On identifiability of deep ReLU neural networks

Joachim Bona-Pellissier

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 1 Capitole (UT1 Capitole)*

Présentée et soutenue le *18 décembre 2023* **par :**

Joachim Bona-Pellissier

L'identifiabilité des réseaux de neurones profonds ReLU

JURY

J. Elisenda GRIGSBY	Boston College	Rapporteure
Rémi GRIBONVAL	INRIA - ENS Lyon	Rapporteur
Francis BACH	INRIA - ENS Paris	Examinateur
Gérard BIAU	Sorbonne Université	Examinateur
Edouard PAUWELS	Université Toulouse 1	Examinateur
Lorenzo ROSASCO	Università di Genova - MIT	Examinateur
François MALGOUYRES	Université Toulouse 3	Directeur de thèse
François BACHOC	Université Toulouse 3	Directeur de thèse

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées
Unité de Recherche :
Institut de Mathématiques de Toulouse (UMR 5219)
Directeur(s) de Thèse :
François MALGOUYRES et *François BACHOC*
Rapporteurs :
J. Elisenda GRIGSBY et *Rémi GRIBONVAL*

Remerciements

Je tiens tout d'abord à remercier les rapporteur.es, Elisenda Grigsby et Rémi Gribonval, d'avoir eu la gentillesse d'accepter de relire mon manuscrit. Merci pour vos retours détaillés et enrichissants, dont j'espère tirer tous les enseignements. Je voudrais aussi remercier Francis Bach, Gérard Biau, Edouard Pauwels, Lorenzo Rosasco, pour avoir accepté de faire partie de mon jury. C'est un honneur de présenter mon travail devant vous.

I would first like to thank the referees, Elisenda Grigsby and Rémi Gribonval, for kindly agreeing to review my manuscript. Thank you for your detailed and enriching feedback, from which I hope to learn all I can. I would also like to thank Francis Bach, Gérard Biau, Edouard Pauwels and Lorenzo Rosasco for agreeing to serve on my jury. It is an honor to present my work to you.

Je tiens à remercier ceux qui m'ont initié à la recherche, et aux côtés de qui j'ai tant appris, mes directeurs François Malgouyres et François Bachoc. Merci de m'avoir fait confiance pour cette thèse, merci de vos conseils avisés, de votre patience et de votre présence pendant ces quatre années. C'était un plaisir de travailler avec vous, j'espère que l'occasion se reproduira.

Parmi les choses que je retiendrai de mes années à l'IMT, je dois citer la bonne ambiance qui règne parmi les doctorant.es (sans oublier les post-doc et ATER). L'entraide et la bienveillance que j'ai pu trouver en leur sein, ainsi que les moments de convivialité, de discussions et de jeux, ont contribué à rendre ce lieu agréable à vivre. La thèse n'étant pas une aventure facile, ce sont je crois des facteurs essentiels de succès. Je tiens, sans exhaustivité, à les remercier ici. Au plus bel accent du labo et au meilleur dessinateur de pingouins, et à la handballeuse badass, accessoirement ma voleuse de café (mais qui prévient) préférée. Nicolas, Sophia, vos rires me manquent déjà. Alexandre (bonjour), j'ai découvert un nouveau centre d'intérêt grâce à ta passion communicative pour les jeux, et mon portefeuille ne te remercie pas. Anthony, tu es un super pote et tes blagues sont toujours réussies (une de ces infos est fausse). Commence à te préparer pour la Corrida 2024. Etienne, ça a été un plaisir de te côtoyer à l'intérieur comme à l'extérieur de l'IMT. Mickey va me manquer. A mes frères de thèse Mehdi et Armand, le voyage à la Nouvelle Orléans était mémorable. Je voudrais aussi remercier Maxime (l'ekiden de l'an prochain nous attend), Chifaa (merci pour toutes les pauses à discuter de nos avenirs, de tout et de rien), Perla (la reine indétrônable du tarot, pourtant on a essayé), Fu-Hsuan (et les cours de danse du mercredi soir), Paola et Alberto (j'espère que vous allez vous plaire dans la Grosse Pomme), Alain (et les nocturnes de l'IMT), Javi (well), Mitja, Fanny, Alejandro, Candice, Adama, Benjamin, Louis B, Louis G, Viviana, Virgile, Corentin, Clément B, Lucas C, Louis D, Florian. À mes cobureaux, Clément, Lucas et Mahmoud. Louis C, Mathis, je compte sur vous pour la relève. Ca a été un plaisir de vous connaître pour ce laps de temps, bonne chance pour votre thèse. Joachim et Michèle, vous avez été les premières personnes à m'accueillir au labo, et j'en garde un souvenir ému. Laetitia je suis certain qu'on se recroisera, on ne peut rien contre le destin.

À toutes celles et ceux que j'ai pu oublier, et de manière générale, à toutes les personnes, étudiant.es, doctorant.es, post-doctorant.es, ATER, ingénieur.es, chercheur.ses, membres du personnel, avec qui j'ai eu l'occasion d'échanger au sein de l'IMT, de l'IRT Saint-Exupéry, et du campus de l'Université Paul Sabatier.

J'ai aussi eu l'occasion de faire quelques belles rencontres hors du laboratoire au cours de cette aventure toulousaine, parmi lesquelles je mentionnerai Camille D, Margaux, Jessica, Camille B, Lola, Pauline et Erwanne.

À la meilleure personne du monde, Alicia. La vie sans toi, c'est comme un feu qui ne crépite pas. Merci pour tous ces moments passés ensemble.

Quittant Toulouse, j'ai une pensée pour l'équipe de MaLGa à Gênes et pour son accueil chaleureux, et en particulier pour Hippolyte, Romain, Vassilis, Nicolas et Antoine pour les petits conseils pour la dernière ligne droite.

Si on revient un peu en arrière, Dylan, Amnay et Antoine, c'est à vos côtés que j'ai réellement découvert le goût des mathématiques, qui ne m'a plus quitté depuis. Luc, ta jovialité et ta passion pour la transmission sont uniques. À Nicolas et Adrien, et à nos belles balades mathématiques et pédestres. Charles, j'ai du mal à croire que l'on touche le but à si peu d'intervalle. À nos discussions à des heures invraisemblables de la nuit. Re-elect Frank Sobotka.

Je suis obligé de dédier cette thèse à l'équipe, même s'ils le savent déjà : Nils, Guillaume, Diwan, Caro, Anmol. 2000 jusqu'à l'infini !

Je voudrais finir par ma famille, qui est celle qui compte le plus pour moi et à qui les mots ne sauraient rendre justice. À ma soeur, coach sportive attitrée et compagnonne d'épreuve. Je nous revois travailler côte à côte sur nos thèses respectives. Félicitations pour ton doctorat, tu l'as tellement mérité. À mes parents à qui je dois une curiosité dont j'espère ne jamais me séparer.

Contents

Chapter 1

Introduction

1.1 Introduction to deep learning

In the era of machine learning, the field of deep learning has emerged as a cornerstone of technological advancement, revolutionizing our ability to process, understand, and extract valuable insights from vast and complex datasets. Deep neural networks, with their multilayered architectures, have played a pivotal role in this transformation, demonstrating unprecedented capabilities in various applications, ranging from image and speech recognition to natural language understanding. As we continue to witness the proliferation of these powerful learning machines, crucial questions regarding their inner workings and behaviors come to the forefront. Amidst this technological revolution, this Ph.D. thesis embarks on a journey into the world of deep learning theory, specifically focusing on the identifiability of neural network parameters—an aspect that has garnered limited attention despite its potential significance. The motivation for this study lies in addressing critical challenges related to privacy preservation, robustness, and generalization capabilities. In an age where data privacy is paramount, understanding the extent to which neural network parameters can be inferred from observed data is vital for safeguarding sensitive information. Additionally, enhancing the robustness of these models to adversarial attacks and improving their generalization performance in diverse real-world scenarios hinges on a deep comprehension of parameter identifiability. Recognizing the ethical and practical importance of safeguarding data privacy, fortifying model resilience, and improving generalization performance, this research seeks to contribute modestly to our understanding of these issues.

This introduction is structured as follows: in Section 1.1, we recall a few basics of machine learning, deep learning, and ReLU geometry, in order to set a few notations and to give a context to the thesis. In Section 1.2, we expand on the symmetries possessed by the parameters of neural networks and the relation between an activation function and the corresponding symmetries. In Section 1.3, we give an introduction of the notion of identifiability and we formalize what it means for ReLU neural networks as well as its motivations. Finally, in Section 1.4, we give an overview of the question of complexity for neural networks and its link with generalization abilities.

1.1.1 Learning problem, classification, regression

In this section and the following, we recall succinctly a few basics of machine learning. The goal here is not to be exhaustive but merely to give a frame to our

study of neural networks.

In the standard supervised learning problem, we consider two random variables: an input variable X and a target Y. The goal is, given the knowledge of X, to predict the behavior of Y.

We will consider two classical learning settings: regression and classification. In the regression setting, we observe $X \in \mathbb{R}^d$, for some $d \in \mathbb{N}^*$, and we try to predict a real value or a vector of real values: we have $Y \in \mathbb{R}^m$, for some $m \in \mathbb{N}^*$. The relationship between X and Y is not necessarily deterministic. In that case, it is impossible to predict exactly Y given X. Rather than predicting the exact value of Y, the aim is typically to predict $\mathbb{E}[Y|X]$.

In classification, we consider a finite set of K classes $C_1, \dots, C_K$, for $K \in \mathbb{N}^*$. We observe $X \in \mathbb{R}^d$ and we want to predict the class of X, represented by the variable $Y \in \{C_1, \dots, C_K\}$. Again, the relationship between X and Y is not necessarily deterministic, and we will typically want to return an estimation of the probability $\mathbb{P}(Y \in C_i \mid X)$, or at least a score for each class C_i, such that for all $i \in [\![1, K]\!]$ the higher the score we assign to the class C_i, the higher the probability $\mathbb{P}(Y \in C_i \mid X)$ is.

In both cases, we want to find a function that takes $X \in \mathbb{R}^d$ as an input and outputs a real vector (such as the predicted $\mathbb{E}[Y|X] \in \mathbb{R}^m$ in regression or the predicted probabilities vector $\mathbb{P}(Y \in C_i \mid X)_{i \in [\![1,K]\!]} \in [0,1]^K$ or score vector $\lambda \in \mathbb{R}^K$ in classification). To do so, we are provided with a parameterized family of functions $(f_\theta)_{\theta \in \mathcal{P}}$, where $\mathcal{P}$ denotes some parameter set. Most of the time for us, $\mathcal{P}$ will correspond to a space $\mathbb{R}^p$, for some $p \in \mathbb{N}^p$. The goal is to find the parameter θ such that the function f_θ that best fits the expected behavior.

1.1.2 True risk, empirical risk

To measure how well a given function f_θ performs in our task, we consider a *loss function* (also called cost function or error function) $\ell : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+$. The loss function allows to compare our prediction $f_\theta(X)$ with the true Y: the higher the value of the loss is, the further we consider $f_\theta(X)$ to be from Y. Generally, the only way to bring $\ell(f_\theta(X), Y)$ down to zero is to predict the exact value $f_\theta(X) = Y$. A typical example of loss in regression is the quadratic loss: $\ell(y, y') = \sum_{i=1}^m (y_i - y'_i)^2$.

Our objective will be to choose the function f_θ in order to minimize the loss on average over the distribution of (X, Y). We thus define the risk, also referred to as population risk, as

$$R(\theta) = \mathbb{E}\left[\ell(f_\theta(X), Y)\right].$$

Although our goal is to minimize $R(\theta)$, we typically do not have access to the true distribution of (X, Y), which makes it impossible to compute $R(\theta)$ for a given parameter θ. Instead, if we are given n examples $x^{(1)}, \dots, x^{(n)}$ that were sampled from (X, Y), for a given integer n, we can consider the empirical risk

$$\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_\theta(x^{(i)}), y^{(i)}\right),$$

which is the average loss over the n examples rather than the whole distribution, and which we are able to compute. Finding conditions guaranteeing that the empirical risk $\hat{R}(\theta)$ is close to the true risk $R(\theta)$ is a vast subject in Machine Learning. We will not expand on it here, but more discussion on the subject can be found in [190, 21]. See also Section 1.4 for a discussion on the subject in the case of neural networks.

1.1.3 Neural networks, a first introduction

A neural network is a family of functions $(f_\theta)_{\theta\in\mathbb{R}^p}$, such that each function $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$ is built as a succession of layers. Indeed, we can write the function f_θ as a succession of compositions between $L \geq 2$ more elementary functions:

$$f_\theta = h_L \circ h_{L-1} \circ \cdots \circ h_1,$$

where for all $l \in [\![1, L]\!]$, the function h_l corresponds to one layer of the network, the layer l. The function h_l implements a map between two vector spaces $\mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$. Here $\mathbb{R}^{N_0}$ and $\mathbb{R}^{N_L}$ are the input and output space of f_θ respectively; we thus have $N_0 = d$ and $N_L = m$.

Typically, and as will be the case in this work, a layer is composed of an affine map $x \mapsto W^l x + b^l$, with $W^l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $b^l \in \mathbb{R}^{N_l}$, followed by an activation function $\sigma_l : \mathbb{R}^l \to \mathbb{R}^l$. Mathematically, it thus writes

$$\forall x \in \mathbb{R}^{N_{l-1}}, \quad h_l(x) = \sigma_l\left(W^l x + b^l\right).$$

The output layer is an exception, as h_L is only composed of a linear map, i.e.

$$\forall x \in \mathbb{R}^{N_{L-1}}, \quad h_L(x) = W^L x + b^L.$$

By abuse of language, sometimes the 'layer l' denotes the space $\mathbb{R}^{N_l}$, and sometimes it denotes the map $h_l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$ and its weights and bias (W^l, b^l).

The function implemented by the neural networks depends on its weights and biases. Together, all the weights and biases form the parameterization of the network

$$\theta = (W^1, \ldots, W^L, b^1, \ldots, b^L) \in \mathbb{R}^p,$$

where $p = N_0N_1 + \ldots N_{L-1}N_L + N_1 + \cdots + N_L$. We sometimes refer to the parameterization θ simply as 'parameter' of the network, as is often done in the literature, although it is in fact a vector of p parameters. It should always be clear from the context whether the word 'parameter' refers to θ or to one component of θ. In particular, the expression 'number of parameters' of the network, often used in this thesis and in the literature, refers to the number p.

In many settings, and in all the settings considered in this thesis, for any $l \in [\![1, L-1]\!]$, the activation σ_l corresponds to the same real-valued activation σ applied component-wise, that is

$$\forall (x_1, \ldots, x_{N_l}) \in \mathbb{R}^{N_l}, \quad \sigma_l(x_1, \ldots, x_{N_l}) = (\sigma(x_1), \ldots, \sigma(x_{N_l})).$$

There exist several classical activation functions σ: sigmoid, hyperbolic tangent (tanh), Rectified Linear Unit (ReLU)... In this thesis, we focus on ReLU: it is defined as $\sigma(t) = \max(t, 0)$ for any $t \in \mathbb{R}$. By default, σ will refer to ReLU. It will be explicitly stated when it is not the case.

1.1.4 Neural networks as graphs

In this section, we present an equivalent formalism to neural networks, which gives them their name. It is a presentation of neural networks as graphs between nodes that are called *neurons*.

We first consider a set of neurons V. This set of neurons is divided into $L+1$ *layers*, with $L \geq 2$: $V = \bigcup_{l=0}^{L} V_l$. The layer V_0 is the *input layer*, V_L is the *output layer* and the layers V_l with $1 \leq l \leq L-1$ are the *hidden layers*. We denote, for all $l \in [\![0, L]\!]$, $N_l = |V_l|$ the size of the layer V_l.

The neurons in consecutive layers are connected by oriented edges: for any $l \in [\![0, L-1]\!]$, if we consider two neurons $v \in V_l$ and $v' \in V_{l+1}$, then we denote by $v \to v'$ the oriented edge from v to v'. We denote by E the set of all such oriented edges:

$$E = \{v \to v' \mid \ v \in V_l, v' \in V_{l+1}, \text{for } l \in [\![0, L-1]\!]\}.$$

Every edge $v \to v' \in E$ of the network is parameterized by a *weight* $w_{v \to v'} \in \mathbb{R}$. Furthermore, we denote by

$$B = \bigcup_{l=1}^{L} V_l$$

the set of all neurons except the input neurons. Each neuron $v \in B$ is parameterized by a *bias* $b_v \in \mathbb{R}$. A network is parameterized by all its weights and biases, gathered in the parameterization θ, with

$$\theta = ((w_{v \to v'})_{v \to v' \in E}, (b_v)_{v \in B}) \quad \in \mathbb{R}^E \times \mathbb{R}^B \simeq \mathbb{R}^p,$$

where $p = |E| + |B| = N_0 N_1 + \ldots N_{L-1} N_L + N_1 + \cdots + N_L$ is the number of parameters.

As mentioned in previous section, the activation function, denoted σ, is most of the time ReLU in our case; otherwise, it will always be explicitly specified. It is defined as $\sigma(t) = \max(t, 0)$ for any $t \in \mathbb{R}$.

ReLU network prediction For a given θ, we define recursively $f_\theta^l : \mathbb{R}^{V_0} \to \mathbb{R}^{V_l}$, for $l \in [\![0, L]\!]$ and $x \in \mathbb{R}^{V_0}$, by

$$\begin{cases} (f_\theta^0(x))_v = x_v, & \text{for } v \in V_0, \\ (f_\theta^l(x))_v = \sigma\left(\sum_{v' \in V_{l-1}} w_{v' \to v} (f_\theta^{l-1}(x))_{v'} + b_v\right), & \text{for } v \in V_l, \text{ when } l \in [\![1, L-1]\!], \\ (f_\theta^L(x))_v = \sum_{v' \in V_{L-1}} w_{v' \to v} (f_\theta^{L-1}(x))_{v'} + b_v, & \text{for } v \in V_L. \end{cases} \tag{1.1.1}$$

We define the function $f_\theta : \mathbb{R}^{V_0} \to \mathbb{R}^{V_L}$ implemented by the network of parameter θ as $f_\theta = f_\theta^L$. We sometimes call it the prediction or the inference.

Equivalence between formalisms The two presented formalisms are equivalent: by simply numbering the neural networks in each layer, we obtain a bijection $V_l \to [\![1, N_l]\!]$, which yields an isomorphism $\mathbb{R}^{V_l} \to \mathbb{R}^{N_l}$. Using these isomorphisms, one can gather the weights between two layers $(w_{v\to v'})_{(v,v')\in V_{l-1}\times V_l}$ in a matrix W^l and the biases $(b_v)_{v\in V_l}$ of a layer in a vector b^l, which allows switching from one notation to another.

1.1.5 Optimization

In this section we provide a very succinct explanation of gradient descent and stochastic gradient descent, that are key for neural network optimization. For a more detailed overview of gradient methods, see [160]. As evoked in Section 1.1.1, the goal of learning is to find a parameter θ such that the corresponding function f_θ best fits some expected behavior. For neural networks, the parameters are real vectors $\theta \in \mathbb{R}^p$, and the common way of searching for the best parameter is through gradient-based optimization methods.

To do so, a positive differentiable function $L : \mathbb{R}^p \longrightarrow \mathbb{R}_+$ is constructed. L takes parameters of the network as inputs, and outputs positive values. The goal is to minimize it, i.e. to find a parameter θ such that $L(\theta)$ is as small as possible (the ideal being to find the global minimum of L). L is named the objective function, or sometimes the loss (but it should not be confused with the loss ℓ of section 1.1.2).

Gradient descent is a very classical family of optimization techniques that roughly works as follows: one chooses an initial parameter θ_0. Then, the parameter is incrementally modified, progressively constructing a sequence of parameters $(\theta_t)_{t\in\mathbb{N}}$. At step t, the current parameter θ_t is modified following the direction of steepest descent of L at the point θ_t, i.e. so that the step $\theta_{t+1} - \theta_t$ is proportional to the opposite of the gradient $\nabla L(\theta_t)$. Such techniques require to compute the gradient of L at each point θ_t. This can be done with an explicit expression of the gradient when possible, or with numerical methods.

As mentioned in Section 1.1.2, loss functions ℓ are used to construct a risk $R(\theta) = \mathbb{E}\left[\ell(f_\theta(X), Y)\right]$, which we would like to minimize. In practice, what we we are able to compute and what we thus consider instead is the empirical risk $\widehat{R}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell\left(f_\theta(x^{(i)}), y^{(i)}\right)$, where $(x^{(i)}, y^{(i)})_{i\in[\![1,n]\!]}$ is a learning sample obtained from (X, Y). It is thus natural to use the empirical risk as objective function: $L(\theta) = \widehat{R}(\theta)$. Training a neural network using the empirical risk as loss is standard and called *empirical risk minimization*. One can also add a regularization term $P(\theta)$ representing additional constraints that we want the final parameter to satisfy, leading to an objective function of the form $L(\theta) = \widehat{R}(\theta) + P(\theta)$.

In practice, the objective L is minimized using a variant of the classical gradient descent, called *stochastic gradient descent*. Instead of computing the gradient of the empirical risk for the full training set, we randomly subdivide the training set into k mini-batches $I_1, \dots, I_k$ of size n_b, such that $\bigcup_{j=1}^k I_k = [\![1, n]\!]$, and each gradient step is performed for a mini-batch. At a given step, we thus consider the gradient

of the following quantity :

$$\frac{1}{n_b} \sum_{i \in I_k} \ell \left(f_\theta(x^{(i)}), y^{(i)} \right).$$

Such an algorithm has been shown to be computationally more efficient (since it is easier to compute the gradient for a mini-batch than for the full training set), and to still yield good performances [29, 69].

To compute the empirical risk, whether it is for the whole training set or for a sub-batch of the training set, one needs to be able to compute the individual gradients of the quantities $\ell \left(f_\theta(x^{(i)}), y^{(i)} \right)$ with respect to θ, for each example $x^{(i)}$. Provided one chooses appropriately ℓ (it should at least be differentiable), neural networks are equipped with an efficient way of computing these gradients, which is called *backpropagation*, and on which we do not expand here.

1.1.6 The piecewise-affine geometry of ReLU

In this section, we expand a bit on the geometric properties of neural networks with ReLU activation function. The functions implemented by neural networks inherit properties from the activation function they use. ReLU is a continuous, piecewise-affine function:

$$\sigma : t \mapsto \max(0, t).$$

As a consequence, a layer of a neural network, which combines an affine function $x \mapsto Wx+b$ and the ReLU activation, implements a continuous piecewise-affine function. Since the composition of continuous piecewise-affine functions is continuous piecewise-affine, the function implemented by a ReLU network is itself continuous piecewise-affine (sometimes also called, although improperly, piecewise-linear).

The affine regions of ReLU networks As continuous piecewise-affine functions, ReLU networks divide their input space in polyhedral regions over which the function implemented by the network coincides with an affine function. These regions are called the *affine regions*, or sometimes linear regions, of the network. The affine regions are polyhedral and their boundaries are made of pieces of hyperplanes.

This property of ReLU networks allows to see them as approximators, that combine simple (affine) blocks to represent more complex functions. It is intuitive to think that the more complex the function we are trying to approximate is, the more blocks will be needed. Further, it is intuitive to think that these blocks will concentrate in the more complex areas of the function, with more curvature, more irregularities, whereas the areas where the function is less complex will need less blocks. Following this intuition, the number and the density of affine regions have been considered as measures of complexity for ReLU networks [125, 150, 86]. To finish with, the total number of affine regions is always finite. The reason for that is that there are a finite number of activation patterns (see the Activation patterns paragraph below).

Separating hyperplanes are crucial for identifiability Two adjacent affine regions are separated by a hyperplane. It appears that these separating hyperplanes are very informative on the parameters of a network. To understand that, let us consider a network made of a single hidden neuron, $L = 2$, $N_0 = d$, $N_1 = N_2 = 1$, of the form

$$f_\theta(x) = a\sigma(w^T x + b) + c,$$

where $x \in \mathbb{R}^d, a, b, c \in \mathbb{R}$. By definition of σ, we have $f_\theta(x) = a\left(w^T x + b\right) + c$ if $w^T x + b \geq 0$ and $f_\theta(x) = c$ otherwise. There are thus two affine regions, which are separated by the hyperplane of equation $w^T x + b = 0$. If $w'^T x + b'$ is another equation defining the same hyperplane, then there exists $\alpha \neq 0$ such that $(w', b') = \alpha(w, b)$. Thus, identifying the hyperplane separating the two linear regions defined by a neuron allows identifying the weights and bias of the hidden neuron, up to a scaling factor. This key property is at the core of most works on identifiability or stable recovery of the parameters of ReLU networks (see Section 1.3.5 for the related works).

Activation patterns For all $l \in [\![1, L-1]\!]$, we now define $s^l(x, \theta) \in \{0, 1\}^{N_l}$ as follows:

$$\forall i \in [\![1, N_l]\!], \quad s_i^l(x, \theta) = \begin{cases} 1 & \text{if } \sum_{j=1}^{N_{l-1}} W_{i,j}^l \left(f_\theta^{l-1}(x)\right)_j + b_i^l \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

For a fixed parameter $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, and a given input $x \in \mathbb{R}^{N_0}$, the list

$$\left(s^1(x, \theta), \ldots, s^{L-1}(x, \theta)\right) \in \{0, 1\}^{N_1} \times \cdots \times \{0, 1\}^{N_{L-1}}$$

is called an *activation pattern*.

We thus have

$$\left(f_\theta^l(x)\right)_i = \sigma\left(\sum_{j=1}^{N_{l-1}} W_{i,j}^l \left(f_\theta^{l-1}(x)\right)_j + b_i^l\right) = s_i^l(x, \theta)\left(\sum_{j=1}^{N_{l-1}} W_{i,j}^l \left(f_\theta^{l-1}(x)\right)_j + b_i^l\right),$$

or written differently

$$\begin{aligned} f_\theta^l(x) &= \mathrm{Diag}(s^l(x, \theta))\left(W^l f_\theta^{l-1}(x) + b^l\right) \\ &= \mathrm{Diag}(s^l(x, \theta)) W^l f_\theta^{l-1}(x) + \mathrm{Diag}(s^l(x, \theta)) b^l. \end{aligned} \tag{1.1.2}$$

For a given $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, suppose we consider a contiguous region $D \subset \mathbb{R}^{N_0}$ over which the activation pattern is fixed, i.e. for all $l \in [\![1, L-1]\!]$ there exists $\delta^l \in \{0, 1\}^{N_l}$ such that for all $x \in D$, $s^l(x, \theta) = \delta^l$. Then, (1.1.2) shows that for any $l \in [\![1, L-1]\!]$, the relation between $f_\theta^{l-1}(x)$ and $f_\theta^l(x)$ is affine. Since the relation between $f_\theta^{L-1}(x)$ and $f_\theta(x)$ is always affine, we can thus easily prove by induction over $l \in [\![1, L]\!]$ that f_θ is affine over D.

This shows that there is a relationship between activation patterns and linear regions. A region over which the activation pattern is fixed is included in an affine

region of f_θ. The reciprocal inclusion is not always true: a given affine region can have several activation patterns. A trivial example is when all the output weights and biases of the network are zero. In this case, the function f_θ is constantly equal to zero, so the whole input space $\mathbb{R}^d$ is a unique affine region, regardless of the different activation patterns that can exist.

ReLU networks as universal representations of piecewise-affine functions Since ReLU networks represent continuous piecewise-affine functions with a finite number of pieces, another natural question is about the reciprocal. Is any continuous piecewise-affine function with a finite number of pieces representable by a ReLU network? The answer is positive, as is shown in [7].

1.2 Neural networks possess intrinsic symmetries

Neural networks are known for generally possessing symmetries, meaning operations on the parameters that do not affect the function they implement. This is a fundamental aspect to consider before being able to discuss the identifiability of neural network parameters, as we will do in Section 1.3. In general, the question of the symmetries and the one of identifiability are closely connected.

Let us clarify immediately a terminology point: various terms are used in the literature to refer to the same phenomenon. We sometimes read about symmetries, invariants, or parameter equivalence of neural networks. Generally, in this thesis, we will say that a parameter θ is equivalent to a parameter θ'. However, when emphasizing the transformations that take a parameter θ to an equivalent parameter θ', especially when discussing the group structure of this set of transformations, as is the case in this whole section, we use the term symmetries.

In this whole section, we consider the neural network architecture and the notations defined in Sections 1.1.3 and 1.1.4, except that σ is not ReLU but a generic activation function. The contributions contained in this thesis only consider neural networks with ReLU activation function, but the more general viewpoint of this section will allows us to discuss how the symmetries of a neural network depend on the chosen activation. We hope that this will allow us to better emphasize the role of ReLU in the symmetries possessed by the neural networks considered in this thesis.

1.2.1 The permutation symmetry

A very general symmetry is the permutation symmetry. Indeed, for classical fully-connected feedforward neural networks, in a given hidden layer, no particular role is given to any neuron: they are all interchangeable. Let us consider a neural network architecture such as in Section 1.1.4, with a given (not necessarily ReLU) activation function σ, and with parameter $\theta \in \mathbb{R}^E \times \mathbb{R}^B$. Consider a hidden layer V_l, for $l \in [\![1, L-1]\!]$ and two neurons $v_1, v_2 \in V_l$. We are going to exchange the role of v_1 and v_2 in the network in order to show that they are interchangeable.

To begin with, suppose we exchange all the incoming weights of v_1 and v_2 as well as their biases: for any neuron $v \in V_{l-1}$, we define

$$w'_{v\to v_2} = w_{v\to v_1}$$

and

$$w'_{v\to v_1} = w_{v\to v_2},$$

and similarly for the biases,

$$b'_{v_2} = b_{v_1}$$

and

$$b'_{v_1} = b_{v_2}.$$

Now assume we define θ' as the parameter obtained from θ by doing the changes described above. Let $x \in \mathbb{R}^{N_0}$. Since we do not change any weights in the previous layers, we would have $f^{l-1}_{\theta'}(x) = f^l_\theta(x)$. The function σ here denotes any activation function. Recalling (1.1.1), one would have

$$\begin{aligned} f^l_{\theta'}(x)_{v_1} &= \sigma\left(\sum_{v\in V_{l-1}} w'_{v\to v_1}(f^{l-1}_\theta(x))_v + b'_{v_1}\right) \\ &= \sigma\left(\sum_{v\in V_{l-1}} w_{v\to v_2}(f^{l-1}_\theta(x))_v + b_{v_2}\right) = f^l_\theta(x)_{v_2}, \end{aligned}$$

and

$$\begin{aligned} f^l_{\theta'}(x)_{v_2} &= \sigma\left(\sum_{v\in V_{l-1}} w'_{v\to v_2}(f^{l-1}_\theta(x))_v + b'_{v_2}\right) \\ &= \sigma\left(\sum_{v\in V_{l-1}} w_{v\to v_1}(f^{l-1}_\theta(x))_v + b_{v_1}\right) = f^l_\theta(x)_{v_1}. \end{aligned}$$

In a quite obvious way, the contents of the neurons v_1 and v_2 are swapped when changing θ to θ'. Now suppose we match this change by changing similarly the outward weights of the two neurons, defining, for each neuron $v \in v_{l+1}$:

$$w'_{v_1\to v} = w_{v_2\to v}$$

and

$$w'_{v_2\to v} = w_{v_1\to v},$$

and suppose these changes are also implemented into θ'. If we consider a neuron $v \in V_{l+1}$, the contribution of the neuron v_1 to $f^{l+1}_\theta(x)_v$ will be $w_{v_1\to v} f^l_\theta(x)_{v_1}$ and the contribution of the neuron v_2 will be $w_{v_2\to v} f^l_\theta(x)_{v_2}$. Now the contribution of the neuron v_1 to $f^{l+1}_{\theta'}(x)_v$ will be $w'_{v_1\to v} f^l_{\theta'}(x)_{v_1} = w_{v_2\to v} f^l_\theta(x)_{v_2}$ and the contribution of the neuron v_2 will be $w'_{v_2\to v} f^l_{\theta'}(x)_{v_2} = w_{v_1\to v} f^l_\theta(x)_{v_1}$. Thus, in the case of θ', the neuron v receives the same information that in the case of θ, although the

roles of v_1 and v_2 are swapped. It is clear that the contribution of the rest of the neurons in V_l is the same for θ as for θ'. This shows that swapping the (inward and outward) weights and biases of two neurons of a same hidden layer does not change the function implemented by the network. By combining such transpositions, we see that any permutation of neurons in a same hidden layer leaves the global function implemented by the network unchanged.

This permutation invariance comes from the structure of fully-connected feed-forward neural networks, where two neurons in a same layer are interchangeable. As such, it is generic and is shared among a wide range of neural architectures.

1.2.2 Activation-dependent symmetries

Other types of symmetries possessed by the parameters of neural networks are the activation-dependent ones. These symmetries indeed reflect the symmetries of the activation function itself. To see that, let $\sigma : \mathbb{R} \to \mathbb{R}$ denote an arbitrary activation function. Assume that σ satisfies a generic relation of the form

$$\forall t \in \mathbb{R}, \qquad \sigma(\lambda t + \mu) = \gamma \sigma(t), \tag{1.2.1}$$

for some given $\lambda, \gamma \in \mathbb{R}^*, \mu \in \mathbb{R}$. This type of relations accounts for the symmetries of many classical activation functions as we will see later in this section.

Considering a neuron v_0 in a hidden layer V_l, suppose we change the inward weights to v_0 as follows:

$$\forall v \in V_{l-1}, \qquad w'_{v \to v_0} = \lambda w_{v \to v_0}$$

and suppose we change the bias as

$$b'_{v_0} = \lambda b_{v_0} + \mu.$$

Now assume we define θ' as the parameter obtained from θ by doing the changes described above. Let $x \in \mathbb{R}^{N_0}$. Since we do not change any weights in the previous layers, we would have $f^{l-1}_{\theta'}(x) = f^{l-1}_{\theta}(x)$. Recalling (1.1.1), one would have

$$f^l_{\theta'}(x)_{v_0} = \sigma\left(\sum_{v \in V_{l-1}} w'_{v \to v_0} f^{l-1}_{\theta}(x)_v + b'_{v_0} \right) \tag{1.2.2}$$

$$= \sigma\left(\sum_{v \in V_{l-1}} \lambda w_{v \to v_0} f^{l-1}_{\theta}(x) + \lambda b_{v_0} + \mu \right) \tag{1.2.3}$$

$$= \gamma \sigma\left(\sum_{v \in V_{l-1}} w_{v \to v_0} f^{l-1}_{\theta}(x) + b_{v_0} \right) \tag{1.2.4}$$

$$= \gamma f^l_{\theta}(x)_{v_0}. \tag{1.2.5}$$

As we can see, the change from θ to θ' multiplies the content of neuron v_0 by γ. Now, in addition to the previous changes, if we multiply all the outward weights of

the neuron v_0 by $\frac{1}{\gamma}$, this will compensate the change of the content of neuron v_0: if we let $w'_{v_0\to v} = \frac{1}{\gamma} w_{v_0\to v}$, then we have

$$w'_{v_0\to v} f_{\theta'}(x)_{v_0} = (\frac{1}{\gamma} w_{v_0\to v})(\gamma f^l_\theta(x)_{v_0}) = w_{v_0\to v} f^l_\theta(x)_{v_0}.$$

As a consequence, the content of the layer V_{l+1} is unchanged, $f^{l+1}_{\theta'}(x) = f^{l+1}_\theta(x)$, and the output of the network is the same: $f_{\theta'}(x) = f_\theta(x)$.

To summarize, let us denote the transformation we just performed by $\tau^{v_0}_{\lambda,\gamma,\mu} : \mathbb{R}^E \times \mathbb{R}^B \to \mathbb{R}^E \times \mathbb{R}^B$. For any $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, the transformed parameter $\theta' = \tau^{v_0}_{\lambda,\gamma,\mu}(\theta)$ is defined by:

$$\begin{cases} w'_{v\to v_0} = \lambda w_{v\to v_0} & \text{for all } v \in V_{l-1} \\ w'_{v_0\to v} = \frac{1}{\gamma} w_{v_0\to v} & \text{for all } v \in V_{l+1} \\ b'_{v_0} = \lambda b_{v_0} + \mu & \\ w'_{v\to v'} = w_{v\to v'} & \text{for all } (v, v') \in E,\ v, v' \neq v_0 \\ b'_v = b_v & \text{for all } v \in B,\ v \neq v_0. \end{cases} \tag{1.2.6}$$

We just showed that for any $\lambda, \gamma \in \mathbb{R}^*, \mu \in \mathbb{R}$ such that the activation σ satisfies the relation (1.2.1), then for any hidden neuron $v_0 \in V_l$, $l \in [\![1, L-1]\!]$, the transformation $\tau^{v_0}_{\lambda,\gamma,\mu} : \mathbb{R}^E \times \mathbb{R}^B \to \mathbb{R}^E \times \mathbb{R}^B$ leaves the function unchanged. For any $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, for any $x \in \mathbb{R}^{N_0}$, we have

$$f_{\tau^{v_0}_{\lambda,\gamma,\mu}(\theta)}(x) = f_\theta(x).$$

Declining this generic result to classical activation functions, we find some well-known invariants.

— For uneven activation functions such as tanh, the relation (1.2.1) is true with $\lambda = \gamma = -1$ and $\mu = 0$. The corresponding invariant transformations $\tau^{v_0}_{-1,-1,0}$ are the well-known 'sign-flips' that have been studied in the literature [181, 4, 61].

— For even activation functions such as the Gaussian activation $\sigma(t) = e^{\frac{-t^2}{2}}$, the relation (1.2.1) is true with $\lambda = -1, \gamma = 1$ and $\mu = 0$. The corresponding invariant transformations $\tau^{v_0}_{-1,1,0}$ are another form of sign-flips that have also been studied in the literature [96].

— For periodic activation functions, such as a sines [171], if T is the period, the relation (1.2.1) holds for $\lambda = \gamma = 1$, $\mu = kT$ for any $k \in \mathbb{Z}$. For each hidden neuron v_0, the transformation $\tau^{v_0}_{1,1,kT}$ simply shifts the bias by k times the period: $b'_{v_0} = b_{v_0} + kT$.

— For monomial activations $\sigma(t) = t^p$, for $p \in \mathbb{N}^*$, the relation (1.2.1) holds for any $\lambda \in \mathbb{R}^*$, with $\gamma = \lambda^p$ and $\mu = 0$. For each hidden neuron v_0, we thus have an infinity of invariant transformations $\tau^{v_0}_{\lambda,\lambda^p,0}$, $\lambda \in \mathbb{R}^*$.

— For the heaviside activation function (or 'step'), the relation (1.2.1) is satisfied for any $\lambda > 0$, with $\gamma = 1$ and $\mu = 0$. Similarly, for each hidden neuron v_0, we thus have an infinity of invariant transformations $\tau^{v_0}_{\lambda,1,0}$, $\lambda \in (0, +\infty)$ [95].

— Finally, for the piecewise-affine functions ReLU and leaky-ReLU, the relation (1.2.1) is satisfied for any $\lambda > 0$, with $\gamma = \lambda$ and $\mu = 0$. For each hidden neuron v_0, we thus have an infinity of invariant transformations $\tau^{v_0}_{\lambda,\lambda,0}$, $\lambda \in (0, +\infty)$. These transformations are well-known (see for instance, amongst many others, [149, 145, 147, 179]), and we refer to them as 'positive rescalings'. Since we focus on ReLU networks, the positive rescalings interests us in particular, and are at the core of a substantial part of the thesis.

The purpose of this synthetic presentation of a whole class of activation-dependent symmetries is to show that there is a common structure to the symmetries of different neural architectures, and to show that there is a direct link between the symmetries of the activation function and the symmetries or redundancies of the parameters of the network. Even though in this thesis we focus on ReLU that is one of the most used nowadays, this similar structure may allow some of the results to transfer to other activation functions.

1.2.3 Group structure of the symmetries

The permutation operations mentioned in Section 1.2.1 above are linear operations on the space of parameters $\mathbb{R}^E \times \mathbb{R}^B$, they are invertible (by the inverse permutation), so they correspond in fact to elements of the linear group $\mathrm{GL}(\mathbb{R}^E \times \mathbb{R}^B)$. In the case $\mu = 0$, the rescaling operations $\tau^{v_0}_{\lambda,\gamma,0}$ are also linear operations, and one can check that the inverse operation of $\tau^{v_0}_{\lambda,\gamma,0}$ is $\tau^{v_0}_{\frac{1}{\lambda},\frac{1}{\gamma},0}$. We can thus consider the subgroup G of $\mathrm{GL}(\mathbb{R}^E \times \mathbb{R}^B)$ generated by all the permutation and rescaling operations. Since G is generated by transformations that do not change the function implemented by the network, the same applies to any transformation in G. The observation that equivalent transformations of the parameters of neural networks are equipped with a group structure is not new [88].

Some natural questions then arise, amongst which:

— Is there a bigger subgroup G' of $\mathrm{GL}(\mathbb{R}^E \times \mathbb{R}^B)$, $G' \supset G$ that leaves the functions f_θ invariant for any $\theta \in \mathbb{R}^E \times \mathbb{R}^B$?

— We know that G transforms the parameters θ while leaving the functions f_θ invariant. Do we have the reciprocal? If $f_\theta = f_{\theta'}$, does there exist $g \in G$ such that $\theta' = g \cdot \theta$?

— What is the structure of G?

— What are the orbits of parameters in $\mathbb{R}^E \times \mathbb{R}^B$ under the action of G?

— How are the different features of a network modified under the action of G?

— Can we exhibit an interesting subset of parameters $H \subset \mathbb{R}^E \times \mathbb{R}^B$ that represents all orbits, that is $G \cdot H = \mathbb{R}^E \times \mathbb{R}^B$?

Fully answering these questions goes well beyond the scope of this thesis. Nevertheless, in the case of ReLU networks, we can mention that the first two questions are directly related to the question of identifiability - we refer to the corresponding section. Furthermore, regarding the fourth question about the nature of the orbits, we can mention that the local geometric structure of these orbits is related to the

question of the local structure of the 'pre-image' set we study in Chapter 5.

Finally, regarding the third question, without developing a complete characterization of G and its group structure, we can at least write down what the elements of G are. To do so, it is easier to adopt the formalism of the layers. We can infer from Chapter 3, Section 3.3.3 and notably (3.3.6) that if $g \in G$ is such a transformation, then there exist

— permutation matrices $(P^0, P^1, \dots, P^L) \in \mathbb{R}^{N_0 \times N_0} \times \cdots \times \mathbb{R}^{N_L \times N_L}$, with $P^0 = \mathrm{Id}_{N_0}$ and $P^L = \mathrm{Id}_{N_L}$,

— diagonal matrices $(D^0, D^1, \dots, D^L) \in \mathbb{R}^{N_0 \times N_0} \times \cdots \times \mathbb{R}^{N_L \times N_L}$, with positive diagonal coefficients and such that $D^0 = \mathrm{Id}_{N_0}$ and $D^L = \mathrm{Id}_{N_L}$,

such that for any $\theta \in \mathbb{R}^p$, is $\tilde{\theta} = g \cdot \theta$, then for all $l \in [\![1, L]\!]$, we have

$$\begin{cases} \tilde{W}^l = P^l D^l W^l (D^{l-1})^{-1} (P^{l-1})^{-1} \\ \tilde{b}^l = P^l D^l b^l, \end{cases} \tag{1.2.7}$$

where W^l, b^l denote the weights and biases corresponding to θ and $\tilde{W}^l, \tilde{b}^l$ those corresponding to $\tilde{\theta}$.

Therefore, choosing an element of G is equivalent to choosing $L-1$ diagonal matrices with positive diagonal coefficients and $L-1$ permutation matrices. We can remark that G is infinite due to the infinite number of positive rescaling factors (the diagonal elements of the matrices). This is in contrast, for instance, to the sign-flips in the tanh case, whose number is finite.

A group action always defines an equivalence relation. We can thus define the following relation: for all $\theta, \tilde{\theta} \in \mathbb{R}^p$, we say that θ and $\tilde{\theta}$ are **equivalent modulo permutation and positive rescaling**, or simply equivalent, and we write $\theta \sim \tilde{\theta}$, if there exists $g \in G$ such that $g \cdot \theta = \tilde{\theta}$, or in other words, if θ and $\tilde{\theta}$ satisfy relation (2.2.7). The equivalence classes modulo $\sim$ are the orbits under the action of G on $\mathbb{R}^p$.

1.3 Identifiability: from linear models to neural networks

1.3.1 Introduction to identifiability: the linear model in finite dimension

Identifiability is a classical notion in statistics [143]. Broadly speaking, it means, given a parametric model $\{P_\theta, \theta \in \mathcal{P}\}$, where for all $\theta \in \mathcal{P}$, P_θ is a probability distribution, that whenever $\theta_1 \neq \theta_2$, we have $P_{\theta_1} \neq P_{\theta_2}$. It is generally desired as it is necessary in order to estimate the parameter θ. This form of identifiability is the general, theoretical form of identifiability. It means that if we had total knowledge of P_θ, we could identify θ.

In particular, let us consider a linear model of the form

$$Y = X\beta + \epsilon,$$

where $X \in \mathbb{R}^{n \times p}$ is deterministic, $\beta \in \mathbb{R}^p$ is the parameter of the model, and ϵ is a random vector, following a centered Gaussian distribution. Here typically the lines of X will be examples $x^{(1)}, \dots, x^{(n)}$. In this model, if the matrix X does not have full column rank, then there exist infinitely many choices for β that lead to the same distribution for Y. In particular, if X has more columns than rows, that is if $n < p$, by definition the model cannot be identifiable. In the classical setting of linear regression, we have $n > p$. However, even in the case $n > p$, the lack of identifiability can still arise from the redundancy of two or more variables. One way of enforcing identifiability is thus removing the redundant variables, which means removing redundant columns of X one by one until identifiability is reached.

1.3.2 Identifiability in high dimension

1.3.2.1 The linear model in high dimension

More recently, with the development of high-dimensional data, the linear model has been considered in a different setting, in which the dimension p greatly exceeds the number n of examples that we have: we have $p \gg n$. This is often the case, for instance, with genomic data. As mentioned in the previous section, when $p > n$, identifiability cannot hold. However we would still like to find the 'true' β. A classical assumption that is then made is sparsity: we assume that β only has a few nonzero coefficients. This means that only a few variables actually impact Y, and finding which ones is a task referred to as 'variable selection'. The sparsity of a vector can be measured with the ℓ_0 'norm', which simply counts the number of nonzero entries:

$$\|\beta\|_0 = |\{j \in [\![1, p]\!] \mid \beta_j \neq 0\}|.$$

1.3.2.2 Compressed sensing

In the 2000's, the field of compressed sensing received a fair amount of attention, following the pioneering work [54, 38]. The setting is very similar to the linear model above: we consider an unknown signal $x \in \mathbb{R}^p$, which is accessed to through a series of linear measurements, that is a list of n vectors a_i, for which we observe the quantity $y_i = \langle a_i, x \rangle$. If we denote by $A \in \mathbb{R}^{n \times p}$ the matrix whose i^{th} line is a_i^T and by y the vector $(y_1, \dots, y_n)^T$, this can be rewritten as

$$y = Ax.$$

This is the same linear inverse problem as before, with A instead of X and the unknown vector being x instead of β. Here we do not consider a noise vector ϵ, although it could be added to take into account the measurement errors. The setting of compressed sensing focuses on problems for which the number n of measurements is limited, for which we have $n \ll p$. As a consequence, A cannot be injective and identifiability does not hold by definition. To obtain identifiability one must add additional constraints on x that reduce the set of solutions to one element.

Again, the standard assumption is sparsity, and we can reformulate the problem as a ℓ_0 minimization problem:

$$\begin{aligned} \min_x \quad & \|x\|_0 \\ \text{s.t.} \quad & y = Ax. \end{aligned} \tag{1.3.1}$$

Interestingly, theory shows that in many cases this problem admits a unique minimizer, and identifiability is thus guaranteed. In that case, if we denote k the minimal sparsity value, and if the true vector x_0 is k-sparse, solving this problem actually recovers x_0.

However, the ℓ_0 norm is hard to optimize. In fact, the problem (1.3.1) has been shown to be NP-hard. A very interesting finding is that it is possible to use the ℓ_1 norm as a convex surrogate of the ℓ_0 norm. The problem then becomes

$$\begin{aligned} \min_x \quad & \|x\|_1 \\ \text{s.t.} \quad & y = Ax \end{aligned} \tag{1.3.2}$$

It has been shown that in many settings, the solution to (1.3.2) is the same as the solution to (1.3.1), meaning that solving (1.3.2) exactly recovers the k-sparse vector x_0. Contrary to (1.3.1), the problem (1.3.2) is practically solvable in polynomial time with classical optimization tools [101].

1.3.3 Identifiability for ReLU Neural networks

1.3.3.1 Several definitions of identifiability

The question of identifiability can be formulated for neural networks: as seen previously, neural networks admit parameters in the form of weights and biases, and given a parameter choice $\theta \in \mathbb{R}^p$, a network implements a function f_θ. The difference with the linear model is that f_θ is not linear with respect to θ in general. The motivations also differ; they are discussed in the next section.

Let us consider a ReLU network $(f_\theta)_{\theta \in \mathbb{R}^p}$. Naively, the general question of identifiability is: if $f_{\theta_1} = f_{\theta_2}$, do we have $\theta_1 = \theta_2$? In fact, as we have seen in Section 1.2, it is possible to transform the parameters of a ReLU network, without changing the function it implements. Indeed, recalling the equivalence relation $\sim$ defined at the end of Section 1.2.3, for any $\theta \in \mathbb{R}^p = \mathbb{R}^E \times \mathbb{R}^B$, we know that if $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$ satisfies $\theta \sim \theta'$, then we have $f_\theta = f'_\theta$. We must thus loosen the definition of identifiability if we want it to be meaningful for ReLU networks.

Definition 1. We say that a parameter $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ of a ReLU network is **identifiable** if for any $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, then

$$f_\theta = f_{\theta'} \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

Definition 2. We say that a parameter $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ of a ReLU network is **locally identifiable** if there exists $\epsilon_\theta > 0$ such that, for any $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, then

$$\|\theta - \theta'\| < \epsilon_\theta \quad \text{and} \quad f_\theta = f_{\theta'} \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

These definition assume that the functions f_θ and $f_{\theta'}$ implemented by the networks of parameters θ and θ' are equal, which means that they coincide on the entirety of the input space. However the distribution of the inputs may have a support smaller than the entire set $\mathbb{R}^{N_0}$, which means that in practice we only get to consider $f_\theta(x)$ and $f_{\theta'}(x)$ in a subset $\Omega \subset \mathbb{R}^{N_0}$. In that case, if f_θ and $f_{\theta'}$ coincide on Ω, it is as if they were equal to us. Taking this into account, we can consider the following definition.

Definition 3. We say that a parameter $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ of a ReLU network is **identifiable from** Ω if for any $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, then

$$\forall x \in \Omega,\ f_\theta(x) = f_{\theta'}(x) \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

The same adaptation can be made for local identifiability.

Definition 4. We say that a parameter $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ of a ReLU network is **locally identifiable from** Ω if there exists $\epsilon_\theta > 0$ such that, for any $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, then

$$\|\theta - \theta'\| < \epsilon_\theta \quad \text{and} \quad \forall x \in \Omega,\ f_\theta(x) = f_{\theta'}(x) \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

A particular case is when Ω is finite. In that case we have a finite list of examples $x^{(i)}$, $1 \leq i \leq n$ for some integer n. The question is thus, if f_θ and $f_{\theta'}$ coincide on these n inputs, do we have $\theta \sim \theta'$? This declination of identifiability is interesting because it corresponds to practical settings. Of course, identifiability from a subset Ω and especially identifiability from a finite list of inputs are harder to achieve since they do not assume full knowledge of f_θ.

1.3.3.2 Inverse stability and stable recovery

We explicit in this section two notions that are close to identifiability and sometimes referred to as identifiability themselves.

The first notion is **inverse stability**. The general idea is to extend the definition of identifiability to small perturbations. Suppose that the functions f_θ and $f_{\theta'}$ are not perfectly equal, but they are close in some sense. The parameters θ and θ' will not be equal, but one can wonder if they are close for some notion of distance. This is an interesting extension to the notion of identifiability, to begin with, because in practice we are generally subject to all sorts of errors which prevent perfect equality. Further, inverse stability can have a lot of theoretical and practical benefits, for instance in terms of optimization [58]. We give here an informal definition of inverse stability

Definition 5. We say that inverse stability holds if there exists $\alpha > 0$ such that for any $\theta, \theta' \in \mathbb{R}^E \times \mathbb{R}^B$, then

$$\|\theta - \theta'\|_{par} \leq \alpha \|f_\theta - f_{\theta'}\|_{fct}.$$

In this definition, many elements must be specified. First, one must determine a meaningful 'norm' (here we put quote marks since we consider norms in a loose sense) $\|\cdot\|_{par}$ on the set of parameters. Such a 'norm' must take into account the equivalences of parameters. Indeed, as we have discussed before, any parameter in a same equivalence class has the same function, so the quantity $\|\theta - \theta'\|_{par}$ should not depend on the chosen member of the class. For instance, any classical norm on the space of parameters $\mathbb{R}^E \times \mathbb{R}^B$ such as the euclidean norm $\|\cdot\|_2$ would fail for ReLU networks since we can use the positive rescaling operation on a parameter θ to construct a sequence $\theta_n \in \mathbb{R}^E \times \mathbb{R}^B$ such that, for all $n \in \mathbb{N}$, $\theta \sim \theta_n$ but $\|\theta - \theta_n\|_2 \to \infty$, which would prevent any kind of bound such as the one in the definition to hold. As an example, [117] propose a norm on the space of parameters for deep structured matrix factorization problems which takes this into account.

Another element that must be specified is the 'norm' $\|\cdot\|_{fct}$ on the space of functions of the network. This is even more complex since the range of norms on such a functional space is very wide. The authors of [58] propose a discussion on the subject, and show some counter-examples for which inverse stability does not hold They show that norms which take the gradients into account such as Sobolev norms, are better suited to characterize the distance between two network realizations.

Finally, distinctly from identifiability and inverse stability, **stable recovery** denotes the search for algorithms which are practically able to recover the parameter θ of a neural network, or an equivalent one, up to a certain error.

1.3.4 Motivations

The motivations for studying identifiability in the case of neural networks are not the same as in the statistical settings presented in Sections 1.3.1 and 1.3.2. In statistics, one of the motivations for estimating the parameters of a model is because they have a physical meaning. In that case, identifiability is necessary in order to estimate the exact value of a particular parameter. In deep learning, it is possible to find in the literature some specific settings where part of the parameters of a neural network have a physical meaning [60, 197], which can in that case be a reason for seeking identifiability. However, most of the time, the parameters of neural networks do not have a physical meaning, and the motivations for studying identifiability are different. We describe some of the motivations in the present section. They are diverse and range from the most practical to the most theoretical ones.

1.3.4.1 Model inversion attacks

Machine Learning models, and notably neural networks, are vastly used nowadays in a wide range of tasks. This development is accompanied by a need for guarantees regarding safety, robustness and privacy of the models used. It is frequent that the user of the model is not the same as the provider [158]. In such a framework, users can query models that are already trained, provided by cloud services, either freely or with a cost associated to each query. The users generally do

not have full access to the model. In particular they do not necessarily know which architecture is used, which optimizer or hyperparameters were used during training, and above all, what the trained parameters are. It can indeed be important for the model providers to keep this information confidential, for reasons we explain below. However, model extraction attacks have been a growing topic over the last years [139]: attacks that are able to uncover some or all the hidden information thanks to the queries. In particular, for neural networks, some algorithms are able to recover in practice the parameters of a neural network or the function implemented by the neural network from queries [39, 159].

A problem for privacy The extraction of the parameters of a neural network by attackers can be a major issue for privacy. Indeed, it is now well known that neural networks memorize some elements of their training databases, storing this information in their parameters. For instance, in 2015, the authors of [67] showed how they were able to reconstruct some pictures of the database used to train a facial recognition system. The technique takes advantage of a confidence indicator returned by the system. It necessitates to compute the gradient $\nabla f_\theta(x)$ of $\theta \mapsto f_\theta(x)$, for any input x. As the authors show, this can be done with numerical methods, but it is very costly and works only when the dimension is small enough. In the majority of the experiments, the authors assume knowledge of θ. More recently, the authors of [85] use the implicit bias of neural networks (see Section 1.4.4) to recover several images from the training database of a neural network. The algorithm is different but the method also assumes knowledge of the trained parameters θ. It thus becomes clear that preventing users from inferring the trained parameters θ is essential for privacy.

A problem for robustness Another major concern is that a network whose parameters are accessible may be less robust against adversarial attacks. Indeed, neural networks are well-known to be vulnerable to such attacks, and extensive work has been done in the past decade to understand better and prevent this phenomenon. The ability of malicious users to trick a neural network based computer vision system can be a major problem, for instance in the case of autonomous vehicles and road signs recognition. Having access to the trained parameters of the network can make a difference in such attacks. Indeed, if some black-box adversarial attacks do exist [180, 165, 49], that is, attacks that work when being only able to query a network, many attacks use the knowledge of the parameters of the network, at least to compute the gradients [183, 79, 104, 142, 41, 127, 126, 13].

A problem for intellectual property Finally, the parameters of a trained neural network can be valuable as they can be the result of costly trainings. Being able to extract the trained parameters of a network allows to replicate it which can thus be a problem for intellectual property of the network or of the training database [196]. Extracting the parameters can also help extracting other valuable

information. For example, the authors of [191] exhibit a hyperparameter extraction attack, which is able to recover a hyperparameter used to train a model (which can be a neural network), by solving a system of equations. This attack assumes the knowledge of the parameterization θ, which shows again that the ability to extract the parameters of a network is crucial.

Model inversion attacks need identifiability The problem of model extraction attacks and specially of parameter extraction is directly linked to the question of identifiability. Indeed, studying theoretically when the function implemented by a network uniquely characterizes its parameters, and what attributes of the function makes the parameters identifiable helps to understand which networks are vulnerable to parameter extraction and how to prevent it. For ReLU networks, existing works on identifiability, including ours, show how the boundaries between linear regions, which take the form of pieces of hyperplanes, convey the most information on the parameters of a network. Building on this knowledge, the authors of [43] developed a method of preventing parameters extraction by artificially complexifying the network without changing its global behavior. This method adds layers to the original network that do not change the global behavior of the function, but that increase significantly the number of linear regions and separating boundaries, in order to make parameter recovery intractable.

Another way of leveraging the knowledge on identifiability to understand better the model inversion attacks is to provide practical conditions allowing to test identifiability. Indeed, the user of a network has access to a list of inputs (the queries) and the corresponding outputs of the network (the responses to the queries). The question is then if this information uniquely characterizes the parameters: it is the question of identifiability from a finite list of inputs (see Section 1.3.3). It is clear that if the number of queries is small (for instance one single query), the information obtained will not suffice to characterize the parameters of the network. For the provider, a way of preventing the recovery of the parameters can then be by guaranteeing that identifiability does not hold, which means checking that a necessary condition of identifiability is not met. In that case, the parameters are not uniquely characterized by the information available, without more prior it is impossible to infer the parameters. An easy way of guaranteeing this can simply be by limiting the number of queries accessible to the user. On the opposite side, guaranteeing that identifiability holds is interesting in the position of an attacker. If the attacker knows the inputs X, to the corresponding outputs $f_\theta(X)$, and is able to compute a $\tilde{\theta}$ such that $f_{\tilde{\theta}}(X) = f_\theta(X)$, the question then becomes: does this guarantee that $\tilde{\theta} = \theta$ or shall the attacker expand X with new queries? The attacker needs a sufficient condition of identifiability.

1.3.4.2 Theoretical guarantees

A frequently mentioned shortcoming of neural networks is that they work as black-box models over which we have little understanding and control. This issue

has been the motivation for a great amount of work in the past decade, which has helped to bring a better light over neural networks and their complex behavior. In spite of this progress, the need for theoretical guarantees and for a better grasp of neural networks behavior persists. Identifiability can also play a role in this effort.

For instance, one line of investigation to understand the training and generalization properties of neural networks is the student-teacher framework, in which one assumes that we train a network -the student- with data that are generated by an unknown network -the teacher. In such a setting, a natural question is in which case training the student actually implies the recovery of the parameters of the teacher network [62]. In that case, the training can be seen as a classical estimation task: finding the true parameters of the model. In particular, if the parameters of the teacher network are identifiable from the training data, then training the student to perfectly fit the training examples is rigorously equivalent to recovering the parameters of the teacher. Numerous articles have followed this framework, see for instance [93, 32, 109, 174, 200].

In this framework, one consequence of identifiability from the training data is that the global minimizer of the empirical risk is unique. As a consequence, provided the training process is able to reach the global minimizer, there is no variability due to the optimization parameters (choice of the algorithm, step, number of epochs...) and to stochasticity (for stochastic optimizers). This guarantees more control on neural network training, in the form of what we could call the reproducibility of the training process. Even if recent works on double descent phenomena, e.g. [24], highlight a benefit of overparameterization (in which we typically will not be able to guarantee identifiability) for increasing prediction performances, a user may be interested in having a number of parameters small enough to retain identifiability, if the loss of performance is mild compared to overparameterization.

The link between the space of parameters and the space of functions implemented by a neural network is complex. As shown by [144], the set of functions $(f_\theta)_{\theta\in\mathbb{R}^p}$ implemented by a neural network with classical activation functions (including ReLU) is neither globally nor locally convex. Furthermore, this set is not closed in any of the L^p spaces. One consequence of this non closedness is for instance the explosion of weights when a sequence of functions f_{θ_n} converges to a limit f that is not in the space of functions of the network. Even worse, the authors show that the sequence f_{θ_n} can converge to a function f_{θ^*} that does belong to the set of functions realized by the network, while the sequence (θ_n) diverges to infinity. This is due to the fact that a function implemented by a network can have an infinite number of parameterizations.

To solve the issues posed by [144], the authors of [58] show that by appropriately choosing the norms on the functions space, and by restricting the set of parameters to prevent degenerate parameterizations, one can guarantee inverse stability. As shown by the authors, inverse stability alongside with appropriate regularization constraints allows to guarantee good optimization properties, such as quasi-optimality of local minima of the objective function.

1.3.4.3 Identifiability as a measure of the diversity of a sample

Identifiability can inform us on a neural network, but it can also be a useful information to characterize the diversity or the representativity of a sample. Suppose for instance that we have a trained neural network with a given parameter $\theta \in \mathbb{R}^p$, which we would like to test, and to do so, we observe the outputs $f_\theta(x^{(i)})$ of the network for a list of inputs $x^{(i)}, i \in [\![1, n]\!]$. Then, identifiability of θ from the testing sample can serve as a measure of if the sample is rich enough. If θ is not identifiable, it is not fully characterized by the sample, meaning that one could add new testing examples to characterize better the function f_θ implemented by the neural network.

1.3.5 Related works

1.3.5.1 Identifiability

Identifiability of the parameters of neural networks has been the topic of a fair amount of work. For smooth activation functions, some results were already established in the 1990s. For shallow networks, results exist for activation functions amongst which tanh [181, 4], the logistic sigmoid [105], or the Gaussian and rational functions [96]. For deep networks, [61] shows that with tanh as activation function, with only a few generic conditions on the parameters, two networks that implement the same function have the same architecture and the same parameters up to some permutations and sign-flip operations.

In the case of ReLU networks, we have seen that two operations are well known to preserve the function implemented by the network: permutation and positive rescaling. These operations define equivalence classes on the set of parameters, and we can at best identify the parameters of a network up to these equivalences. It is shown in [147] that these operations are the only generic operations of this kind for ReLU networks with nonincreasing number of neurons per layer. Indeed, they show that for any fully-connected ReLU network architecture with nonincreasing number of neurons per layer, for any nonempty open set Ω, there exists a parameterization θ such that for any other parameterization $\tilde{\theta}$ satisfying some generic assumption, if $f_{\tilde{\theta}}$ coincides with f_θ on Ω, then $\tilde{\theta}$ is in the equivalence class of θ modulo permutation and positive rescaling.

In the case of shallow ReLU networks, [145] establishes a sufficient condition on the parameters for identifiability. If the condition is satisfied by two two-layer fully-connected feedforward ReLU networks whose functions coincide on all the input space, then the parameters of one network can be obtained from the parameters of the other network by permutation and positive rescaling.

In the case of deep ReLU networks, [159] gives a sufficient condition to be able to reconstruct the architecture, weights and biases of a deep ReLU network by knowing its input-output map on all the input space.

Another kind of property is local identifiability, which is identifiability of a parameter θ amongst a set of parameters that are close to θ, as defined in Section 1.3.3.1. [179] studies this property for shallow and deep networks. For a deep ReLU

network, it first shows that under a trivial assumption, general identifiability up to permutation and positive rescaling implies local identifiability up to positive rescaling, and that the non-existence of 'twin' neurons is necessary to identifiability and local identifiability. Then, [179] gives an abstract necessary and sufficient condition on θ such that there exists a well-chosen *finite* set Ω from which local identifiability holds up to positive rescalings, and it gives a bound on the size of the set.

Finally, another line of work that can be linked to identifiability is the field of lossless compression of neural networks [169, 170].

1.3.5.2 Inverse stability and stable recovery

Establishing identifiability properties is a first step towards establishing inverse stability properties and studying stable recovery algorithms, as defined in Section 1.3.3.2.

Inverse stability does not hold in general with the uniform norm for fully-connected feedforward neural networks. Indeed, [144] shows that for any depth, for any architecture with at least 3 neurons in the first hidden layer and any practically used activation function, there exists a sequence of networks whose function tends uniformly to 0 while any parameterization of these networks tends to infinity.

Many inverse stability and stable recovery results already exist for shallow networks. [58] studies inverse stability directly up to functional equivalence classes. The authors show that inverse stability has interesting implications in terms of optimization. Referring to the counter-example given by [144], the authors of [58] argue that the Sobolev norm is more suited than the uniform norm to the problem of inverse stability. With this norm, they concretely establish an inverse stability result on shallow ReLU networks without bias, under a few conditions on the parameters.

When it comes to stable recovery algorithms, [68] provides a sample complexity under which one can recover the parameters of a shallow network with sigmoid activation function using cross-entropy as a loss. For shallow fully-connected ReLU networks, without bias and with Gaussian input, [70, 200, 201, 203] study the stable recovery of the parameters of a teacher network. They give a sample complexity under which minimizing the empirical risk allows to recover the parameters of the network. [109] studies the same configuration but with an identity mapping that skips one layer. ReLU networks can also be used to recover a network with absolute value as activation function [108]. In fact, a neuron with absolute value can be seen as a sum of two ReLU neurons.

Some results also exist in the case of shallow convolutional networks. [32, 199, 198, 56] establish stable recovery results for convolutional ReLU networks with no overlapping. [93] gives a result in the case of a sigmoidal activation function. The case of convolutional ReLU networks with overlapping is studied in [76].

Stability and stable recovery for *deep* networks is a more complicated question. A few results exist on the subject, but it stays mostly unexplored.

Among them, for deep structured linear networks, [119, 117, 115] use a tensorial lifting technique to establish inverse stability properties. [119, 117] establish nec-

essary and sufficient conditions of inverse stability for a general constraint on the parameters defining the network. [115] specializes the analysis to the sparsity constraint on the parameters, and obtains necessary and sufficient conditions of inverse stability.

The authors of [8] consider deep feed-forward networks with Heavyside activation function which are very sparse and randomly generated. They show that these can be learned with high probability one layer after another.

The authors of [168] consider a deep feed-forward neural network, with an activation function that can be, inter alia, ReLU, sigmoid or softmax. They show that, if the input is Gaussian or its distribution is known, and if the weight matrix of the first layer is sparse, then a method based on moments and sparse dictionary learning can retrieve it exactly. Nothing is said about the stability or the estimation of the other layers.

For deep ReLU networks, in the case where one has full access to the function implemented by the network [159] provides a practical algorithm able to approximately recover the parameters modulo permutation and rescaling, and [39] reconstructs a functionally equivalent network, formulating it as a cryptanalytic problem.

1.3.6 Our contributions : Chapters 3 and 4

We present here two contributions of this thesis on the question of identifiability of deep ReLU networks.

Chapter 3: parameter identifiability from a domain Ω The first contribution, described in Chapter 3, is a result establishing sufficient conditions of identifiability for deep feedforward ReLU neural networks. We suppose that the function f_θ implemented by the network is known over a subdomain Ω of the input space $\mathbb{R}^d$. By analyzing the piecewise-affine structure of ReLU networks, and in particular the structure of the singularities of f_θ and of functions implemented by subnetworks, we derive a set of sufficient conditions, named **P** and defined in Section 3.4.1, under which identifiability holds. The main theorem can be found as Theorem 17 in Section 3.4.2.1.

Chapter 4: local identifiability from a finite sample The second contribution, described in Chapter 4, focuses on local identifiability of a deep ReLU network from a finite sample X. It provides two conditions, a necessary condition C_N and a sufficient condition C_S of local identifiability. Since local identifiability is necessary for global identifiability, the necessary condition is also a necessary condition of global identifiability. Both conditions C_N and C_S apply on the ranks of two operators constructed from θ. The main results can be found in Section 4.4 as Theorem 78 for the necessary condition and Theorem 79 for the sufficient condition.

1.4 Complexity, regularization of neural networks

1.4.1 Some elements of Statistical Learning Theory

The development of machine learning has led to a need of theoretical tools for a better understanding and a better control of the algorithms. We focus here on training strategies that rely on empirical risk minimization. We consider a machine learning model $(f_\theta)_{\theta\in\mathcal{P}}$, where $\mathcal{P}$ is a set of parameters that can be other than $\mathbb{R}^p$. Recall the definition of the risk $R(\theta)$ and the empirical risk $\widehat{R}(\theta)$ given in Section 1.1.2. Empirical risk minimization denotes strategies in which we choose θ by trying to minimize the empirical risk $\hat{R}(\theta)$. Naively, one could hope that the empirical risk $\widehat{R}(\theta)$ of the obtained parameter $\theta \in \mathcal{P}$ is not far from the true risk $R(\theta)$ simply because of the law of large numbers. This would be true if θ (and thus the function f_θ) was independent of the training sample. However, since θ is obtained by minimizing the empirical risk, which depends on the training sample, θ heavily depends on the sample.

To bound the gap between $\widehat{R}(\theta)$ and $R(\theta)$, an idea is then to show that the set of functions $(f_\theta)_{\theta\in\mathcal{P}}$ is not too rich, meaning that in some sense θ cannot depend 'too much' on the training sample. For instance considering a very simple example where there are only two parameters, $\mathcal{P} = \{\theta_1, \theta_2\}$, one could apply the law of large numbers twice to bound the generalization error for θ_1 and θ_2, and then use an union bound to bound the generalization error over $\mathcal{P}$. This would allow to guarantee that even after choosing θ according to the training sample, the generalization error could be bounded. Now imagine we add more parameters θ_i in $\mathcal{P}$. The more choices there are for θ, the more quantities we have to bound simultaneously in the union bound, and the less we are able to bound the gap between $\widehat{R}(\theta)$ and $R(\theta)$. We see here the intuition that the richer the set of functions (f_θ) is, the less we can expect to bound the generalization error.

In general, the sets of functions $(f_\theta)_{\theta\in\mathcal{P}}$ considered are infinite, so the above approach needs to be improved. We need tools to quantify how rich and diverse the set of functions (f_θ) is. This is where a tool such as VC-dimension, first introduced by Vladimir Vapnik and Alexey Chervonenkis, intervenes. In a binary classification setting, VC-dimension allows to quantify the complexity of a family of classifiers. If we consider a set of examples $x^{(i)}, i \in [\![1, n]\!]$, we say that the set $\{x^{(1)}, \dots, x^{(n)}\}$ is *shattered* by the family $(f_\theta)_{\theta\in\mathcal{P}}$ if for any binary label assignment $\epsilon^{(1)}, \dots, \epsilon^{(n)} \in \{0, 1\}$, there exists a classifier f_θ that classifies perfectly the points, i.e. $f_\theta(x^{(i)}) = \epsilon^{(i)}$, for all $i \in [\![1, n]\!]$. The VC-dimension of a model $(f_\theta)_{\theta\in\mathcal{P}}$ is then the biggest $n \in \mathbb{N}$ such that there exists a set of examples $x^{(i)}, i \in [\![1, n]\!]$ that is shattered by $(f_\theta)_{\theta\in\mathcal{P}}$. This notion is extensible to multiclass classification. It is intuitive that the bigger the sets of examples we are able to shatter with $(f_\theta)_{\theta\in\mathcal{P}}$, the richer the model $(f_\theta)_{\theta\in\mathcal{P}}$ is.

VC-dimension is an efficient complexity measure that allows to derive generalization bounds for many machine learning models. Again, the general idea is that the more simple a class of functions is (i.e. the lowest its VC-dimension), the more

we are able to bound the generalization error. The choice of the complexity of a model is thus important, as we discuss in the next section.

1.4.2 The bias-variance trade-off

When trying to choose the best model $(f_\theta)_{\theta\in\mathcal{P}}$ for a learning task, a key component is the complexity of the model. A model too simple, with not enough functions, would adapt difficultly to the given task. It may be hard to fit the training data and to make the empirical risk $\widehat{R}(\theta)$ low, and there may not exist a function f_θ such that the risk $R(\theta)$ is low.

On the other side of the spectrum, a complex model may be rich enough to contain a function f_θ such that the empirical risk $\widehat{R}(\theta)$ is low. However in that case, the complexity of the model makes it more difficult to bound the generalization gap, and indeed, the empirical risk $\widehat{R}(\theta)$ and the true risk $R(\theta)$ may dramatically differ. By fitting too closely the training data, one may fit its noise, which would make the θ obtained by empirical risk minimization too dependent on the training data. This problem is known under the name of overfitting.

In the classical machine learning paradigm, one must thus find a sweet spot between these two phenomenons: choosing the right complexity for the right problem. This is known as the bias-variance trade-off.

For feedforward neural networks, the complexity depends on the architecture of the network: the activation function used, and the size of the network, i.e. the number of layers (depth) and the number of neurons for each layer (width). The choice of the activation is indeed relevant, as for some activation functions, even small networks have infinite VC-dimension. This size of the network is reflected in the number of parameters: the more layers and neurons in the network, the higher the number of parameters.

Extensive work has been done to bound the VC-dimension of neural networks, for smooth activation functions in the first place [19, 6], as well as piecewise-linear activation such as ReLU [16]. The existing bounds for ReLU networks scale at least linearly with the number of parameters [16, 21], which confirms that the number of parameters represent the complexity of a network.

1.4.3 The paradox of deep learning

Since VC-dimension scales at least linearly with the number of parameters, to bound the generalization error, one should make sure that the size of the training sample is large compared to the number of parameters. However, this is at odds with the modern setting of Deep Learning, in which highly overparameterized neural networks have shown great performance in a wide range of situations. Even worse, in many settings, increasing the number of parameters of the network improve generalization performances! In such settings, the existing bounds based on VC-dimension are vacuous.

One could try to find tighter bounds for neural networks, however the existing

bounds are close to being tight [16]. One could also try to find better tools than VC-dimension to account for the complexity of the function classes induced by neural networks. Indeed, it is for instance possible to change the VC-dimension of a network from finite to infinite by adding an arbitrarily small perturbation to ReLU [21]. Such a perturbation would not change the global behavior of the network, however VC-dimension captures fine-grained properties of models, so it is affected by such a change. Other measures might be more robust and more efficient to reflect the complexity of the class of functions implemented by a neural network.

Nevertheless, the problem appears to be more general. Indeed, as shown by [134, 195], neural networks that generalize well on some classification tasks (such as MNIST) are powerful enough to perfectly fit data with random labels. This shows that overparameterized neural networks are indeed very powerful and represent very rich class of functions. In that respect, there exist many parameterizations θ such that the corresponding f_θ is able to fit the training data. The set of minimizers of the objective functions minimized by the optimization algorithms such as SGD are large [51, 106] and contain elements that generalize poorly [192, 134]. Despite this, the optimizers are able to choose functions that generalize well in practice. This shows that global analysis, which considers worst-case bounds on $(f_\theta)_{\theta \in \mathbb{R}^p}$, with the worst possible network that fits the data, will inevitably fail to explain generalization. Instead, there is a need for local complexity measures, that describe the complexity of the functions actually implemented by neural networks optimized via gradient descent.

1.4.4 Implicit regularization and local complexity measures

A substantial research effort has been made to obtain new complexity measures and new generalization bounds for neural networks. The complexity measures serve either as a descriptive tool showing that the optimizers such as SGD are implicitly biased towards functions that are simple in some sense and that are able to generalize, or the measures can be explicitly added as a regularization during optimization.

One can list desiderata that an ideal complexity measure should satisfy [186]. An ideal complexity measure should apply to networks used in practice, and should be able to account for the prediction performances of the different architectures. In particular, it should not grow with the number of parameters, since as explained above, adding parameters even after fitting perfectly the training dataset does not hurt prediction, or in some cases even improves it. The complexity measure should also have the right dependency to the number of training examples [129]. The measure should also account for the complexity of the datasets. For instance, classification on CIFAR10 is harder than on MNIST, and classification on CIFAR100 is harder than classification on CIFAR10. When corrupting part of the labels of the training dataset, as is done in many experiments, one should see the complexity measure grow. Apart from the dataset, the complexity measure should reflect the performances of different optimization methods: optimizer, choice of hyperparameters, regularization techniques... To improve our understanding of neural networks,

a complexity measure should also have a theoretical explanation and ideally come with a generalization bound being able to predict generalization. Such a bound would ideally be close to the true generalization error, but at least, it should be non-vacuous (i.e. it should predict an error rate smaller than 100%, otherwise it is uninformative). Last but not least, a complexity measure should be practically computable in an efficient way. As far as we know, there exist measures satisfying some of the aforementioned desiderata, but none that is able to satisfy, if not all, even just a majority of them.

Many bounds involve the norm of the weights of neural networks, either directly, or measured as a distance to the weights at initialization [137, 134, 15, 77, 128]. Another type of complexity measure that has been investigated is the flatness of the minimum of the objective function output by the optimization algorithm. The idea that flat minima may generalize better than sharp ones is indeed not new [90], and has been explored for neural networks [44, 100]. However, this measure of generalization has limits, as explained in [53]. A notable limit for ReLU networks is that one can use rescalings to change the parameters of the network without affecting the function it implements (thus not changing the generalization performance). Such rescalings can arbitrarily make a minimum of the objective function sharper or wider.

As mentioned in Section 1.1.6, for ReLU networks, the number and the density of affine regions have also been proposed as measures of complexity for ReLU networks. In contrast to norm-based complexity measures, there is no direct way to obtain these measures, but different methods have been proposed to compute these measures with some efficiency [125, 150, 86].

1.4.5 Implicit regularization during optimization

Since the goal of complexity measures is to show that some functions implemented by neural networks have low complexity, and as a consequence have good generalization properties, part of the study should focus on the optimization process, and notably why the optimization algorithms used in practice are biased towards these low-complexity functions. Since this bias exists even without explicitly implementing a regularization term in the objective function, such a phenomenon is studied under the name of implicit bias or implicit regularization.

Implicit regularization is a well understood phenomenon for linear networks and matrix factorization. Existing results show indeed that the optimization implicitly constrains the rank of the prediction matrix [9, 155, 167, 73, 74, 2].

Optimization is less well-understood in the case of non-linear neural networks, such as ReLU networks [10]. The fact that optimization is biased towards low-rank parameters seems less clear for non-linear networks than for linear networks [185].

Some articles show that the optimization process tends to minimize some norm-based quantities [136, 30]. In particular, a line of work studies gradient flow and gradient descent for neural networks in classification, showing that although the logistic loss makes the parameters tend to infinity, the parameters converge in direction, towards max-margin classifiers for some norms [48, 112, 94].

Following other hypotheses, the authors of [151] use Fourier analysis to show that ReLU neural networks are biased towards learning low-frequency functions, and in another fashion the authors of [161] argue that neural networks are biased towards minimization of the number of affine regions.

The properties coming from the stochastic nature of stochastic gradient descent SGD, with respect to gradient descent or to gradient flow, have also drawn interest. It has been shown that stochastic gradient descent introduces noise in the optimization process, compared to classical gradient descent or to gradient flow. In particular, the smaller the batch size, and the bigger the optimization step, the more noise there is. This would explain why neural networks tend to converge to flat minima, as the noise tends to make the optimizer escape of sharp minima [44, 100].

Some authors have tried to explicit the implicit bias of SGD. In particular [172, 14, 71] have shown that under some hypotheses, on average, following stochastic gradient descent on a loss L is equivalent to following a modified gradient flow, including an additionnal bias term taking the form of the squared norm of the gradient of L.

1.4.6 Other work on generalization of neural networks

In this paragraph, we briefly describe some other lines of research around the generalization behavior of neural networks worth mentioning.

First, some authors have tried to better capture the fact that neural networks that perfectly fit noisy data are able to generalize. Indeed, exactly fitting the training data, especially when the data is noisy, is classically considered as overfitting and normally avoided in classical machine learning. These authors have thus tried to understand the settings in which overfitting does not harm prediction, a situation called benign overfitting [18].

Other authors have tried to reconcile the classical machine learning paradigm with modern Deep Learning. In particular, studying the risk as a function of the number of parameters, they have shown the existence of two regimes. In the first regime, when the number of parameters is smaller than the size of the data, the curve has a 'U' shape: when there are few parameters, the bias is predominant. Adding parameters improves the approximation abilities and thus allows the risk to decrease. Then, at some point, the variance takes over and adding parameters makes the risk increase. This is the classical bias-variance trade-off. However, this behavior changes when the number of parameters reaches the size of the training set. After this point (sometimes called interpolation), adding parameters makes the risk decrease again. This particular form of the risk curve has been named 'double descent', and has been explored theoretically and observed empirically [23, 131].

1.4.7 Our contribution: local geometric complexity measures (Chapter 5)

We present here one contribution of this thesis to the topic of complexity and implicit regularization of deep ReLU networks, which corresponds to Chapter 5.

The chapter investigates properties and computational aspects of local complexity measures of deep ReLU neural networks, recently introduced in [82]. The considered complexity measures are linked to the local geometry of the *image set* as defined by $\{f_\theta(X) \mid \theta \text{ varies}\}$ and the *pre-image set* $\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$, where $f_\theta(X)$ is the prediction, for an input sample X, made by the neural network of parameter θ.

The local geometry of the pre-image set and of the image set are linked through the differential $\mathbf{D}f_\theta(X)$ of $\theta \mapsto f_\theta(X)$, and through its rank. The pre-image set represents the redundancies in the parameters of a network. Intuitively, the more there are redundancies in the parameters, the less the space of function represented by the network is rich and complex. Amongst other properties, we notably try to understand how these objects behave during optimization.

The investigation done in this chapter is directly linked to the question of identifiability. Studying identifiability from a finite sample X exactly corresponds to studying the pre-image set. As we know, this set will at least contain the equivalence class of θ modulo permutation and positive rescalings. Identifiability modulo permutation and positive rescaling holds if it only contains this equivalence class. In contrast, the bigger the pre-image set, the more the redundancies in the parameters and the farthest we are from identifiability.

It is thus not surprising that the differential $\mathbf{D}f_\theta(X)$ appears in a slightly different form in Chapter 4, Section 4.4, in the form of the operator $\Gamma(X, \theta)$, and that the rank of this differential also appears in Chapter 4 in the form of the quantity R_Γ.

Chapitre 2

Introduction en français

2.1 Introduction au deep learning

À l'ère de l'apprentissage automatique, le domaine du deep learning s'est imposé comme une pierre angulaire du progrès technologique, révolutionnant notre capacité à traiter, comprendre et extraire des informations précieuses d'ensembles de données vastes et complexes. Les réseaux de neurones profonds, avec leurs architectures multicouches, ont joué un rôle central dans cette transformation, démontrant des capacités sans précédent dans diverses applications, allant de la reconnaissance d'images et de la parole à la au traitement du langage. Alors que nous continuons d'assister à la prolifération de ces puissantes machines d'apprentissage, des questions cruciales concernant leur fonctionnement interne et leur comportement se posent. Cette thèse de doctorat entreprend un voyage dans le monde de la théorie de l'apprentissage profond, en se concentrant spécifiquement sur l'identifiabilité des paramètres des réseaux de neurones - un aspect qui a suscité peu d'attention malgré son importance potentielle. La motivation de cette étude réside dans la résolution de problèmes critiques liés à la préservation de la vie privée, à la robustesse et aux capacités de généralisation. À une époque où la confidentialité des données est primordiale, il est essentiel de comprendre dans quelle mesure les paramètres des réseaux de neurones peuvent être déduits des données observées pour protéger les informations sensibles.

En outre, une compréhension approfondie de l'identifiabilité des paramètres des réseaux de neurones peut contribuer à l'amélioration de la robustesse de ces modèles face aux attaques adversariales et l'amélioration de leur performance de généralisation dans divers scénarios du monde réel. Reconnaissant l'importance éthique et pratique de la protection de la confidentialité des données, du renforcement de la résilience des modèles et de l'amélioration des performances de généralisation, cette recherche vise à contribuer modestement à notre compréhension de ces questions.

Cette introduction est structurée comme suit : dans la section 2.1, nous rappelons quelques bases du machine learning, du deep learning et de la géométrie ReLU, afin d'établir quelques notations et de donner un contexte à la thèse. Dans la section 2.2, nous discutons les symétries que possèdent les paramètres des réseaux de neurones et la relation entre une fonction d'activation et les symétries d'un réseau correspondant. Dans la section 2.3, nous présentons la notion d'identifiabilité et nous formalisons ce qu'elle signifie pour les réseaux de neurones ReLU ainsi que ses motivations. Enfin, dans la section 2.4, nous donnons un aperçu de la question de la complexité des réseaux de neurones et de son lien avec la question de la généralisation.

2.1.1 Apprentissage supervisé, classification, régression

Dans cette section et les suivantes, nous rappelons succinctement quelques principes de base du machine learning. L'objectif n'est pas d'être exhaustif mais simplement de donner un cadre à notre étude des réseaux de neurones.

Dans le problème standard de l'apprentissage supervisé, nous considérons deux variables aléatoires : une variable d'entrée X et une cible Y. L'objectif est, compte tenu de la connaissance de X, de prédire le comportement de Y.

Nous considérerons deux contextes d'apprentissage classiques : la régression et la classification. Dans le cadre de la régression, nous observons $X \in \mathbb{R}^d$, pour un entier $d \in \mathbb{N}^*$ donné, et nous essayons de prédire une valeur réelle ou un vecteur de valeurs réelles : nous avons $Y \in \mathbb{R}^m$, où $m \in \mathbb{N}^*$. La relation entre X et Y n'est pas nécessairement déterministe. Dans ce cas, il est impossible de prédire exactement Y en fonction de X. Plutôt que de prédire la valeur exacte de Y, l'objectif est généralement de prédire $\mathbb{E}[Y|X]$.

En classification, nous considérons un ensemble fini de K classes $C_1, \dots, C_K$, pour $K \in \mathbb{N}^*$. Nous observons $X \in \mathbb{R}^d$ et nous voulons prédire la classe de X, représentée par la variable $Y \in \{C_1, \dots, C_K\}$. Encore une fois, la relation entre X et Y n'est pas nécessairement déterministe, et nous voudrons généralement obtenir une estimation de la probabilité $\mathbb{P}(Y \in C_i \mid X)$, ou au moins un score pour chaque classe C_i, de sorte que pour tous $i \in [\![1, K]\!]$ plus le score que nous attribuons à la classe C_i est élevé, plus la probabilité $\mathbb{P}(Y \in C_i \mid X)$ est élevée.

Dans les deux cas, nous voulons trouver une fonction qui prend $X \in \mathbb{R}^d$ comme entrée et produit un vecteur réel (comme le vecteur prédit $\mathbb{E}[Y|X] \in \mathbb{R}^m$ en régression, ou le vecteur de probabilités prédites $\mathbb{P}(Y \in C_i \mid X)_{i \in [\![1,K]\!]}) \in [0,1]^K$ ou vecteur de score $\lambda \in \mathbb{R}^K$ en classification). Pour ce faire, nous disposons d'une famille de fonctions $(f_\theta)_{\theta \in \mathcal{P}}$, où $\mathcal{P}$ désigne un ensemble de paramètres donné. La plupart du temps, pour nous, $\mathcal{P}$ correspondra à l'espace $\mathbb{R}^p$, pour un certain $p \in \mathbb{N}^*$. L'objectif est de trouver le paramètre θ tel que la fonction f_θ corresponde le mieux au comportement attendu.

2.1.2 Vrai risque, risque empirique

Pour mesurer l'efficacité d'une fonction donnée f_θ dans notre tâche, nous considérons une *fonction de perte* (également appelée fonction de coût ou fonction d'erreur) $\ell : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+$. La fonction de perte permet de comparer notre prédiction $f_\theta(X)$ avec la vraie Y : plus la valeur de la perte est élevée, plus nous considérons que $f_\theta(X)$ est éloigné de Y. En général, la seule façon de ramener $\ell(f_\theta(X), Y)$ à zéro est de prédire la valeur exacte $f_\theta(X) = Y$. Un exemple typique de perte en régression est la perte quadratique : $\ell(y, y') = \sum_{i=1}^m (y_i - y'_i)^2$.

Notre objectif sera de choisir la fonction f_θ afin de minimiser la perte en moyenne sur la distribution de (X, Y). Nous définissons donc le risque, également appelé risque en population, comme suit

$$R(\theta) = [\ell(f_{theta}(X), Y)]\,.$$

Bien que notre objectif soit de minimiser $R(\theta)$, nous n'avons généralement pas accès à la véritable distribution de (X, Y), ce qui rend impossible le calcul de $R(\theta)$ pour un paramètre donné θ. En revanche, si l'on nous donne n exemples $x^{(1)}, \dots, x^{(n)}$ qui ont été échantillonnés à partir de (X, Y), pour un entier n donné, nous pouvons considérer le risque empirique

$$\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell \left(f_\theta(x^{(i)}), y^{(i)} \right),$$

qui est la perte moyenne sur les n exemples plutôt que sur l'ensemble de la distribution, et que nous sommes en mesure de calculer.

Trouver des conditions garantissant que le risque empirique $\hat{R}(\theta)$ est proche du vrai risque $R(\theta)$ est un vaste sujet dans le domaine du machine learning. Nous ne nous étendrons pas sur le sujet ici, mais une discussion plus approfondie peut être trouvée dans [190, 21]. Voir également la section 2.4 pour une discussion sur le sujet dans le cas des réseaux de neurones.

2.1.3 Les réseaux de neurones : première introduction

Un réseau de neurones est une famille de fonctions $(f_\theta)_{\theta \in \mathbb{R}^p}$, telle que chaque fonction $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$ est construite comme une succession de couches. En effet, on peut écrire la fonction f_θ comme une succession de compositions entre $L \geq 2$ fonctions plus élémentaires :

$$f_\theta = h_L \circ h_{L-1} \circ \cdots \circ h_1,$$

où pour tout $l \in [\![1, L]\!]$, la fonction h_l correspond à une couche du réseau, la couche l. La fonction h_l réalise un mapping entre deux espaces vectoriels $\mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$. Ici, $\mathbb{R}^{N_0}$ et $\mathbb{R}^{N_L}$ sont respectivement l'espace d'entrée et l'espace de sortie de f_θ ; nous avons donc $N_0 = d$ et $N_L = m$.

Typiquement, et comme ce sera le cas dans ce travail, une couche est composée d'une fonction affine $x \mapsto W^l x + b^l$, avec $W^l \in \mathbb{R}^{N_l \times N_{l-1}}$ et $b^l \in \mathbb{R}^{N_l}$, suivie d'une fonction d'activation $\sigma_l : \mathbb{R}^l \to \mathbb{R}^l$. Mathématiquement, cela s'écrit donc

$$\forall x \in \mathbb{R}^{N_{l-1}}, \quad h_l(x) = \sigma_l \left(W^l x + b^l \right).$$

La couche de sortie est une exception, car h_L n'est composé que d'une application linéaire, c'est-à-dire

$$\forall x \in \mathbb{R}^{N_{L-1}}, \quad h_L(x) = W^L x + b^L.$$

Par abus de langage, la "couche l" désigne parfois l'espace $\mathbb{R}^{N_l}$, et parfois la fonction $h_l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$ ainsi que ses poids et biais (W^l, b^l).

La fonction implémentée par le réseau de neurones dépend de ses poids et de ses biais. L'ensemble des poids et des biais forme la paramétrisation du réseau

$$\theta = (W^1, \dots, W^L, b^1, \dots, b^L) \in \mathbb{R}^p,$$

avec $p = N_0N_1 + \dots N_{L-1}N_L + N_1 + \cdots + N_L$. On désigne parfois la paramétrisation θ par le terme 'paramètre'comme il est courant de le faire dans la littérature, bien qu'il s'agisse en fait d'un vecteur de p paramètres. Lorsque le mot paramètre est employé, le contexte devrait clarifier si on fait référence à θ ou à un des coefficients de θ. En particulier, l'expression 'nombre de paramètres', souvent employée dans la littérature et dans cette thèse, désigne le nombre p.

Dans de nombreux contextes, et dans tous les contextes considérés dans cette thèse, pour tout $l \in [\![1, L-1]\!]$, l'activation σ_l correspond à la même activation à valeur réelle σ appliquée composante par composante, c'est-à-dire

$$\forall (x_1, \dots, x_{N_l}) \in \mathbb{R}^{N_l}, \quad \sigma_l(x_1, \dots, x_{N_l}) = (\sigma(x_1), \dots, \sigma(x_{N_l})).$$

Il existe plusieurs fonctions d'activation classiques σ : sigmoïde, tangente hyperbolique (tanh), Rectified Linear Unit (ReLU)... Dans cette thèse, nous nous concentrons sur la fonction d'activation ReLU : elle est définie comme $\sigma(t) = \max(t, 0)$ pour tout $t \in \mathbb{R}$. Par défaut, σ fera référence à la ReLU. Dans le cas contraire, ce sera explicité.

2.1.4 Les réseaux de neurones : représentation en graphes

Dans cette section, nous présentons un formalisme équivalent à celui de la section 2.1.3 aux réseaux de neurones, qui leur donne leur nom. Il s'agit d'une présentation des réseaux de neurones sous forme de graphes entre des nœuds appelés *neurones*.

Nous commençons par considérer un ensemble de neurones V. Cet ensemble de neurones est divisé en $L+1$ *couches*, avec $L \geq 2$: $V = \bigcup_{l=0}^{L} V_l$. La couche V_0 est la *couche d'entrée*, V_L est la *couche de sortie* et les couches V_l avec $1 \leq l \leq L-1$ sont les *couches cachées*. Nous notons, pour tout $l \in [\![0, L]\!]$, $N_l = |V_l|$ la taille de la couche V_l.

Les neurones des couches consécutives sont connectés par des arêtes orientées : pour tout $l \in [\![0, L-1]\!]$, si nous considérons deux neurones $v \in V_l$ et $v' \in V_{l+1}$, alors nous désignons par $v \to v'$ l'arête orientée de v vers v'. Nous notons E l'ensemble de toutes ces arêtes orientées :

$$E = \{v \to v' \mid \ v \in V_l, v' \in V_{l+1}, \text{pour } l \in [\![0, L-1]\!]\}.$$

Chaque arête $v \to v' \in E$ du réseau est paramétrée par un *poids* $w_{v \to v'} \in \mathbb{R}$. De plus, on note

$$B = \bigcup_{l=1}^{L} V_l$$

l'ensemble de tous les neurones sauf les neurones d'entrée. Chaque neurone $v \in B$ est paramétré par un *biais* $b_v \in \mathbb{R}$. Un réseau est paramétré par tous ses poids et biais, regroupés dans la paramétrisation θ, avec

$$\theta = ((w_{v \to v'})_{v \to v' \in E}, (b_v)_{v \in B}) \quad \in \mathbb{R}^E \times \mathbb{R}^B \simeq \mathbb{R}^p,$$

où $p = |E| + |B| = N_0N_1 + \dots N_{L-1}N_L + N_1 + \cdots + N_L$ est le nombre de paramètres.

Comme mentionné dans la section précédente, la fonction d'activation, notée σ, est le plus souvent ReLU dans notre cas ; sinon, ce sera toujours spécifié explicitement. Elle est définie comme $\sigma(t) = \max(t, 0)$ pour tout $t \in \mathbb{R}$.

Prédiction du réseau Pour un θ donné, nous définissons de manière récursive $f_\theta^l : \mathbb{R}^{V_0} \to \mathbb{R}^{V_l}$, pour $l \in [\![0, L]\!]$ et $x \in \mathbb{R}^{V_0}$, par

$$\begin{cases} (f_\theta^0(x))_v = x_v, & \text{pour } v \in V_0, \\ (f_\theta^l(x))_v = \sigma\left(\sum_{v' \in V_{l-1}} w_{v' \to v} (f_\theta^{l-1}(x))_{v'} + b_v\right), & \text{pour } v \in V_l, \text{ lorsque } l \in [\![1, L-1]\!], \\ (f_\theta^L(x))_v = \sum_{v' \in V_{L-1}} w_{v' \to v} (f_\theta^{L-1}(x))_{v'} + b_v, & \text{pour } v \in V_L. \end{cases} \tag{2.1.1}$$

Nous définissons la fonction $f_\theta : \mathbb{R}^{V_0} \to \mathbb{R}^{V_L}$ implémentée par le réseau de paramètres θ comme $f_\theta = f_\theta^L$. Nous l'appelons parfois la prédiction ou l'inférence.

Équivalence entre les formalismes Les deux formalismes présentés dans les sections 2.1.3 et 2.1.4 sont équivalents : en numérotant simplement les réseaux de neurones dans chaque couche, nous obtenons une bijection $V_l \to [\![1, N_l]\!]$, ce qui donne un isomorphisme $\mathbb{R}^{V_l} \to \mathbb{R}^{N_l}$. En utilisant ces isomorphismes, on peut regrouper les poids entre deux couches $(w_{v \to v'})_{(v,v') \in V_{l-1} \times V_l}$ dans une matrice W^l et les biais $(b_v)_{v \in V_l}$ d'une couche dans un vecteur b^l, ce qui permet de passer d'une notation à l'autre.

2.1.5 Optimisation

Dans cette section, nous fournissons une explication très succincte de la descente de gradient et de la descente de gradient stochastique, qui sont essentielles pour l'optimisation des réseaux de neurones. Pour une vue d'ensemble plus détaillée des méthodes de gradient, voir [160]. Comme évoqué dans la Section 2.1.1, l'objectif de l'apprentissage est de trouver un paramètre θ tel que la fonction correspondante f_θ s'adapte au mieux à un comportement souhaité. Pour les réseaux de neurones, les paramètres sont des vecteurs réels $\theta \in \mathbb{R}^p$, et la méthode courante pour rechercher le meilleur paramètre consiste à utiliser des méthodes d'optimisation basées sur le gradient.

Pour ce faire, une fonction positive différentiable $L : \mathbb{R}^p \longrightarrow \mathbb{R}_+$ est construite. L prend les paramètres du réseau en entrée et renvoie des valeurs positives. L'objectif est de la minimiser, c'est-à-dire de trouver un paramètre θ tel que $L(\theta)$ soit aussi petit que possible (l'idéal étant de trouver le minimum global de L). L est appelée la fonction objectif, ou parfois la loss (mais il ne faut pas la confondre avec la perte ℓ de la Section 2.1.2).

La descente de gradient est une famille très classique de techniques d'optimisation qui fonctionne approximativement comme suit : on choisit un paramètre initial θ_0. Ensuite, le paramètre est modifié de manière progressive, construisant progressivement une suite de paramètres $(\theta_t)_{t \in \mathbb{N}}$. À l'étape t, le paramètre actuel θ_t est modifié suivant la direction de plus forte descente de L au point θ_t, c'est-à-dire de

manière à ce que le pas $\theta_{t+1} - \theta_t$ soit proportionnel à l'opposé du gradient $\nabla L(\theta_t)$. Un telle technique nécessite de calculer le gradient de L à chaque point θ_t. Cela peut être fait avec une expression explicite du gradient lorsque c'est possible, ou avec des méthodes numériques.

Comme mentionné dans la Section 2.1.2, les fonctions de perte ℓ sont utilisées pour construire un risque $R(\theta) = \mathbb{E}\left[\ell(f_\theta(X), Y)\right]$, que nous aimerions minimiser. En pratique, ce que nous sommes en mesure de calculer et ce que nous considérons donc à la place est le risque empirique $\widehat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(f_\theta(x^{(i)}), y^{(i)}\right)$, où $(x^{(i)}, y^{(i)})_{i\in[\![1,n]\!]}$ est un échantillon d'apprentissage obtenu à partir de (X, Y). Il est donc naturel d'utiliser le risque empirique comme fonction objectif : $L(\theta) = \widehat{R}(\theta)$. Entraîner un réseau de neurones en utilisant le risque empirique comme fonction objectif est courant et appelé *minimisation du risque empirique*. On peut également ajouter un terme de régularisation $P(\theta)$ représentant des contraintes supplémentaires que nous voulons que le paramètre final satisfasse, ce qui conduit à une fonction objectif de la forme $L(\theta) = \widehat{R}(\theta) + P(\theta)$.

En pratique, l'objectif L est minimisé en utilisant une variante de la descente de gradient classique, appelée *descente de gradient stochastique*. Au lieu de calculer le gradient du risque empirique pour l'ensemble complet d'apprentissage, nous subdivisons de manière aléatoire l'ensemble d'apprentissage en k mini-batches $I_1, \ldots, I_k$ de taille n_b, de sorte que $\bigcup_{j=1}^{k} I_k = [\![1, n]\!]$, et chaque étape de gradient est effectuée pour un mini-batch. À une étape donnée, nous considérons donc le gradient de la quantité suivante :

$$\frac{1}{n_b}\sum_{i\in I_k} \ell\left(f_\theta(x^{(i)}), y^{(i)}\right).$$

Il a été montré qu'un tel algorithme est plus efficace du point de vue computationnel (car il est plus facile de calculer le gradient pour un mini-batch que pour l'ensemble complet d'apprentissage) et permet toujours d'obtenir de bonnes performances [29, 69].

Pour calculer le risque empirique, que ce soit pour l'ensemble complet d'apprentissage ou pour un sous-ensemble de l'ensemble d'apprentissage, il est nécessaire de pouvoir calculer les gradients individuels des quantités $\ell\left(f_\theta(x^{(i)}), y^{(i)}\right)$ par rapport à θ, pour chaque exemple $x^{(i)}$. À condition de choisir ℓ de manière appropriée (elle doit au moins être différentiable), les réseaux de neurones sont équipés d'une manière efficace de calculer ces gradients, appelée *backpropagation* ou rétro-propagation en français, que nous n'approfondissons pas ici.

2.1.6 La géométrie affine par morceaux de ReLU

Dans cette section, nous approfondissons un peu les propriétés géométriques des réseaux de neurones avec fonction d'activation ReLU. Les fonctions implémentées par les réseaux de neurones héritent des propriétés de la fonction d'activation qu'ils utilisent. ReLU est une fonction continue et affine par morceaux :

$$\sigma : t \mapsto \max(0, t).$$

Par conséquent, une couche d'un réseau de neurones, qui combine une fonction affine $x \mapsto Wx + b$ et l'activation ReLU, implémente une fonction continue affine par morceaux. Étant donné que la composition de fonctions continues affines par morceaux est continue affine par morceaux, la fonction implémentée par un réseau ReLU est elle-même continue affine par morceaux (parfois également appelée, bien que de manière impropre, linéaire par morceaux).

Les régions affines des réseaux ReLU En tant que fonctions continues affines par morceaux, les réseaux ReLU divisent leur espace d'entrée en régions polyhédrales où la fonction implémentée par le réseau coïncide avec une fonction affine. Ces régions sont appelées les *régions affines*, ou parfois régions linéaires, du réseau. Les régions affines sont polyhédrales et leurs frontières sont composées de morceaux d'hyperplans.

Cette propriété des réseaux ReLU permet de les voir comme des approximateurs, combinant des blocs simples (affines) pour représenter des fonctions plus complexes. Il est intuitif de penser que plus la fonction que nous essayons d'approximer est complexe, plus il y aura besoin de blocs. De plus, il est intuitif de penser que ces blocs se concentreront dans les zones plus complexes de la fonction, avec plus de courbure, plus d'irrégularités, tandis que les zones où la fonction est moins complexe auront besoin de moins de blocs. Suivant cette intuition, le nombre et la densité des régions affines ont été considérés comme des mesures de complexité pour les réseaux ReLU [125, 150, 86]. Pour conclure, le nombre total de régions affines est toujours fini. La raison en est qu'il existe un nombre fini de patterns d'activation (voir le paragraphe sur les patterns d'activation ci-dessous).

Les hyperplans séparateurs sont cruciaux pour l'identifiabilité Deux régions affines adjacentes sont séparées par un hyperplan. Il s'avère que ces hyperplans séparateurs sont très informatifs sur les paramètres d'un réseau. Pour comprendre cela, considérons un réseau constitué d'un seul neurone caché, $L = 2$, $N_0 = d$, $N_1 = N_2 = 1$, de la forme

$$f_\theta(x) = a\sigma(w^T x + b) + c,$$

où $x \in \mathbb{R}^d, a, b, c \in \mathbb{R}$. Par définition de σ, nous avons $f_\theta(x) = a\left(w^T x + b\right) + c$ si $w^T x + b \geq 0$ et $f_\theta(x) = c$ sinon. Il y a donc deux régions affines, séparées par l'hyperplan d'équation $w^T x + b = 0$. Si $w'^T x + b'$ est une autre équation définissant le même hyperplan, alors il existe $\alpha \neq 0$ tel que $(w', b') = \alpha(w, b)$. Ainsi, identifier l'hyperplan séparant les deux régions linéaires définies par un neurone permet d'identifier les poids et le biais du neurone caché, à un facteur près. Cette propriété clé est au cœur de la plupart des travaux sur l'identifiabilité ou la reconstruction stable des paramètres des réseaux ReLU (voir la Section 2.3.5 pour les travaux connexes).

Patterns d'activation Pour tout $l \in [\![1, L-1]\!]$, nous définissons maintenant $s^l(x, \theta) \in \{0, 1\}^{N_l}$ comme suit :

$$\forall i \in [\![1, N_l]\!], \quad s_i^l(x,\theta) = \begin{cases} 1 & \text{si } \sum_{j=1}^{N_{l-1}} W_{i,j}^l \left(f_\theta^{l-1}(x)\right)_j + b_i^l \geq 0 \\ 0 & \text{sinon.} \end{cases}$$

Pour un paramètre fixe $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, et une entrée donnée $x \in \mathbb{R}^{N_0}$, la liste

$$\left(s^1(x,\theta), \ldots, s^{L-1}(x,\theta)\right) \in \{0,1\}^{N_1} \times \cdots \times \{0,1\}^{N_{L-1}}$$

est appelée un *pattern d'activation*.

Nous avons donc

$$\left(f_\theta^l(x)\right)_i = \sigma\left(\sum_{j=1}^{N_{l-1}} W_{i,j}^l \left(f_\theta^{l-1}(x)\right)_j + b_i^l\right) = s_i^l(x,\theta)\left(\sum_{j=1}^{N_{l-1}} W_{i,j}^l \left(f_\theta^{l-1}(x)\right)_j + b_i^l\right),$$

ou écrit différemment

$$\begin{aligned} f_\theta^l(x) &= \mathrm{Diag}(s^l(x,\theta))\left(W^l f_\theta^{l-1}(x) + b^l\right) \\ &= \mathrm{Diag}(s^l(x,\theta)) W^l f_\theta^{l-1}(x) + \mathrm{Diag}(s^l(x,\theta)) b^l. \end{aligned} \tag{2.1.2}$$

Pour un θ donné dans $\mathbb{R}^E \times \mathbb{R}^B$, supposons que nous considérons une région connexe $D \subset \mathbb{R}^{N_0}$ sur laquelle le pattern d'activation est fixe, c'est-à-dire que pour tout $l \in [\![1, L-1]\!]$, il existe $\delta^l \in \{0,1\}^{N_l}$ tel que pour tout $x \in D$, $s^l(x,\theta) = \delta^l$. Alors, (2.1.2) montre que pour tout $l \in [\![1, L-1]\!]$, la relation entre $f_\theta^{l-1}(x)$ et $f_\theta^l(x)$ est affine. Comme la relation entre $f_\theta^{L-1}(x)$ et $f_\theta(x)$ est toujours affine, nous pouvons donc facilement prouver par récurrence sur $l \in [\![1, L]\!]$ que f_θ est affine sur D.

Cela montre qu'il y a une relation entre les patterns d'activation et les régions linéaires. Une région sur laquelle le pattern d'activation est fixe est incluse dans une région affine de f_θ. L'inclusion réciproque n'est pas toujours vraie : une région affine donnée peut avoir plusieurs patterns d'activation. Un exemple trivial est lorsque tous les poids de sortie et les biais du réseau sont nuls. Dans ce cas, la fonction f_θ est constamment égale à zéro, de sorte que tout l'espace d'entrée $\mathbb{R}^d$ est une unique région affine, indépendamment des différents patterns d'activation qui peuvent exister.

Les réseaux ReLU comme représentations universelles des fonctions affines par morceaux Étant donné que les réseaux ReLU représentent des fonctions continues affines par morceaux avec un nombre fini de morceaux, une autre question naturelle concerne la réciproque. Est-ce que n'importe quelle fonction continue affine par morceaux avec un nombre fini de morceaux peut être représentée par un réseau ReLU ? La réponse est positive, comme le montre [7].

2.2 Les réseaux de neurones ont une symétrie intrinsèque

Les réseaux de neurones sont connus pour avoir généralement des symétries, c'est-à-dire des opérations sur les paramètres qui n'affectent pas la fonction qu'ils

implémentent. C'est un aspect fondamental à prendre en compte avant de pouvoir parler d'identifiabilité des paramètres des réseaux de neurones, comme on le fera dans la Section 2.3. De manière générale, la question des symétries et celle de l'identifiabilité sont très liées.

Clarifions tout de suite un point de vocabulaire : plusieurs noms sont utilisés dans la littérature pour désigner le même phénomène. On parle tantôt de symétries, d'invariants, ou d'équivalence de paramètres. En général, dans cette thèse, on parle plutôt d'un paramètre θ qui est équivalent à un paramètre θ'. Cependant, lorsqu'on met l'accent sur les transformations qui font passer d'un paramètre θ à un paramètre équivalent θ', et en particulier lorsqu'on veut parler de la structure de groupe de cet ensemble de transformations, comme c'est le cas dans cette section, on parle plutôt de symétries.

Ces symétries dépendent de l'architecture et des fonctions d'activation utilisées. Dans toute cette section, nous considérons l'architecture des réseaux de neurones et les notations définies dans les sections 2.1.3 et 2.1.4, à l'exception de σ qui n'est pas ReLU mais une fonction d'activation générique. Les travaux dans cette thèse considèrent uniquement des réseaux avec fonction d'activation ReLU, mais le point de vue plus général de cette section nous permettra de discuter de la manière dont les symétries d'un réseau de neurones dépendent de la fonction d'activation choisie. On espère que cela permettra de mieux mettre en valeur le rôle de ReLU dans les symétries possédées par les réseaux considérés dans cette thèse.

2.2.1 La symétrie par permutation

Une symétrie très générale pour les réseaux de neurones est la symétrie par permutation. En effet, pour les réseaux de neurones classiques d'architecture feed-forward fully-connected, dans une couche cachée donnée, aucun rôle particulier n'est attribué à un neurone spécifique : ils sont tous interchangeables. Considérons une architecture de réseau de neurones telle que dans la section 2.1.4, mais avec une fonction d'activation σ quelconque (pas nécessairement ReLU) et un paramètre $\theta \in \mathbb{R}^E \times \mathbb{R}^B$. Considérons une couche cachée V_l, pour $l \in [\![1, L-1]\!]$ et deux neurones $v_1, v_2 \in V_l$. Nous allons échanger le rôle de v_1 et de v_2 dans le réseau, pour montrer qu'ils sont interchangeables.

Pour commencer, supposons que nous échangeons tous les poids entrants de v_1 et v_2 ainsi que leurs biais : pour n'importe quel neurone $v \in V_{l-1}$, nous définissons

$$w'_{v \to v_2} = w_{v \to v_1}$$

et

$$w'_{v \to v_1} = w_{v \to v_2},$$

et de même pour les biais,

$$b'_{v_2} = b_{v_1}$$

et

$$b'_{v_1} = b_{v_2}.$$

Supposons maintenant que nous définissons θ' comme le paramètre obtenu à partir de θ en effectuant les modifications décrites ci-dessus. Soit $x \in \mathbb{R}^{N_0}$. Comme on ne change pas les poids dans les couches précédentes, on a $f_{\theta'}^{l-1}(x) = f_\theta^l(x)$. La fonction σ ici désigne n'importe quelle fonction d'activation. En rappelant (2.1.1), on aurait

$$\begin{aligned} f_{\theta'}^l(x)_{v_1} &= \sigma\left(\sum_{v \in V_{l-1}} w'_{v \to v_1}(f_\theta^{l-1}(x))_v + b'_{v_1}\right) \\ &= \sigma\left(\sum_{v \in V_{l-1}} w_{v \to v_2}(f_\theta^{l-1}(x))_v + b_{v_2}\right) = f_\theta^l(x)_{v_2}, \end{aligned}$$

et

$$\begin{aligned} f_{\theta'}^l(x)_{v_2} &= \sigma\left(\sum_{v \in V_{l-1}} w'_{v \to v_2}(f_\theta^{l-1}(x))_v + b'_{v_2}\right) \\ &= \sigma\left(\sum_{v \in V_{l-1}} w_{v \to v_1}(f_\theta^{l-1}(x))_v + b_{v_1}\right) = f_\theta^l(x)_{v_1}. \end{aligned}$$

On voit que le contenu des neurones v_1 et v_2 est échangé en passant de θ à θ'. Supposons maintenant que l'on échange également les poids sortants des deux neurones de façon correspondante, en définissant, pour chaque neurone $v \in v_{l+1}$:

$$w'_{v_1 \to v} = w_{v_2 \to v}$$

et

$$w'_{v_2 \to v} = w_{v_1 \to v},$$

et supposons que ces changements sont également implémentés dans θ'. Si on considère un neurone $v \in V_{l+1}$, la contribution du neurone v_1 à $f_\theta^{l+1}(x)_v$ sera $w_{v_1 \to v} f_\theta^l(x)_{v_1}$ et la contribution du neurone v_2 sera $w_{v_2 \to v} f_\theta^l(x)_{v_2}$. Ainsi, la contribution du neurone v_1 à $f_{\theta'}^{l+1}(x)_v$ sera $w'_{v_1 \to v} f_{\theta'}^l(x)_{v_1} = w_{v_2 \to v} f_\theta^l(x)_{v_2}$ et la contribution du neurone v_2 sera $w'_{v_2 \to v} f_{\theta'}^l(x)_{v_2} = w_{v_1 \to v} f_\theta^l(x)_{v_1}$. De cette façon, dans le cas de θ', le neurone v reçoit les mêmes informations que dans le cas de θ, bien que les rôles de v_1 et v_2 soient inversés. Il est clair que la contribution du reste des neurones dans V_l est la même pour θ que pour θ'. Cela montre que l'échange des poids (entrants et sortants) et des biais de deux neurones d'une même couche cachée ne modifie pas la fonction implémentée par le réseau. En combinant de telles transpositions, on constate que toute permutation de neurones dans une même couche laisse inchangée la fonction globale implémentée par le réseau.

Cette invariance par permutation découle de la structure des réseaux de neurones, où deux neurones dans une même couche cachée sont interchangeables. En tant que telle, la symétrie par permutation est générique et est partagée par un large éventail d'architectures neuronales.

2.2.2 Symétries dépendant de l'activation

D'autres types de symétries des paramètres des réseaux de neurones sont liés au choix de l'activation en elle-même. Ces symétries reflètent en effet les symétries de la fonction d'activation elle-même. Pour le voir, considérons $\sigma : \mathbb{R} \to \mathbb{R}$ une fonction d'activation arbitraire. Supposons simplement que σ satisfait une relation générique de la forme

$$\forall t \in \mathbb{R}, \qquad \sigma(\lambda t + \mu) = \gamma\sigma(t), \tag{2.2.1}$$

pour certains $\lambda, \gamma \in \mathbb{R}^*, \mu \in \mathbb{R}$ donnés. Ce type de relations tient compte des symétries de nombreuses fonctions d'activation classiques, comme nous le verrons plus tard dans cette section.

En considérant un neurone v_0 dans une couche cachée V_l, supposons que l'on change les poids entrants vers v_0 comme suit :

$$\forall v \in V_{l-1}, \qquad w'_{v\to v_0} = \lambda w_{v\to v_0}$$

et supposons que l'on modifie le biais comme

$$b'_{v_0} = \lambda b_{v_0} + \mu.$$

Supposons maintenant que l'on définit θ' comme le paramètre obtenu à partir de θ en effectuant les changements décrits ci-dessus. Soit $x \in \mathbb{R}^{N_0}$. Comme nous ne changeons pas les poids dans les couches précédentes, nous avons $f^{l-1}_{\theta'}(x) = f^{l-1}_{\theta}(x)$. En rappelant (2.1.1), on a

$$f^l_{\theta'}(x)_{v_0} = \sigma\left(\sum_{v\in V_{l-1}} w'_{v\to v_0} f^{l-1}_\theta(x)_v + b'_{v_0}\right) \tag{2.2.2}$$

$$= \sigma\left(\sum_{v\in V_{l-1}} \lambda w_{v\to v_0} f^{l-1}_\theta(x) + \lambda b_{v_0} + \mu\right) \tag{2.2.3}$$

$$= \gamma\sigma\left(\sum_{v\in V_{l-1}} w_{v\to v_0} f^{l-1}_\theta(x) + b_{v_0}\right) \tag{2.2.4}$$

$$= \gamma f^l_\theta(x)_{v_0}. \tag{2.2.5}$$

Comme nous pouvons le voir, le passage de θ à θ' multiplie le contenu du neurone v_0 par γ. En plus des changements précédents, si on multiplie tous les poids sortants du neurone v_0 par $\frac{1}{\gamma}$, cela va compenser le changement du contenu du neurone v_0 : si on fixe $w'_{v_0\to v} = \frac{1}{\gamma} w_{v_0\to v}$, alors on a

$$w'_{v_0\to v} f_{\theta'}(x)_{v_0} = (\frac{1}{\gamma} w_{v_0\to v})(\gamma f^l_\theta(x)_{v_0}) = w_{v_0\to v} f^l_\theta(x)_{v_0}.$$

En conséquence, le contenu de la couche V_{l+1} reste inchangé, $f^{l+1}_{\theta'}(x) = f^{l+1}_\theta(x)$, et la sortie du réseau est la même : $f_{\theta'}(x) = f_\theta(x)$.

Pour résumer, désignons la transformation que nous venons de réaliser par $\tau^{v_0}_{\lambda,\gamma,\mu} : \mathbb{R}^E \times \mathbb{R}^B \to \mathbb{R}^E \times \mathbb{R}^B$. Pour tout $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, le paramètre transformé $\theta' = \tau^{v_0}_{\lambda,\gamma,\mu}(\theta)$ est défini par :

$$\begin{cases} w'_{v \to v_0} = \lambda w_{v \to v_0} & \text{pour tout } v \in V_{l-1} \\ w'_{v_0 \to v} = \frac{1}{\gamma} w_{v_0 \to v} & \text{pour tout } v \in V_{l+1} \\ b'_{v_0} = \lambda b_{v_0} + \mu & \\ w'_{v \to v'} = w_{v \to v'} & \text{pour tout } (v, v') \in E,\ v, v' \neq v_0 \\ b'_v = b_v & \text{pour tout } v \in B,\ v \neq v_0. \end{cases} \tag{2.2.6}$$

Nous venons de montrer que pour tout $\lambda, \gamma \in \mathbb{R}^*, \mu \in \mathbb{R}$ tels que l'activation σ satisfait la relation (2.2.1), alors pour tout neurone caché $v_0 \in V_l$, $l \in [\![1, L-1]\!]$, la transformation $\tau^{v_0}_{\lambda,\gamma,\mu} : \mathbb{R}^E \times \mathbb{R}^B \to \mathbb{R}^E \times \mathbb{R}^B$ ne modifie pas la fonction implémentée par le réseau. Pour tout $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, pour tout $x \in \mathbb{R}^{N_0}$, nous avons en effet

$$f_{\tau^{v_0}_{\lambda,\gamma,\mu}(\theta)}(x) = f_\theta(x).$$

En appliquant ce résultat générique aux fonctions d'activation classiques, on retrouve des invariants bien connus.

— Pour une fonction d'activation impaire telle que tanh, la relation (2.2.1) est vraie avec $\lambda = \gamma = -1$ et $\mu = 0$. Les transformations invariantes correspondantes $\tau^{v_0}_{-1,-1,0}$ sont les "sign-flips" bien connus qui ont été étudiés dans la littérature [181, 4, 61].

— Pour les fonctions d'activation paires telles que l'activation gaussienne $\sigma(t) = e^{\frac{-t^2}{2}}$, la relation (2.2.1) est vraie avec $\lambda = -1, \gamma = 1$ et $\mu = 0$. Les transformations invariantes correspondantes $\tau^{v_0}_{-1,1,0}$ sont une autre forme de "sign-flips" qui ont également été étudiés dans la littérature [96].

— Pour les fonctions d'activation périodiques, telles que les sinus [171], si T est la période, la relation (2.2.1) est vraie pour $\lambda = \gamma = 1$, $\mu = kT$ pour tout $k \in \mathbb{Z}$. Pour chaque neurone caché v_0, la transformation $\tau^{v_0}_{1,1,kT}$ déplace simplement le biais de k fois la période : $b'_{v_0} = b_{v_0} + kT$.

— Pour les activations polynomiales $\sigma(t) = t^p$, pour $p \in \mathbb{N}^*$, la relation (2.2.1) est vraie pour tout $\lambda \in \mathbb{R}^*$, avec $\gamma = \lambda^p$ et $\mu = 0$. Pour chaque neurone caché v_0, nous avons donc une infinité de transformations invariantes $\tau^{v_0}_{\lambda,\lambda^p,0}$, $\lambda \in \mathbb{R}^*$.

— Pour la fonction d'activation de Heaviside (ou 'échelon'), la relation (2.2.1) est satisfaite pour tout $\lambda > 0$, avec $\gamma = 1$ et $\mu = 0$. De même, pour chaque neurone caché v_0, nous avons donc une infinité de transformations invariantes $\tau^{v_0}_{\lambda,1,0}$, $\lambda \in (0, +\infty)$ [95].

— Enfin, pour les fonctions d'activation affines par morceaux ReLU et leaky-ReLU, la relation (2.2.1) est satisfaite pour tout $\lambda > 0$, avec $\gamma = \lambda$ et $\mu = 0$. Pour chaque neurone caché v_0, nous avons donc une infinité de transformations invariantes $\tau^{v_0}_{\lambda,\lambda,0}$, $\lambda \in (0, +\infty)$. Ces transformations sont bien connues

(voir entre autres [149, 145, 147, 179]), et nous les appelons 'positive rescalings'. Étant donné que nous nous concentrons sur les réseaux ReLU, les positive rescalings nous intéressent particulièrement et sont au cœur d'une partie substantielle de la thèse.

Le but de cette présentation synthétique d'une classe entière de symétries dépendant de l'activation est de montrer qu'il existe une structure commune aux symétries de différentes architectures neuronales, et de montrer qu'il existe un lien direct entre les symétries de la fonction d'activation et les symétries ou redondances des paramètres du réseau. Même si dans cette thèse, nous nous concentrons sur ReLU, l'une des plus utilisées de nos jours, cette structure similaire peut permettre d'envisager le transfert de certains résultats vers d'autres fonctions d'activation.

2.2.3 La structure de groupe des symétries d'un réseau

Les opérations de permutation mentionnées dans la Section 2.2.1 ci-dessus sont des opérations linéaires sur l'espace des paramètres $\mathbb{R}^E \times \mathbb{R}^B$, elles sont inversibles (par la permutation inverse), elles correspondent en fait à des éléments du groupe linéaire $\mathrm{GL}(\mathbb{R}^E \times \mathbb{R}^B)$ (et même, du groupe orthogonal $O(\mathbb{R}^E \times \mathbb{R}^B)$). Dans le cas où $\mu = 0$, les opérations de rescaling $\tau^{v_0}_{\lambda,\gamma,0}$ sont également des opérations linéaires, et on peut vérifier que l'opération inverse de $\tau^{v_0}_{\lambda,\gamma,0}$ est $\tau^{v_0}_{\frac{1}{\lambda},\frac{1}{\gamma},0}$. Nous pouvons donc considérer le sous-groupe G de $\mathrm{GL}(\mathbb{R}^E \times \mathbb{R}^B)$ engendré par toutes les opérations de permutation et de rescaling. Étant donné que G est engendré par des transformations qui ne modifient pas la fonction implémentée par le réseau, il en va de même pour toute transformation de G. L'observation selon laquelle les transformations équivalentes des paramètres des réseaux de neurones sont équipées d'une structure de groupe n'est pas nouvelle [88].

Cela soulève alors certaines questions naturelles, parmi lesquelles :

— Y a-t-il un sous-groupe plus grand G' de $\mathrm{GL}(\mathbb{R}^E \times \mathbb{R}^B)$, $G' \supset G$, qui laisse les fonctions f_θ invariantes pour tout $\theta \in \mathbb{R}^E \times \mathbb{R}^B$?

— On sait que G agit sur les paramètres θ en laissant les fonctions f_θ invariantes. A-t-on la réciproque : si $f_\theta = f_{\theta'}$ existe-t-il $g \in G$ tel que $\theta' = g \cdot \theta$?

— Quelle est la structure de G ?

— Quels sont les orbites des paramètres dans $\mathbb{R}^E \times \mathbb{R}^B$ sous l'action de G ?

— Comment les différentes caractéristiques d'un réseau sont-elles modifiées sous l'action de G ?

— Pouvons-nous exposer un sous-ensemble intéressant de paramètres $H \subset \mathbb{R}^E \times \mathbb{R}^B$ qui représente toutes les orbites, c'est-à-dire $G \cdot H = \mathbb{R}^E \times \mathbb{R}^B$?

Répondre à ces questions de manière générale dépasse largement le cadre de cette thèse. Néanmoins, dans le cas de ReLU, nous pouvons mentionner que les deux premières questions sont directement liées à la question de l'identifiabilité - nous renvoyons à la section correspondante. Par ailleurs, concernant la quatrième question sur la nature des orbites, nous pouvons mentionner que la structure géométrique locale de ces orbites est liée à la question de la structure locale de l'ensemble 'de pré-image'que nous étudions dans le chapitre 5.

Enfin, concernant la troisième question, sans développer une caractérisation complète de G et de sa structure de groupe, nous pouvons au moins écrire ce que sont les éléments de G. Pour ce faire, il est plus facile d'adopter le formalisme de la section 2.1.3. Nous pouvons déduire du Chapitre 3, Section 3.3.3 et notamment de (3.3.6) que si $g \in G$ est une telle transformation, alors il existe

— des matrices de permutation $(P^0, P^1, \dots, P^L) \in \mathbb{R}^{N_0 \times N_0} \times \cdots \times \mathbb{R}^{N_L \times N_L}$, avec $P^0 = \mathrm{Id}_{N_0}$ et $P^L = \mathrm{Id}_{N_L}$,

— des matrices diagonales $(D^0, D^1, \dots, D^L) \in \mathbb{R}^{N_0 \times N_0} \times \cdots \times \mathbb{R}^{N_L \times N_L}$, avec des coefficients diagonaux positifs et telles que $D^0 = \mathrm{Id}_{N_0}$ et $D^L = \mathrm{Id}_{N_L}$,

telles que pour tout $\theta \in \mathbb{R}^p$, si $\tilde{\theta} = g \cdot \theta$, alors pour tous $l \in [\![1, L]\!]$, nous avons

$$\begin{cases} \tilde{W}^l = P^l D^l W^l (D^{l-1})^{-1} (P^{l-1})^{-1} \\ \tilde{b}^l = P^l D^l b^l, \end{cases} \tag{2.2.7}$$

où W^l, b^l désignent les poids et biais associés à θ et $\tilde{W}^l, \tilde{b}^l$ ceux associés à $\tilde{\theta}$.

Par conséquent, choisir un élément de G revient à choisir $L-1$ matrices diagonales avec des coefficients diagonaux positifs et $L-1$ matrices de permutation. Nous pouvons remarquer que G est infini en raison du nombre infini de facteurs de rescaling positifs (les éléments diagonaux des matrices). Cela contraste, par exemple, avec les 'sign-flips' dans le cas de tanh, qui sont en nombre fini.

Une action de groupe définit toujours une relation d'équivalence. Nous pouvons donc définir la relation suivante : pour tout $\theta, \tilde{\theta} \in \mathbb{R}^p$, nous disons que θ et $\tilde{\theta}$ sont **équivalents modulo permutation et rescaling positif**, ou simplement équivalents, et nous écrivons $\theta \sim \tilde{\theta}$, s'il existe $g \in G$ tel que $g \cdot \theta = \tilde{\theta}$, ou autrement dit, si θ et $\tilde{\theta}$ satisfont la relation (2.2.7). Les classes d'équivalence modulo $\sim$ sont les orbites sous l'action de G sur $\mathbb{R}^p$.

2.3 L'identifiabilité : du modèle linéaire aux réseaux de neurones

2.3.1 Introduction à l'identifiabilité : le modèle linéaire en dimension finie

L'identifiabilité est une notion classique en statistiques [143]. En termes généraux, elle signifie que, étant donné un modèle paramétrique $\{P_\theta, \theta \in \mathcal{P}\}$, où pour tout $\theta \in \mathcal{P}$, P_θ est une distribution de probabilité, chaque fois que $\theta_1 \neq \theta_2$, on a $P_{\theta_1} \neq P_{\theta_2}$. Elle est généralement souhaitée car elle est nécessaire pour pouvoir estimer le paramètre θ. Cette forme d'identifiabilité est la forme générale et théorique de l'identifiabilité. Cela signifie que si nous avions une connaissance totale de P_θ, nous pourrions identifier θ.

Comme premier exemple, considérons un modèle linéaire de la forme

$$Y = X\beta + \epsilon,$$

où $X \in \mathbb{R}^{n \times p}$ est déterministe, $\beta \in \mathbb{R}^p$ est le paramètre du modèle, et ϵ est un vecteur aléatoire, suivant une distribution gaussienne centrée. Ici, les lignes de X seront généralement des exemples $x^{(1)}, \ldots, x^{(n)}$. Dans ce modèle, si le rang des colonnes de la matrice X n'est pas plein, alors il existe une infinité de choix pour β qui conduisent à la même distribution pour Y. En particulier, si X a plus de colonnes que de lignes, c'est-à-dire si $n < p$, par définition le modèle ne peut pas être identifiable. Dans le cadre classique de la régression linéaire, nous avons $n > p$. Cependant, dans le cas $n > p$, le manque d'identifiabilité peut encore découler de la redondance de deux variables ou plus. Une manière d'imposer l'identifiabilité est alors de supprimer les variables redondantes, ce qui signifie supprimer les colonnes redondantes de X une par une jusqu'à atteindre l'identifiabilité.

2.3.2 Identifiabilité en grande dimension

2.3.2.1 Le modèle linéaire en grande dimension

Plus récemment, avec le développement des données de grande dimension, le modèle linéaire a été considéré dans un contexte différent, dans lequel la dimension p dépasse largement le nombre n d'exemples dont nous disposons : nous avons $p \gg n$. C'est souvent le cas, par exemple, avec les données génomiques. Comme mentionné dans la section précédente, lorsque $p > n$, il n'y a pas d'identifiabilité. Cependant, nous aimerions toujours trouver le 'vrai' β. Une hypothèse classique qui est alors formulée est la parcimonie : on suppose que β n'a que quelques coefficients non nuls. Cela signifie que seules quelques variables ont réellement un impact sur Y, et trouver lesquelles est une tâche appelée 'sélection de variables'. La parcimonie d'un vecteur peut être mesurée avec la norme ℓ_0, qui compte simplement le nombre d'entrées non nulles :

$$\|\beta\|_0 = |\{j \in [\![1, p]\!] \mid \beta_j \neq 0\}|.$$

2.3.2.2 Compressed sensing

Dans les années 2000, le domaine du compressed sensing a reçu une certaine attention, suite aux travaux pionniers [54, 38]. Le cadre est très similaire au modèle linéaire mentionné précédemment : nous considérons un signal inconnu $x \in \mathbb{R}^p$, auquel nous accédons par le biais d'une série de mesures linéaires, c'est-à-dire une liste de n vecteurs a_i, pour lesquels nous observons la quantité $y_i = \langle a_i, x \rangle$. Si l'on note $A \in \mathbb{R}^{n \times p}$ la matrice dont la $i^{\text{ème}}$ ligne est a_i^T et y le vecteur $(y_1, \ldots, y_n)^T$, cela peut être réécrit comme

$$y = Ax.$$

Il s'agit du même problème inverse linéaire que précédemment, avec A au lieu de X et le vecteur inconnu étant x au lieu de β. Ici, nous ne considérons pas de vecteur de bruit ϵ, bien que cela puisse être ajouté pour prendre en compte les erreurs de mesure. Le cadre du compressed sensing se concentre sur des problèmes pour lesquels le nombre n de mesures est limité, pour lesquels nous avons $n \ll p$. Par conséquent, A ne peut pas être injective et l'identifiabilité n'est pas vérifiée par définition. Pour

obtenir l'identifiabilité, il faut ajouter des contraintes supplémentaires sur x qui réduisent l'ensemble des solutions à un seul élément.

Encore une fois, une hypothèse standard est la parcimonie, et nous pouvons reformuler le problème comme un problème de minimisation de la norme ℓ_0 :

$$\begin{aligned} \min_x \quad & \|x\|_0 \\ \text{s.t.} \quad & y = Ax. \end{aligned} \tag{2.3.1}$$

De manière intéressante, la théorie montre que dans de nombreux cas, ce problème admet un minimiseur unique, ce qui garantit l'identifiabilité dans ce nouveau cadre. Dans ce cas, si l'on note k la valeur minimale de parcimonie, et si le vecteur réel x_0 est k-parsimonieux, résoudre ce problème permet en réalité de récupérer x_0.

Cependant, la norme ℓ_0 est difficile à optimiser. En fait, il a été démontré que le problème (2.3.1) est NP-difficile. Une découverte très intéressante est qu'il est possible d'utiliser la norme ℓ_1 comme substitut convexe de la norme ℓ_0. Le problème devient alors

$$\begin{aligned} \min_x \quad & \|x\|_1 \\ \text{s.t.} \quad & y = Ax. \end{aligned} \tag{2.3.2}$$

Il a été démontré que dans de nombreux contextes, la solution de (2.3.2) est la même que celle de (2.3.1), ce qui signifie que résoudre (2.3.2) permet de récupérer exactement le vecteur k-parsimonieux x_0. Contrairement à (2.3.1), le problème (2.3.2) est pratiquement soluble en temps polynomial avec des outils d'optimisation classiques [101].

2.3.3 L'identifiabilité pour les réseaux de neurones ReLU

2.3.3.1 Plusieurs définitions de l'identifiabilité

La question de l'identifiabilité peut être formulée pour les réseaux de neurones : comme vu précédemment, les réseaux de neurones admettent des paramètres sous forme de poids et de biais, et étant donné un choix de paramètres $\theta \in \mathbb{R}^p$, un réseau implémente une fonction f_θ. La différence avec le modèle linéaire est que f_θ n'est généralement pas linéaire par rapport à θ. Les motivations diffèrent également ; elles sont discutées dans la section suivante.

Considérons un réseau ReLU $(f_\theta)_{\theta \in \mathbb{R}^p}$. De manière naïve, la question générale de l'identifiabilité est : si $f_{\theta_1} = f_{\theta_2}$, a-t-on $\theta_1 = \theta_2$? En fait, comme nous l'avons vu dans la Section 2.2, il est possible de transformer les paramètres d'un réseau ReLU sans modifier la fonction qu'il implémente. En effet, si on rappelle la relation d'équivalence $\sim$ définie à la fin de la Section 2.2.3, alors pour tout $\theta \in \mathbb{R}^p = \mathbb{R}^E \times \mathbb{R}^B$, on sait que si $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$ satisfait $\theta \sim \theta'$, on a $f_\theta = f'_\theta$. Nous devons donc assouplir la définition de l'identifiabilité si nous voulons qu'elle ait un sens pour les réseaux de neurones ReLU.

Definition 6. On dit qu'un paramètre $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ d'un réseau ReLU est **identifiable** si pour tout $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, alors

$$f_\theta = f_{\theta'} \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

Definition 7. On dit qu'un paramètre $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ d'un réseau ReLU est **localement identifiable** s'il existe $\epsilon_\theta > 0$ tel que, pour tout $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, alors

$$\|\theta - \theta'\| < \epsilon_\theta \quad \text{et} \quad f_\theta = f_{\theta'} \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

Ces définitions supposent que les fonctions f_θ et $f_{\theta'}$ implémentées par les réseaux de paramètres θ et θ' sont égales, ce qui signifie qu'elles coïncident sur l'intégralité de l'espace de départ. Cependant, la distribution des entrées peut avoir un support plus petit que l'ensemble complet $\mathbb{R}^{N_0}$, ce qui signifie qu'en pratique, nous ne considérons que $f_\theta(x)$ et $f_{\theta'}(x)$ dans un sous-ensemble $\Omega \subset \mathbb{R}^{N_0}$. Dans ce cas, si f_θ et $f_{\theta'}$ coïncident sur Ω, c'est comme si elles étaient égales pour nous. En tenant compte de cela, nous pouvons considérer la définition suivante.

Definition 8. On dit qu'un paramètre $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ d'un réseau ReLU est **identifiable à partir de** Ω si pour tout $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, alors

$$\forall x \in \Omega,\ f_\theta(x) = f_{\theta'}(x) \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

La même adaptation peut être faite pour l'identifiabilité locale.

Definition 9. On dit qu'un paramètre $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ d'un réseau ReLU est **localement identifiable à partir de** Ω s'il existe $\epsilon_\theta > 0$ tel que, pour tout $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$, alors

$$\|\theta - \theta'\| < \epsilon_\theta \quad \text{et} \quad \forall x \in \Omega,\ f_\theta(x) = f_{\theta'}(x) \qquad \Longrightarrow \qquad \theta \sim \theta'.$$

Un cas particulier est lorsque Ω est fini. Dans ce cas, on a une liste finie d'exemples $x^{(i)}$, $1 \leq i \leq n$ pour un entier n. La question est donc de savoir si f_θ et $f_{\theta'}$ coïncident sur ces n entrées, a-t-on $\theta \sim \theta'$? Cette déclinaison de l'identifiabilité est intéressante car elle correspond aux situations rencontrées en pratique. L'identifiabilité à partir d'un sous-ensemble Ω et en particulier l'identifiabilité à partir d'une liste finie d'entrées sont plus difficiles à garantir car elles ne supposent pas une connaissance complète de f_θ.

2.3.3.2 Stabilité inverse et reconstruction stable

Nous explicitons dans cette section deux notions proches de l'identifiabilité et parfois désignées elles-mêmes comme identifiabilité.

La première notion est la **stabilité inverse**. L'idée générale est d'étendre la définition de l'identifiabilité aux petites perturbations. Supposons que les fonctions f_θ et $f_{\theta'}$ ne soient pas parfaitement égales, mais qu'elles soient proches en un certain sens. Les paramètres θ et θ' ne seront pas égaux, mais on peut se demander s'ils sont proches pour une certaine distance. C'est une extension intéressante de la notion d'identifiabilité, car en pratique, nous sommes généralement confrontés à toutes

sortes d'incertitudes et d'erreurs qui empêchent une égalité parfaite. De plus, la stabilité inverse peut avoir de nombreux avantages théoriques et pratiques, notamment en termes d'optimisation [58]. Voici une définition informelle de la stabilité inverse :

Definition 10. Nous disons que la stabilité inverse est vérifiée s'il existe $\alpha > 0$ tel que, pour tout $\theta, \theta' \in \mathbb{R}^E \times \mathbb{R}^B$, alors

$$\|\theta - \theta'\|_{par} \leq \alpha \|f_\theta - f_{\theta'}\|_{fct}.$$

Dans cette définition informelle, de nombreux éléments doivent être spécifiés. Tout d'abord, il faut déterminer une 'norme' significative (ici, nous mettons des guillemets car nous considérons les normes dans un sens large) $\|\cdot\|_{par}$ sur l'ensemble des paramètres. Une telle 'norme' doit prendre en compte les équivalences des paramètres. En effet, comme nous l'avons dit précédemment, tous les paramètres appartenant à une même classe d'équivalence pour $\sim$ implémentent la même fonction, de sorte que la quantité $\|\theta - \theta'\|_{par}$ ne doit pas dépendre du membre choisi de la classe. Par exemple, toute norme classique sur l'espace des paramètres $\mathbb{R}^E \times \mathbb{R}^B$, telle que la norme euclidienne $\|\cdot\|_2$, échouerait pour les réseaux ReLU car nous pouvons utiliser l'opération de rescalings positifs sur un paramètre θ pour construire une séquence $\theta_n \in \mathbb{R}^E \times \mathbb{R}^B$ telle que, pour tout $n \in \mathbb{N}$, $\theta \sim \theta_n$ mais $\|\theta - \theta_n\|_2 \to \infty$, ce qui empêcherait toute forme de borne comme celle de la définition informelle de tenir. À titre d'exemple, [117] proposent une norme sur l'espace des paramètres pour les problèmes de factorisation matricielle structurée profonde qui tient compte de cela.

Un autre élément qui doit être spécifié est la 'norme' $\|\cdot\|_{fct}$ sur l'espace des fonctions du réseau. Cela est encore plus complexe, car la gamme des normes sur un tel espace fonctionnel est très large. Les auteurs de [58] proposent une discussion sur le sujet et présentent quelques contre-exemples pour lesquels la stabilité inverse n'est pas vérifiée. Ils montrent que les normes qui tiennent compte des gradients, telles que les normes de Sobolev, sont mieux adaptées pour caractériser la distance entre deux fonctions implémentées par un réseau.

Pour finir, contrairement à l'identifiabilité et à la stabilité inverse, la **reconstruction stable** désigne la recherche d'algorithmes capables de reconstruire en pratique le paramètre θ d'un réseau de neurones, ou d'un équivalent, avec une certaine marge d'erreur.

2.3.4 Les motivations de l'identifiabilité

Les motivations pour étudier l'identifiabilité dans le cas des réseaux de neurones ne sont pas les mêmes que dans les contextes statistiques présentés dans les Sections 1.3.1 et 1.3.2. En statistiques, l'une des motivations pour estimer les paramètres d'un modèle est parce qu'ils ont une signification physique. Dans ce cas, l'identifiabilité est nécessaire pour estimer la valeur exacte d'un paramètre particulier. En deep learning, il est possible de trouver dans la littérature des configurations spécifiques où une partie des paramètres d'un réseau de neurones a une signification physique [60, 197],

ce qui peut être une raison de rechercher l'identifiabilité. Cependant, la plupart du temps, les paramètres des réseaux de neurones n'ont pas de signification physique, et les motivations pour étudier l'identifiabilité sont différentes. Nous décrivons certaines des motivations dans la présente section. Elles sont diverses et vont des plus pratiques aux plus théoriques.

2.3.4.1 Les attaques par inversion de modèle

Les modèles de machine learning, et notamment les réseaux de neurones, sont largement utilisés de nos jours dans une grande variété de tâches. Ce développement s'accompagne d'un besoin de garanties en ce qui concerne la sécurité, la robustesse et la confidentialité des modèles utilisés. Il est fréquent que l'utilisateur du modèle ne soit pas le même que le fournisseur [158]. Dans un tel cadre, les utilisateurs peuvent utiliser des modèles déjà entraînés, fournis par des services cloud, soit gratuitement, soit moyennant des frais associés à chaque requête. Les utilisateurs n'ont généralement pas un accès complet au modèle. En particulier, ils ne connaissent pas nécessairement l'architecture utilisée, l'optimiseur ou les hyperparamètres utilisés lors du training, et surtout, quels sont les paramètres obtenus à l'issue du training. Il peut en effet être important pour les fournisseurs de modèles de garder ces informations confidentielles, pour des raisons que nous expliquons ci-dessous. Cependant, les attaques par extraction de modèle ont été un sujet croissant au cours des dernières années [139] : il s'agit d'attaques qui sont capables de récupérer au moins partiellement les informations cachées du modèle à l'aide des requêtes envoyées au modèle. En particulier, pour les réseaux de neurones, certaines attaques sont capables de récupérer en pratique les paramètres d'un réseau ou la fonction implémentée par celui-ci à partir de requêtes envoyées au réseau [39, 159].

Un problème de confidentialité L'extraction des paramètres d'un réseau de neurones par des attaquants peut poser un problème majeur en matière de confidentialité. En effet, il est désormais bien connu que les réseaux de neurones mémorisent certains éléments de leurs bases de données d'entraînement, stockant ces informations dans leurs paramètres. Par exemple, en 2015, les auteurs de [67] ont montré qu'ils étaient capables de reconstruire certaines images de la base de données utilisée pour entraîner un système de reconnaissance faciale. La technique exploite un indicateur de confiance renvoyé par le système. La technique nécessite le calcul du gradient $\nabla f_\theta(x)$ de $\theta \mapsto f_\theta(x)$, pour n'importe quelle entrée x. Comme le montrent les auteurs, cela peut être fait avec des méthodes numériques, mais c'est très coûteux et ne fonctionne que lorsque la dimension est suffisamment petite. Dans la plupart de leurs expériences, les auteurs supposent la connaissance de θ. Plus récemment, les auteurs de [85] ont utilisé le biais implicite des réseaux de neurones (voir Section 2.4.4) pour reconstruire plusieurs images de la base de données d'entraînement d'un réseau de neurones. L'algorithme est différent, mais la méthode suppose à nouveau la connaissance des paramètres entraînés θ. Il devient donc clair qu'empêcher les utilisateurs de récupérer les paramètres du réseau est essentiel pour la confidentia-

lité.

Un problème de robustesse Une autre préoccupation majeure est qu'un réseau dont les paramètres sont accessibles peut être moins robuste face aux attaques adversariales. En effet, il est bien connu que les réseaux de neurones sont vulnérables à de telles attaques, et des travaux considérables ont été réalisés au cours de la dernière décennie pour mieux comprendre ce phénomène et le prévenir. La capacité d'utilisateurs malveillants à tromper un système de vision par ordinateur basé sur un réseau de neurones peut poser des problèmes majeurs de sécurité, par exemple dans le cas des véhicules autonomes et de la reconnaissance des panneaux de signalisation. Avoir accès aux paramètres entraînés du réseau peut faire la différence dans de telles attaques. En effet, si certaines attaques adversariales de type 'boîte noire' existent [180, 165, 49], c'est-à-dire des attaques qui fonctionnent en se contentant d'interroger un réseau, de nombreuses attaques utilisent la connaissance des paramètres du réseau, au moins pour calculer les gradients [183, 79, 104, 142, 41, 127, 126, 13].

Un problème de propriété intellectuelle Enfin, les paramètres d'un réseau de neurones entraîné peuvent avoir de la valeur, car ils peuvent être le résultat d'entraînements coûteux. Pouvoir extraire les paramètres entraînés d'un réseau permet de le reproduire, ce qui peut donc poser un problème en matière de propriété intellectuelle du réseau ou des bases de données d'apprentissage [196]. L'extraction des paramètres peut également aider à extraire d'autres informations précieuses. Par exemple, les auteurs de [191] présentent une attaque d'extraction d'hyperparamètres, qui est capable de récupérer un hyperparamètre utilisé pour entraîner un modèle (qui peut être un réseau de neurones), en résolvant un système d'équations. Cette attaque suppose la connaissance de la paramétrisation θ, donc à nouveau, la capacité d'extraire les paramètres d'un réseau est cruciale.

Le lien avec l'identifiabilité Le problème des attaques par extraction de modèle et en particulier de l'extraction de paramètres est directement lié à la question de l'identifiabilité. En effet, l'étude théorique de quand la fonction implémentée par un réseau caractérise de manière unique ses paramètres, et quelles sont les attributs de la fonction qui rendent les paramètres identifiables, permet de comprendre quels réseaux sont vulnérables à l'extraction de paramètres et comment l'empêcher. Pour les réseaux ReLU, les travaux existants sur l'identifiabilité, y compris les nôtres, montrent comment les frontières entre les régions affines, qui prennent la forme de morceaux d'hyperplans, transmettent le plus d'informations sur les paramètres d'un réseau. En s'appuyant sur cette connaissance, les auteurs de [43] ont développé une méthode pour empêcher l'extraction de paramètres en complexifiant artificiellement le réseau sans changer son comportement global. Cette méthode ajoute des couches au réseau d'origine qui ne modifient pas le comportement global de la fonction, mais qui augmentent significativement le nombre de régions affines et de frontières

de séparation, afin de rendre la récupération des paramètres inextricable.

Une autre manière de tirer parti de la connaissance de l'identifiabilité pour mieux comprendre les attaques d'inversion de modèle consiste à fournir des conditions pratiques permettant de tester l'identifiabilité. En effet, l'utilisateur d'un réseau a accès à une liste d'entrées (les requêtes) et aux sorties correspondantes du réseau (les réponses aux requêtes). La question est alors de savoir si ces informations caractérisent de manière unique les paramètres : il s'agit de la question de l'identifiabilité à partir d'un échantillon fini (voir la Section 2.3.3). Il est clair que si le nombre de requêtes est faible (par exemple une seule requête), les informations obtenues ne suffiront pas à caractériser les paramètres du réseau. Pour le fournisseur, un moyen d'empêcher la récupération des paramètres peut alors être de garantir que l'identifiabilité ne s'applique pas, c'est-à-dire de vérifier qu'une condition nécessaire d'identifiabilité n'est pas satisfaite. Dans ce cas, les paramètres ne sont pas caractérisés de manière unique par les informations disponibles, et sans informations supplémentaires, il est impossible d'inférer les paramètres du réseau. Un moyen simple de garantir cela peut simplement être de limiter le nombre de requêtes accessibles à l'utilisateur. Du côté opposé, s'assurer de l'identifiabilité est intéressant du point de vue d'un attaquant. Si l'attaquant connaît les entrées X, les sorties correspondantes $f_\theta(X)$, et est capable de calculer un $\tilde{\theta}$ tel que $f_{\tilde{\theta}}(X) = f_\theta(X)$, la question devient alors : cela garantit-il que $\tilde{\theta} = \theta$ ou l'attaquant doit-il élargir X avec de nouvelles requêtes ? L'attaquant a besoin d'une condition suffisante d'identifiabilité.

2.3.4.2 Garanties théoriques

Un inconvénient fréquemment mentionné des réseaux de neurones est qu'ils fonctionnent comme des modèles de type 'boîte noire' sur lesquels nous avons peu de compréhension et de contrôle. Ce problème a été la motivation d'un grand nombre de travaux au cours de la dernière décennie, qui ont contribué à éclairer davantage les réseaux de neurones et leur comportement complexe. Malgré ces progrès, le besoin de garanties théoriques et d'une meilleure compréhension du comportement des réseaux de neurones persiste. La question de l'identifiabilité peut jouer un rôle dans cet effort.

Par exemple, une ligne de recherche pour comprendre les propriétés d'entraînement et de généralisation des réseaux de neurones est le cadre 'élève-enseignant', dans lequel on suppose qu'on entraîne un réseau - l'élève - avec des données générées par un réseau inconnu - l'enseignant. Dans un tel cadre, une question naturelle est de savoir dans quel cas la formation du réseau élève implique réellement la récupération des paramètres du réseau enseignant [62]. Dans ce cas, l'apprentissage peut être considérée comme une tâche d'estimation classique : retrouver les vrais paramètres du modèle. En particulier, si les paramètres du réseau enseignant sont identifiables à partir des données d'apprentissage, alors l'entraînement de l'élève pour coller parfaitement aux exemples d'apprentissage est rigoureusement équivalent à la récupération des paramètres de l'enseignant. De nombreux articles ont suivi ce cadre, voir par exemple [93, 32, 109, 174, 200].

Dans ce cadre, une conséquence de l'identifiabilité à partir des données d'entraînement est que le minimiseur global du risque empirique est unique. Par conséquent, à condition que le processus d'optimisation soit capable d'atteindre le minimiseur global, il n'y a pas de variabilité des paramètres appris venant des réglages de l'optimisation (choix de l'algorithme, taille du pas, nombre d'epochs...) ou de la stochasticité (pour les optimiseurs stochastiques). Cela garantit davantage de contrôle sur l'entraînement des réseaux de neurones, sous la forme de ce que nous pourrions appeler la reproductibilité du processus d'entraînement. Même si des travaux récents sur les phénomènes de double descente, par exemple [24], mettent en évidence un avantage du surparamétrage (dans laquelle nous ne pourrons généralement pas garantir l'identifiabilité) pour améliorer les performances de prédiction, un utilisateur peut être intéressé par le fait d'avoir un nombre de paramètres suffisamment petit pour conserver l'identifiabilité, si la perte de performance est limitée par rapport au cadre surparamétré.

Le lien entre l'espace des paramètres et l'espace des fonctions implémentées par un réseau de neurones est complexe. Comme le montre [144], l'ensemble des fonctions $(f_\theta)_{\theta\in\mathbb{R}^p}$ implémentées par un réseau de neurones avec des fonctions d'activation classiques (y compris ReLU) n'est ni globalement ni localement convexe. De plus, cet ensemble n'est pas fermé dans aucun des espaces L^p. Une conséquence de cette non-fermeture est par exemple l'explosion des poids lorsqu'une suite de fonctions f_{θ_n} converge vers une limite f qui n'est pas dans l'espace des fonctions du réseau. Encore pire, les auteurs montrent que la suite f_{θ_n} peut converger vers une fonction f_{θ^*} qui appartient à l'ensemble des fonctions réalisées par le réseau, tandis que la séquence (θ_n) diverge vers l'infini. Cela est dû au fait qu'une fonction implémentée par un réseau peut avoir un nombre infini de paramétrisations.

Pour résoudre les problèmes posés par [144], les auteurs de [58] montrent qu'en choisissant les normes sur l'espace des fonctions de façon adéquate, et en restreignant l'ensemble des paramètres pour éviter les paramétrisations dégénérées, on peut garantir la stabilité inverse. Comme le montrent les auteurs, la stabilité inverse, associée à des contraintes de régularisation appropriées, permet alors de garantir de bonnes propriétés d'optimisation, telles que la quasi-optimalité des minima locaux de la fonction objectif.

2.3.4.3 L'identifiabilité comme mesure de la diversité d'un échantillon

L'identifiabilité peut nous renseigner sur un réseau de neurones, mais elle peut aussi être une information utile pour caractériser la diversité ou la représentativité d'un échantillon donné. Supposons par exemple que nous ayons un réseau de neurones entraîné de paramètre $\theta \in \mathbb{R}^p$ donné, que nous aimerions tester, et pour ce faire, nous observons les sorties $f_\theta(x^{(i)})$ du réseau pour une liste d'entrées $x^{(i)}, i \in [\![1, n]\!]$. Alors, l'identifiabilité de θ à partir de l'échantillon de test peut servir de mesure de la richesse de l'échantillon. Si θ n'est pas identifiable, il n'est pas entièrement caractérisé par l'échantillon, ce qui signifie que l'on pourrait ajouter de nouveaux exemples de test pour mieux caractériser la fonction f_θ implémentée par

le réseau de neurones.

2.3.5 Travaux existants

2.3.5.1 Identifiabilité

L'identifiabilité des paramètres des réseaux de neurones a été le sujet de nombreux travaux. Pour les fonctions d'activation lisses, certains résultats ont déjà été établis dans les années 1990. Pour les réseaux peu profonds, des résultats existent pour des fonctions d'activation telles que tanh [181, 4], la sigmoïde logistique [105], ou les fonctions gaussiennes et rationnelles [96]. Pour les réseaux profonds, [61] montre qu'avec tanh comme fonction d'activation, avec seulement quelques conditions génériques sur les paramètres, deux réseaux qui implémentent la même fonction ont la même architecture et les mêmes paramètres jusqu'à certaines permutations et opérations de changement de signe ('sign-flips').

Dans le cas des réseaux ReLU, nous avons vu dans la Section 2.2 que deux opérations sont bien connues pour préserver la fonction implémentée par le réseau : les permutations et les rescalings positifs. Ces opérations définissent des classes d'équivalence sur l'ensemble des paramètres, et nous pouvons au mieux identifier les paramètres d'un réseau modulo ces équivalences. Il est démontré dans [147] que ces opérations sont les seules opérations génériques de ce type pour les réseaux ReLU avec un nombre décroissant de neurones par couche. En effet, les auteurs montrent que pour n'importe quelle architecture de réseau ReLU fully-connected avec un nombre décroissant de neurones par couche, pour tout ensemble ouvert non vide Ω, il existe une paramétrisation θ telle que pour toute autre paramétrisation $\tilde{\theta}$ satisfaisant une hypothèse générique, si $f_{\tilde{\theta}}$ coïncide avec f_θ sur Ω, alors $\tilde{\theta}$ est dans la classe d'équivalence de θ modulo permutations et rescalings positifs.

Dans le cas des réseaux ReLU peu profonds, [145] établit une condition suffisante sur les paramètres pour l'identifiabilité. Si la condition est satisfaite par deux réseaux de type feedforward fully-connected à deux couches avec des fonctions ReLU, dont les fonctions coïncident sur tout l'espace d'entrée, alors les paramètres d'un réseau peuvent être obtenus à partir des paramètres de l'autre réseau par permutation et rescalings positifs.

Dans le cas des réseaux ReLU profonds, [159] donne une condition suffisante pour être capable de reconstruire l'architecture, les poids et les biais d'un réseau ReLU profond en connaissant sa correspondance entrée-sortie sur tout l'espace d'entrée.

Une autre propriété est l'identifiabilité locale, qui est l'identifiabilité d'un paramètre θ parmi un ensemble de paramètres qui sont proches de θ, telle que définie dans la section 2.3.3.1. [179] étudie cette propriété pour les réseaux peu profonds et profonds. Pour un réseau ReLU profond, il montre d'abord que sous une hypothèse triviale, l'identifiabilité générale modulo permutation et rescalings positifs implique l'identifiabilité locale modulo rescalings positifs, et que la non-existence de "neurones jumeaux" est nécessaire pour l'identifiabilité et l'identifiabilité locale. Ensuite, les auteurs de [179] donnent une condition abstraite nécessaire et suffisante

sur θ telle qu'il existe un ensemble Ω bien choisi et *fini* à partir duquel l'identifiabilité locale est garantie modulo rescalings positifs, et ils donnent une borne sur la taille de l'ensemble.

Enfin, une autre ligne de recherche qui peut être liée à l'identifiabilité est le domaine de la compression sans perte de réseaux de neurones [169, 170].

2.3.5.2 Stabilité inverse and reconstruction stable

Établir des propriétés d'identifiabilité est une première étape vers l'établissement de propriétés de stabilité inverse et l'étude d'algorithmes de reconstruction stable, telle que décrites dans la Section 2.3.3.2.

En général, la stabilité inverse n'est pas vérifiée avec la norme uniforme pour les réseaux de neurones fully-connected feedforward. En effet, [144] montre que, pour n'importe quelle profondeur, pour n'importe quelle architecture avec au moins 3 neurones dans la première couche cachée et n'importe quelle fonction d'activation couramment utilisée, il existe une séquence de réseaux dont la fonction tend uniformément vers 0 tandis que n'importe quelle paramétrisation de ce réseau tend vers l'infini.

De nombreux résultats sur la stabilité inverse et la reconstruction stable existent déjà pour les réseaux peu profonds. [58] étudie directement la stabilité inverse jusqu'aux classes d'équivalence fonctionnelle. Les auteurs montrent que la stabilité inverse a des implications intéressantes en termes d'optimisation. En se référant à l'exemple donné par [144], les auteurs de [58] soutiennent que la norme de Sobolev est plus adaptée que la norme uniforme pour le problème de la stabilité inverse. Avec cette norme, ils établissent concrètement un résultat de stabilité inverse sur les réseaux ReLU peu profonds sans biais, sous quelques conditions sur les paramètres.

En ce qui concerne les algorithmes de reconstruction stables, [68] fournit une complexité d'échantillonnage sous laquelle on peut récupérer les paramètres d'un réseau peu profond avec une fonction d'activation sigmoïde en utilisant l'entropie croisée comme perte. Pour les réseaux ReLU peu profonds fully-connected, sans biais et avec une entrée gaussienne, [70, 200, 201, 203] étudient la reconstruction stable des paramètres d'un réseau enseignant. Ils donnent une complexité d'échantillonnage sous laquelle la minimisation du risque empirique permet de récupérer les paramètres du réseau. [109] étudie la même configuration mais avec une mise en correspondance par identité qui saute une couche. Les réseaux ReLU peuvent également être utilisés pour récupérer un réseau avec une fonction d'activation de valeur absolue [108]. En fait, un neurone avec une valeur absolue peut être vu comme une somme de deux neurones ReLU.

Certains résultats existent également dans le cas de réseaux de convolution peu profonds. [32, 199, 198, 56] établissent des résultats de reconstruction stable pour les réseaux de convolution ReLU sans chevauchement. [93] donne un résultat dans le cas d'une fonction d'activation sigmoïde. Le cas des réseaux de convolution ReLU avec chevauchement est étudié dans [76].

La stabilité et la reconstruction stable pour les réseaux *profonds* sont une ques-

tion plus complexe. Quelques résultats existent sur le sujet, mais il reste en grande partie inexploré.

Parmi eux, pour les réseaux linéaires structurés profonds, [119, 117, 115] utilisent une technique de lifting tensoriel pour établir des propriétés de stabilité inverse. [119, 117] établissent des conditions nécessaires et suffisantes de stabilité inverse pour une contrainte générale sur les paramètres définissant le réseau. [115] spécialise l'analyse à la contrainte de parcimonie sur les paramètres et obtient des conditions nécessaires et suffisantes de stabilité inverse.

Les auteurs de [8] considèrent des réseaux de neurones profonds feedforward avec une fonction d'activation de Heavyside qui sont très parcimonieux et générés de manière aléatoire. Ils montrent que ces réseaux peuvent être appris avec une probabilité élevée couche par couche.

Les auteurs de [168] considèrent un réseau de neurones feedforward profond, avec une fonction d'activation qui peut être, entre autres, ReLU, sigmoïde ou softmax. Ils montrent que, si l'entrée est gaussienne ou si sa distribution est connue, et si la matrice de poids de la première couche est parcimonieuse, alors une méthode basée sur les moments et l'apprentissage de dictionnaires parcimonieux peut la récupérer exactement. Rien n'est dit sur la stabilité ou l'estimation des autres couches.

Pour les réseaux ReLU profonds, dans le cas où l'on a un accès complet à la fonction implémentée par le réseau, [159] fournit un algorithme pratique capable de récupérer approximativement les paramètres modulo permutation et rescaling, et [39] reconstruit un réseau équivalent en termes de fonction, en le formulat comme un problème cryptanalytique.

2.3.6 Nos contributions

Nous présentons ici deux contributions de cette thèse sur la question de l'identifiabilité des réseaux ReLU profonds.

Chapitre 3 : identifiabilité des paramètres à partir d'un domaine Ω La première contribution, décrite au Chapitre 3, est un résultat établissant des conditions suffisantes d'identifiabilité pour les réseaux de neurones ReLU feedforward profonds. Nous supposons que la fonction f_θ implémentée par le réseau est connue sur un sous-domaine Ω de l'espace d'entrée $\mathbb{R}^d$. En analysant la structure affine par morceaux des réseaux ReLU, et en particulier la structure des singularités de f_θ et des fonctions implémentées par les sous-réseaux, nous dérivons un ensemble de conditions suffisantes, nommées **P** et définies dans la Section 3.4.1, qui permettent de garantir l'identifiabilité. Le théorème principal peut être trouvé en tant que Théorème 17 dans la Section 3.4.2.1.

Chapitre 4 : identifiabilité locale à partir d'un échantillon fini La deuxième contribution, décrite au Chapitre 4, se concentre sur l'identifiabilité locale d'un réseau ReLU profond à partir d'un échantillon fini X. Elle fournit deux conditions, une condition nécessaire C_N et une condition suffisante C_S d'identifiabilité locale.

Étant donné que l'identifiabilité locale est nécessaire pour l'identifiabilité globale, la condition nécessaire est également une condition nécessaire pour l'identifiabilité globale. Les deux conditions C_N et C_S s'appliquent aux rangs de deux opérateurs construits à partir de θ. La particularité de ces conditions est d'être calculables en pratique. Les principaux résultats peuvent être trouvés dans la Section 4.4 en tant que Théorème 78 pour la condition nécessaire et Théorème 79 pour la condition suffisante.

2.4 Complexité et régularisation des réseaux de neurones

2.4.1 Quelques éléments de théorie du machine learning

Le développement du machine learning a conduit à la nécessité d'outils théoriques pour une meilleure compréhension et maîtrise des algorithmes. Nous nous concentrons ici sur les stratégies d'apprentissage qui reposent sur la minimisation du risque empirique. Nous considérons un modèle de machine learning $(f_\theta)_{\theta\in\mathcal{P}}$, où $\mathcal{P}$ est un ensemble de paramètres qui peut être autre que $\mathbb{R}^p$. Rappelons la définition du risque $R(\theta)$ et du risque empirique $\widehat{R}(\theta)$ donnée dans la Section 2.1.2. La minimisation du risque empirique désigne les stratégies dans lesquelles on choisit θ en essayant de minimiser le risque empirique $\hat{R}(\theta)$. Naïvement, on pourrait espérer que le risque empirique $\widehat{R}(\theta)$ du paramètre obtenu $\theta \in \mathcal{P}$ n'est pas très éloigné du risque réel $R(\theta)$ simplement en raison de la loi des grands nombres. Cela serait vrai si θ (et donc la fonction f_θ) était indépendant de l'échantillon d'apprentissage. Cependant, puisque θ est obtenu en minimisant le risque empirique, qui dépend de l'échantillon d'apprentissage, θ dépend fortement de l'échantillon.

Pour borner l'écart entre $\widehat{R}(\theta)$ et $R(\theta)$, une idée est alors de montrer que l'ensemble de fonctions $(f_\theta)_{\theta\in\mathcal{P}}$ n'est pas trop riche, c'est-à-dire qu'en quelque sorte θ ne peut pas dépendre 'trop' de l'échantillon d'apprentissage. Par exemple, en considérant un exemple très simple où il n'y a que deux paramètres, $\mathcal{P} = \{\theta_1, \theta_2\}$, on pourrait appliquer la loi des grands nombres deux fois pour borner l'erreur de généralisation pour θ_1 et θ_2, puis simplement utiliser l'inégalité de Boole pour borner l'erreur de généralisation sur $\mathcal{P}$. Cela nous permettrait de garantir que même après avoir choisi θ en fonction de l'échantillon d'apprentissage, l'erreur de généralisation peut être bornée. Maintenant, imaginons que nous ajoutions plus de paramètres θ_i dans $\mathcal{P}$. Plus il y a de choix pour θ, plus nous avons de quantités à borner simultanément, et moins nous sommes capables de borner l'écart entre $\widehat{R}(\theta)$ et $R(\theta)$. Nous voyons ici l'intuition que plus l'ensemble de fonctions (f_θ) est riche, plus il est difficile de borner l'erreur de généralisation.

En général, les ensembles de fonctions $(f_\theta)_{\theta\in\mathcal{P}}$ considérés sont infinis, donc l'approche précédente doit être améliorée. Nous avons besoin d'outils pour quantifier à quel point l'ensemble de fonctions (f_θ) est riche et divers. C'est là qu'intervient un outil tel que la VC-dimension, introduite pour la première fois par Vladimir

Vapnik et Alexey Chervonenkis. Dans un cadre de classification binaire, la VC-dimension permet de quantifier la complexité d'une famille de classificateurs. Si nous considérons un ensemble d'exemples $x^{(i)}, i \in [\![1, n]\!]$, nous disons que l'ensemble $\{x^{(1)}, \dots, x^{(n)}\}$ est *éclaté* par la famille $(f_\theta)_{\theta \in \mathcal{P}}$ si, pour toute attribution binaire d'étiquettes $\epsilon^{(1)}, \dots, \epsilon^{(n)} \in \{0, 1\}$, il existe un classificateur f_θ qui classe parfaitement les points, c'est-à-dire $f_\theta(x^{(i)}) = \epsilon^{(i)}$, pour tous $i \in [\![1, n]\!]$. La VC-dimension d'un modèle $(f_\theta)_{\theta \in \mathcal{P}}$ est alors le plus grand $n \in \mathbb{N}$ tel qu'il existe un ensemble d'exemples $x^{(i)}, i \in [\![1, n]\!]$ qui est éclaté par $(f_\theta)_{\theta \in \mathcal{P}}$. Cette notion est extensible à la classification multiclasse. Il est intuitif que plus les ensembles d'exemples que nous sommes en mesure d'éclater avec $(f_\theta)_{\theta \in \mathcal{P}}$ sont grands, plus le modèle $(f_\theta)_{\theta \in \mathcal{P}}$ est riche.

La VC-dimension est une mesure de complexité efficace qui permet de déduire des bornes de généralisation pour de nombreux modèles de machine learning. Encore une fois, l'idée générale est que plus une classe de fonctions est simple (c'est-à-dire plus sa VC-dimension est faible), plus nous sommes en mesure de borner l'erreur de généralisation. Le choix de la complexité d'un modèle est donc important, comme nous le discutons dans la section suivante.

2.4.2 Le compromis biais-variance

Lorsque l'on essaie de choisir le meilleur modèle $(f_\theta)_{\theta \in \mathcal{P}}$ pour une tâche d'apprentissage, un élément clé est la complexité du modèle. Un modèle trop simple, avec trop peu de fonctions, s'adapterait difficilement à la tâche donnée. Il risque en effet d'être difficile de s'ajuster aux données d'apprentissage et de réduire le risque empirique $\widehat{R}(\theta)$, et il risque de ne pas exister de fonction f_θ telle que le risque $R(\theta)$ soit faible.

D'un autre côté, un modèle complexe peut être assez riche pour contenir une fonction f_θ telle que le risque empirique $\widehat{R}(\theta)$ soit faible. Cependant, dans ce cas, la complexité du modèle rend plus difficile de borner l'erreur de généralisation, et en effet, le risque empirique $\widehat{R}(\theta)$ et le risque réel $R(\theta)$ risquent de différer considérablement. En ajustant trop étroitement le modèle aux données d'apprentissage, on risque d'apprendre le bruit, ce qui risque de rendre le θ obtenu par la minimisation du risque empirique trop dépendant des données d'apprentissage. Ce problème est connu sous le nom d'overfitting (ou surajustement).

Dans le paradigme classique du machine learning, il est donc nécessaire de trouver un juste équilibre entre ces deux phénomènes : choisir la bonne complexité pour le bon problème. C'est ce que l'on appelle le compromis biais-variance.

Pour les réseaux de neurones feedforward, la complexité dépend de l'architecture du réseau : la fonction d'activation utilisée et la taille du réseau, c'est-à-dire le nombre de couches (profondeur) et le nombre de neurones de chaque couche (largeur). Le choix de la fonction d'activation est en effet important, car pour certaines fonctions d'activation, même les petits réseaux ont une dimension VC infinie. La taille du réseau, elle, se reflète dans le nombre de paramètres : plus il y a de couches et de neurones dans le réseau, plus le nombre de paramètres est élevé.

Un travail considérable a été réalisé pour borner la dimension VC des réseaux de neurones, en particulier pour les fonctions d'activation lisses [19, 6], ainsi que pour les activations linéaires par morceaux telles que ReLU [16]. Les bornes existantes pour les réseaux ReLU augmentent de façon au moins linéaire avec le nombre de paramètres [16, 21], ce qui confirme que le nombre de paramètres représente la complexité d'un réseau.

2.4.3 Le paradoxe du deep learning

Étant donné que la dimension VC augmente au moins linéairement avec le nombre de paramètres, pour borner l'erreur de généralisation, il faudrait s'assurer que la taille de l'échantillon d'apprentissage est grande par rapport au nombre de paramètres. Cependant, cela entre en contradiction avec le cadre moderne du deep learning, dans lequel les réseaux de neurones fortement surparamétrés ont montré d'excellentes performances dans un large éventail de situations. Pire encore, dans certaines de ces situations, augmenter le nombre de paramètres du réseau améliore encore les performances de généralisation ! Dans de tels cadres, les bornes existantes basées sur la dimension VC sont non informatives.

On pourrait essayer de trouver des bornes plus fines pour les réseaux de neurones, cependant, les bornes existantes sont proches d'être optimales [16]. On pourrait également essayer de trouver de meilleurs outils que la dimension VC pour rendre compte de la complexité des classes de fonctions induites par les réseaux de neurones. En effet, il est par exemple possible de changer la dimension VC d'un réseau de finie à infinie en ajoutant une perturbation arbitrairement petite à ReLU [21]. Une telle perturbation ne changerait pas le comportement global du réseau, mais la dimension VC est sensible aux propriétés fines des modèles, donc elle est affectée par un tel changement. D'autres mesures pourraient être plus robustes et plus efficaces pour refléter la complexité de la classe de fonctions implémentées par un réseau neuronal.

Néanmoins, le problème semble plus général. En effet, comme le montrent [134, 195], les réseaux de neurones qui généralisent bien sur certaines tâches de classification (comme MNIST) sont suffisamment puissants pour fitter parfaitement des données avec des labels aléatoires. Cela montre que les réseaux de neurones fortement surparamétrés sont en réalité très puissants et représentent une classe de fonctions très riches. À cet égard, il existe de nombreuses paramétrisations θ telles que le f_θ correspondant est capable de fitter parfaitement les données d'apprentissage. L'ensemble des minimiseurs des fonctions objectives minimisées par les algorithmes d'optimisation tels que SGD est large [51, 106] et contient certains éléments qui généralisent mal [192, 134]. Malgré cela, les optimiseurs sont capables de trouver des fonctions qui généralisent bien en pratique. Cela montre que l'analyse globale, qui prend en compte des bornes dans le pire des cas sur $(f_\theta)_{\theta \in \mathbb{R}^p}$, avec le pire réseau possible qui fitte les données, échouera inévitablement à expliquer la généralisation. Au lieu de cela, il faut des mesures de complexité locales, qui décrivent la complexité des fonctions réellement implémentées par les réseaux de neurones optimisés par descente de gradient.

2.4.4 Régularisation implicite et mesures de complexité locales

Un effort de recherche substantiel a été fait pour obtenir de nouvelles mesures de complexité et de nouvelles bornes de généralisation pour les réseaux de neurones. Les mesures de complexité servent soit d'outil descriptif montrant que les optimiseurs tels que SGD sont implicitement biaisés en faveur de fonctions qui sont simples en un certain sens et qui généralisent bien, soit les mesures peuvent être ajoutées explicitement comme terme de régularisation pendant l'optimisation.

On peut énumérer les desiderata qu'une mesure de complexité idéale devrait satisfaire [186]. Une mesure de complexité idéale devrait s'appliquer aux réseaux utilisés en pratique et être capable de rendre compte des performances de prédiction des différentes architectures. En particulier, elle ne devrait pas augmenter avec le nombre de paramètres, car comme expliqué ci-dessus, l'ajout de paramètres même après un fitting parfait de l'échantillon d'apprentissage n'affecte pas la prédiction, voire l'améliore dans certains cas. La mesure de complexité devrait également avoir la bonne dépendance par rapport au nombre d'exemples d'apprentissage [129]. La mesure devrait également rendre compte de la complexité des ensembles de données. Par exemple, la classification sur CIFAR10 est plus difficile que sur MNIST, et la classification sur CIFAR100 est plus difficile que la classification sur CIFAR10. Lorsque l'on corrompt une partie des étiquettes de l'ensemble de données d'apprentissage, comme cela se fait dans de nombreuses expériences, on devrait voir la mesure de complexité augmenter. Outre la difficulté ou complexité d'un dataset, la mesure de complexité devrait refléter les performances de différentes méthodes d'optimisation : optimiseur, choix des hyperparamètres, techniques de régularisation, etc. Pour améliorer notre compréhension des réseaux de neurones, une mesure de complexité devrait également avoir une explication théorique et idéalement être accompagnée d'une borne de généralisation capable de prédire la généralisation. Une telle borne devrait idéalement être proche de la véritable erreur de généralisation, mais a minima, elle ne devrait pas être vide (c'est-à-dire qu'elle devrait prédire un taux d'erreur inférieur à 100 %, sans quoi elle est non informative). Enfin, une mesure de complexité devrait être calculable de manière pratique et efficace. Dans la mesure de nos connaissances, il existe des mesures de complexité satisfaisant certains des desiderata susmentionnés, mais aucune n'est capable de satisfaire, sinon tous, ne serait-ce qu'une majorité d'entre eux.

De nombreuses bornes impliquent la norme des poids des réseaux de neurones, soit directement, soit mesurée comme une distance par rapport aux poids à l'initialisation [137, 134, 15, 77, 128]. Un autre type de mesure de complexité qui a été étudié est la platitude ('flatness') du minimum de la fonction objectif obtenu par l'algorithme d'optimisation. L'idée que les minima plats peuvent généraliser mieux que les minima abrupts ('sharp') n'est en effet pas nouvelle [90], et a été explorée pour les réseaux de neurones [44, 100]. Cependant, cette mesure de généralisation a des limites, comme expliqué dans [53]. Une limite notable pour les réseaux ReLU est que l'on peut utiliser des rescalings pour changer les paramètres du réseau sans affecter la fonction qu'il implémente (et donc sans changer les performances de gé-

néralisation). De tels rescalings peuvent arbitrairement rendre un minimum de la fonction objectif plus abrupt ou plus plat.

Comme mentionné dans la section 2.1.6, pour les réseaux ReLU, le nombre et la densité des régions affines ont également été proposés comme mesures de complexité pour les réseaux ReLU. Contrairement aux mesures de complexité basées sur la norme, il n'y a pas de moyen direct d'obtenir ces mesures, mais différentes méthodes ont été proposées pour calculer ces mesures avec une certaine efficacité [125, 150, 86].

2.4.5 Régularisation implicite lors de l'optimisation

Puisque l'objectif des mesures de complexité est de montrer que certaines fonctions implémentées par les réseaux de neurones ont une faible complexité, et par conséquent de bonnes propriétés de généralisation, une partie de l'étude devrait se concentrer sur le processus d'optimisation, et notamment sur la raison pour laquelle les algorithmes d'optimisation utilisés en pratique sont orientés vers ces fonctions à faible complexité. Étant donné que ce biais existe même sans implémenter explicitement un terme de régularisation dans la fonction objectif, un tel phénomène est étudié sous le nom de biais implicite ou de régularisation implicite.

La régularisation implicite est un phénomène bien compris pour les réseaux linéaires et la factorisation de matrices. Les résultats existants montrent en effet que l'optimisation contraint implicitement le rang de la matrice de prédiction [9, 155, 167, 73, 74, 2].

L'optimisation est moins bien comprise dans le cas des réseaux non linéaires, tels que les réseaux ReLU [10]. Le fait que l'optimisation soit orientée vers des paramètres de faible rang semble moins clair pour les réseaux non linéaires que pour les réseaux linéaires [185].

Certains articles montrent que le processus d'optimisation a tendance à minimiser certaines quantités basées sur les normes [136, 30]. En particulier, une série de travaux étudie le flot de gradient et la descente de gradient pour les réseaux de neurones en classification, montrant que bien que la fonction de perte logistique fasse tendre naturellement les paramètres vers l'infini, les paramètres convergent en direction, vers des classificateurs à marge maximale pour certaines normes [48, 112, 94].

Suivant d'autres hypothèses, les auteurs de [151] utilisent l'analyse de Fourier pour montrer que les réseaux de neurones ReLU sont orientés vers l'apprentissage de fonctions de basse fréquence, et d'une autre manière les auteurs de [161] affirment que les réseaux de neurones sont biaisés vers la minimisation du nombre de régions affines.

Les propriétés découlant de la nature stochastique de la descente de gradient stochastique SGD, par rapport à la descente de gradient classique ou au flot de gradient, ont également suscité de l'intérêt. Il a été montré que la descente de gradient stochastique introduit du bruit dans le processus d'optimisation, par rapport à la descente de gradient classique ou au flot de gradient. En particulier, plus la taille

du mini-batch est petite et plus le pas d'optimisation est grand, plus il y a de bruit. Cela expliquerait pourquoi les réseaux de neurones ont tendance à converger vers des minima plats, car le bruit tend à faire sortir l'optimiseur des minima abrupts [44, 100].

Certains auteurs ont essayé de rendre explicite le biais implicite du gradient stochastique. En particulier, [172, 14, 71] ont montré que sous certaines hypothèses, en moyenne, suivre la descente de gradient stochastique sur une perte L équivaut à suivre un flot de gradient modifié, comprenant un terme de biais supplémentaire prenant la forme de la norme au carré du gradient de L.

2.4.6 Autres travaux sur la généralisation des réseaux de neurones

Dans ce paragraphe, nous décrivons brièvement quelques autres lignes de recherche autour du comportement de généralisation des réseaux de neurones qui méritent d'être mentionnées.

Tout d'abord, certains auteurs ont cherché à mieux capturer le fait que les réseaux de neurones qui fittent parfaitement des données bruitées sont capables de généraliser. En effet, fitter exactement les données d'apprentissage, surtout lorsque les données sont bruitées, est classiquement considéré comme de l'overfitting et normalement évité en machine learning classique. Ces auteurs ont donc cherché à comprendre les situations où le surajustement n'impacte pas négativement la prédiction, une situation appelée 'benign overfitting' [18].

D'autres auteurs ont cherché à réconcilier le paradigme classique du machine learning avec l'apprentissage profond moderne. En particulier, en étudiant le risque en fonction du nombre de paramètres, ils ont montré l'existence de deux régimes. Dans le premier régime, lorsque le nombre de paramètres est inférieur à la taille des données, la courbe a une forme en 'U' : lorsque le nombre de paramètres est faible, le biais est prédominant, et donc ajouter des paramètres améliore les capacités d'approximation et permet ainsi de réduire le risque. En augmentant le nombre de paramètres, à un certain point, la variance prend le dessus et l'ajout de paramètres fait augmenter le risque à nouveau. Il s'agit du compromis classique biais-variance. Cependant, ce comportement change lorsque le nombre de paramètres atteint la taille de l'ensemble d'apprentissage. Après ce point (parfois appelé interpolation), l'ajout de paramètres fait à nouveau diminuer le risque. Cette forme particulière de la courbe de risque a été appelée 'double descente' et a été explorée théoriquement et observée empiriquement [23, 131].

2.4.7 Notre contribution : une mesure géométrique de la complexité locale (Chapitre 5)

Nous présentons ici une contribution de cette thèse au sujet de la complexité et de la régularisation implicite des réseaux ReLU profonds, qui correspond au Chapitre 5.

Le chapitre explore les propriétés et les aspects computationnels des mesures

de complexité locale des réseaux neuronaux ReLU profonds récemment introduites dans [82]. Les mesures de complexité considérées sont liées à la géométrie locale de l'ensemble image tel que défini par $\{f_\theta(X) \mid \theta \text{ varie}\}$ et de l'ensemble de pré-image $\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$, où $f_\theta(X)$ est la prédiction, pour un échantillon d'entrée X, réalisée par le réseau de neurones de paramètre θ.

La géométrie locale de l'ensemble de pré-image et de l'ensemble image est liée par la différentielle $\mathbf{D}f_\theta(X)$ de $\theta \mapsto f_\theta(X)$, et par le rang de celle-ci. L'ensemble de pré-image représente les redondances dans les paramètres d'un réseau. Intuitivement, plus il y a de redondances dans les paramètres, moins l'espace des fonctions représentées par le réseau est riche et complexe. Parmi d'autres propriétés, nous cherchons notamment à comprendre comment ces objets se comportent pendant l'optimisation.

Le travail décrit dans ce chapitre est directement lié à la question de l'identifiabilité. Étudier l'identifiabilité à partir d'un échantillon fini X correspond exactement à étudier l'ensemble de pré-image. Comme nous le savons, cet ensemble contiendra au moins la classe d'équivalence de θ modulo permutation et rescaling positifs. L'identifiabilité modulo permutation et rescalings positifs n'est vérifiée que si cet ensemble ne contient que cette classe d'équivalence. En revanche, plus l'ensemble de pré-image est grand, plus il y a de redondances dans les paramètres et plus nous nous éloignons de l'identifiabilité.

Il n'est donc pas surprenant que la différentielle $\mathbf{D}f_\theta(X)$ apparaisse sous une forme légèrement différente au Chapitre 4, Section 4.4, sous la forme de l'opérateur $\Gamma(X, \theta)$, et que le rang de cette différentielle apparaisse également au Chapitre 4 sous la forme de la quantité R_Γ.

Chapter 3

Parameter identifiability of a deep feedforward ReLU neural network

This chapter consists in the article [26], which is a joint work with François Bachoc and François Malgouyres and was published in *Machine Learning*.

We point out to the reader that although the neural architectures considered are the same as presented in the introduction, the notations of this chapter differ. We mention here the two biggest ones. First, the parameterization of a network is denoted $(\mathbf{M}, \mathbf{b})$ instead of θ, where $\mathbf{M}$ represents all the weights and $\mathbf{b}$ all the biases of the network. The second difference that should be paid attention to in order not to get confused, is that the indexation of the layers is here done in reverse order: the input layer is denoted layer K, and the output layer is layer 0. These differences are specific to Chapter 3, and not present in the subsequent chapters.

Abstract

The possibility for one to recover the parameters –weights and biases– of a neural network thanks to the knowledge of its function on a subset of the input space can be, depending on the situation, a curse or a blessing. On one hand, recovering the parameters allows for better adversarial attacks and could also disclose sensitive information from the dataset used to construct the network. On the other hand, if the parameters of a network can be recovered, it guarantees the user that the features in the latent spaces can be interpreted. It also provides foundations to obtain formal guarantees on the performances of the network.

It is therefore important to characterize the networks whose parameters can be identified and those whose parameters cannot.

In this article, we provide a set of conditions on a deep fully-connected feedforward ReLU neural network under which the parameters of the network are uniquely identified –modulo permutation and positive rescaling– from the function it implements on a subset of the input space.

Contents

3.1 Introduction

The development of Machine Learning and in particular of Deep Learning in the last decade has led to many breakthroughs in fields such as image classification [103], object recognition [156, 157], speech recognition [89, 164, 87], natural language processing [124, 123, 97], anomaly detection [148] or climate sciences [3]. Deep neural networks are now widely used in real-life tasks stemming from those fields and beyond. This development and the diversity of contexts in which neural networks are used require to investigate theoretical properties that permit to guarantee that they can be used safely, are robust to attack, and can be used widely without giving access to sensitive information.

One key problem in these regards is the relation between the parameters and the function implemented by the network. If a parameterization of a network uniquely defines a function, the reverse is not true. Which other parameterizations define the same function, and what do they have in common? Which information on the parameters of a network are we able to infer from the knowledge of its function on a given domain? Addressing these questions is important for different reasons: industrial property, privacy, robustness and efficiency guarantee (see Section 3.2 for further discussions and references).

In this article, we consider fully-connected feedforward neural networks with K layers, $K \geq 2$, with the ReLU activation function (see Section 3.3 for details). The weights and bias parameterizing a neural network are gathered in a list $\mathbf{M}$ of matrices and a list $\mathbf{b}$ of vectors. The corresponding function is denoted [1] $f_{\mathbf{M},\mathbf{b}} : \mathbb{R}^{n_K} \longrightarrow \mathbb{R}^{n_0}$. We say that two parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are *equivalent* if they can be deduced from each other by the permutation of neurons in each hidden layer and by positive rescaling between the inward and outward weights of every neuron of every hidden layer. These two operations, that are precisely defined in Definition 13, are well-known in the literature [149, 145, 147, 159, 178] and will be referred to as 'permutation and positive rescaling'. As is well known and restated for completeness in Proposition 14, if two parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are equivalent, then the corresponding networks implement the same function: for all $x \in \mathbb{R}^{n_K}$, $f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x)$. In other words, parameter equivalence implies *functional equivalence* of the networks.

The main contribution of this article is an *identifiablity statement* (see Theorem 17) which establishes a 'weak' converse of this statement. We consider a set $\Omega \subset \mathbb{R}^{n_K}$ and two parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ sharing the same architecture (number of layers and of neurons per layer). We establish a *sufficient condition* $\mathbf{P}$ such that, if for all $x \in \Omega$, $f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x)$ and the condition $\mathbf{P}$ is met, then the two parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are equivalent. The motivation for the introduction of the set Ω is that, in practice, we may only test the values of $f_{\mathbf{M},\mathbf{b}}$ and $f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$ on a subset of $\mathbb{R}^{n_K}$. Typically, Ω is a subset of the support of the input distribution law. Such a setting also allows to show that two networks which coincide on a given domain actually coincide on the whole input space $\mathbb{R}^{n_K}$. Indeed, if the functions implemented by the networks coincide on Ω and if the sufficient condition $\mathbf{P}$ is satisfied, then the parameters are equivalent and thus by Proposition 14 the functions also coincide on the rest of the input space $\mathbb{R}^{n_K}$. This can be useful to bound the generalization error.

We also reformulate this identifiability statement (see Corollary 18) in a way that illustrates its interest with regard to risk minimization. The corollary considers a random variable X generating the input and an output of the form $Y = f_{\mathbf{M},\mathbf{b}}(X)$, for some parameters $(\mathbf{M}, \mathbf{b})$. It states that, when the condition $\mathbf{P}$ is met, any estimated neural network $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ for which the population risk equals 0 belongs to the equivalence class of $(\mathbf{M}, \mathbf{b})$. In words, the only way to have a perfect prediction is to perfectly recover $(\mathbf{M}, \mathbf{b})$, up to permutation and positive rescaling.

We describe the related works in Section 3.2. In addition to the works providing identifiability, stability or stable recovery statements, we give a few pointers on privacy, robustness and guarantees of efficiency that motivate our study from an applied perspective. We define in Section 3.3 the considered neural networks and provide the (known) properties that are useful in our context. The sufficient condition $\mathbf{P}$ and the main theorems are in Section 3.4. The sketch of the proofs is in

1. For clarity of the proofs, we index the layers from K (input) to 0 (output). The input layer is not counted hence the 'K layers'.

Section 3.5 and the details are in the Appendix.

3.2 Related work

3.2.1 Identifiability, stability and stable recovery

3.2.1.1 Identifiability

Identifiability of the parameters of neural networks has been the topic of a fair amount of work. For smooth activation functions, some results were already established in the 1990s. For shallow networks, results exist for activation functions amongst which tanh [181, 4], the logistic sigmoid [105], or the Gaussian and rational functions [96]. For deep networks, [61] shows that with tanh as activation function, with only a few generic conditions on the parameters, two networks that implement the same function have the same architecture and the same parameters up to some permutations and sign-flip operations.

In the case of ReLU networks, we have seen that two operations are well known to preserve the function implemented by the network: permutation and positive rescaling. These operations define equivalence classes on the set of parameters, and we can at best identify the parameters of a network up to these equivalences. It is shown in [147] that these operations are the only generic operations of this kind for ReLU networks with nonincreasing number of neurons per layer. Indeed, they show that for any fully-connected ReLU network architecture with nonincreasing number of neurons per layer, for any nonempty open set Ω, there exists a parameterization $(\mathbf{M}, \mathbf{b})$ such that for any other parameterization $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ satisfying some generic assumption, if $f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$ coincides with $f_{\mathbf{M},\mathbf{b}}$ on Ω, then $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ in the equivalence class of $(\mathbf{M}, \mathbf{b})$.

In this work, in order to establish identifiability, we take advantage of the piecewise linear geometry of the functions implemented by ReLU networks to identify the parameters. Indeed, it is well known that the function defined by a deep ReLU network is continuous piecewise-linear, i.e. we can partition the input space into polyhedral regions, sometimes called 'linear regions', over which the function is affine. These regions are separated by boundaries that are made of pieces of hyperplanes and that correspond to the non differentiabilities of the function. One crosses such a boundary when the pre-activation value of a neuron (before applying the ReLU function) changes sign. By observing the boundary, one can infer information about the weights and bias of the said neuron.

Other articles adopt similar strategies for shallow [145] or deep networks [159, 147, 178, 179]. The specificity of our proof is to proceed by induction, identifying the weights and bias layer after layer. We discuss the differences between our condition **P** and the sufficient conditions given in [147, 159, 179] in detail in Section 3.4.3.3.

In the case of shallow ReLU networks, [145] establishes a sufficient condition on the parameters for identifiability. If the condition is satisfied by two two-layer fully-connected feedforward ReLU networks whose functions coincide on all the input

space, then the parameters of one network can be obtained from the parameters of the other network by permutation and positive rescaling.

In the case of deep ReLU networks, [159] gives a sufficient condition to be able to reconstruct the architecture, weights and biases of a deep ReLU network by knowing its input-output map on all the input space. The condition concerns the boundaries mentioned above: for each neuron in a hidden layer, the authors define the boundary associated to the neuron as the points at which the pre-activation value of the neuron is zero. Then, the condition requires each boundary associated to a neuron in a layer k to intersect the boundaries associated to all the neurons in layer $k+1$ and $k-1$ (see Section 3.4.3.3 for more details).

Another kind of property is local identifiability, which is identifiability of a parameter $(\mathbf{M}, \mathbf{b})$ amongst a set of parameters that are close to $(\mathbf{M}, \mathbf{b})$. [179] studies this property for shallow and deep networks. For a deep ReLU network, it first shows that under a trivial assumption, general identifiability up to permutation and positive rescaling implies local identifiability up to positive rescaling, and that the non-existence of 'twin' neurons is necessary to identifiability and local identifiability. Then, [179] makes a breakthrough by giving an abstract necessary and sufficient condition on $(\mathbf{M}, \mathbf{b})$ such that there exists a well-chosen *finite* set Ω from which local identifiability holds up to positive rescalings, and it gives a bound on the size of the set. Furthermore, more recently, [28] provided a numerically testable condition for local identifiability also from a finite set Ω.

Finally, another line of work that can be linked to identifiability is the field of lossless compression of neural networks [169, 170].

3.2.1.2 Inverse stability and stable recovery

Establishing identifiability properties is a first step towards establishing inverse stability properties and studying stable recovery algorithms. Given a norm between functions, we say that inverse stability holds when the proximity of the functions implemented by two networks with the same architecture implies the proximity of the corresponding parameters -up to equivalences of parameters, for instance permutations and positive rescalings in the case of ReLU networks. Inverse stability is a stronger property than identifiability, and is necessary for stable recovery algorithms, which goal is to practically recover the parameters of a network from its function.

Inverse stability does not hold in general with the uniform norm for fully-connected feedforward neural networks. Indeed, [144] shows that for any depth, for any architecture with at least 3 neurons in the first hidden layer and any practically used activation function, there exists a sequence of networks whose function tends uniformly to 0 while any parameterization of these networks tends to infinity.

Many inverse stability and stable recovery results already exist for shallow networks. [58] studies inverse stability directly up to functional equivalence classes, without specifying the nature of these classes in terms of parameters -which interests us in this paper. The authors show that inverse stability has interesting

implications in terms of optimization, allowing to link the minima in the parameter space to minima in the realization space (the space of all the functions that can be implemented by a network) and to estimate the quality of local minima in the parameter space based on their radii. Referring to the counter-example given by [144], the authors of [58] argue that the Sobolev norm is more suited than the uniform norm to the problem of inverse stability. With this norm, they concretely establish an inverse stability result on shallow ReLU networks without bias, under a few conditions on the parameters.

When it comes to stable recovery algorithms, [68] provides a sample complexity under which one can recover the parameters of a shallow network with sigmoid activation function using cross-entropy as a loss. For shallow fully-connected ReLU networks, without bias and with Gaussian input, [70, 200, 201, 203] study the stable recovery of the parameters of a teacher network. They give a sample complexity under which minimizing the empirical risk allows to recover the parameters of the network. [109] studies the same configuration but with an identity mapping that skips one layer. ReLU networks can also be used to recover a network with absolute value as activation function [108]. In fact, a neuron with absolute value can be seen as a sum of two ReLU neurons.

Some results also exist in the case of shallow convolutional networks. [32, 199, 198, 56] establish stable recovery results for convolutional ReLU networks with no overlapping. [93] gives a result in the case of a sigmoidal activation function. The case of convolutional ReLU networks with overlapping is studied in [76].

Stability and stable recovery for *deep* networks is a more complicated question. A few results exist on the subject, but it stays mostly unexplored.

Among them, for deep structured linear networks, [119, 117, 115] use a tensorial lifting technique to establish inverse stability properties. [119, 117] establish necessary and sufficient conditions of inverse stability for a general constraint on the parameters defining the network. [115] specializes the analysis to the sparsity constraint on the parameters, and obtains necessary and sufficient conditions of inverse stability.

The authors of [8] consider deep feed-forward networks with Heavyside activation function which are very sparse and randomly generated. They show that these can be learned with high probability one layer after another.

The authors of [168] consider a deep feed-forward neural network, with an activation function that can be, inter alia, ReLU, sigmoid or softmax. They show that, if the input is Gaussian or its distribution is known, and if the weight matrix of the first layer is sparse, then a method based on moments and sparse dictionary learning can retrieve it exactly. Nothing is said about the stability or the estimation of the other layers.

For deep ReLU networks, in the case where one has full access to the function implemented by the network [159] provides a practical algorithm able to approximately recover the parameters modulo permutation and rescaling, and [39] reconstructs a functionally equivalent network, formulating it as a cryptanalytic problem.

Further inverse stability and stable recovery results for deep ReLU networks are

still to be established. Studying identifiability for these networks, as we do in this article, is a first step towards this goal.

3.2.2 Motivations: privacy, robustness and interpretability

The generalization of deep networks in various applications such as life style choices or medical diagnosis has raised new concerns about privacy and security. Indeed, to perform well, neural networks need to be trained with many examples. The training of some models can take up to several weeks, and need huge datasets such as ImageNet, which contains millions of images. For instance, the training of the giant GPT-3 neural network costed an estimated 12 millions of dollars [31]. For this reason, trained models are valuable and their owners may want to protect them from replication.

In many applications, the training dataset also contains sensitive information that could be uncovered [132, 111, 40, 67, 46]. It is crucial, to the deployment of the solutions relying on deep networks, to guarantee that this cannot occur. For example, when the system returns a confidence indicator in the prediction or a notion of margin, the *Model Inversion Attack* described in [67] uncovers learning examples x by maximizing the confidence/margin, under a constraint that $\|f_{\mathbf{M},\mathbf{b}}(x) - y\| \leq \varepsilon$, where y is a target output. In moderate dimension, this can be achieved by simply applying $f_{\mathbf{M},\mathbf{b}}$ several times. In large dimension, the complexity of the computation is too large unless the adversary can compute $\nabla f_{\mathbf{M},\mathbf{b}}(x)$, for any x. To perform this computation, the adversary needs to know $(\mathbf{M}, \mathbf{b})$. Guaranteeing that the parameters cannot be recovered prevents this. With a slightly different objective, (differential) *privacy deep learning* also assumes that the adversary has the knowledge of the network parameters [1].

Furthermore, knowing the architecture and parameters of a network could make easier for a malicious user to attack it, for instance with adversarial attacks. Indeed, if some black-box adversarial attacks do exist [180, 165, 49], many of them use the knowledge of the parameters of the network, at least to compute the gradients [183, 79, 104, 142, 41, 127, 126, 13].

For all these reasons, the authors of [43] developed a method of preventing parameters extraction by artificially complexifying the network without changing its global behavior. This method builds on previous works on stable recovery of the parameters of the ReLU networks, and in particular on the fact that the piecewise-linear structure of the functions implemented by such networks can be used to recover the parameters. Further understanding of stable recovery for deep networks could help improve protection methods.

Another interest of our work is interpretability of deep neural networks. In some uses of deep networks we want to understand what happens at a layer level and how we can interpret the feature spaces defined by the different layers. But such an interpretation is more meaningful if we know that, for a given function implemented by the network, the parameterization is unique -up to elementary operations such as permutations and positive rescalings for ReLU networks.

3.3 Neural networks

In this section, we provide known definitions and properties of neural networks with ReLU activation functions. For a self-contained reading, all the corresponding proofs are provided in the appendix.

3.3.1 Parameterization of neural networks

We consider deep feedforward ReLU networks with $K \geq 2$ layers. To clarify any ambiguity, note that the input layer is not actually counted, as it does not gather any weights. As evoked in the introduction, we index the layers of a deep neural network in reverse order, from K to 0, for some $K \geq 2$. The input layer is the layer K, the output layer is the layer 0, and between them are $K-1$ hidden layers. We denote by $n_k \in \mathbb{N}^*$ the number of neurons of the layer k. The information contained at the layer k is a n_k-dimensional vector.

Let $k \in [\![0, K-1]\!]$. We denote the weights between the layer $k+1$ and the layer k with a matrix $M^k \in \mathbb{R}^{n_k \times n_{k+1}}$. We also consider a bias vector $b^k \in \mathbb{R}^{n_k}$ at the layer k, and the ReLU activation function, that is $\sigma(x) = \max(x, 0)$. By extension, for a vector $x = (x_1, \ldots, x_p)^T \in \mathbb{R}^p$ we also write $\sigma(x) = (\sigma(x_1), \ldots, \sigma(x_p))^T$. We denote by h_k the mapping implemented by the network between the layer $k+1$ and the layer k. If $x \in \mathbb{R}^{n_{k+1}}$ is the information contained at the layer $k+1$, the information contained at the layer k is:

$$h_k(x) = \begin{cases} \sigma(M^k x + b^k) & \text{if } k \neq 0 \\ M^k x + b^k & \text{if } k = 0. \end{cases} \tag{3.3.1}$$

The parameters of the network can be summarized in the couple $(\mathbf{M}, \mathbf{b})$, where $\mathbf{M} = (M^0, M^1, \ldots, M^{K-1}) \in \mathbb{R}^{n_0 \times n_1} \times \cdots \times \mathbb{R}^{n_{K-1} \times n_K}$ and $\mathbf{b} = (b^0, b^1, \ldots, b^{K-1}) \in \mathbb{R}^{n_0} \times \cdots \times \mathbb{R}^{n_{K-1}}$. The function implemented by the network is then $f_{\mathbf{M},\mathbf{b}} = h_0 \circ h_1 \circ \cdots \circ h_{K-1}$, from $\mathbb{R}^{n_K}$ to $\mathbb{R}^{n_0}$. We refer to Figure 3.1 for a representation of a neural network and its parameters.

3.3.2 Continuous piecewise linear functions and neural networks

We will actively use the fact that the function implemented by a deep ReLU network as well as the intermediate functions between layers are continuous piecewise linear, which means that we can partition their domain of definition in closed polyhedral subsets such that they are linear on each subset. In this paper we use indifferently 'linear' or 'affine' to describe functions of the form $x \mapsto Ax + b$, with $A \in \mathbb{R}^{n \times m}$ some matrix and $b \in \mathbb{R}^n$ some vector.

More precisely, for $m \in \mathbb{N}$, a subset $D \subset \mathbb{R}^m$ is a *closed polyhedron* iif there exist $q \in \mathbb{N}$, $a_1, \ldots, a_q \in \mathbb{R}^m$ and $b_1, \ldots b_q \in \mathbb{R}$ such that for all $x \in \mathbb{R}^m$,

$$x \in D \iff \begin{cases} a_1^T x + b_1 \leq 0 \\ \vdots \\ a_q^T x + b_q \leq 0. \end{cases} \tag{3.3.2}$$

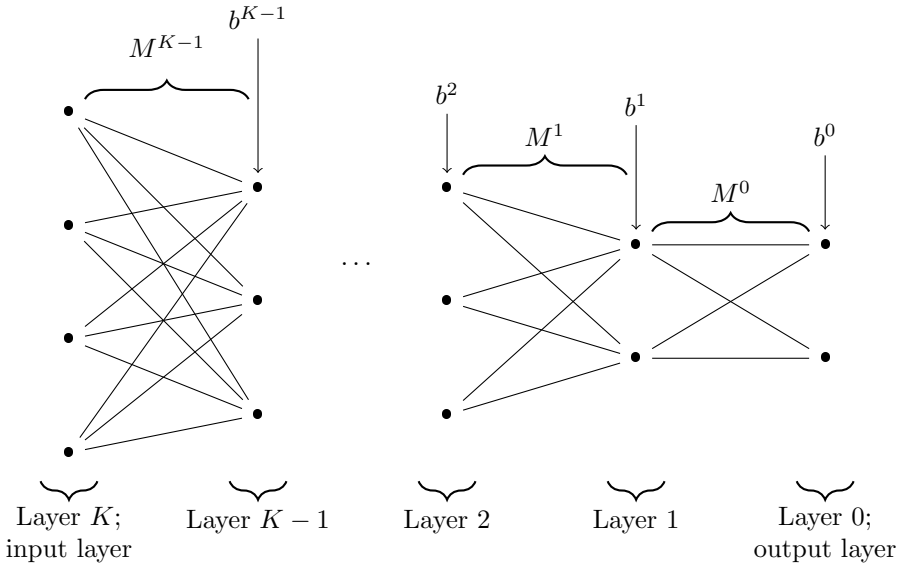

Figure 3.1 – The parameters $\mathbf{M}$ and $\mathbf{b}$ of a neural network.

By convention, if $q = 0$, we obtain an empty system of equations which is satisfied for any $x \in \mathbb{R}^m$, meaning the set $\mathbb{R}^m$ is a closed polyhedron.

We say that a function $g : \mathbb{R}^m \to \mathbb{R}^n$ is *continuous piecewise linear* if there exists a finite set of closed polyhedra whose union is $\mathbb{R}^m$ and such that g is linear over each polyhedron.

It is easy to show (see Proposition 29 in the appendix) that this definition implies the continuity of the function, hence the 'continuous' in the name. We do not require here the polyhedra to be disjoint and in fact, there are always some overlaps between the borders of adjacent polyhedra. For a given continuous piecewise linear function g, there are infinitely many possible sets of closed polyhedra that match the definition. Among them, we can always find one such that all the polyhedra D have nonempty interior $\mathring{D}$ (see Proposition 32 in the appendix). We call such a set admissible, as in the following definition.

Definition 11. Let $g : \mathbb{R}^m \to \mathbb{R}^n$ be a continuous piecewise linear function. Let Π be a set of closed polyhedra of $\mathbb{R}^m$. We say that Π is *admissible* with respect to g if and only if:

$$\begin{cases} \bigcup_{D \in \Pi} D = \mathbb{R}^m, \\ \text{for all } D \in \Pi,\ g \text{ is linear on } D, \\ \text{for all } D \in \Pi,\ \mathring{D} \neq \emptyset. \end{cases} \tag{3.3.3}$$

We now define additional functions associated to a network. Recall the layer functions h_k defined in (3.3.1), that represent the actions of the network between successive layers. Let $k \in [\![0, K]\!]$. We define the following functions:

$$\begin{aligned} f_k &= h_k \circ h_{k+1} \circ \cdots \circ h_{K-1}; \\ g_k &= h_0 \circ h_1 \circ \cdots \circ h_{k-1}. \end{aligned} \tag{3.3.4}$$

Above, by convention, we let $f_K = id_{\mathbb{R}^{n_K}}$ and $g_0 = id_{\mathbb{R}^{n_0}}$, where $id_{\mathbb{R}^m}$ denotes the identity function on $\mathbb{R}^m$. The function $f_k : \mathbb{R}^{n_K} \mapsto \mathbb{R}^{n_k}$ represents the mapping implemented by the network between the input layer and the layer k. The function $g_k : \mathbb{R}^{n_k} \mapsto \mathbb{R}^{n_0}$ represents the mapping implemented by the network between the layer k and the output layer. Hence, for all $k \in [\![0, K]\!]$ we have $g_k \circ f_k = f_{\mathbf{M},\mathbf{b}}$, and in particular $f_0 = g_K = f_{\mathbf{M},\mathbf{b}}$.

For any $\Omega \subset \mathbb{R}^{n_K}$, we also denote for all $k \in [\![0, K]\!]$,

$$\Omega_k = f_k(\Omega). \tag{3.3.5}$$

In particular, $\Omega_K = f_K(\Omega) = \Omega$.

The following proposition is easy to show by induction and using the fact that the composition of two continuous piecewise linear functions is also continuous piecewise linear (see Proposition 42 in the appendix).

Proposition 12. *For all $k \in [\![0, K]\!]$, f_k and g_k are continuous piecewise linear.*

In particular, $f_{\mathbf{M},\mathbf{b}}$ is continuous piecewise linear.

We say that a list of sets of closed polyhedra $\mathbf{\Pi} = (\Pi_1, \ldots, \Pi_{K-1})$ is *admissible* with respect to $(\mathbf{M}, \mathbf{b})$ iif for all $k \in [\![1, K-1]\!]$, the set of closed polyhedra Π_k is admissible with respect to g_k. Since there always exist such Π_k (from Proposition 12 and Proposition 32 in Appendix 3.A), there always exists an admissible list $\mathbf{\Pi}$.

3.3.3 Equivalence between two parameterizations

We are interested in sufficient conditions to identify the parameters of a network from its function. As mentioned in the introduction, some elementary operations on the parameters are well known to preserve the function of a network, so what we shall actually identify is the equivalence class of the parameters modulo these operations. There are two such operations:

— the permutation of neurons of a hidden layer;

— the positive rescalings, that is, multiplying all the outward weights of a hidden neuron by a strictly positive number and dividing the inward weights by the same number.

The invariance to permutation is classical and common to many feedforward architectures. It is described in the foundational articles [88, 45]. The invariance to positive rescalings is more specific to ReLU (and homogeneous activation functions), and is also well-studied, as for instance in [149, 145, 147, 159, 178].

We give in Definition 13 below the formalization we use for the equivalence relation modulo these operations, after introducing some notations. For all $m \in \mathbb{N}^*$, we denote by $\mathfrak{S}_m$ the set of all permutations of $[\![1, m]\!]$. For any permutation $\varphi \in \mathfrak{S}_m$, we denote by P_φ the $m \times m$ permutation matrix associated to φ, whose coefficients are defined as

$$(P_\varphi)_{i,j} = \begin{cases} 1 & \text{if } \varphi(j) = i \\ 0 & \text{otherwise.} \end{cases}$$

We also denote by $\mathbf{1}_m$ the vector $(1, 1, \ldots, 1)^T \in \mathbb{R}^m$, by $\mathbb{R}_+^*$ the set of strictly positive real numbers and by Id_m the $m \times m$ identity matrix.

Definition 13 (Equivalence between parameters)**.** If $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are two parameterizations of a network, we say that $(\mathbf{M}, \mathbf{b})$ is equivalent to $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, and we write $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, if and only if there exist:

— a family of permutations $\boldsymbol{\varphi} = (\varphi_0, \ldots, \varphi_K) \in \mathfrak{S}_{n_0} \times \cdots \times \mathfrak{S}_{n_K}$, with $P_{\varphi_0} = \mathrm{Id}_{n_0}$ and $P_{\varphi_K} = \mathrm{Id}_{n_K}$,

— a family of vectors $\boldsymbol{\lambda} = (\lambda^0, \lambda^1, \ldots, \lambda^K) \in (\mathbb{R}_+^*)^{n_0} \times \cdots \times (\mathbb{R}_+^*)^{n_K}$, with $\lambda^0 = \mathbf{1}_{n_0}$ and $\lambda^K = \mathbf{1}_{n_K}$,

such that for all $k \in [\![0, K-1]\!]$,

$$\begin{cases} \tilde{M}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{(k+1)})^{-1} P_{\varphi_{k+1}}^{-1} \\ \tilde{b}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) b^k. \end{cases} \tag{3.3.6}$$

The relation $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ is an equivalence relation [149, 145, 147]. We include a proof of this fact for completeness in Appendix 3.A (see Proposition 48). We denote by $[\mathbf{M}, \mathbf{b}]$ the equivalence class of $(\mathbf{M}, \mathbf{b})$.

We can now formalize in Proposition 14 the fact discussed at the beginning of the section: two equivalent parameterizations modulo permutation and positive rescaling implement the same function. As mentioned, this result is well-known [149, 145, 147, 159, 178], but we prove it for completeness in Appendix 3.A (see Proposition 49 and Corollary 50).

Proposition 14. *If* $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$*, then* $f_{\mathbf{M},\mathbf{b}} = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$.

In this article we give a set of conditions under which we have a reciprocal, i.e. if two parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ satisfying the conditions lead to the same function on a set Ω, i.e. $f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x)$ for all $x \in \Omega$, then they are equivalent: $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$.

3.4 Main result

The core of our work is exposed in this section. It is structured as follows. In Section 3.4.1, we expose the conditions **P** and in Section 3.4.2 we state our main theorems of identifiability. Then, Section 3.4.3 is dedicated to an extensive discussion of the conditions **P**, with motivating examples and comparison to the state of the art.

3.4.1 Conditions

We expose in this section the conditions under which the main theorem holds. They are formalized in Definition 15 and referred to as conditions **P**.

First, we introduce a few notations. We consider a network with $K \geq 2$ layers and with parameters $(\mathbf{M}, \mathbf{b})$, a list of sets of closed polyhedra $\boldsymbol{\Pi} = (\Pi_1, \ldots, \Pi_{K-1})$

admissible with respect to $(\mathbf{M}, \mathbf{b})$ and a domain $\Omega \subset \mathbb{R}^{n_K}$. Recall the definitions (3.3.1) and (3.3.4) of the functions h_k, f_k and g_k associated to the network. For all $k \in [\![1, K-1]\!]$, g_k is continuous piecewise linear, and since $\mathbf{\Pi}$ is admissible with respect to $(\mathbf{M}, \mathbf{b})$, by definition, the set of closed polyhedra Π_k is admissible with respect to g_k in the sense of Definition 11. For all $D \in \Pi_k$, the function g_k thus coincides with a linear function on D. Since by definition the interior of D is nonempty, we define $V^k(D) \in \mathbb{R}^{n_0 \times n_k}$ and $c^k(D) \in \mathbb{R}^{n_0}$ as the unique couple satisfying, for all $x \in D$:

$$g_k(x) = V^k(D)x + c^k(D). \tag{3.4.1}$$

For $\Omega \subset \mathbb{R}^{n_K}$, recall the definition (39) of Ω_k, for all $k \in [0, K]$. For any $m, n \in \mathbb{N}^*$, for any $m \times n$ matrix Σ, for any $i \in [\![1, m]\!], j \in [\![1, n]\!]$, we denote by $\Sigma_{i,.}$ the i^{th} row vector of Σ and by $\Sigma_{.,j}$ the j^{th} column vector of Σ. We denote $E_i^k = \{x \in \mathbb{R}^{n_k},\ x_i = 0\}$, and $h_k^{lin}(x) = M^k x + b^k$. For any $m \in \mathbb{N}^*$ and any subset $A \subset \mathbb{R}^m$, we denote by ∂A the topological boundary with respect to the standard topology of $\mathbb{R}^m$.

Definition 15. We say that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ satisfies the conditions $\mathbf{P}$ iif for all $k \in [\![1, K-1]\!]$:

$\mathbf{P}.a)$ M^k is full row rank;

$\mathbf{P}.b)$ for all $i \in [\![1, n_k]\!]$, there exists $x \in \mathring{\Omega}_{k+1}$ such that

$$M_{i,.}^k x + b_i^k = 0,$$

or equivalently

$$E_i^k \cap h_k^{lin}(\mathring{\Omega}_{k+1}) \neq \emptyset;$$

$\mathbf{P}.c)$ for all $D \in \Pi_k$, for all $i \in [\![1, n_k]\!]$, if $E_i^k \cap D \cap \Omega_k \neq \emptyset$ then $V_{.,i}^k(D) \neq 0$;

$\mathbf{P}.d)$ for any affine hyperplane $H \subset \mathbb{R}^{n_{k+1}}$,

$$H \cap \mathring{\Omega}_{k+1} \not\subset \bigcup_{D \in \Pi_k} \partial h_k^{-1}(D).$$

The conditions $\mathbf{P}$ are invariant modulo equivalences of parameters. Indeed, as shown by the following proposition, if some parameters $(\mathbf{M}, \mathbf{b})$ satisfy the conditions $\mathbf{P}$, then all the parameters in their equivalence class satisfy them too.

Proposition 16. *Suppose* $(\mathbf{M}, \mathbf{b})$ *and* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ *are two equivalent network parameterizations, and suppose that there exists a list* $\mathbf{\Pi}$ *admissible with respect to* $(\mathbf{M}, \mathbf{b})$ *such that* $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ *satisfies the conditions* $\mathbf{P}$.

Then, there exists a list $\tilde{\mathbf{\Pi}}$ *that is admissible with respect to* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$*, and such that* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ *satisfies the conditions* $\mathbf{P}$.

Proposition 16 is proven as Proposition 60 in Appendix 3.B.

3.4.2 Main theorems

We have now introduced all the necessary material to expose our main result, Theorem 17, as well as an application in terms of risk minimization in Section 3.4.2.2.

3.4.2.1 Identifiability statement

Our main theorem is the following one. We provide a sketch of the proof in Section 3.5. For the complete proof, see Theorem 61 in Appendix 3.*B* and its proof in Section 3.B.4.

Theorem 17. *Let $K \in \mathbb{N}$, $K \geq 2$. Suppose we are given two networks with K layers, identical number of neurons per layer, and with respective parameters $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$. Assume $\mathbf{\Pi}$ and $\tilde{\mathbf{\Pi}}$ are two lists of sets of closed polyhedra that are admissible with respect to $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ respectively. Denote by n_K the number of neurons of the input layer, and suppose we are given a set $\Omega \subset \mathbb{R}^{n_K}$ such that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ satisfy the conditions $\mathbf{P}$, and such that, for all $x \in \Omega$:*

$$f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x).$$

Then:

$$(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}}).$$

As mentioned before, this theorem can be seen as a partial reciprocal to Proposition 14. Indeed, the latter shows that two networks with equivalent parameters modulo permutation and positive rescaling implement the same function. In other words, parameter equivalence implies functional equivalence of the networks. In Theorem 17, we state that under the conditions $\mathbf{P}$, functional equivalence (on a given domain Ω) implies parameter equivalence modulo permutation and positive rescaling.

3.4.2.2 An application to risk minimization

Assume we are given a couple of input-output variables (X, Y) generated by a ground truth network with parameters $(\mathbf{M}, \mathbf{b})$:

$$Y = f_{\mathbf{M},\mathbf{b}}(X).$$

We can use Theorem 17 to show that the only way to bring the population risk to 0 is to find the ground truth parameters -modulo permutation and positive rescaling.

Indeed, let $\Omega \subset \mathbb{R}^{n_K}$ be a domain that is contained in the support of X, and suppose $L : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}_+$ is a loss function such that $L(y, y') = 0 \Rightarrow y = y'$. Consider the population risk:

$$R(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}) = \mathbb{E}[L(f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(X), Y)].$$

We have the following result.

Corollary 18. *Suppose there exists a list of sets of closed polyhedra $\mathbf{\Pi}$ admissible with respect to $(\mathbf{M}, \mathbf{b})$ such that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ satisfies the conditions $\mathbf{P}$.*

If $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ is such that there exists a list $\tilde{\mathbf{\Pi}}$ admissible with respect to $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ such that $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ satisfies the conditions $\mathbf{P}$, and if $(\mathbf{M}, \mathbf{b}) \not\sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, then:

$$R(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}) > 0.$$

For the proof, see Corollary 62 in Appendix 3.B and its proof in Section 3.B.5.

3.4.3 Discussion on the conditions

This section is dedicated to discussing the conditions **P**. We start by explaining the different conditions **P**.a) − **P**.d) and their purpose in Section 3.4.3.1. Then, in Section 3.4.3.2, we provide counter-examples illustrating how non-identifiability arises when they are not satisfied. Finally, we compare the conditions **P** to the state of the art in Sections 3.4.3.3 and 3.4.3.4.

3.4.3.1 The conditions explained

Let us explain the conditions **P**. The first condition, **P**.a), requires the matrix $M^k \in \mathbb{R}^{n_k \times n_{k+1}}$ to have full row rank. This implies that for all $k \in [\![1, K-1]\!]$, the layer k has no more neurons than its predecessor, the layer $k+1$:

$$n_k \leq n_{k+1}.$$

Once this is satisfied, the condition is mild in the sense that it is satisfied for all matrices except a set of matrices of empty Lebesgue measure.

As a first remark about **P**.b), notice that by taking $k = K-1$, it implies that $\mathring{\Omega} = \mathring{\Omega}_K \neq \emptyset$. Thus, in the main result, the set Ω over which the function implemented by the network is assumed to be known needs to have nonempty interior. In particular, Ω cannot be a finite sample set. This limitation is already present in [159], which assumes an access to the function on the whole input space and [147] which considers the function of the network on a bounded open nonempty domain. However, as we discuss in the conclusion, it seems possible to establish a result for a finite Ω, and the conditions formulated here should be a basis for future work.

The conditions **P**.b), **P**.c) and **P**.d) must be satisfied for all $k \in [\![1, K-1]\!]$, but to give a sense of them, let us see what they mean for $k = K-1$.

As explained in Section 3.3.2, the function implemented by a ReLU network is continuous piecewise linear: we can divide the input space $\mathbb{R}^{n_K}$ into polyhedral regions, over each of which the function is linear. We take advantage of this structure to acquire information about the parameters of the network. The boundaries of the polyhedral regions are of particular interest. They are made of pieces of hyperplanes, and they roughly correspond to the points where the function implemented by the network is not differentiable. We use this non differentiability property to identify the boundaries. We go from one linear region to another when there is a change of sign in the pre-activation value (input of σ) of one hidden neuron. The boundary between two linear regions is thus associated to a particular neuron of a particular hidden layer.

We separate the function implemented by the first layer of the network and the function implemented by the rest of the layers thanks to the functions defined in (3.3.1) and (3.3.4), writing

$$f_{\mathbf{M},\mathbf{b}} = g_K = g_{K-1} \circ h_{K-1},$$

$$\mathbb{R}^{n_K} \xrightarrow{h_{K-1}} \mathbb{R}^{n_{K-1}} \xrightarrow{g_{K-1}} \mathbb{R}^{n_0}.$$

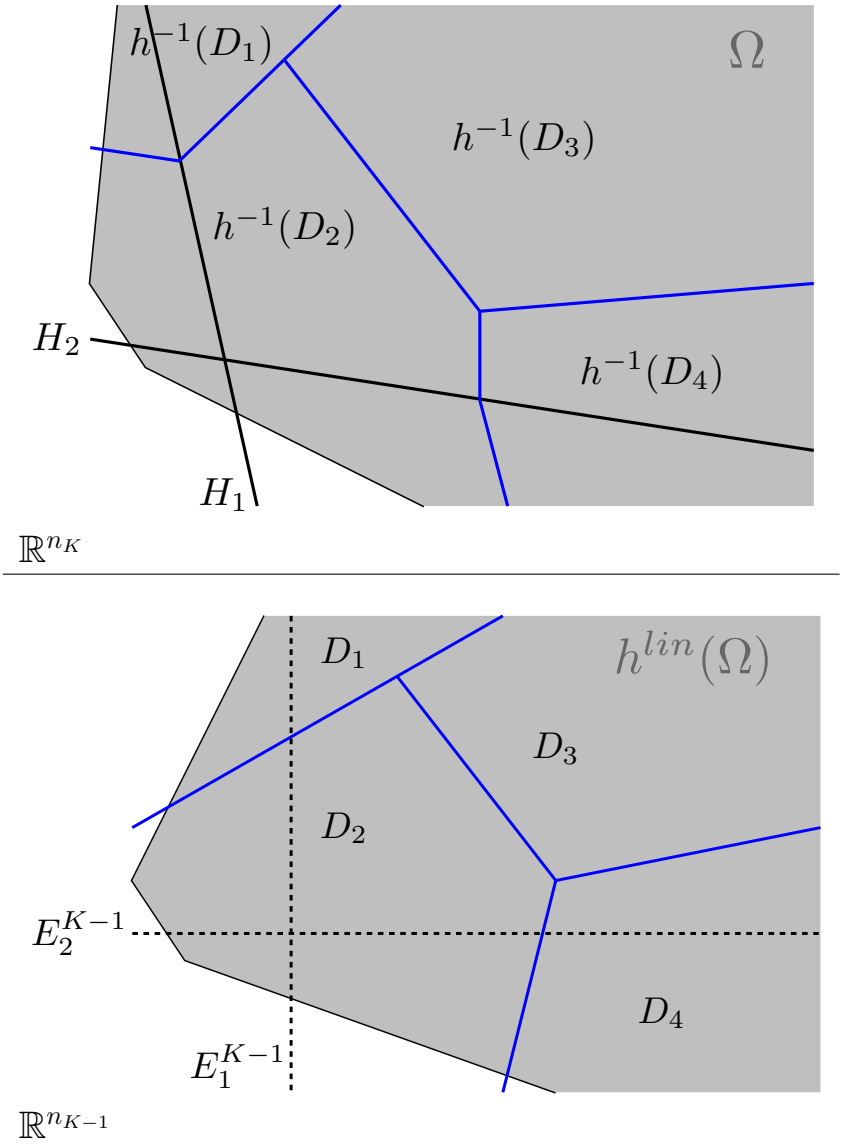

Figure 3.2 – Top. In $\mathbb{R}^{n_K}$, the inverse image by h_{K-1} of the polyhedra $D \in \Pi_{K-1}$. To make the figure lighter we write h instead of h_{K-1}. The grey zone represents Ω. For $i \in \{1, 2\}$, H_i is the hyperplane defined by the equation $M_{i,\cdot}^{K-1}x + b_i^{K-1} = 0$. As a direct consequence, we have $h(H_i) \subset E_i^{K-1}$. Bottom. In $\mathbb{R}^{n_{K-1}}$, the admissible polyhedra $D \in \Pi_{K-1}$ with respect to g_{K-1}. The grey zone corresponds to the image $h^{lin}(\Omega) = M^{K-1}\Omega + b^{K-1}$.

The goal is first to identify the weights and bias of the first layer, M^{K-1} and b^{K-1}. To do so, we focus on the boundaries associated to the neurons in the first hidden layer. These 'first-order' boundaries are hyperplanes defined by the equations $M^{K-1}_{i,\cdot}x + b^{K-1}_i = 0$, for all $i \in [\![1, n_{K-1}]\!]$. The conditions $\mathbf{P}.b)$, $\mathbf{P}.c)$ and $\mathbf{P}.d)$ are made to ensure that we are able to identify the hyperplanes, and consequently, the parameters M^{K-1} and b^{K-1}. The two relevant spaces to visualize the conditions are the input space, $\mathbb{R}^{n_K}$, and the first hidden space, $\mathbb{R}^{n_{K-1}}$, which are represented in Figure 3.2. Let us explain the conditions $\mathbf{P}.b), \mathbf{P}.c)$ and $\mathbf{P}.d)$.

$\mathbf{P}.b)$ The condition $\mathbf{P}.b)$ in the case $k = K-1$ requires the hyperplane defined by the equation $M^{K-1}_{i,\cdot}x + b^{K-1} = 0$ to intersect $\mathring{\Omega}_K = \mathring{\Omega}$. Indeed, we only consider the function implemented by the network over Ω, so the hyperplane must intersect $\mathring{\Omega}_K$ in order to be detectable as a non differentiability. In the example of Figure 3.2, we see that the two such hyperplanes, which are H_1 and H_2, intersect Ω, so the condition is satisfied.

$\mathbf{P}.c)$ Consider a polyhedron $D \in \Pi_{K-1}$. The function g_{K-1} is linear over D, and using the notations defined in (3.4.1), we have for all $u \in D$,

$$g_{K-1}(u) = V^{K-1}(D)u + c^{K-1}(D). \tag{3.4.2}$$

For all $x \in \mathbb{R}^{n_K}$ such that $h_K(x) \in D$, using (3.4.2) we obtain:

$$f_{\mathbf{M},\mathbf{b}}(x) = g_{K-1} \circ h_{K-1}(x) = \sum_{i=1}^{n_{K-1}} V^{K-1}_{\cdot,i}(D)\sigma\left(M^{K-1}_{i,\cdot}x + b^{K-1}_i\right) + c^{K-1}(D).$$

In particular, at the points x such that $M^{K-1}_{i,\cdot}x + b^{K-1}_i = 0$, the function $\sigma(M^{K-1}_{i,\cdot}x + b^{K-1}_i)$ is not differentiable, and this non differentiability can only be reflected in the function $f_{\mathbf{M},\mathbf{b}}$ if $V^{K-1}_{\cdot,i}(D) \neq 0$. The condition $\mathbf{P}.c)$ ensures that. In the example of Figure 3.2 (right part), we see that D_1 intersects E^{K-1}_1 so to satisfy $\mathbf{P}.c)$, we must have $V^{K-1}_{\cdot,1}(D_1) \neq 0$. Similarly, D_2 intersects E^{K-1}_1 and E^{K-1}_2 so we must have $V^{K-1}_{\cdot,1}(D_2) \neq 0$ and $V^{K-1}_{\cdot,2}(D_2) \neq 0$, and the polyhedron D_4 intersects E^{K-1}_2 so we must have $V^{K-1}_{\cdot,2}(D_4) \neq 0$.

$\mathbf{P}.d)$ For the last condition, $\mathbf{P}.d)$, we consider the inverse images $h^{-1}_{K-1}(D)$, for all the polyhedra $D \in \Pi_{K-1}$. Since h_{K-1} is piecewise linear and D is a closed polyhedron, $h^{-1}_{K-1}(D)$ is a finite union of closed polyhedra (see the first point of Proposition 33 in the appendix). In particular, its boundary $\partial h^{-1}_{K-1}(D)$ is made of pieces of hyperplanes. We require the union of these boundaries not to contain any full hyperplane (within the domain Ω). In the example of Figure 3.2, the condition is satisfied.

3.4.3.2 Illustrative counter-examples

To illustrate the necessity for the conditions in $\mathbf{P}$, we give for each of the conditions $\mathbf{P}.a) - \mathbf{P}.d)$ a simple example of a parameterization $(\mathbf{M}, \mathbf{b})$ and a set Ω which do not satisfy it, and we show that $(\mathbf{M}, \mathbf{b})$ is not identifiable by constructing a parameterization $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ that is not equivalent to $(\mathbf{M}, \mathbf{b})$, but such that $f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$ coincides

with $f_{\mathbf{M},\mathbf{b}}$ over Ω. These four examples illustrate the behaviors we want to prevent with the conditions $\mathbf{P}$.

Example 19. We consider an architecture with one hidden layer, i.e. $K = 2$, with $n_2 = 2$, $n_1 = 3$, $n_0 = 1$. We consider the parameterization $(\mathbf{M}, \mathbf{b})$ defined as follows.

$$M^1 = \begin{pmatrix} 0 & 2 \\ 1 & -1 \\ -1 & -1 \end{pmatrix} \qquad b^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$M^0 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \qquad b^0 = 0.$$

For this example, we consider $\Omega = \mathbb{R}$.

The condition $\mathbf{P}.a)$ is not satisfied: the matrix M^1 cannot have full row rank since its dimension is 3×2, and more specifically we have the relation

$$M^1_{1,.} + M^1_{2,.} + M^1_{3,.} = 0. \tag{3.4.3}$$

Let us define $\tilde{M}^1 = -M^1$ and $\tilde{\mathbf{M}} = (\tilde{M}^1, M^0)$. Let us show that $f_{\tilde{\mathbf{M}},\mathbf{b}} = f_{\mathbf{M},\mathbf{b}}$.

Let $x \in \mathbb{R}^2$. We have

$$f_{\mathbf{M},\mathbf{b}}(x) = \sigma\left(M^1_{1,.}x\right) + \sigma\left(M^1_{2,.}x\right) + \sigma\left(M^1_{3,.}x\right).$$

There exist activations $\epsilon_1, \epsilon_2, \epsilon_3 \in \{0, 1\}$, depending on x, such that

$$f_{\mathbf{M},\mathbf{b}}(x) = \epsilon_1 M^1_{1,.}x + \epsilon_2 M^1_{2,.}x + \epsilon_3 M^1_{3,.}x. \tag{3.4.4}$$

Since $\tilde{M}^1 = -M^1$ and $b^1 = 0$, the signs of the activations are switched in $f_{\tilde{\mathbf{M}},\mathbf{b}}$ and thus

$$\begin{aligned} f_{\tilde{\mathbf{M}},\mathbf{b}}(x) &= (1-\epsilon_1)\tilde{M}^1_{1,.}x + (1-\epsilon_2)\tilde{M}^1_{2,.}x + (1-\epsilon_3)\tilde{M}^1_{3,.}x \\ &= \sum_{i=1}^{3}(1-\epsilon_i)(-M^1_{i,.}x) \\ &= \sum_{i=1}^{3}\epsilon_i M^1_{i,.}x - (\sum_{i=1}^{3} M^1_{i,.})x \\ &= f_{\mathbf{M},\mathbf{b}}(x), \end{aligned}$$

where we obtain the last equality thanks to (3.4.3) and (3.4.4).

Now since only the positive rescalings are authorized, $(\tilde{\mathbf{M}}, \mathbf{b})$ is not equivalent to $(\mathbf{M}, \mathbf{b})$, which shows that $(\mathbf{M}, \mathbf{b})$ is not identifiable modulo permutation and rescaling.

Example 20. Let us consider a very simple architecture with one hidden layer and only one neuron per layer, i.e. $n_2 = n_1 = n_0 = 1$. We consider the parameterization $(\mathbf{M}, \mathbf{b}_a)$, for $a > 0$, defined by

$$M^1 = 1, \quad b^1_a = a, \quad M^0 = 1, \quad b^0_a = -a.$$

The function implemented by the network satisfies, for all $x \in \mathbb{R}$,

$$f_{\mathbf{M},\mathbf{b}_a}(x) = \sigma(x+a) - a = \begin{cases} -a & \text{if } x < -a \\ x & \text{if } x \geq -a. \end{cases} \tag{3.4.5}$$

Let us consider $\Omega = [1, +\infty[$. With such a choice of Ω, none of the parameterizations $(\mathbf{M}, \mathbf{b}_a)$ satisfy $\mathbf{P}.b)$, because for all $x \in \Omega$, $M^1 x + b_a^1 = x + a > 0$.

For any $a > 0$, for any $x \in \Omega$, we have $x \geq 1 > -a$, so (3.4.5) shows that $f_{\mathbf{M},\mathbf{b}_a}(x) = x$, i.e. the functions implemented by the parameterizations $(\mathbf{M}, \mathbf{b}_a)$ all coincide over Ω. However, since (3.4.5) shows they do not implement the same function over $\mathbb{R}$, Proposition 14 shows they are not equivalent.

Example 21. We consider an architecture with 2 hidden layers, that is $K = 3$, and again one neuron per layer: $n_3 = n_2 = n_1 = n_0 = 1$. Let us consider the parameterizations $(\mathbf{M}, \mathbf{b}_a)$, defined for $a > 0$ by

$$\begin{aligned} M^2 &= 1, \qquad & b_a^2 &= a, \\ M^1 &= 1, \qquad & b_a^1 &= -1 - a, \\ M^0 &= 1, \qquad & b_a^0 &= 0. \end{aligned}$$

We consider $\Omega = \mathbb{R}$. Let us show that for any $a > 0$ and any admissible $\mathbf{\Pi}$, the condition $\mathbf{P}.c)$ is not satisfied by $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ in the case $k = 2$.

Indeed, we have $h_{2,a}(x) = \sigma(M^2 x + b_a^2) = \sigma(x+a)$, and thus $\Omega_2 = h_{2,a}(\Omega) = \mathbb{R}_+$. Further, the expression of $g_{2,a}$ is

$$g_{2,a}(x) = M^0 \sigma(M^1 x + b_a^1) + b_a^0 = \sigma(x - 1 - a) = \begin{cases} 0 & \text{if } x \leq 1 + a \\ x - 1 - a & \text{if } x > 1 + a. \end{cases}$$

For any set of closed polyhedra Π_2 admissible with respect to $g_{2,a}$, a polyhedron $D \in \Pi_2$ intersecting $E_1^2 = \{0\}$ must satisfy $V^2(D) = 0$ since $g_{2,a}(x) = 0$ for $x \in]-\infty, 1 + a]$. This contradicts $\mathbf{P}.c)$ for $k = 2$.

To exhibit functionally equivalent parameterizations, we now show that for all $a > 0$ and $x \in \mathbb{R}$,

$$f_{\mathbf{M},\mathbf{b}_a}(x) = \sigma(x - 1). \tag{3.4.6}$$

Indeed, let $x \in \mathbb{R}$.

— if $x \in]-\infty, -a[$, we have $\sigma(M^2 x + b_a^2) = \sigma(x + a) = 0$, so

$$\begin{aligned} f_{\mathbf{M},\mathbf{b}_a}(x) &= M^0 \sigma(M^1 \cdot 0 + b_a^1) + b_a^0 \\ &= \sigma(b_a^1) \\ &= 0 = \sigma(x - 1). \end{aligned}$$

— if $x \in [-a, +\infty[$, we have $\sigma(M^2 x + b_a^2) = \sigma(x + a) = x + a$, so

$$\begin{aligned} f_{\mathbf{M},\mathbf{b}_a}(x) &= M^0 \sigma(M^1 (x + a) + b_a^1) + b_a^0 \\ &= \sigma(x + a - 1 - a) \\ &= \sigma(x - 1). \end{aligned}$$

This shows (3.4.6). The function $f_{\mathbf{M},\mathbf{b}_a}$ is therefore independent of $a > 0$, but if $a \neq a'$, $(\mathbf{M}, \mathbf{b}_a) \not\sim (\mathbf{M}, \mathbf{b}_{a'})$.

The lack of identifiability comes here from the fact that we do not 'observe' the non differentiability induced by the first hidden neuron, because $V^2(D) = 0$ for D containing 0. Indeed, if $\mathbf{P}.c)$ was satisfied, we would observe a non differentiability at the point at which the sign of $M^2 x + b_a^2$ changes, which is $x = -a$, and we thus would have $f_{\mathbf{M},\mathbf{b}_a} \neq f_{\mathbf{M},\mathbf{b}_{a'}}$ for $a \neq a'$.

We remark that here, even if $\mathbf{P}.c)$ was satisfied, the condition $\mathbf{P}.d)$ would not be satisfied, as we see next in Example 22.

Example 22. We consider again the architecture of Example 21, wih two hidden layers and one neuron per layer. This time, we consider the parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, defined by

$$\begin{aligned} M^2 &= 1, & b^2 &= 0, \\ M^1 &= -1, & b^1 &= 1, \\ M^0 &= -1, & b^0 &= 0, \end{aligned}$$

and

$$\begin{aligned} \tilde{M}^2 &= -1, & \tilde{b}^2 &= 1, \\ \tilde{M}^1 &= -1, & \tilde{b}^1 &= 1, \\ \tilde{M}^0 &= 1, & \tilde{b}^0 &= -1. \end{aligned}$$

We can remark without waiting further that $(\mathbf{M}, \mathbf{b}) \not\sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, for instance because $b^0 = 0$, and $\tilde{b}^0 = -1$, and the rescalings do not permit such a transformation.

Let $\Omega = \mathbb{R}$. Let us consider the sets $\Pi_2 = \{]-\infty, 1], [1, +\infty[\}$, $\Pi_1 = \{\mathbb{R}\}$ and the list $\mathbf{\Pi} = (\Pi_1, \Pi_2)$. After showing that $\mathbf{\Pi}$ is admissible with respect to $(\mathbf{M}, \mathbf{b})$, we will first show that $(\mathbf{M}, \mathbf{b})$ does not satisfy the condition $\mathbf{P}.d)$ and we will then show that $f_{\mathbf{M},\mathbf{b}} = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$.

Let us show that $\mathbf{\Pi}$ is admissible with respect to $(\mathbf{M}, \mathbf{b})$. Indeed, for all $x \in \mathbb{R}$, we have

$$g_2(x) = M^0 \sigma(M^1 x + b^1) + b^0 = -\sigma(-x + 1).$$

The function g_2 is linear over both the intervals $]-\infty, 1]$ and $[1, +\infty[$, so Π_2 is admissible with respect to g_2. The function g_1 is linear, so Π_1 is admissible with respect to g_1.

Let us now show that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ does not satisfy the condition $\mathbf{P}.d)$. Let us first determine $\bigcup_{D \in \Pi_2} \partial h_2^{-1}(D)$. Since $h_2(x) = \sigma(x)$, we have $h_2^{-1}(]-\infty, 1[) =]-\infty, 1]$ and $h_2^{-1}([1, +\infty[) = [1, +\infty[$. Hence,

$$\bigcup_{D \in \Pi_2} \partial h_2^{-1}(D) = \{1\}.$$

Now, since $\mathring{\Omega}_3 = \mathring{\Omega} = \mathbb{R}$, we have

$$\mathring{\Omega}_3 \cap \{1\} \subset \bigcup_{D \in \Pi_2} \partial h_2^{-1}(D),$$

and since $\{1\}$ is an affine hyperplane of $\mathbb{R}$, this shows that $\mathbf{P}.d)$ is not satisfied for $k = 2$.

Let us now show that for all $x \in \Omega = \mathbb{R}$, we have

$$f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x) = \begin{cases} \sigma(x) - 1 & \text{if } x \leq 1 \\ 0 & \text{if } x > 1. \end{cases} \tag{3.4.7}$$

Let us first determine $f_{\mathbf{M},\mathbf{b}}(x)$, for $x \in \mathbb{R}$. We have

$$f_{\mathbf{M},\mathbf{b}}(x) = M^0\sigma(M^1\sigma(M^2x + b^2) + b^1) + b^0 = -\sigma(-\sigma(x) + 1).$$

— If $x \leq 1$, then $\sigma(x) \leq 1$ and thus $-\sigma(x) + 1 \geq 0$. Thus, $-\sigma(-\sigma(x) + 1) = \sigma(x) - 1$, and $f_{\mathbf{M},\mathbf{b}}(x) = \sigma(x) - 1$.
— If $x > 1$, then $-\sigma(x) + 1 < 0$ and thus $-\sigma(-\sigma(x) + 1) = 0$. We thus have $f_{\mathbf{M},\mathbf{b}}(x) = 0$.

Let us now determine $f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x)$, for $x \in \mathbb{R}$. We have

$$f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x) = \tilde{M}^0\sigma(\tilde{M}^1\sigma(\tilde{M}^2x + \tilde{b}^2) + \tilde{b}^1) + \tilde{b}^0 = \sigma(-\sigma(-x + 1) + 1) - 1.$$

— If $x \leq 1$, then $-x + 1 \geq 0$ and thus $f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x) = \sigma((x - 1) + 1) - 1 = \sigma(x) - 1$.
— If $x > 1$, then $-x + 1 \leq 0$ and thus $f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x) = \sigma(1) - 1 = 0$.

This shows (3.4.7), and as a consequence, $(\mathbf{M}, \mathbf{b})$ is not identifiable.

In this example, the lack of identifiability comes from the fact that the sets of non differentiabilities induced by the first and the second layer are indistinguishable: they are both reduced to a point. This will always be the case for networks with only one neuron per layer and more than one hidden layer. When the input dimension is 2 or higher and the condition $\mathbf{P}.d)$ is satisfied, the non differentiabilities induced by neurons in the first hidden layer are the only ones that correspond to full hyperplanes, and this is how they can be identified, as illustrated for instance in the example of Section 3.4.3.4.

3.4.3.3 Comparison with the existing work

To our knowledge, there are only two existing results on global identifiability of deep ReLU networks (with bias), as we consider here, exposed in the recent contributions [147] and [159]. Let us compare our hypotheses with theirs.

The authors of [147] introduce two notions: the notion of *general* network and the notion of *transparent network*. They note the fact that some boundaries of non differentiablity bend over some others to build a graph of dependency. The main result in [147] applies to networks whose number of neurons per layer n_k is non-increasing, as is the case in the present paper, that are transparent and general, and for which the graphs of dependency of the functions g_k satisfy additional technical conditions.

It can be verified that these hypotheses imply our conditions $\mathbf{P}.a)$, $\mathbf{P}.b)$ and $\mathbf{P}.c)$, which makes $\mathbf{P}.a)$, $\mathbf{P}.b)$ and $\mathbf{P}.c)$ more applicable.

When it comes to our last condition $\mathbf{P}.d)$, it can be compared to the technical conditions on the graph of dependency. These conditions address the way the boundaries associated to some neurons bend over the boundaries associated to neurons in previous layers. $\mathbf{P}.d)$ and this set of conditions are different, and neither implies the other.

The result exposed in [159] has a main strength compared to [147] and to us: it does not require the number of neurons per layer to be non-increasing. However, when it comes to the intersection of boundaries of linear regions, it requires each boundary, associated to some neuron, to intersect the boundaries associated to all the neurons in the previous layer, which appears to be a strong hypothesis to us. In comparison, we ask each boundary to intersect at least one of the boundaries associated to a neuron in a previous layer. Also, in [159], the function is supposed to be known on the whole input space, while [147] as well as us propose conditions on a domain Ω such that the knowledge of the function on Ω is enough. In both cases Ω has nonempty interior. [179, 28] open the way for considering a finite Ω by giving conditions of local identifiability in that case. To our knowledge global identifiability from a finite set has not been tackled yet for deep ReLU networks.

3.4.3.4 A simple comparative example

To shed a better light on the interest of the conditions $\mathbf{P}$, we describe in this section a simple network parameterization for which the conditions $\mathbf{P}$ apply, in contrast to the conditions described in [147, 159].

Let us consider a network architecture with 2 hidden layers (i.e. $K = 3$) and 2 neurons per layer, except the output layer containing 1 neuron: $n_3 = n_2 = n_1 = 2$ and $n_0 = 1$. Let us consider the parameterization $(\mathbf{M}, \mathbf{b})$ defined by

$$M^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad M^1 = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}, \quad M^0 = \begin{pmatrix} 1 & 1 \end{pmatrix},$$

$$b^2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad b^1 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}, \quad b^0 = 0.$$

The network implements a function $f_{\mathbf{M},\mathbf{b}} : \mathbb{R}^2 \to \mathbb{R}$. Here we simply consider $\Omega = \mathbb{R}^2$.

First, we are going to show that there exists a list $\mathbf{\Pi}$ that is admissible with respect to $(\mathbf{M}, \mathbf{b})$, and such that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ satisfies the conditions $\mathbf{P}$. Then, we shall discuss why this network parameterization does not satisfy the conditions in [147, 159].

Let us define the list $\mathbf{\Pi}$ as follows. For $\epsilon_1, \epsilon_2 \in \{-1, 1\}$, we denote by $D_{\epsilon_1,\epsilon_2}$ the closed polyhedron satisfying, for all $x \in \mathbb{R}^2$:

$$x \in D_{\epsilon_1,\epsilon_2} \quad \Leftrightarrow \quad \begin{cases} \epsilon_1 \left(M^1_{1,.} x + b^1_1 \right) \geq 0 \\ \epsilon_2 \left(M^1_{2,.} x + b^1_2 \right) \geq 0. \end{cases} \tag{3.4.8}$$

These 4 polyhedra are displayed in Figure 3.3. In other words, the polyhedron to which x belongs depends on the sign of both components of the vector M^1x+b^1. We define the set $\Pi_2 = \{D_{1,1}, D_{1,-1}, D_{-1,1}, D_{-1,-1}\}$. We also define the set $\Pi_1 = \{\mathbb{R}^2\}$, containing the single polyhedron $\mathbb{R}^2$, and we denote $\mathbf{\Pi} = (\Pi_1, \Pi_2)$. Let us show that $\mathbf{\Pi}$ is admissible with respect to $(\mathbf{M}, \mathbf{b})$. Indeed, the closed polyhedra of Π_2 cover $\mathbb{R}^2$. Furthermore, their interior is nonempty. Finally, for all $x \in \mathbb{R}^2$, we have

$$\begin{aligned} g_2(x) &= M^0\sigma\left(M^1x + b^1\right) + b^0 \\ &= \sigma\left(M^1_{1,.}x + b^1_1\right) + \sigma\left(M^1_{2,.}x + b^1_2\right). \end{aligned} \tag{3.4.9}$$

We derive from (3.4.9), (3.4.8) and the definition of the ReLU activation that for all $D_{\epsilon_1,\epsilon_2} \in \Pi_2$, the function g_2 is affine over $D_{\epsilon_1,\epsilon_2}$ of the form $g_2(x) = V^2(D_{\epsilon_1,\epsilon_2})x + c^2(D_{\epsilon_1,\epsilon_2})$, with the following values

$$\begin{aligned} &V^2(D_{1,1}) = M^1_{1,.} + M^1_{2,.} = \begin{pmatrix} 0 & 1 \end{pmatrix} && c^2(D_{1,1}) = b^1_1 + b^1_2 = 1 \\ &V^2(D_{1,-1}) = M^1_{1,.} = \begin{pmatrix} 1 & -1 \end{pmatrix} && c^2(D_{1,-1}) = b^1_1 = -1 \\ &V^2(D_{-1,1}) = M^1_{2,.} = \begin{pmatrix} -1 & 2 \end{pmatrix} && c^2(D_{-1,1}) = b^1_2 = 2 \\ &V^2(D_{-1,-1}) = \begin{pmatrix} 0 & 0 \end{pmatrix} && c^2(D_{-1,-1}) = 0. \end{aligned} \tag{3.4.10}$$

This shows that the set of closed polyhedra Π_2 is admissible with respect to g_2, and the values in (3.4.10) correspond to those of the definition (3.4.1). Moreover, since g_1 is affine, the set Π_1 is trivially admissible with respect to g_1. We conclude that the list $\mathbf{\Pi}$ is admissible with respect to $(\mathbf{M}, \mathbf{b})$.

Let us show that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ satisfies the conditions $\mathbf{P}$.

The conditions $\mathbf{P}$ must hold for $k \in [\![1, K-1]\!]$, so in our case, for $k = 2$ and $k = 1$. To check them, we will need to compute Ω_3 and Ω_2. Recalling the definition in (3.3.5), we have $\Omega_3 = \Omega = \mathbb{R}^2$. Then, since $h_2(x) = \sigma\left(M^2x + b^2\right) = \sigma(x)$, we have $\Omega_2 = \sigma(\mathbb{R}^2) = (\mathbb{R}_+)^2$.

Let us now check the conditions one by one.

$\mathbf{P}.a)$ The matrices M^2 and M^1 are both full row rank, so $\mathbf{P}.a)$ is satisfied for $k = 2$ and $k = 1$.

$\mathbf{P}.b)$ Let us first show the condition for $k = 2$. We have $\mathring{\Omega}_3 = \mathring{\Omega} = \mathbb{R}^2$, so taking $x_1 = (0,1)^T \in \mathring{\Omega}_3$ and $x_2 = (1,0)^T \in \mathring{\Omega}_3$, we find

$$\begin{cases} M^2_{1,.}x_1 + b^2_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0 \\ M^2_{2,.}x_2 + b^2_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = 0. \end{cases}$$

Let us now show the condition for $k = 1$. We have $\mathring{\Omega}_{k+1} = \mathring{\Omega}_2 = (\mathbb{R}^*_+)^2$. Let us choose $x_3 = (2,1)^T \in \mathring{\Omega}_2$ and $x_4 = (4,1)^T \in \mathring{\Omega}_2$. We have

$$\begin{cases} M^1_{1,.}x_3 + b^1_1 = \begin{pmatrix} 1 & -1 \end{pmatrix}\begin{pmatrix} 2 \\ 1 \end{pmatrix} - 1 = 0 \\ M^1_{2,.}x_4 + b^1_2 = \begin{pmatrix} -1 & 2 \end{pmatrix}\begin{pmatrix} 4 \\ 1 \end{pmatrix} + 2 = 0. \end{cases}$$

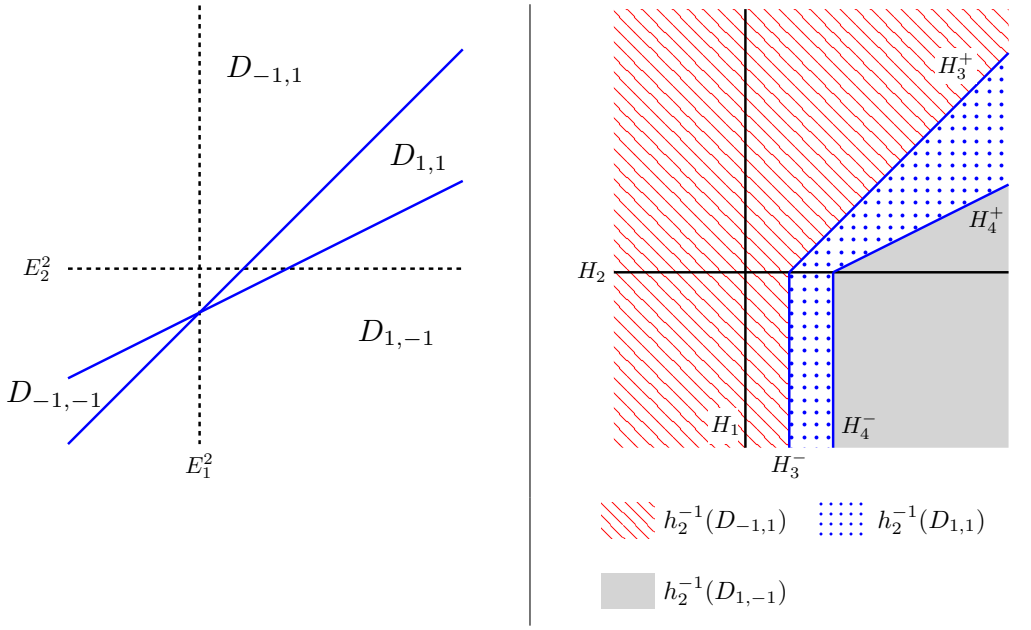

Figure 3.3 – Left. The closed polyhedra of Π_2. Right. The reciprocal images by h_2 of the closed polyhedra of Π_2.

This shows that **P**.*b*) is satisfied for $k = 2$ and $k = 1$.

P.*c*) For $k = 2$, let us recall from (3.4.10) the values of $V^2(D)$ for all $D \in \Pi_2$. In the case of $V^2(D_{1,-1})$ and $V^2(D_{-1,1})$, **P**.*c*) is clearly satisfied. When it comes to $D_{1,1}$, we have $V^2_{\cdot,1}(D_{1,1}) = 0$, but $D_{1,1}$ does not intersect E_1^2 in Ω_2. Finally, we have $V^2_{\cdot,1}(D_{-1,-1}) = V^2_{\cdot,2}(D_{-1,-1}) = 0$, but $D_{-1,-1} \cap \Omega_2 = \emptyset$.

We thus conclude that **P**.*c*) is satisfied for $k = 2$.

The case $k = 1$ is easier, $\Pi_1 = \{\mathbb{R}^2\}$ and for all $x \in \mathbb{R}^2$, we have $g_1(x) = M^0 x + b^0$, so we have $V^1(\mathbb{R}^2) = M^0 = \begin{pmatrix} 1 & 1 \end{pmatrix}$, and **P**.*c*) is clearly satisfied.

P.*d*) Here, the case $k = 1$ is trivial since $\Pi_1 = \{\mathbb{R}^2\}$, and $h_1^{-1}(\mathbb{R}^2) = \mathbb{R}^2$, and thus $\partial h_1^{-1}(\mathbb{R}^2) = \emptyset$.

We thus only need to study the case $k = 2$. Let us first determine the sets $h_2^{-1}(D)$, for $D \in \Pi_2$. We remind that for all $x \in \mathbb{R}^2$, $h_2(x) = \sigma(x)$.

For this, let us divide $\mathbb{R}^2$ in 3 regions. Let $x = (x_1, x_2) \in \mathbb{R}^2$.

— If $x_1 < 0$, then $h_2(x) = (0, \sigma(x_2))^T$. We thus have

$$M^1 h_2(x) + b^1 = \begin{pmatrix} -1(\sigma(x_2) + 1) \\ 2(\sigma(x_2) + 1) \end{pmatrix}.$$

Since $\sigma(x_2) \geq 0$, we see that $h_2(x) \in D_{-1,1}$.

— If $x_1 \geq 0$ and $x_2 < 0$, then $h_2(x) = (x_1, 0)^T$. We thus have

$$M^1 h_2(x) + b^1 = \begin{pmatrix} x_1 - 1 \\ -x_1 + 2 \end{pmatrix}.$$

There are 3 possibilities: if $x_1 \leq 1$, then $h_2(x) \in D_{-1,1}$, if $1 \leq x_1 \leq 2$, $h_2(x) \in D_{1,1}$, and if $2 \leq x_1$, $h_2(x) \in D_{1,-1}$.

— If $x_1, x_2 \geq 0$, then $h_2(x) = x$ and for all $D_{\epsilon_1,\epsilon_2} \in \Pi_2$, $h_2(x) \in D_{\epsilon_1,\epsilon_2} \iff x \in D_{\epsilon_1,\epsilon_2}$. There are 3 possibilities, $x \in D_{-1,1}, x \in D_{1,-1}$ and $x \in D_{1,1}$ since $D_{-1,-1} \cap (\mathbb{R}_+)^2 = \emptyset$.

Summarizing, we find (see also Figure 3.3):

$$h_2^{-1}(D_{-1,1}) = \mathbb{R}_- \times \mathbb{R} \quad \cup \quad [0,1] \times \mathbb{R}_- \quad \cup \quad (\mathbb{R}_+)^2 \cap D_{-1,1}$$

$$h_2^{-1}(D_{1,1}) = [1,2] \times \mathbb{R}_- \quad \cup \quad (\mathbb{R}_+)^2 \cap D_{1,1}$$

$$h_2^{-1}(D_{1,-1}) = [2,+\infty[\times\mathbb{R}_- \quad \cup \quad (\mathbb{R}_+)^2 \cap D_{1,-1}$$

$$h_2^{-1}(D_{-1,-1}) = \emptyset.$$

To express the boundaries of these regions, we define the following pieces of hyperplanes:

$$H_3^+ = \{x = (x_1, x_2) \in \mathbb{R}^2, \quad x_1 \geq 0, x_2 \geq 0, M^1_{1,.}x + b^1_1 = 0\}$$
$$H_4^+ = \{x = (x_1, x_2) \in \mathbb{R}^2, \quad x_1 \geq 0, x_2 \geq 0, M^1_{2,.}x + b^1_2 = 0\}$$
$$H_3^- = \{x = (x_1, x_2) \in \mathbb{R}^2, \quad x_1 = 1, x_2 \leq 0\}$$
$$H_4^- = \{x = (x_1, x_2) \in \mathbb{R}^2, \quad x_1 = 2, x_2 \leq 0\}.$$

We have

$$\partial h_2^{-1}(D_{-1,1}) = H_3^- \cup H_3^+$$
$$\partial h_2^{-1}(D_{1,1}) = H_3^- \cup H_3^+ \cup H_4^- \cup H_4^+$$
$$\partial h_2^{-1}(D_{1,-1}) = H_4^- \cup H_4^+$$
$$\partial h_2^{-1}(D_{-1,-1}) = \emptyset,$$

and thus,

$$\bigcup_{D \in \Pi_2} \partial h_2^{-1}(D) = H_3^- \cup H_3^+ \cup H_4^- \cup H_4^+. \tag{3.4.11}$$

Let us check the condition $\mathbf{P}.d)$ for $k = 2$. Since $\Omega_3 = \mathbb{R}^2$, here $\mathring{\Omega}_3 \cap H = H$. The condition is thus satisfied if and only if $\bigcup_{D\in\Pi_2} \partial h_2^{-1}(D)$ does not contain any full hyperplane H, and (3.4.11) shows that it is the case. The condition $\mathbf{P}.d)$ is satisfied.

Let us now discuss the conditions given in [147, 159] for this example. Let us first define the following hyperplanes:

$$H_1 = \{x \in \mathbb{R}^2, M^2_{1,.}x + b^2_1 = 0\}$$

$$H_2 = \{x \in \mathbb{R}^2, M^2_{2,.}x + b^2_2 = 0\}.$$

To discuss the conditions in [147], we will refer to their concepts of fold-set, of piece-wise linear surface, of canonical representation and dependency graph of a

piece-wise linear surface, as well as to their Lemma 4. We also use their notations $\square_1 S$ and $\square_2 S$. Let us now consider the set S as the fold-set of the function $f_{\mathbf{M},\mathbf{b}}$ implemented by the network. Here, it corresponds to the points x satisfying one of the following equations

$$\begin{cases} M^2_{1,\cdot} x + b^2_1 = 0 \\ M^2_{2,\cdot} x + b^2_2 = 0 \\ M^1_{1,\cdot} h_2(x) + b^1_1 = 0 \\ M^1_{2,\cdot} h_2(x) + b^1_2 = 0. \end{cases}$$

In other words, $S = H_1 \cup H_2 \cup H_3^- \cup H_3^+ \cup H_4^- \cup H_4^+$. The canonical representation of S is the following

$$S = (H_1 \cup H_2) \cup \left(H_3^- \cup H_3^+ \cup H_4^- \cup H_4^+\right),$$

where $\square_1 S = (H_1 \cup H_2)$ and $\square_2 S = S$. Further, it can be checked that the dependency graph of S only contains the edges: $H_2 \to H_3^-$, $H_2 \to H_3^+$, $H_2 \to H_4^-$ and $H_2 \to H_4^+$.

The identifiable networks considered in [147] must satisfy the conditions of Lemma 4 in [147]. In particular, the dependency graph of S must contain at least 2 directed paths of length 1 with distinct starting vertices, which is not the case here since all the paths of length 1 start from H_2. Hence, this network does not fall under the conditions of [147].

Now if we use the concepts and notations of [159], let us denote by z_1 the first neuron of the first hidden layer, whose associated parameters are $M^2_{1,\cdot}$ and b^2_1. Following the definition in [159], the boundary associated to z_1 is $B_{z_1} = H_1$. Let us denote by z_3 the first neuron of the second hidden layer, whose associated parameters are $M^1_{1,\cdot}$ and b^1_1. Its boundary is $B_{z_3} = H_3^- \cup H_3^+$. Since z_1 and z_3 belong to two consecutive layers and are thus linked by an edge of the network, the conditions in [159] (see Theorem 2) require that B_{z_1} and B_{z_3} intersect. It is however clear that they do not (see Figure 3.3).

3.5 Sketch of proof of Theorem 17

Our main result, Theorem 17, is proven in details in Appendix 3.B.4, and we give a sketch of the proof in this section. It is proven by induction. We are given two parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, two lists $\mathbf{\Pi}$ and $\tilde{\mathbf{\Pi}}$ that are admissible with respect to $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ respectively, and a domain Ω that satisfy the hypotheses of Theorem 17 and we want to show that the two parameterizations are equivalent. For this, we identify the layers one after the other. To facilitate identification at a layer level, we begin with a normalisation step.

3.5.1 Normalisation step

Two equivalent parameterizations do not necessarily have equal weights on their layers. Indeed, the neuron permutations but more importantly the rescalings can

change the structure of the intermediate layers. We are going to assume the following normalisation property: for all $k \in [\![1, K-1]\!]$, for all $i \in [\![1, n_k]\!]$, we have

$$\begin{aligned} \|M^k_{i,\cdot}\| &= 1; \\ \|\tilde{M}^k_{i,\cdot}\| &= 1. \end{aligned} \tag{3.5.1}$$

Indeed, we show in the appendix that for a parameterization satisfying the conditions **P**, there always exists an equivalent parameterization that is normalised and that satisfies the conditions **P** (see Propositions 52 and 60). We can thus replace each parameterization $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ by an equivalent normalised parameterization. If we are able to show the normalised parameterizations are equivalent, then the original parameterizations are equivalent too.

3.5.2 Induction

The induction proof relies on Lemma 24 below. Let K be the number of layers of the network, and suppose the theorem is true for the networks with $K-1$ layers. As explained in section 3.4.1, to identify the parameters M^{K-1} and b^{K-1}, we separate the function implemented by the first layer of the network and the function implemented by the rest of the layers. For each network:

$$g_K = g_{K-1} \circ h_{K-1},$$

$$\tilde{g}_K = \tilde{g}_{K-1} \circ \tilde{h}_{K-1}.$$

We know that g_{K-1} and $\tilde{g}_{K-1}$ are continuous piecewise linear, and this will allow us to apply Lemma 24. Before stating it, we introduce a set of conditions, called **C**, that need to be satisfied in order to apply it. These conditions come immediately from **P**, and one can easily check that $(g_{K-1}, M^{K-1}, b^{K-1}, \Omega_K, \Pi_{K-1})$ and $(\tilde{g}_{K-1}, \tilde{M}^{K-1}, \tilde{b}^{K-1}, \Omega_K, \tilde{\Pi}_{K-1})$ satisfy **C**, as a direct consequence of the conditions **P** being satisfied by $(\mathbf{M}, \mathbf{b}, \mathbf{\Omega}, \mathbf{\Pi})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \mathbf{\Omega}, \tilde{\mathbf{\Pi}})$.

Definition 23. Let l, m, n be integers, $M \in \mathbb{R}^{m \times l}$, $b \in \mathbb{R}^m$, $\Omega \subset \mathbb{R}^l$ be an open domain, let $g : \mathbb{R}^m \to \mathbb{R}^n$ a continuous piecewise linear function, and let Π be an admissible set of polyhedra with respect to g.

Let $D \in \Pi$. The function g coincides with a linear function on D. Since the interior of D is nonempty, we define $V(D) \in \mathbb{R}^{n \times m}$ and $c(D) \in \mathbb{R}^n$ as the unique couple satisfying, for all $x \in D$:

$$g(x) = V(D)x + c(D).$$

We denote $E_i = \{x \in \mathbb{R}^m, x_i = 0\}$.

We say that (g, M, b, Ω, Π) satisfies the conditions **C** iif

C.*a*) M is full row rank;

C.*b*) for all $i \in [\![1, m]\!]$, there exists $x \in \mathring{\Omega}$ such that

$$M_{i,\cdot}x + b_i = 0,$$

or equivalently, if we denote by h^{lin} the function $x \mapsto Mx + b$, then

$$E_i \cap h^{lin}(\mathring{\Omega}) \neq \emptyset;$$

C.*c*) for all $D \in \Pi$, for all $i \in [\![1, m]\!]$, if $E_i \cap D \cap h(\Omega) \neq \emptyset$ then $V_{.,i}(D) \neq 0$;
C.*d*) for any affine hyperplane $H \subset \mathbb{R}^l$,

$$H \cap \mathring{\Omega} \not\subset \bigcup_{D \in \Pi} \partial h^{-1}(D).$$

We can now state the lemma.

Lemma 24. *Let $l, m, n \in \mathbb{N}^*$. Suppose $g, \tilde{g} : \mathbb{R}^m \to \mathbb{R}^n$ are continuous piecewise linear functions, $\Omega \subset \mathbb{R}^l$ is a subset and let $M, \tilde{M} \in \mathbb{R}^{m \times l}$, $b, \tilde{b} \in \mathbb{R}^m$. Denote $h : x \mapsto \sigma(Mx + b)$ and $\tilde{h} : x \mapsto \sigma(\tilde{M}x + \tilde{b})$. Assume Π and $\tilde{\Pi}$ are two sets of polyhedra admissible with respect to g and $\tilde{g}$.*

Suppose (g, M, b, Ω, Π) and $(\tilde{g}, \tilde{M}, \tilde{b}, \Omega, \tilde{\Pi})$ satisfy the conditions **C**, *and for all $i \in [\![1, m]\!]$, $\|M_{i,.}\| = \|\tilde{M}_{i,.}\| = 1$.*

Suppose for all $x \in \Omega$:

$$g \circ h(x) = \tilde{g} \circ \tilde{h}(x).$$

Then, there exists a permutation $\varphi \in \mathfrak{S}_m$, such that:
— *$\tilde{M} = P_\varphi M$;*
— *$\tilde{b} = P_\varphi b$;*
— *g and $y \mapsto \tilde{g}(P_\varphi y)$ coincide on $h(\Omega)$.*

Lemma 24 is restated in Appendix 3.B as Lemma 63 and proven in Appendix 3.C.

Applying this lemma to the objects $(g_{K-1}, M^{K-1}, b^{K-1}, \Omega_K, \Pi_{K-1})$ and $(\tilde{g}_{K-1}, \tilde{M}^{K-1}, \tilde{b}^{K-1}, \Omega_K, \tilde{\Pi}_{K-1})$, we conclude that there exists a permutation φ_{K-1} such that

$$\begin{cases} \tilde{M}^{K-1} = P_{\varphi_{K-1}} M^{K-1} \\ \tilde{b}^{K-1} = P_{\varphi_{K-1}} b^{K-1}, \end{cases} \tag{3.5.2}$$

and that g_{K-1} and $y \mapsto \tilde{g}(P_{\varphi_{K-1}} y)$ coincide on $h_{K-1}(\Omega)$.

The functions g_{K-1} and $y \mapsto \tilde{g}(P_{\varphi_{K-1}} y)$ are the functions implemented by the networks $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ once we have removed the first layer, with a permutation of the input for the second one. Since they coincide on $\Omega_{K-1} = h_{K-1}(\Omega)$ and they satisfy the conditions **P**, we can apply the induction hypothesis to conclude the proof of Theorem 17. The complete proof is detailed in the appendices, as discussed above.

3.6 Conclusion

We established a set of conditions **P** under which the function implemented by a deep feedforward ReLU neural network on a subset Ω of the input space uniquely

characterizes its parameters, up to permutation and positive rescaling. This contributes to the understanding of identifiability and stable recovery for deep ReLU networks, which is still largely unexplored. The conditions under which our result holds differ from the conditions of the results established in [147] and [159], which allows us to cover new situations. To be satisfied the conditions **P** need Ω to have nonempty interior, which prevents it from being a sample set. The authors of [179, 28] are able to give a result with a finite set Ω, but for local identifiability only. Obtaining the best of both worlds, that is establishing a global identifiability result for deep ReLU networks with a finite set Ω, would be a major step forward.

Acknowledgements

Our work has benefited from the AI Interdisciplinary Institute ANITI. ANITI is funded by the French "Investing for the Future – PIA3" program under the Grant agreement n°ANR-19-PI3A-0004.

Appendices

In the appendices, we restate all the notations, definitions and results of the main text, for clarity of reading. The appendices are then organized as follows. In Appendix 3.A, we give the complete definitions and basic properties necessary to state and prove the main theorem. In Appendix 3.B, we state the main result, Theorem 61 (Theorem 17 in the main text), and we prove it. Finally, we prove the fundamental lemma used in the proof of the main theorem, Lemma 63 (Lemma 24 in the main text), in Appendix 3.C.

3.A Definitions, notations and preliminary results

Appendix 3.A is structured as follows: after giving some notations in Section 3.A.1, we recall the definition of a continuous piecewise linear function and some corresponding basic properties in Section 3.A.2 and we give our formalization of deep ReLU networks as well as some well-known properties in Section 3.A.3.

3.A.1 Basic notations and definitions

We denote by

$$\begin{array}{rccc} \sigma: & \mathbb{R} & \longrightarrow & \mathbb{R} \\ & t & \longmapsto & \max(t,0) \end{array}$$

the ReLU activation function. If $x = (x_1, \dots, x_m)^T \in \mathbb{R}^m$ is a vector, we denote $\sigma(x) = (\sigma(x_1), \dots, \sigma(x_m))^T$.

If $A \subset \mathbb{R}^m$, we denote by $\mathring{A}$ the interior of A and $\overline{A}$ the closure of A with respect to the standard topology of $\mathbb{R}^m$. We denote by $\partial A = \overline{A} \backslash \mathring{A}$ the topological boundary of A.

For $m, n \in \mathbb{N}^*$, we denote by $\mathbb{R}^n$ the vector space of n-dimensional real vectors and $\mathbb{R}^{m\times n}$ the vector space of real matrices with m lines and n columns. On the space of vectors, we use the norm $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$. For $x \in \mathbb{R}^n$ and $r > 0$, we denote $B(x, r) = \{y \in \mathbb{R}^n, \|y - x\| < r\}$.

For any vector $x \in \mathbb{R}^n$ whose coefficients x_i are all different from zero, we denote by x^{-1} or $\frac{1}{x}$ the vector $\left(\frac{1}{x_1}, \frac{1}{x_2}, \dots, \frac{1}{x_n}\right)^T$.

For any matrix $M \in \mathbb{R}^{m\times n}$, for all $i \in [\![1, m]\!]$, we denote by $M_{i,.}$ the i^{th} line of M. The vector $M_{i,.}$ is a line vector whose j^{th} component is $M_{i,j}$. Similarly, for $j \in [\![1, n]\!]$, we denote by $M_{.,j}$ the j^{th} column of M, which is the column vector whose i^{th} component is $M_{i,j}$. For any matrix $M \in \mathbb{R}^{m\times n}$, we denote by $M^T \in \mathbb{R}^{n\times m}$ the transpose matrix of M.

To avoid any confusion, we will denote by $(M^T)_{i,.}$ the i^{th} line of the matrix M^T and by ${M_{i,.}}^T$ the transpose of the line vector $M_{i,.}$, which is a column vector. Similarly, we will denote by $(M^T)_{.,j}$ the j^{th} column of M^T and ${M_{.,j}}^T$ the transpose of the column vector $M_{.,j}$.

For $n \in \mathbb{N}^*$, we denote by Id_n the $n \times n$ identity matrix and by $\mathbf{1}_n$ the vector $(1, 1, \dots, 1)^T \in \mathbb{R}^n$.

If $\lambda \in \mathbb{R}^n$ is a vector of size n, for some $n \in \mathbb{N}^*$, we denote by $\mathrm{Diag}(\lambda)$ the $n \times n$ matrix defined by:

$$\mathrm{Diag}(\lambda)_{i,j} = \begin{cases} \lambda_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

For any integer $m \in \mathbb{N}^*$, we denote by $\mathfrak{S}_m$ the set of all permutations of $[\![1, m]\!]$. We denote by $id_{[\![1,m]\!]}$ and $id_{\mathbb{R}^m}$ the identity functions on $[\![1, m]\!]$ and $\mathbb{R}^m$ respectively.

For any permutation $\varphi \in \mathfrak{S}_m$, we denote by P_φ the $m \times m$ permutation matrix associated to φ:

$$\forall i, j \in [\![1, m]\!], \quad (P_\varphi)_{i,j} = \begin{cases} 1 & \text{if } \varphi(j) = i \\ 0 & \text{otherwise.} \end{cases} \tag{3.A.1}$$

For all $x \in \mathbb{R}^m$, we have:

$$(P_\varphi x)_i = x_{\varphi^{-1}(i)}. \tag{3.A.2}$$

Using (3.A.2) we see that $P_{\varphi^{-1}} P_\varphi x = x$, which shows, since P_φ is orthogonal, that we have

$$P_\varphi^{-1} = P_{\varphi^{-1}} = P_\varphi^T. \tag{3.A.3}$$

Let $l, m, n \in \mathbb{N}^*$. For any matrix $M \in \mathbb{R}^{m \times l}$ and any function $f : \mathbb{R}^m \to \mathbb{R}^n$, we denote with a slight abuse of notation $f \circ M$ the function $x \mapsto f(Mx)$.

If X and Y are two sets and $h : X \to Y$ is a function, for a subset $A \subset Y$, we denote by $h^{-1}(A)$ the following set:

$$\{x \in X, h(x) \in A\}.$$

Note that this does not require the function h to be injective.

3.A.2 Continuous piecewise linear functions

We now introduce a few definitions and properties around the notion of continuous piecewise linear function.

Definition 25. Let $m \in \mathbb{N}^*$. A subset $D \subset \mathbb{R}^m$ is a closed polyhedron iif there exist $q \in \mathbb{N}^*$, $a_1, \ldots, a_q \in \mathbb{R}^m$ and $b_1, \ldots b_q \in \mathbb{R}$ such that for all $x \in \mathbb{R}^m$,

$$x \in D \quad \iff \quad \begin{cases} a_1^T x + b_1 \leq 0 \\ \vdots \\ a_q^T x + b_q \leq 0. \end{cases}$$

Remarks.
— A closed polyhedron is convex as an intersection of convex sets.
— Since we can fuse the inequation systems of several closed polyhedrons into one system, we see that an intersection of closed polyhedrons is a closed polyhedron.
— For $q = 1$ and $a_1 = 0$, taking $b_1 > 0$ and $b_1 \leq 0$ respectively we can show that $\emptyset$ and $\mathbb{R}^m$ are both closed polyhedra.

Proposition 26. *Let $m, l \in \mathbb{N}^*$. If $h : \mathbb{R}^l \to \mathbb{R}^m$ is linear and C is a closed polyhedron of $\mathbb{R}^m$, then $h^{-1}(C)$ is a closed polyhedron of $\mathbb{R}^l$.*

Proof. The function h is linear so there exist $M \in \mathbb{R}^{m\times l}$ and $b \in \mathbb{R}^m$ such that for all $x \in \mathbb{R}^l$,

$$h(x) = Mx + b.$$

The set C is a closed polyhedron so there exist $a_1, \dots, a_q \in \mathbb{R}^m$ and $b_1, \dots b_q \in \mathbb{R}$ such that $y \in C$ if and only if

$$\begin{cases} a_1^T y + b_1 \leq 0 \\ \vdots \\ a_q^T y + b_q \leq 0. \end{cases}$$

For all $x \in \mathbb{R}^l$,

$$\begin{aligned} x \in h^{-1}(C) &\iff h(x) \in C \\ &\iff \begin{cases} a_1^T(Mx+b) + b_1 \leq 0 \\ \vdots \\ a_q^T(Mx+b) + b_q \leq 0 \end{cases} \\ &\iff \begin{cases} (a_1^T M)x + (a_1^T b + b_1) \leq 0 \\ \vdots \\ (a_q^T M)x + (a_q^T b + b_q) \leq 0. \end{cases} \end{aligned}$$

This shows that $h^{-1}(C)$ is a closed polyhedron. □

Definition 27. We say that a function $g : \mathbb{R}^m \to \mathbb{R}^n$ is continuous piecewise linear if there exists a finite set of closed polyhedra whose union is $\mathbb{R}^m$ and such that g is linear over each polyhedron.

Example. Since $\mathbb{R}^m$ is a closed polyhedron, we see in particular that an affine function $x \mapsto Ax + b$, with $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$, is continuous piecewise linear from $\mathbb{R}^m$ to $\mathbb{R}^n$.

Example 28. The vectorial ReLU function $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ is continuous piecewise linear. Indeed, each of the 2^m closed orthants is a closed polyhedron, defined by a system of the form

$$\begin{cases} \epsilon_1 x_1 \geq 0 \\ \vdots \\ \epsilon_m x_m \geq 0, \end{cases}$$

with $\epsilon_i \in \{-1, 1\}$, and over such an orthant, the ReLU coincides with the affine function

$$(x_1, \dots, x_m) \mapsto \left(\frac{1+\epsilon_1}{2} x_1, \dots, \frac{1+\epsilon_m}{2} x_m \right).$$

In this definition the continuity is not obvious. We show it in the following proposition.

Proposition 29. *A continuous piecewise linear function is continuous.*

Proof. Let $g : \mathbb{R}^m \to \mathbb{R}^n$ be a continuous piecewise linear function. There exists a finite family of closed polyhedra $C_1, \dots, C_r$ such that $\bigcup_{i=1}^r C_i = \mathbb{R}^m$ and g is linear on each closed polyhedron C_i.

Let $x \in \mathbb{R}^m$. Let $\epsilon > 0$.

Let us denote $I = \{i \in [\![1, n]\!],\ x \in C_i\}$. Since the polyhedrons are closed, there exists $r_0 > 0$ such that for all $i \notin I, B(x, r_0) \cap C_i = \emptyset$. We thus have

$$B(x, r_0) = \bigcup_{i=1}^{m} (B(x, r_0) \cap C_i) = \bigcup_{i \in I} (B(x, r_0) \cap C_i)\,.$$

For all $i \in I$, g is linear -therefore continuous- on C_i so there exists $r_i > 0$, such that

$$y \in C_i \cap B(x, r_i) \ \Rightarrow\ \|g(y) - g(x)\| \leq \epsilon.$$

Let $r = \min(r_0, \min_{i \in I}(r_i))$. For all $y \in B(x, r)$ there exists $i \in I$ such that $y \in C_i$, and since $r \leq r_i$, we have

$$\|g(y) - g(x)\| \leq \epsilon.$$

Summarizing, for any $x \in \mathbb{R}^n$ and for any $\epsilon > 0$, there exists $r > 0$ such that

$$y \in B(x, r) \ \Rightarrow\ \|g(y) - g(x)\| \leq \epsilon.$$

This shows g is continuous. □

Proposition 30. *If $h : \mathbb{R}^l \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}^n$ are two continuous piecewise linear functions, then $g \circ h$ is continuous piecewise linear.*

Proof. By definition there exist a family $C_1, \dots, C_r$ of closed polyhedra of $\mathbb{R}^l$ such that $\bigcup_{i=1}^r C_i = \mathbb{R}^l$ and h is linear on each C_i and a family $D_1, \dots, D_s$ of closed polyhedra of $\mathbb{R}^m$ such that $\bigcup_{i=1}^s D_i = \mathbb{R}^m$ and g is linear on each D_i. Let $i \in [\![1, r]\!]$ and $j \in [\![1, s]\!]$. The function h coincides with a linear map $\tilde{h} : \mathbb{R}^l \to \mathbb{R}^m$ on C_i and the inverse image of a closed polyhedron by a linear map is a closed polyhedron (Proposition 26) so $\tilde{h}^{-1}(D_j)$ is a closed polyhedron. Thus $h^{-1}(D_j) \cap C_i = \tilde{h}^{-1}(D_j) \cap C_i$ is a closed polyhedron as an intersection of closed polyhedra. The function h is linear on C_i and g is linear on D_j so $g \circ h$ is linear on $h^{-1}(D_j) \cap C_i$. We have a family of closed polyhedra,

$$\left(h^{-1}(D_j) \cap C_i\right)_{\substack{i \in [\![1,r]\!] \\ j \in [\![1,s]\!]}},$$

each of which $g \circ h$ is linear over. Given that

$$\bigcup_{i=1}^{r} \bigcup_{j=1}^{s} h^{-1}(D_j) \cap C_i = \bigcup_{i=1}^{r} C_i = \mathbb{R}^l,$$

we can conclude that $g \circ h$ is continuous piecewise linear. □

Definition 31. Let $g : \mathbb{R}^m \to \mathbb{R}^n$ be a continuous piecewise linear function. Let Π be a set of closed polyhedra of $\mathbb{R}^m$. We say that Π is admissible with respect to the function g if and only if:

— $\bigcup_{D\in\Pi} D = \mathbb{R}^m$,
— for all $D \in \Pi$, g is linear on D,
— for all $D \in \Pi$, $\mathring{D} \neq \emptyset$.

Proposition 32. *For all $g : \mathbb{R}^m \to \mathbb{R}^n$ continuous piecewise linear, there exists a set of closed polyhedra Π admissible with respect to g.*

Proof. Let $g : \mathbb{R}^m \to \mathbb{R}^n$ be a continuous piecewise linear function. By definition there exists a finite set of closed polyhedra $D_1, \dots, D_s$ such that $\bigcup_{i=1}^s D_i = \mathbb{R}^m$ and g is linear on each D_i.

Let $I = \{i \in [\![1, s]\!], \mathring{D}_i \neq \emptyset\}$. Let us show that $\bigcup_{i\in I} D_i = \mathbb{R}^m$.

We first show that if a polyhedron D_i has empty interior, then it is contained in an affine hyperplane. Indeed, if it is not contained in an affine hyperplane, then there exist $m + 1$ affinely independent points $x_1, \dots, x_{m+1} \in D_i$. Since a closed polyhedron is convex, the convex hull of the points $\mathrm{Conv}(x_1, \dots, x_{m+1})$, which is a m-simplex, is contained in D_i, and thus D_i has nonempty interior.

Let $x \in \mathbb{R}^m$. For all $i \notin I$, D_i is contained in an affine hyperplane, and a finite union of affine hyperplanes does not contain any nontrivial ball. As a consequence, for all $n \in \mathbb{N}$, the ball $B(x, \frac{1}{n})$ is not contained in $\bigcup_{i\notin I} D_i$ and thus there exists $i_n \in I$ such that $D_{i_n} \cap B(x, \frac{1}{n}) \neq \emptyset$. Since I is finite, there exists $i \in I$ such that $i_n = i$ for infinitely many n, and thus $x \in \overline{D_i}$.

We have shown that for all $x \in \mathbb{R}^m$ there exists $i \in I$ such that $x \in \overline{D_i} = D_i$, which means that

$$\bigcup_{i\in I} D_i = \mathbb{R}^m.$$

Hence, the set $\Pi := \{D_i, i \in I\}$ is admissible with respect to g. □

Proposition 33. *Let $h : \mathbb{R}^l \to \mathbb{R}^m$ be a continuous piecewise linear function and let $\mathcal{P}$ be a finite set of closed polyhedra of $\mathbb{R}^m$. Then*

— *for all $D \in \mathcal{P}$, $h^{-1}(D)$ is a finite union of closed polyhedra;*
— *$\bigcup_{D\in\mathcal{P}} \partial h^{-1}(D)$ is contained in a finite union of hyperplanes $\bigcup_{k=1}^s A_k$.*

Proof. Consider Π an admissible set of closed polyhedra with respect to h. Let $D \in \mathcal{P}$. Since $\bigcup_{C\in\Pi} C = \mathbb{R}^l$, we can write

$$h^{-1}(D) = h^{-1}(D) \cap \left(\bigcup_{C\in\Pi} C \right) = \bigcup_{C\in\Pi} \left(h^{-1}(D) \cap C \right).$$

For all $C \in \Pi$, h is linear over C, so $h^{-1}(D) \cap C$ is a polyhedron (see Proposition 26). This shows the first point of the proposition.

Since $h^{-1}(D) \cap C$ is a polyhedron, $\partial\left(h^{-1}(D) \cap C\right)$ is contained in a finite union of hyperplanes. In topology, we have

$$\partial\left[\bigcup_{C\in\Pi}\left(h^{-1}(D)\cap C\right)\right] \subset \bigcup_{C\in\Pi}\partial\left(h^{-1}(D)\cap C\right),$$

which shows that $\partial\left[\bigcup_{C\in\Pi}\left(h^{-1}(D)\cap C\right)\right]$ i.e. $\partial h^{-1}(D)$ is contained in a finite union of hyperplanes too. This is true for any $D \in \mathcal{P}$, and since $\mathcal{P}$ is finite, this is also true of the union $\bigcup_{D\in\mathcal{P}} \partial h^{-1}(D)$. □

3.A.3 Neural networks

We consider fully connected feedforward neural networks, with ReLU activation function. We index the layers in reverse order, from K to 0, for some $K \geq 2$. The input layer is the layer K, the output layer is the layer 0, and between them are $K-1$ *hidden* layers. For $k \in [\![0, K]\!]$, we denote by $n_k \in \mathbb{N}$ the number of neurons of the layer k. This means the information contained at the layer k is a n_k-dimensional vector.

Let $k \in [\![0, K-1]\!]$. We denote the weights between the layer $k+1$ and the layer k with a matrix $M^k \in \mathbb{R}^{n_k \times n_{k+1}}$, and we consider a bias $b^k \in \mathbb{R}^{n_k}$ in the layer k. If $k \neq 0$, we add a ReLU activation function. If $x \in \mathbb{R}^{n_{k+1}}$ is the information contained at the layer $k+1$, the layer k contains:

$$\begin{cases} \sigma(M^k x + b^k) & \text{if } k \neq 0 \\ M^0 x + b^0 & \text{if } k = 0. \end{cases}$$

The parameters of the network can be summarized in the couple $(\mathbf{M}, \mathbf{b})$, where $\mathbf{M} = (M^0, M^1, \ldots, M^{K-1}) \in \mathbb{R}^{n_0 \times n_1} \times \cdots \times \mathbb{R}^{n_{K-1} \times n_K}$ and $\mathbf{b} = (b^0, b^1, \ldots, b^{K-1}) \in \mathbb{R}^{n_0} \times \cdots \times \mathbb{R}^{n_{K-1}}$. We formalize the transformation implemented by one layer of the network with the following definition.

Definition 34. For a network with parameters $(\mathbf{M}, \mathbf{b})$, we define the family of functions $(h_0, \ldots, h_{K-1})$ such that for all $k \in [\![0, K-1]\!]$, $h_k : \mathbb{R}^{n_{k+1}} \to \mathbb{R}^{n_k}$ and for all $x \in \mathbb{R}^{n_k}$,

$$h_k(x) = \begin{cases} \sigma(M^k x + b^k) & \text{if } k \neq 0 \\ M^0 x + b^0 & \text{if } k = 0. \end{cases}$$

The function implemented by the network is then

$$f_{\mathbf{M},\mathbf{b}} = h_0 \circ h_1 \circ \cdots \circ h_{K-1} : \mathbb{R}^{n_K} \longrightarrow \mathbb{R}^{n_0}. \tag{3.A.4}$$

The network with its parameters are represented in Figure 3.1 in the main part.

For all $l \in [\![0, K-1]\!]$, we denote

$$\mathbf{M}^{\leq l} = (M^0, M^1, \ldots, M^l)$$

and

$$\mathbf{b}^{\leq l} = (b^0, b^1, \ldots, b^l).$$

Remark 35. Since the vectorial ReLU function is continuous piecewise linear, Proposition 30 guarantees that the functions h_k are continuous piecewise linear.

We now define a few more functions associated to a network.

Definition 36. For a network with parameters $(\mathbf{M}, \mathbf{b})$, we define the family of functions $(h_0^{lin}, \dots, h_{K-1}^{lin})$ such that for all $k \in [\![0, K-1]\!]$, $h_k^{lin} : \mathbb{R}^{n_{k+1}} \to \mathbb{R}^{n_k}$ and for all $x \in \mathbb{R}^{n_{k+1}}$,

$$h_k^{lin}(x) = M^k x + b^k.$$

The functions h_k^{lin} correspond to the linear part of the transformation implemented by the network between two layers, before applying the activation σ.

Definition 37. For a network with parameters $(\mathbf{M}, \mathbf{b})$, we define the family of functions $(f_K, f_{K-1}, \dots, f_0)$ as follows:

— $f_K = id_{\mathbb{R}^{n_K}}$,

— for all $k \in [\![0, K-1]\!]$, $\quad f_k = h_k \circ h_{k+1} \circ \cdots \circ h_{K-1}$.

Remark. In particular we have $f_0 = f_{\mathbf{M},\mathbf{b}}$.

The function $f_k : \mathbb{R}^{n_K} \mapsto \mathbb{R}^{n_k}$ represents the transformation implemented by the network between the input layer and the layer k.

Definition 38. For a network with parameters $(\mathbf{M}, \mathbf{b})$, we define the sequence $(g_0, \dots, g_K)$ as follows:

— $g_0 = id_{\mathbb{R}^{n_0}}$,

— for all $k \in [\![1, K]\!]$, $\quad g_k = h_0 \circ h_1 \circ \cdots \circ h_{k-1}$.

Remark. We have in particular

— $g_K = f_{\mathbf{M},\mathbf{b}}$;

— for all $k \in [\![0, K]\!]$, $f_{\mathbf{M},\mathbf{b}} = g_k \circ f_k$.

The function $g_k : \mathbb{R}^{n_k} \mapsto \mathbb{R}^{n_0}$ represents the transformation implemented by the network between the layer k and the output layer.

In this paper the functions implemented by the networks are considered on a subset $\Omega \subset \mathbb{R}^{n_K}$. The successive layers of a network project this subset onto the spaces $\mathbb{R}^{n_k}$, inducing a subset Ω_k of $\mathbb{R}^{n_k}$ for all k, as in the following definition.

Definition 39. For a network with parameters $(\mathbf{M}, \mathbf{b})$, for any $\Omega \subset \mathbb{R}^{n_K}$, we denote for all $k \in [\![0, K]\!]$,

$$\Omega_k = f_k(\Omega).$$

Definition 40. For a network with parameters $(\mathbf{M}, \mathbf{b})$, for all $k \in [\![2, K]\!]$, for all $i \in [\![1, n_{k-1}]\!]$, we define

$$H_i^k = \{x \in \mathbb{R}^{n_k},\ M_{i,\cdot}^{k-1} x + b_i^{k-1} = 0\}.$$

Remark. When $M_{i,\cdot}^{k-1} \neq 0$, the set H_i^k is a hyperplane.

Remark 41. The objects defined in Definitions 34, 36, 37, 38, 39 and 40 all depend on $(\mathbf{M}, \mathbf{b})$, but to simplify the notation we do not write it explicitly. To disambiguate when manipulating a second network, whose parameters we will denote by $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, we will denote by $\tilde{h}_k$, $\tilde{h}_k^{lin}$, $\tilde{f}_k$, $\tilde{g}_k$, $\tilde{\Omega}_k$ and $\tilde{H}_i^k$ the corresponding objects.

Proposition 42. *For all $k \in [\![0, K]\!]$, f_k and g_k are continuous piecewise linear.*

Proof. We show this by induction: for the initialisation we have $f_K = id_{\mathbb{R}^{n_K}}$ which is continuous piecewise linear. Now let $k \in [\![0, K-1]\!]$ and assume f_{k+1} is continuous piecewise linear. By definition, we have $f_k = h_k \circ f_{k+1}$. The function h_k is continuous piecewise linear as noted in Remark 35. By Proposition 30, the composition of two continuous piecewise linear functions is continuous piecewise linear, so f_k is continuous piecewise linear. The conclusion follows by induction.

We do the same for $(g_0, \dots, g_K)$ starting with g_0: first we have $g_0 = id_{\mathbb{R}^{n_0}}$ which is continuous piecewise linear, then for all $k \in [\![1, K]\!]$, we have $g_k = g_{k-1} \circ h_{k-1}$, and we conclude by composition of two continuous piecewise linear functions. □

Corollary 43. *The function $f_{\mathbf{M},\mathbf{b}}$ is continuous piecewise linear.*

Proof. It comes immediately from $f_{\mathbf{M},\mathbf{b}} = f_0$ and Proposition 42. □

Recall the definition of an admissible set with respect to a continuous piecewise linear function (Definition 31). Proposition 42 allows the following definition.

Definition 44. Consider a network parameterization $(\mathbf{M}, \mathbf{b})$, and the functions g_k associated to it. We say that a list of sets of closed polyhedra $\mathbf{\Pi} = (\Pi_1, \dots, \Pi_{K-1})$ is *admissible* with respect to $(\mathbf{M}, \mathbf{b})$ iif for all $k \in [\![1, K-1]\!]$, the set Π_k is admissible with respect to g_k.

Remark. For a list $\mathbf{\Pi} = (\Pi_1, \dots, \Pi_{K-1})$, for all $l \in [\![1, K-1]\!]$, we denote $\mathbf{\Pi}^{\leq l} = (\Pi_1, \dots, \Pi_l)$. If $\mathbf{\Pi}$ is admissible with respect to $(\mathbf{M}, \mathbf{b})$, then $\mathbf{\Pi}^{\leq l}$ is admissible with respect to $(\mathbf{M}^{\leq l}, \mathbf{b}^{\leq l})$.

Proposition 45. *For any network parameterization $(\mathbf{M}, \mathbf{b})$, there always exists a list of sets of closed polyhedra $\mathbf{\Pi}$ that is admissible with respect to $(\mathbf{M}, \mathbf{b})$.*

Proof. For all $k \in [\![1, K-1]\!]$, since g_k is continuous piecewise linear, Proposition 32 guarantees that there exists an admissible set of polyhedra Π_k with respect to g_k. We simply define $\mathbf{\Pi} = (\Pi_1, \dots, \Pi_{K-1})$. □

Definition 46. For a parameterization $(\mathbf{M}, \mathbf{b})$ and a list $\mathbf{\Pi}$ admissible with respect to $(\mathbf{M}, \mathbf{b})$, for all $k \in [\![1, K-1]\!]$, for all $D \in \Pi_k$, since g_k is linear over D and D has nonempty interior, we can define $V^k(D) \in \mathbb{R}^{n_0 \times n_k}$ and $c^k(D) \in \mathbb{R}^{n_0}$ as the unique couple that satisfies:

$$\forall x \in D, \quad g_k(x) = V^k(D)x + c^k(D).$$

We now introduce the equivalence relation between parameterizations, often referred to as *equivalence modulo permutation and positive rescaling.*

Definition 47 (Equivalent parameterizations)**.** If $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are two network parameterizations, we say that $(\mathbf{M}, \mathbf{b})$ is *equivalent modulo permutation and positive rescaling*, or simply *equivalent*, to $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, and we write $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, if and only if there exist:

— a family of permutations $\boldsymbol{\varphi} = (\varphi_0, \dots, \varphi_K) \in \mathfrak{S}_{n_0} \times \cdots \times \mathfrak{S}_{n_K}$, with $\varphi_0 = id_{[\![1,n_0]\!]}$ and $\varphi_K = id_{[\![1,n_K]\!]}$,

— a family of vectors $\boldsymbol{\lambda} = (\lambda^0, \lambda^1, \dots, \lambda^K) \in (\mathbb{R}_+^*)^{n_0} \times \cdots \times (\mathbb{R}_+^*)^{n_K}$, with $\lambda^0 = \mathbf{1}_{n_0}$ and $\lambda^K = \mathbf{1}_{n_K}$,

such that for all $k \in [\![0, K-1]\!]$,

$$\begin{cases} \tilde{M}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P_{\varphi_{k+1}}^{-1} \\ \tilde{b}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) b^k. \end{cases} \tag{3.A.5}$$

Remarks.

1. Recall that we denote by $\frac{1}{\lambda^{k+1}}$ the vector whose components are $\frac{1}{\lambda_i^{k+1}}$. Note that $\operatorname{Diag}(\lambda^{k+1})^{-1} = \operatorname{Diag}(\frac{1}{\lambda^{k+1}})$. Using (3.A.2), for all $k \in [\![0, K-1]\!]$, (3.A.5) means that for all $(i,j) \in [\![1, n_k]\!] \times [\![1, n_{k+1}]\!]$,

$$\tilde{M}_{i,j}^k = \frac{\lambda^k_{\varphi_k^{-1}(i)}}{\lambda^{k+1}_{\varphi_{k+1}^{-1}(j)}} M^k_{\varphi_k^{-1}(i), \varphi_{k+1}^{-1}(j)}$$

and

$$\tilde{b}_i^k = \lambda^k_{\varphi_k^{-1}(i)} b^k_{\varphi_k^{-1}(i)}.$$

2. We go from a parameterization to an equivalent one by:
   — permuting the neurons of each hidden layer k with a permutation φ_k;
   — for each hidden layer k, multiplying all the weights of the edges arriving (from the layer $k+1$) to the neuron j, as well as the bias b_j^k, by some positive number λ_j^k, and multiplying all the weights of the edges leaving (towards the layer $k-1$) the neuron j by $\frac{1}{\lambda_j^k}$.

Proposition 48. *The relation $\sim$ is an equivalence relation.*

Proof. Let us first show the following equality, that we are going to use in the proof. For any $n \in \mathbb{N}^*$, $\lambda \in \mathbb{R}^n$ and $\varphi \in \mathfrak{S}_n$,

$$\operatorname{Diag}(\lambda) P_\varphi = P_\varphi \operatorname{Diag}(P_\varphi^{-1} \lambda). \tag{3.A.6}$$

Indeed, $\operatorname{Diag}(\lambda) P_\varphi$ is the matrix obtained by multiplying each line i of P_φ by λ_i, so recalling (3.A.1), for all $i, j \in [\![1, m]\!]$, we have

$$(\operatorname{Diag}(\lambda) P_\varphi)_{i,j} = \begin{cases} \lambda_i & \text{if } \varphi(j) = i \\ 0 & \text{otherwise.} \end{cases}$$

At the same time, $P_\varphi \operatorname{Diag}(P_\varphi^{-1}\lambda)$ is the matrix obtained by multiplying each column j of P_φ by $(P_\varphi^{-1}\lambda)_j = \lambda_{\varphi(j)}$ (see (3.A.2) and (3.A.3)), so for all $i, j \in [\![1, m]\!]$, we have

$$(P_\varphi \operatorname{Diag}(P_\varphi^{-1}\lambda))_{i,j} = \begin{cases} \lambda_{\varphi(j)} & \text{if } \varphi(j) = i \\ 0 & \text{otherwise.} \end{cases}$$

The two matrices are clearly equal.

We can now show the proposition.

— To show reflexivity we can take $\lambda^k = \mathbf{1}_{n_k}$ and $\varphi_k = id_{[\![1,n_k]\!]}$ for all $k \in [\![0, K]\!]$.

— Let us show symmetry. Assume a parameterization $(\mathbf{M}, \mathbf{b})$ is equivalent to another parameterization $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$. Let us denote by $\boldsymbol{\varphi}$ and $\boldsymbol{\lambda}$ the corresponding families of permutations and vectors, as in Definition 47. Inverting the expression of $\tilde{M}^k$ in Definition 47 and using (3.A.6) twice, we have for all $k \in [\![0, K-1]\!]$:

$$\begin{aligned} & \tilde{M}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P_{\varphi_{k+1}}^{-1} \\ \iff \quad & \operatorname{Diag}(\lambda^k)^{-1} P_{\varphi_k}^{-1} \tilde{M}^k P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) = M^k \\ \iff \quad & P_{\varphi_k}^{-1} \operatorname{Diag}(P_{\varphi_k}\lambda^k)^{-1} \tilde{M}^k \operatorname{Diag}(P_{\varphi_{k+1}}\lambda^{k+1}) P_{\varphi_{k+1}} = M^k, \end{aligned}$$

so denoting $\tilde{\varphi}_k = \varphi_k^{-1}$ and $\tilde{\lambda}^k = (P_{\varphi_k}\lambda^k)^{-1}$, and recalling that $P_{\varphi_k^{-1}} = P_{\varphi_k}^{-1}$, we have, for all $k \in [\![0, K-1]\!]$,

$$M^k = P_{\tilde{\varphi}_k} \operatorname{Diag}(\tilde{\lambda}^k) \tilde{M}^k \operatorname{Diag}(\tilde{\lambda}^{k+1})^{-1} P_{\tilde{\varphi}_{k+1}}^{-1}.$$

We show similarly that for all $k \in [\![0, K-1]\!]$,

$$b^k = P_{\tilde{\varphi}_k} \operatorname{Diag}(\tilde{\lambda}_k) \tilde{b}^k.$$

We naturally have $\tilde{\varphi}_0 = id_{[\![1,n_0]\!]}$ and $\tilde{\varphi}_K = id_{[\![1,n_K]\!]}$, as well as $\tilde{\lambda}^0 = \mathbf{1}_{n_0}$ and $\tilde{\lambda}^K = \mathbf{1}_{n_K}$.

This proves the symmetry of the relation.

— Let us show transitivity. Assume $(\mathbf{M}, \mathbf{b})$, $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ and $(\check{\mathbf{M}}, \check{\mathbf{b}})$ are three parameterizations such that $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}) \sim (\check{\mathbf{M}}, \check{\mathbf{b}})$.

As in Definition 47, we denote by $\boldsymbol{\varphi}$, $\tilde{\boldsymbol{\varphi}}$, $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{\lambda}}$ the families of permutations and vectors such that, for all $k \in [\![0, K-1]\!]$,

$$\begin{cases} \tilde{M}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P_{\varphi_{k+1}}^{-1} \\ \tilde{b}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) b^k, \end{cases}$$

and

$$\begin{cases} \check{M}^k = P_{\tilde{\varphi}_k} \operatorname{Diag}(\tilde{\lambda}^k) \tilde{M}^k \operatorname{Diag}(\tilde{\lambda}^{k+1})^{-1} P_{\tilde{\varphi}_{k+1}}^{-1} \\ \check{b}^k = P_{\tilde{\varphi}_k} \operatorname{Diag}(\tilde{\lambda}^k) \tilde{b}^k. \end{cases}$$

Combining these and using (3.A.6), we have

$$\begin{aligned}
\check{M}^k &= P_{\tilde{\varphi}_k} \operatorname{Diag}(\tilde{\lambda}^k) P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P_{\varphi_{k+1}}^{-1} \operatorname{Diag}(\tilde{\lambda}^{k+1})^{-1} P_{\tilde{\varphi}_{k+1}}^{-1} \\
&= P_{\tilde{\varphi}_k} \left(\operatorname{Diag}(\tilde{\lambda}^k) P_{\varphi_k} \right) \operatorname{Diag}(\lambda^k) M^k \\
&\qquad \cdot \operatorname{Diag}(\lambda^{k+1})^{-1} \left(\operatorname{Diag}(\tilde{\lambda}^{k+1}) P_{\varphi_{k+1}} \right)^{-1} P_{\tilde{\varphi}_{k+1}}^{-1} \\
&= P_{\tilde{\varphi}_k} \left(P_{\varphi_k} \operatorname{Diag}(P_{\varphi_k}^{-1} \tilde{\lambda}^k) \right) \operatorname{Diag}(\lambda^k) M^k \\
&\qquad \cdot \operatorname{Diag}(\lambda^{k+1})^{-1} \left(P_{\varphi_{k+1}} \operatorname{Diag}(P_{\varphi_{k+1}}^{-1} \tilde{\lambda}^{k+1}) \right)^{-1} P_{\tilde{\varphi}_{k+1}}^{-1} \\
&= P_{\tilde{\varphi}_k} P_{\varphi_k} \operatorname{Diag}(P_{\varphi_k}^{-1} \tilde{\lambda}^k) \operatorname{Diag}(\lambda^k) M^k \\
&\qquad \cdot \operatorname{Diag}(\lambda^{k+1})^{-1} \operatorname{Diag}(P_{\varphi_{k+1}}^{-1} \tilde{\lambda}^{k+1})^{-1} P_{\varphi_{k+1}}^{-1} P_{\tilde{\varphi}_{k+1}}^{-1},
\end{aligned}$$

and

$$\begin{aligned}
\check{b}^k &= P_{\tilde{\varphi}_k} \operatorname{Diag}(\tilde{\lambda}^k) P_{\varphi_k} \operatorname{Diag}(\lambda^k) b^k \\
&= P_{\tilde{\varphi}_k} P_{\varphi_k} \operatorname{Diag}(P_{\varphi_k}^{-1} \tilde{\lambda}^k) \operatorname{Diag}(\lambda^k) b^k.
\end{aligned}$$

Hence denoting $\check{\varphi}_k = \tilde{\varphi}_k \circ \varphi_k$ and $\check{\lambda}^k = \operatorname{Diag}(P_{\varphi_k}^{-1} \tilde{\lambda}^k) \lambda^k$, for all $k \in [\![0, K]\!]$, we see that, for $k \in [\![0, K-1]\!]$,

$$\check{M}^k = P_{\check{\varphi}_k} \operatorname{Diag}(\check{\lambda}^k) M^k \operatorname{Diag}(\check{\lambda}^{k+1})^{-1} P_{\check{\varphi}_{k+1}}^{-1}$$

and

$$\check{b}^k = P_{\check{\varphi}_k} \operatorname{Diag}(\check{\lambda}^k) b^k.$$

Naturally, we also have $\check{\varphi}_0 = id_{[\![1,n_0]\!]}$ and $\check{\varphi}_K = id_{[\![1,n_K]\!]}$, as well as $\check{\lambda}^0 = \mathbf{1}_{n_0}$ and $\check{\lambda}^K = \mathbf{1}_{n_K}$, which shows that $(\mathbf{M}, \mathbf{b}) \sim (\check{\mathbf{M}}, \check{\mathbf{b}})$.

□

Recall the objects $h_k, f_k, g_k, \Omega_k, H_i^k$ associated to a parameterization $(\mathbf{M}, \mathbf{b})$, defined in Definitions 34, 37, 38, 39 and 40, and recall that we denote by $\tilde{h}_k, \tilde{f}_k, \tilde{g}_k, \tilde{\Omega}_k$ and $\tilde{H}_i^k$ the corresponding objects with respect to another parameterization $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$. We give in the following proposition the relations that link these objects when the two parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are equivalent.

Proposition 49. *Assume* $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ *and consider* $\boldsymbol{\varphi}$ *and* $\boldsymbol{\lambda}$ *as in Definition 47. Let* $\mathbf{\Pi}$ *be a list of sets of closed polyhedra that is admissible with respect to* $(\mathbf{M}, \mathbf{b})$. *Then:*

1. *for all* $k \in [\![0, K-1]\!]$,

$$\tilde{h}_k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) \circ h_k \circ \operatorname{Diag}(\lambda^{k+1})^{-1} P_{\varphi_{k+1}}^{-1},$$

2. *for all* $k \in [\![0, K]\!]$,

$$\begin{aligned}
\tilde{f}_k &= P_{\varphi_k} \operatorname{Diag}(\lambda^k) \circ f_k, \\
\tilde{g}_k &= g_k \circ \operatorname{Diag}(\lambda^k)^{-1} P_{\varphi_k}^{-1}, \\
\tilde{\Omega}_k &= P_{\varphi_k} \operatorname{Diag}(\lambda^k) \Omega_k,
\end{aligned} \tag{3.A.7}$$

3. *for all* $k \in [\![2, K]\!]$, *for all* $i \in [\![1, n_{k-1}]\!]$,

$$\tilde{H}_i^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) H^k_{\varphi_{k-1}^{-1}(i)},$$

4. *for all* $k \in [\![1, K-1]\!]$, *the set of closed polyhedra* $\tilde{\Pi}_k = \{P_{\varphi_k} \operatorname{Diag}(\lambda^k) D, D \in \Pi_k\}$ *is admissible for* $\tilde{g}_k$, *i.e. the list* $\tilde{\mathbf{\Pi}} = (\tilde{\Pi}_1, \dots, \tilde{\Pi}_{K-1})$ *is admissible with respect to* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$.

Proof. 1. Let $k \in [\![0, K-1]\!]$. If $k \neq 0$, we have from Definition 34:

$$\begin{aligned}
\tilde{h}_k(x) &= \sigma(\tilde{M}^k x + \tilde{b}^k) \\
&= \sigma\left(P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} x \right. \\
&\qquad \left. + P_{\varphi_k} \operatorname{Diag}(\lambda^k) b^k\right) \\
&= \sigma\left(P_{\varphi_k} \operatorname{Diag}(\lambda^k) \left[M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} x + b^k\right]\right).
\end{aligned}$$

Denote $y := \left[M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} x + b^k\right]$. Let $i \in [\![1, n_k]\!]$. Using (3.A.2) and the fact that $\lambda^k_{\varphi_k^{-1}(i)}$ is nonnegative, the i^{th} coordinate of $\tilde{h}_k(x)$ is

$$\begin{aligned}
\tilde{h}_k(x)_i = \left[\sigma\left(P_{\varphi_k} \operatorname{Diag}(\lambda^k) y\right)\right]_i &= \sigma\left(\left[P_{\varphi_k} \operatorname{Diag}(\lambda^k) y\right]_i\right) \\
&= \sigma\left(\lambda^k_{\varphi_k^{-1}(i)} y_{\varphi_k^{-1}(i)}\right) \\
&= \lambda^k_{\varphi_k^{-1}(i)} \sigma\left(y_{\varphi_k^{-1}(i)}\right) \\
&= \left[P_{\varphi_k} \operatorname{Diag}(\lambda^k) \sigma(y)\right]_i.
\end{aligned}$$

Finally, we find the expression of $\tilde{h}_k(x)$:

$$\begin{aligned}
\tilde{h}_k(x) &= P_{\varphi_k} \operatorname{Diag}(\lambda^k) \sigma(y) \\
&= P_{\varphi_k} \operatorname{Diag}(\lambda^k) \sigma\left(M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} x + b^k\right) \\
&= P_{\varphi_k} \operatorname{Diag}(\lambda^k) h_k\left(\operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}}(x)\right).
\end{aligned}$$

This concludes the proof when $k \neq 0$.

The case $k = 0$ is proven similarly but replacing the ReLU function σ by the identity.

2. — We prove by induction the expression of $\tilde{f}_k$.

For $k = K$, we have $\tilde{f}_K = f_K = id_{\mathbb{R}^{n_K}}$, and since $P_{\varphi_K} = \mathrm{Id}_{n_K}$ and $\lambda^K = \mathbf{1}_{n_K}$ the equality $\tilde{f}_K = P_{\varphi_K} \operatorname{Diag}(\lambda^K) f_K$ holds.

Now let $k \in [\![0, K-1]\!]$. Suppose the induction hypothesis is true for $\tilde{f}_{k+1}$. Using the expression of $\tilde{h}_k$ we just proved in 1 and the induction

hypothesis, we have

$$\begin{aligned}\tilde{f}_k &= \tilde{h}_k \circ \tilde{f}_{k+1}\\ &= \left(P_{\varphi_k}\operatorname{Diag}(\lambda^k)\circ h_k \circ \operatorname{Diag}(\lambda^{k+1})^{-1}P^{-1}_{\varphi_{k+1}}\right)\circ\left(P_{\varphi_{k+1}}\operatorname{Diag}(\lambda^{k+1})\circ f_{k+1}\right)\\ &= P_{\varphi_k}\operatorname{Diag}(\lambda^k)\circ h_k \circ f_{k+1}\\ &= P_{\varphi_k}\operatorname{Diag}(\lambda^k)\circ f_k.\end{aligned}$$

This concludes the induction.

— We prove similarly the expression of $\tilde{g}_k$, but starting from $k=0$: first we have $\tilde{g}_0 = g_0 = id_{\mathbb{R}^{n_0}}$, and then, for $k \in [\![0, K-1]\!]$, we write $\tilde{g}_{k+1} = \tilde{g}_k \circ \tilde{h}_k$ and we use the induction hypothesis and the expression of $\tilde{h}_k$.

— Using the relation (3.A.7), that we just proved, we obtain

$$\tilde{\Omega}_k = \tilde{f}_k(\Omega) = P_{\varphi_k}\operatorname{Diag}(\lambda^k) f_k(\Omega) = P_{\varphi_k}\operatorname{Diag}(\lambda^k)\Omega_k.$$

3. Let $k \in [\![2, K]\!]$ and $i \in [\![1, n_{k-1}]\!]$. For all $x \in \mathbb{R}^{n_k}$, using (3.A.5) and (3.A.2),

$$\begin{aligned}x \in \tilde{H}^k_i \quad &\iff \quad \tilde{M}^{k-1}_{i,\cdot}x + \tilde{b}^{k-1}_i = 0\\ &\iff \quad \left[P_{\varphi_{k-1}}\operatorname{Diag}(\lambda^{k-1})M^{k-1}\operatorname{Diag}(\lambda^k)^{-1}P^{-1}_{\varphi_k}\right]_{i,\cdot}x\\ &\qquad + \left[P_{\varphi_{k-1}}\operatorname{Diag}(\lambda^{k-1})b^{k-1}\right]_i = 0\\ &\iff \quad \lambda^{k-1}_{\varphi^{-1}_{k-1}(i)} M^{k-1}_{\varphi^{-1}_{k-1}(i),\cdot}\operatorname{Diag}(\lambda^k)^{-1}P^{-1}_{\varphi_k}x + \lambda^{k-1}_{\varphi^{-1}_{k-1}(i)} b^{k-1}_{\varphi^{-1}_{k-1}(i)} = 0\\ &\iff \quad \lambda^{k-1}_{\varphi^{-1}_{k-1}(i)}\left(M^{k-1}_{\varphi^{-1}_{k-1}(i),\cdot}\operatorname{Diag}(\lambda^k)^{-1}P^{-1}_{\varphi_k}x + b^{k-1}_{\varphi^{-1}_{k-1}(i)}\right) = 0\\ &\iff \quad M^{k-1}_{\varphi^{-1}_{k-1}(i),\cdot}\operatorname{Diag}(\lambda^k)^{-1}P^{-1}_{\varphi_k}x + b^{k-1}_{\varphi^{-1}_{k-1}(i)} = 0\\ &\iff \quad \operatorname{Diag}(\lambda^k)^{-1}P^{-1}_{\varphi_k}x \in H^k_{\varphi^{-1}_{k-1}(i)}.\end{aligned}$$

Thus, $\tilde{H}^k_i = P_{\varphi_k}\operatorname{Diag}(\lambda^k)H^k_{\varphi^{-1}_{k-1}(i)}$.

4. For all $D \in \Pi_k$, denote $\tilde{D} = P_{\varphi_k}\operatorname{Diag}(\lambda^k)D$. We have $\tilde{\Pi}_k = \{\tilde{D}, D \in \Pi_k\}$.

Let $D \in \Pi_k$. The matrix $P_{\varphi_k}\operatorname{Diag}(\lambda^k)$ is invertible so, according to Proposition 26, $\tilde{D} = P_{\varphi_k}\operatorname{Diag}(\lambda^k)D$ is a closed polyhedron, and since $\mathring{D} \neq \emptyset$ we also have $\mathring{\tilde{D}} \neq \emptyset$.

Now recall from Item 2 that:

$$\tilde{g}_k = g_k \circ \operatorname{Diag}(\lambda^k)^{-1}P^{-1}_{\varphi_k}.$$

For all $x \in \tilde{D}$, we have $\operatorname{Diag}(\lambda^k)^{-1}P^{-1}_{\varphi_k}x \in D$. Since Π_k is admissible with respect to g_k (by definition of $\mathbf{\Pi}$), g_k is linear on D, and thus the function $\tilde{g}_k$ is linear on $\tilde{D}$.

Again, since Π_k is admissible with respect to g_k, we have $\bigcup_{D\in\Pi_k} D = \mathbb{R}^m$, and thus

$$\begin{aligned}\bigcup_{\tilde{D}\in\tilde{\Pi}_k} \tilde{D} &= \bigcup_{D\in\Pi_k} P_{\varphi_k}\operatorname{Diag}(\lambda^k)D \\ &= P_{\varphi_k}\operatorname{Diag}(\lambda^k)\left(\bigcup_{D\in\Pi_k} D\right) \\ &= P_{\varphi_k}\operatorname{Diag}(\lambda^k)\left(\mathbb{R}^m\right) \\ &= \mathbb{R}^m,\end{aligned}$$

which shows that $\tilde{\Pi}_k$ is admissible with respect to $\tilde{g}_k$.

This being true for any $k \in [\![1, K-1]\!]$, we conclude that $\tilde{\mathbf{\Pi}}$ is admissible with respect to $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$.

□

Corollary 50. *If* $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, *then* $f_{\mathbf{M},\mathbf{b}} = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$.

Proof. Consider $\boldsymbol{\varphi}$ and $\boldsymbol{\lambda}$ as in Definition 47. Looking at (3.A.7) for $k = 0$, and using the fact that $f_0 = f_{\mathbf{M},\mathbf{b}}$ and $\tilde{f}_0 = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$, we obtain from Proposition 49

$$f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}} = P_{\varphi_0}\operatorname{Diag}(\lambda^0) f_{\mathbf{M},\mathbf{b}}.$$

By definition of $\boldsymbol{\varphi}$ and $\boldsymbol{\lambda}$, we have $P_{\varphi_0} = \mathrm{Id}_{n_0}$ and $\lambda^0 = \mathbf{1}_{n_0}$, so we can finally conclude:

$$f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}} = f_{\mathbf{M},\mathbf{b}}.$$

□

Definition 51. We say that $(\mathbf{M}, \mathbf{b})$ is normalized if for all $k \in [\![1, K-1]\!]$, for all $i \in [\![1, n_k]\!]$, we have:

$$\|M^k_{i,\cdot}\| = 1.$$

Proposition 52. *If* $(\mathbf{M}, \mathbf{b})$ *satisfies, for all* $k \in [\![1, K-1]\!]$, *for all* $i \in [\![1, n_k]\!]$, $M^k_{i,\cdot} \neq 0$, *then there exists an equivalent parameterization* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ *that is normalized.*

Proof. We define recursively the family $(\lambda^0, \lambda^1, \ldots, \lambda^K) \in (\mathbb{R}^*_+)^{n_0} \times \cdots \times (\mathbb{R}^*_+)^{n_K}$ by:

— $\lambda^K = \mathbf{1}_{n_K}$;

— for all $k \in [\![1, K-1]\!]$, for all $i \in [\![1, n_k]\!]$,

$$\lambda^k_i = \frac{1}{\|M^k_{i,\cdot}\operatorname{Diag}(\lambda^{k+1})^{-1}\|};$$

— $\lambda^0 = \mathbf{1}_{n_0}$.

Consider the parameterization $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ defined by, for all $k \in [\![0, K-1]\!]$:

$$\begin{cases} \tilde{M}^k = \mathrm{Diag}(\lambda^k) M^k \mathrm{Diag}(\lambda^{k+1})^{-1} \\ \tilde{b}^k = \mathrm{Diag}(\lambda^k) b^k. \end{cases}$$

The parameterization is, by definition, equivalent to $(\mathbf{M}, \mathbf{b})$, and, for all $k \in [\![1, K-1]\!]$, for all $i \in [\![1, n_k]\!]$:

$$\begin{aligned} \|\tilde{M}^k_{i,\cdot}\| &= \left\| \left[\mathrm{Diag}(\lambda^k) M^k \mathrm{Diag}(\lambda^{k+1})^{-1} \right]_{i,\cdot} \right\| \\ &= \left\| \lambda^k_i M^k_{i,\cdot} \mathrm{Diag}(\lambda^{k+1})^{-1} \right\| \\ &= \left\| \frac{1}{\| M^k_{i,\cdot} \mathrm{Diag}(\lambda^{k+1})^{-1} \|} M^k_{i,\cdot} \mathrm{Diag}(\lambda^{k+1})^{-1} \right\| \\ &= 1. \end{aligned}$$

□

Proposition 53. *If* $(\mathbf{M}, \mathbf{b})$ *and* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ *are both normalized, then they are equivalent if and only if there exists a family of permutations* $(\varphi_0, \dots, \varphi_K) \in \mathfrak{S}_{n_0} \times \cdots \times \mathfrak{S}_{n_K}$*, with* $\varphi_0 = id_{[\![1,n_0]\!]}$ *and* $\varphi_K = id_{[\![1,n_K]\!]}$*, such that for all* $k \in [\![0, K-1]\!]$*:*

$$\begin{cases} \tilde{M}^k = P_{\varphi_k} M^k P^{-1}_{\varphi_{k+1}} \\ \tilde{b}^k = P_{\varphi_k} b^k. \end{cases} \tag{3.A.8}$$

Proof. Assume $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are equivalent. Then there exist a family of permutations $(\varphi_0, \dots, \varphi_K) \in \mathfrak{S}_{n_0} \times \cdots \times \mathfrak{S}_{n_K}$ and a family $(\lambda^0, \dots, \lambda^K) \in (\mathbb{R}_+^*)^{n_0} \times \cdots \times (\mathbb{R}_+^*)^{n_K}$ as in Definition 47.

Let us prove by induction that $\lambda^k = \mathbf{1}_{n_k}$ for all $k \in [\![0, K]\!]$.

For $k = K$ it is true by Definition 47.

Let $k \in [\![1, K-1]\!]$, and suppose $\lambda^{k+1} = \mathbf{1}_{n_{k+1}}$. This means $\mathrm{Diag}(\lambda^{k+1}) = \mathrm{Id}_{n_{k+1}}$. Let $i \in [\![1, n_k]\!]$. Since $(\mathbf{M}, \mathbf{b})$ is normalized, $\|M^k_{i,\cdot}\| = 1$. Since $P^{-1}_{\varphi_{k+1}}$ is a permutation matrix, it is orthogonal so $\|M^k_{i,\cdot} P^{-1}_{\varphi_{k+1}}\| = \|M^k_{i,\cdot}\| = 1$. Recalling (3.A.5) and using the fact that $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ is normalized, that $\mathrm{Diag}(\lambda^{k+1}) = \mathrm{Id}_{n_{k+1}}$ and that λ^k_i is positive, we have:

$$\begin{aligned} 1 = \|\tilde{M}^k_{\varphi_k(i),\cdot}\| &= \|\lambda^k_i M^k_{i,\cdot} \mathrm{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}}\| \\ &= \lambda^k_i \|M^k_{i,\cdot} P^{-1}_{\varphi_{k+1}}\| \\ &= \lambda^k_i. \end{aligned}$$

This shows $\lambda^k = \mathbf{1}_{n_k}$.

The case $k = 0$ is also true by Definition 47.

Equation (3.A.5) with $\lambda^k = \mathbf{1}_{n_k}$ for all $k \in [\![0, K]\!]$ is precisely equation (3.A.8).

The reciprocal is clear: (3.A.8) is a particular case of (3.A.5) with $\lambda^k = \mathbf{1}_{n_k}$. □

3.B Main theorem

In Appendix 3.B, we prove the main theorem using the notations and results of Appendix 3.A, and admitting Lemma 63, which is proven in Appendix 3.C.

More precisely, we begin by stating the conditions **C** and **P** in Section 3.B.1, we then state our main result, which is Theorem 61, in Section 3.B.2, and we give a consequence of this result in terms of risk minimization, which is Corollary 62, in Section 3.B.3. Finally we prove Theorem 61 and Corollary 62 in Sections 3.B.4 and 3.B.5 respectively.

3.B.1 Conditions

Assume $g : \mathbb{R}^m \to \mathbb{R}^n$ is a continuous piecewise linear function, Π is a set of closed polyhedra admissible with respect to g, and let $\Omega \subset \mathbb{R}^l$, $M \in \mathbb{R}^{m\times l}$ and $b \in \mathbb{R}^m$.

We define

$$\begin{array}{rccc} h: & \mathbb{R}^l & \longrightarrow & \mathbb{R}^m \\ & x & \longmapsto & \sigma(Mx+b) \end{array}$$

and

$$\begin{array}{rccc} h^{lin}: & \mathbb{R}^l & \longrightarrow & \mathbb{R}^m \\ & x & \longmapsto & Mx+b. \end{array}$$

Definition 54. For all $i \in [\![1, m]\!]$, we denote $E_i = \{x \in \mathbb{R}^m,\ x_i = 0\}$.

Definition 55. Let $D \in \Pi$. The function g coincides with a linear function on D. Since the interior of D is nonempty, we define $V(D) \in \mathbb{R}^{n\times m}$ and $c(D) \in \mathbb{R}^n$ as the unique couple satisfying, for all $x \in D$:

$$g(x) = V(D)x + c(D).$$

Definition 56. We say that (g, M, b, Ω, Π) satisfies the conditions **C** iif:

C.*a*) M is full row rank;

C.*b*) for all $i \in [\![1, m]\!]$, there exists $x \in \mathring{\Omega}$ such that

$$M_{i,.}x + b_i = 0,$$

or equivalently,

$$E_i \cap h^{lin}(\mathring{\Omega}) \neq \emptyset;$$

C.*c*) for all $D \in \Pi$, for all $i \in [\![1, m]\!]$, if $E_i \cap D \cap h(\Omega) \neq \emptyset$ then $V_{.,i}(D) \neq 0$;

C.*d*) for any affine hyperplane $H \subset \mathbb{R}^l$,

$$H \cap \mathring{\Omega} \not\subset \bigcup_{D\in\Pi} \partial h^{-1}(D).$$

Definition 57. For all $k \in [\![1, K-1]\!]$, for all $i \in [\![1, n_k]\!]$, we denote $E_i^k = \{x \in \mathbb{R}^{n_k}, x_i = 0\}$.

We now state the conditions **P** (already stated in the main text in Definition 15).

Definition 58. We say that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ satisfies the conditions **P** iif for all $k \in [\![1, K-1]\!]$, $(g_k, M^k, b^k, \Omega_{k+1}, \Pi_k)$ satisfies the conditions **C**.

Explicitly, for all $k \in [\![1, K-1]\!]$, the conditions are the following:

P.*a*) M^k is full row rank;

P.*b*) for all $i \in [\![1, n_k]\!]$, there exists $x \in \mathring{\Omega}_{k+1}$ such that

$$M^k_{i,\cdot} x + b^k_i = 0,$$

or equivalently

$$E^k_i \cap h^{lin}_k(\mathring{\Omega}_{k+1}) \neq \emptyset;$$

P.*c*) for all $D \in \Pi_k$, for all $i \in [\![1, n_k]\!]$, if $E^k_i \cap D \cap \Omega_k \neq \emptyset$ then $V^k_{\cdot,i}(D) \neq 0$;

P.*d*) for any affine hyperplane $H \subset \mathbb{R}^{n_{k+1}}$,

$$H \cap \mathring{\Omega}_{k+1} \not\subset \bigcup_{D \in \Pi_k} \partial h_k^{-1}(D).$$

Remark 59. The condition **P**.*b*) implies that for all $k \in [\![1, K-1]\!]$, $\mathring{\Omega}_{k+1} \neq \emptyset$, and in particular for $k = K-1$, the set $\Omega = \Omega_K$ has nonempty interior.

The following proposition shows that the conditions **P** are stable modulo permutation and positive rescaling, as defined in Definition 47.

Proposition 60. *Suppose* $(\mathbf{M}, \mathbf{b})$ *and* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ *are two equivalent network parameterizations, and suppose* $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ *satisfies the conditions* **P**. *Then, if we define* $\tilde{\mathbf{\Pi}}$ *as in Item 4 of Proposition 49,* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ *satisfies the conditions* **P**.

Proof. Since $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are equivalent, by Definition 47 there exist

— a family of permutations $(\varphi_0, \dots, \varphi_K) \in \mathfrak{S}_{n_0} \times \cdots \times \mathfrak{S}_{n_K}$, with $\varphi_0 = id_{[\![1,n_0]\!]}$ and $\varphi_K = id_{[\![1,n_K]\!]}$,

— a family $(\lambda^0, \lambda^1, \dots, \lambda^K) \in (\mathbb{R}^*_+)^{n_0} \times \cdots \times (\mathbb{R}^*_+)^{n_K}$, with $\lambda^0 = \mathbf{1}_{n_0}$ and $\lambda^K = \mathbf{1}_{n_K}$,

such that

$$\begin{cases} \tilde{M}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} \\ \tilde{b}^k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) b^k. \end{cases} \tag{3.B.1}$$

Let $k \in [\![1, K-1]\!]$. We know the conditions **P**.*a*) – **P**.*d*) are satisfied by $(g_k, M^k, b^k, \Omega_{k+1}, \Pi_k)$, let us show they are satisfied by $(\tilde{g}_k, \tilde{M}^k, \tilde{b}^k, \tilde{\Omega}_{k+1}, \tilde{\Pi}_k)$.

P.*a*) Since M^k satisfies **P**.*a*), it is full row rank, and using (3.B.1) and the fact that the matrices $P_{\varphi_k}, \operatorname{Diag}(\lambda^k), \operatorname{Diag}(\lambda^{k+1})^{-1}$ and $P^{-1}_{\varphi_{k+1}}$ are invertible, we see that $\tilde{M}^k$ is full row rank.

P.*b*) Let $i \in [\![1, n_k]\!]$. Since $(g_k, M^k, b^k, \Omega_{k+1}, \Pi_k)$ satisfies the condition **P**.*b*), we can choose $x \in \mathring{\Omega}_{k+1}$ such that

$$M^k_{\varphi_k^{-1}(i),\cdot} x + b^k_{\varphi_k^{-1}(i)} = 0. \tag{3.B.2}$$

Recall from Proposition 49 that

$$\tilde{\Omega}_{k+1} = P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) \Omega_{k+1}.$$

Since $P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1})$ is an invertible matrix, it induces an homeomorphism on $\mathbb{R}^{n_{k+1}}$, and thus this identity also holds for the interiors:

$$\mathring{\tilde{\Omega}}_{k+1} = P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) \mathring{\Omega}_{k+1}.$$

Given that $x \in \mathring{\Omega}_{k+1}$, defining $y = P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) x$, we have $y \in \mathring{\tilde{\Omega}}_{k+1}$.

Using (3.B.1), (3.A.2) and (3.B.2), we have

$$\begin{aligned}
\tilde{M}^k_{i,.} y + \tilde{b}^k_i &= [P_{\varphi_k} \operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}}]_{i,.} y + [P_{\varphi_k} \operatorname{Diag}(\lambda^k) b^k]_i \\
&= [\operatorname{Diag}(\lambda^k) M^k \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}}]_{\varphi_k^{-1}(i),.} y + [\operatorname{Diag}(\lambda^k) b^k]_{\varphi_k^{-1}(i)} \\
&= \lambda^k_{\varphi_k^{-1}(i)} M^k_{\varphi_k^{-1}(i),.} \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} y + \lambda^k_{\varphi_k^{-1}(i)} b^k_{\varphi_k^{-1}(i)} \\
&= \lambda^k_{\varphi_k^{-1}(i)} M^k_{\varphi_k^{-1}(i),.} x + \lambda^k_{\varphi_k^{-1}(i)} b^k_{\varphi_k^{-1}(i)} \\
&= 0.
\end{aligned}$$

We showed that there exists $y \in \mathring{\tilde{\Omega}}_{k+1}$ such that

$$\tilde{M}^k_{i,.} y + \tilde{b}^k_i = 0,$$

which concludes the proof of **P**.*b*).

P.*c*) Let $\tilde{D} \in \tilde{\Pi}_k$ and $i \in [\![1, n_k]\!]$. Suppose $E^k_i \cap \tilde{D} \cap \tilde{h}_k(\tilde{\Omega}_{k+1}) \neq \emptyset$, and let us show $\tilde{V}^k_{i,.}(\tilde{D}) \neq 0$.

Let $x \in \tilde{\Omega}_{k+1}$ such that $\tilde{h}_k(x) \in E^k_i \cap \tilde{D}$. Inverting the equalities of Proposition 49 we get

- $h_k = \operatorname{Diag}(\lambda^k)^{-1} P^{-1}_{\varphi_k} \tilde{h}_k \circ P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1})$,
- $H^{k+1}_{\varphi_k^{-1}(i)} = \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} \tilde{H}^{k+1}_i$,
- $\Omega_{k+1} = \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} \tilde{\Omega}_{k+1}$.

Denote $D = \operatorname{Diag}(\lambda^k)^{-1} P^{-1}_{\varphi_k} \tilde{D}$. Since $\tilde{\Pi}_k$ has been defined as in Item 4 of Proposition 49, we know that $D \in \Pi_k$. Let $y = \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} x$. Let us prove that $h_k(y) \in E^k_{\varphi_k(i)^{-1}} \cap D \cap h_k(\Omega_{k+1})$.

Since $x \in \tilde{\Omega}_{k+1}$, we see that $y \in \Omega_{k+1}$, so $h_k(y) \in h_k(\Omega_{k+1})$.

We also have

$$\begin{aligned}
h_k(y) &= \operatorname{Diag}(\lambda^k)^{-1} P^{-1}_{\varphi_k} \tilde{h}_k \circ P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) \left(\operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} x \right) \\
&= \operatorname{Diag}(\lambda^k)^{-1} P^{-1}_{\varphi_k} \tilde{h}_k(x),
\end{aligned}$$

which shows, since $\tilde{h}_k(x) \in \tilde{D}$, that $h_k(y) \in D$.

Since, by hypothesis, $\tilde{h}_k(x) \in E_i^k$, using (3.A.2) and (3.A.3), we have

$$\begin{aligned}
[h_k(y)]_{\varphi_k^{-1}(i)} &= \left[\mathrm{Diag}(\lambda^k)^{-1} P_{\varphi_k}^{-1} \tilde{h}_k(x)\right]_{\varphi_k^{-1}(i)} \\
&= \frac{1}{\lambda^k_{\varphi_k^{-1}(i)}} \left[P_{\varphi_k}^{-1} \tilde{h}_k(x)\right]_{\varphi_k^{-1}(i)} \\
&= \frac{1}{\lambda^k_{\varphi_k^{-1}(i)}} (\tilde{h}_k(x))_i \\
&= 0.
\end{aligned}$$

This proves that $h_k(y) \in E^k_{\varphi_k^{-1}(i)}$.

We proved that

$$h_k(y) \in E^k_{\varphi_k^{-1}(i)} \cap D \cap h_k(\Omega_{k+1}),$$

which shows this intersection is not empty. Since $(g_k, M^k, b^k, \Omega_{k+1}, \Pi_k)$ satisfies **P**.*c*), we have $V^k_{.,\varphi_k^{-1}(i)}(D) \neq 0$.

Since, according to proposition 49,

$$\tilde{g}_k = g_k \circ \mathrm{Diag}(\lambda^k)^{-1} P_{\varphi_k}^{-1},$$

we deduce:

$$\tilde{V}^k(\tilde{D}) = V^k(D)\,\mathrm{Diag}(\lambda^k)^{-1} P_{\varphi_k}^{-1}. \tag{3.B.3}$$

For a matrix A and a permutation φ, we have $[P_\varphi A]_{i,.} = A_{\varphi^{-1}(i),.}$, so by taking the transpose, we see that $[A^T P_\varphi^{-1}]_{.,i} = (A^T)_{.,\varphi^{-1}(i)}$.

Taking the i^{th} column of (3.B.3), we thus obtain

$$\tilde{V}^k_{.,i}(\tilde{D}) = \left[V^k(D)\,\mathrm{Diag}(\lambda^k)^{-1} P_{\varphi_k}^{-1}\right]_{.,i} = \frac{1}{\lambda^k_{\varphi_k^{-1}(i)}} V^k_{.,\varphi_k^{-1}(i)}(D),$$

which shows that $\tilde{V}^k_{.,i}(\tilde{D}) \neq 0$.

P.*d*) Let $\tilde{H} \subset \mathbb{R}^{n_{k+1}}$ be an affine hyperplane. Denote $H = \mathrm{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}} \tilde{H}$. Since **P**.*d*) holds for $(g_k, M^k, b^k, \Omega_{k+1}, \Pi_k)$, using Item 2 of Proposition 49, we have

$$\begin{aligned}
\tilde{H} \cap \mathring{\tilde{\Omega}}_{k+1} &= P_{\varphi_{k+1}} \mathrm{Diag}(\lambda^{k+1}) \left(H \cap \mathring{\Omega}_{k+1}\right) \\
&\not\subset P_{\varphi_{k+1}} \mathrm{Diag}(\lambda^{k+1}) \bigcup_{D \in \Pi_k} \partial h_k^{-1}(D) \\
&= \bigcup_{D \in \Pi_k} P_{\varphi_{k+1}} \mathrm{Diag}(\lambda^{k+1}) \partial h_k^{-1}(D).
\end{aligned} \tag{3.B.4}$$

For all k, $P_{\varphi_{k+1}} \mathrm{Diag}(\lambda^{k+1})$ is an invertible matrix, so it induces an homeomorphism of $\mathbb{R}^{n_{k+1}}$. We thus have

$$P_{\varphi_{k+1}} \mathrm{Diag}(\lambda^{k+1}) \partial h_k^{-1}(D) = \partial \left(P_{\varphi_{k+1}} \mathrm{Diag}(\lambda^{k+1}) h_k^{-1}(D)\right). \tag{3.B.5}$$

Furthermore, by Item 1 of Proposition 49, we have $\tilde{h}_k = P_{\varphi_k} \operatorname{Diag}(\lambda^k) h_k \circ \operatorname{Diag}(\lambda^{k+1})^{-1} P^{-1}_{\varphi_{k+1}}$, so

$$\tilde{h}_k^{-1} = P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) h_k^{-1} \circ \operatorname{Diag}(\lambda^k)^{-1} P^{-1}_{\varphi_k},$$

and since $\tilde{D} = P_{\varphi_k} \operatorname{Diag}(\lambda^k) D$,

$$\tilde{h}_k^{-1}(\tilde{D}) = P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) h_k^{-1}(D). \tag{3.B.6}$$

Combining (3.B.5) and (3.B.6), we obtain

$$P_{\varphi_{k+1}} \operatorname{Diag}(\lambda^{k+1}) \partial h_k^{-1}(D) = \partial \tilde{h}_k^{-1}(\tilde{D}),$$

and we can thus reformulate (3.B.4) as

$$\tilde{H} \cap \mathring{\tilde{\Omega}}_{k+1} \quad \not\subset \quad \bigcup_{\tilde{D} \in \tilde{\Pi}_k} \partial \tilde{h}_k^{-1}(\tilde{D}).$$

□

3.B.2 Identifiability statement

We restate here the main theorem, already stated as Theorem 17 in the main part of the article.

Theorem 61. *Let $K \in \mathbb{N}$, $K \geq 2$. Suppose we are given two networks with K layers, identical number of neurons per layer, and with respective parameters $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$. Assume $\mathbf{\Pi}$ and $\tilde{\mathbf{\Pi}}$ are two lists of sets of closed polyhedra that are admissible with respect to $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ respectively. Denote by n_K the number of neurons of the input layer, and suppose we are given a set $\Omega \subset \mathbb{R}^{n_K}$ such that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ satisfy the conditions $\mathbf{P}$, and such that, for all $x \in \Omega$:*

$$f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x).$$

Then:

$$(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}}).$$

3.B.3 An application to risk minimization

We restate here the consequence of the main result in terms of minimization of the population risk, already stated as Corollary 18 in the main part.

Assume we are given a couple of input-output variables (X, Y) generated by a ground truth network with parameters $(\mathbf{M}, \mathbf{b})$:

$$Y = f_{\mathbf{M},\mathbf{b}}(X).$$

We can use Theorem 61 to show that the only way to bring the population risk to 0 is to find the ground truth parameters -modulo permutation and positive rescaling.

Indeed, let $\Omega \subset \mathbb{R}^{n_K}$ be a domain that is contained in the support of X, and suppose $L : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}_+$ is a loss function such that $L(y, y') = 0 \Rightarrow y = y'$. Consider the population risk:

$$R(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}) = \mathbb{E}[L(f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(X), Y)].$$

We have the following result.

Corollary 62. *Suppose there exists a list of sets of closed polyhedra* $\mathbf{\Pi}$ *admissible with respect to* $(\mathbf{M}, \mathbf{b})$ *such that* $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ *satisfies the conditions* $\mathbf{P}$*.*

If $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ *is also such that there exists a list of sets of closed polyhedra* $\tilde{\mathbf{\Pi}}$ *admissible with respect to* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ *such that* $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ *satisfies the conditions* $\mathbf{P}$*, and if* $(\mathbf{M}, \mathbf{b}) \not\sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$*, then:*

$$R(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}) > 0.$$

3.B.4 Proof of Theorem 61

To prove Theorem 61, we can assume the parameterizations $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are normalized. Indeed, if they are not, by Proposition 52 there exist a normalized parameterization $(\mathbf{M}', \mathbf{b}')$ equivalent to $(\mathbf{M}, \mathbf{b})$ and a normalized parameterization $(\tilde{\mathbf{M}}', \tilde{\mathbf{b}}')$ equivalent to $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$. Note that we can apply Proposition 52 because M^k and $\tilde{M}^k$ are full row rank (condition $\mathbf{P}.a$)) for all $k \in [\![1, K-1]\!]$ so their lines are always nonzero. We derive $\mathbf{\Pi}'$ from $\mathbf{\Pi}$ and $\tilde{\mathbf{\Pi}}'$ from $\tilde{\mathbf{\Pi}}$ as in Item 4 of Proposition 49. By Proposition 60, $(\mathbf{M}', \mathbf{b}', \Omega, \mathbf{\Pi}')$ and $(\tilde{\mathbf{M}}', \tilde{\mathbf{b}}', \Omega, \tilde{\mathbf{\Pi}}')$ also satisfy the conditions $\mathbf{P}$. By Corollary 50, $f_{\mathbf{M}',\mathbf{b}'} = f_{\mathbf{M},\mathbf{b}}$ and $f_{\tilde{\mathbf{M}}',\tilde{\mathbf{b}}'} = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$, so we have, for all $x \in \Omega$:

$$f_{\mathbf{M}',\mathbf{b}'}(x) = f_{\tilde{\mathbf{M}}',\tilde{\mathbf{b}}'}(x).$$

$(\mathbf{M}', \mathbf{b}', \Omega, \mathbf{\Pi}')$ and $(\tilde{\mathbf{M}}', \tilde{\mathbf{b}}', \Omega, \tilde{\mathbf{\Pi}}')$ satisfy the hypotheses of Theorem 61. If we are able to show that $(\mathbf{M}', \mathbf{b}') \sim (\tilde{\mathbf{M}}', \tilde{\mathbf{b}}')$, then $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ follows immediately from the transitivity of the equivalence relation, proven in Proposition 48.

Thus in the proof $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ will be assumed to be normalized.

To prove the theorem, we need the following fundamental lemma (already stated as Lemma 24 in the main text), that is proven in Appendix 3.C.

Lemma 63. *Let* $l, m, n \in \mathbb{N}^*$*. Suppose* $g, \tilde{g} : \mathbb{R}^m \to \mathbb{R}^n$ *are continuous piecewise linear functions,* $\Omega \subset \mathbb{R}^l$ *is a subset and let* $M, \tilde{M} \in \mathbb{R}^{m \times l}$*,* $b, \tilde{b} \in \mathbb{R}^m$*. Denote* $h : x \mapsto \sigma(Mx + b)$ *and* $\tilde{h} : x \mapsto \sigma(\tilde{M}x + \tilde{b})$*. Assume* Π *and* $\tilde{\Pi}$ *are two sets of polyhedra admissible with respect to* g *and* $\tilde{g}$ *respectively as in Definition 31.*

Suppose (g, M, b, Ω, Π) *and* $(\tilde{g}, \tilde{M}, \tilde{b}, \Omega, \tilde{\Pi})$ *satisfy the conditions* $\mathbf{C}$*, and for all* $i \in [\![1, m]\!]$*,* $\|M_{i,.}\| = \|\tilde{M}_{i,.}\| = 1$*.*

Suppose for all $x \in \Omega$*:*

$$g \circ h(x) = \tilde{g} \circ \tilde{h}(x).$$

Then, there exists a permutation $\varphi \in \mathfrak{S}_m$*, such that:*

— $\tilde{M} = P_\varphi M$;
— $\tilde{b} = P_\varphi b$;
— g *and* $\tilde{g} \circ P_\varphi$ *coincide on* $h(\Omega)$.

Proof of Theorem 61. We prove the theorem by induction on K.

Initialization. Assume here $K = 2$. We are going to apply Lemma 63. Since $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ satisfy the conditions $\mathbf{P}$, by definition, both $(g_1, M^1, b^1, \Omega_2, \Pi_1)$ and $(\tilde{g}_1, \tilde{M}^1, \tilde{b}^1, \Omega_2, \tilde{\Pi}_1)$ satisfy the conditions $\mathbf{C}$ (note that $\tilde{\Omega}_2 = \Omega_2 = \Omega$). The network is normalized, so we have, for all $i \in [\![1, n_1]\!]$,

$$\|M^1_{i,\cdot}\| = \|\tilde{M}^1_{i,\cdot}\| = 1.$$

By the assumptions of Theorem 61, for all $x \in \Omega$,

$$g_1 \circ h_1(x) = f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x) = \tilde{g}_1 \circ \tilde{h}_1(x).$$

We can thus apply Lemma 63, which shows that there exists a permutation $\varphi \in \mathfrak{S}_{n_1}$ such that

— $\tilde{M}^1 = P_\varphi M^1$;
— $\tilde{b}^1 = P_\varphi b^1$;
— g_1 and $\tilde{g}_1 \circ P_\varphi$ coincide on $h_1(\Omega)$.

Recall from Definition 40 that for all $i \in [\![1, n_1]\!]$, we denote

$$H^2_i = \{x \in \mathbb{R}^{n_2},\ M^1_{i,\cdot} x + b^1_i = 0\}.$$

Let $(v_1, \dots, v_{n_1})$ be the canonical basis of $\mathbb{R}^{n_1}$. Let us show that for all $i \in [\![1, n_1]\!]$,

$$M^0 v_i = \tilde{M}^0 P_\varphi v_i.$$

Let $i \in [\![1, n_1]\!]$. By $\mathbf{P}.b)$, $H^2_i \cap \mathring{\Omega} \neq \emptyset$. Since M^1 is full row rank by $\mathbf{P}.a)$, none of the hyperplanes H^2_j, with $j \neq i$, is parallel to H^2_i. As a consequence, the intersections $H^2_i \cap H^2_j$ have Hausdorff dimension smaller than $n_2 - 2$, so there exists $x \in \mathring{\Omega} \cap H^2_i \setminus \left(\bigcup_{j \neq i} H^2_j\right)$, and $\epsilon > 0$ such that $B(x, \epsilon) \cap H^2_j = \emptyset$ for all $j \neq i$. Let u be a unit vector such that $M^1_{j,\cdot} u = 0$ for all $j \neq i$ and $M^1_{i,\cdot} u = \alpha > 0$ (this is possible again since M^1 is full row rank).

For all $j \in [\![1, n_1]\!] \setminus \{i\}$, we have

$$\sigma(M^1_{j,\cdot}(x + \epsilon u) + b^1_j) - \sigma(M^1_{j,\cdot} x + b^1_j) = \sigma(M^1_{j,\cdot} x + b^1_j) - \sigma(M^1_{j,\cdot} x + b^1_j) = 0.$$

At the same time, we have

$$\begin{aligned}\sigma(M^1_{i,\cdot}(x + \epsilon u) + b^1_i) - \sigma(M^1_{i,\cdot} x + b^1_i) &= M^1_{i,\cdot}(x + \epsilon u) + b^1_i - M^1_{i,\cdot} x + b^1_i \\ &= \epsilon M^1_{i,\cdot} u \\ &= \epsilon\alpha.\end{aligned}$$

Summarizing,

$$\begin{aligned}h_1(x + \epsilon u) - h_1(x) &= \sigma(M^1(x + \epsilon u) + b^1) - \sigma(M^1 x + b^1) \\ &= \epsilon\alpha v_i.\end{aligned}$$

Let us denote $y_2 = h_1(x + \epsilon u) \in h_1(\Omega)$ and $y_1 = h_1(x) \in h_1(\Omega)$. We have shown $y_2 - y_1 = \epsilon\alpha v_i$, and since g_1 and $\tilde{g}_1 \circ P_\varphi$ coincide on $h_1(\Omega)$, we have

$$\begin{aligned} & g_1(y_2) - g_1(y_1) = \tilde{g}_1 \circ P_\varphi(y_2) - \tilde{g}_1 \circ P_\varphi(y_1) \\ \iff & M^0(y_2 - y_1) = \tilde{M}^0 P_\varphi(y_2 - y_1) \\ \iff & \epsilon\alpha M^0 v_i = \epsilon\alpha \tilde{M}^0 P_\varphi v_i \\ \iff & M^0 v_i = \tilde{M}^0 P_\varphi v_i. \end{aligned}$$

Since this last equality holds for any $i \in [\![1, n_1]\!]$, we conclude that

$$M^0 = \tilde{M}^0 P_\varphi,$$

and using one last time that g_1 and $\tilde{g}_1 \circ P_\varphi$ coincide on $h_1(\Omega)$, we obtain

$$b^0 = \tilde{b}^0,$$

i.e. we have shown

$$\begin{cases} \tilde{M}^0 = M^0 P_\varphi^{-1} \\ \tilde{b}^0 = b^0. \end{cases}$$

Defining $P_{\varphi_1} = P_\varphi$, $P_{\varphi_0} = \mathrm{Id}_{n_0}$ and $P_{\varphi_2} = \mathrm{Id}_{n_2}$, we can use Proposition 53 to conclude that

$$(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}}).$$

Induction step. Let $K \geq 3$ be an integer. Suppose Theorem 61 is true for all networks with $K - 1$ layers.

Consider two networks with parameters $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, with K layers and, for all $k \in [\![0, K]\!]$, same number n_k of neurons per layer. Let $\mathbf{\Pi}$ and $\tilde{\mathbf{\Pi}}$ be two list of sets of closed polyhedra that are admissible with respect to $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ respectively (Definition 44), and let $\Omega \subset \mathbb{R}^{n_K}$ such that $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ satisfy the conditions $\mathbf{P}$ and $f_{\mathbf{M},\mathbf{b}}$ and $f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$ coincide on Ω.

Recall the functions h_k and g_k associated to $(\mathbf{M}, \mathbf{b})$, defined in Definition 34 and Definition 38 respectively, and the corresponding functions $\tilde{h}_k$ and $\tilde{g}_k$ associated to $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$.

We have two matrices M^{K-1} and $\tilde{M}^{K-1} \in \mathbb{R}^{n_{K-1} \times n_K}$, two vectors b^{K-1} and $\tilde{b}^{K-1} \in \mathbb{R}^{n_{K-1}}$, two functions g_{K-1} and $\tilde{g}_{K-1} : \mathbb{R}^{n_{K-1}} \to \mathbb{R}^{n_0}$, two sets Π_{K-1} and $\tilde{\Pi}_{K-1}$ such that:

- $\forall x \in \Omega, \quad g_{K-1} \circ h_{K-1}(x) = g_K(x) = f_{\mathbf{M},\mathbf{b}}(x) = f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x) = \tilde{g}_K(x) = \tilde{g}_{K-1} \circ \tilde{h}_{K-1}(x)$,
- g_{K-1} and $\tilde{g}_{K-1}$ are continuous piecewise linear, and Π_{K-1} and $\tilde{\Pi}_{K-1}$ are admissible with respect to g_{K-1} and $\tilde{g}_{K-1}$ respectively,
- $(g_{K-1}, M^{K-1}, b^{K-1}, \Omega, \Pi_{K-1})$ and $(\tilde{g}_{K-1}, \tilde{M}^{K-1}, \tilde{b}^{K-1}, \Omega, \tilde{\Pi}_{K-1})$ satisfy the conditions $\mathbf{C}$,
- $\forall i \in [\![1, n_{K-1}]\!], \quad \|M^{K-1}_{i,\cdot}\| = \|\tilde{M}^{K-1}_{i,\cdot}\| = 1.$

The third point comes from the fact that the conditions **P** hold for $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$, and the fourth point comes from the fact that $(\mathbf{M}, \mathbf{b})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$ are normalized.

Thus, the objects $g_{K-1}, \tilde{g}_{K-1}, M^{K-1}, b^{K-1}, \tilde{M}^{K-1}, \tilde{b}^{K-1}, \Pi_{K-1}$ and $\tilde{\Pi}_{K-1}$ satisfy the hypotheses of Lemma 63 and hence there exists $\varphi \in \mathfrak{S}_{n_{K-1}}$ such that

$$\begin{cases} \tilde{M}^{K-1} = P_\varphi M^{K-1}, \\ \tilde{b}^{K-1} = P_\varphi b^{K-1}, \end{cases} \tag{3.B.7}$$

and g_{K-1} and $\tilde{g}_{K-1} \circ P_\varphi$ coincide on Ω_{K-1}.

Let us denote $\mathbf{M}^* = (M^0, \dots, M^{K-3}, M^{K-2}P_\varphi^{-1})$. The functions $g_{K-1} \circ P_\varphi^{-1}$ and $\tilde{g}_{K-1}$ are implemented by two networks with $K-1$ layers, indexed from $K-1$ up to 0, with parameters $(\mathbf{M}^*, \mathbf{b}^{\leq K-2})$ and $(\tilde{\mathbf{M}}^{\leq K-2}, \tilde{\mathbf{b}}^{\leq K-2})$ respectively. The previous paragraph shows these functions coincide on $P_\varphi \Omega_{K-1}$. Recalling the definition of $\tilde{\Omega}_{K-1}$ and since, by (3.B.7), $\tilde{f}_{K-1} = \tilde{h}_{K-1} = P_\varphi h_{K-1}$, we have

$$\tilde{\Omega}_{K-1} = \tilde{f}_{K-1}(\Omega) = P_\varphi h_{K-1}(\Omega) = P_\varphi \Omega_{K-1},$$

i.e. the functions $g_{K-1} \circ P_\varphi^{-1} = f_{\mathbf{M}^*, \mathbf{b}^{\leq K-2}}$ and $\tilde{g}_{K-1} = f_{\tilde{\mathbf{M}}^{\leq K-2}, \tilde{\mathbf{b}}^{\leq K-2}}$ coincide on $\tilde{\Omega}_{K-1}$.

Since $(\mathbf{M}, \mathbf{b}, \Omega, \mathbf{\Pi})$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\mathbf{\Pi}})$ satisfy the conditions **P**, we know that $(g_k, M^k, b^k, \Omega_{k+1}, \Pi_k)$ and $(\tilde{g}_k, \tilde{M}^k, \tilde{b}^k, \tilde{\Omega}_{k+1}, \tilde{\Pi}_k)$ satisfy the conditions **C** for all $k \in [\![1, K-1]\!]$ so in particular these conditions are satisfied for $k \in [\![1, K-2]\!]$, so $(\mathbf{M}^{\leq K-2}, \mathbf{b}^{\leq K-2}, \Omega_{K-1}, \mathbf{\Pi}^{\leq K-2})$ and $(\tilde{\mathbf{M}}^{\leq K-2}, \tilde{\mathbf{b}}^{\leq K-2}, \tilde{\Omega}_{K-1}, \tilde{\mathbf{\Pi}}^{\leq K-2)})$ satisfy the conditions **P**.

Let us verify that $(\mathbf{M}^*, \mathbf{b}^{\leq K-2}, \tilde{\Omega}_{K-1}, \mathbf{\Pi}^{\leq K-2})$ also satisfies the conditions **P**. Indeed, the only thing that differs from $(\mathbf{M}^{\leq K-2}, \mathbf{b}^{\leq K-2}, \Omega_{K-1}, \mathbf{\Pi}^{\leq K-2})$ is $\tilde{\Omega}_{K-1}$ and the weights M^{*K-2} between the layer $K-1$ and the layer $K-2$. Writing that $M^{*K-2} = M^{K-2}P_\varphi^{-1}$, $h^*_{K-2} = h_{K-2} \circ P_\varphi^{-1}$, $\tilde{\Omega}_{K-1} = P_\varphi \Omega_{K-1}$ and $H_i^{*K-1} = P_\varphi H_i^{K-1}$, let us check, one by one, that the conditions **C**.a) – **C**.d) also hold for $(g_{K-2}, M^{*K-2}, b^{K-2}, \tilde{\Omega}_{K-1}, \Pi_{K-2})$.

Indeed P_φ^{-1} is invertible, so M^{*K-2} is full row rank and **C**.a) holds.

If $x \in \mathring{\Omega}$ satisfies $M^{K-2}_{i,\cdot} x + b_i^{K-2} = 0$, we define $h^{*lin}_{K-2}(x) = M^{*K-2}x + b^{K-2}$, we have $h^{*lin}_{K-2} = h^{lin}_{K-2} \circ P_\varphi^{-1}$, so

$$E_i \cap h^{*lin}_{K-2}(\mathring{\tilde{\Omega}}_{K-1}) = E_i \cap h^{lin}_{K-2}(\mathring{\Omega}_{K-1}) \neq \emptyset,$$

and **C**.b) is satisfied.

Similarly, the observation $h^*_{K-2}(\tilde{\Omega}_{K-1}) = h_{K-2}(\Omega_{K-1})$ yields **C**.c).

Finally, assume $H^* \subset \mathbb{R}^{n_{K-1}}$ is an affine hyperplane. Let $H = P_\varphi^{-1} H^*$. We have by hypothesis

$$H \cap \mathring{\Omega}_{K-1} \not\subset \bigcup_{D \in \Pi_{K-2}} \partial h^{-1}_{K-2}(D),$$

thus

$$\begin{aligned} H^* \cap \mathring{\tilde{\Omega}}_{K-1} &= P_\varphi \left(H \cap \mathring{\Omega}_{K-1} \right) \\ &\not\subset P_\varphi \bigcup_{D \in \Pi_{K-2}} \partial h_{K-2}^{-1}(D) \\ &= \bigcup_{D \in \Pi_{K-2}} \partial (P_\varphi h_{K-2}^{-1}(D)). \end{aligned}$$

For all $D \in \Pi_{K-2}$ we have

$$\begin{aligned} P_\varphi h_{K-2}^{-1}(D) &= P_\varphi \{ y,\ h_{K-2}(y) \in D \} \\ &= P_\varphi \{ P_\varphi^{-1} x,\ h_{K-2} \circ P_\varphi^{-1}(x) \in D \} \\ &= \{ x,\ h_{K-2}^*(x) \in D \} \\ &= h_{K-2}^{*-1}(D). \end{aligned}$$

Therefore,

$$H^* \cap \mathring{\tilde{\Omega}}_{K-1} = \bigcup_{D \in \Pi_{K-2}} \partial h_{K-2}^{*-1}(D),$$

which proves $\mathbf{C}.d$).

Since the rest stays unchanged, we can conclude.

The induction hypothesis can thus be applied to $(\mathbf{M}^*, \mathbf{b}^{\leq K-2}, \tilde{\Omega}_{K-1}, \mathbf{\Pi}^{\leq K-2})$ and $(\tilde{\mathbf{M}}^{\leq K-2}, \tilde{\mathbf{b}}^{\leq K-2}, \tilde{\Omega}_{K-1}, \tilde{\mathbf{\Pi}}^{\leq K-2})$, to obtain:

$$(\mathbf{M}^*, \mathbf{b}^{\leq K-2}) \sim (\tilde{\mathbf{M}}^{\leq K-2}, \tilde{\mathbf{b}}^{\leq K-2}).$$

Since we also have

$$\forall k \in [\![1, K-3]\!],\ \forall i \in [\![1, n_k]\!], \qquad \|M_{i,\cdot}^{*k}\| = \|M_{i,\cdot}^k\| = 1 \quad \text{and} \quad \|\tilde{M}_{i,\cdot}^k\| = 1,$$

$$\forall i \in [\![1, n_{K-2}]\!], \quad \|M_{i,\cdot}^{*K-2}\| = \|M_{i,\cdot}^{K-2} P_\varphi^{-1}\| = \|M_{i,\cdot}^{K-2}\| = 1 \quad \text{and} \quad \|\tilde{M}_{i,\cdot}^{K-2}\| = 1,$$

Proposition 53 shows that there exists a family of permutations $(\varphi_0, \dots, \varphi_{K-1}) \in \mathfrak{S}_{n_0} \times \cdots \times \mathfrak{S}_{n_{K-1}}$, with $\varphi_0 = id_{[\![1,n_0]\!]}$ and $\varphi_{K-1} = id_{[\![1,n_{K-1}]\!]}$, such that:

$$\forall k \in [\![0, K-3]\!], \quad \begin{cases} \tilde{M}^k = P_{\varphi_k} M^{*k} P_{\varphi_{k+1}}^{-1} = P_{\varphi_k} M^k P_{\varphi_{k+1}}^{-1} \\ \tilde{b}^k = P_{\varphi_k} b^k, \end{cases} \tag{3.B.8}$$

and:

$$\begin{cases} \tilde{M}^{K-2} = P_{\varphi_{K-2}} M^{*K-2} P_{\varphi_{K-1}}^{-1} = P_{\varphi_{K-2}} (M^{K-2} P_\varphi^{-1}) P_{\varphi_{K-1}}^{-1} = P_{\varphi_{K-2}} M^{K-2} P_\varphi^{-1} \\ \tilde{b}^{K-2} = P_{\varphi_{K-2}} b^{K-2}. \end{cases} \tag{3.B.9}$$

We can define $(\psi_0, \dots, \psi_K) \in \mathfrak{S}_{n_0} \times \cdots \times \mathfrak{S}_{n_K}$ by:

— $\psi_0 = id_{[\![1,n_0]\!]}$, $\psi_K = id_{[\![1,n_K]\!]}$;

— $\forall k \in [\![1, K-2]\!]$, $\psi_k = \varphi_k$;

— $\psi_{K-1} = \varphi$;

and using (3.B.8), (3.B.9) and (3.B.7) altogether, we then have, for all $k \in [\![0, K-1]\!]$:

$$\begin{cases} \tilde{M}^k = P_{\psi_k} M^k P_{\psi_{k+1}}^{-1} \\ \tilde{b}^k = P_{\psi_k} b^k. \end{cases}$$

It follows from Proposition 53 that $(\mathbf{M}, \mathbf{b}) \sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$.

□

3.B.5 Proof of Corollary 62

Theorem 62 is an immediate consequence of Theorem 61.

Since $(\mathbf{M}, \mathbf{b}, \Omega, \Pi)$ and $(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}, \Omega, \tilde{\Pi})$ satisfy the conditions **P** and $(\mathbf{M}, \mathbf{b}) \not\sim (\tilde{\mathbf{M}}, \tilde{\mathbf{b}})$, the contrapositive of Theorem 61 shows that there exists $x \in \Omega$ such that $f_{\mathbf{M},\mathbf{b}}(x) \neq f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(x)$. The function $f_{\mathbf{M},\mathbf{b}} - f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}$ is continuous so there exists $r > 0$ such that for all $u \in B(x, r)$, $f_{\mathbf{M},\mathbf{b}}(u) \neq f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(u)$ so $L(f_{\mathbf{M},\mathbf{b}}(u), f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(u)) > 0$. Since Ω is included in the support of X and $x \in \Omega$, denoting $\mathbb{P}_X$ the law of X we have $\mathbb{P}_X(B(x, r)) > 0$ and thus

$$\begin{aligned} R(\tilde{\mathbf{M}}, \tilde{\mathbf{b}}) &= \mathbb{E}[L(f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(X), f_{\mathbf{M},\mathbf{b}}(X))] \\ &\geq \int_{B(x,r)} L(f_{\mathbf{M},\mathbf{b}}(u), f_{\tilde{\mathbf{M}},\tilde{\mathbf{b}}}(u)) \mathrm{d}\mathbb{P}_X(u) \\ &> 0. \end{aligned}$$

3.C Proof of Lemma 63

In this section we prove Lemma 63.

Let (g, M, b, Ω, Π) and $(\tilde{g}, \tilde{M}, \tilde{b}, \Omega, \tilde{\Pi})$ be as in the lemma. In particular, we assume they satisfy the conditions **C** all along Appendix 3.C.

We denote, for all $x \in \mathbb{R}^l$:

$$f(x) = g\left(\sigma(Mx + b)\right).$$

Recall that, for all $x \in \mathbb{R}^l$, $h(x) = \sigma(Mx + b)$ and $\tilde{h}(x) = \sigma(\tilde{M}x + \tilde{b})$.

Recall that, as in Definition 40, we define for all $i \in [\![1, m]\!]$ the sets $H_i = \{x \in \mathbb{R}^l \ , \ M_{i,.}x + b_i = 0\}$ and $\tilde{H}_i = \{x \in \mathbb{R}^l \ , \ \tilde{M}_{i,.}x + \tilde{b}_i = 0\}$. By condition *C.a*), for all $i \in [\![1, m]\!]$, $M_{i,.} \neq 0$ and $\tilde{M}_{i,.} \neq 0$ so H_i and $\tilde{H}_i$ are hyperplanes.

Recall that for all $D \in \Pi$, we define $V(D) \in \mathbb{R}^{n \times m}$ and $c(D) \in \mathbb{R}^n$ as in Definition 55, and similarly for all $\tilde{D} \in \tilde{\Pi}$, we define $\tilde{V}(\tilde{D}) \in \mathbb{R}^{n \times m}$ and $\tilde{c}(\tilde{D}) \in \mathbb{R}^n$ associated to $\tilde{g}$.

We now define $s : \mathbb{R}^l \to \{0, 1\}^m$ as follows:

$$\forall i \in [\![1, m]\!], \quad s_i(x) := \begin{cases} 1 & \text{if } M_{i,.}x + b_i \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.C.1}$$

We define similarly $\tilde{s}$ for $(\tilde{M}, \tilde{b})$. We thus have, for all $i \in [\![1, m]\!]$,

$$\sigma(M_{i,.}x + b_i) = s_i(x)(M_{i,.}x + b_i)$$

and

$$\sigma(\tilde{M}_{i,.}x + \tilde{b}_i) = \tilde{s}_i(x)(\tilde{M}_{i,.}x + \tilde{b}_i).$$

Let $D \in \Pi$. For all $y \in D$, we have, by definition,

$$g(y) = V(D)y + c(D),$$

thus, for all $x \in h^{-1}(D)$,

$$\begin{aligned} f(x) &= V(D)h(x) + c(D) \\ &= V(D)\sigma(Mx + b) + c(D) \\ &= \sum_{k=1}^{m} V_{.,k}(D)s_k(x)(M_{k,.}x + b_k) + c(D). \end{aligned} \tag{3.C.2}$$

Similarly, for all $\tilde{D} \in \tilde{\Pi}$, for all $x \in \tilde{h}^{-1}(\tilde{D})$,

$$f(x) = \sum_{k=1}^{m} \tilde{V}_{.,k}(\tilde{D})\tilde{s}_k(x)(\tilde{M}_{k,.}x + \tilde{b}_k) + \tilde{c}(\tilde{D}). \tag{3.C.3}$$

Proposition 64. *Let $D \in \Pi$. For all $i \in [\![1, m]\!]$, for all $x \in H_i \cap \overbrace{h^{-1}(D)}^{\circ} \cap \mathring{\Omega} \setminus \left(\bigcup_{k \neq i} H_k\right)$, f is not differentiable at the point x.*

Proof. Let $i \in [\![1, m]\!]$ and suppose $x \in H_i \cap \overbrace{h^{-1}(D)}^{\circ} \cap \mathring{\Omega} \setminus \left(\bigcup_{k \neq i} H_k\right)$. Let us consider the function $t \mapsto f(x + tM_{i,.}^T)$. Since $x \in H_i$ and $\|M_{i,.}\| = 1$ by hypothesis,

$$M_{i,.}(x + tM_{i,.}^T) + b_i = tM_{i,.}M_{i,.}^T + M_{i,.}x + b_i = t\|M_{i,.}\|^2 = t. \tag{3.C.4}$$

Given the definition of s in (3.C.1), we thus have

$$s_i(x + tM_{i,.}^T) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{if } t < 0. \end{cases}$$

Since $x \in \overbrace{h^{-1}(D)}^{\circ}$ which is an open set, for t small enough we have $x + tM_{i,.}^T \in \overbrace{h^{-1}(D)}^{\circ}$ and thus, using (3.C.2) and (3.C.4),

$$\begin{aligned} f(x + tM_{i,.}^T) &= \sum_{k=1}^{m} V_{.,k}(D)s_k(x + tM_{i,.}^T)\left(M_{k,.}(x + tM_{i,.}^T) + b_k\right) + c(D) \\ &= \begin{cases} \sum_{k \neq i} V_{.,k}(D)s_k(x + tM_{i,.}^T)\left(M_{k,.}(x + tM_{i,.}^T) + b_k\right) & \\ +c(D) + tV_{.,i}(D) & \text{if } t \geq 0 \\ \sum_{k \neq i} V_{.,k}(D)s_k(x + tM_{i,.}^T)\left(M_{k,.}(x + tM_{i,.}^T) + b_k\right) & \\ +c(D) & \text{if } t < 0. \end{cases} \end{aligned}$$

Since x does not belong to any of the hyperplanes H_k for $k \neq i$, which are closed, there exists $\epsilon > 0$ such that for all $t \in]-\epsilon, \epsilon[$ and for all $k \neq i$, $x + tM_{i,\cdot}{}^T \notin H_k$. Therefore, for all $t \in]-\epsilon, \epsilon[$, for all $k \in [\![1, m]\!] \backslash \{i\}$, $s_k(x + tM_{i,\cdot}{}^T) = s_k(x)$ and

$$f(x + tM_{i,\cdot}{}^T) = \begin{cases} \sum_{k \neq i} V_{\cdot,k}(D) s_k(x)(M_{k,\cdot}(x + tM_{i,\cdot}{}^T) + b_k) + c(D) & \\ +tV_{\cdot,i}(D) & \text{if } t \geq 0 \\ \sum_{k \neq i} V_{\cdot,k}(D) s_k(x)(M_{k,\cdot}(x + tM_{i,\cdot}{}^T) + b_k) + c(D) & \text{if } t < 0. \end{cases}$$

The right derivative of $t \mapsto f(x + tM_{i,\cdot}{}^T)$ at 0 is:

$$\sum_{k \neq i} V_{\cdot,k}(D) s_k(x) M_{k,\cdot} M_{i,\cdot}{}^T + V_{\cdot,i}(D).$$

The left derivative of $t \mapsto f(x + tM_{i,\cdot}{}^T)$ at 0 is:

$$\sum_{k \neq i} V_{\cdot,k}(D) s_k(x) M_{k,\cdot} M_{i,\cdot}{}^T.$$

Since $x \in H_i \cap h^{-1}(D) \cap \Omega$, we have $h(x) \in E_i \cap D \cap h(\Omega)$ so the condition $\mathbf{C}.c)$ implies that $V_{\cdot,i}(D) \neq 0$. We conclude that the left and right derivatives at x do not coincide and thus f is not differentiable at x. □

Lemma 65. *Let $D \in \Pi$. For all $x \in \overbrace{h^{-1}(D)}^{\circ} \setminus (\bigcup_{i=1}^m H_i)$, there exists $r > 0$ such that f is differentiable on $B(x, r)$.*

Proof. Consider $x \in \overbrace{h^{-1}(D)}^{\circ} \setminus (\bigcup_{i=1}^m H_i)$. Since the hyperplanes H_i are closed, there exists a ball $B(x, r) \subset \overbrace{h^{-1}(D)}^{\circ}$ such that for all $i \in [\![1, m]\!]$, $B(x, r) \cap H_i = \emptyset$. As a consequence, for all $y \in B(x, r)$, $s(y) = s(x)$. Using (3.C.2) we get, for all $y \in B(x, r)$,

$$f(y) = \sum_{i=1}^m V_{\cdot,i}(D) s_i(x) \left(M_{i,\cdot} y + b_i\right) + c(D).$$

The right side of this equality is affine in the variable y, so f is differentiable on $B(x, r)$. □

Lemma 66. *Let $\gamma : \mathbb{R}^l \to \mathbb{R}^m$ be a continuous piecewise linear function. Let $\mathcal{P}$ be a finite set of polyhedra of $\mathbb{R}^m$ such that $\bigcup_{D \in \mathcal{P}} D = \mathbb{R}^m$. Let $A_1, \dots A_s$ be a set of hyperplanes such that $\bigcup_{D \in \mathcal{P}} \partial \gamma^{-1}(D) \subset \bigcup_{k=1}^s A_k$ (Proposition 33 shows the existence of such hyperplanes). Let H be an affine hyperplane and $a \in \mathbb{R}^l, b \in \mathbb{R}$ such that $H = \{x \in \mathbb{R}^l, a^T x + b = 0\}$. Denote $I = \{k \in [\![1, s]\!], A_k = H\}$. Let $x \in H$ such that for all $k \in [\![1, s]\!] \backslash I$, $x \notin A_k$. Then there exists $r > 0$, D_- and $D_+ \in \mathcal{P}$ (not necessarily distinct) such that*

$$\begin{aligned} B(x, r) \cap \{y \in \mathbb{R}^l, a^T y + b < 0\} &\subset \gamma^{-1}(D_-) \\ B(x, r) \cap \{y \in \mathbb{R}^l, a^T y + b > 0\} &\subset \gamma^{-1}(D_+). \end{aligned}$$

Proof. Let $r > 0$ such that

$$B(x,r) \cap \left(\bigcup_{k \notin I} A_k \right) = \emptyset.$$

$B(x,r) \backslash H$ has two connected components: $B_- = B(x,r) \cap \{y \in \mathbb{R}^l, a^T y + b < 0\}$ and $B_+ = B(x,r) \cap \{y \in \mathbb{R}^l, a^T y + b > 0\}$. The set B_- (resp. B_+) is convex as an intersection of two convex sets.

Since $\bigcup_{D \in \mathcal{P}} D = \mathbb{R}^m$, there exists $D_- \in \mathcal{P}$ such that $\gamma^{-1}(D_-) \cap B_- \neq \emptyset$. Let us show that

$$B_- \ \subset \ \gamma^{-1}(D_-).$$

Indeed, $B_- \cap \left(\bigcup_{k \notin I} A_k \right) = \emptyset$ and $B_- \cap H = \emptyset$ so $B_- \cap \left(\bigcup_{k \in I} A_k \right) = \emptyset$, therefore we have

$$B_- \cap \left(\bigcup_{D \in \mathcal{P}} \partial \gamma^{-1}(D) \right) \ \subset \ B_- \cap \left(\bigcup_{k=1}^{s} A_k \right) = \emptyset.$$

In particular, $B_- \cap \partial \gamma^{-1}(D_-) = \emptyset$. Let $Y = \gamma^{-1}(D_-) \cap B_-$. Let us denote by $\partial_{B_-} Y$ the topological boundary of Y with respect to the topology of B_-. Let us show the following inclusion:

$$\partial_{B_-} Y \ \subset \ \partial \gamma^{-1}(D_-) \cap B_-.$$

Indeed, let $y \in \partial_{B_-} Y$. By definition, there exist two sequences (u_n) and (v_n) such that $u_n \in Y$, $v_n \in B_- \backslash Y$, and both u_n and v_n tend to y. In particular, $u_n \in \gamma^{-1}(D_-)$ and $v_n \in \mathbb{R}^l \backslash \gamma^{-1}(D_-)$, so $y \in \partial \gamma^{-1}(D_-)$. Since $y \in B_-$, we have $y \in \partial \gamma^{-1}(D_-) \cap B_-$.

This shows $\partial_{B_-} Y = \emptyset$, and as a consequence Y is open and closed in B_-. Since B_- is connex and Y is not empty, we conclude that $Y = B_-$, i.e. $B_- \ \subset \ \gamma^{-1}(D_-)$.

We show similarly that there exists $D_+ \in \Pi$ such that $B_+ \subset \gamma^{-1}(D_+)$. □

Proposition 67. *There exists a bijection $\varphi \in \mathfrak{S}_m$ such that for all $i \in [\![1,m]\!]$, $\tilde{H}_i = H_{\varphi^{-1}(i)}$.*

Proof. We denote by X the set of all points of $\mathring{\Omega}$ at which f is not differentiable. We denote by $\mathcal{G}$ the set of all hyperplanes of $\mathbb{R}^l$. We denote $\mathcal{H} = \{H \in \mathcal{G} \ , \ H \cap \mathring{\Omega} \neq \emptyset \text{ and } H \cap \mathring{\Omega} \subset \overline{X}\}$. We want to show $\mathcal{H} = \{H_i, i \in [\![1,m]\!]\}$.

Indeed, once this established, since $\mathcal{H}$ only depends on Ω and f, we also have $\mathcal{H} = \{\tilde{H}_i, i \in [\![1,m]\!]\}$, and thus $\{H_i, i \in [\![1,m]\!]\} = \{\tilde{H}_i, i \in [\![1,m]\!]\}$. Since, using *C.a)*, for all i, j, $i \neq j$, we have $H_i \neq H_j$ and $\tilde{H}_i \neq \tilde{H}_j$, we can conclude that there exists a permutation $\varphi \in \mathfrak{S}_m$ such that, for all $i \in [\![1,m]\!]$, $\tilde{H}_i = H_{\varphi^{-1}(i)}$.

– **Let us show $\mathcal{H} \subset \{H_i, i \in [\![1,m]\!]\}$.**

To begin, let us show that $\overline{X} \cap \mathring{\Omega} \ \subset \ \bigcup_{D \in \Pi} \partial h^{-1}(D) \cup \bigcup_{i=1}^{m} H_i$. Let $x \in \overline{X} \cap \mathring{\Omega}$. Let $D \in \Pi$ such that $h(x) \in D$. Since $x \in \overline{X}$, there does not exist any $r > 0$ such that f is differentiable on $B(x,r)$. The contrapositive of Lemma 65 shows that

$x \notin \overbrace{h^{-1}(D)}^{\circ} \setminus (\bigcup_{i=1}^m H_i)$, so either $x \in \bigcup_{i=1}^m H_i$ or $x \notin \overbrace{h^{-1}(D)}^{\circ}$. In the latter case, since $x \in h^{-1}(D)$ by definition of D, we have $x \in h^{-1}(D) \setminus \overbrace{h^{-1}(D)}^{\circ} \subset \partial h^{-1}(D)$.

This shows:

$$\overline{X} \cap \mathring{\Omega} \quad \subset \quad \bigcup_{D \in \Pi} \partial h^{-1}(D) \cup \bigcup_{i=1}^m H_i. \tag{3.C.5}$$

Let $H \in \mathcal{H}$. We are going to show that there exists $i \in [\![1, m]\!]$ such that $H = H_i$.

We know by condition *C.d* that $H \cap \mathring{\Omega} \not\subset \bigcup_{D \in \Pi} \partial h^{-1}(D)$. Let $x \in (H \cap \mathring{\Omega}) \setminus \left(\bigcup_{D \in \Pi} \partial h^{-1}(D)\right)$. The set $\bigcup_{D \in \Pi} \partial h^{-1}(D)$ is closed, so there exists a ball

$$B(x, r) \subset \mathring{\Omega} \setminus \left(\bigcup_{D \in \Pi} \partial h^{-1}(D) \right). \tag{3.C.6}$$

By definition of $\mathcal{H}$,

$$H \cap \mathring{\Omega} \ \subset \ \overline{X} \cap \mathring{\Omega},$$

so using the fact that $B(x, r) \subset \mathring{\Omega}$ we have:

$$B(x, r) \cap H \ = \ B(x, r) \cap H \cap \mathring{\Omega} \ \subset \ B(x, r) \cap \overline{X} \cap \mathring{\Omega}.$$

Thus, using (3.C.5),

$$\begin{aligned} B(x, r) \cap H \ &\subset \ B(x, r) \cap \overline{X} \cap \mathring{\Omega} \\ &\subset \ B(x, r) \cap \left(\bigcup_{D \in \Pi} \partial h^{-1}(D) \cup \bigcup_{i=1}^m H_i \right) \\ &= \ \left(B(x, r) \cap \bigcup_{D \in \Pi} \partial h^{-1}(D) \right) \cup \left(B(x, r) \cap \bigcup_{i=1}^m H_i \right), \end{aligned}$$

and since by (3.C.6) the first set of the last equality is empty, we have

$$B(x, r) \cap H \ \subset \ B(x, r) \cap \bigcup_{i=1}^m H_i.$$

Therefore,

$$\begin{aligned} B(x, r) \cap H = (B(x, r) \cap H) \cap \left(B(x, r) \cap \bigcup_{i=1}^m H_i \right) \\ = B(x, r) \cap H \cap \bigcup_{i=1}^m H_i \\ = B(x, r) \cap \bigcup_{i=1}^m (H \cap H_i). \end{aligned}$$

Assume, by contradiction, that for all $i \in [\![1, m]\!]$ we have $H \neq H_i$. Then $H \cap H_i$ is an affine space of dimension less or equal to $l - 2$ so it has Hausdorff

dimension smaller or equal to $l-2$. A finite union of sets of Hausdorff dimension smaller or equal to $l-2$ has Hausdorff dimension smaller or equal to $l-2$. Thus, $B(x,r)\cap H = B(x,r)\cap\bigcup_{i=1}^{m}(H\cap H_i)$ has Hausdorff dimension smaller or equal to $l-2$, which is absurd since $x\in H$ so $B(x,r)\cap H$ has Hausdorff dimension $l-1$. Hence there exists $i\in[\![1,m]\!]$ such that $H=H_i$.

We have shown

$$\mathcal{H}\subset\{H_i, i\in[\![1,m]\!]\}. \tag{3.C.7}$$

– **Let us show** $\{H_i, i\in[\![1,m]\!]\}\subset\mathcal{H}$**.**

Let $i\in[\![1,m]\!]$. Let us prove $H_i\in\mathcal{H}$.

First, by condition *C.b)* we know that $E_i\cap h^{lin}(\mathring{\Omega})\neq\emptyset$, so there exists $x\in\mathring{\Omega}$ such that $h^{lin}(x)\in E_i$. Since $h^{lin}(x)=Mx+b$ and E_i is the space of vectors whose i^{th} coordinate is 0, this is equivalent to

$$M_{i,\cdot}x+b_i=0,$$

or said otherwise $x\in H_i$. This proves that $H_i\cap\mathring{\Omega}\neq\emptyset$. We still need to prove $H_i\cap\mathring{\Omega}\subset\overline{X}$.

Let $x\in H_i\cap\mathring{\Omega}$. Let us prove $x\in\overline{X}$.

Since M is full row rank, the line vectors $M_{1,\cdot},\dots,M_{m,\cdot}$ are linearly independent, and thus for all $k\in[\![1,m]\!]\backslash\{i\}$, $H_k\cap H_i$ has Hausdorff dimension smaller or equal to $l-2$.

Proposition 33 shows that $\bigcup_{D\in\Pi}\partial h^{-1}(D)$ is contained in a finite union of hyperplanes $\bigcup_{k=1}^{s}A_k$. Let $I=\{k\in[\![1,s]\!]\ ,\ A_k=H_i\}$. For all $k\in[\![1,s]\!]\backslash I$, $A_k\cap H_i$ is either empty, or an intersection of two non parallel hyperplanes, in both cases it is an affine space of dimension smaller than $l-2$.

Thus,

$$H_i\cap\left((\bigcup_{k\neq i}H_k)\cup(\bigcup_{k\notin I}A_k)\right)$$

has Hausdorff dimension strictly smaller than $l-1$, so for any $r>0$ there exists

$$y\in B(x,r)\cap H_i\cap\mathring{\Omega}\backslash\left((\bigcup_{k\neq i}H_k)\cup(\bigcup_{k\notin I}A_k)\right). \tag{3.C.8}$$

In the rest of the proof, we show that such a y is an element of X. Once this is established, since it is true for all $r>0$, we conclude that $x\in\overline{X}$ and therefore $H_i\in\mathcal{H}$.

If there exists $D\in\Pi$ such that $y\in\overset{\circ}{\overbrace{h^{-1}(D)}}$, then

$$y\in H_i\cap\overset{\circ}{\overbrace{h^{-1}(D)}}\cap\mathring{\Omega}\backslash\left(\bigcup_{k\neq i}H_k\right)$$

therefore we can use Proposition 64 to conclude that f is not differentiable at y.

Otherwise we can use Lemma 66 to find $R_1 > 0$, D_- and $D_+ \in \Pi$ such that

$$\begin{aligned} B(y, R_1) \cap \{z \in \mathbb{R}^l, M_{i,.}z + b_i < 0\} &\subset h^{-1}(D_-) \\ B(y, R_1) \cap \{z \in \mathbb{R}^l, M_{i,.}z + b_i > 0\} &\subset h^{-1}(D_+). \end{aligned}$$

Since for all $j \neq i$, $y \notin H_j$ and since these hyperplanes are closed, there exists $R_2 > 0$ such that for all $j \neq i$, $B(y, R_2) \cap H_j = \emptyset$. Let $R = \min(R_1, R_2)$ and denote $B_- = B(y, R) \cap \{z \in \mathbb{R}^l, M_{i,.}z + b_i < 0\}$ and $B_+ = B(y, R) \cap \{z \in \mathbb{R}^l, M_{i,.}z + b_i > 0\}$.

For all $z \in B_-$, using (3.C.2) with the fact that $s_i(z) = 0$ and $s_k(z) = s_k(y)$ for all $k \neq i$, we have

$$f(z) = \sum_{k \neq i} V_{.,k}(D_-) s_k(y)(M_{k,.}z + b_k) + c(D_-). \tag{3.C.9}$$

For all $z \in B_+$, using this time that $s_i(z) = 1$, we have

$$f(z) = \sum_{k \neq i} V_{.,k}(D_+) s_k(y)(M_{k,.}z + b_k) + c(D_+) + V_{.,i}(D_+)(M_{i,.}z + b_i). \tag{3.C.10}$$

If f was differentiable at y, we would derive from (3.C.9) the expression of the Jacobian matrix

$$J_f(y) = \sum_{k \neq i} V_{.,k}(D_-) s_k(y) M_{k,.}, \tag{3.C.11}$$

but we would also derive from (3.C.10) the expression

$$J_f(y) = \sum_{k \neq i} V_{.,k}(D_+) s_k(y) M_{k,.} + V_{.,i}(D_+) M_{i,.}, \tag{3.C.12}$$

hence subtracting (3.C.11) to (3.C.12) we would find

$$\sum_{k \neq i} (V_{.,k}(D_+) - V_{.,k}(D_-)) s_k(y) M_{k,.} + V_{.,i}(D_+) M_{i,.} = 0.$$

Since M is full row rank, this would imply that $V_{.,i}(D_+) = 0$.

However since $h^{-1}(D_+)$ is closed and contains B_+, we have $y \in \overline{B_+} \subset h^{-1}(D_+)$. Recalling (3.C.8) we thus have

$$y \in H_i \cap h^{-1}(D_+) \cap \mathring{\Omega},$$

thus

$$h(y) \in E_i \cap D_+ \cap h(\mathring{\Omega}),$$

which shows the latter intersection is not empty. By assumption $C.c$) this implies that $V_{.,i}(D_+) \neq 0$, which is a contradiction. Therefore f is not differentiable at y.

As a conclusion, we have showed that for all $r > 0$, there exists $y \in B(x, r)$ such that f is not differentiable at y and $y \in \mathring{\Omega}$. In other words, $x \in \overline{X}$.

Since x is arbitrary in $H_i \cap \mathring{\Omega}$, we have shown that for all $i \in [\![1, m]\!]$,

$$H_i \cap \mathring{\Omega} \subset \overline{X},$$

i.e., since we have already shown that $H_i \cap \mathring{\Omega} \neq \emptyset$,

$$H_i \in \mathcal{H}.$$

Finally $\{H_i, i \in [\![1,m]\!]\} \subset \mathcal{H}$, and, using (3.C.7),

$$\mathcal{H} = \{H_i, i \in [\![1,m]\!]\}.$$

□

Proposition 68. *For all $i \in [\![1,m]\!]$, there exists $\epsilon_{\varphi^{-1}(i)} \in \{-1,1\}$ such that*

$$\tilde{M}_{i,.} = \epsilon_{\varphi^{-1}(i)} M_{\varphi^{-1}(i),.} \qquad \text{and} \qquad \tilde{b}_i = \epsilon_{\varphi^{-1}(i)} b_{\varphi^{-1}(i)}.$$

Proof. Let $i \in [\![1,m]\!]$. We know that $\tilde{H}_i = H_{\varphi^{-1}(i)}$, so the equations $\tilde{M}_{i,.}x + \tilde{b}_i = 0$ and $M_{\varphi^{-1}(i),.}x + b_{\varphi^{-1}(i)} = 0$ define the same hyperplanes. This is only possible if the parameters of the equation are proportional (but nonzero): there exists $\epsilon_{\varphi^{-1}(i)} \in \mathbb{R}^*$ such that $\tilde{M}_{i,.} = \epsilon_{\varphi^{-1}(i)} M_{\varphi^{-1}(i),.}$ and $\tilde{b}_i = \epsilon_{\varphi^{-1}(i)} b_{\varphi^{-1}(i)}$. But since $\|\tilde{M}_{i,.}\| = \|M_{\varphi^{-1}(i),.}\| = 1$ by hypothesis, we necessarily have $\epsilon_{\varphi^{-1}(i)} \in \{-1,1\}$. □

Proposition 69. *For all $i \in [\![1,m]\!]$,*

- *$\tilde{M}_{i,.} = M_{\varphi^{-1}(i),.}$;*
- *$\tilde{b}_i = b_{\varphi^{-1}(i)}$.*

Proof. By Proposition 68, we know that there exists $(\epsilon_i)_{1\leq i\leq m} \in \{-1,1\}^m$ such that for all $i \in [\![1,m]\!]$,

$$\tilde{M}_{i,.} = \epsilon_{\varphi^{-1}(i)} M_{\varphi^{-1}(i),.} \qquad \text{and} \qquad \tilde{b}_i = \epsilon_{\varphi^{-1}(i)} b_{\varphi^{-1}(i)}. \tag{3.C.13}$$

We need to prove that for all $i \in [\![1,m]\!]$, $\epsilon_{\varphi^{-1}(i)} = 1$.

Let $i \in [\![1,m]\!]$.

Applying Proposition 33 to h and Π, we see that $\bigcup_{D\in\Pi} \partial h^{-1}(D)$ is contained in a finite union of hyperplanes $\bigcup_{k=1}^{s} A_k$. Applying it to $\tilde{h}$ and $\tilde{\Pi}$, we see similarly that $\bigcup_{\tilde{D}\in\tilde{\Pi}} \partial \tilde{h}^{-1}(\tilde{D})$ is contained in a finite union of hyperplanes $\bigcup_{k=1}^{r} B_k$.

Let $I = \{k \in [\![1,s]\!] \ , \ A_k = H_i\}$ and $J = \{k \in [\![1,r]\!] \ , \ B_k = H_i\}$. For all $k \in [\![1,s]\!]\backslash I$, since $A_k \neq H_i$, $A_k \cap H_i$ is either empty, or an intersection of two non parallel hyperplanes, in both cases it is an affine space of dimension smaller than $l-2$. The same applies for all $k \in [\![1,r]\!]\backslash J$ to $B_k \cap H_i$. For all $j \neq i$, $H_j \neq H_i$ so $H_j \cap H_i$ is also an affine space of dimension smaller than $l-2$. Since $H_i \cap \mathring{\Omega}$ is nonempty by **C**.*b*), we can thus find a vector

$$x \quad \in \quad \mathring{\Omega} \ \cap \ H_i \setminus \left((\bigcup_{k\notin I} A_k) \cup (\bigcup_{k\notin J} B_k) \cup (\bigcup_{j\neq i} H_j) \right).$$

Applying Lemma 66 with Π, h, H_i and $(M_{i,.}, b_i)$, we find $r_1 > 0$, D_- and $D_+ \in \Pi$ such that

$$\begin{aligned} B(x,r_1) \cap \{y \in \mathbb{R}^l, M_{i,.}y + b_i < 0\} \quad &\subset \quad h^{-1}(D_-) \\ B(x,r_1) \cap \{y \in \mathbb{R}^l, M_{i,.}y + b_i > 0\} \quad &\subset \quad h^{-1}(D_+). \end{aligned} \tag{3.C.14}$$

Applying the same lemma with $\tilde{\Pi}$, $\tilde{h}$, H_i and $(M_{i,.}, b_i)$ we find $r_2 > 0$, $\tilde{D}_-$ and $\tilde{D}_+ \in \tilde{\Pi}$ such that

$$\begin{aligned} B(x, r_2) \cap \{y \in \mathbb{R}^l, M_{i,.}y + b_i < 0\} &\subset \tilde{h}^{-1}(\tilde{D}_-) \\ B(x, r_2) \cap \{y \in \mathbb{R}^l, M_{i,.}y + b_i > 0\} &\subset \tilde{h}^{-1}(\tilde{D}_+). \end{aligned} \tag{3.C.15}$$

Since the hyperplanes H_j are closed, we can also find $r_3 > 0$ such that for all $j \neq i$, $B(x, r_3) \cap H_j = \emptyset$. Taking $r = \min(r_1, r_2, r_3)$ and denoting

$$B_+ = B(x, r) \cap \{y \in \mathbb{R}^l, M_{i,.}y + b_i > 0\},$$

we derive from (3.C.14) and (3.C.15) that

$$B_+ \subset h^{-1}(D_+) \cap \tilde{h}^{-1}(\tilde{D}_+).$$

Since $r \leq r_3$, we have $B_+ \cap \left(\bigcup_{j \neq i} H_j\right) = \emptyset$, and by definition $B_+ \cap \{y \in \mathbb{R}^l, M_{i,.}y + b_i = 0\} = \emptyset$, so $B_+ \cap H_i = \emptyset$. We have $B_+ \cap \left(\bigcup_{j=1}^m H_j\right) = \emptyset$, so for all $j \in [\![1, m]\!]$, there exist $\delta_j \in \{0, 1\}$ such that for all $y \in B_+$, $s_j(y) = \delta_j$. We have $\bigcup_{j=1}^m \tilde{H}_j = \bigcup_{j=1}^m H_j$ so similarly, $B_+ \cap \bigcup_{j=1}^m \tilde{H}_j = \emptyset$ and there exists $\tilde{\delta}_j \in \{0, 1\}$ such that for all $j \in [\![1, m]\!]$, for all $y \in B_+$, $\tilde{s}_j(y) = \tilde{\delta}_j$.

For all $y \in B_+$, we thus have, using (3.C.2),

$$\sum_{j=1}^m V_{.,j}(D_+)\delta_j \left(M_{j,.}y + b_j\right) + c(D_+) = \sum_{j=1}^m \tilde{V}_{.,j}(\tilde{D}_+)\tilde{\delta}_j \left(\tilde{M}_{j,.}y + \tilde{b}_j\right) + \tilde{c}(\tilde{D}_+).$$

B_+ is a nonempty open set so we have the equality

$$\begin{aligned} \sum_{j=1}^m V_{.,j}(D_+)\delta_j M_{j,.} &= \sum_{j=1}^m \tilde{V}_{.,j}(\tilde{D}_+)\tilde{\delta}_j \tilde{M}_{j,.} \\ &= \sum_{j=1}^m \tilde{V}_{.,j}(\tilde{D}_+)\tilde{\delta}_j \epsilon_{\varphi^{-1}(j)} M_{\varphi^{-1}(j),.} \\ &= \sum_{j=1}^m \tilde{V}_{.,\varphi(j)}(\tilde{D}_+)\tilde{\delta}_{\varphi(j)} \epsilon_j M_{j,.}. \end{aligned} \tag{3.C.16}$$

The condition **C**.*a*) states that M is full row rank, so the vectors $M_{j,.}$ are linearly independent. Applied to (3.C.16), this information yields, for all $j \in [\![1, m]\!]$,

$$V_{.,j}(D_+)\delta_j = \tilde{V}_{.,\varphi(j)}(\tilde{D}_+)\tilde{\delta}_{\varphi(j)}\epsilon_j,$$

and in particular,

$$V_{.,i}(D_+)\delta_i = \tilde{V}_{.,\varphi(i)}(\tilde{D}_+)\tilde{\delta}_{\varphi(i)}\epsilon_i. \tag{3.C.17}$$

Since $h^{-1}(D_+)$ and $\tilde{h}^{-1}(\tilde{D}_+)$ are closed, we have

$$\overline{B_+} \subset h^{-1}(D_+) \cap \tilde{h}^{-1}(\tilde{D}_+),$$

and since $x \in \overline{B_+}$, we have $h^{-1}(D_+) \cap H_i \neq \emptyset$ and $\tilde{h}^{-1}(\tilde{D}_+) \cap H_i \neq \emptyset$. The condition *C.c)* implies that $V_{.,i}(D_+) \neq 0$ and $\tilde{V}_{.,\varphi(i)}(\tilde{D}_+) \neq 0$ (recall that $H_i = \tilde{H}_{\varphi(i)}$). We also have $\epsilon_i \neq 0$, so from (3.C.17) we obtain

$$\delta_i = 0 \Leftrightarrow \tilde{\delta}_{\varphi(i)} = 0.$$

By definition, the coefficient δ_i depends on the sign of $M_{i,.}y + b_i$: if $M_{i,.}y + b_i$ is positive, $\delta_i = 1$ and if $M_{i,.}y + b_i$ is negative then $\delta_i = 0$ ($M_{i,.}y + b_i$ cannot be equal to zero since $y \notin H_i$). The coefficient $\tilde{\delta}_{\varphi(i)}$ depends similarly on the sign of $\tilde{M}_{\varphi(i),.}y + \tilde{b}_{\varphi(i)}$. Thus, $M_{i,.}y + b_i$ and $\tilde{M}_{\varphi(i),.}y + \tilde{b}_{\varphi(i)}$ have same sign.

Since $\epsilon_i \in \{-1, 1\}$ and

$$\tilde{M}_{\varphi(i),.}y + \tilde{b}_{\varphi(i)} = \epsilon_i M_{i,.}y + \epsilon_i b_i = \epsilon_i \left(M_{i,.}y + b_i\right),$$

we conclude that $\epsilon_i = 1$.

□

We can now finish the proof of Lemma 63. It results from the above that:

$$\tilde{M} = P_\varphi M$$

$$\tilde{b} = P_\varphi b.$$

We have by hypothesis, for all $x \in \Omega$,

$$\tilde{g}(\sigma(\tilde{M}x + \tilde{b})) = g(\sigma(Mx + b)),$$

but since $\tilde{M} = P_\varphi M$ and $\tilde{b} = P_\varphi b$ we also have:

$$\tilde{g}(\sigma(\tilde{M}x + \tilde{b})) = \tilde{g}(\sigma(P_\varphi Mx + P_\varphi b)) = \tilde{g}(P_\varphi \sigma(Mx + b)).$$

Combining these, we have for all $x \in \Omega$,

$$\tilde{g} \circ P_\varphi(h(x)) = g(h(x)),$$

i.e. $\tilde{g} \circ P_\varphi$ and g coincide on $h(\Omega)$.

Chapter 4

Local identifiability of deep ReLU neural networks: the theory

This chapter consists in the article [28], which is a joint work with François Bachoc and François Malgouyres, and which was published at *Neurips 2022*.

Abstract

Is a sample rich enough to determine, at least locally, the parameters of a neural network? To answer this question, we introduce a new local parameterization of a given deep ReLU neural network by fixing the values of some of its weights. This allows us to define local lifting operators whose inverses are charts of a smooth manifold of a high dimensional space. The function implemented by the deep ReLU neural network composes the local lifting with a linear operator which depends on the sample. We derive from this convenient representation a geometric necessary and sufficient condition of local identifiability. Looking at tangent spaces, the geometric condition provides: 1/ a sharp and testable necessary condition of identifiability and 2/ a sharp and testable sufficient condition of local identifiability. The validity of the conditions can be tested numerically using backpropagation and matrix rank computations.

Contents

4.1 Introduction

4.1.1 Context and motivations

Neural networks are famous for their capacity to perform complex tasks in a wide variety of domains such as image classification [103], object recognition [156, 157], speech recognition [89, 164, 87], natural language processing [124, 123, 97], anomaly detection [148] or climate sciences [3].

The following properties of the parameters of neural networks have recently drawn attention: identifiability, inverse stability and stable recovery; from weaker to stronger. Let $f_\theta(X)$ be the outputs of a network parameterized by the parameters θ, for given inputs X. Global identifiability means that if $f_\theta(X) = f_{\tilde{\theta}}(X)$ then $\theta = \tilde{\theta}$, up to identified invariances, for instance neuron permutation and rescaling for ReLU networks. Local identifiability restricts this analysis for θ and $\tilde{\theta}$ sufficiently close. Then, inverse stability means that the distance between θ and $\tilde{\theta}$ (up to invariances) is bounded by a function of the distance between $f_\theta(X)$ and $f_{\tilde{\theta}}(X)$. Finally, stable recovery consists in obtaining an algorithm to approximately recover θ from a noisy version of $f_\theta(X)$, with quantitative guarantees. In all cases, we must distinguish between statements for X being a finite list of inputs, in which case we would like X to be small, and for infinite X (for instance determining θ from the entire function f_θ).

Identifiability from finite X, which is the focus of this paper, is important for different reasons. In the first place, model extraction attacks for neural networks have been a growing topic over the last years. Indeed, some algorithms are able to recover in practice the parameters of a neural network from queries [39, 159]. This can be a concern since neural network providers may wish to keep these parameters secret, for security [104], for privacy [67, 40], or for intellectual property [196].

A way of preventing such a recovery can be by guaranteeing that identifiability does not hold, that is to check that a necessary condition of identifiability is not met. On the opposite side, guaranteeing that identifiability holds is interesting in the position of an attacker. If the attacker has access to X, to $f_\theta(X)$, and is able to compute a $\tilde{\theta}$ such that $f_{\tilde{\theta}}(X) = f_\theta(X)$, the question then becomes: does this guarantee that $\tilde{\theta} = \theta$ or shall the attacker expand X with new queries? The attacker needs a sufficient condition of identifiability.

Another important motivation for identifiability is having a better understanding and control of neural networks. Indeed, if the learning sample has the form $(X, f_\theta(X))$, with θ the parameters of a teaching network, global identifiability from X means that the global minimizer of the empirical risk is unique. In this case, if the global minimizer is reached, there will typically be no variability due to the optimization parameters (choice of the algorithm, number of epochs,...) and to stochasticity (for stochastic optimizers). Even if very recent works on double descent phenomena, e.g. [24], highlight a benefit of overparameterization (thus absence of identifiability) for increasing prediction performances, a user may be interested in a small enough number of parameters to retain identifiability, if the loss of performance is mild compared to overparameterization.

Note that, of course, global identifiability is more relevant than local identifiability to the above motivations. This work nevertheless focuses on local identifiability, which is a necessary condition for global identifiability, and which analysis can be a first step to analyzing global identifiability. Local identifiability is also arguably insightful on the geometry of the relationship between the parameter space of θ and its image $\{f_\theta(X),\ \theta \text{ varies}\}$. Note that most existing identifiability, inverse stability and stable recovery results (see the next section) are also local.

4.1.2 Existing work on identifiability, inverse stability, stable recovery and attacks

Identifiability: Even though it has regained interest recently, the question of identifiability for neural networks is not new. Indeed, in the 1990s, some positive results of identifiability for networks with smooth activation functions (tanh, logistic sigmoid or Gaussian for instance) have been established [181, 4, 105, 96, 61]. These results are mainly theoretical, they concern activation functions which are not the most used nowadays (in particular, they do not apply to ReLU networks), and assume full knowledge of the function f_θ implemented by the network, which is impossible in practice.

When it comes to ReLU, for shallow [145, 178] as well as deep [147, 26] neural networks, some positive results of identifiability have been recently established. They show that under some conditions on the architecture and parameters of the network, the function implemented by the network uniquely characterizes its parameters, up to neuron permutation and rescaling operations. Although they apply to ReLU networks, these results share a limitation with those of previous paragraph: they assume the function implemented by the network to be known on the whole input space, or at least on an open subset of it.

As far as we know, there exists only one identifiability result for deep ReLU networks assuming the knowledge of f_θ on a *finite* sample only. [179] give a theoretical condition for the existence of a finite set which locally identifies the parameters of a deep neural network. It is an *existence* result: it does not concretely provide such a finite set, nor does it allow to test local identifiability for any finite sample, as we propose in this work. The construction in [179] shares similarities with previous

works on deep structured matrix factorization [119, 117, 115]. The present article also lies in this line of research.

Inverse stability and stable recovery: Closely related to identifiability are the topics of inverse stability and stable recovery of the parameters of a network. Some negative [144] as well as positive [58, 119, 117, 115] results of inverse stability exist. The articles [119, 117, 115] examine the case of structured networks with the identity as activation function. Only [115] considers a finite X. The authors of [58] consider a general class of networks amongst which ReLU networks, but the result only holds for one-hidden-layer neural networks. Furthermore this result also requires the knowledge of f_θ on a whole domain.

Several stable recovery algorithms have also been proposed, for one-hidden-layer neural networks in a first place, for smooth activation function [68], as well as ReLU in the fully-connected case [70, 200, 201, 203] or in the convolutional case [32, 199]. These references consider a finite X but provide a large sample complexity under which a smartly constructed initialization followed by a first order algorithm allows to stably recover the parameters of the network.

For deep networks, some stable recovery algorithms also exist, for instance for Heavyside activation function [8], or for only recovering the first layer with sparsity assumptions [168] in the ReLU case, but to the best of our knowledge there does not exist any algorithm recovering fully a deep ReLU network from a finite sample.

Model inversion attacks: For deep ReLU networks, when one has full access to the function implemented by the network, a practical algorithm [159] sequentially constructs a sample X and approximately recovers the architecture and the parameters modulo permutation and rescaling. Similarly, formulating the problem as a cryptanalytic problem, [39] reconstructs a functionally equivalent network with fewer requests. As mentioned in Section 4.1.1, these two references are related to identifiability, but consider a different setting. In this article we consider an arbitrary given X, while they work mostly on its construction.

4.1.3 Contributions

1/ We establish a necessary and sufficient geometric condition of local identifiability from a finite sample X for deep fully-connected ReLU networks. The condition is that the intersection between an affine space and a smooth manifold is reduced to a single point. See Figure 4.1 for an illustration.

2/ Considering tangent spaces, we then provide a computable necessary condition of local identifiability from a finite sample X. Since global identifiability implies local identifiability, it is also a computable necessary condition of global identifiability.

3/ We also establish a computable sufficient condition of local identifiability, which is close to the necessary condition. To the best of our knowledge, these are the first

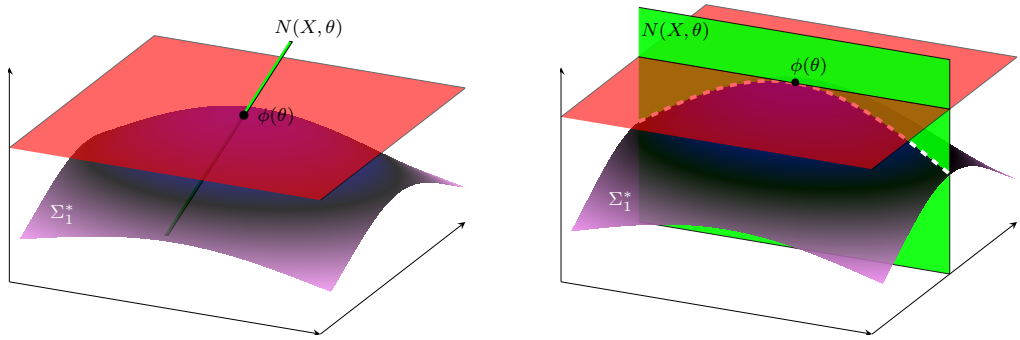

Figure 4.1 – The local intersection between the affine space $N(X,\theta)$ (in green) and the smooth manifold Σ_1^* (color gradient). We also represent in red the tangent space to Σ_1^* at $\phi(\theta)$. Left: The identifiable case. The intersection is reduced to $\{\phi(\theta)\}$. Right: The non identifiable case. The intersection, represented with a dashed white line, is not reduced to $\{\phi(\theta)\}$.

testable conditions of local identifiability for any finite input sample. In particular, [179] provides a theoretical condition equivalent to the existence of a finite sample for which local identifiability holds, but does not provide the sample explicitly, nor does it characterize local identifiability for any arbitrary sample.

4/ To prove these results, we develop geometric tools which can be of independent interest for theoretically understanding deep ReLU networks as well as for possible applications. Namely, we introduce local reparameterizations ρ_θ of the network by fixing some weight values as constants. Building on these local parameterizations, we introduce local lifting operators ψ^θ and we decompose the function implemented by the network $f_\theta(x)$ as a composition of ψ^θ, which only depends on the parameters, and a piecewise constant operator α which depends on θ and the inputs x^i. For almost any parameterization θ, the operator α is constant in a neighborhood of θ and consists in applying a linear function to ψ^θ. We show that in fact, the operators ψ^θ are the inverses of coordinate charts of a smooth manifold Σ_1^*, contained in a high dimensional space. We find Σ_1^* to be of particular interest in representing geometrically some properties of the network parameters (in particular to establish 1/, 2/ and 3/ above).

4.1.4 Overview of the article

This work is structured as follows. We start by introducing basic tools and already known results, and we state the definition of local identifiability in Section 4.2. We then introduce the local parameterizations ρ_θ and the set Σ_1^*, and we show that the latter is a smooth manifold in Section 4.3. This allows us to state our main results in Section 4.4, that is the geometric and the numerically testable conditions of local identifiability. Finally we discuss in Section 4.5 the numerical computations needed to test the latter conditions. All the proofs are provided in the appendices.

4.2 ReLU networks, lifting operator and rescaling of the parameters

4.2.1 ReLU networks

Let us introduce our notations for deep fully-connected ReLU networks. In this paper, a network is a graph (E, V) of the following form.

— V is a set of neurons, which is divided in $L+1$ layers, with $L \geq 2$: $V = (V_l)_{l \in [\![0,L]\!]}$.

V_0 is the input layer, V_L the output layer and the layers V_l with $1 \leq l \leq L-1$ are the hidden layers. Using the notation $|C|$ for the cardinal of a finite set C, we denote, for all $l \in [\![0, L]\!]$, $N_l = |V_l|$ the size of the layer V_l.

— E is the set of all oriented edges $v \to v'$ between neurons in consecutive layers, that is

$$E = \{v \to v',\ v \in V_l, v' \in V_{l+1}, \text{for } l \in [\![0, L-1]\!]\}.$$

A network is parameterized by weights and biases, gathered in its parameterization θ, with

$$\theta = ((w_{v\to v'})_{v\to v' \in E}, (b_v)_{v \in B}) \quad \in \mathbb{R}^E \times \mathbb{R}^B,$$

where $B = \bigcup_{l=1}^{L} V_l$. It is also convenient to consider the weights and biases in matrix/vector form: for a given θ, we denote, for $l \in [\![1, L]\!]$,

$$W_l = (w_{v\to v'})_{v' \in V_l, v \in V_{l-1}} \in \mathbb{R}^{N_l \times N_{l-1}} \qquad \text{and} \qquad b_l = (b_v)_{v \in V_l} \in \mathbb{R}^{N_l}.$$

When dealing with two parameterizations θ and $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, we take as a convention that $w_{v\to v'}$ and b_v as well as W_l and b_l denote the weights and biases associated to θ, and $\tilde{w}_{v\to v'}$ and $\tilde{b}_v$ as well as $\widetilde{W}_l$ and $\tilde{b}_l$ denote those associated to $\tilde{\theta}$.

The activation function, denoted σ, is always ReLU: for any $p \in \mathbb{N}^*$ and any vector $x = (x_1, \ldots, x_p)^T \in \mathbb{R}^p$, it is defined as $\sigma(x) = (\max(x_1, 0), \ldots, \max(x_p, 0))^T$.

For a given θ, we define recursively $f_l : \mathbb{R}^{V_0} \to \mathbb{R}^{V_l}$ (we omit the dependency in θ in the notation for simplicity), for $l \in [\![0, L]\!]$, by

— $\forall x \in \mathbb{R}^{V_0}, \qquad f_0(x) = x$;

— $\forall l \in [\![1, L-1]\!], \forall x \in \mathbb{R}^{V_0}, \qquad f_l(x) = \sigma\left(W_l f_{l-1}(x) + b_l\right)$;

— $\forall x \in \mathbb{R}^{V_0}, \qquad f_L(x) = W_L f_{L-1}(x) + b_L$.

We define the function $f_\theta : \mathbb{R}^{V_0} \to \mathbb{R}^{V_L}$ implemented by the network of parameter θ as $f_\theta = f_L$.

4.2.2 The lifting operator ϕ and the activation operator α

For a fixed $x \in \mathbb{R}^{V_0}$, the value of $f_\theta(x)$ is a non-linear function of θ. The goal of this section is to obtain a higher-dimensional representation of θ, that will be written $\phi(\theta)$, and such that $f_\theta(x)$ is locally a linear function of $\phi(\theta)$. This will be achieved with Proposition 70. The function ϕ is called a lifting operator, a wording borrowed from category theory and commonly used in compressed sensing and dictionary

learning, for instance in [36]. The components of $\phi(\theta)$ will be associated to paths in the neural network. Linearity in Proposition 70 will correspond to summing over these paths.

We now introduce the paths notations. For all $l \in [\![0, L-1]\!]$, we define

$$\mathcal{P}_l = V_l \times \cdots \times V_{L-1},$$

which is the set of all paths in the network starting from layer l and ending in layer $L-1$. We consider an additional element β which can be interpreted as an empty path and whose role will be clear once ϕ has been defined and Proposition 70 stated. We define

$$\mathcal{P} = \left(\bigcup_{l=0}^{L-1} \mathcal{P}_l \right) \cup \{\beta\}.$$

In a similar way to [179], we can now define the above-mentioned 'lifting operator'

$$\begin{array}{rccc} \phi: & \mathbb{R}^E \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^{\mathcal{P} \times V_L} \\ & \theta & \longmapsto & (\phi_{p,v}(\theta))_{p \in \mathcal{P}, v \in V_L} \end{array} \tag{4.2.1}$$

by:

— for all $l \in [\![0, L-1]\!]$ and all $p = (v_l, \ldots, v_{L-1}) \in \mathcal{P}_l$, and for all $v_L \in V_L$,

$$\phi_{p,v_L}(\theta) = \begin{cases} \prod_{l'=0}^{L-1} w_{v_{l'} \to v_{l'+1}} & \text{if } l = 0 \\ b_{v_l} \prod_{l'=l}^{L-1} w_{v_{l'} \to v_{l'+1}} & \text{if } l \geq 1; \end{cases}$$

— for $p = \beta$ and $v_L \in V_L$, $\quad \phi_{\beta,v_L}(\theta) = b_{v_L}$.

To define the activation operator, we first define, for all $l \in [\![1, L-1]\!]$, all $v \in V_l$, all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and $x \in \mathbb{R}^{V_0}$,

$$a_v(x, \theta) = \begin{cases} 1 & \text{if } (W_l f_{l-1}(x) + b_l)_v \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

which is the activation indicator of neuron v. We then define the 'activation operator'

$$\begin{array}{rccc} \alpha: & \mathbb{R}^{V_0} \times \left(\mathbb{R}^E \times \mathbb{R}^B \right) & \longrightarrow & \mathbb{R}^{1 \times \mathcal{P}} \\ & (x, \theta) & \longmapsto & (\alpha_p(x, \theta))_{p \in \mathcal{P}} \end{array} \tag{4.2.2}$$

by:

— for all $l \in [\![0, L-1]\!]$ and all $p = (v_l, \ldots, v_{L-1}) \in \mathcal{P}_l$:

$$\alpha_p(x, \theta) = \begin{cases} x_{v_0} \prod_{l'=1}^{L-1} a_{v_{l'}}(x, \theta) & \text{if } l = 0 \\ \prod_{l'=l}^{L-1} a_{v_{l'}}(x, \theta) & \text{if } l \geq 1; \end{cases}$$

— for $p = \beta$, $\quad \alpha_\beta(x, \theta) = 1$.

We then have the announced linear representation of the function f_θ implemented by the network.

Proposition 70. *For all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and all $x \in \mathbb{R}^{V_0}$, $\quad f_\theta(x)^T = \alpha(x, \theta)\phi(\theta)$.*

This result, which is proven in Appendix 4.A.2, is for instance also stated in [179, Sec. 4] with slightly different notations. Note that each component of the vector $f_\theta(x)$ above is written as a sum over a (very large) number of paths.

Let us reformulate Proposition 70 with several inputs. We consider, for some $n \in \mathbb{N}^*$, some given inputs $x^i \in \mathbb{R}^{V_0}$, with $i \in [\![1, n]\!]$. We denote by $X \in \mathbb{R}^{n \times V_0}$ the matrix whose lines are the transpose $(x^i)^T$ of the inputs. For all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, we denote by $f_\theta(X) \in \mathbb{R}^{n \times V_L}$ the matrix whose lines are the transpose $f_\theta(x^i)^T$ of the corresponding outputs. We also denote by $\alpha(X, \theta) \in \mathbb{R}^{n \times \mathcal{P}}$ the matrix whose lines are the line vectors $\alpha(x^i, \theta)$. Using Proposition 70 for all the x^i, we have the relation

$$f_\theta(X) = \alpha(X, \theta)\phi(\theta). \tag{4.2.3}$$

We prove in Appendix 4.A.2 the next proposition, which states that $\theta \mapsto \alpha(X, \theta)$ is piecewise constant.

Proposition 71. *For all $n \in \mathbb{N}^*$, for all $X \in \mathbb{R}^{n \times V_0}$, the mapping*

$$\begin{array}{rccc} \alpha_X : & \mathbb{R}^E \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^{n \times \mathcal{P}} \\ & \theta & \longmapsto & \alpha(X, \theta) \end{array}$$

is piecewise-constant, with a finite number of pieces. Furthermore, the boundary of each piece has Lebesgue measure zero. We call Δ_X the union of all these boundaries. The set $\Delta_X \subset \mathbb{R}^E \times \mathbb{R}^B$ is closed and has Lebesgue measure zero.

As discussed before, for a given $X \in \mathbb{R}^{n \times V_0}$, when studying the function $\theta \mapsto f_\theta(X)$, Proposition 71 alongside (4.2.3) shows that on a piece over which α_X is constant, $f_\theta(X)$ depends linearly on $\phi(\theta)$. Since Δ_X is closed with measure zero, for almost all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, there exists a neighborhood of $\tilde{\theta}$ over which α_X is constant. As noted for instance by [179, Sec. 2], for any θ in such a neighborhood, we thus have

$$f_\theta(X) - f_{\tilde{\theta}}(X) = \alpha(X, \tilde{\theta}) \left(\phi(\theta) - \phi(\tilde{\theta}) \right). \tag{4.2.4}$$

Hence, studying ϕ will allow us to understand better how $f_\theta(X)$ locally depends on θ.

4.2.3 Invariant rescaling operations on θ

Some well-known rescaling operations on the parameters θ do not affect the value of $\phi(\theta)$. Before detailing them, let us define, for all $t \in \mathbb{R}$, the sign indicator $\mathrm{sign}(t)$ as $1, 0$ or -1 depending on whether $t > 0$, $t = 0$ or $t < 0$ respectively. For any $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, we then define

$$\mathrm{sign}(\theta) = \left((\mathrm{sign}(w_{v \to v'})_{v \to v' \in E}, (\mathrm{sign}(b_v))_{v \in B} \right) \in \{-1, 0, 1\}^E \times \{-1, 0, 1\}^B.$$

We can now describe the rescaling operations.

Definition 72. Let $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$.

— We say that θ is equivalent to $\tilde{\theta}$ modulo rescaling, and we write $\theta \overset{R}{\sim} \tilde{\theta}$ iff there exists a family of vectors $(\lambda^0, \dots, \lambda^L) \in (\mathbb{R}^*)^{V_0} \times \cdots \times (\mathbb{R}^*)^{V_L}$, with $\lambda^0 = \mathbf{1}_{V_0}$ and $\lambda^L = \mathbf{1}_{V_L}$, such that, for all $l \in [\![1, L]\!]$,

$$\begin{cases} W_l = \mathrm{Diag}(\lambda^l)\widetilde{W}_l \,\mathrm{Diag}(\lambda^{l-1})^{-1} \\ b_l = \mathrm{Diag}(\lambda^l)\tilde{b}_l. \end{cases} \tag{4.2.5}$$

— We say that θ is equivalent to $\tilde{\theta}$ modulo positive rescaling, and we write $\theta \sim \tilde{\theta}$ iff

$$\theta \overset{R}{\sim} \tilde{\theta} \quad \text{and} \quad \mathrm{sign}(\theta) = \mathrm{sign}(\tilde{\theta}).$$

For all $l \in [\![1, L]\!]$, to satisfy (4.2.5) is equivalent to satisfy, for all $(v_{l-1}, v_l) \in V_{l-1} \times V_l$,

$$\begin{cases} w_{v_{l-1}\to v_l} = \frac{\lambda^l_{v_l}}{\lambda^{l-1}_{v_{l-1}}} \tilde{w}_{v_{l-1}\to v_l} \\ b_{v_l} = \lambda^l_{v_l} \tilde{b}_{v_l}. \end{cases} \tag{4.2.6}$$

The relations $\overset{R}{\sim}$ and $\sim$ are equivalence relations on the set of parameters $\mathbb{R}^E \times \mathbb{R}^B$. The equivalence modulo positive rescaling $\sim$ is a well-known invariant for ReLU networks [178, 179, 26, 135, 194]. We have indeed the following property: if $\theta \sim \tilde{\theta}$, for all $x \in \mathbb{R}^{V_0}$,

$$f_\theta(x) = f_{\tilde{\theta}}(x). \tag{4.2.7}$$

One of the interests of the operator ϕ is that it captures this invariant, as described by [179, Sec. 2.4]. Propositions 73 and 74 are similar to their results and are restated here and proven in Appendix 4.A.2 for completeness. Indeed, combining the definition of ϕ with (4.2.6), we have the following property.

Proposition 73. *For all $\theta, \tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, we have*

$$\theta \overset{R}{\sim} \tilde{\theta} \quad \Longrightarrow \quad \phi(\theta) = \phi(\tilde{\theta}),$$

and thus in particular

$$\theta \sim \tilde{\theta} \quad \Longrightarrow \quad \phi(\theta) = \phi(\tilde{\theta}).$$

The reciprocal of Proposition 73 holds provided we exclude some degenerate cases. Let us denote, for any $l \in [\![1, L-1]\!]$ and any $v \in V_l$, by $w_{\bullet\to v}$ the vector $(w_{v'\to v})_{v'\in V_{l-1}} \in \mathbb{R}^{V_{l-1}}$ and by $w_{v\to\bullet}$ the vector $(w_{v\to v'})_{v'\in V_{l+1}} \in \mathbb{R}^{V_{l+1}}$. We define the following set, which is close to the notion of 'non admissible parameter' in [179]:

$$S = \{\theta \in \mathbb{R}^E \times \mathbb{R}^B, \exists v \in V_1 \cup \cdots \cup V_{L-1},\ w_{v\to\bullet} = 0 \quad \text{or} \quad (w_{\bullet\to v}, b_v) = (0, 0)\}.$$

When $w_{v\to\bullet} = 0$, all the outward weights of v are zero. When $(w_{\bullet\to v}, b_v) = (0, 0)$, all the inward weights as well as the bias of v are zero, so for any input the information flowing through neuron v is always zero. In both cases, the neuron v does not contribute to the output and could be removed from the network without changing the function f_θ. Since the set S is a finite union of linear subspaces of codimension

larger than 1, it is closed and has Lebesgue measure zero. We can thus exclude the degenerate cases in S without loss of generality. Proposition 74 states that the reciprocal of Proposition 73 holds over $\left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$.

Proposition 74. *For all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$, for all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$,*

$$\phi(\theta) = \phi(\tilde{\theta}) \quad \Longrightarrow \quad \theta \overset{R}{\sim} \tilde{\theta}.$$

4.2.4 Local identifiability

We have now introduced all the concepts used in the formal definition of 'local identifiability'.

Definition 75. Let $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \mathbb{R}^E \times \mathbb{R}^B$. We say that θ is *locally identifiable from X* if there exists $\epsilon > 0$ such that for all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, if $\|\theta - \tilde{\theta}\|_\infty < \epsilon$,

$$f_\theta(X) = f_{\tilde{\theta}}(X) \quad \Longrightarrow \quad \theta \sim \tilde{\theta}.$$

4.3 The smooth manifold Σ_1^*

We explained in the previous section that studying ϕ allows to better understand how the output $f_\theta(X)$ locally depends on θ. The image of ϕ is of particular interest in this study and is the subject of this section. We define

$$\Sigma_1^* = \{\phi(\theta),\ \theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S\}.$$

The main result of this section, Theorem 76, states that Σ_1^* is a smooth manifold. This result is a key element of the article. Indeed, it allows to consider tangent spaces to Σ_1^*, and by doing so, to linearize the geometric characterization of Theorem 77 illustrated in Figure 4.1. Instead of considering the intersection between a smooth manifold and an affine space as in Theorem 77, this indeed allows to consider the intersection between two affine spaces, which can be characterized with rank computations as in Theorems 78 and 79.

To show this result, we need local injectivity. In this aim, let us consider a fixed θ and analyze the functions $u \mapsto f_{\theta+u}(X)$ and $u \mapsto \phi(\theta + u)$ for u around 0. We can select $N_1 + \cdots + N_{L-1}$ scalar scaling parameters (each in a neighborhood of 1), and use them to "rescale" $\theta + u$ as in Definition 72, leaving $f_{\theta+u}(X)$ and $\phi(\theta + u)$ unchanged ((4.2.7) and Proposition 73). Locally, at first order, this means that there are $N_1 + \cdots + N_{L-1}$ linear combinations of u which leave $f_{\theta+u}(X)$ and $\phi(\theta + u)$ invariant. In order to obtain injectivity with respect to u, locally around 0, we will fix $N_1 + \cdots + N_{L-1}$ components of u as follows.

For each neuron v in a hidden layer, we choose the outward edge $v \to v'$ whose weight $w_{v \to v'}$ has largest (absolute) value (if there are several such edges, we choose one arbitrarily). We denote by $s^\theta_{\max}(v)$ such a neuron v'. For each neuron v in a hidden layer V_l, there is exactly one neuron $s^\theta_{\max}(v)$ in the layer V_{l+1}, and one corresponding edge $v \to s^\theta_{\max}(v)$. See Figure 4.2 for an illustration. We will set

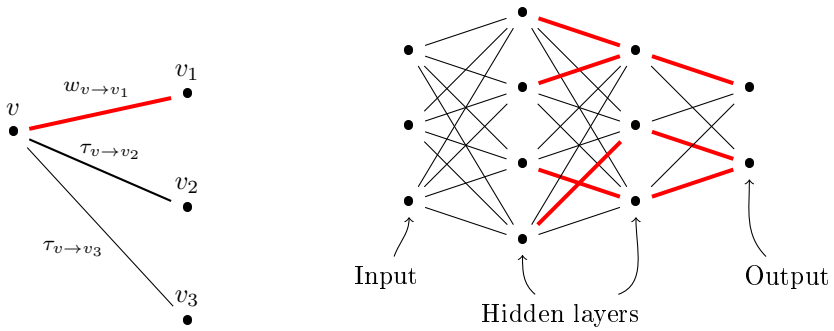

Figure 4.2 – Left: The outward edges of a hidden neuron v and their weights. In this example, $v_1 = s^\theta_{\max}(v)$, so the weight of the edge in red, $v \to v_1$, has its value fixed as $w_{v\to v_1}$. The weights of the remaining edges, $\tau_{v\to v_2}$ and $\tau_{v\to v_3}$, are free to vary. Right: In red, all the edges whose weights are fixed. The remaining edges, in black, constitute the set F_θ.

to 0 the components of u corresponding to all the edges of the form $v \to s^\theta_{\max}(v)$. Intuitively, it will not limit the set of functions $f_{\tilde\theta}$, in the vicinity of f_θ; but will permit to obtain a one-to-one correspondence between u and $f_{\theta+u}$.

More precisely, let us denote by $F_\theta \subset E$ the set of remaining edges, which is formally defined as [1]

$$F_\theta = E \setminus \left(\bigcup_{l=1}^{L-1} \left\{ (v, s^\theta_{\max}(v)), v \in V_l \right\} \right). \tag{4.3.1}$$

The mapping from the space of restricted parameters $\mathbb{R}^{F_\theta} \times \mathbb{R}^B$ to the parameter space $\mathbb{R}^E \times \mathbb{R}^B$ locally around θ is simply given by the following application

$$\begin{array}{rccl} \rho_\theta : & \mathbb{R}^{F_\theta} \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^E \times \mathbb{R}^B \\ & \tau & \longmapsto & \tilde\theta \quad \text{such that} \begin{cases} \forall (v,v') \in F_\theta, \quad \tilde w_{v\to v'} = \tau_{v\to v'} \\ \forall (v,v') \in E \backslash F_\theta, \quad \tilde w_{v\to v'} = w_{v\to v'} \\ \forall v \in B, \quad \tilde b_v = \tau_v. \end{cases} \end{array} \tag{4.3.2}$$

In particular, if we define $\tau_\theta \in \mathbb{R}^{F_\theta} \times \mathbb{R}^B$ by $(\tau_\theta)_{v\to v'} = w_{v\to v'}$ and $(\tau_\theta)_v = b_v$, we have $\rho_\theta(\tau_\theta) = \theta$. The function ρ_θ is affine and injective. We define

$$U_\theta = \rho_\theta^{-1} \left(\left(\mathbb{R}^E \times \mathbb{R}^B \right) \backslash S \right), \tag{4.3.3}$$

which is an open set of $\mathbb{R}^{F_\theta} \times \mathbb{R}^B$. We define, for all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B \right) \backslash S$, the local lifting operator

$$\begin{array}{rccl} \psi^\theta : & U_\theta & \longrightarrow & \mathbb{R}^{\mathcal{P}\times V_L} \\ & \tau & \longmapsto & \phi \circ \rho_\theta(\tau). \end{array} \tag{4.3.4}$$

1. Note, in the definition of F_θ, the index l starting at $l = 1$ and not $l = 0$.

One can show that ψ^θ is C^∞ and that it is a homeomorphism from U_θ onto its image (see the proofs in Appendix 4.A.3), which we denote V_θ and is thus an open subset of Σ_1^* (with the topology induced on Σ_1^* by the standard topology on $\mathbb{R}^{\mathcal{P}\times V_L}$). In particular, since $\rho_\theta(\tau_\theta) = \theta$, we have $\phi(\theta) = \psi^\theta(\tau_\theta) \in V_\theta$. We have the following fundamental result that will allow us to consider and make use the tangent spaces of Σ_1^*.

Theorem 76. *Σ_1^* is a smooth manifold of $\mathbb{R}^{\mathcal{P}\times V_L}$ of dimension*

$$|F_\theta| + |B| = N_0N_1 + N_1N_2 + \cdots + N_{L-1}N_L + N_L,$$

and the family $(V_\theta, (\psi^\theta)^{-1})_{\theta\in(\mathbb{R}^E\times\mathbb{R}^B)\setminus S}$ is an atlas.

Theorem 76 is proven in Appendix 4.A.3. Besides being key in Section 4.4, Theorem 76 (both the smooth manifold nature of Σ_1^* and the explicit atlas $(V_\theta, (\psi^\theta)^{-1})_{\theta\in(\mathbb{R}^E\times\mathbb{R}^B)\setminus S}$) may also be considered of more general independent interest. To our knowledge, such a result has not been established elsewhere in the literature. Notice that, as announced, despite the use of restricted parameters in $\mathbb{R}^{F_\theta}\times\mathbb{R}^B$, we can represent the *whole* tangent space at any point of Σ_1^*. The only consequence of the restriction is the uniqueness of the representation of the elements of tangent spaces.

4.4 Main results: necessary and sufficient conditions for local identifiability

The main results of this paper rely on the decomposition (4.2.4) introduced in Section 4.2. To reformulate (4.2.4), let us introduce the linear operator $A(X,\theta)$, which simply corresponds to the matrix product with $\alpha(X,\theta)$:

$$\begin{array}{rccc} A(X,\theta): & \mathbb{R}^{\mathcal{P}\times V_L} & \longrightarrow & \mathbb{R}^{n\times V_L} \\ & \eta & \longmapsto & \alpha(X,\theta)\eta, \end{array}$$

where $\alpha(X,\theta)\eta$ is the matrix product between $\alpha(X,\theta)\in\mathbb{R}^{n\times\mathcal{P}}$ and $\eta\in\mathbb{R}^{\mathcal{P}\times V_L}$. The operator $A(X,\theta)$ inherits the properties of $\alpha(X,\theta)$, in particular those stated in Proposition 71. Using $A(X,\theta)$, the relation (4.2.4) satisfied by $\tilde\theta$ in the neighborhood of θ becomes

$$f_\theta(X) - f_{\tilde\theta}(X) = A(X,\theta)\cdot\left(\phi(\theta) - \phi(\tilde\theta)\right). \tag{4.4.1}$$

Let us also define the affine space (set-sum of a fixed point and a vector space)

$$N(X,\theta) = \phi(\theta) + \operatorname{Ker} A(X,\theta).$$

If a parameterization $\tilde\theta\in\mathbb{R}^E\times\mathbb{R}^B$ is such that $f_{\tilde\theta}(X) = f_\theta(X)$ and (4.4.1) holds, then $\phi(\theta) - \phi(\tilde\theta)\in\operatorname{Ker} A(X,\theta)$, so by definition $\phi(\tilde\theta)\in N(X,\theta)$. Since for $\tilde\theta$ in the neighborhood of θ, we also have $\phi(\tilde\theta)\in\Sigma_1^*$, we see that local identifiability is closely related to the nature of the intersection between the smooth manifold Σ_1^* and the affine subspace $N(X,\theta)$.

A similar observation is already present in [179, Theorem 5]. In fact, their result can be adapted to the case of a given finite sample X to show, translated in our notations, that a parameterization θ is locally identifiable from X if and only if for any $\tilde{\theta}$ close enough to θ, we have $\phi(\tilde{\theta}) - \phi(\theta) \in \operatorname{Ker} A(X, \theta) \Rightarrow \phi(\tilde{\theta}) = \phi(\theta)$.

Our work in Section 4.3 in which we prove that Σ_1^* is a smooth manifold and in which we construct the charts $(\psi^\theta)^{-1}$, allows us to obtain a variant of this result where we consider a ball around $\phi(\theta)$ instead of θ. We can thus formulate the condition as a purely geometric criterion, corresponding to the nature of the local intersection between a smooth manifold, Σ_1^*, and an affine space, $N(X, \theta)$.

Indeed, let us denote by $B_\infty(\phi(\theta), \epsilon) = \{\eta \in \mathbb{R}^{\mathcal{P} \times V_L}, \|\phi(\theta) - \eta\|_\infty < \epsilon\}$ the ball of center $\phi(\theta)$ and of radius $\epsilon > 0$. We have the following geometric necessary and sufficient condition of local identifiability, which states that local identifiability of θ holds if and only if the intersection between Σ_1^* and $N(X, \theta)$ is locally reduced to the single point $\{\phi(\theta)\}$.

Theorem 77. *For any $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$, the two following statements are equivalent.*

i) θ is locally identifiable from X.

ii) There exists $\epsilon > 0$ such that $B_\infty(\phi(\theta), \epsilon) \cap \Sigma_1^* \cap N(X, \theta) = \{\phi(\theta)\}$.

Theorem 77 is proven in Appendix 4.A.4, and is illustrated in Figure 4.1. This geometric condition is crucial for showing the next two results which give testable conditions of identifiability. Theorems 78 and 79 rely on the rank of $A(X, \theta)$ and of another linear operator $\Gamma(X, \theta)$, which we now define. Since, as we said, the function ψ^θ is C^∞, let us denote by $D\psi^\theta(\tau) : \mathbb{R}^{F_\theta} \times \mathbb{R}^B \to \mathbb{R}^{\mathcal{P} \times V_L}$ its differential at the point τ, for any $\tau \in U_\theta$. We define the linear operator $\Gamma(X, \theta) : \mathbb{R}^{F_\theta} \times \mathbb{R}^B \to \mathbb{R}^{n \times V_L}$ by

$$\Gamma(X, \theta) = A(X, \theta) \circ D\psi^\theta(\tau_\theta). \tag{4.4.2}$$

We denote $R_A = \operatorname{rank}(A(X, \theta))$ and $R_\Gamma = \operatorname{rank}(\Gamma(X, \theta))$. Since $\Gamma(X, \theta)$ is defined on $\mathbb{R}^{F_\theta} \times \mathbb{R}^B$, we have $0 \leq R_\Gamma \leq |F_\theta| + |B|$, and the expression (4.4.2) shows that we also have $0 \leq R_\Gamma \leq R_A$. We can now define the two following conditions.

Condition C_N. Condition C_N is satisfied by (θ, X) iif $R_\Gamma < R_A$ or $R_\Gamma = |F_\theta| + |B|$.

Condition C_S. Condition C_S is satisfied by (θ, X) iif $R_\Gamma = |F_\theta| + |B|$.

The following result states that C_N is necessary for local and therefore global identifiability.

Theorem 78 (Necessary condition of identifiability)**.** *Let $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$. If C_N is not satisfied, then θ is not locally identifiable from X (thus not globally identifiable).*

The following result states that C_S is a sufficient condition of local identifiability.

Theorem 79 (Sufficient condition of local identifiability)**.** *Let $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$. If C_S is satisfied, then θ is locally identifiable from X.*

Both theorems are proven in Appendix 4.A.4. To discuss these two results, let us point out that the output spaces of $\Gamma(X,\theta)$ and $A(X,\theta)$ have the same dimension, equal to nN_L. Each new input adds N_L to this dimension. One can verify that $R_A - R_\Gamma$ is initially 0 and cannot decrease when new inputs are added. If a new input leads to $R_A > R_\Gamma$, it can be discarded to preserve $R_A = R_\Gamma$. Moreover, such an input seems unlikely when $R_A < |F_\theta| + |B|$. If the equality $R_\Gamma = R_A$ is enforced, the condition $R_\Gamma = |F_\theta| + |B|$ is both necessary and sufficient. Finally, to satisfy $R_\Gamma = |F_\theta| + |B|$, the dimensions must satisfy $nN_L \geq |F_\theta| + |B|$. The general belief is that the latter is the condition of identifiability since nN_L is the number of scalar measurements and $|F_\theta| + |B|$ is the number of independent free parameters, see Theorem 76.

4.5 Checking the conditions numerically

The key benefit of the conditions C_N and C_S, compared to the existing literature, is that they can be numerically tested for any fixed finite sample. They need the computation of the rank of two linear operators, namely $\Gamma(X,\theta)$ and $A(X,\theta)$. The operator $\Gamma(X,\theta)$ satisfies the following:

Proposition 80. *Let $X \in \mathbb{R}^{n\times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash (S \cup \Delta_X)$. The function $\tau \mapsto f_{\rho_\theta(\tau)}(X)$, for $\tau \in U_\theta$ is differentiable in a neighborhood of τ_θ, and we denote by $D_\tau f_{\rho_\theta(\tau_\theta)}(X)$ its differential at τ_θ. We have*

$$D_\tau f_{\rho_\theta(\tau_\theta)}(X) = \Gamma(X,\theta). \tag{4.5.1}$$

The proof of Proposition 80 is in Appendix 4.A.5. Since the reparameterization with ρ_θ simply consists in fixing the weights of the edges $v \to s^\theta_{\max}(v)$ to the value $w_{v\to s^\theta_{\max}(v)}$, (4.5.1) shows that the coefficients of $\Gamma(X,\theta)$ can be computed by a classic backpropagation algorithm N_L times for each input x^i, simply omitting the derivatives with respect to the edges of the form $v \to s^\theta_{\max}(v)$. An explicit expression of the coefficients of $\Gamma(X,\theta)$ is given in the Appendix 4.A.5.

To be satisfied, C_S needs the dimensions of $\Gamma(X,\theta)$ to satisfy $nN_L \geq |F_\theta| + |B|$. One then needs to compute the rank R_Γ of $\Gamma(X,\theta)$, which means computing the rank of a $nN_L \times (|F_\theta| + |B|)$ matrix. Existing algorithms allow to do this with a complexity $O(nN_L(|F_\theta| + |B|)^{\omega-1})$ (up to polylog terms), where ω is the matrix multiplication exponent and satisfies $\omega < 2.38$ [47].

When it comes to C_N, one needs in addition to know the rank R_A of $A(X,\theta)$, which, as Proposition 81 states, requires to compute the rank of $\alpha(X,\theta)$.

Proposition 81. *Let $X \in \mathbb{R}^{n\times V_0}$ and $\theta \in \mathbb{R}^E \times \mathbb{R}^B$. We have $R_A = N_L \mathrm{rank}\,(\alpha(X,\theta))$.*

The dimensions of $\alpha(X,\theta)$ are sensibly larger, with $|\mathcal{P}|$ columns and n lines, and typically $|\mathcal{P}| >> n$. However it may have some sparsity properties, as its entries consist in products of activation indicators (with possibly one input $x^i_{v_0}$), any one of them being zero causing many entries to vanish. The question of the efficient computation of R_A still needs to be explored and is left as open for future work.

4.6 Conclusion

This paper is the first to characterize local identifiability for deep ReLU networks for any given finite sample, with testable conditions. The practical use of these conditions deserves follow-up research, and so does an extension of our approach to inverse stability. The role of ReLU is crucial in our approach, especially for the necessary condition of local identifiability and with the linear representation (Proposition 70). In the end, from Theorem 79 and Proposition 80, the sufficient condition for local indentifiability is expressed from the Jacobian matrix of the neural network function with respect to its parameters. Extending this to other activation functions than ReLU is an interesting perspective.

Acknowledgements

The authors would like to thank Pierre Stock and Rémi Gribonval for the fruitful discussions around this work, notably regarding the construction of ϕ and its link to the question of local identifiability.

This work has benefited from the AI Interdisciplinary Institute ANITI. ANITI is funded by the French "Investing for the Future – PIA3" program under the Grant agreement n°ANR-19-PI3A-0004.

The authors gratefully acknowledge the support of the DEEL project. [2]

2. https://www.deel.ai/

4.A Appendix

4.A.1 Notations

In this section, we define notations, many of which are standard, that are useful in the proofs.

We denote by $\mathbb{N}$ the set of all natural numbers, including 0, and by $\mathbb{N}^*$ the set $\mathbb{N}$ without 0. We denote by $\mathbb{Z}$ the set of all integers. For any $a, b \in \mathbb{Z}$, we denote by $[\![a, b]\!]$ the set of all integers $k \in \mathbb{Z}$ satisfying $a \leq k \leq b$. For any finite set A, we denote by $|A|$ the cardinal of A.

For $n, N \in \mathbb{N}^*$, we denote by $\mathbb{R}^N$ the N-dimensional real vector space and by $\mathbb{R}^{n\times N}$ the vector space of real matrices with n lines and N columns. For a vector $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$, we use the norm $\|x\|_\infty = \max_{i\in[\![1,N]\!]} |x_i|$. For $x \in \mathbb{R}^N$ and $r > 0$, we denote $B_\infty(x, r) = \{y \in \mathbb{R}^N, \|y - x\|_\infty < r\}$.

For any vector $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$, we define

$$\operatorname{sign}(x) = (\operatorname{sign}(x_1), \ldots, \operatorname{sign}(x_N))^T \in \{-1, 0, 1\}^N$$

as the vector whose i^{th} component is equal to

$$\operatorname{sign}(x_i) = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i = 0 \\ -1 & \text{if } x_i < 0. \end{cases}$$

For any matrix $M \in \mathbb{R}^{n\times N}$, for all $i \in [\![1, n]\!]$, we denote by $M_{i,:}$ the i^{th} line of M. The vector $M_{i,:}$ is a line vector whose j^{th} component is $M_{i,j}$. Similarly, for $j \in [\![1, N]\!]$, we denote by $M_{:,j}$ the j^{th} column of M, which is the column vector whose i^{th} component is $M_{i,j}$. For any matrix $M \in \mathbb{R}^{n\times N}$, we denote by $M^T \in \mathbb{R}^{N\times n}$ the transpose matrix of M.

We denote by Id_N the $N\times N$ identity matrix and by $\mathbf{1}_N$ the vector $(1, \ldots, 1)^T \in \mathbb{R}^N$. If $\lambda \in \mathbb{R}^N$ is a vector of size N, for some $N \in \mathbb{N}^*$, we denote by $\operatorname{Diag}(\lambda)$ the $N \times N$ matrix defined by:

$$\operatorname{Diag}(\lambda)_{i,j} = \begin{cases} \lambda_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

If X and Y are two sets and $h : X \to Y$ is a function, for a subset $A \subset Y$, we denote by $h^{-1}(A)$ the preimage of A under f, that is

$$h^{-1}(A) = \{x \in X, h(x) \in A\}.$$

Note that this does not require the function h to be injective.

For any $n, N \in \mathbb{N}^*$ and any differentiable function $f : \mathbb{R}^n \to \mathbb{R}^N$, for all $x \in \mathbb{R}^n$, we denote by $Df(x)$ its differential at the point x, i.e. the linear application $Df(x) : \mathbb{R}^n \to \mathbb{R}^N$ satisfying, for all $h \in \mathbb{R}^n$,

$$f(x + h) = f(x) + Df(x) \cdot h + o(h).$$

If we denote by x_j and h_j the components of x and h, for $j \in [\![1, n]\!]$, we have

$$Df(x) \cdot h = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x) h_j,$$

where for all j, $\frac{\partial f}{\partial x_j}(x) \in \mathbb{R}^N$. If $f : \mathbb{R}^n \to \mathbb{R}^N$ is a linear application, we denote by $\operatorname{Ker} f$ the set $\{x \in \mathbb{R}^n, f(x) = 0\}$, which is a linear subset of $\mathbb{R}^n$.

4.A.2 The lifting operator ϕ

Let us introduce the notion of 'path', extending the definition in Section 4.2.2. A path is a sequence of neurons $(v_k, v_{k+1}, \dots, v_l) \in V_k \times V_{k+1} \times \cdots \times V_l$, for integers k, l satisfying $0 \leq k \leq l \leq L$. In particular, for all $l \in [\![0, L-1]\!]$, the set $\mathcal{P}_l$ defined in Section 4.2.2 contains all the paths starting from layer l and ending in layer $L-1$. We recall

$$\mathcal{P} = \left(\bigcup_{l=0}^{L-1} \mathcal{P}_l \right) \cup \{\beta\}.$$

If $k, l, m \in \mathbb{N}$ are three integers satisfying $0 \leq k < l \leq m \leq L$, and $p = (v_k, \dots, v_{l-1}) \in V_k \times \cdots \times V_{l-1}$ and $p' = (v_l, \dots, v_m) \in V_l \times \cdots \times V_m$ are two paths such that p ends in the layer preceding the starting layer of p', we define the union of the paths by

$$p \cup p' = (v_k, \dots, v_{l-1}, v_l, \dots v_m) \in V_k \times \cdots \times V_m.$$

Before proving Proposition 70, let us compare briefly our construction to [179]. The lifting operator ϕ introduced in Section 4.2.2 is similar to the operator $\mathbf{\Phi}$ in [179], except that $\mathbf{\Phi}$ does not take a matrix form. The operator $\alpha(x, \theta)$ introduced in Section 4.2.2 corresponds partly to the object $\overline{\boldsymbol{\alpha}}(\theta, x)$ in [179]. One of the differences is that $\overline{\boldsymbol{\alpha}}(\theta, x)$ does not include any product with x_{v_0} in its entries, as does $\alpha(x, \theta)$. Finally, a similar statement to Proposition 70 and a similar proof can be found in [179]. However, one of the present contributions is to simplify the construction.

Let us now prove Proposition 70, which we restate here.

Proposition 82. *For all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and all $x \in \mathbb{R}^{V_0}$,*

$$f_\theta(x)^T = \alpha(x, \theta) \phi(\theta).$$

Proof. Let us prove first the following expression, for all $v_L \in V_L$:

$$f_\theta(x)_{v_L} = \left(\sum_{\substack{v_0 \in V_0 \\ \vdots \\ v_{L-1} \in V_{L-1}}} x_{v_0} w_{v_0 \to v_1} \prod_{l=1}^{L-1} a_{v_l}(x, \theta) w_{v_l \to v_{l+1}} \right) + \left(\sum_{l=1}^{L-1} \sum_{\substack{v_l \in V_l \\ \vdots \\ v_{L-1} \in V_{L-1}}} b_{v_l} \prod_{l'=l}^{L-1} a_{v_{l'}}(x, \theta) w_{v_{l'} \to v_{l'+1}} \right) + b_{v_L}. \tag{4.A.1}$$

We prove this by induction on the number L of layers of the network.

Initialization ($L = 2$). Let $v_2 \in V_2$.

$$
\begin{aligned}
f_\theta(x)_{v_2} &= (W_2)_{v_2,:}\, \sigma\left(W_1 x + b_1\right) + b_{v_2} \\
&= \left(\sum_{v_1 \in V_1} w_{v_1 \to v_2} \left[\sigma\left(W_1 x + b_1\right)\right]_{v_1} \right) + b_{v_2} \\
&= \left(\sum_{v_1 \in V_1} w_{v_1 \to v_2} \sigma\left((W_1)_{v_1,:}\, x + b_{v_1}\right) \right) + b_{v_2} \\
&= \left(\sum_{v_1 \in V_1} w_{v_1 \to v_2} a_{v_1}(x,\theta) \left(\sum_{v_0 \in V_0} w_{v_0 \to v_1} x_{v_0} + b_{v_1} \right) \right) + b_{v_2} \\
&= \left(\sum_{\substack{v_0 \in V_0 \\ v_1 \in V_1}} w_{v_1 \to v_2} a_{v_1}(x,\theta) w_{v_0 \to v_1} x_{v_0} \right) + \left(\sum_{v_1 \in V_1} w_{v_1 \to v_2} a_{v_1}(x,\theta) b_{v_1} \right) + b_{v_2} \\
&= \left(\sum_{\substack{v_0 \in V_0 \\ v_1 \in V_1}} x_{v_0} w_{v_0 \to v_1} a_{v_1}(x,\theta) w_{v_1 \to v_2} \right) + \left(\sum_{v_1 \in V_1} b_{v_1} a_{v_1}(x,\theta) w_{v_1 \to v_2} \right) + b_{v_2}
\end{aligned}
$$

which proves (4.A.1), when $L = 2$.

Now let $L \geq 3$ and suppose (4.A.1) holds for all ReLU networks with $L - 1$ layers. Let us consider a network with L layers.

Let us denote by $g_\theta(x)$ the output of the $L - 1$ first layers of the network pre-activation (before applying the ReLUs of the layer $L - 1$). The function g_θ is that of a ReLU network with $L - 1$ layers, and we have

$$f_\theta(x) = W_L \sigma(g_\theta(x)) + b_L.$$

Let $v_L \in V_L$. We thus have

$$f_\theta(x)_{v_L} = \sum_{v_{L-1} \in V_{L-1}} w_{v_{L-1} \to v_L} \sigma(g_\theta(x)_{v_{L-1}}) + b_{v_L}. \tag{4.A.2}$$

By the induction hypothesis, for all $v_{L-1} \in V_{L-1}$, $g_\theta(x)_{v_{L-1}}$ can be expressed with (4.A.1). Considering that $\sigma(g_\theta(x)_{v_{L-1}}) = a_{v_{L-1}}(x,\theta) g_\theta(x)_{v_{L-1}}$ and replacing

$g_\theta(x)_{v_{L-1}}$ by its expression using (4.A.1), (4.A.2) becomes

$$
\begin{aligned}
f_\theta(x)_{v_L} = \sum_{v_{L-1}\in V_{L-1}} w_{v_{L-1}\to v_L} a_{v_{L-1}}(x,\theta)\Bigg[\Bigg(\sum_{\substack{v_0\in V_0\\ \vdots \\ v_{L-2}\in V_{L-2}}} x_{v_0} w_{v_0\to v_1} \prod_{l=1}^{L-2} a_{v_l}(x,\theta) w_{v_l\to v_{l+1}}\Bigg) \\
+ \Bigg(\sum_{l=1}^{L-2} \sum_{\substack{v_l\in V_l\\ \vdots \\ v_{L-2}\in V_{L-2}}} b_{v_l} \prod_{l'=l}^{L-2} a_{v_{l'}}(x,\theta) w_{v_{l'}\to v_{l'+1}}\Bigg) + b_{v_{L-1}}\Bigg] + b_{v_L} \\
= \Bigg(\sum_{\substack{v_0\in V_0\\ \vdots \\ v_{L-1}\in V_{L-1}}} w_{v_{L-1}\to v_L} a_{v_{L-1}}(x,\theta) x_{v_0} w_{v_0\to v_1} \prod_{l=1}^{L-2} a_{v_l}(x,\theta) w_{v_l\to v_{l+1}}\Bigg) \\
+ \Bigg(\sum_{l=1}^{L-2} \sum_{\substack{v_l\in V_l\\ \vdots \\ v_{L-1}\in V_{L-1}}} w_{v_{L-1}\to v_L} a_{v_{L-1}}(x,\theta) b_{v_l} \prod_{l'=l}^{L-2} a_{v_{l'}}(x,\theta) w_{v_{l'}\to v_{l'+1}}\Bigg) \\
+ \Bigg(\sum_{v_{L-1}\in V_{L-1}} w_{v_{L-1}\to v_L} a_{v_{L-1}}(x,\theta) b_{v_{L-1}}\Bigg) + b_{v_L} \\
= \Bigg(\sum_{\substack{v_0\in V_0\\ \vdots \\ v_{L-1}\in V_{L-1}}} x_{v_0} w_{v_0\to v_1} \prod_{l=1}^{L-1} a_{v_l}(x,\theta) w_{v_l\to v_{l+1}}\Bigg) \\
+ \Bigg(\sum_{l=1}^{L-1} \sum_{\substack{v_l\in V_l\\ \vdots \\ v_{L-1}\in V_{L-1}}} b_{v_l} \prod_{l'=l}^{L-1} a_{v_{l'}}(x,\theta) w_{v_{l'}\to v_{l'+1}}\Bigg) + b_{v_L},
\end{aligned}
$$

which proves (4.A.1) holds for ReLU networks with L layers. This ends the induction, and we conclude that (4.A.1) holds for all ReLU networks.

We can now use this expression to prove Proposition 82. The first sum in (4.A.1) is taken over all the paths $p = (v_0, \dots, v_{L-1}) \in \mathcal{P}_0$, and each summand can be written as

$$
\begin{aligned}
x_{v_0} w_{v_0\to v_1} \prod_{l=1}^{L-1} a_{v_l}(x,\theta) w_{v_l\to v_{l+1}} &= \Bigg(x_{v_0} \prod_{l=1}^{L-1} a_{v_l}(x,\theta)\Bigg)\Bigg(\prod_{l=0}^{L-1} w_{v_l\to v_{l+1}}\Bigg) \\
&= \alpha_p(x,\theta)\phi_{p,v_L}(\theta).
\end{aligned}
$$

For all $l \in [\![1, L-1]\!]$, the inner sum of the double sum in (4.A.1) is taken over all

the paths $p = (v_l, \dots, v_{L-1}) \in \mathcal{P}_l$, and each summand can be written as

$$b_{v_l} \prod_{l'=l}^{L-1} a_{v_{l'}}(x,\theta) w_{v_{l'} \to v_{l'+1}} = \left(\prod_{l'=l}^{L-1} a_{v_{l'}}(x,\theta) \right) \left(b_{v_l} \prod_{l'=l}^{L-1} w_{v_{l'} \to v_{l'+1}} \right) = \alpha_p(x,\theta) \phi_{p,v_L}(\theta).$$

And finally, we can also write

$$b_{v_L} = \alpha_\beta(x,\theta) \phi_{\beta,v_L}(\theta).$$

Joining all these sums and denoting $\phi_{:,v_L}(\theta) = (\phi_{p,v_L}(\theta))_{p \in \mathcal{P}} \in \mathbb{R}^{\mathcal{P}}$, we have

$$f_\theta(x)_{v_L} = \sum_{p \in \mathcal{P}} \alpha_p(x,\theta) \phi_{p,v_L}(\theta) = \alpha(x,\theta) \phi_{:,v_L}(\theta),$$

so in other words,

$$f_\theta(x)^T = \alpha(x,\theta) \phi(\theta).$$

□

We restate here and prove Proposition 71.

Proposition 83. *For all $n \in \mathbb{N}^*$, for all $X \in \mathbb{R}^{n \times V_0}$, the mapping*

$$\begin{array}{rccc} \alpha_X : & \mathbb{R}^E \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^{n \times \mathcal{P}} \\ & \theta & \longmapsto & \alpha(X,\theta) \end{array}$$

appearing in (4.2.3) is piecewise-constant, with a finite number of pieces. Furthermore, the boundary of each piece has Lebesgue measure zero. We call Δ_X the union of all the boundaries. The set Δ_X is closed and has Lebesgue measure zero.

Proof. Let us first notice that for any $i \in [\![1, n]\!]$, for any $l \in [\![1, L-1]\!]$,

$$\left(a_v(x^i,\theta)\right)_{v \in V_1 \cup \dots \cup V_{l-1}} \in \{0,1\}^{V_1 \cup \dots \cup V_{l-1}}$$

takes at most $2^{N_1 + \dots + N_{l-1}}$ distinct values, so the mapping $\theta \mapsto \left(a_v(x^i,\theta)\right)_{v \in V_1 \cup \dots \cup V_{l-1}}$ is piecewise constant, with a finite number of pieces.

Let $i \in [\![1, n]\!]$. Let $l \in [\![1, L-1]\!]$ and $v \in V_l$. Recall the definition of f_{l-1}, as given in Section 4.2.1. The function $\theta \to a_v(x^i,\theta)$ takes only two values, 1 or 0, and its values are determined by the sign of

$$\sum_{v' \in V_{l-1}} w_{v' \to v} f_{l-1}(x^i)_{v'} + b_v. \tag{4.A.3}$$

For all $v' \in V_{l-1}$, the value of $f_{l-1}(x^i)_{v'}$ depends on θ. On a piece $P \subset \mathbb{R}^E \times \mathbb{R}^B$ such that $\left(a_{v''}(x^i,\theta)\right)_{v'' \in V_1 \cup \dots \cup V_{l-1}}$ is constant, this dependence is polynomial. Thus, on P, the value of (4.A.3) is a polynomial function of θ, and since the coefficient applied to b_v is equal to 1, the corresponding polynomial is non constant. Since the values of $a_v(x^i,\theta)$ are determined by the sign of (4.A.3), inside P, the boundary between $\{\theta \in \mathbb{R}^E \times \mathbb{R}^B, a_v(x^i,\theta) = 0\}$ and $\{\theta \in \mathbb{R}^E \times \mathbb{R}^B, a_v(x^i,\theta) = 1\}$ is included

in the set of θ for which (4.A.3) equals 0. This piece of boundary is thus contained in a level set of a non constant polynomial, whose Lebesgue measure is zero.

Since there is a finite number of pieces P, the Lebesgue measure of the boundary between $\{\theta \in \mathbb{R}^E \times \mathbb{R}^B, a_v(x^i, \theta) = 0\}$ and $\{\theta \in \mathbb{R}^E \times \mathbb{R}^B, a_v(x^i, \theta) = 1\}$, which is contained in the union of the boundaries on all the pieces P, is thus equal to 0.

Since this is true for all $l \in [\![1, L-1]\!]$ and all $v \in V_l$, the boundary of a piece over which $\big(a_v(x^i, \theta)\big)_{v \in V_1 \cup \cdots \cup V_{L-1}}$ is constant also has Lebesgue measure zero.

Now since, for all x^i, the value of $\alpha(x^i, \theta)$ only depends on $\big(a_v(x^i, \theta)\big)_{v \in V_1 \cup \cdots \cup V_{L-1}}$ and since $\alpha_X(\theta)$ is a matrix whose lines are the vectors $\alpha(x^i, \theta)$, we can conclude that $\begin{array}{rccc} \alpha_X : & \mathbb{R}^E \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^{n \times \mathcal{P}} \\ & \theta & \longmapsto & \alpha(X, \theta) \end{array}$ is piecewise-constant, with a finite number of pieces, and that the boundary of each piece has Lebesgue measure zero.

A boundary is, by definition, closed. Finally, a finite union of closed sets with Lebesgue measure 0, as Δ_X is, is closed and has Lebesgue measure 0. □

For convenience, we introduce the two following notations. Let $l \in [\![0, L]\!]$. For any $l' \in [\![0, l]\!]$ and any path $p_i = (v_{l'}, \ldots, v_l) \in V_{l'} \times \cdots \times V_l$, we denote

$$\theta_{p_i} = \begin{cases} \prod_{k=0}^{l-1} w_{v_k \to v_{k+1}} & \text{if } l' = 0 \\ b_{l'} \prod_{k=l'}^{l-1} w_{v_k \to v_{k+1}} & \text{if } l' \geq 1, \end{cases} \tag{4.A.4}$$

where as a classic convention, an empty product is equal to 1. In particular, if $l = 0$, for any $p_i = (v_0) \in V_0$, we have $\theta_{p_i} = 1$. For any path $p_o = (v_l, \ldots, v_L) \in V_l \times \cdots \times V_L$, we denote

$$\theta_{p_o} = \prod_{k=l}^{L-1} w_{v_k \to v_{k+1}}, \tag{4.A.5}$$

with again the convention that an empty product is equal to 1, so if $l = L$, $\theta_{p_o} = 1$.

Some attention must be paid to the fact that for any $l' \in [\![1, L]\!]$, if we take p_i in the case $l = L$ and p_o in the case $l = l'$, it is possible to have

$$p_i = (v_{l'}, \ldots, v_L) = p_o,$$

but in that case we DO NOT have $\theta_{p_i} = \theta_{p_o}$, since $\theta_{p_i} = b_{l'} \prod_{k=l'}^{L-1} w_{v_k \to v_{k+1}}$ and $\theta_{p_o} = \prod_{k=l'}^{L-1} w_{v_k \to v_{k+1}}$. We will always denote the paths p_i and p_o with an i (as in 'input') or an o (as in 'output') to clarify which definition is used.

When considering another parameterization $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, we denote by $\tilde{\theta}_{p_i}$ and $\tilde{\theta}_{p_o}$ the corresponding objects.

We establish different characterizations of the set S defined in Section 4.2.3 that will be useful in the proofs. As mentioned in Section 4.2.3, the subset of parameters $\big(\mathbb{R}^E \times \mathbb{R}^B\big) \setminus S$ is close to the notion of 'admissible' parameter in [179], but is slightly larger since the condition $w_{\bullet \to v} \neq 0$ is replaced by $(w_{\bullet \to v}, b_v) \neq (0, 0)$, for each hidden neuron v.

Proposition 84. *Let $\theta \in \mathbb{R}^E \times \mathbb{R}^B$. The following statements are equivalent.*

i) $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$.

ii) For all $l \in [\![1, L-1]\!]$ *and all* $v_l \in V_l$, *there exist* $l' \in [\![0, l]\!]$, *a path* $p_i = (v_{l'}, \dots, v_l) \in V_{l'} \times \cdots \times V_l$ *and a path* $p_o = (v_l, \dots, v_L) \in V_l \times \cdots \times V_L$ *such that*

$$\theta_{p_i} \neq 0 \quad \textit{and} \quad \theta_{p_o} \neq 0.$$

iii) For all $l \in [\![1, L-1]\!]$ *and all* $v_l \in V_l$, *there exist* $l' \in [\![0, l]\!]$, *a path* $p = (v_{l'}, \dots, v_l, \dots, v_{L-1}) \in \mathcal{P}_{l'}$ *and* $v_L \in V_L$ *such that*

$$\phi_{p, v_L}(\theta) \neq 0.$$

Proof. Let us show successively that *i)* $\Rightarrow$ *ii)*, *ii)* $\Rightarrow$ *iii)* and *iii)* $\Rightarrow$ *i)*.

i) $\rightarrow$ *ii)* Let $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$. Let us show *ii)* holds.

Let $l \in [\![1, L]\!]$ and $v_l \in V_l$. To form a path p_i satisfying the condition, we follow the procedure:

$p_i \leftarrow (v_l)$
$k \leftarrow l$
while $k \geq 1$ and $b_k = 0$ **do**
  $\exists v_{k-1} \in V_{k-1}, w_{v_{k-1} \to v_k} \neq 0$
  $p_i \leftarrow (v_{k-1}, p_i)$
  $k \leftarrow k - 1$
end while
$l' \leftarrow k$

The existence of v_{k-1} in the loop is guaranteed by the fact that $\theta \notin S$ and $b_k = 0$ in the condition of the while loop. In the end, we obtain a path $p_i = (v_{l'}, \dots, v_l)$ with either $l' > 0$ and $b_{l'} \neq 0$, or $l' = 0$. In both cases, we have by construction

$$\theta_{p_i} \neq 0.$$

We do similarly the other way to form a path $p_o = (v_l, \dots, v_L)$. We follow the procedure:

$p_o \leftarrow (v_l)$
$k \leftarrow l$
while $k \leq L - 1$ **do**
  $\exists v_{k+1} \in V_{k+1}, w_{v_k \to v_{k+1}} \neq 0$
  $p_o \leftarrow (p_o, v_{k+1})$
  $k \leftarrow k + 1$
end while

The existence of v_{k+1} in the loop is guaranteed by the fact that $\theta \notin S$. In the end, we obtain a path $p_o = (v_l, \dots, v_L)$ satisfying by construction

$$\theta_{p_o} \neq 0.$$

ii) $\rightarrow$ *iii)* Let $l \in [\![1, L-1]\!]$ and $v_l \in V_l$. There exist $l' \in [\![0, l]\!]$, a path $p_i = (v_{l'}, \dots, v_l) \in V_{l'} \times \cdots \times V_l$ and a path $p_o = (v_l, \dots, v_L) \in V_l \times \cdots \times V_L$ such that

$$\theta_{p_i} \neq 0 \quad \text{and} \quad \theta_{p_o} \neq 0.$$

Denoting $p = (v_{l'}, \dots, v_l, \dots, v_{L-1})$, we have

$$\phi_{p,v_L}(\theta) = \theta_{p_i}\theta_{p_o} \neq 0.$$

iii) $\to$ *i)* Let us show the contrapositive: let $\theta \in S$, and let us show the statement *iii)* is not true. Indeed, if $\theta \in S$, there exist $l \in [\![1, L-1]\!]$ and $v_l \in V_l$ such that $(w_{\bullet \to v_l}, b_{v_l}) = (0,0)$ or $w_{v_l \to \bullet} = 0$. Consider a path $p = (v_{l'}, \dots, v_l, \dots, v_{L-1})$ and $v_L \in V_L$. We have

$$\phi_{p,v_L}(\theta) = \begin{cases} b_{v_{l'}} w_{v_{l'} \to v_{l'+1}} \dots w_{v_{l-1} \to v_l} w_{v_l \to v_{l+1}} \dots w_{v_{L-1} \to v_L} & \text{if } l' \geq 1 \\ w_{v_0 \to v_1} \dots w_{v_{l-1} \to v_l} w_{v_l \to v_{l+1}} \dots w_{v_{L-1} \to v_L} & \text{if } l' = 0. \end{cases}$$

If $(w_{\bullet \to v_l}, b_{v_l}) = (0,0)$, either $l' = l$ and $b_{v_{l'}} = 0$ so $\phi_{p,v_L}(\theta) = 0$, or $l' < l$ and since $w_{v_{l-1} \to v_l} = 0$, we have $\phi_{p,v_L}(\theta) = 0$.

If $w_{v_l \to \bullet} = 0$, $w_{v_l \to v_{l+1}} = 0$ so $\phi_{p,v_L}(\theta) = 0$. Thus *iii)* is not satisfied. □

We restate and prove Proposition 73.

Proposition 85. *For all $\theta, \tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, we have*

$$\theta \overset{R}{\sim} \tilde{\theta} \quad \Longrightarrow \quad \phi(\theta) = \phi(\tilde{\theta}),$$

and thus in particular

$$\theta \sim \tilde{\theta} \quad \Longrightarrow \quad \phi(\theta) = \phi(\tilde{\theta}).$$

Proof. Let $\theta, \tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\theta \overset{R}{\sim} \tilde{\theta}$. There exists a family $(\lambda^0, \dots, \lambda^L) \in (\mathbb{R}^*)^{V_0} \times \dots \times (\mathbb{R}^*)^{V_L}$, with $\lambda^0 = \mathbf{1}_{V_0}$ and $\lambda^L = \mathbf{1}_{V_L}$, such that for all $l \in [\![1, L]\!]$, for all $(v_{l-1}, v_l) \in V_{l-1} \times V_l$, (4.2.6) holds. We consider first a path $p = (v_0, \dots, v_{L-1}) \in \mathcal{P}_0$ and $v_L \in V_L$. Using (4.2.6) and the fact that $\lambda^0_{v_0} = \lambda^L_{v_L} = 1$, we have

$$\phi_{p,v_L}(\theta) = \prod_{l=1}^{L} w_{v_{l-1} \to v_l} = \prod_{l=1}^{L} \frac{\lambda^l_{v_l}}{\lambda^{l-1}_{v_{l-1}}} \tilde{w}_{v_{l-1} \to v_l} = \frac{\lambda^L_{v_L}}{\lambda^0_{v_0}} \prod_{l=1}^{L} \tilde{w}_{v_{l-1} \to v_l} = \phi_{p,v_L}(\tilde{\theta}).$$

Similarly, for $l \in [\![1, L-1]\!]$ and a path $p = (v_l, \dots, v_{L-1}) \in \mathcal{P}_l$, and for all $v_L \in V_L$, we have, using (4.2.6) and the fact that $\lambda^L_{v_L} = 1$,

$$\begin{aligned} \phi_{p,v_L}(\theta) &= b_{v_l} \prod_{l'=l+1}^{L} w_{v_{l'-1} \to v_{l'}} = \lambda^l_{v_l} \tilde{b}_{v_l} \prod_{l'=l+1}^{L} \frac{\lambda^{l'}_{v_{l'}}}{\lambda^{l'-1}_{v_{l'-1}}} \tilde{w}_{v_{l'-1} \to v_{l'}} = \lambda^L_{v_L} \tilde{b}_{v_l} \prod_{l'=l+1}^{L} \tilde{w}_{v_{l'-1} \to v_{l'}} \\ &= \phi_{p,v_L}(\tilde{\theta}). \end{aligned}$$

Finally, for $p = \beta$ and $v_L \in V_L$, we have

$$\phi_{p,v_L}(\theta) = b_{v_L} = \lambda^L_{v_L} \tilde{b}_{v_L} = \tilde{b}_{v_L} = \phi_{p,v_L}(\tilde{\theta}).$$

This shows $\phi(\theta) = \phi(\tilde{\theta})$.

For the second implication, we simply use the fact that if $\theta \sim \tilde{\theta}$, in particular, $\theta \overset{R}{\sim} \tilde{\theta}$. □

Corollary 86. *The set* $\left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$ *is stable by rescaling equivalence: if* $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$*, and* $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$ *satisfies* $\theta \overset{R}{\sim} \tilde{\theta}$*, then* $\tilde{\theta} \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$*.*

Proof. Let $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$ and $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\theta \overset{R}{\sim} \tilde{\theta}$. Proposition 85 shows that $\phi(\tilde{\theta}) = \phi(\theta)$.

Let $l \in [\![1, L]\!]$ and $v \in V_l$. Since $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$, according to Proposition 84 there exists $l' \in [\![0, l]\!]$, a path $p = (v_{l'}, \dots, v_l, \dots, v_{L-1})$ and $v_L \in V_L$ such that $\phi_{p,v_L}(\theta) \neq 0$. We have

$$\phi_{p,v_L}(\tilde{\theta}) = \phi_{p,v_L}(\theta) \neq 0,$$

and since this is true for any $l \in [\![1, L]\!]$ and $v \in V_l$, Proposition 84 shows that $\tilde{\theta} \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$. □

We restate and prove Proposition 74.

Proposition 87. *For all* $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$*, for all* $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$*,*

$$\phi(\theta) = \phi(\tilde{\theta}) \quad \Longrightarrow \quad \theta \overset{R}{\sim} \tilde{\theta}.$$

Proof. Let us choose $(\lambda^0, \dots, \lambda^L) \in (\mathbb{R}^*)^{V_0} \times \cdots \times (\mathbb{R}^*)^{V_L}$ as follows. For all $l \in [\![1, L-1]\!]$ and all $v_l \in V_l$, since $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$, Proposition 84 shows that there exists a path $p_o(v_l) = (v_l, \dots, v_L) \in V_l \times \cdots \times V_L$ such that $\theta_{p_o(v_l)} \neq 0$. Let us define $\lambda^0 = \mathbf{1}_{V_0}$, $\lambda^L = \mathbf{1}_{V_L}$ and for all $l \in [\![1, L-1]\!]$,

$$\lambda^l_{v_l} = \frac{\tilde{\theta}_{p_o(v_l)}}{\theta_{p_o(v_l)}}.$$

The value of $\lambda^l_{v_l}$ a priori depends on the choice of the path $p_o(v_l)$, but the first of the two following facts, that we are going to prove, shows it only depends on v_l, since in (4.A.6), p_i does not depend on $p_o(v_l)$.

— For all $l \in [\![0, L]\!]$, for all $v_l \in V_l$, for any $l' \in [\![0, l]\!]$ and any $p_i = (v_{l'}, \dots, v_l) \in V_{l'} \times \cdots \times V_l$,

$$\theta_{p_i} = \lambda^l_{v_l} \tilde{\theta}_{p_i}. \tag{4.A.6}$$

— For all $l \in [\![0, L]\!]$, for all $v_l \in V_l$, $\lambda^l_{v_l} \neq 0$.

Indeed, let $l \in [\![0, L]\!]$ and let us consider $l' \in [\![0, l]\!]$ and a path $p_i = (v_{l'}, \dots, v_l) \in V_{l'} \times \cdots \times V_l$. Let $v_{l+1}, \dots, v_L \in V_{l+1} \times \cdots \times V_L$ such that $p_o(v_l) = (v_l, v_{l+1}, \dots, v_L)$. Let $p = (v_{l'}, \dots, v_l, \dots, v_{L-1}) \in \mathcal{P}_{l'}$ so that $p_i \cup p_o(v_l) = p \cup (v_L)$. We have by hypothesis

$$\theta_{p_i}\theta_{p_o(v_l)} = \phi_{p,v_L}(\theta) = \phi_{p,v_L}(\tilde{\theta}) = \tilde{\theta}_{p_i}\tilde{\theta}_{p_o(v_l)},$$

thus

$$\theta_{p_i} = \frac{\tilde{\theta}_{p_o(v_l)}}{\theta_{p_o(v_l)}} \tilde{\theta}_{p_i} = \lambda^l_{v_l} \tilde{\theta}_{p_i},$$

which proves the first point. To prove the second point, we simply use Proposition 84 to consider a path p_i such that $\theta_{p_i} \neq 0$, and (4.A.6) shows that $\lambda^l_{v_l} \neq 0$.

Let us now prove the rescaling equivalence. Let $l \in [\![1, L]\!]$, and let $(v_{l-1}, v_l) \in V_{l-1} \times V_l$. Let us consider, thanks to Proposition 84, $l' \in [\![0, l-1]\!]$ and a path $p_i = (v_{l'}, \dots, v_{l-1}) \in V_{l'} \times \cdots \times V_{l-1}$ such that $\theta_{p_i} \neq 0$. The relation (4.A.6) shows we also have $\tilde{\theta}_{p_i} \neq 0$. Let $p_i' = p_i \cup (v_l)$. Using (4.A.6) with $\theta_{p_i'}$ we have

$$\theta_{p_i} w_{v_{l-1} \to v_l} = \theta_{p_i'} = \lambda_{v_l}^l \tilde{\theta}_{p_i'} = \lambda_{v_l}^l \tilde{\theta}_{p_i} \tilde{w}_{v_{l-1} \to v_l}.$$

At the same time, using (4.A.6) with θ_{p_i} we have,

$$\theta_{p_i} w_{v_{l-1} \to v_l} = \lambda_{v_{l-1}}^{l-1} \tilde{\theta}_{p_i} w_{v_{l-1} \to v_l},$$

so combining both equalities, we have

$$\lambda_{v_l}^l \tilde{\theta}_{p_i} \tilde{w}_{v_{l-1} \to v_l} = \lambda_{v_{l-1}}^{l-1} \tilde{\theta}_{p_i} w_{v_{l-1} \to v_l}.$$

Using the fact that $\tilde{\theta}_{p_i} \neq 0$ and $\lambda_{v_{l-1}}^{l-1} \neq 0$, we finally obtain, for all $l \in [\![1, L]\!]$ and all $(v_{l-1}, v_l) \in V_{l-1} \times V_l$:

$$w_{v_{l-1} \to v_l} = \frac{\lambda_{v_l}^l}{\lambda_{v_{l-1}}^{l-1}} \tilde{w}_{v_{l-1} \to v_l}.$$

For all $l \in [\![1, L]\!]$ and all $v_l \in V_l$, using (4.A.6) with $p_i = (v_l)$, we obtain

$$b_{v_l} = \lambda_{v_l}^l \tilde{b}_{v_l}.$$

This shows that (4.2.6) is satisfied for all $(v_{l-1}, v_l) \in V_{l-1} \times V_l$, and thus $\theta \overset{R}{\sim} \tilde{\theta}$. □

The following proposition is useful in the proof of Theorem 95 and allows to improve identifiability modulo rescaling into identifiability modulo positive rescaling.

Proposition 88. *For all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$, there exists $\epsilon > 0$ such that for all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$,*

$$\|\theta - \tilde{\theta}\|_\infty < \epsilon \text{ and } \theta \overset{R}{\sim} \tilde{\theta} \quad \Longrightarrow \quad \theta \sim \tilde{\theta}.$$

Proof. Let $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$. We define

$$\epsilon = \min\Big(\big\{|w_{v \to v'}|,\ v \to v' \in E \text{ and } w_{v \to v'} \neq 0\big\} \ \cup\ \big\{|b_v|,\ v \in B \text{ and } b_v \neq 0\big\}\Big).$$

Let $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\|\theta - \tilde{\theta}\|_\infty < \epsilon$ and $\theta \overset{R}{\sim} \tilde{\theta}$. To prove $\theta \sim \tilde{\theta}$, we simply have to prove $\operatorname{sign}(\theta) = \operatorname{sign}(\tilde{\theta})$. There exists $(\lambda^0, \dots, \lambda^L) \in (\mathbb{R}^*)^{V_0} \times \cdots \times (\mathbb{R}^*)^{V_L}$, with $\lambda^0 = \mathbf{1}_{V_0}$ and $\lambda^L = \mathbf{1}_{V_L}$, such that, for all $l \in [\![1, L]\!]$, for all $(v_{l-1}, v_l) \in V_{l-1} \times V_l$, (4.2.6) holds. Let us show that $\operatorname{sign}(\theta) = \operatorname{sign}(\tilde{\theta})$.

Indeed, let $l \in [\![1, L]\!]$, and let $(v, v') \in V_{l-1} \times V_l$. If $w_{v \to v'} \neq 0$, then since $|w_{v \to v'} - \tilde{w}_{v \to v'}| < \epsilon$ and by definition $\epsilon \leq |w_{v \to v'}|$, we have $\operatorname{sign}(w_{v \to v'}) = \operatorname{sign}(\tilde{w}_{v \to v'})$. Otherwise, if $w_{v \to v'} = 0$, (4.2.6) shows that we have

$$\tilde{w}_{v \to v'} = \frac{\lambda_v^{l-1}}{\lambda_{v'}^l} w_{v \to v'} = 0,$$

so we still have $\operatorname{sign}(w_{v\to v'}) = \operatorname{sign}(\tilde{w}_{v\to v'})$.

Now let $l \in [\![1, L]\!]$ and let $v \in V_l$. Similarly, if $b_v \neq 0$, we have $|b_v - \tilde{b}_v| < \epsilon \leq |b_v|$, so $\operatorname{sign}(b_v) = \operatorname{sign}(\tilde{b}_v)$, and if $b_v = 0$, we have

$$\tilde{b}_v = \frac{b_v}{\lambda_v^l} = 0,$$

so again $\operatorname{sign}(b_v) = \operatorname{sign}(\tilde{b}_v)$.

This shows $\operatorname{sign}(\theta) = \operatorname{sign}(\tilde{\theta})$, so $\theta \sim \tilde{\theta}$. □

4.A.3 The smooth manifold structure of Σ_1^*

In this section, we prove Theorem 76, which is restated as Theorem 94. Before doing so, we establish intermediary results, some of which are evoked in Section 4.3.

Let us discuss the cardinal of F_θ defined in Section 4.3. The set F_θ is obtained by removing the edges of the form $v \to s^\theta_{\max}(v)$ for $v \in V_1 \cup \cdots \cup V_{L-1}$. Note that we do not remove the edges of the form $v \to s^\theta_{\max}(v)$ for $v \in V_0$. For all $l \in [\![1, L-1]\!]$, there are precisely N_l edges of the form $(v, s^\theta_{\max}(v))$ with $v \in V_l$, so

$$\begin{aligned} |F_\theta| &= |E| - (N_1 + \cdots + N_{L-1}) \\ &= N_0N_1 + \cdots + N_{L-1}N_L - N_1 - \cdots - N_{L-1}. \end{aligned}$$

As a consequence, since $|B| = N_1 + \cdots + N_L$, we have in particular

$$\begin{aligned} |F_\theta| + |B| &= N_0N_1 + \cdots + N_{L-1}N_L - N_1 - \cdots - N_{L-1} + N_1 + \cdots + N_L \\ &= N_0N_1 + \cdots + N_{L-1}N_L + N_L. \end{aligned} \tag{4.A.7}$$

The following proposition is a first step towards Proposition 90, which states that ψ^θ is a homeomorphism.

Proposition 89. *For all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$, the function $\psi^\theta : U_\theta \to \mathbb{R}^{\mathcal{P}\times V_L}$ is injective.*

Proof. Let $\tau, \tilde{\tau} \in U_\theta$ such that $\psi^\theta(\tau) = \psi^\theta(\tilde{\tau})$. Let us show $\tau = \tilde{\tau}$. We have $\phi(\rho_\theta(\tau)) = \phi(\rho_\theta(\tilde{\tau}))$ and by definition of U_θ, $\rho_\theta(\tau) \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$, so by Proposition 87 we have the rescaling equivalence

$$\rho_\theta(\tau) \overset{R}{\sim} \rho_\theta(\tilde{\tau}).$$

By definition of the rescaling equivalence, in its formulation (4.2.6), there exists $(\lambda^0, \ldots, \lambda^L) \in (\mathbb{R}^*)^{V_0} \times \cdots \times (\mathbb{R}^*)^{V_L}$, with $\lambda^0 = \mathbf{1}_{V_0}$ and $\lambda^L = \mathbf{1}_{V_L}$, such that, for all $l \in [\![1, L]\!]$, for all $(v_{l-1}, v_l) \in V_{l-1} \times V_l$,

$$\begin{cases} \rho_\theta(\tau)_{v_{l-1}\to v_l} = \frac{(\lambda^l)_{v_l}}{(\lambda^{l-1})_{v_{l-1}}} \rho_\theta(\tilde{\tau})_{v_{l-1}\to v_l} \\ b_{v_l} = \lambda^l_{v_l} \tilde{b}_{v_l}. \end{cases} \tag{4.A.8}$$

Let $l \in [\![2, L]\!]$ and let $v_{l-1} \in V_{l-1}$. Let $v_l = s^\theta_{\max}(v_{l-1})$. According to (4.A.8) we have

$$\rho_\theta(\tau)_{v_{l-1}\to v_l} = \frac{(\lambda^l)_{v_l}}{(\lambda^{l-1})_{v_{l-1}}}\rho_\theta(\tilde{\tau})_{v_{l-1}\to v_l}.$$

But since $v_l = s^\theta_{\max}(v_{l-1})$ and $v_{l-1} \in V_{l-1}$ with $l-1 \in [\![1, L-1]\!]$, we have $v_{l-1} \to v_l \in E\backslash F_\theta$, so by definition of ρ_θ in (4.3.2),

$$\rho_\theta(\tau)_{v_{l-1}\to v_l} = w_{v_{l-1}\to v_l} = \rho_\theta(\tilde{\tau})_{v_{l-1}\to v_l} \neq 0,$$

so $\frac{(\lambda^l)_{v_l}}{(\lambda^{l-1})_{v_{l-1}}} = 1$.

We have shown that for all $l \in [\![2, L]\!]$, for all $v_{l-1} \in V_{l-1}$, there exists $v_l \in V_l$ such that

$$(\lambda^{l-1})_{v_{l-1}} = (\lambda^l)_{v_l}.$$

As a consequence, if l is such that $\lambda^l = \mathbf{1}_{V_l}$, then $\lambda^{l-1} = \mathbf{1}_{V_{l-1}}$.

Starting from $\lambda^L = \mathbf{1}_{V_L}$, this shows by induction that for all $l \in [\![1, L]\!]$,

$$\lambda^l = \mathbf{1}_{V_l}.$$

By hypothesis we also have $\lambda^0 = \mathbf{1}_{V_0}$. Using (4.A.8), this shows that

$$\rho_\theta(\tau) = \rho_\theta(\tilde{\tau}).$$

The injectivity of ρ_θ allows us to conclude that

$$\tau = \tilde{\tau}.$$

□

The following proposition shows, as mentioned in Section 4.3, that ψ^θ is a homeomorphism. This is a necessary step to prove that $(V_\theta, (\psi^\theta)^{-1})_{\theta\in(\mathbb{R}^E\times\mathbb{R}^B)\backslash S}$ is an atlas of Σ_1^*.

Proposition 90. *For all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right)\backslash S$, ψ^θ is a homeomorphism from U_θ onto its image V_θ.*

Proof. We already know from Proposition 89 that ψ^θ is injective, so we need to prove that ψ^θ is continuous and its inverse is continuous. The function ρ_θ is affine and ϕ is a polynomial function, so the function $\psi^\theta = \phi \circ \rho_\theta$ is a polynomial function, and in particular it is continuous.

To prove that $(\psi^\theta)^{-1}$ is continuous, we consider a sequence (τ_n) taking values in U_θ and $\tau \in U_\theta$ such that $\psi^\theta(\tau_n) \to \psi^\theta(\tau)$, and we want to show that $\tau_n \to \tau$.

Let us first show that for all $v \in B$, $(\tau_n)_v \to \tau_v$. Indeed, let $l \in [\![1, L]\!]$ and let $v_l \in V_l$, so that v_l is an arbitrary element of B. Let us define $v_{l+1} = s^\theta_{\max}(v_l)$, then $v_{l+2} = s^\theta_{\max}(v_{l+1})$ and so on up to $v_L = s^\theta_{\max}(v_{L-1})$. Since for all $l' \in [\![l, L-1]\!]$, $v_{l'+1} = s^\theta_{\max}(v_{l'})$, by definition of F_θ and ρ_θ (see (4.3.1) and (4.3.2)), we have

$$\rho_\theta(\tau_n)_{v_{l'}\to v_{l'+1}} = w_{v_{l'}\to v_{l'+1}}, \tag{4.A.9}$$

and

$$\rho_\theta(\tau)_{v_{l'}\to v_{l'+1}} = w_{v_{l'}\to v_{l'+1}}. \tag{4.A.10}$$

In particular, since $\theta \notin S$, for all $l' \in [\![l, L-1]\!]$ we have $w_{v_{l'}\to\bullet} \neq 0$, so by definition of $s^\theta_{\max}$, $w_{v_{l'}\to v_{l'+1}} \neq 0$. We thus have

$$w_{v_l\to v_{l+1}} \dots w_{v_{L-1}\to v_L} \neq 0. \tag{4.A.11}$$

If we denote $p = (v_l, \dots, v_{L-1})$, we have, using the definition of ϕ and (4.A.9),

$$\psi^\theta_{p,v_L}(\tau_n) = (\tau_n)_{v_l} w_{v_l\to v_{l+1}} \dots w_{v_{L-1}\to v_L}$$

and using (4.A.10),

$$\psi^\theta_{p,v_L}(\tau) = (\tau)_{v_l} w_{v_l\to v_{l+1}} \dots w_{v_{L-1}\to v_L}.$$

Using (4.A.11) and the fact that

$$\psi^\theta(\tau_n) \to \psi^\theta(\tau),$$

we conclude that

$$(\tau_n)_{v_l} \to \tau_{v_l}.$$

Let us now prove that for all $(v, v') \in E$, $(\tau_n)_{v\to v'} \to \tau_{v\to v'}$. Let us show by induction on $l \in [\![1, L]\!]$ the following hypothesis

$$\forall l' \in [\![1, l]\!], \quad \forall (v, v') \in (V_{l'-1} \times V_{l'}) \cap F_\theta, \quad (\tau_n)_{v\to v'} \longrightarrow \tau_{v\to v'}. \tag{H_l}$$

Initialization. Let $(v_0, v_1) \in (V_0 \times V_1) \cap F_\theta$. We define $v_2 = s^\theta_{\max}(v_1)$, then we define $v_3 = s^\theta_{\max}(v_2)$, and so on up to $v_L = s^\theta_{\max}(v_{L-1})$. Let $p = (v_0, \dots, v_{L-1}) \in \mathcal{P}$.

As above, using the definition of ρ_θ, F_θ and ϕ, we have

$$\psi^\theta_{p,v_L}(\tau_n) = (\tau_n)_{v_0\to v_1} w_{v_1\to v_2} \dots w_{v_{L-1}\to v_L}$$

and

$$\psi^\theta_{p,v_L}(\tau) = (\tau)_{v_0\to v_1} w_{v_1\to v_2} \dots w_{v_{L-1}\to v_L},$$

and since $\theta \notin S$, we also have , as above,

$$w_{v_1\to v_2} \dots w_{v_{L-1}\to v_L} \neq 0. \tag{4.A.12}$$

Since

$$\psi^\theta(\tau_n) \longrightarrow \psi^\theta(\tau)$$

we conclude using (4.A.12) that

$$(\tau_n)_{v_0\to v_1} \longrightarrow \tau_{v_0\to v_1}.$$

We have shown H_1.

Induction step. Let $l \in [\![2, L]\!]$ and let us assume that H_{l-1} holds.

Let $(v_{l-1}, v_l) \in (V_{l-1} \times V_l) \cap F_\theta$. We define $v_{l+1} = s^\theta_{\max}(v_l), v_{l+2} = s^\theta_{\max}(v_{l+1})$, and so on up to $v_L = s^\theta_{\max}(v_{L-1})$. Let us denote $p_o = (v_l, \dots, v_L)$. Recalling the notation defined in (4.A.5), we have

$$\rho_\theta(\tau_n)_{p_o} = w_{v_l \to v_{l+1}} \dots w_{v_{L-1} \to v_L} = \rho_\theta(\tau)_{p_o} \neq 0. \tag{4.A.13}$$

At the same time, since $\tau \in U_\theta$, Proposition 84 shows there exist $l' \in [\![0, l-1]\!]$ and a path $p_i = (v_{l'}, \dots, v_{l-2}, v_{l-1})$ such that

$$\rho_\theta(\tau)_{p_i} \neq 0. \tag{4.A.14}$$

If $l' \geq 1$, we have shown in the first part of the proof that $(\tau_n)_{v_{l'}} \longrightarrow \tau_{v_{l'}}$. Moreover, whatever the value of l' is, for $k \in [\![l', l-2]\!]$, if $(v_k, v_{k+1}) \in E \backslash F_\theta$,

$$\rho_\theta(\tau_n)_{v_k \to v_{k+1}} = w_{v_k \to v_{k+1}} = \rho_\theta(\tau)_{v_k \to v_{k+1}},$$

and if $(v_k, v_{k+1}) \in F_\theta$, according to H_{l-1},

$$\rho_\theta(\tau_n)_{v_k \to v_{k+1}} = (\tau_n)_{v_k \to v_{k+1}} \longrightarrow \tau_{v_k \to v_{k+1}} = \rho_\theta(\tau)_{v_k \to v_{k+1}}.$$

We therefore have

$$\rho_\theta(\tau_n)_{p_i} \longrightarrow \rho_\theta(\tau)_{p_i}, \tag{4.A.15}$$

and in particular, since $\rho_\theta(\tau)_{p_i} \neq 0$, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,

$$\rho_\theta(\tau_n)_{p_i} \neq 0. \tag{4.A.16}$$

We can write

$$\psi^\theta_{p,v_L}(\tau_n) = \rho_\theta(\tau_n)_{p_i}\, (\tau_n)_{v_{l-1} \to v_l}\, \rho_\theta(\tau_n)_{p_o}$$

and

$$\psi^\theta_{p,v_L}(\tau) = \rho_\theta(\tau)_{p_i}\, (\tau)_{v_{l-1} \to v_l}\, \rho_\theta(\tau)_{p_o},$$

so using (4.A.13), (4.A.16) and (4.A.15), we have

$$(\tau_n)_{v_{l-1} \to v_l} = \frac{\psi^\theta_{p,v_L}(\tau_n)}{\rho_\theta(\tau_n)_{p_i} \rho_\theta(\tau_n)_{p_o}} \quad \longrightarrow \quad \frac{\psi^\theta_{p,v_L}(\tau)}{\rho_\theta(\tau)_{p_i} \rho_\theta(\tau)_{p_o}} = \tau_{v_{l-1} \to v_l}.$$

We have shown H_l, which concludes the induction step.

In particular, H_L is satisfied, and finally $\tau_n \to \tau$.

This shows that ψ^θ is a homeomophism. □

The following lemma is necessary for the proof of Proposition 92.

Lemma 91. *Let $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$. Let $(v, v') \in E$ (resp. $v \in B$). If $w_{v \to v'} \neq 0$ (resp. $b_v \neq 0$), then there exists $\epsilon > 0$ such that for all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, if $\|\phi(\theta) - \phi(\tilde{\theta})\|_\infty < \epsilon$, then $\tilde{w}_{v \to v'} \neq 0$ (resp. $\tilde{b}_v \neq 0$).*

Proof. Let $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$ and $(v, v') \in E$ such that $w_{v\to v'} \neq 0$. Denote $l \in [\![0, L-1]\!]$ such that $v \in V_l$. If $l = 0$, we take $p_i = (v)$ so that by convention $\theta_{p_i} = 1 \neq 0$, and if $l \geq 1$, we use Proposition 84 which states that there exists $l' \in [\![0, l-1]\!]$ and a path $p_i = (v_{l'}, \dots, v_{l-2}, v)$ such that $\theta_{p_i} \neq 0$. Similarly, if $l = L-1$, we take $p_o = (v')$ so that by convention $\theta_{p_o} = 1 \neq 0$ and if $l < L-1$, we use Proposition 84 which states that there exists a path $p_o = (v', v_{l+1}, \dots, v_L)$ such that $\theta_{p_o} \neq 0$. If we denote

$$p = \begin{cases} (v, v', v_{l+2}, \dots, v_{L-1}) & \text{if } l = 0 \\ (v_{l'}, \dots, v_{l-1}, v, v') & \text{if } l = L-1 \\ (v_{l'}, \dots, v_{l-1}, v, v', v_{l+2}, \dots, v_{L-1}) & \text{otherwise,} \end{cases}$$

we have

$$\phi_{p,v_L}(\theta) = \theta_{p_i} w_{v\to v'} \theta_{p_o} \neq 0.$$

We define $\epsilon = |\phi_{p,v_L}(\theta)| > 0$. For all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\|\phi(\tilde{\theta}) - \phi(\theta)\|_\infty < \epsilon$ we have

$$\phi_{p,v_L}(\tilde{\theta}) \neq 0.$$

Since $\phi_{p,v_L}(\tilde{\theta}) = \tilde{\theta}_{p_i} \tilde{w}_{v\to v'} \tilde{\theta}_{p_o}$, this implies in particular that

$$\tilde{w}_{v\to v'} \neq 0.$$

The proof is similar in the case $v \in B$ and $b_v \neq 0$. □

The following proposition, which states that for any $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$, $V_\theta = \psi^\theta(U_\theta)$ is open with respect to the topology induced on Σ_1^* by the standard topology of $\mathbb{R}^{\mathcal{P}\times V_L}$, is necessary to show that $(V_\theta, (\psi^\theta)^{-1})_{\theta \in (\mathbb{R}^E \times \mathbb{R}^B)\backslash S}$ is an atlas of Σ_1^*.

Proposition 92. *For any $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$, for any $\tau \in U_\theta$, there exists $\epsilon > 0$ such that*

$$\Sigma_1^* \cap B_\infty(\psi^\theta(\tau), \epsilon) \subset V_\theta.$$

Proof. Let us first construct ϵ and then consider an element of the set on the left of the inclusion and prove it belongs to V_θ. Let $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$ and $\tau \in U_\theta$. For all $l \in [\![1, L-1]\!]$, for all $v \in V_l$, by definition of F_θ and ρ_θ, we have $\rho_\theta(\tau)_{v\to s^\theta_{\max}(v)} = w_{v\to s^\theta_{\max}(v)}$, and since $\theta \notin S$, by definition of $s^\theta_{\max}$, $w_{v\to s^\theta_{\max}(v)} \neq 0$, so according to Lemma 91 there exists $\epsilon_v > 0$ such that for all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$,

$$\|\phi(\rho_\theta(\tau)) - \phi(\tilde{\theta})\|_\infty < \epsilon_v \quad \Longrightarrow \quad \tilde{w}_{v\to s^\theta_{\max}(v)} \neq 0.$$

Let $\epsilon = \min_{v \in V_1 \cup \dots \cup V_{L-1}} \epsilon_v$.

Let us now show the inclusion: let $\tilde{\theta} \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$ such that $\|\phi(\rho_\theta(\tau)) - \phi(\tilde{\theta})\|_\infty < \epsilon$, and let us show that $\phi(\tilde{\theta}) \in V_\theta$. Notice first that for all $l \in [\![1, L-1]\!]$ and $v \in V_l$, by definition of ϵ, $w_{v\to s^\theta_{\max}(v)} \neq 0$ and $\tilde{w}_{v\to s^\theta_{\max}(v)} \neq 0$. We are going to define $\tilde{\tau} \in U_\theta$ such that $\rho_\theta(\tilde{\tau}) \overset{R}{\sim} \tilde{\theta}$, so that using Proposition 85, $\psi^\theta(\tilde{\tau}) = \phi(\tilde{\theta})$.

Let us define recursively a family $(\lambda^0, \dots, \lambda^L) \in (\mathbb{R}^*)^{V_0} \times \dots \times (\mathbb{R}^*)^{V_L}$ as follows:

— we define $\lambda^L = \mathbf{1}_{V_L}$;
— for all $l \in [\![1, L-1]\!]$, for all $v \in V_l$, we define

$$\lambda_v^l = \frac{\tilde{w}_{v \to s^\theta_{\max}(v)}}{w_{v \to s^\theta_{\max}(v)}} \lambda^{l+1}_{s^\theta_{\max}(v)}. \tag{4.A.17}$$

— we define finally $\lambda^0 = \mathbf{1}_{V_0}$.

Note that for all $l \in [\![0, L]\!]$ and for all $v \in V_l$, $\lambda_v^l \neq 0$. Also note that for all $l \in [\![2, L]\!]$, for all $v \in V_{l-1}$, reformulating (4.A.17) in a way that will be useful later, we have

$$\frac{\lambda^l_{s^\theta_{\max}(v)}}{\lambda_v^{l-1}} = \frac{w_{v \to s^\theta_{\max}(v)}}{\tilde{w}_{v \to s^\theta_{\max}(v)}}. \tag{4.A.18}$$

We then define $\tilde{\tau} \in \mathbb{R}^{F_\theta} \times \mathbb{R}^B$ by:
— for all $l \in [\![1, L]\!]$, for all $(v, v') \in (V_{l-1} \times V_l) \cap F_\theta$,

$$\tilde{\tau}_{v \to v'} = \frac{\lambda^l_{v'}}{\lambda_v^{l-1}} \tilde{w}_{v \to v'}; \tag{4.A.19}$$

— for all $l \in [\![1, L]\!]$, for all $v \in V_l$,

$$\tilde{\tau}_v = \lambda_v^l \tilde{b}_v. \tag{4.A.20}$$

Let us show $\rho_\theta(\tilde{\tau}) \overset{R}{\sim} \tilde{\theta}$. Indeed, let $l \in [\![1, L]\!]$ and let $(v, v') \in V_{l-1} \times V_l$. If $v \in V_0$ or $v \in V_1 \cup \cdots \cup V_{L-1}$ and $v' \neq s^\theta_{\max}(v)$, then by definition (4.3.1) of F_θ, we have $v \to v' \in F_\theta$, so using (4.3.2) and (4.A.19) we have

$$\rho_\theta(\tilde{\tau})_{v \to v'} = \tilde{\tau}_{v \to v'} = \frac{\lambda^l_{v'}}{\lambda_v^{l-1}} \tilde{w}_{v \to v'}. \tag{4.A.21}$$

If $v \in V_1 \cup \cdots \cup V_{L-1}$ and $v' = s^\theta_{\max}(v)$, then by definition (4.3.1) of F_θ, we have $v \to v' \in E \backslash F_\theta$, and since in that case, $l \geq 2$, using (4.3.2) and (4.A.18), we see that

$$\rho_\theta(\tilde{\tau})_{v \to v'} = w_{v \to v'} = \frac{\lambda^l_{v'}}{\lambda_v^{l-1}} \tilde{w}_{v \to v'}. \tag{4.A.22}$$

If $v \in B$, using (4.3.2) and (4.A.20), we have

$$\rho_\theta(\tilde{\tau})_v = \tilde{\tau}_v = \lambda_v^l \tilde{b}_v. \tag{4.A.23}$$

Equations (4.A.21), (4.A.22) and (4.A.23) prove that

$$\rho_\theta(\tilde{\tau}) \overset{R}{\sim} \tilde{\theta}.$$

Using Corollary 86, since $\tilde{\theta} \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$ and $\rho_\theta(\tilde{\tau}) \overset{R}{\sim} \tilde{\theta}$, we also have $\rho_\theta(\tilde{\tau}) \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$. Since, by definition, $U_\theta = \rho_\theta^{-1}\left(\left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S\right)$, we have $\tilde{\tau} \in U_\theta$. We have shown

$$\Sigma_1^* \cap B_\infty(\psi^\theta(\tau), \epsilon) \ \subset \ V_\theta.$$

□

The following proposition is necessary in order to show that $(V_\theta, (\psi^\theta)^{-1})_{\theta\in(\mathbb{R}^E\times\mathbb{R}^B)\setminus S}$ is an atlas of Σ_1^*.

Proposition 93. *For all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right)\setminus S$, the function ψ^θ is C^∞ and its differential $D\psi^\theta(\tau)$ is injective for all $\tau \in U_\theta$.*

Proof. Let $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right)\setminus S$. First of all, ψ^θ is a polynomial function as a composition of ϕ and ρ_θ which are both polynomial functions. So, ψ^θ is C^∞.

In order to show the injectivity of the differential $D\psi^\theta(\tau)$ for all $\tau \in U_\theta$, let us compute the partial derivatives of $\psi^\theta_{p,v_L}(\tau)$. Let $p \in \mathcal{P}$ and $v_L \in V_L$. Using the definition of ψ^θ and ϕ, three cases are possible.

Case 1. The path p is of the form $(v_0, v_1, \dots, v_{L-1})$. We have

$$\psi^\theta_{p,v_L}(\tau) = \rho_\theta(\tau)_{v_0\to v_1} \dots \rho_\theta(\tau)_{v_{L-1}\to v_L}.$$

Case 2. The path p is of the form $(v_l, \dots, v_{L-1})$ with $l \in [\![1, L-1]\!]$. We have, for all $\tau \in U_\theta$,

$$\psi^\theta_{p,v_L}(\tau) = \tau_{v_l}\rho_\theta(\tau)_{v_l\to v_{l+1}} \dots \rho_\theta(\tau)_{v_{L-1}\to v_L}.$$

Case 3. For $p = \beta$, we have, for all $\tau \in U_\theta$,

$$\psi^\theta_{p,v_L}(\tau) = \tau_{v_L}.$$

Let $(v, v') \in F_\theta$, and let us compute $\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v\to v'}}(\tau)$.

Case 1. We have $p = (v_0, \dots, v_{L-1}) \in \mathcal{P}_0$. If $\{v, v'\} \subset \{v_0, \dots, v_L\}$, there exists $l \in [\![0, L-1]\!]$ such that $(v, v') = (v_l, v_{l+1})$, in which case, since $(v, v') \in F_\theta$, $\rho_\theta(\tau)_{v_l\to v_{l+1}} = \tau_{v_l\to v_{l+1}}$ and

$$\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v\to v'}}(\tau) = \prod_{\substack{k\in[\![0,L-1]\!]\\ k\neq l}} \rho_\theta(\tau)_{v_k\to v_{k+1}}. \tag{4.A.24}$$

Otherwise if $\{v, v'\} \not\subset \{v_0, \dots, v_L\}$,

$$\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v\to v'}}(\tau) = 0.$$

Case 2. We have $p = (v_l, \dots, v_{L-1}) \in \mathcal{P}_l$, for $l \in [\![1, L-1]\!]$. If $\{v, v'\} \subset \{v_l, \dots, v_L\}$, there exists $l' \in [\![l, L-1]\!]$ such that $(v, v') = (v_{l'}, v_{l'+1})$, in which case, since $(v, v') \in F_\theta$, $\rho_\theta(\tau)_{v_{l'}\to v_{l'+1}} = \tau_{v_{l'}\to v_{l'+1}}$ and

$$\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v\to v'}}(\tau) = \tau_{v_l} \prod_{\substack{k\in[\![l,L-1]\!]\\ k\neq l'}} \rho_\theta(\tau)_{v_k\to v_{k+1}}. \tag{4.A.25}$$

Otherwise if $\{v, v'\} \not\subset \{v_l, \dots, v_L\}$,

$$\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v\to v'}}(\tau) = 0.$$

Case 3. We have $p = \beta$. In that case, we have

$$\frac{\partial \psi^{\theta}_{p,v_L}}{\partial \tau_{v \to v'}}(\tau) = 0.$$

Now let $v \in B$, and let us compute $\frac{\partial \psi^{\theta}_{p,v_L}}{\partial \tau_v}(\tau)$.
Case 1. We have $p = (v_0, \dots, v_{L-1}) \in \mathcal{P}_0$ and

$$\frac{\partial \psi^{\theta}_{p,v_L}}{\partial \tau_v}(\tau) = 0.$$

Case 2. We have $p = (v_l, \dots, v_{L-1}) \in \mathcal{P}_l$ for $l \in [\![1, L-1]\!]$. If $v = v_l$, then

$$\frac{\partial \psi^{\theta}_{p,v_L}}{\partial \tau_v}(\tau) = \prod_{k \in [\![l, L-1]\!]} \rho_\theta(\tau)_{v_k \to v_{k+1}}.$$

If $v \neq v_l$,

$$\frac{\partial \psi^{\theta}_{p,v_L}}{\partial \tau_v}(\tau) = 0.$$

Case 3. We have $p = \beta$ and

$$\frac{\partial \psi^{\theta}_{p,v_L}}{\partial \tau_v}(\tau) = \begin{cases} 1 & \text{if } v = v_L \\ 0 & \text{if } v \neq v_L. \end{cases}$$

Now that we know the partial derivatives, let us show $D\psi^\theta(\tau)$ is injective for all $\tau \in U_\theta$. Let $\tau \in U_\theta$ and let $h \in \mathbb{R}^{F_\theta} \times \mathbb{R}^B$ such that

$$D\psi^\theta(\tau) \cdot h = 0.$$

We need to prove that $h = 0$.

Let us show first that for all $v \in B$, $h_v = 0$. Let $l \in [\![1, L-1]\!]$, and let $v_l \in V_l$ so that v_l is arbitrary in $B \backslash V_L$. Let us define $v_{l+1} = s^\theta_{max}(v_l)$, then $v_{l+2} = s^\theta_{\max}(v_{l+1})$, and so on up to $v_L = s^\theta_{max}(v_{L-1})$. Let us denote $p = (v_l, \dots, v_{L-1})$. We have

$$\psi^{\theta}_{p,v_L}(\tau) = \tau_{v_l} w_{v_l \to v_{l+1}} \dots w_{v_{L-1} \to v_L},$$

so

$$\left[D\psi^\theta(\tau) \cdot h\right]_{p,v_L} = \frac{\partial \psi^{\theta}_{p,v_L}}{\partial \tau_{v_l}}(\tau) h_{v_l} = w_{v_l \to v_{l+1}} \dots w_{v_{L-1} \to v_L} h_{v_l}.$$

Since $\left[D\psi^\theta(\tau) \cdot h\right]_{p,v_L} = 0$ and $w_{v_l \to v_{l+1}} \dots w_{v_{L-1} \to v_L} \neq 0$, we conclude that $h_{v_l} = 0$. Now let $v_L \in V_L$. We consider $p = \beta$ and we have

$$\left[D\psi^\theta(\tau) \cdot h\right]_{p,v_L} = h_{v_L}.$$

Since $\left[D\psi^\theta(\tau) \cdot h\right]_{p,v_L} = 0$, we also conclude in that case that $h_{v_L} = 0$.

Let us now show that for all $(v, v') \in F_\theta$, $h_{v \to v'} = 0$. Let $l \in [\![1, L]\!]$ and let $(v_{l-1}, v_l) \in (V_{l-1} \times V_l) \cap F_\theta$ so that (v_{l-1}, v_l) is arbitrary in F_θ. If $l = 1$, we define

$p_i = (v_{l-1})$ and we have by convention $\theta_{p_i} = 1 \neq 0$. If $l > 1$, using Proposition 84 there exist $l' \in [\![0, l-1]\!]$ and a path $p_i = (v_{l'}, \dots, v_{l-1})$ such that $\rho_\theta(\tau)_{p_i} \neq 0$. If $l < L$, we define $v_{l+1} = s^\theta_{\max}(v_l)$, then $v_{l+2} = s^\theta_{\max}(v_{l+1})$, and so on up to $v_L = s^\theta_{\max}(v_{L-1})$, and we denote $p = p_i \cup (v_{l-1}, v_l, \dots, v_{L-1})$. If $l = L$, we denote $p = p_i$. Let us show the following expression.

$$\Big[D\psi^\theta(\tau)\cdot h\Big]_{p,v_L} = \sum_{\substack{k\in[\![l',l-1]\!]\\(v_k,v_{k+1})\in F_\theta}} \frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_k\to v_{k+1}}}(\tau)h_{v_k\to v_{k+1}} \tag{4.A.26}$$

Indeed, if $l' \geq 1$, we have

$$\psi^\theta_{p,v_L}(\tau) = \tau_{v_{l'}} \prod_{k=l'}^{l-1} \rho_\theta(\tau)_{v_k\to v_{k+1}} \prod_{k=l}^{L-1} w_{v_k\to v_{k+1}},$$

with the classical convention that if $l = L$, the product on the right is empty thus equal to 1. We thus have

$$\begin{aligned}\Big[D\psi^\theta(\tau)\cdot h\Big]_{p,v_L} &= \frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_{l'}}}(\tau)h_{v_{l'}} + \sum_{\substack{k\in[\![l',l-1]\!]\\(v_k,v_{k+1})\in F_\theta}} \frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_k\to v_{k+1}}}(\tau)h_{v_k\to v_{k+1}}\\ &= \sum_{\substack{k\in[\![l',l-1]\!]\\(v_k,v_{k+1})\in F_\theta}} \frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_k\to v_{k+1}}}(\tau)h_{v_k\to v_{k+1}},\end{aligned}$$

since we have already shown that $h_{v_{l'}} = 0$.

If $l' = 0$, we have

$$\psi^\theta_{p,v_L}(\tau) = \prod_{k=0}^{l-1} \rho_\theta(\tau)_{v_k\to v_{k+1}} \prod_{k=l}^{L-1} w_{v_k\to v_{k+1}},$$

with the same convention that when $l = L$ the product on the right is equal to 1, so again

$$\Big[D\psi^\theta(\tau)\cdot h\Big]_{p,v_L} = \sum_{\substack{k\in[\![0,l-1]\!]\\(v_k,v_{k+1})\in F_\theta}} \frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_k\to v_{k+1}}}(\tau)h_{v_k\to v_{k+1}}.$$

This concludes the proof of (4.A.26).

We can now show by induction the following statement, for $l \in [\![0, L]\!]$.

$$\forall l' \in [\![1, l]\!],\ \forall (v, v') \in (V_{l'-1} \times V_{l'}) \cap F_\theta,\ h_{v\to v'} = 0. \tag{H_l}$$

Since $[\![1, 0]\!] = \emptyset$, H_0 is trivially true. Now let $l \in [\![1, L]\!]$ and suppose H_{l-1} is true. We consider $(v_{l-1}, v_l) \in (V_{l-1} \times V_l) \cap F_\theta$, and $l' \in [\![0, l]\!]$, p_i and p just as before.

Since for all $k \in [\![0, l-2]\!]$, the induction hypothesis guarantees that $h_{v_k \to v_{k+1}} = 0$, (4.A.26) becomes

$$\left[D\psi^\theta(\tau) \cdot h\right]_{p,v_L} = \frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v_{l-1}\to v_l}}(\tau) h_{v_{l-1}\to v_l}.$$

Using (4.A.24) and (4.A.25), we obtain

$$\left[D\psi^\theta(\tau) \cdot h\right]_{p,v_L} = \begin{cases} \rho_\theta(\tau)_{p_i} w_{v_l \to v_{l+1}} \dots w_{v_{L-1}\to v_L} h_{v_{l-1}\to v_l} & \text{if } l < L \\ \rho_\theta(\tau)_{p_i} h_{v_{l-1}\to v_l} & \text{if } l = L. \end{cases}$$

Since $\rho_\theta(\tau)_{p_i} \neq 0$, and for $l < L$, $w_{v_l \to v_{l+1}} \dots w_{v_{L-1}\to v_L} \neq 0$, we conclude that $h_{v_{l-1}\to v_l} = 0$ and that H_l holds.

This induction leads to the conclusion that $h = 0$ and $D\psi^\theta(\tau)$ is injective. □

We are now equipped to prove Theorem 76, which we restate here.

Theorem 94. *Σ^1_* is a smooth manifold of $\mathbb{R}^{\mathcal{P}\times V_L}$ of dimension*

$$|F_\theta| + |B| = N_0N_1 + N_1N_2 + \dots + N_{L-1}N_L + N_L,$$

and the family $(V_\theta, (\psi^\theta)^{-1})_{\theta \in (\mathbb{R}^E \times \mathbb{R}^B)\setminus S}$ is an atlas.

Proof. Our goal is to show that the family $(V_\theta, (\psi^\theta)^{-1})_{\theta \in (\mathbb{R}^E \times \mathbb{R}^B)\setminus S}$ is a smooth atlas, which will show that Σ^*_1 is a smooth manifold.

We already know from Proposition 92 that for any $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right)\setminus S$, V_θ is an open subset of Σ^*_1 and from Proposition 90 that $(\psi^\theta)^{-1}$ is a homeomorphism from V_θ onto U_θ. Since for any $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right)\setminus S$, $\tau_\theta \in U_\theta$, we have $\phi(\theta) = \psi^\theta(\tau_\theta) \in V_\theta$ which shows that $(V_\theta)_{\theta \in (\mathbb{R}^E \times \mathbb{R}^B)\setminus S}$ covers Σ^*_1.

Let $\theta, \tilde\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right)\setminus S$, let us show that the transition map

$$(\psi^\theta)^{-1} \circ \psi^{\tilde\theta} \ : \ (\psi^{\tilde\theta})^{-1}(V_\theta \cap V_{\tilde\theta}) \ \to \ (\psi^\theta)^{-1}(V_\theta \cap V_{\tilde\theta})$$

is smooth.

Let $\tau_0 \in U_{\tilde\theta}$ such that $\tau_0 \in (\psi^{\tilde\theta})^{-1}(V_\theta \cap V_{\tilde\theta})$. We are going to show that the function $(\psi^\theta)^{-1} \circ \psi^{\tilde\theta}$ is C^∞ in a neighborhood of τ_0.

For ease of reading, let us denote $\psi^{\tilde\theta}(\tau_0)$ by η_0. By definition, $\eta_0 \in V_\theta \cap V_{\tilde\theta}$. In particular, since $\eta_0 \in V_\theta$, we can define $\tau_1 = (\psi^\theta)^{-1}(\eta_0)$. See Figure 4.3 for a representation.

Let $T = \operatorname{Im} D\psi^\theta(\tau_1)$, and let us consider a linear subspace G such that $T \oplus G = \mathbb{R}^{\mathcal{P}\times V_L}$. Let $N_C = |\mathcal{P}|N_L - |F_\theta| - |B| = \dim(G)$. Let $i : \mathbb{R}^{N_C} \to G$ be linear and invertible. Let us consider the function

$$\begin{array}{rccc} \varphi_\theta : & U_\theta \times \mathbb{R}^{N_C} & \longrightarrow & \mathbb{R}^{\mathcal{P}\times V_L} \\ & (\tau, x) & \longmapsto & \psi^\theta(\tau) + i(x). \end{array}$$

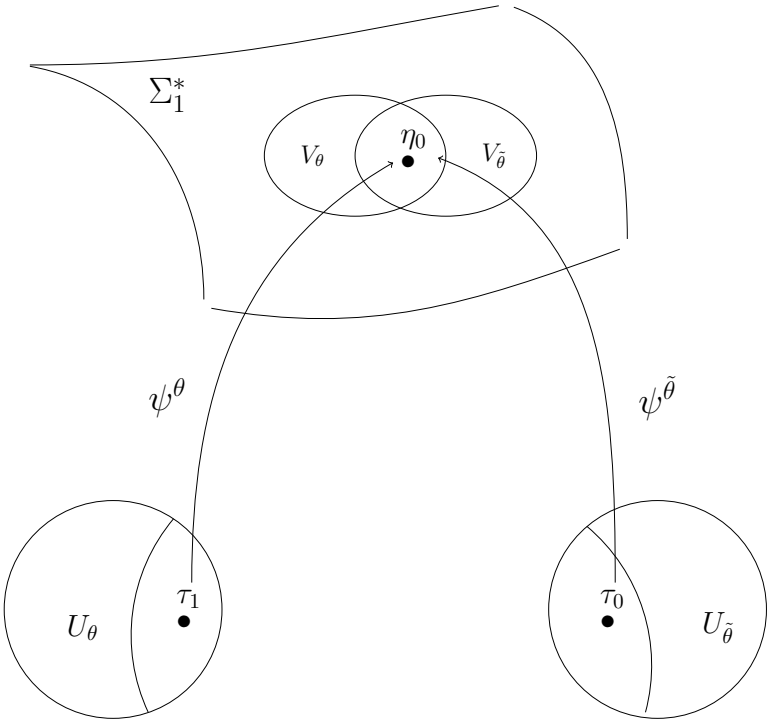

Figure 4.3 – The points η_0, τ_0, τ_1 and the inverse charts ψ^θ and $\psi^{\tilde\theta}$.

We are going to show that there exist an open neighborhood $\tilde{U}$ of $(\tau_1, 0)$ in $(\mathbb{R}^{F_\theta} \times \mathbb{R}^B) \times \mathbb{R}^{N_C}$ and an open neighborhood $\tilde{V}$ of η_0 in $\mathbb{R}^{\mathcal{P}\times V_L}$ such that φ_θ is a C^∞ diffeomorphism from $\tilde{U}$ onto $\tilde{V}$ satisfying

$$\varphi_\theta\Big(\left[(\mathbb{R}^{F_\theta} \times \mathbb{R}^B) \times \{0\}^{N_C}\right] \cap \tilde{U}\Big) = \Sigma_1^* \cap \tilde{V}.$$

Let us first show that φ_θ is a C^∞-diffeomorphism from a neighborhood of $(\tau_1, 0)$ in $(\mathbb{R}^{F_\theta} \times \mathbb{R}^B) \times \mathbb{R}^{N_C}$ onto a neighborhood of η_0 in $\mathbb{R}^{\mathcal{P}\times V_L}$. As shown in Proposition 93, ψ^θ is C^∞ and i is a linear function, so φ_θ is C^∞. Let us prove that the differential $D\varphi_\theta(\tau_1, 0)$ is injective. For all $(\tau, x) \in \left(\mathbb{R}^{F_\theta} \times \mathbb{R}^B\right) \times \mathbb{R}^{N_C}$,

$$D\varphi_\theta(\tau_1, 0) \cdot (\tau, x) = D\psi^\theta(\tau_1) \cdot \tau + i(x).$$

Since $D\psi^\theta(\tau_1)\cdot\tau \in T$, $i(x) \in G$, and T and G are in direct sum, if $D\varphi_\theta(\tau_1, 0)\cdot(\tau, g) = 0$, then we have

$$\begin{cases} D\psi^\theta(\tau_1) \cdot \tau = 0 \\ i(x) = 0. \end{cases}$$

Since as shown in Proposition 93 $D\psi^\theta(\tau_1)$ is injective, and since i is invertible, we have

$$(\tau, x) = (0, 0).$$

Hence, $D\varphi_\theta(\tau_1, 0)$ is injective. Since $\dim(\left(\mathbb{R}^{F_\theta} \times \mathbb{R}^B\right) \times \mathbb{R}^{N_C}) = |F_\theta| + |B| + N_C = |\mathcal{P}|N_L$, the differential $D\varphi_\theta(\tau_1, 0)$ is bijective. Using the inverse function theorem,

there exists an open set $U \subset U_\theta \times \mathbb{R}^{N_C}$ containing $(\tau_1, 0)$, an open set $V \subset \mathbb{R}^{\mathcal{P}\times V_L}$ containing η_0 such that φ_θ is a C^∞-diffeomorphism from U onto V.

We have

$$\varphi_\theta\Big(\left[(\mathbb{R}^{F_\theta} \times \mathbb{R}^B) \times \{0\}^{N_C}\right] \cap U\Big) \quad \subset \quad V_\theta \cap V.$$

In fact, if V is small enough, this inclusion is an equality. We are going to construct open subsets $\tilde{U} \subset U$ and $\tilde{V} \subset V$ so that it is the case. Let us define

$$O = \{\tau \in U_\theta, (\tau, 0) \in U\}.$$

Since U is an open set containing $(\tau_1, 0)$, O is an open set containing $\tau_1 = (\psi^\theta)^{-1}(\eta_0)$. Since, according to Proposition 90, ψ^θ is a homeomorphism, $\psi^\theta(O)$ is an open subset of V_θ so there exists $\epsilon > 0$ such that

$$V_\theta \cap B_\infty(\eta_0, \epsilon) \ \subset \ \psi^\theta(O). \tag{4.A.27}$$

We can now define $\tilde{V} = V \cap B_\infty(\eta_0, \epsilon)$, and $\tilde{U} = \{(\tau, x) \in U, \ \varphi_\theta(\tau, x) \in \tilde{V}\}$, which are open sets such that $(\tau_1, 0) \in \tilde{U}$, $\eta_0 \in \tilde{V}$, and φ_θ is a C^∞-diffeomorphism from $\tilde{U}$ onto $\tilde{V}$. Let us show that

$$\varphi_\theta\Big(\left[(\mathbb{R}^{F_\theta} \times \mathbb{R}^B) \times \{0\}^{N_C}\right] \cap \tilde{U}\Big) = V_\theta \cap \tilde{V}. \tag{4.A.28}$$

The direct inclusion is immediate: if $(\tau, 0) \in \left[(\mathbb{R}^{F_\theta} \times \mathbb{R}^B) \times \{0\}^{N_C}\right] \cap \tilde{U}$, then

$$\varphi_\theta(\tau, 0) = \psi^\theta(\tau) \in V_\theta \cap \tilde{V}.$$

For the reciprocal inclusion, if $\tau \in U_\theta$ is such that $\psi^\theta(\tau) \in V_\theta \cap \tilde{V}$, then by definition of ϵ and $\tilde{V}$, (4.A.27) guarantees, since ψ^θ is injective, that $\tau \in O$. By definition of O, we have $(\tau, 0) \in U$, and since

$$\varphi_\theta(\tau, 0) = \psi^\theta(\tau) \in \tilde{V},$$

this shows $(\tau, 0) \in \tilde{U}$. This shows the reciprocal inclusion, and thus (4.A.28) holds.

Let us now define

$$\begin{array}{rccc} P_\theta : & \mathbb{R}^{F_\theta} \times \mathbb{R}^B \times \mathbb{R}^{N_C} & \longrightarrow & \mathbb{R}^{F_\theta} \times \mathbb{R}^B \\ & (\tau, x) & \longmapsto & \tau \end{array}$$

the restriction to the first component, and let us observe that over $V_\theta \cap \tilde{V}$, we have

$$P_\theta \circ (\varphi_\theta)^{-1} = (\psi^\theta)^{-1}. \tag{4.A.29}$$

Indeed, if $\eta \in V_\theta \cap \tilde{V}$, then by (4.A.28) there exists $\tau \in U_\theta$ such that $(\tau, 0) \in \tilde{U}$ and $\varphi_\theta(\tau, 0) = \eta$. Since $\varphi_\theta(\tau, 0) = \psi^\theta(\tau)$, this shows that $\tau = (\psi^\theta)^{-1}(\eta)$ and thus

$$(\psi^\theta)^{-1}(\eta) = P_\theta(\tau, 0) = P_\theta \circ (\varphi_\theta)^{-1}(\eta).$$

Now recall that $\eta_0 = \psi^{\tilde{\theta}}(\tau_0)$. By continuity of $\psi^{\tilde{\theta}}$, there exists $\epsilon' > 0$ such that $B_\infty(\tau_0, \epsilon') \subset (\psi^{\tilde{\theta}})^{-1}(V_\theta \cap V_{\tilde{\theta}})$ and

$$\psi^{\tilde{\theta}}(B_\infty(\tau_0, \epsilon')) \subset \tilde{V}.$$

For any $\tau \in B_\infty(\tau_0, \epsilon')$, we have $\psi^{\tilde{\theta}}(\tau) \in V_\theta \cap \tilde{V}$ so, as we just proved with (4.A.29), $(\psi^\theta)^{-1} \circ \psi^{\tilde{\theta}}(\tau) = P_\theta \circ (\varphi_\theta)^{-1} \circ \psi^{\tilde{\theta}}(\tau)$. Since the functions $\psi^{\tilde{\theta}}$, $(\varphi_\theta)^{-1}$ and P_θ are all C^∞, we conclude that the transition map $(\psi^\theta)^{-1} \circ \psi^{\tilde{\theta}}$ is C^∞ over $B_\infty(\tau_0, \epsilon')$, for all $\tau_0 \in (\psi^{\tilde{\theta}})^{-1}(V_\theta \cap V_{\tilde{\theta}})$. We conclude that $(\psi^\theta)^{-1} \circ \psi^{\tilde{\theta}}$ is C^∞ over $(\psi^{\tilde{\theta}})^{-1}(V_\theta \cap V_{\tilde{\theta}})$.

We have showed that $(V_\theta, (\psi^\theta)^{-1})_{\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus S}$ is a smooth atlas, and thus that Σ_1^* is a smooth submanifold of $\mathbb{R}^{\mathcal{P} \times V_L}$. As computed in (4.A.7), its dimension is

$$|F_\theta| + |B| = N_0 N_1 + N_1 N_2 + \cdots + N_{L-1} N_L + N_L.$$

□

4.A.4 Conditions of local identifiability

Let us restate (using Definition 75) and prove Theorem 77.

Theorem 95. *For any $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$, the two following statements are equivalent.*

i) θ is locally identifiable from X.

ii) There exists $\epsilon > 0$ such that $B_\infty(\phi(\theta), \epsilon) \cap \Sigma_1^* \cap N(X, \theta) = \{\phi(\theta)\}$.

Proof.

i) $\Rightarrow$ ii) Suppose *i)* is satisfied for some $\epsilon_1 > 0$. We first construct $\epsilon' > 0$ and then consider $\eta \in B_\infty(\phi(\theta), \epsilon') \cap \Sigma_1^* \cap N(X, \theta)$, and we prove that $\eta = \phi(\theta)$. Since $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$ and since, according to Proposition 83, Δ_X is closed, there exists $\epsilon_2 > 0$ such that for any $\tilde{\theta} \in B_\infty(\theta, \epsilon_2)$,

$$\alpha(X, \theta) = \alpha(X, \tilde{\theta}),$$

i.e.

$$A(X, \theta) = A(X, \tilde{\theta}).$$

Consider $\epsilon = \min(\epsilon_1, \epsilon_2)$. Since, according to Proposition 90, $\rho_\theta \circ (\psi^\theta)^{-1}$ is continuous at $\phi(\theta) \in \psi^\theta(U_\theta)$, and since $\rho_\theta \circ (\psi^\theta)^{-1}(\phi(\theta)) = \rho_\theta(\tau_\theta) = \theta$, there exists $\epsilon' > 0$ such that for all $\tau \in U_\theta$,

$$\begin{aligned} \|\psi^\theta(\tau) - \phi(\theta)\|_\infty < \epsilon' \implies \|\rho_\theta(\tau) - \theta\|_\infty &= \|\rho_\theta \circ (\psi^\theta)^{-1}(\psi^\theta(\tau)) - \rho_\theta \circ (\psi^\theta)^{-1}(\phi(\theta))\|_\infty \\ &< \epsilon. \end{aligned} \tag{4.A.30}$$

Since $\phi(\theta) = \psi^\theta(\tau_\theta)$, Proposition 92 guarantees that, modulo a decrease of ϵ', we can assume that

$$B_\infty(\phi(\theta), \epsilon') \cap \Sigma_1^* \subset \psi^\theta(U_\theta). \tag{4.A.31}$$

Now let $\eta \in B_\infty(\phi(\theta), \epsilon') \cap \Sigma_1^* \cap N(X, \theta)$. Let us prove that $\eta = \phi(\theta)$. Using (4.A.31), there exists $\tau \in U_\theta$ such that $\eta = \psi^\theta(\tau)$. Since $\|\phi(\theta) - \eta\|_\infty < \epsilon'$, we have using (4.A.30)

$$\|\rho_\theta(\tau) - \theta\|_\infty < \epsilon. \tag{4.A.32}$$

Since $\epsilon < \epsilon_2$, we have

$$A(X, \theta) = A(X, \rho_\theta(\tau)). \tag{4.A.33}$$

Since $\psi^\theta(\tau) = \eta \in N(X, \theta)$, we have by definition of $N(X, \theta)$ that $\psi^\theta(\tau) - \phi(\theta) \in \operatorname{Ker} A(X, \theta)$, so

$$A(X, \theta) \cdot \psi^\theta(\tau) = A(X, \theta) \cdot \phi(\theta) \tag{4.A.34}$$

Using successively (4.2.3), (4.A.33), (4.A.34) and (4.2.3) again, we have

$$\begin{aligned} f_{\rho_\theta(\tau)}(X) &= A(X, \rho_\theta(\tau)) \cdot \phi(\rho_\theta(\tau)) \\ &= A(X, \theta) \cdot \phi(\rho_\theta(\tau)) \\ &= A(X, \theta) \cdot \phi(\theta) \\ &= f_\theta(X). \end{aligned}$$

Since the hypothesis $i)$ holds for ϵ_1, using (4.A.32) and the fact that $\epsilon < \epsilon_1$, we have

$$\theta \sim \rho_\theta(\tau).$$

We conclude using Proposition 85 that

$$\eta = \phi(\rho_\theta(\tau)) = \phi(\theta),$$

which shows

$$B_\infty(\phi(\theta), \epsilon') \cap \Sigma_1^* \cap N(X, \theta) \ \subset \ \{\phi(\theta)\}.$$

The converse inclusion trivially holds and therefore $ii)$ holds.

$ii) \Rightarrow i)$ Suppose $ii)$ is satisfied for some $\epsilon' > 0$.

We first construct ϵ and prove $i)$ holds. Since $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$, using Proposition 83, there exists $\epsilon_1 > 0$ such that for all $\tilde{\theta} \in B_\infty(\theta, \epsilon_1)$,

$$\alpha(X, \theta) = \alpha(X, \tilde{\theta}),$$

i.e.

$$A(X, \theta) = A(X, \tilde{\theta}). \tag{4.A.35}$$

Since ϕ is continuous, there exists $\epsilon_2 > 0$ such that

$$\|\theta - \tilde{\theta}\|_\infty < \epsilon_2 \quad \Longrightarrow \quad \|\phi(\theta) - \phi(\tilde{\theta})\|_\infty < \epsilon'.$$

Using Proposition 88, there exists $\epsilon_3 > 0$ such that

$$\theta \overset{R}{\sim} \tilde{\theta} \text{ and } \|\theta - \tilde{\theta}\|_\infty < \epsilon_3 \quad \Longrightarrow \quad \theta \sim \tilde{\theta}.$$

Since $\theta \notin S$ and S is closed, there exists $\epsilon_4 > 0$ such that for all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$, if $\|\theta - \tilde{\theta}\|_\infty < \epsilon_4$, then

$$\tilde{\theta} \notin S.$$

Let $\epsilon = \min(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$. Let $\tilde{\theta} \in B_\infty(\theta, \epsilon)$, and suppose

$$f_\theta(X) = f_{\tilde{\theta}}(X).$$

Let us prove that $\theta \sim \tilde{\theta}$. Reformulating the above equality using (4.2.3) for both sides, and using the definition of A given in the beginning of Section 4.4, we have

$$A(X, \theta) \cdot \phi(\theta) = A(X, \tilde{\theta}) \cdot \phi(\tilde{\theta}).$$

Since $\|\theta - \tilde{\theta}\|_\infty < \epsilon \leq \epsilon_1$, we have the equality (4.A.35) and thus

$$A(X, \theta) \cdot \phi(\theta) = A(X, \theta) \cdot \phi(\tilde{\theta}).$$

In other words, $\phi(\tilde{\theta}) - \phi(\theta) \in \operatorname{Ker} A(X, \theta)$. Since $\epsilon < \epsilon_4$, $\phi(\tilde{\theta}) \in \Sigma_1^*$. Since $\epsilon < \epsilon_2$, $\phi(\tilde{\theta}) \in B_\infty(\phi(\theta), \epsilon')$. Summarizing,

$$\phi(\tilde{\theta}) \in B_\infty(\phi(\theta), \epsilon') \cap \Sigma_1^* \cap N(X, \theta),$$

and using the hypothesis ii), we conclude that

$$\phi(\tilde{\theta}) = \phi(\theta).$$

By Proposition 87, we have $\theta \overset{R}{\sim} \tilde{\theta}$, and since $\epsilon < \epsilon_3$, we conclude that

$$\theta \sim \tilde{\theta}.$$

□

We are now going to prove Theorems 78 and 79, which we restate as Theorems 96 and 97 respectively (using Definition 75).

Theorem 96 (Necessary condition)**.** *Let $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$. If C_N is not satisfied, then θ is not locally identifiable from X (thus not globally identifiable).*

Theorem 97 (Sufficient condition)**.** *Let $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$. If C_S is satisfied, then θ is locally identifiable from X.*

To prove the theorems, we need to prove first the following lemmas.

Lemma 98. *Let us denote by $T = \operatorname{Im} D\psi^\theta(\tau_\theta)$ the direction of the tangent plane to Σ_1^* at $\phi(\theta)$. Let us denote by H the intersection $\operatorname{Ker} A(X, \theta) \cap T$. We have*

$$\dim(H) = |F_\theta| + |B| - R_\Gamma. \tag{4.A.36}$$

Proof. Let $\eta \in T$. There exists $h \in \mathbb{R}^{F_\theta} \times \mathbb{R}^B$ such that $\eta = D\psi^\theta(\tau_\theta) \cdot h$. We have the following equivalence:

$$\begin{aligned} \eta \in \operatorname{Ker} A(X,\theta) &\iff A(X,\theta)\cdot\eta \;=\; 0 \\ &\iff A(X,\theta)\circ D\psi^\theta(\tau_\theta)\cdot h \;=\; 0 \\ &\iff \Gamma(X,\theta)\cdot h \;=\; 0 \\ &\iff h \in \operatorname{Ker}\Gamma(X,\theta). \end{aligned}$$

This shows that $D\psi^\theta(\tau_\theta)^{-1}(\operatorname{Ker} A(X,\theta)\cap T) = \operatorname{Ker}\Gamma(X,\theta) \subset \mathbb{R}^{F_\theta}\times\mathbb{R}^B$. Since $D\psi^\theta(\tau_\theta)$ is injective, we thus have

$$\dim(H) = \dim(\operatorname{Ker}\Gamma(X,\theta)) = |F_\theta| + |B| - R_\Gamma.$$

□

Lemma 99. *Let G be a supplementary subspace of $\operatorname{Ker} A(X,\theta)$ such that*

$$H \oplus G = \operatorname{Ker} A(X,\theta). \tag{4.A.37}$$

If $R_\Gamma = R_A$, there exist an open set $\mathcal{O} \subset U_\theta \times G$ containing $(\tau_\theta, 0)$ and an open set $\mathcal{V} \subset \mathbb{R}^{\mathcal{P}\times V_L}$ containing $\phi(\theta)$ such that

$$\begin{array}{rccc} \xi: & \mathcal{O} & \longrightarrow & \mathcal{V} \\ & (\tau, g) & \longmapsto & \psi^\theta(\tau) + g \end{array}$$

is a diffeomorphism from $\mathcal{O}$ onto $\mathcal{V}$.

Proof. Let us first show that

$$T \oplus G = \mathbb{R}^{\mathcal{P}\times V_L}. \tag{4.A.38}$$

Indeed, since $\operatorname{Ker} A(X,\theta) = H \oplus G$ and $T \cap \operatorname{Ker} A(X,\theta) = H$, we have $T \cap G = \{0\}$. We of course have

$$T \oplus G \subset \mathbb{R}^{\mathcal{P}\times V_L}. \tag{4.A.39}$$

Let us show that $\dim(G) = \dim(\mathbb{R}^{\mathcal{P}\times V_L}) - \dim(T)$. First note that we have

$$\dim(\operatorname{Ker} A(X,\theta)) = \dim(\mathbb{R}^{\mathcal{P}\times V_L}) - \operatorname{rank}(A(X,\theta)) = |\mathcal{P}|N_L - R_A. \tag{4.A.40}$$

Using (4.A.37) and (4.A.40), we have

$$\begin{aligned} \dim(G) &= \dim(\operatorname{Ker} A(X,\theta)) - \dim(H) \\ &= |\mathcal{P}|N_L - R_A - \dim(H). \end{aligned}$$

Using (4.A.36) and the hypothesis $R_\Gamma = R_A$ we thus have

$$\begin{aligned} \dim(G) &= |\mathcal{P}|N_L - R_A + R_\Gamma - |F_\theta| - |B| \\ &= |\mathcal{P}|N_L - |F_\theta| - |B| \\ &= |\mathcal{P}|N_L - \dim(T), \end{aligned}$$

where the last equality comes from the injectivity of $D\psi^\theta(\tau_\theta)$, shown in Proposition 93. Together with (4.A.39), this proves (4.A.38).

Let us now consider the function

$$\begin{array}{rccc} \xi: & U_\theta \times G & \longrightarrow & \mathbb{R}^{\mathcal{P}\times V_L} \\ & (\tau, g) & \longmapsto & \psi^\theta(\tau) + g. \end{array}$$

For all $(h, g) \in (\mathbb{R}^{F_\theta} \times \mathbb{R}^B) \times G$, we have

$$D\xi(\tau_\theta, 0) \cdot (h, g) = D\psi^\theta(\tau_\theta)h + g.$$

The differential $D\xi(\tau_\theta, 0)$ is injective. Indeed, if

$$D\xi(\tau_\theta, 0) \cdot (h, g) = 0,$$

then since $D\psi^\theta(\tau_\theta)h \in T$ and $g \in G$, we have

$$\begin{cases} D\psi^\theta(\tau_\theta)h = 0 \\ g = 0, \end{cases}$$

and since $D\psi^\theta(\tau_\theta)$ is injective, $h = 0$ and $D\xi(\tau_\theta, 0)$ is injective. Since, using (4.A.38),

$$\dim(\mathbb{R}^{F_\theta} \times \mathbb{R}^B) + \dim(G) = |\mathcal{P}|N_L,$$

$D\xi(\tau_\theta, 0)$ is bijective.

We can thus apply the inverse function theorem: there exists an open set $\mathcal{O} \subset U_\theta \times G$ containing $(\tau_\theta, 0)$, an open set $\mathcal{V} \subset \mathbb{R}^{\mathcal{P}\times V_L}$ containing $\phi(\theta)$ such that ξ is a diffeomorphism from $\mathcal{O}$ into $\mathcal{V}$. □

We can now prove the theorems.

Proof of Theorem 96. If C_N is not satisfied, then we have $R_\Gamma = R_A < |F_\theta| + |B|$. We can thus apply Lemma 99: there exist an open set $\mathcal{O} \subset U_\theta \times G$ containing $(\tau_\theta, 0)$ and an open set $\mathcal{V} \subset \mathbb{R}^{\mathcal{P}\times V_L}$ containing $\phi(\theta)$ such that

$$\begin{array}{rccc} \xi: & \mathcal{O} & \longrightarrow & \mathcal{V} \\ & (\tau, g) & \longmapsto & \psi^\theta(\tau) + g \end{array}$$

is a diffeomorphism from $\mathcal{O}$ onto $\mathcal{V}$.

Consider $\epsilon > 0$. We define the open set $\tilde{\mathcal{O}} = \mathcal{O} \cap (\psi^\theta)^{-1}(B(\phi(\theta), \epsilon) \times G$ and its image $\tilde{\mathcal{V}} = \xi(\tilde{\mathcal{O}})$.

Using the computation of $\dim(H)$ shown in Lemma 98, we have

$$\dim(H) = |F_\theta| + |B| - R_\Gamma > 0,$$

so there exists a nonzero $w \in H$ such that $\phi(\theta) + w \in \tilde{\mathcal{V}}$. Since ξ induces a diffeomorphism from $\tilde{\mathcal{O}}$ onto $\tilde{\mathcal{V}}$, there exists $(\tau, g) \in \tilde{\mathcal{O}}$ such that

$$\phi(\theta) + w = \psi^\theta(\tau) + g$$

i.e.

$$\psi^\theta(\tau) - \phi(\theta) = w - g. \tag{4.A.41}$$

Let us denote $\tilde{\theta} = \rho_\theta(\tau)$ and let us show that Theorem 95.ii) does not hold. By definition, $\phi(\tilde{\theta}) = \psi^\theta(\tau)$ and since $(\tau, g) \in \tilde{\mathcal{O}}$, $\|\phi(\theta) - \phi(\tilde{\theta})\|_\infty < \epsilon$. Since $H \cap G = \{0\}$, $w \in H$, $g \in G$ and $w \neq 0$, (4.A.41) shows that

$$\phi(\tilde{\theta}) - \phi(\theta) \neq 0.$$

Furthermore, since $w \in H \subset \operatorname{Ker} A(X, \theta)$ and $g \in G \subset \operatorname{Ker} A(X, \theta)$, (4.A.41) shows that

$$\phi(\tilde{\theta}) - \phi(\theta) \in \operatorname{Ker} A(X, \theta),$$

so

$$\phi(\tilde{\theta}) \in N(X, \theta).$$

Summarizing, for any $\epsilon > 0$ there exists $\tilde{\theta} \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \backslash S$ such that $\phi(\tilde{\theta}) \in B_\infty(\phi(\theta), \epsilon) \cap \Sigma_1^* \cap N(X, \theta) \backslash \{\phi(\theta)\}$. The second item of Theorem 95 does not hold. Since it is equivalent, the first item of Theorem 95 does not hold either. In other words, the conclusion of Theorem 96 holds. □

Proof of Theorem 97. Suppose that C_S is satisfied. Using Lemma 98 and using C_S, we obtain

$$\dim(T \cap \operatorname{Ker} A(X, \theta)) = |F_\theta| + |B| - R_\Gamma = 0.$$

We thus have

$$T \cap \operatorname{Ker} A(X, \theta) = \{0\}. \tag{4.A.42}$$

In order to apply Theorem 95, let us show by contradiction that there exists $\epsilon > 0$ such that

$$B_\infty(\phi(\theta), \epsilon) \cap \Sigma_1^* \cap N(X, \theta) = \{\phi(\theta)\}. \tag{4.A.43}$$

More precisely, we suppose that for all $n \in \mathbb{N}^*$, there exists $\phi_n \in N(X, \theta) \cap \Sigma_1^*$ such that $\phi_n \neq \phi(\theta)$ and $\|\phi(\theta) - \phi_n\|_\infty < \frac{1}{n}$ and prove that it leads to $T \cap \operatorname{Ker} A(X, \theta) \neq \{0\}$, which contradicts (4.A.42).

Using Proposition 92, there exists $n_0 \in \mathbb{N}^*$ such that for all $n \geq n_0$, there exists $\tau_n \in U_\theta$ such that $\phi_n = \psi^\theta(\tau_n)$. Since ψ^θ is a homeomorphism and $\psi^\theta(\tau_\theta) = \phi(\theta)$,

$$\phi_n \to \phi(\theta)$$

implies that

$$\tau_n \to \tau_\theta.$$

Moreover, for all $n \geq n_0$, $\tau_n \neq \tau_\theta$.

When n tends to infinity, we can thus write

$$\phi_n - \phi(\theta) = \psi^\theta(\tau_n) - \psi^\theta(\tau_\theta) = D\psi^\theta(\tau_\theta) \cdot (\tau_n - \tau_\theta) + o(\tau_n - \tau_\theta).$$

Let us apply $A(X,\theta)$ and divide by $\|\tau_n - \tau_\theta\|$.

$$\frac{1}{\|\tau_n - \tau_\theta\|} A(X,\theta)\cdot(\phi_n - \phi(\theta)) = A(X,\theta)\circ D\psi^\theta(\tau_\theta)\cdot\left(\frac{\tau_n-\tau_\theta}{\|\tau_n-\tau_\theta\|}\right) + \frac{1}{\|\tau_n-\tau_\theta\|}A(X,\theta)o\left(\tau_n-\tau_\theta\right). \quad (4.A.44)$$

Since $\phi_n \in N(X,\theta)$ for all $n\in\mathbb{N}^*$,

$$\frac{1}{\|\tau_n - \tau_\theta\|} A(X,\theta)\cdot(\phi_n - \phi(\theta)) = 0.$$

Since $\frac{\tau_n-\tau_\theta}{\|\tau_n-\tau_\theta\|}$ belongs to the unit sphere, we can extract a subsequence that converges to a limit h with norm equal to 1. Taking the limit in (4.A.44) according to this subsequence, we obtain

$$0 = A(X,\theta)\circ D\psi^\theta(\tau_\theta)\cdot h,$$

which shows that $D\psi^\theta(\tau_\theta)\cdot h \in \operatorname{Ker} A(X,\theta)$. Since $h\neq 0$ and $D\psi^\theta(\tau_\theta)$ is injective, $D\psi^\theta(\tau_\theta)h \neq 0$ and

$$T\cap \operatorname{Ker} A(X,\theta) \neq \{0\}.$$

This is in contradiction with (4.A.42).

We have proven (4.A.43). We can now conclude thanks to Lemma 95: there exists $\epsilon'>0$ such that for any $\tilde\theta\in\mathbb{R}^E\times\mathbb{R}^B$, if $\|\theta-\tilde\theta\|<\epsilon'$, then

$$f_\theta(X) = f_{\tilde\theta}(X) \quad\Longrightarrow\quad \theta\sim\tilde\theta.$$

□

4.A.5 Checking the conditions numerically

We restate and prove Proposition 81.

Proposition 100. *Let $X\in\mathbb{R}^{n\times V_0}$ and $\theta\in\mathbb{R}^E\times\mathbb{R}^B$. We have*

$$R_A = N_L \operatorname{rank}\left(\alpha(X,\theta)\right).$$

Proof. Let $\eta\in\mathbb{R}^{\mathcal{P}\times V_L}$. We have

$$A(X,\theta)\cdot\eta = \alpha(X,\theta)\eta.$$

If we denote by $\eta^1,\dots,\eta^{N_L}\in\mathbb{R}^{\mathcal{P}}$ the N_L columns of η, the columns of $A(X,\theta)\cdot\eta$ are $\alpha(X,\theta)\eta^1,\dots,\alpha(X,\theta)\eta^{N_L}$. If we consider the matrix η as a family of N_L vectors of $\mathbb{R}^{\mathcal{P}}$ and the matrix $A(X,\theta)\cdot\eta$ as a family of N_L vectors of $\mathbb{R}^n$, the operator $A(X,\theta)$ can then be equivalently described as

$$\begin{array}{rccc} A(X,\theta): & (\mathbb{R}^{\mathcal{P}})^{N_L} & \longrightarrow & (\mathbb{R}^n)^{N_L} \\ & (\eta^1,\dots,\eta^{N_L}) & \longmapsto & (\alpha(X,\theta)\eta^1,\dots,\alpha(X,\theta)\eta^{N_L}). \end{array}$$

The rank of such an operator is $N_L\operatorname{rank}(\alpha(X,\theta))$. □

We restate and prove Proposition 80.

Proposition 101. *Let $X \in \mathbb{R}^{n\times V_0}$ and $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus (S \cup \Delta_X)$. The function*

$$\begin{array}{rcl} U_\theta & \longrightarrow & \mathbb{R}^{n\times V_L} \\ \tau & \longmapsto & f_{\rho_\theta(\tau)}(X) \end{array}$$

is differentiable in a neighborhood of τ_θ, and we denote by $D_\tau f_{\rho_\theta(\tau_\theta)}(X)$ its differential at τ_θ. We have

$$D_\tau f_{\rho_\theta(\tau_\theta)}(X) = \Gamma(X, \theta). \tag{4.A.45}$$

Proof. Using (4.2.3) at $\rho_\theta(\tau)$ and the definition of ψ^θ in (4.3.4), we have

$$f_{\rho_\theta(\tau)}(X) = A(X, \theta) \cdot \psi^\theta(\tau).$$

Taking the differential of

$$\begin{array}{rcl} U_\theta & \longrightarrow & \mathbb{R}^{n\times V_L} \\ \tau & \longmapsto & f_{\rho_\theta(\tau)}(X) \end{array}$$

at τ_θ, and using (4.4.2), we obtain

$$D_\tau f_{\rho_\theta(\tau_\theta)}(X) = A(X, \theta) \circ D\psi^\theta(\tau_\theta) = \Gamma(X, \theta).$$

□

To finish with, the following proposition gives explicit expressions of the coefficients of $\Gamma(X, \theta)$. These expressions are given for the sake of theoretical completeness. Note that when it comes to computing $\Gamma(X, \theta)$ in practice (in order to compute R_Γ), the correct approach is using backpropagation as described in Section 4.5 rather than evaluating the expressions in Proposition 102 which involve sums with very large numbers of summands.

Proposition 102. *If we decompose it in the canonical bases of $\mathbb{R}^{F_\theta} \times \mathbb{R}^B$ and $\mathbb{R}^{[\![1,n]\!]\times V_L}$, $\Gamma(X, \theta)$ is a $(nN_L) \times (|F_\theta| + |B|)$ matrix. For lighter notations, let us drop the dependency in (X, θ) and denote by γ^{i,v_L} the lines of $\Gamma(X, \theta)$, for $i \in [\![1, n]\!]$ and $v_L \in V_L$, which satisfy $(\gamma^{i,v_L})^T \in \mathbb{R}^{F_\theta} \times \mathbb{R}^B$. For any $(i, v_L) \in [\![1, n]\!] \times V_L$, let us express the coefficients of γ^{i,v_L}, i.e. express $\gamma^{i,v_L}_{v_l \to v_{l+1}}$ for any $v_l \to v_{l+1} \in F_\theta$ and express $\gamma^{i,v_L}_{v_l}$ for any $v_l \in B$.*

— *For any* $l \in [\![0, L-1]\!]$ *and any* $(v_l, v_{l+1}) \in V_l \times V_{l+1}$ *such that* $v_l \to v_{l+1} \in F_\theta$,

$$\gamma^{i,v_L}_{v_l \to v_{l+1}} = \sum_{\substack{v_0 \in V_0 \\ \vdots \\ v_{l-1} \in V_{l-1} \\ v_{l+2} \in V_{l+2} \\ \vdots \\ v_{L-1} \in V_{L-1}}} x^i_{v_0} \overline{w}_{v_0 \to v_1} \overline{a}_{v_l}(x^i, \theta) \prod_{\substack{1 \le k \le L-1 \\ k \neq l}} a_{v_k}(x^i, \theta) w_{v_k \to v_{k+1}}$$
$$+ \sum_{l'=1}^{L} \sum_{\substack{v_{l'} \in V_{l'} \\ \vdots \\ v_{l-1} \in V_{l-1} \\ v_{l+2} \in V_{l+2} \\ \vdots \\ v_{L-1} \in V_{L-1}}} b_{v_{l'}} \overline{a}_{v_l}(x^i, \theta) \prod_{\substack{l' \le k \le L-1 \\ k \neq l}} a_{v_k}(x^i, \theta) w_{v_k \to v_{k+1}}, \quad (4.A.46)$$

where $\overline{w}_{v_0 \to v_1} = w_{v_0 \to v_1}$ *and* $\overline{a}_{v_l}(x^i, \theta) = a_{v_l}(x^i, \theta)$ *except when* $l = 0$ *in which case* $\overline{w}_{v_0 \to v_1} = 1$ *and* $\overline{a}_{v_l}(x^i, \theta) = 1$.

— *For any* $l \in [\![1, L]\!]$ *and any* $v_l \in V_l$,

$$\gamma^{i,v_L}_{v_l} = \sum_{\substack{v_{l+1} \in V_{l+1} \\ \vdots \\ v_{L-1} \in V_{L-1}}} \prod_{l \le k \le L-1} a_{v_k}(x^i, \theta) w_{v_k \to v_{k+1}}. \quad (4.A.47)$$

Proof. Let $(i, v_L) \in [\![1, n]\!] \times V_L$.

Let us compute $\gamma^{i,v_L}_{v_l \to v_{l+1}}$, for $l \in [\![0, L-1]\!]$ and $(v_l, v_{l+1}) \in V_l \times V_{l+1}$ such that $v_l \to v_{l+1} \in F_\theta$. $\gamma^{i,v_L}_{v_l \to v_{l+1}}$ is the coefficient corresponding to the line (i, v_L) and the column $(v_l \to v_{l+1})$ of $\Gamma(X, \theta)$. Let us denote by $h^{v_l \to v_{l+1}} \in \mathbb{R}^{F_\theta} \times \mathbb{R}^B$ the vector whose component indexed by $v_l \to v_{l+1}$ is equal to 1 and whose other components are zero. Let us denote by $d^{i,v_L} \in \mathbb{R}^{n \times V_L}$ the element whose component indexed by (i, v_L) is equal to 1 and whose other components are zero. Let us denote by $\langle \cdot, \cdot \rangle_{\mathbb{R}^{n \times V_L}}$ the scalar product of the euclidean space $\mathbb{R}^{n \times V_L}$. We have

$$\begin{aligned}
\gamma^{i,v_L}_{v_l \to v_{l+1}} &= \left\langle d^{i,v_L} \ , \ \Gamma(X, \theta) \cdot h^{v_l \to v_{l+1}} \right\rangle_{\mathbb{R}^{n \times V_L}} \\
&= \left\langle d^{i,v_L} \ , \ A(X, \theta) \circ D\psi^\theta(\tau_\theta) \cdot h^{v_l \to v_{l+1}} \right\rangle_{\mathbb{R}^{n \times V_L}} \\
&= \left\langle d^{i,v_L} \ , \ A(X, \theta) \cdot \frac{\partial \psi^\theta}{\partial \tau_{v_l \to v_{l+1}}}(\tau_\theta) \right\rangle_{\mathbb{R}^{n \times V_L}} \\
&= \left\langle d^{i,v_L} \ , \ \alpha(X, \theta) \frac{\partial \psi^\theta}{\partial \tau_{v_l \to v_{l+1}}}(\tau_\theta) \right\rangle_{\mathbb{R}^{n \times V_L}} \\
&= \left[\alpha(X, \theta) \frac{\partial \psi^\theta}{\partial \tau_{v_l \to v_{l+1}}}(\tau_\theta) \right]_{i, v_L},
\end{aligned}$$

where $\left[\alpha(X, \theta) \frac{\partial \psi^\theta}{\partial \tau_{v_l \to v_{l+1}}}(\tau_\theta)\right]_{i, v_L}$ denotes the coefficient (i, v_L) of the product $\alpha(X, \theta) \frac{\partial \psi^\theta}{\partial \tau_{v_l \to v_{l+1}}}(\tau_\theta)$. Let us remind the dimensions in this product. For the left

factor, recalling the definition given in the beginning of Section 4.A.4, we have $\alpha(X,\theta) \in \mathbb{R}^{n\times\mathcal{P}}$. Concerning the right factor, since for any $\tau \in U_\theta$, we have $\psi^\theta(\tau) \in \mathbb{R}^{\mathcal{P}\times V_L}$, the partial derivative satisfies $\frac{\partial\psi^\theta}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta) \in \mathbb{R}^{\mathcal{P}\times V_L}$. Hence, the dimension of the product is

$$\alpha(X,\theta)\frac{\partial\psi^\theta}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta) \in \mathbb{R}^{n\times V_L}.$$

To obtain the coefficient (i, v_L) of this product, we keep the i^{th} line of the left factor, which is equal to $\alpha(x^i,\theta)$, and the column v_L of the right factor, which is equal to $\frac{\partial\psi^\theta_{v_L}}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta)$. We thus have

$$\left[\alpha(X,\theta)\frac{\partial\psi^\theta}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta)\right]_{i,v_L} = \alpha(x^i,\theta)\frac{\partial\psi^\theta_{v_L}}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta) \;=\; \sum_{p\in\mathcal{P}}\alpha_p(x^i,\theta)\frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta).$$

Let $p \in \mathcal{P}$. If $p = (v_0,\dots,v_L) \in \mathcal{P}_0$, looking at the case 1 in the proof of Proposition 93, we have

$$\frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta) = \mathbf{1}_{\{v_l\to v_{l+1}\in p\}}\prod_{\substack{k\in[\![0,L-1]\!]\\k\neq l}} w_{v_k\to v_{k+1}}.$$

Recalling the definition of $\alpha_p(x^i,\theta)$ in the case $p \in \mathcal{P}_0$, given in (4.2.2), we also have

$$\alpha_p(x^i,\theta) = x^i_{v_0}\prod_{k=1}^{L-1} a_{v_k}(x^i,\theta),$$

and thus

$$\alpha_p(x^i,\theta)\frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta) = \mathbf{1}_{\{v_l\to v_{l+1}\in p\}}\; x^i_{v_0}\prod_{k=1}^{L-1} a_{v_k}(x^i,\theta)\prod_{\substack{k\in[\![0,L-1]\!]\\k\neq l}} w_{v_k\to v_{k+1}}. \tag{4.A.48}$$

Now if $p = (v_{l'},\dots,v_L) \in \mathcal{P}_{l'}$, for $l' \in [\![1,\dots,L-1\}$, looking at the case 2 in the proof of Proposition 93, we have

$$\frac{\partial\psi^\theta_{p,v_L}}{\partial\tau_{v_l\to v_{l+1}}}(\tau_\theta) = \mathbf{1}_{\{v_l\to v_{l+1}\in p\}}\; b_{v_{l'}}\prod_{\substack{k\in[\![l',L-1]\!]\\k\neq l}} w_{v_k\to v_{k+1}}.$$

Recalling the definition of $\alpha_p(x^i,\theta)$ in the case $p \in \mathcal{P}_{l'}$, given in (4.2.2), we also have

$$\alpha_p(x^i,\theta) = \prod_{k=l'}^{L-1} a_{v_k}(x^i,\theta),$$

and thus

$$\alpha_p(x^i,\theta)\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v_l\to v_{l+1}}}(\tau_\theta) = \mathbf{1}_{\{v_l\to v_{l+1}\in p\}}\ b_{v_{l'}}\prod_{k=l'}^{L-1} a_{v_k}(x^i,\theta)\prod_{\substack{k\in[\![l',L-1]\!]\\ k\neq l}} w_{v_k\to v_{k+1}}. \tag{4.A.49}$$

Finally, if $p=\beta$, looking at the case 3 in the proof of Proposition 93, we have

$$\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v_l\to v_{l+1}}}(\tau_\theta) = 0,$$

and thus

$$\alpha_p(x^i,\theta)\frac{\partial \psi^\theta_{p,v_L}}{\partial \tau_{v_l\to v_{l+1}}}(\tau_\theta) = 0. \tag{4.A.50}$$

Assembling (4.A.48), (4.A.49) and (4.A.50), we can sum over all $p\in\mathcal{P}$, and obtain

$$\begin{aligned}\gamma^{i,v_L}_{v_{l+1}\to v_l} = &\sum_{\substack{p\in\mathcal{P}_0\\ p=(v_0,\dots,v_{L-1})}} \mathbf{1}_{\{v_l\to v_{l+1}\in p\}}\ x^i_{v_0}\prod_{k=1}^{L-1} a_{v_k}(x^i,\theta)\prod_{\substack{k\in[\![0,L-1]\!]\\ k\neq l}} w_{v_k\to v_{k+1}}\\ &+\sum_{l'=1}^{L}\sum_{\substack{p\in\mathcal{P}_{l'}\\ p=(v_{l'},\dots,v_{L-1})}} \mathbf{1}_{\{v_l\to v_{l+1}\in p\}}\ b_{v_{l'}}\prod_{k=l'}^{L-1} a_{v_k}(x^i,\theta)\prod_{\substack{k\in[\![l',L-1]\!]\\ k\neq l}} w_{v_k\to v_{k+1}}\end{aligned}$$

which can be reformulated, getting rid of the zero sums when $v_l\to v_{l+1}\notin p$, as

$$\begin{aligned}\gamma^{i,v_L}_{v_{l+1}\to v_l} = &\sum_{\substack{v_0\in V_0\\ \vdots\\ v_{l-1}\in V_{l-1}\\ v_{l+2}\in V_{l+2}\\ \vdots\\ v_{L-1}\in V_{L-1}}} x^i_{v_0}\overline{w}_{v_0\to v_1}a_{v_l}(x^i,\theta)\prod_{\substack{k\in[\![1,L-1]\!]\\ k\neq l}} a_{v_k}(x^i,\theta)w_{v_k\to v_{k+1}}\\ &+\sum_{l'=1}^{L}\sum_{\substack{v_{l'}\in V_{l'}\\ \vdots\\ v_{l-1}\in V_{l-1}\\ v_{l+2}\in V_{l+2}\\ \vdots\\ v_{L-1}\in V_{L-1}}} a_{v_l}(x^i,\theta)b_{v'_l}\prod_{\substack{k\in[\![l',L-1]\!]\\ k\neq l}} a_{v_k}(x^i,\theta)w_{v_k\to v_{k+1}},\end{aligned}$$

which shows (4.A.46).

The proof of (4.A.47) is similar to the one of (4.A.46). □

Chapter 5

Geometry induced implicit regularization: theoretical insights and numerical evidence.

This chapter is a joint work with François Bachoc and François Malgouyres. It will soon be submitted for publication.

Contents

5.1 Introduction

On the importance of local complexity measures for neural networks Deep neural networks has a huge impact on many practical aspects of our lives. Training a neural network requires optimizing a non-convex function, in a large dimensional space. Surprisingly, in many cases, although the number of parameters defining the neural network exceeds by far the amount of training data, the learned neural network generalizes and performs well with unseen data [195]. This is surprising because in this setting the set of global minimizers is large [51, 106] and contains elements that generalize poorly [192, 134].

Not surprisingly, the good generalization behavior is not explained by the classical statistical learning theory (e.g., [6, 83]) that considers the worst possible parameters in the parameter set. For instance, the Vapnik-Chervonenkis dimension of feedforward neural networks of depth L, with W parameters, with the ReLU activation function is[1] $\widetilde{O}(LW)$ [16, 20, 113], leading to an upper bound on the generalization gap of order[1] $\widetilde{O}(\sqrt{\frac{LW}{n}})$, where n is the sample size. This worst-case analysis is not accurate enough to explain the success of deep learning, when $W \gg n$.

This leads to the conclusion that a global analysis, that applies to all global minima and the worst possible neural network that fits the data, will not permit to explain the success of deep learning. A local analysis is needed.

Despite tremendous research efforts in this direction (see, e.g., [83] and references below) an explanation for the good generalization behavior in deep learning is still lacking. The attempts of explanation suggest that stochastic algorithms discover 'good minima' . These are minima having special properties that authors would like to model using local complexity measures that are pivotal in the mathematical explanation. Authors aim to establish that stochastic algorithms prioritize outputs (parameterizations at convergence) with low local complexity and to demonstrate that low local complexity explains the good generalization to unseen data [18, 44, 34, 99]. This is sometimes also expressed as some form of implicit regularization [92, 22, 134].

In this spirit, many authors contend that the excellent generalization behavior can be attributed to the fulfillment of conditions regarding the flatness of the landscape in the proximity of the algorithm's output [99, 63, 42, 90]. This is known however not to fully capture the good generalization phenomenon [53]. Other studies explain the good generalization performances by constraints involving norms of the neural network weights [18, 137, 77, 15]. Despite being supported by partial arguments, none of the aforementioned local complexity measures fully explain the experimentally observed behaviors.

This is in sharp contrast with linear networks for which implicit regularization is better understood. The consensus is that implicit regularization constrains the rank of the prediction matrix, the matrix obtained when multiplying all the factors

1. The notation $\widetilde{O}(\cdot)$ ignores logarithmic factors.

of the linear network [9, 155, 167, 73, 74, 2].

Local dimensions of the image and pre-image sets This article investigates properties and computational aspects of local complexity measures of deep ReLU neural networks, recently introduced in [82]. The considered complexity measures relate to the local geometry of the *image set* as defined by $\{f_\theta(X) \mid \theta \text{ varies}\}$ and the *pre-image set* $\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$, where $f_\theta(X)$ is the prediction, for an input sample X, made by the neural network of parameter θ. When the differential $\mathbf{D}f_\theta(X)$ of $\theta \mapsto f_\theta(X)$ is appropriately defined, these concepts of complexity are associated with the local dimension of these sets, see Corollary 105, and related to the rank of the aforementioned differential, denoted $\operatorname{rank}(\mathbf{D}f_\theta(X))$ and called *batch functional dimension* in [82]. Notice that, before [82], the batch functional dimension already appeared in an identifiability condition in [28].

Main contributions and organization of the paper

— In Theorem 103 (Section 5.3), up to a negligible set, we decompose the parameter space as a finite union of open sets. On each set, the *batch functional dimension*

$$\operatorname{rank}(\mathbf{D}f_\theta(X))$$

is well defined and constant. The construction of the sets shows that almost everywhere, the activation pattern (defined in Section 5.2) determines the batch functional dimension. We also establish in Proposition 104 (Section 5.3) that the batch functional dimension is invariant under classical invariants of ReLU neural networks, positive rescaling, and neuron permutation as defined in Section 5.2. We also provide examples in Section 5.3.

— In Section 5.4, we illustrate the consequences of the statements of Section 5.3 when learning a deep ReLU network. In particular, we explain the links between the batch functional dimension, and the local dimensions of the image and pre-image sets, see Corollary 105 and Figure 5.1. We also illustrate in this figure how the described geometry impacts the iterates trajectory for small learning rates, see Figure 5.1, and describe the geometry induced implicit regularization (Section 5.4).

— In Section 5.5, we study the *computable full functional dimension*[1] defined by

$$r^*(\theta) = \max_X \operatorname{rank}(\mathbf{D}f_\theta(X)).$$

The first result of the section states that the *achievable activation patterns* for θ determine $r^*(\theta)$, see Theorem 108. It also shows that when more activation patterns can be achieved, r^* increases. As for the batch functional dimension, we establish that the computable full functional dimension is invariant under positive rescalings and neuron permutations. We finish the section with a

1. As its name indicates, the *computable full functional dimension* is a variant of the *full functional dimension* defined in [82], that we can compute.

connection between the computable full functional dimension and the fat-shattering dimension of neural networks.

— In Section 5.6 we provide the details on the practical computation of $\mathrm{rank}(\mathbf{D}f_\theta(X))$, for given X and θ. We also establish in Theorem 113 that, for a given θ, a random X of sufficient size can be used to compute $r^*(\theta)$. Indeed, we upper bound the probability of not reaching $r^*(\theta)$, as a vanishing function of the number of columns of X. The upper bound depends on two natural quantities p and $n^*(\theta)$ (see Theorem 113 for details).

— Finally, we provide experiments on the MNIST dataset in Section 5.7. In Section 5.7.2, we analyze the behavior of the local complexity measures when the width of the network increases. We also describe their behavior during the learning phase. We also show how they behave when the distribution of (X, Y) is artificially complexified.

All the proofs are in the Appendices and the codes are available at [27].

Related works To the best of our knowledge, the functional dimensions of deep ReLU neural networks has only been explicitly studied by [82, 80]. The article [82] is very rich and it is difficult to summarize it in a few lines[2]. The authors establish sufficient conditions guaranteeing that $\theta \longmapsto f_\theta(X)$ is differentiable. The conditions are comparable but weaker than the one presented here. The benefit of the difference is that our conditions guarantee the value of the batch functional dimension, allowing us to make the connection between the activation patterns and the batch functional dimension. They define and provide examples to illustrate that the batch functional dimension and the computable full functional dimension vary. They also prove that for all narrowing architectures[3], the *functional dimension* as defined by $\max_\theta \max_X \mathrm{rank}(\mathbf{D}f_\theta(X))$ reaches its upper-bound $W - W'$ where W' is the number of positive rescalings. They finish their article with several examples showing that the global structure of the *pre-image set* $\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$ can vary in several regards. Grigsby, Lindsey, and Rolnick prove that when the input size lower-bounds the other widths there exist parameters for which the batch functional dimension reaches the upper-bound $W - W'$. They also numerically estimate, for several neural network architectures, the size of the sets of parameters that reach this upper bound.

Geometric properties of the pre-image set of a global minimizer have been studied in [51]. Topological properties of a variant of the image set included in function spaces, $\{f_\theta \mid \theta \text{ varies}\}$, have been established in [144].

There are many articles devoted to the identifiability of neural networks [145, 39, 159, 179, 28, 26]. For a given θ, they study conditions guaranteeing that the pre-image set of[4] $f_\theta(X)$ coincides with the set obtained when considering all the

2. A weakness of [82] is that it considers neural networks whose last layer undergoes a ReLU activation.

3. Narrowing architectures are such that widths decrease.

4. In these articles X sometimes contains infinitely many examples, in which case we let $f_\theta(X)$ denote the function f_θ restricted to X.

positive rescalings of θ. Of particular interest in our context, the authors of [28] establish that the condition $\operatorname{rank}(\mathbf{D}f_\theta(X)) = W - W'$ is sufficient to guarantee local identifiability. The same condition is also involved in a necessary condition of local identifiability.

Other local complexity measures, not related to the geometry of neural networks, have been considered. In [125, 150, 86], the authors measure complexity using the number of achievable activation patterns.

The objects studied in this article are also related to the properties of the landscape of the empirical risk. Studies of these properties for instance permit to guarantee that first-order algorithms find a global minimizer [176, 138, 162, 55], focus on the shape at the bottom of the empirical risk [72, 163, 84] and on flatness [99, 63, 42, 90].

The local properties studied in the present article also have an impact on the iterates trajectory and therefore the biases induced by the optimization as studied in [18, 44, 34, 99].

Finally, [11, 182] establish generalization bounds of compressed neural networks. This might provide hints for the construction of upper-bounds of the generalization gap based on the local geometric complexity measures considered in this article.

5.2 ReLU networks and notations

ReLU network architecture Let us introduce our notations for deep fully-connected ReLU networks. In this paper, a network is a graph (E, V) of the following form.

— V is a set of neurons, which is divided into $L+1$ layers, with $L \geq 2$: $V = \bigcup_{\ell=0}^{L} V_\ell$. Throughout the paper, for $a, b \in \mathbb{N}$, $a \leq b$, $[\![a, b]\!]$ is the set of consecutive integers $\{a, a+1, \dots, b\}$. The layer V_0 is the input layer, V_L is the output layer and the layers V_ℓ with $1 \leq \ell \leq L-1$ are the hidden layers. Using the notation $|C|$ for the cardinal of a finite set C, we denote, for all $\ell \in [\![0, L]\!]$, $N_\ell = |V_\ell|$ the size of the layer V_ℓ.

— E is the set of all oriented edges $v \to v'$ between neurons in consecutive layers, that is

$$E = \{v' \to v \mid \ v' \in V_{\ell-1}, v \in V_\ell, \text{for } \ell \in [\![1, L]\!]\}.$$

A network is parameterized by weights and biases, gathered in its parameterization θ, with

$$\theta = ((w_{v' \to v})_{v' \to v \in E}, (b_v)_{v \in B}) \quad \in \mathbb{R}^E \times \mathbb{R}^B,$$

where $B = \bigcup_{\ell=1}^{L} V_\ell$. We have $W = |E| + |B|$

The activation function, denoted σ, is always ReLU: for any $p \in \mathbb{N}^*$ and any vector
$x = (x_1, \dots, x_p)^T \in \mathbb{R}^p$, it is defined as $\sigma(x) = (\max(x_1, 0), \dots, \max(x_p, 0))^T$. Here and in the sequel, the symbol $\mathbb{N}^*$ denotes the set of natural numbers without 0.

ReLU network prediction For a given θ, we define recursively $f_\theta^\ell : \mathbb{R}^{V_0} \to \mathbb{R}^{V_\ell}$, for $\ell \in [\![0, L]\!]$ and $x \in \mathbb{R}^{V_0}$, by

$$\begin{cases} (f_\theta^0(x))_v = x_v, & \text{for } v \in V_0, \\ (f_\theta^\ell(x))_v = \sigma\left(\sum_{v' \in V_{\ell-1}} w_{v' \to v} (f_\theta^{\ell-1}(x))_{v'} + b_v\right), & \text{for } v \in V_\ell, \text{ when } \ell \in [\![1, L-1]\!], \\ (f_\theta^L(x))_v = \sum_{v' \in V_{L-1}} w_{v' \to v} (f_\theta^{L-1}(x))_{v'} + b_v, & \text{for } v \in V_L. \end{cases} \tag{5.2.1}$$

We define the function $f_\theta : \mathbb{R}^{V_0} \to \mathbb{R}^{V_L}$ implemented by the network of parameter θ as $f_\theta = f_\theta^L$. We call it the prediction or the inference. For all $n \in \mathbb{N}^*$, we usually write an input set of size n for a neural network as a matrix $X = (x^{(i)})_{i \in [\![1,n]\!]} \in \mathbb{R}^{N_0 \times n}$, where $x^{(i)}$ is the i-th column of X and the i-th input of the network. We also allow to write f_θ as operating on an input set X, that is we may write $f_\theta : \mathbb{R}^{N_0 \times n} \to \mathbb{R}^{N_L \times n}$ and we define $f_\theta(X)$ as the matrix gathering the outputs $(f_\theta(x^{(i)}))_{i \in [\![1,n]\!]}$.

Among other quantities, we study in this article the set

$$\{f_\theta(X) \mid \theta \in \mathbb{R}^E \times \mathbb{R}^B\},$$

for $X \in \mathbb{R}^{N_0 \times n}$ fixed, which we call an *image set*. When it is differentiable at θ, we denote by $\mathbf{D} f_\theta(X)$ the differential of the mapping

$$\begin{array}{ccc} \mathbb{R}^E \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^{N_L \times n} \\ \theta' & \longmapsto & f_{\theta'}(X) \end{array}$$

at the point θ. We recall that the differential at θ is the linear map

$$\mathbf{D} f_\theta(X) : \mathbb{R}^E \times \mathbb{R}^B \longrightarrow \mathbb{R}^{N_L \times n} \tag{5.2.2}$$

such that, for $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$ in a neighborhood of zero,

$$f_{\theta+\theta'}(X) = f_\theta(X) + \mathbf{D} f_\theta(X)(\theta') + o(\|\theta'\|). \tag{5.2.3}$$

Invariance by positive rescaling and neuron permutations Consider two parameters $\theta, \widetilde{\theta} \in \mathbb{R}^{E \times B}$, with $\widetilde{\theta} = \left((\widetilde{w}_{v \to v'})_{v \to v' \in E}, (\widetilde{b}_v)_{v \in B}\right)$. Then we say that θ and $\widetilde{\theta}$ are equivalent modulo positive rescaling, and we write $\theta \sim_s \widetilde{\theta}$, when the following holds. There are $(\lambda_v)_{v \in V_0 \cup \cdots \cup V_L} \in (0, \infty)^{V_0 \cup \cdots \cup V_L}$ such that $\lambda_v = 1$ for $v \in V_0 \cup V_L$ and for $\ell \in [\![1, L]\!]$, $v \in V_{\ell-1}$, $v' \in V_\ell$,

$$w_{v \to v'} = \frac{\lambda_{v'}}{\lambda_v} \widetilde{w}_{v \to v'}, \tag{5.2.4}$$

$$b_{v'} = \lambda_{v'} \widetilde{b}_{v'}. \tag{5.2.5}$$

Then it is a well-known property of ReLU networks [26, 28, 135, 178, 179, 194] that if $\theta \sim_s \widetilde{\theta}$ then $f_\theta = f_{\widetilde{\theta}}$, that is, positive rescalings are a invariant transformations of the network parameterization.

Another classic invariant consists in swapping neurons, and their corresponding weights, within each hidden layer. If $\widetilde{\theta}$ stands for the permuted weights, we denote the corresponding equivalence relation $\widetilde{\theta} \sim_p \theta$. Again, when $\widetilde{\theta} \sim_p \theta$, we have $f_{\widetilde{\theta}} = f_\theta$.

We say that $\widetilde{\theta} \sim \theta$ if there exists θ' such that $\widetilde{\theta} \sim_p \theta'$ and $\theta' \sim_s \theta$. Again, if $\widetilde{\theta} \sim \theta$, then $f_\theta = f_{\widetilde{\theta}}$.

Activation patterns For any $\ell \in [\![1, L-1]\!]$, $v \in V_\ell$, $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and $x \in \mathbb{R}^{N_0}$, we define the activation indicator at neuron v by

$$a_v(x,\theta) = \begin{cases} 1 & \text{if } \sum_{v' \in V_{\ell-1}} w_{v' \to v} (f_\theta^{\ell-1}(x))_{v'} + b_v \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Using (5.2.1), we have for the ReLU activation function σ, any $\ell \in [\![1, L-1]\!]$ and $v \in V_\ell$,

$$(f_\theta^\ell(x))_v = a_v(x,\theta) \left(\sum_{v' \in V_{\ell-1}} w_{v' \to v} (f_\theta^{\ell-1}(x))_{v'} + b_v \right). \tag{5.2.6}$$

We then define the *activation pattern* as the mapping

$$\begin{array}{rcl} a : \mathbb{R}^{N_0} \times \left(\mathbb{R}^E \times \mathbb{R}^B \right) & \longrightarrow & \{0,1\}^{N_1 + \cdots + N_{L-1}} \\ (x,\theta) & \longmapsto & (a_v(x,\theta))_{v \in V_1 \cup \cdots \cup V_{L-1}}. \end{array}$$

For $X \in \mathbb{R}^{N_0 \times n}$ as considered above, we let $a(X,\theta) \in \{0,1\}^{(N_1+\cdots+N_{L-1}) \times n}$ be defined by, for $i \in [\![1,n]\!]$ and $v \in V_1 \cup \cdots \cup V_{L-1}$, $a_{v,i}(X,\theta) = a_v(x^{(i)},\theta)$. We also call *activation patterns* the elements of $\{0,1\}^{N_1+\cdots+N_{L-1}}$ or $\{0,1\}^{(N_1+\cdots+N_{L-1}) \times n}$.

Further notation We use the notation $\mathrm{rank}(\cdot)$ for the rank of linear maps and matrices. The determinant of a square matrix M is denoted $\det(M)$. If the matrix $M \in \mathbb{R}^{a \times b}$ for $a, b \in \mathbb{N}^*$, then for $i \in [\![1,a]\!]$, we write $M_{i,:}$ for the row i of M.

All vector spaces are endowed with the standard Euclidean topology. For two sets $C \subset D$ of a topological space, we denote $\mathrm{Int}(C)$ the topological interior of C, ∂C its boundary and $C^c = D \setminus C$ the complement of C in D. When D is the whole space containing C, we just refer to this last set as 'the complement of C'. For all Euclidean space V, all element $x \in V$, and all real number $r \geq 0$, the open Euclidean ball of radius r centered at x is denoted by $B(x,r)$.

5.3 Rank properties

In this section, we give the key technical theorem on which the remaining of the article relies. We then illustrate the theorem with examples showing the diversity of situations that might occur.

For $n \in \mathbb{N}^*$, the function

$$\begin{array}{rcl} \mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B) & \longrightarrow & \{0,1\}^{(N_1+\cdots+N_{L-1}) \times n} \\ (X,\theta) & \longmapsto & a(X,\theta) \end{array}$$

takes a finite set of values, that we write $\Delta_1, \ldots, \Delta_q$. Let us write, for $j \in [\![1,q]\!]$,

$$\widetilde{\mathcal{O}}_j^n = \mathrm{Int}\left\{ (X,\theta) \in \mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B) \mid a(X,\theta) = \Delta_j \right\}, \tag{5.3.1}$$

and let us only keep the non-empty $\widetilde{\mathcal{O}}_j^n$. If $m_n \in [\![1,q]\!]$ is the number of such non-empty sets, up to a re-ordering, we can assume that we keep $\widetilde{\mathcal{O}}_1^n, \ldots, \widetilde{\mathcal{O}}_{m_n}^n$. As will

be formally established in Lemma 115, on the sets $\widetilde{\mathcal{O}}_j^n$, the function $\theta \mapsto f_\theta(X)$ is differentiable. We can therefore define, for $n \in \mathbb{N}^*$ and $j \in [\![1, m_n]\!]$,

$$r_j^n = \max_{(X,\theta)\in\widetilde{\mathcal{O}}_j^n} \operatorname{rank}(\mathbf{D}f_\theta(X)). \tag{5.3.2}$$

We then define the subset of $\widetilde{\mathcal{O}}_j^n$ on which the rank is maximal. For $n \in \mathbb{N}^*$ and $j \in [\![1, m_n]\!]$,

$$\mathcal{O}_j^n = \{(X,\theta) \in \widetilde{\mathcal{O}}_j^n \mid \operatorname{rank}(\mathbf{D}f_\theta(X)) = r_j^n\}. \tag{5.3.3}$$

Similarly, for $n \in \mathbb{N}^*$ and $X \in \mathbb{R}^{N_0\times n}$, the function $\theta \mapsto a(X,\theta)$ takes a finite number of values $\Delta_1^X, \dots, \Delta_{q^X}^X$, and we define, for $j \in [\![1, q^X]\!]$,

$$\widetilde{\mathcal{U}}_j^X = \operatorname{Int}\{\theta \in \mathbb{R}^E \times \mathbb{R}^B \mid a(X,\theta) = \Delta_j^X\}. \tag{5.3.4}$$

Similarly, we keep only the nonempty such sets, and if $p_X \in [\![1, q^X]\!]$ is the number of such sets, we can assume up to a re-ordering that we keep $\widetilde{\mathcal{U}}_1^X, \dots, \widetilde{\mathcal{U}}_{p_X}^X$. Again, as we will establish in Lemma 115, on the sets $\widetilde{\mathcal{U}}_j^X$, the function $\theta \mapsto f_\theta(X)$ is differentiable. We can therefore define, for $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{N_0\times n}$ and $j \in [\![1, p_X]\!]$,

$$r_j^X = \max_{\theta\in\widetilde{\mathcal{U}}_j^X} \operatorname{rank}(\mathbf{D}f_\theta(X)). \tag{5.3.5}$$

We finally define the subset of $\widetilde{\mathcal{U}}_j^X$ on which the rank is maximal. For $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{N_0\times n}$ and $j \in [\![1, p_X]\!]$,

$$\mathcal{U}_j^X = \{\theta \in \widetilde{\mathcal{U}}_j^X \mid \operatorname{rank}(\mathbf{D}f_\theta(X)) = r_j^X\}. \tag{5.3.6}$$

The following theorem is composed of two parts. In the first one, we study $(X,\theta) \mapsto f_\theta(X)$ over $\mathbb{R}^{N_0\times n} \times (\mathbb{R}^E \times \mathbb{R}^B)$, and we provide properties of the sets $\mathcal{O}_1^n, \dots, \mathcal{O}_{m_n}^n$. In the second one, we study $\theta \mapsto f_\theta(X)$ over $\mathbb{R}^E \times \mathbb{R}^B$, for X fixed, and we provide properties of the sets $\mathcal{U}_1^X, \dots, \mathcal{U}_{p_X}^X$.

Note that for both parts (i) and (ii), Items 1, 2 and 3 hold trivially by definition, while Items 4, 5 and 6 require detailed proofs.

Theorem 103. *Consider any deep fully-connected ReLU network (E, V).*

(i) For all $n \in \mathbb{N}^*$, by definition,

- *the sets $\mathcal{O}_1^n, \dots, \mathcal{O}_{m_n}^n$ are non-empty and disjoint,*
- *for all $j \in [\![1, m_n]\!]$, the function $(X,\theta) \mapsto a(X,\theta)$ is constant on $\mathcal{O}_j^n$ and takes m_n distinct values on $\cup_{j=1}^{m_n}\mathcal{O}_j^n$;*
- *for all $j \in [\![1, m_n]\!]$, $(X,\theta) \mapsto \operatorname{rank}(\mathbf{D}f_\theta(X))$ is constant on $\mathcal{O}_j^n$ and equal to r_j^n.*

Furthermore,

- *the sets $\mathcal{O}_1^n, \dots, \mathcal{O}_{m_n}^n$ are open,*
- *$\left(\cup_{j=1}^{m_n}\mathcal{O}_j^n\right)^c$ is a closed set with Lebesgue measure zero;*
- *for all $j \in [\![1, m_n]\!]$, $(X,\theta) \mapsto f_\theta(X)$ is a polynomial function with degree less than or equal to $L+1$ on $\mathcal{O}_j^n$.*

(ii) For all $n \in \mathbb{N}^*$, for all $X \in \mathbb{R}^{N_0 \times n}$, by definition,
- *the sets $\mathcal{U}_1^X, \dots, \mathcal{U}_{p_X}^X$ are non-empty and disjoint,*
- *for all $j \in [\![1, p_X]\!]$, the function $\theta \mapsto a(X, \theta)$ is constant on each $\mathcal{U}_j^X$ and takes p_X distinct values on $\cup_{j=1}^{p_X} \mathcal{U}_j^X$;*
- *for all $j \in [\![1, p_X]\!]$, $\theta \mapsto \operatorname{rank}(\mathbf{D} f_\theta(X))$ is constant on $\mathcal{U}_j^X$ and equal to r_j^X.*

Furthermore,
- *the sets $\mathcal{U}_1^X, \dots, \mathcal{U}_{p_X}^X$ are open,*
- *$\left(\cup_{j=1}^{p_X} \mathcal{U}_j^X\right)^c$ is a closed set with Lebesgue measure zero;*
- *for all $j \in [\![1, p_X]\!]$, $\theta \mapsto f_\theta(X)$ is a polynomial function with degree less than or equal to L on $\mathcal{U}_j^X$.*

The proof of the theorem is in Appendix 5.A.1.

This theorem formalizes that the sets $(\mathcal{O}_j^n)_{j \in [\![1, m_n]\!]}$ (resp $(\mathcal{U}_j^X)_{j \in [\![1, p_X]\!]}$) almost cover the spaces $\mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B)$ (resp. $\mathbb{R}^E \times \mathbb{R}^B$). Moreover, on each set $\mathcal{O}_j^n$ (resp. $\mathcal{U}_j^X$) the activation pattern is constant, and the function $(X, \theta) \mapsto f_\theta(X)$ (resp. $\theta \mapsto f_\theta(X)$) is polynomial. We only state that it is a polynomial but we would like to emphasize here that the structure of the polynomials is very particular. For instance, every variable appears with a degree at most one, and all monomials have the same degree. A more complete description of the polynomial structure is, for instance, given in [28, 179]. Also, looking at the definition of $\widetilde{\mathcal{O}}_j^n$ (resp $\widetilde{\mathcal{U}}_j^X$) and $\mathcal{O}_j^n$ (resp $\mathcal{U}_j^X$), using that $\left(\cup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c$ (resp $\left(\cup_{j=1}^{p_X} \mathcal{U}_j^X\right)^c$) is a closed set with Lebesgue measure zero, we find that,

$$\mathcal{O}_j^n \text{ is open and dense in } \widetilde{\mathcal{O}}_j^n \qquad \text{and} \qquad \mathcal{U}_j^X \text{ is open and dense in } \widetilde{\mathcal{U}}_j^X.$$

In other words, modulo negligible sets, the activation pattern determines $\operatorname{rank}(\mathbf{D} f_\theta(X))$. Finally, the conclusions concerning $\operatorname{rank}(\mathbf{D} f_\theta(X))$ have direct consequences on the dimensions of the image $\{f_{\theta'}(X) \mid \theta' \in B(\theta, \varepsilon)\}$ and the pre-image $\{\theta' \in B(\theta, \varepsilon) \mid f_{\theta'}(X) = f_\theta(X)\}$, where $\varepsilon > 0$ is small enough. The consequences and their implications in machine learning applications are described in greater detail in the next sections.

When compared to the existing similar statements in [179, 82, 28, 81], the particularity of Theorem 103 is that the construction of the sets $\mathcal{O}_j^n$ and $\mathcal{U}_j^X$ permits to include, in the third item of (i) and (ii), a statement on $\operatorname{rank}(\mathbf{D} f_\theta(X))$. To the best of our knowledge, this quantity appears for the first time in conditions of local parameter identifiability, in [28]. It appears independently a few months later, as the core quantity of a study dedicated to the geometric analysis of neural networks, in [82]. In the latter article, this quantity is called the 'batch functional dimension' and we will use this name in this article.

Because the input space of $\mathbf{D} f_\theta(X)$ is always $\mathbb{R}^E \times \mathbb{R}^B$, the quantity $\operatorname{rank}(\mathbf{D} f_\theta(X))$ is upper bounded by the number of parameters $|E| + |B|$. Furthermore, as formalized in [82], because of the invariance by positive rescaling, see the definition and discussion of the relation $\sim$ in Section 5.2, we even have $\operatorname{rank}(\mathbf{D} f_\theta(X)) \leq |E| + |B| - N_1 - \cdots - N_{L-1}$. In fact, when $\operatorname{rank}(\mathbf{D} f_\theta(X)) = |E| + |B| - N_1 - \cdots - N_{L-1}$,

under mild conditions on θ, the network function is locally identifiable around θ, see [28]. That is, $f_\theta(X) = f_{\theta'}(X)$ and $\|\theta - \theta'\|$ small enough imply $\theta \sim \theta'$. Beyond the case of maximal rank value, $\mathrm{rank}(\mathbf{D}f_\theta(X)) = |E| + |B| - N_1 - \cdots - N_{L-1}$, leading to local identifiability, examples of non-identifiable neural networks and rank deficient parameters are already given in [82, 26, 175, 80].

Let us emphasize a simple example illustrating that several rank values can be achieved.

Examples 1. Consider $L \geq 3$, any neuron $v \in V_l$, for $l \in \{2, \ldots, L-1\}$, and $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ such that

$$b_v < 0 \qquad \text{and} \qquad w_{v' \to v} < 0, \text{ for all } v' \in V_{l-1}. \tag{5.3.7}$$

Since, because of the ReLU activation function, for all $x \in \mathbb{R}^{N_0}$ and all $v' \in V_{l-1}$, we have $(f_\theta^{l-1}(x))_{v'} \geq 0$, (5.2.1) guarantees that $(f_\theta^l(x))_v = 0$. This holds for all θ in the open set defined by the equations (5.3.7). In this set, the parameters $(w_{v' \to v})_{v' \in V_{l-1}}$ and b_v have no impact on $f_\theta(X)$, which leads to a rank deficiency of $\mathbf{D}f_\theta(X)$. Going further, consider any $\theta \in \mathbb{R}^E \times \mathbb{R}^B$. According to the above remark, we can change the weights arriving to a given neuron v, and assign them negative values so that (5.3.7) holds, to diminish $\mathrm{rank}(\mathbf{D}f_\theta(X))$. We can redo this operation to many neurons to diminish the rank further. As a conclusion to the example, many values of $\mathrm{rank}(\mathbf{D}f_\theta(X))$ are reached at different places in the parameter/input space.

Let us conclude the section by showing that the quantity $\mathrm{rank}(\mathbf{D}f_\theta(X))$, as well as the sets $\widetilde{\mathcal{O}}_1^n, \ldots, \widetilde{\mathcal{O}}_{m_n}^n$ and $\widetilde{\mathcal{U}}_1^X, \ldots, \widetilde{\mathcal{U}}_{p_X}^X$, are invariant with respect to the positive rescaling and/or neuron permutation relations defined in Section 5.2.

Proposition 104. *Consider any deep fully-connected ReLU network (E, V).*
For any $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{N_0 \times n}$, $j \in [\![1, p_X]\!]$, and $\theta \in \widetilde{\mathcal{U}}_j^X$ we have:

— *for any $\widetilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\widetilde{\theta} \sim \theta$, $\mathbf{D}f_{\widetilde{\theta}}(X)$ is well defined and we have*

$$\mathrm{rank}\big(\mathbf{D}f_{\widetilde{\theta}}(X)\big) = \mathrm{rank}(\mathbf{D}f_\theta(X)).$$

— *for any $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\theta' \sim_s \theta$, we have $\theta' \in \widetilde{\mathcal{U}}_j^X$ and if there is $j' \in [\![1, m_n]\!]$ such that $(X, \theta) \in \widetilde{\mathcal{O}}_{j'}^n$, then $(X, \theta') \in \widetilde{\mathcal{O}}_{j'}^n$.*

The proof of the proposition is in Appendix 5.A.2.

This invariance is a benefit of the complexity measure $\mathrm{rank}(\mathbf{D}f_\theta(X))$. For instance, it does not hold for the local flatness of the empirical risk function studied in [42, 63, 90, 99]. This leads to undesired behaviors [53]. Similarly, complexity measures defined by norms [15, 18, 77, 137] are not invariant to positive rescalings [5].

Consider $X \in \mathbb{R}^{N_0 \times n}$ and $j \in [\![1, p_X]\!]$. Note that $\mathcal{U}_j^X$ corresponds to full-rank cases (for $\mathbf{D}f_\theta(X)$ and in $\widetilde{\mathcal{U}}_j^X$) and $\widetilde{\mathcal{U}}_j^X \backslash \mathcal{U}_j^X$ corresponds to deficient-rank cases.

5. For both flatness and norms, it is, of course, possible to consider the minimum of the complexity criterion over the equivalence class of a θ element. However, this is an additional burden that is not necessary for criteria based on the functional dimension.

Thus, Proposition 104 shows that the rank is stable by the relation $\sim$, and therefore $\sim_s$, when it is full and when it is deficient. As a consequence, not only the set $\widetilde{\mathcal{U}}_j^X$ is stable by the relation $\sim_s$, but also the sets $\mathcal{U}_j^X$ and $\widetilde{\mathcal{U}}_j^X \setminus \mathcal{U}_j^X$. For instance if $\theta \in \mathcal{U}_j^X$ and $\theta' \sim_s \theta$, then $\theta' \in \widetilde{\mathcal{U}}_j^X$ and $\operatorname{rank}(\mathbf{D}f_{\theta'}(X)) = \operatorname{rank}(\mathbf{D}f_\theta(X)) = r_j^X$, thus $\theta' \in \mathcal{U}_j^X$.

Similarly, the sets $\mathcal{O}_j^n$ and $\widetilde{\mathcal{O}}_j^n \setminus \mathcal{O}_j^n$ are stable by the relation $\sim_s$, for $j \in [\![1, m_n]\!]$. For instance, by the same arguments as above, if $(X, \theta) \in \mathcal{O}_j^n$ and $\theta' \sim_s \theta$, then $(X, \theta') \in \mathcal{O}_j^n$.

5.4 Geometric interpretation when X is fixed

The statement of Theorem 103, (i) is used in Section 5.5. In this section, we mostly describe the consequences of Theorem 103, (ii). The next corollary is a straightforward consequence of the constant rank theorem and Theorem 103, (ii).

Corollary 105. *Consider any deep fully-connected ReLU network (E, V).*

For any $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{N_0 \times n}$, $j \in [\![1, p_X]\!]$ and $\theta \in \mathcal{U}_j^X$, there exists $\varepsilon_{X,\theta} > 0$ such that

— *the local image set*

$$\{f_{\theta'}(X) \in \mathbb{R}^{N_L \times n} \mid \|\theta' - \theta\| < \varepsilon_{X,\theta}\}$$

is a smooth manifold of dimension $\operatorname{rank}(\mathbf{D}f_\theta(X))$*;*

— *the local pre-image set*

$$\{\theta' \in \mathbb{R}^E \times \mathbb{R}^B \mid f_{\theta'}(X) = f_\theta(X) \text{ and } \|\theta' - \theta\| < \varepsilon_{X,\theta}\}$$

is a smooth manifold of dimension $|E| + |B| - \operatorname{rank}(\mathbf{D}f_\theta(X))$.

The corollary is illustrated in Figure 5.1. There we consider a regression problem with a fixed target data matrix $Y \in \mathbb{R}^{N_L \times n}$ corresponding to the input matrix $X \in \mathbb{R}^{N_0 \times n}$. We consider the square loss $\|Y - f_\theta(X)\|^2$, for $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, where $\|\cdot\|$ is the Euclidean (Frobenius) norm. We also consider θ^* minimizing the square loss.

Geometry induced implicit regularization In the figure, we display a (fictive) illustrative case, that can be considered as representative of the practice of deep neural networks, and of our numerical experiments in Section 5.7. We consider here that in Corollary 105, $p_X = 7$. Hence, there are 7 sets $\mathcal{U}_1^X, \dots, \mathcal{U}_7^X$ forming a partition of $\mathbb{R}^E \times \mathbb{R}^B$. On the figure, for $j \in [\![1, 7]\!]$, the image of $\mathcal{U}_j^X$, $\{f_\theta(X) \mid \theta \in \mathcal{U}_j^X\}$, is drawn with the same color as $\mathcal{U}_j^X$. Locally has the structure of a smooth manifold of dimension r_j^X. The rank values are $r_1^X = 1$, $r_2^X = 2$, $r_3^X = 1$, $r_4^X = 0$, $r_5^X = 1$, $r_6^X = 1$, $r_7^X = 1$ and thus the full image set $\{f_\theta(X) \mid \theta \in \mathbb{R}^E \times \mathbb{R}^B\}$ is a two-dimensional object. In the figure, this full image set is mainly covered by the two-dimensional image set $\{f_\theta(X) \mid \theta \in \mathcal{U}_2^X\}$, and the six other image sets

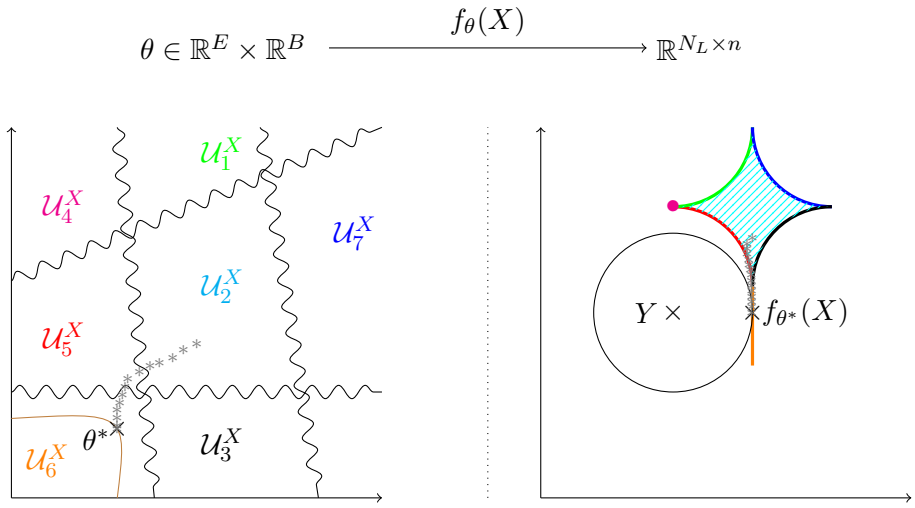

Figure 5.1 – Schematic representation of the sets $\mathcal{U}_j^X$ (left) and the corresponding local image sets $\{f_\theta(X) \mid \theta \in \mathcal{U}_j^X\}$, $j \in [\![1, 7]\!]$ (right). We have $r_1^X = 1$, $r_2^X = 2$, $r_3^X = 1$, $r_4^X = 0$, $r_5^X = 1$, $r_6^X = 1$, $r_7^X = 1$. The image of $\mathcal{U}_2^X$ is the curved diamond-shaped area, hatched in cyan (right). The images of the sets $\mathcal{U}_j^X$ with $r_j^X = 1$ are represented with lines of their respective colors (right). The image of $\mathcal{U}_4^X$ with $r_4^X = 0$ is represented by a magenta bullet point (right). We consider the square loss in $\mathbb{R}^{N_L \times n}$. The target $Y \in \mathbb{R}^{N_L \times n}$ and the global solution $f_{\theta^*}(X)$ of the regression problem are represented (right). The pre-image of $f_{\theta^*}(X)$ is displayed in brown (left). A minimizing sequence is represented by gray stars, in the parameter space (left) and the image space (right).

$\{f_\theta(X) \mid \theta \in \mathcal{U}_j^X\}$, $j \in [\![1, 7]\!] \setminus \{2\}$, of dimension one or zero, are at the boundary of $\{f_\theta(X) \mid \theta \in \mathbb{R}^E \times \mathbb{R}^B\}$. Hence, intuitively they are 'exposed', meaning in particular that if Y does not belong to the full image set, then the optimal prediction $f_{\theta^*}(X)$ is in one of the smaller dimensional $\{f_\theta(X) \mid \theta \in \mathcal{U}_j^X\}$, $j \in [\![1, 7]\!] \setminus \{2\}$. This is an illustration of the **geometry induced implicit regularization** phenomenon put to evidence in this article. In practice, parameters found by minimizing the empirical risk numerically tend to have a small complexity as measured by $\mathrm{rank}(\mathbf{D} f_\theta(X))$, where X is the learning sample. This illustrative situation in Figure 5.1 corresponds to the empirical observations made in Section 5.7.2. In this section, we will even see that, consistently in our experiments, a larger optimal loss leads to a smaller batch functional dimension. We will also see empirically in Section 5.7.2 that for large parameter complexities (W large), the batch functional dimension computed on the learning sample remains moderate. There are two complementary explanations. First, even though the training error is null, because of the soft-max activation on the last layer, the cross-entropy loss slightly differs from zero. Secondly, there may exist θ for which the loss is exactly zero, but this θ is apparently not in the convergence basin in which the local search algorithm optimizes.

Influence of the geometry on the optimization trajectory In Figure 5.1, we also display a (fictive) minimizing sequence, that is a set of pairs $(\theta_n, f_{\theta_n}(X))_{n\in\mathbb{N}}$ obtained by a numerical gradient-descent-based optimization procedure. This sequence is initialized in $\mathcal{U}_2^X$, then passes in $\mathcal{U}_5^X$ and then $\mathcal{U}_6^X$, where the optimal solution lies. This illustrative example is an illustration of the experimental results of Section 5.7.3. There, during the learning phase, the sequence $(\mathrm{rank}(\mathbf{D}f_{\theta_n}(X)))_{n\in\mathbb{N}}$ typically decreases. According to Corollary 105, this corresponds to an objective landscape that becomes flatter and flatter, in the sense that the local dimension of the pre-image of $f_{\theta_n}(X)$ increases. Locally in the parameter space, the objective function is constant along a smooth manifold of a larger dimension. This resembles but slightly differs from the notion of 'flat minima' usually considered to explain the good generalization properties of deep learning [99, 53, 63, 42, 90].

5.5 Rank saturating X, when θ is fixed

In this section, we define dense subset of $\mathbb{R}^E \times \mathbb{R}^B$ and for θ in this subset, we analyze the maximum value of $\mathrm{rank}(\mathbf{D}f_\theta(X))$, for any X of any size, in a dense set. This is a natural notion of complexity, that we call 'computable full functional dimension'. In particular, it is independent of X and measures the expressive potential of the neural network defined by θ. It is linked to the full functional dimension in [82], but can be computed (see Section 5.6.2), thus the name.

After giving the main definitions and establishing the first properties of the considered mathematical objects, we give the main result of this section (Theorem 108), which states that the computable full functional dimension depends only on the attainable activation patterns for the considered θ, when X varies. We also show the invariance of the computable full functional dimension with respect to neural permutation and positive rescaling. We finally establish a simple link with the (local and global) fat-shattering dimensions of the neural networks of architecture (E, V).

For a fixed $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and any activation pattern $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, we denote

$$D_\delta(\theta) = \{x \in \mathbb{R}^{N_0} \mid a(x,\theta) = \delta\}. \tag{5.5.1}$$

It is well known (among many others, see [26]) that the restriction of the function $x \mapsto f_\theta(x)$ to $D_\delta(\theta)$ is affine. It is therefore smooth in $\mathrm{Int}(D_\delta(\theta))$ but, generically, the function is not differentiable at the boundary of $D_\delta(\theta)$. For any $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, following [179], we also define the *achievable activation patterns*

$$A(\theta) = \{\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}} \mid \mathrm{Int}(D_\delta(\theta)) \neq \emptyset\} \tag{5.5.2}$$

and

$$\mathcal{X}_\theta = \bigcup_{\delta\in A(\theta)} \mathrm{Int}(D_\delta(\theta)).$$

Before going further, let us establish that $\mathcal{X}_\theta$ is dense in $\mathbb{R}^{N_0}$. Consider the partial neural network functions $x \mapsto f_\theta^\ell(x)$, $\ell \in [\![0, L]\!]$, see (5.2.1). These functions

are continuous piecewise linear from [26, Proposition 32] (among other references stating this well-known result). Hence the complement set $\mathcal{X}_\theta^c$ is contained in a finite union of hyperplanes. Hence, the set $\mathcal{X}_\theta$ is dense (and open) in $\mathbb{R}^{N_0}$.

We extend this definition to samples and set, for $n \in \mathbb{N}^*$ and $\theta \in \mathbb{R}^E \times \mathbb{R}^B$,

$$\mathcal{X}_\theta^n = \{X \in \mathbb{R}^{N_0 \times n} \mid \forall i \in [\![1, n]\!], x^{(i)} \in \mathcal{X}_\theta\}. \tag{5.5.3}$$

The set $\mathcal{X}_\theta^n$ is the nth order tensor product of the set $\mathcal{X}_\theta$ with itself. By construction, the set $\mathcal{X}_\theta^n$ is open and dense in $\mathbb{R}^{N_0 \times n}$, for all n and θ.

The results of this section will apply to all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ except those in a subset $\mathcal{Z}$, which will turn out to be of Lebesgue measure zero – see Proposition 106. To define the set $\mathcal{Z}$, we first define, for all $n \in \mathbb{N}^*$ and all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$,

$$z_n(\theta) = \left\{X \in \mathbb{R}^{N_0 \times n} \mid (X, \theta) \in \left(\cup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c\right\}, \tag{5.5.4}$$

where $\mathcal{O}_1^n, \dots, \mathcal{O}_{m_n}^n$ are defined in (5.3.3) and described in Theorem 103. The set $z_n(\theta)$ is closed and therefore Lebesgue measurable. We write, for all $n \in \mathbb{N}^*$,

$$\mathcal{Z}_n = \left\{\theta \in \mathbb{R}^E \times \mathbb{R}^B \mid z_n(\theta) \text{ has positive Lebesgue measure in } \mathbb{R}^{N_0 \times n}\right\} \tag{5.5.5}$$

and $\mathcal{Z} = \cup_{n \in \mathbb{N}^*} \mathcal{Z}_n$. We state in the following proposition the most important properties of $\mathcal{Z}$, used in the remaining of the article.

Proposition 106. *Consider any deep fully-connected ReLU network (E, V).*

(i) For all $n \in \mathbb{N}^*$, the set $\mathcal{Z}_n$ is Lebesgue measurable and has zero Lebesgue measure on $\mathbb{R}^E \times \mathbb{R}^B$.

(ii) The set $\mathcal{Z}$ is Lebesgue measurable and has zero Lebesgue measure on $\mathbb{R}^E \times \mathbb{R}^B$.

(iii) For all $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$, all $n \in \mathbb{N}^*$, and all $X \in \mathcal{X}_\theta^n$, the function $\theta' \mapsto f_{\theta'}(X)$ is a polynomial function in a neighborhood of θ and it is therefore differentiable at the point θ.

The proof of the proposition is in Appendix 5.B.1.

Using Proposition 106, *(iii)*, we can define, for all $n \in \mathbb{N}^*$ and all $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$, the main objects studied in this section

$$r_n^*(\theta) = \max_{X \in \mathcal{X}_\theta^n} \operatorname{rank}(\mathbf{D} f_\theta(X)), \tag{5.5.6}$$

and the computable full functional dimension

$$r^*(\theta) = \max_{n \in \mathbb{N}^*} r_n^*(\theta). \tag{5.5.7}$$

Notice that, although $\mathcal{X}_\theta^n$ is open and dense in $\mathbb{R}^{N_0 \times n}$ and the rank is lower semi-continuous, the existence of $X \in \mathbb{R}^{N_0 \times n} \setminus \mathcal{X}_\theta^n$ such that $\mathbf{D} f_\theta(X)$ is well defined and $\operatorname{rank}(\mathbf{D} f_\theta(X)) > r_n^*(\theta)$ is not excluded. The computable full functional dimension $r^*(\theta)$ therefore may slightly differ from the full functional dimension defined in [82]. It lower bounds the full functional dimension. We will see in Section 5.6.2

that its advantage is that it can be computed with a random X. Notice finally that in Example 1 the rank deficiency caused by negative weights is independent of X. Therefore, $r^*(\theta)$ achieves several values, as θ varies.

Notice also that, although we take the maximum over all $n \in \mathbb{N}^*$, we know that since, for all $n \in \mathbb{N}^*$ and all $X \in \mathbb{R}^{N_0 \times n}$, $\theta \mapsto \mathbf{D}f_\theta(X)$ always has the same input dimension $|E|+|B|$, see (5.2.2), the maximum is reached for $n \leq |E|+|B|$ (see also Proposition 111 below).

The following proposition states that $r_n^*(\theta)$ equals the largest of all the r_j^n, as defined in (5.3.2), that are reachable when X varies, for the given θ.

Proposition 107. *Consider any deep fully-connected ReLU network (E, V).*
For any $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and $n \in \mathbb{N}^*$,

$$r_n^*(\theta) = \max_{j \in I_n(\theta)} r_j^n,$$

where

$$I_n(\theta) = \{j \in [\![1, m_n]\!] \mid \exists X \in \mathbb{R}^{N_0 \times n}, (X, \theta) \in \mathcal{O}_j^n\}. \tag{5.5.8}$$

The proposition is proved in Appendix 5.B.2.

The following theorem states that the achievable activation patterns, contained in $A(\theta)$, determine $r^*(\theta)$. It also states that when the prediction has more affine areas, that is for a fixed θ, $X \mapsto f_\theta(X)$ is piece-wise affine with with more pieces, then this prediction is more complex, in the sense of r^*.

Theorem 108. *Consider any deep fully-connected ReLU network (E, V).*
For any θ and θ' in $(\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$,

$$\text{if} \quad A(\theta) \subseteq A(\theta') \quad \text{then} \quad r^*(\theta) \leq r^*(\theta');$$

as a consequence,

$$\text{if} \quad A(\theta) = A(\theta') \quad \text{then} \quad r^*(\theta) = r^*(\theta').$$

The proof of the theorem is in Appendix 5.B.3.

Next, we show that $r_n^*(\theta)$, $r^*(\theta)$ and $\mathcal{Z}$ are invariant by neuron permutation and positive rescaling (recall the relations $\sim$, $\sim_s$ and $\sim_p$ presented in Section 5.2).

Proposition 109. *Let $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and $\widetilde{\theta} \sim \theta$. Then $\theta \in \mathcal{Z} \iff \widetilde{\theta} \in \mathcal{Z}$. Also, if $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$, $r_n^*(\theta) = r_n^*(\widetilde{\theta})$ for $n \in \mathbb{N}^*$ and $r^*(\theta) = r^*(\widetilde{\theta})$.*

The proof of the proposition is in Appendix 5.B.4.

Let us conclude this section by showing that $r^*(\theta)$ provides a lower-bound on the (local and global) fat-shattering dimensions for neural networks. The fat-shattering dimension of a family of regression functions is a well-known measure of complexity, see for instance [6, Chapter 11]. In the rest of the section, we let $N_L = 1$ and, for a subset $A \subseteq \mathbb{R}^E \times \mathbb{R}^B$, for $\gamma > 0$, the *fat-shattering dimension* of the family $\{f_\theta \mid \theta \in A\}$, that we write $\mathrm{fS}_{A,\gamma}$, is defined as follows. It is the largest $n \in \mathbb{N}^*$ such that

there exist $x^{(1)}, \dots, x^{(n)} \in \mathbb{R}^{N_0}$ and $t_1, \dots, t_n \in \mathbb{R}$ such that for all $I \subseteq [\![1, n]\!]$, there is $\theta \in A$ such that for $i \in I$, $f_\theta(x^{(i)}) \geq t_i + \gamma$ and for $i \in [\![1, n]\!] \setminus I$, $f_\theta(x^{(i)}) \leq t_i - \gamma$. If this property holds for all n then we let $\mathrm{fS}_{A,\gamma} = \infty$.

The intuition is that $\mathrm{fS}_{A,\gamma}$ is the largest number n of input points for which all 2^n combinations of being above or below the threshold t_i by a margin γ, $i \in [\![1, n]\!]$, can be reached by the functions $\{f_\theta \mid \theta \in A\}$, see [6]. When A is a small ball centered at a parameter of interest we shall call $\mathrm{fS}_{A,\gamma}$ a local fat-shattering dimension, and when $A = \mathbb{R}^E \times \mathbb{R}^B$, we shall call $\mathrm{fS}_{A,\gamma}$ a global fat-shattering dimension. The next proposition shows the announced lower bound.

Proposition 110. *Consider any deep fully-connected ReLU network (E, V). Let $N_L = 1$ and $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$. Then for any $\varepsilon > 0$, there is $\gamma > 0$ such that we have the following lower bound on the local fat-shattering dimension,*

$$\mathrm{fS}_{B(\theta,\varepsilon),\gamma} \geq r^*(\theta). \tag{5.5.9}$$

As a consequence, there is $\gamma' > 0$ such that the global fat-shattering dimension is lower bounded as follows,

$$\mathrm{fS}_{\mathbb{R}^E \times \mathbb{R}^B, \gamma'} \geq \max_{\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}} r^*(\theta).$$

The proof of the proposition is in Appendix 5.B.5.

It consists first in obtaining local continuous differentiability, with an invertible square Jacobian matrix and second in applying the inverse function theorem. Variations of this second step were already carried out in the literature, in particular in [59].

Remark that the same proof would also apply to other measures of complexity, for instance the Vapnik–Chervonenkis (VC) dimension [6, Chapter 3] of the binary classifiers indexed by θ and obtained by taking the sign of $f_\theta - f_{\theta_0}$ for any fixed $\theta_0 \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$.

Our motivation, for studying the fat-shattering dimension (or the VC-dimension), is their relationships with notions of generalization errors in machine learning, and with uniform convergence in probability and statistics, see in particular [5, 17, 50, 187] and references therein. In particular, Proposition 110 indicates that the computable full functional dimension $r^*(\theta)$ can be seen as relevant for studying the generalization error of neural networks in machine learning. This is argued in Section 5.1 and confirmed numerically in Section 5.7.

Finally, remark that [16, 20, 113] relate the global VC dimension of neural networks to their number of parameters (and their depths) while in Proposition 110 we consider local or global dimensions and relate them to the (smaller) computable full functional dimension.

5.6 Computational considerations

5.6.1 How to compute $\mathrm{rank}(\mathbf{D}f_\theta(X))$

For a given $X \in \mathbb{R}^{N_0 \times n}$ and a given $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, $\mathrm{rank}(\mathbf{D}f_\theta(X))$ is computed using the backpropagation and numerical linear tools computing the rank of a matrix. To justify the computations, let us first recall the classical backpropagation algorithm for computing the gradients with respect to the parameters of the network, for a given loss $Lo : \mathbb{R}^{N_L} \times \mathbb{R}^{N_L} \longrightarrow \mathbb{R}$. We will then describe how to use the backpropagation to compute $\mathrm{rank}(\mathbf{D}f_\theta(X))$. We conclude with implementation recommendations.

For a given input $x \in \mathbb{R}^{N_0}$ and a given output $y \in \mathbb{R}^{N_L}$, backpropagation computes the gradient $\nabla Lo(f_\theta(x), y)$ of the function $\theta \longmapsto Lo(f_\theta(x), y)$. To do so, it first computes $f_\theta(x)$ and stores the intermediate values $y^\ell = f_\theta^\ell(x) \in \mathbb{R}^{N_\ell}$, for $\ell \in [\![0, L]\!]$. This is known as the 'forward pass'. Then, backpropagation computes the vector of errors η^L defined by

$$\eta^L = \frac{\partial Lo}{\partial y_1}(f_\theta(x), y),$$

where $\frac{\partial Lo}{\partial y_1}(f_\theta(x), y) \in \mathbb{R}^{N_L}$ is the gradient of $y_1 \longmapsto Lo(y_1, y)$, at the point $f_\theta(x)$. This vector is then backpropagated, from $\ell = L$ to $\ell = 1$ thanks to the equation

$$\forall v \in V_{\ell-1} \qquad \eta_v^{\ell-1} = \sigma'(y_v^{\ell-1}) \sum_{v' \in V_\ell} w_{v \to v'} \eta_{v'}^\ell \tag{5.6.1}$$

where $\sigma'(t) = 1$ if $t > 0$ and $\sigma'(t) = 0$ if $t < 0$. This allows to recursively obtain the error vectors $\eta^\ell \in \mathbb{R}^{N_\ell}$, for all $\ell \in [\![1, L]\!]$. This yields the gradients thanks to the formulas

$$\forall \ell \in [\![1, L]\!], \forall v \in V_{\ell-1}, v' \in V_\ell, \qquad \frac{\partial Lo(f_\theta(x), y)}{\partial w_{v \to v'}} = y_v^{\ell-1} \eta_{v'}^\ell$$

and

$$\forall \ell \in [\![1, L]\!], \forall v \in V_\ell, \qquad \frac{\partial Lo(f_\theta(x), y)}{\partial b_v} = \eta_v^\ell.$$

This allows computing the gradients for one example (x, y). For a batch, the algorithm is repeated for each example $(x^{(i)}, y^{(i)})$, and the average of the so obtained gradients is computed.

Let us now make the connection between backpropagation and the computation of $\mathrm{rank}(\mathbf{D}f_\theta(X))$. Vectorizing both the input and output spaces of $\theta \longmapsto f_\theta(X)$, we first notice that $\mathrm{rank}(\mathbf{D}f_\theta(X)) = \mathrm{rank}(Jf_\theta(X))$, where the Jacobian matrix $Jf_\theta(X) \in \mathbb{R}^{nN_L \times (|E|+|B|)}$ takes the form

$$Jf_\theta(X) = \begin{pmatrix} Jf_\theta(x^{(1)}) \\ \vdots \\ Jf_\theta(x^{(n)}) \end{pmatrix}$$

and, for all $i \in [\![1, n]\!]$, $Jf_\theta(x^{(i)}) \in \mathbb{R}^{N_L \times (|E|+|B|)}$ is the Jacobian matrix of $\theta \longmapsto f_\theta(x^{(i)})$. We construct the matrix $Jf_\theta(X)$ by successively computing each of its lines, i.e. computing each line of $Jf_\theta(x^{(i)})$ for all $i \in [\![1, n]\!]$.

For a given $i \in [\![1, n]\!]$ and $v \in V_L$, the line corresponding to v of $Jf_\theta(x^{(i)})$ is indeed simply obtained as the transpose of $\nabla Lo_v(f_\theta(x^{(i)}), y^{(i)})$ for the function $Lo_v : \mathbb{R}^{N_L} \times \mathbb{R}^{N_L} \longrightarrow \mathbb{R}$ defined by $Lo_v(y_1, y_2) = (y_1)_v$, for all $(y_1, y_2) \in \mathbb{R}^{N_L} \times \mathbb{R}^{N_L}$. We indeed have $Lo_v(f_{\theta'}(x^{(i)}), y^{(i)}) = f_{\theta'}(x^{(i)})_v$ for all θ'. The gradient $\nabla Lo_v(f_\theta(x^{(i)}), y^{(i)})$ is obtained using the backpropagation algorithm described above. Notice that for a given $v \in V_L$, using the definition of Lo_v, we always have $\eta_v^L = 1$ and $\eta_{v'}^L = 0$ for all $v' \neq v$. We need however to compute the forward pass in order to compute the vectors y^ℓ, for $\ell \in [\![0, L-1]\!]$. Finally, once $Jf_\theta(x^{(i)})$ is computed its rank is obtained using standard linear algebra algorithms.

Our implementation uses existing the automatic differentiation of Tensorflow. It is possible to call the method GradientTape.gradients, which can compute $Jf_\theta(x)$ for a single example x, and to repeat it for each example $x^{(i)}$. However, it is more efficient to use GradientTape.jacobian which allows to compute directly $Jf_\theta(X)$. We do not report the details of the experiments here but we found even more efficient to cut X in sub-batches and repeatedly call GradientTape.jacobian, when appropriately choosing the size of the sub-batches.

Once $Jf_\theta(X)$ built, the value of $\text{rank}(Jf_\theta(X))$ can be computed with the np.linalg.rank function of Numpy, or using the accelerated rank computation of Pytorch with a GPU, which improves the speed by some factors. Note that the limiting factor when computing $\text{rank}(Jf_\theta(X))$ for large networks and for n large is the computation of the rank and not the construction of $Jf_\theta(X)$.

The codes are available at [27].

5.6.2 How to compute r^*

In this section, our goal is to estimate the maximal rank $r^*(\theta)$ from $\text{rank}(\mathbf{D}f_\theta(X))$, where $X \in \mathbb{R}^{N_0 \times n}$ is a random dataset composed of n i.i.d samples. Such an estimate is already considered in [80]. Intuitively, the bigger n is, the better the estimation. Indeed, we provide an upper bound on the probability that $\text{rank}(\mathbf{D}f_\theta(X)) < r^*(\theta)$ as a function of n, see Theorem 113. This probability also depends on the probability of generating an example in the least probable linear region of $x \mapsto f_\theta(x)$. This result can be compared to the smallest possible sample size $n^*(\theta)$ obtained if an optimal $X \in \mathbb{R}^{N_0 \times n^*(\theta)}$ was provided by an oracle, see Proposition 111.

The following proposition proves that the smallest possible sample size has the order of magnitude of $r^*(\theta)$. Before stating the proposition, we remind that, since the input space of $\mathbf{D}f_\theta(X)$ is always $\mathbb{R}^E \times \mathbb{R}^B$, we always have[6] $r^*(\theta) \leq |E| + |B|$.

Proposition 111. *Consider any deep fully-connected ReLU network (E, V).*

6. A tighter upper-bound taking into account the positive rescaling invariance of ReLU networks is given in [82].

Let $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$. Consider the sequence $(r_n^*(\theta))_{n\in\mathbb{N}^*}$. There exists (a unique) $n^*(\theta) \in \mathbb{N}^*$ such that this sequence is increasing on $[\![1, n^*(\theta)]\!]$ and stationary (constant) on $\mathbb{N}^* \setminus [\![1, n^*(\theta)]\!]$. We also have

$$\frac{r^*(\theta)}{N_L} \leq n^*(\theta) \leq r^*(\theta).$$

The proof of the proposition is in Appendix 5.C.1. As its proof shows, the following proposition is a direct consequence of the proposition 107. It already gurantees that, without any knowledge of the problem, a random X following a sufficiently spread distribution can be used to calculate $r_n^*(\theta)$ and therefore $r^*(\theta)$. Its purpose is to illustrate how the statements in the previous sections can be used to calculate $r^*(\theta)$. A better statement is given in Theorem 113.

Proposition 112. *Consider any deep fully-connected ReLU network (E, V).*

Let $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$. Let $n \in \mathbb{N}^*$. The set $\{X \in \mathcal{X}_\theta^n \mid \operatorname{rank}(\mathbf{D}f_\theta(X)) = r_n^*(\theta)\}$ has non-zero Lebesgue measure (on $\mathbb{R}^{N_0 \times n}$).

Proof of Proposition 112. From Proposition 107, there exists $j \in I_n(\theta)$ such that $r_j^n = r_n^*(\theta)$. We then have the inclusion

$$\{X \in \mathcal{X}_\theta^n \mid (X, \theta) \in \mathcal{O}_j^n\} \subseteq \{X \in \mathcal{X}_\theta^n \mid \operatorname{rank}(\mathbf{D}f_\theta(X)) = r_n^*(\theta)\}.$$

Since $\mathcal{O}_j^n$ is open, the left-hand set above is an open set, which is non-empty by definition of $I_n(\theta)$, in (5.5.8). Hence, the right-hand set above has a non-zero Lebesgue measure. □

Theorem 113. *Consider any deep fully-connected ReLU network (E, V).*

Let us consider a distribution $\mathcal{G}$ over $\mathbb{R}^{N_0}$, that is absolutely continuous with respect to Lebesgue measure, with strictly positive density. Assume we sample randomly and independently the vectors $x^{(i)} \in \mathbb{R}^{N_0}, i \in [\![1, n]\!]$, following the distribution $\mathcal{G}$, for some $n \in \mathbb{N}^*$. Let $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$.

Let

$$p = \min_{\delta \in A(\theta)} \mathbb{P}\left(x^{(1)} \in \operatorname{Int} D_\delta(\theta)\right),$$

where $A(\theta)$ and $D_\delta(\theta)$ are defined in (5.5.2) and (5.5.1). Note that we have $p > 0$, because $A(\theta)$ is finite and, for all $\delta \in A(\theta)$, $D_\delta(\theta)$ has nonempty interior.

Then the following holds.

1. *Consider i.i.d. Bernoulli random variables $B_1, \ldots, B_n$, with $\mathbb{P}(B_1 = 1) = p$ and $\mathbb{P}(B_1 = 0) = 1 - p$. We have*

$$\mathbb{P}\left(\operatorname{rank}\left(\mathbf{D}f_\theta\left((x^{(i)})_{1\leq i\leq n}\right)\right) < r^*(\theta)\right) \leq \mathbb{P}(B_1 + \cdots + B_n < n^*(\theta)).$$

2. *As a consequence, if $n \geq 2n^*(\theta)/p$,*

$$\mathbb{P}\left(\operatorname{rank}\left(\mathbf{D}f_\theta\left((x^{(i)})_{1\leq i\leq n}\right)\right) < r^*(\theta)\right) \leq \frac{4}{np}.$$

The proof of the proposition is in Appendix 5.C.2.

A first consequence of the theorem is that if one simply adds columns to an input matrix X randomly and independently, the corresponding value of $\operatorname{rank}(\mathbf{D}f_\theta(X))$ will reach the computable full functional dimension, almost surely (this consequence alone could be seen/proved more simply). This can for instance help understand the experimental results of [80].

The proposition then provides two upper bounds (Items 1 and 2) on the probability of not reaching the computable full functional dimension, as a vanishing function of the number of columns n.

A beneficial feature of these upper bounds is that they are based on p, the smallest probability of reaching a given region $\operatorname{Int} D_\delta(\theta)$, relative to a given activation pattern δ, for a *single* column sample. Importantly, the probabilities of reaching several given activation patterns *simultaneously* over multiple samplings of the columns, which would be typically much smaller than p, are not involved.

Also, the upper bounds are based on $n^*(\theta)$, the smallest number of columns for X such that we can have $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r^*(\theta)$. This is natural since the larger $n^*(\theta)$, the more samples we need to have a non-zero probability of reaching the computable full functional dimension. Typically, for n of the order of magnitude of $n^*(\theta)$ (for instance such that $np \gtrsim 2n^*(\theta)$), since $n^*(\theta)$ is usually large, we already have a high probability that the lower-bound $\operatorname{rank}\Big(\mathbf{D}f_\theta\left((x^{(i)})_{1\leq i\leq n}\right)\Big)$ coincides with $r^*(\theta)$.

The first upper bound (Item 1) is the tighter and most general. The second one (Item 2) simply follows from Chebyshev's inequality and is provided for the sake of obtaining a straightforward compact bound. Other upper bounds could be obtained simply from Item 1, using for instance the Hoeffding inequality.

5.7 Experiments

When finalizing the Ph.D. report, we realized there was a small discrepancy between the network considered in the experiments and the networks studied in the article but did not have time to correct it. In all the experiments, the last layer includes a softmax activation function that is taken into account by the backpropagation and therefore affects the functional dimensions.

The setting of the experiments is described in Section 5.7.1. In Section 5.7.2, we describe the results of an experiment in which we compute the functional dimensions when the number of parameters of the network grows. In Section 5.7.3, we compute functional dimensions during the learning phase, emphasizing the *geometry induced implicit regularization* illustrated in Figure 5.1. In Section 5.7.4, we investigate the impact of the corruption of the inputs of the learning sample on functional dimensions. In Section 5.7.5, we do the same experiment but corrupt the outputs of the learning sample.

The Python codes implementing the experiments described in this section are available at [27].

5.7.1 Experiments description

In the experiments of Section 5.7.2, 5.7.3, 5.7.4 and 5.7.5, we evaluate the behavior of different complexity measures for the classification of a subpart of the MNIST dataset increases.

We consider a fully-connected feed-forward network of depth $L = 4$, of width $(N_0, N_1, N_2, N_3, N_4) = (784, w, w, w, 10)$, for different values of $w \in [\![1, 60]\!]$. The tested values of w depend on the experiment/section. The total number of parameters of the network is equal to $784w + 2w^2 + 10w + 3w + 10$. The hidden layers (1, 2, 3) include a ReLU activation function. The last layer includes a soft-max activation function. We randomly extract a training sample $(X_{\text{train}}, Y_{\text{train}})$ and a test sample $(X_{\text{test}}, Y_{\text{test}})$ from MNIST. The sizes of the samples depend on the experiment/section. They are tuned so that the computing time of each experiment remains reasonable.

For given w and $(X_{\text{train}}, Y_{\text{train}})$, we tune the parameters of the network to minimize the cross-entropy on the training set X_{train}. This is achieved using the Glorot uniform initialization for the weights while the biases are initialized to 0, and using the stochastic gradient descent 'sgd' as optimizer with a learning rate of 0.1 and a batch size of 256. The number of epochs depends on the experiment/section.

In the figures presenting the results of the experiments, we display the following quantities:

— Max rank: the maximal theoretically possible value of $\text{rank}(\mathbf{D}f_\theta(X))$ for any sample X and parameter θ. It is equal to $|E| + |B| - N_1 - \cdots - N_{L-1} = N_0N_1 + N_1N_2 + \cdots + N_{L-1}N_L + N_L$ (see the bound provided in [82], Theorem 7.1). With the architecture described above, for a given w, the Max_rank is equal to $2w^2+794w+10$. This is very close to the number of parameters $2w^2+797w+10$. With the values of w considered in the forthcoming experiments, the predominant term is $794w$.

— Rank X_random: an evaluation of the computable full functional dimension, according to the statement of Section 5.6.2, by computing $\text{rank}(\mathbf{D}f_\theta(X_{\text{random}}))$ with a random iid sample X_{random}, where each example of the sample is a Gaussian random vector. The number of examples in the sample depends on the experiment/section.

— Rank X_test: It corresponds to $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$, where X_{test} is the test sample introduced above. It is meant to provide an estimation of the functional dimension over the distribution of the inputs (the MNIST images), in contrast with X_{random} which samples images outside the distribution of the inputs.

— Rank X_train: It corresponds to $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$, where X_{train} is the training sample mentioned above. It is the batch functional dimension.

— Train loss: the final value of the training loss at the end of the training.

— Test error: the proportion of images of X_{test} that are misclassified by the network.

— Train error: the proportion of images of X_{train} that are misclassified by the

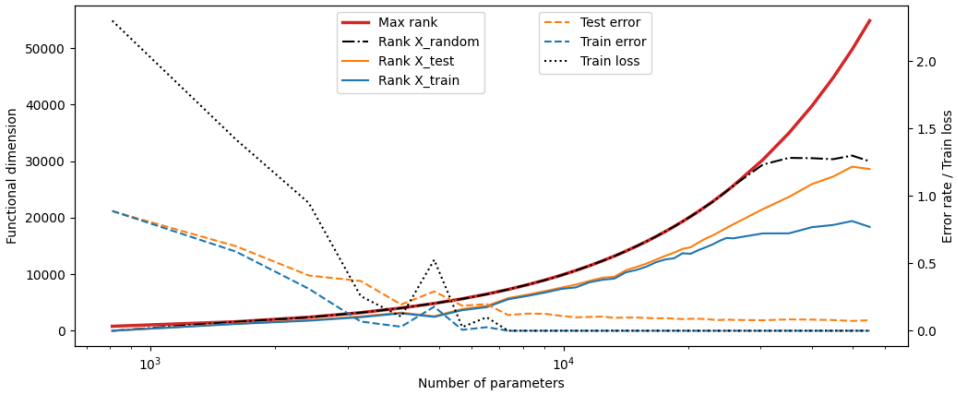

Figure 5.2 – Behavior of different complexity measures as the size of the network increases. The values of the several ranks are to be read on the left axis, titled 'functional dimension'. The values for the test and train errors and the train loss are to be read on the right axis.

network.

5.7.2 Behavior of the functional dimensions as the network width increases

In this experiment, we evaluate the functional dimensions when the width w varies between 1 and 60. More precisely, we consider all $w \in [\![1, 30]\!]$ and all $w \in \{35, 40, 45, 50, 55, 60\}$. The number of parameters of the network varies between 809 and 55030

We randomly extract a training sample $(X_{\text{train}}, Y_{\text{train}})$ of size $n = 4000$ and a test sample $(X_{\text{test}}, Y_{\text{test}})$ of size 10000 from MNIST. The size of the random sample X_{random} is 40000. We optimize the network parameters during 1000 epochs.

The results of the experiment are in Figure 5.2. Except for $w = 6$, when increasing the number of parameters, the train loss, the train error and the test error decrease. For $w \geq 9$, i.e. when the number of parameters is superior or equal to 7345, the train error is equal to 0: the network is able to fit perfectly the training images. However, the test error continues to decrease even after the train error reaches 0: from 0.117 when $w = 9$ to 0.078 when $w = 60$.

As we can see, the quantity $\text{rank}(\mathbf{D}f_\theta(X_{\text{random}}))$ in the case of the 40000 images generated as Gaussian vectors is nearly equal to its maximum theoretical value Max_rank, when the number of parameters is smaller than 25000. Then, it saturates close to 30000.

The ranks $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ and $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$ are nearly equal when the number of parameters is smaller than 15000 ($w = 19$). For these sizes, it seems to indicate that $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$ is indeed equal to the functional dimension over the distribution of inputs, and that $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ already attains it, which means

that adding MNIST images to X_{train} would not increase $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$. Then, for higher numbers of parameters, a gap appears between the two ranks, which seems to indicate that, in contrast to the previous case, the number of images in X_{train} limits $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$. Furthermore, while both ranks are not far from the maximum rank for small numbers of parameters, the gap increases with the number of parameters, to the point where the shape of the curves seem to diverge: while the maximum rank is close to proportional to the number of parameters, the ranks $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ and $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$ seem to increase less and less with the number of parameters.

5.7.3 Behavior of the functional dimensions during training

We consider the setting described in Section 5.7.1, with a train set X_{train} of size 4000, a test set X_{test} and a random set X_{random} both of size 20000. We fix the value of w to 30. The architecture is $(784, 30, 30, 30, 10)$, which corresponds to a total number of parameters equal to 25720. The previous experiment only computes quantities after the training is done. This time, we fix a total number of epoch to 1400 and we consider what happens during training, throughout the epochs.

We compute repeatedly the rank $\mathbf{D}f_\theta(X)$ for $X = X_{\text{train}}$ $X = X_{\text{test}}$ and $X = X_{\text{random}}$. These computations are performed during a single training, at epochs $\{40, 80, 120, 160, 200, 240, 280, 320, 360,$
$400\} \cup \{600, 800, 1000, 1200, 1400\}$.

The quantities Max rank, Rank X_random, Rank X_test, Rank X_train, Train loss, Test error and Train error are the same as in the previous experiment, except they are considered at different times of the training rather than after the training. We plot these quantities in Figure 5.3.

The rank $\text{rank}(\mathbf{D}f_\theta(X_{\text{random}}))$ is always equal to the maximum rank, here equal to 25630, up to a small error of 1 or 2.

We plot the train loss which decreases throughout the epochs, and the train error which decreases and reaches 0 at epoch 120, after what all training images are always correctly classified. The test error decreases the most in the first 80 epochs, after what it continues to decrease, although really slowly.

We observe that the rank $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ consistently decreases during training. Such a behavior is consistent with the geometric interpretation of Section 5.4. The rank $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$ also decreases, with a more gentle slope. This indicates that not only the batch functional dimension with respect to X_{train} decreases, but also the functional dimension with respect to the distribution of inputs.

5.7.4 Behavior of the functional dimensions when X is corrupted

We consider the same setting as the experiments of Section 5.7.1, with $w = 30$, which corresponds to a total number of parameters equal to 25720. The size of the test sample is 20000.

The network is trained, during 3000 epochs, repeatedly over different train sets

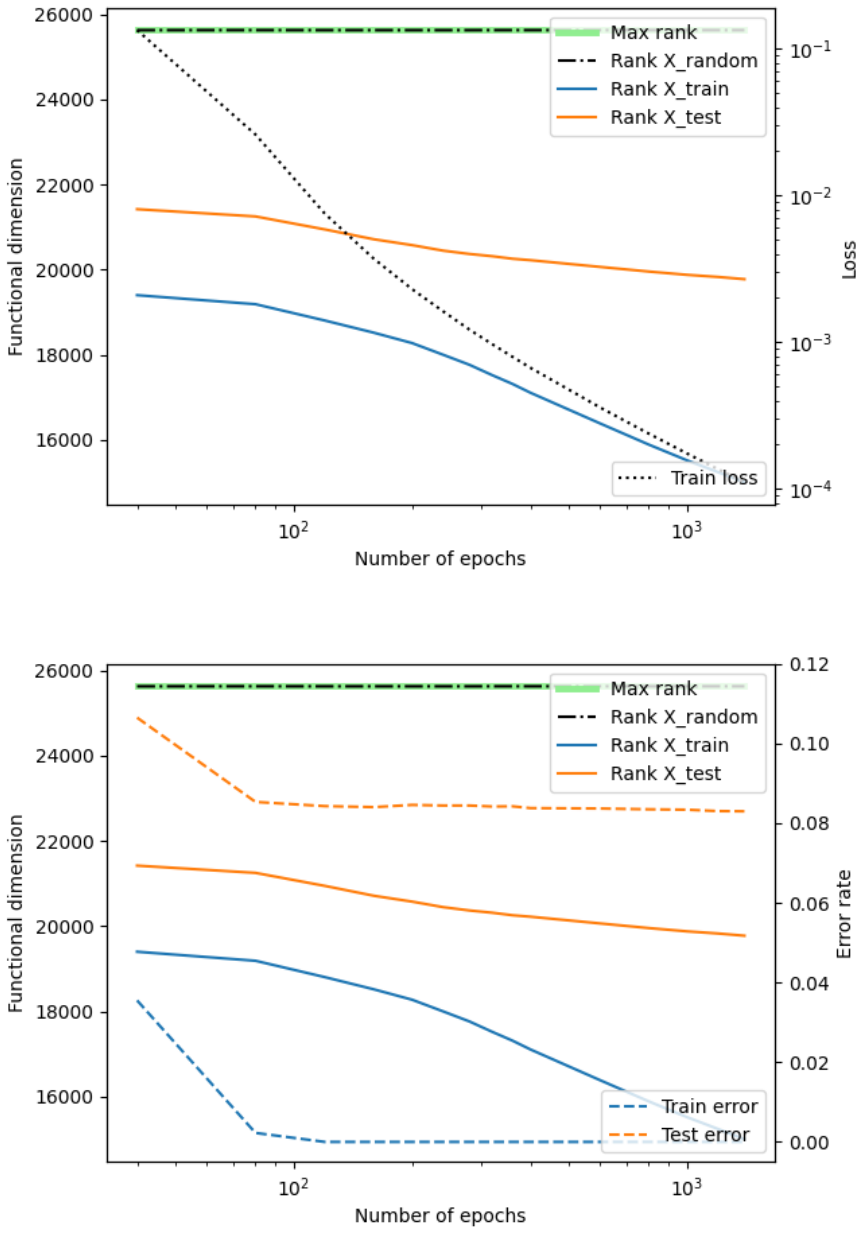

Figure 5.3 – Behavior of different complexity measures during training. The values of the different ranks are to be read on the left axis, titled 'functional dimension'. The values of the train loss (on the top figure), and the values of the test and train errors (on the bottom figure) are to be read on the right axis.

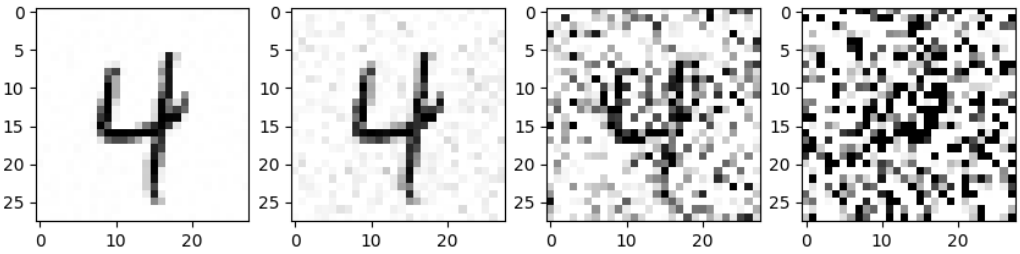

Figure 5.4 – Effect of the noise on a MNIST image for different amplitudes. From left to right, a MNIST image to which has been added a Gaussian noise, of amplitude $10^{-2}, 10^{-1}, 5 \times 10^{-1}$ and 10^{0} respectively.

of size 6000 made of MNIST images. We add to the train images a Gaussian noise, before clipping the values of the pixels between 0 and 1, to stay consistent with black and white images. We do the same for the test images. For each training, we use a different noise amplitude, which overall varies between 0 and 1. We represent visually an image with different levels of noise in Figure 5.4. The network is trained to the point it is able to interpolate the training examples: for all the settings, the final train error is equal to 0.

Once the training done, we compute the quantities Max rank, Rank X_test, Rank X_train, Test error and Train error described in Section 5.7.1, for the different noise levels. We plot these quantities in Figure 5.5.

As already said, the expressiveness of the network permits to fit the learning data perfectly. The training error is zero for all noise levels. However, the noise has two effects: an effect on the distribution of inputs which becomes more complex and an effect on the difficulty of the problem. Indeed, as is reflected by the increase of the test error, the problem becomes more difficult. The curve representing $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ also tends to increase. This phenomenon is coherent with the fact that the batch functional dimension is linked to activation patterns, which are linked to the distribution of the inputs, which –as already said– are made more complex by the noise. The quantity $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$ seems quite stable. Interestingly, the batch functional dimensions $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ and $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$ seem to converge with increase of the noise.

5.7.5 Behavior of the functional dimensions when Y is corrupted

We consider the setting described in Section 5.7.1. We take $w = 25$, which corresponds to a number of parameters equal to 21185. The size of the test set X_{test} and of the random Gaussian images X_{random} are both equal to 15000. The learning performs 3000 epochs.

Following [134, 195], we study what happens when part of the labels of the training set are corrupted with random labels. The network is trained repeatedly over different train sets of size 4000 made of MNIST images. A varying number of

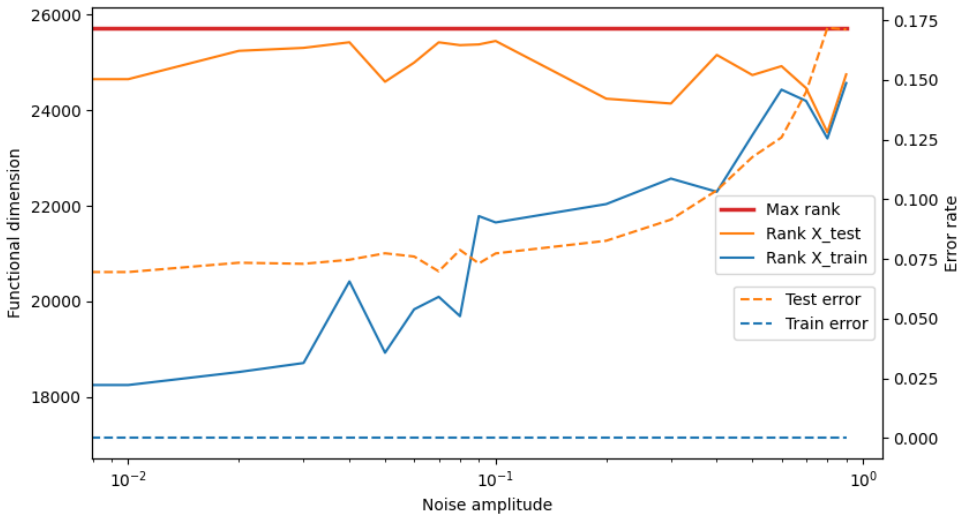

Figure 5.5 – Behavior of different complexity measures when noise is added to the input images. The values of $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ are to be read on the left axis, titled 'functional dimension'. The values of the test and train errors are to be read on the right axis.

the labels associated to the training images are set to a random value, according to the uniform distribution over $\{0, 1, \ldots, 9\}$. The quantity of images with random labels varies from 0 to 3500. Each time, the network is trained to the point it is able to interpolate the training set, even with the random labels.

Intuitively, the more corrupted labels there are, the more complex the function interpolating the training data should be. The distribution of the inputs is however the same and it is not clear if the linear regions, defined by the activation patterns, and used to describe the interpolating function need to change much when corruption increases. The purpose of this experiment is to see if the functional dimension increases with the proportion of corrupted labels.

The quantities Max_rank, Rank_X_random, Rank_X_test, Rank_X_train, Train loss, Test error and Train error are described in Section 5.7.1. We plot these quantities as a function of the number of corrupted labels in Figure 5.6.

As expected, the test error (the proportion of images in X_{test} that are misclassified) increases with the number of corrupted labels: from 0.08 when no label is corrupted to 0.83 when 3500 images out of 4000 have random labels.

In this experiment, again, we observe that $\text{rank}(\mathbf{D}f_\theta(X_{\text{random}}))$ is always nearly equal to the maximum rank, which is here equal to 21110.

The ranks $\text{rank}(\mathbf{D}f_\theta(X_{\text{train}}))$ and $\text{rank}(\mathbf{D}f_\theta(X_{\text{test}}))$ do not seem to increase with the corruption of the labels. Beside when 250 labels are corrupted, they remain stable.

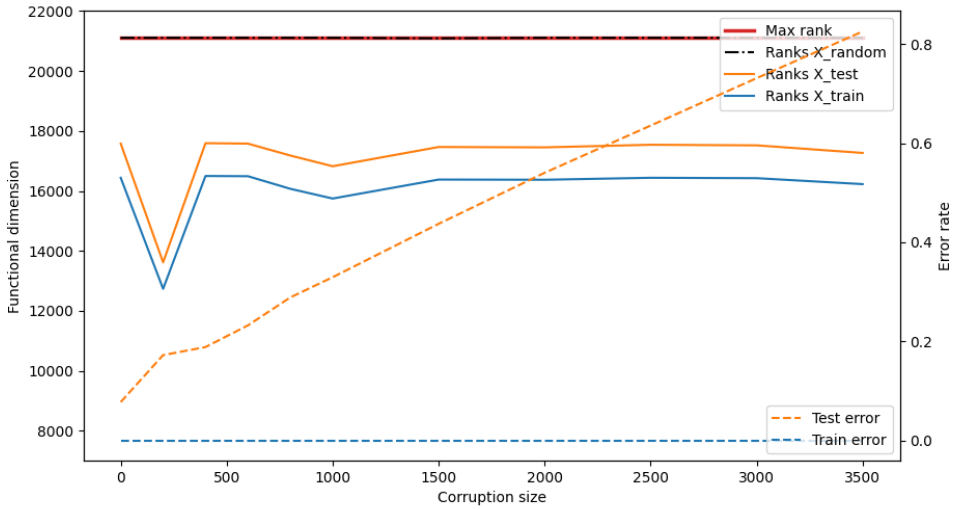

Figure 5.6 – Behavior of different complexity measures as some corrupted labels are introduced in the training set. The values of the different ranks are to be read on the left axis, titled 'functional dimension'. The values of the test and train errors are to be read on the right axis.

5.8 Conclusion and perspectives

In this article, we describe the geometry of deep ReLU neural networks. The study shows that the image of a sample X by deep ReLU neural networks of a fixed architecture is a set whose local dimension varies. The local dimension is called the *batch functional dimension* in [82]. Empirically, the pieces of small dimensions are on the outside of the ones of large dimensions. They are favored by the optimization. We call this phenomenon *geometry induced implicit regularization*. We also study the maximal dimension, when X is allowed to vary. We call it the *computable full functional dimension*. Both notions of local complexity are determined by the activation patterns. We investigate the practical computation of the functional dimensions and provide experiments emphasizing the *geometry induced implicit regularization* and the link between functional dimensions and the distribution of the inputs.

This opens many perspectives in deep learning theory. The formal connection between the notions of local complexity and the generalization gap is still lacking. It would permit us to obtain a theory explaining the good performance of deep learning. It would be interesting to study more systematically how the functional dimensions of the learned parameters depend on the distribution of the learned phenomenon. To do so, it would be interesting to study instances in large dimensions. Algorithms of a better complexity for computing the functional dimensions are needed.

Acknowledgments

This work has benefited from the AI Interdisciplinary Institute ANITI. ANITI is funded by the French "Investing for the Future – PIA3" program under the Grant agreement n°ANR-19-PI3A-0004.

The authors gratefully acknowledge the support of the DEEL project. [7]

7. https://www.deel.ai/

Appendices

5.A Proofs of Section 5.3

5.A.1 Proof of Theorem 103

Let us define, for $\ell \in [\![1, L-1]\!]$ and $v \in V_\ell$ the set

$$\mathcal{T}_v = \left\{ (x,\theta) \in \mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B) \mid \sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x) \right)_{v'} + b_v = 0 \right\}, \quad (5.A.1)$$

and let

$$\mathcal{T} = \cup_{\ell=1}^{L-1} \cup_{v \in V_\ell} \mathcal{T}_v. \quad (5.A.2)$$

Similarly, for any $x \in \mathbb{R}^{N_0}$, $\ell \in [\![1, L-1]\!]$ and $v \in V_\ell$, we define the set

$$\mathcal{T}_v^x = \left\{ \theta \in \mathbb{R}^E \times \mathbb{R}^B \mid \sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x) \right)_{v'} + b_v = 0 \right\},$$

and let

$$\mathcal{T}^x = \cup_{\ell=1}^{L-1} \cup_{v \in V_\ell} \mathcal{T}_v.$$

Lemma 114. *(i) Over $\mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B)$, the function $(x,\theta) \mapsto a(x,\theta) \in \{0,1\}^{N_1+\cdots+N_{L-1}}$ exactly takes $2^{N_1+\cdots+N_{L-1}}$ distinct values.*

For $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, we write

$$A_\delta = \{(x,\theta) \in \mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B) \mid a(x,\theta) = \delta\}. \quad (5.A.3)$$

Then: On A_δ, the function $(x,\theta) \mapsto f_\theta(x)$ is polynomial with degree less than or equal to $L+1$.

The set $\mathcal{T}$ has Lebesgue measure zero and $\cup_{\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}} \partial A_\delta = \mathcal{T}$. Therefore, for any $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, ∂A_δ is a closed set of zero Lebesgue measure in $\mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B)$.

(ii) For any fixed $x \in \mathbb{R}^{N_0}$, the function $\theta \mapsto a(x,\theta)$ exactly takes $2^{N_1+\cdots+N_{L-1}}$ distinct values. For $x \in \mathbb{R}^{N_0}$ and $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, we write

$$A_\delta^x = \{\theta \in \mathbb{R}^E \times \mathbb{R}^B \mid a(x,\theta) = \delta\}.$$

Then: On A_δ^x, the function $\theta \mapsto f_\theta(x)$ is polynomial with a degree less than or equal to L.

The set $\mathcal{T}^x$ has Lebesgue measure zero and $\cup_{\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}} \partial A_\delta^x = \mathcal{T}^x$.

For any $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, ∂A_δ^x is a closed set with zero Lebesgue measure in $\mathbb{R}^E \times \mathbb{R}^B$.

Proof of Lemma 114. To avoid repetitions, we only detail the proof of (i). The proof of (ii) is very similar, considering functions of θ only (with x fixed) rather

than (x, θ), and noting that since x is not a variable, the polynomial functions of θ, similar to the one of (x, θ) considered below, have one degree less.

We first prove that all activation patterns are reached.

The set $\{0,1\}^{N_1+\cdots+N_{L-1}}$ is finite and its cardinal is $2^{N_1+\cdots+N_{L-1}}$. Observe that for any $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, by taking $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ such that $w_{v\to v'} = 0$ for any $(v \to v') \in E$, $b_v = 0$ for $v \in V_L$ and $b_v = (-1)^{1+\delta_v}$ for any $v \in V_1 \cup \cdots \cup V_{L-1}$, then, for any $x \in \mathbb{R}^{N_0}$ and any $v \in V_1 \cup \cdots \cup V_{L-1}$, we have $a_v(x, \theta) = \delta_v$, i.e. $a(x, \theta) = \delta$. We recall the notations $A_\delta, \delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, in (5.A.3).

In order to prove that the function $(x, \theta) \mapsto f_\theta(x)$ is polynomial, we remind the definition of f^l_θ, in (5.2.1), define

$$a_{\leq \ell}(x, \theta) = \begin{cases} (a_v(x, \theta))_{v \in V_1 \cup \cdots \cup V_\ell} & \text{if } \ell \geq 1, \\ 1 & \text{if } \ell = 0, \end{cases}$$

and prove by induction that the assertion

$$H_\ell : \begin{cases} \forall D \subseteq \mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B), \quad \text{if } (x, \theta) \mapsto a_{\leq \ell}(x, \theta) \text{ is constant on } D, \text{ then} \\ (x, \theta) \mapsto f^\ell_\theta(x) \text{ is polynomial on } D \text{ with degree less than or equal to } \ell + 1, \end{cases}$$

holds, for all $\ell \in [\![0, L-1]\!]$.

The assertion H_0 indeed holds because $f^0_\theta(x) = x$ is polynomial in (x, θ) of degree 1 on any subset of $\mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B)$. Assume now that for some $\ell \in [\![1, L-1]\!]$, $H_{\ell-1}$ holds, and let us prove H_ℓ.

Let $D \subseteq \mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B)$ such that $a_{\leq \ell}(x, \theta)$ is constant on D. For $(x, \theta) \in D$ and $v \in V_\ell$, using (5.2.6), we have

$$\left(f^\ell_\theta(x)\right)_v = a_v(x, \theta) \left(\sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f^{\ell-1}_\theta(x)\right)_{v'} + b_v \right).$$

The quantity $a_{\leq \ell-1}(x, \theta)$ is constant on D and thus from $H_{\ell-1}$, for all $v' \in V_{\ell-1}$, $(x, \theta) \mapsto (f^{\ell-1}_\theta(x))_{v'}$ is a polynomial function of (x, θ) on D with degree less than or equal to ℓ. Since $a_v(x, \theta)$ is constant on D, $\left(f^\ell_\theta(x)\right)_v$ is a polynomial function of (x, θ) of degree less than or equal to $\ell + 1$. This concludes the proof by induction that H_ℓ holds for all $\ell \in [\![0, L-1]\!]$.

Let $v \in V_L$. We have

$$(f_\theta(x))_v = \sum_{v' \in V_{L-1}} w_{v' \to v} (f^{L-1}_\theta(x))_{v'} + b_v.$$

For $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, $a_{\leq L-1}(x, \theta) = a(x, \theta)$ is constant on A_δ and thus from H_{L-1}, $(x, \theta) \mapsto f^{L-1}_\theta(x)$ is polynomial of degree less than or equal to L on A_δ, and thus $(x, \theta) \mapsto f_\theta(x)$ is polynomial of degree less than or equal to $L + 1$ on A_δ. This proves the second statement of Lemma 114, (i).

Let us now show that $\mathcal{T}$ has Lebesgue measure zero. For that, let us show that for all $\ell \in [\![1, L-1]\!]$ and $v \in V_\ell$, $\mathcal{T}_v$ has Lebesgue measure zero. To do so, since $\cup_\delta A_\delta =$

$\mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B)$, we consider, $\ell \in [\![1, L-1]\!]$ and $v \in V_\ell$, and prove that, for all $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, $\mathcal{T}_v \cap A_\delta$ has zero Lebesgue measure. Let $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, $a_{\leq \ell-1}(x,\theta)$ is constant on A_j and thus from $H_{\ell-1}$, $(x,\theta) \mapsto f_\theta^{\ell-1}(x)$ is a polynomial function of (x,θ) on A_δ and thus $\sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x)\right)_{v'} + b_v$ also is. Since b_v is not present in $f_\theta^{\ell-1}(x)$, it only appears in a single monomial of degree and coefficient 1 of $\sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x)\right)_{v'} + b_v$. The latter polynomial function is therefore non-constant. Hence the set $\mathcal{T}_v \cap A_\delta$, constituted by the zeros of this polynomial function, has zero Lebesgue measure. Since $\cup_\delta A_\delta = \mathbb{R}^{N_0} \times (\mathbb{R}^E \times \mathbb{R}^B)$, we finally conclude that, for any $\ell \in [\![1, L-1]\!]$ and $v \in V_\ell$, $\mathcal{T}_v$ has Lebesgue measure zero.

The set

$$\mathcal{T} = \cup_{\ell=1}^{L-1} \cup_{v \in V_\ell} \mathcal{T}_v$$

is thus also of zero Lebesgue measure.

Let us now prove the set equality:

$$\bigcup_\delta \partial A_\delta = \mathcal{T}. \tag{5.A.4}$$

We first show the inclusion $\bigcup_\delta \partial A_\delta \subset \mathcal{T}$. Consider $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$ and let us now show that $\partial A_\delta \subset \mathcal{T}$. To do so, consider $(x,\theta) \in \partial A_\delta$. Since $(x,\theta) \notin \mathrm{Int}(A_\delta)$, for any ε there exists $\delta_\varepsilon \neq \delta$ such that $B((x,\theta),\varepsilon) \cap A_{\delta_\varepsilon} \neq \emptyset$. Since the set of all possible δ_ε is finite, we are sure that there exists $\delta' \neq \delta$ such that $(x,\theta) \in \overline{A}_{\delta'}$. Let $\ell \in [\![1, L-1]\!]$ and $v \in V_\ell$ such that $\delta_v \neq \delta'_v$. We assume without loss of generality that $\delta_v = 0$. The proof is indeed similar when $\delta_v = 1$. There exists $(x_n, \theta_n) \in A_\delta^{\mathbb{N}^*}$ such that $(x_n, \theta_n) \to (x,\theta)$ as $n \to \infty$ and there exists $(x'_n, \theta'_n) \in A_{\delta'}^{\mathbb{N}^*}$ such that $(x'_n, \theta'_n) \to (x,\theta)$ as $n \to \infty$. We have $a_v(x_n,\theta_n) = 0$ and $a_v(x'_n,\theta'_n) = 1$ for all n.

Using that $(x,\theta) \mapsto \sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x)\right)_{v'} + b_v$ is continuous and taking the limit of this function at (x_n,θ_n), as n goes to infinity, we obtain that $\sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x)\right)_{v'} + b_v \leq 0$. Reasoning similarly with the sequence $(x'_n, \theta'_n)_{n \in \mathbb{N}^*}$ we obtain the reverse inequality and conclude that $\sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x)\right)_{v'} + b_v = 0$. This shows that $(x,\theta) \in \mathcal{T}_v \subseteq \mathcal{T}$. This finishes the proof of $\partial A_\delta \subseteq \mathcal{T}$.

Let us now show the reciprocal inclusion $\mathcal{T} \subset \bigcup_\delta \partial A_\delta$. Indeed, let $(x,\theta) \in \mathcal{T}$. There exists $v \in V_1 \cup \cdots \cup V_{L-1}$ such that $(x,\theta) \in \mathcal{T}_v$. There also exists $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$ such that $(x,\theta) \in A_\delta$. In particular, since

$$\sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x)\right)_{v'} + b_v = 0,$$

we have $\delta_v = a_v(x,\theta) = 1$. For any $\varepsilon > 0$, by replacing b_v by $b_v - \varepsilon$, we obtain a θ_ε satisfying $\|\theta - \theta_\varepsilon\| \leq \varepsilon$ and $a_v(x,\theta_\varepsilon) = 0 \neq \delta_v$, which shows $(x,\theta_\varepsilon) \notin A_\delta$. This shows $(x,\theta) \in \partial A_\delta \subset \bigcup_\delta \partial A_\delta$. This shows the desired inclusion, and thus the equality (5.A.4).

For all $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$, ∂A_δ is closed by definition of a boundary. Since $\mathcal{T}$ has been shown to have Lebesgue measure zero, then ∂A_δ has Lebesgue measure zero and thus the proof of the part (i) is concluded. □

We state and prove another lemma before proving Theorem 103. The lemma resembles Theorem 103 but does not include the statements on $\operatorname{rank}(\mathbf{D}f_\theta(X))$.

For $n \in \mathbb{N}^*$, reminding the definition $\mathcal{T}$ in (5.A.2), let us now define

$$\mathcal{T}_n = \cup_{i=1}^n \left\{ (X,\theta) \in \mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B) \;\middle|\; (x^{(i)},\theta) \in \mathcal{T} \right\}. \tag{5.A.5}$$

Similarly, for $n \in \mathbb{N}^*$ and $X \in \mathbb{R}^{N_0 \times n}$, we define

$$\mathcal{T}^X = \cup_{i=1}^n \left\{ \theta \in \mathbb{R}^E \times \mathbb{R}^B \mid (x^{(i)},\theta) \in \mathcal{T} \right\}.$$

Lemma 115. *(i) For all $n \in \mathbb{N}^*$, the sets $\widetilde{\mathcal{O}}_1^n, \dots, \widetilde{\mathcal{O}}_{m_n}^n$ defined in (5.3.1) are non-empty, open and disjoint, and they satisfy*

- *$\left(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\right)^c = \mathcal{T}_n$, and in particular the complement $\left(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\right)^c$ is a closed set with zero Lebesgue measure;*
- *for all $j \in [\![1, m_n]\!]$, the function $(X,\theta) \mapsto a(X,\theta)$ is constant on each $\widetilde{\mathcal{O}}_j^n$ and takes m_n distinct values on $\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n$;*
- *for all $j \in [\![1, m_n]\!]$, $(X,\theta) \mapsto f_\theta(X)$ is a polynomial function of degree less than or equal to $L+1$ on $\widetilde{\mathcal{O}}_j^n$.*

(ii) For all $n \in \mathbb{N}^*$, for all $X \in \mathbb{R}^{N_0 \times n}$, the sets $\widetilde{\mathcal{U}}_1^X, \dots, \widetilde{\mathcal{U}}_{p_X}^X$ defined in (5.3.4) are non-empty, open and disjoint, and they satisfy

- *$\left(\cup_{j=1}^{p_X} \widetilde{\mathcal{U}}_j^X\right)^c = \mathcal{T}^X$, and in particular the complement $\left(\cup_{j=1}^{p_X} \widetilde{\mathcal{U}}_j^X\right)^c$ is a closed set with zero Lebesgue measure;*
- *for all $j \in [\![1, p_X]\!]$, the function $\theta \mapsto a(X,\theta)$ is constant on each $\widetilde{\mathcal{U}}_j^X$ and takes p_X distinct values on $\cup_{j=1}^{p_X} \widetilde{\mathcal{U}}_j^X$;*
- *for all $j \in [\![1, p_X]\!]$, $\theta \mapsto f_\theta(X)$ is a polynomial function of degree less than or equal to L on $\widetilde{\mathcal{U}}_j^X$.*

Proof of Lemma 115. As in the proof of Lemma 114, the proofs of (i) and (ii) are very similar. To avoid repetitions, we only detail the proof of (i).

By definition, see (5.3.1), the sets $\widetilde{\mathcal{O}}_1^n, \dots, \widetilde{\mathcal{O}}_{m_n}^n$ are non-empty, open and disjoint. Let us establish that $\mathcal{T}_n$ is of Lebesgue measure zero, before proving the first point of (i).

The characterization of $\mathcal{T}$ in Lemma 114 (i) shows that we have

$$\mathcal{T}_n = \cup_{i=1}^n \cup_{\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}} \left\{ (X,\theta) \in \mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B) \mid (x^{(i)},\theta) \in \partial A_\delta \right\}.$$

Then one can check that since, for all δ, ∂A_δ is of Lebesgue measure zero, $\mathcal{T}_n$ is a finite union of such sets and is, therefore, a set of Lebesgue measure zero in $\mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B)$.

Let us now show the first point of (i). Let us prove that $(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c = \mathcal{T}_n$.

To do so, let us first show that $(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c \subset \mathcal{T}_n$. Let $(X,\theta) \in (\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c$. Consider the $\Delta_1, \dots, \Delta_q$ defined just before (5.3.1). There exists $j \in [\![1, q]\!]$ such that $a(X,\theta) = \Delta_j$. Since $(X,\theta) \notin \widetilde{\mathcal{O}}_j^n$, there exists a sequence $(X_k,\theta_k)_{k\in\mathbb{N}^*}$ such that $(X_k,\theta_k) \to (X,\theta)$, as $k \to \infty$ and $a(X_k,\theta_k) \neq \Delta_j$, for all k. Modulo the extraction of a sub-sequence, we can assume that there exist $i \in [\![1, n]\!]$ and $\delta' \in \{0,1\}^{N_1+\cdots+N_{L-1}}$

such that $a(x_k^{(i)}, \theta_k) = \delta' \neq \delta$, where $x_k^{(i)}$ is the i^{th} column of X_k, and where we denote $\delta = (\Delta_j)_i$. Thus, we have $(x_k^{(i)}, \theta_k) \in A_{\delta'}$, for all k, and $(x^{(i)}, \theta) \in \partial A_\delta$. Finally, $(X, \theta) \in \mathcal{T}_n$. This shows $(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c \subseteq \mathcal{T}_n$.

Let us now show that $(\mathcal{T}_n \subset \cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c$. If $(X, \theta) \in \mathcal{T}_n$, there exists $i \in [\![1, n]\!]$ and $\delta \in \{0,1\}^{N_1+\cdots+N_{L-1}}$ such that $(x^{(i)}, \theta) \in \partial A_\delta$. Thus, for any $\varepsilon > 0$, $(x, \theta') \mapsto a(x, \theta')$ is not constant over $B((x^{(i)}, \theta), \varepsilon)$. As a consequence, (X, θ) does not belong to any of the open sets $\widetilde{\mathcal{O}}_{j'}^n$.

This shows $(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c = \mathcal{T}_n$, and thus, since $\mathcal{T}_n$ is of Lebesgue measure zero, $(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c$ has zero Lebesgue measure. Adding that the complement of an open set is closed, $(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)^c$ is closed, which ends the proof of the first point of (i).

The second point of (i) holds by definition of $\widetilde{\mathcal{O}}_1^n, \dots, \widetilde{\mathcal{O}}_{m_n}^n$.

Let us now show the third point of (i). Let $j \in [\![1, m_n]\!]$. The function $(X, \theta) \mapsto a(X, \theta)$ is constant on $\widetilde{\mathcal{O}}_j^n$. The set $\widetilde{\mathcal{O}}_j^n$ is associated to Δ_j in (5.3.1) and the latter is of the form $(\delta^1, \dots, \delta^n)$ with $\delta^i \in \{0,1\}^{N_1+\cdots+N_{L-1}}$ for $i \in [\![1, n]\!]$. Fix $i' \in [\![1, n]\!]$. Then for $X = (x^{(i)})_{i \in [\![1,n]\!]}$ with $(X, \theta) \in \widetilde{\mathcal{O}}_j^n$, $(x^{(i')}, \theta) \in A_{\delta^{i'}}$. Hence, Lemma 114 (i) shows that $f_\theta(x^{(i')})$ is a polynomial function of $(x^{(i')}, \theta)$ and thus of (X, θ), of degree less than or equal to $L + 1$. The quantity $f_\theta(X)$ is a matrix which columns are $f_\theta(x^{(i)})$, $i \in [\![1, n]\!]$. Hence $(X, \theta) \mapsto f_\theta(X)$ is a polynomial function of degree less than or equal to $L + 1$ on $\widetilde{\mathcal{O}}_j^n$, which concludes the proof of (i). □

Proof of Theorem 103. Again, the proofs of (i) and (ii) are very similar and we only detail the proof of (i).

Consider $n \in \mathbb{N}^*$. The sets $\mathcal{O}_1^n, \dots, \mathcal{O}_{m_n}^n$ are non-empty by definition of the quantities $r_1^n, \dots, r_{m_n}^n$, and they are disjoint because of the inclusion $\mathcal{O}_j^n \subseteq \widetilde{\mathcal{O}}_j^n$ for all j, and because the sets $\widetilde{\mathcal{O}}_1^n, \dots, \widetilde{\mathcal{O}}_{m_n}^n$ are disjoint as shown in Lemma 115. Hence the first item holds. The second item is a direct consequence of the definition of $\mathcal{O}_j^n$, in (5.3.3). The third item holds by definition.

To see that $\mathcal{O}_j^n$ is open, first recall that $\widetilde{\mathcal{O}}_j^n$ is open, then note that since the function $(X, \theta) \mapsto f_\theta(X)$ is polynomial over $\widetilde{\mathcal{O}}_j^n$, the function $(X, \theta) \mapsto \mathbf{D}f_\theta(X)$ is continuous over $\widetilde{\mathcal{O}}_j^n$. Since the rank is lower semicontinuous, if $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r_j^n$, then there exists $\epsilon > 0$ such that for any $(X', \theta') \in B((X, \theta), \epsilon)$, we have $\operatorname{rank}(\mathbf{D}f_{\theta'}(X')) \geq r_j^n$, which by maximality of r_j^n, is equivalent to $\operatorname{rank}(\mathbf{D}f_{\theta'}(X')) = r_j^n$ and to $(X', \theta') \in \mathcal{O}_j^n$. This shows that $\mathcal{O}_j^n$ is open. Hence Item 4 holds.

Item 6 come directly from Lemma 115 and from the inclusion $\mathcal{O}_j^n \subseteq \widetilde{\mathcal{O}}_j^n$.

To finish the proof, we need to prove Item 5, stating that $\left(\cup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c$ is a closed set with Lebesgue measure zero. Let us consider a basis $(e_1, \dots, e_{|E|+|B|})$ of $\mathbb{R}^E \times \mathbb{R}^B$ and a basis $(\varepsilon_1, \dots, \varepsilon_{nN_L})$ of $\mathbb{R}^{N_L \times n}$. For all X, let us write $M_\theta(X)$ for the matrix of the differential $\mathbf{D}f_\theta(X)$ of the function $\theta \mapsto f_\theta(X)$ in these two bases. Then $(X, \theta) \mapsto M_\theta(X)$ is a polynomial function on $\widetilde{\mathcal{O}}_j^n$. Recall the notation $r_j^n = \max_{(X,\theta) \in \widetilde{\mathcal{O}}_j^n} \operatorname{rank}(\mathbf{D}f_\theta(X))$, and let $(X', \theta') \in \widetilde{\mathcal{O}}_j^n$ such that $\operatorname{rank}(\mathbf{D}f_{\theta'}(X')) = r_j^n$. We thus have $\operatorname{rank}(M_{\theta'}(X')) = r_j^n$, and thus there exists a sub-matrix $N_{\theta'}(X')$ of

$M_{\theta'}(X')$, of size $r_j^n \times r_j^n$, such that $\det N_{\theta'}(X') \neq 0$. The function $(X, \theta) \mapsto M_\theta(X)$ is a polynomial function on $\widetilde{\mathcal{O}}_j^n$ and thus $(X, \theta) \mapsto \det(N_\theta(X))$ also is. This latter function is not uniformly zero on $\widetilde{\mathcal{O}}_j^n$ and thus the set of its zeroes, that we write $\mathcal{Y}_j$, is a closed set of zero Lebesgue measure.

For all $(X, \theta) \in \widetilde{\mathcal{O}}_j^n \setminus \mathcal{Y}_j$, we have $\det N_\theta(X) \neq 0$ and thus $\operatorname{rank}(N_\theta(X)) = r_j^n$ and thus $\operatorname{rank}(M_\theta(X)) \geq r_j^n$. We also have $\operatorname{rank}(M_\theta(X)) = \operatorname{rank}(\mathbf{D}f_\theta(X)) \leq r_j^n$ by definition of r_j^n. Hence $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r_j^n$. This shows $\widetilde{\mathcal{O}}_j^n \setminus \mathcal{Y}_j \subset \mathcal{O}_j^n$.

Finally,

$$\begin{aligned}
\left(\cup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c &= \cap_{j=1}^{m_n} \left(\mathcal{O}_j^n\right)^c \\
&\subseteq \cap_{j=1}^{m_n} \left(\widetilde{\mathcal{O}}_j^n \setminus \mathcal{Y}_j\right)^c \\
&= \cap_{j=1}^{m_n} \left(\left(\widetilde{\mathcal{O}}_j^n\right)^c \cup \mathcal{Y}_j\right) \\
&\subseteq \cap_{j=1}^{m_n} \left(\left(\widetilde{\mathcal{O}}_j^n\right)^c \cup \left(\cup_{j'=1}^{m_n} \mathcal{Y}_{j'}\right)\right) \\
&= \left(\cap_{j=1}^{m_n} \left(\widetilde{\mathcal{O}}_j^n\right)^c\right) \cup \left(\cup_{j=1}^{m_n} \mathcal{Y}_j\right) \\
&= \left(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\right)^c \cup \left(\cup_{j=1}^{m_n} \mathcal{Y}_j\right).
\end{aligned}$$

We know from Lemma 115 that $\left(\cup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\right)^c$ has zero Lebesgue measure. Also each $\mathcal{Y}_j$ has zero Lebesgue measure, thus $\cup_{j=1}^{m_n} \mathcal{Y}_j$ has zero Lebesgue measure. Hence, $\left(\cup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c$ has zero Lebesgue measure, which concludes the proof of (i) in the theorem. □

5.A.2 Proof of Proposition 104

Let $\widetilde{\theta} \sim \theta$ as described in Proposition 104. Let us prove the first item.

By definition of the relation $\sim$, in Section 5.2, there is an invertible linear map $M : \mathbb{R}^E \times \mathbb{R}^B \to \mathbb{R}^E \times \mathbb{R}^B$ such that $\widetilde{\theta} = M\theta$. Note that when expressed in the canonical basis of $\mathbb{R}^E \times \mathbb{R}^B$, the matrix corresponding to M is the product of a permutation matrix and a diagonal matrix, with strictly positive diagonal components whose values are given by (5.2.4) and (5.2.5). Notice that since M corresponds to positive rescalings and neuron permutations, as discussed after (5.2.5), we have,

$$\text{for any } \theta' \in \mathbb{R}^E \times \mathbb{R}^B, \qquad f_{\theta'}(X) = f_{M\theta'}(X). \tag{5.A.6}$$

For all $u \in \mathbb{R}^E \times \mathbb{R}^B$, the following calculation holds because M is invertible, because of (5.A.6) and because, using Item 6 of Theorem 103 (ii), $\theta' \mapsto f_{\theta'}(X)$ is differentiable at θ and we can use (5.2.3),

$$\begin{aligned}
f_{\widetilde{\theta}+u}(X) = f_{M\theta+u}(X) &= f_{M(\theta+M^{-1}u)}(X) \\
&= f_{\theta+M^{-1}u}(X) \\
&= f_\theta(X) + \mathbf{D}f_\theta(X)(M^{-1}u) + o(\|M^{-1}u\|) \\
&= f_\theta(X) + \mathbf{D}f_\theta(X)(M^{-1}u) + o(\|u\|).
\end{aligned}$$

Hence, $\theta' \mapsto f_{\theta'}(X)$ is differentiable at $\widetilde{\theta}$ and for all $u \in \mathbb{R}^E \times \mathbb{R}^B$,

$$\mathbf{D}f_{\widetilde{\theta}}(X)(u) = \mathbf{D}f_{\theta}(X)(M^{-1}u).$$

Since M^{-1} is invertible, it follows that $\operatorname{rank}\big(\mathbf{D}f_{\widetilde{\theta}}(X)\big) = \operatorname{rank}(\mathbf{D}f_{\theta}(X))$. This concludes the proof of the first item.

Let $\theta' \sim_s \theta$ as described in Proposition 104. It is well-known, see for instance [26, Proposition 39], that since $\theta' \sim_s \theta$, the activations of the networks parameterized by θ and θ', for the input X, are the same, i.e. $a(X, \theta') = a(X, \theta)$. Since $\theta \in \widetilde{\mathcal{U}}_j^X$, using (5.3.4), we obtain

$$\theta' \in \{\theta'' \in \mathbb{R}^E \times \mathbb{R}^B \mid a(X, \theta'') = \Delta_j^X\}.$$

By definition $\widetilde{\mathcal{U}}_j^X$ is the interior of the above set. It contains θ. Using again that $\theta' = M\theta$ and that M is invertible, we obtain $\theta' \in \widetilde{\mathcal{U}}_j^X$. Similarly, if $(X, \theta) \in \widetilde{\mathcal{O}}_{j'}^n$, then $(X, \theta') \in \widetilde{\mathcal{O}}_{j'}^n$.

This concludes the proof.

5.B Proofs of Section 5.5

5.B.1 Proof of Proposition 106

Before proving Proposition 106, we state and prove a lemma connecting the sets $\widetilde{\mathcal{O}}_j^n$, defined in (5.3.1), and $\mathcal{X}_\theta^n$, defined in (5.5.3).

Lemma 116. *Let $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$, and let $n \in \mathbb{N}^*$. We have*

$$\mathcal{X}_\theta^n = \{X \in \mathbb{R}^{N_0 \times n} \mid (X, \theta) \in \bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\}.$$

Proof. Consider $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and $n \in \mathbb{N}^*$ and let us first prove that

$$\{X \in \mathbb{R}^{N_0 \times n} \mid (X, \theta) \in \bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\} \subseteq \mathcal{X}_\theta^n.$$

Let $X = (x^{(i)})_{i \in [\![1,n]\!]} \in \mathbb{R}^{N_0 \times n}$ and let $j \in [\![1, m_n]\!]$ such that $(X, \theta) \in \widetilde{\mathcal{O}}_j^n$. Denote $\delta^1, \dots, \delta^n \in \{0,1\}^{N_1 + \cdots + N_{L-1}}$ such that $X \in \prod_{i=1}^n D_{\delta^i}(\theta)$. Using Lemma 115, (i), Item 2, we have

$$X \in \{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\} \subseteq \prod_{i=1}^n D_{\delta^i}(\theta).$$

Moreover, $\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\}$ is open, since $\widetilde{\mathcal{O}}_j^n$ is open, and therefore

$$X \in \operatorname{Int}\left(\prod_{i=1}^n D_{\delta^i}(\theta)\right) = \prod_{i=1}^n \operatorname{Int}(D_{\delta^i}(\theta)).$$

Using the definition of $\mathcal{X}_\theta^n$, in (5.5.3), we conclude that $X \in \mathcal{X}_\theta^n$. This concludes the proof of the first inclusion.

Before proving the converse inclusion, let us establish that $\mathcal{T}_n \subseteq \left(\bigcup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c$, where the open sets $\mathcal{O}_1^n, \dots, \mathcal{O}_{m_n}^n$ are as in Theorem 103. To do so, since using Lemma 115, (i), Item 1, $\mathcal{T}_n = \left(\bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\right)^c$, it suffices to prove that for all $j \in [\![1, m_n]\!]$, there exists $j' \in [\![1, m_n]\!]$ such that $\mathcal{O}_j^n \subseteq \widetilde{\mathcal{O}}_{j'}^n$. Let $j \in [\![1, m_n]\!]$. Theorem 103 states that $(X, \theta) \mapsto a(X, \theta)$ is constant over $\mathcal{O}_j^n$. If we denote Δ this constant value, since $\mathcal{O}_j^n$ is open, we have $\mathcal{O}_j^n \subseteq \operatorname{Int}\left\{(X, \theta) \in \mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B) \mid a(X, \theta) = \Delta\right\}$. Using that $\mathcal{O}_j^n$ is non-empty and recalling the definition of $\widetilde{\mathcal{O}}_{j'}^n$ in (5.3.1), there exists $j' \in [\![1, m_n]\!]$ such that $\mathcal{O}_j^n \subseteq \widetilde{\mathcal{O}}_{j'}^n$. This being true for any $j \in [\![1, m_n]\!]$, it shows that $(\bigcup_{j=1}^{m_n} \mathcal{O}_j^n) \subseteq (\bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n)$. This concludes the proof of the inclusion

$$\mathcal{T}_n = \left(\bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\right)^c \subset \left(\bigcup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c \tag{5.B.1}$$

Let us now prove the inclusion $\mathcal{X}_\theta^n \subseteq \{X \in \mathbb{R}^{N_0 \times n} \mid (X, \theta) \in \bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\}$.

Let $X = (x^{(i)})_{i \in [\![1,n]\!]} \in \mathcal{X}_\theta^n$. Since, using Lemma 115, (i), $\left(\bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\right)^c = \mathcal{T}_n$, where $\mathcal{T}_n$ is defined in (5.A.5), proving that $(X, \theta) \in \bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n$ is equivalent to proving that $(X, \theta) \notin \mathcal{T}_n$.

Assume by contradiction that $(X, \theta) \in \mathcal{T}_n$. There exists $i \in [\![1, n]\!]$ and $v \in V_1 \cup \dots \cup V_{L-1}$ such that $(x^{(i)}, \theta) \in \mathcal{T}_v$, which means that

$$\sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x^{(i)})\right)_{v'} + b_v = 0.$$

Since $X \in \mathcal{X}_\theta^n$, we have $x^{(i)} \in \mathcal{X}_\theta$, and there exists $\delta \in A(\theta)$ such that $x^{(i)} \in \operatorname{Int}(D_\delta(\theta))$. Let us show that this implies $\operatorname{Int}(D_\delta(\theta)) \times \{\theta\} \subset \mathcal{T}_v$. Indeed, the function

$$x \longmapsto \sum_{v' \in V_{\ell-1}} w_{v' \to v} \left(f_\theta^{\ell-1}(x)\right)_{v'} + b_v$$

is affine over the open set $\operatorname{Int}(D_\delta(\theta))$. If it is not constantly equal to zero over this set, then since its value at $x^{(i)}$ is zero, it takes both positive and negative values over $\operatorname{Int}(D_\delta(\theta))$, and thus $a_v(x, \theta)$ is not constant over $\operatorname{Int}(D_\delta(\theta))$. This contradicts the definition of $D_\delta(\theta)$. Thus, the function is constantly equal to zero on $\operatorname{Int}(D_\delta(\theta))$ and, using the definition of $\mathcal{T}_v$, in (5.A.1), $\operatorname{Int}(D_\delta(\theta)) \times \{\theta\} \subseteq \mathcal{T}_v$. Therefore, $\operatorname{Int}(D_\delta(\theta)) \times \{\theta\} \subset \mathcal{T}$ and, using the definition of $\mathcal{T}_n$ in (5.A.5) and (5.B.1)

$$\left(\left\{X' \in \mathbb{R}^{N_0 \times n} \mid (x')^{(i)} \in \operatorname{Int}(D_\delta(\theta))\right\} \times \{\theta\}\right) \subseteq \mathcal{T}_n \subseteq \left(\bigcup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c.$$

Therefore, $\left\{X' \in \mathbb{R}^{N_0 \times n} \mid (x')^{(i)} \in \operatorname{Int}(D_\delta(\theta))\right\} \subseteq z_n(\theta)$, where $z_n(\theta)$ is defined in (5.5.4). Since $\operatorname{Int}(D_\delta(\theta))$ is non-empty, the Lebesgue measure of

$$\left\{X' \in \mathbb{R}^{N_0 \times n} \mid (x')^{(i)} \in \operatorname{Int}(D_\delta(\theta))\right\}$$

is not zero and therefore $\theta \in \mathcal{Z}_n$, as defined in (5.5.5). This contradicts the hypothesis on θ and finishes the proof of the statement $(X, \theta) \notin \mathcal{T}_n$.

This concludes the proof of the inclusion $\mathcal{X}_\theta^n \subseteq \{X \in \mathbb{R}^{N_0 \times n} \mid (X, \theta) \in \bigcup_{j=1}^{m_n} \widetilde{\mathcal{O}}_j^n\}$ and finishes the proof of Lemma 116. □

Proof of Proposition 106. We remind that for all $n \in \mathbb{N}^*$ and all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$,

$$z_n(\theta) = \left\{ X \in \mathbb{R}^{N_0 \times n} \;\middle|\; (X, \theta) \in \left(\cup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c \right\}$$

and

$$\mathcal{Z}_n = \left\{ \theta \in \mathbb{R}^E \times \mathbb{R}^B \;\middle|\; z_n(\theta) \text{ has strictly positive Lebesgue measure} \right\}$$

and $\mathcal{Z} = \cup_{n \in \mathbb{N}^*} \mathcal{Z}_n$.

Let us first prove (i).

Let us write λ for Lebesgue measure on $\mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B)$, and λ^1 for Lebesgue measure on $\mathbb{R}^{N_0 \times n}$. Note that $\left(\cup_{j=1}^{m_n} \mathcal{O}_j^n\right)^c$ is measurable in $\mathbb{R}^{N_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B)$, as a closed set, and thus for all $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, $z_n(\theta)$ is measurable in $\mathbb{R}^{N_0 \times n}$. With similar arguments, the function $\theta \mapsto \lambda^1(z_n(\theta))$ is measurable on $\mathbb{R}^E \times \mathbb{R}^B$ as the integral with respect to X of a measurable function of X and θ. Hence, $\mathcal{Z}_n$, as the set where this function is strictly positive, is indeed measurable.

Let $n \in \mathbb{N}^*$. We first prove that $\mathcal{Z}_n$ has zero Lebesgue measure on $\mathbb{R}^E \times \mathbb{R}^B$. Let us assume by contradiction that $\mathcal{Z}_n$ has strictly positive Lebesgue measure.

Let us write $\mathcal{C} = \cup_{\theta \in \mathcal{Z}_n} (z_n(\theta) \times \{\theta\})$. For all θ, for all $X \in z_n(\theta)$, $(X, \theta) \in \left(\cup_{j=1}^{m} \mathcal{O}_j^n\right)^c$. Hence, we have, for all θ, $z_n(\theta) \times \{\theta\} \subseteq \left(\cup_{j=1}^{m} \mathcal{O}_j^n\right)^c$, and thus $\mathcal{C} \subseteq \left(\cup_{j=1}^{m} \mathcal{O}_j^n\right)^c$. We have

$$\begin{aligned}
\lambda(\mathcal{C}) &= \int_{\mathbb{R}^E \times \mathbb{R}^B} \int_{\mathbb{R}^{N_0 \times n}} 1_{\mathcal{C}}(X, \theta) dX d\theta \\
&= \int_{\mathcal{Z}_n} \int_{z_n(\theta)} 1 dX d\theta \\
&= \int_{\mathcal{Z}_n} \lambda^1(z_n(\theta)) d\theta \\
&> 0,
\end{aligned}$$

as the integral of the strictly positive function $\lambda^1(z_n(\theta))$, on the set $\mathcal{Z}_n$ with non-zero Lebesgue measure, is strictly positive (note that since all the functions in the above display are non-negative and measurable, their integrability is guaranteed). This is in contradiction with the fact that $\mathcal{C} \subseteq \left(\cup_{j=1}^{m} \mathcal{O}_j^n\right)^c$, since $\left(\cup_{j=1}^{m} \mathcal{O}_j^n\right)^c$ has zero Lebesgue measure from Theorem 103 (i).

This concludes the proof of the statement $\mathcal{Z}_n$ has zero Lebesgue measure on $\mathbb{R}^E \times \mathbb{R}^B$ and concludes the proof of (i).

The item (ii) is an immediate consequence of (i), since $\mathcal{Z}$ is a countable union of measurable sets of measure 0.

Let us now prove (iii).

For any $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$, $n \in \mathbb{N}^*$, and $X \in \mathcal{X}_\theta^n$, Lemma 116 shows that there exists $j \in [\![1, m_n]\!]$ such that $(X, \theta) \in \widetilde{\mathcal{O}}_j^n$. Lemma 115 (i) Item 3 shows that $(X', \theta') \mapsto f_{\theta'}(X')$ is a polynomial function on the open set $\widetilde{\mathcal{O}}_j^n$. Its restriction to the open neighborhood $\{\theta' \in \mathbb{R}^E \times \mathbb{R}^B \mid (X, \theta') \in \widetilde{\mathcal{O}}_j^n\}$ of θ is also a polynomial function. In particular, $\theta' \mapsto f_{\theta'}(X)$ is differentiable at θ. □

5.B.2 Proof of Proposition 107

Consider $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and $n \in \mathbb{N}^*$ and let us first prove that $r_n^*(\theta) \geq \max_{j \in I_n(\theta)} r_j^n$.

Consider $j \in I_n(\theta)$ such that $r_j^n = \max_{j' \in I_n(\theta)} r_{j'}^n$. Because of the definition of $I_n(\theta)$, in (5.5.8), there exists $X \in \mathbb{R}^{N_0 \times n}$ such that $(X, \theta) \in \mathcal{O}_j^n$. As a first consequence, using Theorem 103, (i) third item, we know that $r_j^n = \operatorname{rank}(\mathbf{D} f_\theta(X))$. As a second consequence, since $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and $\mathcal{O}_j^n \subset \widetilde{\mathcal{O}}_j^n$ we can use Lemma 116, at the beginning of Section 5.B.1, and we have $X \in \mathcal{X}_\theta^n$.

This leads to the conclusion that

$$\max_{j' \in I_n(\theta)} r_{j'}^n = r_j^n = \operatorname{rank}(\mathbf{D} f_\theta(X)) \leq \max_{X' \in \mathcal{X}_\theta^n} \operatorname{rank}\big(\mathbf{D} f_\theta(X')\big) = r_n^*(\theta).$$

Let us now prove that $r_n^*(\theta) \leq \max_{j \in I_n(\theta)} r_j^n$.

Because the rank can only take a finite number of values and considering the definition of $r_n^*(\theta)$ in (5.5.6), we know there exists $X = (x^{(i)})_{i \in [\![1,n]\!]} \in \mathcal{X}_\theta^n$ such that

$$\operatorname{rank}(\mathbf{D} f_\theta(X)) = r_n^*(\theta).$$

Notice that if we were certain that $(X, \theta) \in \mathcal{O}_j^n$, for some $j \in [\![1, m_n]\!]$, the conclusion would be immediate. It might however occur that $(X, \theta) \in \big(\cup_{j=1}^{m_n} \mathcal{O}_j^n\big)^c$. In the remaining of the proof, we use X to construct j and X' such that $(X', \theta) \in \mathcal{O}_j^n$ and $\operatorname{rank}(\mathbf{D} f_\theta(X')) \geq r_n^*(\theta)$.

Since $\theta \notin \mathcal{Z}$ and $X \in \mathcal{X}_\theta^n$, we know thanks to Lemma 116 that there exists $j \in [\![1, m_n]\!]$ such that $(X, \theta) \in \widetilde{\mathcal{O}}_j^n$. Lemma 115 shows that $(X', \theta') \longmapsto f_{\theta'}(X')$ is polynomial over $\widetilde{\mathcal{O}}_j^n$. Hence, $(X', \theta') \longmapsto \mathbf{D} f_{\theta'}(X')$ is well defined and polynomial over $\widetilde{\mathcal{O}}_j^n$, and in particular, $X' \longmapsto \mathbf{D} f_\theta(X')$ is polynomial over the set $\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\}$.

Since by definition we have $\mathcal{O}_j^n \subseteq \widetilde{\mathcal{O}}_j^n$, we have the inclusion

$$\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \mathcal{O}_j^n\} \quad \subseteq \quad \{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\}. \tag{5.B.2}$$

Let us prove that the set on the left of this inclusion is open and non-empty. First, both sets are open since $\mathcal{O}_j^n$ and $\widetilde{\mathcal{O}}_j^n$ are open, and the set on the right is non-empty since it contains X. Assume by contradiction that the set on the left is empty. Since for all $j' \in [\![1, m_n]\!] \setminus \{j\}$, we have $\mathcal{O}_{j'}^n \cap \widetilde{\mathcal{O}}_j^n = \emptyset$, this means, recalling the definition of $z_n(\theta)$ in (5.5.4), that we have

$$\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\} \quad \subseteq \quad \Big\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \big(\cup_{j=1}^{m_n} \mathcal{O}_j^n\big)^c\Big\} = z_n(\theta).$$

Since, as we have just shown, the set $\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\}$ is open and non-empty, this means that $z_n(\theta)$ has positive Lebesgue measure, and thus, using (5.5.5), that $\theta \in \mathcal{Z}_n$. This contradicts the hypothesis $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$. This concludes the proof establishing that the open set $\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \mathcal{O}_j^n\}$ is non-empty.

Similarly to what was done in the proof of Theorem 103, we consider a basis $(e_1, \dots, e_{|E|+|B|})$ of $\mathbb{R}^E \times \mathbb{R}^B$ and a basis $(\varepsilon_1, \dots, \varepsilon_{nN_L})$ of $\mathbb{R}^{N_L \times n}$. For all θ' and all $X' \in \mathbb{R}^{N_0 \times n}$ such that $\mathbf{D}f_{\theta'}(X')$ is well defined, we write $Jf_{\theta'}(X')$ for the matrix of $\mathbf{D}f_{\theta'}(X')$ in these two bases. Since $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r_n^*(\theta)$, there exists a sub-matrix $N_\theta(X)$ of $Jf_\theta(X)$, of size $r_n^*(\theta) \times r_n^*(\theta)$, such that $\det(N_\theta(X)) \neq 0$. Since $X' \mapsto \mathbf{D}f_\theta(X')$ is polynomial over the set $\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\}$, we conclude that $X' \mapsto \det(N_\theta(X'))$ coincides with a polynomial over the set $\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \widetilde{\mathcal{O}}_j^n\}$. Since this set contains X, for which we have $\det(N_\theta(X)) \neq 0$, the polynomial is non-zero and the set containing all its roots is therefore closed and of Lebesgue measure zero in $\mathbb{R}^{N_0 \times n}$. The set of the roots cannot contain the non-empty open subset $\{X' \in \mathbb{R}^{N_0 \times n} \mid (X', \theta) \in \mathcal{O}_j^n\}$, which shows that there exists $X' \in \mathbb{R}^{N_0 \times n}$ such that $(X', \theta) \in \mathcal{O}_j^n$ and $\det(N_\theta(X')) \neq 0$.

We conclude, using Theorem 103 (i), that

$$r_j^n = \operatorname{rank}\big(\mathbf{D}f_\theta(X')\big) \geq \operatorname{rank}\big(N_\theta(X')\big) = r_n^*(\theta).$$

Finally, we obtain

$$r_n^*(\theta) \leq \max_{j \in I_n(\theta)} r_j^n$$

which concludes the proof.

5.B.3 Proof of Theorem 108

We begin the proof with a lemma. To state the lemma, we define a new notation. For any $n \in \mathbb{N}^*$ and any $j \in [\![1, m_n]\!]$, Theorem 103 (i), second item, ensures that the mapping $(X, \theta) \mapsto a(X, \theta)$, is constant on $\mathcal{O}_j^n$. We denote its value $\delta^{j,n} = (\delta_i^{j,n})_{i \in [\![1,n]\!]}$. We therefore have, for all $(X, \theta) = ((x^{(i)})_{i \in [\![1,n]\!]}, \theta) \in \mathcal{O}_j^n$,

$$a(x^{(i)}, \theta) = \delta_i^{j,n} \in \{0,1\}^{N_1 + \dots + N_{L-1}} \qquad \text{, for all } i \in [\![1, n]\!].$$

Lemma 117. *For any $n \in \mathbb{N}^*$, $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, and $j \in [\![1, m_n]\!]$, we have*

$$j \in I_n(\theta) \qquad \Longrightarrow \qquad \forall i \in [\![1, n]\!], \quad \delta_i^{j,n} \in A(\theta).$$

Conversely, for any $n \in \mathbb{N}^*$, $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$, and $j \in [\![1, m_n]\!]$, we have

$$j \in I_n(\theta) \qquad \Longleftarrow \qquad \forall i \in [\![1, n]\!], \quad \delta_i^{j,n} \in A(\theta).$$

Proof. Consider $n \in \mathbb{N}^*$, $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and $j \in [\![1, m_n]\!]$.

Let us first prove that if $j \in I_n(\theta)$ then, for all $i \in [\![1, n]\!]$, $\delta_i^{j,n} \in A(\theta)$. To do so, assume $j \in I_n(\theta)$ and consider $i \in [\![1, n]\!]$. We want to prove that $\delta_i^{j,n} \in A(\theta)$. Given the definition of $A(\theta)$, in (5.5.2), it is sufficient to prove $\operatorname{Int}(D_{\delta_i^{j,n}}(\theta)) \neq \emptyset$.

Due to Theorem 103, (i), second item and the above definition of $(\delta_i^{j,n})_{i\in[\![1,n]\!]}$, we have for all $(X',\theta') = (((x')^{(i)})_{i\in[\![1,n]\!]},\theta') \in \mathcal{O}_j^n$,

$$a((x')^{(i)},\theta') = \delta_i^{j,n}. \tag{5.B.3}$$

Since $j \in I_n(\theta)$ and $i \in [\![1,n]\!]$, there exists $(X,\theta) = ((x^{(i)})_{i\in[\![1,n]\!]},\theta) \in \mathcal{O}_j^n$. Moreover, since $\mathcal{O}_j^n$ is open

$$\{X' \in \mathbb{R}^{N_0\times n} \mid (X',\theta) \in \mathcal{O}_j^n\}$$

is open and, since it contains X, non-empty. We also have using (5.B.3)

$$\{X' \in \mathbb{R}^{N_0\times n} \mid (X',\theta) \in \mathcal{O}_j^n\} \quad \subset \quad D_{\delta_i^{j,n}}(\theta)$$

and conclude that $\mathrm{Int}(D_{\delta_i^{j,n}}(\theta)) \neq \emptyset$ and therefore $\delta_i^{j,n} \in A(\theta)$. This finishes the proof of the first implication.

Consider $n \in \mathbb{N}^*$, $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and $j \in [\![1,m_n]\!]$.

Let us now prove that if, for all $i \in [\![1,n]\!]$, $\delta_i^{j,n} \in A(\theta)$, then $j \in I_n(\theta)$. To do so, assume that for all $i \in [\![1,n]\!]$, $\delta_i^{j,n} \in A(\theta)$. Considering the definition of $I_n(\theta)$, in (5.5.8), it suffices to put to evidence $X \in \mathbb{R}^{N_0\times n}$ such that $(X,\theta) \in \mathcal{O}_j^n$ to prove that $j \in I_n(\theta)$.

Since, for all $i \in [\![1,n]\!]$, $\delta_i^{j,n} \in A(\theta)$, using the definition of $A(\theta)$ in (5.5.2), $\mathrm{Int}(D_{\delta_i^{j,n}}(\theta)) \neq \emptyset$. Therefore, $\Pi_{i=1}^n \mathrm{Int}(D_{\delta_i^{j,n}}(\theta)) \subset \mathbb{R}^{N_0\times n}$ is a non-empty open set. Moreover, since $\theta \notin \mathcal{Z}$,

$$\{X \in \mathbb{R}^{N_0\times n} \mid (X,\theta) \in (\cup_{j=1}^{m_n}\mathcal{O}_j^n)^c\}$$

has zero Lebesgue measure in $\mathbb{R}^{N_0\times n}$. Therefore, since $\Pi_{i=1}^n \mathrm{Int}(D_{\delta_i^{j,n}}(\theta))$ is a non-empty open set, there exists $j' \in [\![1,m_n]\!]$ such that

$$\Pi_{i=1}^n \mathrm{Int}(D_{\delta_i^{j,n}}(\theta)) \quad \cap \quad \{X \in \mathbb{R}^{N_0\times n} \mid (X,\theta) \in \mathcal{O}_{j'}^n\} \quad \neq \quad \emptyset.$$

Consider X in this set, we have

$$a(X,\theta) = \delta^{j,n} = \delta^{j',n}.$$

Using Theorem 103, (i), second item, we conclude that $j = j'$. Finally, $(X,\theta) \in \mathcal{O}_j^n$ and $j \in I_n(\theta)$. This concludes the proof.

□

We deduce from Lemma 117 the following result.

Lemma 118. *For any θ and θ' in $(\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$*

$$A(\theta) = A(\theta') \quad \iff \quad \forall n \in \mathbb{N}^*, \quad I_n(\theta) = I_n(\theta').$$

Proof. Consider θ and θ' in $(\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$.

Let us first prove the implication $\Rightarrow$.

Assume $A(\theta) = A(\theta')$ and consider $n \in \mathbb{N}^*$. We want to prove that $I_n(\theta) = I_n(\theta')$. Let $j \in I_n(\theta)$, using Lemma 117, we know that for all $i \in [\![1, n]\!]$, $\delta_i^{j,n} \in A(\theta)$. Therefore, for all $i \in [\![1, n]\!]$, $\delta_i^{j,n} \in A(\theta')$ and using Lemma 117 again, we obtain that $j \in I_n(\theta')$. As a conclusion, $I_n(\theta) \subseteq I_n(\theta')$. We finish the proof using the fact that θ and θ' are interchangeable in the statement of Lemma 118.

Let us now prove the implication $\Leftarrow$.

Assume that for all $n \in \mathbb{N}^*$, $I_n(\theta) = I_n(\theta')$. We prove below that $A(\theta) \subset A(\theta')$. The conclusion then follows by remarking that θ and θ' are interchangeable in the statement of Lemma 118.

Let $\delta \in A(\theta)$, using the definition of $A(\theta)$, in (5.5.2), we know that there exists $x \in \mathrm{Int}(D_\delta(\theta)) \subseteq \mathbb{R}^{N_0}$ such that $a(x, \theta) = \delta$. Since $\theta \notin \mathcal{Z}_1$, we know that $\{x \in \mathbb{R}^{1\times N_0} \mid (x, \theta) \in \left(\cup_{j=1}^{m_1} \mathcal{O}_j^1\right)^c\}$ is of Lebesgue measure zero. Its intersection with the non-empty open set $\mathrm{Int}(D_\delta(\theta))$ is therefore of Lebesgue measure zero and there exists $j' \in I_1(\theta)$ and $x' \in \mathrm{Int}(D_\delta(\theta))$ such that $(x', \theta) \in \mathcal{O}_{j'}^1$. We therefore have $\delta = a(x', \theta) = \delta_1^{j,1}$. Using $I_1(\theta) = I_1(\theta')$ and Lemma 117, we conclude that $\delta \in A(\theta')$. As a conclusion $A(\theta) \subseteq A(\theta')$.

This concludes the proof of the lemma. □

Proof of Theorem 108. Consider θ and θ' in $(\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and assume $A(\theta) \subseteq A(\theta')$.

Let $n \in \mathbb{N}^*$ and $j \in I_n(\theta)$. Using Lemma 117, we know that for all $i \in [\![1, n]\!]$, $\delta_i^{j,n} \in A(\theta)$, and therefore for all $i \in [\![1, n]\!]$, $\delta_i^{j,n} \in A(\theta')$. Using Lemma 117 again, we obtain that $j \in I_n(\theta')$. As a conclusion, for all $n \in \mathbb{N}^*$, $I_n(\theta) \subseteq I_n(\theta')$.

Using Proposition 107, we obtain that for all $n \in \mathbb{N}^*$, $r_n^*(\theta) \leq r_n^*(\theta')$. We conclude, using the definition of r^*, in (5.5.6), that $r^*(\theta) \leq r^*(\theta')$.

This concludes the proof of the first statement of Theorem 108.

The second statement follows by applying the first statement twice. Once with the hypothesis $A(\theta) \subseteq A(\theta')$ and a second time with the hypothesis $A(\theta') \subseteq A(\theta)$. This concludes the proof of the theorem. □

5.B.4 Proof of Proposition 109

Let $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ and let $\widetilde{\theta} \sim \theta$. For $n \in \mathbb{N}^*$, let us show that $z_n(\theta) = z_n(\widetilde{\theta})$. For that, let us consider $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\theta' \sim_s \theta$ and $\widetilde{\theta} \sim_p \theta'$.

Let $X \notin z_n(\theta)$. Then, there is $j \in [\![1, m_n]\!]$ such that $(X, \theta) \in \mathcal{O}_j^n$. Then from Proposition 104 and Theorem 103, we have $\mathrm{rank}(\mathbf{D}f_{\theta'}(X)) = \mathrm{rank}(\mathbf{D}f_\theta(X)) = r_j^n$ and $(X, \theta') \in \widetilde{\mathcal{O}}_j^n$. Hence $(X, \theta') \in \mathcal{O}_j^n$ from (5.3.3) and thus $X \notin z_n(\theta')$. Hence $z_n(\theta') \subseteq z_n(\theta)$.

Now, let $X \notin z_n(\theta')$. There exists $j \in [\![1, m_n]\!]$ such that $(X, \theta') \in \mathcal{O}_j^n$. The function $(X'', \theta'') \mapsto a(X'', \theta'')$ is locally constant at (X, θ'), since $(X, \theta') \in \mathcal{O}_j^n$. Hence this function is also locally constant at $(X, \widetilde{\theta})$. Indeed, permuting the neurons yields the same permutation of the activation values. Hence $(X, \widetilde{\theta}) \in \widetilde{\mathcal{O}}_{j'}^n$ for some $j' \in [\![1, m_n]\!]$. Next, the function $(X'', \theta'') \longmapsto \mathrm{rank}(\mathbf{D}f_{\theta''}(X''))$ is locally constant

at (X, θ'), because $\mathcal{O}_j^n$ is open. Hence the function $(X'', \theta'') \longmapsto \operatorname{rank}(\mathbf{D}f_{\theta''}(X''))$ is also locally constant at $(X, \widetilde{\theta})$, with the same constant value as the previous function, from Proposition 104. Thus, $\operatorname{rank}\big(\mathbf{D}f_{\widetilde{\theta}}(X)\big) = r_{j'}^n$ because $r_{j'}^n$ is the only value that can be taken on a subset of $\widetilde{\mathcal{O}}_{j'}^n$ of non-zero Lebesgue measure. Hence $(X, \widetilde{\theta}) \in \mathcal{O}_{j'}^n$ and thus $X \notin z_n(\widetilde{\theta})$. Hence $z_n(\widetilde{\theta}) \subseteq z_n(\theta')$, and thus $z_n(\widetilde{\theta}) \subseteq z_n(\theta)$.

Finally, swapping the roles of θ and $\widetilde{\theta}$, we have $z_n(\theta) = z_n(\widetilde{\theta})$ for $\theta \sim \widetilde{\theta}$. From this, it follows directly that $\theta \in \mathcal{Z} \iff \widetilde{\theta} \in \mathcal{Z}$. This concludes the proof of the first statement of Proposition 109.

We now prove the second statement. Consider $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and $\widetilde{\theta} \sim \theta$. Let us fix $n \in \mathbb{N}^*$ and show that $r_n^*(\theta) = r_n^*(\widetilde{\theta})$. We have, using Proposition 107, for any $\theta'' \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$

$$\begin{aligned} r_n^*(\theta'') &= \max_{\substack{j \in [\![1, m_n]\!] \\ \exists X \in \mathbb{R}^{N_0 \times n} | (X, \theta'') \in \mathcal{O}_j^n}} r_j^n \\ &= \max_{\substack{j \in [\![1, m_n]\!] \\ \exists X \in \mathbb{R}^{N_0 \times n} | (X, \theta'') \in \widetilde{\mathcal{O}}_j^n}} \max_{(X, \theta'') \in \widetilde{\mathcal{O}}_j^n} \operatorname{rank}(\mathbf{D}f_{\theta''}(X)), \end{aligned} \tag{5.B.4}$$

where we have used also (5.3.2) for the last equality. Consider j_0 and X that reach the above maximum when $\theta'' = \theta$, with $(X, \theta) \in \widetilde{\mathcal{O}}_{j_0}^n$ and $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r_n^*(\theta)$.

Since $\widetilde{\theta} \sim \theta$, there exists $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$ such that $\theta' \sim_s \theta$ and $\widetilde{\theta} \sim_p \theta'$.

From Proposition 104, we have $(X, \theta') \in \widetilde{\mathcal{O}}_{j_0}^n$ and $\operatorname{rank}(\mathbf{D}f_{\theta'}(X)) = \operatorname{rank}(\mathbf{D}f_\theta(X)) = r_n^*(\theta)$. Hence, from (5.B.4), $r_n^*(\theta) \leq r_n^*(\theta')$.

From similar arguments as above, $(X, \widetilde{\theta}) \in \widetilde{\mathcal{O}}_{j'}^n$ for some $j' \in [\![1, m_n]\!]$. Since also $\operatorname{rank}\big(\mathbf{D}f_{\widetilde{\theta}}(X)\big) = \operatorname{rank}(\mathbf{D}f_{\theta'}(X)) = r_n^*(\theta')$ from Proposition 104, then from (5.B.4), $r_n^*(\theta') \leq r_n^*(\widetilde{\theta})$.

This shows $r_n^*(\theta) \leq r_n^*(\widetilde{\theta})$, and by swapping the roles of θ and $\widetilde{\theta}$, we obtain $r_n^*(\theta) = r_n^*(\widetilde{\theta})$.

Finally, since for $\widetilde{\theta} \sim \theta$, $r_n^*(\theta) = r_n^*(\widetilde{\theta})$ for all $n \in \mathbb{N}^*$, then (5.5.7) guarantees that $r^*(\theta) = r^*(\widetilde{\theta})$. This concludes the proof of Proposition 109.

5.B.5 Proof of Proposition 110

Similarly to what was done in the proof of Theorem 103, all along the proof of Proposition 110, we consider the canonical basis $(e_1, \ldots, e_{|E|+|B|})$ of $\mathbb{R}^E \times \mathbb{R}^B$ and the canonical basis $(\varepsilon_1, \ldots, \varepsilon_n)$ of $\mathbb{R}^n$ (recall $N_L = 1$). For all $\theta' \in \mathbb{R}^E \times \mathbb{R}^B$ and all $X' \in \mathbb{R}^n$ for which $\theta'' \longmapsto f_{\theta''}(X')$ is differentiable at θ', we write $Jf_{\theta'}(X')$ for the $n \times (|E| + |B|)$ matrix of $\mathbf{D}f_{\theta'}(X')$ in these two bases.

Consider $N_L = 1$, $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ and $\varepsilon > 0$ as in the proposition first statement. From Proposition 107, there is $n \in \mathbb{N}^*$, $j \in [\![1, m_n]\!]$ and $\bar{X} = (x^{(1)}, \ldots, x^{(n)})$ such that $(\bar{X}, \theta) \in \mathcal{O}_j^n$ and $\operatorname{rank}\big(\mathbf{D}f_\theta(\bar{X})\big) = r^*(\theta)$. Also, from Theorem 103 (i), there is $\epsilon' > 0$ such that $\theta' \mapsto f_{\theta'}(\bar{X})$ is continuously differentiable on $B(\theta, \epsilon')$. We consider such an ϵ' satisfying $\epsilon' \leq \epsilon$.

Then, $r^*(\theta) = \operatorname{rank}\big(\mathbf{D}f_\theta(\bar{X})\big) = \operatorname{rank}\big(Jf_\theta(\bar{X})\big)$. Hence we can extract $r^*(\theta)$ rows from $Jf_\theta(\bar{X})$ such that, up to reordering, we have, with $X = (x^{(1)}, \dots, x^{(r^*(\theta))})$, $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r^*(\theta)$. Furthermore, still, $\theta' \longmapsto f_{\theta'}(X)$ is continuously differentiable on $B(\theta, \epsilon')$.

Then, we can extract $r^*(\theta)$ columns from $Jf_\theta(X)$ for which the resulting $r^*(\theta) \times r^*(\theta)$ matrix is invertible. These $r^*(\theta)$ columns are associated to a subset of $E \cup B$, that we write S. For $\phi \in S$, we define $\theta(\phi) \in \mathbb{R}^E \times \mathbb{R}^B$ by $\theta(\phi)_w = \theta_w$ for $w \in (E \cup B) \setminus S$ and by $\theta(\phi)_w = \phi_w$ for $w \in S$. We also let $\theta_{|S} = (\theta_w)_{w \in S}$. Then we consider the function $g : \mathbb{R}^S \longrightarrow \mathbb{R}^{r^*(\theta)}$ defined by the row vector $g(\phi) = f_{\theta(\phi)}(X)$.

The differential of g at $\theta_{|S}$ is defined by the invertible $r^*(\theta) \times r^*(\theta)$ matrix extracted from $Jf_\theta(X)$, discussed above, in the canonical bases of $\mathbb{R}^S$ and $\mathbb{R}^{r^*(\theta)}$. In addition, g is continuously differentiable on $B(\theta_{|S}, \epsilon')$.

Hence we can apply the inverse function theorem. There is an open set $U \subseteq B(\theta_{|S}, \varepsilon')$ containing $\theta_{|S}$ and an open set V containing $g(\theta_{|S}) = f_{\theta(\theta_{|S})}(X) = f_\theta(X)$ such that g is bijective from U to V. We let g^{-1} be the inverse of g. Then, there is $\gamma > 0$ small enough such that $f_\theta(X) + \gamma\{-1, 1\}^{r^*(\theta)} \subseteq V$, where "+" denotes the Minkowski sum.

Let $t = (t_1, \dots, t_{r^*(\theta)}) = f_\theta(X)$. Then, for each $I \subseteq [\![1, r^*(\theta)]\!]$, define $t' \in \mathbb{R}^{r^*(\theta)}$ by $t'_i = t_i + \gamma$ if $i \in I$ and $t'_i = t_i - \gamma$ if $i \notin I$. Since $t' \in t + \gamma\{-1, 1\}^{r^*(\theta)} \subseteq V$, we can define $\theta' = \theta(g^{-1}(t'))$ with $\theta' \in \theta(U) \subseteq B(\theta, \varepsilon') \subseteq B(\theta, \varepsilon)$. This yields, $f_{\theta'}(X) = f_{\theta(g^{-1}(t'))}(X) = g(g^{-1}(t')) = t'$. Hence, for $i \in I$, $f_{\theta'}(x^{(i)}) = t'_i = t_i + \gamma$ and for $i \notin I$, $f_{\theta'}(x^{(i)}) = t'_i = t_i - \gamma$. By definition, this implies that $\mathrm{fS}_{B(\theta,\varepsilon),\gamma} \geq r^*(\theta)$. This shows the first part of the proposition.

The second part of the proposition is a consequence of the first part. Indeed, let us fix some $\varepsilon > 0$, let us take $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$ such that $r^*(\theta) = \max_{\widetilde{\theta} \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}} r^*(\widetilde{\theta})$, and let us take $\gamma' > 0$ such that (5.5.9) holds with γ there replaced by γ' here. Then we have

$$\mathrm{fS}_{\mathbb{R}^E \times \mathbb{R}^B, \gamma'} \geq \mathrm{fS}_{B(\theta,\varepsilon),\gamma'} \geq r^*(\theta).$$

Hence the proof is concluded.

5.C Proofs of Section 5.6

5.C.1 Proof of Proposition 111

Before proving Proposition 111, we state and prove two lemmas.

Lemma 119. *Let $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$.*

Let $n, n' \in \mathbb{N}^*$, let $X \in \mathcal{X}_\theta^n$ and let $X' \in \mathcal{X}_\theta^{n'}$. Let us write $X'' \in \mathcal{X}_\theta^{n+n'}$ the matrix obtained by concatenating the columns of X and X'. We then have the following inequalities

$$\begin{aligned}
\operatorname{rank}(\mathbf{D}f_\theta(X)) &\leq \operatorname{rank}\big(\mathbf{D}f_\theta(X'')\big), \\
\operatorname{rank}\big(\mathbf{D}f_\theta(X')\big) &\leq \operatorname{rank}\big(\mathbf{D}f_\theta(X'')\big), \\
\operatorname{rank}\big(\mathbf{D}f_\theta(X'')\big) &\leq \operatorname{rank}(\mathbf{D}f_\theta(X)) + \operatorname{rank}\big(\mathbf{D}f_\theta(X')\big),
\end{aligned}$$

where using Proposition 106, (iii), all the differentials are well defined.

Proof of Lemma 119. Let us consider the canonical basis $\mathcal{B}$ of $\mathbb{R}^E \times \mathbb{R}^B$. Also, for any $m \in \mathbb{R}^*$, let us consider the canonical basis $\mathcal{B}_m$ of $\mathbb{R}^{N_L \times m}$. With the bases $\mathcal{B}$ and $\mathcal{B}_n$, the linear operator $\mathbf{D}f_\theta(X) : \mathbb{R}^E \times \mathbb{R}^B \to \mathbb{R}^{N_L \times n}$ corresponds to a Jacobian matrix that we write $Jf_\theta(X)$. We define $Jf_\theta(X')$ similarly with the bases $\mathcal{B}$ and $\mathcal{B}_{n'}$ and $Jf_\theta(X'')$ similarly with the bases $\mathcal{B}$ and $\mathcal{B}_{n+n'}$. Writing ∇_θ the gradient of a scalar quantity depending on $\theta \in \mathbb{R}^E \times \mathbb{R}^B$, then the matrix $Jf_\theta(X)$ is composed of the nN_L rows

$$Jf_\theta(X)_{i(N_L-1)+j,:} = \left(\nabla_\theta \left[f_\theta(x^{(i)})_j\right]\right)^\top \qquad \text{, for all } i \in [\![1, n]\!] \text{ and } j \in [\![1, N_L]\!],$$

the matrix $Jf_\theta(X')$ is composed of the $n'N_L$ rows

$$Jf_\theta(X')_{i(N_L-1)+j,:} = \left(\nabla_\theta \left[f_\theta(x'^{(i)})_j\right]\right)^\top \qquad \text{, for all } i \in [\![1, n']\!] \text{ and } j \in [\![1, N_L]\!],$$

and the matrix $Jf_\theta(X'')$ is the concatenation of $Jf_\theta(X)$ and $Jf_\theta(X')$. Well-known properties of the matrix rank yield

$$\begin{aligned}
\operatorname{rank}(Jf_\theta(X)) &\leq \operatorname{rank}\big(Jf_\theta(X'')\big),\\
\operatorname{rank}\big(Jf_\theta(X')\big) &\leq \operatorname{rank}\big(Jf_\theta(X'')\big),\\
\operatorname{rank}\big(Jf_\theta(X'')\big) &\leq \operatorname{rank}(Jf_\theta(X)) + \operatorname{rank}\big(Jf_\theta(X')\big).
\end{aligned}$$

These three inequalities are equivalent to the three inequalities of the lemma. □

Lemma 120. *Let $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \backslash \mathcal{Z}$. Let $n \in \mathbb{N}^*$ and let $X \in \mathcal{X}_\theta^n$. If $\operatorname{rank}(\mathbf{D}f_\theta(X)) < r^*(\theta)$, then there exists $x \in \mathcal{X}_\theta$ such that, writing $X_x \in \mathcal{X}_\theta^{n+1}$ for the matrix obtained by adding x as an additional last column to X, we have*

$$\operatorname{rank}(\mathbf{D}f_\theta(X_x)) \geq \operatorname{rank}(\mathbf{D}f_\theta(X)) + 1.$$

Furthermore, the set of such x's has non-zero Lebesgue measure on $\mathbb{R}^{N_0}$.

Proof of Lemma 120. Let $\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus \mathcal{Z}$, $n \in \mathbb{N}^*$ and $X \in \mathcal{X}_\theta^n$ such that $\operatorname{rank}(\mathbf{D}f_\theta(X)) < r^*(\theta)$. Let $n' \in \mathbb{N}^*$ and $X' \in \mathcal{X}_\theta^{n'}$ such that $\operatorname{rank}(\mathbf{D}f_\theta(X')) = r^*(\theta)$ (the existence being guaranteed by definition of $r^*(\theta)$). Let $X'' \in \mathcal{X}_\theta^{n+n'}$ be obtained by concatenating the columns of X and X'. From Lemma 119, we have

$$\operatorname{rank}\big(\mathbf{D}f_\theta(X'')\big) \geq \operatorname{rank}\big(\mathbf{D}f_\theta(X')\big) = r^*(\theta) \geq \operatorname{rank}(\mathbf{D}f_\theta(X)) + 1.$$

Using the matrix notations defined in the proof of Lemma 119, there exists a row of $Jf_\theta(X')$, written $\left(\nabla_\theta \left[f_\theta(x'^{(i)})_j\right]\right)^\top$ for $i \in [\![1, n']\!]$ and $j \in [\![1, N_L]\!]$, with $x'^{(i)} \in \mathcal{X}_\theta$, that does not belong to the range of $Jf_\theta(X)$. Recall that for $x \in \mathcal{X}_\theta$, we write X_x for the matrix of $\mathcal{X}_\theta^{n+1}$ obtained by concatenating X and x. From the above, we have $\operatorname{rank}\big(\mathbf{D}f_\theta(X_{x'^{(i)}})\big) \geq \operatorname{rank}(\mathbf{D}f_\theta(X)) + 1$.

This proves the first statement of the lemma and we still need to prove that the set of such x's has non-zero Lebesgue measure on $\mathbb{R}^{N_0}$.

Consider $x^{'(i)} \in \mathcal{X}_\theta$ as constructed in the first part of the proof. By continuity of $x \mapsto \mathbf{D}f_\theta(X_x)$ at $x^{'(i)}$, and since the rank is a lower semi-continuous function and $\mathcal{X}_\theta$ is open, then there exists $\varepsilon > 0$ such that $B(x^{'(i)}, \varepsilon) \subseteq \mathcal{X}_\theta$ and for all $x \in B(x^{'(i)}, \varepsilon)$,

$$\operatorname{rank}(\mathbf{D}f_\theta(X_x)) \geq \operatorname{rank}\big(\mathbf{D}f_\theta(X_{x^{'(i)}})\big) \geq \operatorname{rank}(\mathbf{D}f_\theta(X)) + 1.$$

The ball $B(x^{'(i)}, \varepsilon)$ has a non-zero Lebesgue measure, which concludes the proof. □

We can now prove Proposition 111.

Proof of Proposition 111. Let us first show that the sequence $(r_n^*(\theta))_{n\in\mathbb{N}^*}$ is non-decreasing. Let $n \in \mathbb{N}^*$ and $X \in \mathcal{X}_\theta^n$ such that $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r_n^*(\theta)$. For any $x \in \mathcal{X}_\theta$, using the notation X_x of Lemma 120, Lemma 119 shows that $r_n^*(\theta) = \operatorname{rank}(\mathbf{D}f_\theta(X)) \leq \operatorname{rank}(\mathbf{D}f_\theta(X_x))$, and thus $r_n^*(\theta) \leq r_{n+1}^*(\theta)$. Hence, the sequence $(r_n^*(\theta))_{n\in\mathbb{N}^*}$ is non-decreasing.

Since the input space of $\mathbf{D}f_\theta(X)$ is $\mathbb{R}^E \times \mathbb{R}^B$ for all X, the sequence is also upper bounded by $r^*(\theta) \leq |E| + |B|$. Therefore, since it only takes integer values, there exists n such that $r_n^*(\theta) = r^*(\theta)$ and we can write $n^*(\theta) \in \mathbb{N}^*$ for the smallest of these n. Let $n \in [\![1, n^*(\theta) - 1]\!]$ and $X \in \mathcal{X}_\theta^n$ such that $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r_n^*(\theta)$. We have $r_n^*(\theta) < r^*(\theta)$ and using Lemma 120, there exists $x \in \mathcal{X}_\theta$ such that $\operatorname{rank}(\mathbf{D}f_\theta(X_x)) \geq \operatorname{rank}(\mathbf{D}f_\theta(X)) + 1$. Hence we have

$$r_{n+1}^*(\theta) \geq r_n^*(\theta) + 1. \tag{5.C.1}$$

The sequence is therefore increasing on $[\![1, n^*(\theta)]\!]$. Because of the definition $n^*(\theta)$, it is also stationary (constant) on $\mathbb{N}^* \setminus [\![1, n^*(\theta)]\!]$.

Let us now prove the upper and lower bounds on $n^*(\theta)$.

Now, consider $n \in [\![1, n^*(\theta) - 1]\!]$ and $X \in \mathcal{X}_\theta^{n+1}$ such that $\operatorname{rank}(\mathbf{D}f_\theta(X)) = r_{n+1}^*(\theta)$. Let us write $X_n \in \mathcal{X}_\theta^n$ for the matrix obtained by removing the last column $x^{(n+1)}$ from X. Lemma 119 shows that

$$\operatorname{rank}(\mathbf{D}f_\theta(X)) \leq \operatorname{rank}(\mathbf{D}f_\theta(X_n)) + \operatorname{rank}\Big(\mathbf{D}f_\theta(x^{(n+1)})\Big).$$

Hence

$$\begin{aligned} r_{n+1}^*(\theta) = \operatorname{rank}(\mathbf{D}f_\theta(X)) &\leq \operatorname{rank}(\mathbf{D}f_\theta(X_n)) + N_L \\ &\leq r_n^*(\theta) + N_L. \end{aligned} \tag{5.C.2}$$

Grouping (5.C.1) and (5.C.2), we have

$$r_n^*(\theta) + 1 \leq r_{n+1}^*(\theta) \leq r_n^*(\theta) + N_L. \tag{5.C.3}$$

Furthermore, for $x \in \mathcal{X}_\theta$, we have $1 \leq \operatorname{rank}(\mathbf{D}f_\theta(x)) \leq N_L$. Indeed, the upper bound is due to the size of output space of $\mathbf{D}f_\theta(x)$ and the lower bound holds for instance because for any $v \in V_L$, $\partial(f_\theta(x))_v / \partial b_v = 1$. Hence $1 \leq r_1^*(\theta) \leq N_L$. Using (5.C.3) we can show by induction that for all $n \in [\![1, n^*(\theta)]\!]$, we have $n \leq r_n^*(\theta) \leq nN_L$. Applying these latter inequalities to $n = n^*(\theta)$ yields

$$\frac{r^*(\theta)}{N_L} \leq n^*(\theta) \leq r^*(\theta)$$

and the proof is concluded. □

5.C.2 Proof of Theorem 113

Lemma 121. *Fix $p \in (0,1)$ and $k \in \mathbb{N}^*$. Consider random variables $X_1, \dots, X_k$ valued in $\{0,1\}$ such that $\mathbb{P}(X_1 = 1) \geq p$ and for any $i \in [\![2,k]\!]$, for any $x_1, \dots, x_{i-1} \in \{0,1\}$, $\mathbb{P}(X_i = 1 | X_1 = x_1, \dots, X_{i-1} = x_{i-1}) \geq p$. Consider $B_1, \dots, B_k$ independent Bernoulli random variables such that for $i \in [\![1,k]\!]$, $\mathbb{P}(B_i = 1) = p$ and $\mathbb{P}(B_i = 0) = 1 - p$.*

Then there exists a finite probability space $(\Omega_k, \mathbb{P}_k)$ and random variables $Y_1, \dots, Y_k$, $C_1, \dots, C_k$ from Ω_k to $\{0,1\}$ such that

(1) For $i \in [\![1,k]\!]$ and $\omega \in \Omega_k$, $Y_i(\omega) \geq C_i(\omega)$;

(2) $(X_1, \dots, X_k)$ and $(Y_1, \dots, Y_k)$ have the same distribution;

(3) $(B_1, \dots, B_k)$ and $(C_1, \dots, C_k)$ have the same distribution.

As a consequence, for each $t \geq 0$,

$$\mathbb{P}(X_1 + \cdots + X_k \leq t) \leq \mathbb{P}(B_1 + \cdots + B_k \leq t). \tag{5.C.4}$$

Proof of Lemma 121. We prove the first part of the lemma by induction. Let $\mathcal{L}_k$ correspond to the statements (1) to (3) for a given $k \in \mathbb{N}^*$.

Let us first show that $\mathcal{L}_1$ is true. Let $\Omega_1 = \{1,2,3\}$ and $\mathbb{P}_1(1) = p$, $\mathbb{P}_1(2) = \mathbb{P}(X_1 = 1) - p$ and $\mathbb{P}_1(3) = 1 - \mathbb{P}(X_1 = 1)$. Let then $Y_1(1) = Y_1(2) = 1$, $Y_1(3) = 0$, $C_1(1) = 1$, $C_1(2) = C_1(3) = 0$. With this choice of Ω_1 , $\mathbb{P}_1$, Y_1 and C_1, $\mathcal{L}_1$ indeed holds.

Assume now that $\mathcal{L}_k$ holds for some $k \in \mathbb{N}^*$ and let us show that $\mathcal{L}_{k+1}$ holds. We thus consider Ω_k, $\mathbb{P}_k$, $Y_1, \dots, Y_k$ and $C_1, \dots, C_k$ as in the statements (1) to (3). We define $\Omega_{k+1} = \Omega_k \times \{1,2,3\}$. For $\omega \in \Omega_k$, $i \in [\![1,k]\!]$ and $j \in \{1,2,3\}$, let us define $Y_i(\omega, j) = Y_i(\omega)$ and $C_i(\omega, j) = C_i(\omega)$. Note that we use the convenient abuse of notation of defining $Y_1, \dots, Y_k$ and $C_1, \dots, C_k$ as both functions on Ω_k and Ω_{k+1}. For $\omega \in \Omega_k$, let us define $Y_{k+1}(\omega, 1) = Y_{k+1}(\omega, 2) = 1$, $Y_{k+1}(\omega, 3) = 0$, $C_{k+1}(\omega, 1) = 1$ and $C_{k+1}(\omega, 2) = C_{k+1}(\omega, 3) = 0$. Then Item (1) of $\mathcal{L}_{k+1}$ is satisfied.

In order to define $\mathbb{P}_{k+1}$, we denote, for $\omega \in \Omega_k$,

$$\mathbb{P}_\omega = \mathbb{P}(X_{k+1} = 1 | X_1 = Y_1(\omega), \dots, X_k = Y_k(\omega)).$$

Then, for $\omega \in \Omega_k$, we define $\mathbb{P}_{k+1}(\omega, 1) = \mathbb{P}_k(\omega) p$, $\mathbb{P}_{k+1}(\omega, 2) = \mathbb{P}_k(\omega)(\mathbb{P}_\omega - p)$ and $\mathbb{P}_{k+1}(\omega, 3) = \mathbb{P}_k(\omega)(1 - \mathbb{P}_\omega)$. Using that $\mathbb{P}_k$ is a probability measure on Ω_k, it is simple to see that $\mathbb{P}_{k+1}$ is a probability measure on Ω_{k+1}.

Consider now $x_1, \dots, x_k \in \{0,1\}$. If $\mathbb{P}(Y_1 = x_1, \dots, Y_k = x_k) = 0$ then, from Item (2) of $\mathcal{L}_k$, $\mathbb{P}(X_1 = x_1, \dots, X_k = x_k) = 0$. In this case, for all $x_{k+1} \in \{0,1\}$, we have

$$0 = \mathbb{P}(Y_1 = x_1, \dots, Y_k = x_k, Y_{k+1} = x_{x+1}) = \mathbb{P}(X_1 = x_1, \dots, X_k = x_k, X_{k+1} = x_{x+1}).$$

Consider then the case where $\mathbb{P}(Y_1 = x_1, \dots, Y_k = x_k) > 0$. We have

$$\mathbb{P}(Y_1 = x_1, \dots, Y_k = x_k, Y_{k+1} = 1) = \sum_{j=1}^{2} \sum_{\substack{\omega \in \Omega_k \text{ s.t.} \\ Y_i(\omega) = x_i \\ \text{for } i \in [\![1,k]\!]}} \mathbb{P}_{k+1}(\omega, j) = \sum_{\substack{\omega \in \Omega_k \text{ s.t.} \\ Y_i(\omega) = x_i \\ \text{for } i \in [\![1,k]\!]}} \mathbb{P}_k(\omega) \mathbb{P}_\omega.$$

Using now Item (2) of $\mathcal{L}_k$, we have

$$\begin{aligned}
&\mathbb{P}\left(Y_1 = x_1, \dots, Y_k = x_k, Y_{k+1} = 1\right) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k\right) \frac{1}{\mathbb{P}\left(Y_1 = x_1, \dots, Y_k = x_k\right)} \sum_{\substack{\omega \in \Omega_k \text{ s.t.} \\ Y_i(\omega) = x_i \\ \text{for } i \in [\![1,k]\!]}} \mathbb{P}_k(\omega) \mathbb{P}_\omega \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k\right) \frac{1}{\mathbb{P}\left(Y_1 = x_1, \dots, Y_k = x_k\right)} \sum_{\substack{\omega \in \Omega_k \text{ s.t.} \\ Y_i(\omega) = x_i \\ \text{for } i \in [\![1,k]\!]}} \mathbb{P}_k(\omega) \\
&\quad \mathbb{P}(X_{k+1} = 1 | X_1 = x_1, \dots, X_k = x_k) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k\right) \mathbb{P}(X_{k+1} = 1 | X_1 = x_1, \dots, X_k = x_k) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k, X_{k+1} = 1\right).
\end{aligned}$$

We treat the case $Y_{k+1} = 0$ similarly, writing the details for the sake of completeness. We have

$$\begin{aligned}
&\mathbb{P}\left(Y_1 = x_1, \dots, Y_k = x_k, Y_{k+1} = 0\right) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k\right) \frac{1}{\mathbb{P}\left(Y_1 = x_1, \dots, Y_k = x_k\right)} \sum_{\substack{\omega \in \Omega_k \text{ s.t.} \\ Y_i(\omega) = x_i \\ \text{for } i \in [\![1,k]\!]}} \mathbb{P}_{k+1}(\omega, 3) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k\right) \frac{1}{\mathbb{P}\left(Y_1 = x_1, \dots, Y_k = x_k\right)} \sum_{\substack{\omega \in \Omega_k \text{ s.t.} \\ Y_i(\omega) = x_i \\ \text{for } i \in [\![1,k]\!]}} \mathbb{P}_k(\omega)(1 - \mathbb{P}_\omega) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k\right) \frac{1}{\mathbb{P}\left(Y_1 = x_1, \dots, Y_k = x_k\right)} \sum_{\substack{\omega \in \Omega_k \text{ s.t.} \\ Y_i(\omega) = x_i \\ \text{for } i \in [\![1,k]\!]}} \mathbb{P}_k(\omega) \\
&\quad \mathbb{P}(X_{k+1} = 0 | X_1 = x_1, \dots, X_k = x_k) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k\right) \mathbb{P}(X_{k+1} = 0 | X_1 = x_1, \dots, X_k = x_k) \\
&= \mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k, X_{k+1} = 0\right).
\end{aligned}$$

Hence Item (2) of $\mathcal{L}_{k+1}$ holds.

Let us now show Item (3) of $\mathcal{L}_{k+1}$. The method is similar as above, but we give the details for completeness. Consider $c_1, \dots, c_k, c_{k+1} \in \{0, 1\}$. Using the definition of C_k and C_{k+1}, we have

$$\mathbb{P}\left(C_1 = c_1, \dots, C_k = c_k, C_{k+1} = c_{k+1}\right) = \mathbb{P}_{k+1}\left(A_{c_1} \times \cdots \times A_{c_{k+1}}\right),$$

where for $i \in [\![1, k+1]\!]$, $A_{c_i} = \{1\}$ if $c_i = 1$ and $A_{c_i} = \{2, 3\}$ if $c_i = 0$. If $c_{k+1} = 1$,

then

$$\begin{aligned}\mathbb{P}_{k+1}\left(A_{c_1}\times\cdots\times A_{c_{k+1}}\right) &= \mathbb{P}_{k+1}\left(A_{c_1}\times\cdots\times A_{c_k}\times\{1\}\right)\\ &= \sum_{\substack{i_1\in A_{c_1}\\ \cdots\\ i_k\in A_{c_k}}} \mathbb{P}_{k+1}\left((i_1,\ldots,i_k,1)\right)\\ &= \sum_{\substack{i_1\in A_{c_1}\\ \cdots\\ i_k\in A_{c_k}}} \mathbb{P}_k\left((i_1,\ldots,i_k)\right)p\\ &= \mathbb{P}_k(C_1=c_1,\ldots,C_k=c_k)p\\ &= \mathbb{P}_k(B_1=c_1,\ldots,B_k=c_k)p\\ &= \mathbb{P}_k(B_1=c_1,\ldots,B_k=c_k,B_{k+1}=c_{k+1}),\end{aligned}$$

where we have used Item 3 of $\mathcal{L}_k$ for the second equality before last above and the definition of $B_1,\ldots,B_{k+1}$, in the lemma statement, for the last equality. Similarly, we have, if $c_{k+1}=0$,

$$\begin{aligned}\mathbb{P}_{k+1}\left(A_{c_1}\times\cdots\times A_{c_{k+1}}\right) =&\mathbb{P}_{k+1}\left(A_{c_1}\times\cdots\times A_{c_k}\times\{2,3\}\right)\\ =&\sum_{\substack{i_1\in A_{c_1}\\ \cdots\\ i_k\in A_{c_k}}} \left(\mathbb{P}_{k+1}\left((i_1,\ldots,i_k,2)\right)+\mathbb{P}_{k+1}\left((i_1,\ldots,i_k,3)\right)\right)\\ =&\sum_{\substack{i_1\in A_{c_1}\\ \cdots\\ i_k\in A_{c_k}}} \mathbb{P}_k\left((i_1,\ldots,i_k)\right)(1-p)\\ =&\mathbb{P}_k(C_1=c_1,\ldots,C_k=c_k)(1-p)\\ =&\mathbb{P}_k(B_1=c_1,\ldots,B_k=c_k)(1-p)\\ =&\mathbb{P}_k(B_1=c_1,\ldots,B_k=c_k,B_{k+1}=c_{k+1}).\end{aligned}$$

Hence in all cases,

$$\mathbb{P}\left(C_1=c_1,\ldots,C_k=c_k,C_{k+1}=c_{k+1}\right)=\mathbb{P}\left(B_1=c_1,\ldots,B_k=c_k,B_{k+1}=c_{k+1}\right)$$

and $\mathcal{L}_{k+1}$ is proved. This finishes the proof by induction that $\mathcal{L}_k$ holds for all $k\in\mathbb{N}^*$.

Let us now show (5.C.4). We have from Items (2) and (3) of $\mathcal{L}_k$,

$$\begin{aligned}&\mathbb{P}\left(X_1+\cdots+X_k\le t\right)-\mathbb{P}\left(B_1+\cdots+B_k\le t\right)\\ =&\mathbb{P}\left(Y_1+\cdots+Y_k\le t\right)-\mathbb{P}\left(C_1+\cdots+C_k\le t\right)\\ =&\mathbb{E}\left(1_{\{Y_1+\cdots+Y_k\le t\}}-1_{\{C_1+\cdots+C_k\le t\}}\right)\\ \le&0,\end{aligned}$$

because from Item (1) of $\mathcal{L}_k$, the random variable

$$1_{\{Y_1+\cdots+Y_k\le t\}}-1_{\{C_1+\cdots+C_k\le t\}}$$

takes the values -1 or 0. □

Proof of Theorem 113. Let us first show that, almost surely, for all $\ell \in [\![1, n]\!]$, there exists $j \in [\![1, m_\ell]\!]$ such that $((x^{(i)})_{1\le i\le \ell}, \theta) \in \mathcal{O}_j^\ell$. Indeed, for any $\ell \in [\![1, n]\!]$, since $\theta \notin \mathcal{Z}$, the set $\{X \in \mathbb{R}^{N_0\times \ell} \mid (X, \theta) \in (\cup_{j=1}^{m_\ell} \mathcal{O}_j^\ell)^c\}$ has Lebesgue measure zero. Since the vectors $x^{(i)}$ are independent, the matrix $(x^{(i)})_{1\le i\le \ell} \in \mathbb{R}^{N_0\times \ell}$ follows the product distribution $\mathcal{G}^\ell$, which is absolutely continuous with respect to Lebesgue measure of $\mathbb{R}^{N_0\times \ell}$. Therefore, we have

$$\mathbb{P}\left(((x^{(i)})_{1\le i\le \ell}, \theta) \in (\cup_{j=1}^{m_\ell} \mathcal{O}_j^\ell)^c\right) = 0,$$

and since this is true for all $\ell \in [\![1, n]\!]$, we thus have

$$\mathbb{P}\left(\exists \ell \in [\![1, n]\!] \text{ such that } ((x^{(i)})_{1\le i\le \ell}, \theta) \in (\cup_{j=1}^{m_\ell} \mathcal{O}_j^\ell)^c\right) = 0.$$

As a consequence, for the rest of the proof, up to intersecting with an event of probability one, we will assume that for all $\ell \in [\![1, n]\!]$, there exists $j \in [\![1, m_\ell]\!]$ such that $((x^{(i)})_{1\le i\le \ell}, \theta) \in \mathcal{O}_j^\ell$.

To ease the reading, let us denote $N = n^*(\theta)$ in this proof. By definition of N, we have $r_N^*(\theta) = r^*(\theta)$, and Proposition 107 shows that there exists $j \in I_N(\theta)$ such that

$$r_N^*(\theta) = r_j^N.$$

Consequently, there exist deterministic $\widetilde{x}^{(1)}, \dots, \widetilde{x}^{(N)} \in \mathbb{R}^{N_0}$ such that if $\widetilde{X} = (\widetilde{x}^{(i)})_{1\le i\le N}$, we have $(\widetilde{X}, \theta) \in \mathcal{O}_j^N$ and

$$\operatorname{rank}\Big(\mathbf{D} f_\theta(\widetilde{X})\Big) = r^*(\theta).$$

For $\delta \in A(\theta)$, let us define the deterministic integer

$$c_\delta^* = \operatorname{Card}\{i \in [\![1, N]\!] \mid \widetilde{x}^{(i)} \in \operatorname{Int} D_\delta(\theta)\}.$$

We have

$$\sum_{\delta\in A(\theta)} c_\delta^* = N.$$

Let us also define, for $\ell \in [\![1, n]\!]$ and $\delta \in A(\theta)$, the random integers

$$c_\delta(\ell) = \min\left(\operatorname{Card}\{i \in [\![1, \ell]\!] \mid x^{(i)} \in \operatorname{Int} D_\delta(\theta)\} \quad , \quad c_\delta^*\right),$$

and

$$c(\ell) = \sum_{\delta\in A(\theta)} c_\delta(\ell).$$

We have $c_\delta(\ell) \le c_\delta^*$, for all $\delta \in A(\theta)$, thus $c(\ell) \le N$. The sequence $c(\ell)$ is nondecreasing, and at each step, the increment $c(\ell + 1) - c(\ell)$ is either 0 or 1.

Let us first show that, almost surely, for $\ell \in [\![1, n]\!]$,

$$\{c(\ell) = N\} \implies \left\{\operatorname{rank}\Big(\mathbf{D} f_\theta\left((x^{(i)})_{1\le i\le \ell}\right)\Big) = r^*(\theta)\right\}. \tag{5.C.5}$$

Suppose indeed that, for some $\ell \in [\![1, n]\!]$, $c(\ell) = N$. Then for all $\delta \in A(\theta)$, we have $c_\delta(\ell) = c^*_\delta$. Up to a re-ordering, we can assume that for all $i \in [\![1, N]\!]$,

$$a(x^{(i)}, \theta) = a(\widetilde{x}^{(i)}, \theta). \tag{5.C.6}$$

As assumed earlier, there exists $j' \in [\![1, m_N]\!]$ such that $((x^{(i)})_{1\le i\le N}, \theta) \in \mathcal{O}^N_{j'}$. The equality (5.C.6) and Item 2 of Theorem 103 (i) show that $j' = j$. Item 3 of Theorem 103 (i) shows that the rank is constant over $\mathcal{O}^N_j$, and thus $\operatorname{rank}\Big(\mathbf{D}f_\theta((x^{(i)})_{1\le i\le N})\Big) = r^*(\theta)$. This shows (5.C.5) as desired.

Define now $\bar{c}(\ell)$ by $\bar{c}(\ell) = c(\ell)$ if $c(\ell) < N$ and by $\bar{c}(\ell) = N + (\ell - M)$ if $c(\ell) = N$, where M is the smallest index i such that $c(i) = N$. Then, for all $\ell \in [\![1, n]\!]$, $c(\ell) = N \iff \bar{c}(\ell) \ge N$. Hence, we have

$$\begin{aligned}\mathbb{P}\left(\operatorname{rank}\Big(\mathbf{D}f_\theta\left((x^{(i)})_{1\le i\le n}\right)\Big) = r^*(\theta)\right) \ge& \mathbb{P}\left(c(n) = N\right)\\ =& \mathbb{P}\left(\bar{c}(n) \ge N\right).\end{aligned}$$

Thus

$$\mathbb{P}\left(\operatorname{rank}\Big(\mathbf{D}f_\theta\left((x^{(i)})_{1\le i\le n}\right)\Big) < r^*(\theta)\right) \le \mathbb{P}\left(\bar{c}(n) < N\right). \tag{5.C.7}$$

Define $X_1 = \bar{c}(1)$ and $X_k = \bar{c}(k) - \bar{c}(k-1)$ for $k \in [\![2, n]\!]$. Notice that $X_1 + \cdots + X_k = \bar{c}(k)$ for all $k \ge 2$. Consider also the i.i.d. Bernoulli variables $B_1, \dots, B_n$ from the first item of the theorem statement. We will apply Lemma 121 to p, as defined in Theorem 113, $(X_1, \dots, X_n)$ and $(B_1, \dots, B_n)$. To do so we need to prove that $(X_1, \dots, X_n)$ satisfies the hypotheses of Lemma 121. First, $\mathbb{P}(X_1 = 1)$ is the probability that $x^{(1)}$ falls into $\operatorname{Int} D_\delta(\theta)$ for some $\delta \in A(\theta)$. This probability is thus equal to p, and therefore lower bounded by p. Now let us show that, for $k \in [\![1, n-1]\!]$, $x_1, \dots, x_k \in \{0, 1\}$,

$$\mathbb{P}(X_{k+1} = 1 | X_1 = x_1, \dots, X_k = x_k) \ge p.$$

Let us work conditionnally to $X_1 = x_1, \dots, X_k = x_k$. If $x_1 + \cdots + x_k \ge N$, then $\bar{c}(k) = x_1 + \cdots + x_k \ge N$ and thus, by definition of $\bar{c}(k+1)$ and X_{k+1}, $X_{k+1} = 1$. So $\mathbb{P}(X_{k+1} = 1 | X_1 = x_1, \dots, X_k = x_k) = 1 \ge p$.

Consider now the case $x_1 + \cdots + x_k < N$. Conditionally to $x^{(1)}, \dots, x^{(k)}$ for which $X_1 = x_1, \dots, X_k = x_k$ with $x_1 + \cdots + x_k < N$, there is at least one $\delta \in A(\theta)$ such that $c_\delta(k) < c^*_\delta$, and if $x^{(k+1)}$ falls into $\operatorname{Int} D_\delta(\theta)$, then $X_{k+1} = 1$ because $c(k+1) = c(k)+1$ and thus $\bar{c}(k+1) = \bar{c}(k)+1$. Hence $\mathbb{P}\left(X_{k+1} = 1 | x^{(1)}, \dots, x^{(k)}\right) \ge p$ and thus

$$\begin{aligned}&\mathbb{P}\left(X_{k+1} = 1 | X_1 = x_1, \dots, X_k = x_k\right)\\ =&\frac{1}{\mathbb{P}(X_1 = x_1, \dots, X_k = x_k)}\mathbb{P}\left(X_1 = x_1, \dots, X_k = x_k, X_{k+1} = 1\right)\\ =&\frac{1}{\mathbb{P}(X_1 = x_1, \dots, X_k = x_k)}\mathbb{E}\left[\mathbb{E}\left[1_{\{X_1 = x_1, \dots, X_k = x_k\}} 1_{\{X_{k+1} = 1\}} \middle| x^{(1)}, \dots, x^{(k)}\right]\right]\\ =&\frac{1}{\mathbb{P}(X_1 = x_1, \dots, X_k = x_k)}\mathbb{E}\left[1_{\{X_1 = x_1, \dots, X_k = x_k\}}\mathbb{P}\left(X_{k+1} = 1 | x^{(1)}, \dots, x^{(k)}\right)\right]\\ \ge& p.\end{aligned}$$

Hence we can apply Lemma 121. From this lemma and (5.C.7), for all $n \in \mathbb{N}$,

$$\begin{aligned}\mathbb{P}\Big(\operatorname{rank}\Big(\mathbf{D}f_\theta\left((x^{(i)})_{1\leq i\leq n}\right)\Big) < r^*(\theta)\Big) \leq& \mathbb{P}(\bar{c}(n) < N)\\ =& \mathbb{P}(X_1 + \cdots + X_n < N)\\ \leq& \mathbb{P}(B_1 + \cdots + B_n < N).\end{aligned}$$

Hence Item 1 of Theorem 113 holds. Let us now consider Item 2. The expectation of $B_1 + \cdots + B_n$ is np and the variance is $np(1-p) \leq np$. Hence using Chebyshev's inequality, for $np \geq 2N$

$$\begin{aligned}\mathbb{P}\left(\operatorname{rank}\Big(\mathbf{D}f_\theta\left((x^{(i)})_{1\leq i\leq n}\right)\Big) < r^*(\theta)\right) \leq& \mathbb{P}(B_1 + \cdots + B_n < N)\\ =& \mathbb{P}(B_1 + \cdots + B_n - np < N - np)\\ \leq& \frac{np}{(np-N)^2}\\ \leq& \frac{4}{np}.\end{aligned}$$

This concludes the proof. □

Chapter 6

Conclusion

This thesis focuses on the question of identifiability for fully-connected ReLU networks. While identifiability is a classical concern in statistics, its application to neural networks is less studied, these models being often assumed to be non identifiable because of their large number of parameters, and the question of identifying their parameter values being less considered because the meaning of these parameters is less understood.

However, the question of identifiability is at the intersection of several fundamental issues. As described in the introduction, it relates to the issues of robustness and privacy, which are fundamental in regard to the large-scale deployment of real-world systems based on neural network models. Furthermore, the study of identifiability questions the relationship between the parameter space and the functional space associated to a neural network. This relationship is far from trivial, yet understanding it carries significant theoretical and practical implications. Understanding the redundancy of neural network parameters is indeed crucial to fully understand the nature and complexity of the aforementioned functional space, as well as the nature of the trajectories followed by the optimizers in the parameter space; two questions that are deeply linked to the performances of neural networks.

In Chapter 3, we establish a condition (or in fact, a set of conditions) that is sufficient for global identifiability modulo permutation and positive rescaling. The nature of these conditions allows to better understand the piecewise-affine structure or ReLU networks, and in particular, it shows that the singularities of a ReLU network are very informative about its parameters. We also provide in this chapter a toy example of a network satisfying the conditions, showing that the set of parameters satisfying the condition (and thus being identifiable) is nonempty. These conditions are however theoretical: they assume that the network is known on an input set with nonempty interior, which is far from what we can expect to know in practice, i.e. the values of the function on a finite list of inputs. The following chapter corresponds to an attempt to go towards more practical settings.

Indeed in Chapter 4, we establish two conditions of local identifiability from any finite sample: a necessary condition on one side and a sufficient condition on the other side. The form of identifiability that is tested is weaker since it is only local, but it allows us on the other hand to obtain conditions than can be of practical use: for any (fully-connected ReLU) network, and any finite input sample, the conditions can be tested via the rank computation of two linear operators. This work thus provides a tool whose use in a wide range of situations could help understand better the big picture regarding the parameter redundancy in fully-connected ReLU networks:

which architectures, and which parameters tend to permit local identifiability? For a specific parameter, how to choose the inputs to have identifiability?

In Chapter 5, we develop this study further by focusing on one of the quantities presented in Chapter 4, viewed as a local complexity measure and called batch functional dimension. When the batch functional dimension is maximal, it corresponds to the (locally) identifiable case studied in Chapter 5, but when it is not, it is also informative, by giving us an indication on the local complexity of the neural network. More precisely, we show that this measure has a geometric nature: it represents the dimension of the local image set of the network, for a fixed list of inputs. We show that the parameter space is divided into pieces over which the batch functional dimension is fixed. We observe empirically that the batch functional dimension decreases throughout training, which indicates an implicit regularization phenomenon. Furthermore, our experiments indicate that the batch functional dimension reached at the end of training is positively correlated to the complexity of the learning problem: the more complex the task, the higher the batch functional dimension. Finally, our experiments also indicate that the choice of the inputs is crucial, and that often, for some choices, it is possible to reach the maximal value, leading to local identifiability, as described in Chapter 4. To consolidate these empirical observations, we also show a theoretical link between this complexity measure and the (local) fat-shattering dimension of the neural network.

Overall, this thesis shows the relevance and the fruitfulness of the question of identifiability for neural networks. It first shows that it is possible to establish - theoretical as well as practical- conditions of identifiability for fully-connected deep ReLU neural networks, and to put to evidence some networks that satisfy them, which can be of practical interest for robustness or privacy. But this thesis also shows how, by measuring the parameter redundancy, the question of identifiability is also linked to the question of implicit regularization, which is fundamental to better understand the performances of neural networks.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.

[2] El Mehdi Achour, François Malgouyres, and Sébastien Gerchinovitz. *The loss landscape of deep linear neural networks: a second-order analysis*. 2022. arXiv: 2107.13289 [math.ST].

[3] Rilwan A Adewoyin, Peter Dueben, Peter Watson, Yulan He, and Ritabrata Dutta. "TRU-NET: a deep learning approach to high resolution prediction of rainfall". In: *Machine Learning* 110.8 (2021), pp. 2035–2062.

[4] Francesca Albertini, Eduardo D Sontag, and Vincent Maillot. "Uniqueness of weights for neural networks". In: *Artificial Neural Networks for Speech and Vision* (1993), pp. 115–125.

[5] Noga Alon, Shai Ben-David, Nicolo Cesa-Bianchi, and David Haussler. "Scale-sensitive dimensions, uniform convergence, and learnability". In: *Journal of the ACM (JACM)* 44.4 (1997), pp. 615–631.

[6] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.

[7] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. "Understanding deep neural networks with rectified linear units". In: *arXiv preprint arXiv:1611.01491* (2016).

[8] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. "Provable Bounds for Learning Some Deep Representations". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, 2014, pp. 584–592.

[9] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. "Implicit regularization in deep matrix factorization". In: *Advances in Neural Information Processing Systems* 32 (2019).

[10] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 322–332.

[11] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. "Stronger Generalization Bounds for Deep Nets via a Compression Approach". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. 2018, pp. 254–263.

[12] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. "Stronger generalization bounds for deep nets via a compression approach". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 254–263.

[13] Shumeet Baluja and Ian Fischer. "Adversarial transformation networks: Learning to generate adversarial examples". In: *arXiv preprint arXiv:1703.09387* (2017).

[14] David Barrett and Benoit Dherin. "Implicit Gradient Regularization". In: *International Conference on Learning Representations*. 2020.

[15] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. "Spectrally-normalized margin bounds for neural networks". In: *Advances in neural information processing systems* 30 (2017).

[16] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. "Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks." In: *Journal of Machine Learning Research* 20.63 (2019), pp. 1–17.

[17] Peter L Bartlett and Philip M Long. "Prediction, learning, uniform convergence, and scale-sensitive dimensions". In: *Journal of Computer and System Sciences* 56.2 (1998), pp. 174–190.

[18] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30063–30070.

[19] Peter L Bartlett and Wolfgang Maass. "Vapnik-Chervonenkis dimension of neural nets". In: *The handbook of brain theory and neural networks* (2003), pp. 1188–1192.

[20] Peter L Bartlett, Vitaly Maiorov, and Ron Meir. "Almost Linear VC-Dimension Bounds for Piecewise Polynomial Networks". In: *Neural Computation* 10.8 (1998), pp. 2159–2173.

[21] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. "Deep learning: a statistical viewpoint". In: *Acta numerica* 30 (2021), pp. 87–201.

[22] Mikhail Belkin. "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation". In: *Acta Numerica* 30 (2021), pp. 203–248.

[23] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.

[24] Mikhail Belkin, Daniel Hsu, and Ji Xu. "Two models of double descent for weak features". In: *SIAM Journal on Mathematics of Data Science* 2.4 (2020), pp. 1167–1180.

[25] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402.

[26] Joachim Bona-Pellissier, François Bachoc, and François Malgouyres. "Parameter identifiability of a deep feedforward ReLU neural network". In: *Machine Learning* (2023), pp. 1–63.

[27] Joachim Bona-Pellissier, François Malgouyres, and François Bachoc. *https://github.com/JoachimBP/Functional-dimension*. Code of the experiments of this article. 2023.

[28] Joachim Bona-Pellissier, François Malgouyres, and François Bachoc. "Local Identifiability of Deep ReLU Neural Networks: the Theory". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27549–27562.

[29] Léon Bottou, Frank E Curtis, and Jorge Nocedal. "Optimization methods for large-scale machine learning". In: *SIAM review* 60.2 (2018), pp. 223–311.

[30] Etienne Boursier, Loucas Pillaud-Vivien, and Nicolas Flammarion. "Gradient flow dynamics of shallow relu networks for square loss and orthogonal inputs". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 20105–20118.

[31] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. 2020, pp. 1877–1901.

[32] Alon Brutzkus and Amir Globerson. "Globally optimal gradient descent for a ConvNet with Gaussian inputs". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 605–614.

[33] James R Bunch and John E Hopcroft. "Triangular factorization and inversion by fast matrix multiplication". In: *Mathematics of Computation* 28.125 (1974), pp. 231–236.

[34] Alexander Camuto, George Deligiannidis, Murat A Erdogdu, Mert Gurbuzbalaban, Umut Simsekli, and Lingjiong Zhu. "Fractal structure and generalization properties of stochastic optimization algorithms". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18774–18788.

[35] Emmanuel J Candes. "The restricted isometry property and its implications for compressed sensing". In: *Comptes rendus mathematique* 346.9-10 (2008), pp. 589–592.

[36] Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. "Phase retrieval via matrix completion". In: *SIAM review* 57.2 (2015), pp. 225–251.

[37] Emmanuel J Candes, Thomas Strohmer, and Vladislav Voroninski. "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming". In: *Communications on Pure and Applied Mathematics* 66.8 (2013), pp. 1241–1274.

[38] Emmanuel J Candès, Justin Romberg, and Terence Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transactions on information theory* 52.2 (2006), pp. 489–509.

[39] Nicholas Carlini, Matthew Jagielski, and Ilya Mironov. "Cryptanalytic extraction of neural network models". In: *Annual International Cryptology Conference*. Springer. 2020, pp. 189–218.

[40] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. "The secret sharer: Evaluating and testing unintended memorization in neural networks". In: *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 2019, pp. 267–284.

[41] Nicholas Carlini and David Wagner. "Towards evaluating the robustness of neural networks". In: *2017 ieee symposium on security and privacy (sp)*. IEEE. 2017, pp. 39–57.

[42] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. "Swad: Domain generalization by seeking flat minima". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 22405–22418.

[43] Hervé Chabanne, Vincent Despiegel, and Linda Guiga. "A Protection against the Extraction of Neural Network Models". In: *arXiv preprint arXiv:2005.12782* (2020).

[44] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. "Entropy-sgd: Biasing gradient descent into wide valleys". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12 (2019), p. 124018.

[45] An Mei Chen, Haw-minn Lu, and Robert Hecht-Nielsen. "On the geometry of feedforward neural network error surfaces". In: *Neural computation* 5.6 (1993), pp. 910–927.

[46] Jiyu Chen, Yiwen Guo, Qianjun Zheng, and Hao Chen. "Protect privacy of deep classification networks by exploiting their generative power". In: *Machine Learning* 110.4 (2021), pp. 651–674.

[47] Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. "Fast matrix rank algorithms and applications". In: *Journal of the ACM (JACM)* 60.5 (2013), pp. 1–25.

[48] Lenaic Chizat and Francis Bach. "Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss". In: *Conference on Learning Theory*. PMLR. 2020, pp. 1305–1338.

[49] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. "Houdini: Fooling deep structured prediction models". In: *arXiv preprint arXiv:1707.05373* (2017).

[50] Roberto Colomboni, Emmanuel Esposito, and Andrea Paudice. "An Improved Uniform Convergence Bound with Fat-Shattering Dimension". In: *arXiv preprint arXiv:2307.06644* (2023).

[51] Yaim Cooper. "Global minima of overparameterized neural networks". In: *SIAM Journal on Mathematics of Data Science* 3.2 (2021), pp. 676–691.

[52] Don Coppersmith and Shmuel Winograd. "Matrix multiplication via arithmetic progressions". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. 1987, pp. 1–6.

[53] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. "Sharp minima can generalize for deep nets". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1019–1028.

[54] David L Donoho. "Compressed sensing". In: *IEEE Transactions on information theory* 52.4 (2006), pp. 1289–1306.

[55] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. "Gradient descent finds global minima of deep neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 1675–1685.

[56] Simon S Du, Jason D Lee, and Yuandong Tian. "When is a Convolutional Filter Easy to Learn?" In: *International Conference on Learning Representations*. 2018.

[57] Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[58] Dennis Maximilian Elbrächter, Julius Berner, and Philipp Grohs. "How degenerate is the parametrization of neural networks with the ReLU activation function?" In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[59] Yossi Erlich, Dan Chazan, Scott Petrack, and Avraham Levy. "Lower Bound on VC-Dimension by Local Shattering". In: *Neural Computation* 9.4 (1997), pp. 771–776. DOI: 10.1162/neco.1997.9.4.771.

[60] Xing-Rong Fan, Meng-Zhen Kang, Ep Heuvelink, Philippe De Reffye, and Bao-Gang Hu. "A knowledge-and-data-driven modeling approach for simulating plant growth: A case study on tomato growth". In: *Ecological Modelling* 312 (2015), pp. 363–373.

[61] Charles Fefferman. "Reconstructing a neural net from its output". In: *Revista Matemática Iberoamericana* 10.3 (1994), pp. 507–555.

[62] Christian Fiedler, Massimo Fornasier, Timo Klock, and Michael Rauchensteiner. "Stable recovery of entangled weights: Towards robust identification of deep neural networks from minimal samples". In: *Applied and Computational Harmonic Analysis* 62 (2023), pp. 123–172.

[63] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. "Sharpness-aware Minimization for Efficiently Improving Generalization". In: *International Conference on Learning Representations*. 2021.

[64] Massimo Fornasier, Timo Klock, Marco Mondelli, and Michael Rauchensteiner. "Finite Sample Identification of Wide Shallow Neural Networks with Biases". In: *arXiv preprint arXiv:2211.04589* (2022).

[65] Massimo Fornasier, Timo Klock, and Michael Rauchensteiner. "Robust and resource-efficient identification of two hidden layer neural networks". In: *Constructive Approximation* (2019), pp. 1–62.

[66] Massimo Fornasier, Jan Vybíral, and Ingrid Daubechies. "Robust and resource efficient identification of shallow neural networks by fewest samples". In: *Information and Inference: A Journal of the IMA* 10.2 (2021), pp. 625–695.

[67] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. "Model inversion attacks that exploit confidence information and basic countermeasures". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 2015, pp. 1322–1333.

[68] Haoyu Fu, Yuejie Chi, and Yingbin Liang. "Guaranteed recovery of one-hidden-layer neural networks via cross entropy". In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 3225–3235.

[69] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. "Escaping from saddle points—online stochastic gradient for tensor decomposition". In: *Conference on learning theory*. PMLR. 2015, pp. 797–842.

[70] Rong Ge, Jason D Lee, and Tengyu Ma. "Learning one-hidden-layer neural networks with landscape design". In: *6th International Conference on Learning Representations, ICLR 2018*. 2018.

[71] Jonas Geiping, Micah Goldblum, Phil Pope, Michael Moeller, and Tom Goldstein. "Stochastic Training is Not Necessary for Generalization". In: *International Conference on Learning Representations*. 2021.

[72] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. "An investigation into neural net optimization via Hessian eigenvalue density". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2232–2241.

[73] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. "Implicit regularization of discrete gradient dynamics in linear neural networks". In: *Advances in Neural Information Processing Systems* 32 (2019).

[74] Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. "The Implicit Bias of Depth: How Incremental Learning Drives Generalization". In: *International Conference on Learning Representations*. 2019.

[75] Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. "On the Symmetries of Deep Learning Models and their Internal Representations". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. 2022.

[76] Surbhi Goel, Adam Klivans, and Raghu Meka. "Learning One Convolutional Layer with Overlapping Patches". In: *International Conference on Machine Learning*. 2018, pp. 1783–1791.

[77] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. "Size-independent sample complexity of neural networks". In: *Conference On Learning Theory*. PMLR. 2018, pp. 297–299.

[78] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[79] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*. 2015.

[80] Elisenda Grigsby, Kathryn Lindsey, and David Rolnick. "Hidden symmetries of ReLU networks". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 11734–11760.

[81] J Elisenda Grigsby and Kathryn Lindsey. "On transversality of bent hyperplane arrangements and the topological expressiveness of ReLU neural networks". In: *SIAM Journal on Applied Algebra and Geometry* 6.2 (2022), pp. 216–242.

[82] J Elisenda Grigsby, Kathryn Lindsey, Robert Meyerhoff, and Chenxi Wu. "Functional dimension of feedforward ReLU neural networks". In: *arXiv preprint arXiv:2209.04036* (2022).

[83] Philipp Grohs and Gitta Kutyniok, eds. *Mathematical Aspects of Deep Learning*. Cambridge University Press, 2022.

[84] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. "Gradient descent happens in a tiny subspace". In: *arXiv preprint arXiv:1812.04754* (2018).

[85] Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. "Reconstructing training data from trained neural networks". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 22911–22924.

[86] Boris Hanin and David Rolnick. "Complexity of linear regions in deep networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2596–2604.

[87] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. "Deep speech: Scaling up end-to-end speech recognition". In: *arXiv preprint arXiv:1412.5567* (2014).

[88] Robert Hecht-Nielsen. "On the algebraic structure of feedforward network weight spaces". In: *Advanced Neural Computers*. Elsevier, 1990, pp. 129–135.

[89] Geoffrey Hinton et al. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[90] Sepp Hochreiter and Jürgen Schmidhuber. "Flat minima". In: *Neural computation* 9.1 (1997), pp. 1–42.

[91] Oscar H Ibarra, Shlomo Moran, and Roger Hui. "A generalization of the fast LUP matrix decomposition algorithm and applications". In: *Journal of Algorithms* 3.1 (1982), pp. 45–56.

[92] Masaaki Imaizumi and Johannes Schmidt-Hieber. "On generalization bounds for deep networks based on loss surface implicit regularization". In: *IEEE Transactions on Information Theory* 69.2 (2023).

[93] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods". In: *arXiv preprint arXiv:1506.08473* (2015).

[94] Ziwei Ji and Matus Telgarsky. "Directional convergence and alignment in deep learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17176–17186.

[95] Paul C Kainen, Vera Kurková, and Andrew Vogt. "An integral formula for Heaviside neural networks". In: *Neural Network World* 10 (2000), pp. 313–319.

[96] Paul C Kainen, Věra Kůrková, Vladik Kreinovich, and Ongard Sirisaengtaksin. "Uniqueness of network parametrization and faster learning". In: *Neural, Parallel & Scientific Computations* 2.4 (1994), pp. 459–466.

[97] Nal Kalchbrenner and Phil Blunsom. "Recurrent continuous translation models". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, pp. 1700–1709.

[98] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. "Generalization in deep learning". In: *arXiv preprint arXiv:1710.05468* 1.8 (2017).

[99] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima". In: *International Conference on Learning Representations*. 2017.

[100] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836* (2016).

[101] Niklas Koep, Arash Behboodi, and Rudolf Mathar. "An introduction to compressed sensing". In: *Compressed Sensing and Its Applications: Third International MATHEON Conference 2017*. Springer. 2019, pp. 1–65.

[102] Tjalling C Koopmans and Olav Reiersol. "The identification of structural characteristics". In: *The Annals of Mathematical Statistics* 21.2 (1950), pp. 165–181.

[103] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[104] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world". In: *International Conference on Learning Representations* (2017).

[105] Věra Kůrková and Paul C Kainen. "Functionally equivalent feedforward neural networks". In: *Neural Computation* 6.3 (1994), pp. 543–558.

[106] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. "Measuring the Intrinsic Dimension of Objective Landscapes". In: *International Conference on Learning Representations*. 2018.

[107] Gen Li, Ganghua Wang, and Jie Ding. "Provable Identifiability of Two-Layer ReLU Neural Networks via LASSO Regularization". In: *IEEE Transactions on Information Theory* (2023).

[108] Yuanzhi Li, Tengyu Ma, and Hongyang R Zhang. "Learning Over-Parametrized Two-Layer Neural Networks beyond NTK". In: *Conference on Learning Theory*. PMLR. 2020, pp. 2613–2682.

[109] Yuanzhi Li and Yang Yuan. "Convergence Analysis of Two-layer Neural Networks with ReLU Activation". In: *Advances in neural information processing systems*. 2017, pp. 597–607.

[110] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. "Fisher-rao metric, geometry, and complexity of neural networks". In: *The 22nd international conference on artificial intelligence and statistics*. PMLR. 2019, pp. 888–896.

[111] Grigorios Loukides, Joshua C Denny, and Bradley Malin. "The disclosure of diagnosis codes can breach research participants' privacy". In: *Journal of the American Medical Informatics Association* 17.3 (2010), pp. 322–327.

[112] Kaifeng Lyu and Jian Li. "Gradient Descent Maximizes the Margin of Homogeneous Neural Networks". In: *International Conference on Learning Representations*. 2019.

[113] Wolfgang Maass. "Neural nets with superlinear VC-dimension". In: *Neural Computation* 6.5 (1994), pp. 877–884.

[114] François Malgouyres. "On the stable recovery of deep structured linear networks under sparsity constraints". In: *Mathematical and Scientific Machine Learning*. Princeton, United States, 2020.

[115] Francois Malgouyres. "On the stable recovery of deep structured linear networks under sparsity constraints". In: *Mathematical and Scientific Machine Learning*. PMLR. 2020, pp. 107–127.

[116] François Malgouyres and Joseph Landsberg. "Multilinear compressive sensing and an application to convolutional linear networks". In: *SIAM Journal on Mathematics of Data Science* 1.3 (2019), pp. 446–475.

[117] François Malgouyres and Joseph Landsberg. "Multilinear compressive sensing and an application to convolutional linear networks". In: *SIAM Journal on Mathematics of Data Science* 1.3 (2019), pp. 446–475.

[118] François Malgouyres and Joseph Landsberg. "On the identifiability and stable recovery of deep/multi-layer structured matrix factorization". In: *Information Theory Workshop*. Proceedings of the Information Theory Workshop. Cambridge, United Kingdom, 2016.

[119] François Malgouyres and Joseph Landsberg. "On the identifiability and stable recovery of deep/multi-layer structured matrix factorization". In: *IEEE, Info. Theory Workshop*. 2016.

[120] François Malgouyres and Joseph Landsberg. "Stable recovery of the factors from a deep matrix product". In: *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*. proceedings of SPARS. Lisbonne, Portugal, 2017.

[121] Flavio Martinelli, Berfin Simsek, Johanni Brea, and Wulfram Gerstner. "Expand-and-Cluster: Exact Parameter Recovery of Neural Networks". In: *arXiv preprint arXiv:2304.12794* (2023).

[122] Eldad Meller, Alexander Finkelstein, Uri Almog, and Mark Grobman. "Same, same but different: Recovering neural network quantization error through weight factorization". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4486–4495.

[123] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. 2010, pp. 1045–1048.

[124] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2013.

[125] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. "On the number of linear regions of deep neural networks". In: *Advances in neural information processing systems* 27 (2014).

[126] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. "Universal adversarial perturbations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1765–1773.

[127] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582.

[128] Vaishnavh Nagarajan and J Zico Kolter. "Generalization in deep networks: The role of distance from initialization". In: *arXiv preprint arXiv:1901.01672* (2019).

[129] Vaishnavh Nagarajan and J Zico Kolter. "Uniform convergence may be unable to explain generalization in deep learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[130] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. "Data-free quantization through weight equalization and bias correction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1325–1334.

[131] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. "Deep double descent: Where bigger models and more data hurt". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12 (2021), p. 124003.

[132] Arvind Narayanan and Vitaly Shmatikov. "Robust de-anonymization of large sparse datasets". In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 111–125.

[133] Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. "Equivariant architectures for learning in deep weight spaces". In: *arXiv preprint arXiv:2301.12780* (2023).

[134] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. "Exploring generalization in deep learning". In: *Advances in neural information processing systems* 30 (2017).

[135] Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. "Path-SGD: Path-normalized optimization in deep neural networks". In: *Advances in neural information processing systems* 28 (2015).

[136] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. "In search of the real inductive bias: On the role of implicit regularization in deep learning". In: *arXiv preprint arXiv:1412.6614* (2014).

[137] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. "Norm-based capacity control in neural networks". In: *Conference on learning theory*. PMLR. 2015, pp. 1376–1401.

[138] Quynh Nguyen and Matthias Hein. "The loss surface of deep and wide neural networks". In: *International conference on machine learning*. PMLR. 2017, pp. 2603–2612.

[139] Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. "I know what you trained last summer: A survey on stealing machine learning models and defences". In: *ACM Computing Surveys* (2023).

[140] Samet Oymak and Mahdi Soltanolkotabi. "Learning a deep convolutional neural network via tensor decomposition". In: *Information and Inference: A Journal of the IMA* (2021).

[141] Rina Panigrahy, Ali Rahimi, Sushant Sachdeva, and Qiuyi Zhang. "Convergence Results for Neural Networks via Electrodynamics". In: *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2018.

[142] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[143] Carlos Daniel Mimoso Paulino and Carlos Alberto de Bragança Pereira. "On identifiability of parametric statistical models". In: *Journal of the Italian Statistical Society* 3 (1994), pp. 125–151.

[144] Philipp Petersen, Mones Raslan, and Felix Voigtlaender. "Topological properties of the set of functions generated by neural networks of fixed size". In: *Foundations of computational mathematics* 21.2 (2021), pp. 375–444.

[145] Henning Petzka, Martin Trimmel, and Cristian Sminchisescu. "Notes on the Symmetries of 2-Layer ReLU-Networks". In: *Proceedings of the Northern Lights Deep Learning Workshop*. Vol. 1. 2020, pp. 6–6.

[146] Mary Phuong and Christoph H Lampert. "Functional vs. parametric equivalence of relu networks". In: *International Conference on Learning Representations*. 2019.

[147] Mary Phuong and Christoph H. Lampert. "Functional vs. parametric equivalence of ReLU networks". In: *International Conference on Learning Representations*. 2020.

[148] José Pedro Pinto, André Pimenta, and Paulo Novais. "Deep learning and multivariate time series for cheat detection in video games". In: *Machine Learning* 110.11 (2021), pp. 3037–3057.

[149] Arya A Pourzanjani, Richard M Jiang, and Linda R Petzold. "Improving the identifiability of neural networks for Bayesian inference". In: *NIPS Workshop on Bayesian Deep Learning*. Vol. 4. 2017, p. 31.

[150] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. "On the expressive power of deep neural networks". In: *international conference on machine learning*. PMLR. 2017, pp. 2847–2854.

[151] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. "On the spectral bias of neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5301–5310.

[152] Zhi-Yong Ran and Bao-Gang Hu. "Determining structural identifiability of parameter learning machines". In: *Neurocomputing* 127 (2014), pp. 88–97.

[153] Zhi-Yong Ran and Bao-Gang Hu. "Parameter Identifiability in Statistical Machine Learning: A Review". In: ().

[154] Zhi-Yong Ran and Bao-Gang Hu. "Parameter identifiability in statistical machine learning: a review". In: *Neural Computation* 29.5 (2017), pp. 1151–1203.

[155] Noam Razin and Nadav Cohen. "Implicit regularization in deep learning may not be explainable by norms". In: *Advances in neural information processing systems* 33 (2020), pp. 21174–21187.

[156] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[157] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[158] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. "Mlaas: Machine learning as a service". In: *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE. 2015, pp. 896–902.

[159] David Rolnick and Konrad Kording. "Reverse-engineering deep ReLU networks". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. 2020, pp. 8178–8187.

[160] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[161] Itay Safran, Gal Vardi, and Jason D Lee. "On the effective number of linear regions in shallow univariate ReLU networks: Convergence guarantees and implicit bias". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 32667–32679.

[162] Itay M Safran, Gilad Yehudai, and Ohad Shamir. "The effects of mild over-parameterization on the optimization landscape of shallow ReLU neural networks". In: *Conference on Learning Theory*. PMLR. 2021, pp. 3889–3934.

[163] Levent Sagun, Leon Bottou, and Yann LeCun. "Eigenvalues of the Hessian in deep learning: Singularity and beyond". In: *arXiv preprint arXiv:1611.07476* (2016).

[164] Haşim Sak, Andrew Senior, and Françoise Beaufays. "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling". In: *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

[165] Sayantan Sarkar, Ankan Bansal, Upal Mahbub, and Rama Chellappa. "UPSET and ANGRI: Breaking high performance image classifiers". In: *arXiv preprint arXiv:1707.01159* (2017).

[166] Andrew M Saxe, James L McClelland, and Surya Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks". In: *arXiv preprint arXiv:1312.6120* (2013).

[167] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. "A mathematical theory of semantic development in deep neural networks". In: *Proceedings of the National Academy of Sciences* 116.23 (2019), pp. 11537–11546.

[168] Hanie Sedghi and Anima Anandkumar. "Provable methods for training neural networks with sparse connectivity". In: *Deep Learning and representation learning workshop: NIPS*. 2014.

[169] Thiago Serra, Abhinav Kumar, and Srikumar Ramalingam. "Lossless compression of deep neural networks". In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 17th International Conference, CPAIOR 2020, Vienna, Austria, September 21–24, 2020, Proceedings*. Springer. 2020, pp. 417–430.

[170] Thiago Serra, Xin Yu, Abhinav Kumar, and Srikumar Ramalingam. "Scaling up exact neural network compression by ReLU stability". In: *Advances in neural information processing systems* 34 (2021), pp. 27081–27093.

[171] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. "Implicit neural representations with periodic activation functions". In: *Advances in neural information processing systems* 33 (2020), pp. 7462–7473.

[172] Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. "On the Origin of Implicit Regularization in Stochastic Gradient Descent". In: *International Conference on Learning Representations*. 2020.

[173] Samuel L Smith and Quoc V Le. "A bayesian perspective on generalization and stochastic gradient descent". In: *arXiv preprint arXiv:1710.06451* (2017).

[174] Mahdi Soltanolkotabi. "Learning relus via gradient descent". In: *Advances in neural information processing systems*. 2017, pp. 2007–2017.

[175] Sho Sonoda, Isao Ishikawa, and Masahiro Ikeda. "Ghosts in Neural Networks: Existence, Structure and Role of Infinite-Dimensional Null Space". In: *arXiv preprint arXiv:2106.04770* (2021).

[176] Daniel Soudry and Yair Carmon. "No bad local minima: Data independent training error guarantees for multilayer neural networks". In: *arXiv preprint arXiv:1605.08361* (2016).

[177] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. "The implicit bias of gradient descent on separable data". In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 2822–2878.

[178] Pierre Stock. "Efficiency and Redundancy in Deep Learning Models : Theoretical Considerations and Practical Applications". PhD thesis. Université de Lyon, 2021.

[179] Pierre Stock and Rémi Gribonval. "An embedding of ReLU networks and an analysis of their identifiability". In: *Constructive Approximation* (2022), pp. 1–47.

[180] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks". In: *IEEE Transactions on Evolutionary Computation* 23.5 (2019), pp. 828–841.

[181] Héctor J Sussmann. "Uniqueness of the weights for minimal feedforward nets with a given input-output map". In: *Neural networks* 5.4 (1992), pp. 589–593.

[182] Taiji Suzuki, Hiroshi Abe, and Tomoaki Nishimura. "Compression based bound for non-compressed network: unified generalization error analysis of large compressible deep neural network". In: *International Conference on Learning Representations*. 2020.

[183] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. 2014.

[184] GM Tallis and Peter Chesson. "Identifiability of mixtures". In: *Journal of the Australian Mathematical Society* 32.3 (1982), pp. 339–348.

[185] Nadav Timor, Gal Vardi, and Ohad Shamir. "Implicit regularization towards rank minimization in relu networks". In: *International Conference on Algorithmic Learning Theory*. PMLR. 2023, pp. 1429–1459.

[186] Guillermo Valle-Pérez and Ard A Louis. "Generalization bounds for deep learning". In: *arXiv preprint arXiv:2012.04115* (2020).

[187] VN Vapnik and A Ya Chervonenkis. "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". In: *Theory of Probability and its Applications* 16.2 (1971), p. 264.

[188] Verner Vlačić and Helmut Bölcskei. "Affine symmetries and neural network identifiability". In: *Advances in Mathematics* 376 (2021), p. 107485.

[189] Verner Vlačić and Helmut Bölcskei. "Neural network identifiability for a family of sigmoidal nonlinearities". In: *Constructive Approximation* 55.1 (2022), pp. 173–224.

[190] N Vapnik Vladimir and Vlamimir Vapnik. "Statistical learning theory". In: *Xu JH and Zhang XG. translation. Beijing: Publishing House of Electronics Industry, 2O04* (1998).

[191] Binghui Wang and Neil Zhenqiang Gong. "Stealing hyperparameters in machine learning". In: *2018 IEEE symposium on security and privacy (SP)*. IEEE. 2018, pp. 36–52.

[192] Lei Wu, Zhanxing Zhu, and Weinan E. "Towards understanding generalization of deep learning: Perspective of loss landscapes". In: *Workshop 'Principled Approaches to Deep Learning', ICML* (2017).

[193] Dmitry Yarotsky. "Optimal approximation of continuous functions by very deep ReLU networks". In: *Conference on learning theory*. PMLR. 2018, pp. 639–649.

[194] Mingyang Yi, Qi Meng, Wei Chen, Zhi-ming Ma, and Tie-Yan Liu. "Positively scale-invariant flatness of ReLU neural networks". In: *arXiv preprint arXiv:1903.02237* (2019).

[195] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[196] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. "Protecting intellectual property of deep neural networks with watermarking". In: *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 2018, pp. 159–172.

[197] Ningyi Zhang, Xiaohan Zhou, Mengzhen Kang, Bao-Gang Hu, Ep Heuvelink, and Leo FM Marcelis. "Machine learning versus crop growth models: an ally, not a rival". In: *AoB Plants* 15.2 (2023), plac061.

[198] Shuai Zhang, Meng Wang, Sijia Liu, Pin-Yu Chen, and Jinjun Xiong. "Guaranteed Convergence of Training Convolutional Neural Networks via Accelerated Gradient Descent". In: *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2020, pp. 1–6.

[199] Shuai Zhang, Meng Wang, Jinjun Xiong, Sijia Liu, and Pin-Yu Chen. "Improved Linear Convergence of Training CNNs With Generalizability Guarantees: A One-Hidden-Layer Case". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.6 (2020), pp. 2622–2635.

[200] Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. "Learning One-hidden-layer ReLU Networks via Gradient Descent". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1524–1534.

[201] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. "Recovery Guarantees for One-hidden-layer Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 4140–4149.

[202] Allan Zhou, Kaien Yang, Kaylee Burns, Yiding Jiang, Samuel Sokota, J Zico Kolter, and Chelsea Finn. "Permutation equivariant neural functionals". In: *arXiv preprint arXiv:2302.14040* (2023).

[203] Mo Zhou, Rong Ge, and Chi Jin. "A local convergence theory for mildly over-parameterized two-layer neural network". In: *arXiv preprint arXiv:2102.02410* (2021).

[204] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. "Non-vacuous Generalization Bounds at the ImageNet Scale: a PAC-Bayesian Compression Approach". In: *International Conference on Learning Representations*. 2018.

L'identifiabilité des réseaux de neurones profonds ReLU

Résumé: Cette thèse étudie la question de l'identifiabilité des réseaux de neurones profonds ReLU. Les réseaux de neurones admettent des paramètres sous forme de poids et de biais, et avec un choix de paramètres donné, un réseau implémente une fonction. La question générale de l'identifiabilité est la suivante : si les fonctions implémentées par deux réseaux sont égales, ou si elles coïncident sur un ensemble donné, ont-elles les mêmes paramètres ? Dans cette thèse, nous proposons trois contributions qui tournent autour de ce sujet.

Première contribution : La possibilité de récupérer les paramètres – poids et biais – d'un réseau de neurones grâce à la connaissance de sa fonction sur un sous-ensemble de l'espace d'entrée peut être, selon la situation, une malédiction ou une bénédiction. D'un côté, récupérer les paramètres facilite les attaques adversariales et pourrait également révéler des informations sensibles du jeu de données utilisé pour entraîner le réseau. D'un autre côté, si les paramètres d'un réseau peuvent être récupérés, cela garantit à l'utilisateur que les caractéristiques dans les espaces latents peuvent être interprétées. Cela fournit également les bases pour obtenir des garanties formelles sur les performances du réseau. Il est donc important de caractériser les réseaux dont les paramètres peuvent être identifiés et ceux dont les paramètres ne le peuvent pas. Dans ce travail, nous fournissons un ensemble de conditions sur un réseau de neurones profond feedforward fully-connected ReLU, garantissant que les paramètres du réseau sont identifiables de manière unique – modulo permutation et rescalings positifs– à partir de la fonction que le réseau implémente sur un sous-ensemble de l'espace d'entrée.

Deuxième contribution : Un échantillon est-il suffisamment riche pour déterminer, au moins localement, les paramètres d'un réseau de neurones ? Pour répondre à cette question, nous introduisons une nouvelle paramétrisation locale d'un réseau de neurones ReLU profond donné en fixant les valeurs de certains de ses poids. Cela nous permet de définir des opérateurs de lifting locaux dont les inverses sont des cartes d'une variété lisse d'un espace de grande dimension. La fonction implémentée par le réseau de neurones ReLU profond compose le lifting local avec un opérateur linéaire qui dépend de l'échantillon. Nous dérivons de cette représentation pratique une condition géométrique nécessaire et suffisante d'identifiabilité locale. En examinant les espaces tangents, la condition géométrique fournit : 1/ une condition nécessaire sharp et testable d'identifiabilité et 2/ une condition suffisante sharp et testable d'identifiabilité locale. La validité des conditions peut être testée numériquement à l'aide de la rétropropagation et du calcul du rang des matrices.

Troisième contribution : Nous examinons les propriétés et les aspects computationnels des mesures de complexité locale des réseaux de neurones ReLU profonds, récemment introduites dans (Grigsby et al. 2022). Les mesures de complexité considérées sont liées à la géométrie locale d'un ensemble d'images et d'un ensemble de pré-images de la fonction implémentée par le réseau, pour un échantillon donné. La géométrie locale de l'ensemble de pré-images et de l'ensemble d'images est liée par la différentielle des sorties du réseau par rapport aux paramètres, pour un échantillon fini X. En particulier, nous considérons le rang de cette différentielle. L'ensemble de pré-images représente les redondances dans les paramètres d'un réseau. Intuitivement, plus il y a de redondances dans les paramètres, moins l'espace des fonctions représentées par le réseau est riche et complexe. Parmi d'autres propriétés, nous cherchons notamment à comprendre comment ces objets se comportent pendant l'optimisation. Le travail mené dans cette contribution est directement liée à la question de l'identifiabilité, l'ensemble de pré-images représentant les redondances des paramètres du réseau.

On identifiability of deep ReLU neural networks

Abstract: This thesis is focused on the question of identifiability for deep ReLU neural networks. Neural networks admit parameters in the form of weights and biases, and given a parameter choice, a network implements a function. The general question of identifiability is: if the functions implemented by two networks are equal, or if they coincide on a given set, do they have the same parameters? In this thesis, we propose three contributions revolving around this subject.

First contribution: The possibility for one to recover the parameters –weights and biases– of a neural network thanks to the knowledge of its function on a subset of the input space can be, depending on the situation, a curse or a blessing. On one hand, recovering the parameters allows for better adversarial attacks and could also disclose sensitive information from the dataset used to construct the network. On the other hand, if the parameters of a network can be recovered, it guarantees the user that the features in the latent spaces can be interpreted. It also provides foundations to obtain formal guarantees on the performances of the network. It is therefore important to characterize the networks whose parameters can be identified and those whose parameters cannot. In this work, we provide a set of conditions on a deep fully-connected feedforward ReLU neural network under which the parameters of the network are uniquely identified –modulo permutation and positive rescaling– from the function it implements on a subset of the input space.

Second contribution: Is a sample rich enough to determine, at least locally, the parameters of a neural network? To answer this question, we introduce a new local parameterization of a given deep ReLU neural network by fixing the values of some of its weights. This allows us to define local lifting operators whose inverses are charts of a smooth manifold of a high dimensional space. The function implemented by the deep ReLU neural network composes the local lifting with a linear operator which depends on the sample. We derive from this convenient representation a geometric necessary and sufficient condition of local identifiability. Looking at tangent spaces, the geometric condition provides: 1/ a sharp and testable necessary condition of identifiability and 2/ a sharp and testable sufficient condition of local identifiability. The validity of the conditions can be tested numerically using backpropagation and matrix rank computations.

Third contribution: We investigate properties and computational aspects of local complexity measures of deep ReLU neural networks, recently introduced in (Grigsby et. al 2022). The considered complexity measures are linked to the local geometry of an image set and a pre-image set of the function implemented by the network, for a given sample. The local geometry of the pre-image set and of the image set are linked through the differential of the outputs of the network with respect to the parameters, for a given finite sample X. In particular, we consider the rank of this differential. The pre-image set represents the redundancies in the parameters of a network. Intuitively, the more there are redundancies in the parameters, the less the space of function represented by the network is rich and complex. Amongst other properties, we notably try to understand how these objects behave during optimization. The investigation done in this contribution is directly linked to the question of identifiability, as the pre-image set represents the redundancies of the parameters of the network.