Local Identifiability of Deep ReLU Neural Networks: the Theory

Joachim Bona-Pellissier[ab*], **François Malgouyres**[b], **François Bachoc**[b]
Institut de Mathématiques de Toulouse ; UMR 5219
Université de Toulouse ; CNRS
[a] UT1, F-31042 Toulouse, France
[b] UPS, F-31062 Toulouse Cedex 9, France

[*]Corresponding author: Joachim Bona-Pellissier; joachim.bona-pellissier@univ-toulouse.fr

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.

1 Introduction

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 [20], object recognition [33, 34], speech recognition [17, 36, 16], natural language processing [27, 26, 19], anomaly detection [32] or climate sciences [1].

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

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

from queries [9, 35]. This can be a concern since neural network providers may wish to keep these parameters secret, for security [21], for privacy [13, 8], or for intellectual property [42].

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. [4], 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.

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 [40, 2, 22, 18, 12]. 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 [30, 38] as well as deep [31, 5] 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. Stock and Gribonval [39] 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 [39] shares similarities with previous works on deep structured matrix factorization [23, 24, 25]. 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 [29] as well as positive [11, 23, 24, 25] results of inverse stability exist. The articles [23, 24, 25] examine the case of structured networks with the identity as activation function. Only [25] considers a finite X. The authors of [11] 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 [14], as well as ReLU in the fully-connected case

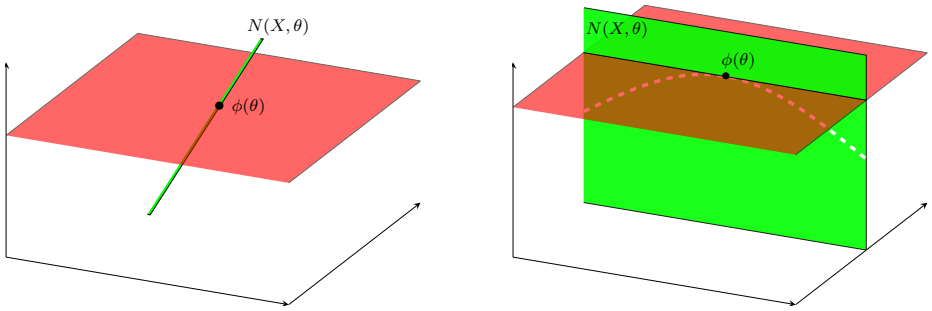

Figure 1: The local intersection between the affine space $N(X,\theta)$ (in green) and the smooth manifold Σ_1^* (color gradient). We also represent in red the tangent space to Σ_1^* at $\phi(\theta)$. Left: The identifiable case. The intersection is reduced to $\{\phi(\theta)\}$. Right: The non identifiable case. The intersection, represented with a dashed white line, is not reduced to $\{\phi(\theta)\}$.

[15, 44, 45, 46] or in the convolutional case [6, 43]. 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 [3], or for only recovering the first layer with sparsity assumptions [37] 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 [35] 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, [9] reconstructs a functionally equivalent network with fewer requests. As mentioned in Section 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.

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 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 testable conditions of local identifiability for any finite input sample. In particular, [39] 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).

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 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 3. This allows us to state our main results in Section 4, that is the geometric and the numerically testable conditions of local identifiability. Finally we discuss in Section 5 the numerical computations needed to test the latter conditions. All the proofs are provided in the appendices.

2 ReLU networks, lifting operator and rescaling of the parameters

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 $f_\theta = f_L$.

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 1. 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 [7]. The components of $\phi(\theta)$ will be associated to paths in the neural network. Linearity in Proposition 1 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 1 stated. We define

$$\mathcal{P} = \left(\bigcup_{l=0}^{L-1} \mathcal{P}_l \right) \cup \{\beta\}.$$

In a similar way to [39], we can now define the above-mentioned 'lifting operator'

$$\begin{array}{rccc} \phi: & \mathbb{R}^E \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^{\mathcal{P} \times V_L} \\ & \theta & \longmapsto & (\phi_{p,v}(\theta))_{p \in \mathcal{P}, v \in V_L} \end{array} \tag{1}$$

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'

$$\begin{array}{rccc} \alpha: & \mathbb{R}^{V_0} \times \left(\mathbb{R}^E \times \mathbb{R}^B \right) & \longrightarrow & \mathbb{R}^{1 \times \mathcal{P}} \\ & (x, \theta) & \longmapsto & (\alpha_p(x, \theta))_{p \in \mathcal{P}} \end{array} \tag{2}$$

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 1. *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 B, is for instance also stated in [39, 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 1 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 1 for all the x^i, we have the relation

$$f_\theta(X) = \alpha(X, \theta)\phi(\theta). \tag{3}$$

We prove in Appendix B the next proposition, which states that $\theta \mapsto \alpha(X, \theta)$ is piecewise constant.

Proposition 2. *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 2 alongside (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 Stock and Gribonval [39, Sec. 2], for any θ in such a neighborhood, we thus have

$$f_\theta(X) - f_{\tilde{\theta}}(X) = \alpha(X, \tilde{\theta}) \left(\phi(\theta) - \phi(\tilde{\theta}) \right). \tag{4}$$

Hence, studying ϕ will allow us to understand better how $f_\theta(X)$ locally depends on θ.

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

$$\mathrm{sign}(\theta) = \Big((\mathrm{sign}(w_{v \to v'})_{v \to v' \in E}, (\mathrm{sign}(b_v))_{v \in B} \Big) \in \{-1, 0, 1\}^E \times \{-1, 0, 1\}^B.$$

We can now describe the rescaling operations.

Definition 3. 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 $l \in [\![1, L]\!]$,

$$\begin{cases} W_l = \mathrm{Diag}(\lambda^l) \widetilde{W_l} \, \mathrm{Diag}(\lambda^{l-1})^{-1} \\ b_l = \mathrm{Diag}(\lambda^l) \tilde{b}_l. \end{cases} \tag{5}$$

- 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 (5) is equivalent to satisfy, for all $(v_{l-1}, v_l) \in V_{l-1} \times V_l$,

$$\begin{cases} w_{v_{l-1} \to v_l} = \frac{\lambda^l_{v_l}}{\lambda^{l-1}_{v_{l-1}}} \tilde{w}_{v_{l-1} \to v_l} \\ b_{v_l} = \lambda^l_{v_l} \tilde{b}_{v_l}. \end{cases} \tag{6}$$

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 [38, 39, 5, 28, 41]. We have indeed the following property: if $\theta \sim \tilde{\theta}$, for all $x \in \mathbb{R}^{V_0}$,

$$f_\theta(x) = f_{\tilde{\theta}}(x). \tag{7}$$

One of the interests of the operator ϕ is that it captures this invariant, as described by Stock and Gribonval [39, Sec. 2.4]. Propositions 4 and 5 are similar to their results and are restated here and proven in Appendix B for completeness. Indeed, combining the definition of ϕ with (6), we have the following property.

Proposition 4. *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 4 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 [39]:

$$S = \{\theta \in \mathbb{R}^E \times \mathbb{R}^B, \exists v \in V_1 \cup \cdots \cup V_{L-1},\ w_{v \to \bullet} = 0 \quad \text{or} \quad (w_{\bullet \to v}, b_v) = (0, 0)\}.$$

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 5 states that the reciprocal of Proposition 4 holds over $\left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$.

Proposition 5. *For all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B\right) \setminus S$, for all $\tilde{\theta} \in \mathbb{R}^E \times \mathbb{R}^B$,*

$$\phi(\theta) = \phi(\tilde{\theta}) \quad \Longrightarrow \quad \theta \overset{R}{\sim} \tilde{\theta}.$$

2.4 Local identifiability

We have now introduced all the concepts used in the formal definition of 'local identifiability'.

Definition 6. 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 $\|\theta - \tilde{\theta}\|_\infty < \epsilon$,

$$f_\theta(X) = f_{\tilde{\theta}}(X) \quad \Longrightarrow \quad \theta \sim \tilde{\theta}.$$

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 7, 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 8 illustrated in Figure 1. Instead of considering the intersection between a smooth manifold and an affine space as in Theorem 8, this indeed allows to consider the intersection between two affine spaces, which can be characterized with rank computations as in Theorems 9 and 10.

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 3, leaving $f_{\theta+u}(X)$ and $\phi(\theta + u)$ unchanged (7) and Proposition 4). 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 2 for an illustration. We will set to 0 the components of u corresponding to all the edges of the form $v \to s^\theta_{\max}(v)$. Intuitively,

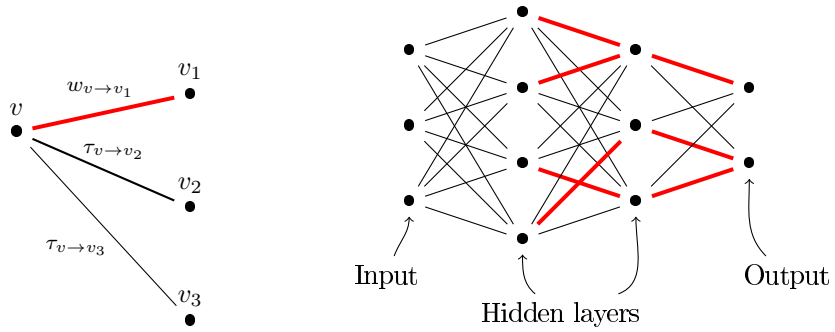

Figure 2: Left: The outward edges of a hidden neuron v and their weights. In this example, $v_1 = s^\theta_{\max}(v)$, so the weight of the edge in red, $v \to v_1$, has its value fixed as $w_{v\to v_1}$. The weights of the remaining edges, $\tau_{v\to v_2}$ and $\tau_{v\to v_3}$, are free to vary. Right: In red, all the edges whose weights are fixed. The remaining edges, in black, constitute the set F_θ.

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]

$$F_\theta = E \setminus \left(\bigcup_{l=1}^{L-1} \left\{ (v, s^\theta_{\max}(v)), v \in V_l \right\} \right). \tag{8}$$

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

$$\begin{array}{rccl} \rho_\theta : & \mathbb{R}^{F_\theta} \times \mathbb{R}^B & \longrightarrow & \mathbb{R}^E \times \mathbb{R}^B \\ & \tau & \longmapsto & \tilde\theta \quad \text{such that} \begin{cases} \forall (v,v') \in F_\theta, \quad \tilde w_{v\to v'} = \tau_{v\to v'} \\ \forall (v,v') \in E \backslash F_\theta, \quad \tilde w_{v\to v'} = w_{v\to v'} \\ \forall v \in B, \quad \tilde b_v = \tau_v. \end{cases} \end{array} \tag{9}$$

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

$$U_\theta = \rho_\theta^{-1} \left(\left(\mathbb{R}^E \times \mathbb{R}^B \right) \setminus S \right), \tag{10}$$

which is an open set of $\mathbb{R}^{F_\theta} \times \mathbb{R}^B$. We define, for all $\theta \in \left(\mathbb{R}^E \times \mathbb{R}^B \right) \setminus S$, the local lifting operator

$$\begin{array}{rccl} \psi^\theta : & U_\theta & \longrightarrow & \mathbb{R}^{\mathcal{P} \times V_L} \\ & \tau & \longmapsto & \phi \circ \rho_\theta(\tau). \end{array} \tag{11}$$

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 C), 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 7. *Σ_1^* is a smooth manifold of $\mathbb{R}^{\mathcal{P}\times V_L}$ of dimension*

$$|F_\theta| + |B| = N_0 N_1 + N_1 N_2 + \cdots + N_{L-1} N_L + N_L,$$

and the family $(V_\theta, (\psi^\theta)^{-1})_{\theta \in (\mathbb{R}^E \times \mathbb{R}^B) \setminus S}$ is an atlas.

Theorem 7 is proven in Appendix C. Besides being key in Section 4, Theorem 7 (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.

[1] Note, in the definition of F_θ, the index l starting at $l = 1$ and not $l = 0$.

4 Main results: necessary and sufficient conditions for local identifiability

The main results of this paper rely on the decomposition (4) introduced in Section 2. To reformulate (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 2. Using $A(X,\theta)$, the relation (4) satisfied by $\tilde{\theta}$ in the neighborhood of θ becomes

$$f_\theta(X) - f_{\tilde{\theta}}(X) = A(X,\theta)\cdot\left(\phi(\theta)-\phi(\tilde{\theta})\right). \tag{12}$$

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 (12) 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 $N(X,\theta)$.

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 8. *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 8 is proven in Appendix D, and is illustrated in Figure 1. This geometric condition is crucial for showing the next two results which give testable conditions of identifiability. Theorems 9 and 10 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

$$\Gamma(X,\theta) = A(X,\theta)\circ D\psi^\theta(\tau_\theta). \tag{13}$$

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 $0\le R_\Gamma\le|F_\theta|+|B|$, and the expression (13) shows that we also have $0\le R_\Gamma\le R_A$. 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 9 (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 10 (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 D. 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_Γ 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 7.

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 11. *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 $\tau \mapsto f_{\rho_\theta(\tau)}(X)$, 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*

$$D_\tau f_{\rho_\theta(\tau_\theta)}(X) = \Gamma(X, \theta). \tag{14}$$

The proof of Proposition 11 is in Appendix E. Since the reparameterization with ρ_θ simply consists in fixing the weights of the edges $v \to s^\theta_{\max}(v)$ to the value $w_{v \to s^\theta_{\max}(v)}$, (59) 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 E.

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 $\omega < 2.38$ [10].

When it comes to C_N, one needs in addition to know the rank R_A of $A(X, \theta)$, which, as Proposition 12 states, requires to compute the rank of $\alpha(X, \theta)$.

Proposition 12. *Let $X \in \mathbb{R}^{n \times V_0}$ and $\theta \in \mathbb{R}^E \times \mathbb{R}^B$. We have $R_A = N_L \operatorname{rank}(\alpha(X, \theta))$.*

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.

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 1). In the end, from Theorem 10 and Proposition 11, 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.

Acknowledgments and Disclosure of Funding

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/

References

[1] Rilwan A Adewoyin, Peter Dueben, Peter Watson, Yulan He, and Ritabrata Dutta. Tru-net: a deep learning approach to high resolution prediction of rainfall. *Machine Learning*, 110(8): 2035–2062, 2021.

[2] Francesca Albertini, Eduardo D Sontag, and Vincent Maillot. Uniqueness of weights for neural networks. *Artificial Neural Networks for Speech and Vision*, pages 115–125, 1993.

[3] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 584–592, Bejing, China, 22–24 Jun 2014. PMLR.

[4] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

[5] Joachim Bona-Pellissier, François Bachoc, and François Malgouyres. Parameter identifiability of a deep feedforward ReLU neural network. *arXiv preprint arXiv:2112.12982*, 2021.

[6] 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*, pages 605–614, 2017.

[7] Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251, 2015.

[8] 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)*, pages 267–284, 2019.

[9] Nicholas Carlini, Matthew Jagielski, and Ilya Mironov. Cryptanalytic extraction of neural network models. In *Annual International Cryptology Conference*, pages 189–218. Springer, 2020.

[10] Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast matrix rank algorithms and applications. *Journal of the ACM (JACM)*, 60(5):1–25, 2013.

[11] 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*, volume 32, 2019.

[12] Charles Fefferman. Reconstructing a neural net from its output. *Revista Matemática Iberoamericana*, 10(3):507–555, 1994.

[13] 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*, pages 1322–1333, 2015.

[14] Haoyu Fu, Yuejie Chi, and Yingbin Liang. Guaranteed recovery of one-hidden-layer neural networks via cross entropy. *IEEE Transactions on Signal Processing*, 68:3225–3235, 2020.

[15] 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.

[16] 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. *arXiv preprint arXiv:1412.5567*, 2014.

[17] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[18] Paul C Kainen, Věra Kůrková, Vladik Kreinovich, and Ongard Sirisaengtaksin. Uniqueness of network parametrization and faster learning. *Neural, Parallel & Scientific Computations*, 2(4): 459–466, 1994.

[19] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709, 2013.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations*, 2017.

[22] Věra Kůrková and Paul C Kainen. Functionally equivalent feedforward neural networks. *Neural Computation*, 6(3):543–558, 1994.

[23] François Malgouyres and Joseph Landsberg. On the identifiability and stable recovery of deep/multi-layer structured matrix factorization. In *IEEE, Info. Theory Workshop*, Sept. 2016.

[24] François Malgouyres and Joseph Landsberg. Multilinear compressive sensing and an application to convolutional linear networks. *SIAM Journal on Mathematics of Data Science*, 1(3):446–475, 2019.

[25] Francois Malgouyres. On the stable recovery of deep structured linear networks under sparsity constraints. In *Mathematical and Scientific Machine Learning*, pages 107–127. PMLR, 2020.

[26] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048, 2010.

[27] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[28] Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. Path-SGD: Path-normalized optimization in deep neural networks. *Advances in neural information processing systems*, 28, 2015.

[29] Philipp Petersen, Mones Raslan, and Felix Voigtlaender. Topological properties of the set of functions generated by neural networks of fixed size. *Foundations of Computational Mathematics*, 21:375–444, 2021.

[30] 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*, volume 1, pages 6–6, 2020.

[31] Mary Phuong and Christoph H. Lampert. Functional vs. parametric equivalence of ReLU networks. In *International Conference on Learning Representations*, 2020.

[32] José Pedro Pinto, André Pimenta, and Paulo Novais. Deep learning and multivariate time series for cheat detection in video games. *Machine Learning*, 110(11):3037–3057, 2021.

[33] 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*, pages 779–788, 2016.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[35] David Rolnick and Konrad Kording. Reverse-engineering deep ReLU networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8178–8187, 13–18 Jul 2020.

[36] 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.

[37] Hanie Sedghi and Anima Anandkumar. Provable methods for training neural networks with sparse connectivity. In *Deep Learning and representation learning workshop: NIPS*, 2014.

[38] Pierre Stock. *Efficiency and Redundancy in Deep Learning Models : Theoretical Considerations and Practical Applications*. PhD thesis, Université de Lyon, April 2021. URL https://tel.archives-ouvertes.fr/tel-03208517.

[39] Pierre Stock and Rémi Gribonval. An Embedding of ReLU Networks and an Analysis of their Identifiability. *Constructive Approximation*, 2022. URL https://hal.archives-ouvertes.fr/hal-03292203.

[40] Héctor J Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural networks*, 5(4):589–593, 1992.

[41] Mingyang Yi, Qi Meng, Wei Chen, Zhi-ming Ma, and Tie-Yan Liu. Positively scale-invariant flatness of ReLU neural networks. *arXiv preprint arXiv:1903.02237*, 2019.

[42] 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*, pages 159–172, 2018.

[43] 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. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2622–2635, 2020.

[44] 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*, pages 1524–1534. PMLR, 2019.

[45] 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*, pages 4140–4149, 2017.

[46] Mo Zhou, Rong Ge, and Chi Jin. A local convergence theory for mildly over-parameterized two-layer neural network. *arXiv preprint arXiv:2102.02410*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The outline in Section 1.4 ("Overview of the article") provides pointers to where the claimed contributions of the paper are provided.
   (b) Did you describe the limitations of your work? [Yes] Section 5 acknowledges the open problem of an efficient computation of the rank of $\alpha(X, \theta)$ and Section 6 describes other remaining open questions.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This is a theoretical/foundation work that adds to the theory and methodology of deep learning. As for any such contributions, the positive or negative societal impact will depend on the application case. We do not promote any harmful use of this theory, but we expand on the existing knowledge.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] See previous question.

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All our results explicitly refer to their required assumptions. Some general assumptions that hold throughout the paper are also stated at the beginning.
   (b) Did you include complete proofs of all theoretical results? [Yes] All the proofs are provided in the supplement.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] We did not run experiments.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] We did not use existing assets (code, data or models) nor cure/release new assets (code, data or models).
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing nor conducted research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]