Journal of Machine Learning Research 22 (2021) 1-34 Submitted 7/19; Revised 1/21; Published 7/21

Non-attracting Regions of Local Minima in Deep and Wide Neural Networks

Henning Petzka HENNING.PETZKA@MATH.LTH.SE
Lund University

Cristian Sminchisescu CRISTIAN.SMINCHISESCU@MATH.LTH.SE
Google Research
Lund University

Editor: Suvrit Sra

Abstract

Understanding the loss surface of neural networks is essential for the design of models with predictable performance and their success in applications. Experimental results suggest that sufficiently deep and wide neural networks are not negatively impacted by suboptimal local minima. Despite recent progress, the reason for this outcome is not fully understood. Could deep networks have very few, if at all, suboptimal local optima? or could all of them be equally good? We provide a construction to show that suboptimal local minima (i.e., non-global ones), even though degenerate, exist for fully connected neural networks with sigmoid activation functions. The local minima obtained by our construction belong to a connected set of local solutions that can be escaped from via a non-increasing path on the loss curve. For extremely wide neural networks of decreasing width after the wide layer, we prove that every suboptimal local minimum belongs to such a connected set. This provides a partial explanation for the successful application of deep neural networks. In addition, we also characterize under what conditions the same construction leads to saddle points instead of local minima for deep neural networks.

Keywords: Deep learning, neural network, local minima, global minima, path

1. Introduction

At the heart of most optimization problems lies the search for the global minimum of a loss function. The common approach to finding a solution is to initialize at random in parameter space and subsequently follow directions of decreasing loss based on local methods. This approach lacks a global progress criteria, which leads to descent into one of the nearest local minima. The common approach of using gradient descent variants on non-convex loss curves of deep neural networks is vulnerable precisely to that problem.

Authors pursuing the early approaches to local descent by back-propagating gradients (Rumelhart et al., 1986) experimentally noticed that suboptimal local minima appeared surprisingly harmless. More recently, for deep neural networks, the earlier observations were further supported by experiments of, e.g., Zhang et al. (2017). Several authors aimed to provide theoretical insight for this behavior. Some, aiming at explanations, rely on simplifying modeling assumptions. Others investigate neural networks under realistic assumptions,

but often focus on failure cases only. Recently, Nguyen and Hein (2017) provide partial explanations for deep and *extremely wide* neural networks for a class of activation functions including the commonly used sigmoid. Extreme width is characterized by a "wide" layer that has more neurons than input patterns to learn. For almost every instantiation of parameter values $\mathbf{w}$ (i.e., for all but a set of parameter values of measure zero) it is shown that, if the loss function has a local minimum at $\mathbf{w}$, then this local minimum must be a global one. This extends results by Gori and Tesi (1992) who required the input layer to be extremely wide. This suggests that for deep and wide neural networks with sigmoid activation functions, possibly every local minimum is global. The question on what happens at the null set of parameter values, for which the result does not hold, remained unanswered.

Similar observations for shallow neural networks with one hidden layer were made earlier by Poston et al. (1991). Poston et al. (1991) show for a neural network with one hidden layer and sigmoid activation function that, if the hidden layer has more nodes than there are training patterns, then the error function (squared sum of prediction losses over the samples) has no suboptimal "local minimum" and "each point is arbitrarily close to a point from which a strictly decreasing path starts, so such a point cannot be separated from a so called "good" point by a barrier of any positive height" (Poston et al., 1991). It was criticized by Sprinkhuizen-Kuyper and Boers (1999) that the definition of a local minimum used in the proof of Poston et al. (1991) was rather strict and unconventional. In particular, the results do not imply that no suboptimal local minima, defined in the usual way, exist. As a consequence, the notions of attracting and non-attracting regions of local minima were introduced and the authors proved that non-attracting regions exist by providing an example for the extended XOR problem. The existence of these regions imply that a gradient-based approach descending the loss surface using local information may still not converge to the global minimum. The main objective of this work is to revisit the problem of such non-attracting regions and show that they also exist in deep and extremely wide networks. In particular, a gradient based approach may get stuck in a suboptimal local minimum also in these networks. Most importantly, the performance of deep and wide neural networks cannot be explained by the analysis of the loss curve alone, without taking proper initialization or the stochasticity of stochastic gradient descent (SGD) into account.

Our observations are not fundamentally negative. At first, the local minima we find are rather degenerate. With proper initialization, a local descent technique is unlikely to get stuck in one of the degenerate, suboptimal local minima.[1] Secondly, the minima reside on a non-attracting region of local minima (see Definition 1). Due to its exploration properties, stochastic gradient descent will eventually be able to escape from such a region (see Wei et al., 2008). It is conceivable that in sufficiently wide and deep networks, except for a null set of parameter values as starting points, there is always a monotonically decreasing path down to the global minimum. This was shown for neural networks with one hidden layer, sigmoid activation function and square loss (Poston et al., 1991), and we generalize this result to deep neural networks. This implies that in such networks every local minimum belongs to a non-attracting region of local minima. (More precisely, our result holds for all extremely wide neural networks with square loss and a class of activation functions including the sigmoid, where the sequence of dimensions of hidden layers is non-increasing

1. That a proper initialization largely improves training performance is well-known. See, e.g., Wessels and Barnard (1992).

between the extremely wide layer and the output layer, i.e., the network architecture has no bottleneck layer of strictly lower dimension than both its neighboring layers.)

Our proof of the existence of suboptimal local minima even in extremely wide and deep networks is based on a construction of local minima in shallow neural networks given by Fukumizu and Amari (2000). By relying on a careful computation we are able to characterize when this construction is applicable to deep neural networks. Interestingly, in deeper layers, the construction rarely seems to lead to local minima, but more often to saddle points. The argument that saddle points rather than suboptimal local minima are the main problem in deep networks has been raised before (Dauphin et al., 2014) but a theoretical justification (Choromanska et al., 2015) uses strong assumptions that do not exactly hold in neural networks. Here, we provide the first analytical argument, under realistic assumptions on the neural network structure, describing when certain critical points (i.e., points with gradient zero) of the training loss lead to saddle points in deeper networks.

In summary, our results contain the following insight: There exist non-attracting regions of local minima and, in particular, suboptimal local minima in the loss surface of arbitrarily wide neural networks for a class of analytic activation functions including the sigmoid function. The minima can be both of finite type or only exist in the limit as some parameters converge to infinity. This disproves a conjecture made by Nguyen and Hein (2017) stating that for the therein studied extremely wide neural networks all local minima are globally optimal. Non-attracting regions of local minima, however, allow for non-increasing paths to the global minimum by first following degenerate directions of the local minimum. In sufficiently wide neural networks with no bottleneck layer, all local minima belong to non-attracting regions of local minima.

The extremely wide neural networks considered have zero loss at global minima. Naturally, training for zero global loss is not desirable in practice, neither is the use of fully connected extremely wide deep neural networks necessarily. The results of this paper are of theoretical importance. To be able to understand the complex learning behavior of deep neural networks in practice, it is a necessity to understand the networks with the most fundamental structure. In this regard, our results offer new understanding of the multidimensional loss surface of deep neural networks and their learning behavior.

2. Related Work

We discuss related work on suboptimal minima of the loss surface. In addition, we refer the reader to the overview article by Vidal et al. (2017) for a discussion on the non-convexity in neural network training.

It is known that learning the parameters of neural networks is, in general, a hard problem. Blum and Rivest (1992) prove NP-completeness for a specific neural network. It has also been shown that local minima and other critical points exist in the loss function of neural network training (Auer et al., 1995; Fukumizu and Amari, 2000; Nitta, 2017; Sminchisescu and Triggs, 2005; Sprinkhuizen-Kuyper and Boers, 1999; Wessels et al., 1990; Yun et al., 2018). The understanding of these critical points has led to significant improvements in neural network training. This includes weight initialization techniques (Wessels and Barnard, 1992), improved backpropagation algorithms to avoid saturation effects in neurons (Wang et al., 2004), entirely new activation functions, or the use of second order

information (Mizutani and Dreyfus, 2010; Amari, 1998). That suboptimal local minima must become rather degenerate if a neural network with sigmoid activation functions becomes sufficiently large was observed for networks with one hidden layer by Poston et al. (1991). Extending work by Gori and Tesi (1992), Nguyen and Hein (2017, 2018) generalized this result to deeper networks containing an extremely wide hidden layer. Our contribution can be considered as a continuation of this work.

To explain the persuasive performance of deep neural networks, Dauphin et al. (2014) experimentally show that there is a similarity in the behavior of critical points of the neural network's loss function with theoretical properties of critical points found for Gaussian fields on high-dimensional spaces (Bray and Dean, 1989). Choromanska et al. (2015) supply a theoretical connection, but they require strong (arguably unrealistic) assumptions on the network structure. The results imply that (under their assumptions on the deep network) the loss at a local minimum must be close to the loss of the global minimum with high probability. In this line of research, Sagun et al. (2015) experimentally show a similarity between spin glass models and the loss curve of neural networks.

To gain better insight into theoretical aspects, some papers considered linear networks, where the activation function is the identity. The classic result by Baldi and Hornik (1989) shows that linear two-layer neural networks have a unique global minimum and all other critical values are saddle points. Kawaguchi (2016), Lu and Kawaguchi (2017) and Yun et al. (2018) discuss generalizations of the results by Baldi and Hornik (1989) to deep linear networks, and Laurent and Brecht (2018a) finally show that for linear networks with no bottleneck layer, all minima are global.

There is a growing number of papers making considerable progress on the existence of suboptimal local minima for ReLU and LeakyReLU networks, where the space becomes combinatorial in terms of a positive activation, compared to a stalled (or weak) signal. Yun et al. (2019) prove existence of bad local minima in two-layer ReLU networks for generic data sets by tuning the weight parameters in such a way that all neurons are active and the network becomes locally linear. He et al. (2020) extend this to deep networks with piecewise linear activation functions, arbitrary continuously differentiable loss and multi-dimensional output. The existence of bad local minima had previously been shown under stronger assumptions by Du et al. (2018); Zhou and Liang (2018) and Swirszcz et al. (2016), who construct data sets that allow them to find suboptimal local minima in overparameterized networks. For the hinge loss, Laurent and Brecht (2018b) study one-hidden-layer networks and show that Leaky-ReLU networks don't have bad local minima, while ReLU networks do. Conditions for ReLU networks characterizing when no bad local minima exist or how to eliminate them is discussed by Liang et al. (2018a,b). Soudry and Hoffer (2017) probabilistically compare the volume of regions (for a specific measure) containing bad local and global minima in the limit, as the number of data points goes to infinity. For networks with one hidden layer and ReLU activation function, Freeman and Bruna (2017) quantify the amount of hill-climbing necessary to go from one point in the parameter space to another and find that for increasing overparameterization, all level sets become connected. Instead of analyzing local minima, Xie et al. (2016) consider regions where the derivative of the loss is small for two-layer ReLU networks. Soudry and Carmon (2016) consider LeakyReLU activation functions to find, similarly to the result of Nguyen

and Hein (2017), that for almost every combination of activation patterns in two consecutive mildly wide layers, a local minimum has global optimality.

The existence of non-increasing paths on the loss curve down to the global minimum is studied by Poston et al. (1991) for extremely wide two-layer neural networks with sigmoid activation functions. Venturi et al. (2019) similarly show that non-increasing paths down to the global minimum exist in wide two-layer networks, but their results also apply to the population risk. Nguyen et al. (2019) show existence of non-increasing paths for a special type of architecture having as many skip connections to the output as there are input patterns to learn. Very recently, Nguyen (2019) shows that there is always a continuous path of non-increasing loss down to an optimum for the same deep and extremely wide networks we consider, but for a class of piecewise linear activation functions that includes LeakyReLU activations. Our Theorem 5 shows that their result on the nonexistence of "bad local valleys" carries over to networks with sigmoid and tanh activation functions. For ReLU networks, Safran and Shamir (2016) show that, if one starts at a sufficiently high initialization loss, then there is a strictly decreasing path of parameters into the global minimum. Haeffele and Vidal (2017) consider a specific class of ReLU networks with regularization, give a sufficient condition that a local minimum is globally optimal, and show that a non-increasing path down to the global minimum exists.

Finally, worth mentioning is the study of Liao and Poggio (2017) who use polynomial approximations to argue, by relying on Bezout's theorem, that the loss function should have many local minima with zero empirical loss. Why deep networks perform better than shallow ones is also investigated by Poggio et al. (2017) by considering a class of compositional functions. Also relevant is the observation by Brady et al. (2006) showing that, if the global minimum is not of zero loss, then a perfect predictor may have a larger loss in training than one producing worse classification results.

3. Main Results

We consider **neural network functions** with fully connected layers of size $n_l,\ 0 \leq l \leq L$ given by

$$f(x) = \mathbf{w}^L(\sigma(\mathbf{w}^{L-1}(\sigma(\dots\sigma(\mathbf{w}^2(\sigma(\mathbf{w}^1 x + \mathbf{w}_0^1)) + \mathbf{w}_0^2)\dots)) + \mathbf{w}_0^{L-1})) + \mathbf{w}_0^L,$$

where $\mathbf{w^l} \in \mathbb{R}^{\mathbf{n_l} \times \mathbf{n_{l-1}}}$ denotes the **weight matrix** of the l-th layer, $1 \leq l \leq L$, $\mathbf{w}_0^l$ the **bias** terms, and σ a nonlinear **activation function**. The neural network function is denoted by f and we notationally suppress its dependence on parameters. We assume the activation function σ to belong to the class of strict monotonically increasing, analytic, bounded functions on $\mathbb{R}$ with image an interval (c, d) such that $0 \in [c, d]$, a **class denoted by $\mathcal{A}$**. As prominent examples, the sigmoid activation function $\sigma(t) = \frac{1}{1+\exp(-t)}$ and $\sigma(t) = \tanh(x)$ lie in $\mathcal{A}$. We assume no activation function at the output layer. All the networks considered in this paper are **regression networks** mapping into the real numbers $\mathbb{R}$, i.e., $\mathbf{n_L} = \mathbf{1}$ and $\mathbf{w}^L \in \mathbb{R}^{1 \times n_{L-1}}$. We train on a **finite data set** $(x_\alpha, y_\alpha)_{1 \leq \alpha \leq N}$ of size N with **input patterns** $x_\alpha \in \mathbb{R}^{n_0}$ and desired **target value** $y_\alpha \in \mathbb{R}$. We suppose throughout that the input patterns are pairwise different. We aim to minimize the **squared loss** $\mathcal{L} = \sum_{\alpha=1}^{N}(f(x_\alpha) - y_\alpha)^2$. Further, M denotes the **total number of parameters** and $\mathbf{w} \in \mathbb{R}^M$ denotes the **collection of all $\mathbf{w}^l$ and $\mathbf{w}_0^l$**.

The dependence of the neural network function f on $\mathbf{w}$ translates into a dependence $\mathcal{L} = \mathcal{L}(\mathbf{w})$ of the loss function on the parameters $\mathbf{w}$. Due to assumptions on σ, $\mathcal{L}(\mathbf{w})$ is twice continuously differentiable. The goal of training a neural network consists of minimizing $\mathcal{L}(\mathbf{w})$ over $\mathbf{w}$. There is a unique value $\mathcal{L}_0$ denoting the infimum of the neural network's loss (most often $\mathcal{L}_0 = 0$ in our examples). Any set of weight parameters $\mathbf{w}_\bullet$ that satisfies $\mathcal{L}(\mathbf{w}_\bullet) = \mathcal{L}_0$ is called a **global minimum**. Due to its non-convexity, the loss function $\mathcal{L}(\mathbf{w})$ of a neural network is in general known to potentially suffer from local minima (precise definition of a local minimum below). We will study the existence of **suboptimal local minima** in the sense that a local minimum $\mathbf{w}_*$ is suboptimal if its loss $\mathcal{L}(\mathbf{w}_*)$ is strictly larger than $\mathcal{L}_0$.

We refer to **deep neural networks** as networks with more than one hidden layer. Further, we refer to **extremely wide neural networks** as the type of networks considered in other theoretical work (Gori and Tesi, 1992; Poston et al., 1991; Nguyen and Hein, 2017, 2018; Nguyen, 2019) with one hidden layer containing at least as many neurons as input patterns (i.e., $n_l \geq N$ for some $1 \leq l < L$ in our notation).

3.1 A Special Kind of Local Minimum

The standard definition of a **local minimum**, which is also used here, is a point $\mathbf{w}_*$ such that $\mathbf{w}_*$ has a neighborhood U with $\mathcal{L}(\mathbf{w}) \geq \mathcal{L}(\mathbf{w}_*)$ for all $\mathbf{w} \in U$. Since local minima do not need to be **isolated** (i.e., $\mathcal{L}(\mathbf{w}) > \mathcal{L}(\mathbf{w}_*)$ for all $\mathbf{w} \in U \backslash \{w_*\}$) two types of connected regions of local minima may be distinguished. In the following definition, a **continuous path** is a continuous map $\mathbf{w}_\Gamma : [0,1] \to \mathbb{R}^M$ assigning each $t \in [0,1]$ a choice of parameters values $\mathbf{w}_\Gamma(t)$ with loss $\mathcal{L}(\mathbf{w}_\Gamma(t))$. We call the path **non-increasing** in $\mathcal{L}$ if $\mathcal{L}(\mathbf{w}_\Gamma(t)) \leq \mathcal{L}(\mathbf{w}_\Gamma(s))$ for all $t \geq s$. A non-increasing path $\mathbf{w}_\Gamma(t)$ **decreases the loss maximally**, if it cannot be extended as a non-increasing path to a parameter setting of lower loss, or formally, if there exists no non-increasing path $\tilde{\mathbf{w}}_\Gamma(t)$ such that for each t in $[0,1]$ there is s in $[0,1]$ with $\mathbf{w}_\Gamma(t) = \tilde{\mathbf{w}}_\Gamma(s)$ and such that $\mathcal{L}(\tilde{\mathbf{w}}_\Gamma(1)) < \mathcal{L}(\mathbf{w}_\Gamma(1))$.

Definition 1 *(Sprinkhuizen-Kuyper and Boers, 1999) Let $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. Suppose R is a maximal connected subset of parameter values $\mathbf{w} \in \mathbb{R}^m$, such that every $\mathbf{w} \in R$ is a local minimum of $\mathcal{L}$ with value $\mathcal{L}(\mathbf{w}) = c$.*

- *R is called an **attracting region of local minima**, if there is a neighborhood U of R such that every continuous path $\mathbf{w}_\Gamma(t)$, which is non-increasing in $\mathcal{L}$, which starts at some $\mathbf{w}_\Gamma(0) = \mathbf{w} \in U$ and which decreases the loss maximally, ends in R.*
- *R is called a **non-attracting region of local minima**, if every neighborhood U of R contains a point from where a continuous path $\mathbf{w}_\Gamma(t)$ exists that is non-increasing in $\mathcal{L}$ and ends in a point $\mathbf{w}_\Gamma(1)$ with $\mathcal{L}(\mathbf{w}_\Gamma(1)) < c$.*

Attracting regions of local minima R are called attracting, as decreasing paths starting in a neighborhood of R eventually end up in R. Our notion differs from the one of Sprinkhuizen-Kuyper and Boers (1999) by considering non-increasing paths instead of strictly decreasing ones (see also Hamey, 1998). Despite its non-attractive nature, a non-attracting region R of local minima may be harmful for a gradient descent approach. A

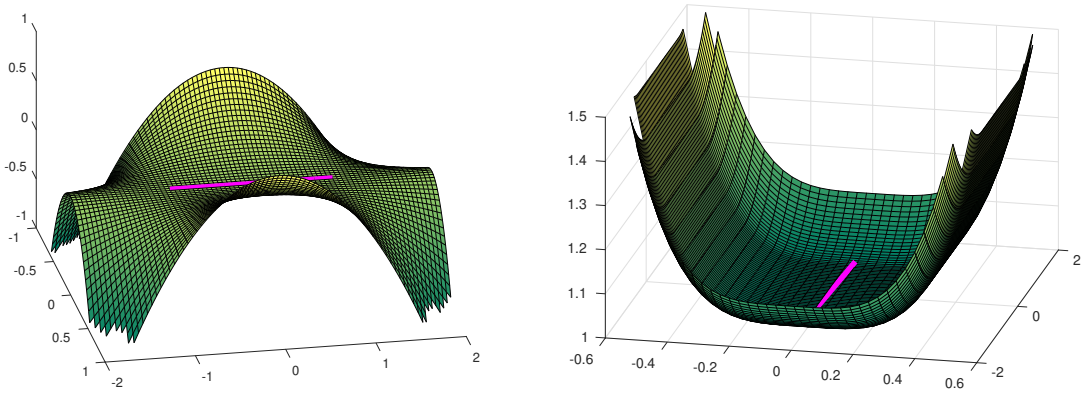


Figure 1: Left: A non-attracting region of local minima given by $R = \{(x, y) \mid x = 0, y \in (-1, 1)\}$ illustrated by the function $f(x, y) = x^2(1 - y^2)$. Right: An attracting region of local minima at the same region R for comparison. (These examples do not exactly appear in neural networks considered in this paper, but are of similar nature.)

path of greatest descent can end in a local minimum on R. However, no point z on R needs to have a neighborhood of attraction in the sense that following the path of greatest descent from a point in a neighborhood of z will lead back to z. (The path can lead to a different local minimum on R close by or reach points with strictly smaller values than c.) A rough illustration of a non-attracting region of local minima is depicted in Figure 1.[2] Such non-attracting regions of local minima are considered for neural networks with one hidden layer by Fukumizu and Amari (2000) and Wei et al. (2008) under the name of *singularities*. Their regions of local minima are characterized by singularities in the parameter space leading to a loss value strictly larger than the global loss. The dynamics around such a region are investigated by Wei et al. (2008).

Non-attracting regions of local minima do not only exist for shallow two-layer neural networks, but also for deep and arbitrary wide networks. A construction of such regions is shown in Section 5.3, proving the following result.

Theorem 2 *There exist deep and extremely wide fully-connected neural networks with sigmoid activation function such that the squared loss function of a finite data set has a non-attracting region of local minima (at finite parameter values).*

Corollary 3 *Any attempt to show for fully connected deep neural networks with sigmoid activations that a gradient descent technique will always lead to a global minimum only based on a description of the loss curve will fail if it doesn't take into consideration properties of*

2. While one might be tempted to term regions of local minima "generalized saddle points", we note that, under the usual mathematical definition, they do consists of a set of local minima.

the learning procedure (such as the stochasticity of stochastic gradient descent), properties of a suitable initialization technique, or assumptions on the data set.

On the positive side, we point out that a stochastic method such as stochastic gradient descent has a good chance to escape a non-attracting region of local minima due to noise. With infinite time at hand and sufficient exploration, the region can be escaped from with high probability (see Wei et al., 2008, for a more detailed discussion). In Section 5.1 we will further characterize when the method used to construct examples of regions of non-attracting local minima is applicable. This characterization limits us to the construction of extremely degenerate examples. We argue why assuring the necessary assumptions for the construction becomes difficult for wider and deeper networks and why it is natural to expect a lower suboptimal loss (where the suboptimal minima are less "bad") the less degenerate the constructed minima are and the more parameters a neural network possesses.

A different type of non-attracting regions of local minima is considered for the 2—3—1 XOR network by Sprinkhuizen-Kuyper and Boers (1999), where the region of local minima (of higher loss than the global loss) resides at points in parameter space with some coordinates being infinite. For this, we consider the extended parameter space, where parameters can take on values $\pm\infty$. The standard topology on this space considers open neighborhoods of ∞ defined by sets $\{v \mid v > a\}$ for some a. A **local minimum at infinity** then satisfies, by definition, that for sufficiently large values of a parameter, the loss is higher at finite values than at the limit as the parameter tends to infinity. In particular, a gradient descent approach may lead to diverging parameters in that case. However, a different non-increasing path down to the global minimum can still exist. It can be shown that such generalized local minima at infinity also exist for deep neural networks. (Our proof uses similar ideas as the proof for the 2—3—1- XOR network by Sprinkhuizen-Kuyper and Boers (1999, Section III), but needs additional arguments due to a more general setting. The proof can be found in Appendix A.2.)

Theorem 4 *Let $\mathcal{L}$ denote the squared loss of a fully connected regression neural network with sigmoid activation functions, having at least one hidden layer and each hidden layer containing at least two neurons. Then, for almost every finite data set, the loss function $\mathcal{L}$ possesses a generalized local minimum in the extended parameter space with some coordinates being infinite. The generalized local minimum is suboptimal whenever data set and neural network are such that a constant function is not an optimal solution.*

3.2 Non-increasing Path to a Global Minimum

By definition, all points belonging to a non-attracting region of local minima R are local minima with the same loss value. Further, being non-attractive means that every neighborhood of R contains points from where a non-increasing path to a value less than the value of the region exists. The question therefore arises under what conditions there is such a non-increasing path all the way down to a global minimum from almost everywhere in parameter space. The measure-theoretic term **almost everywhere** here refers to the Lebesgue measure, i.e., a condition holds almost everywhere when it holds for all points except for a set of Lebesgue measure zero. If the last hidden layer is the extremely wide layer having more neurons than input patterns (for example consider an extremely wide

two-layer neural network), then indeed it holds true that non-increasing paths to the global minimum exist from almost everywhere in parameter space by the results of Nguyen and Hein (2017) (and Gori and Tesi (1992); Poston et al. (1991); Venturi et al. (2019)). We show the same conclusion to hold for extremely wide deep neural networks with sigmoid and tanh activations, whenever the sequence of hidden dimensions is non-increasing, $n_{l+1} \leq n_l$, for all layers following the wide layer. The existence of such paths was also recently shown for the same deep and wide networks but with LeakyReLU activations by Nguyen (2019).

Theorem 5 *Consider a fully connected regression neural network with activation function in the class $\mathcal{A}$ (as defined in the beginning of Section 3) equipped with the squared loss function for a finite data set. Assume that a hidden layer contains more neurons than the number of input patterns and the sequence of dimensions of all subsequent layers is non-increasing. Then, for each set of parameters $\mathbf{w}$ and all $\epsilon > 0$, there is $\mathbf{w}'$ such that $||\mathbf{w} - \mathbf{w}'|| < \epsilon$ and such that a path, non-increasing in loss from $\mathbf{w}'$ to a global minimum (where $f(x_\alpha) = y_\alpha$ for each α), exists.*

Corollary 6 *Consider an extremely wide, fully connected regression neural network with non-increasing hidden dimensions following the wide layer, activation function in the class $\mathcal{A}$ and trained to minimize the squared loss over a finite data set. Then all suboptimal local minima are contained in a non-attracting region of local minima.*

The rest of the paper contains the arguments leading to the given results and an experimental construction of local minima in a deep and wide network.

4. Notation

We fix additional notation aside the problem definition from Section 3. For input x_α we denote the pattern vector of values at all neurons at layer l before activation by $\mathsf{n}(l; x_\alpha)$ and after activation by $\mathsf{act}(l; x_\alpha)$.

In general, we will denote column vectors of size n with coefficients z_i by $[z_i]_{1 \leq i \leq n}$ or simply $[z_i]_i$ and matrices with entries $a_{i,j}$ at position (i, j) by $[a_{i,j}]_{i,j}$. The neuron value pattern $\mathsf{n}(l; x_\alpha)$ is then a vector of size n_l denoted by $\mathsf{n}(l; x_\alpha) = [\mathsf{n}(l, k; x_\alpha)]_{1 \leq k \leq n_l}$, and the activation pattern $\mathsf{act}(l; x_\alpha) = [\mathsf{act}(l, k; x_\alpha)]_{1 \leq k \leq n_l}$.

For a labeled data point (x_α, y_α), we will further denote the squared loss on it by ℓ_α. The loss ℓ_α can be considered as a function of the neuron values $\mathsf{n}(l, k; x_\alpha)$, so that we consider partial derivatives of the loss ℓ_α with infinitesimal changes of neuron values at $\mathsf{n}(l, k; x_\alpha)$. For convenience of the reader, a tabular summary of all notation is provided in Appendix B.1–3.

5. Construction of Local Minima

We recall the construction of suboptimal local minima given by Fukumizu and Amari (2000) and extend it to deep networks. Once we have fixed a layer l, we denote the parameters of the incoming linear transformation by $[u_{p,i}]_{p,i}$, so that $u_{p,i}$ denotes the contribution of neuron i in layer $l - 1$ to neuron p in layer l, and the parameters of the outgoing linear transformation by $[v_{s,q}]$, where $v_{s,q}$ denotes the contribution of neuron q in layer l to neuron

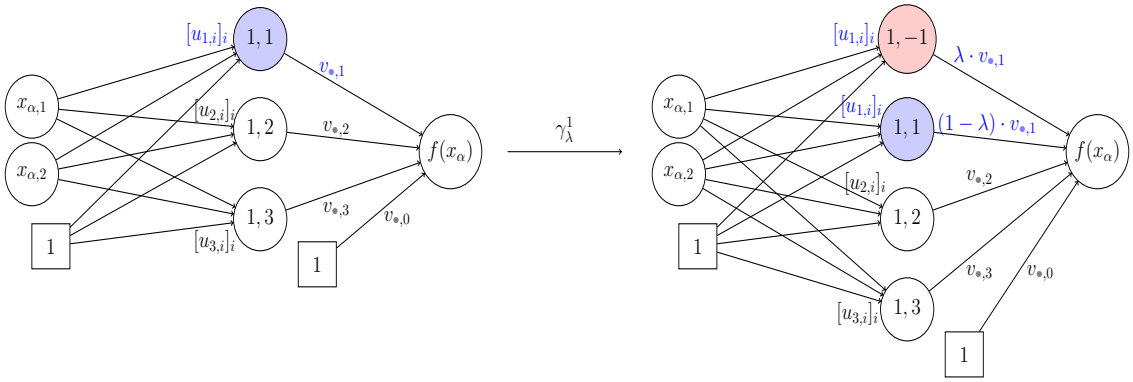


Figure 2: Embedding a smaller two-layer neural network into a larger one. Weights of the larger network are defined by the weights of the smaller network and the embedding map γ_λ^1. Numbers in hidden nodes (circles) denote the index of a neuron in form of (layer, neuron index) with negative index for the added neuron. Rectangles correspond to bias terms.

s in layer $l+1$. For weights of the output layer (into a single neuron), we write $w_{\bullet,j}$ instead of $w_{1,j}$. For the construction of **critical points** (i.e., points with gradient zero), we add one additional neuron $\mathsf{n}(l,-1;x)$ to a hidden layer l. (Negative indices are unused for neurons, which allows us to add a neuron with this index.)

A function γ_λ^r describes the mapping from the parameters of the original network to the parameters after adding a neuron $\mathsf{n}(l,-1;x)$. For a chosen neuron with index r in layer l of the smaller network, γ_λ^r is determined by incoming weights $u_{-1,i}$ into $\mathsf{n}(l,-1;x)$, outgoing weights $v_{s,-1}$ of $\mathsf{n}(l,-1;x)$, and a change of the outgoing weights $v_{s,r}$ of $\mathsf{n}(l,r;x)$. Sorting the network parameters in a convenient way, the embedding of the smaller network into the larger one is given, for any $\lambda \in \mathbb{R}$, by a function γ_λ^r mapping parameters $\{([u_{r,i}]_i, [v_{s,r}]_s, \bar{\mathbf{w}}\}$ of the smaller network to parameters $\{([u_{-1,i}]_i, [v_{s,-1}]_s, [u_{r,i}]_i, [v_{s,r}]_s, \bar{\mathbf{w}})\}$ of the larger network and is defined by

$$\gamma_\lambda^r\left([u_{r,i}]_i, [v_{s,r}]_s, \bar{\mathbf{w}}\right) := \left([u_{r,i}]_i, [\lambda \cdot v_{s,r}]_s, [u_{r,i}]_i, [(1-\lambda) \cdot v_{s,r}]_s, \bar{\mathbf{w}}\right).$$

Here $\bar{\mathbf{w}}$ denotes the collection of all remaining network parameters, i.e., all $[u_{p,i}]_i, [v_{s,q}]_s$ for $p, q \notin \{-1, r\}$ and all parameters from linear transformation of layers with index smaller than l or larger than $l+1$, if existent. A visualization of γ_λ^1 is shown in Figure 2.

Important fact: For the network functions φ, f of smaller and larger network at parameters $([u_{r,i}^*]_i, [v_{s,r}^*]_s, \bar{\mathbf{w}}^*)$ and $\gamma_\lambda^r([u_{r,i}^*]_i, [v_{s,r}^*]_s, \bar{\mathbf{w}}^*)$ respectively, we have $\varphi(x) = f(x)$ for all x. More generally, the activation values of all neurons in the smaller network agree with the activation values of their corresponding neuron in the larger network, i.e., $\mathsf{n}^\varphi(l,k;x) = \mathsf{n}(l,k;x)$ and $\mathsf{act}^\varphi(l,k;x) = \mathsf{act}(l,k;x)$ for all l, x and $k \geq 0$. This is independent of $\lambda \in \mathbb{R}$ and the continuous change of λ defines a direction in the parameter space with constant loss value.

5.1 Characterization of Critical Points Constructed Hierarchically by γ

Using some γ^r_λ to embed a smaller deep neural network into a larger one with one additional neuron, it has been shown that critical points get mapped to critical points.

Theorem 7 (Nitta (2017)) *Consider two neural networks as in Section 3, which differ by one neuron in layer l with index $\mathsf{n}(l,-1;x)$ in the larger network. If parameter choices $([u^*_{r,i}]_i, [v^*_{s,r}]_s, \bar{\mathbf{w}}^*)$ determine a critical point for the squared loss over a finite data set in the smaller network then, for each $\lambda \in \mathbb{R}$, $\gamma^r_\lambda([u^*_{r,i}]_i, [v^*_{s,r}]_s, \bar{\mathbf{w}}^*)$ determines a critical point in the larger network.*

As a consequence, whenever an embedding of a local minimum with γ^r_λ into a larger network does not lead to a local minimum, then it leads to a **saddle point** instead, i.e., critical points where the Hessian has both strictly positive and strictly negative eigenvalues. (There are no local maxima in the networks we consider, since the loss function is convex with respect to the parameters of the last layer.) The invariance of the network function under a change of λ shows that the critical points have a degenerate direction of constant loss. For shallow neural networks with one hidden layer, it was characterized when a critical point leads to a (degenerate) local minimum.

Theorem 8 (Fukumizu and Amari (2000)) *Consider two neural networks as in Section 3 with only one hidden layer and which differ by one neuron in the hidden layer with index $\mathsf{n}(1,-1;x)$ in the larger network. Assume that parameters $([u^*_{r,i}]_i, v^*_{\bullet,r}, \bar{\mathbf{w}}^*)$ determine an isolated local minimum for the squared loss over a finite data set in the smaller neural network and that $\lambda \notin \{0,1\}$.*

Then $\gamma^r_\lambda([u^*_{r,i}]_i, v^*_{\bullet,r}, \bar{\mathbf{w}}^*)$ determines a local minimum in the larger network if the matrix $[B^r_{i,j}]_{i,j}$ given by

$$B^r_{i,j} = \sum_\alpha (f(x_\alpha) - y_\alpha) \cdot v^*_{\bullet,r} \cdot \sigma''(\mathsf{n}(1,r;x_\alpha)) \cdot x_{\alpha,i} \cdot x_{\alpha,j}$$

is positive definite and $0 < \lambda < 1$, or if $[B^r_{i,j}]_{i,j}$ is negative definite and $\lambda < 0$ or $\lambda > 1$. (Here, we denote the k-th input dimension of input x_α by $x_{\alpha,k}$.)

We extend the previous theorem to a characterization in the case of deep neural networks. We note that a similar computation has been previously performed for neural networks with two hidden layers by Mizutani and Dreyfus (2010).

Theorem 9 *Consider two (possibly deep) neural networks as in Section 3, which differ by one neuron in layer l with index $\mathsf{n}(l,-1;x)$ in the larger network. Assume that the parameter choices $([u^*_{r,i}]_i, [v^*_{s,r}]_s, \bar{\mathbf{w}}^*)$ determine an isolated local minimum for the squared loss over a finite data set in the smaller network. If the matrix $[B^r_{i,j}]_{i,j}$ defined by*

$$B^r_{i,j} := \sum_\alpha \sum_k \frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1,k;x_\alpha)} \cdot v^*_{k,r} \cdot \sigma''(\mathsf{n}(l,r;x_\alpha)) \cdot \mathsf{act}(l-1,i;x_\alpha) \cdot \mathsf{act}(l-1,j;x_\alpha) \quad (1)$$

is either

- *positive definite and* $\lambda \in \mathcal{I} := (0, 1)$*, or*
- *negative definite and* $\lambda \in \mathcal{I} := (-\infty, 0) \cup (1, \infty)$*,*

then $\left\{\gamma_\lambda^r([u_{r,i}^*]_i, [v_{s,r}^*]_s, \bar{\mathbf{w}}^*) \mid \lambda \in \mathcal{I}\right\}$ *determines a non-attracting region of local minima in the larger network if and only if*

$$D_i^{r,s} := \sum_\alpha \frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1, s; x_\alpha)} \cdot \sigma'(\mathsf{n}(l, r; x_\alpha)) \cdot \mathsf{act}(l-1, i; x_\alpha) \tag{2}$$

is zero, $D_i^{r,s} = 0$*, for all* i, s*.*

Remark 10 *In the case of a neural network with only one hidden layer as considered in Theorem 8, the partial derivative* $\frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1,s;x_\alpha)}$ *reduces to the residual* $(f(x_\alpha) - y_\alpha)$ *and the matrix* $[B_{i,j}^r]_{i,j}$ *in Equation 1 reduces to the matrix* $[B_{i,j}^r]_{i,j}$ *in Theorem 8. The condition that* $D_i^{r,s} = 0$ *for all* i, s *does hold for shallow neural networks with one hidden layer as we show in Proposition 15(*i*). This proves Theorem 9 to be consistent with Theorem 8.*

Remark 11 *The assumption of starting from an isolated local minimum can be relaxed by special consideration of the degenerate directions (the eigenvectors of the Hessian with eigenvalue zero), which we will take advantage of.*

The theorem follows from the computation of the Hessian of the loss function $\mathcal{L}(\mathbf{w})$ (i.e., we calculate the matrix of second order derivatives of the loss function with respect to the network parameters), characterizing when it is positive (or negative) semidefinite and checking that the loss function does not change along directions that correspond to an eigenvector of the Hessian with eigenvalue 0. We state the outcome of the computation of the loss Hessian in Lemma 12 and refer the reader interested in a full proof of Theorem 9 to Appendix A.1.

Lemma 12 *Consider two (possibly deep) neural networks as in Section 3, which differ by one neuron in layer* l *with index* $\mathsf{n}(l, -1; x)$ *in the larger network. Fix* $1 \leq r \leq n_l$*. Assume that the parameter choices* $([u_{r,i}^*]_i, [v_{s,r}^*]_s, \bar{\mathbf{w}}^*)$ *determine a critical point in the smaller network.*

Let $\mathcal{L}$ *denote the loss function of the larger network and* ℓ *the loss function of the smaller network. Let* $\alpha \neq -\beta \in \mathbb{R}$ *such that* $\lambda = \frac{\beta}{\alpha+\beta}$*.*

With respect to the basis of the parameter space of the larger network given by $([u_{-1,i} + u_{r,i}]_i, [v_{s,-1} + v_{s,r}]_s, \bar{\mathbf{w}}, [\alpha \cdot u_{-1,i} - \beta \cdot u_{r,i}]_i, [v_{s,-1} - v_{s,r}]_s)$*, the Hessian of the loss* $\mathcal{L}$ *at* $\gamma_\lambda^r([u_{r,i}^*]_i, [v_{s,r}^*]_s, \bar{\mathbf{w}}^*)$ *is given by*

$$\begin{pmatrix}
[\frac{\partial^2 \ell}{\partial u_{r,i} \partial u_{r,j}}]_{i,j} & 2[\frac{\partial^2 \ell}{\partial u_{r,i} \partial v_{s,r}}]_{i,s} & [\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}}\ \partial u_{r,i}}]_{i,\bar{\mathbf{w}}} & 0 & 0 \\
2[\frac{\partial^2 \ell}{\partial u_{r,i} \partial v_{s,r}}]_{s,i} & 4[\frac{\partial^2 \ell}{\partial v_{s,r} \partial v_{t,r}}]_{s,t} & 2[\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}}\ \partial v_{s,r}}]_{s,\bar{\mathbf{w}}} & (\alpha - \beta)[D_i^{r,s}]_{s,i} & 0 \\
[\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}}\ \partial u_{r,i}}]_{\bar{\mathbf{w}},i} & 2[\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}}\ \partial v_{s,r}}]_{\bar{\mathbf{w}},s} & [\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}}\ \partial \bar{\mathbf{w}}'}]_{\bar{\mathbf{w}},\bar{\mathbf{w}}'} & 0 & 0 \\
0 & (\alpha - \beta)[D_i^{r,s}]_{i,s} & 0 & \alpha\beta[B_{i,j}^r]_{i,j} & (\alpha + \beta)[D_i^{r,s}]_{i,s} \\
0 & 0 & 0 & (\alpha + \beta)[D_i^{r,s}]_{s,i} & 0
\end{pmatrix}$$

5.2 Shallow Networks with a Single Hidden Layer

For the construction of suboptimal local minima in extremely wide two-layer networks, we begin by following the experiments of Fukumizu and Amari (2000) that prove the existence of suboptimal local minima in (non-wide) two-layer neural networks.

Consider a neural network of size 1—2—1. We use the corresponding network function f to construct a data set $(x_\alpha, y_\alpha)_{\alpha=1}^N$ by randomly choosing x_α and letting $y_\alpha = f(x_\alpha)$. By construction, we know that a neural network of size 1—2—1 can perfectly fit the data set with zero error.

Consider now a smaller network of size 1—1—1 having too little expressibility for a global fit of all data points. We find parameters $[u^*_{1,1}, v^*_{\bullet,1}]$ where the loss function of the neural network is in an isolated local minimum with non-zero loss. For this small example, the required positive definiteness of $[B^1_{i,j}]_{i,j}$ from Equation 1 for a use of γ^1_λ with $\lambda \in (0,1)$ reduces to checking a real number $B^1_{1,1}$ for positivity. An empirical example is easily found, and we assume the positivity condition to hold true. We can now apply γ^1_λ and Theorem 8 to find parameters for a neural network of size 1—2—1 that determine a suboptimal local minimum. This concludes the construction of Fukumizu and Amari (2000). The obtained network now serves as a base for a proof by induction to show that suboptimal local minima also exist in arbitrarily wide neural networks.

Theorem 13 *There is an extremely wide two-layer neural network with sigmoid activation functions and arbitrarily many neurons in the hidden layer that has a non-attracting region of suboptimal local minima.*

Proof Having already established the existence of parameters for a (small) neural network leading to a suboptimal local minimum, it suffices to note that iteratively adding neurons using Theorem 8 is possible. Iteratively at step t, we add a neuron $\mathsf{n}(1, -t; x)$ (negatively indexed) to the network by an application of γ^1_λ with the same $\lambda \in (0,1)$, using the same neuron with index 1 in each iteration. The corresponding matrix from Equation 1 remains a positive number

$$B^{1,(t)}_{1,1} = \sum_\alpha (f(x_\alpha) - y_\alpha) \cdot (1-\lambda)^{t-1} \cdot v^*_{\bullet,1} \cdot \sigma''(\mathsf{n}(l, 1; x_\alpha)) \cdot x^2_{\alpha,1}.$$

(We use here that neither $f(x_\alpha)$ nor $\mathsf{n}(l, 1; x_\alpha)$ ever change during this construction and that the outgoing weight from the hidden neuron with index 1 changes by multiplication with $(1-\lambda)$.) The idea is to apply Theorem 8 to guarantee that adding neurons as above always leads to a suboptimal minimum with nonzero loss for the network for $\lambda \in (0,1)$. The only problem is that the addition of previous neurons creates degenerate local minima with non-trivial kernel of the Hessian, i.e., the assumption in Theorem 8 of having an isolated minimum is violated. However, the computation of Lemma 12 is still valid and to ensure a local minimum from a positive semi-definite Hessian it is only necessary to additionally make sure that no reduction is possible along the degenerate directions in the kernel of the Hessian. When adding the t-th neuron, these degenerate directions in parameter space are defined by $v_{\bullet,1} - v_{\bullet,-t'}$ for $1 \leq t' \leq t$, since weight can be redistributed from $v_{\bullet,1}$ to $v_{\bullet,-t'}$ without a change of the network function. (This corresponds to the invariance of the network function under a change of $\lambda \in \mathbb{R}$ used in γ^1_λ when adding a neuron with index

$-t'$.) Since the redistribution of weight from $v_{\bullet,1}$ to $v_{\bullet,-t'}$ for some t' does not affect the possibility to redistribute weight from $v_{\bullet,1}$ to $v_{\bullet,-t''}$ for some other t'', the network function and hence the loss remain constant along any direction in the kernel of the Hessian In this case, positive semi-definiteness is sufficient to ensure a local minimum.

In particular, we may add an arbitrary number of neurons to the hidden layer and make the network extremely wide. Further, a continuous change of the λ belonging to the last added neuron via γ_λ to a value outside of $[0, 1]$ does not change the network function, but eventually leads to a saddle point (as we introduce a negative factor $\alpha\beta < 0 \Leftrightarrow \lambda \notin (0, 1)$ into the calculation of the Hessian at the position of $\alpha\beta B^{1,(t)}_{1,1}$, see Lemma 12). Hence, we found a non-attracting region of suboptimal minima. ■

Remark 14 *Since we started the construction from a network of size 1—1—1, our constructed example is extremely degenerate: The suboptimal local minima of the wide network have identical incoming weight vectors for each hidden neuron. Obviously, the suboptimality of this parameter setting is easily discovered by inspection of the parameters. Also with proper initialization, the chance of landing in this local minimum is vanishing.*

However, one may also start the construction from a more complex network with a larger network with several hidden neurons. In this case, when adding a few more neurons using γ^1_λ, it is much harder to detect the suboptimality of the parameters from visual inspection.

5.3 Deep Neural Networks

According to Theorem 9, for deep networks there is a second condition for the construction of local minima using the map γ^r_λ. Next to positive definiteness of the matrix $B^r_{i,j}$ for some r, we additionally require that $[D^{r,s}_i]_{i,s} = 0$ for all i, s and the same r. We consider sufficient conditions for $D^{r,s}_i = 0$.

Proposition 15 *Suppose we have constructed a critical point of the squared loss of a neural network by starting from a local minimum of a smaller network and by adding a neuron into layer l with index $\mathsf{n}(l, -1; x)$ by application of the map γ^r_λ to a neuron $\mathsf{n}(l, r; x)$. Suppose further that for the outgoing weights $v^*_{s,r}$ of $\mathsf{n}(l, r; x)$ we have $\sum_s v^*_{s,r} \neq 0$, and suppose that $D^{r,s}_i$ is defined as in Equation 2. Then $D^{r,s}_i = 0$ if one of the following holds.*

(i) The layer l is the last hidden layer. (This condition includes the case $l = 1$ indexing the hidden layer in a two-layer network.)

(ii) For all t, t', α, we have

$$\frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1, t; x_\alpha)} = \frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1, t'; x_\alpha)}.$$

(iii) For each α and each t,

$$\frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1, t; x_\alpha)} = 0.$$

(This condition holds in the case of the weight infinity attractors in the proof to Theorem 4 for $l + 1$ the second last layer. It also holds in a global minimum.)

The proof of the proposition is contained in Appendix A.3, and we apply it to construct a non-attracting region of suboptimal local minima in a deep and wide neural network.

Proof (Theorem 2) For simplicity of the presentation, we show the existence of local minima for a three-layer neural network, but the same construction naturally generalizes to deeper networks. We begin with a regression network of size 2—n_1—n_2—1 for input dimension $n_0 = 2$, hidden layers of dimension n_1, n_2, and the output layer mapping into $\mathbb{R}$. We use this network to construct a finite data set and train a network of size 2—1—1—1 with one neuron in each hidden layer on this data set to find an isolated local minimum.

In Equations 1 and 2 we have suppressed the choice of the layer to simplify the notation. To distinguish layers here, we write $[B^r_{i,j}(1)]_{i,j}$, $[D^{r,s}_i(1)]_{i,s}$ and $[B^r_{i,j}(2)]_{i,j}$, $[D^{r,s}_i(2)]_{i,s}$ for the matrices of the first and second hidden layer respectively. We assume that the local minimum satisfies that $[B^1_{i,j}(1)]_{i,j}$ is positive definite and $B^1_{1,1}(2) > 0$. Existence of such an example is easily verified empirically and we provide an example in the following section. Starting with the first hidden layer, the condition of Proposition 15 (ii) is trivially satisfied as there is only one neuron in the second layer. Hence $D^{1,1}_1(1) = 0$ and we can we can iteratively add arbitrarily many neurons to the first hidden layer by repetitively applying γ^1_λ on $\mathsf{n}(1, 1; x)$ and Theorem 9 (with $\lambda \in (0, 1)$). (Precisely as in the iterative application of Theorem 8 in the proof of Theorem 13, we can see that in this iterative process the assumption of starting from an isolated local minimum can be relaxed.) When adding the t-th neuron with negative index $-t$, the relevant matrices are given by $[B^{1,(t)}_{i,j}(1)]_{i,j} = (1-\lambda)^{t-1}[B^{1,(1)}_{i,j}(1)]_{i,j}$ which are positive definite, and $D^{1,1}_1(1)$ does not change and remains zero. We can therefore find a local minimum in a network of size 2—m_1—1—1 for any m_1.

For the second layer, the matrix $B(2) := [B^1_{i,j}(2)]_{i,j}$ equals the constant matrix of size $(m_1 \times m_1)$ with entry the positive number $B^1_{1,1}(2)$ calculated before adding neurons to the first layer and is positive semidefinite. Now, Proposition 15 (i) applies to show that we can apply Theorem 9 to iteratively add arbitrarily many neurons to the second hidden layer using γ^1_λ on $\mathsf{n}(2, 1; x)$. When adding the s-th neuron with negative index $-s$ to the second layer, we have $[B^{1,(s)}_{i,j}(2)]_{i,j} = (1-\lambda)^{s-1}B(2)$, which is positive semidefinite. Since the matrix $B(2)$ is only positive semidefinite and not positive definite, we again need special consideration to the degenerate directions: With $u_{\cdot,\cdot}$ and $v_{\bullet,\cdot}$ the weights from the second and third layer respectively, the kernel of the Hessian for m_1 neurons in the first and m_2 neurons in the second layer is given by $\ker(H) = \operatorname{span}\{u_{1,1} - u_{1,-t}, u_{-s,1} - u_{-s,-t}, v_{\bullet,1} - v_{\bullet,-s} \mid 1 \le t \le m_1,\ 1 \le s \le m_2\}$. Since the loss is constant in the direction of any parameter change in the kernel of H, we can exclude the possibility of loss reduction along degenerate directions and positive semi-definiteness is sufficient to guarantee a local minimum.

This leads to a local minimum in a network of size 2—m_1—m_2—1. If $m_1 \ge n_1$, $m_2 \ge n_2$, then we know the local minimum to be suboptimal by construction. Adding sufficiently many neurons, we can construct the network to be extremely wide. A continuous change of the parameter $\lambda \in (0, 1)$ used for adding the t-th neuron, changes the sign of the corresponding B-matrix for negative λ, which offers a direction in weight space for further reduction of the loss. This shows the existence of a non-attracting region in a deep and wide neural network, proving Theorem 2. ■

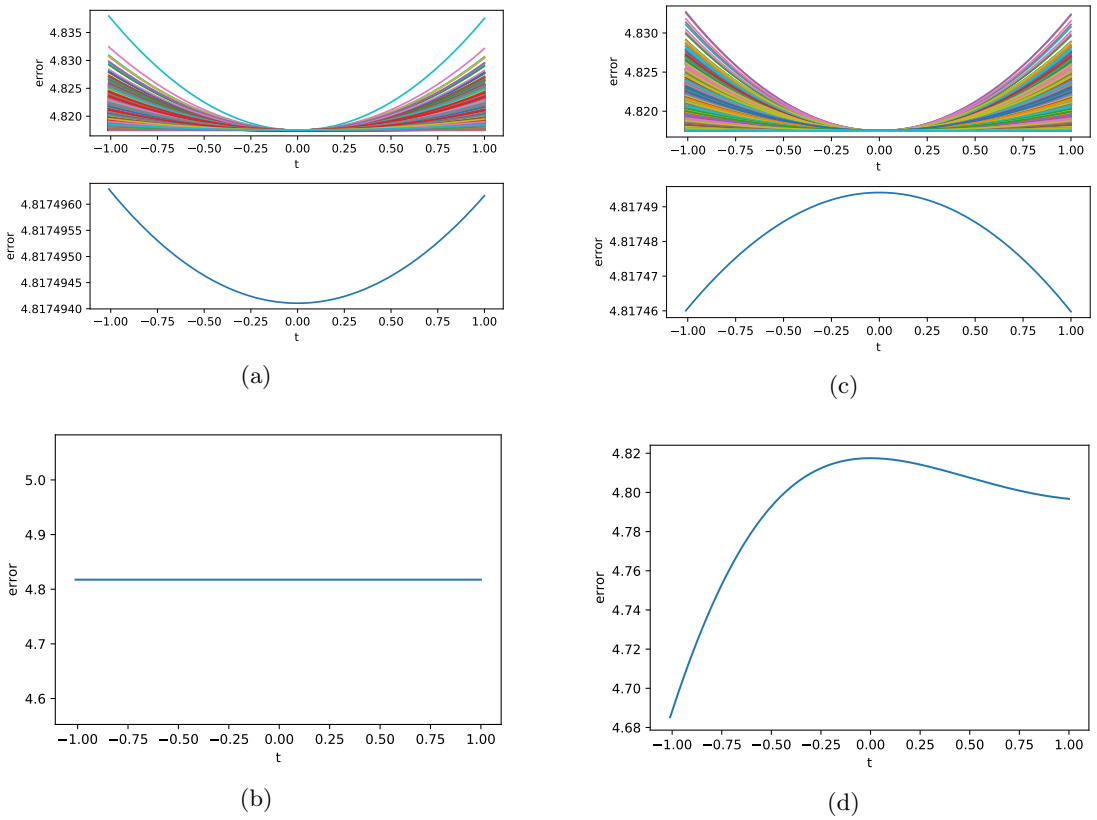


Figure 3: A non-attracting region of local minima in a deep and wide neural network. (a) Local minimum. *Top:* Loss evolution for 5000 random directions. *Bottom:* Minimum over sampled directions. (b) Path along a degenerate direction to a saddle point. (c) Saddle point with the same loss value. *Top:* Loss evolution for 5000 random directions. *Bottom:* Minimum over sampled directions. (d) Error evolution along an analytically known direction of descent at the saddle point.

5.4 Experiment for Deep Neural Networks

We empirically validate the construction of a suboptimal local minimum in a deep and extremely wide neural network as given in the proof of Theorem 2.[3] We start by considering a three-layer network of size 2—5—5—1, i.e., we have two input dimensions, one output dimension and hidden layers of five neurons. We use its network function f to create a data set of 20 samples $(x_\alpha, f(x_\alpha))$, hence we know that a network of size 2—5—5—1 can attain zero loss.

We initialize a new neural network of size 2—1—1—1 and train it until convergence to find a local minimum of total loss (sum over the 20 data points) of 4.817. We check for positive definiteness of the matrix $B^1_{i,j}(1)$ (eigenvalues here given by 0.0182, 0.0004)

3. The accompanying code can be found at `https://github.com/petzkahe/nonattracting_regions_of_local_minima.git`.

and positivity of $B^1_{1,1}(2)$ (here 0.0757). Following the proof to Theorem 2, we add twenty neurons to both hidden layers to construct a local minimum in an extremely wide network of size 2—21—21—1. The local minimum must be suboptimal by construction of the data set. Experimentally, we show not only that indeed we end up with a suboptimal minimum, but also that it belongs to a non-attracting region of local minima.

Figure 3 shows results of this construction. The plot in (a) shows the loss in the neighborhood of the local minimum in parameter space. The top image shows the loss curve into 5000 randomly generated directions, the bottom displays the minimal loss over all these directions. Evidently, we were not able to find a direction in parameter space that allows us to reduce the loss. The plot in (b) shows the change of loss along one of the degenerate directions that leads to a saddle point, which is shown in plot (c). Again, the top image shows the loss curve into 5000 randomly generated directions, and the bottom displays the minimum loss over all these directions. Most random directions in parameter space lead to an increase in loss, but the minimum loss shows the existence of directions that lead to a reduction of loss. In such a saddle point, we know a direction for loss reduction at this saddle point from Lemma 12. The plot in (d) shows a significant reduction in loss for the analytically known direction. Being able to reach a saddle point from a local minimum by a path of non-increasing loss shows that we found a non-attracting region of local minima.

5.5 A Discussion of Limitations and of the Loss of Non-attracting Regions of Suboptimal Minima

We theoretically proved the possibility of suboptimal local minima in deep and wide networks and empirically validated their existence. Due to the degeneracy of the constructed examples, the following questions on consequences in practice remain unanswered. (i) How frequent are suboptimal local minima with high loss, and (ii) how degenerate must they be to have high loss?

Suppose we aim to find a suboptimal local minimum using the above construction in a network consisting of L layers with hidden dimensions n_l, and we want the local minimum to be less degenerate than the above examples while still having considerably high loss. We therefore want to start from a local minimum in a smaller network with hidden layer dimension ν_i and add only a few neurons to each of the layers. We want each ν_i small enough to find a (non-degenerate) local minimum in the smaller network with large loss, but not too small so that the addition of several neurons renders the local minimum degenerate. For the construction to work, we need to find in each layer l a neuron with index r_l such that the n_{l-1}-many eigenvalues of the corresponding B-matrix are either all positive or all negative so that the matrix is either positive definite or negative definite (determining a suitable choice for λ according to Theorem 9). In addition, the same neuron must satisfy $D_i^{r_l,s} = 0$ for all i, s, adding $n_{l-1}n_{l+1}$ many conditions. To find such an example is difficult both theoretically (the sufficient conditions from Proposition 15 for a vanishing D-matrix are rather strong) as well as empirically. Whenever any of the necessary conditions is violated, then we cannot use the above construction to find a local minimum in a larger network. In other words, whenever we find a local minimum in the smaller network such that no neuron exists that satisfies all the necessary conditions (positive definiteness of B and zero D-matrix), then the above construction leads to a saddle point. While these saddle

points can be close to being a local minimum (in the sense that slight perturbations of the parameters yields a higher loss in most cases), there is at least one direction in parameter space with negative curvature of the loss surface.

It is therefore conceivable that suboptimal local minima with high loss are extremely rare in practical applications with sufficiently large networks, which is in line with empirical observations. From this perspective, our results suggest that for sufficiently wide deep networks *almost* all local minima are global, but the existence of degenerate local minima makes a general statement impossible.

6. Proving the Existence of a Non-increasing Path to the Global Minimum

In the previous section we showed the existence of non-attracting regions of local minima and Theorem 4 showed the existence of generalized local minima at infinity that can cause divergent parameters during loss reduction. These type of local minima do not rule out the possibility of non-increasing paths to the global minimum from almost everywhere in parameter space. In this section, we sketch the proof to Theorem 5 illustrated in form of several lemmas, where up to the basic assumptions on the neural network structure as in Section 3 (with activation function in $\mathcal{A}$), the assumption of one lemma is given by the conclusion of the previous ones. The corresponding proofs can be found in Appendix A.4.

We consider vectors that we call activation vectors, different from the activation pattern vectors $\mathsf{act}(l; x)$ from above. The **activation vector** at neuron k in layer l is denoted by $\mathbf{a}_k^l$, and defined by all values at the given neuron for samples x_α:

$$\mathbf{a}_k^l := [\mathsf{act}(l, k; x_\alpha)]_\alpha.$$

In other words while we fix l and x for the activation pattern vectors $\mathsf{act}(l; x)$ and let k run over its possible values, we fix l and k for the activation vectors $\mathbf{a}_k^l$ and let x run over all samples x_α in the data set. We denote by $\mathbf{a}^l$ the matrix $[\mathbf{a}_k^l]_k = [\mathsf{act}(l, k; x_\alpha)]_{k,\alpha}$ of size $(n_l \times N)$ containing the activation values of all neurons and samples at layer l. Similarly, we denote by $\mathbf{n}^l$ the matrix $\mathbf{n}^l = [\mathsf{n}(l, k; x_\alpha)]_{k,\alpha}$ of size $(n_l \times N)$ containing the pre-activation neuron values for all neurons and samples at layer l

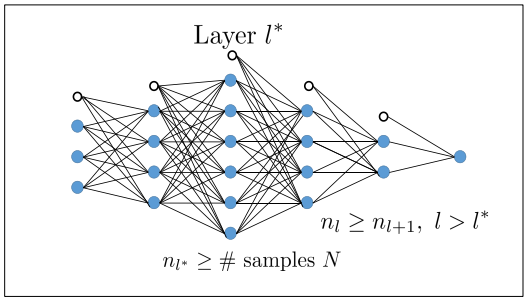
The **first step** of the proof is to use the freedom given by $\epsilon > 0$ to change the starting point in parameter space to satisfy that the activation vectors $\mathbf{a}_k^{l^*}$ of the extremely wide layer l^* span the entire space $\mathbb{R}^N$.

Lemma 16 *(Nguyen and Hein, 2017, Corollary 4.5) For each choice of parameters* $\mathbf{w}$ *and all* $\epsilon > 0$ *there is* $\mathbf{w}'$ *such that (i)* $||\mathbf{w} - \mathbf{w}'|| < \epsilon$*, (ii) the activation vectors* $\mathbf{a}_k^{l^*}$ *of the extremely wide layer* l^* *(containing more neurons than the number of training samples* N*) at parameters* $\mathbf{w}'$ *satisfy*

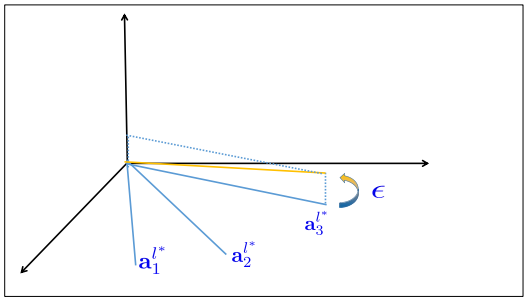
$$\operatorname{span}_k \mathbf{a}_k^{l^*} = \mathbb{R}^N,$$

and (iii) the weight matrices $(\mathbf{w}')^l$ *have full rank for all* $l > l^* + 1$*.*

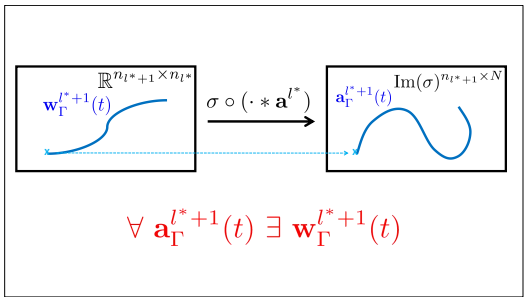
The **second step** of the proof is to guarantee that we can then induce any continuous change of activation vectors in layer l^*+1 by suitable paths in the parameter space changing only the weights of the same layer. The following two lemmas ensure exactly that. We first



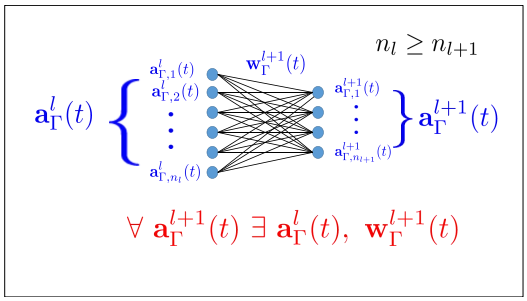
Theorem 5: Visualization of considered neural network architecture.



Lemma 16: An ϵ-change guarantees linear independence.



Lemma 17&18: Realizing paths of activation vectors by paths in parameter space



Lemma 18&19: Realizing paths of activation vectors for decreasing hidden dimensions.

Figure 4: Non-increasing paths to the global minimum exist from almost everywhere. Visualization of proof ideas.

consider pre-activation values and then consider the application of the activation function. We slightly abuse notation in the statement when adding a vector to a matrix, which shall mean the addition of the vector to all columns of the matrix.

Lemma 17 *Assume that in the extremely wide layer l^* we have that the activation vectors at a set of parameters $\mathbf{w}$ satisfy $\mathrm{span}_k[\mathbf{a}_k^{l^*}] = \mathbb{R}^N$. Then, for any continuous path $\mathbf{n}_\Gamma^{l^*+1} : [0,1] \to \mathbb{R}^{n_{l^*+1} \times N}$ with starting point $\mathbf{n}_\Gamma^{l^*+1}(0) = \mathbf{n}^{l^*+1} = \mathbf{w}^{l^*+1} \cdot \mathbf{a}^{l^*} + \mathbf{w}_0^{l^*+1}$, there is a continuous path of parameters $\mathbf{w}_\Gamma^{l^*+1} : [0,1] \to \mathbb{R}^{n_{l^*+1} \times n_{l^*}}$ of the (l^*+1)-th layer with $\mathbf{w}_\Gamma^{l^*+1}(0) = \mathbf{w}^{l^*+1}$ and such that*

$$\mathbf{n}_\Gamma^{l^*+1}(t) = \mathbf{w}_\Gamma^{l^*+1}(t) \cdot \mathbf{a}^{l^*} + \mathbf{w}_0^{l^*+1}.$$

Lemma 18 *For all continuous paths $\mathbf{a}_\Gamma(t)$ in $Im(\sigma)^{n \times N}$, i.e., the $(n \times N)$-fold copy of the image of an activation function $\sigma \in \mathcal{A}$, there is a continuous path $\mathbf{n}_\Gamma(t)$ in $\mathbb{R}^{n \times N}$ such that $\mathbf{a}_\Gamma(t) = \sigma(\mathbf{n}_\Gamma(t))$ for all t.*

With activations values $\mathbf{a}^l$ depending on parameters $\mathbf{w}^\iota$ of previous layers with index $\iota \leq l$, we denote this functional dependence by $\mathbf{a}^l(\mathbf{w})$. We say that a continuous path $\mathbf{a}^l_\Gamma : [0,1] \to \mathrm{Im}(\sigma)^{n_l \times N}$ of activation values in the l-th layer is **realized by a path of parameters** $\mathbf{w}_\Gamma(t)$, if the path $\mathbf{w}_\Gamma(t)$ induces a change of activation values at layer l according to the desired path $\mathbf{a}^l_\Gamma$, i.e., $\mathbf{a}^l(\mathbf{w}_\Gamma(t)) = \mathbf{a}^l_\Gamma(t)$. Using this terminology, Lemma 17 and 18 show that in a layer following an extremely wide one, any continuous path of activation values can be realized by a path of parameters of the same layer.

The **third step** guarantees that, as long as the sequence of dimensions of subsequent hidden layers never increases, realizability of arbitrary paths in layer l implies realizability of arbitrary paths in layer $l+1$ for both activation and pre-activation values.

Lemma 19 *Assume that $n_{l+1} \leq n_l$ and that the weight matrix $\mathbf{w}^{l+1} \in \mathbb{R}^{n_{l+1} \times n_l}$ has full rank n_{l+1}. Let $\mathbf{a}^l = \mathbf{a}^l(\mathbf{w})$ and $\mathbf{n}^{l+1} = \mathbf{n}^{l+1}(\mathbf{w})$ be the matrices of activation values at the l-th layer and pre-activations at the $(l+1)$-th layer for parameters $\mathbf{w}$ respectively. Then, for any continuous path $\mathbf{n}^{l+1}_\Gamma : [0,1] \to \mathbb{R}^{n_{l+1} \times N}$ with $\mathbf{n}^{l+1}_\Gamma(0) = \mathbf{n}^{l+1} = \mathbf{w}^{l+1} \cdot \mathbf{a}^l + \mathbf{w}^{l+1}_0$, there are continuous paths of (full-rank) weight matrices $\mathbf{w}^{l+1}_\Gamma : [0,1] \to \mathbb{R}^{n_{l+1} \times n_l}$ and bias terms $\mathbf{w}^{l+1}_{\Gamma,0} : [0,1] \to \mathbb{R}^{n_{l+1}}$ in the $(l+1)$-th layer and a continuous path $\mathbf{a}^l_\Gamma : [0,1] \to Im(\sigma)^{n_l \times N}$ of activation values in the l-th layer, such that $\mathbf{w}^{l+1}_\Gamma(0) = \mathbf{w}^{l+1}$, $\mathbf{w}^{l+1}_{\Gamma,0}(0) = \mathbf{w}^{l+1}_0$, $\mathbf{a}^l_\Gamma(0) = \mathbf{a}^l$ and for all $t \in [0,1]$,*

$$\mathbf{n}^{l+1}_\Gamma(t) = \mathbf{w}^{l+1}_\Gamma(t) \cdot \mathbf{a}^l_\Gamma(t) + \mathbf{w}^{l+1}_{\Gamma,0}(t).$$

Combining Lemma 18 and 19, we can realize any continuous path of activation values $\mathbf{a}^{l+1}_\Gamma(t)$ in layer $l+1$ by a path of parameters, if arbitrary paths of activation values $\mathbf{a}^l_\Gamma(t)$ can be realized in the previous layer l. By Lemma 17 and 18, we can indeed realize arbitrary paths in the layer following the extremely wide layer. Hence, by induction over the layers we find that any path at the output is realizable. In the following result, we denote the dependence of the network function on its parameters $\mathbf{w}$ at a training sample x_α by $f(\mathbf{w}; x_\alpha)$.

Lemma 20 *Assume a neural network structure as above with activation vectors $\mathbf{a}^{l^*}_k$ of the extremely wide hidden layer spanning $\mathbb{R}^N$, hidden dimensions $n_{l+1} \leq n_l$ for all $l > l^*$ and weight matrices $\mathbf{w}^{l+1} \in \mathbb{R}^{n_{l+1} \times n_l}$ of full rank n_{l+1} for all $l > l^*$. Then for any continuous path $f_\Gamma : [0,1] \to \mathbb{R}^N$ with $f_\Gamma(0) = [f(\mathbf{w}; x_\alpha)]_\alpha$ there is a continuous path $\mathbf{w}_\Gamma(t)$ from the current weights $\mathbf{w}_\Gamma(0) = \mathbf{w}$ that realizes $f_\Gamma(t)$ as the output of the neural network function, $f_\Gamma(t) = [f(\mathbf{w}_\Gamma(t); x_\alpha)]_\alpha$.*

With fixed $z_\alpha = f(\mathbf{w}; x_\alpha)$, the prediction for the current weights, let $\mathbf{z} = [z_\alpha]_\alpha$ denote the vector of predictions, and let $\mathbf{y} = [y_\alpha]_\alpha$ denote the vector of all target values. An obvious path of decreasing loss at the output layer is then given by $f_\Gamma(t) = \mathbf{z} + t \cdot (\mathbf{y} - \mathbf{z})$, inducing the loss $\mathcal{L} = ||\mathbf{z} + t \cdot (\mathbf{y} - \mathbf{z}) - \mathbf{y}||_2^2 = (1-t)||\mathbf{y} - \mathbf{z}||_2^2$. This concludes the proof of Theorem 5 by applying Lemma 20 to this choice of $f_\Gamma(t)$ after a possible change of the starting parameters $\mathbf{w}$ to arbitrarily close parameters $\mathbf{w}'$ using Lemma 17.

We limit our contribution to Theorem 5 and its proof, but a slightly stronger version of the theorem could in principle be shown, stating that there exists a non-increasing path to a global minimum from anywhere in parameter space instead of the weaker existence

from *almost* everywhere. The following steps provide a sketch for an extended proof: (1) Inductively go through the layers preceding the wide layer l^* to find a path of constant loss that ensures mutually different activation patterns in the (l^*-1)-th layer at its endpoint. This requires a composition of arguments from Theorem 8 of Venturi et al. (2019) and Lemma 4.3 of Nguyen and Hein (2017); (2) Use the arguments from Theorem 8 of Venturi et al. (2019) to ensure part (ii) of Lemma 16 through a path of constant loss; (3) Use the arguments of Lemma 3.4 of Nguyen (2019) in combination with Lemma 18 and 19 to ensure part (iii) of Lemma 16 through a path of constant loss. Now Lemma 20 can be applied as before with the only difference that its assumptions are satisfied through a path of constant loss instead of an arbitrarily small displacement.

7. Conclusion

We have proved the existence of suboptimal local minima for regression neural networks with sigmoid activation functions of arbitrary width. We established that the nature of local minima is such that they live in a special region of the cost function called a non-attractive region, and showed that a non-increasing path to a configuration with lower loss than that of the region can always be found. For sufficiently wide neural networks with decreasing hidden layer dimensions after the extremely wide layer, all local minima belong to such a region. We generalized a procedure to find such regions in shallow networks, introduced by Fukumizu and Amari (2000), to deep networks and described conditions for the construction to work. The necessary conditions become hard to satisfy in wider and deeper networks and, if they fail, the construction leads to saddle points instead. The appearance of an additional condition when extending Fukumizu and Amari (2000)'s construction to deeper networks suggests that such hierarchically constructed local minima are rare and degenerate in deep networks, but their existence shows that no general statement about all local minima being global can be made.

Acknowledgments

This work was supported in part by the European Research Council Consolidator grant SEED, CNCS-UEFISCDI (PN-III-P4-ID-PCE-2016-0535, PN-III-P4-ID-PCCF-2016-0180), the EU Horizon 2020 grant DE-ENIGMA (688835), and SSF.

Appendix A. Proofs

A.1 Proofs for the Construction of Local Minima

Here we prove Theorem 9, which follows from two lemmas, with the first lemma being Lemma 12 containing the computation of the Hessian of the cost function $\mathcal{L}$ of the larger network at parameters $\gamma_\lambda^r([u_{r,i}^*]_i, [v_{s,r}^*]_s, \bar{\mathbf{w}}^*)$ with respect to a suitable basis.

Proof of Lemma 12 The proof only requires a tedious, but not complicated calculation (using the relation $\alpha\lambda - \beta(1-\lambda) = 0$ multiple times. To keep the argumentation streamlined, we moved all the necessary calculations into the supplementary material. ■

The second lemma determines when matrices of the form as calculated in Lemma 12 are positive definite.

Lemma A.1 *Let $a, b, c, d, e, f, g, h, x$ be matrices of appropriate sizes.*

(a) A matrix of the form

$$\begin{pmatrix} a & 2b & c & 0 \\ 2b^T & 4d & 2e & 0 \\ c^T & 2e^T & f & 0 \\ 0 & 0 & 0 & x \end{pmatrix}$$

is positive semidefinite if and only if both x and the matrix

$$\begin{pmatrix} a & b & c \\ b^T & d & e \\ c^T & e^T & f \end{pmatrix}$$

are positive semidefinite.

(b) A matrix x of the form

$$x = \begin{pmatrix} g & h \\ h^T & 0 \end{pmatrix}$$

is positive semidefinite if and only if g is positive semidefinite and $h = 0$.

Proof

(a) By definition, a matrix A is positive semidefinite if and only if $z^T A z \geq 0$ for all z. Note now that

$$(z_1^T, z_2^T, z_3^T, z_4^T) \begin{pmatrix} a & 2b & c & 0 \\ 2b^T & 4d & 2e & 0 \\ c^T & 2e^T & f & 0 \\ 0 & 0 & 0 & x \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = (z_1^T, 2z_2^T, z_3^T, z_4^T) \begin{pmatrix} a & b & c & 0 \\ b^T & d & e & 0 \\ c^T & e^T & f & 0 \\ 0 & 0 & 0 & x \end{pmatrix} \begin{pmatrix} z_1 \\ 2z_2 \\ z_3 \\ z_4 \end{pmatrix}$$

(b) It is clear that the matrix x is positive semidefinite for g positive semidefinite and $h = 0$. To show the converse, first note that if g is not positive semidefinite and z is such that $z^T g z < 0$ then

$$(z^T, 0) \begin{pmatrix} g & h \\ h^T & 0 \end{pmatrix} \begin{pmatrix} z \\ 0 \end{pmatrix} = z^T g z < 0.$$

It therefore remains to show that also $h = 0$ is a necessary condition. Assume $h \neq 0$ and find z such that $hz \neq 0$. Then for any $\lambda \in \mathbb{R}$ we have

$$((hz)^T, -\lambda z^T) \begin{pmatrix} g & h \\ h^T & 0 \end{pmatrix} \begin{pmatrix} hz \\ -\lambda z \end{pmatrix} = (hz)^T g(hz) - 2(hz)^T h \lambda z$$

$$= (hz)^T g(hz) - 2\lambda ||hz||_2^2.$$

For sufficiently large λ, the last term is negative.

■

In addition, to find local minima from positive semi-definiteness, one needs to explain away all degenerate directions, i.e., we need to show that the loss function actually does not change into the direction of eigenvectors of the Hessian with eigenvalue 0. Otherwise a higher derivative into this direction could be nonzero and potentially lead to a saddle point.
Proof of Theorem 9 In Lemma 12, we calculated the Hessian of $\mathcal{L}$ with respect to a suitable basis at a the critical point $\gamma_\lambda^r([u^*_{r,i}]_i, [v^*_{s,r}]_s, \bar{\mathbf{w}}^*)$. If the matrix $[D_i^{r,s}]_{i,s}$ is nonzero, then by Lemma A.1(b) the Hessian is not positive semidefinite, hence none of the critical points are local minima.

If, on the other hand, the matrix $[D_i^{r,s}]_{i,s}$ is zero, then by Lemma A.1(a+b) the Hessian is positive semidefinite, since

$$\begin{pmatrix} [\frac{\partial^2 \ell}{\partial u_{r,i} \partial u_{r,j}}]_{i,j} & [\frac{\partial^2 \ell}{\partial u_{r,i} \partial v_{s,r}}]_{i,s} & [\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}} \ \partial u_{r,i}}]_{i,\bar{\mathbf{w}}} \\ [\frac{\partial^2 \ell}{\partial u_{r,i} \partial v_{s,r}}]_{s,i} & [\frac{\partial^2 \ell}{\partial v_{s,r} \partial v_{t,r}}]_{s,t} & [\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}} \ \partial v_{s,r}}]_{s,\bar{\mathbf{w}}} \\ [\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}} \ \partial u_{r,i}}]_{\bar{\mathbf{w}},i} & [\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}} \ \partial v_{s,r}}]_{\bar{\mathbf{w}},s} & [\frac{\partial^2 \ell}{\partial \bar{\mathbf{w}} \ \partial \bar{\mathbf{w}}'}]_{\bar{\mathbf{w}},\bar{\mathbf{w}}'} \end{pmatrix}$$

is positive definite by assumption (isolated minimum), and $\alpha\beta[B^r_{i,j}]_{i,j}$ is positive definite if $\lambda \in (0,1) \Leftrightarrow \alpha\beta > 0$ and $[B^r_{i,j}]_{i,j}$ is positive definite, or if ($\lambda < 0$ or $\lambda > 1$) $\Leftrightarrow \alpha\beta < 0$ and $[B^r_{i,j}]_{i,j}$ is negative definite. In each case we can alter λ to values leading to saddle points without changing the network function or loss. Therefore, the critical points can only be saddle points or local minima on a non-attracting region of local minima.

To determine whether the critical points in question lead to local minima when $[D_i^{r,s}]_{i,s} = 0$, it is insufficient to only prove the Hessian to be positive semidefinite (in contrast to (strict) positive definiteness), but we need to consider directions for which the second order information is insufficient. We know that the loss is at a minimum with respect to all coordinates except for the degenerate directions defined by a change of $[v_{s,-1} - v_{s,r}]_s$ that keeps $[v_{s,-1} + v_{s,r}]_s$ constant. However, the network function $f(x)$ is constant along $[v_{s,-1} - v_{s,r}]_s$ (keeping $[v_{s,-1} + v_{s,r}]_s$ constant) at the critical point where $u_{-1,i} = u_{r,i}$ for all i. Hence, no higher order information leads to saddle points and it follows that the critical point lies on a region of local minima. ■

A.2 Local Minima at Infinity in Neural Networks

In this section we prove Theorem 4, showing the existence of local minima at infinity in neural networks.
Proof of Theorem 4 We will show that, if all bias terms $u_{i,0}$ of the last hidden layer are sufficiently large, then there are parameters $u_{i,k}$ for $k \neq 0$ and parameters v_i of the output layer such that the minimal loss is achieved at $u_{i,0} = \infty$ for all i.

We note that, if $u_{i,0} = \infty$ for all i, all neurons of the last hidden layer are fully active for all samples, i.e., $\mathsf{act}(L-1, i; x_\alpha) = 1$ for all i. Therefore, in this case $f(x_\alpha) = \sum_i v_{\bullet,i}$ for all α. A constant function $f(x_\alpha) = \sum_i v_{\bullet,i} = c$ minimizes the loss $\sum_\alpha (c - y_\alpha)^2$ uniquely for $c := \frac{1}{N}\sum_{\alpha=1}^N y_\alpha$. We will assume that the $v_{\bullet,i}$ are chosen such that $\sum_i v_{\bullet,i} = c$ does

hold. That is, for fully active hidden neurons at the last hidden layer, the $v_{\bullet,i}$ are chosen to minimize the loss.

We write $f(x_\alpha) = c + \epsilon_\alpha$. Then

$$\mathcal{L} = \frac{1}{2}\sum_\alpha (f(x_\alpha) - y_\alpha)^2 = \frac{1}{2}\sum_\alpha (c + \epsilon_\alpha - y_\alpha)^2$$

$$= \frac{1}{2}\sum_\alpha (\epsilon_\alpha + (c - y_\alpha))^2$$

$$= \underbrace{\frac{1}{2}\sum_\alpha (c - y_\alpha)^2}_{\text{Loss at } u_{i,0} = \infty \text{ for all } i} + \underbrace{\frac{1}{2}\sum_\alpha \epsilon_\alpha^2}_{\geq 0} + \underbrace{\sum_\alpha \epsilon_\alpha (c - y_\alpha)}_{(*)}.$$

The idea is now to ensure that $(*) \geq 0$ for sufficiently large $u_{i,0}$ and in a neighborhood of the $v_{\bullet,i}$ chosen as above. Then the loss $\mathcal{L}$ is larger than at infinity, and any such point in parameter space with $u_{i,0} = \infty$ and $v_{\bullet,i}$ with $\sum_i v_{\bullet,i} = c$ is a local minimum.

To study the behavior at $u_{i,0} = \infty$, we consider $p_i = \exp(-u_{i,0})$. Note that

$$\lim_{u_{i,0}\to\infty} p_i = 0.$$

We have

$$f(x_\alpha) = \sum_i v_{\bullet,i}\sigma(u_{i,0} + \sum_k u_{i,k}\mathsf{act}(L-2,k;x_\alpha))$$

$$= \sum_i v_{\bullet,i} \cdot \frac{1}{1 + p_i \cdot \exp(-\sum_k u_{i,k}\mathsf{act}(L-2,k;x_\alpha))}$$

Now for p_i close to 0 we can use Taylor expansion of $g(p_i) := \frac{1}{1+p_i\exp(q)}$ to get $g(p_i) = 1 - \exp(q)p_i + \mathcal{O}(|p_i|^2)$. Therefore

$$f(x_\alpha) = c - \sum_i v_{\bullet,i} p_i \exp(-\sum_k u_{i,k}\mathsf{act}(L-2,k;x_\alpha)) + \mathcal{O}(p_i^2)$$

and we find that $\epsilon_\alpha = -\sum_i v_{\bullet,i} p_i \exp(-\sum_k u_{i,k}\mathsf{act}(L-2,k;x_\alpha)) + \mathcal{O}(p_i^2)$.

Recalling that we aim to ensure

$$(*) = \sum_\alpha \epsilon_\alpha (c - y_\alpha) \geq 0$$

we consider

$$\sum_\alpha \epsilon_\alpha (c - y_\alpha) = -\sum_\alpha (c - y_\alpha)(\sum_i v_{\bullet,i} p_i \exp(-\sum_k u_{i,k}\mathsf{act}(L-2,k;x_\alpha))) + \mathcal{O}(p_i^2)$$

$$= -\sum_i v_{\bullet,i} p_i \sum_\alpha (c - y_\alpha)\exp(-\sum_k u_{i,k}\mathsf{act}(L-2,k;x_\alpha)) + \mathcal{O}(p_i^2)$$

We are still able to choose the parameters $u_{i,k}$ for $i \neq 0$, the parameters from previous layers, and the $v_{\bullet,i}$ subject to $\sum_i v_{\bullet,i} = c$. If now

$$v_{\bullet,i} > 0 \text{ whenever } \sum_\alpha (c - y_\alpha) \exp(-\sum_k u_{i,k} \mathsf{act}(L-2,k;x_\alpha)) < 0, \text{ and}$$
$$v_{\bullet,i} < 0 \text{ whenever } \sum_\alpha (c - y_\alpha) \exp(-\sum_k u_{i,k} \mathsf{act}(L-2,k;x_\alpha)) > 0,$$

then the term $(*)$ is strictly positive, hence the overall loss is larger than the loss at $p_i = 0$ for sufficiently small p_i and in a neighborhood of $v_{\bullet,i}$. The only obstruction we have to get around is the case where we need all $v_{\bullet,i}$ of the opposite sign of c (in other words, $\sum_\alpha (c-y_\alpha) \exp(-\sum_k u_{i,k} \mathsf{act}(L-2,k;x_\alpha))$ has the same sign as c), conflicting with $\sum_i v_{\bullet,i} = c$. To avoid this case, we impose the mild condition that $\sum_\alpha (c-y_\alpha) \mathsf{act}(L-2,r;x_\alpha) \neq 0$ for some r, which can be arranged to hold for almost every data set by fixing all parameters of layers with index smaller than $L-2$. By Lemma A.2 below (with $d_\alpha = (c-y_\alpha)$ and $a^r_\alpha = \mathsf{act}(L-2,r;x_\alpha)$), we can find $u^>_k$ such that $\sum_\alpha (c-y_\alpha) \exp(-\sum_k u^>_k \mathsf{act}(L-2,k;x_\alpha)) > 0$ and $u^<_k$ such that $\sum_\alpha (c-y_\alpha) \exp(-\sum_k u^<_k \mathsf{act}(L-2,k;x_\alpha)) < 0$. We fix $u_{i,k}$ for $k \geq 0$ such that there is some i_1 with $[u_{i_1,k}]_k = [u^>_k]_k$ and some i_2 with $[u_{i_2,k}]_k = [u^<_k]_k$. This assures that we can choose the $v_{\bullet,i}$ of opposite sign to $\sum_\alpha (c-y_\alpha) \exp(-\sum_k u_{i,k} \mathsf{act}(L-2,k;x_\alpha))$ and such that $\sum_i v_{\bullet,i} = c$, leading to a local minimum at infinity.

The local minimum is suboptimal whenever a constant function is not the optimal network function for the given data set.

■

Lemma A.2 *Suppose that m is a positive integer, $m \geq 2$, and for $\alpha = 1,\ldots,N$ and $r = 1,\ldots,m$ we have numbers d_α, a^r_α in $\mathbb{R}$ such that*

$$\sum_{\alpha=1}^N d_\alpha = 0, \text{ and } \sum_\alpha d_\alpha a^r_\alpha \neq 0 \text{ for some } r.$$

Then there are $u^<_k, k = 1,2,...,m$ such that

$$\sum_\alpha d_\alpha \exp(-\sum_k u^<_k a^k_\alpha) < 0$$

and $u^>_k, k = 1,2,...,m$ such that

$$\sum_\alpha d_\alpha \exp(-\sum_k u^>_k a^k_\alpha) > 0.$$

Proof Consider the function

$$\phi(u_1,u_2,\ldots,u_m) := \sum_\alpha d_\alpha \exp(-\sum_k u_k a^k_\alpha).$$

We have

$$\phi(0,0,\ldots,0) = \sum_\alpha d_\alpha = 0.$$

Further

$$\frac{\partial \phi}{\partial u_{r\,|(0,0,\dots,0)}} = -\sum_{\alpha} d_\alpha a_\alpha^r.$$

By assumption, there is r such that the last term is nonzero. Hence, using coordinate r, we can choose $w = (0, 0, \dots, 0, w_r, 0, \dots, 0)$ such that $\phi(w)$ is positive and we can choose w such that $\phi(w)$ is negative. ■

A.3 Construction of Local Minima in Deep Networks

Proof of Proposition 15 The fact that property (i) suffices uses that $\frac{\partial \ell_\alpha}{\mathsf{n}(l+1,\bullet;x_\alpha)}$ reduces to $(f(x_\alpha) - y_\alpha)$. Then, considering a regression network as before, our assumption says that $v^*_{\bullet,r} \neq 0$, hence its reciprocal can be factored out of the sum in Equation 2. Denoting incoming weights into $\mathsf{n}(l, r; x)$ by $u_{r,i}$ as before, this leads to

$$\begin{aligned} D_i^{r,\bullet} &= \frac{1}{v^*_{\bullet,r}} \cdot \sum_\alpha (f(x_\alpha) - y_\alpha) \cdot v^*_{\bullet,r} \cdot \sigma'(\mathsf{n}(l, r; x_\alpha)) \cdot \mathsf{act}(l-1, i; x_\alpha) \\ &= \frac{1}{v^*_{\bullet,r}} \cdot \frac{\partial \ell}{\partial u_{r,i}} = 0 \end{aligned}$$

In the case of (ii),

$$\frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1, s; x_\alpha)} = \frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1, t; x_\alpha)}$$

for all s, t and we can factor out the reciprocal of $\sum_t v^*_{r,t} \neq 0$ in Equation 2 to obtain for each i, s

$$\begin{aligned} D_i^{r,s} &:= \sum_\alpha \frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1, s; x_\alpha)} \cdot \sigma'(\mathsf{n}(l, r; x_\alpha)) \cdot \mathsf{act}(l-1, i; x_\alpha) \\ &= \frac{1}{\left(\sum_t v^*_{r,t}\right)} \sum_\alpha \sum_t v^*_{r,t} \cdot \frac{\ell_\alpha}{\partial \mathsf{n}(l+1, t; x_\alpha)} \cdot \sigma'(\mathsf{n}(l, r; x_\alpha)) \cdot \mathsf{act}(l-1, i; x_\alpha) \\ &= \frac{1}{\left(\sum_t v^*_{r,t}\right)} \cdot \frac{\partial \ell}{\partial u_{r,i}} = 0 \end{aligned}$$

Part (iii) is evident since in this case clearly every summand in Equation 2 is zero. ■

A.4 Proofs for the Non-increasing Path to a Global Minimum

This section contains the proofs to all statements of Section 6, that show how in extremely wide neural networks with a non-increasing sequence of dimensions of hidden layers following the extremely wide layer, a path to the global minimum that is non-increasing in loss may be found from almost everywhere in the parameter space. This leads to the proof of Theorem 5 at the end of the section.

Proof of Lemma 17 We write $\mathbf{n}_\Gamma^{l^*+1}(t) = \mathbf{n}^{l^*+1} + \tilde{\mathbf{n}}_\Gamma(t)$ with $\tilde{\mathbf{n}}_\Gamma(0) = 0$. We will find $\tilde{\mathbf{w}}_\Gamma(t)$ such that $\tilde{\mathbf{n}}_\Gamma(t) = \tilde{\mathbf{w}}_\Gamma(t) \cdot \mathbf{a}^{l^*}$ with $\tilde{\mathbf{w}}_\Gamma(0) = 0$. Then $\mathbf{w}_\Gamma^{l^*+1}(t) := \mathbf{w}^{l^*+1} + \tilde{\mathbf{w}}_\Gamma(t)$ does the job.

Since, by assumption, $\mathbf{a}^{l^*} = [\mathsf{act}(l^*, k; x_\alpha)]_{k,\alpha}$ has full rank, we can find an invertible submatrix $\bar{A} \in \mathbb{R}^{N\times N}$ of $\mathbf{a}^{l^*}$. Then we can define a continuous path ω in $\mathbb{R}^{n_{l^*+1}\times N}$ given by $\omega(t) := \tilde{\mathbf{n}}_\Gamma(t) \cdot \bar{A}^{-1}$, which satisfies $\omega(t) \cdot \bar{A} = \tilde{\mathbf{n}}_\Gamma(t)$ and $\omega(0) = 0$. Extending $\omega(t)$ to a path $\tilde{\mathbf{w}}_\Gamma(t)$ in $\mathbb{R}^{n_{l^*+1}\times n_{l^*}}$ by zero columns at positions corresponding to rows of $\mathbf{a}^{l^*}$ missing in $\bar{A}$ gives $\tilde{\mathbf{w}}_\Gamma(t) \cdot \mathbf{a}^{l^*} = \tilde{\mathbf{n}}_\Gamma(t)$ and $\tilde{\mathbf{w}}_\Gamma(0) = 0$ as desired.

■

Proof of Lemma 18 Since $\sigma : \mathbb{R}^{n\times N} \to \mathrm{Im}(\sigma)^{n\times N}$ is invertible with a continuous inverse, take

$$\mathbf{n}_\Gamma(t) = \sigma^{-1}(\mathbf{a}_\Gamma(t)).$$

■

Proof of Lemma 19 Since the matrix $\mathbf{w}^{l+1}$ has full rank, it contains an invertible submatrix $W \in \mathbb{R}^{n_{l+1}\times n_{l+1}}$. Since we can permute indices of neurons in each layer, we can assume without loss of generality that this submatrix consists of the first n_{l+1} columns of $\mathbf{w}^{l+1}$.

For some suitable paths $\lambda(t) \in \mathbb{R}_{>0}$ and $\delta(t) \in \mathbb{R}^{n_{l+1}}$, which will be chosen later, we define

$$\begin{aligned}
\tilde{\mathbf{n}}_\Gamma(t) &:= \mathbf{n}_\Gamma^{l+1}(t) - \mathbf{n}_\Gamma^{l+1}(0)\\
\mathbf{w}_\Gamma^{l+1}(t) &:= \lambda(t) \cdot \mathbf{w}^{l+1}\\
\mathbf{w}_{\Gamma,0}^{l+1}(t) &:= \mathbf{w}_0^{l+1} - \delta(t)\\
\mathbf{a}_\Gamma^{l}(t) &:= \frac{1}{\lambda(t)} \left(\mathbf{a}^l + \begin{bmatrix} W^{-1}\tilde{\mathbf{n}}_\Gamma(t) \\ 0_{n_l - n_{l+1}} \end{bmatrix} + \begin{bmatrix} W^{-1}\delta(t) \\ 0_{n_l - n_{l+1}} \end{bmatrix} \right).
\end{aligned}$$

We then have $\tilde{\mathbf{n}}_\Gamma(0) = 0$, and we choose $\lambda(0) = 1$ and $\delta(0) = 0$. This implies that at $t = 0$ we have $\mathbf{a}_\Gamma^l(0) = \mathbf{a}^l \in \mathrm{Im}(\sigma)^{n_l\times N}$. Further

$$\begin{aligned}
\mathbf{w}_\Gamma^{l+1}(t) \cdot \mathbf{a}_\Gamma^l(t) + \mathbf{w}_{\Gamma,0}^{l+1}(t) &= \lambda(t) \cdot \mathbf{w}^{l+1} \cdot \mathbf{a}_\Gamma^l(t) + \mathbf{w}_0^{l+1} - \delta(t)\\
&= \mathbf{w}^{l+1}\mathbf{a}^l + \mathbf{w}_0^{l+1} + \underbrace{\mathbf{w}^{l+1}}_{=[W\ *]} \left(\begin{bmatrix} W^{-1}\tilde{\mathbf{n}}_\Gamma(t) \\ 0_{n_l - n_{l+1}} \end{bmatrix} + \begin{bmatrix} W^{-1}\delta(t) \\ 0_{n_l - n_{l+1}} \end{bmatrix} \right) - \delta(t)\\
&= \mathbf{n}_\Gamma^{l+1}(0) + \tilde{\mathbf{n}}_\Gamma(t) + \delta(t) - \delta(t) = \mathbf{n}_\Gamma^{l+1}(t)
\end{aligned}$$

as desired. Note that with $\delta(t) = 0$ and $\lambda(t) = 1$ for all t, we would obtain suitable paths with $\mathbf{a}_\Gamma^l(t) \in \mathbb{R}^{n_l\times N}$, but due to the activations in the previous layer we must require that $\mathbf{a}_\Gamma^l(t) \in \mathrm{Im}(\sigma)^{n_l\times N}$. Here, we use the full freedom of choosing $\delta(t)$ and $\lambda(t)$ to ensure this. In the case that $0 \in (c, d) = \mathrm{Im}(\sigma)$ it suffices to fix $\delta(t) = 0$ and to always choose sufficiently large $\lambda(t) > 0$ such that $\mathbf{a}_\Gamma^l(t) \in (c, d)^{n_l\times N}$. In the case that 0 lies on the boundary of the interval $[c, d]$, we also need to to choose $\delta(t)$ to guarantee the correct sign in each component of $\mathbf{a}_{\Gamma,k}^l(t)$, i.e., if $c = 0$ then choose $\delta(t)$ such that each entry of $W^{-1}\delta(t)$ is sufficiently large

to guarantee that $\left(\mathbf{a}^l + \begin{bmatrix} W^{-1}\tilde{\mathbf{n}}_\Gamma(t) \\ 0_{n_l - n_{l+1}} \end{bmatrix} + \begin{bmatrix} W^{-1}\delta(t) \\ 0_{n_l - n_{l+1}} \end{bmatrix}\right) \in \mathbb{R}^{n_l \times N}_{>0}$. ■

Proof of Lemma 20 As outlined before the statement of the lemma, the proof only requires a composition of previous lemmas. We first show by induction over l that, for each $l^* < l \leq L-1$, every continuous path $\mathbf{a}^l_\Gamma(t) \in C([0,1], \mathbb{R}^{n_l \times N})$ can be realized by a continuous change of parameters from previous layers. That is, for all $l^* < l \leq L-1$ and all continuous paths $\mathbf{a}^l_\Gamma(t) \in C([0,1], \mathbb{R}^{n_l \times N})$ starting at the activations values in layer l for parameters $\mathbf{w}^l$, i.e., $\mathbf{a}^l_\Gamma(0) = \mathbf{a}^l(\mathbf{w})$, there is a continuous path of parameters $\mathbf{w}_\Gamma(t)$ with $\mathbf{w}_\Gamma(0) = \mathbf{w}$ such that the activation values $\mathbf{a}$ as a function of $\mathbf{w}_\Gamma(t)$ satisfy $\mathbf{a}(\mathbf{w}_\Gamma(t)) = \mathbf{a}_\Gamma(t)$.

The base case for the induction, $l = l^* + 1$, holds true by combining Lemma 17 and 18. The induction step is further shown by combining Lemma 18 and 19.

This guarantees all necessary assumptions to also apply Lemma 19 to the last layer, showing that any path $f_\Gamma(t)$ can be realized at the output. ■

Proof of Theorem 5 Let $\mathbf{w}$ be a given set of parameters for the neural network and $\epsilon > 0$. Applying Lemma 17 we find $\mathbf{w}'$ such that (i) $||\mathbf{w} - \mathbf{w}'|| < \epsilon$, the activation vectors $\mathbf{a}^{l^*}_k$ of the extremely wide layer l^* (containing more neurons than the number of training samples N) at parameters $\mathbf{w}'$ satisfy

$$\operatorname{span}_k \mathbf{a}^{l^*}_k = \mathbb{R}^N,$$

and (iii) the weight matrices $(\mathbf{w}')^l$ have full rank for all $l > l^* + 1$.

Part (ii) and (iii) together with the assumption on the architecture of the network guarantee the assumptions of Lemma 20 for $\mathbf{w}'$, so that for any continuous path $f_\Gamma : [0,1] \to \mathbb{R}^N$ with $f_\Gamma(0) = [f(\mathbf{w}'; x_\alpha)]_\alpha$ there is a continuous path $\mathbf{w}'_\Gamma(t)$ with $\mathbf{w}'_\Gamma(0) = \mathbf{w}'$ and $f_\Gamma(t) = [f(\mathbf{w}'_\Gamma(t); x_\alpha)]_\alpha$. So we only need to specify a desired path at the output, which we can then realize by a continuous change of parameters of the neural network.

With fixed $z_\alpha = f(\mathbf{w}'; x_\alpha)$, the prediction for the current weights, let $\mathbf{z} = [z_\alpha]_\alpha$ denote the vector of predictions, and let $\mathbf{y} = [y_\alpha]_\alpha$ denote the vector of all target values. An obvious path of decreasing loss at the output layer is then given by $t \in [0,1] \mapsto \mathbf{z} + t \cdot (\mathbf{y} - \mathbf{z})$, inducing the loss $\mathcal{L} = ||\mathbf{z} + t \cdot (\mathbf{y} - \mathbf{z}) - \mathbf{y}||_2^2 = (1-t)||\mathbf{y} - \mathbf{z}||_2^2$. ■

Appendix B. Notation

B.1 General Notation

$[x_i]_i$	$\mathbb{R}^n$	column vector with entries $x_i \in \mathbb{R}$
$[x_{i,j}]_{i,j}$	$\in \mathbb{R}^{n_1 \times n_2}$	matrix with entry $x_{i,j}$ at position (i,j)
Im(f)	$\subseteq \mathbb{R}$	image of a function f
$C(X,Y)$		continuous function from X to Y
N	$\in \mathbb{N}$	number of data samples in training set
x_α	$\in \mathbb{R}^{n_0}$	training sample input
y_α	$\in \mathbb{R}$	target output for sample x_α
$\mathcal{A}$	$\in \mathrm{C}(\mathbb{R})$	class of real-analytic, strictly monotonically increasing, bounded (activation) functions such that the closure of the image contains zero
σ	$\in \mathcal{A} \subset C(\mathbb{R},\mathbb{R})$	a nonlinear activation function in class $\mathcal{A}$
f	$\in C(\mathbb{R}^{n_0},\mathbb{R})$	neural network function
l	$1 \leq l \leq L$	index of a layer
L	$\in \mathbb{N}$	number of layers excluding the input layer
l=0		input layer
$l = L$		output layer
n_l	$\in \mathbb{N}$	number of neurons in layer l
M	$= \sum_{l=1}^{L} n_l \cdot (n_{l-1}+1)$	number of all network parameters
k	$1 \leq k \leq n_l$	index of a neuron in layer l
$\mathbf{w}^l$	$\in \mathbb{R}^{n_l \times n_{l-1}}$	weight matrix of the l-th layer
$\mathbf{w}_0^l$	$\in \mathbb{R}^{n_l}$	bias terms of the l-th layer
$\mathbf{w}$	$\in \mathbb{R}^M$	collection of all $\mathbf{w}^l$
$w_{i,j}^l$	$\in \mathbb{R}$	the weight from neuron j of layer $l-1$ to neuron j of layer l
$w_{\bullet,j}^L$	$\in \mathbb{R}$	the weight from neuron j of layer $L-1$ to the output
$\mathbf{w}_\Gamma$	$\in \mathrm{C}([0,1],\mathbb{R}^M)$	a path in parameter space
$\mathcal{L}, \ell$	$\in \mathbb{R}_+$	squared loss over training samples
ℓ_α	$\in \mathbb{R}_+$	the squared loss for data sample x_α
$\mathsf{n}(l,k;x)$	$\in \mathbb{R}$	value at neuron k in layer l before activation for input pattern x
$\mathsf{n}(l;x)$	$\in \mathbb{R}^{n_l}$	neuron pattern at layer l before activation for input pattern x
$\mathsf{act}(l,k;x)$	$\in \mathrm{Im}(\sigma)$	activation pattern at neuron k in layer l for input x
$\mathsf{act}(l;x)$	$\in \mathrm{Im}(\sigma)^{n_l}$	neuron pattern at layer l for input x

B.2 Notation Section V

In Section V, where we fix a layer l, we additionally use the following notation.

$[u_{p,i}]_{p,i}$	$\in \mathbb{R}^{n_l \times n_{l-1}}$	weights of the given layer l.
$[v_{s,q}]_{s,q}$	$\in \mathbb{R}^{n_l \times n_{l+1}}$	weights of the layer $l+1$.
r	$\in \{1, 2, \dots, n_l\}$	the index of the neuron of layer l that we use for the addition of one additional neuron
M	$\in \mathbb{N}$	$= \sum_{t=1}^{L} n_t \cdot (n_{t-1} + 1)$, the number of weights in the smaller neural network
$\bar{\mathbf{w}}$	$\in \mathbb{R}^{M - n_{l-1} - n_{l+1} - 1}$	all weights except $u_{r,i}$ and $v_{s,r}$ for all i and s
γ_λ^r	$\in \mathrm{C}(\mathbb{R}^M, \mathbb{R}^{M'})$	the map defined in Section 5 to add a neuron in layer l using the neuron with index r in layer l
$M' =$	$M + n_{l-1} + n_{l+1} + 1$	
$B_{i,j}^r$	$\in \mathbb{R}$	$= \sum_\alpha \sum_k \frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1,k;x_\alpha)} \cdot v_{k,r} \cdot \sigma''(\mathsf{n}(l,r;x_\alpha)) \cdot \mathsf{act}(l-1,i;x_\alpha) \cdot \mathsf{act}(l-1,j;x_\alpha)$
$D_i^{r,s}$	$\in \mathbb{R}$	$= \sum_\alpha \frac{\partial \ell_\alpha}{\partial \mathsf{n}(l+1,s;x_\alpha)} \cdot \sigma'(\mathsf{n}(l,r;x_\alpha)) \cdot \mathsf{act}(l-1,i;x_\alpha)$
$B = [B_{i,j}^r]_{i,j}$	$\in \mathbb{R}^{n_{l-1} \times n_{l-1}}$	matrix needs to be pos. or neg. def. for local min.
$D = [D_i^{r,s}]_{i,s}$	$\in \mathbb{R}^{n_{l-1} \times n_{l+1}}$	matrix needs to be 0 for local min.

B.3 Notation Section VI

In Section VI, we additionally use the following notation.

$\mathbf{a}_k^l$	$\in \mathrm{Im}(\sigma)^N$	activation vector at neuron k in layer l given by $\mathbf{a}_k^l = [\mathsf{act}(l,k;x_\alpha)]_\alpha$
$\mathbf{a}^l$	$\in \mathrm{Im}(\sigma)^{n_l \times N}$	matrix of activations in layer l given by $\mathbf{a}^l = [\mathbf{a}_k^l]_k$
$\mathbf{a}^l(\mathbf{w})$	$\in \mathrm{Im}(\sigma)^{n_l \times N}$	activation vector at layer l as a function of the parameters $\mathbf{w}$
$\mathbf{w}_\Gamma$	$\in \mathrm{C}([0,1], \mathbb{R}^M)$	a path in parameter space
$\mathbf{w}_\Gamma^l$	$\in \mathrm{C}([0,1], \mathbb{R}^{n_l \times n_{l-1}})$	a path in parameter space at layer l
$\mathbf{n}^l$	$\in \mathbb{R}^{n_l \times N}$	matrix of pre-activation values in layer l given by $\mathbf{n}^l = [\mathsf{n}(l,k;x_\alpha)]_{k,\alpha}$
$\mathbf{n}_\Gamma^l$	$\in \mathrm{C}([0,1], \mathbb{R}^{n_l \times N})$	a path of neuron values in layer l
$\mathbf{a}_\Gamma^l$	$\in \mathrm{C}([0,1], \mathbb{R}^{n_l \times N})$	a path of activation values in layer l
$\mathbf{a}_{\Gamma,j}^l$	$\in \mathrm{C}([0,1], \mathbb{R}^N)$	a path of activation vectors at neuron j in layer l
$f(\mathbf{w}; x_\alpha)$	$\in \mathbb{R}$	network as function of parameters $\mathbf{w}$ and sample x_α
f_Γ^l	$\in \mathrm{C}([0,1], \mathbb{R}^N)$	a path of outputs at all training samples

References

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10 (2):251–276, 1998.

Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially many local minima for single neurons. In *Advances in Neural Information Processing Systems 8, NIPS 1995*, pages 316–322, 1995.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is NP-complete. *Neural Networks*, 5(1):117–127, 1992.

Martin Lee Brady, Raghu Raghavan, and Joseph Slawny. Back propagation fails to separate where perceptrons succeed. *IEEE Transactions on Circuits and Systems*, 36(5):665–674, 2006.

Alan J. Bray and David S. Dean. The statistics of critical points of gaussian fields on large-dimensional spaces. *Physical Review Letters*, 98:150–201, 1989.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015*, 2015.

Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems 27, NIPS 2014*, pages 2933–2941, 2014.

Simon S. Du, Jason D Lee, Yuandong Tian, Aarti Singh, and Barnabas Poczos. Gradient descent learns one-hidden-layer CNN: Dont be afraid of spurious local minima. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 1338–1347, 2018.

C. Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, 2017.

Kenji Fukumizu and Shun-ichi Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 13(3):317–327, 2000.

Marco Gori and Alberto Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.

Benjamin D. Haeffele and Rene Vidal. Global optimality in neural network training. *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 17*, pages 4390–4398, 2017.

Leonard G. C. Hamey. XOR has no local minima: A case study in neural network error surface analysis. *Neural Networks*, 11(4):669–681, 1998.

Fengxiang He, Bohan Wang, and Dacheng Tao. Piecewise linear activations substantially shape the loss surfaces of neural networks. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020*, 2020.

Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems 29, NIPS 2016*, pages 586–594, 2016.

Thomas Laurent and James Brecht. Deep linear networks with arbitrary loss: All local minima are global. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 2908–2913, 2018a.

Thomas Laurent and James Brecht. The multilinear structure of relu networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 2914–2922, 2018b.

Shiyu Liang, Ruoyu Sun, Jason D Lee, and Rayadurgam Srikant. Adding one neuron can eliminate all bad local minima. In *Advances in Neural Information Processing Systems 31, NIPS 2018*, pages 4355–4365, 2018a.

Shiyu Liang, Ruoyu Sun, Yixuan Li, and Rayadurgam Srikant. Understanding the loss surface of neural networks for binary classification. In *In International Conference on Machine Learning 35, ICML 2018*, pages 2840–2849, 2018b.

Qianli Liao and Tomaso Poggio. Theory of deep learning II: Landscape of the empirical risk in deep learning. *arXiv e-prints*, arXiv:1703.09833, 2017.

Haihao Lu and Kenji Kawaguchi. Depth creates no bad local minima. *arXiv e-prints*, arXiv:1702.08580, 2017.

Eiji Mizutani and Stuart Dreyfus. An analysis on negative curvature induced by singularity in multi-layer neural-network learning. In *Advances in Neural Information Processing Systems 23, NIPS 2010*, pages 1669–1677, 2010.

Quynh Nguyen. On connected sublevel sets in deep learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, pages 4790–4799, 2019.

Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pages 2603–2612, 2017.

Quynh Nguyen and Matthias Hein. Optimization landscape and expressivity of deep cnn. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 3727–3736, 2018.

Quynh Nguyen, Mahesh C. Mukkamala, and Matthias Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019*, 2019.

Tohru Nitta. Resolution of singularities introduced by hierarchical structure in deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2282–2293, 2017.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

Timothy Poston, Chung-Nim Lee, YoungJu Choie, and Yonghoon Kwon. Local minima and back propagation. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 2, pages 173–176, 1991.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362, 1986.

Itay Safran and Ohad Shamir. On the quality of the initial basin in overspecified neural networks. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, pages 774–782, 2016.

Levent Sagun, V. Ugur Güney, Gérard Ben Arous, and Yann LeCun. Exploration on high dimensional landscapes. *arXiv e-prints*, arXiv:1412.6615, 2015.

Cristian Sminchisescu and Bill Triggs. Building roadmaps of minima and transitions in visual models. *International Journal of Computer Vision*, 61(1):81–101, 2005.

Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv e-prints*, arXiv:1605.08361, 2016.

Daniel Soudry and Elad Hoffer. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv e-prints*, arXiv:1702.05777, 2017.

Ida G. Sprinkhuizen-Kuyper and Egbert J. W. Boers. A local minimum for the 2-3-1 XOR network. *IEEE Transactions on Neural Networks*, 10(4):968–971, 1999.

Grzegorz Swirszcz, Wojciech M. Czarnecki, and Razvan Pascanu. Local minima in training of neural networks. *arXiv e-prints*, arXiv:1611.06310, 2016.

Luca Venturi, Afonso S. Bandeira, and Joan Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. *Journal of Machine Learning Research*, 20(133): 1–34, 2019.

Rene Vidal, Joan Bruna, Raja Giryes, and Stefan. Mathematics of deep learning. *arXiv e-prints*, arXiv:1712.04741, 2017.

Xu Gang Wang, Zheng Tang, Hiroki Tamura, Masahiro Ishii, and Wei Dong Sun. An improved backpropagation algorithm to avoid the local minima problem. *Neurocomputing*, 56:455–460, 2004.

Haikun Wei, Jun Zhang, Florent Cousseau, Tomoko Ozeki, and Shun-ichi Amari. Dynamics of learning near singularities in layered networks. *Neural Computation*, 20(3):813–843, 2008.

Lodewyk F.A. Wessels and Etienne Barnard. Avoiding false local minima by proper initialization of connections. *IEEE Transactions on Neural Networks*, 3(6):899–905, 1992.

Lodewyk F.A. Wessels, Etienne Barnard, and Eugene van Rooyen. The physical correlates of local minima. In *International Neural Network Conference*, pages 985–985, 1990.

Bo Xie, Yingyu Liang, and Le Song. Diversity leads to generalization in neural networks. *arXiv e-prints*, arXiv:1611.03131, 2016.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, 2018.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small nonlinearities in activation functions create bad local minima in neural networks. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv e-prints*, arXiv:1611.03530, 2017.

Yi Zhou and Yingbin Liang. Critical points of neural networks: Analytical forms and landscape properties. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, 2018.