilde{F} : w \in \mathbb{R}^d \mapsto \frac{1}{2}\mathbb{E}_{x\sim\rho}\left[(w^\top x - f^*(x))^2\right]$ *does*

not lead to the same optimization path, even though in this case, both converge to the best linear predictor.

The assumption on f^* is obviously restrictive, but it shows that in this particular setting, it is possible to obtain a convergence rate for the WGF, although no rates are known in general. Other settings have also been studied in the literature in order to provide convergence rates: E et al. (2020) are able to prove *local convergence* in $O(1/t)$ for one-dimensional inputs, and Daneshmand and Bach (2022) also prove global convergence at the rate $O(1/t)$ for inputs in two dimension and target functions with a finite number of atoms and a well-designed activation function.

Lower dimensional gradient flow dynamics

Finally, we turn our attention to the case where $f^*(x) = f_H(x^H)$ where H is a low-dimensional sub-space of $\mathbb{R}^d$, $f_H : H \to \mathbb{R}$, and x^H is the orthogonal projection of x onto H. Such an f^* is invariant by all the orthogonal transformations which preserve H, and it easily follows from the results on orthogonal symmetries discussed above that this is also the case for the predictor $f(\mu_t; \cdot)$. In particular, this implies that there is only a dependency on the orthogonal of H through the norm: $f(\mu_t; x) = \tilde{f}_t(x^H, ||x^\perp||)$ where $x^\perp = x - x^H$ is the orthogonal projection onto the orthogonal of H.

The challenge now is to show that as $t \to \infty$, the dependence on $||x^\perp||$ fades out, leaving only the dependence on the orthogonal projection onto H. This would mean that the features learned by the network have *adapted* to the low-dimensional structure of the problem. This is a difficult problem to solve theoretically. Yet, numerically, one can indeed observe that the dependence on $||x^\perp||$ vanishes with time. Although it is difficult to prove that the measure μ_t tends to be supported on the sub-space H for large t, it is possible to show that the training dynamics themselves can be reduced to a lower-dimensional gradient flow. In the setting where f^* depends only on the orthogonal projection onto H, we show that the WGF can be reduced to a Wasserstein-Fisher-Rao gradient flow over a smaller number of dimensions. Amongst those dimensions, one component corresponds to the direction of input neurons in the unit sphere of H and the other to the angle between input neurons and H.

We go even further when the function f_H is positively 1-homogeneous and prove in Theorem 3.4.3 that the WGF can be reduced to a Wasserstein-Fisher-Rao gradient flow on a **single parameter** corresponding to the angle between the input neurons and H. We show that there exists a pair of non-negative measures $\tau_t^+, \tau_t^- \in \mathcal{M}_+([0, \pi/2])$, satisfying the advection-reaction equation

$$\partial_t \tau_t^\pm = -\text{div}\left(\pm V_t \tau_t^\pm\right) \pm 2 G_t \tau_t^\pm$$

where the reaction term G_t is the first variation of some objective functional over $\mathcal{M}([0, \pi/2])$ incorporating the invariances of the problem, and the advection term

$V_t = G_t'$ is the derivative of the reaction term. This one dimensional reduction allows to easily simulate the PDEs, and we demonstrate numerically that the WGF dynamics lead to a measure which appears to be supported on H as $t \to \infty$, confirming that in this setting, infinitely wide networks are able to learn the low-dimensional sub-space which matters for prediction.

Discussion

Although we show rigorously that the WGF dynamics reduce to lower-dimensional dynamics, it is still an open question whether it can be proved that as $t \to \infty$ the measure μ_t converges to some measure μ_∞ that is supported on H. Other works (such as Mousavi-Hosseini et al., 2022; Abbe et al., 2022) have studied this convergence with modified dynamics, but in general, the problem of proving the convergence to the sub-space H in the long run has not yet been solved. One other future direction would be to try to extend the proof technique of Mousavi-Hosseini et al. (2022) to the setting of infinitely wide two-layer networks where both layers are trained.

1.3.3 . Optimization over the space of measures: dynamically adding and pruning neurons

In Chapter 4, we consider generic convex objectives $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$ over the space of measures on the sphere and propose algorithms for their minimization. This setting covers (but is not restricted to) the optimization of two-layer networks with an unconstrained number of neurons. Our goal is two-fold: (i) provide an algorithm with global convergence guarantees at an explicit rate when the objective is smooth, (ii) propose methods which behave well in practice both in terms of performance and computation-wise.

While Wasserstein GF provably converges to a global minimum for infinitely wide two-layer networks (Nguyen and Pham, 2020; Chizat and Bach, 2018; Wojtowytsch, 2020), no convergence rate is known in general. We propose an algorithm to minimize smooth and convex F with a rate of $k^{-\frac{1}{d}}$ where k is the iteration number. This algorithm is inspired by coordinate descent methods for optimization in finite dimension (see, *e.g.*, Wright, 2015) and involves sampling a new neuron at each step which makes it prohibitively expensive to use in practice.

To mitigate that issue, we consider *penalizing* the smooth objective to encourage sparsity and limit the number of neurons and thus the computation cost incurred by the algorithm. We thus consider objectives of the form $F = J + \lambda H$ where J is a smooth term (such as the empirical loss for two-layer networks parameterized by measures) and H is a sparsity-inducing penalty. We study two different types of penalties: first a total variation penalty which is the analog of an L^1-penalty in finite dimension, leading to proximal algorithms in the space of measures to deal with the non-smoothness of the total variation. Secondly, we consider kernel penalties with either attractive or repulsive kernels. While the latter do

not explicitly remove neurons, the corresponding dynamics induce some neurons to grow closer to each other and they can eventually be merged in an ad-hoc fashion beyond some threshold.

We stress that the work presented in Chapter 4 is still under progress at the time of writing this thesis and some parts may thus feel incomplete.

Global convergence of coordinate descent in the space of measures

In finite dimension, many different techniques exist for convex optimization depending on the context: is the objective smooth or not, do Łojasiewicz-type conditions hold, do we use the full-gradient or a single coordinate at each step ? We review such techniques in Section 4.3 as a lot of the ideas are useful in our setting. Taking inspiration from these methods, we consider a smooth and convex objective $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$ and propose a coordinate descent algorithm in the space of signed measures which allows to minimize it at a rate of $k^{-\frac{1}{d}}$.

In this setting, a coordinate is viewed as a neuron $u \in \mathbb{S}^{d-1}$ and the objective of coordinate descent is to minimize, given a measure $\mu \in \mathcal{M}(\mathbb{S}^{d-1})$, an upper bound on $F(\mu+t\delta_u)$ over $t \in \mathbb{R}$ where δ_u is the Dirac measure at μ. Starting from a single atom $\mu_0 = c_0\delta_{u_0}$, this gives rise to an algorithm where at every iteration k the current iterate has the form $\mu_k = \sum_{i=0}^{k} c_i\delta_{u_i}$ and we sample a new neuron $u_{k+1} \in \mathbb{S}^{d-1}$ uniformly over the sphere and set its weight c_{k+1} by minimizing over $t \in \mathbb{R}$ an upper bound on $F(\mu_k + t\delta_{u_{k+1}})$. We prove in Lemma 4.4.2 that a Łojasiewicz-type inequality holds for the iterates and then deduce with similar arguments as in finite dimension the convergence of this algorithm to a global minimizer in expectation with an explicit rate. Precisely, we show in Theorem 4.4.3 that there is a constant $C > 0$ such that for any $k \geq 1$ it holds:

$$0 \leq \mathbb{E}[F(\mu_k) - F^*] \leq \frac{C}{k^{1/d}},$$

where F^* is the minimum of the objective F. This coordinate descent in $\mathcal{M}(\mathbb{S}^{d-1})$ is to be understood in the L^2 geometry as each step is equivalent, in expectation, to the minimization, over $\nu \in L^2(\omega_d)$ (where ω_d is the uniform distribution on $\mathbb{S}^{d-1}$), of an upper bound on $F(\mu_k + \nu)$ involving the squared norm $||\nu||^2_{L^2(\omega_d)}$. In practice, such an algorithm can be mixed with descent steps in the Wasserstein geometry which often have good empirical behaviour although they do not provide convergence rates in this setting.

The inconvenient of the coordinate descent algorithm presented above is that the number of neurons grows linearly with the iteration number k which makes it prohibitively expensive to use in practice. Thus, we discuss below the addition of penalties to the smooth objective with encourage sparsity and offer a balance between global convergence and computational cost.

Proximal algorithms for total variation penalties

We now consider a composite objective $F(\mu) = J(\mu) + \lambda|\mu|_{TV}$ where J is smooth and $|\mu|_{TV}$ is the total variation norm of $\mu \in \mathcal{M}(\mathbb{S}^{d-1})$. This is akin to an L^1 penalty in finite dimension which is known to induce sparsity. The total variation penalty is not smooth and taking again inspiration from convex methods in finite dimension, we propose a proximal coordinate descent algorithm for the minimization of the penalized objective. While in finite dimension convergence rates can still be obtained for proximal methods, in our setting global convergence is lost and we have no explicit control over the number of neurons. Indeed, the proximal coordinate descent step is given by a *soft-thresholding* operator: at each iteration the number of neurons can either increase by one or stay constant but it stays constant only if there is no change in objective value from one iteration to the next. In this context, sparsity and global convergence are incompatible: decrease in the objective can only be obtained by adding a new neuron.

To alleviate this issue, we consider a modification to the proximal algorithm where we alternate between sampling a new neuron on the sphere and sampling from the existing neurons of the current iterate μ_k. When sampling from existing neurons, the proximal step is also given by a soft-thresholding operator but this time the number of neurons can either stay fixed or decrease by one from one iteration to the next, and we can have both a decrease of the number of neurons *and* a decrease in the objective. Unfortunately, we still have no theoretical guarantees of convergence and nor do we have a control over the number of neurons but it appears that this method behaves well in practice and manages to both decrease the objective as well limit the growth of the number of neurons.

Smooth kernel penalties

Another approach we take is to study smooth kernel penalties which either attract or repulse neighboring neurons. In this setting we also consider a composite objective $F(\mu) = J(\mu) + \lambda H(\mu)$ where $H(\mu) = \int K(u,v)\mathrm{d}|\mu|(u)\mathrm{d}|\mu|(v)$ is a kernel penalty, and $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ is a symmetric, smooth, and non-negative kernel. The kernels we consider are dot-product kernels $K(u,v) = \kappa(\langle u, v\rangle)$ with $\kappa : \mathbb{R} \to \mathbb{R}_+$.

We say that the kernel is *attractive* if κ is a decreasing function, and that it is *repulsive* if κ is a increasing function. Typically, we consider $\kappa_{a,\sigma}(s) = 1 - e^{(s-1)/\sigma^2}$ for attractive kernels and $\kappa_{r,\sigma}(s) = e^{(s-1)/\sigma^2}$ for repulsive kernels. $\sigma > 0$ is a parameter controlling the range of interaction of different atoms on the sphere. While such kernels do not explicitly remove neurons within the iterations of the algorithm, the Wasserstein-Fisher-Rao dynamics (see Sections 1.2.4 and 4.5 for more details) induce some particles to get closer (even in the repulsive case as repulsive forces from many different particles might push some of them towards each other) to the point that we can effectively merge them if their distance is

smaller than some threshold. We show an example of this on the case of two particles interacting in Section 4.5.1.

These methods therefore implicitly induce some control over the number of neurons. We alternate between coordinate descent steps which should decrease the objective but at the cost of adding neurons, and Wasserstein-Fisher-Rao steps which should enable the merging of neurons while still behaving well empirically in terms of decrease of the objective.

While such kernel penalties are theoretically motivated, there is unfortunately no guarantee of convergence or control of the growth of the number of particles.

Discussion

The proximal algorithm we present for minimizing the non-smooth objective with the total variation penalty has good empirical behaviour but it is still an open question whether a proof of convergence can be obtained in this setting. Designing algorithms which provide both a theoretical guarantee of convergence and at the same time are computationally feasible (at least empirically) is difficult, and we leave the exploration of alternative approaches than the ones we present for future work.

Introduction (Français)

"Je pense, donc je suis", écrivait le philosophe et mathématicien français du XVIIe siècle Descartes, suggérant que la connaissance de sa propre conscience est un élément clé des êtres intelligents. Les machines seront-elles un jour capables de produire un raisonnement similaire ? La quête pour développer des machines capables de penser, de raisonner, de calculer et de résoudre des problèmes a occupé des scientifiques de différentes époques, remontant au moins aux efforts de Pascal et Leibniz pour produire une machine arithmétique capable d'effectuer diverses opérations algébriques.

Depuis lors, la technologie a évolué au point où un logiciel installé sur un ordinateur peut aider à résoudre des problèmes mathématiques, à traduire du texte dans différentes langues ou à jouer au jeu d'échecs à un niveau surhumain. Un regain d'intérêt pour le sujet de l'*intelligence artificielle* (IA) a eu lieu après la Seconde Guerre mondiale avec les travaux de Turing sur les machines de calcul et l'intelligence (Turing, 1950). De nombreuses initiatives (telles que le Logic Theorist, le Dartmouth Research Project ou la Cybernétique) ont commencé à apparaître dans le but de développer des systèmes experts capables de reproduire les compétences de résolution de problèmes et de raisonnement des humains. À l'époque, la recherche était principalement théorique, se concentrant sur les idées pour construire une IA et sur la manière de tester l'intelligence des machines.

Sur la base de l'observation selon laquelle l'activité cérébrale se réduit simplement à des impulsions électriques qui pourraient être reproduites dans un ordinateur, la recherche autour des réseaux neuronaux artificiels et des modèles mathématiques de neurones a rapidement émergé. Un exemple est le perceptron de Rosenblatt (Rosenblatt, 1958), où un ensemble de potentiomètres mettant en œuvre des poids adaptatifs est capable de reconnaître des lettres fournies en entrée au système par le biais d'un ensemble de 400 cellules photoélectriques. Cependant, l'intérêt pour de tels modèles s'est rapidement estompé, car un certain nombre de réserves ont commencé à émerger autour des réseaux neuronaux artificiels. Par exemple, le perceptron a été critiqué pour son incapacité à classer correctement les données qui ne sont pas linéairement séparables (même dans des contextes simples tels que le problème XOR), ce qui implique la nécessité de réseaux plus profonds accompagnés de nombreuses difficultés pratiques et théoriques. De plus, les avancées pratiques étaient limitées par les ressources de calcul et le coût élevé des ordinateurs.

Les réseaux neuronaux sont donc sortis de la mode jusqu'à la fin des années 1980, et la recherche en intelligence artificielle s'est principalement concentrée sur des systèmes experts effectuant un raisonnement symbolique, c'est-à-dire suivant un ensemble de règles artisanales pour résoudre une tâche spécifique. Néanmoins, certains groupes ont continué à étudier les réseaux neuronaux artificiels, et de

nouveaux résultats empiriques et théoriques ont ravivé l'intérêt pour de tels modèles. Rumelhart et al. (1985) ont dérivé les règles de la *rétropropagation* pour calculer de manière algorithmique les dérivées partielles de la fonction de coût par rapport aux poids d'un réseau en utilisant la règle de la chaîne. LeCun et al. (1989, 1998) ont montré que les réseaux neuronaux peuvent être appliqués avec succès à la reconnaissance de chiffres manuscrits, de codes postaux et de documents, et Barron (1993) et Pinkus (1999) ont démontré la propriété d'approximation universelle des réseaux neuronaux à deux couches.

Cependant, les progrès sont lents car les ressources de calcul sont toujours limitées, et beaucoup d'expérience pratique est nécessaire pour concevoir des réseaux neuronaux. La théorie et la pratique des systèmes d'apprentissage automatique de différents types sont développées à la fin des années 1990, et des compétitions sont même organisées pour désigner les meilleurs algorithmes sur des tâches telles que la reconnaissance d'images ou le traitement automatique du langage naturel (NLP). Ces systèmes concentrent souvent leurs efforts sur le problème crucial de l'extraction de caractéristiques : les transformations des données d'entrée sont conçues manuellement par des experts humains avant d'être fournies à une couche linéaire dont les paramètres sont appris de manière algorithmique. Un moment clé de l'histoire des réseaux neuronaux artificiels est lorsque le réseau neuronal AlexNet (Krizhevsky et al., 2012) a remporté la première place lors du concours ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2012 avec une erreur de seulement $15,3\%$ sur l'ensemble de test. Dans ce système, toutes les caractéristiques sont apprises automatiquement par le réseau, ainsi que la dernière couche.

Depuis lors, les réseaux neuronaux ont connu de nombreux succès en pratique, apprenant à jouer à des jeux Atari (Mnih et al., 2013), atteignant une erreur de moins de 5% sur l'ensemble de données ImageNet (He et al., 2016), comprenant et traduisant du texte ainsi que répondant à des questions sur un document (Vaswani et al., 2017), jouant au go, aux échecs et au shogi à un niveau surhumain (Silver et al., 2017), et générant du texte et des images sans supervision (Goodfellow et al., 2014; Devlin et al., 2018; Rombach et al., 2022).

Les progrès rapides ont été rendus possibles par une puissance de calcul en constante augmentation, permettant aux réseaux de devenir plus profonds et plus larges avec un grand nombre de paramètres (jusqu'à *des centaines de milliards* pour des systèmes comme ChatGPT), ainsi que par des recettes pratiques pour former les réseaux neuronaux modernes, telles que les connexions résiduelles (He et al., 2016), la batch normalization ou la layer normalization (Ioffe and Szegedy, 2015; Ba et al., 2016), des algorithmes basés sur le gradient adaptatif utilisant du momentum, ou des couches d'attention dans les architectures de type transformers (Vaswani et al., 2017). Malgré les nombreux succès des réseaux neuronaux modernes, les avancées théoriques sont en retard et notre compréhension des raisons de ces prouesses reste limitée.

Contexte général

Le but de cette thèse est d'approfondir notre compréhension théorique de la dynamique de l'algorithme d'entraînement des réseaux neuronaux dans la limite où le nombre de neurones dans une couche devient infini. L'asymptote de largeur infinie est récemment apparue comme un moyen de fournir des éclairages sur la dynamique d'entraînement des réseaux neuronaux d'un point de vue mathématique (voir la Section 1.2), et la recherche que nous présentons ici s'inscrit dans cette lignée.

De nombreuses variantes d'architectures de réseaux neuronaux (entièrement connectées, convolutionnelles, récurrentes, transformateurs, etc.) et d'algorithmes d'entraînement (momentum, ajustement du taux d'apprentissage, gradient batch ou complet) existent. Dans cette thèse, puisque nous cherchons à étudier ces objets de manière rigoureuse, nous nous concentrons sur ce qui est probablement la forme la plus simple pour réduire la complexité : les réseaux neuronaux entièrement connectés (parfois avec seulement deux couches, parfois plus) entraînés avec la descente de gradient classique (stochastique) ((S)GD). Bien que notre travail soit de nature théorique, cette thèse ne développe pas une nouvelle théorie mathématique des réseaux neuronaux artificiels, mais utilise les outils mathématiques disponibles pour éclairer le comportement des réseaux neuronaux utilisés en pratique en étudiant des versions idéalisées des dynamiques d'entraînement de réseaux de largeur finie du monde réel. Nous faisons donc un certain nombre de simplifications (en plus de l'étude de la limite où le réseau devient infiniment large) dont la nature dépend du contexte que nous considérons, allant de la connaissance de la distribution complète des données (objectif de risque population), à l'utilisation d'une taille de pas infinitésimale (dynamique du flot de gradient), en passant par l'étude de fonctions d'activation lisses ou positivement homogènes.

Le problème de minimisation du risque

En apprentissage automatique, différents types de paradigmes existent tels que l'apprentissage non supervisé, l'apprentissage par renforcement, les modèles graphiques probabilistes, et peut-être le plus omniprésent, celui de l'apprentissage supervisé. En apprentissage supervisé, on dispose d'une distribution de données $\mathcal{D}$ de paires $(x, y) \in \mathbb{R}^d \times \mathbb{R}^K$, ainsi qu'une fonction de perte $\ell : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$, et l'objectif est de minimiser

$$\mathbb{E}_{x,y\sim\mathcal{D}}\left[\ell(y, f(x))\right]$$

sur une classe $f \in \mathcal{F}$ de fonctions, appelées *prédicteurs*, de $\mathbb{R}^d$ à $\mathbb{R}^K$. Généralement, y, appelé la *cible*, est modélisé par $y = f^*(x)$ ou $y = f^*(x) + \epsilon$ où $f^* : \mathbb{R}^d \to \mathbb{R}^K$ est la *fonction cible* et ϵ est une variable aléatoire représentant un éventuel bruit dans les données. K est le nombre de catégories ou de classes, y peut être soit une variable continue, comme dans le *problème de régression*, soit un entier dans $\{1, \ldots, K\}$, comme dans le *problème de classification*.

Dans cette thèse, pour simplifier, nous nous concentrons sur le problème de régression avec une seule cible réelle ($K = 1$), bien que les extensions à la régression multidimensionnelle soient envisageables. Nous considérons également des cibles $y = f^*(x)$, $f^* : \mathbb{R}^d \to \mathbb{R}$ sans bruit, car nous nous intéressons principalement à la **trajectoire d'optimisation** plutôt qu'aux propriétés statistiques des modèles que nous examinons. Dans ce cadre, nous considérons donc une distribution ρ sur les données d'entrée et essayons de résoudre :

$$\min_{\theta \in \Theta} \Big\{ F(\theta) := \mathbb{E}_{x \sim \rho} \left[\ell(f^*(x), f(\theta; x)) \right] \Big\}$$

où F est appelé le *risque*, et la minimisation porte sur une classe de *fonctions paramétriques* $f(\theta; \cdot)$ avec un domaine de paramètres donné par un ensemble Θ. La distribution ρ peut être soit la *distribution empirique* $\hat{\rho}_n := \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$, conduisant au problème de *minimisation du risque empirique* (MRE) suivant

$$\min_{\theta \in \Theta} \Big\{ F(\theta) := \frac{1}{n} \sum_{i=1}^{n} \ell(f^*(x_i), f(\theta; x_i)) \Big\},$$

ou ρ peut être la distribution théorique complète des données (*e.g.*, gaussienne ou uniforme sur une certaine variété), conduisant au problème de *minimisation du risque de population*. La tâche consistant à trouver une bonne valeur θ^* pour le paramètre (une valeur qui minimise l'objectif F, ou qui s'en approche) est appelée *apprentissage*.

Lorsque l'on fait de l'MRE, l'objectif final n'est pas d'être capable d'apprendre une bonne valeur θ^* pour le risque empirique, mais pour le risque de population, souvent évalué à l'aide d'un ensemble de données, appelé ensemble de test (car la véritable distribution des données est inconnue), différent de celui utilisé pour apprendre θ^*, appelé ensemble d'entraînement. Une mesure de la pertinence de la valeur apprise θ^* est l'*erreur de généralisation*, c'est-à-dire la différence entre le risque empirique et le risque de population du paramètre optimal θ^* appris sur le risque empirique. Parfois, un terme de *pénalisation* est ajouté au terme d'ajustement des données afin d'induire de bonnes propriétés de généralisation, et l'objectif F devient $F(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(f^*(x_i), f(\theta; x_i)) + \lambda H(\theta)$ pour une certaine pénalité H qui est souvent convexe (*e.g.*, pénalité L^1 ou L^2). Lorsque l'on essaie d'apprendre θ^*, on se tourne vers le **paysage d'optimisation** de F, tandis que la valeur de l'erreur de généralisation est plus liée aux propriétés statistiques de θ^*. Cette thèse se concentre sur le premier aspect du problème lié à l'optimisation, et nous considérons principalement le risque sans pénalité.

Modèles linéaires

L'exemple le plus étudié de fonctions paramétriques est probablement celui des *modèles linéaires* $f(\theta; x) = \theta^\top \Phi(x)$ où $\theta \in \mathbb{R}^p$ est le paramètre et $\Phi : \mathbb{R}^d \to \mathbb{R}^p$ est un *extracteur de caractéristiques*. Ici, l'objectif F du problème de minimisation

du risque est convexe dès que la perte ℓ est convexe dans son deuxième argument, et donc des minimiseurs globaux existent dans la plupart des cas, ou du moins (S)GD est garanti d'approcher un minimum global sous des hypothèses légères. Pendant longtemps, c'était la manière standard de procéder : l'extracteur de caractéristiques Φ est conçu manuellement en fonction de la tâche à accomplir, et le paramètre θ du modèle linéaire est appris à l'aide de (S)GD.

Dans ce cadre, la valeur de θ^* optimal peut souvent être calculée explicitement, ou du moins avec des garanties théoriques conduisant à une analyse précise de l'erreur de généralisation et permettant de quantifier rigoureusement la justesse statistique de la méthode d'apprentissage.

Réseaux neuronaux

Les réseaux neuronaux sont des fonctions paramétriques définies par une succession d'opérations linéaires (ou affines) suivies d'une non-linéarité, définie par l'expression suivante : $f(\theta; x) = W^L\sigma(W^{L-1}\sigma(\dots\sigma(W^1x + b^1)) + b^{L-1}) + b^L$, qui peut également être écrite de manière récursive comme suit :

$$\begin{aligned} f(\theta; x) &= W^L x^{L-1} + b^L, \\ x^l &= \sigma(h^l), \quad h^l = W^l x^{l-1} + b^l, \quad l \in [1, L-1] \\ x^0 &= x, \end{aligned}$$

où $\sigma : \mathbb{R} \to \mathbb{R}$ est une fonction réelle appelée *fonction d'activation*, et est appliquée *élément par élément* aux vecteurs, et l'entier L est appelé la profondeur du réseau (c'est-à-dire le nombre total de couches). Pour $l \in [1, L]$, les matrices $W^l \in \mathbb{R}^{m_l \times m_{l-1}}$ sont appelées les *poids* et leurs rangées sont appelées les *neurones*, et les vecteurs $b^l \in \mathbb{R}^{m_l}$ sont appelés les *intercepteurs* ou les termes de biais. L'entier m_l est appelé la largeur de la l-ième couche (c'est-à-dire le nombre de neurones dans la couche l), avec $m_L = 1$ dans notre cadre, et $m_0 = d$ la dimension des entrées x. Les vecteurs $h^l \in \mathbb{R}^{m_l}$ sont appelés les *pré-activations* de la couche l, et x^l les *activations*, ou caractéristiques intermédiaires, tandis que les activations de l'avant-dernière couche x^{L-1} sont génériquement appelées les "features" "caractéristiques". Dans ce cadre, le paramètre θ est la concaténation de toutes les matrices de poids et des intercepteurs de toutes les couches.

Pour simplifier la présentation, dans cette thèse, nous omettons souvent les intercepts b^l, mais cela ne devrait pas avoir un impact énorme sur la généralité de notre travail (l'extension aux réseaux avec intercepts est assez simple, et la transformation : $x \mapsto w^\top x + b$ peut toujours être réécrite : $x \mapsto \tilde{w}^\top \tilde{x}$ où $\tilde{w} := (w, b)$ et $\tilde{x} := (x, 1)$). De plus, nous considérons toujours $L \geq 1$, de sorte qu'il y ait au moins 2 couches.

La Figure 1.1 illustre une architecture de réseau neuronal entièrement connecté typique avec un total de 6 couches.

Dans un réseau neuronal, la première couche, également appelée *couche d'entrée* ($l = 1$) et la dernière couche, également appelée *couche de sortie ou de prédiction*,

se comportent souvent un peu différemment des couches intermédiaires, et pour des raisons pratiques, nous considérons toujours des réseaux avec $L+1$ couches, où $l = 1$ correspond à la couche d'entrée, $l = L+1$ à la couche de sortie et $l \in [2, L]$ aux couches intermédiaires. Nous considérons également des réseaux où toutes les couches, sauf la dernière (qui a une largeur de 1 dans notre configuration), ont une largeur commune $m \in \mathbb{N}$ pour simplifier, et nous sommes intéressés par la description de la limite $m \to \infty$. Dans cette thèse, nous considérons toujours $L \geq 1$, de sorte qu'il y ait au moins 2 couches.

Apprentissage avec des réseaux neuronaux

Une des principales différences de modélisation entre les réseaux neuronaux et les modèles d'apprentissage automatique plus traditionnels (comme les modèles linéaires ou les méthodes à noyau) est que les caractéristiques ne sont pas conçues manuellement, mais sont en réalité apprises automatiquement par le réseau lui-même. Par exemple, on peut toujours écrire la fonction de prédiction, également appelée fonction du réseau, $f(\theta; x) = (\theta^{L+1})^\top \Phi(\theta^L; x)$, mais maintenant l'extracteur de caractéristiques Φ a ses propres paramètres (θ^L est la concaténation des poids des L premières couches), qui peuvent être appris simultanément avec les paramètres de la couche de prédiction $\theta^{L+1} = W^{L+1}$.

Bien que cela semble attrayant, cela entraîne de nombreuses complications. Tout d'abord, même lorsque la perte ℓ est convexe, la fonction objective F correspondant au problème de minimisation du risque n'est **pas convexe** dès que $L \geq 1$, c'est-à-dire lorsqu'il y a au moins deux couches ou plus. Par conséquent, aucune garantie en termes d'optimisation ne peut être attendue a priori, et pire encore, à mesure que le nombre de couches et de paramètres augmente, le problème devient très non convexe, et l'on peut s'attendre à de nombreux plateaux et/ou points de selle où la descente de gradient peut rester bloquée. Il n'existe aucune connaissance de la valeur θ^* à laquelle l'algorithme d'apprentissage convergera (s'il le fait), ce qui rend difficile l'étude de ses propriétés statistiques, et avec le nombre de paramètres étant beaucoup plus élevé que le nombre de points de données d'entraînement, il est à prévoir que les réseaux profonds auront tendance à surajuster et à mal se comporter sur les données de test (erreur de généralisation). D'où la nécessité d'étudier l'ensemble de la trajectoire d'entraînement le long du chemin d'optimisation, mais la nature non linéaire de la dynamique et le grand nombre de paramètres à suivre en font une tâche difficile.

Descente de gradient pour les réseaux neuronaux

Bien que les réseaux neuronaux soient des fonctions paramétriques complexes, l'algorithme pour les entraîner est étonnamment simple. On initialise les paramètres de manière aléatoire : les poids de différentes couches sont initialisés *indépendamment*, et dans une couche donnée, toutes les entrées W^l_{ij} de W^l sont initialisées i.i.d. suivant une loi donnée, par exemple, gaussienne ou uniforme. Dans

cette thèse, nous considérons généralement le cas d'une initialisation gaussienne $W_{ij}^l(0) \sim \mathcal{N}(0, \sigma_l^2)$ i.i.d. pour i, j, avec une variance σ_l^2 qui dépend de la largeur m. À partir de cette initialisation, on suit le gradient négatif de la fonction objective : pour tout $t \geq 0$

$$W^l(t+1) = W^l(t) - \eta_l \nabla_{W^l} F(\theta),$$

où $\eta_l > 0$ est le *taux d'apprentissage* associé à la couche l, qui peut varier d'une couche à l'autre et dépend souvent de m également. De nombreuses variantes différentes de la descente de gradient (stochastique) existent, où le taux d'apprentissage peut également dépendre de l'instant t, et les différentes coordonnées des gradients peuvent être mises à l'échelle différemment en fonction des directions de croissance plus rapide.

Bien qu'aucune garantie n'existe a priori pour les algorithmes basés sur le gradient dans un contexte non convexe, cette recette relativement simple a connu un énorme succès en pratique. Cependant, il est difficile de l'analyser mathématiquement en raison

de la structure hautement non linéaire et compositionnelle des réseaux neuronaux. Il est courant dans les études théoriques de trouver la version de la descente de gradient où $\eta_l \to 0^+$, qui est appelée *flot de gradient* (GF). Il s'agit de l'équivalent en temps continu de la descente de gradient discrète, et il est décrit par l'équation différentielle ordinaire (EDO) $\frac{d}{dt} W^l(t) = -\nabla_{W^l} F(\theta(t))$.

Algorithme de descente de gradient stochastique. Lorsque le nombre n d'échantillons d'entraînement est élevé, il est courant de calculer le gradient sur un sous-ensemble, appelé un *lot* ou un *mini-lot*, plutôt que sur l'ensemble complet d'entraînement. En appelant $F_i(\theta) := \ell(f^*(x_i), f(\theta; x_i))$ pour $i \in [1, n]$, le *gradient complet* est $\nabla_{W^l} F(\theta) = \frac{1}{n} \sum_{i=1}^n \nabla_{W^l} F_i(\theta)$, tandis que pour $\mathcal{B} \subset [1, n]$, le gradient par lot (gradient approximatif) est $\hat{\nabla}_{W^l} F(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_{W^l} F_i(\theta)$. Nous appelons **descente de gradient** (GD) l'optimisation de F en calculant directement les gradients complets à chaque étape, et nous appelons **descente de gradient stochastique** (SGD) l'optimisation de F en sous-échantillonnant des lots parmi l'ensemble d'entraînement à chaque étape, jusqu'à la limite où il pourrait y avoir un seul échantillon $x_t \in \{x_1, \ldots, x_n\}$ à chaque étape.

Passages avant, arrière et rétropropagation. Les calculs des gradients impliqués dans les mises à jour des poids sont appelés le **passage en arrière**, et ils sont calculés de manière récursive à l'aide des équations de rétropropagation. Le gradient par rapport à n'importe quelle variable z peut toujours être décomposé de la manière suivante : $\nabla_z F_i(\theta) = \partial_2 \ell(f^*(x_i), f(\theta; x_i)) \nabla_z f(\theta; x_i)$, et les équations

récursives se lisent comme suit :

$$\begin{aligned}
\nabla_{x^L} f(\theta; x_i) &= W^{L+1}, \quad \nabla_{h^L} f(\theta; x_i) = W^{L+1} \odot \sigma'(h^L), \\
\nabla_{W^{L+1}} f(\theta; x_i) &= x^L, \\
\nabla_{x^l} f(\theta; x_i) &= (W^{l+1})^\top \nabla_{h^{l+1}} f(\theta; x_i), \quad l \in [1, L-1] \\
\nabla_{h^l} f(\theta; x_i) &= (\nabla_{x^l} f(\theta; x_i)) \odot \sigma'(h^l), \quad l \in [1, L-1] \\
\nabla_{W^l} f(\theta; x_i) &= \nabla_{h^l} f(\theta; x_i)(x^{l-1})^\top, \quad l \in [1, L]
\end{aligned}$$

En revanche, le calcul des (pré-) activations h^l, x^l et la sortie $f(\theta; x)$ est appelé le **passage avant**.

Variance initiale et taux d'apprentissage. Le choix de la variance initiale et des taux d'apprentissage peut avoir un impact considérable sur le comportement du réseau pendant et après l'entraînement, en particulier à large largeur (voir la Section 1.2 et le Chapitre 2). À largeur finie, une manière de mettre à l'échelle les variances et les taux d'apprentissage à l'initialisation est discutée dans (Glorot and Bengio, 2010; He et al., 2015), mais une analyse au-delà du premier passage avant et arrière fait encore défaut car les équations récursives et la présence de non-linéarités imbriquées entravent rapidement toute analyse théorique. Cependant, à large largeur, le Tensor Program (Yang and Hu, 2021), que nous présentons brièvement dans la Section 1.2.5, permet d'analyser précisément les magnitudes des passages avant et arrière à chaque étape pour les réseaux de neurones avec une initialisation *i.i.d.*, et ce type d'analyse joue un rôle important dans cette thèse.

Questions ouvertes et orientations de recherche

Les réseaux neuronaux modernes nécessitent de nombreux ingrédients pour atteindre un haut niveau de performance sur des tâches difficiles, tels qu'une grande largeur et profondeur, des couches de normalisation, des connexions résiduelles ou des méthodes de gradient adaptatives. Ces ingrédients sont essentiels pour le succès pratique des réseaux neuronaux, mais ils sont difficiles à analyser mathématiquement, et il n'est pas clair quel est le bénéfice exact de chacun d'eux d'un point de vue théorique. De plus, de nombreuses questions sur les réseaux neuronaux restent non résolues : comment l'apprentissage basé sur le gradient parvient-il à trouver de bonnes valeurs pour les poids et à converger compte tenu du grand nombre de paramètres ? Comment parviennent-ils à atteindre (presque) une perte nulle lorsque l'objectif n'est pas convexe ? Pourquoi ces modèles généralisent-ils si bien alors qu'ils pourraient facilement surajuster, avec de nombreuses valeurs de paramètres conduisant à une perte nulle et aucun contrôle sur le paramètre appris θ^* à la fin de l'entraînement ? Que ces modèles apprennent-ils réellement et que signifient les valeurs des poids appris ?

Dans cette thèse, nous nous concentrons sur les aspects théoriques liés à la compréhension des propriét

és de la dynamique d'entraînement (c'est-à-dire le chemin d'optimisation) des réseaux à largeur infinie dans différents contextes.

Organisation de la thèse

Le reste de la thèse est organisé comme suit : la Section 1.2 est consacrée à la présentation de la littérature et des outils mathématiques autour des limites de largeur infinie, la Section 1.3 met en avant les principales contributions de cette thèse, le Chapitre 2 étudie la limite de largeur infinie des réseaux profonds dans la paramétrisation intégrable (voir la Section 1.2.3 pour une définition) et le Chapitre 3 est consacré à l'étude des symétries qui émergent dans la dynamique des réseaux à deux couches de largeur infinie. Enfin, le Chapitre 4 étudie les propriétés des algorithmes d'optimisation sur l'espace des mesures où les neurones peuvent être ajoutés ou supprimés de manière dynamique au sein des itérations des algorithmes.

Limites de largeur infinie, un chemin prometteur pour étudier le problème de manière rigoureuse

Contexte général et motivation

Une longue lignée de recherches autour des réseaux de neurones à largeur infinie

Les limites de largeur infinie des réseaux de neurones ont une longue histoire, remontant à Barron (1993) et Neal (1995). Le premier montre que toute fonction suffisamment régulière peut être approximée de manière uniforme sur des boules fermées par des réseaux de neurones à deux couches avec des fonctions d'activation de type sigmoïde (*i.e.*, des fonctions mesurables bornées satisfaisant $\lim_{-\infty} \sigma = 0$ et $\lim_{+\infty} \sigma = 1$) et le niveau d'approximation obtenu peut être arbitrairement petit à condition que le nombre de neurones de la première couche soit autorisé à *croître indéfiniment*. Pinkus (1999) va même plus loin en montrant que l'approximation uniforme sur n'importe quel ensemble compact est garantie si et seulement si la fonction d'activation n'est pas polynomiale, à condition qu'elle soit continue. La réserve ici est que bien que les fonctions puissent être approximées avec une précision arbitraire sur des ensembles compacts par des réseaux de neurones, trouver effectivement de bonnes valeurs de paramètres qui réalisent cette approximation à partir de données finies est difficile a priori. Dans une ligne de travail distincte, Neal (1995) adopte un point de vue bayésien et prouve que la fonction du réseau de neurones converge vers un processus gaussien lorsque le nombre de paramètres tend vers l'infini et que leur distribution est gaussienne.

Plus récemment, Bengio et al. (2006) démontrent que l'objectif d'entraînement des réseaux de neurones à deux couches peut être convexe (dans un espace potentiellement de dimension infinie) dès lors que la fonction de perte est convexe,

si l'on considère un nombre infini de neurones, ce qui conduit à des algorithmes pouvant potentiellement atteindre le minimum global. Suivant cette idée, Bach (2017) montre que les réseaux à deux couches de largeur infinie avec des activations positivement homogènes forment une classe de fonctions ayant des propriétés statistiques favorables : notamment que, en présence d'une structure de dimension inférieure, l'erreur de généralisation dépend uniquement de la dimension du sous-espace et non de celle de l'espace ambiant. Cependant, il est souligné que la minimisation du risque empirique dans ce contexte (ou sa version attendue) est un problème difficile du point de vue computationnel.

Pourquoi étudier les réseaux de largeur infinie ?

Les réseaux neuronaux profonds (même dans leur forme la plus simple) sont des objets très non linéaires et leur dynamique d'entraînement correspond à l'optimisation de fonctions complexes et non convexes, ce qui les rend difficiles à analyser sur le plan théorique. Cependant, comme présenté ci-dessus, d'importants résultats théoriques ont été obtenus en considérant des limites où le nombre de neurones dans une couche peut devenir indéfiniment grand. Comme nous le discutons tout au long de cette section, il y a eu un regain d'intérêt récent pour les asymptotiques de grande largeur en raison de plusieurs résultats qui éclairent le comportement des réseaux de neurones et aident à comprendre pourquoi ils fonctionnent si bien en pratique. Parmi ces résultats, on trouve la **convergence globale** de la descente de gradient (*e.g.*, Mei et al., 2018; Chizat and Bach, 2018; Wojtowytsch, 2020; Jacot et al., 2018), des informations sur leur dynamique d'entraînement en révélant une forme de **biais implicite** (Chizat and Bach, 2020), ainsi que des résultats statistiques sur les propriétés de généralisation de ces modèles (Bach, 2017; Chizat and Bach, 2020).

De plus, avec l'accélération rendue possible par les avancées dans le matériel moderne, les réseaux neuronaux de pointe ont un grand nombre de paramètres (jusqu'à *plusieurs centaines de milliards*), ce qui rend l'étude de la limite où le nombre de paramètres tend vers l'infini non déraisonnable. De plus, il est montré dans (Nguyen and Pham, 2020) que la dynamique des réseaux de largeur infinie suit de près celle des réseaux avec suffisamment de neurones, et Yang and Hu (2021); Yang et al. (2022) démontrent que les résultats théoriques sur les réseaux de largeur infinie peuvent se traduire en connaissances pratiques sur les réseaux de largeur finie du monde réel dont le comportement est parfois bien décrit par la théorie de leur homologue de largeur infinie.

En résumé, les limites de largeur infinie des réseaux de neurones semblent être un moyen élégant d'adopter un point de vue théorique tout en conduisant à des intuitions pratiques : elles se prêtent bien à l'analyse théorique et représentent une approche mathématiquement fondée qui a porté ses fruits pour approfondir notre compréhension de certaines questions liées à l'optimisation et la généralisation.

Approche intuitive de la limite de largeur infinie

Nous présentons ici des idées informelles et des calculs qui permettent de comprendre en quoi consiste la limite de largeur infinie des réseaux neuronaux à deux couches (parfois aussi appelés réseaux à une couche cachée) et comment on pourrait envisager l'objet limite. Prendre rigoureusement cette limite est souvent subtil et nécessite beaucoup de travail technique, c'est pourquoi de nombreux articles de la littérature (que nous examinerons ci-dessous) se penchent sur ces questions. Notre travail, en revanche, ne se concentre pas tant sur la rigueur mathématique de la limite que sur l'exploitation des outils disponibles pour produire de nouvelles idées et de nouveaux résultats sur les réseaux de largeur infinie.

Rappelons qu'un réseau neuronal à deux couches de largeur m avec une sortie réelle est une fonction paramétrique $f(\theta;\cdot) : \mathbb{R}^d \to \mathbb{R}$ définie comme suit :

$$f(\theta; x) = \sum_{j=1}^{m} w_j^2 \sigma(x^\top w_j^1),$$

où $\theta = ((w_j^2, w_j^1))_{j\in[1,m]} \in (\mathbb{R} \times \mathbb{R}^d)^m$ est la liste des paramètres composée de la matrice de poids d'*entrée* $w^1 = (w_1^1, \dots, w_m^1)$ et des poids de *sortie* $(w_1^2, \dots, w_m^2)$, et $\sigma : \mathbb{R} \to \mathbb{R}$ est la fonction d'activation. Nous soulignons que w_j^1 représente le j-ème neurone de la première couche ($l = 1$) et w_j^2 représente la j-ème entrée du seul neurone de la deuxième couche ($l = 2$), de sorte que les indices supérieures **ne représentent pas** des puissances.

Prendre la limite de largeur infinie revient à prendre la limite $m \to \infty$, ce qui implique une somme infinie et donc des problèmes liés à la convergence. Une manière naturelle de s'assurer que la somme reste finie lorsque $m \to \infty$ est d'ajouter un facteur d'échelle devant la somme qui est une puissance négative fixe de m, c'est-à-dire de considérer la nouvelle paramétrisation $f(\theta; x) = m^{-a} \sum_{j=1}^{m} w_j^2 \sigma(x^\top w_j^1)$ du réseau neuronal avec $a > 0$. Remarquez que cela ne change pas la classe de fonctions que nous considérons, car le facteur m^{-a} peut être incorporé dans les poids de sortie w_j^2. Toutes les valeurs de a ne garantissent pas la convergence de la somme, mais des valeurs suffisamment grandes évitent les problèmes.

Il faut considérer les paramètres $(w_j^2, w_j^1)_{j\in[1,m]}$ comme des variables aléatoires, c'est le cas à l'initialisation et cela reste vrai tout au long de l'entraînement. En tant que tels, il existe différents modes de convergence pour la somme. En général, avec des paramètres i.i.d. $((w_j^2, w_j^1))_{j\in[1,m]}$ (comme c'est le cas à l'initialisation), on a une convergence en loi pour $a = 1/2$ (par le théorème central limite) et une convergence presque sûre pour $a = 1$ (par la loi des grands nombres) lorsque $m \to \infty$. Il s'avère que les échelles $a = 1/2$ et $a = 1$ sont largement étudiées dans la littérature sur les réseaux de largeur infinie, la première étant appelée la paramétrisation du neural tangent kernel (NTK) et la seconde étant souvent appelée la paramétrisation mean-field, bien que nous trouvions que cette dénomination soit quelque peu ambiguë pour les réseaux avec plus de deux couches, car

la généralisation correcte aux couches plus profondes pose des difficultés dans le cadre "mean-field" (discuté plus en détail dans la Section 1.2.3), et nous préférons donc le terme *Paramétrisation Intégrable* (PI) en référence au fait que la somme rénormalisée est absolument convergente. Comme nous le verrons bientôt (voir Sections 1.2.2, 1.2.3 et 1.2.5), ces différentes échelles conduisent à des comportements très différents pour le modèle limite, et cette thèse se concentre sur la paramétrisation intégrable.

Paramétrisations de réseaux de n'importe quelle profondeur

Généraliser aux réseaux plus profonds l'intuition présentée ci-dessus pour la limite de largeur infinie des réseaux à deux couches n'est pas toujours simple. En effet, on peut toujours introduire des facteurs m^{-a_l} pour chaque couche l (ou du moins pour les couches $l \geq 2$) avec $a_l \geq 0$, ce qui conduit à la paramétrisation d'un réseau à L couches cachées comme suit :

$$\begin{aligned} f(\theta; x) &= m^{-a_{L+1}} (w^{L+1})^\top x^L \\ x^l &= \sigma(h^l), \quad h^l = m^{-a_l} w^l x^{l-1}, \quad l \in [2, L], \\ x^1 &= \sigma(h^1), \quad h^1 = w^1 x \end{aligned}$$

où $w^{L+1} \in \mathbb{R}^m$, $w^l \in \mathbb{R}^{m \times m}$ pour $l \in [2, L]$, $w^1 \in \mathbb{R}^{m \times d}$, et h^l désigne les pré-activations à la couche l, et x^l les activations à la couche l, et par défaut x^0 désigne simplement l'entrée x alimentée à la première couche du réseau. Nous n'avons pas besoin de réduire l'échelle de la première couche car les sommes dans les produits scalaires qui y apparaissent sont toujours finies, comprenant autant de termes que la dimension d'entrée d.

Prendre la limite $m \to \infty$ lorsque $L \geq 2$ rend les choses plus difficiles (même avec $\sigma = \mathrm{id}$), car il faut gérer des sommes infinies imbriquées. Comme cela est examiné dans les sections suivantes, il existe divers cadres mathématiques et outils pour prendre cette limite en fonction de la paramétrisation envisagée, mais le Tensor Programiel (décrit à la Section 1.2.5) fournit un point de vue complet pour dériver rigoureusement la limite de n'importe quelle paramétrisation avec des techniques et des idées issues de la littérature de physique statistique pour traiter des matrices aléatoires de taille tendant vers l'infini.

Il s'avère que pour comprendre la dynamique d'entraînement de tels modèles dans la limite $m \to \infty$, une description plus complète d'une ***paramétrisation*** du réseau est donnée en ajoutant des facteurs d'échelle m^{-b_l} ($b_l \geq 0$) à l'écart-type de la distribution initiale des poids dans la couche l et des facteurs d'échelle m^{-c_l} pour le taux d'apprentissage de la couche l appliqué aux mises à jour des poids. Autrement dit, les matrices w^l sont initialisées i.i.d. avec une loi telle que $\tilde{W}^l_{ij}(0) := m^{b_l} w^l_{ij}(0)$ ait une variance égale à un (ou du moins indépendante de m), et la règle de mise à jour des poids de la couche l est donnée par $w^l(t+1) = w^l(t) - \eta_l m^{-c_l} \nabla_{w^l} F(\theta(t))$. Cela s'appelle la abc-paramétrisation d'un réseau

neuronal dans (Yang and Hu, 2021). Il y a une *redondance* entre les trois échelles a_l, b_l et c_l, car deux d'entre elles suffisent pour fournir une image complète : on peut par exemple toujours choisir d'initialiser les matrices avec une variance unitaire (c'est-à-dire $b_l = 0$) ou utiliser alternativement un taux d'apprentissage unitaire ($c_l = 0$) sans restreindre la classe des paramétrisations considérées. En effet, en considérant les matrices de poids effectives $W^l(t) = m^{-a_l} w^l(t)$ qui sont réellement utilisées dans le calcul, on a $\nabla_{W^l} F(\theta(t)) = m^{-a_l} \nabla_{w^l} F(\theta(t))$ et donc

$$W^l(t) = m^{-(a_l+b_l)} \tilde{W}^l(0) - \eta m^{-(2a_l+c_l)} \sum_{s=0}^{t-1} \nabla_{W^l} F(\theta(s)). \tag{1.10}$$

Il est alors clair que, en partant de la même initialisation $\tilde{W}^l(0)$, n'importe quelle paramétrisation pour laquelle $a_l + b_l$ et $2a_l + c_l$ ont la même valeur conduira aux mêmes poids effectifs et donc à la même fonction. Par conséquent, n'importe quelle paramétrisation peut être exprimée avec $b_l = 0$ ou $c_l = 0$ (mais pas les deux en même temps). Alors que Yang and Hu (2021) décident d'ignorer les valeurs du taux d'apprentissage (ils considèrent principalement $c_l = 0$), nous choisissons de considérer des paramétrisations ac où les matrices de poids initiales sont toujours initialisées avec une variance unitaire ($b_l = 0$). Le nom d'une paramétrisation fait principalement référence au choix de l'échelle pour les poids (a_l), par exemple NTK ($a_l = 1/2$) ou PI ($a_l = 1$), bien que le choix du taux d'apprentissage (c_l) ait également son importance.

La paramétrisation NTK

Lorsqu'on utilise une initialisation gaussienne i.i.d. pour les poids d'un réseau neuronal, l'échelonnement des écarts types initiaux en tant que $m^{-1/2}$ est apparu comme une manière naturelle de préserver le signal dans les premières passes avant et arrière (Glorot and Bengio, 2010; He et al., 2015). Comme détaillé ci-dessus, cet échelonnement de l'écart type initial peut être également compris comme un préfacteur d'échelle de $m^{-1/2}$ devant les poids, ce qui caractérise la paramétrisation NTK. Avec ce facteur d'échelle, il est déjà compris depuis Neal (1995) que cela aboutit à un processus gaussien pour la sortie des réseaux peu profonds à l'initialisation lorsque la largeur $m \to \infty$. Jacot et al. (2018), qui ont inventé le terme "Noyau Tangent Neuronal" (NTK), vont même plus loin en prouvant que la dynamique d'entraînement des réseaux entièrement connectés de n'importe quelle profondeur dans cette paramétrisation peut être décrite comme une méthode du noyau avec un noyau spécifique que nous détaillons ci-dessous. Yang (2020a) dérive rigoureusement la généralisation de cette description du noyau à n'importe quelle architecture.

La description du noyau de la dynamique de la paramétrisation NTK dans la limite de largeur infinie revient à dire que la fonction de prédiction évolue comme

$$f(\theta(t+1); x) = f(\theta(t); x) - \frac{\eta}{n} \sum_{i=1}^{n} \chi_{t,i} K(x, x_i)$$

pour un certain noyau $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, où $(x_i)_{i\in[1,n]}$ sont les n échantillons dans l'ensemble de données d'entraînement et $\chi_{t,i} := \partial F(\theta(t))/\partial f(\theta(t); x_i) = \partial_2 \ell(f^*(x_i), f(\theta(t); x_i))$ est la dérivée de la perte sur l'échantillon x_i au temps t. Comme démontré dans (Chizat et al., 2019), cette propriété de descente de noyau dans la paramétrisation NTK est la conséquence naturelle du fait que l'échelle (ou la magnitude) des mises à jour des poids est beaucoup plus petite que celle des poids initiaux. Tout d'abord, nous expliquons comment cela conduit intuitivement au comportement de noyau décrit ci-dessus, puis nous détaillons pourquoi cette propriété est vraie dans la limite de largeur infinie.

Linéarisation autour de l'initialisation pour le NTK

Supposons qu'à l'initialisation, les gradients des poids dans la paramétrisation NTK sont tels que les mises à jour ont une magnitude bien plus petite que les valeurs initiales des poids $W^l(0)$ pour de grandes valeurs de m. La première mise à jour des paramètres sur un lot de n échantillons s'écrit comme $\Delta\theta = -\frac{\eta}{n}\sum_{i=1}^n \nabla_\theta F_i(\theta(0))$, et le gradient par rapport aux paramètres peut être décomposé en $\nabla_\theta F_i(\theta(0)) = \chi_{0,i}\nabla_\theta f(\theta(0); x_i)$. En supposant que $||\Delta\theta||$ est bien plus petit que $||\theta(0)||$, on peut **linéariser** la prédiction autour de ses paramètres initiaux :

$$\begin{aligned} f(\theta(1); x) &= f(\theta(0) + \Delta\theta; x) \\ &\approx f(\theta(0); x) + \Delta\theta^\top \nabla_\theta f(\theta(0); x) \\ &= f(\theta(0); x) - \frac{\eta}{n}\sum_{i=1}^n \chi_{0,i} \nabla_\theta f(\theta(0); x_i)^\top \nabla_\theta f(\theta(0); x) \end{aligned}$$

ce qui correspond exactement à une descente de noyau avec un noyau appelé le noyau tangent neuronal (Jacot et al., 2018), défini par la formule $K_m(x, y) := \nabla_\theta f(\theta(0); x)^\top \nabla_\theta f(\theta(0); y)$ pour une largeur m. Il s'agit d'un noyau produit scalaire, bien que dans un espace dont la dimension tend vers l'infini à mesure que m devient grand. Deux faits sont remarquables à propos de ce noyau : (i) il converge (presque certainement) vers un noyau limite *déterministe* K_∞ lorsque $m \to \infty$; et (ii) il reste en réalité constant dans le temps dans la limite de largeur infinie, *i.e.*, $\lim_{m\to\infty} \nabla_\theta f(\theta(t); x)^\top \nabla_\theta f(\theta(t); y) = \lim_{m\to\infty} \nabla_\theta f(\theta(0); x)^\top \nabla_\theta f(\theta(0); y)$ pour tout $t \geq 0$. Le deuxième point découle également du fait que les mises à jour des poids dans la paramétrisation NTK sont beaucoup plus petites en magnitude que les poids initiaux. Ce phénomène, appelé *"lazy training"* dans (Chizat et al., 2019), ne peut pas expliquer aussi bien l'apprentissage de caractéristiques que les capacités de transfert des réseaux neuronaux utilisés en pratique (dans les systèmes de vision par ordinateur ou dans les grands modèles de langage).

Déplacement infinitésimal des caractéristiques dans le NTK

Expliquons maintenant pourquoi les mises à jour des poids ont une magnitude bien plus petite que les poids initiaux. Pour fixer les idées, considérons que les gradients

sont calculés à l'aide d'un seul échantillon et que l'initialisation est gaussienne, de sorte que les poids initiaux (effectifs) s'écrivent comme $W^l(0) = m^{-1/2}\tilde{W}^l(0)$ pour $l \in [2, L+1]$ et $W^1(0) = \tilde{W}^1(0)$ où $\tilde{W}^l(0)$ a des entrées i.i.d. suivant $\mathcal{N}(0,1)$ pour tout l. Rappelons que les mises à jour des poids pour n'importe quelle paramétrisation sont données dans l'Équation (1.1). Les gradients par rapport aux poids W^l sont donnés par les équations de rétropropagation : $\nabla_{W^l} F(\theta(t)) = \chi_t \nabla_{h^l} f(\theta(t); x_t)(x_t^{l-1})^\top$, où $\chi_t = \partial F(\theta(t))/\partial f(\theta(t), x_t)$ est la dérivée de la perte sur l'échantillon d'entraînement x_t au temps t. Pour la paramétrisation NTK, les premières mises à jour des poids sont données par

$$\begin{aligned} W^1(1) &= \tilde{W}^1(0) - \eta\chi_0 \nabla_{h^1} f(\theta(0); x_0) x_0^\top \\ W^l(1) &= m^{-1/2}\tilde{W}^l(0) - \eta m^{-1}\chi_0 \nabla_{h^l} f(\theta(0); x_0)(x_0^{l-1})^\top, \quad l \in [2, L] \\ W^{L+1}(1) &= m^{-1/2}\tilde{W}^{L+1}(0) - \eta m^{-1}\chi_0 x_0^L. \end{aligned}$$

Pour les couches $l \geq 2$, le facteur m^{-1} dans la mise à jour des poids par rapport au facteur $m^{-1/2}$ présent dans le poids initial suggère déjà la différence de magnitude entre ces deux contributions au poids $W^l(1)$ pour de grandes valeurs de m. Il reste à analyser la magnitude réelle des entrées du terme $\nabla_{h^l} f(\theta(0); x_0)(x_0^{l-1})^\top$ lorsque m devient grand. La magnitude des activations x_0^l pour la paramétrisation NTK est bien comprise depuis Neal (1995) : les facteurs $m^{-1/2}$ associés à l'initialisation gaussienne i.i.d. garantissent que la première passe initiale est de l'ordre de 1 (voir la Section 1.2.5 pour plus de détails). L'échelle des gradients $\nabla_{h^l} f(\theta(0); x_0)$, en revanche, n'est pas aussi directe et doit être dérivée de manière récursive. Essentiellement, il découle des équations de rétropropagation que les coordonnées de ces gradients sont de l'ordre de $m^{-1/2}$. On en déduit donc la magnitude relative des mises à jour $\Delta W^l = W^l(1) - W^l(0)$ par rapport à l'initialisation : $||\Delta W^l||/||W^l(0)||$ est de l'ordre de $m^{-1/2}$ pour les premières et dernières couches $l \in \{1, L+1\}$ et de l'ordre de m^{-1} pour les couches intermédiaires $l \in [2, L]$.

Les poids se déplacent donc loin de leur initialisation uniquement d'une quantité infinitésimale dans la paramétrisation NTK en limite de grande largeur. Mais comment se fait-il alors que la fonction de sortie évolue encore pendant l'entraînement et ne reste pas à sa valeur initiale ? C'est parce que bien que toutes les entrées soient individuellement petites par rapport à l'initialisation, elles induisent *collectivement* un résultat non nul dans les produits scalaires impliqués dans les multiplications matricielles de la passe avant. Le programme tensoriel permet précisément de dériver rigoureusement les échelles des mises à jour et les produits scalaires lorsque $m \to \infty$ et justifie les calculs informels présentés ci-dessus.

Dans la paramétrisation NTK, il semble donc que l'évolution soit uniquement décrite dans l'espace des fonctions : les paramètres du réseau ne semblent pas s'éloigner significativement de leur initialisation. **Cruciallement**, il est même prouvé dans (Yang and Hu, 2021) que lorsque $m \to \infty$, les caractéristiques x_t^l de n'importe quelle couche l au temps t ne s'éloignent pas significativement de leur initialisation non plus, au sens que pour la même entrée $x \in \mathbb{R}^d$, $||x_t^l - x_0^l||^2/m$

converge vers 0 lorsque $m \to \infty$, où le terme $1/m$ est simplement ici pour renormaliser une somme qui devient infinie et qui pourrait autrement exploser. Ce n'est pas surprenant puisque les dynamiques NTK ressemblent à l'apprentissage avec une méthode de noyau, ce qui revient à apprendre un prédicteur linéaire sur des caractéristiques fixes (bien que de dimension infinie). Par conséquent, bien que la paramétrisation NTK et sa limite à largeur infinie soient attrayantes pour leurs propriétés théoriques, elles ne suffisent pas à saisir la richesse des dynamiques des réseaux neuronaux profonds du monde réel.

Convergence globale du NTK

Malgré ses inconvénients en termes d'apprentissage de caractéristiques, la limite à largeur infinie de la paramétrisation NTK produit toujours des résultats théoriques intéressants tels que la convergence de l'objectif vers un minimum global. En effet, il est démontré dans (Jacot et al., 2018) que si la perte est convexe, les dynamiques NTK conduisent à une convergence vers le minimum global. Par exemple, pour l'objectif de perte quadratique $F(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(\theta; x_i) - y_i)^2$, les dynamiques NTK conduisent à une convergence exponentiellement rapide de $f(\theta(t); x_i)$ vers y_i en limite de largeur infinie. Cela découle du fait que lorsque l'on considère le flot de gradient de l'objectif (la limite de la descente de gradient lorsque le pas $\eta \to 0^+$), le vecteur de prédiction $\bar{y}_t = (f(\theta(t); x_i))_{i\in[1,n]}$ satisfait $\frac{d}{dt}(\bar{y}_t - y^*) = -\bar{K}_\infty(\bar{y}_t - y^*)$ où $y^* = (y_i)_{i\in[1,n]}$ est le vecteur des cibles et $\bar{K}_\infty$ est la matrice NTK définie par $\bar{K}_{\infty,ij} = K_\infty(x_i, x_j)$. Cela conduit à $\bar{y}_t = y^* + e^{-t\bar{K}_\infty}(\bar{y}_0 - y^*)$ ce qui garantit la convergence de $\bar{y}_t$ vers y^* lorsque $t \to \infty$, à condition que $\bar{K}_\infty$ soit défini positif.

Paramétrisation intégrable

Les paramétrisations intégrables se caractérisent par le facteur d'échelle m^{-1} devant les poids et présentent des propriétés assez différentes de la paramétrisation NTK. Cette thèse se concentre sur les PIs, c'est pourquoi la littérature et les résultats autour de ces types de modèles, que nous examinons dans cette section, sont particulièrement pertinents pour notre travail.

Limite à largeur infinie

Avec deux couches, la paramétrisation intégrable a la forme suivante :

$$f(\theta; x) = \frac{1}{m}\sum_{j=1}^{m} w_j^2 \sigma(x^\top w_j^1), \tag{1.11}$$

et est souvent appelée un modèle "mean-field", car les moyennes de ce type sont fréquentes en physique statistique, où l'étude de systèmes avec un nombre croissant de particules en interaction est courante. L'intuition est que lorsque m (ici le nombre de neurones, mais on peut le considérer comme le nombre de particules d'un système) est grand, en raison du terme $1/m$ devant la somme, la fonction

se comportera comme une moyenne sur une certaine mesure. En effet, lorsque $m \to \infty$, il est naturel de remplacer la somme par une intégrale (on peut penser à la loi des grands nombres) par rapport à une mesure de probabilité $\mu \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R})$, ce qui conduit à la paramétrisation suivante :

$$f(\mu; x) = \int_{(w^1, w^2) \in \mathbb{R}^{d+1}} w^2 \sigma(x^\top w^1) \mathrm{d}\mu(w^1, w^2) \tag{1.12}$$

dans la limite à largeur infinie. Pour éviter que l'intégrale ne diverge, nous pouvons restreindre la classe de fonctions paramétrisées aux mesures de probabilité $\mu \in \mathcal{P}_2(\mathbb{R}^d \times \mathbb{R})$, qui ont un moment d'ordre deux fini si σ a au plus une croissance linéaire. Dans ce cadre, n'importe quel réseau à deux couches de largeur m tel que dans l'équation (1.2) peut être obtenu avec une mesure atomique $\mu_m = \frac{1}{m} \sum_{j=1}^m \delta_{(mw_j^2, w_j^1)}$, où δ_w est la mesure de Dirac en w.

Dynamiques sur les mesures

L'objectif à minimiser est désormais une fonctionnelle F dans l'espace des mesures. En général, nous cherchons à minimiser le risque d'une nouvelle classe de fonctions:

$$\min_{\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})} \Big\{ F(\mu) := \mathbb{E}_{x \sim \rho} \left[\ell(f^*(x), f(\mu; x)) \right] \Big\}. \tag{1.13}$$

À largeur finie, nous utilisons la descente de gradient (stochastique) pour réduire la valeur de l'objectif, mais comment procède-t-on dans l'espace des mesures de probabilité ? Des outils issus de la théorie du transport optimal ont été développés à cet effet, et la réponse est les flots de gradient de Wasserstein (WGF), qui sont l'équivalent de la descente de gradient dans l'espace des mesures avec une étape infinitésimale. Les dynamiques correspondantes sont décrites par l'équation aux dérivées partielles (EDP) connue sous le nom d'**équation de continuité** (voir Ambrosio et al., 2005):

$$\begin{aligned} \partial_t \mu_t &= -\mathrm{div}(v_t \mu_t), \\ v_t &= -\nabla F'_{\mu_t} \end{aligned} \tag{1.14}$$

à comprendre dans au sens des distributions. Dans l'équation (1.5), la mesure initiale $\mu_0 \in \mathcal{P}_2(\mathbb{R}^{d+1})$ évolue en fonction d'un champ vectoriel $v_t = -\nabla F'_{\mu_t}$ donné par le gradient de la première variation, ou dérivée de Fréchet, F'_{μ_t} (une fonction de $\mathbb{R}^{d+1}$ à $\mathbb{R}$) de la fonctionnelle F à μ_t. Plus de détails et d'explications mathématiques sur les flots de gradient de Wasserstein, la première variation des fonctionnelles sur les mesures de probabilité et l'équation de continuité sont fournis dans la Section 1.2.4.

L'interprétation naturelle de cette équation est qu'à tout moment t donné, la masse est *déplacée* (ou advectée) selon un champ vectoriel v_t, modifiant ainsi la distribution de la masse μ_t à l'instant t. En fait, une description alternative de l'équation de continuité peut être fournie en utilisant le point de vue d'un

système de particules infiniment nombreuses interagissant les unes avec les autres: considérez une distribution initiale μ_0 de particules $w \in \mathbb{R}^{d+1}$, et considérez le flot $X_t(w)$ défini pour tout $w \in \mathbb{R}^{d+1}$ par

$$\begin{aligned} X_0(w) &= w, \\ \frac{d}{dt} X_t(w) &= v_t(X_t(w)). \end{aligned} \tag{1.15}$$

$X_t(w) \in \mathbb{R}^{d+1}$ représente la position à l'instant t d'une particule initialement située à $w \in \mathbb{R}^{d+1}$ et qui interagit avec toutes les autres particules (à d'autres endroits) à travers le champ de vitesse v_t. Ensuite, étant donné le flot X_t, la solution de l'équation de continuité (1.5) à partir de μ_0 est donnée par le push-forward $\mu_t = X_{t\#}\mu_0$ de la mesure μ_0 par la fonction $X_t(\cdot)$. Autrement dit, la mesure μ_t est simplement la distribution des particules à l'instant t, initialement réparties selon μ_0, et qui ont évolué selon le système (1.6). Notez que de ce point de vue, la distribution des particules à l'instant t détermine la mesure μ_t et donc aussi le champ de vitesse v_t, qui à son tour déterminera dans quelle direction les particules évoluent, de sorte que les particules interagissent effectivement puisque la vitesse d'une particule à un instant donné est déterminée par la position de toutes les autres particules.

Il est important de noter que le WGF (1.5) **retrouve la descente de gradient** sur l'objectif des réseaux de largeur finie. En effet, pour une mesure initiale atomique $\mu_0 = \frac{1}{m}\sum_{j=1}^m \delta_{(w^1(0),w^2(0))}$, le WGF (1.5) correspond exactement à la descente de gradient en temps continu sur les paramètres d'un réseau de largeur finie. En d'autres termes, le WGF $(\mu_{m,t})_{t\geq 0}$ à partir d'une mesure initiale atomique $\mu_{m,0} = \frac{1}{m}\sum_{j=1}^m \delta_{(w^1(0),w^2(0))}$ a la forme $\mu_{m,t} = \frac{1}{m}\sum_{j=1}^m \delta_{(w_j^1(t),w_j^2(t))}$, et les paramètres $\theta(t) = ((w_j^1(t), w_j^2(t)))_{j\in[1,m]}$ sont en réalité donnés par la descente de gradient $\theta'(t) = -m\nabla F_m(\theta(t))$ sur l'objectif de largeur finie défini par $F_m(\theta) = \mathbb{E}_{x\sim\rho}\left[\ell(f^*(x), f_m(\theta; x))\right]$ avec $f_m(\theta; x) = \frac{1}{m}\sum_{j=1}^m w_j^2 \sigma(x^\top w_j^1)$. Le facteur m dans la descente de gradient compense le terme $1/m$ dans la définition de f_m qui réduit l'échelle des gradients. Inversement, si $\theta(t)$ est la descente de gradient de l'objectif de largeur finie F_m, c'est-à-dire $\theta'(t) = -m\nabla F_m(\theta(t))$, alors la mesure atomique $\mu_{m,t} = \frac{1}{m}\sum_{j=1}^m \delta_{(w_j^1(t),w_j^2(t))}$ est le WGF de la fonctionnelle F à partir de $\mu_{m,0}$. De plus, si $\mu_{m,0}$ converge (en distance de Wasserstein) vers μ_0 lorsque $m \to \infty$, alors $\mu_{m,t}$ converges, lorsque $m \to \infty$, vers le WGF de la fonctionnelle F à partir de μ_0 sur n'importe quel intervalle de temps borné. Pour plus de détails sur l'équivalence entre le WGF pour les mesures atomiques et la descente de gradient de largeur finie, voir la Section 1.2.4.

Revue de la littérature

Les modèles de champ moyen sont omniprésents en physique mathématique, mais les PI ont été étudiées récemment en tant que modèles pour les réseaux neuronaux

de largeur infinie. Ils ont rapidement suscité un intérêt en tant qu'approche intéressante pour l'étude des réseaux à deux couches, puis des réseaux plus profonds. Les questions qui se posent lors de l'étude de la limite de largeur infinie des réseaux dans la paramétrisation intégrable sont de nature diverse : existe-t-il une solution à l'équation de continuité (1.5) dans le cadre typique des réseaux neuronaux ? Dans quelle mesure les dynamiques des réseaux de largeur finie diffèrent-elles de la description de largeur infinie ? Peut-on donner des bornes quantitatives en fonction de m ? Y a-t-il convergence des dynamiques lorsque $t \to \infty$? Comment ces modèles se comportent-ils numériquement ?

Le cas des réseaux à deux couches. Une série de travaux étudient ces questions d'un point de vue mathématique pour les réseaux à deux couches (Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Chizat and Bach, 2018; Araújo et al., 2019; Wojtowytsch, 2020; Sirignano and Spiliopoulos, 2020), et établissent la bien-posée de l'Équation (1.5) dans le contexte des réseaux neuronaux à deux couches sous des hypothèses légères sur la fonction de perte et la fonction d'activation, ainsi que la convergence, lorsque le nombre de neurones m tend vers l'infini, des dynamiques de descente de gradient de largeur finie vers les dynamiques dans l'espace des mesures données par l'Équation (1.5). De plus, la convergence de μ_t vers un *minimum global* de l'objectif F lorsque $t \to \infty$ est également démontrée dans (Mei et al., 2018; Chizat and Bach, 2018; Wojtowytsch, 2020) lorsque la perte ℓ est convexe sous des hypothèses légères sur l'initialisation μ_0.

Le résultat de convergence globale nécessite des preuves techniques, mais il est à noter que bien que l'objectif de largeur finie soit non convexe pour les réseaux à deux couches, si la perte ℓ est convexe, puisque la paramétrisation d'un réseau de largeur infinie par une mesure de probabilité comme dans l'Équation (1.3) est *linéaire* par rapport à la mesure μ, l'objectif F est maintenant convexe par rapport à μ. Il s'agit d'une bonne propriété pour l'optimisation, mais cela ne garantit pas toujours la convergence globale du WGF en général, la bonne propriété est celle de *convexité par déplacement* (convexité le long des géodésiques), mais elle ne s'applique pas toujours dans le contexte des réseaux neuronaux. Il est important de noter que les résultats de convergence globale pour les réseaux à deux couches dans les paramétrisations intégrables ont une *nature différente* de ceux discutés pour la paramétrisation NTK. En effet, dans la paramétrisation intégrable, les dynamiques données par le WGF (1.7) sont vraiment *non linéaires* et impliquent que les poids évoluent de manière non triviale par rapport à leur initialisation : les caractéristiques sont effectivement apprises par le réseau au fur et à mesure de l'entraînement.

Résultats statistiques. Pour les réseaux à deux couches, la classe de fonctions représentée par la limite de largeur infinie de la paramétrisation intégrable présente également des propriétés statistiques intéressantes. Bach (2017) étudie

leurs propriétés statistiques et d'approximation et montre que lorsque la fonction cible ne dépend que de la projection sur un sous-espace de dimension réduite (inconnu), ces réseaux évitent la malédiction de la dimension avec des bornes d'approximation et de généralisation qui dépendent uniquement de manière exponentielle de la dimension du sous-espace.

Dans le contexte de la classification binaire, Chizat and Bach (2020) montrent que pour des pertes à queues exponentielles, le WGF (1.5) conduit à un prédicteur qui est un classifieur à marge maximale lorsque $t \to \infty$. Il s'agit d'une forme de *biais implicite* des dynamiques de descente de gradient : le WGF ne converge pas vers un minimiseur global, mais vers un minimiseur qui réalise la marge maximale, et possède donc des propriétés de généralisation favorables. En effet, lorsque qu'il existe un sous-espace de dimension réduite pour lequel la projection des données présente une distance inter-classe suffisamment grande, la marge est indépendante de la dimension ambiante, ce qui conduit à une borne supérieure sur la probabilité de classification incorrecte qui dépend uniquement de la dimension du sous-espace.

Les résultats solides discutés ci-dessus pour les réseaux à deux couches, ainsi que le fait que les poids s'éloignent effectivement de leur initialisation, démontrent que la limite de largeur infinie des paramétrisations intégrables est une avenue de recherche prometteuse pour approfondir notre compréhension des réseaux neuronaux et justifient la croissance du nombre de travaux sur ces modèles.

Réseaux à couches multiples. Généraliser le résultat obtenu pour les réseaux à deux couches à des réseaux plus profonds n'est pas facile (voir Nguyen and Pham, 2020). En effet, la particularité des réseaux à deux couches est qu'il y a une *échangeabilité* des neurones due à l'invariance par permutation dans la somme de l'Équation (1.2). Pour trois couches ou plus, certains poids apparaîtront dans tous les termes de la somme, ce qui entraîne des complications (nous ne faisons pas la somme sur des parties indépendantes de l'ensemble de paramètres) et la notion d'échangeabilité de base est perdue. Cependant, il existe encore de nombreux travaux qui étudient des réseaux plus profonds dans la paramétrisation intégrable (Nguyen and Pham, 2020; Fang et al., 2020; Sirignano and Spiliopoulos, 2021; Araújo et al., 2019). Cependant, ils soulignent tous la difficulté de décrire correctement les dynamiques de la limite de largeur infinie lorsque le réseau comporte plus de trois couches, et ils présentent tous différentes descriptions des dynamiques de la limite de largeur infinie, nécessitant soit des hypothèses spécifiques, soit entraînant des propriétés indésirables. Parmi les difficultés qui se posent dans les versions plus profondes de la paramétrisation intégrable, les questions de la manière de prendre la limite (séquentiellement ou toutes les couches à la fois), de la manière de mettre à l'échelle correctement les couches et leurs taux d'apprentissage pour obtenir des dynamiques non dégénérées, et de la manière de décrire les dynamiques résultantes sont d'un intérêt particulier. Nous verrons dans la Section 1.2.5 que le Tensor Program (Yang, 2019, 2020a,b; Yang and Hu, 2021)

permet de répondre de manière rigoureuse à ces questions.

Malgré les difficultés rencontrées pour les réseaux profonds, Nguyen and Pham (2020) et Sirignano and Spiliopoulos (2021) parviennent toujours à prouver des résultats de convergence globale pour les réseaux avec trois couches ou plus sous des ensembles spécifiques d'hypothèses. De plus, en plus des résultats de convergence de la dynamique de largeur finie vers une dynamique idéalisée à largeur infinie lorsque la largeur m tend vers l'infini, Fang et al. (2020), Nguyen and Pham (2020) et Araújo et al. (2019) fournissent des bornes quantitatives sur la distance entre la dynamique de largeur finie et sa contrepartie idéalisée à largeur infinie par rapport au nombre de neurones m, qui varie approximativement en $m^{-1/2}$.

Il est cependant clair d'après la littérature que le comportement des paramétrisations intégrables avec plus de quatre couches et une initialisation i.i.d. est dégénéré et que les gradients des différentes couches ont des magnitudes différentes par rapport à la largeur m. Par exemple, il est mentionné dans (Araújo et al., 2019) et (Nguyen and Pham, 2020) que sous une initialisation i.i.d. avec plus de quatre couches, les poids de différentes couches évoluent indépendamment des autres couches dans la limite de largeur infinie, et en outre, tous les poids de la même couche évoluent de la même quantité déterministe qui ne dépend que du temps. Bien que ces pièges soient clairement identifiés, le cadre et/ou les hypothèses sont ajustés (*e.g.*, initialisation non-i.i.d. , entraînement uniquement de certaines couches, nombre restreint de couches) afin de les contourner et d'établir une théorie de la limite de largeur infinie pour les réseaux profonds. L'objectif du Chapitre 2 de cette thèse est de traiter ces problèmes dans le cadre standard utilisé en pratique en adoptant une approche alternative à l'aide du Tensor Program.

Équations d'évolution dans l'espace des mesures

Dans cette section, nous passons en revue les outils mathématiques concernant les fonctionnelles sur les espaces de mesures et les flot de gradient de Wasserstein, car cela constitue une partie essentielle du travail présenté dans cette thèse, en particulier dans le Chapitre 3. Les notions que nous discutons ici sont présentées en détail dans (Ambrosio et al., 2005) et (Santambrogio, 2017, 2015).

Espaces des mesures de probabilité et distances de Wasserstein

Soit $q \geq 1$ un scalaire, et considérons l'espace $\mathcal{P}_q(\mathbb{R}^p)$ des mesures de probabilité sur $\mathbb{R}^p$ satisfaisant $\int ||x||^q \mathrm{d}\mu(x) < \infty$. On peut définir une distance sur $\mathcal{P}_q(\mathbb{R}^p)$, appelée la distance de Wasserstein-q, par

$$W_q(\mu, \nu) := \left(\min_{\gamma \in \Gamma(\mu,\nu)} \int ||x-y||^q \mathrm{d}\gamma(x,y) \right)^{1/q}$$

où, pour tout $\mu, \nu \in \mathcal{P}_q(\mathbb{R}^p)$, $\Gamma(\mu, \nu)$ est l'ensemble des *plans de transport* de μ à ν, c'est-à-dire l'ensemble des mesures de probabilité sur $\mathbb{R}^p \times \mathbb{R}^p$ dont les

marginales sont égales à μ et ν. Formellement,

$$\Gamma(\mu,\nu) = \{\gamma \in \mathcal{P}(\mathbb{R}^p \times \mathbb{R}^p) \,:\, \gamma_{\#}\pi_x = \mu, \gamma_{\#}\pi_y = \nu\}$$

où $\pi_x : (x,y) \in \mathbb{R}^p \times \mathbb{R}^p \mapsto x$ et $\pi_y : (x,y) \in \mathbb{R}^p \times \mathbb{R}^p \mapsto y$ sont les projections canoniques sur la première et la deuxième composante respectivement. La distance de Wasserstein (également connue sous le nom de distance de Monge-Kantorovich ou distance de Kantorovich-Rubinstein) provient de la théorie du transport optimal, dont l'objectif est de comprendre comment déplacer de manière optimale une masse d'une distribution à une autre selon un certain coût.

L'espace $\mathcal{P}_q(\mathbb{R}^p)$ muni de la distance W_q forme un espace métrique complet et convexe $(\mathcal{P}_q(\mathbb{R}^p), W_q)$ pour lequel la convergence selon la distance W_q est approximativement équivalente à la convergence faible des mesures (parfois aussi appelée convergence étroite), comme le montre le résultat suivant : pour toute séquence $(\mu_n)_{n\in\mathbb{N}}$ et μ dans $\mathcal{P}_q(\mathbb{R}^p)$, on a

$$W_q(\mu_n,\mu) \to 0 \text{ si et seulement si } \mu_n \rightharpoonup \mu \text{ et } \int ||x||^q \mathrm{d}\mu_n(x) \to \int ||x||^q \mathrm{d}\mu(x),$$

où $\mu_n \rightharpoonup \mu$ désigne la convergence faible des mesures, c'est-à-dire

$$\int \varphi \mathrm{d}\mu_n \to \int \varphi \mathrm{d}\mu$$

pour toute φ dans l'espace $\mathcal{C}_b(\mathbb{R}^p)$ des fonctions continues et bornées sur $\mathbb{R}^p$. Si l'on remplace l'ensemble entier $\mathbb{R}^p$ par un sous-ensemble compact $\Omega \subset \mathbb{R}^p$, l'énoncé d'équivalence ci-dessus reste vrai sans la condition sur la convergence de l'intégrale de la norme.

Remarquez que pour tout $q_2 \geq q_1 \geq 1$, l'inégalité de Jensen garantit que pour toute $\gamma \in \mathcal{P}(\mathbb{R}^p \times \mathbb{R}^p)$, $\left(\int ||x-y||^{q_1} \mathrm{d}\gamma(x,y)\right)^{1/q_1} \leq \left(\int ||x-y||^{q_2}\right)^{1/q_2} \mathrm{d}\gamma(x,y)$, ce qui implique que $W_{q_1}(\mu,\nu) \leq W_{q_2}(\mu,\nu)$. Dans cette thèse, nous nous concentrerons uniquement sur l'espace $(\mathcal{P}_2(\mathbb{R}^p), W_2)$.

Fonctionnelles des mesures de probabilité et première variation

Une fonctionnelle F sur $\mathcal{P}_2(\mathbb{R}^p)$ est une fonction $F : \mathcal{P}_2(\mathbb{R}^p) \to \mathbb{R}$. On aimerait définir une notion de dérivée sur $\mathcal{P}_2(\mathbb{R}^p)$ de manière similaire à la notion de dérivée ou de gradient en dimension finie, mais le problème est que $\mathcal{P}_2(\mathbb{R}^p)$ est un espace convexe de dimension infinie et non un espace euclidien. Par conséquent, il faut être prudent lors de la définition de cette notion, qui est un peu plus subtile dans ce contexte. Étant donné $\mu \in \mathcal{P}_2(\mathbb{R}^p)$, la *première variation* ou le *dérivé de Fréchet* de F en μ, s'il existe, est une fonction mesurable de $\mathbb{R}^p$ à $\mathbb{R}$, notée $\frac{\delta F}{\delta \mu}(\mu)$ ou simplement F'_μ, satisfaisant, pour toute perturbation appropriée ν,

$$\frac{d}{dt} F(\mu + t\nu)\Big|_{t=0} = \int \frac{\delta F}{\delta \mu}(\mu) \mathrm{d}\nu.$$

Les perturbations admissibles ν doivent satisfaire $\mu + t\nu \in \mathcal{P}_2(\mathbb{R}^p)$ pour t suffisamment petit, et sont donc choisies de la forme $\nu = \tilde{\nu} - \mu$, où $\tilde{\nu}$ est une mesure de probabilité avec une densité bornée et un support compact.

Notez que ν n'est pas une mesure de probabilité mais plutôt un élément de l'ensemble $\mathcal{M}(\mathbb{R}^p)$ des mesures signées, et en tant que différence de deux mesures de probabilité (de masse totale 1), elle satisfait $\int \mathrm{d}\nu = 0$, de sorte que la première variation est définie à une constante additive près, mais est unique modulo cette invariance.

Notez que la définition de la première variation est similaire à l'égalité satisfaite par le gradient en dimension finie : $\frac{d}{dt} f(x + ty)\Big|_{t=0} = \langle \nabla f(x), y \rangle$, et en tant que telle, l'intégrale $\int \frac{\delta F}{\delta \mu}(\mu) \mathrm{d}\nu$ peut être interprétée comme une sorte de produit intérieur (ou plutôt une notation de dualité) $\langle \frac{\delta F}{\delta \mu}(\mu), \nu \rangle$ qui représente l'action de la mesure ν sur la fonction mesurable $\frac{\delta F}{\delta \mu}(\mu)$.

Exemples classiques de fonctionnelles et de leurs premières variations. Étant donné $V : \mathbb{R}^p \to \mathbb{R}$, $W : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ et $f : \mathbb{R} \to \mathbb{R}$, on peut définir les fonctionnelles suivantes,

$$\begin{aligned}
\mathcal{V}(\mu) &= \int V(x) \mathrm{d}\mu(x), \\
\mathcal{W}(\mu) &= \int W(x, y) \mathrm{d}\mu(x) \mathrm{d}\mu(y), \\
\mathcal{F}(\mu) &= \begin{cases} \int f\left(\frac{d\mu}{d\lambda}\right) \mathrm{d}\lambda(x) \text{ si } \mu \in L^1(\lambda) \\ +\infty \text{ sinon} \end{cases},
\end{aligned}$$

où λ désigne la mesure de Lebesgue sur $\mathbb{R}^p$, et il est facile de vérifier que leurs premières variations respectives sont

$$\begin{aligned}
\frac{\delta \mathcal{V}}{\delta \mu}(\mu)(x) &= V(x), \\
\frac{\delta \mathcal{W}}{\delta \mu}(\mu)(x) &= \int W(x, y) \mathrm{d}\mu(y) + \int W(y, x) \mathrm{d}\mu(y), \\
\frac{\delta \mathcal{F}}{\delta \mu}(\mu)(x) &= f'\left(\frac{d\mu}{d\lambda}(x)\right) \text{ pour } \mu \in L^1(\lambda).
\end{aligned}$$

Première variation de l'objectif pour les réseaux à deux couches de largeur infinie. Dans le cas de la fonctionnelle objectif F définie dans l'équation (1.4), il est facile de déduire que pour n'importe quel $w = (w^1, w^2) \in \mathbb{R}^{d+1}$,

$$F'_\mu(w) = \frac{\delta F}{\delta \mu}(\mu)(w) = \int_x \partial_2 \ell(f^*(x), f(\mu; x)) w^2 \sigma(x^\top w^1) \mathrm{d}\rho(x).$$

Première variation de la distance de Wasserstein-2. La définition de la distance de Wasserstein-2 peut être vue comme un problème de minimisation contraint dans un espace de dimension infinie, où la contrainte est que la mesure de probabilité $\gamma \in \mathcal{P}(\mathbb{R}^p \times \mathbb{R}^p)$ doit avoir des marginales égales à μ et ν. Comme la dualité de Lagrange permet de traiter les problèmes d'optimisation contraints en dimension finie, Kantorovich a développé une théorie, appelée dualité de Kantorovich, qui permet de traiter les problèmes d'optimisation contraints dans l'espace des mesures, et en particulier les problèmes de transport optimal. Le problème dual associé au problème de transport optimal avec les distances de Wasserstein se lit comme suit :

$$\max_{\varphi,\psi\in\Lambda} \int \varphi \mathrm{d}\mu + \int \psi \mathrm{d}\nu - \int \Big(\varphi(x) + \psi(y)\Big) \mathrm{d}\gamma(x,y),$$
$$\Lambda := \Big\{\varphi, \psi \in \mathcal{C}_b(\mathbb{R}^p) \,:\, \varphi(x) + \psi(y) \leq ||x-y||^q\Big\}.$$

Des descriptions alternatives du problème dual, qui permettent de l'étudier en profondeur, sont assez complexes et nécessiteraient l'introduction de nouvelles notations et concepts. Ce n'est pas l'objet de cette thèse, et nous renvoyons à (Santambrogio, 2017)[Section 4.1] pour une présentation détaillée. Cependant, nous notons que cette dualité permet de déduire la forme de la solution du problème de transport optimal définissant les distances de Wasserstein. En particulier, il peut être démontré (voir Santambrogio, 2017[Théorème 4.2]) que si μ est absolument continue, il existe une application $T : \mathbb{R}^p \to \mathbb{R}^p$ appelée une *application de transport optimal*, et une fonction $\varphi : \mathbb{R}^p \to \mathbb{R}$, appelée un *potentiel de Kantorovich* (provenant de la dualité de Kantorovich), satisfaisant les trois conditions suivantes:

(i) $\frac{\delta W_2}{\delta \mu}(\mu, \nu) = \varphi$,

(ii) la mesure image $\gamma^* := (\mathrm{id}, T)_{\#}\mu$ réalise le minimum dans la définition de W_2,

(iii) $\nabla\varphi = \mathrm{id} - T$.

Flux de gradient de Wasserstein et conditions d'optimalité

Nous nous tournons maintenant vers les flot de gradient de Wasserstein, qui sont l'outil théorique principal pour optimiser les fonctionnelles sur $\mathcal{P}_2(\mathbb{R}^p)$. Considérons une fonctionnelle $F : \mathcal{P}_2(\mathbb{R}^p) \to \mathbb{R}$ qui admet une première variation en chaque $\mu \in \mathcal{P}_2(\mathbb{R}^p)$ et qui est différentiable presque partout. À partir d'une mesure initiale $\mu_0 \in \mathcal{P}_2(\mathbb{R}^p)$, le flot de gradient de Wasserstein de la fonctionnelle F est un chemin $(\mu_t)_{t\geq 0}$ dans l'espace $\mathcal{P}_2(\mathbb{R}^p)$ satisfaisant, *au sens des distributions*, l'équation de continuité suivante :

$$\partial_t \mu_t = -\mathrm{div}\left(-\nabla\left(\frac{\delta F}{\delta \mu}(\mu_t)\right)\mu_t\right). \tag{1.16}$$

Une paire $(\mu_t, v_t)_{t\geq 0}$ constituée d'un chemin dans $\mathcal{P}_2(\mathbb{R}^p)$ et d'un champ de vecteurs dépendant du temps $v_t : \mathbb{R}^p \to \mathbb{R}^p$ satisfait l'équation de continuité $\partial_t \mu_t = -\text{div}(v_t \mu_t)$ au sens des distributions si, pour toute fonction de test φ dans l'espace $\mathcal{C}_c^1(\mathbb{R}^p)$ des fonctions continûment différentiables à support compact, on a :

$$\frac{d}{dt}\int \varphi \mathrm{d}\mu_t = \int \nabla\varphi^\top v_t \mathrm{d}\mu_t.$$

En particulier, en choisissant φ comme la fonction constante égale à 1, cela montre que la masse totale est conservée dans le flot de gradient de Wasserstein : la masse n'est ni injectée ni perdue le long du flot, mais simplement déplacée.

Schéma de minimisation du mouvement. D'où vient cette équation ? Dans des espaces métriques tels que l'espace des mesures de probabilité, il est difficile de définir une notion de dérivée en raison de l'absence de structure linéaire, et on a généralement recours aux *schémas de minimisation du mouvement*. Dans $\mathbb{R}^p$ (ou tout espace de Hilbert), soit $\tau > 0$ un paramètre et considérons une fonction différentiable $f : \mathbb{R}^p \to \mathbb{R}$, ainsi qu'une séquence $(x_k)_{k\geq 0}$ dans $\mathbb{R}^p$ satisfaisant, pour tout k,

$$x_{k+1} \in \operatorname*{argmin}_y f(y) + \frac{1}{2\tau}||y - x_k||^2.$$

Ceci est appelé un schéma de minimisation du mouvement : en effet, on tente de minimiser f tout en restant proche de l'estimation actuelle x_k. La séquence d'estimations satisfait $\nabla f(x_{x+1}) = -\frac{x_{k+1}-x_k}{\tau}$, et en considérant une fonction $\tilde{x} : \mathbb{R}_+ \to \mathbb{R}^p$ interpolant les x_k (c'est-à-dire, $\tilde{x}(k\tau) = x_k$), on a $\frac{\tilde{x}(k\tau+\tau)-\tilde{x}(k\tau)}{\tau} = -\nabla f(\tilde{x}((k+1)\tau)$ et lorsque $\tau \to 0^+$, on obtient une courbe satisfaisant $\tilde{x}'(t) = -\nabla f(\tilde{x}(t))$, qui est exactement le flux de gradient de la fonction f (le schéma de minimisation du mouvement est en fait le schéma d'Euler implicite pour discrétiser le flux de gradient). La convergence de la fonction d'interpolation vers un flux de gradient lorsque $\tau \to 0^+$ peut être rendue rigoureuse si f est continûment différentiable ou si elle est convexe (voir Santambrogio, 2017[Proposition 2.3]).

Revenant à l'espace des mesures, on peut essayer de dériver un schéma de minimisation du mouvement similaire comme suit :

$$\mu_{k+1} \in \operatorname*{argmin}_\mu F(\mu) + \frac{1}{2\tau} W_2(\mu, \mu_k)^2. \tag{1.17}$$

Pour comprendre les conditions que μ_{k+1} doit satisfaire, nous devons d'abord expliquer quelles sont les conditions d'optimalité pour les fonctionnelles sur $\mathcal{P}_2(\mathbb{R}^p)$. De manière similaire au cas de dimension finie, si l'on souhaite minimiser $G(\mu)$ pour une certaine fonctionnelle G admettant une première variation en chaque μ, alors l'optimalité d'un certain minimiseur μ^* est liée à la valeur de la première variation $\frac{\delta G}{\delta \mu}(\mu^*)$ de G en μ^*. Il est affirmé dans (Santambrogio, 2015)[Proposition 7.20]

que, sous les hypothèses de régularité appropriées, pour un minimiseur μ^* de G, la première variation $\frac{\delta G}{\delta \mu}(\mu^*)$ doit être constante sur le support de μ^*. Pour le schéma de minimisation du mouvement (1.8) ci-dessus, la première variation de la distance quadratique de Wasserstein 2 est liée au transport optimal de μ_{k+1} à μ_k, et la condition d'optimalité se traduit par

$$\frac{\delta F}{\delta \mu}(\mu_{k+1}) + \frac{1}{\tau}\varphi_\tau = C$$

pour une certaine constante C, où φ_τ est le potentiel de Kantorovich associé au transport de μ_{k+1} à μ_k. La théorie du transport optimal nous dit que $\nabla\varphi_\tau = x - T_\tau(x)$ pour l'application de transport optimal T_τ de μ_{k+1} à μ_k, de sorte que, en prenant le gradient de l'équation ci-dessus, on obtient ce qui suit :

$$\nabla\left(\frac{\delta F}{\delta \mu}(\mu_{k+1})\right)(x) = -\frac{T_\tau(x) - x}{\tau}.$$

Par conséquent, si nous souhaitons transporter de la masse de μ_k pour minimiser la quantité dans (1.8), le déplacement de masse $\frac{T_\tau(x)-x}{\tau}$ en tout point x doit être égal au champ de vecteurs $v_{k+1}(x) := -\nabla\left(\frac{\delta F}{\delta \mu}(\mu_{k+1})\right)(x)$. Lorsque $\tau \to 0^+$, il s'ensuit que le changement de masse induit par le schéma de minimisation itéré ci-dessus doit satisfaire l'équation de flux de gradient de Wasserstein (1.7) : à tout moment t, la masse située en x est déplacée avec une vitesse $v_t(x) = -\nabla\left(\frac{\delta F}{\delta \mu}(\mu_t)\right)(x)$. La preuve de la convergence du schéma de minimisation itéré ci-dessus est technique dans des espaces métriques généraux et est décrite dans (Santambrogio, 2015, 2017).

Propriétés du flot de gradient de Wasserstein. L'existence d'une solution au problème de flot de gradient (ou de l'équation de continuité) est garantie lorsque suffisamment de régularité est supposée sur la fonctionnelle F (et sur le gradient de sa première variation $\nabla\frac{\delta F}{\delta \mu}$, qui est la vitesse négative dans l'équation de continuité). En ce qui concerne l'unicité, elle nécessite souvent une certaine notion de convexité (*e.g.*, semi-convexité géodésique) sur F ou certaines hypothèses sur la mesure initiale (*e.g.*, qu'elle ait une densité par rapport à la mesure de Lebesgue) pour être garantie en général. Pour les réseaux à deux couches de largeur infinie, (Chizat and Bach, 2018; Mei et al., 2018; Wojtowytsch, 2020) montrent l'existence et l'unicité du WGF (1.7) sous des hypothèses faibles. Cependant, les hypothèses de régularité ne sont pas satisfaites par ReLU, ce qui doit être traité séparément. Nous discutons de cela dans le paragraphe suivant.

En ce qui concerne les flots de gradient en dimension finie, il peut être démontré que le flot de gradient de Wasserstein diminue toujours la fonctionnelle que l'on cherche à minimiser, comme le montre la relation suivante :

$$\frac{d}{dt}F(\mu_t) = -\int \left|\left|\nabla\left(\frac{\delta F}{\delta \mu}(\mu_t)\right)\right|\right|^2 \mathrm{d}\mu_t \leq 0.$$

Équivalence entre le WGF pour les mesures atomiques et le GF de largeur finie

Nous détaillons ici la dérivation de la relation entre le WGF (1.7) de l'objectif en dimension infinie F sur l'espace $\mathcal{P}_2(\mathbb{R}^{d+1})$ et le flot de gradient de l'objectif F_m sur les poids d'un réseau à deux couches de largeur m. Tout d'abord, nous dérivons la description du flot de l'équation de continuité, puis nous procédons à montrer l'équivalence entre le WGF pour les mesures atomiques et le flot de gradient de largeur finie.

Description du flot de l'équation de continuité. Soit $(\mu_t, v_t)_{t\geq 0}$ un couple formé d'une trajectoire dans l'espace $\mathcal{P}_2(\mathbb{R}^p)$ et d'un champ de vecteurs dépendant du temps $v_t : \mathbb{R}^p \to \mathbb{R}^p$, satisfaisant, au sens des distributions, l'équation de continuité

$$\partial_t \mu_t = -\mathsf{div}(v_t \mu_t),$$

et considérons le flot défini, pour tout $w \in \mathbb{R}^d$, par l'ODE

$$\begin{aligned} X_0(w) &= w, \\ \frac{d}{dt} X_t(w) &= v_t\left(X_t(w)\right). \end{aligned}$$

Alors, il est vrai que pour tout $t \geq 0$, $\mu_t = (X_t)_{\#}\mu_0$. En effet, en définissant $\nu_t := (X_t)_{\#}\mu_0$, puisque X_0 est l'application identité de $\mathbb{R}^d$, on a $\nu_0 = \mu_0$, et l'unicité de la solution de l'équation de continuité suffira pour conclure à l'égalité. Soit $t \geq 0$, et $\varphi \in \mathcal{C}_c^1(\mathbb{R}^p)$. Nous avons

$$\begin{aligned} \frac{d}{dt}\int \varphi \mathrm{d}\nu_t &= \frac{d}{dt}\int \varphi \circ X_t \mathrm{d}\mu_0 \\ &= \int \left\langle (\nabla\varphi)\circ X_t, \frac{d}{dt}X_t \right\rangle \mathrm{d}\mu_0 \\ &= \int \langle (\nabla\varphi)\circ X_t, v_t \circ X_t\rangle \,\mathrm{d}\mu_0 \\ &= \int \nabla\varphi^\top v_t \mathrm{d}\nu_t, \end{aligned}$$

ce qui signifie que ν_t satisfait l'équation de continuité au sens des distributions avec la condition initiale $\nu_0 = \mu_0$. L'unicité d'une telle solution permet de conclure que $\nu_t = \mu_t$ pour tout $t \geq 0$.

Équivalence entre le WGF et le GF. Appelons $\phi : \mathbb{R}^{d+1} \times \mathbb{R}^d \to \mathbb{R}$ défini, pour tout $w = (w^1, w^2) \in \mathbb{R}^d \times \mathbb{R}$ par $\phi(w; x) = w^2\sigma(x^\top w^1)$. Pour tout $m \in \mathbb{N}$, notons f_m la paramétrisation intégrable d'un réseau à deux couches de largeur m, définie par $f_m(\theta; x) = \frac{1}{m}\sum_{j=1}^m \phi(\theta_j; x)$ où $\theta_j = (w_j^1, w_j^2) \in \mathbb{R}^{d+1}$. De plus,

définissons F_m comme l'objectif de largeur finie défini, pour tout $\theta \in (\mathbb{R}^{d+1})^m$, par $F_m(\theta) = \mathbb{E}_\rho\left[\ell(f^*(x), f_m(\theta; x))\right]$, et soit F l'objectif en dimension infinie sur les mesures, défini, pour tout $\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})$, par $F(\mu) = \mathbb{E}_\rho\left[\ell(f^*(x), f(\mu; x))\right]$ où $f(\mu; x) = \int \phi(w; x)\mathrm{d}\mu(w)$.

Tout d'abord, remarquons qu'en définissant la mesure atomique suivante: $\mu_m = \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j}$, il est vrai que $f(\mu_m; \cdot) = f_m(\theta; \cdot)$, et par conséquent $F(\mu_m) = F_m(\theta)$. Ensuite, remarquons que parce que la première variation de F en μ est donnée, pour tout $w \in \mathbb{R}^{d+1}$, par $F'_\mu(w) = \int_x \partial_2\ell(f^*(x), f(\mu; x))\phi(w; x)\mathrm{d}\rho(x)$, il en découle que son gradient est donné par la formule suivante: $\nabla F'_\mu(w) = \int_x \partial_2\ell(f^*(x), f(\mu; x))\nabla_w\phi(w; x)\mathrm{d}\rho(x)$. D'où l'égalité $m\nabla_{\theta_j}F_m(\theta) = \nabla F'_{\mu_m}(\theta_j)$.

Considérons les poids initiaux $((w_1^1(0), w_1^2(0)), \ldots, (w_m^1(0), w_m^2(0)))$, ainsi que la mesure atomique initiale $\mu_{m,0} = \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j(0)}$ où $\theta_j(0) = (w_j^1(0), w_j^2(0))$. Soit $(\mu_{m,t})_{t\geq 0}$ le WGF de l'objectif F à partir de $\mu_{m,0}$, et soit X_t le flot associé à l'équation de continuité avec le champ de vecteurs $v_t = -\nabla F'_{\mu_{m,t}}$ comme dans le paragraphe précédent. Il est vrai que $\mu_{m,t} = (X_t)_\# \mu_{m,0}$ et comme une image de la mesure atomique $\mu_{m,0}$, $\mu_{m,t}$ est également une mesure atomique et ses masses sont situées aux images des masses de $\mu_{m,0}$ par la map de poussée en avant, c'est-à-dire $\mu_{m,t} = \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j(t)}$ avec $\theta_j(t) := X_t(\theta_j(0))$. Montrer que $\theta(t) = (\theta_j(t))_{j\in[1,m]}$ est un flot de gradient pour F_m découle facilement de l'ODE satisfaite par le flot X_t :

$$\begin{aligned}
\frac{d}{dt}\theta_j(t) &= \frac{d}{dt}X_t(\theta_j(0)) \\
&= -\nabla F'_{\mu_{m,t}}\Big(X_t(\theta_j(0))\Big) \\
&= -m\nabla_{\theta_j}F_m(\theta_j(t)).
\end{aligned}$$

Inversement, définissons $(\theta(t))_{t\geq 0}$ comme le flot de gradient de F_m à partir de $\theta(0) = (\theta_j(0))_{j\in[1,m]}$, c'est-à-dire $\frac{d}{dt}\theta(t) = -m\nabla F_m(\theta(t))$, et définissons $\mu_{m,t} := \frac{1}{m}\sum_{j=1}^m \delta_{\theta_j(t)}$. Ensuite, montrer que $\mu_{m,t}$ est un WGF pour F découle facilement de l'ODE satisfaite par $\theta(t)$. En effet, soit $\varphi \in \mathcal{C}_c^1(\mathbb{R}^{d+1})$. On a :

$$\begin{aligned}
\frac{d}{dt}\int \varphi\mathrm{d}\mu_{m,t} &= \frac{d}{dt}\left(\frac{1}{m}\sum_{j=1}^m \varphi(\theta_j(t))\right) \\
&= \frac{1}{m}\sum_{j=1}^m \nabla\varphi(\theta_j(t))^\top\left(\frac{d}{dt}\theta_j(t)\right) \\
&= \frac{1}{m}\sum_{j=1}^m \nabla\varphi(\theta_j(t))^\top\left(-m\nabla_{\theta_j}F_m(\theta_j(t))\right) \\
&= \frac{1}{m}\sum_{j=1}^m \nabla\varphi(\theta_j(t))^\top\left(-\nabla F'_{\mu_{m,t}}(\theta_j(t))\right) \\
&= \int \nabla\varphi^\top\left(-\nabla F'_{\mu_{m,t}}\right)\mathrm{d}\mu_{m,t}
\end{aligned}$$

ce qui montre que $\mu_{m,t}$ satisfait l'équation de continuité au sens des distributions avec le champ de vecteurs $v_t = -\nabla F'_{\mu_{m,t}}$, c'est-à-dire que $(\mu_{m,t})_{t\geq 0}$ est le WGF de F à partir de $\mu_{m,0}$.

Homogénéité et réduction aux mesures sur la sphère

Les activations positivement homogènes sont très courantes dans la littérature sur les réseaux neuronaux, en particulier dans le cadre des études théoriques, car elles conduisent souvent à des simplifications. L'activation ReLU (rectified linear unit), définie par $\sigma(z) = \max(0, z)$, est un exemple courant qui est omniprésent tant en théorie qu'en pratique. Cependant, elle peut également entraîner des difficultés techniques en raison de sa non-différentiabilité et de la non-continuité de sa dérivée $\mathbb{1}_{z>0}$ en 0. En particulier, l'existence du WGF (1.5) ne peut pas être garantie en général lorsque l'on utilise ReLU comme fonction d'activation. Néanmoins, il est possible de contourner ce problème technique grâce à l'homogénéité positive de ReLU et à des hypothèses spécifiques sur la distribution initiale μ_0. Il est démontré dans (Wojtowytsch, 2020) et (Chizat and Bach, 2020) que lorsque la mesure initiale $\mu_0 \in \mathcal{P}_2(\mathbb{R}^{d+1})$ est supportée sur le cône $\{(w^1, w^2) \in \mathbb{R}^{d+1} : ||w^1|| = |w^2|\}$, le WGF (1.5) est bien défini avec une activation ReLU. De plus, il est démontré qu'avec cette initialisation, la mesure μ_t reste supportée sur le cône à tout moment t.

Réduction aux mesures signées sur la sphère. L'homogénéité positive de ReLU permet également d'adopter un point de vue alternatif pour le WGF (1.5). Pour toute mesure $\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})$, on peut définir une paire de mesures non négatives supportées sur la sphère $\nu_+, \nu_- \in \mathcal{M}_+(\mathbb{S}^{d-1})$ grâce à la caractérisation suivante, particulièrement adaptée à l'homogénéité des réseaux à deux couches avec une activation ReLU : pour toute fonction de test continue $\varphi : \mathbb{S}^{d-1} \to \mathbb{R}$, il doit être vérifié que

$$\int \varphi \mathrm{d}\nu^{\pm} = \int_{\pm w^2 \geq 0, w^1} |w^2| \, ||w^1|| \varphi\left(\frac{w^1}{||w^1||}\right) \mathrm{d}\mu(w).$$

Essentiellement, cela peut être compris comme une forme de projection qui élimine la redondance induite par la propriété d'homogénéité entre la norme des poids de la première couche et la magnitude des poids de la couche de sortie. Avec cette définition, la fonction du réseau peut être exprimée comme

$$f(\mu; x) = \int w^1 \sigma(x^\top w^2) \mathrm{d}\mu(w) = \int \sigma(x^\top u) \mathrm{d}\nu(u),$$

avec $\nu = \nu^+ - \nu^- \in \mathcal{M}(\mathbb{S}^{d-1})$ une mesure signée sur la sphère. De ce point de vue, les neurones de la première couche sont considérés comme des directions sur la sphère, tandis que les poids de la deuxième couche sont considérés comme des masses (signées) pesant sur ces directions. La masse dans cette paramétrisation

prend en compte à la fois les poids de la deuxième couche et la norme des poids de la première couche dans la paramétrisation d'origine de l'Équation (1.3). De ce point de vue, le problème consiste à apprendre un réseau à deux couches de largeur infinie en vue d'apprendre les positions et les masses des neurones de la première couche. La masse totale de ν mesurée par la norme de variation totale est donnée par $|\nu|(\mathbb{S}^{d-1}) = \int \mathrm{d}(\nu^+ + \nu^-) = \int ||w^1||\,|w^2|\mathrm{d}\mu(w)$.

Considérez le WGF (1.5) avec une activation ReLU, avec μ_0 supportée sur le cône. Ensuite, en définissant $\nu_t^\pm$ à partir de μ_t comme indiqué ci-dessus, le fait que μ_t soit supportée sur le cône à n'importe quel instant permet de dériver des équations d'évolution pour les mesures $\nu_t^\pm$. Nous soulignons que *ceci n'est pas possible* en général (même si l'on suppose seulement l'homogénéité). La paire (ν_t^+, ν_t^-) satisfait les équations suivantes, connues sous le nom d'équations d'advection-réaction (ou de gradient de Wasserstein-Fisher-Rao Gallouët et al., 2019), au sens des distributions :

$$\partial_t \nu_t^\pm = -\mathrm{div}(\pm \tilde{v}_t \nu_t^\pm) \pm 2 g_t \nu_t^\pm, \tag{1.18}$$

avec $g_t(u) = F'_{\mu_t}(u, 1)$ et $\tilde{v}_t(u) = -\mathrm{proj}_{\{u\}^\perp}(\nabla g_t(u))$ pour $u \in \mathbb{S}^{d-1}$. Autrement dit, pour toute fonction de test $\varphi \in \mathcal{C}_c^1(\mathbb{R}^d)$, on a

$$\frac{d}{dt}\int \varphi \mathrm{d}\nu_t^\pm = \pm \int \tilde{v}_t^\top \nabla\varphi \mathrm{d}\nu_t^\pm \pm 2\int \varphi g_t \mathrm{d}\nu_t^\pm.$$

La dérivation de cette équation découle de l'équation de continuité satisfaite par μ_t, de la définition de $\nu_t^\pm$ à partir de μ_t en utilisant l'homogénéité, et du fait que $|w^2| = ||w^1||$ sur le support de μ_t. Voir l'Annexe B.5.1 pour plus de détails sur la dérivation de l'équation de Wasserstein-Fisher-Rao. En d'autres termes, la mesure signée $\nu_t = \nu_t^+ - \nu_t^-$ satisfait l'équation

$$\partial_t \nu_t = -\mathrm{div}(\tilde{v}_t(\nu_t^+ + \nu_t^-)) + 2 g_t(\nu_t^+ + \nu_t^-)$$

où ν_t^+ et ν_t^- représentent respectivement la partie positive et la partie négative de ν_t. Ce point de vue est largement utilisé dans le chapitre 3. Dans cette perspective, la masse n'est pas préservée et le changement de masse est gouverné par le terme de réaction g_t, tandis que l'advection (déplacement) de la masse est gouvernée par le champ de vecteurs $\tilde{v}_t$, qui est tangent à la sphère. En particulier, la masse totale $|\nu_t|(\mathbb{S}^{d-1})$ évolue selon $\frac{d}{dt}|\nu_t|(\mathbb{S}^{d-1}) = \int g_t \mathrm{d}\nu_t$.

Tensor Program et limites à largeur infinie de n'importe quelle paramétrisation

Le Tensor Program est un cadre développé dans une série de travaux (Yang, 2019, 2020a,b; Yang and Hu, 2021; Yang et al., 2022) dans le but de mieux comprendre et de décrire rigoureusement la limite à largeur infinie de diverses paramétrisations (comme introduit dans la Section 1.2.1) de réseaux neuronaux. L'objectif est de comprendre précisément l'ampleur des quantités impliquées dans

les passes avant et arrière d'un réseau neuronal lorsque $m \to \infty$. Ce faisant, l'un des obstacles particuliers est de comprendre comment les différentes quantités sont corrélées entre elles.

Les idées et techniques développées dans la série du Tensor Program trouvent leur origine dans la littérature sur la physique statistique (Bayati and Montanari, 2011; Bolthausen, 2014), où elles ont émergé pour décrire le comportement d'algorithmes (comme la propagation de messages) impliquant de grandes matrices aléatoires et des non-linéarités en utilisant la technique de conditionnement gaussien. Le bénéfice supplémentaire du Tensor Program est de fournir un formalisme pour appliquer systématiquement ces techniques dans le contexte des réseaux neuronaux. Nous utilisons abondamment le Tensor Program dans les démonstrations de la plupart des résultats présentés dans le chapitre 2.

Le premier travail de la série du Tensor Program (Yang, 2019) est consacré à la compréhension du type de fonction calculé par les réseaux neuronaux profonds *à l'initialisation* avec des matrices gaussiennes i.i.d. dont l'écart-type évolue comme $m^{-1/2}$. Alors que la réponse est connue pour les réseaux peu profonds entièrement connectés depuis Neal (1995), plusieurs travaux récents (Lee et al., 2017; Matthews et al., 2018; Novak et al., 2018; Garriga-Alonso et al., 2018) ont généralisé ce résultat à des réseaux plus profonds ou à une architecture convolutionnelle. La première version du Tensor Program dans (Yang, 2019) fournit des outils mathématiques pour démontrer systématiquement que les réseaux neuronaux de *n'importe quelle architecture* se comportent comme des processus gaussiens à l'initialisation dans la limite de largeur infinie.

La deuxième version du Tensor Program (Yang, 2020a) étend l'analyse à la première passe en arrière (les gradients à l'initialisation) et prouve que le noyau de la tangente neurale $\nabla f(\theta; x)^\top \nabla f(\theta; y)$ (voir la Section 1.2.2) converge presque sûrement, lorsque $m \to \infty$, vers une limite déterministe à l'initialisation pour n'importe quelle architecture dans la paramétrisation NTK.

La troisième version du Tensor Program (Yang, 2020b) se concentre sur l'extension des outils mathématiques précédemment développés pour couvrir les passes avant et arrière à n'importe quel pas de temps. Une étape cruciale est la capacité à décrire la limite des quantités où à la fois une matrice de poids W^l et sa transposée $(W^l)^\top$ sont impliquées, et à gérer les corrélations potentielles qui pourraient en résulter.

Enfin, Yang and Hu (2021) utilisent le cadre du Tensor Program pour catégoriser différents types de paramétrisations dans la limite à largeur infinie. Cette catégorisation spécifie si une paramétrisation abc (voir Section 1.2.1 et Yang and Hu, 2021) se trouve dans le *régime du noyau* ou dans le *régime d'apprentissage des caractéristiques* en fonction des valeurs des exposants a_l, b_l et c_l. De plus, une nouvelle paramétrisation appelée μP est proposée, correspondant aux valeurs suivantes pour les exposants : $a_1 = -1/2$, $a_l = 0$ pour $l \in [2, L]$ et $a_{L+1} = 1/2$, $b_l = 1/2$, pour $l \in [1, L+1]$, et $c_l = 0$ pour $l \in [1, L+1]$. De manière équiva-

lente, les exposants pour cette paramétrisation peuvent également être donnés par : $a_1 = 0$,

$a_l = 1/2$ pour $l \in [2, L]$ et $a_{L+1} = 1$, $b_l = 0$ pour $l \in [1, L+1]$, $c_l = -1$ pour $l \in [1, L+1]$. Il s'agit de l'extension appropriée des modèles "mean-field" pour plus de deux couches (ils sont identiques pour les réseaux à deux couches), et elle "maximise" l'apprentissage dans toutes les couches (d'une manière précisée dans Yang and Hu, 2021). Cependant, l'analyse de Yang and Hu (2021) exclut toute paramétrisation pour laquelle les (pré-)activations pourraient disparaître à l'initialisation lorsque $m \to \infty$, ce qui est le cas des IP avec trois couches ou plus. D'où la nécessité d'un traitement spécial que nous présentons dans le chapitre 2.

Intuition derrière la technique

Nous présentons brièvement ici l'intuition derrière le Tensor Program ainsi que son formalisme et les principaux résultats qui y sont associés. Nous commençons par décrire la situation lors de la passe avant à l'initialisation, où les choses sont plus faciles à comprendre, puis nous passons à la description des calculs impliqués dans la première passe en arrière, pour enfin expliquer comment traiter les calculs généraux dans les passes avant et en arrière ultérieures.

Première passe avant. L'élément clé à étudier ici est le comportement de sommes de type $m^{-1/2} \sum_{j=1}^{m} w_j x_j$ pour de grandes valeurs de m, où $w \in \mathbb{R}^m$ est un vecteur gaussien avec des entrées i.i.d. suivant une loi $\mathcal{N}(0,1)$ et $x \in \mathbb{R}^m$ est un vecteur aléatoire indépendant de w. Lorsque les entrées de x sont i.i.d., le théorème central limite garantit que cette quantité converge en loi vers une variable gaussienne lorsque $m \to \infty$. En fait, ce résultat est également valable dès que $||x||^2/m$ converge presque sûrement vers une limite σ_∞^2 (voir Yang, 2019[Proposition G.4]). La situation est plus complexe lorsque x et w sont corrélés, et nous abordons ce cas plus tard (il est traité dans la troisième version du Tensor Program Yang, 2020b). Il en découle facilement que les entrées des pré-activations $h^l = m^{-1/2} w^l x^{l-1}$ d'un réseau dans la paramétrisation NTK deviennent gaussiennes lorsque $m \to \infty$. La convergence de $||x^{l-1}||^2/m$ est due au fait que les entrées tendent à être approximativement indépendantes dans la limite, car les différentes lignes (qui sont i.i.d.) de la matrice gaussienne w^{l-1} sont utilisées pour calculer les différentes entrées. Il est donc clair qu'indépendamment de la fonction d'activation σ et de la topologie de l'architecture du réseau, la sortie du réseau tend à être gaussienne dans la limite de grande valeur de m dès que l'initialisation gaussienne est adaptée.

Dans le cadre de (Yang and Hu, 2021), tout autre choix de facteurs d'échelle pour les poids (le a dans les paramétrisations abc) conduira soit à la disparition, soit à l'explosion de la passe avant.

Première passe en arrière. Lors de l'étude de la passe en arrière à l'initialisation, la quantité clé est le gradient de la sortie du réseau par rapport aux activations x^l, c'est-à-dire $\nabla_{x^l} f(\theta; x) = (m^{-1/2} w^{l+1})^\top \nabla_{h^{l+1}} f(\theta; x)$. Essentiellement, la situation est la même que lors de la première passe avant : étant donné que $\nabla_{h^{l+1}} f(\theta; x)$ est calculé à l'aide de matrices w^k pour $k \geq l+2$, il est indépendant de w^{l+1}. La différence réside simplement dans l'utilisation de la transposée des matrices initiales dans les multiplications, mais comme celles-ci ont des entrées initialisées i.i.d., la même logique que dans la passe avant s'applique. Comme $\nabla_{x^L} f(\theta; x) = m^{-1/2} w^{L+1}$, ses entrées sont de l'ordre de $m^{-1/2}$ et ce facteur se propage aux gradients de toutes les couches grâce aux équations de rétropropagation. Cela se traduit par le fait que $\nabla_{w^l} f(\theta; x)$ est de l'ordre de m^{-1}, ce qui est nettement plus petit que la magnitude initiale, entraînant une linéarisation, comme discuté dans la Section 1.2.2, si les taux d'apprentissage ne sont pas adaptés. La magnitude des gradients lors de la première passe en arrière est donc bien comprise.

Calculs généraux dans les étapes suivantes. La correction des échelles de gradient en utilisant un taux d'apprentissage de $m^{1/2}$ pour les couches intermédiaires $l \in [2, L]$ résout le problème décrit ci-dessus pour les gradients à l'initialisation. Avec cette correction, les entrées des mises à jour des poids $\Delta W^l = W^l(1) - W^l(0)$ sont de l'ordre de m^{-1}, tandis que celles de $W^l(0)$ sont de l'ordre de $m^{-1/2}$. C'est essentiellement ce que fait μP, sauf qu'il corrige également l'échelle de la couche de sortie de manière à ce que les poids soient de l'ordre de m^{-1} afin d'empêcher la sortie du réseau de diverger après l'initialisation.

Cela soulève la question de savoir en quoi cela diffère du comportement du NTK puisque la magnitude des mises à jour est toujours beaucoup plus petite que celle de l'initialisation. La réponse est subtile, et il faut étudier la passe avant suivante à l'étape $t = 1$ pour comprendre pourquoi ces magnitudes sont correctes. En bref, la raison en est que bien que $W^l(0)$ et ΔW^l aient des magnitudes différentes par rapport à m, $W^l(0) x_1^{l-1}$ et $\Delta W^l x_1^{l-1}$ sont tous deux de même ordre (c'est-à-dire de l'ordre de 1) par rapport à m en raison des non-linéarités et des corrélations impliquées dans le second terme. Ainsi, il n'y a pas d'effet de linéarisation ici.

En effet, avec la correction d'échelle induite par les taux d'apprentissage décrits ci-dessus, la contribution de la mise à jour des poids aux pré-activations h_1^l à la couche l et à l'étape $t = 1$ s'exprime comme suit : $\frac{(x_1^{l-1})^\top x_0^{l-1}}{m} \nabla_{h^l} f(\theta(0); x_0)$. Bien que l'échelle ne soit pas en $m^{-1/2}$ ici comme à l'initialisation, il est clair que les calculs impliqués sont d'une nature différente : les deux vecteurs multipliés dans le produit scalaire n'ont aucune raison d'avoir des coordonnées gaussiennes, et en outre, ils ne sont *pas indépendants*, car la matrice gaussienne w^{l-1} est utilisée pour calculer les deux termes. Il s'agit de l'un des résultats les plus importants de la série du Tensor Program, résumé dans un théorème principal (voir Yang, 2020b[Théorème 2.10], Yang and Hu, 2021[Théorème 7.4]), qui prouve que les produits scalaires de ce type, mis à l'échelle par m^{-1}, convergent presque sûrement

vers une limite lorsque $m \to \infty$, justifiant ainsi l'échelle en m^{-1} par rapport à l'échelle en $m^{-1/2}$ à l'initialisation.

L'idée intuitive de la convergence des produits scalaires mis à l'échelle par m^{-1} est que les coordonnées des (pré-)activations restent approximativement i.i.d. pendant toute la phase d'entraînement pour de grandes valeurs de m. Comprendre comment les produits scalaires et les multiplications avec des matrices gaussiennes i.i.d. évoluent avec m et prendre en compte la corrélation potentielle entre différentes quantités est précisément ce qui permet de comprendre comment il convient de mettre à l'échelle l'initialisation et les taux d'apprentissage dans la limite de largeur infinie pour obtenir des mises à jour des poids qui contribuent de manière maximale sans entraîner d'explosion.

Suivre les échelles et les corrélations à mesure que l'entraînement progresse devient rapidement fastidieux pour les étapes de temps $t \geq 1$, et le Tensor Program offre un moyen de rendre les calculs systématiques dans la limite de largeur infinie. Il existe trois types d'objets dans le cadre du Tensor Program : (i) des matrices gaussiennes i.i.d. $\hat{W}$ de taille $m \times m$ avec un écart-type de $m^{-1/2}$, (ii) des vecteurs $z \in \mathbb{R}^m$ avec des coordonnées approximativement i.i.d. , et (iii) des scalaires $\omega \in \mathbb{R}$. Alors que les matrices gaussiennes représentent essentiellement les matrices d'un réseau neuronal à l'initialisation (ou des versions redimensionnées de celles-ci), les vecteurs peuvent être obtenus de deux manières : soit par un calcul de multiplication matricielle $z = \hat{W}x$ avec un autre vecteur x, soit par le biais d'une non-linéarité $z = \psi(z^1, \ldots, z^p; \omega_1, \ldots, \omega_q)$, où $\psi : \mathbb{R}^{p+q} \to \mathbb{R}$ est une fonction paramétrique appliquée *élément par élément*, c'est-à-dire que pour tout $j \in [1, m]$, $z_j = \psi(z_j^1, \ldots, z_j^p; \omega_1, \ldots, \omega_q)$. Ces vecteurs représentent les (pré-)activations ou les gradients par rapport aux (pré-)activations, et les scalaires sont obtenus grâce à des produits scalaires mis à l'échelle $x^\top y/m$ pour certains vecteurs x et y.

Les règles du Tensor Program, détaillées dans (Yang, 2020b; Yang and Hu, 2021), décrivent un système de calculs qui permet de dériver la limite de largeur infinie de toute série de calculs (appelée Tensor Program) en utilisant les trois opérations présentées ci-dessus. Ces règles établissent que tout scalaire $\omega = x^\top y/m$ converge presque sûrement vers une valeur finie lorsque $m \to \infty$ (c'est le résultat du Théorème principal). De plus, dans la limite de largeur infinie, les coordonnées d'un vecteur z ont toutes la même distribution décrite par la loi d'une seule variable aléatoire $Z \in \mathbb{R}$. En fait, les entrées du vecteur z convergent en loi vers la variable aléatoire Z. Si $z = \hat{W}x$, la loi de Z est donnée par $Z = \hat{Z} + \dot{Z}$ où $\hat{Z}$ est gaussienne, centrée et indépendante de x, et $\dot{Z}$ est une variable aléatoire tenant compte de la corrélation potentielle entre x et $\hat{W}$, le cas le plus crucial étant $x = \hat{W}^\top y$. Si $z = \psi(z^1, \ldots, z^p; \omega_1, \ldots, \omega_q)$, lorsque $m \to \infty$, la loi de Z est donnée par $Z = \psi(Z^{z^1}, \ldots, Z^{z^p}; \bar{\omega}_1, \ldots, \bar{\omega}_q)$ où Z^{z^r} est la loi limite des entrées du vecteur z^r et $\bar{\omega}_s$ est la limite presque sûre de ω_s. Enfin, la limite presque sûre de $\omega = x^\top y/m$ est égale à $\bar{\omega} = \mathbb{E}[Z^x Z^y]$. La description précise des règles d'un Tensor Program est fournie dans (Yang, 2020b; Yang and Hu, 2021), ainsi que les démonstrations

que tout système de calculs de largeur finie utilisant les trois opérations décrites ci-dessus (tel qu'un réseau neuronal avec pratiquement n'importe quelle architecture) peut être décrit dans la limite de largeur infinie $m \to \infty$ par un système correspondant de calculs sur les variables aléatoires limites Z.

Limitations du programme tensor. Nous discutons brièvement ici certaines des limitations du cadre du Tensor Program, que nous développons davantage dans le chapitre 2.

Une limitation évidente est que la définition des variables Z limites est récursive, avec des formules qui deviennent rapidement inextricables pour les réseaux de neurones profonds entraînés pendant plus d'une étape de l'optimisation par descente de gradient (SGD), de sorte que bien que la description de la limite soit claire, l'utilisation du Tensor Program pour étudier les propriétés de la dynamique d'entraînement au-delà des premières étapes de l'entraînement peut être peu pratique, sauf dans certains cas spécifiques (comme les paramétrisations intégrables, et la raison en est discutée dans le chapitre 2). Un autre inconvénient du Tensor Program est que seules les non-linéarités ψ ayant une certaine régularité sont autorisées pour que les résultats soient valables, ce qui empêche l'étude des réseaux de neurones avec une activation ReLU directement, bien que le cadre puisse éventuellement être étendu pour traiter des activations non lisses, mais au prix de preuves techniques fastidieuses.

De plus, le Tensor Program sous sa forme initiale (Yang and Hu, 2021) n'autorise qu'une initialisation gaussienne pour les matrices de poids. Cependant, l'universalité des calculs du Tensor Program et de son théorème principal a récemment été démontrée dans (Golikov and Yang, 2022), permettant des initialisations i.i.d. générales.

Contributions

Dans cette section, nous mettons en avant les objectifs poursuivis dans cette thèse ainsi que les principales contributions. Les réseaux IP, avec au moins deux couches, s'éloignent du comportement du noyau observé dans la paramétrisation NTK et produisent en réalité des dynamiques où les caractéristiques évoluent avec le temps, un fait qui semble crucial pour le succès empirique des réseaux neuronaux. L'objectif de cette thèse est d'étudier la dynamique des réseaux neuronaux à largeur infinie dans la paramétrisation intégrable, parfois profonde, parfois peu profonde. La première partie de cette thèse (Chapitre 2) est consacrée à une meilleure compréhension des dégénérescences qui surviennent pour les réseaux profonds dans la paramétrisation intégrable, et comment on peut les entraîner dans la limite de largeur infinie dans un cadre aussi proche que possible de ce qui se fait en pratique. La deuxième partie (Chapitre 3) se concentre sur la manière dont la dynamique des réseaux à deux couches à largeur infinie s'adapte aux symétries et à la struc-

ture d'une tâche donnée, et étudie en particulier le problème de l'apprentissage de sous-espaces de basse dimension. Enfin, la troisième partie (Chapitre 4) étudie différents algorithmes d'optimisation dans l'espace des mesures, qui fournissent soit des résultats théoriques de convergence globale avec un taux explicite, soit des méthodes pratiques où les neurones peuvent être ajoutés ou supprimés dynamiquement pendant l'entraînement.

Dynamique à largeur infinie des paramétrisations intégrables

Les paramétrisations intégrables à deux couches semblent avoir des propriétés favorables par rapport à la paramétrisation NTK, mais il semble qu'elles aient un comportement dégénéré avec plus de quatre couches dans le cadre standard où les poids d'une couche donnée sont initialisés indépendamment et identiquement. Notre objectif dans le Chapitre 2 est de relier différentes lignes de travail autour des réseaux neuronaux à largeur infinie, telles que les limites "mean-field" et le Tensor Program. En particulier, nous souhaitons mieux comprendre la nature de cette dégénérescence, proposer une solution au problème tout en restant dans un cadre aussi proche que possible des méthodes pratiques, et étudier les propriétés du modèle résultant dans la limite de largeur infinie.

Dégénérescence des paramétrisations intégrables

Araújo et al. (2019); Nguyen and Pham (2020) étudient la dynamique idéalisée de l'écoulement de gradient des IP avec des initialisations i.i.d. et plus de quatre couches, et observent que dans la limite de largeur infinie, les poids dans une couche intermédiaire donnée se déplacent tous de la même quantité déterministe dépendant uniquement du temps. Nous allons plus loin et prouvons que cette quantité est nulle même avec SGD, de sorte que les poids ne bougent pas du tout lorsque $m \to \infty$, ce qui fait que la fonction de prédiction est la même qu'à l'initialisation à n'importe quelle étape de temps. Le résultat suivant apparaît dans la Proposition 2.3.1 du Chapitre 2 : pour tout $t \geq 0$ et tout x,

$$\lim_{m\to\infty} f(\theta(t); x) = \lim_{m\to\infty} f(\theta(0); x) = 0,$$

où la convergence est presque sûre.

Dynamique avec de grands taux d'apprentissage initiaux

La question qui se pose naturellement est de savoir s'il existe une solution à ce problème dans le cas des initialisations i.i.d.. Nous répondons positivement à cette question. En étudiant précisément la magnitude des gradients à l'initialisation pour les IP profondes grâce au Tensor Program et à l'hypothèse d'homogénéité positive que nous considérons sur l'activation σ, nous remarquons que de grands taux d'apprentissage permettent à la fonction de prédiction d'évoluer de manière *non triviale* après la première étape d'entraînement. Cependant, il est important

de noter que le problème est beaucoup plus subtil que d'accélérer la dynamique (en utilisant des taux d'apprentissage plus élevés à mesure que m augmente) pour permettre l'apprentissage dans la limite de largeur infinie (comme c'est le cas pour deux couches où (S)GD pour les IP nécessite des taux d'apprentissage de l'ordre de m, voir Section 1.2.3). La subtilité réside dans le fait que, pour les IP profondes, les taux d'apprentissage à l'initialisation et aux étapes ultérieures ne peuvent pas avoir la même valeur pour permettre un entraînement stable dans la limite $m \to \infty$: si la croissance des taux d'apprentissage avec m est trop rapide, les (pré-)activations divergent après la première étape, et s'ils sont trop petits, elles restent bornées mais les poids ne bougent pas.

Le point de vue correct est que les fluctuations aléatoires doivent être amplifiées à l'initialisation via des **grands taux d'apprentissage initiaux** (LLR) avant de revenir aux taux d'apprentissage "standard" que l'on trouve dans la littérature sur les IP. Nous montrons que les magnitudes correctes pour les taux d'apprentissage sont, à $t = 0$ (initialisation), $\eta_1 = \eta_{L+1} = m^{(L+1)/2}$ et $\eta_l = m^{(L+2)/2}$ pour $l \in [2, L]$, et pour $t \geq 1$, $\eta_1 = \eta_{L+1} = m$ et $\eta_l = m^2$ pour $l \in [2, L]$. Sous des hypothèses légères sur la valeur initiale de la perte et sur les données d'entrée, nous prouvons dans le Théorème 2.4.1 que lorsque l'on utilise ces taux d'apprentissage, les résultats suivants sont vérifiés :

$$
\begin{aligned}
&f(\theta(0); x) \xrightarrow[m\to\infty]{a.s.} 0, \\
&f(\theta(1); x) \xrightarrow[m\to\infty]{a.s.} f_1^\infty, \quad 0 < |f_1^\infty| < \infty \ a.s., \\
&f(\theta(2); x) \xrightarrow[m\to\infty]{a.s.} f_2^\infty, \quad |f_2^\infty| < \infty \ a.s.
\end{aligned}
$$

L'hypothèse d'homogénéité est cruciale ici, bien que les magnitudes des premières passes avant et arrière puissent également être bien comprises lorsque $\sigma'(0) \neq 0$, mais l'étude complète nécessiterait une analyse séparée.

Connexion avec μP

Nous souhaitons maintenant comprendre les propriétés d'un réseau formé avec le programme d'apprentissage que nous venons de proposer, que nous appelons IP-LLR. Nous établissons un lien entre IP-LLR et μP récemment proposé (Yang and Hu, 2021) : nous montrons que, dans la limite de largeur infinie, IP-LLR est en fait une version modifiée de μP où les matrices de poids à $t = 0$ sont initialisées avec les premières mises à jour de poids de μP au lieu de l'initialisation gaussienne aléatoire habituelle. Autrement dit, nous "oublions" l'initialisation aléatoire de μP après la première étape de gradient.

Résultats numériques et autres alternatives

Nous explorons également dans le Chapitre 2 d'autres alternatives qui permettent l'entraînement pour les IP profondes i.i.d. et montrons (théoriquement ainsi

qu'empiriquement) que les deux autres options naturelles que nous considérons mènent à des comportements dégénérés. Nous complétons nos résultats théoriques par des expériences numériques approfondies pour corroborer nos résultats et démontrer que nos énoncés mathématiques semblent valables avec des hypothèses beaucoup plus générales (fonctions d'activation non homogènes ou non lisses). Les directions futures incluent l'extension de nos résultats théoriques à des fonctions non homogènes ou non lisses ainsi que l'analyse plus précise des différences qualitatives dans la dynamique d'entraînement d'IP-LLR et de μP.

Symétries dans la dynamique des réseaux à deux couches de largeur infinie

Dans la quête théorique visant à mieux comprendre comment les réseaux neuronaux apprennent des représentations des données d'entrée pour résoudre la tâche qui leur est présentée, il est naturel de se pencher sur le problème de savoir comment les réseaux s'adaptent aux symétries de la fonction qu'ils essaient d'apprendre. Les symétries peuvent revêtir diverses natures, mais nous nous concentrons dans le Chapitre 3 sur les **symétries orthogonales**, et en particulier sur le cas où la fonction cible f^* ne dépend que de la projection orthogonale dans un sous-espace de basse dimension de $\mathbb{R}^d$. Nous étudions les symétries induites par celles de f^* sur la dynamique de l'écoulement de gradient des réseaux ReLU à deux couches de largeur infinie. Dans ce contexte, les réseaux de largeur infinie ont l'avantage de permettre l'émergence de symétries dans la dynamique d'apprentissage qui ne sont qu' approximatives à largeur finie.

Comme mentionné dans la Section 1.2.3, le cas où f^* ne dépend que de la projection dans un sous-espace de dimension inférieure a déjà été étudié du point de vue statistique dans (Bach, 2017; Chizat and Bach, 2020), en mettant l'accent sur les propriétés favorables en termes de généralisation des réseaux à deux couches de largeur infinie avec des activations positivement homogènes. Cependant, la question de savoir si (S)GD est effectivement capable d'apprendre ce sous-espace ou non n'est pas abordée. De manière similaire, Cloninger and Klock (2021) et Damian et al. (2022) étudient comment une seule étape de SGD sur les poids de la couche d'entrée est déjà capable d'induire des propriétés statistiques favorables avec des bornes ne dépendant que de la dimension du sous-espace et non de celle de l'espace ambiant. Plus proche de notre approche, Mousavi-Hosseini et al. (2022) montrent que faire (S)GD uniquement sur la première couche aligne déjà les poids avec le sous-espace de basse dimension lorsque l'on utilise une régularisation L^2 suffisamment forte. Abbe et al. (2022) parviennent à prouver que la dynamique de l'écoulement de gradient est capable d'apprendre la structure de basse dimension dans un cadre similaire au nôtre grâce à leur hypothèse forte selon laquelle les données sont des variables de Rademacher (c'est-à-dire que leurs entrées appartiennent à $\{-1, 1\}$).

En étudiant les symétries, un autre objectif est d'évaluer si des résultats de convergence quantitative peuvent être obtenus avec les hypothèses de symétrie

ajoutées. Bien que de nombreux résultats de convergence globale existent dans la littérature pour les réseaux à deux couches (Chizat and Bach, 2018; Nguyen and Pham, 2020; Sirignano and Spiliopoulos, 2020; Wojtowytsch, 2020), aucune vitesse de convergence n'est généralement disponible. Nous démontrons que pour des instances particulières, une convergence exponentielle peut être prouvée.

Dans le Chapitre 3, nous étudions la dynamique d'entraînement de réseaux à deux couches de largeur infinie où les deux couches sont entraînées, et nous nous concentrons sur la dynamique WGF plutôt que sur les propriétés statistiques. Nous travaillons en supposant que la distribution des données d'entrée est sphériquement symétrique, et nous optimisons l'objectif de risque de population pour permettre l'émergence de symétries exactes.

Résultats généraux pour les symétries orthogonales

Nous montrons d'abord que dans notre cadre, si f^* est invariant sous une transformation orthogonale quelconque, alors la mesure μ_t et le prédicteur $f(\mu_t;\cdot)$ héritent de cette invariance (voir plus de détails dans la Proposition 3.2.1). Nous appliquons ensuite ce résultat à des cas spécifiques où f^* est invariant sous un sous-groupe quelconque de transformations orthogonales.

Convergence exponentielle pour les fonctions cibles impaires

Une conséquence du résultat discuté précédemment est que si f^* est une fonction impaire, alors $f(\mu_t;\cdot)$ est également impaire. Il découle ensuite de l'identité $\sigma(z)-\sigma(-z)=z$ satisfaite par la ReLU que le prédicteur est en réalité linéaire : $f(\mu_t;x)=w(t)^\top x$ avec $w(t)=\frac{1}{2}\int w^1\, w^2 \mathrm{d}\mu_t(w^1,w^2)$. Cette linéarisation est différente du comportement du NTK : les poids des deux couches évoluent de manière non triviale, mais les symétries du problème impliquent une dégénérescence vers des prédicteurs linéaires. En fait, cette dégénérescence n'est pas surprenante, car le minimiseur du risque doit être linéaire dans ce contexte. Nous montrons dans le Théorème 3.3.2 que la dynamique WGF converge de manière **exponentielle** vers ce minimiseur global de l'objectif d'entraînement : étant donné le minimum global F^*, nous montrons qu'il existe une constante positive $c>0$ et un instant $t_0\geq 0$, tels que pour tout $t\geq t_0$, on a

$$F(\mu_t)-F^*\leq e^{-c(t-t_0)}\left(F(\mu_{t_0})-F^*\right).$$

Il convient de noter que dans ce cadre, bien que le prédicteur soit linéaire, la dynamique WGF reste non linéaire, car le chemin d'optimisation est différent de l'optimisation de la paramétrisation linéaire $f(w;x)=w^\top x$: définir $w(t)=\frac{1}{2}\int w^1\, w^2 \mathrm{d}\mu_t(w^1,w^2)$ ou $w(t)$ comme le gradient du critère $\tilde{F}:w\in\mathbb{R}^d\mapsto \frac{1}{2}\mathbb{E}_{x\sim\rho}\left[(w^\top x-f^*(x))^2\right]$ *n'entraîne pas* le même chemin d'optimisation, même si dans ce cas, les deux convergent vers le meilleur prédicteur linéaire.

L'hypothèse sur f^* est évidemment restrictive, mais elle montre que dans ce cas particulier, il est possible d'obtenir une vitesse de convergence pour la WGF, bien que aucune vitesse ne soit connue en général. D'autres paramétrages ont également été étudiés dans la littérature afin de fournir des taux de convergence : E et al. (2020) parviennent à prouver une *convergence locale* en $O(1/t)$ pour les entrées unidimensionnelles, et Daneshmand and Bach (2022) prouvent également une convergence globale au taux de $O(1/t)$ pour les entrées bidimensionnelles et les fonctions cibles avec un nombre fini d'atomes et une fonction d'activation bien conçue.

Dynamique de flot de gradient de dimension inférieure

Enfin, nous nous intéressons au cas où $f^*(x) = f_H(x^H)$ où H est un sous-espace de basse dimension de $\mathbb{R}^d$, $f_H : H \to \mathbb{R}$, et x^H est la projection orthogonale de x sur H. Une telle fonction f^* est invariante sous toutes les transformations orthogonales qui préservent H, et il découle facilement des résultats sur les symétries orthogonales discutés précédemment que c'est également le cas pour le prédicteur $f(\mu_t; \cdot)$. En particulier, cela implique qu'il n'y a de dépendance que par rapport à l'orthogonal de H à travers la norme : $f(\mu_t; x) = \tilde{f}_t(x^H, ||x^\perp||)$ où $x^\perp = x - x^H$ est la projection orthogonale sur l'orthogonal de H.

Le défi maintenant est de montrer que lorsque $t \to \infty$, la dépendance par rapport à $||x^\perp||$ s'estompe, ne laissant que la dépendance par rapport à la projection orthogonale sur H. Cela signifierait que les caractéristiques apprises par le réseau se sont *adaptées* à la structure de basse dimension du problème. Il s'agit d'un problème difficile à résoudre théoriquement. Cependant, numériquement, on peut effectivement observer que la dépendance à $||x^\perp||$ disparaît avec le temps. Bien qu'il soit difficile de prouver que la mesure μ_t tend à être supportée sur le sous-espace H pour de grandes valeurs de t, il est possible de montrer que les dynamiques d'entraînement elles-mêmes peuvent être réduites à un gradient flow de dimension inférieure. Dans le cas où f^* dépend uniquement de la projection orthogonale sur H, nous montrons que la WGF peut être réduite à un gradient flow Wasserstein-Fisher-Rao sur un nombre plus restreint de dimensions. Parmi ces dimensions, une composante correspond à la direction des neurones d'entrée sur la sphère unitaire de H, et l'autre correspond à l'angle entre les neurones d'entrée et H.

Nous allons encore plus loin lorsque la fonction f_H est positivement 1-homogène et nous prouvons dans le Théorème 3.4.3 que la WGF peut être réduite à un gradient flow Wasserstein-Fisher-Rao sur un **seul paramètre** correspondant à l'angle entre les neurones d'entrée et H. Nous montrons qu'il existe une paire de mesures non négatives $\tau_t^+, \tau_t^- \in \mathcal{M}_+([0, \pi/2])$, satisfaisant à l'équation d'advection-réaction suivante

$$\partial_t \tau_t^\pm = -\mathrm{div}\left(\pm V_t \tau_t^\pm\right) \pm 2 G_t \tau_t^\pm$$

où le terme de réaction G_t est la première variation d'une fonction objective sur $\mathcal{M}([0, \pi/2])$ incorporant les invariances du problème, et le terme d'advection $V_t = G_t'$ est la dérivée du terme de réaction. Cette réduction unidimensionnelle permet de simuler facilement les EDP, et nous démontrons numériquement que les dynamiques de la WGF conduisent à une mesure qui semble être supportée sur H lorsque $t \to \infty$, confirmant que dans ce cadre, les réseaux de largeur infinie sont capables d'apprendre le sous-espace de basse dimension qui importe pour la prédiction.

Discussion

Bien que nous montrions rigoureusement que les dynamiques WGF se réduisent à des dynamiques de dimension inférieure, il reste une question ouverte de savoir s'il peut être prouvé que lorsque $t \to \infty$, la mesure μ_t converge vers une mesure μ_∞ qui est supportée sur H. D'autres travaux (comme Mousavi-Hosseini et al., 2022; Abbe et al., 2022) ont étudié cette convergence avec des dynamiques modifiées, mais en général, le problème de prouver la convergence vers le sous-espace H à long terme n'a pas encore été résolu. Une autre direction future consisterait à essayer d'étendre la technique de preuve de Mousavi-Hosseini et al. (2022) au cadre des réseaux à deux couches de largeur infinie où les deux couches sont entraînées.

Optimisation sur l'espace des mesures : ajout et élagage dynamiques des neurones

Dans le chapitre 4, nous considérons des objectifs convexes génériques $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$ sur l'espace des mesures sur la sphère et proposons des algorithmes pour les minimiser. Ce cadre couvre en particulier l'optimisation de réseaux à deux couches avec un nombre de neurones non contraint. Notre objectif est double : (i) fournir un algorithme avec des garanties de convergence globale à un taux explicite lorsque l'objectif est régulier, (ii) proposer des méthodes qui se comportent bien en pratique à la fois en termes de performance et de calcul.

Bien que le Wasserstein GF converge de manière prouvée vers un minimum global pour les réseaux à deux couches de largeur infinie (Nguyen and Pham, 2020; Chizat and Bach, 2018; Wojtowytsch, 2020), aucune vitesse de convergence n'est connue en général. Nous proposons un algorithme pour minimiser F régulier et convexe avec un taux de $k^{-\frac{1}{d}}$ où k est le numéro d'itération. Cet algorithme s'inspire des méthodes de coordonnées pour l'optimisation en dimension finie (voir, par exemple, Wright, 2015) et implique l'échantillonnage d'un nouveau neurone à chaque étape, ce qui le rend prohibitif en termes de coût computationnel pour une utilisation en pratique.

Pour atténuer ce problème, nous envisageons de *pénaliser* l'objectif régulier pour encourager la parcimonie et limiter le nombre de neurones, et ainsi réduire les coûts de calcul encourus par l'algorithme. Nous considérons donc des objectifs de la forme $F = J + \lambda H$, où J est un terme régulier (tel que la perte empirique

pour les réseaux à deux couches paramétrés par des mesures) et H est une pénalité induisant la parcimonie. Nous étudions deux types différents de pénalités : tout d'abord une pénalité de variation totale, qui est l'analogue d'une pénalité L^1 en dimension finie, ce qui conduit à des algorithmes proximaux dans l'espace des mesures pour traiter la non-régularité de la variation totale. Ensuite, nous considérons des pénalités de noyau régulier avec des noyaux attractifs ou répulsifs. Alors que ces derniers ne suppriment pas explicitement les neurones, les dynamiques correspondantes amènent certains neurones à se rapprocher les uns des autres, et ils peuvent éventuellement être fusionnés de manière ad hoc au-delà d'un certain seuil.

Nous tenons à souligner que le travail présenté dans le chapitre 4 est toujours en cours au moment de la rédaction de cette thèse, et que certaines parties peuvent donc sembler incomplètes.

Convergence globale de la descente de coordonnées dans l'espace des mesures

En dimension finie, de nombreuses techniques différentes existent pour l'optimisation convexe en fonction du contexte : l'objectif est-il régulier ou non, les conditions de type Łojasiewiczsont-elles satisfaites, utilisons-nous le gradient complet ou une seule coordonnée à chaque étape ? Nous passons en revue de telles techniques dans la Section 4.3, car bon nombre des idées sont utiles dans notre cadre, et en nous inspirant de ces méthodes, nous considérons un objectif régulier et convexe $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$ et proposons un algorithme de descente de coordonnées dans l'espace des mesures signées qui permet de le minimiser à un taux de $k^{-\frac{1}{d}}$.

Dans ce cadre, une coordonnée est considérée comme un neurone $u \in \mathbb{S}^{d-1}$ et l'objectif de la descente de coordonnées est de minimiser, étant donnée une mesure $\mu \in \mathcal{M}(\mathbb{S}^{d-1})$, une borne supérieure sur $F(\mu + t\delta_u)$ sur $t \in \mathbb{R}$ où δ_u est la mesure de Dirac en μ. À partir d'un seul atome $\mu_0 = c_0\delta_{u_0}$, cela donne naissance à un algorithme où à chaque itération k, l'itéré actuel a la forme $\mu_k = \sum_{i=0}^{k} c_i\delta_{u_i}$ et nous échantillonnons un nouveau neurone $u_{k+1} \in \mathbb{S}^{d-1}$ uniformément sur la sphère et définissons son poids c_{k+1} en minimisant sur $t \in \mathbb{R}$ une borne supérieure sur $F(\mu_k + t\delta_{u_{k+1}})$. Nous montrons dans le Lemme 4.4.2 qu'une inégalité de type Łojasiewicz est vérifiée pour les itérés, puis déduisons avec des arguments similaires à ceux de la dimension finie la convergence de cet algorithme vers un minimiseur global en espérance avec un taux explicite. Nous montrons dans le Theorem 4.4.3 qu'il existe une constante $C > 0$ telle que pour tout $k \geq 1$, il tient :

$$0 \leq \mathbb{E}[F(\mu_k) - F^*] \leq \frac{C}{k^{1/d}}.$$

Cette descente de coordonnées dans $\mathcal{M}(\mathbb{S}^{d-1})$ doit être comprise dans la géométrie L^2, car chaque étape est équivalente, en espérance, à la minimisation d'une borne supérieure sur $F(\mu_k + \nu)$ impliquant la norme au carré $||\nu||^2_{L^2(\omega_d)}$

sur $\nu \in L^2(\omega_d)$, où ω_d est la distribution uniforme sur $\mathbb{S}^{d-1}$. En pratique, un tel algorithme peut être combiné avec des étapes de descente dans la géométrie de

Wasserstein, qui ont souvent un bon comportement empirique bien qu'elles ne fournissent pas de taux de convergence.

L'inconvénient de l'algorithme de descente de coordonnées présenté ci-dessus est que le nombre de neurones augmente linéairement avec le numéro d'itération k, ce qui le rend prohibitif en termes de coût computationnel pour une utilisation en pratique. Ainsi, nous discutons ci-dessous de l'ajout de pénalités à l'objectif régulier qui encouragent la parcimonie et offrent un équilibre entre la convergence globale et le coût computationnel.

Algorithmes proximaux pour les pénalités de variation totale

Nous examinons maintenant un objectif composite du type $F(\mu) = J(\mu) + \lambda|\mu|_{TV}$ où J est régulier et $|\mu|_{TV}$ est la norme de variation totale de $\mu \in \mathcal{M}(\mathbb{S}^{d-1})$. Cela ressemble à une pénalité L^1 en dimension finie, qui est connue pour induire de la parcimonie. La pénalité de variation totale n'est pas régulière, et en nous inspirant à nouveau des méthodes convexes en dimension finie, nous proposons un algorithme proximal pour la minimisation de l'objectif pénalisé. Alors qu'en dimension finie, des taux de convergence peuvent encore être obtenus pour les méthodes proximales, dans notre cadre, la convergence globale est perdue et nous n'avons aucun contrôle explicite sur le nombre de neurones. En effet, l'étape de descente de coordonnées proximale est donnée par un opérateur de *soft-thresholding* : à chaque itération, le nombre de neurones peut augmenter d'un ou rester constant, mais il reste constant uniquement s'il n'y a pas de changement dans la valeur de l'objectif d'une itération à l'autre. Dans ce contexte, parcimonie et convergence globale sont incompatibles : la diminution de l'objectif ne peut être obtenue qu'en ajoutant un nouveau neurone.

Pour atténuer ce problème, nous envisageons une modification de l'algorithme proximal où nous alternons entre l'échantillonnage d'un nouveau neurone sur la sphère et l'échantillonnage parmi les neurones existants de l'itéré courant μ_k. Lors de l'échantillonnage parmi les neurones existants, l'étape proximale est également donnée par un opérateur de *soft-thresholding*, mais cette fois, le nombre de neurones peut rester constant ou diminuer d'un d'une itération à l'autre, et nous pouvons obtenir à la fois une diminution du nombre de neurones et une diminution de l'objectif. Malheureusement, nous n'avons toujours pas de garanties théoriques de convergence ni de contrôle sur le nombre de neurones, mais il semble que cette méthode se comporte bien en pratique et parvient à à la fois diminuer l'objectif et limiter la croissance du nombre de neurones.

Pénalités de noyau lisse

Une autre approche que nous adoptons est l'étude de pénalités de noyau lisse qui attirent ou repoussent les neurones voisins. Dans ce cadre, nous considérons

également un objectif composite du type $F(\mu) = J(\mu) + \lambda H(\mu)$ où $H(\mu) = \int K(u,v)\mathrm{d}|\mu|(u)\mathrm{d}|\mu|(v)$ est une pénalité de noyau, et $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ est un noyau symétrique, lisse et non négatif. Les noyaux que nous considérons sont des noyaux de produit scalaire $K(u,v) = \kappa(\langle u, v\rangle)$ avec $\kappa : \mathbb{R} \to \mathbb{R}_+$.

Nous disons que le noyau est *attractif* si κ est une fonction décroissante, et qu'il est *répulsif* si κ est une fonction croissante. En général, nous considérons $\kappa_{a,\sigma}(s) = 1 - e^{s/\sigma^2}$ pour les noyaux attractifs et $\kappa_{r,\sigma}(s) = e^{s/\sigma^2}$ pour les noyaux répulsifs, où $\sigma > 0$ est un paramètre contrôlant la portée de l'interaction entre différents atomes sur la sphère. Bien que de tels noyaux n'éliminent pas explicitement les neurones dans les itérations de l'algorithme, les dynamiques correspondantes induisent certains neurones à se rapprocher (même dans le cas répulsif car les forces répulsives de nombreux neurones différents peuvent en pousser certains les uns vers les autres) au point que nous pouvons effectivement les fusionner si leur distance est inférieure à un certain seuil. Par conséquent, ces méthodes induisent implicitement un certain contrôle sur le nombre de neurones. Nous alternons entre des étapes de descente de coordonnées qui devraient diminuer l'objectif mais au prix de l'ajout de neurones, et des étapes de Wasserstein-Fisher-Rao qui devraient permettre la fusion des particules tout en se comportant bien en termes de diminution de l'objectif sur le plan empirique.

Bien que de telles pénalités de noyau soient théoriquement motivées, il n'y a aucune garantie directe de convergence ni de contrôle de la croissance du nombre de particules, mais nous montrons qu'elles ont un comportement intéressant d'un point de vue empirique.

Discussion

L'algorithme proximal que nous présentons pour minimiser l'objectif non lisse avec la pénalité de variation totale présente un bon comportement empirique, mais il reste encore une question ouverte de savoir si une preuve de convergence peut être obtenue dans ce cadre. Concevoir des algorithmes qui fournissent à la fois une garantie théorique de convergence et qui sont en même temps réalisables sur le plan computationnel (du moins empiriquement) est difficile, et nous laissons l'exploration d'approches alternatives que celles que nous présentons pour des travaux futurs.

Notation

Integers

- d: the input dimension.
- m: the width of a network.
- n: the number of samples in the training dataset.
- $[a, b]$: the set of integers $\{a, \dots, b\}$ for $a, b \in \mathbb{N}$ and $a \leq b$. There should be no confusion with the segment of real numbers between a and b given the context.

Spaces of measures

- $\mathcal{P}_q(\Omega)$: the space of probability measures on a domain $\Omega \subset \mathbb{R}^p$ with finite q-th moment, *i.e.*, $\int_\Omega ||x||^q \mathrm{d}\mu(x) < \infty$.
- $\mathcal{M}_+(\Omega)$: the set of non-negative measures on a domain $\Omega \subset \mathbb{R}^p$ with finite mass, *i.e.*, $\mu(\Omega) < \infty$.
- $\mathcal{M}(\Omega)$: the set of signed measures on a domain $\Omega \subset \mathbb{R}^p$ with finite total variation, *i.e.*, $|\mu|(\Omega) < \infty$ where $|\mu| = \mu^+ + \mu^- \in \mathcal{M}_+(\Omega)$ is the sum of the positive and negative parts of μ.

Spaces of functions

- $\mathcal{C}(\Omega)$: the space of continuous functions on a domain $\Omega \subset \mathbb{R}^p$.
- $\mathcal{C}_b(\Omega)$: the space of continuous and bounded functions on a domain $\Omega \subset \mathbb{R}^p$.
- $\mathcal{C}_c^1(\Omega)$: the space of continuous and compactly supported functions on a domain $\Omega \subset \mathbb{R}^p$.

Symbols

- ∇: the gradient of a differentiable function.
- div: the divergence operator, defined for differentiable functions $f : \Omega \subset \mathbb{R}^p \to \mathbb{R}^p$ by $\mathrm{div}(f) = \sum_{i=1}^p \frac{\partial f_i}{\partial x_i}$.
- id: the identity map $\mathrm{id} : x \mapsto x$.
- $\langle \cdot, \cdot \rangle$: canonical euclidean inner product in $\mathbb{R}^p$.
- $||x||$: the euclidean norm of a vector x, *i.e.*, $||x|| = \sqrt{\sum_{i=1}^p x_i^2}$.

- $(\cdot)^\top$: the transposition operator.
- $\mathbb{S}^{p-1}$: the unit sphere of $\mathbb{R}^p$, *i.e.*, the set $\{x \in \mathbb{R}^p : ||x|| = 1\}$.
- $\mathbb{R}_+$: the set of non negative real numbers, *i.e.*, $[0, \infty)$.
- $\mathcal{N}(\mu, \Sigma)$: the Gaussian distribution with mean vector $\mu \in \mathbb{R}^p$ and covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$.
- $X_{\#}\mu$: the push-forward of the measure μ by the map X, characterized by the equality $\int \varphi \mathrm{d}(X_{\#}\mu) = \int \varphi \circ X \mathrm{d}\mu$ which holds for any measurable φ.

Abbreviations

- (S)GD: (Stochastic) Gradient Descent
- (W)GF: (Wasserstein) Gradient Flow
- NTK: Neural Tangent Kernel
- IP(s): Integrable parameterization(s)
- ReLU: the Rectified Linear Unit, *i.e.*, : $z \in \mathbb{R} \mapsto \max(0, z)$.

2 - Infinite-width limit of integrable parameterizations of deep neural networks

2.1 . Introduction

While artificial neural networks routinely achieve state-of-the art performance in various real-world machine learning tasks, it is still a theoretical challenge to understand why and under which conditions they perform so well. The training algorithm—typically a variant of stochastic gradient descent (SGD) with random initialization—plays a central role in this performance but is difficult to analyze for general neural network architectures, because of their highly non-linear and compositional structure. Large-width asymptotics, which have previously been considered for other purposes (Neal, 1995; Bengio et al., 2006), have recently been proposed to overcome some of these difficulties and have brought numerous insights on the training behavior of neural networks (Nitanda and Suzuki, 2017; Mei et al., 2018; Jacot et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Chizat and Bach, 2018; Sirignano and Spiliopoulos, 2020).

One of these insights is that the magnitude of the random weights at initialization has a dramatic impact on the learning behavior of neural networks (Chizat et al., 2019). For two-layer networks and with suitable learning rates, initializing the output layer weights with a standard deviation of $1/m$, where m is the width of the network, leads to *feature learning* when m is large, while the same network initialized with a standard deviation of $1/\sqrt{m}$ leads to the Neural Tangent Kernel (NTK) regime, a.k.a. *lazy* regime, where the network simply learns a linear predictor on top of fixed features. This observation suggests that *parameterizations*—that is, the choice of the scaling factors, with the width m, of the initial magnitude and of the learning rates of each layer of a neural network—are of fundamental importance in the theory of neural networks. While standard deep learning packages offer various choices of scale at initialization (Glorot and Bengio, 2010; He et al., 2015), those have been designed with the sole criterion in mind to have a non-vanishing first forward and backward passes for arbitrary depths. Theory now offers the tools to explore a larger space of parameterizations and study their dynamics beyond the first forward and backward passes in the infinite-width limit.

With more than two layers, the categorization of parameterizations is more subtle and there are disparate lines of work. On the one hand, some parameterizations still lead to the kernel regime, which is subject to an intense research activity (*e.g.*, Jacot et al., 2018, 2019; Allen-Zhu et al., 2019; Du et al., 2019; Arora et al., 2019; Geiger et al., 2020a,c; Yang, 2020a). Since this regime reduces to learning a linear predictor on top of fixed features in the large width limit, this parameteriza-

tion is of limited relevance to understand representation learning in networks used in practice (although it should be noted that non-asymptotic analyses reveal interesting effects, *e.g.*, Hanin and Nica, 2019). On the other hand, there is a growing literature around parameterizations where weights are initialized with a standard deviation of $1/m$ (except for the first layer). These are often called "mean-field" models but we prefer to call them *integrable parameterizations* (IPs) in this work[1], in reference to the fact that sums of m terms with standard deviation of order of $1/m$ are absolutely convergent. There already exists mathematical tools to describe the evolution of the parameters of IPs in the infinite-width limit but they are not fully satisfactory to understand the properties of the learned function in the standard setting used in practice (see review in Section 2.1.3).

Going beyond the dichotomy between the scales $1/m$ and $1/\sqrt{m}$, Yang and Hu (2021) have exhibited, using a technique called the *Tensor Program* (Yang, 2019, 2020a,b), a general categorization of parameterizations, in particular between those which allow feature learning and those which do not. As a result from their analysis, they singled out a *maximal update* parameterization μP where, as for the NTK parameterization, the intermediate layers' weights are initialized with a standard deviation of $1/\sqrt{m}$, but the last layer weights are initialized with a standard deviation of $1/m$: they show that with appropriate learning rates, this leads to maximal feature learning (in a certain sense). This parameterization had been previously considered in (Geiger et al., 2020b) where the authors study empirically the effect of the *scale* (Chizat et al., 2019) on learning.

In (Yang and Hu, 2021), IPs have been excluded from the analysis on the basis that they are *trivial*: if one follows the usual training procedure—which we refer to as *Naive-IP*—the network starts on a stationary point in the infinite-width limit and the learned function remains at its initial value.

2.1.1 . Motivation

We study IPs as they have been studied extensively in the mean-field literature and global convergence results can be obtained under specific assumptions. Additionally, shallow IPs have been shown to perform feature learning and thus appear to have a favourable behaviour compared to their NTK counterpart. We focus on homogeneous activation functions as they appear as a natural way to understand the magnitudes of the forward and backward passes *but we also show* that those magnitudes can be rigorously analyzed at initialization for commonly used activation functions such as ELU, GeLU or tanh. We also focus on i.i.d. initializations since this is what is used in practice. While global convergence results exist for IPs with i.i.d. initialization at depth 2 or 3 (Nguyen and Pham, 2020; Chizat and Bach, 2018), IPs have been shown to have degenerate behaviour for

[1] For deep neural networks, it is somewhat arbitrary to associate the term *mean-field* with a specific choice of scaling so we believe that this term lacks precision when it comes to discussing various parameterizations.

larger depths in this setting (Nguyen and Pham, 2020; Fang et al., 2020). With different assumptions, it is possible to remove those degeneracies and recover convergence results (Nguyen and Pham, 2020; Fang et al., 2020), however, our aim is to understand and characterize precisely the nature of the degeneracy with i.i.d. initializations and to propose a fix while staying as close as possible to what is done in practice. We stress that our goal is not to propose a new competitive method, but rather to clarify the literature by exhibiting the hidden link between different approaches.

2.1.2 . Contributions

Our goal is to draw connections between the various lines of research discussed above, in particular between "mean-field" limits and μP—which emerged through separate lines of work—and to improve our understanding of integrable parameterizations: when and why are they trivial? How can we avoid triviality and actually learn features? What are the salient properties of the resulting networks in the infinite-width limit? To answer these questions rigorously, we leverage the Tensor Program technique developed in (Yang, 2019, 2020a,b; Yang and Hu, 2021). Specifically, our contributions are the following:

- We first show in Theorem 2.3.1 that with learning rates constant in time, the functions learned using SGD for integrable parameterizations of neural networks with four layers or more either *remain at their value at initialization or explode* in the infinite-width limit when the weights are initialized using the standard zero-mean i.i.d. schemes used in practice.

- We show in Theorem 2.4.1 that using large learning rates, which grow as a power of m, for the first gradient step—and that step only—allows SGD to escape the initial stationary point for integrable parameterizations and to initiate a non-trivial learning phase. In fact, we prove in Theorem 2.4.2 that the resulting dynamic is equivalent to a modification of the dynamic of μP where the initial weights of intermediate layers are replaced with the first update obtained with the large learning rates. This highlights a non-obvious link between IPs as previously studied in the mean-field literature and μP through the Tensor Program when the initialization is i.i.d. While Theorem 2.4.1 uses the Tensor Program extensively, actually proving that the learned function moves away from its initialization is subtle and the proof technique is different from what can be found in (Yang and Hu, 2021).

- Other alternatives to using large learning rates exist to escape the initial stationary point and we study two of them which seem like natural choices: (1) using a non-centered law at initialization and (2) removing the scale factor in $1/m$ on the bias terms. While this is not an exhaustive list, both methods seem like natural candidates and we show that they lead to degenerate dynamics (see Section 2.5) compared to using large initial learning

rates and we confirm those findings numerically.

Advantages and drawbacks of the Tensor Program. The drawback of relying on the Tensor Program is that it restricts the class of activation functions one can consider (*e.g.*, ReLU and other variants such as LeakyReLU have to be excluded because of their non-smoothness). Two possible workarounds would be either to reprove a version of the Tensor Program allowing for a non-smooth activation functions, or to express non-smooth activations as limits of smooth function (*e.g.*, ReLU can be expressed as the limit of a parameterized GeLU) and take this limit inside the Tensor Program, both of which would require considerable technical work. Our numerical experiments confirm that our results appear to hold empirically with much less restrictive assumptions on the activation function.

On the other hand, we emphasize that although we present our results with i.i.d. Gaussian initializations for simplicity, the universality of the Tensor Program has been proved in (Golikov and Yang, 2022), showing that the Tensor Program framework and its Master Theorem are valid for much more general i.i.d. initializations.

Furthermore, the Tensor Program allows us to study SGD with *mini-batches* and *discrete step-size* to stay as close as possible to what is done in practice while it is standard in the mean-field literature to study continuous-time (infinitely small step-size) and full-batch GD (on the whole training set). However, because of the recursive nature of the state evolution equations of the Tensor Program, analyzing quantitatively long-term dynamics is challenging compared to the classical mean-field setting where such analyses are available and global convergence guarantees can be obtained under the proper assumptions.

Finally, we highlight that, in comparison to (Yang and Hu, 2021), because we restrict ourselves to certain classes of activation functions, our results hold for any choice of the scalar (width-independent) learning rate $\eta > 0$ while most of their results hold for some small value of η but for a more general class of activation functions.

The code to reproduce the results of the numerical experiments can be found at:
https://github.com/karl-hajjar/wide-networks.

2.1.3 . Related Work

While the study of infinitely wide neural networks has a long history (Barron, 1993; Neal, 1995, 1996; Kurková and Sanguineti, 2001; Mhaskar, 2004; Bengio et al., 2006; Bach, 2017), it is only recently that their training dynamics have been investigated. Two-layer neural networks with the IP enjoy some global convergence properties (Chizat and Bach, 2018) and favorable guarantees in terms of generalization (Bach, 2017; Chizat and Bach, 2020). Going beyond two layers, Nguyen and Pham (2020) and Pham and Nguyen (2020) study the infinite-width

limit of IPs and also prove global convergence results for networks with three layers or more. However, those results hold for standard zero-mean i.i.d. initialization schemes only for networks with two or three layers (which is consistent with the results of Section 2.3.1): for deeper networks they require non-standard (correlated) initializations. Nguyen and Pham (2020) show for deep networks that with i.i.d. initializations, the weights of any given intermediate layer all translate by the the same quantity in the limit when the initial bias is zero. We in fact show in Proposition 2.3.1 that when the initialization is centered around zero the degeneracy is even stronger as the learned function does not move away from its initial value in the limit.

Several other works describe the infinite-width limit of multi-layer IPs: Araújo et al. (2019) characterize the infinite-width dynamics via a model of McKean-Vlasov type, for which they prove existence and uniqueness of solutions, and Sirignano and Spiliopoulos (2021) prove a global convergence result for three-layer networks. They take the number of units in each layer to infinity sequentially and describe the dynamics of the limit as a system of differential equations over the weights/parameters. On the other hand, Fang et al. (2020) take the infinite-width limit for all layers at once (as in Araújo et al., 2019; Nguyen and Pham, 2020; Pham and Nguyen, 2020) and describe the resulting dynamics as an ODE over functions of the features (pre-activations) of the network. It is interesting to note that Araújo et al. (2019); Sirignano and Spiliopoulos (2021); Pham and Nguyen (2020) all discuss the difficulties associated with describing the dynamics of the infinite-width of IPs with more than three layers. As noted in (Araújo et al., 2019), and appropriately addressed by Nguyen and Pham (2020); Fang et al. (2020); Sirignano and Spiliopoulos (2021), there is a separation of time scales as soon as there are two hidden layers or more, where the gradients of the intermediate layers appear to scale as m^{-2} whereas the gradients of the input and output layers appear to scale as m^{-1}, requiring separate learning rate values which can make the analysis of the infinite-width limit more difficult.

In a separate line of work, Yang and Hu (2021) provide with the Tensor Program a theoretical tool to describe the infinite-width limit of different parameterizations of neural networks and categorize them between feature learning and kernel-like behavior. However, IPs with three layers or more are left out of this categorization. Using the same tools, we show that IPs with more than four layers are indeed trivial at any time step if the initial learning rates are not appropriately scaled with m under standard zero-mean i.i.d. initializations. This closes the gap with (Nguyen and Pham, 2020) which proves global convergence results for IPs with two or three layers initialized using those standard schemes. We also demonstrate in Section 2.4 how scaling the initial learning rates appropriately allows to properly train an IP—inducing a feature learning regime as defined in (Yang and Hu, 2021)—and connect the resulting model with a version of the maximal update parameterization μP (Yang and Hu, 2021) where the initial weights of the intermediate layers are

replaced by zero in the first update. We stress that our aim in the numerical experiments of Section 2.6 is not demonstrate that IP-LLR performs empirically better than μP but rather to show that on top of the theoretical connection between those two models, IP-LLR seems to be a valid way of training IPs with comparable performance to μP.

The setting where non-centered i.i.d. initialization laws are used is covered in (Nguyen and Pham, 2020), where it is shown that a certain collapse phenomenon occurs, namely that the updates of the entries of the weight matrix in a given layer are all equal to the same deterministic quantity in the large-width limit. We recover this result in Section 2.5.1 using different theoretical tools.

Tensor Program *vs.* other formalisms. In contrast to prior literature on IPs, we do not use the description of the infinite-width limit as a composition of integral transforms. With the standard (centered i.i.d.) initializations considered in this chapter, that description does not offer much insight about the limit beyond the fact that it starts on a stationary point. In order to escape this initial stationary point, we propose in this chapter to amplify the random fluctuations around the limit using large initial learning rates. The strength of the Tensor Program formalism (Yang, 2019, 2020a,b; Yang and Hu, 2021) is precisely that it is able to describe rigorously the magnitudes of these fluctuations and allows us to analyze the functions learned with various choices of learning rates. This formalism relies on techniques initiated in the statistical physics literature (Bayati and Montanari, 2011; Bolthausen, 2014) that use the Gaussian conditioning technique to describe the behavior of algorithms (such as message passing) involving random matrices and non-linearities.

2.1.4 . Organisation of the Chapter and Notations

We define and analyze integrable parameterizations in Section 2.3 and show that they are trivial for common choices of learning rates. In Section 2.4, we describe how a specific scaling of the learning rates allows to escape the initial stationary point, and further investigate the connection between IPs with large initial learning rates and μP. In Section 2.5, we present two alternative modifications of IPs to escape the initial stationary point and discuss the impact of each on the learning dynamics. Finally in Section 2.6 we present our numerical results.

We defer all the rigorous proofs of our theoretical results to the Appendix, so as to make the core message of our work stand out more clearly, and keep the flow of the results structured and easy to follow. Among other things, this prevents us from diving too deep into the Tensor Program formalism and calculations (which can be somewhat tedious and abstruse) in the main part of our work. Most proofs require heavy inductions on the time step t, and proving the induction step itself often involves inductions on l in the forward pass (from $l = 1$ to $l = L$) and in the backward pass (from $l = L$ to $l = 1$). Breaking down all these steps makes for

a lengthy Appendix, but the ideas of the proof are relatively straightforward, only their proper formal writing is tedious.

Throughout the chapter, for two integers p, q, we denote by $[p]$ the set $\{1, \dots, p\}$ and by $[p, q]$ the set $\{p, \dots, q\}$. We write $u \odot v$ for the Hadamard (*i.e.*, element-wise) product of two vectors u and v. We use Landau notations for comparing two real sequences (u_m) and (v_m): we write $u_m = O(v_m)$ when there exists a constant $C > 0$ such that $|u_m| \leq C|v_m|$ for large enough m, and $u_m = \Theta(v_m)$ when we both have $u_m = O(v_m)$ and $u_m = O(v_m)$. We similarly use the O (respectively Θ) notation for two sequences of real-valued random variables (u_m) and (v_m) when, almost surely, $u_m = O(v_m)$ (respectively $u_m = \Theta(v_m)$).

2.2 . General Setting

In this section, we introduce the general setting we consider for this work, as well as the corresponding notations. We also define precisely the notion of parameterization of a neural network and discuss examples of parameterizations commonly found in the literature.

2.2.1 . Network and Data

Training data. We consider a training dataset $\left\{(\xi^{(i)}, y^{(i)})\right\}_{i \in [n]}$ containing n (input, output) pairs with $\xi^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$. We will use $\xi^{(i)}$ or $y^{(i)}$ when we refer to the i-th sample in the training dataset, but use ξ_t and y_t to denote the sample(s) fed to train the network at time step t, that is for the $(t+1)$-th step of optimization.

Width and depth. Throughout this work, we consider a feed-forward fully connected neural network, with L hidden layers and a common width m. The total number of layers, *i.e.*, weight matrices and bias vectors will thus be $L + 1$, and most of our results are concerned with four or more layers, that is $L \geq 3$, and in the limit $m \to \infty$. The integer $l \in [L + 1]$ will always be used to index the layers of a network, and we call the **intermediate layers** of a network the layers indexed by $l \in [2, L]$ (*i.e.*, excluding input and output layers).

Activation function. We assume that all the neurons in the network share the same activation function $\sigma : \mathbb{R} \to \mathbb{R}$. The activation is always taken entry-wise and for any vector $h \in \mathbb{R}^m$, we denote by $\sigma(h)$ the vector $(\sigma(h_p))_{p \in [m]} \in \mathbb{R}^m$.

Weights and forward pass. We denote by $W^l(t)$ and $B^l(t)$ respectively the weight matrix and bias vector of layer l at time step t (*i.e.*, after t steps of SGD), and thus have $W^1(t) \in \mathbb{R}^{m \times d}$, $W^l(t) \in \mathbb{R}^{m \times m}$ for $l \in [2, L]$ and $W^{L+1}(t) \in \mathbb{R}^m$. At any time step t we denote by $h_t^l(\xi)$ and $x_t^l(\xi)$ the pre-activations and activations

respectively coming out of the l-th layer when feeding input ξ to the network (with the convention that $x_t^0(\xi) = \xi$). That is

$$h_t^l(\xi) := W^l(t)x_t^{l-1}(\xi) + B^l(t), \quad \text{and} \quad x_t^l(\xi) := \sigma(h_t^l(\xi)), \quad \text{for } l \in [1, L]. \tag{2.1}$$

Output. We denote the output of the network by

$$f_t(\xi) = f(\theta(t); \xi) := (W^{L+1}(t))^\top x_t^L(\xi) + B^{L+1}(t), \tag{2.2}$$

where $\theta(t)$ denotes the set of all network parameters at time t. We often drop the dependency of the forward pass on the input ξ for brevity and simply use h_t^l, x_t^l instead of $h_t^l(\xi), x_t^l(\xi)$ as it should always be clear from the context which input is being fed to the network. Note that the weights and biases as well as all the (pre-)activations depend on the width m of the network (through their dimensions) but we omit this dependency for clarity.

Loss. We denote by ℓ the loss function used to train the network, which is a function from $\mathbb{R}^2$ to $\mathbb{R}$. The fit of a prediction $\widehat{y}$ is thus measured by $\ell(y, \widehat{y})$ where y is the desired output. In all this work, we make the following assumption on the loss function ℓ, which is met by most common loss functions:

Assumption 1 (Smooth loss *w.r.t.* second argument)**.** The loss ℓ is differentiable with respect to its second argument and $\partial_2\ell(y, \cdot)$ is a continuous function for any $y \in \mathbb{R}$.

Assumption 1 is essentially here to guarantee that if the sequence $(\widehat{y}^{(m)})_{m\in\mathbb{N}^*}$ converges almost surely to some $\widehat{y}^{(\infty)}$, then $\partial_2\ell(y, \widehat{y}^{(m)})$ also converges almost surely to $\partial_2\ell(y, \widehat{y}^{(\infty)})$.

2.2.2 . Parameterizations of Neural Networks

The fact that the magnitude of the initialization of the weights and of the scale pre-factor for the weights are key quantities that determine the learning regime achieved by neural networks—and more generally by differentiable models—was pointed out in (Chizat et al., 2019). In this chapter, we are interested in the behavior of neural networks when their width m goes to infinity, and we refer to as a *parameterization* of a neural network the choice of how (a) the pre-factor of the weights, (b) the standard deviation at initialization and (c) the learning rates, evolve as a function of m. This concept was called an abc-parameterization by Yang and Hu (2021), because these dependencies are given by m^{-a}, m^{-b} and m^{-c}.

As explained by these authors, one of those three choices is actually redundant, and one can do with only the choice of two among those three scales. We take the point view considering a parameterization as a choice of scale for the pre-factor

of the weights (a) and a choice of scale for the learning rates (c) while the random weights are always initialized (b) with standard i.i.d. Gaussians $\mathcal{N}(0,1)$. We make this (arbitrary) choice as typically in the literature, different models of the infinite-width limit correspond to different choices of scales for the weights' pre-factors, *e.g.*, NTK corresponds to a pre-factor in $1/\sqrt{m}$ while "mean-field" models correspond to a choice of pre-factor in $1/m$ for the weights. We thus define below ac-parameterizations which are a slight variation of the abc-parameterizations introduced in (Yang and Hu, 2021).

Definition 2.2.1. (ac-parameterization). An ac-parameterization of an L-hidden layer fully-connected neural network is a choice of scalar exponents $(a_1, \dots, a_{L+1})$, and $(c_1, \dots, c_{L+1})$ such that for any layer $l \in [L+1]$,

(i) the **learnable weights** (*i.e.*, those over which we optimize) are initialized with independent standard Gaussian random variables $w^l_{jq}(0) \sim \mathcal{N}(0,1)$, i.i.d. over (l,j,q), *i.e.*, $w^l(0) = U^l$ with $(U^l)_{l\in[L+1]}$ independent random matrices with i.i.d. standard Gaussian entries,

(ii) the **learnable biases** are initialized independently of the weights, with $b^l_j(0) \sim \mathcal{N}(0,1)$, i.i.d. over (l,j), *i.e.*, $b^l(0) = v^l$ with $(v^l)_{l\in[L+1]}$ independent standard Gaussian random vectors, independent of U^l,

(iii) the **effective weights** $W^l(t)$ used to compute the pre-activations at time t are $W^l(t) = m^{-a_l} w^l(t)$, and the **effective biases** are $B^l(t) = m^{-a_l} b^l(t)$, so that the pre-activations are

$$h^l_t = W^l(t) x^{l-1}_t + B^l(t) = m^{-a_l}\left(w^l(t)\sigma(h^{l-1}_t) + b^l(t)\right), \quad l \in [1, L],$$

and the output is

$$f(\theta(t);\xi) = m^{-a_{L+1}}\left(w^{L+1}(t)^T \sigma(h^L_t(\xi)) + b^{L+1}(t)\right),$$

(iv) the $(t+1)$-th update of learnable weights and biases is given by the update rules

$$\Delta w^l(t+1) := w^l(t+1) - w^l(t) = -\eta m^{-c_l} \nabla_{w^l} \ell\left(y_t, f(\theta(t);\xi_t)\right),$$
$$\Delta b^l(t+1) := b^l(t+1) - b^l(t) = -\eta m^{-c_l} \nabla_{b^l} \ell\left(y_t, f(\theta(t);\xi_t)\right),$$

where $\theta(t) = \left\{(w^1(t), b^1(t)), \dots, (w^{L+1}(t), b^{L+1}(t))\right\}$ is the full set of all network parameters, (ξ_t, y_t) represent the input(s) and target(s) to the network at step t and $\eta \in \mathbb{R}^*_+$ is the scalar part of the learning rate which does not depend on m and which we call the **base learning rate**. We denote by $\eta_l := \eta m^{-c_l}$ the full learning rate for layer l.

Remark.

1. Compared to the definition of (Yang and Hu, 2021), we allow for different values of c_l at different layers and remove the redundant initialization scale (that is the b in abc-parameterizations). Any abc-parameterization with constant c for all layers (as presented in Yang and Hu, 2021) can be recovered (same effective weights and biases at any time step) with an ac-parameterization with individual learning rates at each layer via the re-parameterization $a_l \leftarrow a_l + b_l$, $b_l \leftarrow 0$, $c_l := c - 2b_l$.

2. As we study the infinite-width limit $m \to \infty$, we need to consider an infinite number of random weights at initialization. To this end, we consider for any $l \in [2, L]$, two infinite lists of i.i.d. standard Gaussian variables, independent of each other: $(U^l_{jq})_{j,q\in\mathbb{N}^*}$ and $(v^l_j)_{p\in\mathbb{N}^*}$, and often simply call, by an abuse of notations, $U^l = (U^l_{jq})_{1\leq j,q\leq m}$ for the corresponding matrix at width m and $v^l = (v^l_j)_{1\leq j\leq m}$ the corresponding bias vector at width m. We proceed similarly at initialization for the input weights U^1 and the output vector U^{L+1}.

3. The $(t+1)$-th update of the effective weights is given by $\Delta W^l(t+1) := W^l(t+1) - W^l(t) = -\eta m^{-(2a_l+c_l)}\nabla_{W^l}\ell(y_t, f(\theta(t);\xi_t))$, and the update of the effective biases by the following equation $\Delta B^l(t+1) := B^l(t+1) - B^l(t) = -\eta m^{-(2a_l+c_l)}\nabla_{B^l}\ell(y_t, f(\theta(t);\xi_t))$

Examples of ac-parameterizations:

NTK parameterization. For the NTK parametrization (Jacot et al., 2018) the scaling is $a_1 = 0$ for the input layer, and $a_l = 1/2$ for all the other layers $l \in [2, L+1]$. The scaling of the learning rates is $c_l = 0$ for all layers. Neural networks in the NTK parametrization have been shown to behave as kernel methods in the infinite-width limit (Jacot et al., 2018; Yang, 2020a) and there is no feature learning in that limit.

μP. To avoid the lazy training phenomenon arising in the NTK parameterization, Yang and Hu (2021) propose to adjust the scale of the output layer by setting $a_{L+1} = 1$, while keeping $a_1 = 0$ and $a_l = 1/2$ for the intermediate layers $l \in [2, L]$. The learning rates are appropriately adjusted: $c_l = -1$ for any layer l. With this parameterization, Yang and Hu (2021) show that feature learning (see Definition A.2.1 in Appendix A.2.3 for a precise statement) occurs at every layer.

Integrable Parameterizations (IPs). The limits investigated in Araújo et al. (2019); Sirignano and Spiliopoulos (2021); Pham and Nguyen (2020); Weinan and Wojtowytsch (2020) are associated to a scale multiplier in $1/m$ for all layers except the first one. This corresponds to the choice $a_1 = 0$ and $a_l = 1$ for

$l \in [2, L+1]$. We choose the adjective "integrable" in reference to the absolute convergence of sums of the form $(1/m)\sum_q x_q$ for i.i.d. random variables with finite expectation. Integrable parameterizations really refer to a class of ac-parameterizations, because various choices for the learning rate exponents c_l are admissible.

Naive-IP. In the mean-field literature, integrable parameterizations often come with the standard learning rates corresponding to $c_1 = c_{L+1} = -1$ for the input/output layers and $c_l = -2$ for the intermediate layers $l \in [2, L]$, see *e.g.*, (Araújo et al., 2019, Remark 3.4), (Fang et al., 2020, Algorithm 1), (Weinan and Wojtowytsch, 2020, Lemma 5.1), and (Sirignano and Spiliopoulos, 2021, Equation 4.3). Mean-field models with these learning rates are the natural counterparts of the infinite-width limits where sums are replaced by integrals, and we call the integrable parameterization with this specific choice of learning rates the *Naive Integrable Parameterization*.

When $L = 1$, μP and the Naive-IP coincide. For deeper networks, in the setting of abc-parameterizations described in (Yang and Hu, 2021), μP and Naive-IP correspond to the same parameterization (same values for a and c) except that the weights of the intermediate layers are initialized with a standard deviation of $1/m$ for Naive-IP instead of $1/\sqrt{m}$ for μP, that is they are downscaled by $1/\sqrt{m}$ compared to μP. In Section 2.4.2, we show that there is also a close relationship between μP and IP with large initial learning rates.

We give below an intuitive explanation for the choice $c_1 = c_{L+1} = -1$ and $c_l = -2$ for $l \in [2, L]$ for the scaling of the learning rates in Naive-IP. For $l \in [2, L]$, we have $h_t^l = m^{-1}(w^l(t)x_t^{l-1} + b^l(t))$, so that $\nabla_{w^l} f_t(\xi_t) = m^{-1}(\nabla_{h^l} f_t(\xi_t))(x_t^{l-1})^\top$. In addition $\nabla_{w^{L+1}} f_t(\xi_t) = x_t^L/m$ and $\nabla_{w^1} f_t(\xi_t) = (\nabla_{h^1} f_t(\xi_t))(\xi_t)^\top$. So for one step of SGD:

$$\begin{aligned}
\Delta W^1(t+1)\xi_{t+1} &= -\eta \partial_2 \ell(y_t, f_t(\xi_t))({\xi_t}^\top \xi_{t+1}) m^{-(1+c_1)}(m\nabla_{h^1} f_t(\xi_t)).\\
\Delta W^l(t+1)x_{t+1}^{l-1} &= -\eta \partial_2 \ell(y_t, f_t(\xi_t)) m^{-(2+c_l)} \frac{(x_t^{l-1})^\top x_{t+1}^{l-1}}{m}(m\nabla_{h^l} f_t(\xi_t)), \quad \text{for } l \in [2, L]\\
(\Delta W^{L+1}(t+1))^\top x_{t+1}^L &= -\eta \partial_2 \ell(y_t, f_t(\xi_t)) m^{-(1+c_{L+1})} \frac{(x_t^L)^\top x_{t+1}^L}{m}.
\end{aligned} \tag{2.3}$$

In addition, from the equations of backpropagation, we get

$$\nabla_{h_t^L} f_t(\xi_t) = \frac{1}{m} w^{L+1}(t) \odot \sigma'(h_t^L) \quad \text{and} \quad \nabla_{h^l} f_t(\xi_t) = \frac{(w^l(t))^\top \nabla_{h_t^{l+1}} f_t(\xi_t)}{m} \odot \sigma'(h_t^l),$$

for $l \in [1, L-1]$, so that, by a simple induction, $\nabla_{h^l} f_t(\xi_t) = O(1/m)$ for $l \in [1, L]$. In addition, the averaged inner products $(x_t^{l-1})^\top x_{t+1}^{l-1}/m$ in Equation (2.3) converge as $m \to \infty$. This point is somewhat technical and is handled within the framework of the Tensor Program. The choice of c_l in Naive-IP thus ensures that the updates

are $O(1)$ when m goes to infinity. This is a desirable behaviour as it implies there is no explosion of the (pre-)activations as $m \to \infty$. However, as shown in the following Section 2.3, those updates are **not** in $\Theta(1)$ and actually converge to 0 as $m \to \infty$. As discussed below, this is essentially due to the scales of $\nabla_{h^l} f_t(\xi_t)$ becoming increasingly smaller as we go deeper in the network (from $l = L$ to $l = 1$.)

We conclude this section by giving the definition of a training routine which consists in the combination of the base learning rate, the sequence of training samples and a loss function:

Definition 2.2.2 (Training routine)**.** A training routine is the list consisting of the base learning rate $\eta > 0$, $(a_l, c_l)_{l\in[L+1]}$ in the ac-parameterization, the loss ℓ and the sequence of training samples $(\xi_0, y_0), \ldots, (\xi_{T-1}, y_{T-1})$ used to train a network for T steps.

2.3 . Deep Networks with Naive Integrable Parameterization are Trivial

In this section, we point out that, in the wide limit, neural networks in the Naive-IP remain at their initial value. We then prove that no choice for the learning rates exponents $(c_l)_{l\in[L+1]}$ which is constant in time can induce non-degenerate learning.

2.3.1 . No learning in Deep Networks with Naive Integrable Parameterization

To start with, we show that the functions learned by networks with more than four layers in the naive integrable parameterization, as described in prior work (Araújo et al., 2019; Rotskoff and Vanden-Eijnden, 2019; Fang et al., 2020; Nguyen and Pham, 2020; Weinan and Wojtowytsch, 2020; Sirignano and Spiliopoulos, 2021), remain at their value at initialization in the infinite-width limit: they are identically equal to zero at any time step. Our proof of this result is based on the Tensor Program framework (Yang, 2020b; Yang and Hu, 2021), which requires some regularity assumptions on the activation function.

Definition 2.3.1. (Pseudo-Lipschitz functions). A function $\psi : \mathbb{R}^k \to \mathbb{R}$ is pseudo-Lipschitz of degree $p > 0$ if there exists a constant $K > 0$, such that, for any $x, y \in \mathbb{R}^k$,

$$|\psi(x) - \psi(y)| \leq K||x - y|| \left(1 + \sum_{r=1}^{k} |x_r|^p + \sum_{r=1}^{k} |y_r|^p\right).$$

A function is pseudo-Lipschitz, if it is pseudo-Lipschitz of degree p for some $p > 0$.

In particular, functions with polynomially bounded weak derivatives are pseudo-Lipschitz. In the next proposition, we require the activation function σ and its derivative to be pseudo-Lipschitz.

Assumption 2 (Smooth activation)**.** The activation function σ is differentiable and both σ and its derivative σ' are pseudo-Lipschitz and not identically zero.

Remark. Note that all common activation functions which are smooth (*e.g.*, ELU, GeLU, tanh, sigmoid) satisfy Assumption 2.

Proposition 2.3.1 (Naive-IP is trivial)**.** Let $L \geq 3$ and consider the naive integrable parameterization of a network with L-hidden layers, and an activation function satisfying Assumption 2 and $\sigma(0) = 0$. Then, for any training routine which has a loss satisfying Assumption 1, the function learned by SGD remains at its value at initialization in the infinite-width limit:

$$\forall\, t \geq 0, \quad \forall\, \xi \in \mathbb{R}^d, \quad \lim_{m\to\infty} f_t(\xi) = \lim_{m\to\infty} f_0(\xi) = 0 \quad \text{almost surely.}$$

Remark.

1. In the above statement, "almost surely" is relative to the randomness of the initialization.

2. The smoothness Assumption 2 on σ is met by common activation functions such as GeLU (Hendrycks and Gimpel, 2016), ELU (Clevert et al., 2016), tanh and the sigmoid activations, but it excludes ReLU and all the other variants of Leaky ReLU. This assumption is required to apply (Yang and Hu, 2021, Theorem 7.4) (which we recall in Appendix A.2.2) which is the main theoretical result of the Tensor Program series (Yang, 2019, 2020a,b; Yang and Hu, 2021), but the result is likely to hold with weaker assumptions, as observed numerically in Section 2.6, and we leave this for future work.

3. The assumption $\sigma(0) = 0$ is met by the activation functions mentioned above (except the sigmoid) and is necessary to prove that the network does not move at any layer. Without this assumption, learning is degenerate but not trivial at all layers. It is trivial at step $t = 1$ at all layers except the last two: the coordinates of h_1^L and $f_1(\xi)$ converge, with m, to quantities which are not 0 but which are independent of the input ξ to the network, similarly to the effect described in Section 2.5.2.

The proof of Proposition 2.3.1, presented in Appendix A.4, proceeds by induction over t to show that the forward and backward passes vanish at any time step. For any time t, we proceed again by induction over l (from $l = 1$ to $l = L + 1$ for the forward pass and from $l = L + 1$ to $l = 1$ for the backward pass) to prove this vanishing occurs given the magnitudes of the previous forward and backward passes.

The informal idea of the proof is the following: essentially, the multiplications of the activation vectors by $m^{-1/2}U^l$ yield vectors whose coordinates are distributed as a Gaussian with finite variance as $m \to \infty$ for $l \geq 2$ (see Appendix A.2.1 for more details). At initialization, since $w^l(0) = m^{-1}U^l$ for $l \geq 2$ for IPs, the coordinates of h_0^l converge towards 0 as fast as $m^{-1/2}$ and that of x_0^l towards $\sigma(0)$ for σ continuous at 0. For the same reasons, $f_0(\xi_0)$ converges to 0. In the first backward pass, multiplications by $(W^l(0))^\top$ also yield vectors whose coordinates are in $O(m^{-1/2})$. In contrast to the forward pass, these scales propagate from $l = L$ to $l = 1$ and thus compound with depth, and since the last layer's gradient x_0^L/m is in $O(m^{-1})$, all the gradients' coordinates vanish as $m \to \infty$ and there is no learning. This reasoning can be repeated at later time steps as there are no correlations between the initial weight matrices and the vectors they multiply because of the degeneracy of the (pre)-activations (their coordinates become equal to the constant $\sigma(0)$ as $m \to \infty$). Those informal calculations are made rigorous by the Tensor Program.

Proposition 2.3.1 shows that the parameters of neural networks in the integrable parameterization are stuck in a stationary point of the objective function in the infinite-width limit, and no learning occurs. It might appear obvious that using larger learning rates to correct the scale with m of the weight updates can avoid this pitfall, but as discussed in the following Section 2.3.2—where we study which choices of learning rates can lead to stable learning with homogeneous activation functions—the issue is more subtle.

2.3.2 . No stable learning with learning rates constant over time

As m grows, to compensate the vanishing gradients in the first SGD step, one can use larger learning rates than in the Naive-IP. Yet, as explained below, exponents $(c_l)_{l\in[L+1]}$ for the learning rates which allow to escape the stationary point at initialization will induce an explosion of the pre-activations, if the same values of the exponents are used in the subsequent gradient steps. Indeed, the next informal statement of Theorem 2.3.2 shows that, with IPs, one cannot have non-trivial and stable learning with learning rate scales c_l constant in time.

Theorem 2.3.1 (Informal)**.** *Consider an L-hidden layer fully-connected neural network with $L \geq 3$ in the integrable parameterization. Assume that the contributions of the first and second updates $\Delta W^l(1)x_1^{l-1}$ and $\Delta W^l(2)x_2^{l-1}$ are non-vanishing and non-exploding with m at every layer l. Then, the learning rates scales c_l cannot have the same value at $t = 0$ and $t = 1$.*

In a nutshell, one needs large learning rates to escape the initial stationary point, but keeping those initial values at later time steps would make the pre-activations blow-up as $m \to \infty$. The formal version of the previous Theorem 2.3.1 is given in Theorem 2.3.2 below. For this formal statement, we introduce some definitions and assumptions.

Assumption 3 (Smooth non-negative homogeneous activation)**.** The activation function σ is non-negative, not identically zero and it is positively p- homogeneous with $p \geq 2$, *i.e.*, $\sigma(\lambda z) = \lambda^p \sigma(z)$ for any $\lambda > 0$ and $z \in \mathbb{R}$. Additionally, σ has faster growth on the positive part of the real line: $\exists z > 0 \; s.t. \; \sigma(z) > \sigma(-z)$.

Remark.

1. While the homogeneity assumption is core to the calculation of scales with integrable parameterization, the fact that $p \geq 2$, and that σ is non-negative and has faster growth on the positive part of the real line are simply here to avoid cumbersome technical difficulties in the proofs. It is clear that ReLU^p satisfies Assumption 3 for any $p \geq 2$.

2. With the assumption that $p \geq 2$, σ also satisfies Assumption 2, so that the rules of the Tensor Program can be applied.

3. While leveraging homogeneity is natural to understand the magnitudes of the forward and backward passes using the Tensor Program it might still seem unpractical as most commonly used activation functions are not positively homogeneous (except for ReLU which is positively 1- homogeneous). Yet, we explain in the following Section 2.3.3 how—under certain assumptions met by common choices of activation functions such as GeLU, ELU or tanh—IPs induce a similar behaviour *for the first update* as with homogeneity due to the vanishing of the first forward pass as presented in Section 2.3.1. Therefore, the magnitudes described below with homogeneity turn out to be also valid with those assumptions for non-homogeneous activation functions, albeit with $p = 1$.

Definition 2.3.2 (Scales of first updates with homogeneity)**.** Let $p > 0$. We define the following exponents:

$$\gamma_1(p) = \gamma_{L+1}(p) = -\frac{1}{2}\left(1 + \sum_{k=0}^{L-1} p^k\right),$$
$$\text{and} \quad \gamma_l(p) = -1 - \frac{1}{2}\sum_{k=0}^{L-1} p^k, \quad \text{for } l \in [2, L].$$

Theorem 2.3.2 (Formal version)**.** *Consider an L-hidden layer fully-connected neural network with $L \geq 3$ in the integrable parameterization, and with no bias terms, except for the first layer. Assume that the activation function σ satisfies Assumption 3, the loss ℓ satisfies Assumption 1 and that $\lim_{m\to\infty} \partial_2 \ell(y_0, f_0(\xi_0)) \neq 0$, and $\lim_{m\to\infty} \partial_2 \ell(y_1, f_1(\xi_1)) \neq 0$ almost surely. Assume further that $\xi_0, \xi_1, \xi_2 \in \mathbb{R}^d$ are all distinct vectors such that $\xi_0^\top \xi_1 \neq 0$ and $\xi_1^\top \xi_2 \neq 0$. Finally assume that:*

$$\begin{cases} \frac{1}{m}||\Delta W^l(1) x_1^{l-1}||^2 = \Theta(1), & l \in [1, L] \\ \left(\Delta W^{L+1}(1)\right)^\top x_1^L = \Theta(1) \end{cases} \tag{2.4}$$

and

$$\begin{cases} \frac{1}{m}||\Delta W^l(2)x_2^{l-1}||^2 = \Theta(1), \quad l \in [1, L] \\ (\Delta W^{L+1}(2))^\top x_2^L = \Theta(1). \end{cases} \tag{2.5}$$

Then, one necessarily has that:

(i) at $t = 0$, $c_l = \gamma_l(p)$ *for any* $l \in [1, L+1]$ *(see Definition 2.3.2),*

(ii) at $t = 1$, $c_1 = c_{L+1} = -1$, *and* $c_l = -2$ *for* $l \in [2, L]$.

Let us comment briefly on the hypotheses of Theorem 2.3.2. The proof of Theorem 2.3.2 relies on an analysis of the SGD steps involving both (Yang and Hu, 2021, Theorem 7.4) and the homogeneity property of the activation function. The requirement that $p \geq 2$ allows to satisfy the smoothness assumption of (Yang and Hu, 2021, Theorem 7.4) and the removal of the bias terms allows to fully exploit homogeneity. In Section 2.6, we numerically check that the result still holds with $\sigma = \text{ReLU}$, which is $p = 1$ homogeneous. The corresponding scales for the learning rates in the ReLU case are $\gamma_1(1) = -(L+1)/2$, $\gamma_l(1) = -(L+2)/2$ and $\gamma_{L+1}(1) = -(L+1)/2$.

We give below an informal explanation for the values of the learning rates appearing in Theorem 2.3.2 in the case of a positively 1-homogeneous activation function. As previously mentioned in Section 2.3.1, each multiplication by $W^l(0) = m^{-1}U^l$ or its transpose yields a factor in $m^{-1/2}$ for $l \geq 2$. Because of the homogeneity property, this scale propagates from layer to layer starting from layer 2, and the coordinates of h_0^l and x_0^l are thus in $\Theta(m^{-(l-1)/2})$ for $l \in [1, L]$. For the backward pass, the first gradient $\nabla_{x^L} f_0(\xi_0) = U^{L+1}/m$ has coordinates in $\Theta(m^{-1})$, and, as already discussed in Section 2.3.1, from $l = L$ to $l = 2$, each multiplication by $(W^l(0))^\top$ yields an additional factor in $m^{-1/2}$ and those compound with depth so that the coordinates of $\nabla_{h^l} f_0(\xi_0)$ are in $\Theta(m^{-1}m^{-(L-l)/2})$. Therefore, calling $\tilde{x}_0^l := m^{(l-1)/2}x_0^l$, and $d\tilde{h}_0^l := m^{1+(L-l)/2}\nabla_{h^l} f_0(\xi_0)$, we have after the first weight update

$$\begin{aligned} \Delta W^1(1)\xi_1 &= -\eta \partial_2 \ell(y_0, f_t(\xi_0))(\xi_0{}^\top \xi_1) m^{-c_1} m^{-(L+1)/2} d\tilde{h}_0^1, \\ \Delta W^l(1)x_1^{l-1} &= -\eta \partial_2 \ell(y_0, f_0(\xi_0)) m^{-c_l} m^{-2} m^{-(L-l)/2-(l-2)/2} \frac{(\tilde{x}_0^{l-1})^\top x_1^{l-1}}{m} d\tilde{h}_0^l, \quad l \in [2, L], \\ (\Delta W^{L+1}(1))^\top x_1^L &= -\eta \partial_2 \ell(y_0, f_0(\xi_0)) m^{-c_{L+1}} m^{-1} m^{-(L-1)/2} \frac{(\tilde{x}_0^L)^\top x_1^L}{m}. \end{aligned}$$

Since $d\tilde{h}_0^l$ and $\tilde{x}_0^l$ have coordinates in $\Theta(1)$ by design, and since averaged inner products of the type $(\tilde{x}_0^{l-1})^\top x_1^{l-1}/m$ converge to finite expectations (by the rules of the Tensor Program, see Yang and Hu, 2021, Theorem 7.4), we see that the choice $c_1 = -(L+1)/2$, $c_l = -(L+2)/2$ for $l \in [2, L]$, and $c_{L+1} = -(L+1)/2$ is the only way to ensure that the updates induce contributions which have coordinates

in $\Theta(1)$ at $t = 1$. Given this choice for the learning rate scales $c_1, \dots, c_{L+1}$ at $t = 0$, we readily get that the coordinates of h_1^l and x_1^l are in $\Theta(1)$ because the contributions $W^l(0)x_1^{l-1}$ have coordinates in $O(m^{-1/2})$ for intermediate layers, and in $O(1)$ for the input and output layers. From the Equations (2.3) with $t = 1$, we see that for the second gradient step, $m\nabla_{h^l} f_1(\xi_1)$ has coordinates in $\Theta(1)$ because the multiplications by $(W^l(1))^\top$ do not yield a factor in $m^{-1/2}$ due to the scale correction introduced in the first update. At $t = 1$, this leads to the choice $c_1 = c_{L+1} = -1$, and $c_l = -2$ for $l \in [2, L]$, in order to have update contributions with coordinates in $\Theta(1)$ at $t = 2$. These informal calculations are made rigorous in the proof of Theorem 2.3.2 using the Tensor Program (Yang, 2020b).

2.3.3 . Recovering results without homogeneity: linearization of the first step

As we have described above, the crux of the matter for IPs is understanding the magnitudes of the first forward and backward passes in order to escape the initial stationary point and induce stable learning. With IPs, it is in fact possible to analyze rigourously those as with homogeneity: indeed, under a set of less restrictive assumptions, the fact that the initial forward pass converges to 0 as m grows large allows to effectively linearize σ around 0 *in the first gradient step*, thereby recovering a similar effect as a 1-homogeneous activation function. We consider the following assumption on σ:

Assumption 4 (Linearization of σ)**.** The activation function σ is continuously differentiable with $\sigma(0) = 0$, $\sigma'(0) \neq 0$ and there is an $M > 0$ such that for any $z \in \mathbb{R}$, $|\sigma(z) - \sigma'(0)z| \leq \frac{M}{2}z^2$.

Remark.

1. Note that the last inequality is satisfied as soon as σ'' is bounded—which is the case for $\sigma \in \{\text{GeLU}, \tanh\}$—and that ELU also satisfies Assumption 4 so that the latter is satisfied by commonly used activation functions, and we recall that those activation functions also satisfy Assumption 2.

2. Understanding the magnitudes of the first gradients with $\sigma = \text{GeLU}$ theoretically would allow extending this analysis to ReLU (which is non-smooth) as explained in (Yang and Hu, 2021) since $\text{ReLU}(z) = \lim_{\alpha\to\infty} \text{GeLU}(\alpha z)/\alpha$. However, taking this limit within the Tensor Program adds cumbersome technical work and we leave this for future work.

Intuitive idea for the linearization. Let us explain briefly why this linearization occurs and how it allows to obtain the scales of the first forward and backward passes as with a 1-homogeneous function. Consider the second layer preactivation at initialization $h_0^2 = m^{-1/2}\hat{W}^2 x_0^1$ which has coordinates in $\Theta(m^{-1/2})$ which thus converge to 0 as $m \to \infty$. With σ satisfying Assumption 4 for the activation $x_0^2 = \sigma(h_0^2)$ we get $x_0^2 \simeq \sigma'(0)h_0^2$ which also has coordinates in $\Theta(m^{-1/2})$

and the upper bound in Assumption 4 allows to make this rigorous in the limit $m \to \infty$. By induction it is clear that h_0^l, x_0^l have coordinates in $\Theta(m^{-(l-1)/2})$. Note that the calculations above suggest setting the standard deviation of the Gaussians to $|\sigma'(0)|^{-1}$ at initialization to avoid issues with depth (see more details in Remark A.7.2) which is what we do in the numerical experiments of Section 2.6.

For the backward pass, $\nabla_{x_0^L} f_0(\xi) = m^{-1} U^{L+1}$ and $\nabla_{h_0^L} f_0(\xi) = \nabla_{x_0^L} f_0(\xi) \odot \sigma'(h^L) \simeq m^{-1}\sigma'(0) U^{L+1}$ which have coordinates in $\Theta(m^{-1})$, and $\nabla_{w^{L+1}(0)} f_0(\xi) = m^{-1} x_0^L$ has coordinates in $\Theta(m^{-(L+1)/2})$. The successive multiplications by $(W^l(0))^\top = m^{-1/2}(\hat{W}^l)^\top$ in the backward pass each contribute to an additional factor $m^{-1/2}$ and the fact that the forward pass vanishes yields $\nabla_{x_0^l} f_0(\xi), \nabla_{h_0^l} f_0(\xi) = \Theta(m^{-1} m^{-(L-l)/2})$ and thus, since $\nabla_{w^l(0)} f_0(\xi) = m^{-1} \nabla_{h_0^l} f_0(\xi) (x_0^{l-1})^\top$, the first weight gradients $\nabla_{w^l(0)} f_0(\xi)$ have coordinates which scale in $\Theta(m^{-1} m^{-(L-l)/2} m^{-1} m^{-(l-2)/2}) = \Theta(m^{-(L+2)/2})$ for $l \in [2, L]$. Finally, for $l = 1$, one has $\nabla_{w^1(0)} f_0(\xi) = \nabla_{h^1(0)} f_0(\xi) \xi^\top$ whose coordinates are in $\Theta(m^{-1} m^{-(L-1)/2}) = \Theta(m^{-(L+1)/2})$. This suggests correcting the initial gradients by the scales $m^{(L+1)/2}$ for the first and last layers and $m^{(L+2)/2}$ for intermediate layers in order to obtain gradients which are non-vanishing and non-exploding.

Formalization and differences with homogeneity. We formalize the intuitive ideas presented above in Appendix A.6 where we rigorously derive the linearization and the scale of the first update under Assumption 4. It is noteworthy that this linearization of the first step induces a behaviour which is similar to but different from homogeneity. Indeed, the linearization, along with the fact that $\sigma'(h_0^l)$ converges to the constant $\sigma'(0)$ removes the correlation usually introduced by the first update in some of the intermediate layers which results in a different behaviour. This technicality, along with the fact that we have no homogeneity or linearization after $t = 0$ under Assumption 4, make it difficult to adapt the proofs of homogeneity to this setting beyond the analysis of the first forward and backward passes. In particular, the equivalence between IP-LLR and μP—as described in Section 2.4.2—is not exact this setting. Because of the reasons mentioned above, the setting where linearization occurs would require a separate study and we leave this for future work.

2.4 . Large Initial Learning Rates Induce Learning

In this section, we show that with positively homogeneous activation functions, using large initial learning rates (polynomial in m) allows the network to escape from the initial stationary point and to initiate a non-trivial training phase in the infinite-width limit. Because we use the homogeneity property extensively for our results, in all this section, as in Section 2.3.2, we consider a version of integrable parameterizations where the bias terms are removed except for the first layer.

As observed in Section 2.3.2, beyond the fact that IPs require large learning rates (for the first gradient step) to be trained, one crucial characteristic of the degeneracy in IPs is that no choice of learning rate scales (c_l) which are constant in time can induce a favorable learning behavior: one has to first use large learning rates to escape the stationary point at initialization $(t = 0)$ and then revert to the Naive-IP learning rates for $t \geq 1$ to induce stable learning.

Definition 2.4.1 (IP with large initial learning rates)**.** Let σ be a positively p-homogeneous activation function with $p > 0$. We define *the integrable parameterization with large initial learning rates* (IP-LLR) as the integrable parameterization of an L-hidden layer fully connected-network with activation σ such that:

(i) At $t = 0$: $c_l = \gamma_l(p)$, for $l \in [1, L + 1]$;

(ii) At $t \geq 1$: $c_1 = c_{L+1} = -1$ and $c_l = -2$, for $l \in [2, L]$,

where the values of the $\gamma_l(p)$ are given in Definition 2.3.2.

Remark.

1. The definition means that $\Delta w^l(1) = -\eta m^{-\gamma_l(p)} \nabla_{w^l} \ell(y_0, f_0(\xi_0))$ for the first weight update after the forward-backward pass at time $t = 0$, and for $t \geq 1$, the $(t+1)$-th weight update is $\Delta w^l(t+1) = -\eta m^{-2} \nabla_{w^l} \ell(y_t, f_t(\xi_t))$ for $l \in [2, L]$, and $\Delta w^1(t + 1) = -\eta m^{-1} \nabla_{w^1} \ell(y_t, f_t(\xi_t))$, $\Delta w^{L+1}(t + 1) = -\eta m^{-1} \nabla_{w^{L+1}} \ell(y_t, f_t(\xi_t))$ after the forward-backward pass at time t.

2. We give the definition with an arbitrary degree of homogeneity p (the values of the $\gamma_l(p)$ are given in Definition 2.3.2) as for some theorems where we use the Tensor Program for the proof, we need sufficient smoothness of the activation function, which is achieved only when $p \geq 2$, but we always use $\sigma = \text{ReLU}$ (which corresponds to $p = 1$) in our informal derivations and numerical experiments. Note that since the values of $c_1, \ldots, c_{L+1}$ at $t = 0$ depend on p, the definition of an IP-LLR parameterization also implicitly depends on the degree of homogeneity p.

3. Since $a_1 = 0$ for IPs, we leverage the homogeneity property only for layers $l \in [2, L]$ (see Appendix A.7.2 for more details), so that we might as well assume $L \geq 2$ whenever we study IP-LLR.

2.4.1 . Non-trivial and Stable Learning for Integrable Parameterizations

Theorem 2.4.1 (Non-trivial and non-exploding learning with IP-LLR)**.** *Consider the IP-LLR parameterization of an L-hidden layer neural network with no bias*

terms, except for the first layer, and with an activation function σ satisfying Assumption 3 and a loss function ℓ satisfying Assumption 1. Let $\xi \in \mathbb{R}^d$ be an input to the network, and assume $\partial_2 \ell(y_0, 0) \neq 0$. Then, one has:

$$
\begin{aligned}
&(i)\ \ f_0(\xi) \xrightarrow[m\to\infty]{a.s.} 0. \\
&(ii)\ \ f_1(\xi) \xrightarrow[m\to\infty]{a.s.} \mathring{f}_1(\xi), \quad 0 < |\mathring{f}_1(\xi)| < \infty \ a.s. \\
&(iii)\ \ f_2(\xi) \xrightarrow[m\to\infty]{a.s.} \mathring{f}_2(\xi), \quad |\mathring{f}_2(\xi)| < \infty \ a.s.
\end{aligned}
$$

Theorem 2.4.1 essentially shows that IP-LLR is able to induce both a non-trivial dynamic since $\lim_{m\to\infty} (f_1(\xi) - f_0(\xi)) \neq 0$ as well as a stable one since the prediction function remains bounded even after two gradient steps in the infinite-width limit.

Remark.

1. We show in our numerical experiments (see Section 2.6) that with $\sigma =$ ReLU (*i.e.*, $p = 1$), the choice of learning rates for IP-LLR is indeed able to induce learning for networks deeper than four layers without creating instabilities.

2. For positively p-homogeneous activations with $p \geq 2$, we have $\sigma'(0) = 0$ and the behavior of the network is inherently different from that of a network where the first forward pass can effectively be linearized (the setting described in Section 2.3.3). This difference appears in the numerical experiments presented in Section 2.6 where we also discuss the reasons for such a qualitatively different behavior.

3. In IP-LLR, the initial gradient direction will be determined by the first sample (ξ_0, y_0) fed to the network. To avoid giving too much importance to a single sample, one can in practice average the gradients over a batch of many training samples instead, which is what we do in our numerical experiments in Section 2.6.

4. The most subtle and technical part of the theorem is the fact that $|f_1(\xi)| > 0$ in the second point. That the movement in the prediction function is of order 1 is guaranteed by the choice of learning rates in IP-LLR, and there is no reason for $f_1(\xi)$ to be zero a priori but actually proving that the resulting function indeed produces an output different from 0 is challenging.

The idea of the proof essentially lies in the informal calculations of Section 2.3.2 which are made rigorous using the framework of the Tensor Program. Point (ii) stems from the fact that at $t = 1$, the output is the difference between two expectations in the limit $m \to \infty$, which can both be shown to be different from 0 and of opposite signs.

We highlight that the proof technique is much different from (Yang and Hu, 2021) as the result holds for any value of the scalar learning rate $\eta > 0$ as soon as the initial loss is different from 0 (in the limit $m \to \infty$) while most results on feature learning in (Yang and Hu, 2021) are valid only for some small value of η (*e.g.*, Definition 3.5, Theorem 3.6, Definition 5.2, Theorem 5.6). The difference mainly resides in the precise analysis we make of the first gradient step which is made possible by the non-negativity assumption on the activation function and the fact that the contributions of the initial weights vanish in IP-LLR for intermediate layers, both of which allow to analyze the expectations resulting from the Tensor Program in an inductive manner.

2.4.2 . IP-LLR is a Modified μP

In this section, we analyze the behavior of IP-LLR more in detail and show that this model is actually equivalent to a modification of μP where the initial weights are removed from the first weight update for all of the intermediate layers. Said differently, IP-LLR is the same as using μP but initializing the intermediate layers with the first update with large initial learning rates instead of the Gaussian initialization of μP. We first show an equivalence at finite-width in Section 2.4.2 with mild assumptions, and then extend those results to the infinite-width limit in Section 2.4.2 with slightly more restrictive assumptions on the activation function σ. Since we study the IP-LLR parameterization, we consider positively p-homogeneous activation functions, and only the degree of homogeneity allowed will vary between Sections 2.4.2 and 2.4.2. In short, the main idea behind this equivalence is that since IP-LLR and μP are both designed to have maximal update contributions at $t = 0$, they will induce the same update at initialization, and the only difference at later time steps is that the initial weights of IP-LLR contribute vanishingly to the pre-activations whereas those of μP contribute in $\Theta(1)$. In this regard, it is not a surprise that the learning dynamics of IP-LLR and μP are closely related. Nevertheless, we believe this novel connection is worth exploring and we devote this section to highlighting in detail the precise link between IP-LLR and μP.

Finite-Width Equivalence

As explained in Section 2.2.2 in the examples of ac-parameterizations, from the point of view of abc-parameterizations (see Yang and Hu, 2021), both μP and Naive-IP follow the same training procedure for the effective weights W^l, the only difference being the standard deviation at initialization which is downscaled by $1/\sqrt{m}$ for Naive-IP compared to μP. We detail this connection in this section.

Recall that for μP one has $W^1_{\mu\text{P}}(0) = U^1$, $W^l_{\mu\text{P}}(0) = m^{-1/2}U^l$ for $l \in [2, L]$, and $W^{L+1}_{\mu\text{P}}(0) = m^{-1}U^{L+1}$ whereas for any integrable parameterization, one has $W^1_{\text{IP}}(0) = U^1$, $W^l_{\text{IP}}(0) = m^{-1}U^l$ for $l \in [2, L+1]$. Consider the following *hybrid parameterization* (HP) which consists in training with the maximal update parameterization μP all along, but simply replacing, for all intermediate layers $l \in$

$[2, L]$, the first update $W^l(1) = W^l(0) + \Delta W^l(1)$ by $W^l(1) = m^{-1}U^l + \Delta W^l(1)$. In other words, this simply consists in using the weight pre-factors of μP for the intermediate layers in the initial forward and backward passes, and then using the pre-factors from IP for the initial weights of the intermediate layers in any subsequent update.

Proposition 2.4.1 (Finite width equivalence between IP-LLR and HP)**.** Consider the IP-LLR and HP parameterizations with a p-homogeneous activation function σ with $p \geq 1$ and without any bias term except at the first layer. Let us sub/super-script the variables of each model with IP and HP respectively. Assume the full sequence of training samples $(\xi_0, y_0), \ldots, (\xi_s, y_s), \ldots$ and the loss ℓ are the same for both parameterizations. Assume further that $\partial_2 \ell(y_0, f_0^{HP}(\xi_0)) \neq 0$, and denote by η the base learning rate of the IP-LLR parameterization. Finally consider the following schedule for the base learning rate of HP:

$$\eta_{\mathsf{HP}}(0) = \frac{\partial_2 \ell(y_0, f_0^{IP}(\xi_0))}{\partial_2 \ell(y_0, f_0^{HP}(\xi_0))} \eta,$$
$$\eta_{\mathsf{HP}}(s) = \eta, \qquad s \geq 1.$$

Then one has:

$$\forall t \geq 1, \quad \forall \xi \in \mathbb{R}^d, \quad f_t^{\mathsf{HP}}(\xi) = f_t^{\mathsf{IP}}(\xi).$$

The proof, presented in Appendix A.11.1, simply shows inductively that the effective weight matrices for both models are equal for all $t \geq 1$. Since the Tensor Program is not needed here as we consider only finite-width networks, we can work with any positively homogeneous activation function (not necessarily smooth, so that $p = 1$ is not precluded).

Infinite-Width Equivalence

Similarly to HP, we now consider another hybrid parameterization where the initial weights $W^l(0)$ are simply replaced by 0 in the first update of the intermediate layers. We thus consider the following *hybrid parameterization with zero re-initialization* (HPZ): we train with μP all along, but simply replace, for all intermediate layers $l \in [2, L]$, the first update $W^l(1) = W^l(0) + \Delta W^l(1)$ by $W^l(1) = \Delta W^l(1)$. In other words, this is the same as initializing the intermediate layers of μP with $\Delta W^l(1)$ (where the update is computed with either IP-LLR or μPas they are the same *for positively homogeneous activations*). This can also be seen as using $W^l(0)$ the weight pre-factors of μP for the intermediate layers in the initial forward and backward passes, and then forgetting the contribution of the initial weights of the intermediate layers in any subsequent update. As already discussed in Section 2.3.1, the contribution of the initial weights of the intermediate layers $m^{-1}U^l$ vanishes as $m \to \infty$ for IP, so that HPZ is simply the infinite-width equivalent of HP.

Theorem 2.4.2 (HPZ and IP-LLR are equivalent)**.** *Consider the IP-LLR and HPZ parameterizations with a p-homogeneous activation function σ with $p \geq 2$, and with no bias terms except at the first layer. Let us sub/super-script the variables of each models with IP and HPZ respectively. Assume that the training routine is the same for both parameterizations, and assume further that the loss ℓ satisfies Assumption 1. Then, one has:*

$$\forall t \geq 0, \quad \forall \xi \in \mathbb{R}^d, \quad \lim_{m \to \infty} f_t^{HPZ}(\xi) = \lim_{m \to \infty} f_t^{IP}(\xi) \quad \textit{almost surely}.$$

The proof, presented in Appendix A.11.2, proceeds by induction to show that the quantities appearing in the forward and backward passes at every layer are the same for both models at every time step in the infinite-width limit. We use the Tensor Program framework for this proof so we need smoothness of σ ($p \geq 2$) for this result.

In essence, Theorem 2.4.2 shows that the IP-LLR parameterization is equivalent to μP where we simply forget the initialization after the first forward and backward passes. Said differently, IP-LLR is the same as μP, except that IP-LLR initializes the weights of the intermediate layers $l \in [2, L]$ at $t = 1$ with $W^l(1) = \Delta W^l(1)$, *i.e.*, with the first update computed after the first forward-backward pass. It is not entirely clear whether forgetting the initial weights in one step is beneficial or detrimental to learning. On the one hand, it would seem like forgetting the random initialization could make the network learn faster and be more robust to perturbations (but this is only speculative at this point, and we leave this open for future work), on the other hand the large rank of the initial weight matrices with i.i.d. Gaussian entries might increase the stability of the training dynamics. In other words, while the randomness from initialization propagates to every layer at every times step for μP, it is forgotten in one step of SGD for IP-LLR in the infinite-width limit. In Section 2.6 we explore the performance of both models numerically and show that IP-LLR appears to be a valid way of training IPs as it seems to perform on par with μP which we know has maximal update properties.

Another distinguishing factor between IP-LLR and μP is that for any intermediate layer $l \in [2, L]$, while $(W_{jq}^l(t) - W_{jq}^l(0))/W_{jq}^l(0) = \Theta(m^{-1/2})$ for μP, so that the effective weights only move infinitesimally (in the infinite-width limit) relatively to their initial values, we have $(W_{jq}^l(t) - W_{jq}^l(0))/W_{jq}^l(0) = \Theta(1)$ for IP-LLR so that the effective weights actually move in the infinite-width limit (see more details in Remark A.7.2). The latter behaviour is the one observed empirically for neural networks used in practice (even for wide models) with normalization layers (such as BatchNorm Ioffe and Szegedy, 2015 or LayerNorm Ba et al., 2016) but we leave the connection between IP-LLR and the effect of normalization layers for future work.

2.5 . Alternative Methods for Escaping the Initial Stationary Point

As discussed in Section 2.4, using large initial learning rates in combination with a positively homogeneous activation function allows escaping the initial stationary point and induces stable learning. In this section, we present two alternatives to escape this initial stationary point and discuss the properties of the resulting models. This is not by any means an exhaustive list of alternatives to escape the initial stationary point, but they are two natural examples (one of which appears in (Nguyen and Pham, 2020) and we recover their result in a slightly different setting) and we show that they result in degenerate behaviours, so that it appears that using large initial learning rates is the only valid way to get non-degenerate behaviour for i.i.d. IPs. In contrast to the setting of Section 2.4, in all this section, we consider IPs with bias terms at every layer.

A first alternative to escape the initial stationary point, which we discuss in Section 2.5.1, is to simply initialize the weight matrices with i.i.d. Gaussian distributions which are not centered around 0, as suggested by Nguyen and Pham (2020). This method is able to escape the stationary point without large initial learning rates and without any homogeneity assumption on the activation function. It turns out that the computations in that setting are well described within the Tensor Program framework and we show that, as highlighted in (Nguyen and Pham, 2020, Corollary 37), a collapse phenomenon occurs, where all the individual entries in the weight matrix of an intermediate layer evolve by the same deterministic quantity in the infinite-width limit. Using the Tensor Program, we recover this result in the context of SGD on with a loss computed on mini-batches instead of gradient flow on the empirical loss.

Another natural alternative is to remove the pre-factor m^{-1} in front of the bias terms of layers $l \geq 2$. Indeed, as observed in Section 2.3.1, the vanishing of the forward pass and the weight updates in integrable parameterizations is mostly due to the multiplications by the weight matrices $m^{-1}U^l$ which results in pre-activations whose coordinates are $\Theta(m^{-1/2})$ for $l \in [2, L]$. Since the bias terms are decoupled from the input to the layer, re-scaling them appropriately avoids vanishing of the forward pass for IPs. Escaping the initial stationary point can then be achieved without any homogeneity assumption on the activation function σ. However, one issue which arises then is that the bias terms have the dominant contribution to the pre-activations, and since the input signal propagates through the network via the weight multiplications, the output of the trained network is only "weakly" dependent on its input and the training data. We now study these two alternatives in more detail.

2.5.1 . Using Non-Centered i.i.d. Initialization

In this section, we consider the following modified version of IPs which we call *IP-non-centered* : the forward pass is computed exactly as in IPs but the

weight matrices of layers $l \geq 2$ are initialized with $w^l_{jq}(0) = U^l_{jq} + u_l \sim \mathcal{N}(u_l, 1)$ i.i.d. over (j, q) with $u_l \neq 0$. This simply consists in setting $w^l(0) = U^l + u_l J$ for $l \in [2, L]$ and $w^{L+1}(0) = U^{L+1} + u_{L+1}\mathbf{1}$ where J is the square matrix full of ones (whose variable size is the same as U^2 and thus equal to m) and $\mathbf{1}$ is the vector (of variable size equal to m) full of ones. As we will see shortly, the effect of this type of initialization is similar to removing the pre-factor in m^{-1} on the bias terms in that the vanishing of the matrix multiplications $m^{-1}U^l x^{l-1}_t$ is offset by the appearance of an additional term in the expression of h^l_t whose coordinates are all equal and *depend* on the input data.

General intuition. In short, the idea is that if $W \in \mathbb{R}^{m\times m}$ is a Gaussian matrix with entries following $\mathcal{N}(0, 1)$ i.i.d.(be it in a forward or backward pass), and u a scalar different from zero, the effect of multiplying a vector $x \in \mathbb{R}^m$ by a non-centered Gaussian matrix with standard deviation $1/m$ corresponds to the result of $m^{-1}(W + uJ)x = m^{-1}Wx + m^{-1}uJx$. We have already seen that the first term converges to 0 as $m \to \infty$ while all the coordinates the second term are equal to $u\frac{1}{m}\sum_{j=1}^m x_j$, which converges by the rules of the Tensor Program to some finite expectation. Thus provides the main contribution in the multiplication $m^{-1}(W+uJ)x$ is a vector whose coordinates all converge to the same deterministic constant in the infinite-width limit. Essentially, this phenomena holds true in the forward and backward passes at any time step for IP-non-centered and causes the forward and backward passes, as well as the weight updates to collapse to deterministic constants for layers $l \in [2, L-1]$. In particular, the first forward and backward passes do not vanish in this context and the "usual" mean-field learning rates as in Naive-IP: $c_1 = c_{L+1} = -1$, and $c_l = -2$ for $l \in [2, L]$ induce learning at any time step.

We summarize this result in the following informal theorem:

Theorem 2.5.1 (Informal)**.** *Consider IP-non-centered with the Naive-IP learning rates at every time step, and let $t \geq 0$ and $\xi \in \mathbb{R}^d$ be an input to the network. Then, one has that:*

(i) for any $l \in [2, L-1]$, the coordinates of h^l_t (resp. x^l_t) all converge to the same deterministic constant,

(ii) for any $l \in [2, L-1]$, the coordinates of $m\nabla_{x^l_t} f_t(\xi_t)$ (resp. $m\nabla_{h^l_t} f_t(\xi_t)$) all converge to the same deterministic constant,

(iii) for any $l \in [3, L-1]$, the entries of $(W^l(t) - W^l(0))$ all converge to the same deterministic constant.

The rigorous version of this theorem, and its proof, formalized within the framework of the Tensor Program, are presented in Appendix A.12.2.

2.5.2 . Not Scaling the Bias Terms

In this section, we consider a version of IPs where we remove the pre-factor $1/m$ for the bias terms of layers $l \geq 2$. We thus consider the following computations in the forward pass:

$$\begin{aligned} h_t^1 &= w^1(t)\xi + b^1(t), \\ h_t^l &= \left(m^{-1} w^l(t) x_t^{l-1}\right) + b^l(t), \qquad l \in [2, L] \\ f_t(\xi) &= \left(m^{-1} (w^{L+1}(t))^\top x_t^L\right) + b^{L+1}(t), \end{aligned} \tag{2.6}$$

which in other terms simply means that $B^l(t) = b^l(t)$ for $l \in [1, L+1]$. We use the same initialization for the bias terms as in IPs: $b^l(0) = v^l$ for $l \in [1, L+1]$, where the entries of v^l are i.i.d. following $\mathcal{N}(0, 1)$. We call *IP-bias* the modified version of the integrable parameterization described by Equations (2.6).

General intuition. As in IP-non-centered, the idea is now that the non-scaled bias terms will provide the main (non-vanishing) contribution compared to the multiplication with i.i.d. Gaussian matrices scaled by $1/m$, and induce non-vanishing forward passes. However, in comparison to IP-non-centered, the backward pass still vanishes and needs to be corrected but with learning rates that are not as large as in IP-LLR in the first update because the first forward pass is of order 1. Once this is accounted for in the first weight update, the following weight updates use the same learning rate exponents as for Naive-IP: $c_1 = c_{L+1} = -1$ and $c_l = -2$ for $l \in [2, L]$. The degeneracy comes from the fact that the main contribution in the pre-activations h_t^l comes from the initial Gaussian bias term $b^l(0) = v^l$: indeed, although the weight updates have non-vanishing contribution, they are multiplied by the scalar learning rate η which tends to be less than one, and the dominant contribution is that of the initial bias term which does not depend on the data, and thus makes the output of the network only weakly data-dependent which is detrimental to the practical performance of those models as observed in the numerical experiments of Section 2.6.

Those ideas are formalized in Appendix A.12.1.

2.6 . Numerical Experiments

In this section we investigate numerically the behavior of the models previously introduced in this work, namely Naive-IP, IP-LLR, IP-bias, IP-non-centered and μP. In contrast to the theoretical analysis carried out in Sections 2.3, 2.4, and 2.5, we examine the performance of the models on a multi-class classification task (instead of a single output prediction) and we train them using mini-batch SGD (instead of single-sample SGD). In addition to these two points, we adopt the following slight modifications compared to our theoretical setting.

Standard deviation of initial weights. In our numerical experiments, we allow the initial Gaussian weight matrices U^l and vectors v^l to have entries drawn from $\mathcal{N}(0, \delta_l^2)$ where δ_l can be different from 1 for $l \in [1, L]$, but is independent of m. As hinted in Section 2.3.3 and explained more in detail in Remarks A.7.1 and A.7.2, this is to avoid issues (vanishing or explosion of the forward/backward pass) with the depth L. The value of the standard deviation for ReLU comes from the analysis in Appendix A.14. The choices of the standard deviation of the Gaussian depend on the activation function and are summarized in Table 2.1.

activation	ReLU	GeLU	ELU	tanh
init. std	$\sqrt{2}$	2	1	1

Table 2.1: Standard deviation δ_l of the initial Gaussian entries of layers $l \in [1, L]$ for different choices of activation functions.

Re-scaling the standard deviation of the first layer. All the models we consider have $a_1 = 0$ so that, as mentioned in Section 2.5.2, the coordinates of h_0^1 follow $\mathcal{N}(0, ||\xi||^2+1)$ and the variance is equal to $\sum_{k=1}^d \xi_k^2+1$. To avoid having too large a variance when the (fixed) dimension d is large, we re-scale the standard deviation of the first layer's weights and bias term at initialization by dividing it by $\sqrt{d+1}$, that is we use the Gaussian law $\mathcal{N}(0, \delta_1^2/(d+1))$ to initialize the entries of $w^1(0)$ and $b^1(0)$.

Calibrating the initial base learning rates for IP-LLR. As discussed in Section 2.4.2, IP-LLR basically amounts to training with μP but forgetting the initialization in the intermediate layers for the first update. We thus roughly have $W^l(1) \simeq \Delta W^l(1)$ for any $l \in [2, L]$, and the base learning rate η directly influences the magnitude of $\Delta W^l(1)$ and thus that of h_1^l. Typical values for the learning rates, the initial loss derivative $\partial_2 \ell(y_0, 0)$, and the averaged inner products involved in the second forward pass are rather small (*e.g.*, $\leq 10^{-1}$), and this will cause the pre-activations of the second forward pass to be of small magnitude, and this effect compounds quickly with depth as the pre-activations of layer $(l-1)$ are then multiplied by $\Delta W^l(1)$. This will in turn lead to very small values for the second weight updates $\Delta W^l(2)$ and can considerably slow down learning in practice. To overcome this issue, we simply calibrate the initial values of the base learning rates η_l of layers $l \in [2, L]$ at $t = 0$, so that the magnitude of the pre-activation of the intermediate layers in the second forward pass is equal to 1 on average over the second training batch.

Note that this calibration results in base learning rates η_l which **do not** depend on m (they do depend on L however) in the large-width limit as the coordinates of h_1^l have non-zero and finite values for large m. In contrast, this is not possible with the Naive-IP as the coordinates of h_1^l converge to zero as fast as some power

of m, which would result in the base learning rate η_l depending on m which is prohibited (by definition of the base learning rate).

All the points above can be handled within the framework of the Tensor Program, but they would unnecessarily over-complicate the analysis and the formulas, which is why we used a simpler setting in our theoretical analysis.

2.6.1 . Experimental Setup

We evaluate the performance of the different models on two datasets: MNIST[2], containing 50,000 training samples, 10,000 validation samples and 10,000 test samples, and CIFAR-10[3], 40,000 training samples, 10,000 validation samples and 10,000 test samples. Both datasets consist in a 10-class image classification task. Since we consider only fully-connected networks, we use gray-scale images which we also flatten for both datasets, which means the input dimension is $d = 28 \times 28 = 784$ for MNIST and $d = 32 \times 32 = 1024$ for CIFAR-10.

We train for 5000 SGD steps on MNIST and 6000 steps on CIFAR-10 using a wide range of values for the base learning rate of $\eta \in \mathcal{R} := \{p10^{-q} : p, q \in [1, 9] \times [2, 4]\} \cup \{0.1\}$, a batch-size $B = 512$, and the cross-entropy loss, which satisfies Assumption 1. We selected the number of steps for each dataset so as to ensure that training has converged for all models, that is the validation accuracy starts to decrease. We used with a wide range of base learning rates because different models might favor different values. For each experiment, we run $N_{\text{trials}} = 5$ trials with different random initializations. The hyperparameters are summarized in Table 2.2.

L	m	d_{MNIST}	d_{CIFAR}	ℓ	η	B	N_{trials}
5	1024	784	1024	cross-ent.	range $\mathcal{R}$	512	5

Table 2.2: Hyperparameters for training models.

2.6.2 . IP-LLR *vs.* μP

We compare the numerical performance of IP-LLR and μP on both MNIST and CIFAR-10 and summarize them in Tables 2.3 and 2.4. The performance, as measured by the accuracy on the test set, is consistent for both μP and IP-LLR. The best test accuracy for μP and IP-LLR are comparable: the former achieves 0.979 test accuracy on MNIST and 0.413 test accuracy on CIFAR-10 while the latter achieves 0.980 test accuracy on MNIST and 0.434 test accuracy on CIFAR-10.

2.6.3 . Learning is Degenerate for IP-bias and IP-non-centered

[2] http://yann.lecun.com/exdb/mnist/

[3] https://www.cs.toronto.edu/~kriz/cifar.html

activation / model / lr	ReLU	GeLU	ELU	tanh
IP-LLR	0.970	0.977	0.976	0.980
optimal η	0.1	0.009	0.002	0.03
μP	0.978	0.979	0.977	0.970
optimal η	0.08	0.03	0.1	0.1

Table 2.3: Test accuracies and optimal learning rates on MNIST for various activation functions.

activation / model / lr	ReLU	GeLU	ELU	tanh
IP-LLR	0.365	0.434	0.427	0.357
optimal η	0.09	0.009	0.005	0.003
μP	0.395	0.413	0.398	0.309
optimal η	0.1	0.003	0.04	0.08

Table 2.4: Test accuracies and optimal learning rates on CIFAR-10 for various activation functions.

In this section we show numerically that IP-non-centered and IP-bias (see Sections 2.5.1 and 2.5.2 respectively) are able to escape the initial stationary point but that the resulting dynamics do not seem effective as observed through the final test performance.

As summarized in Table 2.5, IP-non-centered and IP-bias appear to have poor test accuracy even after extensively long training comparatively with IP-LLR and μP, even with the best choice of activation function and with the optimal learning rate. Similarly, it appears that Naive-IP is not able to escape the initial stationary point even with a relative large base learning rate and with extensively long training as its test performance is barely better than random chance.

model / acc / hparams	IP-LLR	μP	IP-bias	IP-non-centered	Naive-IP
CIFAR-10 accuracy	0.434	0.413	0.320	0.173	0.118
optimal η	0.009	0.003	0.1	0.1	0.1
optimal σ	GeLU	GeLU	GeLU	ELU	ELU

Table 2.5: Test accuracies (averaged over 5 random runs) at the end of training on CIFAR-10. For each model, we show the maximum (averaged) accuracy over all activation functions and learning rates.

2.7 . Conclusion

Recent research has shown that the parameterization of a neural network has a dramatic impact on its training dynamics, and therefore, on the type of functions that it is able to learn. Until now, the parameterizations used by practitioners have been restricted to standard schemes which rely on the analysis of the the first forward and backward passes. In the present work, pushing the analysis beyond the first gradient step (which is made possible by the Tensor Program framework), we have studied how to train neural networks with parameterizations that enjoy radically different behaviors, such as forgetting the contribution of the initial weights after the first weight update.

The parameterizations we have analyzed, which we refer to as *integrable parameterizations*, have been previously described with tools from the *mean-field* literature, and we have deepened our understanding of these models with a different perspective. Indeed, we have shown that these parameterizations are trivial for deep networks with centered i.i.d. initialization and a constant learning rate: they are stuck at initialization. This observation led us to explore various ways to escape this initial stationary point and initiate learning. Among those methods, we found that the only one that does not lead to a degenerate behaviour is to use large learning rates for the first gradient step. We proved that in the infinite-width limit the resulting dynamic is equivalent to a modification of μP where the initial weights are removed after the first gradient step. Importantly, the random fluctuations around the limit—which are ignored in the mean-field description—turn out to actually be essential for our analysis, since it is by amplifying them that we are able to escape the stationary point.

Extending our theoretical results to a more general class of activation functions requires more thorough technical work and is left as an open problem. Also, analyzing rigorously the impact of the presence or absence of the initial weight matrices on the learning behavior appears to be an interesting avenue for future research. Finally, understanding the generalization properties of IP-LLR and μP remains an important open question but is beyond the scope of this chapter.

3 - Symmetries in the dynamics of infinitely wide two-layer neural networks

3.1 . Introduction

The ability of neural networks to learn rich representations—or features—of their input data is commonly observed in state-of-the art models (Zeiler and Fergus, 2014; Cammarata et al., 2020) and often thought to be the reason behind their good practical performance (Goodfellow et al., 2016, Chap. 1). Yet, our theoretical understanding of how feature learning arises from simple gradient-based training algorithms remains limited. Much progress (discussed in Section 3.1.3) has been made recently to understand the power and limitations of gradient-based learning with neural networks, showing in particular their superiority over fixed-feature methods on some difficult tasks. However, positive results are often obtained for algorithms that differ in substantial ways from plain (stochastic) gradient descent (e.g. the layers trained separately, or the algorithm makes just one truly non-linear step, etc).

In this work, we take the algorithm as a given and instead adopt a descriptive approach. Our goal is to improve our understanding of how neural networks behave in the presence of symmetries in the data with plain gradient descent (GD) on two-layer fully-connected ReLU neural networks. To this end, we investigate situations with strong symmetries on the data, the target function and on the initial parameters, and study the properties of the training dynamics and the learned predictor in this context.

3.1.1 . Problem setting

We denote by d the input dimension, ρ the input data distribution which we assume to be uniform over the unit sphere $\mathbb{S}^{d-1}$ of $\mathbb{R}^d$, and by $\mathcal{P}_2(\Omega)$ the space of probability measures with finite second moments over a measurable space Ω. We call σ the activation function, which we take to be ReLU, that is $\sigma(z) = \max(0, z)$, $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ the loss function, which we assume to be continuous in both arguments and continuously differentiable *w.r.t.* its second argument and we denote by $\partial_2 \ell$ this derivative.

Mean-field limit of two-layer networks. In this work, we consider the infinite-width limit in the mean-field regime of the training dynamics of two-layer networks without intercept with a ReLU activation function. Given a measure $\mu \in \mathcal{P}_2(\mathbb{R} \times \mathbb{R}^d)$, we consider the infinitely wide two-layer network parameterized

by μ, defined, for any input $x \in \mathbb{R}^d$, by

$$f(\mu; x) = \int_{c \in \mathbb{R}^{1+d}} \phi(c; x) \mathrm{d}\mu(c), \tag{3.1}$$

where, for any $c = (a, b) \in \mathbb{R} \times \mathbb{R}^d$, $\phi(c; x) = a\sigma\left(b^\top x\right)$. Note that width-$m$ two-layer networks with input weights $(b_j)_{j \in [1,m]} \in (\mathbb{R}^d)^m$ and output weights $(a_j)_{j \in [1,m]} \in \mathbb{R}^m$ can be recovered by a measure $\mu_m = (1/m) \sum_{j=1}^m \delta_{(ma_j, b_j)}$ with m atoms.

Objective and Wasserstein gradient flow. We consider the problem of minimizing the *population loss* objective for a given target function $f^* : \mathbb{R}^d \to \mathbb{R}$, which we assume to be bounded on the unit sphere, that is

$$\min_{\mu \in \mathcal{P}_2(\mathbb{R} \times \mathbb{R}^d)} \Big(F(\mu) := \mathbb{E}_{x \sim \rho} \left[\ell \left(f^*(x), f(\mu; x) \right) \right] \Big). \tag{3.2}$$

The Fréchet derivative of the objective function F at μ is given by the function $F'_\mu(c) = \mathbb{E}_{x \sim \rho} \left[\partial_2 \ell \left(f^*(x), f(\mu; x) \right) \phi(c; x) \right]$ for any $c = (a, b) \in \mathbb{R} \times \mathbb{R}^d$ (for more details, see Appendix B.2.1). Starting from a given measure $\mu_0 \in \mathcal{P}_2(\mathbb{R} \times \mathbb{R}^d)$, we study the Wasserstein gradient flow (GF) of the objective (3.2) which is a path $(\mu_t)_{t \geq 0}$ in the space of probability measures satisfying, in the sense of distributions, the partial differential equation (PDE) known as the continuity equation:

$$\begin{aligned} \partial_t \mu_t &= -\mathrm{div}\left(v_t \, \mu_t \right), \\ v_t(c) &:= -\nabla F'_{\mu_t}(c). \end{aligned} \tag{3.3}$$

Initialization. We make the following assumption on the initial measure $\mu_0 \in \mathcal{P}_2(\mathbb{R} \times \mathbb{R}^d)$: μ_0 decomposes as $\mu_0 = \mu_0^1 \otimes \mu_0^2$ where $\mu_0^1, \mu_0^2 \in \mathcal{P}_2(\mathbb{R}) \times \mathcal{P}_2(\mathbb{R}^d)$. This follows the standard initialization procedure at finite width. Because no direction should *a priori* be favored, we assume μ_0^2 to have spherical symmetry, *i.e.*, it is invariant under any orthogonal transformation, and we additionally assume that $|a| = ||b||$ almost surely at initialization. It is shown in (Chizat and Bach, 2020, Lemma 26), and (Wojtowytsch, 2020, Section 2.5), that with this assumption, μ_t stays supported on the set $\{|a| = ||b||\}$ for any $t \geq 0$.

Comment on the assumptions. The assumption that μ_0 decomposes as a product of two measures is to stay as close as possible to what is done in practice (independent initialization for different layers). The assumption that $|a| = ||b||$ is of a technical nature, and, along with the regularity conditions on the loss ℓ and the input data distribution ρ, ensures that the Wasserstein GF (3.3) is well-defined (Wojtowytsch, 2020, Lemma 3.1, Lemma 3.9) when using ReLU as an activation function (which bears technical difficulties because of its non-smoothness). The results of Section 3.2 hold for other activation functions which

potentially require less restrictive assumptions on μ_0 and ρ but still require μ_0 to decompose as a product of measures. In contrast, the results of Sections 3.3 and 3.4 are specific to $\sigma = \text{ReLU}$ and thus require the assumptions above on μ_0 and ρ. Since our work focuses mostly on ReLU, we choose to state the results of all sections with the (more restrictive) assumptions stated above on μ_0 and ρ.

Relationship with finite-width GD. If $\mu_0 = (1/m)\sum_{j=1}^{m} \delta_{(a_j(0),b_j(0))}$ is discrete, the Wasserstein GF (3.3) is exactly continuous-time GD on the parameters of a standard finite-width neural network, and discretization errors (*w.r.t.* the number of neurons) can be provided (Mei et al., 2018; Nguyen and Pham, 2020).

3.1.2 . Summary of contributions

Our main object of study is the gradient flow of the *population* risk of *infinitely wide* two-layer ReLU neural networks without intercept. Our motivation to consider this idealistic setting—infinite data and infinite width—is that it allows, under suitable choices for ρ and μ_0, the emergence of exact symmetries which are only approximate in the non-asymptotic setting[1].

Symmetries, structure, and convergence. In this work, we are interested in the structures learned by the predictor $f(\mu_t; \cdot)$ under GF as t grows large. Specifically, we make the following contributions:

- In Section 3.2, we prove that if f^* is invariant under some orthogonal linear map T, then $f(\mu_t; \cdot)$ inherits this invariance under GF (Proposition 3.2.1).
- In Section 3.3, we study the case when f^* is an *odd* function and show that the network converges to the best linear approximator of f^* at an exponential rate (Theorem 3.3.2). Linear predictors are optimal over the hypothesis class in that case, in particular because there is no intercept in our model.
- In Section 3.4, we consider the *multi-index model* where f^* depends *only* on the orthogonal projection of its input onto some sub-space H of dimension d_H. We prove that the dynamics can be reduced to a PDE in dimension d_H. If in addition, f^* is the Euclidean norm of the projection of the input, we show that the dynamics reduce to a one-dimensional PDE (Theorem 3.4.3). In the latter case, we were not able to prove theoretically the convergence of the neurons of the first layer towards H, and leave this as an open problem but we provide numerical evidence in favor of this result.

The code to reproduce the results of the numerical experiments can be found at: https://github.com/karl-hajjar/learning-structure.

[1] In contrast, our focus on GF is only for theoretical convenience and most of our results could be adapted to the case of GD.

3.1.3 . Related work

Infinite-width dynamics. It has been shown rigourously that for infinitely wide networks there is a clear distinction between a feature-learning regime and a kernel regime (Chizat et al., 2019; Yang and Hu, 2021). For shallow networks, this difference stems from a different scale (*w.r.t.* width) of the initialization where a large initialization leads to the Neural Tangent Kernel (NTK) (*a.k.a.* the "lazy regime") which is equivalent to a kernel method with random features (Jacot et al., 2018) whereas a small initialization leads to the so-called *mean-field* (MF) limit where features are learned from the first layer (Chizat et al., 2019; Yang and Hu, 2021). However, it is unclear in this setting exactly what those features are and what underlying structures are learned by the network. The aim of the present work is to study this phenomenon from a theoretical perspective for infinitely wide networks and to understand the relationship between the ability of networks to learn specific structures and the symmetries of a given task.

A flurry of works study the dynamics of infinitely wide two-layer neural networks. Chizat and Bach (2018); Mei et al. (2018); Rotskoff and Vanden-Eijnden (2018); Wojtowytsch (2020); Sirignano and Spiliopoulos (2020) study the gradient flow dynamics of the MF limit and show that they are well-defined in general settings and lead to convergence results (local or global depending on the assumptions). On the other hand, Jacot et al. (2018) study the dynamic of the NTK parameterization in the infinite-width limit and show that it amounts to learning a *linear* predictor on top of random features (fixed kernel), so that there is no feature learning.

Convergence rates. In the MF limit, convergence rates are in general difficult to obtain in a standard setting. For instance, Chizat and Bach (2018); Wojtowytsch (2020) show the convergence of the GF to a global optimum in a general setting but this does not allow convergence rates to be provided. To illustrate the convergence of the parameterizing measure to a global optimum in the MF limit, E et al. (2020) prove *local* convergence (see Section 7) for one-dimensional inputs and a specific choice of target function in $O(t^{-1})$ where t is the time step. At finite-width, Daneshmand and Bach (2022) also prove convergence of the parameters to a global optimum in $O(t^{-1})$ using an algebraic idea which is specific to the ad-hoc structure they consider (inputs in two dimensions and target functions with finite number of atoms).

In Section 3.3, we show convergence of the MF limit at an exponential rate when the target function is odd. In the setting of this section, the training dynamics are degenerate and although input neurons move, the symmetries of the problem imply that the predictor is linear.

Low-dimensional structure. Studying how neural networks can adapt to hidden low-dimensional structures is a way of approaching theoretically the feature-

learning abilities of neural networks. Bach (2017) studies the statistical properties of infinitely wide two-layer networks, and shows that when the target function only depends on the projection on a low-dimensional sub-space, these networks circumvent the curse of dimensionality with generalization bounds which only depend on the dimension of the sub-space. In a slightly different context, Chizat and Bach (2020) show that for a binary classification task, when there is a low-dimensional sub-space for which the projection of the data has sufficiently large inter-class distance, only the dimension of the sub-space (and not that of the ambient space) appears in the upper bound on the probability of misclassification. Whether or not such a low-dimensional sub-space is actually learned by GD is not addressed in these works.

Similarly, Cloninger and Klock (2021); Damian et al. (2022) focus on learning functions which have a hidden low-dimensional structure with neural networks. They consider a single step of GD on the input layer weights and show that the approximation / generalization error adapts to the structure of the problem: they provide bounds on the number of data points / parameters needed to achieve negligible error, which depend on the reduced dimension and not the dimension of the ambient space. In a similar context, Mousavi-Hosseini et al. (2022) consider (S)GD on the first layer only of a finite-width two-layer network and show that with sufficient L_2-regularization and with a standard normal distribution on the input data the first layer weights align with the lower-dimensional sub-space when trained for long enough. They then use this property to provide statistical results on networks trained with SGD.

In a setting close to ours but on a classification task with finite-data and at finite-width, Paccolat et al. (2021) compare the feature learning regime with the NTK regime in the presence of hidden low-dimensional structure and quantify for each regime the scaling law of the test error *w.r.t.* the number of training samples, mostly focusing on the case $d_H = 1$.

In a similar setting to that of (Bach, 2017), Abbe et al. (2022) study how GF for infinitely wide two-layer networks can learn specific classes of functions which have a hidden low-dimensional structure when the inputs are Rademacher variables. This strong symmetry assumption ensures that the learned predictor shares the same low-dimensional structure at any time step (from the $t = 0$) and this allows them to characterize precisely what classes of target functions can or cannot be learned by GF in this setting. In contrast, we are interested in how infinitely wide networks *learn* those low-dimensional structures during training, and in the role of symmetries in enabling such a behaviour after initialization.

Learning representations. An existing line of work (Yehudai and Shamir, 2019; Allen-Zhu et al., 2019; Abbe et al., 2021; Damian et al., 2022; Ba et al., 2022) studies in depth the representations learned by neural networks trained with (S)GD at finite-width from a different perspective focusing on the advantages of

feature-learning in terms of performance comparatively to using random features. In contrast, our aim is to describe the representations themselves in relationship with the symmetries of the problem.

Symmetries. We stress that the line of work around symmetries of neural networks dealing with finding network architectures for which the output is invariant (*w.r.t.* to its input or parameters) by some *group* of transformations (see Bloem-Reddy and Teh, 2020; Ganev and Walters, 2021; Głuch and Urbanke, 2021, and references therein) is entirely different from what we are concerned with in the present work. In contrast, the setting of (Mei et al., 2018) is much closer to ours as they study how the invariances of the target function / input data can lead to simplifications in the dynamics of infinitely wide two-layer networks in the mean-field regime which allows them to prove global convergence results.

3.1.4 . Notations

We denote by $\mathcal{M}_+(\Omega)$ the space of non-negative measures over a measurable space Ω. For any measure μ and measurable map T, $T_\#\mu$ denotes the pushforward measure of μ by T. We denote by $\mathcal{O}(p)$ and $\mathrm{id}_{\mathbb{R}^p}$ respectively the orthogonal group and the identity map of $\mathbb{R}^p$ for any $p \in \mathbb{N}$. Finally, $\langle \cdot, \cdot \rangle$ is the Euclidean inner product and $||\cdot||$ the corresponding norm.

3.2 . Invariance under orthogonal symmetries

In this section, we demonstrate that if the target function f^* is invariant under some orthogonal transformation T, since the input data distribution is also invariant under T, then $f(\mu_t; \cdot)$ is invariant under T as well for any $t \geq 0$. This invariance property of the dynamics *w.r.t.* orthogonal symmetries is possible with an infinite number of neurons but is only approximate at finite-width. It is noteworthy that the results of this section hold for any activation function σ and input data distribution ρ which has the same symmetries as f^*, provided that the Wasserstein GF (3.3) is unique. We start with a couple of definitions:

Definition 3.2.1 (Function invariance)**.** Let T be a map from $\mathbb{R}^d$ to $\mathbb{R}^d$, and $f : \mathbb{R}^d \to \mathbb{R}$. Then, f is said to be *invariant* (*resp. anti-invariant*) under T if for any $x \in \mathbb{R}^d$, $f(T(x)) = f(x)$ (*resp.* $f(T(x)) = -f(x)$).

Definition 3.2.2 (Measure invariance)**.** Let $\Omega \subset \mathbb{R}^d$, T be a measurable map from Ω to Ω, and μ be a measure on Ω. Then, μ is said to be invariant under T if $T_\#\mu = \mu$, or equivalently, if for any continuous and compactly supported $\varphi : \Omega \to \mathbb{R}$, $\int \varphi(x)\mathrm{d}\mu(x) = \int \varphi(T(x))\mathrm{d}\mu(x)$.

We are now ready to state the two main results of this section.

Proposition 3.2.1 (Learning invariance)**.** Let $T \in \mathcal{O}(d)$, and assume that f^* is invariant under T. Then, for any $t \geq 0$, the Wasserstein GF μ_t of Equation (3.3) is invariant under $\tilde{T} : (a, b) \in \mathbb{R} \times \mathbb{R}^d \mapsto (a, T(b))$, and the corresponding predictor $f(\mu_t; \cdot)$ is invariant under T.

Proposition 3.2.2 (Learning anti-invariance)**.** Under the same assumptions as in Proposition 3.2.1 except now we assume f^* is anti-invariant under T, and assuming further that $\partial_2 \ell(-y, -\hat{y}) = -\partial_2 \ell(y, \hat{y})$ for any $y, \hat{y} \in \mathbb{R}$, and that μ_0^1 is symmetric around 0 (*i.e.*, invariant under : $a \in \mathbb{R} \mapsto -a$), we then have that for any $t \geq 0$, the Wasserstein GF μ_t in Equation (3.3) is invariant under $\tilde{T} : (a, b) \in \mathbb{R} \times \mathbb{R}^d \mapsto (-a, T(b))$, and the corresponding predictor $f(\mu_t; \cdot)$ is anti-invariant under T.

Remark. The results above also hold for networks with intercepts at both layers. The conditions of Proposition 3.2.2 are satisfied by both the squared loss and the logistic loss (*a.k.a.* the cross-entropy loss).

Essentially, those results show that training with GF preserves the orthogonal symmetries of the problem: the invariance of the target function under an orthogonal transformation leads to the same invariance for μ_t and $f(\mu_t; \cdot)$. The proof, presented in Appendix B.3, relies crucially on the fact that T is an orthogonal map which combines well with the structure of $\phi(c; x)$ involving an inner product. The idea is essentially that the orthogonality of T allows us to relate the gradient of ϕ (and consequently of F'_{μ_t}) *w.r.t.* c at $(T(c); x)$ to the same gradient at $(c; T^{-1}(x))$ and then to use the invariance of f^* and ρ to conclude.

In the following sections we discuss the particular cases where functions are (anti-)invariant under $-\mathrm{id}_{\mathbb{R}^d}$ (*i.e.*, even or odd functions) or some sub-group of $\mathcal{O}(d)$.

3.3 . Exponential convergence for odd target functions

We consider here an odd target, function, *i.e.*, for any $x \in \mathbb{R}^d$, $f^*(-x) = -f^*(x)$.

Linearity of odd predictors. Proposition 3.2.2 ensures that the predictor $f(\mu_t; \cdot)$ associated with the Wasserstein GF of Equation (3.3) is also odd at any time $t \geq 0$, and we can thus write, for any x, $f(\mu_t; x) = \frac{1}{2}\left(f(\mu_t; x) - f(\mu_t; -x)\right)$, which yields

$$f(\mu_t; x) = \frac{1}{2}\left(\int_{a,b} a\left[\sigma(b^\top x) - \sigma(-b^\top x)\right] \mathrm{d}\mu_t(a, b)\right) = \frac{1}{2}\int_{a,b} a\left(b^\top x\right) \mathrm{d}\mu_t(a, b),$$

where the last equality stems from the fact that for ReLU, $\sigma(x) - \sigma(-x) = x$. Put differently, the predictor is **linear**: it is the same as replacing σ by $\frac{1}{2}\mathrm{id}_{\mathbb{R}^d}$, and

$f(\mu_t; x) = w(t)^\top x$, where

$$w(t) := \frac{1}{2}\int_{a,b} a\, b\, \mathrm{d}\mu_t(a,b) \in \mathbb{R}^d. \tag{3.4}$$

This degeneracy is not surprising as in fact, a linear predictor is the best one can hope for in this setting. Indeed, consider the following assumption and the next lemma:

Assumption 5 (Squared loss function)**.** The loss function ℓ is the squared loss, *i.e.*, $\ell(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$, and thus satisfies the condition of Proposition 3.2.2.

We make this assumption in order to provide an explicit convergence rate in Theorem 3.3.2 below.

Lemma 3.3.1 (Optimality of odd predictors)**.** *Let f be a predictor in the hypothesis class $\mathcal{F} := \left\{ : x \mapsto \int a\sigma(b^\top x)\mathrm{d}\mu(a,b); \mu \in \mathcal{P}_2(\mathbb{R} \times \mathbb{R}^d)\right\}$. Then, denoting $f_{odd}(x) := \frac{1}{2}(f(x) - f(-x))$ (resp. $f_{even} := \frac{1}{2}(f(x) + f(-x))$) the odd (resp. even) part of f, one has:*

(i) $f_{odd} \in \mathcal{F}$,

(ii) $L(f) := \mathbb{E}_{x\sim\rho}\left[(f^*(x) - f(x))^2\right] \geq \mathbb{E}_{x\sim\rho}\left[(f^*(x) - f_{odd}(x))^2\right] =: L(f_{odd})$,

(iii) *equality holds if and only if f is odd ρ-almost surely*.

Proof. The result readily follows from the decomposition $f = f_{\text{odd}} + f_{\text{even}}$ which leads to

$$L(f) = L(f_{\text{odd}}) + \underbrace{\mathbb{E}_{x\sim\rho}\left[(f_{\text{even}}(x))^2\right]}_{\geq 0} - 2\underbrace{\mathbb{E}_{x\sim\rho}\left[(f^*(x) - f_{\text{odd}}(x))\, f_{\text{even}}(x)\right]}_{0 \text{ by symmetry}}.$$

We then get that $L(f) \geq L(f_{\text{odd}})$ with equality if and only if $\mathbb{E}_{x\sim\rho}\left[(f_{\text{even}}(x))^2\right] = 0$, *i.e.*, $f_{\text{even}}(x) = 0$ for ρ-almost every x. Finally, if $\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})$, then $\nu := \frac{1}{2}(\mu + S_\#\mu) \in \mathcal{P}_2(\mathbb{R}^{d+1})$, where $S : (a,b) \in \mathbb{R}^{d+1} \mapsto (-a,-b)$, and $f(\nu;\cdot) = f_{\text{odd}}(\mu;\cdot)$, which shows $f_{\text{odd}}(\mu;\cdot) \in \mathcal{F}$. □

Since, as shown above, any odd predictor turns out to be linear because of the symmetries of ReLU, in this context, the best one can expect is thus to learn the best linear predictor.

Exponential convergence for linear networks. We are thus reduced to studying the dynamics of linear networks (which in our case are infinitely wide), which is an interesting object of study in its own right (Ji and Telgarsky 2018 show a result similar to our result below in the finite-width case with the logistic loss on a binary classification task). In this case, the Wasserstein GF (3.3) (with ReLU

replaced by $\frac{1}{2}\mathrm{id}_{\mathbb{R}^d}$) is defined for more general input distributions $\mathbb{P} \in \mathcal{P}_2(\mathbb{R}^d)$ (*e.g.*, empirical measures) and target functions f^*. The objective in this context is thus to learn:

$$w^\star \in \operatorname*{argmin}_{w \in \mathbb{R}^d} \left(Q(w) := \frac{1}{2}\mathbb{E}_{x\sim\mathbb{P}}\left[(f^*(x) - \langle w,\, x\rangle)^2 \right] \right) \tag{3.5}$$

with the dynamics of linear infinitely wide two-layer networks described by the Wasserstein GF (3.3) where the activation function σ is replaced by $\frac{1}{2}\mathrm{id}_{\mathbb{R}^d}$. Theorem 3.3.2 below shows exponential convergence to a global minimum of Q as soon as the problem is strongly convex. Note that although in this case both $\phi(\cdot;\cdot)$ (see Equation (3.1)) and the predictor in the objective Q are linear *w.r.t.* the input, only the predictor in Q is linear in the parameters (ordinary least squares).

Theorem 3.3.2. *Assume that the smallest eigenvalue λ_{min} of $\mathbb{E}_{x\sim\mathbb{P}}[xx^\top]$ is positive. Let $(\mu_t)_{t\geq 0}$ be the Wasserstein GF associated to* (3.3) *with activation function $\frac{1}{2}\,\mathrm{id}_{\mathbb{R}^d}$ instead of $\sigma =$ ReLU, and call $w(t) = \frac{1}{2}\int ab\,\mathrm{d}\mu_t(a,b) \in \mathbb{R}^d$. Then, there exits $\eta > 0$ and $t_0 > 0$ such that, for any $t \geq t_0$,*

$$\Big(Q(w(t)) - Q(w^\star)\Big) \leq e^{-2\eta\lambda_{min}(t-t_0)}\Big(Q(w(t_0)) - Q(w^\star)\Big).$$

Remark. Note that as soon as $\mathbb{P}$ has spherical symmetry, the problem becomes strongly convex by Lemma B.1.3. Note that although $F(\mu_t) = Q(w(t))$, $(w(t))_{t\geq 0}$ is **not** a gradient flow for the (strongly) convex objective Q (which would immediately guarantee exponential convergence to the global minimum).

The proof, provided in Appendix B.4, proceeds in two steps: first it is shown that $w'(t) = -H(t)\nabla Q(w(t))$ for some positive definite matrix $H(t)$ whose smallest eigenvalue is always lower-bounded by a positive quantity, then we prove that this leads to exponential convergence. Figure 3.1 illustrates that the dynamics of GF on F remain non-linear in that they do not reduce to GF on Q (although the paths are close). To simulate GF on F we use a large (but finite) number of neurons $m = 1,024$ and a small (but positive) step-size 10^{-2} and simply proceed to do GD on the corresponding finite-dimensional objective (see comment in Section 3.1.1 on relationship between the Wasserstein GF and finite-width GD).

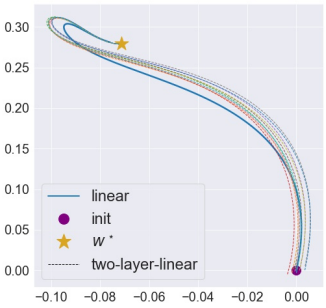


Figure 3.1: GD path for two coordinates: two-layer linear network *vs* pure linear model.

3.4 . Learning the low-dimensional structure of the problem

Consider a linear sub-space H of dimension $d_H < d$ (potentially much smaller than the ambient dimension), and assume f^* has the following structure: $f^*(x) = f_H(p_H(x))$ where p_H is the orthogonal projection onto H (which we also write x^H for simplicity, and we reserve sub-scripts for denoting entries of vectors) and $f_H : H \to \mathbb{R}$ is a given function.

In this context it is natural to study whether the learned function shares the same structure as f^*. As observed in Figure 3.2 this is not the case in finite time, but it is reasonable however to think that the learned predictor $f(\mu_t; \cdot)$ shares the same structure as f^* as $t \to \infty$, and we give numerical evidence in this direction. On the other hand, we prove rigorously that the structure of the problem allows to reduce the dynamics to a lower-dimensional PDE. In this section, we consider for simplicity that μ_0^1 is the uniform distribution over $\{-1, +1\}$ and that μ_0^2 is the uniform distribution over $\mathbb{S}^{d-1}$.

Comment on the assumptions for this section. The assumptions that $|a| = ||b||$ on the support of μ_0 is crucial here. This ensures that the Wasserstein GF (3.3) is well-defined and that μ_t stays supported on the set $\{|a| = ||b||\}$ for any $t \geq 0$, a fact which is used in the proofs. The assumption that ρ is the uniform distribution over the unit sphere bears some importance but could likely be replaced by other measures with spherical symmetry provided that the dynamics would still be well-defined and at the cost of more technical proofs.

3.4.1 . Symmetries and invariance

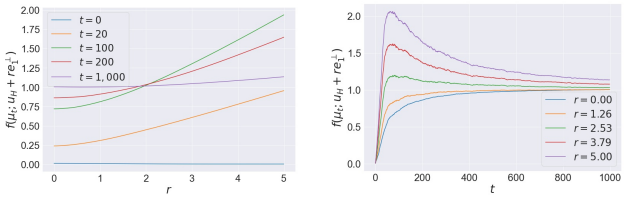
The structure of f^* implies that it is invariant by any $T \in \mathcal{O}(d)$ which preserves H, *i.e.*, such that its restrictions to H and $H^\perp$ are $T_{|H} = \mathrm{id}_H$ and $T_{|H^\perp} \in \mathcal{O}(d_\perp)$, where $\mathcal{O}(d_\perp)$ is the orthogonal group of $H^\perp$ whose dimension is $d_\perp = d - d_H$. By Proposition 3.2.1, such transformations also leave the predictor $f(\mu_t; \cdot)$ invariant for any $t \geq 0$ since ρ is spherically symmetric. Lemma 3.4.1 below then ensures that $f(\mu_t; x)$ depends on the projection $x^\perp$ onto $H^\perp$ only through its norm, that is $f(\mu_t; x) = \tilde{f}_t(x^H, ||x^\perp||)$ for some $\tilde{f}_t : H \times \mathbb{R}_+ \to \mathbb{R}$.

Lemma 3.4.1 (Invariance by a sub-group of $\mathcal{O}(d)$)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be invariant under any $T \in \mathcal{O}(d)$ such that $T_{|H} = \mathrm{id}_H$ and $T_{|H^\perp} \in \mathcal{O}(d_\perp)$. Then, there exists some $\tilde{f} : H \times \mathbb{R}_+ \to \mathbb{R}$ such that for any $x \in \mathbb{R}^d$, $f(x) = \tilde{f}(x^H, ||x^\perp||)$.*

Proof. Consider $\tilde{f} : (x^H, r) \in H \times \mathbb{R}_+ \mapsto f(x^H + re_1^\perp)$ where $e_1^\perp$ is the first vector of an orthonormal basis of $H^\perp$, and let $x \in \mathbb{R}^d$. If $x^\perp = 0$, the result is obvious. Otherwise, consider an orthogonal linear map T_x such that $T_{x|H} = \mathrm{id}_H$ and T_x sends $x^\perp / ||x^\perp||$ on $e_1^\perp$. The invariance of f under T_x implies $f(x) = f(T_x(x)) = f(x^H + ||x^\perp|| e_1^\perp) = \tilde{f}(x^H, ||x^\perp||)$. □

Figure 3.2 shows that the dependence in $||x^\perp||$ cannot be removed in finite time: $f(\mu_t; u_H + re_1^\perp)$ does depend on the distance $r \in \mathbb{R}_+$ to H, but this dependence tends to vanish as $t \to \infty$. The plots of Figure 3.2 are obtained

by discretizing the initial measure $\mu_{0,m} = \frac{1}{m}\sum_{j=1}^{m} \delta_{(a_j(0),b_j(0))}$ with $m = 1,024$ atoms, and sampling $a_j(0) \sim \mathcal{U}(\{-1,+1\})$ and $b_j(0) \sim \mathcal{U}(\mathbb{S}^{d-1})$. We perform GD with a finite step-size $\eta =$ and a finite number $n = 256$ of fresh i.i.d. samples from the data distribution per step with $f^*(x) = ||x^H||$, $d = 20$ and $d_H = 5$.



(a) $f(\mu_t; u_H + re_1^\perp)$ *vs* r (b) $f(\mu_t; u_H + re_1^\perp)$ *vs* t

Figure 3.2: $f(\mu_t; u_H + re_1^\perp)$ *vs* r and t for a random $u_H \in \mathbb{S}^{d_H-1}$ with $d = 20$, $d_H = 5$.

Dynamics over the sphere $\mathbb{S}^{d-1}$. Using the positive 1-homogeneity of ReLU, and with the assumptions on μ_0, the dynamics on $\mu_t \in \mathcal{P}_2(\mathbb{R}^{d+1})$ can be reduced to dynamics on the space $\mathcal{M}_+(\mathbb{S}^{d-1})$ of non-negative measures over $\mathbb{S}^{d-1}$: only the direction of neurons matter and their norm only affects the total mass. From this point of view, neurons with positive and negative output weights behave differently and have separate dynamics. Indeed, consider the pair of measures $(\nu_t^+, \nu_t^-) \in \mathcal{M}_+(\mathbb{S}^{d-1})^2$ characterized by the property that for any continuous test function $\varphi : \mathbb{S}^{d-1} \to \mathbb{R}$,

$$\int_u \varphi(u)\mathrm{d}\nu_t^\pm(u) = \int_{\pm a \geq 0, b} |a|\,||b||\varphi\left(\frac{b}{||b||}\right)\mathrm{d}\mu_t(a,b), \tag{3.6}$$

where we have used the superscript $^\pm$ to denote either ν_t^+ or ν_t^- and the right-hand side is changed accordingly (the integration domain) depending on the sign $+$ or $-$. Because ReLU is positively 1-homogeneous, we have $f(\mu_t; x) = \int \sigma(u^\top x)\mathrm{d}(\nu_t^+ - \nu_t^-)(x)$. It is shown in Appendix B.5.1 that $\nu_t^\pm$ satisfies, in the sense of distributions, the equation

$$\partial_t \nu_t^\pm = -\mathrm{div}\left(\pm \tilde{v}_t \nu_t^\pm\right) \pm 2 g_t \nu_t^\pm, \tag{3.7}$$

where, for any $u \in \mathbb{S}^{d-1}$,

$$\begin{aligned} g_t(u) &= -\int_y \partial_2 \ell\Big(f^*(y),\, f(\mu_t; y)\Big)\sigma(u^\top y)\mathrm{d}\rho(y), \\ \tilde{v}_t(u) &= -\int_y \partial_2 \ell\Big(f^*(y),\, f(\mu_t; y)\Big)\sigma'(u^\top y)\left[y - (u^\top y)u\right]\mathrm{d}\rho(y). \end{aligned} \tag{3.8}$$

Equation (3.7) can be interpreted as a Wasserstein-Fisher-Rao GF Gallouët et al. (2019) on the sphere since $\tilde{v}_t(u) = \mathrm{proj}_{\{u\}^\perp}(\nabla g_t(u))$.

Closed dynamics over $[0, \pi/2] \times \mathbb{S}^{d_H - 1}$. The dynamics on the pair (ν_t^+, ν_t^-) can be further reduced to dynamics over $[0, \pi/2] \times \mathbb{S}^{d_H - 1}$. Indeed, by positive 1-homogeneity of $f(\mu_t; \cdot)$ we may restrict ourselves to inputs $u \in \mathbb{S}^{d-1}$, and $f(\mu_t; u)$ depends only on u^H and $||u^\perp||$. However, because $||u^H||^2 + ||u^\perp||^2 = 1$, this dependence translates into a dependence on the direction $u^H/||u^H||$ of the projection onto H and the norm $||u^H||$. The former is an element of $\mathbb{S}^{d_H - 1}$ while the latter is given by the angle θ between u and H, that is $\theta := \arccos(u^\top u^H/||u^H||) = \arccos(||u^H||)$. This simplification leads to the following lemma:

Lemma 3.4.2. *Define the measures τ_t^+, τ_t^- by $\tau_t^\pm = P_\# \nu_t^\pm \in \mathcal{M}_+([0, \pi/2] \times \mathbb{S}^{d_H - 1})$ via $P : u \in \mathbb{S}^{d-1} \backslash H^\perp \mapsto (\arccos(||u_H||), u_H/||u_H||) \in [0, \pi/2] \times \mathbb{S}^{d_H - 1}$. Then, the measures τ_t^+, τ_t^- satisfy the equation*

$$\partial \tau_t^\pm = -\text{div}\left(\pm V_t \tau_t^\pm\right) \pm 2 G_t \tau_t^\pm, \tag{3.9}$$

*where $G_t : [0, \pi/2] \times \mathbb{S}^{d_H - 1} \to \mathbb{R}$, and $V_t : [0, \pi/2] \times \mathbb{S}^{d_H - 1} \to \mathbb{R}^{d_H + 1}$ are functions depending **only** on (τ_t^+, τ_t^-), and furthermore, $f(\mu_t; \cdot)$ can be expressed solely using τ_t^+, τ_t^- (exact formulas are provided in Appendix B.5.1).*

Abbe et al. (2022) show a similar result with a lower-dimensional dynamics in the context of infinitely wide two-layer networks when the input data have i.i.d coordinates distributed uniformly over $\{-1, +1\}$ (*i.e.*, Rademacher variables), except that they do not have the added dimension due to the angle θ, as we do, thanks to their choice of input data distribution.

Lemma 3.4.2 above illustrates how the GF dynamics of infinitely wide two-layer networks adapts to the lower-dimensional structure of the problem: the learned predictor and the dynamics can be described only in terms of the angle θ between the input neurons and H and their projection on the unit sphere of H. In essence, this means that the knowledge of the dynamics of the angle of the particles with H is enough to provide a complete picture of the system. This simplification matches the structure of the problem at hand and shows that the training dynamics of infinitely wide two-layer networks adapt to geometry of the problem in this case.

3.4.2 . One dimensional reduction

Since the predictors we consider are positively homogeneous, one cannot hope to do better than learn a positively homogeneous function. A natural choice of such a target function to learn is the Euclidean norm. With the additional structure that the target only depends on the projection onto H, this leads to considering $f^*(x) = ||x^H||$ which has additional symmetries compared to the general case presented above: it is invariant by any linear map T such that $T_{|H} \in \mathcal{O}(d_H)$ and $T_{|H^\perp} \in \mathcal{O}(d_\perp)$. By Proposition 3.2.1 those symmetries are shared by μ_t and $f(\mu_t; \cdot)$, and we show that in this case the dynamic reduces to a one-dimensional dynamic over the angle θ between input neurons and H.

We prove a general disintegration result for the uniform measure on the sphere in the Appendix (see Lemma B.1.4) which allows, along with some spherical harmonics analysis, to describe the reduced dynamics and characterize the objective that they optimize. This leads to the following result:

Theorem 3.4.3 (1d dynamics over the angle θ)**.** *Assume that $f^*(x) = ||x^H||$, and define the measures $(\tau_t^+, \tau_t^-) \in \mathcal{M}_+([0,\pi/2])^2$ from (ν_t^+, ν_t^-) via $P : u \in \mathbb{S}^{d-1} \mapsto \arccos(||u_H||) \in [0,\pi/2]$: $\tau_t^\pm = P_\# \nu_t^\pm$. Then, the pair (τ_t^+, τ_t^-) follows the Wasserstein-Fisher-Rao GF for the objective defined by $A(\tau^+, \tau^-) := \mathbb{E}\left[\ell\left(f(\tau^+, \tau^-; x), f^*(x)\right)\right]$ over the space $\mathcal{M}_+([0,\pi/2]) \times \mathcal{M}_+([0,\pi/2])$, where $f(\tau^+, \tau^-; x)$ is the expression (with a slight overloading of notations) of $f(\mu; x)$ in function of (τ^+, τ^-) (see Appendix B.5.2 for more details):*

$$\begin{aligned} \mathrm{d}\tau_0^\pm(\theta) &= \frac{1}{B\left(\frac{d_H}{2}, \frac{d_\perp}{2}\right)} \cos(\theta)^{d_H - 1} \sin(\theta)^{d_\perp - 1} \mathrm{d}\theta, \\ \partial_t \tau_t^\pm &= -\mathsf{div}\left(\pm V_t \tau_t^\pm\right) \pm 2 G_t \tau_t^\pm, \end{aligned} \tag{3.10}$$

where B is the Beta function, and

$$\begin{aligned} G_t(\theta) &= -\int_y \partial_2 \ell\Big(f^*(y),\, f(\mu_t; y)\Big) \sigma\left(\cos(\theta) y_1^H + \sin(\theta) y_1^\perp\right) \mathrm{d}\rho(y), \\ V_t(\theta) &= G_t'(\theta). \end{aligned}$$

*Additionally, $f(\mu_t; \cdot)$, G_t, and V_t **only depend** on the pair (τ_t^+, τ_t^-), and for any $t \geq 0$, it holds that $F(\mu_t) = A(\tau_t^+, \tau_t^-)$.*

Remark. The result should still hold for general ρ which are spherically symmetric as long as the Wasserstein GF (3.3) is well-defined but the proof is more technical. In addition, this result shows that even with more structure than in Lemma 3.4.2, the dynamics of infinitely wide two-layer networks are still able to adapt to this setting: these dynamics, as well as the learned predictor, can be *fully characterized* solely by the one-dimensional dynamics over the angle θ between input neurons and H. This is noteworthy since this angle determines the alignment of the neurons with H, and thus measures how much the representations learned by the network have adapted to the structure of the problem. Furthermore, as discussed below, this reduction with exact formulas enables efficient numerical simulation in one dimension.

Daneshmand and Bach (2022) prove the global convergence of a reduced one-dimensional dynamics in a context similar to ours but their original problem is two-dimensional and with a choice of activation function that leads to specific algebraic properties.

Expression of $f(\mu_t;\cdot)$. Because of the symmetries of $f(\mu_t;\cdot)$, which result from that of f^*, $f(\mu_t;x)$ depends only on $||x^H||$ and $||x^\perp||$. What is more, since $f(\mu_t;\cdot)$ is positively 1-homogeneous (because ReLU is) it actually holds that $f(\mu_t;x) = ||x||\tilde{f}_t(\varphi_x)$ where $\varphi_x = \arccos(||x^H||/||x||)$ is the angle between x and H, and $\tilde{f}_t(\varphi) := \int_\theta \tilde{\phi}(\theta;\varphi)\mathrm{d}(\tau_t^+ - \tau_t^-)(\theta)$, $\tilde{\phi}$ depending only on σ and fixed probability measures (see Appendix B.5.2 for an exact formula).

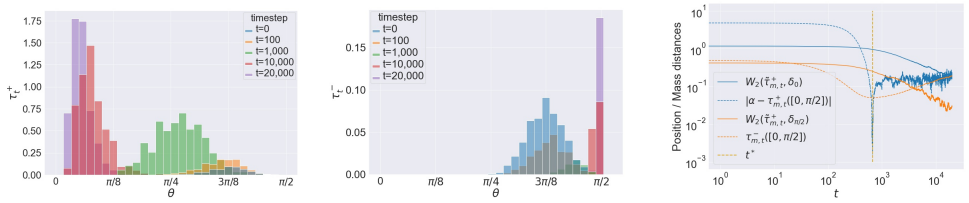
Learning the low-dimensional structure as $t \to \infty$. Although, as shown in Figure 3.2, $f(\mu_t;\cdot)$ does not learn the low-dimensional structure in finite-time, it is reasonable to expect that as $t \to \infty$, the measures $\tau_t^\pm$ put mass only on $\theta = 0$, indicating that the only part of the space that the predictor is concerned with for large t is the sub-space H. Since we assume here that the target function f^* is non-negative, the most natural limits for τ_t^+ and τ_t^- are $\tau_t^+ \to \alpha\delta_0$ with $\alpha > 0$, and $\tau_t^- \to 0$ (in the sense that $\tau_t^-([0,\pi/2]) \to 0$) as $t \to \infty$, because then the "negative" output weights do not participate in the prediction in the large t limit.

The global convergence result of Chizat and Bach (2018); Wojtowytsch (2020) still holds but is not quantitative and moreover does not guarantee that the limit is the one described above. We leave the proof of this result as an open problem, but we provide numerical evidence supporting this conjecture. Indeed, we take advantage of the one-dimensional reduction from Theorem 3.4.3, and numerically simulate the resulting dynamics by parameterizing $\tau_t^\pm$ via weight and position (Chizat, 2022) as $\mu_{m,t} = (1/m)\sum_{j=1}^m c_j^\pm(t)\delta_{\theta_j^\pm(t)}$, and simulating the corresponding dynamics for $c_j^\pm(t)$ and $\theta_j^\pm(t)$. The corresponding results are depicted in Figure 3.3 which are again obtained by discretizing the initial measures τ_0^+, τ_0^- and performing GD with finite step-size (see more details in Appendix B.6). Figures 3.3a and 3.3b show that the mass of τ_t^+ tends to concentrate around 0 while that of τ_t^- tends to concentrate around $\pi/2$, indicating that τ_t^+ adapts to the part of the space relevant to learning f^* while τ_t^- puts mass close to the orthogonal to that space.

Total mass of particles at convergence. If $\tau_\infty^- = 0$ and $\tau_\infty^+ = \alpha\delta_0$ as described above, we have $f(\mu_\infty;x) = \alpha||x||\tilde{\phi}(0;\varphi_x) = \alpha\frac{\Gamma(d_H/2)}{2\sqrt{\pi}\Gamma((d_H+1)/2)}||x||\cos(\varphi_x) = \frac{\alpha\Gamma(d_H/2)}{2\sqrt{\pi}\Gamma((d_H+1)/2)}||x^H||$. To recover exactly f^*, it must hold that $\alpha = \tau_\infty^+([0,\pi/2]) = \frac{2\sqrt{\pi}\Gamma((d_H+1)/2)}{\Gamma(d_H/2)}$. Defining the normalized measure $\tilde{\tau}_t^\pm = \tau_t^\pm/\tau_t^\pm([0,\pi/2])$, we expect $\tilde{\tau}_t^+$ to grow close to δ_0 and $\tilde{\tau}_t^-$ to $\delta_{\pi/2}$. In terms of total mass, we expect that $\tau_t^+([0,\pi/2])$ gets closer to α while $\tau_t^-([0,\pi/2])$ gets closer to 0.

The numerical behaviour depicted in Figure 3.3c seems to follow our intuitive description, at least until a critical time t^* in the numerical simulation which corresponds to the first time t where $\tau_t^+([0,\pi/2]) > \alpha$. While the total mass of $\tau_t^\pm$ (dashed lines) seems to approach its limit rapidly before t^* it slowly moves further

away from it for $t \geq t^*$. On the other hand, while the angles only slowly change before t^*, they start converging fast towards the corresponding Dirac measures after t^*. It is unclear whether this slight difference in behaviour (around the critical time t^*) between what we intuitively expected and the numerical simulation is an artefact of the finite width and finite step size or if it actually corresponds to some phenomenon present in the limiting model. For more details concerning the numerical experiments, see Appendix B.6. Note that there is *a priori* not a unique



(a) τ_t^+ distributions (b) τ_t^- distributions (c) Position / mass distances

Figure 3.3: Angle distributions τ_t^+/τ_t^- and position / mass distances with $m = 1024$, $d = 30$ and $d_H = 5$. (a) (*resp.* (b)) τ_t^+ (*resp.* τ_t^-) as a histogram for different t. (c) distances (in log-log scales) of the mass and positions of positive (blue) / negative (orange) particles to the intuitively expected limits: the distance in position is the Wasserstein-2 distance of the normalized (probability) measures $\tilde{\tau}_t^{\pm}$ to the corresponding Dirac measures while the distance in mass is the absolute error to the expected mass as $t \to \infty$.

global optimum: τ_∞^+ and τ_∞^- (if they exist) can compensate on parts of the space $[0, \pi/2]$ and lead to the same optimal predictor for different choices of measures. Our numerical experiments suggest that the GF dynamics select a "simple" solution where τ_∞^+ is concentrated on $\{\theta = 0\}$ and τ_∞^- vanishes (puts 0 mass everywhere), which is a form of implicit bias.

3.5 . Conclusion

We have explored the symmetries of infinitely wide two-layer ReLU networks and we have seen that: (i) they adapt to the orthogonal symmetries of the problem, (ii) they reduce to the dynamics of a linear network in the case of an odd target function and lead to exponential convergence, and (iii) when the target function depends only on the orthogonal projection onto a lower-dimensional sub-space H, the dynamics can be reduced to a lower-dimensional PDE. In particular, when f^* is the Euclidean norm, this PDE is over a one-dimensional space corresponding to the angle θ between the particles and H. We have presented numerical experiment indicating that the positive particles converge to the subspace H in this case and

leave the proof of this result as an open problem. We also leave as an open question whether the results of Section 3.2 extend to deeper networks.

4 - Coordinate descent over measures and dynamic optimization of two-layer networks

This chapter is devoted to studying the optimization properties of two-layer networks where the number of neurons is not fixed but can evolve dynamically during the course of training. The objective is two-fold: (i) derive methods which provide quantitative convergence bounds, and (ii) explore methods for training neural networks where the number of neurons is adjusted within the optimization procedure while still yielding a good value for the objective being optimized.

We stress that the work presented in this chapter is still under progress at the time of writing this thesis and some parts may thus appear incomplete.

4.1 . Introduction

Global convergence results exist for infinitely-wide two-layer networks (Nguyen and Pham, 2020; Chizat and Bach, 2018; Wojtowytsch, 2020) but no convergence rates are known in general. From a practical standpoint, while finite-width dynamics can be shown to closely track their infinite-width counterpart on bounded time intervals, the number of neurons needs to grow unbounded to guarantee convergence to a global minimum. We present here a method, similar to random coordinate descent in finite dimension, which adds a new neuron at each time step and is guaranteed to converge to a global minimum with a rate of $O(k^{\frac{-1}{d}})$ *w.r.t.* the iteration k. We show that the objective satisfies a condition akin to a Łojasiewicz-type inequality in the space of measures, and adapt the classical analysis from convex optimization to our setting to obtain a convergence rate.

While this method provides an explicit convergence rate, it is impractical computationally since the number of neurons needs to grow unbounded to approach optimality. We explore two alternatives in order to restrict the number of neurons within the optimization procedure by adding a penalization term to the objective. The first setting we study is a penalization by the total variation norm, which is akin to an L^1-penalty. Smoothness of the objective is lost in this setting and we extend the analysis of proximal optimization methods to the space of measures do deal with the non-smoothness. The second penalty we consider is a smooth kernel which is either attractive (drawing neighbouring neurons closer towards each other) or repulsive (pushing neighboring neurons away from each other).

Unfortunately, we could not obtain theoretical guarantees for these methods with a penalized objective: global convergence guarantees is lost *a priori* and there is no explicit bound on the number of neurons during the course of training. However, we show numerically that these sparsity-inducing penalties have good empirical performance and that the number of neurons grows sub-linearly with the

time step k.

4.2 . Setting

We consider a convex optimization problem over the set $\mathcal{M}(\mathbb{S}^{d-1})$ of signed measures on the sphere:

$$\min_{\mu \in \mathcal{M}(\mathbb{S}^{d-1})} F(\mu)$$

with $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$ convex. While Wasserstein gradient descent (or gradient flow) does not always enjoy global convergence guarantees, it often exhibits good practical behaviour and guarantees good local behaviour in certain settings (see, *e.g.*, Wojtowytsch, 2020; Chizat, 2022). Although the domain over which we optimize is infinite-dimensional, we take inspiration from convex methods in finite-dimension to propose an algorithm to minimize F. In particular, random coordinate descent allows to obtain convergence guarantees when F is smooth, and we also study its proximal variant when F has a non-smooth part. We describe algorithms for generic objectives F which can correspond to different contexts. This covers (but is not reduced to) the optimization of the (penalized) empirical risk for infinitely wide two-layer networks.

We consider an atomic measure μ_k at iteration k, initialized with a single neuron sampled on the sphere. At each iteration, a new neuron is sampled and its weight is set carefully. This is akin to random coordinate descent where a coordinate i is replaced by a neuron $u \in \mathbb{S}^{d-1}$. Since the dimension is infinite we never circle back to the same coordinate (here a neuron $u \in \mathbb{S}^{d-1}$). The resulting algorithm produces a sequence of measures $\mu_k = \sum_{j=1}^{k} c_j \delta_{u_j}$ where $c_j \in \mathbb{R}$ is the (signed) weight assigned to neuron $u_j \in \mathbb{S}^{d-1}$. In practical implementations, we alternate between coordinate descent steps and Wasserstein gradient steps as the latter often enhances the performance empirically while remaining a true descent step (the objective is guaranteed not to increase) which does not affect global convergence guarantees for descent methods.

While the coordinate descent algorithm provides a global convergence guarantee with an explicit rate, the number of neurons grows linearly with the number of iterations. Therefore we explore adding sparsity-inducing penalties to balance between optimizing the initial objective and limiting the computation cost incurred by the algorithm. We thus consider composite objectives of the form $F(\mu) = J(\mu) + \lambda H(\mu)$. We assume J to be smooth, *i.e.*, that it admits a *continuous* first variation (or Fréchet derivative, see Section 1.2.4) $V[\mu] : \mathbb{S}^{d-1} \to \mathbb{R}$ at every μ and V is L-Lipschitz for some $L > 0$, *i.e.*,

$$||V[\mu] - V[\nu]||_\infty \leq L|\mu - \nu|_{TV}.$$

We first study the case where $H = 0$ (no penalty) and provide an explicit convergence rate for the coordinate descent algorithm. Then, we explore two different

options for the penalty H: the first is the total variation norm which is akin to an L^1-penalty, and gives rise to proximal coordinate descent algorithms in the space of measures to deal with the non-smoothness of the total variation norm. The other option we consider consists of smooth kernels which either attract or repulse neighbouring particles. The intuition in the latter case is that sparsity will not be enforced explicitly as with the total variation penalty, but rather *induced implicitly* by the resulting dynamics which will tend to aggregate particles and thus effectively merge them if they are close enough.

4.2.1 . Organisation of the chapter

We first review in Section 4.3 techniques involved in gradient descent, coordinate descent and proximal methods as a lot of the ideas are relevant to our setting. Then, in Section 4.4 we study the case of a smooth F (no penalty) as well as the addition of a (non-smooth) total variation penalty, and we try to adapt some of the proof techniques to the infinite-dimensional setting of optimization over the space of measures. Finally, in Section 4.5 we study smooth kernel penalties and explore empirically the sparsity induced by such methods.

4.3 . A review of gradient descent, coordinate descent and proximal methods

In this section we review classical techniques and results in convex optimization. We start by the convergence of gradient descent for functions satisfying the Polyak-Łojasiewicz or general Łojasiewicz inequalities (which includes strongly convex functions), and then present a convergence proof for smooth convex functions. There is a variety of proof techniques when it comes to smooth convex optimization (see, *e.g.*, Richtárik and Takáč, 2014; Wright, 2015; Karimi et al., 2016 and the references therein), and our focus on Łojasiewicz-type inequalities is because they simplify some of the arguments and avoid relying too much on the Euclidean structure of the space, which will be relevant when we try to adapt these methods to the space of measures. Next we review the coordinate descent variants of gradient descent for both convex and strongly convex functions, and finally present proximal algorithms for taking into account convex but non-smooth penalties such as the L^1-penalty. We take the time to present the techniques and proofs in the finite-dimensional setting as they will come in handy when we deal with the infinite-dimensional setting of optimizing over the space of measures where similar arguments can be used (sometimes requiring additional assumptions).

In all this section we consider a generic convex function $f : \mathbb{R}^m \to \mathbb{R}$ which we assume to be smooth, that is differentiable with an L-Lipschitz gradient, *i.e.*, for any $x, y \in \mathbb{R}^m$ it holds

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||,$$

where $||-||$ denotes the L^2 norm. We assume that the minimum M^* of f is

realized at least for one $x^* \in \mathbb{R}^m$, and we consider the gradient descent algorithm (or its coordinate descent variant) where x_0 is a fixed initial point and for all $k \in \mathbb{N}$, $x_{k+1} = x_k - \eta \nabla f(x_k)$ where $\eta > 0$ is a step-size parameter.

Descent property implied by smoothness. The smoothness property already allows to derive a descent property for the gradient descent iterates. First, let us a give a useful property following from the smoothness assumption. For any $x, y \in \mathbb{R}^m$ it holds that

$$f(x+u) \leq f(x) + \nabla f(x)^\top u + \frac{L}{2}||u||^2. \tag{4.1}$$

Proof. This comes from noticing that $f(x+u) - f(x) = \int_0^1 \frac{d}{dt} f(x+tu) \mathrm{d}t$ and bounding the integral using the smoothness property:

$$\begin{aligned}
f(x+u) - f(x) &= \int_0^1 \frac{d}{dt} f(x+tu) \mathrm{d}t \\
&= \int_0^1 \nabla f(x+tu)^\top u \mathrm{d}t \\
&= \nabla f(x)^\top u + \int_0^1 \langle \nabla f(x+tu) - \nabla f(x), u \rangle \mathrm{d}t \\
&\leq \nabla f(x)^\top u + \int_0^1 L||u||^2 t \mathrm{d}t \\
&\leq \nabla f(x)^\top u + \frac{L}{2}||u||^2.
\end{aligned}$$

□

This inequality readily provides a **descent property**: for any step-size $\eta > 0$, it holds for any $x \in \mathbb{R}^m$

$$f(x - \eta \nabla f(x)) - f(x) \leq -\eta \left(1 - \frac{\eta L}{2}\right) ||\nabla f(x)||^2, \tag{4.2}$$

so that $f(x - \eta \nabla f(x)) \leq f(x)$ for any $\eta \in (0, \frac{2}{L})$, and what is more there is a strict descent as soon as x is not optimal since $\nabla f(x) \neq 0$ in this case. Thus, for smooth functions, the gradient descent iterates with appropriate step-size are always decreasing: $f(x_{k+1}) \leq f(x_k)$. The upper bound on the decrease in objective value on the right-hand-side of Inequality (4.2) is minimized for $\eta = 1/L$ and yields $f(x - \eta \nabla f(x)) - f(x) \leq -\frac{1}{2L}||\nabla f(x)||^2$, so that we often consider the step-size $\eta = 1/L$ in what follows. We note that although in practice the value of the constant L is *not known*, since the strict decrease property holds for a small enough step-size $\eta > 0$, all the convergence proofs we present still hold with small enough η, but we choose to use $\eta = 1/L$ for simplicity.

4.3.1 . Polyak-Łojasiewicz and generalized Łojasiewicz conditions

From the descent property (4.2) it follows that if the norm of the gradient at x can be lower-bounded by the sub-optimality gap $f(x) - M^*$, then a convergence rate is directly accessible. The Polyak-Łojasiewicz (PL) condition precisely states that there exists a constant $\tau > 0$ such that for all $x \in \mathbb{R}^m$

$$\frac{1}{2}||\nabla f(x)||^2 \geq \tau(f(x) - M^*). \tag{4.3}$$

Note that this inequality implies that all critical points are global minimizers since $\nabla f(x_c) = 0$ for any critical point x_c. The proof of convergence along with the appropriate rate easily follows even if the function f is not convex: from the descent property (4.2) with a step-size $1/L$, it holds

$$\begin{aligned} f(x_{k+1}) - M^* &\leq f(x_k) - M^* - \frac{1}{2L}||\nabla f(x_k)||^2 \\ &\leq \left(1 - \frac{\tau}{L}\right)(f(x_k) - M^*) \end{aligned}$$

and thus $0 \leq f(x_k) - M^* \leq \left(1 - \frac{\tau}{L}\right)^k (f(x_0) - M^*)$. Note that τ cannot be larger than L (which ensures $1 - \tau/L \in [0, 1)$), otherwise this would imply $f(x_{k+1}) < M^*$ which is impossible.

Polyak-Łojasiewicz condition for strongly convex functions. Strongly convex functions satisfy the following inequality: for any $x, y \in \mathbb{R}^m$, it holds

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\tau}{2}||x - y||^2$$

for some $\tau > 0$. Note that because Inequality (4.1) provides a similar upper bound on $f(y) - f(x)$ it must hold that $\tau \leq L$ for a smooth f. The strong convexity inequality above implies the Polyak-Łojasiewicz inequality. Indeed, for a fixed x, minimizing both sides of the inequality *w.r.t.* y yields $M^* - f(x)$ on the left-hand-side and the quadratic right-hand-side is minimized for $y = x - \frac{1}{\tau}\nabla f(x)$, yielding a minimal value of $-\frac{1}{2\tau}||\nabla f(x)||^2$, which shows that $M^* - f(x) \geq -\frac{1}{2\tau}||\nabla f(x)||^2$ which is the desired Polyak-Łojasiewicz inequality.

Therefore, convergence of gradient descent with step-size $1/L$ is guaranteed for strongly convex functions at a rate of $(1 - \tau/L)^k$. For strongly convex functions the minimizer is unique, and applying the strong convexity inequality with $y = x_k$ and $x = x^*$, since $\nabla f(x^*) = 0$ it holds that $\frac{\tau}{2}||x_k - x^*||^2 \leq f(x_k) - M^* \leq (1 - \tau/L)^k (f(x_0) - M^*)$, which shows convergence in (squared) distance at the same rate.

Generalized Łojasiewicz condition. The Polyak-Łojasiewicz condition can be generalized into the following form: there exists constants $\tau, \gamma > 0$ such that for all $x \in \mathbb{R}^m$ it holds

$$\frac{1}{2}||\nabla f(x)||^2 \geq \tau(f(x) - M^*)^\gamma. \tag{4.4}$$

This is known as a Łojasiewicz inequality. While it is not ubiquitous in the convex optimization literature, it is of particular relevance to the work we present in Section 4.4. For smooth functions, this inequality also ensures convergence of gradient descent. Indeed, calling $\Delta_k := f(x_k) - M^*$ the sub-optimality gap at iteration k, using the descent property (4.2) with $\eta = 1/L$ and the Łojasiewicz inequality, it holds that $\Delta_k - \Delta_{k+1} \geq \frac{\tau}{L}\Delta_k^\gamma$. Since the sequence $(\Delta_k)_{k\geq 0}$ is non-negative and decreasing, it must converge to some limit $\Delta_\infty \geq 0$. By taking the limit in the previous inequality, it holds $0 \leq \frac{\tau}{L}(\Delta_\infty)^\gamma \leq 0$ which implies that this limit is $\Delta_\infty = 0$ which entails $f(x_k) \to M^*$.

For the convergence rate two cases appear: $\gamma \in (0, 1]$ and $\gamma \in (1, \infty)$. The case $\gamma \in (0, 1]$ can be dealt with as for $\gamma = 1$ since for large enough k, $\Delta_k \in [0, 1]$, and thus $\Delta_k - \Delta_{k+1} \geq \frac{\tau}{L}{\Delta_k}^\gamma \geq \frac{\tau}{L}\Delta_k$, which leads to $\Delta_k \leq (1 - \tau/L)^{k-k_0}\Delta_{k_0}$ for $k \geq k_0$.

The following lemma allows to provide a convergence rate in the case $\gamma > 1$.

Lemma 4.3.1. *Let $(\phi_k)_{k\geq 0}$ be a sequence of positive numbers satisfying the inequality: $\phi_k - \phi_{k+1} \geq c\phi_k^\gamma$ for some constants $c > 0$, $\gamma > 1$. Then, there is a constant $C > 0$ such that for any $k \geq 1$ it holds that*

$$0 \leq \phi_k \leq \left(\phi_0^{1-\gamma} + kc(\gamma - 1)\right)^{-\frac{1}{\gamma-1}} \leq \left(\frac{C}{k}\right)^{\frac{1}{\gamma-1}}.$$

Proof. By convexity, $u \mapsto u^{1-\gamma}$ is above its tangent curves on $(0, \infty)$ and therefore $v^{1-\gamma} - u^{1-\gamma} \geq (\gamma - 1)u^{-\gamma}(u - v)$ for any $u, v > 0$. It thus holds that

$$\begin{aligned}
\phi_{k+1}^{1-\gamma} - \phi_k^{1-\gamma} &\geq (\gamma - 1)\phi_k^{-\gamma}(\phi_k - \phi_{k+1}), \\
&\geq (\gamma - 1)\phi_k^{-\gamma} c\phi_k^\gamma, \\
&\geq c(\gamma - 1).
\end{aligned}$$

From this we get

$$\phi_k^{1-\gamma} \geq \phi_0^{1-\gamma} + kc(\gamma - 1),$$

and thus

$$0 \leq \phi_k \leq \left(\phi_0^{1-\gamma} + kc(\gamma - 1)\right)^{-\frac{1}{\gamma-1}} \leq (kc(\gamma - 1))^{-\frac{1}{\gamma-1}}.$$

□

4.3.2 . Gradient descent without Łojasiewicz-type assumptions

In finite dimension, the strict descent property is enough to guarantee convergence to a minimum for convex functions. Indeed, in this case it can be shown that a Łojasiewicz condition holds with a power $\gamma = 2$ for the iterates generated by gradient descent, meaning that there is a constant $\tau > 0$ such that for any k,

$\tau(f(x_k) - M^*)^2 \leq \frac{1}{2}||\nabla f(x_k)||^2$. We give the proof of the Łojasiewicz inequality for the iterates with $\gamma = 2$ below. First, let us start with the inequality

$$f(x) - f(y) \leq \langle \nabla f(x), x - y \rangle - \frac{1}{2L}||\nabla f(x) - \nabla f(y)||^2, \tag{4.5}$$

which holds for any $x, y \in \mathbb{R}^m$.

Proof. Using the convexity of f and Inequality (4.1) we get

$$\begin{aligned} f(x) - f(y) &= f(x) - f(z) + f(z) - f(y), \\ &\leq \langle \nabla f(x), x - z \rangle + \langle \nabla f(y), z - y \rangle + \frac{L}{2}||z - y||^2. \end{aligned}$$

Minimizing the right-hand-side over z gives $z = y - \frac{1}{L}(\nabla f(y) - \nabla f(x))$, and substituting back in the inequality above yields the desired result. □

Now that we have the Inequality (4.5), since $\nabla f(x^*) = 0$, it follows by taking $y = x^*$ that $f(x) - f(x^*) \leq \langle \nabla f(x), x - x^* \rangle - \frac{1}{2L}||\nabla f(x)||^2$ and thus $\frac{1}{2L}||\nabla f(x)||^2 - \langle \nabla f(x), x - x^* \rangle \leq f(x^*) - f(x) \leq 0$. Considering the gradient descent iterates x_k with step-size $1/L$, we have

$$\begin{aligned} ||x_{k+1} - x^*||^2 &= ||x_k - x^*||^2 + \frac{1}{L^2}||\nabla f(x_k)||^2 - \frac{2}{L}\langle \nabla f(x_k), x_k - x^* \rangle, \\ &= ||x_k - x^*||^2 + \frac{2}{L}\left(\underbrace{\frac{1}{2L}||\nabla f(x_k)||^2 - \langle \nabla f(x_k), x_k - x^* \rangle}_{\leq 0} \right), \\ &\leq ||x_k - x^*||^2. \end{aligned}$$

This shows that the distance of the iterates to any of the minimizers of f decreases during gradient descent. Finally, by convexity, it holds

$$\begin{aligned} f(x_k) - f(x^*) &\leq \langle \nabla f(x_k), x_k - x^* \rangle \leq ||\nabla f(x_k)||\,||x_k - x^*||, \\ &\leq ||\nabla f(x_k)||\,||x_0 - x^*||, \end{aligned}$$

which is equivalent to a Łojasiewicz inequality with $\gamma = 2$ and $\tau = 1/(2||x_0 - x^*||^2)$. This ensures the convergence of gradient descent towards the global minimum at a rate of $O(1/k)$ by Lemma 4.3.1.

4.3.3 . Coordinate descent

Coordinate descent is a popular variant of the gradient descent algorithm and involves sampling a coordinate (or a sub-group of coordinates) and descending along the projected gradient onto those coordinates. Many different flavours of coordinate descent exist as well as various proof techniques and convergence properties. We refer to (Wright, 2015) for a comprehensive review. In this chapter, we consider the following algorithm:

$$x_{k+1} = x_k - \eta \nabla_{i_k} f(x_k) e_{i_k}$$

where i_k is the selected coordinate at iteration, sampled uniformly in $[1, m]$ $\nabla_{i_k} f = \partial f / \partial x_{i_k}$, and e_i is the i-th basis vector of the canonical basis of $\mathbb{R}^m$ for any $i \in [1, m]$ where $[1, m] := \{1, 2, \ldots, m\}$. As we review below, coordinate descent methods essentially have the same convergence rates, in expectation, as for full-gradient methods but with constants which have a bad dependency in the dimension m: they increase (often linearly) with m.

We first derive the descent inequality for coordinate descent: applying Inequality (4.1) with $u = te_i$ yields for smooth functions:

$$f(x + te_i) \leq f(x) + t\nabla_i f(x) + \frac{L}{2}t^2$$

and minimizing the right-hand-side over t yields $t = -\frac{1}{L}\nabla_i f(x)$ and the descent property

$$f(x - \frac{1}{L}(\nabla_i f(x))e_i) \leq f(x) - \frac{1}{2L}(\nabla_i f(x))^2. \tag{4.6}$$

Therefore, it holds for the coordinate descent algorithm with step-size $\eta = 1/L$ that

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}(\nabla_{i_k} f(x_k))^2.$$

Taking the expectation conditionally on x_k yields

$$\begin{aligned}
\mathbb{E}[f(x_{k+1})|x_k] &\leq f(x_k) - \frac{1}{2L}\mathbb{E}_{i_k}[(\nabla_{i_k} f(x_k))^2], \\
&\leq f(x_k) - \frac{1}{2mL}\sum_{i=1}^{m}(\nabla_i f(x_k))^2, \\
&\leq f(x_k) - \frac{1}{2mL}||\nabla f(x_k)||^2.
\end{aligned}$$

Now, finally taking the expectation over x_k yields

$$\mathbb{E}[f(x_{k+1})] \leq \mathbb{E}[f(x_k)] - \frac{1}{2mL}\mathbb{E}[||\nabla f(x_k)||^2], \tag{4.7}$$

which is essentially the same as the descent property (4.1) in expectation with the constant L replaced by mL. This is due to the sampling of a single coordinate at each time step: it takes m-times more iterations to compute the whole gradient compared to gradient descent. However, each iteration is less compute-intensive as it requires the computation of a single gradient coordinate.

Polyak-Łojasiewicz condition. If f satisfies the Polyak-Łojasiewicz Inequality (4.3), if follows from the descent property in expectation (4.7) that

$$\mathbb{E}[f(x_{k+1})] \leq \mathbb{E}[f(x_k)] - \frac{\tau}{2mL}\mathbb{E}[f(x_k) - M^*],$$

and thus, calling $\Delta_k := \mathbb{E}[f(x_k) - M^*]$, it holds $\Delta_{k+1} \leq \left(1 - \frac{\tau}{mL}\right) \Delta_k$ which entails $\Delta_k \leq \left(1 - \frac{\tau}{mL}\right)^k \Delta_0$. This is essentially the same convergence rate as for gradient descent, but we observe that the constant now depends on m in an unfavorable way: $1 - \tau/(mL)$ increases with m towards 1 which implies a slower decrease of the optimality gap.

Generalized Łojasiewicz condition. If f satisfies the Łojasiewicz Inequality (4.4), it holds

$$\mathbb{E}[f(x_{k+1})] \leq \mathbb{E}[f(x_k)] - \frac{\tau}{2mL}\mathbb{E}[(f(x_k) - M^*)^\gamma].$$

If $\gamma > 1$, : $u \mapsto u^\gamma$ is convex and Jensen's inequality ensures that $\mathbb{E}[(f(x_k) - M^*)^\gamma] \geq \mathbb{E}[(f(x_k) - M^*)]^\gamma$. Thus, calling $\Delta_k := \mathbb{E}[f(x_k) - M^*]$, it holds $\Delta_{k+1} \leq \Delta_k - \frac{\tau}{mL}\Delta_k^\gamma$, and Lemma 4.3.1 guarantees that $\Delta_k \leq \left(\Delta_0^{1-\gamma} + \frac{\tau(\gamma-1)}{mL}k\right)^{-\frac{1}{\gamma-1}}$. This is similar to gradient descent but now with a dependency in m which is again unfavourable (slower decrease as m grows larger).

Plain convexity without Łojasiewicz-type assumptions. With a similar argument to the one leading to the descent property in expectation (4.7), and using Inequality (4.5), it holds

$$\mathbb{E}\left[||x_{k+1} - x^*||^2\right] \leq \mathbb{E}\left[||x_k - x^*||^2\right] \leq \mathbb{E}\left[||x_0 - x^*||^2\right].$$

Then, with the same reasoning as in Section 4.3.2, the convexity of f and Cauchy-Schwarz's inequality ensure that

$$\begin{aligned}\mathbb{E}\left[f(x_k) - M^*\right]^2 &\leq \mathbb{E}\left[||\nabla f(x_k)||\,||x_k - x^*||\right]^2, \\ &\leq \mathbb{E}\left[||\nabla f(x_k)||^2\right]\mathbb{E}\left[||x_0 - x^*||^2\right].\end{aligned}$$

Defining $\tau = 1/(2\mathbb{E}\left[||x_0 - x^*||^2\right])$, and plugging the inequality above into Inequality (4.7), it holds:

$$\Delta_{k+1} \leq \Delta_k - \frac{\tau}{mL}\Delta_k^2,$$

which by Lemma 4.3.1 ensures convergence at a rate of $O(\frac{mL}{\tau k})$.

4.3.4 . Proximal methods

We now review the techniques associated to proximal methods. In this section, we consider instead of the objective f, the composite objective $g = f + h$ where f is convex and smooth (as before) and h is convex but not smooth and "easy" to optimize. This includes penalized objectives such as $f(x) + \lambda||x||_1$ where $||x||_1 = \sum_{i=1}^m |x_i|$ is the L^1 norm of x. In this setting, one cannot rely solely on the gradient of f to minimize the objective. However, a variant of Inequality (4.1) still holds:

$$g(y) - g(x) \leq \langle \nabla f(x), y - x\rangle + \frac{L}{2}||y - x||^2 + h(y) - h(x). \tag{4.8}$$

Let us define $\hat{\mathcal{D}}(x,y) := -2L(\langle \nabla f(x), y-x\rangle + \frac{L}{2}||y-x||^2 + h(y) - h(x))$ and $\mathcal{D}(x) := -2L \min_y -\frac{1}{2L}\hat{\mathcal{D}}(x,y)$, which is the minimal value, if it exists, of the upper bound on the right-hand-side. Depending on the form h has, minimizers might not exist, might not be unique or their value or that of the minimum might be difficult to write explicitly. But in certain cases, a unique minimum can be guaranteed to exist with a tractable expression.

We first review the methods when we assume minimizers are well-defined and then give the example of the L^1-penalty which is particularly relevant to our setting. First we observe that since $\hat{\mathcal{D}}(x,x) = 0$, $\mathcal{D}(x)$ must be **non-negative**. Next observe that the Inequality (4.8) can be written, for any $x, y \in \mathbb{R}^m$, as

$$g(y) \leq g(x) - \frac{1}{2L}\hat{\mathcal{D}}(x,y)$$

and thus that for any minimizer $T(x)$ of $-\frac{1}{2L}\hat{\mathcal{D}}(x,y)$ *w.r.t.* y, it holds

$$g(T(x)) \leq g(x) - \frac{1}{2L}\mathcal{D}(x), \tag{4.9}$$

so that there is a descent property. We thus consider proximal algorithms of the form

$$x_{k+1} \in \operatorname*{argmin}_y \hat{\mathcal{D}}(x_k, y).$$

When a unique minimizer exists, this is often written in the form

$$x_{k+1} = \mathsf{prox}_{1/L}\left(x_k - \frac{1}{L}\nabla f(x_k)\right),$$
$$\mathsf{prox}_\eta(z) := \operatorname*{argmin}_{y\in\mathbb{R}^m}\left\{\frac{1}{2}||y-z||^2 + \eta h(z)\right\}.$$

While, many different convergence proofs exist for proximal methods depending on the context and the assumptions, it is highlighted in (Karimi et al., 2016) that Polyak-Łojasiewicz-type assumptions provide a simple setting to cover a wide range of different assumptions as well as simpler arguments and streamlined proofs. We review the main arguments involved under Polyak-Łojasiewicz-type assumptions.

Proximal Polyak-Łojasiewicz condition. There is a variant of the Polyak-Łojasiewicz condition in the case of proximal algorithms which can be expressed as follows: there exists a constant $\tau > 0$ such that, for any $x \in \mathbb{R}^m$, it holds

$$\frac{1}{2}\mathcal{D}(x) \geq \tau(g(x) - M^*), \tag{4.10}$$

where $M^* = \min_x g(x)$ is assumed to be attained in at least one x^*. Similarly to the case of the gradient descent algorithm, this inequality is enough to guarantee

convergence of the proximal algorithm. Indeed, it follows from Inequality (4.10) that

$$g(x_{k+1}) \leq g(x_k) - \frac{\tau}{L}(g(x_k) - M^*),$$

which yields $g(x_k) - M^* \leq (1 - \tau/L)^k (g(x_0) - M^*)$ as with gradient descent.

Proximal Łojasiewicz condition. The equivalent of the generalized Łojasiewicz condition for proximal algorithms has the following form: there exists $\tau > 0$ and $\gamma > 0$ such that, for any $x \in \mathbb{R}^m$, it holds

$$\frac{1}{2}\mathcal{D}(x) \geq \tau(g(x) - M^*)^\gamma.$$

As for gradient descent, calling $\Delta_k := g(x_k) - M^*$, it holds that $\Delta_{k+1} \leq \Delta_k - \frac{\tau}{L}\Delta_k^\gamma$. The same reasoning as in the case of gradient descent guarantees the convergence towards a global minimum at a rate of $O\left((1 - \frac{\tau}{L})^k\right)$ if $\gamma \in (0, 1]$ and $O\left(k^{-\frac{1}{\gamma - 1}}\right)$ if $\gamma > 1$.

Convergence with simple convexity. We review here the ideas presented in (Richtárik and Takáč, 2014) for the proof of convergence when f is simply convex with the additional assumption that the iterates x_k generated by the proximal gradient algorithm are bounded, which ensures that $||x_k - x^*|| \leq R$ for some $R > 0$ (which might depend on x^*) for any minimizer x^* of g.

Remark. The boundedness assumption on $||x_k||$ is not unreasonable. In the typical setting where f is lower-bounded by some M_f and $h(x) = \lambda N(x)$ is proportional to some norm N on $\mathbb{R}^m$, any descent algorithm would ensure that

$$M_f + \lambda N(x_k) \leq f(x_k) + \lambda N(x_k) = g(x_k) \leq g(x_0) = f(x_0) + \lambda N(x_0),$$

so that by equivalence of the norms in finite dimension, there is a constant $C > 0$ such that $||x_k|| \leq CN(x_k) \leq \frac{C}{\lambda}(f(x_0) + \lambda N(x_0) - M_f)$ for any k.

The main idea of the proof is to show that a proximal Łojasiewicz-type inequality holds with $\gamma = 2$ and $\eta = 1/L$ for the iterates of the proximal gradient algorithm, that is for any k,

$$\frac{1}{2}\mathcal{D}(x_k) \geq \tau(g(x_k) - g^*)^2$$

for some $\tau > 0$. This ensures convergence at a rate of $O(k^{-1})$ by the previous paragraph. This is similar to the setting of smooth and convex functions, except the proof technique is much different. We first introduce the following lemma which shows a Łojasiewicz-type inequality but with a constant τ which depends on x:

Lemma 4.3.2 (Lower bound on the proximal descent)**.** *For any x which is not a minimizer of g, it holds:*

$$\frac{1}{2}\mathcal{D}(x) \geq \frac{L}{2}\min\left(\frac{1}{g(x)-M^*}, \frac{1}{L||x-x^*||^2}\right)(g(x)-M^*)^2.$$

Proof.

$$\begin{aligned}
-\frac{1}{2L}\mathcal{D}(x) &= \min_y \langle \nabla f(x), y-x\rangle + \frac{L}{2}||y-x||^2 + h(y) - h(x) \\
&= \min_y f(x) + \langle \nabla f(x), y-x\rangle + \frac{L}{2}||y-x||^2 + h(y) - h(x) - f(x) \\
&\leq \min_y f(y) + \frac{L}{2}||y-x||^2 + h(y) - g(x) \\
&= \min_y g(y) + \frac{L}{2}||y-x||^2 - g(x) \\
&\leq \min_{\alpha\in[0,1]} g(\alpha x^* + (1-\alpha)x) + \frac{L}{2}\alpha^2||x-x^*||^2 - g(x) \\
&\leq \min_{\alpha\in[0,1]} \alpha g(x^*) + (1-\alpha)g(x) + \frac{L}{2}\alpha^2||x-x^*||^2 - g(x) \\
&= \min_{\alpha\in[0,1]} -\alpha(g(x)-M^*) + \frac{L}{2}\alpha^2||x-x^*||^2
\end{aligned}$$

The minimum of the final expression on the right-hand-side is obtained for $\alpha^* = \min\left(1, \frac{g(x)-M^*}{L||x-x^*||^2}\right)$. The corresponding minimal value is thus equal to:

$$\begin{cases} -(g(x)-M^*) + \frac{L}{2}||x-x^*||^2 & \text{if } \alpha^* = 1 \\ -\frac{(g(x)-M^*)^2}{2L||x-x^*||^2} & \text{if } \alpha^* = \frac{g(x)-M^*}{L||x-x^*||^2} \end{cases}.$$

Note that in the first case, if $\alpha^* = 1$, then $\frac{g(x)-M^*}{L||x-x^*||^2} \geq 1$, which means $L||x-x^*||^2 \leq g(x)-M^*$, so that the minimum in that case is $\leq -\frac{g(x)-M^*}{2}$. In any case, it holds that $-\frac{1}{2L}\mathcal{D}(x)$ is less than the largest of the two possible values:

$$-\frac{1}{2L}\mathcal{D}(x) \leq \max\left(-\frac{g(x)-M^*}{2}, -\frac{(g(x)-M^*)^2}{2L||x-x^*||^2}\right).$$

The result stated in the lemma readily follows. □

The proximal Łojasiewicz inequality then follows from the lemma above and the boundedness assumption on $||x_k - x^*||$. Indeed, defining the constant $\tau := \frac{L}{2}\min\left(\frac{1}{g(x_0)-M^*}, \frac{1}{LR^2}\right)$, it holds that for any k

$$\frac{L}{2}\min\left(\frac{1}{g(x_k)-M^*}, \frac{1}{L||x_k-x^*||^2}\right) \geq \tau.$$

This inequality stems from the fact that $||x_k - x^*||^2 \leq R^2$ and that $(g(x_k))_{k\geq 0}$ is a decreasing sequence so that $g(x_k) - M^* \leq g(x_0) - M^*$. Therefore, it follows from Lemma 4.3.2 and the descent property (4.9) that

$$g(x_{k+1}) - M^* \leq g(x_k) - M^* - \frac{\tau}{L}(g(x_k) - M^*)^2.$$

Lemma 4.3.1 then ensures that $g(x_k) - M^*$ converges towards 0 at a rate of $O(1/k)$.

Proximal coordinate descent. As in the case of smooth functions, coordinate descent variants of the proximal algorithm lead to the same rate of convergence for the expected optimality gap as the standard proximal gradient method but with constants which have unfavourable dependence on the dimension m. The proof technique is the same as for smooth functions but requires an additional assumption on h, namely that it is *separable*, *i.e.*, $h(x) = \sum_{i=1}^m h_i(x_i)$ with $h_i : \mathbb{R} \to \mathbb{R}$. First, it follows from Inequality (4.8) with $y = x + te_i$ that:

$$g(x + te_i) \leq g(x) - \frac{1}{2L}\hat{\mathcal{D}}(x, x + te_i).$$

Defining, if they exist, $\mathcal{D}_i(x) := -2L \min_{t\in\mathbb{R}} -\frac{1}{2L}\hat{\mathcal{D}}(x, x + te_i)$ and $T_i(x) := \operatorname{argmin}_{t\in\mathbb{R}} -\frac{1}{2L}\hat{\mathcal{D}}(x, x + te_i)$, it holds that

$$\sum_{i=1}^m \mathcal{D}_i(x) = \mathcal{D}(x).$$

Indeed, the separability of h ensures that $h(x + te_i) - h(x) = h_i(x_i + t) - h_i(x_i)$, so that

$$-\frac{1}{2L}\hat{\mathcal{D}}(x, x + te_i) = t\nabla_i f(x) + \frac{L}{2}t^2 + h_i(x_i + t) - h_i(x_i).$$

This leads to $\sum_{i=1}^m \hat{\mathcal{D}}(x, x + u_i e_i) = \hat{\mathcal{D}}(x, x + u)$ from which it easily follows that $\sum_{i=1}^m \mathcal{D}_i(x) = \mathcal{D}(x)$. We define the proximal coordinate descent algorithm via

$$x_{k+1} = x_k + T_{i_k}(x_k)e_{i_k} = \operatorname*{argmin}_{y = x_k + te_{i_k}} -\frac{1}{2L}\hat{\mathcal{D}}(x_k, y),$$

where i_k is sampled uniformly over $[1, m]$ at each iteration. It thus holds that

$$g(x_{k+1}) \leq g(x_k) - \frac{1}{2L}\mathcal{D}_{i_k}(x_k).$$

Independently of the coordinate sampled, this provides a descent step since for $t = 0$, $-\frac{1}{2L}\hat{\mathcal{D}}(x, x + te_i) = \frac{1}{2L}\hat{\mathcal{D}}(x, x) = 0$. Taking the expectation over the sampling of i_k, conditionally on x_k, we have

$$\mathbb{E}[g(x_{k+1})|x_k] \leq g(x_k) - \frac{1}{2Lm}\sum_{i=1}^m \mathcal{D}_i(x_k) = g(x_k) - \frac{1}{2Lm}\mathcal{D}(x_k).$$

If the proximal Polyak-Łojasiewicz condition or the generalized proximal Łojasiewicz condition holds, using the inequality above, the same proof technique as for standard coordinate descent (see Section 4.3.3) yields convergence to the global minimum in expectation at a rate of $(1 - \frac{\tau}{mL})^k$ in the first case, and $O((\frac{mL}{\tau k})^{\frac{1}{\gamma-1}})$ in the second case. Finally, in the case where we only assume that the iterates x_k generated by the proximal coordinate descent algorithm are bounded, the same arguments as for the full proximal gradient method ($||x_k - x^*||$ is bounded, and $(g(x_k))_{k\geq 0}$ is decreasing) ensure that a Łojasiewicz-type inequality holds for the iterates generated by proximal coordinate descent, in that case with $\gamma = 2$, guaranteeing a convergence at a rate of $O(\frac{mL}{\tau k})$.

Example of an L^1-penalty. The penalty $h(x) = \lambda||x||_1 = \sum_{i=1}^m |x_i|$ fits the criteria for the proximal algorithm: it is non-smooth, separable and easily optimized. Minimizing the upper bound $-\frac{1}{2L}\hat{\mathcal{D}}(x, x + te_i)$ over $t \in \mathbb{R}$ is equivalent to finding

$$T_i(x) = \operatorname*{argmin}_{t\in\mathbb{R}} t\nabla_i f(x) + \frac{L}{2}t^2 + \lambda|t + x_i|.$$

In this case, it is known that the minimizer is unique and given by

$$T_i(x) = -x_i + \left(x_i - \frac{1}{L}\nabla_i f(x)\right) \max\left(0, 1 - \frac{\lambda}{|Lx_i - \nabla_i f(x)|}\right). \tag{4.11}$$

The term $T_i(x) + x_i$ is akin to the result of a *soft thresholding* (see *e.g.*, Bredies and Lorenz, 2008) of $x_i - \frac{1}{L}\nabla_i f(x)$. See Appendix C.1 for the proof.

4.4 . Coordinate descent in the space of measures

We now present an algorithm to minimize a smooth and convex objective $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$ on the space of signed measures over the sphere as introduced in Section 4.2. The Franck-Wolfe algorithm (see Bach, 2017) is one approach to solve the problem, providing a convergence guarantee at a rate of $1/k$ but where each iteration has exponential computational complexity. In this section, we present a coordinate descent algorithm where the computational cost of an iteration is polynomial but the "curse of dimensionality" occurs in the convergence rate we obtain in $k^{-1/d}$.

We assume that there is at least one minimizer μ^* realizing the minimum $F^* = \min_{\mu\in\mathcal{M}(\mathbb{S}^{d-1})} F(\mu)$. We first derive the analog of Inequality (4.1) through the following lemma:

Lemma 4.4.1 (Smoothness inequality)**.** *For a smooth $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$, it holds for any $\mu, \nu \in \mathcal{M}(\mathbb{S}^{d-1})$:*

$$F(\nu) \leq F(\mu) + \int V[\mu]\mathrm{d}(\nu - \mu) + \frac{L}{2}|\nu - \mu|_{TV}^2. \tag{4.12}$$

Proof. The proof is similar to the finite-dimensional case: by definition of the first variation V of F, it holds $\frac{d}{dt}F(\mu + t(\nu - \mu)) = \int V[\mu + t(\nu - \mu)]\mathrm{d}(\nu - \mu)$, and thus:

$$\begin{aligned}
F(\nu) - F(\mu) &= \int_0^1 \int V[\mu + t(\nu - \mu)]\mathrm{d}(\nu - \mu)\mathrm{d}t, \\
&= \int V[\mu]\mathrm{d}(\nu - \mu) + \int_0^1 \int \left(V[\mu + t(\nu - \mu)] - V[\mu]\right)\mathrm{d}(\nu - \mu)\mathrm{d}t, \\
&\leq \int V[\mu]\mathrm{d}(\nu - \mu) + \int_0^1 ||V[\mu + t(\nu - \mu)] - V[\mu]||_\infty \, |\nu - \mu|_{TV}\mathrm{d}t, \\
&\leq \int V[\mu]\mathrm{d}(\nu - \mu) + \int_0^1 Lt|\nu - \mu|_{TV}^2\mathrm{d}t, \\
&\leq \int V[\mu]\mathrm{d}(\nu - \mu) + \frac{L}{2}|\nu - \mu|_{TV}^2.
\end{aligned}$$

□

As such, the infimum over $\nu \in \mathcal{M}(\mathbb{S}^{d-1})$ of the upper bound $\int V[\mu]\mathrm{d}(\nu - \mu) + \frac{L}{2}|\nu - \mu|_{TV}^2$ on $F(\nu) - F(\mu)$ provided by Inequality (4.12) is not always tractable, in particular because $|\mu - \nu|_{TV}^2$ does not always have a simple expression (it is analogous to a squared L^1 norm in finite dimension). We discuss below cases of upper bounds where the infimum is computable explicitly.

Square-integrable functions. Let ω be a reference probability measure on $\mathbb{S}^{d-1}$ such as the uniform measure on the sphere. If $\mu, \nu \in L^1(\omega)$, denoting by f_μ and f_ν their densities *w.r.t.* ω, it holds $|\nu - \mu|_{TV} = \int |f_\nu - f_\mu|\mathrm{d}\omega = ||f_\nu - f_\mu||_{L^1(\omega)}$. If $\mu, \nu \in L^2(\omega) \subset L^1(\omega)$, then $||f_\nu - f_\mu||_{L^1(\omega)} \leq ||f_\nu - f_\mu||_{L^2(\omega)}$ by Jensen's inequality, and thus we obtain an upper bound which is more amenable to optimization for square-integrable functions:

$$\min_{\nu \in L^2(\omega)} \int V[\mu]\mathrm{d}(\nu - \mu) + \frac{L}{2}||\nu - \mu||_{L^2(\omega)}^2 = -\frac{1}{2L}\int V[\mu]^2\mathrm{d}\omega = -\frac{1}{2L}||V[\mu]||_{L^2(\omega)}^2.$$

The minimum is obtained when ν has density $-\frac{1}{L}V[\mu]$ *w.r.t.* ω, which is reminiscent of the case of smooth functions in finite-dimension. Indeed, w:hen $\mu, \nu \in L^2(\omega)$, denoting f_μ and f_ν the (square-integrable) densities of μ and ν *w.r.t.* to ω, it holds:

$$\begin{aligned}
\int V[\mu]\mathrm{d}(\nu - \mu) + \frac{L}{2}||\nu - \mu||_{L^2(\omega)}^2 &= \int V[\mu](f_\nu - f_\mu)\mathrm{d}\omega + \frac{L}{2}\int (f_\nu - f_\mu)^2\mathrm{d}\omega, \\
&= \int \left(V[\mu](f_\nu - f_\mu) + \frac{L}{2}(f_\nu - f_\mu)^2\right)\mathrm{d}\omega.
\end{aligned}$$

The minimization of the left-hand-side over $\nu \in L^2(\omega)$ follows from the *point-wise* minimization of the integrand on the right-hand-side. Indeed, as derived in

Section 4.3, the minimum of : $y \in \mathbb{R} \mapsto a(y - y_0) + \frac{L}{2}(y - y_0)^2$ is obtained for $y = y_0 - \frac{1}{L}a$ and is equal to $-\frac{1}{2L}a^2$. Note that the density $f = f_\mu - \frac{1}{L}V[\mu]$ is indeed in $L^2(\omega)$ since $f_\mu \in L^2(\omega)$ and $V[\mu]$ is continuous over the compact set $\mathbb{S}^{d-1}$ and thus in $L^\infty(\omega)$.

When μ is not in $L^1(\omega)$, there is no explicit expression for $|\nu - \mu|_{TV}$, even when $\nu \in L^2(\omega)$, which makes it difficult to minimize the upper bound explicitly.

Coordinate descent variant. The space $\mathcal{M}(\mathbb{S}^{d-1})$ is infinite-dimensional and there is no clear notion of what a "coordinate" is in that setting. Intuitively, for a measure ν, the mass it puts on a given vector $u \in \mathbb{S}^{d-1}$ could be a candidate for the coordinate along the direction $\delta_u \in \mathcal{M}(\mathbb{S}^{d-1})$, where δ_u is the Dirac measure at u. However, there is a lack of a good "basis" in $\mathcal{M}(\mathbb{S}^{d-1})$, in the sense that there is no reference measure $\omega \in \mathcal{M}(\mathbb{S}^{d-1})$ such that for any $\mu \in \mathcal{M}(\mathbb{S}^{d-1})$, and for any μ-measurable φ, there exists a φ_μ (depending on μ) such that $\int \varphi \mathrm{d}\mu = \int \varphi_\mu(u)\mathrm{d}\omega(u)$, unless we consider the subset $L^1(\omega) \subset \mathcal{M}(\mathbb{S}^{d-1})$. As observed from the finite-dimensional case in Section 4.3.3, the coordinate descent method relies crucially on the fact that the expectation of certain quantities projected along a given coordinate (*e.g.*, norms or gradients) gives back the whole quantity (*e.g.*, $||x||^2 = \sum_i x_i^2$, $\langle \nabla f(x), y \rangle = \sum_i \nabla_i f(x) y_i$). This property is lost in general in the space $\mathcal{M}(\mathbb{S}^{d-1})$ as the "projection" of a measure along δ_u does not make sense.

However, if $\nu \in L^2(\omega)$, this is possible: denoting by f_ν the density, it holds

$$\int V[\mu]\mathrm{d}\nu = \int_u \left(\int V[\mu] f_\nu \mathrm{d}\delta_u \right) \mathrm{d}\omega(u),$$
$$||\nu||^2_{L^2(\omega)} = \int_u \left(\int f_\nu \mathrm{d}\delta_u \right)^2 \mathrm{d}\omega(u)$$

since $G(u) = \int G \mathrm{d}\delta_u$ for any G. This suggests doing coordinate descent by sampling a vector $u \in \mathbb{S}^{d-1}$ and searching for a minimizer of an upper bound on $F(\mu + t\delta_u)$ over $t \in \mathbb{R}$. Plugging $\nu = \mu + t\delta_u$ in Inequality (4.12), we get

$$F(\mu + t\delta_u) - F(\mu) \leq tV[\mu](u) + \frac{L}{2}t^2, \tag{4.13}$$

and minimizing the upper bound on the right-hand-side gives:

$$F(\mu + T_u(\mu)\delta_u) \leq F(\mu) - \frac{1}{2L}V[\mu](u)^2, \tag{4.14}$$
$$T_u(\mu) := -\frac{1}{L}V[\mu](u). \tag{4.15}$$

This is the analog of Inequality (4.6) for finite-dimensional coordinate descent where here the direction $u \in \mathbb{S}^{d-1}$ plays the role of the coordinate $i \in [1, m]$, the Dirac measure δ_u plays the role of the basis vector e_i, $V[\mu]$ plays the role of the gradient $\nabla f(x)$, and $T_u(\mu)$ plays the role of the minimizer $T_i(x)$ *w.r.t.* the selected coordinate.

As presented in Section 1.2.4, when doing gradient flow, ∇V plays the role of the gradient in finite dimension (see Equation (1.7)). So why not here ? This is due to the L^2-geometry which is implicitly at play in this formulation of coordinate descent over $\mathcal{M}(\mathbb{S}^{d-1})$. Indeed, taking the expectation of the coordinate descent step over the sampling of $u \in \mathbb{S}^{d-1}$ *w.r.t.* ω, that is, the minimization of the upper bound on $F(\mu + t\delta_u)$ over t, it holds

$$\begin{aligned}\mathbb{E}_{u\sim\omega}[F(\mu + T_u(\mu))] &= \int_u F(\mu + T_u(\mu))\mathrm{d}\omega(u),\\ &\leq F(\mu) - \frac{1}{2L}\int_u V[\mu](u)^2\mathrm{d}\omega(u),\\ &\leq F(\mu) - \frac{1}{2L}||V[\mu]||^2_{L^2(\omega)}.\end{aligned}$$

This is reminiscent of the inequality obtained in expectation for coordinate descent in finite dimension, but note that the difference here is that summing (or in this case, integrating) $V[\mu](u)^2$ over the "coordinate" u does not give the norm $|V[\mu]|^2_{TV}$ (where $V[\mu]$ is seen as a density *w.r.t.* ω) which appears in the initial bound (4.12), but rather the squared L^2 norm $||V[\mu]||^2_{L^2(\omega)}$ which is larger than $|V[\mu]|^2_{TV}$.

$V[\mu]$ belongs to an infinite-dimensional space, namely $\mathcal{C}(\mathbb{S}^{d-1})$, and multiple choices of norms are possible to upper bound $F(\nu) - F(\mu)$ which are no equivalent as they would be in finite-dimension. The choice of norm is therefore of importance in our setting. This makes sense since the definition of the gradient depends on the choice of norm in infinite dimension (as it is a limit, which inherently depends on the norm), so it is not surprising that different norms lead to different methods in infinite-dimension.

It turns out that minimizing the upper bound of Inequality (4.13) when sampling a random u is the same, in expectation, as minimizing an upper bound with an L^2 norm on $F(\mu+\nu)$ for $\nu \in L^2(\omega)$. Indeed, calling f_ν the density of ν *w.r.t.* ω, since $|\nu|^2_{TV} \leq ||\nu||^2_{L^2(\omega)}$ as discussed in the paragraph on square-integrable functions, it holds for any $\nu \in L^2(\omega)$:

$$F(\mu + \nu) - F(\mu) \leq \int V[\mu] f_\nu \mathrm{d}\omega + \frac{L}{2}\int f_\nu^2 \mathrm{d}\omega,$$

and we have already shown above that the upper bound is minimized by minimizing point-wise the integrand giving the minimizer $T(\mu) = -\frac{1}{L}V[\mu]$ and a minimal value of $-\frac{1}{2L}||V[\mu]||^2_{L^2(\omega)}$, which is the same as the expectation of the minimum of the upper bound derived for coordinate descent.

To make the analogy with Section 1.2.4, if we were to change the norm and minimize the quantity $\int V[\mu]\mathrm{d}(\nu-\mu) + \frac{1}{2\tau}W_2(\nu,\mu)^2$ over $\nu \in \mathcal{P}_2(\mathbb{R}^p)$, as $\tau \to 0^+$, we would obtain for any $\mu \in \mathcal{P}_2(\mathbb{R}^p)$ the minimizer ν^* as the pushforward of μ by the map $-\nabla V[\mu]$, which corresponds to the gradient *w.r.t.* the Wasserstein-2 metric, and not the L^2 metric.

4.4.1 . Convergence of the coordinate descent method

In this section, we extend the techniques reviewed in Section 4.3 in finite-dimension to provide a convergence proof for coordinate descent in $\mathcal{M}(\mathbb{S}^{d-1})$. We assume here that the reference measure ω from which we sample is the **uniform measure** on the sphere $\mathbb{S}^{d-1}$, denoted by ω_d. This leads to the following algorithm: starting from a single atom $\mu_0 = c_0\delta_{u_0}$, at each iteration k we sample a new $u_{k+1} \in \mathbb{S}^{d-1}$ uniformly and set

$$\mu_{k+1} = \mu_k + c_{k+1}\delta_{u_{k+1}} = \underset{\nu=\mu_k+t\delta_{u_{k+1}}}{\operatorname{argmin}} \int V[\mu_k]\mathrm{d}(\nu-\mu_k) + \frac{L}{2}|\nu-\mu_k|_{TV}^2,$$

where $c_{k+1} = T_{u_{k+1}}(\mu_k)$ is set using Equation (4.15). By the descent property (4.14), it holds:

$$F(\mu_{k+1}) \leq F(\mu_k) - \frac{1}{2L}V[\mu_k](u_{k+1})^2. \tag{4.16}$$

We make the following assumptions:

Assumption 6 (Uniform Lipschitzness of the first variation)**.** There exists a constant $K > 0$ such that for any $\mu \in \mathcal{M}(\mathbb{S}^{d-1})$, $V[\mu]$ is K-Lipschitz, *i.e.*, for any $u_1, u_2 \in \mathbb{S}^{d-1}$,

$$|V[\mu](u_1) - V[\mu](u_2)| \leq K||u_1 - u_2||.$$

Assumption 7 (Boundedness of the norms)**.** The iterates μ_k generated by coordinate descent have bounded total variation: there exists a constant $R > 0$ such that for any k, $|\mu_k - \mu^*|_{TV} \leq R$.

Łojasiewicz inequality for the iterates of coordinate descent. We show that the iterates μ_k generated by coordinate descent satisfy a Łojasiewicz inequality with $\gamma = d + 1$. This is summarized in the following lemma:

Lemma 4.4.2 (Łojasiewicz inequality for μ_k)**.** *There exists a constant $\tau > 0$ such that for any k, it holds*

$$\frac{1}{2}||V[\mu_k]||_{L^2(\omega_d)}^2 \geq \tau(F(\mu_k) - F^*)^{d+1}.$$

Proof. By convexity of F and Assumption 7, we have

$$F(\mu_k) - F^* \leq \int V[\mu_k]\mathrm{d}(\mu_k - \mu^*) \leq ||V[\mu_k]||_\infty|\mu_k - \mu^*|_{TV} \leq R||V[\mu_k]||_\infty.$$

Call $M := ||V[\mu_k]||_\infty$ and let $v_k \in \mathbb{S}^{d-1}$ such that $|V[\mu_k](v_k)| = ||V[\mu_k]||_\infty$. By Inequality (4.16), we have $V[\mu_k](u_k)^2 \leq 2L(F(\mu_k) - F(\mu_{k+1})) \leq 2L(F(\mu_0) - F^*)$, and using Assumption 6, it holds for any $u \in \mathbb{S}^{d-1}$ that $|V[\mu_k](u)| \leq 2K + \sqrt{2L(F(\mu_0) - F^*)}$. Since the right-hand-side $K' := 2K + \sqrt{2L(F(\mu_0) - F^*)}$ is

larger than K, have both $|V[\mu](u)-V[\mu](v)|_\infty \leq K'||u-v||$ and $||V[\mu_k]||_\infty \leq K'$ for any k.

Defining $g(u) = \max(0, M - K'||u - v_k||)$, since $|V[\mu_k](u) - V[\mu_k](v_k)| \leq K'||u-v_k||$, it holds $|V[\mu_k](u)| \geq g(u) \geq 0$, and thus $||V[\mu_k]||^2_{L^2(\omega_d)} \geq ||g||^2_{L^2(\omega_d)}$. We now compute the latter term and lower bound it by a constant times M^{d+1}:

$$\begin{aligned} ||g||^2_{L^2(\omega_d)} &= \int \max(0, M - K'||u - v_k||)^2 \mathrm{d}\omega_d(u), \\ &= M^2 \int \max\left(0, 1 - \frac{K'}{M}\sqrt{2(1 - \langle u, v_k\rangle)}\right)^2 \mathrm{d}\omega_d(u). \end{aligned}$$

Since ω_d is the uniform measure on the sphere, we can use some spherical harmonic analysis to simplify the integral. By (Atkinson and Han, 2012)[Theorem 2.22] with $n = 0$, there is a constant $C > 0$ such that

$$\begin{aligned} ||g||^2_{L^2(\omega_d)} &= CM^2 \int_{-1}^{1} \max\left(0, 1 - \frac{K'}{M}\sqrt{2(1-t)}\right)^2 (1-t^2)^{(d-3)/2} \mathrm{d}t \\ &= CM^2 \int_0^2 \max\left(0, 1 - \frac{K'}{M} r\right)^2 r^{d-1} \left(1 - \frac{r^2}{4}\right)^{(d-3)/2} \mathrm{d}r \\ &= CM^2 \frac{M^{d-1}}{(K')^{d-1}} \int_0^{M/K'} \left(1 - \frac{K'}{M} r\right)^2 r^{d-1} \left(1 - \frac{r^2}{4}\right)^{(d-3)/2} \mathrm{d}r \\ &= C \frac{M^{d+1}}{(K')^{d-1}} \int_0^1 (1-s)^2 s^{d-1} \left(1 - \frac{M^2}{(K')^2}\frac{s^2}{4}\right)^{(d-3)/2} \mathrm{d}s \\ &\geq C \frac{M^{d+1}}{(K')^{d-1}} (1/2)^{(d-3)/2} \int_0^1 (1-s)^2 s^{d-1} \mathrm{d}s. \end{aligned}$$

We have used in the second line the change of variable $\sqrt{2(1-t)} = r$, in the third that $1 - (K'/M)r \geq 0 \iff r \leq M/K'$ and that $M/K' \leq 1$, in the penultimate line the change of variable $(K'/M)r = s$, and in the last line that $(1 - M^2 s^2/(2(K')^2)) \geq (1 - M^2/(2(K')^2)) \geq (1 - 1/2) = 1/2$ for $s \in [0, 1]$. Calling $\tau := \frac{CR^{d+1}\int_0^2 (1-s)^2 s^{d-1}\mathrm{d}s}{2^{(d-3)/2}(K')^{d-1}}$ yields the desired inequality since $||V[\mu_k]||^2_{L^2(\omega_d)} \geq ||g||^2_{L^2(\omega_d)}$, and $RM \geq F(\mu_k) - F^*$. □

Remark. With similar arguments, an equivalent result can be obtained in finite-dimension if one considers the analog of Assumption 6. Indeed, in the setting of Section 4.3.3, if we assume that for any $x \in \mathbb{R}^m$, and for any $i, j \in [1, m]$, $|\nabla_i f(x) - \nabla_j f(x)| \leq K|i - j|$, then it holds that

$$\frac{1}{2}||\nabla f(x)||^2 \geq \tau(f(x) - M^*)^3 \tag{4.17}$$

on bounded sets $||x - x^*|| \leq R$. See Appendix C.2 for a complete proof.

Taking the expectation of Inequality (4.16) over the sampling of $u_k \in \mathbb{S}^{d-1}$ *w.r.t.* the uniform measure ω_d (that is, conditionally on μ_k), it holds:

$$\begin{aligned}\mathbb{E}[F(\mu_{k+1})|\mu_k] &\leq F(\mu_k) - \frac{1}{2L}\int V[\mu_k](u)^2 \mathrm{d}\omega_d(u) \\ &\leq F(\mu_k) - \frac{\tau}{L}\left(F(\mu_k) - F^*\right)^{d+1},\end{aligned}$$

and taking now the expectation over μ_k finally gives

$$\begin{aligned}\mathbb{E}[F(\mu_{k+1}) - F^*] &\leq \mathbb{E}[F(\mu_k) - F^*] - \frac{\tau}{L}\mathbb{E}[(F(\mu_k) - F^*)^{d+1}], \\ &\leq \mathbb{E}[F(\mu_k) - F^*] - \frac{\tau}{L}\mathbb{E}\left[F(\mu_k) - F^*\right]^{d+1},\end{aligned}$$

where we have used the convexity of : $z \mapsto z^{d+1}$ and Jensen's inequality in the last line. Lemma 4.3.1 allows to conclude to the following result:

Theorem 4.4.3 (Global convergence of coordinate descent in $\mathcal{M}(\mathbb{S}^{d-1})$)**.** *If Assumptions 6 and 7 hold, then the iterates μ_k generated by coordinate descent satisfy, for any $k \geq 1$,*

$$0 \leq \mathbb{E}[F(\mu_k) - F^*] \leq \frac{C}{k^{1/d}}$$

for some constant $C > 0$.

4.4.2 . A proximal algorithm for L^1-penalized coordinate descent

Although coordinate descent provides a global convergence guarantee with an explicit rate, the number of atoms of μ_k grows linearly with k which is an issue in practice. In the context of a two-layer neural network, this would mean having a network where a new neuron is added at each iteration, making it prohibitively expensive computationally to use. In finite dimension, it is known that the L^1-penalty has a sparsifying effect. Here we consider an analogous penalty in infinite dimension given by the total variation norm: that is $F(\mu) = J(\mu) + \lambda|\mu|_{TV}$ where J is smooth. The total variation norm penalty has a similar sparsifying effect as the L^1 penalty in finite dimension and the aim is that adding such a penalty should reduce the number of atoms and therefore the computational cost incurred.

Since the total variation norm is not smooth, we resort to a proximal descent method similarly to what we presented in Section 4.3.4 in finite dimension. As in finite dimension, one can extend Inequality (4.12) with the added penalty, and it holds

$$F(\nu) \leq F(\mu) + \int V[\mu]\mathrm{d}(\nu - \mu) + \frac{L}{2}|\nu - \mu|_{TV}^2 + \lambda|\nu|_{TV} - \lambda|\mu|_{TV}.$$

Applying this inequality to $\nu = \mu + t\delta_u$ yields

$$F(\mu + t\delta_u) - F(\mu) \leq tV[\mu](u) + \frac{L}{2}t^2 + \lambda|\mu + t\delta_u|_{TV} - \lambda|\mu|_{TV}. \tag{4.18}$$

This provides a descent property as the upper bound on the right-hand-side is 0 when $t = 0$ and the minimum over $t \in \mathbb{R}$ must therefore be ≤ 0.

If $\mu \perp \delta_u$, that is if μ and δ_u are mutually singular (this amounts to $\mu(\{u\}) = 0$ since δ_u is the Dirac measure at u) then $|\mu + t\delta_u|_{TV} = |\mu|_{TV} + |t|$ and the upper bound on the right-hand-side of (4.18) becomes $tV[\mu](u) + \frac{L}{2}t^2 + \lambda|t|$. In particular, this is the case (with probability 1) if $\mu = \sum_{i=1}^m c_j \delta_{u_j}$ where u_j and u are sampled independently and uniformly on the sphere. One difference with the finite-dimensional case is that the total variation norm is not "separable", meaning that unless $\mu \in L^1(\omega_d)$, there is no f_μ such that $|\mu|_{TV} = \int f_\mu(u) \mathrm{d}\omega_d(u)$ in general.

The proximal coordinate descent algorithm in this setting is given by: $\mu_0 = c_0 \delta_{u_0}$ and for each iteration k, we sample $u_{k+1} \in \mathbb{S}^{d-1}$ uniformly (*i.e.*, *w.r.t.* ω_d) and set

$$\mu_{k+1} = \mu_k + T_{u_{k+1}}(\mu_k)\delta_{u_{k+1}}$$
$$T_u(\mu) := \operatorname*{argmin}_{t \in \mathbb{R}} \left\{ -\frac{\hat{\mathcal{D}}_u(\mu, t)}{2L} := tV[\mu](u) + \frac{L}{2}t^2 + \lambda|\mu + t\delta_u|_{TV} - \lambda|\mu|_{TV} \right\}.$$

Note that because $\mu_k = \sum_{i=0}^k c_i \delta_{u_i}$ is atomic with u_i sampled uniformly in $\mathbb{S}^{d-1}$, with probability 1 over the sampling of $u_{k+1} \in \mathbb{S}^{d-1}$, μ_k and $\delta_{u_{k+1}}$ are mutually singular, which means $\hat{\mathcal{D}}_{u_{k+1}}(\mu_k, t) = tV[\mu_k](u_{k+1}) + \frac{L}{2}t^2 + \lambda|t|$ with probability 1. The minimization of the latter quantity *w.r.t.* t is akin to the minimization involved in a proximal algorithm with an L^1 penalty and Equation (4.11) gives

$$T_{u_{k+1}}(\mu_k) = \left(-\frac{V[\mu_k](u_{k+1})}{L} \right) \max\left(0, 1 - \frac{\lambda}{|V[\mu_k](u_{k+1})|} \right).$$

Sparsifying effect of the total variation penalty. We already observe the sparsifying effect of the total variation penalty: as soon as $|V[\mu_k](u_{k+1})| \leq \lambda$, it holds $T_{u_{k+1}}(\mu_k) = 0$, which entails $\mu_{k+1} = \mu_k$ and the number of atoms stays constant from iteration k to iteration $k + 1$. This opens the door for a sub-linear growth of the number of atoms with the iteration k which is an improvement on pure coordinate descent on the smooth part J only.

Global convergence guarantee is lost. In contrast to what occurs in the finite-dimensional case, the addition of the L^1 penalty comes with the drawback that we cannot provide a global convergence guarantee as we did for pure coordinate descent on J. However, the proximal coordinate descent algorithm still provides a true descent step: it is easily checked that

$$-\frac{1}{2L}\hat{\mathcal{D}}_{u_{k+1}}(\mu_k, T_{u_{k+1}}(\mu_k)) = -\frac{1}{2L}\max\left(0, |V[\mu_k](u_{k+1})| - \lambda\right)^2. \tag{4.19}$$

See Appendix C.3 for more details. It thus holds:

$$F(\mu_{k+1}) - F(\mu_k) \leq -\frac{1}{2L}\max\left(0, |V[\mu_k](u_{k+1})| - \lambda\right)^2,$$

but the upper bound on the right-hand-side can be 0 if $|V[\mu_k](u_{k+1})| \leq \lambda$. In expectation, conditionally on μ_k, this gives:

$$\mathbb{E}[F(\mu_{k+1})|\mu_k] \leq F(\mu_k) - \frac{1}{2L} \int_u \max\left(0, |V[\mu_k](u)| - \lambda\right)^2 \mathrm{d}\omega_d(u).$$

Remark. We observe that there is a trade-off between descent and sparsity: a large λ will encourage sparsity as it is more likely that $\mu_{k+1} = \mu_k$ but then there is no change in objective function. Conversely, a small λ will ensure that $F(\mu_{k+1}) \leq F(\mu_k)$ (or at least in expectation), but this will require adding a new atom. In this setting, we gain sparsity at a given iteration only if there is no change in the measure μ_k, which shows that we cannot have global convergence and a sparse measure at the same time: the number of atoms has to grow indefinitely if we wish to decrease the objective, albeit at a sub-linear rate.

4.4.3 . Sampling from existing atoms: a modified proximal algorithm

An alternative to enforce a higher level of sparsity is to sample from existing atoms half of the time. This leads to the following proximal algorithm: $\mu_0 = c_0 \delta_{u_0}$, and for each k

$$\begin{aligned}
\mu_{2k+1} &= \mu_{2k} + T_{u_{2k+1}}(\mu_{2k})\delta_{u_{2k+1}}, \qquad u_{2k+1} \sim \mathcal{U}(\mathbb{S}^{d-1}), \\
\mu_{2k+2} &= \mu_{2k+1} + T_{u_i(2k+1)}(\mu_{2k+1})\delta_{u_i(2k+1)}, \qquad i \sim \mathcal{U}(\{1, 2, \dots, m_{2k+1}\}),
\end{aligned}$$

where $\mathcal{U}$ denotes the uniform distribution and m_k is the size of the support of μ_k, *i.e.*, the number of atoms at iteration k. Since the number of neurons does not grow linearly and the weight assigned to a give neuron can change during later iterations, we now denote the iterates μ_k by $\mu_k = \sum_{i=1}^{m_k} c_i(k)\delta_{u_i(k)}$. The algorithm amounts to sampling a new atom $u_{2k+1} \in \mathbb{S}^{d-1}$ for odd iterations $2k+1$ and an existing neuron from $\{u_1(2k+1), \dots, u_{m_{2k+1}}(2k+1)\}$ for even iterations $2(k+1)$.

Remark. This modified algorithm is equivalent to sampling at each iteration, an atom from the distribution $\frac{1}{2}\omega_d + \frac{1}{2}\bar{\mu}_k$ where $\bar{\mu}_k = \mathcal{U}(\{u_1(k), \dots, u_{m_k}(k)\})$ (at least in expectation).

Descent property. By the definition of $T_u(\mu)$ and the descent property (4.16) both steps (sampling uniformly over the sphere or over existing atoms) are true descent steps which guarantees $F(\mu_{k+1}) \leq F(\mu_k) \leq F(\mu_0)$.

Boundedness of the total variation of the iterates. Consider $\mu = \sum_{i=1}^{m} c_i \delta_{u_i}$ and assume that $F(\mu) \leq F(\mu_0)$. Denoting by J^* the minimum of the smooth part J, it holds $J^* + \lambda|\mu|_{TV} \leq F(\mu) \leq F(\mu_0)$ and thus $||c|| \leq ||c||_1 = |\mu|_{TV} \leq (F(\mu_0) - J^*)/\lambda$, which proves that the euclidean norm of $c =$

$(c_1, \dots, c_m) \in \mathbb{R}^m$ is bounded by a fixed constant as soon as $F(\sum_{i=1}^m c_i \delta_{u_i}) \leq F(\mu_0)$. This in particular applies to the iterates μ_k so that $||c(k)|| \leq B := (F(\mu_0) - J^*)/\lambda$ for any k, where $c(k) = (c_1(k), \dots, c_{m_k}(k)) \in \mathbb{R}^{m_k}$.

Correspondence with a finite-dimensional objective. Consider fixed atoms $u_1, \dots, u_m \in \mathbb{R}^m$ and the objective $F_m : c \in \mathbb{R}^m \mapsto F(\sum_{i=1}^m c_i \delta_{u_i}) = J_m(c) + \lambda ||c||_1$ with $J_m(c) = J(\sum_{i=1}^m c_i \delta_{u_i})$. It is easily checked that $\nabla_i J_m(c) = V[\mu(c)](u_i)$ where $\mu(c) = \sum_{i=1}^m c_i \delta_{u_i}$. It then follows from the smoothness of J that the gradient is Lipschitz: $||\nabla J_m(c_1) - \nabla J_m(c_2)|| \leq \sqrt{m} L ||c_1 - c_2||_1 \leq mL||c_1 - c_2||$. Doing one step of the proximal coordinate descent algorithm on J_m from a given iterate $c \in \mathbb{R}^m$ thus amounts to sampling an atom u_i from $\{u_1, \dots, u_m\}$ and setting the $c' = c + T_i(c) e_i$. By the results of Section 4.3.4 it holds in expectation conditionally on c:

$$\mathbb{E}[F(\mu(c'))|c] \leq F(\mu(c)) - \frac{1}{2mL} \mathcal{D}(c),$$
$$\frac{-1}{2mL} \mathcal{D}(c) = \min_{\gamma \in \mathbb{R}^m} \sum_{i=1}^m \gamma_i V[\mu(c)](u_i) + \frac{m^2 L}{2} ||\gamma||^2 + \lambda ||\gamma + c||_1 - \lambda ||c||_1.$$

Note that the above minimum is also the minimum over $\gamma - c$ when $\gamma \in \mathbb{R}^m$.

Sampling among existing atoms. For odd iterations $2k+1$, the step is the same as for the proximal coordinate descent algorithm presented in Section 4.4.2. On the other hand, for even iterations $2k+2$, $T_{u_i(k)}(\mu_{2k+1})$ is obtained as:

$$T_{u_i(k)}(\mu_{2k+1}) = \operatorname*{argmin}_{t \in \mathbb{R}} t V[\mu_{2k+1}](u_i(2k+1) + \frac{L}{2} t^2 + \lambda |c_i(2k+1) + t|.$$

Calling $V_{i,2k+1} := V[\mu_{2k+1}](u_i(2k+1))$, the exact formula for $T_{u_i(k)}(\mu_{2k+1})$ is given by Equation (4.11):

$$T_{u_i(k)}(\mu_{2k+1}) = -c_i(2k+1) + \left(c_i(2k+1) - \frac{V_{i,2k+1}}{L} \right) \max \left(0, 1 - \frac{\lambda}{|V_{i,2k+1} - L c_i(2k+1)|} \right).$$

In this case, the total mass for the selected atom $i \in [1, m_{2k+1}]$ is $c_i(2k+2) = c_i(2k+1) + T_{u_i(2k+1)}(\mu_{2k+1})$ and is thus zero if and only if $|V_{i,2k+1} - L c_i(2k+1)| \leq \lambda$. As described in the previous paragraph, this step is basically equivalent to a step of the proximal algorithm in dimension m_{2k+1} as presented in Section 4.3.4 with a smoothness constant equal to $m_k L$. We thus have in expectation conditionally on μ_{2k+1},

$$\mathbb{E}[F(\mu_{2k+2})|\mu_{2k+1}] \leq F(\mu_{2k+1}) - \frac{1}{2L m_{2k+1}^2} \mathcal{D}(c(2k+1))$$

where $c(2k+1) = (c_1(2k+1), \dots, c_{m_{2k+1}}(2k+1))$ and

$$-\frac{\mathcal{D}(c(2k+1))}{2m_{2k+1}L} = \min_{c \in \mathbb{R}^{m_{2k+1}}} \Big(\sum_{i=1}^{m_{2k+1}} c_i V[\mu_{2k+1}](u_i(2k+1)) + \frac{m_{2k+1}L}{2}||c||^2 + \lambda||c + c(2k+1)||_1 - \lambda||c(2k+1)||_1 \Big).$$

Calling $F^*_{2k+1} = \min_{c \in \mathbb{R}^{m_{2k+1}}} F(\sum_{i=1}^{m_{2k+1}} c_i \delta_{u_i(2k+1)})$, and $c^*(2k+1)$ a minimizer of the latter objective, since $F(\sum_{i=1}^{m_{2k+1}} c_i^*(2k+1)\delta_{u_i(2k+1)}) \leq F(\mu_{2k+1}) \leq F(\mu_0)$, by the paragraph on the boundedness of the iterates above, it holds $||c(2k+1)|| \leq B$ and $||c^*(2k+1)|| \leq B$ which ensures $||c(2k+1) - c^*(2k+1)|| \leq R := 2B$. Therefore, from the analysis in Lemma 4.3.2 it holds:

$$\frac{1}{2}\mathcal{D}(c(2k+1)) \geq \frac{1}{2} \min\left(\frac{Lm_{2k+1}}{F(\mu_{2k+1}) - F^*_{2k+1}}, \frac{1}{||c(2k+1) - c^*(2k+1)||^2} \right) \times \left(F(\mu_{2k+1}) - F^*_{2k+1}\right)^2.$$

We now lower-bound the right-hand-side: we have $m_{2k+1} \geq 1$, and since $F^* \leq F^*_{2k+1} \leq F(\mu_{2k+1}) \leq F(\mu_0)$ it holds $0 \leq F(\mu_{2k+1}) - F^*_{2k+1} \leq F(\mu_0) - F^*$, which entails $\frac{Lm_{2k+1}}{F(\mu_{2k+1}) - F^*_{2k+1}} \geq \frac{L}{F(\mu_0) - F^*}$. For the second term in the minimum, it holds $||c(2k+1) - c^*(2k+1)||^2 \leq R^2$ and thus $\frac{1}{||c(2k+1) - c^*(2k+1)||^2} \geq \frac{1}{R^2}$. Therefore, calling $\tau := \frac{1}{2}\min\left(\frac{L}{F(\mu_0) - F^*}, \frac{1}{R^2}\right)$ which is a constant (not depending on k or m_{2k+1}), we have

$$\frac{1}{2}\mathcal{D}(c(2k+1)) \geq \tau \left(F(\mu_{2k+1}) - F^*_{2k+1}\right)^2, \tag{4.20}$$

from which it follows that:

$$\mathbb{E}[F(\mu_{2k+2}) - F^*_{2k+1} | \mu_{2k+1}] \leq (F(\mu_{2k+1}) - F^*_{2k+1}) - \frac{\tau}{Lm^2_{2k+1}} \left(F(\mu_{2k+1}) - F^*_{2k+1}\right)^2.$$

Global convergence is lost. Unfortunately, this is not enough to guarantee convergence to the global minimum F^*. One option would be to try proving a similar bound as in Inequality (4.20) with F^* instead of F^*_{2k+1}, but there is no reason for such an inequality to hold in general and the proof technique used in Lemma 4.3.2 does not apply here because μ^* has no reason to be of the form $\sum_{i=1}^{m_{2k+1}} c_i \delta_{u_i(2k+1)}$. A more favourable case is if we assume μ^* is sparse (an atomic measure with a finite number of atoms), in which case we could hope to prove global convergence if we can prove that $m_k \geq m^*$ for $k \geq k_0$ where m^* is the number of atoms of μ^*. However, it is not obvious that the latter property holds. Also note that the constant which guarantees sufficient decrease $\tau/(Lm^2_{2k+1})$ becomes smaller with the number of atoms which is an unfavourable behaviour.

Lack of control of the number of atoms m_k. One issue with the coordinate descent algorithm we have presented in this section is that it is difficult to control the growth of the number of atoms m_k. For odd iterations $2k+1$ we can have either $m_{2k+1} = m_{2k}$ or $m_{2k+1} = m_{2k} + 1$, and for even iterations we can have either $m_{2k+2} = m_{2k+1}$ or $m_{2k+2} = m_{2k+1} - 1$, and we have no control over the probability of each event.

We present the results of the numerical experiments on this proximal algorithm with a total variation penalty in Section 4.6 to validate empirically that this method is indeed able to limit the growth of the number of neurons m_k (and thus the computational cost) while still maintaining a good performance on the initial objective J.

4.5 . Kernel penalties

An alternative to the total variation penalty (which has an explicit sparsifying effect) is to penalize with smooth kernels which implicitly encourages sparsity by drawing atoms towards each other or pushing them away. This will not explicitly remove atoms but we can merge atoms together (that is, sum their masses up) if they are sufficiently close. This method shares some similarities with the expend-and-cluster strategy used by Martinelli et al. (2023): they consider training multiple deep networks with bounded width and then cluster the neurons together in order to remove inefficient neurons and reduce the size of the network.

We consider symmetric, non-negative dot-product kernels on the sphere of the form $K : (u, v) \in \mathbb{R}^d \times \mathbb{R}^d \mapsto K(u, v) = \kappa(\langle u, v\rangle)$ for some $\kappa : \mathbb{R} \to \mathbb{R}_+$. Since the atoms are on the sphere, this is equivalent to assuming that $K(u, v) = \phi(||u-v||^2)$ because $||u - v||^2 = 2(1 - \langle u, v\rangle)$. We say that the kernel is *attractive* if κ is an decreasing function, and that it is *repulsive* if κ is a increasing function. Typically, we consider $\kappa_{a,\sigma}(s) = 1 - e^{(s-1)/\sigma^2}$ for attractive kernels and $\kappa_{r,\sigma}(s) = e^{(s-1)/\sigma^2}$ for repulsive kernel where $\sigma > 0$ is a parameter controlling the range of interaction of different atoms on the sphere.

Penalized objective. In this setting, the objective we consider has the form:

$$F(\mu) = J(\mu) + \lambda \int_{u,v} K(u, v)\mathrm{d}|\mu|(u)\mathrm{d}|\mu|(v)$$

where J is a smooth term, typically a data-fitting term such as the empirical risk. For an atomic measure $\mu_m = \sum_{i=1}^m c_i \delta_{u_i}$, the kernel penalty is equal to $\int K\mathrm{d}|\mu_m|\mathrm{d}|\mu_m| = \sum_{i,j=1}^m |c_i||c_j|K(u_i, u_j)$, and can thus be interpreted as an interaction kernel between the particles (or atoms, or neurons) $(u_1, \dots, u_m) \in (\mathbb{S}^{d-1})^m$ which depends only on the absolute values of their weights $|c_i|$. The alternative version where we remove the absolute values and consider instead the penalty $\int K\mathrm{d}\mu\mathrm{d}\mu$ is also possible but we focus here on a penalty involving $|\mu|$.

Evolution equation. From an initial $\mu_0 \in \mathcal{M}(\mathbb{S}^{d-1})$, we consider the evolution given by the Wasserstein-Fisher-Rao (WFR) gradient flow as in Equation (1.9) with a minor adjustment to account for the absolute values:

$$\partial_t \mu_t^{\pm} = -\mathrm{div}(\pm \tilde{v}_t^{\pm} \mu_t^{\pm}) \pm 2 g_t^{\pm} \mu_t^{\pm} \tag{4.21}$$

which is to be understood in the sense of distributions, where $\mu_t^+ \in \mathcal{M}_+(\mathbb{S}^{d-1})$ (*resp.* $\mu_t^- \in \mathcal{M}_+(\mathbb{S}^{d-1})$) is the positive part (*resp.* negative part) of $\mu_t \in \mathcal{M}(\mathbb{S}^{d-1})$. The advection term $\tilde{v}_t^{\pm}$ and the reaction term $g_t^{\pm}$ are given by:

$$\begin{aligned} g_t^{\pm}(u) &= -\left(\pm V[\mu_t](u) + \lambda \int K(u,v) \mathrm{d}|\mu_t|(v) \right), \\ \tilde{v}_t^{\pm}(u) &= \mathrm{proj}_{\{u\}^\perp} \left(\nabla g_t^{\pm}(u) \right). \end{aligned}$$

where V is the first variation of J. Note that when $K(u,v)$ has the form $\kappa(\langle u, v \rangle)$, its gradient simplifies to $\nabla_u K(u,v) = \kappa'(\langle u, v \rangle) v$ and thus $\mathrm{proj}_{\{u\}^\perp} (\nabla_u K(u,v)) = \kappa'(\langle u, v \rangle) (v - \langle u, v \rangle u)$. Note in particular that $\mathrm{proj}_{\{u\}^\perp} (\nabla_u K(u,u)) = 0$ for $u \in \mathbb{S}^{d-1}$. As discussed in Section 1.2.4, in the context of infinitely-wide two-layer networks with a positively-homogeneous activation function and initialized on the cone $\{(a,b) \in \mathbb{R} \times \mathbb{R}^d : |a| = ||b||\}$, (4.21) is an equivalent formulation to the Wasserstein gradient flow in the space $\mathcal{P}_2(\mathbb{R}^{d+1})$ of probability measures with finite second moment, for the objective

$$\tilde{F}(\mu) = J(\mu) + \lambda \int |a_1| \, |a_2| \, ||b_1|| \, ||b_2|| \, K\left(\frac{b_1}{||b_1||}, \frac{b_2}{||b_2||} \right) \mathrm{d}\mu(a_1, b_1) \mathrm{d}\mu(a_2, b_2)$$

for $\mu \in \mathcal{P}_2(\mathbb{R}^{d+1})$.

Evolution for atomic measures. If $\mu_{m,0} = \sum_{i=1}^m c_j(0) \delta_{u_j(0)}$, as discussed in Section 1.2.4, the WFR gradient flow (4.21) $\mu_{m,t}$ starting from $\mu_{m,0}$ is also atomic with m atoms: $\mu_{m,t} = \sum_{i=1}^m c_i(t) \delta_{u_i(t)}$, and the dynamics translate into evolution equations on the (signed) masses (or weights) $c_i(t)$ and on the position $u_i(t) \in \mathbb{S}^{d-1}$, given by:

$$\begin{aligned} \frac{d}{dt} c_i(t) &= 2 g_t^{\epsilon_i}(u_i(t)) c_i(t), \\ \frac{d}{dt} u_i(t) &= \tilde{v}_t^{\epsilon_i}(u_i(t)), \end{aligned}$$

with $\epsilon_i := \mathrm{sign}(c_i(0)) \in \{-1, +1\}$.

4.5.1 . An example of attraction/repulsion with two particles

Here, we consider the most simple setting where there are only two particles $u_1(t), u_2(t)$ interacting. We show that for attractive kernels, *i.e.*, κ decreasing, the particles end up merging and for repulsive kernels, *i.e.*, κ increasing, the two particles end up at opposite ends of the sphere.

We consider only the interaction term in the objective corresponding to the kernel penalty, that is we assume $J = 0$, and set $\lambda = 1$ for simplicity. In this context, the reaction and advection terms simplify to $g_t^{\pm}(u) = -\int K(u,v)\mathrm{d}|\mu_t|(v)$ and $\tilde{v}_t^{\pm}(u) = -\int \kappa'(\langle u,v\rangle)\,(v - \langle u,v\rangle u)\,\mathrm{d}|\mu_t|(v)$, and the dynamics of the two particles are given by:

$$\begin{aligned}
c_1'(t) &= -2\Big(|c_1(t)|\kappa(1) + |c_2(t)|\kappa(\varphi(t))\Big)c_1(t)\\
c_2'(t) &= -2\Big(|c_2(t)|\kappa(1) + |c_1(t)|\kappa(\varphi(t))\Big)c_2(t)\\
u_1'(t) &= -|c_2(t)|\kappa'(\varphi(t))\Big(u_2(t) - \varphi(t)u_1(t))\Big)\\
u_2'(t) &= -|c_1(t)|\kappa'(\varphi(t))\Big(u_1(t) - \varphi(t)u_2(t))\Big)
\end{aligned}$$

where $\varphi(t) := \langle u_1(t), u_2(t)\rangle$. The weights $c_1(t)$ and $c_2(t)$ keep the same sign all along the dynamics. Indeed, if $c_i(t_0) = 0$ for some t_0 then $c_i(t) = 0$ for all t since : $t \mapsto 0$ and c_i would then solve the same ODE and both take the value 0 at t_0, which is absurd since we initialize with $|c_i(0)| \neq 0$. Calling $\epsilon_i = \mathrm{sign}(c_i(0))$, it holds:

$$\begin{aligned}
c_i'(t) &= -G_i(t)\epsilon_i|c_i(t)|\\
\epsilon_i c_i'(t) &= -G_i(t)|c_i(t)|\\
(\epsilon_i c_i)'(t) &= -G_i(t)|c_i(t)|\\
|c_i|'(t) &= -G_i(t)|c_i(t)| \leq 0
\end{aligned}$$

with $G_i(t) \geq 0$ since $\kappa \geq 0$, which shows that $|c_i|$ is decreasing. This makes sense considering that because K is non-negative, the kernel penalty term is minimized when there is no mass, *i.e.*, $\mu = 0$. However, to understand the effect of the kernel penalization on the dynamics of the positions of the particles we can assume the masses fixed for simplicity or at least that they are lower-bounded in absolute value: $|c_i(t)| \geq C > 0$. In any case, the ODE satisfied by the inner product φ is easily derived as:

$$\varphi'(t) = -(|c_1(t)| + |c_2(t)|)\kappa'(\varphi(t))\Big(1 - \varphi(t)^2\Big).$$

We make one additional assumption on the kernel: $0 < A \leq |\kappa'|$ on $[-1,1]$. This is satisfied by the kernels $\kappa_{a,\sigma}(s) = 1 - e^{s/\sigma^2}$ and $\kappa_{r,\sigma}(s) = e^{s/\sigma^2}$.

Attractive case. In the case of an attractive kernel, $\kappa' \leq 0$ which implies that $\varphi'(t) \geq 0$: φ is an increasing function. With the assumptions on $|\kappa'|$ and $|c_i(t)|$, this leads to $\varphi'(t) \geq 2CA(1 - \varphi(t))(1 + \varphi(t)) \geq C_0(1 - \varphi(t))$ with $C_0 := 2CA(1 + \varphi(0))$, so that

$$\frac{d}{dt}\left(1 - \varphi(t)\right) \leq -C_0\left(1 - \varphi(t)\right)$$

which implies $1-\varphi(t) \leq (1-\varphi(0))\exp(-C_0 t)$ by Gronwall's lemma, and thus

$$\varphi(t) \geq 1 - (1-\varphi(0))\exp(-C_0 t)\,.$$

Since $C_0 > 0$, this proves that $\varphi(t) \to 1$ as $t \to \infty$, from which it follows that $||u_1(t) - u_2(t)||^2 \to 0$.

Repulsive case. For a repulsive kernel, $\kappa' \geq 0$ which means that φ is decreasing. Similar arguments to the attractive case lead to $\varphi'(t) \leq -2CA(1-\varphi(0))(1+\varphi(t))$, and with the same reasoning as in the previous paragraph we get

$$\varphi(t) \leq -1 + (1+\varphi(0))\exp(-C_0 t))$$

with $C_0 > 0$, which implies that $\langle u_1(t), u_2(t)\rangle \to -1$ as $t \to \infty$ showing that for large t the particles $u_1(t)$ and $u_2(t)$ end up at opposite sides of the sphere.

4.5.2 . A coordinate descent algorithm

A practical algorithm to minimize J while keeping a reasonable number of particles is alternate between coordinate descent steps on the smooth part J and Wasserstein-Fisher-Rao steps on the penalized objective $F(\mu) = J(\mu) + \lambda \int K \mathrm{d}|\mu|\mathrm{d}|\mu|$. With λ small enough, the coordinate descent step should decrease the value of J, while the Wasserstein-Fisher-Rao steps should encourage the merging of particles if λ is large enough (even if particles are globally pushed away from one another, some particles might be pushed towards the same direction). The Wasserstein-Fisher-Rao step corresponds to the discretization of the WFR gradient flow (4.21) as follows: given an atomic measure $\mu_k = \sum_{i=1}^{m_k} c_i(k)\delta_{u_i(k)}$ at iteration k, the WFR step computes the measure $\mu_{k+1} = \sum_{i=1}^{m_k} c_i(k+1)\delta_{u_i(k+1)}$ with

$$\begin{aligned}
c_i(k+1) =& c_i(k)\left(1 - 2\eta\epsilon_i V[\mu_k](u_i(k)) - 2\eta\sum_{j=1}^{m_k}|c_j(k)|K(u_i(k),u_j(k))\right),\\
u_i(k+1) =& \mathsf{proj}_{\mathbb{S}^{d-1}}\Big(u_i(k) - \eta\epsilon_i\nabla V[\mu_k](u_i(k)) - \\
&\qquad \eta\sum_{j=1}^{m_k}|c_j(k)|\mathsf{proj}_{\{u_i(k)\}^\perp}(\nabla K(u_i(k),u_j(k)))\Big)
\end{aligned}$$

where $\eta > 0$ is a small step-size. Unfortunately, we cannot provide any theoretical guarantees with this method, either on the global convergence or on the control of the number of atoms. Although this approach is theoretically motivated, the numerical experiments were inconclusive at this stage on the effective ability of such methods to limit the growth of the number of neurons: it does not appear clearly that the kernel penalties allow to induce more sparsity than doing coordinate descent mixed with "conic" steps in the Wasserstein geometry.

4.6 . Numerical experiments

We present in this section results for the numerical experiments with the proximal algorithm for the total variation penalty. The code for the numerical experiments is available at https://github.com/karl-hajjar/pop-conic.

4.6.1 . Proximal algorithm for the total variation penalty

We consider in this section different practical variants of the proximal algorithms presented in Section 4.4.2 as well as the reference coordinate descent algorithm on the smooth objective without penalization in the $L^2(\omega_d)$ geometry. We detail the name and description of each method below:

- $\boldsymbol{L^2(\omega_d)}$ (purple): pure coordinate descent on the smooth objective only (no penalty). This is the algorithm introduced in Section 4.4.1

- $\boldsymbol{L^2(\omega_d)}$-**prox** (blue): proximal coordinate descent on the objective penalized with a total variation penalty. This is the algorithm introduced in Section 4.4.2.

- $\boldsymbol{L^2(\mu_m)}$-**prox** (orange): proximal coordinate descent on a measure with a finite number m of atoms. This is akin to the coordinate proximal algorithm in finite dimension: we sample m atoms at initialization and at each iteration we sample one of the atoms and set its weight by using the proximal update rule for the penalized objective. This corresponds to repeating, in the algorithm described in Section 4.4.3, the even step $(2k+2)$ where we sample from existing neurons.

- **mix** (green): proximal coordinate descent on the penalized objective where we alternate between sampling a new atom on the sphere and one of the existing atoms. This is the algorithm introduced in Section 4.4.3.

- **mix-conic** (red): proximal coordinate descent on the penalized objective where we alternate between three different kinds of steps for each iteration: (i) sampling a new atom on the sphere and doing a proximal step, (ii) sampling one of the existing atoms and doing a proximal step, and (iii) doing a "conic" descent step in the Wasserstein geometry as one would in the discretization of the Wasserstein GF (that is, regular gradient descent on all the neurons). This is similar to the algorithm introduced in Section 4.4.3 with an added Wasserstein "conic" descent step every three iterations.

- **GD-conic** (brown): pure conic descent in Wasserstein geometry on a measure initialized with m atoms. This is equivalent to doing gradient descent on a two-layer neural network with m neurons initialized on the cone (see Section 1.2.4 for more details) and a ReLU activation.

We train all algorithms for a total of $3,000$ optimization steps. The smooth objective J is the empirical risk on some toy data generated by a teacher two-layer network with $m^* = 50$ neurons and input data are sampled uniformly on the sphere. The input dimension is $d = 10$ and we use $m = 500$ neurons for the algorithms $\boldsymbol{L^2(\mu_m)}$-**prox** and **conic-GD**.

We show in Figure 4.1 the performance of the different variants of the proximal algorithm on the penalized objective. They all seem to perform relatively comparably except the $\boldsymbol{L^2(\omega_d)}$-**prox** variant which appears to stagnate somewhat while the other algorithms continuously decrease the objective, albeit quite slowly for large iterations.

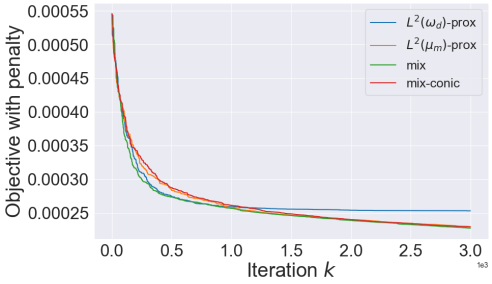


Figure 4.1: Penalized objective $F(\mu_k)$ *vs.* iteration k

Next we compare the performance of all the algorithms mentioned above on the objective without penalization (which only $\boldsymbol{L^2(\omega_d)}$ and **conic-GD** optimize) to see how the penalized versions fare on the original objective compared to algorithms without any penalization. Figure 4.2 shows that the penalized variants perform slightly worse than the ones without penalization, which is no surprise. However, the gap in performance does not appear too large. It is also interesting to note that the **conic-GD** algorithm, while a little slower than pure coordinate descent $\boldsymbol{L^2(\omega_d)}$, manages to reach the same final performance.

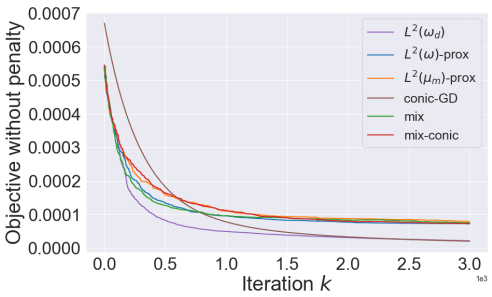
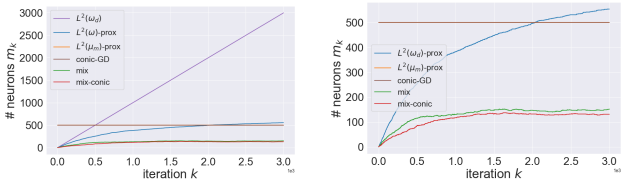


Figure 4.2: Original objective $J(\mu_k)$ *vs.* iteration k

Finally, we compare the computational cost incurred by the different algorithms. We analyze two quantities: first the evolution of the number of neurons (or atoms)

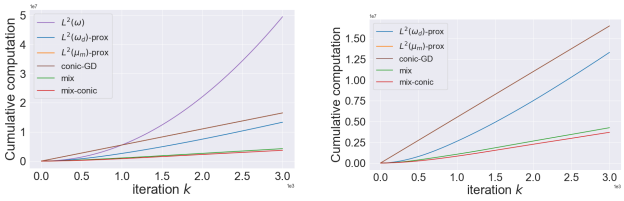
m_k during the optimization procedure in Figure 4.3, and second the computational complexity as measured by the number of operations needed for the forward pass on a single sample, which is $O(m_k(d+1))$, in Figure 4.4. For both the number of neurons and the computational complexity of the forward pass, we first present the results for all algorithms (a), and then for all algorithms except the pure coordinate descent $\boldsymbol{L^2(\omega_d)}$ (b) as the number of neurons grows linearly for the latter.



(a) All algorithms

(b) Removing pure coordinate descent algorithm

Figure 4.3: Computational complexity of the forward pass accumulated over iterations



(a) All algorithms

(b) Removing pure coordinate descent algorithm

Figure 4.4: Computational complexity of the forward pass accumulated over iterations

As expected, all the proximal variants are able to limit the number of neurons and thus the computational complexity of the method. While the number of neurons seems to keep increasing for $\boldsymbol{L^2(\omega_d)}$-**prox** (but at a sub-linear rate), using algorithms which sample from existing neurons (**mix** and **mix-conic**) allows to reduce much more strongly the number of neurons. The latter appears to stagnate after some time for **mix** and **mix-conic**. Additionally, the reduced number of neurons does not seem to hurt the performance too much as shown in Figure 4.1. This shows that even though those methods do not offer any explicit control over m_k or any global convergence guarantee, they are effectively able to prevent a huge computational complexity while still performing well on the initial objective J. However, it is noteworthy that all methods, even the ones adapt dynamically the number of neurons have many more neurons then the target network which has $m^* = 50$ neurons.

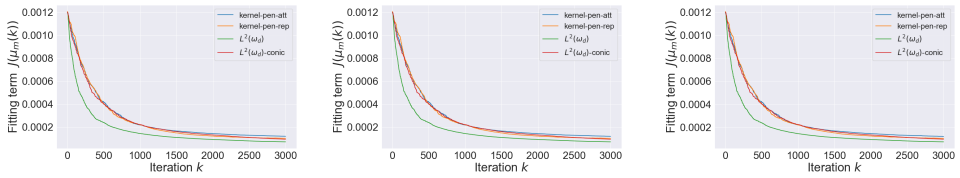
4.6.2 . Kernel penalization

In this section, we present the results from our first numerical experiments with kernel penalties. We stress that the experiments are by no means exhaustive and only more thorough experimentation—which we leave for future work—would allow to determine the practical value of such methods.

In the practical implementation of the algorithm described in Section 4.5.2 we alternate between two kinds of steps: (i) coordinate descent steps where we sample a new neuron on the sphere and set its weight using the coordinate descent minimization, and (ii) "conic" descent steps in the Wasserstein geometry which is akin to doing gradient descent on the positions and weights of the atoms of the current iterate μ_k. To effectively obtain sparsity, we merge together neurons which are closer than a certain threshold ϵ after the conic descent step. This parameter is key to controlling the number of neurons during the iterations of the algorithm.

We compare the algorithm with both an attractive kernel $\kappa_a(s) = 1 - e^{(s-1)/\sigma^2}$ (**kernel-pen-att**) and a repulsive kernel $\kappa_a(s) = e^{(s-1)/\sigma^2}$ (**kernel-pen-rep**) to a pure coordinate descent method ($\boldsymbol{L^2(\omega_d)}$, sampling a new neuron at each step as described in Section 4.4.1) as well as a variant where every other step we do conic descent in Wasserstein geometry as described above ($\boldsymbol{L^2(\omega_d)}$**-conic**). The number of neurons grows linearly for the latter two algorithms a priori. To make the comparison fair with the kernel penalization methods we also use the ad-hoc rule for merging neurons in the practical implementation of these algorithms.

The objective is the (penalized) empirical risk on data generated by a teacher network with $m^* = 50$ neurons and input data sampled uniformly on the unit sphere in dimension $d = 5$. We use $\sigma = 0.2$, a learning rate of $\eta = 0.01$ for the conic steps, and $\lambda = 0.005$ for attractive kernels and $\lambda = 2.0$ for repulsive kernels. We also use three different values for ϵ: $\{0.1, 0.15, 0.25\}$. Those values induce different levels of sparsity but do not seem to affect performance too much as depicted in Figure 4.5.

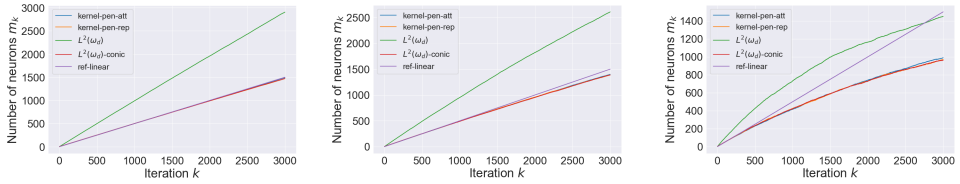


(a) $\epsilon = 0.1$ (b) $\epsilon = 0.15$ (c) $\epsilon = 0.25$

Figure 4.5: Initial objective $J(\mu_k)$ *vs.* k.

Figure 4.6 depicts the evolution of the number of neurons m_k versus the iteration number k. Unsurprisingly, the algorithms become more sparse as ϵ increases. We have also shown for reference the theoretical growth of an algorithm which would add a new neuron every two steps (which is the case of **kernel-pen-att**, **kernel-pen-rep** and $\boldsymbol{L^2(\omega_d)}$**-conic** if ϵ is too small). The results do not seem to

indicate that the kernel penalty is able to induce more sparsity than simply doing conic steps in the Wasserstein geometry with the ad-hoc rule we adopt. They do indicate however that conic descent seems to push particles closer to one another compared to sampling them uniformly on the sphere (which is what happens in pure coordinate descent). With $\epsilon = 0.1$, none of the methods appear to reduce the number of neurons. With $\epsilon = 0.15$ the kernel penalization methods as well as $\boldsymbol{L^2(\omega_d)}$**-conic** seem to induce some sparsity but the number of neurons m_k still seems to grow linearly with k albeit at a rate αk with $\alpha < 1/2$. Finally, with $\epsilon = 0.25$ the kernel penalization methods as well as $\boldsymbol{L^2(\omega_d)}$**-conic** seem to induce a sub-linear growth of m_k at approximately the same rate, without damaging performance too much. We have repeated the experiments with multiple values



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.15$ (c) $\epsilon = 0.25$

Figure 4.6: Number of neurons m_k *vs.* k.

of the penalty coefficient λ and of the standard deviation σ but the results seem to be similar. However we have not carried out an extensive parameter search and there could very well be some appropriate choice of η, λ, σ and ϵ which induce more sparsity for the kernel penalties than for the other methods—and we do not rule out this eventuality—but the experiments we have carried out do no allow to conclude to such a fact.

4.7 . Discussion

Taking inspiration from convex analysis in finite dimension, we have explored a range of algorithms to minimize convex objectives over the space of signed measures. We have first shown that a coordinate descent algorithm in the L^2 geometry provides a global convergence rate of $k^{-1/d}$, but with the drawback that the number of atoms grows linearly with the number of iterations. To circumvent this issue we have studied penalized version of the objective which encourage sparsity. The first is a total variation penalty for which we present different variants of a proximal algorithm. While we could not obtain theoretical guarantees for these methods in terms of the control of the number of atoms or of the objective value, we show empirically that they have the desired behaviour. We leave as an open problem the theoretical proof that this algorithm has the appropriate behaviour. We have also studied kernel penalties which encourage sparsity implicitly by pushing

atoms closer or further away from each other, and we leave for future work the experimental validation of such methods which remains inconclusive at this stage.

Conclusion

Studying the limit of neural networks where the width tends to infinity has brought many insights on their behaviour, in particular on the dynamics induced by gradient descent from a random initialization. For i.i.d. initializations, this limit allows to understand precisely how one should scale the variance at initialization as well as the learning rates during training in order to induce favorable learning properties. Different parameterizations (corresponding to different choices of variance and learning rate) lead to different types of behaviour in the large-width limit. The neural tangent kernel has global optimality properties, but is akin to a *linear* kernel method in an infinite-dimensional space. On the other hand, "mean-field" limits of two-layer networks enjoy favorable statistical properties while still providing global optimality guarantees. However, such "mean-field" limits with three layers or more—which we call integrable parameterizations—have a degenerate behaviour with i.i.d. initializations. Recently, a parameterization called μP has been proposed so as to "maximize" learning in the infinite-width limit, but it is unclear how it relates to other parameterizations.

In the first part of this thesis, we have shown that the nature of the degeneracy of integrable parameterizations can be precisely understood using the tools of the Tensor Program, and that there is a solution to train deep networks in that parameterization in the infinite-width limit. By using large learning rates for the first weight updates, one can correct the scales of the gradients in order to amplify the random fluctuations at initialization and initiate a non-trivial learning phase. Furthermore, this results in a dynamic which is close to that of μP: the only difference is that the contribution of the initial random weights disappears after the first weight update.

In the second part, we have demonstrated that the dynamics of infinitely wide two-layer networks adapt to the orthogonal symmetries of the target function when the input data has spherical symmetry. When the target function is odd, we have proved that the dynamics lead to an exponential convergence to the global minimum of the training objective. When the target function depends only on the orthogonal projection onto an unknown lower-dimensional sub-space, we have showed that the Wasserstein gradient flow dynamics reduce to lower dimensional PDEs. Although rigorously proving the convergence of the features to the low-dimensional sub-space is still an open problem, we gave informal and numerical arguments which indicate that this convergence occurs.

In the third part, we have explored different optimization algorithms on the space of measures inspired by techniques in convex optimization in finite dimension. For smooth objectives over the space of measures, we have designed a coordinate descent method which convergences to the global minimum at a rate of $k^{-1/d}$ where k is the iteration number and d the dimension. The latter algo-

rithm is impractical because it implies adding a new neuron at every step, making it computationally costly to run the algorithm until convergence. We have thus, in a second step, proposed penalized versions of the objective which encourage sparsity and strike a balance between global convergence and computational cost. The first penalty we considered is the non-smooth total variation penalty, akin to an L^1-penalization in finite dimension, for which we proposed various proximal algorithms to minimize the corresponding penalized objective. The second type of penalties we considered consists in kernel penalties which induce sparsity implicitly by pushing neurons towards or away from each other. Although we could not obtain theoretical guarantees for these penalized methods, we show experimentally that they have a good behaviour: decreasing the objective while limiting the number of neurons and thus the computational cost of the method.

APPENDIX

A - Appendix for Chapter 2

A.1 . Notations for the appendix

We introduce here some additional notations that will come in handy in the text and equations presented in the Appendix.

Hat matrices. We define the following matrices and output weight vector (see Definition 2.2.1 for the definitions of the matrices U^l):

$$\begin{cases} \widehat{W}^1 = U^1 \\ \widehat{W}^l = m^{-1/2} U^l, \quad l \in [2, L+1]. \end{cases} \tag{A.1}$$

The pre-factor in $m^{-1/2}$ is the natural re-scaling of the i.i.d. Gaussian matrices when their input dimension grows to infinity due to the central limit theorem (CLT).

Omegas. For any ac-parameterization, we define $\omega_1 := m^{-a_1}$, and for any $l \in [2, L+1]$, $\omega_l := m^{1/2-a_l}$. To avoid blow-up or vanishing in the first layer, all the parameterizations we study have $\omega_1 = 1$. This is the case for integrable parameterizations, the NTK parameterization and for μP. For integrable parameterizations we also have $\omega_l = m^{-1/2}$ for $l \in [2, L+1]$, but for μP, $\omega_l = 1$ if $l \in [2, L]$ and $\omega_{L+1} = m^{-1/2}$ (see Section A.2.3 for a detailed description of μP).

Those ω_l naturally appear in the calculations as the magnitudes of the first forward pass of an ac-parameterization of a neural network. The term m^{-a_l} comes from the scaling pre-factor of the effective weights, and the added $m^{1/2}$ appears when expressing the computation in function of the naturally scaled $\widehat{W}^l$: $W^l(0) = \omega_l \widehat{W}^l$.

Scalar limits. For any scalar ω which depends on m, we denote by $\mathring{\omega}$ the almost sure limit (when it exists) of this scalar as $m \to \infty$.

Gradients. We define for any t and l,

$$\begin{cases} dh_t^l := \nabla_{h_t^l} f_t(\xi_t) \\ dx_t^l := \nabla_{x_t^l} f_t(\xi_t) \\ dw^l(t) := \nabla_{w^l(t)} f_t(\xi_t) \\ db^l(t) := \nabla_{b^l(t)} f_t(\xi_t) \\ \chi_t := \partial_2 \ell(y_t, f_t(\xi_t)). \end{cases}$$

The equations of backpropagation give:

$$
\begin{aligned}
dx_t^L &= W^{L+1}(t) \\
dw^L(t) &= m^{-a_{L+1}} x_t^L \\
dh_t^l &= dx_t^l \odot \sigma'(h_t^l) \\
dx_t^{l-1} &= (W^l(t))^\top dh_t^l \\
dw^l(t) &= m^{-a_l} dh_t^l (x_t^{l-1})^\top, \\
db^l(t) &= m^{-a_l} dh_t^l.
\end{aligned}
$$

As noted in Definition 2.2.1 Remark 2.2.2, one has for $l \in [1, L]$,

$$\Delta w^l(t) = -\eta m^{-c_l} \chi_t dw^l(t) = -\eta m^{-(a_l + c_l)} \chi_t dh_t^l (x_t^{l-1})^\top, \tag{A.2}$$

$$\Delta W^l(t) = m^{-a_l} \Delta w^l(t) = -\eta m^{-(2a_l + c_l)} \chi_t dh_t^l (x_t^{l-1})^\top, \tag{A.3}$$

$$\Delta B^l(t) = m^{-a_l} \Delta b^l(t) = -\eta m^{-(2a_l + c_l)} \chi_t dh_t^l, \tag{A.4}$$

and for $l = L + 1$

$$\Delta w^{L+1}(t) = -\eta m^{-c_l} \chi_t dw^{L+1}(t) = -\eta m^{-(a_{L+1} + c_{L+1})} \chi_t x_t^L, \tag{A.5}$$

$$\Delta W^{L+1}(t) = m^{-a_{L+1}} \Delta w^{L+1}(t) = -\eta m^{-(2a_{L+1} + c_{L+1})} \chi_t x_t^L, \tag{A.6}$$

$$\Delta B^{L+1}(t) = m^{-a_{L+1}} \Delta b^{L+1}(t) = -\eta m^{-(2a_{L+1} + c_{L+1})} \chi_t. \tag{A.7}$$

Z variables. As described in Section A.2.2, the variables Z with a superscript will be used to denote the random variable whose law describes the evolution of all coordinates of a given vector of the forward or backward pass at a given layer in the limit $m \to \infty$.

Tilde variables. For $z \in \{h_t^l, x_t^l, dh_t^l, dx_t^l\}$, we will use $\tilde{z}$ to denote a variable *"without scale"*, *i.e.*, such that $Z^{\tilde{z}}$ has positive and finite variance (see Definition A.7.1). When we do so, we always have $z = \lambda \tilde{z}$ for some scalar λ (which might depend on m). The tilde variables of the backward pass for $t \geq 1$ might have different expressions in different contexts or in different proofs, but we still use the same notation every time as the exact definition should always be clear from the context.

A.2 . An overview of the Tensor Program technique

The Tensor Program technique, first introduced by in Yang (2019), was initially developed to better understand the behavior at initialization of networks whose weights are initialized i.i.d. with standard Gaussians as the number of units in each layer grows to infinity. Since the output of a hidden unit in layer $l \geq 2$ is given by $\sum_{q=1}^m W_{pq}^l(0) x_{0,q}^{l-1}$, the magnitude of the weights need to be downscaled by some

negative power of m to avoid blow-up as $m \to \infty$. Scalings which have naturally appeared in the literature are $m^{-1/2}$ and m^{-1}, and lead to different types of limits.

Using a first version of the Tensor Program (referred to as NETSOR), it is shown in (Yang, 2019) that the output at initialization of a neural network of **any architecture** (fully-connected, recurrent, convolutional, with normalization, attention, ...) whose weights are initialized with $W^l(0) = m^{-1/2}U^l$ for $l \geq 2$ (*i.e.*, $a_l = 0$ and $b_l = 1/2$ for $l \geq 2$ in the ac-parameterization) is a Gaussian process in the infinite-width limit.

Going further, and in the light of the recent literature on the neural tangent kernel, Yang (2020a) studies the first backward pass of networks initialized as above in the limit where $m \to \infty$ and has shown that the neural tangent kernel at initialization, defined as $K(\xi, \bar{\xi}) := \left\langle \nabla_\theta f_0(\theta(0); \xi), \nabla_\theta f_0(\theta(0); \bar{\xi}) \right\rangle$ converges to a deterministic limit for any architecture.

Finally, and most importantly for our work, the Tensor Program is extended in (Yang, 2020b) to cover the forward and backward passes of networks of any architecture **at any time step** and not just at initialization. The crucial step taken in (Yang, 2020b) is to be able to describe the evolution of quantities where both a weight matrix W^l and its transpose $(W^l)^\top$ are involved. (Yang and Hu, 2021) then applies the results and theorems of (Yang, 2020b) in the particular context of ac-parameterizations (or rather abc-parameterizations as defined by Yang and Hu, 2021) to describe the infinite-width limits of neural networks with different parameterizations.

A.2.1 . Intuition behind the technique

To explain the intuition behind the Tensor Program technique and how it comes into play for neural networks, let us first look at the forward pass of a fully-connected network with L hidden layers after t steps of SGD. Assume single samples $(\xi_0, y_0), \dots (\xi_{t-1}, y_{t-1})$ are used at each step for simplicity. Consider a neural network in any ac-parameterization and an input ξ to the network. Using Equation (A.3) for the updates, the forward pass of the network at time t is given by:

$$
\begin{aligned}
h_t^1 &= W^1(0)\xi - \eta m^{-(2a_1+c_1)} \sum_{s=0}^{t-1} \chi_s \left(\xi_s^\top \xi\right) dh_s^1 \\
h_t^l &= W^l(0)x_t^{l-1} - \eta m^{-(2a_l+c_l)} \sum_{s=0}^{t-1} \chi_s \left((x_s^{l-1})^\top x_t^{l-1}\right) dh_s^l \qquad l \in [2, L] \\
f_t(\xi) &= \left(W^{L+1}(0)\right)^\top x_t^L - \eta m^{-(2a_{L+1}+c_{L+1})} \sum_{s=0}^{t-1} \chi_s (x_s^L)^\top x_t^L .
\end{aligned}
$$

To understand what happens in the forward pass, one thus needs to understand the behavior of the multiplication by i.i.d. Gaussian matrices, that of vectors dh_s^l of the backward pass as well as that of the inner products $(x_s^{l-1})^\top x_t^{l-1}$. As $m \to \infty$,

the sums defining the matrix multiplications and inner products involve an infinity of terms and one must therefore understand how those quantities scale in the limit.

Before we dive into the matrix multiplications, let us look more precisely at what the vectors dh_s^l look like. We have:

$$dh_s^l = dx_s^l \odot \sigma'(h_s^l)$$
$$dx_s^l = (W^{l+1}(0))^\top dh_s^{l+1} - \eta m^{-(2a_l+c_l)} \sum_{u=0}^{s} \chi_u \left((dh_u^{l+1})^\top dh_s^{l+1} \right) x_u^l \quad l \in [2, L].$$

We observe that inner products appear again, and that in contrast with the forward pass, it is now the multiplication by the transpose of i.i.d. Gaussian matrices which appears.

We already see that two main quantities appear in the calculations: The initial i.i.d. Gaussian matrices, and vectors which are generated either (i) through the multiplication of another vector with a Gaussian matrix or its transpose, or (ii) through some form of non-linearity involving other vectors as well as the activation function σ and/or its derivative σ'. Before trying to understand how the inner products behave, let us first dive into the multiplication by i.i.d. Gaussian matrices.

Multiplication by i.i.d. Gaussian matrices

The multiplication of a random vector by an i.i.d. Gaussian matrix can happen in two different scenarios: (i) the input vector is independent of the Gaussian weights, and (ii) the input vector is correlated with the Gaussian weights, which, in the case of neural networks, will translate into saying that the transpose of the weight matrix is used somewhere to compute the input vector.

Independent input vector. Consider a list $(x_q)_{q \in \mathbb{N}^*}$ of i.i.d. random variables with finite first and second moments, independent of U^l, and consider multiplying this vector by the i.i.d. Gaussian matrix U^l. At any finite-width m the p-th entry of $U^l x$ is given by

$$\sum_{q=1}^{m} U_{pq}^l x_q \underset{m \to \infty}{\simeq} m^{1/2} \mathcal{N}(0, \mathbb{E}[x_1^2])$$

The terms $(U_{pq}^l x_q)_{q \geq 1}$ are i.i.d. with mean 0 and finite variance $\mathbb{E}[x_1^2]$ because x_q is independent of U_{pq}^l. Therefore, by a central limit argument, the sum will behave like $m^{1/2}\mathcal{N}(0, \mathbb{E}[x_1^2])$ for large m. It is thus natural to scale the sum by $m^{-1/2}$, or equivalently to consider $\widehat{W}^l = m^{-1/2} U^l$ (as defined in Equation A.1) for matrix multiplications.

With the above result in mind, we take a look at the first forward pass at initialization of a network where all the weight matrices are initialized as $W^l(0) = \widehat{W}^l$ (*i.e.*, $a_1 = 0$, $a_l = 1/2$, $l \in [2, L+1]$). We consider an input $\xi \in \mathbb{R}^d$ to the network and compute the pre-activations of each layer recursively. For the first layer, we get that for any $p \in [m]$,

$$\begin{aligned} h^1_{0,p} &= (\widehat{W}^1 \xi)_p = (U^1 \xi)_p \\ &= \sum_{q=1}^{d} \xi_q U^1_{pq} \sim \mathcal{N}(0, ||\xi||^2) \end{aligned}$$

Since the $(U^1_{pq})_q$ are i.i.d. standard Gaussians, the linear combination above is also a Gaussian with mean 0 and variance $\sum_q \xi_q^2 = ||\xi||^2$. Note that since the lists $(U^1_{pq})_q$ are independent for different p, the vector h^1_0 has i.i.d. coordinates all distributed as $\mathcal{N}(0, ||\xi||^2)$. We also note that adding a bias term initialized as $\mathcal{N}(0,1)$ would simply change the variance to $||\xi||^2+1$.

Then for the second layer we get that for any $p \in [m]$:

$$h^2_{0,p} = \frac{1}{\sqrt{m}} \sum_{q=1}^{m} U^2_{pq} \sigma(h^1_{0,q}) \xrightarrow[m\to\infty]{law} \mathcal{N}(0, \mathbb{E}[\sigma(h^1_{0,1})^2])$$

The terms $(U^2_{pq}\sigma(h^1_{0,q}))_q$ are i.i.d. with mean zero, and by a central limit argument, we have that the coordinates of h^2_0 converge in law towards $\mathcal{N}(0, \mathbb{E}[\sigma(h^1_{0,1})^2])$ where $\mathbb{E}[\sigma(h^1_{0,1})^2])$ is simply $\mathbb{E}[\sigma(Z)^2])$ with $Z \sim \mathcal{N}(0, ||\xi||^2)$. Those coordinates are also independent (and Gaussian at any finite width m) **conditionally** on h^1_0 because the lists $(U^2_{pq})_q$ are independent (Gaussians) for different p. The different coordinates of h^2_0 are identically distributed at any finite width m and remain so in the limit. They are not strictly speaking independent at finite width but the intuition is that they become so in the limit $m \to \infty$ as they also become Gaussian, and that is how they should be thought of in the context of the Tensor Program.

Repeating the calculations above at every layer, we can intuitively describe the forward pass in the infinite-width limit by describing the law of a single random variable Z_l for each layer (whose law is the common law of all the coordinates of the pre-activations h^l_0), and by the hand-wavy calculations above, we get the following recursion for the variables Z:

$$\begin{aligned} Z_1 &\sim \mathcal{N}(0, ||\xi||^2) \\ Z_{l+1} &\sim \mathcal{N}(0, \mathbb{E}[\sigma(Z_l)^2]), \qquad\qquad l \in [1, L] \end{aligned}$$

Having discussed the case where the input vectors are not correlated with the weight matrix, we now move on to the case where there is some correlation between the two.

Correlated input vector. As the simplest form of correlation, we consider a vector $x = (\widehat{W}^l)^\top z$ where $(z_q)_{q\in\mathbb{N}^*}$ is a list of i.i.d. random variables independent of $\widehat{W}^l$ with finite first and second moments, and we consider the result of the multiplication $h = \widehat{W}^l x$. For any $p \in [m]$, we have

$$\begin{aligned} h_p &= \sum_{q=1}^{m}\sum_{r=1}^{m} \widehat{W}^l_{pq}\widehat{W}^l_{rq} z_r \\ &= \left[\frac{1}{m}\sum_{q=1}^{m}\left(U^l_{pq}\right)^2\right] z_p + \frac{1}{\sqrt{m}}\sum_{r\neq p} z_r \left(\frac{1}{\sqrt{m}}\sum_{q=1}^{m} U^l_{pq}U^l_{rq}\right) \end{aligned}$$

By the law of large numbers, the first term will converge almost surely to z_p as $m \to \infty$. For the second term, the intuition is that for any $r \neq p$ the terms $(1/\sqrt{m})\sum_q U^l_{pq}U^l_{rq}$ become distributed as independent Gaussians as m becomes large by a central limit argument. Then, by another central limit argument, intuitively, the sum over $r \neq p$ should also becomes distributed as $\mathcal{N}(0, \mathbb{E}[z_1^2])$. In the limit $m \to \infty$, we thus expect the coordinates of $h = \widehat{W}^l(\widehat{W}^l)^\top z$ to be the sum of two terms: a first term distributed as z_1 where the correlation between the entries of $\widehat{W}^l$ and $(\widehat{W}^l)^\top$ comes into play, and a second term distributed as $\mathcal{N}(0, \mathbb{E}[z_1^2])$ which is purely Gaussian and where the correlation between the entries of $\widehat{W}^l$ and $(\widehat{W}^l)^\top$ has no effect.

The aim of the Tensor Program series (Yang, 2019, 2020a,b) is to formalize those intuitions into theorems and rigorous calculations. Of course, the calculations become more complex when we introduce non-linearities and consider later steps in training than the initialization, but what the Tensor Program shows is that the intuitions above still hold.

To summarize, the intuition is that in the large-width limit, the coordinates of pre-activation vectors become i.i.d. and we thus only need to track the law of a single real-valued random variable. Therefore, any average of some function of the coordinates should converge to an expectation in the limit $m \to \infty$ by a law of large number argument. Finally, any multiplication by $\widehat{W}^l$ yields two terms where one is purely Gaussian and the other depends on the expression of the vector that is multiplied by $\widehat{W}^l$ in function of $(\widehat{W}^l)^\top$.

A.2.2 . Mathematical formalism

The mathematical formalism of the Tensor Program goes beyond neural network computations and describes the evolution of any computational systems (with some restrictions) in the limit $m \to \infty$. The computational system is comprised of different vectors whose dimensions are equal to m which can be generated from a set of initial vectors in various ways. The Tensor Program is defined by the

sequence of mathematical operations which produce the vectors from previously generated vectors. The operations are the same at any given width m, only the size of the vectors and matrices involved change with m, and the aim of the Tensor Program is to provide the tools (formalism and theorems) to be able to described the behavior of the system in the limit $m \to \infty$. As described in the intuitions of the previous section A.2.1, the coordinates of vectors in the program are roughly i.i.d. as $m \to \infty$ and variables Z are introduced to described the common law of the coordinates in the limit $m \to \infty$.

Initial vectors. Consider a set $\mathcal{V} := \left\{v^1, \dots, v^N\right\} \in (\mathbb{R}^m)^N$ of *initial vectors* such that:

(i) the coordinates $(v_p)_{p\in[m]}$ are i.i.d. for any $v \in \mathcal{V}$ and any m. We call Z^v a real-valued random variable whose law is the same as that of all the coordinates.

(ii) The joint law of $Z^{\mathcal{V}} := (Z^{v^1}, \dots, Z^{v^N})$ is a Gaussian $\mathcal{N}(\mu_{\text{init}}, \Sigma_{\text{init}})$ for any m (the variables Z^v do not actually depend on m, but this is simply to say that at any width m and for any $p \in [m]$, the law of $(v_p^1, \dots, v_p^N)$ is the same N-dimensional Gaussian).

Initial scalars. Similarly, we define a list of initial scalars $\theta_1, \dots, \theta_M$ which can depend on m and for which the only requirement is that each θ_r converges almost surely to some finite limit $\mathring{\theta}_r$ as $m \to \infty$.

Initial Gaussian matrices. Consider a set $\mathcal{W} := \left\{\widehat{W}^1, \dots, \widehat{W}^P\right\} \in (\mathbb{R}^{m\times m})^P$, such that $\widehat{W}^r_{pq} \sim \mathcal{N}(0, 1/m)$ i.i.d. over p, q for any r, and the $(\widehat{W}^r)_{r\in[P]}$ are independent of each other and independent of the vectors in $\mathcal{V}$. Since we consider a more general setting than neural networks, we do not index those matrices by l and can have $P \neq L$ but for neural networks, those initial matrices will always be the initialization of the weight matrices of the intermediate layers $l \in [2, L]$, appropriately scaled.

Generation of new vectors/scalars. Given previously generated vectors $v^1, \dots, v^k$, previously generated scalars $\theta_1, \dots, \theta_r$, and a non-linearity $\psi(\cdot\,;\cdot) : \mathbb{R}^k \times \mathbb{R}^r \to \mathbb{R}$, we can, in the following ways, generate:

`MatMul` a vector $z = \widehat{W}v$ for any $v \in \{v^1, \dots, v^k\}$ and $\widehat{W} \in \mathcal{W}$.

`NonLin` a vector $z = \psi(v^1, \dots, v^k\,;\, \theta_1, \dots, \theta_r)$ where ψ is taken element-wise, *i.e.*, $z_p = \psi(v_p^1, \dots, v_p^k\,;\, \theta_1, \dots, \theta_r)$ for any $p \in [m]$ and for any m.

`Moment` a scalar $\omega = \frac{1}{m}\sum_{p=1}^m \psi(v_p^1, \dots, v_p^k\,;\, \theta_1, \dots, \theta_r) \in \mathbb{R}$.

The non-linearity used does not have to actually depend on all the previous vectors and/or scalars, but we present the operations this way for simplicity.

Given those operations, the Tensor Program framework allows to seamlessly describe the infinite-width limit of the computational system defining a given Tensor Program by tracking recursively the laws of the variables Z whose law represents the common law of the coordinates of a given vector. Indeed, every vector z in the program (initial or generated using previous vectors in the program) will roughly have i.i.d. coordinates in the limit $m \to \infty$, and the Tensor Program associates a real-valued random variable Z^z to the vector z. Then, associated with the operations on vectors and scalars above are the following operations on the corresponding variables Z which come as their natural counterparts in the infinite-width limit to track the evolution of the laws of the variables Z:

`ZInit` For initial vectors $v \in \mathcal{V}$, define $\dot{Z}^v = 0$ and $\widehat{Z}^v = Z^v$. The purpose of those notations will become clear in the `ZMatMul` section.

`ZMoment` Given a scalar $\omega = (1/m)\sum_{p=1}^m \psi(z_p^1, \dots, z_p^k\,;\, \theta_1, \dots, \theta_r)$, define

$$\mathring{\omega} = \mathbb{E}\left[\psi(Z^{z^1}, \dots, Z^{z^k}\,;\, \mathring{\theta}_1, \dots, \mathring{\theta}_1)\right] \tag{A.8}$$

`ZNonLin` Given $z = \psi(z^1, \dots, z^k\,;\, \theta_1, \dots, \theta_r)$, define:

$$Z^z = \psi(Z^{z^1}, \dots, Z^{z^k}\,;\, \mathring{\theta}_1, \dots \mathring{\theta}_r) \tag{A.9}$$

`ZMatMul` Given $z = \widehat{W}v$ for a previous vector v and $\widehat{W} \in \mathcal{W}$, $Z^z = \widehat{Z}^z + \dot{Z}^z$ is the sum of two terms:

`ZHat` $\widehat{Z}^z \sim \mathcal{N}\left(0, \mathbb{E}\left[(Z^v)^2\right]\right)$ is a purely Gaussian term. Additionally, if we let $\mathcal{W}_{\widehat{W}}$ be the set of all vectors in the program of the form $\widehat{W}u$ for some u in the program, the vector $Z^{\mathcal{W}_{\widehat{W}}} = (Z^h)_{h \in \mathcal{W}_{\widehat{W}}}$ is defined to be jointly Gaussian with covariance matrix given by:

$$\mathrm{cov}(Z^{Wx}, Z^{Wy}) = \mathbb{E}[Z^x Z^y]$$

Moreover, the vector $Z^{\mathcal{W}_{\widehat{W}}}$ is defined to be mutually independent of the list of Z^u for u in $\{\widehat{Z}^v : v \in \mathcal{V} \cup_{W \in \mathcal{W} \cup \mathcal{W}^\top, W \neq \widehat{W}} \mathcal{W}_W\}$ where $\mathcal{W}^\top := \{\widehat{W}^\top : \widehat{W} \in \mathcal{W}\}$, and $\mathcal{W}_W$ is the set of vectors in the program of the form Wu for some vector u in the program.

`ZDot` $\dot{Z}^z$ comes from the potential interactions (correlations) between $\widehat{W}$ and $\widehat{W}^\top$ in the computation of z. One can always unwind the expression of Z^v and express it in function of the $\widehat{Z}^{\widehat{W}^\top y}$ for some y in

the program, that is we can always write Z^v by expanding the variables as $Z^v = \phi(\widehat{Z}^{\widehat{W}^\top y^1}, \ldots, \widehat{Z}^{\widehat{W}^\top y^k}, \widehat{Z}^{x^1}, \ldots \widehat{Z}^{x^r}; \mathring{\theta}_1, \ldots, \mathring{\theta}_s)$ with $x^1, \ldots, x^r$ such that $\widehat{W}^\top$ is never used in the computation of those vectors. Then, define:

$$\dot{Z}^z = \sum_{j=1}^{k} \mathbb{E}\left[\frac{\partial Z^v}{\partial Z^{\widehat{W}^\top y_j}}\right] Z^{y_j} \tag{A.10}$$

where $\partial Z^v / \partial Z^{\widehat{W}^\top y_j}$ is simply defined as the j-th partial derivative of ϕ above when expressing Z^v as required for $\dot{Z}$. As noted in (Yang and Hu, 2021), if ϕ is not everywhere differentiable, one can leverage Stein's lemma to replace the formula in Equation (A.10) by a linear algebra formula.

Now that we have introduced the necessary concepts and described the content of a Tensor Program, we can move on to present the main theorem derived in (Yang and Hu, 2021) which connects the mathematical operations used at finite-width with the infinite-width limit of the computational system defining a Tensor Program. The "master theorem" formulated in (Yang and Hu, 2021) is surprisingly simple (although the proof is much more intricate) yet very powerful, and goes as follows (see Yang and Hu, 2021, Theorem 7.4):

Theorem A.2.1 (Master Theorem)**.** *Given a Tensor Program, for any vectors $x^1, \ldots, x^k$ and scalars $\theta_1, \ldots, \theta_r$ in the program, and for any pseudo-Lipschitz non-linearity ψ (see Definition 2.3.1, page 108), one has that:*

$$\frac{1}{m}\sum_{p=1}^{m} \psi(x_p^1, \ldots, x_p^k; \theta_1, \ldots, \theta_r) \xrightarrow[m\to\infty]{a.s.} \mathbb{E}\left[\psi\left(Z^{x^1}, \ldots, Z^{x^k}; \mathring{\theta}_1, \ldots, \mathring{\theta}_r\right)\right]$$

Remark.

1. The theorem essentially states that even though the coordinates of vectors in the program are not rigorously i.i.d**,** they appear so from the perspective of the average by a suitable non-linearity so that a law of large number type of result holds. Note that for neural networks, even though the coordinates of the (pre-)activations follow the same law when using i.i.d. initialization for the weights, it is not *a priori* clear that we can consider them as independent copies, and thus that we can summarize the computations using a single real-valued variable, but the master theorem shows that from the perspective of averaging, this is in fact the case in the infinite-width limit.

2. In (Yang and Hu, 2021), different versions of the Tensor Program are presented in the sense that different classes of non-linearities are allowed.

These differences induce minor subtleties in the master theorem and in the proofs. However, most of the results in the main text of the paper require that the non-linearities be pseudo-Lipschitz (which is the stronger assumption), both in `NonLin` and in the master theorem. The Assumption 2 on the activation function σ and its derivative σ' ensures that any quantity appearing in the forward or backward computation of a neural network can be expressed as pseudo-Lipschitz non-linearity.

3. What the Tensor Program and its master theorem show is that to understand the behavior of the computational system in the infinite-width limit, one simply needs to track the operations on the variables Z which mimic the recursive operations in the computational system. Then, quantities which involve sum over coordinates such as inner products between the vectors in the program (which occur in the forward and backward passes of a neural network, as well as in the computation of the neural tangent kernel), or norm computations are easily described, when properly re-normalized, through expectations involving the corresponding variables Z. The main difficulty is that it is actually hard (computationally and in the mathematical formulation) to track the correlations between different Z because, as explained in (Yang and Hu, 2021), of the necessary unwinding in the definition of $\dot{Z}$, so that the computational graph associated with the operations on the variables Z is hard to implement in practice.

A.2.3 . The maximal update parameterization μP

We close this section by presenting briefly the maximal update parameterization considered in (Yang and Hu, 2021). To quantify the learning abilities of a given parameterization, Yang and Hu (2021) introduce the notions of *feature learning* and *feature kernel evolution* at a given layer $l \in [1, L]$, which we recall below. Both these definitions concern the large-width limit of the networks:

Definition A.2.1 (Feature Learning)**.** An ac-parameterization is said to admit **feature learning** at the l-th layer if the quantity $\Delta x_t^l(\xi) := x_t^l(\xi) - x_0^l(\xi)$ is such that there exists a training routine for which, almost surely, there exists a constant $C > 0$ such that $||\Delta x_t^l(\xi)||^2/m \geq C$ for large enough m.

Definition A.2.2 (Kernel Evolution)**.** An ac-parameterization is said to **evolve the feature kernel** at the l-th layer if the following quantity $\Delta F_t^l(\xi, \bar{\xi}) := \left[x_t^l(\xi)^\top x_t^l(\bar{\xi}) - x_0^l(\xi)^\top x_0^l(\bar{\xi})\right]/m$ is such that there exists a training routine for which, almost surely, there exists a constant $C > 0$ such that for large enough m, $\Delta F_t^l(\xi, \bar{\xi}) \geq C$.

(Yang and Hu, 2021) goes about categorizing whether different ac-parameterizations admit feature learning or not. One of the striking result presented is that there is essentially a dichotomy (depending on the values of $(a_l, c_l)_{l\in[L+1]}$) among ac-parameterizations: an ac-parameterization either admits feature learning (and

evolves the feature kernel) or is in the kernel regime, meaning that the quantities in definitions A.2.1 and A.2.2 converge to 0 almost surely so that in the infinite width limit, the evolution of the prediction function f_t is deterministic and depends only on the previous prediction function f_{t-1} and the loss at time $(t-1)$ through a (deterministic) kernel $K(\xi, \bar{\xi}) = \lim_{m\to\infty} (x_0^L(\xi))^\top x_0^L(\bar{\xi})/m$ (or a rescaled version thereof).

The categorization result proved in (Yang and Hu, 2021) holds for a certain class of ac-parameterizations which are deemed *stable* and *non-trivial*. Stable refers to the fact that the pre-activations and output (h_0^l and $f_0(\xi)$ respectively) at initialization do not blow-up as $m \to \infty$ at any layer. As already hinted in Section A.2.1, this corresponds to having $a_1 = 0$ and $a_l \geq 1/2$ for $l \in [2, L+1]$. Non-trivial refers to the fact that the pre-activations of all layers do not converge to 0 almost surely as $m \to \infty$ at initialization. This corresponds to having $a_1 \leq 0$ and $a_l \leq 1/2$ for $l \in [2, L]$. It is mentioned in (Yang and Hu, 2021) that those parameterizations for which the pre-activations of the intermediate layers converge to 0 almost surely should stay at their initialization throughout the course of training, and we actually prove in Section 2.3, using the Tensor Program technique, that this is the case when $L \geq 3$ in the setting where $a_1 = 0$ and $a_l = 1$ for $l \in [2, L+1]$ (*i.e.*, integrable parameterizations) unless one uses large (polynomial in m) initial learning rates, a scenario which is not covered in (Yang and Hu, 2021). We show that in this case, integrable parameterizations are only trivial at initialization (the pre-activations of all layers except the first one converge to 0 in the infinite-width limit) and are actually in a feature learning regime at all layers after the first gradient step ($t \geq 1$).

The maximal update parameterization μP introduced in (Yang and Hu, 2021) is the result of the analysis of the values of a_l, and c_l for which the parameterization admits feature learning at every layer, and maximally so in the sense that if we were to reduce the value of a_l then the Δx_t^l introduced in Definition A.2.1 or the pre-activations h_t^l would blow-up as $m \to \infty$. In essence, μP corresponds to the values of a_l, and c_l for which Δx_t^l is as large as possible (with regards to its dependency on m) at every layer without creating any instabilities (pre-activations or updates blowing-up) in the limit $m \to \infty$. A quick analysis of the updates at $t = 0$ shows that the choice $a_1 = 0$, $a_l = 1/2$ for $l \in [2, L]$, and $a_{L+1} = 1$ associated with $c_l = -1$ for all $l \in [L+1]$ achieves this, and it is rigorously shown in (Yang and Hu, 2021) that this choice of ac-parameterization induces an update such that, $||\Delta W^l(t) x_t^{l-1}||^2/m = \Theta(1)$. We thus adopt the following definition for μP which is the same as in (Yang and Hu, 2021, Definition 5.1) but re-parameterized to remove the redundant b in the abc-parameterization:

Definition A.2.3 (μP)**.** The maximal update parameterization μP is defined by

the following choice of parameterization:

$$\begin{aligned} &a_1 = 0, && c_1 = -1, \\ &a_l = 1/2, && c_l = -1, \qquad l \in [2, L], \\ &a_{L+1} = 1, && c_{L+1} = -1. \end{aligned}$$

A.3 . Useful preliminary results

We show in this section a couple of useful results which will prove helpful in the proofs.

A.3.1 . Positive finite moments of pseudo-Lipschitz functions of Gaussians

Lemma A.3.1 (Positive finite moments with polynomially bounded non-linearities)**.** *Let ϕ be a polynomially bounded non-linearity which is not almost everywhere 0, and let $Z \sim \mathcal{N}(0, v^2)$ with $v^2 < \infty$. Then, for any $p \in \mathbb{R}_+$:*

(i) $0 \leq \mathbb{E}[|\phi(Z)|^p] < \infty$,

(ii) if in addition $v^2 > 0$, $0 < \mathbb{E}[|\phi(Z)|^p] < \infty$.

Proof. If $v^2 = 0$, and then $\phi(Z) = \phi(0)$ almost surely, so that $\mathbb{E}[|\phi(Z)|^p] = |\phi(0)|^p < \infty$.

Now, assume $v^2 > 0$. Since ϕ is bounded by a polynomial of some degree $r > 0$, $|\phi(z)| \leq C(1 + |z|^r)$ for some $C > 0$. Then, $|\phi(z)|^p = \exp(p \ln(|\phi(z)|)) \leq C^p(1 + |z|^r)^p$. Since $v^2 > 0$, we have

$$\begin{aligned} \mathbb{E}[|\phi(Z)|^p] &= \frac{1}{\sqrt{2\pi v^2}} \int_{\mathbb{R}} |\phi(z)|^p e^{-z^2/2v^2} \mathrm{d}z \\ &\leq \frac{1}{\sqrt{2\pi v^2}} \int_{\mathbb{R}} C^p(1 + |z|^r)^p e^{-z^2/2v^2} \mathrm{d}z < \infty. \end{aligned}$$

Finally, since ϕ is not almost everywhere 0, neither is $|\phi|^p$ which shows the integral in the first equality above is not 0, and gives $\mathbb{E}[|\phi(Z)|^p] > 0$. □

A.3.2 . The Z dots are 0 in the first forward-backward pass

Lemma A.3.2 ($\dot{Z} = 0$ in the first forward-backward pass)**.** *Consider an acparameterization of an L-hidden layer fully-connected neural network with $a_1 \geq 0$ and $a_l \geq 1/2$ for $l \in [2, L + 1]$, and with a non-linearity satisfying Assumption 2. Then for any $l \geq 2$, $\dot{Z}^{\widehat{W}^l x_0^{l-1}} = 0$, and for any $l \in [1, L]$, $\dot{Z}^{(\widehat{W}^l)^\top dh_0^l} = 0$.*

Remark. This lemma applies to the NTK, μP, and integrable parameterizations (in particular IP-LLR) as well as HP and HPZ.

Proof. Consider any ac-parameterization of a fully-connected neural network which has $a_1 \geq 0$ and $a_l \geq 1/2$ for $l \in [2, L+1]$, and with a non-linearity satisfying Assumption 2. Define $\omega_1 = m^{-a_1}$ and $\omega_l = m^{-(a_l - 1/2)}$ for $l \geq 2$, and the initial scalar $\alpha_{L+1} := m^{-a_{L+1}}$. The conditions on the a_l guarantee that the ω_l converge almost surely to either 0 or 1 and and α_{L+1} converges almost surely to 0, which allows applying the rules of the Tensor Program.

For any $l \in [2, L]$, since the computation of x_0^{l-1}, and thus of $Z^{x_0^{l-1}}$ do not involve $(\widehat{W}^l)^\top$, $\dot{Z}^{\widehat{W}^l x_0^{l-1}} = 0$ as per the `ZDot` rule of the Tensor Program. In addition, $Z^{h_1^l} = \omega_1(\widehat{Z}^{\widehat{W}^1 \xi} + \widehat{Z}^{v^1})$ and by definition, $\widehat{Z}^{\widehat{W}^1 \xi} \sim \mathcal{N}(0, ||\xi||^2)$ and $\widehat{Z}^{v^1} \sim \mathcal{N}(0,1)$ are independent Gaussians, which shows that $Z^{h_1^l} \sim \mathcal{N}(0, \mathring{\omega}_1^2(||\xi||^2+1))$ whose variance is finite because $\mathring{\omega}_1^2 \in \{0,1\}$. By Lemma A.3.1, this also shows that $\mathbb{E}[(Z^{x_0^1})^2] < \infty$. Let $l \in [2, L]$ and assume that $\mathbb{E}[(Z^{h_0^{l-1}})^2] < \infty$ and $\mathbb{E}[(Z^{x_0^{l-1}})^2] < \infty$. We have $h_0^l = \omega_l \widehat{W}^l x_0^{l-1} + m^{-a_l} v^l$. Since m^{-2a_l} converges to 0 almost surely, we can consider it as an initial scalar in the program, which gives by `ZNonLin` $Z^{h_0^l} = \mathring{\omega}_l \widehat{Z}^{\widehat{W}^l x_0^{l-1}} + 0 \times \widehat{Z}^{v_1^l}$. $\widehat{Z}^{v_1^l} \sim \mathcal{N}(0,1)$ by definition since v^l is an initial vector in the program, so that $Z^{h_0^l} = \mathring{\omega}_l \widehat{Z}^{\widehat{W}^l x_0^{l-1}} \sim \mathcal{N}(0, \mathring{\omega}_l^2 \mathbb{E}[Z^{x_0^{l-1}})^2])$ whose variance is finite by the induction hypothesis and because $\mathring{\omega}_l \in \{0,1\}$. Then by Lemma A.3.1, we also get that $\mathbb{E}[(Z^{x_0^l})^2] < \infty$, which concludes the induction.

Let us now deal with the first backward pass for any ac-parameterization. The result will essentially boil down to having the expectation of the derivatives defining the $\dot{Z}$ being 0 because the weight matrices are initialized with 0 mean and because of an independence argument. We have $dx_0^L = W^{L+1}(0) = m^{-a_{L+1}} U^{L+1}$, and $dh_0^L = dx_0^L \odot \sigma'(Z^{h_0^L})$. By `ZNonLin` we thus have

$$\begin{aligned} Z^{dx_0^L} &= \mathring{\alpha}_{L+1} Z^{U^{L+1}}, \\ Z^{dh_0^L} &= \mathring{\alpha}_{L+1} Z^{U^{L+1}} \sigma'(Z^{h_0^L}). \end{aligned}$$

Now let $l \in [1, L]$. $dx_0^{l-1} = (\widehat{W}^l)^\top dh_0^l$ gives

$$Z^{(\widehat{W}^l)^\top dh_0^l} = \widehat{Z}^{(\widehat{W}^l)^\top dh_0^l} + \dot{Z}^{(\widehat{W}^l)^\top dh_0^l},$$

and to understand what $\dot{Z}^{(\widehat{W}^l)^\top dh_0^l}$ is, we need to expand the expression of $Z^{dh_0^l}$ in function of variables which were generated with $\widehat{W}^l$. So far, the only variable where $\widehat{W}^l$ was used is $h_0^l = \omega_l \widehat{W}^l x_0^{l-1}$ (with the convention that $x_0^0 = \xi_0$). We thus need to expand the expression of $Z^{dh_0^l}$ in function of $\widehat{Z}^{\widehat{W}^l x_0^{l-1}}$. We have, for $l = L$

$$\begin{aligned} Z^{dh_0^L} &= \mathring{\alpha}_{L+1} Z^{U^{L+1}} \sigma'(\mathring{\omega}_L Z^{\widehat{W}^l x_0^{L-1}}) \\ &= \mathring{\alpha}_{L+1} \widehat{Z}^{U^{L+1}} \sigma'(\mathring{\omega}_L \widehat{Z}^{\widehat{W}^l x_0^{L-1}}), \end{aligned}$$

where the last equality stems from the fact that $Z^{\widehat{W}^L x_0^{L-1}} = \widehat{Z}^{\widehat{W}^L x_0^{L-1}}$ in the first forward pass, and the fact that U^{L+1} is an initial vector in the program which gives by definition $\widehat{Z}^{U^{L+1}} = Z^{U^{L+1}}$. We can formally write this as

$$Z^{dh_0^L} = \Psi(\widehat{Z}^{\widehat{W}^L x_0^{L-1}}, \widehat{Z}^{U^{L+1}}; \mathring{\alpha}_{L+1}, \mathring{\omega}_L),$$

where $\Psi(z_1, z_2; \theta_1, \theta_2) := \theta_1 z_2 \sigma'(\theta_2 z_1)$ is a pseudo-Lipschitz function because σ' is, and we have

$$\frac{\partial \Psi}{\partial z_1}(z_1, z_2; \theta_1, \theta_2) = \theta_1 \theta_2 z_2 \sigma''(\theta_2 z_1).$$

We get that by definition

$$\begin{aligned}
\dot{Z}^{(\widehat{W}^L)^\top dh_0^L} &= \mathbb{E}\left[\frac{\partial Z^{dh_0^L}}{\partial \widehat{Z}^{\widehat{W}^L x_0^{L-1}}}\right] Z^{x_0^{L-1}} \\
&= \mathbb{E}\left[\frac{\partial \Psi}{\partial z_1}(\widehat{Z}^{\widehat{W}^L x_0^{L-1}}, \widehat{Z}^{U^{L+1}}; \mathring{\alpha}_{L+1}, \mathring{\omega}_L)\right] Z^{x_0^{L-1}} \\
&= \mathring{\alpha}_{L+1}\mathring{\omega}_L \mathbb{E}[Z^{U^{L+1}} \sigma''(\mathring{\omega}_L \widehat{Z}^{\widehat{W}^L x_0^{L-1}})] Z^{x_0^{L-1}} \\
&= \mathring{\alpha}_{L+1}\mathring{\omega}_L \underbrace{\mathbb{E}[\widehat{Z}^{U^{L+1}}]}_{0} \underbrace{\mathbb{E}[\sigma''(\mathring{\omega}_L \widehat{Z}^{\widehat{W}^L x_0^{L-1}})]}_{<\infty} \underbrace{Z^{x_0^{L-1}}}_{<\infty \ a.s.},
\end{aligned}$$

where the last equality stems from the fact that by `ZHat`, $\widehat{Z}^{\widehat{W}^L x_0^{L-1}}$ is independent of $\widehat{Z}^{U^{L+1}}$ because U^{L+1} is an initial vector in the program. The fact that the second expectation finite is because $\mathring{\omega}_L \in \{0, 1\}$, σ'' is polynomially bounded, and $\widehat{Z}^{\widehat{W}^L x_0^{L-1}}$ is a Gaussian with mean 0 and finite variance since $\mathbb{E}[(Z^{x_0^{L-1}})^2] < \infty$. This gives $\dot{Z}^{(\widehat{W}^l)^\top dh_0^L} = 0$.

Now suppose $l \in [1, L-1]$ and assume $\dot{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}} = 0$ which gives $Z^{(\widehat{W}^{l+1})^\top dh_0^{l+1}} = \widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}}$. We have

$$\begin{aligned}
Z^{dh_0^l} &= Z^{dx_0^l} \sigma'(Z^{h_0^l}) \\
&= \mathring{\omega}_{l+1} Z^{(\widehat{W}^{l+1})^\top dh_0^{l+1}} \sigma'(\mathring{\omega}_l Z^{\widehat{W}^l x_0^{l-1}}) \\
&= \mathring{\omega}_{l+1} \widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}} \sigma'(\mathring{\omega}_l \widehat{Z}^{\widehat{W}^l x_0^{l-1}})
\end{aligned}$$

where we have used that previous $\dot{Z}$ are 0 to replace the Z with $\widehat{Z}$. We can once more formally write this as

$$Z^{dh_0^l} = \Psi(\widehat{Z}^{\widehat{W}^l x_0^{l-1}}, \widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}}; \mathring{\omega}_{l+1}, \mathring{\omega}_l)$$

with exactly the same Ψ as for $l = L$. We get that by definition

$$\begin{aligned}
\dot{Z}^{(\widehat{W}^l)^\top dh_0^l} &= \mathbb{E}\left[\frac{\partial Z^{dh_0^l}}{\partial \widehat{Z}^{\widehat{W}^l x_0^{l-1}}}\right] Z^{x_0^{l-1}} \\
&= \mathbb{E}\left[\frac{\partial \Psi}{\partial z_1}(\widehat{Z}^{\widehat{W}^l x_0^{l-1}}, \widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}}; \mathring{\omega}_{l+1}, \mathring{\omega}_l)\right] Z^{x_0^{l-1}} \\
&= \mathring{\omega}_{l+1}\mathring{\omega}_l \mathbb{E}[\widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}} \sigma''(\mathring{\omega}_l \widehat{Z}^{\widehat{W}^l x_0^{l-1}})] Z^{x_0^{l-1}} \\
&= \mathring{\omega}_{l+1}\mathring{\omega}_l \underbrace{\mathbb{E}[\widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}}]}_{0} \underbrace{\mathbb{E}[\sigma''(\mathring{\omega}_l \widehat{Z}^{\widehat{W}^l x_0^{l-1}})]}_{<\infty} \underbrace{Z^{x_0^{l-1}}}_{<\infty a.s.} \\
&= 0
\end{aligned}$$

Where the first expectation is 0 because by definition $\widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}}$ is a Gaussian with 0 mean and an easy induction (from $l = L$ to $l = 1$) shows that, as for the forward pass, $\mathbb{E}[(Z^{dx_0^l})^2] < \infty$ and $\mathbb{E}[(Z^{dh_0^l})^2] < \infty$, which implies that $\widehat{Z}^{(\widehat{W}^{l+1})^\top dh_0^{l+1}}$ has finite variance. The second expectation is finite because $\mathring{\omega}_l \in \{0, 1\}$, $\widehat{Z}^{\widehat{W}^l x_0^{l-1}}$ is a Gaussian with 0 mean by definition and finite variance, and because σ'' is polynomially bounded since σ' is pseudo-Lipschitz. □

A.3.3 . Gaussian output in the infinite-width limit

Lemma A.3.3 (Gaussian output)**.** *For every $m \in \mathbb{N}^*$, let x^m and w^m be independent random vectors in $\mathbb{R}^m$ such that*

$$\begin{cases} \frac{1}{m}||x^m||^2 \xrightarrow[m\to\infty]{a.s.} \sigma_\infty^2 \\ w_j^m \sim \mathcal{N}(0, 1/m) \textit{ i.i.d. over } j = 1, \dots, m. \end{cases}$$

Then

$$(w^m)^\top x^m \xrightarrow[m\to\infty]{law} \mathcal{N}(0, \sigma_\infty^2)$$

Proof. Consider two sequences of independent vectors of growing dimension $(w^m)_m$ and $(x^m)_m$ as in Lemma A.3.3. Conditionally on x^m, the random variable $(w^m)^\top x^m$ follows a Gaussian $\mathcal{N}(0, ||x^m||^2/m)$ distribution. Since $||x^m||^2/m$ converges to σ_∞^2 almost surely, the conditional distribution of $(w^m)^\top x^m$ given x^m converges to a Gaussian $\mathcal{N}(0, \sigma_\infty^2)$ distribution. The lemma follows. □

A.3.4 . Convergence of the coordinates to the limiting distribution Z

Lemma A.3.4 (Convergence to the limit distribution)**.** *For any vector h in the Tensor Program we have for any $\alpha \in \mathbb{N}^*$,*

$$h_\alpha \xrightarrow[m\to\infty]{law} Z^h$$

Remark.

1. Let $h^1, \dots, h^k$ be k vectors in the program, let $\theta_1, \dots, \theta_p$ be p scalars in the program, and let $\phi : \mathbb{R}^{k+p} \to \mathbb{R}$ be a pseudo-Lipschitz function. Then applying the previous Lemma A.3.4 to $h := \phi(h^1, \dots, h^k; \theta_1, \dots, \theta_p)$ (which is in the program by `NonLin`), shows that for any α, $\phi(h^1_\alpha, \dots, h^k_\alpha; \theta_1, \dots, \theta_p)$ converges in law to $Z^h = \phi(Z^{h^1}, \dots, Z^{h^k}; \mathring{\theta}_1, \dots, \mathring{\theta}_p)$.
2. A stronger form of convergence can occur depending on the parameterization we look at and the context. Indeed, if for example Z^h turns out to be a constant, then we already get convergence in probability instead of in law. If in addition the convergence is "fast enough", it can occur almost surely.

Proof. Let h be a vector in the program, and consider the corresponding random variable Z^h. All we need is to prove that for any $\alpha \in \mathbb{N}^*$ and any bounded 1-Lipschitz function ϕ, we have $\mathbb{E}[\phi(h_\alpha)] \to \mathbb{E}[\phi(Z^h)]$, as m goes to infinity. We first observe that the Master Theorem A.2.1 ensures the convergence

$$\frac{1}{m}\sum_{\beta=1}^{m} \phi(h_\beta) \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[\phi(Z^h)].$$

Secondly, for any m, the distribution of $h_1, \dots, h_m$ is exchangeable by symmetry, so that we get

$$\mathbb{E}[\phi(h_\alpha)] = \mathbb{E}\left[\frac{1}{m}\sum_{\beta=1}^{m} \phi(h_\beta)\right] \xrightarrow[m\to\infty]{} \mathbb{E}[\phi(Z^h)],$$

where the convergence is obtained by dominated convergence, which concludes the proof. □

A.4 . Proof of the triviality of IPs: Proposition 2.3.1

Proof. Fix a time $t \geq 0$ and an input $\xi \in \mathbb{R}^d$ for the whole proof. We first show that the coordinates of the (pre-)activations of any layer $l \geq 2$ converge to 0 almost surely at initialization. To that end, we prove that the corresponding Z's are equal to 0. Then we show a similar result for the backward pass, and finally conclude the proof by an induction.

A.4.1 . Proof at $t = 0$

First forward pass

Tensor program setup: We consider a Tensor Program as defined in

$$\begin{cases} \widehat{W}^{l+1} = U^{L+1}, \\ U^1\xi_0, \dots, U^1\xi_t, U^1\xi, \\ v^1, \dots, v^L, \end{cases}$$

and the initial scalars

$$\begin{cases} \chi_0, \dots, \chi_t, \\ \omega := m^{-1/2}, \nu := m^{-1}, \tau := m^{-2}, \\ m^{-1} v^{L+1}; \end{cases}$$

and with initial weight matrices

$$\widehat{W}^2, \dots, \widehat{W}^L.$$

Recall that the $\widehat{W}^l$ are defined in Equation (A.1) of Appendix A.1. Note that for any $m \in \mathbb{N}^*$ and $j \in [m]$, we have

$$\left(U_j^{L+1}, (U^1\xi_0)_j, \dots, (U^1\xi_t)_j, (U^1\xi)_j, v_j^1, \dots, v_j^L\right) \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & I_L \end{pmatrix}\right),$$

where $M := \text{Gram}(\xi_0, \dots, \xi_t, \xi) = (\xi_r^\top \xi_s)_{0 \le r,s \le t+1}$ and I_L is the identity matrix of size $L \times L$. where we have set $\xi_{t+1} := \xi$.

Convergence of the initial scalars: ω, ν, τ as well as $m^{-1}v^{L+1}$ all converge almost surely towards 0. For the χ_s we will show below in the proof that they all converge to constants almost surely, thereby meeting the requirements of the Tensor Program. It is important to note that there is no circular logic to prove the χ_s converge almost surely. Indeed, each time we apply the master theorem to prove the convergence of $f_s(\xi_s)$ to a constant almost surely and thus that of χ_s, we apply it to a restricted Tensor Program where only the scalars $(\chi_r)_{0 \le r < s}$ appear (and there is no such scalar needed to prove the convergence of χ_0 as shown below) which will already have been proved to converge almost surely.

1st forward pass: We drop the dependency of the forward and backward passes on ξ for brevity. $h_0^1 = U^1\xi + v^1$ is the sum of two initial vectors in the program and has iid Gaussian coordinates $\mathcal{N}(0, ||\xi||^2+1)$. By definition, $\widehat{Z}^{h_0^1} = \widehat{Z}^{U^1\xi} + \widehat{Z}^{v^1} \sim \mathcal{N}(0, ||\xi||^2+1)$ since the two Gaussians appearing in the sum are independent. By `NonLin`, we have that since $x_0^1 = \sigma(h_0^1)$, $Z^{x_0^1} = \sigma(Z^{h_0^1})$. Note that $\mathbb{E}[\sigma(Z^{h_0^1})^2] < \infty$ since $Z^{h_0^1}$ is Gaussian with finite variance and σ is pseudo-Lipschitz and thus polynomially bounded.

Since $L \ge 2$, we can write $h_0^2 = m^{-1/2}\widehat{W}^2 x_0^1 + m^{-1}v^2$ (otherwise there is no h_0^2 and we simply have $f_0(\xi_0) = m^{-1}(U^2)^\top x_0^1$), which implies by `NonLin` that $Z^{h_0^2} = \mathring{\omega} Z^{\widehat{W}^2 x_0^1} + \mathring{\nu} Z^{v^2}$ with $\mathring{\omega} = \mathring{\nu} = 0$ and

$$Z^{\widehat{W}^2 x_0^1} = \widehat{Z}^{\widehat{W}^2 x_0^1} + \dot{Z}^{\widehat{W}^2 x_0^1}.$$

$\dot{Z}^{\widehat{W}^2 x_0^1} = 0$ by Lemma A.3.2, and $\widehat{Z}^{\widehat{W}^2 x_0^1} \sim \mathcal{N}(0, \mathbb{E}[(Z^{x_0^1})^2])$ and $0 \leq \mathbb{E}[(Z^{x_0^1})^2] < \infty$. We thus have $Z^{h_0^2} = \mathring{\omega}\widehat{Z}^{\widehat{W}^2(0) x_0^1} = 0$. Similarly, we also get that $\mathring{nu} Z^{v^2} = 0$. We then have by `ZNonLin` $Z^{x_0^2} = \sigma(Z^{h_0^2}) = \sigma(0) = 0$.

Let $l \in [2, L-1]$ and assume $Z^{h_0^l} = 0$. Then, $Z^{x_0^l} = \sigma(Z^{h_0^l}) = 0$, and since $h_0^{l+1} = \omega \widehat{W}^{l+1} x_0^l + \nu v^{l+1}$, by `ZNonLin`, $Z^{h_0^{l+1}} = \mathring{\omega} Z^{\widehat{W}^{l+1} x_0^l} + \mathring{\nu} Z^{v^{l+1}}$ where by `ZMatMul`,

$$Z^{\widehat{W}^{l+1} x_0^l} = \widehat{Z}^{\widehat{W}^{l+1} x_0^l} + \dot{Z}^{\widehat{W}^{l+1} x_0^l},$$

and $\dot{Z}^{\widehat{W}^{l+1} x_0^l} = 0$ by Lemma A.3.2. By `ZHat`, $\widehat{Z}^{\widehat{W}^{l+1} x_0^l} \sim \mathcal{N}(0, \mathbb{E}[(Z^{x_0^l})^2])$, and since $\mathring{\omega} = 0$, $\mathring{\omega}\widehat{Z}^{\widehat{W}^{l+1} x_0^l} = 0$. Similarly, $\mathring{\nu} Z^{v^{l+1}} = 0$. Then, by `ZNonLin` $Z^{x_0^{l+1}} = \sigma(Z^{h_0^{l+1}}) = \sigma(0) = 0$, which concludes the induction.

We thus have only to deal with the last layer $L+1$ to finish the first forward pass. We have $f_0(\xi) = m^{-1}((U^{L+1}(0))^\top x_0^L + v^{L+1}) = (1/m)\sum_{i=1}^m U_i^{L+1} x_{0,i}^L + m^{-1} v^{L+1}$. Since U^{L+1} and x_0^L are vectors in the program, $(1/m)\sum_{i=1}^m U_i^{L+1} x_{0,i}^L$ is a scalar in the program by the `Moment` rule, and it therefore converges almost surely to $\mathbb{E}[Z^{U^{L+1}} Z^{x_0^L}]$ by the Master Theorem. Now because U^{L+1} is an initial vector in the program, by definition, $Z^{U^{L+1}} = \widehat{Z}^{U^{L+1}} \sim \mathcal{N}(0,1)$ is independent of $Z^{x_0^L}$. We thus get $\mathbb{E}[Z^{U^{L+1}} Z^{x_0^L}] = \mathbb{E}[Z^{U^{L+1}}]\mathbb{E}[Z^{x_0^L}] = 0$. On the other hand, $m^{-1} v^L$ is an initial scalar in the program which converges to 0 almost surely, so that $f_0(\xi)$ converges almost surely to 0.

First backward pass

1st backward pass: We can apply the previous reasoning of the forward pass with ξ_0 instead of ξ and we get that $f_0(\xi_0) \to 0$ almost surely. Therefore, since $\chi_0 = \partial_2 \ell(y_0, f_0(\xi_0))$ and $\partial_2 \ell(y_0, \cdot)$ is continuous by assumption, $\chi_0 \to \partial_2 \ell(y_0, 0) =: \mathring{\chi}_0$ almost surely. We have $dx_0^L = m^{-1} U^{L+1}$ which makes it a vector in the program by `NonLin`, and $Z^{dx_0^L} = \mathring{\nu} Z^{U^{L+1}}$. Since $Z^{U^{L+1}} \sim \mathcal{N}(0,1)$ has finite variance and $\mathring{\nu} = 0$, we have $Z^{dx_0^L} = 0$. $dh_0^L = dx_0^L \odot \sigma'(h_0^L)$ implies by `ZNonLin` $Z^{dh_0^L} = Z^{dx_0^L} \sigma'(Z^{h_0^L}) = 0 \times \sigma'(0) = 0$.

One has:

$$Z^{m dx_0^{L-1}} = \mathring{\omega}(\widehat{Z}^{(\widehat{W}^L)^\top (m dh_0^L)} + \dot{Z}^{(\widehat{W}^L)^\top (m dh_0^L)}),$$

where $m dh_0^L = U^{L+1} \odot \sigma'(h_0^L)$. By Lemma A.3.2, $\dot{Z}^{(\widehat{W}^L)^\top (m dh_0^L)} = 0$ (essentially, $\widehat{W}^l$ never appears in the computation of dh_0^L), and by `ZHat`, $\widehat{Z}^{(\widehat{W}^L)^\top (m dh_0^L)} \sim \mathcal{N}(0, \mathbb{E}[(Z^{m dh_0^L})^2])$, and by independence of $Z^{U^{L+1}}$ and $Z^{h_0^L}$,

$$\mathbb{E}[(Z^{m dh_0^L})^2]) = \mathbb{E}[(Z^{U^{L+1}})^2])\mathbb{E}[\sigma'(Z^{h_0^L})^2]) = \sigma'(0)^2$$

which is finite. Since $\mathring{\omega} = 0$ we get $Z^{mdx_0^{L-1}} = 0$. $dh_0^{L-1} = dx_0^{L-1} \odot \sigma'(h_0^{L-1})$ implies by `ZNonLin` $Z^{mdh_0^{L-1}} = Z^{mdx_0^{L-1}} \sigma'(Z^{h_0^{L-1}}) = 0 \times \sigma'(Z^{h_0^{L-1}}) = 0$.

Let $l \in [2, L]$ (which is non-empty since $L \geq 2$) and assume $Z^{mdx_0^l} = Z^{mdh_0^l} = 0$. $mdx_0^{l-1} = \omega(\widehat{W}^l)^\top (mdh_0^l)$ implies by `ZMatMul`

$$Z^{mdx_0^{l-1}} = \mathring{\omega}(\widehat{Z}^{(\widehat{W}^l)^\top (mdh_0^l)} + \dot{Z}^{(\widehat{W}^l)^\top (mdh_0^l)}).$$

By Lemma A.3.2, $\dot{Z}^{(\widehat{W}^l)^\top (mdh_0^l)} = 0$, and by `ZHat`, $\widehat{Z}^{(\widehat{W}^l)^\top (mdh_0^l)} \sim \mathcal{N}(0, \mathbb{E}[(Z^{mdh_0^l})^2])$. By the assumption above, $\mathbb{E}[(Z^{mdh_0^l})^2]) = 0$, and since $\mathring{\omega} = 0$ we get $Z^{mdx_0^{l-1}} = 0$. $dh_0^{l-1} = dx_0^{l-1} \odot \sigma'(h_0^{l-1})$ implies by `ZNonLin` $Z^{mdh_0^{l-1}} = Z^{mdx_0^{l-1}} \sigma'(Z^{h_0^{l-1}}) = 0 \times \sigma'(Z^{h_0^{l-1}})$. $Z^{h_0^{l-1}}$ is not 0 if $l = 2$, but since it is Gaussian with finite variance, and σ' is pseudo-Lipschitz by assumption, $\sigma'(Z^{h_0^{l-1}})$ is finite almost surely, and $Z^{mdh_0^{l-1}} = 0$ almost surely, which concludes the induction.

A.4.2 . Induction step

Induction: Since we proved the result of the theorem for $t = 0$ in the first forward pass, we might as well assume $t \geq 1$. Let $s \in [0, t-1]$ be an integer. In all that follows, for any $r \in [0, s]$, for $z \in \{h_r^l, x_r^l, dh_r^l, dx_r^l\}$, we use z to denote $z(\xi_r)$. We make the following induction hypothesis: for any $r \in [0, s]$

$$\begin{cases} Z^{h_r^1} = Z^{U^1 \xi_r + v^1} \sim \mathcal{N}(0, ||\xi_r||^2 + 1) \\ Z^{h_r^l} = 0 \text{ almost surely}, \ l \in [2, L] \\ f_r(\xi_r), f_r(\xi) \to 0 \text{ almost surely} \\ \chi_r \to \mathring{\chi}_r := \partial_2 \ell(y_r, 0) \text{ almost surely} \\ Z^{mdx_r^l} = Z^{mdh_r^l} = 0 \text{ almost surely}, \ l \in [1, L-1], \\ Z^{mdx_r^L} = U^{L+1}. \end{cases}$$

The aim is then to prove the same claims for $r = s + 1$. Let us first start with the expressions of $\Delta W^l(s+1)$ and $\Delta B^l(s+1)$. We will use Equation (A.3) and the fact that $c_l + 2 \geq 0$ if $l \in [2, L]$, and $c_l + 1 \geq 0$ for $l = 1$, and $l = L + 1$. We have by Equations (A.3) and (A.6)

$$\begin{aligned} \Delta W^1(s+1) &= -\eta m^{-c_l} \sum_{r=0}^{s} \chi_r dh_r^1 \xi_r^\top, \\ \Delta W^l(s+1) &= -\eta m^{-(2+c_l)} \sum_{r=0}^{s} \chi_r dh_r^l (x_r^{l-1})^\top, \quad l \in [2, L], \\ \Delta W^{L+1}(s+1) &= -\eta m^{-(1+c_{L+1})} \sum_{r=0}^{s} \chi_r x_r^L / m, \end{aligned}$$

and by Equations (A.4) and (A.7)

$$\Delta B^1(s+1) = -\eta m^{-c_l} \sum_{r=0}^{s} \chi_r dh_r^1,$$
$$\Delta B^l(s+1) = -\eta m^{-(2+c_l)} \sum_{r=0}^{s} \chi_r dh_r^l, \quad l \in [2, L],$$
$$\Delta B^{L+1}(s+1) = -\eta m^{-(1+c_{L+1})} \sum_{r=0}^{s} \chi_r / m.$$

In the following, we use for $z \in \{h_{s+1}^l, x_{s+1}^l, dh_{s+1}^l, dx_{s+1}^l\}$, we use z to denote $z(\xi)$ (and not $z(\xi_{s+1})$ for now). Using that in the Naive-IP, $c_1 = c_{L+1} = -1$, and $c_l = -2$ for $l \in [2, L]$, we have

$$\Delta W^1(s+1)\xi + \Delta B^1(s+1) = -\eta \sum_{r=0}^{s} (\xi_s^\top \xi + 1)\chi_r (m dh_r^1),$$
$$\Delta W^l(s+1)x_{s+1}^{l-1} + \Delta B^l(s+1) = -\eta \sum_{r=0}^{s} \chi_r \frac{((x_r^{l-1})^\top x_{s+1}^{l-1}) + 1}{m} (m dh_r^l), \;\; l \in [2, L],$$
$$(\Delta W^{L+1}(s+1))^\top x_{s+1}^L + \Delta B^{L+1}(s+1) = -\eta \sum_{r=0}^{s} \chi_r \frac{(x_r^L)^\top x_{s+1}^L + 1}{m}.$$

To prove the claims above for $r = s+1$, we will first induct from $l = 1$ to $l = L$ for the forward pass and then induct from $l = L$ to $l = 1$ for the backward pass.

Forward pass at step $s+1$

Forward pass at step $(s+1)$**:** $h_{s+1}^1 = U^1\xi + v^1 + \Delta W^1(s+1)\xi + \Delta b^1(s+1)$ and by `ZNonLin`

$$\begin{aligned} Z^{h_{s+1}^1} &= \widehat{Z}^{U^1\xi} - \eta \sum_{r=0}^{s} (\xi_s^\top \xi + 1) \mathring{\chi}_r \underbrace{Z^{m dh_r^1}}_{0 \; a.s.} \\ &= \widehat{Z}^{U^1\xi} = Z^{h_0^1(\xi)} \quad \text{almost surely.} \end{aligned}$$

Note that the scalars $(\chi_r)_{0 \le r \le s}$ are now valid scalars in the program by the induction hypothesis which allows applying the Tensor Program rules with those scalars as well as the master theorem. This gives $Z^{h_{s+1}^1} \sim \mathcal{N}(0, ||\xi||^2 + 1)$, and we then have $Z^{x_{s+1}^1} = \sigma(\widehat{Z}^{h_0^1(\xi)}) = Z^{x_0^1(\xi)}$ for which we have already proven $\mathbb{E}[(Z^{x_0^1(\xi)})^2] < \infty$.

$$h_{s+1}^2 = \omega \widehat{W}^2 x_{s+1}^1 + \tau v^2 - \eta \sum_{r=0}^{s} \chi_r \frac{(x_r^1)^\top x_{s+1}^1 + 1}{m} (m dh_r^2).$$

Because x^1_{s+1} is a vector in the program, by `ZMatMul`

$$Z^{\widehat{W}^2 x^1_{s+1}} = \widehat{Z}^{\widehat{W}^2 x^1_{s+1}} + \dot{Z}^{\widehat{W}^2 x^1_{s+1}},$$

and because $Z^{x^1_{s+1}} = \sigma(\widehat{Z}^{U^1\xi+v^1})$ is only a function of the initial vectors $U^1\xi$ and v^1, and not of any vector computed used $(\widehat{W}^2)^\top$, $\dot{Z}^{\widehat{W}^2 x^1_{s+1}} = 0$ by `ZDot`, and $\widehat{Z}^{\widehat{W}^2 x^1_{s+1}} \sim \mathcal{N}(0, \mathbb{E}[(Z^{x^1_{s+1}})^2])$ is a Gaussian with finite variance by `ZHat`. $(x^1_r)^\top x^1_{s+1}/m$ is a valid scalar in the program by the moment rule, and by the Master theorem,

$$((x^1_r)^\top x^1_{s+1} + 1)/m \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{x^1_r} Z^{x^1_{s+1}}] = \mathbb{E}[\sigma(Z^{U^1\xi_r+v^1})\sigma(Z^{U^1\xi+v^1})],$$

and because $U^1\xi_r$, v^1 and $U^1\xi$ are initial vectors in the program, $(Z^{U^1\xi_r+v^1}, Z^{U^1\xi+v^1})$ is jointly Gaussian by definition with finite covariance matrix

$$\begin{pmatrix} ||\xi_r||^2+1 & \xi_r^\top\xi + 1 \\ \xi^\top\xi_r + 1 & ||\xi||^2+1 \end{pmatrix},$$

which ensures the expectation above is finite because σ is polynomially bounded since it is pseudo-Lipschitz. We thus habe

$$Z^{h^2_{s+1}} = 0 \times \underbrace{\widehat{Z}^{\widehat{W}^2 x^1_{s+1}}}_{<\infty} + 0 \times Z^{v^2} - \eta \sum_{r=0}^{s} \underbrace{\mathring{\chi}_r}_{<\infty} \underbrace{\mathbb{E}[Z^{x^1_r} Z^{x^1_{s+1}}]}_{<\infty} \underbrace{Z^{mdh^2_r}}_{0}$$

$$Z^{h^2_{s+1}} = 0.$$

We then get $Z^{x^2_{s+1}} = \sigma(0) = 0$ and thus $\mathbb{E}[(Z^{x^2_{s+1}})^2] = 0$.

Let $l \in [2, L-1]$ and assume $Z^{h^l_{s+1}} = 0$.

$$h^{l+1}_{s+1} = \omega\widehat{W}^{l+1} + \tau v^{l+1} + x^l_{s+1} - \eta \sum_{r=0}^{s} \chi_r \frac{(x^l_r)^\top x^l_{s+1} + 1}{m} (mdh^{l+1}_r).$$

Now, since x^l_{s+1} is a vector in the program, $((x^l_r)^\top x^l_{s+1} + 1)/m$ is a scalar in the program by the `Moment` operation, which converges almost surely, by the Master Theorem, to

$$\mathbb{E}[Z^{x^l_r} Z^{x^l_{s+1}}] = \mathbb{E}[\sigma(Z^{h^l_r})\sigma(Z^{h^l_{s+1}})] = \sigma(0)^2 = 0.$$

By `ZNonLin`,

$$Z^{h^{l+1}_{s+1}} = \mathring{\omega} Z^{\widehat{W}^{l+1} x^l_{s+1}} + \mathring{\tau} Z^{v^{l+1}} - \eta \sum_{r=0}^{s} \underbrace{\mathring{\chi}_r}_{<\infty} \underbrace{\mathbb{E}[Z^{x^l_r} Z^{x^l_{s+1}}]}_{<\infty} \underbrace{Z^{mdh^{l+1}_r}}_{0}.$$

On the other hand,

$$Z^{\widehat{W}^{l+1} x^l_{s+1}} = \widehat{Z}^{\widehat{W}^{l+1} x^l_{s+1}} + \dot{Z}^{\widehat{W}^{l+1} x^l_{s+1}},$$

and since $Z^{x^l_{s+1}} = \sigma(Z^{h^l_{s+1}}) = \sigma(0) = 0$ is a constant almost surely, the derivatives defining $\dot{Z}$ are equal to 0 (its expression as a function of the previous $\widehat{Z}$ is a constant because any $\widehat{Z}$ gets multiplied by 0) so that $\dot{Z}^{\widehat{W}^{l+1}x^l_{s+1}} = 0$, and $\widehat{Z}^{\widehat{W}^{l+1}x^l_{s+1}} \sim \mathcal{N}(0, \mathbb{E}[(Z^{x^l_{s+1}})^2]) = 0$. With $\mathring{\omega}_{l+1} = 0$ and $\mathring{\tau} = 0$, this yields $Z^{h^{l+1}_{s+1}} = 0$, and therefore $\mathbb{E}[(Z^{x^{l+1}_{s+1}})^2] = \mathbb{E}[\sigma(Z^{h^{l+1}_{s+1}})^2] = \sigma(0)^2 = 0$.

We now deal with the last layer $l = L + 1$ in the forward pass.

$$f_{s+1}(\xi) = m^{-1}(U^{L+1})^\top x^L_{s+1} - \eta \sum_{r=0}^{s} \chi_r \frac{(x^L_r)^\top x^L_{s+1} + 1}{m}.$$

Since $U^{L+1}, x^L_{s+1}, x^L_r$ are vectors in the program, by the Master Theorem, we have:

$$m^{-1}(U^{L+1})^\top x^L_{s+1} \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x^L_{s+1}}] = \sigma(0)\underbrace{\mathbb{E}[Z^{U^{L+1}}]}_{0} = 0,$$

and

$$\frac{(x^L_r)^\top x^L_{s+1} + 1}{m} \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{x^L_r} Z^{x^L_{s+1}}] = \sigma(0)^2 = 0.$$

We thus get

$$\sum_{r=0}^{s} \chi_r \frac{(x^L_r)^\top x^L_{s+1} + 1}{m} \xrightarrow[m\to\infty]{a.s.} \sum_{r=0}^{s} \underbrace{\mathring{\chi}_r}_{<\infty} \times 0 = 0.$$

This shows that

$$f_{s+1}(\xi) \xrightarrow[m\to\infty]{a.s.} 0.$$

Doing the exact same reasoning as above with ξ_{s+1} instead of ξ for $r = s + 1$ gives us the first 3 claims of the induction hypothesis for $r = s + 1$.

Backward pass at step $s + 1$

Backward pass at step $(s + 1)$**:** the fourth claim $\chi_{s+1} \to \mathring{\chi}_{s+1} = \partial_2\ell(y_{s+1}, 0)$ is a consequence of the fact that $f_{s+1}(\xi_{s+1}) \to 0$ almost surely, combined with the facts that $\chi_{s+1} = \partial_2\ell(y_{s+1}, f_{s+1}(\xi_{s+1}))$ and that $\partial_2\ell(y_{s+1}, \cdot)$ is continuous by assumption. In all the rest of this proof, for $z \in \{h^l_{s+1}, x^l_{s+1}, dh^l_{s+1}, dx^l_{s+1}\}$ we now use z to denote $z(\xi_{s+1})$ and not $z(\xi)$ anymore.

$mdx^L_{s+1} = w^{L+1}(s + 1) = U^{L+1} - \eta \sum_{r=0}^{s} \chi_r x^L_r$ yields by `ZNonLin`

$$Z^{mdx^L_{s+1}} = Z^{U^{L+1}} - \eta \sum_{r=0}^{s} \mathring{\chi}_r \underbrace{Z^{x^L_r}}_{0}$$

$$Z^{dx^L_{s+1}} = Z^{U^{L+1}}.$$

We thus have $Z^{dx^L_{s+1}} = \mathring{\tau} Z^{mdx^L_{s+1}} = 0$, and $Z^{dh^L_{s+1}} = Z^{dx^L_{s+1}} \sigma'(Z^{h^L_{s+1}}) = 0 \times \sigma'(0) = 0$ almost surely.

One has:

$$mdx^{L-1}_{s+1} = \omega(\widehat{W}^L)^\top (mdh^L_{s+1}) - \eta \sum_{r=0}^{s} \chi_r \frac{(mdh^l_r)^\top mdh^L_{s+1}}{m} x^{L-1}_r,$$

so that

$$Z^{mdx^{L-1}_{s+1}} = \mathring{\omega} Z^{(\widehat{W}^L)^\top (mdh^L_{s+1})} - \eta \sum_{r=0}^{s} \mathring{\chi}_r \mathbb{E}[Z^{mdh^l_r} Z^{mdh^l_{s+1}}] Z^{x^{L-1}_r}.$$

Now, we have $\mathbb{E}[Z^{mdh^l_r} Z^{mdh^l_{s+1}}] = \mathbb{E}[(Z^{U^{L+1}})^2] \sigma'(0)^2 = \sigma'(0)^2$ which is finite. On the other hand, because $Z^{mdh^L_{s+1}} = Z^{U^{L+1}}$ does not depend on $Z^{\widehat{W}^L}$, we get that $\dot{Z}^{(\widehat{W}^L)^\top (mdh^L_{s+1})} = 0$ and $\widehat{Z}^{(\widehat{W}^L)^\top (mdh^L_{s+1})} \sim \mathcal{N}(0,1)$ so that $\mathring{\omega} Z^{(\widehat{W}^L)^\top (mdh^L_{s+1})} = 0$. It follows that $Z^{mdx^{L-1}_{s+1}} = 0$, and since $Z^{mdh^{L-1}_{s+1}} = Z^{mdx^{L-1}_{s+1}} \sigma'(Z^{h^{L-1}_0})$ we also get $Z^{mdh^{L-1}_{s+1}} = 0$.

Let $l \in [2, L]$ and assume $Z^{mdx^l_{s+1}} = Z^{mdh^l_{s+1}} = 0$. Then

$$mdx^{l-1}_{s+1} = \omega(\widehat{W}^l)^\top (mdh^l_{s+1}) - \eta \sum_{r=0}^{s} \chi_r \frac{(mdh^l_r)^\top mdh^l_{s+1}}{m} x^{l-1}_r.$$

Since $(mdh^l_r)^\top$ and mdh^l_{s+1} are vectors in the program, $(mdh^l_r)^\top mdh^l_{s+1}/m$ is a scalar in the program which converges almost surely, by the Master Theorem, to $\mathbb{E}[Z^{mdh^l_r} Z^{mdh^l_{s+1}}] = 0$. On the other hand $\dot{Z}^{(\widehat{W}^l)^\top (mdh^l_{s+1})} = 0$ because $Z^{mdh^l_{s+1}}$ is a constant (its expression in function of the previous $\widehat{Z}$ is constant equal to 0), and $\widehat{Z}^{(\widehat{W}^l)^\top (mdh^l_0)} \sim \mathcal{N}(0, \mathbb{E}[(Z^{mdh^l_0})^2])$ is almost surely 0 because $\mathbb{E}[(Z^{mdh^l_0})^2]) = 0$. By `ZNonLin` we have

$$Z^{dx^{l-1}_{s+1}} = \underbrace{\mathring{\omega}}_{0} \underbrace{\widehat{Z}^{(\widehat{W}^l)^\top dh^l_0}}_{0} - \eta \sum_{r=0}^{s} \underbrace{\mathring{\chi}_r}_{<\infty} \underbrace{\mathbb{E}[Z^{dh^l_r} Z^{dh^l_{s+1}}]}_{0} \underbrace{Z^{x^{l-1}_r}}_{0}$$
$$Z^{dx^{l-1}_{s+1}} = 0.$$

Finally, $Z^{dh^{l-1}_{s+1}} = Z^{dx^{l-1}_{s+1}} \sigma'(Z^{h^{l-1}_{s+1}})$ yields $Z^{dh^{l-1}_{s+1}} = 0$ because $Z^{h^{l-1}_{s+1}} = 0$. This proves the last claim of the induction hypothesis for $r = s+1$ and thus concludes the induction and therefore the proof. □

A.5 . Preliminaries on positively homogeneous functions

In this section we give a description of activation functions σ satisfying Assumption 3. The fact that σ is positively p-homogeneous translates as

$$\sigma(z) = \begin{cases} \alpha z^p & \text{if } z \geq 0 \\ \beta |z|^p & \text{if } z < 0. \end{cases}$$

Additionally, one has

$$\sigma'(z) = \begin{cases} \alpha p z^{p-1} & \text{if } z \geq 0 \\ -\beta p |z|^{p-1} & \text{if } z < 0, \end{cases}$$

so that σ' is positively $(p-1)$-homogeneous with $\sigma'(0) = 0$. Since $p \geq 2$, both σ and σ' are continuous and σ' is differentiable everywhere except at 0 if $p = 2$. It is immediate to check that both σ and σ' are pseudo-Lipschitz and that σ, σ' and σ'' are also polynomially bounded functions. The non-negativity assumption on σ gives $\alpha \geq 0, \beta \geq 0$, the fact that σ is not identically 0 leads to $\alpha > 0$ or $\beta > 0$, and finally the fact that σ has faster growth on the positive part of the real line yields $\alpha > \beta \geq 0$. One notices that the faster growth assumption is stronger than the assumption that σ is not identically zero, and the latter could thus be gotten rid of. The conditions on α and β can thus simply be summarized as

$$\alpha > \beta \geq 0 \tag{A.11}$$

With these conditions, we have that $\sigma(z) > 0$ for $z > 0$, and $\sigma'(z)z \geq 0$ for $z \neq 0$, that is $\text{sign}(\sigma'(z)) = \text{sign}(z)$.

A.6 . Simplification of the first update for IPs with Assumption 4

Before we present the dynamics of IPs with smooth and positively homogeneous activation functions in the next Appendix A.7, we first show how under Assumption 4 the first forward and backward passes is effectively linearized in intermediate layers for IPs which thus behave as an IP with a "smooth" and 1-homogeneous activation function.

In all this section we consider and activation function satisfying Assumptions 2 and 4, and we consider parameterizations with no bias terms except at the first layer. We denote $\rho := \sigma'(0)$ for simplicity which is $\neq 0$ by assumption.

A.6.1 . Tilde variables

Definition A.6.1 (Scaleless variables at initialization)**.** Let $\xi \in \mathbb{R}^d$ be an input vector. We consider the following variables "without scale" at initialization:

$$\begin{cases} \tilde{h}_0^1(\xi) := U^1 \xi + v^1 \\ \tilde{x}_0^1(\xi) := \sigma(\tilde{h}_0^1(\xi)) \end{cases} \qquad \forall l \in [2, L], \begin{cases} \tilde{h}_0^l(\xi) := \widehat{W}^l \tilde{x}_0^{l-1}(\xi) \\ \tilde{x}_0^l(\xi) := \tilde{h}_0^l(\xi) \end{cases}$$

and define $\tilde{f}_0(\xi) := (\widehat{W}^{L+1})^\top \tilde{x}_0^L$, as well as

$$\begin{cases} d\tilde{x}_0^L(\xi) := U^{L+1} \\ d\tilde{h}_0^L(\xi) := U^{L+1} \end{cases} \quad \forall l \in [L-1], \begin{cases} d\tilde{x}_0^l(\xi) := (\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}(\xi) \\ d\tilde{h}_0^l(\xi) := d\tilde{x}_0^l(\xi) \end{cases}$$

where the $\widehat{W}^l$ are defined in Equation (A.1).

Remark. Notice that there is a slight difference in the tilde variables with positively homogeneous activation functions as in Definition A.7.1: in the forward pass, the term $\tilde{x}_0^l$ is simply equal to $\tilde{h}_0^l$ because of the linearization which occurs, yielding a similar effect to σ being identity, and in the backward pass, in $d\tilde{h}_0^l$ the term $\sigma'(h_0^l)$ is replaced with $\sigma'(\tilde{h}_0^l)$ in the presence of homogeneity. This is because with homogeneity, the "scale" of h_0^l can effectively be taken out of σ'. This is a small difference which does not matter for the core of the arguments: the proofs we present in Appendix A.7 below still work in the case of Assumption 4 with minor adjustments which we will discuss further.

In light of the remark above, Lemma A.7.1 also holds here with the slight difference that in the forward pass $Z^{\tilde{x}_0^l} = Z^{\tilde{h}_0^l}$ for $l \in [2, L]$, and in the backward pass, as we already saw in Appendix A.4, $Z^{h_0^l} = 0$ for $l \in [2, L]$, so that $Z^{d\tilde{h}_0^l} = \rho Z^{d\tilde{x}_0^l}$ for $l \in [2, L]$ which even simplifies the arguments in the proof of Lemma A.7.1.

A.6.2 . First forward pass

We start with a preliminary lemma before stating the results for the first forward pass.

Lemma A.6.1 (Linearization property around 0)**.** *Let z be a variable in a Tensor Program such that $|Z^z| < \infty$ almost surely and let σ satisfy Assumption 4. Then for any $a > 0$, it holds*

$$Z^{m^a \sigma(m^{-a} z)} = \rho Z^z.$$

Proof. Let a and σ be as in the lemma. For any m, by taking the first order expansion with remainder of σ around 0, and since $\sigma(0) = 0$ and $\sigma'(0) = \rho$, it holds $\sigma(m^{-a}z) = \rho m^{-a} z + R(m^{-a}z)$ where the first order remainder is $R(u) := \sigma(u) - \rho u$, and thus $m^a \sigma(m^{-a}z) = \rho z + m^a R(m^{-a}z)$. The assumption on σ ensures that the second term converges to 0: $|m^a R(m^{-a}z)| \leq m^a \frac{M}{2} m^{-2a} z^2 = \frac{M m^{-a}}{2} z$. Because Z^z is finite almost surely and $a > 0$, it follows that $\frac{M m^{-a}}{2} z^2$ converges to 0 almost surely and thus so does $m^a R(m^{-a}z)$, from which it follows that $Z^{m^a R(m^{-a}z)} = 0$. By the rules of the tensor program, it then comes $Z^{m^a \sigma(m^{-a}z)} = \rho Z^z + Z^{m^a R(m^{-a}z)} = \rho Z^z$. □

Lemma A.6.2 (First forward pass linearization)**.** *Consider an integrable parameterization of an L-layer neural network with σ satisfying Assumptions 2 and 4*

and a loss satisfying Assumption 1. Then, it holds:

$$Z^{h_0^1} = Z^{\tilde{h}_0^1}, \quad Z^{x_0^1} = Z^{\tilde{x}_0^1}$$
$$Z^{m^{(l-1)/2}h_0^l} = \rho^{l-2} Z^{\tilde{h}_0^l}, \quad Z^{m^{(l-1)/2}x_0^l} = \rho^{l-1} Z^{\tilde{h}_0^l}, \quad l \in [2, L]$$

This shows that the first forward pass is effectively linearized, and if $|\rho| \neq 1$, one should scale the standard deviation of the initial Gaussians by $|\rho|^{-1} = |\sigma'(0)|^{-1}$ to avoid explosion or vanishing when the (fixed) depth L is large.

Proof. Because $a_1 = 0$, there is no multiplying factor in front of $W_1(0) = \hat{W}^1$ and thus $Z^{h_0^1} = Z^{\tilde{h}_0^1}$ and $Z^{x_0^1} = Z^{\tilde{x}_0^1}$.

$h_0^2 = m^{-1/2}\hat{W}^2 x_0^1 = m^{-1/2}\tilde{h}_0^2$, which shows $Z^{m^{1/2}h_0^2} = Z^{\tilde{h}_0^2}$. In addition, one has $m^{1/2}x_0^2 = m^{1/2}\sigma(m^{-1/2}\tilde{h}_0^2)$ and by Lemma A.6.1, it holds $Z^{m^{1/2}x_0^2} = Z^{\rho Z^{\tilde{h}_0^2}}$.

Assume that $m^{(l-1)/2}Z^{h_0^l} = \rho^{l-2}Z^{\tilde{h}_0^l}$ and $Z^{m^{(l-1)/2}x_0^l} = \rho^{l-1}Z^{\tilde{h}_0^l}$ for some $l \in [2, L-1]$. Then $m^{l/2}h_0^{l+1} = \hat{W}^{l+1}(m^{(l-1)/2}x_0^l)$, and by the rules of the Tensor Program, we have $Z^{m^{l/2}h_0^l} = \hat{Z}^{\hat{W}^{l+1}(m^{(l-1)/2}x_0^l))} \sim \mathcal{N}(0, \mathbb{E}[(Z^{m^{(l-1)/2}x_0^l})^2]) = \rho^{l-1}\mathcal{N}(0, \mathbb{E}[(Z^{\tilde{x}_0^l})^2])$, which is the same law as $\rho^{l-1}Z^{\hat{W}^{l+1}Z^{\tilde{x}_0^l}} = \rho^{l-1}Z^{\tilde{h}_0^{l+1}}$ and thus we get $Z^{m^{l/2}h_0^l} = \rho^{l-1}Z^{\tilde{h}_0^{l+1}}$ where the equality is understood in law, which is the only thing that matters in the Tensor Program. Then, $m^{l/2}x_0^l = m^{l/2}\sigma(m^{-l/2}\left(m^{l/2}h_0^l\right))$, and thus by Lemma A.6.1, we get $Z^{m^{l/2}x_0^l} = \rho Z^{m^{l/2}h_0^l} = \rho^{l/2}Z^{\tilde{h}_0^{l+1}}$, which concludes the induction and with it the proof. □

A.6.3 . First backward pass

Lemma A.6.3 (First backward pass linearization)**.** *Consider an integrable parameterization of an L-layer neural network with σ satisfying Assumptions 2 and 4 and a loss satisfying Assumption 1. Then, it holds:*

$$Z^{mdx_0^L} = Z^{\tilde{dx}_0^L} = Z^{U^{L+1}}, \quad Z^{mdh_0^L} = \rho Z^{\tilde{dh}_0^L} = \rho Z^{U^{L+1}}$$
$$Z^{m^1 m^{(L-l)/2}dx_0^l} = \rho^{L-l} Z^{d\tilde{x}_0^l}, \quad Z^{m^1 m^{(L-l)/2}dh_0^l} = \rho^{L-l+1} Z^{d\tilde{h}_0^l}, \quad l \in [1, L-1]$$

Similarly to the first forward pass, if $|\rho| \neq 1$, one should scale the standard deviation of the initial Gaussians by $|\rho|^{-1} = |\sigma'(0)|^{-1}$ to avoid explosion or vanishing in the first backward pass when the (fixed) depth L is large.

Proof. By definition, $dx_0^L = W^{L+1}(0) = m^{-1}U^{L+1}$, and $dh_0^L = dx_0^L \odot \sigma'(h_0^L)$, and by the `ZNonLin` rule of the tensor program, given that $Z^{h_0^L} = 0$ by Appendix A.4, we have $Z^{mdx_0^L} = Z^{U^{L+1}} = Z^{d\tilde{x}_0^L}$ and $Z^{mdh_0^L} = Z^{U^{L+1}}\sigma'(0) = \rho Z^{d\tilde{h}_0^L}$.

Then, $m^1 m^{1/2} dx_0^{L-1} = (\hat{W}^L)^\top (mdh_0^L)$, and thus by `ZHat` it follows that $Z^{m^1 m^{1/2} dx_0^{L-1}} \sim \rho\mathcal{N}(0, \mathbb{E}[(Z^{d\tilde{h}_0^L})^2])$ which is the same law as $\rho Z^{d\tilde{x}_0^{L-1}}$, and thus $Z^{m^1 m^{1/2} dx_0^{L-1}} = \rho Z^{d\tilde{x}_0^{L-1}}$. $m^1 m^{1/2} dh_0^{L-1} = m^1 m^{1/2} dx_0^{L-1} \odot \sigma'(h_0^{L-1})$, and thus by `ZNonLin` $Z^{m^1 m^{1/2} dh_0^{L-1}} = \rho^2 Z^{d\tilde{h}_0^{L-1}}$.

Assume that $Z^{m^1 m^{(L-l)/2} dx_0^l} = \rho^{L-l} Z^{d\tilde{x}_0^l}$ and $Z^{m^1 m^{(L-l)/2} dh_0^l} = \rho^{L-l+1} Z^{d\tilde{h}_0^l}$ for some $l \in [2, L-1]$. Then, $m^1 m^{(L-l+1)/2} dx_0^{l-1} = (\hat{W}^l)^\top (m^1 m^{(L-l)/2} dh_0^l)$ and thus by `ZHat` we get that $Z^{m^1 m^{(L-l+1)/2} dx_0^{l-1}} = \hat{Z}^{(\hat{W}^l)^\top (m^1 m^{(L-l)/2} dh_0^l)} \sim \rho^{L-l+1} \mathcal{N}(0, \mathbb{E}[(Z^{d\tilde{h}_0^l})^2])$ which is the same law as $\rho^{L-l+1} Z^{d\tilde{x}_0^{l-1}}$, which shows $Z^{m^1 m^{(L-l+1)/2} dx_0^{l-1}} = \rho^{L-l+1} Z^{d\tilde{x}_0^{l-1}}$. Finally, as above, the following equality holds: $m^1 m^{(L-l+1)/2} dh_0^{l-1} = m^1 m^{(L-l+1)/2} dx_0^{l-1} \odot \sigma'(h_0^{L-1})$, and thus by `ZNonLin` we get $Z^{m^1 m^{(L-l+1)/2} dh_0^{l-1}} = \rho^{L-l+2} Z^{d\tilde{h}_0^{l-1}}$, which concludes the induction and with it the proof. □

A.6.4 . First gradient scales

Lemma A.6.4 (First weight updates' scales)**.** *Consider an integrable parameterization of an L-layer neural network with σ satisfying Assumptions 2 and 4 and a loss satisfying Assumption 1. Call $\mathring{\chi}_0 := \partial_2 \ell(0, y_0)$, and finally, let $\xi \in \mathbb{R}^d$ and $z_2, \dots, z_L$ be vectors in the program such that $\mathbb{E}[(Z^{z_l})^2] < \infty$, it holds:*

$$
\begin{aligned}
Z^{m^{(L+1)/2} \Delta W^1(1) \xi_1} &= -\eta \mathring{\chi}_0 \rho^{L-1} (\xi_0^\top \xi_1) Z^{d\tilde{h}_0^1} \\
Z^{m^{(L+2)/2} \Delta W^l(1) z_{l-1}} &= -\eta \mathring{\chi}_0 \rho^{L-1} \mathbb{E}[Z^{z_{l-1}} Z^{\tilde{x}_0^{l-1}}] Z^{d\tilde{h}_0^l} \\
m^{(L+1)/2} (\Delta W^{L+1})^\top z_L &\xrightarrow[m \to \infty]{a.s.} -\eta \mathring{\chi}_0 \rho^{L-1} \mathbb{E}[Z^{z_L} Z^{\tilde{x}_0^L}]
\end{aligned}
$$

Note that as soon as $\partial_2 \ell(0, y_0), \xi_0^\top \xi_1 \neq 0$ and the expectations are not equal to zero (we know they are finite by Lemma A.7.1), all the first weight updates generate variables which have positive and finite second moment. Also note that the scale correction needed to have updates of the appropriate magnitude corresponds exactly the using negative powers of m which are equal to $\gamma_l(p)$ for $p = 1$ which would correspond to 1-homogeneity.

Proof. This results from the scale analysis of the forward and backward passes above and from the formulas for the updates $\Delta W^1(1) = -\eta \chi_0 dh_0^1 \xi_0^\top$, $\Delta W^l(1) = -m^{-1} \eta \chi_0 dh_0^l (x_0^{l-1})^\top$, $\Delta W^{L+1}(1) = -\eta \chi_0 m^{-1} x_0^{L-1}$. Noticing that one has $\rho^{L-1} = \rho^{(L-l+1)} \rho^{(l-1-1)}$, as well as $m^{-(L+1)/2} = m^{-1} m^{-(L-1)/2}$ and finally $m^{-(L+2)/2} = m^{-1} m^{-(l-1)/2} m^{-(L-l+1)/2}$ allows to conclude. □

A.6.5 . Final comments on Assumption 4

With the lemmas presented above, it is clear that Assumption 4 induce a behaviour in the first update step that is similar to homogeneity with $p = 1$ which bears a couple of differences with Assumption 3. First , $\sigma'(0) \neq 0$ which means we cannot amplify the signal of the forward pass in $\sigma'(h_0^l)$ and the latter simply converges to $\sigma'(0)$. Because $\sigma'(h_0^l)$ appears in the first weight updates, and that for μP it does not converge to 0, one cannot obtain exact equivalence between IP-LLR and μP without a higher degree of homogeneity. However, after the first weight update, IP-LLR with σ satisfying Assumption 4 has the same updates as μP, but with a different initialization.

A.7 . Preliminaries for Theorem 2.3.2 and Theorem 2.4.1

In all this section since we assume positive homogeneity of the activation function, we also consider parameterizations with no bias terms except at the first layer.

A.7.1 . Tilde variables

Definition A.7.1 (Scaleless variables at initialization)**.** Let $\xi \in \mathbb{R}^d$ be an input vector. Independently of any parameterization, we consider the following variables "without scale" at initialization :

$$\begin{cases} \tilde{h}_0^1(\xi) := U^1\xi + v^1 \\ \tilde{x}_0^1(\xi) := \sigma(\tilde{h}_0^1(\xi)) \end{cases} \qquad \forall l \in [2, L], \begin{cases} \tilde{h}_0^l(\xi) := \widehat{W}^l \tilde{x}_0^{l-1}(\xi) \\ \tilde{x}_0^l(\xi) := \sigma(\tilde{h}_0^l(\xi)) \end{cases}$$

and define $\tilde{f}_0(\xi) := (\widehat{W}^{L+1})^\top \tilde{x}_0^L$, as well as

$$\begin{cases} d\tilde{x}_0^L(\xi) := U^{L+1} \\ d\tilde{h}_0^L(\xi) := d\tilde{x}_0^L(\xi) \odot \sigma'(\tilde{h}_0^L(\xi)) \end{cases} \quad \forall l \in [L-1], \begin{cases} d\tilde{x}_0^l(\xi) := (\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}(\xi) \\ d\tilde{h}_0^l(\xi) := d\tilde{x}_0^l(\xi) \odot \sigma'(\tilde{h}_0^l(\xi)) \end{cases}$$

where the $\widehat{W}^l$ are defined in Equation (A.1).

Remark. The tilde variables are independent of the choice of parameterization because, independently of the parameterization, $\widehat{W}_{pq}^l = m^{-1/2} U_{pq}^l \sim \mathcal{N}(0, 1/m)$ for $l \in [2, L+1]$ and $\widehat{W}_{pq}^1 = U_{pq}^1 \sim \mathcal{N}(0, 1)$. Those variables essentially reproduce the computations that take place in the forward (without any bias terms except at the first layer) and backward passes of any acparameterization but the magnitudes (the multiplying scalars ω_l) have been set to 1, essentially removing the additional scales which lead to explosion or vanishing as $m \to \infty$. The tilde variables of the forward pass at initialization correspond to the NTK parameterization. However this is not the case for the backward pass as the backward pass of NTK vanishes at initialization whereas the corresponding tilde variables have positive (> 0) variance as shown in Lemma A.7.1 below.

Lemma A.7.1 (Scaleless variables have positive and finite second moment)**.** *Let $\xi \in \mathbb{R}^d$ be an input vector, and consider a non-linearity σ satisfying Assumption 2. Then, dropping the dependency of the tilde variables on ξ, one has that for any $l \in [1, L]$, and for any $z \in \{\tilde{h}_0^l, \tilde{x}_0^l, d\tilde{h}_0^l, d\tilde{x}_0^l\}$, the second moment is positive*

and finite: $0 < \mathbb{E}[(Z^z)^2] < \infty$*. More precisely, one has:*

$$
\begin{aligned}
&Z^{\tilde{h}_0^1} \sim \mathcal{N}(0, ||\xi||^2{+}1), && 0 < \mathbb{E}[(Z^{\tilde{x}_0^1})^2] < \infty && \\
&Z^{\tilde{h}_0^l} \sim \mathcal{N}(0, V_{h,l}^2), && 0 < V_{h,l}^2 := \mathbb{E}[(Z^{\tilde{x}_0^{l-1}})^2] < \infty, && l \in [2, L], \\
&0 < \mathbb{E}[(Z^{\tilde{x}_0^l})^2] < \infty, && && l \in [2, L], \\
&\tilde{f}_0(\xi) \xrightarrow[m\to\infty]{law} \mathcal{N}(0, V_f^2), && 0 < V_f^2 := \mathbb{E}[(Z^{\tilde{x}_0^L})^2] < \infty, && \\
&Z^{d\tilde{x}_0^L} \sim \mathcal{N}(0, 1) && && \\
&Z^{d\tilde{x}_0^l} \sim \mathcal{N}(0, V_{dx,l}^2), && 0 < V_{dx,l}^2 := \mathbb{E}[(Z^{d\tilde{h}_0^{l+1}})^2] < \infty, && l \in [1, L-1], \\
&0 < \mathbb{E}[(Z^{d\tilde{h}_0^l})^2] < \infty, && && l \in [1, L].
\end{aligned}
$$

Remark. As shown in Appendix A.14, those expectations, as well as the means (first and second moment) are tractable with $\sigma = \text{ReLU}$ and have simple expressions (for the first forward and backward passes). As shown in Appendices A.14.3 and A.14.5, the recursive formulas for the variances of the forward and backward passes can be unrolled, and to avoid explosion or vanishing with the depth L, one must initialize the i.i.d. Gaussian entries with a standard deviation of $\sqrt{2}$ to preserve the norm of the input signal.

Proof. Let $\xi \in \mathbb{R}^d$ be an input vector. We omit the dependency of the forward and backward passes on ξ for simplicity. We first induct from $l = 1$ to $l = L$ for the forward pass and then from $l = L$ to $l = 1$ for the backward pass. $\tilde{h}_0^1 = U^1\xi + v^1$ is the sum of two initial vectors in the program, which follows two independent Gaussian laws by definition: $Z^{U^1\xi} \sim \mathcal{N}(0, ||\xi||^2)$, and $Z^{v^1} \sim \mathcal{N}(0, 1)$ independently of $Z^{U^1\xi}$. We thus have $Z^{\tilde{h}_0^1} \sim \mathcal{N}(0, ||\xi||^2{+}1)$, which shows its variance is finite and > 0, and by Lemma A.3.1, $0 < \mathbb{E}[(Z^{\tilde{x}_0^1})^2] < \infty$ since $Z^{\tilde{x}_0^1} = \sigma(Z^{\tilde{h}_0^1})$.

Now let $l \in [1, L-1]$ and assume $Z^{\tilde{h}_0^l} \sim \mathcal{N}(0, V_{h,l}^2)$ with $0 < V_{h,l}^2 < \infty$, and $0 < \mathbb{E}[(Z^{\tilde{x}_0^l})^2] < \infty$. By `ZMatMul`, $Z^{\tilde{h}_0^{l+1}} = Z^{\widehat{W}^{l+1}x_0^l}$ which is equal to $\widehat{Z}^{\widehat{W}^{l+1}x_0^l}$ by Lemma A.3.2. now by definition, $\widehat{Z}^{\widehat{W}^{l+1}\tilde{x}_0^l} \sim \mathcal{N}(0, \mathbb{E}[(Z^{\tilde{x}_0^l})^2])$, and the variance is > 0 and finite by the induction hypothesis, so that $0 < \mathbb{E}[(Z^{\tilde{h}_0^{l+1}})^2] < \infty$. Now by Lemma A.3.1 again, since $Z^{\tilde{x}_0^{l+1}} = \sigma(Z^{\tilde{h}_0^{l+1}})$, we also get that $0 < \mathbb{E}[(Z^{\tilde{x}_0^{l+1}})^2] < \infty$ which concludes the induction for the first L layers of the forward pass.

$\tilde{f}_0(\xi) = (\widehat{W}^{l+1})^\top \tilde{x}_0^L$ and $\widehat{W}_j^{L+1} \sim \mathcal{N}(0, 1/m)$ for every m, and by the Master Theorem, since $||\tilde{x}_0^L||^2/m$ is a scalar in the program defined by the moment operation, it converges almost surely to $\mathbb{E}[(Z^{\tilde{x}_0^L})^2]$. Finally, since $\tilde{x}_0^L$ is computed using only the $\widehat{W}^l$ for $l \leq L$, $\tilde{x}_0^L$ is independent of $\widehat{W}^{l+1}$. By Lemma A.3.3, $\tilde{f}_0(\xi)$ converges in law towards $\mathcal{N}(0, \mathbb{E}[(Z^{\tilde{x}_0^L})^2])$, and it holds that $0 < \mathbb{E}[(Z^{\tilde{x}_0^L})^2] < \infty$ by the previous induction.

$Z^{d\tilde{x}_0^L} = Z^{U^{L+1}}$ and since U^{L+1} is an initial vector in the program whose coordinates are iid following $\mathcal{N}(0,1)$, we have by definition $Z^{U^{L+1}} \sim \mathcal{N}(0,1)$. $Z^{d\tilde{h}_0^L} = Z^{d\tilde{x}_0^L \sigma'(Z^{\tilde{h}_0^L})} = \widehat{Z}^{U^{l+1}} \sigma'(\widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}})$. Now by definition in `ZHat`, $\widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}$ is independent of $\widehat{Z}^{U^{l+1}}$ since U^{L+1} is an initial vector in the program. This yields

$$\begin{aligned}\mathbb{E}[(Z^{d\tilde{h}_0^L})^2] &= \mathbb{E}\left[(\widehat{Z}^{U^{L+1}})^2\right] \mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^L})^2\right] \\ &= 1 \times \mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^L})^2\right].\end{aligned}$$

By assumption, σ' is pseudo-Lipschitz and thus polynomially bounded, and is not almost everywhere 0. By the induction above, $Z^{\tilde{h}_0^L} \sim \mathcal{N}(0, \mathbb{E}[(Z^{\tilde{x}_0^{L-1}})^2])$ with $0 < \mathbb{E}[(Z^{\tilde{x}_0^{L-1}})^2] < \infty$. By Lemma A.3.1 we thus have $0 < \mathbb{E}[\sigma'(Z^{\tilde{h}_0^L})^2] < \infty$, which shows $0 < \mathbb{E}[(Z^{d\tilde{h}_0^L})^2] < \infty$.

Now let $l \in [2, L]$ and assume $Z^{d\tilde{x}_0^l} \sim \mathcal{N}(0, V_{dx,l}^2)$ with $0 < V_{dx,l}^2 < \infty$, and assume $0 < \mathbb{E}[(Z^{d\tilde{h}_0^l})^2] < \infty$. $Z^{d\tilde{x}_0^{l-1}} = Z^{(\widehat{W}^l)^\top d\tilde{h}_0^l}$ and $Z^{(\widehat{W}^l)^\top d\tilde{h}_0^l} = \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}$ by Lemma A.3.2. By definition, $\widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l} \sim \mathcal{N}(0, \mathbb{E}[(Z^{d\tilde{h}_0^l})^2])$, so that $\mathbb{E}[(Z^{d\tilde{x}_0^{l-1}})^2] = \mathbb{E}[(Z^{d\tilde{h}_0^l})^2]$ and thus $0 < \mathbb{E}[(Z^{d\tilde{x}_0^{l-1}})^2] < \infty$ by the induction hypothesis. We have

$$Z^{d\tilde{h}_0^{l-1}} = Z^{d\tilde{x}_0^{l-1}} \sigma'(Z^{\tilde{h}_0^{l-1}}) = \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l} \sigma'(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}})$$

if $l \geq 3$, and

$$Z^{d\tilde{h}_0^1} = Z^{d\tilde{x}_0^1} \sigma'(Z^{\tilde{h}_0^1}) = \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2} \sigma'(\widehat{Z}^{\widehat{W}^1 \xi_0 + v^1})$$

if $l = 2$. In any case, the random variable inside σ' is independent of the other variable in the product. We thus get

$$\mathbb{E}[(Z^{d\tilde{h}_0^{l-1}})^2] = \underbrace{\mathbb{E}\left[(Z^{d\tilde{x}_0^{l-1}})^2\right]}_{>0, <\infty} \underbrace{\mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^{l-1}})^2\right]}_{>0, <\infty}$$

where the bounds on the second expectation are obtained using Lemma A.3.1. This concludes the induction for the backward pass and thus the proof. □

A.7.2 . Expression of the forward and backward passes of ac-parameterizations in function of the tilde variables with homogeneity

Lemma A.7.2 (Forward pass with homogeneity at $t = 0$)**.** *Consider any ac-parameterization of an L-hidden layer neural network with a p-homogeneous activation function, and $p \geq 1$. Let $\xi \in \mathbb{R}^d$ be an input to the network. Then, omitting the dependency of the forward pass and the tilde variables on ξ, one has:*

$$h_0^l = \gamma_{f,l} \tilde{h}_0^l, \qquad l \in [1, L], \tag{A.12}$$

$$x_0^l = (\gamma_{f,l})^p \tilde{x}_0^l, \qquad l \in [1, L], \tag{A.13}$$

$$f_0(\xi) = \gamma_{f,L+1} \tilde{f}_0(\xi), \tag{A.14}$$

where, for any $l \in [1, L+1]$

$$\gamma_{f,l} := \left(\prod_{k=1}^{l} \omega_k^{p^{l-k}} \right).$$

Remark.

1. $(\gamma_{f,l})^p = \left(\prod_{k=1}^{l} \omega_k^{p^{l-k+1}} \right)$.
2. When $p = 1$, $\gamma_{f,l}$ and $(\gamma_{f,l})^p$ simply reduce to $\omega_l \dots \omega_1$.
3. For integrable parameterizations, for any $l \in [1, L+1]$, $\gamma_{f,l} = m^{-\sum_{k=0}^{l-2} p^k/2}$. The latter term is 1 when $l = 1$, and otherwise $m^{-(l-1)/2}$ if $p = 1$ and $m^{-(p^{l-1}-1)/2(p-1)}$ if $p > 1$. For μP, $\gamma_{f,l} = 1$ for any $l \in [1, L]$ because $\omega_l = 1$ for μP if $l \in [1, L]$.
4. Instead of homogeneity, assume σ is differentiable, has non-zero derivative in 0 and $\sigma(0) = 0$. Also assume that $\omega_1 = 1$ (*i.e.*, $a_1 = 0$) and $\omega_l \to 0$ (*i.e.*, $a_l > 1/2$) for $l \in [2, L]$, which is the case in integrable parameterizations. Then, we have $h_0^1 = \tilde{h}_0^1$, and $h_0^2 = \omega_2 \tilde{h}_0^2$, so that $x_0^2 = \sigma(\omega_2 \tilde{h}_0^2)$ and as $m \to \infty$, $x_0^2 \simeq \omega_2 \sigma'(0) \tilde{h}_0^2$. Then similarly, we have for $h_0^3 \simeq \omega_3 \omega_2 \sigma'(0) \widehat{W}^3 \tilde{h}_0^2$ and $x_0^3 \simeq \omega_3 \omega_2 \sigma'(0)^2 \widehat{W}^3 \tilde{h}_0^2$. An easy induction then gives $h_0^l = \sigma'(0)^{l-2} (\omega_l \dots \omega_2) \widehat{W}^l \dots \widehat{W}^2 \tilde{h}_0^2$. This thus resembles the case of a $p = 1$ positively homogeneous function, except that the first forward pass is effectively linearized after layer 1, but the magnitude of the forward pass at different layers is also well understood in this case so that the learning rates for the first update can be chosen appropriately (*e.g.*, for integrable parameterizations). In particular, the initial learning rates of IP-LLR for $p = 1$ will also produce non-trivial weight updates at $t = 0$ in this setting, which will in turn induce learning. Finally, setting the initial standard deviations of the weight matrices equal to $|\sigma'(0)|^{-1}$ instead of 1 for the intermediate layers avoids problems with the depth L.

Proof. $h_0^1 = m^{-a_1}$ implies that $h_0^1 = \omega_1 (U^1 \xi + v^1) = \omega_1 \tilde{h}_0^1$, which entails $x_0^1 = \omega_1^p \tilde{x}_0^1$ because σ is positively p-homogeneous and $\omega_1 \geq 0$. Now let $l \in [1, L-1]$ and assume $h_0^l = (\prod_{k=1}^{l} \omega_k^{p^{l-k}}) \tilde{h}_0^l$, and $x_0^l = (\prod_{k=1}^{l} \omega_k^{p^{l-k+1}}) \tilde{x}_0^l$. Then

$$\begin{aligned} h_0^{l+1} &= \omega_{l+1} \widehat{W}^{l+1}(0) x_0^l \\ &= \omega_{l+1} \left(\prod_{k=1}^{l} \omega_k^{p^{l-k+1}} \right) \widehat{W}^{l+1}(0) \tilde{x}_0^l \\ &= \left(\prod_{k=1}^{l+1} \omega_k^{p^{l+1-k}} \right) \tilde{h}_0^{l+1} \end{aligned}$$

Since σ is positively homogeneous, we have

$$\begin{aligned} x_0^{l+1} &= \sigma(h_0^{l+1}) \\ &= \left(\prod_{k=1}^{l+1} \omega_k^{p^{l+1-k}}\right)^p \sigma(\tilde{h}_0^{l+1}) \\ &= \left(\prod_{k=1}^{l+1} \omega_k^{p^{l+2-k}}\right) \sigma(\tilde{h}_0^{l+1}) \end{aligned}$$

This concludes the induction and gives the result for any $l \in [1, L]$. To conclude the proof we compute the expression of $f_0(\xi)$:

$$\begin{aligned} f_0(\xi) = \omega_{L+1}(\widehat{W}^{L+1}(0))^\top x_0^L = \omega_{L+1}\left(\prod_{k=1}^{L} \omega_k^{p^{L-k+1}}\right)(\widehat{W}^{L+1}(0))^\top \tilde{x}_0^L \\ = \left(\prod_{k=1}^{L+1} \omega_k^{p^{L+1-k}}\right) \tilde{f}_0(\xi). \end{aligned}$$

□

Lemma A.7.3 (Backward pass with homogeneity at $t = 0$)**.** *Consider any acparameterization of an L-hidden layer neural network with a positively p- homogeneous activation function, and $p \geq 1$. Let $\xi_0 \in \mathbb{R}^d$ be the first training input. Then, omitting the dependency of the forward and backward passes, as well as that of the tilde variables on ξ_0, one has for any $l \in [1, L]$:*

$$dx_0^l = m^{-a_{L+1}} \gamma_{b,l} \left(\prod_{k=l+1}^{L} \gamma_{f,k}\right)^{p-1} d\tilde{x}_0^l, \tag{A.15}$$

$$dh_0^l = m^{-a_{L+1}} \gamma_{b,l} \left(\prod_{k=l}^{L} \gamma_{f,k}\right)^{p-1} d\tilde{h}_0^l, \tag{A.16}$$

where, for any $l \in [1, L]$,

$$\gamma_{b,l} = \prod_{k=l+1}^{L} \omega_k.$$

Remark.

1. By swapping the products, one has that

$$\prod_{k=l+1}^{L} \gamma_{f,k} = \prod_{k=1}^{L} \omega_k^{\sum_{r=\max(k,l+1)}^{L} p^{r-k}}.$$

2. When $p = 1$, $\left(\prod_{k=l}^{L} \gamma_{f,k}\right)^{p-1} = 1$ for any $l \in [1, L+1]$.

3. For integrable parameterizations, $\gamma_{b,l} = m^{-(L-l)/2}$ for any $l \in [1, L]$. For μP, $\gamma_{b,l} = 1$ for any $l \in [1, L]$.

4. For $l = L$, $\gamma_{b,L} = 1$, $\prod_{k=l+1}^{L} \gamma_{f,k} = 1$, $\prod_{k=l}^{L} \gamma_{f,k} = \gamma_{f,L}$.

Proof. $dx_0^L = W^L(0) = m^{-a_{L+1}} U^{L+1} = m^{-a_{L+1}} d\tilde{x}_0^L$,

$$\begin{aligned} dh_0^L &= dx_0^L \odot \sigma'(h_0^L) \\ &= m^{-a_{L+1}} d\tilde{x}_0^L \odot \sigma'(\gamma_{f,L} \tilde{h}_0^L) \\ &= m^{-a_{L+1}} (\gamma_{f,L})^{p-1} d\tilde{x}_0^L \odot \sigma'(\tilde{h}_0^L) \end{aligned}$$

where the second equality stems from Lemma A.7.2 and the last equality stems from $\omega_L \dots \omega_1 > 0$ and the positive $(p-1)$-homogeneity of σ'. Let $l \in [2, L]$ and assume that dx_0^l satisfies Equation (A.15) and dh_0^l satisfies Equation (A.16). Then

$$\begin{aligned} dx_0^{l-1} &= \omega_l (\widehat{W}^l)^\top dh_0^l \\ &= m^{-a_{L+1}} \omega_l \gamma_{b,l} \left(\prod_{k=l}^{L} \gamma_{f,k} \right)^{p-1} (\widehat{W}^l)^\top d\tilde{h}_0^l \\ &= m^{-a_{L+1}} \gamma_{b,l-1} \left(\prod_{k=(l-1)+1}^{L} \gamma_{f,k} \right)^{p-1} d\tilde{x}_0^{l-1}, \end{aligned}$$

and

$$\begin{aligned} dh_0^{l-1} &= dx_0^l \odot \sigma'(h_0^l) \\ &= m^{-a_{L+1}} \gamma_{b,l-1} \left(\prod_{k=l}^{L} \gamma_{f,k} \right)^{p-1} d\tilde{x}_0^{l-1} \odot \sigma'(\gamma_{f,l} \tilde{h}_0^{l-1}) \\ &= m^{-a_{L+1}} \gamma_{b,l-1} \left(\prod_{k=l}^{L} \gamma_{f,k} \right)^{p-1} (\gamma_{f,l-1})^{p-1} d\tilde{x}_0^l \odot \sigma'(\tilde{h}_0^l) \\ &= m^{-a_{L+1}} \gamma_{b,l-1} \left(\prod_{k=l-1}^{L} \gamma_{f,k} \right)^{p-1} d\tilde{h}_0^{l-1}, \end{aligned}$$

where we have used Lemma A.7.2 in the second equality, the positive $(p-1)$-homogeneity of σ' combined with $\omega_l \dots \omega_1 > 0$ in the third equality and the definition of $d\tilde{h}_0^{l-1}$ in the last. This thus concludes the proof by induction. □

Lemma A.7.4 (Weight updates with homogeneity at $t = 0$)**.** *Consider any ac-parameterization of an L-hidden layer neural network with a positively p- homogeneous activation function, and $p \geq 1$. Let $\xi_0 \in \mathbb{R}^d$ be the first training input.*

Then, omitting the dependency of the forward and backward passes, as well as that of the tilde variables on ξ_0, one has:

$$\Delta W^1(1) = -\eta\chi_0 m^{-(a_{L+1}+2a_1+c_1)}\omega_1^{p^L-1}\left(\prod_{k=2}^{L}\omega_k^{p^{L-k+1}}\right)d\tilde{h}_0^1\xi_0^\top,$$

$$\Delta B^1(1) = -\eta\chi_0 m^{-(a_{L+1}+2a_1+c_1)}\omega_1^{p^L-1}\left(\prod_{k=2}^{L}\omega_k^{p^{L-k+1}}\right)d\tilde{h}_0^1,$$

$$\Delta W^l(1) = -\eta\chi_0 m^{-(a_{L+1}+2a_l+c_l-1)}\left(\prod_{k=1}^{L}\omega_k^{p^{L-k+1}}\right)\omega_l^{-1}\frac{d\tilde{h}_0^l(\tilde{x}_0^{l-1})^\top}{m}, \qquad l\in[2,L],$$

$$\Delta W^{L+1}(1) = -\eta\chi_0 m^{-(2a_{L+1}+c_{L+1}-1)}\left(\prod_{k=1}^{L}\omega_k^{p^{L-k+1}}\right)\tilde{x}_0^L/m.$$

Remark. For $p=1$, we have

$$\omega_1^{p^L-1}\left(\prod_{k=2}^{L}\omega_k^{p^{L-k+1}}\right) = \omega_1\dots\omega_L$$

$$\left(\prod_{k=1}^{L}\omega_k^{p^{L-k+1}}\right)\omega_l^{-1} = \omega_1\dots\omega_{l-1}\omega_{l+1}\dots\omega_L$$

$$\prod_{k=1}^{L}\omega_k^{p^{L-k+1}} = \omega_1\dots\omega_L$$

Proof. Before we begin with the proof, we start with a first basic result which will be used repeatedly in the proof. Let $N\in\mathbb{N}^*$. By Equation (A.3), we have

$$(p-1)\sum_{r=0}^{N}p^r = \sum_{r=1}^{N+1}p^r - \sum_{r=0}^{N}p^r = p^{N+1}-1.$$

Now that this is established, let us look at the update for the first layer. We have

$$\begin{aligned}\Delta W^1(1) &= -\eta m^{-(2a_1+c_1)}\chi_0 dh_0^1\xi_0^\top\\ &= -\eta m^{-(2a_1+c_1+a_{L+1})}\gamma_{b,1}\left(\prod_{k=1}^{L}\gamma_{f,k}\right)^{p-1}\chi_0 d\tilde{h}_0^1\xi_0^\top,\end{aligned}$$

where we have used Lemmas A.7.2 and A.7.3 in the second equality. Now, we have

$$\gamma_{b,1} = \prod_{k=2}^{L}\omega_k,$$

and by the first point in Remark A.7.2, we have (with $l = 1$)

$$\begin{aligned}\left(\prod_{k=1}^{L}\gamma_{f,k}\right)^{p-1} &= \prod_{k=1}^{L}\omega_k^{(p-1)\sum_{r=k}^{L}p^{r-k}}\\ &= \prod_{k=1}^{L}\omega_k^{(p-1)\sum_{r=0}^{L-k}p^{r}}\\ &= \prod_{k=1}^{L}\omega_k^{p^{L-k+1}-1}.\end{aligned}$$

It follows that

$$\gamma_{b,1}\left(\prod_{k=1}^{L}\gamma_{f,k}\right)^{p-1} = \omega_1^{p^L-1}\prod_{k=2}^{L}\omega_k^{p^{L-k+1}}$$

The formula for $\Delta B^1(1)$ follows from the expression of dh_0^1 in function of $d\tilde{h}_0^1$ and from Equation (A.4).

Let $l \in [2, L]$

$$\begin{aligned}\Delta W^l(1) &= -\eta m^{-(2a_l+c_l)}\chi_0 dh_0^l (x_0^{l-1})^\top\\ &= -\eta m^{-(2a_l+c_l+a_{L+1}-1)}\chi_0\,\gamma_{b,l}\,(\gamma_{f,l-1})^p\left(\prod_{k=l}^{L}\gamma_{f,k}\right)^{p-1}\frac{d\tilde{h}_0^l(\tilde{x}_0^{l-1})^\top}{m}\end{aligned}$$

Now, we have

$$\gamma_{b,l} = \prod_{k=l+1}^{L}\omega_k.$$

In addition, by the first point of Remark A.7.2, we have

$$(\gamma_{f,l-1})^p = \left(\prod_{k=1}^{l-1}\omega_k^{p^{l-k}}\right),$$

and by the first point in Remark A.7.2

$$\begin{aligned}\left(\prod_{k=l}^{L}\gamma_{f,k}\right)^{p-1} &= \left(\prod_{k=1}^{l-1}\omega_k^{(p-1)\sum_{r=l}^{L}p^{r-k}}\right)\times\left(\prod_{k=l}^{L}\omega_k^{(p-1)\sum_{r=k}^{L}p^{r-k}}\right)\\ &= \left(\prod_{k=1}^{l-1}\omega_k^{(p-1)p^{l-k}\sum_{r=l}^{L}p^{r-l}}\right)\times\left(\prod_{k=l}^{L}\omega_k^{(p-1)\sum_{r=0}^{L-k}p^{r}}\right)\\ &= \left(\prod_{k=1}^{l-1}\omega_k^{(p-1)p^{l-k}\sum_{r=0}^{L-l}p^{r}}\right)\times\left(\prod_{k=l}^{L}\omega_k^{(p-1)\sum_{r=0}^{L-k}p^{r}}\right).\end{aligned}$$

Let us now look, for each $k \in [1, L]$, at the power of ω_k which appears in the product $\gamma_{b,l}(\gamma_{f,l-1})^p \left(\prod_{k=l}^{L} \gamma_{f,k}\right)^{p-1}$. If $k \in [1, l-1]$, the exponent for ω_k is equal to

$$\begin{aligned} p^{l-k} + (p-1)p^{l-k} \sum_{r=0}^{L-l} p^r &= p^{l-k} \left((p-1) \sum_{r=0}^{L-l} p^r + 1 \right) \\ &= p^{l-k} \left(p^{L-l+1} - 1 + 1 \right) \\ &= p^{L-k+1}. \end{aligned}$$

If $k = l$, the exponent for ω_l is equal to

$$(p-1) \sum_{r=0}^{L-l} p^r = p^{L-l+1} - 1.$$

If $k \in [l+1, L]$, the exponent for ω_k is equal to

$$1 + (p-1) \sum_{r=0}^{L-k} p^r = 1 + p^{L-k+1} - 1 = p^{L-k+1}.$$

Thus, for every $k \neq l$, the exponent for ω_k is equal to p^{L-k+1}, and for $k = l$, the exponent for ω_l is equal to $p^{L-l+1} - 1$. It follows that

$$\gamma_{b,l}(\gamma_{f,l-1})^p \left(\prod_{k=l}^{L} \gamma_{f,k} \right)^{p-1} = \left(\prod_{k=1}^{L} \omega_k^{p^{L-k+1}} \right) w_l^{-1}.$$

Finally,

$$\begin{aligned} \Delta W^{L+1}(1) &= -\eta m^{-(2a_{L+1}+c_{L+1})} \chi_0 x_0^L \\ &= -\eta m^{-(2a_{L+1}+c_{L+1}-1)} \chi_0 (\gamma_{f,L})^p \tilde{x}_0^L / m, \end{aligned}$$

where we have used Lemma A.7.2 in the second equality. From the first point of Remark A.7.2, we get that

$$(\gamma_{f,L})^p = \prod_{k=1}^{L} \omega_k^{p^L - k + 1},$$

which concludes the proof.

□

Corollary A.7.4.1 (Weight updates of IP with homogeneity at $t = 0$)**.** *Consider an integrable parameterization of an L-hidden layer neural network with no bias terms except at the first layer, and a positively p-homogeneous activation function, and $p \geq 1$. Let $\xi_0 \in \mathbb{R}^d$ be the first training input. Then, omitting the dependency*

of the forward and backward passes, as well as that of the tilde variables on ξ_0, one has:

$$\begin{aligned}
\Delta W^1(1) &= -\eta\chi_0 m^{-(c_1-\gamma_1(p))} d\tilde{h}_0^1 \xi_0^\top, \\
\Delta B^1(1) &= -\eta\chi_0 m^{-(c_1-\gamma_1(p))} d\tilde{h}_0^1, \\
\Delta W^l(1) &= -\eta\chi_0 m^{-(c_l-\gamma_l(p))} \frac{d\tilde{h}_0^l (\tilde{x}_0^{l-1})^\top}{m}, \qquad l \in [2, L], \\
\Delta W^{L+1}(1) &= -\eta\chi_0 m^{-(c_{L+1}-\gamma_{L+1}(p))} \tilde{x}_0^L / m,
\end{aligned}$$

where the $\gamma_l(p)$ are given in Definition 2.3.2.

Proof. For integrable parameterizations, $\omega_1 = 1$, $\omega_l = m^{-1/2}$ for $l \in [2, L]$, and $a_{L+1} = 1$. For the first layer, we have $a_{L+1} + 2a_1 + c_1 = 1$. On the other hand,

$$\begin{aligned}
\omega_1^{p^L-1} \left(\prod_{k=2}^{L} \omega_k^{p^{L-k+1}} \right) &= \prod_{k=2}^{L} m^{-p^{L-k+1}/2} \\
&= m^{-\sum_{k=2}^{L} p^{L-k+1}/2} \\
&= m^{-\sum_{k=1}^{L-1} p^k/2} \\
&= m^{-1/2(\sum_{k=0}^{L-1} p^k - 1)},
\end{aligned}$$

so that

$$\begin{aligned}
m^{-(a_{L+1}+2a_1+c_1)} \omega_1^{p^L-1} \left(\prod_{k=2}^{L} \omega_k^{p^{L-k+1}} \right) &= m^{-c_1} m^{-1/2(\sum_{k=0}^{L-1} p^k - 1)} m^{-1} \\
&= m^{-c_1} m^{-1/2(\sum_{k=0}^{L-1} p^k + 1)} \\
&= m^{-c_1} m^{\gamma_1(p)},
\end{aligned}$$

by Definition 2.3.2, which gives the result for the first layer's update ($\Delta W^1(1)$ and $\Delta B^1(1)$). Let $l \in [2, L]$. $a_{L+1} + 2a_l - 1 = 1 + 2 - 1 = 2$. On the other hand,

$$\begin{aligned}
\left(\prod_{k=1}^{L} \omega_k^{p^{L-k+1}} \right) \omega_l^{-1} &= m^{-1/2(\sum_{k=0}^{L-1} p^k - 1)} m^{1/2} \\
&= m^{-1/2 \sum_{k=0}^{L-1} p^k + 1},
\end{aligned}$$

so that

$$\begin{aligned}
m^{-(a_{L+1}+2a_l+c_l)} \left(\prod_{k=1}^{L} \omega_k^{p^{L-k+1}} \right) \omega_l^{-1} &= m^{-c_l} m^{-1/2 \sum_{k=0}^{L-1} p^k + 1} m^{-2} \\
&= m^{-c_l} m^{-1/2 \sum_{k=0}^{L-1} p^k - 1} \\
&= m^{-c_l} m^{\gamma_l(p)},
\end{aligned}$$

by Definition 2.3.2, which proves the result for the updates of the intermediate layers. Finally, we have $2a_{L+1} - 1 = 2 - 1 = 1$, and on the other hand, because $\omega_1 = 1$, as in the first update, we find

$$\prod_{k=1}^{L} \omega_k^{p^{L-k+1}} = m^{-1/2(\sum_{k=0}^{L-1} p^k - 1)},$$

so that

$$\begin{aligned} m^{-(2a_{L+1}+c_{L+1}-1)} \prod_{k=1}^{L} \omega_k^{p^{L-k+1}} &= m^{-c_{L+1}} m^{-1/2(\sum_{k=0}^{L-1} p^k - 1)} m^{-1} \\ &= m^{-c_{L+1}} m^{-1/2(\sum_{k=0}^{L-1} p^k + 1)} \\ &= m^{-c_{L+1}} m^{\gamma_{L+1}(p)}, \end{aligned}$$

by Definition 2.3.2, which gives the result for the last layer's update and therefore concludes the proof. □

Corollary A.7.4.2 (Weight updates of IP-LLR at $t = 0$)**.** *Consider an IP-LLR parameterization of an L-hidden layer neural network with a p-homogeneous activation function, and $p \geq 1$. Let $\xi_0 \in \mathbb{R}^d$ be the first training input. Then, omitting the dependency of the forward and backward passes of IP-LLR, as well as that of the tilde variables on ξ_0, one has:*

$$\begin{aligned} \Delta W^1(1) &= -\eta \chi_0 d\tilde{h}_0^1 \xi_0^\top, \\ \Delta B^1(1) &= -\eta \chi_0 d\tilde{h}_0^1, \\ \Delta W^l(1) &= -\eta \chi_0 \frac{d\tilde{h}_0^l (\tilde{x}_0^{l-1})^\top}{m}, \quad l \in [2, L], \\ \Delta W^{L+1}(1) &= -\eta \chi_0 \tilde{x}_0^L / m. \end{aligned}$$

Proof. This is a simple consequence of Corollary A.7.4.1 and the fact that for IP-LLR $c_l = \gamma_l(p)$ at $t = 0$ by definition (see Definition 2.4.1) for any $l \in [1, L + 1]$. □

Lemma A.7.5 (Weight updates of μP at $t = 0$)**.** *Consider the μP parameterization given in Definition A.2.3 with a differentiable activation function σ. Let $\xi_0 \in \mathbb{R}^d$ be the first training input. Then, omitting the dependency of the forward and backward passes of μP, as well as that of the tilde variables on ξ_0, one has:*

$$\begin{aligned} \Delta W^1(1) &= -\eta \chi_0 d\tilde{h}_0^1 \xi_0^\top \\ \Delta B^1(1) &= -\eta \chi_0 d\tilde{h}_0^1 \\ \Delta W^l(1) &= -\eta \chi_0 \frac{d\tilde{h}_0^l (\tilde{x}_0^{l-1})^\top}{m}, \quad l \in [2, L] \\ \Delta W^{L+1}(1) &= -\eta \chi_0 \tilde{x}_0^L / m \end{aligned}$$

Remark.

1. Although the formulas are identical with those for IP-LLR when the activation function is positively p-homogeneous, this **does not** mean that the weight updates are exactly equal. Indeed, although the tilde variables do not depend on the choice of parameterization and will thus be the same in μP as in IP-LLR, the variable χ_0 which appears in the formulas is parameterization-dependent as it depends on $f_0(\xi)$ which itself depends on the choice of parameterization.

2. There is no strong assumption on the activation function here (*e.g.*, homogeneity) as μP is designed to have such updates which induce feature learning at all layers.

3. Note that the coordinates of $\Delta W^l(1)$ are in $\Theta(m^{-1})$ whereas that of $W^l(0)$ are in $\Theta(m^{-1/2})$ for $l \in [2, L]$, so that paradoxically, even though μP is designed to produce "maximal updates" (in a certain sense), we have that $\Delta W^l_{jq}(1)/W^l_{jq}(0) = \Theta(m^{-1/2}) \to 0$ as $m \to \infty$: the relative displacement of the weights is zero in the infinite-width limit. More generally, we have that for μP $(W^l_{jq}(t) - W^l_{jq}(0))/W^l_{jq}(0) \to 0$ as $m \to \infty$ if $t \geq 1$, which means that weights of the intermediate layers do not move away from their initialization in the infinite-width limit for μP, even if the (pre-)activations of every layer are maximally updated. This is in stark contrast with IP-LLR for which both $W^l(0)$ and $\Delta W^l(1)$ are in $\Theta(m^{-1})$ for the intermediate layers $l \in [2, L]$: the weights do move relatively to their initialization in the infinite-width limit.

Proof. μP is designed so that its forward pass has $h^l_0 = \tilde{h}^l_0$ for any $l \in [1, L]$. Indeed, the choice of pre-factors for the weights with μP lead to the same recursive equations for the forward pass as the tilde variables, except for $f_0(\xi)$ which is equal to $m^{-1/2}\tilde{f}_0(\xi)$. For the backward pass, one has that for μP, $dx^L_0 = W^{L+1}(0) = m^{-1}U^{L+1} = m^{-1}d\tilde{x}^L_0$. We then have

$$\begin{aligned} dh^L_0 &= dx^L_0 \odot \sigma'(h^L_0) \\ &= m^{-1} d\tilde{x}^L_0 \odot \sigma'(\tilde{h}^L_0) \\ &= m^{-1} d\tilde{h}^L_0. \end{aligned}$$

Let $l \in [1, L-1]$, and assume that $dx^{l+1}_0 = m^{-1}d\tilde{x}^{l+1}_0$ and $dh^{l+1}_0 = m^{-1}d\tilde{h}^{l+1}_0$. Then, we have

$$\begin{aligned} dx^l_0 &= (W^{l+1}(0))^\top dh^{l+1}_0 \\ &= m^{-1}(\widehat{W}^{l+1})^\top d\tilde{h}^{l+1}_0 \\ &= m^{-1} d\tilde{x}^l_0. \end{aligned}$$

Similarly, we have

$$\begin{aligned} dh_0^l &= dx_0^l \odot \sigma'(h_0^l) \\ &= m^{-1} d\tilde{x}_0^l \odot \sigma'(\tilde{h}_0^l) \\ &= m^{-1} d\tilde{h}_0^l, \end{aligned}$$

which proves by induction that for any $l \in [1, L]$, $dx_0^l = m^{-1} d\tilde{x}_0^l$ and $dh_0^l = m^{-1} d\tilde{h}_0^l$ for μP. Recall that for μP, $a_1 = 0$, $a_l = 1/2$ for $l \in [2, L]$ and $a_{L+1} = 1$, and $c_l = -1$ for any $l \in [1, L+1]$. Now by Equations (A.3) and (A.4), the first weight updates give:

$$\begin{aligned} \Delta W^1(1) &= -\eta \chi_0 m^{-c_1} m^{-1} d\tilde{h}_0^1 \xi_0^\top \\ &= -\eta \chi_0 d\tilde{h}_0^1 \xi_0^\top, \end{aligned}$$

and

$$\begin{aligned} \Delta B^1(1) &= -\eta \chi_0 m^{-c_1} m^{-1} d\tilde{h}_0^1 \\ &= -\eta \chi_0 d\tilde{h}_0^1, \end{aligned}$$

For $l \in [2, L]$, we have

$$\begin{aligned} \Delta W^l(1) &= -\eta \chi_0 m^{-(1+c_l)} m^{-1} d\tilde{h}_0^l (x_0^{l-1})^\top \\ &= -\eta \chi_0 \frac{d\tilde{h}_0^l (\tilde{x}_0^{l-1})^\top}{m}. \end{aligned}$$

Finally,

$$\begin{aligned} \Delta W^{L+1}(1) &= -\eta \chi_0 m^{-(2+c_{L+1})} x_0^L \\ &= -\eta \chi_0 \tilde{x}_0^L / m, \end{aligned}$$

which concludes the proof. □

A.8 . Dynamics of the infinite-width limit of IP-LLR

Lemma A.8.1 (IP-LLR is zero at initialization)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then, for any input vector $\xi \in \mathbb{R}^d$, one has that $\tilde{h}_0^l(\xi), \tilde{x}_0^l(\xi), d\tilde{x}_0^l, d\tilde{h}_0^l$ are vectors in the Tensor Program program for any $l \in [2, L]$, and additionally:*

$$\begin{aligned} f_0(\xi) &\xrightarrow[m\to\infty]{a.s.} 0 \\ \chi_0 &\xrightarrow[m\to\infty]{a.s.} \mathring{\chi}_0 := \partial_2 \ell(y_0, 0) \end{aligned}$$

Remark. The result on the almost sure convergence of χ_0 ensures that the latter is a valid initial scalar in the Tensor Program defining the computations associated with the IP-LLR parameterization.

Proof. Because σ and σ' are pseudo-Lipschitz (since $p \geq 2$, see Appendix A.5), the tilde variables of the first forward and backward passes $(\tilde{h}_0^l, \tilde{x}_0^l, d\tilde{x}_0^l, d\tilde{x}_1^l)$ are vectors in the program given Definition A.7.1 by the `ZNonLin` and `ZMatMul` rules. Additionally, by Lemma A.7.2,

$$\begin{aligned} f_0(\xi_0) &= m^{-\sum_{k=0}^{L-1} p^k/2} m^{-1/2} (U^{L+1})^\top \tilde{x}_0^l \\ &= m^{1/2} m^{-\sum_{k=0}^{L-1} p^k/2} m^{-1} (U^{L+1})^\top \tilde{x}_0^L \\ &= m^{-\sum_{k=1}^{L-1} p^k/2} m^{-1} (U^{L+1})^\top \tilde{x}_0^L \end{aligned}$$

Now, $m^{-1}(U^{L+1})^\top \tilde{x}_0^L \to \mathbb{E}[Z^{U^{L+1}} Z^{\tilde{x}_0^L}]$ almost surely by the master theorem, and $Z^{\tilde{x}_0^L} = \sigma(Z^{\widehat{W}^L \tilde{x}_0^{L-1}})$. By the Lemma A.3.2, $Z^{\widehat{W}^L \tilde{h}_0^{L-1}} = \widehat{Z}^{\widehat{W}^L \tilde{h}_0^{L-1}}$, and by the `ZHat` rule, the latter variable is independent of $Z^{U^{L+1}}$ since $Z^{U^{L+1}}$ is an initial vector in the program. This gives $\mathbb{E}[Z^{U^{L+1}} Z^{\tilde{x}_0^L}] = \mathbb{E}[Z^{U^{L+1}}]\mathbb{E}[Z^{\tilde{x}_0^L}] = 0 \times \mathbb{E}[Z^{\tilde{x}_0^L}]$. By Lemma A.7.1, and Lemma A.3.1, $\mathbb{E}[Z^{\tilde{x}_0^L}] < \infty$ because σ is polynomially bounded. We thus get $\mathbb{E}[Z^{U^{L+1}} Z^{\tilde{x}_0^L}] = 0$, and since $m^{-\sum_{k=1}^{L-1} p^k/2} \in (0, 1]$, $f_0(\xi_0) \to 0$ almost surely. Recall that by definition (see Appendix A.1) $\chi_0 = \partial_2 \ell(y_0, f_0(\xi_0))$. Since $f_0(\xi_0) \to 0$ almost surely, and since $\partial_2 \ell(y_0, \cdot)$ is continuous by assumption, we have that $\chi_0 \to \partial_2 \ell(y_0, 0) =: \mathring{\chi}_0$, which concludes the proof. □

Definition A.8.1 (Tilde variables in the backward pass after initialization)**.** For any ac-parameterization with $a_{L+1} = 1$, define for any $t \geq 1$,

$$\begin{aligned} &d\tilde{x}_t^L = m dx_t^L,\ d\tilde{h}_t^L = d\tilde{x}_t^L \odot \sigma'(h_t^L), \\ &d\tilde{x}_t^l = (W^{l+1}(t))^\top d\tilde{h}_t^{l+1},\ \ l \in [1, L-1], \\ &d\tilde{h}_t^l = d\tilde{x}_t^l \odot \sigma'(h_t^l),\ \ l \in [1, L-1]. \end{aligned}$$

Remark.

1. One could in general define $d\tilde{x}_t^l$ to be equal to $m^{a_{L+1}} dx_t^l$ but since all the ac-parameterizations we study in this paper, *i.e.*, integrable parameterizations, μP, or hybrid versions thereof have $a_{L+1} = 1$, we limit the formulas to this case. The tilde variables are the right quantity to look at because of the term $m^{-a_{L+1}}$ which appears in the gradient *w.r.t.*to x_t^L and then propagate to all the other variables of the backward pass by the equations of backpropagation.

2. Recall that in the definition above, it is implicitly assumed that the computations of the forward and backward passes at any time step s are done with the input $\xi = \xi_s$.

Lemma A.8.2 (Relationship between tilde and non-tilde variables)**.** *For any ac-parameterization with $a_{L+1} = 1$, for any $t \geq 1$, and for any ξ, dropping the dependency of the forward and backward passes on ξ at time t, one has:*

$$\forall l \in [1, L+1],\ dx_t^l = m^{-1} d\tilde{x}_t^l,\ dh_t^l = m^{-1} d\tilde{h}_t^l.$$

Proof. $dx_t^L = m^{-1} d\tilde{x}_t^L$. $dh_t^L = dx_t^L \odot \sigma'(h_t^L) = m^{-1} d\tilde{x}_t^L \odot \sigma'(h_t^L) = m^{-1} d\tilde{h}_t^L$. Now let $l \in [2, L]$ and assume $dx_t^l = m^{-1} d\tilde{x}_t^l$, $dh_t^l = m^{-1} d\tilde{h}_t^l$. Then $dx_t^{l-1} = (W^l(t))^\top dh_t^l = m^{-1}(W^l(t))^\top d\tilde{h}_t^l = m^{-1} d\tilde{x}_t^{l-1}$, and $dh_t^{l-1} = dx_t^{l-1}\sigma(h_t^{l-1}) = m^{-1} d\tilde{x}_t^{l-1}\sigma(h_t^{l-1}) = m^{-1} d\tilde{h}_t^{l-1}$ which concludes the proof by induction. □

Lemma A.8.3 (Weight updates for IP-LLR at any time step)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 1$, and let $t \geq 1$. Then, dropping the dependency of the forward and backward passes on ξ_t at time t, one has:*

$$\begin{aligned}
\Delta W^{L+1}(t+1) &= -\eta \chi_t x_t^L / m, \\
\Delta W^l(t+1) &= -\eta \chi_t \frac{d\tilde{h}_t^l (x_t^{l-1})^\top}{m}, \qquad l \in [2, L], \\
\Delta W^1(t+1) &= -\eta \chi_t d\tilde{h}_t^1 \xi_t^\top, \\
\Delta B^1(t+1) &= -\eta \chi_t d\tilde{h}_t^1.
\end{aligned}$$

Proof. Using Equation (A.3), we have $\Delta W^{L+1}(t) = -\eta \chi_t m^{-(2a_{L+1}+c_{L+1})} x_t^L = -\eta \chi_t x_t^L / m$ because $2a_{L+1} + c_{L+1} = 2 - 1 = 1$ in IP-LLR since $t \geq 1$. For $l \in [2, L]$

$$\begin{aligned}
\Delta W^l(t) &= -\eta \chi_t m^{-(2a_l + c_l)} dh_t^l (x^{l-1})^\top \\
&= -\eta \chi_t \frac{d\tilde{h}_t^l (x_t^{l-1})^\top}{m},
\end{aligned}$$

by Lemma A.8.2 and because $2a_l + c_l = 2 - 2 = 0$ for $t \geq 1$ in IP-LLR. $\Delta W^1(t) = -\eta \chi_t m^{-(2a_1+c_1)} dh_t^1 \xi_t = -\eta \chi_t d\tilde{h}_t^1 \xi_t^\top$ by Lemma A.8.2 and because $2a_1 + c_1 = 0 - 1 = -1$ for $t \geq 1$ in IP-LLR. Finally, by Equation (A.4), we have $\Delta B^1(t) = -\eta \chi_t m^{-(2a_1+c_1)} dh_t^1 = -\eta \chi_t d\tilde{h}_t^1$ by Lemma A.8.2 and because $2a_1 + c_1 = -1$. □

Theorem A.8.4 (Weights in IP-LLR at time t)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 1$. Then, for any $t \geq 1$, one has:*

(i) $W^1(t) = U^1 - \eta \chi_0 d\tilde{h}_0^1 \xi_0^\top - \eta \left(\sum_{s=1}^{t-1} \chi_s d\tilde{h}_s^1 \xi_s^\top \right)$,

(ii) $B^1(t) = v^1 - \eta \chi_0 d\tilde{h}_0^1 - \eta \left(\sum_{s=1}^{t-1} \chi_s d\tilde{h}_s^1 \right)$,

(iii) $W^l(t) = \omega_l \widehat{W}^l - \eta \chi_0 \frac{d\tilde{h}_0^l (\tilde{x}_0^{l-1})^\top}{m} - \eta \left(\sum_{s=1}^{t-1} \chi_s \frac{d\tilde{h}_s^l (x_s^{l-1})^\top}{m} \right), \qquad l \in [2, L]$,

(iv) $W^{L+1}(t) = U^{L+1}/m - \eta \chi_0 \tilde{x}_0^L / m - \eta \left(\sum_{s=1}^{t-1} \chi_s x_s^L / m \right)$.

Proof. We have already seen the formulas are correct for $t = 1$ by Corollary A.7.4.2. Then, by Lemma A.8.3, an easy induction immediately yields the result. □

Lemma A.8.5 (Backward pass of IP-LLR at time t)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 1$. Then, for any $t \geq 1$, dropping the dependency of the forward pass at time t on ξ_t, and of the previous forward and backward passes on the corresponding ξ_s, one has:*

(i) $d\tilde{x}_t^L = w^{L+1}(t) = U^{L+1} - \eta\chi_0\tilde{x}_0^L - \eta\sum_{s=1}^{t-1}\chi_s x_s^L$,

(ii) $d\tilde{x}_t^{l-1} = \omega_l(\widehat{W}^l)^\top d\tilde{h}_t^l - \eta\chi_0\frac{(d\tilde{h}_0^l)^\top d\tilde{h}_t^l}{m}\tilde{x}_0^{l-1} - \eta\sum_{s=1}^{t-1}\chi_s\frac{(d\tilde{h}_s^l)^\top d\tilde{h}_t^l}{m}x_s^{l-1}, \qquad l \in [2, L]$.

Proof. By definition, we have

$$\begin{aligned} d\tilde{x}_t^L &= m dx_t^L \\ &= mW^{L+1}(t) \\ &= U^{L+1} - \eta\chi_0\tilde{x}_0^L - \eta\sum_{s=1}^{t-1}\chi_s x_s^L \end{aligned}$$

where the last equality stems from Theorem A.8.4.

Let $l \in [2, L]$, we have:

$$\begin{aligned} d\tilde{x}_t^{l-1} &= (W^l(t))^\top d\tilde{h}_t^l \\ &= \omega_l(\widehat{W}^l)^\top d\tilde{h}_t^l - \eta\chi_0\frac{(d\tilde{h}_0^l)^\top d\tilde{h}_t^l}{m}\tilde{x}_0^{l-1} - \eta\sum_{s=1}^{t-1}\chi_s\frac{(d\tilde{h}_s^l)^\top d\tilde{h}_t^l}{m}x_s^{l-1} \end{aligned}$$

where the second equality stems from Theorem A.8.4. □

Lemma A.8.6 (Z for the forward pass of IP-LLR at time $t = 1$)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Let $\xi \in \mathbb{R}^d$ be an input to the network. Then, for any $l \in [1, L]$, $h_1^l(\xi), x_1^l(\xi), d\tilde{x}_1^l, d\tilde{h}_1^l$ are vectors in the program, $f_1(\xi)$ is a scalar in the program, and χ_1 is a valid initial scalar in the program. Additionally, dropping the dependency of the forward pass at time $t = 1$ on ξ, and of the first forward and backward passes on ξ_0, one has:*

(i) $Z^{h_1^1} = Z^{W^1(1)\xi + B^1(1)} = Z^{U^1\xi} + Z^{v^1} - \eta\mathring{\chi}_0(\xi_0^\top\xi + 1)Z^{d\tilde{h}_0^1}$,

(ii) $Z^{h_1^l} = Z^{W^l(1)x_1^{l-1}} = \mathring{\omega}_l Z^{\widehat{W}^l x_1^{l-1}} - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^{l-1}}Z^{x_1^{l-1}}]Z^{d\tilde{h}_0^l}, \qquad l \in [2, L]$,

(iii) $f_1(\xi) = (W^{L+1}(1))^\top x_1^L \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}}Z^{x_1^L}] - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^L}Z^{x_1^L}]$.

Proof. By Theorem A.8.4, with $t = 1$, one has that $h_1^1 = U^1\xi + v^1 - \eta\chi_0(\xi_0^\top\xi + 1)d\tilde{h}_0^1$. By Lemma A.8.1, $d\tilde{h}_0^1$ is a vector in the Tensor Program and χ_0 is a valid initial scalar in the program which has an almost sure limit $\mathring{\chi}_0 := \partial_2\ell(y_0, 0)$ as $m \to \infty$. In addition, $U^1\xi$ and v^1 are initial vectors in the program, which thus shows that h_1^1 is a vector in the program by the `NonLin` operation. This also

gives that $x_1^1 = \sigma(h_1^1)$ is a vector in the program since σ is pseudo-Lipschitz (see Appendix A.5). Moreover, by `ZNonLin`, we have $Z^{h_1^1} = Z^{U^1\xi} + Z^{v^1} - \mathring{\eta\chi}_0(\xi_0^\top \xi + 1)Z^{d\tilde{h}_0^1}$. Let $l \in [2, L]$ and assume that h_1^{l-1}, x_1^{l-1} are vectors in the program. Then, by Theorem A.8.4 with $t = 1$, we get

$$h_1^l = \omega_l \widehat{W}^l x_1^{l-1} - \eta\chi_0 \frac{(\tilde{x}_0^{l-1})^\top x_1^{l-1}}{m} d\tilde{h}_0^l.$$

$(\tilde{x}_0^{l-1})^\top x_1^{l-1}/m$ is a scalar in the program by the `Moment` operation, and thus by the `MatMul` and `NonLin` operations, h_1^l is a vector in the program and thus so is $x_1^l = \sigma(h_1^l)$, which proves by induction that this is the case for any $l \in [2, L]$. By `ZNonLin` we thus have

$$Z^{h_1^l} = \mathring{\omega}_l Z^{\widehat{W}^l x_1^{l-1}} - \mathring{\eta\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}] Z^{d\tilde{h}_0^l}.$$

We then have by Theorem A.8.4 with $t = 1$,

$$f_1(\xi) = m^{-1}(U^{L+1})^\top x_1^L - \eta\chi_0 \frac{(\tilde{x}_0^L)^\top x_1^L}{m}$$

$U^{L+1} - \eta\chi_0 \tilde{x}_0^L$ is a vector in the program by the `NonLin` operation, and the quantity $m^{-1}(U^{L+1} - \eta\chi_0\tilde{x}_0^L)^\top x_1^L$ is thus a scalar in the program by the `Moment` operation, and by the master theorem, we get $f_1(\xi) \to \mathbb{E}[Z^{U^{L+1}} Z^{x_1^L}] - \mathring{\eta\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_1^L}]$ almost surely, since both expectations are finite by Lemma A.13.1. Since we did the previous reasoning with an arbitrary ξ, we also get that $h_1^l(\xi_1), x_1^l(\xi_1)$ are vectors in the program for any $l \in [1, L]$ and that the formulas in (i), (ii), and (iii) hold when the input is ξ_1. In particular, $f_1(\xi_1)$ converges to a finite almost sure limit $\mathring{f}_1(\xi_1)$, and thus the continuity of $\partial_2 \ell(y_1, \cdot)$ ensures the almost sure convergence of χ_1 towards $\mathring{\chi}_1 := \partial_2 \ell(y_1, \mathring{f}_1(\xi_1))$, which means χ_1 is a valid initial scalar in the Tensor Program. Then, dropping the dependency of the second forward pass (at $t = 1$) on ξ_1, we get by Lemma A.8.5 with $t = 1$:

$$d\tilde{x}_1^L = U^{L+1} - \eta\chi_0 \tilde{x}_0^L$$

which is a vector in the program by `NonLin`. Then $d\tilde{h}_1^L = d\tilde{x}_1^L \odot \sigma'(h_1^L)$ is also a vector in the program since σ' is pseudo-Lipschitz. Let $l \in [2, L-1]$ and assume that $d\tilde{x}_1^{l+1}$ and $d\tilde{h}_1^{l+1}$ are vectors in the program. Then by Lemma A.8.5 with $t = 1$, we have

$$d\tilde{x}_1^l = \omega_{l+1}(\widehat{W}^{l+1})^\top d\tilde{h}_1^{l+1} - \eta\chi_0 \frac{(d\tilde{h}_0^{l+1})^\top d\tilde{h}_1^{l+1}}{m} \tilde{x}_0^l$$

$(d\tilde{h}_0^{l+1})^\top d\tilde{h}_1^{l+1}/m$ is a scalar in the program by the `Moment` operation and by `MatMul` and `NonLin` we thus get that $d\tilde{x}_1^l$ is a vector in the program. Then $d\tilde{h}_1^l = d\tilde{x}_1^l \odot \sigma'(h_1^l)$ is also a vector in the program since σ' is pseudo-Lipschitz, which concludes the induction and with it the proof. □

Theorem A.8.7 (Z for the forward pass of IP-LLR at time t)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Let $\xi \in \mathbb{R}^d$ be an input to the network. Then, for any $l \in [1, L]$, $h_s^l(\xi), x_s^l(\xi), d\tilde{x}_s^l, d\tilde{h}_s^l$ are vectors in the program, $f_s(\xi)$ is a scalar in the program, and χ_s is a valid initial scalar in the program. Additionally, dropping the dependency of the forward pass at time t on ξ, and of the previous forward and backward passes on the corresponding ξ_s, one has:*

(i) $Z^{h_t^1} = Z^{W^1(t)\xi + B^1(t)} = Z^{U^1\xi} + Z^{v^1} - \eta \mathring{\chi}_0 (\xi_0^\top \xi + 1) Z^{d\tilde{h}_0^1} - \eta \left(\sum_{s=1}^{t-1} \mathring{\chi}_s (\xi_s^\top \xi + 1) Z^{d\tilde{h}_s^1} \right)$,

(ii) for any $l \in [2, L]$,

$$Z^{h_t^l} = Z^{W^l(t) x_t^{l-1}} = \mathring{\omega}_l Z^{\widehat{W}^l x_t^{l-1}} - \eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_t^{l-1}}] Z^{d\tilde{h}_0^l} - \eta \left(\sum_{s=1}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^{l-1}} Z^{x_t^{l-1}}] Z^{d\tilde{h}_s^l} \right),$$

(iii) $f_t(\xi) = (W^{L+1}(t))^\top x_t^L \xrightarrow[m \to \infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x_t^L}] - \eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_t^L}] - \eta \left(\sum_{s=1}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^L} Z^{x_t^L}] \right)$.

Proof. We prove that the vectors and scalars in the claim of the theorem are part of the program by induction. Then the formulas of (i), (ii), and (iii) are a simple consequence of the `ZNonLin` operation. The case $t = 1$ has been treated in Lemma A.8.6. Let $t \geq 1$ and assume that the vectors and scalars in the claim of the theorem are part of the program for any $s \in [1, t]$. By Theorem A.8.4, one has that

$$\begin{aligned} h_{t+1}^1 &= W^1(t+1)\xi + B^1(t+1) \\ &= U^1\xi + v^1 - \eta \chi_0 (\xi_0^\top \xi + 1) d\tilde{h}_0^1 - \eta \left(\sum_{s=1}^{t} \chi_s (\xi_s^\top \xi + 1) d\tilde{h}_s^1 \right) \end{aligned}$$

By the induction hypothesis and `NonLin`, we thus get that h_{t+1}^1 is a vector in the program and thus so is $x_{t+1}^1 = \sigma(h_{t+1}^1)$ since σ is polynomially bounded. Let $l \in [2, L]$ and assume that $h_{t+1}^{l-1}, x_{t+1}^{l-1}$ are vectors in the program. Then, by Theorem A.8.4, we get

$$h_{t+1}^l = \omega_l \widehat{W}^l x_{t+1}^{l-1} - \eta \chi_0 \frac{(\tilde{x}_0^{l-1})^\top x_{t+1}^{l-1}}{m} d\tilde{h}_0^l - \eta \left(\sum_{s=1}^{t} \chi_s \frac{(x_s^{l-1})^\top x_{t+1}^{l-1}}{m} d\tilde{h}_s^l \right).$$

For any $s \in [1, t]$, $(x_s^{l-1})^\top x_1^{l-1}/m$ and $(\tilde{x}_0^{l-1})^\top x_1^{l-1}/m$ are scalars in the program by the induction hypothesis and the `Moment` operation. Thus by the `MatMul` and `NonLin` operations, h_{t+1}^l is a vector in the program and thus so is $x_{t+1}^l = \sigma(h_{t+1}^l)$, which proves by induction that this is the case for any $l \in [2, L]$. We then have by Theorem A.8.4,

$$f_{t+1}(\xi) = m^{-1} \left(U^{L+1} - \eta \chi_0 \tilde{x}_0^L - \eta \sum_{s=1}^{t} \chi_s x_s^L \right)^\top x_{t+1}^L$$

$U^{L+1} - \eta\chi_0\tilde{x}_0^L - \eta\sum_{s=1}^t \chi_s x_s^L$ is a vector in the program by the induction hypothesis and the `NonLin` operation. Then, by the `Moment` operation, $f_{t+1}(\xi)$ is a scalar in the program since x_{t+1}^L is also a vector in the program, and by the master theorem, we have

$$f_{t+1}(\xi) \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x_{t+1}^L}] - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_{t+1}^L}] - \eta\sum_{s=1}^t \mathring{\chi}_s\mathbb{E}[Z^{x_s^L} Z^{x_{t+1}^L}].$$

The limit is finite by Lemma A.13.1 since by an easy induction any Z which appears is a polynomially bounded function of a Gaussian vector with finite covariance matrix. Since we did the previous reasoning with an arbitrary ξ, we also get that $h_{t+1}^l(\xi_{t+1}), x_{t+1}^l(\xi_{t+1})$ are vectors in the program for any $l \in [1, L]$. In particular, $f_{t+1}(\xi_{t+1})$ converges to an almost sure limit $\mathring{f}_{t+1}(\xi_{t+1})$, and thus the continuity of $\partial_2\ell(y_{t+1}, \cdot)$ ensures the almost sure convergence of χ_{t+1} towards $\mathring{\chi}_{t+1} := \partial_2\ell(y_{t+1}, \mathring{f}_{t+1}(\xi_{t+1}))$, which means χ_{t+1} is a valid initial scalar in the Tensor Program. Then, dropping the dependency of the forward pass at $t+1$ on ξ_{t+1}, we get by Lemma A.8.5:

$$d\tilde{x}_{t+1}^L = U^{L+1} - \eta\chi_0\tilde{x}_0^L - \eta\sum_{s=1}^t \chi_s x_s^L$$

which is a vector in the program by `NonLin`. Then $d\tilde{h}_{t+1}^L = d\tilde{x}_{t+1}^L \odot \sigma'(h_{t+1}^L)$ is also a vector in the program since σ' is pseudo-Lipschitz. Let $l \in [2, L-1]$ and assume that $d\tilde{x}_{t+1}^{l+1}$ and $d\tilde{h}_{t+1}^{l+1}$ are vectors in the program. Then by Lemma A.8.5, we have

$$d\tilde{x}_{t+1}^l = \omega_{l+1}(\widehat{W}^{l+1})^\top d\tilde{h}_{t+1}^{l+1} - \eta\chi_0\frac{(d\tilde{h}_0^{l+1})^\top d\tilde{h}_{t+1}^{l+1}}{m}\tilde{x}_0^l - \eta\sum_{s=1}^t \chi_s\frac{(d\tilde{h}_s^{l+1})^\top d\tilde{h}_{t+1}^{l+1}}{m}x_s^l$$

$(d\tilde{h}_s^{l+1})^\top d\tilde{h}_{t+1}^{l+1}/m$ is a scalar in the program for any $s \in [0, t]$ by the `Moment` operation and by `MatMul` and `NonLin` we thus get that $d\tilde{x}_{t+1}^l$ is a vector in the program. Then $d\tilde{h}_{t+1}^l = d\tilde{x}_{t+1}^l \odot \sigma'(h_{t+1}^l)$ is also a vector in the program since σ' is pseudo-Lipschitz, which concludes the induction. Then we get the claims of (i), (ii) and (iii) simply by applying the `ZNonLin` rule to the formulas derived above for the pre-activations h_{t+1}^l. □

Corollary A.8.7.1 (Z for the forward pass of IP-LLR at time t)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then, for any $t \geq 1$, and for any input $\xi \in \mathbb{R}^d$, dropping the dependency of the forward pass at time t on ξ, and of the previous forward and backward passes on the corresponding ξ_s, one has:*

(i) $Z^{h_t^1} = Z^{W^1(t)\xi + B^1(t)} = Z^{U^1\xi} + Z^{v^1} - \eta\mathring{\chi}_0(\xi_0^\top\xi + 1)Z^{d\tilde{h}_0^1} - \eta\left(\sum_{s=1}^{t-1}\mathring{\chi}_s(\xi_s^\top\xi + 1)Z^{d\tilde{h}_s^1}\right.$

(ii) for any $l \in [2, L]$,
$Z^{h_t^l} = Z^{W^l(t)x_t^{l-1}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_t^{l-1}}] Z^{d\tilde{h}_0^l} - \eta \left(\sum_{s=1}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^{l-1}} Z^{x_t^{l-1}}] Z^{d\tilde{h}_s^l} \right)$,

(iii) $f_t(\xi) = (W^{L+1}(t))^\top x_t^L \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x_t^L}] - \eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_t^L}] - \eta \left(\sum_{s=1}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^L} Z^{x_t^L}] \right)$.

Proof. The formulas are readily obtained by Theorem A.8.7 coupled with the fact that we have $\mathring{\omega}_l Z^{\widehat{W}^l x_t^{l-1}} = 0$ for any $l \in [2, L]$, and $t \geq 1$, which stems from Theorem A.13.9. □

Remark. Note that there is no circular logic here since only Theorem A.8.7 is used to prove the results of Appendix A.13.1 (and in particular Theorem A.13.9), so that using Theorem A.13.9 for Corollary A.8.7.1 does not lead to any issue.

Theorem A.8.8 (Zs of backward pass of IP-LLR at time t)**.** *Consider the IP-LLR parameterization with a positively* p*-homogeneous activation function, and* $p \geq 2$*. Then, for any* $t \geq 1$*, dropping the dependency of the forward pass at time* t *on* ξ_t*, and of the previous forward and backward passes on the corresponding* ξ_s*, one has:*

(i) $Z^{d\tilde{x}_t^L} = Z^{w^{L+1}(t)} = Z^{U^{L+1}} - \eta \mathring{\chi}_0 Z^{\tilde{x}_0^L} - \eta \sum_{s=1}^{t-1} \mathring{\chi}_s Z^{x_s^L}$,

(ii) $Z^{d\tilde{x}_t^{l-1}} = \mathring{\omega}_l Z^{(\widehat{W}^l)^\top d\tilde{h}_t^l} - \eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_t^l}] Z^{\tilde{x}_0^{l-1}} - \eta \sum_{s=1}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{d\tilde{h}_s^l} Z^{d\tilde{h}_t^l}] Z^{x_s^{l-1}}$, $l \in [2, L]$.

Proof. We have already proved in Theorem A.8.7 that for any $s \in [1, t]$ the vectors of the forward (h_s^l, x_s^l for $l \in [1, L]$) and the backward pass ($d\tilde{x}_s^l, d\tilde{h}_s^l$ for $l \in [1, L]$) at time s are part of the program and similarly at $t = 0$ by Lemma A.8.1. Then, claims (i) and (ii) readily follow from applying the `ZNonLin` rule to the formulas of Lemma A.8.5. □

Corollary A.8.8.1 (Zs of backward pass of IP-LLR at time t)**.** *Consider the IP-LLR parameterization with a positively* p*-homogeneous activation function, and* $p \geq 2$*. Then, for any* $t \geq 1$*, dropping the dependency of the forward pass at time* t *on* ξ_t*, and of the previous forward and backward passes on the corresponding* ξ_s*, one has:*

(i) $Z^{d\tilde{x}_t^L} = Z^{w^{L+1}(t)} = Z^{U^{L+1}} - \eta \mathring{\chi}_0 Z^{\tilde{x}_0^L} - \eta \sum_{s=1}^{t-1} \mathring{\chi}_s Z^{x_s^L}$

(ii) $Z^{d\tilde{x}_t^{l-1}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_t^l}] Z^{\tilde{x}_0^{l-1}} - \eta \sum_{s=1}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{d\tilde{h}_s^l} Z^{d\tilde{h}_t^l}] Z^{x_s^{l-1}}$, $l \in [2, L]$.

Proof. The formulas are readily obtained by Theorem A.8.8 and the fact that $Z^{(\widehat{W}^l)^\top d\tilde{h}_t^l} = 0$ for any $l \in [2, L]$ and $t \geq 1$, which stems from Theorem A.13.9. □

Remark. Note that a similar statement can be made as in Remark A.8 regarding circular logic since only Theorem A.8.8 is used to prove the results of Appendix A.13.1.

A.8.1 . Second forward pass of IP-LLR ($t = 1$)

In this section, we prove that for IP-LLR, we have $0 < \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_1^l}] < \infty$ for any $l \in [1, L]$ under the assumption that $\mathring{\chi}_0 := \lim_{m\to\infty} \chi_0 \neq 0$. To obtain those results, we use the formulas from Corollary A.8.7.1 for $t = 1$, which are obtained using the main result from Appendix A.13, namely Theorem A.13.9. We choose to put Appendix A.13 towards the end of the Appendix section as its main result is quite intuitive: any multiplication by matrices with pre-factors in m^{-1} result in a vector whose coordinates (the corresponding Z) converge to 0 almost surely at any time step. The proof however requires a long and cumbersome induction and we thus leave it for the later stages of the Appendix so as not to break the narrative of the Appendix.

The finiteness of the expectations $\mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_1^l}]$ is a simple consequence of Lemma A.13.1, but the fact that they are > 0 requires more work as we will see below. Since we work with IP-LLR, recall that we consider a bias term at the first layer only.

Lemma A.8.9 (1st layer of forward pass of IP-LLR at $t = 1$)**.** *Consider the IP-LLR parameterization with an activation function σ satisfying Assumption 3. Let ξ be an input to the network, and assume $\mathring{\chi}_0 \neq 0$. Then, dropping the dependency of the first forward-backward pass on ξ_0, and that of the second forward pass on ξ, one has:*

(i) $Z^{h_1^1} = Z^{\tilde{h}_0^1(\xi)} - \eta\mathring{\chi}_0(\xi_0^\top \xi + 1) Z^{d\tilde{h}_0^1} = Z^{\tilde{h}_0^1(\xi)} - \eta\mathring{\chi}_0(\xi_0^\top \xi + 1) Z^{d\tilde{x}_0^1}\sigma'(Z^{\tilde{h}_0^1})$,

(ii) $(Z^{\tilde{h}_0^1}, Z^{\tilde{h}_0^1(\xi)}, Z^{d\tilde{x}_0^1}) \sim \mathcal{N}\left(0, \begin{pmatrix} ||\xi_0||^2+1 & \xi_0^\top \xi + 1 & 0 \\ \xi^\top \xi_0 + 1 & ||\xi||^2+1 & 0 \\ 0 & 0 & \mathbb{E}[(Z^{d\tilde{h}_0^2})^2] \end{pmatrix}\right)$,

(iii) $0 < \mathbb{E}[Z^{\tilde{x}_0^1} Z^{x_1^1}] < \infty$.

Proof. We have by Corollary A.8.7.1 at time $t = 1$

$$\begin{aligned} Z^{x_1^1} &= \sigma\left(Z^{h_1^1(\xi)}\right) \\ &= \sigma\left(Z^{\tilde{h}_0^1(\xi)} - \eta\mathring{\chi}_0(\xi_0^\top \xi + 1) Z^{d\tilde{h}_0^1}\right). \end{aligned}$$

Moreover, since $d\tilde{h}_0^1 = d\tilde{x}_0^1 \odot \sigma'(\tilde{h}_0^1)$, since all the vectors are part of the Tensor Program, by `ZNonLin` we have $Z^{d\tilde{h}_0^1} = Z^{d\tilde{x}_0^1}\sigma'(Z^{\tilde{h}_0^1})$, so that

$$Z^{x_1^1} = \sigma\left(Z^{\tilde{h}_0^1(\xi)} - \eta\mathring{\chi}_0(\xi_0^\top \xi + 1) Z^{d\tilde{x}_0^1}\sigma'(Z^{\tilde{h}_0^1})\right).$$

Finally, we have

$$Z^{\tilde{x}_0^1} = \sigma(Z^{\tilde{h}_0^1}).$$

From the rules of `ZInit` and `ZHat`, we have that

$$(Z^{\tilde{h}^1_0}, Z^{\tilde{h}^1_0(\xi)}, Z^{d\tilde{x}^1_0}) \sim \mathcal{N}\left(0, \begin{pmatrix} S & 0 \\ 0 & \mathbb{E}[(Z^{d\tilde{h}^2_0})^2] \end{pmatrix}\right),$$

with

$$S := \begin{pmatrix} ||\xi_0||^2+1 & \xi_0^\top \xi + 1 \\ \xi_0^\top \xi + 1 & ||\xi||^2+1 \end{pmatrix}.$$

By Lemma A.7.1, $\mathbb{E}[(Z^{d\tilde{h}^2_0})^2] < \infty$, so that the covariance matrix is finite and thus $Z^{\tilde{x}^1_0} Z^{x^1_1}$ is a polynomially bounded function of a Gaussian vector which shows that the expectation is finite by Lemma A.13.1. It is also non-negative since σ is non-negative. To prove that it is positive, one needs only prove that the integrand is not almost everywhere 0. By Lemma A.7.1, $\mathbb{E}[(Z^{d\tilde{h}^2_0})^2] > 0$ so that the covariance matrix is invertible if and only if S is invertible. We have

$$\det(S) = \left(||\xi_0||^2||\xi||^2 - (\xi_0^\top \xi)^2\right) + ||\xi_0 - \xi||^2,$$

which is the sum of two non-negative terms by Cauchy-Schwarz's inequality, and is thus 0 if and only if both terms are zero. The first term is zero only when ξ and ξ_0 are proportional, and if in addition the second term is zero than $\xi = \xi_0$. The distribution of the Gaussian vector appearing in $Z^{\tilde{x}^1_0} Z^{x^1_1}$ thus depends on whether or not ξ_0 and ξ are equal.

Case when $\xi = \xi_0$. Then, calling $\lambda := -\eta \mathring{\chi}_0(\xi_0^\top \xi + 1)$, we have

$$\mathbb{E}[Z^{\tilde{x}^1_0} Z^{x^1_1}] = \int \sigma(z)\sigma\left(z - \lambda u \sigma'(z)\right) p_z(z) p_u(u) \mathrm{d}z \mathrm{d}u,$$

where p_z and p_u are the densities of the two Gaussians $\mathcal{N}(0, ||\xi_0||^2+1)$ and $\mathcal{N}(0, \mathbb{E}[(Z^{d\tilde{h}^2_0})^2])$ respectively, which are not degenerate, so that $p_z(z) > 0$ for any z and similarly for $p_u(u)$. Since $Z^{d\tilde{x}^1_0}$ and $-Z^{d\tilde{x}^1_0}$ have the same distribution and since it is independent of $Z^{\tilde{h}^1_0}$, we can assume $\lambda \geq 0$ W.L.O.G (if $\lambda \leq 0$ we can always do the change of variable $u \leftarrow -u$ in the integral above since $p_u(-u) = p_u(u)$). Consider the point $(z^*, u^*) := (1, -1)$, at which the integrand in the integral above is > 0, because σ and σ' are > 0 on the positive part of the real line (see Appendix A.5) and $\lambda \geq 0$. The integral is then positive, because the integrand is a continuous function, since σ and σ' are continuous (see again Appendix A.5).

Case when $\xi \neq \xi_0$. Then, we have

$$\mathbb{E}[Z^{\tilde{x}^1_0} Z^{x^1_1}] = \int \sigma(u)\sigma\left(v - \lambda z \sigma'(u)\right) p_{u,v}(u,v) p_z(z) \mathrm{d}u \mathrm{d}v \mathrm{d}z,$$

where $p_{u,v}$ and and p_z are the densities of non-degenerate Gaussians and are thus well-defined and positive everywhere. Again, we can assume $\lambda \geq 0$ W.L.O.G. We consider the point $(u^*, v^*, z^*) = (1, 1, -1)$ at which the integrand is > 0 since σ and σ' are positive on the positive part of the real line. Hence, the integral is > 0 because the integrand is a continuous function, since σ and σ' are continuous, which concludes the proof. □

Lemma A.8.10 (Intermediate layer of forward pass of IP-LLR at $t = 1$)**.** *Consider the IP-LLR parameterization with an activation function σ satisfying Assumption 3. Let ξ be an input to the network, let $l \in [2, L]$, and assume $\mathring{\chi}_0 \neq 0$. Then, dropping the dependency of the first forward-backward pass on ξ_0, and that of the second forward pass on ξ, one has:*

(i) $Z^{h_1^l} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}] Z^{d\tilde{h}_0^l} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}] \widehat{Z}^{d\tilde{x}_0^l} \sigma'(\widehat{Z}^{\tilde{h}_0^l})$,

(ii) $Z^{\tilde{h}_0^l}$ *and* $Z^{d\tilde{x}_0^l}$ *are independent,*

(iii) $0 < \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_1^l}] < \infty$.

Proof. We prove the result by induction on l, the case of $l = 1$ has already been dealt with in Lemma A.8.9. Let $l \in [1, L-1]$, and assume $0 < \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_1^l}] < \infty$. Calling $\lambda := -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_1^l}]$, we have $\lambda \neq 0$ by assumption and by the induction hypothesis. Then, by Corollary A.8.7.1 with $t = 1$, we have

$$Z^{h_1^{l+1}} = -\lambda Z^{d\tilde{h}_0^{l+1}}.$$

Moreover, $d\tilde{h}_0^{l+1} = d\tilde{x}_0^{l+1} \odot \sigma'(\tilde{h}_0^{l+1})$, and since all the vectors are part of the Tensor Program, we have by `ZNonLin` $Z^{d\tilde{h}_0^{l+1}} = Z^{d\tilde{x}_0^{l+1}} \sigma'(Z^{\tilde{h}_0^{l+1}})$. On the other hand, by Lemma A.3.2, we have $Z^{d\tilde{h}_0^{l+1}} = \widehat{Z}^{d\tilde{h}_0^{l+1}}$ and $Z^{\tilde{h}_0^{l+1}} = \widehat{Z}^{\tilde{h}_0^{l+1}}$, and finally by the `ZHat` rule, since $\tilde{h}_0^{l+1} = \widehat{W}^{l+1} \tilde{x}_1^l$ and $d\tilde{x}_0^{l+1} = U^{L+1}$ if $l = L - 1$ and $(\widehat{W}^{l+2})^\top d\tilde{h}^{l+2}$ otherwise, we get that $\widehat{Z}^{\tilde{h}_0^{l+1}}$ and $\widehat{Z}^{d\tilde{h}_0^{l+1}}$ are independent. In addition, we have

$$\mathbb{E}[Z^{\tilde{x}_0^{l+1}} Z^{x_1^{l+1}}] = \mathbb{E}[\sigma(Z^{\tilde{h}_0^{l+1}}) \sigma(-\lambda Z^{d\tilde{x}_0^{l+1}} \sigma'(Z^{\tilde{h}_0^{l+1}}))].$$

The expectation is non-negative because σ is and it is finite by Lemma A.13.1 because the integrand is a polynomially bounded function of the Gaussian vector $(Z^{\tilde{h}_0^{l+1}}, Z^{d\tilde{x}_0^{l+1}})$ (and thus of Z_0, see Definition A.13.1). Using the positive p-homogeneity of σ and the fact that $\text{sign}(\sigma'(z)) = \text{sign}(z)$ (see Appendix A.5), and calling $\epsilon = \text{sign}(\lambda) \in \{-1, 1\}$, we have

$$\begin{aligned}
\mathbb{E}[Z^{\tilde{x}_0^{l+1}} Z^{x_1^{l+1}}] &= \mathbb{E}\left[\mathbb{E}\left[Z^{\tilde{x}_0^{l+1}} Z^{x_1^{l+1}} \middle| Z^{\tilde{h}_0^{l+1}}\right]\right] \\
&= |\lambda|^p \, \mathbb{E}\left[\sigma(Z^{\tilde{h}_0^{l+1}}) |\sigma'(Z^{\tilde{h}_0^{l+1}})|^p \mathbb{E}\left[\sigma(-\epsilon \, \text{sign}(Z^{\tilde{h}_0^{l+1}}) Z^{d\tilde{x}_0^{l+1}}) \middle| Z^{\tilde{h}_0^{l+1}}\right]\right],
\end{aligned}$$

Now since $\epsilon\,\mathrm{sign}(Z^{\tilde{h}_0^{l+1}}) \in \{-1,1\}$, $Z^{d\tilde{x}_0^{l+1}}$ and $\epsilon\,\mathrm{sign}(Z^{\tilde{h}_0^{l+1}})Z^{d\tilde{x}_0^{l+1}}$ have the same distribution conditionally on $Z^{\tilde{h}_0^{l+1}}$, so that

$$\begin{aligned}\mathbb{E}\left[\sigma(-\epsilon\,\mathrm{sign}(Z^{\tilde{h}_0^{l+1}})Z^{d\tilde{x}_0^{l+1}})\middle|Z^{\tilde{h}_0^{l+1}}\right] &= \mathbb{E}\left[\sigma(Z^{d\tilde{x}_0^{l+1}})\middle|Z^{\tilde{h}_0^{l+1}}\right]\\ &= \mathbb{E}\left[\sigma(Z^{d\tilde{x}_0^{l+1}})\right].\end{aligned}$$

We thus get

$$\mathbb{E}[Z^{\tilde{x}_0^{l+1}}Z^{x_1^{l+1}}] = |\lambda|^p\mathbb{E}[\sigma(Z^{\tilde{h}_0^{l+1}})|\sigma'(Z^{\tilde{h}_0^{l+1}})|^p]\,\mathbb{E}[\sigma(Z^{d\tilde{x}_0^{l+1}})],$$

and both expectations are positive because they are non-negative and their integrands are > 0 on the positive part of the real line and the Gaussians involved have non-zero density on this subset of $\mathbb{R}$ as they are not degenerate by Lemma A.7.1. This proves $\mathbb{E}[Z^{\tilde{x}_0^{l+1}}Z^{x_1^{l+1}}] > 0$ and concludes the proof by induction. □

Lemma A.8.11 (Last layer of forward pass of IP-LLR at $t = 1$)**.** *Consider the IP-LLR parameterization with an activation function σ satisfying Assumption 3. Let ξ be an input to the network, and assume $\mathring{\chi}_0 \neq 0$. Then, dropping the dependency of the first forward-backward pass on ξ_0, and that of the second forward pass on ξ, one has:*

$$(i)\quad f_1(\xi) = (W^{L+1}(1))^\top x_1^L \xrightarrow[m\to\infty]{a.s.} \mathring{f}_1(\xi) := \mathbb{E}[Z^{U^{L+1}}Z^{x_1^L}] - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^L}Z^{x_1^L}],$$

$$(ii)\quad Z^{U^{L+1}} \text{ and } Z^{\tilde{h}_0^L} \text{ are independent},$$

$$(ii)\quad 0 < |\mathring{f}_1(\xi)| < \infty.$$

Proof. Claim (i) comes from Lemma A.8.6, in which we have already proved that the limit $\mathring{f}_1(\xi)$ is finite as a result of Lemma A.13.1 and the fact that the integrands are polynomially bounded functions of the Gaussian vector $(Z^{\tilde{h}_0^L}, Z^{U^{L+1}})$ which has finite (and diagonal as we will see shortly) covariance matrix. In addition, by Lemma A.3.2, we have $Z^{\tilde{h}_0^L} = \widehat{Z}^{\tilde{h}_0^L}$ and by definition in `ZInit` $Z^{U^{L+1}} = \widehat{Z}^{U^{L+1}}$. Finally, by the `ZHat` rule, the latter two random variables are independent since $\tilde{h}_0^L = \widehat{W}^L\tilde{x}_0^{L-1}$. Let $\epsilon := \mathrm{sign}(\mathring{\chi}_0)$ and $\lambda_l := \mathbb{E}[Z^{\tilde{x}_0^l}Z^{x_1^l}]$ for $l \in \{L-1, L\}$. We have $\lambda_{L-1}, \lambda_L > 0$ by Lemma A.8.10, and using again the fact that $\mathrm{sign}(\sigma'(z)) = \mathrm{sign}(z)$ and the positive p-homogeneity of σ, we have

$$\begin{aligned}\mathbb{E}[Z^{U^{L+1}}Z^{x_1^L}] &= \mathbb{E}\left[\mathbb{E}\left[Z^{U^{L+1}}Z^{x_1^L}\middle|Z^{\tilde{h}_0^L}\right]\right]\\ &= \mathbb{E}\left[\mathbb{E}\left[Z^{U^{L+1}}\sigma(-\eta\mathring{\chi}_0\lambda_{L-1}Z^{U^{L+1}}\sigma'(Z^{\tilde{h}_0^L}))\middle|Z^{\tilde{h}_0^L}\right]\right]\\ &= |\eta\lambda_{L-1}\mathring{\chi}_0|^p\,\mathbb{E}\left[|\sigma'(Z^{\tilde{h}_0^L})|^p\mathbb{E}\left[Z^{U^{L+1}}\sigma(-\epsilon\,\mathrm{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})\middle|Z^{\tilde{h}_0^L}\right]\right].\end{aligned}$$

Since $Z^{U^{L+1}}$ and $-Z^{U^{L+1}}$ have the same distribution, and it is independent of $Z^{\tilde{h}_0^L}$, and since $\epsilon\,\mathsf{sign}(Z^{\tilde{h}_0^L}) \in \{-1, 1\}$, we have

$$\begin{aligned}\mathbb{E}\left[-\epsilon\,\mathsf{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}}\sigma(-\epsilon\,\mathsf{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})\middle|Z^{\tilde{h}_0^L}\right] &= \mathbb{E}\left[Z^{U^{L+1}}\sigma(Z^{U^{L+1}})\middle|Z^{\tilde{h}_0^L}\right] \\ &= \mathbb{E}\left[Z^{U^{L+1}}\sigma(Z^{U^{L+1}})\right],\end{aligned}$$

so that

$$\begin{aligned}\mathbb{E}\left[|\sigma'(Z^{\tilde{h}_0^L})|^p\mathbb{E}\left[Z^{U^{L+1}}\sigma(-\epsilon\,\mathsf{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})\middle|Z^{\tilde{h}_0^L}\right]\right] = \\ -\epsilon\,\mathbb{E}\left[\mathsf{sign}(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^p\right]\mathbb{E}\left[Z^{U^{L+1}}\sigma(Z^{U^{L+1}})\right].\end{aligned}$$

We thus get

$$\mathbb{E}[Z^{U^{L+1}}Z^{x_1^L}] = -\epsilon\,|\eta\lambda_{L-1}\mathring{\chi}_0|^p\,\mathbb{E}\left[\mathsf{sign}(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^p\right]\mathbb{E}\left[Z^{U^{L+1}}\sigma(Z^{U^{L+1}})\right]$$

We now prove that both expectations are positive. This is where the assumption that $\alpha > \beta$ (see Appendix A.5) appears to be crucial. We start with the first one. Since $Z^{\tilde{h}_0^L}$ has a zero-mean Gaussian distribution with positive variance (by Lemma A.7.1), its density p_z is positive everywhere and symmetric, and we have

$$\begin{aligned}\mathbb{E}\left[\mathsf{sign}(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^p\right] &= \int_{z=0}^{+\infty}(\alpha p)^p z^{p(p-1)}p_z(z)\mathrm{d}z + \int_{z=-\infty}^{0}-(\beta p)^p(-z)^{p(p-1)}p_z(z)\mathrm{d}z \\ &= (\alpha p)^p\int_{z=0}^{+\infty}z^{p(p-1)}p_z(z)\mathrm{d}z - (\beta p)^p\int_{z=0}^{+\infty}z^{p(p-1)}\mathrm{d}z \\ &= (\alpha^p-\beta^p)p^p\int_{z=0}^{+\infty}z^{p(p-1)}p_z(z)\mathrm{d}z.\end{aligned}$$

The second equality stems from the change of variable $z \leftarrow -z$ in the second integral and from the symmetry of p_z with respect to $z = 0$. The last integral is > 0 because its integrand is > 0 on the corresponding domain, and $\alpha^p - \beta^p > 0$ since $\alpha > \beta$ by assumption and $p > 0$. For the second expectation, we get with a similar reasoning that

$$\begin{aligned}\mathbb{E}\left[Z^{U^{L+1}}\sigma(Z^{U^{L+1}})\right] &= \int_{u=0}^{+\infty}u\alpha u^p p_u(u)\mathrm{d}u + \int_{u=-\infty}^{0}u\beta(-u)^p p_u(u)\mathrm{d}u \\ &= (\alpha-\beta)\int_{u=0}^{+\infty}u^{p+1}p_u(u)\mathrm{d}u,\end{aligned}$$

which shows the expectation is > 0.

We now look at the second term in $\mathring{f}_1(\xi)$: $-\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^L}Z^{x_1^L}] = -\epsilon\eta|\mathring{\chi}_0|\lambda_L$. Summing this up with the first term, we get

$$\mathring{f}_1(\xi) = -\epsilon\left[\underbrace{|\eta\lambda_{L-1}\mathring{\chi}_0|^p\,\mathbb{E}\left[\mathsf{sign}(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^p\right]\mathbb{E}\left[Z^{U^{L+1}}\sigma(Z^{U^{L+1}})\right] + \eta|\mathring{\chi}_0|\lambda_L}_{>0}\right]$$

which concludes the proof. □

Theorem A.8.12 (Non-trivial learning of IP-LLR at $t = 1$)**.** *Consider an IP-LLR parameterization of an L-hidden layer neural network with an activation function σ satisfying Assumption 3. Let $\xi \in \mathbb{R}^d$ be an input to the network, and assume $\xi_0, \xi, \overset{\circ}{\chi}_0 \neq 0$. Then, one has:*

$$(i)\ \ f_0(\xi) \xrightarrow[m\to\infty]{a.s.} 0$$
$$(ii)\ \ f_1(\xi) \xrightarrow[m\to\infty]{a.s.} \overset{\circ}{f}_1(\xi) \neq 0$$

Proof. Claim (i) has already been proved in Lemma A.8.1, and claim (ii) has been proved in Lemma A.8.11 above. □

Remark. Note that since only quantities of the first ($t = 0$) forward and backward passes and second ($t = 1$) forward pass appear in Lemmas A.8.9, A.8.10, A.8.11, and Theorem A.8.12 we only need to assume we have an integrable parameterization with $c_l = \gamma_l(p)$ for any $l \in [1, L + 1]$ at $t = 0$.

A.9 . Proof that no constant learning rate is possible: Theorem 2.3.2

In this section we prove the result of Theorem 2.3.2 by splitting the proof in two steps. First we show in Lemma A.9.1 that to have stable and non-vanishing updates for integrable parameterizations at $t = 1$, one must use the learning rate exponents $c_l = \gamma_l(p)$ for any $l \in [1, L+1]$ at $t = 0$. Then we show some preliminary results on the second backward pass (at $t = 1$) for integrable parameterizations when $c_l = \gamma_l(p)$ for any $l \in [1, L + 1]$ at $t = 0$, and some other preliminary results on the third forward pass (at $t = 2$) when additionally one uses $c_1 = -1$, $c_l = -2$ for $l \in [2, L]$ and $c_{L+1} = -1$ at $t = 1$. Then we show in Lemma A.9.4, using those preliminary results, that assuming we have $c_l = \gamma_l(p)$ for any $l \in [1, L+1]$ at $t = 0$, to have stable and non-vanishing updates at $t = 2$ for integrable parameterizations, one must use the learning rate exponents $c_1 = -1$, $c_l = -2$ for $l \in [2, L]$ and $c_{L+1} = -1$ at $t = 1$.

A.9.1 . Proof of the first implication for the learning rates at $t = 0$

Lemma A.9.1 (Learning rates for stable learning with IP at $t = 0$)**.** *Consider an L-hidden layer fully-connected neural network with $L \geq 3$ in the integrable parameterization, and with no bias terms, except for the first layer. Assume that the activation function σ satisfies Assumption 3, and that $\lim_{m\to\infty} \partial_2 \ell(y_0, f_0(\xi_0)) \neq 0$. Assume further that $\xi_0^\top \xi_1 \neq 0$. Finally assume that Equation* (2.4) *holds:*

$$\begin{cases} \frac{1}{m}||\Delta W^l(1) x_1^{l-1}||^2 = \Theta(1), \quad l \in [1, L] \\ \left(\Delta W^{L+1}(1)\right)^\top x_1^L = \Theta(1) \end{cases}$$

Then, one necessarily has that at $t = 0$, $c_l = \gamma_l(p)$ for any $l \in [1, L+1]$ (see Definition 2.3.2).

Proof. With the notations introduced in Appendix A.1, the assumptions on the limit of the loss terms at $t = 0$ imply $\mathring{\chi}_0 \neq 0$. Let us consider the updates at $t = 0$. By Corollary A.7.4.1, we have

$$\Delta W^1(1)\xi_1 = -m^{-(c_1-\gamma_1(p))}\eta\chi_0(\xi_0^\top\xi)d\tilde{h}_0^1,$$

so that

$$\frac{1}{m}||\Delta W^1(1)\xi_1||^2 = m^{-2(c_1-\gamma_1(p))}\left[\eta\chi_0(\xi_0^\top\xi_1)\right]^2\frac{1}{m}\sum_{q=1}^m\left(d\tilde{h}_{0,q}^1\right)^2.$$

From the master theorem, we get that $\sum_{q=1}^m (d\tilde{h}_{0,q}^1)^2/m$ converges almost surely towards $\mathbb{E}[(Z^{d\tilde{h}_0^1})^2]$ which is > 0 and finite by Lemma A.7.1. On the other hand, $\left[\eta\chi_0(\xi_0^\top\xi_1)\right]^2$ converges almost surely to $\left[\eta\mathring{\chi}_0(\xi_0^\top\xi_1)\right]^2$, which is > 0 by assumption, and finite.

If $c_1 > \gamma_1(p)$, then $c_1 - \gamma_1(p) > 0$, and $||\Delta W^1(1)\xi_1||^2/m \to 0$ almost surely, which is impossible since by assumption, almost surely, there exits $A > 0$ such that for large enough m, $A \leq ||\Delta W^1(1)\xi_1||^2/m$.

If $c_1 < \gamma_1(p)$, then $c_1 - \gamma_1(p) < 0$, and $||\Delta W^1(1)\xi_1||^2/m \to \infty$ almost surely, which is impossible since by assumption, almost surely, there exits $B > 0$ such that for large enough m, $||\Delta W^1(1)\xi_1||^2/m \leq B$.

We thus have that $c_1 = \gamma_1(p)$. Let $l \in [1, L-1]$ and assume that $c_k = \gamma_k(p)$ for $k \in [1, l]$. Then by Lemmas A.8.9 and A.8.10, we have $0 < \mathbb{E}[Z^{\tilde{x}_0^k}Z^{x_1^k}] < \infty$ for any $k \in [1, l]$. We have

$$\frac{1}{m}||\Delta W^{l+1}(1)x_1^l||^2 = m^{-2(c_{l+1}-\gamma_{l+1}(p))}\left[\eta\chi_0\frac{(\tilde{x}_0^l)^\top x_1^l}{m}\right]^2\frac{1}{m}\sum_{q=1}^m\left(d\tilde{h}_{0,q}^{l+1}\right)^2.$$

From the master theorem, we get that $\sum_{q=1}^m (d\tilde{h}_{0,q}^{l+1})^2/m$ converges almost surely towards $\mathbb{E}[(Z^{d\tilde{h}_0^{l+1}})^2]$ which is > 0 and finite by Lemma A.7.1. On the other hand, $\left[\eta\chi_0(\tilde{x}_0^l)^\top x_1^l/m\right]^2$ converges almost surely to $\left[\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^l}Z^{x_1^l}]\right]^2$, which is > 0 and finite.

If $c_{l+1} > \gamma_{l+1}(p)$, then $c_{l+1} - \gamma_{l+1}(p) > 0$, and $||\Delta W^{l+1}(1)x_1^l||^2/m \to 0$ almost surely, which is impossible since by assumption, almost surely, there exits $A > 0$ such that for large enough m, $A \leq ||\Delta W^{l+1}(1)x_1^l||^2/m$.

If $c_{l+1} < \gamma_{l+1}(p)$, then $c_{l+1} - \gamma_1(p) < 0$, and $||\Delta W^{l+1}(1)x_1^l||^2/m \to \infty$ almost surely, which is impossible since by assumption, almost surely, there exits $B > 0$ such that for large enough m, $||\Delta W^{l+1}(1)x_1^l||^2/m \leq B$.

Therefore, we have $c_{l+1} = \gamma_{l+1}(p)$. By induction, we thus get that $c_l = \gamma_l(p)$ for any $l \in [1, L]$, which means in particular that $0 < \mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_1^L}] < \infty$ by Lemma A.8.10. Finally, we have

$$(\Delta W^{L+1}(1))^\top x_1^L = -m^{-(c_{L+1}-\gamma_{L+1}(p))}\eta\chi_0 \frac{(\tilde{x}_0^L)^\top x_1^L}{m}.$$

The term $\eta\chi_0(\tilde{x}_0^L)^\top x_1^L/m$ converges almost surely towards $\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_1^L}]$, whose absolute value is > 0 and finite. Therefore, if $c_{L+1} > \gamma_{L+1}(p)$ then $c_{L+1} - \gamma_{L+1}(p) > 0$ so that $(\Delta W^{L+1}(1))^\top x_1^L \to 0$ almost surely, which is impossible since by assumption, almost surely, there exits $A > 0$ such that for large enough m, $A \leq |(\Delta W^{L+1}(1))^\top x_1^L|$. If $c_{L+1} < \gamma_{L+1}(p)$ then $c_{L+1} - \gamma_{L+1}(p) < 0$ so that $(\Delta W^{L+1}(1))^\top x_1^L \to \infty$ almost surely, which is impossible since by assumption, almost surely, there exits $B > 0$ such that for large enough m, $|(\Delta W^{L+1}(1))^\top x_1^L| \leq B$. Thus, we must have $c_{L+1} = \gamma_{L+1}(p)$, which concludes the proof for the first part. □

A.9.2 . Preliminaries on the second backward pass ($t = 1$)

Before we move on to the proof of the second part of the claim of Theorem 2.3.2, we stop and prove some preliminary results on the second backward pass (at $t = 1$) which will come in handy later on. Similarly to what we did for $\mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_1^l}]$, we wish to prove that the quantity $0 < \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}] < \infty$ for any $l \in [2, L]$.

Lemma A.9.2 (Backward pass of IP-LLR at $t = 1$)**.** *Consider the IP-LLR parameterization of an L hidden-layer network, and assume that the activation function σ satisfies Assumption 3, and that $\lim_{m\to\infty} \partial_2 \ell(y_0, f_0(\xi_0)) \neq 0$. Then, one has that for any $l \in [2, L]$,*

$$0 < \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}] < \infty$$

Remark. Note that since only quantities of the first ($t = 0$) and second ($t = 1$) forward and backward passes appear, we only need to assume we have an integrable parameterization with $c_l = \gamma_l(p)$ for any $l \in [1, L + 1]$ at $t = 0$.

Proof. We start with $l = L$, and then induct over l from $l = L$ to $l = 2$, and we recall that $\lim_{m\to\infty} \partial_2 \ell(y_0, f_0(\xi_0)) =: \mathring{\chi}_0$ by definition (see Appendix A.1), which is thus $\neq 0$ by assumption.

The case $l = L$. By Corollary A.8.8.1, we have $Z^{d\tilde{h}_0^L} = Z^{U^{L+1}}\sigma'(Z^{\tilde{h}_0^L})$ and

$Z^{d\tilde{h}_1^L} = (Z^{U^{L+1}} - \eta\mathring{\chi}_0\sigma(Z^{\tilde{h}_0^L}))\sigma'(Z^{h_1^L})$. We thus have

$$\mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_1^L}] = \underbrace{\mathbb{E}[(Z^{U^{L+1}})^2\sigma'(Z^{\tilde{h}_0^L})\sigma'(Z^{h_1^L})]}_{:=A} + \eta|\mathring{\chi}_0|\underbrace{\mathbb{E}[-\epsilon Z^{U^{L+1}}\sigma'(Z^{\tilde{h}_0^L})\sigma(Z^{\tilde{h}_0^L})\sigma'(Z^{h_1^L})]}_{:=B},$$

with $\epsilon := \text{sign}(\mathring{\chi}_0)$ and we deal with both terms separately. First, by Corollary A.8.7.1 we re-write $Z^{h_1^L}$ as

$$Z^{h_1^L} = -\eta|\mathring{\chi}_0|\epsilon\lambda Z^{U^{L+1}}\sigma'(Z^{\tilde{h}_0^L}),$$

where $\lambda := \mathbb{E}[Z^{\tilde{x}_0^{L-1}} Z^{x_1^{L-1}}] > 0$ by Lemma A.8.10. Using the fact that $\text{sign}(\sigma'(z)) = \text{sign}(z)$ and the positive $(p-1)$-homogeneity of σ', we have

$$\sigma'(Z^{h_1^L}) = (\eta|\mathring{\chi}_0|\lambda)^{p-1}|\sigma'(Z^{\tilde{h}_0^L})|^{p-1}\sigma'(-\epsilon\text{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}}).$$

The first term in $\mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_1^L}]$ is thus equal to

$$\begin{aligned}
A &= (\eta|\mathring{\chi}_0|\lambda)^{p-1}\mathbb{E}\left[\mathbb{E}\left[(Z^{U^{L+1}})^2\sigma'(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^{p-1}\sigma'(-\epsilon\text{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})\middle| Z^{\tilde{h}_0^L}\right]\right] \\
&= (\eta|\mathring{\chi}_0|\lambda)^{p-1}\mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^{p-1}\mathbb{E}\left[(Z^{U^{L+1}})^2\sigma'(-\epsilon\text{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})\middle| Z^{\tilde{h}_0^L}\right]\right] \\
&= (\eta|\mathring{\chi}_0|\lambda)^{p-1}\mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^{p-1}\right]\mathbb{E}\left[(Z^{U^{L+1}})^2\sigma'(Z^{U^{L+1}})\right].
\end{aligned}$$

The third equality stems from the fact that $-\epsilon\text{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}}$ and $Z^{U^{L+1}}$ have the same distribution conditionally on $Z^{\tilde{h}_0^L}$, and from the fact that $(Z^{U^{L+1}})^2 = (-\epsilon\text{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})^2$. We now show that both expectations are > 0. Calling p_z the density of the Gaussian $Z^{\tilde{h}_0^L}$ which is symmetric and positive everywhere since $Z^{\tilde{h}_0^L}$ is not degenerate, the first term is equal to

$$\begin{aligned}
\mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^{p-1}\right] &= (\alpha p)^p\int_{z=0}^{+\infty} z^{p(p-1)}p_z(z)\mathrm{d}z - (\beta p)^p\int_{z=-\infty}^{0}(-z)^{p(p-1)}p_z(z)\mathrm{d}z \\
&= (\alpha^p - \beta^p)p^p\int_{z=0}^{+\infty} z^{p(p-1)}p_z(z)\mathrm{d}z,
\end{aligned}$$

where we have used the change of variable $z \leftarrow -z$ in the second equality, and the last quantity is > 0 since $\alpha > \beta$. With similar calculations, we get

$$\mathbb{E}\left[(Z^{U^{L+1}})^2\sigma'(Z^{U^{L+1}})\right] = (\alpha - \beta)p\int_{u=0}^{+\infty} u^{p+1}p_u(u)\mathrm{d}u > 0,$$

where p_u is the density of the standard Gaussian $Z^{U^{L+1}}$. This thus shows that $A > 0$.

We now turn to the second term B. We have:

$$\begin{aligned}
B = (\eta|\mathring{\chi}_0|\lambda)^{p-1}\times\ & \mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^L})\sigma(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^{p-1}\text{sign}(Z^{\tilde{h}_0^L})\times\right. \\
& \left.\mathbb{E}\left[(-\epsilon\text{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})\sigma'(-\epsilon\text{sign}(Z^{\tilde{h}_0^L})Z^{U^{L+1}})\middle| Z^{\tilde{h}_0^L}\right]\right] \\
= (\eta|\mathring{\chi}_0|\lambda)^{p-1}\times\ & \mathbb{E}\left[\sigma'(Z^{\tilde{h}_0^L})\sigma(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^{p-1}\text{sign}(Z^{\tilde{h}_0^L})\right]\mathbb{E}\left[Z^{U^{L+1}}\sigma'(Z^{U^{L+1}})\right] \\
= (\eta|\mathring{\chi}_0|\lambda)^{p-1}\times\ & \mathbb{E}\left[\sigma(Z^{\tilde{h}_0^L})|\sigma'(Z^{\tilde{h}_0^L})|^{p}\right]\mathbb{E}\left[Z^{U^{L+1}}\sigma'(Z^{U^{L+1}})\right].
\end{aligned}$$

We now prove again that both expectations are > 0. The first integrand is non-negative everywhere and positive on the positive part of the real line where the Gaussian $Z^{\tilde{h}_0^L}$ has non-zero density, which shows the first expectation is > 0. The same argument holds for the second expectation since $Z^{U^{L+1}}$ and $\sigma'(Z^{U^{L+1}})$ are of the same sign, which also leads to a positive expectation, which finally gives $B > 0$, thereby concluding the proof.

The case $l \in [2, L-1]$**.**
Let $l \in [2, L-1]$ and assume $0 < \nu := \mathbb{E}[Z^{d\tilde{h}_0^{l+1}} Z^{d\tilde{h}_1^{l+1}}] < \infty$. Calling $\epsilon := \text{sign}(\mathring{\chi}_0)$, on the one hand, we have by Corollary A.8.8.1

$$Z^{d\tilde{x}_1^l} = -\eta|\mathring{\chi}_0|\nu\epsilon\sigma(Z^{\tilde{h}_0^l}),$$

and on the other hand, with $\lambda := \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}]$, which is > 0 by Lemmas A.8.10 and A.8.9 (if $l = 2$)

$$\sigma'(Z^{h_1^l}) = (\eta|\mathring{\chi}_0|\lambda)^{p-1}|\sigma'(Z^{\tilde{h}_0^l})|^{p-1}\sigma'(-\epsilon\text{sign}(Z^{\tilde{h}_0^l})Z^{d\tilde{x}_0^l}).$$

Recalling that $Z^{d\tilde{h}_0^l} = Z^{d\tilde{x}_0^l}\sigma'(Z^{\tilde{h}_0^l})$ and $Z^{d\tilde{h}_1^l} = Z^{d\tilde{x}_1^l}\sigma'(Z^{h_1^l})$, this leads to

$$\mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}] = \eta|\mathring{\chi}_0|\nu(\eta|\mathring{\chi}_0|\lambda)^{p-1}\mathbb{E}[(-\epsilon\text{sign}(Z^{\tilde{h}_0^l})Z^{d\tilde{x}_0^l})\sigma'(-\epsilon\text{sign}(Z^{\tilde{h}_0^l})Z^{d\tilde{x}_0^l}) \\ \text{sign}(Z^{\tilde{h}_0^l})\sigma'(Z^{\tilde{h}_0^l})|\sigma'(Z^{\tilde{h}_0^l})|^{p-1}\sigma(Z^{\tilde{h}_0^l})],$$

which, by conditioning on $Z^{\tilde{h}_0^l}$ and since $-\epsilon\text{sign}(Z^{\tilde{h}_0^l})Z^{d\tilde{x}_0^l}$ and $Z^{d\tilde{x}_0^l}$ have the same distribution conditionally on $Z^{\tilde{h}_0^l}$, and since $\text{sign}(\sigma'(z)) = \text{sign}(z)$, gives

$$\mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}] = \eta|\mathring{\chi}_0|\nu(\eta|\mathring{\chi}_0|\lambda)^{p-1}\mathbb{E}\left[Z^{d\tilde{x}_0^l}\sigma'(Z^{d\tilde{x}_0^l})\right]\mathbb{E}\left[|\sigma'(Z^{\tilde{h}_0^l})|^p\sigma(Z^{\tilde{h}_0^l})\right].$$

The term in front of the expectations is positive by assumption, and both expectations are positive because their integrands are both non-negative and positive on the positive part of the real line where the Gaussians $Z^{d\tilde{x}_0^l}$ and $Z^{\tilde{h}_0^l}$ have non-zero density. The expectations are also finite by Lemma A.13.1 because their integrands are polynomially bounded functions of some Gaussian vector with finite covariance variance matrix. By induction, we thus get that $0 < \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}] < \infty$ for any $l \in [2, L]$, which concludes the proof. □

A.9.3 . Preliminaries on the third forward pass ($t = 2$)

In this section we wish to prove that similarly to the second forward pass, the quantities the quantities $\mathbb{E}[Z^{x_1^l} Z^{x_2^l}]$ and $\mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}]$ (which appear in the third forward pass at $t = 2$) are > 0 for any $l \in [1, L]$ when using the IP-LLR learning rates at $t = 0$ and $t = 1$. We assume here that the training samples ξ_0, ξ_1, ξ_2 are all distinct, which is probably not necessary for the result to hold but simplifies somewhat some parts of the proof and is in any case a very natural assumption.

Lemma A.9.3 (Forward pass of IP-LLR at $t = 2$)**.** *Consider the IP-LLR parameterization of an L hidden-layer network, and assume that the activation function σ satisfies Assumption 3, and that $\lim_{m\to\infty} \partial_2 \ell(y_0, f_0(\xi_0)) \neq 0$ and $\lim_{m\to\infty} \partial_2 \ell(y_1, f_1(\xi_1)) \neq 0$. Assume further that the first three training samples ξ_0, ξ_1, ξ_2 are all distinct. Then, one has that for any $l \in [1, L]$,*

$$0 < \mathbb{E}[Z^{x_1^l} Z^{x_2^l}] < \infty$$
$$0 < \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}] < \infty$$

Proof. We start with the case $l = 1$ and then induct over l from $l = 1$ to $l = L$ for both expectations simultaneously as the derivations are very similar.

The case $l = 1$.
Let us first unwind the expressions of $Z^{h_1^1}$ and $Z^{h_2^1}$. We have

$$Z^{h_1^1} = Z^{\tilde{h}_0^1(\xi_1)} - \eta \mathring{\chi}_0 (\xi_0^\top \xi_1 + 1) Z^{d\tilde{x}_0^1} \sigma'(Z^{\tilde{h}_0^1}),$$

and

$$Z^{h_2^1} = Z^{\tilde{h}_0^1(\xi_2)} - \eta \mathring{\chi}_0 (\xi_0^\top \xi_2 + 1) Z^{d\tilde{x}_0^1} \sigma'(Z^{\tilde{h}_0^1}) - \eta \mathring{\chi}_1 (\xi_1^\top \xi_2 + 1) Z^{d\tilde{x}_1^1} \sigma'(Z^{h_1^1}).$$

The case of $\mathbb{E}[Z^{x_1^1} Z^{x_2^1}]$.
Recalling that $Z^{d\tilde{x}_1^1} = -\eta \mathring{\chi}_0 \nu \sigma(Z^{\tilde{h}_0^1})$ where $\nu := \mathbb{E}[Z^{d\tilde{h}_0^2} Z^{d\tilde{h}_1^2}]$. With the assumption that ξ_0, ξ_1, ξ_2 are all distinct, the vector $(Z^{\tilde{h}_0^1}, Z^{\tilde{h}_0^1(\xi_1)}, Z^{\tilde{h}_0^1(\xi_2)}, Z^{d\tilde{x}_0^1})$ has a non-degenerate Gaussian distribution, and we thus get

$$\mathbb{E}[Z^{x_1^1} Z^{x_2^1}] = \int \sigma\left(u_1 - \mu_0 z \sigma'(u_0)\right) \sigma\left(u_2 - \mu_1 z \sigma'(u_0) + \mu_2 \sigma(u_0) \sigma'\left(u_1 - \mu_0 z \sigma'(u_0)\right)\right) \times q(u_0, u_1, u_2) p_z(z) \mathrm{d}(u_0, u_1, u_2) \mathrm{d}z,$$

where $\mu_0 := \eta \mathring{\chi}_0 (\xi_0^\top \xi_1 + 1)$, $\mu_1 := \eta \mathring{\chi}_0 (\xi_0^\top \xi_2 + 1)$ and $\mu_2 := \eta^2 \mathring{\chi}_0 \mathring{\chi}_1 \nu (\xi_1^\top \xi_2 + 1)$, and q and p_z are the densities of non-degenerate Gaussians and are thus positive everywhere. Now the integrand is non-negative everywhere and we wish to show that it is positive at some given point of $\mathbb{R}^4$, and it is also a polynomially bounded function of $(Z^{\tilde{h}_0^1}, Z^{\tilde{h}_0^1(\xi_1)}, Z^{\tilde{h}_0^1(\xi_2)}, Z^{d\tilde{x}_0^1})$ which shows that the expectation is finite. Since $Z^{d\tilde{x}_0^1}$ and $-Z^{d\tilde{x}_0^1}$ have the same distribution and it is independent of $(Z^{\tilde{h}_0^1}, Z^{\tilde{h}_0^1(\xi_1)}, Z^{\tilde{h}_0^1(\xi_2)})$, we can assume that $\mu_0 \geq 0$ W.L.O.G. Consider the point $(u_0^*, u_1^*, u_2^*, z^*)$ defined as $u_0^* = u_1^* = 1$, $z^* = -1$ and

$$u_2^* := |\mu_1| \sigma'(1) + |\mu_2| \sigma(1) \sigma'\left(1 + \mu_0 \sigma'(1)\right) + 1.$$

We show below that the integrand is > 0 at $(u_0^*, u_1^*, u_2^*, z^*)$. Since it is also a continuous function of (u_0, u_1, u_2, z), we get that the expectation is positive.

Let us now show that the integrand is > 0 at $(u_0^*, u_1^*, u_2^*, z^*)$. We have

$$u_1^* - \mu_0 z^* \sigma'(u_0^*) = 1 + \mu_0 \sigma'(1) \geq 1 > 0,$$

and

$$-\mu_1 z^* \sigma'(u_0^*) = \mu_1 \sigma'(1) \geq -|\mu_1| \sigma'(1),$$

and finally

$$\mu_2 \sigma(u_0^*) \sigma' \left(u_1^* - \mu_0 z^* \sigma'(u_0^*)\right) = \mu_2 \sigma(1) \sigma' \left(1 + \mu_0 \sigma'(1)\right) \geq -|\mu_2| \sigma(1) \sigma' \left(1 + \mu_0 \sigma'(1)\right).$$

With the choice for u_2^*, one has that $u_2^* - \mu_1 z \sigma'(u_0^*) + \mu_2 \sigma(u_0^*) \sigma' \left(u_1^* - \mu_0 z^* \sigma'(u_0^*)\right) \geq 1 > 0$, which concludes the proof because σ is positive on the positive part of the real line.

The case of $\mathbb{E}[Z^{\tilde{x}_0^1} Z^{x_2^1}]$.

We have

$$\mathbb{E}[Z^{\tilde{x}_0^1} Z^{x_2^1}] = \int \sigma(u_0) \, \sigma \left(u_2 - \mu_1 z \sigma'(u_0) + \mu_2 \sigma(u_0) \sigma' \left(u_1 - \mu_0 z \sigma'(u_0)\right)\right) \times q(u_0, u_1, u_2) p_z(z) \mathrm{d}(u_0, u_1, u_2) \mathrm{d}z,$$

As for the case of $\mathbb{E}[Z^{x_1^1} Z^{x_2^1}]$, we show that the integrand is > 0 at the same point $(u_0^*, u_1^*, u_2^*, z^*)$ as above, and since it is also a continuous function of (u_0, u_1, u_2, z), we get that the expectation is positive. It is also finite by Lemma A.13.1 because its integrand is a polynomially bounded function of $(Z^{\tilde{h}_0^1}, Z^{\tilde{h}_0^1(\xi_1)}, Z^{\tilde{h}_0^1(\xi_2)}, Z^{d\tilde{x}_0^1})$.

The case $l \in [2, L-1]$.

Let $l \in [2, L-1]$ and assume $\tau := \mathbb{E}[Z^{x_1^{l-1}} Z^{x_2^{l-1}}] > 0$ and $\rho := \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_2^{l-1}}] > 0$. Calling $\lambda := \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}]$ which is > 0 by Lemma A.8.10, and $\nu := \mathbb{E}[Z^{d\tilde{h}_0^{l+1}} Z^{d\tilde{h}_1^{l+1}}]$ which is also > 0 by Lemma A.9.2, we have

$$Z^{h_1^l} = -\eta \mathring{\chi}_0 \lambda Z^{d\tilde{x}_0^l} \sigma'(Z^{\tilde{h}_0^l}),$$

and

$$Z^{h_2^l} = -\eta \mathring{\chi}_0 \rho Z^{d\tilde{x}_0^l} \sigma'(Z^{\tilde{h}_0^l}) - \eta \mathring{\chi}_1 \tau Z^{d\tilde{x}_1^l} \sigma'(Z^{h_1^l}).$$

Finally recall that $Z^{d\tilde{x}_1^l} = -\eta \mathring{\chi}_0 \nu \sigma(Z^{\tilde{h}_0^l})$, and let us call $\mu_0 := \eta \mathring{\chi}_0 \lambda$, $\mu_1 := \eta \mathring{\chi}_0 \rho$ and $\mu_2 := \eta^2 \mathring{\chi}_0 \mathring{\chi}_1 \tau \nu$. μ_0 is $\neq 0$ because of the assumption on $\mathring{\chi}_0$. Since $Z^{d\tilde{x}_0^l}$ and $-Z^{d\tilde{x}_0^l}$ have the same distribution and it is independent of $Z^{\tilde{h}_0^l}$, we can assume $\mu_0 > 0$ W.L.O.G. Note then that since μ_1 is of the same sign as μ_0 ($\lambda \rho > 0$), this also implies $\mu_1 > 0$, and μ_2 has the sign of $\mathring{\chi}_1$. By assumption, $\mathring{\chi}_1 \neq 0$, and by the induction hypothesis and Lemma A.9.2 we have $\mu_2 \neq 0$.

The case of $\mathbb{E}[Z^{x_1^l} Z^{x_2^l}]$.
We have

$$\mathbb{E}[Z^{x_1^l} Z^{x_2^l}] = \int \sigma\left(-\mu_0 z\sigma'(u)\right) \sigma\left(-\mu_1 z\sigma'(u) + \mu_2\sigma(u)\sigma'\left(-\mu_0 z\sigma'(u)\right)\right) p_u(u)p_z(z)\mathrm{d}u\mathrm{d}z,$$

where p_u and p_z are the densities of non-degenerate Gaussians ($Z^{\tilde{h}_0^l}$ and $Z^{d\tilde{x}_0^l}$ respectively) and are thus positive everywhere. Now the integrand is non-negative everywhere and we wish to show that it is positive at some given point of $\mathbb{R}^2$. The integrand is also a polynomially bounded function of $(Z^{\tilde{h}_0^l}, Z^{d\tilde{x}_0^l})$ which shows that the expectation is finite by Lemma A.13.1. Let $z^* = -1$ and $u > 0$. Then, $-\mu_0 z^*\sigma'(u) = \mu_0\sigma'(u) > 0$ so that $\sigma(-\mu_0 z^*\sigma'(u)) > 0$. On the other hand, $-\mu_1 z^*\sigma'(u) = \mu_1\alpha p u^{p-1}$, and

$$\begin{aligned}\mu_2\sigma(u)\sigma'(-\mu_0 z^*\sigma'(u)) &= \mu_2\alpha u^p \alpha p(\mu_0\alpha p)^{p-1}u^{(p-1)^2} \\ &\geq -(\alpha p)|\mu_2|\alpha(\mu_0\alpha p)^{p-1}u^{p-1}u^{(p-1)^2+1}.\end{aligned}$$

This leads to

$$-\mu_1 z^*\sigma'(u) + \mu_2\sigma(u)\sigma'(-\mu_0 z^*\sigma'(u)) \geq \alpha p u^{p-1}\left[\mu_1 - |\mu_2|\alpha(\mu_0\alpha p)^{p-1}u^{(p-1)^2+1}\right].$$

The quantity in the bracket is > 0 as soon as

$$u < \left[\frac{\mu_1}{|\mu_2|\alpha(\mu_0\alpha p)^{p-1}}\right]^{\frac{1}{(p-1)^2+1}} =: \varepsilon$$

Calling $u^* := \epsilon/2$, we thus get that the integrand is > 0 at (u^*, z^*), and since it is a continuous function of (u, z), the integral is positive.

The case of $\mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}]$.
We have

$$\mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}] = \int \sigma\left(u\right) \sigma\left(-\mu_1 z\sigma'(u) + \mu_2\sigma(u)\sigma'\left(-\mu_0 z\sigma'(u)\right)\right) p_u(u)p_z(z)\mathrm{d}u\mathrm{d}z,$$

The integrand is non-negative everywhere and with $z^* = -1$ and $u^* = \varepsilon/2$ as above, one shows that the integrand is > 0 at (u^*, z^*) which in turn implies that the expectation is positive. It is also finite for the same reasons as $\mathbb{E}[Z^{x_1^l} Z^{x_2^l}]$. This now concludes the induction over $l \in [1, L-1]$ which thus shows that $\mathbb{E}[Z^{x_1^l} Z^{x_2^l}]$ and $\mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}]$ are > 0 and finite for any $l \in [1, L-1]$. Those expectations are also finite as their integrands are polynomially bounded functions of Gaussian vectors which have finite covariance matrices.

The case $l = L$.

Let $\tau := \mathbb{E}[Z^{x_1^{L-1}} Z^{x_2^{L-1}}] > 0$ and $\rho := \mathbb{E}[Z^{\tilde{x}_0^{L-1}} Z^{x_2^{L-1}}] > 0$ by the previous induction. Calling $\lambda := \mathbb{E}[Z^{\tilde{x}_0^{L-1}} Z^{x_1^{L-1}}]$ which is > 0 by Lemma A.8.10, we have

$$Z^{h_1^L} = -\eta \mathring{\chi}_0 \lambda Z^{U^{L+1}} \sigma'(Z^{\tilde{h}_0^L}),$$

and

$$Z^{h_2^L} = -\eta \mathring{\chi}_0 \rho Z^{U^{L+1}} \sigma'(Z^{\tilde{h}_0^L}) - \eta \mathring{\chi}_1 \tau Z^{d\tilde{x}_1^L} \sigma'(Z^{h_1^L}).$$

Finally recall that $Z^{d\tilde{x}_1^L} = Z^{U^{L+1}} - \eta \mathring{\chi}_0 \sigma(Z^{\tilde{h}_0^L})$, and let us call $\mu_0 := \eta \mathring{\chi}_0 \lambda$, $\mu_1 := \eta \mathring{\chi}_0 \rho$, $\mu_2 := \eta \mathring{\chi}_1 \tau$, and finally $\mu_3 := \eta^2 \mathring{\chi}_0 \mathring{\chi}_1 \tau$. Since $Z^{d\tilde{x}_0^l}$ and $-Z^{d\tilde{x}_0^l}$ have the same distribution and it is independent of $Z^{\tilde{h}_0^l}$, we can assume $\mu_0 > 0$ W.L.O.G. Note then that since μ_1 is of the same sign as μ_0, this also implies $\mu_1 > 0$, and μ_2 has the sign of $\mathring{\chi}_1$. In addition, with the assumptions and previous results, we have $\mu_2 \neq 0$ and $\mu_3 \neq 0$.

The case of $\mathbb{E}[Z^{x_1^L} Z^{x_2^L}]$.
We have

$$\mathbb{E}[Z^{x_1^L} Z^{x_2^L}] = \int \sigma\left(-\mu_0 z \sigma'(u)\right) \sigma\left(-\mu_1 z \sigma'(u) + (-\mu_2 z + \mu_3 \sigma(u)) \sigma'\left(-\mu_0 z \sigma'(u)\right)\right) \times p_u(u) p_z(z) \mathrm{d}u \mathrm{d}z,$$

where p_u and p_z are the densities of non-degenerate Gaussians ($Z^{\tilde{h}_0^L}$ and $Z^{U^{L+1}}$ respectively) and are thus positive everywhere. Now the integrand is non-negative everywhere and we wish to show that it is positive at some point of $\mathbb{R}^2$. The integrand is also a polynomially bounded function of $(Z^{\tilde{h}_0^L}, Z^{U^{L+1}})$ and the expectation is thus finite by Lemma A.13.1. We first take a closer look at the second term inside σ. Let $z \leq 0, u \geq 0$. We have

$$-\mu_1 z \sigma'(u) = \mu_1 |z| \alpha p u^{p-1},$$

as well as

$$-\mu_2 z + \mu_3 \sigma(u) = -\mu_2 z + \mu_3 \alpha u^p,$$

and

$$\sigma'(-\mu_0 z \sigma'(u)) = \alpha p (\mu_0 \alpha p)^{p-1} |z|^{p-1} u^{(p-1)^2}.$$

We thus get that

$$-\mu_1 z \sigma'(u) + (-\mu_2 z + \mu_3 \sigma(u)) \sigma'\left(-\mu_0 z \sigma'(u)\right) = \alpha p |z| u^{p-1} \left[\underbrace{\mu_1 + (-\mu_2 z + \mu_3 \alpha |u|^p)(\mu_0 \alpha p)^{p-1} |z|^{p-2} |u|^{(p-1)(p-2)}}_{F(u,z)} \right]$$

Because $p-2 \geq 0$, the function F is continuous over $\mathbb{R}^2$, and we have $F(0,0) = \mu_1 > 0$. Therefore, there exists $u^* > 0$ and $z^* < 0$ such that $F(u^*, z^*) > 0$. With such a pair (u^*, z^*) we get that the integrand is > 0 at (u^*, z^*), and since it is a continuous function of (u, z), it follows that the expectation is positive.

The case of $\mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_2^L}]$.
A similar argument to the case of $\mathbb{E}[Z^{x_1^L} Z^{x_2^L}]$ applies and we get that the expectation is positive, which concludes the proof. □

A.9.4 . Proof of the second implication

Lemma A.9.4 (Learning rates for stable learning with IP at $t = 1$)**.** *Consider an L-hidden layer fully-connected neural network with $L \geq 3$ in the integrable parameterization, and with no bias terms, except at the first layer. Assume that the activation function σ satisfies Assumption 3, and that $\lim_{m\to\infty} \partial_2 \ell(y_0, f_0(\xi_0)) \neq 0$ and $\lim_{m\to\infty} \partial_2 \ell(y_1, f_0(\xi_1)) \neq 0$. Assume further that $\xi_1^\top \xi_2 \neq 0$, that the first three training samples ξ_0, ξ_1, ξ_2 are all distinct, and that at $t = 0$ (i.e., to compute $\Delta W^l(1)$) $c_l = \gamma_l(p)$ (see Definition 2.3.2) for any $l \in [1, L+1]$. Finally assume that Equation* (2.5) *holds:*

$$\begin{cases} \frac{1}{m}||\Delta W^l(2) x_2^{l-1}||^2 = \Theta(1), \quad l \in [1, L] \\ (\Delta W^{L+1}(2))^\top x_2^L = \Theta(1) \end{cases}$$

Then, one necessarily has that at $t = 1$, $c_1 = c_{l+1} = -1$ and $c_l = -2$ for any $l \in [2, L]$.

Proof. We first treat the case $l = 1$ and then induct over l from $l = 2$ to $l = L$ and conclude by the case $l = L + 1$. Note that because of the assumptions, Lemma A.9.2 holds and the claim of Lemma A.9.3 will hold at layer l as soon as we show $c_1 = -1$ and $c_k = -2$ for $k \in [2, L]$.

The case $l = 1$.
We have

$$\Delta W^1(2)\xi_2 = -\eta m^{-(1+c_1)} \chi_1({\xi_1}^\top \xi_2) d\tilde{x}_1^1 \odot \sigma'(h_1^1),$$

so that

$$\frac{1}{m}||\Delta W^1(2)\xi_2||^2 = m^{-2(1+c_1)} (\eta \chi_1({\xi_1}^\top \xi_2))^2 \frac{1}{m}||d\tilde{x}_1^1 \odot \sigma'(h_1^1)||^2.$$

Recall that $d\tilde{x}_1^1 = -\eta \chi_0((d\tilde{h}_0^2)^\top d\tilde{h}_1^2)/m \, \sigma(\tilde{h}_0^1)$, so that by the Master Theorem,

$$\frac{1}{m}||d\tilde{x}_1^1 \odot \sigma'(h_1^1)||^2 \xrightarrow[m\to\infty]{a.s.} (\eta \mathring{\chi}_0 \nu)^2 \mathbb{E}[\sigma(Z^{\tilde{h}_0^1})^2 \sigma'(Z^{h_1^1})^2],$$

where $\nu := \mathbb{E}[Z^{d\tilde{h}_0^2} Z^{d\tilde{h}_1^2}] > 0$ by Lemma A.9.2. The term in front of the expectation is > 0 with the assumptions. On the other hand, the term $(\eta \chi_1({\xi_1}^\top \xi_2))^2$

converges almost surely towards $(\eta\mathring{\chi}_1({\xi_1}^\top\xi_2))^2$ which is also >0 with the assumptions. We show below that the expectation is >0, which proves that c_1 must be equal to 1 since by assumption $\frac{1}{m}||\Delta W^1(2)\xi_2||^2=\Theta(1)$. Recall that

$$Z^{h_1^1}=Z^{\tilde{h}_0^1(\xi_1)}-\eta\mathring{\chi}_0(\xi_0^\top\xi_1+1)Z^{d\tilde{x}_0^1}\sigma'(Z^{\tilde{h}_0^1}).$$

The integrand in the expectation is non-negative, and it simply remains to show that is not almost surely zero. Because $Z^{d\tilde{x}_0^1}$ and $-Z^{d\tilde{x}_0^1}$ have the same distribution, and since it is independent of $(Z^{\tilde{h}_0^1},Z^{\tilde{h}_0^1(\xi_1)})$, we can assume W.L.O.G. that $\mu:=\eta\mathring{\chi}_0(\xi_0^\top\xi_1+1)\geq 0$. As usual, the vector $(Z^{\tilde{h}_0^1},Z^{\tilde{h}_0^1(\xi_1)})$ has a Gaussian distribution which is degenerate only if $\xi_1=\xi_0$, which is precluded by the assumptions. Note that in any case, the expectation is finite by Lemma A.13.1 since its integrand is a polynomially bounded function of a Gaussian vector with finite covariance matrix. Since $\xi\neq\xi_0$ by assumption, we have

$$\mathbb{E}[\sigma(Z^{\tilde{h}_0^1})^2\sigma'(Z^{h_1^1})^2]=\int\sigma(u)^2\sigma'\left(v-\mu z\sigma'(u)\right)^2p_{u,v}(u,v)p_z(z)\mathrm{d}u\mathrm{d}v\mathrm{d}z,$$

where $p_{u,v}$ and and p_z are the densities of non-degenerate Gaussians ($(Z^{\tilde{h}_0^1},Z^{\tilde{h}_0^1(\xi_1)})$ and $Z^{d\tilde{x}_0^1}$ respectively) and are thus well-defined and positive everywhere. Again, one sees that at point $(u^*,v^*,z^*)=(1,1,-1)$ the integrand is >0, and since it is a continuous function, this proves that the expectation is positive. It is also finite by Lemma A.13.1 since the integrand is a polynomially bounded function of a Gaussian vector with finite covariance matrix.

The case $l\in[1,L-1]$

Let $l\in[2,L-1]$. We have already shown that $c_1=-1$. Assume now that $c_k=-2$ for $k\in[2,l-1]$ (note that if $l=2$ this means no additional assumption). Then we have

$$\Delta W^l(2)x_2^{l-1}=-\eta m^{-(2+c_l)}\chi_1\frac{(x_1^{l-1})^\top x_2^{l-1}}{m}d\tilde{x}_1^l\odot\sigma'(h_1^l),$$

so that

$$\frac{1}{m}||\Delta W^l(2)x_2^{l-1}||^2=m^{-2(2+c_l)}\left(\eta\chi_1\frac{(x_1^{l-1})^\top x_2^{l-1}}{m}\right)^2\frac{1}{m}||d\tilde{x}_1^l\odot\sigma'(h_1^l)||^2.$$

In addition, we have $d\tilde{x}_1^l=-\eta\chi_0((d\tilde{h}_0^{l+1})^\top d\tilde{h}_1^{l+1})/m\sigma(\tilde{h}_0^l)$, so that by the Master Theorem,

$$\frac{1}{m}||d\tilde{x}_1^1\odot\sigma'(h_1^1)||^2\xrightarrow[m\to\infty]{a.s.}(\eta\mathring{\chi}_0\nu)^2\mathbb{E}[\sigma(Z^{\tilde{h}_0^l})^2\sigma'(Z^{h_1^l})^2],$$

where $\nu:=\mathbb{E}[Z^{d\tilde{h}_0^{l+1}}Z^{d\tilde{h}_1^{l+1}}]$ is such that $0<\nu<\infty$ by Lemma A.9.2. Recall that

$$Z^{h_1^l}=-\eta\mathring{\chi}_0\lambda Z^{d\tilde{x}_0^l}\sigma'(Z^{\tilde{h}_0^l}),$$

with $\lambda := \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}]$ such that $0 < \lambda < \infty$ by Lemmas A.8.9 and A.8.10, which leads to

$$\mathbb{E}[\sigma(Z^{\tilde{h}_0^l})^2 \sigma'(Z^{h_1^l})^2] = \int \sigma(u)^2 \sigma'(-\mu z \sigma'(u))^2 p_u(u) p_z(z) \mathrm{d}u \mathrm{d}z,$$

where $\mu := \eta \overset{\circ}{\chi}_0 \lambda$ which is $\neq 0$ with the assumptions, and p_u and p_z are the densities of two non-degenerate Gaussians ($Z^{\tilde{h}_0^l}$ and $Z^{d\tilde{x}_0^l}$ respectively) and are thus positive everywhere. Since $Z^{d\tilde{x}_0^l}$ and $-Z^{d\tilde{x}_0^l}$ have the same distribution and it is independent of $Z^{\tilde{h}_0^l}$ we can assume $\mu > 0$ W.L.O.G. Then, we see that at point $(u^*, z^*) = (1, -1)$ the integrand is > 0, and since it is a continuous function, this proves that the expectation is positive. It is also finite by Lemma A.13.1 since the integrand is a polynomially bounded function of a Gaussian vector with finite covariance matrix. The term $(\eta \overset{\circ}{\chi}_0 \nu)^2$ in front of the expectation is > 0 and finite with the assumptions. Finally the term $(\eta \chi_1 ((x_1^{l-1})^\top x_2^{l-1})/m)^2$ converges almost surely towards $(\eta \overset{\circ}{\chi}_1 \tau)^2$ by the Master Theorem, where $\tau := \mathbb{E}[Z^{x_1^{l-1}} Z^{x_2^{l-1}}]$ is > 0 and finite by Lemma A.9.3, which shows that $(\eta \overset{\circ}{\chi}_1 \tau)^2$ is > 0 and finite with the assumptions. Since $||d\tilde{x}_1^1 \odot \sigma'(h_1^1)||^2/m = \Theta(1)$ by assumption, then c_l must be equal to -2 otherwise $||d\tilde{x}_1^1 \odot \sigma'(h_1^1)||^2/m$ would either converge towards 0 or diverge towards ∞ almost surely.

The case $l = L$. We have already proved that at $t = 1$, $c_1 = -1$ and $c_l = -2$ for $l \in [2, L]$. We have

$$|(\Delta W^{L+1}(2))^\top x_2^L| = \eta m^{-(1+c_{L+1})} |\chi_1| \left| \frac{(x_1^L)^\top x_2^L}{m} \right|.$$

By the Master Theorem,

$$\left| \frac{(x_1^L)^\top x_2^L}{m} \right| \xrightarrow[m \to \infty]{a.s.} \mathbb{E}[Z^{x_1^L} Z^{x_2^L}]$$

which is > 0 and finite by Lemma A.9.3. On the other hand, $\eta|\chi_1|$ converges almost surely towards $\eta|\overset{\circ}{\chi}_1|$ which is also > 0 and finite. This shows that since $|(\Delta W^{L+1}(2))^\top x_2^L| = \Theta(1)$ then we must have $c_{L+1} = -1$ to avoid vanishing towards 0 or explosion towards $+\infty$ as $m \to \infty$, which concludes the proof. □

A.10 . Proof of the non-triviality of IP-LLR: Theorem 2.4.1

Proof. Claims (i) and (ii) of Theorem 2.4.1 have already been shown in Theorem A.8.12. Claim (iii) simply stems from Corollary A.8.7.1 with $t = 2$ and the fact that all the variables Z which appear are polynomially bounded functions of the vector Z_0 (see Definition A.13.1) by a simple induction. □

A.11 . Proof of the equivalence between IP-LLR and μP: Proposition 2.4.1 and Theorem 2.4.2

In this section, we present the proofs of the equivalence between IP-LLR and hybrid versions of μP both at finite-width and in the large-width limit. Because we need to use the homogeneity property, we consider a positively p-homogeneous activation function σ and no bias terms except at the first layer for all the parameterizations we consider. We assume $p \geq 1$ for the finite-width case, which includes ReLU, and $p \geq 2$ in the infinite-width case as we use the Tensor Program framework for the proof and thus require some smoothness.

A.11.1 . Finite-width equivalence: Proposition 2.4.1

We start with a preliminary Lemma showing the equivalence at $t = 1$ and then do the proof of Proposition 2.4.1 by induction.

Equivalence at $t = 1$

Lemma A.11.1 (First weight updates of HP)**.** *Consider the IP-LLR and HP parameterizations with a positively p-homogeneous activation function, and $p \geq 1$, and no bias terms except at the first layer, and let us sub/super-script the variables of each models with IP and HP respectively. Assume the first training sample (ξ_0, y_0) and the loss ℓ are the same for both parameterizations. Assume further that $\chi_0^{HP} \neq 0$, and simply denote by η the base learning rate of the IP-LLR parameterization. Finally consider for HP the initial learning rate: $\eta_{HP}(0) = (\chi_0^{IP}/\chi_0^{HP})\eta$, and let $\xi \in \mathbb{R}^d$ be an input to both networks. Then, dropping the dependency of the weights at $t = 1$ on η and η_{HP}, one has:*

$$\begin{aligned} \forall l \in [1, L+1], \quad W_{HP}^l(1) &= W_{IP}^l(1) \\ B_{HP}^1(1) &= B_{IP}^1(1) \\ f_1^{HP}(\xi) &= f_1^{IP}(\xi) \end{aligned}$$

Proof. By definition (see Section 2.4.2), we have

$$\begin{aligned} W_{\mathsf{HP}}^1(1) &= W_{\mathsf{IP}}^1(0) + \Delta W_{\mu\mathsf{P}}^1(1) \\ B_{\mathsf{HP}}^1(1) &= B_{\mathsf{IP}}^1(0) + \Delta B_{\mu\mathsf{P}}^1(1) \\ W_{\mathsf{HP}}^l(1) &= W_{\mathsf{IP}}^l(0) + \Delta W_{\mu\mathsf{P}}^l(1), \qquad l \in [2, L] \\ W_{\mathsf{HP}}^{L+1}(1) &= W_{\mathsf{IP}}^{L+1}(0) + \Delta W_{\mu\mathsf{P}}^{L+1}(1) \end{aligned}$$

Using Corollaries A.7.4.2, and Lemma A.7.5, and the fact that $\eta_{\mathsf{HP}}(0)\chi_0^{\mu\mathsf{P}} = \eta\chi_0^{\mathsf{IP}}$ we have:

$$\begin{aligned} \Delta W_{\mu\mathsf{P}}^1(1) &= -\eta_{\mathsf{HP}}(0)\chi_0^{\mu\mathsf{P}} d\tilde{h}_0^1 \xi_0^\top \\ &= -\eta\chi_0^{\mathsf{IP}} d\tilde{h}_0^1 \xi_0^\top \\ &= \Delta W_{\mathsf{IP}}^1(1), \end{aligned}$$

$$\begin{aligned}\Delta B^1_{\mu\mathsf{P}}(1) &= -\eta_{\mathsf{HP}}(0)\chi_0^{\mu\mathsf{P}} d\tilde{h}_0^1 \\ &= -\eta\chi_0^{\mathsf{IP}} d\tilde{h}_0^1 \\ &= \Delta B^1_{\mathsf{IP}}(1),\end{aligned}$$

and, for $l \in [2, L]$

$$\begin{aligned}\Delta W^l_{\mu\mathsf{P}}(1) &= -\eta_{\mathsf{HP}}\chi_0^{\mu\mathsf{P}} \frac{d\tilde{h}_0^l(\tilde{x}_0^{l-1})^\top}{m} \\ &= -\eta\chi_0^{\mathsf{IP}} \frac{d\tilde{h}_0^l(\tilde{x}_0^{l-1})^\top}{m} \\ &= \Delta W^l_{\mathsf{IP}}(1),\end{aligned}$$

and finally

$$\begin{aligned}\Delta W^{L+1}_{\mu\mathsf{P}}(1) &= -\eta_{\mathsf{HP}}\chi_0^{\mu\mathsf{P}} \tilde{x}_0^L/m \\ &= -\eta\chi_0^{\mathsf{IP}} \tilde{x}_0^L/m \\ &= \Delta W^{L+1}_{\mathsf{IP}}(1),\end{aligned}$$

where the $\Delta W^l_{\mathsf{IP}}(1)$ and $\Delta B^1(1)$ are computed with the base learning rate η. We then get $W^l_{\mathsf{HP}}(1) = W^l_{\mathsf{IP}}(1)$ for all l, and it follows that for any input ξ, $f_1^{\mathsf{HP}}(\xi) = f_1^{\mathsf{IP}}(\xi)$. □

Proof of Proposition 2.4.1

Proof. We first show by induction that the effective weight matrices and the effective biases of the first layer are the same for both parameterizations at any time step ≥ 1, which will then immediately yield the result. We have already shown in Lemma A.11.1 that with the choice of initial learning rate for HP, $W^l_{\mathsf{HP}}(1) = W^l_{\mathsf{IP}}(1)$ for all $l \in [1, L+1]$, and $B^l_{\mathsf{HP}}(1) = B^l_{\mathsf{IP}}(1)$ as well as $f_1^{\mathsf{HP}}(\xi) = f_1^{\mathsf{IP}}(\xi)$.

Now let $s \geq 1$, and assume that for all $l \in [1, L+1]$, $W^l_{\mathsf{HP}}(s) = W^l_{\mathsf{IP}}(s)$, and $B^1_{\mathsf{HP}}(s) = B^1_{\mathsf{HP}}(s)$. We want to show that this also holds true for the next time step $s+1$. An easy induction shows that since the effective weights of all layers are equal, and since by assumption the s-th training sample (ξ_s, y_s) is the same for both parameterization, we get that for any $l \in [1, L+1]$, $x^l_{s,\mathsf{HP}} = x^l_{s,\mathsf{IP}}$, $h^l_{s,\mathsf{HP}} = h^l_{s,\mathsf{IP}}$, as well as $f_s^{\mathsf{HP}}(\xi_s) = f_s^{\mathsf{IP}}(\xi_s)$, and therefore $\chi_s^{\mathsf{HP}} = \chi_s^{\mathsf{IP}}$ since by assumption both parameterization use the same loss. This in turn will give by another easy induction that for any $l \in [1, L+1]$, $dx^l_{s,\mathsf{HP}} = dx^l_{s,\mathsf{IP}}$, $dh^l_{s,\mathsf{HP}} = dh^l_{s,\mathsf{IP}}$. Now, by Equation (A.3) we have, on the one hand (recall that $s+1 \geq 2$ so that the base learning for both models for the $(s+1)$-th SGD step is η)

$$\Delta W^1_{\mathsf{HP}}(s+1) = -\eta m^{-(2a_1^{\mu\mathsf{P}} + c_1^{\mu\mathsf{P}})} dh^1_{s,\mathsf{HP}} \xi_s^\top$$

and for $l \in [2, L]$

$$\Delta W_{\mathsf{HP}}^l(s) = -\eta m^{-(2a_l^{\mu\mathsf{P}} + c_l^{\mu\mathsf{P}})} dh_{s,\mathsf{HP}}^l x_{s,\mathsf{HP}}^{l-1}$$

and finally

$$\Delta W_{\mathsf{HP}}^{L+1}(s) = -\eta m^{-(2a_{L+1}^{\mu\mathsf{P}} + c_{L+1}^{\mu\mathsf{P}})} x_{s,\mathsf{HP}}^{L}$$

On the other hand, we have

$$\Delta W_{\mathsf{IP}}^1(s) = -\eta m^{-(2a_1^{\mathsf{IP}} + c_1^{\mathsf{IP}})} dh_{s,\mathsf{IP}}^1 \xi_s^\top$$

and for $l \in [2, L]$

$$\Delta W_{\mathsf{IP}}^l(s) = -\eta m^{-(2a_l^{\mathsf{IP}} + c_l^{\mathsf{IP}})} dh_{s,\mathsf{IP}}^l x_{s,\mathsf{HP}}^{l-1}$$

and finally

$$\Delta W_{\mathsf{IP}}^{L+1}(s) = -\eta m^{-(2a_{L+1}^{\mathsf{IP}} + c_{L+1}^{\mathsf{IP}})} x_{s,\mathsf{IP}}^{L}$$

To see that the quantities are equal, we only need to observe that since $s+1 \geq 1$

$$\begin{aligned} 2a_1^{\mu\mathsf{P}} + c_1^{\mu\mathsf{P}} &= -1 = 2a_1^{\mathsf{IP}} + c_1^{\mathsf{IP}} \\ 2a_l^{\mu\mathsf{P}} + c_l^{\mu\mathsf{P}} &= 0 = 2a_l^{\mathsf{IP}} + c_l^{\mathsf{IP}} \\ 2a_{L+1}^{\mu\mathsf{P}} + c_{L+1}^{\mu\mathsf{P}} &= 1 = 2a_{L+1}^{\mathsf{IP}} + c_{L+1}^{\mathsf{IP}} \end{aligned}$$

(recall that for $s \geq 1$, $c_1^{\mathsf{IP}} = c_{L+1}^{\mathsf{IP}} = -1$, and $c_l^{\mathsf{IP}} = -2$ for $l \in [2, L]$). We thus find $\Delta W_{\mathsf{HP}}^l(s) = \Delta W_{\mathsf{IP}}^l(s)$ for all l, and since $W_{\mathsf{HP}}^l(s) = W_{\mathsf{IP}}^l(s)$ by assumption, we get $W_{\mathsf{HP}}^l(s+1) = W_{\mathsf{IP}}^l(s+1)$ for all l which concludes the induction.

The effective weights being equal in both parameterizations for all time steps ≥ 1, we get that at time step $t \geq 1$, for any input $\xi \in \mathbb{R}^d$, the outputs $f_t^{\mathsf{HP}}(\xi)$ and $f_t^{\mathsf{IP}}(\xi)$ are the same, which concludes the proof. □

A.11.2 . Infinite-width equivalence: Theorem 2.4.2

In this section we prove Theorem 2.4.2 which states the equivalence between IP-LLR (see Definition 2.4.1) and HPZ (see Section 2.4.2). We start by a couple of preliminary results on the dynamics of HPZ, then proceed to prove the main induction step over t, and finally conclude by putting the results together to prove the theorem.

Preliminary results

Lemma A.11.2 (μP is zero at initialization)**.** *Consider the μP parameterization with an activation function satisfying Assumption 2 and a loss function ℓ satisfying*

Assumption 1, and no bias terms except at the first layer. Let $\xi \in \mathbb{R}^d$ be an input to the network. One has:

$$f_0(\xi) \xrightarrow[m\to\infty]{a.s.} 0$$
$$\chi_0 \xrightarrow[m\to\infty]{a.s.} \mathring{\chi}_0 := \partial_2 \ell(y_0, 0)$$

Remark. The result on the almost sure convergence of χ_0 ensures that the latter is a valid initial scalar in the Tensor Program defining the computations associated with μP (and thus and HPZ). Also note that the limit of χ_0 is the same as for IP-LLR (see Lemma A.8.1).

Proof. μP is designed so that $h_0^l = \tilde{h}_0^l$ and $x_0^l = \tilde{x}_0^l$ for any $l \in [2, L]$, and as already proved in Lemma A.8.1, the tilde variables are vectors in the Tensor Program. Since $f_0(\xi) = m^{-1}(U^{L+1})^\top \tilde{x}_0^L$ we get by the master theorem that $f_0(\xi)$ converges almost surely towards $\mathbb{E}[Z^{U^{L+1}} Z^{\tilde{x}_0^L}]$. By Lemma A.13.2, $Z^{\tilde{h}_0^L} = \widehat{Z}^{\tilde{h}_0^L}$ and $Z^{U^{L+1}} = \widehat{Z}^{U^{L+1}}$ by definition, and by the ZHat rule, $\widehat{Z}^{\tilde{h}_0^L}$ and $\widehat{Z}^{U^{L+1}}$ are independent, and since $\mathbb{E}[Z^{U^{L+1}}] = 0$ and $\mathbb{E}[(Z^{\tilde{x}_0^L})^2] < \infty$ we get that $f_0(\xi)$ converges almost surely towards 0. The result on the limit of χ_0 is then simply a consequence of the fact that $\partial_2 \ell(y_0, \cdot)$ is continuous by assumption. □

Lemma A.11.3 (Weight updates for μP at any time step)**.** *Consider the μP parameterization with a differentiable activation function σ and no bias terms except at the first layer, and let $t \geq 1$. Then, dropping the dependency of the forward and backward passes on ξ_t at time t, one has:*

$$\begin{aligned}
\Delta W^{L+1}(t+1) &= -\eta \chi_t x_t^L / m, \\
\Delta W^l(t+1) &= -\eta \chi_t \frac{d\tilde{h}_t^l (x_t^{l-1})^\top}{m}, \qquad l \in [2, L], \\
\Delta W^1(t+1) &= -\eta \chi_t d\tilde{h}_t^1 \xi_t^\top, \\
\Delta B^1(t+1) &= -\eta \chi_t d\tilde{h}_t^1.
\end{aligned}$$

Remark. Because HPZ and μP have the same parameterization for $t \geq 1$ (see Section 2.4.2), the formulas above for the updates are the same for HPZ, the only difference is that, at finite width, the x_t^l and $d\tilde{h}_t^l$ differ from HPZ to μP because $W_{\mathsf{HPZ}}^l(t) = W_{\mu\mathsf{P}}^l(t) - W_{\mu\mathsf{P}}^l(0)$. Note that the formulas are also exactly the same as for IP-LLR (see Lemma A.8.3) but again the quantities x_t^l and $d\tilde{h}_t^l$ differ for μP and IP-LLR because of the initial weight contribution in $W^l(t)$ which is different for the intermediate layers of both parameterizations.

Proof. By Equation (A.6), we have

$$\begin{aligned}
\Delta W^{L+1}(t+1) &= -\eta m^{-(2a_{L+1}+c_{L+1})} \chi_t x_t^L \\
&= -\eta \chi_t x_t^L / m,
\end{aligned}$$

because $2a_{L+1} + c_{L+1} = 2 - 1 = 1$ for μP. For $l \in [2, L]$, we have by Equation (A.3)

$$\begin{aligned}\Delta W^l(t+1) &= -\eta\chi_t m^{-(2a_l+c_l)} dh_t^l (x_t^{l-1})^\top \\ &= -\eta\chi_t \frac{d\tilde{h}_t^l (x_t^{l-1})^\top}{m},\end{aligned}$$

because $dh_t^l = m^{-1} d\tilde{h}_t^l$ and $2a_l + c_l = 1 - 1 = 0$ for μP. Finally, for $l = 1$ we have again by Equation (A.3)

$$\begin{aligned}\Delta W^1(t+1) &= -\eta\chi_t m^{-(2a_1+c_1)} dh_t^1 \xi_t^\top \\ &= -\eta\chi_t d\tilde{h}_t^1 \xi_t^\top,\end{aligned}$$

because $2a_1 + c_1 = -1$ for μP and $dh_t^1 = d\tilde{h}_t^1$. A similar argument holds for $\Delta B^1(t+1)$, which concludes the proof. □

Theorem A.11.4 (Weights in HPZ at time t)**.** *Consider the HPZ parameterization with a differentiable activation function σ and no bias terms except at the first layer. Then, for any $t \geq 1$, one has:*

(i) $W^1(t) = U^1 - \eta\chi_0 d\tilde{h}_0^1 \xi_0^\top - \eta\left(\sum_{s=1}^{t-1} \chi_s d\tilde{h}_s^1 \xi_s^\top\right)$,

(ii) $B^1(t) = v^1 - \eta\chi_0 d\tilde{h}_0^1 - \eta\left(\sum_{s=1}^{t-1} \chi_s d\tilde{h}_s^1\right)$,

(iii) $W^l(t) = -\eta\chi_0 \frac{d\tilde{h}_0^l (\tilde{x}_0^{l-1})^\top}{m} - \eta\left(\sum_{s=1}^{t-1} \chi_s \frac{d\tilde{h}_s^l (x_s^{l-1})^\top}{m}\right), \qquad l \in [2, L]$,

(iv) $W^{L+1}(t) = U^{L+1}/m - \eta\chi_0 \tilde{x}_0^L/m - \eta\left(\sum_{s=1}^{t-1} \chi_s x_s^L/m\right)$.

Proof. The formulas are correct at $t = 1$ by definition of HPZ and by Lemma A.7.5 which gives the first weight updates for μP. Then, an easy induction using Lemma A.11.3 yields the result. □

Lemma A.11.5 (Backward pass of HPZ at time t)**.** *Consider the HPZ parameterization with a differentiable activation function σ and no bias terms except at the first layer. Then, for any $t \geq 1$, dropping the dependency of the forward pass at time t on ξ_t, and of the previous forward and backward passes on the corresponding ξ_s, one has:*

(i) $d\tilde{x}_t^L = w^{L+1}(t) = U^{L+1} - \eta\chi_0 \tilde{x}_0^L - \eta\sum_{s=1}^{t-1} \chi_s x_s^L$,

(ii) $d\tilde{x}_t^{l-1} = -\eta\chi_0 \frac{(d\tilde{h}_0^l)^\top d\tilde{h}_t^l}{m} \tilde{x}_0^{l-1} - \eta\sum_{s=1}^{t-1} \chi_s \frac{(d\tilde{h}_s^l)^\top d\tilde{h}_t^l}{m} x_s^{l-1}, \qquad l \in [2, L]$.

Proof. By definition, we have

$$\begin{aligned} d\tilde{x}_t^L &= m dx_t^L \\ &= mW^{L+1}(t) \\ &= U^{L+1} - \eta\chi_0 \tilde{x}_0^L - \eta \sum_{s=1}^{t-1} \chi_s x_s^L \end{aligned}$$

where the last equality stems from Theorem A.11.4.

Let $l \in [2, L]$, we have:

$$\begin{aligned} d\tilde{x}_t^{l-1} &= (W^l(t))^\top d\tilde{h}_t^l \\ &= -\eta\chi_0 \frac{(d\tilde{h}_0^l)^\top d\tilde{h}_t^l}{m} \tilde{x}_0^{l-1} - \eta \sum_{s=1}^{t-1} \chi_s \frac{(d\tilde{h}_s^l)^\top d\tilde{h}_t^l}{m} x_s^{l-1} \end{aligned}$$

where the second equality stems from Theorem A.11.4. □

Lemma A.11.6 (Z for the forward pass of HPZ at time $t = 1$)**.** *Consider the HPZ parameterization with an activation function σ satisfying Assumption 2 and no bias terms except at the first layer. Let $\xi \in \mathbb{R}^d$ be an input to the network. Then, for any $l \in [1, L]$, $h_1^l(\xi), x_1^l(\xi), d\tilde{x}_1^l, d\tilde{h}_1^l$ are vectors in the program, $f_1(\xi)$ is a scalar in the program, and χ_1 is a valid initial scalar in the program. Additionally, dropping the dependency of the forward pass at time $t = 1$ on ξ, and of the first forward and backward passes on ξ_0, one has:*

(i) $Z^{h_1^1} = Z^{W^1(1)\xi + B^1(1)} = Z^{U^1\xi} + Z^{v^1} - \eta \mathring{\chi}_0 (\xi_0^\top \xi + 1) Z^{d\tilde{h}_0^1}$,

(ii) $Z^{h_1^l} = Z^{W^l(1)x_1^{l-1}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}] Z^{d\tilde{h}_0^l}, \qquad l \in [2, L]$,

(iii) $f_1(\xi) = (W^{L+1}(1))^\top x_1^L \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x_1^L}] - \eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_1^L}]$.

Proof. By Theorem A.11.4, with $t = 1$, one has that $h_1^1 = U^1\xi + v^1 - \eta\chi_0(\xi_0^\top \xi + 1) d\tilde{h}_0^1$. By Lemma A.8.1, $d\tilde{h}_0^1$ is a vector in the Tensor Program (recall that the tilde variables at initialization do not depend on the choice of parameterization) and by Lemma A.11.2 χ_0 is a valid initial scalar in the program which has an almost sure limit $\mathring{\chi}_0 := \partial_2 \ell(y_0, 0)$ as $m \to \infty$ (see Remark A.11.2). In addition, $U^1\xi$ and v^1 are initial vectors in the program, which thus shows that h_1^1 is a vector in the program by the `NonLin` operation. This also gives that $x_1^1 = \sigma(h_1^1)$ is a vector in the program since σ is pseudo-Lipschitz by assumption. Moreover, by `ZNonLin`, we have $Z^{h_1^1} = Z^{U^1\xi} + Z^{v^1} - \eta \mathring{\chi}_0 (\xi_0^\top \xi + 1) Z^{d\tilde{h}_0^1}$. Let $l \in [2, L]$ and assume that h_1^{l-1}, x_1^{l-1} are vectors in the program. Then, by Theorem A.11.4 with $t = 1$, we get

$$h_1^l = -\eta\chi_0 \frac{(\tilde{x}_0^{l-1})^\top x_1^{l-1}}{m} d\tilde{h}_0^l.$$

$(\tilde{x}_0^{l-1})^\top x_1^{l-1}/m$ is a scalar in the program by the `Moment` operation, and thus by the `MatMul` and `NonLin` operations, h_1^l is a vector in the program and thus so is $x_1^l = \sigma(h_1^l)$, which proves by induction that this is the case for any $l \in [2, L]$. By `ZNonLin` we thus have

$$Z^{h_1^l} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_1^{l-1}}] Z^{d\tilde{h}_0^l}.$$

We then have by Theorem A.11.4 with $t = 1$,

$$f_1(\xi) = m^{-1}(U^{L+1})^\top x_1^L - \eta\chi_0 \frac{(\tilde{x}_0^L)^\top x_1^L}{m}$$

$U^{L+1} - \eta\chi_0 \tilde{x}_0^L$ is a vector in the program by the `NonLin` operation, and the quantity $m^{-1}(U^{L+1} - \eta\chi_0\tilde{x}_0^L)^\top x_1^L$ is thus a scalar in the program by the `Moment` operation, and by the master theorem, we get the following convergence: $f_1(\xi) \to \mathbb{E}[Z^{U^{L+1}} Z^{x_1^L}] - \eta\mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_1^L}]$ almost surely, since both expectations are finite by Lemma A.13.1. Since we did the previous reasoning with an arbitrary ξ, we also get that $h_1^l(\xi_1), x_1^l(\xi_1)$ are vectors in the program for any $l \in [1, L]$ and that the formulas in (i), (ii), and (iii) hold when the input is ξ_1. In particular, $f_1(\xi_1)$ converges to a finite almost sure limit $\mathring{f}_1(\xi_1)$, and thus the continuity of $\partial_2\ell(y_1, \cdot)$ ensures the almost sure convergence of χ_1 towards $\mathring{\chi}_1 := \partial_2\ell(y_1, \mathring{f}_1(\xi_1))$, which means χ_1 is a valid initial scalar in the Tensor Program. Then, dropping the dependency of the second forward pass (at $t = 1$) on ξ_1, we get by Theorem A.11.5 with $t = 1$:

$$d\tilde{x}_1^L = U^{L+1} - \eta\chi_0\tilde{x}_0^L$$

which is a vector in the program by `NonLin`. Then $d\tilde{h}_1^L = d\tilde{x}_1^L \odot \sigma'(h_1^L)$ is also a vector in the program by `NonLin` since σ' is pseudo-Lipschitz. Let $l \in [2, L-1]$ and assume that $d\tilde{x}_1^{l+1}$ and $d\tilde{h}_1^{l+1}$ are vectors in the program. Then by Theorem A.11.5 with $t = 1$, we have

$$d\tilde{x}_1^l = -\eta\chi_0 \frac{(d\tilde{h}_0^{l+1})^\top d\tilde{h}_1^{l+1}}{m} \tilde{x}_0^l$$

$(d\tilde{h}_0^{l+1})^\top d\tilde{h}_1^{l+1}/m$ is a scalar in the program by the `Moment` operation and by `MatMul` and `NonLin` we thus get that $d\tilde{x}_1^l$ is a vector in the program. Then $d\tilde{h}_1^l = d\tilde{x}_1^l \odot \sigma'(h_1^l)$ is also a vector in the program since σ' is pseudo-Lipschitz, which concludes the induction and with it the proof. □

Lemma A.11.7 (Zs of HPZ and IP-LLR are equal at $t = 1$)**.** *Consider the HPZ and IP-LLR parameterization with an activation function σ satisfying Assumption 3, and no bias terms except at the first layer, and let us sub/super-script the variables of each models with HPZ and IP respectively. Let $\xi \in \mathbb{R}^d$ be an input to the networks, and assume that HPZ and IP-LLR share the same training samples (ξ_0, y_0) and*

(ξ_1, y_1) *at* $t = 0$ *and* $t = 1$*, the same loss function* ℓ *satisfying Assumption 1, and the same base learning rate* η*. Then dropping the dependency of the first forward and backward passes on* ξ_0 *and that of the second forward passes on* ξ*, we have:*

(i) $Z^{h^l_{1,\mathit{HPZ}}} = Z^{h^l_{1,\mathit{IP}}}, \quad Z^{x^l_{1,\mathit{HPZ}}} = Z^{x^l_{1,\mathit{IP}}}, \qquad l \in [1, L]$*,*

(ii) $\lim_{m\to\infty} f_1^{\mathit{HPZ}}(\xi) = \lim_{m\to\infty} f_1^{\mathit{IP}}(\xi)$*,*

(iii) $\mathring{\chi}_1^{\mathit{HPZ}} = \mathring{\chi}_1^{\mathit{IP}}$*,*

(iv) $Z^{d\tilde{x}^l_{1,\mathit{HPZ}}} = Z^{d\tilde{x}^l_{1,\mathit{IP}}}, \quad Z^{d\tilde{h}^l_{1,\mathit{HPZ}}} = Z^{d\tilde{h}^l_{1,\mathit{IP}}}, \qquad l \in [1, L]$*.*

Proof. By Lemmas A.8.1 and A.8.1 we have $\mathring{\chi}_0^{\mathsf{IP}} = \mathring{\chi}_0^{\mu\mathsf{P}} = \mathring{\chi}_0^{\mathsf{HPZ}} = \partial_2\ell(y_0, 0)$, which we simply call $\mathring{\chi}_0$ in the remainder of this proof for simplicity. By Corollary A.8.6 and Lemma A.11.6 we have

$$Z^{h^1_{1,\mathsf{IP}}} = Z^{U^1\xi} + Z^{v^1} - \eta\mathring{\chi}_0(\xi_0{}^\top\xi + 1)Z^{d\tilde{h}^1_0},$$

and

$$Z^{h^1_{1,\mathsf{HPZ}}} = Z^{U^1\xi} + Z^{v^1} - \eta\mathring{\chi}_0(\xi_0{}^\top\xi + 1)Z^{d\tilde{h}^1_0},$$

and since the tilde variables are computed independently of any parameterization, we have $Z^{h^1_{1,\mathsf{HPZ}}} = Z^{h^1_{1,\mathsf{IP}}}$. Because IP and HPZ share the same activation function we also get $Z^{x^1_{1,\mathsf{HPZ}}} = Z^{x^1_{1,\mathsf{IP}}}$. Now let $l \in [2, L]$ and assume $Z^{h^{l-1}_{1,\mathsf{HPZ}}} = Z^{h^{l-1}_{1,\mathsf{IP}}}$ as well as $Z^{x^{l-1}_{1,\mathsf{HPZ}}} = Z^{x^{l-1}_{1,\mathsf{IP}}}$. By Corollary A.8.6 and Lemma A.11.6 we have

$$Z^{h^l_{1,\mathsf{IP}}} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}^{l-1}_0} Z^{x^{l-1}_{1,\mathsf{IP}}}]Z^{d\tilde{h}^l_0},$$

and

$$Z^{h^l_{1,\mathsf{HPZ}}} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}^{l-1}_0} Z^{x^{l-1}_{1,\mathsf{HPZ}}}]Z^{d\tilde{h}^l_0},$$

which shows $Z^{h^l_{1,\mathsf{IP}}} = Z^{h^l_{1,\mathsf{HPZ}}}$ since the tilde variables are independent of any choice of parameterization. Since the activation function σ is the same for both models we also get $Z^{x^l_{1,\mathsf{IP}}} = Z^{x^l_{1,\mathsf{HPZ}}}$ which concludes the induction. For the output of the networks, we have by Corollary A.8.6 and Lemma A.11.6

$$f_1^{\mathsf{IP}}(\xi) \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x^L_{1,\mathsf{IP}}}] - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}^L_0} Z^{x^L_{1,\mathsf{IP}}}],$$

and

$$f_1^{\mathsf{HPZ}}(\xi) \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x^L_{HPZ}}] - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}^L_0} Z^{x^L_{1,\mathsf{HPZ}}}],$$

and since $Z^{x^L_{1,\mathsf{IP}}} = Z^{x^L_{1,\mathsf{HPZ}}}$ by the previous induction and the tilde variables are independent of the parameterization, it follows that $\lim_{m\to\infty} f_1^{\mathsf{HPZ}}(\xi) =$

$\lim_{m\to\infty} f_1^{\mathsf{IP}}(\xi) =: \mathring{f}_1(\xi)$. Since $\chi_1^{\mathsf{IP}} = \partial_2(y_1, f_1^{\mathsf{IP}}(\xi))$ and $\chi_1^{\mathsf{HPZ}} = \partial_2(y_1, f_1^{\mathsf{HPZ}}(\xi))$, by continuity of $\partial_2\ell(y_1,\cdot)$ we get that $\mathring{\chi}_1^{\mathsf{HPZ}} = \partial_2\ell(y_1, \mathring{f}_1(\xi)) = \mathring{\chi}_1^{\mathsf{IP}}$.

For the backward pass, we have by Lemma A.11.5 that $d\tilde{x}^L_{1,\mathsf{HPZ}} = U^{L+1} - \eta\chi_{0,\mathsf{HPZ}}\tilde{x}_0^L$ which gives by `NonLin` $Z^{d\tilde{x}^L_{1,\mathsf{HPZ}}} = ZU^{L+1} - \eta\mathring{\chi}_0 Z^{\tilde{x}_0^L}$ which is also equal to $Z^{d\tilde{x}^L_{1,\mathsf{IP}}}$ by Lemma A.8.5 since the tilde variables are independent of the choice of parameterization. Then, we also get $Z^{d\tilde{h}^L_{1,\mathsf{HPZ}}} = Z^{d\tilde{x}^L_{1,\mathsf{HPZ}}}\sigma'(Z^{h^L_{1,\mathsf{HPZ}}})$ and $Z^{d\tilde{h}^L_{1,\mathsf{IP}}} = Z^{d\tilde{x}^L_{1,\mathsf{HPZ}}}\sigma'(Z^{h^L_{1,\mathsf{IP}}})$ which shows $Z^{d\tilde{h}^L_{1,\mathsf{HPZ}}} = Z^{d\tilde{h}^L_{1,\mathsf{IP}}}$. Let $l \in [1, L-1]$ and assume $Z^{d\tilde{x}^{l+1}_{1,\mathsf{HPZ}}} = Z^{d\tilde{x}^{l+1}_{1,\mathsf{IP}}}$ as well as $Z^{d\tilde{h}^{l+1}_{1,\mathsf{HPZ}}} = Z^{d\tilde{h}^{l+1}_{1,\mathsf{IP}}}$. By Lemma A.11.5, we have

$$d\tilde{x}^l_{1,\mathsf{HPZ}} = -\eta\chi_0 \frac{(d\tilde{h}_0^{l+1})^\top d\tilde{h}^{l+1}_{1,\mathsf{HPZ}}}{m}\tilde{x}_0^l$$

which gives by the master theorem and the `ZNonLin`

$$Z^{d\tilde{x}^l_{1,\mathsf{HPZ}}} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{d\tilde{h}_0^{l+1}} Z^{d\tilde{h}^{l+1}_{1,\mathsf{HPZ}}}]Z^{\tilde{x}_0^l}$$

which is the same expression as $Z^{d\tilde{x}^l_{1,\mathsf{IP}}}$ by Lemma A.8.5. It then follows that $Z^{d\tilde{h}^l_{1,\mathsf{HPZ}}} = Z^{d\tilde{h}^l_{1,\mathsf{IP}}}$, which concludes the induction and with it the proof. □

Theorem A.11.8 (Z for the forward pass of HPZ at time t)**.** *Consider the HPZ parameterization with an activation function σ satisfying Assumption 2 and no bias terms except at the first layer. Let $\xi \in \mathbb{R}^d$ be an input to the network. Then, for any $l \in [1, L]$, $h^l_s(\xi), x^l_s(\xi), d\tilde{x}^l_s, d\tilde{h}^l_s$ are vectors in the program, $f_s(\xi)$ is a scalar in the program, and χ_s is a valid initial scalar in the program. Additionally, dropping the dependency of the forward pass at time t on ξ, and of the previous forward and backward passes on the corresponding ξ_s, one has:*

(i) $Z^{h^1_t} = Z^{W^1(t)\xi + B^1(t)} = Z^{U^1\xi} + Z^{v^1} - \eta\mathring{\chi}_0(\xi_0^\top\xi + 1)Z^{d\tilde{h}^1_0} - \eta\left(\sum_{s=1}^{t-1}\mathring{\chi}_s(\xi_s^\top\xi + 1)Z^{d\tilde{h}^1_s}\right)$,

(ii) $Z^{h^l_t} = Z^{W^l(t)x^{l-1}_t} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{x_t^{l-1}}]Z^{d\tilde{h}^l_0} - \eta\left(\sum_{s=1}^{t-1}\mathring{\chi}_s\mathbb{E}[Z^{x_s^{l-1}} Z^{x_t^{l-1}}]Z^{d\tilde{h}^l_s}\right), \quad l \in [2, L]$,

(iii) $f_t(\xi) = (W^{L+1}(t))^\top x_t^L \xrightarrow[m\to\infty]{a.s.} \mathbb{E}[Z^{U^{L+1}} Z^{x_t^L}] - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^L} Z^{x_t^L}] - \eta\left(\sum_{s=1}^{t-1}\mathring{\chi}_s\mathbb{E}[Z^{x_s^L} Z^{x_t^L}]\right)$.

Proof. The proof is exactly the same as for Theorem A.8.7 except that whenever a multiplication by $W^l(0)$ appears with $l \in [2, L]$, it is now replaced by 0, but the reasoning and all the arguments are the same, which in summary uses an induction over t as well as the master theorem and the `ZNonLin` rule from the Tensor Program. □

Theorem A.11.9 (Zs of backward pass of HPZ at time t)**.** *Consider the HPZ parameterization with an activation function σ satisfying Assumption 2 and no*

bias terms except at the first layer. Then, for any $t \geq 1$, dropping the dependency of the forward pass at time t on ξ_t, and of the previous forward and backward passes on the corresponding ξ_s, one has:

(i) $Z^{d\tilde{x}_t^L} = Z^{w^{L+1}(t)} = Z^{U^{L+1}} - \eta \mathring{\chi}_0 Z^{\tilde{x}_0^L} - \eta \sum_{s=1}^{t-1} \mathring{\chi}_s Z^{x_s^L}$,

(ii) $Z^{d\tilde{x}_t^{l-1}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_t^l}] Z^{\tilde{x}_0^{l-1}} - \eta \sum_{s=1}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{d\tilde{h}_s^l} Z^{d\tilde{h}_t^l}] Z^{x_s^{l-1}}, \quad l \in [2, L]$.

Proof. As for Theorem A.11.8, the proof follows exactly the same pattern as for Theorem A.11.9 except that whenever a multiplication by $W^l(0)$ appears with $l \in [2, L]$, it is now replaced by 0. □

Induction on t

Lemma A.11.10 (Induction step on the Zs of the forward pass)**.** *Consider the IP-LLR and HPZ parameterizations with an activation function σ satisfying Assumption 3 and no bias terms except at the first layer, and let us sub/super-script the variables of each models with IP and HP respectively. Let $s \geq 1$, $\xi \in \mathbb{R}^d$ be an input to the networks, and assume that the training routine (see Definition 2.2.2) is the same for both models with a loss satisfying Assumption 1. Assume further that, dropping the dependency of the forward and backward passes at time $t = r$ on ξ_r, for all $r \in [1, s]$, we have:*

(i) $Z^{h^l_{HPZ,r}} = Z^{h^l_{IP,r}}, \quad Z^{x^l_{HPZ,r}} = Z^{x^l_{IP,r}}, \qquad l \in [1, L]$,

(ii) $\lim_{m\to\infty} f_r^{HPZ}(\xi) = \lim_{m\to\infty} f_r^{IP}(\xi)$,

(iii) $\mathring{\chi}_r^{HPZ} = \mathring{\chi}_r^{IP}$,

(iv) $Z^{d\tilde{h}^l_{HPZ,r}} = Z^{d\tilde{h}^l_{IP,r}}, \quad Z^{d\tilde{x}^l_{HPZ,r}} = Z^{d\tilde{x}^l_{IP,r}}, \qquad l \in [1, L]$.

Then, dropping the dependency of the forward pass at time $t = s + 1$ on ξ, one has:

(v) $Z^{h^l_{HPZ,s+1}} = Z^{h^l_{IP,s+1}}, \quad Z^{x^l_{HPZ,s+1}} = Z^{x^l_{IP,s+1}}, \qquad l \in [1, L]$,

(vi) $\lim_{m\to\infty} f_{s+1}^{HPZ}(\xi) = \lim_{m\to\infty} f_{s+1}^{IP}(\xi)$,

(vii) $\mathring{\chi}_{s+1}^{HPZ} = \mathring{\chi}_{s+1}^{IP}$,

(viii) $Z^{d\tilde{h}^l_{HPZ,r}} = Z^{d\tilde{h}^l_{IP,s+1}}, \quad Z^{d\tilde{x}^l_{HPZ,s+1}} = Z^{d\tilde{x}^l_{IP,s+1}}, \qquad l \in [1, L]$.

Proof. Since by assumption, for any $r \in [1, s]$, the Zs of the forward and backward passes are equal for both parameterizations, we drop the dependency

of those quantities on the model, and for $z \in \{h_r^l, x_r^l, d\tilde{h}_r^l, d\tilde{x}_r^l\}$, we simply call $Z^{z_{\mathsf{HPZ}}} = Z^{z_{\mathsf{IP}}} = Z^z$. Similarly we simply call $\mathring{\chi}_r^{\mathsf{HPZ}} = \mathring{\chi}_r^{\mathsf{IP}} = \mathring{\chi}_r$. We have A.8.7

$$\begin{aligned} Z^{h^1_{\mathsf{HPZ},s+1}} &= Z^{U^1\xi_{s+1}} - \eta\mathring{\chi}_0(\xi_0^\top\xi_{s+1}+1)Z^{d\tilde{h}_0^1} - \eta\sum_{r=1}^{s}\mathring{\chi}_r(\xi_r^\top\xi_{s+1}+1)Z^{d\tilde{h}_r^1} \\ &= Z^{h^1_{\mathsf{IP},s+1}} \end{aligned}$$

where the first equality stems from Theorem A.11.8 and the second one from Theorem A.8.7. Since both parameterizations use the same linearity σ, we get $Z^{x^1_{\mathsf{HPZ},s+1}} = \sigma(Z^{h^1_{\mathsf{HPZ},s+1}}) = \sigma(Z^{h^1_{\mathsf{IP},s+1}}) = Z^{x^1_{\mathsf{IP},s+1}}$.

Let $l \in [2, L]$ and assume $Z^{h^{l-1}_{\mathsf{HPZ},s+1}} = Z^{h^{l-1}_{\mathsf{IP},s+1}}$, $Z^{x^{l-1}_{\mathsf{HPZ},s+1}} = Z^{x^{l-1}_{\mathsf{IP},s+1}}$. By Theorem A.11.8, we have

$$\begin{aligned} Z^{h^l_{\mathsf{HPZ},s+1}} &= -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^{l-1}}Z^{x^{l-1}_{\mathsf{HPZ},s+1}}]Z^{d\tilde{h}_0^l} - \eta\sum_{r=1}^{s}\mathring{\chi}_r\mathbb{E}[Z^{x_r^{l-1}}Z^{x^{l-1}_{\mathsf{HPZ},s+1}}]Z^{d\tilde{h}_r^l} \\ &= -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^{l-1}}Z^{x^{l-1}_{\mathsf{IP},s+1}}]Z^{d\tilde{h}_0^l} - \eta\sum_{r=1}^{s}\mathring{\chi}_r\mathbb{E}[Z^{x_r^{l-1}}Z^{x^{l-1}_{\mathsf{IP},s+1}}]Z^{d\tilde{h}_r^l} \\ &= Z^{h^l_{\mathsf{HPZ},s+1}} \end{aligned}$$

where the last equality stems from Theorem A.8.7. Since both parameterizations used the same non-linearity σ, we get $Z^{x^{l+1}_{\mathsf{HPZ},s+1}} = Z^{x^{l+1}_{\mathsf{IP},s+1}}$.

By induction, we thus get that for any $l \in [1, L]$, $Z^{h^l_{\mathsf{HPZ},s+1}} = Z^{h^l_{\mathsf{IP},s+1}}$, and $Z^{x^l_{\mathsf{HPZ},s+1}} = Z^{x^l_{\mathsf{IP},s+1}}$, which proves (v). We can thus drop the dependency of h^l_{s+1} and x^l_{s+1} on the model HPZ or IP. Now, we thus have by Theorem A.11.8

$$\begin{aligned} \lim_{m\to\infty} f^{\mathsf{HPZ}}_{s+1}(\xi) &= \mathbb{E}[Z^{U^{L+1}}Z^{x^L_{s+1}}] - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^L}Z^{x^L_{s+1}}] - \eta\left(\sum_{r=1}^{s}\mathring{\chi}_r\mathbb{E}[Z^{x_r^L}Z^{x^L_{s+1}}]\right) \\ &= \lim_{m\to\infty} f^{\mathsf{IP}}_{s+1}(\xi) \end{aligned}$$

where the last equality stems from Theorem A.8.7, which proves (vi). Then (vi) combined with the continuity of $\partial_2\ell(y_{s+1}, \cdot)$ proves (vii), and we can thus imply denote $\mathring{\chi}^{\mathsf{IP}}_{s+1} = \mathring{\chi}^{\mathsf{HPZ}}_{s+1} = \mathring{\chi}_{s+1}$. By Theorems A.11.9 and A.8.8 we get $Z^{d\tilde{x}^L_{s+1,\mathsf{HPZ}}} = Z^{d\tilde{x}^L_{s+1,\mathsf{IP}}}$, from which it follows that $Z^{d\tilde{h}^L_{s+1,\mathsf{HPZ}}} = Z^{d\tilde{h}^L_{s+1,\mathsf{IP}}}$ by (v) and since both models share the same activation function. Finally, given the previous result, with $(i), (iii), (iv), (v)$ and (vii), an easy induction gives $(viii)$ with the formulas of Theorems A.11.9 and A.8.8, which concludes the proof. □

Proof of Theorem 2.4.2

Proof. The claim has already been proved at $t = 0$ by Lemmas A.8.1 and A.11.2, and at $t = 1$ by Lemma A.11.7. Then, by Lemma A.11.10, we get the result at any time step $t \geq 1$ by induction. □

A.12 . Formal versions of the results for the alternative methods of Section 2.5

A.12.1 . Formalization of the degeneracy of Section 2.5.2

Theorem A.12.1 (Formal)**.** *Consider the IP-bias parameterization as in Equations* (2.6)*, with the initial learning rates* $c_1 = -(L+1)/2$, $c_l = -(L-l+4)/2$ *for* $l \in [2, L]$*, and* $c_{L+1} = -1$ *for the weights, and* $\epsilon_1 = c_1 = -(L+1)/2$, $\epsilon_l = -(L-l+2)/2$ *for* $l \in [2, L]$*, and* $\epsilon_{L+1} = 0$*. for the bias terms. Assume the activation function* σ *satisfies Assumption 2 and the loss* ℓ *satisfies Assumption 1. Then, for any input* $\xi \in \mathbb{R}^d$ *to the network,* $Z^{h_0^l(\xi)}, Z^{x_0^l(\xi)}$ *for* $l \geq 2$*, and* $\lim_{m\to\infty} f_0(\xi)$ *do not depend on* ξ*. In addition, for any vector* x *in the program such that* Z^x *does not depend on on the first training input* ξ_0*,* $Z^{\Delta W^l(1)x}$ *for* $l \in [3, L]$*, and* $\lim_{m\to\infty} (\Delta W^{L+1}(1))^\top x$ *do not depend on* ξ_0*.*

Proof. We have $h_0^1 = U^1\xi + v^1$ so that $Z^{h_0^1} = \widehat{Z}^{U^1\xi} + \widehat{Z}^{v^1} \sim \mathcal{N}(0, ||\xi||^2+1)$, and $Z^{x_0^1} = \sigma(Z^{h_0^1})$. At the second layer $l = 2$, we have $h_0^2 = m^{-1/2}\widehat{W}^2 x_0^1 + v^2$ so that by `ZNonLin` $Z^{h_0^2} = 0 \times \widehat{Z}^{\widehat{W}^2 x_0^1} + Z^{v^2}$, and $\widehat{Z}^{\widehat{W}^2 x_0^1} \sim \mathcal{N}(0, \mathbb{E}[(Z^{x_0^1})^2])$. Because σ is pseudo-Lipschitz, it is also polynomially bounded, and the variance of the Gaussian is finite by Lemma A.3.1, so that $Z^{h_0^2} = Z^{v^2} \sim \mathcal{N}(0, 1)$ which does not depend on ξ. Therefore, $Z^{x_0^2} = \sigma(Z^{h_0^2})$ also does not depend on ξ. Let $l \in [3, L]$ and assume that $Z^{h_0^{l-1}} = Z^{v^{l-1}}$ and $Z^{x_0^{l-1}} = \sigma(Z^{v^{l-1}})$. Then, we have $Z^{h_0^l} = 0 \times \widehat{Z}^{\widehat{W}^l x_0^{l-1}} + Z^{v^l}$, and $\widehat{Z}^{\widehat{W}^l x_0^{l-1}} \sim \mathcal{N}(0, \mathbb{E}[(Z^{x_0^{l-1}})^2])$, and the variance is again finite by the same arguments as for $l = 2$. We thus get $Z^{h_0^l} = Z^{v^l}$ and $Z^{x_0^l} = \sigma(Z^{h_0^l}) = \sigma(Z^{v^l})$ which concludes the induction and shows that $Z^{h_0^l}$ and $Z^{x_0^l}$ do not depend on ξ for all intermediate layers l.

For the output of the network, it directly follows from the master theorem that $m^{-1}(U^{L+1})^\top x_0^L$ converges almost surely to $\mathbb{E}[Z^{U^{L+1}}\sigma(Z^{v^L})] = 0$ since $Z^{U^{L+1}}$ has mean 0 and is independent of Z^{v^L}. Since $f_0(\xi) = m^{-1}(U^{L+1})^\top x_0^L + v^{L+1}$ where $v^{L+1} \sim \mathcal{N}(0, 1)$, we have that $f_0(\xi)$ converges almost surely to the Gaussian variable v^{L+1} which does not depend on ξ. For the backward pass, recall the following definitions: $d\tilde{x}_0^l := m^{-1}m^{-(L-l)/2}\nabla_{x^l} f_0(\xi_0)$ and $d\tilde{h}_0^l := m^{-1}m^{-(L-l)/2}\nabla_{h^l} f_0(\xi_0)$. Then, we have $d\tilde{x}_0^L = U^{L+1}$, $d\tilde{h}_0^L = U^{L+1} \odot \sigma'(h_0^L)$, and a simple induction shows that for any $l \in [1, L-1]$, $d\tilde{x}_0^l = (\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}$, $d\tilde{h}_0^l = d\tilde{x}_0^l \odot \sigma'(h_0^l)$. We thus have $Z^{d\tilde{x}_0^L} = Z^{U^{L+1}}$ and $Z^{d\tilde{h}_0^L} = Z^{U^{L+1}}\sigma'(Z^{v^L})$ which does not depend on the first training input ξ_0. With the recursive formulas above, and since $Z^{h_0^l} = Z^{v^l}$ for $l \in [2, L]$, it is clear that $Z^{d\tilde{x}_0^l}$ and $Z^{d\tilde{h}_0^l}$

do not depend on ξ_0 for $l \in [2, L]$.

Finally, let x be a vector in the program for which Z^x does not depend on ξ, and let $l \in [3, L]$. Then, by design, with the initial learning rates as described above for the weights with IP-bias, we have

$$\Delta W^l(1)x = -\eta\chi_0 \frac{(x_0^{l-1})^\top x}{m} d\tilde{h}_0^l,$$

so that by `ZNonLin`

$$Z^{\Delta W^l(1)x} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{x_1^{l-1}} Z^x] Z^{d\tilde{h}_0^l},$$

where $\mathring{\chi}_0 := \partial_2 \ell(y_0, v^{L+1})$. Since v^{L+1}, $Z^{x_0^{l-1}}$, Z^x and $Z^{d\tilde{h}_0^l}$ do not depend on ξ_0 ($l-1$ and l are both in $[2, L]$), $Z^{\Delta W^l(1)x}$ also does not depend on the first training input ξ_0. To conclude, we have by the master theorem that $(\Delta W^{L+1}(1))^\top x$ converges almost surely towards $-\eta\mathring{\chi}_0 \mathbb{E}[Z^{x_0^L} Z^x]$ which is does not depend on ξ_0 since this is the case for v^{L+1}, $Z^{x_0^L}$ and Z^x, which concludes the proof.

□

A.12.2 . Formal version of Theorem 2.5.1

Theorem A.12.2 (Formal)**.** *Consider IP-non-centered with the Naive-IP learning rates at every time step. Assume the activation function σ satisfies Assumption 2 and the loss ℓ satisfies Assumption 1, and let $t \geq 0$ and $\xi \in \mathbb{R}^d$ be an input to the network. Then, calling $d\tilde{x}_s^l := m\nabla_{x^l} f_s(\xi_s)$ and $d\tilde{h}_s^l := m\nabla_{h^l} f_s(\xi_s)$, one has that:*

(i) for any $l \in [2, L-1]$, $Z^{h_t^l}$ and $Z^{x_t^l}$ are deterministic constants,

(ii) for any $l \in [2, L-1]$, $Z^{d\tilde{x}_t^l}$ and $Z^{d\tilde{h}_t^l}$ deterministic constants,

(iii) for any $l \in [3, L-1]$, and for any vector x in the program, we have that $Z^{(W^l(t+1)-W^l(0))x} = \left(-\eta \sum_{s=0}^t \mathring{\chi}_s Z^{d\tilde{h}_s^l} Z^{x_s^{l-1}}\right) \mathbb{E}[Z^x]$.

Remark. Point (iii) highlights the fact that in the infinite-width limit the (random) matrix operator $(w^l(t) - w^l(0))$ acts on a vector x as if all the entries of the matrix operator were equal to a single deterministic constant which reads as $\left(-\eta \sum_{s=0}^{t-1} \mathring{\chi}_s Z^{d\tilde{h}_s^l} Z^{x_s^{l-1}}\right)$, because then the averages over the coordinates of x involved in $(W^l(t) - W^l(0))x$ would simply yield $\mathbb{E}[Z^x]$ by the master theorem of the Tensor Program.

The proof Theorem A.12.2 can be found in Appendix A.12.2. The proof is done by inducting over t, and we present the case $t = 0$ and the induction step first in Appendix A.12.2.

Preliminaries

Lemma A.12.3 (First forward-backward pass and weight updates)**.** *Claims* (i)*,* (ii) *and* (iii) *of Theorem A.12.2 hold at* $t = 0$*.*

Proof. h_0^1 and $x_0^1 = \sigma(h_0^1)$ are vectors in the program by the `MatMul` and `NonLin` rules since σ is pseudo-Lipschitz by assumption, and $Z^{h_0^1} = Z^{U^1\xi} + Z^{v^1} \sim \mathcal{N}(0, ||\xi||^2+1)$, and finally $Z^{x_0^1} = \sigma(Z^{h_0^1})$. Now, we have (recall that as defined in Section 2.5.1 J is the matrix full of ones)

$$h_0^2 = m^{-1/2}\widehat{W}^2 x_0^1 + m^{-1}v^2 + u_2 m^{-1} J x_0^1.$$

$m^{-1/2}\widehat{W}^2 x_0^1 + m^{-1}v^2$ is a valid vector in the program by `MatMul` and `NonLin` because the initial scalars $m^{-1/2}$ and m^{-1} converge to 0 almost surely, and $Z^{m^{-1/2}\widehat{W}^2 x_0^1 + m^{-1}v^2} = 0 \times \widehat{Z}^{\widehat{W}^2 x_0^1} + 0 \times \widehat{Z}^{v^2}$. By the `ZHat` rule we get that $\widehat{Z}^{\widehat{W}^2 x_0^1} \sim \mathcal{N}(0, \mathbb{E}[(Z^{x_0^1})^2])$, with finite variance by Lemma A.3.1 since σ is pseudo-Lipschitz and thus polynomially bounded, and $\widehat{Z}^{v^2} \sim \mathcal{N}(0,1)$. It thus follows that get $Z^{m^{-1/2}(\widehat{W}^2 x_0^1 + v^2)} = 0$. On the other hand, $\theta := (1/m)\sum_{q=1}^m x_{0,q}^1$ is a valid scalar in the program by the `Moment` rule and it converges almost surely to $\mathring{\theta} = \mathbb{E}[Z^{x_0^1}]$ by the master theorem. The coordinates of $u_2 m^{-1} J x_0^1$ are thus all equal to $u_2\theta$, and the vector $u_2 m^{-1} J x_0^1$ is thus equal to $\psi(x_0^1; \theta)$ coordinate-wise where the function $\psi(\cdot;\cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is pseudo-Lipschitz and depends **only** on the second variable with $\psi(x;\alpha) = u_2\alpha$. By the `NonLin` rule $u_2 m^{-1} J x_0^1$ is thus a vector in the program and by `ZNonLin` we thus get $Z^{u_2 m^{-1} J x_0^1} = \psi(Z^{x_0^1}; \mathring{\theta}) = u_2 \mathbb{E}[Z^{x_0^1}]$. We thus finally get

$$Z^{h_0^2} = u_2 \mathbb{E}[Z^{x_0^1}],$$

which is a (finite) deterministic constant. Then the same statement holds for $Z^{x_0^2} = \sigma(u_2 \mathbb{E}[Z^{x_0^1}])$. Let $l \in [3, L]$ and assume that h_0^{l-1} and x_0^{l-1} are vectors in the program and that $Z^{h_0^{l-1}}$ and $Z^{x_0^{l-1}}$ are deterministic constants. Then, we have

$$h_0^l = m^{-1/2}\widehat{W}^l x_0^{l-1} + m^{-1}v^l + u_l m^{-1} J x_0^{l-1}$$

As for the case $l = 2$, we get that $m^{-1/2}\widehat{W}^l x_0^{l-1} + m^{-1}v^l$ is a vector in the program with $Z^{m^{-1/2}\widehat{W}^l x_0^{l-1} + m^{-1}v^l} = 0$, and $u_l m^{-1} J x_0^{l-1} = \psi(x_0^{l-1}; \theta)$ is a vector in the program with $\psi(z;\alpha) = u_l\alpha$ (recall that ψ is taken coordinate-wise) depending only on the second variable and $\theta := (1/m)\sum_{q=1}^m x_{0,q}^{l-1}$ is a valid scalar in the program by the `Moment` rule, which, by the master theorem, converges almost surely towards $\mathring{\theta} = \mathbb{E}[Z^{x_0^{l-1}}] = Z^{x_0^{l-1}}$ since the latter is a deterministic constant by the induction hypothesis. By `NonLin` h_0^l is a vector in the program and by `ZNonLin` $Z^{h_0^l} = \psi(Z^{x_0^{l-1}}; \mathring{\theta}) = u_l Z^{x_0^{l-1}}$ which is a deterministic

constant. The same claim holds for $Z^{x_0^l} = \sigma(u_l Z^{x_0^{l-1}})$, which concludes the induction for the forward pass. For the backward pass we get $d\tilde{x}_0^L = w^{L+1}(0) = U^{L+1} + u_{L+1}\mathbf{1}$ so that by `ZNonLin` $Z^{d\tilde{x}_0^L} = Z^{U^{L+1}} + u_{L+1} \sim \mathcal{N}(u_{L+1}, 1)$ since u_{L+1} is a valid initial scalar in the program as it converges almost surely to u_{L+1}. We then have $Z^{d\tilde{h}_0^L} = Z^{d\tilde{x}_0^L}\sigma'(Z^{h_0^L})$. Note that both $Z^{d\tilde{x}_0^L}$ and $Z^{d\tilde{h}_0^L}$ are not deterministic constants because U^{L+1} is Gaussian with variance 1. We then have:

$$d\tilde{x}_0^{L-1} = m^{-1/2}(\widehat{W}^L)^\top d\tilde{h}_0^L + u_L m^{-1} J^\top d\tilde{h}_0^L$$

As usual the first term $m^{-1/2}(\widehat{W}^L)^\top d\tilde{h}_0^L$ is a vector in the program by `MatMul` and `NonLin` and $Z^{m^{-1/2}(\widehat{W}^L)^\top d\tilde{h}_0^L} = 0$. For the second term, since $J^\top = J$, $m^{-1}J^\top d\tilde{h}_0^L$ is also a vector in the program and $Z^{m^{-1}J^\top d\tilde{h}_0^L} = u_L \mathbb{E}[Z^{d\tilde{h}_0^L}]$. We thus get that $d\tilde{x}_0^{L-1}$ is a vector in the program with $Z^{d\tilde{x}_0^{L-1}} = u_L \mathbb{E}[Z^{d\tilde{h}_0^L}]$ which is a deterministic constant. Then, $d\tilde{h}_0^{l-1}$ is also a vector in the program and by `ZNonLin` $Z^{d\tilde{h}_0^{L-1}} = Z^{d\tilde{x}_0^{L-1}}\sigma'(Z^{h_0^L})$ is a deterministic constant. Repeating the reasoning above at any layer $l \in [2, L-1]$, an easy induction (as in the forward pass) shows that $d\tilde{x}_0^l$ and $d\tilde{h}_0^l$ are vectors in the program and that $Z^{d\tilde{x}_0^l}$ and $Z^{d\tilde{h}_0^l}$ are deterministic constants. Note that $Z^{d\tilde{x}_0^1} = u_2\mathbb{E}[Z^{d\tilde{h}_0^2}] = u_2 Z^{d\tilde{h}_0^2}$ is also a deterministic constant but that $Z^{d\tilde{h}_0^1} = Z^{d\tilde{x}_0^1}\sigma'(Z^{h_0^1})$ is not because $Z^{h_0^1} \sim \mathcal{N}(0, ||\xi||^2+1)$. Let $l \in [3, L-1]$, and let x be a vector in the program. With the Naive-IP learning rates, we have

$$\Delta W^l(1) = -\eta\chi_0 \frac{d\tilde{h}_0^l (x_0^{l-1})^\top}{m}$$

Since $l \in [3, L-1]$, $Z^{d\tilde{h}_0^l}$ is a deterministic constant, and since $l-1 \in [2, L-2]$, $Z^{x_0^{l-1}}$ is also a deterministic constant. By `ZNonLin` and `ZMoment`we get

$$\begin{aligned} Z^{\Delta W^l(1)x} &= -\eta\mathring{\chi}_0 \mathbb{E}[Z^{x_0^{l-1}} Z^x] Z^{d\tilde{h}_0^l} \\ &= -\eta\mathring{\chi}_0 Z^{d\tilde{h}_0^l} Z^{x_0^{l-1}} \mathbb{E}[Z^x] \end{aligned}$$

which concludes the proof. Note that χ_0 is a valid initial scalar in the program because $f_0(\xi_0) = m^{-1}(U^{L+1})^\top x_0^L + u_{L+1}m^{-1}\mathbf{1}^\top x_0^L$ converges almost surely, by the master theorem, to $\mathbb{E}[Z^{U^{L+1}} Z^{x_0^L}] + u_{L+1}\mathbb{E}[Z^{x_0^L}] = u_{L+1}Z^{x_0^L}$ since $Z^{x_0^L}$ is a deterministic constant and $Z^{U^{L+1}} \sim \mathcal{N}(0,1)$ has mean zero. Since $\partial_2\ell(y_0, \cdot)$ is continuous by assumption, χ_0 converges almost surely towards $\mathring{\chi}_0 := \partial_2\ell(y_0, u_{L+1}Z^{x_0^L})$. □

Lemma A.12.4 (Induction step at time $t \geq 1$)**.** *Let $t \geq 1$ and assume claims (i), (ii) and (iii) of Theorem A.12.2 hold at all time steps $s \in [0, t-1]$. Then claims (i), (ii) and (iii) also hold at time step t.*

Proof. With the Naive-IP learning rate exponents, we get that for any $t \geq 1$,

$$
\begin{aligned}
W^1(t) &= U^1 - \eta \sum_{s=0}^{t-1} \chi_s d\tilde{h}_s^1 \xi_s^\top, \\
B^1(t) &= v^1 - \eta \sum_{s=0}^{t-1} \chi_s d\tilde{h}_s^1, \\
W^l(t) &= m^{-1}(U^l + u_l J) - \eta \sum_{s=0}^{t-1} \chi_s \frac{d\tilde{h}_s^l (x_s^{l-1})^\top}{m}, && l \in [2, L], \\
B^l(t) &= m^{-1} v^l - \eta m^{-1} \sum_{s=0}^{t-1} \chi_s d\tilde{h}_s^l, && l \in [2, L], \\
W^{L+1}(t) &= m^{-1}(U^{L+1} + u_{L+1}) - \eta \sum_{s=0}^{t-1} \chi_s \frac{x_s^L}{m}, \\
B^{L+1}(t) &= m^{-1} v^{L+1} - \eta m^{-1} \sum_{s=0}^{t-1} \chi_s.
\end{aligned}
$$

By a simple induction, all the h_s^l, x_s^l and $d\tilde{x}_s^l, d\tilde{h}_s^l$ are part of and the scalars χ_s are valid scalars in the program which have a constant almost sure limit, and by `ZNonLin` we get:

$$
Z^{h_t^1} = Z^{U^1 \xi} + Z^{v^1} - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s (\xi_s^\top \xi + 1) Z^{d\tilde{h}_s^1}
$$

and $Z^{x_t^1} = \sigma(Z^{h_t^1})$ is not a deterministic constant because $Z^{U^1 \xi} + Z^{v^1} \sim \mathcal{N}(0, ||\xi||^2 + 1)$. Let $l \in [2, L-1]$. We have

$$
\begin{aligned}
Z^{h_t^l} &= 0 \times Z^{\widehat{W}^l x_t^{l-1}} + 0 \times Z^{v^l} + u_l \mathbb{E}[Z^{x_t^{l-1}}] - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^{l-1}} Z^{x_t^{l-1}}] Z^{d\tilde{h}_s^l} \\
&= u_l \mathbb{E}[Z^{x_t^{l-1}}] - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^{l-1}} Z^{x_t^{l-1}}] Z^{d\tilde{h}_s^l},
\end{aligned}
$$

which is a deterministic constant with the assumption on the $Z^{d\tilde{h}_s^l}$ since $l \in [2, L-1]$. Note that if $l \in [3, L-1]$, we even have that the expectations simplify and we get $Z^{h_t^l} = (u_l - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s Z^{x_s^{l-1}} Z^{d\tilde{h}_s^l}) Z^{x_t^{l-1}}$. In any case, $Z^{x_t^l} = \sigma(Z^{h_t^l})$ is also a deterministic constant. For the output of the network, we have

$$
f_t(\xi) = \frac{(U^{L+1})^\top x_t^L}{m} + u_{L+1} \frac{\mathbf{1}^\top x_t^L}{m} + m^{-1}(v^{L+1} - \eta \sum_{s=0}^{t-1} \chi_s) - \eta \sum_{s=0}^{t-1} \chi_s \frac{(x_s^L)^\top x_t^L}{m}
$$

so that even if the x_s^L are not deterministic, $f_t(\xi)$ still converges almost surely, by the master theorem, to $\mathbb{E}[(Z^{U^{L+1}} + u_{L+1}) Z^{x_t^L}] - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^L} Z^{x_t^L}]$, and

since $\partial_2 \ell(y_t, \cdot)$ is continuous by assumption, χ_t converges almost surely towards the constant $\partial_2 \ell(y_t, \mathbb{E}[(Z^{U^{L+1}} + u_{L+1}) Z^{x_t^L}] - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{x_s^L} Z^{x_t^L}])$. For the backward pass, we get:

$$Z^{d\tilde{x}_t^L} = Z^{w^{L+1}(t)} = Z^{U^{L+1}} + u_{L+1} - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s Z^{x_s^L}$$

and $Z^{d\tilde{h}_t^L} = Z^{d\tilde{x}_t^L} \sigma'(Z^{h_t^L})$. Let $l \in [2, L-1]$, we have

$$\begin{aligned} d\tilde{x}_t^l &= (W^{l+1}(t))^\top d\tilde{h}_t^{l+1} \\ &= m^{-1/2} (\widehat{W}^{l+1})^\top d\tilde{h}_t^{l+1} + m^{-1} u_{l+1} J d\tilde{h}_t^{l+1} - \eta \sum_{s=0}^{t-1} \chi_s \frac{(d\tilde{h}_s^{l+1})^\top d\tilde{h}_t^{l+1}}{m} x_s^l, \end{aligned}$$

so that by `ZNonLin` we get

$$Z^{d\tilde{x}_t^l} = u_{l+1} \mathbb{E}[Z^{d\tilde{h}_t^{l+1}}] - \eta \sum_{s=0}^{t-1} \mathring{\chi}_s \mathbb{E}[Z^{d\tilde{h}_s^{l+1}} Z^{d\tilde{h}_t^{l+1}}] Z^{x_s^l},$$

and since $l \in [2, L-1]$, $Z^{x_s^l}$ is a deterministic constant and thus so is $Z^{d\tilde{x}_t^l}$. Then, $Z^{d\tilde{h}_t^l} = Z^{d\tilde{x}_t^l} \sigma'(Z^{h_t^l})$ and since $l \in [2, L-1]$, $Z^{h_t^l}$ is a deterministic constant. Finally, let $l \in [3, L-1]$, and let x be a vector in the program. We have

$$(W^l(t+1) - W^l(0))x = -\eta \sum_{s=0}^{t} \chi_s \frac{(x_s^{l-1})^\top x}{m} d\tilde{h}_s^l,$$

and by `ZNonLin`

$$\begin{aligned} Z^{(W^l(t+1) - W^l(0))x} &= -\eta \sum_{s=0}^{t} \mathring{\chi}_s \mathbb{E}[Z^{x_s^{l-1}} Z^x] Z^{d\tilde{h}_s^l} \\ &= \left(-\eta \sum_{s=0}^{t} \mathring{\chi}_s Z^{d\tilde{h}_s^l} Z^{x_s^{l-1}} \right) \mathbb{E}[Z^x], \end{aligned}$$

where the last equality stems from the fact that since $l \in [3, L-1]$, $l-1 \in [2, L-2]$ and $Z^{x_s^{l-1}}$ is a deterministic constant for any $s \in [0, t]$. Since $l \in [2, L-1]$, $Z^{d\tilde{h}_s^l}$ is also a deterministic constant, so that $-\eta \sum_{s=0}^{t} \mathring{\chi}_s Z^{d\tilde{h}_s^l} Z^{x_s^{l-1}}$ is a deterministic constant, which concludes the proof. □

Proof of Theorem A.12.2

Proof. The result comes by induction over t using Lemmas A.12.3 and A.12.4. □

A.13 . The variables associated with the initial weights vanish in IP-LLR

In this section we wish to study more precisely the evolution and the expression of the variables Z in the dynamics of IP-LLR at any time step t. To this end, we will show that the Zs of all the forward and backward variables in IP-LLR are functions only of the $\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}$ and $\widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}$, as well as the initial vectors $U^1\xi_0, \dots, U^1\xi_t$, v^1, U^{L+1}. We will thus write

$$Z^z = \psi\left(\left(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}\right)_l, \left(\widehat{Z}^{(\widehat{W}^k)^\top d\tilde{h}_0^k}\right)_k, \left(U^1\xi_s\right)_s, v^1, U^{L+1}\right)$$

to **generically** denote that the variable Z^z is a function **only** of the variables which appear in the arguments: $\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}$, $\widehat{Z}^{(\widehat{W}^k)^\top d\tilde{h}_0^k}$, $U^1\xi_s$, v^1, and U^{L+1}, (where multiple values of l, k and s might actually appear in the argument). This function ψ (we will sometimes also use ϕ) will of course depend on the z under consideration, and we might denote it by ψ^z (or ϕ^z,) but most of the time we will omit this dependency and simply use the symbol ψ for different variables to express that the variable Z^z is a function of the arguments of ψ only.

We will see that the function ψ appearing will always be polynomially bounded by some form of composition or product of polynomially bounded functions, which will allow us to prove that the corresponding Z^z is finite almost surely since its arguments, considered as a vector, follow a Gaussian distribution with finite variance (and thus finite moments of any order). Note that in the proofs, we will use extensively (without explicitly saying so) that if ϕ and ψ are polynomially bounded then $\phi \times \psi$ is also polynomially bounded, and if φ is a polynomially bounded function of a single variable then $\varphi \circ \psi$ is also polynomially bounded. We introduce the following definition and lemma which we will use extensively in the proof by induction:

Definition A.13.1 (Vector of initial vectors and first forward-backward)**.** Let $t \geq 1$. Then, dropping the dependency on t, we define the random vector:

$$\begin{aligned} Z_0 = Z_{0,t} := \Big(&\widehat{Z}^{U^1\xi_0}, \dots, \widehat{Z}^{U^1\xi_t}, Z^{v^1}, \widehat{Z}^{U^{L+1}}, \\ &\widehat{Z}^{\widehat{W}^2 \tilde{x}_0^1}, \dots, \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}, \\ &\widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}, \dots, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L}\Big) \end{aligned}$$

Remark.

1. Note that any function of $Z_{0,s}$ will also be a function of $Z_{0,t}$ for $t \geq s$, which is also why we suppress the dependency on t as we can always take the largest possible t when we make a specific claim which involves Z_0.

2. Also note that by the ZDot rule of the Tensor Program, for any vector z in the Tensor Program such that Z^z is a function only of Z_0, then for any $l \in [2, L]$:

$$\begin{cases} \dot{Z}^{\widehat{W}^l z} = \mathbb{E}\left[\frac{\partial Z^z}{\partial \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}}\right] Z^{d\tilde{h}_0^l} \\ \\ \dot{Z}^{(\widehat{W}^l)^\top z} = \mathbb{E}\left[\frac{\partial Z^z}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}}\right] Z^{\tilde{x}_0^{l-1}} \end{cases}$$

Lemma A.13.1 (Distribution of Z_0 and moments)**.** *One has*

(i) $Z_0 \sim \mathcal{N}\left(0, \begin{pmatrix} S & 0 & 0 \\ 0 & D_f & 0 \\ 0 & 0 & D_b \end{pmatrix}\right)$ *with*

$$S := \begin{pmatrix} \Sigma & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(t+3)\times(t+3)}, \qquad \Sigma_{rs} = \xi_r^\top \xi_s,$$

$$D_f := \begin{bmatrix} \mathbb{E}[(Z^{\tilde{x}_0^1})^2] & & \\ & \ddots & \\ & & \mathbb{E}[(Z^{\tilde{x}_0^{L-1}})^2] \end{bmatrix} \in \mathbb{R}^{(L-1)\times(L-1)},$$

$$D_b := \begin{bmatrix} \mathbb{E}[(Z^{d\tilde{h}_0^2})^2] & & \\ & \ddots & \\ & & \mathbb{E}[(Z^{d\tilde{h}_0^L})^2] \end{bmatrix} \in \mathbb{R}^{(L-1)\times(L-1)}.$$

(ii) $|\mathbb{E}[\psi(Z_0)]| < \infty$*, and* $|\psi(Z_0)| < \infty$ *almost surely for any polynomially bounded function* $\psi : \mathbb{R}^{t+2L} \longrightarrow \mathbb{R}$*.*

Remark. Note that the lemma stays valid even if ψ does not depend on the whole list of variables inside Z_0 but only on a couple of them, which will be the case in the Tensor Program. Point (ii) will be used repeatedly in different proofs to show that the expectations appearing in the forward and backward passes are finite.

Proof. Claim (i) simply comes from the definition of the initial vectors $U^1\xi_0$, ..., $U^1\xi_t$, U^{L+1} and from the ZHat rule in a Tensor Program. Claim (ii) then follows because all entries in the covariance matrix are finite by Lemma A.7.1, and since ψ is polynomially bounded and the moments of a Gaussian with finite variance are finite, $|\mathbb{E}[\psi(Z_0)]| \leq \mathbb{E}[|\psi(Z_0)|] < \infty$ and thus $|\psi(Z_0)| < \infty$ almost surely. □

Note that by Lemmas A.7.2 and A.7.3, the first forward and backward passes of IP-LLR easily express in function of the entries of Z_0. Let us now take care of the forward and backward passes at $t = 1$. As the dynamics evolve with time, the expression of the forward and backward passes of IP-LLR in function of Z_0

(or rather of some of the entries of Z_0) get more intricate. They are still easy to develop explicitly for $t = 1$ but we choose to simply express what variables appear in the expression of the forward and backward passes instead of giving the expression explicitly.

Lemma A.13.2 (Multiplications by initial weight matrices vanish with polynomially bounded variables)**.** *Consider the IP-LLR parameterization and let z be a vector in the program such that $Z^z = \psi(Z_0)$ with ψ polynomially bounded. Then, one has that for any $l \in [2, L]$:*

(i) if $\frac{\partial Z^z}{\partial \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}} = \phi(Z_0)$ with ϕ polynomially bounded, then $Z^{W^l(0)z} = 0$.

(ii) if $\frac{\partial Z^z}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}} = \phi(Z_0)$ with ϕ polynomially bounded, then $Z^{(W^l(0))^\top z} = 0$.

Proof. Let $l \in [2, L]$. We simply write

$$Z^{W^l(0)z} = \mathring{\omega}_l \widehat{Z}^{\widehat{W}^l z} + \mathring{\omega}_l \dot{Z}^{\widehat{W}^l z}$$

where $\widehat{Z}^{W^l(0)z} \sim \mathcal{N}(0, \mathbb{E}[(Z^z)^2])$ and the variance is finite by Lemma A.13.1 because $(Z^z)^2$ is a polynomially bounded function of Z_0 since Z^z is. This shows that $|\widehat{Z}^{W^l(0)z}| < \infty$ almost surely and thus that $\mathring{\omega}_l \widehat{Z}^{\widehat{W}^l z} = 0$ since $\mathring{\omega}_l = 0$ in IPs. On the other hand,

$$\dot{Z}^{\widehat{W}^l z} = \mathbb{E}\left[\frac{\partial Z^z}{\partial \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}}\right] Z^{d\tilde{h}_0^l}$$

and the expectation is finite by Lemma A.13.1 since $\partial Z^z / \partial \widehat{Z}^{(\widehat{W}^l)^\top} = \phi(Z_0)$ with ϕ polynomially bounded, and

$$Z^{d\tilde{h}_0^l} = \begin{cases} \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}l+1_0} \sigma'(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}) \text{ if } l \in [2, L-1] \\ \widehat{Z}^{U^{L+1}} \sigma'(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}) \text{ if } l = L \end{cases}$$

In any case, $Z^{d\tilde{h}_0^l}$ is a polynomially bounded function of Z_0 and is thus finite almost surely, which entails $\mathring{\omega}_l \widehat{Z}^{\widehat{W}^l z} = 0$, and therefore $\widehat{Z}^{W^l(0)z} = 0$ which gives (i).

The same reasoning with $W^l(0)^\top$ gives (ii) if $\partial Z^z / \partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}} = \phi(Z_0)$ with ϕ polynomially bounded.

□

The case $t = 1$

Lemma A.13.3 (Z_0 in the forward pass of IP-LLR at $t = 1$)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then, dropping the dependency of the forward pass on ξ_1, one has:*

(i) $Z^{h_1^1} = \psi\left(\widehat{Z}^{U^1\xi_0},\ \widehat{Z}^{U^1\xi_1},\ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right)$

(ii) $Z^{h_1^l} = \psi\left(\widehat{Z}^{\widehat{W}^l\tilde{x}_0^{l-1}},\ \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}\right),\ l \in [2, L-1]$

(iii) $Z^{h_1^L} = \psi\left(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L\tilde{x}_0^{L-1}}\right)$

and

(iv) $\frac{\partial Z^{h_1^{l-1}}}{\widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}} = \psi\left(Z_0\right)$, $l \in [2, L]$

(v) $\frac{\partial Z^{h_1^l}}{\widehat{Z}^{\widehat{W}^l\tilde{x}_0^{l-1}}} = \psi\left(Z_0\right)$, $l \in [2, L]$

and ***all*** *the different* ψ *that appear are polynomially bounded.*

Remark.

1. Recall that we simply use ψ or ϕ to mean that the variable **is a function of** the arguments of ψ (or ϕ) only, and that the different ψ and ϕ which appear in the different claims (i) to (v) are **not** actually the same.
2. For the partial derivatives we chose not to make a precise statement on which variables exactly appear in the expression as this will not matter and would only over-complicate things for close to none added-value.
3. Note that with the claims above, one can prove that $\mathring{\omega}_l Z^{\widehat{W}^l x_1^{l-1}} = 0$ because both of the terms $\widehat{Z}$ and $\dot{Z}$ defining $Z^{\widehat{W}^l x_1^{l-1}}$ are polynomially bounded functions of Gaussians which has finite covariance matrices, and $\mathring{\omega}_l = 0$ in IPs.

Proof. Using Theorem A.8.7 with $\xi = \xi_1$ and $t = 1$ we have claim (i) because, first $\widehat{Z}^{d\tilde{x}_0^1} = \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}$, and second σ' is polynomially bounded (see Appendix A.5). Claim (ii) also stems from Theorem A.8.7 since it holds that $\widehat{Z}^{d\tilde{x}_0^l} = \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}$, $\widehat{Z}^{\tilde{h}_0^l} = \widehat{Z}^{\widehat{W}^l\tilde{x}_0^{l-1}}$, and σ' is polynomially bounded. Finally, claim (iii) also stems from Theorem A.8.7 since $\widehat{Z}^{d\tilde{x}_0^L} = Z^{U^{L+1}}$, $\widehat{Z}^{\tilde{h}_0^L} = \widehat{Z}^{\widehat{W}^l\tilde{x}_0^{L-1}}$, and σ' is polynomially bounded.

From Theorem A.8.7, we get:

$$\frac{\partial Z^{h_1^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} = -\eta\mathring{\chi}_0(\xi_0^\top\xi_1)\sigma'(\widehat{Z}^{U^1\xi_0})$$

For $l \in [3, L]$, from Theorem A.8.7, we get

$$\frac{\partial Z^{h_1^{l-1}}}{Z^{(\widehat{W}^l)^\top d\tilde{h}_0^l}} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^{l-2}}Z^{\tilde{x}_1^{l-2}}]\sigma'(\widehat{Z}^{\widehat{W}^l\tilde{x}_0^{l-1}})$$

which immediately gives claim (iv) since σ' is polynomially bounded and with claim (ii) and Lemma A.13.1, we also have $|\mathbb{E}[Z^{\tilde{x}_0^{l-2}} Z^{\tilde{x}_1^{l-2}}]| < \infty$. Similarly, for $l \in [2, L]$, from Theorem A.8.7, we get

$$\frac{\partial Z^{h_1^l}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^l}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^{l-1}} Z^{\tilde{x}_1^{l-1}}] Z^{d\tilde{x}_0^l} \sigma''(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}})$$

and $Z^{d\tilde{x}_0^{l-1}} = \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}$ if $l \in [2, L-1]$, and $Z^{d\tilde{x}_0^{l-1}} = \widehat{Z}^{U^{L+1}}$ if $l = L$. Since the expectation is finite by claim (ii) and Lemma A.13.1, and since σ'' is polynomially bounded, we get claim (v). □

Lemma A.13.4 (Z_0 in the backward pass of IP-LLR at $t = 1$)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then, dropping the dependency of the forward and backward passes on ξ_1, one has:*

(i) $Z^{d\tilde{x}_1^L} = \psi\left(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}\right)$,
$Z^{d\tilde{h}_1^L} = \psi\left(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}\right)$

(ii) $Z^{d\tilde{x}_1^{l-1}} = \psi\left(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}^{l-2}}\right)$,
$Z^{d\tilde{h}_1^{l-1}} = \psi\left(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}^{l-2}},\ \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}\right)$, $l \in [3, L]$

(iii) $Z^{d\tilde{x}_1^1} = \psi\left(\widehat{Z}^{U^1 \xi_0}\right)$,
$Z^{d\tilde{h}_1^1} = \psi\left(\widehat{Z}^{U^1 \xi_0},\ \widehat{Z}^{U^1 \xi_1},\ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right)$

and

(iv) $\frac{\partial Z^{d\tilde{h}_1^l}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}} = \psi(Z_0)$, $l \in [1, L]$

(v) $\frac{\partial Z^{d\tilde{h}_1^l}}{\partial \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}} = \psi(Z_0)$, $l \in [1, L-1]$

*and **all** the different ψ that appear are polynomially bounded.*

Proof. For the backward pass, we have by definition of the tilde variables for $t \geq 1$, $d\tilde{x}_1^L = w^{L+1}(1) = U^{L+1} - \eta \chi_0 \tilde{x}_1^L$ by Lemma A.7.4.2, and thus

$$Z^{d\tilde{x}_1^L} = \widehat{Z}^{U^{L+1}} - \eta \mathring{\chi}_0 \sigma(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$$

Then,

$$Z^{d\tilde{h}_1^L} = Z^{d\tilde{x}_1^L} \sigma'(Z^{h_1^L})$$

which gives claim (i) since σ and σ' are polynomially bounded, and $Z^{h_1^L} = \psi(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}})$ and ψ is polynomially bounded by Lemma A.13.3.

For $l = L-1$, we have

$$\begin{aligned} d\tilde{x}_1^{L-1} &= (W^L(1))^\top d\tilde{h}_1^L \\ &= \omega_L(\widehat{W}^l)^\top d\tilde{h}_1^L - \eta\chi_0 \frac{(d\tilde{h}_0^L)^\top d\tilde{h}_1^L}{m} \tilde{x}_0^{L-1} \end{aligned}$$

which gives

$$Z^{d\tilde{x}_1^{L-1}} = \mathring{\omega}_L Z^{(\widehat{W}^l)^\top d\tilde{h}_1^L} - \eta\mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_1^L}] Z^{\tilde{x}_0^{L-1}}$$

Now, by the previous expression of $Z^{d\tilde{h}_1^L}$ and by Lemma A.13.3, we get

$$\frac{\partial Z^{d\tilde{h}_1^L}}{\partial \widehat{W}^l \tilde{x}_0^{L-1}} = -\eta\mathring{\chi}_0 \sigma'(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}) \sigma'(Z^{h_1^L}) + Z^{d\tilde{x}_1^L} \frac{\partial Z^{h_1^L}}{\widehat{W}^l \tilde{x}_0^{L-1}} \sigma''(Z^{h_1^L})$$

and by claim (i) and Lemma A.13.3 we get

$$\frac{\partial Z^{d\tilde{h}_1^L}}{\partial \widehat{W}^l \tilde{x}_0^{L-1}} = \psi(Z_0)$$

with ψ polynomially bounded since σ' and σ'' are polynomially bounded. Therefore, by Lemma A.13.2, we get $\mathring{\omega}_L Z^{(\widehat{W}^l)^\top d\tilde{h}_1^L} = 0$.

We thus simply get

$$Z^{d\tilde{x}_1^{L-1}} = -\eta\mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_1^L}] Z^{\tilde{x}_0^{L-1}}.$$

$Z^{d\tilde{h}_0^L}$ and $Z^{d\tilde{h}_1^L}$ are polynomially bounded functions of Z_0 and thus this is also true for $Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_1^L}$, and by Lemma A.13.1, $|\mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_1^L}]| < \infty$. Since $Z^{d\tilde{x}_1^{L-1}} = \psi(Z^{\tilde{x}_0^{L-1}})$ with ψ polynomially bounded, we thus get $Z^{d\tilde{x}_1^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}})$ and ψ is polynomially bounded (indeed: $\psi(z) = -\eta\mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_1^L}] \sigma(z)$).

We have

$$Z^{d\tilde{h}_1^{L-1}} = Z^{d\tilde{x}_1^{L-1}} \sigma'(Z^{h_1^{L-1}})$$

and since by Lemma A.13.3, $Z^{h_1^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}}, \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^L})$ with ψ polynomially bounded, by the previous result for $Z^{d\tilde{x}_1^{L-1}}$ and since σ' is polynomially bounded we get

$$Z^{d\tilde{h}_1^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}}, \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^L})$$

with ψ polynomially bounded.

We have

$$\begin{aligned}\frac{\partial Z^{d\tilde{h}_1^{L-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}}} =& \frac{\partial Z^{d\tilde{x}_1^{L-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}}}\sigma'(Z^{h_1^{L-1}}) + Z^{d\tilde{x}_1^{L-1}}\frac{\partial Z^{h_1^{L-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}}}\sigma''(Z^{h_1^{L-1}}) \\ =& -\eta\mathring{\chi}_0\mathbb{E}[Z^{d\tilde{h}_0^L}Z^{d\tilde{h}_1^L}]\sigma'(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}})\sigma'(Z^{h_1^{L-1}}) + \\ & Z^{d\tilde{x}_1^{L-1}}\frac{\partial Z^{h_1^{L-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}}}\sigma''(Z^{h_1^{L-1}})\end{aligned}$$

By Lemma A.13.3 and since σ' and σ'' are polynomially bounded, and we have already proven that $Z^{d\tilde{x}_1^{L-1}} = \psi(Z_0)$ with ψ polynomially bounded, as well as $|\mathbb{E}[Z^{d\tilde{h}_0^L}Z^{d\tilde{h}_1^L}]| < \infty$, we get

$$\frac{\partial Z^{d\tilde{h}_1^{L-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{L-2}}} = \psi(Z_0)$$

with ψ polynomially bounded.

Similarly, we have

$$\begin{aligned}\frac{\partial Z^{d\tilde{h}_1^{L-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L}} =& \frac{\partial Z^{d\tilde{x}_1^{L-1}}}{\partial Z^{(\widehat{W}^L)^\top d\tilde{h}_0^L}}\sigma'(Z^{h_1^{L-1}}) + Z^{d\tilde{x}_1^{L-1}}\frac{\partial Z^{h_1^{L-1}}}{\partial Z^{(\widehat{W}^L)^\top d\tilde{h}_0^L}}\sigma''(Z^{h_1^{L-1}}) \\ =& Z^{d\tilde{x}_1^{L-1}}\frac{\partial Z^{h_1^{L-1}}}{\partial Z^{(\widehat{W}^L)^\top d\tilde{h}_0^L}}\sigma''(Z^{h_1^{L-1}})\end{aligned}$$

Now, we have shown above that $Z^{d\tilde{x}_1^{L-1}} = \psi(Z_0)$ with ψ polynomially bounded, and by Lemma A.13.3 we have that both $\partial Z^{h_1^{L-1}}/\partial Z^{(\widehat{W}^L)^\top d\tilde{h}_0^L}$ and $Z^{h_1^{L-1}}$ are polynomially bounded functions of Z_0, which gives

$$\frac{\partial Z^{d\tilde{h}_1^{L-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L}} = \psi(Z_0)$$

with ψ polynomially bounded.

Let $l \in [2, L-1]$ and assume claims (ii), (iv) and (v) are true for layer l. We have

$$Z^{d\tilde{x}_1^{l-1}} = \mathring{\omega}_l Z^{(\widehat{W}^l)^\top d\tilde{h}_1^l} - \eta\mathring{\chi}_0\mathbb{E}[Z^{d\tilde{h}_0^l}Z^{d\tilde{h}_1^l}]Z^{\tilde{x}_0^{l-1}}$$

Since by the induction hypothesis $\partial Z^{d\tilde{h}_1^l}/\partial \widehat{Z}^{\widehat{W}^l\tilde{x}_0^{l-1}} = \psi(Z_0)$ with ψ polynomially bounded, and $Z^{d\tilde{h}_1^l}$ is a polynomially bounded function of Z_0, by Lemma A.13.2 we get $\mathring{\omega}_l Z^{(\widehat{W}^l)^\top d\tilde{h}_1^l} = 0$. Then, we simply get

$$Z^{d\tilde{x}_1^{l-1}} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{d\tilde{h}_0^l}Z^{d\tilde{h}_1^l}]Z^{\tilde{x}_0^{l-1}}$$

Again here, since both $Z^{d\tilde{h}_0^l}$ and $Z^{d\tilde{h}_1^l}$ are polynomially bounded functions of Z_0, then so is $Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}$, which shows by Lemma A.13.1 that $|\mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}]| < \infty$. If $l \geq 3$, since $Z^{\tilde{x}_0^{l-1}} = \sigma(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}})$ and σ is polynomially bounded, we get that

$$Z^{d\tilde{x}_1^{l-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}})$$

If $l = 2$, since $Z^{\tilde{x}_0^1} = \sigma(Z^{U^1\xi_0})$ and σ is polynomially bounded we get:

$$Z^{d\tilde{x}_1^{l-1}} = \psi(Z^{U^1\xi_0})$$

with ψ polynomially bounded.

We then have

$$Z^{d\tilde{h}_1^{l-1}} = Z^{d\tilde{x}_1^{l-1}} \sigma'(Z^{h_1^{l-1}})$$

and thus

$$\begin{aligned}
\frac{\partial Z^{d\tilde{h}_1^{l-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}} =& \frac{\partial Z^{d\tilde{x}_1^{l-1}}}{\partial \widehat{Z}^{\widehat{W}^{l}-1(0)\tilde{x}_0^{l-2}}} \sigma'(Z^{h_1^{l-1}}) + Z^{d\tilde{x}_1^{l-1}} \frac{\partial Z^{h_1^{l-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}} \sigma''(Z^{h_1^{l-1}}) \\
=& -\eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}] \sigma'(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}) \sigma'(Z^{h_1^{l-1}}) + \\
& Z^{d\tilde{x}_1^{l-1}} \frac{\partial Z^{h_1^{l-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}} \sigma''(Z^{h_1^{l-1}})
\end{aligned}$$

By Lemma A.13.3 as well as the previous result on $Z^{d\tilde{x}_1^{l-1}}$, and since it holds that $|\mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_1^l}]| < \infty$, and σ' and σ'' are polynomially bounded, we get that

$$\frac{\partial Z^{d\tilde{h}_1^{l-1}}}{\partial \widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}} = \psi(Z_0)$$

with ψ polynomially bounded.

Similarly

$$\begin{aligned}
\frac{\partial Z^{d\tilde{h}_1^{l-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}} =& \frac{\partial Z^{d\tilde{x}_1^{l-1}}}{\partial \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^l}} \sigma'(Z^{h_1^{l-1}}) + Z^{d\tilde{x}_1^{l-1}} \frac{\partial Z^{h_1^{l-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}} \sigma''(Z^{h_1^{l-1}}) \\
=& Z^{d\tilde{x}_1^{l-1}} \frac{\partial Z^{h_1^{l-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}} \sigma''(Z^{h_1^{l-1}})
\end{aligned}$$

and the three quantities in the product are polynomially bounded functions of Z_0 (shown above for the first term and by Lemma A.13.3 for the two other

terms). We thus get

$$\frac{\partial Z^{d\tilde{h}_1^{l-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}} = \psi(Z_0)$$

with ψ polynomially bounded. This concludes the induction and thus proves claims (ii), (iii), (iv) and (v) by induction.

□

Corollary A.13.4.1 (Multiplications by the initial weight matrices vanish in IP-LLR at $t = 1$)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then for any $l \in [2, L]$, one has:*

$$\begin{cases} Z^{W^l(0)x_1^{l-1}} = \mathring{\omega}_l Z^{\widehat{W}^l x_1^{l-1}} = 0 \\ Z^{(W^l(0))^\top d\tilde{h}_1^l} = \mathring{\omega}_l Z^{(\widehat{W}^l)^\top d\tilde{h}_1^l} = 0 \end{cases}$$

Proof. Those results are actually hidden in the proof of Lemma A.13.4 and come from Lemma A.13.2. □

The case $t = 2$

Lemma A.13.5 (Z_0 in the forward pass of IP-LLR at $t = 2$)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then, dropping the dependency of the forward pass on ξ_2, one has:*

(i) $Z^{h_2^1} = \psi\left(\widehat{Z}^{U^1\xi_0},\ \widehat{Z}^{U^1\xi_1},\ \widehat{Z}^{U^1\xi_2},\ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right)$

(ii) $Z^{h_2^l} = \psi\left(\widehat{Z}^{\widehat{W}^l\tilde{x}_0^{l-1}},\ \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}\right),\ l \in [2, L-1]$

(iii) $Z^{h_2^L} = \psi\left(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L\tilde{x}_0^{L-1}}\right)$

and

(iv) $\frac{\partial Z^{h_2^{l-1}}}{\widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}} = \psi(Z_0)$, $l \in [2, L]$

(v) $\frac{\partial Z^{h_2^l}}{\widehat{Z}^{\widehat{W}^l\tilde{x}_0^{l-1}}} = \psi(Z_0)$, $l \in [2, L]$

and

(vi) $Z^{W^l(0)x_2^{l-1}} = 0$, $l \in [2, L]$

*and **all** the different ψ that appear are polynomially bounded.*

Proof. We have

$$h_2^1 = U^1\xi_2 - \eta\chi_0(\xi_0^\top\xi_2)d\tilde{h}_0^1 - \eta\chi_1(\xi_1^\top\xi_2)d\tilde{h}_1^1$$

which gives

$$Z^{h_2^1} = Z^{U^1\xi_2} - \eta\mathring{\chi}_0(\xi_0^\top\xi_2)Z^{d\tilde{h}_0^1} - \eta\mathring{\chi}_1(\xi_1^\top\xi_2)Z^{d\tilde{h}_1^1}$$

By Lemma A.13.4 $Z^{d\tilde{h}_1^1} = \psi(\widehat{Z}^{U^1\xi_0}, \widehat{Z}^{U^1\xi_1}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2})$ and we also have $Z^{d\tilde{h}_0^1} = \psi(Z^{U^1\xi_0}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2})$ where the different ψ are polynomially bounded, which gives claim (i).

We have

$$\frac{\partial Z^{h_2^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} = -\eta\mathring{\chi}_0(\xi_0^\top\xi_2)\frac{\partial Z^{d\tilde{h}_0^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} - \eta\mathring{\chi}_1(\xi_1^\top\xi_2)\frac{\partial Z^{d\tilde{h}_1^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}}$$

with

$$\frac{\partial Z^{d\tilde{h}_0^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} = \sigma'(Z^{U^1\xi_0})$$

which is a polynomially bounded function of Z_0 and so is $\partial Z^{d\tilde{h}_1^1}/\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}$ by Lemma A.13.4. We thus get claim (iv) for $l = 2$.

We have

$$Z^{h_2^2} = \mathring{\omega}_2 Z^{\widehat{W}^2 x_2^1} - \eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^1}Z^{x_2^1}]Z^{d\tilde{h}_0^2} - \eta\mathring{\chi}_1\mathbb{E}[Z^{x_1^1}Z^{x_2^1}]Z^{d\tilde{h}_1^2}$$

Now $Z^{x_2^1} = \sigma(Z^{h_2^1})$ is a polynomially bounded function of Z_0 because $Z^{h_2^1}$ is and σ is polynomially bounded. Secondly, we have

$$\frac{\partial Z^{x_2^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} = \frac{\partial Z^{h_2^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}}\sigma'(Z^{h_2^1})$$

which is a polynomially bounded function of Z_0 by the previous results. By Lemma A.13.2 we get that $\mathring{\omega}_2 Z^{\widehat{W}^2 x_2^1} = 0$ which gives claim (vi) for $l = 2$. In addition, this yields

$$Z^{h_2^2} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^1}Z^{x_2^1}]Z^{d\tilde{h}_0^2} - \eta\mathring{\chi}_1\mathbb{E}[Z^{x_1^1}Z^{x_2^1}]Z^{d\tilde{h}_1^2}$$

which gives claim (ii) for $l = 2$ by the results for the backward passes at time $t = 0$ and $t = 1$ and because the expectations are finite since the integrands are polynomially bounded functions of Z_0, as they are products of such variables by the induction hypothesis. Additionally, we have

$$\frac{\partial Z^{h_2^2}}{\partial \widehat{Z}^{(\widehat{W}^3(0))^\top d\tilde{h}_0^3}} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{\tilde{x}_0^1}Z^{x_2^1}]\frac{\partial Z^{d\tilde{h}_0^2}}{\partial \widehat{Z}^{(\widehat{W}^3(0))^\top d\tilde{h}_0^3}} - \eta\mathring{\chi}_1\mathbb{E}[Z^{x_1^1}Z^{x_2^1}]\frac{\partial Z^{d\tilde{h}_1^2}}{\partial \widehat{Z}^{(\widehat{W}^3(0))^\top d\tilde{h}_0^3}}$$

and we have

$$\frac{\partial Z^{d\tilde{h}_0^2}}{\partial \widehat{Z}^{(\widehat{W}^3(0))^\top d\tilde{h}_0^3}} = \sigma'(\widehat{Z}^{\widehat{W}^2\tilde{x}_0^1})$$

and $\partial Z^{d\tilde{h}_1^2}/\partial \widehat{Z}^{(\widehat{W}^3(0))^\top d\tilde{h}_0^3}$ is a polynomially bounded function of Z_0 by Lemma A.13.4. Once again, since the expectations are finite, we thus get that

$$\frac{\partial Z^{h_2^2}}{\partial \widehat{Z}^{(\widehat{W}^3(0))^\top d\tilde{h}_0^3}} = \psi(Z_0)$$

with ψ polynomially bounded. A similar reasoning would prove that

$$\frac{\partial Z^{h_2^2}}{\partial \widehat{Z}^{\widehat{W}^3(0)\tilde{x}_0^2}} = \psi(Z_0)$$

with ψ polynomially bounded because

$$\frac{\partial Z^{d\tilde{h}_0^1}}{\partial \widehat{Z}^{\widehat{W}^3(0)\tilde{x}_0^2}} = \widehat{Z}^{(\widehat{W}^3(0))^\top d\tilde{h}_0^3}\sigma''(\widehat{Z}^{\widehat{W}^2\tilde{x}_0^1})$$

and $\partial Z^{d\tilde{h}_1^1}/\partial \widehat{Z}^{\widehat{W}^3(0)\tilde{x}_0^2} = \psi(Z_0)$ with ψ polynomially bounded by Lemma A.13.4.

Let $l \in [2, L-1]$ and assume claims (ii), (iv), (v), and (vi) for layer l. Then, we have:

$$Z^{h_2^{l+1}} = \mathring{\omega}_{l+1} Z^{\widehat{W}^{l+1}x_2^l} - \eta\mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}] Z^{d\tilde{h}_0^{l+1}} - \eta\mathring{\chi}_1 \mathbb{E}[Z^{x_1^l} Z^{x_2^l}] Z^{d\tilde{h}_1^{l+1}}$$

Now $Z^{x_2^l} = \sigma(Z^{h_2^l})$ is a polynomially bounded function of Z_0 because $Z^{h_2^l}$ is by the induction hypothesis and σ is polynomially bounded. Secondly, we have

$$\frac{\partial Z^{x_2^l}}{\partial \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}} = \frac{\partial Z^{h_2^l}}{\partial \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}} \sigma'(Z^{h_2^l})$$

which is a polynomially bounded function of Z_0 by the induction hypothesis. By Lemma A.13.2 we get that $\mathring{\omega}_{l+1} Z^{\widehat{W}^{l+1}x_2^l} = 0$ which gives claim (vi) for layer $l+1$. In addition, this yields

$$Z^{h_2^{l+1}} = -\eta\mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}] Z^{d\tilde{h}_0^{l+1}} - \eta\mathring{\chi}_1 \mathbb{E}[Z^{x_1^1} Z^{x_2^1}] Z^{d\tilde{h}_1^{l+1}}$$

which gives claim (ii) for layer $l+1$ by the results for the backward passes at time $t=0$ and $t=1$ and because the expectations are finite since the integrands are polynomially bounded functions of Z_0, as they are products of such variables. The only thing that one has to be careful with is that if $l+1 = L$, then $Z^{\tilde{h}_0^{l+1}} = \psi(Z^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l\tilde{x}_0^{L-1}})$ and $Z^{\tilde{h}_1^{l+1}} = \psi(Z^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l\tilde{x}_0^{L-1}})$ with both ψ polynomially bounded, which gives claim (iii). Otherwise, if $l+1 \leq L-1$,

$Z^{\tilde{h}_0^{l+1}} = \psi(\widehat{Z}^{\widehat{W}^{l+1}\tilde{x}_0^l}, \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}^{l+2})}$ and $Z^{\tilde{h}_1^{l+1}} = \psi(\widehat{Z}^{\widehat{W}^{l+1}\tilde{x}_0^l}, \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}^{l+2})}$ with both ψ polynomially bounded, which gives claim (ii) for layer $l+1$.

Now, if $l+1 \leq L-1$,

$$\frac{\partial Z^{h_2^{l+1}}}{\partial \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}_0^{l+2}}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{\tilde{x}_0^l} Z^{x_2^l}] \frac{\partial Z^{d\tilde{h}_0^{l+1}}}{\partial \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}_0^{l+2}}} - \eta \mathring{\chi}_1 \mathbb{E}[Z^{x_1^l} Z^{x_2^l}] \frac{\partial Z^{d\tilde{h}_1^{l+1}}}{\partial \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}_0^{l+2}}}$$

and we have

$$\frac{\partial Z^{d\tilde{h}_0^{l+1}}}{\partial \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}_0^{l+2}}} = \sigma'(\widehat{Z}^{\widehat{W}^{l+1}\tilde{x}_0^l})$$

and $\partial Z^{d\tilde{h}_1^{l+1}} / \partial \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}_0^{l+2}}$ is a polynomially bounded function of Z_0 by Lemma A.13.4. Once again, since the expectations are finite, we thus get that

$$\frac{\partial Z^{h_2^{l+1}}}{\partial \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}_0^{l+2}}} = \psi(Z_0)$$

with ψ polynomially bounded, which proves claim (iv) for layer $l+1$. A similar reasoning would prove that

$$\frac{\partial Z^{h_2^{l+1}}}{\partial \widehat{Z}^{\widehat{W}^{l+1}\tilde{x}_0^l}} = \psi(Z_0)$$

with ψ polynomially bounded because

$$\frac{\partial Z^{d\tilde{h}_0^{l+1}}}{\partial \widehat{Z}^{\widehat{W}^{l+2}(0)\tilde{x}_0^l}} = \begin{cases} \widehat{Z}^{(\widehat{W}^{l+2}(0))^\top d\tilde{h}_0^{l+2}} \sigma''(\widehat{Z}^{\widehat{W}^{l+1}\tilde{x}_0^l}) \text{ if } l+1 \leq L-1 \\ \widehat{Z}^{U^{L+1}} \sigma''(\widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}) \text{ if } l+1 = L \end{cases}$$

and $\partial Z^{d\tilde{h}_1^{l+1}} / \partial \widehat{Z}^{\widehat{W}^{l+1}\tilde{x}_0^l} = \psi(Z_0)$ with ψ polynomially bounded by Lemma A.13.4. This proves claim (v) and thus concludes the induction and with it the proof. □

Lemma A.13.6 (Z_0 in the backward pass of IP-LLR at $t=2$)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then, dropping the dependency of the forward and backward passes on ξ_2, one has:*

(i) $Z^{d\tilde{x}_2^L} = \psi\left(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}\right)$,
$Z^{d\tilde{h}_2^L} = \psi\left(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}\right)$

(ii) $Z^{d\tilde{x}_2^{l-1}} = \psi\left(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}^{l-2}}, \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}\right)$,
$Z^{d\tilde{h}_2^{l-1}} = \psi\left(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}^{l-2}}, \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}\right)$, $l \in [3, L]$

(iii) $Z^{d\tilde{x}_2^1} = \psi\left(\widehat{Z}^{U^1\xi_0},\ \widehat{Z}^{U^1\xi_1},\ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right)$,
$Z^{d\tilde{h}_2^1} = \psi\left(\widehat{Z}^{U^1\xi_0},\ \widehat{Z}^{U^1\xi_1},\ \widehat{Z}^{U^1\xi_2},\ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right)$

and

(iv) $\frac{\partial Z^{d\tilde{h}_2^l}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}} = \psi(Z_0)$, $l \in [2, L]$

(v) $\frac{\partial Z^{d\tilde{h}_2^{l-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}} = \psi(Z_0)$, $l \in [2, L]$

and

(vi) $Z^{(W^l(0))^\top d\tilde{h}_2^l} = 0$, $l \in [2, L]$

and ***all*** *the different* ψ *that appear are polynomially bounded.*

Proof. We have:

$$Z^{d\tilde{x}_2^L} = \widehat{Z}^{U^{L+1}} - \eta\mathring{\chi}_0 Z^{\tilde{x}_0^L} - \eta\mathring{\chi}_1 Z^{x_1^L}$$

where $Z^{\tilde{x}_0^L} = \sigma(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$ and $Z^{x_1^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$ with ψ polynomially bounded by Lemma A.13.3. Combining all this gives

$$Z^{d\tilde{x}_2^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$$

with ψ polynomially bounded since σ is also polynomially bounded. Then

$$Z^{d\tilde{h}_2^L} = Z^{d\tilde{x}_2^L} \sigma'(Z^{h_2^L})$$

and since $Z^{h_2^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$ with ψ polynomially bounded by Lemma A.13.5, we get

$$Z^{d\tilde{h}_2^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$$

with ψ polynomially bounded since σ' is also polynomially bounded. This thus proves claim (i). Now, we have

$$\frac{\partial Z^{d\tilde{h}_2^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} = \frac{\partial Z^{d\tilde{x}_2^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} \sigma'(Z^{h_2^L}) + Z^{d\tilde{x}_2^L} \frac{\partial Z^{h_2^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} \sigma''(Z^{h_2^L})$$

where $Z^{h_2^L}$, $\partial Z^{h_2^L} / \partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}$, and $Z^{d\tilde{x}_2^L}$ are polynomially bounded functions of Z_0 by the previous result and by Lemma A.13.5. We have

$$\frac{\partial Z^{d\tilde{x}_2^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} = -\eta\mathring{\chi}_0 \sigma'(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}) - \eta\mathring{\chi}_1 \frac{\partial Z^{h_1^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} \sigma'(Z^{h_1^L})$$

which is a polynomially bounded function of Z_0 since σ' is polynomially bounded and by Lemma A.13.3. We thus get

$$\frac{\partial Z^{d\tilde{h}_2^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} = \psi(Z_0)$$

with ψ polynomially bounded since σ' and σ'' are polynomially bounded. This proves (iv) for $l = L$.

We have:

$$Z^{d\tilde{x}_2^{L-1}} = \mathring{\omega}_L \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_2^L} - \eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_2^L}] Z^{\tilde{x}_0^{L-1}} - \eta \mathring{\chi}_1 \mathbb{E}[Z^{d\tilde{h}_1^L} Z^{d\tilde{h}_2^L}] Z^{x_1^{L-1}}$$

From the previous step we have that both $Z^{d\tilde{h}_2^L}$ and $\partial Z^{d\tilde{h}_2^L} / \partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}$ are polynomially bounded functions of Z_0. By Lemma A.13.2, this first shows that $\mathring{\omega}_L \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_2^L} = 0$, and thus gives (vi) for $l = L$, leading to:

$$Z^{d\tilde{x}_2^{L-1}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_2^L}] Z^{\tilde{x}_0^{L-1}} - \eta \mathring{\chi}_1 \mathbb{E}[Z^{d\tilde{h}_1^L} Z^{d\tilde{h}_2^L}] Z^{x_1^{L-1}}$$

Now $Z^{\tilde{x}_0^{L-1}} = \sigma(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-2}})$ and by Lemma A.13.3, we also have that $Z^{x_1^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L})$ with ψ polynomially bounded. As always the expectations are finite by Lemma A.13.1 because the integrands are polynomially bounded functions of Z_0 as products of such variables. Since σ is also polynomially bounded, this gives

$$Z^{d\tilde{x}_2^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-1}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L})$$

Then, we have

$$Z^{d\tilde{h}_2^{L-1}} = Z^{d\tilde{x}_2^{L-1}} \sigma'(Z^{h_2^{L-1}})$$

and since $Z^{h_2^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-1}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L})$ with ψ polynomially bounded by Lemma A.13.5, we get

$$Z^{d\tilde{h}_2^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-1}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L})$$

with ψ polynomially bounded since σ' is also polynomially bounded. This thus proves claim (ii) for $l = L-1$. Now, let $Z \in \{\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-1}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^L}\}$. We have

$$\frac{\partial Z^{d\tilde{h}_2^{L-1}}}{\partial Z} = \frac{\partial Z^{d\tilde{x}_2^{L-1}}}{\partial Z} \sigma'(Z^{h_2^{L-1}}) + Z^{d\tilde{x}_2^{L-1}} \frac{\partial Z^{h_2^{L-1}}}{\partial Z} \sigma''(Z^{h_2^{L-1}})$$

where $Z^{h_2^{L-1}}$, $\partial Z^{h_2^{L-1}} / \partial Z$, and $Z^{d\tilde{x}_2^{L-1}}$ are polynomially bounded functions of Z_0 by the previous result and by Lemma A.13.5. We have

$$\frac{\partial Z^{d\tilde{x}_2^{L-1}}}{\partial Z} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^L} Z^{d\tilde{h}_2^L}] \frac{\partial Z^{\tilde{h}_0^{L-1}}}{\partial Z} \sigma'(Z^{\tilde{h}_0^{L-1}}) - \eta \mathring{\chi}_1 \mathbb{E}[Z^{d\tilde{h}_1^L} Z^{d\tilde{h}_2^L}] \frac{\partial Z^{h_1^{L-1}}}{\partial Z} \sigma'(Z^{h_1^{L-1}})$$

which is a polynomially bounded function of Z_0 since σ' is polynomially bounded and by Lemma A.13.3.

For both possible values of Z, the expression of $\partial Z^{\tilde{h}_0^{L-1}}/\partial Z$ is easy to obtain and is a polynomially bounded function of Z_0 (this has actually already been shown for the proofs at time $t = 1$), and $Z^{h_1^{L-1}}/\partial Z = \psi(Z_0)$ with ψ polynomially bounded by Lemma A.13.3. Since the expectations are finite and σ' is polynomially bounded, we get

$$\frac{\partial Z^{d\tilde{x}_2^{L-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded and thus

$$\frac{\partial Z^{d\tilde{h}_2^{L-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded. This proves (iv) and (v) for $l = L - 1$.

Let $l \in [2, L-1]$, and assume claims (ii), (iv), (v), are true at layer l and claim (vi) is true at layer $l+1$. We have:

$$Z^{d\tilde{x}_2^{l-1}} = \mathring{\omega}_l \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_2^l} - \eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_2^l}] Z^{\tilde{x}_0^{l-1}} - \eta \mathring{\chi}_1 \mathbb{E}[Z^{d\tilde{h}_1^l} Z^{d\tilde{h}_2^l}] Z^{x_1^{l-1}}$$

From the induction hypothesis we have that both $Z^{d\tilde{h}_2^l}$ and $\partial Z^{d\tilde{h}_2^l}/\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}$ are polynomially bounded functions of Z_0. By Lemma A.13.2, this first shows that $\mathring{\omega}_l \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_2^l} = 0$, and thus gives (vi) for layer l, leading to:

$$Z^{d\tilde{x}_2^{l-1}} = -\eta \mathring{\chi}_0 \mathbb{E}[Z^{d\tilde{h}_0^l} Z^{d\tilde{h}_2^l}] Z^{\tilde{x}_0^{l-1}} - \eta \mathring{\chi}_1 \mathbb{E}[Z^{d\tilde{h}_1^l} Z^{d\tilde{h}_2^l}] Z^{x_1^{l-1}}$$

Now, if $l - 1 \geq 2$, $Z^{\tilde{x}_0^{l-1}} = \sigma(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{l-2}})$ and by Lemma A.13.3, we also have that $Z^{x_1^{l-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l})$ with ψ polynomially bounded. On the other hand, if $l - 1 = 1$, we have $Z^{\tilde{x}_0^{l-1}} = \sigma(\widehat{Z}^{U^1 \xi_0})$ and we also have that $Z^{x_1^{l-1}} = \psi(\widehat{Z}^{U^1 \xi_0}, \widehat{Z}^{U^1 \xi_1}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2})$ by Lemma A.13.3. As always the expectations are finite by Lemma A.13.1 because the integrands are polynomially bounded functions of Z_0 as products of such variables. Since σ is also polynomially bounded, this gives

$$Z^{d\tilde{x}_2^{l-1}} = \begin{cases} \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}) \text{ if } l - 1 \geq 2 \\ \psi(\widehat{Z}^{U^1 \xi_0}, \widehat{Z}^{U^1 \xi_1}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}) \text{ if } l - 1 = 1 \end{cases}$$

Since $Z^{d\tilde{h}_2^{l-1}} = Z^{d\tilde{x}_2^{l-1}} \sigma'(Z^{h_2^{l-1}})$, by Lemma A.13.3 we get

$$Z^{d\tilde{h}_2^{l-1}} = \begin{cases} \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}) \text{ if } l - 1 \geq 2 \\ \psi(\widehat{Z}^{U^1 \xi_0}, \widehat{Z}^{U^1 \xi_1}, \widehat{Z}^{U^1 \xi_2}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}) \text{ if } l - 1 = 1 \end{cases}$$

This gives claim (ii) for layer $l-1$ and claim (iii) for the case when $l-1=1$. Now, let $Z \in \{\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}\}$. We have

$$\frac{\partial Z^{d\tilde{h}_2^{l-1}}}{\partial Z} = \frac{\partial Z^{d\tilde{x}_2^{l-1}}}{\partial Z}\sigma'(Z^{h_2^{l-1}}) + Z^{d\tilde{x}_2^{l-1}}\frac{\partial Z^{h_2^{l-1}}}{\partial Z}\sigma''(Z^{h_2^{l-1}})$$

where $Z^{h_2^{l-1}}$ and $Z^{d\tilde{x}_2^{l-1}}$ are polynomially bounded functions of Z_0 by the previous result and by Lemma A.13.5. Also by Lemma A.13.5, we have

$$\frac{\partial Z^{h_2^{l-1}}}{\partial Z} = \begin{cases} 0 \text{ if } l-1=1 \text{ and } Z = \widehat{Z}^{\widehat{W}^2\tilde{x}_0^1} \\ \psi(Z_0) \text{ otherwise} \end{cases}$$

with ψ polynomially bounded. In any case, $\partial Z^{h_2^{l-1}}/\partial Z$ is a polynomially bounded function of Z_0. On the other hand, we have

$$\frac{\partial Z^{d\tilde{x}_2^{l-1}}}{\partial Z} = -\eta\mathring{\chi}_0\mathbb{E}[Z^{d\tilde{h}_0^l}Z^{d\tilde{h}_2^l}]\frac{\partial Z^{\tilde{h}_0^{l-1}}}{\partial Z}\sigma'(Z^{\tilde{h}_0^{l-1}}) - \eta\mathring{\chi}_1\mathbb{E}[Z^{d\tilde{h}_1^l}Z^{d\tilde{h}_2^l}]\frac{\partial Z^{h_1^{l-1}}}{\partial Z}\sigma'(Z^{h_1^{l-1}})$$

For both possible values of Z, $\partial Z^{\tilde{h}_0^{l-1}}/\partial Z$ has an easy expression and is a polynomially bounded function of Z_0 (essentially because σ and its derivatives are polynomially bounded). On the other hand, $\partial Z^{\tilde{h}_1^{l-1}}/\partial Z$ is a polynomially bounded function of Z_0 by Lemma A.13.4. σ' is polynomially bounded, and the expectations are finite by Lemma A.13.1 since the integrands are polynomially bounded functions of Z_0 as they are products of such functions by Lemma A.13.4 and by the induction hypothesis. We thus get that

$$\frac{\partial Z^{d\tilde{x}_2^{l-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded. We thus have that:

$$\frac{\partial Z^{d\tilde{h}_2^{l-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded, which proves claims (iv) and (v) at layer $l-1$. This thus concludes the induction, and with it the proof. □

The case $t \geq 2$

We have now treated the base case $t=2$ and are thus equipped to do the induction for $t \geq 2$. To make things easier we first introduce some equations. Let $t \geq 2$, we define the following assertions, where the different ψ appearing are assumed to be polynomially bounded:

Forward pass at time t:

$$(i) \quad Z^{h_t^1} = \psi\left(\widehat{Z}^{U^1\xi_0}, \ \ldots, \ \widehat{Z}^{U^1\xi_t}, \ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right) \tag{A.17}$$

For $l \in [2, L]$,

$$(i) \quad Z^{h_t^l} = \psi\left(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}},\ \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}\right) \tag{A.18}$$

$$(iii) \quad Z^{h_t^L} = \psi\left(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}\right) \tag{A.19}$$

For $l \in [2, L]$,

$$(iv) \quad \frac{\partial Z^{h_t^{l-1}}}{\widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}} = \psi\left(Z_0\right) \tag{A.20}$$

$$(v) \quad \frac{\partial Z^{h_t^l}}{\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}} = \psi\left(Z_0\right) \tag{A.21}$$

$$(vi) \quad Z^{W^l(0) x_t^{l-1}} = 0 \tag{A.22}$$

Backward pass at time t:

$$(i1) \quad Z^{d\tilde{x}_t^L} = \psi\left(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}\right) \tag{A.23}$$

$$(i2) \quad Z^{d\tilde{h}_t^L} = \psi\left(\widehat{Z}^{U^{L+1}},\ \widehat{Z}^{\widehat{W}^L \tilde{x}_0^{L-1}}\right) \tag{A.24}$$

For $l \in [3, L]$,

$$(ii1) \quad Z^{d\tilde{x}_t^{l-1}} = \psi\left(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}^{l-2}},\ \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}\right) \tag{A.25}$$

$$(ii2) \quad Z^{d\tilde{h}_t^{l-1}} = \psi\left(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}^{l-2}},\ \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}_0^l}\right) \tag{A.26}$$

$$(iii1) \quad Z^{d\tilde{x}_t^1} = \psi\left(\widehat{Z}^{U^1 \xi_0},\ \ldots,\ \widehat{Z}^{U^1 \xi_{t-1}},\ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right) \tag{A.27}$$

$$(iii2) \quad Z^{d\tilde{h}_t^1} = \psi\left(\widehat{Z}^{U^1 \xi_0},\ \ldots,\ \widehat{Z}^{U^1 \xi_t},\ \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}\right) \tag{A.28}$$

For $l \in [2, L]$,

$$(iv) \quad \frac{\partial Z^{d\tilde{h}_t^l}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}} = \psi(Z_0) \tag{A.29}$$

$$(v) \quad \frac{\partial Z^{d\tilde{h}_t^{l-1}}}{\partial \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}_0^l}} = \psi(Z_0) \tag{A.30}$$

$$(vi) \quad Z^{(W^l(0))^\top d\tilde{h}_t^l} = 0 \tag{A.31}$$

Note that we have proved in Appendix A.13 that all the assertions above hold for $t = 2$. Our goal is now to show by induction that they hold for any $t \geq 2$. For this we prove the following two lemmas. The proofs will essentially follow exactly the same pattern as for $t = 2$, the only difference is that the formulas will involve more terms, but since any finite sum of polynomially bounded functions is polynomially bounded, we will get the same results. Before proving the lemmas, we introduce the following quantities for $0 \leq s < t$:
For $l \in [2, L]$

$$\gamma^f_{s,t,l} := \begin{cases} \mathbb{E}[Z^{x_0^{l-1}} Z^{x_t^{l-1}}] & \text{if } \ s = 0 \\ \mathbb{E}[Z^{x_s^{l-1}} Z^{x_t^{l-1}}] & \text{otherwise} \end{cases} \tag{A.32}$$

For $l \in [1, L-1]$

$$\gamma^b_{s,t,l} := \mathbb{E}[Z^{d\tilde{h}_s^{l+1}} Z^{d\tilde{h}_t^{l+1}}] \tag{A.33}$$

$\gamma^f_{s,t,l}$ (*resp.* $\gamma^b_{s,t,l}$) will appear when expressing the variables of the l-th layer at time t in the forward (*resp.* backward) pass. We will show in the proofs that as for $t = 1$ and $t = 2$, those expectations are finite by Lemma A.13.1 because the integrands are polynomially bounded functions of Z_0 as they are products of such variables.

Lemma A.13.7 (Induction step in IP-LLR, forward pass)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Let $t \geq 2$, and assume that all of the assertions of Equation* (A.17) *up until Equation* (A.31) *hold for every time step $s \in [2, t]$. Then, the assertions of the forward pass, i.e., from Equation* (A.17) *up until Equation* (A.22)*, hold at time $t+1$.*

Proof. We follow the proof of Lemma A.13.5. By Theorem A.8.7, we have

$$Z^{h_{t+1}^1} = Z^{U^1 \xi_{t+1}} - \eta \sum_{s=0}^{t} \mathring{\chi}_s(\xi_s^\top \xi_{t+1}) Z^{d\tilde{h}_s^1}$$

By Lemma A.13.4 $Z^{d\tilde{h}_1^1} = \psi(\widehat{Z}^{U^1\xi_0}, \widehat{Z}^{U^1\xi_1}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2})$ and and by assumption we also have $Z^{d\tilde{h}_s^1} = \psi(Z^{U^1\xi_0}, \ldots, Z^{U^1\xi_s}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2})$ where the different ψ are polynomially bounded, which gives claim (i) at time $t+1$.

We have

$$\frac{\partial Z^{h_{t+1}^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} = -\eta \sum_{s=0}^{t} \mathring{\chi}_s(\xi_s^\top \xi_{t+1}) \frac{\partial Z^{d\tilde{h}_s^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}}$$

with

$$\frac{\partial Z^{d\tilde{h}_0^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} = \sigma'(Z^{U^1\xi_0})$$

which is a polynomially bounded function of Z_0 and so is $\partial Z^{d\tilde{h}_1^1}/\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}$ by Lemma A.13.4. In addition, by assumption, for $s \in [2,t]$, $\partial Z^{d\tilde{h}_s^1}/\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2} = \psi(Z_0)$ with ψ polynomially bounded. We thus get claim (iv) for $l=2$ at time $t+1$.

We have by Theorem A.8.7

$$Z^{h_{t+1}^2} = \mathring{\omega}_2 Z^{\widehat{W}^2 x_{t+1}^1} - \eta \sum_{s=0}^{t} \mathring{\chi}_s \gamma_{s,t+1,2}^f Z^{d\tilde{h}_s^2}$$

Now $Z^{x_{t+1}^1} = \sigma(Z^{h_{t+1)}^1}$ is a polynomially bounded function of Z_0 because $Z^{h_{t+1}^1}$ is and σ is polynomially bounded. Secondly, we have

$$\frac{\partial Z^{x_{t+1}^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} = \frac{\partial Z^{h_{t+1}^1}}{\partial \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}_0^2}} \sigma'(Z^{h_{t+1)}^1}$$

which is a polynomially bounded function of Z_0 by the previous results and because σ' is polynomially bounded. By Lemma A.13.2 we get that $\mathring{\omega}_2 Z^{\widehat{W}^2 x_{t+1}^1} = 0$ which gives claim (vi) for $l=2$ at time $t+1$. In addition, this yields

$$Z^{h_{t+1}^2} = -\eta \mathring{\chi}_0 \gamma_{0,t+1,2}^f Z^{d\tilde{h}_0^2} - \eta \sum_{s=0}^{t} \mathring{\chi}_s \gamma_{s,t+1,2}^f Z^{d\tilde{h}_s^2}$$

The expectations defining the γ^f are finite by Lemma A.13.1 since the integrands are polynomially bounded functions of Z_0, as they are products of such variables by the previous result on $Z^{x_{t+1}^1}$ and by the assumption. This gives claim (ii) for $l=2$ by the results for the backward passes at time $t=0$ and $t=1$ and by the assumptions. Let $Z \in \{\widehat{Z}^{\widehat{W}^2 \tilde{x}_0^1}, \widehat{Z}^{(\widehat{W}^3)^\top d\tilde{h}_0^3}\}$. We have

$$\frac{\partial Z^{h_{t+1}^2}}{\partial Z} = -\eta \sum_{s=0}^{t} \mathring{\chi}_s \gamma_{s,t+1,2}^f \frac{\partial Z^{d\tilde{h}_s^2}}{\partial Z}$$

$\partial Z^{d\tilde{h}_0^2}/\partial Z$ has a simple expression and is a polynomially bounded function of Z_0. Additionally, by the results of the backward pass for $t=1$, and by assumption, for $s \in [1,t]$, $\partial Z^{d\tilde{h}_s^2}/\partial Z = \psi(Z_0)$ with ψ polynomially bounded. Since the γ^f are finite, we thus get

$$\frac{\partial Z^{h_{t+1}^2}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded. This gives claims (iv) and (v) at time $t+1$.

Let $l \in [2, L-1]$ and assume claims (ii), (iv), (v), and (vi) for layer l at time $t+1$. Then, by Theorem A.8.7 we have:

$$Z^{h_{t+1}^{l+1}} = \mathring{\omega}_{l+1} Z^{\widehat{W}^{l+1} x_{t+1}^l} - \eta \sum_{s=0}^{t} \mathring{\chi}_s \gamma^f_{s,t+1,l+1} Z^{d\tilde{h}_s^{l+1}}$$

Now $Z^{x_{t+1}^l} = \sigma(Z^{h_{t+1}^l})$ is a polynomially bounded function of Z_0 because $Z^{h_{t+1}^l}$ is by the induction hypothesis and σ is polynomially bounded. Secondly, we have

$$\frac{\partial Z^{x_{t+1}^l}}{\partial \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}} = \frac{\partial Z^{h_{t+1}^l}}{\partial \widehat{Z}^{(\widehat{W}^{l+1})^\top d\tilde{h}_0^{l+1}}} \sigma'(Z^{h_{t+1}^l})$$

which is a polynomially bounded function of Z_0 by the induction hypothesis. By Lemma A.13.2 we get that $\mathring{\omega}_{l+1} Z^{\widehat{W}^2 x_{t+1}^l} = 0$ which gives claim (vi) for layer $l+1$ at time $t+1$. In addition, this yields

$$Z^{h_{t+1}^{l+1}} = -\eta \sum_{s=0}^{t} \mathring{\chi}_s \gamma^f_{s,t+1,l+1} Z^{d\tilde{h}_s^{l+1}}$$

The expectations defining the γ^f are finite by Lemma A.13.1 since the integrands are polynomially bounded functions of Z_0, as they are products of such variables by the assumption and by the induction hypothesis. If $l+1 = L$, we have, for any $s \in [0, s]$, $Z^{d\tilde{h}_s^{l+1}} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$ with ψ polynomially bounded, which shows

$$Z^{h_{t+1}^{l+1}} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$$

with ψ polynomially bounded, which gives claim (iii). If $l+1 \leq L-1$, for any $s \in [0, s]$, $Z^{d\tilde{h}_s^{l+1}} = \psi(\widehat{Z}^{\widehat{W}^{l+1} \tilde{x}_0^l}, \widehat{Z}^{(\widehat{W}^{l+2}()^\top d\tilde{h}_0^{l+2}})$ with ψ polynomially bounded, which shows

$$Z^{h_{t+1}^{l+1}} = \psi(\widehat{Z}^{\widehat{W}^{l+1} \tilde{x}_0^l}, \widehat{Z}^{(\widehat{W}^{l+2})^\top d\tilde{h}_0^{l+2}})$$

ψ polynomially bounded, which shows claim (ii) at layer $l+1$ for time $t+1$. Let $Z \in \{\widehat{Z}^{\widehat{W}^{l+1} \tilde{x}_0^l}, \widehat{Z}^{(\widehat{W}^{l+2})^\top d\tilde{h}_0^{l+2}}\}$. Note that the second value is only valid if $l+1 \leq L-1$. Whenever Z is well-defined, we have

$$\frac{\partial Z^{h_{t+1}^{l+1}}}{\partial Z} = -\eta \sum_{s=0}^{t} \mathring{\chi}_s \gamma^f_{s,t+1,l+1} \frac{\partial Z^{d\tilde{h}_s^{l+1}}}{\partial Z}$$

For both possible values of Z, $\partial Z^{d\tilde{h}_0^{l+1}} / \partial Z$ has a simple expression and is a polynomially bounded function of Z_0. $Z^{d\tilde{h}_1^{l+1}} / \partial Z$ is a polynomially bounded

function of Z_0 by the results of the backward pass at time $t = 1$ (Lemma A.13.4), and finally for $s \in [2, t]$, $Z^{d\tilde{h}_1^{l+1}}/\partial Z = \psi(Z_0)$ with ψ polynomially bounded by assumption. Since the γ^f are finite, this gives

$$\frac{\partial Z^{h_{t+1}^{l+1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded. This proves claim (iv) and (v) for layer $l + 1$ at time $t + 1$, and thus concludes the induction on l and with it the proof. □

Lemma A.13.8 (Induction step in IP-LLR, backward pass)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Let $t \geq 2$, and assume that all of the assertions of Equation* (A.17) *up until Equation* (A.31) *for every time step $s \in [2, t]$. Additionally assume that the assertions of the forward pass, i.e., from Equation* (A.17) *up until Equation* (A.22)*, hold at time $t+1$. Then, the assertions of the backward pass, i.e., from Equation* (A.23) *up until Equation* (A.31)*, hold at time $t + 1$.*

Proof. We follow the proof of Lemma A.13.6. We have:

$$Z^{d\tilde{x}_{t+1}^L} = \widehat{Z}^{U^{L+1}} - \eta \overset{\circ}{\chi}_0 Z^{\tilde{x}_0^L} - \eta \sum_{s=1}^{t} \overset{\circ}{\chi}_s Z^{x_t^L}$$

where $Z^{\tilde{x}_0^L} = \sigma(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$, $Z^{x_1^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$ with ψ polynomially bounded by Lemma A.13.3 and for $s \in [2, t]$, $Z^{x_s^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$ with ψ polynomially bounded by assumption. Combining all this gives

$$Z^{d\tilde{x}_{t+1}^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$$

with ψ polynomially bounded since σ is also polynomially bounded. Then

$$Z^{d\tilde{h}_{t+1}^L} = Z^{d\tilde{x}_{t+1}^L} \sigma'(Z^{h_{t+1}^L})$$

and since $Z^{h_{t+1}^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$ with ψ polynomially bounded by assumption, we get

$$Z^{d\tilde{h}_{t+1}^L} = \psi(\widehat{Z}^{U^{L+1}}, \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}})$$

with ψ polynomially bounded since σ' is also polynomially bounded. This thus proves claim (i). Now, we have

$$\frac{\partial Z^{d\tilde{h}_{t+1}^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} = \frac{\partial Z^{d\tilde{x}_{t+1}^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} \sigma'(Z^{h_{t+1}^L}) + Z^{d\tilde{x}_{t+1}^L} \frac{\partial Z^{h_{t+1}^L}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} \sigma''(Z^{h_{t+1}^L})$$

where $Z^{h^L_{t+1}}$, $\partial Z^{h^L_{t+1}}/\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}$, and $Z^{d\tilde{x}^L_{t+1}}$ are polynomially bounded functions of Z_0 by assumption and by the previous result on $Z^{d\tilde{x}^L_{t+1}}$. Additionally, we have

$$\frac{\partial Z^{d\tilde{x}^L_{t+1}}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} = -\eta \mathring{\chi}_0 \sigma'(\widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}) - \eta \sum_{s=1}^{t} \mathring{\chi}_s \frac{\partial Z^{h^L_s}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} \sigma'(Z^{h^L_s})$$

σ' is polynomially bounded and by the results of the forward pass at $t = 1$ (Lemma A.13.3) $\partial Z^{h^L_1}/\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}} = \psi(Z_0)$ with ψ polynomially bounded. In addition, by assumption, for any $s \in [2, t]$, $\partial Z^{h^L_s}/\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}} = \psi(Z_0)$ with ψ polynomially bounded. This thus gives

$$\frac{\partial Z^{d\tilde{x}^L_{t+1}}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} = \psi(Z_0)$$

with ψ polynomially bounded, and thus

$$\frac{\partial Z^{d\tilde{h}^L_{t+1}}}{\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}} = \psi(Z_0)$$

with ψ polynomially bounded since σ' and σ'' are polynomially bounded. This proves (iv) for $l = L$ at time $t + 1$.

We have:

$$Z^{d\tilde{x}^{L-1}_{t+1}} = \mathring{\omega}_L \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}^L_{t+1}} - \eta \mathring{\chi}_0 \gamma^b_{0,t+1,L-1} Z^{\tilde{x}_0^{L-1}} - \eta \sum_{s=1}^{t} \mathring{\chi}_s \gamma^b_{s,t+1,L-1} Z^{x_s^{L-1}}$$

From the previous step we have that both $Z^{d\tilde{h}^L_{t+1}}$ and $\partial Z^{d\tilde{h}^L_{t+1}}/\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{L-1}}$ are polynomially bounded functions of Z_0. By Lemma A.13.2, this first shows that $\mathring{\omega}_L \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}^L_{t+1}} = 0$, and thus gives (vi) for $l = L$, leading to:

$$Z^{d\tilde{x}^{L-1}_{t+1}} = -\eta \mathring{\chi}_0 \gamma^b_{0,t+1,L} Z^{\tilde{x}_0^{L-1}} - \eta \sum_{s=1}^{t} \mathring{\chi}_s \gamma^b_{s,t+1,L} Z^{x_s^{L-1}}$$

Now, $Z^{\tilde{x}_0^{L-1}} = \sigma(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-2}})$ and $Z^{x_1^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^L_0})$ with ψ polynomially bounded by Lemma A.13.3. In addition, we have or any $s \in [2, t]$, we get $Z^{x_s^{L-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^L_0})$ with ψ polynomially bounded by assumption since it is the case for $Z^{h_s^{L-1}}$ and σ is polynomially bounded. As always the expectations defining the γ^b are finite by Lemma A.13.1 because the integrands are polynomially bounded functions of Z_0 as products of such variables by the results for the backward pass at times $t = 0$ and $t = 1$, by the assumptions and by the previous result on $Z^{d\tilde{h}^L_{t+1}}$. Since σ is also polynomially bounded, this gives

$$Z^{d\tilde{x}^{L-1}_{t+1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1} \tilde{x}_0^{L-1}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^L_0})$$

Then, we have

$$Z^{d\tilde{h}_{t+1}^{L-1}} = Z^{d\tilde{x}_{t+1}^{L-1}} \sigma'(Z^{h_{t+1)}^{L-1}}$$

and since $Z^{h_{t+1}^{L-1}} = \psi(\widehat{Z}\widehat{W}^{l-1}\tilde{x}_0^{L-1}, \widehat{Z}(\widehat{W}^L)^\top d\tilde{h}_0^L)$ with ψ polynomially bounded by assumption

$$Z^{d\tilde{h}_2^{L-1}} = \psi(\widehat{Z}\widehat{W}^{l-1}\tilde{x}_0^{L-1}, \widehat{Z}(\widehat{W}^L)^\top d\tilde{h}_0^L)$$

with ψ polynomially bounded since σ' is also polynomially bounded. This thus proves claim (ii) for $l = L-1$. Now, let $Z \in \{\widehat{Z}\widehat{W}^{l-1}\tilde{x}_0^{L-1}, \widehat{Z}(\widehat{W}^L)^\top d\tilde{h}_0^L\}$. We have

$$\frac{\partial Z^{d\tilde{h}_{t+1}^{L-1}}}{\partial Z} = \frac{\partial Z^{d\tilde{x}_{t+1}^{L-1}}}{\partial Z}\sigma'(Z^{h_{t+1)}^{L-1}} + Z^{d\tilde{x}_{t+1}^{L-1}} \frac{\partial Z^{h_{t+1}^{L-1}}}{\partial Z}\sigma''(Z^{h_{t+1)}^{L-1}}$$

where $Z^{h_{t+1}^{L-1}}$, $\partial Z^{h_{t+1}^{L-1}}/\partial Z$, and $Z^{d\tilde{x}_{t+1}^{L-1}}$ are polynomially bounded functions of Z_0 by assumption and by the previous result. We have

$$\frac{\partial Z^{d\tilde{x}_{t+1}^{L-1}}}{\partial Z} = -\eta\mathring{\chi}_0\gamma^b_{0,t+1,L-1}\frac{\partial Z^{\tilde{h}_0^{L-1}}}{\partial Z}\sigma'(Z^{\tilde{h}_0^{L-1}}) - \eta\sum_{s=1}^{t}\mathring{\chi}_s\gamma^b_{s,t+1,L-1}\frac{\partial Z^{h_s^{L-1}}}{\partial Z}\sigma'(Z^{h_s^{L-1}})$$

For both possible values of Z, $\partial Z^{\tilde{h}_0^{L-1}}/\partial Z$ has a simple expression and is a polynomially bounded function of Z_0, as is $\tilde{h}_0^{L-1}$. In addition, $Z^{h_1^{L-1}}$ and $\partial Z^{h_1^{L-1}}/\partial Z$ are polynomially bounded functions of Z_0 by the results of the forward pass at $t = 1$, and finally, for $s \in [2, t]$, $Z^{h_1^{L-1}}$ and $\partial Z^{h_1^{L-1}}/\partial Z$ are polynomially bounded functions of Z_0 by assumption. Since the γ^b are finite and σ' is polynomially bounded, we get

$$\frac{\partial Z^{d\tilde{x}_{t+1}^{L-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded and thus

$$\frac{\partial Z^{d\tilde{h}_{t+1}^{L-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded since σ' and σ'' are polynomially bounded. This proves (iv) and (v) for $l = L-1$.

Let $l \in [2, L-1]$, and assume claims (ii), (iv), (v), are true at layer l and claim (vi) is true at layer $l+1$. We have:

$$Z^{d\tilde{x}_{t+1}^{l-1}} = \mathring{\omega}_l\widehat{Z}(\widehat{W}^l)^\top d\tilde{h}_{t+1}^l - \eta\mathring{\chi}_0\gamma^b_{0,t+1,l-1}Z^{\tilde{x}_0^{l-1}} - \eta\sum_{s=1}^{t}\mathring{\chi}_s\gamma^b_{s,t+1,l-1}Z^{x_s^{l-1}}$$

From the induction hypothesis we have that both $Z^{d\tilde{h}^l_{t+1}}$ and $\partial Z^{d\tilde{h}^l_{t+1}}/\partial \widehat{Z}^{\widehat{W}^l \tilde{x}_0^{l-1}}$ are polynomially bounded functions of Z_0. By Lemma A.13.2, this first shows that $\mathring{\omega}_l \widehat{Z}^{(\widehat{W}^l)^\top d\tilde{h}^l_{t+1}} = 0$, and thus gives (vi) for layer l, leading to:

$$Z^{d\tilde{x}^{l-1}_{t+1}} = -\eta \mathring{\chi}_0 \gamma^b_{0,t+1,l-1} Z^{\tilde{x}_0^{l-1}} - \eta \sum_{s=1}^{t} \mathring{\chi}_s \gamma^b_{s,t+1,l-1} Z^{x_s^{l-1}}$$

Now, if $l-1 \geq 2$, $Z^{\tilde{x}_0^{l-1}} = \sigma(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}})$ and by Lemma A.13.3, we also have that $Z^{x_1^{l-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^l_0})$ with ψ polynomially bounded because it is the case for $Z^{h_1^{l-1}}$ and σ is polynomially bounded. In addition, by assumption, we have for any $s \in [2,t]$, $Z^{x_s^{l-1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^l_0})$ with ψ polynomially bounded since it is the case for $Z^{h_s^{l-1}}$ and σ is polynomially bounded. As always the expectations defining the γ^b are finite by Lemma A.13.1 because the integrands are polynomially bounded functions of Z_0 as products of such variables by the results of the backward passes at times $t=0$ and $t=1$ and by the induction hypothesis. We thus get

$$Z^{d\tilde{x}^{l-1}_{t+1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^l_0})$$

with ψ polynomially bounded. On the other hand, if $l-1=1$, we have $Z^{\tilde{x}_0^1} = \sigma(\widehat{Z}^{U^1\xi_0})$ and by Lemma A.13.3, we have $Z^{x_1^1} = \psi(\widehat{Z}^{U^1\xi_0}, \widehat{Z}^{U^1\xi_1}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}^2_0})$ with ψ polynomially bounded. In addition, by assumption we have for $s \in [2,t]$, $Z^{x_s^1} = \psi(\widehat{Z}^{U^1\xi_0}, \ldots, \widehat{Z}^{U^1\xi_s}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}^2_0})$ with ψ polynomially bounded. Since σ is also polynomially bounded, this gives

$$Z^{d\tilde{x}^1_{t+1}} = \psi(\widehat{Z}^{U^1\xi_0}, \ldots, \widehat{Z}^{U^1\xi_t}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}^2_0})$$

ψ polynomially bounded. Since $Z^{d\tilde{h}^{l-1}_{t+1}} = Z^{d\tilde{x}^{l-1}_{t+1}} \sigma'(Z^{h^{l-1}_{t+1}})$, and by assumption $Z^{h^{l-1}_{t+1}} = \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^l_0})$ if $l-1 \geq 2$, and otherwise $Z^{h^1_{t+1}} = \psi(\widehat{Z}^{U^1\xi_0}, \ldots, \widehat{Z}^{U^1\xi_{t+1}}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}^2_0})$, we get

$$Z^{d\tilde{h}^{l-1}_{t+1}} = \begin{cases} \psi(\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^l_0}) \text{ if } l-1 \geq 2 \\ \psi(\widehat{Z}^{U^1\xi_0}, \widehat{Z}^{U^1\xi_1}, \ldots, \widehat{Z}^{U^1\xi_{t+1}}, \widehat{Z}^{(\widehat{W}^2)^\top d\tilde{h}^2_0}) \text{ if } l-1=1 \end{cases}$$

This gives claim (ii) for layer $l-1$ and claim (iii) for the case when $l-1=1$. Now, let $Z \in \{\widehat{Z}^{\widehat{W}^{l-1}\tilde{x}_0^{l-2}}, \widehat{Z}^{(\widehat{W}^L)^\top d\tilde{h}^l_0}\}$. We have

$$\frac{\partial Z^{d\tilde{h}^{l-1}_{t+1}}}{\partial Z} = \frac{\partial Z^{d\tilde{x}^{l-1}_{t+1}}}{\partial Z} \sigma'(Z^{h^{l-1}_{t+1}}) + Z^{d\tilde{x}^{l-1}_{t+1}} \frac{\partial Z^{h^{l-1}_{t+1}}}{\partial Z} \sigma''(Z^{h^{l-1}_{t+1}})$$

where $Z^{h^{l-1}_{t+1}}$ and $Z^{d\tilde{x}^{l-1}_{t+1}}$ are polynomially bounded functions of Z_0 by assumption and by the previous result on $Z^{d\tilde{x}^{l-1}_{t+1}}$. Also by assumption, we have

$$\frac{\partial Z^{h^{l-1}_{t+1}}}{\partial Z} = \begin{cases} 0 \text{ if } l-1=1 \text{ and } Z = \widehat{Z}^{\widehat{W}^2\tilde{x}_0^1} \\ \psi(Z_0) \text{ otherwise} \end{cases}$$

with ψ polynomially bounded. In any case, $\partial Z^{h_{t+1}^{l-1}}/\partial Z$ is a polynomially bounded function of Z_0. On the other hand, we have

$$\frac{\partial Z^{d\tilde{x}_2^{l-1}}}{\partial Z} = -\eta \mathring{\chi}_0 \gamma^b_{0,t+1,l-1} \frac{\partial Z^{\tilde{h}_0^{l-1}}}{\partial Z} \sigma'(Z^{\tilde{h}_0^{l-1}}) - \eta \sum_{s=1}^{t} \mathring{\chi}_t \gamma^b_{s,t+1,l-1} \frac{\partial Z^{h_s^{l-1}}}{\partial Z} \sigma'(Z^{h_s^{l-1}})$$

For both possible values of Z, $\partial Z^{\tilde{h}_0^{l-1}}/\partial Z$ has an easy expression and is a polynomially bounded function of Z_0 (essentially because σ and its derivatives are polynomially bounded). On the other hand, $\partial Z^{\tilde{h}_s^{l-1}}/\partial Z$ is a polynomially bounded function of Z_0 by assumption. σ' is polynomially bounded, and the γ^b are finite. We thus get that

$$\frac{\partial Z^{d\tilde{x}_{t+1}^{l-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded, and thus:

$$\frac{\partial Z^{d\tilde{h}_{t+1}^{l-1}}}{\partial Z} = \psi(Z_0)$$

with ψ polynomially bounded, which proves claims (iv) and (v) at layer $l-1$ for time $t+1$. This thus concludes the induction on l, and with it the proof. □

A.13.1 . Main result

Theorem A.13.9 (Multiplications by the initial weight matrices vanish in IP-LLR for $t \geq 1$)**.** *Consider the IP-LLR parameterization with a positively p-homogeneous activation function, and $p \geq 2$. Then, for any $t \geq 1$, and for any $l \in [2, L]$, one has:*

$$\begin{cases} Z^{W^l(0)x_t^{l-1}} = \mathring{\omega}_l Z^{\widehat{W}^l x_t^{l-1}} = 0 \\ Z^{(W^l(0))^\top d\tilde{h}_t^l} = \mathring{\omega}_l Z^{(\widehat{W}^l)^\top d\tilde{h}_t^l} = 0 \end{cases}$$

Proof. The result for $t = 1$ has essentially been proved already early on in Corollary A.13.4.1 (which stems from Lemmas A.13.3 and A.13.4). For $t = 2$, the result has been proved in Lemmas A.13.5 and A.13.6. Then we can prove the result for any $t \geq 2$ by induction using Lemmas A.13.7 and A.13.8. □

A.14 . Expectations with ReLU

In all this section, we consider $Z \sim \mathcal{N}(0, \sigma^2)$, so that $Z = \sigma U$ where $U \sim \mathcal{N}(0, 1)$.

A.14.1 . First moment

For $\phi(z) = \max(0, z)$ and $Z \sim \mathcal{N}(0, \sigma^2)$, we have

$$\mathbb{E}[\phi(Z)] = \mathbb{E}[\phi(\sigma U)] = \frac{\sigma}{\sqrt{2\pi}} \int_0^\infty u e^{-u^2/2} \mathrm{d}u = \frac{\sigma}{\sqrt{2\pi}}.$$

A.14.2 . Second moment

For $\phi(z) = \max(0, z)$ and $Z \sim \mathcal{N}(0, \sigma^2)$, we have

$$\mathbb{E}[\phi(Z)^2] = \frac{1}{2}\mathbb{E}[Z^2] = \frac{\sigma^2}{2}.$$

A.14.3 . First forward pass moments

We have, for any $l \in [1, L]$, with $\sigma_0 := \sqrt{||\xi_0||^2+1}$,

$$\mathbb{E}[\widehat{Z}^{\tilde{h}_0^l}] = 0, \quad \mathbb{E}[(\widehat{Z}^{\tilde{h}_0^l})^2] = \frac{\sigma_0^2}{2^{l-1}}$$
$$\mathbb{E}[Z^{\tilde{x}_0^l}] = \frac{\sigma_0}{\sqrt{2^l\pi}}, \quad \mathbb{E}[(Z^{\tilde{x}_0^l})^2] = \frac{\sigma_0^2}{2^l}$$

A.14.4 . First derivative moments

For $\phi(z) = \max(0, z)$, we have $\phi'(z) = \mathbb{1}_{z\geq 0}$ almost everywhere, so for $Z \sim \mathcal{N}(0, \sigma^2)$, we have

$$\mathbb{E}[\phi'(Z)] = \mathbb{P}(Z \geq 0) = 1/2.$$

Note that since $\phi'(z)^p = \phi'(z)$ for any $p > 0$, all the moments of $\phi'(Z)$ are equal to the first moment.

A.14.5 . First backward pass moments

We have, for any $l \in [1, L]$, with,

$$\mathbb{E}[\widehat{Z}^{d\tilde{x}_0^l}] = 0, \quad \mathbb{E}[(\widehat{Z}^{d\tilde{x}_0^l})^2] = \frac{1}{2^{L-l}}$$
$$\mathbb{E}[Z^{d\tilde{h}_0^l}] = 0, \quad \mathbb{E}[(Z^{d\tilde{h}_0^l})^2] = \frac{1}{2^{L-l+1}}$$

B - Appendix for Chapter 3

B.1 . Additional notations and preliminary results

B.1.1 . Notations for the appendix

We introduce in this section additional notation that we use throughout the Appendix.

Residual: we call $R_t(y) := -\partial_2 \ell(f^*(y), f(\mu_t; y))$, the "residual", which is equal to the difference $f^*(y) - f(\mu_t; y)$ when ℓ is the squared loss.

Identity matrix: we denote by I_p the identity matrix in $\mathbb{R}^{p \times p}$ for any $p \in \mathbb{N}$.

Indicator functions: we denote by $\mathbf{1}_A$ the indicator of a set A, that is $\mathbf{1}_A(z) = 1 \iff z \in A$, and $\mathbf{1}_A(z) = 0$ otherwise.

Total variation: for any measure ν, we denote by $|\nu|$ its total variation, which should cause no confusion with the absolute value given the context.

Beta / Gamma function and distribution: for $\alpha, \beta > 0$, we denote by $B(\alpha, \beta)$ the Beta function equal to $\Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$ where Γ is the Gamma function, and by Beta(α, β) the beta law with density equal to $u^{\alpha-1}u^{\beta-1}/B(\alpha, \beta)$ on $[0, 1]$.

Gaussian / spherical measures: we call ρ_p the standard Gaussian measure in $\mathbb{R}^p$ (corresponding to $\mathcal{N}(0, I_p)$) for any $p \in \mathbb{N}$.

Whenever $\tau \in \mathcal{M}_+(\Omega)$ has finite and non-zero total variation, we denote by $\tilde{\tau} \in \mathcal{P}_2(\Omega)$ its normalized counterpart (which is a probability measure), that is $\tilde{\tau} = \tau/\tau(\Omega) = \tau/|\tau|$.

For any $p \in \mathbb{N}$, we call ω_p the Lebesgue (spherical) measure over the unit sphere $\mathbb{S}^{p-1}$ of $\mathbb{R}^p$, that is the measure such that $\tilde{\omega}_p$ is the *uniform* measure on $\mathbb{S}^{p-1}$. We then denote by $|\mathbb{S}^{p-1}|$ the surface area of $\mathbb{S}^{p-1}$, that is $|\mathbb{S}^{p-1}| := |\omega_p| = \omega_p(\mathbb{S}^{p-1}) = 2\pi^{p/2}/\Gamma(p/2)$.

Smooth functions: we denote by $\mathcal{C}(\Omega)$ (*resp.* $\mathcal{C}^1_c(\Omega)$) the set of continuous (*resp.* continuously differentiable and compactly supported) functions from a set Ω to $\mathbb{R}$.

B.1.2 . General results on invariance for measures and functions

In this section, we list a number of lemmas related to symmetries of measures and functions which will prove helpful in the proofs presented in the Appendix.

Lemma B.1.1 (Invariance under invertible maps)**.** *Let μ be a measure invariant under some measurable and invertible map T. Then, assuming T^{-1} is also measurable, one has that μ is also invariant under T^{-1}.*

Remark. A similar result holds for a function f invariant under an invertible map.

Proof. Because μ is invariant under T, we have for any measurable set A, $\mu(A) = \mu(T^{-1}(A))$. Since T^{-1} is assumed to be measurable, for any measurable set A, $T(A)$ is also measurable ($T(A) = (T^{-1})^{-1}(A)$) and thus $\mu(T(A)) = \mu(T^{-1}(T(A)) = \mu(A)$ which shows μ is invariant under T^{-1}. □

Lemma B.1.2 (Invariance of the density)**.** *Let ν be a measure with density p w.r.t. some measure μ, and assume both ν and μ are σ-finite and invariant under some measurable and invertible map T, whose inverse T^{-1} is also measurable. Then p is also invariant under T μ-almost everywhere, i.e., $p(T(x)) = p(x)$ for μ-almost every x.*

Proof. For any measurable φ (*w.r.t.* μ, and thus *w.r.t.* ν as well), $\varphi \circ T^{-1}$ is also measurable, and we have, on the one hand

$$\int \varphi \circ T^{-1} \mathrm{d}\nu = \int \left(\varphi \circ T^{-1}\right) p \, \mathrm{d}\mu = \int \varphi \left(p \circ T\right) \mathrm{d}\mu,$$

and on the other hand

$$\int \varphi \circ T^{-1} \mathrm{d}\nu = \int \varphi \mathrm{d}\nu = \int \varphi \, p \, \mathrm{d}\mu,$$

which shows that $\int \varphi \left(p \circ T\right) \mathrm{d}\mu = \int \varphi \, p \, \mathrm{d}\mu$, and thus that $p \circ T = p$ μ-almost everywhere. □

Lemma B.1.3 (Projected variance with spherical symmetry)**.** *Let ζ be a spherically symmetric measure on $\mathbb{R}^p$ (i.e., such that for any orthogonal linear map $T \in \mathcal{O}(p)$, $T_{\#}\zeta = \zeta$), with finite second moment. Then we have the following matrix identity:*

$$\int_z z z^\top \mathrm{d}\zeta(z) = v_\zeta I_p, \qquad v_\zeta := \int_z (z_1)^2 \mathrm{d}\zeta(z) = \frac{1}{p} \int_z ||z||^2 \mathrm{d}\zeta(z).$$

Proof. The (i,j)-th entry of the matrix on the left-hand-side is $\int_z z_i z_j \mathrm{d}\zeta(z)$, and it is readily seen that the terms outside the diagonal are 0. Indeed, let $(i,j) \in [1,p]^2$ with $i \neq j$, and consider the orthogonal map $T_j : z \in \mathbb{R}^p \mapsto (z_1, \ldots, z_{j-1}, -z_j, z_{j+1}, \ldots, z_p)^\top$. The spherical symmetry of ρ implies that it

is invariant under T_j, which yields $\int_z z_i z_j \mathrm{d}\rho(z) = -\int_z z_i z_j \mathrm{d}\rho(z)$, thereby showing that the latter is 0. To see that the diagonal terms are all equal, it suffices to consider the orthogonal map S_i which swaps the 1st and i-th coordinates of a vector z. The invariance of ρ under S_i yields $\int_z (z_1)^2 \mathrm{d}\rho(z) = \int_z (z_i)^2 \mathrm{d}\rho(z)$, which concludes the proof. □

B.1.3 . A disintegration result on the unit sphere $\mathbb{S}^{d-1}$

Consider a $u \in \mathbb{S}^{d-1}$. u is determined by: (i) its angle $\theta := \arccos(||u^H||) \in [0, \pi/2]$ with H (*i.e.*, its angle with its projection u^H onto H), (ii) the direction $z^H = u^H/||u^H|| \in \mathbb{S}^{d_H-1}$ of its projection u^H onto H, and finally (iii) the direction $z^\perp = u^\perp/||u^\perp|| \in \mathbb{S}^{d_\perp-1}$ of its projection $u^\perp$ onto $H^\perp$. Since $||u^H||^2 + ||u^\perp||^2 = 1$, the angle θ gives both the norms of the projections onto H and $H^\perp$: $||u^H|| = \cos(\theta)$ and $||u^\perp|| = \sin(\theta)$.

When z ranges over the unit sphere $\mathbb{S}^{d-1}$, the angle θ and the directions $z^H, z^\perp$ range over $[0, \pi/2]$, $\mathbb{S}^{d_H-1}$, and $\mathbb{S}^{d_\perp-1}$ respectively. We wish to understand what measures we obtain on these three sets when z is distributed on the sphere according to the Lebesgue measure ω_d. We show below below that after the change of coordinates described above (from $u \in \mathbb{S}^{d-1}$ to $(\theta, z^H, z^\perp) \in [0, \pi/2] \times \mathbb{S}^{d_H-1} \times \mathbb{S}^{d_\perp-1}$), the corresponding measures over $\mathbb{S}^{d_H-1}$ and $\mathbb{S}^{d_\perp-1}$ are uniform measures and the measure over θ is given by a push-forward of a Beta distribution as defined below:

Definition B.1.1 (Distribution γ of the angle θ)**.** We define the measure γ on $[0, \pi/2]$ with the following density *w.r.t.* the Lebesgue measure on $[0, \pi/2]$:

$$\mathrm{d}\gamma(\theta) := \cos(\theta)^{d_H-1} \sin(\theta)^{d_\perp-1} \mathrm{d}\theta.$$

Remark. γ is in fact simply given by $(\arccos \circ \sqrt{\cdot})_\# \mathrm{Beta}(d_H/2, d_\perp/2)$. Note that the total variation of gamma is $|\gamma| = \gamma([0, \pi/2]) = \frac{1}{2} B(\frac{d_H}{2}, \frac{d_\perp}{2})$, and the corresponding normalized (probability) measure is

$$\mathrm{d}\tilde{\gamma}(\theta) = \mathrm{d}\gamma(\theta)/|\gamma| = \frac{2}{B(\frac{d_H}{2}, \frac{d_\perp}{2})} \cos(\theta)^{d_H-1} \sin(\theta)^{d_\perp-1} \mathrm{d}\theta.$$

We now state the disintegration theorem and give its proof:

Theorem B.1.4 (Disintegration of the Lebesgue measure on the sphere)**.** *Let ω_d denote the Lebesgue measure on the sphere measure on the sphere of $\mathbb{R}^d$, and let γ be the measure of Definition B.1.1. Then, one has*

$$\omega_d = \Phi_\#(\omega_{d_H} \otimes \omega_{d_\perp} \otimes \gamma)$$

where

$$\begin{aligned} \Phi : [0, \pi/2] \times \mathbb{S}^{d_H-1} \times \mathbb{S}^{d_\perp-1} &\to \mathbb{S}^{d-1} \\ (\theta, \quad z_H, \quad z_\perp) &\mapsto \cos(\theta) z_H + \sin(\theta) z_\perp. \end{aligned}$$

Proof. Denoting $\tilde{\omega}_d$ the uniform measure on the sphere, $|\mathbb{S}^{d-1}|:= \frac{2\pi^{d/2}}{\Gamma(d/2)}$ the surface are of the sphere in dimension d, and ρ_p the standard Gaussian distribution in $\mathbb{R}^p$ for any p. Using the well-known fact that $\tilde{\omega}_d = \Pi_{\#}\rho_d$ with $\Pi : x \in \mathbb{R}^d \backslash \{0\} \mapsto x/||x|| \in \mathbb{S}^{d-1}$, we have, for any measurable test function $\varphi : \mathbb{S}^{d-1} \to \mathbb{R}$,

$$\begin{aligned}
\int \varphi \mathrm{d}\omega_d &= |\mathbb{S}^{d-1}| \int \varphi d\tilde{\omega}_d \\
&= |\mathbb{S}^{d-1}| \int_x \varphi\left(\frac{x}{||x||}\right) \mathrm{d}\rho_d(x) \\
&= |\mathbb{S}^{d-1}| \int_{x_H, x_\perp} \varphi\left(\frac{x_H + x_\perp}{||x_H + x_\perp||}\right) \mathrm{d}\rho_{d_H}(x_H) \mathrm{d}\rho_{d_\perp}(x_\perp) \\
&= C_d \int \varphi\left(\frac{r_H z_H + r_\perp z_\perp}{||r_H z_H + r_\perp z_\perp||}\right) r_H^{d_H - 1} e^{-r_H^2/2} r_\perp^{d_\perp - 1} e^{-r_\perp^2/2} \mathrm{d}r_H \mathrm{d}r_\perp \mathrm{d}\omega_{d_H}(z_H) \mathrm{d}\omega_{d_\perp}(z_\perp) \\
&= C_d \int_{z_H, z_\perp} \int_{r_H, r_\perp} \varphi\left(\frac{r_H z_H + r_\perp z_\perp}{\sqrt{r_H^2 + r_\perp^2}}\right) r_H^{d_H - 1} r_\perp^{d_\perp - 1} e^{-(r_H^2 + r_\perp^2)/2} \mathrm{d}r_H \mathrm{d}r_\perp \mathrm{d}\omega_{d_H}(z_H) \mathrm{d}\omega_{d_\perp}(z_\perp),
\end{aligned}$$

with

$$C_d := \frac{|\mathbb{S}^{d-1}|}{(2\pi)^{d_H/2}(2\pi)^{d_\perp/2}} = \frac{|\mathbb{S}^{d-1}|}{(2\pi)^{d/2}} = \frac{2\pi^{d/2}}{2^{d/2}\pi^{d/2}\Gamma(d/2)} = \frac{1}{2^{(d-2)/2}\Gamma(d/2)}.$$

Doing the polar change of variables $(r_H, r_\perp) \in \mathbb{R}_+^2 \to (R, \theta) \in \mathbb{R}_+ \times [0, \pi/2]$, we get:

$$\int \varphi \mathrm{d}\omega_d = C'_d \int_{z_H, z_\perp} \int_\theta \varphi\left(\cos(\theta) z_H + \sin(\theta) z_\perp\right) \cos(\theta)^{d_H - 1} \sin(\theta)^{d_\perp - 1} d\theta \mathrm{d}\omega_{d_H}(z_H) \mathrm{d}\omega_{d_\perp}(z_\perp)$$

where

$$\begin{aligned}
C'_d :=& C_d \int_0^{+\infty} R^{d-2} e^{-R^2/2} R dR \\
=& C_d \int_0^{+\infty} R^{d-1} e^{-R^2/2} dR \\
=& C_d \times 2^{(d-2)/2} \Gamma(d/2) \\
=& 1.
\end{aligned}$$

which concludes the proof. □

Remark. A similar disintegration result holds for the uniform measure $\tilde{\omega}_d$ on the sphere. The corresponding measures which are then pushed-forward by the same Φ are the normalized counterparts of the measures in the theorem above: $\tilde{\omega}_d = \Phi_{\#}(\tilde{\omega}_{d_H} \otimes \tilde{\omega}_{d_\perp} \otimes \tilde{\gamma})$. This readily comes from noting that a simple calculation yields $|\omega_d| = |\omega_{d_H}|\,|\omega_{d_\perp}|\,|\gamma|$.

B.2 . Gradient flows on the space of probability measures

B.2.1 . First variation of a functional over measures

Given a functional $F : \mathcal{P}_2(\mathbb{R}^p) \to \mathbb{R}$, its *first variation* or *Fréchet derivative* at $\mu \in \mathcal{P}_2(\mathbb{R}^p)$ is defined as a measurable function, denoted $\frac{\delta F}{\delta \mu}(\mu) : \mathbb{R}^p \to \mathbb{R}$, such that, for any $\nu \in \mathcal{P}_2(\mathbb{R}^p)$ for which $\mu + t\nu \in \mathcal{P}_2(\mathbb{R}^p)$ in a neighborhood (in t) of $t = 0$,

$$\frac{d}{dt} F(\mu + t\nu)\Big|_{t=0} = \int_z \frac{\delta F}{\delta \mu}(\mu)[z] \mathrm{d}\nu(z).$$

See Santambrogio (Santambrogio, 2015, Definition 7.12), or (Santambrogio, 2017, p.29) for more details on the first variation.

In the case of the functional defined in Equation (3.2) corresponding to the population loss objective, using the differentiability of the loss ℓ *w.r.t.* its second argument, one readily has that

$$F'_\mu(c) := \frac{\delta F}{\delta \mu}(\mu)[c] = \int_x \partial_2 \ell(f^*(x), f(\mu; x)) \phi(c; x) \mathrm{d}\rho(x)$$

since

$$\frac{d}{dt} \ell\Big(f^*(x), f(\mu; x) + t f(\nu; x)\Big) = \partial_2 \ell\Big(f^*(x), f(\mu; x) + t f(\nu; x)\Big) \int_c \phi(c; x) \mathrm{d}\nu(c).$$

B.2.2 . Wasserstein gradient flows in the space $\mathcal{P}_2(\mathbb{R}^{d+1})$

A Wasserstein gradient flow for the objective F defined in Equation (3.2) is a path $(\mu_t)_{t \geq 0}$ in the space of probability measures $\mathcal{P}_2(\mathbb{R}^{d+1})$ which satisfies the continuity equation with a vector field v_t which is equal to the opposite of the gradient of the first variation of the functional F. This means that we have, in the sense of distributions,

$$\partial_t \mu_t = -\mathrm{div}\left(-\nabla\left(\frac{\delta F}{\delta \mu}\right)\mu_t\right).$$

That a pair $((\mu_t)_{t \geq 0}, v_t)$ consisting of a path in $\mathcal{P}_2(\mathbb{R}^p)$ and a (time-dependent) vector field in $\mathbb{R}^p$ satisfies the continuity equation $\partial_t \mu_t = -\mathrm{div}(v_t \mu_t)$ in the sense of the distributions simply means that for any test function $\varphi \in \mathcal{C}^1_c(\mathbb{R}^p)$,

$$\partial_t \int \varphi \, \mathrm{d}\mu_t = \int v_t^\top \nabla \varphi \, \mathrm{d}\mu_t,$$

where ∂_t stands for the time derivative $\frac{d}{dt}$. Similarly, when we say that the advection-reaction equation $\partial_t \mu_t = -\mathrm{div}\,(v_t \mu_t) + g_t \mu_t$ is satisfied for some function $g_t : \mathbb{R}^p \to \mathbb{R}$, we mean that it is in the sense of distributions: for any test function $\varphi \in \mathcal{C}^1_c(\mathbb{R}^p)$,

$$\partial_t \int \varphi \, \mathrm{d}\mu_t = \int (v_t^\top \nabla \varphi + g_t) \, \mathrm{d}\mu_t.$$

An alternative description of the Wasserstein gradient flow of the objective F is to consider a flow $X_\bullet(\cdot)$ in $\mathbb{R}_+ \times \mathbb{R}^{d+1}$ such that, for any $c \in \mathbb{R}^{d+1}$,

$$\begin{aligned} X_0(c) &= c \\ \frac{d}{dt} X_t(c) &= -\nabla \left(\frac{\delta F}{\delta \mu} \right) (X_t(c)) \end{aligned}$$

and to define $\mu_t = (X_t)_\# \mu_0$.

For more details on Wasserstein gradient flows in the space of probability measures see Santambrogio (Santambrogio, 2015, Section 5.3), and (Santambrogio, 2017, Section 4), and for more details on the equivalence between the continuity equation and the flow-based representation of the solution see Santambrogio (Santambrogio, 2015, Theorem 4.4).

B.3 . Proofs of the symmetry results of Section 3.2

There are two main ideas behind the proof. Call $\tilde{T} : (a, b) \in \mathbb{R} \times \mathbb{R}^d \mapsto (\pm a, T(b))$ (depending on whether f^* is invariant or anti-invariant under T) and consider the following two facts:

Structure of $\phi((a, b); x)$. Since T is orthogonal, so is $\tilde{T}$, and the structure of $\phi((a, b); x) = a\sigma(b^\top x)$ is such that $\phi(\tilde{T}(a, b); x) = \pm\phi((a, b); T^{-1}(x))$ because T is orthogonal (its adjoint is thus its inverse).

Conjugate gradients. Computing the gradient of a function whose input has been transformed by $\tilde{T}^{-1}$ is the same as the conjugate action of $\tilde{T}$ on the gradient: $\nabla(\varphi \circ \tilde{T}^{-1}) = \tilde{T} \circ (\nabla\varphi) \circ \tilde{T}^{-1}$ (this is due to the fact that the adjoint of $\tilde{T}^{-1}$ is $\tilde{T}$ because $\tilde{T}$ is orthogonal). Note that we similarly get $\nabla(\varphi \circ \tilde{T}) = \tilde{T}^{-1} \circ (\nabla\varphi) \circ \tilde{T}$.

B.3.1 . Preliminaries

We present here arguments that are present in both the proofs of Proposition 3.2.1 and 3.2.2. Let T be a linear orthogonal map such that $f^*(T(x)) = \pm f^*(x)$, where the $\pm$ is because we deal with both cases at the same time since the logic is the same. Let $t \geq 0$, and define $\nu_t := \tilde{T}^{-1}_\# \mu_t$. We aim to show that $(\nu_t)_{t \geq 0}$ is also a Wasserstein gradient flow for the same objective as $(\mu_t)_{t \geq 0}$.

Prediction function. Let $x \in \mathbb{R}^d$. We have, using the fact that T is orthogonal (and thus that $\langle T(x),\, y\rangle = \big\langle x,\, T^{-1}(y)\big\rangle$),

$$\begin{aligned} f(\nu_t; x) &= \int_{a,b} a\sigma(b^\top x)\mathrm{d}\nu_t(a,b) \\ &= \int_{a,b} \pm a\sigma(T^{-1}(b)^\top x)\mathrm{d}\mu_t(a,b) \\ &= \pm \int_{a,b} a\sigma(b^\top T(x))\mathrm{d}\mu_t(a,b) \\ &= \pm f(\mu_t; T(x)). \end{aligned}$$

Time derivative. Let $\varphi \in \mathcal{C}^1_c(\mathbb{R}^d)$. Because μ_t satisfies the continuity Equation (3.3) in the sense of distributions, and using the remark above on conjugate gradients as well as the orthogonality of $\tilde{T}$, we have:

$$\begin{aligned} \partial_t \int \varphi \mathrm{d}\nu_t &= \partial_t \int \varphi \circ \tilde{T}^{-1} \mathrm{d}\mu_t \\ &= \int \Big\langle \nabla(\varphi \circ \tilde{T}^{-1}),\, v_t \Big\rangle \mathrm{d}\mu_t \\ &= \int \Big\langle \tilde{T} \circ \nabla\varphi \circ \tilde{T}^{-1},\, v_t \Big\rangle \mathrm{d}\mu_t \\ &= \int \Big\langle \nabla\varphi \circ \tilde{T}^{-1},\, \tilde{T}^{-1} \circ v_t \Big\rangle \mathrm{d}\mu_t \\ &= \int \Big\langle \nabla\varphi,\, \tilde{T}^{-1} \circ v_t \circ \tilde{T} \Big\rangle \mathrm{d}\nu_t. \end{aligned}$$

Conjugate velocity field. The equality above actually shows that ν_t satisfies the continuity equation with the conjugate velocity field $\tilde{T}^{-1} \circ v_t \circ \tilde{T}$ instead of v_t. We show below that the former is closely related to the latter (and is in fact equal to $-\nabla F'_{\nu_t}$ with sufficient assumptions on $\partial_2 \ell$, which is the step proven in Appendices B.3.2 and B.3.3). Indeed, because v_t is a gradient: $v_t = -\nabla F'_{\mu_t}$, we have using again the remark above on conjugate gradients:

$$\tilde{T}^{-1} \circ v_t \circ \tilde{T} = -\nabla \left(F'_{\mu_t} \circ \tilde{T} \right).$$

Computing the function on the right-hand-side, for any $(a,b) \in \mathbb{R} \times \mathbb{R}^d$, we get, using the remark above on the structure of ϕ,

$$\begin{aligned} F'_{\mu_t}(\tilde{T}(a,b)) &= \int_y \partial_2 \ell\Big(f^*(y), f(\mu_t; y)\Big) \phi\Big(\tilde{T}(a,b); y\Big) \mathrm{d}\rho(y) \\ &= \pm \int_y \partial_2 \ell\Big(f^*(y), f(\mu_t; y)\Big) \phi\Big((a,b); T^{-1}(y)\Big) \mathrm{d}\rho(y). \end{aligned}$$

ρ is invariant under T since it spherically symmetric by assumption (and thus invariant under any orthogonal map) and we can therefore replace y by $T(y)$ in

the integral above, which yields

$$F'_{\mu_t}(\tilde{T}(a,b)) = \pm \int_y \partial_2 \ell\Big(f^*(T(y)), f(\mu_t; T(y))\Big)\phi\Big((a,b); y\Big)\mathrm{d}\rho(y)$$
$$= \pm \int_y \partial_2 \ell\Big(\pm f^*(y), \pm f(\nu_t; y)\Big)\phi\Big((a,b); y\Big)\mathrm{d}\rho(y),$$

and thus we get

$$\nabla\left(F'_{\mu_t} \circ \tilde{T}\right)(a,b) = \pm \int_y \partial_2 \ell\Big(\pm f^*(y), \pm f(\nu_t; y)\Big)\nabla_{(a,b)}\phi\Big((a,b); y\Big)\mathrm{d}\rho(y).$$

One can already notice that if f^* is invariant under T (as opposed to anti-invariant), that is if we keep the "+" in $\pm$, we get $\tilde{T}^{-1} \circ v_t \circ \tilde{T} = -\nabla F'_{\nu_t}$.

B.3.2 . Proof of Proposition 3.2.1

Proof. We first prove $\nu_0 = \mu_0$ and then prove that both $(\mu_t)_{t\geq 0}$ and $(\nu_t)_{t\geq 0}$ are Wasserstein gradient flows of the objective F defined in Equation (3.2), starting from the initial condition μ_0 at $t = 0$. The unicity of such a gradient flow then guarantees that $\mu_t = \nu_t$ and thus $f(\mu_t; T(x)) = f(\mu_t; x)$ by the preliminaries above on the prediction function (see Appendix B.3.1).

Initialization: $\nu_0 = \mu_0$. By definition, $\tilde{T}(a,b) = (a, T(b))$. Since $\mu_0 = \mu_0^1 \otimes \mu_0^2$ by assumption, and μ_0^2 is invariant under T since it has spherical symmetry, it is clear that μ_0 is invariant under $\tilde{T}$, and thus under $\tilde{T}^{-1}$ by Lemma B.1.1, which gives $\nu_0 = \mu_0$ because $\nu_t = \tilde{T}^{-1}_{\#}\mu_t$ for any t by definition. □

Time derivative. From the preliminary results above (see Appendix B.3.1) we have

$$\partial_t \nu_t = -\mathrm{div}\left(-\nabla F'_{\nu_t}\, \nu_t\right),$$

which shows that $(\nu_t)_{t\geq 0}$ is also a Wasserstein gradient flow of the objective F. By unicity of the latter (starting from the initial condition μ_0), it must hold that $\mu_t = \nu_t$ for any $t \geq 0$ which concludes the proof. é

B.3.3 . Proof of Proposition 3.2.2

The proof follows the exact same pattern as that of Proposition 3.2.1 (see Appendix B.3.2). We now have by definition, $\tilde{T}(a,b) = (-a, T(b))$ and the added symmetry assumption on μ_0^1 ensures that $\nu_0 = \mu_0$ still holds in this case. As for the time derivative, the preliminaries above (see Appendix B.3.1) ensure that

$$\nabla\left(F'_{\mu_t} \circ \tilde{T}\right)(a,b) = -\int_y \partial_2 \ell\Big(-f^*(y), -f(\nu_t; y)\Big)\nabla_{(a,b)}\phi\Big((a,b); y\Big)\mathrm{d}\rho(y)$$
$$= \int_y \partial_2 \ell\Big(f^*(y), f(\nu_t; y)\Big)\nabla_{(a,b)}\phi\Big((a,b); y\Big)\mathrm{d}\rho(y),$$

where we have used the extra assumption that $\partial_2 \ell(-y, -\hat{y}) = -\partial_2 \ell(y, \hat{y})$. This yields

$$\partial_t \nu_t = -\mathrm{div}\left(-\nabla F'_{\nu_t} \nu_t\right)$$

and the conclusion follows from the same logic as for Proposition 3.2.1.

B.4 . Proof of the exponential convergence for linear networks: Theorem 3.3.2

Proof. The proof is divided in three steps: (i) we derive the dynamics in time of the vector $w(t) = \frac{1}{2}\int ab \,\mathrm{d}\mu_t(a,b)$, (ii) we show that the positive definite matrix $H(t)$ appearing in these dynamics has its smallest eigenvalue lower-bounded by some positive constant after some $t_0 > 0$, and (iii) we show that this implies the exponential convergence to the global minimum.

Generalities on the objective Q. Expanding the square in the definition of Q (3.5), we have

$$\begin{aligned} Q(w) &= \frac{1}{2}\Big[\mathbb{E}_{x\sim\mathbb{P}}[f^*(x)^2] - 2\beta^\top w + w^\top C w\Big], \\ C &:= \mathbb{E}_{x\sim\mathbb{P}}[xx^\top] \in \mathbb{R}^{d\times d}, \\ \beta &:= \mathbb{E}_{x\sim\mathbb{P}}[f^*(x)x] \in \mathbb{R}^d. \end{aligned}$$

If $C \neq 0$, $Q(w) \to \infty$ as $||w|| \to \infty$ and since Q is lower-bounded by 0, it thus admits at least one global minimum. This minimizer $w^\star$ is unique as soon as Q is strongly convex, *i.e.*, C is definite positive, which holds in this case as we have assumed the smallest eigenvalue $\lambda_{\min}$ of C to be > 0. Note that $\nabla Q(w) = Cw - \beta = \int_x \left((x^\top w) - f^*(x)\right) x \mathrm{d}\mathbb{P}(x) \in \mathbb{R}^d$.

First step: dynamics of $w(t)$. Let $k \in \{1, \dots, d\}$, the k-th coordinate $w_k(t)$ of $w(t)$ is given by $w_k(t) = \int ab_k \,\mathrm{d}\mu_t(a,b)$, and its time derivative is given by

$$w_k'(t) = \frac{1}{2}\int \Big(\nabla_{(a,b)}(ab_k)\Big)^\top v_t(a,b) \mathrm{d}\mu_t(a,b)$$

where v_t is given by Equation (3.3) except we replace σ by $\frac{1}{2}\mathrm{id}_{\mathbb{R}^d}$ and ρ by $\mathbb{P}$ in F_{μ_t}, that is

$$v_t(a,b) = \frac{1}{2}\int_y R_t(y) \begin{pmatrix} b^\top y \\ ay \end{pmatrix} \mathrm{d}\mathbb{P}(y) \in \mathbb{R}^{1+d}.$$

On the other hand, $\nabla_{(a,b)}(ab_k) = \begin{pmatrix} b_k \\ ae_k \end{pmatrix} \in \mathbb{R}^{1+d}$ where e_k is the k-th element of the canonical orthonormal basis of $\mathbb{R}^d$. Note that here, $R_t(y) = f^*(y) -$

$\langle w(t), y\rangle$. We thus get

$$w_k'(t) = \frac{1}{4}\left\langle \int_{a,b} b_k b \mathrm{d}\mu_t(a,b), \int_y \left(f^*(y) - (w(t)^\top y)\right) y \mathrm{d}\mathbb{P}(y)\right\rangle + \frac{1}{4}\left\langle \int_{a,b} a^2 e_k \mathrm{d}\mu_t(a,b), \int_y \left(f^*(y) - (w(t)^\top y)\right) y \mathrm{d}\mathbb{P}(y)\right\rangle.$$

Note that the term on the right in the inner products is in fact equal to $-\nabla Q(w(t))$, which yields the following dynamics for the vector $w(t)$:

$$w'(t) = -H(t)\nabla Q(w(t)),$$
$$H(t) := \frac{1}{4}\left(\int bb^\top \mathrm{d}\mu_t(a,b) + \int a^2 \mathrm{d}\mu_t(a,b) I_d\right) \in \mathbb{R}^{d\times d}.$$

Second step: lower bound on the smallest eigenvalue of $H(t)$. At initialization, by symmetry one has $w(0) = 0$, and using Lemma B.1.3, one has that $H(0) = \frac{1}{4}\left(\frac{1}{d} + 1\right) I_d$, so that

$$\begin{aligned}\frac{d}{dt} Q(w(t))\Big|_{t=0} &= \left\langle w'(0), \nabla Q(w(0))\right\rangle \\ &= -\frac{d+1}{4d}||\nabla Q(0)||^2 \\ &= -\frac{d+1}{4d}||\beta||^2\end{aligned}$$

If $\beta = 0$, then $\nabla Q(0) = 0$ and since $w(0) = 0$, $w(t)$ starts at the global optimum and thus stays constant equal to 0. Otherwise, if $||\beta|| > 0$, one has $\frac{d}{dt} Q(w(t))\Big|_{t=0} < 0$, which ensures that there is a $t_0 > 0$ such that $Q(w(t)) < Q(w(0)) = Q(0)$ for any $t \in (0, t_0]$. Call $\varepsilon := [Q(0) - Q(w(t_0))]/2 > 0$. The continuity of Q at 0 guarantees that there is a $\delta > 0$ such that for any $w \in \mathbb{R}^d$, if $||w|| < \delta$, then $|Q(w) - Q(0)| \leq \varepsilon$.

Now assume that there exists $t_1 \geq t_0$ such that $\int a^2 \mathrm{d}\mu_{t_1}(a,b) \leq \delta$. Then, one has

$$\begin{aligned}||w(t_1)|| &= \left|\left|\frac{1}{2}\int ab \mathrm{d}\mu_{t_1}(a,b)\right|\right| \\ &\leq \frac{1}{2}\int |a|\,||b|| \mathrm{d}\mu_{t_1}(a,b) \\ &\leq \frac{1}{2}\int a^2 \mathrm{d}\mu_{t_1}(a,b) \\ &\leq \frac{\delta}{2} < \delta,\end{aligned}$$

where we have used in the penultimate inequality that μ_{t_1} is supported on the set $\{|a| = ||b||\}$ because of the assumptions on the initialization μ_0 (see Section 3.1.1). This ensures that $|Q(w(t_1)) - Q(0)| \leq \varepsilon$. Since : $t \mapsto Q(w(t))$ is

decreasing ($Q(w(t)) = F(\mu_t)$ and it is classical that the objective is decreasing along the gradient flow path, see third step below) and $t_1 \geq t_0$, this means that

$$0 < Q(0) - Q(w(t_0)) \leq Q(0) - Q(w(t_1)) \leq \varepsilon = [Q(0) - Q(w(t_0))]/2$$

which is a contradiction. Therefore, for any $t \geq t_0$, $\int a^2 \mathrm{d}\mu_t(a,b) \geq \delta$. Calling $\eta := \delta/4 > 0$, we thus have that for any $t \geq t_0$, the smallest eigenvalue of $H(t)$ is larger than η because $H(t)$ is the sum of the positive semi-definite matrix $\frac{1}{4}\int bb^\top \mathrm{d}\mu_t(a,b)$ and of the positive definite matrix $\frac{1}{4}\int a^2 \mathrm{d}\mu_t(a,b) I_d$ whose smallest eigenvalue is at least η for $t \geq t_0$.

Third step: exponential convergence. We have:

$$\begin{aligned}\frac{d}{dt}Q(w(t)) &= \langle w'(t), \nabla Q(w(t)) \\ &= -\nabla Q(w(t))^\top H(t) \nabla Q(w(t)) \leq 0,\end{aligned}$$

which shows that because $H(t)$ is positive definite, the objective Q is decreasing along the path $(w(t))_{t\geq 0}$. Since after $t_0 > 0$, the smallest eigenvalue of $H(t)$ is lower bounded by a constant $\eta > 0$, we have that, for any $t \geq t_0$:

$$\frac{d}{dt}Q(w(t)) \leq -\eta ||\nabla Q(w(t))||^2. \tag{B.1}$$

Because Q is λ_{min}-strongly convex (as the smallest eigenvalue of C is $\lambda_{\text{min}} > 0$), one has the classical inequality

$$\frac{1}{2}||\nabla Q(w)||^2 \geq \lambda_{\text{min}}\Big(Q(w) - Q(w^\star)\Big).$$

Plugging this into Equation (B.1) gives

$$\frac{d}{dt}\Big(Q(w(t)) - Q(w^\star)\Big) \leq -2\eta\lambda_{\text{min}}\Big(Q(w(t)) - Q(w^\star)\Big),$$

which by Gronwall's lemma in turn yields for any $t \geq t_0$

$$0 \leq Q(w(t)) - Q(w^\star) \leq e^{-2\eta\lambda_{\text{min}}(t-t_0)}\Big(Q(w(t_0)) - Q(w^\star)\Big),$$

thereby proving exponential convergence.

Exponential convergence in distance. Given that $\nabla Q(w^\star) = 0$ because $w^\star$ is and optimum, it holds $Cw^\star = \beta$. Using this fact, it easily follows that

$$Q(w) - Q(w^\star) = \frac{1}{2}\langle C(w - w^\star), w - w^\star\rangle,$$

and the right-hand-side is lower bounded by $\frac{\lambda_{\min}}{2}||w - w^\star||^2$, from which we conclude that

$$||w(t) - w^\star||^2 \leq \frac{2}{\lambda_{\min}}\Big(Q(w(t)) - Q(w^\star)\Big),$$

and the exponential decrease of the right-hand-side allows to conclude.

□

B.5 . Proofs of Section 3.4: f^* depends only on the projection on a sub-space H

B.5.1 . The general case

Closed dynamics on the sphere $\mathbb{S}^{d-1}$

We wish to show here that the pair of measures (ν_t^+, ν_t^-) defined through Equation (3.6) satisfy Equation (3.7) and that the corresponding dynamic is closed is the sense that it can be expressed solely using (ν_t^+, ν_t^-) (without requiring to express quantities in function of μ_t). Below, we use $\kappa(z) = \max(0, z)$. We do this do differentiate it from the activation function σ (which is also equal to ReLU) so as avoid confusion because the κ which appears below has nothing to do with the activation function of the network and simply comes from the integration domain in the calculations.

Equations of the dynamics on the sphere. Let $\varphi \in \mathcal{C}_c^1(\mathbb{S}^{d-1})$. One has

$$\begin{aligned}
\partial_t \int \varphi \mathrm{d}\nu_t^\pm &= \partial_t \int_{\pm a \geq 0, b} |a|||b||\varphi\left(\frac{b}{||b||}\right) \mathrm{d}\mu_t(a,b) \\
&= \partial \int_{a,b} \kappa(\pm a)||b||\varphi\left(\frac{b}{||b||}\right) \mathrm{d}\mu_t(a,b) \\
&= \int_{a,b} \nabla_{(a,b)} \left(\kappa(\pm a)||b||\varphi\left(\frac{b}{||b||}\right)\right)^\top v_t(a,b) \mathrm{d}\mu_t(a,b)
\end{aligned}$$

Let us compute the components of the gradient above. We have

$$\nabla_a \left(\kappa(\pm a)||b||\varphi\left(\frac{b}{||b||}\right)\right) = \pm\kappa'(\pm a)||b||\varphi\left(\frac{b}{||b||}\right) = \mathbf{1}_{\{\pm a \geq 0\}}||b||\varphi\left(\frac{b}{||b||}\right).$$

The Jacobian of the map : $b \in \mathbb{R}^d \mapsto b/||b||$ is equal to $\frac{1}{||b||}(I_d - bb^\top/||b||^2)$ which is a symmetric (or self-adjoint) matrix, so that the gradient *w.r.t.* b is

$$\nabla_b \left(\kappa(\pm a)||b||\varphi\left(\frac{b}{||b||}\right)\right) = \mathbf{1}_{\{\pm a \geq 0\}}|a|\left[\varphi\left(\frac{b}{||b||}\right)\frac{b}{||b||} + \left(I_d - \frac{b}{||b||}\left(\frac{b}{||b||}\right)^\top\right)\nabla\varphi\left(\frac{b}{||b||}\right)\right].$$

On the other hand, the first component of $v_t(a,b)$ (corresponding to the gradient *w.r.t.* a) is

$$v_t^1(a,b) = \int_y R_t(y)\kappa(b^\top y)\mathrm{d}\rho(y) = ||b|| \int_y R_t(y)\kappa\left(\left(\frac{b}{||b||}\right)^\top y\right)\mathrm{d}\rho(y),$$

and the last d components (corresponding to the gradient *w.r.t.* b) are

$$v_t^2(a,b) = \int_y R_t(y)a\kappa'(b^\top y)y\mathrm{d}\rho(y) = a\int_y R_t(y)\kappa'\left(\left(\frac{b}{||b||}\right)^\top y\right)y\mathrm{d}\rho(y).$$

When computing the inner product $\nabla_{(a,b)}\left(\kappa(\pm a)||b||\varphi\left(\frac{b}{||b||}\right)\right)^\top v_t(a,b)$, we can re-arrange the terms to keep one term where φ appears and the other where $\nabla\varphi$ appears. Using the facts that the Jacobian computed above is symmetric, that $\kappa(z) = \kappa'(z)z$ for any $z \in \mathbb{R}$, and that $\mathbf{1}_{\{\pm a\geq 0\}}a = \pm\mathbf{1}_{\{\pm a\geq 0\}}|a| = \pm\kappa(\pm a)$, we get,

$$\begin{aligned}\nabla_{(a,b)}\left(\kappa(\pm a)||b||\varphi\left(\frac{b}{||b||}\right)\right)^\top v_t(a,b) = \; & \pm\mathbf{1}_{\{\pm a\geq 0\}}||b||\,||b||\varphi\left(\frac{b}{||b||}\right)g_t\left(\frac{b}{||b||}\right) + \\ & \pm\mathbf{1}_{\{\pm a\geq 0\}}|a|\,|a|\varphi\left(\frac{b}{||b||}\right)g_t\left(\frac{b}{||b||}\right) + \\ & \pm\mathbf{1}_{\{\pm a\geq 0\}}|a|\,|a|\nabla\varphi\left(\frac{b}{||b||}\right)^\top\tilde{v}_t\left(\frac{b}{||b||}\right),\end{aligned}$$

where, for $u \in \mathbb{S}^{d-1}$

$$\begin{aligned}g_t(u) &:= \int_y R_t(y)\sigma(u^\top y)\mathrm{d}\rho(y), \\ \tilde{v}_t(u) &:= \int_y R_t(y)\sigma'(u^\top y)\left[y - (u^\top y)u\right]\mathrm{d}\rho(y).\end{aligned}$$

Finally, because μ_t stays on the cone $\{(a,b) \in \mathbb{R}^{d+1}; |a| = ||b||\}$ for any t (see Chizat and Bach (Chizat and Bach, 2020, Lemma 26), Wojtowytsch (Wojtowytsch, 2020, Section 2.5)), when integrating against μ_t, we can replace $||b||$ by $|a|$ and vice-versa. We thus get that the time derivative we initially computed is the sum of two terms:

$$\begin{aligned}\partial_t\int\varphi\mathrm{d}\nu_t^\pm &= 2\int_{\pm a\geq 0,b}|a|||b||\varphi\left(\frac{b}{||b||}\right)g_t\left(\frac{b}{||b||}\right)\mathrm{d}\mu_t(a,b) + \\ &\qquad \int_{\pm a\geq 0,b}|a|||b||\nabla\varphi\left(\frac{b}{||b||}\right)^\top\tilde{v}_t\left(\frac{b}{||b||}\right)\mathrm{d}\mu_t(a,b) \\ &= 2\int_{u\in\mathbb{S}^{d-1}}\varphi(u)\,g_t(u)\,\mathrm{d}\nu_t^\pm(u) + \\ &\qquad \int_{u\in\mathbb{S}^{d-1}}\nabla\varphi(u)^\top\tilde{v}_t(u)\,\mathrm{d}\nu_t^\pm(u),\end{aligned}$$

which shows that $\nu_t^\pm$ satisfies Equation (3.7) in the sense of distributions.

Closed dynamics. We want to show that g_t and $\tilde{v}_t$ can be expressed using only ν_t^+ and ν_t^-. Both these quantities depend on t only through the residual R_t, which itself only depends on t through $f(\mu_t;\cdot)$. We thus show that the latter can be expressed using only ν_t^+ and ν_t^-, which easily follows from writing, for any $y \in \mathbb{R}^d$,

$$\begin{aligned}
f(\mu_t; y) &= \int a\sigma\left(b^\top y\right) \mathrm{d}\mu_t(a,b) \\
&= \int a||b||\sigma\left(\left\langle \frac{b}{||b||}, y \right\rangle\right) \mathrm{d}\mu_t(a,b) \\
&= \int_{a\geq 0,b} |a|\,||b||\sigma\left(\left\langle \frac{b}{||b||}, y \right\rangle\right) \mathrm{d}\mu_t(a,b) - \int_{a\leq 0,b} |a|\,||b||\sigma\left(\left\langle \frac{b}{||b||}, y \right\rangle\right) \mathrm{d}\mu_t(a,b) \\
&= \int_{u\in\mathbb{S}^{d-1}} \sigma\left(u^\top y\right) \mathrm{d}\nu_t^+(u) - \int_{u\in\mathbb{S}^{d-1}} \sigma\left(u^\top y\right) \mathrm{d}\nu_t^-(u)
\end{aligned}$$

Closed dynamics in $d_H + 1$ dimensions

Proof. We first prove that the Equation (3.9) for $\tau_t^\pm$ holds in the sense of distributions, and then show that the corresponding dynamics are **closed** because the V_t and g_t appearing in Equation (3.9) can be expressed with (τ_t^+, τ_t^-) (and not only with (ν_t^+, ν_t^-) for instance). We show this by expressing $f(\mu_t;\cdot)$ only in function of the pair (τ_t^+, τ_t^-).

The pair (τ_t^+, τ_t^-) satisfy Equation (3.9). First, we show that g_t and $\tilde{v}_t$ defined in Equation (3.8) admit modified expressions that match the structure of the pushforward transforming $\nu_t^\pm$ into $\tau_t^\pm$. Indeed, since ρ is assumed to be spherically symmetric, it is invariant by any orthogonal transformation. In particular, for a fixed $u \in \mathbb{S}^{d-1}$ such that $u^\perp \neq 0$, we consider the orthogonal map $T^u : \mathbb{R}^d \to \mathbb{R}^d$ such that $T^u_{|H} = \mathrm{id}_H$ and $T^u_{|H^\perp}$ sends the canonical orthonormal basis $(e_1^\perp, \dots, e_{d_\perp}^\perp)$ of $H^\perp$ on $(u^\perp/||u^\perp||, u_2, \dots, u_{d_\perp})$ where $(u_2, \dots, u_{d_\perp}) \in (H^\perp)^{d_\perp - 1}$ is an orthonormal family, orthogonal to $u^\perp$, so that for any $y^\perp \in H^\perp$ with coordinates $y_1^\perp, \dots, y_{d_\perp}^\perp$ in the basis $(e_1^\perp, \dots, e_{d_\perp}^\perp)$, $T^u_{|H^\perp}(y^\perp) = y_1^\perp u^\perp/||u^\perp|| + h_u(y_\perp)$ with $h_u(y_\perp) \perp u^\perp$.

Note that since $f^*(y) = f_H(y^H)$ and $f(\mu_t; y) = \tilde{f}_t(y^H, ||y^\perp||)$, the residual $R_t(y) = f^*(y) - f(\mu_t; y)$ is invariant by any orthogonal transformation which preserves H (and in particular by T^u). We thus have

$$\begin{aligned}
g_t(u) &= \int_y R_t(y)\sigma\left(\left\langle u^H, y^H \right\rangle + y_1^\perp ||u^\perp||\right) \mathrm{d}\rho =: \tilde{g}_t(u^H, ||u^\perp||), \\
\tilde{v}_t(u) &= \int_y R_t(y)\sigma'\left(\left\langle u^H, y^H \right\rangle + y_1^\perp ||u^\perp||\right) \left[y^H + T^u_{|H^\perp}(y^\perp) - \left(\left\langle u^H, y^H \right\rangle + y_1^\perp ||u^\perp||\right) u\right] \mathrm{d}\rho.
\end{aligned}$$

Now consider, for any $(\theta, z^H) \in [0, \pi/2] \times \mathbb{S}^{d_H-1}$,

$$G_t(\theta, z^H) := \tilde{g}_t(\cos(\theta)z^H, \sin(\theta))$$
$$V_t(\theta, z^H) := \int_y R_t(y)\sigma'\left(\cos(\theta)\left\langle z^H, y^H\right\rangle + y_1^\perp \sin(\theta)\right)\begin{pmatrix} y_1^\perp \cos(\theta) - \sin(\theta)\left\langle z^H, y^H\right\rangle \\ \frac{y^H}{\cos(\theta)} - \left\langle z^H, y^H\right\rangle \frac{z^H}{\cos(\theta)}\end{pmatrix} \mathrm{d}\rho$$

We show below that (τ_t^+, τ_t^-) satisfy Equation (3.9) with the G_t and V_t defined above. Let $\varphi \in \mathcal{C}_c^1([0, \pi/2] \times \mathbb{S}^{d_H-1})$. Since $\tau_t^\pm$ is defined as a push-forward measure obtained from $\nu_t^\pm$ we have:

$$\begin{aligned}\partial_t \int \varphi(\theta, z^H)\mathrm{d}\tau_t^\pm(\theta, z^H) &= \partial_t \int \varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\mathrm{d}\nu_t^\pm(u) \\ &= \pm 2\int \varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\tilde{g}_t(u^H, ||u^\perp||)\mathrm{d}\nu_t^\pm(u) + \\ &\quad \pm \int \nabla_u\left(\varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\right)^\top \tilde{v}_t(u)\mathrm{d}\nu_t^\pm(u).\end{aligned}$$

By definition of the pushforward, and since $u^H = \cos(\arccos(||u^H||))u^H/||u^H||$ and $||u^\perp|| = \sin(\arccos(||u^H||))$ for $u \in \mathbb{S}^{d-1}$, the first integral is equal to the following integral: $\int \varphi(\theta, z^H)G_t(\theta, z^H)\mathrm{d}\tau_t^\pm(u)$. For the second integral, let us first compute the gradient. One has

$$\begin{aligned}\nabla_u\left(\varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\right) =& \partial_\theta\varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\frac{-1}{\sqrt{1-||u^H||^2}}\frac{u^H}{||u^H||} + \\ & \frac{1}{||u^H||}\left[I_{d_H} - \frac{u^H(u^H)^\top}{||u^H||^2}\right](\nabla_{z^H}\varphi)\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right).\end{aligned}$$

We observe that the gradient above belongs to H which implies that when computing its inner product with $\tilde{v}_t(u)$ we can consider only the component of the latter along H. Additionally, we note that $I_{d_H} - u^H(u^H)^\top/||u^H||^2$ is actually the orthogonal projection onto $\{u^H\}^\perp$, so that it yields 0 when applied to u. Using that $||u^\perp|| = \sqrt{1-||u^H||^2}$ for $u \in \mathbb{S}^{d-1}$, we then get:

$$\nabla_u\left(\varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\right)^\top \tilde{v}_t(u) = \nabla\varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)^\top V_t\left((\arccos(||u^H||), \frac{u^H}{||u^H||}\right).$$

where $\nabla\varphi(\theta, z^H) = \begin{pmatrix}\partial_\theta\varphi(\theta, z^H) \\ \nabla_{z^H}\varphi(\theta, z^H)\end{pmatrix}$. This shows that

$$\begin{aligned}\partial_t \int \varphi(\theta, z^H)\mathrm{d}\tau_t^\pm(\theta, z^H) = &\pm 2\int \varphi(\theta, z^H)G_t(\theta, z^H)\mathrm{d}\tau_t^\pm(\theta, z^H) + \\ &\pm \int \nabla\varphi(\theta, z^H)^\top V_t(\theta, z^H)\mathrm{d}\tau_t^\pm(\theta, z^H),\end{aligned}$$

which proves that $\tau_t^\pm$ indeed satisfies Equation (3.8) in the sense of distributions.

The dynamics are closed in the pair (τ_t^+, τ_t^-). The only thing left to prove to show that the dynamics are closed for the pair (τ_t^+, τ_t^-) is that G_t and V_t can be expressed using only the pair (τ_t^+, τ_t^-). The only dependence of these quantities on t is through the residual R_t which itself depends on t only through $f(\mu_t; \cdot)$. Let $y \in \mathbb{R}^d$. We have already shown at the end of the previous Section B.5.1 that by definition of ν_t^+ and ν_t^-, we have

$$f(\mu_t; y) = \int_{u \in \mathbb{S}^{d-1}} \sigma\left(u^\top y\right) \mathrm{d}\left(\nu_t^+ - \nu_t^-\right)(u).$$

On the other hand, we show below that the integral of any measurable function $\varphi : \mathbb{S}^{d-1} \to \mathbb{R}$ against $\nu_t^\pm$ can be expressed as an integral against $\tau_t^\pm$ in the case where $\nu_t^\pm$ admits a density *w.r.t.* the uniform measure on $\mathbb{S}^{d-1}$ (which is the case for $\nu_0^\pm$), the case of a general measure $\nu_t^\pm$ being a simple extension via a weak convergence argument. Thus call $p_t^\pm$ the density of $\nu_t^\pm$ *w.r.t.* $\tilde{\omega}_d$. Since $\nu_t^\pm$ is invariant by any linear map T such that $T_{|H} = \mathrm{id}_H \; T_{|H^\perp} \in \mathcal{O}(d_\perp)$ (because of the symmetries on μ_t given by Proposition 3.2.1), and since this is also the case for $\tilde{\omega}_d$ because $\tilde{\omega}_d$ has spherical symmetry and T is orthogonal, we have by Lemma B.1.2 that $p_t^\pm$ is invariant by any such T, which then leads to p_t having the form $p_t(u) = \tilde{p}_t^\pm(u^H, ||u^\perp||)$ by Lemma 3.4.1.

First step. We show that $\tau_t^\pm$ has the following density

$$q_t^\pm(\theta, z^H) = |\mathbb{S}^{d_\perp - 1}| \tilde{p}_t^\pm(\cos(\theta) z^H, \sin(\theta))$$

w.r.t. $\tilde{\gamma} \otimes \tilde{\omega}_{d_H}$ where the measure $\tilde{\gamma}$ is the normalized counterpart of the measure in Definition B.1.1. Indeed, let $\varphi : [0, \pi/2] \times \mathbb{S}^{d-1} \to \mathbb{R}$ be any measurable function *w.r.t.* $\tau_t^\pm$. Using the disintegration Lemma B.1.4 on $\tilde{\omega}_d$, one has that

$$\begin{aligned}
\int \varphi(\theta, z^H) \mathrm{d}\tau_t^\pm(\theta, z^H) &= \int \varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right) \mathrm{d}\nu_t^\pm(u) \\
&= \int \varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right) \tilde{p}_t^\pm(u^H, ||u^\perp||) \mathrm{d}\tilde{\omega}_d(u) \\
&= \int \varphi\left(\theta, z^H\right) \tilde{p}_t^\pm(\cos(\theta) z^H, \sin(\theta)) \mathrm{d}\tilde{\gamma}(\theta) \mathrm{d}\tilde{\omega}_{d_H}(z^H) \mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp) \\
&= \int \varphi\left(\theta, z^H\right) \tilde{p}_t^\pm(\cos(\theta) z^H, \sin(\theta)) \mathrm{d}\tilde{\gamma}(\theta) \mathrm{d}\tilde{\omega}_{d_H}(z^H),
\end{aligned}$$

which proves the desired density for $\tau_t^\pm$.

Second step. Consider a measurable $\varphi : \mathbb{S}^{d-1} \to \mathbb{R}$ *w.r.t.* $\nu_t^{\pm}$. One has with similar calculations as above

$$\begin{aligned}\int_u \varphi(u)\mathrm{d}\nu_t^{\pm}(u) &= \int_u \varphi(u)\tilde{p}_t^{\pm}(u^H, ||u^{\perp}||)\mathrm{d}\tilde{\omega}_d(u) \\ &= \int \varphi(\cos(\theta)z^H + \sin(\theta)z^{\perp})\tilde{p}_t^{\pm}(\cos(\theta)z^H, \sin(\theta))\mathrm{d}\tilde{\gamma}(\theta)\mathrm{d}\tilde{\omega}_{d_H}(z^H)\mathrm{d}\tilde{\omega}_{d_\perp}(z^{\perp}) \\ &= \int_{\theta, z^H}\left(\int_{z^{\perp}} \varphi(\cos(\theta)z^H + \sin(\theta)z^{\perp})\mathrm{d}\tilde{\omega}_{d_\perp}(z^{\perp})\right) q_t^{\pm}(\theta, z^H)\mathrm{d}\tilde{\gamma}(\theta)\mathrm{d}\tilde{\omega}_{d_H}(z^H) \\ &= \int_{\theta, z^H}\left(\int_{z^{\perp}} \varphi(\cos(\theta)z^H + \sin(\theta)z^{\perp})\mathrm{d}\tilde{\omega}_{d_\perp}(z^{\perp})\right) \mathrm{d}\tau_t^{\pm}(\theta, z^H).\end{aligned}$$

Applying this to $f(\mu_t; y)$ shows that the latter quantity can be expressed solely using (τ_t^+, τ_t^-), which proves that the dynamics is indeed closed and therefore concludes the proof when $\nu_t^{\pm}$ has a density.

Third step: extending to any measure. It is known that for any measure ν over $\mathbb{S}^{d-1}$, there exists a sequence of measure $(\nu_n)_{n\in\mathbb{N}}$ such that: (i) ν_n has a density p_n *w.r.t.* the uniform measure $\tilde{\omega}_d$ over $\mathbb{S}^{d-1}$, and (ii) the sequence $(\nu_n)_{n\in\mathbb{N}}$ converges weakly to ν, that is, for any continuous (and thus automatically bounded because the unit sphere is compact) φ, $\int \varphi \mathrm{d}\nu_n \xrightarrow[n\to\infty]{} \int \varphi \mathrm{d}\nu$. Let thus $\nu \in \mathcal{M}_+(\mathbb{S}^{d-1})$, and consider a sequence $(\nu_n)_{n\in\mathbb{N}}$ with density converging weakly towards ν. Let τ (*resp.* τ_n) be defined from ν (*resp.* ν_n) as $\tau_t^{\pm}$ is defined from $\nu_t^{\pm}$, that is for any measurable $\varphi : [0, \pi/2] \times \mathbb{S}^{d_H-1} \to \mathbb{R}$,

$$\begin{aligned}\int \varphi(\theta, z^H)\mathrm{d}\tau(\theta, z^H) &= \int \varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\mathrm{d}\nu(u), \\ \int \varphi(\theta, z^H)\mathrm{d}\tau_n(\theta, z^H) &= \int \varphi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right)\mathrm{d}\nu_n(u).\end{aligned}$$

Let thus φ be a continuous map from $\mathbb{S}^{d-1} \to \mathbb{R}$ (having in mind the example of : $u \mapsto \sigma(u^\top y)$ for a fixed y). By the result of Step 2, since ν_n has a density for every n, we have that

$$\int \varphi(u)\mathrm{d}\nu_n(u) = \int_{\theta, z^H}\left(\int_{z^{\perp}} \varphi(\cos(\theta)z^H + \sin(\theta)z^{\perp})\mathrm{d}\tilde{\omega}_{d_\perp}(z^{\perp})\right)\mathrm{d}\tau_n(\theta, z^H), \tag{B.2}$$

and taking the limit $n \to \infty$, the left-hand-side of Equation (B.2) converges to $\int \varphi \mathrm{d}\nu$ by assumption. Now let us look at the right-hand-side of (B.2). Calling $\psi(\theta, z^H) = \int_{z^{\perp}} \varphi(\cos(\theta)z^H + \sin(\theta)z^{\perp})\mathrm{d}\tilde{\omega}_{d_\perp}(z^{\perp})$ and $\Phi(u) = \int_{z^{\perp}} \varphi(u^H + ||u^{\perp}||z^{\perp})\mathrm{d}\tilde{\omega}_{d_\perp}(z^{\perp})$, the right-hand-side is in fact $\int \psi \mathrm{d}\tau_n$ and, for any $n \in \mathbb{N}$, is

equal to:

$$\begin{aligned}\int \psi \mathrm{d}\tau_n &= \int_u \psi\left(\arccos(||u^H||), \frac{u^H}{||u^H||}\right) \mathrm{d}\nu_n(u) \\ &= \int_u \int_{z^\perp} \varphi\left(||u^H|| \frac{u^H}{||u^H||} + ||u^\perp|| z^\perp\right) \mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp) \mathrm{d}\nu_n(u) \\ &= \int_u \int_{z^\perp} \varphi\left(u^H + ||u^\perp|| z^\perp\right) \mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp) \mathrm{d}\nu_n(u) \\ &= \int \Phi \mathrm{d}\nu_n,\end{aligned}$$

and a similar result holds for τ and ν. Now, the continuity of Φ is readily obtained from that of φ, and thus the right-hand-side in the last equality above converges to $\int \Phi \mathrm{d}\nu$ which is also equal to $\int \psi \mathrm{d}\tau$ by the same calculations as above. The right-hand-side in (B.2) therefore converges to $\int \psi \mathrm{d}\tau$, and since the limits of both sides are equal, we get $\int \varphi \mathrm{d}\nu = \int \psi \mathrm{d}\tau$, which is the claim of Step 2 for a general measure ν which does not necessarily admit a density, thereby concluding the proof. □

B.5.2 . Case when f^* is the euclidean norm: Theorem 3.4.3

Here, we give the proof of Theorem 3.4.3 which shows that when $f^*(x) = ||x^H||$ the dynamics can be reduced to a single variable: the angle $\theta \in [0, \pi/2]$ between particles and the subs-space H.

We decompose the proof in three steps: first we show that the pair of measures $(\tau_t^+, \tau_t^-) \in \mathcal{M}_+([0, \pi/2])$ as defined in Section 3.4.2 indeed follows Equation (3.10); then we show that the dynamics are indeed closed by proving that the terms V_t and G_t appearing in the GF depend only on (τ_t^+, τ_t^-); and finally, we show that Equation (3.10) indeed corresponds to a Wasserstein-Fisher-Rao GF on a given objective functional over $\mathcal{M}_+([0, \pi/2])^2$.

Proof of the GF equation

Proof. We first use the added symmetry to simplify the terms g_t and $\tilde{v}_t$ which appear in the GF with (ν_t^+, ν_t^-) (see Section 3.4.1) and express them only with $||u^H||$ and $||u^\perp||$. Then we use the equations satisfied by (ν_t^+, ν_t^-) to obtain equations for (τ_t^+, τ_t^-).

Equations for (τ_t^+, τ_t^-). Let $\varphi \in \mathcal{C}_c^1([0, \pi/2])$. We have

$$\begin{aligned}\partial_t \int \varphi \mathrm{d}\tau_t^\pm =& \partial_t \int \varphi\left(\arccos(||u^H||)\right) \mathrm{d}\nu_t^\pm(u) \\ =& \pm \int \nabla_u \left(\varphi\left(\arccos(||u^H||)\right)\right)^\top \tilde{v}_t(u) \mathrm{d}\nu_t^\pm(u) \\ & \pm 2 \int \varphi\left(\arccos(||u^H||)\right) g_t(u) \mathrm{d}\nu_t^\pm(u).\end{aligned}$$

One has that

$$\nabla_u \left(\varphi\left(\arccos(||u^H||)\right)\right) = \varphi'\left(\arccos(||u^H||)\right) \times \frac{-1}{\sqrt{1-||u^H||^2}} \frac{u^H}{||u^H||},$$

which belongs to H. We recall here the expressions of $\tilde{v}_t$ and g_t: for any $u \in \mathbb{S}^{d-1}$, we have

$$\begin{aligned} g_t(u) &= \int_y R_t(y)\sigma(u^\top y)\mathrm{d}\rho(y), \\ \tilde{v}_t(u) &= \int_y R_t(y)\sigma'(u^\top y)[y - (u^\top y)u]\mathrm{d}\rho(y). \end{aligned}$$

Since, $f^*(x) = ||x^H||$, f^* is now invariant under any orthogonal map T preserving H and $H^\perp$, that is such that the restrictions $T_{|H} \in \mathcal{O}(d_H)$ and $T_{|H^\perp} \in \mathcal{O}(d_\perp)$. Proposition 3.2.1 then ensures that so is $f(\mu_t, \cdot)$, which in turn implies that the residual $R_t(\cdot) = \partial_2 \ell(f(\mu_t; \cdot), f^*(\cdot))$ also shares that invariance property. Using a similar change of variable as in Appendix B.5.1, and because ρ is spherically symmetric, one gets that g_t can be re-written

$$g_t(u) = \int_y R_t(y)\sigma\left(y_1^H||u^H|| + y_1^\perp||u^\perp||\right)\mathrm{d}\rho(y).$$

Calling

$$G_t(\theta) := \int_y R_t(y)\sigma\left(y_1^H\cos(\theta) + y_1^\perp\sin(\theta)\right)\mathrm{d}\rho(y),$$

one has $g_t(u) = G_t(\arccos(||u^H||))$ because $u \in \mathbb{S}^{d-1}$, so that $||u^\perp|| = \sqrt{1-||u^H||^2}$. Then, by definition of $\tau_t^\pm$, the second integral in the time derivative above is equal to $\int \varphi(\theta)G_t(\theta)\mathrm{d}\tau_t^\pm$. For the first integral appearing in that time derivative, we get

$$\nabla_u \left(\varphi\left(\arccos(||u^H||)\right)\right)^\top \tilde{v}_t(u) = \frac{\varphi'\left(\arccos(||u^H||)\right)}{||u^\perp||\,||u^H||} \int_y R_t(y)\sigma'(u^\top y)[(u^\top y)u - y]^\top u^H \mathrm{d}\rho.$$

Expanding the inner product inside the integral, we have

$$\begin{aligned} [(u^\top y)u - y]^\top u^H &= (\langle u^H, y^H\rangle + \langle u^\perp, y^\perp\rangle)||u^H||^2 - \langle u^H, y^H\rangle \\ &= ||u^H||^2\langle u^\perp, y^\perp\rangle - (1-||u^H||^2)\langle u^H, y^H\rangle \\ &= ||u^H||^2\langle u^\perp, y^\perp\rangle - ||u^\perp||^2\langle u^H, y^H\rangle. \end{aligned}$$

Calling

$$V_t(\theta) := \int_y R_t(y)\sigma'\left(y_1^H\cos(\theta) + y_1^\perp\sin(\theta)\right)[y_1^\perp\cos(\theta) - y_1^H\sin(\theta)]\mathrm{d}\rho(y) = G'(\theta),$$

and using again the spherical symmetry of ρ, with the same change of variable in the integral as for g_t, we get that

$$\nabla_u \left(\varphi\left(\arccos(||u^H||)\right)\right)^\top \tilde{v}_t(u) = \varphi'\left(\arccos(||u^H||)\right) V_t(\arccos(||u^H||)).$$

Finally, this combined with the previous result on the integral with g_t yields

$$\partial_t \int \varphi \mathrm{d}\tau_t^\pm = \pm \int \varphi'(\theta) V_t(\theta) \mathrm{d}\tau_t^\pm(\theta) \pm 2 \int \varphi(\theta) G_t(\theta) \mathrm{d}\tau_t^\pm(\theta),$$

which leads to the desired equation

$$\partial \tau_t^\pm = -\mathsf{div}\left(\pm V_t \tau_t^\pm\right) \pm 2 G_t \tau_t^\pm.$$

□

Proof that the dynamics on the angle θ are closed

The proof follow closely that of Appendix B.5.1 (where we prove closed dynamics), except here we take advantage of the added symmetry of the dynamics. As in Appendix B.5.1, we have

$$f(\mu_t; y) = \int_{u \in \mathbb{S}^{d-1}} \sigma\left(u^\top y\right) \mathrm{d}\left(\nu_t^+ - \nu_t^-\right)(u),$$

and the only thing to prove is that this quantity can be expressed using only (τ_t^+, τ_t^-). As in Appendix B.5.1, we first prove this when $\nu_t^\pm$ has a density, which is the case for $\nu_0^\pm$ and should thus remain so during the dynamics.

Similarly to what occurs in Appendix B.5.1, $\nu_t^\pm$ is invariant by any orthogonal map T which preserves H and $H^\perp$ because μ_t has those symmetries given by Proposition 3.2.1, and if $\nu_t^\pm$ has a density $p_{\nu_t^\pm}$ *w.r.t.* $\tilde{\omega}_d$, then $p_t^\pm$ is also invariant by any such map T, and thus depends only on the norms $||u^H||$ and $||u^\perp||$ of its input $u \in \mathbb{S}^{d-1}$. But since its input is on the sphere, those norms are determined by the angle $\theta = \arccos(||u^H||)$ between the input u and H. Calling $q_t^\pm$ such that $p_t^\pm(u) = q_t^\pm(\arccos(||u^H||))$, this will lead $\tau_t^\pm$ to have the density $q_t^\pm$ *w.r.t.* $\tilde{\gamma}$. Then, we show below that similarly to Appendix B.5.1, the integral of any measurable $\varphi : \mathbb{S}^{d-1} \to \mathbb{R}$ against $\nu_t^\pm$ can be expressed as an integral against $\tau_t^\pm$. Indeed, using the disintegration Lemma B.1.4,

$$\begin{aligned}
\int \varphi d\nu_t^\pm &= \int_{\theta \in [0, \pi/2]} \varphi(u) q_t^\pm(\arccos(||u^H||)) \mathrm{d}\tilde{\omega}_d(u) \\
&= \int_u \varphi\left(\cos(\theta) z^H + \sin(\theta) z^\perp\right) q_t^\pm(\theta) \mathrm{d}\tilde{\omega}_{d_H}(z^H) \mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp) \mathrm{d}\tilde{\gamma}(\theta) \\
&= \int_{\theta \in [0, \pi/2]} \tilde{\varphi}(\theta) q_t^\pm(\theta) \mathrm{d}\tilde{\gamma}(\theta) \\
&= \int_{\theta \in [0, \pi/2]} \tilde{\varphi}(\theta) \mathrm{d}\tau_t^\pm(\theta)
\end{aligned}$$

where

$$\tilde{\varphi}(\theta) := \int_{z^H, z^\perp} \varphi\left(\cos(\theta) z^H + \sin(\theta) z^\perp\right) \mathrm{d}\tilde{\omega}_{d_H}(z^H) \mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp),$$

which concludes the proof if $\nu_t^\pm$ has a density *w.r.t.* the uniform measure $\tilde{\omega}_d$ on the sphere $\mathbb{S}^{d-1}$. The general case is obtained by a weak convergence argument (of measures with density) as in the third step of Section B.5.1.

Proof of the Wasserstein-Fisher-Rao GF

Proof. Recall that γ is the measure in Definition B.1.1, and consider the following objective functional over $\mathcal{M}([0, \pi/2])^2$:

$$\begin{aligned}
A(\tau^+, \tau^-) :&= \int_{\varphi \in [0, \pi/2]} \ell\Big(\cos(\varphi),\, \tilde{f}(\tau^+, \tau^-; \varphi)\Big) \mathrm{d}\tilde{\gamma}(\varphi), \\
\tilde{f}(\tau^+, \tau^-; \varphi) :&= \int_{\theta \in [0, \pi/2]} \tilde{\phi}\,(\theta; \varphi)\, \mathrm{d}(\tau^+ - \tau^-)(\theta), \\
\tilde{\phi}\,(\theta; \varphi) :&= \int_{r,s \in [-1,1]} \sigma\Big(r \cos(\varphi)\cos(\theta) + s \sin(\varphi)\sin(\theta)\Big) \mathrm{d}\tilde{\gamma}_{d_H}(r) \mathrm{d}\tilde{\gamma}_{d_\perp}(s)
\end{aligned}$$

where, for any $p \in \mathbb{N}$, $\mathrm{d}\gamma_p(r) = (1 - r^2)^{(p-3)/2} \mathrm{d}r$, and $\tilde{\gamma}_p = \gamma_p / |\gamma_p|$ with the normalizing factor $|\gamma_p| = B\,(1/2, (p-1)/2) = \sqrt{\pi}\Gamma((p-1)/2)/\Gamma(p/2) = |\mathbb{S}^{p-1}|/|\mathbb{S}^{p-2}|$. Note that $\tilde{\gamma}_p$ can be simply expressed as the law of $\epsilon \times \sqrt{X}$ where $\epsilon \sim \mathcal{U}(\{-1, +1\})$ and $X \sim \mathrm{Beta}(1/2, (p-1)/2)$.

Computing the first variation or Fréchet derivative of the functional A *w.r.t.* its first and second argument yields, for any $\theta \in [0, \pi/2]$,

$$\frac{\delta A}{\delta \tau^\pm}(\tau^+, \tau^-)[\theta] = \pm \int_\varphi \partial_2 \ell\Big(\cos(\varphi),\, \tilde{f}(\tau^+, \tau^-; \varphi)\Big) \tilde{\phi}(\theta; \varphi) \mathrm{d}\tilde{\gamma}(\varphi).$$

To conclude one needs only observe that the quantity above is simply equal to $G_t(\theta)$, up to a fixed multiplicative constant. Since we have assumed ρ to be the uniform measure over $\mathbb{S}^{d-1}$ to ensure that the Wasserstein GF (3.3) is well-defined, the constant is one here but in the case of a general ρ with spherical symmetry, the result should also hold (as long as the Wasserstein GF (3.3) is well-defined) but the proof is more technical and different constants might appear.

Simplifying $f(\mu_t; \cdot)$. Using the results from Appendix B.5.2, we have for any $\varphi, z^H, z^\perp \in [0, \pi/2] \times \mathbb{S}^{d_H - 1} \times \mathbb{S}^{d_\perp - 1}$ (so that $u = \cos(\varphi) z^H + \sin(\varphi) z^\perp \in \mathbb{S}^{d-1}$)

$$\begin{aligned}
&f(\mu_t; \cos(\varphi) z^H + \sin(\varphi) z^\perp) = \\
&\qquad \int_\psi \int_{\xi^H, \xi^\perp} \sigma\left(\cos(\psi)\cos(\varphi)\langle \xi^H, z^H \rangle + \sin(\psi)\sin(\varphi)\langle \xi^\perp, z^\perp \rangle\right) \mathrm{d}\tilde{\omega}_{d_H}(\xi^H) \mathrm{d}\tilde{\omega}_{d_\perp}(\xi^\perp) \mathrm{d}(\tau_t^+ - \tau_t^-)(\psi)
\end{aligned}$$

Now, because of the integration against uniform measures on the unit spheres, and the inner products involved, we can use some spherical harmonics theory to simplify those calculations. Using The Funk-Hecke formula (see Atkinson and Han (Atkinson and Han, 2012, Theorem 2.22), $n = 0$, $d = d_H$ or $d = d_\perp$), we get

$$
\begin{aligned}
&f(\mu_t; \cos(\varphi) z^H + \sin(\varphi) z^\perp) = \\
&\quad \frac{|\mathbb{S}^{d_H-2}||\mathbb{S}^{d_\perp-2}|}{|\mathbb{S}^{d_H-1}||\mathbb{S}^{d_\perp-1}|} \int_\psi \int_{r,s} \sigma\left(r\cos(\psi)\cos(\varphi) + s\sin(\psi)\sin(\varphi)\right) \mathrm{d}\gamma_{d_H}(r)\mathrm{d}\gamma_{d_\perp}(s)\mathrm{d}(\tau_t^+ - \tau_t^-)(\psi) \\
&\quad = \frac{1}{|\gamma_{d_H}||\gamma_{d\perp}|}|\gamma_{d_H}||\gamma_{d_\perp}| \int_{\psi\in[0,\pi/2]} \tilde{\phi}(\psi;\varphi)\mathrm{d}(\tau_t^+ - \tau_t^-)(\psi) \\
&\quad = \tilde{f}(\tau_t^+, \tau_t^-; \varphi).
\end{aligned}
$$

Simplifying $f^*(\cos(\varphi)z^H + \sin(\varphi)z^\perp)$. Because $f^*(y) = ||y^H||$, $f^*(\cos(\varphi)z^H + \sin(\varphi)z^\perp)$ is simply $||\cos(\varphi)z^H|| = \cos(\varphi)$ because $z^H \in \mathbb{S}^{d_H-1}$.

With the previous expressions for $f(\mu_t; \cdot)$ and f^* we have that for any function $\Phi : \mathbb{R}^2 \to \mathbb{R}$,

$$
\Phi\Big(f^*(\cos(\varphi)z^H + \sin(\varphi)z^\perp),\ f(\mu_t; \cos(\varphi)z^H + \sin(\varphi)z^\perp)\Big) = \Phi\Big(\cos(\varphi),\ \tilde{f}(\tau_t^+, \tau_t^-; \varphi)\Big).
$$

Note that this applies both to $\Phi(y, \hat{y}) = \ell(y, \hat{y})$ and $\Phi(y, \hat{y}) = -\partial_2\ell(y, \hat{y})$.

Proof that $F(\mu_t) = A(\tau_t^+, \tau_t^-)$. Using the disintegration Lemma B.1.4 for the uniform measure on the unit sphere $\mathbb{S}^{d-1}$, we have

$$
\begin{aligned}
F(\mu_t) &= \int_y \ell\Big(f^*(y),\ f(\mu_t; y)\Big)\mathrm{d}\rho(y) \\
&= \int \ell \circ \left(f^*(\cdot),\ f(\mu_t; \cdot)\right)\Big(\cos(\varphi)z^H + \sin(\varphi)z^\perp\Big)\,\mathrm{d}\tilde{\omega}_{d_H}(z^H)\mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp)\mathrm{d}\tilde{\gamma}(\varphi) \\
&= \int \ell\Big(\cos(\varphi),\ \tilde{f}(\tau_t^+, \tau_t^-; \varphi)\Big)\mathrm{d}\tilde{\omega}_{d_H}(z^H)\mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp)\mathrm{d}\tilde{\gamma}(\varphi) \\
&= \int \ell\Big(\cos(\varphi),\ \tilde{f}(\tau_t^+, \tau_t^-; \varphi)\Big)\mathrm{d}\tilde{\gamma}(\varphi),
\end{aligned}
$$

where we have used in the last equality the fact that the integrand does not depend on z^H or $z^\perp$ and that $\tilde{\omega}_{d_H}$ and $\tilde{\omega}_{d_\perp}$ are probability measures (and thus their total mass is 1).

Simplifying G_t. Using the disintegration Lemma B.1.4, we have:

$$
\begin{aligned}
G_t(\theta) &= \int_y R_t(y)\sigma\Big(y_1^H\cos(\theta) + y_1^\perp\sin(\theta)\Big)\,\mathrm{d}\rho(y) \\
&= \int R_t(\cos(\varphi)z^H + \sin(\varphi)z^\perp)\sigma\Big(z_1^H\cos(\varphi)\cos(\theta) + z_1^\perp\sin(\varphi)\sin(\theta)\Big)\,\tilde{\omega}_{d_H}(z^H)\mathrm{d}\tilde{\omega}_{d_\perp}(z^\perp)\mathrm{d}\tilde{\gamma}(\varphi).
\end{aligned}
$$

Similarly to what we did for simplifying $f(\mu_t;\cdot)$, we can simplify the integrals against $\tilde{\omega}_{d_H}$ and $\tilde{\omega}_{d_\perp}$ using spherical harmonics theory to get:

$$G_t(\theta) = -\int_{\varphi\in[0,\pi/2]} \partial_2\ell\Big(\cos(\varphi),\ \tilde{f}(\tau_t^+,\tau_t^-;\varphi)\Big)\tilde{\phi}(\varphi;\theta)\mathrm{d}\tilde{\gamma}(\varphi).$$

This shows that

$$\begin{aligned} -\frac{\delta A}{\delta\tau^+}(\tau_t^+,\tau_t^-)[\theta] &= G_t(\theta) \\ \frac{\delta A}{\delta\tau^-}(\tau_t^+,\tau_t^-)[\theta] &= G_t(\theta), \end{aligned}$$

which proves that Equation (3.10) indeed describes the evolution of the Wasserstein-Fisher-Rao for the objective functional A over $\mathcal{M}([0,\pi/2])^2$, given by the pair (τ_t^+,τ_t^-). □

B.6 . Numerical simulations in one dimension

Measure discretization. Discretizing μ_t via $\mu_{m,t} = \frac{1}{m}\sum_{j=1}^m \delta_{(a_j(t),b_j(t))}$, we get that $\tau_{m,t} := \tau_{m,t}^+ - \tau_{m,t}^- = \frac{1}{m}\sum_{j=1}^m c_j(t)\delta_{\theta(t)}$ where

$$\begin{aligned} c_j(t) &= \varepsilon_j|a_j(t)|\,||b_j(t)||, \\ \varepsilon_j &= \mathsf{sign}(a_j(0)), \\ \theta_j(t) &= \arccos\left(\frac{b_j(t)}{||b_j(t)||}\right). \end{aligned}$$

Initializing through $a_j(0)\sim\mathcal{U}\{-1,+1\}$ and $b_j(0)\sim\tilde{\omega}_d=\mathcal{U}(\mathbb{S}^{d-1})$, yields $c_j(0)\sim\mathcal{U}\{-1,+1\}$ and $\theta_j(0)\sim\tilde{\gamma}$, i.i.d. over j. The gradient flows of Equation (3.10) translates into the following ODEs on $(c_j)_{j\in[1,m]}$ and $(\theta_j)_{j\in[1,m]}$:

$$\begin{aligned} \frac{d}{dt}c_j(t) &= 2\varepsilon_j G_t(\theta_j(t))c_j(t), \\ \frac{d}{dt}\theta_j(t) &= \varepsilon_j V_t(\theta_j(t)). \end{aligned}$$

where $\varepsilon_j = a_j(0)\in\{-1,+1\}$ denotes whether the corresponding quantity appears in $\tau_{t,m}^+$ ($\varepsilon=+1$) or $\tau_{t,m}^-$ ($\varepsilon=-1$).

Time discretization. Simulating these ODEs via the discrete Euler scheme with step $\eta>0$, leads, for any iteration $k\in\mathbb{N}$, to:

$$\begin{aligned} c_j(k+1) &= \Big(1+2\eta\varepsilon_j G_k(\theta_j(k))\Big)c_j(k) \\ \theta_j(k+1) &= \theta_j(k+1) + \eta\varepsilon_j V_k(\theta_j(k)). \end{aligned} \tag{B.3}$$

Approximating integrals numerically. The only thing that needs to be dealt with numerically is estimating the values of G_t and V_t which are defined by integrals. With the discretization of the measures, we have:

$$\begin{aligned}
G_k(\theta) &= \int_{\varphi} \Big(\cos(\varphi) - \tilde{f}(\tau_k^+, \tau_k^-; \varphi)\Big)\, \tilde{\phi}(\varphi; \theta) \mathrm{d}\tilde{\gamma}(\varphi), \\
\tilde{f}(\tau_k^+, \tau_k^-; \varphi) &= \sum_{j=1}^{m} c_j(k) \tilde{\phi}(\theta_j(k); \varphi), \\
\tilde{\phi}\,(\theta; \varphi) &= \int_{r,s \in [-1,1]} \sigma\Big(r\cos(\varphi)\cos(\theta) + s\sin(\varphi)\sin(\theta)\Big) \mathrm{d}\tilde{\gamma}_{d_H}(r) \mathrm{d}\tilde{\gamma}_{d_\perp}(s).
\end{aligned}$$

We thus get:

$$G_k(\theta) = \int \frac{\psi(r, s; \theta, \varphi)}{m} \sum_{j=1}^{m} \Big(\cos(\varphi) - c_j(k)\psi(r', s'; \theta_j(k), \varphi)\Big) \mathrm{d}\tilde{\gamma}(\varphi) (\mathrm{d}\tilde{\gamma}_{d_H})^2(r, r')) (\mathrm{d}\tilde{\gamma}_{d_\perp})^2(s, s'),$$

with

$$\psi(r, s; \theta, \varphi) := \sigma\Big(r\cos(\varphi)\cos(\theta) + s\sin(\varphi)\sin(\theta)\Big).$$

Similarly, we have:

$$V_k(\theta) = \int \frac{\chi(r, s; \theta, \varphi)}{m} \sum_{j=1}^{m} \Big(\cos(\varphi) - c_j(k)\psi(r', s'; \theta_j(k), \varphi)\Big) \mathrm{d}\tilde{\gamma}(\varphi) (\mathrm{d}\tilde{\gamma}_{d_H})^2(r, r')) (\mathrm{d}\tilde{\gamma}_{d_\perp})^2(s, s'),$$

with

$$\begin{aligned}
\chi(r, s; \theta, \varphi) &:= \frac{\partial}{\partial\theta}\psi(r, s; \theta, \varphi) \\
&= \sigma'\Big(\cos(\theta)\cos(\varphi)r + \sin(\theta)\sin(\varphi)s\Big)\Big[-\sin(\theta)\cos(\varphi)r + \cos(\theta)\sin(\varphi)s\Big].
\end{aligned}$$

We use Monte-Carlo estimation through sampling to approximate the integrals against the five variables (φ, r, r', s, s') by drawing N samples from the corresponding distributions. We get:

$$\begin{aligned}
G_k(\theta_j(k)) &\approx \frac{1}{mN} \sum_{i=1}^{N} \sum_{l=1}^{m} \Psi_{ji}\Big(\cos(\Phi_i) - c_l(k)\tilde{\Psi}_{li}\Big), \\
\Psi_{ji}(k) &= \psi(R_i, S_i; \theta_j(k), \Phi_i) \\
\tilde{\Psi}_{ji}(k) &= \psi(R_i', S_i'; \theta_j(k), \Phi_i),
\end{aligned}$$

and similarly

$$\begin{aligned}
V_k(\theta_j(k)) &\approx \frac{1}{mN} \sum_{i=1}^{N} \sum_{j=1}^{m} \chi_{ji}\Big(\cos(\Phi_i) - c_l(k)\tilde{\Psi}_{li}\Big), \\
\chi_{ji}(k) &= \chi(R_i, S_i; \theta_j(k), \Phi_i),
\end{aligned}$$

where we have drawn the samples i.i.d. over $i \in [1, N]$:

$$\begin{aligned}\Phi_i &\sim \tilde{\gamma},\\ R_i, R_i' &\sim \tilde{\gamma}_{d_H},\\ S_i, S_i' &\sim \tilde{\gamma}_{d_\perp}.\end{aligned}$$

Iterations in the numerical simulation. Defining the vectors $c(k) = (c_j(k))_{j\in[1,m]}$, $\theta(k) = (\theta_j)_{j\in[1,m]}$, and $\varepsilon = (\varepsilon_j)_{j\in[1,m]}$, the update Equations (B.3) can then be written in terms of update rules using the matrices $\Psi(k) = (\Psi_{ji}(k))_{j,i\in[1,m]\times[1,N]}$, $\tilde{\Psi}(k) = (\tilde{\Psi}_{ji}(k))_{j,i\in[1,m]\times[1,N]}$, and finally $\chi = (\chi_{ji}(k))_{j,i\in[1,m]\times[1,N]}$, and the vectors $(\Phi, R, R', S, S') = (\Phi_i, R_i, R_i', S_i, S_i')_{i\in[1,N]}$, which are re-sampled at each iteration $k \in [0, K]$, where $K \in \mathbb{N}$:

$$\begin{aligned}c(k+1) &= (1 + 2\eta\varepsilon \odot \hat{G}_k) \odot c(k),\\ \theta(k+1) &= \theta(k) + \eta\varepsilon \odot \hat{V}_k,\end{aligned}$$

where

$$\begin{aligned}\hat{G}_k &= \frac{d}{N_b}\Psi\left(\cos(\Phi) - \frac{1}{m}\tilde{\Psi}^\top c(k)\right),\\ \hat{V}_k &= \frac{d}{N_b}\chi\left(\cos(\Phi) - \frac{1}{m}\tilde{\Psi}^\top c(k)\right),\end{aligned}$$

and $\odot$ denotes the Hadamard (element-wise) product of two vectors. One can compute the loss through sampling in a similar way.

Experimental value for α and parameters of the numerical simulation. For the numerical simulations, we fix the number of atoms of the measure (or equivalently the width of the network) to $m = 1,024$, the learning rate to $\eta = 5.10^{-3}$, the number of samples for the Monte-Carlo scheme to $N = 1,000$, and the total number of iterations to $K = 20,000$. The experimental value for α (see Section 3.4.2) is computed through $\alpha_{\text{exp}} = \tau^+_{m,K}([0, \pi/2])$, that is

$$\begin{aligned}\alpha_{\text{exp}} &= \frac{1}{m}\sum_{j\in J^+} c_j(K),\\ J^+ &:= \{j \in [1, m]\ ;\ \varepsilon_j = 1\}.\end{aligned}$$

As mentioned in the main text, the behaviour of the numerical simulation depends a lot on the step-size η. Some of the differences between our observations and our intuitive description of the limiting model (infinite-width and continuous time) can come from too big a step-size. We have thus run the numerical simulation with $\eta = 2.10^{-5}$ as well, for $K = 230,000$ steps but the same differences still appear (*e.g.*, $\tau^+_{m,k}([0, \pi/2])$ still grows larger than the theoretically expected limit α after some time, albeit by a smaller margin) and after the critical t^*, some negative

particles seem to go slightly beyond $\pi/2$, even with a very small step-size, a fact which cannot happen for the limiting model. Consequently, in Figure 3.3, the first histogram bin right after $\pi/2$ has been merged with the one before.

C - Appendix for Chapter 4

C.1 . Proximal step for the L^1 penalty

The function $\phi : t \in \mathbb{R} \mapsto t\nabla_i f(x) + Lt + \lambda|x_i + t|$ is convex and thus a given t is a minimizer if and only if $0 \in \partial\phi(t)$ where $\partial\phi(t)$ is the sub-gradient of ϕ at t, and is given by $\partial\phi(t) = \nabla_i f(x) + Lt + \lambda\partial\psi(t)$ where $\psi : t \mapsto |x_i + t|$ and $\partial\psi(t) = \text{sign}(x_i + t)$ if $t \neq -x_i$ and $\partial\psi(t) = [-1, 1]$ otherwise. If t is a minimizer, it must thus hold that $\nabla_i f(x) + Lt + \lambda\text{sign}(x_i + t) = 0$ if $t \neq x_i$, and if $t = -x_i$, it must hold that $\nabla_i f(x) - Lx_i \in [-\lambda, \lambda]$.

Case where $t \neq -x_i$. Then we have $L(x_i + t) + \lambda\text{sign}(x_i + t) = -\nabla_i f(x) + Lx_i$ which implies that $x_i + t$ and $-\nabla_i f(x) + Lx_i$ have the same sign, and thus that $L|x_i + t| + \lambda = |-\nabla_i f(x) + Lx_i|$, which ensures that in this case $|\nabla_i f(x) - Lx_i| > \lambda$. Using that $x_i + t$ and $-\nabla_i f(x) + Lx_i$ have the same sign, we thus have:

$$\begin{aligned} L|x_i + t| &= |-\nabla_i f(x) + Lx_i| - \lambda \\ x_i + t &= -\frac{1}{L}\nabla_i f(x) + x_i - \frac{\lambda}{L}\text{sign}(-\frac{1}{L}\nabla_i f(x) + x_i) \\ t &= -x_i + \left(-\frac{1}{L}\nabla_i f(x) + x_i\right)\left(1 - \frac{\lambda}{|Lx_i - \nabla_i f(x)|}\right). \end{aligned}$$

In any case, ϕ has a unique minimum: if $-\nabla_i f(x) + Lx_i \in [-\lambda, \lambda]$ then $t = -x_i$ is the only value for which $0 \in \partial\phi(t)$, and is thus the unique minimizer. If $|-\nabla_i f(x) + Lx_i| \geq \lambda$, $t = -x_i + \left(-\frac{1}{L}\nabla_i f(x) + x_i\right)\left(1 - \frac{\lambda}{|Lx_i - \nabla_i f(x)|}\right)$ is the only value for which $0 \in \partial\phi(t)$, and is thus the unique minimizer. The two cases can be summarized in a unique formula as below:

$$\operatorname*{argmin}_{t \in \mathbb{R}} \phi(t) = -x_i + \left(-\frac{1}{L}\nabla_i f(x) + x_i\right)\max\left(0, 1 - \frac{\lambda}{|Lx_i - \nabla_i f(x)|}\right).$$

C.2 . A proof of inequality (4.17)

We first start with a useful lemma:

Lemma C.2.1. *Let $\phi_\alpha(s) = (1 - \alpha s)^2$, and let $\beta \geq 2$. Then, if $\alpha \leq 1$, it holds:*

$$\frac{1}{\beta}\sum_{j=0}^{\lfloor\beta\rfloor} \phi_\alpha(j/\lfloor\beta\rfloor) \geq \int_0^{1/2} \phi_\alpha \geq \frac{1}{8}.$$

Proof. Define $s_j := j/\beta$ for $j \in \{0, \ldots, \lfloor\beta\rfloor\}$. ϕ_α is non-negative, and since $\alpha \leq 1$, ϕ_α is decreasing on $[0, 1]$. Thus, by comparing the sum to the integral,

it holds

$$\begin{aligned}\frac{1}{\beta}\sum_{j=0}^{\lfloor\beta\rfloor}\phi_\alpha(j/\beta) &\geq \sum_{j=0}^{\lfloor\beta\rfloor-1}(s_{j+1}-s_j)\phi_\alpha(s_j)\\ &\geq \sum_{j=0}^{\lfloor\beta\rfloor-1}\int_{s_j}^{s_{j+1}}\phi_\alpha(s)\mathrm{d}s\\ &\geq \int_0^{\lfloor\beta\rfloor/\beta}\phi_\alpha.\end{aligned}$$

Now by definition of the floor function it holds $\lfloor\beta\rfloor/\beta \geq (\beta-1)/\beta = 1-1/\beta$ which is $\geq 1/2$ because $\beta \geq 2$. Therefore, using that $1/2 \leq 1-\alpha s \leq 1$ for $s \in [0,1/2]$, and that the squared function is increasing on $[0,1]$, we have

$$\frac{1}{\beta}\sum_{j=0}^{\lfloor\beta\rfloor}\phi_\alpha(j/\beta) \geq \int_0^{1/2}\phi_\alpha \geq (1/2)\times(1/4) = 1/8$$

□

We now give the proof of inequality (4.17) which is adapted from the *"Mémoire de TER"* of Adrien Prevost, assuming $m \geq 4$.

Proof. Let $S := \{x \in \mathbb{R}^m : ||x-x^*||_1 \leq R\}$. For any $x \in S$, it holds by convexity:

$$f(x) - M^* \leq \langle \nabla f(x), x-x^*\rangle \leq ||\nabla f(x)||_\infty ||x-x^*||_1 \leq R||\nabla f(x)||_\infty.$$

Fix $x \in S$, and let $i^* \in [1,m]$ such that $M := ||\nabla f(x)||_\infty = |\nabla_{i^*} f(x)|$. Call $g_i := \max(0, M - K|i-i^*|)$, it holds $|\nabla f_i(x)| \geq M - K|i-i^*|$, and thus $|g_i| \leq |\nabla_i f(x)|$.
Case when $\frac{M}{K} \leq 2$**.** Then, it holds $||\nabla f(x)||^2 \geq ||\nabla f(x)||_\infty^2 = ||\nabla f(x)||_\infty^3/M \geq ||\nabla f(x)||_\infty^3/(2K)$.

Case when $\frac{M}{K} > 2$**.** Then, we have

$$\begin{aligned}||\nabla f(x)||^2 &\geq \sum_{i=1}^m g_i^2\\ &\geq M^2\sum_{i=1}^m \max\left(0, 1-\frac{K}{M}|i-i^*|\right)^2\\ &\geq M^2\sum_{j=0}^m \max\left(0,1-\frac{K}{M}j\right)^2 \text{ card}\,\{i\in[1,m] : |i-i^*| = j\}\\ &\geq M^2\sum_{j=0}^{\lfloor m/2\rfloor}\max\left(0,1-\frac{K}{M}j\right)^2,\end{aligned}$$

where the last line comes from the fact that for $j \in \{0, \dots, \lfloor m/2 \rfloor\}$, it holds card $\{i \in [1, m] : |i - i^*| = j\} \geq 1$. Indeed, if $i^* \in \{0, \dots, \lfloor m/2 \rfloor\}$, then we have $[0, \lfloor m/2 \rfloor] = \{|i - i^*| : i \in [i^*, i^* + \lfloor m/2 \rfloor]$. If $i^* \in [\lfloor m/2 \rfloor + 1, m]$, then $[0, \lfloor m/2 \rfloor] = \{|i - i^*| : i \in [i^* - \lfloor m/2 \rfloor, i^*]$. We thus get

$$||\nabla f(x)||^2 \geq \frac{M^3}{K} \sum_{j=0}^{\min(\lfloor m/2 \rfloor, \lfloor M/K \rfloor)} \frac{K}{M} \left(1 - \frac{K}{M} j\right)^2.$$

If $M/K \leq m/2$**,** then

$$||\nabla f(x)||^2 \geq \frac{M^3}{K} \sum_{j=0}^{\min(\lfloor M/K \rfloor)} \frac{K}{M} \left(1 - \frac{K}{M} j\right)^2.$$

we apply Lemma C.2.1 with $\alpha = 1$ and $\beta = M/K \geq 2$ to obtain that

$$||\nabla f(x)||^2 \geq \frac{M^3}{K} \times \frac{1}{8}.$$

If $M/K > m/2$**,** then since $K/M \geq 2/m$, we have

$$\begin{aligned} ||\nabla f(x)||^2 &\geq \frac{M^3}{K} \sum_{j=0}^{\min(\lfloor m/2 \rfloor)} \frac{K}{M} \left(1 - \frac{K}{M} j\right)^2, \\ &\geq \frac{M^3}{K} \sum_{j=0}^{\min(\lfloor m/2 \rfloor)} \frac{2}{m} \left(1 - \frac{Km}{2M} j \frac{2}{m}\right)^2. \end{aligned}$$

We apply Lemma C.2.1 with $\alpha = \frac{Km}{2M} \leq 1$ and $\beta = m/2$ which is ≥ 2 by the assumption that $m \geq 4$. We obtain that

$$||\nabla f(x)||^2 \geq \frac{M^3}{K} \times \frac{1}{8}.$$

We have thus shown that in any case it holds $||\nabla f(x)||^2 \geq \frac{M^3}{8K}$, which entails that

$$\frac{1}{2}||\nabla f(x)||^2 \geq \tau (f(x) - M^*)^3$$

with $\tau := \frac{1}{16KR^3}$. □

C.3 . Proof of Equation (4.19)

Proof. Let $V \in \mathbb{R} \backslash \{0\}$ and call $\phi(t) = tV + \frac{L}{2}t^2 + \lambda|t|$. Calling $M := \max(0, 1 - \frac{\lambda}{|V|})$, and $T := -\frac{V}{L}M$, it holds

$$\phi(T) = -\frac{|V|^2}{L} \left((1 - \lambda)M - \frac{M^2}{2}\right)$$

and the term between parenthesis is equal to 0 if $1 - \frac{\lambda}{|V|} \leq 0$ and to $\frac{1}{2}(1 - \frac{\lambda}{M})^2$ otherwise, which means that in any case it is equal to $\frac{1}{2}\max(0, 1 - \frac{\lambda}{|V|})^2$, which gives

$$\begin{aligned}\phi(T) &= -\frac{|V|^2}{2L}M^2 \\ &= -\frac{1}{2L}\max(0, |V| - \lambda)^2.\end{aligned}$$

□

Bibliography

Emmanuel Abbe, Enric Boix-Adsera, Matthew S. Brennan, Guy Bresler, and Dheeraj Nagaraj. The staircase property: How hierarchical structure can guide deep learning. *Advances in Neural Information Processing Systems*, 34:26989–27002, 2021.

Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. *arXiv preprint arXiv:2202.08658*, 2022.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 6158–6169, 2019.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Dyego Araújo, Roberto I. Oliveira, and Daniel Yukimura. A mean-field limit for certain deep neural networks, 2019.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

Kendall Atkinson and Weimin Han. *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction*, volume 2044. Springer, 01 2012. ISBN 978-3-642-25982-1. doi: 10.1007/978-3-642-25983-8.

Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *arXiv preprint arXiv:2205.01445*, 2022.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

Andrew Barron. Barron, a.e.: Universal approximation bounds for superpositions of a sigmoidal function. ieee trans. on information theory 39, 930-945. *Information Theory, IEEE Transactions on*, 39:930 – 945, 06 1993. doi: 10.1109/18.256500.

Mohsen Bayati and Andrea Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57(2):764–785, 2011.

Yoshua Bengio, Nicolas Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2005/file/0fc170ecbb8ff1afb2c6de48ea5343e7-Paper.pdf.

Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *J. Mach. Learn. Res.*, 21:90–1, 2020.

Erwin Bolthausen. An iterative construction of solutions of the TAP equations for the Sherrington–Kirkpatrick model. *Communications in Mathematical Physics*, 325(1):333–366, 2014.

Kristian Bredies and Dirk A Lorenz. Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications*, 14:813–837, 2008.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: Circuits. *Distill*, 2020. doi: 10.23915/distill.00024. https://distill.pub/2020/circuits.

Lenaic Chizat. Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, 194(1):487–532, 2022.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3040–3050, 2018.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/ae614c557843b1df326cb29c57225459-Paper.pdf.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.07289.

Alexander Cloninger and Timo Klock. A deep network construction that adapts to intrinsic dimensionality beyond the domain. *Neural Networks*, 141:404–419, 2021.

Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, pages 5413–5452. PMLR, 2022.

Hadi Daneshmand and Francis Bach. Polynomial-time sparse measure recovery. *arXiv preprint arXiv:2204.07879*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

Weinan E, Chao Ma, and Lei Wu. Machine learning from a continuous viewpoint, i. *Science China Mathematics*, 63(11):2233–2266, sep 2020. doi: 10.1007/s11425-020-1773-8. URL https://doi.org/10.1007%2Fs11425-020-1773-8.

Cong Fang, Jason D. Lee, Pengkun Yang, and Tong Zhang. Modeling from features: a mean-field framework for over-parameterized deep neural networks, 2020.

Thomas Gallouët, Maxime Laborde, and Leonard Monsaingeon. An unbalanced optimal transport splitting scheme for general advection-reaction-diffusion problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 25:8, 2019.

Iordan Ganev and Robin Walters. The qr decomposition for radial neural networks. *arXiv preprint arXiv:2107.02550*, 2021.

Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. *arXiv preprint arXiv:1808.05587*, 2018.

Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stephane d'Ascoli, Giulio Biroli, Clement Hongler, and Matthieu Wyart. Scaling

description of generalization with number of parameters in deep learning. *Journal Of Statistical Mechanics-Theory And Experiment*, 2020(ARTICLE):023401, 2020a.

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020b.

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020c.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

Grzegorz Głuch and Rüdiger Urbanke. Noether: The more things change, the more stay the same. *arXiv preprint arXiv:2104.05508*, 2021.

Eugene Golikov and Greg Yang. Non-gaussian tensor programs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 21521–21533. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8707924df5e207fa496f729f49069446-Paper-Conference.pdf.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. In *International Conference on Learning Representations*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *CoRR*, abs/1806.07572, 2018. URL http://arxiv.org/abs/1806.07572.

Arthur Jacot, Franck Gabriel, François Ged, and Clément Hongler. Order and chaos: Ntk views on dnn normalization, checkerboard and boundary artifacts. *arXiv preprint arXiv:1907.05715*, 2019.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811. Springer, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Vera Kurková and Marcello Sanguineti. Bounds on rates of variable-basis and neural-network approximation. *Information Theory, IEEE Transactions on*, 47: 2659 – 2665, 10 2001. doi: 10.1109/18.945285.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324, 1998.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Flavio Martinelli, Berfin Simsek, Johanni Brea, and Wulfram Gerstner. Expand-and-cluster: Exact parameter recovery of neural networks. *arXiv preprint arXiv:2304.12794*, 2023.

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Hrushikesh Mhaskar. On the tractability of multivariate integration and approximation by neural networks. *J. Complexity*, 20:561–590, 08 2004. doi: 10.1016/j.jco.2003.11.004.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Alireza Mousavi-Hosseini, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A Erdogdu. Neural networks efficiently learn low-dimensional representations with sgd. *arXiv preprint arXiv:2209.14863*, 2022.

Radford M Neal. *BAYESIAN LEARNING FOR NEURAL NETWORKS*. PhD thesis, University of Toronto, 1995.

Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

Phan-Minh Nguyen and Huy Tuan Pham. A rigorous framework for the mean field limit of multilayer neural networks. *CoRR*, abs/2001.11443, 2020. URL https://arxiv.org/abs/2001.11443.

Atsushi Nitanda and Taiji Suzuki. Stochastic particle gradient descent for infinite ensembles. *arXiv preprint arXiv:1712.05438*, 2017.

Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

Jonas Paccolat, Leonardo Petrini, Mario Geiger, Kevin Tyloo, and Matthieu Wyart. Geometric compression of invariant manifolds in neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(4):044001, apr 2021. doi: 10.1088/1742-5468/abf1f3. URL https://dx.doi.org/10.1088/1742-5468/abf1f3.

Huy Tuan Pham and Phan-Minh Nguyen. A note on the global convergence of multilayer neural networks in the mean field regime. *CoRR*, abs/2006.09355, 2020. URL https://arxiv.org/abs/2006.09355.

Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999. doi: 10.1017/S0962492900002919.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Grant Rotskoff and Eric Vanden-Eijnden. Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, 2018.

Grant M. Rotskoff and Eric Vanden-Eijnden. Trainability and accuracy of neural networks: An interacting particle system approach, 2019.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2): 725–752, 2020.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of deep neural networks. *Mathematics of Operations Research*, 2021.

A. M. Turing. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX (236):433–460, 10 1950. ISSN 0026-4423. doi: $10.1093/\mathrm{mind}/\mathrm{LIX}.236.433$. URL https://doi.org/10.1093/mind/LIX.236.433.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

E. Weinan and Stephan Wojtowytsch. On the banach spaces associated with multi-layer relu networks: Function representation, approximation theory and gradient descent dynamics. *ArXiv*, abs/2007.15623, 2020.

Stephan Wojtowytsch. On the convergence of gradient descent training for two-layer relu-networks in the mean field regime. *arXiv preprint arXiv:2005.13530*, 2020.

Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5e69fda38cda2060819766569fd93aa5-Paper.pdf.

Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *ArXiv*, abs/2006.14548, 2020a.

Greg Yang. Tensor programs III: neural matrix laws. *CoRR*, abs/2009.10685, 2020b. URL https://arxiv.org/abs/2009.10685.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the*

38th International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 11727–11737. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.