Mathematics of deep learning : generalization, optimization, continuous-time models

Pierre Marion

SORBONNE UNIVERSITÉ

Mathematics of deep learning: generalization, optimization, continuous-time models

Thèse de doctorat de Sorbonne Université
préparée au Laboratoire de Probabilités, Statistique et Modélisation

Discipline : Mathématiques appliquées
Spécialité : Statistique

École doctorale de Sciences Mathématiques de Paris Centre, ED 386

Thèse soutenue à Paris le 20 novembre 2023 par

Pierre Marion

Rapporteurs et membres du jury :

Peter Bartlett Professeur, UC Berkeley	*Rapporteur*
Christophe Giraud Professeur, Université Paris-Saclay	*Rapporteur*
Francis Bach Directeur de recherche, Inria	*Président du jury*
Quentin Berthet Research scientist, Google DeepMind	*Examinateur*
Claire Boyer Maître de conférences, Sorbonne Université	*Examinatrice*
Stéphane Chrétien Professeur, Université Lyon 2	*Examinateur*
Anna Korba Maître de conférences, ENSAE	*Examinatrice*
Gérard Biau Professeur, Sorbonne Université	*Directeur de thèse*
Jean-Philippe Vert Chief R&D Officer, Owkin	*Co-directeur de thèse*

Thèse de doctorat

This work was supported by the Paris Ile-de-France Region via the DIM Math Innov program. We also acknowledge support from Google via a Google PhD Fellowship and from MINES Paris - PSL.

Abstract

Deep learning has emerged as a transformative paradigm in the past decade, with major impact in various fields of artificial intelligence. However, the properties of this family of machine learning methods are not yet fully understood. In this PhD thesis, we present contributions, mostly theoretical in nature, to the field of deep learning. We study various families of neural networks (shallow neural networks, residual networks, recurrent networks, Transformer) and various types of mathematical problems, most notably in the fields of statistics (generalization bounds) and optimization (convergence of the gradient flow). A first setting that is of particular interest for us is the large-depth limit of residual networks. It has been remarked that this large-depth limit may correspond to a neural ordinary differential equations. Under appropriate conditions, we show that it is indeed the case, although other limits such as stochastic differential equations can also hold. We investigate optimization and statistical properties of neural networks in this setting. In the second part of the thesis, we move on to prove results on finite-depth neural networks. We prove convergence of gradient flow for shallow neural networks with a moderate number of neurons in a simple setting. Finally, we investigate properties of the more recent Transformer architecture from a more practical point of view.

Keywords: theory of deep learning, neural networks, statistical learning, non-convex optimization

Résumé

L'apprentissage profond a largement transformé le paysage de l'apprentissage automatique au cours de la dernière décennie, avec un impact majeur dans divers domaines de l'intelligence artificielle. Cependant, les propriétés des méthodes d'apprentissage profond ne sont pas encore entièrement comprises. Dans cette thèse de doctorat, nous présentons des contributions, principalement d'ordre théorique, dans ce domaine. Nous étudions différentes familles de réseaux neuronaux (réseaux neuronaux à une couche cachée, réseaux résiduels, réseaux récurrents, Transformer) et différents types de problèmes mathématiques, notamment en statistique (bornes de généralisation) et en optimisation (convergence du flot de gradient). Dans un premier temps, nous nous intéressons à la limite en grande profondeur des réseaux résiduels. Il a été remarqué dans la littérature que cette limite en grande profondeur pourrait correspondre à une équation différentielle ordinaire neuronale. Sous des conditions appropriées, nous montrons que c'est effectivement le cas, bien que d'autres objets limites peuvent aussi apparaître, en particulier une équation différentielle stochastique. Nous étudions les propriétés d'optimisation et statistiques des réseaux neuronaux dans ce cadre. Dans la deuxième partie de la thèse, nous nous intéressons à des réseaux neuronaux de profondeur finie. Nous prouvons la convergence du flot de gradient pour des réseaux à une couche cachée avec un nombre modéré de neurones dans un cadre simple. Enfin, nous étudions les propriétés de l'architecture plus récente du Transformer avec une approche plus pratique.

Mots-clefs : théorie de l'apprentissage profond, réseaux de neurones, apprentissage statistique, optimisation non-convexe

Remerciements

Je tiens à remercier en premier lieu mes directeurs de thèse, Gérard et Jean-Philippe. Gérard, ta confiance dès notre première rencontre fin 2018, puis ta présence au quotidien, ton humour, ta gentillesse, l'autonomie que tu m'as laissée, ont été cruciaux pour moi tout au long de ma thèse. Nos séances de relectures interminables mais infaillibles seront inoubliables pour moi, ainsi que l'alignement des slides à la règle et au compas. Jean-Philippe, merci pour tes conseils avisés, ton recul, ta bonne humeur, ta capacité à comprendre et à proposer des solutions à mes questions de maths en moins de temps qu'il n'en faut pour dire condition de Polyak-Łojasiewicz. Je ne serai pas là où j'en suis sans ton aide.

Je remercie également les membres du jury, et en premier lieu les rapporteurs de la thèse, pour m'avoir consacré de leur temps que je sais précieux et pour leurs remarques pertinentes. Thank you to the members of the thesis committee, and foremost to the referees, for their precious time and relevant remarks.

Merci aux co-auteurs des travaux présentés dans ce manuscrit, Adeline, Francesco, Gérard, Jean-Philippe, Michael, Paweł, Raphaël, Yu-Han. Travailler avec chacun d'entre vous a été une grande source de joie et d'apprentissage pour moi. J'ai une pensée particulière pour Adeline avec qui j'ai travaillé pendant la période difficile du Covid au début de ma thèse. Cette première année de thèse n'aurait pas été la même sans toi et je t'en remercie. Thanks to the co-authors of the papers presented in this manuscript, Adeline, Francesco, Gérard, Jean-Philippe, Michael, Paweł, Raphaël, Yu-Han. Working with each of you has been a great source of joy and learning for me.

Merci à l'équipe du Groupe de Travail des Thésard·e·s, Alexis, Antonio, Loïc, Miguel, Nicolas. Longue vie au GTT (qui s'appelle désormais le SD, j'ai un pied dans la porte et je suis déjà un boomer) !

Merci aux habitués de l'éphémère groupe de lecture des doctorants de statistique et machine learning, en particulier Alexis, Ariane, Iqraa, Ludovic, Miguel. Ce fut bref mais intense.

Merci à mes co-chargés de TD, Alexis, Alice, Gloria, Iqraa, Ludovic, Miguel, Paul, Pierre. Comme disait le général de Gaulle, des chercheurs qui cherchent, on en trouve ; mais des personnes qui forment une aussi bonne équipe que vous pour enseigner, c'est moins courant. Ou quelque chose dans ce goût-là, je ne suis plus sûr.

Merci aux équipes d'organisation de NeurIPS@Paris 2021 et 2022, Adeline, Edouard, Francis, Gauthier, Gérard, Jean-Philippe, Jules, Linus, Liva. On ne réunit pas si facilement deux cents personnes pendant une journée au milieu du Covid, mais nous y sommes arrivés !

Merci à l'administration qui nous accompagne au quotidien, avec une pensée particulière pour Corentin, Hugues, Kevin, Louise, Nora. Science sans administration pour nous aider à remplir tous les papiers et à faire marcher nos ordinateurs n'est que ruine de l'âme.

Merci aux habitués de la salle café et des mots croisés, en particulier Anna, Anna, Antoine, Arnaud, Claire, Stéphane, vous m'avez ouvert des horizons pour une reconversion comme cruciverbiste. Je ne suis pas certain de mettre à profit cette expérience dans mon prochain travail, mais je le garde dans un coin de ma tête en cas de crise existentielle.

Merci à mes co-bureaux, Ariane, Cyprien, Eddy, Iqraa. Notre bureau fut marqué par un gradient assez notable dans la direction de la fenêtre en termes de décoration. Mine de rien, mon style épuré a fait le quart du travail et son entretien demande plus de soin qu'il n'y paraît au premier abord. Ne me remerciez pas, c'est tout naturel.

Merci à tous les doctorantes et doctorants du LPSM que j'ai côtoyés pendant ces trois années. Je prends le risque de faire une liste dont je sais qu'elle sera non exhaustive, et je m'excuse platement auprès de celles et ceux que je n'aurais pas cité·e·s. Merci Adeline, Alexandra, Alexis, Alice, Antonio, Ariane, Camila, Cyprien, Eddy, Francesco, Gloria, Iqraa, Lucas, Ludovic, Miguel, Nathan, Nicklas, Paul, Thibault, Ugo, Yazid. Ma thèse n'aurait pas été la même sans vous. Sortant de temps en temps du couloir 15-25 2ème étage, j'ai eu l'occasion de rencontrer des doctorantes et doctorants d'autres laboratoires. Je remercie celles et ceux qui ont croisé ma route à de nombreuses reprises et avec qui les discussions furent toujours enrichissantes. Merci Bénédicte, Clément, Corentin, Eloïse, Guillaume, Linus, Margaux, Robin.

Merci à tous les professeurs de science qui m'ont transmis leur savoir, vous êtes trop nombreux pour que je vous cite tous ici, mais vous faites un travail formidable ! J'ai une pensée particulière pour MM. Choimet, Presle, Ridde, et Mme Vince. Si les dessins dans les preuves et les sapins de Noël n'ont plus de secret pour moi, c'est grâce à vous.

Je suis le dernier (du moins dans un futur proche) à rendre hommage à Godalle Marmanthier. S'il fallait dresser un portrait d'iel, je dirais que c'est d'abord des rencontres. Des gens qui m'ont tendu la main, peut-être à un moment où je ne pouvais pas, où j'étais seul chez moi. Comme on dit dans le milieu des mathématiciens gaulois.

Je tiens à remercier mes ami·e·s, en particulier celles et ceux que je n'ai pas déjà mentionné·e·s ci-dessus (car oui, étonnamment, je fréquente des gens qui font autre chose qu'une thèse en mathématiques). Je ne connais pas la moitié d'entre vous à moitié autant que je le voudrais, et j'aime moins que la moitié d'entre vous à moitié aussi bien que vous le méritez. Que je vous connaisse depuis un an ou dix, merci pour tous les moments que nous avons passés ensemble.

Je ne saurais en ces lignes conclusives exprimer ma gratitude envers ma famille à la mesure de qu'elle mériterait. Merci Claude pour ta gentillesse et ta présence. Merci à ma mère de m'avoir toujours soutenu et de m'avoir donné le goût des mathématiques (mais également de LaTeX !). Merci Maxime d'avoir été à mes côtés de manière constante et inconditionnelle depuis cinq ans, et j'espère pour longtemps encore.

Contributions and thesis outline

The thesis is organized in two parts, preceded by an introduction and followed by a conclusion. Each part is separated in several chapters, which each correspond to a standalone contribution. As a consequence, the chapters are independent and self-contained. The notation may vary from chapter to chapter, although we try to keep some identical conventions throughout the thesis. Each chapter has led to or should lead to a publication, as detailed below.

Part I: From discrete to continuous architectures, neural networks in the large-depth regime

This part is dedicated to the mathematical analysis of the large-depth limit of residual networks, both from a statistical and optimization point of view.

[Marion et al., 2022] *Scaling residual networks in the large-depth regime*, P.M., Adeline Fermanian (Sorbonne Université at the time, now Califrais), Gérard Biau (Sorbonne Université), and Jean-Philippe Vert (Google Research at the time, now Owkin). Submitted.

[Marion et al., 2023] *Implicit regularization of deep residual networks towards neural ODEs*, P.M., Yu-Han Wu (Sorbonne Université), Michael E. Sander (ENS Paris), and Gérard Biau. Submitted.

[Marion, 2023] *Generalization bounds for neural ordinary differential equations and deep residual networks*. Published at NeurIPS 2023.

[Fermanian et al., 2021] *Framing RNN as a kernel method: A neural ODE approach*, Adeline Fermanian, P.M., Jean-Philippe Vert, and Gérard Biau. Published at NeurIPS 2021 (oral presentation).

Part II: Contributions to finite-depth neural networks

This part gathers two contributions related to modern topics in deep learning, this time for finite-depth neural networks. The presented contributions are the following:

[Marion and Berthier, 2023] *Leveraging the two-timescale regime to demonstrate convergence of neural networks*, P.M. and Raphaël Berthier (EPFL). Published at NeurIPS 2023.

[Marion et al., 2021] *Structured context and high-coverage grammar for conversational question answering over knowledge graphs*, P.M., Paweł K. Nowak (Google Research) and Francesco Piccinno (Google Research). Published at EMNLP 2021. This contribution differs from the others in the thesis since it has a more applied flavor. It follows a work carried out during an internship at Google Research in 2020.

Contents

1 Introduction

This introduction presents the general context of the manuscript and an overview of our contributions. We start by an introduction to the mathematics of deep learning and to some deep learning architectures in Section 1.1. In the following sections, we delve more into the details of the models we consider in this thesis, to introduce the specific context to each model and present our contributions. Section 1.2 presents the first part of this manuscript (Chapters 2 to 5), while Section 1.3 discusses the second part of the manuscript (Chapter 6 and 7). Section 1.4 gives a detailed summary of our work in French.

Contents

1.1 Mathematics of deep learning

1.1.1 Why study the mathematics of deep learning?

Deep learning is a subset of machine learning regrouping a large variety of algorithms that have three major characteristics. First, they involve an iterative data-driven optimization procedure, whose goal is to tune a parameterized function called a neural network. Second, the neural network can take different forms but is always a non-convex function of its parameters, which involves the composition of successive elementary parameterized operations. Third, the number of parameters of this function is large, and thus they also require a massive amount of data and compute to be tuned. The algorithm returns the trained neural network, which can then be used to perform a given task (classification, regression, generation, etc.).

Deep learning methods have allowed major breakthroughs in various fields in the past decade, such as computer vision or natural language processing, by taking advantage of an explosion in available compute resources and data, as well as algorithmic improvements. More recently, deep learning is behind the current successes in generative artificial intelligence (epitomized by chatGPT), and the renewed expectancy of major societal impacts of artificial intelligence.

The great empirical successes and even greater promises of deep learning methods call for a mathematical theory of neural networks. More precisely, the need for strong mathematical grounding of deep learning is justified for at least three major reasons: efficiency, effectiveness, explainability.

First, the call for more efficient learning approaches has been growing, in order to make it less resource-intensive and accessible to a wider community. This concern is particularly important to limit the environmental impact of machine learning.

Second, a deeper understanding of the fundamental underpinnings for the empirical success of deep learning may lead to proposing more effective methods, by unlocking some of the issues in the field. For instance, large language models such as chatGPT have a known tendency to produce false statements. Designing algorithms with theoretical guarantees on the validity of their outputs is therefore a crucial endeavor.

Third, explainability is necessary to foster adoption in critical applications (health, banking, cyber-security, transports, etc.) and will probably also be increasingly demanded in the Internet industry.

It is essential to emphasize at this point that deep learning is an extremely active area of research encompassing efforts at the intersection of many fields in mathematics (approximation theory, probability, statistics, optimization) and computer science (algorithmic, software engineering, hardware design), not to mention all the fields of applications. Therefore, our ambition in the following is not to give an exhaustive presentation of deep learning, but rather to introduce the models and ideas which are useful to understand the main contributions of this manuscript. We refer to the textbooks by Goodfellow et al. (2016) and Fleuret (2023) for general introductions to deep learning, and to Anthony and Bartlett (1999) for an introduction to the statistical theory of deep learning. Section 1.1.2 presents the general framework of deep learning theory, in the simplest case of shallow neural networks. We then briefly present in Section 1.1.3 the main deep learning models this thesis will be concerned with. In Section 1.1.4, we describe where our contributions stand in the landscape of deep learning theory and models.

1.1.2 What are the main mathematical challenges of deep learning?

In this section, we give an overview of some main questions related to deep learning theory. To fix ideas, we take as our running example a simple setting, namely shallow neural network trained with stochastic gradient descent on a non-parametric regression task. Although this algorithm

has been replaced in practice by empirically stronger methods, it remains a central object for mathematical study, in particular to understand its optimization and statistical properties. Furthermore, note that most of the exposition in this section applies to many other settings in deep learning and beyond in machine learning in general.

Shallow neural networks. Let us begin by describing our class of neural networks. Shallow neural networks are parameterized functions

$$f_{v,W,b} : x \mapsto v \cdot \sigma(Wx + b), \tag{1.1}$$

where $x \in \mathbb{R}^d$ is the input, the outputs belongs to $\mathbb{R}$, while $v \in \mathbb{R}^q$, $W \in \mathbb{R}^{q \times d}$ and $b \in \mathbb{R}^q$ are the parameters of the neural network. v and W are referred to as weights and b as the bias. Finally, $\sigma : \mathbb{R} \to \mathbb{R}$ is a function applied component-wise, called an activation function. Common activation functions include the ReLU function $x \mapsto \max(x, 0)$, the logistic function $x \mapsto (1 + \exp(-x))^{-1}$, and the hyperbolic tangent function $x \mapsto (e^x - e^{-x})/(e^x + e^{-x})$. The shallow neural network can equivalently be written

$$f_{v,W,b}(x) = \sum_{k=1}^{q} v_k \sigma(W_k \cdot x + b_k), \tag{1.2}$$

where the triple (v_k, W_k, b_k) is referred to as a neuron. In the following of this section, for simplicity, we omit the bias b, the exposition easily extends to the case where it is present.

Non-parametric regression. Take X a random variable in $\mathbb{R}^d$, $f^* : \mathbb{R}^d \to \mathbb{R}$ a regression function, and $Y = f^*(X) + \varepsilon$, where ε is some additive noise independent of X. The regression problem is the following: given an i.i.d. sample $(x_i, y_i)_{1 \leqslant i \leqslant n}$ with the same distribution as (X, Y), find v and W such that $f_{v,W} \approx f^*$. More formally, can we find v and W such that $\mathbb{E}(|f_{v,W}(X) - f^*(X)|^2)$ is small? Note that this is equivalent to finding (v, W) such that

$$\mathcal{R}(v, W) := \mathbb{E}(|f_{v,W}(X) - Y|^2) \approx \mathbb{E}(|f^*(X) - Y|^2) = \min_{f \text{ measurable}} \mathbb{E}(|f(X) - Y|^2),$$

or in other words, to find $f_{v,W}$ achieving the optimal Bayes risk for predicting Y given X.

The stochastic gradient descent (SGD) algorithm aims at solving this problem by tuning (or training) the parameters v and W by an iterative optimization procedure. The algorithm picks at each step a data point (x_i, y_i) and updates the parameters in the direction opposite to their gradient with respect to the squared loss evaluated at $(F_{v,W}(x_i), y_i)$. The pseudocode is presented in Algorithm 1.

We now present a first line of theoretical analysis of this algorithm, which is an instantiation of the classical statistical learning framework.

1.1.2.1 The classical waltz (uniform law of large numbers)

The analysis of Algorithm 1, and beyond of many machine learning algorithms, has historically been decomposed in three subproblems, each corresponding to a field of applied mathematics. More precisely, given classes of parameters $\mathcal{V}$ and $\mathcal{W}$, this analysis requires the three following conditions to hold simultaneously (see, e.g., Bach, 2023, Sections 4.2 and 5.1, for similar decompositions).

- **Approximation:** there exist weights $v^* \in \mathcal{V}$ and $W^* \in \mathcal{W}$ such that f_{v^*,W^*} is close to f^* (according to some norm), or in other words, such that

$$\mathcal{R}(v^*, W^*) \approx \min_{f \text{ measurable}} \mathbb{E}(|f(X) - Y|^2).$$

Algorithm 1 Training of shallow neural networks with stochastic gradient descent on a regression task

Input: Sample $(x_i, y_i)_{1 \leqslant i \leqslant n}$, initial weights v_0 and W_0, learning rate γ, number of steps P
Output: Trained weights v_{final} and W_{final}

Let $r(x, y, v, W) = |f_{v,W}(x) - y|^2$.
for $k = 1, 2, \dots, P$ **do**
    Choose uniformly at random some $i \in \{1, \dots n\}$.
    $v_{k+1} \leftarrow v_k - \gamma \frac{\partial r}{\partial v}(x_i, y_i, v_k, W_k)$.
    $W_{k+1} \leftarrow W_k - \gamma \frac{\partial r}{\partial W}(x_i, y_i, v_k, W_k)$.
end for
Return $(v_{\text{final}}, W_{\text{final}}) = (v_P, W_P)$.

- **Statistics:** for any $(v, W) \in (\mathcal{V}, \mathcal{W})$, the difference between the empirical risk

$$\hat{\mathcal{R}}_n(v, W) := \frac{1}{n} \sum_{i=1}^{n} |f_{v,W}(x_i) - y_i|^2 \tag{1.3}$$

  and the theoretical risk

$$\mathcal{R}(v, W) = \mathbb{E}(|f_{v,W}(X) - Y|^2)$$

  is small, typically decaying to zero as n grows, uniformly over all $(v, W) \in (\mathcal{V}, \mathcal{W})$.

- **Optimization:** Algorithm 1 converges to v_{final} and W_{final} such that the empirical risk

$$\hat{\mathcal{R}}_n(v_{\text{final}}, W_{\text{final}}) = \frac{1}{n} \sum_{i=1}^{n} |f_{v_{\text{final}}, W_{\text{final}}}(x_i) - y_i|^2$$

  is small, typically not far away from the global minimum of the empirical risk over all possible parameters $(v, W) \in (\mathcal{V}, \mathcal{W})$.

If these three conditions hold at the same time, then simple algebra shows that it is possible to bound the difference between the theoretical risk of the trained network $\mathcal{R}(v_{\text{final}}, W_{\text{final}})$ and the Bayes risk $\min_{f \text{ measurable}} \mathbb{E}(|f(X) - Y|^2)$. Indeed, denoting $(\hat{v}_n, \hat{W}_n)$ a minimizer of the empirical risk $\hat{\mathcal{R}}_n$ over $(\mathcal{V}, \mathcal{W})$, we have

$$\begin{aligned}
\mathcal{R}(v_{\text{final}}, W_{\text{final}}) - \min_{f \text{ measurable}} \mathbb{E}(|f(X) - Y|^2) \leqslant\ & |\mathcal{R}(v_{\text{final}}, W_{\text{final}}) - \hat{\mathcal{R}}_n(v_{\text{final}}, W_{\text{final}})| \\
& + \hat{\mathcal{R}}_n(v_{\text{final}}, W_{\text{final}}) - \hat{\mathcal{R}}_n(\hat{v}_n, \hat{W}_n) \\
& + |\hat{\mathcal{R}}_n(\hat{v}_n, \hat{W}_n) - \mathcal{R}(v^*, W^*)| \\
& + \mathcal{R}(v^*, W^*) - \min_{f \text{ measurable}} \mathbb{E}(|f(X) - Y|^2).
\end{aligned}$$

The four terms can be controlled under the conditions presented above. The connection is straightforward for all of them, except perhaps for the third term, which can be controlled by the statistical error since

$$\begin{aligned}
|\hat{\mathcal{R}}_n(\hat{v}_n, \hat{W}_n) - \mathcal{R}(v^*, W^*)| &= \Big| \inf_{(v,W) \in (\mathcal{V}, \mathcal{W})} \hat{\mathcal{R}}_n(v, W) - \inf_{(v,W) \in (\mathcal{V}, \mathcal{W})} R(v, W) \Big| \\
&\leqslant \sup_{(v,W) \in (\mathcal{V}, \mathcal{W})} |\hat{\mathcal{R}}_n(v, W) - R(v, W)|.
\end{aligned}$$

Let us now examine whether it is reasonable to hope that the three conditions above be satisfied.

Approximation. The first approximation results for shallow neural networks were proven in a landmark paper by Cybenko (1989). This paper shows that shallow neural networks are universal approximators of continuous functions when the width q of the hidden layer can grow arbitrarily large, as a consequence of the Stone-Weierstrass theorem. We refer to DeVore et al. (2021) for a review of more recent and sophisticated results including rates of approximation on various function spaces.

Statistics. Bounding the difference between the empirical and the theoretical risk requires a uniform law of large numbers, in the sense that it amounts to bounding the difference between an expectation and an empirical mean uniformly over a function class. This condition is typically proven by bounding the capacity of the function class, here a class of neural networks. Statistical learning provides us with a toolbox to do so. For instance, using covering numbers argument, Anthony and Bartlett (1999, Theorem 19.2) show that, under the assumption that $f_{v,W}$ takes its values in $[0,1]$ for all $(v,W) \in (\mathcal{V},\mathcal{W})$, we have

$$\mathbb{P}\big(\exists (v,W) \in (\mathcal{V},\mathcal{W}), |\hat{\mathcal{R}}_n(v,W) - \mathcal{R}(v,W)| \geqslant \varepsilon\big) \leqslant \mathcal{O}\Big(\Big(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon})\Big)^D \exp(-\varepsilon^2 n)\Big), \tag{1.4}$$

where the probability $\mathbb{P}$ holds over the sample $(x_i,y_i)_{1\leqslant i\leqslant n}$, and D is the so-called pseudo-dimension of the neural network, which can in turn be bounded by $\mathcal{O}(dq\log(dq))$ (Anthony and Bartlett, 1999, Theorems 8.8 and 14.1). In particular, if the sample size n is sufficiently large with respect to the pseudo-dimension of the neural network, then the right-hand side of (1.4) is small, and therefore the uniform bound on the difference between the empirical and the theoretical risks holds with high probability. Note that this implies in particular that the number of neurons q should be less than the sample size n. Nevertheless, it is also possible to construct bounds that do not depend explicitly on q but instead on the magnitude of the parameters measured according to some norm. Let us present a result in this spirit: denote $\|W\|_{1,\infty}$ the maximum of the ℓ_1-norm of the rows of W. Then, under the assumptions that $f_{v,W}$ takes its values in $[0,1]$ for all $(v,W) \in (\mathcal{V},\mathcal{W})$, that σ is Lipschitz-continuous and bounded, and that X is bounded almost surely, we have (Anthony and Bartlett, 1999, Corollary 14.16 and Theorem 17.1)

$$\mathbb{P}\big(\exists (v,W) \in (\mathcal{V},\mathcal{W}), |\hat{\mathcal{R}}_n(v,W) - \mathcal{R}(v,W)| \geqslant \varepsilon\big) \leqslant \mathcal{O}\Big(\exp\Big(\frac{M^6}{\varepsilon^4}\log d - \varepsilon^2 n\Big)\Big), \tag{1.5}$$

where M is such that

$$\forall (v,W) \in (\mathcal{V},\mathcal{W}), \|v\|_1 + \|W\|_{1,\infty} \leqslant M.$$

We see that in this case, the measure of statistical complexity of the neural network is not the number of parameters anymore, but rather a norm of the parameters. The key insight behind the proof is that a network with bounded weights can be approximated by one with few weights.

We refer to Bartlett et al. (2017, 2019) for more recent results on the statistical complexity of neural networks.

Optimization. The problematic time in our waltz is optimization: since the shallow neural network (1.1) is non-convex in W, the objective function $\hat{\mathcal{R}}_n$ of the optimization algorithm is also non-convex. This breaks the key convexity assumption ensuring that gradient-based optimization procedures (such as stochastic gradient descent) converge to a global minimum. For this reason, there is no guarantee that SGD converges to a global minimum, and there are even negative results describing settings where SGD for shallow neural networks converges to local minima with high probability (Brutzkus et al., 2018). Nevertheless, in practice, it is observed that trained neural networks do converge to very low training errors. For instance, it is shown in Zhang et al.

(2021) that it is easy to train neural networks to interpolation (i.e., zero training error), even with random labels. A framework to explain this phenomenon has emerged in the past years, which we present next.

1.1.2.2 The modern tango (large number of neurons, implicit regularization)

The classical decomposition into the trio approximation-statistics-optimization, which we sketched above, is not entirely satisfactory for several reasons. First, it provides no guarantee that the optimization algorithm converges close to a global minimizer of the empirical risk. Second, the statistical guarantees are typically given over a bounded subset of the space of parameters, and it is not a priori clear that the output of the optimization algorithm belongs to this subset. Finally, it has been remarked that uniform capacity control provably leads to vacuous bonds in some settings (Nagarajan and Kolter, 2019).

These reasons have prompted a more complex line of analysis (see Belkin, 2021, for a general presentation), which analyzes *jointly* the optimization and statistical aspects instead of relying on a uniform deviation bound over a class of parameters. This analysis holds when the number of neurons is large enough (typically larger than the sample size). The reasoning in this new line of analysis is two-fold: first, prove that, when the network is wide enough, the optimization landscape becomes more favorable, which allows proving *global convergence* of the empirical risk to zero despite the lack of convexity. Second, even more agreeable, the optimization algorithm does not converge towards just any minimizer of the empirical risk, but towards a minimizer that possesses structural properties. More precisely, among all minimizers of the empirical risk, the one found by SGD minimizes some measure of complexity of the parameters. In other words, the optimization algorithm *implicitly* solves the problem

$$\min_{\substack{v,W\in(\mathcal{V},\mathcal{W})\\ \hat{\mathcal{R}}_n(v,W)=0}} c(v,W),$$

where c is an appropriate measure of complexity, which is typically a norm (see below) or a matrix rank (Razin and Cohen, 2020). Furthermore, neural networks such that $c(v,W)$ is low enjoy favorable generalization properties, hence the name of *implicit regularization* given to this phenomenon.

Let us now delve a bit more into the details of the approach, first regarding the proof of global convergence, then the implicit regularization.

Global convergence of SGD. Regarding the first facet of the reasoning, global convergence of gradient-based algorithms for (shallow) neural networks can be explained in the case where the width q of the network is large enough with respect to the number of data n. Recent works in this direction can roughly be clustered in two main categories.

On the one hand, the neural tangent kernel (NTK) regime (Jacot et al., 2018; Allen-Zhu et al., 2019; Du et al., 2019; Zou et al., 2020a) corresponds to small movements of the parameters of the neural network. In this case, the neural network can be linearized around its initial point, and thus behaves like a linear regression. In spirit, provided that there are more parameters than data points, the linear regression is then overparameterized, which means that there exists a solution to the linear regression problem with a null empirical risk.

On the other hand, the mean-field regime (Chizat and Bach, 2018; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2020) describes the dynamics in the regime $q \gg 1$ through a partial differential equation (PDE) on the density of neurons, which takes the form of a Wasserstein gradient flow. It can then be shown that, under the assumption that this PDE converges, then the limit distribution must be optimal. This regime is sometimes

referred to as 'rich' (see, e.g. Woodworth et al., 2020) since it describes non-linear feature learning, contrarily to the NTK regime which is akin to a kernel method. The transition from the kernel to the rich regime depends on the scale of the initialization (Chizat et al., 2019; Woodworth et al., 2020).

Implicit regularization. A number of recent works study implicit regularization for neural networks. Let us present one specific example, which fits in the framework of regression with shallow neural networks presented above, in order to give the flavor of these results. Boursier et al. (2022) study the case of a ReLU activation function, a large enough number of neurons q and orthogonal inputs, which in particular implies that $d \geqslant n$. They show that, if the weights are initialized infinitesimally close to zero (corresponding to the rich regime described above), then the gradient flow converges to a global minimizer of the empirical risk, achieving a zero empirical risk. Furthermore, this minimizer is a solution of the problem

$$\min_{\substack{v,W \in \mathbb{R}^q \times \mathbb{R}^{q \times d} \\ \hat{\mathcal{R}}_n(v,W)=0}} \|v\|^2 + \|W\|^2,$$

where $\|\cdot\|$ denotes the Euclidean norm. This result can be rephrased in terms of the prediction function $f_{v,W}$: it shows that the prediction function found by gradient flow minimizes the so-called variation norm among all interpolators of the training data that write as an infinite-width neural network. Furthermore, it is known that prediction functions with low variation norm generalize well (Bach, 2017, Proposition 7).

Note that many other implicit regularization results have been shown, we refer to Vardi (2023) for a review.

Remark 1.1. *Since the analysis presented in this section holds when the number of neurons is large, it is often referred to as the 'overparameterized' regime, by opposition to an 'underparameterized' regime which would correspond to the analysis of the previous section. However, this formulation is somewhat improper since it was already known that uniform deviation bounds can hold independently of the number of parameters, as shown for instance by the bound* (1.4)*. It seems more precise to distinguish both analyses by stating that the uniform deviation bound is replaced by a characterization of the implicit regularization implied by the optimization algorithm.*

1.1.3 From shallow neural networks to Transformer

Having sketched the landscape of deep learning theory, we now present in this section some important neural network architectures by increasing level of complexity. Let us stress that we give a simple and unified presentation of the models, at the cost of generality. For instance, we remove biases and normalizations (batch normalization or layer normalization). More general models will be considered in the manuscript, with slight variations depending on the chapters.

Deep neural networks. The shallow neural network model introduced in Section 1.1.2 can be generalized to deep neural networks, which consists in composing linear mappings and non-linear component-wise activations, according to the relations

$$h_0 = Ax, \tag{1.6}$$
$$h_{k+1} = \sigma(W_{k+1}h_k), \quad 0 \leqslant k \leqslant L-1, \tag{1.7}$$
$$F(x) = Bh_L, \tag{1.8}$$

where L denotes the depth of the network, $A \in \mathbb{R}^{q \times d}$, $W_1, \dots, W_L \in \mathbb{R}^{q \times q}$, and $B \in \mathbb{R}^{d' \times q}$ are weight matrices, and σ is still an activation function. The quantity h_k is referred to as a hidden

layer. The weight matrices can either be full matrices, in which case the neural network is called fully-connected, or sparse with weight-sharing, corresponding to convolutional neural networks. In practice, convolutions play a critical role in the performance of neural networks for visions tasks, in particular because they encode symmetries of the network with respect to translation in the image. Nevertheless, we do not present them in more details here, since the results in this manuscript are given for fully-connected neural networks. We refer the interested reader to Goodfellow et al. (2016, Section 9) for a detailed presentation of convolutional neural networks.

Residual neural networks. In practice, vanilla deep neural networks (1.6)–(1.8) are difficult to train when L is larger than a few units, due to instabilities in the training procedure, as demonstrated in particular in a landmark paper by He et al. (2016a). These authors propose important improvements, allowing to reach depth of order to several hundreds or even thousands, thereby achieving (at the time) state-of-the-art results in image recognition. The crucial modification is the presence of skip connections, meaning that the definition of the hidden layer (1.7) is replaced by

$$h_{k+1} = h_k + \sigma(W_{k+1} h_k), \quad 0 \leqslant k \leqslant L-1. \tag{1.9}$$

Comparing (1.7) and (1.9), we see that, in the second case, the mapping $h \mapsto W_{k+1}\sigma(h)$ does not parameterize directly the new hidden layer h_{k+1} as in (1.7), but rather the difference between two successive hidden layers $h_{k+1} - h_k$, hence the name residual neural network given to this model. This intuition is key to the developments on the properties of this model made in the first part of this manuscript.

Recurrent neural networks. Note that, for now, we have placed ourselves in the simple case where the input and the output are vectors, belonging respectively to $\mathbb{R}^d$ and $\mathbb{R}^{d'}$. Another key context is where the input data consists in a collection of vectors $\mathbf{x} \in \mathbb{R}^{L \times d}$. This setting encompasses time series (in which case L corresponds to the number of time steps) or textual data (where L corresponds to the length of the text). We now present a first model that is adapted to this context, recurrent neural networks (Rumelhart et al., 1986). They are defined by a sequence of hidden states $h_1, \dots, h_L \in \mathbb{R}^q$. These hidden states obey the following recurrence, for an input $\mathbf{x} = (x_1, \dots, x_L)$:

$$\begin{aligned} h_0 &= 0, \\ h_{k+1} &= \sigma(W h_k + U x_{k+1}), \quad 0 \leqslant k \leqslant L-1 \\ F(x) &= B h_L, \end{aligned} \tag{1.10}$$

where σ is an activation function, and $W \in \mathbb{R}^{q \times q}$, $U \in \mathbb{R}^{q \times d}$, and $B \in \mathbb{R}^{d' \times q}$ are weight matrices. Note that here, contrarily to the models presented above, the weight matrices do not depend on the layer index k. Naturally, it is also possible to consider a residual version of recurrent neural networks (see, e.g., Yue et al., 2018). The iteration then writes

$$h_{k+1} = h_k + \sigma(W h_k + U x_{k+1}), \quad 0 \leqslant k \leqslant L-1. \tag{1.11}$$

Transformer. Recurrent neural networks have several drawbacks that make them unsuitable in some use cases (Kolen and Kremer, 2001); they are slow to train and suffer from instabilities during training. They also have difficulties picking up long-range dependencies in the data. An important mechanism to mitigate these problems is the so-called attention (Bahdanau et al., 2014; Luong et al., 2015), which is in particular present in Transformer (Vaswani et al., 2017). The Transformer architecture forms the basis of modern natural language processing and in particular of large language models. There are different variants of this architecture, we focus

here on the Transformer encoder (a.k.a. BERT-like Transformer) to give a flavor of the main ideas. Just like recurrent neural networks, the Transformer encoder handles a collection of vectors $\mathbf{x} = (x_1, \dots, x_L) \in \mathbb{R}^{L \times d}$. However, instead of performing sequential computations involving successively each x_i for $i \in \{1, \dots, L\}$, it manipulates the whole sequence at once. More precisely, the Transformer encoder consists in a series of blocks, where each block takes as input a matrix $\mathbf{x} \in \mathbb{R}^{L \times d}$ and returns a matrix $B(\mathbf{x}) \in \mathbb{R}^{L \times d}$, so several blocks can be composed together to form the encoder. Each block consists in two layers, an attention layer and a feedforward layer, which are defined as follows:

$$A(\mathbf{x}) = \mathbf{x} + \sum_{h=1}^{H} \sigma(\mathbf{x} Q_h K_h^\top \mathbf{x}^\top) \mathbf{x} V_h O_h^\top$$
$$B(\mathbf{x}) = A(\mathbf{x}) + \mathrm{ReLU}(A(\mathbf{x}) W_1) W_2$$

The parameters of the attention layer are Q_h, K_h, V_h, O_h that are all $\mathbb{R}^{d \times r}$ matrices, for $h \in \{1, \dots, H\}$, and the parameters of the feedforward layer are $W_1 \in \mathbb{R}^{d \times m}$ and $W_2 \in \mathbb{R}^{m \times d}$. The dimensions r and m are hyperparameters of the model. The function $\sigma : \mathbb{R}^{L \times L} \to \mathbb{R}^{L \times L}$ performs a row-wise softmax operation, that is, for $M \in \mathbb{R}^{L \times L}$, we have

$$\sigma(M)_{ij} = \frac{\exp(M_{ij})}{\sum_{k=1}^{L} \exp(M_{ik})}.$$

Finally, $\mathrm{ReLU} : x \mapsto \max(x, 0)$ is the activation function applied element-wise. We emphasize that the architecture presented here is a simplification of the actual Transformer models, which eludes important concepts (in particular the contrast between encoder-only, decoder-only, and encoder-decoder architectures). We refer to Phuong and Hutter (2022) for a more in-depth presentation of Transformer-related algorithms.

1.1.4 What to expect in this manuscript?

Now that we have presented a brief panorama of deep learning theory and of neural networks models, we are in a position to examine where our contributions stand in this landscape. The situation is summarized in Table 1.1, where the chapters are sorted by architecture, and Table 1.2, where the chapters are sorted according to the nature of the results.

Architecture	Type of results
Residual networks	Other properties (Chap. 2) Optimization (Chap. 3) Statistics (Chap. 4)
Recurrent networks	Statistics (Chap. 5)
Shallow networks	Optimization (Chap. 6)
Transformer	Applications (Chap. 7)

Table 1.1: Chapters of this manuscript organized by architecture.

1.2 From discrete to continuous architectures: neural networks in the large-depth regime [Part I of the manuscript]

The analysis of shallow models presented in Section 1.1.2 does not fully adapt to deep networks, and furthermore does not provide a consensual explanation for the role of depth, despite its

	Theory		Applications
	The classical waltz	The modern tango	
Statistics	Chap. 4 (residual networks) Chap. 5 (recurrent networks)		Chap. 7
Optimization	Chap. 3 (residual networks) Chap. 6 (shallow networks)	Chap. 3 (residual nets)	(Transformer)
Other properties	Chap. 2 (residual networks)		

Table 1.2: Chapters of this manuscript organized by nature of the results.

empirically-proven importance (see, e.g., Wang et al., 2022). This remark, along with the empirical success of residual networks (He et al., 2016a), motivated a line of research devoted to understanding the properties of residual networks in the limit where the depth tends to infinity. Let us consider in this section a more general formulation of residual networks than (1.9), which writes

$$\begin{aligned} h_0 &= Ax, \\ h_{k+1} &= h_k + \frac{1}{L^\beta} V_{k+1} \sigma(W_{k+1} h_k), \quad 0 \leqslant k \leqslant L-1, \\ F(x) &= B h_L, \end{aligned} \tag{1.12}$$

where we added a scaling factor $1/L^\beta$ with $\beta > 0$, additional weight matrices $V_k \in \mathbb{R}^{q \times q}$ for $k = 1, \dots, L$, and we recall that, as in (1.9), $A \in \mathbb{R}^{q \times d}$, $B \in \mathbb{R}^{d' \times q}$, $W_k \in \mathbb{R}^{q \times q}$ for $k = 1, \dots, L$, and σ is an activation function. The presence of a scaling factor differs from the residual network models presented above. This matter is thoroughly discussed in Chapter 2; in a nutshell, the scaling factor is necessary for the model to be well-posed in the absence of other normalization techniques such as batch normalization. As for the additional weight matrices V_k, they make the formulation of the model actually closer to the original residual networks of He et al. (2016a), and they are required for some results below.

Looking at the discrete recursion (1.12), a natural idea is to substitute it with a continuous counterpart

$$\frac{dH}{ds} = V \sigma(WH), \quad s \in [0, 1]. \tag{1.13}$$

In other words, the discrete layer index k is transformed into a continuous layer index s and the discrete updates are replaced by an ordinary differential equation. This idea has been popularized by the seminal paper of Chen et al. (2018a), which coined the terminology of neural ordinary differential equations (neural ODEs). Subsequently, numerous works have relied on the intuition that the neural ODE (1.13) is the limit of the residual network (1.12) in the large-depth limit $L \to \infty$ (see, e.g., Haber and Ruthotto, 2017; E et al., 2019; Dong et al., 2020; Massaroli et al., 2020; Kidger, 2022), although the picture is more complex as we will see in Section 1.2.1.

Remark 1.2. *Equation* (1.13) *is ambiguous in that it does not specify whether* V *and* W *depend on* s*. The answer depends on the context, which is why we leave it ambiguous for now and make more precise statements in the following. More precisely, neural ODEs were first introduced by Chen et al. (2018a) with constant (i.e., depth-independent) parameters. However, since the parameters of* (1.12) *depend on the layer index, it seems clear, and will be made formal in the following, that the continuous version of* (1.12) *has depth-dependent parameters* $V(s)$ *and* $W(s)$*. Finally, we also study in Chapter 5 a continuous version of recurrent neural networks, in which case the parameters of the limit ODE are depth-independent (see Section 1.2.4).*

The continuous-depth limit (1.13) appeals both from an algorithmic and a theoretical point of view. Let us delve more into the details of both aspects.

Algorithmic advantages of the large-depth limit. The continuous-depth viewpoint on deep learning has received a lot of attention from an algorithmic and practical point of view. We review a few relevant works here, although we emphasize that this manuscript focuses mostly on theoretical analysis. For this reason, the reader interested in algorithmic developments is encouraged to investigate the PhD thesis of Kidger (2022), which contains a comprehensive overview of models and algorithms related to neural ODEs. This being said, an important motivation for continuous-depth models is that they offer memory efficient training by using the adjoint method to retrieve gradients, which removes the need to store the value of hidden layers (Chen et al., 2018a). This property was later extended to finite-depth residual networks (Sander et al., 2022b). Another line of work has used the continuous viewpoint to design new architectures that discretize more efficiently continuous-depth differential equations, and thereby benefit from favorable properties of their continuous-depth equivalent, such as stability (Haber and Ruthotto, 2017; Chang et al., 2019; Benning et al., 2019). Novel optimization algorithms coming from optimal control have also been proposed by casting the optimization problem over continuous-depth networks as an optimal control problem (Li et al., 2017). The continuous viewpoint was also used to design efficient generative models (Chen et al., 2018a; Grathwohl et al., 2019; Kidger et al., 2021), which sample noise then continuously transforms it into the target distribution. In particular, the recently acclaimed diffusion models can be seen as instances of neural ODEs (Song et al., 2021). Finally, in a time series context, continuous-depth networks have been praised for natively handling irregularly sampled data (Rubanova et al., 2019; Kidger et al., 2020), contrarily to standard models.

Theoretical advantages of the large-depth limit. From a theoretical point of view, the large-depth limit is conceptually simple and provides a very natural interpretation of depth as the time flow of a differential equation. Furthermore, it allows to leverage the well-established tools of differential equations in a deep learning context. This prompted interest into deriving mathematical properties of deep learning models by leveraging the continuous viewpoint, as proposed by E (2017). For instance, E et al. (2019) leverages the optimal control theory to obtain statistical guarantees on infinite-depth residual networks, by formulating a Pontryagin's maximum principle version of the learning problem. Cuchiero et al. (2020) also uses tools from optimal control (Lie brackets and controllability) to show that infinite-depth residual networks are able to interpolate arbitrary training sets with a limited number of parameters.

However, the precise derivation of the connection between (1.12) and (1.13), and the exploration of the consequences of the continuous viewpoint on residual networks remain not yet fully investigated. Our goal in this first part of the thesis is to provide mathematical statements on these two challenges. We will answer the following four key questions: first, under what conditions and in which sense does the neural ODE limit hold for deep neural networks, both at initialization and after training? Second, are there other possible deep limits than neural ODEs? Third, can global convergence of the training algorithm be proven in this setting? Fourth, what does this ODE-like regime imply in terms of the generalization abilities of deep neural networks?

Before delving into the core of our contributions, let us note that the setting and precise assumptions vary slightly between the different chapters. We give here a unified and simplified presentation, and leave rigorous statements, related work, as well as generalizations of the results sketched below, to the core of the manuscript. In particular, rather than providing a literature review on the theory of large-depth residual networks at this point, we refer the reader to the related work sections of Chapters 2 to 5 for an in-depth survey of the literature relevant to the

topics addressed in each chapter.

1.2.1 Scaling of residual networks at initialization [Chapter 2 of the manuscript]

We begin by investigating in Chapter 2 whether the scaled residual network (1.12) converges at initialization towards a differential equation in the large-depth limit, depending on the initialization scheme and on the scaling factor β. Several possible initialization schemes are investigated, the two main ones being i.i.d. initialization and smooth initialization. I.i.d. initialization corresponds to the standard practice of initializing every weight as an i.i.d. random variable, e.g., uniform or Gaussian. Smooth initialization is less common and amounts to taking the V_k and W_k as discretizations of some smooth (possibly random) functions $\mathcal{V} : [0,1] \to \mathbb{R}^{q\times q}$ and $\mathcal{W} : [0,1] \to \mathbb{R}^{q\times q}$, that is, $V_k = \mathcal{V}(k/L)$ and $W_k = \mathcal{W}(k/L)$ for $k \in \{1,\dots,L\}$. This includes in particular the case where the weights are initialized weight-tied across the depth, i.e., where $\mathcal{V}$ and $\mathcal{W}$ are constant functions. Another more complex case is where the entries of $\mathcal{V}$ and $\mathcal{W}$ are independent Gaussian processes with expectation zero and squared exponential covariance.

Our main contribution in this chapter is to show that the large-depth properties of the network at initialization depend on the joint choice of β and of the initialization scheme, as reported in Table 1.3. Both in the i.i.d. case and in the smooth case, a specific value of β corresponds to a differential equation limit that separates two antithetical dynamics, explosion and identity. Furthermore, the differential equation limit is a stochastic differential equation (SDE) in the i.i.d. case and an ODE in the smooth case.

Scaling factor	$0 < \beta < 1/2$	$\beta = 1/2$	$1/2 < \beta < 1$	$\beta = 1$	$\beta > 1$
I.i.d. initialization	Explosion	SDE limit	Identity	Identity	Identity
Smooth initialization	Explosion	Explosion	Explosion	ODE limit	Identity

Table 1.3: Properties of the residual network at initialization as a function of the scaling factor and of the initialization. Explosion means that the output of the network tends to infinity when the depth L goes to infinity. Identity corresponds to the fact that $h_L \approx h_0$ when L goes to infinity.

To make things more precise, the first row of Table 1.3 can be formalized by the following result, which is a simplified version of Corollary 2.4 in Chapter 2.

Theorem 1.3. *Consider the residual network* (1.12) *with i.i.d. weights.*

(i) If $\beta > 1/2$, then

$$\frac{\|h_L - h_0\|}{\|h_0\|} \xrightarrow[L\to\infty]{\mathbb{P}} 0.$$

(ii) If $\beta < 1/2$, then

$$\frac{\|h_L - h_0\|}{\|h_0\|} \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$$

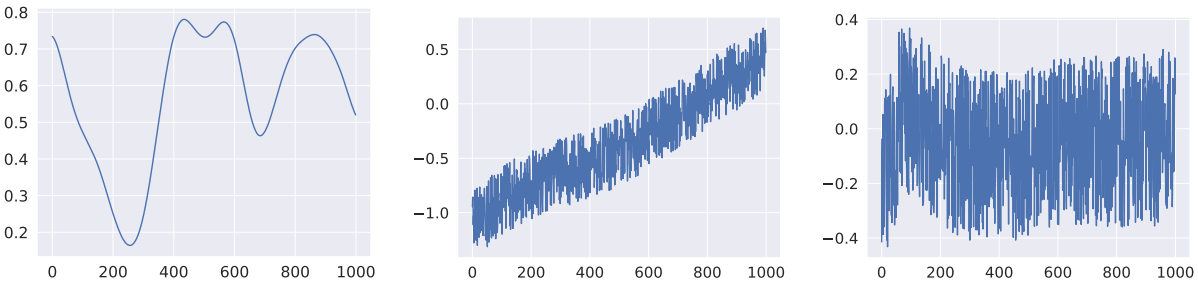
(iii) If $\beta = 1/2$, then, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\exp\left(\frac{3}{8} - \sqrt{\frac{22}{q\delta}}\right) - 1 < \frac{\|h_L - h_0\|^2}{\|h_0\|^2} < \exp\left(1 + \sqrt{\frac{10}{q\delta}}\right) + 1.$$

The proof of this result relies crucially on the martingale structure of $(\|h_k\|)_{0\leqslant k\leqslant L}$, as well as on state-of-the-art concentration inequalities for random matrices with sub-Gaussian entries.

In summary, among the possibilities we examine, the only case where the ODE limit (1.13) holds is when initializing with smooth weights and taking a scaling factor $\beta = 1$ (corresponding to the framed cell in Table 1.3). This is the case which we consider in the following chapters.

Furthermore, we also perform some preliminary experiments showing that this ODE-like weight structure is preserved after training, as shown in Figure 1.1. This is the topic of the next chapter.



(a) $\beta = 1$, smooth initialization (b) $\beta = 1$, i.i.d. initialization (c) $\beta = 1/2$, i.i.d. initialization

Figure 1.1: Plot of a given coordinate of V_k, after training, as a function of the layer index k ranging from 1 to the depth $L = 1000$ for three different choices of β and initializations. The left plot depicts a smooth ODE-like weight structure, contrarily to the other plots.

1.2.2 Implicit regularization of deep residual networks towards neural ODEs [Chapter 3 of the manuscript]

In Chapter 3, we also consider model (1.12), this time focusing on the case of a smooth initialization and a $1/L$ scaling factor. According to the previous chapter, we know that, at initialization, the neural network converges towards a neural ODE. We are interested in showing that the *trained* neural network still exhibits an ODE-like weight structure, or in other words, to shed light on Figure 1.1a. To this aim, we assume that the neural network is trained with a gradient flow that writes as the following ODEs

$$\frac{\partial V_k}{\partial t}(t) = -L\frac{\partial \hat{\mathcal{R}}_n}{\partial V_k}(t), \quad \frac{\partial W_k}{\partial t}(t) = -L\frac{\partial \hat{\mathcal{R}}_n}{\partial W_k}(t), \quad t \geqslant 0, \tag{1.14}$$

where $\hat{\mathcal{R}}_n$ denotes the empirical risk (as defined in (1.3)). Gradient flow can be seen a simplifying limit of SGD (see Algorithm 1) when the learning rate γ tends to zero. It is a standard tool in the analysis of optimization algorithms. We refer to Bach (2020) for a user-friendly introduction to gradient flows in a machine learning context. The scaling factor L in (1.14) is the counterpart of the scaling factor $1/L$ in (1.12), and is necessary to avoid gradient vanishing as L goes to infinity. Note that the time variable t of the ODE differs from the continuous-depth variable s in the neural ODE (1.13). Our first main contribution in this chapter is to prove that the neural ODE limit holds after training, as shown in the following theorem (Theorem 3.4 of Chapter 3), stated informally here.

Theorem 1.4. *Consider the neural network* (1.12) *with* $\beta = 1$. *Assume that it is initialized smoothly in the sense of Section 1.2.1, and trained with the gradient flow* (1.14) *for* $t \in [0, T]$.

Then there exist Lipschitz-continuous functions $\mathcal{V}, \mathcal{W} : [0,1] \times [0,T]$ *such that, uniformly over* $s \in [0,1]$ *and* $t \in [0,T]$, $V_{\lfloor sL \rfloor}(t) \to \mathcal{V}(s,t)$ *and* $W_{\lfloor sL \rfloor}(t) \to \mathcal{W}(s,t)$ *when the depth* L *goes to infinity.*

Furthermore, for any $k \in \{0, \dots L\}$, denote $h_k(t)$ the value of the layer k at training time t. Then, for any $s \in [0,1]$, the layer $h_{\lfloor sL \rfloor}(t)$ converges when L goes to infinity to the value at time s of the solution of the neural ODE

$$\begin{aligned} H(0) &= x \\ \frac{dH}{ds}(s) &= \mathcal{V}(s,t)\sigma(\mathcal{W}(s,t)H(s)), \quad s \in [0,1]. \end{aligned}$$

This convergence is uniform over $s \in [0,1]$, $t \in [0,T]$ and $x \in K$ for any compact $K \subset \mathbb{R}^d$.

This large-depth convergence holds for any finite training time $t \in [0,T]$. However, the convergence of the optimization algorithm when T goes to infinity is not guaranteed without further assumptions, due to the non-convexity of the optimization problem. We can obtain such a convergence by proving a Polyak-Łojasiewicz (PL) condition, which is a key modern tool in analyzing the properties of optimization algorithms for deep neural networks (Liu et al., 2022). Importantly, the PL condition implies the convergence of gradient flow to a global minimum. As our second main contribution in this chapter, we prove that such a condition holds when the width q of the hidden layers is larger than some constant times the number of data n. As a consequence, we obtain the convergence with high probability in large depth and large training time, namely the existence of Lipschitz-continuous functions $\mathcal{V}_\infty$ and $\mathcal{W}_\infty$ such that the trained neural network converges when *both* L and T go to infinity to the ODE

$$\frac{dH}{ds}(s) = \mathcal{V}_\infty(s)\sigma(\mathcal{W}_\infty(s)H(s)), \quad s \in [0,1]. \tag{1.15}$$

The convergence holds in a sense similar to the one of Theorem 1.4. Furthermore, the limit ODE achieves zero training loss. This analysis is a first step towards understanding the implicit regularization (see Section 1.1.2.2) of gradient flow for deep residual networks, that is, characterizing the properties of the trained network among all minimizers of the empirical risk.

1.2.3 Generalization bounds for neural ODEs and residual networks [Chapter 4 of the manuscript]

Now that we know that trained deep residual networks converge in the large-depth limit towards neural ODEs of the form (1.15), we are interested in understanding the statistical properties of this class of models. This is our aim in Chapter 4. We examine the simpler case where the inner weights $\mathcal{W}_\infty$ are equal to the identity matrix, or in other words, we only consider outer weights $\mathcal{V}_\infty$. Even with this simplification, this remains a delicate problem since the output of the model is a non-linear function of the parameters, and the latter belong to an infinite-dimensional space. Nevertheless, we are able to bound their statistical complexity by leveraging results on the covering number of bounded Lipschitz-continuous functions. More precisely, for a matrix-valued function V, denote $\|V\|_{1,1,\infty} = \max_{s\in[0,1]} \sum_{i,j} |V_{ij}(s)|$. Then we consider the set of neural ODEs

$$\begin{aligned} H(0) &= x \\ \frac{dH}{ds}(s) &= V(s)\sigma(H(s))ds, \quad s \in [0,1] \\ F_V(x) &= H(1), \end{aligned} \tag{1.16}$$

where V belongs to the set of functions

$$\begin{aligned} \mathscr{V} = \{V : [0,1] \to \mathbb{R}^{q\times q}, \|V\|_{1,1,\infty} \leqslant R_{\mathscr{V}} \text{ and} \\ V_{ij} \text{ is } K_{\mathscr{V}}\text{-Lipschitz-continuous for } i,j \in \{1,\dots,q\}\}. \end{aligned} \tag{1.17}$$

It is a simplification of the large-depth limit model (1.15) obtained in the previous chapter, in the sense that we removed the initial and final projections matrices A and B, as well as the inner weights $W(s)$. We place ourselves in a standard supervised learning setup with n i.i.d. data samples, which we assume to be almost surely bounded. We consider some general Lipschitz-continuous loss function. The empirical risk is denoted by $\hat{\mathcal{R}}_n$, the theoretical risk by $\mathcal{R}$, and the empirical risk minimizer over $\mathscr{V}$ by $\hat{V}_n$. The following result can then be proven as a straightforward corollary of Theorem 4.4 of Chapter 4.

Theorem 1.5. *Consider the class of neural ODEs* (1.16)*, where V belongs to* (1.17)*. Then, there exists a constant $B > 0$ depending on the parameters of the problem such that for n large enough, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\mathscr{R}(\hat{V}_n) \leqslant \hat{\mathscr{R}}_n(\hat{V}_n) + B(q+1)\sqrt{\frac{\log(R_{\mathscr{V}} mn)}{n}} + B\frac{q^2\sqrt{K_{\mathscr{V}}}}{n^{1/4}} + \frac{B}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}.$$

This result belongs to the family of generalization bounds, which aim at bounding the difference between the theoretical risk and the empirical risk of $\hat{V}_n$. It is a consequence of a uniform law of large numbers as presented in Section 1.1.2.1. Three terms appear in our upper bound of $\mathscr{R}(\hat{V}_n) - \hat{\mathscr{R}}_n(\hat{V}_n)$. The first and the third ones are classical (see, e.g. Bach, 2023, Sections 4.4 and 4.5). On the contrary, the second term is more surprising with its convergence rate in $\mathcal{O}(n^{-1/4})$. This slower convergence rate is due to the fact that the space of parameters $\mathscr{V}$ is infinite-dimensional. However, we retrieve a classical $\mathcal{O}(n^{-1/2})$ convergence rate in the case where $K_{\mathscr{V}} = 0$. This corresponds to constant functions V, i.e., depth-independent weights, which belong to a finite-dimensional space.

As a second main contribution in this chapter, we prove similar generalization bounds for finite-depth residual networks of the form (1.12). Although these bounds hold for finite-depth networks, their derivation is strongly inspired by the infinite-depth analysis.

1.2.4 Recurrent neural networks as kernel methods [Chapter 5 of the manuscript]

In Chapter 5, we consider a slightly different model, namely residual *recurrent* networks. Similarly to the previous chapter, we introduce a scaling factor $1/L$ and thus consider the rescaled version of the residual recurrent network (1.11), which writes as follows

$$h_{k+1} = h_k + \frac{1}{L}\sigma(Uh_k + Vx_{k+1}). \tag{1.18}$$

We further assume that each input series $(x_k)_{1\leqslant k\leqslant L}$ corresponds to the discretization of a continuous time process $X : [0,1] \to \mathbb{R}^d$. Our goal is to show that the rescaled residual recurrent network is actually a kernel method, yielding as a byproduct generalization and stability bounds. Since we recognize in (1.18) an Euler discretization of an ODE, we first show (in a precise sense) that the network approaches at distance $\mathcal{O}(1/L)$ the continuous-time network

$$\frac{dH}{ds}(s) = \sigma(UH(s) + VX(s)), \quad H(0) = 0, \tag{1.19}$$

where the output of the network is a linear projection of the value of the solution of the ODE at time $t = 1$, that is, $y = BH(1)$. Then, our main theorem in this chapter states the existence of a Hilbert space $\mathcal{H}$ and of mappings $\mathcal{S}$ and ς such that, under some regularity assumptions,

$$y = \langle \mathcal{S}(X), \varsigma(B,U,V)\rangle_{\mathcal{H}}. \tag{1.20}$$

In this equation, $\mathcal{S}(X)$ is the signature of the time series X, which is a feature map for time series (Levin et al., 2013), while $\varsigma(B,U,V)$ is an expression involving powers of B, U, V and

higher-order derivatives of σ. The key insight of the proof is to use a specific variant of the Taylor expansion that applies for ODEs involving a time-dependent parameter X, which are known as controlled differential equations (Lyons et al., 2007).

This result allows to reinterpret the action of the recurrent network as a scalar product in an (infinite-dimensional) Hilbert space, thereby framing the recurrent network as a kernel method. Hence we can use the usual kernel machinery to derive generalization bounds and stability bounds. Let us illustrate the second idea, whose derivation is easier: for two time series X and X', we can bound the difference between the corresponding outputs y and y' by using the Cauchy-Schwartz inequality, as follows

$$|y - y'| = |\langle \mathcal{S}(X), \varsigma(B,U,V)\rangle_{\mathcal{H}} - \langle \mathcal{S}(X'), \varsigma(B,U,V)\rangle_{\mathcal{H}}| \leq \|\mathcal{S}(X) - \mathcal{S}(X')\|_{\mathcal{H}} \|\varsigma(B,U,V)\|_{\mathcal{H}}. \tag{1.21}$$

Since the mapping $\mathcal{S}$ is continuous, $\|\mathcal{S}(X) - \mathcal{S}(X')\|_{\mathcal{H}}$ is small if X and X' are sufficiently close. Hence this result shows that the Hilbert norm of the weights mapping $\|\varsigma(B,U,V)\|_{\mathcal{H}}$ controls the stability of the network. This suggests using this quantity as a regularizer when training the network.

1.3 Contributions to finite-depth neural networks [Part II of the manuscript]

This second part of the manuscript gathers various contributions related to modern topics in deep learning, this time for finite-depth neural networks, contrarily to the first part of the manuscript which focused on large-depth limits. We first present results on shallow neural networks, then on Transformer.

1.3.1 Convergence of shallow neural networks in the two-timescale regime [Chapter 6 of the manuscript]

In Chapter 6, we study the training dynamics of shallow neural networks of the form (1.2), in a two-timescale regime in which the step sizes for the inner layer are much smaller than those for the outer layer. We consider a simple univariate setting where the neural network writes

$$f_{v,b}(x) = \sum_{k=1}^{q} v_k \sigma(x - b_k),$$

for an input $x \in \mathbb{R}$ and parameters $v_k, b_k \in \mathbb{R}$. Comparing with the general formulation of shallow neural networks (1.2), we see that we fix the multiplicative inner weights w_k to 1. Similarly to Chapter 3 (see equation (1.14)), we assume that the neural network is trained by gradient flow, this time directly on the theoretical risk

$$\mathcal{R}(v,b) = \mathbb{E}(|f_{v,b}(X) - f^*(X)|^2),$$

where X follows a uniform law on $[0,1]$. Minimizing directly the theoretical risk instead of the empirical risk allows us to set aside statistical issues to focus on optimization, which remains a delicate matter since the risk is non-convex in b. The gradient flow equations write

$$\frac{\partial v}{\partial t}(t) = -\frac{\partial \mathcal{R}}{\partial v}(t), \quad \frac{\partial b}{\partial t}(t) = -\varepsilon \frac{\partial \mathcal{R}}{\partial b}(t), \quad t \geqslant 0, \tag{1.22}$$

where ε parameterizes the ratio between the step sizes for the inner layer and for the outer layer. By taking $\varepsilon \ll 1$, the neural network can be thought of as a fitted linear regression with slowly

evolving features $\sigma(\cdot - b_k)$, $k = 1, \dots, q$. This reduction enables us to precisely describe the movement of the inner layer parameters b_k. In this two-timescale regime, we prove convergence of the gradient flow to a global optimum in the case where f^* is a piecewise constant function. More precisely, we show the following theorem (Theorem 6.2 of Chapter 6), stated in a simplified manner here.

Theorem 1.6. *Let $\xi, \delta > 0$, and f^* a piecewise constant function with N pieces whose sizes are all lower-bounded by Δv. Assume that the neural network has q neurons with*

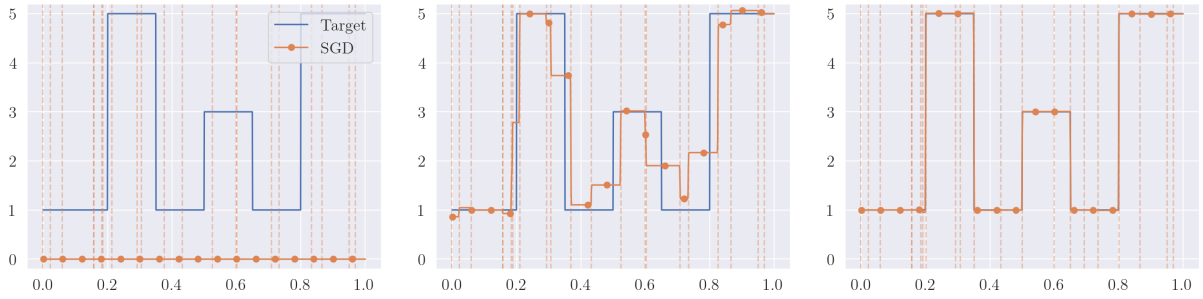
$$q \geqslant \frac{6}{\Delta v}\big(4 + \log N + \log \frac{1}{\delta}\big)\,. \tag{1.23}$$

Assume that, at initialization, the biases $b_1, \dots, b_q$ are i.i.d. uniformly distributed on $[0, 1]$ and the weights v are equal to zero. Then there exist an activation function σ, $\varepsilon > 0$, and $T > 0$, such that, with probability at least $1 - \delta$, the solution to the gradient flow (1.22) is defined at least until T, and

$$\mathcal{R}(v(T), b(T)) \leqslant \xi\,.$$

Note that the lower bound on the number of neurons (1.23) does not depend on the target precision ξ, but only on the function f^* and on the probability of failure δ. This distinguishes our result from the neural tangent kernel or mean-field regimes, mentioned in Section 1.1.2, which both require the width of the network to grow large in order to achieve an arbitrary precision.

Experimental illustration is provided, showing that the stochastic gradient descent behaves according to our description of the gradient flow and thus converges to a global optimum in the two-timescale regime (see Figure 1.2), but can fail outside this regime (see Figure 1.3).

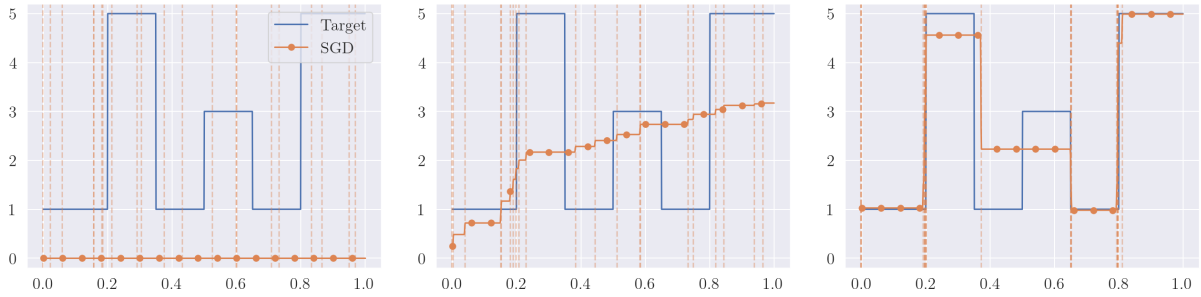


(a) At initialization (b) After $p = 5.4{\cdot}10^6$ steps of SGD (c) After $p = 1.8{\cdot}10^8$ steps of SGD

Figure 1.2: Simulation in the two-timescale regime ($\varepsilon = 2 \cdot 10^{-5}$). The target function is in blue and the neural network is in orange. The biases $b_1, \dots, b_q$ of the neurons are indicated with vertical dotted lines (the dots are only present for black and white visibility). In a first short phase, only the weights $v_1, \dots, v_q$ of the neurons evolve to match as well as possible the target function (second plot). Then, in a longer phase, the neuron closest to each target discontinuity moves towards it (third plot). Recovery is achieved.

1.3.2 Structured context and high-coverage grammar for conversational question answering over knowledge graphs [Chapter 7 of the manuscript]

We present in Chapter 7 results on the Transformer architecture, turning our attention to more algorithmic matters related to natural language processing. Since this topic may be less familiar to the reader than the rest of the thesis, we present the context and our contribution in a somewhat more detailed fashion than the other chapters. Our goal is to design a Transformer-like



(a) At initialization $p = 0$ (b) After $p = 10^4$ steps of SGD (c) After $p = 10^6$ steps of SGD

Figure 1.3: Simulation outside the two-timescale regime ($\varepsilon = 1$). The target function is in blue and the neural network is in orange. The biases $b_1, \ldots, b_q$ of the neurons are indicated with vertical dotted lines (the dots are only present for black and white visibility). The dynamics create a zone with no neuron, hindering recovery.

neural network for the task of conversational question answering over knowledge graphs. Let us first describe the task, before presenting our approach, contributions and results.

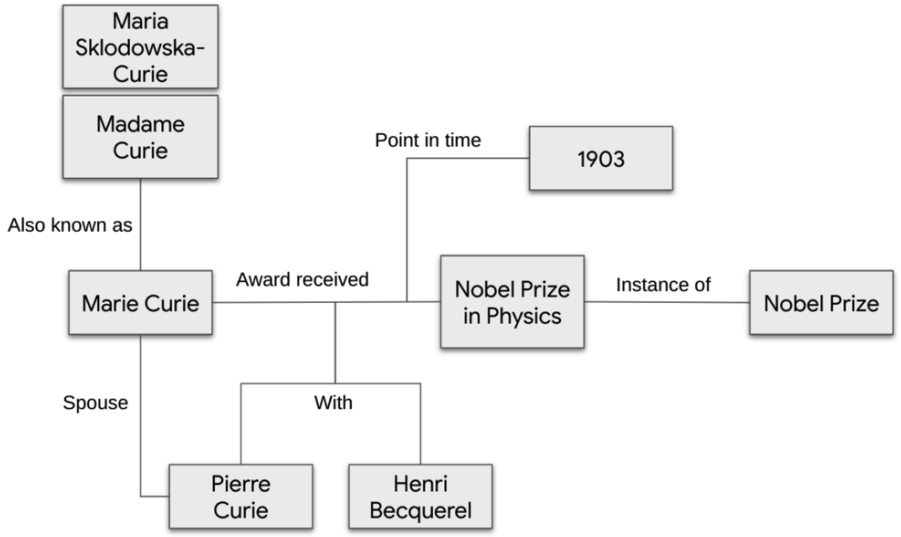


Figure 1.4: A small subgraph of Wikidata

To begin with, a *knowledge graph* (KG) is a particular type of graphs that encodes real-world factual information. It is a directed graph consisting of vertices, called *entities*, which represent concepts, and labeled edges, which represent relations between concepts. For instance, Figure 1.4 presents a small subgraph of Wikidata, which is the largest publicly available KG and which we work with in this chapter. Question answering over KGs refers to the task of building a language model that can answer factual questions by querying the graph. Furthermore, we want our system to handle conversations, that is, not only be able to answer a one-off question but also follow-up questions on the same topic. This task is referred to as conversational question answering over KGs. An example of such a conversation is presented in Table 1.4. It is taken from ConvQuestions (Christmann et al., 2019), which, together with CSQA (Saha et al., 2018), is one of the main publicly available datasets for this task. Note that conversational question answering over KGs is of direct interest for developing personal assistants that can reliably answer factual-based questions. One of the difficulties associated with this task is the scale of the graph:

for instance, there are over a hundred million nodes and a billion edges in Wikidata, and private KGs can be orders of magnitude larger.

Q	Who played the joker in The Dark Knight?
A	Heath Ledger
Q	When did he die?
A	22 January 2008
Q	Batman actor?
A	Christian Bale
Q	Director?
A	Christopher Nolan
Q	Sequel name?
A	The Dark Knight Rises

Table 1.4: Example of a conversation in ConvQuestions (Christmann et al., 2019). The goal of the system is to find the right answer at each turn of the conversation.

We address this task by building a so-called *semantic parsing* model. A semantic parser is a language model that takes as input a sentence phrased in natural language (typically English) and transforms it into a query formulated in some programming language, which can then be executed, or *evaluated*, to return an answer to the question. From a high-level perspective, a semantic parser can be seen as translating English into a machine-executable language. Let us introduce some vocabulary that is heavily used in Chapter 7: the query is called a *logical form*, and the programming language in which it is expressed is called a *grammar*.

Our semantic parser works in several steps depicted in Figure 1.5. Given a question, entities in the graph that are likely to be relevant are identified using a technique called *named entity linking*. Then local exploration of the graph around these entities is performed to extract a context of tractable size. The context is organized in a tree structure, where each node of the tree contains a vector of information. Technically, this tree takes the form of a JSON file. In a third step, a Transformer-like neural network takes this context tree as input and returns the logical form.

Our work features two main methodological contributions: first, we build a grammar that can model a larger scope of questions over knowledge graphs than previous propositions. Second, we introduce a variant of the Transformer architecture that can operate on tree-structured data, contrarily to the standard version, which can only process a list of vectors. One of the difficulties we face is that we do not have at our disposal readily-available supervised data to train our Transformer model, since there exists no dataset mapping English questions to logical forms in our grammar. To circumvent this issue, we first have to craft such a dataset, before training our model. This setting is referred to as *weak supervision*. The dataset is created in a compute-intensive manner: for every (question, answer) pair in CSQA and ConvQuestions, we explore the space of all possible logical forms to find one whose execution gives the right answer, and which has reasonable chances to be a correct formalization of the question. Heuristic metrics are used to assess the accordance between the question and the logical form. The space of logical forms can be seen as a tree, which we explore with breadth-first search in order to favor the simplest possible logical forms.

The capabilities of our system are evaluated on the two datasets mentioned above. Our approach improves over the state-of-the-art on both datasets in terms of answer accuracy.

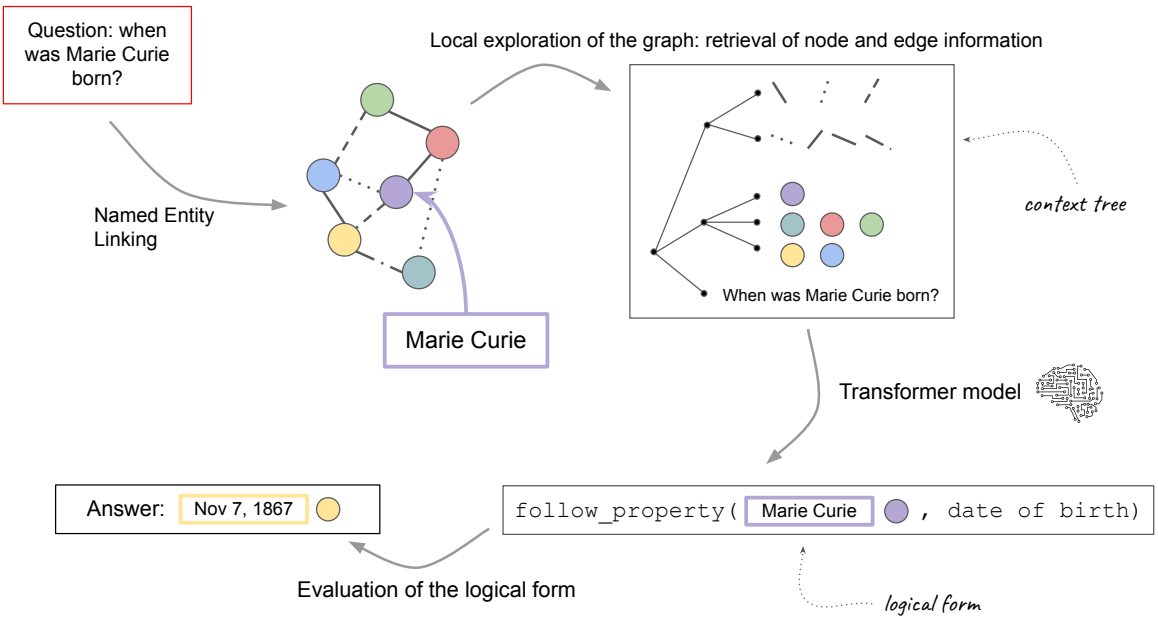


Figure 1.5: Data pipeline in our model. The question is first processed by named entity linking in order to find entities in the graph that are contained in the question. In a second step, we explore the graph locally around the found entities to construct a context tree, which forms the input to the Transformer model. It outputs a logical form which is evaluated to produce a candidate answer.

1.4 Résumé détaillé en français

Cette thèse présente des contributions à la théorie des réseaux de neurones, séparées en deux parties. La première partie s'intéresse à l'analyse mathématique des réseaux résiduels dans la limite en grande profondeur, et comporte quatre chapitres. La deuxième partie présente des contributions reliées cette fois aux réseaux de profondeur finie, et regroupe deux chapitres. Nous introduisons ici les chapitres dans l'ordre du manuscrit. La présentation est moins détaillée que celle qui précède en anglais, aussi nous recommandons au locuteur anglophone de se référer à la version anglaise.

1.4.1 Échelle des réseaux résiduels à l'initialisation [Chapitre 2 du manuscrit]

Nous commençons dans le Chapitre 2 par nous intéresser aux propriétés des réseaux de neurones résiduels qui s'écrivent

$$\begin{aligned} h_0 &= Ax, \\ h_{k+1} &= h_k + \frac{1}{L^\beta} V_{k+1} \sigma(W_{k+1} h_k), \quad 0 \leqslant k \leqslant L-1, \\ F(x) &= B h_L, \end{aligned}$$

où la donnée est $x \in \mathbb{R}^d$, la matrice A appartient à $R^{q \times d}$, les états cachés h_k sont dans $\mathbb{R}^q$, les matrices V_{k+1}, W_{k+1} appartiennent à $R^{q \times q}$, et $B \in \mathbb{R}^{d' \times q}$. L'objectif de ce chapitre est de déterminer sous quelles conditions ce réseau converge vers une équation différentielle ordinaire (EDO) ou stochastique (EDS) dans la limite en grande profondeur $L \gg 1$, en fonction du schéma

d'initialisation et du paramètre d'échelle $\beta > 0$. Plusieurs schémas d'initialisation sont étudiés, principalement l'initialisation i.i.d. et l'initialisation régulière. L'initialisation i.i.d. correspond à la pratique usuelle d'initialiser les poids comme des variables aléatoires i.i.d., par exemple uniformes ou Gaussiennes. L'initialisation régulière est moins courante, et correspond à prendre les V_k et W_k comme des discrétisations de fonctions régulières (potentiellement aléatoires) $\mathcal{V} : [0,1] \to \mathbb{R}^{q\times q}$ et $\mathcal{W} : [0,1] \to \mathbb{R}^{q\times q}$, soit $V_k = \mathcal{V}(k/L)$ and $W_k = \mathcal{W}(k/L)$ pour $k \in \{1,\dots,L\}$.

Notre contribution principale dans ce chapitre est de montrer que la limite en grande profondeur du réseau à l'initialisation dépend conjointement de β et du schéma d'initialisation, comme présenté dans le Tableau 1.5.

Facteur d'échelle	$0 < \beta < 1/2$	$\beta = 1/2$	$1/2 < \beta < 1$	$\beta = 1$	$\beta > 1$
Initialisation i.i.d.	Explosion	Limite EDS	Identité	Identité	Identité
Initialisation régulière	Explosion	Explosion	Explosion	Limite EDO	Identité

Table 1.5: Propriétés du réseau résiduel à l'initialisation en fonction du facteur d'échelle et du schéma d'initialisation. L'explosion signifie que la sortie du réseau diverge vers l'infini quand la profondeur L tend vers l'infini. L'identité correspond au fait que $h_L \approx h_0$ quand L tend vers l'infini.

En résumé, parmi tous les cas examinés, la limite est une EDO seulement dans le cas d'une initialisation régulière et d'un facteur d'échelle $\beta = 1$. L'EDO limite s'écrit alors

$$\begin{aligned} H(0) &= Ax, \\ \frac{dH}{ds}(s) &= \mathcal{V}(s)\sigma(\mathcal{W}(s)H(s)), \quad s \in [0,1], \\ F(x) &= BH(1). \end{aligned} \tag{1.24}$$

C'est le cas auquel on s'intéresse dans les deux chapitres qui suivent.

1.4.2 Régularisation implicite des réseaux de neurones résiduels vers des EDO neuronales [Chapitre 3 du manuscrit]

Dans le Chapitre 3, nous nous intéressons au même modèle que dans le chapitre précédent, en se concentrant cette fois sur le cas d'une initialisation régulière et d'un facteur d'échelle $1/L$. Le chapitre précédent montre que dans ce cas, à l'initialisation, le réseau converge vers une EDO. Dans ce chapitre, nous montrons que les poids du réseau *entraîné* présentent toujours une structure de type EDO. À cette fin, nous faisons l'hypothèse que le réseau est entraîné par flot de gradient, selon les équations d'évolution

$$\frac{\partial V_k}{\partial t}(t) = -L\frac{\partial \hat{\mathcal{R}}_n}{\partial V_k}(t), \quad \frac{\partial W_k}{\partial t}(t) = -L\frac{\partial \hat{\mathcal{R}}_n}{\partial W_k}(t), \quad t \geqslant 0,$$

où $\hat{\mathcal{R}}_n$ désigne un risque empirique. Notons en particulier que la variable temporelle t de l'EDO qui décrit l'évolution des poids n'est pas la même que la variable s de l'EDO neuronale (1.24) qui décrit la limite en large profondeur.

Notre première contribution dans ce chapitre est de prouver que la convergence (lorsque L tend vers l'infini) du réseau résiduel vers une EDO neuronale est également valide après entraînement. Cette convergence est valide pour tout temps d'entraînement fini $t \in [0,T]$.

Néanmoins, la convergence de l'algorithme d'optimisation lorsque T tend vers l'infini n'est pas garantie sans hypothèse supplémentaire, du fait de la non-convexité du problème d'optimisation.

Nous prouvons cette convergence grâce à une condition de type Polyak-Łojasiewicz (PL), un outil majeur dans l'analyse des algorithmes d'optimisation pour les réseaux de neurones (Liu et al., 2022). La condition PL implique la convergence du flot de gradient vers un minimum global. Notre seconde contribution dans ce chapitre est de prouver que cette condition est vérifiée lorsque la largeur q des couches cachées est plus grande qu'une constante fois la taille de l'échantillon n. Nous obtenons par conséquent la convergence en grande profondeur et en grand temps d'entraînement, c'est-à-dire l'existence de fonctions Lipschitz $\mathcal{V}_\infty$ et $\mathcal{W}_\infty$ telles que le réseau de neurones entraîné converge lorsque L *et* T tendent vers l'infini vers l'EDO

$$\frac{dH}{ds}(s) = \mathcal{V}_\infty(s)\sigma(\mathcal{W}_\infty(s)H(s)), \quad s \in [0,1]. \tag{1.25}$$

De plus, l'erreur d'entraînement de l'EDO limite est égale à zéro. Cette analyse représente une première étape dans la compréhension de la régularisation implicite du flot de gradient pour les réseaux résiduels, c'est-à-dire la caractérisation des propriétés du réseau entraîné parmi tous les minimiseurs du risque empirique.

1.4.3 Bornes de généralisation pour EDO neuronales et réseaux de neurones résiduels [Chapitre 4 du manuscrit]

Maintenant que l'on sait que certains réseaux de neurones résiduels entraînés convergent dans la limite en large profondeur vers des EDO neuronales de la forme (1.25), nous nous intéressons dans le Chapitre 4 à comprendre les propriétés statistiques de cette classe.

Nous examinons le cas simplifié où les matrices intérieures $\mathcal{W}_\infty$ sont égales à la matrice identité. Même avec cette simplification, le problème reste délicat comme la sortie du modèle est une fonction non-linéaire des paramètres et que ces derniers appartiennent à un espace de dimension infinie. Nous sommes tout de même en mesure de borner la complexité statistique du modèle en tirant parti de résultats sur les nombres de couverture des fonctions bornées Lipschitz.

Plus précisément, pour une fonction V à valeur matricielle, introduisons la norme matricielle $\|V\|_{1,1,\infty} = \max_{s\in[0,1]} \sum_{i,j} |V_{ij}(s)|$. On s'intéresse alors aux EDO neuronales qui s'écrivent sous la forme

$$\begin{aligned} H(0) &= x \\ \frac{dH}{ds}(s) &= V(s)\sigma(H(s))ds, \quad s \in [0,1] \\ F_V(x) &= H(1), \end{aligned} \tag{1.26}$$

où V appartient à l'ensemble de fonction

$$\mathscr{V} = \{V : [0,1] \to \mathbb{R}^{q\times q},\ \|V\|_{1,1,\infty} \leqslant R_{\mathscr{V}} \text{ et } V_{ij} \text{ est } K_{\mathscr{V}}\text{-Lipschitz pour } i,j \in \{1,\dots,q\}\}. \tag{1.27}$$

C'est une simplification du modèle limite obtenu au chapitre précédent. On se place dans le cadre classique de l'apprentissage supervisé avec un échantillon de n données i.i.d., que l'on suppose presque sûrement bornées. On prend une fonction de perte Lipschitz générale. Notre résultat principal est alors de montrer que l'erreur de généralisation est bornée par

$$\mathcal{O}\Big((q+1)\sqrt{\frac{\log(R_{\mathscr{V}} mn)}{n}} + \frac{q^2\sqrt{K_{\mathscr{V}}}}{n^{1/4}} + \frac{1}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}\Big).$$

Les premier et dernier termes sont classiques (voir par exemple Bach, 2023, Sections 4.4 and 4.5). Le second terme est plus surprenant à cause de son taux de convergence en $\mathcal{O}(n^{-1/4})$. Ce taux plus lent est dû au fait que l'espace de paramètre $\mathscr{V}$ est de dimension infinie.

Notre seconde contribution principale dans ce chapitre est de prouver une borne de généralisation similaire pour des réseaux résiduels de profondeur finie, en utilisant une analyse inspirée par le cas de la profondeur infinie.

1.4.4 Certains réseaux de neurones récurrents sont des méthodes à noyaux [Chapitre 5 du manuscrit]

Dans le Chapitre 5, nous nous intéressons à un modèle légèrement différent, les réseaux de neurones récurrents résiduels. Comme dans le chapitre précédent, nous introduisons un facteur d'échelle $1/L$, et considérons donc le réseau défini par l'itération suivante

$$h_{k+1} = h_k + \frac{1}{L}\sigma(Uh_k + Vx_{k+1}),$$

où $(x_k)_{1\leqslant k\leqslant L}$ est une donnée séquentielle. Nous faisons de plus l'hypothèse que la donnée $(x_k)_{1\leqslant k\leqslant L}$ est la discrétisation d'un processus en temps continu $X : [0,1] \to \mathbb{R}^d$. Notre but dans ce chapitre est de montrer que le réseau de neurones récurrent résiduel s'écrit en fait comme une méthode à noyaux. Nous reconnaissons dans l'équation précédente la discrétisation d'Euler d'une EDO, et prouvons ainsi que le réseau approche à une distance $\mathcal{O}(1/L)$ son équivalent en profondeur continue

$$\frac{dH}{ds}(s) = \sigma(UH(s) + VX(s)), \quad H(0) = 0, \tag{1.28}$$

où maintenant la sortie du réseau s'écrit comme une projection linéaire de la valeur de la solution de l'EDO au temps $t = 1$, soit $y = BH(1)$. Notre théorème principal dans ce chapitre s'écrit alors comme l'existence d'un espace de Hilbert $\mathcal{H}$ et de fonctions $\mathcal{S}$ et ς tels que, sous certaines hypothèses de régularité,

$$y = \langle \mathcal{S}(X), \varsigma(B, U, V)\rangle_{\mathcal{H}}.$$

La fonction $\mathcal{S}$ est de plus une fonction connue dans la littérature, il s'agit de la signature de la série temporelle X (Levin et al., 2013). Ce résultat nous permet de réinterpréter l'action du réseau de neurones récurrent comme un produit scalaire dans un espace de Hilbert de dimension infinie, c'est-à-dire comme une méthode à noyau. Cela nous permet d'utiliser l'outillage habituel des méthodes à noyaux pour en déduire des bornes de généralisation et de stabilité.

1.4.5 Convergence des réseaux de neurones à une couche cachée dans la limite bi-échelle [Chapitre 6 du manuscrit]

Le chapitre 6 débute la seconde partie du manuscrit. Il s'intéresse à la dynamique d'apprentissage des réseaux de neurones à une couche cachée, dans un régime bi-échelle où les pas de gradients pour la couche intérieure sont négligeables devant ceux pour la couche extérieure.

On considère un cas univarié où le réseau de neurones s'écrit

$$f_{v,b}(x) = \sum_{k=1}^{q} v_k \sigma(x - b_k),$$

pour une entrée $x \in \mathbb{R}$ et des paramètres $v_k, b_k \in \mathbb{R}$. De façon analogue au chapitre 3, nous faisons l'hypothèse que le réseau de neurones est entraîné par flot de gradient, cette fois directement sur le risque théorique

$$\mathcal{R}(v, b) = \mathbb{E}(|f_{v,b}(X) - f^*(X)|^2),$$

où X suit une loi uniforme sur $[0, 1]$. Minimiser directement le risque théorique plutôt que le risque empirique nous permet de mettre de côté les problèmes statistiques pour se concentrer sur le problème d'optimisation, qui reste un problème délicat puisque le risque est non convexe en b. Le flot de gradient s'écrit

$$\frac{\partial v}{\partial t}(t) = -\frac{\partial \mathcal{R}}{\partial v}(t), \quad \frac{\partial b}{\partial t}(t) = -\varepsilon\frac{\partial \mathcal{R}}{\partial b}(t), \quad t \geqslant 0, \tag{1.29}$$

où ε paramétrise le ratio entre les pas sur la couche intérieure et ceux sur la couche extérieure. En prenant $\varepsilon \ll 1$, le réseau de neurones peut être vu comme une régression linéaire ajustée, avec des facteurs $\sigma(\cdot - b_k)$, $k = 1, \ldots, q$ qui évoluent lentement. Cette simplification nous permet de décrire précisément la dynamique des paramètres. Ainsi, dans ce régime bi-échelle, nous prouvons la convergence du flot de gradient dans le cas où f^* est une fonction constante par morceaux, et pour un choix approprié de non-linéarité σ. En outre, le nombre minimal de neurones pour obtenir la convergence ne dépend pas de la précision souhaitée, mais seulement des propriétés de la fonction cible f^*. Cela nous distingue des approches courantes pour l'analyse de la dynamique d'apprentissage des réseaux de neurones, le noyau tangent et le régime à champ moyen, qui requièrent toutes deux que le nombre de neurones grandisse afin d'obtenir une précision arbitraire.

1.4.6 Contexte structuré et grammaire à haut taux de couverture pour les questions-réponses conversationnelles basées sur un graphe de connaissance [Chapitre 7 du manuscrit]

Nous présentons dans le Chapitre 7 des résultats dédiés à l'architecture Transformer, cette fois avec une approche plus algorithmique. Notre objectif est de construire un réseau de neurones de type Transformer pour la tâche de question-réponse conversationnelle basée sur un graphe de connaissance. Décrivons brièvement cette tâche : un graphe de connaissance est un graphe qui encode des informations factuelles, où les nœuds représentent des concepts et les arêtes (dirigées et labellisées) représentent des relations entre les concepts. L'objectif de la tâche est donc de répondre à une série de questions factuelles en cherchant la réponse dans le graphe. Une des difficultés est la taille du graphe (qui est de l'ordre du milliard d'arêtes voire davantage).

Nous nous attaquons à cette tâche en construisant un modèle d'analyse sémantique, c'est-à-dire un modèle de langue qui transforme une phrase en anglais en une requête formelle formulée dans un langage de programmation, qui peut ensuite être exécutée pour obtenir une réponse. Le modèle fonctionne en deux étapes : dans un premier temps, un contexte de taille raisonnable est extrait du graphe et organisé dans une structure d'arbre, où chaque nœud de l'arbre contient un vecteur d'information. Dans un second temps, un réseau de neurones de type Transformer prend en entrée cet arbre et retourne la requête formelle.

Notre travail présente deux contributions méthodologiques principales : premièrement, nous construisons un langage de programmation (une "grammaire") de taille réduite, mais adapté à la modélisation d'une large gamme de questions sur des arbres de connaissance. Deuxièmement, nous introduisons une variante de Transformer qui peut opérer sur des données organisées sous forme d'arbre. Une des difficultés rencontrées est l'absence de données supervisées pour entraîner notre modèle. Nous procédons donc à la création d'un tel jeu de données, en partant d'ensembles de questions-réponses en anglais, et en trouvant par force brute des requêtes logiques qui correspondent aux questions. Nous pouvons ensuite entraîner notre modèle Transformer à l'aide de ces données.

Nous évaluons notre système sur deux jeux de données de questions-réponses basées sur le graphe Wikidata, qui est le plus grand graphe de connaissance public au monde. Notre approche obtient de meilleures performances en termes de précision des réponses par rapport à l'état de l'art sur les deux jeux de données.

Part I

From discrete to continuous architectures: neural networks in the large-depth regime

2

Scaling residual networks in the large-depth regime

Deep ResNets are recognized for achieving state-of-the-art results in complex machine learning tasks. However, the remarkable performance of these architectures relies on a training procedure that needs to be carefully crafted to avoid vanishing or exploding gradients, particularly as the depth L increases. No consensus has been reached on how to mitigate this issue, although a widely discussed strategy consists in scaling the output of each layer by a factor α_L. We show in a probabilistic setting that with standard i.i.d. initializations, the only non-trivial dynamics is for $\alpha_L = 1/\sqrt{L}$—other choices lead either to explosion or to identity mapping. This scaling factor corresponds in the continuous-time limit to a neural stochastic differential equation, contrarily to a widespread interpretation that deep ResNets are discretizations of neural ordinary differential equations. By contrast, in the latter regime, stability is obtained with specific correlated initializations and $\alpha_L = 1/L$. Our analysis suggests a strong interplay between scaling and regularity of the weights as a function of the layer index. Finally, in a series of experiments, we exhibit a continuous range of regimes driven by these two parameters, which jointly impact performance before and after training.

Contents

2.1 Introduction

2.1.1 Deep residual neural networks

Residual neural networks (ResNets), introduced by He et al. (2016a) in the field of computer vision, were the first deep neural network models successfully trained with several thousand layers. Since then, extensive empirical evidence has shown that increasing the depth leads to significant improvements in performance, while raising new challenges in terms of training (e.g., Wang et al., 2022). From a high-level perspective, the key feature of ResNets is the presence of skip connections between successive layers. In mathematical terms, this means that the $(k+1)$-th hidden state $h_{k+1} \in \mathbb{R}^d$ follows sequentially from the previous hidden state via the recurrence relation

$$h_{k+1} = h_k + f(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1, \tag{2.1}$$

where $f(\cdot, \theta_{k+1}) : \mathbb{R}^d \to \mathbb{R}^d$ is the layer function parameterized by $\theta_{k+1} \in \mathbb{R}^p$ and L is the number of layers. The skip connection corresponds to the addition of h_k on the right-hand side of (2.1), which is absent in classical feedforward networks. This refinement prevents instability issues during training when L is large, provided training is performed carefully (He et al., 2015). The idea of adding skip connections has become common practice in the field of deep learning, and is today incorporated in many other models such as Transformers in natural language processing (Vaswani et al., 2017). For simplicity, in the rest of the chapter, we continue to use the terminology ResNets to denote any architecture of the form (2.1), keeping in mind that this framework goes beyond the original model of He et al. (2016a).

The most common architectures have 50-150 layers, but ResNets can be trained with depths up to the order of thousand layers (He et al., 2016b). Yet, the training procedure needs to be carefully crafted to avoid vanishing or exploding gradients, particularly as the depth increases. As pointed out by, e.g., Shao et al. (2020), these instabilities are related to a shift in the magnitude of the variance of a signal as it passes through the network. In the original approach of He et al. (2016a), the issue was mitigated by adding a normalization step, called batch normalization (Ioffe and Szegedy, 2015), which rescales the output of each layer via centering and unit variance normalization. However, this normalization stage introduces practical and theoretical difficulties, among which computational overhead and strong dependence on the batch size (see Brock et al., 2021, and the references therein). A widespread alternative to stabilize training in deep models, explored for example by Yang and Schoenholz (2017), Arpit et al. (2019), Zhang et al. (2019b), and De and Smith (2020), is to incorporate a scaling factor α_L in front of the residual term in (2.1), yielding the model

$$h_{k+1} = h_k + \alpha_L f(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1. \tag{2.2}$$

There is strong evidence that this scaling factor α_L should depend on L, without however any consensus to date on the exact form of this dependence, nor on the mathematical grounding of

the approach. Thus, despite progresses on the empirical side, the mathematical forces in action behind the stability of deep ResNets are still poorly understood, although they are key to unlock training at arbitrary depth.

Our goal in the present chapter is to take a step forward towards a better theoretical understanding of deep ResNets by providing a thorough probabilistic analysis of the sequence $(h_k)_{0\leqslant k\leqslant L}$ at initialization when L is large, and by leveraging a continuous-time interpretation of model (2.2) via the so-called neural ordinary differential equation (neural ODE, Chen et al., 2018a) paradigm. In a nutshell, our results highlight the intimate connection that exists at initialization between stability of the learning process, the regularity of the weights, and the scaling factor α_L. We offer in particular a proper mathematical grounding on why and how to choose the parameter α_L as a function of the depth L and the distribution of the weights.

2.1.2 Our contributions

Scaling at initialization. The optimal parameters of ResNets are learned by minimizing some empirical risk function via a gradient descent algorithm. As highlighted for example by Yang and Schoenholz (2017), Hanin and Rolnick (2018), and Arpit et al. (2019), a good parameter initialization of this learning phase plays a major role in the quality of the learned model, in particular to avoid vanishing gradients and deadlock at initialization, or exploding gradients and quick divergence of the model parameters at the beginning of training. Moreover, a good initialization allows the use of larger learning rates, which have been shown to correlate with better generalization (Jastrzkebski et al., 2017). It is thus of great interest to study and understand the role played by scaling of deep ResNets at initialization. This is the context in which we place ourselves in the sequel.

At initialization stage, the weights $(\theta_k)_{1\leqslant k\leqslant L}$ are usually chosen as (realizations of) independent and identically distributed (i.i.d.) random variables, which typically follow a uniform or Gaussian distribution on $\mathbb{R}^p$. Accordingly, the sequence $(h_k)_{0\leqslant k\leqslant L}$ that results from the recursion (2.2) for a given input to the network takes the form of a sequence of random variables that are not i.i.d. but are actually a martingale. Thus, denoting informally by $\mathscr{L}$ the differentiable loss associated with the learning task (classification or regression), the distributions of $(h_k)_{0\leqslant k\leqslant L}$ and $(\frac{\partial \mathscr{L}}{\partial h_k})_{0\leqslant k\leqslant L}$ as L becomes large carry useful information on the stability of training. For instance, exploding gradients in the backpropagation phase of learning correspond to the fact that, with high probability, $\|\frac{\partial \mathscr{L}}{\partial h_0}\| \gg \|\frac{\partial \mathscr{L}}{\partial h_L}\|$, where $\|\cdot\|$ denotes the Euclidean norm. Our first contribution, in Section 2.2, is to provide thorough mathematical statements on the behavior of these distributions (both for finite and infinite L), depending on the value of α_L. Among other results, we show that only the choice $\alpha_L \approx 1/\sqrt{L}$ yields a non-trivial behavior at initialization, thereby confirming empirical findings in the literature (Arpit et al., 2019; De and Smith, 2020). For $\alpha_L \gg 1/\sqrt{L}$, the norms explode exponentially fast with L, which is inappropriate for training. For $\alpha_L \ll 1/\sqrt{L}$, the network is almost equivalent to identity, that is, $h_L \approx h_0$. The analysis of the different cases as a function of α_L is mathematically involved and makes extensive use of concentration tools from random matrix theory.

The continuous approach. As noticed by several authors (Chen et al., 2018a; Thorpe and van Gennip, 2022; E et al., 2019), model (2.2) with a scaling factor $\alpha_L = 1/L$ (and not $1/\sqrt{L}$) is formally similar to the discretization of a differential equation. Thus, when L tends to infinity, the weights and hidden states change continuously with the layer according to the equation

$$\frac{dH_t}{dt} = f(H_t, \Theta_t), \quad t \in [0,1]. \tag{2.3}$$

Here, time t is the continuous analogue of the layer index k, $H : [0,1] \to \mathbb{R}^d$ is a continuous-time hidden state, and $\Theta : [0,1] \to \mathbb{R}^p$ a continuous-time parameter. This important connection between ResNets and differential equations has been identified in the past years under the umbrella name of neural ODE. Since the original article of Chen et al. (2018a), this point of view has led to the development of a variety of new continuous-time models, together with innovative architectures and efficient training algorithms (Chang et al., 2019; Grathwohl et al., 2019; Kidger et al., 2021). The neural ODE paradigm also enabled to leverage the rich theory of differential equations to better understand the mechanisms at work behind deep ResNets (E et al., 2019). However, there is a debated question in the neural ODE community about the choice $\alpha_L = 1/L$, which guarantees convergence of the discrete model (2.2) to its continuous-time counterpart (2.3). As a matter of fact, it seems that this choice is guided by more mathematical than practical considerations, and several authors have suggested that it is inconsistent with what is done in practice (Cohen et al., 2021; Bayer et al., 2023). Moreover, letting $\alpha_L = 1/L$ is somewhat contradictory with the results discussed above, which highlighted that the only non-trivial limit at initialization is $\alpha_L = 1/\sqrt{L}$. Thus, as a second contribution, we clarify the problem in Section 2.3 by leveraging our previous results on stability. We show that the value $\alpha_L = 1/\sqrt{L}$ corresponds in the continuous world to a neural stochastic differential equation (SDE) of the form (2.3), where now $\Theta : [0,1] \to \mathbb{R}^p$ takes the form of a continuous-time stochastic process, typically a Brownian motion. By contrast, we also prove that the neural ODE regime with $\alpha_L = 1/L$ corresponds to the limit of a ResNet, not with i.i.d. weights as considered before, but with more complex and correlated weight distributions. For these weight distributions, the scaling $\alpha_L = 1/L$ is also a critical value between explosion and identity.

Going further, our third contribution is to exhibit in Section 2.4 a continuous range of regimes that are controlled by the choice of α_L (beyond the cases $1/\sqrt{L}$ and $1/L$) and the distribution of $(\theta_k)_{1 \leqslant k \leqslant L}$ at initialization, derived from a continuous-time process Θ with a regularity different from a Brownian motion. More precisely, we show experimentally that there is a strong interplay (with the same three cases—explosion, identity mapping, non-trivial behavior) between the choice of α_L and the regularity of $(\theta_k)_{1 \leqslant k \leqslant L}$ as a function of the layer index k, and this will be further investigated in Chapter 3. In addition, empirical evidence suggests that this interplay impacts both the behavior and performance of the networks during training, beyond initialization.

Organization of the chapter. The proofs of the results are postponed to the end of the chapter in Section 2.A. Sections 2.B to 2.D contain results that are useful for the proofs. Finally, Section 2.E details our experimental setting.

2.1.3 Related work

The choice of scaling for ResNets has been discussed in many papers, without however reaching a clear consensus on the form this scaling factor should take. For instance, Hanin and Rolnick (2018) state that stability requires $\alpha_L \leqslant 1/L$, while Zhang et al. (2019b) show that $\alpha_L \leqslant 1/\sqrt{L}$ is enough to ensure stability. On the other hand, Cohen et al. (2021) claim that the scaling factor observed in practice in trained ResNets is of the form $1/L^\beta$ with $\beta \approx 0.7$. Other authors have proposed more complex choices for α_L (e.g., Zhang et al., 2019a; Shao et al., 2020). Taking another point of view, De and Smith (2020) observe that batch normalization is empirically equivalent to taking a $1/\sqrt{L}$ normalization factor. Bachlechner et al. (2021) suggest learning a scaling parameter α_k that is allowed to vary from one layer to another, whereas, in (2.4), α_L is kept constant across layers. These authors observe a great acceleration for training compared to traditional ResNets with no scaling. They also suggest a similar architecture for Transformers and then notice that $\alpha_k \approx 1/L$ at the end of training.

Closest to our analysis at initialization are the papers of Arpit et al. (2019) and Zhang et al. (2019b). Arpit et al. (2019) develop a theoretical analysis based on mean field approximation that suggests that a scaling factor $\alpha_L = 1/\sqrt{L}$ prevents vanishing/exploding gradients at initialization, and provide experimental evidence that this approach is competitive with batch normalization. However, the authors do not provide rigorous mathematical statements for the three different cases $\alpha_L \ll 1/\sqrt{L}$, $\alpha_L \approx 1/\sqrt{L}$, and $\alpha_L \gg 1/\sqrt{L}$, nor do they highlight the connection with the continuous-time interpretation. Interestingly, the idea of exploiting the martingale structure to analyze the magnitude of the hidden states is present in Zhang et al. (2019b), who study the convergence of gradient descent for over-parameterized ResNets with different values of α_L. Nevertheless, they consider a specific model with Gaussian weights, and only provide asymptotic results when both width and depth tend to infinity.

The connection between the choice of scaling and the continuous-time point of view has previously been noticed by Zhang et al. (2019c), then studied in detail by Cohen et al. (2021). The latter show that, under assumptions on the form of the weights, it is possible to derive limiting (stochastic or ordinary) differential equations for the hidden states. However, they do not discuss the transition between these two regimes, nor do they link differential equations regimes with the stability of the network.

2.2 Scaling at initialization

Our goal in this section is to study the effect of the scaling factor α_L on the stability of ResNets at initialization, assuming that the weights are i.i.d. random variables. We start by making more precise the model and the learning problem introduced in (2.1).

2.2.1 Model and assumptions

Model. The data is a sample of n pairs $(x_i, y_i)_{1 \leqslant i \leqslant n}$, where x_i is the input vector in $\mathbb{R}^{n_{\text{in}}}$ and $y_i \in \mathbb{R}^{n_{\text{out}}}$ is the output vector to be predicted. This setting includes regression and classification (after one-hot encoding of the labels). Specifying the informal recurrence (2.1), for any input $x \in \mathbb{R}^{n_{\text{in}}}$, we consider the output $F_\pi(x) \in \mathbb{R}^{n_{\text{out}}}$ of the ResNet model defined by

$$\begin{aligned} h_0 &= Ax, \\ h_{k+1} &= h_k + \alpha_L V_{k+1} g(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1, \\ F_\pi(x) &= Bh_L, \end{aligned} \tag{2.4}$$

where $\alpha_L > 0$ is the scaling factor of the ResNet and $\pi = (A, B, (\theta_k)_{1 \leqslant k \leqslant L}, (V_k)_{1 \leqslant k \leqslant L})$ are its parameters, with $A \in \mathbb{R}^{d \times n_{\text{in}}}$, $B \in \mathbb{R}^{n_{\text{out}} \times d}$, $\theta_k \in \mathbb{R}^p$ and $V_k \in \mathbb{R}^{d \times d}$ for $k = 1, \dots, L$. The almost-everywhere differentiable function $g : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ encodes the choice of architecture. We note that the model includes initial and final linear layers in order to map the input space $\mathbb{R}^{n_{\text{in}}}$ into the space of hidden states $\mathbb{R}^d$, and symmetrically to map the last hidden state h_L into the output space $\mathbb{R}^{n_{\text{out}}}$. These two transformations are of little interest to us, since we mostly focus on the behavior of the sequence of hidden states $(h_k)_{0 \leqslant k \leqslant L}$. Let us finally notice that the results of this section can be adapted to hidden layers that do not have the same width, at the cost of increased technicality.

An important feature of model (2.4) is that the layer function takes the form of a matrix-vector multiplication, which will prove crucial to make use of concentration results on random matrices. We stress that this setting is standard in practice and that it encompasses many types of ResNets. It includes for example simple ResNets where $g(h, \theta) = \sigma(h)$ with σ the activation function, and the original ResNets from He et al. (2016a), which have

$$g(h, \theta) = \text{ReLU}(Wh + b),$$

where the parameter is a pair $\theta = (W, b)$ with $W \in \mathbb{R}^{d\times d}$ a weight matrix and $b \in \mathbb{R}^d$ a bias, and ReLU: $x \mapsto \max(x, 0)$ is applied element-wise. This setting also includes attention layers, where g corresponds to the scaled dot-product between keys and queries, as well as convolutional layers. Although the assumptions we make later have to be slightly modified to cover this context, the rationale should extend. We leave this extension for future work.

Throughout the chapter, we let $\ell : \mathbb{R}^{n_{\text{out}}} \times \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}_+$ be a loss function, differentiable w.r.t. its first parameter, for example the squared loss or the cross-entropy loss. The objective of learning is to find the optimal parameter π that minimizes the empirical risk $\mathscr{L}(\pi) = \sum_{i=1}^n \ell(F_\pi(x_i), y_i)$.

Probabilistic setting at initialization. The minimization of the empirical risk is usually performed by stochastic gradient descent or one of its variants (Goodfellow et al., 2016, Chapter 8). The gradient descent is initialized by choosing the weights as (realizations of) i.i.d. random variables. The parameters $\theta_1, V_1, \dots, \theta_L, V_L$ in model (2.4) are therefore assumed to be an i.i.d. collection of random variables, where we recall that $\theta_k \in \mathbb{R}^p$ and $V_k \in \mathbb{R}^{d\times d}$ parameterize the k-th layer of the network. In this stochastic context, the successive hidden states $h_0, \dots, h_L$ given a fixed input x are also random variables, but their distribution is not i.i.d.—in fact, under our assumptions, this sequence is a martingale. To avoid unnecessary technicalities, we assume that the sequence $(h_k)_{0\leqslant k\leqslant L}$ is non-atomic. This is for example the case if the distribution of the parameters is absolutely continuous w.r.t. the Lebesgue measure. In particular, this ensures that the sequence $(h_k)_{0\leqslant k\leqslant L}$ almost surely does not hit the non-differentiability points of g.

It is stressed that the distribution of the parameters are assumed to be independent of the depth, so that all the dependence on L is captured in the scaling factor α_L. This model enables us to consider multiple architectures at once, via the function g. By contrast, some authors formulate the problem of scaling as a choice of the variance at initialization (e.g., Yang and Schoenholz, 2017; Wang et al., 2022), which makes the analysis architecture-dependent. However, for a given architecture, these two approaches are essentially equivalent since $\operatorname{Var}(\alpha_L V_k) = \alpha_L^2 \operatorname{Var}(V_k)$.

The quantity $\|h_L - h_0\|/\|h_0\|$ carries key information on the behavior of the network at initialization. On the one hand, if $\|h_L - h_0\| \ll \|h_0\|$, the network is essentially equal to the identity function. On the other hand, if $\|h_L - h_0\| \gg \|h_0\|$, the output of the network explodes. An intermediate situation is when $\|h_L - h_0\| \approx \|h_0\|$. In addition, another source of information is provided by the gradients of the hidden states with respect to the empirical risk $\mathscr{L}$. If $\|\frac{\partial \mathscr{L}}{\partial h_0} - \frac{\partial \mathscr{L}}{\partial h_L}\| \ll \|\frac{\partial \mathscr{L}}{\partial h_L}\|$, the gradients do not change as they flow through the network, which means that the exact same information is backpropagated throughout the network. Conversely, if $\|\frac{\partial \mathscr{L}}{\partial h_0} - \frac{\partial \mathscr{L}}{\partial h_L}\| \gg \|\frac{\partial \mathscr{L}}{\partial h_L}\|$, the gradients explode during backpropagation. By exploiting the martingale structure of $(\|h_k\|)_{0\leqslant k\leqslant L}$, as well as state-of-the-art concentration inequalities for random matrices with sub-Gaussian entries, we provide in this section probabilistic bounds on the magnitude of these various quantities.

Assumptions. Some assumptions are needed on the choices of architecture and initialization. Recall that a centered real-valued random variable X is said to be s^2 sub-Gaussian (van Handel, 2016, Chapter 3) if for all $\lambda \in \mathbb{R}$, $\mathbb{E}(\exp(\lambda X)) \leqslant \exp(\lambda^2 s^2/2)$. The sub-Gaussian property is a constraint on the tail of the probability distribution. As an example, Gaussian random variables on the real line are sub-Gaussian and so are bounded random variables.

The following assumptions will be needed throughout the section: for any $1 \leqslant k \leqslant L$,

(A_1) For some $s \geqslant 1$, the entries of $\sqrt{d}V_k$ are centered i.i.d. s^2 sub-Gaussian random variables, independent of d and L, with unit variance.

(A_2) For some $C > 0$, independent of d and L, and for any $h \in \mathbb{R}^d$,

$$\frac{\|h\|^2}{2} \leqslant \mathbb{E}\big(\|g(h, \theta_k)\|^2\big) \leqslant \|h\|^2 \quad \text{and} \quad \mathbb{E}\big(\|g(h, \theta_k)\|^8\big) \leqslant C\|h\|^8.$$

Assumption (A_1) is mild and satisfied by all initializations used in practice. For example, the classical Glorot initialization (Glorot and Bengio, 2010)—which is the default implementation in the Keras package (Chollet et al., 2015)—takes the entries of V_k as uniform $\mathcal{U}(-\sqrt{3/d}, \sqrt{3/d})$ variables. This means that $\sqrt{d}V_k$ is initialized with $\mathcal{U}(-\sqrt{3}, \sqrt{3})$ random variables, which satisfy (A_1). Other examples include the Gaussian $\mathcal{N}(0, 1/d)$ initialization of He et al. (2015) and, for example, initialization with Rademacher variables.

The first part of Assumption (A_2) ensures that $g(\cdot, \theta_k)$ is not too far away from being an isometry in expectation. The second part is more technical and, roughly, allows to upper bound the deviations of the norm of $g(h_{k-1}, \theta_k)$. Our next Proposition 2.1 shows that most classical ResNet architectures verify Assumption (A_2). For the sake of readability, these models, together with their parameters, are summarized in Table 2.1 below.

	Name	**Recurrence relation**	**Parameters**
`res-1`	Simple ResNet	$h_{k+1} = h_k + \alpha_L V_{k+1}\sigma(h_k)$	$\theta_{k+1} = \emptyset$
`res-2`	Parametric ResNet	$h_{k+1} = h_k + \alpha_L V_{k+1}\sigma(W_{k+1}h_k)$	$\theta_{k+1} = W_{k+1}$
`res-3`	Original ResNet	$h_{k+1} = h_k + \alpha_L V_{k+1}\,\mathrm{ReLU}(W_{k+1}h_k)$	$\theta_{k+1} = W_{k+1}$

Table 2.1: Examples of ResNet architectures considered in the chapter. In the first two cases, the activation function σ is such that, for all $x \in \mathbb{R}, a|x| \leqslant |\sigma(x)| \leqslant b|x|$, $1/\sqrt{2} \leqslant a < b \leqslant 1$. In the last two cases, $W_{k+1} \in \mathbb{R}^{d\times d}$.

Proposition 2.1. *Let* `res-1`*,* `res-2`*, and* `res-3` *be the models defined in Table 2.1. Then*

(*i*) *Assumption* (A_2) *is satisfied for* `res-1`*.*

(*ii*) *Assumption* (A_2) *is satisfied for* `res-2` *and* `res-3`*, as soon as the entries of* $\sqrt{d}W_{k+1}$*,* $0 \leqslant k \leqslant L-1$*, are centered i.i.d. sub-Gaussian random variables, independent of* d *and* L*, with unit variance.*

Proof. See Section 2.A.1. □

In the models `res-1` and `res-2`, σ can be, for instance, taken as the parametric ReLU function, i.e., $\sigma(x) = x_+ + sx_-$, where x_+ (resp. x_-) denotes the positive (resp. negative) part and the slope $s \in [1/\sqrt{2}, 1]$ is a parameter of the model. Observe also that `res-2` differs from `res-3` since the classical ReLU function is defined by $\mathrm{ReLU}(x) = x_+$ and thus does not satisfy the condition $|\sigma(x)| \geqslant a|x|$. Note that there is no bias term in these three models, as this term is commonly initialized to zero, and we are interested in the behavior at initialization.

2.2.2 Probabilistic bounds on the norm of the hidden states

The next two propositions describe how the quantity $\|h_L - h_0\|/\|h_0\|$ changes as a function of $L\alpha_L^2$. Proposition 2.2 provides a high-probability bound of interest when $L\alpha_L^2 \ll 1$. In this case, we see that, with high probability, the network acts as the identity function, directly mapping h_0 to h_L. On the other hand, Proposition 2.3 provides information in the two cases $L\alpha_L^2 \gg 1$ and $L\alpha_L^2 \approx 1$. When $L\alpha_L^2 \gg 1$, the lower bound (i) indicates an explosion with high probability of the norm of the last hidden state. On the other hand, when $L\alpha_L^2 \approx 1$, the bounds (i) and (ii) show that h_L randomly varies around h_0 with fluctuation sizes bounded from below and above.

Proposition 2.2. *Consider a ResNet* (2.4) *such that Assumptions* (A_1) *and* (A_2) *are satisfied. If* $L\alpha_L^2 \leqslant 1$*, then, for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\frac{\|h_L - h_0\|^2}{\|h_0\|^2} \leqslant \frac{2L\alpha_L^2}{\delta}.$$

Proof. See Section 2.A.2. □

Proposition 2.3. *Consider a ResNet* (2.4) *such that Assumptions* (A_1) *and* (A_2) *are satisfied.*

(*i*) *Assume that* $d \geqslant 64$ *and* $\alpha_L^2 \leqslant \frac{2}{(\sqrt{C}s^4+4\sqrt{C}+16s^4)d}$*. Then, for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\frac{\|h_L - h_0\|^2}{\|h_0\|^2} > \exp\left(\frac{3L\alpha_L^2}{8} - \sqrt{\frac{11L\alpha_L^2}{d\delta}}\right) - 1,$$

provided that

$$2L\exp\left(-\frac{d}{64\alpha_L^2 s^2}\right) \leqslant \frac{\delta}{11}. \tag{2.5}$$

(*ii*) *Assume that* $\alpha_L^2 \leqslant \frac{1}{\sqrt{C}(d+128s^4)}$*. Then, for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\frac{\|h_L - h_0\|^2}{\|h_0\|^2} < \exp\left(L\alpha_L^2 + \sqrt{\frac{5L\alpha_L^2}{d\delta}}\right) + 1.$$

Proof. See Section 2.A.3. □

Note that the assumptions of Proposition 2.3 on d and α_L are mild, since in the learning tasks where deep ResNets are involved, one typically has $\alpha_L = 1/L^\beta$ with $\beta > 0$, $d \geqslant 10^2$ and $L \geqslant 10^2$. Note also that condition (2.5) is not severe since, when d and L are large, it encompasses all reasonable values of δ. Propositions 2.2 and 2.3 are interesting in the sense that they provide finite-depth high-probability bounds on the behavior of the hidden states, depending on the magnitude of $L\alpha_L^2$. The results become clearer by letting $\alpha_L = 1/L^\beta$, with $\beta > 0$, as shown in the following corollary.

Corollary 2.4. *Consider a ResNet* (2.4) *such that Assumptions* (A_1) *and* (A_2) *are satisfied, and let* $\alpha_L = 1/L^\beta$*, with* $\beta > 0$*.*

(*i*) *If* $\beta > 1/2$*, then*

$$\frac{\|h_L - h_0\|}{\|h_0\|} \xrightarrow[L\to\infty]{\mathbb{P}} 0.$$

(*ii*) *If* $\beta < 1/2$ *and* $d \geqslant 9$*, then*

$$\frac{\|h_L - h_0\|}{\|h_0\|} \xrightarrow[L\to\infty]{\mathbb{P}} \infty.$$

(*iii*) *If* $\beta = 1/2$*,* $d \geqslant 64$*,* $L \geqslant (\frac{1}{2}\sqrt{C}s^4 + 2\sqrt{C} + 8s^4)d + 96\sqrt{C}s^4$*, then, for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\exp\left(\frac{3}{8} - \sqrt{\frac{22}{d\delta}}\right) - 1 < \frac{\|h_L - h_0\|^2}{\|h_0\|^2} < \exp\left(1 + \sqrt{\frac{10}{d\delta}}\right) + 1,$$

provided that

$$2L\exp\Big(-\frac{Ld}{64s^2}\Big) \leqslant \frac{\delta}{11}.$$

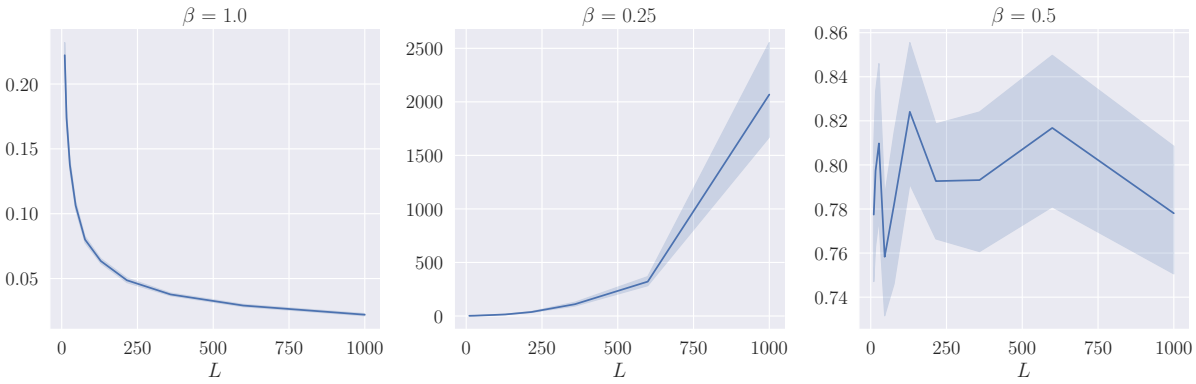
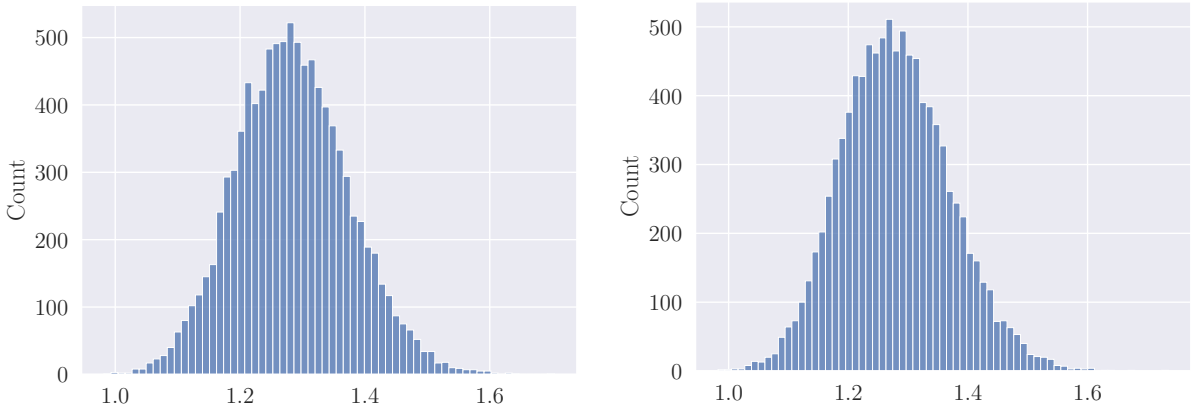


Figure 2.1: Evolution of $\|h_L - h_0\|/\|h_0\|$ as a function of L for different values of β and an i.i.d. $\mathcal{U}(-\sqrt{3/d}, \sqrt{3/d})$ initialization of model `res-3`, with $d = 40$. The input is a random Gaussian observation x in dimension $n_{\text{in}} = 64$. The experiment is repeated with 50 independent randomizations.



(a) Distribution of $\|h_L\|/\|h_0\|$

(b) Distribution of $\|\frac{\partial \mathscr{L}}{\partial h_0}\|/\|\frac{\partial \mathscr{L}}{\partial h_L}\|$

Figure 2.2: Empirical distributions of the norms for $\beta = 1/2$, $L = 10^3$, $d = 100$. The experiment is repeated with 10^4 independent randomizations.

Proof. See Section 2.A.4. □

Corollary 2.4 highlights three different asymptotic behaviors for $\|h_L\|$, depending on the values of β. For $\beta > 1/2$, statement (i) tells that h_L converges towards h_0 in probability, as L tends to infinity, which means that the neural network is essentially equivalent to an identity mapping. On the other hand, for $\beta < 1/2$, the norm of h_L explodes with high probability. Finally, for the critical value $\beta = 1/2$, we see that h_L fluctuates around h_0, with a fluctuation size independent of L. Observe that the lower bound in (iii) is not trivial as soon as $\exp(3/8 - \sqrt{11/d\delta}) > 1$, i.e., $d > 99/64\delta$. The message of Corollary 2.4 is that the only scaling leading to a non-degenerate distribution at initialization is for $\beta = 1/2$.

The three statements of Corollary 2.4 are illustrated in Figure 2.1. In this experiment, we consider model `res-3`, a random Gaussian observation x in dimension $n_{\text{in}} = 64$, and parameters initialized with a uniform distribution $\mathcal{U}(-\sqrt{3/d}, \sqrt{3/d})$. We refer to Appendix 2.E for a detailed setup of all the experiments of the chapter. Figure 2.2a shows the empirical distribution of $\|h_L\|/\|h_0\|$ when $\beta = 1/2$ for a large number of realizations. This figure illustrates in particular that our bounds are reasonably sharp, since the bounds indicate that the first quartile of the

distribution is larger than 0.87 (whereas the first quartile of the empirical histogram is equal to 1.21) and the third quartile is less than 2.06 (whereas the third quartile of the empirical histogram is equal to 1.34). Determining the exact distribution of $\|h_L\|/\|h_0\|$ is an interesting avenue for future research that is beyond the scope of the present chapter. There is however a strong indication that the ratio follows a log-normal distribution, as confirmed by a normality test on (the log of) the empirical distribution.

In a nutshell, the proofs of Propositions 2.2 and 2.3 rest upon controlling of the norm of the hidden states, which obeys the recurrence

$$\|h_{k+1}\|^2 = \|h_k\|^2 + \alpha_L^2 \|V_{k+1} g(h_k, \theta_{k+1})\|^2 + 2\alpha_L \langle h_k, V_{k+1} g(h_k, \theta_{k+1}) \rangle, \tag{2.6}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard scalar product in $\mathbb{R}^d$. Taking the expectations on both side, one deduces with Assumptions (A_1) and (A_2) that

$$\begin{aligned} \mathbb{E}\big(\|V_{k+1} g(h_k, \theta_{k+1})\|^2\big) &= \mathbb{E}\Big(\mathbb{E}\big(\|V_{k+1} g(h_k, \theta_{k+1})\|^2\big) \,\Big|\, h_k, \theta_{k+1}\Big) \\ &= \mathbb{E}\big(\|g(h_k, \theta_{k+1})\|^2\big) \approx \|h_k\|^2 \end{aligned} \tag{2.7}$$

and

$$\mathbb{E}\big(\langle h_k, V_{k+1} g(h_k, \theta_{k+1}) \rangle\big) = \mathbb{E}\Big(\mathbb{E}\big(\langle h_k, V_{k+1} g(h_k, \theta_{k+1}) \rangle \big| h_k, \theta_{k+1}\big)\Big) = 0. \tag{2.8}$$

The equalities (2.7) and (2.8) allow deriving without further work bounds in expectation on $\|h_L\|$, as already observed by Arpit et al. (2019). However, the results we are after are stronger since they involve high-probability bounds. A finer control of the deviations of $\|V_{k+1} g(h_k, \theta_{k+1})\|^2$ and $\langle h_k, V_{k+1} g(h_k, \theta_{k+1}) \rangle$ is then needed. This involves concentration inequalities on random matrices with sub-Gaussian entries.

2.2.3 Probabilistic bounds on the gradients

Propositions 2.2 and 2.3 provide insights on the output of the network when L is large. However, they do not carry information on the backwards dynamics of propagation of the gradients of the loss $p_k = \frac{\partial \mathscr{L}}{\partial h_k} \in \mathbb{R}^d$. Assessing the dynamics of the $(p_k)_{0 \leqslant k \leqslant L}$ as a function of L is important since the behavior of this sequence impacts trainability of the network at initialization. Thus, in this subsection, we are interested in quantifying the magnitude of $\|p_0 - p_L\|/\|p_L\|$, when L is large. Notice that, contrarily to the previous subsection where we were mostly interested in the last hidden state h_L, the quantity of interest is now p_0 (not p_L), the gradient at index 0. The reason is that the sequence $(p_k)_{0 \leqslant k \leqslant L}$ is defined backwardly, as we will see below. We also stress that $(p_k)_{0 \leqslant k \leqslant L}$ is the sequence of derivatives of the loss w.r.t. the hidden states h_k, and not w.r.t. the parameters. The reason for considering this sequence is that the p_k are involved in the backpropagation algorithm and are therefore essential for assessing the stability of the gradient descent (see, e.g., Arpit et al., 2019).

Analyzing the behavior of the sequence $(p_k)_{0 \leqslant k \leqslant L}$ is challenging since, according to the backpropagation (or reverse-mode differentiation) formula, one has

$$p_k = p_{k+1} + \alpha_L \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1}.$$

Taking the norm,

$$\|p_k\|^2 = \|p_{k+1}\|^2 + \alpha_L^2 \Big\| \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1} \Big\|^2 + 2\alpha_L \Big\langle p_{k+1}, \frac{\partial g(h_k, \theta_{k+1})}{\partial h}^\top V_{k+1}^\top p_{k+1} \Big\rangle.$$

Although the equation looks qualitatively similar to (2.6), it has the unpleasant feature that $\frac{\partial g(h_k, \theta_{k+1})}{\partial h}$ depends on h_k, hence on $\theta_1, V_1, \ldots, \theta_k, V_k$, while p_{k+1} depends on $\theta_{k+2}, V_{k+2}, \ldots,$

θ_L, V_L. This forbids applying directly the same proof techniques as for the hidden states. Therefore, to extract useful information from this recurrence equation, one needs to characterize the dependence of the distribution of $\frac{\partial g(h_k,\theta_{k+1})}{\partial h}$ with respect to h_k. To do so, it is sometimes assumed that these two quantities are independent (see, e.g., Yang and Schoenholz, 2017). However, assuming independence remains a strong requirement, which is not verified for many network architectures (for example model `res-1`). We tackle the problem from a different point of view and propose an alternative approach based on forward-mode differentiation, valid under a much weaker assumption. The cost we pay is that we obtain results in expectation and not in high probability.

Let us sketch our approach before stating the results more formally. We denote by $z \in \mathbb{R}^d$ an independent random variable that will be used as a tool to assess the magnitude of the gradients. For any $0 \leqslant i, j \leqslant L$, let $\frac{\partial h_j}{\partial h_i} \in \mathbb{R}^{d\times d}$ be the Jacobian matrix of h_j with respect to h_i. Recall that the (m, n)-th entry of this matrix equals the derivative of the m-th coordinate of h_j w.r.t. the n-th coordinate of h_i. Then, letting $q_k(z) = \frac{\partial h_k}{\partial h_0} z$, we have, by the chain rule,

$$q_{k+1}(z) = \frac{\partial h_{k+1}}{\partial h_k} q_k(z) = q_k(z) + \alpha_L V_{k+1} \frac{\partial g(h_k, \theta_{k+1})}{\partial h} q_k(z). \tag{2.9}$$

Identity (2.9), which is similar to (2.4), expresses $q_{k+1}(z)$ as a function of $q_k(z)$, and therefore respects the flow of information. Next, assuming that z is random with a Gaussian distribution, it is possible to express one of our quantities of interest, $\|p_0\|/\|p_L\|$, as a function of the last vector $q_L(z)$, by taking the expectation over z. Indeed,

$$\frac{\|p_0\|^2}{\|p_L\|^2} = \frac{1}{\|p_L\|^2} \mathbb{E}_{z\sim\mathcal{N}(0,I_d)}\Big(\big|p_0^\top z\big|^2\Big) = \mathbb{E}_{z\sim\mathcal{N}(0,I_d)}\left(\Big|\Big(\frac{p_L}{\|p_L\|}\Big)^\top q_L(z)\Big|^2\right), \tag{2.10}$$

where I_d is the identity matrix in $\mathbb{R}^d$ and the second equality is a consequence of

$$p_0^\top z = \Big(\frac{\partial \mathscr{L}}{\partial h_0}\Big)^\top z = \Big(\frac{\partial \mathscr{L}}{\partial h_L}\Big)^\top \frac{\partial h_L}{\partial h_0} z = p_L^\top q_L(z).$$

In summary, the recurrence (2.9) allows us to derive bounds on the norm of $q_L(z)$, which can then transfer to $\|p_0\|/\|p_L\|$ via (2.10). For this, it is necessary to make the following assumption on the ratio $p_L/\|p_L\|$:

(A_3) Let $b = p_L/\|p_L\|$. Then $\mathbb{E}(b|h_L) = 0$ and $\mathbb{E}(b^\top b|h_L) = I_d/d$.

It is a mild assumption, which is verified for instance if $n_{\text{out}} = 1$ with squared error (for regression) or cross-entropy (for binary classification). In these cases, $p_L/\|p_L\| = {}^{B^\top}/_{\|B\|_F}$, where $\|\cdot\|_F$ is the Frobenius norm and B is the weight matrix of the last layer. We finally need the following assumption, which is the equivalent of Assumption (A_2) for the gradients.

(A_4) One has, almost surely,

$$\frac{\|q_k\|^2}{2} \leqslant \mathbb{E}\left(\Big\|\frac{\partial g(h_k, \theta_{k+1})}{\partial h} q_k\Big\|^2 \Big| h_k, q_k\right) \leqslant \|q_k\|^2.$$

Assumption (A_4) is satisfied by all the standard architectures listed in Table 2.1, as shown by the next proposition.

Proposition 2.5. *Let `res-1`, `res-2`, and `res-3` be the models defined in Table 2.1. Assume that (A_1) is satisfied and σ is almost everywhere differentiable, with $a \leqslant \sigma' \leqslant b$. Then*

(*i*) *Assumption (A_4) is satisfied for `res-1`.*

(*ii*) *Assumption* (A_4) *is satisfied for* `res-2` *and* `res-3`*, when the entries of* $\sqrt{d}W_k, 1 \leqslant k \leqslant L$*, are centered i.i.d. random variables, independent of* d *and* L*, with unit variance.*

Proof. See Section 2.A.5. □

The next two propositions are the counterparts of Proposition 2.2 and Proposition 2.3 for the gradient dynamics.

Proposition 2.6. *Consider a ResNet* (2.4) *such that Assumptions* (A_1)*-*(A_4) *are satisfied. If* $L\alpha_L^2 \leqslant 1$*, then, for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\frac{\|p_0 - p_L\|^2}{\|p_L\|^2} \leqslant \frac{2L\alpha_L^2}{\delta}.$$

Proof. See Section 2.A.6. □

Proposition 2.7. *Consider a ResNet* (2.4) *such that Assumptions* (A_1)*-*(A_4) *are satisfied. Then*

$$\big(1 + \frac{1}{2}\alpha_L^2\big)^L - 1 \leqslant \mathbb{E}\bigg(\frac{\|p_0 - p_L\|^2}{\|p_L\|^2}\bigg) \leqslant (1+\alpha_L^2)^L - 1.$$

Proof. See Section 2.A.7. □

A simple corollary of the propositions above is as follows.

Corollary 2.8. *Consider a ResNet* (2.4) *such that Assumptions* (A_1)*-*(A_4) *are satisfied, and take* $\alpha_L = 1/L^\beta$*, with* $\beta > 0$*. Then*

(*i*) *If* $\beta > 1/2$*,*

$$\frac{\|p_0 - p_L\|}{\|p_L\|} \xrightarrow[L\to\infty]{\mathbb{P}} 0.$$

(*ii*) *If* $\beta < 1/2$*,*

$$\mathbb{E}\bigg(\frac{\|p_0 - p_L\|^2}{\|p_L\|^2}\bigg) \xrightarrow{L\to\infty} \infty.$$

(*ii*) *If* $\beta = 1/2$*,*

$$\exp\Big(\frac{1}{2}\Big) - 1 \leqslant \mathbb{E}\bigg(\frac{\|p_0 - p_L\|^2}{\|p_L\|^2}\bigg) \leqslant \exp(4) - 1.$$

Proof. See Section 2.A.8. □

Corollary 2.8 is illustrated in Figure 2.3. The experimental protocol is the same as in Figure 2.1, but we now track p_0 and p_L, the gradients of the loss $\mathscr{L}$ with respect to the first and the last hidden states. In accordance with our results, when $\beta > 1/2$, the gradient remains the same from one layer to another (left plot). On the other hand, the middle plot clearly shows that when $\beta < 1/2$ the gradient explodes. Once again, the case $\beta = 1/2$ (right plot) is the only one for which the distribution of gradients at initialization is non-trivial. Figure 2.2b illustrates that the empirical distribution of gradients in this case also seems to be log-normal.

In summary, this and the previous subsection both point towards the same conclusion: there are three different cases, depending on the value of β—explosion when $\beta < 1/2$, non-degenerate limit when $\beta = 1/2$, and identity when $\beta > 1/2$. In the explosion case, it is well known that the network cannot be trained (Yang and Schoenholz, 2017). The theory thus points out that the value $1/2$ plays a pivotal role. Remarkably, this value has a specific interpretation in the continuous-time point of view of ResNets, in terms of SDE. This is the topic that we address in the next section.

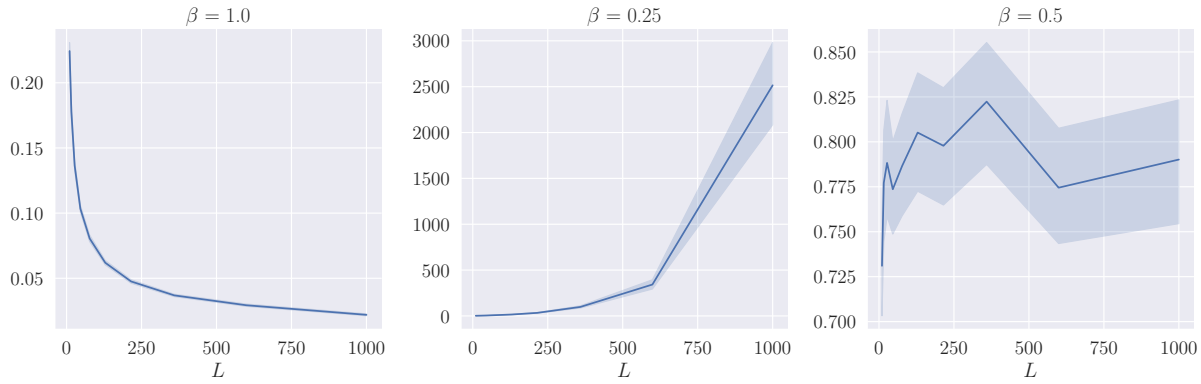


Figure 2.3: Evolution of $\|p_0 - p_L\|/\|p_L\|$ as a function of L for different values of β and an i.i.d. $\mathcal{U}(-\sqrt{3/d}, \sqrt{3/d})$ initialization of model `res-3`, with $d = 40$. The input is a random Gaussian observation x in dimension $n_{\text{in}} = 64$. The experiment is repeated with 50 independent randomizations.

2.3 Scaling in the continuous-time setting

Starting with the discrete ResNet (2.4), it is tempting to let L go to infinity and consider the network as the discretization of a differential equation where the layer index $k \in \{0, \dots, L\}$ is replaced by the time index $t \in [0, 1]$. This interpretation of deep neural networks has been popularized by Chen et al. (2018a) and is often referred to as the neural ODE paradigm. Notice that this setting is different from the so-called mean-field analysis, where the width of the network is assumed to be infinite beforehand. In our setting, the width d is assumed to be finite and fixed.

2.3.1 Convergence towards a SDE in the large-depth regime

One of the main messages of Section 2.2 is that the standard initialization with i.i.d. parameters leads to a non-degenerate model for large values of L only if $L\alpha_L^2 \approx 1$ (Propositions 2.2 and 2.3), or, equivalently, if $\beta = 1/2$ when $\alpha_L = 1/L^\beta$ (Corollary 2.4). Remarkably, in the continuous-time limit, this regime corresponds to the discretization of a SDE. Indeed, consider for simplicity the (discrete) ResNet model `res-1`

$$h_0 = Ax, \quad h_{k+1} = h_k + \frac{1}{\sqrt{L}} V_{k+1} \sigma(h_k), \quad 0 \leqslant k \leqslant L - 1, \tag{2.11}$$

where the entries of all $(V_k)_{1 \leqslant k \leqslant L}$ are assumed to be i.i.d. $\mathcal{N}(0, 1/d)$. Recall the following definition:

Definition 2.9. *A one-dimensional Brownian motion $(B_t)_{t \in [0,1]}$ is a continuous-time stochastic process with $B_0 = 0$, almost surely continuous, with independent increments, and such that for any $0 \leqslant s < t \leqslant 1$, $B_t - B_s \sim \mathcal{N}(0, t - s)$.*

Now, let $\mathbf{B} : [0, 1] \to \mathbb{R}^{d \times d}$ be a $(d \times d)$-dimensional Brownian motion, in the sense that the $(B_{ij})_{1 \leqslant i,j \leqslant d}$ are independent one-dimensional Brownian motions. Thus, for any $0 \leqslant k \leqslant L - 1$ and any $1 \leqslant i, j \leqslant d$, we have

$$\mathbf{B}_{(k+1)/L,i,j} - \mathbf{B}_{k/L,i,j} \sim \mathcal{N}\Big(0, \frac{1}{L}\Big),$$

and the increments for different values of (i, j, k) are independent. As a consequence, the recurrence (2.11) is equivalent in distribution to the recurrence

$$h_{k+1}^\top = h_k^\top + \sqrt{\frac{1}{d}}\sigma(h_k^\top)(\mathbf{B}_{(k+1)/L} - \mathbf{B}_{k/L}), \quad 0 \leqslant k \leqslant L-1.$$

(Note that this is true because V_{k+1} has the same distribution as $V_{k+1}^\top$.) We recognize the Euler-Maruyama discretization (Kloeden and Platen, 1992) on the $\{k/L, 0 \leqslant k \leqslant L\}$ mesh of the SDE

$$H_0 = Ax, \quad dH_t^\top = \sqrt{\frac{1}{d}}\sigma(H_t^\top)d\mathbf{B}_t, \quad t \in [0,1], \tag{2.12}$$

where the output of the network is now a function of the final value of H, that is, H_1. The link between the discrete ResNet (2.11) and the SDE (2.12) is formalized in the next proposition.

Proposition 2.10. *Consider the* `res-1` *model, where the entries of V_k are i.i.d. Gaussian $\mathcal{N}(0, 2/d)$ random variables. Assume that the activation function σ is Lipschitz continuous. Then the SDE* (2.12) *has a unique solution H and, for any $0 \leqslant k \leqslant L$,*

$$\mathbb{E}\big(\|H_{k/L} - h_k\|\big) \leqslant \frac{c}{\sqrt{L}},$$

for some $c > 0$.

Proof. See Section 2.A.9. □

Notice that the requirement that σ is Lipschitz continuous is satisfied by most classical activation functions, including ReLU. This proposition is interesting for several reasons. First, the scaling $\beta = 1/2$, which is exactly the one that yields a non-trivial dynamics at initialization, corresponds in the continuous world to a remarkably 'simple' model of diffusion. This shows that very deep neural networks properly initialized with i.i.d. weights are equivalent to solutions of SDE. This analogy opens interesting perspectives for training deep networks using automatic differentiation for solutions of neural SDE (Li et al., 2020b).

Second, we stress that the emergence of a SDE instead of an ODE carries an important message. Several authors (including, e.g., Thorpe and van Gennip, 2022) have shown that, under appropriate assumptions, a deep ResNet converges in the large depth limit to an ODE and not a SDE. The reason why we obtain a SDE here is intrinsically connected with the choice of i.i.d. initialization for the weights, which makes a Brownian motion appear at the limit, as highlighted above. In other words, the i.i.d. initialization, the choice $\beta = 1/2$ (the relevant critical value exhibited in Section 2.2), and the emergence of a SDE are intimately linked together. On the other hand, the case $\beta = 1$ matches with an ODE if the initialization is not i.i.d., as we will see in Subsection 2.3.2.

Finally, we point out that Proposition 2.10 states the convergence of a ResNet towards a SDE for the basic architecture `res-1` and for Gaussian initialization. The extension to more general settings is an interesting direction of research, although clearly beyond the scope of the present chapter (see, e.g., Peluchetti and Favaro, 2020, and Cohen et al., 2021, for results in this direction).

2.3.2 Scaling in the neural ODE setting

Convergence towards an ODE. The basic message of our Proposition 2.10 is that an i.i.d. initialization, together with $\beta = 1/2$, leads to a SDE rather than an ODE. A natural question is then whether a different choice of weight distributions (at initialization) and scaling can lead to a classical neural ODE.

To answer this question and leave the world of i.i.d. initialization, we assume that the weights $(V_k)_{1\leqslant k\leqslant L}$ and $(\theta_k)_{1\leqslant k\leqslant L}$ are discretizations of smooth functions $\mathscr{V} : [0,1] \to \mathbb{R}^{d\times d}$ and $\Theta : [0,1] \to \mathbb{R}^p$. We then consider the general iteration (2.4) with $\alpha_L = 1/L$, that is,

$$h_0 = Ax, \quad h_{k+1} = h_k + \frac{1}{L} V_{k+1} g(h_k, \theta_{k+1}), \quad 0 \leqslant k \leqslant L-1, \tag{2.13}$$

where $V_k = \mathscr{V}_{k/L}$ and $\theta_k = \Theta_{k/L}$. Of course, it is still possible to consider $(V_k)_{1\leqslant k\leqslant L}$ (resp. $(\theta_k)_{1\leqslant k\leqslant L}$) as random variables, by letting $(\mathscr{V}_t)_{t\in[0,1]}$ (resp. $(\Theta_t)_{t\in[0,1]}$) be a continuous-time stochastic process. In this model, we shall need the following assumption:

(A_5) For any $1 \leqslant k \leqslant L$, one has $V_k = \mathscr{V}_{k/L}$ and $\theta_k = \Theta_{k/L}$, where the stochastic processes $\mathscr{V}$ and Θ are almost surely Lipschitz continuous and bounded.

More precisely, almost surely, there exist $K_{\mathscr{V}}, K_\Theta, C_{\mathscr{V}}, C_\Theta > 0$, such that, for any $s, t \in [0,1]$,

$$\|\mathscr{V}_t - \mathscr{V}_s\| \leqslant K_{\mathscr{V}}|t-s|, \quad \|\Theta_t - \Theta_s\| \leqslant K_\Theta |t-s|, \quad \|\mathscr{V}_t\| \leqslant C_{\mathscr{V}}, \quad \|\Theta_t\| \leqslant C_\Theta.$$

A typical model that satisfies Assumption (A_5) is obtained by letting the entries of $\mathscr{V}$ and Θ be independent Gaussian processes with expectation zero and squared exponential covariance $K(x,x') = \exp(-\frac{(x-x')^2}{2\ell^2})$, where $\ell > 0$.

We shall also need the following requirement on g, which is satisfied by all our models as soon as σ is Lipschitz continuous:

(A_6) The function g is Lipschitz continuous on compact sets, in the sense that for any compact $\mathscr{P} \subseteq \mathbb{R}^p$, there exists $K_{\mathscr{P}} > 0$ such that, for all $h, h' \in \mathbb{R}^d$, $\theta \in \mathscr{P}$,

$$\|g(h,\theta) - g(h',\theta)\| \leqslant K_{\mathscr{P}} \|h - h'\|,$$

and for any compact $\mathscr{D} \subseteq \mathbb{R}^d$, there exists $K_{\mathscr{D},\mathscr{P}} > 0$ such that, for all $h \in \mathscr{D}$, $\theta, \theta' \in \mathscr{P}$,

$$\|g(h,\theta) - g(h,\theta')\| \leqslant K_{\mathscr{D},\mathscr{P}} \|\theta - \theta'\|.$$

Under Assumptions (A_5) and (A_6), the recurrence (2.13) almost surely converges towards the neural ODE given by

$$H_0 = Ax, \quad dH_t = \mathscr{V}_t g(H_t, \Theta_t) dt, \quad t \in [0,1], \tag{2.14}$$

as shown by the proposition below.

Proposition 2.11. *Consider model* (2.13) *such that Assumptions* (A_5) *and* (A_6) *are satisfied. Then the ODE* (2.14) *has a unique solution* H*, and, almost surely, there exists some* $c > 0$ *such that, for any* $0 \leqslant k \leqslant L$*,*

$$\|H_{k/L} - h_k\| \leqslant \frac{c}{L}.$$

Proof. See Section 2.A.10. □

It should be stressed that the transition from the discrete recurrence (2.13) to the continuous-time differential equation (2.14) relies on the assumptions that the weight sequences $(\theta_k)_{1\leqslant k\leqslant L}$ and $(V_k)_{1\leqslant k\leqslant L}$ are the discretizations of smooth limiting processes Θ and $\mathscr{V}$ on the one hand, and that the scaling α_L is chosen as $1/L$ on the other hand. From a practical perspective, Proposition 2.11 shows that it is possible to initialize ResNets in the ODE regime, by choosing a smooth stochastic process, discretizing it at each layer, and taking a $1/L$ scaling. This is in sharp contrast with the results of Sections 2.2 and 2.3.1, which show that the usual i.i.d. procedure leads to a neural SDE.

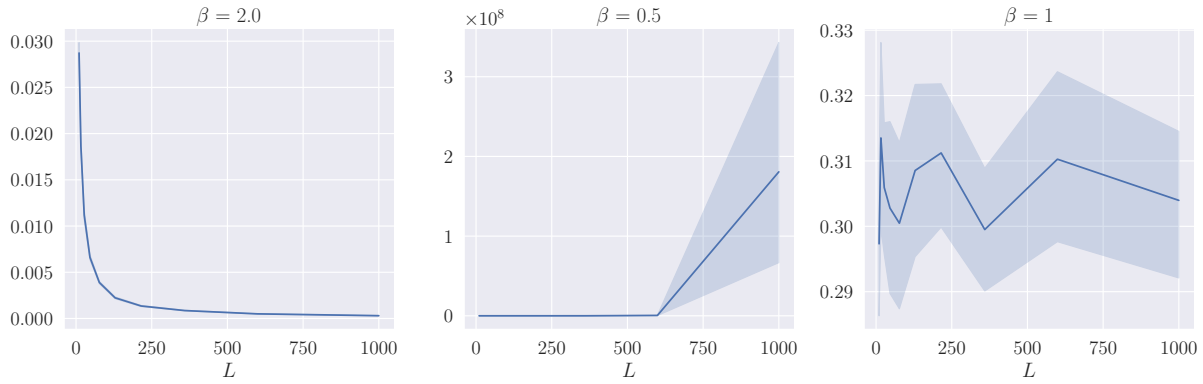


Figure 2.4: Evolution of $\|h_L - h_0\|/\|h_0\|$ as a function of L for different values of β and a smooth initialization of model `res-3`, with $d = 40$. The input is a random Gaussian observation x in dimension $n_{\text{in}} = 64$. The experiment is repeated with 50 independent randomizations.

Stability and scaling. Assuming that the weights of the network are discretizations of a smooth function (Assumption (A_5)), it is possible to obtain stability results, depending on the value of β, similarly to what has been done in Section 2.2. We show below that $\beta = 1$ is a critical value, by examining the hidden states, in the same way as $\beta = 1/2$ is a critical value in the i.i.d. setting. Similar results can be shown for the gradients. We begin by a proposition handling the cases $\beta > 1$ and $\beta = 1$.

Proposition 2.12. *Consider a ResNet* (2.4) *such that Assumptions* (A_5) *and* (A_6) *are satisfied. Let* $\alpha_L = 1/L^\beta$*, with* $\beta > 0$*.*

(i) If $\beta > 1$*, then, almost surely,*

$$\frac{\|h_L - h_0\|}{\|h_0\|} \xrightarrow{L\to\infty} 0.$$

(ii) If $\beta = 1$*, then, almost surely, there exists some* $c > 0$ *such that*

$$\frac{\|h_L - h_0\|}{\|h_0\|} \leqslant c.$$

Proof. See Section 2.A.11. □

The explosion case ($\beta < 1$) is more delicate to deal with. We prove it for a linear model, and leave for future work the extension to more general cases.

Proposition 2.13. *Consider the* `res-1` *model, taking* σ *as the identity function. Assume that Assumption* (A_5) *is satisfied and that* $\mathscr{V}_0^T$ *has a positive eigenvalue. Let* $\alpha_L = 1/L^\beta$*, with* $\beta \in (0, 1)$*. Then, almost surely,*

$$\max_k \frac{\|h_k - h_0\|}{\|h_0\|} \xrightarrow{L\to\infty} \infty.$$

Proof. See Section 2.A.12. □

The assumption of the existence of a positive eigenvalue for $\mathscr{V}_0^\top$ is mild. For instance, if the entries of $\mathscr{V}_0$ are i.i.d. random variables with finite moments of all order, Götze and Jalowy (2021) show that such an eigenvalue exists with probability at least $1 - 1/d$ for d large enough.

In this setting, we observe experimentally a behavior of the output and of the gradients when L grows large similar to the one explored in Section 2.2. This is illustrated in Figures 2.4 and 2.5,

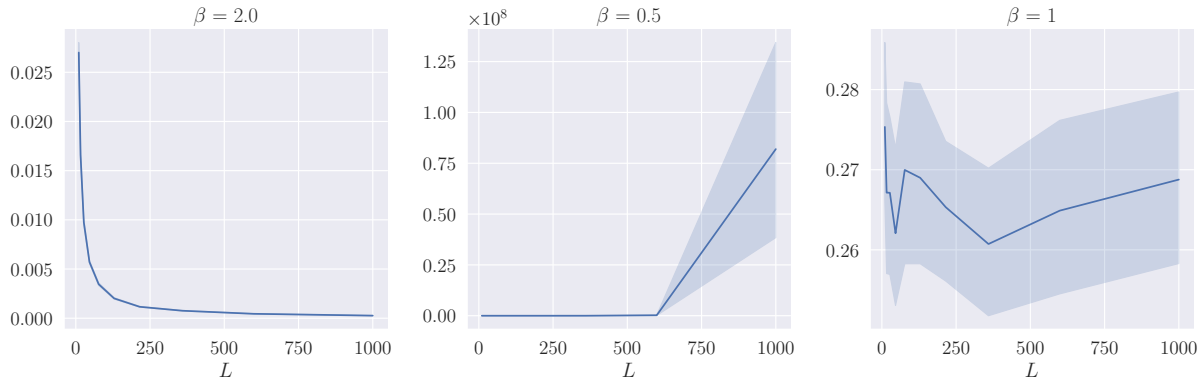


Figure 2.5: Evolution of $\|p_0 - p_L\|/\|p_L\|$ as a function of L for different values of β and a smooth initialization of model `res-3`, with $d = 40$. The input is a random Gaussian observation x in dimension $n_{\text{in}} = 64$. The experiment is repeated with 50 independent randomizations.

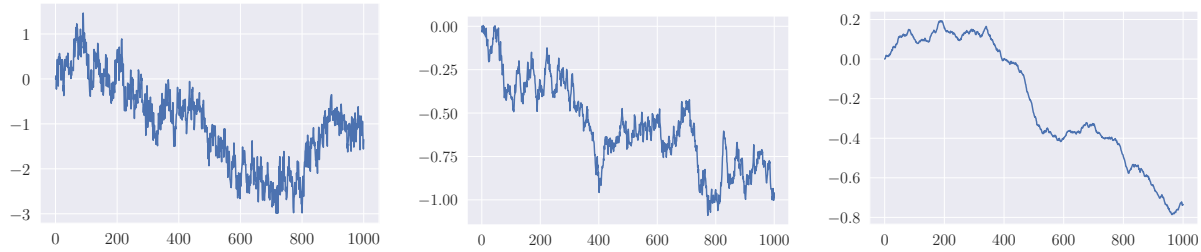
which mirror Figures 2.1 and 2.3 in Section 2.2. The figures clearly show that there exist three cases for the output and for the gradients: an identity case (left plots), an explosion case (middle), and a non-trivial case separating explosion and identity (right). However, the remarkable point is that the separation occurs for $\beta = 1$, and not $\beta = 1/2$, as predicted by Propositions 2.12 and 2.13.

2.4 Experiments

We experimentally investigate in this section two questions. The first one is to know whether there exists a range of scaling factors $\beta > 0$ and weight initializations, beyond the i.i.d. and the smooth regimes. The second question is whether our analysis, which pertains to the initialization phase, provides insights into the training phase, beyond initialization.

2.4.1 Intermediate regimes

In order to describe the transition between the i.i.d. and smooth cases, a possible route is to consider that the weights are increments of a γ-Hölder stochastic process. This model is interesting insofar as the Brownian motion (SDE regime) is $(1/2 - \varepsilon)$-Hölder ($\varepsilon > 0$) and a Lipschitz continuous stochastic process (ODE regime) is 1-Hölder.



(a) $H = 0.2$ (b) $H = 0.5$ (c) $H = 0.8$

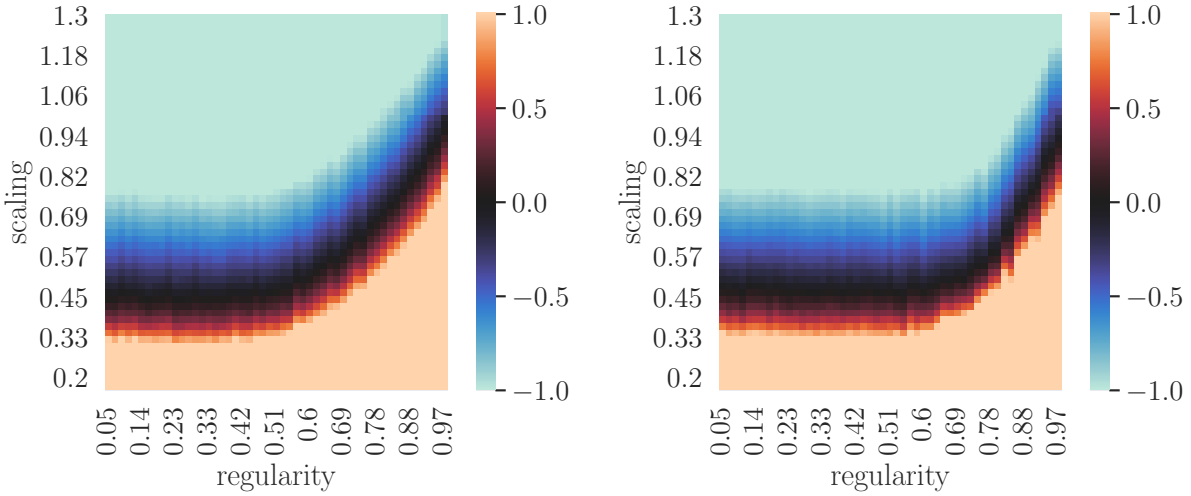
Figure 2.6: Examples of realizations of a fractional Brownian motion B^H for different Hurst indexes H. Note that the smaller the value of H, the more irregular the trajectory is.

In line with the above, in a series of experiments, we initialize the weights as increments of a fractional Brownian motion $(B_t^H)_{t\in[0,1]}$. Recall that B^H is a continuous-time Gaussian process,

starting at zero, with zero expectation for all $t \in [0,1]$, and covariance function

$$\mathbb{E}(B_s^H B_t^H) = \frac{1}{2}(|s|^{2H} + |t|^{2H} - |t-s|^{2H}), \quad 0 \leqslant s, t \leqslant 1,$$

where $H \in (0,1)$ is called the Hurst index. This index describes the raggedness of the process, with a higher value leading to a smoother process. When $H = 1/2$, the process is a standard Brownian motion (Definition 2.9), whose increments are independent by construction. When $H > 1/2$, the increments of the process are positively correlated, while if $H < 1/2$ the increments are negatively correlated. Importantly, a fractional Brownian motion with Hurst index H is $(H-\varepsilon)$-Hölder continuous for any $\varepsilon > 0$. In the limit when $H \to 1$, the trajectories converge to linear functions (whose increments satisfy (A_5)). As an illustration, Figure 2.6 depicts three realizations of a fractional Brownian motion with $H = 0.2$ (left), $H = 0.5$ (middle), and $H = 0.8$ (right).



(a) $\ln_{10}(\|h_L - h_0\|/\|h_0\|)$ (b) $\ln_{10}(\|p_0 - p_L\|/\|p_L\|)$

Figure 2.7: Magnitude of the outputs and of the gradients as a function of the regularity of the weights (Hurst index H) and of the scaling factor β. The orange zone corresponds to the explosion regime, i.e., $\|h_L - h_0\| \gg \|h_0\|$ and $\|p_0 - p_L\| \gg \|p_L\|$. The blue zone corresponds to the identity regime, i.e., $\|h_L - h_0\| \ll \|h_0\|$ and $\|p_0 - p_L\| \ll \|p_L\|$. Finally, the black zone is an intermediate regime, where $\|h_L - h_0\| \approx \|h_0\|$ and $\|p_0 - p_L\| \approx \|p_L\|$.

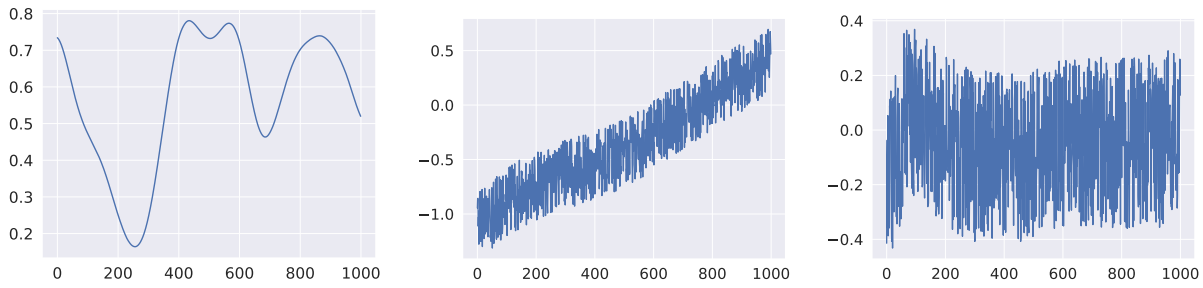
In order to assess the effect of the scaling factor β and the Hurst index H, we initialize a neural network `res-3` with $d = 40$, $L = 1000$, various values of $\beta \in [0.2, 1.3]$, and with weights taken as increments of fractional Brownian motions with various Hurst indices $H \in (0,1)$. Figure 2.7 depicts the empirical magnitude of the output and the gradients at initialization as a function of the Hurst index H and the scaling factor β. First note that we recover the two regimes (i.i.d. and smooth) discussed so far. For $H = 1/2$, the i.i.d. regime kicks in, with explosion ($\beta < 1/2$, orange zone), non-trivial behavior ($\beta = 1/2$, black zone), and identity ($\beta > 1/2$, blue zone). Likewise, we see at $H = 1$ a similar pattern in the smooth regime, with, as predicted by Proposition 2.12, a critical value $\beta = 1$. Beyond these two specific cases, we observe for an index H varying in $(1/2, 1)$ a whole range of intermediate situations, where the transition between identity and explosion seems to happen for a critical $\beta = H$. Interestingly, for $H < 1/2$, the transition seems to saturate at the value $\beta = 1/2$.

The take-home message is that the choice of the scaling of a ResNet seems to be closely linked to the regularity of the weights as a function of the layer. More precisely, for all regimes,

the critical scaling factor between explosion and identity seems to have a natural interpretation as the (Hölder) regularity of the underlying continuous-time stochastic process. We believe that the mathematical understanding of this connection, beyond the fractional Brownian motion case, is a promising research direction for the future. Finally, these experiments suggest that it is sensible to initialize a ResNet for any value of the scaling $\beta \in (1/2, 1)$, while avoiding the identity and explosion situations, by simulating a fractional Brownian motion of Hurst index $H = \beta$ and initializing the weights as the increments of this process.

2.4.2 Beyond initialization

At initialization, before the gradient descent, the distribution of the weights $(\theta_k)_{1 \leqslant k \leqslant L}$ and $(V_k)_{1 \leqslant k \leqslant L}$ is chosen by the practitioner. By contrast, during and after training, control is lost on these distributions, making the picture more complex. In particular, the existence and characterization of a continuous-time stochastic process whose discretization matches the trained ResNet is an interesting but difficult problem. Attacking this question requires a fine understanding of the interaction between training dynamics and the regularity of the sequence of the weights during the gradient descent. However, there is experimental evidence that the trained weights exhibit strong structure as a function of the layer index k (Cohen et al., 2021; Bayer et al., 2023), and that their regularity strongly depends on the choice of initialization. Figure 2.8 depicts this mechanism by plotting a given coordinate of θ_k as a function of the layer index k ranging from 1 to the depth $L = 1000$, after training. This will be further investigated in Chapter 3.

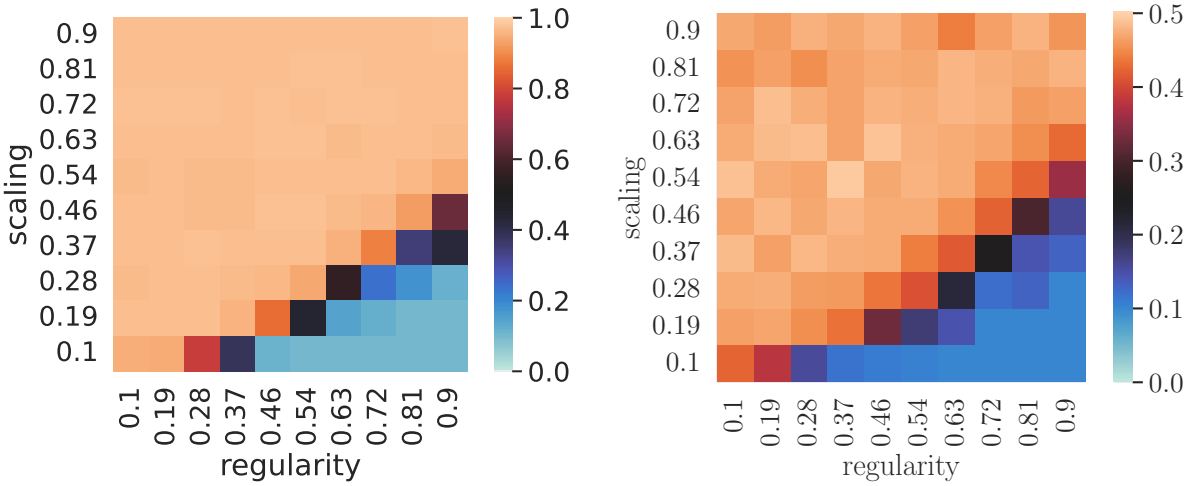


(a) $\beta = 1$, smooth initialization (b) $\beta = 1$, i.i.d. initialization (c) $\beta = 1/2$, i.i.d. initialization

Figure 2.8: Plot of a given coordinate of θ_k, after training, as a function of the layer index k ranging from 1 to the depth $L = 1000$ for three different choices of β and initializations.

To investigate the link between regularity of the weights at initialization, scaling, and performance after training, we train ResNets on the datasets MNIST (Deng, 2012) and CIFAR-10 (Krizhevsky, 2009). As in Subsection 2.4.1, we initialize the ResNets with various scaling factors and weights that are increments of fractional Brownian motions with different regularities. Then, for each combination of weight initialization and scaling factor, the ResNet is trained using the Adam optimizer (Kingma and Ba, 2015) for 10 epochs. The results in terms of accuracy are presented in Figure 2.9 (light orange = good performance, blue = bad performance). We observe a pattern similar to the one of Figure 2.7, however shifted downwards. This means that, for a given regularity, the network is unable to learn if it is initialized with a scaling too far below the critical value, which of course is connected with the gradient explosion issue discussed previously. On the other hand, and perhaps more surprisingly, the performance seems to be more or less stable in the identity region, with perhaps a small degradation in the case of CIFAR-10. This somewhat contrasts with the results from Yang and Schoenholz (2017), who exhibit a decrease in performance for i.i.d. initialization and a large scaling factor β. Note however that, in the

experiments reported in Figure 2.7, we adapt the learning rate of the gradient descent on a grid by cross-validation. When taking a fixed learning rate, we also observe a decrease in performance for large scaling factor β. The interplay between the learning rate and the scaling factor is one of the keys to better assess how the performance of the trained network is connected with the scaling.



(a) On MNIST

(b) On CIFAR-10

Figure 2.9: Accuracy after training as a function of the regularity of the weights at initialization and scaling. For each point of the heatmap, the model was trained on a grid of learning rates, and the best performance is shown.

2.A Proofs

Throughout the proofs, the i-th coordinate of a vector v is denoted by v_i. Similarly, the i-th row of a matrix M is denoted by M_i, and its (i,j)-th entry by M_{ij}.

2.A.1 Proof of Proposition 2.1

Statement (i) is clear (with $C = 1$) since, for any $h \in \mathbb{R}^d$,

$$\|\sigma(h)\|^2 \in \left[a^2\|h\|^2, b^2\|h\|^2\right] \subseteq \left[\frac{1}{2}\|h\|^2, \|h\|^2\right].$$

With respect to statement (ii), it is enough to show that for any $h \in \mathbb{R}^d$ and any random matrix W satisfying the assumptions of the proposition, one has

$$\frac{\|h\|^2}{2} \leqslant \mathbb{E}\big(\|\sigma(Wh)\|^2\big) \leqslant \|h\|^2 \quad \text{and} \quad \mathbb{E}\big(\|\sigma(Wh)\|^8\big) \leqslant C\|h\|^8,$$

as well as

$$\mathbb{E}\big(\|\operatorname{ReLU}(Wh)\|^2\big) = \frac{\|h\|^2}{2} \quad \text{and} \quad \mathbb{E}\big(\|\operatorname{ReLU}(Wh)\|^8\big) \leqslant C\|h\|^8.$$

The two claims with the squared norms are consequences of Lemmas 2.16 and 2.17 in Appendix 2.B, together with the fact that the variance of the entries of W equals $1/d$. In order to prove the other

two statements, first note that $\mathbb{E}(\|\sigma(Wh)\|^8) \leqslant \mathbb{E}(\|Wh\|^8)$ and $\mathbb{E}(\|\operatorname{ReLU}(Wh)\|^8) \leqslant \mathbb{E}(\|Wh\|^8)$. The results are then consequences of Lemma 2.21 in Appendix 2.C, which states that

$$\mathbb{E}\left(\frac{\|Wh\|^8}{\|h\|^8}\right) \leqslant 1 + \frac{384s^4}{d} + \frac{3072s^6}{d^2} \leqslant 1 + 384s^4 + 3072s^6.$$

2.A.2 Proof of Proposition 2.2

According to Lemma 2.14 below, one has

$$\mathbb{E}\Big(\frac{\|h_L - h_0\|^2}{\|h_0\|^2}\Big) \leqslant \big(\big(1+\alpha_L^2\big)^L - 1\big).$$

But, for $L\alpha_L^2 \leqslant 1$, we have $(1+\alpha_L^2)^L - 1 \leqslant \exp(L\alpha_L^2) - 1 \leqslant 2L\alpha_L^2$. Therefore,

$$\mathbb{E}\Big(\frac{\|h_L - h_0\|^2}{\|h_0\|^2}\Big) \leqslant 2L\alpha_L^2,$$

and the result follows from Markov's inequality.

Lemma 2.14. *Consider a ResNet* (2.4) *such that Assumptions* (A_1) *and* (A_2) *are satisfied. Then*

$$\left(\Big(1+\frac{\alpha_L^2}{2}\Big)^L - 1\right) \leqslant \mathbb{E}\Big(\frac{\|h_L - h_0\|^2}{\|h_0\|^2}\Big) \leqslant \Big(\big(1+\alpha_L^2\big)^L - 1\Big).$$

Proof (Lemma 2.14) Taking the squared norm of the forward update rule (2.4) and dividing by $\|h_0\|^2$ yields

$$\frac{\|h_{k+1}\|^2}{\|h_0\|^2} = \frac{1}{\|h_0\|^2}\Big(\|h_k\|^2 + \alpha_L^2\|V_{k+1}g(h_k,\theta_{k+1})\|^2 + 2\alpha_L\langle h_k, V_{k+1}g(h_k,\theta_{k+1})\rangle\Big). \tag{2.15}$$

We deduce by Assumptions (A_1) and (A_2) that

$$\Big(1+\frac{\alpha_L^2}{2}\Big)\mathbb{E}\Big(\frac{\|h_k\|^2}{\|h_0\|^2}\Big) \leqslant \mathbb{E}\Big(\frac{\|h_{k+1}\|^2}{\|h_0\|^2}\Big) \leqslant (1+\alpha_L^2)\mathbb{E}\Big(\frac{\|h_k\|^2}{\|h_0\|^2}\Big).$$

Therefore, by recurrence, we are led to

$$\Big(1+\frac{\alpha_L^2}{2}\Big)^k \leqslant \mathbb{E}\Big(\frac{\|h_k\|^2}{\|h_0\|^2}\Big) \leqslant (1+\alpha_L^2)^k. \tag{2.16}$$

Now, observe that $h_L = h_0 + \alpha_L \sum_{k=0}^{L-1} V_{k+1}g(h_k,\theta_{k+1})$. Thus, we have

$$\mathbb{E}\Big(\frac{\|h_L - h_0\|^2}{\|h_0\|^2}\Big) = \alpha_L^2 \sum_{k,k'=0}^{L-1} \mathbb{E}\Big(\frac{g(h_k,\theta_{k+1})^\top V_{k+1}^\top V_{k'+1} g(h_{k'},\theta_{k'+1})}{\|h_0\|^2}\Big).$$

By conditioning on all random variables except $V_{k'+1}$ for $k < k'$ (and V_{k+1} for $k > k'$), it is easy to see that the only non-zero terms are when $k = k'$. This yields

$$\begin{aligned}
\mathbb{E}\Big(\frac{\|h_L - h_0\|^2}{\|h_0\|^2}\Big) &= \alpha_L^2 \sum_{k=0}^{L-1} \mathbb{E}\Big(\frac{\|V_{k+1} g(h_k, \theta_{k+1})\|^2}{\|h_0\|^2}\Big) \\
&\leqslant \alpha_L^2 \sum_{k=0}^{L-1} \mathbb{E}\Big(\frac{\|h_k\|^2}{\|h_0\|^2}\Big) \\
&\quad \text{(by Assumptions } A_1 \text{ and } A_2\text{)} \\
&\leqslant \alpha_L^2 \sum_{k=0}^{L-1} (1+\alpha_L^2)^k \\
&\quad \text{(by (2.16))} \\
&= \big((1+\alpha_L^2)^L - 1\big).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathbb{E}\Big(\frac{\|h_L - h_0\|^2}{\|h_0\|^2}\Big) &\geqslant \frac{\alpha_L^2}{2} \sum_{k=0}^{L-1} \mathbb{E}\Big(\frac{\|h_k\|^2}{\|h_0\|^2}\Big) \\
&= \Big(\Big(1+\frac{\alpha_L^2}{2}\Big)^L - 1\Big).
\end{aligned}$$

■

2.A.3 Proof of Proposition 2.3

Dividing (2.15) by $\|h_k\|^2$ and taking the logarithm leads to

$$\ln(\|h_{k+1}\|^2) = \ln(\|h_k\|^2) + \ln\left(1 + \alpha_L^2 \frac{\|V_{k+1} g(h_k, \theta_{k+1})\|^2}{\|h_k\|^2} + 2\alpha_L \Big\langle \frac{h_k}{\|h_k\|}, \frac{V_{k+1} g(h_k, \theta_{k+1})}{\|h_k\|} \Big\rangle\right).$$

Let

$$Y_{k,1} = \alpha_L^2 \frac{\|V_{k+1} g(h_k, \theta_{k+1})\|^2}{\|h_k\|^2}, \quad Y_{k,2} = 2\alpha_L \Big\langle \frac{h_k}{\|h_k\|}, \frac{V_{k+1} g(h_k, \theta_{k+1})}{\|h_k\|} \Big\rangle,$$

and $Y_k = Y_{k,1} + Y_{k,2}$. The proof of Proposition 2.3 strongly relies on the following lemma, which provides technical information on the moments of $Y_{k,1}$ and $Y_{k,2}$. For the sake of clarity, its proof is postponed to Appendix 2.B.

Lemma 2.15. *Assume that Assumptions* (A_1) *and* (A_2) *are satisfied. Then*

(E_1) $\mathbb{E}(Y_{k,2}|h_k) = 0.$

(E_2) $\frac{\alpha_L^2}{2} \leqslant \mathbb{E}(Y_{k,1}|h_k) \leqslant \alpha_L^2.$

(E_3) $\mathbb{E}(Y_{k,1} Y_{k,2}|h_k) = 0.$

(E_4) $\mathbb{E}(Y_{k,2}^2|h_k) \leqslant 4\frac{\alpha_L^2}{d}.$

(E_5) $\mathbb{E}(Y_{k,2}^4|h_k) \leqslant 2048 \frac{s^4 \alpha_L^4}{d^2}.$

(E_6) $\mathbb{E}(Y_{k,1}^4|h_k) \leqslant C\left(3072\frac{s^6}{d^2} + 384\frac{s^4}{d} + 1\right)\alpha_L^8.$

(E_7) $\mathbb{E}(Y_{k,1}^2|h_k) \leqslant \sqrt{C}\left(128\frac{s^4}{d} + 1\right)\alpha_L^4.$

For $c > 0$, we have

$$\begin{aligned}
\mathbb{P}\Big(\frac{\|h_L\|^2}{\|h_0\|^2} \geqslant c\Big) &= \mathbb{P}\Big(\ln(\|h_L\|^2) - \ln(\|h_0\|^2) \geqslant \ln(c)\Big) \\
&= \mathbb{P}\Big(\sum_{k=0}^{L-1} \ln(1+Y_k) \geqslant \ln(c)\Big) \\
&\leqslant \mathbb{P}\Big(\sum_{k=0}^{L-1} Y_k \geqslant \ln(c)\Big) \\
&\quad \text{(using } \ln(1+x) \leqslant x \text{ for } x > -1\text{)}.
\end{aligned}$$

Let $S = \sum_{k=0}^{L-1} Y_k - \mathbb{E}(Y_k|h_k)$. By (E_1) and (E_2),

$$\sum_{k=0}^{L-1} \mathbb{E}(Y_k|h_k) \leqslant L\alpha_L^2.$$

So, for $c > \exp(L\alpha_L^2)$,

$$\begin{aligned}
\mathbb{P}\Big(\frac{\|h_L\|^2}{\|h_0\|^2} \geqslant c\Big) &\leqslant \mathbb{P}\Big(S \geqslant \ln(c) - \sum_{k=0}^{L-1} \mathbb{E}(Y_k|h_k)\Big) \\
&\leqslant \mathbb{P}(S \geqslant \ln(c) - L\alpha_L^2) \\
&\leqslant \mathbb{P}\Big(S^2 \geqslant \big(\ln(c) - L\alpha_L^2\big)^2\Big) \\
&\leqslant \frac{\mathbb{E}(S^2)}{(\ln(c) - L\alpha_L^2)^2} \qquad (2.17) \\
&\quad \text{(by Markov's inequality.)}
\end{aligned}$$

It remains to upper bound $\mathbb{E}(S^2)$. To this aim, note that

$$\begin{aligned}
\mathbb{E}(S^2) = \sum_{k=0}^{L-1} \mathbb{E}\Big(\big(Y_k - \mathbb{E}(Y_k|h_k)\big)^2\Big) &\leqslant \sum_{k=0}^{L-1} \mathbb{E}\big(Y_k^2\big) \\
&\leqslant 4\frac{L\alpha_L^2}{d} + 128\sqrt{C}\frac{L\alpha_L^4 s^4}{d} + \sqrt{C}L\alpha_L^4 \\
&\quad \text{(by } (E_3)\text{, } (E_4)\text{, and } (E_7)\text{)} \\
&\leqslant 5\frac{L\alpha_L^2}{d}.
\end{aligned}$$

The last inequality is true for $\alpha_L^2 \leqslant \frac{1}{\sqrt{C}(d+128s^4)}$. Therefore, by inequality (2.17), we obtain, for $c > \exp(L\alpha_L^2)$,

$$\mathbb{P}\Big(\frac{\|h_L\|^2}{\|h_0\|^2} \geqslant c\Big) \leqslant \frac{5L\alpha_L^2}{d\,\big(\ln(c) - L\alpha_L^2\big)^2}.$$

We conclude that, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\frac{\|h_L\|^2}{\|h_0\|^2} < \exp\left(L\alpha_L^2 + \sqrt{\frac{5L\alpha_L^2}{d\delta}}\right).$$

This shows statement (ii) of the proposition.

Next, to prove statement (i), observe that $c > 0$,

$$\begin{aligned}\mathbb{P}\Big(\frac{\|h_L\|^2}{\|h_0\|^2} \leqslant c\Big) &= \mathbb{P}\Big(\ln(\|h_L\|^2) - \ln(\|h_0\|^2) \leqslant \ln(c)\Big)\\ &= \mathbb{P}\Big(\sum_{k=0}^{L-1} \ln(1+Y_k) \leqslant \ln(c)\Big)\\ &= \mathbb{P}\Big(\sum_{k=0}^{L-1} \ln(1+Y_k) \leqslant \ln(c) \text{ and } \forall k, Y_k \geqslant -\frac{1}{2}\Big)\\ &\quad + \mathbb{P}\Big(\sum_{k=0}^{L-1} \ln(1+Y_k) \leqslant \ln(c) \text{ and } \exists k, Y_k < -\frac{1}{2}\Big).\end{aligned}$$

Using the inequality $\ln(1+x) \geqslant x - x^2$ for $x \geqslant -1/2$, we obtain

$$\begin{aligned}\mathbb{P}\Big(\frac{\|h_L\|^2}{\|h_0\|^2} \leqslant c\Big) &\leqslant \mathbb{P}\Big(\sum_{k=0}^{L-1} Y_k - Y_k^2 \leqslant \ln(c) \text{ and } \forall k, Y_k \geqslant -\frac{1}{2}\Big)\\ &\quad + \mathbb{P}\Big(\sum_{k=0}^{L-1} \ln(1+Y_k) \leqslant \ln(c) \text{ and } \exists k, Y_k < -\frac{1}{2}\Big).\end{aligned}$$

Thus,

$$\mathbb{P}\Big(\frac{\|h_L\|^2}{\|h_0\|^2} \leqslant c\Big) \leqslant \mathbb{P}\Big(\sum_{k=0}^{L-1} Y_k - Y_k^2 \leqslant \ln(c)\Big) + \sum_{k=0}^{L-1} \mathbb{P}\Big(Y_{k,2} < -\frac{1}{2}\Big). \tag{2.18}$$

We handle the two terms above on the right-hand side separately. For the first term, let $Z_k = Y_k - Y_k^2$ and $S = \sum_{k=0}^{L-1} Z_k - \mathbb{E}(Z_k|h_k)$. Observe that, by (E_1)-(E_4) and (E_7),

$$\sum_{k=0}^{L-1} \mathbb{E}(Z_k|h_k) \geqslant \frac{L\alpha_L^2}{2} - 4\frac{L\alpha_L^2}{d} - 128\sqrt{C}\frac{L\alpha_L^4 s^4}{d} - \sqrt{C}L\alpha_L^4 \geqslant \frac{3}{8}L\alpha_L^2, \tag{2.19}$$

where the last inequality is valid for $d \geqslant 64$ and $\alpha_L^2 \leqslant \frac{1}{16\sqrt{C}(2s^4+1)}$. Hence, for $0 < c < \exp(3L\alpha_L^2/8)$,

$$\begin{aligned}\mathbb{P}\Big(\sum_{k=0}^{L-1} Y_k - Y_k^2 \leqslant \ln c\Big) &= \mathbb{P}\Big(S \leqslant \ln(c) - \sum_{k=0}^{L-1} \mathbb{E}(Z_k|h_k)\Big)\\ &\leqslant \mathbb{P}\Big(S \leqslant \ln(c) - \frac{3L\alpha_L^2}{8}\Big)\\ &\leqslant \mathbb{P}\Big(S^2 \geqslant \Big(\ln(c) - \frac{3L\alpha_L^2}{8}\Big)^2\Big)\\ &\leqslant \frac{\mathbb{E}(S^2)}{\big(\ln(c) - \frac{3L\alpha_L^2}{8}\big)^2}\\ &\qquad \text{(by Markov's inequality.)}\end{aligned}$$

Using the c_r-inequality $(a+b)^n \leqslant 2^{n-1}(a^n + b^n)$ respectively for $n = 2$ and $n = 4$, we see that

$$\begin{aligned}\mathbb{E}(S^2) &= \sum_{k=0}^{L-1} \mathbb{E}\Big(\big(Z_k - \mathbb{E}(Z_k|h_k)\big)^2\Big) \leqslant \sum_{k=0}^{L-1} \mathbb{E}\big(Z_k^2\big) \leqslant 2\sum_{k=0}^{L-1} \mathbb{E}\big(Y_k^2\big) + \mathbb{E}\big(Y_k^4\big)\\ &\leqslant 2\sum_{k=0}^{L-1} \mathbb{E}(Y_{k,1}^2) + \mathbb{E}(Y_{k,2}^2) + 2\mathbb{E}(Y_{k,1}Y_{k,2}) + 8\mathbb{E}(Y_{k,1}^4) + 8\mathbb{E}(Y_{k,2}^4).\end{aligned}$$

By (E_3)-(E_7), it is easy to verify that, for $d \geqslant 64$ and $\alpha_L^2 \leqslant \frac{1}{(\sqrt{C}s^4/16+2\sqrt{C}+8s^4)d}$,

$$\mathbb{E}(S^2) \leqslant 10\frac{L\alpha_L^2}{d}.$$

This shows that, for $c < \exp(3L\alpha_L^2/8)$,

$$\mathbb{P}\Big(\sum_{k=0}^{L-1} Y_k - Y_k^2 \leqslant \ln(c)\Big) \leqslant \frac{10L\alpha_L^2}{d\big(\ln(c) - \frac{3L\alpha_L^2}{8}\big)^2}.$$

To conclude the proof, it remains to upper bound the second term of inequality (2.18). According to inequality (2.22) in the proof of Lemma 2.15 (with $t = 1/2$), one has

$$\sum_{k=0}^{L-1} \mathbb{P}\Big(Y_{k,2} < -\frac{1}{2}\Big) \leqslant 2L\exp\Big(-\frac{d}{64\alpha_L^2 s^2}\Big).$$

Putting everything together, we are led to

$$\mathbb{P}\Big(\frac{\|h_L\|^2}{\|h_0\|^2} \leqslant c\Big) \leqslant \frac{10L\alpha_L^2}{d\big(\ln(c) - \frac{3L\alpha_L^2}{8}\big)^2} + 2L\exp\Big(-\frac{d}{64\alpha_L^2 s^2}\Big).$$

Take $\delta \in (0,1)$. Then, if $2L\exp\big(-\frac{d}{64\alpha_L^2 s^2}\big) \leqslant \frac{\delta}{11}$, with probability at least $1-\delta$,

$$\frac{\|h_L\|^2}{\|h_0\|^2} > \exp\left(\frac{3L\alpha_L^2}{8} - \sqrt{\frac{11L\alpha_L^2}{d\delta}}\right).$$

Notice that this inequality is valid under the assumption $\alpha_L^2 \leqslant \frac{2}{(\sqrt{C}s^4+4\sqrt{C}+16s^4)d}$.

2.A.4 Proof of Corollary 2.4

Statement (i) is a consequence of Proposition 2.2, whereas (ii) is a consequence of Proposition 2.3 (i). The latter is valid under the conditions $d \geqslant 64$ and $\alpha_L \leqslant \frac{2}{(\sqrt{C}s^4+4\sqrt{C}+16s^4)d}$, which is automatically satisfied for all L large enough. Furthermore, an inspection of the proof of Proposition 2.3 reveals that the divergence in high probability of $\|h_L\|$ can be proved under the relaxed assumption $d \geqslant 9$. Indeed, the main constraint on d comes from the lower bound (2.19), where one needs to make sure that $\frac{L\alpha_L^2}{2} - 4\frac{L\alpha_L^2}{d} > 0$, which is the case for $d = 9$.

To prove (iii), we use a union bound on both statements of Proposition 2.3.

2.A.5 Proof of Proposition 2.5

The first claim follows from the observation that

$$\frac{\partial g(h_k, \theta_{k+1})}{\partial h} q_k = \begin{pmatrix} \sigma'(h_{k,1}) & 0 & \dots & 0 \\ 0 & \sigma'(h_{k,2}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma'(h_{k,d}) \end{pmatrix} q_k,$$

from (A_1), and from the assumption on σ'.

Let us now prove (ii). In the rest of the proof, the subscript k is ignored to lighten the notation. Observe that

$$\frac{\partial g(h,\theta)}{\partial h}q = V\begin{pmatrix}\sigma'(\langle W_1,h\rangle) & 0 & \dots & 0\\ 0 & \sigma'(\langle W_2,h\rangle) & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & \sigma'(\langle W_d,h\rangle)\end{pmatrix}Wq.$$

Denote by D the matrix in the middle of the right-hand side. Then

$$\mathbb{E}\Big(\Big\|\frac{\partial g(h,\theta)}{\partial h}q\Big\|^2\Big|h,q\Big) = \mathbb{E}\big(\|VDWq\|^2|h,q\big) = \mathbb{E}\big(\|DWq\|^2|h,q\big)$$
$$\text{(by } (A_1)\text{)}$$

For model `res-2`, we have

$$\mathbb{E}\Big(\Big\|\frac{\partial g(h,\theta)}{\partial h}q\Big\|^2\Big|h,q\Big) = \mathbb{E}\Big(\sum_{i=1}^d\Big(\sum_{j=1}^d W_{ij}q_j\Big)^2\sigma'(\langle W_i,h\rangle)\Big|h,q\Big).$$

The conclusion follows from the hypothesis that $a\leqslant\sigma'\leqslant b$ and $\mathbb{E}\big(\|Wq\|^2|q\big)=\|q\|^2$. For model `res-3`, we have

$$\mathbb{E}\Big(\Big\|\frac{\partial g(h,\theta)}{\partial h}q\Big\|^2\Big|h,q\Big) = \mathbb{E}\Big(\sum_{i=1}^d\Big(\sum_{j=1}^d W_{ij}q_j\Big)^2\mathbf{1}_{\sum_{j=1}^d W_{ij}h_j\geqslant 0}\Big|h,q\Big).$$

Since the $(W_{ij})_{1\leqslant i,j\leqslant d}$ are centered random variables, we conclude that

$$\mathbb{E}\Big(\Big\|\frac{\partial g(h,\theta)}{\partial h}q\Big\|^2\Big|h,q\Big) = \frac{1}{2}\mathbb{E}\Big(\sum_{i=1}^d\Big(\sum_{j=1}^d W_{ij}q_j\Big)^2\Big|q\Big) = \frac{1}{2}\mathbb{E}\big(\|Wq\|^2|q\big) = \frac{\|q\|^2}{2}.$$

2.A.6 Proof of Proposition 2.6

Letting $b = p_L/\|p_L\|$, as in Assumption (A_3), and taking expectation in (2.10), we obtain

$$\mathbb{E}\Big(\frac{\|p_0\|^2}{\|p_L\|^2}\Big) = \mathbb{E}(|b^\top q_L(z)|^2) = \frac{1}{d}\mathbb{E}(\|q_L(z)\|^2) \tag{2.20}$$
$$\text{(by } (A_3)\text{)}.$$

The rest of the proof is similar to the proof of Proposition 2.2. From (2.9), we have

$$\|q_{k+1}(z)\|^2 = \|q_k(z)\|^2 + \alpha_L^2\Big\|V_{k+1}\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\|^2 + 2\alpha_L\Big\langle q_k(z), V_{k+1}\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\rangle.$$

By independence of V_{k+1} from $q_k(z)$ and $\frac{\partial g(h_k,\theta_{k+1})}{\partial h}$,

$$\mathbb{E}\Big(\Big\langle q_k(z), V_{k+1}\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\rangle\Big) = 0.$$

Next,

$$\begin{aligned}\mathbb{E}\Big(\Big\|V_{k+1}\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\|^2\Big) &= \mathbb{E}\bigg(\mathbb{E}\Big(\Big\|V_{k+1}\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\|^2\Big|h_k,\theta_{k+1},q_k(z)\Big)\bigg)\\ &= \mathbb{E}\left(\Big\|\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\|^2\right)\\ &\quad\text{(by }(A_1)\text{)}\\ &= \mathbb{E}\left(\mathbb{E}\left(\Big\|\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\|^2\Big|h_k,q_k(z)\right)\right).\end{aligned}$$

By Assumption (A_3), we are led to

$$\big(1+\frac{1}{2}\alpha_L^2\big)\mathbb{E}(\|q_k(z)\|^2)\leqslant\mathbb{E}(\|q_{k+1}(z)\|^2)\leqslant(1+\alpha_L^2)\mathbb{E}(\|q_k(z)\|^2),$$

and thus, by induction, since $q_0(z)=z$ and $\mathbb{E}(\|z\|^2)=d$,

$$d\big(1+\frac{1}{2}\alpha_L^2\big)^k\leqslant\mathbb{E}(\|q_k(z)\|^2)\leqslant d(1+4\alpha_L^2)^k.$$

In particular, for $k=L$,

$$d\big(1+\frac{1}{2}\alpha_L^2\big)^L\leqslant\mathbb{E}(\|q_L(z)\|^2)\leqslant d(1+\alpha_L^2)^L.$$

Therefore, by (2.20),

$$\big(1+\frac{1}{2}\alpha_L^2\big)^L\leqslant\mathbb{E}\bigg(\frac{\|p_0\|^2}{\|p_L\|^2}\bigg)\leqslant(1+\alpha_L^2)^L.$$

To finish the proof, observe that

$$\frac{1}{\|p_L\|}(p_0-p_L)^\top z=b^\top(q_L(z)-z).$$

Using arguments similar to (2.20), we may write

$$\mathbb{E}\bigg(\frac{\|p_0-p_L\|^2}{\|p_L\|^2}\bigg)=\frac{1}{d}\mathbb{E}\Big(\frac{\|q_L(z)-z\|^2}{\|z\|^2}\Big).$$

Now, upon noting that $q_L(z)-z=q_L(z)-q_0(z)=\alpha_L\sum_{k=0}^{L-1}V_{k+1}\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)$,

$$\begin{aligned}\mathbb{E}(\|q_L(z)-z\|^2) &= \alpha_L^2\sum_{k,k'=0}^{L-1}\mathbb{E}\left(q_k(z)^\top\frac{\partial g(h_k,\theta_{k+1})^\top}{\partial h}V_{k+1}^\top V_{k'+1}\frac{\partial g(h'_k,\theta_{k'+1})}{\partial h}q_{k'}(z)\right)\\ &= \alpha_L^2\sum_{k=0}^{L-1}\mathbb{E}\left(\Big\|V_{k+1}\frac{\partial g(h_k,\theta_{k+1})}{\partial h}q_k(z)\Big\|^2\right)\\ &\leqslant d\alpha_L^2\sum_{k=0}^{L-1}(1+\alpha_L^2)^k\\ &= d\big((1+\alpha_L^2)^L-1\big)\leqslant d\big(\exp(L\alpha_L^2)-1\big)\leqslant 2dL\alpha_L^2,\end{aligned}$$

for $L\alpha_L^2\leqslant 1$. Note that the second equality is obtained by conditioning on every random variable except $V_{k'+1}$ for $k<k'$ (and V_{k+1} for $k>k'$). Finally, by using Markov's inequality, we conclude that, for any $\varepsilon>0$,

$$\mathbb{P}\big(\|p_0-p_L\|^2\geqslant\varepsilon\|p_L\|^2\big)\leqslant\frac{2L\alpha_L^2}{\varepsilon}.$$

2.A.7 Proof of Proposition 2.7

The proof of Proposition 2.6 reveals that

$$\mathbb{E}\Big(\frac{\|p_0 - p_L\|^2}{\|p_L\|^2}\Big) \leqslant (1+\alpha_L^2)^L - 1.$$

Using similar arguments, one has

$$\mathbb{E}\Big(\frac{\|p_0 - p_L\|^2}{\|p_L\|^2}\Big) = \frac{1}{d}\mathbb{E}\Big(\frac{\|q_L(z) - z\|^2}{\|z\|^2}\Big) \geqslant \alpha_L^2 \sum_{k=0}^{L-1} \big(1 + \frac{1}{2}\alpha_L^2\big)^k = \big(1 + \frac{1}{2}\alpha_L^2\big)^L - 1.$$

2.A.8 Proof of Corollary 2.8

The first statement is an immediate consequence of Proposition 2.6. The second one is a consequence of Proposition 2.7 and the fact that, for $\beta < 1$,

$$\Big(1 + \frac{1}{L^\beta}\Big)^L = \exp\Big(L \ln\Big(1 + \frac{1}{L^\beta}\Big)\Big) \sim \exp\big(L^{1-\beta}\big) \to \infty.$$

Finally, (*iii*) follows from Proposition 2.7.

2.A.9 Proof of Proposition 2.10

The proposition is a consequence of Kloeden and Platen (1992, Theorems 4.5.3 and 10.2.2) for the SDE

$$dH_t^\top = \sqrt{\frac{1}{d}}\sigma(H_t^\top)dB_t.$$

Letting $a(h,t) = 0$ and $b(h,t) = \sqrt{\frac{1}{d}}\sigma(h)$, we need to check the following assumptions:

(H_1) The functions $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are jointly measurable on $\mathbb{R}^d \times [0,1]$.

(H_2) There exists a constant $C_1 > 0$ such that, for any $x, y \in \mathbb{R}^d$, $t \in [0,1]$,

$$\|a(x,t) - a(y,t)\| + \|b(x,t) - b(y,t)\| \leqslant C_1\|x - y\|.$$

(H_3) There exists a constant $C_2 > 0$ such that, for any $x \in \mathbb{R}^d$, $t \in [0,1]$,

$$\|a(x,t)\| + \|b(x,t)\| \leqslant C_2(1 + \|x\|).$$

(H_4) $\mathbb{E}\big(\|H_0\|^2\big) < \infty$.

(H_5) There exists a constant $C_3 > 0$ such that, for any $x \in \mathbb{R}^d$, $s, t \in [0,1]$,

$$\|a(x,t) - a(x,s)\| + \|b(x,t) - b(x,s)\| \leqslant C_3(1 + \|x\|)|t - s|^{1/2}.$$

Assumptions (H_1), (H_4), and (H_5) readily follow from the definitions. Assumption (H_2) is true since σ is Lipschitz continuous, and (H_3) follows from

$$\|\sigma(x)\| \leqslant b\|x\| \leqslant \|x\| \leqslant 1 + \|x\|.$$

2.A.10 Proof of Proposition 2.11

Let $\psi : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ be defined for any $h \in \mathbb{R}^d$, $t \in [0,1]$, by $\psi(h,t) = \mathscr{V}_t g(h, \Theta_t)$. With this notation, the ODE (2.14) is equivalent to the initial value problem

$$dH_t = \psi(H_t, t)dt, \quad H_0 = Ax.$$

By Assumptions (A_5) and (A_6), ψ is Lipschitz continuous in its first argument, in the sense that there exists $K > 0$ such that, for all $h, h' \in \mathbb{R}^d, t \in [0,1]$,

$$\|\psi(h,t) - \psi(h',t)\| \leqslant K\|h - h'\|.$$

In addition, it is continuous in its second one. Thus, according to the Picard-Lindelöf theorem (Theorem 2.22 in Appendix 2.D), this is enough to show that the neural ODE (2.14) has a unique solution on $[0,1]$. Note that the solution H is continuous on $[0,1]$ and is therefore bounded by a constant $M > 0$.

In order to prove the approximation bound of Proposition 2.11, we start by proving that both ψ and H are Lipschitz continuous in t. Under (A_5) and (A_6), this is clear for ψ since H is bounded. Moreover, for any $[s,t] \subset [0,1]$, we have

$$\begin{aligned}\|H_t - H_s\| = \Big\| \int_s^t \psi(H_u, u)du \Big\| &\leqslant \int_s^t \|\psi(H_u, u)\|du \\ &\leqslant (t-s) \sup_{\substack{u\in[0,1] \\ h\in\mathbb{R}^d, \|h\|\leqslant M}} \|\psi(h,u)\|.\end{aligned}$$

Now, let K_1 and K_2 denote the Lipschitz constants of ψ (in both arguments) and H respectively, and, for any $0 \leqslant k \leqslant L$, let $t_k = k/L$. Then we have, for $k \geqslant 1$,

$$\begin{aligned}&\|H_{t_k} - h_k\| \\ &= \Big\|H_{t_{k-1}} + \int_{t_{k-1}}^{t_k} \psi(H_u, u)du - h_{k-1} - \frac{1}{L}\psi(h_{k-1}, t_{k-1})\Big\| \\ &\leqslant \|H_{t_{k-1}} - h_{k-1}\| + \int_{t_{k-1}}^{t_k} \|\psi(H_u, u) - \psi(h_{k-1}, t_{k-1})\|du \\ &\leqslant \|H_{t_{k-1}} - h_{k-1}\| + K_1 \int_{t_{k-1}}^{t_k} \|H_u - h_{k-1}\|du + K_1 \int_{t_{k-1}}^{t_k} |u - t_{k-1}|du \\ &\leqslant \Big(1 + \frac{K_1}{L}\Big)\|H_{t_{k-1}} - h_{k-1}\| + K_1 \int_{t_{k-1}}^{t_k} \|H_u - H_{t_{k-1}}\|du + K_1 \int_{t_{k-1}}^{t_k} |u - t_{k-1}|du \\ &\leqslant \Big(1 + \frac{K_1}{L}\Big)\|H_{t_{k-1}} - h_{k-1}\| + (K_2+1)K_1 \int_{t_{k-1}}^{t_k} |u - t_{k-1}|du \\ &= \Big(1 + \frac{K_1}{L}\Big)\|H_{t_{k-1}} - h_{k-1}\| + \frac{(K_2+1)K_1}{2L^2}.\end{aligned}$$

By recurrence, we obtain

$$\begin{aligned}\|H_{t_k} - h_k\| &\leqslant \sum_{j=0}^{k-1} \Big(1 + \frac{K_1}{L}\Big)^j \times \frac{(K_2+1)K_1}{2L^2} \leqslant L\Big(1 + \frac{K_1}{L}\Big)^L \times \frac{(K_2+1)K_1}{2L^2} \\ &\leqslant e^{K_1} \frac{(K_2+1)K_1}{2L},\end{aligned}$$

which concludes the proof.

2.A.11 Proof of Proposition 2.12

Starting from (2.4) and using Assumption (A_6), one easily obtains the existence of C_1 and C_2 (whose values depend on the realization of $\mathscr{V}$ and Θ) such that

$$\|h_{k+1}\| \leqslant (1 + C_1\alpha_L)\|h_k\| + C_2\alpha_L.$$

By recurrence,

$$\|h_{k+1}\| \leqslant (1 + C_1\alpha_L)^k\Big(\|h_0\| + \frac{C_2}{C_1}\Big).$$

Hence, using $\alpha_L \leqslant 1/L$,

$$\|h_{k+1}\| \leqslant \exp(C_1)\Big(\|h_0\| + \frac{C_2}{C_1}\Big).$$

Since g is Lipschitz continuous on compact sets, it is bounded on every ball of $\mathbb{R}^d \times \mathbb{R}^p$. The result is then a consequence of the identity

$$h_L - h_0 = \alpha_L \sum_{k=0}^{L-1} V_{k+1} g(h_k, \theta_{k+1}),$$

since we showed that each term in the sum is bounded by some constant $C_3 > 0$, independent of L and k. Hence we have that

$$\|h_L - h_0\| \leqslant C_3 L\alpha_L = C_3 L^{1-\beta},$$

yielding the results depending on the value of β.

2.A.12 Proof of Proposition 2.13

In the linear case, (2.4) can be written

$$h_{k+1} = h_k + \alpha_L V_{k+1} h_k, \quad 0 \leqslant k \leqslant L-1.$$

Take y a unit-norm eigenvector of $\mathscr{V}_0^\top$ with associated eigenvalue $\lambda > 0$. Then

$$\begin{aligned}
\langle h_{k+1}, y\rangle &= \langle h_k + \alpha_L V_{k+1} h_k, y\rangle \\
&= \langle h_k, y\rangle + \alpha_L \langle h_k, V_{k+1}^\top y\rangle \\
&= \langle h_k, y\rangle + \lambda\alpha_L \langle h_k, y\rangle + \alpha_L \langle h_k, (V_{k+1} - \mathscr{V}_0)^\top y\rangle.
\end{aligned}$$

Since $\mathscr{V}$ is Lipschitz and $V_{k+1} = \mathscr{V}_{k+1/L}$, there exists c such that $\|V_{k+1} - \mathscr{V}_0\| \leqslant c\frac{k+1}{L}$. Hence

$$|\langle h_{k+1}, y\rangle| \geqslant (1 + \lambda\alpha_L)|\langle h_k, y\rangle| - c\alpha_L \frac{k+1}{L}\|h_k\|.$$

Then, by recurrence,

$$\begin{aligned}
|\langle h_L, y\rangle| &\geqslant (1 + \lambda\alpha_L)^L |\langle h_0, y\rangle| - c\frac{\alpha_L}{L} \sum_{k=0}^{L-1} (k+1)(1 + \lambda\alpha_L)^k \|h_k\| \\
&\geqslant (1 + \lambda\alpha_L)^L |\langle h_0, y\rangle| - c\alpha_L (1 + \lambda\alpha_L)^L \max_k \|h_k\|.
\end{aligned}$$

Let $M = \frac{|\langle h_0, y\rangle|}{2c\alpha_L}$, and suppose that $\|h_k\| \leqslant M$ for all $0 \leqslant k \leqslant L$. Then

$$\begin{aligned}\|h_L\| &\geqslant |\langle h_L, y\rangle| \\ &\qquad \text{(by the Cauchy-Schwartz inequality)} \\ &\geqslant (1+\lambda\alpha_L)^L\big(|\langle h_0, y\rangle| - cM\alpha_L\big).\end{aligned}$$

Then, for $\lambda\alpha_L \leqslant 1$,

$$\|h_L\| \geqslant \frac{1}{2}(1+\lambda\alpha_L)^L|\langle h_0, y\rangle| \geqslant \frac{1}{2}\exp\big(\frac{\lambda L\alpha_L}{2}\big)|\langle h_0, y\rangle|.$$

Thus, since $L\alpha_L = L^{1-\beta}$, we have that $\|h_L\| \to \infty$, which contradicts our assumption that $\|h_k\| \leqslant M$ for all $0 \leqslant k \leqslant L$. We deduce that, for all L large enough,

$$\max_k \|h_k\| > \frac{|\langle h_0, y\rangle|}{2c\alpha_L} \xrightarrow{L\to\infty} \infty.$$

Furthermore,

$$\max_k \frac{\|h_k - h_0\|}{\|h_0\|} > \frac{|\langle h_0, y\rangle|}{2c\|h_0\|\alpha_L} - 1 \xrightarrow{L\to\infty} \infty.$$

2.B Technical results

2.B.1 Lemmas 2.16 and 2.17

Lemma 2.16. *Let $W \in \mathbb{R}^{d\times d}$ be a matrix whose entries are centered i.i.d. random variables, with finite variance, and let σ be an activation function such that, for all $x \in \mathbb{R}, a|x| \leqslant |\sigma(x)| \leqslant b|x|$, $1/\sqrt{2} \leqslant a < b \leqslant 1$. Then, for any $x \in \mathbb{R}^d$,*

$$\frac{1}{2}\mathbb{E}\big(\|Wx\|^2\big) \leqslant \mathbb{E}\big(\|\sigma(Wx)\|^2\big) \leqslant \mathbb{E}\big(\|Wx\|^2\big) \quad \text{and} \quad \mathbb{E}\big(\|\operatorname{ReLU}(Wx)\|^2\big) = \frac{1}{2}\mathbb{E}\big(\|Wx\|^2\big).$$

Proof. The first part is a consequence of the assumption on σ. To prove the equality, let $X_i = \sum_{j=1}^d W_{ij}x_j$. Then

$$\mathbb{E}\big(\|\operatorname{ReLU}(Wx)\|^2\big) = \mathbb{E}\Big(\sum_{i=1}^d \Big(\sum_{j=1}^d W_{ij}x_j\Big)^2 \mathbf{1}_{\sum_{j=1}^d W_{ij}x_j \geqslant 0}\Big) = \mathbb{E}\Big(\sum_{i=1}^d X_i^2 \mathbf{1}_{X_i\geqslant 0}\Big).$$

Since the $(W_{ij})_{1\leqslant j\leqslant d}$ are centered and independent random variables, X_i is also centered. Hence $\mathbb{E}(X_i^2\mathbf{1}_{X_i\geqslant 0}) = 1/2\mathbb{E}(X_i^2)$, which concludes the proof. □

Lemma 2.17. *Let $W \in \mathbb{R}^{d\times d}$ be a matrix whose entries are centered i.i.d. random variables, with finite variance s^2. Then, for any $x \in \mathbb{R}^d$, $\mathbb{E}\big(\|Wx\|^2\big) = s^2 d\|x\|^2$.*

Proof. For any $1 \leqslant i \leqslant d$,

$$|Wx|_i^2 = \Big(\sum_{j=1}^d W_{ij}x_j\Big)^2 = \sum_{j,j'=1}^d W_{ij}W_{ij'}x_jx_{j'}.$$

Thus, by independence,

$$\mathbb{E}\big(|Wx|_i^2\big) = \mathbb{E}\Big(\sum_{j,j'=1}^d W_{ij}W_{ij'}x_jx_{j'}\Big) = \sum_{j=1}^d \mathbb{E}(W_{ij}^2)x_j^2 = s^2\|x\|^2. \tag{2.21}$$

The result follows by summing over all $i \in \{1,\dots,d\}$. □

2.B.2 Proof of Lemma 2.15

(E_1) and (E_2) are simple consequences of Assumptions (A_1) and (A_2).

To prove (E_3), let $f(h_k, \theta_{k+1}) = V_{k+1} g(h_k, \theta_{k+1})$. Then

$$\begin{aligned}\mathbb{E}(Y_{k,2} Y_{k,1} | h_k) &= \frac{1}{\|h_k\|^4} \mathbb{E}\big(\|f(h_k, \theta_{k+1})\|^2 \langle h_k, f(h_k, \theta_{k+1})\rangle \big| h_k\big) \\ &= \mathbb{E}\Big(\sum_{i=1}^d \sum_{j=1}^d f(h_k, \theta_{k+1})_i^2 (h_k)_j f(h_k, \theta_{k+1})_j \Big| h_k\Big).\end{aligned}$$

It is easy to verify that, under Assumption (A_1), each term of the sum above has zero expectation. This shows (E_3).

To establish (E_4), we start by noting that

$$\begin{aligned}\mathbb{E}\Big(\Big\langle \frac{h_k}{\|h_k\|}, \frac{f(h_k, \theta_{k+1})}{\|h_k\|} \Big\rangle^2 \Big| h_k\Big) &= \frac{1}{\|h_k\|^4} \mathbb{E}\big(h_k^\top f(h_k, \theta_{k+1}) f(h_k, \theta_{k+1})^\top h_k | h_k\big) \\ &= \frac{1}{\|h_k\|^4} h_k^\top \mathbb{E}\big(f(h_k, \theta_{k+1}) f(h_k, \theta_{k+1})^\top | h_k\big) h_k.\end{aligned}$$

Clearly, $\mathbb{E}(f(h_k, \theta_{k+1})_i f(h_k, \theta_{k+1})_j) = 0$ for $i \neq j$. Since, furthermore, the coordinates of $f(h_k, \theta_{k+1})$ are identically distributed conditionally on h_k, we obtain

$$\mathbb{E}\big(f(h_k, \theta_{k+1}) f(h_k, \theta_{k+1})^\top | h_k\big) = \frac{1}{d} \mathbb{E}(\|f(h_k, \theta_{k+1})\|^2 | h_k) I_d.$$

Thus,

$$\mathbb{E}\Big(\Big\langle \frac{h_k}{\|h_k\|}, \frac{f(h_k, \theta_{k+1})}{\|h_k\|} \Big\rangle^2 \Big| h_k\Big) = \frac{1}{d\|h_k\|^4} \mathbb{E}(\|f(h_k, \theta_{k+1})\|^2 | h_k) h_k^\top h_k \leqslant \frac{1}{d},$$

by Assumptions (A_1) and (A_2).

To prove (E_5), let $\varphi = \frac{\langle V_{k+1} g(h_k, \theta_{k+1}), h_k\rangle}{\|g(h_k, \theta_{k+1})\| \|h_k\|}$. Then, for any $t > 0$,

$$\begin{aligned}\mathbb{P}(|Y_{k,2}| > t) &= \mathbb{P}\Big(|\varphi| > \frac{t\|h_k\|}{2\alpha_L \|g(h_k, \theta_{k+1})\|}\Big) \\ &= \mathbb{E}\Big(\mathbb{P}\Big(|\varphi| > \frac{t\|h_k\|}{2\alpha_L \|g(h_k, \theta_{k+1})\|} \Big| h_k, \theta_{k+1}\Big)\Big).\end{aligned}$$

So, by Lemma 2.20 in Appendix 2.C,

$$\begin{aligned}\mathbb{P}(|Y_{k,2}| > t) &\leqslant \mathbb{E}\Big(2\exp\Big(- \frac{dt^2 \|h_k\|^2}{16\alpha_L^2 s^2 \|g(h_k, \theta_{k+1})\|^2}\Big)\Big) \\ &= \mathbb{E}\Big(\mathbb{E}\Big(2\exp\Big(- \frac{dt^2 \|h_k\|^2}{16\alpha_L^2 s^2 \|g(h_k, \theta_{k+1})\|^2}\Big) \Big| h_k\Big)\Big) \\ &\leqslant \mathbb{E}\Big(2\exp\Big(- \frac{dt^2 \|h_k\|^2}{16\alpha_L^2 s^2 \mathbb{E}(\|g(h_k, \theta_{k+1})\|^2 | h_k)}\Big)\Big),\end{aligned}$$

by Jensen's inequality. Finally, using Assumption (A_2), we deduce that

$$\mathbb{P}(|Y_{k,2}| > t) \leqslant \mathbb{E}\Big(2\exp\Big(- \frac{dt^2}{16\alpha_L^2 s^2}\Big)\Big) = 2\exp\Big(- \frac{dt^2}{16\alpha_L^2 s^2}\Big). \tag{2.22}$$

In particular, for all $q \geqslant 1$ (see, e.g., Pauwels, 2020),

$$\mathbb{E}(Y_{k,2}^{2q}) \leqslant q! \Big(\frac{32 s^2 \alpha_L^2}{d}\Big)^q.$$

The result is obtained by taking $q = 2$.

Finally, (E_6) and (E_7) are consequences of Lemma 2.21 in Appendix 2.C.

2.C Concentration of sub-Gaussian random matrices

In this appendix, we are interested in the concentration of linear and quadratic forms of sub-Gaussian matrices (Lemma 2.20 and Lemma 2.21). These two propositions are byproducts of the main result of Kontorovich (2014), which generalizes McDiarmid's inequality to sub-Gaussian variables. We start by a technical result regarding the sub-Gaussian diameter introduced by Kontorovich (2014), whose definition is recalled below.

Definition 2.18. *Let X be a real-valued random variable, X' an independent copy of X, and ε a Rademacher random variable, independent of X and X'. Then the sub-Gaussian diameter of X is defined as the smallest t such that $\varepsilon|X - X'|$ is t^2 sub-Gaussian.*

Lemma 2.19. *Let X be an s^2 sub-Gaussian centered random variable. Then the sub-Gaussian diameter of X is less than $\sqrt{2}s$.*

Proof. Let $\lambda \in \mathbb{R}$. Then, using the notation of Definition 2.18, one has

$$\begin{aligned}\mathbb{E}(\exp^{\lambda\varepsilon|X-X'|}) &= \mathbb{E}(\exp^{\lambda(X-X')}\,\mathbf{1}_{\varepsilon=1}) + \mathbb{E}(\exp^{\lambda\varepsilon(X'-X)}\,\mathbf{1}_{\varepsilon=-1})\\ &= \mathbb{E}(\exp^{\lambda(X-X')})\\ &= \mathbb{E}(\exp^{\lambda X})^2\\ &\leqslant \exp^{2\lambda^2 s^2},\end{aligned}$$

where the last equality is a consequence of the symmetry of X about 0. □

We are now ready to prove the two main results of this appendix.

Lemma 2.20 (Bound on the deviation of linear forms)**.** *Let V be a $\mathbb{R}^{d\times d}$ matrix whose entries are i.i.d s^2/d sub-Gaussian random variables. Then, for any $x, y \in \mathbb{R}^d$, $x, y \neq 0$,*

$$\mathbb{P}\Big(\frac{\langle Vx, y\rangle}{\|x\|\|y\|} \geqslant t\Big) \leqslant 2\exp\Big(-\frac{dt^2}{4s^2}\Big).$$

Proof. For any $1 \leqslant i, j \leqslant d$, set $X_{ij} = \frac{x_i V_{ij} y_j}{\|x\|\|y\|}$. Let $\mathcal{X} = \mathbb{R}^{d^2}$ endowed with the ℓ_1 norm, let X be the vector in $\mathcal{X}$ whose $(id+j)$-th coordinate is X_{ij}, and let the function φ be defined by

$$\varphi : \mathcal{X} \ni Y \longmapsto \sum_{i=1}^{d^2} Y_i.$$

By the triangle inequality, φ is a Lipschitz continuous function, with Lipschitz constant equal to 1. Observe also that X_{ij} is a centered $x_i^2 s^2 y_j^2/d\|x\|^2\|y\|^2$ sub-Gaussian random variable. Thus, according to Lemma 2.19, the sub-Gaussian diameter of X_{ij} is less than $\sqrt{2}x_i s y_j/\sqrt{d}\|x\|\|y\|$. By Kontorovich (2014, Theorem 1), for any $t > 0$, one has

$$\mathbb{P}\left(\varphi(X) \geqslant t\right) \leqslant 2\exp\left(-\frac{t^2}{2\sum_{i,j=1}^{d}\frac{2s^2x_i^2y_j^2}{d\|x\|^2\|y\|^2}}\right),$$

that is

$$\mathbb{P}\left(\frac{\langle Vx, y\rangle}{\|x\|\|y\|} \geqslant t\right) \leqslant 2\exp\Big(-\frac{dt^2}{4s^2}\Big).$$

□

Lemma 2.21 (Bound of moments of quadratic forms)**.** *Let V be a $\mathbb{R}^{d\times d}$ matrix whose entries are i.i.d ${}^{s^2}\!/_{d}$ sub-Gaussian random variables, with variance ${}^{1}\!/_{d}$. Then, for any $x \in \mathbb{R}^d$, $x \neq 0$,*

$$\mathbb{E}\left(\frac{\|Vx\|^4}{\|x\|^4}\right) \leqslant 1 + \frac{128s^4}{d} \quad \text{and} \quad \mathbb{E}\left(\frac{\|Vx\|^8}{\|x\|^8}\right) \leqslant 1 + \frac{384s^4}{d} + \frac{3072s^6}{d^2}.$$

Proof. The proof is similar to the one of Lemma 2.20, with $X_{ij} = \frac{V_{ij}x_j}{\|x\|}$, $\mathcal{X} = \mathbb{R}^d$, and

$$\varphi_i : \mathcal{X} \ni X \mapsto \sum_{j=1}^{d} X_{ij}.$$

Each function φ_i is a Lipschitz continuous function, with Lipschitz constant equal to 1. Observe now that the random variable X_{ij} is ${}^{x_j^2 s^2}\!/_{d\|x\|^2}$ sub-Gaussian and centered. Thus, according to Lemma 2.19, the sub-Gaussian diameter of X_{ij} is less than ${}^{\sqrt{2}x_j s}\!/_{\sqrt{d}\|x\|}$. Therefore, according to Kontorovich (2014, Theorem 1), for any $t > 0$,

$$\mathbb{P}\left(\varphi_i(X) \geqslant t\right) \leqslant 2\exp\left(-\frac{t^2}{2\sum_{j=1}^{d}\frac{2s^2x_j^2}{d\|x\|^2}}\right),$$

that is

$$\mathbb{P}\left(\frac{|\langle V_i, x\rangle|}{\|x\|} \geqslant t\right) \leqslant 2\exp\left(-\frac{dt^2}{4s^2}\right).$$

Hence (see, e.g., Pauwels, 2020),

$$\mathbb{E}\left(\left(\frac{\langle V_i, x\rangle}{\|x\|}\right)^{2q}\right) \leqslant q!\left(\frac{8s^2}{d}\right)^q. \tag{2.23}$$

From identity (2.21) in the proof of technical Lemma 2.17, given in Appendix 2.B, we obtain that, for $q = 1$,

$$\mathbb{E}\left(\left(\frac{\langle V_i, x\rangle}{\|x\|}\right)^{2}\right) = \frac{1}{d}, \tag{2.24}$$

which is an improvement by a factor $8s^2$ over the previous upper bound. To conclude, it remains to conclude $\|Vx\|^4$ and $\|Vx\|^8$ with the $\langle V_i, x\rangle$. To do so, observe that

$$\|Vx\|^4 = \left(\sum_{i=1}^{d}\langle V_i, x\rangle^2\right)^2 = \sum_{\substack{i,j=1\\ i\neq j}}^{d}\langle V_i, x\rangle^2\langle V_j, x\rangle^2 + \sum_{i=1}^{d}\langle V_i, x\rangle^4.$$

Hence, by independence of the $(V_i)_{1\leqslant i\leqslant d}$,

$$\begin{aligned}\mathbb{E}\left(\frac{\|Vx\|^4}{\|x\|^4}\right) &= \sum_{\substack{i,j=1\\ i\neq j}}^{d}\mathbb{E}\left(\frac{\langle V_i, x\rangle^2}{\|x\|^2}\right)\mathbb{E}\left(\frac{\langle V_j, x\rangle^2}{\|x\|^2}\right) + \sum_{i=1}^{d}\mathbb{E}\left(\frac{\langle V_i, x\rangle^4}{\|x\|^4}\right)\\ &= d(d-1)\frac{1}{d^2} + d\frac{2(8s^2)^2}{d^2} \leqslant 1 + \frac{128s^4}{d}\\ &\qquad \text{(by (2.23) and (2.24))}\end{aligned}$$

Similarly,

$$\|Vx\|^8 = \left(\sum_{i=1}^{d}\langle V_i, x\rangle^2\right)^3 = \sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{d}\langle V_i, x\rangle^2\langle V_j, x\rangle^2\langle V_j, x\rangle^2 + \sum_{\substack{i,j=1\\ i\neq j}}^{d}\langle V_i, x\rangle^2\langle V_j, x\rangle^4 + \sum_{i=1}^{d}\langle V_i, x\rangle^8.$$

Hence,

$$\begin{aligned}\mathbb{E}\Big(\frac{\|Vx\|^8}{\|x\|^8}\Big) &= \sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{d} \mathbb{E}\Big(\frac{\langle V_i,x\rangle^2}{\|x\|^2}\Big)\mathbb{E}\Big(\frac{\langle V_j,x\rangle^2}{\|x\|^2}\Big)\mathbb{E}\Big(\frac{\langle V_k,x\rangle^2}{\|x\|^2}\Big)\\ &\quad + \sum_{\substack{i,j=1\\ i\neq j}}^{d} \mathbb{E}\Big(\frac{\langle V_i,x\rangle^4}{\|x\|^4}\Big)\mathbb{E}\Big(\frac{\langle V_j,x\rangle^2}{\|x\|^2}\Big) + \sum_{i=1}^{d}\mathbb{E}\Big(\frac{\langle V_i,x\rangle^8}{\|x\|^8}\Big)\\ &= d(d-1)(d-2)\frac{1}{d^2} + 3d(d-1)\frac{2(8s^2)^2}{d^3} + d\frac{6(8s^2)^3}{d^3}\\ &\leqslant 1 + \frac{384s^4}{d} + \frac{3072s^6}{d^2}.\end{aligned}$$

□

2.D A version of the Picard-Lindelöf theorem

Theorem 2.22. *Assume that $f : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is Lipschitz continuous in its first argument and continuous in its second one. Then, for any $z \in \mathbb{R}^d$, the initial value problem*

$$dH_t = f(H_t, t)dt, \quad H_0 = z, \tag{2.25}$$

admits a unique solution $H : [0,1] \to \mathbb{R}^d$.

Proof. Let $\mathscr{C}([s,t],\mathbb{R}^d)$ be the set of continuous functions from $[s,t]$ to $\mathbb{R}^d$. For any $[s,t] \subset [0,1]$, $\zeta \in \mathbb{R}^d$, let Ψ be the function

$$\begin{aligned}\Psi : \mathscr{C}([s,t],\mathbb{R}^d) &\to \mathscr{C}([s,t],\mathbb{R}^d)\\ Y &\mapsto \big(v \mapsto \zeta + \int_s^v f(Y_u,u)du\big).\end{aligned}$$

For any $Y, Y' \in \mathscr{C}([s,t],\mathbb{R}^d)$, $v \in [s,t]$, one has, denoting by K_f the Lipschitz constant of f in its first argument,

$$\begin{aligned}\|\Psi(Y)_v - \Psi(Y')_v\| &\leqslant \int_s^v \big\|\big(f(Y_u,u) - f(Y'_u,u)\big)du\big\|\\ &\leqslant \int_s^v K_f\|Y_u - Y'_u\|du\\ &\leqslant K_f\int_s^v \|Y - Y'\|_\infty du\\ &\leqslant K_f\|Y - Y'\|_\infty(t-s).\end{aligned}$$

This yields

$$\|\Psi(Y) - \Psi(Y')\|_\infty \leqslant K_f(t-s)\|Y - Y'\|_\infty,$$

which means that the function Ψ is Lipschitz continuous on $\mathscr{C}([s,t],\mathbb{R}^d)$ endowed with the supremum norm, with Lipschitz constant $K_f(t-s)$. So, on any interval $[s,t]$ of length smaller than $\delta = 1/2K_f$, the function Ψ is a contraction. Thus, by the Banach fixed-point theorem, for any initial value ζ, Ψ has a unique fixed point. Hence, there exists a unique solution to (2.25) on any interval of length δ with any initial condition. To obtain a solution on $[0,1]$ it is sufficient to concatenate these solutions. □

2.E Detailed experimental setting

Our code is available at `https://github.com/PierreMarion23/scaling-resnets`.

To obtain Figures 2.1 to 2.3, we initialize ResNets from `res-3` with the hyperparameters of Table 2.2.

Name	Value
d	40
n_{in}	64
n_{out}	1
L	10 to 1000
β	$0.25, 0.5, 1$
weight distribution	$\mathcal{U}(-\sqrt{3/d}, \sqrt{3/d})$
data distribution	standard Gaussian

Table 2.2: Hyperparameters of Figures 2.1 to 2.3

Each experiment is repeated 50 times, with independent data and weight sampling.

For Figures 2.4 and 2.5, we take the same hyperparameters except for β, which now takes values in $\{0.5, 1, 2\}$, and for the weight distribution. The weights are now initialized as discretizations of a Gaussian process. More precisely, each entry of $\mathscr{V}$ and Θ is an independent Gaussian process with zero mean and an RBF kernel of variance 10^{-2}.

To obtain Figure 2.7, we take the hyperparameters of Table 2.3.

Name	Value
d	40
n_{in}	64
n_{out}	1
L	1000
β	0.2 to 1.3
weight distribution	fractional Brownian motion with Hurst index from 0.05 to 0.97
data distribution	standard Gaussian

Table 2.3: Hyperparameters of Figure 2.7

More precisely, for each $1 \leqslant i, j \leqslant d$, we let $(V_{k+1,i,j})_{0 \leqslant k \leqslant L-1}$ be the increments of a fractional Brownian motion (fBm), where the various fBm involved are independent. The procedure is the same for θ.

In Figure 2.9, we use `res-1`, with the hyperparameters of Table 2.4.

We train on MNIST[1] and CIFAR-10[2] using the Adam optimizer (Kingma and Ba, 2015) for 10 epochs. The learning rate is divided by 10 after 5 epochs. The best performance on the learning rate grid is reported in the figure.

Figure 2.8 is obtained by plotting a random coordinate of θ_k, after training on MNIST.

[1] `http://yann.lecun.com/exdb/mnist`

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

Name	**Value**
d	30
L	1000
β	0.2 to 1.3
weight distribution	fractional Brownian motion with Hurst index from 0.05 to 0.97
learning rate grid	$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1$

Table 2.4: Hyperparameters of Figure 2.9

3

Implicit regularization of deep residual networks towards neural ODEs

In this chapter, we take a further step towards establishing a solid mathematical link between residual neural networks and neural ordinary differential equations (ODEs), by proving an implicit regularization of deep residual networks towards neural ODEs. Our result holds for nonlinear networks trained with gradient flow. We prove that if the network is initialized as a discretization of a neural ODE, then such a discretization holds throughout training. Our results are valid for a finite training time, and also as the training time tends to infinity provided that the network satisfies a Polyak-Łojasiewicz condition. Importantly, this condition holds for a family of residual networks where the residuals are two-layer perceptrons with an overparameterization in width that is only linear, and implies the convergence of the gradient flow to a global minimum of the loss. Our results are illustrated by numerical experiments.

Contents

3.1 Introduction

Residual networks are a successful family of deep learning models popularized by breakthrough results in computer vision (He et al., 2016a). The key idea of residual networks, namely the presence of skip connections, is now ubiquitous in deep learning, and can be found, for example, in Transformer models (Vaswani et al., 2017). The main advantage of skip connections is to allow successful training with depth of the order of a thousand layers, in contrast to vanilla neural networks, leading to significant performance improvements (e.g., Wang et al., 2022). This has motivated research on the properties of residual networks in the limit where the depth tends to infinity. One of the main explored directions is the neural ordinary differential equation (ODE) limit (Chen et al., 2018a).

To present the neural ODE principle, we first introduce the mathematical formalism of deep residual networks. We consider a single model throughout the chapter to simplify the exposition, but most of our results apply to more general models, as will be discussed later. We consider the formulation

$$h_{k+1} = h_k + \frac{1}{L\sqrt{m}} V_{k+1} \sigma\Big(\frac{1}{\sqrt{q}} W_{k+1} h_k\Big), \quad k \in \{0, \dots, L-1\}, \tag{3.1}$$

where L is the depth of the network, $h_k \in \mathbb{R}^q$ is the output of the k-th hidden layer, $V_k \in \mathbb{R}^{q \times m}$, $W_k \in \mathbb{R}^{m \times q}$ are the weights of the k-th layer, and σ is an activation function applied element-wise. Scaling with the square root of the width is classical, although it often appears as an equivalent condition on the variance at initialization (Glorot and Bengio, 2010; LeCun et al., 1998; He et al., 2015). We make the scaling factors explicit to have weights of magnitude $\mathcal{O}(1)$ independently of the width and the depth. The $1/L$ scaling factor is less common, but it is necessary for the correspondence with neural ODEs to hold. More precisely, if there exist Lipschitz continuous functions $\mathcal{V}$ and $\mathcal{W}$ such that $V_k = \mathcal{V}(k/L)$ and $W_k = \mathcal{W}(k/L)$, then the residual network (3.1) converges, as $L \to \infty$, to the ODE

$$\frac{dH}{ds}(s) = \frac{1}{\sqrt{m}} \mathcal{V}(s) \sigma\Big(\frac{1}{\sqrt{q}} \mathcal{W}(s) H(s)\Big), \quad s \in [0, 1], \tag{3.2}$$

where s is the continuous-depth version of the layer index. It is important to note that this correspondence holds for *fixed* limiting functions $\mathcal{V}$ and $\mathcal{W}$. This is especially true at initialization, for example by setting the V_k to zero and the W_k to weight-tied Gaussian matrices. In this case, the initial residual network is trivially equal to the neural ODE $\frac{dH}{ds}(s) = 0$. Of course, more sophisticated initialization choices are possible, as shown, e.g., in Chapter 2 and Sander et al. (2022b). However, regardless of an ODE structure at initialization, a more challenging question is that of the structure of the network *after* training. Since the weights are updated during training, there is no a priori guarantee that an ODE limit still holds after training, even if it does at initialization.

The question of a possible ODE structure for the trained network is not a mere technical one. In fact, it is important for at least three reasons. First, it gives a precise answer to the question of the connection between (trained) residual networks and neural ODEs, providing more solid ground to a common statement in the community that both can coincide in the large-depth limit (see, e.g., Haber and Ruthotto, 2017; E et al., 2019; Dong et al., 2020; Massaroli et al., 2020; Kidger, 2022). Second, it opens exciting perspectives for understanding residual networks. Indeed, if trained residual networks are discretizations of neural ODEs, then it is possible to apply results from neural ODEs to the large family of residual networks. In particular, from a theoretical point of view, the approximation capabilities of neural ODEs are well understood (Teshima et al., 2020; Zhang et al., 2020a) and it is relatively easy to obtain generalization bounds

for these models (see Hanson and Raginsky, 2022 and Chapter 4 of this manuscript). From a practical standpoint, advantages of neural ODEs include memory-efficient training (Chen et al., 2018a; Sander et al., 2022b) and weight compression (Queiruga et al., 2021). This is important because in practice memory is a bottleneck for training residual networks (Gomez et al., 2017). Finally, our analysis is a first step towards understanding the implicit regularization (Neyshabur et al., 2015b; Vardi, 2023) of gradient descent for deep residual networks, that is, characterizing the properties of the trained network among all minimizers of the empirical risk.

Throughout the document, it is assumed that the network is trained with gradient flow, which is a continuous analog of gradient descent. The parameters V_k are updated according to an ODE of the form $\frac{dV_k}{dt}(t) = -L\frac{\partial \ell}{\partial V_k}(t)$ for $t \geq 0$, where ℓ is an empirical risk (the exact mathematical context and assumptions are detailed in Section 3.3), and similarly for W_k. The scaling factor L is the counterpart of the factor $1/L$ in (3.1), and prevents exploding or vanishing gradients as L tends to infinity. Note that the gradient flow is defined with respect to a time index t different from the layer index s.

Contributions. Our first main contribution (Section 3.4.1) is to show that a neural ODE limit holds after training up to time t, i.e., there exists a function $\mathcal{V}(s,t)$ such that the residual network converges, as L tends to infinity, to the ODE

$$\frac{dH}{ds}(s) = \frac{1}{\sqrt{m}}\mathcal{V}(s,t)\sigma\Big(\frac{1}{\sqrt{q}}\mathcal{W}(s,t)H(s)\Big), \quad s \in [0,1].$$

This large-depth limit holds for any finite training time $t \geqslant 0$. However, the convergence of the optimization algorithm as t tends to infinity, which we refer to as the *long-time limit* to distinguish it from the large-depth limit $L \to \infty$, is not guaranteed without further assumptions, due to the non-convexity of the optimization problem. We attack the question (Section 3.4.2) when the width is large enough by proving a Polyak-Łojasiewicz (PL) condition, which is now state of the art in analyzing the properties of optimization algorithms for deep neural networks (Liu et al., 2022). The main assumption for our PL condition to hold is that the width m of the hidden layers should be greater than some constant times the number of data n. As a second main contribution, we show that the PL condition yields the long-time convergence of the gradient flow for residual networks with linear overparameterization. Finally, we prove the convergence with high probability in the long-time limit, namely the existence of functions $\mathcal{V}_\infty$ and $\mathcal{W}_\infty$ such that the discrete trajectory defined by the trained residual network (3.1) converges as *both* L and t tend to infinity to the solution of the neural ODE (3.2) with $\mathcal{V} = \mathcal{V}_\infty$ and $\mathcal{W} = \mathcal{W}_\infty$. In addition, our approach points out that this limiting ODE interpolates the training data. Finally, our results are illustrated by numerical experiments (Section 3.5).

Organization of the chapter. Section 3.2 presents some related work. We then move on to detail the mathematical context and notation in Section 3.3 before giving our main results in Section 3.4. Section 3.5 is devoted to numerical experiments. We conclude the main part of the chapter in Section 3.6. Then, in Section 3.A, we prove results on a more general residual network model that encompasses the one presented so far. These results are then instantiated in the specific case of the residual network (3.3) in Section 3.B, thus proving the results of the main part of the chapter. Section 3.C contains some lemmas that are useful for the proofs. We present in Section 3.D a counter-example showing that a residual network with the ReLU activation can move away from the neural ODE structure during training. Finally, Section 3.E presents some experimental details.

3.2 Related work

Deep residual networks and neural ODEs. Several works study the large-depth convergence of residual networks to differential equations, but without considering the training dynamics, such as Chapter 2 of this manuscript or Cohen et al. (2021); Thorpe and van Gennip (2022); Hayou (2023). Closer to our setting, Cont et al. (2022) and Sander et al. (2022b) analyze the dynamics of gradient descent for deep residual networks, as we do, but with significant differences. Cont et al. (2022) consider a $1/\sqrt{L}$ scaling factor in front of the residual branch, resulting in a limit that is not a neural ODE. In addition, only W is trained. Furthermore, to obtain convergence in the long-time limit, it is assumed that the data points are nearly orthogonal. Sander et al. (2022b) prove the existence of an ODE limit for trained residual networks, but in the simplified case of a linear activation and under a more restricted setting.

Long-time convergence of wide residual networks. Polyak-Łojasiewicz conditions are a modern tool to prove long-time convergence of overparameterized neural networks (Liu et al., 2022). These conditions are a relaxation of convexity, and mean that the gradients of the loss with respect to the parameters cannot be small when the loss is large. They have been applied to residual networks with both linear (Bartlett et al., 2018; Wu et al., 2019; Zou et al., 2020b) and nonlinear activations (Allen-Zhu et al., 2019; Frei et al., 2019; Barboni et al., 2022; Cont et al., 2022; MacDonald et al., 2022). Building on the proof technique of Nguyen and Mondelli (2020) for non-residual networks, we need only a linear overparameterization to prove our PL condition, i.e., we require $m = \Omega(n)$. This compares favorably with results requiring polynomial overparameterization (Allen-Zhu et al., 2019; Barboni et al., 2022) or assumptions on the data, either a margin condition (Frei et al., 2019) or a sample size smaller than the dimension of the data space (Cont et al., 2022; MacDonald et al., 2022).

Implicit regularization. This chapter can be related to a line of work on the implicit regularization of gradient-based algorithms for residual networks (Neyshabur et al., 2015b). We show that the optimization algorithm does not just converge to any residual network that minimizes the empirical risk, but rather to the discretization of a neural ODE. Note that most implicit regularization results state that the optimization algorithm converges to an interpolator that minimizes some complexity measure, which can be a margin (Lyu and Li, 2020), a norm (Boursier et al., 2022), or a matrix rank (Li et al., 2021b). Thus, an interesting next step is to understand if the neural ODE found by gradient flow actually minimizes some complexity measure, and to characterize its generalization properties.

3.3 Definitions and notation

This section is devoted to specifying the setup outlined in Section 3.1.

Residual network. A (scaled) residual network of depth $L \in \mathbb{N}^*$ is defined by

$$\begin{aligned} h_0^L &= A^L x \\ h_{k+1}^L &= h_k^L + \frac{1}{L\sqrt{m}} V_{k+1}^L \sigma\Big(\frac{1}{\sqrt{q}} W_{k+1}^L h_k^L\Big), \quad k \in \{0, \dots, L-1\}, \\ F^L(x) &= B^L h_L^L. \end{aligned} \tag{3.3}$$

To allow the hidden layers $h_k^L \in \mathbb{R}^q$ to have a different dimension than the input $x \in \mathbb{R}^d$, we first map x to h_0^L with a weight matrix $A^L \in \mathbb{R}^{q \times d}$. We assume that the hidden layers belong

to a higher dimensional space than the input and output, i.e., $q \geqslant \max(d, d')$. The residual transformations are two-layer perceptrons parameterized by the weight matrices $V_k^L \in \mathbb{R}^{q\times m}$ and $W_k^L \in \mathbb{R}^{m\times q}$. This is standard in the literature (e.g., He et al., 2016b; Chen et al., 2018a; Dupont et al., 2019; Barboni et al., 2022). The last weight matrix $B^L \in \mathbb{R}^{d'\times q}$ maps the last hidden layer to the output $F^L(x)$ in $\mathbb{R}^{d'}$. Also, $\sigma : \mathbb{R} \to \mathbb{R}$ is an element-wise activation function assumed to be $\mathcal{C}^2$, non-constant, Lipschitz continuous, bounded, and such that $\sigma(0) = 0$. The convenient shorthand $Z_k^L = (V_k^L, W_k^L)$ is occasionally used, and we denote $\|Z_k^L\|_F$ the sum of the Frobenius norms $\|V_k^L\|_F + \|W_k^L\|_F$.

Data and loss. The data is a sample of n pairs $(x_i, y_i)_{1\leq i\leq n} \in (\mathcal{X} \times \mathcal{Y})^n$ where $\mathcal{X} \times \mathcal{Y}$ is a compact set of $\mathbb{R}^d \times \mathbb{R}^{d'}$. The empirical risk is the mean squared error $\ell^L = \frac{1}{n}\sum_{i=1}^n \|F^L(x_i) - y_i\|^2$.

Initialization. We initialize $A^L = (I_{\mathbb{R}^{d\times d}}, 0_{\mathbb{R}^{(q-d)\times d}})$ as the identity matrix in $\mathbb{R}^{d\times d}$ concatenated row-wise with the zero matrix in $\mathbb{R}^{(q-d)\times d}$, to act as a simple projection of the input onto the higher dimensional space $\mathbb{R}^q$, and similarly $B^L = (0_{\mathbb{R}^{d'\times(q-d')}}, I_{\mathbb{R}^{d'\times d'}})$. The weights V_k^L are initialized to zero and the W_k^L as weight-tied standard Gaussian matrices, i.e., for all $k \in \{1, \dots, L\}$, $W_k^L = W \sim \mathcal{N}(0, 1)^{\otimes(m\times q)}$. Initializing outer matrices to zero is standard practice (Zhang et al., 2019a), while taking weight-tied matrices instead of i.i.d. ones is less common. We show in Section 3.5 that it is still possible to learn with this initialization scheme on real world data. As explained in Section 3.4.3, other initialization choices are possible, provided they correspond to the discretization of a Lipschitz continuous function, but we focus on this one in the main text for simplicity.

Training algorithm. Gradient flow is the limit of gradient descent as the learning rate tends to zero. The parameters are set at time $t = 0$ by the initialization, and then evolve according to the ODE

$$\frac{dA^L}{dt}(t) = -\frac{\partial \ell^L}{\partial A^L}(t), \quad \frac{dZ_k^L}{dt}(t) = -L\frac{\partial \ell^L}{\partial Z_k^L}(t), \quad \frac{dB^L}{dt}(t) = -\frac{\partial \ell^L}{\partial B^L}(t), \quad t \geqslant 0, \tag{3.4}$$

for $k \in \{1, \dots, L\}$. In the following, the dependence in t is made explicit when necessary, e.g., we write $h_k^L(t)$ instead of h_k^L, and $F^L(x; t)$ instead of $F^L(x)$.

It turns out that, without further assumptions, the gradient flow can diverge in finite time. This is because the dynamics are not (globally) Lipschitz continuous, breaking the conditions of the Picard-Lindelöf theorem (see Lemma 3.19) for existence and uniqueness of ODE solutions. A common practice (Goodfellow et al., 2016, Section 10.11.1) is to consider instead a clipped gradient flow

$$\frac{dA^L}{dt}(t) = \pi\Big(-\frac{\partial \ell^L}{\partial A^L}(t)\Big), \quad \frac{dZ_k^L}{dt}(t) = \pi\Big(-L\frac{\partial \ell^L}{\partial Z_k^L}(t)\Big), \quad \frac{dB^L}{dt}(t) = \pi\Big(-\frac{\partial \ell^L}{\partial B^L}(t)\Big), \tag{3.5}$$

where π is a generic notation for a bounded Lipschitz continuous operator. For example, clipping each coordinate of the gradient at some $C > 0$ amounts to taking π as the projection on the ball centered at 0 of radius C for the ℓ_∞ norm. Clipping ensures that the dynamics are well defined, as shown in the next proposition that is a consequence of the Picard-Lindelöf theorem.

Proposition 3.1. *The (clipped) gradient flow* (3.5) *has a unique solution for all* $t \geqslant 0$.

Proof. See Section 3.B.1. □

In Section 3.4.2, we make additional assumptions to prove the long-time convergence of the gradient flow. We then prove that these assumptions ensure that the dynamics of the gradient flow (3.4) are well defined, eliminating the need for clipping.

Neural ODE. The neural ODE corresponding to the residual network (3.3) is defined by

$$\begin{aligned} H(0) &= Ax \\ \frac{dH}{ds}(s) &= \frac{1}{\sqrt{m}}\mathcal{V}(s)\sigma\Big(\frac{1}{\sqrt{q}}\mathcal{W}(s)H(s)\Big), \quad s \in [0,1], \\ F(x) &= BH(1), \end{aligned} \tag{3.6}$$

where $x \in \mathbb{R}^d$ is the input, $H \in \mathbb{R}^q$ is the variable of the ODE, $\mathcal{V} : [0,1] \to \mathbb{R}^{q\times m}$ and $\mathcal{W} : [0,1] \to \mathbb{R}^{m\times q}$ are Lipschitz continuous functions, $A \in \mathbb{R}^{q\times d}$ and $B \in \mathbb{R}^{d'\times q}$ are matrices, and the output is $F(x) \in \mathbb{R}^{d'}$. The following proposition shows that the neural ODE is well defined. In addition, its output is close to the residual network (3.3) provided the weights are discretizations of $\mathcal{V}$ and $\mathcal{W}$.

Proposition 3.2. *The neural ODE* (3.6) *has a unique solution $H : [0,1] \to \mathbb{R}^q$. Consider, moreover, the residual network* (3.3) *with $A^L = A$, $V_k^L = \mathcal{V}(k/L)$ and $W_k^L = \mathcal{W}(k/L)$ for $k \in \{1,\dots,L\}$, and $B^L = B$. Then there exists $C > 0$ such that, for all $L \in \mathbb{N}^*$, $\sup_{x\in\mathcal{X}} \|F(x) - F^L(x)\| \leq \frac{C}{L}$.*

Proof. See Section 3.B.2. □

Clearly, our choices of V_k^L and W_k^L at initialization are discretizations of the Lipschitz continuous (in fact, constant) functions $\mathcal{V}(s) \equiv 0$ and $\mathcal{W}(s) \equiv W \sim \mathcal{N}(0,1)^{\otimes(m\times q)}$. Thus, Proposition 3.2 holds at initialization, and the residual network is equivalent to the trivial ODE $\frac{dH}{ds}(s) = 0$. The next section shows that after training we obtain non-trivial dynamics, which still discretize neural ODEs.

3.4 Large-depth limit of residual networks

We study the large-depth limit of trained residual networks in two settings. In Section 3.4.1, we consider the case of a finite training time. We move in Section 3.4.2 to the case where the training time tends to infinity, which is tractable under a Polyak-Łojasiewicz condition.

3.4.1 Clipped gradient flow and finite training time

We first consider the case where the neural network is trained with clipped gradient flow (3.5) on some training time interval $[0,T]$, $T > 0$. This allows us to prove large-depth convergence to a neural ODE without further assumptions. We emphasize that stopping training after a finite training time is a common technique in practice, referred to as early stopping (Goodfellow et al., 2016, Section 7.8). It is considered as a form of implicit regularization, and our result sheds light on this intuition by showing that the complexity of the trained networks increases with T.

The following proposition is a key step in proving the main theorem of this section.

Proposition 3.3. *There exist $M, K > 0$ such that, for any $t \in [0,T]$, $L \in \mathbb{N}^*$, and $k \in \{1,\dots,L\}$,*

$$\max\big(\big\|A^L(t)\big\|_F, \big\|V_k^L(t)\big\|_F, \big\|W_k^L(t)\big\|_F, \big\|B^L(t)\big\|_F\big) \leq M,$$

and, for $k \in \{1,\dots,L-1\}$,

$$\max\big(\big\|V_{k+1}^L(t) - V_k^L(t)\big\|_F, \big\|W_{k+1}^L(t) - W_k^L(t)\big\|_F\big) \leq \frac{K}{L}.$$

Moreover, with probability at least $1 - \exp\big(-\frac{3qm}{16}\big)$, the following expressions hold for M and K:

$$M = TM_\pi + 2\sqrt{qm}, \quad K = \beta T e^{\alpha T}, \tag{3.7}$$

where M_π is the supremum of π in Frobenius norm, and α and β depend on $\mathcal{X}$, $\mathcal{Y}$, M, and σ.

Proof. See Section 3.B.3. □

This proposition ensures that the size of the weights and the difference between successive weights remain bounded throughout training. We can now state the main result, which states the convergence, for any training time in $[0, T]$, of the neural network to a neural ODE as $L \to \infty$. Recall that a sequence of functions f^L of some variable u is said to converge uniformly over $u \in U$ to f if $\sup_{u \in U} \|f^L(u) - f(u)\| \to 0$.

Theorem 3.4. *Consider the residual network* (3.3) *with the training dynamics* (3.5)*. Then the following statements hold* ***as*** L ***tends to infinity****:*

(i) There exist functions $A : [0, T] \to \mathbb{R}^{q \times d}$ and $B : [0, T] \to \mathbb{R}^{d' \times q}$ such that $A^L(t)$ and $B^L(t)$ converge uniformly over $t \in [0, T]$ to $A(t)$ and $B(t)$.

(ii) There exists a Lipschitz continuous function $\mathcal{Z} : [0, 1] \times [0, T] \to \mathbb{R}^{q \times m} \times \mathbb{R}^{m \times q}$ such that

$$\mathcal{Z}^L : [0, 1] \times [0, T] \to \mathbb{R}^{q \times m} \times \mathbb{R}^{m \times q}, \ (s, t) \mapsto \mathcal{Z}^L(s, t) = Z^L_{\lfloor (L-1)s \rfloor + 1}(t) \tag{3.8}$$

converges uniformly over $s \in [0, 1]$ and $t \in [0, T]$ to $\mathcal{Z} = (\mathcal{V}, \mathcal{W})$.

(iii) Uniformly over $s \in [0, 1]$, $t \in [0, T]$, and $x \in \mathcal{X}$, the hidden layer $h^L_{\lfloor Ls \rfloor}(t)$ converges to the solution at time s of the neural ODE

$$\begin{aligned} H(0, t) &= A(t)x \\ \frac{\partial H}{\partial s}(s, t) &= \frac{1}{\sqrt{m}} \mathcal{V}(s, t) \sigma\Big(\frac{1}{\sqrt{q}} \mathcal{W}(s, t) H(s, t)\Big), \quad s \in [0, 1]. \end{aligned} \tag{3.9}$$

(iv) Uniformly over $t \in [0, T]$ and $x \in \mathcal{X}$, the output $F^L(x; t)$ converges to $B(t)H(1, t)$.

Proof. See Section 3.B.4. □

Let us sketch the proof of statement (ii), which is the cornerstone of the theorem. A first key idea is to introduce in (3.8) the piecewise-constant continuous-depth interpolation $\mathcal{Z}^L$ of the weights, whose ambient space does not depend on L, in contrast to the discrete weight sequence Z^L_k. Since the weights remain bounded during training by Proposition 3.3, the Arzelà-Ascoli theorem guarantees the existence of an accumulation point for $\mathcal{Z}^L$. We show that the accumulation point is unique because it is the solution of an ODE satisfying the conditions of the Picard-Lindelöf theorem. The uniqueness of the accumulation point then implies the existence of a limit for the weights.

There are two notable byproducts of our proof. The first one is an explicit description of the training dynamics of the limiting weights A, B, and $\mathcal{Z}$, as the solution of an ODE system, as presented in Appendix 3.A.5. The second one, which we now describe, consists of norm bounds on the weights. Proposition 3.3 bounds the discrete weights and the difference between two consecutive weights respectively by some $M, K > 0$. The proof of Theorem 3.4 shows that this bound carries over to the continuous weights, in the sense that $A(t)$, $\mathcal{V}(s, t)$, $\mathcal{W}(s, t)$, and $B(t)$ are uniformly bounded by M, and $\mathcal{V}(\cdot, t)$ and $\mathcal{W}(\cdot, t)$ are uniformly Lipschitz continuous with Lipschitz constant K. Formally, this last property means that, for any $s, s' \in [0, 1]$ and $t \in [0, T]$,

$$\|\mathcal{V}(s', t) - \mathcal{V}(s, t)\|_F \leq K|s' - s| \quad \text{and} \quad \|\mathcal{W}(s', t) - \mathcal{W}(s, t)\|_F \leq K|s' - s|.$$

The boundedness and Lipschitz continuity of the weights are important features because they limit the statistical complexity of neural ODEs (see Chapter 4). More generally, norm-based bounds are a common approach in the statistical theory of deep learning (see, e.g., Bartlett

et al., 2017, and references therein). Looking at the formula (3.7) for M and K, one can see in particular that the bounds diverge exponentially with T, providing an argument in favor of early stopping.

Our approach so far characterizes the large-depth limit of the neural network for a finite training time T, but two questions remain open. A first challenge is to characterize the value of the loss after training. A second one is to provide insight into the convergence of the optimization algorithm in the long-time limit, i.e., as T tends to infinity. To answer these questions, we move to the setting where the width of the network is large enough, which allows us to prove a Polyak-Łojasiewicz condition and thereby the long-time convergence of the training loss to zero.

3.4.2 Convergence in the long-time limit for wide networks

Proving convergence of gradient-based optimization algorithms for neural networks is a major difficulty in deep learning theory. One direction recently explored considers sufficiently wide neural networks, with the Polyak-Łojasiewicz (PL) condition. In our setting, they are written as follows (with the notation $Z^L = (V_k^L, W_k^L)_{k\in\{1,\dots,L\}}$):

Definition 3.5. *For $M, \mu > 0$, the residual network (3.3) is said to satisfy the (M,μ)-local PL condition around a set of parameters $(\bar{A}^L, \bar{Z}^L, \bar{B}^L)$ if, for every set of parameters (A^L, Z^L, B^L) such that*

$$\|A^L - \bar{A}^L\|_F \leqslant M, \qquad \sup_{k\in\{1,\dots,L\}} \|Z_k^L - \bar{Z}_k^L\|_F \leqslant M, \quad \|B^L - \bar{B}^L\|_F \leqslant M,$$

one has

$$\Big\|\frac{\partial \ell^L}{\partial A^L}\Big\|_F^2 + L\sum_{k=1}^{L}\Big\|\frac{\partial \ell^L}{\partial Z_k^L}\Big\|_F^2 + \Big\|\frac{\partial \ell^L}{\partial B^L}\Big\|_F^2 \geq \mu \ell^L,$$

where the loss ℓ^L is evaluated at the set of parameters (A^L, Z^L, B^L).

The next important point is to observe that, under the setup of Section 3.3 and some additional assumptions, the residual network satisfies the local PL condition of Definition 3.5.

Proposition 3.6. *Assume that the sample points (x_i, y_i) are i.i.d. such that $\|x_i\|_2 = \sqrt{q}$. Then there exist $c_1, \dots, c_4 > 0$ (depending only on σ) and $\delta > 0$ such that, if*

$$q \geq d + d', \quad m \geq c_1 n, \quad L \geq c_2\sqrt{nq},$$

then, with probability at least $1-\delta$, the residual network (3.3) satisfies the (M,μ)-local PL condition around its initialization, with $M = \dfrac{c_3}{\sqrt{nq}}$ and $\mu = \dfrac{c_4}{n\sqrt{nq}}$.

Proof. See Section 3.B.5. □

We emphasize that Proposition 3.6 requires the width m to scale only linearly with the sample size n, which improves on the literature (see Section 3.2). The other assumptions are mild. Note that our proof shows that the parameter δ is small if n grows at most polynomially with d (see Appendix 3.B.5).

We are now ready to state convergence in the long-time and large-depth limits to a global minimum of the empirical risk, when the local PL condition holds and the norm of the targets y_i is small enough.

Theorem 3.7. *Consider the residual network* (3.3) *with the training dynamics* (3.4)*, and assume that the assumptions of Proposition 3.6 hold. Then there exist* $C, \delta > 0$ *such that, if* $\frac{1}{n}\sum_{i=1}^n \|y_i\|^2 \leq C$*, then, with probability as least* $1-\delta$*, the gradient flow is well defined on* $\mathbb{R}_+$*, and, for* $t \in \mathbb{R}_+$ *and* $L \in \mathbb{N}^*$*,*

$$\ell^L(t) \leq \exp\big(-\frac{C't}{n\sqrt{nq}}\big)\ell^L(0), \tag{3.10}$$

for some $C' > 0$ *depending on* σ*. Moreover, the following statements hold* ***as*** t ***and*** L ***tend to infinity****:*

(i) There exist matrices $A_\infty \in \mathbb{R}^{q\times d}$ *and* $B_\infty \in \mathbb{R}^{d'\times q}$ *such that* $A^L(t)$ *and* $B^L(t)$ *converge to* A_∞ *and* B_∞*.*

(ii) There exists a Lipschitz continuous function $\mathcal{Z}_\infty : [0,1] \to \mathbb{R}^{q\times m} \times \mathbb{R}^{m\times q}$ *such that*

$$\mathcal{Z}^L : [0,1]\times\mathbb{R}_+ \to \mathbb{R}^{q\times m}\times\mathbb{R}^{m\times q},\ (s,t) \mapsto \mathcal{Z}^L(s,t) = Z^L_{\lfloor (L-1)s\rfloor+1}(t)$$

converges uniformly over $s \in [0,1]$ *to* $\mathcal{Z}_\infty = (\mathcal{V}_\infty, \mathcal{W}_\infty)$*.*

(iii) Uniformly over $s \in [0,1]$ *and* $x \in \mathcal{X}$*, the hidden layer* $h^L_{\lfloor Ls\rfloor}(t)$ *converges to the solution at time* s *of the neural ODE*

$$\begin{aligned} H(0) &= A_\infty x \\ \frac{dH}{ds}(s) &= \frac{1}{\sqrt{m}}\mathcal{V}_\infty(s)\sigma\Big(\frac{1}{\sqrt{q}}\mathcal{W}_\infty(s)H(s)\Big), \quad s\in[0,1]. \end{aligned}$$

(iv) Uniformly over $x \in \mathcal{X}$*, the output* $F^L(x;t)$ *converges to* $F_\infty(x) = B_\infty H(1)$*. Furthermore,* $F_\infty(x_i) = y_i$ *for all* $i \in \{1,\dots,n\}$*.*

Proof. See Section 3.B.6. □

This theorem proves two important results of separate interest. On the one hand, equation (3.10) shows the long-time convergence of the gradient flow for deep residual networks under the linear overparameterization assumption $m \geq c_1 n$ of Proposition 3.6. On the other hand, when both t and L tend to infinity, the network converges to a neural ODE that further interpolates the training data. Note that the order in which t and L tend to infinity does not matter by uniform convergence properties.

3.4.3 Generalizations to other architectures and initialization

To simplify the exposition, we have so far considered a particular residual architecture defined in (3.3). However, most of our results hold for a more general residual network of the form

$$h^L_{k+1} = h^L_k + \frac{1}{L}f(h^L_k, Z^L_{k+1}), \quad k \in \{0,\dots,L-1\}, \tag{3.11}$$

where $f : \mathbb{R}^q \times \mathbb{R}^p \to \mathbb{R}^q$ is a $\mathcal{C}^2$ function such that $f(0,\cdot) \equiv 0$ and $f(\cdot, z)$ is uniformly Lipschitz for z in any compact. All our results are shown in the appendix for this general model, except the PL condition of Proposition 3.6, which we prove only for the specific setup of Section 3.3. In particular, the conclusions of Theorem 3.4 hold for the general model (3.11), as well as those of Theorem 3.7 if the network satisfies a (M,μ)-local PL condition with μ sufficiently large (see Appendix 3.B for details).

It is easy to see that our residual network of interest (3.3) is a special case of the general model (3.11) if σ satisfies the assumptions of Section 3.3. However, other choices are possible, such as convolutional layers or a Lipschitz continuous version of Transformer (Kim et al., 2021). This latter application is particularly interesting in the light of the literature analyzing the Transformer architecture from a neural ODE point of view (Lu et al., 2019; Sander et al., 2022a; Geshkovski et al., 2023).

Moreover, the initialization assumption made in Section 3.3 can also be relaxed to include any so-called *smooth* initialization of the weights (see Chapter 2). A smooth initialization corresponds to taking $V_k^L(0)$ and $W_k^L(0)$ as discretizations of some Lipschitz continuous functions $\mathcal{V}_0 : [0,1] \to \mathbb{R}^{q\times m}$ and $\mathcal{W}_0 : [0,1] \to \mathbb{R}^{m\times q}$, that is, for $k \in \{1,\dots,L\}$, $V_k^L(0) = \mathcal{V}_0(\frac{k}{L})$ and $W_k^L(0) = \mathcal{W}_0(\frac{k}{L})$. A typical concrete example is to let the entries of $\mathcal{V}_0$ and $\mathcal{W}_0$ be independent Gaussian processes with expectation zero and squared exponential covariance $K(x,x') = \exp(-\frac{(x-x')^2}{2\ell^2})$, for some $\ell > 0$. As shown by Proposition 3.2, a smooth initialization means that the network discretizes a neural ODE.

3.5 Numerical experiments

We now present numerical experiments to validate our theoretical findings, using both synthetic and real-world data. Experimental details are given in Appendix 3.E. Our code will be open sourced.

3.5.1 Synthetic data

We consider the residual network (3.3) with the initialization scheme of Section 3.3. So, the V_k^L are initialized to zero and the W_k^L to weight-tied standard Gaussian matrices. To ease the presentation, we consider the case where $q = d = d'$, and we do not train the weights A^L and B^L, which therefore stay equal to the identity. The activation function is GELU (Hendrycks and Gimpel, 2016), which is a smooth approximation of ReLU: $x \mapsto \max(x,0)$. The sample points $(x_i,y_i)_{1\le i\le n}$ follow independent standard Gaussian distributions. Note that it does not hurt to take x and y independent since, in this subsection, our focus is on optimization results only and not on statistical aspects. The mean-squared error is minimized using full-batch gradient descent. The following experiments exemplify the large-depth ($t \in [0,T]$, $L \to \infty$) and long-time ($t \to \infty$, L finite) limits.

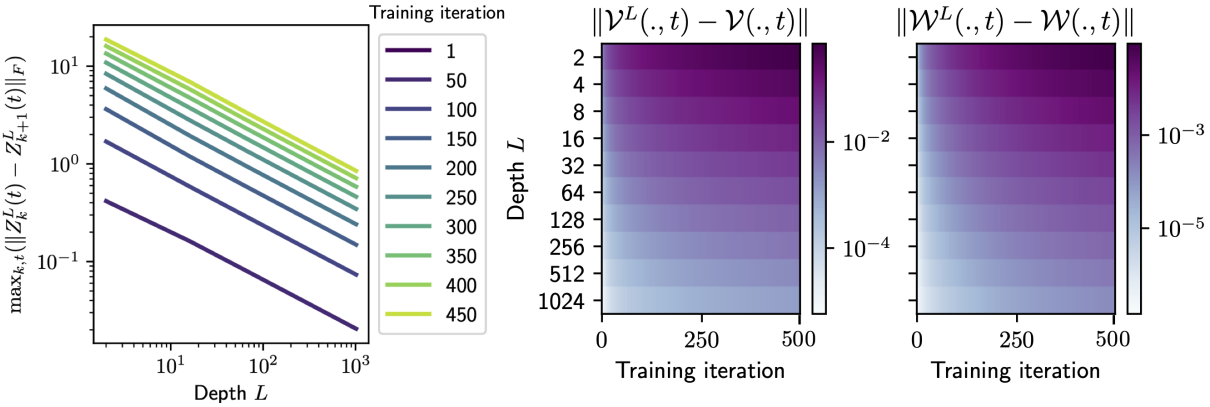


Figure 3.1: **Left**: $1/L$ convergence of the maximum distance between two successive weight matrices $\max_{1\le k\le L, t\in[0,T]}(\|Z_k^L(t) - Z_{k+1}^L(t)\|_F)$. **Right**: uniform convergence of $\mathcal{Z}^L$ to its large-depth limit $\mathcal{Z}$. Here, for a matrix-valued function f, $\|f\|$ denotes $(\int_0^1 \|f(s)\|_F^2 ds)^{1/2}$.

Large-depth limit. We illustrate key insights of Proposition 3.3 and Theorem 3.4, with $T = 500$. In Figure 3.1 (left), we plot the maximum distance between two successive weight matrices, i.e., $\max_{1\leq k\leq L, t\in[0,T]}(\|Z_k^L(t) - Z_{k+1}^L(t)\|_F)$, for different values of L and training time $t \in [0, T]$. We observe a $1/L$ convergence rate, as predicted by Proposition 3.3. Moreover, for a fixed L, the distance between two successive weight matrices increases with the training time, however at a much slower pace than the exponential upper bound on K given in identity (3.7). Figure 3.1 (right) depicts the uniform convergence of $\mathcal{Z}^L$ to its large-depth limit $\mathcal{Z}$, illustrating statement (ii) of Theorem 3.4. The function $\mathcal{Z}$ is computed using $\mathcal{Z}^L$ for $L = 2^{14}$. Note that the convergence is slower for larger training times.

Long-time limit. We now turn to the long-time training setup, training for 80,000 iterations with $L = 64$. In Figure 3.2, we plot a specific (randomly-chosen) entry of matrices V_k^L and W_k^L across layers, for different training times. This illustrates Theorem 3.7 in a practical setting since, visually, the weights behave as a Lipschitz continuous function for any training time and converge to a Lipschitz continuous function as $t \to \infty$. We also display the loss as a function of the training time, corroborating the convergence of the loss to zero in Theorem 3.7.

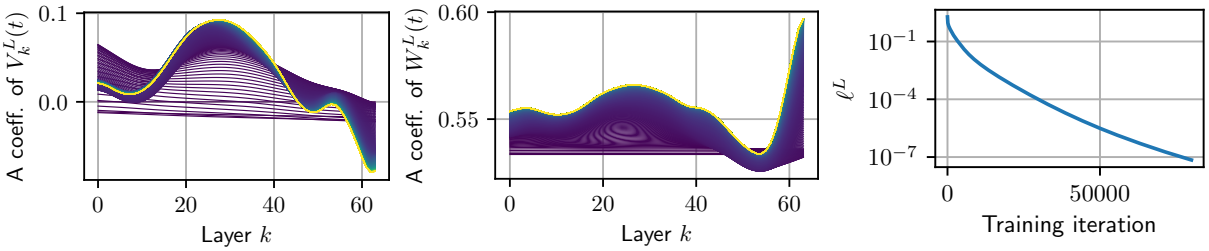


Figure 3.2: **Left**: Randomly-chosen entry of the weight matrices across layers (x-axis) for various training times t (lighter color indicates higher training time). **Right**: Loss against training time.

3.5.2 Real-world data

We now investigate the properties of deep residual networks on the CIFAR 10 dataset (Krizhevsky, 2009). We deviate from the mathematical model (3.3) by using convolutions instead of fully connected layers. More precisely, A^L is replaced by a trainable convolutional layer, and the residual layers write

$$h_{k+1}^L = h_k^L + \frac{1}{L}\mathrm{bn}_{2,k}^L(\mathrm{conv}_{2,k}^L(\sigma(\mathrm{bn}_{1,k}^L(\mathrm{conv}_{1,k}^L(h_k^L))))), \quad k \in \{0, \dots, L-1\},$$

where $\mathrm{conv}_{i,k}^L$ are convolutions and $\mathrm{bn}_{i,k}^L$ are batch normalizations. The output of the residual layers is mapped to logits through a linear layer B^L. We initialize $\mathrm{bn}_{2,\mathrm{k}}^{\mathrm{L}}$ to 0, and $\mathrm{bn}_{1,\mathrm{k}}^{\mathrm{L}}$ and $\mathrm{conv}_{\mathrm{i,k}}^{\mathrm{L}}$ either to weight-tied or to i.i.d. Gaussian. Table 3.1 reports the accuracy of the trained network, and whether it has Lipschitz continuous (or smooth) weights after training, depending on the activation function σ and on the initialization scheme. To assess the smoothness of the weights, we simply resort to visual inspection. For example, Figure 3.3 (left) shows two random entries of the convolutions across layers with GELU and a weight-tied initialization: the smoothness is preserved after training. Smooth weights indicate that the residual network discretizes a neural ODE (see, e.g., Proposition 3.2). On the contrary, if an i.i.d. initialization is used, smoothness is not preserved after training, as shown in Figure 3.3 (right), and the residual network does not discretize a neural ODE.

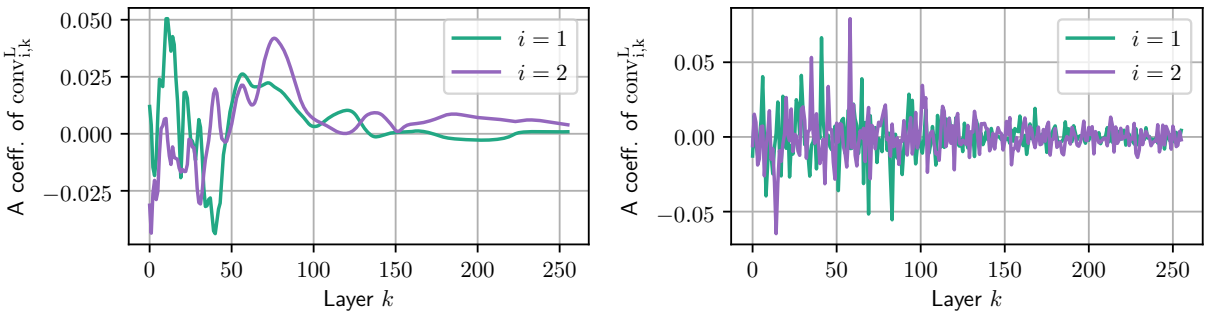


Figure 3.3: Random entries of the convolutions across layers (x-axis) after training. **Left:** Weight-tied initialization leads to smooth weights. **Right:** i.i.d. initialization leads to non-smooth weights.

Act. function	Init. scheme	Train Acc.	Test Acc.	Smooth trained weights
Identity	Weight-tied	56.5 ± 0.1	59.8 ± 0.7	✓
	i.i.d.	56.1 ± 0.3	59.6 ± 0.7	×
GELU	Weight-tied	80.5 ± 0.7	79.9 ± 0.2	✓
	i.i.d.	89.8 ± 0.5	85.7 ± 0.1	×
ReLU	Weight-tied	97.4 ± 0.6	88.1 ± 0.1	×
	i.i.d.	98.4 ± 0.1	88.4 ± 0.5	×

Table 3.1: Accuracy and smoothness of the trained weights depending on the choice of activation function σ and initialization scheme. We display the median over 5 runs and the interquartile range between the first and third quantile. Smooth weights correspond to a neural ODE structure.

Table 3.1 conveys several important messages. First, in accordance with our theory (Theorem 3.4), we obtain a neural ODE structure when using a smooth activation function and weight-tied initialization (lines 1 and 3 of Table 3.1). This is not the case when using the non-smooth ReLU activation and/or i.i.d. initialization. In fact, we prove in Appendix 3.D that the smoothness of the weights is lost when training with ReLU in a simple setting, confirming this experimental observation. Furthermore, the third line of Table 3.1 shows that it is possible to obtain a reasonable accuracy with a neural ODE structure, which, as emphasized in Section 3.1, also comes with theoretical and practical advantages. Nevertheless, we obtain an improvement in accuracy in the cases corresponding to non-smooth weights, i.e., to a residual network that does *not* discretize an ODE. Extending our theory to such cases is left for future work.

3.6 Conclusion

We study the convergence of deep residual networks to neural ODEs. When properly scaled and initialized, residual networks trained with fixed-horizon gradient flow converge to neural ODEs as the depth tends to infinity. This result holds for very general architectures. In the case where both training time and depth tend to infinity, convergence holds under a local Polyak-Łojasiewicz condition. We prove such a condition for a family of deep residual networks with linear overparameterization.

The setting of neural ODE-like networks comes with strong guarantees, at the cost of some performance gap when compared with i.i.d. initialization as highlighted by the experimental section. Extending the mathematical large-depth study to the i.i.d. case is an interesting problem for future research. Previous work suggests that the correct limit object is then a stochastic

differential equation instead of an ODE, such as Chapter 2 of this manuscript or Cohen et al. (2021); Cont et al. (2022).

3.A Some results for general residual networks

Lipschitz continuity. Let $(\mathscr{U}, \|\cdot\|)$, $(\mathscr{V}, \|\cdot\|)$, and $(\mathscr{W}, \|\cdot\|)$ be generic normed spaces. Then a function of two variables $g : \mathscr{U} \times \mathscr{V} \to \mathscr{W}$ is:

(*i*) (Globally) Lipschitz continuous if there exists $K \geq 0$ such that, for $(u, v), (u', v') \in \mathscr{U} \times \mathscr{V}$,

$$\|g(u, v) - g(u', v)\| \leq K\|u - u'\| + K\|v - v'\|.$$

(*ii*) Locally Lipschitz continuous in its first variable if, for any compacts $E \subset \mathscr{U}, E' \subset \mathscr{V}$, there exists $K \geq 0$ such that, for $(u, v), (u', v) \in E \times E'$,

$$\|g(u, v) - g(u', v)\| \leq K\|u - u'\|.$$

Equivalent definitions hold for a function of one variable. Moreover, $g(\cdot, v)$ is said to be uniformly Lipschitz continuous for v in $\mathscr{V}$ if there exists $K \geq 0$ such that, for $(u, v), (u', v) \in \mathscr{U} \times \mathscr{V}$,

$$\|g(u, v) - g(u', v)\| \leq K\|u - u'\|,$$

and uniformly Lipschitz continuous for v in any compact if, for any compact $E' \subset \mathscr{V}$, there exists $K \geq 0$ such that, for $(u, v), (u', v) \in \mathcal{U} \times E'$,

$$\|g(u, v) - g(u', v)\| \leq K\|u - u'\|.$$

Throughout, we refer to a Lipschitz continuous function with Lipschitz constant $K \geq 0$ as K-Lipschitz.

Model. As explained in Section 3.4.3, most of our results are proven for the general residual network

$$\begin{aligned} h_0^L(t) &= A^L(t)x \\ h_{k+1}^L(t) &= h_k^L(t) + \frac{1}{L} f(h_k^L(t), Z_{k+1}^L(t)), \quad k \in \{0, \dots, L-1\}, \\ F^L(x; t) &= B^L(t) h_L^L(t), \end{aligned} \tag{3.12}$$

where $Z^L(t) = (Z_1^L(t), \dots, Z_L^L(t)) \in (\mathbb{R}^p)^L$ and $f : \mathbb{R}^q \times \mathbb{R}^p \to \mathbb{R}^q$ is a $\mathcal{C}^2$ function such that $f(0, \cdot) \equiv 0$ and $f(\cdot, z)$ is uniformly Lipschitz for z in any compact. Let us introduce the backpropagation equations, which are instrumental in the study of the gradient flow dynamics. These equations define the backward state $p_k^L(t) \in \mathbb{R}^q$ through the backward recurrence

$$\begin{aligned} p_L^L(t) &= 2B^L(t)^\top (F^L(x; t) - y) \\ p_k^L(t) &= p_{k+1}^L(t) + \frac{1}{L} \partial_1 f(h_k^L(t), Z_{k+1}^L(t)) p_{k+1}^L(t), \quad k \in \{0, \dots, L-1\}, \end{aligned} \tag{3.13}$$

where $\partial_1 f \in \mathbb{R}^{q \times q}$ stands for the Jacobian matrix of f with respect to its first argument. Similarly, we let $\partial_2 f \in \mathbb{R}^{q \times p}$ be the Jacobian matrix of f with respect to its second argument. For a sample $(x_i, y_i)_{1 \leq i \leq n} \in (\mathcal{X} \times \mathcal{Y})^n$, we let $h_{k,i}^L(t)$ and $p_{k,i}^L(t)$ be, respectively, the hidden layer $h_k^L(t)$ and

the backward state $p_k^L(t)$ associated with the i-th input x_i. Denoting the mean squared error associated with the sample by ℓ^L, we have, by the chain rule,

$$\frac{\partial \ell^L}{\partial A^L}(t) = \frac{1}{n}\sum_{i=1}^{n} p_{0,i}^L(t) x_i^\top \tag{3.14}$$

$$\frac{\partial \ell^L}{\partial Z_k^L}(t) = \frac{1}{nL}\sum_{i=1}^{n} \partial_2 f(h_{k-1,i}^L(t), Z_k^L(t))^\top p_{k-1,i}^L(t), \quad k \in \{1, \dots, L\}, \tag{3.15}$$

$$\frac{\partial \ell^L}{\partial B^L}(t) = \frac{2}{n}\sum_{i=1}^{n} (F^L(x_i; t) - y_i) h_{L,i}^L(t)^\top. \tag{3.16}$$

Initialization. The parameters $(Z_k^L(t))_{1\le k\le L}$ are initialized to $Z_k^L(0) = Z^{\text{init}}\big(\frac{k}{L}\big)$, where $Z^{\text{init}} : [0,1] \to \mathbb{R}^p$ is a Lipschitz continuous function. Furthermore, we initialize $A^L(0)$ to some matrix $A^{\text{init}} \in \mathbb{R}^{q\times d}$ and $B^L(0) = B^{\text{init}} \in \mathbb{R}^{d'\times q}$. Note that this initialization scheme is a generalization of the one presented in Section 3.3.

Additional notation. For a vector x, $\|x\|$ denotes the Euclidean norm. For a matrix A, the operator norm induced by the Euclidean norm is denoted by $\|A\|_2$, and the Frobenius norm is denoted by $\|A\|_F$. Finally, we use the notation A^L (resp. Z_k^L, B^L) to denote the function $t \mapsto A^L(t)$ (resp. $t \mapsto Z_k^L(t)$, $t \mapsto B^L(t)$), since the parameters are considered as functions of the training time throughout this appendix.

Overview of Appendix A. First, in Section 3.A.1, we study the case of the (clipped) gradient flow (3.5). We show that the weights and the difference between successive weights are bounded during the entire training. Section 3.A.2 shows a similar result for the standard gradient flow (3.4) under a PL condition. In Section 3.A.3, we show a generalized version of the Arzelà-Ascoli theorem, which allows us to prove the existence of a converging subsequence of the weights in the large-depth limit. Section 3.A.4 is devoted to the convergence of the Euler scheme for parameterized ODEs. We then proceed to prove in Section 3.A.5 our main result, i.e., the large-depth convergence of the gradient flow. The key step is to establish the uniqueness of the adherence point of the weights. Finally, in Section 3.A.6, we prove the existence of a double limit for the weights and the hidden states when both the depth and the training time tend to infinity.

3.A.1 The trained weights are bounded in the finite training-time setup

Before stating the result, let us introduce the notation $\partial_{22} f(h,z) \in \mathbb{R}^{q\times p\times p}$, which is the third-order tensor of second partial derivatives of f with respect to z. We endow the space $\mathbb{R}^{q\times p\times p}$ with the operator norm $\|\cdot\|_2$ induced by the Euclidean norm in $\mathbb{R}^p$ and the $\|\cdot\|_2$ norm in $\mathbb{R}^{q\times p}$. In other words,

$$\|\partial_{22} f(h,z)\|_2 = \sup_{u\in\mathbb{R}^p, \|u\|=1} \|\partial_{22} f(h,z) u\|_2,$$

where $\partial_{22} f(h,z)u \in \mathbb{R}^{q\times p}$ is the tensor product of $\partial_{22} f(h,z)$ against u. Similarly, $\partial_{21} f(h,z) \in \mathbb{R}^{q\times p\times q}$ denotes the third-order tensor of cross second partial derivatives of f, and the space $\mathbb{R}^{q\times p\times q}$ is endowed with the operator norm $\|\cdot\|_2$ induced by the Euclidean norm in $\mathbb{R}^q$ and the $\|\cdot\|_2$ norm in $\mathbb{R}^{q\times p}$.

Proposition 3.8. *Consider the residual network* (3.12) *initialized as explained in Appendix 3.A and trained with the gradient flow* (3.5) *on* $[0,T]$*, for some* $T \in (0,\infty)$*. Let*

$$M_\pi = \max\Big(\max_{A\in\mathbb{R}^{q\times d}} \|\pi(A)\|_F, \max_{Z\in\mathbb{R}^p} \|\pi(Z)\|, \max_{B\in\mathbb{R}^{d'\times q}} \|\pi(B)\|_F \Big),$$
$$M_0 = \max\Big(\|A^{\text{init}}\|_F, \sup_{s\in[0,1]} \|Z^{\text{init}}(s)\|, \|B^{\text{init}}\|_F \Big) \quad \textit{and} \quad M = M_0 + TM_\pi.$$

Then the gradient flow is well defined on $[0,T]$*, and, for* $t\in[0,T]$*,* $L\in\mathbb{N}^*$*, and* $k\in\{1,\dots,L\}$*,*

$$\|A^L(t)\|_F \le M, \quad \|Z_k^L(t)\| \le M, \quad \textit{and} \quad \|B^L(t)\|_F \le M. \tag{3.17}$$

Moreover, there exist $\alpha,\beta>0$ *such that, for* $t\in[0,T]$ *and* $k\in\{1,\dots,L-1\}$*,*

$$\|Z_{k+1}^L(t) - Z_k^L(t)\| \le \Big(\|Z_{k+1}^L(0) - Z_k^L(0)\| + \frac{\beta T}{L} \Big) e^{\alpha T}.$$

The following expressions for α *and* β *hold:*

$$\alpha = 2e^K K' M(e^K M^2 M_X + M_Y) \quad \textit{and} \quad \beta = 2Ke^K M(K + e^K K' M M_X)(e^K M^2 M_X + M_Y),$$

where

$$M_X = \sup_{x\in\mathcal{X}} \|x\|, \quad M_Y = \sup_{y\in\mathcal{Y}} \|y\|, \quad K_1 = \sup_{\|z\|\le M} \big\|\partial_1 f(h,z)\big\|_2 \tag{3.18}$$
$$E = \{(h,z)\in\mathbb{R}^d\times\mathbb{R}^p, \|h\| \le e^{K_1} M M_X,\ \|z\|\le M\} \tag{3.19}$$
$$K_2 = \sup_{(h,z)\in E} \big\|\partial_2 f(h,z)\big\|_2, \quad K = \max(K_1,K_2)$$
$$K' = \sup_{(h,z)\in E} \Big(\max\big(\big\|\partial_{22} f(h,z)\big\|_2, \big\|\partial_{21} f(h,z)\big\|_2 \big) \Big).$$

Proof. The time-independent dynamics

$$(A^L, Z_k^L, B^L) \mapsto \Big(\pi\Big(-\frac{\partial \ell^L}{\partial A^L}\Big), \pi\Big(-L\frac{\partial \ell^L}{\partial Z_k^L}\Big), \pi\Big(-\frac{\partial \ell^L}{\partial B^L}\Big) \Big)$$

defining the gradient flow (3.5) are locally Lipschitz continuous, hence the gradient flow is defined on a maximal interval $[0,T_{\max})$ by the Picard-Lindelöf theorem (see Lemma 3.19). Let us show by contradiction that $T_{\max} = T$. Assume that $T_{\max} < T$. If this is true, again by the Picard-Lindelöf theorem, we know that the parameters diverge to infinity at $T_{\max}$. However, for any $t\in[0,T_{\max})$, we have

$$\|A^L(t)\|_F \le \|A^L(0)\|_F + \int_0^t \Big\|\frac{dA^L}{dt}(\tau)\Big\|_F d\tau \le M_0 + \int_0^t M_\pi d\tau \le M_0 + TM_\pi = M.$$

Bounds on B^L and Z_k^L by M can be shown similarly. This contradicts the divergence of the parameters at $t = T_{\max}$. We conclude that the gradient flow is well defined on $[0,T]$ and that the bounds (3.17) hold.

It remains to bound the difference $\|Z_{k+1}^L(t) - Z_k^L(t)\|$. We have, for $t\in[0,T]$ and $k\in$

$\{1,\dots,L-1\}$,

$$\begin{aligned}\Big\|\frac{dZ_{k+1}^L}{dt}(t)-\frac{dZ_k^L}{dt}(t)\Big\| &= L\Big\|\frac{\partial \ell^L}{\partial Z_{k+1}^L}(t)-\frac{\partial \ell^L}{\partial Z_k^L}(t)\Big\| \\ &\le \sum_{i=1}^n \frac{1}{n}\big\|\partial_2 f(h_{k,i}^L(t),Z_{k+1}^L(t))^\top p_{k,i}^L(t)-\partial_2 f(h_{k-1,i}^L(t),Z_k^L(t))^\top p_{k-1,i}^L(t)\big\| \\ &\le \frac{1}{n}\sum_{i=1}^n \big\|\partial_2 f(h_{k,i}^L(t),Z_{k+1}^L(t))\big\|_2\,\big\|p_{k,i}^L(t)-p_{k-1,i}^L(t)\big\| \\ &\quad+\big\|p_{k-1,i}^L(t)\big\|\,\big\|\partial_2 f(h_{k,i}^L(t),Z_{k+1}^L(t))-\partial_2 f(h_{k-1,i}^L(t),Z_k^L(t))\big\|_2 \qquad (3.20)\end{aligned}$$

Furthermore, for $t\in[0,T]$, $k\in\{0,\dots,L-1\}$, and $i\in\{1,\dots,n\}$,

$$\|h_{k+1,i}^L(t)\| = \|h_{k,i}^L(t)+\frac{1}{L}f(h_{k,i}^L(t),Z_{k+1}^L(t))\| \le (1+\frac{K_1}{L})\|h_{k,i}^L(t)\|,$$

since $f(\cdot,Z_{k+1}^L(t))$ is K_1-Lipschitz, where K_1 is defined by (3.18), and $f(0,Z_{k+1}^L(t))=0$. Therefore, for any $k\in\{1,\dots,L\}$,

$$\|h_{k,i}^L(t)\| \le e^{K_1}\|h_{0,i}^L(t)\| = e^{K_1}\|A^L(t)x_i\| \le e^{K_1}MM_X. \qquad (3.21)$$

This bound shows that the pair $(h_{k,i}^L(t),Z_{k+1}^L(t))$ belongs to the compact E defined in (3.19) for every $t\in[0,T]$, $k\in\{1,\dots,L\}$, and $i\in\{1,\dots,n\}$. In particular, $\|\partial_2 f(h_{k-1,i}^L(t),Z_k^L(t))\|_2\le K$, and

$$\begin{aligned}&\big\|\partial_2 f(h_{k,i}^L(t),Z_{k+1}^L(t))-\partial_2 f(h_{k-1,i}^L(t),Z_k^L(t))\big\|_2 \\ &\qquad\le K'\|h_{k,i}^L(t)-h_{k-1,i}^L(t)\|+K'\|Z_{k+1}^L(t)-Z_k^L(t)\|.\end{aligned}$$

Returning to (3.20), we obtain

$$\begin{aligned}\Big\|\frac{dZ_{k+1}^L}{dt}(t)-\frac{dZ_k^L}{dt}(t)\Big\| &\le \frac{1}{n}\sum_{i=1}^n K\|p_{k,i}^L(t)-p_{k-1,i}^L(t)\| \\ &\quad+K'\|p_{k-1,i}^L(t)\|\big(\|h_{k,i}^L(t)-h_{k-1,i}^L(t)\|+\|Z_{k+1}^L(t)-Z_k^L(t)\|\big).\end{aligned}$$

For $k\in\{1,\dots,L\}$ and $i\in\{1,\dots,n\}$,

$$\big\|p_{k,i}^L(t)-p_{k-1,i}^L(t)\big\| = \frac{1}{L}\big\|\partial_1 f(h_{k-1,i}^L(t),Z_k^L(t))p_{k,i}^L(t)\big\| \le \frac{K}{L}\big\|p_{k,i}^L(t)\big\|,$$

and, similarly,

$$\big\|h_{k,i}^L(t)-h_{k-1,i}^L(t)\big\| = \frac{1}{L}\|f(h_{k-1,i}^L(t),Z_k^L(t))\| \le \frac{K}{L}\big\|h_{k-1,i}^L(t)\big\| \le \frac{Ke^KMM_X}{L}.$$

Thus,

$$\Big\|\frac{dZ_{k+1}^L}{dt}(t)-\frac{dZ_k^L}{dt}(t)\Big\| \le \frac{1}{n}\sum_{i=1}^n \|p_{k,i}^L(t)\|\Big(\frac{K^2}{L}+\frac{K'K}{L}e^KMM_X+K'\|Z_{k+1}^L(t)-Z_k^L(t)\|\Big).$$

Moreover, for $k\in\{0,\dots,L\}$ and $i\in\{1,\dots,n\}$,

$$\|p_{k,i}^L(t)\| \le \|p_{k+1,i}^L(t)\|+\frac{1}{L}\big\|\partial_1 f(h_{k,i}^L(t),Z_{k+1}^L(t))p_{k+1,i}^L(t)\big\| \le \|p_{k+1,i}^L(t)\|+\frac{K}{L}\|p_{k+1,i}^L(t)\|.$$

Hence

$$\begin{aligned}\|p_{k,i}^L(t)\| &\leq e^K \|p_{L,i}^L(t)\| = 2e^K \|B^L(t)^\top (F^L(x_i;t) - y_i)\| \\ &\leq 2e^K M\big(\|B^L(t) h_{L,i}^L(t)\| + \|y_i\|\big) \leq 2e^K M(e^K M^2 M_X + M_Y),\end{aligned}$$

where we use (3.17) and (3.21) for the last inequality. Putting all the pieces together, we obtain

$$\Big\|\frac{dZ_k^L}{dt}(t) - \frac{dZ_{k+1}^L}{dt}(t)\Big\| \leq \alpha \|Z_k^L(t) - Z_{k+1}^L(t)\| + \frac{\beta}{L}.$$

Integrating between 0 and t, we see that

$$\|Z_{k+1}^L(t) - Z_k^L(t)\| \leq \|Z_{k+1}^L(0) - Z_k^L(0)\| + \frac{\beta t}{L} + \int_0^t \alpha \|Z_k^L(\tau) - Z_{k+1}^L(\tau)\| d\tau.$$

Applying Grönwall's inequality (see, e.g., Dragomir, 2003), we conclude that $\|Z_{k+1}^L(t) - Z_k^L(t)\| \leq (\|Z_{k+1}^L(0) - Z_k^L(0)\| + \frac{\beta T}{L}) e^{\alpha T}$, as desired. □

3.A.2 The trained weights are bounded under the local PL condition

Proposition 3.9. *Consider the residual network* (3.12) *initialized as explained in Appendix 3.A and trained with the gradient flow* (3.4) *on* $[0, \infty]$. *Then, for* $M > 0$, *there exists* $\mu > 0$ *such that, if the residual network satisfies the* (M, μ)*-local PL condition* (3.5) *around its initialization for any* $L \in \mathbb{N}^*$, *then:*

(i) The gradient flow is well defined on $\mathbb{R}_+$, *and, for* $t \in \mathbb{R}_+$, $L \in \mathbb{N}^*$, *and* $k \in \{1, \dots, L\}$,

$$\|A^L(t)\|_F \leq M_A, \quad \|Z_k^L(t)\| \leq M_Z, \quad \text{and} \quad \|B^L(t)\|_F \leq M_B,$$

where

$$M_A = \|A^{\text{init}}\|_2 + M, \quad M_Z = \sup_{s \in [0,1]} \|Z^{\text{init}}(s)\| + M, \quad \text{and} \quad M_B = \|B^{\text{init}}\|_2 + M.$$

(ii) There exists $\tilde{K} > 0$ *such that, for* $t \in \mathbb{R}_+$, $L \in \mathbb{N}^*$, *and* $k \in \{1, \dots, L\}$,

$$\|Z_k^L(t) - Z_{k+1}^L(t)\| \leq \frac{\tilde{K}}{L}.$$

(iii) There exists a bounded integrable function $b : \mathbb{R}_+ \to \mathbb{R}$ *such that, for* $t \in \mathbb{R}_+$, $L \in \mathbb{N}^*$, *and* $k \in \{1, \dots, L\}$,

$$\max\Big(\Big\|\frac{dA^L}{dt}(t)\Big\|, \Big\|\frac{dZ_k^L}{dt}(t)\Big\|, \Big\|\frac{dB^L}{dt}(t)\Big\|\Big) \leq b(t)$$

(iv) $A^L(t)$, $B^L(t)$, *and* $Z_k^L(t)$ *admit a limit uniformly over* $L \in \mathbb{N}^*$ *and* $k \in \{1, \dots, L\}$ *as* $t \to \infty$.

(v) For $t \in \mathbb{R}_+$ *and* $L \in \mathbb{N}^*$, $\ell^L(t) \leq e^{-\mu t} \ell^L(0)$.

Moreover, the following expression for μ *hold:*

$$\mu = \max(M_B K, M_B M_X, M_A M_X) \frac{8e^K}{M} \sup_{L \in \mathbb{N}^*} \sqrt{\ell^L(0)}, \tag{3.22}$$

where

$$M_X = \sup_{x\in\mathcal{X}} \|x\|, \quad K_1 = \sup_{\|z\|\le M_Z} \big\|\partial_1 f(h,z)\big\|$$
$$E = \{(h,z)\in\mathbb{R}^d\times\mathbb{R}^p, \|h\| \le e^{K_1} M_A M_X,\ \|z\|\le M_Z\}$$
$$K_2 = \sup_{(h,z)\in E} \big\|\partial_2 f(h,z)\big\|, \quad K = \max(K_1, K_2).$$

Proof. Let $M > 0$, μ defined by (3.22), and assume that the residual network satisfies the (M,μ)-local PL condition (3.5) around its initialization for any $L \in \mathbb{N}^*$.

The time-independent dynamics

$$(A^L, Z_k^L, B^L) \mapsto \Big(-\frac{\partial \ell^L}{\partial A^L}, -L\frac{\partial \ell^L}{\partial Z_k^L}, -\frac{\partial \ell^L}{\partial B^L}\Big)$$

defining the gradient flow (3.5) are locally Lipschitz continuous, hence the gradient flow is defined on a maximal interval $[0, T_{\max})$ by the Picard-Lindelöf theorem (see Lemma 3.19). Let us show by contradiction that $T_{\max} = \infty$. Assume that $T_{\max} < \infty$. If this is true, again by the Picard-Lindelöf theorem, we know that the parameters diverge to infinity at $T_{\max}$. In particular, there exist $t \in (0, T_{\max})$ and $k \in \{1,\dots,L\}$ such that

$$\|A^L(t) - A^L(0)\|_F > M \text{ or } \|Z_k^L(t) - Z_k^L(0)\| > M \text{ or } \|B^L(t) - B^L(0)\|_F > M.$$

Let $t^* \in (0, T_{\max})$ be the infimum of such times t. Then, for $t < t^*$ and $k \in \{1,\dots,L\}$,

$$\|A^L(t) - A^L(0)\|_F \le M \text{ and } \|Z_k^L(t) - Z_k^L(0)\| \le M \text{ and } \|B^L(t) - B^L(0)\|_F \le M, \tag{3.23}$$

and, by continuity of A^L, B^L, and Z_k^L, these inequalities also hold for $t = t^*$. By definition, this means that the (M,μ)-local PL condition is satisfied for $t \le t^*$, and ensures that

$$\Big\|\frac{\partial \ell^L}{\partial A^L}(t)\Big\|_F^2 + L\sum_{k=1}^L \Big\|\frac{\partial \ell^L}{\partial Z_k^L}(t)\Big\|^2 + \Big\|\frac{\partial \ell^L}{\partial B^L}(t)\Big\|_F^2 \ge \mu \ell^L(t).$$

Therefore, by definition of the gradient flow (3.4),

$$\begin{aligned}
\frac{d\ell^L}{dt}(t) &= \Big\langle \frac{\partial \ell^L}{\partial A^L}(t), \frac{dA^L}{dt}(t)\Big\rangle + \sum_{k=1}^L \Big\langle \frac{\partial \ell^L}{\partial Z_k^L}(t), \frac{dZ_k^L}{dt}(t)\Big\rangle + \Big\langle \frac{\partial \ell^L}{\partial B^L}(t), \frac{dB^L}{dt}(t)\Big\rangle \\
&= -\Big\|\frac{\partial \ell^L}{\partial A^L}(t)\Big\|_F^2 - L\sum_{k=1}^L \Big\|\frac{\partial \ell^L}{\partial Z_k^L}(t)\Big\|^2 - \Big\|\frac{\partial \ell^L}{\partial B^L}(t)\Big\|_F^2 \\
&\le -\mu \ell^L(t).
\end{aligned}$$

Thus, by Grönwall's inequality, for $t \le t^*$,

$$\ell^L(t) \le e^{-\mu t}\ell^L(0). \tag{3.24}$$

Furthermore, by (3.23) and the definition of M_A, M_B, M_Z, we have, for $t \le t^*$ and $k \in \{1,\dots,L\}$,

$$\|A^L(t)\|_F \le M_A, \quad \|Z_k^L(t)\| \le M_Z, \quad \text{and} \quad \|B^L(t)\|_F \le M_B.$$

A quick scan through the proof of Proposition 3.8 reveals that by similar arguments, we have, for $t \le t^*$, $k \in \{1,\dots,L\}$, and $i \in \{1,\dots,n\}$,

$$(h_{k-1,i}^L(t), Z_k^L(t)) \in E \quad \text{and} \quad \|p_{k-1,i}^L(t)\| \le 2e^K \|p_{L,i}^L(t)\| \le 2e^K M_B \|F^L(x_i;t) - y_i\|.$$

Thus, for $k \in \{0, \dots, L\}$,

$$\frac{1}{n}\sum_{i=1}^{n} \|p_{k,i}^L(t)\| \leq \frac{2e^K M_B}{n} \sum_{i=1}^{n} \|F^L(x_i;t) - y_i\| \leq 2e^K M_B \sqrt{\ell^L(t)} \leq 2e^K M_B e^{-\frac{\mu t}{2}} \sqrt{\ell^L(0)}, \quad (3.25)$$

where the second inequality is a consequence of the Cauchy-Schwartz inequality. Let us now bound $\|Z_k^L(t^*) - Z_k^L(0)\|$. We have, for $k \in \{1, \dots, L\}$,

$$\begin{aligned}
\|Z_k^L(t^*) - Z_k^L(0)\| &\leq \int_0^{t^*} \Big\| \frac{dZ_k^L}{dt}(t) \Big\| dt \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \int_0^{t^*} \big\| \partial_2 f(h_{k-1,i}^L(t), Z_k^L(t))^\top p_{k-1,i}^L(t) \big\| dt \\
&\quad \text{(by (3.15)).} \\
&\leq \frac{K}{n}\sum_{i=1}^{n} \int_0^{t^*} \|p_{k-1,i}^L(t)\| dt,
\end{aligned}$$

since $(h_{k-1,i}^L(t), Z_k^L(t)) \in E$ and $\|\partial_2 f(h,z)\| \leq K$ for $(h,z) \in E$. Therefore, by (3.25),

$$\|Z_k^L(t^*) - Z_k^L(0)\| \leq 2Ke^K M_B \int_0^{t^*} e^{-\frac{\mu t}{2}} \sqrt{\ell^L(0)} dt \leq \frac{4Ke^K M_B}{\mu} \sqrt{\ell^L(0)} \leq \frac{M}{2},$$

where the last inequality is a consequence of the definition of μ. Similarly, by (3.14) and (3.25),

$$\begin{aligned}
\|A^L(t^*) - A^L(0)\|_F &\leq \int_0^{t^*} \Big\| \frac{dA^L}{dt}(t) \Big\|_F dt \\
&\leq \int_0^{t^*} \frac{1}{n}\sum_{i=1}^{n} \big\| p_{0,i}^L(t) x_i^\top \big\|_F dt \\
&\leq 2e^K M_B M_X \sqrt{\ell^L(0)} \int_0^{t^*} e^{-\frac{\mu t}{2}} dt \\
&\leq \frac{4e^K M_B M_X}{\mu} \sqrt{\ell^L(0)} \\
&\leq \frac{M}{2}.
\end{aligned}$$

Finally, by (3.16),

$$\begin{aligned}
\|B^L(t^*) - B(0)\|_F &\leq \int_0^{t^*} \Big\| \frac{dB^L}{dt}(t) \Big\|_F dt \\
&\leq \int_0^{t^*} \frac{2}{n}\sum_{i=1}^{n} \|(F^L(x_i;t) - y_i) h_{L,i}^L(t)^\top\|_F dt \\
&\leq 2e^K M_A M_X \sqrt{\ell^L(0)} \int_0^{t^*} e^{-\frac{\mu t}{2}} dt \\
&\leq \frac{4e^K M_A M_X}{\mu} \sqrt{\ell^L(0)} \\
&\leq \frac{M}{2},
\end{aligned}$$

where the third inequality is a consequence of the Cauchy-Schwartz inequality and of the fact that $\|h^L_{L,i}(t)\| \leq e^K M_A M_X$. By continuity of A^L, Z^L_k, and B^L, these three bounds contradict the definition of t^*. We conclude that $T_{\max} = \infty$ and that the parameters stay within a ball of radius M of their initialization, yielding the inequalities, for $t \in \mathbb{R}_+$, $L \in \mathbb{N}^*$, and $k \in \{1, \dots, L\}$,

$$\|A^L(t)\|_F \leq M_A, \quad \|B^L(t)\|_F \leq M_B, \quad \|Z^L_k(t)\| \leq M_Z.$$

This proves statement (i) of the proposition. Moreover, the analysis above show that the derivatives of A^L, Z^L_k, and B^L are bounded by a bounded integrable function independent of L and k. This shows (iii), together with the fact that the functions $A^L(t)$, $Z^L_k(t)$, and $B^L(t)$ admit limits as $t \to \infty$. Furthermore, the convergence towards their limit is uniform over L and k, as we show for example for $A^L(t)$. If we denote by A^L_∞ its limit, and apply the same steps as for bounding $\|A^L(t^*) - A^L(0)\|_F$, we obtain, for any $t \geq 0$,

$$\begin{aligned}
\|A^L_\infty - A^L(t)\|_F &\leq \int_t^\infty \Big\| \frac{dA^L}{d\tau}(\tau) \Big\|_F d\tau \\
&\leq 2e^K M_B M_X \sqrt{\ell^L(0)} \int_t^\infty e^{\frac{-\mu\tau}{2}} d\tau \\
&= \frac{4e^K M_B M_X}{\mu} e^{\frac{-\mu t}{2}} \sqrt{\ell^L(0)} \\
&\leq \frac{M}{2} e^{\frac{-\mu t}{2}},
\end{aligned}$$

where the last inequality comes from the definition of μ. The bound is independent of L, proving statement (iv). Statement (v) readily follows from (3.24).

To complete the proof, it remains to prove statement (ii) by bounding the differences $\|Z^L_{k+1}(t) - Z^L_k(t)\|$. Now that we know that the weights are bounded, we can follow the same steps as in the proof of Proposition 3.8 and show the existence of C_1, $C_2 > 0$ such that

$$\Big\| \frac{dZ^L_{k+1}}{dt}(t) - \frac{dZ^L_k}{dt}(t) \Big\| \leq \frac{1}{n} \sum_{i=1}^n \|p^L_{k,i}(t)\| \Big(\frac{C_1}{L} + C_2 \|Z^L_{k+1}(t) - Z^L_k(t)\| \Big).$$

Using (3.25), we obtain

$$\Big\| \frac{dZ^L_{k+1}}{dt}(t) - \frac{dZ^L_k}{dt}(t) \Big\| \leq 2e^K M_B e^{-\frac{\mu t}{2}} \sqrt{\ell^L(0)} \Big(\frac{C_1}{L} + C_2 \|Z^L_{k+1}(t) - Z^L_k(t)\| \Big).$$

Integrating between 0 and t, we obtain

$$\begin{aligned}
\|Z^L_{k+1}(t) - Z^L_k(t)\| &\leq \|Z^L_{k+1}(0) - Z^L_k(0)\| + \int_0^t 2e^K M_B e^{-\frac{\mu\tau}{2}} \sqrt{\ell^L(0)} \frac{C_1}{L} d\tau \\
&\quad + \int_0^t 2e^K M_B e^{-\frac{\mu\tau}{2}} \sqrt{\ell^L(0)} C_2 \|Z^L_{k+1}(\tau) - Z^L_k(\tau)\| d\tau \\
&\leq \|Z^L_{k+1}(0) - Z^L_k(0)\| + \frac{C_1 M}{2M_X L} \\
&\quad + \int_0^t 2e^K M_B e^{-\frac{\mu\tau}{2}} \sqrt{\ell^L(0)} C_2 \|Z^L_{k+1}(\tau) - Z^L_k(\tau)\| d\tau,
\end{aligned}$$

where the second inequality uses the definition of μ. By Grönwall's inequality,

$$\begin{aligned}
\|Z^L_{k+1}(t) - Z^L_k(t)\| &\leq \Big(\|Z^L_{k+1}(0) - Z^L_k(0)\| + \frac{C_1 M}{2M_X L} \Big) \exp\Big(\int_0^t 2e^K M_B e^{-\frac{\mu\tau}{2}} \sqrt{\ell^L(0)} C_2 d\tau \Big) \\
&\leq \Big(\|Z^L_{k+1}(0) - Z^L_k(0)\| + \frac{C_1 M}{2M_X L} \Big) \exp\Big(\frac{C_2 M}{2M_X} \Big),
\end{aligned}$$

again by definition of μ. Finally, since $Z_k^L(0) = Z^{\text{init}}(\frac{k}{L})$ and Z^{init} is Lipschitz continuous, this proves the existence of $\tilde{K} > 0$ (independent of L, t and k) such that $\|Z_{k+1}^L(t) - Z_k^L(t)\| \leq \frac{\tilde{K}}{L}$, which yields statement (ii). □

3.A.3 Generalized Arzelà–Ascoli theorem

Proposition 3.10 (Generalized Arzelà–Ascoli theorem)**.** *Let $I \subseteq \mathbb{R}_+$ be an interval. We denote by $(Z_k^L)_{L\in\mathbb{N}^*,1\leq k\leq L}$ be a family of $\mathcal{C}^1$ functions from I to $\mathbb{R}^p$. Define*

$$\mathcal{Z}^L : [0,1] \times I \to \mathbb{R}^p,\ (s,t) \mapsto \mathcal{Z}^L(s,t) = Z^L_{\lfloor (L-1)s \rfloor + 1}(t).$$

Assume that there exist a constant $C > 0$ and a bounded integrable function $b : I \to \mathbb{R}$ such that the following statements hold for any $t \in I$ and $L \in \mathbb{N}^*$:

(i) For $k \in \{1, \dots, L-1\}$, $\|Z_{k+1}^L(t) - Z_k^L(t)\| \leq \frac{C}{L}$,

(ii) For $k \in \{1, \dots, L\}$, $\|Z_k^L(t)\| \leq C$ and $\|\frac{dZ_k^L}{dt}(t)\| \leq b(t)$.

Then there exist a subsequence $(\mathcal{Z}^{\phi(L)})_{L\in\mathbb{N}^*}$ of $(\mathcal{Z}^L)_{L\in\mathbb{N}^*}$ and a Lipschitz continuous function $\mathcal{Z}^\phi : [0,1] \times I \to \mathbb{R}^p$ such that $\mathcal{Z}^{\phi(L)}(s,t)$ tends to $\mathcal{Z}^\phi(s,t)$ uniformly over s and t.

Note that if I is a compact interval, then the existence of a (uniformly) convergent subsequence is guaranteed by the standard Arzelà–Ascoli theorem. Indeed, the uniform equicontinuity is a consequence of assumptions (i) and (ii), while (ii) provides a uniform bound. However, if I is not compact, more involved arguments are needed.

Proof. Assume, without loss of generality, that b is also bounded by C. According to assumption (i), for $t \in I$ and $i, j \in \{1, \dots, L\}$,

$$\|Z_i^L(t) - Z_j^L(t)\| \leq \frac{C|i-j|}{L}.$$

Also, according to (ii), for $t, t' \in I$ and $k \in \{1, \dots, L\}$,

$$\|Z_k^L(t) - Z_k^L(t')\| = \Big\| \int_{t'}^{t} \frac{dZ_k^L}{d\tau}(\tau) d\tau \Big\| \leq C|t - t'|.$$

It follows that, for $s, s' \in [0,1]$ and $t, t' \in I$,

$$\begin{aligned}\|\mathcal{Z}^L(s,t) - \mathcal{Z}^L(s',t')\| &\leq \|\mathcal{Z}^L(s,t) - \mathcal{Z}^L(s,t')\| + \|\mathcal{Z}^L(s,t') - \mathcal{Z}^L(s',t')\| \\ &\leq C|t-t'| + \frac{C|\lfloor (L-1)s \rfloor - \lfloor (L-1)s' \rfloor|}{L}.\end{aligned}$$

Therefore, with some simple algebra, we obtain

$$\|\mathcal{Z}^L(s,t) - \mathcal{Z}^L(s',t')\| \leq C|t-t'| + C|s-s'| + \frac{C}{L}. \tag{3.26}$$

The statement of the proposition is then a consequence of the next three steps.

There exists a convergent subsequence of $(\mathcal{Z}^L(s,t))_{L\in\mathbb{N}^*}$. First, let $((s_i,t_i))_{i\in\mathbb{N}} = (\mathbb{Q}\cap[0,1])\times(\mathbb{Q}\cap I)$. By (ii), the sequence $(\mathcal{Z}^L(s_i,t_i))_{L\in\mathbb{N}^*,i\in\mathbb{N}}$ is bounded. It is therefore possible to construct by a diagonal procedure a subsequence $(\mathcal{Z}^{\phi(L)})_{L\in\mathbb{N}^*}$ such that, for each $i\in\mathbb{N}$, $(\mathcal{Z}^{\phi(L)}(s_i,t_i))_{L\in\mathbb{N}^*}$ is a convergent sequence.

Let us now show that $(\mathcal{Z}^{\phi(L)}(s,t))_{L\in\mathbb{N}^*}$ converges for any $s\in[0,1]$ and $t\in I$, by proving that it is a Cauchy sequence in the complete metric space $\mathbb{R}^p$. Let $\varepsilon>0$, $s\in[0,1]$, and $t\in I$. Since $((s_i,t_i))_{i\in\mathbb{N}}$ is dense in $[0,1]\times I$, there exists some $j\in\mathbb{N}$ such that $|s_j-s|\le\varepsilon$ and $|t_j-t|\le\varepsilon$. Then, for $L,M\in\mathbb{N}^*$, we have

$$\begin{aligned}\|\mathcal{Z}^{\phi(L)}(s,t)-\mathcal{Z}^{\phi(M)}(s,t)\|&\\ &\le\|\mathcal{Z}^{\phi(L)}(s,t)-\mathcal{Z}^{\phi(L)}(s_j,t_j)\|+\|\mathcal{Z}^{\phi(L)}(s_j,t_j)-\mathcal{Z}^{\phi(M)}(s_j,t_j)\|\\ &\quad+\|\mathcal{Z}^{\phi(M)}(s_j,t_j)-\mathcal{Z}^{\phi(M)}(s,t)\|\\ &\le 2C\varepsilon+\frac{C}{\phi(L)}+\|\mathcal{Z}^{\phi(L)}(s_j,t_j)-\mathcal{Z}^{\phi(M)}(s_j,t_j)\|+2C\varepsilon+\frac{C}{\phi(M)},\end{aligned}$$

where we used inequality (3.26) twice. Since $(\mathcal{Z}^{\phi(L)}(s_j,t_j))_{L\in\mathbb{N}^*}$ is a convergent sequence, it is a Cauchy sequence. Thus, the bound can be made arbitrarily small for L,M large enough. This shows that $(\mathcal{Z}^{\phi(L)}(s,t))_{L\in\mathbb{N}^*}$ is also a Cauchy sequence. It is therefore convergent, and we denote by $\mathcal{Z}^{\phi}(s,t)$ its limit.

The function $\mathcal{Z}^\phi$ is Lipschitz continuous. By considering (3.26) for the subsequence $\phi(L)$ and letting $L\to\infty$, we have that, for any $s,s'\in[0,1]$ and $t,t'\in I$,

$$\|\mathcal{Z}^{\phi}(s,t)-\mathcal{Z}^{\phi}(s',t')\|\le C(|s-s'|+|t-t'|). \tag{3.27}$$

The convergence of $(\mathcal{Z}^{\phi(L)}(s,t))_{L\in\mathbb{N}^*}$ to $\mathcal{Z}^\phi(s,t)$ is uniform over s and t. Let $\varepsilon>0$, $s\in[0,1]$, and $t\in I$. Then, by (3.26) and (3.27), it is possible to find $\delta>0$ such that, for any $s',s''\in[0,1]$ and $t',t''\in I$ satisfying $|s'-s''|\le\delta$ and $|t'-t''|\le\delta$,

$$\|\mathcal{Z}^{\phi(L)}(s',t')-\mathcal{Z}^{\phi(L)}(s'',t')\|\le\varepsilon+\frac{C}{\phi(L)}\quad\text{and}\quad\|\mathcal{Z}^{\phi}(s',t')-\mathcal{Z}^{\phi}(s'',t')\|\le\varepsilon, \tag{3.28}$$

and

$$\|\mathcal{Z}^{\phi(L)}(s',t')-\mathcal{Z}^{\phi(L)}(s',t'')\|\le\varepsilon+\frac{C}{\phi(L)}\quad\text{and}\quad\|\mathcal{Z}^{\phi}(s',t')-\mathcal{Z}^{\phi}(s',t'')\|\le\varepsilon. \tag{3.29}$$

Furthermore, there exists a finite set $\{s_1,\dots,s_S\}\subset[0,1]$ such that

$$[0,1]\subset\bigcup_{i=1}^{S}(s_i-\delta,s_i+\delta).$$

In the sequel, we denote by s^* an element of $\{s_1,\dots,s_S\}$ that is at distance at most δ from s.

If I is unbounded, then, by assumption (ii) and since b is integrable, there exists some $t_0>0$ such that, for $t\ge t_0$,

$$\|\mathcal{Z}^{\phi(L)}(s,t)-\mathcal{Z}^{\phi(L)}(s,t_0)\|\le\int_{t_0}^{t}\Big\|\frac{d}{dt}Z^{\phi(L)}_{\lfloor(\phi(L)s-1)\rfloor+1}(\tau)\Big\|d\tau\le\int_{t_0}^{t}b(\tau)d\tau\le\varepsilon. \tag{3.30}$$

The same inequality holds for $\mathcal{Z}^\phi$ by letting L tend to infinity. If I is bounded, we simply let $t_0=\sup I$.

We may then pick a finite set $\{t_1, \dots, t_T\} \subset [0, t_0]$ such that

$$[0, t_0] \subset \bigcup_{i=1}^{T} (t_i - \delta, t_i + \delta).$$

Two cases may arise depending on the value of t. If $t \in [0, t_0]$, then there exists an element of the set $\{t_1, \dots, t_T\}$ at distance at most δ from t, and we denote it by t^*. If $t > t_0$, we let $t^* = t_0$. According to (3.29) and (3.30), we then have in both cases that

$$\|\mathcal{Z}^{\phi(L)}(s,t) - \mathcal{Z}^{\phi(L)}(s,t^*)\| \leqslant \varepsilon + \frac{C}{\phi(L)} \quad \text{and} \quad \|\mathcal{Z}^{\phi}(s,t) - \mathcal{Z}^{\phi}(s,t^*)\| \leqslant \varepsilon. \tag{3.31}$$

To conclude, we have to bound the term $\|\mathcal{Z}^{\phi(L)}(s,t) - \mathcal{Z}^{\phi}(s,t)\|$ uniformly over s and t. We first have

$$\begin{aligned}
&\|\mathcal{Z}^{\phi(L)}(s,t) - \mathcal{Z}^{\phi}(s,t)\| \\
&\qquad \leq \|\mathcal{Z}^{\phi(L)}(s,t) - \mathcal{Z}^{\phi(L)}(s,t^*)\| + \|\mathcal{Z}^{\phi(L)}(s,t^*) - \mathcal{Z}^{\phi}(s,t^*)\| \\
&\qquad\quad + \|\mathcal{Z}^{\phi}(s,t^*) - \mathcal{Z}^{\phi}(s,t)\| \\
&\qquad \leq 2\varepsilon + \frac{C}{\phi(L)} + \|\mathcal{Z}^{\phi(L)}(s,t^*) - \mathcal{Z}^{\phi}(s,t^*)\|,
\end{aligned}$$

where the last inequality is a consequence of (3.31). The last term can be bounded as follows:

$$\begin{aligned}
&\|\mathcal{Z}^{\phi(L)}(s,t^*) - \mathcal{Z}^{\phi}(s,t^*)\| \\
&\qquad \leq \|\mathcal{Z}^{\phi(L)}(s,t^*) - \mathcal{Z}^{\phi(L)}(s^*,t^*)\| + \|\mathcal{Z}^{\phi(L)}(s^*,t^*) - \mathcal{Z}^{\phi}(s^*,t^*)\| \\
&\qquad\quad + \|\mathcal{Z}^{\phi}(s^*,t^*) - \mathcal{Z}^{\phi}(s,t^*)\| \\
&\qquad \leq 2\varepsilon + \frac{C}{\phi(L)} + \max_{i \in \{1,\dots,S\}} \|\mathcal{Z}^{\phi(L)}(s_i,t^*) - \mathcal{Z}^{\phi}(s_i,t^*)\|,
\end{aligned}$$

by using (3.28) and the fact that $s^* \in \{s_1, \dots, s_S\}$. Putting all the pieces together, we finally obtain

$$\|\mathcal{Z}^{\phi(L)}(s,t) - \mathcal{Z}^{\phi}(s,t)\| \leq 4\varepsilon + \frac{2C}{\phi(L)} + \max_{i \in \{1,\dots,S\}, j \in \{1,\dots,T\}} \|\mathcal{Z}^{\phi(L)}(s_i,t_j) - \mathcal{Z}^{\phi}(s_i,t_j)\|.$$

By taking L large enough, independent of s and t, the sum of the last two terms can be made less than ε. Since ε is arbitrary, this concludes the proof. □

A consequence of this result is a simplified version for sequences of functions only indexed by L and not k, as follows.

Corollary 3.11. *Let $I \subseteq \mathbb{R}_+$ be an interval, and $(Z^L)_{L \in \mathbb{N}^*}$ be a family of $\mathcal{C}^1$ functions from I to $\mathbb{R}^p$. Assume that there exist a constant $C > 0$ and a bounded integrable function $b : I \to \mathbb{R}$ such that, for any $t \in I$ and $L \in \mathbb{N}^*$, $\|Z^L(t)\| \leq C$ and $\|\frac{dZ^L}{dt}(t)\| \leq b(t)$. Then there exist a subsequence $(Z^{\phi(L)})_{L \in \mathbb{N}^*}$ of $(Z^L)_{L \in \mathbb{N}^*}$ and a function $Z^\phi : I \to \mathbb{R}^p$ such that $Z^{\phi(L)}(t)$ tends to $Z^\phi(t)$ uniformly over t.*

3.A.4 Consistency of the Euler scheme for parameterized ODEs

Proposition 3.12 (Consistency of the Euler scheme for parameterized ODEs.)**.** *We denote by $(\theta_k^L)_{L\in\mathbb{N}^*,1\leq k\leq L}$ be a bounded family of vectors of $\mathbb{R}^p$, and let*

$$\Theta^L : [0,1] \to \mathbb{R}^p,\ s \mapsto \theta^L_{\lfloor (L-1)s\rfloor+1}.$$

Assume that there exists $\Theta : [0,1] \to \mathbb{R}^p$ a Lipschitz continuous function such that $\Theta^L(s)$ tends to $\Theta(s)$ uniformly over s. Let $(a^L)_{L\in\mathbb{N}^*}$ be a sequence of vectors in some compact $E \subset \mathbb{R}^d$ converging to $a \in E$. Let $g : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ be a $\mathcal{C}^1$ function such that $g(0,\cdot) \equiv 0$ and $g(\cdot,\theta)$ is uniformly Lipschitz continuous for θ in any compact of $\mathbb{R}^p$. Consider the discrete scheme

$$\begin{aligned} u_0^L &= a^L \\ u_{k+1}^L &= u_k^L + \frac{1}{L} g(u_k^L, \theta_{k+1}^L), \quad k \in \{0,\dots,L-1\}. \end{aligned} \tag{3.32}$$

Then $u^L_{\lfloor Ls\rfloor}$ tends to $U(s)$ uniformly over $s \in [0,1]$, where U is the unique solution of the ODE

$$\begin{aligned} U(0) &= a \\ \frac{dU}{ds}(s) &= g(U(s), \Theta(s)), \quad s \in [0,1]. \end{aligned} \tag{3.33}$$

Moreover, the convergence only depends on the sequence $(a^L)_{L\in\mathbb{N}^*}$ and on its limit $a \in E$ through $(\|a^L - a\|)_{L\in\mathbb{N}^*}$.

Proof. Let M be a bound of the sequence $(\theta_k^L)_{L\in\mathbb{N}^*,1\leq k\leq L}$. By definition of Θ^L, the sequence $(\Theta^L)_{L\in\mathbb{N}^*}$ is also uniformly bounded by M, and the same is true for Θ. Then the function $g(\cdot,\Theta(s))$ is uniformly Lipschitz for $s \in [0,1]$. Furthermore, $(U,s) \mapsto g(U,\Theta(s))$ is continuous in s because g and Θ are continuous. Thus the ODE (3.33) has a unique solution on $[0,1]$ by the Picard-Lindelöf theorem (see Lemma 3.19).

Denote by C the uniform Lipschitz constant of $g(\cdot,\theta)$ for $\|\theta\| \leq M$. Since $g(0,\cdot) \equiv 0$ and $g(\cdot,\Theta(s))$ is C-Lipschitz, one has

$$\Big\|\frac{dU}{ds}(s)\Big\| = \|g(U(s),\Theta(s))\| \leq C\|U(s)\|.$$

Therefore, by Grönwall's inequality,

$$\|U(s)\| \leq \|U(0)\|\exp(C) = \|a\|\exp(C) \leq D_E \exp(C),$$

where $D_E = \sup_{x\in E}\|x\| < \infty$. A similar reasoning applies to the discrete scheme (3.32), using the discrete version of Grönwall's inequality. More precisely, for any $k \in \{0,\dots,L-1\}$,

$$\|u_{k+1}^L\| \leq \|u_k^L\| + \frac{1}{L}\|g(u_k^L,\theta_{k+1}^L)\| \leq \Big(1+\frac{C}{L}\Big)\|u_k^L\|.$$

Thus,

$$\|u_k^L\| \leq \|u_0^L\|\exp(C) = \|a^L\|\exp(C) \leq D_E\exp(C).$$

Overall, we can consider a restriction of g to a compact set depending only on M, C, and E, which we will still denote by g with a slight abuse of notation.Since g is $\mathcal{C}^1$, it is therefore bounded and Lipschitz continuous, and we still let C be its Lipschitz constant.

For $L \in \mathbb{N}^*$ and $k \in \{0,\dots,L\}$, we denote by Δ_k^L the gap between the continuous and the discrete schemes, i.e.,

$$\Delta_k^L = \Big\|U\Big(\frac{k}{L}\Big) - u_k^L\Big\|.$$

The next step is to recursively bound the size of this gap, first observing that $\Delta_0^L = \|a^L - a\|$. We have that

$$s \mapsto \frac{dU}{ds}(s) = g(U(s), \Theta(s)) \tag{3.34}$$

is a Lipschitz continuous function with some Lipschitz constant $\tilde{C}$. To see this, just note that U itself is Lipschitz continuous in s, since g is bounded, and therefore the function (3.34) is a composition of Lipschitz continuous functions. In particular, $\frac{dU}{ds}$ is almost everywhere differentiable, and its derivative $\frac{d^2U}{ds^2}(s)$ is bounded in the supremum norm by $\tilde{C}$. As a consequence, for $k \in \{0, \dots, L-1\}$, the Taylor expansion of U on $[\frac{k}{L}, \frac{k+1}{L}]$ takes the form

$$U\Big(\frac{k+1}{L}\Big) = U\Big(\frac{k}{L}\Big) + \frac{1}{L}\frac{dU}{ds}\Big(\frac{k}{L}\Big) + \int_{k/L}^{(k+1)/L} \Big(\frac{k+1}{L} - s\Big)\frac{d^2U}{ds^2}(s)ds,$$

where the norm of the remainder term is less than $\tilde{C}/L^2$. Therefore,

$$\begin{aligned}
\Delta_{k+1}^L &= \Big\|U\Big(\frac{k+1}{L}\Big) - u_{k+1}^L\Big\| \\
&= \Big\|U\Big(\frac{k}{L}\Big) + \frac{1}{L}g\Big(U\Big(\frac{k}{L}\Big), \Theta\Big(\frac{k}{L}\Big)\Big) + \int_{k/L}^{(k+1)/L} \Big(\frac{k+1}{L} - s\Big)\frac{d^2U}{ds^2}(s)ds \\
&\quad - u_k^L - \frac{1}{L}g(u_k^L, \theta_{k+1}^L)\Big\| \\
&\leq \Big\|U\Big(\frac{k}{L}\Big) - u_k^L\Big\| + \Big\|\frac{1}{L}g\Big(U\Big(\frac{k}{L}\Big), \Theta\Big(\frac{k}{L}\Big)\Big) - \frac{1}{L}g(u_k^L, \theta_{k+1}^L)\Big\| \\
&\quad + \int_{k/L}^{(k+1)/L} \Big(\frac{k+1}{L} - s\Big)\Big\|\frac{d^2U}{ds^2}(s)\Big\|ds \\
&\leqslant \Delta_k^L + \frac{C}{L}\Delta_k^L + \frac{C}{L}\Big\|\Theta\Big(\frac{k}{L}\Big) - \theta_{k+1}^L\Big\| + \frac{\tilde{C}}{L^2}.
\end{aligned}$$

In the last inequality, we used the fact that g is C-Lipschitz. Since,by definition, $\theta_{k+1}^L = \Theta^L(\frac{k}{L-1})$, we obtain, for $k \in \{0, \dots, L-1\}$,

$$\begin{aligned}
\Delta_{k+1}^L &\leqslant \Big(1 + \frac{C}{L}\Big)\Delta_k^L + \frac{C}{L}\Big\|\Theta\Big(\frac{k}{L}\Big) - \Theta^L\Big(\frac{k}{L-1}\Big)\Big\| + \frac{\tilde{C}}{L^2} \\
&\leqslant \Big(1 + \frac{C}{L}\Big)\Delta_k^L + \frac{C}{L}\sup_{s\in[0,1]}\|\Theta(s) - \Theta^L(s)\| + \frac{C}{L}\Big\|\Theta\Big(\frac{k}{L}\Big) - \Theta\Big(\frac{k}{L-1}\Big)\Big\| + \frac{\tilde{C}}{L^2} \\
&\leqslant \Big(1 + \frac{C}{L}\Big)\Delta_k^L + \frac{C}{L}\sup_{s\in[0,1]}\|\Theta(s) - \Theta^L(s)\| + \frac{CC_\Theta}{L^2} + \frac{\tilde{C}}{L^2},
\end{aligned}$$

where C_Θ is the Lipschitz constant of Θ. By the discrete Grönwall's inequality, we deduce that, for $k \in \{0, \dots, L-1\}$,

$$\begin{aligned}
\Delta_{k+1}^L &\leqslant \Big(\Delta_0^L + \sup_{s\in[0,1]}\|\Theta(s) - \Theta^L(s)\| + \frac{C_\Theta}{L} + \frac{\tilde{C}}{LC}\Big)e^C \\
&= \Big(\|a^L - a\| + \sup_{s\in[0,1]}\|\Theta(s) - \Theta^L(s)\| + \frac{C_\Theta}{L} + \frac{\tilde{C}}{LC}\Big)e^C.
\end{aligned} \tag{3.35}$$

This shows that the gaps Δ_k^L converge to zero uniformly over $k \in \{0, \dots, L\}$ as L tends to infinity.

We conclude by observing that, for any $s \in [0, 1]$,

$$\|U(s) - u_{\lfloor Ls\rfloor}^L\| \leqslant \Big\|U(s) - U\Big(\frac{\lfloor Ls\rfloor}{L}\Big)\Big\| + \Big\|U\Big(\frac{\lfloor Ls\rfloor}{L}\Big) - u_{\lfloor Ls\rfloor}^L\Big\| \leq \frac{C_U}{L} + \Delta_{\lfloor Ls\rfloor}^L, \tag{3.36}$$

where C_U is the Lipschitz constant of U. Both terms converge to zero uniformly over s as L tends to infinity. Finally, an inspection of our bounds shows that the convergence only depends on $(a^L)_{L\in\mathbb{N}^*} \in E^{\mathbb{N}^*}$ through $\|a^L - a\|$. □

The results of Proposition 3.12 can be extended without much effort to two other related cases. First, the parameters θ_k^L may depend on some other variable t, as long as all assumptions are verified uniformly over t. Second, these parameters may converge to some limit parameters as both L and t go to infinity. This is encapsulated in the following two corollaries.

Corollary 3.13. *Let $I \subseteq \mathbb{R}_+$ be an interval. Let $(\theta_k^L)_{L\in\mathbb{N}^*,1\leq k\leq L}$ be a uniformly bounded family of functions from I to $\mathbb{R}^p$, and let*

$$\Theta^L : [0,1] \times I \to \mathbb{R}^p,\ (s,t) \mapsto \theta^L_{\lfloor (L-1)s\rfloor+1}(t).$$

Assume that there exists a function $\Theta : [0,1] \times I \to \mathbb{R}^p$ such that $\Theta^L(s,t)$ tends to $\Theta(s,t)$ uniformly over s and t, and $\Theta(\cdot,t)$ is uniformly Lipschitz continuous for $t \in I$. Let $(a^L)_{L\in\mathbb{N}^*}$ be a family of functions from I to some compact $E \subset \mathbb{R}^d$, uniformly converging to $a : I \to E$. Let $g : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ be a $\mathcal{C}^1$ function such that $g(0,\cdot) \equiv 0$ and $g(\cdot,\theta)$ is uniformly Lipschitz continuous for θ in any compact of $\mathbb{R}^p$. Consider the discrete scheme, for $t \in I$,

$$\begin{aligned} u_0^L(t) &= a^L(t) \\ u_{k+1}^L(t) &= u_k^L(t) + \frac{1}{L} g(u_k^L(t), \theta_{k+1}^L(t)), \quad k \in \{0,\dots,L-1\}. \end{aligned}$$

Then $u^L_{\lfloor Ls\rfloor}(t)$ tends to $U(s,t)$ uniformly over $s \in [0,1]$ and $t \in I$, where $U(\cdot,t)$ is the unique solution of the ODE

$$\begin{aligned} U(0,t) &= a(t) \\ \frac{\partial U}{\partial s}(s,t) &= g(U(s,t), \Theta(s,t)), \quad s \in [0,1]. \end{aligned}$$

Moreover, the convergence only depends on the sequence $(a^L)_{L\in\mathbb{N}^*}$ and on its limit $a \in E^I$ through $(\sup_{t\in I} \|a^L(t) - a(t)\|)_{L\in\mathbb{N}^*}$.

Corollary 3.14. *Let $I \subseteq \mathbb{R}_+$ be an interval. Let $(\theta_k^L)_{L\in\mathbb{N}^*,1\leq k\leq L}$ be a uniformly bounded family of functions from I to $\mathbb{R}^p$, and let*

$$\Theta^L : [0,1] \times \mathbb{R}_+ \to \mathbb{R}^p,\ (s,t) \mapsto \theta^L_{\lfloor (L-1)s\rfloor+1}(t).$$

Assume that there exists a function $\Theta_\infty : [0,1] \to \mathbb{R}^p$ such that $\Theta^L(s,t)$ tends to $\Theta_\infty(s)$ uniformly over s as $L,t \to \infty$, and Θ_∞ is Lipschitz continuous. Let $(a^L)_{L\in\mathbb{N}^*}$ be a family of functions from I to some compact $E \subset \mathbb{R}^d$, and converging to $a_\infty \in E$ as $L,t \to \infty$. Let $g : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ be a $\mathcal{C}^1$ function such that $g(0,\cdot) \equiv 0$ and $g(\cdot,\theta)$ is uniformly Lipschitz continuous for θ in any compact of $\mathbb{R}^p$. Consider the discrete scheme, for $t \in I$,

$$\begin{aligned} u_0^L(t) &= a^L(t) \\ u_{k+1}^L(t) &= u_k^L(t) + \frac{1}{L} g(u_k^L(t), \theta_{k+1}^L(t)), \quad k \in \{0,\dots,L-1\}. \end{aligned}$$

Then $u^L_{\lfloor Ls\rfloor}(t)$ tends to $U(s)$ uniformly over $s \in [0,1]$ as $L,t \to \infty$, where U is the unique solution of the ODE

$$\begin{aligned} U(0) &= a_\infty \\ \frac{dU}{ds}(s) &= g(U(s), \Theta_\infty(s)), \quad s \in [0,1]. \end{aligned}$$

Moreover, the convergence only depends on the sequence $(a^L)_{L\in\mathbb{N}^*}$ and on its limit $a \in E^I$ through $(\sup_{t\in I} \|a^L(t) - a(t)\|)_{L\in\mathbb{N}^*}$.

3.A.5 Large-depth convergence of the gradient flow

This section is devoted to proving the main result of Appendix 3.A, namely the large-depth convergence of the gradient flow. The setting we consider encompasses both Section 3.4.1 (finite training time and clipped gradient flow) and Section 3.4.2 (arbitrary training time and standard gradient flow). To this end, we consider a training interval $I = [0, T] \subseteq \mathbb{R}_+$, for $T \leq \infty$, and the gradient flow formulation (3.5), which is equivalent to the standard gradient flow (3.4) if π equals the identity. Note that we do not need to assume in the following proof that π is bounded (but only Lipschitz continuous). Therefore, the proof also holds in the case where π equals the identity.

Theorem 3.15. *Consider the residual network* (3.12) *initialized as explained in Appendix 3.A and trained with the gradient flow* (3.5) *on $I = [0, T] \subseteq \mathbb{R}_+$, for some $T \in (0, \infty]$. Assume that there exists a unique solution to the gradient flow, such that $(A^L)_{L\in\mathbb{N}^*}$ and $(B^L)_{L\in\mathbb{N}^*}$ each satisfies the assumptions of Corollary 3.11, and $(Z_k^L)_{L\in\mathbb{N}^*, 1\leq k\leq L}$ satisfies the assumptions of Proposition 3.10. Then the following four statements hold **as L tends to infinity**:*

(i) There exist functions $A : I \to \mathbb{R}^{q\times d}$ and $B : I \to \mathbb{R}^{d'\times q}$ such that $A^L(t)$ and $B^L(t)$ converge uniformly over $t \in I$ to $A(t)$ and $B(t)$.

(ii) There exists a Lipschitz continuous function $\mathcal{Z} : [0, 1] \times I \to \mathbb{R}^p$ such that

$$\mathcal{Z}^L : [0, 1] \times I \to \mathbb{R}^p, \ (s, t) \mapsto \mathcal{Z}^L(s, t) = Z^L_{\lfloor (L-1)s\rfloor+1}(t)$$

converges uniformly over $s \in [0, 1]$ and $t \in I$ to $\mathcal{Z}(s, t)$.

(iii) Uniformly over $s \in [0, 1]$, $t \in I$, and $x \in \mathcal{X}$, the hidden layer $h^L_{\lfloor Ls\rfloor}(t)$ converges to the solution at time s of the neural ODE

$$\begin{aligned} H(0, t) &= A(t)x \\ \frac{\partial H}{\partial s}(s, t) &= f(H(s, t), \mathcal{Z}(s, t)), \quad s \in [0, 1]. \end{aligned}$$

(iv) Uniformly over $t \in I$ and $x \in \mathcal{X}$, the output $F^L(x; t)$ converges to $B(t)H(1, t)$.

Proof. According to Proposition 3.10, there exists a subsequence $(\mathcal{Z}^{\phi(L)})_{L\in\mathbb{N}^*}$ of $(\mathcal{Z}^L)_{L\in\mathbb{N}^*}$ and a Lipschitz continuous function $\mathcal{Z}^\phi : [0, 1] \times I \to \mathbb{R}^p$ such that $\mathcal{Z}^{\phi(L)}(s, t)$ tends to $\mathcal{Z}^\phi(s, t)$ uniformly over s and t. Similarly, by Corollary 3.11, there exists subsequences of $(A^L)_{L\in\mathbb{N}^*}$ and $(B^L)_{L\in\mathbb{N}^*}$ that converge uniformly. With a slight abuse of notation, we still denote these subsequences by ϕ, and the corresponding limits by A^ϕ and B^ϕ.

In the remainder, we prove the uniqueness of the accumulation point (Z^ϕ, A^ϕ, B^ϕ) by showing that it is the solution of an ODE that satisfies the assumptions of the Picard-Lindelöf theorem. The statements (i) to (iv) then follow easily.

Consider a general input $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and let $H^L(s, t) = h^L_{\lfloor Ls\rfloor}(t)$ (recall that $h^L_k(t)$ is defined by the forward propagation (3.12)). Corollary 3.13, with $\theta^L_k = Z^{\phi(L)}_k$, $\Theta = \mathcal{Z}^\phi$, $a^L = A^{\phi(L)}x$, $g = f$, ensures that $H^{\phi(L)}(s, t)$ converges uniformly (over s and t) to $H^\phi(s, t)$ that is the solution at time s of the ODE

$$\begin{aligned} H^\phi(0, t) &= A^\phi(t)x \\ \frac{\partial H^\phi}{\partial s}(s, t) &= f(H^\phi(s, t), \mathcal{Z}^\phi(s, t)), \quad s \in [0, 1]. \end{aligned}$$

By inspecting the proof of the corollary, we also have that $(h_k^{\phi(L)})_{L\in\mathbb{N}^*,1\leq k\leq\phi(L)}$ and $(H^{\phi(L)})_{L\in\mathbb{N}^*}$ are uniformly bounded and that $H^\phi(\cdot,t)$ is uniformly Lipschitz continuous for $t\in I$.

We now turn our attention to the backpropagation recurrence (3.13), which defines the backward state $p_k^L(t)$. First observe that the convergence of $H^{\phi(L)}$ implies that

$$p_{\phi(L)}^{\phi(L)}(t) = 2B^{\phi(L)}(t)^\top(B^{\phi(L)}(t)h_{\phi(L)}^{\phi(L)}(t) - y) = 2B^{\phi(L)}(t)^\top(B^{\phi(L)}(t)H^{\phi(L)}(1,t) - y)$$

converges uniformly to $2B^\phi(t)^\top(B^\phi(t)H^\phi(1,t)-y)\in\mathbb{R}^d$. Now, let $P^L(s,t) = p^L_{\lfloor Ls\rfloor}(t)$. We apply again Corollary 3.13, this time to the backpropagation recurrence (3.13), with $\theta_k^L = (h_k^{\phi(L)}, Z_k^{\phi(L)})$, $\Theta = (H^\phi, \mathcal{Z}^\phi)$, $g:(p,(h,Z))\mapsto \partial_1 f(h,Z)p$, and $a^L = 2(B^{\phi(L)})^\top(B^{\phi(L)}H^{\phi(L)}(1,\cdot)-y)$. Let us quickly check that the conditions of the corollary are met:

- The sequence $(h_k^{\phi(L)})_{L\in\mathbb{N}^*,1\leq k\leq\phi(L)}$ is bounded, as noted previously, and the same holds for $(Z_k^{\phi(L)})_{L\in\mathbb{N}^*,1\leq k\leq\phi(L)}$ by the assumptions of Theorem 3.15.
- The function $H^\phi(\cdot,t)$ is uniformly Lipschitz continuous for $t\in I$, as noted previously, and the same is true for $Z^\phi(\cdot,t)$ since Z^ϕ is Lipschitz continuous.
- The function $h^{\phi(L)}_{\lfloor(\phi(L)-1)s\rfloor+1}(t)$ tends to $H^\phi(s,t)$ uniformly over s and t, as seen in the beginning of the proof. More precisely, we know that $H^{\phi(L)}(s,t) = h^{\phi(L)}_{\lfloor\phi(L)s\rfloor}(t)$ tends to $H^\phi(s,t)$. Simple algebra and the fact that two successive iterates of (3.12) are separated by a distance proportional to $1/L$ show that both statements are equivalent. Furthermore, $\mathcal{Z}^{\phi(L)}(s,t)$ tends to $\mathcal{Z}^\phi(s,t)$ uniformly over s and t as noted above.
- The sequence $(a^L)_{L\in\mathbb{N}^*}$ is uniformly bounded, since $B^{\phi(L)}$ and $H^{\phi(L)}(1,\cdot)$ are. It also converges uniformly to $a: t\mapsto 2B^\phi(t)^\top(B^\phi(t)H^\phi(1,t)-y)$.
- The function g is $\mathcal{C}^1$ since f is $\mathcal{C}^2$. We clearly have $g(0,\cdot)\equiv 0$. Finally, $g(\cdot,(h,Z))$ is uniformly Lipschitz continuous for (h,Z) in any compact since $\partial_1 f$ is continuous.

Overall, we obtain that $P^{\phi(L)}(s,t)$ converges uniformly (over s and t) to $P^\phi(s,t)$, the solution at time s of the backward ODE

$$\begin{aligned} P^\phi(1,t) &= 2B^\phi(t)^\top(B^\phi(t)H^\phi(1,t)-y)\\ \frac{\partial P^\phi}{\partial s}(s,t) &= \partial_1 f(H^\phi(s,t),\mathcal{Z}^\phi(s,t))P^\phi(s,t),\quad s\in[0,1].\end{aligned}$$

Furthermore, the proof of the corollary shows that $(P^{\phi(L)})_{L\in\mathbb{N}^*}$ is uniformly bounded. Now, recall that the gradient flow for $Z_k^{\phi(L)}(t)$, given by (3.5) and (3.15), takes the following form, for $t\in I$ and $k\in\{1,\dots,\phi(L)\}$,

$$\frac{\partial Z_k^{\phi(L)}(t)}{\partial t} = \pi\Big(-\frac{1}{n}\sum_{i=1}^n \partial_2 f(h_{k-1,i}^{\phi(L)}(t), Z_k^{\phi(L)}(t))^\top p_{k-1,i}^{\phi(L)}(t)\Big),$$

where the i subscript corresponds to the i-th input x_i. By definition, for $s\in[0,1]$, $\mathcal{Z}^{\phi(L)}(s,t) = Z^{\phi(L)}_{\lfloor(\phi(L)-1)s\rfloor+1}(t)$. Thus, the equation above can be rewritten, for $s\in[0,1]$ and $t\in I$,

$$\frac{\partial \mathcal{Z}^{\phi(L)}(s,t)}{\partial t} = \pi\Big(-\frac{1}{n}\sum_{i=1}^n \partial_2 f(h^{\phi(L)}_{\lfloor(\phi(L)-1)s\rfloor,i}(t), Z^{\phi(L)}_{\lfloor(\phi(L)-1)s\rfloor+1}(t))^\top p^{\phi(L)}_{\lfloor(\phi(L)-1)s\rfloor,i}(t)\Big). \quad (3.37)$$

The term inside π can be rewritten as

$$-\frac{1}{n}\sum_{i=1}^{n}\partial_2 f\Big(H_i^{\phi(L)}\Big(\frac{\lfloor(\phi(L)-1)s\rfloor}{\phi(L)},t\Big),\mathcal{Z}^{\phi(L)}(s,t)\Big)^\top P_i^{\phi(L)}\Big(\frac{\lfloor(\phi(L)-1)s\rfloor}{\phi(L)},t\Big).$$

Since f is $\mathcal{C}^2$, $\partial_2 f$ is locally Lipschitz continuous. Applying the first part of the proof to the specific case of x_i, we know that $H_i^{\phi(L)}$ and $P_i^{\phi(L)}$ uniformly bounded, and that $H_i^{\phi(L)}(s,t)$ and $P_i^{\phi(L)}(s,t)$ converge uniformly to $H_i^{\phi}(s,t)$ and $P_i^{\phi}(s,t)$. Therefore, the right-hand side of (3.37) converges uniformly over s and t to

$$\pi\Big(-\frac{1}{n}\sum_{i=1}^{n}\partial_2 f(H_i^{\phi}(s,t),\mathcal{Z}^{\phi}(s,t))^\top P_i^{\phi}(s,t)\Big).$$

We have just shown the uniform convergence of the derivative in t of $\mathcal{Z}^{\phi(L)}(s,t)$. Furthermore, we know that, for $s\in[0,1]$, the sequence $(t\mapsto\mathcal{Z}^{\phi(L)}(s,t))_{L\in\mathbb{N}^*}$ converges to $\mathcal{Z}^{\phi}(s,\cdot)$. These two statements imply that $\mathcal{Z}^{\phi}$ is differentiable with respect to t and that, for $s\in[0,1]$, its derivative satisfies the ordinary differential equation

$$\frac{\partial\mathcal{Z}^{\phi}(s,t)}{\partial t}=\pi\Big(-\frac{1}{n}\sum_{i=1}^{n}\partial_2 f(H_i^{\phi}(s,t),\mathcal{Z}^{\phi}(s,t))^\top P_i^{\phi}(s,t)\Big). \tag{3.38}$$

Moreover, by our initialization scheme,

$$\mathcal{Z}^{\phi}(s,0)=Z^{\text{init}}(s). \tag{3.39}$$

A similar approach reveals that $A^{\phi}(t)$ and $B^{\phi}(t)$ are differentiable and that they verify the equations

$$\frac{dA^{\phi}}{dt}(t)=\pi\Big(-\frac{1}{n}\sum_{i=1}^{n}P_i^{\phi}(0,t)x_i^\top\Big),\qquad A^{\phi}(0)=A^{\text{init}}, \tag{3.40}$$

$$\frac{dB^{\phi}}{dt}(t)=\pi\Big(-\frac{2}{n}\sum_{i=1}^{n}(B^{\phi}(t)H_i^{\phi}(1,t)-y_i)H_i^{\phi}(1,t)^\top\Big),\qquad B^{\phi}(0)=B^{\text{init}}. \tag{3.41}$$

The equations (3.38) to (3.41) can be seen as an initial value problem whose variables are the function $\mathcal{Z}^{\phi}(\cdot,t):[0,1]\to\mathbb{R}^p$ and the matrices $A^{\phi}(t)\in\mathbb{R}^{q\times d},B^{\phi}(t)\in\mathbb{R}^{d'\times q}$. To complete the proof, it remains to show, using the Picard-Lindelöf theorem (see Lemma 3.19), that there exists a unique solution to this problem. First, note that the space $\mathcal{B}([0,1],\mathbb{R}^p)$ of bounded functions from $[0,1]$ to $\mathbb{R}^p$ endowed with the supremum norm is a Banach space, which is the proper space in which to apply the Picard-Lindelöf theorem. We therefore endow the space of parameters $\mathcal{B}([0,1],\mathbb{R}^p)\times\mathbb{R}^{q\times d}\times\mathbb{R}^{d'\times q}$ with the norm

$$\|(\mathcal{Z},A,B)\|:=\sup_{s\in[0,1]}\|\mathcal{Z}(s)\|+\|A\|_2+\|B\|_2,$$

which makes it a Banach space. We have to show that the mapping

$$\begin{aligned}(\mathcal{Z},A,B)\mapsto\Big(&s\mapsto\pi\Big(-\frac{1}{n}\sum_{i=1}^{n}\partial_2 f(H_i(s),\mathcal{Z}(s))^\top P_i(s)\Big),\\&\pi\Big(-\frac{1}{n}\sum_{i=1}^{n}P_i(0)x_i^\top\Big),\ \pi\Big(-\frac{2}{n}\sum_{i=1}^{n}(BH_i(1)-y_i)H_i(1)^\top\Big)\Big)\end{aligned} \tag{3.42}$$

is locally Lipschitz continuous with respect to this norm, where we recall that $H_i(s)$ in (3.42) is the solution at time s of the initial value problem

$$\begin{aligned} H_i(0) &= Ax_i \\ \frac{dH_i}{ds}(s) &= f(H_i(s), \mathcal{Z}(s)), \quad s \in [0,1], \end{aligned} \tag{3.43}$$

and $P_i(s)$ is the solution at time s of the initial value problem

$$\begin{aligned} P_i(1) &= 2B^\top(BH_i(1) - y_i) \\ \frac{dP_i}{ds}(s) &= \partial_1 f(H_i(s), \mathcal{Z}(s))P_i(s), \quad s \in [0,1]. \end{aligned} \tag{3.44}$$

To prove that the mapping (3.42) is locally Lipschitz continuous, we first check that it is well defined. Since $\mathcal{Z}$ is assumed to be only bounded (and not continuous), the solutions of the initial value problems (3.43) and (3.44) are well defined in the sense of the Caratheodory conditions, which are given in Lemma 3.20.

Next, we can show that $(\mathcal{Z}, A, B) \mapsto H_i$ is locally Lipschitz continuous for $i \in \{1, \dots, n\}$. To do this, consider two sets of parameters $(\mathcal{Z}, A, B)$ and $(\tilde{\mathcal{Z}}, \tilde{A}, \tilde{B})$ belonging to a compact set D. Let H_i and $\tilde{H}_i$ denote the corresponding hidden states. As in the proof of Proposition 3.12, it holds that H_i and $\tilde{H}_i$ belong to some compact set E that depends only on D and f. Let K_f be the Lipschitz constant of the $\mathcal{C}^1$ function f on $E \times D$. Then,

$$\begin{aligned} \|\tilde{H}_i(s) - H_i(s)\| &\leq \|\tilde{H}_i(0) - H_i(0)\| + \int_0^s \Big\| \frac{d\tilde{H}_i}{dr}(r) - \frac{dH_i}{dr}(r) \Big\| dr \\ &\leq \|\tilde{H}_i(0) - H_i(0)\| + \int_0^s \|f(\tilde{H}_i(r), \tilde{\mathcal{Z}}(r)) - f(H_i(r), \mathcal{Z}(r))\| dr. \end{aligned}$$

The norm inside the integral can be bounded by

$$\begin{aligned} &\|f(\tilde{H}_i(r), \tilde{\mathcal{Z}}(r)) - f(\tilde{H}_i(r), \mathcal{Z}(r))\| + \|f(\tilde{H}_i(r), \mathcal{Z}(r)) - f(H_i(r), \mathcal{Z}(r))\| \\ &\qquad \leq K_f \sup_{r \in [0,1]} \|\tilde{\mathcal{Z}}(r) - \mathcal{Z}(r)\| + K_f \|\tilde{H}_i(r) - H_i(r)\|. \end{aligned}$$

Therefore,

$$\|\tilde{H}_i(s) - H_i(s)\| \leq \|\tilde{A} - A\|_2 \|x_i\| + K_f \sup_{r \in [0,1]} \|\tilde{\mathcal{Z}}(r) - \mathcal{Z}(r)\| + \int_0^s K_f \|\tilde{H}_i(r) - H_i(r)\| dr.$$

Using Grönwall's inequality, we obtain, for any $s \in [0,1]$,

$$\|\tilde{H}_i(s) - H_i(s)\| \leq \Big(\|\tilde{A} - A\|_2 \|x_i\| + K_f \sup_{r \in [0,1]} \|\tilde{\mathcal{Z}}(r) - \mathcal{Z}(r)\| \Big) \exp(K_f).$$

This shows that the function $(\mathcal{Z}, A, B) \mapsto H_i$ is locally Lipschitz continuous. One proves by similar arguments that the function $(\mathcal{Z}, A, B) \mapsto P_i$ is locally Lipschitz continuous. Thus, overall, the mapping (3.42) is locally Lipschitz continuous as a composition of locally Lipschitz continuous functions.

The Picard-Lindelöf theorem guarantees the uniqueness of the maximal solution of the initial value problem (3.38)–(3.41) in the space $\mathcal{B}([0,1], \mathbb{R}^p) \times \mathbb{R}^{d \times q} \times \mathbb{R}^{d' \times q}$. Since any accumulation point $(\mathcal{Z}^\phi, A^\phi, B^\phi)$ is a solution belonging to this space, this proves the uniqueness of the accumulation point, which we therefore denote as $(\mathcal{Z}, A, B)$.

The uniform convergence of $(\mathcal{Z}^L, A^L, B^L)$ to $(\mathcal{Z}, A, B)$ is then easily shown by contradiction. Suppose that uniform convergence does not hold. If this is true, then there exists a subsequence that stays at distance $\varepsilon > 0$ from $(\mathcal{Z}, A, B)$ (in the sense of the uniform norm). Then arguments similar to the beginning of the proof show the existence of a second accumulation point, which is a contradiction. This shows the uniform convergence, yielding statements (i) and (ii) of the theorem.

Finally, reapplying Corollary 3.13 with $\theta_k^L = Z_k^L$, $\Theta = \mathcal{Z}$, $a^L = A^L x$, $g = f$, completes the proof by proving statements (iii) and (iv). □

Training dynamics of the limiting weights. Interestingly, the proof of Theorem 3.15 provides us with an explicit description of the evolution of the continuous-depth limiting weights during training. With the notation of the proof, the continuous weights satisfy the training dynamics:

$$\begin{aligned}
\frac{dA}{dt}(t) &= \pi\Big(-\frac{1}{n}\sum_{i=1}^n P_i(0,t) x_i^\top \Big) \\
\frac{\partial \mathcal{Z}}{\partial t}(s,t) &= \pi\Big(-\frac{1}{n}\sum_{i=1}^n \partial_2 f(H_i(s,t), \mathcal{Z}(s,t))^\top P_i(s,t) \Big) \\
\frac{dB}{dt}(t) &= \pi\Big(-\frac{2}{n}\sum_{i=1}^n (B(t)H_i(1,t) - y_i) H_i(1,t)^\top \Big),
\end{aligned}$$

where we recall that $H_i(s,t)$ is the solution at time s of the initial value problem

$$\begin{aligned}
H_i(0,t) &= A(t)x_i \\
\frac{\partial H_i}{\partial s}(s,t) &= f(H_i(s,t), \mathcal{Z}(s,t)), \quad s \in [0,1],
\end{aligned}$$

and $P_i(s,t)$ is the solution at time s of the problem

$$\begin{aligned}
P_i(1,t) &= 2B(t)^\top (B(t)H_i(1,t) - y_i) \\
\frac{\partial P_i}{\partial s}(s,t) &= \partial_1 f(H_i(s,t), \mathcal{Z}(s,t)) P_i(s,t), \quad s \in [0,1].
\end{aligned}$$

These equations can be thought of as the continuous-depth equivalent of the backpropagation equations.

3.A.6 Existence of the double limit when L, t tend to infinity

Proposition 3.16. *Consider the residual network* (3.12)*, and assume that:*

(i) $A^L(t)$*,* $Z^L_{\lfloor Ls \rfloor}(t)$*, and* $B^L(t)$ *converge uniformly over* $L \in \mathbb{N}^*$ *and* $s \in [0,1]$ *as* $t \to \infty$*.*

(ii) $A^L(t)$*,* $Z^L_{\lfloor Ls \rfloor}(t)$*, and* $B^L(t)$ *converge uniformly over* $t \in \mathbb{R}_+$ *and* $s \in [0,1]$ *as* $L \to \infty$*.*

(iii) The loss $\ell^L(t)$ *converges to* 0 *uniformly over* $L \in \mathbb{N}^*$ *as* $t \to \infty$*.*

Then the following four statements hold ***as*** t ***and*** L ***tend to infinity****:*

(i) There exist matrices $A_\infty \in \mathbb{R}^{q \times d}$ *and* $B_\infty \in \mathbb{R}^{d' \times q}$ *such that* $A^L(t)$ *and* $B^L(t)$ *converge to* A_∞ *and* B_∞*.*

(ii) There exists a Lipschitz continuous function $\mathcal{Z}_\infty : [0,1] \to \mathbb{R}^p$ such that $Z^L_{\lfloor Ls \rfloor}(t)$ converges to $\mathcal{Z}_\infty(t)$ uniformly over $s \in [0,1]$.

(iii) Uniformly over $s \in [0,1]$ and $x \in \mathcal{X}$, the hidden layer $h^L_{\lfloor Ls \rfloor}(t)$ converges to the solution at time s of the ODE

$$\begin{aligned} H(0) &= A_\infty x \\ \frac{dH}{ds}(s) &= f(H(s), \mathcal{Z}_\infty(s)), \quad s \in [0,1]. \end{aligned} \tag{3.45}$$

(iv) Uniformly over $x \in \mathcal{X}$, the output $F^L(x;t)$ converges to $F_\infty(x) = B_\infty H(1)$. Furthermore, $F_\infty(x_i) = y_i$ for $i \in \{1, \dots, n\}$.

Proof. The existence of limits A_∞ and B_∞ to $A^L(t)$ and $B^L(t)$ as L and t tend to infinity is given by Lemma 3.22. The same argument applies to $Z^L_{\lfloor sL \rfloor}(t)$, which provides a limit $\mathcal{Z}_\infty(s)$ to the sequence. Furthermore, following the proof of the lemma, we see that the convergence of $Z^L_{\lfloor sL \rfloor}(t)$ to $\mathcal{Z}_\infty(s)$ is uniform over $s \in [0,1]$. Corollary 3.14, applied with $\theta^L_k = Z^L_k$, $\Theta_\infty = \mathcal{Z}_\infty$, $a^L = A^L x$, $g = f$, then ensures that $h^L_{\lfloor Ls \rfloor}(t)$ converges uniformly (over $s \in [0,1]$ and $x \in \mathcal{X}$) to $H(s)$ that is the solution at time s of (3.45), as L and t tend to infinity. As a consequence, $F^L(x;t)$ converges uniformly over x to $F_\infty(x)$ as $L, t \to \infty$. Furthermore, recall that

$$\ell^L(t) = \frac{1}{n} \sum_{i=1}^n \|F^L(x_i;t) - y_i\|_2^2.$$

The left-hand side converges as $L, t \to \infty$ to 0 by assumption of the proposition, while the right-hand side converges to

$$\frac{1}{n} \sum_{i=1}^n \|F_\infty(x_i) - y_i\|_2^2.$$

Therefore, $F_\infty(x_i) = y_i$ for $i \in \{1, \dots, n\}$, and the proof is complete. □

3.B Proofs of the results of the main part of the chapter

Most of the results follow from those presented in Section 3.A. The only substantial proof is that of Proposition 3.6, which shows the local PL condition. It uses a result of Nguyen and Mondelli (2020) involving the Hermite transform and the sub-Gaussian variance proxy, which we define briefly. We refer to Debnath and Bhatta (2014, Chapter 17) and Vershynin (2018, Sections 2.5.2 and 3.4.1), respectively, for more detailed explanations.

Hermite transform. The r-th normalized probabilist's Hermite polynomial is given by

$$h_r(x) = \frac{1}{\sqrt{r!}} (-1)^r e^{x^2/2} \frac{d^r}{dx^r} e^{-x^2/2}, \quad r \geq 0.$$

This family of polynomials forms an orthonormal basis of square-integrable functions for the inner product

$$\langle f_1, f_2 \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f_1(x) f_2(x) e^{-x^2/2} dx.$$

Therefore, any function σ such that $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \sigma^2(x) e^{-x^2/2} dx < \infty$ can be decomposed on this basis. The r-th coefficient of this decomposition is denoted by $\eta_r(\sigma)$.

Sub-Gaussian random vector. A random vector $x \in \mathbb{R}^d$ is sub-Gaussian with variance proxy $v_x > 0$ if, for every $y \in \mathbb{R}^d$ of unit norm,

$$\mathbb{P}(|\langle x, y\rangle| \geq t) \leq 2\exp\Big(-\frac{t^2}{2v_x^2}\Big).$$

Additional notation. For a matrix A, we let $s_{\min}$ and $s_{\max}$ its minimum and maximum singular values, and similarly, $\lambda_{\min}$ and $\lambda_{\max}$ its minimum and maximum eigenvalues (whenever they exist).

Before delving into the proofs, we briefly describe the parts of this section that make use of the specific model (3.3). The most important one is the proof of Proposition 3.6, i.e., the proof that the residual network satisfies the (M, μ)-local PL condition. Additionally, in the proof of Proposition 3.3, the expressions for M and K are valid only for the specific model (3.3). Finally, in the proof of Theorem 3.7, the beginning of the proof reveals that condition (3.22) of Proposition 3.9 on μ can be expressed as a condition on the norm of the labels y_i. This applies only to the specific model (3.3). Observe that, if one assumes that the general residual network of Section 3.A satisfies the (M, μ)-local PL condition with μ given by (3.22), then the rest of the proof of Theorem 3.7 unfolds, and the conclusions of the theorem hold for the general model.

3.B.1 Proof of Proposition 3.1

Proposition 3.1 is a consequence of Proposition 3.8 with $f(h, (V, W)) = \frac{1}{\sqrt{m}} V\sigma(\frac{1}{\sqrt{q}} Wh)$.

3.B.2 Proof of Proposition 3.2

Proposition 3.12, with $\theta_k^L = (V_k^L, W_k^L)$, $\Theta = (\mathcal{V}, \mathcal{W})$, $a^L = Ax$, $g(h, (V, W)) = \frac{1}{\sqrt{m}} V\sigma(\frac{1}{\sqrt{q}} Wh)$, gives the existence and uniqueness of the solution of the neural ODE (3.6). Moreover, inspecting the proof of Proposition 3.12, equations (3.35) gives that, for any input $x \in \mathcal{X}$, the difference between the last hidden layer h_L^L of the discrete residual network (3.3) and its continuous counterpart $H(1)$ in the neural ODE (3.6) is bounded by

$$C'\Big(\frac{1}{L} + \sup_{s\in[0,1]} \|\Theta(s) - \Theta^L(s)\|\Big),$$

where $C' > 0$ is independent of L and $x \in \mathcal{X}$, and $\Theta^L(s) = \theta^L_{\lfloor (L-1)s \rfloor + 1}$. The function Θ^L is a piecewise-constant interpolation of Θ with pieces of length $\frac{1}{L-1}$. Since Θ is Lipschitz continuous, the distance between Θ and Θ^L decreases as C''/L for some $C'' > 0$ depending on Θ but not on L. This yields $\|h_L^L - H(1)\| \leq \frac{C'(1+C'')}{L}$, where C' and C'' are independent of L and $x \in \mathcal{X}$. Since $F^L(x) = Bh_L^L$ and $F(x) = BH(1)$, the result is proven.

3.B.3 Proof of Proposition 3.3

We apply Proposition 3.8 with $f(h, (V, W)) = \frac{1}{\sqrt{m}} V\sigma(\frac{1}{\sqrt{q}} Wh)$. Recall that the parameters $Z = (V, W)$ are considered in Proposition 3.8 as a vector. In particular, $\|Z\| = \|V\|_F + \|W\|_F$. Therefore, Proposition 3.8 shows that, for $t \in [0, T]$, $L \in \mathbb{N}^*$, and $k \in \{1, \dots, L\}$,

$$\|A^L(t)\|_F \leq M, \quad \|V_k^L(t)\|_F + \|W_k^L(t)\|_F \leq M, \quad \text{and} \quad \|B^L(t)\|_F \leq M,$$

where

$$
\begin{aligned}
M &= M_0 + TM_\pi \\
M_0 &= \max\Big(\|A^L(0)\|_F,\, \|V_0^L(0)\|_F + \|W_0^L(0)\|_F,\, \|B^L(0)\|_F\Big) \\
M_\pi &= \max\Big(\max_{A\in\mathbb{R}^{q\times d}} \|\pi(A)\|_F,\, \max_{Z\in\mathbb{R}^{q\times m}\times\mathbb{R}^{m\times q}} \|\pi(Z)\|,\, \max_{B\in\mathbb{R}^{d'\times q}} \|\pi(B)\|_F\Big).
\end{aligned}
$$

Furthermore, due to our initialization scheme described in Section 3.3,

$$
\|A^L(0)\|_F = \sqrt{d}, \quad \|V_0^L(0)\|_F = 0, \quad \|W_0^L(0)\|_F \leq 2\sqrt{qm}, \quad \|B^L(0)\|_F = \sqrt{d'},
$$

where the third inequality holds with probability at least $1-\exp(-\frac{3qm}{16})$ by Lemma 3.23. Since we take $q \geq \max(d, d')$, this implies that, with high probability, $M_0 \leq 2\sqrt{qm}$, yielding the formula for M in Proposition 3.3. Finally, the existence of $K = \beta T e^{\alpha T}$ such that the difference between two successive weight matrices is bounded by K/L, as well as the dependence of α and β on $\mathcal{X}$, $\mathcal{Y}$, M, and σ, follows easily from Proposition 3.8, given that our initialization scheme ensures that $Z_k^L(0) = Z_{k+1}^L(0)$ for all $L \in \mathbb{N}^*$ and $k \in \{1, \dots, L\}$.

3.B.4 Proof of Theorem 3.4

By Proposition 3.3 and the fact that π is bounded, the sequences $(A^L)_{L\in\mathbb{N}^*}$ and $(B^L)_{L\in\mathbb{N}^*}$ each satisfy the assumptions of Corollary 3.11, and $(Z_k^L)_{L\in\mathbb{N}^*, 1\leq k\leq L}$ satisfies the assumptions of Proposition 3.10. Theorem 3.4 then follows directly from Theorem 3.15, by taking, as previously, $f(h,(V,W)) = \frac{1}{\sqrt{m}} V\sigma(\frac{1}{\sqrt{q}} Wh)$.

3.B.5 Proof of Proposition 3.6

We drop the L superscripts for this proof, since L is fixed. Denote by $\bar{A}, \bar{B}, \bar{V}_k, \bar{W}_k$ parameters sampled according to the initialization scheme of Section 3.3, which means in particular that $\bar{V}_k = 0$ and $\bar{W}_k = \bar{W} \sim \mathcal{N}^{\otimes(m\times q)}$. Since, by assumption, the activation function σ is bounded and not constant, it cannot be a polynomial function. As a consequence, there are infinitely many non-zero coefficients $\eta_r(\sigma)$ in its Hermite expansion (defined at the beginning of Section 3.B). Throughout, we let $r \geq 2$ be an integer such that $\eta_r(\sigma)$ is nonzero. We also let K_σ be the Lipschitz constant of σ and M_σ its supremum norm. Now, let A, B, V_k, W_k be parameters at distance at most $M = \min(\frac{\eta_r(\sigma)}{32K_\sigma\sqrt{2nq}}, \frac{1}{2})$ from $\bar{A}, \bar{B}, \bar{V}_k, \bar{W}_k$ in the sense of Definition 3.5.

It is useful for this proof to introduce a matrix-valued version of the residual network (3.3). More specifically, given data matrices $\mathbf{x} \in \mathbb{R}^{d\times n}$ and $\mathbf{y} \in \mathbb{R}^{d'\times n}$, the matrix-valued residual network writes

$$
\begin{aligned}
\mathbf{h}_0 &= A\mathbf{x} \\
\mathbf{h}_{k+1} &= \mathbf{h}_k + \frac{1}{L\sqrt{m}} V_{k+1}\sigma\Big(\frac{1}{\sqrt{q}} W_{k+1}\mathbf{h}_k\Big), \quad k \in \{0, \dots, L-1\},
\end{aligned} \tag{3.46}
$$

where now $\mathbf{h}_k \in \mathbb{R}^{q\times n}$. The loss is equal to $\ell = \frac{1}{n}\|B\mathbf{h}_L - \mathbf{y}\|_F^2$ and we let $\mathbf{p}_k = \frac{\partial \ell}{\partial \mathbf{h}_k} \in \mathbb{R}^{q\times n}$ be the matrix-valued backward state. Observe that the columns of $\mathbf{x}$ are bounded and thus sub-Gaussian. In the sequel, we denote by v_x the sub-Gaussian variance proxy of the columns of $\sqrt{d/q}\mathbf{x}$.

Now that we have introduced the necessary notation, we can proceed to prove some preliminary estimates. Since $M \leq \frac{1}{2} \leq \sqrt{2qm}$, we have, for $k \in \{1, \dots, n\}$,

$$
\|A - \bar{A}\|_F \leq M, \quad \|B - \bar{B}\|_F \leq \frac{1}{2}, \quad \|V_k\|_F \leq 1, \quad \|W_k - \bar{W}\|_F \leq \frac{1}{2} \leq \sqrt{2qm}. \tag{3.47}
$$

By Lemma 3.23, with probability at least $1 - \exp\big(-\frac{qm}{16}\big)$, one has $\|\bar{W}\|_F \le \sqrt{2qm}$. Together with the previous inequalities, this implies

$$\|A\|_2 \le 2, \quad s_{\min}(B) \ge \frac{1}{2}, \quad \|B\|_2 \le \frac{3}{2}, \quad \|V_k\|_F \le 1, \quad \|W_k\|_F \le 2\sqrt{2qm}, \tag{3.48}$$

where the second inequality is a consequence of Lemma 3.21, as follows:

$$s_{\min}(B) \ge s_{\min}(\bar{B}) - \|B - \bar{B}\|_F = 1 - \|B - \bar{B}\|_F \ge \frac{1}{2}.$$

Let us now bound $\|\mathbf{h}_k\|_F$ and $\|\mathbf{p}_k\|_F$. We have

$$\|\mathbf{h}_0\|_F = \|A\mathbf{x}\|_F \le \|A\|_2 \|\mathbf{x}\|_F \le 2\sqrt{qn}. \tag{3.49}$$

Moreover, by (3.46), for any $k \in \{0, \dots, L-1\}$,

$$\|\mathbf{h}_{k+1}\|_F \le \|\mathbf{h}_k\|_F + \frac{K_\sigma}{L\sqrt{m}\sqrt{q}} \|V_{k+1}\|_F \|W_{k+1}\|_F \|\mathbf{h}_k\|_F \le \Big(1 + \frac{2\sqrt{2}K_\sigma}{L}\Big)\|\mathbf{h}_k\|_F,$$

where the second inequality is a consequence of (3.48). Therefore, by (3.49),

$$\|\mathbf{h}_k\|_F \le \exp(2\sqrt{2}K_\sigma)\|\mathbf{h}_0\|_F \le 2\exp(2\sqrt{2}K_\sigma)\sqrt{qn}. \tag{3.50}$$

Moving on to $\|\mathbf{p}_k\|_F$, the chain rule leads to

$$\mathbf{p}_k = \mathbf{p}_{k+1} + \frac{1}{L\sqrt{qm}} W_{k+1}^\top \Big((V_{k+1}^\top \mathbf{p}_{k+1}) \odot \sigma'\big(\frac{1}{\sqrt{q}} W_{k+1}\mathbf{h}_k\big)\Big), \quad k \in \{0, \dots, L-1\},$$

where $\odot$ denotes the element-wise product. Noting that $|\sigma'| \le K_\sigma$ and using (3.48), we obtain

$$\|\mathbf{p}_k\|_F \ge \|\mathbf{p}_{k+1}\|_F - \frac{K_\sigma}{L\sqrt{qm}} \|W_{k+1}\|_F \|V_{k+1}\|_F \|\mathbf{p}_{k+1}\|_F \ge \Big(1 - \frac{2\sqrt{2}K_\sigma}{L}\Big)\|\mathbf{p}_{k+1}\|_F.$$

It follows that $\|\mathbf{p}_k\|_F \ge \exp(-2\sqrt{2}K_\sigma)\|\mathbf{p}_L\|_F$. In addition,

$$\mathbf{p}_L = \frac{\partial \ell}{\partial \mathbf{h}_L} = \frac{2}{n} B^\top (B\mathbf{h}_L - \mathbf{y}).$$

Therefore, by Lemma 3.21, since $d' \le q$,

$$\|\mathbf{p}_L\|_F \ge \frac{2}{n} s_{\min}(B) \|B\mathbf{h}_L - \mathbf{y}\|_F \ge \frac{1}{\sqrt{n}} \sqrt{\ell}.$$

Collecting bounds, we conclude that, for $k \in \{0, \dots, L\}$,

$$\|\mathbf{p}_k\|_F \ge \frac{1}{\sqrt{n}} \exp(-2\sqrt{2}K_\sigma)\sqrt{\ell}. \tag{3.51}$$

A similar proof reveals that, for $k \in \{0, \dots, L\}$,

$$\|\mathbf{p}_k\|_F \le \frac{3}{\sqrt{n}} \exp(2\sqrt{2}K_\sigma)\sqrt{\ell}.$$

Having established these preliminary estimates, our goal in the remainder of the proof is to lower bound the quantity $\|\frac{\partial \ell}{\partial V_{k+1}}\|_F$. First note that, by the chain rule, for any $k \in \{0, \dots, L-1\}$,

$$\frac{\partial \ell}{\partial V_{k+1}} = \frac{1}{L\sqrt{m}} \mathbf{p}_{k+1} \sigma\Big(\frac{1}{\sqrt{q}} W_{k+1}\mathbf{h}_k\Big)^\top.$$

As a consequence, when $m \geq n$, by Lemma 3.21,

$$\begin{aligned}\Big\|\frac{\partial \ell}{\partial V_{k+1}}\Big\|_F &\geq \frac{1}{L\sqrt{m}}\|\mathbf{p}_{k+1}\|_F \cdot s_{\min}\Big(\sigma\Big(\frac{1}{\sqrt{q}}W_{k+1}\mathbf{h}_k\Big)\Big) \\ &\geq \frac{1}{L\sqrt{mn}}\exp(-2\sqrt{2}K_\sigma)\sqrt{\ell}\cdot s_{\min}\Big(\sigma\Big(\frac{1}{\sqrt{q}}W_{k+1}\mathbf{h}_k\Big)\Big), \end{aligned} \tag{3.52}$$

using (3.51). Next, by Lemma 3.21,

$$s_{\min}\Big(\sigma\Big(\frac{1}{\sqrt{q}}W_{k+1}\mathbf{h}_k\Big)\Big) \geq s_{\min}\Big(\sigma\Big(\frac{1}{\sqrt{q}}\bar{W}\bar{A}\mathbf{x}\Big)\Big) - \Big\|\sigma\Big(\frac{1}{\sqrt{q}}W_{k+1}\mathbf{h}_k\Big) - \sigma\Big(\frac{1}{\sqrt{q}}\bar{W}\bar{A}\mathbf{x}\Big)\Big\|_F.$$

Let us first lower bound the first term. Since, by our choice of initialization, $\bar{A} = (I_{\mathbb{R}^{d\times d}}, 0_{\mathbb{R}^{(q-d)\times d}})$, we have

$$s_{\min}\Big(\sigma\Big(\frac{1}{\sqrt{q}}\bar{W}\bar{A}\mathbf{x}\Big)\Big) = s_{\min}(\sigma(\tilde{W}\tilde{\mathbf{x}})),$$

where $\tilde{W} \sim \mathcal{N}(0,1)^{\otimes(m\times d)}$ and $\tilde{\mathbf{x}} = \frac{1}{\sqrt{q}}\mathbf{x} \in \mathbb{R}^{d\times n}$ has i.i.d. unitary columns independent of $\tilde{W}$. Therefore, by Lemma 3.24, with probability at least $1 - \exp\big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\big) - 2n^2\exp\big(-\frac{d}{2v_x n^{2/r}}\big)$,

$$s_{\min}\Big(\sigma\Big(\frac{1}{\sqrt{q}}W\bar{A}\mathbf{x}\Big)\Big) \geq \frac{\sqrt{m}\eta_r(\sigma)}{4}.$$

Next,

$$\begin{aligned}\Big\|\sigma\Big(\frac{1}{\sqrt{q}}W_{k+1}\mathbf{h}_k\Big) - \sigma\Big(\frac{1}{\sqrt{q}}\bar{W}\bar{A}\mathbf{x}\Big)\Big\|_F \leq \frac{K_\sigma}{\sqrt{q}}\Big(&\|W_{k+1}-\bar{W}\|_F\|\mathbf{h}_k\|_F + \|\bar{W}\|_F\|\mathbf{h}_k - A\mathbf{x}\|_F \\ &+ \|\bar{W}\|_F\|A\mathbf{x}-\bar{A}\mathbf{x}\|_F\Big).\end{aligned}$$

Clearly,

$$\|\mathbf{h}_k - A\mathbf{x}\|_F = \Big\|\sum_{j=1}^{k}\frac{1}{L\sqrt{m}}V_j\sigma\Big(\frac{1}{\sqrt{q}}W_j\mathbf{h}_{j-1}\Big)\Big\|_F \leq \frac{4\sqrt{2}K_\sigma k}{L}\exp(2\sqrt{2}K_\sigma)\sqrt{qn},$$

by (3.48) and (3.50). Also,

$$\|A\mathbf{x} - \bar{A}\mathbf{x}\|_F \leq \|A-\bar{A}\|_F\|\mathbf{x}\|_F \leq \frac{\eta_r(\sigma)}{32\sqrt{2}K_\sigma},$$

by (3.47) and by definition of M. Putting together the two bounds above as well as (3.47), (3.48), and (3.50), we obtain

$$\begin{aligned}\Big\|\sigma\Big(\frac{1}{\sqrt{q}}W_{k+1}\mathbf{h}_k\Big) - \sigma\Big(\frac{1}{\sqrt{q}}W\bar{A}\mathbf{x}\Big)\Big\|_F &\leq K_\sigma\exp(2\sqrt{2}K_\sigma)\sqrt{n}\Big(1+\sqrt{qm}\frac{8K_\sigma k}{L}\Big) + \sqrt{m}\frac{\eta_r(\sigma)}{32} \\ &\leq C_1\sqrt{n} + C_2\frac{\sqrt{nqm}k}{16L} + \sqrt{m}\frac{\eta_r(\sigma)}{32},\end{aligned}$$

where $C_1 = K_\sigma\exp(2\sqrt{2}K_\sigma)$ and $C_2 = 128C_1K_\sigma$. Thus, when $C_1\sqrt{n} \leq \frac{1}{32}\sqrt{m}\eta_r(\sigma)$, we have

$$s_{\min}\Big(\sigma\Big(\frac{1}{\sqrt{q}}W_{k+1}\mathbf{h}_k\Big)\Big) \geq \sqrt{m}\Big(\frac{3}{16}\eta_r(\sigma) - \frac{C_2}{16}\sqrt{nq}\frac{k}{L}\Big) \geq \frac{1}{8}\sqrt{m}\eta_r(\sigma)$$

for $k \leq \frac{L\eta_r(\sigma)}{C_2\sqrt{nq}}$. As a consequence, for $k \leq \frac{L\eta_r(\sigma)}{C_2\sqrt{nq}}$, returning to (3.52),

$$\Big\|\frac{\partial \ell}{\partial V_{k+1}}\Big\|_F \geq \frac{1}{8L\sqrt{n}}\eta_r(\sigma)\exp(-2\sqrt{2}K_\sigma)\sqrt{\ell} = \frac{C_3\eta_r(\sigma)}{L\sqrt{n}}\sqrt{\ell},$$

letting $C_3 = \frac{\exp(-2\sqrt{2}K_\sigma)}{8}$. Therefore,

$$\begin{aligned}\Big\|\frac{\partial \ell}{\partial A}\Big\|_F^2 + L\sum_{k=1}^{L}\Big\|\frac{\partial \ell}{\partial Z_{k+1}}\Big\|_F^2 + \Big\|\frac{\partial \ell}{\partial B}\Big\|_F^2 &\geq L\sum_{k=1}^{\lfloor \frac{L\eta_r(\sigma)}{C_2\sqrt{nq}}\rfloor}\Big\|\frac{\partial \ell}{\partial V_{k+1}}\Big\|_F^2 \\ &\geq L\Big\lfloor\frac{L\eta_r(\sigma)}{C_2\sqrt{nq}}\Big\rfloor\frac{C_3^2\eta_r(\sigma)^2}{L^2 n}\ell \\ &\geq \frac{C_3^2\eta_r(\sigma)^3}{2C_2 n\sqrt{nq}}\ell,\end{aligned}$$

where we used the inequality $\lfloor x \rfloor \geq x/2$ for $x \geq 1$. This proves the result, with

$$\begin{aligned}c_1 &= \max\Big(\frac{2^{10}C_1^2}{\eta_r(\sigma)^2}, 1\Big) = \max\Big(\frac{2^{10}K_\sigma^2\exp(4\sqrt{2}K_\sigma)}{\eta_r(\sigma)^2}, 1\Big) \\ c_2 &= \frac{C_2}{\eta_r(\sigma)} = \frac{128K_\sigma^2\exp(2\sqrt{2}K_\sigma)}{\eta_r(\sigma)} \\ c_3 &= \min\Big(\frac{\eta_r(\sigma)}{32\sqrt{2}K_\sigma}, \frac{1}{2}\Big) \\ c_4 &= \frac{C_3^2\eta_r(\sigma)^3}{2C_2} = \frac{\eta_r(\sigma)^3}{2^{14}K_\sigma^2\exp(6\sqrt{2}K_\sigma)} \\ \delta &= \exp\big(-\frac{qm}{16}\big) + n\exp\Big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\Big) + 2n^2\exp\Big(-\frac{d}{2v_x n^{2/r}}\Big).\end{aligned}$$

Remark 3.17. *With appropriate values of r and m, the probability of failure δ can be made as small as*

$$\varepsilon + 2n^2\exp\Big(-\frac{d}{2v_x n^{\varepsilon}}\Big), \tag{3.53}$$

for any $\varepsilon > 0$. This is possible first by choosing r such that $2/r \geq \varepsilon$, then by choosing m such that the first two terms are less than ε. Moreover, we refer the interested reader to Goel et al. (2020, Lemmas A.2 and A.9) for quantitative estimates of $\eta_r(\sigma)$ for ReLU and sigmoid activations. Finally, the expression (3.53) *is essentially the same as the one appearing in Nguyen and Mondelli (2020, Theorem 3.3). As in this chapter, we note that this expression is small if n grows at most polynomially with d, in which case the exponential term in d dominates the polynomial term in n.*

3.B.6 Proof of Theorem 3.7

By Proposition 3.6, there exists $\delta > 0$ such that, with probability at least $1 - \delta$, the residual network (3.3) satisfies the (M, μ)-local PL condition around its initialization, with

$$M = \frac{c_3}{\sqrt{nq}} \quad \text{and} \quad \mu = \frac{c_4}{n\sqrt{nq}},$$

for c_3 and c_4 depending on σ. We now apply Proposition 3.9 with $f(h, (V, W)) = \frac{1}{\sqrt{m}}V\sigma(\frac{1}{\sqrt{q}}Wh)$. The only assumption of Proposition 3.9 that requires some care to check is that the PL condition

holds for the value of μ given by equation (3.22). Since the (M, μ)-local PL condition implies the $(M, \tilde{\mu})$-local PL condition for any $\tilde{\mu} \in (0, \mu)$, it is the case if

$$\frac{c_4}{n\sqrt{n}q} \geq \max(M_B K, M_B M_X, M_A M_X) \frac{8e^K}{M} \sup_{L \in \mathbb{N}^*} \sqrt{\ell^L(0)},$$

with M_X, M_A, M_B, and K defined in Proposition 3.9. Due to the initialization scheme of Section 3.3, we have, for any input $x \in \mathcal{X}$, $h_L^L(0) = h_0^L(0)$, hence $F^L(x) = B^L(0)A^L(0)x = 0$ since $q \geq d + d'$. As a consequence, $\ell^L(0) = \frac{1}{n}\sum_{i=1}^n \|y_i\|^2$. Therefore, the condition becomes

$$\frac{1}{n}\sum_{i=1}^n \|y_i\|^2 \leq \frac{c_3^2 c_4^2}{64 n^4 q^3 \max(M_B K, M_B M_X, M_A M_X)^2 e^{2K}},$$

where we replaced M by its value. Define C to be equal to the constant on the right-hand side. Then, according to the above, as soon as $\frac{1}{n}\sum_{i=1}^n \|y_i\|^2 \leq C$, we can apply Proposition 3.9, which gives several guarantees. First, the gradient flow is well defined on $\mathbb{R}_+$. Moreover, the proposition and the expression of μ given above yield the bound on the empirical risk. In particular, the empirical risk converges uniformly to zero. Furthermore, Proposition 3.9 shows the uniform convergence of the weights as $t \to \infty$. Finally, the proposition ensures that the sequences $(A^L)_{L \in \mathbb{N}^*}$ and $(B^L)_{L \in \mathbb{N}^*}$ each satisfy the assumptions of Corollary 3.11, and that $(Z_k^L)_{L \in \mathbb{N}^*, 1 \leq k \leq L}$ satisfies the assumptions of Proposition 3.10. We can therefore apply Theorem 3.15, with f defined above and π equal to the identity. This gives the uniform convergence of the weights as $L \to \infty$. The four asymptotic statements of Theorem 3.7 are then a consequence of Proposition 3.16.

Remark 3.18. *A close examination of the quantities involved in the definition of C reveals that it depends only on $\mathcal{X}$, σ, n, and q. In particular, it does not depend on the dimension m.*

3.C Some technical lemmas

We start by recalling the Picard-Lindelöf theorem (see, e.g., Luk, 2017, for a self-contained presentation, and Arnold, 1992, for a textbook).

Lemma 3.19 (Picard-Lindelöf theorem)**.** *Let $I = [0, T] \subset \mathbb{R}_+$ be an interval, for some $T \in (0, \infty]$. Consider the initial value problem*

$$U(s) = U_0 + \int_0^s g(U(r), r)dr, \quad s \in I, \tag{3.54}$$

where $g : \mathbb{R}^d \times I \to \mathbb{R}^d$ is continuous and locally Lipschitz continuous in its first variable. Then the initial value problem is well defined on an interval $[0, T_{\max}) \subset I$, i.e., there exists a unique maximal solution on this interval. Moreover, if $T_{\max} < T$, then $\|U(s)\|$ tends to infinity when s tends to $T_{\max}$. Finally, if $g(\cdot, r)$ is uniformly Lipschitz continuous for r in any compact, then $T_{\max} = T$.

We define time-dependent dynamics (3.54) for generality, but the time-independent case $U(s) = U_0 + \int_0^s g(U(r))dr$ is also of interest. In this case, the existence and uniqueness of the maximal solution holds if g is locally Lipschitz continuous, and the solution is defined on I if g is Lipschitz continuous. Besides, the first statement of Lemma 3.19 (existence and uniqueness of the maximal solution) also holds if $\mathbb{R}^d$ is replaced by any (potentially infinite-dimensional) Banach space.

The next lemma gives conditions for the existence and uniqueness of the global solution of the initial value problem (3.54) when the assumption of continuity of g in its second variable is removed, thereby generalizing the Picard-Lindelöf theorem.

Lemma 3.20 (Caratheodory conditions for the existence and uniqueness of the global solution of an initial value problem)**.** *Consider the initial value problem*

$$U(s) = U_0 + \int_0^s g(U(r), r)dr, \quad s \in [0, 1],$$

where $g : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is measurable and the integral is understood in the sense of Lebesgue integration. Assume that $g(\cdot, r)$ is uniformly Lipschitz continuous for almost all $r \in [0, 1]$, and that $g(0, r) \equiv 0$. Then there exists a unique solution to the initial value problem, defined on $[0, 1]$.

Proof. The proof is a consequence of Filippov (1988, Theorems 1, 2, and 4). More specifically, denote by $C > 0$ the uniform Lipschitz constant of $g(\cdot, r)$. According to Filippov (1988, Theorems 1 and 2), under the conditions of the lemma, there exists a unique maximal solution to the initial value problem. Let us now consider a restricted version of the problem, where g is defined on $D \times [0, 1]$, with D a compact of $\mathbb{R}^d$ large enough to contain in its interior the ball of center 0 and radius $\|U_0\| \exp(C)$. There exists a unique maximal solution to this problem as well, also according to Filippov (1988, Theorems 1 and 2), and, according to Filippov (1988, Theorem 4), it is defined until it reaches the boundary of $D \times [0, 1]$, which it reaches at some point (U^*, s^*). If $s^* < 1$, it means that U^* is on the boundary of D, and in particular that $\|U^*\| > \|U_0\| \exp(C)$. But, on the other hand, for almost every $r \in [0, 1]$,

$$\|g(U(r), r)\| \leq \|g(0, r)\| + \|g(U(r), r) - g(0, r)\| \leq C\|U(r)\|.$$

Hence, by Grönwall's inequality, for $s \leq s^*$,

$$\|U(s)\| \leq \|U_0\| \exp(C).$$

Thus, $\|U^*\| \leq \|U_0\| \exp(C)$, which is impossible. Hence the maximal solution of the restricted problem is defined on $[0, 1]$. Furthermore, the maximal solution of the original problem coincides with the restricted one whenever $U(s) \in D$, which is the case for every $s \in [0, 1]$, hence the maximal solution is defined on $[0, 1]$. □

The next three lemmas recall well-known results from linear algebra, analysis, and random matrix theory. Recall that $s_{\min}$ and $\lambda_{\min}$ denote respectively the minimum singular value and eigenvalue of a matrix.

Lemma 3.21. *Let $A, A' \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$. Then*

$$s_{\min}(A + A') \geq s_{\min}(A) - \|A'\|_F.$$

If $m \geq r$, then $\|AB\|_F \geq s_{\min}(A)\|B\|_F$. Furthermore, if $n \geq r$, then $\|AB\|_F \geq \|A\|_F s_{\min}(B)$.

Proof. The first statement is a consequence of, e.g., Loyka (2015), which establishes that $s_{\min}(A + A') \geq s_{\min}(A) - s_{\max}(A')$, yielding the first inequality since $s_{\max}(A') = \|A\|_2 \leq \|A\|_F$. As for the second one, we have

$$\|AB\|_F^2 = \mathrm{Tr}(ABB^\top A^\top) = \mathrm{Tr}(BB^\top A^\top A) \geq \lambda_{\min}(A^\top A)\mathrm{Tr}(BB^\top) = \lambda_{\min}(A^\top A)\|B\|_F^2.$$

Since $m \geq r$, the rightmost quantity is equal to $s_{\min}(A)\|B\|_F$, proving the second statement of the lemma. The third statement is similar. □

Lemma 3.22. *Let $(e_{x,y})_{x \in \mathbb{R}_+, y \in \mathbb{R}_+} \subset E$, where E is a Banach space, such that $e_{x,y}$ converges uniformly to $e_{\infty,y}$ when $x \to \infty$, and converges uniformly to $e_{x,\infty}$ when $y \to \infty$. Then there exists $e_\infty \in E$ such that*

$$\lim_{x,y \to \infty} e_{x,y} = \lim_{x \to \infty} e_{x,\infty} = \lim_{y \to \infty} e_{\infty,y} = e_\infty.$$

Proof. Let $\varepsilon > 0$. Since $e_{x,y}$ converges uniformly to $e_{\infty,y}$ as $x \to \infty$, there exists $x_0 \in \mathbb{R}_+$ such that, for $x_1, x_2 > x_0$ and $y \in \mathbb{R}_+$,

$$\|e_{x_1,y} - e_{x_2,y}\| \leq \frac{\varepsilon}{2}.$$

Similarly, there exists $y_0 \in \mathbb{R}_+$ such that, for $x \in \mathbb{R}_+$ and $y_1, y_2 > y_0$,

$$\|e_{x,y_1} - e_{x,y_2}\| \leq \frac{\varepsilon}{2}.$$

Hence, for $x_1, x_2 > x_0$ and $y_1, y_2 > y_0$,

$$\|e_{x_1,y_1} - e_{x_2,y_2}\| \leq \|e_{x_1,y_1} - e_{x_1,y_2}\| + \|e_{x_1,y_2} - e_{x_2,y_2}\| \leq \varepsilon.$$

We conclude that $(e_{x,y})_{x\in\mathbb{R}_+,y\in\mathbb{R}_+}$ is a Cauchy sequence, which therefore converges to some limit $e_\infty \in E$. □

Lemma 3.23. *Let $W \in \mathbb{R}^{q\times m}$ be a standard Gaussian random matrix. Then, for $M_W \geqslant \sqrt{2}$, with probability at least $1 - \exp(-\frac{(M_W^2-1)qm}{16})$, one has $\|W\|_F \leq M_W\sqrt{q}\sqrt{m}$.*

Proof. The quantity $\|W\|_F^2$ follows a chi-squared distribution with qm degrees of freedom. Hence, according to Laurent and Massart (2000, Lemma 1), for $x \geq 0$,

$$\mathbb{P}(\|W\|_F^2 - qm \geq 2\sqrt{qmx} + 2x) \leq \exp(-x).$$

Taking $x = \frac{(M_W^2-1)qm}{16}$, we see that

$$2\sqrt{qmx} = \frac{1}{2}\sqrt{M_W^2 - 1}qm \leq \frac{1}{2}(M_W^2 - 1)qm,$$

where the bound follows from $M_W \geqslant \sqrt{2}$. Since furthermore $2x \leq \frac{1}{2}(M_W^2 - 1)qm$, we obtain

$$2\sqrt{qmx} + 2x \leq (M_W^2 - 1)qm,$$

and thus

$$\mathbb{P}(\|W\|_F^2 > M_W^2 qm) \leq \mathbb{P}(\|W\|_F^2 - qm \geq 2\sqrt{qmx} + 2x) \leq \exp(-x),$$

yielding the result. □

Finally, the last lemma of the section gives a lower bound on the smallest singular value of a matrix of the form $\sigma(A)$, where σ is a bounded function applied element-wise and A belongs to a family of random matrix. The lower bound involves the Hermite transform of σ, which is defined in Section 3.B.

Lemma 3.24. *Let σ be a function bounded by some $M_\sigma > 0$. Let $W \in \mathbb{R}^{m\times d}$ be a standard Gaussian random matrix, and $X \in \mathbb{R}^{d\times n}$ a random matrix with i.i.d. unitary columns independent of W. Then, for any integer $r \geq 2$, there exists $\delta > 0$ such that, with probability at least $1 - \delta$, the smallest singular value of $\sigma(WX)$ is greater than $\frac{1}{4}\sqrt{m}\eta_r(\sigma)$, where $\eta_r(\sigma)$ is the r-th coefficient in the Hermite transform of σ. Furthermore, the following expression for δ holds:*

$$\delta = n\exp\Big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\Big) + 2n^2\exp\Big(-\frac{d}{2Cn^{2/r}}\Big),$$

where C is the sub-Gaussian variance proxy of the columns of $\sqrt{d}X$.

Proof. Denoting by w_i the i-th row of W and letting

$$M_i = \sigma(X^\top w_i^\top)\sigma(w_i X),$$

our goal is to lower bound the smallest eigenvalue value $\lambda_{\min}(M)$ of $M = \sum_{i=1}^m M_i$. Observe that

$$\begin{aligned}\mathbb{E}(M|X) &= m\mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,I_d)}\Big(\sigma(X^\top\tilde{w}^\top)\sigma(\tilde{w}X)\Big|X\Big)\\ &= m\mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\frac{1}{d}I_d)}\Big(\sigma\big((\sqrt{d}X)^\top\tilde{w}^\top\big)\sigma\big(\tilde{w}(\sqrt{d}X)\big)\Big|X\Big).\end{aligned}$$

Letting $\lambda_{\min}(\mathbb{E}(M|X))$ be the smallest eigenvalue of this matrix and $r \geq 2$ be an integer, Nguyen and Mondelli (2020, Lemma 3.4) show that, with probability at least $1 - 2n^2\exp(-\frac{d}{2Cn^{2/r}})$ over the matrix X,

$$\lambda_{\min}(\mathbb{E}(M|X)) \geq \frac{m\eta_r^2(\sigma)}{8}. \tag{3.55}$$

We now apply a matrix Chernoff's bound to lower bound with high probability the smallest eigenvalue $\lambda_{\min}(M|X)$ of M conditionally on X, as a function of $\lambda_{\min}(\mathbb{E}(M|X))$. By Tropp (2012, Remark 5.3), we have, for $t \in [0,1]$,

$$\mathbb{P}(\lambda_{\min}(M) \leq t\lambda_{\min}(\mathbb{E}(M|X))|X) \leq n\exp\Big(-\frac{(1-t^2)\lambda_{\min}(\mathbb{E}(M|X))}{2R(X)}\Big),$$

where $R(X)$ is an almost sure upper bound on the largest eigenvalue of $M_i|X$, which we can take equal to $M_\sigma^2 n$ since the largest eigenvalue of M_i is equal to $\|\sigma(w_i X)\|_2^2 \leq M_\sigma^2 n$. Taking $t = 1/2$, we obtain, on the event $[\lambda_{\min}(\mathbb{E}(M|X)) \geq \frac{m\eta_r^2(\sigma)}{8}]$,

$$\mathbb{P}\Big(\lambda_{\min}(M) \geq \frac{\lambda_{\min}(\mathbb{E}(M|X))}{2}\Big|X\Big) \geq 1 - n\exp\Big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\Big),$$

thus, on the event $[\lambda_{\min}(\mathbb{E}(M|X)) \geq \frac{m\eta_r^2(\sigma)}{8}]$,

$$\mathbb{P}\Big(\lambda_{\min}(M) \geq \frac{m\eta_r^2(\sigma)}{16}\Big) \geq 1 - n\exp\Big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\Big).$$

Using (3.55), we obtain

$$\begin{aligned}\mathbb{P}\Big(\lambda_{\min}(M) \geq \frac{m\eta_r^2(\sigma)}{16}\Big) &\geq \Big(1 - n\exp\Big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\Big)\Big)\mathbb{P}\Big(\lambda_{\min}(\mathbb{E}(M|X)) \geq \frac{m\eta_r^2(\sigma)}{8}\Big)\\ &\geq \Big(1 - n\exp\Big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\Big)\Big)\Big(1 - 2n^2\exp\Big(-\frac{d}{Cn^{2/r}}\Big)\Big)\\ &\geq 1 - n\exp\Big(-\frac{3m\eta_r^2(\sigma)}{64M_\sigma^2 n}\Big) - 2n^2\exp\Big(-\frac{d}{2Cn^{2/r}}\Big).\end{aligned}$$

□

3.D Counter-example for the ReLU case.

This section gives a proof sketch to illustrate that, with the ReLU activation $\sigma : x \mapsto \max(0, x)$, the smoothness of the weights can be lost during training. More precisely, we show a case where successive weights are at distance $\mathcal{O}(\frac{1}{L})$ at initialization and at distance $\Omega(1)$ after training.

For the sake of simplicity, we will assume that the depth is even, and denote it as $2L$. We place ourselves in a one-dimensional setting (i.e., $d = 1$). The parameters are $(w_1, \cdots, w_{2L}) \in \mathbb{R}^{2L}$, and the residual network writes as follows, for an input $x \in \mathbb{R}$:

$$h_0(t) = x$$
$$h_{k+1}(t) = h_k(t) + \frac{1}{2L}\sigma(w_{k+1}(t)h_k(t)), \quad k \in \{0, \dots, 2L-1\}.$$

We consider a sample consisting of a single point $(x, Cx) \in \mathbb{R}^2_+$, with $C > 1$ (independent of L), and define the empirical risk as $\ell(t) = (h_{2L}(t) - Cx)^2$. The risk is minimized by gradient flow.

The weights are initialized to $w_k(0) = \frac{(-1)^k}{2L}$. For $x \in \mathbb{R}_+$ we have that $h_k(t) \geq 0$ for all $k \in \{0, \dots, 2L\}$. Note that the argument of σ on the odd layers is negative. Therefore, by definition of σ, the gradient of the loss with respect to the odd layers is zero and we have, for $k \in \{0, \dots, L-1\}$, $w_{2k+1}(t) = w_{2k+1}(0)$. On the other hand, the argument of σ is positive on the even layers, and thus,

$$h_{2L}(t) = \prod_{j=1}^{L} \Big(1 + \frac{w_{2j}(t)}{2L}\Big)x.$$

As a consequence, the gradient flow equation for the even layers is, for $k \in \{1, \dots, L\}$,

$$\frac{dw_{2k}}{dt}(t) = -\frac{\partial \ell}{\partial w_{2k}}(t) = 2x\Big(C - \prod_{j=1}^{L}\Big(1 + \frac{w_{2j}(t)}{2L}\Big)\Big)\prod_{j=1, j\neq k}^{L}\Big(1 + \frac{w_{2j}(t)}{2L}\Big).$$

Due to the symmetry of these equations for $k \in \{1, \dots, L\}$ and the fact that all the $w_{2k}(0)$ are equal, the parameters on each even layer coincide at all times and are equal to $w(t)$ such that

$$\frac{dw}{dt}(t) = 2x\Big(C - \Big(1 + \frac{w(t)}{2L}\Big)^L\Big)\Big(1 + \frac{w(t)}{2L}\Big)^{L-1}.$$

An analysis of this ODE reveals that $w(t)$ tends as $t \to \infty$ to $w^\star > 0$ satisfying that

$$\Big(1 + \frac{w^\star}{2L}\Big)^L = C. \tag{3.56}$$

This can be seen by letting $y(t) = C - (1 + \frac{w(t)}{2L})^L$, and applying Grönwall's inequality to y. Therefore, as $t \to \infty$, one has $w_{2k+1}(t) \to -\frac{1}{2L}$ and $w_{2k}(t) \to w^\star$, where (3.56) implies that $w^\star \geq 2\log(C)$. This shows that the final weights are not smooth in the sense that the distance between two successive weights is $\Omega(1)$.

This result contrasts sharply with Proposition 3.8, which shows that successive weights remain at a distance $\mathcal{O}(\frac{1}{L})$ throughout training, when initialized as a discretization of a Lipschitz continuous function, and with a smooth activation function. In fact, Proposition 3.8 can be generalized to any initialization such that successive weights are at distance $\mathcal{O}(\frac{1}{L})$ at initialization, which is the case in the counter-example. This means that the only broken assumption in our counter-example is the non-smoothness of the activation function. This non-smoothness causes the gradient flow dynamics for two successive weights to deviate, even though the weights are initially close to each other, because they are separated by the kink of ReLU at zero.

3.E Experimental details

We use PyTorch (Paszke et al., 2019).

Large-depth limit. We take $n = 100$, $d = 16$, $m = 32$. We train for 500 iterations, and set the learning rate to $L \times 10^{-2}$. The scaling of the learning rate with L is the equivalent of the L factor in the gradient flow (3.4).

Long-time limit. We take $n = 50$, $d = 16$, $m = 64$, $L = 64$, and train for 80,000 iterations with a learning rate of $5L \times 10^{-3}$.

Real-world data. We take $L = 256$. The first layer is a trainable convolutional layer with a kernel size of 5×5, a stride of 2, a padding of 1, and 16 out channels. We then iterate the residual layers

$$h_{k+1}^L = h_k^L + \frac{1}{L}\mathrm{bn}_{2,k}^L(\mathrm{conv}_{2,k}^L(\sigma(\mathrm{bn}_{1,k}^L(\mathrm{conv}_{1,k}^L(h_k^L))))), \quad k \in \{0, \dots, L-1\},$$

where $\mathrm{conv}_{i,k}^L$ are convolutions with kernel size 3, stride of 2, and padding of 1, and $\mathrm{bn}_{i,k}^L$ are batch normalizations, as is standard in residual networks (He et al., 2016a). The model is trained using stochastic gradient descent on the cross-entropy loss for 180 epochs. The initial learning rate is 4×10^{-2} and is gradually decreased using a cosine learning rate scheduler.

4 Generalization bounds for neural ODEs and deep residual networks

Neural ordinary differential equations (neural ODEs) are a popular family of continuous-depth deep learning models. In this work, we consider a large family of parameterized ODEs with continuous-in-time parameters, which include time-dependent neural ODEs. We derive a generalization bound for this class by a Lipschitz-based argument. By leveraging the analogy between neural ODEs and deep residual networks, our approach yields in particular a generalization bound for a class of deep residual networks. The bound involves the magnitude of the difference between successive weight matrices. We illustrate numerically how this quantity affects the generalization capability of neural networks.

Contents

4.1 Introduction

Neural ordinary differential equations (neural ODEs, Chen et al., 2018a) are a flexible family of neural networks used in particular to model continuous-time phenomena. Along with variants such as neural stochastic differential equations (neural SDEs, Tzen and Raginsky, 2019) and neural controlled differential equations (Kidger et al., 2020), they have been used in diverse fields such as pharmokinetics (Lu et al., 2021; Qian et al., 2021), finance (Gierjatowicz et al., 2020), and transportation (Zhou et al., 2021). We refer to Massaroli et al. (2020) for a self-contained introduction to this class of models.

Despite their empirical success, the statistical properties of neural ODEs have not yet been fully investigated. What is more, neural ODEs can be thought of as the infinite-depth limit of (properly scaled) residual neural networks (He et al., 2016a), a connection made by, e.g., E (2017); Haber and Ruthotto (2017); Lu et al. (2018). Since standard measures of statistical complexity of neural networks grow with depth (see, e.g., Bartlett et al., 2019), it is unclear why infinite-depth models, including neural ODEs, should enjoy favorable generalization properties.

To better understand this phenomenon, our goal in this chapter is to study the statistical properties of a class of time-dependent neural ODEs that write

$$dH_t = W_t \sigma(H_t) dt, \tag{4.1}$$

where $W_t \in \mathbb{R}^{d \times d}$ is a weight matrix that depends on the time index t, and $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function applied component-wise. Time-dependent neural ODEs were first introduced by Massaroli et al. (2020) and generalize time-independent neural ODEs

$$dH_t = W \sigma(H_t) dt, \tag{4.2}$$

as formulated in Chen et al. (2018a), where $W \in \mathbb{R}^{d \times d}$ now denotes a weight matrix independent of t. There are two crucial reasons to consider time-dependent neural ODEs rather than the more restrictive class of time-independent neural ODEs. On the one hand, the time-dependent formulation is more flexible, leading to competitive results on image classification tasks (Queiruga et al., 2020, 2021). As a consequence, obtaining generalization guarantees for this family of models is a valuable endeavor by itself. On the other hand, time dependence is required for the correspondence with general residual neural networks to hold. More precisely, the time-dependent neural ODE (4.1) is the limit, when the depth L goes to infinity, of the deep residual network

$$H_{k+1} = H_k + \frac{1}{L} W_{k+1} \sigma(H_k), \quad 0 \leqslant k \leqslant L-1, \tag{4.3}$$

where $(W_k)_{1 \leq k \leq L} \in \mathbb{R}^{d \times d}$ are weight matrices and σ is still an activation function. We refer to Chapter 2 and to Sander et al. (2022b); Thorpe and van Gennip (2022) for statements that make precise under what conditions and in which sense this limit holds, as well as its consequences for learning. These two key reasons compel us to consider the class of time-dependent ODEs (4.1) for our statistical study, which in turn will inform us on the properties of the models (4.2) and (4.3).

In fact, we extend our study to the larger class of *parameterized ODEs*, which we define as the mapping from $x \in \mathbb{R}^d$ to the value at time $t = 1$ of the solution of the initial value problem

$$H_0 = x, \qquad dH_t = \sum_{i=1}^{m} \theta_i(t) f_i(H_t) dt, \tag{4.4}$$

where H_t is the variable of the ODE, θ_i are functions from $[0, 1]$ into $\mathbb{R}$ that parameterize the ODE, and f_i are fixed functions from $\mathbb{R}^d$ into $\mathbb{R}^d$. Time-dependent neural ODEs (4.1) are obtained by

setting a specific entrywise form for the functions f_i in (4.4), with $m = d^2$ (see Section 4.3.3 for details).

Since the parameters θ_i belong to an infinite-dimensional space, in practice they need to be approximated in a finite-dimensional basis of functions. For example, the residual neural networks (4.3) can be seen as an approximation of the neural ODEs (4.1) on a piecewise constant basis of function. But more complex choices are possible, such as B-splines (Yu et al., 2022). However, the formulation (4.4) is agnostic from the choice of finite-dimensional approximation. This more abstract point of view is fruitful to derive generalization bounds, for at least two reasons. First, the statistical properties of the parameterized ODEs (4.4) only depend on the characteristics of the functions θ_i and not on the specifics of the approximation scheme, so it is more natural and convenient to study them at the continuous level. Second, their properties can then be transferred to any specific discretization, such as the deep residual networks (4.3), resulting in generalization bounds for the latter.

Regarding the characteristics of the functions θ_i, we make the structural assumption that they are Lipschitz-continuous and uniformly bounded. This is a natural assumption to ensure that the initial value problem (4.4) has a unique solution in the usual sense of the Picard-Lindelöf theorem. Remarkably, this assumption on the parameters also enables us to obtain statistical guarantees despite the fact that we are working with an infinite-dimensional set of parameters.

Contributions. We provide a generalization bound for the large class of parameterized ODEs (4.4), which include time-dependent and time-independent neural ODEs (4.1) and (4.2). To the best of our knowledge, this is the first available bound for neural ODEs. By leveraging the connection between (time-dependent) neural ODEs and deep residual networks, our approach allows us to provide a depth-independent generalization bound for the class of deep residual networks (4.3). The bound is precisely compared with earlier results. Our bound depends in particular on the magnitude of the difference between successive weight matrices, which is, to our knowledge, a novel way of controlling the statistical complexity of neural networks. Numerical illustration is provided to show the relationship between this quantity and the generalization ability of neural networks.

Organization of the chapter. Section 4.2 presents additional related work. In Section 4.3, we specify our class of parameterized ODEs, before stating the generalization bound for this class and for neural ODEs as a corollary. The generalization bound for residual networks is presented in Section 4.4 and compared to other bounds, before some numerical illustration. Section 4.5 concludes the chapter. The proof technique is discussed in the main part of the chapter, but the core of the proofs is relegated to Section 4.A. Finally, Section 4.B contains the details of the numerical illustrations presented in Section 4.4.3.

4.2 Related work

Hybridizing deep learning and differential equations. The fields of deep learning and dynamical systems have recently benefited from sustained cross-fertilization. On the one hand, a large line of work is aimed at modeling complex continuous-time phenomena by developing specialized neural architectures. This family includes neural ODEs, but also physics-informed neural networks (Raissi et al., 2019), neural operators (Li et al., 2021a) and neural flows (Biloš et al., 2021). On the other hand, successful recent advances in deep learning, such as diffusion models, are theoretically supported by ideas from differential equations (Huang et al., 2021).

Generalization for continuous-time neural networks. Obtaining statistical guarantees for continuous-time neural networks has been the topic of a few recent works. For example, we consider in Chapter 5 a class of continuous-time recurrent neural networks (RNNs) that can be written as input-driven ODEs. They show that these models are actually kernel methods, which entails a generalization bound. Lim et al. (2021) also show a generalization bound for ODE-like RNNs, and argue that adding stochasticity (that is, replacing ODEs with SDEs) helps with generalization. Yin et al. (2021) propose a neural ODE model to enable transfer learning across multiple environments and provide a generalization bound in this setting.

Lipschitz-based generalization bounds for deep neural networks. From a high-level perspective, our proof technique is similar to previous works (Bartlett et al., 2017; Neyshabur et al., 2018) that show generalization bounds for deep neural networks, which scale at most polynomially with depth. More precisely, these authors show that the network satisfies some Lipschitz continuity property (either with respect to the input or to the parameters), then exploit results on the statistical complexity of Lipschitz function classes. Under stronger norm constraints, these bounds can even be made depth-independent (Golowich et al., 2018). However, their approach differs from ours insofar as we consider neural ODEs and the associated family of deep neural networks, whereas they are solely interested in finite-depth neural networks. As a consequence, their hypotheses on the class of neural networks differ from ours. Section 4.4 develops a more thorough comparison. Similar Lipschitz-based techniques have also been applied to obtain generalization bounds for deep equilibrium networks (Pabbaraju et al., 2021). Going beyond statistical guarantees, Béthune et al. (2022) study approximation and robustness properties of Lipschitz neural networks.

4.3 Generalization bounds for parameterized ODEs

We start by recalling the usual supervised learning setup and introduce some notation in Section 4.3.1, before presenting our parameterized ODE model and the associated generalization bound in Section 4.3.2. We then apply the bound to the specific case of time-invariant neural ODEs in Section 4.3.3.

4.3.1 Learning procedure

We place ourselves in a supervised learning setting. Let us introduce the notation that is used throughout the chapter (up to Section 4.4.1). The input data is a sample of n i.i.d. pairs (x_i, y_i) with the same distribution as some generic pair (x, y), where x (resp. y) takes its values into some bounded ball $\mathcal{X} = B(0, R_{\mathcal{X}})$ (resp. $\mathcal{Y} = B(0, R_{\mathcal{Y}})$) of $\mathbb{R}^d$, for some $R_{\mathcal{X}}, R_{\mathcal{Y}} > 0$. This setting encompasses regression but also classification tasks by (one-hot) encoding labels in $\mathbb{R}^d$. Note that we assume for simplicity that the input and output have the same dimension, but our analysis easily extends to the case where they have different dimensions by adding (parameterized) projections at the beginning or at the end of our model. Given a parameterized class of models $\mathcal{F}_\Theta = \{F_\theta, \theta \in \Theta\}$, the parameter θ is fitted by empirical risk minimization using a loss function $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ that we assume to be Lipschitz with respect to its first argument, with a Lipschitz constant $K_\ell > 0$. In the following, we write for the sake of conciseness that such a function is K_ℓ-Lipschitz. We also assume that $\ell(x, x) = 0$ for all $x \in \mathbb{R}^d$. The theoretical and empirical risks are respectively defined, for any $\theta \in \Theta$, by

$$\mathscr{R}(\theta) = \mathbb{E}[\ell(F_\theta(x), y)] \quad \text{and} \quad \widehat{\mathscr{R}}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(F_\theta(x_i), y_i\big),$$

where the expectation $\mathbb{E}$ is evaluated with respect to the distribution of (x, y). Letting $\widehat{\theta}_n$ a minimizer of the empirical risk, the generalization problem consists in providing an upper bound on the difference $\mathscr{R}(\widehat{\theta}_n) - \widehat{\mathscr{R}}_n(\widehat{\theta}_n)$.

4.3.2 Generalization bound

Model. We start by making more precise the parameterized ODE model introduced in Section 4.1. The setup presented here can easily be specialized to the case of neural ODEs, as we will see in Section 4.3.3. Let $f_1, \dots, f_m : \mathbb{R}^d \to \mathbb{R}^d$ be fixed K_f-Lipschitz functions for some $K_f > 0$. Denote by M their supremum on $\mathcal{X}$ (which is finite since these functions are continuous). The parameterized ODE F_θ is defined by the following initial value problem that maps some $x \in \mathbb{R}^d$ to $F_\theta(x) \in \mathbb{R}^d$:

$$\begin{aligned} H_0 &= x \\ dH_t &= \sum_{i=1}^{m} \theta_i(t) f_i(H_t) dt \\ F_\theta(x) &= H_1, \end{aligned} \tag{4.5}$$

where the parameter $\theta = (\theta_1, \dots, \theta_m)$ is a function from $[0, 1]$ to $\mathbb{R}^m$. We have to impose constraints on θ for the model F_θ to be well-defined. To this aim, we endow (essentially bounded) functions from $[0, 1]$ to $\mathbb{R}^m$ with the following $(1, \infty)$-norm

$$\|\theta\|_{1,\infty} = \sup_{0 \leqslant t \leqslant 1} \sum_{i=1}^{m} |\theta_i(t)|. \tag{4.6}$$

We can now define the set of parameters

$$\Theta = \{\theta : [0, 1] \to \mathbb{R}^m,\ \|\theta\|_{1,\infty} \leqslant R_\Theta \text{ and } \theta_i \text{ is } K_\Theta\text{-Lipschitz for } i \in \{1, \dots, m\}\}, \tag{4.7}$$

for some $R_\Theta > 0$ and $K_\Theta \geqslant 0$. Then, for $\theta \in \Theta$, the following Proposition, which is a consequence of the Picard-Lindelöf Theorem, shows that the mapping $x \mapsto F_\theta(x)$ is well-defined.

Proposition 4.1 (Well-posedness of the parameterized ODE)**.** *For $\theta \in \Theta$ and $x \in \mathbb{R}^d$, there exists a unique solution to the initial value problem* (4.5)*.*

Proof. See Section 4.A.1. □

An immediate consequence of Proposition 4.1 is that it is legitimate to take $\mathcal{F}_\Theta = \{F_\theta, \theta \in \Theta\}$ for our model class.

When $K_\Theta = 0$, the parameter space Θ is finite-dimensional since each θ_i is constant. This setting corresponds to the time-independent neural ODEs of Chen et al. (2018a). In this case, the norm (4.6) reduces to the $\|\cdot\|_1$ norm over $\mathbb{R}^m$. Note that, to fit exactly the formulation of Chen et al. (2018a), the time t can be added as a variable of the functions f_i, which amounts to adding a new coordinate to H_t. This does not change the subsequent analysis. In the richer time-dependent case where $K_\Theta > 0$, the set Θ belongs to an infinite-dimensional space and therefore, in practice, θ_i is approximated in a finite basis of functions, such as Fourier series, Chebyshev polynomials, and splines. We refer to Massaroli et al. (2020) for a more detailed discussion, including formulations of the back-propagation algorithm (a.k.a. the adjoint method) in this setting.

Note that we consider the case where the dynamics at time t are linear with respect to the parameter $\theta_i(t)$. Nevertheless, we emphasize that the mapping $x \mapsto F_\theta(x)$ remains a highly non-linear function of each $\theta_i(t)$. To fix ideas, this setting can be seen as analogous to working with pre-activation residual networks instead of post-activation (see He et al., 2016b, for definitions of the terminology), which is a mild modification.

Statistical analysis. Since Θ is a subset of an infinite-dimensional space, complexity measures based on the number of parameters cannot be used. Instead, our approach is to resort to Lipschitz-based complexity measures. More precisely, to bound the complexity of our model class, we propose two building blocks: we first show that the model F_θ is Lipschitz-continuous with respect to its parameters θ. This allows us to bound the complexity of the model class depending on the complexity of the parameter class. In a second step, we assess the complexity of the class of parameters itself.

Starting with our first step, we show the following estimates for our class of parameterized ODEs. Here and in the following, $\|\cdot\|$ denotes the ℓ_2 norm over $\mathbb{R}^d$.

Proposition 4.2 (The parameterized ODE is bounded and Lipschitz)**.** *Let θ and $\tilde{\theta} \in \Theta$. Then, for any $x \in \mathcal{X}$,*

$$\|F_\theta(x)\| \leqslant R_{\mathcal{X}} + M R_\Theta \exp(K_f R_\Theta)$$

and

$$\|F_\theta(x) - F_{\tilde{\theta}}(x)\| \leqslant 2M K_f R_\Theta \exp(2K_f R_\Theta) \|\theta - \tilde{\theta}\|_{1,\infty}.$$

Proof. See Section 4.A.2. □

The proof, given in the Appendix, makes extensive use of Grönwall's inequality, a standard tool to obtain estimates in the theory of ODEs, in order to bound the magnitude of the solution H_t of (4.5).

The next step is to assess the magnitude of the covering number of Θ. Recall that, for $\varepsilon > 0$, the ε-covering number of a metric space is the number of balls of radius ε needed to completely cover the space, with possible overlaps.

Proposition 4.3 (Covering number of the ODE parameter class)**.** *For $\varepsilon > 0$, let $\mathcal{N}(\varepsilon)$ be the ε-covering number of Θ endowed with the $(1,\infty)$-norm (4.6). Then*

$$\log \mathcal{N}(\varepsilon) \leqslant m \log\Big(\frac{16 m R_\Theta}{\varepsilon}\Big) + \frac{m^2 K_\Theta \log(4)}{\varepsilon}.$$

Proof. See Section 4.A.3. □

Proposition 4.3 is a consequence of a classical result, see, e.g., Kolmogorov and Tikhomirov (1959, example 3 of paragraph 2). A self-contained proof is given in the Appendix for completeness. We also refer to Gottlieb et al. (2017) for more general results on covering numbers of Lipschitz functions.

The two propositions above and an ε-net argument allow to prove the first main result of the chapter.

Theorem 4.4 (Generalization bound for parameterized ODEs)**.** *Consider the class of parameterized ODEs $\mathcal{F}_\Theta = \{F_\theta, \theta \in \Theta\}$, where F_θ is given by (4.5) and Θ by (4.7). Let $\delta > 0$.*

Then, for $n \geq 9 \max(m^{-2} R_\Theta^{-2}, 1)$, with probability at least $1 - \delta$,

$$\mathscr{R}(\widehat{\theta}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + B\sqrt{\frac{(m+1)\log(R_\Theta m n)}{n}} + B\frac{m\sqrt{K_\Theta}}{n^{1/4}} + \frac{B}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}},$$

where B is a constant depending on $K_\ell, K_f, R_\Theta, R_{\mathcal{X}}, R_{\mathcal{Y}}, M$. More precisely,

$$B = 6 K_\ell K_f \exp(K_f R_\Theta)\big(R_{\mathcal{X}} + M R_\Theta \exp(K_f R_\Theta) + R_{\mathcal{Y}}\big).$$

Proof. See Section 4.A.4. □

Three terms appear in our upper bound of $\mathscr{R}(\widehat{\theta}_n) - \widehat{\mathscr{R}}_n(\widehat{\theta}_n)$. The first and the third ones are classical (see, e.g. Bach, 2023, Sections 4.4 and 4.5). On the contrary, the second term is more surprising with its convergence rate in $\mathcal{O}(n^{-1/4})$. This slower convergence rate is due to the fact that the space of parameters is infinite-dimensional. In particular, for $K_\Theta = 0$, corresponding to a finite-dimensional space of parameters, we recover the usual $\mathcal{O}(n^{-1/2})$ convergence rate, however at the cost of considering a much more restrictive class of models. Finally, it is noteworthy that the dimensionality appearing in the bound is not the input dimension d but the number of mappings m.

Note that this result is general and may be applied in a number of contexts that go beyond deep learning, as long as the instantaneous dependence of the ODE dynamics to the parameters is linear. One such example is the predator-prey model, describing the evolution of two populations of animals, which reads $dx_t = x_t(\alpha - \beta y_t)dt$ and $dy_t = -y_t(\gamma - \delta x_t)dt$, where x_t and y_t are real-valued variables and α, β, γ and δ are model parameters. This ODE falls into the framework of this section, if one were to estimate the parameters by empirical risk minimization. We refer to Deuflhard and Röblitz (2015, section 3) for other examples of parameterized biological ODE dynamics and methods for parameter identification.

Nevertheless, for the sake of brevity, we focus on applications of this result to deep learning, and more precisely to neural ODEs, which is the topic of the next section.

4.3.3 Application to neural ODEs

As explained in Section 4.1, parameterized ODEs include both time-dependent and time-independent neural ODEs. Since the time-independent model is more common in practice, we develop this case here and leave the time-dependent case to the reader (the result is actually given as Theorem 1.5 in Chapter 1). We thus consider the following neural ODE:

$$\begin{aligned} H_0 &= x \\ dH_t &= W\sigma(H_t)dt \\ F_W(x) &= H_1, \end{aligned} \tag{4.8}$$

where $W \in \mathbb{R}^{d\times d}$ is a weight matrix, and $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function applied component-wise. We assume σ to be K_σ-Lipschitz for some $K_\sigma > 0$. This assumption is satisfied by all common activation functions. To put the model in the form of Section 4.3.2, denote $e_1, \dots, e_d$ the canonical basis of $\mathbb{R}^d$. Then the dynamics (4.8) can be reformulated as

$$dH_t = \sum_{i,j=1}^{d} W_{ij}\sigma_{ij}(H_t)dt,$$

where $\sigma_{ij}(x) = \sigma(x_j)e_i$. Each σ_{ij} is itself K_σ-Lipschitz, hence we fall in the framework of Section 4.3.2. In other words, the functions f_i of our general parameterized ODE model form a shallow neural network with pre-activation. Denote by $\|W\|_{1,1}$ the sum of the absolute values of the elements of W. We consider the following set of parameters, which echoes the set Θ of Section 4.3.2:

$$\mathcal{W} = \{W \in \mathbb{R}^{d\times d}, \|W\|_{1,1} \leqslant R_{\mathcal{W}}\}, \tag{4.9}$$

for some $R_{\mathcal{W}} > 0$. We can then state the following result as a consequence of Theorem 4.4.

Corollary 4.5 (Generalization bound for neural ODEs)**.** *Consider the class of neural ODEs $\mathcal{F}_{\mathcal{W}} = \{F_W, W \in \mathcal{W}\}$, where F_W is given by* (4.8) *and $\mathcal{W}$ by* (4.9)*. Let $\delta > 0$.*

Then, for $n \geq 9R_{\mathcal{W}}^{-1}\max(d^{-4}R_{\mathcal{W}}^{-1}, 1)$*, with probability at least* $1-\delta$*,*

$$\mathscr{R}(\widehat{W}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{W}_n) + B(d+1)\sqrt{\frac{\log(R_{\mathcal{W}}dn)}{n}} + \frac{B}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}},$$

where B *is a constant depending on* $K_\ell, K_\sigma, R_{\mathcal{W}}, R_{\mathcal{X}}, R_{\mathcal{Y}}, M$*. More precisely,*

$$B = 6\sqrt{2}K_\ell K_\sigma \exp(K_\sigma R_{\mathcal{W}})\big(R_{\mathcal{X}} + MR_{\mathcal{W}}\exp(K_\sigma R_{\mathcal{W}}) + R_{\mathcal{Y}}\big).$$

Proof. See Section 4.A.5. □

Note that the term in $\mathcal{O}(n^{-1/4})$ from Theorem 4.4 is now absent. Since we consider a time-independent model, we are left with the other two terms, recovering a standard $\mathcal{O}(n^{-1/2})$ convergence rate.

4.4 Generalization bounds for deep residual networks

As highlighted in Section 4.1, there is a strong connection between neural ODEs and discrete residual neural networks. The previous study of the continuous case in Section 4.3 paves the way for deriving a generalization bound in the discrete setting of residual neural networks, which is of great interest given the pervasiveness of this architecture in modern deep learning.

We begin by presenting our model and result in Section 4.4.1, before detailing the comparison of our approach with other papers in Section 4.4.2 and giving some numerical illustration in Section 4.4.3.

4.4.1 Model and generalization bound

Model. We consider the following class of deep residual networks:

$$\begin{aligned} H_0 &= x \\ H_{k+1} &= H_k + \frac{1}{L}W_{k+1}\sigma(H_k), \quad 0 \leqslant k \leqslant L-1 \\ F_{\mathbf{W}}(x) &= H_L, \end{aligned} \tag{4.10}$$

where the parameter $\mathbf{W} = (W_k)_{1\leq k\leq L} \in \mathbb{R}^{L\times d\times d}$ is a set of weight matrices and σ is still a K_σ-Lipschitz activation function. To emphasize that $\mathbf{W}$ is here a third-order tensor, as opposed to the case of time-invariant neural ODEs in Section 4.3.3, where W was a matrix, we denote it with a bold notation. We also assume in the following that $\sigma(0) = 0$. This assumption could be alleviated at the cost of additional technicalities. Owing to the $1/L$ scaling factor, the deep limit of this residual network is a (time-dependent) neural ODE of the form studied in Section 4.3. We refer to Chapter 2 for further discussion on the link between scaling factors and deep limits. We simply note that this scaling factor is not common practice, but preliminary experiments show it does not hurt performance and can even improve performance in a weight-tied setting (Sander et al., 2022b). The space of parameters is endowed with the following $(1,1,\infty)$-norm

$$\|\mathbf{W}\|_{1,1,\infty} = \sup_{1\leqslant k\leqslant L}\sum_{i,j=1}^{d}|W_{k,i,j}|. \tag{4.11}$$

Also denoting $\|\cdot\|_\infty$ the element-wise maximum norm for a matrix, we consider the class of matrices

$$\begin{aligned} \mathcal{W} = \Big\{\mathbf{W} \in \mathbb{R}^{L\times d\times d}, \quad & \|\mathbf{W}\|_{1,1,\infty} \leqslant R_{\mathcal{W}} \quad \text{and} \\ & \|W_{k+1} - W_k\|_\infty \leqslant \frac{K_{\mathcal{W}}}{L} \ \text{ for } \ 1 \leq k \leq L-1\Big\}, \end{aligned} \tag{4.12}$$

for some $R_{\mathcal{W}} > 0$ and $K_{\mathcal{W}} \geqslant 0$, which is a discrete analogous of the set Θ defined by (4.7). In particular, the upper bound on the difference between successive weight matrices is to our knowledge a novel way of constraining the parameters of a neural network. It corresponds to the discretization of the Lipschitz continuity of the parameters introduced in (4.7). By analogy, we refer to it as a constraint on the Lipschitz constant of the weights. Note that, for standard initialization schemes, the difference between two successive matrices is of the order $\mathcal{O}(1)$ and not $\mathcal{O}(1/L)$, or, in other words, $K_{\mathcal{W}}$ scales as $\mathcal{O}(L)$. This issue can be solved by adding correlations across layers at initialization by taking, for $k \in \{1, \dots, L\}$ and $i, j \in \{1, \dots, d\}$, $\mathbf{W}_{k,i,j} = \frac{1}{\sqrt{d}} f_{i,j}(\frac{k}{L})$, where $f_{i,j}$ a smooth function, for example a Gaussian process with an RBF kernel. Note that such a non-i.i.d. initialization scheme is necessary for the correspondence between deep residual networks and neural ODEs to hold, as shown in Chapter 2. Furthermore, Sander et al. (2022b) prove that, with such an initialization scheme, the constraint on the Lipschitz constant also holds for the *trained* network, with $K_{\mathcal{W}}$ independent of L.

Statistical analysis. At first sight, a reasonable strategy would be to bound the distance between the model (4.10) and its limit $L \to \infty$ that is a parameterized ODE, then *apply* Theorem 4.4. This idea is straightforward, but comes at the cost of an additional $\mathcal{O}(1/L)$ term in the generalization bound, as a consequence of the discretization error between the discrete iterations (4.10) and their continuous limit. For example, this strategy is used in Chapter 5 to prove a generalization bound for discrete RNNs and the additional error term is incurred. We follow another way by mimicking all the proof with a finite L. This is a longer approach but it yields a sharper result since we avoid the $\mathcal{O}(1/L)$ discretization error. The proof structure is similar to Section 4.3: the following two Propositions are the discrete counterparts of Propositions 4.2 and 4.3.

Proposition 4.6 (The residual network is bounded and Lipschitz)**.** *Let* $\mathbf{W}$ *and* $\tilde{\mathbf{W}} \in \mathcal{W}$*. Then, for any* $x \in \mathcal{X}$*,*

$$\|F_{\mathbf{W}}(x)\| \leqslant R_{\mathcal{X}} \exp(K_\sigma R_{\mathcal{W}})$$

and

$$\|F_{\mathbf{W}}(x) - F_{\tilde{\mathbf{W}}}(x)\| \leqslant \frac{R_{\mathcal{X}}}{R_{\mathcal{W}}} \exp(2K_\sigma R_{\mathcal{W}}) \|\mathbf{W} - \tilde{\mathbf{W}}\|_{1,1,\infty}.$$

Proof. See Section 4.A.6. □

Proposition 4.7 (Covering number of the residual network parameter class)**.** *Let* $\mathcal{N}(\varepsilon)$ *be the covering number of* $\mathcal{W}$ *endowed with the* $(1, 1, \infty)$*-norm* (4.11)*. Then*

$$\log \mathcal{N}(\varepsilon) \leqslant d^2 \log\Big(\frac{16 d^2 R_{\mathcal{W}}}{\varepsilon}\Big) + \frac{d^4 K_{\mathcal{W}} \log(4)}{\varepsilon}.$$

Proof. See Section 4.A.7. □

The proof of Proposition 4.6 is a discrete analogous of Proposition 4.2. On the other hand, Proposition 4.7 can be proven as a *consequence* of Proposition 4.3, by showing the existence of an injective isometry from $\mathcal{W}$ into a set of the form (4.7). Equipped with these two propositions, we are now ready to state the generalization bound for our class of residual neural networks.

Theorem 4.8 (Generalization bound for deep residual networks)**.** *Consider the class of neural networks* $\mathcal{F}_{\mathcal{W}} = \{F_{\mathbf{W}}, \mathbf{W} \in \mathcal{W}\}$*, where* $F_{\mathbf{W}}$ *is given by* (4.10) *and* $\mathcal{W}$ *by* (4.12)*. Let* $\delta > 0$*.*

Then, for $n \geq 9 R_{\mathcal{W}}^{-1} \max(d^{-4} R_{\mathcal{W}}^{-1}, 1)$*, with probability at least* $1 - \delta$*,*

$$\mathscr{R}(\widehat{\mathbf{W}}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\mathbf{W}}_n) + B(d+1)\sqrt{\frac{\log(R_{\mathcal{W}} d n)}{n}} + B\frac{d^2\sqrt{K_{\mathcal{W}}}}{n^{1/4}} + \frac{B}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}, \tag{4.13}$$

where B is a constant depending on $K_\ell, K_\sigma, R_\mathcal{W}, R_\mathcal{X}, R_\mathcal{Y}$. More precisely,

$$B = 6\sqrt{2}K_\ell \max\Big(\frac{\exp(K_\sigma R_\mathcal{W})}{R_\mathcal{W}}, 1\Big)(R_\mathcal{X}\exp(K_\sigma R_\mathcal{W}) + R_\mathcal{Y}).$$

Proof. See Section 4.A.8. □

We emphasize that this result is non-asymptotic and valid for any width d and depth L. Furthermore, the depth L does not appear in the upper bound (4.13). This should not surprise the reader since Theorem 4.4 can be seen as the deep limit $L \to \infty$ of this result, hence we expect that our bound remains finite when $L \to \infty$ (otherwise the bound of Theorem 4.4 would be infinite). However, L appears as a scaling factor in the definition of the neural network (4.10) and of the class of parameters (4.12). This is crucial for the depth independence to hold, as we will comment further on in the next section.

Furthermore, the depth independence comes at the price of a $\mathcal{O}(n^{-1/4})$ convergence rate. Note that, by taking $K_\mathcal{W} = 0$, we obtain a generalization bound for weight-tied neural networks with a faster convergence rate in n, since the term in $\mathcal{O}(n^{-1/4})$ vanishes.

4.4.2 Comparison with other bounds

As announced in Section 4.2, we now compare Theorem 4.8 with the results of Bartlett et al. (2017) and Golowich et al. (2018). Beginning by Bartlett et al. (2017), we first state a slightly weaker version of their result to match our notations and facilitate comparison.

Corollary 4.9 (of Theorem 1.1 of Bartlett et al. (2017))**.** *Consider the class of neural networks $\mathcal{F}_{\tilde{\mathcal{W}}} = \{F_\mathbf{W}, \mathbf{W} \in \tilde{\mathcal{W}}\}$, where $F_\mathbf{W}$ is given by (4.10) and $\tilde{\mathcal{W}} = \{\mathbf{W} \in \mathbb{R}^{L\times d\times d}, \|\mathbf{W}\|_{1,1,\infty} \leqslant R_\mathcal{W}\}$.*

Assume that $L \geq R_\mathcal{W}$ and $K_\sigma = 1$, and let $\gamma, \delta > 0$. Consider $(x, y), (x_1, y_1), \dots, (x_n, y_n)$ drawn i.i.d. from any probability distribution over $\mathbb{R}^d \times \{1, \dots d\}$ such that a.s. $\|x\| \leq R_\mathcal{X}$.

Then, with probability at least $1 - \delta$, for every $\mathbf{W} \in \tilde{\mathcal{W}}$,

$$\mathbb{P}\Big(\operatorname*{argmax}_{1\leq j\leq d} F_\mathbf{W}(x)_j \neq y\Big) \leq \widehat{\mathscr{R}}_n(\mathbf{W}) + C\frac{R_\mathcal{X} R_\mathcal{W}\exp(R_\mathcal{W})\log(d)\sqrt{L}}{\gamma\sqrt{n}} + \frac{C}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}, \tag{4.14}$$

where $\widehat{\mathscr{R}}_n(\mathbf{W}) \leq n^{-1}\sum_{i=1}^n \mathbf{1}_{F_\mathbf{W}(x_i)_{y_i} \leq \gamma + \max_{j\neq y_i} f(x_i)_j}$ and C is a universal constant.

Proof. See Section 4.A.9. □

We first note that the setting is slightly different from ours: they consider a large margin predictor for a multi-class classification problem, whereas we consider a general Lipschitz-continuous loss ℓ. This being said, the model class is identical to ours, except for one notable difference: the constraint on the Lipschitz constant of the weights appearing in equation (4.12) is not required here.

Comparing (4.13) and (4.14), we see that our bound enjoys a better dependence on the depth L but a worse dependence on the width d. Regarding the depth, our bound (4.13) does not depend on L, whereas the bound (4.14) scales as $\mathcal{O}(\sqrt{L})$. This comes from the fact that we consider a smaller set of parameters (4.12), by adding the constraint on the Lipschitz norm of the weights. This constraint allows us to control the complexity of our class of neural networks independently of depth, as long as $K_\mathcal{W}$ is independent of L. If $K_\mathcal{W}$ scales as $\mathcal{O}(L)$, which is the case for i.i.d. initialization schemes, our result also features a scaling in $\mathcal{O}(\sqrt{L})$. As for the width, Bartlett et al. (2017) achieve a better dependence by a subtle covering numbers argument that takes into account the geometry induced by matrix norms. Since the chapter focuses on a depth-wise analysis by leveraging the similarity between residual networks and their

infinite-depth counterpart, improving the scaling of our bound with width is left for future work. Finally, note that both bounds have a similar exponential dependence in $R_{\mathcal{W}}$.

As for Golowich et al. (2018), they consider non-residual neural networks of the form $x \mapsto M_L\sigma(M_{L-1}\sigma(\dots\sigma(M_1x)))$. These authors show that the generalization error of this class scales as

$$\mathcal{O}\Bigg(R_{\mathcal{X}}\frac{\Pi_F\sqrt{\log\left(\frac{\Pi_F}{\pi_S}\right)}}{n^{1/4}}\Bigg),$$

where Π_F is an upper-bound on the product of the Frobenius norms $\prod_{k=1}^L \|M_k\|_F$ and π_S is a lower-bound on the product of the spectral norms $\prod_{k=1}^L \|M_k\|$. Under the assumption that both Π_F and Π_F/π_S are bounded independently of L, their bound is indeed depth-independent, similarly to ours. Interestingly, as ours, the bound presents a $\mathcal{O}(n^{-1/4})$ convergence rate instead of the more usual $\mathcal{O}(n^{-1/2})$. However, the assumption that Π_F is bounded independently of L does not hold in our residual setting, since we have $M_k = I + \frac{1}{L}W_k$ and thus we can lower-bound

$$\prod_{k=1}^L \|M_k\|_F \geqslant \prod_{k=1}^L \Big(\|I\|_F - \frac{1}{L}\|M_k\|_F\Big) \geq \big(\sqrt{d} - \frac{R_{\mathcal{W}}}{L}\big)^L \approx d^{\frac{L}{2}}e^{-\frac{R_{\mathcal{W}}}{\sqrt{d}}}.$$

In our setting, it is a totally different assumption, the constraint that two successive weight matrices should be close to one another, which allows us to derive depth-independent bounds.

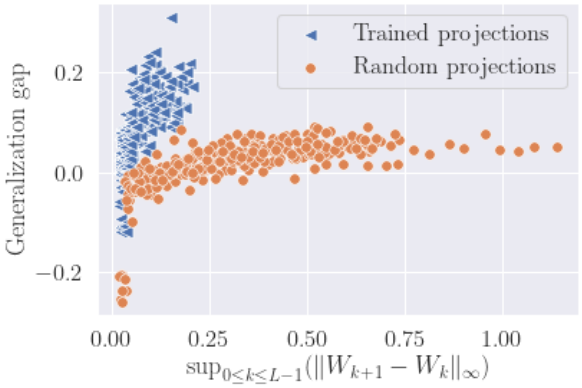
4.4.3 Numerical illustration

The bound of Theorem 4.8 features two quantities that depend on the class of neural networks, namely $R_{\mathcal{W}}$ that bounds a norm of the weight matrices and $K_{\mathcal{W}}$ that bounds the maximum *difference* between two successive weight matrices, a.k.a. the Lipschitz constant of the weights. The first one belongs to the larger class of norm-based bounds that has been extensively studied (see, e.g., Neyshabur et al., 2015a). We are therefore interested in getting a better understanding of the role of the second quantity, which is much less common, in the generalization ability of deep residual networks.

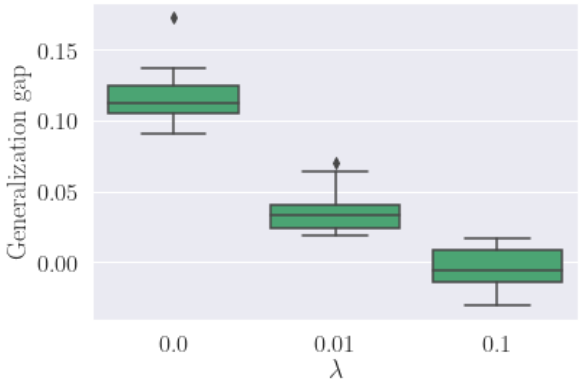
To this aim, we train deep residual networks (4.10) (of width $d = 30$ and depth $L = 1000$) on MNIST. We prepend the network with an initial weight matrix to project the data x from dimension 768 to dimension 30, and similarly postpend it with another matrix to project the output $M_{\mathbf{W}}(x)$ into dimension 10 (i.e. the number of classes in MNIST). Finally, we consider two training settings: either the initial and final matrices are trained, or they are fixed random projections. We use the initialization scheme outlined in Section 4.4.1. Further experimental details are postponed to the Appendix.

We report in Figure 4.1a the generalization gap of the trained networks, that is, the difference between the test and train errors (in terms of cross entropy loss), as a function of the maximum Lipschitz constant of the weights $\sup_{0\leq k\leq L-1}(\|W_{k+1} - W_k\|_\infty)$. We observe a positive correlation between these two quantities. To further analyze the relationship between the Lipschitz constant of the weights and the generalization gap, we then add the penalization term $\lambda \cdot \big(\sum_{k=0}^{L-1}\|W_{k+1} - W_k\|_F^2\big)^{1/2}$ to the loss, for some $\lambda \geqslant 0$. The obtained generalization gap is reported in Figure 4.1b as a function of λ. We observe that this penalization allows to reduce the generalization gap. These two observations go in support of the fact that a smaller Lipschitz constant improves the generalization power of deep residual networks, in accordance with Theorem 4.8.

However, note that we were not able to obtain an improvement on the test loss by adding the penalization term. This is not all too surprising since previous work has investigated a related penalization, in terms of the Lipschitz norm of the layer sequence $(H_k)_{0\leq k\leq L}$, and was similarly not able to report any improvement on the test loss (Kelly et al., 2020).



(a) Generalization gap as a function of the maximum Lipschitz constant of the weights. Each dot corresponds to a network trained with a varying number of epochs (between 1 and 30).



(b) Generalization gap as a function of the penalization factor λ. The experiment is repeated 20 times for each value of λ. Each time, the network is trained for 50 epochs. The initial and final matrices are random.

Figure 4.1: Link between the generalization gap and the Lipschitz constant of the weights

4.5 Conclusion

We provide a generalization bound that applies to a wide range of parameterized ODEs. As a consequence, we obtain the first generalization bounds for time-independent and time-dependent neural ODEs. By discretizing our reasoning, we also provide a bound for a class of deep residual networks. In the future, it should also be interesting to extend our result to the more involved case of neural SDEs, which have also been found to be deep limits of a large class of residual neural networks, as shown in Chapter 2 and in Cohen et al. (2021).

4.A Proofs

4.A.1 Proof of Proposition 4.1

The function

$$(t,h)\mapsto\sum_{i=1}^{m}\theta_i(t)f_i(h)$$

is locally Lipschitz-continuous with respect to its first variable and globally Lipschitz-continuous with respect to its second variable. Therefore the existence and uniqueness of the solution of the initial value problem (4.5) for $t\geq 0$ comes as a consequence of the Picard-Lindelöf theorem (see, e.g., Luk, 2017 for a self-contained presentation and Arnold, 1992 for a textbook).

4.A.2 Proof of Proposition 4.2

For $x\in\mathcal{X}$, let H be the solution of the initial value problem (4.5) with parameter θ and with the initial condition $H_0=x$. Let us first upper-bound $\|f_i(H_t)\|$ for all $i\in\{1,\dots,m\}$ and $t>0$.

To this aim, for $t \geq 0$, we have

$$\begin{aligned}
\|H_t - H_0\| &= \Big\| \int_0^t \sum_{i=1}^m \theta_i(s) f_i(H_s) ds \Big\| \\
&\leqslant \int_0^t \sum_{i=1}^m |\theta_i(s)| \|f_i(H_0)\| ds + \int_0^t \sum_{i=1}^m |\theta_i(s)| \|f_i(H_s) - f_i(H_0)\| ds \\
&\leqslant M \int_0^t \sum_{i=1}^m |\theta_i(s)| ds + K_f \int_0^t \Big(\|H_s - H_0\| \sum_{i=1}^m |\theta_i(s)| \Big) ds \\
&\leqslant tMR_\Theta + K_f R_\Theta \int_0^t \|H_s - H_0\| ds.
\end{aligned}$$

Next, Grönwall's inequality yields, for $t \in [0,1]$,

$$\|H_t - H_0\| \leqslant tMR_\Theta \exp(tK_f R_\Theta) \leqslant MR_\Theta \exp(K_f R_\Theta).$$

Hence

$$\|H_t\| \leqslant \|H_0\| + \|H_t - H_0\| \leqslant R_{\mathcal{X}} + MR_\Theta \exp(K_f R_\Theta),$$

yielding the first result of the proposition. Furthermore, for any $i \in \{1, \dots, m\}$,

$$\|f_i(H_t)\| \leqslant \|f_i(H_t) - f_i(H_0)\| + \|f_i(H_0)\| \leqslant M\big(K_f R_\Theta \exp(K_f R_\Theta) + 1\big) =: C.$$

Now, let $\tilde{H}$ be the solution of the initial value problem (4.5) with another parameter $\tilde{\theta}$ and with the same initial condition $\tilde{H}_0 = x$. Then, for any $t \geqslant 0$,

$$H_t - \tilde{H}_t = \int_0^t \sum_{i=1}^m \theta_i(s) f_i(H_s) ds - \int_0^t \sum_{i=1}^m \tilde{\theta}_i(s) f_i(\tilde{H}_s) ds.$$

Hence

$$\begin{aligned}
\|H_t - \tilde{H}_t\| &= \Big\| \int_0^t \sum_{i=1}^m (\theta_i(s) - \tilde{\theta}_i(s)) f_i(H_s) ds + \int_0^t \sum_{i=1}^m \tilde{\theta}_i(s) (f_i(H_s) - f_i(\tilde{H}_s)) ds \Big\| \\
&\leqslant \int_0^t \sum_{i=1}^m |\theta_i(s) - \tilde{\theta}_i(s)| \|f_i(H_s)\| ds + \int_0^t \sum_{i=1}^m |\tilde{\theta}_i(s)| \|f_i(H_s) - f_i(\tilde{H}_s)\| ds \\
&\leqslant \int_0^t \sum_{i=1}^m |\theta_i(s) - \tilde{\theta}_i(s)| \|f_i(H_s)\| ds + K_f \int_0^t \Big(\|H_s - \tilde{H}_s\| \sum_{i=1}^m |\tilde{\theta}_i(s)| \Big) ds \\
&\leqslant tC \|\theta - \tilde{\theta}\|_{1,\infty} + K_f R_\Theta \int_0^t \|H_s - \tilde{H}_s\| ds.
\end{aligned}$$

Then Grönwall's inequality implies that, for $t \in [0,1]$,

$$\begin{aligned}
\|H_t - \tilde{H}_t\| &\leqslant tC \|\theta - \tilde{\theta}\|_{1,\infty} \exp(tK_f R_\Theta) \\
&\leqslant M(K_f R_\Theta \exp(K_f R_\Theta) + 1) \exp(K_f R_\Theta) \|\theta - \tilde{\theta}\|_{1,\infty} \\
&\leqslant 2MK_f R_\Theta \exp(2K_f R_\Theta) \|\theta - \tilde{\theta}\|_{1,\infty}
\end{aligned}$$

since $1 \leqslant K_f R_\Theta \exp(K_f R_\Theta)$ because $K_f \geqslant 1$, $R_\Theta \geqslant 1$.

4.A.3 Proof of Proposition 4.3

We first prove the result for $m = 1$. Let G_x be an $\varepsilon/2K_\Theta$-grid of $[0,1]$ and G_y an $\varepsilon/2$-grid of $[-R_\Theta, R_\Theta]$. Formally, we can take

$$G_x = \Big\{ \frac{k\varepsilon}{2K_\Theta},\, 0 \leqslant k \leqslant \Big\lceil \frac{2K_\Theta}{\varepsilon} \Big\rceil \Big\} \quad \text{and} \quad G_y = \Big\{ -R_\Theta + \frac{k\varepsilon}{2},\, 1 \leqslant k \leqslant \Big\lfloor \frac{4R_\Theta}{\varepsilon} \Big\rfloor \Big\}$$

Our cover consists of all functions that start at a point of G_y, are piecewise linear with kinks in G_x, where each piece has slope $+K_\Theta$ or $-K_\Theta$. Hence our cover is of size

$$\mathcal{N}_1(\varepsilon) = |G_y| 2^{|G_x|} \leqslant \frac{4R_\Theta}{\varepsilon} 2^{\frac{2K_\Theta}{\varepsilon}+2} = \frac{16R_\Theta}{\varepsilon} 4^{\frac{K_\Theta}{\varepsilon}}.$$

Now take a function $f : [0,1] \to \mathbb{R}$ that is uniformly bounded by R_Θ and K_Θ-Lipschitz. We construct a cover member at distance ε from f as follows. Choose a point y_0 in G_y at distance at most $\varepsilon/2$ from $f(0)$. Since $f(0) \in [-R_\Theta, R_\Theta]$, this is clearly possible, except perhaps at the end of the interval. To verify that it is possible at the end of the interval, note that R_Θ is at a distance less than $\varepsilon/2$ of the last element of the grid, since

$$R_\Theta - \Big(-R_\Theta + \Big\lfloor \frac{4R_\Theta}{\varepsilon} \Big\rfloor \frac{\varepsilon}{2} \Big) = 2R_\Theta - \Big\lfloor \frac{4R_\Theta}{\varepsilon} \Big\rfloor \frac{\varepsilon}{2} \in \Big[2R_\Theta - \frac{4R_\Theta}{\varepsilon}\frac{\varepsilon}{2}, 2R_\Theta - \Big(\frac{4R_\Theta}{\varepsilon} - 1 \Big) \frac{\varepsilon}{2} \Big] = \Big[0, \frac{\varepsilon}{2} \Big].$$

Then, among the cover members that start at y_0, choose the one which is closest to f at each point of G_x (in case of equality, pick any one). Let us denote this cover member as $\tilde{f}$. Let us show recursively that f is at ℓ_∞-distance at most ε from $\tilde{f}$. More precisely, let us first show by induction on k that for all $k \in \{0, \dots, \lceil \frac{2K_\Theta}{\varepsilon} \rceil\}$,

$$\big| f\big(\frac{k\varepsilon}{2K_\Theta}\big) - \tilde{f}\big(\frac{k\varepsilon}{2K_\Theta}\big) \big| \leqslant \frac{\varepsilon}{2}. \tag{4.15}$$

First, $|f(0) - \tilde{f}(0)| \leqslant \frac{\varepsilon}{2}$. Then, assume that (4.15) holds for some k. Then we have the following inequalities:

$$\begin{aligned}
\tilde{f}\big(\frac{k\varepsilon}{2K_\Theta}\big) - \varepsilon &\leqslant f\big(\frac{k\varepsilon}{2K_\Theta}\big) - \frac{\varepsilon}{2} && \text{(by induction)} \\
&\leqslant f\big(\frac{(k+1)\varepsilon}{2K_\Theta}\big) && (f \text{ is } K_\Theta\text{-Lipschitz}) \\
&\leqslant f\big(\frac{k\varepsilon}{2K_\Theta}\big) + \frac{\varepsilon}{2} && (f \text{ is } K_\Theta\text{-Lipschitz}) \\
&\leqslant \tilde{f}\big(\frac{k\varepsilon}{2K_\Theta}\big) + \varepsilon && \text{(by induction).}
\end{aligned}$$

Moreover, by definition, $\tilde{f}\big(\frac{(k+1)\varepsilon}{K_\Theta}\big)$ is the closest point to $f\big(\frac{(k+1)\varepsilon}{K_\Theta}\big)$ among

$$\Big\{ \tilde{f}\big(\frac{k\varepsilon}{K_\Theta}\big) - \frac{\varepsilon}{2}, \tilde{f}\big(\frac{k\varepsilon}{K_\Theta}\big) + \frac{\varepsilon}{2} \Big\}.$$

The bounds above show that, among those two points, at least one is at distance no more than $\varepsilon/2$ from $f\big(\frac{(k+1)\varepsilon}{K_\Theta}\big)$. This shows (4.15) at rank $k+1$.

To conclude, take now $x \in [0,1]$. There exists $k \in \{0, \dots, \lceil \frac{2K_\Theta}{\varepsilon} \rceil\}$ such that x is at distance at most $\varepsilon/4K_\Theta$ from $\frac{k\varepsilon}{2K_\Theta}$. Again, this is clear except perhaps at the end of the interval, where it is also true since

$$1 - \Big\lceil \frac{2K_\Theta}{\varepsilon} \Big\rceil \frac{\varepsilon}{2K_\Theta} \leq 1 - \frac{2K_\Theta}{\varepsilon}\frac{\varepsilon}{2K_\Theta} = 0,$$

meaning that 1 is located between two elements of the grid G_x, showing that it is at distance at most $\varepsilon/4K_\Theta$ from one element of the grid. Then, we have

$$\begin{aligned}|f(x) - \tilde{f}(x)| &\leqslant \Big|f(x) - f\big(\frac{k\varepsilon}{2K_\Theta}\big)\Big| + \Big|f\big(\frac{k\varepsilon}{2K_\Theta}\big) - \tilde{f}\big(\frac{k\varepsilon}{2K_\Theta}\big)\Big| + \Big|\tilde{f}\big(\frac{k\varepsilon}{2K_\Theta}\big) - \tilde{f}(x)\Big| \\ &\leqslant \frac{\varepsilon}{4} + \frac{\varepsilon}{2} + \frac{\varepsilon}{4},\end{aligned}$$

where the first and third terms are upper-bounded because f and $\tilde{f}$ are K_Θ-Lip, while the second term is upper bounded by (4.15). Hence $\|f - \tilde{f}\|_\infty \leqslant \varepsilon$, proving the result for $m = 1$.

Finally, to prove the result for a general m, note that the Cartesian product of ε/m-covers for each coordinate of θ gives an ε-cover for θ. Indeed, consider such covers and take $\theta \in \Theta$. Since each coordinate of θ is uniformly bounded by R_Θ and K_Θ-Lipschitz, the proof above shows the existence of a cover member $\tilde{\theta}$ such that, for all $i \in \{1, \dots, m\}$, $\|\theta_i - \tilde{\theta}_i\|_\infty \leq \varepsilon/m$. Then

$$\|\theta - \tilde{\theta}\|_{1,\infty} = \sup_{0 \leqslant t \leqslant 1} \sum_{i=1}^{m} |\theta_i(t) - \tilde{\theta}_i(t)| \leq \sup_{0 \leqslant t \leqslant 1} \sum_{i=1}^{m} \|\theta_i - \tilde{\theta}_i\|_\infty \leq \varepsilon.$$

As a consequence, we conclude that

$$\mathcal{N}(\varepsilon) \leq \Big(\mathcal{N}_1\big(\frac{\varepsilon}{m}\big)\Big)^m = \Big(\frac{16mR_\Theta}{\varepsilon}\Big)^m 4^{\frac{m^2 K_\Theta}{\varepsilon}}.$$

Taking the logarithm yields the result.

4.A.4 Proof of Theorem 4.4

First note that, for any $\theta \in \Theta$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,

$$|\ell(F_\theta(x), y)| \leqslant |\ell(F_\theta(x), y) - \ell(y, y)| + |\ell(y, y)| \leqslant K_\ell \|F_\theta(x) - y\|,$$

since, by assumption, ℓ is K_ℓ-Lipschitz with respect to its first variable and $\ell(y, y) = 0$. Thus

$$|\ell(F_\theta(x), y)| \leqslant K_\ell(\|F_\theta(x)\| + \|y\|) \leqslant K_\ell\big(R_\mathcal{X} + MR_\Theta \exp(K_f R_\Theta) + R_\mathcal{Y}\big) =: \overline{M},$$

by Proposition 4.2.

Now, taking $\delta > 0$, a classical computation involving McDiarmid's inequality (see, e.g., Wainwright, 2019, proof of Theorem 4.10) yields that, with probability at least $1 - \delta$,

$$\mathscr{R}(\widehat{\theta}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + \mathbb{E}\Big[\sup_{\theta \in \Theta} |\mathscr{R}(\theta) - \widehat{\mathscr{R}}_n(\theta)|\Big] + \frac{\overline{M}\sqrt{2}}{\sqrt{n}} \sqrt{\log \frac{1}{\delta}}.$$

Denote $C = 2MK_f R_\Theta \exp(2K_f R_\Theta)$. Then we show that $\mathscr{R}$ and $\widehat{\mathscr{R}}_n$ are CK_ℓ-Lipschitz with respect to $(\theta, \|\cdot\|_{1,\infty})$: for $\theta, \tilde{\theta} \in \Theta$,

$$\begin{aligned}|\mathscr{R}(\theta) - \mathscr{R}(\tilde{\theta})| &\leqslant \mathbb{E}\big[|\ell(F_\theta(x), y) - \ell(F_{\tilde{\theta}}(x), y)|\big] \\ &\leqslant K_\ell \mathbb{E}\big[\|F_\theta(x) - F_{\tilde{\theta}}(x)\|\big] \\ &\leqslant CK_\ell \|\theta - \tilde{\theta}\|_{1,\infty},\end{aligned}$$

according to Proposition 4.2. The proof for the empirical risk is very similar.

Let now $\varepsilon > 0$ and $\mathcal{N}(\varepsilon)$ be the covering number of Θ endowed with the $(1, \infty)$-norm. By Proposition 4.3,

$$\log \mathcal{N}(\varepsilon) \leqslant m \log\Big(\frac{16mR_\Theta}{\varepsilon}\Big) + \frac{m^2 K_\Theta \log(4)}{\varepsilon}.$$

Take $\theta^{(1)}, \dots, \theta^{(\mathcal{N}(\varepsilon))}$ the associated cover elements. Then, for any $\theta \in \Theta$, denoting $\theta^{(i)}$ the cover element at distance at most ε from θ,

$$\begin{aligned}|\mathscr{R}(\theta) - \widehat{\mathscr{R}}_n(\theta)| &\leqslant |\mathscr{R}(\theta) - \mathscr{R}(\theta^{(i)})| + |\mathscr{R}(\theta^{(i)}) - \widehat{\mathscr{R}}_n(\theta^{(i)})| + |\widehat{\mathscr{R}}_n(\theta^{(i)}) - \widehat{\mathscr{R}}_n(\theta)| \\ &\leqslant 2CK_\ell\varepsilon + \sup_{i\in\{1,\dots,\mathcal{N}(\varepsilon)\}} |\mathscr{R}(\theta^{(i)}) - \widehat{\mathscr{R}}_n(\theta^{(i)})|.\end{aligned}$$

Hence

$$\mathbb{E}\Big[\sup_{\theta\in\Theta} |\mathscr{R}(\theta) - \widehat{\mathscr{R}}_n(\theta)|\Big] \leqslant 2CK_\ell\varepsilon + \mathbb{E}\Big[\sup_{i\in\{1,\dots,\mathcal{N}(\varepsilon)\}} |\mathscr{R}(\theta^{(i)}) - \widehat{\mathscr{R}}_n(\theta^{(i)})|\Big].$$

Since $\widehat{\mathscr{R}}_n(\theta)$ is the average of n independent random variables, which are each almost surely bounded by $\overline{M}$, it is $\overline{M}/\sqrt{n}$ sub-Gaussian, hence we have the classical inequality on the expectation of the maximum of sub-Gaussian random variables (see, e.g., Rigollet and Hütter, 2017, Theorem 1.14)

$$\mathbb{E}\Big[\sup_{i\in\{1,\dots,\mathcal{N}(\varepsilon)\}} |\mathscr{R}(\theta^{(i)}) - \widehat{\mathscr{R}}_n(\theta^{(i)})|\Big] \leqslant \overline{M}\sqrt{\frac{2\log(2\mathcal{N}(\varepsilon))}{n}}.$$

The remainder of the proof consists in computations to put the result in the required format. More precisely, we have

$$\begin{aligned}\mathbb{E}\Big[\sup_{\theta\in\Theta} |\mathscr{R}(\theta) - \widehat{\mathscr{R}}_n(\theta)|\Big] &\leqslant 2CK_\ell\varepsilon + \overline{M}\sqrt{\frac{2\log(2\mathcal{N}(\varepsilon))}{n}} \\ &\leqslant 2CK_\ell\varepsilon + \overline{M}\sqrt{\frac{2\log(2) + 2m\log\left(\frac{16mR_\Theta}{\varepsilon}\right) + \frac{2m^2K_\Theta}{\varepsilon}\log(4)}{n}} \\ &\leqslant 2CK_\ell\varepsilon + \overline{M}\sqrt{\frac{2(m+1)\log\left(\frac{16mR_\Theta}{\varepsilon}\right) + \frac{2m^2K_\Theta}{\varepsilon}\log(4)}{n}}.\end{aligned}$$

The third step is valid if $\frac{16mR_\Theta}{\varepsilon} \geqslant 2$. We will shortly take ε to be equal to $\frac{1}{\sqrt{n}}$, thus this condition holds true under the assumption from the Theorem that $mR_\Theta\sqrt{n} \geq 3$. Hence we obtain

$$\mathscr{R}(\widehat{\theta}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + 2CK_\ell\varepsilon + \overline{M}\sqrt{\frac{2(m+1)\log\left(\frac{16mR_\Theta}{\varepsilon}\right) + \frac{2m^2K_\Theta}{\varepsilon}\log(4)}{n}} + \frac{\overline{M}\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}. \quad (4.16)$$

Now denote $\tilde{B} = 2\overline{M}K_f\exp(K_fR_\Theta)$. Then $CK_\ell \leqslant \tilde{B}$ and $2\overline{M} \leqslant \tilde{B}$. Taking $\varepsilon = \frac{1}{\sqrt{n}}$, we obtain

$$\begin{aligned}\mathscr{R}(\widehat{\theta}_n) &\leqslant \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + \frac{2\tilde{B}}{\sqrt{n}} + \frac{\tilde{B}}{2}\sqrt{\frac{2(m+1)\log(16mR_\Theta\sqrt{n})}{n} + \frac{2m^2K_\Theta\log(4)}{\sqrt{n}}} + \frac{\tilde{B}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}} \\ &\leqslant \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + \frac{2\tilde{B}}{\sqrt{n}} + \frac{\tilde{B}}{2}\sqrt{\frac{2(m+1)\log(16mR_\Theta\sqrt{n})}{n}} + \frac{\tilde{B}}{2}\frac{m\sqrt{2K_\Theta\log(4)}}{n^{1/4}} + \frac{\tilde{B}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}} \\ &\leqslant \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + \frac{3\tilde{B}}{2}\sqrt{\frac{2(m+1)\log(16mR_\Theta\sqrt{n})}{n}} + \tilde{B}\frac{m\sqrt{K_\Theta}}{n^{1/4}} + \frac{\tilde{B}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}},\end{aligned}$$

since $2 \leq 2\sqrt{\log(2)} \leq \sqrt{2(m+1)\log(16mR_\Theta\sqrt{n})}$ since $16mR_\Theta\sqrt{n} \geq 2$ by the Theorem's assumptions, and $\sqrt{2\log(4)} \leqslant 2$. We finally obtain that

$$\mathscr{R}(\widehat{\theta}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + 3\tilde{B}\sqrt{\frac{(m+1)\log(mR_\Theta n)}{n}} + \tilde{B}\frac{m\sqrt{K_\Theta}}{n^{1/4}} + \frac{\tilde{B}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}},$$

by noting that $n \geq 9\max(m^{-2}R_\Theta^{-2}, 1)$ implies that

$$\log(16mR_\Theta\sqrt{n}) \leq 2\log(mR_\Theta n).$$

The result unfolds since the constant B in the Theorem is equal to $3\tilde{B}$.

4.A.5 Proof of Corollary 4.5

The corollary is an immediate consequence of Theorem 4.4. To obtain the result, note that $m = d^2$, thus in particular $\sqrt{m+1} = \sqrt{d^2+1} \leq d+1$, and besides $\log(R_{\mathcal{W}} d^2 n) \leq 2\log(R_{\mathcal{W}} dn)$ since $R_{\mathcal{W}} n \leq R_{\mathcal{W}}^2 n^2$ by assumption on n.

4.A.6 Proof of Proposition 4.6

For $x \in \mathcal{X}$, let $(H_k)_{0 \leq k \leq L}$ be the values of the layers defined by the recurrence (4.10) with the weights $\mathbf{W}$ and the input $H_0 = x$. We denote by $\|\cdot\|$ the ℓ_2-norm for vectors and the spectral norm for matrices. Then, for $k \in \{0, \dots, L-1\}$, we have

$$\|H_{k+1}\| \leq \|H_k\| + \frac{1}{L}\|W_k \sigma(H_k)\| \leqslant \|H_k\| + \frac{1}{L}\|W_k\| \, \|\sigma(H_k)\| \leqslant \Big(1 + \frac{K_\sigma R_{\mathcal{W}}}{L}\Big)\|H_k\|,$$

where the last inequality uses that the spectral norm of a matrix is upper-bounded by its $(1,1)$-norm and that $\sigma(0) = 0$. As a consequence, for any $k \in \{0, \dots, L\}$,

$$\|H_k\| \leq \Big(1 + \frac{K_\sigma R_{\mathcal{W}}}{L}\Big)^k \|H_0\| \leq \exp(K_\sigma R_{\mathcal{W}}) R_{\mathcal{X}} =: C,$$

yielding the first claim of the Proposition.

Now, let $\tilde{H}$ be the values of the layers (4.10) with another parameter $\tilde{\mathbf{W}}$ and with the same input $\tilde{H}_0 = x$. Then, for any $k \in \{0, \dots, L-1\}$,

$$H_{k+1} - \tilde{H}_{k+1} = H_k - \tilde{H}_k + \frac{1}{L}(W_k \sigma(H_k) - \tilde{W}_k \sigma(\tilde{H}_k)).$$

Hence, using again that the spectral norm of a matrix is upper-bounded by its $(1,1)$-norm and that $\sigma(0) = 0$,

$$\begin{aligned}
\|H_{k+1} - \tilde{H}_{k+1}\| &\leq \|H_k - \tilde{H}_k\| + \frac{1}{L}\|W_k(\sigma(H_k) - \sigma(\tilde{H}_k))\| + \frac{1}{L}\|(W_k - \tilde{W}_k)\sigma(\tilde{H}_k)\| \\
&\leqslant \Big(1 + K_\sigma \frac{R_{\mathcal{W}}}{L}\Big)\|H_k - \tilde{H}_k\| + \frac{K_\sigma}{L}\|W_k - \tilde{W}_k\| \, \|\tilde{H}_k\| \\
&\leqslant \Big(1 + K_\sigma \frac{R_{\mathcal{W}}}{L}\Big)\|H_k - \tilde{H}_k\| + \frac{C K_\sigma}{L}\|W_k - \tilde{W}_k\|.
\end{aligned}$$

Then, dividing by $(1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^{k+1}$ and using the method of differences, we obtain that

$$\begin{aligned}
\frac{\|H_k - \tilde{H}_k\|}{(1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^k} &\leq \|H_0 - \tilde{H}_0\| + \frac{C K_\sigma}{L} \sum_{j=0}^{k-1} \frac{\|W_j - \tilde{W}_j\|}{(1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^{j+1}} \\
&\leq \frac{C K_\sigma}{L}\|\mathbf{W} - \tilde{\mathbf{W}}\|_{1,1,\infty} \sum_{j=0}^{k-1} \frac{1}{(1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^{j+1}}.
\end{aligned}$$

Finally note that

$$\begin{aligned}
\sum_{j=0}^{k-1} \frac{(1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^k}{(1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^{j+1}} &= \sum_{j=0}^{k-1} (1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^j \\
&= \frac{L}{K_\sigma R_{\mathcal{W}}}\Big((1 + K_\sigma \frac{R_{\mathcal{W}}}{L})^k - 1\Big) \\
&\leq \frac{L}{K_\sigma R_{\mathcal{W}}}(\exp(K_\sigma R_{\mathcal{W}}) - 1).
\end{aligned}$$

We conclude that

$$\|H_k - \tilde{H}_k\| \leq \frac{C}{R_{\mathcal{W}}}(\exp(K_\sigma R_{\mathcal{W}}) - 1)\|\mathbf{W} - \tilde{\mathbf{W}}\|_{1,1,\infty} \leq \frac{R_{\mathcal{X}}}{R_{\mathcal{W}}} \exp(2K_\sigma R_{\mathcal{W}})\|\mathbf{W} - \tilde{\mathbf{W}}\|_{1,1,\infty}.$$

4.A.7 Proof of Proposition 4.7

For two integers a and b, denote respectively $a//b$ and $a\%b$ the quotient and the remainder of the Euclidean division of a by b. Then, for $\mathbf{W} \in \mathbb{R}^{L\times d\times d}$, let $\phi(\mathbf{W}) : [0,1] \to \mathbb{R}^{d^2}$ the piecewise affine function defined as follows: $\phi(\mathbf{W})$ is affine on every interval $\left[\frac{k}{L}, \frac{k+1}{L}\right]$ for $k \in \{0,\dots,L-1\}$; for $k \in \{1,\dots,L\}$ and $i \in \{1,\dots,d^2\}$,

$$\phi(\mathbf{W})_i\Big(\frac{k}{L}\Big) = \mathbf{W}_{\frac{k}{L},(i//d)+1,(i\%d)+1}\,,$$

and $\phi(\mathbf{W})_i(0) = \phi(\mathbf{W})_i(1/L)$. Then $\phi(\mathbf{W})$ satisfies two properties. First, it is a linear function of $\mathbf{W}$. Second, for $\mathbf{W} \in \mathbb{R}^{L\times d\times d}$,

$$\|\phi(\mathbf{W})\|_{1,\infty} = \|\mathbf{W}\|_{1,1,\infty},$$

because, for $x \in [0,1]$, $\phi(\mathbf{W})(x)$ is a convex combination of two vectors that are bounded in ℓ_1-norm by $\|\mathbf{W}\|_{1,1,\infty}$, so it is itself bounded in ℓ_1-norm by $\|\mathbf{W}\|_{1,1,\infty}$, implying that $\|\phi(\mathbf{W})\|_{1,\infty} \leq \|\mathbf{W}\|_{1,1,\infty}$. Reciprocally,

$$\|\phi(\mathbf{W})\|_{1,\infty} = \sup_{0\leqslant t\leqslant 1} \|\phi(\mathbf{W})(x)\|_1 \geqslant \sup_{1\leq k\leq L} \Big\|\phi(\mathbf{W})\Big(\frac{k}{L}\Big)\Big\|_1 = \|\mathbf{W}\|_{1,1,\infty}.$$

Now, take $\mathbf{W} \in \mathcal{W}$. The second property of ϕ implies that $\|\phi(\mathbf{W})\|_{1,\infty} \leq R_{\mathcal{W}}$. Moreover, each coordinate of $\phi(\mathbf{W})$ is $K_{\mathcal{W}}$-Lipschitz, since the slope of each piece of $\phi(\mathbf{W})_i$ is at most $K_{\mathcal{W}}$. As a consequence, $\phi(\mathbf{W})$ belongs to

$$\Theta_{\mathcal{W}} = \{\theta : [0,1] \to \mathbb{R}^{d^2},\ \|\theta\|_{1,\infty} \leqslant R_{\mathcal{W}} \text{ and } \theta_i \text{ is } K_{\mathcal{W}}\text{-Lipschitz for } i \in \{1,\dots,d^2\}\}.$$

Therefore $\phi(\mathcal{W})$ is a subset of $\Theta_{\mathcal{W}}$, thus its covering number is less than the one of $\Theta_{\mathcal{W}}$. Moreover, ϕ is clearly injective, thus we can define ϕ^{-1} on its image. Consider an ε-cover $(\theta_1,\dots,\theta_N)$ of $(\phi(\mathcal{W}), \|\cdot\|_{1,\infty})$. Let us show that $(\phi^{-1}(\theta_1),\dots,\phi^{-1}(\theta_N))$ is an ε-cover of $(\mathcal{W}, \|\cdot\|_{1,1,\infty})$: take $\mathbf{W} \in \mathcal{W}$ and consider θ_i a cover member at distance less than ε from $\phi(\mathbf{W})$. Then

$$\|\mathbf{W} - \phi^{-1}(\theta_i)\|_{1,1,\infty} = \|\phi(\mathbf{W} - \phi^{-1}(\theta_i))\|_{1,\infty} = \|\phi(\mathbf{W}) - \theta_i\|_{1,\infty} \leq \varepsilon,$$

where the second equality holds by linearity of ϕ. Therefore, the covering number of $(\mathcal{W}, \|\cdot\|_{1,1,\infty})$ is upper bounded by the one of $(\phi(\mathcal{W}), \|\cdot\|_{1,\infty})$, which itself is upper bounded by the one of $(\Theta_{\mathcal{W}}, \|\cdot\|_{1,\infty})$, yielding the result by Proposition 4.3.

4.A.8 Proof of Theorem 4.8

The proof structure is the same as the one of Theorem 4.4, but some constants change. Similarly to (4.16), we obtain that, if $\frac{16d^2R_{\mathcal{W}}}{\varepsilon} \geqslant 2$ (which holds true for $\varepsilon = 1/\sqrt{n}$ and under the assumption of the Theorem),

$$\mathscr{R}(\widehat{\mathbf{W}}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\mathbf{W}}_n) + 2CK_\ell\varepsilon + \overline{M}\sqrt{\frac{2(d^2+1)\log\left(\frac{16d^2R_{\mathcal{W}}}{\varepsilon}\right) + \frac{2d^4K_{\mathcal{W}}}{\varepsilon}\log(4)}{n}} + \frac{\overline{M}\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}},$$

with

$$\overline{M} = K_\ell(R_{\mathcal{X}}\exp(K_\sigma R_{\mathcal{W}}) + R_{\mathcal{Y}})$$

and

$$C = \frac{R_{\mathcal{X}}}{R_{\mathcal{W}}}\exp(2K_\sigma R_{\mathcal{W}}).$$

Finally denote

$$\tilde{B} = 2\overline{M} \max\Big(\frac{\exp(K_\sigma R_{\mathcal{W}})}{R_{\mathcal{W}}}, 1\Big).$$

Then $CK_\ell \leqslant \tilde{B}$ and $2\overline{M} \leqslant \tilde{B}$. Taking $\varepsilon = \frac{1}{\sqrt{n}}$, we obtain as in the proof of Theorem 4.4 that

$$\mathscr{R}(\widehat{\mathbf{W}}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\mathbf{W}}_n) + 3\tilde{B}\sqrt{\frac{(d^2+1)\log(d^2 R_{\mathcal{W}} n)}{n}} + \tilde{B}\frac{d^2\sqrt{K_{\mathcal{W}}}}{n^{1/4}} + \frac{\tilde{B}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}$$

for $n \geqslant 9R_{\mathcal{W}}^{-1}\max(d^{-4}R_{\mathcal{W}}^{-1}, 1)$. Thus

$$\mathscr{R}(\widehat{\mathbf{W}}_n) \leqslant \widehat{\mathscr{R}}_n(\widehat{\mathbf{W}}_n) + 3\sqrt{2}\tilde{B}(d+1)\sqrt{\frac{\log(dR_{\mathcal{W}} n)}{n}} + \tilde{B}\frac{d^2\sqrt{K_{\mathcal{W}}}}{n^{1/4}} + \frac{\tilde{B}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}},$$

since $\sqrt{d^2+1} \leq d+1$ and $R_{\mathcal{W}} n \leq R_{\mathcal{W}}^2 n^2$ by assumption on n. The result unfolds since the constant B in the Theorem is equal to $3\sqrt{2}\tilde{B}$.

4.A.9 Proof of Corollary 4.9

Let

$$A(\mathbf{W}) = \Big(\prod_{k=1}^{L}\Big\|I + \frac{1}{L}W_k\Big\|\Big)\Big(\sum_{k=1}^{L}\frac{\|W_k^T\|_{2,1}^{2/3}}{L^{2/3}\|I+\frac{1}{L}W_k\|^{2/3}}\Big)^{3/2},$$

where $\|\cdot\|_{2,1}$ denotes the $(2,1)$-norm defined as the ℓ_1-norm of the ℓ_2-norms of the columns, and I is the identity matrix (and we recall that $\|\cdot\|$ denotes the spectral norm). We apply Theorem 1.1 from Bartlett et al. (2017) by taking as reference matrices the identity matrix. The theorem shows that, under the assumptions of the corollary,

$$\mathbb{P}\Big(\operatorname*{argmax}_{1\leq j\leq d} F_{\mathbf{W}}(x)_j \neq y\Big) \leq \widehat{\mathscr{R}}_n(\mathbf{W}) + C\frac{R_{\mathcal{X}} A(\mathbf{W})\log(d)}{\gamma\sqrt{n}} + \frac{C}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}},$$

where, as in the corollary, $\widehat{\mathscr{R}}_n(\mathbf{W}) \leq n^{-1}\sum_{i=1}^n \mathbf{1}_{F_{\mathbf{W}}(x_i)_{y_i} \leq \gamma + \max_{j\neq y_i} f(x_i)_j}$ and C is a universal constant. Let us upper bound $A(\mathbf{W})$ to conclude. On the one hand, we have

$$\begin{aligned}
\prod_{k=1}^{L}\Big\|I + \frac{1}{L}W_k\Big\| &\leq \prod_{k=1}^{L}\Big(\|I\| + \frac{1}{L}\|W_k\|\Big) \\
&\leq \prod_{k=1}^{L}\Big(1 + \frac{1}{L}\|W_k\|_{1,1}\Big) \\
&\leq \prod_{k=1}^{L}\Big(1 + \frac{1}{L}R_{\mathcal{W}}\Big) \\
&\leq \exp(R_{\mathcal{W}})
\end{aligned}$$

On the other hand, for any $k \in \{1, \ldots, L\}$,

$$\|W_k^T\|_{2,1} \leq \|W_k^T\|_{1,1} \leq R_{\mathcal{W}},$$

while

$$\|I + \frac{1}{L}W_k\| \geq 1 - \frac{1}{L}\|W_k\| \geq 1 - \frac{R_{\mathcal{W}}}{L} \geq \frac{1}{2},$$

under the assumption that $L \geq R_{\mathcal{W}}$. All in all, we obtain that

$$A(\mathbf{W}) \leq \exp(R_{\mathcal{W}})\big(2^{2/3}L^{1/3}R_{\mathcal{W}}^{2/3}\big)^{3/2} = 2R_{\mathcal{W}}\exp(R_{\mathcal{W}})\sqrt{L},$$

which yields the result.

4.B Experimental details

Our code is available at `https://github.com/PierreMarion23/generalization-ode-resnets`.

We use the following model, corresponding to model (4.10) with additional projections at the beginning and at the end:

$$\begin{aligned} H_0 &= Ax \\ H_{k+1} &= H_k + \frac{1}{L} W_{k+1} \sigma(H_k), \quad 0 \leqslant k \leqslant L-1 \\ F_{\mathbf{W}}(x) &= BH_L, \end{aligned}$$

where $x \in \mathbb{R}^{768}$ is a vectorized MNIST image, $A \in \mathbb{R}^{d \times 768}$, and $B \in \mathbb{R}^{10 \times d}$. Table 4.1 gives the value of the hyperparameters.

Name	Value
d	30
L	1000
σ	ReLU

Table 4.1: Values of the model hyperparameters.

We use the initialization scheme outlined in Section 4.4.1: we initialize, for $k \in \{1, \dots, L\}$ and $i, j \in \{1, \dots, d\}$,

$$\mathbf{W}_{k,i,j} = \frac{1}{\sqrt{d}} f_{i,j}\Big(\frac{k}{L}\Big),$$

where $f_{i,j}$ are independent Gaussian processes with an RBF kernel (with bandwidth equal to 0.1). We refer to 2 and to Sander et al. (2022b) for further discussion on this initialization scheme. However, A and B are initialized with a more usual scheme, namely with i.i.d. $\mathcal{N}(0, 1/c)$ random variables, where c denotes the number of columns of A (resp. B).

In Figure 4.1a, we repeat training 10 times independently. Each time, we perform 30 epochs, and compute after each epoch both the Lipschitz constant of the weights and the generalization gap. This gives 300 pairs (Lipschitz constant, generalization gap), which each corresponds to one dot in the figure. Furthermore, we report results for two setups: when A and B are trained or when they are fixed random matrices.

In Figure 4.1b, A and B are not trained. The reason is to assess the effect of the penalization on $\mathbf{W}$ for a fixed scale of A and B. If we allow A and B to vary, then it is possible that the effect of the penalization might be neutralized by a scale increase of A and B during training.

For all experiments, we use the standard MNIST data split (60k training samples and 10k testing samples). We train using the cross entropy loss, mini-batches of size 128, and the optimizer Adam (Kingma and Ba, 2015) with default parameters and a learning rate of 0.02.

We use PyTorch (Paszke et al., 2019) and PyTorch Lightning for our experiments.

The code takes about 60 hours to run on a standard laptop (no GPU).

5 Framing RNN as a kernel method: a neural ODE approach

Building on the interpretation of a recurrent neural network (RNN) as a continuous-time neural differential equation, we show, under appropriate conditions, that the solution of an RNN can be viewed as a linear function of a specific feature set of the input sequence, known as the signature. This connection allows us to frame an RNN as a kernel method in a suitable reproducing kernel Hilbert space. As a consequence, we obtain theoretical guarantees on generalization and stability for a large class of recurrent networks. Our results are illustrated on simulated datasets.

Contents

5.1 Introduction

Recurrent neural networks (RNN) are among the most successful methods for modeling sequential data. They have achieved state-of-the-art results in difficult problems such as natural language processing (e.g., Mikolov et al., 2010; Collobert et al., 2011) or speech recognition (e.g., Hinton et al., 2012a; Graves et al., 2013). This class of neural networks has a natural interpretation in terms of (discretization of) ordinary differential equations (ODE), which casts them in the field of neural ODE (Chen et al., 2018a). This observation has led to the development of continuous-depth models for handling irregularly-sampled time-series data, including the ODE-RNN model (Rubanova et al., 2019), GRU-ODE-Bayes (De Brouwer et al., 2019), or neural CDE models (Kidger et al., 2020; Morrill et al., 2021). In addition, the time-continuous interpretation of RNN allows to leverage the rich theory of differential equations to develop new recurrent architectures (Chang et al., 2019; Herrera et al., 2020; Erichson et al., 2021), which are better at learning long-term dependencies.

On the other hand, the development of kernel methods for deep learning offers theoretical insights on the functions learned by the networks (Cho and Saul, 2009; Belkin et al., 2018; Jacot et al., 2018). Here, the general principle consists of defining a reproducing kernel Hilbert space (RKHS)—that is, a function class $\mathscr{H}$—, which is rich enough to describe the architectures of networks. A good example is the construction of Bietti and Mairal (2017, 2019), who exhibit an RKHS for convolutional neural networks. This kernel perspective has several advantages. First, by separating the representation of the data from the learning process, it allows studying invariances of the representations learned by the network. Next, by reducing the learning problem to a linear one in $\mathscr{H}$, generalization bounds can be more easily obtained. Finally, the Hilbert structure of $\mathscr{H}$ provides a natural metric on neural networks, which can be used for example for regularization (Bietti et al., 2019).

Contributions. By taking advantage of the neural ODE paradigm for RNN, we show that RNN are, in the continuous-time limit, linear predictors over a specific space associated with the signature of the input sequence (Levin et al., 2013). The signature transform, first defined by Chen (1958) and central in rough path theory (Lyons et al., 2007; Friz and Victoir, 2010), summarizes sequential inputs by a graded feature set of their iterated integrals. Its natural environment is a tensor space that can be endowed with an RKHS structure (Király and Oberhauser, 2019). We exhibit general conditions under which classical recurrent architectures such as feedforward RNN, Gated Recurrent Units (GRU, Cho et al., 2014), or Long Short-Term Memory networks (LSTM, Hochreiter and Schmidhuber, 1997), can be framed as a kernel method in this RKHS. This enables us to provide generalization bounds for RNN as well as stability guarantees via regularization. The theory is illustrated with some experimental results.

Related works. The neural ODE paradigm was first formulated by Chen et al. (2018a) for residual neural networks. It was then extended to RNN in several articles, with a focus on handling irregularly sampled data (Rubanova et al., 2019; Kidger et al., 2020) and learning long-term dependencies (Chang et al., 2019). The signature transform has recently received the attention of the machine learning community (Levin et al., 2013; Kidger et al., 2019; Liao et al., 2019; Toth and Oberhauser, 2020; Fermanian, 2021) and, combined with deep neural networks, has achieved state-of-the-art performance for several applications (Yang et al., 2016, 2022; Perez Arribas, 2018; Wang et al., 2019; Morrill et al., 2020). Király and Oberhauser (2019) use the signature transform to define kernels for sequential data and develop fast computational methods. The connection between continuous-time RNN and signatures has been pointed out by Lim (2021) for a specific model of stochastic RNN. Deriving generalization bounds for RNN is an

active research area (Zhang et al., 2018; Akpinar et al., 2019; Tu et al., 2019). By leveraging the theory of differential equations, our approach encompasses a large class of RNN models, ranging from feedforward RNN to LSTM. This is in contrast with most existing generalization bounds, which are architecture-dependent. Close to our point of view is the work of Bietti and Mairal (2017) for convolutional neural networks.

Mathematical context. We place ourselves in a supervised learning setting. The input data is a sample of n i.i.d. vector-valued sequences $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$, where $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_T^{(i)}) \in (\mathbb{R}^d)^T$, $T \geq 1$. The outputs of the learning problem can be either labels (classification setting) or sequences (sequence-to-sequence setting). Even if we only observe discrete sequences, each $\mathbf{x}^{(i)}$ is mathematically considered as a regular discretization of a continuous-time process $X^{(i)} \in BV^c([0,1], \mathbb{R}^d)$, where $BV^c([0,1], \mathbb{R}^d)$ is the space of continuous functions from $[0,1]$ to $\mathbb{R}^d$ of finite total variation. Informally, the total variation of a process corresponds to its length. Formally, for any $[s,t] \subset [0,1]$, the total variation of a process $X \in BV^c([0,1], \mathbb{R}^d)$ on $[s,t]$ is defined by

$$\|X\|_{TV;[s,t]} = \sup_{(t_0,\dots,t_k) \in D_{s,t}} \sum_{j=1}^{k} \|X_{t_j} - X_{t_{j-1}}\|,$$

where $D_{s,t}$ denotes the set of all finite partitions of $[s,t]$ and $\|\cdot\|$ the Euclidean norm. We therefore have that $x_j^{(i)} = X_{j/T}^{(i)}$, $1 \leq j \leq T$, where $X_t^{(i)} := X^{(i)}(t)$. We make two assumptions on the processes $X^{(i)}$. First, they all begin at zero, and second, their total variations are bounded by $L \in (0,1)$. These assumptions are not too restrictive, since they amount to data translation and normalization, common in practice. Accordingly, we denote by $\mathscr{X}$ the subset of $BV^c([0,1], \mathbb{R}^d)$ defined by

$$\mathscr{X} = \left\{ X \in BV^c([0,1], \mathbb{R}^d) \,|\, X_0 = 0 \quad \text{and} \quad \|X\|_{TV;[0,1]} \leq L \right\}$$

and assume therefore that $X^{(1)}, \dots, X^{(n)}$ are i.i.d. according to some $X \in \mathscr{X}$. The norm on all spaces $\mathbb{R}^m$, $m \geq 1$, is always the Euclidean one. Observe that assuming that $X \in \mathscr{X}$ implies that, for any $t \in [0,1]$, $\|X_t\| = \|X_t - X_0\| \leq \|X\|_{TV;[0,1]} \leq L < 1$.

Recurrent neural networks. Classical RNN are defined by a sequence of hidden states $h_1, \dots, h_T$ that all belong to $\mathbb{R}^e$, where, for $\mathbf{x} = (x_1, \dots, x_T)$ a generic data sample,

$$h_0 = 0 \quad \text{and} \quad h_{j+1} = f(h_j, x_{j+1}) \quad \text{for } 0 \leq j \leq T-1.$$

At each time step $1 \leq j \leq T$, the output of the network is $z_j = \psi(h_j)$, where ψ is a linear function. In this chapter, we rather consider the following residual version, which is a natural adaptation of classical RNN in the neural ODE framework (see, e.g., Yue et al., 2018):

$$h_0 = 0 \quad \text{and} \quad h_{j+1} = h_j + \frac{1}{T} f(h_j, x_{j+1}) \quad \text{for } 0 \leq j \leq T-1. \tag{5.1}$$

The simplest choice for the function f is the feedforward model, say f_{RNN}, defined by

$$f_{\text{RNN}}(h, x) = \sigma(Uh + Vx + b), \tag{5.2}$$

where σ is an activation function, $U \in \mathbb{R}^{e \times e}$ and $V \in \mathbb{R}^{e \times d}$ are weight matrices, and $b \in \mathbb{R}^e$ is the bias. The function f_{RNN}, equipped with a smooth activation σ (such as the logistic or hyperbolic tangent functions), will be our leading example throughout the chapter. However, the GRU and LSTM models can also be rewritten under the form (5.1), as shown in Appendix 5.A.1. Thus, model (5.1) is flexible enough to encompass most recurrent networks used in practice.

Overview. Section 5.2 is devoted to framing RNN as linear functions in a suitable RKHS. We start by embedding iteration (5.1) into a continuous-time model, which takes the form of a controlled differential equation (CDE). This allows, after introducing the signature transform, to define the appropriate RKHS, and, in turn, to show that model (5.1) boils down, in the continuous-time limit, to a linear problem on the signature. This framework is used in Section 5.3 to derive generalization bounds and stability guarantees. We provide some experiments in Section 5.4 before discussing our results in Section 5.5. All proofs are postponed to the end of the chapter. In Section 5.A, we present some useful additional definitions and lemmas. The proofs are given in Section 5.B, and rely on some algebra rules over tensor spaces that are given in Section 5.C. Finally, the experimental details are presented in Section 5.D.

5.2 Framing RNN as a kernel method

Roadmap. First, we quantify the difference between the discrete recurrent network (5.1) and its continuous-time counterpart (Proposition 5.1). Then, we rewrite the corresponding ODE as a CDE (Proposition 5.2). Under appropriate conditions, Proposition 5.9 shows that the solution of this equation is a linear function of the signature of the driving process. Importantly, these assumptions are valid for a feedforward RNN, as stated by Proposition 5.10. We conclude in Theorem 5.11.

5.2.1 From discrete to continuous time

Recall that $h_0, \dots, h_T$ denote the hidden states of the RNN (5.1), and let $H : [0,1] \to \mathbb{R}^e$ be the solution of the ODE

$$dH_t = f(H_t, X_t)dt, \quad H_0 = h_0. \tag{5.3}$$

By bounding the difference between $H_{j/T}$ and h_j, the following proposition shows how to pass from discrete to continuous time, provided f satisfies the following assumption:

(A_1) The function f is Lipschitz continuous in h and x, with Lipschitz constants K_h and K_x. We let $K_f = \max(K_h, K_x)$.

Proposition 5.1. *Assume that* (A_1) *is verified. Then there exists a unique solution* H *to* (5.3) *and, for any* $0 \leq j \leq T$,

$$\|H_{j/T} - h_j\| \leq \frac{c_1}{T},$$

where $c_1 = K_f e^{K_f}\big(L + \sup_{\|h\|\leq M, \|x\|\leq L} \|f(h,x)\| e^{K_f}\big)$ *and* $M = \sup_{\|x\|\leq L} \|f(h_0, x)\| e^{K_f}$. *Moreover, for any* $t \in [0,1]$, $\|H_t\| \leq M$.

Proof. See Section 5.B.1. □

Then, following Kidger et al. (2020), we show that the ODE (5.3) can be rewritten under the form of a CDE. At the cost of increasing the dimension of the hidden state from e to $e + d$, this allows us to reframe model (5.3) as a linear model in dX, in the sense that X has been moved 'outside' of f.

Proposition 5.2. *Assume that* (A_1) *is verified. Let* $H : [0,1] \to \mathbb{R}^e$ *be the solution of* (5.3), *and let* $\bar{X} : [0,1] \to \mathbb{R}^{d+1}$ *be the time-augmented process* $\bar{X}_t = (X_t^\top, \frac{1-L}{2}t)^\top$. *Then there exists a tensor field* $\mathbf{F} : \mathbb{R}^{\bar{e}} \to \mathbb{R}^{\bar{e} \times \bar{d}}$, $\bar{e} = e + d$, $\bar{d} = d + 1$, *such that if* $\bar{H} : [0,1] \to \mathbb{R}^{\bar{e}}$ *is the solution of the CDE*

$$d\bar{H}_t = \mathbf{F}(\bar{H}_t)d\bar{X}_t, \quad \bar{H}_0 = (H_0^\top, X_0^\top)^\top, \tag{5.4}$$

then its first e *coordinates are equal to* H.

Proof. See Section 5.B.2. □

We introduce in the proposition the time-augmented process $\bar{X}$ with an additional coordinate corresponding to time. Note that this coordinate is rescaled by $\frac{1-L}{2}$ to ensure that the total variation of $\bar{X}$ is less than 1, since $L < 1$. Equation (5.4) can be better understood by the following equivalent integral equation:

$$\bar{H}_t = \bar{H}_0 + \int_0^t \mathbf{F}(\bar{H}_u) d\bar{X}_u,$$

where the integral should be understood as Riemann-Stieljes integral (Friz and Victoir, 2010, Section I.2). Thus, the output of the RNN can be approximated by the solution of the CDE (5.4), and, according to Proposition 5.1, the approximation error is $\mathscr{O}(1/T)$.

Example 5.3. *Consider f_{RNN} as in* (5.2)*. If σ is Lipschitz continuous with constant K_σ, then, for any $h_1, h_2 \in \mathbb{R}^e$, $x_1, x_2 \in \mathbb{R}^d$,*

$$\begin{aligned}\|f_{\mathrm{RNN}}(h_1, x_1) - f_{\mathrm{RNN}}(h_2, x_1)\| &= \|\sigma(Uh_1 + Vx_1 + b) - \sigma(Uh_2 + Vx_1 + b)\| \\ &\leq K_\sigma \|U\|_{\mathrm{op}} \|h_1 - h_2\|,\end{aligned}$$

where $\|\cdot\|_{\mathrm{op}}$ denotes the operator norm—see Appendix 5.A.3. Similarly, $\|f(h_1, x_1) - f(h_1, x_2)\| \leq K_\sigma \|V\|_{\mathrm{op}} \|x_1 - x_2\|$. Thus, assumption (A_1) is satisfied. The tensor field $\mathbf{F}_{\mathrm{RNN}}$ of Proposition 5.2 corresponding to this network is defined for any $\bar{h} \in \mathbb{R}^{\bar{e}}$ by

$$\mathbf{F}_{\mathrm{RNN}}(\bar{h}) = \begin{pmatrix} 0_{e\times d} & \frac{2}{1-L}\sigma(W\bar{h} + b) \\ I_{d\times d} & 0_{d\times 1} \end{pmatrix}, \quad \text{where} \quad W = \begin{pmatrix} U & V \end{pmatrix} \in \mathbb{R}^{e\times \bar{e}}. \tag{5.5}$$

5.2.2 The signature

An essential ingredient towards our construction is the signature of a continuous-time process, which we briefly present here. We refer to Chevyrev and Kormilitzin (2016) for a gentle introduction and to Lyons et al. (2007); Levin et al. (2013) for details.

Tensor Hilbert spaces. We denote by $(\mathbb{R}^d)^{\otimes k}$ the kth tensor power of $\mathbb{R}^d$ with itself, which is a Hilbert space of dimension d^k. The key space to define the signature and, in turn, our RKHS, consists in infinite square-summable sequences of tensors of increasing order:

$$\mathscr{T} = \Big\{ a = (a_0, \dots, a_k, \dots) \,\Big|\, a_k \in (\mathbb{R}^d)^{\otimes k}, \sum_{k=0}^{\infty} \|a_k\|^2_{(\mathbb{R}^d)^{\otimes k}} < \infty \Big\}, \tag{5.6}$$

where the norm is associated to the entrywise scalar product $\langle \cdot, \cdot \rangle_{(\mathbb{R}^d)^{\otimes k}}$. Endowed with the scalar product $\langle a, b \rangle_{\mathscr{T}} := \sum_{k=0}^{\infty} \langle a_k, b_k \rangle_{(\mathbb{R}^d)^{\otimes k}}$, $\mathscr{T}$ is a Hilbert space. We refer to Appendix 5.A.4 for more precise explanations.

Definition 5.4. *Let $X \in BV^c([0,1], \mathbb{R}^d)$. For any $t \in [0,1]$, the signature of X on $[0,t]$ is defined by $S_{[0,t]}(X) = (1, \mathbb{X}^1_{[0,t]}, \dots, \mathbb{X}^k_{[0,t]}, \dots)$, where, for each $k \geq 1$,*

$$\mathbb{X}^k_{[0,t]} = k! \int\limits_{0\leq u_1 < \cdots < u_k \leq t} \cdots \int dX_{u_1} \otimes \cdots \otimes dX_{u_k} \in (\mathbb{R}^d)^{\otimes k}.$$

Although this definition is technical, the signature should simply be thought of as a feature map that embeds a bounded variation process into an infinite-dimensional tensor space. The signature has several good properties that make it a relevant tool for machine learning (e.g., Levin et al., 2013; Chevyrev and Kormilitzin, 2016; Fermanian, 2021). In particular, under certain assumptions, $S(X)$ characterizes X up to translations and reparameterizations, and has good approximation properties. We also highlight that fast libraries exist for computing the signature (Reizenstein and Graham, 2020; Kidger and Lyons, 2021).

The expert reader is warned that this definition differs from the usual one by the normalization of $\mathbb{X}^k_{[0,t]}$ by $k!$, which is more adapted to our context. In the sequel, for any index $(i_1, \dots, i_k) \in \{1, \dots, d\}^k$, $S^{(i_1,\dots,i_k)}_{[0,t]}(X)$ denotes the term associated with the coordinates $(i_1, \dots, i_k)$ of $\mathbb{X}^k_{[0,t]}$. When the signature is taken on the whole interval $[0, 1]$, we simply write $S(X)$, $S^{(i_1,\dots,i_k)}(X)$, and $\mathbb{X}^k$.

Example 5.5. *Let X be the d-dimensional linear path defined by $X_t = (a_1 + b_1 t, \dots, a_d + b_d t)^\top$, $a_i, b_i \in \mathbb{R}$. Then $S^{(i_1,\dots,i_k)}(X) = b_{i_1} \cdots b_{i_k}$ and $\mathbb{X}^k = b^{\otimes k}$.*

The next proposition, which ensures that $S_{[0,t]}(\bar{X}) \in \mathscr{T}$, is an important step.

Proposition 5.6. *Let $X \in \mathscr{X}$ and $\bar{X}_t = (X_t^\top, \frac{1-L}{2}t)^\top$ as in Proposition 5.2. Then, for any $t \in [0, 1]$, $\|S_{[0,t]}(\bar{X})\|_{\mathscr{T}} \leq 2(1 - L)^{-1}$.*

Proof. See Section 5.B.3. □

The signature kernel. By taking advantage of the structure of Hilbert space of $\mathscr{T}$, it is natural to introduce the following kernel:

$$\begin{aligned} K : \mathscr{X} \times \mathscr{X} &\to \mathbb{R} \\ (X, Y) &\mapsto \langle S(\bar{X}), S(\bar{Y}) \rangle_{\mathscr{T}}, \end{aligned}$$

which is well-defined according to Proposition 5.6. We refer to Király and Oberhauser (2019) for a general presentation of kernel methods with signatures and to Salvi et al. (2021) for a kernel trick. The RKHS associated with K is the space of functions

$$\mathscr{H} = \big\{ \xi_\alpha : \mathscr{X} \to \mathbb{R} \,|\, \xi_\alpha(X) = \langle \alpha, S(\bar{X}) \rangle_{\mathscr{T}}, \alpha \in \mathscr{T} \big\}, \tag{5.7}$$

with scalar product $\langle \xi_\alpha, \xi_\beta \rangle_{\mathscr{H}} = \langle \alpha, \beta \rangle_{\mathscr{T}}$ (see, e.g., Schölkopf and Smola, 2002).

5.2.3 From the CDE to the signature kernel

An important property of signatures is that the solution of the CDE (5.4) can be written, under certain assumptions, as a linear function of the signature of the driving process X. This operation can be thought of as a Taylor expansion for CDE. More precisely, let us rewrite (5.4) as

$$dH_t = \mathbf{F}(H_t) dX_t = \sum_{i=1}^{d} F^i(H_t) dX_t^i, \tag{5.8}$$

where $X_t = (X_t^1, \dots, X_t^d)^\top$, $\mathbf{F} : \mathbb{R}^e \to \mathbb{R}^{e \times d}$, and $F^i : \mathbb{R}^e \to \mathbb{R}^e$ are the columns of $\mathbf{F}$—to avoid heavy notation, we momentarily write e, d, H, and X instead of $\bar{e}$, $\bar{d}$, $\bar{H}$, and $\bar{X}$. Throughout, the bold notation is used to distinguish tensor fields and vector fields. We recall that a vector field $F : \mathbb{R}^e \to \mathbb{R}^e$ or a tensor field $\mathbf{F} : \mathbb{R}^e \to \mathbb{R}^{e \times d}$ are said to be smooth if each of their coordinates is $\mathscr{C}^\infty$.

Definition 5.7. *Let $F, G : \mathbb{R}^e \to \mathbb{R}^e$ be smooth vector fields and denote by $J(\cdot)$ the Jacobian matrix. Their differential product is the smooth vector field $F \star G : \mathbb{R}^e \to \mathbb{R}^e$ defined, for any $h \in \mathbb{R}^e$, by*

$$(F \star G)(h) = \sum_{j=1}^{e} \frac{\partial G}{\partial h_j}(h) F_j(h) = J(G)(h)F(h).$$

In differential geometry, $F \star G$ is simply denoted by FG. Since the $\star$ operation is not associative, we take the convention that it is evaluated from right to left, i.e., $F^1 \star F^2 \star F^3 := F^1 \star (F^2 \star F^3)$.

Taylor expansion. Let H be the solution of (5.8), where $\mathbf{F}$ is assumed to be smooth. We now show that H can be written as a linear function of the signature of X, which is the crucial step to embed the RNN in the RKHS $\mathscr{H}$. The step-N Taylor expansion of H (Friz and Victoir, 2008) is defined by

$$H_t^N = H_0 + \sum_{k=1}^{N} \frac{1}{k!} \sum_{1 \le i_1, \dots, i_k \le d} S_{[0,t]}^{(i_1,\dots,i_k)}(X) F^{i_1} \star \cdots \star F^{i_k}(H_0).$$

Throughout, we let

$$\Lambda_k(\mathbf{F}) = \sup_{\|h\| \le M, 1 \le i_1, \dots, i_k \le d} \|F^{i_1} \star \cdots \star F^{i_k}(h)\|.$$

Example 5.8. *Let $\mathbf{F} = \mathbf{F}_{\mathrm{RNN}}$ defined by (5.5) with an identity activation. Then, for any $\bar{h} \in \mathbb{R}^{\bar{e}}$, $1 \le i \le d+1$, $F_{\mathrm{RNN}}^i(\bar{h}) = W_i \bar{h} + b_i$, where b_i is the $(i+d)$th vector of the canonical basis of $\mathbb{R}^{\bar{e}}$, and*

$$W_i = 0_{\bar{e} \times \bar{e}}, \quad W_{d+1} = \begin{pmatrix} \frac{2}{1-L} W \\ 0_{d \times \bar{e}} \end{pmatrix}, \quad \text{and} \quad b_{d+1} = \begin{pmatrix} \frac{2}{1-L} b \\ 0_d \end{pmatrix}.$$

The vector fields F_{RNN}^i are then affine, $J(F_{\mathrm{RNN}}^i) = W_i$, and the iterated star products have a simple expression: for any $1 \le i_1, \dots, i_k \le d$, $F_{\mathrm{RNN}}^{i_1} \star \cdots \star F_{\mathrm{RNN}}^{i_k}(\bar{h}) = W_{i_k} \cdots W_{i_2}(W_{i_1} \bar{h} + b_{i_1})$.

The next proposition shows that the step-N Taylor expansion H^N is a good approximation of H.

Proposition 5.9. *Assume that the tensor field $\mathbf{F}$ is smooth. Then, for any $t \in [0,1]$,*

$$\|H_t - H_t^N\| \le \frac{d^{N+1}}{(N+1)!} \Lambda_{N+1}(\mathbf{F}). \tag{5.9}$$

Proof. See Section 5.B.4. □

Thus, provided that $\Lambda_N(\mathbf{F})$ is not too large, the right-hand side of (5.9) converges to zero, hence

$$H_t = H_0 + \sum_{k=1}^{\infty} \frac{1}{k!} \sum_{1 \le i_1, \dots, i_k \le d} S_{[0,t]}^{(i_1,\dots,i_k)}(X) F^{i_1} \star \cdots \star F^{i_k}(H_0). \tag{5.10}$$

We conclude from the above representation that the solution H of (5.8) is in fact a linear function of the signature of X. A natural concern is to know whether the upper bound of Proposition 5.9 vanishes with N for standard architectures. This property is encapsulated in the following more general assumption:

$$(A_2) \quad \text{The tensor field } \mathbf{F} \text{ is smooth and } \sum_{k=0}^{\infty} \Big(\frac{d^k}{k!} \Lambda_k(\mathbf{F}) \Big)^2 < \infty.$$

Clearly, if (A_2) is verified, then the right-hand side of (5.9) converges to 0. The next proposition states formally the conditions under which (A_2) is verified for $\mathbf{F}_{\mathrm{RNN}}$. It is further illustrated in Figure 5.1, which shows that the convergence is fast with two common activation functions. We let $\|\sigma\|_\infty = \sup_{\|h\|\leq M, \|x\|\leq L} \|\sigma(Uh+Vx+b)\|$ and $\|\sigma^{(k)}\|_\infty = \sup_{\|h\|\leq M, \|x\|\leq L} \|\sigma^{(k)}(Uh+Vx+b)\|$, where $\sigma^{(k)}$ is the derivative of order k of σ.

Proposition 5.10. *Let $\mathbf{F}_{RNN}$ be defined by (5.5). If σ is the identity function, then (A_2) is satisfied. In the general case, (A_2) holds if σ is smooth and there exists $a > 0$ such that, for any $k \geq 0$,*

$$\|\sigma^{(k)}\|_\infty \leq a^{k+1}k! \quad \text{and} \quad \|W\|_F < \frac{1-L}{8a^2 d}, \tag{5.11}$$

where $\|\cdot\|_F$ is the Frobenius norm. Moreover, $\Lambda_N(\mathbf{F}_{RNN}) \leq \sqrt{2}a\Big(\frac{8a^2\|W\|_F}{1-L}\Big)^{N-1} N!$.

Proof. See Section 5.B.5. □

The proof of Proposition 5.10, based on the manipulation of higher-order derivatives of tensor fields, is highly non-trivial. We highlight that the conditions on σ are mild and verified for common smooth activations. For example, they are verified for the logistic function (with $a = 2$) and for the hyperbolic tangent function (with $a = 4$)—see Appendix 5.A.5. The second inequality of (5.11) puts a constraint on the norm of the weights, and can be regarded as a radius of convergence for the Taylor expansion.

Putting everything together. We now have all the elements at hand to embed the RNN into the RKHS $\mathscr{H}$. To fix the idea, we assume in this paragraph that we are in a ± 1 classification setting. In other words, given an input sequence $\mathbf{x}$, we are interested in the final output $z_T = \psi(h_T) \in \mathbb{R}$, where h_T is the solution of (5.1). The predicted class is $2 \cdot \mathbf{1}(z_T > 0) - 1$.

By Propositions 5.1 and 5.2, z_T is approximated by the first e coordinates of the solution of the CDE (5.4), which outputs a $\mathbb{R}^{e+d}$-valued process $\bar{H}$. According to Proposition 5.9, $\bar{H}$ is a linear function of the signature of the time-augmented process $\bar{X}$. Thus, on top of $\bar{H}$, it remains to successively apply the projection Proj on the e first coordinates followed by the linear function ψ to obtain an element of the RKHS $\mathscr{H}$. This mechanism is summarized in the following theorem.

Theorem 5.11. *Assume that (A_1) and (A_2) are verified. Then there exists a function $\xi_\alpha \in \mathscr{H}$ such that*

$$|z_T - \xi_\alpha(X)| \leq \|\psi\|_{\mathrm{op}} \frac{c_1}{T}, \tag{5.12}$$

where $\xi_\alpha(X) = \langle \alpha, S(\bar{X})\rangle_{\mathscr{T}}$ and $\bar{X}_t = (X_t^\top, \frac{1-L}{2}t)^\top$. We have $\alpha = (\alpha_k)_{k=0}^\infty$, where each $\alpha_k \in (\mathbb{R}^d)^{\otimes k}$ is defined by

$$\alpha_k^{(i_1,\dots,i_k)} = \frac{1}{k!}\psi \circ \mathrm{Proj}\big(F^{i_1} \star \cdots \star F^{i_k}(\bar{H}_0)\big).$$

Moreover, $\|\alpha\|_{\mathscr{T}}^2 \leq \|\psi\|_{\mathrm{op}}^2 \sum_{k=0}^\infty \Big(\frac{d^k}{k!}\Lambda_k(\mathbf{F})\Big)^2$.

Proof. See Section 5.B.6. □

We conclude that in the continuous-time limit, the output of the network can be interpreted as a scalar product between the signature of the (time-augmented) process $\bar{X}$ and an element of $\mathscr{T}$. This interpretation is important for at least two reasons: (i) it facilitates the analysis of generalization of RNN by leveraging the theory of kernel methods, and (ii) it provides new insights on regularization strategies to make RNN more robust. These points will be explored in

the next section. Finally, we stress that the approach works for a large class of RNN, such as GRU and LSTM. The derivation of conditions (A_1) and (A_2) beyond the feedforward RNN is left for future work.

5.3 Generalization and regularization

5.3.1 Generalization bounds

Learning procedure. A first consequence of framing an RNN as a kernel method is that it gives natural generalization bounds under mild assumptions. In the learning setup, we are given an i.i.d. sample $\mathscr{D}_n$ of n random pairs of observations $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \in (\mathbb{R}^d)^T \times \mathscr{Y}$, where $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_T^{(i)})$. We distinguish the binary classification problem, where $\mathscr{Y} = \{-1, 1\}$, from the sequential prediction problem, where $\mathscr{Y} = (\mathbb{R}^p)^T$ and $\mathbf{y}^{(i)} = (y_1^{(i)}, \dots, y_T^{(i)})$. The RNN is assumed to be parameterized by $\theta \in \Theta \subset \mathbb{R}^q$, where Θ is a compact set. To clarify the notation, we use a θ subscript whenever a quantity depends on θ (e.g., f_θ for f, etc.). In line with Section 5.2, it is assumed that the tensor field $\mathbf{F}_\theta$ associated with f_θ satisfies (A_1) and (A_2), keeping in mind that Proposition 5.10 guarantees that these requirements are fulfilled by a feedforward recurrent network with a smooth activation function.

Let $g_\theta : (\mathbb{R}^d)^T \to \mathscr{Y}$ denote the output of the recurrent network. The parameter θ is fitted by empirical risk minimization using a loss function $\ell : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}^+$. The theoretical and empirical risks are respectively defined, for any $\theta \in \Theta$, by

$$\mathscr{R}(\theta) = \mathbb{E}[\ell(\mathbf{y}, g_\theta(\mathbf{x}))] \quad \text{and} \quad \widehat{\mathscr{R}}_n(\theta) = \frac{1}{n} \sum_{i=1}^n \ell\big(\mathbf{y}^{(i)}, g_\theta(\mathbf{x}^{(i)})\big),$$

where the expectation $\mathbb{E}$ is evaluated with respect to the distribution of the generic random pair $(\mathbf{x}, \mathbf{y})$. We let $\widehat{\theta}_n \in \operatorname{argmin}_{\theta \in \Theta} \widehat{\mathscr{R}}_n(\theta)$ and aim at upper bounding $\mathbb{P}(\mathbf{y} \neq g_{\widehat{\theta}_n}(\mathbf{x}))$ in the classification regime (Theorem 5.12) and $\mathscr{R}(\widehat{\theta}_n)$ in the sequential regime (Theorem 5.14). To reach this goal, our strategy is to approximate the RNN by its continuous version and then use the RKHS machinery of Section 5.2.

Binary classification. In this context, the network outputs a real number $g_\theta(\mathbf{x}) = \psi(h_T) \in \mathbb{R}$ and the predicted class is $2 \cdot \mathbf{1}(g_\theta(\mathbf{x}) > 0) - 1$. The loss $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ is assumed to satisfy the assumptions of Bartlett and Mendelson (2002, Theorem 7), that is, for any $y \in \{-1, 1\}$, $\ell(\mathbf{y}, g_\theta(\mathbf{x})) = \phi(\mathbf{y} g_\theta(\mathbf{x}))$, where $\phi(u) \geq \mathbf{1}(u \leq 0)$, and ϕ is Lipschitz-continuous with constant K_ℓ. For example, the logistic loss satisfies such assumptions. We let $\xi_{\alpha_\theta} \in \mathscr{H}$ be the function of Theorem 5.11 that approximates the RNN with parameter θ. Thus, $z_T \approx \xi_{\alpha_\theta}(\bar{X}) = \langle \alpha_\theta, S(\bar{X}) \rangle_{\mathscr{T}}$, up to a $\mathscr{O}(1/T)$ term.

Theorem 5.12. *Assume that for all $\theta \in \Theta$, (A_1) and (A_2) are verified. Assume, in addition, that there exists a constant $B > 0$ such that for any $\theta \in \Theta$, $\|\xi_{\alpha_\theta}\|_{\mathscr{H}} \leq B$. Then with probability at least $1 - \delta$,*

$$\mathbb{P}\big(\mathbf{y} \neq g_{\widehat{\theta}_n}(\mathbf{x}) | \mathscr{D}_n\big) \leq \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + \frac{c_2}{T} + \frac{8BK_\ell}{(1-L)\sqrt{n}} + \frac{2BK_\ell}{1-L}\sqrt{\frac{\log(1/\delta)}{2n}}, \tag{5.13}$$

where $c_2 = K_\ell \sup_\theta \Big(\|\psi\|_{\mathrm{op}} K_{f_\theta} e^{K_{f_\theta}} \big(L + \|f_\theta\|_\infty e^{K_{f_\theta}}\big) \Big)$.

Proof. See Section 5.B.7. □

Close to our result are the bounds obtained by Zhang et al. (2018), Tu et al. (2019), and Chen et al. (2020). The main difference is that the term in $1/T$ does not usually appear, since it comes from the Euler discretization error, whereas the speed in $1/\sqrt{n}$ is the same. For instance, Chen et al. (2020) show that, under some assumptions, the excess risk is of order $\sqrt{de+e^2}T^{\alpha}K_{\ell}n^{-1/2}$. We refer to Section 5.5 for further discussion on the dependency of the different bounds to the parameter T. The take-home message is that the detour by continuous-time neural ODE provides a theoretical framework adapted to RNN, at the modest price of an additional $\mathscr{O}(1/T)$ term. Moreover, we note that the bound (5.13) is 'simple' and holds under mild conditions for a large class of RNN. More precisely, for any recurrent network of the form (5.1), provided (A_1) and (A_2) are satisfied, then (5.13) is valid with constants c_2 and B depending on the architecture. Such constants are given below in the example of a feedforward RNN. We stress that Theorem 5.12 can be extended without significant effort to the multi-class classification task, with an appropriate choice of loss function.

Example 5.13. *Take a feedforward RNN with logistic activation, and* $\Theta = \{(W,b,\psi) \,|\, \|W\|_F \leq K_W < (1-L)/32d, \|b\| \leq K_b, \|\psi\|_{\text{op}} \leq K_\psi\}$. *Then, Proposition 5.10 states that* (A_2) *is satisfied and, with Theorem 5.11, ensures that*

$$\sup_{\theta\in\Theta}\|\xi_{\alpha_\theta}\|_{\mathscr{H}} \leq \frac{\sqrt{2}K_\psi(1-L)}{1-L-32dK_W} := B, \quad K_{f_\theta} = \max(\|U\|_{\text{op}}, \|V\|_{\text{op}}), \quad \textit{and} \quad \|f_\theta\|_\infty = 1.$$

Sequence-to-sequence learning. We conclude by showing how to extend both the RKHS embedding of Theorem 5.11 and the generalization bound of Theorem 5.12 to the setting of sequence-to-sequence learning. In this case, the output of the network is a sequence

$$g_\theta(\mathbf{x}) = (z_1, \ldots, z_T) \in (\mathbb{R}^p)^T.$$

An immediate extension of Theorem 5.11 ensures that there exist p elements $\alpha_{1,\theta}, \ldots, \alpha_{p,\theta} \in \mathscr{T}$ such that, for any $1 \leq j \leq T$,

$$\left\| z_j - \left(\langle \alpha_{1,\theta}, S_{[0,j/T]}(\bar{X}) \rangle_{\mathscr{T}}, \ldots, \langle \alpha_{p,\theta}, S_{[0,j/T]}(\bar{X}) \rangle_{\mathscr{T}} \right)^\top \right\| \leq \|\psi\|_{\text{op}} \frac{c_1}{T}. \tag{5.14}$$

The properties of the signature guarantee that $S_{[0,j/T]}(X) = S(\tilde{X}_{[j]})$ where $\tilde{X}_{[j]}$ is the process equal to $\bar{X}$ on $[0, j/T]$ and then constant on $[j/T, 1]$—see Appendix 5.A.6. With this trick, we have, for any $1 \leq \ell \leq p$, $\langle \alpha_{\ell,\theta}, S_{[0,j/T]}(\bar{X}) \rangle_{\mathscr{T}} = \langle \alpha_{\ell,\theta}, S(\tilde{X}_{[j]}) \rangle_{\mathscr{T}}$, so that we are back in $\mathscr{H}$. Observe that the only difference with (5.12) is that we consider vector-valued sequential outputs, which requires to introduce the process $\tilde{X}_{[j]}$, but that the rationale is exactly the same.

We let $\ell : (\mathbb{R}^p)^T \times (\mathbb{R}^p)^T \to \mathbb{R}^+$ be the L_2 distance, that is, for any $\mathbf{y} = (y_1, \ldots, y_T)$, $\mathbf{y}' = (y'_1, \ldots, y'_T)$, $\ell(\mathbf{y}, \mathbf{y}') = \frac{1}{T}\sum_{j=1}^T \|y_j - y'_j\|^2$. It is assumed that $\mathbf{y}$ takes its values in a compact subset of $\mathbb{R}^q$, i.e., there exists $K_y > 0$ such that $\|y_j\| \leq K_y$.

Theorem 5.14. *Assume that for all* $\theta \in \Theta$, (A_1) *and* (A_2) *are verified. Assume, in addition, that there exists a constant* $B > 0$ *such that for any* $1 \leq \ell \leq p$, $\theta \in \Theta$, $\|\xi_{\alpha_{\ell,\theta}}\|_{\mathscr{H}} \leq B$. *Then with probability at least* $1-\delta$,

$$\mathscr{R}(\widehat{\theta}_n) \leq \widehat{\mathscr{R}}_n(\widehat{\theta}_n) + \frac{c_3}{T} + \frac{4pc_4B(1-L)^{-1}}{\sqrt{n}} + \sqrt{\frac{2c_5\log(1/\delta)}{n}}, \tag{5.15}$$

where $c_3 = \sup_\theta\big(c_{1,\theta} + \|\psi\|_{\text{op}}\|f_\theta\|_\infty\big) + 2\sqrt{p}B(1-L)^{-1} + 2K_y$, $c_4 = B(1-L)^{-1} + K_y$, *and* $c_5 = 4pB(1-L)^{-1}c_4 + K_y^2$.

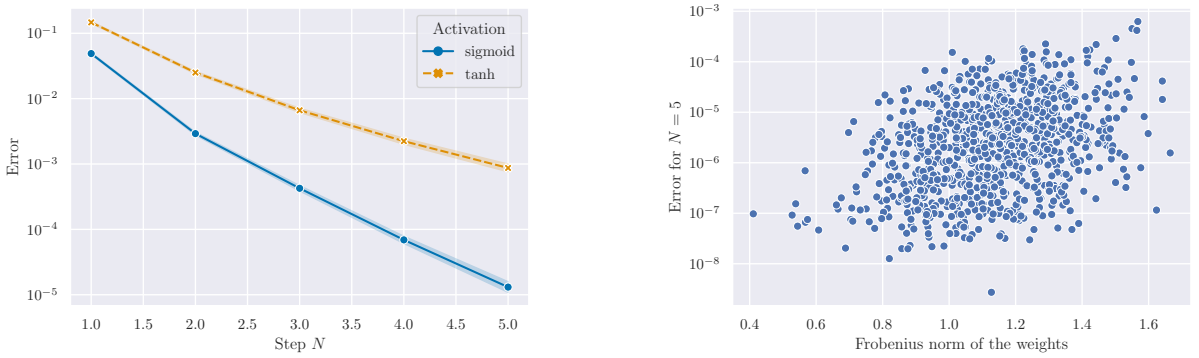
Proof. See Section 5.B.8. □

5.3.2 Regularization and stability

In addition to providing a sound theoretical framework, framing deep learning in an RKHS provides a natural norm, which can be used for regularization, as shown for example in the context of convolutional neural networks by Bietti et al. (2019). This regularization ensures stability of predictions, which is crucial in particular in a small sample regime or in the presence of adversarial examples (Gao et al., 2018; Ko et al., 2019). In our binary classification setting, for any inputs $\mathbf{x}, \mathbf{x}' \in (\mathbb{R}^d)^T$, by the Cauchy-Schwartz inequality, we have

$$\|z_T - z'_T\| \leq 2\|\psi\|_{\mathrm{op}}\|\frac{c_1}{T} + \|\xi_{\alpha_\theta}(\bar{X}) - \xi_{\alpha_\theta}(\bar{X}')\| \leq 2\|\psi\|_{\mathrm{op}}\|\frac{c_1}{T} + \|\xi_{\alpha_\theta}\|_{\mathscr{H}}\|S(\bar{X}) - S(\bar{X}')\|_{\mathscr{T}}.$$

If $\mathbf{x}$ and $\mathbf{x}'$ are close, so are their associated continuous processes X and X' (which can be approximated for example by taking a piecewise linear interpolation), and so are their signatures. The term $\|S(\bar{X}) - S(\bar{X}')\|_{\mathscr{T}}$ is therefore small (Friz and Victoir, 2010, Proposition 7.66). Therefore, when T is large, we see that the magnitude of $\|\xi_{\alpha_\theta}\|_{\mathscr{H}}$ determines how close the predictions are. A natural training strategy to ensure stable predictions, for the types of networks covered in this chapter, is then to penalize the problem by minimizing the loss $\widehat{\mathscr{R}}_n(\theta) + \lambda\|\xi_{\alpha_\theta}\|^2_{\mathscr{H}}$. From a computational point of view, it is possible to compute the norm in $\mathscr{H}$, up to a truncation at N of the Taylor expansion, which we know by Proposition 5.9 to be reasonable. It remains that computing this norm is a non-trivial task, and implementing smart surrogates is an interesting problem for the future. Note however that computing the signature of the data is not necessary for this regularization strategy.



(a) Error on a logarithmic scale as a function of N

(b) Error as a function of the norm of the weights

Figure 5.1: Approximation of the RNN ODE by the step-N Taylor expansion

5.4 Numerical illustrations

This section is here for illustration purposes. Our objective is not to achieve competitive performance, but rather to illustrate the theoretical results. We refer to Appendix 5.D for implementation details.

Convergence of the Taylor expansion towards the solution of the ODE. We illustrate Proposition 5.9 on a toy example. The process X is a 2-dimensional spiral, and we take feedforward RNN with 2 hidden units. Repeating this procedure with 10^3 uniform random weight initializations, we observe in Figure 5.1a that the signature approximation converges exponentially fast in N. As seen in Figure 5.1b, the rate of convergence depends in particular on

the norm of the weight matrices, as predicted by Proposition 5.10. However, condition (5.11) seems to be over-restrictive, since convergence happens even for weights with norm larger than the bound (we have $1/(8a^2d) \simeq 0.01$ here).

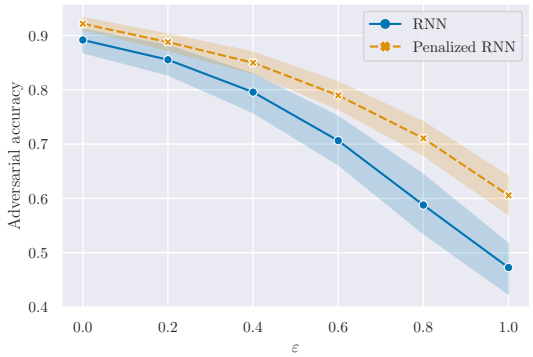


Figure 5.2: Adversarial accuracy as a function of the adversarial perturbation ε

Adversarial robustness. We illustrate the penalization proposed in Section 5.3.2 on a toy task that consists in classifying the rotation direction of 2-dimensional spirals. We take a feedforward RNN with 32 hidden units and hyperbolic tangent activation. It is trained on 50 examples, with and without penalization, for 200 epochs. Once trained, the RNN is tested on adversarial examples, generated with the projected gradient descent algorithm with Frobenius norm (Madry et al., 2018), which modifies test examples to maximize the error while staying in a ball of radius ε. We observe in Figure 5.2 that adding the penalization seems to make the network more stable.

Comparison of the trained networks. The evolution of the Frobenius norm of the weights $\|W\|_F$ and the RKHS norm $\|\xi_{\alpha_\theta}\|_{\mathscr{H}}$ during training is shown in Figure 5.3. This points out that the penalization, which forces the RNN to keep a small norm in $\mathscr{H}$, leads indeed to learning different weights than the non-penalized RNN. The results also suggest that the Frobenius and RKHS norms are decoupled, since both networks have Frobenius norms of similar magnitude but very different RKHS norms. The figures show one random run, but we observe similar qualitative behavior on others.

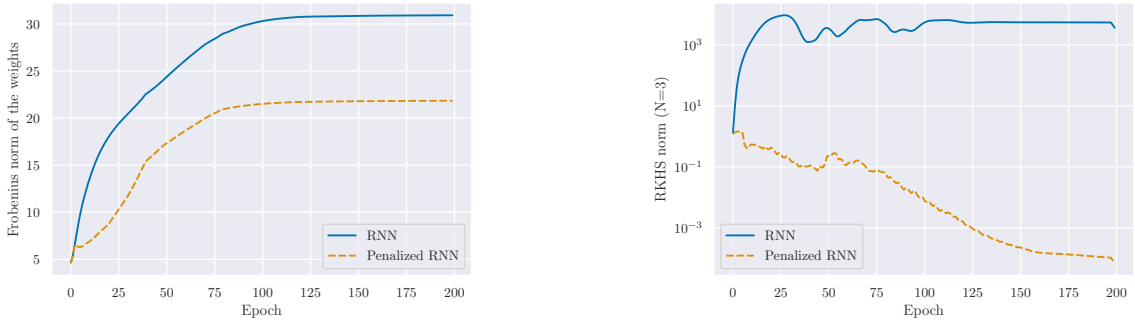


Figure 5.3: Evolution of the Frobenius norm of the weights and of the RKHS norm during training

5.5 Discussion and conclusion

Role of the discretization procedure. The starting point of the chapter was motivated by the fact that the classical residual RNN formulation coincides with an Euler discretization of the ODE (5.3). This choice of discretization translates into a $1/T$ term in Theorems 5.12 and 5.14. However, we could have considered higher-order discretization schemes, such as Runge-Kutta schemes, for which the discretization error decreases as $1/T^p$. Such schemes correspond to alternative architectures, which were already proposed by Wang and Lin (1998), among others. At the limit, we could also consider directly the continuous model (5.3), as proposed by Chen et al. (2018a), in which case the discretization error term vanishes. Of course, such an option requires to be able to sample the continuous-time data at arbitrary times.

Long-term stability. RNN are known to be poor at learning long-term dependencies (Bengio et al., 1993; Hochreiter and Schmidhuber, 1997). This is reflected in the literature by performance bounds increasing in T, which is not the case of our results (5.13) and (5.15), seemingly indicating that we fail to capture this phenomenon. This apparent paradox is related to our assumption that the total variation of X is bounded. Indeed, if a time series is observed for a long time, then its total variation may become large. In this case, it is no longer valid to assume that $\|X\|_{\mathrm{TV}}$ is bounded by L. In other words, in our context, the parameter encapsulating the notion of "long-term" is not T but the regularity of X measured by its total variation. Note that the choice of defining X on $[0,1]$ and not another interval $[0,U]$ is arbitrary and does not carry any meaning on the problem of learning long-term dependencies. A thorough analysis of these questions is an interesting research direction for future work.

Radius of convergence. The assumptions $\|X\|_{TV;[0,1]} \leq L < 1$ and $\|W\|_F \leq K_W < (1 - L)/32d$ can be seen as radii of convergence of the Taylor expansion (5.10). They allow using the Taylor approximation—which is of a local nature—to prove a global result, the RKHS embedding. In return, the condition on the Frobenius norm of the weights puts restrictions on the admissible parameters of the neural network. However, this bound can be improved, in particular by considering more exotic norms, which we did not make explicit for clarity purposes.

Conclusion. By bringing together the theory of neural ODE, the signature transform, and kernel methods, we have shown that a recurrent network can be framed in the continuous-time limit as a linear function in a well-chosen RKHS. In addition to giving theoretical insights on the function learned by the network and providing generalization guarantees, this framing suggests regularization strategies to obtain more robust RNN. We have only scratched the surface of the potentialities of leveraging this theory to practical applications, which is a subject of its own and will be tackled in future work.

5.A Some additional definitions and lemmas

5.A.1 Writing the GRU and LSTM in the neural ODE framework

GRU. Recall that the equations of a GRU take the following form: for any $1 \leq j \leq T$,

$$\begin{aligned}
r_{j+1} &= \sigma(W_r x_{j+1} + b_r + U_r h_j) \\
z_{j+1} &= \sigma(W_z x_{j+1} + b_z + U_z h_j) \\
n_{j+1} &= \tanh\big(W_n x_{j+1} + b_n + r_{j+1} * (U_n h_j + c_n)\big) \\
h_{j+1} &= (1 - z_{j+1}) * h_j + z_{j+1} * n_{j+1},
\end{aligned}$$

where σ is the logistic activation, tanh the hyperbolic tangent, $*$ the Hadamard product, r_j the reset gate vector, z_j the update gate vector, W_r, U_r, W_z, U_z, W_n, U_n weight matrices, and b_r, b_z, b_n, c_n biases. Since r_{j+1}, z_{j+1}, and n_{j+1} depend only on x_{j+1} and h_j, it is clear that these equations can be rewritten in the form

$$h_{j+1} = h_j + f(h_j, x_{j+1}).$$

We then obtain equation (5.1) by normalizing f by $1/T$.

LSTM. The LSTM networks are defined, for any $1 \leq j \leq T$, by

$$\begin{aligned}
i_{j+1} &= \sigma(W_i x_{j+1} + b_i + U_i h_j) \\
f_{j+1} &= \sigma(W_f x_{j+1} + b_f + U_f h_j) \\
g_{j+1} &= \tanh(W_g x_{j+1} + b_g + U_g h_j) \\
o_{j+1} &= \sigma(W_o x_{j+1} + b_o + U_o h_j) \\
c_{j+1} &= f_{j+1} * c_j + i_{j+1} * g_{j+1} \\
h_{j+1} &= o_{j+1} * \tanh(c_{j+1}),
\end{aligned}$$

where σ is the logistic activation, tanh the hyperbolic tangent, $*$ the Hadamard product, i_j the input gate, f_j the forget gate, g_j the cell gate, o_j the output gate, c_j the cell state, W_i, U_i, W_f, U_f, W_g, U_g W_o, U_o weight matrices, and b_i, b_f, b_g, b_o biases. Since i_{j+1}, f_{j+1}, g_{j+1}, o_{j+1} depend only on x_{j+1} and h_j, these equations can be rewritten in the form

$$\begin{aligned}
h_{j+1} &= f_1(h_j, x_{j+1}, c_{j+1}) \\
c_{j+1} &= f_2(h_j, x_{j+1}, c_j).
\end{aligned}$$

Let $\tilde{h}_j = (h_j^\top, c_j^\top)^\top$ be the hidden state defined by stacking the hidden and cell state. Then, clearly, $\tilde{h}$ follows an equation of the form

$$\tilde{h}_{j+1} = f(\tilde{h}_j, x_{j+1}).$$

We obtain (5.1) by subtracting $\tilde{h}_j$ and normalizing by $1/T$.

5.A.2 Picard-Lindelöf theorem

Consider a CDE of the form (5.8). We recall the Picard-Lindelöf theorem as given by Lyons et al. (2007, Theorem 1.3), and provide a proof for the sake of completeness.

Theorem 5.15 (Picard-Lindelöf theorem)**.** *Assume that $X \in BV^c([0,1], \mathbb{R}^d)$ and that $\mathbf{F}$ is Lipschitz-continuous with constant $K_{\mathbf{F}}$. Then, for any $H_0 \in \mathbb{R}^e$, the differential equation* (5.8) *admits a unique solution $H : [0,1] \to \mathbb{R}^e$.*

Proof. Let $\mathscr{C}([s,t]), \mathbb{R}^e)$ be the set of continuous functions from $[s,t]$ to $\mathbb{R}^e$. For any $[s,t] \subset [0,1]$, $\zeta \in \mathbb{R}^e$, let Ψ be the function

$$\begin{aligned}
\Psi : \mathscr{C}([s,t]), \mathbb{R}^e) &\to \mathscr{C}([s,t], \mathbb{R}^e) \\
Y &\mapsto \big(v \mapsto \zeta + \int_s^v \mathbf{F}(Y_u) dX_u\big).
\end{aligned}$$

For any $Y, Y' \in \mathscr{C}([s,t]), \mathbb{R}^e)$, $v \in [s,t]$,

$$\begin{aligned}
\|\Psi(Y)_v - \Psi(Y')_v\| &\leq \int_s^v \left\| \big(\mathbf{F}(Y_u) - \mathbf{F}(Y'_u)\big) dX_u \right\| \\
&\leq \int_s^v \|\mathbf{F}(Y_u) - \mathbf{F}(Y'_u)\|_{\mathrm{op}} \|dX_u\| \\
&\leq \int_s^v K_{\mathbf{F}} \|Y_u - Y'_u\| \|dX_u\| \\
&\leq K_{\mathbf{F}} \|Y - Y'\|_\infty \int_s^v \|dX_u\| \\
&\leq K_{\mathbf{F}} \|Y - Y'\|_\infty \|X\|_{TV;[s,t]}.
\end{aligned}$$

This shows that the function Ψ is Lipschitz-continuous on $\mathscr{C}([s,t]), \mathbb{R}^e)$ endowed with the supremum norm, with Lipschitz constant $K_{\mathbf{F}} \|X\|_{TV;[s,t]}$. Clearly, the function $t \mapsto \|X\|_{TV;[0,t]}$ is non-decreasing and uniformly continuous on the compact interval $[0,1]$. Therefore, for any $\varepsilon > 0$, there exists $\delta > 0$ such that

$$|t - s| < \delta \Rightarrow \big| \|X\|_{TV;[0,t]} - \|X\|_{TV;[0,s]} \big| < \varepsilon.$$

Take $\varepsilon = 1/K_{\mathbf{F}}$. Then on any interval $[s,t]$ of length smaller than δ, one has $\|X\|_{TV;[s,t]} = \|X\|_{TV;[0,t]} - \|X\|_{TV;[0,s]} < 1/K_{\mathbf{F}}$, so that the function Ψ is a contraction. By the Banach fixed-point theorem, for any initial value ζ, Ψ has a unique fixed point. Hence, there exists a solution to (5.8) on any interval of length δ with any initial condition. To obtain a solution on $[0,1]$ it is sufficient to concatenate these solutions. □

A corollary of this theorem is a Picard-Lindelöf theorem for initial value problems of the form

$$dH_t = f(H_t, X_t)dt, \quad H_0 = \zeta, \tag{5.16}$$

where $f : \mathbb{R}^e \times \mathbb{R}^d \to \mathbb{R}^e$, $\zeta \in \mathbb{R}^e$.

Corollary 5.16. *Assume that f is Lipschitz continuous in its first variable. Then, for any $\zeta \in \mathbb{R}^e$, the initial value problem* (5.16) *admits a unique solution.*

Proof. Let $f_X : (h,t) \mapsto f(h, X_t)$. Then the solution of (5.16) is solution of the differential equation

$$dH_t = f_X(H_t, t)dt.$$

Let $d = 1$, $\bar{e} = e + 1$, and $\mathbf{F}$ be the vector field defined by

$$\mathbf{F} : h \mapsto \begin{pmatrix} f_X(h^{1:e}, h^{e+1}) \\ 1 \end{pmatrix},$$

where $h^{1:e}$ denotes the projection of h on its first e coordinates. Then, since f_X is Lipschitz, so is the vector field $\mathbf{F}$. Theorem 5.15 therefore applies to the differential equation

$$dH_t = \mathbf{F}(H_t)dt, \quad H_0 = (\zeta^\top, 0)^\top.$$

Projecting this differential equation on the last coordinate gives $dH_t^{e+1} = dt$, that is, $H_t^{e+1} = t$. Projecting on the first e coordinates exactly provides equation (5.16), which therefore has a unique solution, equal to $H^{1:e}$. □

5.A.3 Operator norm

Definition 5.17. *Let $(E, \|\cdot\|_E)$ and $(F, \|\cdot\|_F)$ be two normed vector spaces and let $f \in \mathscr{L}(E,F)$, where $\mathscr{L}(E,F)$ is the space of linear functions from E to F. The operator norm of f is defined by*

$$\|f\|_{\mathrm{op}} = \sup_{u\in E, \|u\|_E=1} \|f(u)\|_F.$$

Equipped with this norm, $\mathscr{L}(E,F)$ is a normed vector space.

This definition is valid when f is represented by a matrix.

5.A.4 Tensor Hilbert space

Let us first briefly recall some elements on tensor spaces. If $e_1, \dots, e_d$ is the canonical basis of $\mathbb{R}^d$, then $(e_{i_1} \otimes \cdots \otimes e_{i_k})_{1\le i_1,\dots,i_k\le d}$ is a basis of $(\mathbb{R}^d)^{\otimes k}$. Any element $a \in (\mathbb{R}^d)^{\otimes k}$ can therefore be written as

$$a = \sum_{1\le i_1,\dots,i_k\le d} a^{(i_1,\dots,i_k)} e_{i_1} \otimes \cdots \otimes e_{i_k},$$

where $a^{(i_1,\dots,i_k)} \in \mathbb{R}$. The tensor space $(\mathbb{R}^d)^{\otimes k}$ is a Hilbert space of dimension d^k, with scalar product

$$\langle a, b\rangle_{(\mathbb{R}^d)^{\otimes k}} = \sum_{1\le i_1,\dots,i_k\le d} a^{(i_1,\dots,i_k)} b^{(i_1,\dots,i_k)}$$

and associated norm $\|\cdot\|_{(\mathbb{R}^d)^{\otimes k}}$.

We now consider the space $\mathscr{T}$ defined by (5.6). The sum, multiplication by a scalar, and scalar product on $\mathscr{T}$ are defined as follows: for any $a = (a_0, \dots, a_k, \dots) \in \mathscr{T}$, $b = (b_0, \dots, b_k, \dots) \in \mathscr{T}$, $\lambda \in \mathbb{R}$,

$$a + \lambda b = (a_0 + \lambda b_0, \dots, a_k + \lambda b_k, \dots) \quad \text{and} \quad \langle a, b\rangle_{\mathscr{T}} = \sum_{k=0}^{\infty} \langle a_k, b_k\rangle_{(\mathbb{R}^d)^{\otimes k}},$$

with the convention $(\mathbb{R}^d)^{\otimes 0} = \mathbb{R}$.

Proposition 5.18. *$(\mathscr{T}, +, \cdot, \langle\cdot,\cdot\rangle_{\mathscr{T}})$ is a Hilbert space.*

Proof. By the Cauchy-Schwartz inequality, $\langle\cdot,\cdot\rangle_{\mathscr{T}}$ is well-defined: for any $a, b \in \mathscr{T}$,

$$\begin{aligned}|\langle a, b\rangle_{\mathscr{T}}| \le \sum_{k=0}^{\infty} |\langle a_k, b_k\rangle_{(\mathbb{R}^d)^{\otimes k}}| &\le \sum_{k=0}^{\infty} \|a_k\|_{(\mathbb{R}^d)^{\otimes k}} \|b_k\|_{(\mathbb{R}^d)^{\otimes k}} \\ &\le \Big(\sum_{k=0}^{\infty} \|a_k\|^2_{(\mathbb{R}^d)^{\otimes k}}\Big)^{1/2} \Big(\sum_{k=0}^{\infty} \|b_k\|^2_{(\mathbb{R}^d)^{\otimes k}}\Big)^{1/2} < \infty.\end{aligned}$$

Moreover, $\mathscr{T}$ is a vector space: for any $a, b \in \mathscr{T}$, $\lambda \in \mathbb{R}$, since

$$a + \lambda b = (a_0 + \lambda b_0, \dots, a_k + \lambda b_k, \dots),$$

and

$$\begin{aligned}\sum_{k=0}^{\infty} \|a_k + \lambda b_k\|^2_{(\mathbb{R}^d)^{\otimes k}} &= \sum_{k=0}^{\infty} \|a_k\|^2_{(\mathbb{R}^d)^{\otimes k}} + \lambda^2 \sum_{k=0}^{\infty} \|b_k\|^2_{(\mathbb{R}^d)^{\otimes k}} \\ &\quad + 2\lambda \sum_{k=0}^{\infty} \langle a_k, b_k\rangle_{(\mathbb{R}^d)^{\otimes k}} \\ &\le \sum_{k=0}^{\infty} \|a_k\|^2_{(\mathbb{R}^d)^{\otimes k}} + \lambda^2 \sum_{k=0}^{\infty} \|b_k\|^2_{(\mathbb{R}^d)^{\otimes k}} + 2\lambda \langle a, b\rangle_{\mathscr{T}} < \infty,\end{aligned}$$

we see that $a + \lambda b \in \mathscr{T}$. The operation $\langle \cdot, \cdot \rangle_{\mathscr{T}}$ is also bilinear, symmetric, and positive definite:

$$\langle a, a \rangle_{\mathscr{T}} = 0 \Leftrightarrow \sum_{k=0}^{\infty} \|a_k\|^2_{(\mathbb{R}^d)^{\otimes k}} = 0 \Leftrightarrow \forall k \in \mathbb{N}, \|a_k\|^2_{(\mathbb{R}^d)^{\otimes k}} = 0 \Leftrightarrow \forall k \in \mathbb{N}, a_k = 0 \Leftrightarrow a = 0.$$

Therefore $\langle \cdot, \cdot \rangle_{\mathscr{T}}$ is an inner product on $\mathscr{T}$. Finally, let $(a^{(n)})_{n \in \mathbb{N}}$ be a Cauchy sequence in $\mathscr{T}$. Then, for any $n, m \geq 0$,

$$\|a^{(n)} - a^{(m)}\|^2_{\mathscr{T}} = \sum_{k=0}^{\infty} \|a_k^{(n)} - a_k^{(m)}\|^2_{(\mathbb{R}^d)^{\otimes k}},$$

so for any $k \in \mathbb{N}$, the sequence $(a_k^{(n)})_{n \in \mathbb{N}}$ is Cauchy in $(\mathbb{R}^d)^{\otimes k}$. Since $(\mathbb{R}^d)^{\otimes k}$ is a Hilbert space, $(a_k^{(n)})_{n \in \mathbb{N}}$ converges to a limit $a_k^{(\infty)} \in (\mathbb{R}^d)^{\otimes k}$. Let $a^{(\infty)} = (a_0^{(\infty)}, \dots, a_k^{(\infty)}, \dots)$. To finish the proof, we need to show that $a^{(\infty)} \in \mathscr{T}$ and that $a^{(n)}$ converges to $a^{(\infty)}$ in $\mathscr{T}$. First, note that there exists a constant $B > 0$ such that for any $n \in \mathbb{N}$,

$$\|a^{(n)}\|_{\mathscr{T}} \leq B.$$

To see this, observe that for $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for any $n \geq N$, $\|a^{(n)} - a^{(N)}\|_{\mathscr{T}} < \varepsilon$, and so $\|a^{(n)}\|_{\mathscr{T}} \leq \varepsilon + \|a^{(N)}\|_{\mathscr{T}}$. Take $B = \max(\|a^{(1)}\|_{\mathscr{T}}, \dots, \|a^{(N)}\|_{\mathscr{T}}, \varepsilon + \|a^{(N)}\|_{\mathscr{T}})$. Then, for any $K \in \mathbb{N}$,

$$\sum_{k=0}^{K} \|a_k^{(n)}\|^2_{(\mathbb{R}^d)^{\otimes k}} \leq \|a^{(n)}\|_{\mathscr{T}} \leq B.$$

Letting $K \to \infty$, we obtain that $\|a^{(\infty)}\|_{\mathscr{T}} \leq B$, and therefore $a^{(\infty)} \in \mathscr{T}$. Finally, let $\varepsilon > 0$ and let $N \in \mathbb{N}$ be such that for any $n, m \geq N$, $\|a^{(n)} - a^{(m)}\|_{\mathscr{T}} < \varepsilon$. Clearly, for any $K \in \mathbb{N}$,

$$\sum_{k=0}^{K} \|a_k^{(n)} - a_k^{(m)}\|^2_{(\mathbb{R}^d)^{\otimes k}} < \varepsilon^2.$$

Letting $m \to \infty$ leads to

$$\sum_{k=0}^{K} \|a_k^{(n)} - a_k^{(\infty)}\|^2_{(\mathbb{R}^d)^{\otimes k}} < \varepsilon^2,$$

and letting $K \to \infty$ gives

$$\|a^{(n)} - a^{(\infty)}\|_{\mathscr{T}} < \varepsilon,$$

which completes the proof. □

5.A.5 Bounding the derivatives of the logistic and hyperbolic tangent activations

Lemma 5.19. *Let σ be the logistic function defined, for any $x \in \mathbb{R}$, by $\sigma(x) = 1/(1+e^{-x})$. Then, for any $n \geq 0$,*

$$\|\sigma^{(n)}\|_{\infty} \leq 2^{n-1} n!\,.$$

Proof. For any $x \in \mathbb{R}$, one has (Minai and Williams, 1993, Theorem 2)

$$\sigma^{(n)}(x) = \sum_{k=1}^{n+1} (-1)^{k-1}(k-1)! \left\{ {n+1 \atop k} \right\} \sigma(x)^k,$$

where $\left\{ {n \atop k} \right\}$ stands for the Stirling number of the second kind (see, e.g., Riordan, 1958). Let

$$u_n = \sum_{k=1}^{n+1} (k-1)! \left\{ {n+1 \atop k} \right\}$$

for $n \geq 1$ and $u_0 = 1$. Since $0 \leq \sigma(x) \leq 1$, it is clear that $|\sigma^{(n)}(x)| \leq u_n$. Using the fact that the Stirling numbers satisfy the recurrence relation

$$\left\{ {n+1 \atop k} \right\} = k \left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\},$$

valid for all $0 \leq k \leq n$, we have

$$\begin{aligned} u_n &= \sum_{k=1}^{n} (k-1)! \left(k \left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\} \right) + n! = \sum_{k=1}^{n} k! \left\{ {n \atop k} \right\} + \sum_{k=0}^{n-1} k! \left\{ {n \atop k} \right\} + n! = 2 \sum_{k=1}^{n} k! \left\{ {n \atop k} \right\} \\ &\qquad \text{(since } \left\{ {n \atop 0} \right\} = 0\text{)} \\ &\leq 2n \sum_{k=1}^{n} (k-1)! \left\{ {n \atop k} \right\} = 2n u_{n-1}. \end{aligned}$$

Thus, by induction, $u_n \leq 2^{n-1} n!$, from which the claim follows. □

Lemma 5.20. *Let* tanh *be the hyperbolic tangent function. Then, for any* $n \geq 0$*,*

$$\|\tanh^{(n)}\|_\infty \leq 4^n n! \,.$$

Proof. Let σ be the logistic function. Straightforward calculations yield the equality, valid for any $x \in \mathbb{R}$,

$$\tanh(x) = 2\sigma(2x) - 1.$$

But, for any $n \geq 1$,

$$\tanh^{(n)}(x) = 2^{n+1} \sigma^{(n)}(2x),$$

and thus, by Lemma 5.19,

$$\|\tanh^{(n)}\|_\infty \leq 2^{n+1} \|\sigma^{(n)}\|_\infty \leq 4^n n! \,.$$

The inequality is also true for $n = 0$ since $\|\tanh\|_\infty \leq 1$. □

5.A.6 Chen's formula

First, note that it is straightforward to extend the definition of the signature to any interval $[s,t] \subset [0,1]$. The next proposition, known as Chen's formula (Lyons et al., 2007, Theorem 2.9), tells us that the signature can be computed iteratively as tensor products of signatures on subintervals.

Proposition 5.21. *Let* $X \in BV^c([s,t], \mathbb{R}^d)$ *and* $u \in (s,t)$*. Then*

$$S_{[s,t]}(X) = S_{[s,u]}(X) \otimes S_{[u,t]}(X).$$

Next, it is clear that the signature of a constant path is equal to $\mathbf{1} = (1, 0, \dots, 0, \dots)$ which is the null element in $\mathscr{T}$. Indeed, let $Y \in BV^c([s,t], \mathbb{R}^d)$ be a constant path. Then, for any $k \geq 1$,

$$\mathbb{Y}^k_{[s,t]} = k! \int_{s \leq u_1 < \cdots < u_k \leq t} \cdots \int dY_{u_1} \otimes \cdots \otimes dY_{u_k} = k! \int_{s \leq u_1 < \cdots < u_k \leq t} \cdots \int 0 \otimes \cdots \otimes 0 = 0.$$

Now let $X \in BV^c([0,1], \mathbb{R}^d)$ and consider the path $\tilde{X}_{[j]}$ equal to the time-augmented path $\bar{X}$ on $[0, j/T]$ and then constant on $[j/T, 1]$—see Figure 5.4. We have by Proposition 5.21

$$S_{[0,1]}(\tilde{X}_{[j]}) = S_{[0,j/T]}(\tilde{X}_{[j]}) \otimes S_{[j/T,1]}(\tilde{X}_{[j]}) = S_{[0,j/T]}(\bar{X}) \otimes \mathbf{1} = S_{[0,j/T]}(\bar{X}).$$

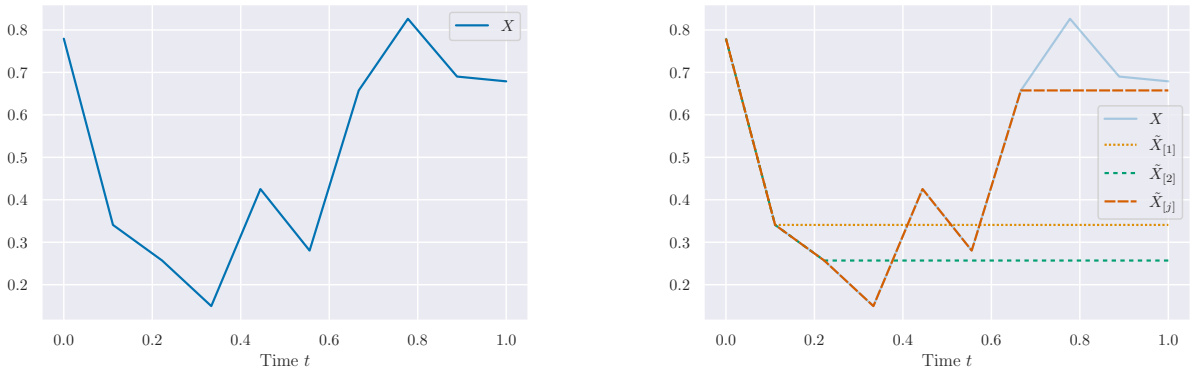


Figure 5.4: Example of a path $X \in BV^c([0,1], \mathbb{R})$ (left) and its corresponding paths $\tilde{X}_{[j]}$, plotted against time, for different values of $j \in \{1, \dots, T\}$ (right)

5.B Proofs

5.B.1 Proof of Proposition 5.1

According to Assumption (A_1), for any $h_1, h_2 \in \mathbb{R}^e, x_1, x_2 \in \mathbb{R}^d$, one has

$$\|f(h_1, x_1) - f(h_2, x_1)\| \leq K_f \|h_1 - h_2\| \quad \text{and} \quad \|f(h_1, x_1) - f(h_1, x_2)\| \leq K_f \|x_1 - x_2\|.$$

Under assumption (A_1), by Corollary 5.16, the initial value problem (5.3) admits a unique solution H. Let us first show that for any $t \in [0,1]$, H_t is bounded independently of X. For any $t \in [0,1]$,

$$\begin{aligned}
\|H_t - H_0\| = \Big\| \int_0^t f(H_u, X_u) du \Big\| &\leq \int_0^t \|f(H_u, X_u)\| du \\
&= \int_0^t \|f(H_u, X_u) - f(H_0, X_u) + f(H_0, X_u)\| du \\
&\leq \int_0^t \|f(H_u, X_u) - f(H_0, X_u)\| + \int_0^t \|f(H_0, X_u)\| du \\
&\leq K_f \int_0^t \|H_u - H_0\| du + t \sup_{\|x\| \leq L} \|f(H_0, x)\|.
\end{aligned}$$

Applying Grönwall's inequality to the function $t \mapsto \|H_t - H_0\|$ yields

$$\|H_t - H_0\| \leq t \sup_{\|x\| \leq L} \|f(H_0, x)\| \exp\Big(\int_0^t K_f du \Big) \leq \sup_{\|x\| \leq L} \|f(H_0, x)\| e^{K_f} := M.$$

Given that $H_0 = h_0 = 0$, we conclude that $\|H_t\| \leq M$.

Next, let

$$\|f\|_\infty = \sup_{\|x\|\leq L, \|h\|\leq M} f(h, x).$$

By similar arguments, for any $[s, t] \subset [0, 1]$, Grönwall's inequality applied to the function $t \mapsto \|H_t - H_s\|$ yields

$$\|H_t - H_s\| \leq (t - s)\|f\|_\infty e^{K_f}.$$

Therefore, for any partition $(t_0, \ldots, t_k)$ of $[s, t]$,

$$\sum_{i=1}^{k} \|H_{t_i} - H_{t_{i-1}}\| \leq \|f\|_\infty e^{K_f} \sum_{i=1}^{k} (t_i - t_{i-1}) \leq \|f\|_\infty e^{K_f}(t - s),$$

and, taking the supremum over all partitions of $[s, t]$, $\|H\|_{TV;[s,t]} \leq \|f\|_\infty e^{K_f}(t - s)$. In other words, H is of bounded variation on any interval $[s, t] \subset [0, 1]$. Let $(t_0, \ldots, t_T)$ denote the regular partition of $[0, 1]$ with $t_j = j/T$. For any $1 \leq j \leq T$, we have

$$\begin{aligned}
\|H_{t_j} - h_j\| &= \big\|H_{t_{j-1}} + \int_{t_{j-1}}^{t_j} f(H_u, X_u)du - h_{j-1} - \frac{1}{T} f(h_{j-1}, x_j)\big\| \\
&\leq \|H_{t_{j-1}} - h_{j-1}\| + \int_{t_{j-1}}^{t_j} \big\|f(H_u, X_u) - f(h_{j-1}, x_j)\big\| du.
\end{aligned}$$

Writing

$$\begin{aligned}
\big\|f(H_u, X_u) - f(h_{j-1}, x_j)\big\| &= \big\|f(H_u, X_u) - f(H_u, x_j) + f(H_u, x_j) - f(h_{j-1}, x_j)\big\| \\
&\leq \big\|f(H_u, X_u) - f(H_u, x_j)\big\| + \big\|f(H_u, x_j) - f(h_{j-1}, x_j)\big\| \\
&\leq K_f \big\|X_u - x_j\big\| + K_f \big\|H_u - h_{j-1}\big\|,
\end{aligned}$$

we obtain

$$\begin{aligned}
\|H_{t_j} - h_j\| &\leq \|H_{t_{j-1}} - h_{j-1}\| + K_f \int_{t_{j-1}}^{t_j} \|H_u - h_{j-1}\| du + K_f \int_{t_{j-1}}^{t_j} \|X_u - x_j\| du \\
&\leq \|H_{t_{j-1}} - h_{j-1}\| + K_f \int_{t_{j-1}}^{t_j} \big(\|H_u - H_{t_{j-1}}\| + \|H_{t_{j-1}} - h_{j-1}\|\big) du \\
&\quad + \frac{K_f}{T} \|X\|_{TV;[t_{j-1}, t_j]} \\
&\leq \big(1 + \frac{K_f}{T}\big) \|H_{t_{j-1}} - h_{j-1}\| + \frac{K_f}{T} \big(\|H\|_{TV;[t_{j-1}, t_j]} + \|X\|_{TV;[t_{j-1}, t_j]}\big).
\end{aligned}$$

By induction, we are led to

$$\begin{aligned}
\|H_{t_j} - h_j\| &\leq \frac{K_f}{T} \sum_{k=0}^{j-1} \Big(1 + \frac{K_f}{T}\Big)^k \big(\|H\|_{TV;[t_k, t_{k+1}]} + \|X\|_{TV;[t_k, t_{k+1}]}\big) \\
&\leq \frac{K_f}{T} \Big(1 + \frac{K_f}{T}\Big)^T \big(\|X\|_{TV;[0,1]} + \|H\|_{TV;[0,1]}\big) \\
&\leq \frac{K_f e^{K_f}}{T} \big(L + \|f\|_\infty e^{K_f}\big),
\end{aligned}$$

which concludes the proof.

5.B.2 Proof of Proposition 5.2

Let $\bar{h} \in \mathbb{R}^{\bar{e}}$ and let $\bar{h}^{i:j} = (\bar{h}^i, \dots, \bar{h}^j)$ be its projection on a subset of coordinates. It is sufficient to take $\mathbf{F}$ defined by

$$\mathbf{F}(\bar{h}) = \begin{pmatrix} 0_{e\times d} & \frac{2}{1-L} f(\bar{h}^{1:e}, \bar{h}^{e+1:e+d}) \\ I_{d\times d} & 0_{d\times 1} \end{pmatrix},$$

where $I_{d\times d}$ denotes the identity matrix and $0_{\times\cdot}$ the matrix full of zeros. The function $\bar{H}$ is then solution of

$$d\bar{H}_t = \begin{pmatrix} 0_{e\times d} & \frac{2}{1-L} f(\bar{H}_t^{1:e}, \bar{H}_t^{e+1:e+d}) \\ I_{d\times d} & 0_{d\times 1} \end{pmatrix} \begin{pmatrix} dX_t \\ \frac{1-L}{2} dt \end{pmatrix}.$$

Note that under assumption (A_1), the tensor field $\mathbf{F}$ satisfies the assumptions of the Picard-Lindelöf theorem (Theorem 5.15) so that $\bar{H}$ is well-defined. The projection of this equation on the last d coordinates gives

$$d\bar{H}_t^{e+1:e+d} = dX_t, \quad \bar{H}_0^{e+1:e+d} = X_0,$$

and therefore $\bar{H}_t^{e+1:e+d} = X_t$. The projection on the first e coordinates gives

$$d\bar{H}_t^{1:e} = \frac{2}{1-L} f(\bar{H}_t^{1:e}, X_t) \frac{1-L}{2} dt = f(\bar{H}_t^{1:e}, X_t) dt, \quad \bar{H}_0^{1:e} = h_0,$$

which is exactly (5.3).

5.B.3 Proof of Proposition 5.6

According to Lyons (2014, Lemma 5.1), one has

$$\|\bar{\mathbb{X}}^k_{[0,t]}\|_{(\mathbb{R}^d)^{\otimes k}} \leq \|\bar{X}\|^k_{TV;[0,t]}.$$

Let $(t_0, \dots, t_k)$ be a partition of $[0, t]$. Then

$$\begin{aligned} \sum_{j=1}^k \|\bar{X}_{t_j} - \bar{X}_{t_{j-1}}\| &= \sum_{j=1}^k \sqrt{\|X_{t_j} - X_{t_{j-1}}\|^2 + \Big(\frac{1-L}{2}\Big)^2 (t_j - t_{j-1})^2} \\ &\leq \sum_{j=1}^k \|X_{t_j} - X_{t_{j-1}}\| + \frac{1-L}{2} \sum_{j=1}^k (t_j - t_{j-1}) \\ &= \sum_{j=1}^k \|X_{t_j} - X_{t_{j-1}}\| + \frac{1-L}{2} t. \end{aligned}$$

Taking the supremum over any partition of $[0, t]$ we obtain

$$\|\bar{X}\|_{TV;[0,t]} \leq \|X\|_{TV;[0,t]} + \frac{1-L}{2} t \leq L + \frac{1-L}{2} = \frac{1+L}{2} < 1,$$

and thus $\|\bar{\mathbb{X}}^k_{[0,t]}\|_{(\mathbb{R}^d)^{\otimes k}} \leq \Big(\frac{1+L}{2}\Big)^k$. It is then clear that

$$\|S_{[0,t]}(\bar{X})\|_{\mathscr{T}} = \Big(\sum_{k=0}^\infty \|\bar{\mathbb{X}}^k_{[0,t]}\|^2_{(\mathbb{R}^d)^{\otimes k}}\Big)^{1/2} \leq \sum_{k=0}^\infty \|\bar{\mathbb{X}}^k_{[0,t]}\|_{(\mathbb{R}^d)^{\otimes k}} \leq \sum_{k=0}^\infty \Big(\frac{1+L}{2}\Big)^k = 2(1-L)^{-1}.$$

5.B.4 Proof of Proposition 5.9

We first recall the fundamental theorem of calculus for line integrals (also known as gradient theorem).

Theorem 5.22. *Let $g : \mathbb{R}^e \to \mathbb{R}$ be a continuously differentiable function, and let $\gamma : [a, b] \to \mathbb{R}^e$ be a smooth curve in $\mathbb{R}^e$. Then*

$$\int_a^b \nabla g(\gamma_t) d\gamma_t = g(\gamma_b) - g(\gamma_a),$$

where ∇g denotes the gradient of g.

The identity above immediately generalizes to a function $g : \mathbb{R}^e \to \mathbb{R}^e$:

$$\int_a^b J(g)(\gamma_t) d\gamma_t = g(\gamma_b) - g(\gamma_a),$$

where $J(g) \in \mathbb{R}^{e \times e}$ is the Jacobian matrix of g. Let us apply Theorem 5.22 to the vector field F^i between 0 and t, with $\gamma = H$. We have

$$\begin{aligned} F^i(H_t) - F^i(H_0) &= \int_0^t J(F^i)(H_u) dH_u = \int_0^t J(F^i)(H_u) \sum_{j=1}^d F^j(H_u) dX_u \\ &= \sum_{j=1}^d \int_0^t J(F^i)(H_u) F^j(H_u) dX_u = \sum_{j=1}^d \int_0^t F^j \star F^i(H_u) dX_u. \end{aligned}$$

Iterating this procedure $(N-1)$ times for the vector fields $F^1, \dots, F^d$ yields

$$\begin{aligned} H_t &= H_0 + \sum_{i=1}^d \int_0^t F^i(H_u) dX_u^i \\ &= H_0 + \sum_{i=1}^d \int_0^t F^i(H_0) dX_u^i + \sum_{i=1}^d \int_0^t \sum_{j=1}^d \int_0^u F^j \star F^i(H_v) dX_v^j dX_u^i \\ &= H_0 + \sum_{i=1}^d F^i(H_0) S_{[0,t]}^{(i)}(X) + \sum_{1 \le i,j \le d} \int_{0 \le v \le u \le t} F^j \star F^i(H_v) dX_v^j dX_u^i \\ &= \cdots \\ &= H_0 + \sum_{k=1}^N \sum_{1 \le i_1, \dots, i_k \le d} F^{i_1} \star \cdots \star F^{i_k}(H_0) \frac{1}{k!} S_{[0,t]}^{(i_1, \dots, i_k)}(X) \\ &\quad + \sum_{1 \le i_1, \dots, i_{N+1} \le d} \int_{\Delta_{N+1;[0,t]}} F^{i_1} \star \cdots \star F^{i_{N+1}}(H_{u_1}) dX_{u_1}^{i_1} \cdots dX_{u_{N+1}}^{i_{N+1}}, \end{aligned}$$

where $\Delta_{N;[0,t]} := \{(u_1, \cdots, u_N) \in [0,t]^N \,|\, 0 \le u_1 < \cdots < u_N \le t\}$ is the simplex in $[0,t]^N$. The

first $(N+1)$ terms equal H_t^N. Hence,

$$\begin{aligned}
&\|H_t - H_t^N\| \\
&\quad = \Big\| \sum_{1\le i_1,\dots,i_{N+1}\le d} \int_{\Delta_{N+1;[0,t]}} F^{i_1} \star \cdots \star F^{i_{N+1}}(H_{u_1}) dX_{u_1}^{i_1} \cdots dX_{u_{N+1}}^{i_{N+1}} \Big\| \\
&\quad \le \sum_{1\le i_1,\dots,i_{N+1}\le d} \int_{\Delta_{N+1;[0,t]}} \|F^{i_1} \star \cdots \star F^{i_{N+1}}(H_{u_1})\| |dX_{u_1}^{i_1}| \cdots |dX_{u_{N+1}}^{i_{N+1}}| \\
&\quad \le \sum_{1\le i_1,\dots,i_{N+1}\le d} \int_{\Delta_{N+1;[0,t]}} \sup_{1\le i_1,\dots,i_{N+1}\le d, \|h\|\le M} \|F^{i_1} \star \cdots \star F^{i_{N+1}}(h)\| |dX_{u_1}^{i_1}| \cdots |dX_{u_{N+1}}^{i_{N+1}}| \\
&\quad \le \Lambda_{N+1}(\mathbf{F}) \sum_{1\le i_1,\dots,i_{N+1}\le d} \int_{\Delta_{N+1;[0,t]}} |dX_{u_1}^{i_1}| \cdots |dX_{u_{N+1}}^{i_{N+1}}|.
\end{aligned}$$

Thus,

$$\begin{aligned}
\|H_t - H_t^N\| &\le \Lambda_{N+1}(\mathbf{F}) \sum_{1\le i_1,\dots,i_{N+1}\le d} \int_{\Delta_{N+1;[0,t]}} |dX_{u_1}^{i_1}| \cdots |dX_{u_{N+1}}^{i_{N+1}}| \\
&\le \Lambda_{N+1}(\mathbf{F}) \sum_{1\le i_1,\dots,i_{N+1}\le d} \int_{\Delta_{N+1;[0,t]}} \|dX_{u_1}\| \cdots \|dX_{u_{N+1}}\| \\
&= \Lambda_{N+1}(\mathbf{F}) \frac{d^{N+1}}{(N+1)!} \int_{[0,t]^{N+1}} \|dX_{u_1}\| \cdots \|dX_{u_{N+1}}\| \\
&= \Lambda_{N+1}(\mathbf{F}) \frac{d^{N+1}}{(N+1)!} \Big(\int_0^t \|dX_u\| \Big)^{N+1} \\
&= \Lambda_{N+1}(\mathbf{F}) \frac{d^{N+1}}{(N+1)!} \|X\|_{TV;[0,t]}^{N+1} \le \Lambda_{N+1}(\mathbf{F}) \frac{d^{N+1}}{(N+1)!}.
\end{aligned}$$

5.B.5 Proof of Proposition 5.10

For simplicity of notation, since the context is clear, we now use the notation $\|\cdot\|$ instead of $\|\cdot\|_{(\mathbb{R}^e)^{\otimes k}}$. According to Proposition 5.1, the solution $\bar{H}$ of (5.4) verifies $\|\bar{H}_t\| \le M + L := \bar{M}$. We therefore place ourselves in the ball $\mathscr{B}_{\bar{M}}$. Recall that for any $1 \le i_1, \dots, i_N \le d$, $\bar{h} \in \mathscr{B}_{\bar{M}}$,

$$F^{i_1} \star \cdots \star F^{i_N}(\bar{h}) = J(F^{i_2} \star \cdots \star F^{i_N})(\bar{h}) F^{i_1}(\bar{h}). \tag{5.17}$$

Linear case. We start with the proof of the linear case before moving on to the general case. When σ is chosen to be the identity function, each F_{RNN}^i is an affine vector field, in the sense that $F_{\mathrm{RNN}}^i(\bar{h}) = W_i \bar{h} + b_i$, where $W_i = 0_{\bar{e}\times\bar{e}}$, b_i is the $i+d$th vector of the canonical basis of $\mathbb{R}^{e+d}$, and

$$W_{d+1} = \begin{pmatrix} \frac{2}{1-L} W \\ 0_{d\times\bar{e}} \end{pmatrix} \quad \text{and} \quad b_{d+1} = \begin{pmatrix} \frac{2}{1-L} b \\ 0_d \end{pmatrix}.$$

Since $J(F_{\mathrm{RNN}}^i) = W_i$, we have, for any $\bar{h} \in \mathbb{R}^{e+d}$ and any $1 \le i_1, \dots, i_k \le d$,

$$F_{\mathrm{RNN}}^{i_1} \star \cdots \star F_{\mathrm{RNN}}^{i_k}(\bar{h}) = W_{i_k} \cdots W_{i_2}(W_{i_1}\bar{h} + b_{i_1}).$$

Thus, for any $\bar{h} \in \mathscr{B}_{\bar{M}}$,

$$\|F_{\mathrm{RNN}}^{i_1} \star \cdots \star F_{\mathrm{RNN}}^{i_k}(\bar{h})\| \le \|W_{i_k}\|_{\mathrm{op}} \cdots \|W_{i_2}\|_{\mathrm{op}} (\|W_{i_1}\|_{\mathrm{op}} \bar{M} + \|b_{i_1}\|).$$

For $i \neq d+1$, $\|W_{i_1}\|_{\mathrm{op}} = 0$, and so

$$\Lambda_k(\mathbf{F}_{\mathrm{RNN}}) \leq C\|W_{d+1}\|_{\mathrm{op}}^{k-1},$$

with $C = \|W_{d+1}\|_{\mathrm{op}}\bar{M} + \max(1, 2(1-L)^{-1}\|b\|)$. Therefore,

$$\sum_{k=1}^{\infty} \frac{d^k}{k!}\Lambda_k(\mathbf{F}_{\mathrm{RNN}}) \leq Cd\sum_{k=0}^{\infty}\frac{1}{k!}\big(2d(1-L)^{-1}\|W\|_{\mathrm{op}}\big)^{k-1} < \infty.$$

General case. In the general case, the proof is two-fold. First, we upper bound (5.17) by a function of the norms of higher-order Jacobians of $F^{i_1}, \dots, F^{i_N}$. We then apply this bound to the specific case $\mathbf{F} = \mathbf{F}_{\mathrm{RNN}}$. We refer to Appendix 5.C for details on higher-order derivatives in tensor spaces. Let $F : \mathbb{R}^e \to \mathbb{R}^e$ be a smooth vector field. If $F(h) = (F_1(h), \dots, F_e(h))^\top$, each of its coordinates F_i is a function from $\mathbb{R}^e$ to $\mathbb{R}$, $\mathscr{C}^\infty$ with respect to all its input variables. We define the derivative of order k of F as the tensor field

$$\begin{aligned} J^k(F) : \mathbb{R}^e &\to (\mathbb{R}^e)^{\otimes k+1} \\ h &\mapsto J^k(F)(h), \end{aligned}$$

where

$$J^k(F)(h) = \sum_{1 \leq j, i_1, \dots, i_k \leq e} \frac{\partial^k F_j(h)}{\partial h_{i_1} \dots \partial h_{i_k}} e_j \otimes e_{i_1} \otimes \cdots \otimes e_{i_k}.$$

We take the convention $J^0(F) = F$, and note that $J(F) = J^1(F)$ is the Jacobian matrix, and that $J^k(J^{k'}(F)) = J^{k+k'}(F)$.

Lemma 5.23. *Let $A^1, \dots, A^k : \mathbb{R}^e \to \mathbb{R}^e$ be smooth vector fields. Then, for any $h \in \mathbb{R}^e$*

$$\big\|A^k \star \cdots \star A^1(h)\big\| \leq \sum_{n_1 + \cdots + n_k = k-1} C(k; n_1, \dots, n_k)\|J^{n_1}(A^1)(h)\| \cdots \|J^{n_k}(A^k)(h)\|,$$

where $C(k; n_1, \dots, n_k)$ is defined by the following recurrence on k: $C(1; 0) = 1$ and for any $n_1, \dots, n_{k+1} \geq 0$,

$$\begin{aligned} C(k+1; n_1, \dots, n_{k+1}) &= \sum_{\ell=1}^{k} C(k; n_1, \dots, n_\ell - 1, \dots, n_k) && \text{if} \quad n_{k+1} = 0, \qquad (5.18) \\ C(k+1; n_1, \dots, n_{k+1}) &= 0 && \text{otherwise.} \end{aligned}$$

Proof. We refer to Appendix 5.C for the definitions of the tensor dot product $\odot$ and tensor permutations, as well as for computation rules involving these operations. We show in fact by induction a stronger result, namely that there exist tensor permutations π_p such that

$$A^k \star \cdots \star A^1(h) = \sum_{n_1 + \cdots + n_k = k-1} \sum_{1 \leq p \leq C(k; n_1, \dots, n_k)} \pi_p \Big[J^{n_1}(A^1)(h) \odot \cdots \odot J^{n_k}(A^k)(h)\Big]. \quad (5.19)$$

Note that we do not make explicit the permutations nor the axes of the tensor dot operations since we are only interested in bounding the norm of the iterated star products. Also, for simplicity, we denote all permutations by π, even though they may change from line to line.

We proceed by induction on k. For $k = 1$, the formula is clear. Assume that the formula is true at order k. Then

$$\begin{aligned}
&J(A^k \star \cdots \star A^1) \\
&\quad = \sum_{n_1+\cdots+n_k=k-1} \sum_{1\le p\le C(k;n_1,\ldots,n_k)} J\Big[\ \pi_p[\, J^{n_1}(A^1) \odot \cdots \odot J^{n_k}(A^k)\,]\ \Big] \\
&\quad = \sum_{n_1+\cdots+n_k=k-1} \sum_{1\le p\le C(k;n_1,\ldots,n_k)} \pi_p\Big[\ J[\, J^{n_1}(A^1) \odot \cdots \odot J^{n_k}(A^k)\,]\ \Big] \\
&\quad = \sum_{n_1+\cdots+n_k=k-1} \sum_{1\le p\le C(k;n_1,\ldots,n_k)} \sum_{\ell=1}^{k} \pi_p \circ \pi_\ell\Big[J^{n_1}(A^1) \odot \\
&\qquad\qquad\qquad\qquad \cdots \odot J^{n_\ell+1}(A^\ell) \odot \cdots \odot J^{n_k}(A^k)\Big].
\end{aligned}$$

In the inner sum, we introduce the change of variable $p_i = n_i$ for $i \neq \ell$ and $p_\ell = n_\ell + 1$. This yields

$$\begin{aligned}
&J(A^k \star \cdots \star A^1) \\
&\quad = \sum_{p_1+\cdots+p_k=k} \sum_{\ell=1}^{k} \sum_{1\le p\le C(k;p_1,\ldots,p_\ell-1,\ldots,p_k)} \pi_p \circ \pi_\ell\Big[\ J^{n_1}\,(A^1) \odot \\
&\qquad\qquad\qquad\qquad \cdots \odot J^{n_\ell+1}(A^\ell) \odot \cdots \odot J^{n_k}(A^k)\ \Big] \\
&\quad = \sum_{p_1+\cdots+p_{k+1}=k} \sum_{1\le q\le C(k+1;p_1,\ldots,p_{k+1})} \pi_q\Big[J^{n_1}(A^1) \odot \cdots \odot J^{p_k}(A^k)\Big],
\end{aligned}$$

where in the last sum the only non-zero term is for $p_{k+1} = 0$. To conclude the induction, it remains to note that

$$A^{k+1} \star \cdots \star A^1 = J(A^k \star \cdots \star A^1) \odot A^{k+1} = J(A^k \star \cdots \star A^1) \odot J^0(A^{k+1}).$$

Hence,

$$\begin{aligned}
&A^{k+1} \star \cdots \star A^1 \\
&\quad = \sum_{p_1+\cdots+p_{k+1}=k} \sum_{1\le q\le C(k+1;p_1,\ldots,p_{k+1})} \pi_q\left[J^{n_1}(A^1) \odot \cdots \odot J^{p_k}(A^k)\right] \odot J^{p_{k+1}}(A^{k+1}) \\
&\quad = \sum_{p_1+\cdots+p_{k+1}=k} \sum_{1\le q\le C(k+1;p_1,\ldots,p_{k+1})} \pi_q\left[J^{n_1}(A^1) \odot \cdots \odot J^{p_k}(A^k) \odot J^{p_{k+1}}(A^{k+1})\right].
\end{aligned}$$

The result is then a consequence of (5.19) and of Lemma 5.29. □

We now restrict ourselves to the case $\mathbf{F} = \mathbf{F}_{\mathrm{RNN}}$ as defined by (5.5) and give an upper bound on the higher-order derivatives of the tensor fields $F^{i_1}, \ldots, F^{i_N}$.

Lemma 5.24. *For any $i \in \{1, \ldots, d+1\}$, $\bar{h} \in \mathscr{B}_{\bar{M}}$, for any $k \geq 0$,*

$$\|J^k(F^i_{\mathrm{RNN}})(\bar{h})\| \leq \Big(\frac{2}{1-L}\|W\|_F\Big)^k \|\sigma^{(k)}\|_\infty.$$

Proof. For any $1 \leq i \leq d$, $F^i_{\mathrm{RNN}}(\bar{h})$ is constant, so $J^k(F^1_{\mathrm{RNN}}) = \cdots = J^k(F^d_{\mathrm{RNN}}) = 0$. For $i = d+1$, we have, for any $1 \leq j \leq e$,

$$\frac{\partial^k F^{d+1}_{\mathrm{RNN},j}(\bar{h})}{\partial \bar{h}_{i_1} \ldots \partial \bar{h}_{i_k}} = \Big(\frac{2}{1-L}\Big)^k W_{ji_1} \cdots W_{ji_k} \sigma^{(k)}(W_{j\cdot}\bar{h} + b),$$

where $W_{j\cdot}$ denotes the jth row of W and for $e+1 \le j \le \bar{e}$, $F_j^{d+1} = 0$. Therefore,

$$\begin{aligned}\|J^k(F_{\mathrm{RNN}}^{d+1})(\bar{h})\|^2 &\le \Big(\frac{2}{1-L}\Big)^{2k} \sum_{1\le j,i_1,\dots,i_k\le e} |W_{ji_1}\cdots W_{ji_k}\sigma^{(k)}(W_{j\cdot}\bar{h}+b)|^2 \\ &= \Big(\frac{2}{1-L}\Big)^{2k} \|\sigma^{(k)}\|_\infty^2 \sum_j \big(\sum_i |W_{ji}|^2\big)^k \\ &\le \Big(\frac{2}{1-L}\Big)^{2k} \|\sigma^{(k)}\|_\infty^2 \|W\|_F^{2k}.\end{aligned}$$

□

We are now in a position to conclude the proof using condition (5.11). By Lemma 5.23 and 5.24, for any $1 \le i_1,\dots,i_N \le d+1$,

$$\begin{aligned}&\big\|F_{\mathrm{RNN}}^{i_1} \star\cdots\star F_{\mathrm{RNN}}^{i_N}(\bar{h})\big\| \\ &\quad\le \sum_{n_1+\cdots+n_N=N-1} C(N;n_N,\dots,n_1)\|J^{n_N}(F_{\mathrm{RNN}}^{i_N})(\bar{h})\|\cdots\|J^{n_1}(F_{\mathrm{RNN}}^{i_1})(\bar{h})\| \\ &\quad\le \Big(\frac{2}{1-L}\|W\|_F\Big)^{N-1} \sum_{n_1+\cdots+n_N=N-1} C(N;n_N,\dots,n_1)a^{n_1+1}n_1!\cdots a^{n_N+1}n_N! \\ &\quad\le a\Big(\frac{2}{1-L}a^2\|W\|_F\Big)^{N-1} \sum_{n_1+\cdots+n_N=N-1} C(N;n_N,\dots,n_1)n_1!\cdots n_N!\,.\end{aligned}$$

Assume for the moment that $C(N;n_N,\dots,n_1)$ is smaller than the multinomial coefficient $\binom{N}{n_N,\dots,n_1}$. Then, using the fact that there are $\binom{n+k-1}{k-1}$ weak compositions of n in k parts and Stirling's approximation, we have

$$\begin{aligned}\Lambda_N(\mathbf{F}) &\le a\Big(\frac{2}{1-L}a^2\|W\|_F\Big)^{N-1} N!\times\ \mathrm{Card}\big(\{n_1+\cdots+n_N=N-1\}\big) \\ &\le a\Big(\frac{2}{1-L}a^2\|W\|_F\Big)^{N-1} N!\binom{2N-2}{N-1} \\ &\le \frac{a}{2}\Big(\frac{2}{1-L}a^2\|W\|_F\Big)^{N-1} N!\binom{2N}{N} \\ &\le a\frac{\sqrt{2}e}{\pi}\Big(\frac{8}{1-L}a^2\|W\|_F\Big)^{N-1}\frac{N!}{\sqrt{N}}.\end{aligned}$$

Hence, provided $\|W\|_F < {}^{(1-L)}\!/_{8a^2d}$,

$$\sum_{k=1}^\infty \frac{d^k}{k!}\Lambda_k(\mathbf{F}) \le ad\frac{\sqrt{2}e}{\pi}\sum_{k=1}^\infty \Big(\frac{8da^2\|W\|_F}{1-L}\Big)^{k-1}\frac{1}{\sqrt{k}} < \infty,$$

and (A_2) is verified.

To conclude the proof, it remains to prove the following lemma.

Lemma 5.25. *For any $k \ge 1$ and $n_1,\dots,n_k \ge 0$, $C(k;n_1,\dots,n_k) \le \binom{k-1}{n_1,\dots,n_k}$.*

Proof. The proof is done by induction, by comparing the recurrence formula (5.18) with the following recurrence formula for multinomial coefficients:

$$\binom{k}{n_1,\dots,n_{k+1}} = \sum_{\ell=1}^{k+1}\binom{k-1}{n_1,\dots,n_\ell-1,\dots,n_{k+1}}.$$

More precisely, for $k = 1$, $C(1;0) = 1 \leq \binom{0}{0} = 1$ and $C(1;1) = 0 \leq \binom{0}{1} = 0$. Assume that the formula is true at order k. Then, at order $k+1$, there are two cases. If $n_{k+1} \neq 0$, $C(k+1; n_1, \dots, n_{k+1}) = 0$, and the result is clear. On the other hand, if $n_{k+1} = 0$,

$$\begin{aligned} C(k+1; n_1, \dots, n_k, 0) &= \sum_{\ell=1}^{k} C(k; n_1, \dots, n_\ell - 1, \dots, n_k) \\ &\leq \sum_{\ell=1}^{k} \binom{k-1}{n_1, \dots, n_\ell - 1, \dots, n_k} \\ &\leq \sum_{\ell=1}^{k+1} \binom{k-1}{n_1, \dots, n_\ell - 1, \dots, n_{k+1}} \\ &\leq \binom{k}{n_1, \dots, n_{k+1}}. \end{aligned}$$

□

5.B.6 Proof of Theorem 5.11

First, Propositions 5.1 and 5.2 state that if $\bar{H}$ is the solution of (5.4) and Proj denotes the projection on the first e coordinates, then

$$\big|z_T - \psi\big(\mathrm{Proj}(\bar{H}_1)\big)\big| = \big|\psi(h_T) - \psi\big(\mathrm{Proj}(\bar{H}_1])\big)\big| \leq \|\psi\|_{\mathrm{op}} \big\|h_T - \mathrm{Proj}(\bar{H}_1)\big\| \leq \|\psi\|_{\mathrm{op}} \frac{c_1}{T}.$$

For any $1 \leq k \leq N$, we let $\mathscr{D}^k(\bar{H}_0) : (\mathbb{R}^d)^{\otimes k} \to \mathbb{R}^e$ be the linear function defined by

$$\mathscr{D}^k(\bar{H}_0)(e_{i_1} \otimes \cdots \otimes e_{i_k}) = F^{i_1} \star \cdots \star F^{i_k}(\bar{H}_0), \tag{5.20}$$

where $e_1, \dots, e_d$ denotes the canonical basis of $\mathbb{R}^{\bar{d}}$. Then, under assumptions (A_1) and (A_2), if $\bar{\mathbb{X}}^k$ denotes the signature of order k of the path $\bar{X}_t = (X_t^\top, \frac{1-L}{2}t)^\top$, according to Propositions 5.9 and 5.10,

$$\bar{H}_1 = \bar{H}_0 + \sum_{k=1}^{\infty} \frac{1}{k!} \sum_{1 \leq i_1, \dots, i_k \leq d} S_{[0,t]}^{(i_1, \dots, i_k)}(X) F^{i_1} \star \cdots \star F^{i_k}(\bar{H}_0) = \sum_{k=1}^{\infty} \frac{1}{k!} \mathscr{D}^k(\bar{H}_0)(\mathbb{X}_{[0,t]}^k),$$

and

$$\psi \circ \mathrm{Proj}(\bar{H}_1) = \psi \circ \mathrm{Proj}\Big(\sum_{k=0}^{\infty} \frac{1}{k!} \mathscr{D}^k(\bar{H}_0)(\bar{\mathbb{X}}^k) \Big) = \sum_{k=0}^{\infty} \frac{1}{k!} \psi \circ \mathrm{Proj}\big(\mathscr{D}^k(\bar{H}_0)(\bar{\mathbb{X}}^k) \big),$$

by linearity of ψ and Proj. Since the maps $\mathscr{D}^k(\bar{H}_0) : (\mathbb{R}^d)^{\otimes k} \to \mathbb{R}^e$ are linear, the above equality takes the form

$$\psi \circ \mathrm{Proj}(\bar{H}_1) = \sum_{k=0}^{\infty} \langle \alpha^k, \bar{\mathbb{X}}^k \rangle_{(\mathbb{R}^d)^{\otimes k}}, \tag{5.21}$$

where $\alpha^k \in (\mathbb{R}^d)^{\otimes k}$ is the coefficient of the linear map $\frac{1}{k!} \psi \circ \mathrm{Proj} \circ \mathscr{D}^k(\bar{H}_0)$ in the canonical basis. Let $\alpha = (\alpha^0, \dots, \alpha^k, \dots)$. Under assumption (A_2),

$$\begin{aligned} \sum_{k=0}^{\infty} \|\alpha^k\|_{(\mathbb{R}^d)^{\otimes k}}^2 &\leq \sum_{k=0}^{\infty} \sum_{1 \leq i_1, \dots, i_k \leq d} \Big(\frac{1}{k!}\Big)^2 \|\psi\|_{\mathrm{op}}^2 \|F^{i_1} \star \cdots \star F^{i_k}(\bar{H}_0)\|^2 \\ &\leq \|\psi\|_{\mathrm{op}}^2 \sum_{k=0}^{\infty} \sum_{1 \leq i_1, \dots, i_k \leq d} \Big(\frac{1}{k!}\Big)^2 \Lambda_k(\mathbf{F})^2 \\ &\leq \|\psi\|_{\mathrm{op}}^2 \sum_{k=0}^{\infty} \Big(\frac{d^k}{k!} \Lambda_k(\mathbf{F})\Big)^2 < \infty. \end{aligned}$$

This shows that $\alpha \in \mathscr{T}$, and therefore, using (5.21), we conclude

$$\|z_T - \langle \alpha, S(\bar{X})\rangle_{\mathscr{T}}\| \leq \|\psi\|_{\mathrm{op}} \frac{c_1}{T}.$$

5.B.7 Proof of Theorem 5.12

Let

$$\mathscr{G} = \Big\{ g_\theta : (\mathbb{R}^d)^T \to \mathbb{R} \,|\, g_\theta(\mathbf{x}) = z_T, \theta \in \Theta \Big\}$$

be the function class of (discrete) RNN and

$$\mathscr{S} = \Big\{ \xi_{\alpha_\theta} : \mathscr{X} \to \mathbb{R} \,|\, \xi_{\alpha_\theta}(X) = \langle \alpha_\theta, S(\bar{X})\rangle_{\mathscr{T}}, \theta \in \Theta \Big\},$$

be the class of their RKHS embeddings, where α_θ is defined by (5.21). For any $\theta \in \Theta$, we let

$$\mathscr{R}_{\mathscr{G}}(\theta) = \mathbb{E}[\ell(\mathbf{y}, g_\theta(\mathbf{x}))], \quad \text{and} \quad \mathscr{R}_{\mathscr{S}}(\theta) = \mathbb{E}[\ell(\mathbf{y}, \xi_{\alpha_\theta}(\bar{X}))],$$

and denote by $\widehat{\mathscr{R}}_{n,\mathscr{G}}$ and $\widehat{\mathscr{R}}_{n,\mathscr{S}}$ the corresponding empirical risks. We also let $\theta^*_{\mathscr{G}}$, $\theta^*_{\mathscr{S}}$, $\widehat{\theta}_{n,\mathscr{G}}$, and $\widehat{\theta}_{n,\mathscr{S}}$ be the corresponding minimizers. We have

$$\begin{aligned}
\mathbb{P}\big(\mathbf{y} \neq g_{\widehat{\theta}_{n,\mathscr{G}}}(\mathbf{x})\big) - \widehat{\mathscr{R}}_{n,\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) &\leq \mathbb{E}\big[\ell(\mathbf{y}, g_{\widehat{\theta}_{n,\mathscr{G}}}(\mathbf{x}))\big] - \widehat{\mathscr{R}}_{n,\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) \\
&= \mathscr{R}_{\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) - \widehat{\mathscr{R}}_{n,\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) \\
&= \mathscr{R}_{\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) - \mathscr{R}_{\mathscr{S}}(\widehat{\theta}_{n,\mathscr{G}}) + \mathscr{R}_{\mathscr{S}}(\widehat{\theta}_{n,\mathscr{G}}) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\widehat{\theta}_{n,\mathscr{G}}) \\
&\quad + \widehat{\mathscr{R}}_{n,\mathscr{S}}(\widehat{\theta}_{n,\mathscr{G}}) - \widehat{\mathscr{R}}_{n,\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) \\
&\leq \sup_\theta |\mathscr{R}_{\mathscr{G}}(\theta) - \mathscr{R}_{\mathscr{S}}(\theta)| + \sup_\theta |\mathscr{R}_{\mathscr{S}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)| \\
&\quad + \sup_\theta |\widehat{\mathscr{R}}_{n,\mathscr{G}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)|.
\end{aligned}$$

Using Theorem 5.11, we have

$$\begin{aligned}
\sup_\theta |\mathscr{R}_{\mathscr{G}}(\theta) - \mathscr{R}_{\mathscr{S}}(\theta)| &= \sup_\theta \big|\mathbb{E}\big[\ell(\mathbf{y}, g_\theta(\mathbf{x})) - \ell(\mathbf{y}, \xi_{\alpha_\theta}(\bar{X}))\big]\big| \\
&\leq \sup_\theta \mathbb{E}\big[|\phi(\mathbf{y} g_\theta(\mathbf{x})) - \phi(\mathbf{y}\xi_{\alpha_\theta}(\bar{X}))|\big] \\
&\leq \sup_\theta \mathbb{E}\big[K_\ell |\mathbf{y}| \times |g_\theta(\mathbf{x}) - \xi_{\alpha_\theta}(\bar{X})|\big] \\
&\leq K_\ell \sup_\theta (\|\psi\|_{\mathrm{op}} c_{1,\theta}) \frac{1}{T} := \frac{c_2}{2T},
\end{aligned}$$

where $c_{1,\theta} = K_{f_\theta} e^{K_{f_\theta}} \big(L + \|f_\theta\|_\infty e^{K_{f_\theta}}\big)$ (the infinity norm $\|f_\theta\|_\infty$ is taken on the balls $\mathscr{B}_L$ and $\mathscr{B}_M$). One proves with similar arguments that

$$\sup_\theta |\widehat{\mathscr{R}}_{n,\mathscr{G}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)| \leq \frac{c_2}{2T}.$$

Under the assumption of the theorem, there exists a ball $\mathscr{B} \subset \mathscr{H}$ of radius B such that $\mathscr{S} \subset \mathscr{B}$. This yields

$$\sup_\theta |\mathscr{R}_{\mathscr{S}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)| \leq \sup_{\alpha \in \mathscr{T}, \|\alpha\|_{\mathscr{T}} \leq B} |\mathscr{R}_{\mathscr{B}}(\alpha) - \widehat{\mathscr{R}}_{n,\mathscr{B}}(\alpha)|,$$

where

$$\mathscr{R}_{\mathscr{B}}(\alpha) = \mathbb{E}[\ell(Y, \xi_\alpha(\bar{X}))] \quad \text{and} \quad \widehat{\mathscr{R}}_{n,\mathscr{B}}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \ell(Y^{(i)}, \xi_\alpha(\bar{X}^{(i)})).$$

We now have reached a familiar situation where the supremum is over a ball in an RKHS. A slight extension of Bartlett and Mendelson (2002, Theorem 8) yields that with probability at least $1-\delta$,

$$\sup_{\alpha\in\mathscr{T}, \|\alpha\|_{\mathscr{T}}\le B} |\mathscr{R}_{\mathscr{B}}(\alpha) - \widehat{\mathscr{R}}_{n,\mathscr{B}}(\alpha)| \le 4K_\ell \mathbb{E}\mathrm{Rad}_n(\mathscr{B}) + 2BK_\ell(1-L)^{-1}\sqrt{\frac{\log(1/\delta)}{2n}},$$

where $\mathrm{Rad}_n(\mathscr{B})$ denotes the Rademacher complexity of $\mathscr{B}$. Observe that we have used the fact that the loss is bounded by $2BK_\ell(1-L)^{-1}$ since, for any $\xi_\alpha \in \mathscr{B}$, by the Cauchy-Schwartz inequality,

$$\begin{aligned}\ell(\mathbf{y}, \xi_\alpha(\bar{X})) = \phi(\mathbf{y}\langle \alpha, S(\bar{X})\rangle_{\mathscr{T}}) \le K_\ell |\mathbf{y}\langle \alpha, S(\bar{X})\rangle_{\mathscr{T}}| &\le K_\ell \|\alpha\|_{\mathscr{T}} \|S(\bar{X})\|_{\mathscr{T}} \\ &\le 2K_\ell B(1-L)^{-1}.\end{aligned}$$

Finally, the proof follows by noting that Rademacher complexity of $\mathscr{B}$ is bounded by

$$\mathrm{Rad}_n(\mathscr{B}) \le \frac{B}{n}\sqrt{\sum_{i=1}^{n} K(X^{(i)}, X^{(i)})} = \frac{B}{n}\sqrt{\sum_{i=1}^{n} \|S(\bar{X}^{(i)})\|_{\mathscr{T}}^2} \le \frac{2B(1-L)^{-1}}{\sqrt{n}}.$$

5.B.8 Proof of Theorem 5.14

Let

$$\mathscr{G} = \Big\{ g_\theta : (\mathbb{R}^d)^T \to (\mathbb{R}^p)^T \,|\, g_\theta(\mathbf{x}) = \big(z_1, \dots, z_T\big), \theta \in \Theta \Big\}$$

be the function class of discrete RNN in a sequential setting. Let

$$\mathscr{S} = \Big\{ \Gamma_\theta : \mathscr{X} \to (\mathbb{R}^p)^T \,|\, \Gamma_\theta(X) = \big(\Xi_\theta(\tilde{X}_{[1]}), \dots, \Xi_\theta(\tilde{X}_{[T]})\big) \Big\},$$

be the class of their RKHS embeddings, where $\tilde{X}_{[j]}$ is the path equal to X on $[0, j/T]$ and then constant on $[j/T, 1]$ (see Figure 5.4). For any $X \in \mathscr{X}$,

$$\Xi_\theta(a) = \begin{pmatrix} \langle \alpha_{1,\theta}, S(\bar{X})\rangle_{\mathscr{T}} \\ \vdots \\ \langle \alpha_{p,\theta}, S(\bar{X})\rangle_{\mathscr{T}} \end{pmatrix} = \begin{pmatrix} \xi_{\alpha_{1,\theta}}(X) \\ \vdots \\ \xi_{\alpha_{p,\theta}}(X) \end{pmatrix} \in \mathbb{R}^p,$$

where $(\alpha_{1,\theta}, \dots, \alpha_{p,\theta})^\top \in (\mathscr{T})^p$ are the coefficients of the linear maps $\frac{1}{k!}\psi \circ \mathrm{Proj} \circ \mathscr{D}^k(\bar{H}_0) : (\mathbb{R}^d)^{\otimes k} \to \mathbb{R}^p$, $k \ge 0$, in the canonical basis, where $\mathscr{D}^k$ is defined by (5.20).

We start the proof as in Theorem 5.12, until we obtain

$$\begin{aligned}\mathscr{R}_{\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) - \widehat{\mathscr{R}}_{n,\mathscr{G}}(\widehat{\theta}_{n,\mathscr{G}}) \le \sup_\theta |\mathscr{R}_{\mathscr{G}}(\theta) - \mathscr{R}_{\mathscr{S}}(\theta)| + \sup_\theta |\mathscr{R}_{\mathscr{S}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)| \\ + \sup_\theta |\widehat{\mathscr{R}}_{n,\mathscr{G}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)|.\end{aligned}$$

By definition of the loss, for any $\theta \in \Theta$,

$$\begin{aligned}
|\mathscr{R}_{\mathscr{G}}(\theta) - \mathscr{R}_{\mathscr{S}}(\theta)| &= \Big|\mathbb{E}\big[\ell\big(\mathbf{y}, g_\theta(\mathbf{x})\big) - \ell\big(\mathbf{y}, \Gamma_\theta(X)\big)\big]\Big| \\
&\leq \mathbb{E}\Big[\Big|\frac{1}{T}\sum_{j=1}^{T}\big(\|y_j - z_j\|^2 - \|y_j - \Xi_\theta(\tilde{X}_{[j]})\|^2\big)\Big|\Big] \\
&\leq \mathbb{E}\Big[\frac{1}{T}\sum_{j=1}^{T}\big|\big\langle z_j + \Xi_\theta(\tilde{X}_{[j]}) - 2y_j, z_j - \Xi_\theta(\tilde{X}_{[j]})\big\rangle\big|\Big] \\
&\leq \mathbb{E}\Big[\frac{1}{T}\sum_{j=1}^{T}\|z_j + \Xi_\theta(\tilde{X}_{[j]}) - 2y_j\| \times \|z_j - \Xi_\theta(\tilde{X}_{[j]})\|\Big] \\
&\qquad \text{(by the Cauchy-Schwartz inequality).}
\end{aligned}$$

According to inequality (5.14), one has

$$\|z_j - \Xi_\theta(\tilde{X}_{[j]})\| \leq \|\psi\|_{\mathrm{op}}\frac{c_{1,\theta}}{T},$$

where $c_{1,\theta} = K_{f_\theta}e^{K_{f_\theta}}\big(L + \|f_\theta\|_\infty e^{K_{f_\theta}}\big)$. Moreover,

$$\big\|\Xi_\theta(\tilde{X}_{[j]})\big\|^2 = \sum_{\ell=1}^{p}\big|\langle\alpha_{\ell,\theta}, S(\tilde{X}_{[j]})\rangle_{\mathscr{T}}\big|^2 \leq \sum_{\ell=1}^{p}\|\alpha_{\ell,\theta}\|_{\mathscr{T}}^2\|S(\tilde{X}_{[j]})\|_{\mathscr{T}}^2 \leq pB^2\big(2(1-L)^{-1}\big)^2,$$

since $\|S(\tilde{X}_{[j]})\|_{\mathscr{T}} = \|S_{[0,j/T]}(\bar{X})\|_{\mathscr{T}} \leq \|S(\bar{X})\|_{\mathscr{T}}$. This yields

$$\begin{aligned}
\|z_j + \Xi_\theta(\tilde{X}_{[j]}) - 2y_j\| &\leq \|z_j\| + \|\Xi_\theta(\tilde{X}_{[j]})\| + 2\|y_j\| \\
&\leq \|\psi\|_{\mathrm{op}}\|f_\theta\|_\infty + 2\sqrt{p}B(1-L)^{-1} + 2K_y.
\end{aligned}$$

Finally,

$$\sup_\theta|\mathscr{R}_{\mathscr{G}}(\theta) - \mathscr{R}_{\mathscr{S}}(\theta)| \leq \frac{c_3}{2T},$$

where $c_3 = \sup_\theta\big(c_{1,\theta} + \|\psi\|_{\mathrm{op}}\|f_\theta\|_\infty\big) + 2\sqrt{p}B(1-L)^{-1} + 2K_y$. One proves with similar arguments that

$$\sup_\theta|\widehat{\mathscr{R}}_{n,\mathscr{G}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)| \leq \frac{c_3}{2T}.$$

We now turn to the term $\sup_\theta|\mathscr{R}_{\mathscr{S}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)|$. We have

$$\begin{aligned}
&\mathscr{R}_{\mathscr{S}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta) \\
&\quad= \mathbb{E}[\ell(\mathbf{y}, \Gamma_\theta(X))] - \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{y}^{(i)}, \Gamma_\theta(X^{(i)})) \\
&\quad= \frac{1}{T}\sum_{j=1}^{T}\Big(\mathbb{E}[\|y_j - \Xi_\theta(\tilde{X}_{[j]})]\|^2 - \frac{1}{n}\sum_{i=1}^{n}\big\|y_j^{(i)} - \Xi_\theta(\tilde{X}_{[j]}^{(i)})\big\|^2\Big).
\end{aligned}$$

Therefore,

$$\sup_\theta|\mathscr{R}_{\mathscr{S}}(\theta) - \widehat{\mathscr{R}}_{n,\mathscr{S}}(\theta)| \leq \frac{1}{T}\sum_{j=1}^{T}\sup_\theta\Big|\mathbb{E}[\|y_j - \Xi_\theta(\tilde{X}_{[j]})]\|^2 - \frac{1}{n}\sum_{i=1}^{n}\big\|y_j^{(i)} - \Xi_\theta(\tilde{X}_{[j]}^{(i)})\big\|^2\Big|.$$

Note that for a fixed j, the pairs $(\tilde{X}_{[j]}^{(i)}, y_j^{(i)})$ are i.i.d. Under the assumptions of the theorem, there exists a ball $\mathscr{B} \subset \mathscr{H}$ such that for any $1 \leq \ell \leq p$, $\theta \in \Theta$, $\xi_{\alpha_{\ell,\theta}} \in \mathscr{B}$. We denote by $\mathscr{B}_p$ the sum of p such spaces, that is,

$$\mathscr{B}_p = \big\{ f_\alpha : \mathscr{X} \to \mathbb{R}^p \,|\, f_\alpha(X) = (f_{\alpha_1}(X), \dots, f_{\alpha_p}(X))^\top, f_{\alpha_\ell} \in \mathscr{B} \big\}.$$

Clearly, $\Xi_\theta \in \mathscr{B}_p$, and it follows that

$$\begin{aligned}
&\sup_\theta \Big| \mathbb{E}[\|y_j - \Xi_\theta(\tilde{X}_{[j]})]\|^2 - \frac{1}{n} \sum_{i=1}^n \big\| y_j^{(i)} - \Xi_\theta(\tilde{X}_{[j]}^{(i)}) \big\|^2 \Big| \\
&\qquad \leq \sup_{f_\alpha \in \mathscr{B}_p} \Big| \mathbb{E}\big[\|y_j - f_\alpha(\tilde{X}_{[j]})\|^2 \big] - \frac{1}{n} \sum_{i=1}^n \|y_j^{(i)} - f_\alpha(\tilde{X}_{[j]}^{(i)})\|^2 \Big|.
\end{aligned}$$

We have once again reached a familiar situation, which can be dealt with by an easy extension of Bartlett and Mendelson (2002, Theorem 12). For any $f_\alpha \in \mathscr{B}_p$, let $\tilde{\phi} \circ f_\alpha : \mathscr{X} \times \mathbb{R}^p : (X, y) \mapsto \|y - f_\alpha(X)\|^2 - \|y\|^2$. Then, $\tilde{\phi} \circ f_\alpha$ is upper bounded by

$$\begin{aligned}
|\tilde{\phi} \circ f_\alpha(X, y)| = \big| \|y - f_\alpha(X)\|^2 - \|y\|^2 \big| &\leq \|f_\alpha(X)\| \big(\|f_\alpha(X)\| + 2\|y\| \big) \\
&\leq 2\sqrt{p} B (1 - L)^{-1} (2\sqrt{p} B (1 - L)^{-1} + 2K_y) \\
&\leq 4pB(1 - L)^{-1} (B(1 - L)^{-1} + K_y).
\end{aligned}$$

Let $c_4 = B(1 - L)^{-1} + K_y$ and $c_5 = 4pB(1 - L)^{-1} c_4 + K_y^2$. Then with probability at least $1 - \delta$,

$$\sup_{f_\alpha \in \mathscr{B}_p} \Big| \mathbb{E}\big[\|y_j - f_\alpha(\tilde{X}_{[j]})\| \big] - \frac{1}{n} \sum_{i=1}^n \|y_j^{(i)} - f_\alpha(\tilde{X}_{[j]}^{(i)})\| \Big| \leq \mathrm{Rad}_n(\tilde{\phi} \circ \mathscr{B}_p) + \sqrt{\frac{2c_5 \log(1/\delta)}{n}},$$

where $\tilde{\phi} \circ \mathscr{B}_p = \big\{ (X, y) \mapsto \tilde{\phi} \circ f_\alpha(X, y) | f_\alpha \in \mathscr{B}_p \big\}$. Elementary computations on Rademacher complexities yield

$$\mathrm{Rad}_n(\tilde{\phi} \circ \mathscr{B}_p) \leq 2pc_4 \mathrm{Rad}_n(\mathscr{B}) \leq \frac{4pc_4 B(1 - L)^{-1}}{\sqrt{n}},$$

which concludes the proof.

5.C Differentiation with higher-order tensors

5.C.1 Definition

We define the generalization of matrix product between square tensors of order k and ℓ.

Definition 5.26. *Let $a \in (\mathbb{R}^e)^{\otimes k}$, $b \in (\mathbb{R}^e)^{\otimes \ell}$, $p \in \{1, \dots, k\}$, $q \in \{1, \dots, \ell\}$. Then the tensor dot product along (p, q), denoted by $a \odot_{p,q} b \in (\mathbb{R}^e)^{\otimes (k+\ell-2)}$, is defined by*

$$(a \odot_{p,q} b)_{(i_1, \dots, i_{k-1}, j_1, \dots, j_{\ell-1})} = \sum_{j=1}^e a_{(i_1, \dots, i_{p-1}, j, i_p, \dots, i_{k-1})} b_{(j_1, \dots, j_{q-1}, j, j_q, \dots, j_{\ell-1})}.$$

This operation just consists in computing $a \otimes b$, and then summing the pth coordinate of a with the qth coordinate of b. The $\odot$ operator is not associative. To simplify notation, we take the convention that it is evaluated from left to right, that is, we write $a \odot b \odot c$ for $(a \odot b) \odot c$.

Definition 5.27. *Let $a \in (\mathbb{R}^e)^{\otimes k}$. For a given permutation π of $\{1, \dots, k\}$, we denote by $\pi(a)$ the permuted tensor in $(\mathbb{R}^e)^{\otimes k}$ such that*

$$\pi(a)_{(i_1, \dots, i_k)} = a_{(i_{\Pi(1)}, \dots, i_{\Pi(k)})}.$$

Example 5.28. *If A is a matrix, then $A^T = \pi(A)$, with π defined by $\pi(1) = 2, \pi(2) = 1$.*

5.C.2 Computation rules

We need to obtain two computation rules for the tensor dot product: bounding the norm (Lemma 5.29) and differentiating (Lemma 5.30).

Lemma 5.29. *Let $a \in (\mathbb{R}^e)^{\otimes k}$, $b \in (\mathbb{R}^e)^{\otimes \ell}$. Then, for all p, q,*

$$\|a \odot_{p,q} b\|_{(\mathbb{R}^e)^{\otimes k+\ell-2d}} \leq \|a\|_{(\mathbb{R}^e)^{\otimes k}} \|b\|_{(\mathbb{R}^e)^{\otimes \ell}}.$$

Proof. By the Cauchy-Schwartz inequality,

$$\begin{aligned}
&\|a \odot_{p,q} b\|^2_{(\mathbb{R}^e)^{\otimes k+\ell-2}} \\
&\quad = \sum_{1 \leq i_1,\dots,i_{k-1},j_1,\dots,j_{\ell-1} \leq e} (a \odot_{p,q} b)^2_{(i_1,\dots,i_{k-1},j_1,\dots,j_{\ell-1})} \\
&\quad = \sum_{1 \leq i_1,\dots,i_{k-1},j_1,\dots,j_{\ell-1} \leq e} \Big(\sum_{1 \leq j \leq e} a_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} b_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \Big)^2 \\
&\quad \leq \sum_{i_1,\dots,i_{k-1},j_1,\dots,j_{\ell-1}} \Big(\sum_j a^2_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} \Big) \Big(\sum_j b^2_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \Big) \\
&\quad \leq \sum_{i_1,\dots,i_{k-1},j} a^2_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} \sum_{j_1,\dots,j_{\ell-1},j} b^2_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \\
&\quad \leq \|a\|^2_{(\mathbb{R}^e)^{\otimes k}} \|b\|^2_{(\mathbb{R}^e)^{\otimes \ell}}.
\end{aligned}$$

□

Lemma 5.30. *Let $A : \mathbb{R}^e \to (\mathbb{R}^e)^{\otimes k}$, $B : \mathbb{R}^e \to (\mathbb{R}^e)^{\otimes \ell}$ be smooth vector fields, $p \in \{1,\dots,k\}$, $q \in \{1,\dots,\ell\}$. Let $A \odot_{p,q} B : \mathbb{R}^e \to (\mathbb{R}^e)^{\otimes k+\ell-2}$ be defined by $A \odot_{p,q} B(h) = A(h) \odot_{p,q} B(h)$. Then there exists a permutation π such that*

$$J(A \odot_{p,q} B) = \pi(J(A) \odot_{p,q} B) + A \odot_{p,q} J(B).$$

Proof. The left-hand side takes the form

$$\begin{aligned}
(J(A \odot_{p,q} B))_{i_1,\dots,i_{k-1},j_1,\dots,j_{\ell-1},m} = \sum_j \Big[&\frac{\partial A}{\partial h_m}_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} B_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \\
&+ A_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} \frac{\partial B}{\partial h_m}_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \Big].
\end{aligned}$$

The first term of the right-hand side writes

$$(J(A) \odot_{p,q} B)_{i_1,\dots,i_{k-1},m,j_1,\dots,j_{\ell-1}} = \sum_j \Big[\frac{\partial A}{\partial h_m}_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} B_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \Big],$$

and the second one

$$(A \odot_{p,q} J(B))_{i_1,\dots,i_{k-1},j_1,\dots,j_{\ell-1},m} = \sum_j \Big[A_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} \frac{\partial B}{\partial h_m}_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \Big].$$

Let us introduce the permutation π which keeps the first $(k-1)$ axes unmoved, and rotates the remaining ℓ ones such that the last axis ends up in kth position. Then

$$\pi(J(A) \odot_{p,q} B)_{i_1,\dots,i_{k-1},j_1,\dots,j_{\ell-1},m} = \sum_j \Big[\frac{\partial A}{\partial h_m}_{(i_1,\dots,i_{p-1},j,i_p,\dots,i_{k-1})} B_{(j_1,\dots,j_{q-1},j,j_q,\dots,j_{\ell-1})} \Big].$$

Hence $J(A \odot_{p,q} B) = \pi(J(A) \odot_{p,q} B) + A \odot_{p,q} J(B)$, which concludes the proof. □

The following two lemmas show how to compose the Jacobian and the tensor dot operations with permutations. Their proofs follow elementary operations and are therefore omitted.

Lemma 5.31. *Let $A : \mathbb{R}^e \to (\mathbb{R}^e)^{\otimes k}$ and π a permutation of $\{1, \dots, k\}$. Then there exists a permutation $\tilde{\pi}$ of $\{1, \dots, k+1\}$ such that*

$$J(\pi(A)) = \tilde{\pi}(J(A)).$$

Lemma 5.32. *Let $a \in (\mathbb{R}^e)^{\otimes k}$, $b \in (\mathbb{R}^e)^{\otimes \ell}$, $p \in \{1, \dots, k\}$, $q \in \{1, \dots, \ell\}$, π a permutation of $\{1, \dots, k\}$. Then there exists $\tilde{p} \in \{1, \dots, k\}$, $\tilde{q} \in \{1, \dots, \ell\}$, and a permutation $\tilde{\pi}$ of $\{1, \dots, k + \ell - 2\}$ such that*

$$\pi(a) \odot_{p,q} b = \tilde{\pi}(a \odot_{\tilde{p},\tilde{q}} b).$$

The following result is a generalization of Lemma 5.30 to the case of a dot product of several tensors.

Lemma 5.33. *For $\ell \in \{1, \dots, k\}$, $n_\ell \in \mathbb{N}$, let $A_\ell : \mathbb{R}^e \to (\mathbb{R}^e)^{\otimes n_\ell}$ be smooth tensor fields. For any $(p_\ell)_{1 \le \ell \le k-1}$ and $(q_\ell)_{1 \le \ell \le k-1}$ such that $p_\ell \in \{1, \dots, n_\ell\}$, $q_\ell \in \{1, \dots, n_{\ell+1}\}$, there exist k permutations $(\pi_\ell)_{1 \le \ell \le k}$ such that*

$$J(A_1 \odot_{p_1,q_1} A_2 \odot_{p_2,q_2} \cdots \odot_{p_{k-1},q_{k-1}} A_k) = \sum_{\ell=1}^{k} \pi_\ell \left[A_1 \odot A_2 \odot \cdots \odot J(A_\ell) \odot \cdots \odot A_k \right],$$

where the dot products of the right-hand side are along some axes that are not specify for simplicity.

Proof. The proof is done by induction on k. The formula for $k = 1$ is straightforward. Assume that the formula is true at order k. As before, we do not specify indexes for tensor dot products as we are only interested in their existence. By Lemma 5.32, we have

$$\begin{aligned}
&J(A_1 \odot \cdots \odot A_{k+1}) \\
&\quad = J((A_1 \odot \cdots \odot A_k) \odot A_{k+1}) \\
&\quad = \pi(J(A_1 \odot \cdots \odot A_k) \odot A_{k+1}) + A_1 \odot \cdots \odot A_k \odot J(A_{k+1}) \\
&\quad = \pi\left[\sum_{\ell=1}^{k} \pi_\ell \left[A_1 \odot A_2 \odot \cdots \odot J(A_\ell) \odot \cdots \odot A_k \right] \odot A_{k+1} \right] + A_1 \odot \cdots \odot A_k \odot J(A_{k+1}) \\
&\quad = \pi\left[\sum_{\ell=1}^{k} \tilde{\pi}_\ell \left[A_1 \odot A_2 \odot \cdots \odot J(A_\ell) \odot \cdots \odot A_k \odot A_{k+1} \right] \right] + A_1 \odot \cdots \odot A_k \odot J(A_{k+1}) \\
&\quad = \sum_{\ell=1}^{k} \hat{\pi}_\ell \left[A_1 \odot A_2 \odot \cdots \odot J(A_\ell) \odot \cdots \odot A_k \odot A_{k+1} \right] + A_1 \odot \cdots \odot A_k \odot J(A_{k+1}) \\
&\qquad (\text{where } \hat{\pi} = \pi \circ \tilde{\pi}) \\
&\quad = \sum_{\ell=1}^{k+1} \hat{\pi}_\ell \left[A_1 \odot A_2 \odot \cdots \odot J(A_\ell) \odot \cdots \odot A_k \odot A_{k+1} \right].
\end{aligned}$$

□

5.D Experimental details

All the code to reproduce the experiments is available on GitHub at `https://github.com/afermanian/rnn-kernel`. Our experiments are based on the PyTorch (Paszke et al., 2019) framework. When not specified, the default parameters of PyTorch are used.

Convergence of the Taylor expansion. For Figure 5.1, 10^3 random RNN with 2 hidden units are generated, with the default weight initialization. The activation is either the logistic or the hyperbolic tangent. In Figure 5.1b, only the results with the logistic activation are plotted. The process X is taken as a 2-dimensional spiral. The reference solution to the ODE (5.3) is computed with a numerical integration method from SciPy (Virtanen et al., 2020, `scipy.integrate.solve_ivp` with the 'LSODA' method). The signature in the step-N Taylor expansion is computed with the package Signatory (Kidger and Lyons, 2021).

The step-N Taylor expansion requires computing higher-order derivatives of tensor fields (up to order N). This is a highly non-trivial task since standard deep learning frameworks are optimized for first-order differentiation only. We refer to, for example, Kelly et al. (2020), for a discussion on higher-order differentiation in the context of a deep learning framework. To compute it efficiently, we manually implement forward-mode higher-order automatic differentiation for the operations needed in our context (described in Appendix 5.C). A more efficient and general approach is left for future work. Our code is optimized for GPU.

Penalization on a toy example. For Figure 5.2, the RNN is taken with 32 hidden units and hyperbolic tangent activation. The data are 50 examples of spirals, sampled at 100 points and labeled ± 1 according to their rotation direction. We do not use batching and the loss is taken as the cross entropy. It is trained for 200 epochs with Adam (Kingma and Ba, 2015) with an initial learning rate of 0.1. The learning rate is divided by 2 every 40 epochs. For the penalized RNN, the RKHS norm is truncated at $N = 3$ and the regularization parameter is selected at $\lambda = 0.1$. Earlier experiments show that this order of magnitude is sensible. We do not perform hyperparameter optimization since our goal is not to achieve high performance. The initial hidden state h_0 is learned (for simplicity of presentation, our theoretical results were written with $h_0 = 0$ but they extend to this case). The accuracy is computed on a test set of size 1000. We generate adversarial examples using 50 steps of projected gradient descent (following Bietti et al., 2019). The whole methodology (data generation + training) is repeated 20 times. The average training time on a Tesla V100 GPU for the RNN is 8.5 seconds and for the penalized RNN 12 seconds.

Figure 5.3 is obtained by selecting randomly one run among the 20 of Figure 5.2.

Libraries. We use PyTorch (Paszke et al., 2019) as our overall framework, Signatory (Kidger and Lyons, 2021) to compute the signatures, and SciPy (Virtanen et al., 2020) for ODE integration. We use Sacred (Klaus Greff et al., 2017) for experiment management.

Part II

Contributions to finite-depth neural networks

6 Leveraging the two-timescale regime to demonstrate convergence of neural networks

We study the training dynamics of shallow neural networks, in a two-timescale regime in which the step sizes for the inner layer are much smaller than those for the outer layer. In this regime, we prove convergence of the gradient flow to a global optimum of the non-convex optimization problem in a simple univariate setting. The number of neurons needs not be asymptotically large for our result to hold, distinguishing our result from popular recent approaches such as the neural tangent kernel or mean-field regimes. Experimental illustration is provided, showing that the stochastic gradient descent behaves according to our description of the gradient flow and thus converges to a global optimum in the two-timescale regime, but can fail outside this regime.

Contents

6.1 Introduction

Artificial neural networks are among the most successful modern machine learning methods, in particular because their non-linear parametrization provides a flexible way to implement feature learning (see, e.g., Goodfellow et al., 2016, chapter 15). Following this empirical success, a large

body of work has been dedicated to understanding their theoretical properties, and in particular to analyzing the optimization algorithm used to tune their parameters. It usually consists in minimizing a loss function through stochastic gradient descent (SGD) or a variant (Bottou et al., 2018). However, the non-linearity of the parametrization implies that the loss function is non-convex, breaking the standard convexity assumption that ensures global convergence of gradient descent algorithms.

In this chapter, we study the training dynamics of *shallow* neural networks, i.e., of the form

$$f(x; a, u) = a_0 + \sum_{j=1}^{m} a_j g(x; u_j)\,,$$

where m denotes the number of hidden neurons, $a = (a_0, \dots, a_m)$ and $u = (u_1, \dots, u_m)$ denote respectively the outer and inner layer parameters, and $g(x; u)$ denotes a non-linear function of x and u. The novelty of this work lies in the use of a so-called *two-timescale regime* (Borkar, 1997) to train the neural network: we set step sizes for the inner layer u to be an order of magnitude smaller than the step sizes of the outer layer a. This ratio is controlled by a parameter ε. In the regime $\varepsilon \ll 1$, the neural network can be thought of as a fitted linear regression with slowly evolving features $g(x; u_j)$, $j = 1, \dots, m$: this reduction enables us to precisely describe the movement of the inner layer parameters u_j.

Our approach proves convergence of the *gradient flow* to a global optimum of the non-convex landscape with a *fixed* number m of neurons. The gradient flow can be seen as the simplifying yet insightful limit of the SGD dynamics as the step size h vanishes. Proving convergence with a fixed number of neurons contrasts with two other popular approaches that require to take the limit $m \to \infty$: the neural tangent kernel (Jacot et al., 2018; Allen-Zhu et al., 2019; Du et al., 2019; Zou et al., 2020a) and the mean-field approach (Chizat and Bach, 2018; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2020). As a consequence, this chapter is intended as a step towards understanding feature learning with a moderate number of neurons.

While our approach through the two-timescale regime is general, our description of the solution of the two-timescale dynamics and our convergence results are specific to a simple example showcasing the approach. More precisely, we consider univariate data $x \in [0, 1]$ and non-linearities of the form $g(x; u_j) = \sigma(\eta^{-1}(x - u_j))$, where u_j is a variable translation parameter, η is a fixed dilatation parameter, and σ is a sigmoid-like non-linearity. Finally, we restrict ourselves to the approximation of piecewise constant functions.

Organization of this chapter. In Section 6.2, we detail our setting and state our main theorem on the convergence of the gradient flow to a global optimum. Section 6.3 articulates this chapter with related work. Section 6.4 provides a self-contained introduction to the *two-timescale limit* $\varepsilon \to 0$. We explain how it simplifies the analysis of neural networks, and provides heuristic predictions for the movement of neurons in our setting. Section 6.5 gives a rigorous derivation of our result. We prove convergence first in the two-timescale limit $\varepsilon \to 0$, then in the two-timescale regime with ε small but positive. Section 6.6 presents numerical experiments showing that the SGD dynamics follow closely those of the gradient flow in the two-timescale regime, and therefore exhibit convergence to a global optimum. On the contrary, SGD can fail to reach a global optimum outside the two-timescale regime. We give a conclusion in Section 6.7 before moving on to the details of the proofs of this chapter. In Section 6.A, we introduce additional notations that will be used throughout in the detailed proofs, then proceed to prove useful technical lemmas. We proceed in Section 6.B to present the detailed proofs of the results of the chapter. Finally, Section 6.C contains details about our experimental settings as well as some additional simulations.

6.2 Setting and main result

We present a setting in which a piecewise constant univariate function $f^* : [0,1] \to \mathbb{R}$ is learned with gradient flow on a shallow neural network. Our notations are summarized on Figure 6.1. We begin by introducing our class of functions of interest.

Definition 6.1. *Let $n \geqslant 2$, $\Delta v \in (0,1)$, $\Delta f > 0$ and $M \geqslant 1$. We denote $\mathcal{F}_{n,\Delta v,\Delta f,M}$ the class of functions $f^* : [0,1] \to \mathbb{R}$ satisfying the following conditions:*

- *f^* is piecewise constant: there exists*

$$0 = v_0 < v_1 < \cdots < v_{n-1} < v_n = 1$$

and $f_0^*, \dots, f_{n-1}^* \in \mathbb{R}$ such that

$$\forall x \in (v_i, v_{i+1}), f^*(x) = f_i^*,$$

- *for all $i \in \{1, \dots, n\}, v_i - v_{i-1} \geqslant \Delta v$,*
- *for all $i \in \{1, \dots, n-1\}, |f_i^* - f_{i-1}^*| \geqslant \Delta f$,*
- *for all $i \in \{0, \dots, n-1\}$, $|f_i^*| \leqslant M$.*

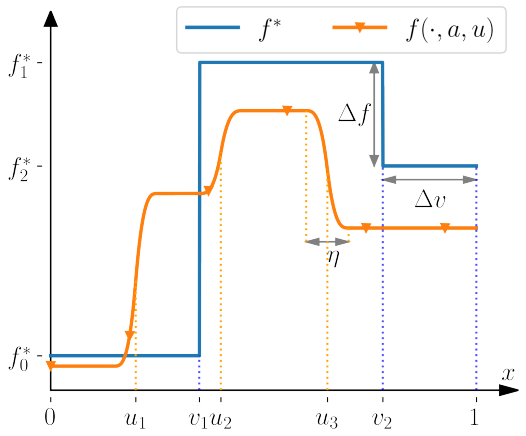


Figure 6.1: Notations of the chapter. The target f^* is in blue and the neural network $f(\cdot; a, u)$ in orange.

Let us now define our class of neural networks. Consider $\sigma : \mathbb{R} \to \mathbb{R}$ an increasing, twice continuously differentiable non-linearity such that $\sigma(x) = 0$ if $x \leqslant -1/2$, $\sigma(x) = 1$ if $x \geqslant 1/2$, and $\sigma - 1/2$ is odd. Then, our class of shallow neural networks is defined by

$$f(x; a, u) = a_0 + \sum_{j=1}^{m} a_j \sigma_\eta(x - u_j)\,, \qquad \sigma_\eta(x) = \sigma(\eta^{-1} x)\,,$$

where $0 < \eta \leq 1$ measure the sharpness of the non-linearity σ_η. Note that that inner layer parameter u_j determines the translation of the non-linearity; no parameterized multiplicative operation on x is performed in this layer. We refer to the parameter u as the "positions" of the neurons (or, sometimes, simply as the "neurons") and to the parameter a as the "weights" of the neurons. The quadratic loss is defined as

$$L(a, u) = \frac{1}{2} \int_0^1 (f^*(x) - f(x; a, u))^2 \, \mathrm{d}x\,.$$

We use gradient flow on L to fit the parameters a and u: they evolve according to the dynamics

$$\frac{\mathrm{d}a}{\mathrm{d}t}(t) = -\nabla_a L(a(t), u(t))\,, \qquad \frac{\mathrm{d}u}{\mathrm{d}t}(t) = -\varepsilon \nabla_u L(a(t), u(t))\,, \tag{6.1}$$

where ε corresponds to the ratio of the step sizes of the two iterations.

Main result. By leveraging the two-timescale regime where ε is small, our theorem shows that, with high probability, a neural network trained with gradient flow is able to recover an arbitrary piecewise constant function to an arbitrary precision. The proof is relegated to the Appendix.

Theorem 6.2. *Let $\xi, \delta > 0$, and f^* a piecewise constant function from $\mathcal{F}_{n,\Delta v,\Delta f,M}$. Assume that the neural network has m neurons with*

$$m \geqslant \frac{6}{\Delta v}\big(4 + \log n + \log \frac{1}{\delta}\big)\,. \tag{6.2}$$

Assume that, at initialization, the positions $u_1, \dots, u_m$ of the neurons are i.i.d. uniformly distributed on $[0,1]$ and their weights $a_0, \dots, a_m$ are equal to zero.

Then there exists $Q_1 > 0$ and $Q_2 > 0$ depending on $\xi, \delta, m, \Delta f, M$ such that, if

$$\eta \leqslant Q_1\,, \qquad\qquad \varepsilon \leqslant Q_2\,, \tag{6.3}$$

then, with probability at least $1 - \delta$, the solution to the gradient flow (6.1) is defined at least until $T = \frac{6}{\varepsilon(\Delta f)^2}$, and

$$\int_0^1 |f^*(x) - f(x; a(T), u(T))|^2 \,\mathrm{d}x \leqslant \xi\,.$$

Further, $Q_1 = \dfrac{C_1}{M^2(m+1)} \min\Big(\dfrac{\delta^2(\Delta f)^2}{(m+1)^4}, \xi\Big)$ and $Q_2 = \dfrac{C_2\delta^2}{M^4(m+1)^{17/2}} \min\Big(\dfrac{\delta(\Delta f)^2}{m+1}, \xi\Big)$ for some universal constants $C_1, C_2 > 0$.

Proof. See Section 6.B.6. □

For this result to hold, the inequality (6.2) requires the number of neurons in the neural network to be large enough. Note that the minimum number of neurons required to approximate the n pieces of f^* is equal to n. If the length of all the intervals is of the same order of magnitude, then $\Delta v = \Theta(1/n)$ and thus the condition is $m = \Omega(n(1+\log n+\log 1/\delta))$. In this case, condition (6.2) only adds a logarithmic factor in n and δ. Moreover, the lower bound on m does not depend on the target precision ξ. Thus we observe some non-linear feature learning phenomenon: with a fixed number m of neurons, gradient flow on a neural network can approximate any element from the infinite-dimensional space of piecewise constant functions to an arbitrary precision.

The recovery result of Theorem 6.2 is provided under two conditions (6.3). The first one should not surprise the reader: the condition on η enables the non-linearity to be sharp enough in order to approximate well the jumps of the piecewise constant function f^*. The novelty of our work lies in the condition on ε, that we refer to as the *two-timescale regime*. This condition ensures that the step sizes taken in the positions u are much smaller than the step sizes taken in the weights a. As a consequence, the weights a are constantly close to the best linear fit given the current positions u. This property decouples the dynamics of the two layers of the neural network; this enables a sharp description of the gradient flow trajectories and thus the recovery result shown above. This intuition is detailed in Section 6.4.

6.3 Related work

Two-timescale regime. Systems with two timescales, or *slow-fast systems*, have a long history in physics and mathematics, see Berglund and Gentz (2006, Chapter 2) for an introduction. In particular, iterative algorithms with two timescales have been used in stochastic approximation and optimization, see Borkar (1997) or Borkar (2009, Section 6). For instance, they are used in the training of generative adversarial networks, to decouple the dynamics of the generator from those of the discriminator (Heusel et al., 2017), in reinforcement learning, to decouple the value function estimation from the temporal difference learning (Szepesvári, 2010), or more generally in bilevel optimization, to decouple the outer problem dynamics from the inner problem dynamics (Hong et al., 2023). However, to the best of our knowledge, the two-timescale regime has not been used to show convergence results for neural networks.

Layer-wise learning rates. Practitioners are interested in choosing learning rates that depend on the layer index to speed up training or improve performance. Using smaller learning rates for the first layers and higher learning rates for the last layer(s) improves performance for fine-tuning (Howard and Ruder, 2018; Ro and Choi, 2021) and is a common practice for transfer learning (see, e.g., Li et al., 2022). Another line of work proposes to update layer-wise learning rates depending on the norm of the gradients on each layer (Singh et al., 2015; You et al., 2017; Ko et al., 2022). However, they aim to compensate the differences across gradient norms in order to learn all the parameters at the same speed, while on the contrary we enjoy the theoretical benefits of learning different speeds.

Theory of neural networks. A key novelty of the analysis of this chapter is that we show recovery with a fixed number of neurons. We now detail the comparison with other analyses.

The neural tangent kernel regime (Jacot et al., 2018; Allen-Zhu et al., 2019; Du et al., 2019; Zou et al., 2020a) corresponds to small movements of the parameters of the neural network. In this case, the neural network can be linearized around its initial point, and thus behaves like a linear regression. However, in this regime, the neural network can approximate only a finite dimensional space of functions, and thus it is necessary to take $m \to \infty$ to be able to approximate the infinite-dimensional space of piecewise constant functions to an arbitrary precision.

The mean-field regime (Chizat and Bach, 2018; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2020) describes the dynamics of two-layer neural networks in the regime $m \gg 1$ through a partial differential equation on the density of neurons. This regime is able to describe some non-linear feature learning phenomena, but does not explain the observed behavior with a moderate number of neurons. In this chapter, we show that in the two-timescale regime, only a single neuron aligns with each of the discontinuities of the function f^*. However, it should be noted that the neural tangent kernel and mean-field regimes have been applied to show recovery in a wide range of settings, while our work is restricted to the recovery of piecewise constant functions. Extending the application of the two-timescale regime is left for future work.

Our work includes a detailed analysis of the alignment of the positions of the neurons with the discontinuities of the target function f^*. This is analogous to a line of work (see, e.g., Saad and Solla (1995); Goldt et al. (2020); Veiga et al. (2022)) interested in the alignment of a "student" neural network with the features of a "teacher" neural network that generated the data, for high-dimensional Gaussian input. In general, the non-linear evolution equations describing this alignment are hard to study theoretically. On the contrary, thanks to the two-timescale regime and to the simple setting of this chapter, we are able to give a precise description of the movement of the neurons.

Our study bears high-level similarities with the recent work of Safran et al. (2022). In a univariate classification setting, they show that a two-layer neural network achieves recovery with a number of neurons analogous to (6.2): inversely proportional to the length of the smallest constant interval of the target, up to logarithmic terms in the number of constant intervals and in the failure probability. However, the two works are not comparable due to differences in the settings: Safran et al. (2022) consider classification with ReLU activations while we consider regression with sigmoid-like activations. More importantly, the authors do not use the two-timescale regime. Instead, by a specific scale of the initialization, they ensure that the neural network has a first phase in a lazy regime where the positions of the neurons do not move significantly. For the second rich phase, they describe the implicit bias of the limiting point; this approach does not lead to an estimate of the convergence time while the fine description of the two-timescale limit does.

6.4 A non-rigorous introduction to the two-timescale limit

This section introduces the core ideas of our analysis in a non-rigorous way. Section 6.4.1 introduces the limit of the dynamics when $\varepsilon \to 0$, called the *two-timescale limit*. Section 6.4.2 applies the two-timescale limit to predict the movement of the neurons.

6.4.1 Introduction to the two-timescale limit

Let us consider the gradient flow equations (6.1) and perform the change of variables $\tau = \varepsilon t$:

$$\frac{\mathrm{d}a}{\mathrm{d}\tau} = -\frac{1}{\varepsilon}\nabla_a L(a,u)\,, \qquad\qquad \frac{\mathrm{d}u}{\mathrm{d}\tau} = -\nabla_u L(a,u)\,. \tag{6.4}$$

In the two-timescale regime $\varepsilon \ll 1$, the rate of the gradient flow in the weights a is much larger than then the rate in the positions u. Note that L is marginally convex in a, and thus, for a fixed u, the gradient flow in a must converge to a global minimizer of $a \mapsto L(a,u)$. More precisely, assume that $\{\sigma_\eta(.-u_1),\dots,\sigma_\eta(.-u_m)\}$ forms an independent set of functions in $L^2([0,1])$. Then the global minimizer of $a \mapsto L(a,u)$ is unique; we denote it as $a^*(u)$. In the limit $\varepsilon \to 0$, we expect that a evolves sufficiently quickly with respect to u so that it converges instantaneously to $a^*(u)$. In other words, the gradient flow system (6.4) reduces to its so-called *two-timescale limit* when $\varepsilon \to 0$:

$$a = a^*(u)\,, \qquad\qquad \frac{\mathrm{d}u}{\mathrm{d}\tau} = -\nabla_u L(a^*(u),u)\,. \tag{6.5}$$

The two-timescale limit considerably simplifies the study of the gradient flow system because it substitutes the weights a, determined to be equal to $a^*(u)$. However, showing that (6.4) reduces to (6.5) requires some mathematical care, including checking that $a^*(u)$ is well-defined.

Remark 6.3 (abuse of notation)**.** *Equation* (6.5) *contains the ambiguous notation $\nabla_u L(a^*(u),u)$: does it denote the gradient in u of the map $L^* : u \mapsto L(a^*(u),u)$ or the gradient $(\nabla_u L)(a,u)$ taken at $a = a^*(u)$? In fact, by definition of $a^*(u)$, both quantities coincide:*

$$(\nabla_u L^*)(u) = \Big(\frac{\mathrm{d}a^*}{\mathrm{d}u}(u)\Big)^\top (\nabla_a L)(a^*(u),u) + (\nabla_u L)(a^*(u),u) = (\nabla_u L)(a^*(u),u)\,,$$

where $\frac{\mathrm{d}a^*}{\mathrm{d}u}$ denotes the differential of a^* in u and, by definition of $a^*(u)$, $(\nabla_a L)(a^*(u),u) = 0$. This is a special case of the envelope theorem (Border, 2015, Sec. 5.10).

The discussion in this section is not specific to the setting of Section 6.2. Using the two-timescale limit to decouple the dynamics of the outer layer a and the inner layer u is a general tool that may be used in the study of any two-layer neural network. We chose the specific setting of this chapter so that the two-timescale limit (6.5) can be easily studied, thereby showcasing the approach. The next section is devoted to a sketch of this study.

6.4.2 Sketch of the dynamics of the two-timescale limit

In this section, in order to simplify the exposition of the behavior of the two-timescale limit (6.5), we consider the limiting case $\eta \to 0$. Note that this is coherent with Theorem 6.2 that requires η to be small. This limit is a neural network with a non-linearity equal to the Heaviside function

$$\sigma_0(x) = 0 \quad \text{if } x < 0\,, \qquad \sigma_0(x) = 1/2 \quad \text{if } x = 0\,, \qquad \sigma_0(x) = 1 \quad \text{if } x > 0\,.$$

Note that σ_0 would be a poor choice of non-linearity in practice: as its derivative is 0 almost everywhere, the positions u would not move. However, it is a relevant tool to get an intuition

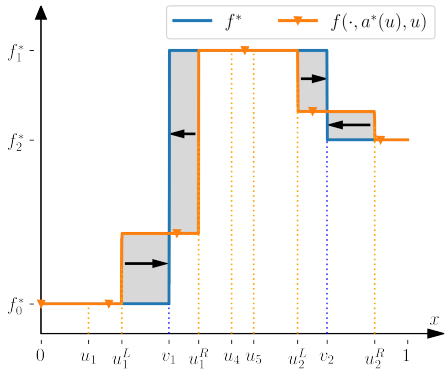


Figure 6.2: Sketch of the dynamics of the neurons in the two-timescale limit with a Heaviside non-linearity. Only the neurons next to a discontinuity of the target move.

about the dynamics of our system for a small η. Moreover, as we will see in Section 6.6, the dynamics sketched here match closely those of the SGD (with $\eta > 0$).

The set $\{1, \sigma_0(. - u_1), \dots, \sigma_0(. - u_m)\}$ generates the space of functions that are piecewise constant with respect to the subdivision $\{u_1, \dots, u_m\}$. Furthermore, if $u_1, \dots, u_m$ are distinct and in $(0, 1)$, then this set is an independent set of functions in $L^2([0, 1])$. Thus $a^*(u)$ is well-defined and represents the coefficients of the best piecewise constant approximation of f^* with subdivision $\{u_1, \dots, u_m\}$.

This quantity is straightforward to describe under the mild additional assumption that there are at least two neurons u_j in each interval (v_{i-1}, v_i) between two points of discontinuity of f^*. For each $1 \leqslant i \leqslant n$, let u_i^{L} denote the largest position of neurons below v_i and u_i^{R} denote the smallest position above v_i (with convention $u_0^{\mathrm{R}} = 0$ and $u_{n+1}^{\mathrm{L}} = 1$). By assumption, $0 = u_0^{\mathrm{R}} < u_1^{\mathrm{L}} < u_1^{\mathrm{R}} < \cdots < u_n^{\mathrm{L}} < u_n^{\mathrm{R}} < u_{n+1}^{\mathrm{L}} = 1$ are distinct. A simple computation then shows the following identities:

- for all $i \in \{1, \dots, n\}$, for all $x \in (u_i^{\mathrm{L}}, u_i^{\mathrm{R}})$, $f(x; a^*(u), u) = \dfrac{v_i - u_i^{\mathrm{L}}}{u_i^{\mathrm{R}} - u_i^{\mathrm{L}}} f_{i-1}^* + \dfrac{u_i^{\mathrm{R}} - v_i}{u_i^{\mathrm{R}} - u_i^{\mathrm{L}}} f_i^*$,
- and for all $i \in \{1, \dots, n+1\}$, for all $x \in (u_{i-1}^{\mathrm{R}}, u_i^{\mathrm{L}})$, $f(x; a^*(u), u) = f_{i-1}^*$,

where we recall that f_i^* denotes the value of f^* on the interval (v_i, v_{i+1}). Figure 6.2 illustrates the situation. Moreover, the loss $L(a^*(u), u)$, which is half of the square L^2-error of this optimal approximation, can be written

$$L(a^*(u), u) = \frac{1}{2} \sum_{i=1}^{n-1} \frac{(v_i - u_i^{\mathrm{L}})(u_i^{\mathrm{R}} - v_i)}{u_i^{\mathrm{R}} - u_i^{\mathrm{L}}} (f_i^* - f_{i-1}^*)^2 \,. \tag{6.6}$$

The dynamics of the two-timescale limit (6.5) corresponds to the local optimization of the subdivision u in order to minimize the loss (6.6). A remarkable property of this loss is that it decomposes as a sum of local losses around the jump points v_i for $i \in \{1, \dots, n-1\}$. Each element of the sum involves only the two neurons located at u_i^{L} and u_i^{R}. As a consequence, the dynamics of the two-timescale limit (6.5) decompose as n independent systems of two neurons u_i^{L} and u_i^{R}: for all $i \in \{1, \dots, n-1\}$,

$$\begin{aligned} \frac{\mathrm{d}u_i^{\mathrm{L}}}{\mathrm{d}\tau} &= -\frac{\mathrm{d}L}{\mathrm{d}u_i^{\mathrm{L}}}(a^*(u), u) = +\frac{1}{2} \frac{(u_i^{\mathrm{R}} - v_i)^2}{(u_i^{\mathrm{R}} - u_i^{\mathrm{L}})^2} (f_i^* - f_{i-1}^*)^2 \,, \\ \frac{\mathrm{d}u_i^{\mathrm{R}}}{\mathrm{d}\tau} &= -\frac{\mathrm{d}L}{\mathrm{d}u_i^{\mathrm{R}}}(a^*(u), u) = -\frac{1}{2} \frac{(v_i - u_i^{\mathrm{L}})^2}{(u_i^{\mathrm{R}} - u_i^{\mathrm{L}})^2} (f_i^* - f_{i-1}^*)^2 \,. \end{aligned} \tag{6.7}$$

All neurons other than $u_1^{\mathrm{L}}, u_1^{\mathrm{R}}, \dots, u_{n-1}^{\mathrm{L}}, u_{n-1}^{\mathrm{R}}$ do not play a role in the expression (6.6), thus they do not move in the two-timescale limit (6.5). The position u_i^{L} moves right and u_i^{R} moves left, until one of them hits the point v_i. This shows that the positions of the neurons eventually align with the jumps of the function f^*, and thus that the function f^* is recovered.

6.5 Convergence of the gradient flow

In this section, we give precise mathematical statements leading to the convergence of the gradient flow to a global optimum, first in the two-timescale limit $\varepsilon \to 0$, then in the two-timescale regime with ε small but positive. All proofs are relegated to the Appendix.

6.5.1 In the two-timescale limit

This section analyzes rigorously the two-timescale limit (6.5), which we recall for convenience:

$$a^*(u) = \operatorname*{argmin}_a L(a, u)\,, \qquad \frac{\mathrm{d}u}{\mathrm{d}\tau} = -\nabla_u L(a^*(u), u)\,. \tag{6.8}$$

We start by giving a rigorous meaning to these equations. First, for L to be differentiable in u, we require the parameter η of the non-linearity to be positive. Second, for $a^*(u)$ to be well-defined, we need $u \mapsto L(a, u)$ to have a unique minimum. Obviously, if the u_i are not distinct, then the features $\{\sigma_\eta(. - u_1), \dots, \sigma_\eta(. - u_m)\}$ are not independent and thus the minimum can not be unique. We restrict the state space of our dynamics to circumvent this issue. For $u \in [0,1]^m$, we denote

$$\Delta(u) = \min_{0 \leqslant j,k \leqslant m+1,\, j \neq k} |u_j - u_k|\,,$$

with the convention that $u_0 = -\eta/2$ and $u_{m+1} = 1 + \eta/2$. Furthermore, let us define the set $\mathcal{U} = \{u \in [0,1]^m \,|\, \Delta(u) > 2\eta\}$. The proposition below shows that $\mathcal{U}$ gives a good candidate for a set supporting solutions of (6.8).

Proposition 6.4. *For $u \in \mathcal{U}$, the Hessian $H(u)$ of the quadratic function $L(., u)$ is positive definite and its smallest eigenvalue is greater than $\Delta(u)/8$. In particular, $L(., u)$ has a unique minimum $a^*(u)$.*

Proof. See Section 6.B.1. □

The bound on the Hessian is useful in the following, in particular in the proof of the following result.

Proposition 6.5. *Let $G(u) = \nabla_u L(a^*(u), u)$ for $u \in \mathcal{U}$. Then $G : \mathcal{U} \to \mathbb{R}^m$ is Lipschitz-continuous.*

Proof. See Section 6.B.2. □

Then, the Picard-Lindelöf theorem (see, e.g., Luk, 2017 for a self-contained presentation and Arnold, 1992 for a textbook) guarantees, for any initialization $u(0) \in \mathcal{U}$, the existence and uniqueness of a maximal solution of (6.8) taking values in $\mathcal{U}$. This solution is defined on a maximal interval $[0, T_{\max})$ where it could be that $T_{\max} < \infty$ if u hits the boundary of $\mathcal{U}$. However, the results below show that the target function f^* is recovered before this happens (with high probability over the initialization), and thus that this notion of solution is sufficient for our purposes. To this aim, we first define some sufficient conditions that the initialization should satisfy.

Definition 6.6. *Let D be a positive real. We say that a vector of positions $u \in [0,1]^m$ is D-good if*

(a) for all $i \in \{0, \dots, n-1\}$, there are at least 6 positions u_j in each interval $[v_i, v_{i+1}]$,

(b) $\Delta(u) \geqslant D$, and

(c) for all $i \in \{1, \dots, n-1\}$, denoting u_i^{L} the position closest to the left of v_i and u_i^{R} the position closest to the right, we have $|u_i^{\mathrm{R}} + u_i^{\mathrm{L}} - 2v_i| \geqslant D$.

Condition (a) is related to the fact that the derivation in Section 6.4.2 is valid only if there are at least two neurons per piece. Condition (b) indicates that the neurons have to be sufficiently spaced at initialization, which is not surprising since we have to guarantee that $\Delta(u(\tau)) > 2\eta$, that is, $u(\tau) \in \mathcal{U}$, for all τ until the recovery of f^* happens. Finally, condition (c) also helps to control the distance between neurons: although u_i^{L} and u_i^{R} move towards each other, as shown by (6.7), their distance can be controlled throughout the dynamics as a function of $|u_i^{\mathrm{R}} + u_i^{\mathrm{L}} - 2v_i|$.

We can now state the Proposition showing the recovery in finite time. The proof resembles the sketch of Section 6.4.2 with additional technical details since we need to control the distance between neurons, and the fact that $\eta > 0$ makes the dynamics more delicate to describe.

Proposition 6.7. *Let $f^* \in \mathcal{F}_{n,\Delta v,\Delta f,M}$. Assume that the initialization $u(0)$ is D-good with $D = 2^{13/2}(m+1)^{1/2}M\eta^{1/2}(\Delta f)^{-1}$. Then the maximal solution of (6.8) taking values in $\mathcal{U}$ is defined at least on $[0, \mathcal{T}]$ for $\mathcal{T} = 6/(\Delta f)^2$, and at the end of this time interval, there is a neuron at distance less than η from each discontinuity of f^*.*

Proof. See Section 6.B.3. □

This Proposition is the main building block to show recovery in the next Theorem, along with some high-probability bounds to ensure that an i.i.d. uniform initialization is D-good.

Theorem 6.8. *Let $\xi, \delta > 0$, and f^* a piecewise constant function from $\mathcal{F}_{n,\Delta v,\Delta f,M}$. Assume that the neural network has m neurons with*

$$m \geqslant \frac{6}{\Delta v}\big(4 + \log n + \log \frac{1}{\delta}\big)\,.$$

Assume that, at initialization, the positions $u_1, \dots, u_m$ of the neurons are i.i.d. uniformly distributed on $[0,1]$. Then there exists Q depending on $\xi, \delta, m, \Delta f, M$ such that, if

$$\eta \leqslant Q\,,$$

then, with probability at least $1-\delta$, the maximal solution to the two-timescale limit (6.8) is defined at least until $\mathcal{T} = \frac{6}{(\Delta f)^2}$, and

$$\int_0^1 |f^*(x) - f(x; a^*(u(\mathcal{T})), u(\mathcal{T}))|^2 \mathrm{d}x \leqslant \xi\,.$$

Furthermore, we have $Q = \dfrac{C}{M^2}\min\Big(\dfrac{\delta^2(\Delta f)^2}{(m+1)^5}, \dfrac{\xi}{n}\Big)$ for some universal constant $C > 0$.

Proof. See Section 6.B.4. □

6.5.2 From the two-timescale limit to the two-timescale regime

We now briefly explain how the proof for the two-timescale limit can be adapted for the gradient flow problem in the two-timescale regime (6.1), that is with a small but non-vanishing ε. First note that the existence and uniqueness of the maximal solution to the dynamics (6.1) follow from the local Lipschitz-continuity of $\nabla_a L$ and $\nabla_u L$ with respect to both their variables.

The heuristics of Section 6.4.1 indicate that, for ε small enough, at any time t, the weights $a(t)$ are close to $a^*(u(t))$, the global minimizer of $L(\cdot, u(t))$. The next Proposition formalizes this intuition.

Proposition 6.9. *Assume that $a(0) = 0$ and that, for all $s \in [0,t]$, there at least 2 positions $u_j(s)$ in each interval $[v_i, v_{i+1}]$ and $\Delta(u(s)) \geqslant D/2$ for some $D \geq 32\eta$. Finally, assume that $\varepsilon \leqslant 2^{-16} D^2 M^{-2} (m+1)^{-5/2}$. Then*

$$\|a(t) - a^*(u(t))\| \leqslant 3M\sqrt{m+1}\exp^{-\frac{D}{16}t} + \frac{2^{17}M^3(m+1)^3}{D^2}\varepsilon\,.$$

Proof. See Section 6.B.5. □

The crucial condition in the Proposition is $\Delta(u(s)) \geq D/2$; it is useful to control the conditioning of the quadratic form $L(\cdot, u(s))$. The Proposition shows that $\|a(t) - a^*(u(t))\|$ is upper bounded by the sum of two terms; the first term is a consequence of the initial gap between $a(0)$ and $a^*(u(0))$ and decays exponentially quickly. The second term is negligible in the regime $\varepsilon \ll 1$.

Armed with this Proposition, we show that the two-timescale regime has the same behavior as the two-timescale limit and thereby prove Theorem 6.2.

6.6 Numerical experiments

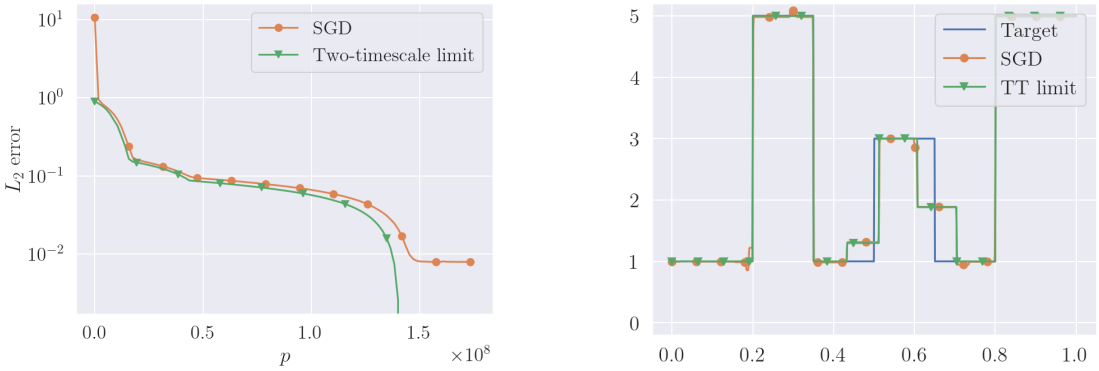
We place ourselves in the setting of Section 6.2. We first compare the dynamics of the gradient flow in the two-timescale limit presented in Section 6.4.2 with the dynamics of SGD. To simulate the SGD dynamics, we assume that we have access to noisy observations of the value of $f^* \in \mathcal{F}_{n,\Delta v,\Delta f,M}$: let $(X_p, Y_p)_{p\geqslant 1}$ be i.i.d. random variables such that X_p is uniformly distributed on $[0,1]$, and $Y_p = f^*(X_p) + N_p$ where N_p is additive noise. The (one-pass) SGD updates are then given by

$$\begin{aligned} a_{p+1} &= a_p - h\nabla_a \ell(X_{p+1}, Y_{p+1}; a_p, u_p)\,, \\ u_{p+1} &= u_p - \varepsilon h \nabla_u \ell(X_{p+1}, Y_{p+1}; a_p, u_p)\,, \end{aligned} \tag{6.9}$$

with $\ell(X, Y; a, u) = \frac{1}{2}(Y - f(X; a, u))^2$. The experimental settings, as well as additional results, are given in the Appendix.

Remarkably, the dynamics of SGD in the two-timescale regime with η small match closely the gradient flow in the two-timescale limit with $\eta = 0$, as illustrated in Figure 6.3. This validates the use of the gradient flow to understand the training dynamics with SGD. Both dynamics are close until the two-timescale limit achieves perfect recovery of the target function, at which point the SGD stabilizes to a small non-zero error. The fact that SGD does not achieve perfect recovery is not surprising, since SGD is performed with $\eta > 0$ and f^* is not in the span of $\{1, \sigma_\eta(x - u_1), \ldots, \sigma_\eta(x - u_m)\}$ for any $u_1, \ldots, u_m$ and for $\eta > 0$. On the contrary, we simulated the dynamics of gradient flow for $\eta = 0$, as presented in Section 6.4.2, enabling perfect recovery in that case.

Next, we compare the SGD dynamics in the two-timescale regime ($\varepsilon \ll 1$) and outside this regime ($\varepsilon \approx 1$). In Figure 6.4, we see that the network trained by SGD (in orange) in the

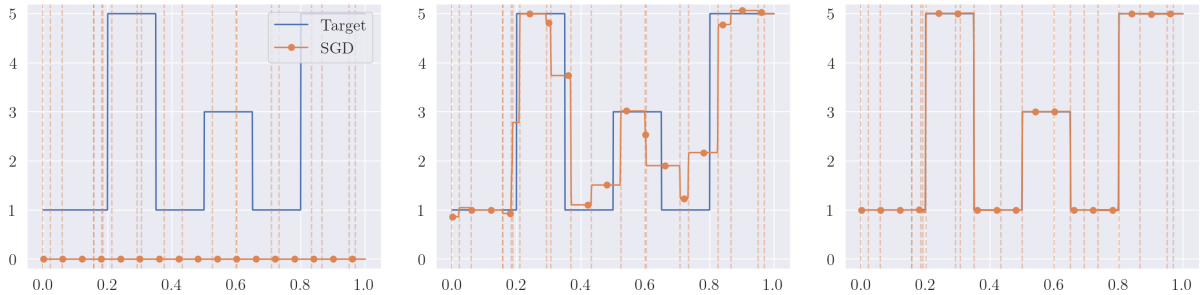


(a) Loss L as a function of the number of steps p (b) Plot of the functions after $p = 2.7 \cdot 10^7$ steps

Figure 6.3: Comparison between the SGD (6.9) with $\eta = 4 \cdot 10^{-3}$ in the two-timescale regime ($\varepsilon = 2 \cdot 10^{-5}$) and the gradient flow in the two-timescale limit (6.5) with $\eta = 0$. In the left-hand plot, to align the SGD and the two-timescale limit, we take $\tau = \varepsilon h p$. In the right-hand plot, the target function is in blue, the gradient flow in the two-timescale limit is in green, and the SGD is in orange.

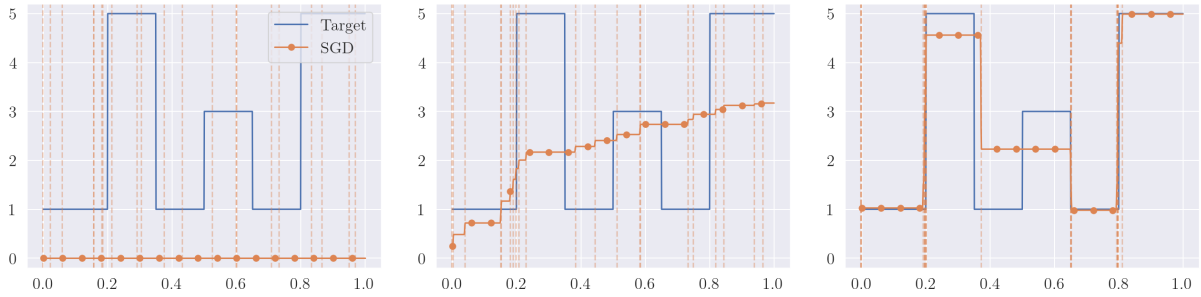
two-timescale regime $\varepsilon = 2 \cdot 10^{-5}$, achieves near-perfect recovery. If we change ε to 1, while keeping all other parameters equal, the algorithm fails to recover the target function (Figure 6.5). This shows that, in our setting with a moderate number of neurons m, recovery can fail away from the two-timescale regime.

Note that the dynamics of the neurons in Figures 6.4 and 6.5 are different. In the two-timescale regime, only the neurons closest to a discontinuity move significantly, while the others do not. These dynamics correspond to the sketch of Section 6.4. Interestingly, it means that in this regime, the neural network learns a sparse representation of the target function, meaning that only n out of the m neurons are active after training. On the contrary, when $\varepsilon = 1$, all neurons move to align with discontinuities of the target function, thus the learned representation is not sparse. Furthermore, since the number of neurons is moderate, one of the discontinuities is left without any neuron.



(a) At initialization $p = 0$ (b) After $p = 5.4 \cdot 10^6$ steps (c) After $p = 1.8 \cdot 10^8$ steps

Figure 6.4: Simulation in the two-timescale regime ($\varepsilon = 2 \cdot 10^{-5}$). The target function is in blue and the SGD (6.9) is in orange with $\eta = 4 \cdot 10^{-3}$, $h = 10^{-5}$. The positions $u_1, \dots, u_m$ of the neurons are indicated with vertical dotted lines. In a first short phase, only the weights $a_1, \dots, a_m$ of the neurons evolve to match as good as possible the target function (second plot). Then, in a longer phase, the neuron closest to each target discontinuity moves towards it (third plot). Recovery is achieved.



(a) At initialization $p = 0$ (b) After $p = 10^4$ steps (c) After $p = 10^6$ steps

Figure 6.5: Simulation outside the two-timescale regime ($\varepsilon = 1$). The target function is in blue and the SGD (6.9) is in orange with $\eta = 4 \cdot 10^{-3}$, $h = 10^{-5}$. The positions $u_1, \dots, u_m$ of the neurons are indicated with vertical dotted lines. The dynamics create a zone with no neuron, hindering recovery.

6.7 Conclusion

The two-timescale regime decouples the dynamics of the two layers of the neural network. As a consequence, it is a useful theoretical tool to simplify the evolution equations. In this chapter showcasing the approach, the two-timescale regime enables to show the alignment of the neurons with the discontinuities of the target function, and thus to prove recovery. We leave the exploration of the benefits of the two-timescale regime in other settings for future work. It would be interesting to prove theoretical results for SGD dynamics, in multivariate settings, or for other types of non-linearities, in particular the recovery of continuous piecewise affine functions by ReLU networks.

6.A Additional notations and technical lemmas

For a vector a, we denote $\|a\|$ its ℓ^2-norm, $\|a\|_1$ its ℓ^1-norm and $\|a\|_\infty$ its ℓ^∞-norm. For matrices H, $\|H\|$ denotes the operator norm associated to the ℓ^2 norm and $\|H\|_{\mathrm{F}}$ denotes the Frobenius norm. Finally, for real-valued functions f, $\|f\|_\infty$ denotes the supremum norm.

In all the appendix of this chapter, we denote $u_0 = -\eta/2$ and $u_{m+1} = 1 + \eta/2$. Note that $\sigma_\eta(x - u_0) = 1$ for all $x \in [0, 1]$, meaning that $\sigma_\eta(\cdot - u_0)$ corresponds to the bias term. This notation allows to treat the bias term in a unified fashion with respect to the other terms of $f(x; a, u)$. Since $u_i \in (0, 1)$ for $i \in \{1, \dots, m\}$, we assume in the following w.l.o.g. that the $(u_i)_{0 \leqslant i \leqslant m+1}$ are ordered in increasing order. Note that we prove in the following that the $(u_i)_{1 \leqslant i \leqslant m}$ do not cross during the dynamics, so they remain ordered throughout the dynamics.

The proofs involve comparisons of some quantities when $\eta > 0$ and when $\eta = 0$. To avoid confusion, we make explicit the dependency of L on $\eta \geqslant 0$, i.e., we let $L_\eta(a, u)$ in place of $L(a, u)$ of the main part of the chapter, and similarly, when the argmin is well-defined and unique,

$$a^*_\eta(u) = \operatorname*{argmin}_{a \in \mathbb{R}^{m+1}} L_\eta(a, u)\,,$$

in place of $a^*(u)$. Similarly, we now make explicit the dependence of f on $\eta \geqslant 0$, i.e., we denote

$$f_\eta(x; a, u) = a_0 + \sum_{j=1}^{m} a_j \sigma_\eta(x - u_j) = \sum_{j=0}^{m} a_j \sigma_\eta(x - u_j)\,.$$

The Hessian of the quadratic function $L_\eta(\cdot, u)$ is denoted $H_\eta(u) \in \mathbb{R}^{(m+1)\times(m+1)}$ (in place of $H(u)$), and satisfies that, for $i, j \in \{0, \dots, m\}$,

$$H_{\eta,ij}(u) = \int_0^1 \sigma_\eta(x - u_i)\sigma_\eta(x - u_j)\mathrm{d}x\,.$$

Also let, for $\eta \geqslant 0$ and $u \in \mathbb{R}^m$, $b_\eta(u) \in \mathbb{R}^{m+1}$ such that, for $j \in \{0, \dots, m\}$,

$$b_{\eta,j}(u) = \int_0^1 f^*(x)\sigma_\eta(x - u_j)\mathrm{d}x\,.$$

Finally, we let $\mathcal{U}_\eta$ in place of $\mathcal{U}$ in the main part of the chapter.

With these notations, we have, for $\eta \geqslant 0$ and $a, u \in \mathbb{R}^m$,

$$\begin{aligned}\frac{\partial L_\eta}{\partial u_j}(a, u) &= \int_0^1 \frac{\partial f_\eta(x; a, u)}{\partial u_j}\left(f_\eta(x; a, u) - f^*(x)\right)\mathrm{d}x \\ &= -a_j \int_0^1 \sigma_\eta'(x - u_j)\Big(\sum_{k=0}^m a_k\sigma_\eta(x - u_k) - f^*(x)\Big)\mathrm{d}x\,. \end{aligned} \tag{6.10}$$

and

$$\begin{aligned}\frac{\partial L_\eta}{\partial a_j}(a, u) &= \int_0^1 \frac{\partial f_\eta(x; a, u)}{\partial a_j}\left(f_\eta(x; a, u) - f^*(x)\right)\mathrm{d}x \\ &= \int_0^1 \sigma_\eta(x - u_j)\Big(\sum_{k=0}^m a_k\sigma_\eta(x - u_k) - f^*(x)\Big)\mathrm{d}x \\ &= H_{\eta,j}(u)^\top a - b_{\eta,j}(u)\,. \end{aligned} \tag{6.11}$$

We now move on to a series to lemmas that will be helpful in the proofs of Appendix 6.B.

Lemma 6.10. *For $\eta \geqslant 0$ and $u \in \mathbb{R}^m$, we have*

$$\|b_\eta(u) - b_0(u)\| \leqslant M\eta\sqrt{m+1} \quad \text{and} \quad \|b_\eta(u)\| \leqslant M\sqrt{m+1}\,.$$

Proof. For any $j \in \{0, \dots, m\}$,

$$\begin{aligned}|b_{\eta,j}(u) - b_{0,j}(u)| &= \Big|\int_0^1 f^*(x)(\sigma_\eta(x - u_j) - \sigma_0(x - u_j))\mathrm{d}x\Big| \\ &\leqslant \|f^*\|_\infty \int_0^1 |\sigma_\eta(x - u_j) - \sigma_0(x - u_j)|\,\mathrm{d}x \\ &\leqslant M\eta\,, \end{aligned}$$

where in the last step we use that $\|f^*\|_\infty \leqslant M$ and that $\sigma_\eta(x) = 0$ for $x \leqslant -\eta/2$, $\sigma_\eta(x) \in [0, 1]$ for $-\eta/2 < x < \eta/2$ and $\sigma_\eta(x) = 1$ for $x \geqslant \eta/2$.

Similarly,

$$|b_{\eta,j}(u)| = \Big|\int_0^1 f^*(x)\sigma_\eta(x - u_j)\mathrm{d}x\Big| \leqslant \|f^*\|_\infty \leqslant M\,.$$

□

Lemma 6.11. *For $\eta \geq 0$ and $u \in \mathcal{U}_\eta$, $H_\eta(u) = H_0(u) + D_\eta$, where D_η is a diagonal matrix whose elements are independent of u and bounded in absolute value by $\eta/2$.*

Proof. Let $i, j \in \{0, \dots, m\}$, and denote $c = \max(u_i, u_j, 0)$. Then

$$H_{0,ij}(u) = \int_0^1 \sigma_0(x - u_i)\sigma_0(x - u_j)dx = 1 - c\,.$$

If $i = j = 0$, $\max(u_i, u_j) = -\eta/2$, and $H_{\eta,ij}(u) = 1 = H_{0,ij}(u)$. If $i = j \neq 0$,

$$\begin{aligned} H_{\eta,ij}(u) &= \int_0^1 \sigma_\eta(x - c)^2 dx \\ &= 1 - c - \frac{\eta}{2} + \int_{c-\eta/2}^{c+\eta/2} \sigma_\eta(x - c)^2 dx \\ &= H_{0,ij}(u) - \frac{\eta}{2} + \eta \int_{-1/2}^{1/2} \sigma^2\,. \end{aligned}$$

Note that the last integral is non-negative and less than 1, hence $|H_{\eta,ij}(u) - H_{0,ij}(u)| \leq \eta/2$. Finally, if $i \neq j$, since $|u_i - u_j| > \eta$,

$$H_{\eta,ij}(u) = \int_0^1 \sigma_\eta(x - u_i)\sigma_\eta(x - u_j)dx = \int_0^1 \sigma_\eta(x - \max(u_i, u_j))dx\,.$$

Furthermore, $0 < \max(u_i, u_j) < 1 - \frac{\eta}{2}$, thus

$$H_{\eta,ij}(u) = \int_0^1 \sigma_\eta(x - c)dx = 1 - c - \frac{\eta}{2} + \int_{c-\eta/2}^{c+\eta/2} \sigma_\eta(x - c)dx = 1 - c\,,$$

where the last equality comes from the oddness of $\sigma - 1/2$. □

Lemma 6.12. *For $\eta > 0$, let $a_\eta^* : u \in \mathcal{U}_\eta \mapsto a_\eta^*(u)$. Then a_η^* is differentiable and for any $u \in \mathcal{U}_\eta$,*

$$\Big\| \frac{\partial a_\eta^*(u)}{\partial u} \Big\| \leqslant \frac{8}{\Delta(u)} \Big(2(m+1)\|a_\eta^*(u)\| + M \Big)\,.$$

Proof. By Proposition 6.4 (whose proof does not rely on this lemma), for $u \in \mathcal{U}_\eta$, $L_\eta(\cdot, u)$ has a unique minimizer $a_\eta^*(u)$, which is equal to $H_\eta(u)^{-1} b_\eta(u)$ by (6.11). Furthermore, H_η and b_η are differentiable with respect to u, hence a_η^* is also differentiable with respect to u, and we have

$$\frac{\partial a_\eta^*(u)}{\partial u_k} = -H_\eta(u)^{-1} \frac{\partial H_\eta}{\partial u_k}(u) a_\eta^*(u) + H_\eta(u)^{-1} \frac{\partial b_\eta}{\partial u_k}(u)\,.$$

Denote $w_k(u) := \frac{\partial H_\eta}{\partial u_k}(u) a_\eta^*(u)$ and $W(u)$ the $(m+1) \times (m+1)$ matrix formed by stacking column-wise the vectors $(w_k(u))_{0 \leqslant k \leqslant m}$. Then

$$\frac{\partial a_\eta^*(u)}{\partial u} = -H_\eta(u)^{-1} W(u) + H_\eta(u)^{-1} \frac{\partial b_\eta}{\partial u}(u)\,.$$

We now estimate the Frobenius norm of the matrix $W(u)$. By Lemma 6.11, for $u \in \mathcal{U}_\eta$, $H_\eta(u) = H_0(u) + D_\eta$. Take $i, j \in \{0, \dots, m\}$, then

$$H_{\eta,ij}(u) = H_{0,ij}(u) + D_{\eta,ij} = \int_0^1 \sigma_0(x - u_i)\sigma_0(x - u_j)dx + D_{\eta,ij} = 1 - \max(u_i, u_j, 0) + D_{\eta,ij}\,.$$

Hence $\frac{\partial H_{\eta,ij}}{\partial u_k} = 0$ if $i, j \neq k$. Further, if $i = k$ and $j \neq k$,

$$\Big| \frac{\partial H_{\eta,ij}}{\partial u_k}(u) \Big| = \Big| \frac{\partial}{\partial u_i}(1 - \max(u_i, u_j)) \Big| \leqslant 1\,.$$

Of course, the bound $|\frac{\partial H_{\eta,ij}}{\partial u_k}(u)| \leqslant 1$ also holds when $j = k$ and $i \neq j$. Finally, a similar bound shows that $|\frac{\partial H_{\eta,ij}}{\partial u_k}(u)| \leqslant 2$ when $i = j = k$.

As a consequence, for $k, i \in \{0, \dots, m\}$,

$$|w_{k,i}(u)| \leqslant \sum_{j=0}^{m} \Big|\frac{\partial H_{\eta,ij}}{\partial u_k}(u)\Big| |a^*_{\eta,j}(u)| \leq \begin{cases} |a^*_{\eta,k}(u)| & \text{if } i \neq k\,, \\ |a^*_{\eta,k}(u)| + \|a^*_\eta(u)\|_1 & \text{if } i = k\,. \end{cases}$$

Thus

$$\begin{aligned} \|W(u)\|_{\mathrm{F}} &= \Big(\sum_{i=0}^{m}\sum_{k=0}^{m} |w_{k,i}(u)|^2\Big)^{1/2} \leqslant \Big(\sum_{i=0}^{m}\Big(\sum_{k=0}^{m} |w_{k,i}(u)|\Big)^2\Big)^{1/2} \\ &\leqslant \Big(\sum_{i=0}^{m} \big(2\|a^*_\eta(u)\|_1\big)^2\Big)^{1/2} = 2\sqrt{m+1}\|a^*_\eta(u)\|_1\,. \end{aligned}$$

With a reasoning similar to the above, note that $\frac{\partial b_\eta}{\partial u}(u)$ is a diagonal matrix with diagonal entries in $[-M, M]$. Finally, putting these elements together, using Proposition 6.4 and that $\|W(u)\| \leqslant \|W(u)\|_{\mathrm{F}}$, we obtain

$$\Big\|\frac{\partial a^*_\eta(u)}{\partial u}\Big\| \leqslant \|H_\eta(u)^{-1}\|\|W(u)\|_{\mathrm{F}} + \|H_\eta(u)^{-1}\|\Big\|\frac{\partial b_\eta(u)}{\partial u}\Big\| \leqslant \frac{8}{\Delta(u)}\Big(2\sqrt{m+1}\|a^*_\eta(u)\|_1 + M\Big)\,.$$

□

The following lemma gives exact formulae for the derivative of the loss L_η with respect to the positions of the neurons, evaluated for $a = a^*_0(u)$, that is the best piecewise constant approximation of f^* with subdivision $\{u_1, \dots, u_m\}$. Note that the formulae are the same as in Section 6.4.2, but the derivation is slightly more intricate since we consider here the loss L_η and not L_0.

Lemma 6.13. *Take $\eta > 0$ and $u \in \mathcal{U}_\eta$ such that there are at least two neurons on each piece $[v_i, v_{i+1}]$ of f^*. Then, if u_j does not flank a discontinuity of f^*,*

$$\frac{\partial L_\eta}{\partial u_j}(a^*_0(u), u) = 0.$$

Furthermore, for a discontinuity v_i, denote u_i^{L} is the closest neuron to its left and u_i^{R} the closest neuron to its right. If $v_i - u_i^{\mathrm{L}} \geqslant \frac{\eta}{2}$ and $u_i^{\mathrm{R}} - v_i \geqslant \frac{\eta}{2}$, then

$$\begin{aligned} \frac{\partial L_\eta}{\partial u_i^{\mathrm{L}}}(a^*_0(u), u) &= -\frac{1}{2}\frac{(u_i^{\mathrm{R}} - v_i)^2}{(u_i^{\mathrm{R}} - u_i^{\mathrm{L}})^2}(f^*_i - f^*_{i-1})^2\,, \\ \frac{\partial L_\eta}{\partial u_i^{\mathrm{R}}}(a^*_0(u), u) &= \frac{1}{2}\frac{(v_i - u_i^{\mathrm{L}})^2}{(u_i^{\mathrm{R}} - u_i^{\mathrm{L}})^2}(f^*_i - f^*_{i-1})^2\,. \end{aligned}$$

Proof. In this proof, let us denote for simplicity $a = a^*_0(u)$. At the condition that there is at least two neurons on each piece of f^*, Section 6.4.2 gives the optimal approximation $f_0(x; a, u)$ of f^* that is piecewise constant with respect to the subdivision $\{u_1, \dots, u_m\}$. As a consequence, we easily get the value of a. Namely, if u_j does not flank a discontinuity of f^*, the value of $f_0(x; a, u)$ is locally constant around u_j, thus $a_j = 0$. Plugging into (6.10), we obtain

$$\frac{\partial L_\eta}{\partial u_j}(a, u) = 0\,.$$

Further, for a discontinuity v_i, denote respectively a_i^{L} and a_i^{R} the coefficients associated to u_i^{L} and u_i^{R}. At u_i^{L}, the value of $f_0(x;a,u)$ jumps from f_{i-1}^* to $\frac{v_i-u_i^{\mathrm{L}}}{u_i^{\mathrm{R}}-u_i^{\mathrm{L}}}f_{i-1}^* + \frac{u_i^{\mathrm{R}}-v_i}{u_i^{\mathrm{R}}-u_i^{\mathrm{L}}}f_i^*$, thus

$$a_i^{\mathrm{L}} = \frac{v_i-u_i^{\mathrm{L}}}{u_i^{\mathrm{R}}-u_i^{\mathrm{L}}}f_{i-1}^* + \frac{u_i^{\mathrm{R}}-v_i}{u_i^{\mathrm{R}}-u_i^{\mathrm{L}}}f_i^* - f_{i-1}^* = \frac{u_i^{\mathrm{R}}-v_i}{u_i^{\mathrm{R}}-u_i^{\mathrm{L}}}(f_i^*-f_{i-1}^*)\,.$$

Similarly, we have

$$a_i^{\mathrm{R}} = \frac{v_i-u_i^{\mathrm{L}}}{u_i^{\mathrm{R}}-u_i^{\mathrm{L}}}(f_i^*-f_{i-1}^*)\,.$$

We now compute, using (6.10),

$$\begin{aligned}\frac{\partial L_\eta}{\partial u_i^{\mathrm{L}}}(a,u) &= -a_i^{\mathrm{L}}\int_0^1 \sigma_\eta'(x-u_i^{\mathrm{L}})\,(f_\eta(x;a,u)-f^*(x))\,\mathrm{d}x \\ &= -a_i^{\mathrm{L}}\int_{u_i^{\mathrm{L}}-\eta/2}^{u_i^{\mathrm{L}}+\eta/2} \sigma_\eta'(x-u_i^{\mathrm{L}})\,(f_\eta(x;a,u)-f^*(x))\,\mathrm{d}x\,.\end{aligned}$$

Using that $\Delta(u) > 2\eta$ and that there are at least two neurons on each piece of f^*, we have that $u_i^{\mathrm{L}} - v_{i-1} \geqslant 2\eta$. Since, in addition, by assumption, $v_i - u_i^{\mathrm{L}} \geqslant \frac{\eta}{2}$, we get that for $x \in \left[u_i^{\mathrm{L}}-\frac{\eta}{2}, u_i^{\mathrm{L}}+\frac{\eta}{2}\right]$, $f^*(x) = f_{i-1}^*$. Moreover, using again $\Delta(u) \geqslant 2\eta$ that σ_η is equal to σ_0 on $(-\infty,-\eta/2]$ and $[\eta/2,\infty)$, we have for $x \in \left[u_i^{\mathrm{L}}-\frac{\eta}{2}, u_i^{\mathrm{L}}+\frac{\eta}{2}\right]$,

$$f_\eta(x;a,u) = \sum_{k=0}^{m} a_k\sigma_\eta(x-u_k) = f_0\left(u_i^{\mathrm{L}}-\frac{\eta}{2};a,u\right) + a_i^{\mathrm{L}}\sigma_\eta(x-u_i^{\mathrm{L}}) = f_{i-1}^* + a_i^{\mathrm{L}}\sigma_\eta(x-u_i^{\mathrm{L}})\,.$$

Thus we obtain

$$\begin{aligned}\frac{\partial L_\eta}{\partial u_i^{\mathrm{L}}}(a,u) &= -a_i^{\mathrm{L}}\int_{u_i^{\mathrm{L}}-\eta/2}^{u_i^{\mathrm{L}}+\eta/2} \sigma_\eta'(x-u_i^{\mathrm{L}})a_i^{\mathrm{L}}\sigma_\eta(x-u_i^{\mathrm{L}})\mathrm{d}x \\ &= -\frac{(a_i^{\mathrm{L}})^2}{2}\Big(\sigma_\eta\big(\frac{\eta}{2}\big)^2 - \sigma_\eta\big(-\frac{\eta}{2}\big)^2\Big) \\ &= -\frac{(a_i^{\mathrm{L}})^2}{2} \\ &= -\frac{1}{2}\frac{(u_i^{\mathrm{R}}-v_i)^2}{(u_i^{\mathrm{R}}-u_i^{\mathrm{L}})^2}(f_i^*-f_{i-1}^*)^2\,.\end{aligned}$$

The computation of $\frac{\partial L_\eta}{\partial u_i^{\mathrm{R}}}(a,u)$ is similar.

□

Lemma 6.14. *Consider $\eta \geq 0$ and $u \in \mathcal{U}_\eta$ such that there are at least two neurons on each piece $[v_i, v_{i+1}]$ of f^*. Then, for all $x \in [0,1]$, $|f_\eta(x;a_0^*(u),u)| \leq M$.*

Proof. In the case where $\eta = 0$, the result easily follows from the expressions for $f_0(x;a_0^*(u),u)$ provided in Section 6.4.2. We now assume $\eta > 0$.

Denote $A_k^*(u) = \sum_{j=0}^k a_{0,j}^*(u)$ (with the convention $A_{-1}^*(u) = 0$). Recall the convention

$u_0 = -\eta/2$. We compute

$$\begin{aligned} f_\eta(x; a_0^*(u), u) &= \sum_{k=0}^{m} a_{0,k}^*(u)\sigma_\eta(x-u_k) \\ &= \sum_{k=0}^{m} \big(A_k^*(u) - A_{k-1}^*(u)\big)\,\sigma_\eta(x-u_k) \\ &= \sum_{k=0}^{m-1} A_k^*(u)\,(\sigma_\eta(x-u_k) - \sigma_\eta(x-u_{k+1})) + A_m^*(u)\sigma_\eta(x-u_m) \end{aligned}$$

Note that $A_k^*(u) = \lim_{x\to u_k+} f_0(x; a_0^*(u), u)$, and thus, from the case $\eta = 0$, we have $|A_k^*(u)| \leqslant M$. Moreover, σ_η is increasing and the u_k are in increasing order. We thus get

$$\begin{aligned} |f_\eta(x; a_0^*(u), u)| &\leqslant M\Big(\sum_{k=0}^{m-1} (\sigma_\eta(x-u_k) - \sigma_\eta(x-u_{k+1})) + \sigma_\eta(x-u_m)\Big) \\ &= M\sigma_\eta(x-u_0) \leqslant M\,. \end{aligned}$$

□

Lemma 6.15. *Consider $\eta > 0$ and $u \in \mathcal{U}_\eta$ such that there are at least two neurons on each piece $[v_i, v_{i+1}]$ of f^*. Then, for $j \in \{0, \dots, m\}$,*

$$|a_{0,j}^*(u)| \leqslant 2M$$

and, for any $a \in \mathbb{R}^{m+1}$,

$$\Big|\frac{\partial L_\eta}{\partial u_j}(a,u) - \frac{\partial L_\eta}{\partial u_j}(a_0^*(u),u)\Big| \leqslant 2M(\sqrt{m+1}+1)\|a - a_0^*(u)\| + \sqrt{m+1}\|a - a_0^*(u)\|^2\,.$$

Proof. The first statement of the Lemma comes from the explicit formulae for $a_0^*(u)$ given in the proof of Lemma 6.13, namely each $a_{0,j}^*(u)$ is either zero or less in magnitude than the gap between two pieces of f^* that is less than $2M$.

By (6.10), we have

$$\begin{aligned} &\Big|\frac{\partial L_\eta}{\partial u_j}(a,u) - \frac{\partial L_\eta}{\partial u_j}(a_0^*(u),u)\Big| \\ &\qquad = \Big| a_j \int_0^1 \sigma_\eta'(x-u_j)\Big(f_\eta(x;a,u) - f^*(x)\Big)\mathrm{d}x \\ &\qquad\qquad - a_{0,j}^*(u)\int_0^1 \sigma_\eta'(x-u_j)\Big(f_\eta(x;a_0^*(u),u) - f^*(x)\Big)\mathrm{d}x\Big| \\ &\qquad \leqslant |a_j - a_{0,j}^*(u)| \int_0^1 \sigma_\eta'(x-u_j)\,|f_\eta(x;a_0^*(u),u) - f^*(x)|\,\mathrm{d}x \\ &\qquad\qquad + |a_j| \int_0^1 \sigma_\eta'(x-u_j)\,|f_\eta(x;a,u) - f_\eta(x;a_0^*(u),u)|\,\mathrm{d}x\,. \end{aligned}$$

We bound the two terms separately. For the first term, we use Lemma 6.14.

$$\begin{aligned} \int_0^1 \sigma_\eta'(x-u_j)\,|f_\eta(x;a_0^*(u),u) - f^*(x)| &\leq \int_0^1 \sigma_\eta'(x-u_j)\,(|f_\eta(x;a_0^*(u),u)| + |f^*(x)|) \\ &\leqslant 2M\int_0^1 \sigma_\eta'(x-u_j)\mathrm{d}x \leqslant 2M\,. \end{aligned}$$

We now continue with the second term.

$$|f_\eta(x;a,u) - f_\eta(x;a_0^*(u),u)| = \Big|\sum_{k=0}^{m}(a_k - a_{0,k}^*(u))\sigma_\eta(x-u_k)\Big| \leqslant \|a - a_0^*(u)\|_1\,,$$

and thus

$$\begin{aligned}\int_0^1 \sigma_\eta'(x-u_j)\,|f_\eta(x;a,u) - f_\eta(x;a_0^*(u),u)|\,\mathrm{d}x &\leqslant \|a-a_0^*(u)\|_1 \int_0^1 \sigma_\eta'(x-u_j)\mathrm{d}x\\ &\leqslant \|a-a_0^*(u)\|_1\,.\end{aligned}$$

Returning to our initial upper bound, we obtain, using the first statement of the Lemma,

$$\begin{aligned}\Big|\frac{\partial L_\eta}{\partial u_j}(a,u) - \frac{\partial L_\eta}{\partial u_j}(a_0^*(u),u)\Big| &\leqslant 2M\|a-a_0^*(u)\| + (|a_{0,j}^*(u)| + |a_j - a_{0,j}^*(u)|)\|a-a_0^*(u)\|_1\\ &\leqslant 2M\|a-a_0^*(u)\| + (2M + \|a-a_0^*(u)\|)\sqrt{m+1}\|a-a_0^*(u)\|\\ &= 2M(\sqrt{m+1}+1)\|a-a_0^*(u)\| + \sqrt{m+1}\|a-a_0^*(u)\|^2\,.\end{aligned}$$

□

Lemma 6.16. *For $\eta \geq 0$ and $u \in \mathcal{U}_\eta$,*

$$\|a_\eta^*(u) - a_0^*(u)\| \leqslant \frac{16M\sqrt{m+1}\eta}{\Delta(u)}\,.$$

Proof. By (6.11),

$$H_\eta(u)a_\eta^*(u) = b_\eta(u)$$

and by (6.11) and by Lemma 6.11,

$$H_\eta(u)a_0^*(u) = H_0(u)a_0^*(u) + D_\eta a_0^*(u) = b_0(u) + D_\eta a_0^*(u).$$

According to Proposition 6.4 (whose proof does not rely on this lemma), $H_\eta(u)$ is invertible with $\|H_\eta(u)^{-1}\| \leqslant 8/\Delta(u)$. We thus have

$$\begin{aligned}\|a_\eta^*(u) - a_0^*(u)\| &= \|H_\eta(u)^{-1}(H_\eta(u)a_\eta^*(u) - H_\eta(u)a_0^*(u))\|\\ &\leq \frac{8}{\Delta(u)}\|b_\eta(u) - b_0(u) - D_\eta a_0^*(u)\|\\ &\leq \frac{8}{\Delta(u)}\big(\|b_\eta(u) - b_0(u)\| + \|D_\eta a_0^*(u)\|\big)\\ &\leq \frac{8}{\Delta(u)}\big(\|b_\eta(u) - b_0(u)\| + \frac{\eta}{2}\|a_0^*(u)\|\big)\,.\end{aligned}$$

The result then unfolds from Lemmas 6.10 and 6.15. □

Lemma 6.17. *Let $\eta > 0$, $u \in \mathbb{R}^m$ and $a, a' \in \mathbb{R}^{m+1}$. Then*

$$\begin{aligned}\|\nabla_u L_\eta(a,u)\| &\leqslant \sqrt{m+1}\|a\|^2 + M\|a\|\,,\\ \|\nabla_a L_\eta(a,u)\| &\leqslant \sqrt{m+1}(\|a\|\sqrt{m+1} + M)\,.\end{aligned}$$

As a consequence of the second inequality, by the fundamental theorem of calculus for line integrals,

$$|L_\eta(a,u) - L_\eta(a',u)| \leqslant \sqrt{m+1}\left(\max(\|a\|,\|a'\|)\sqrt{m+1} + M\right)\|a-a'\|\,.$$

Proof. Recall that, for all $j \in \{1, \dots, m\}$, and for all $a, u \in \mathbb{R}^m$,

$$\frac{\partial L_\eta}{\partial u_j}(a,u) = -a_j \int_0^1 \sigma_\eta'(x-u_j)\Big(\sum_{k=0}^m a_k \sigma_\eta(x-u_k) - f^*(x)\Big)\mathrm{d}x\,,$$
$$\frac{\partial L_\eta}{\partial a_j}(a,u) = \int_0^1 \sigma_\eta(x-u_j)\Big(\sum_{k=0}^m a_k \sigma_\eta(x-u_k) - f^*(x)\Big)\mathrm{d}x\,.$$

From the first equality, we have

$$\begin{aligned}\Big|\frac{\partial L_\eta}{\partial u_j}(a,u)\Big| &\leq |a_j| \int_0^1 |\sigma_\eta'(x-u_j)|\Big(\sum_{k=1}^m |a_k|\sigma_\eta(x-u_k) + |f^*(x)|\Big)dx \\ &\leqslant |a_j|(\|a\|_1 + M)\int_0^1 |\sigma_\eta'(x-u_j)|dx \\ &\leqslant |a_j|(\|a\|_1 + M)\,.\end{aligned}$$

As a consequence,

$$\|\nabla_u L_\eta(a,u)\| \leqslant \|a\|(\|a\|_1 + M) \leqslant \sqrt{m+1}\|a\|^2 + M\|a\|\,.$$

Similarly, from the second equality, we have

$$\left|\frac{\partial L_\eta}{\partial a_j}(a,u)\right| \leqslant \|a\|_1 + M\,.$$

As a consequence,

$$\|\nabla_a L_\eta(a,u)\| \leqslant \sqrt{m+1}(\|a\|_1 + M) = \sqrt{m+1}(\|a\|\sqrt{m+1} + M)\,.$$

□

Lemma 6.18. *Consider $\eta \geq 0$ and $u \in \mathcal{U}_\eta$ such that there is a neuron at distance less than η from each discontinuity of f^* and $3\eta \leq \Delta v$. Then*

$$\int_0^1 |f_\eta(x; a_\eta^*(u), u) - f^*(x)|^2 \mathrm{d}x \leqslant 6M^2\eta n\,.$$

Proof. By definition of $a_\eta^*(u)$,

$$\int_0^1 |f_\eta(x; a_\eta^*(u), u) - f^*(x)|^2 \mathrm{d}x = \min_{a \in \mathbb{R}^{m+1}} \int_0^1 |f_\eta(x; a, u) - f^*(x)|^2 \mathrm{d}x\,.$$

Thus it is enough to exhibit some a for which the latter integral is smaller than $6M^2\eta n$ to conclude.

We construct such an a as follows: set $a_0 = f^*(0)$, and for each discontinuity v_i, set the coefficient of a neuron at distance less than η to the value $f_i^* - f_{i-1}^*$ and set all other neurons to zero. Note that the active neurons are distinct since $3\eta \leq \Delta v$.

Then the neural network is equal to the target function everywhere except on an interval of size $3\eta/2$ around each discontinuity, where they disagree (in infinite norm) by at most $2M$.

□

Lemma 6.19. *Let m be a positive integer and $u_1, \dots, u_m$ be i.i.d. uniform random variables in $[0,1]$. Assume that*

$$m \geq \frac{6}{\Delta v}\Big(4 + \log n + \log\frac{1}{\delta}\Big)\,.$$

Then, with probability at least $1-\delta$, the vector u is D-good with $D = \frac{\delta}{6(m+1)^2}$.

Proof. We define the following events:

(a) A is the event "there are at least 6 positions u_j in each interval $[v_i, v_{i+1}]$ for $i \in \{0, \dots, n-1\}$",

(b) B is the event "$\Delta(u) \geq D$",

(c) for all $i \in \{1, \dots, n-1\}$, E_i is the event "there are at least one neuron on the left and on the right of v_i" and C_i is the event "E_i holds and $|u_i^{\mathrm{R}} + u_i^{\mathrm{L}} - 2v_i| \geq D$".

Note that by Definition 6.6, u is D-good if and only if the event $A \cap B \cap (\bigcap_i C_i)$ holds. To show that this holds with high probability, we bound the probability of the complement

$$\begin{aligned}\Big(A \cap B \cap \Big(\bigcap_i C_i\Big)\Big)^c &= A^c \cup B^c \cup \Big(\bigcup_i C_i^c\Big) = A^c \cup B^c \cup \Big(\bigcup_i (C_i^c \cap A)\Big) \\ &\subset A^c \cup B^c \cup \Big(\bigcup_i (C_i^c \cap E_i)\Big) \qquad\qquad (\text{as } A \subset E_i)\,.\end{aligned}$$

Thus

$$\mathbb{P}(u \text{ is not } D\text{-good}) \leq \mathbb{P}(A^c) + \mathbb{P}(B^c) + \sum_{i=1}^{n-1} \mathbb{P}(C_i^c \cap E_i)\,.$$

Below, we bound separately the three terms of the right-hand side.

(a) Denote $m' = \lfloor m/6 \rfloor$. For any $i \in \{0, \dots, n-1\}$, the set $\mathcal{A}_i = \{j \in \{1, \dots, m'\} \,|\, u_j \in [v_i, v_{i+1}]\}$ is empty with probability $(1 - (v_{i+1} - v_i))^{m'} \leqslant (1 - \Delta v)^{m'}$. Thus by the union bound, the probability that at least one of $\mathcal{A}_1, \dots, \mathcal{A}_n$ is empty is upper bounded by $n(1-\Delta v)^{m'}$.

We now check that $n(1-\Delta v)^{m'} \leqslant \delta/18$. Indeed,

$$m' = \Big\lfloor \frac{m}{6} \Big\rfloor \geq \frac{m}{6} - 1 \geq \frac{3 + \log n + \log\frac{1}{\delta}}{\Delta v} \geq \frac{\log n + \log\frac{18}{\delta}}{\Delta v} \geq -\frac{\log n + \log\frac{18}{\delta}}{\log(1-\Delta v)}\,,$$

where we use $\Delta v \leq 1$, $3 \geqslant \log(18)$, and $\log(1-\Delta v) \leq -\Delta v < 0$. This gives the desired inequality.

In other words, the probability that at least one of the intervals $[v_i, v_{i+1}]$ contains none of the $u_1, \dots, u_{m'}$ is bounded by $\delta/18$. As a consequence, by the union bound, the probability that at least one of the intervals $[v_i, v_{i+1}]$ contains strictly less than 6 of the $u_1, \dots, u_m$ is bounded by $\delta/3$, i.e., $\mathbb{P}(A^c) \leq \delta/3$.

(b) Recall that by convention, $u_0 = -\frac{\eta}{2}$ and $u_{m+1} = 1 + \frac{\eta}{2}$. For all $i \in \{0, \dots, m+1\}$, denote $I_i = (u_i - D, u_i + D)$. Denote F_j the event "$u_j \in I_i$ for some $i \in \{0, \dots, m+1\}$, $i \neq j$". Note that $B^c = \cup_{j=1}^m F_j$.

Fix $j = 1, \dots, m$. By conditioning on u_i for all $i \in \{0, \dots, m+1\}$, $i \neq j$, we see that $\mathbb{P}(F_j) \leq 2(m+1)D$. By the union bound,

$$\mathbb{P}(B^c) \leq 2m(m+1)D \leq \frac{\delta}{3}.$$

(c) Take $i \in \{1, \dots, n-1\}$. For convenience, we define the random variable u_i^{L} (resp. u_i^{R}) on the full probability space by setting $u_i^{\mathrm{L}} = 0$ (resp. $u_i^{\mathrm{R}} = 1$) when there is no neuron on the left (resp. the right) of v_i. We compute the joint cumulative distribution function of $(u_i^{\mathrm{L}}, u_i^{\mathrm{R}})$ (with a convenient change of inequality): for all $0 \le y \le v_i \le z \le 1$,

$$\mathbb{P}(u_i^{\mathrm{L}} \le y, u_i^{\mathrm{R}} \ge z) = \mathbb{P}(\forall j \in \{1, \dots, m\},\, u_j \notin [y, z]) = (1 - (z - y))^m \,.$$

We observe that the joint cumulative distribution function of $(u_i^{\mathrm{L}}, u_i^{\mathrm{R}})$ is a smooth function of (y, z) when $(y, z) \in (0, v_i) \times (v_i, 1)$. Note that the events E_i and $\{(u_i^{\mathrm{L}}, u_i^{\mathrm{R}}) \in (0, v_i) \times (v_i, 1)\}$ are equal up to a null set. Therefore, on this event, $(u_i^{\mathrm{L}}, u_i^{\mathrm{R}})$ is an absolutely continuous random variable with density $g : (0, v_i) \times (v_i, 1) \to \mathbb{R}$,

$$g(y, z) = -\frac{\partial^2}{\partial y \partial z} \mathbb{P}(u_i^{\mathrm{L}} \le y, u_i^{\mathrm{R}} \ge z) = m(m-1)\,(1 - (z - y))^{m-2} \,.$$

We compute

$$\begin{aligned}
\mathbb{P}(C_i^c \cap E_i) &= \mathbb{P}(\{|u_i^{\mathrm{R}} + u_i^{\mathrm{L}} - 2v_i| \le D\} \cap E_i) \\
&= \int_{\{0 < y < v_i < z < 1\}} m(m-1)\,(1 - (z - y))^{m-2}\, \mathbf{1}_{\{|y+z-2v_i| \le D\}} \mathrm{d}y \mathrm{d}z \,.
\end{aligned}$$

We make the change of variables $\theta = z - y$, $\nu = z + y$.

$$\begin{aligned}
\mathbb{P}(C_i^c \cap E_i) &= \frac{m(m-1)}{2} \int_{\{0 < \frac{\nu-\theta}{2} < v_i < \frac{\nu+\theta}{2} < 1\}} (1-\theta)^{m-2} \mathbf{1}_{|\nu - 2v_i| \le D} \mathrm{d}\theta \mathrm{d}\nu \\
&\le \frac{m(m-1)}{2} \Big(\int_0^1 (1-\theta)^{m-2} \mathrm{d}\theta \Big) \Big(\int_{-\infty}^{\infty} \mathbf{1}_{|\nu - 2v_i| \le D} \mathrm{d}\nu \Big) \\
&= Dm \,.
\end{aligned}$$

Using $m \ge 24/\Delta v \ge 24n$, we have

$$\sum_{i=1}^{n-1} \mathbb{P}(C_i^c \cap E_i) \le (n-1)Dm \le \frac{\delta}{24 \times 6} \le \frac{\delta}{3} \,.$$

This concludes the proof.

□

6.B Proofs of the results

6.B.1 Proof of Proposition 6.4

Let us lower-bound the smallest eigenvalue of $H_\eta(u)$ which is equal to

$$\min_{\|a\|=1} a^\top H_\eta(u) a \,.$$

Now for $a \in \mathbb{R}^{m+1}$ such that $\|a\| = 1$,

$$a^\top H_\eta(u) a = \sum_{i,j=0}^{m} a_i a_j \int_0^1 \sigma_\eta(x - u_i) \sigma_\eta(x - u_j) \mathrm{d}x = \int_0^1 \left(\sum_{i=0}^{m} a_i \sigma_\eta(x - u_i) \right)^2 \mathrm{d}x \,.$$

Since $\Delta u > 2\eta$ (because $u \in \mathcal{U}$) and $u_0 = -\eta/2$, $u_{m+1} = 1+\eta/2$, the intervals $[u_i+\eta/2, u_{i+1}-\eta/2]$ for $i \in \{0,\dots,m\}$ are disjoint and included in $[0,1]$. Thus

$$a^\top H_\eta(u) a \geqslant \sum_{i=0}^{m} \int_{u_i+\eta/2}^{u_{i+1}-\eta/2} \left(\sum_{i=0}^{m} a_i \sigma_\eta(x-u_i) \right)^2 \mathrm{d}x \,.$$

Since $\sigma(x) = 0$ if $x < -1/2$ and $\sigma(x) = 1$ if $x > 1/2$, we have that $\sigma_\eta(x) = 0$ if $x < -\eta/2$ and $\sigma_\eta(x) = 1$ if $x > \eta/2$. Further recall that the u_i are ordered in increasing order. As a consequence,

$$\begin{aligned} a^\top H_\eta(u) a &\geqslant \sum_{i=0}^{m} \int_{u_i+\eta/2}^{u_{i+1}-\eta/2} \left(\sum_{k=0}^{i} a_k \right)^2 \mathrm{d}x \\ &= \sum_{i=0}^{m} (u_{i+1} - u_i - \eta) \left(\sum_{k=0}^{i} a_k \right)^2 \\ &\geqslant \frac{\Delta(u)}{2} \sum_{i=0}^{m} \left(\sum_{k=0}^{i} a_k \right)^2 , \end{aligned} \tag{6.12}$$

where in the last step, we used that $\Delta(u) > 2\eta$ and thus $u_{i+1} - u_i - \eta \geqslant \Delta(u) - \eta \geqslant \Delta(u) - \Delta(u)/2 = \Delta(u)/2$. Now, denote $c_0 = 0$ and $c_i = \sum_{k=0}^{i-1} a_k$. Then $\|a\| = 1$ writes

$$\sum_{i=0}^{m} (c_{i+1} - c_i)^2 = 1.$$

Furthermore,

$$\sum_{i=0}^{m} (c_{i+1} - c_i)^2 = \sum_{i=0}^{m} c_{i+1}^2 + \sum_{i=0}^{m} c_i^2 - 2 \sum_{i=0}^{m} c_{i+1} c_i \leqslant 4 \sum_{i=0}^{m+1} c_i^2 \,.$$

Hence

$$\sum_{i=0}^{m+1} c_i^2 \geqslant \frac{1}{4} \,,$$

which shows in conjunction with (6.12) that the smallest eigenvalue of $H_\eta(u)$ is lower-bounded by $\frac{\Delta u}{8}$.

6.B.2 Proof of Proposition 6.5

To show that $G(u) = (\nabla_u L_\eta)(a_\eta^*(u), u)$ is Lipschitz-continuous on $\mathcal{U}_\eta$, we show that it is differentiable on $\mathcal{U}_\eta$ and that its derivatives are uniformly bounded. The chain rule gives

$$\frac{\partial G_j}{\partial u_k} = \sum_{l=0}^{m} \frac{\partial a_{\eta,l}^*}{\partial u_k}(u) \frac{\partial^2 L_\eta}{\partial u_j \partial a_l}(a_\eta^*(u), u) + \frac{\partial^2 L_\eta}{\partial u_j \partial u_k}(a_\eta^*(u), u) \,.$$

From (6.10), using that σ is twice continuously differentiable, it can be checked that $\frac{\partial L_\eta}{\partial u_j}$ is differentiable in both its arguments and its derivatives are uniformly upper-bounded when a is bounded. Furthermore, for $u \in \mathcal{U}_\eta$,

$$\|a_\eta^*(u)\| \leqslant \|H_\eta(u)^{-1}\| \, \|b_\eta(u)\| \leqslant \frac{8M\sqrt{m+1}}{\Delta(u)} \,,$$

by Lemma 6.10 and Proposition 6.4. Finally, according to Lemma 6.12, a_η^* is differentiable with derivatives uniformly upper-bounded on $\mathcal{U}_\eta$. This concludes the proof.

6.B.3 Proof of Proposition 6.7

In this proof, we denote $u_i^{\mathrm{L}}(\tau)$ (resp. $u_i^{\mathrm{R}}(\tau)$) the position at time τ of the neuron that is *at initialization* closest to v_i to the left (resp. the right). Note that because of the movement of the neurons, it could be that u_i^{L} (resp. u_i^{R}) does not remain the neuron closest to the left (resp. the right) throughout the dynamics. Our proof discusses when this phenomenon occurs. Similarly, denote u_i^{LL} (resp. u_i^{RR}) the neuron second closest to the left (resp. the right) of v_i. Since the initialization is D-good, note that all these neurons are distinct.

Denote $\overline{\mathcal{T}}$ the minimal time $\tau \in [0, \mathcal{T}_{\max})$ such that $\Delta(u(\tau)) \leqslant D/2$ or there are less than two neurons in some piece $[v_i, v_{i+1}]$ of f^*. Note that by assumption, $\Delta(u(0)) \geqslant D > D/2$ and there are at least 6 neurons in each interval at initialization, thus $\overline{\mathcal{T}} > 0$. Furthermore, using the trivial inequalities $M \geqslant \Delta f/2$, $m+1 \geqslant 1$ and $\eta^{1/2} \geqslant \eta$, we have $\frac{D}{2} = \frac{2^{11/2} M\sqrt{m+1}\sqrt{\eta}}{\Delta f} \geqslant 8\eta > 2\eta$. Recall that 2η is the quantity defining the set $\mathcal{U}_\eta$ supporting the maximal solution of the equation (6.8). As a consequence, we do have $\overline{\mathcal{T}} < \mathcal{T}_{\max}$. At the end of the proof, we check that $\mathcal{T} < \overline{\mathcal{T}}$, by controlling carefully the movement of each neuron.

Let us first bound the difference between the dynamics of u and the dynamics that we would have if at each time τ, the weights a were given by $a_0^*(u(\tau))$, the best approximation of f^* by a piecewise constant function with subdivision $u(\tau)$. For any $\tau < \overline{\mathcal{T}}$ and $j \in \{1, \dots, m\}$, by Lemma 6.15, we have

$$\begin{aligned}
&\Big| \frac{\mathrm{d}u_j}{\mathrm{d}\tau}(\tau) + \frac{\partial L_\eta}{\partial u_j}(a_0^*(u(\tau)), u(\tau)) \Big| \\
&\quad = \Big| \frac{\partial L_\eta}{\partial u_j}(a_\eta^*(u(\tau)), u(\tau)) - \frac{\partial L_\eta}{\partial u_j}(a_0^*(u(\tau)), u(\tau)) \Big| \\
&\quad \leqslant 2M(\sqrt{m+1}+1)\|a_\eta^*(u(\tau)) - a_0^*(u(\tau))\| + \sqrt{m+1}\|a_\eta^*(u(\tau)) - a_0^*(u(\tau))\|^2 \,. \qquad (6.13)
\end{aligned}$$

We are therefore led to bounding $\|a_\eta^*(u(\tau)) - a_0^*(u(\tau))\|$, as follows:

$$\begin{aligned}
\|a_\eta^*(u(\tau)) - a_0^*(u(\tau))\| &\leqslant \frac{2^4 M\sqrt{m+1}\eta}{\Delta(u(\tau))} && \text{(by Lemma 6.16)} \\
&\leqslant \frac{2^5 M\sqrt{m+1}\eta}{D} && \text{(since } \Delta(u(\tau)) \geqslant D/2\text{)} \\
&= \frac{D(\Delta f)^2}{2^8 M\sqrt{m+1}} && \text{(by definition of } D\text{)}.
\end{aligned}$$

Then the first term in (6.13) is less than

$$\frac{(\sqrt{m+1}+1)D(\Delta f)^2}{2^7\sqrt{m+1}} \leqslant \frac{D(\Delta f)^2}{2^6} \,,$$

and the second term in (6.13) is less than

$$\frac{D^2(\Delta f)^4}{2^{16}M^2\sqrt{m+1}} \leqslant \frac{D(\Delta f)^2}{2^{14}} \,, \qquad \text{using } D \leqslant \Delta(u(0)) \leqslant 1,\ \Delta f \leqslant 2M \text{ and } m+1 \geqslant 1.$$

Hence we obtain, for any $\tau < \overline{\mathcal{T}}$ and $j \in \{1, \dots, m\}$,

$$\Big| \frac{\mathrm{d}u_j}{\mathrm{d}\tau}(\tau) + \frac{\partial L_\eta}{\partial u_j}(a_0^*(u(\tau)), u(\tau)) \Big| \leqslant \frac{D(\Delta f)^2}{60} =: \Delta g \qquad (6.14)$$

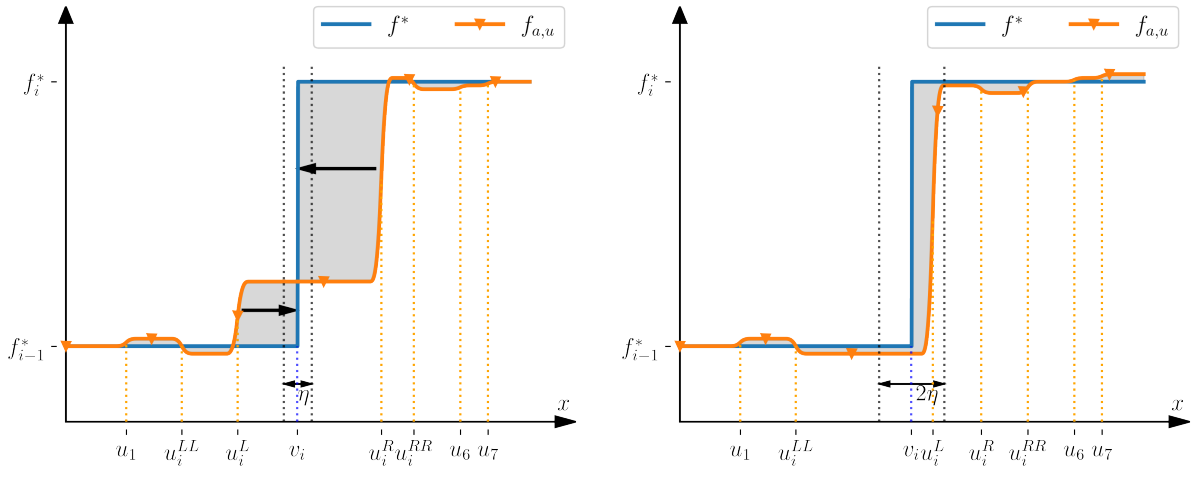
Now, let us examine how the neurons move, by leveraging Lemma 6.13 that gives exact formulae for $\frac{\partial L_\eta}{\partial u_j}(a_0^*(u(\tau)), u(\tau))$. First, if u_j is not next to a discontinuity, $\frac{\partial L_\eta}{\partial u_j}(a_0^*(u(\tau)), u(\tau)) = 0$, hence

$$|u_j(\tau) - u_j(0)| \leqslant (\Delta g)\tau \,.$$

Let us now study what happens next to a discontinuity v_i. Denote $(\delta f)_i = f_i^* - f_{i-1}^*$. W.l.o.g., assume that

$$u_i^{\mathrm{R}}(0) - v_i > v_i - u_i^{\mathrm{L}}(0)\,.$$

In the reverse case, the proof is the same by swapping the roles of u_i^{L} and u_i^{R}, and of u_i^{LL} and u_i^{RR}. We are going to show that the dynamics of u_i^{L} are divided into two phases. Define $\mathcal{T}_i$ as the minimal $\tau \in [0, \overline{\mathcal{T}}]$ such that $u_i^{\mathrm{L}}(\tau) = v_i - \eta/2$. In the first phase $[0, \mathcal{T}_i]$, we have $u_i^{\mathrm{L}}(\tau) < v_i - \frac{\eta}{2}$ and u_i^{L} moves towards v_i. In the second phase $[\mathcal{T}_i, \overline{\mathcal{T}}]$, we show below that $u_i^{\mathrm{L}}(\tau) \in [v_i - \eta, v_i + \eta]$. Note that we can have $\mathcal{T}_i = 0$ if $u_i^{\mathrm{L}}(0) \geqslant v_i - \frac{\eta}{2}$. It is also possible that $\mathcal{T}_i = \infty$ a priori; this means that the second phase does not exist. We show below that this case does not happen. Figure 6.6 depicts the two phases.



(a) Phase 1 (b) Phase 2

Figure 6.6: Dynamics of the neurons next to a discontinuity v_i. In the first phase, u_i^{L} and u_i^{R} move towards v_i, until the closest neuron (in this case u_i^{L}) reaches the interval of size η centered in v_i. In the second phase, u_i^{L} remains in an interval of size 2η around v_i, and none of the neurons move significantly.

Beginning by the first phase, we have, while $u_i^{\mathrm{L}}(\tau) < v_i - \frac{\eta}{2}$ and $u_i^{\mathrm{R}}(\tau) > v_i + \frac{\eta}{2}$, according to Lemma 6.13,

$$\begin{aligned}
\frac{\partial L_\eta}{\partial u_i^{\mathrm{L}}}(a_0^*(u(\tau)), u(\tau)) &= -\frac{1}{2}\frac{(u_i^{\mathrm{R}}(\tau) - v_i)^2(\delta f)_i^2}{(u_i^{\mathrm{R}}(\tau) - u_i^{\mathrm{L}}(\tau))^2}\,,\\
\frac{\partial L_\eta}{\partial u_i^{\mathrm{R}}}(a_0^*(u(\tau)), u(\tau)) &= \frac{1}{2}\frac{(v_i - u_i^{\mathrm{L}}(\tau))^2(\delta f)_i^2}{(u_i^{\mathrm{R}}(\tau) - u_i^{\mathrm{L}}(\tau))^2}\,.
\end{aligned}$$

For ease of computation, let $d_i^{\mathrm{L}}(\tau) = v_i - u_i^{\mathrm{L}}(\tau)$ and $d_i^{\mathrm{R}}(\tau) = u_i^{\mathrm{R}}(\tau) - v_i$ be the distances between the neurons and v_i. Then, by (6.14),

$$\begin{aligned}
\frac{\mathrm{d}d_i^{\mathrm{R}}}{\mathrm{d}\tau}(\tau) + \frac{\mathrm{d}d_i^{\mathrm{L}}}{\mathrm{d}\tau}(\tau) &\leqslant -\frac{1}{2}\frac{((d_i^{\mathrm{R}}(\tau))^2 + (d_i^{\mathrm{L}}(\tau))^2)(\delta f)_i^2}{(d_i^{\mathrm{L}}(\tau) + d_i^{\mathrm{R}}(\tau))^2} + 2\Delta g\\
&\leqslant -\frac{(\Delta f)^2}{4} + 2\frac{D(\Delta f)^2}{60} \leqslant -\frac{(\Delta f)^2}{5}
\end{aligned}$$

since $D \leqslant \Delta(u(0)) \leqslant 1$. Thus, in some time less than $\mathcal{T} = \frac{6}{(\Delta f)^2}$, $d_i^{\mathrm{R}}(\tau) + d_i^{\mathrm{L}}(\tau) \leqslant \eta$, that is, either u_i^{L} reaches $v_i - \frac{\eta}{2}$ or u_i^{R} reaches $v_i + \frac{\eta}{2}$. Let us check that the second event cannot actually

happen: while $u_i^{\mathrm{L}}(\tau) < v_i - \frac{\eta}{2}$ and $u_i^{\mathrm{R}}(\tau) > v_i + \frac{\eta}{2}$, we also have

$$\begin{aligned}\frac{\mathrm{d}d_i^{\mathrm{R}}}{\mathrm{d}\tau}(\tau) - \frac{\mathrm{d}d_i^{\mathrm{L}}}{\mathrm{d}\tau}(\tau) &\geqslant \frac{((d_i^{\mathrm{R}}(\tau))^2 - (d_i^{\mathrm{L}}(\tau))^2)(\delta f)_i^2}{(d_i^{\mathrm{L}}(\tau) + d_i^{\mathrm{R}}(\tau))^2} - 2\Delta g \\ &= \frac{(d_i^{\mathrm{R}}(\tau) - d_i^{\mathrm{L}}(\tau))(\delta f)_i^2}{d_i^{\mathrm{L}}(\tau) + d_i^{\mathrm{R}}(\tau)} - 2\Delta g\,.\end{aligned}$$

By condition (c) of Definition 6.6 and by (6.14), we have $d_i^{\mathrm{R}}(0) - d_i^{\mathrm{L}}(0) \geqslant D = \frac{60\Delta g}{(\Delta f)^2} \geq \frac{60\Delta g}{(\delta f)_i^2}$, and furthermore $d_i^{\mathrm{L}}(\tau) + d_i^{\mathrm{R}}(\tau) \leq 1$. An easy reasoning then shows that $d_i^{\mathrm{R}} - d_i^{\mathrm{L}}$ is increasing. Therefore u_i^{R} must remain further away from v_i than u_i^{L}.

In summary, we showed that there exists some time $\mathcal{T}_i \leqslant \mathcal{T}$ when $u_i^{\mathrm{L}}(\mathcal{T}_i) = v_i - \frac{\eta}{2}$, which marks the end of the first phase, and we also have

$$d_i^{\mathrm{R}}(\mathcal{T}_i) - d_i^{\mathrm{L}}(\mathcal{T}_i) \geqslant d_i^{\mathrm{R}}(0) - d_i^{\mathrm{L}}(0) \geqslant D\,.$$

Moving on to the study of the second phase, let us show that $u_i^{\mathrm{L}}(\tau)$ stays in the interval $[v_i - \eta, v_i + \eta]$ for $\tau \in [\mathcal{T}_i, \overline{\mathcal{T}})$. Consider any $\tau \leqslant \overline{\mathcal{T}}$ such that $u_i^{\mathrm{L}}(\tau) = v_i - \eta$. Then we have by (6.14) and Lemma 6.13

$$\frac{\mathrm{d}u_i^{\mathrm{L}}}{\mathrm{d}\tau}(\tau) \geqslant \frac{(u_i^{\mathrm{R}}(\tau) - v_i)^2(\delta f)_i^2}{(u_i^{\mathrm{R}}(\tau) - v_i + \eta)^2} - \Delta g \geqslant \Delta g\,, \tag{6.15}$$

where the second upper bound comes from the fact that we have $u_i^{\mathrm{R}}(\tau) - v_i \geqslant \frac{D}{2} - \eta$ since $\Delta(u(\tau)) \geqslant D/2$, and furthermore, $x \mapsto \frac{x^2}{(x+\eta)^2}$ is increasing, hence

$$\frac{(u_i^{\mathrm{R}} - v_i)^2(\delta f)_i^2}{(u_i^{\mathrm{R}} - v_i + \eta)^2} \geqslant \Big(\frac{\frac{D}{2} - \eta}{\frac{D}{2}}\Big)^2 \Delta f^2 \underset{(D/2 \geqslant 2\eta)}{\geqslant} \frac{(\Delta f)^2}{4} \geqslant 2\Delta g\,.$$

Equation (6.15) implies that $u_i^{\mathrm{L}}(\tau) \geqslant v_i - \eta$ for all $\tau \in [\mathcal{T}_i, \overline{\mathcal{T}})$. Similarly, consider any $\tau \leqslant \overline{\mathcal{T}}$ such that $u_i^{\mathrm{L}}(\tau) = v_i + \eta$. Note that, for such a τ, $u_i^{\mathrm{L}}(\tau)$ is now on the right of v_i, and the neurons flanking v_i are u_i^{LL} and u_i^{L}. Thus we have by (6.14) and Lemma 6.13

$$\frac{\mathrm{d}u_i^{\mathrm{L}}}{\mathrm{d}\tau}(\tau) \leqslant -\frac{(v_i - u_i^{\mathrm{LL}}(\tau))^2(\delta f)_i^2}{(v_i + \eta - u_i^{\mathrm{LL}}(\tau))^2} + \Delta g \leqslant -\Delta g\,,$$

where the second lower bound unfolds similarly as previously. This shows that $u_i^{\mathrm{L}}(\tau) \leqslant v_i + \eta$ for all $\tau \in [\mathcal{T}_i, \overline{\mathcal{T}})$.

We now check that $\mathcal{T} < \overline{\mathcal{T}}$, that is, for all $\tau \leqslant \mathcal{T}$, $\Delta(u(\tau)) > D/2$ and there are at least two neurons in each interval $[v_i, v_{i+1}]$. Starting with the first condition, we say that neurons u_j and u_k collide if $|u_j(\tau) - u_k(\tau)| = D/2$ for some $\tau \leqslant \mathcal{T}$. Let us show that no pair of neurons collide.

We start by showing that there is no collision between u_i^{LL} and u_i^{L}. In the first phase $[0, \mathcal{T}_i]$, we have $\frac{\mathrm{d}u_i^{\mathrm{LL}}}{\mathrm{d}\tau}(\tau) \leqslant \Delta g$. Recall that we also have $\frac{\mathrm{d}u_i^{\mathrm{L}}}{\mathrm{d}\tau}(\tau) \geqslant -\Delta g$ and thus for $\tau \leqslant \mathcal{T}_i$,

$$u_i^{\mathrm{L}}(\tau) - u_i^{\mathrm{LL}}(\tau) \geqslant u_i^{\mathrm{L}}(0) - u_i^{\mathrm{LL}}(0) - 2\mathcal{T}\Delta g \geqslant \frac{4D}{5}$$

since $u_i^{\mathrm{L}}(0) - u_i^{\mathrm{LL}}(0) \geqslant D$ and $\mathcal{T}\Delta g = D/10$ by definition of $\mathcal{T}$ and Δg. As a consequence, u_i^{LL} and u_i^{L} do not collide during the first phase, and we have

$$u_i^{\mathrm{LL}}(\mathcal{T}_i) \leqslant u_i^{\mathrm{L}}(\mathcal{T}_i) - \frac{4D}{5} = v_i - \frac{\eta}{2} - \frac{4D}{5}\,. \tag{6.16}$$

In the second phase, we can have $u_i^{\mathrm{L}} \in [v_i, v_i + \eta]$ in which case u_i^{LL} becomes the neuron flanking v_i to the left and u_i^{L} the neuron flanking to the right. Then (6.14) and Lemma 6.13 give

$$\frac{\mathrm{d}u_i^{\mathrm{LL}}}{\mathrm{d}\tau} \leqslant \frac{(u_i^{\mathrm{L}}(\tau) - v_i)^2 (\delta f)_i^2}{(u_i^{\mathrm{L}}(\tau) - u_i^{\mathrm{LL}}(\tau))^2} + \Delta g \leqslant \frac{16\eta^2 M^2}{D^2} + \Delta g \,.$$

Of course, this bound also holds when $u_i^{\mathrm{L}} \in [v_i - \eta, v_i]$, because then $\frac{\mathrm{d}u_i^{\mathrm{LL}}}{\mathrm{d}\tau} \leqslant \Delta g$. Thus, in the second phase $\tau \in [\mathcal{T}_i, \mathcal{T}]$, by the previous upper bound and the fact that $u_i^{\mathrm{L}}(\tau) \geqslant v_i - \frac{\eta}{2}$,

$$\begin{aligned} u_i^{\mathrm{L}}(\tau) - u_i^{\mathrm{LL}}(\tau) &\geqslant v_i - \frac{\eta}{2} - \Big(u_i^{\mathrm{LL}}(\mathcal{T}_i) + (\tau - \mathcal{T}_i)\Big(\frac{16\eta^2 M^2}{D^2} + \Delta g \Big) \Big) \\ &\geqslant \frac{4D}{5} - \mathcal{T}\Big(\frac{16\eta^2 M^2}{D^2} + \Delta g \Big) \,, \end{aligned}$$

by (6.16). Let us now upper-bound each of the last two terms by $D/10$ to conclude. By definition of D,

$$\eta = \frac{(\Delta f)^2 D^2}{2^{13}(m+1)M^2} \,.$$

Thus

$$\frac{16\eta^2 M^2 \mathcal{T}}{D^2} = \frac{3(\Delta f)^2 D^2}{2^{21}(m+1)^2 M^2} \leqslant \frac{D}{10}$$

using the definition of $\mathcal{T}$, $D \leqslant \Delta(u(0)) \leqslant 1$, $\Delta f \leqslant 2M$ and $m + 1 \geqslant 1$. Finally, $\mathcal{T}\Delta g = D/10$. Thus u_i^{LL} and u_i^{L} do not collide.

We now show that u_i^{L} and u_i^{R} do not collide. In the first phase $\tau \in [0, \mathcal{T}_i]$, we have

$$u_i^{\mathrm{R}}(\tau) - u_i^{\mathrm{L}}(\tau) \geqslant u_i^{\mathrm{R}}(\tau) - v_i = d_i^{\mathrm{R}}(\tau) \geqslant d_i^{\mathrm{R}}(\tau) - d_i^{\mathrm{L}}(\tau) \geqslant D \,.$$

As a consequence, u_i^{L} and u_i^{R} do not collide during the first phase, and we have

$$u_i^{\mathrm{R}}(\mathcal{T}_i) \geqslant D + u_i^{\mathrm{L}}(\mathcal{T}_i) = D + v_i - \frac{\eta}{2} \,. \tag{6.17}$$

In the second phase, u_i^{R} plays a role symmetric to u_i^{LL}: it can be, or not, the neuron closest to the right of v_i, depending on whether $u_i^{\mathrm{L}} \in [v_i - \eta, v_i]$ or $u_i^{\mathrm{L}} \in [v_i, v_i + \eta]$. As for u_i^{LL}, we can show that in any case, for $\tau \in [\mathcal{T}_i, \mathcal{T}]$,

$$\frac{\mathrm{d}u_i^{\mathrm{R}}}{\mathrm{d}\tau} \geqslant -\frac{16\eta^2 M^2}{D^2} - \Delta g \,.$$

Thus one concludes as before: for $\tau \in [\mathcal{T}_i, \mathcal{T}]$, by the previous lower bound and the fact that $u_i^{\mathrm{L}}(\tau) \leqslant v_i + \frac{\eta}{2}$,

$$u_i^{\mathrm{R}}(\tau) - u_i^{\mathrm{L}}(\tau) \geqslant u_i^{\mathrm{R}}(\mathcal{T}_i) - (\tau - \mathcal{T}_i)\Big(\frac{16\eta^2 M^2}{D^2} + \Delta g \Big) - \Big(v_i + \frac{\eta}{2} \Big) \,.$$

Then, by (6.17),

$$u_i^{\mathrm{R}}(\tau) - u_i^{\mathrm{L}}(\tau) \geqslant D - \eta - \mathcal{T}\Big(\frac{16\eta^2 M^2}{D^2} + \Delta g \Big) > \frac{D}{2} \,,$$

where the last lower-bound unfolds similarly as for u_i^{LL} and u_i^{L}. Thus there is no collision between u_i^{L} and u_i^{R}.

The reader can check that all other pairs of neurons do not collide, including those involving $u_0 = -\eta/2$ and $u_{m+1} = 1 + \eta/2$. In fact, the proof is easier than for $u_i^{\mathrm{LL}}, u_i^{\mathrm{L}}$ and $u_i^{\mathrm{L}}, u_i^{\mathrm{R}}$ because the discontinuity at v_i attracts these neurons together.

Furthermore, we proved that before time $\mathcal{T}$ at most one neuron can escape on each side of a piece $[v_i, v_{i+1}]$ of f. Since we start with at least four (and even six) neurons per piece, there is always before $\mathcal{T}$ at least two neurons per piece.

This shows that $\mathcal{T} < \overline{\mathcal{T}}$, and we also proved that at time $\mathcal{T}$, all discontinuities have finished their first phase, hence there is a neuron at distance less than η from each discontinuity of the target function.

6.B.4 Proof of Theorem 6.8

Take $C = 2^{-19}$. Then by assumption of Theorem 6.8,

$$\eta \leqslant \frac{\delta^2 (\Delta f)^2}{2^{19} M^2 (m+1)^5} \, .$$

Moreover, by the definition of D from Proposition 6.7,

$$\eta = \frac{(\Delta f)^2 D^2}{2^{13} M^2 (m+1)} \, .$$

This implies that

$$D^2 \leqslant \frac{\delta^2}{2^6 (m+1)^4} \, ,$$

and in consequence

$$D \leqslant \frac{\delta}{6(m+1)^2} \, .$$

Then Lemma 6.19 shows that the initialization is D-good with probability at least $1 - \delta$ (since the D-good property is monotonous in D).

Hence, with probability at least $1 - \delta$, according to Proposition 6.7, the maximal solution to (6.8) is defined at least until $\mathcal{T}$ and at that time, there is a neuron at distance less than η from each discontinuity of the target function. Furthermore, $3\eta \leq \frac{1}{m+1} \leq \frac{1}{n} \leq \Delta v$, hence Lemma 6.18 applies. This implies that

$$\int_0^1 |f^*(x) - f(x; a^*(u(\mathcal{T})), u(\mathcal{T}))|^2 \mathrm{d}x \leqslant 6 M^2 \eta n \, .$$

The assumption on η allow to conclude that the upper-bound is less than ξ.

Remark 6.20. *We did not try to optimize the value of C since our goal was to show convergence to a global optimum and the dependency of the dynamics on the parameters (for instance, it is remarkable that $\mathcal{T}$ does not depend on ξ).*

6.B.5 Proof of Proposition 6.9

For $s \leq t$, Proposition 6.4 holds since for $\Delta(u(s)) \geq 16\eta > 2\eta$. Thus $a_\eta^*(u(s))$ is well-defined and verifies

$$\nabla_a L_\eta(a_\eta^*(u(s)), u(s)) = 0 \, .$$

Let, for $s \leq t$, $V(s) = \|a(s) - a_\eta^*(u(s))\|$. Recall that, by (6.11),

$$\nabla_a L_\eta(a, u) = H_\eta(u) a - b_\eta(u) \, .$$

Hence, for $s \le t$,

$$\begin{aligned}
&\langle a(s) - a_\eta^*(u(s)), \nabla_a L_\eta(a(s), u(s))\rangle \\
&\qquad = \langle a(s) - a_\eta^*(u(s)), \nabla_a L_\eta(a(s), u(s)) - \nabla_a L_\eta(a_\eta^*(u(s)), u(s))\rangle \\
&\qquad = \langle a(s) - a_\eta^*(u(s)), H_\eta(u(s))(a(s) - a_\eta^*(u(s)))\rangle \\
&\qquad \geqslant \frac{\Delta(u(s))}{8} V(s)^2 \\
&\qquad \geq \frac{D}{16} V(s)^2 \,,
\end{aligned}$$

where the first lower bound is a consequence of Proposition 6.4. Then we have, for any $s \le t$,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}s}\Big(\frac{1}{2}V(s)^2\Big) &= \Big\langle a(s) - a_\eta^*(u(s)), \frac{\mathrm{d}a}{\mathrm{d}s}(s) - \frac{\mathrm{d}}{\mathrm{d}s}a_\eta^*(u(s))\Big\rangle \\
&= \Big\langle a(s) - a_\eta^*(u(s)), -\nabla_a L_\eta(a(s), u(s)) - \frac{\mathrm{d}}{\mathrm{d}s}a_\eta^*(u(s))\Big\rangle \\
&\leqslant -\frac{D}{16}V(s)^2 + \Big\|\frac{\mathrm{d}}{\mathrm{d}s}a_\eta^*(u(s))\Big\| V(s) \,.
\end{aligned}$$

Let us now upper bound the norm appearing in the second term. We first have by the chain rule

$$\frac{\mathrm{d}}{\mathrm{d}s}a_\eta^*(u(s)) = \frac{\partial a_\eta^*}{\partial u}(u(s))\frac{\mathrm{d}u}{\mathrm{d}s}(s) \,.$$

By Lemma 6.12 (which holds since for $\Delta(u(s)) \ge 16\eta > 2\eta$),

$$\Big\|\frac{\partial a_\eta^*}{\partial u}(u(s))\Big\| \leqslant \frac{8}{\Delta(u(s))}\big(2(m+1)\|a_\eta^*(u(s))\| + M\big) \,.$$

Besides,

$$\Big\|\frac{\mathrm{d}u}{\mathrm{d}s}(s)\Big\| \le \varepsilon\|\nabla_u L_\eta(a(s), u(s))\| \,.$$

By Lemma 6.17,

$$\|\nabla_u L_\eta(a(s), u(s))\| \leqslant \sqrt{m+1}\|a(s)\|^2 + M\|a(s)\| \,. \tag{6.18}$$

Furthermore,

$$\|a(s)\| \le \|a_\eta^*(u(s))\| + \|a(s) - a_\eta^*(u(s))\| = \|a_\eta^*(u(s))\| + V(s) \,.$$

By Lemmas 6.15 and 6.16, which apply since $\Delta(u(s)) > 2\eta$ and since there are at least two positions $u_j(s)$ in each interval $[v_i, v_{i+1}]$ for $s \le t$,

$$\begin{aligned}
\|a_\eta^*(u(s))\| &\leqslant \|a_0^*(u(s))\| + \|a_0^*(u(s)) - a_\eta^*(u(s))\| \\
&\leqslant 2M\sqrt{m+1} + \frac{16M\sqrt{m+1}\eta}{\Delta(u(s))} \\
&\leqslant 2M\sqrt{m+1} + \frac{32M\sqrt{m+1}\eta}{D} \\
&\leqslant 3M\sqrt{m+1} \,,
\end{aligned}$$

where the last upper bound is implied by the assumption $D \ge 32\eta$.

Now define $T_{\max} = \inf\big\{s \geqslant 0, V(s) > 3M\sqrt{m+1}\big\}$ and assume $s \le \min(t, T_{\max})$ so that $V(s) \le 3M\sqrt{m+1}$. Then we proved that $\|a(s)\| \le 6M\sqrt{m+1}$. Going back to (6.18), we deduce that

$$\|\nabla_u L_\eta(a(s), u(s))\| \leqslant 36M^2(m+1)^{3/2} + 6M^2\sqrt{m+1} \leqslant 2^6 M^2 (m+1)^{3/2} \,. \tag{6.19}$$

Putting everything together, we obtain

$$\Big\|\frac{\mathrm{d}}{\mathrm{d}s}a_\eta^*(u(s))\Big\| \leqslant \frac{2^9M^2(m+1)^{3/2}}{\Delta(u(s))}\big(6M(m+1)^{3/2}+M\big)\varepsilon \leq \frac{2^{13}M^3(m+1)^3}{D}\varepsilon\,.$$

All in all,

$$\frac{\mathrm{d}}{\mathrm{d}s}\Big(\frac{1}{2}V(s)^2\Big) \leqslant -\frac{D}{16}V(s)^2 + \frac{2^{13}M^3(m+1)^3}{D}\varepsilon V(s)\,.$$

Hence

$$\frac{\mathrm{d}}{\mathrm{d}s}(V(s)) = \frac{1}{V(s)}\frac{\mathrm{d}}{\mathrm{d}s}\big(\frac{1}{2}V(s)^2\big) \leqslant -\frac{D}{16}V(s) + \frac{2^{13}M^3(m+1)^3}{D}\varepsilon\,.$$

By Grönwall's inequality, for all $s \leqslant \min(t, T_{\max})$,

$$V(s) \leqslant \exp^{-\frac{D}{16}s}V(0) + \frac{2^{17}M^3(m+1)^3}{D^2}\varepsilon(1-\exp^{-\frac{D}{16}s}) \tag{6.20}$$

$$\leqslant \exp^{-\frac{D}{16}s}V(0) + \frac{2^{17}M^3(m+1)^3}{D^2}\varepsilon\,. \tag{6.21}$$

Finally note that $V(0) = \|a_\eta^*(0)\| \leqslant 2M\sqrt{m+1}$ and $\frac{2^{17}M^3(m+1)^3\varepsilon}{D^2} \leqslant 2M\sqrt{m+1}$ by the assumption of the Proposition on ε. Hence (6.20) implies that for all $s \leqslant \min(t, T_{\max})$, $V(s)$ is a (weighted) average of two terms less than $2M\sqrt{m+1}$ hence it is less than $2M\sqrt{m+1}$. This shows that $T_{\max} \geqslant t$, which concludes the proof since (6.21) is then valid for $s = t$.

6.B.6 Proof of Theorem 6.2

In the proof, we take $C_1 = 2^{-21}$ and $C_2 = 2^{-36}$. Denote

$$D = \frac{\delta}{6(m+1)^2}\,.$$

Lemma 6.19 shows that the initialization is D-good with probability at least $1-\delta$. In the following, we study the case where this event happens.

Denote $\overline{T}$ the minimal time $t > 0$ such that $\Delta(u(t)) \leqslant D/2$ or there are less than two neurons in some piece $[v_i, v_{i+1}]$ of f^* or $\|a(t)\| > 7M\sqrt{m+1}$. Note that $\overline{T} > 0$ since the initialization is D-good. By Lemma 6.17, $\nabla_u L_\eta$ and $\nabla_a L_\eta$ are Lipschitz-continuous on compacts, hence the solution of the gradient flow is well-defined for $t < \overline{T}$ since $\overline{T}$ defines a compact set of parameters.

Then all the assumptions of Proposition 6.9 are satisfied on the time interval $[0, t]$ for any $t < \overline{T}$. More precisely, the assumptions that do not come directly from the definition of $\overline{T}$ are the lower bound for D and the upper bound for ε. The lower bound for D come from

$$D = \frac{\delta}{6(m+1)^2} \geq 32\eta \tag{6.22}$$

by (6.3) and the simple bounds $\delta \leq 1$, $\Delta f \leq 2M$, $m+1 \geq 1$. The upper bound for ε comes from (6.3) since

$$\varepsilon \leq \frac{\delta^3(\Delta f)^2}{2^{36}M^4(m+1)^{19/2}} \leq \frac{\delta^2}{36\cdot 2^{16}M^2(m+1)^{13/2}} = \frac{D^2}{2^{16}M^2(m+1)^{5/2}}\,,$$

where the second upper bound uses $m \geq 0$, $\delta \leq 1$ and $\Delta f \leq 2M$. Therefore, according to Proposition 6.9,

$$\|a(t) - a_\eta^*(u(t))\| \leqslant 3M\sqrt{m+1}\exp^{-\frac{D}{16}t} + \frac{2^{17}M^3(m+1)^3}{D^2}\varepsilon\,, \tag{6.23}$$

Furthermore, the proof of Proposition 6.9 actually implies that

$$\|a_\eta^*(u(t))\| \leqslant 3M\sqrt{m+1} \quad \text{and} \quad \|a(t)\| \leqslant 6M\sqrt{m+1}. \tag{6.24}$$

The second bound implies that the condition $\|a(t)\| > 7M\sqrt{m+1}$ in the definition of $\overline{T}$ is actually never active. Let us distinguish between two phases: letting

$$T_0 = \frac{16}{D}\log\Big(\frac{2^{16}M^2(m+1)^3}{\delta(\Delta f)^2}\Big) = \frac{96(m+1)^2}{\delta}\log\Big(\frac{2^{16}M^2(m+1)^3}{\delta(\Delta f)^2}\Big),$$

then the first phase corresponds to $t \leq T_0$ and the second phase for $t \geq T_0$.

Analysis of the first phase. In the first phase, each neuron moves at most by

$$\varepsilon T_0 \max_j \Big|\frac{\partial L_\eta}{\partial u_j}(a(t),u(t))\Big| \leqslant \varepsilon T_0 \|\nabla_u L_\eta(a(s),u(s))\| \leqslant 2^6 \varepsilon T_0 M^2 (m+1)^{3/2}\,,$$

where the second upper bound comes from (6.19) in the proof of Proposition 6.9. This quantity is less than $\frac{D}{8}$ if

$$\frac{6144(m+1)^{7/2}M^2}{\delta}\log\Big(\frac{2^{16}M^2(m+1)^3}{\delta(\Delta f)^2}\Big)\varepsilon \leqslant \frac{\delta}{48(m+1)^2}\,.$$

Let us check this condition: we have

$$\begin{aligned}
&\frac{6144(m+1)^{7/2}M^2}{\delta}\log\Big(\frac{2^{16}M^2(m+1)^3}{\delta(\Delta f)^2}\Big)\varepsilon \\
&\qquad = \frac{16\cdot 6144(m+1)^{7/2}M^2}{\delta}\log\Big(\frac{2M^{1/8}(m+1)^{3/16}}{\delta^{1/16}(\Delta f)^{1/8}}\Big)\varepsilon \\
&\qquad \leqslant \frac{16\cdot 6144(m+1)^{7/2}M^2}{\delta}\log\Big(\frac{4M(m+1)}{\delta\Delta f}\Big)\varepsilon\,,
\end{aligned}$$

since $m+1 \geq 1$, $\delta \leq 1$, and $2M/\Delta f \geq 1$, hence $(2M/\Delta f)^{1/8} \leq 2M/\Delta f$. Next, upper-bounding $\log(x)$ by x, we have, by (6.3),

$$\begin{aligned}
\frac{768(m+1)^{7/2}M^2}{\delta}\log\Big(\frac{2^{16}M^2(m+1)^3}{\delta(\Delta f)^2}\Big)\varepsilon &\leqslant \frac{64\cdot 6144(m+1)^{9/2}M^3}{\delta^2\Delta f}\varepsilon \\
&\leqslant \frac{6144\delta(\Delta f)}{2^{29}M(m+1)^5} \\
&\leqslant \frac{\delta}{48(m+1)^2}
\end{aligned}$$

using $\Delta f \leq 2M$ and $m \geq 0$. Note that the upper bound $2^6\varepsilon T_0 M^2(m+1)^{3/2} \leq D/8$ also implies that

$$T_0 \leq \frac{D}{2^9\varepsilon M^2(m+1)^{3/2}} \leq \frac{1}{2\varepsilon(\Delta f)^2} = \frac{T}{12} \tag{6.25}$$

since $m \geq 0$, $D \leq 1$ and $\Delta f \leq 2M$. Since each neuron moves by at most $D/8$ in the time interval $[0, T_0]$ and since $\Delta(u(0)) \geq D$, we deduce that

$$\Delta(u(T_0)) \geq \frac{3}{4}D. \tag{6.26}$$

Similarly, by condition (c) of the definition of a D-good vector, for all discontinuities v_i,

$$|u_i^{\mathrm{R}}(0) + u_i^{\mathrm{L}}(0) - 2v_i| \geqslant D,$$

thus

$$|u_i^{\mathrm{R}}(T_0) + u_i^{\mathrm{L}}(T_0) - 2v_i| \geqslant \frac{3}{4}D\,. \tag{6.27}$$

Furthermore, there at least four neurons on each piece of f at T_0, because at most one neuron can move out of each piece by either side between 0 and T_0.

Analysis of the second phase. Let

$$\Delta a = \frac{D(\Delta f)^2}{2^9 M\sqrt{m+1}} = \frac{\delta(\Delta f)^2}{6\cdot 2^9 M(m+1)^{5/2}}\,.$$

In the second phase $t \geq T_0$, we are able to control by Δa the distance between $a(t)$ and the weights $a_0^*(u(t))$ that are the best approximation of f^* by a piecewise affine function with subdivision $u(t)$. To show this, first note that the first term in (6.23) is smaller than $\frac{\Delta a}{4}$ when

$$3M\sqrt{m+1}\exp^{-\frac{D}{16}t} \leqslant \frac{\Delta a}{4}\,,$$

which is equivalent to

$$t \geqslant \log\Big(\frac{12M\sqrt{m+1}}{\Delta a}\Big)\frac{16}{D}\,,$$

which is implied by $t \geqslant T_0$. Furthermore, the second term in (6.23) is smaller than $\frac{\Delta a}{4}$ because, by definition of D and by (6.3),

$$\frac{2^{17}M^3(m+1)^3}{D^2}\varepsilon = \frac{36\cdot 2^{17}M^3(m+1)^7}{\delta^2}\varepsilon \leq \frac{6^2\delta(\Delta f)^2}{2^{19}M(m+1)^{5/2}} = \frac{6^3\Delta a}{2^{10}} \leq \frac{\Delta a}{4}\,.$$

Hence, for all $T_0 \leqslant t < \overline{T}$,

$$\|a(t) - a_\eta^*(u(t))\| \leqslant \frac{\Delta a}{2}\,.$$

Furthermore, note that the assumption of Lemma 6.16 applies for $t < \overline{T}$ since $\Delta(u(t)) \geq \frac{D}{2} > 2\eta$ by (6.22). Therefore, by Lemma 6.16 and by (6.3),

$$\begin{aligned}
\|a_\eta^*(u(t)) - a_0^*(u(t))\| &\leqslant \frac{2^4 M\sqrt{m+1}}{\Delta(u(t))}\eta \\
&\leq \frac{2^5 M\sqrt{m+1}}{D}\eta \\
&= \frac{2^5\cdot 6M(m+1)^{5/2}}{\delta}\eta \\
&\leq \frac{6\delta(\Delta f)^2}{2^{16}M(m+1)^{5/2}} \\
&= \frac{6^2\Delta a}{2^7} \leqslant \frac{\Delta a}{2}\,.
\end{aligned}$$

By the triangular inequality, we deduce the upper bound that we were after, that is

$$\|a(t) - a_0^*(u(t))\| \leqslant \Delta a\,.$$

As in the proof of Proposition 6.7, we can now control the distance between the true dynamics and the one that we would have if the weights were equal to $a_0^*(u)$. Namely, for any $T_0 \leq t \leq \overline{T}$ and $j \in \{1, \dots, m\}$, by Lemma 6.15 (which applies since $\Delta(u(t)) > 2\eta$ by (6.22)), we have

$$\begin{aligned}
&\Big|\frac{\mathrm{d}u_j}{\mathrm{d}t}(t) + \frac{\partial L_\eta}{\partial u_j}(a_0^*(u(t)), u(t))\Big| \\
&\quad = \Big|\frac{\partial L_\eta}{\partial u_j}(a(t), u(t)) - \frac{\partial L_\eta}{\partial u_j}(a_0^*(u(t)), u(t))\Big| \\
&\quad \leqslant 2M(\sqrt{m+1}+1)\|a(t) - a_0^*(u(t))\| + \sqrt{m+1}\|a(t) - a_0^*(u(t))\|^2 .
\end{aligned}$$

The first term is less than

$$2M(\sqrt{m+1}+1)\Delta a = \frac{(\sqrt{m+1}+1)D(\Delta f)^2}{2^8\sqrt{m+1}} \leqslant \frac{D(\Delta f)^2}{2^7} ,$$

and the second term is less than

$$\sqrt{m+1}(\Delta a)^2 = \frac{D^2(\Delta f)^4}{2^{18}M^2\sqrt{m+1}} \leqslant \frac{D(\Delta f)^2}{2^{16}} ,$$

using $D \leqslant \Delta(u(0)) \leqslant 1$, $\Delta f \leqslant 2M$ and $m+1 \geqslant 1$. Hence we obtain, for any $T_0 \leq t \leq \overline{T}$ and $j \in \{1, \dots, m\}$,

$$\Big|\frac{\mathrm{d}u_j}{\mathrm{d}t}(t) + \frac{\partial L_\eta}{\partial u_j}(a_0^*(u(t)), u(t))\Big| \leqslant \frac{D(\Delta f)^2}{120} .$$

We are therefore in a situation very similar to the proof of Proposition 6.7, starting from (6.14). One can check that all the arguments used in the proof also apply here. On top of the estimate above that resembles (6.14), the crucial facts that make the argument of Proposition 6.7 work here are the bounds (6.26) and (6.27) as well as the fact that there are at least four neurons on each piece f at T_0, which together are very similar to the conditions ensuring that $u(0)$ is D-good in the proof of Proposition 6.7. Another key point is (6.25), ensuring that a time at least equal to $11T/12$ remains after the first phase of this proof, which is enough time for the dynamics described in the proof of Proposition 6.7 to unfold.

This yields that $T < \overline{T}$, and that at time T, there is a neuron at distance less than η from each discontinuity of f^*. Furthermore, $3\eta \leq \frac{1}{m+1} \leq \frac{1}{n} \leq \Delta v$, hence Lemma 6.18 applies. Thus

$$\int_0^1 (f_\eta(x; a_\eta^*(u(T)), u(T)) - f^*(x))^2 \mathrm{d}x \leqslant 6M^2\eta n \leqslant \frac{\xi}{2} ,$$

where the second upper bound comes from $n \leq m+1$ and from (6.3). Furthermore, by (6.24) and by Lemma 6.17,

$$\begin{aligned}
|L_\eta(a(T), u(T)) - L_\eta(a_\eta^*(u(T)), u(T))| &\leqslant \sqrt{m+1}(6M(m+1)+M)\|a(T) - a_\eta^*(u(T))\| \\
&\leq 16M(m+1)^{3/2}\|a(T) - a_\eta^*(u(T))\| .
\end{aligned}$$

Let us show that this term is less than $\xi/4$. Recall that, by (6.23),

$$\|a(T) - a_\eta^*(u(T))\| \leqslant 3M\sqrt{m+1}\exp^{-\frac{D}{16}T} + \frac{2^{17}M^3(m+1)^3}{D^2}\varepsilon .$$

By definition of D and T, by using $\exp(-x) \leq 1/x$ for $x \geq 1$ and by (6.3),

$$\begin{aligned} 16M(m+1)^{3/2} \cdot 3M\sqrt{m+1}\exp^{-\frac{D}{16}T} &= 48M^2(m+1)^2 \exp\Big(-\frac{\delta}{16(m+1)^2(\Delta f)^2\varepsilon}\Big) \\ &\leq \frac{48 \cdot 16M^2(m+1)^4(\Delta f)^2}{\delta}\varepsilon \\ &\leq \frac{48(\Delta f)^2\delta}{2^{31}M^2(m+1)^{9/2}}\xi \\ &\leqslant \frac{\xi}{8} \end{aligned}$$

using $\Delta f \leqslant 2M$, $\delta \leq 1$, and $m+1 \geqslant 1$. Furthermore, by (6.3), we get that

$$16M(m+1)^{3/2} \cdot \frac{2^{17}M^3(m+1)^3}{D^2}\varepsilon = \frac{36 \cdot 2^{21}M^4(m+1)^{17/2}}{\delta^2}\varepsilon \leqslant \frac{\xi}{8}\,.$$

We therefore obtain the sought $\xi/4$ upper-bound and can conclude that

$$\begin{aligned} \int_0^1 (f_\eta(x; a(T), u(T)) - f^*(x))^2 \mathrm{d}x &\leqslant \int_0^1 (f_\eta(x; a_\eta^*(u(T)), u(T)) - f^*(x))^2 \mathrm{d}x \\ &\quad + 2|L_\eta(a(T), u(T)) - L_\eta(a_\eta^*(u(T)), u(T))| \\ &\leqslant \xi\,. \end{aligned}$$

6.C Experimental details

Setting Our code is available at `https://github.com/PierreMarion23/two-timescale-nn`. To obtain Figures 6.3 and 6.4, we use the parameters of Table 6.1. For Figure 6.5, we use the parameters of Table 6.2.

Name	Value
m	20
ε	$2 \cdot 10^{-5}$
η	$4 \cdot 10^{-3}$
P	$1.8 \cdot 10^8$
h	10^{-5}
Additive noise	Uniform on $[-1, 1]$

Table 6.1: Parameters of Figures 6.3 and 6.4.

The number of iterations in Table 6.1 is much larger than the one in Table 6.2, due to the fact that the positions u evolve at a speed εh, which is much smaller in Table 6.1. However, note that it is possible to increase h in Table 6.1 while keeping the same behavior (in our experiment, h is kept to the same value as in Table 6.2 in order to facilitate the comparison). More precisely, taking $h = 10^{-3}$ in Table 6.1 yields similar results while dividing the computational cost by 100.

Our target function is defined by $f^* = 1$ on $[0., 0.2], [0.35, 0.5], [0.65, 0.8]$, $f^* = 2$ on $[0.5, 0.65]$ and $f^* = 4$ elsewhere.

Additional plot We re-run the same SGD experiment as above twenty times, and plot the average $L2$ distance to the target as a function of ε, averaging over the initialization randomness

Name	Value
m	20
ε	1
η	$4 \cdot 10^{-3}$
P	10^6
h	10^{-5}
Additive noise	Uniform on $[-1, 1]$

Table 6.2: Parameters of Figure 6.5.

and SGD randomness. This confirms that, in our setting, the SGD is able to recover the target function in the two-timescale regime ($\varepsilon \ll 1$), but fails outside the two-timescale regime ($\varepsilon = 1$). The transition between the two regimes seems to occur for $\varepsilon \approx 0.1$.

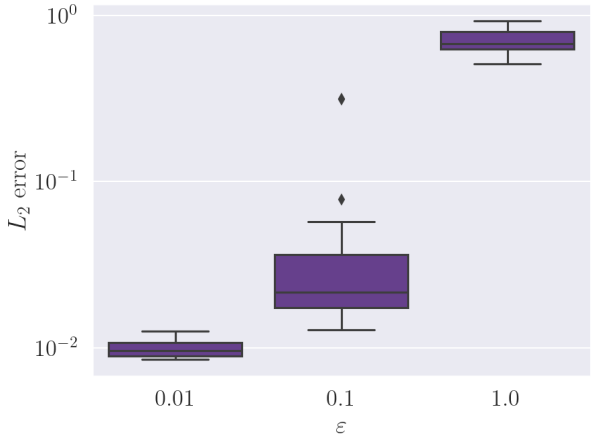


Figure 6.7: L_2 distance with the target as a function of ε, with 20 repeats

7

Structured context and high-coverage grammar for conversational question answering over knowledge graphs

We tackle the problem of weakly-supervised conversational Question Answering over large Knowledge Graphs using a neural semantic parsing approach. We introduce a new Logical Form (LF) grammar that can model a wide range of queries on the graph while remaining sufficiently simple to generate supervision data efficiently. Our Transformer-based model takes a JSON-like structure as input, allowing us to easily incorporate both Knowledge Graph and conversational contexts. This structured input is transformed to lists of embeddings and then fed to standard attention layers. We validate our approach, both in terms of grammar coverage and LF execution accuracy, on two publicly available datasets, CSQA and ConvQuestions, both grounded in Wikidata. On CSQA, our approach increases the coverage from 80% to 96.2%, and the LF execution accuracy from 70.6% to 75.6%, with respect to previous state-of-the-art results. On ConvQuestions, we achieve competitive results with respect to the state-of-the-art.

Contents

7.1 Introduction

Graphs are a common abstraction of real-world data. Large-scale knowledge bases can be represented as directed labeled graphs, where entities correspond to nodes and subject-predicate-object triplets are encoded by labeled edges. These so-called *Knowledge Graphs* (KGs) are used both in open knowledge projects (YAGO, Wikidata) and in the industry (Yahoo, Google, Microsoft, etc.). A prominent task on KGs is *factual conversational Question Answering* (Conversational KG-QA) and it has spurred interest recently, in particular due to the development of AI-driven personal assistants.

The Conversational KG-QA task involves difficulties of different nature: entity disambiguation, long tails of predicates (Saha et al., 2018), conversational nature of the interaction. The topology of the underlying graph is also problematic. Not only can KGs be huge (up to several billion entities), they also exhibit hub entities with numerous neighbors.

A recent prominent approach has been to cast the problem as neural *semantic parsing* (Jia and Liang, 2016; Dong and Lapata, 2016, 2018; Shen et al., 2019). In this setting, a semantic parsing model learns to map a natural language question to a *logical form* (LF), *i.e.* a tree of operators over the KG. These operators belong to some grammar, either standard like SPARQL or ad-hoc. The logical form is then *evaluated* over the KG to produce the candidate answer. In the *weak supervision* training setup, the true logical form is not available, but only the answer utterance is (as well as annotated entities in some cases, see Section 7.5.4). Hence the training data is not given but it is instead generated, in the format of *(question, logical form)* pairs. We refer to this data as *silver data* or *silver LFs*, as opposed to unknown gold ground truth.

However, this approach has two main issues. First, the silver data generation step is a complex and often resource-intensive task. The standard procedure employs a Breadth-First Search (BFS) exploration (Guo et al., 2018; Shen et al., 2019), but this simple strategy is prone to failure, especially when naively implemented, for questions that are mapped to nested LFs. This reduces the *coverage*, i.e. the percentage of training questions associated to a Logical Form. Shen et al. (2020) proposes to add a neural component for picking the best operator, in order to reduce the computational cost of this task, however complicating the model. Cao et al. (2020) proposes a two-step semantic parser: the question is first paraphrased into a "canonical utterance", which is then mapped to a LF. This approach simplifies the LF generation by separating it from the language understanding task.

Second, most of the semantic parsing models do not leverage much of the underlying KG structure to predict the LF, as in Dong and Lapata (2016); Guo et al. (2018). Yet, this contextual

graph information is rich (Tong et al., 2019), and graph-based models leveraging this information yield promising results for KG-QA tasks (Vakulenko et al., 2019; Christmann et al., 2019). However these alternative approaches to semantic parsing, that rely on node classification, have their inherent limitations, as they handle less naturally certain queries (see Appendix 7.C.3) and their output is less interpretable. This motivates the desire for semantic parsing models that can make use of the KG context.

Approach, contributions and overview of the chapter. In Section 7.3, we design a new grammar, which can model a large range of queries on the KG, yet is simple enough for BFS to work well. We obtain a high coverage on two KG-QA datasets. On CSQA (Saha et al., 2018), we achieve a coverage of 96%, a 16% improvement over the baseline (Shen et al., 2020). On ConvQuestions (Christmann et al., 2019), a dataset with a large variety of queries, we reach a coverage of 86%.

To leverage the rich information contained in the underlying KG, we propose in Section 7.4 a semantic parsing model that uses the KG contextual data in addition to the utterances. Different options could be considered for the KG context, *e.g.* lists of relevant entities, annotated with metadata or pre-trained entity embeddings that are graph-aware (Zhang et al., 2020b). The problem is that this information does not come as unstructured textual data, which is common for language models, but is structured.

To enable the use of context together with a strong language model, we propose the Object-Aware Transformer (OAT) model, which can take as input structured data in a JSON-like format. The model then transforms the structured input into embeddings, before feeding them into standard Transformer layers. With this approach, as reported in Section 7.5, we improve the overall execution accuracy on CSQA by 5.0% compared to a strong baseline (Shen et al., 2019). On ConvQuestions, we improve the precision by 4.7% compared to Christmann et al. (2019).

Appendices 7.A to 7.D present further description of the experimental setup, comparisons with baselines, and experimental results.

7.2 Related work

Neural semantic parsing Our work falls within the neural semantic parsing approaches for Knowledge-Based QA (Dong and Lapata, 2016; Liang et al., 2017; Dong and Lapata, 2018; Guo et al., 2019b; Hwang et al., 2019). The more specific task of conversational KG-QA has been the focus of recent work. Guo et al. (2018) introduces D2A, a neural symbolic model with memory augmentation. This model has been extended by S2A+MAML (Guo et al., 2019a) with a meta-learning strategy to account for context, and by D2A+ES (Shen et al., 2020) with a neural component to improve BFS. Saha et al. (2019) proposes a Reinforcement Learning model to benefit from denser supervision signals. Finally, Shen et al. (2019) introduces MaSP, a multi-task model that performs both entity linking and semantic parsing, with the hope of reducing erroneous entity linking (see Appendix 7.C.2 for a comparison with our setup). Recently, Plepi et al. (2021) extended the latter in CARTON. They first predict the LF using a Transformer architecture, then specify the KG items using pointer networks.

Learning on Knowledge Graphs Classical graph learning techniques can be applied to the specific case of KGs. In CONVEX (Christmann et al., 2019), at each turn, a subgraph is expanded by matching the utterance with neighboring entities. Then a candidate answer is found by a node classifier. Other methods include unsupervised message passing (Vakulenko et al., 2019). However, these approaches lack strong NLP components. Other directions include learning differentiable operators over a KG (Cohen et al., 2019), or applying Graph Neural Networks

(Kipf and Welling, 2017; Hamilton et al., 2017) (GNNs) to the KG, which has been done for entity classification and link prediction tasks (Schlichtkrull et al., 2018). GNNs have also been used to model relationships between utterances and entities (Shaw et al., 2019).

Structured Input for neural models Our approach of using JSON-like input falls in the line of computing neural embeddings out of structured inputs. Tai et al. (2015) introduced Tree-LSTM for computing tree embeddings bottom-up. It has then been applied for many tasks, including computer program translation (Chen et al., 2018b), semantic tree structure learning such as JSON or XML (Woof and Chen, 2020), and supervised KG-QA tasks (Tong et al., 2019; Zafar et al., 2019; Athreya et al., 2021). In the latter context, Tree-LSTM is used to model the syntactic structure of the question. Other related approaches include tree transformer (Harer et al., 2019) and tree attention (Ahmed et al., 2019). Syntactic structures were also modeled as graphs (Xu et al., 2018; Li et al., 2020a). Specific positional embeddings can also be used to encode structures (Shiv and Quirk, 2019; Herzig et al., 2020).

7.3 A grammar for KG exploration

Several previous KG-QA works were based on the grammar from D2A (Guo et al., 2018). We also take inspiration from their grammar, but redesign it to model a wider range of queries. By defining more generic operators, we achieve this without increasing the number of operators nor the average depth of LFs. Section 7.3.4 presents a comparison.

7.3.1 Definitions

An *entity* (e.g. `Marie Curie`) is a node in the KG. Two entities can be related through a directed labeled edge called a *property* (e.g. `award received`). A property can also relate an entity to a *value*, which can be a date, a boolean, a quantity or a string. Entities and properties have several attributes, prominently a name and an integer ID. The `membership` property is treated separately; it relates a member entity (e.g. `Marie Curie`) to a *class* entity (e.g. `human being`).

The *objects* we will consider in the following are entities, properties, classes, and values. The grammar consists of a list of *operators* that take objects or sets of objects as arguments. A Logical Form is a binary expression tree of operators.

In several places, we perform *Named Entity Linking (NEL)*, i.e. mapping an utterance to a list of KG objects. Section 7.5.4 details how this is done.

Table 7.1 lists the operators we use, grouped in five categories. Most of them are straightforward, except meta-operators, which we explain next.

7.3.2 Meta-operators

Meta-operators are useful for questions such as: *"Which musical instrument is played by the maximum number of persons?"*. To answer this question, we first compute the set of all musical instruments in the KG. For each entity in this set, we then follow the property `played by`, producing a set of people who play that instrument. Finally, we compute the max cardinality of all these sets and return the associated instrument.

The corresponding LF is the following:

```
argmax(
   cardinality(
      follow_property(
         for_each(
```

Category	Name	Signature	Description
Graph operators	`follow_property`	(SE, P) → SE	Returns the **entities** which are linked by property P **to** at least one element of SE.
	`follow_backward`	(SE, P) → SE	Returns the **entities** which are linked by property P **from** at least one element of SE.
	`get_value`	(SE, P) → SV	Returns the **values** which are linked by property P to at least one element of SE.
Numerical operators	`max`, `min`	SV → SV	Returns the max (resp. min) value from SV.
	`greater_than`, `equals`, `lesser_than`	(SV, V) → SV	Filters SV to keep values strictly greater than (resp. equal to, strictly lesser than) V.
	`cardinality`	SE → V	Returns the cardinality of SE.
Set operators	`is_in`	(a: SE, b: SE) → SV	Returns a boolean set: for each entity in a, the mask equals True if the entity is in b.
	`get_first`	SE → SE	Returns the first entity from SE.
	`union`, `intersect`, `difference`	(SE,SE) → SE	Returns the union (resp. intersection, difference) of input sets.
Class operators	`members`	SC → SE	Returns the members of classes in SC.
	`keep`	(SE,SC) → SE	Filters SE to keep the members of SC.
Meta-operators	`for_each`	SE → SE	Initializes a parallel computation over all entities in the input set.
	`arg`	SV → SE *or* SE → SE	Ends a parallel computation by returning all entities that gave a non-empty result.
	`argmax`, `argmin`	SV → SE	Ends a parallel computation by returning all entities that gave the max (resp. min) value.

Table 7.1: List of operators in our grammar. Their variables can be entities (E), classes (C), values (V), ordered sets of such elements (resp. SE, SC and SV), or properties (P).

```
        members(musical instrument),
    ),
    played by)))
```

`for_each` creates a parallel computation over each entity in its argument, which can be terminated by three operators (`arg`, `argmax` and `argmin`). We refer to Appendix 7.B.1 for details.

7.3.3 Silver LF generation

To generate silver LFs, we explore the space of LFs with a BFS strategy, similarly to Guo et al. (2018); Shen et al. (2019). More precisely, to initialize the exploration, we perform NEL to find relevant entities, values and classes that appear in the question. LFs of depth 0 simply return an annotated object. Then, assume that LFs of depth less or equal to n have been generated and we want to generate those of depth $n + 1$. We loop through all possible operators; for each operator, we choose each of its arguments among the already-generated LFs. This algorithm brings two challenges, as highlighted in Shen et al. (2020): computational cost and spurious LFs. We refer to Appendix 7.B.2 for implementation details that mitigate these difficulties.

7.3.4 Comparison with D2A

Section 7.5.5 shows that our grammar achieves higher coverage with a similar average LF depth. A more thorough quantitative comparison is delicate, as it would require reimplementing D2A within our framework, which is beyond the scope of this chapter. On a qualitative basis, we use more elementary types: in addition to theirs, we introduce set of classes, strings and set of values (which can be strings, numerals or booleans). We use eight fewer operators than D2A; among our operators, six are in common (`follow_property`, `follow_backward`, `cardinality`, `union`, `intersect`, `difference`), four are modified (`keep`, `is_in`, `argmax`, `argmin`), and the other ten are new. New intents that can be modeled include numerical reasoning (e.g. *What actor plays the younger child?*), temporal reasoning (e.g. *What is the number of seasons until 2018?*), ordinal reasoning (e.g. *What was the first episode date?*), textual form reasoning (e.g. *What was Elvis Presley given name?*). We refer to Appendix 7.C.1 for more details and comparison methodology.

7.4 Model

7.4.1 Overview

The model, called Object-Aware Transformer (OAT), is a Transformer-based (Vaswani et al., 2017) autoregressive neural semantic parser. As illustrated in Figure 7.1, the model has several steps.

The first step consists of retrieving relevant objects, annotated with metadata, that might appear in the resulting LF. This step is performed using NEL on the utterances and KG lookups to retrieve the graph context information. At this point, the input is composed of lists of *objects* with their *fields*. After embedding each field value in a vector space, we perform successive layers of input flattening and Transformer encoding. The *Flattener* layer is useful to transform the structured input into a list of embeddings. Then a decoder Transformer layer produces a linearized LF, i.e. a list of string tokens. Finally, we evaluate the LF over the KG to produce a candidate answer. In the next sections, we describe each step in details.

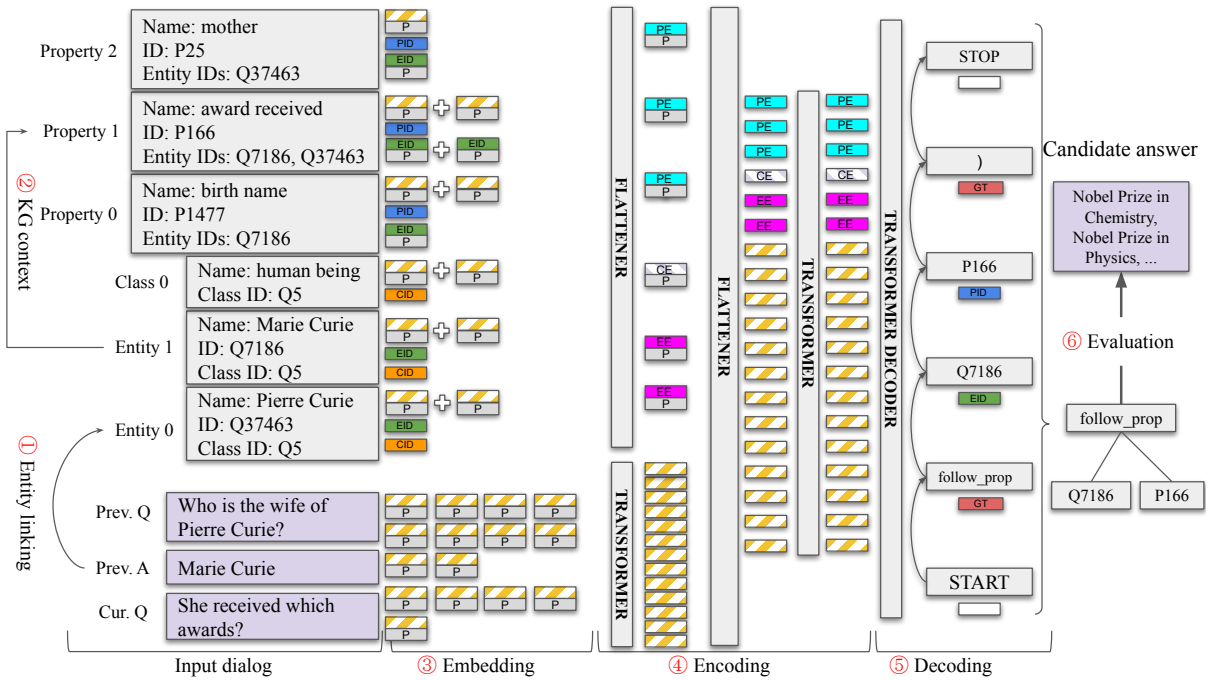


Figure 7.1: Architecture of the proposed model. The initial field embeddings are Positional (P), Property ID (PID), Entity ID (EID), and Class ID (CID). After the first Flattener layer, we obtain Property Embeddings (PE), Class Embeddings (CE), Entity Embeddings (EE). There are also Grammar Token (GT) embeddings in the output. Note that the entity IDs are actually randomized (not shown here).

7.4.2 Structured Input computation

Hierarchical structure For each query, we construct the input as a JSON-like structure, consisting of lists of objects with their fields (represented in the left part of Figure 7.1). We chose this representation as it allows incorporating general structured information into the model. A field can be a string, a KG integer ID, a numerical value, or a list thereof.

To construct the input, we start from the last utterances in the dialog: the current query, previous query and previous answer. We first perform NEL to retrieve a list of entities $\mathcal{E}$ (and numerical values) matching the utterances. The KG is then queried to retrieve additional contextual information: the classes of the entities, and all outgoing and incoming properties from these entities $\mathcal{E}$. This gives a list of properties $\mathcal{P}$. For each property $p \in \mathcal{P}$, we fill several fields: its ID, its name, and an `Entity IDs` field, which corresponds to all entities $e \in \mathcal{E}$ such that at least one *graph operator* gives a non-empty result when applied to e and p. For instance, in Wikidata, the property `birth name` (P1477) is filled for Marie Curie but not for Pierre Curie, so the `Entity IDs` field of the `birth name` property only contains Marie Curie.

Let us introduce some formal notations, useful to explain the computation of the input's embeddings (Section 7.4.3). The input is a tree where the root corresponds to the whole input, and each leaf contains the primitive values. For a non-leaf node x, we denote by $c(x)$ its children. For instance, in Figure 7.1, the node `Property 2` has three children (leaves) whose values are `mother`, `P25` and `Q37463`. A node x has also a type, and $T_{\rightsquigarrow}(x)$ denotes the types of all nodes on the path from the root to x. For instance, for the `mother` node, $T_{\rightsquigarrow}(x)$ is equal to `(root, property, name)`. In our setup, the depth of the input is at most 2.

ID randomization Directly giving the entity ID to the model would mean training a categorical classifier with millions of possible outcomes, which would lead to poor generalization. To avoid this, we replace the integer ID with a random one, thereby forcing the model to learn to predict the correct entity from the list of entities in input by copying their randomized entity ID to the output.

For numerical values, we associate each value to an arbitrary random ID, that the model should learn to copy in the output. For properties and classes, since there are fewer possibilities in the graph (a few thousand), we do not randomize them.

7.4.3 Embedding

Preprocessing We apply BERT tokenization (Devlin et al., 2018) to textual inputs. A vocabulary $\mathcal{V}_t$ is generated for each of the non-textual input types t.

Token embedding The goal of this step is to associate an embedding to each field of each object in the input. We do so by using a learned embedding for each input type: BERT embeddings (Devlin et al., 2018) for textual inputs, and categorical embeddings for non-textual inputs. When the input is a list (textual tokens or `Entity IDs` field), we add to this embedding a positional embedding. To reduce the size of the model, list embeddings are averaged into a single embedding. Formally, the embedding step associates a matrix of embeddings $h(x) \in \mathbb{R}^{1\times d_h}$ to each leaf of the input tree.

7.4.4 Encoding layers

There are two types of encoding layers: Flattener layers and Transformer layers.

Flattener The goal of these layers is to compute the embeddings of tree nodes bottom-up. They are successively applied until we are able to compute the embedding of the root node, i.e. of the whole input. This operation can be seen as flattening the JSON-like structure, hence their name.

Say we want to compute the embedding of some parent node x. An affine projection is first applied to the embedding of each child, then the embedding of the parent node is computed by applying a reduction operation $\mathcal{R}$, which can be either a sum or a concatenation. The weights of the projections are shared between all nodes having the same types $T_{\rightsquigarrow}(x)$. For example, all class name nodes - with types `(root, class, name)` - share the same weights, but they do not share the weights of entity name nodes - with types `(root, entity, name)`. Hence the embedding of x is

$$h(x) = \mathcal{R}_{y\in c(x)}\left(\left\{W^{T_{\rightsquigarrow}(y)}_{project}h(y) + b^{T_{\rightsquigarrow}(y)}_{project})\right\}\right).$$

If the reduction is a sum, all children embeddings need to be matrices of the same dimension, and the dimension of the parent embedding is also the same. If the reduction is a concatenation, the dimension of the parent embedding is $\left(\sum_{y\in c(x)} d(y), d_h\right)$.

Transformer This layer is a classical multi-head Transfomer encoder layer (Vaswani et al., 2017), taking as input a matrix of embeddings of dimensions (n, d_h), performing self-attention between the input embeddings, and outputting another matrix of the same dimensions. Detailed setup can be found in Appendix 7.B.4. We also refer to Section 1.1.3 for an introduction to the mechanics of the Transformer encoder.

Architecture We apply a first Transformer layer only to the utterances, and in parallel a first Flattener layer with sum reduction to all other inputs. The latter computes one embedding for each object. We add a positional embedding to each object, to account for its position in the list of objects of the same type. Then we apply a second Flattener layer with concatenation to all outputs of the first layer. This creates a single matrix of embeddings containing the embeddings of all the objects and utterances. Finally, a second Transformer layer is applied to this matrix.

7.4.5 Decoding layers

The goal of the decoding layers is to produce a list of tokens that corresponds to a prefix representation of the LF tree. Note that the model architecture is grammar-agnostic, in the sense that this output structure is independent of the grammar and we do not use grammar-guided decoding. The tokens can belong to one of the non-textual input types or be a grammar token. Remember that we computed a vocabulary $\mathcal{V}_t$ for all token types $t \in \mathcal{T}$. We augment the vocabulary with a STOP token.

The decoder predicts the output list of tokens iteratively. Assume that we are given the first j tokens $y_1, \ldots, y_j$. We then apply an autoregressive Transformer decoder (see Vaswani et al., 2017 or Phuong and Hutter, 2022) on the full sequence, and we condition on the last embedding h_j of the sequence to predict the next token $\hat{y}_{j+1}$. Several categorical classifiers are used to predict $\hat{y}_{j+1}$. We first decide whether we should stop decoding:

$$\hat{s}_{j+1} = \operatorname{argmax} p_{stop,j} \tag{7.1}$$

where $p_{stop,j} = \operatorname{softmax}(W_{stop} h_j)$ is a distribution over $\{0, 1\}$ given by a binary classifier. If $\hat{s}_{j+1} = 1$, the decoding is finished, and we set $\hat{y}_{j+1}$ to STOP; otherwise, we predict the type of the token:

$$\hat{t}_{j+1} = \operatorname{argmax} p_{type,j} \tag{7.2}$$

where $p_{type,j} = \operatorname{softmax}(W_{type} h_j)$ is a distribution over $\mathcal{T}$ given by a $|\mathcal{T}|$-class classifier.

Finally, depending on the predicted type, we predict the token itself

$$\hat{y}_{j+1} = \operatorname{argmax} p^{\hat{t}_{j+1}}_{token,j} \tag{7.3}$$

where $p^{\hat{t}_{j+1}}_{token,j} = \operatorname{softmax}(W^{\hat{t}_{j+1}}_{token} h_j)$ is a distribution over $\mathcal{V}_{\hat{t}_{j+1}}$ given by a $\left|\mathcal{V}_{\hat{t}_{j+1}}\right|$-class classifier.

Training We train by teacher forcing: for a given training sample $(\mathbf{x}, [y_1, \cdots, y_M])$ and for each step j, the embedding h_j is computed using the true output at previous steps: $h_j = h(y_j; \mathbf{x}, y_1, \cdots, y_{j-1})$. The loss is the cross-entropy between the true output y_{j+1} and the probability distributions produced by the model. More precisely, let $T : \bigcup_{t \in \mathcal{T}} \mathcal{V}_t \to \mathcal{T}$ be the mapping which projects tokens to their type. We denote by $p(x)$ the value of a categorical probability distribution p for the category x. Then, omitting the step subscripts j, the loss equals

$$\begin{aligned} l(p, y) = &- \log(p_{stop}(0)) \\ &- \left[\log(p_{type}(T(y))) + \log(p^{T(y)}_{token}(y))\right] \end{aligned}$$

for all steps except the last, and $l(p, y) = -\log(p_{stop}(1))$ for the last step. p_{stop}, p_{type}, and p_{token} are computed as explained above. The total loss is obtained by averaging over all training samples and over all steps.

	CSQA	ConvQuestions
Average length of a dialog	8 turns	5 turns
Possible change of topic inside a conversation	Yes	No
Answer type	Entities, boolean, quantity	Entities (usually a single one), boolean, date, quantity, string
Entity annotations in the dataset	Yes, with coreference resolution	Only the *seed entity* (topic of the dialog) and the answer entities
Coreferences in questions	Yes, to the previous turn	Yes, to any preceding turn

Table 7.2: Some characteristics of the benchmark datasets.

Methods		**D2A**	**D2A+ES**	**S2A+MAML**	**MaSP**	**OAT**
Question type	**# Ex.**			**F1**		
Simple (Direct)	82k	91.41	83.00	**92.66**	85.18	82.69
Simple (Coreferenced)	55k	69.83	64.62	71.18	76.47	**79.23**
Simple (Ellipsis)	10k	81.98	83.94	82.21	83.73	**84.44**
Logical	22k	43.62	72.93	44.34	69.04	**81.57**
Quantitative	9k	50.25	63.95	50.30	73.75	**74.83**
Comparative	15k	44.20	55.05	48.13	68.90	**70.76**
Question type	**# Ex.**			**Accuracy**		
Verification (Boolean)	27k	45.05	45.80	50.16	60.63	**66.39**
Quantitative (Count)	23k	40.94	41.35	46.43	43.39	**71.79**
Comparative (Count)	15k	17.78	20.93	18.91	22.26	**36.00**
Total Average	260k	64.47	64.75	66.54	70.56	**75.57**

Table 7.3: QA performance on CSQA. Our method is the last one (OAT). The metric is the F1 score for question types above the vertical separator, and accuracy for those under. The Total Average score is an average over all question types.

7.5 Experiments

Additional comments about the datasets, setups, and additional results can be found in the appendix.

7.5.1 Datasets

We use two weakly supervised conversational QA datasets to evaluate our method, Complex Sequential Question Answering (CSQA)[1] (Saha et al., 2018) and ConvQuestions[2] (Christmann et al., 2019), both grounded in Wikidata[3]. CSQA consists of about 1.6M turns in 200k conversations (152k/16k/28k splits), versus 56k turns in 11k conversations (7k/2k/2k splits) for ConvQuestions.

[1] `https://amritasaha1812.github.io/CSQA`
[2] `https://convex.mpi-inf.mpg.de`
[3] `https://www.wikidata.org`

CSQA was created by asking crowd workers to write turns following some predefined patterns, then turns were stitched together into conversations. The questions are organized in different categories, e.g. simple, logical or comparative questions.

For ConvQuestions, crowd workers wrote a 5-turn dialog in a predefined domain (*e.g.* Books or Movies). The dialogs are more realistic than in CSQA, however at the cost of a smaller dataset.

As presented in Table 7.2, the datasets have different characteristics, which make them an interesting test bed to assess the generality of our approach.

7.5.2 CSQA Experimental Setup

Metrics To evaluate our grammar, we report the coverage, i.e. the percentage of training questions for which we found a candidate Logical Form.

To evaluate the QA capabilities, we use the same metrics as in Saha et al. (2018). F1 Score is used for questions whose answers are entities, while accuracy is used for questions whose answer is boolean or numerical. We don't report results for "Clarification" questions, as this question type can be accurately modeled with a simple classification task, as reported in Appendix 7.A. Similarly the average metric "Overall" (as defined in Saha et al. 2018) is not reported in Table 7.3, as it depends on "Clarification", but can be found in the Appendix.

Baselines We compare our results with several baselines introduced in Section 7.2: D2A (Guo et al., 2018), D2A+ES (Shen et al., 2020), S2A+MAML (Guo et al., 2019a), and MaSP (Shen et al., 2019).

7.5.3 ConvQuestions Experimental Setup

Metrics We use the coverage as above, and the P@1 metric as defined in Christmann et al. (2019).

Baseline The only baseline to our knowledge is CONVEX (Christmann et al., 2019), which casts the problem to a node classification task. For comparison, we tried to make our setup as close as possible to theirs, and refer to Appendix 7.C.3 for details.

Data augmentation Given the small size of the dataset, we merge it with two other data sources: CSQA, and 3.6M examples generated by random sampling. The latter are single-turn dialogs made from graph triplets, *e.g.* the triplet *(Marie Curie, instance of, human)* generates the dialog: *Q: Marie Curie instance of? A: Human.* More details are given in Appendix 7.B.3. The ConvQuestions dataset is upsampled to match the other data sources sizes.

7.5.4 Named Entity Linking setup

We tried to use a similar setup as baselines for fair comparison. For CSQA, the previous gold answer is given to the model in an oracle-like mode, as in baselines. In addition, we use simple string matching between utterances and names of Wikidata entities to retrieve candidates that are given in input to the model. For ConvQuestions, we use the gold seed entity (as in the CONVEX baseline we compare with), and the Google Cloud NLP service. We refer to Appendices 7.B.2 and 7.B.4 for details.

Regarding CARTON (Plepi et al., 2021), their results are not directly comparable as their model uses gold entity annotations as input and hence is not affected by NEL errors. This different NEL setup does have a strong influence on the performance, as running our model on CSQA with a setup similar to CARTON improves our Total Average score by over 10%. We

Question type	D2A	D2A+ES	Ours
Comparative	28.6	45	**84.9**
Logical	48.2	92	**100.0**
Quantitative	58.1	62	**91.1**
Simple	94.4	96	**99.7**
Verification	77.9	85	**91.4**
Overall	74.3	80	**96.2**

Table 7.4: Coverage per question type for CSQA.

Depth	1	2	3	4+
CSQA (D2A)	0.0	47.0	30.9	22.0
CSQA (Ours)	5.5	67.9	7.4	19.2
ConvQuestions	53.9	43.7	2.4	0.0

Table 7.5: Silver LF depth distribution for both datasets.

refer to Appendix 7.D.3 for details. More generally, a more thorough study of the impact of the NEL step on the end-to-end performance would be an interesting direction of future work (see also Section 7.5.6).

7.5.5 Results

Our grammar reaches high coverage. With approximately the same numbers of operators as in baselines, we improve the CSQA coverage by 16%, as presented in Table 7.4. The improvement is particularly important for the most complex questions. We reach a coverage of 86.2% on ConvQuestions, whose questions are more varied than in CSQA.

Most queries can be expressed as relatively shallow LFs in our grammar, as illustrated by Table 7.5. This is especially interesting for the ConvQuestions dataset, composed of more realistic dialogs. On CSQA, the average depth of our LFs (2.9) is slightly lower than with D2A grammar (3.2).

We improve the QA performance over baseline on both datasets. For CSQA, our model outperforms baselines for all question types but Direct Simple questions, as shown in Table 7.3. Overall, our model improves the performance by 5%. For ConvQuestions, Table 7.6 shows that our model improves over the baseline for all domains but one, yielding an overall improvement of 4.7%. A precise evaluation of the impact of the various components of our KG-QA approach (grammar, entity linking, model inputs, model architecture, size of the training data, etc.) on the end-to-end performance was out of the scope of this chapter, and is left for future work. Nevertheless, the fact that we are able to improve over baselines for two types of Simple questions and for Logical questions, for which the grammar does not matter so much, as these question types correspond to relatively shallow LFs, suggests that our proposed model architecture is effective.

7.5.6 Error analysis

CSQA By comparing the silver and the predicted LFs on 10k random errors, we could split the errors in two main categories: first, the LF general form could be off, meaning that the model

Domain	1st turn	Follow-up	CONVEX
Books	68.1	**20.9**	19.8
Movies	54.2	**31.3**	25.9
Music	37.5	18.1	**19.0**
Soccer	43.8	**22.8**	18.8
TV	66.3	**31.8**	17.8
Overall	54.0	**25.0**	20.3

Table 7.6: ConvQuestions results by domain. The first two columns are our results. The baseline (Oracle+CONVEX) only reports follow-up turns.

did not pick up the user intent. Or the form of the LF could be right, but (at least) one of the tokens is wrong. Table 7.7 details the error statistics. The most frequent errors concern entity disambiguation. There are two types of errors: either the correct entity was not part of the model input, due to insufficient recall of the NEL system. Or the model picked the wrong entity from the input due to insufficient precision. It is known that the noise from NEL strongly affects model performance (Shen et al., 2019). We tried an oracle experiment with perfect recall NEL (see Appendix 7.D.3), which corroborates this observation, in particular for Simple questions. As we focused on modeling complex questions, improving NEL was not our main focus, but would an interesting direction for future work, in particular via multi-task approaches (Shen et al., 2019).

Error category	Overall	Simple Dir.
LF general form	31.8	24.1
Entity ID token	36.2	38.2
insuff. recall	17.1	16.8
insuff. precision	19.6	21.6
Property ID token	4.2	2.9
Class ID token	24.7	37.6
Grammar token	11.6	2.6

Table 7.7: Distribution of errors in CSQA. The numbers are (non-exclusive) percentages. We also report statistics for the Simple Direct type, as it is the largest.

ConvQuestions We manually analyzed 100 examples. Errors were mostly due to the LF general form, then to a wrong property token.

The model learns the grammar rules. In all inspected cases, the predicted LF is a valid LF according to the grammar, i.e. it could be evaluated successfully. This shows that grammar-guided decoding is not needed to achieve high performance.

7.6 Conclusion

For the problem of weakly-supervised conversational KG-QA, we proposed Object-Aware Transformer, a model capable of processing structured input in a JSON-like format. This allows to flexibly provide the model with structured KG contextual information. We also introduced a KG

grammar with increased coverage, which can hence be used to model a wider range of queries. These two contributions are fairly independent : on the one hand, since the model predicts LFs as a list of tokens, it is grammar agnostic, and thus it could be used with another grammar. On the other hand, the grammar is not tied to the model, and can be used to generate training data for other model architectures. Experiments on two datasets validate our approach. We plan to extend our model to include a richer KG context, as we believe there is significant headroom for improvements.

7.A Clarification Questions in CSQA

Take the following dialog as example:

T1	*Can you tell me which cities border Verderio Inferiore?* Cornate d'Adda, Bernareggio, Robbiate
T2	*And which cities flank that one?* Did you mean Robbiate?
T3	*No, I meant Cornate d'Adda.* Bottanuco, Busnago, Trezzo sull'Adda

The second turn is a "Clarification" question: the system asks the user for disambiguation. The disambiguation question usually takes the form "Did you mean", followed by an entity chosen among the previous turn answers. This choice appears to be entirely random. For this reason, we found that it would not be very interesting to try to predict this entity, as baselines propose. Hence we only ask the model to predict that the question is a Clarification (via a special `clarification` operator).

We report in Table 7.8 the scores for Clarification questions, as well as the "Overall" score, as defined in Saha et al. (2018). The results are not directly comparable as the baseline systems report an F1 score, while our approach uses accuracy.

Question type	**# Ex.**	**D2A**	**D2A+ES**	**S2A+MAML**	**MaSP**	**OAT (Ours)**
Clarification	12k	18.31	36.66	19.12	80.79	**99.63**
Overall	206k	62.88	72.02	N/R	79.26	**81.49**

Table 7.8: QA performance on CSQA, including "Clarification" questions. The "Overall" metric is the average F1 scores of the following question types: "Simple (Direct)", "Simple (Coreferenced)", "Simple (Ellipsis)", "Logical", "Quantitative", "Comparative" and "Clarification".

7.B Detailed experimental setup

7.B.1 Meta-operators

Take the example given in the main part of the chapter: *"Which musical instrument is played by the maximum number of persons?"*. The corresponding LF is:

```
argmax(
   cardinality(
      follow_property(
         for_each(
            members(musical instrument)),
```

```
    played by)))
```

Assume that the KG contains exactly two musical instruments, piano and violin, or in other words, `members(musical instrument)` equals `{piano, violin}`.

`for_each` creates a dictionary of entities. Each (key, value) pair corresponds to one entity in the argument of `for_each`, where the key is the entity itself and the value is a singleton set containing the entity. Here `for_each({piano, violin})` gives the following dictionary:

```
{
   piano:  {piano},
   violin:  {violin}
}
```

We then apply the same computation to each of the dictionary values, while keeping the keys untouched. In our example, we apply the expression

```
cardinality(
   follow_backward(., played by)
),
```

which gives the result `{piano:  20392, violin:  7918}`.

Finally, an aggregation operator is computed over the values, and the result is a subset of the keys. In the example, `argmax` returns the set of keys associated with the maximum values, here `{piano}`. In other cases, we want to return all the keys associated to a non-empty value, `arg` allows doing so.

7.B.2 Silver LF generation

Wikidata version For CSQA, we used the preprocessed version of Wikidata made available by the authors, which contains 21.2M triplets over 12.8M entities and 567 distinct properties. For ConvQuestions, we used a more recent version of Wikidata, containing 1.1B triplets over 91.8M entities and 7869 distinct properties.

Named Entity Linking For ConvQuestions, we use gold entity annotations and Google Cloud NLP entity linking service. For CSQA, we use gold entity annotations.

To resolve the coreferences, in ConvQuestions, we use entity annotations from previous utterances during the silver LF generation step. In CSQA, since coreferences are already resolved by the gold annotations, we just use annotations from the current utterance.

Simplifying the BFS We observed that reaching a depth of 4+ is needed for some queries (see Table 7.5 of the main part of the chapter), but is impractical by exhaustive BFS, as the size of the space of LFs grows very quickly with their depth. To improve the efficiency, we used the following ideas:

- **Stopping criteria to abort the exploration:** timeout $t_{\max}$ and maximum depth $d_{\max}$.
- **Type checking:** by leveraging the operators' signatures (presented in Table 7.1 of the main part), we only construct legal LFs.
- **Putting constraints on the form of the LF:** we manually forbid certain combinations of operators, e.g. `follow_backward` after `follow_property`.
- **Restriction of the list of operators:** for ConvQuestions, we use the graph operators, the numerical operators, `is_in`, and `get_first`. The removal of some set operators and of

meta-operators strongly reduces the complexity of the BFS. For CSQA, all operators are needed, but we add more constraints in order to keep the BFS simple enough.

We choose $d_{\max} = 3$ for ConvQuestions and $d_{\max} = 7$ for CSQA, and $t_{\max} = 1200$ seconds.

All LFs found by BFS are evaluated over the KG, which gives candidate answers. We keep the LFs whose candidate answers have the highest F1 score w.r.t. the gold answer. The minimal F1 score for keeping a LF is 0.3.

Scores for LF ranking The BFS often returns several LFs (with the top F1 score, as explained above), among which some are spurious: they do not correspond to the semantic meaning of the question, but their evaluation over the KG yields the correct result by chance. As we keep only one for training, we need a way to rank the candidate LFs. We use the following heuristic scores to do so:

- **Complexity:** the score is $1 - {}^{(d-1)}/_{(d_{\max}-1)}$ where d is the depth of the LF and $d_{\max}$ is defined above.
- **Property lexical matching:** for each property appearing in the LF, we compute the Jaccard index of the words appearing in its name and of the words of the question.
- **Annotation coverage:** among the entities retrieved by NEL, we compute the percentage of entities which appear in the LF.

As these three scores are between 0 and 1, we average them and keep the LF with the highest total score. We found that this simple method is a good way to reduce spurious LFs, which are often either too complex or not matching lexically the question.

7.B.3 Random examples generation

To generate the random examples for data augmentation for ConvQuestions training, we first sample uniformly 80k entities from the graph. Then, for each entity, we generate a conversation for each triplet that links it to other entities. The question text is made by stitching the entity name and the property name. For instance, the triplet `(Marie Curie, native language, Polish)` generates the dialog: *Q: Marie Curie native language? A: Polish.* We also generate variants where the property name is replaced by aliases, which are alternative names in Wikidata, e.g. *mother tongue* for `native language`. When the question or answer has more than 256 characters, we eliminate it.

7.B.4 Modeling

Wikidata version As in 7.B.2.

Named Entity Linking NEL is performed again, this time to create the structured context in input to the model. Due to the randomization step described in Section 7.4.2, missing entities in the input cannot be retrieved by the model, so we want to have a high NEL recall. The trade-off is that the NEL precision is low, meaning that we have many spurious entities in the input, which the model has to learn to ignore.

For CSQA, we use simple string matching between the utterances and the names of Wikidata entities. Note that in our model, as well as in all CSQA baselines (in particular Guo et al., 2018; Shen et al., 2019), the previous gold answer is given as input to the model in an oracle-like setup.

For ConvQuestions, we use the gold seed entity and the Google Cloud NLP entity linking service.

To resolve coreferences, we use entities from the dialog history: all preceding turns for ConvQuestions and only the previous turn for CSQA.

Implementation details We tokenize the input using the BERT-uncased tokenizer. All embeddings in the model have dimension 768. The two transformer encoders share the same configuration: 2 layers each, with output dimension 768, intermediate dimension 2048, 12 attention heads, and dropout probability set to 0.1. The model has 260M parameters. The transformer implementation is based on publicly available BERT code. We initialize the word embedding from a BERT checkpoint, but do *not* load the transformer layer weights, instead training them from scratch. We train for 600k steps with batch size 128, using the ADAM optimizer (Kingma and Ba, 2015) with learning rate 3×10^{-5}. Training takes around 14 hours on 16 TPU v3 with 32 cores.

7.C Comparison with baselines

7.C.1 Comparison with D2A grammar

Intent	**Question example**	**Missing in D2A**
Textual form reasoning	What was Elvis Presley given name?	string type
Numerical reasoning	What actor plays the younger child?	`get_value`, `for_each`, `argmin`
Numerical reasoning	How old is the younger child?	`min`
Selection of the members of a class	Which television programs have been dubbed by at least 20 people ?	`members`
Temporal reasoning	What is the number of seasons until 2018?	`for_each`, `get_value`, `lesser_than`, `arg`
Ordinal reasoning	What was the first episode date?	`get_first`

Table 7.9: Examples of questions that are difficult to model with the D2A grammar. Examples are mostly chosen from ConvQuestions, as their questions look more realistic than CSQA.

The D2A (Guo et al., 2018) grammar is the main baseline in previous KG-QA works. We compare with the grammar implemented in their open-sourced code[4], which is a bit different from the published one. Numbers for D2A in Tables 7.4 and 7.5 were computed thanks to the results of the BFS gracefully provided by the authors.

Table 7.9 presents some intents which we are able to model in our grammar and are not straightforward to model with D2A grammar. First, *textual form reasoning* corresponds to questions about string attributes of entities, which are not included in the D2A grammar. Second, to handle *numerical and temporal reasoning*, computations based on numerical values are needed, which is not possible with the D2A grammar. Finally, the D2A grammar does not model the order of relations in the graph and the selection of class members, which we start to tackle with respectively the `get_first` and `members` operators.

[4] https://github.com/guoday/Dialog-to-Action/blob/bb2cbb9de474c0633bac6d01c10eca24c79b951f/BFS/parser.py

7.C.2 Comparison with MaSP architecture

Similarly to ours, the MaSP (Shen et al., 2019) model follow the semantic parsing approach, where the LF is encoded as a sequence of operators and graph items IDs. Regarding the model input, theirs consist only of the utterances, whereas we add additional KG context structured as a JSON tree. The training method is different: MaSP uses multi-task learning to learn jointly entity linking and semantic parsing, whereas we chain both, and trust the model to pick the good entity. Our approach is simpler in this regard, but we pay this by having a slightly lower performance on Simple Direct questions (see Table 7.3). Finally, we do not use beam search for the decoding, contrarily to them.

7.C.3 Comparison with node classification approaches

An alternative to the semantic parsing approach is to train classifiers to predict entities as nodes of the KG. A precise comparison of both approaches is out of the scope of this chapter. Nevertheless, we think that the semantic parsing approach is better suited to our purpose of modeling complex questions. For instance, complex intents involving numerical comparisons can be expressed naturally by a LF, but would be difficult to perform using solely node classifiers. Examples include the numerical and temporal reasoning in Table 7.9. Additional examples include *The series consists of which amount of books?* (ConvQuestions) or *Which television programs have been dubbed by at least 20 people ?, How many episodes is it longer than the second longest season out of the three?* (CSQA).

Convex (Christmann et al., 2019) is an example of such an approach. It is an unsupervised graph exploration method: at each turn, a subgraph is expanded by matching the utterance with neighboring entities. Then a candidate answer is found in the subgraph by a node classifier. On our side, we propose a semantic parsing approach that makes use of entities annotated by an external entity linking service. This is a similar setup to the CSQA baselines (Shen et al., 2019), which we re-purposed for ConvQuestions in order to assess the quality of our proposal on another dataset. In order to be closer to the Convex baseline, we changed our CSQA setup by applying the entity linker only to the questions' text and not to the answers' text. In addition, as we use the gold seed entity, we compare with the Oracle+Convex setup of Christmann et al. (2019), which also uses the gold seed entity (and the gold first turn answer entity). Finally, we make use of data augmentation to train our model on ConvQuestions, whereas the baseline does not.

7.C.4 BERT or no BERT, that is the question

The baselines of Table 7.3 do not use BERT. MaSP authors provide an additional BERT variant of their model that uses a fine-tuned BERT base architecture. The Total Average score of this variant is 72.60%, which is 2% above their vanilla variant and 3% under our model. Since we are only using the word embeddings (loaded from a publicly available BERT base checkpoint) and not loading the transformer layer weights, we decided to compare with the vanilla variant of MaSP, and not the BERT one. Finally, CARTON is using a pre-trained BERT base model as a sentence encoder.

7.D Additional results

7.D.1 Coverage results

We present in Table 7.10 the coverage per domain for ConvQuestions. Besides, the evolution of the coverage over turns is stable for both datasets, hence we do not report this result.

Domain	Coverage
Books	88.6
Movies	87.6
Music	90.0
Soccer	78.6
TV	86.2
Overall	86.2

Table 7.10: ConvQuestions coverage per domain.

7.D.2 Performance over turns

Tables 7.11 and 7.12 show the evolution of the performance over turns for both datasets. For CSQA, the performance drops after the first two turns, then remains constant. For ConvQuestions, the performance decreases throughout the turns. There is a sharp decrease after the first turn, probably because it is simpler as there is no coreference or ellipsis. The different behavior between the datasets may be due to the realism of ConvQuestions.

Turns	**0**	**1**	**2**	**3**	**4**
Score	85	87	75	74	74

Turns	**5-6**	**7-8**	**9-10**	**11-12**	**13+**
Score	75	75	75	74	74

Table 7.11: Average performance over turns for CSQA. For brevity, we average over turn ranges after turn 5.

Turn	**0**	**1**	**2**	**3**	**4**
Av. P@1	54	35	20	29	15

Table 7.12: Performance over turns for ConvQuestions.

7.D.3 Oracle setup and comparison with CARTON

CARTON (Plepi et al., 2021) gives the entities annotated in the dataset as part of the model input (entities appearing in the previous turn and in the current question), contrarily to the models in Table 7.3 which all use an entity linker. For a fair comparison, we tested our model in an oracle setup, where we also give the gold annotations as input. As shown in Table 7.13, the Total Average score of our model increases by 10% w.r.t. the baseline approach. The improvement is particularly important for the most simple question types (Simple and Logical Questions). In this setup, our performance is 8% higher than CARTON, and we obtain a better score for 7 out of 10 question types.

7.D.4 Further error analysis

An alternative approach for error analysis is to assess the performance of the decoding classifiers (see Section 7.4.5) in a teacher forcing setup, *i.e.* to assess how often they predict the next token

Question type	CARTON	Ours
Simple (Direct)	85.92	**96.95**
Simple (Coreferenced)	87.09	**94.77**
Simple (Ellipsis)	85.07	**96.66**
Logical	80.80	**95.54**
Quantitative	**80.62**	76.44
Comparative	62.00	**76.66**
Verification (Boolean)	**77.82**	67.02
Quantitative (Count)	57.04	**75.89**
Comparative (Count)	**38.31**	35.10
Total Average	77.89	**85.85**

Table 7.13: QA performance on CSQA in oracle mode.

Metric	CSQA	ConvQuestions
Token type	99.88	94.77
Grammar token	98.86	82.12
Entity ID	92.47	50.79
Property ID	99.45	30.26
Class ID	94.64	N/A
Numerical value	99.91	N/A
Avg. token	97.70	61.53

Table 7.14: LF token accuracy metrics, on the evaluation splits. For ConvQuestions, Class ID, numerical value and their relative operators are not used (see 7.B.2).

correctly, given the true previous tokens. Table 7.14 reports the results on the evaluation split of both datasets. The results corroborate the analysis presented in Section 7.5.6. First, the model learns the grammar rules, as it nearly always predicts the good token type. For CSQA, the most frequent errors concern entity ID and class ID. For ConvQuestions, they concern primarily entities and properties.

7.D.5 Case study

Table 7.15 presents examples from ConvQuestions where we are able to predict the good LFs, although there exists very similar properties in the graph. The textual forms of the questions are not sufficient to infer the good property to use, implying that the model had to learn elements from the graph structure in order to answer correctly these questions. Nevertheless, Table 7.14 shows that there is still significant room for improvement in that direction.

Question	Property
When did Seinfeld first air?	start time (P580)
When did Camp Rock come out?	publication date (P577)
Who screen wrote it?	screenwriter (P58)
Who wrote it?	author (P50)
What country are they from?	country (P17)
Belleville of which country?	country (P17)
What country did the band Black Sabbath originally come from?	country of origin (P945)
What country is Son Heung-min from originally?	country for sport (P1532)

Table 7.15: Examples of ConvQuestions questions for which the model was able to pick up the good property, although there are very similar properties in the graph.

8 Conclusion

Despite its immense successes, paving the way to superhuman performance of computers in numerous intellectual endeavors, deep learning remains a strikingly data-inefficient and energy-inefficient method, in particular compared to the human brain. Innovation towards more efficient approaches will come from a combination of various efforts, among which striving for more mathematically-grounded approaches may occupy a foremost role. In this thesis, we presented several contributions to deep learning research, and in particular to the theory of deep learning, which we hope can contribute at their scale to this effort.

An important axis in our work was to leverage the large realm of differential equations to study properties of neural networks, be it via the continuous-depth approach in the first part of the thesis, or with the two-timescale (a.k.a. fast-slow) regime for dynamical systems in Chapter 6. Of course, these contributions only scratch the surface of the interplay between deep learning and analysis of differential equations, and there are many more subjects of interest regarding this bridge between two behemoths of modern applied sciences. As a conclusive note, let us sketch a few final comments and perspectives relative to the topics of the thesis.

From neural ODEs to neural SDEs. A major focus in this PhD thesis is the study of the connection between deep residual networks and neural ordinary differential equations. In Chapter 2, we highlight conditions under which this connection holds true at initialization, namely that the residual branch should be rescaled by a factor inversely proportional to the depth of the network, and that the weights should be initialized with correlations across depth. In Chapter 3, we show that the neural ODE limit then also holds during training. Under an additional assumption on the width of the network, we show a Polyak-Łojasiewicz inequality, which allows proving convergence of the training algorithm towards a neural ODE that interpolates the training data. Finally, in Chapter 4, we show that neural ODEs, and subsequently deep ODE-like residual networks, satisfy generalization bounds.

It would be of particular significance to investigate whether these results can extend to neural stochastic differential equations, which we showed in Chapter 2 to be another possible deep limit for residual networks. The extension to neural SDEs is important for at least two reasons. First, as we showed in Chapter 2, standard i.i.d. initialization schemes seem to rather correspond to SDEs than to ODEs. Moving from neural ODEs to neural SDEs would therefore bring our analysis closer to the practice. Second, stochasticity has empirically been shown to play a key role in deep learning training, for instance via dropout (Hinton et al., 2012b) or stochastic gradient descent. Recent theoretical results prove that stochasticity enables the optimization algorithm to reach minima with better generalization properties (Pesme et al., 2021). As a consequence, understanding the effect of stochasticity due to initialization would be very interesting, and the

comparison of the neural SDE and neural ODE limits could be a fruitful axis to do so.

Moving away from supervised learning. All our results are presented in a supervised learning setup, either for regression or classification. Could they be extended to other learning contexts? In particular, several recent generative models, among which the acclaimed diffusion models, can be written as a neural ODE (Song et al., 2021), where the ODE flow corresponds to a transport equation mapping Gaussian noise to the target distribution. Understanding how to translate to this sampling context the theoretical results obtained in supervised learning would be worthwhile.

Better characterizing the implicit regularization for deep residual networks. Chapters 3 and 4 show a gap between the optimization and statistical points of views: our global convergence result in Chapter 3 requires the width of the residual network to grow linearly with the sample size, while on the contrary our statistical result of Chapter 4 requires the sample size to be larger than the width. The literature on implicit regularization presented in Section 1.1.2.2 suggests that it is unlikely that this gap could be bridged by refinements of the current approach. A more promising direction would be to obtain further results on the implicit regularization of gradient algorithms for deep residual networks. More precisely, we showed in Chapter 3 that the gradient flow for a class of residual networks converges towards a neural ODE, but it remains to understand more finely the properties of this limiting neural ODE, and in particular its generalization abilities.

Extending global convergence results to other settings. Chapters 3 and 6 present global convergence results for neural networks obtained with very different techniques, either through a Polyak-Łojasiewicz inequality or by studying explicitly the gradient flow dynamics. Adapting these techniques to other architectures would be particularly beneficial. The proof technique of Chapter 6, which enables us to prove convergence of the gradient flow for sigmoid networks to approximate one-dimensional piecewise constant target functions, is delicate to adapt to a multidimensional setting, but other one-dimensional settings such as using ReLU networks to learn piecewise affine functions could be considered. On the other side, the proof technique of Chapter 3 via a Polyak-Łojasiewicz inequality is fairly general, and it might be applicable to other residual architectures such as Transformer.

Going beyond on the Transformer architecture. The contributions of Chapter 7 are a first attempt at tackling the numerous questions raised by Transformer. In particular, an overarching question is to understand to what extent the striking results of this architecture are linked to the form of the attention mechanism. More precisely, is there something specific to attention which allows it to model particularly well some sequential data? Or is attention an instance of a larger class of mechanisms that work just as well? For instance, Lee-Thorp et al. (2022) propose to replace the attention layer by a non-parametrized Fourier layer. This layer computes the 2D Fourier transform of the sequence of embeddings. The performance trails not much behind the standard Transformer, and the authors suggest that the "token-mixing" property of the Fourier layer is enough to obtain good performance. A mathematical analysis of their results, and of the comparison with standard attention layers, would be of the utmost interest.

Bibliography

M. Ahmed, M. R. Samee, and R. E. Mercer. Improving Tree-LSTM with Tree Attention. In *2019 IEEE 13th International Conference on Semantic Computing*, pages 247–254, 2019. (p. 214)

N.-J. Akpinar, B. Kratzwald, and S. Feuerriegel. Sample complexity bounds for recurrent neural networks with application to combinatorial graph problems. *arXiv:1901.10289*, 2019. (p. 143)

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 242–252, 2019. (p. 18, 80, 178, 181)

M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*, volume 9. Cambridge University Press, 1999. (p. 14, 17)

V. Arnold. *Ordinary Differential Equations*. Springer Textbook. Springer Berlin Heidelberg, 1992. (p. 114, 132, 184)

D. Arpit, V. Campos, and Y. Bengio. How to initialize your network? Robust initialization for WeightNorm & ResNets. 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, pages 10902–10911. Curran Associates, Inc., 2019. (p. 40, 41, 43, 48)

R. G. Athreya, S. K. Bansal, A.-C. N. Ngomo, and R. Usbeck. Template-based Question Answering using Recursive Neural Networks. In *2021 IEEE 15th International Conference on Semantic Computing*, pages 195–198, 2021. (p. 214)

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18:1–53, 2017. (p. 19)

F. Bach. Effortless optimization through gradient flows, 2020. Blog post. URL: `https://francisbach.com/gradient-flows/` (version: 2023-06-14). (p. 25)

F. Bach. Learning theory from first principles, 2023. Book draft. URL: `https://www.di.ens.fr/~fbach/ltfp_book.pdf` (version: 2023-02-05). (p. 15, 27, 34, 127)

T. Bachlechner, B. Majumder, H. Mao, G. Cottrell, and J. Auley. ReZero is all you need: Fast convergence at large depth. In C. de Campos and M. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pages 1352–1361. PMLR, 2021. (p. 42)

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014. (p. 20)

R. Barboni, G. Peyré, and F.-X. Vialard. On global convergence of ResNets: From finite to infinite width using linear parameterization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16385–16397. Curran Associates, Inc., 2022. (p. 80, 81)

P. Bartlett, D. Helmbold, and P. Long. Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 521–530. PMLR, 2018. (p. 80)

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002. (p. 149, 169, 171)

P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6240–6249. Curran Associates, Inc., 2017. (p. 17, 83, 124, 130, 139)

P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20:1–17, 2019. (p. 17, 122)

C. Bayer, P. K. Friz, and N. Tapia. Stability of deep neural networks via discrete rough paths. *SIAM Journal on Mathematics of Data Science*, 5:50–76, 2023. (p. 42, 57)

M. Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021. (p. 18)

M. Belkin, S. Ma, and S. Mandal. To understand deep learning we need to understand kernel learning. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 541–549. PMLR, 2018. (p. 142)

Y. Bengio, P. Frasconi, and P. Simard. The problem of learning long-term dependencies in recurrent networks. In *1993 IEEE International Conference on Neural Networks*, pages 1183–1188, 1993. (p. 153)

M. Benning, E. Celledoni, M. J. Ehrhardt, B. Owren, and C.-B. Schönlieb. Deep learning as optimal control problems: Models and numerical methods. *Journal of Computational Dynamics*, 6:171–198, 2019. (p. 23)

N. Berglund and B. Gentz. *Noise-induced phenomena in slow-fast dynamical systems: a sample-paths approach*. Springer Science & Business Media, 2006. (p. 180)

L. Béthune, T. Boissin, M. Serrurier, F. Mamalet, C. Friedrich, and A. G. Sanz. Pay attention to your loss : understanding misconceptions about lipschitz neural networks. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 20077–20091. Curran Associates, Inc., 2022. (p. 124)

A. Bietti and J. Mairal. Invariance and stability of deep convolutional representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6210–6220. Curran Associates, Inc., 2017. (p. 142, 143)

A. Bietti and J. Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *Journal of Machine Learning Research*, 20:1–49, 2019. (p. 142)

A. Bietti, G. Mialon, D. Chen, and J. Mairal. A kernel perspective for regularizing deep neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 664–674. PMLR, 2019. (p. 142, 151, 174)

M. Biloš, J. Sommer, S. S. Rangapuram, T. Januschowski, and S. Günnemann. Neural flows: Efficient alternative to neural ODEs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21325–21337. Curran Associates, Inc., 2021. (p. 123)

K. Border. Miscellaneous notes on optimization theory and related topics, 2015. URL: `https://healy.econ.ohio-state.edu/kcb/AddedByPJ/Maximization.pdf` (version: 2023-02-08). (p. 182)

V. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29:291–294, 1997. (p. 178, 180)

V. Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009. (p. 180)

L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60:223–311, 2018. (p. 178)

E. Boursier, L. Pillaud-Vivien, and N. Flammarion. Gradient flow dynamics of shallow relu networks for square loss and orthogonal inputs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 20105–20118. Curran Associates, Inc., 2022. (p. 19, 80)

A. Brock, S. De, and S. Smith. Characterizing signal propagation to close the performance gap in unnormalized ResNets. In *International Conference on Learning Representations*, 2021. (p. 40)

A. Brutzkus, A. Globerson, E. Malach, and S. Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. In *International Conference on Learning Representations*, 2018. (p. 17)

R. Cao, S. Zhu, C. Yang, C. Liu, R. Ma, Y. Zhao, L. Chen, and K. Yu. Unsupervised dual paraphrasing for two-stage semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6806–6817. Association for Computational Linguistics, 2020. (p. 212)

B. Chang, M. Chen, E. Haber, and E. H. Chi. AntisymmetricRNN: A dynamical system view on recurrent neural networks. In *Proceedings of the 7th International Conference on Learning Representations*, 2019. (p. 23, 42, 142)

K.-T. Chen. Integration of paths–a faithful representation of paths by non-commutative formal power series. *Transactions of the American Mathematical Society*, 89:395–407, 1958. (p. 142)

M. Chen, X. Li, and T. Zhao. On generalization bounds of a family of recurrent neural networks. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1233–1243. PMLR, 2020. (p. 150)

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 6571–6583. Curran Associates, Inc., 2018a. (p. 22, 23, 41, 42, 51, 78, 79, 81, 122, 125, 142, 153)

X. Chen, C. Liu, and D. Song. Tree-to-tree neural networks for program translation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 2547–2557. Curran Associates, Inc., 2018b. (p. 214)

I. Chevyrev and A. Kormilitzin. A primer on the signature method in machine learning. *arXiv:1603.03788*, 2016. (p. 145, 146)

L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 3036–3046. Curran Associates, Inc., 2018. (p. 18, 178, 181)

L. Chizat, E. Oyallon, and F. 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, pages 2937–2947. Curran Associates, Inc., 2019. (p. 19)

K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. Association for Computational Linguistics, 2014. (p. 142)

Y. Cho and L. Saul. Kernel methods for deep learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 342–350. Curran Associates, Inc., 2009. (p. 142)

F. Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015. (p. 45)

P. Christmann, R. S. Roy, A. Abujabal, J. Singh, and G. Weikum. Look before you Hop: Conversational Question Answering over Knowledge Graphs Using Judicious Context Expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 729–738, 2019. (p. 30, 31, 213, 220, 221, 228)

A.-S. Cohen, R. Cont, A. Rossier, and R. Xu. Scaling properties of deep residual networks. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2039–2048. PMLR, 2021. (p. 42, 43, 52, 57, 80, 89, 132)

W. W. Cohen, M. Siegler, and A. Hofer. Neural Query Language: A Knowledge Base Query Language for Tensorflow. *arXiv:1905.06209*, 2019. (p. 213)

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011. (p. 142)

R. Cont, A. Rossier, and R. Xu. Convergence and implicit regularization properties of gradient descent for deep residual networks. *arXiv:2204.07261*, 2022. (p. 80, 89)

C. Cuchiero, M. Larsson, and J. Teichmann. Deep neural networks, generic universal interpolation, and controlled odes. *SIAM Journal on Mathematics of Data Science*, 2:901–919, 2020. (p. 23)

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989. (p. 17)

S. De and S. Smith. Batch normalization biases residual blocks towards the identity function in deep networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19964–19975. Curran Associates, Inc., 2020. (p. 40, 41, 42)

E. De Brouwer, J. Simm, A. Arany, and Y. Moreau. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. 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, pages 7379–7390. Curran Associates, Inc., 2019. (p. 142)

L. Debnath and D. Bhatta. *Integral Transforms and Their Applications*. CRC press, Boca Raton, third edition, 2014. (p. 108)

L. Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29:141–142, 2012. (p. 57)

P. Deuflhard and S. Röblitz. Parameter identification in ODE models. In *A Guide to Numerical Modelling in Systems Biology*, pages 89–138. Springer International Publishing, 2015. (p. 127)

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018. (p. 218)

R. DeVore, B. Hanin, and G. Petrova. Neural network approximation. *Acta Numerica*, 30: 327–444, 2021. (p. 17)

C. Dong, L. Liu, Z. Li, and J. Shang. Towards adaptive residual network training: A neural-ODE perspective. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2616–2626. PMLR, 2020. (p. 22, 78)

L. Dong and M. Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 33–43. Association for Computational Linguistics, 2016. (p. 212, 213)

L. Dong and M. Lapata. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 731–742. Association for Computational Linguistics, 2018. (p. 212, 213)

S. S. Dragomir. *Some Gronwall Type Inequalities and Applications*. Nova Science Publishers, 2003. (p. 93)

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1675–1685, 2019. (p. 18, 178, 181)

E. Dupont, A. Doucet, and Y. W. Teh. Augmented neural odes. 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, pages 3140–3150. Curran Associates, Inc., 2019. (p. 81)

W. E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5:1–11, 2017. (p. 23, 122)

W. E, J. Han, and Q. Li. A mean-field optimal control formulation of deep learning. *Research in the Mathematical Sciences*, 6:10, 2019. (p. 22, 23, 41, 42, 78)

N. B. Erichson, O. Azencot, A. Queiruga, L. Hodgkinson, and M. W. Mahoney. Lipschitz recurrent neural networks. In *International Conference on Learning Representations*, 2021. (p. 142)

A. Fermanian. Embedding and learning with signatures. *Computational Statistics & Data Analysis*, 157:107148, 2021. (p. 142, 146)

A. Fermanian, P. Marion, J.-P. Vert, and G. Biau. Framing RNN as a kernel method: A neural ODE approach. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3121–3134. Curran Associates, Inc., 2021. (p. 9)

A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Springer, Dordrecht, 1988. (p. 115)

F. Fleuret. *The Little Book of Deep Learning*. Université de Genève, 2023. (p. 14)

S. Frei, Y. Cao, and Q. Gu. Algorithm-dependent generalization bounds for overparameterized deep residual networks. 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, pages 14797–14807. Curran Associates, Inc., 2019. (p. 80)

P. Friz and N. Victoir. Euler estimates for rough differential equations. *Journal of Differential Equations*, 244:388–412, 2008. (p. 147)

P. K. Friz and N. B. Victoir. *Multidimensional Stochastic Processes as Rough Paths: Theory and Applications*, volume 120 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2010. (p. 142, 145, 151)

J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops*, pages 50–56, 2018. (p. 151)

B. Geshkovski, C. Letrouit, Y. Polyanskiy, and P. Rigollet. The emergence of clusters in self-attention dynamics. *arXiv:2305.05465*, 2023. (p. 86)

P. Gierjatowicz, M. Sabate-Vidales, D. Šiška, L. Szpruch, and Z. Zurič. Robust pricing and hedging via neural sdes. *arXiv:2007.04154*, 2020. (p. 122)

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256. PMLR, 2010. (p. 45, 78)

S. Goel, A. Gollakota, Z. Jin, S. Karmalkar, and A. Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In H. Daumé III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3587–3596. PMLR, 2020. (p. 113)

S. Goldt, M. Advani, A. Saxe, F. Krzakala, and L. Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *Journal of Statistical Mechanics: Theory and Experiment*, 2020:124010, 2020. (p. 181)

N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*. PMLR, 2018. (p. 124, 130, 131)

A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse. The reversible residual network: Backpropagation without storing activations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (p. 79)

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. (p. 14, 20, 44, 81, 82, 177)

L.-A. Gottlieb, A. Kontorovich, and R. Krauthgamer. Efficient regression in metric spaces via approximate lipschitz extension. *IEEE Transactions on Information Theory*, 63:4838–4849, 2017. (p. 126)

W. Grathwohl, R. T. Q. Chen, J. Bettencourt, and D. Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, 2019. (p. 23, 42)

A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013. (p. 142)

D. Guo, D. Tang, N. Duan, M. Zhou, and J. Yin. Dialog-to-action: Conversational question answering over a large-scale knowledge base. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 2942–2951. Curran Associates, Inc., 2018. (p. 212, 213, 214, 216, 221, 226, 227)

D. Guo, D. Tang, N. Duan, M. Zhou, and J. Yin. Coupling retrieval and meta-learning for context-dependent semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 855–866. Association for Computational Linguistics, 2019a. (p. 213, 221)

J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J.-G. Lou, T. Liu, and D. Zhang. Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535. Association for Computational Linguistics, 2019b. (p. 213)

F. Götze and J. Jalowy. Rate of convergence to the circular law via smoothing inequalities for log-potentials. *Random Matrices: Theory and Applications*, 10:2150026, 2021. (p. 54)

E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse problems*, 34: 014004, 2017. (p. 22, 23, 78, 122)

W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 1024–1034. Curran Associates, Inc., 2017. (p. 214)

B. Hanin and D. Rolnick. How to start training: The effect of initialization and architecture. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 569–579. Curran Associates, Inc., 2018. (p. 41, 42)

J. Hanson and M. Raginsky. Fitting an immersed submanifold to data via sussmann's orbit theorem. In *2022 IEEE 61st Conference on Decision and Control*, pages 5323–5328, 2022. (p. 79)

J. Harer, C. Reale, and P. Chin. Tree-Transformer: A Transformer-Based Method for Correction of Tree-Structured Data. *arXiv:1908.00449*, 2019. (p. 214)

S. Hayou. On the infinite-depth limit of finite-width neural networks. *Transactions on Machine Learning Research*, 2023. (p. 80)

K. He, X. Zhang, S. Ren, and J. 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. IEEE Computer Society, 2015. (p. 40, 45, 78)

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016a. (p. 20, 22, 40, 43, 78, 119, 122)

K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645. Springer International Publishing, 2016b. (p. 40, 81, 125)

D. Hendrycks and K. Gimpel. Gaussian Error Linear Units (GELUs). *arXiv:1606.08415*, 2016. (p. 86)

C. Herrera, F. Krach, and J. Teichmann. Theoretical guarantees for learning conditional expectation using controlled ODE-RNN. *arXiv:2006.04727*, 2020. (p. 142)

J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. Eisenschlos. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333. Association for Computational Linguistics, 2020. (p. 214)

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6626–6637. Curran Associates, Inc., 2017. (p. 180)

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29: 82–97, 2012a. (p. 142)

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012b. (p. 233)

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997. (p. 142, 153)

M. Hong, H.-T. Wai, Z. Wang, and Z. Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33:147–180, 2023. (p. 180)

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 328–339, 2018. (p. 181)

C.-W. Huang, J. H. Lim, and A. C. Courville. A variational perspective on diffusion-based generative models and score matching. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22863–22876. Curran Associates, Inc., 2021. (p. 123)

W. Hwang, J. Yim, S. Park, and M. Seo. A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization. *arXiv:1902.01069*, 2019. (p. 213)

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456. PMLR, 2015. (p. 40)

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in 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, pages 8580–8589. Curran Associates, Inc., 2018. (p. 18, 142, 178, 181)

S. Jastrzkebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Three factors influencing minima in SGD. *arXiv:1711.04623*, 2017. (p. 41)

R. Jia and P. Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 12–22. Association for Computational Linguistics, 2016. (p. 212)

J. Kelly, J. Bettencourt, M. J. Johnson, and D. K. Duvenaud. Learning differential equations that are easy to solve. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4370–4380. Curran Associates, Inc., 2020. (p. 131, 174)

P. Kidger. *On Neural Ordinary Differential Equations*. PhD thesis, 2022. (p. 22, 23, 78)

P. Kidger and T. Lyons. Signatory: Differentiable computations of the signature and logsignature transforms, on both CPU and GPU. In *International Conference on Learning Representations*, 2021. (p. 146, 174)

P. Kidger, P. Bonnier, I. Perez Arribas, C. Salvi, and T. Lyons. Deep signature transforms. 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, pages 3099–3109. Curran Associates, Inc., 2019. (p. 142)

P. Kidger, J. Morrill, J. Foster, and T. Lyons. Neural controlled differential equations for irregular time series. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6696–6707. Curran Associates, Inc., 2020. (p. 23, 122, 142, 144)

P. Kidger, J. Foster, X. C. Li, and T. Lyons. Efficient and accurate gradients for neural sdes. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18747–18761. Curran Associates, Inc., 2021. (p. 23, 42)

H. Kim, G. Papamakarios, and A. Mnih. The lipschitz constant of self-attention. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5562–5571. PMLR, 2021. (p. 86)

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015. (p. 57, 74, 140, 174, 227)

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017. (p. 214)

F. J. Király and H. Oberhauser. Kernels for sequentially ordered data. *Journal of Machine Learning Research*, 20:1–45, 2019. (p. 142, 146)

Klaus Greff, Aaron Klein, Martin Chovanec, Frank Hutter, and Jürgen Schmidhuber. The Sacred Infrastructure for Computational Research. In Katy Huff, David Lippa, Dillon Niederhut, and M. Pacer, editors, *Proceedings of the 16th Python in Science Conference*, pages 49 – 56, 2017. (p. 174)

P. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, Berlin, 1992. (p. 52, 66)

C.-Y. Ko, Z. Lyu, L. Weng, L. Daniel, N. Wong, and D. Lin. POPQORN: Quantifying robustness of recurrent neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3468–3477. PMLR, 2019. (p. 151)

Y. Ko, D. Lee, and S.-W. Kim. Not all layers are equal: A layer-wise adaptive approach toward large-scale DNN training. In *Proceedings of the ACM Web Conference 2022*, page 1851–1859, 2022. (p. 181)

J. F. Kolen and S. C. Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*, pages 237–243. John Wiley & Sons, 2001. (p. 20)

A. Kolmogorov and V. Tikhomirov. ε-entropy and ε-capacity of sets in functional spaces. *Uspekhi Mat. Nauk*, 14:3–86, 1959. (p. 126)

A. Kontorovich. Concentration in unbounded metric spaces and algorithmic stability. In E. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 28–36. PMLR, 2014. (p. 71, 72)

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. (p. 57, 87)

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28:1302–1338, 2000. (p. 116)

Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller. *Efficient BackProp*, pages 9–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. (p. 78)

J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313. Association for Computational Linguistics, 2022. (p. 234)

D. Levin, T. Lyons, and H. Ni. Learning from the past, predicting the statistics for the future, learning an evolving system. *arXiv:1309.0260v6*, 2013. (p. 27, 35, 142, 145, 146)

F.-F. Li, J. Wu, and R. Gao. Deep learning for computer vision course, 2022. URL: `https://cs231n.github.io/transfer-learning/` (version: 2023-02-02). (p. 181)

Q. Li, L. Chen, C. Tai, and E. Weinan. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18:5998–6026, 2017. (p. 23)

S. Li, L. Wu, S. Feng, F. Xu, F. Xu, and S. Zhong. Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2841–2852. Association for Computational Linguistics, 2020a. (p. 214)

X. Li, T.-K. L. Wong, R. Chen, and D. Duvenaud. Scalable gradients and variational inference for stochastic differential equations. In C. Zhang, F. Ruiz, T. Bui, A. Dieng, and D. Liang, editors, *Proceedings of the 2nd Symposium on Advances in Approximate Bayesian Inference*, volume 118, pages 1–28. PMLR, 2020b. (p. 52)

Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021a. (p. 123)

Z. Li, Y. Luo, and K. Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2021b. (p. 80)

C. Liang, J. Berant, Q. Le, K. D. Forbus, and N. Lao. Neural symbolic machines: Learning semantic parsers on Freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 23–33. Association for Computational Linguistics, 2017. (p. 213)

S. Liao, T. Lyons, W. Yang, and H. Ni. Learning stochastic differential equations using RNN with log signature features. *arXiv:1908.08286*, 2019. (p. 142)

S. H. Lim. Understanding recurrent neural networks using nonequilibrium response theory. *Journal of Machine Learning Research*, 22:1–48, 2021. (p. 142)

S. H. Lim, N. B. Erichson, L. Hodgkinson, and M. W. Mahoney. Noisy recurrent neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5124–5137. Curran Associates, Inc., 2021. (p. 124)

C. Liu, L. Zhu, and M. Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022. Special Issue on Harmonic Analysis and Machine Learning. (p. 26, 34, 79, 80)

S. Loyka. On singular value inequalities for the sum of two matrices. *arXiv:1507.06630*, 2015. (p. 115)

J. Lu, K. Deng, X. Zhang, G. Liu, and Y. Guan. Neural-ODE for pharmacokinetics modeling and its advantage to alternative machine learning models in predicting new dosing regimens. *iScience*, 24:102804, 2021. (p. 122)

Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3276–3285. PMLR, 10–15 Jul 2018. (p. 122)

Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv:1906.02762*, 2019. (p. 86)

J. Luk. Notes on existence and uniqueness theorems for ODEs, 2017. URL: `http://web.stanford.edu/~jluk/math63CMspring17/Existence.170408.pdf` (version: 2023-01-19). (p. 114, 132, 184)

T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics, 2015. (p. 20)

T. Lyons. Rough paths, signatures and the modelling of functions on streams. *arXiv:1405.4537*, 2014. (p. 161)

T. J. Lyons, M. J. Caruana, and T. Lévy. *Differential Equations Driven by Rough Paths*, volume 1908 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007. (p. 28, 142, 145, 154, 158)

K. Lyu and J. Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020. (p. 80)

L. E. MacDonald, H. Saratchandran, J. Valmadre, and S. Lucey. A global analysis of global optimisation. *arXiv:2210.05371*, 2022. (p. 80)

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. (p. 152)

P. Marion. Generalization bounds for neural ordinary differential equations and deep residual networks. In A. Oh, T. Naumann, A. Globerson, M. Hardt, S. Levine, and K. Saenko, editors, *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. (p. 9)

P. Marion and R. Berthier. Leveraging the two-timescale regime to demonstrate convergence of neural networks. In A. Oh, T. Naumann, A. Globerson, M. Hardt, S. Levine, and K. Saenko, editors, *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. (p. 9)

P. Marion, P. Nowak, and F. Piccinno. Structured context and high-coverage grammar for conversational question answering over knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8813–8829. Association for Computational Linguistics, 2021. (p. 9)

P. Marion, A. Fermanian, G. Biau, and J.-P. Vert. Scaling ResNets in the large-depth regime. *arXiv:2206.06929*, 2022. (p. 9)

P. Marion, Y.-H. Wu, M. E. Sander, and B. Gérard. Implicit regularization of deep residual networks towards neural odes. *arXiv:2309.01213*, 2023. (p. 9)

S. Massaroli, M. Poli, J. Park, A. Yamashita, and H. Asama. Dissecting neural ODEs. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3952–3963. Curran Associates, Inc., 2020. (p. 22, 78, 122, 125)

S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115:7665–7671, 2018. (p. 18, 178, 181)

T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, volume 2, pages 1045–1048, 2010. (p. 142)

A. A. Minai and R. D. Williams. On the derivatives of the sigmoid. *Neural Networks*, 6:845–853, 1993. (p. 158)

J. Morrill, C. Salvi, P. Kidger, and J. Foster. Neural rough differential equations for long time series. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7829–7838. PMLR, 18–24 Jul 2021. (p. 142)

J. H. Morrill, A. Kormilitzin, A. J. Nevado-Holgado, S. Swaminathan, S. D. Howison, and T. J. Lyons. Utilization of the signature method to identify the early onset of sepsis from multivariate physiological time series in critical care monitoring. *Critical Care Medicine*, 48:976–981, 2020. (p. 142)

V. Nagarajan and J. Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning. 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, pages 11615–11626. Curran Associates, Inc., 2019. (p. 18)

B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In P. Grünwald, E. Hazan, and S. Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1376–1401. PMLR, 2015a. (p. 131)

B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv:1412.6614*, 2015b. (p. 79, 80)

B. Neyshabur, S. Bhojanapalli, and N. Srebro. A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018. (p. 124)

Q. N. Nguyen and M. Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11961–11972. Curran Associates, Inc., 2020. (p. 80, 108, 113, 117)

C. Pabbaraju, E. Winston, and J. Z. Kolter. Estimating lipschitz constants of monotone deep equilibrium models. In *International Conference on Learning Representations*, 2021. (p. 124)

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 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, pages 8026–8037. Curran Associates, Inc., 2019. (p. 118, 140, 173, 174)

E. Pauwels. *Statistics, Optimization and Algorithms in High Dimension*. Lecture Notes, Toulouse 3 Paul Sabatier University, 2020. (p. 70, 72)

S. Peluchetti and S. Favaro. Infinitely deep neural networks as diffusion processes. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1126–1136. PMLR, 2020. (p. 52)

I. Perez Arribas. Derivatives pricing using signature payoffs. *arXiv:1809.09466*, 2018. (p. 142)

S. Pesme, L. Pillaud-Vivien, and N. Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29218–29230. Curran Associates, Inc., 2021. (p. 233)

M. Phuong and M. Hutter. Formal algorithms for transformers. *arXiv:2207.09238*, 2022. (p. 21, 219)

J. Plepi, E. Kacupaj, K. Singh, H. Thakkar, and J. Lehmann. Context transformer with stacked pointer networks for conversational question answering over knowledge graphs. In R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, P. Ristoski, and M. Alam, editors, *The Semantic Web*, pages 356–371. Springer International Publishing, 2021. (p. 213, 221, 229)

Z. Qian, W. Zame, L. Fleuren, P. Elbers, and M. van der Schaar. Integrating expert ODEs into neural ODEs: Pharmacology and disease progression. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11364–11383. Curran Associates, Inc., 2021. (p. 122)

A. F. Queiruga, N. B. Erichson, D. Taylor, and M. W. Mahoney. Continuous-in-depth neural networks. *arXiv:2008.02389*, 2020. (p. 122)

A. F. Queiruga, N. B. Erichson, L. Hodgkinson, and M. W. Mahoney. Stateful ODE-nets using basis function expansions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21770–21781. Curran Associates, Inc., 2021. (p. 79, 122)

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. (p. 123)

N. Razin and N. Cohen. Implicit regularization in deep learning may not be explainable by norms. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21174–21187. Curran Associates, Inc., 2020. (p. 18)

J. F. Reizenstein and B. Graham. Algorithm 1004: The iisignature library: Efficient calculation of iterated-integral signatures and log signatures. *ACM Transactions on Mathematical Software*, 46:8, 2020. (p. 146)

P. Rigollet and J.-C. Hütter. High dimensional statistics, 2017. Lecture notes. URL: `https://math.mit.edu/~rigollet/PDFs/RigNotes17.pdf` (version: 2019-11-05). (p. 136)

J. Riordan. *An Introduction to Combinatorial Analysis*. John Wiley & Sons, New York, 1958. (p. 158)

Y. Ro and J. Y. Choi. AutoLR: Layer-wise pruning and auto-tuning of learning rates in fine-tuning of deep networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35: 2486–2494, 2021. (p. 181)

G. Rotskoff and E. Vanden-Eijnden. Trainability and accuracy of neural networks: An interacting particle system approach. *arXiv:1805.00915*, 2018. (p. 18, 178, 181)

Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. 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, pages 5320–5330. Curran Associates, Inc., 2019. (p. 23, 142)

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. (p. 20)

D. Saad and S. Solla. On-line learning in soft committee machines. *Physical Review E*, 52: 4225–4243, 1995. (p. 181)

I. Safran, G. Vardi, and J. D. Lee. On the effective number of linear regions in shallow univariate relu networks: Convergence guarantees and implicit bias. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 32667–32679. Curran Associates, Inc., 2022. (p. 181)

A. Saha, V. Pahuja, M. M. Khapra, K. Sankaranarayanan, and S. Chandar. Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. (p. 30, 212, 213, 220, 221, 224)

A. Saha, G. A. Ansari, A. Laddha, K. Sankaranarayanan, and S. Chakrabarti. Complex program induction for querying knowledge bases in the absence of gold programs. *Transactions of the Association for Computational Linguistics*, 7:185–200, 2019. (p. 213)

C. Salvi, T. Cass, J. Foster, T. Lyons, and W. Yang. The signature kernel is the solution of a goursat pde. *SIAM Journal on Mathematics of Data Science*, 3:873–899, 2021. (p. 146)

M. E. Sander, P. Ablin, M. Blondel, and G. Peyré. Sinkformers: Transformers with doubly stochastic attention. In *Proceedings of the Twenty Fifth International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 3515–3530. PMLR, 2022a. (p. 86)

M. E. Sander, P. Ablin, and G. Peyré. Do residual neural networks discretize neural ordinary differential equations? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 36520–36532. Curran Associates, Inc., 2022b. (p. 23, 78, 79, 80, 122, 128, 129, 140)

M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web, Proceedings of the 15th International Conference, ESWC 2018*, pages 593–607. Springer, 2018. (p. 214)

B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, Cambridge, Massachusetts, 2002. (p. 146)

J. Shao, K. Hu, C. Wang, X. Xue, and B. Raj. Is normalization indispensable for training deep neural network? In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13434–13444. Curran Associates, Inc., 2020. (p. 40, 42)

P. Shaw, P. Massey, A. Chen, F. Piccinno, and Y. Altun. Generating logical forms from graph representations of text and entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 95–106. Association for Computational Linguistics, 2019. (p. 214)

T. Shen, X. Geng, T. Qin, D. Guo, D. Tang, N. Duan, G. Long, and D. Jiang. Multi-task learning for conversational question answering over a large-scale knowledge base. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2442–2451. Association for Computational Linguistics, 2019. (p. 212, 213, 216, 221, 223, 226, 228)

T. Shen, X. Geng, G. Long, J. Jiang, C. Zhang, and D. Jiang. Effective search of logical forms for weakly supervised knowledge-based question answering. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 2227–2233. International Joint Conferences on Artificial Intelligence Organization, 2020. (p. 212, 213, 216, 221)

V. Shiv and C. Quirk. Novel positional encodings to enable tree-based transformers. 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, pages 12081–12091. Curran Associates, Inc., 2019. (p. 214)

B. Singh, S. De, Y. Zhang, T. Goldstein, and G. Taylor. Layer-specific adaptive learning rates for deep networks. In *2015 IEEE 14th International Conference on Machine Learning and Applications*, pages 364–368, 2015. (p. 181)

J. Sirignano and K. Spiliopoulos. Mean field analysis of neural networks: a central limit theorem. *Stochastic Processes and their Applications*, 130:1820–1852, 2020. (p. 18, 178, 181)

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. (p. 23, 234)

C. Szepesvári. *Algorithms for reinforcement learning*. Springer, 2010. (p. 180)

K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1556–1566. Association for Computational Linguistics, 2015. (p. 214)

T. Teshima, K. Tojo, M. Ikeda, I. Ishikawa, and K. Oono. Universal approximation property of neural ordinary differential equations. *arXiv:2012.02414*, 2020. (p. 78)

M. Thorpe and Y. van Gennip. Deep limits of residual neural networks. *Research in the Mathematical Sciences*, 10:6, 2022. (p. 41, 52, 80, 122)

P. Tong, Q. Zhang, and J. Yao. Leveraging Domain Context for Question Answering Over Knowledge Graph. *Data Science and Engineering*, 4:323–335, 2019. (p. 213, 214)

C. Toth and H. Oberhauser. Bayesian learning from sequential data using Gaussian processes with signature covariances. In H. Daumé III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9548–9560. PMLR, 2020. (p. 142)

J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12:389–434, 2012. (p. 117)

Z. Tu, F. He, and D. Tao. Understanding generalization in recurrent neural networks. In *International Conference on Learning Representations*, 2019. (p. 143, 150)

B. Tzen and M. Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv:1905.09883*, 2019. (p. 122)

S. Vakulenko, J. Fernández, A. Polleres, M. de Rijke, and M. Cochez. Message passing for complex question answering over knowledge graphs. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1431–1440. ACM, 2019. (p. 213)

R. van Handel. *Probability in High Dimension.* APC 550 Lecture Notes, Princeton University, 2016. (p. 44)

G. Vardi. On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66: 86–93, 2023. (p. 19, 79)

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017. (p. 20, 40, 78, 216, 218, 219)

R. Veiga, L. Stephan, B. Loureiro, F. Krzakala, and L. Zdeborová. Phase diagram of stochastic gradient descent in high-dimensional two-layer neural networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022. (p. 181)

R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018. (p. 108)

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman,

N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. (p. 174)

M. J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019. (p. 135)

B. Wang, M. Liakata, H. Ni, T. Lyons, A. J. Nevado-Holgado, and K. Saunders. A path signature approach for speech emotion recognition. In *Proceedings of Interspeech 2019*, pages 1661–1665, 2019. (p. 142)

H. Wang, S. Ma, L. Dong, S. Huang, D. Zhang, and F. Wei. Deepnet: Scaling transformers to 1,000 layers. *arXiv:2203.00555*, 2022. (p. 22, 40, 44, 78)

Y.-J. Wang and C.-T. Lin. Runge-Kutta neural network for identification of dynamical systems in high accuracy. *IEEE Transactions on Neural Networks*, 9:294–307, 1998. (p. 153)

B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. In J. Abernethy and S. Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3635–3673. PMLR, 2020. (p. 19)

W. Woof and K. Chen. A Framework for End-to-End Learning on Semantic Tree-Structured Data. *arXiv:2002.05707*, 2020. (p. 214)

L. Wu, Q. Wang, and C. Ma. Global convergence of gradient descent for deep linear residual networks. 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, pages 13389–13398. Curran Associates, Inc., 2019. (p. 80)

K. Xu, L. Wu, Z. Wang, M. Yu, L. Chen, and V. Sheinin. Exploiting rich syntactic information for semantic parsing with graph-to-sequence model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 918–924. Association for Computational Linguistics, 2018. (p. 214)

G. Yang and S. Schoenholz. Mean field residual networks: On the edge of chaos. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 2865–2873. Curran Associates, Inc., 2017. (p. 40, 41, 44, 49, 50, 57)

W. Yang, L. Jin, and M. Liu. DeepWriterID: An end-to-end online text-independent writer identification system. *IEEE Intelligent Systems*, 31:45–53, 2016. (p. 142)

W. Yang, T. Lyons, H. Ni, C. Schmid, and L. Jin. Developing the path signature methodology and its application to landmark- based human action recognition. In *Stochastic Analysis, Filtering, and Stochastic Optimization: A Commemorative Volume to Honor Mark H. A. Davis's Contributions*, pages 431–464. Springer International Publishing, 2022. (p. 142)

Y. Yin, I. Ayed, E. de Bézenac, N. Baskiotis, and P. Gallinari. LEADS: Learning dynamical systems that generalize across environments. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 7561–7573. Curran Associates, Inc., 2021. (p. 124)

Y. You, I. Gitman, and B. Ginsburg. Scaling SGD batch size to 32k for imagenet training. *arXiv:1708.03888*, 2017. (p. 181)

D. Yu, H. Miao, and H. Wu. Neural generalized ordinary differential equations with layer-varying parameters. *arXiv:2209.10633*, 2022. (p. 123)

B. Yue, J. Fu, and J. Liang. Residual recurrent neural networks for learning sequential representations. *Information*, 9:56, 2018. (p. 20, 143)

H. Zafar, G. Napolitano, and J. Lehmann. Deep Query Ranking for Question Answering over Knowledge Bases. In U. Brefeld, E. Curry, E. Daly, B. MacNamee, A. Marascu, F. Pinelli, M. Berlingerio, and N. Hurley, editors, *Machine Learning and Knowledge Discovery in Databases, Proceedings of the European Conference*, Lecture Notes in Computer Science, pages 635–638. Springer International Publishing, 2019. (p. 214)

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64:107–115, 2021. (p. 17)

H. Zhang, Y. Dauphin, and T. Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019a. (p. 42, 81)

H. Zhang, D. Yu, M. Yi, W. Chen, and T.-Y. Liu. Convergence theory of learning over-parameterized ResNet: A full characterization. *arXiv:1903.07120*, 2019b. (p. 40, 42, 43)

H. Zhang, X. Gao, J. Unterman, and T. Arodz. Approximation capabilities of neural ODEs and invertible residual networks. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11086–11095. PMLR, 2020a. (p. 78)

J. Zhang, Q. Lei, and I. Dhillon. Stabilizing gradients for deep neural networks via efficient SVD parameterization. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5806–5814. PMLR, 2018. (p. 143, 150)

J. Zhang, B. Han, L. Wynter, B. Low, and M. Kankanhalli. Towards robust ResNet: A small step but a giant leap. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 4285–4291. International Joint Conferences on Artificial Intelligence Organization, 2019c. (p. 43)

J. Zhang, H. Zhang, C. Xia, and L. Sun. Graph-Bert: Only Attention is Needed for Learning Graph Representations. *arXiv:2001.05140*, 2020b. (p. 213)

F. Zhou, L. Li, K. Zhang, and G. Trajcevski. Urban flow prediction with spatial–temporal neural ODEs. *Transportation Research Part C: Emerging Technologies*, 124:102912, 2021. (p. 122)

D. Zou, Y. Cao, D. Zhou, and Q. Gu. Gradient descent optimizes over-parameterized deep ReLU networks. *Machine Learning*, 109:467–492, 2020a. (p. 18, 178, 181)

D. Zou, P. M. Long, and Q. Gu. On the global convergence of training deep linear resnets. In *International Conference on Learning Representations*, 2020b. (p. 80)