ETH zürich

The curse of dimensionality and gradient-based training of neural networks: shrinking the gap between theory and applications

Doctoral thesis of

Florian Rossmannek

Diss. ETH No. 29083

DISS. ETH NO. 29083

The curse of dimensionality and gradient-based training of neural networks: shrinking the gap between theory and applications

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

Presented by

FLORIAN ROSSMANNEK

MSc ETH Mathematics, ETH Zurich
born on 06.01.1998

Accepted on the recommendation of

Prof. Dr. Patrick Cheridito
Prof. Dr. Arnulf Jentzen

2023

In memory of Will Merry († 2022)

Acknowledgments

The journey that led to this thesis would not have been possible without my advisors Prof. Dr. Arnulf Jentzen and Prof. Dr. Patrick Cheridito. I am deeply grateful to have had the opportunity to work together with them, for their guidance throughout my doctoral studies, and for providing me with the tools to evolve into a proper researcher. I am impressed with how Arnulf stayed on top of everything even when being on the other side of the globe. I would like to extend my thanks to Prof. Dr. Dylan Possamaï for acting as a chair of my examination committee. I also acknowledge the support by Swiss National Science Foundation Research Grant 175699.

Special thanks go to Adrian Riekert for a fruitful collaboration and to Philippe von Wurstemberger for helpful comments and suggestions. I am grateful to Philipp Zimmermann and Robert Crowell for providing me with food for thought for my research. They somehow managed to help me get unstuck more often than I got stuck to begin with. I am also thankful to Martin Štefánik and Philipp Zimmermann for their proofreading.

I would like to thank Robert, Laurin, and the board members of VMM, particularly Tim, for accompanying me in my involvement in the department, which had been a very rewarding experience alongside my studies. I thank all of Group 3 at the department for the community that reached beyond the walls of the university building and into the pubs and restaurants of Zurich. There are a few names I would like to mention particularly: Martin, Nikolay, Robert, Philipp, Matteo, and most of all Moritz. I am indebted to Philipp for inspiring and motivating me throughout my postgraduate job hunt. I also thank Alessio for many great lunch and coffee breaks.

There are no words to properly thank my parents for their constant love, support, and encouragement. My biggest thanks go to my brother Max for being both the best brother and the most terrific flatmate.

Contents

Preface

This is a cumulative thesis based on the published articles [15, 16, 18] and the preprint [17]. Specifically, Chapter 2 is based on [16]; Chapter 3 is based on [15]; Chapter 4 is based on [18]; and Chapter 5 is based on [17]. The introductions of the original articles have been modified to account for a coherent presentation in this thesis.

For each of those four articles, I have made major contributions: I led the development of the new concepts, the results, the proofs, and the writing. In the course of my doctoral studies, I also made a contribution to the published article [14] and proved a theoretical result about efficient numerical Sobolev approximations of solutions to linear parabolic partial differential equations, the preprint of which is in its final stage but is unpublished as of the submission date of this thesis.

Summary

Neural networks have gained widespread attention due to their remarkable performance in various applications. Two aspects are particular striking: on the one hand, neural networks seem to enjoy superior approximation capacities than classical methods. On the other hand, neural networks are trained successfully with gradient-based algorithms despite the training task being a highly nonconvex optimization problem. This thesis advances the theory behind these two phenomena.

On the aspect of approximation, we develop a framework for showing that neural networks can break the so-called curse of dimensionality in different high-dimensional approximation problems, meaning that the complexity of the neural networks involved scales at most polynomially in the dimension. Our approach is based on the notion of a catalog network, which is a generalization of a feed-forward neural network in which the nonlinear activation functions can vary from layer to layer as long as they are chosen from a predefined catalog of functions. As such, catalog networks constitute a rich family of continuous functions. We show that, under appropriate conditions on the catalog, these catalog networks can efficiently be approximated with rectified linear unit (ReLU)-type networks and provide precise estimates of the number of parameters needed for a given approximation accuracy. As special cases of the general results, we obtain different classes of functions that can be approximated with ReLU networks without the curse of dimensionality.

On the aspect of optimization, we investigate the interplay between neural networks and gradient-based training algorithms by studying the loss surface. On the one hand, we discover an obstruction to successful learning due to an unfortunate interplay between the architecture of the network and the initialization of the algorithm. More precisely, we demonstrate that stochastic gradient descent fails to converge for ReLU networks if their depth is much larger than their width and the number of random initializations does not increase to infinity fast enough. On the other hand, we establish positive results by conducting a landscape analysis and applying dynamical systems theory. These positive results deal with the landscape of the true loss of neural networks with one hidden layer and ReLU, leaky ReLU, or quadratic activation. In all three cases, we provide a complete classification of the critical points in the case where the target function is affine and one-dimensional. Next, we prove a new variant of a dynamical systems result, a center-stable manifold theorem, in which we relax some of the regularity requirements usually imposed. We verify that ReLU networks with one hidden layer fit into the new framework. Building on our classification of critical points, we deduce that gradient descent avoids most saddle points. We proceed to prove convergence to global minima if the initialization is sufficiently good, which is expressed by an explicit threshold on the limiting loss.

Zusammenfassung

Neuronale Netze haben aufgrund ihrer bemerkenswerten Leistungsfähigkeit in verschiedenen Anwendungen viel Aufmerksamkeit bekommen. Zwei Aspekte stechen dabei heraus: einerseits besitzen neuronale Netze bessere Approximationseigenschaften als klassische Methoden. Andererseits können neuronale Netze effizient mit Gradienten-basierten Algorithmen trainiert werden, obwohl das Training eines Netzes ein stark nichtkonvexes Problem ist. Diese Dissertation entwickelt die Theorie hinter diesen beiden Phänomenen weiter.

Im Rahmen der Approximationstheorie entwickeln wir ein Konzept, welches illustriert, dass neuronale Netze den sogenannten Fluch der Dimensionalität brechen können. Präziser bedeutet dies, dass die Komplexität eines neuronales Netzes höchstens polynomiell mit der Dimension wächst. Unsere Herangehensweise basiert auf dem Begriff eines Katalognetzes, welches eine Verallgemeinerung eines vorwärts gerichteten neuronalen Netzes ist, bei der die nicht linearen Aktivierungsfunktionen von Schicht zu Schicht unterschiedlich sein können solange sie aus einem Katalog von vorgegebenen Funktionen stammen. Somit bilden Katalognetze eine umfassende Familie von stetigen Funktionen. Wir zeigen, dass Katalognetze unter bestimmten Annahmen effizient mit "recitified linear unit" (ReLU)-artigen Netzen approximiert werden können, und wir liefern genaue Abschätzungen an die Anzahl Parameter, die für eine gegebene Approximationsgenauigkeit benötigt werden. Als Spezialfälle der allgemeinen Theorie erhalten wir verschiedene Klassen von Funktionen, die ohne den Fluch der Dimensionalität mit ReLU Netzen approximiert werden können.

Im Rahmen der Optimierungstheorie untersuchen wir das Zusammenspiel von neuronalen Netzen und Gradienten-basierten Trainingsalgorithmen, indem wir die Verlustoberfläche untersuchen. Einerseits charakterisieren wir eine Schwierigkeit für erfolgreiches Lernen, welche aus einem unvorteilhaften Zusammenspiel von der Architektur des Netzes und der Initialisierung des Algorithmus herrührt. Genauer gesagt zeigen wir, dass das stochastische Gradientenverfahren zu konvergieren fehlschlägt sobald die Tiefe von ReLU Netzen viel größer ist als ihre Breite und die Anzahl von zufälligen Initialisierungen nicht genügend schnell gegen unendlich wächst. Andererseits leiten wir positive Resultate her, indem wir eine Landschaftsanalyse durchführen und einen Satz aus der Theorie dynamischer Systeme verwenden. Diese positiven Resultate beschäftigen sich mit der Landschaft der tatsächlichen Verlustfunktion neuronaler Netze mit einer verborgenen Schicht und ReLU, "leaky" ReLU oder quadratischer Aktivierungsfunktion. In allen drei Fällen erstellen wir eine vollständige Klassifizierung der kritischen Punkte für eine affine eindimensionale Zielfunktion. Anschließend beweisen wir eine neue Variante eines Resultats über dynamische Systeme, einen Satz über zentral-stabile Mannigfaltigkeiten, in der wir einige der üblichen Regularitätsannahmen abschwächen. Wir stellen sicher, dass ReLU Netze mit einer verborgenen Schicht diesen Bedingungen genügen. Aufbauend auf unserer Klassifizierung kritischer Punkte schließen wir, dass das Gradientenverfahren die meisten Sattelpunkte vermeidet. Des Weiteren beweisen wir Konvergenz zum globalen Minimum unter der Voraussetzung, dass die Initialisierung hinreichend akkurat ist. Dies wird durch eine explizite Schranke an die Verlustfunktion quantifiziert.

CHAPTER 1

Introduction

The two faces of neural network theory

Neural networks have received a lot of attention in the past decade. Originally studied in the past century, they had still been dominated by classical or other machine learning methods, causing interest in them to temporarily stagnate; [112]. The emergence of increased computing power announced the dawn of a new era of interest in neural networks. Motivated by their widespread success in various applications, research on their mathematical theory resurged; [47, 81, 112]. Two aspects observed in applications are particularly striking: on the one hand, neural networks seem to enjoy good approximation capacities. More precisely, simulations suggest that they are able to break the so-called curse of dimensionality, meaning that the complexity of a neural network needed to solve a certain approximation task on a high-dimensional space scales at most polynomially in the dimension; [35,47,81]. On the other hand, training a neural network poses a highly nonconvex optimization problem; [104, 110]. The algorithms used for this optimization task are typically variants of gradient descent – an algorithm only known to work on convex problems. It is a long-standing open problem why these algorithms perform well on the nonconvex task of training a neural network; [28,88,115]. Understanding the success of neural networks entails both of these topics – approximation and optimization. This thesis explores theoretical results advancing our understanding of the subject matter in both topics. The results will shed light on why neural networks are successful, but also reveal some limitations of the theory.

A *standard, fully connected, feedforward neural network*, which we simply call a *neural network* or a *network*, encodes a succession of affine maps alternating with a fixed nonlinearity, called the *activation (function)*. An activation function we will encounter frequently is the *rectified linear unit (ReLU)* $\rho(x) = \max\{x, 0\}$. The *architecture* of a neural network refers to the number of affine maps (number of *layers*), which is called the *depth* of the network, and the dimension of the domain of each of these affine maps (the number of *neurons* in each layer). The first and the last layer are referred to as the *input* and the *output layer*, respectively; the layers in between are called *hidden layers*. A network of depth one (no hidden layer) is an affine function and, as such, not particularly interesting in itself. A *shallow* network is a network of depth two. The antonymous adjective *deep* is not used consistently in the literature. While practitioners tend to call a network deep if the number of layers is heuristically large (what that means is subjective to the respective current state of the art), theorists tend to call a network deep if it is not shallow, that is it has two or more hidden layers.

Approximation

Since the late 80's and early 90's, neural networks (of any fixed depth) have been known to be universal approximators, meaning that any function from a reasonable class of functions can be approximated arbitrarily well by a neural network; [24, 42, 56–59, 86, 93]. Typically, this reasonable class of functions is taken to be the class of continuous functions or the class of L^p-functions. The approximation is then measured in the supremum norm on compact sets or the L^p-norm, respectively. These early approximation results did not specify a rate, that is how the required number of neurons depends on the desired approximation accuracy. Later, such rates have been established, but it was found that neural networks suffer from the curse of dimensionality. Even more so, on a general approximation task, neural networks do not perform better than polynomial regression; [105, 106, 129]. This is in contrast to the performance observed in empirical studies; [35, 47, 81]. Noteworthy is that the negative results are not constructive. While we know that there are functions that cannot be approximated efficiently, we do not know how they look. This leaves hope that any approximation task encountered in real-world applications is not described by such a function. In particular, it makes sense to restrict the attention, for example, to subsets of the set of continuous functions, trying to exclude the above nonconstructive counterexamples. This is the strategy we adopt in Chapter 2. Therein, we introduce the concept of a catalog network. A catalog network is a generalization of a neural network. Instead of having a fixed nonlinearity that acts component-wise between any two affine layers, we allow the nonlinearity to be a stacking of functions drawn from a prespecified catalog of functions for each layer. The idea is that the catalog contains "simple" functions and a catalog network can represent highly complicated functions. We derive rates of approximation for neural networks when the target function is given by such a catalog network. Naturally, this rate is affected by the chosen catalog. The strength of the approximation results in Chapter 2 is that the complicated analysis of a catalog network reduces to the simpler analysis of the functions in the catalog. We will demonstrate this in examples, showing that catalog networks provide a very general framework for constructing classes of functions, in the approximation of which neural networks break the curse of dimensionality. Due to the constructive nature of catalog networks, we can cook up these examples explicitly. The key property of neural networks for this to work is their compositional structure because it enables us to concatenate and parallelize several networks into one; [49]. To improve the approximation rates, we develop a new way to parallelize networks, which is more efficient than the way it was previously done in the literature.

In the following, we contrast our approach with existing methods. [6, 71] have proved an $\mathcal{O}(n^{-1/2})$-rate for approximating functions in the L^2-norm with shallow sigmoidal networks with n neurons. In particular, this breaks the curse of dimensionality, but it only applies to a special class of functions. Since then, their results have been applied and generalized in different directions, always yielding rates of the same nature, but always applicable only to similarly restricted classes of functions. For example, in [31, 45] these results have been extended to the L^p-norm for $1 \leq p < \infty$ and $p = \infty$, respectively, and in [80] the approximation rate has been improved to a geometric rate for single functions. However, the basis δ of the geometric rate $\mathcal{O}((1-\delta)^n)$ is usually not known. It could be so small that the geometric rate does not give useful bounds for typical sizes of n. For further generalizations, see, e.g., [5, 7, 51, 72, 73, 75, 77–79]. All of them use shallow networks. However, deep networks have shown better performance in a number of applications; [47, 81]. This has also been

supported by theoretical evidence; for instance, in [39], an example of a simple continuous function on $\mathbb{R}^d$ has been given that is expressible as a small network with two hidden layers but cannot be approximated with a shallow network to a given constant accuracy unless its width is exponential in the dimension. Similarly, it has been shown in [109] that indicator functions of d-dimensional balls can be approximated much more efficiently with two hidden layers than with one. Related results for functions on the product of two d-dimensional spheres have been provided by [25].

[50, 92, 106] have constructed special activation functions which, in principle, allow to approximate every continuous function $f\colon [0,1]^d \to \mathbb{R}$ to any desired precision when used in a two-hidden-layers network with as few as d neurons in the first and $2d+2$ neurons in the second hidden layer. Theoretically, this breaks the curse of dimensionality quite spectacularly. However, it can be shown that the approximation result only holds if the size of the network weights is allowed to grow faster than polynomially in the inverse of the approximation error; [11, 105].

Further studies of the approximation capacity of neural networks with standard activation functions include [89, 93, 105, 127, 129]. Their approach is based on approximating functions with polynomials and then approximating these polynomials with neural networks. Polynomials can approximate smooth functions reasonably well, and neural networks are known to be able to approximate monomials efficiently. However, since the number of monomials needed to generate all polynomials in d variables of order k is $\binom{k+d}{d}$, the intermediate step from monomials to polynomials introduces the curse of dimensionality. It has been shown in [129] that this cannot be side-stepped. For instance, it is provably impossible to approximate the unit ball in the Sobolev space of any regularity with ReLU networks without the curse of dimensionality. To break the curse of dimensionality with ReLU networks, one has to concentrate on special classes of functions. [6, 71] and their extensions offer one such class. Coming from a different angle, [94] has obtained the same rate for periodic functions with an absolutely convergent Fourier series. In [113], the approximability of "separately holomorphic" maps via Taylor expansions and applications to parametric partial differential equations have been studied. The approach of [113] is again based on the intermediate approximation of polynomials, but the holomorphy ensures that the approximating polynomials contain only few monomials. [48, 60, 68] have proved that solutions of various partial differential equations admit neural network approximations without the curse of dimensionality. Their arguments use the hierarchical structure of neural networks, which has more extensively been exploited in [11, 38, 82]. These papers are similar in spirit to Chapter 2 in this thesis since they also start from a "basis" of functions, which they approximate with neural networks, and then use this basis to build more complex functions. However, [11, 38] do not study approximation rates in terms of the dimension. On the other hand, in [82] the curse of dimensionality is broken, but the "basis" in [82] consists of the functions considered in [6]. In Chapter 2, we consider more explicit classes of functions and provide bounds on the number of parameters needed to approximate d-dimensional functions up to accuracy ε.

Optimization

When aiming to numerically solve an approximation task with neural networks, the available results on approximation capacities discussed above give an idea of what a reasonable choice for an architecture might be. The theory may state that there is a network with a certain architecture that solves the task. But we do not know a priori which network that is.

The unknowns of a network are the parameters that specify the affine maps. This poses an optimization problem: we need to find parameters for which the network is a good approximator. This optimization problem is highly nonconvex; [104, 110]. To date, there are no efficient algorithms that are guaranteed to solve nonconvex optimization problems. Instead, practitioners rely on variants of (stochastic) gradient descent. These are known to find the optima of convex problems, even with good rates; [12, 96, 114]. But in a nonconvex problem, gradient methods can get stuck at saddle points or local minima. This has not stopped practitioners from achieving remarkable results with stochastic gradient descent in the training of neural networks; [47, 81, 112]. Advances in the theory indicate that there is something specific about the interplay of stochastic gradient descent and neural networks that makes it work. But a full theoretical understanding of gradient-based methods in network models is still lacking. In this thesis, we take a closer look at this interplay between gradient-based methods and neural networks.

To obtain optimal approximation results, several hyper-parameters have to be fine-tuned. The first one we discussed above: the architecture of the network determines what type of functions can be approximated. To be able to efficiently approximate complicated functions, the network needs to be sufficiently wide and deep. Secondly, the goal is to approximate a target function, but the target function itself is unknown and we only have access to a finite amount of data points. Phrased in the language of machine learning: we intend to approximate the target with respect to the true risk, but the algorithm only has access to the empirical risk. The gap between the two goes to zero as the amount of training data increases to infinity. Thirdly, the gradient method attempts to minimize the empirical risk, and the chance of finding a good approximate minimum increases with the number of gradient steps. Finally, since a single gradient trajectory may not yield good results, it is common to run several of them with different random initializations. [8, 69] have shown that general networks converge if their size, the amount of training data, and the number of random initializations are increased to infinity in the correct way, albeit with an extremely slow speed of convergence. In general, one cannot hope to overcome the slow speed of converge; [116]. On the other hand, it has been shown that, for the training error, faster convergence can be guaranteed with certain probabilities if overparametrized networks are used, that is an exceedingly large number of neurons; see [2, 21, 34, 36, 120, 131] and the references therein. The initialization method is important for any type of network. But for ReLU networks it plays a special role due to the particular form of the ReLU activation function; [52, 54, 90, 117].

The main contribution of Chapter 3 is a demonstration that stochastic gradient descent fails to converge for ReLU networks if the number of random initializations does not increase fast enough compared to the size of the network. Our arguments are based on an analysis of regions in the parameter space related to "inactive" neurons (sometimes also coined "dead" neurons in the literature). In these regions, the neural network function is constant not only in its argument but also in the network parameter. Suppose θ denotes the vector containing the parameters of a network. Recall that these are the parameters of the affine functions. Let $\mathcal{A}_1^\theta$ be the affine function from the input to the first hidden layer. If θ contains only strictly negative parameters, then $\rho \circ \mathcal{A}_1^\theta \circ \rho(x)$ is constantly zero in x and in a neighborhood of θ. As a consequence, stochastic gradient descent will not be able to escape from a neighborhood of θ. The fact that random initialization can render parts of a ReLU network inactive has already been noticed in [90, 117]. While the focus of [90, 117] is on the design of alternative random initialization schemes to make the training more efficient, we give precise estimates on the probability that the whole network becomes inactive and deduce that stochastic

gradient descent fails to converge if the number of random initializations does not increase fast enough relative to the size of the architecture. In our proof, the depth grows much faster than the maximal width of all layers. This imbalance between the depth and the width has the effect that the training procedure does not converge.

The negative result about convergence we just discussed is based on a careful study of the interplay between the architecture of the network and the initialization of the algorithm. Another approach, which we pursue in Chapter 4, is a landscape analysis of the loss surface. This landscape analysis provides an indirect tool for studying the dynamics of gradient-based algorithms, as these dynamics are governed by the loss surface. One goal of landscape analysis is a better understanding of the occurrence and frequency of critical points of the loss function and obtaining information about their type, that is, whether they constitute extrema, local extrema, or saddle points. Using the hierarchical structure of networks, some partial results have been obtained; [41]. Though, the choice of the activation function in the network model can have a significant impact on the landscape. For instance, it is known that the loss surface of a linear network, that is a network with the identity function as activation, only has global minima and saddle points but no non-global local minima; [4, 74]. However, the picture becomes less clear if a nonlinearity is introduced; [110, 111, 125].

In the last decade, progress has been made in this more difficult nonlinear case. In [22], the loss surface has been studied by relating it to a model from statistical physics. This way, detailed results have been obtained about the frequency and quality of local minima. Although the findings of [22] are theoretically insightful, their theory is based on assumptions that are not met in practice; [23]. In [122], similar results have been obtained for shallow networks with less unrealistic assumptions. We refer to [28] for experimental findings, on which [22, 122] is based.

Besides the work studying the effects of overparametrization on gradient-based methods directly as mentioned further above, there have also been investigations of its impact on the loss landscape. For instance, it has been shown in [108] that taking larger networks increases the likelihood to start from a good initialization with a small loss or from which there exists a monotonically decreasing path to a global minimum. However, it is still not fully understood in which situations a gradient-based training algorithm follows such a path. If the quadratic activation function is used in a shallow network, then, in the overparametrized regime, only global minima and strict saddle points remain, but no non-global local minima; [33, 126]. Even for deeper architectures, all non-global local minima disappear with high probability for any activation function if the width of the last hidden layer is increased and, under some regularity assumptions on the activation, this continues to hold if any of the hidden layers is sufficiently wide and the proceeding layers have a pyramidal structure; [88, 97, 120, 121]. However, note that these results only apply in this level of generality if the loss is measured with respect to a finite set of data. In particular, these global minima are (potentially) prone to overfitting.

In contrast to the literature mentioned above, our results in Chapter 4 concern the landscape of the true loss instead of the empirical loss. The final goal in machine learning is to minimize not only the empirical loss, but the true loss, so it is of essence to understand its landscape. In Chapter 4, we consider shallow networks with (leaky) ReLU or quadratic activation. As an alternative to the popular theme of overparametrization, we do not impose assumptions on the network model that are not met in practice, but instead focus on special target functions. In [14], this strategy has been pursued with constant target functions. In Chapter 4, we expand the scope from constant to affine functions. This represents a first step

towards a better understanding of the true loss landscape corresponding to general target functions.

In this framework with affine target functions, we provide a complete classification of the critical points of the true loss. We do so by unfolding the combinatorics of the problem, governed by different types of hidden neurons appearing in a network. We find that ReLU networks admit non-global local minima regardless of the number of hidden neurons. At the same time, it turns out that these local minima are solely caused by the dead ReLU neurons we found in Chapter 3. In particular, for leaky ReLU networks, which are often used to avoid the problem of dead neurons, there are only saddle points and global minima. This provides further theoretical evidence that leaky ReLU can avoid issues with training that appear for ReLU networks; see also [52]. Interestingly, also for the quadratic activation, non-global local minima do not appear, which is in line with the observations in [33, 126] for the discretized loss but does not require overparametrization. In addition, for networks with quadratic activation, all saddle points have a constant realization function, whereas for (leaky) ReLU networks we show that there exist saddle points with a nonconstant realization.

These complete classifications in the proposed approach to consider special target functions shed new light on important aspects of gradient-based methods in the training of networks. Knowledge of the loss surface can be transformed into results about convergence of such methods as done in, e.g., [67]. The set of non-global local minima, being caused by dead ReLU neurons, consists of a single connected component in the parameter space. We had already discovered these non-global local minima in Chapter 3. But we had not known that these are in fact the only ones. Chapter 3 revealed issues with initializing in that set of local minima. For the dynamics after initialization, originally, local minima were assumed to pose the greater challenge still, but recent results suggest that saddle points are the main obstacle; [22, 28, 126]. An important ingredient in tackling saddle points is *strictness*, meaning that the Hessian of the loss function has a strictly negative eigenvalue at these saddle points. The strictness ensures that there is a direction along which the loss surface declines significantly. Under the strictness assumption, a stochastic version of gradient descent with suitable noise in each step has the ability to avoid saddle points because the noise ensures that we discover the declining direction; [43, 70, 102]. The noise even guarantees a polynomial speed in escaping these saddle points; [32].

In the case of vanilla gradient descent, there is no noise to rely on, and one needs more involved analytic methods. A useful tool in this context is the *stable manifold theorem*, which is a cornerstone of classical dynamical systems theory; [118]. It has recently been applied to prove that vanilla gradient descent with suitable random initialization avoids strict saddle points with probability one if the loss function is sufficiently regular; [84, 99]. We remark that the applicability of the stable manifold theorem goes beyond vanilla gradient descent; see [26, 83, 98] for its application to variants of gradient descent and other first-order methods.

Accumulation points of gradient descent trajectories are critical points. Under typical assumptions like boundedness of trajectories and, e.g., validity of Łojasiewicz-type inequalities, it is also known that trajectories converge to a critical point; see [29, 40] for the stochastic and [1, 85] for the non-stochastic version. It follows that, with probability one, these limit critical points are local minima or nonstrict saddle points. The strictness assumption has been discussed in the literature and has been shown to hold in a variety of settings; e.g, in matrix recovery [10, 44, 123], phase retrieval [124], tensor decomposition [43], shallow quadratic networks, [33, 120], and deep linear networks [3, 74]. In particular, nonstrict saddle points appear to be less common than strict ones, and the above results shrink the gap to

proving convergence to local minima.

Whereas strictness has been discussed in abundance, less attention has been given to the regularity assumptions imposed on the loss function. In [26, 83, 84, 98], the loss function is taken to be twice continuously differentiable with a globally Lipschitz continuous gradient, and in [99] these conditions are assumed to hold on a forward-invariant convex open set. This level of regularity makes the classical dynamical systems theory directly applicable to gradient descent algorithms. However, in many modern machine learning applications, the loss function is neither twice continuously differentiable nor is its gradient uniformly Lipschitz continuous on suitable invariant sets. One of the main difficulties on the side of the dynamical systems theory is to provide a variant of the center-stable manifold theorem that relaxes these restrictions. To this end, we extend a result of [100] in Chapter 5, no longer requiring uniform Lipschitz continuity. A regularity requirement in this new center-stable manifold theorem persists. To deal with that, we tweak the framework to which we apply that theorem. More precisely, we will modify the gradient of the loss for ReLU networks so that it fulfills the assumptions and do it in a way that we can recover the dynamics of the original gradient. The final ingredient is strictness of saddle points. The strictness will be deduced from the classification of critical points from Chapter 4. Thanks to the classification being explicit, we can study the spectrum of the Hessian of the loss function as previously pursued in, e.g., [33, 103].

With Chapters 4 and 5, we have gained a good understanding of why gradient-based methods can successfully train shallow ReLU networks on affine target functions as long as the obstacle observed in Chapter 3 is taken care of. This also serves as a basis for understanding the case of more general architectures and target functions. Indeed, after the publication of the articles corresponding to this thesis, our results have been used as a starting point for further investigations. We will survey these in Chapter 6 to conclude this thesis.

CHAPTER 2

Efficient approximation of high-dimensional functions with neural networks

This chapter is an adaptation of the published article [16]. The proofs have been moved from appendices to the main body.

1. Introduction

In this chapter, we prove that different classes of high-dimensional functions admit a neural network approximation without the curse of dimensionality. To do that, we introduce the notion of a catalog network, which is a generalization of a standard neural network in which the nonlinear activation functions can vary from one layer to another as long as they are chosen from a given catalog of continuous functions. We first study the approximability of different catalogs with neural networks. Then, we show how the approximability of a catalog translates into the approximability of the corresponding catalog networks. An important building block of our proofs is a new way of parallelizing networks that saves parameters compared to the standard parallelization. As special cases of our general results, we obtain that different combinations of one-dimensional Lipschitz functions, sums, maxima and products as well as certain ridge functions and generalized Gaussian radial basis function networks admit a neural network approximation without the curse of dimensionality.

The remainder of this chapter is organized as follows. In Section 2, we first establish the notation. Then, we recall basic facts from [49, 68, 105] on concatenating and parallelizing neural networks before we introduce a new way of network parallelization. In Section 3, we introduce the concepts of an approximable catalog and a catalog network. Section 4 is devoted to different concrete examples of catalogs and a careful study of their approximability. In Sections 5 and 6, we derive bounds on the number of parameters needed to approximate a given catalog network to a desired accuracy with neural networks. Theorems 5.2 and 6.3 are the main results of this chapter. In Section 7, we derive different classes of high-dimensional functions that are approximable with ReLU networks without the curse of dimensionality.

2. Notation and preliminary results

A neural network encodes a succession of affine and nonlinear transformations. Let us denote $\mathbb{N} = \{1, 2, \dots\}$ and consider the set of neural network skeletons

$$\mathcal{N} = \bigcup_{D\in\mathbb{N}} \bigcup_{(l_0,\dots,l_D)\in\mathbb{N}^{D+1}} \prod_{k=1}^{D} (\mathbb{R}^{l_k\times l_{k-1}} \times \mathbb{R}^{l_k}).$$

We denote the depth of a neural network skeleton $\phi \in \mathcal{N}$ by $\mathcal{D}(\phi) = D$, the number of neurons in the kth layer by $l_k^\phi = l_k$, $k \in \{0, \dots, D\}$, and the number of network parameters by $\mathcal{P}(\phi) = \sum_{k=1}^{D} l_k(l_{k-1} + 1)$. Moreover, if $\phi \in \mathcal{N}$ is given by $\phi = [(V_1, b_1), \dots, (V_D, b_D)]$, we denote by $\mathcal{A}_k^\phi \in C(\mathbb{R}^{l_{k-1}}, \mathbb{R}^{l_k})$, $k \in \{1, \dots, D\}$, the affine function $x \mapsto V_k x + b_k$. Let $a\colon \mathbb{R} \to \mathbb{R}$ be a continuous activation function. As usual, we extend it, for every positive integer d, to a function from $\mathbb{R}^d$ to $\mathbb{R}^d$ mapping $(x_1, \dots, x_d)$ to $(a(x_1), \dots, a(x_d))$. Then the a-realization of $\phi \in \mathcal{N}$ is the function $\mathcal{R}_a^\phi \in C(\mathbb{R}^{l_0}, \mathbb{R}^{l_D})$ given by

$$\mathcal{R}_a^\phi = \mathcal{A}_D^\phi \circ a \circ \mathcal{A}_{D-1}^\phi \circ \cdots a \circ \mathcal{A}_1^\phi.$$

We recall that suitable $\phi_1, \phi_2 \in \mathcal{N}$ can be composed such that the a-realization of the resulting network equals the concatenation $\mathcal{R}_a^{\phi_2} \circ \mathcal{R}_a^{\phi_1}$. This is done by combining the output layer of ϕ_1 with the input layer of ϕ_2. More precisely, if $\phi_1 = [(V_1, b_1), \dots, (V_D, b_D)]$ and $\phi_2 = [(W_1, c_1), \dots, (W_E, c_E)]$ satisfy $l^{\phi_1}_{\mathcal{D}(\phi_1)} = l_0^{\phi_2}$, then the concatenation $\phi_2 \circ \phi_1 \in \mathcal{N}$ is given by

$$\phi_2 \circ \phi_1 = [(V_1, b_1), \dots, (V_{D-1}, b_{D-1}), (W_1V_D, W_1b_D + c_1), (W_2, c_2), \dots, (W_E, c_E)].$$

The following result is straight-forward from the definition. A formal proof can be found in [49].

Proposition 2.1. *The concatenation*

$$(\cdot) \circ (\cdot)\colon \{(\phi_1, \phi_2) \in \mathcal{N} \times \mathcal{N}\colon l^{\phi_1}_{\mathcal{D}(\phi_1)} = l_0^{\phi_2}\} \to \mathcal{N}$$

is associative and for all $\phi_1, \phi_2 \in \mathcal{N}$ with $l^{\phi_1}_{\mathcal{D}(\phi_1)} = l_0^{\phi_2}$ one has

(i) $\mathcal{R}_a^{\phi_2\circ\phi_1} = \mathcal{R}_a^{\phi_2} \circ \mathcal{R}_a^{\phi_1}$ *for all* $a \in C(\mathbb{R}, \mathbb{R})$,

(ii) $\mathcal{D}(\phi_2 \circ \phi_1) = \mathcal{D}(\phi_1) + \mathcal{D}(\phi_2) - 1$,

(iii) $l_k^{\phi_2\circ\phi_1} = l_k^{\phi_1}$ *if* $k \in \{0, \dots, \mathcal{D}(\phi_1) - 1\}$,

(iv) $l_k^{\phi_2\circ\phi_1} = l^{\phi_2}_{k+1-\mathcal{D}(\phi_1)}$ *if* $k \in \{\mathcal{D}(\phi_1), \dots, \mathcal{D}(\phi_2 \circ \phi_1)\}$,

(v) $\mathcal{P}(\phi_2 \circ \phi_1) = \mathcal{P}(\phi_1) + \mathcal{P}(\phi_2) + l_1^{\phi_2} l^{\phi_1}_{\mathcal{D}(\phi_1)-1} - l_0^{\phi_2} l_1^{\phi_2} - l^{\phi_1}_{\mathcal{D}(\phi_1)}(l^{\phi_1}_{\mathcal{D}(\phi_1)-1} + 1)$,

(vi) $\mathcal{P}(\phi_2 \circ \phi_1) \le \mathcal{P}(\phi_1)$ *if* $\mathcal{D}(\phi_2) = 1$ *and* $l_1^{\phi_2} \le l^{\phi_1}_{\mathcal{D}(\phi_1)}$

(vii) *and* $\mathcal{P}(\phi_2 \circ \phi_1) \le \mathcal{P}(\phi_2)$ *if* $\mathcal{D}(\phi_1) = 1$ *and* $l_0^{\phi_1} \le l_0^{\phi_2}$.

The next lemma is a direct consequence of the above and will be used later to estimate the number of parameters in our approximating networks.

Lemma 2.2. *Let $a \in C(\mathbb{R},\mathbb{R})$ and $\phi \in \mathcal{N}$. Suppose that $\psi_1, \psi_2 \in \mathcal{N}$ satisfy $\mathcal{D}(\psi_1) = \mathcal{D}(\psi_2) = 2$, $l_0^{\psi_1} = l_2^{\psi_1} = l_0^{\phi}$ and $l_0^{\psi_2} = l_2^{\psi_2} = l_{\mathcal{D}(\phi)}^{\phi}$. Denote $m = l_1^{\psi_1}$ if $\mathcal{D}(\phi) = 1$ and $m = l_{\mathcal{D}(\phi)-1}^{\phi}$ if $\mathcal{D}(\phi) \geq 2$. Then,*

$$\mathcal{P}(\psi_2 \circ \phi \circ \psi_1) = \mathcal{P}(\phi) + l_1^{\psi_1}(l_0^{\phi}+1) + l_1^{\psi_2}(l_{\mathcal{D}(\phi)}^{\phi}+1) + l_1^{\phi}(l_1^{\psi_1} - l_0^{\phi}) + m(l_1^{\psi_2} - l_{\mathcal{D}(\phi)}^{\phi}).$$

Proof. Abbreviate $D = \mathcal{D}(\phi)$. By Proposition 2.1 and the fact that $\mathcal{P}(\psi_1) = l_1^{\psi_1}(l_0^{\psi_1}+1) + l_2^{\psi_1}(l_1^{\psi_1}+1)$, we have

$$\mathcal{P}(\phi \circ \psi_1) = \mathcal{P}(\phi) + l_1^{\psi_1}(l_0^{\phi}+1) + l_1^{\phi}(l_1^{\psi_1} - l_0^{\phi})$$

and $l_D^{\phi\circ\psi_1} = m$. So, by applying Proposition 2.1 once more and observing $l_{D+1}^{\phi\circ\psi_1} = l_D^{\phi} = l_2^{\psi_2}$, we obtain

$$\mathcal{P}(\psi_2 \circ \phi \circ \psi_1) = \mathcal{P}(\phi \circ \psi_1) + l_1^{\psi_2}(l_D^{\phi}+1) + m(l_1^{\psi_2} - l_D^{\phi}),$$

which completes the proof. ∎

The standard parallelization of two network skeletons $\phi_1 = [(V_1, b_1), \ldots, (V_D, b_D)]$ and $\phi_2 = [(W_1, c_1), \ldots, (W_D, c_D)]$ of the same depth is given by $p(\phi_1, \phi_2) =$

$$\left[\left(\begin{bmatrix} V_1 & 0 \\ 0 & W_1 \end{bmatrix}, \begin{bmatrix} b_1 \\ c_1 \end{bmatrix}\right), \ldots, \left(\begin{bmatrix} V_D & 0 \\ 0 & W_D \end{bmatrix}, \begin{bmatrix} b_D \\ c_D \end{bmatrix}\right)\right].$$

From there, arbitrarily many network skeletons $\phi_1, \ldots, \phi_n \in \mathcal{N}$, $n \in \mathbb{N}_{\geq 3}$, of the same depth can be parallelized iteratively:

$$p(\phi_1, \ldots, \phi_n) = p(p(\phi_1, \ldots, \phi_{n-1}), \phi_n).$$

The first three statements of the next proposition follow immediately from the definition. The last one is shown in [49].

Proposition 2.3. *The parallelization*

$$p\colon \bigcup_{n\in\mathbb{N}} \{(\phi_1, \ldots, \phi_n) \in \mathcal{N}^n \colon \mathcal{D}(\phi_1) = \cdots = \mathcal{D}(\phi_n)\} \to \mathcal{N}$$

satisfies for all $\phi_1, \ldots, \phi_n \in \mathcal{N}$, $n \in \mathbb{N}$, with the same depth

(i) $\mathcal{R}_a^{p(\phi_1,\ldots,\phi_n)}(x_1, \ldots, x_n) = (\mathcal{R}_a^{\phi_1}(x_1), \ldots, \mathcal{R}_a^{\phi_n}(x_n))$ *for all* $x_1 \in \mathbb{R}^{l_0^{\phi_1}}, \ldots, x_n \in \mathbb{R}^{l_0^{\phi_n}}$ *and each* $a \in C(\mathbb{R},\mathbb{R})$,

(ii) $l_k^{p(\phi_1,\ldots,\phi_n)} = \sum_{j=1}^n l_k^{\phi_j}$ *for all* $k \in \{0, \ldots, \mathcal{D}(\phi_1)\}$,

(iii) $\mathcal{P}(p(\phi_1, \ldots, \phi_n)) \leq n^2 \mathcal{P}(\phi_1)$ *whenever* $l_k^{\phi_i} = l_k^{\phi_j}$ *for all* $k \in \{0, \ldots, \mathcal{D}(\phi_1)\}$ *and all* $i, j \in \{1, \ldots, n\}$

(iv) *and* $\mathcal{P}(p(\phi_1, \ldots, \phi_n)) \leq \frac{1}{2}\big[\sum_{j=1}^n \mathcal{P}(\phi_j)\big]^2$.

Neural networks with different depths can still be parallelized, but only for a special class of activation functions.

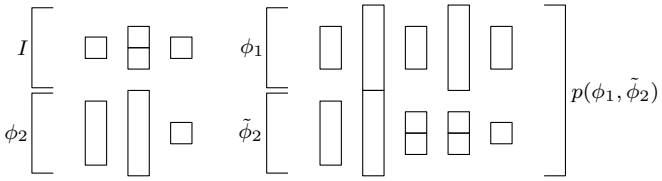

Figure 2.1: Parallelization of a network ϕ_1 (depth 4) and a shorter network ϕ_2 (depth 2) obtained by concatenating ϕ_2 twice with a network I arising from the 2-identity requirement, resulting in $\tilde{\phi}_2 = I \circ I \circ \phi_2$.

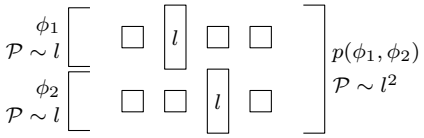

Figure 2.2: The parallelization of ϕ_1 with architecture $(1, l, 1, 1)$ and ϕ_2 with architecture $(1, 1, l, 1)$ has more than l^2 parameters.

Definition 2.4. We say a function $a \in C(\mathbb{R}, \mathbb{R})$ fulfills the c-identity requirement for a number $c \geq 2$ if there exists $I \in \mathcal{N}$ such that $\mathcal{D}(I) = 2$, $l_1^I \leq c$ and $\mathcal{R}_a^I = \mathrm{id}_{\mathbb{R}}$.

Note that if I satisfies $\mathcal{R}_a^I = \mathrm{id}_{\mathbb{R}}$, one can also realize the identity function $\mathrm{id}_{\mathbb{R}^d}$ for any $d \in \mathbb{N}$, using d-fold parallelization $I_d = p(I, \ldots, I)$. Obviously, $l_1^{I_d} \leq cd$.

The most prominent example satisfying Definition 2.4 is the rectified linear unit activation $\mathbb{R} \to \mathbb{R}$, $x \mapsto \max\{x, 0\}$. It fulfills the 2-identity requirement with $I = [([1 \ -1]^T, [0 \ 0]^T), ([1 \ -1], 0)]$. However, it is easy to see that generalized ReLU functions of the form

$$a(x) = \begin{cases} rx & \text{if } x \geq 0 \\ sx & \text{if } x < 0 \end{cases}$$

for $(r, s) \in \mathbb{R}^2$ with $r + s \neq 0$, such as leaky ReLU, also satisfy the 2-identity requirement.[1]

Using the identity requirement, one can parallelize networks of arbitrary depths. If $\phi_1, \ldots, \phi_n \in \mathcal{N}$ have different depths, one simply concatenates the shorter ones with identity networks until all have the same depth. Then one applies the standard parallelization; see Fig. 2.1 for an illustration.

Although this successfully parallelizes networks with arbitrary architecture, one can do better in terms of parameter counts. The estimate in Proposition 2.3.(iv) contains a square of $\sum_{j=1}^n \mathcal{P}(\phi_j)$. This is not due to lax estimates, but a square can actually appear if, for some j, there are two large consecutive layers in $p(\phi_j, \phi_{j+1})$ which in ϕ_j and ϕ_{j+1} were next to small layers; see Fig. 2.2. To avoid this, we introduce a new parallelization which uses identity networks to shift $\phi_1, \ldots, \phi_n$ away from each other and, as a result, achieves a parameter count that is linear in $\sum_{j=1}^n \mathcal{P}(\phi_j)$. For instance, to parallelize ϕ_1 and ϕ_2, we add $\mathcal{D}(\phi_2)$ identity networks after ϕ_1 and $\mathcal{D}(\phi_1)$ identity networks in front of ϕ_2 before applying p. The realization of the resulting network still is $(x_1, x_2) \mapsto (\mathcal{R}_a^{\phi_1}(x_1), \mathcal{R}_a^{\phi_2}(x_2))$. Extending this construction to more than two networks is straight-forward; see Fig. 2.3. We denote it by p_I, where $I \in \mathcal{N}$ is the network satisfying the identity requirement. The following proposition shows that p_I achieves our goal of a linear parameter count in $\sum_{j=1}^n \mathcal{P}(\phi_j)$.

[1] Other activation functions satisfying the identity requirement are polynomials. For example, $\frac{1}{2}((x+1)^2 - x^2 - 1) = x$ shows this for x^2.

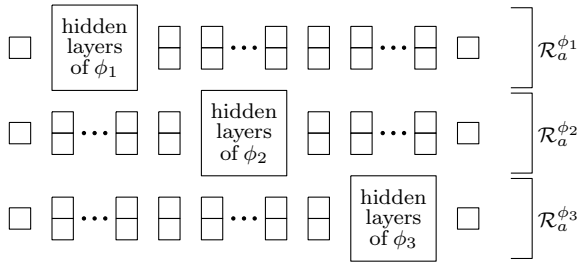

Figure 2.3: New "diagonalized" parallelization resulting from shifting ϕ_1, ϕ_2, ϕ_3 away from each other.

Proposition 2.5. *Assume $a \in C(\mathbb{R}, \mathbb{R})$ fulfills the c-identity requirement for a number $c \geq 2$ with $I \in \mathcal{N}$. Then the parallelization $p_I \colon \bigcup_{n\in\mathbb{N}} \mathcal{N}^n \to \mathcal{N}$ satisfies*

$$\mathcal{P}(p_I(\phi_1, \ldots, \phi_n)) \leq \left(\tfrac{11}{16}c^2 l^2 n^2 - 1\right) \sum_{j=1}^{n} \mathcal{P}(\phi_j)$$

for all $n \in \mathbb{N}$ and $\phi_1, \ldots, \phi_n \in \mathcal{N}$, where we denote $l = \max_{j\in\{1,\ldots,n\}} \max\{l_0^{\phi_j}, l_{\mathcal{D}(\phi_j)}^{\phi_j}\}$.

Proof. Assume without loss of generality that $n \geq 2$. To simplify notation, let us introduce some abbreviations. Write $D_j = \mathcal{D}(\phi_j)$, $l_j^i = l_j^{\phi_i}$, $E_i = \sum_{j=1}^{i} D_j$, $S_i = \sum_{j=1}^{i} l_{D_j}^j$ and $T_i = \sum_{j=i}^{n} l_0^j$. Moreover, denote $c_i = c$ if $i \in \{1, \ldots, n-1\}$ and $c_i = 1$ if $i \in \{0, n\}$. Consider the network architecture $(L_0, \ldots, L_{E_n})$ of depth E_n given by

$$L_k = \begin{cases} c_i S_i + c_i T_{i+1} & \text{if } k = E_i, \\ cS_{i-1} + l_m^i + cT_{i+1} & \text{if } k = E_{i-1} + m, \end{cases}$$

where m is ranging from 1 to $D_i - 1$. As discussed in the paragraph preceding Proposition 2.5, there is a skeleton $\psi \in \mathcal{N}$ with this architecture that realizes the parallelization of $\phi_1, \ldots, \phi_n$; see also Fig. 2.3. One has $\mathcal{P}(\psi) = \sum_{i=1}^{n} P_i$ for $P_i = \sum_{k=E_{i-1}+1}^{E_i} L_k(L_{k-1}+1)$. In the remainder of the proof, we show that $P_i \leq (\frac{11}{16}c^2 l^2 n^2 - 1)\mathcal{P}(\phi_i)$. We distinguish the cases $D_i \geq 2$ and $D_i = 1$. Let us begin with the former case. By the definition of L_k, we have

$$\begin{aligned} P_i = {} & (cS_{i-1} + l_1^i + cT_{i+1})(c_{i-1}S_{i-1} + c_{i-1}T_i + 1) \\ & + \sum_{m=2}^{D_i - 1} (cS_{i-1} + l_m^i + cT_{i+1})(cS_{i-1} + l_{m-1}^i + cT_{i+1} + 1) \\ & + (c_i S_i + c_i T_{i+1})(cS_{i-1} + l_{D_i-1}^i + cT_{i+1} + 1). \end{aligned}$$

Now we use $c \geq 2$, $c \geq c_i$, $S_i = S_{i-1} + l_{D_i}^i$, $T_i = l_0^i + T_{i+1}$ and $S_{i-1} + T_{i+1} \leq l(n-1)$, and reorder the resulting terms to obtain

$$\begin{aligned} P_i \leq {} & \sum_{m=1}^{D_i} l_m^i (l_{m-1}^i + 1) + c^2 l(n-1) \sum_{m=0}^{D_i} l_m^i \\ & + (c-1) l_1^i l_0^i + (c-1) l_{D_i}^i (l_{D_i-1}^i + 1) + D_i cl(n-1)(cl(n-1) + 1). \end{aligned}$$

Then, we bound the second line by $2(c-1)\mathcal{P}(\phi_i)$, the sum $\sum_{m=0}^{D_i} l_m^i$ by $\mathcal{P}(\phi_i)$ and the depth D_i by $\frac{1}{2}\mathcal{P}(\phi_i)$ to find

$$P_i \leq \mathcal{P}(\phi_i)\big[2c - 1 + c^2 l(n-1) + \tfrac{1}{2}cl(n-1)(cl(n-1) + 1)\big].$$

Finally, since $c \geq 2$ and $n \geq 2$, the term in the brackets can be bounded by $\frac{11}{16}c^2l^2n^2 - 1$, and the proposition follows in the case $D_i \geq 2$. Now assume $D_i = 1$ so that $\mathcal{P}(\phi_i) = l_1^i(l_0^i + 1)$. Then, by the same inequalities as in the previous case,

$$\begin{aligned} P_i &= (c_iS_i + c_iT_{i+1})(c_{i-1}S_{i-1} + c_{i-1}T_i + 1) \\ &\leq c^2(l(n-1) + l_1^i)(l(n-1) + l_0^i + \tfrac{1}{2}). \end{aligned}$$

If $l_0^i = 1$, then $\mathcal{P}(\phi_i) = 2l_1^i$ and, hence,

$$\begin{aligned} P_i + \mathcal{P}(\phi_i) &\leq c^2l^2(n - 1 + \tfrac{1}{2}\mathcal{P}(\phi_i))(n + \tfrac{1}{2}) + \mathcal{P}(\phi_i) \\ &\leq c^2l^2\mathcal{P}(\phi_i)\big[\tfrac{n}{2}(n + \tfrac{1}{2}) + \tfrac{1}{4}\big]. \end{aligned}$$

Since $n \geq 2$, we have $\frac{n}{2}(n + \frac{1}{2}) + \frac{1}{4} \leq \frac{11}{16}n^2$, which concludes the case $l_0^i = 1$. Finally, if $l_0^i \geq 2$, then $\mathcal{P}(\phi_i) \geq 3$ and $l \geq 2$, so we obtain

$$\begin{aligned} P_i + \mathcal{P}(\phi_i) &\leq c^2l^2(n - 1 + \tfrac{1}{2}l_1^i)(n - \tfrac{3}{4} + \tfrac{1}{2}l_0^i) + \mathcal{P}(\phi_i) \\ &\leq c^2l^2\big[(n-1)(n - \tfrac{3}{4}) + \tfrac{n-1}{2}l_0^i + (\tfrac{n-1}{2} + \tfrac{1}{8})l_1^i + \tfrac{1}{4}l_0^il_1^i\big] + \tfrac{1}{16}c^2l^2\mathcal{P}(\phi_i) \\ &\leq c^2l^2\mathcal{P}(\phi_i)\big[\tfrac{1}{3}(n-1)(n - \tfrac{3}{4}) + \tfrac{n-1}{2} + \tfrac{5}{16}\big], \end{aligned}$$

and the term in the brackets is bounded by $\frac{1}{3}n^2$, which finishes the last remaining case. ■

It can be seen from the proof that the inequality of Proposition 2.5 is never an equality. However, it can be shown that it is asymptotically sharp up to a constant for large n. Indeed, if $c = 2$ (as is the case for ReLU) and if $\phi_1 = \cdots = \phi_n$ has depth at least two ($\mathcal{D}(\phi_1) \geq 2$) and a single neuron in each layer ($l_k^{\phi_1} = 1$ for all k), then

$$(2n^3 - n^2)\mathcal{P}(\phi_1) = \mathcal{P}(p_I(\phi_1, \ldots, \phi_n)) \leq \big(\tfrac{11}{4}n^3 - n\big)\mathcal{P}(\phi_1).$$

The inequality on the right is a consequence of Proposition 2.5. We verify the equality on the left: with the notation from the previous proof, we have $S_i = i$ and $T_i = n - i + 1$. Thus, the formula for P_i reads

$$P_i = (2n-1)2n(D_i - 2) + \begin{cases} (2n-1)(n+1) + 4n^2, & \text{if } i = 1, \\ (2n-1)(2n+1) + 4n^2, & \text{if } 2 \leq i \leq n-1, \\ (2n-1)(2n+1) + 2n^2, & \text{if } i = n. \end{cases}$$

Since $\mathcal{P}(\phi_1) = 2D_i$, we find for the diagonal parallelization

$$\mathcal{P}(p_I(\phi_1, \ldots, \phi_n)) = P_1 + (n-2)P_2 + P_n = (2n^3 - n^2)\mathcal{P}(\phi_1).$$

Hence, the bound in the proposition is asymptotically sharp up to a factor of at most $\frac{11}{8}$.

Proposition 2.5 illustrates that there is a fundamental difference between counting the number of neurons and counting the number of parameters. As already observed in [49,68,105], this also plays a role for the concatenation. The standard concatenation of two networks ϕ_1 and ϕ_2 has roughly $\mathcal{P}(\phi_1) + \mathcal{P}(\phi_2)$ neurons. But the parameter count may increase much more dramatically. If, e.g., most of the neurons of ϕ_1 are in the last hidden layer and most of the neurons of ϕ_2 in the first hidden layer, then $\phi_2 \circ \phi_1$ has roughly $\mathcal{P}(\phi_1) \cdot \mathcal{P}(\phi_2)$ parameters; see Fig. 2.4. To counter this, one can use the concatenation

$$I_{l^{\phi_2}_{\mathcal{D}(\phi_2)}} \circ \phi_2 \circ I_{l_0^{\phi_2}} \circ I_{l^{\phi_1}_{\mathcal{D}(\phi_1)}} \circ \phi_1 \circ I_{l_0^{\phi_1}}$$

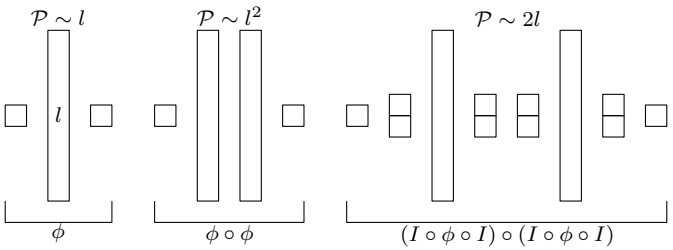

Figure 2.4: Concatenation with and without additional identity networks. Here, ϕ is a network of depth 2 with l neurons in its hidden layer, and I is assumed to satisfy the 2-identity requirement.

instead of $\phi_2 \circ \phi_1$, where I_d is an identity network in d dimensions. Even though this results in more neurons, it reduces the parameter count. The following estimate is a consequence of Lemma 2.2.

Corollary 2.6. *Assume $a \in C(\mathbb{R}, \mathbb{R})$ satisfies the c-identity requirement for a number $c \geq 2$ with $I \in \mathcal{N}$ and denote $I_d = p(I, \dots, I)$ for all $d \in \mathbb{N}$. Let $\phi \in \mathcal{N}$ and abbreviate $m = \max\{l_0^\phi, l_{\mathcal{D}(\phi)}^\phi\}$. Then*

$$\mathcal{P}(I_{l_{\mathcal{D}(\phi)}^\phi} \circ \phi \circ I_{l_0^\phi}) \leq \tfrac{5}{6} cm\mathcal{P}(\phi) + \tfrac{29}{12} c^2 m^2.$$

Proof. Abbreviate $D = \mathcal{D}(\phi)$, $k = l_0^\phi$ and $n = l_{\mathcal{D}(\phi)}^\phi$. First, assume $D \geq 2$. Lemma 2.2 yields

$$\mathcal{P}(I_n \circ \phi \circ I_k) = \mathcal{P}(\phi) + l_1^{I_k}(k+1) + l_1^{I_n}(n+1) + l_1^\phi(l_1^{I_k} - k) + l_{D-1}^\phi(l_1^{I_n} - n).$$

Note that $l_1^{I_k}$ and $l_1^{I_n}$ are at most ck and cn, respectively. This and the fact that $l_1^\phi + l_{D-1}^\phi \leq \frac{2}{3}\mathcal{P}(\phi)$ imply

$$\begin{aligned}\mathcal{P}(I_n \circ \phi \circ I_k) &\leq \mathcal{P}(\phi) + 2cm(m+1) + (l_1^\phi + l_{D-1}^\phi)(cm - 1) \\ &\leq \tfrac{5}{6} cm\mathcal{P}(\phi) + 2c^2m^2,\end{aligned}$$

where the last inequality holds because $c \geq 2$. Now, suppose $D = 1$. Then, by Lemma 2.2,

$$\begin{aligned}\mathcal{P}(I_n \circ \phi \circ I_k) &\leq n + ck(k+1) + cn(n+1) + c^2nk \\ &\leq \tfrac{5}{6} c\mathcal{P}(\phi) + c^2m^2 + \tfrac{7}{6} cm(m+1) + m \\ &\leq \tfrac{5}{6} cm\mathcal{P}(\phi) + \tfrac{29}{12} c^2m^2,\end{aligned}$$

where we again used $c \geq 2$. ∎

Corollary 2.6 will be used in our proofs to estimate the number of parameters of

$$I_{l_{\mathcal{D}(\phi_2)}^{\phi_2}} \circ \phi_2 \circ I_{l_0^{\phi_2}} \circ I_{l_{\mathcal{D}(\phi_1)}^{\phi_1}} \circ \phi_1 \circ I_{l_0^{\phi_1}}.$$

3. Catalog networks

In this section, we generalize the concept of a neural network by allowing the activation functions to change from one layer to the next as long as they belong to a predefined

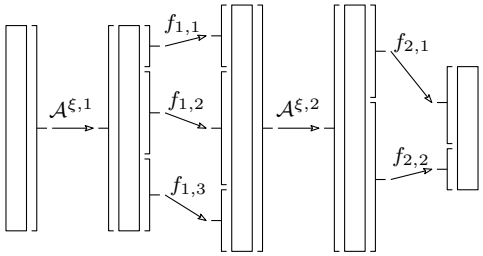

Figure 2.5: Realization of an example catalog network.

catalog $\mathcal{F} \subseteq \bigcup_{m,n\in\mathbb{N}} C(\mathbb{R}^m, \mathbb{R}^n)$. We denote the dimension of the domain of a function $f \in \bigcup_{m,n\in\mathbb{N}} C(\mathbb{R}^m, \mathbb{R}^n)$ by $\mathcal{I}(f)$ and the dimension of its target space by $\mathcal{O}(f)$, so that $f \in C(\mathbb{R}^{\mathcal{I}(f)}, \mathbb{R}^{\mathcal{O}(f)})$. For a catalog $\mathcal{F}$ and numbers $D \in \mathbb{N}$, $l_0, \dots, l_{2D} \in \mathbb{N}$, we define $\mathcal{C}_{\mathcal{F}}^{l_0,\dots,l_{2D}}$ as

$$\prod_{k=1}^{D} \mathbb{R}^{l_{2k-1}\times l_{2k-2}} \times \mathbb{R}^{l_{2k-1}} \times \bigcup_{n\in\mathbb{N}} \Big\{(f_1,\dots,f_n) \in \mathcal{F}^n \colon \sum_{j=1}^{n} \mathcal{I}(f_j) = l_{2k-1} \text{ and } \sum_{j=1}^{n} \mathcal{O}(f_j) = l_{2k}\Big\}.$$

The set of all catalog networks corresponding to $\mathcal{F}$ is given by

$$\mathcal{C}_{\mathcal{F}} = \bigcup_{D\in\mathbb{N}} \bigcup_{l_0,\dots,l_{2D}\in\mathbb{N}} \mathcal{C}_{\mathcal{F}}^{l_0,\dots,l_{2D}}.$$

An element $\xi \in \mathcal{C}_{\mathcal{F}}^{l_0,\dots,l_{2D}}$ is of the form

$$\xi = [(V_1, b_1, (f_{1,1}, \dots, f_{1,n_1})), \dots, (V_D, b_D, (f_{D,1}, \dots, f_{D,n_D}))].$$

For each $k \in \{1, \dots, D\}$, we let $\mathcal{A}^{\xi,k} \in C(\mathbb{R}^{l_{2k-2}}, \mathbb{R}^{l_{2k-1}})$ be the affine function $x \mapsto V_k x + b_k$. By $\mathcal{G}^{\xi,k} \in C(\mathbb{R}^{l_{2k-1}}, \mathbb{R}^{l_{2k}})$, we denote the function mapping $x \in \mathbb{R}^{l_{2k-1}}$ to

$$\mathcal{G}^{\xi,k}(x) = \big[f_{k,1}\big(x_1, \dots, x_{\mathcal{I}(f_{k,1})}\big), f_{k,2}\big(x_{\mathcal{I}(f_{k,1})+1}, \dots, x_{\mathcal{I}(f_{k,1})+\mathcal{I}(f_{k,2})}\big), \dots, f_{k,n_k}\big(x_{\mathcal{I}(f_{k,1})+\cdots+\mathcal{I}(f_{k,n_k-1})+1}, \dots, x_{\mathcal{I}(f_{k,1})+\cdots+\mathcal{I}(f_{k,n_k})}\big)\big],$$

that is, we apply $f_{k,1}$ to the first $\mathcal{I}(f_{k,1})$ entries of x, $f_{k,2}$ to the next $\mathcal{I}(f_{k,2})$ entries and so on; see Fig. 2.5. This is well-defined due to the sum conditions in the definition of $\mathcal{C}_{\mathcal{F}}^{l_0,\dots,l_{2D}}$. The overall realization function $\mathcal{R}^{\xi} \in C(\mathbb{R}^{l_0}, \mathbb{R}^{l_{2D}})$ of the catalog network ξ is

$$\mathcal{R}^{\xi} = \mathcal{G}^{\xi,D} \circ \mathcal{A}^{\xi,D} \circ \cdots \circ \mathcal{G}^{\xi,1} \circ \mathcal{A}^{\xi,1}.$$

We define the depth of ξ as $\mathcal{D}_\xi = D$. Its input dimension is $\mathcal{I}_\xi = l_0$, its output dimension $\mathcal{O}_\xi = l_{2D}$, and its maximal width $\mathcal{W}_\xi = \max\{l_0, \dots, l_{2D}\}$.

Our goal is to show that catalog networks can efficiently be approximated with neural networks with respect to some weight function, by which we mean any function $w \colon [0,\infty) \to (0,\infty)$.

Definition 3.1. We say the decay of a weight function w is controlled by $(s_1, s_2) \in [1,\infty) \times [0,\infty)$ if

$$s_1 r^{s_2} w(r \max\{x, 1\}) \geq w(x)$$

for all $x \in [0,\infty)$ and $r \in [1,\infty)$.

Controlled decay is a general concept applicable to different types of weight functions. The inequality in Definition 3.1 is exactly what is needed in the proofs of our results. Useful weight functions are constants and functions of the form $(1+x^q)^{-1}$ or $(\max\{1,x^q\})^{-1}$ for some $q \in (0,\infty)$. Constant weight functions have decay controlled by $(1,0)$. The functions $(1+x^q)^{-1}$ and $(\max\{1,x^q\})^{-1}$ are covered by the following result.

Lemma 3.2. *Let $\delta \in (0,\infty)$ and consider a nondecreasing function $f\colon [0,\infty) \to (0,\infty)$. Moreover, let $g\colon [0,\infty) \to [0,\infty)$ be of the form $x \mapsto \sum_{j=0}^{q} a_j x^{b_j}$ for $q \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$ and $a_0, b_0, \dots, a_q, b_q \in [0,\infty)$. Then the decay of the weight function $w(x) = f(x)(\max\{g(x),\delta\})^{-1}$ is controlled by $(\max\{g(1)/\delta, 1\}, \max\{b_0,\dots,b_q\})$.*

Proof. Denote $s = \max\{b_0,\dots,b_q\}$. Since the coefficients $a_0,\dots,a_q$ of g are nonnegative, one has $g(rx) \leq r^s g(x)$ for all $x \in [0,\infty)$ and $r \in [1,\infty)$. This and the assumption that f is nondecreasing yield

$$w(x) \leq \frac{f(rx)}{\max\{g(x),\delta\}} \leq \frac{f(rx)r^s}{\max\{g(rx),\delta\}} = r^s w(rx)$$

for all $x \in [0,\infty)$, $r \in [1,\infty)$. That f is nondecreasing also gives

$$w(x) \leq \frac{f(1)}{\max\{g(x),\delta\}} \leq \frac{\max\{g(1),\delta\}}{\delta} w(1)$$

for all $x \in [0,1)$. Combining the previous two estimates yields

$$\frac{\delta}{\max\{g(1),\delta\}} w(x) \leq w(\max\{x,1\}) \leq r^s w(r\max\{x,1\})$$

for all $x \in [0,\infty)$, $r \in [1,\infty)$, which finishes the proof. ∎

Our main interest is in catalogs of functions that are well approximable with neural networks. For the proofs of our main results to work we need the approximations to be Lipschitz continuous with a Lipschitz constant independent of the accuracy. To make this precise, we denote the Euclidean norm by $\|\cdot\|$.

Definition 3.3. Consider an activation function $a \in C(\mathbb{R},\mathbb{R})$ and a weight function w. Fix constants $L \in [0,\infty)$ and $\varepsilon \in (0,1]$. Given a function $f \in \bigcup_{m,n\in\mathbb{N}} C(\mathbb{R}^m,\mathbb{R}^n)$ and a set $Q \subseteq \mathbb{R}^{\mathcal{I}(f)}$, we define the approximation cost $\mathrm{Cost}_{a,w}(f,Q,L,\varepsilon)$ as the infimum of the set

$$\left\{ \mathcal{P}(\phi) \in \mathbb{N}\colon \begin{array}{l} \phi \in \mathcal{N} \text{ with } \mathcal{R}_a^\phi \in C(\mathbb{R}^{\mathcal{I}(f)},\mathbb{R}^{\mathcal{O}(f)}) \\ \text{s.t. } \mathcal{R}_a^\phi \text{ is } L\text{-Lipschitz on } \mathbb{R}^{\mathcal{I}(f)} \text{ and} \\ \sup_{x\in Q} w(\|x\|)\|f(x) - \mathcal{R}_a^\phi(x)\| \leq \varepsilon \end{array} \right\},$$

where, as usual, $\inf(\emptyset)$ is understood as ∞.

The next definition specifies the class of catalogs for which we will be able to prove Theorem 5.2 on the approximability of catalog networks.

Definition 3.4. Let $a \in C(\mathbb{R},\mathbb{R})$, $\kappa = (\kappa_0,\kappa_1,\kappa_2,\kappa_3) \in [1,\infty)^2 \times [0,\infty)^2$, $\varepsilon \in (0,1]$, and suppose w is a weight function. Consider a subset $\mathcal{F} \subseteq \bigcup_{m,n\in\mathbb{N}} C(\mathbb{R}^m,\mathbb{R}^n)$ together with a family of sets $Q = (Q_f)_{f\in\mathcal{F}}$ such that $Q_f \subseteq \mathbb{R}^{\mathcal{I}(f)}$ contains 0 for all $f \in \mathcal{F}$ and a collection of

Lipschitz constants $L = (L_f)_{f\in\mathcal{F}} \subseteq [0,\infty)$. Then we call $\mathcal{F}$ an $[a, w, Q, L, \varepsilon, \kappa]$-approximable catalog if $\sup_{f\in\mathcal{F}} \|f(0)\| \leq \kappa_0$ and

$$\mathrm{Cost}_{a,w}(f, Q_f, L_f, \delta) \leq \kappa_1 \max\{\mathcal{I}(f), \mathcal{O}(f)\}^{\kappa_2}\delta^{-\kappa_3}$$

for all $f \in \mathcal{F}$ and $\delta \in (0, \varepsilon]$.

Note that if $\mathcal{F}$ is $[a, w, Q, L, \varepsilon, \kappa]$-approximable, then every $f \in \mathcal{F}$ must be L_f-Lipschitz continuous on the set Q_f. Indeed, the definition implies that for all $\delta \in (0, \varepsilon]$ there exists $\phi_\delta \in \mathcal{N}$ such that $w(\|x\|)\|f(x) - \mathcal{R}_a^{\phi_\delta}(x)\| \leq \delta$ and $\|\mathcal{R}_a^{\phi_\delta}(x) - \mathcal{R}_a^{\phi_\delta}(y)\| \leq L_f\|x - y\|$ for all $x, y \in Q_f$. Hence, one obtains from the triangle inequality that

$$\|f(x) - f(y)\| \leq \frac{\delta}{w(\|x\|)} + L_f\|x - y\| + \frac{\delta}{w(\|y\|)}$$

for all $x, y \in Q_f$ and $\delta > 0$, which shows that f is L_f-Lipschitz on Q_f.

If $\mathcal{F}$ is a catalog approximable on sets $Q = (Q_f)_{f\in\mathcal{F}}$ with Lipschitz constants $L = (L_f)_{f\in\mathcal{F}} \subseteq [0,\infty)$, we define for a catalog network $\xi \in \mathcal{C}_{\mathcal{F}}^{l_0,\ldots,l_{2D}}$ of the form $\xi = [(V_1, b_1, (f_{1,1}, \ldots, f_{1,n_1})), \ldots, (V_D, b_D, (f_{D,1}, \ldots, f_{D,n_D}))]$,

$$Q^{\xi,k} = \prod_{j=1}^{n_k} Q_{f_{k,j}} \subseteq \prod_{j=1}^{n_k} \mathbb{R}^{\mathcal{I}(f_{k,j})} = \mathbb{R}^{l_{2k-1}}$$

and

$$L^{\xi,k} = \max_{j\in\{1,\ldots,n_k\}} L_{f_{k,j}}$$

for all $k \in \{1, \ldots, \mathcal{D}_\xi\}$. Then the following holds.

Lemma 3.5. *Let $\xi \in \mathcal{C}_{\mathcal{F}}$ be a catalog network based on an $[a, w, Q, L, \varepsilon, \kappa]$-approximable catalog $\mathcal{F}$. Then*

$$\|\mathcal{G}^{\xi,k}(x) - \mathcal{G}^{\xi,k}(y)\| \leq L^{\xi,k}\|x - y\|$$

for all $k \in \{1, \ldots, \mathcal{D}_\xi\}$ and $x, y \in Q^{\xi,k}$.

Proof. Assume ξ is of the form $[(V_1, b_1, (f_{1,1}, \ldots, f_{1,n_1})), \ldots, (V_D, b_D, (f_{D,1}, \ldots, f_{D,n_D}))]$. As discussed after Definition 3.4, every $f \in \mathcal{F}$ is L_f-Lipschitz continuous on the set Q_f. For $k \in \{1, \ldots, D\}$, $j \in \{1, \ldots, n_k\}$ and $x \in \mathbb{R}^{l_{2k-1}}$, denote by $x_{(k,j)}$ the vector

$$x_{(k,j)} = (x_{\mathcal{I}(f_{k,1})+\cdots+\mathcal{I}(f_{k,j-1})+1}, \ldots, x_{\mathcal{I}(f_{k,1})+\cdots+\mathcal{I}(f_{k,j})}) \in \mathbb{R}^{\mathcal{I}(f_{k,j})}.$$

Then, for all $k \in \{1, \ldots, D\}$ and $x, y \in Q^{\xi,k}$,

$$\begin{aligned}\|\mathcal{G}^{\xi,k}(x) - \mathcal{G}^{\xi,k}(y)\|^2 &= \sum_{j=1}^{n_k} \|f_{k,j}(x_{(k,j)}) - f_{k,j}(y_{(k,j)})\|^2 \\ &\leq \sum_{j=1}^{n_k} L_{f_{k,j}}^2\|x_{(k,j)} - y_{(k,j)}\|^2 \leq [L^{\xi,k}]^2\|x - y\|^2,\end{aligned} \tag{3.1}$$

which is what we wanted to show. ∎

4. Examples of approximable catalogs

In this section, we provide different examples of approximable catalogs that will be used in Section 7 to show that various high-dimensional functions admit neural network approximations without the curse of dimensionality. Our catalogs are based on one-dimensional Lipschitz functions, the maximum function, the square, the product and the decreasing exponential function. They will be collected in Examples 4.1, 4.2, 4.4, and 4.6.

First, consider a K-Lipschitz function $f\colon \mathbb{R} \to \mathbb{R}$ for a constant $K \in [0,\infty)$. For any given $r \in (0,\infty)$, f can be approximated on $[-r,r]$ with a piecewise linear function supported on $N+1$ equidistributed points with accuracy Kr/N. Such a piecewise linear function can be realized with a ReLU network ϕ_N with one hidden layer and N hidden neurons. This results in $\mathcal{P}(\phi_N) = 3N+1$, from which it follows that

$$\mathrm{Cost}_{\mathrm{ReLU},1}(f,[-r,r],K,\varepsilon) \leq \mathcal{P}(\phi_{\lceil Kr\varepsilon^{-1}\rceil}) \leq 3Kr\varepsilon^{-1}+4.$$

Alternatively, one can approximate f on the entire real line with respect to a weight function of the form $w_q(x) = (1+x^q)^{-1}$ for some $q \in (1,\infty)$. Then

$$\mathrm{Cost}_{\mathrm{ReLU},w_q}(f,\mathbb{R},K,\varepsilon) \leq 2^{1/q-1}3(K\varepsilon^{-1})^{q/q-1}+4,$$

the proof of which is a variant of [60, Corollary 3.13]. Indeed, set $r = (2K\varepsilon^{-1})^{1/(q-1)}$ and $N = \lceil Kr\varepsilon^{-1}\rceil$. Using ϕ_N as above, we have $|f(x)-\phi_N(x)| \leq \varepsilon$ for all $x \in [-r,r]$ and $|f(x)-\phi_N(x)| \leq 2K|x|$ for all $x \in \mathbb{R}\backslash[-r,r]$. The choice of r then ensures that $w_q(|x|)|f(x)-\phi_N(x)| \leq \varepsilon$ for all $x \in \mathbb{R}$. In the notation of approximable catalogs, we can summarize as follows.

Example 4.1. *Let $r \in (0,\infty)$ and consider the weight function $w_q(x) = (1+x^q)^{-1}$ for a $q \in (1,\infty)$. For $K \in [1,\infty)$, introduce the K-Lipschitz catalog*

$$\mathcal{F}_K^{\mathrm{Lip}} = \big\{f \in C(\mathbb{R},\mathbb{R})\colon f \text{ is } K\text{-Lipschitz and } |f(0)| \leq K\big\}.$$

Set $L_{\mathrm{id}_\mathbb{R}} = 1$ and $L_f = K$ for $f \in \mathcal{F}_K^{\mathrm{Lip}}\backslash\{\mathrm{id}_\mathbb{R}\}$. If we define approximation sets by $Q_{\mathrm{id}_\mathbb{R}} = \mathbb{R}$ and

(i) *$Q_f = [-r,r]$ for all $f \in \mathcal{F}_K^{\mathrm{Lip}}\backslash\{\mathrm{id}_\mathbb{R}\}$, then $\mathcal{F}_K^{\mathrm{Lip}}$ is a $[\rho,1,Q,L,1,\kappa]$-approximable catalog for $\kappa = (K,3Kr+4,0,1)$*

(ii) *$Q_f = \mathbb{R}$ for all $f \in \mathcal{F}_K^{\mathrm{Lip}}\backslash\{\mathrm{id}_\mathbb{R}\}$, then $\mathcal{F}_K^{\mathrm{Lip}}$ is a $[\rho,w_q,Q,L,1,\kappa]$-approximable catalog for[2] $\kappa = (K,5(2K)^{q/(q-1)},0,q/q-1)$.*

Let us now turn to the maximum functions $\max_d\colon \mathbb{R}^d \to \mathbb{R}$, $x \mapsto \max\{x_1,\dots,x_d\}$, $d \in \mathbb{N}$. They admit an exact representation with ReLU networks. Indeed, $\max_1$ is simply the identity and $\max_2$ is the ReLU-realization of

$$\phi_2 = \left[\left(\begin{bmatrix}1 & -1\\ 0 & 1\\ 0 & -1\end{bmatrix}, \begin{bmatrix}0\\0\\0\end{bmatrix}\right), \left(\begin{bmatrix}1 & 1 & -1\end{bmatrix}, 0\right)\right].$$

If $I \in \mathcal{N}$ is a skeleton for which ReLU satisfies the 2-identity requirement and we define $I_d = p(I,\dots,I)$, $d \in \mathbb{N}$, then it easily follows by induction that $\max_d$, $d \in \mathbb{N}_{\geq 3}$, is the

[2]Here we use that $4 \leq 2(2K\varepsilon^{-1})^{q/(q-1)}$ since $K \geq 1$.

ReLU-realization of $\phi_d = \phi_{d-1} \circ p(\phi_2, I_{d-2})$, whose architecture is $(d, 2d-1, 2d-3, \dots, 3, 1)$. From this, we obtain $\mathcal{P}(\phi_d) = \frac{1}{3}(4d^3 + 3d^2 - 4d + 3) \leq 2d^3$. In other words, for all $d \in \mathbb{N}$ and any weight function w,

$$\mathrm{Cost}_{\mathrm{ReLU},w}(\max_d, \mathbb{R}^d, 1, \varepsilon) \leq 2d^3.$$

Adding the maximum functions to the Lipschitz catalog, we obtain the following.

Example 4.2. *Adopt the setting of Example 4.1 and define the K-Lipschitz-maximum catalog $\mathcal{F}_K^{\mathrm{Lip,max}} = \mathcal{F}_K^{\mathrm{Lip}} \cup \{\max_d \colon d \in \mathbb{N}\}$. Add the approximation set $Q_{\max_d} = \mathbb{R}^d$ and the Lipschitz constant $L_{\max_d} = 1$ for all $d \in \mathbb{N}$. Then $\mathcal{F}_K^{\mathrm{Lip,max}}$ is*

(i) *a $[\rho, 1, Q, L, 1, \kappa]$-approximable catalog for $\kappa = (K, 3Kr + 4, 3, 1)$ and Q as in Example 4.1.(i).*

(ii) *a $[\rho, w_q, Q, L, 1, \kappa]$-approximable catalog for $\kappa = (K, 5(2K)^{q/(q-1)}, 3, q/q-1)$ and Q as in Example 4.1.(ii).*

Next, we study the approximability of the square function $\mathrm{sq} \colon \mathbb{R} \to \mathbb{R},\ x \mapsto x^2$. It has been shown by different authors that it can be approximated with accuracy $\varepsilon > 0$ on the unit interval by the ReLU-realization of a skeleton $\phi_\varepsilon \in \mathcal{N}$ satisfying $\mathcal{P}(\phi_\varepsilon) = \mathcal{O}(\log_2(\varepsilon^{-1}))$; see [38, 49, 113, 129]. A precise estimate of the required number of parameters is given in Proposition 3.3 of [49]. In our language it can be stated as

$$\mathrm{Cost}_{\mathrm{ReLU},1}(\mathrm{sq}, [0,1], 2, \varepsilon) \leq \max\{13, 10\log_2(\varepsilon^{-1}) - 7\}.$$

Moreover, the neural network $\mathcal{R}_{\mathrm{ReLU}}^{\phi_\varepsilon}$ achieving this cost is 2-Lipschitz and satisfies $\mathcal{R}_{\mathrm{ReLU}}^{\phi_\varepsilon} = \mathrm{ReLU}$ on $\mathbb{R}\backslash[0,1]$. Using a mirroring and scaling argument, we can deduce the following estimate for approximating the square function on the interval $[-r, r]$ for any $r \in (0, \infty)$.

Lemma 4.3. *For all $r \in (0, \infty)$ and $\varepsilon \in (0, 1]$, there exists a skeleton $\psi_{r,\varepsilon} \in \mathcal{N}$ such that $\mathcal{R}_{\mathrm{ReLU}}^{\psi_{r,\varepsilon}} \in C(\mathbb{R}, \mathbb{R})$ is $2r$-Lipschitz, $\sup_{x \in [-r,r]} |\mathcal{R}_{\mathrm{ReLU}}^{\psi_{r,\varepsilon}}(x) - x^2| \leq \varepsilon$, $\mathcal{R}_{\mathrm{ReLU}}^{\psi_{r,\varepsilon}}(x) = r|x|$ for all $x \in \mathbb{R}\backslash[-r, r]$ and*

$$\mathcal{P}(\psi_{r,\varepsilon}) \leq \max\{52, 80\log_2(r) + 40\log_2(\varepsilon^{-1}) - 28\}.$$

Proof. Choose $\phi_{r,1}, \phi_{r,2} \in \mathcal{N}$ of depth 1 such that $\mathcal{R}_{\mathrm{ReLU}}^{\phi_{r,1}} \in C(\mathbb{R}, \mathbb{R}^2)$ realizes $x \mapsto (\frac{x}{r}, -\frac{x}{r})$ and $\mathcal{R}_{\mathrm{ReLU}}^{\phi_{r,2}} \in C(\mathbb{R}^2, \mathbb{R})$ realizes $(x, y) \mapsto r^2(x+y)$. If $(\phi_\varepsilon)_{\varepsilon \in (0,1]} \subseteq \mathcal{N}$ are the ε-approximations of the square function on $[0,1]$ derived in Proposition 3.3 of [49], then[3] the ReLU-realization of $\psi_{r,\varepsilon} = \phi_{r,2} \circ p(\phi_{r^{-2}\varepsilon}, \phi_{r^{-2}\varepsilon}) \circ \phi_{r,1}$ approximates the square function on $[-r, r]$ with accuracy ε. To see this, note that $\mathcal{R}_{\mathrm{ReLU}}^{\psi_{r,\varepsilon}}(x) = r^2 \mathcal{R}_{\mathrm{ReLU}}^{\phi_{r^{-2}\varepsilon}}(\frac{|x|}{r})$ for all $x \in \mathbb{R}$ since $\mathcal{R}_{\mathrm{ReLU}}^{\phi_{r^{-2}\varepsilon}} = \mathrm{ReLU}$ on $\mathbb{R}\backslash[0,1]$. This also implies $\mathcal{R}_{\mathrm{ReLU}}^{\psi_{r,\varepsilon}}(x) = r|x|$ for all $x \in \mathbb{R}\backslash[-r, r]$ as well as the $2r$-Lipschitz continuity. Finally, (vi) and (vii) of Proposition 2.1 together with (iii) of Proposition 2.3 assure that $\mathcal{P}(\psi_{r,\varepsilon}) \leq 4\mathcal{P}(\phi_{r^{-2}\varepsilon})$, which concludes the proof. ∎

More concisely, for all $r \in [2, \infty)$ and $\varepsilon \in (0, 1]$, the statement of Lemma 4.3 can be written as

$$\mathrm{Cost}_{\mathrm{ReLU},1}(\mathrm{sq}, [-r, r], 2r, \varepsilon) \leq 80\log_2(r) + 40\log_2(\varepsilon^{-1}) - 28.$$

[3] Here, we understand $\phi_{r^{-2}\varepsilon}$ as ϕ_1 if $r^{-2}\varepsilon > 1$.

Now, let us take a closer look at the decreasing exponential function $\mathrm{e}\colon \mathbb{R} \to \mathbb{R},\ x \mapsto e^{-x}$. Its restriction to $[0,\infty)$ is covered by the general approximation result for Lipschitz functions. But exploiting its exponential decrease, we can obtain better estimates. More precisely, e can be approximated to a given accuracy $\varepsilon \in (0,1]$ uniformly on $[0,\infty)$ with a piecewise linear interpolation supported on the $\lfloor \varepsilon^{-1} \rfloor$ points $-\log(n\varepsilon)$, $n \in \{1,\dots,\lfloor \varepsilon^{-1} \rfloor\}$ which is constant on $\mathbb{R}\backslash[-\log(\lfloor \varepsilon^{-1} \rfloor \varepsilon), -\log(\varepsilon)]$. Realizing this piecewise linear function with a ReLU network with one hidden layer yields

$$\mathrm{Cost}_{\mathrm{ReLU},1}(\mathrm{e}, [0,\infty), 1, \varepsilon) \leq 3\lfloor \varepsilon^{-1} \rfloor + 1 \leq 4\varepsilon^{-1}.$$

Together with $\mathrm{id}_{\mathbb{R}}$ and sq, e gives rise to the following catalog, which we will use to approximate generalized Gaussian radial basis function networks in Section 7.

Example 4.4. *Let $r \in [5,\infty)$. Define the catalog $\mathcal{F}^{\mathrm{RBF}} = \{\mathrm{id}_{\mathbb{R}}, \mathrm{e}, \mathrm{sq}\}$ with approximation sets $Q_{\mathrm{id}_{\mathbb{R}}} = \mathbb{R}$, $Q_{\mathrm{e}} = [0,\infty)$, $Q_{\mathrm{sq}} = [-r,r]$ and Lipschitz constants $L_{\mathrm{id}_{\mathbb{R}}} = L_{\mathrm{e}} = 1$, $L_{\mathrm{sq}} = 2r$. Then $\mathcal{F}^{\mathrm{RBF}}$ is[4] a $[\rho, 1, Q, L, r^{-3}, (1,4,0,1)]$-approximable catalog.*

Using the identity $xy = \frac{1}{4}((x+y)^2 - (x-y)^2)$, we can also estimate the approximation rate of the product function $\mathrm{pr}\colon \mathbb{R}^2 \to \mathbb{R},\ (x,y) \mapsto xy$. This trick has already been used before by, e.g., [87, 129]. We still provide a proof of the following proposition since the results in the existing literature do not specify the Lipschitz constant. Our proofs of both, Lemma 4.3 and Proposition 4.5, follow the reasoning of Section 3 in [49].

Proposition 4.5. *For all $r \in (0,\infty)$ and $\varepsilon \in (0, \frac{1}{2}]$, one has*

$$\mathrm{Cost}_{\mathrm{ReLU},1}(\mathrm{pr}, [-r,r]^2, \sqrt{8}r, \varepsilon) \leq \max\{208, 320\log_2(r) + 160\log_2(\varepsilon^{-1}) + 48\}.$$

Proof. Choose $\psi_1, \psi_2 \in \mathcal{N}$ of depth 1 such that $\mathcal{R}^{\psi_1}_{\mathrm{ReLU}} \in C(\mathbb{R}^2, \mathbb{R}^2)$ realizes $(x,y) \mapsto (x+y, x-y)$ and $\mathcal{R}^{\psi_2}_{\mathrm{ReLU}} \in C(\mathbb{R}^2, \mathbb{R})$ realizes $(x,y) \mapsto \frac{1}{4}(x-y)$. If $(\psi_{r,\varepsilon})_{r\in(0,\infty),\varepsilon\in(0,1]} \subseteq \mathcal{N}$ denote the ε-approximations of the square function on the interval $[-r,r]$ from Lemma 4.3, then the ReLU-realization of $\chi_{r,\varepsilon} = \psi_2 \circ p(\psi_{2r,2\varepsilon}, \psi_{2r,2\varepsilon}) \circ \psi_1$ approximates the product function on $[-r,r]^2$ with accuracy ε. Furthermore, $\mathcal{R}^{\chi_{r,\varepsilon}}_{\mathrm{ReLU}}$ is $\sqrt{8}r$-Lipschitz continuous because $\mathcal{R}^{\psi_{2r,2\varepsilon}}_{\mathrm{ReLU}}$ is $4r$-Lipschitz continuous and, hence,

$$\begin{aligned} |\mathcal{R}^{\chi_{r,\varepsilon}}_{\mathrm{ReLU}}(x_1,x_2) - \mathcal{R}^{\chi_{r,\varepsilon}}_{\mathrm{ReLU}}(y_1,y_2)| &\leq 2r(|x_1-y_1| + |x_2-y_2|) \\ &\leq \sqrt{8}r\|(x_1-y_1, x_2-y_2)\|. \end{aligned}$$

As in the proof of Lemma 4.3, combining (vi) and (vii) of Proposition 2.1 with (iii) of Proposition 2.3, shows that $\mathcal{P}(\chi_{r,\varepsilon}) \leq 4\mathcal{P}(\psi_{2r,2\varepsilon})$, from which the proposition follows. ∎

The following is our last example of an approximable catalog.

Example 4.6. *Take $\mathcal{F}^{\mathrm{Lip}}_K$ from Example 4.1, let $R \in (0,\infty)$, and define the K-Lipschitz-product catalog $\mathcal{F}^{\mathrm{Lip,prod}}_K = \mathcal{F}^{\mathrm{Lip}}_K \cup \{\mathrm{pr}\}$. The approximation sets and Lipschitz constants are defined as in (i) of Example 4.1 for $\mathcal{F}^{\mathrm{Lip}}_K$ and $Q_{\mathrm{pr}} = [-R,R]^2$, $L_{\mathrm{pr}} = \sqrt{8}R$. Then $\mathcal{F}^{\mathrm{Lip,prod}}_K$ is a $[\rho, 1, Q, L, \delta, (K, M, 0, 1)]$-approximable catalog for[5] $\delta = \min\{1/2, R^2/2\}$ and $M = \max\{3Kr + 4, 105R^2\}$.*

[4] That $r \geq 5$ and $\varepsilon \leq r^{-3}$ shows $80\log_2(r) + 40\log_2(\varepsilon^{-1}) - 28 \leq 4\varepsilon^{-1}$.

[5] This specific choice of δ and M ensures that $\max\{208, 320\log_2(R) + 160\log_2(\varepsilon^{-1}) + 48\} \leq M\varepsilon^{-1}$ for all $\varepsilon \in (0,\delta]$.

5. Approximation results

In this section, we state the first of our main results, Theorem 5.2, on the approximability of catalog networks with neural networks and explore the special case of ReLU activation in Corollaries 5.5 and 5.6. The next lemma is crucial for the proof of Theorem 5.2. It establishes the approximability of the functions $\mathcal{G}^{\xi,k}$, $k \in \{1,\dots,\mathcal{D}_\xi\}$, in a catalog network $\xi \in \mathcal{C}_\mathcal{F}$. Since $\mathcal{G}^{\xi,k}$ is composed of functions $f_{k,1},\dots,f_{k,n_k}$ from the catalog $\mathcal{F}$, it can be approximated by approximating $f_{k,1},\dots,f_{k,n_k}$ with neural networks and then parallelizing them as in Fig. 2.3. Proposition 2.5 allows us to keep track of the resulting number of parameters.

Lemma 5.1. *Assume $a \in C(\mathbb{R},\mathbb{R})$ satisfies the c-identity requirement for some $c \geq 2$. Let $\mathcal{F}$ be an $[a,w,Q,L,\varepsilon,\kappa]$-approximable catalog for a nonincreasing weight function w, and consider a catalog network $\xi \in \mathcal{C}_\mathcal{F}^{l_0,\dots,l_{2D}}$ for some $D \in \mathbb{N}$ and $l_0,\dots,l_{2D} \in \mathbb{N}$. Then for all $k \in \{1,\dots,D\}$ and $\delta \in (0,\varepsilon]$, there exists a skeleton $\phi \in \mathcal{N}$ with a-realization $\mathcal{R}_a^\phi \in C(\mathbb{R}^{l_{2k-1}},\mathbb{R}^{l_{2k}})$ such that*

(i) $\sup_{x\in Q^{\xi,k}} w(\|x\|)\|\mathcal{G}^{\xi,k}(x) - \mathcal{R}_a^\phi(x)\| \leq \delta$,

(ii) $\mathcal{R}_a^\phi$ *is $L^{\xi,k}$-Lipschitz continuous on $\mathbb{R}^{l_{2k-1}}$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{11}{16}\kappa_1 c^2 \max\{l_{2k-1},l_{2k}\}^{\kappa_2+\kappa_3/2+5}\delta^{-\kappa_3}$.

If, in addition, $\mathcal{I}(f) \leq d$ and $\mathcal{O}(f) \leq d$ for some $d \in \mathbb{N}$ and all $f \in \mathcal{F}$, then one also has

(iv) $\mathcal{P}(\phi) \leq \frac{11}{16}\kappa_1 c^2 d^2 \max\{l_{2k-1},l_{2k}\}^{\kappa_2+\kappa_3/2+3}\delta^{-\kappa_3}$.

Proof. Assume ξ is of the form $[(V_1,b_1,(f_{1,1},\dots,f_{1,n_1})),\dots,(V_D,b_D,(f_{D,1},\dots,f_{D,n_D}))]$. Fix any $k \in \{1,\dots,D\}$ and $\delta \in (0,\varepsilon]$. The assumption that $\mathcal{F}$ is $[a,w,Q,L,\varepsilon,\kappa]$-approximable for $\kappa = (\kappa_0,\kappa_1,\kappa_2,\kappa_3)$ guarantees that there exist skeletons $\psi_j \in \mathcal{N}$, $j \in \{1,\dots,n_k\}$, such that the a-realizations $\mathcal{R}_a^{\psi_j} \in C(\mathbb{R}^{\mathcal{I}(f_{k,j})},\mathbb{R}^{\mathcal{O}(f_{k,j})})$ satisfy

(i) $\|\mathcal{R}_a^{\psi_j}(x) - \mathcal{R}_a^{\psi_j}(y)\| \leq L_{f_{k,j}}\|x-y\|$ for all $x,y \in \mathbb{R}^{\mathcal{I}(f_{k,j})}$,

(ii) $w(\|x\|)\|f_{k,j}(x) - \mathcal{R}_a^{\psi_j}(x)\| \leq \frac{\delta}{\sqrt{n_k}}$ for all $x \in Q_{f_{k,j}}$

(iii) and $\mathcal{P}(\psi_j) \leq \kappa_1 \max\{\mathcal{I}(f_{k,j}),\mathcal{O}(f_{k,j})\}^{\kappa_2} n_k^{\kappa_3/2}\delta^{-\kappa_3}$.

Pick an $I \in \mathcal{N}$ such that a fulfills the c-identity requirement with I, and let $\phi \in \mathcal{N}$ be the I-parallelization $\phi = p_I(\psi_1,\dots,\psi_{n_k})$. For $j \in \{1,\dots,n_k\}$ and $x \in \mathbb{R}^{l_{2k-1}}$, denote $x_{(k,j)} = (x_{\mathcal{I}(f_{k,1})+\cdots+\mathcal{I}(f_{k,j-1})+1},\dots,x_{\mathcal{I}(f_{k,1})+\cdots+\mathcal{I}(f_{k,j})})$. Then, for all $x,y \in \mathbb{R}^{l_{2k-1}}$,

$$\begin{aligned}\|\mathcal{R}_a^\phi(x) - \mathcal{R}_a^\phi(y)\|^2 &= \sum_{j=1}^{n_k} \|\mathcal{R}_a^{\psi_j}(x_{(k,j)}) - \mathcal{R}_a^{\psi_j}(y_{(k,j)})\|^2 \\ &\leq \sum_{j=1}^{n_k} L_{f_{k,j}}^2 \|x_{(k,j)} - y_{(k,j)}\|^2 \leq [L^{\xi,k}]^2\|x-y\|^2.\end{aligned} \tag{5.1}$$

Moreover, since w is nonincreasing, we obtain, for all $x \in Q^{\xi,k}$,

$$\begin{aligned}\|\mathcal{G}^{\xi,k}(x) - \mathcal{R}_a^\phi(x)\|^2 &= \sum_{j=1}^{n_k} \|f_{k,j}(x_{(k,j)}) - \mathcal{R}_a^{\psi_j}(x_{(k,j)})\|^2 \\ &\leq \frac{\delta^2}{n_k}\sum_{j=1}^{n_k}[w(\|x_{(k,j)}\|)]^{-2} \leq \delta^2[w(\|x\|)]^{-2}.\end{aligned} \tag{5.2}$$

It remains to estimate the number of parameters $\mathcal{P}(\phi)$. Since $l_0^{\psi_j} = \mathcal{I}(f_{k,j}) \leq l_{2k-1}$ and $l_{\mathcal{D}(\psi_j)}^{\psi_j} = \mathcal{O}(f_{k,j}) \leq l_{2k}$ for each $j \in \{1, \dots, n_k\}$, Proposition 2.5 yields

$$\begin{aligned} \mathcal{P}(\phi) &\leq \tfrac{11}{16} c^2 \max\{l_{2k-1}, l_{2k}\}^2 n_k^2 \sum_{j=1}^{n_k} \mathcal{P}(\psi_j) \\ &\leq \tfrac{11}{16} \kappa_1 c^2 \max\{l_{2k-1}, l_{2k}\}^{\kappa_2+2} n_k^{3+\frac{\kappa_3}{2}} \delta^{-\kappa_3}. \end{aligned} \tag{5.3}$$

Note that we always have $n_k \leq \max\{l_{2k-1}, l_{2k}\}$, which yields (iii). If $\mathcal{I}(f) \leq d$ and $\mathcal{O}(f) \leq d$ for some $d \in \mathbb{N}$ and all $f \in \mathcal{F}$, then we use the estimate $l_0^{\psi_j} = \mathcal{I}(f_{k,j}) \leq d$ instead of $l_0^{\psi_j} = \mathcal{I}(f_{k,j}) \leq l_{2k-1}$ (and similarly for $l_{\mathcal{D}(\psi_j)}^{\psi_j}$) to obtain $\mathcal{P}(\phi) \leq \frac{11}{16} c^2 d^2 n_k^2 \sum_{j=1}^{n_k} \mathcal{P}(\psi_j)$ in (5.3), which shows (iv). ∎

Before we can formulate Theorem 5.2, we have to introduce a few more concepts. Let $\mathcal{F}$ be a catalog that is approximable on sets $Q = (Q_f)_{f \in \mathcal{F}}$ with Lipschitz constants $L = (L_f)_{f \in \mathcal{F}} \subseteq [0, \infty)$. Then, for any catalog network $\xi = [(V_1, b_1, (f_{1,1}, \dots, f_{1,n_1})), \dots, (V_D, b_D, (f_{D,1}, \dots, f_{D,n_D}))] \in \mathcal{C}_{\mathcal{F}}$, we define

$$\begin{aligned} &\mathrm{Dom}_{Q,\xi} \\ &= \left\{ x \in \mathbb{R}^{\mathcal{I}_\xi} \colon \text{for all } k \in \{1, \dots, \mathcal{D}_\xi\} \colon \left(\mathcal{A}^{\xi,k} \circ \mathcal{G}^{\xi,k-1} \circ \dots \circ \mathcal{A}^{\xi,2} \circ \mathcal{G}^{\xi,1} \circ \mathcal{A}^{\xi,1}\right)(x) \in Q^{\xi,k} \right\} \end{aligned}$$

and

$$\mathrm{Lip}_{L,\xi} = \prod_{k=1}^{\mathcal{D}_\xi} L^{\xi,k} \|V_k\|,$$

where $\|\cdot\|$ denotes the operator norm when applied to matrices. The set $\mathrm{Dom}_{Q,\xi}$ describes where we will be able to approximate the catalog network ξ. It takes into account that each each layer function $\mathcal{G}^{\xi,k}$ can only be approximated on the set $Q^{\xi,k}$. The number $\mathrm{Lip}_{L,\xi}$ represents the worst-case Lipschitz constant of the catalog network.

To estimate the approximation error, we need two more quantities. The first one is

$$\mathcal{B}_\xi = \max\left\{1, \|\mathcal{A}^{\xi,1}(0)\|, \dots, \|\mathcal{A}^{\xi,\mathcal{D}_\xi}(0)\|\right\},$$

which simply measures the maximal norm of the inhomogeneous parts of the affine transformations (capped from below by 1). When using weight functions of the type $w_q(x) = (1 + x^q)^{-1}$, functions in the catalog are approximated better close to the origin. The quantity $\mathcal{B}_\xi$ together with the κ_0-boundedness of the catalog in the origin will be used to control how far away one is from the region where one has the best approximation. However, this becomes irrelevant for constant weight functions, as can be seen in Corollary 5.6 below.

The last quantity we need is $\mathcal{T}_{L,\xi}$, defined as the maximum of 1 and

$$\max_{k \in \{0, \dots, D-1\}} \max\{1, L^{\xi,k}, L^{\xi,D}\} \|V_D\| \prod_{j=k+1}^{D-1} L^{\xi,j} \|V_j\|$$

where we abbreviate $D = \mathcal{D}_\xi$ and use the convention $L^{\xi,0} = 0$. This combines the Lipschitz constants of the affine and nonlinear functions appearing in the different layers of the catalog network ξ.

Theorem 5.2. *Suppose $a \in C(\mathbb{R},\mathbb{R})$ fulfills the c-identity requirement for some number $c \geq 2$ and let w be a nonincreasing weight function whose decay is controlled by (s_1,s_2) for some $s_1 \in [1,\infty)$ and $s_2 \in [0,\infty)$. Consider a catalog network $\xi \in \mathcal{C}_{\mathcal{F}}$ for an $[a,w,Q,L,\varepsilon,\kappa]$-approximable catalog $\mathcal{F}$. Then there exists a skeleton $\phi \in \mathcal{N}$ with a-realization $\mathcal{R}_a^\phi \in C(\mathbb{R}^{\mathcal{I}_\xi},\mathbb{R}^{\mathcal{O}_\xi})$ such that*

(i) $\sup_{x\in \mathrm{Dom}_{Q,\xi}} w(\|x\|)\|\mathcal{R}^\xi(x) - \mathcal{R}_a^\phi(x)\| \leq \varepsilon$,

(ii) $\mathcal{R}_a^\phi$ *is* $\mathrm{Lip}_{L,\xi}$*-Lipschitz continuous on* $\mathbb{R}^{\mathcal{I}_\xi}$ *and*

(iii) $\mathcal{P}(\phi) \leq C\mathcal{B}_\xi^{t-\kappa_3}\mathcal{T}_{L,\xi}^t\mathcal{D}_\xi^{t+1}\mathcal{W}_\xi^{\kappa_2+t/2+6}\varepsilon^{-\kappa_3}$ *for* $t = \kappa_3(s_2+1)$ *and* $C = \frac{81}{32}c^3(4\kappa_0)^{t-\kappa_3}\kappa_1 s_1{}^{\kappa_3}$.

Proof. We split the proof into two parts. In the first part, we construct an approximating neural network and bound the approximation error. In the second part, we estimate the number of parameters of the network. Assume $\xi \in \mathcal{C}_{\mathcal{F}}^{l_0,\ldots,l_{2D}}$ is of the form $\xi = [(V_1,b_1,(f_{1,1},\ldots,f_{1,n_1})),\ldots,(V_D,b_D,(f_{D,1},\ldots,f_{D,n_D}))]$. Denote $G_0 = \mathrm{id}_{\mathbb{R}^{l_0}}$ and

$$G_k = \mathcal{G}^{\xi,k}\circ\mathcal{A}^{\xi,k}\circ\mathcal{G}^{\xi,k-1}\circ\cdots\circ\mathcal{G}^{\xi,1}\circ\mathcal{A}^{\xi,1}$$

for $k \in \{1,\ldots,D\}$. Before constructing the approximating network, we show by induction over k that

$$\begin{aligned}\|(\mathcal{A}^{\xi,k}\circ G_{k-1})(x)\| \leq{}& \|V_k\|\Big(\prod_{j=1}^{k-1}L^{\xi,j}\|V_j\|\Big)\|x\| + \|b_k\| \\ &+ \sum_{i=1}^{k-1}\|V_k\|\Big(\prod_{j=i+1}^{k-1}L^{\xi,j}\|V_j\|\Big)\big(L^{\xi,i}\|b_i\| + \|\mathcal{G}^{\xi,i}(0)\|\big)\end{aligned} \tag{5.4}$$

for all $k \in \{1,\ldots,D\}$ and $x \in \mathrm{Dom}_{Q,\xi}$. The base case $k=1$ reduces to the obvious inequality $\|\mathcal{A}^{\xi,1}(x)\| \leq \|V_1\|\|x\| + \|b_1\|$. For the induction step, suppose the claim is true for a given $k \in \{1,\ldots,D-1\}$. Then we obtain from Lemma 3.5 that

$$\begin{aligned}\|(\mathcal{A}^{\xi,k+1}\circ G_k)(x)\| &\leq \|V_{k+1}\|\|(\mathcal{G}^{\xi,k}\circ\mathcal{A}^{\xi,k}\circ G_{k-1})(x)\| + \|b_{k+1}\| \\ &\leq \|V_{k+1}\|L^{\xi,k}\|(\mathcal{A}^{\xi,k}\circ G_{k-1})(x)\| + \|V_{k+1}\|\|\mathcal{G}^{\xi,k}(0)\| + \|b_{k+1}\|.\end{aligned}$$

for all $x \in \mathrm{Dom}_{Q,\xi}$, where we used that the sets Q_f contain 0. Applying the induction hypothesis to $\|(\mathcal{A}^{\xi,k}\circ G_{k-1})(x)\|$, we obtain that (5.4) holds for $k+1$. Next, observe that

$$\|\mathcal{G}^{\xi,k}(0)\|^2 = \sum_{j=1}^{n_k}\|f_{k,j}(0)\|^2 \leq \kappa_0^2 n_k \leq \kappa_0^2\mathcal{W}_\xi \tag{5.5}$$

for all $k \in \{1,\ldots,D\}$. Hence, (5.4) yields

$$\begin{aligned}\|(\mathcal{A}^{\xi,D}\circ G_{D-1})(x)\| &\leq \mathcal{T}_{L,\xi}\|x\| + \mathcal{B}_\xi + D\mathcal{T}_{L,\xi}\big(\mathcal{B}_\xi + \kappa_0\sqrt{\mathcal{W}_\xi}\big) \\ &\leq 4\kappa_0\mathcal{B}_\xi\mathcal{D}_\xi\sqrt{\mathcal{W}_\xi}\mathcal{T}_{L,\xi}\max\{1,\|x\|\}\end{aligned} \tag{5.6}$$

for all $x \in \mathrm{Dom}_{Q,\xi}$.

To construct an approximating network, we note that it follows from Lemma 5.1 that for all $\delta \in (0,\varepsilon]$ and $k \in \{1,\ldots,D\}$, there exists $\psi_{\delta,k} \in \mathcal{N}$ such that the a-realization $\mathcal{R}_a^{\psi_{\delta,k}} \in C(\mathbb{R}^{l_{2k-1}}\mathbb{R}^{l_{2k}})$ satisfies

(i) $\|\mathcal{R}_a^{\psi_{\delta,k}}(x) - \mathcal{R}_a^{\psi_{\delta,k}}(y)\| \leq L^{\xi,k}\|x-y\|$ for all $x, y \in \mathbb{R}^{l_{2k-1}}$,

(ii) $w(\|x\|)\|\mathcal{G}^{\xi,k}(x) - \mathcal{R}_a^{\psi_{\delta,k}}(x)\| \leq \delta$ for all $x \in Q^{\xi,k}$ and

(iii) $\mathcal{P}(\psi_{\delta,k}) \leq \frac{11}{16}\kappa_1 c^2 \mathcal{W}_\xi^{\kappa_2+\frac{\kappa_3}{2}+5}\delta^{-\kappa_3}$.

Moreover, since each $\mathcal{A}^{\xi,k}$ is an affine function, there exist unique $\chi_k \in \mathcal{N}$ of depth 1, such that $\mathcal{R}_a^{\chi_k} = \mathcal{A}^{\xi,k}$ for all $k \in \{1,\dots,D\}$. Let $\varphi_{\delta,k} \in \mathcal{N}$ be given by $\varphi_{\delta,k} = \psi_{\delta,k} \circ \chi_k \circ \cdots \circ \psi_{\delta,1} \circ \chi_1$. The a-realization of $\varphi_{\delta,D}$ will be our approximation network. To verify that it does the job in terms of the approximation precision, we show that

$$\|G_k(x) - \mathcal{R}_a^{\varphi_{\delta,k}}(x)\| \leq \sum_{i=1}^{k} \frac{\delta \prod_{j=i+1}^{k} L^{\xi,j}\|V_j\|}{w(\|\mathcal{A}^{\xi,i} \circ G_{i-1}(x)\|)} \tag{5.7}$$

for all $k \in \{1,\dots,D\}$, $\delta \in (0,\varepsilon]$ and $x \in \mathrm{Dom}_{Q,\xi}$ by induction over k. The base case $k=1$ holds by the approximation property of $\psi_{\delta,1}$ and the fact that $\mathcal{A}^{\xi,1}(x) \in Q^{\xi,1}$ for all $x \in \mathrm{Dom}_{Q,\xi}$. For the induction step, we assume (5.7) holds for a given $k \in \{1,\dots,D-1\}$. By the Lipschitz and approximation properties of $\psi_{\delta,k+1}$, we obtain

$$\|G_{k+1}(x) - \mathcal{R}_a^{\varphi_{\delta,k+1}}(x)\| \leq \delta[w(\|(\mathcal{A}^{\xi,k+1} \circ G_k)(x)\|)]^{-1} + L^{\xi,k+1}\|V_{k+1}\|\|G_k(x) - \mathcal{R}_a^{\varphi_{\delta,k}}(x)\|$$

for all $\delta \in (0,\varepsilon]$ and $x \in \mathrm{Dom}_{Q,\xi}$, where we used that $(\mathcal{A}^{\xi,k+1} \circ G_k)(x) \in Q^{\xi,k+1}$ for all $x \in \mathrm{Dom}_{Q,\xi}$. Using the induction hypothesis on $\|G_k(x) - \mathcal{R}_a^{\varphi_{\delta,k}}(x)\|$, we obtain that (5.7) holds for $k+1$. Now, we combine (5.6), (5.7) and the assumption that the decay of w is controlled by (s_1, s_2) to find that

$$\begin{aligned} \|G_D(x) - \mathcal{R}_a^{\varphi_{\delta,D}}(x)\| &\leq \delta D \mathcal{T}_{L,\xi}\big[w\big(4\kappa_0 \mathcal{B}_\xi \mathcal{D}_\xi \sqrt{\mathcal{W}_\xi}\mathcal{T}_{L,\xi}\max\{1,\|x\|\}\big)\big]^{-1} \\ &\leq s_1 \delta [4\kappa_0 \mathcal{B}_\xi]^{s_2}[\mathcal{D}_\xi \mathcal{T}_{L,\xi}]^{s_2+1}\mathcal{W}_\xi^{s_2/2}[w(\|x\|)]^{-1} \end{aligned} \tag{5.8}$$

for all $\delta \in (0,\varepsilon]$ and $x \in \mathrm{Dom}_{Q,\xi}$.

Next note that it follows from the fact that the concatenation of Lipschitz functions is again Lipschitz with constant equal to the product of the original Lipschitz constants that $\mathcal{R}_a^{\varphi_{\delta,D}}$ is $\mathrm{Lip}_{L,\xi}$-Lipschitz.

It remains to estimate the number of parameters of the constructed network. To do this, we slightly modify $\varphi_{\delta,D}$ by interposing identity networks. This does not change the realization but reduces the worst-case parameter count, as discussed before Corollary 2.6. Choose $I \in \mathcal{N}$ for which a fulfills the c-identity requirement and let $I_d = p(I,\dots,I)$, $d \in \mathbb{N}$. Define $\rho_{\delta,k} = I_{l_{2k}} \circ \psi_{\delta,k} \circ I_{l_{2k-1}}$ for $k \in \{1,\dots,D\}$ and $\delta \in (0,\varepsilon]$. Combining Corollary 2.6 with our parameter bound for $\mathcal{P}(\psi_{\delta,k})$, we obtain

$$\begin{aligned} \mathcal{P}(\rho_{\delta,k}) &\leq \tfrac{55}{96}\kappa_1 c^3 \mathcal{W}_\xi^{\kappa_2+\kappa_3/2+6}\delta^{-\kappa_3} + \tfrac{29}{12}c^2\mathcal{W}_\xi^2 \\ &\leq \tfrac{57}{32}\kappa_1 c^3 \mathcal{W}_\xi^{\kappa_2+\kappa_3/2+6}\delta^{-\kappa_3} \end{aligned} \tag{5.9}$$

for all $\delta \in (0,\varepsilon]$ and $k \in \{1,\dots,D\}$, where we used that $c \geq 2$. Since $l_1^{I_d} \leq cd$ for all $d \in \mathbb{N}$, Proposition 2.1 yields that

$$\mathcal{P}(\rho_{\delta,k} \circ \chi_k) \leq \mathcal{P}(\rho_{\delta,k}) + c\mathcal{W}_\xi^2 \tag{5.10}$$

for all $\delta \in (0,\varepsilon]$ and $k \in \{1,\dots,D\}$. Next, we show that

$$\mathcal{P}(\rho_{\delta,k} \circ \chi_k \circ \cdots \circ \rho_{\delta,1} \circ \chi_1) \leq (k-1)c^2\mathcal{W}_\xi^2 + \sum_{j=1}^{k} \mathcal{P}(\rho_{\delta,j} \circ \chi_j) \tag{5.11}$$

for all $k \in \{1, \dots, D\}$ and $\delta \in (0, \varepsilon]$ by induction over k. The base case $k = 1$ is trivially satisfied. For the induction step, we assume (5.11) holds for $k \in \{1, \dots, D-1\}$. Then Proposition 2.1 and the induction hypothesis show that

$$\begin{aligned}
&\mathcal{P}(\rho_{\delta,k+1} \circ \chi_{k+1} \circ \cdots \circ \rho_{\delta,1} \circ \chi_1) \\
&\leq \mathcal{P}(\rho_{\delta,k+1} \circ \chi_{k+1}) + \mathcal{P}(\rho_{\delta,k} \circ \chi_k \circ \cdots \circ \rho_{\delta,1} \circ \chi_1) + c^2 l_{2(k+1)-1} l_{2k} \\
&\leq kc^2 \mathcal{W}_\xi^2 + \sum_{j=1}^{k+1} \mathcal{P}(\rho_{\delta,j} \circ \chi_j)
\end{aligned}$$

for all $\delta \in (0, \varepsilon]$, which completes the induction. Now we combine (5.10) and (5.11) to obtain

$$\mathcal{P}(\rho_{\delta,D} \circ \chi_D \circ \cdots \circ \rho_{\delta,1} \circ \chi_1) \leq \tfrac{3}{2} c^2 D \mathcal{W}_\xi^2 + \sum_{j=1}^{D} \mathcal{P}(\rho_{\delta,j})$$

for all $\delta \in (0, \varepsilon]$, where we used $c \geq 2$ again. Using (5.9) gives

$$\begin{aligned}
\mathcal{P}(\rho_{\delta,D} \circ \chi_D \circ \cdots \circ \rho_{\delta,1} \circ \chi_1) &\leq \tfrac{3}{2} c^2 D \mathcal{W}_\xi^2 + \tfrac{57}{32} D \kappa_1 c^3 \mathcal{W}_\xi^{\kappa_2+\kappa_3/2+6} \delta^{-\kappa_3} \\
&\leq \tfrac{81}{32} \kappa_1 c^3 \mathcal{D}_\xi \mathcal{W}_\xi^{\kappa_2+\kappa_3/2+6} \delta^{-\kappa_3}
\end{aligned} \tag{5.12}$$

for all $\delta \in (0, \varepsilon]$. We conclude by summarizing what we have proved so far. Motivated by (5.8), let $\eta \in (0, \varepsilon]$ be given by

$$\eta = \varepsilon \big(s_1 [4\kappa_0 \mathcal{B}_\xi]^{s_2} [\mathcal{D}_\xi \mathcal{T}_{L,\xi}]^{s_2+1} \mathcal{W}_\xi^{s_2/2} \big)^{-1},$$

and define $\phi = \rho_{\eta,D} \circ \chi_D \circ \cdots \circ \rho_{\eta,1} \circ \chi_1$. Then, $\mathcal{R}_a^\phi = \mathcal{R}_a^{\varphi_{\eta,D}}$ and $G_D = \mathcal{R}^\xi$. So, one obtains from (5.8) that

$$\sup_{x \in \mathrm{Dom}_{Q,\xi}} w(\|x\|) \|\mathcal{R}^\xi(x) - \mathcal{R}_a^\phi(x)\| \leq \varepsilon,$$

and (5.12) gives

$$\mathcal{P}(\phi) \leq \tfrac{81}{32} c^3 \kappa_1 {s_1}^{\kappa_3} [4\kappa_0 \mathcal{B}_\xi]^{\kappa_3 s_2} \mathcal{T}_{L,\xi}^{\kappa_3(s_2+1)} \mathcal{D}_\xi^{\kappa_3(s_2+1)+1} \mathcal{W}_\xi^{\kappa_2+\frac{\kappa_3}{2}(s_2+1)+6} \varepsilon^{-\kappa_3},$$

which completes the proof. ∎

The conclusion of Theorem 5.2 could be written more concisely as

$$\mathrm{Cost}_{a,w}(\mathcal{R}^\xi, \mathrm{Dom}_{Q,\xi}, \mathrm{Lip}_{L,\xi}, \varepsilon) \leq C \mathcal{B}_\xi^{t-\kappa_3} \mathcal{T}_{L,\xi}^t \mathcal{D}_\xi^{t+1} \mathcal{W}_\xi^{\kappa_2+t/2+6} \varepsilon^{-\kappa_3}.$$

We point out that the rate in the accuracy is $\mathcal{O}(\varepsilon^{-\kappa_3})$, the same as for the underlying catalog $\mathcal{F}$.

In the proof of Theorem 5.2, we combine the approximations of the functions $\mathcal{G}^{\xi,k}$ obtained in Lemma 5.1 with the affine maps $\mathcal{A}^{\xi,k}$. When concatenating different approximating networks, we interpose identity networks. This reduces the parameter count in worst-case scenarios but can lead to slightly looser estimates in certain other situations.

Remark 5.3. If $\mathcal{I}(f) \leq d$ and $\mathcal{O}(f) \leq d$ for some $d \in \mathbb{N}$ and all $f \in \mathcal{F}$, we can use (iv) instead of (iii) of Lemma 5.1 in the proof of Theorem 5.2 to obtain the following modified version of the parameter bound in Theorem 5.2:

$$\mathcal{P}(\phi) \leq C d^2 \mathcal{B}_\xi^{t-\kappa_3} \mathcal{T}_{L,\xi}^t \mathcal{D}_\xi^{t+1} \mathcal{W}_\xi^{\kappa_2+t/2+4} \varepsilon^{-\kappa_3}.$$

Since in many of the example catalogs of Section 4, the maximal input/output dimension is 1 or 2, this will allow us to obtain better estimates in some of the applications in Section 7 below.

Remark 5.4. A careful inspection of the proof of Theorem 5.2 shows that it does not only work for the Euclidean norm but also, for instance, the sup-norm.

Proof. To see that the proof of Theorem 5.2 also works for the sup-norm, we note that in (3.1), (5.1), (5.2) and (5.5), we used the property $\|x\|^2 = \sum_{j=1}^{n_k} \|x_{(k,j)}\|^2$ of the Euclidean norm. However, since the sup-norm satisfies $\|x\|_\infty^2 = \max_{j\in\{1,\dots,n_k\}} \|x_{(k,j)}\|_\infty^2$, and we did not use any specific property of the Euclidean norm anywhere else in the proof, Theorem 5.2 still holds for the sup-norm. ∎

We know that the ReLU activation function satisfies the 2-identity requirement. Theorem 5.2 recast for ReLU activation and the weight function $w_q(x) = (1+x^q)^{-1}$ reads as follows:

Corollary 5.5. *Consider the weight function $w_q(x) = (1+x^q)^{-1}$ for some $q \in (0,\infty)$. Let $\xi \in \mathcal{C}_\mathcal{F}$ be a catalog network for a $[\rho, w_q, Q, L, \varepsilon, \kappa]$-approximable catalog $\mathcal{F}$. Then there exists a skeleton $\phi \in \mathcal{N}$ with ReLU-realization $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^{\mathcal{I}_\xi}, \mathbb{R}^{\mathcal{O}_\xi})$ such that*

(i) $\sup_{x\in \mathrm{Dom}_{Q,\xi}} (1+\|x\|^q)^{-1} \|\mathcal{R}^\xi(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)\| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is* $\mathrm{Lip}_{L,\xi}$*-Lipschitz continuous on* $\mathbb{R}^{\mathcal{I}_\xi}$ *and*

(iii) $\mathcal{P}(\phi) \leq C\mathcal{B}_\xi^{t-\kappa_3}\mathcal{T}_{L,\xi}^t \mathcal{D}_\xi^{t+1}\mathcal{W}_\xi^{\kappa_2+t/2+6}\varepsilon^{-\kappa_3}$ *for* $t = \kappa_3(q+1)$ *and* $C = \frac{81}{4}2^{2t-\kappa_3}\kappa_0^{t-\kappa_3}\kappa_1$.

For the weight function $w \equiv 1$, the parameter estimate in Theorem 5.2 simplifies considerably. This is because the decay of $w \equiv 1$ is controlled by $(1,0)$, which makes the translation size and the bound of the catalog in the origin irrelevant.

Corollary 5.6. *Let $\xi \in \mathcal{C}_\mathcal{F}$ be a catalog network for a $[\rho, 1, Q, L, \varepsilon, \kappa]$-approximable catalog $\mathcal{F}$. Then there exists a skeleton $\phi \in \mathcal{N}$ with ReLU-realization $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^{\mathcal{I}_\xi}, \mathbb{R}^{\mathcal{O}_\xi})$ such that*

(i) $\sup_{x\in \mathrm{Dom}_{Q,\xi}} \|\mathcal{R}^\xi(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)\| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is* $\mathrm{Lip}_{L,\xi}$*-Lipschitz continuous on* $\mathbb{R}^{\mathcal{I}_\xi}$ *and*

(iii) $\mathcal{P}(\phi) \leq \frac{81}{4}\kappa_1 \mathcal{T}_{L,\xi}^{\kappa_3}\mathcal{D}_\xi^{\kappa_3+1}\mathcal{W}_\xi^{\kappa_2+\kappa_3/2+6}\varepsilon^{-\kappa_3}$.

6. Log-approximable catalogs

In this section, we modify the way we measure the approximation cost and derive corresponding approximation results.

Definition 6.1. With the setup of Definition 3.4 and $\varepsilon \leq 1/2$, $\mathcal{F} \subseteq \bigcup_{m,n\in\mathbb{N}} C(\mathbb{R}^m, \mathbb{R}^n)$ is called $[a, w, Q, L, \varepsilon, \kappa]$-log-approximable if $\sup_{f\in\mathcal{F}} \|f(0)\| \leq \kappa_0$ and

$$\mathrm{Cost}_{a,w}(f, Q_f, L_f, \delta) \leq \kappa_1 \max\{\mathcal{I}(f), \mathcal{O}(f)\}^{\kappa_2} [\log_2(\delta^{-1})]^{\kappa_3}$$

for all $f \in \mathcal{F}$ and $\delta \in (0, \varepsilon]$.

This log-modification is designed for catalogs made of functions like the square or the product, which can be approximated with rate $\mathcal{O}(\log_2(\varepsilon^{-1}))$, as we have seen in Lemma 4.3 and Proposition 4.5. Its usefulness will become apparent in Proposition 7.6, which is based on the following catalog.

Example 6.2. Let $\mathcal{F}^{\text{prod}} = \{\text{id}_{\mathbb{R}}, \text{pr}\}$ be the product catalog and fix $r \in [1, \infty)$ and $d \in \mathbb{N}$. Consider the approximation sets $Q_{\text{id}_{\mathbb{R}}} = \mathbb{R}$, $Q_{\text{pr}} = [-r^d, r^d]^2$ and the Lipschitz constants $L_{\text{id}_{\mathbb{R}}} = 1$, $L_{\text{pr}} = \sqrt{8}r^d$. Then $\mathcal{F}^{\text{prod}}$ is a $[\rho, 1, Q, L, \min\{1/2, 1/r\}, (1, 320d + 208, 0, 1)]$-log-approximable catalog, where κ_1 can also be chosen as 208 instead of $320d + 208$ if $r = 1$.

Using the notion of log-approximable catalogs, we can derive the following analogue of Theorem 5.2.

Theorem 6.3. *Assume $a \in C(\mathbb{R}, \mathbb{R})$ satisfies the c-identity requirement for some number $c \geq 2$, and let w be a nonincreasing weight function whose decay is controlled by (s_1, s_2) for some $s_1 \in [1, \infty)$ and $s_2 \in [0, \infty)$. Consider a catalog network $\xi \in \mathcal{C}_{\mathcal{F}}$ for an $[a, w, Q, L, \varepsilon, \kappa]$-log-approximable catalog $\mathcal{F}$. Then there exists a skeleton $\phi \in \mathcal{N}$ with a-realization $\mathcal{R}_a^{\phi} \in C(\mathbb{R}^{\mathcal{I}_\xi}, \mathbb{R}^{\mathcal{O}_\xi})$ such that*

(i) $\sup_{x \in \text{Dom}_{Q,\xi}} w(\|x\|)\|\mathcal{R}^{\xi}(x) - \mathcal{R}_a^{\phi}(x)\| \leq \varepsilon$,

(ii) $\mathcal{R}_a^{\phi}$ *is* $\text{Lip}_{L,\xi}$*-Lipschitz continuous on* $\mathbb{R}^{\mathcal{I}_\xi}$ *and*

(iii) $\mathcal{P}(\phi) \leq \frac{81}{32} c^3 \kappa_1 \mathcal{D}_\xi \mathcal{W}_\xi^{\kappa_2+6} \big[\log_2\big(s_1 [4\kappa_0 \mathcal{B}_\xi]^{s_2} [\mathcal{T}_{L,\xi} \mathcal{D}_\xi \sqrt{\mathcal{W}_\xi}]^{s_2+1} \varepsilon^{-1}\big)\big]^{\kappa_3}$.

Proof. In case the catalog $\mathcal{F}$ in Lemma 5.1 is $[a, w, Q, L, \varepsilon, \kappa]$-log-approximable instead of $[a, w, Q, L, \varepsilon, \kappa]$-approximable, a slight modification of the proof yields a version of Lemma 5.1, for which the parameter bound in (iii) is

$$\tfrac{11}{16} \kappa_1 c^2 \max\{l_{2k-1}, l_{2k}\}^{\kappa_2+5} \Big[\log_2\Big(\frac{\sqrt{\max\{l_{2k-1}, l_{2k}\}}}{\delta}\Big)\Big]^{\kappa_3}.$$

Then, in the proof of Theorem 5.2, one only needs to replace (iii) with

$$\mathcal{P}(\psi_{\delta,k}) \leq \tfrac{11}{16} \kappa_1 c^2 \mathcal{W}_\xi^{\kappa_2+5} \big[\log_2\big(\sqrt{\mathcal{W}_\xi} \delta^{-1}\big)\big]^{\kappa_3},$$

(5.9) with

$$\mathcal{P}(\rho_{\delta,k}) \leq \tfrac{57}{32} \kappa_1 c^3 \mathcal{W}_\xi^{\kappa_2+6} \big[\log_2\big(\sqrt{\mathcal{W}_\xi} \delta^{-1}\big)\big]^{\kappa_3}$$

and (5.12) with

$$\mathcal{P}(\rho_{\delta,D} \circ \chi_D \circ \cdots \circ \rho_{\delta,1} \circ \chi_1) \leq \tfrac{81}{32} \kappa_1 c^3 \mathcal{D}_\xi \mathcal{W}_\xi^{\kappa_2+6} \big[\log_2\big(\sqrt{\mathcal{W}_\xi} \delta^{-1}\big)\big]^{\kappa_3}.$$

The rest of the proof of Theorem 5.2 carries over without any changes. ∎

The following is the analogue of Corollary 5.6 for log-approximable catalogs.

Corollary 6.4. *Let $\xi \in \mathcal{C}_{\mathcal{F}}$ be a catalog network for a $[\rho, 1, Q, L, \varepsilon, \kappa]$-log-approximable catalog $\mathcal{F}$. Then there exists a skeleton $\phi \in \mathcal{N}$ with ReLU-realization $\mathcal{R}_{\text{ReLU}}^{\phi} \in C(\mathbb{R}^{\mathcal{I}_\xi}, \mathbb{R}^{\mathcal{O}_\xi})$ such that*

(i) $\sup_{x \in \text{Dom}_{Q,\xi}} \|\mathcal{R}^{\xi}(x) - \mathcal{R}_{\text{ReLU}}^{\phi}(x)\| \leq \varepsilon$,

(ii) $\mathcal{R}_{\text{ReLU}}^{\phi}$ *is* $\text{Lip}_{L,\xi}$*-Lipschitz continuous on* $\mathbb{R}^{\mathcal{I}_\xi}$ *and*

(iii) $\mathcal{P}(\phi) \leq \frac{81}{4} \kappa_1 \mathcal{D}_\xi \mathcal{W}_\xi^{\kappa_2+6} \big[\log_2\big(\mathcal{T}_{L,\xi} \mathcal{D}_\xi \sqrt{\mathcal{W}_\xi} \varepsilon^{-1}\big)\big]^{\kappa_3}$.

7. Overcoming the curse of dimensionality

Next, we apply the theory of catalog networks to show that different high-dimensional functions admit a ReLU neural network approximation without the curse of dimensionality. We use the catalogs introduced in Sections 4 and 6 to construct families of functions indexed by the dimension of their domain that are of the same form for each dimension. The results in this section are proved by finding catalog network representations of the high-dimensional target functions, so that one of the general approximation results of Sections 5 and 6 can be applied. The mere approximability of these functions with ReLU networks follows from classical universal approximation results such as [86]. But the quantitative estimates on the number of parameters in terms of the dimension d and the accuracy ε are new.

Our general results cover a wide range of interesting examples, but it is possible that, for some of them, the estimates could be improved by using their special structure.

Proposition 7.1. *Fix $K, r \in [1,\infty)$, and let $f_i\colon \mathbb{R} \to \mathbb{R}$, $i \in \mathbb{N}$, be K-Lipschitz continuous with $|f_i(0)| \leq K$. Define $g_d\colon \mathbb{R}^d \to \mathbb{R}$ by $g_d(x) = \sum_{i=1}^d f_i(x_i)$. Then for all $d \in \mathbb{N}$ and $\varepsilon \in (0,1]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^d, \mathbb{R})$ such that*

(i) $\sup_{x\in[-r,r]^d} |g_d(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is $\sqrt{d}K$-Lipschitz continuous on $\mathbb{R}^d$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{4}{7} 10^3 K^2 r d^5 \varepsilon^{-1}$.

Proof. Let $\mathcal{F} = \mathcal{F}_K^{\mathrm{Lip}}$ be the K-Lipschitz catalog and suppose $Q = (Q_f)_{f\in\mathcal{F}}$ and $L = (L_f)_{f\in\mathcal{F}}$ are defined as in (i) of Example 4.1. For $d \in \mathbb{N}$, let $V_d \in \mathbb{R}^{1\times d}$ be the matrix $V_d = (1,\dots,1)$ and $\xi_d \in \mathcal{C}_\mathcal{F}$ the catalog network $\xi_d = [(\mathrm{id}_{\mathbb{R}^d}, 0, (f_1,\dots,f_d)), (V_d, 0, \mathrm{id}_\mathbb{R})]$. Then $\mathcal{R}^{\xi_d} = g_d$, $\mathcal{D}_{\xi_d} = 2$, $\mathcal{W}_{\xi_d} = d$ and $\mathcal{T}_{L,\xi_d} \leq \sqrt{d}K$. Moreover, $\mathrm{Dom}_{Q,\xi_d} = Q^{\xi_d,1} \supseteq [-r,r]^d$ because $Q^{\xi_d,2} = Q_{\mathrm{id}_\mathbb{R}} = \mathbb{R}$. Now, the proposition follows from Remark 5.3 and Corollary 5.6. ∎

The following is a generalization, whose proof only requires a slight adjustment.

Proposition 7.2. *Fix $K, r \in [1,\infty)$, and let $f_i\colon \mathbb{R} \to \mathbb{R}$, $i \in \mathbb{N}_0$, be K-Lipschitz continuous with $|f_i(0)| \leq K$. Define $g_d\colon \mathbb{R}^d \to \mathbb{R}$ by $g_d(x) = f_0(\sum_{i=1}^d f_i(x_i))$. Then for all $d \in \mathbb{N}$ and $\varepsilon \in (0,1]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^d, \mathbb{R})$ such that*

(i) $\sup_{x\in[-r,r]^d} |g_d(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is $\sqrt{d}K^2$-Lipschitz continuous on $\mathbb{R}^d$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{5}{6} 10^3 K^4 r d^6 \varepsilon^{-1}$.

Proof. This proposition is proved as Proposition 7.1, except that we fix d in the beginning, use (i) of Example 4.1 with $dK(r+1)$ instead of r and define

$$\xi_d = [(\mathrm{id}_{\mathbb{R}^d}, 0, (f_1,\dots,f_d)), (V_d, 0, f_0)].$$

Then, $\mathcal{T}_{L,\xi_d} \leq \sqrt{d}K^2$ and $Q^{\xi_d,2} = [-dK(r+1), dK(r+1)]$, which ensures that $\mathrm{Dom}_{Q,\xi_d} \supseteq [-r,r]^d$. ∎

Note that the parameter estimates in Propositions 7.1 and 7.2 depend on r since we approximate uniformly on the hypercube $[-r,r]^d$. However, the estimate is only linear in r, even though the volume of the hypercube is of the order r^d.

Proposition 7.3. *Let $K, r \in [1,\infty)$ and suppose $f_i\colon \mathbb{R} \to \mathbb{R}$ is K-Lipschitz with $|f_i(0)| \leq K$ and $g_i\colon \mathbb{R} \to \mathbb{R}$ 1-Lipschitz with $g_i(0) = 0$, $i \in \mathbb{N}$. Let $h_d\colon \mathbb{R}^d \to \mathbb{R}$, $d \in \mathbb{N}$, be given by $h_1 = f_1$ and $h_d(x) = g_d(\max\{h_{d-1}(x_1,\dots,x_{d-1}), f_d(x_d)\})$ for $d \geq 2$. Then for all $d \in \mathbb{N}$ and $\varepsilon \in (0,1]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^d, \mathbb{R})$ such that*

(i) $\sup_{x\in[-r,r]^d} |h_d(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is K-Lipschitz continuous on $\mathbb{R}^d$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{3}{7} 10^4 K^3 r d^{13/2} \varepsilon^{-1}$.

Proof. Let $\mathcal{F} = \mathcal{F}_K^{\mathrm{Lip,max}}$ be the K-Lipschitz-maximum catalog and suppose $Q = (Q_f)_{f\in\mathcal{F}}$ and $L = (L_f)_{f\in\mathcal{F}}$ are defined as in (i) of Example 4.2 (with $R = K(r+1)$ instead of r). Since we know that the functions g_i are 1-Lipschitz, we may actually set $L_{g_i} = 1$ for all $i \in \mathbb{N}$ without affecting the approximability of the catalog. Consider $\xi_d \in \mathcal{C}_\mathcal{F}$, $d \in \mathbb{N}_{\geq 2}$, given by

$$\begin{aligned}\xi_d = \big[&\big(\mathrm{id}_{\mathbb{R}^d}, 0, (f_1,\dots,f_d)\big), \big(\mathrm{id}_{\mathbb{R}^d}, 0, (\max_2, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big), \big(\mathrm{id}_{\mathbb{R}^{d-1}}, 0, (g_2, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big),\\ &\big(\mathrm{id}_{\mathbb{R}^{d-1}}, 0, (\max_2, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big), \dots, \big(\mathrm{id}_{\mathbb{R}^2}, 0, (g_{d-1}, \mathrm{id}_\mathbb{R})\big), \big(\mathrm{id}_{\mathbb{R}^2}, 0, \max_2\big), \big(\mathrm{id}_\mathbb{R}, 0, g_d\big)\big].\end{aligned}$$

Then $\mathcal{R}^{\xi_d} = h_d$, $\mathcal{D}_{\xi_d} = 2d-1$, $\mathcal{W}_{\xi_d} = d$ and $\mathcal{T}_{L,\xi_d} \leq K$. Moreover, $Q^{\xi_d,2i} = \mathbb{R}^{d-i+1}$ and $Q^{\xi_d,2i+1} = [-R,R] \times \mathbb{R}^{d-i-1}$ for all $i \in \{1,\dots,d-1\}$ as well as $Q^{\xi_d,1} = [-R,R]^d$. If we define $H_i\colon \mathbb{R}^i \to \mathbb{R}$, $i \in \mathbb{N}_{\geq 2}$, by $H_i(x) = \max\{h_{i-1}(x_1,\dots,x_{i-1}), f_i(x_i)\}$, then it follows by induction that H_i is K-Lipschitz with respect to the sup-norm and $|H_i(0)| \leq K$. This proves $(H_i(x_1,\dots,x_i), x_{i+1},\dots,x_d) \in Q^{\xi_d,2i-1}$ for all $x \in [-r,r]^d$ and $i \in \{2,\dots,d\}$. Since this corresponds to the evaluation of the first $2(i-1)$ layers of ξ, we have shown $[-r,r]^d \subseteq \mathrm{Dom}_{Q,\xi_d}$. Now we conclude with[6] Remark 5.3 and Corollary 5.6. ∎

Note that if the functions g_i, for $1 \leq i \leq d-1$, are chosen to be the identity and $g_d = f_0$, then h_d reduces to $f_0(\max\{f_1(x_1),\dots,f_d(x_d)\})$.

The functions in the previous propositions were approximated on bounded domains, but if one is willing to pay a higher approximation cost, one can also approximate the family of functions from, e.g., Proposition 7.1 on the entire space without the curse of dimensionality for an appropriate weight function.

Proposition 7.4. *Let $q \in (1,\infty)$, $K \in [1,\infty)$ and suppose $f_i\colon \mathbb{R} \to \mathbb{R}$, $i \in \mathbb{N}$, are K-Lipschitz continuous with $|f_i(0)| \leq K$. Define $g_d\colon \mathbb{R}^d \to \mathbb{R}$ by $g_d(x) = \sum_{i=1}^d f_i(x_i)$. Then for all $d \in \mathbb{N}$ and $\varepsilon \in (0,1]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^d, \mathbb{R})$ such that*

(i) $\sup_{x\in\mathbb{R}^d} (1+\|x\|^q)^{-1} |g_d(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is $\sqrt{d}K$-Lipschitz continuous on $\mathbb{R}^d$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{2}{9} 10^3 2^{3t(q+1)} K^{2t(q+1)} d^{t(q+1)+4} \varepsilon^{-t}$ *for $t = q/q-1$.*

Proof. Let $\mathcal{F} = \mathcal{F}_K^{\mathrm{Lip}}$ be the K-Lipschitz catalog and suppose $Q = (Q_f)_{f\in\mathcal{F}}$ and $L = (L_f)_{f\in\mathcal{F}}$ are defined as in (ii) of Example 4.1. For all $d \in \mathbb{N}$, let $V_d \in \mathbb{R}^{1\times d}$ be the matrix $V_d = (1,\dots,1)$ and $\xi_d \in \mathcal{C}_\mathcal{F}$ the catalog network $\xi_d = [(\mathrm{id}_{\mathbb{R}^d}, 0, (f_1,\dots,f_d)), (V_d, 0, \mathrm{id}_\mathbb{R})]$. Then $\mathcal{R}^{\xi_d} = g_d$, $\mathcal{B}_{\xi_d} = 1$, $\mathcal{D}_{\xi_d} = 2$, $\mathcal{W}_{\xi_d} = d$ and $\mathcal{T}_{L,\xi_d} \leq \sqrt{d}K$. Moreover, $\mathrm{Dom}_{Q,\xi_d} = Q^{\xi_d,1} = \mathbb{R}^d$ because $Q^{\xi_d,2} = Q_{\mathrm{id}_\mathbb{R}} = \mathbb{R}$. Now, the proposition follows from Remark 5.3 and Corollary 5.5. ∎

[6] Strictly speaking, Remark 5.3 is not applicable to the Lipschitz-maximum catalog, but we only used $\max_2$ and could remove $\max_d$, $d \geq 3$, from the catalog. This also enables us to use $\kappa = (K, 13K^2 r, 0, 1)$ instead of $\kappa_2 = 3$.

An analogous result with the maximum function instead of the sum can be shown similarly.

In the next two propositions, we replace the sum by a product. Then we cannot establish the approximation on an arbitrarily large domain since the Lipschitz constant of the product function on $[-r,r]^2$ grows linearly in r.

Proposition 7.5. *Let $K \in [1,\infty)$ and suppose $f_i\colon \mathbb{R} \to \mathbb{R}$ is K-Lipschitz and $g_i\colon \mathbb{R} \to \mathbb{R}$ 1-Lipschitz with $f_i(0) = 0 = g_i(0)$, $i \in \mathbb{N}$. Let $h_d\colon \mathbb{R}^d \to \mathbb{R}$, $d \in \mathbb{N}$, be given by $h_1 = f_1$ and $h_d(x) = g_d(h_{d-1}(x_1,\dots,x_{d-1})f_d(x_d))$ for $d \geq 2$. Set $r = 1/\sqrt{8}K$. Then for all $d \in \mathbb{N}$ and $\varepsilon \in (0, 1/16]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^d,\mathbb{R})$ such that*

(i) $\sup_{x\in[-r,r]^d} |h_d(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is K-Lipschitz continuous on $\mathbb{R}^d$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{3}{7}10^4 K^2 d^{13/2}\varepsilon^{-1}$.

Proof. Let $\mathcal{F} = \mathcal{F}_K^{\mathrm{Lip,prod}}$ be the K-Lipschitz-product catalog and suppose $Q = (Q_f)_{f\in\mathcal{F}}$ and $L = (L_f)_{f\in\mathcal{F}}$ are defined as in Example 4.6 (with $1/\sqrt{8}$ instead of r and $R = 1/\sqrt{8}$). Since we know that the functions g_i are 1-Lipschitz, we may actually set $L_{g_i} = 1$ for all $i \in \mathbb{N}$ without affecting the approximability of the catalog. Consider $\xi_d \in \mathcal{C}_\mathcal{F}$, $d \in \mathbb{N}_{\geq 2}$, given by

$$\begin{aligned}\xi_d = \big[&\big(\mathrm{id}_{\mathbb{R}^d}, 0, (f_1, f_2, \dots, f_d)\big), \big(\mathrm{id}_{\mathbb{R}^d}, 0, (\mathrm{pr}, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big), \big(\mathrm{id}_{\mathbb{R}^{d-1}}, 0, (g_2, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big),\\ &\big(\mathrm{id}_{\mathbb{R}^{d-1}}, 0, (\mathrm{pr}, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big), \dots, \big(\mathrm{id}_{\mathbb{R}^2}, 0, (g_{d-1}, \mathrm{id}_\mathbb{R})\big), \big(\mathrm{id}_{\mathbb{R}^2}, 0, \mathrm{pr}\big), \big(\mathrm{id}_\mathbb{R}, 0, g_d\big)\big].\end{aligned}$$

Then $\mathcal{R}^{\xi_d} = h_d$, $\mathcal{D}_{\xi_d} = 2d-1$, $\mathcal{W}_{\xi_d} = d$ and $\mathcal{T}_{L,\xi_d} \leq K$. Moreover, $Q^{\xi_d,2i} = [-R,R]^2 \times \mathbb{R}^{d-i-1}$ and $Q^{\xi_d,2i+1} = [-R,R] \times \mathbb{R}^{d-i-1}$ for all $i \in \{1,\dots,d-1\}$ as well as $Q^{\xi_d,1} = [-R,R]^d$. If we define $H_i\colon \mathbb{R}^i \to \mathbb{R}$, $i \in \mathbb{N}_{\geq 2}$, by $H_i(x) = h_{i-1}(x_1,\dots,x_{i-1})f_i(x_i)$, then $H_i(0) = 0$ and it follows by induction that $|H_i(x)| \leq R^i$ and $|h_i(x)| \leq R^i$ for all $x \in [-r,r]^d$ and $i \geq 2$. This shows that $[-r,r]^d \subseteq \mathrm{Dom}_{Q,\xi_d}$, and we can conclude with Remark 5.3 and Corollary 5.6. ∎

This example includes the special case $f_0(\prod_{i=1}^d f_i(x_i))$. Since on large hypercubes the quantity $\mathcal{T}_{L,\xi_d}$, where ξ_d is a catalog network representing the function h_d, starts to grow exponentially in the dimension, the approximators in the proof of Proposition 7.5 can only be built on the hypercube $[-1/\sqrt{8}K, 1/\sqrt{8}K]^d$.

However, it has been shown in Proposition 3.3. of [113] that the product $\prod_{i=1}^d x_i$ can be approximated without the curse of dimensionality on the hypercube $[-1,1]^d$. Applying the log-modification of our theory, we can recover this result and even allow for arbitrarily large hypercubes.

Proposition 7.6. *Consider the functions $f_d\colon \mathbb{R}^d \to \mathbb{R}$, $d \in \mathbb{N}$, given by $f_d(x) = \prod_{i=1}^d x_i$, and let $r \in [1,\infty)$. Then for all $d \in \mathbb{N}_{\geq 2}$ and $\varepsilon \in (0, \min\{1/2, r^{-1}\}]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\mathrm{ReLU}} \in C(\mathbb{R}^d,\mathbb{R})$ such that*

(i) $\sup_{x\in[-r,r]^d} |f_d(x) - \mathcal{R}^\phi_{\mathrm{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\mathrm{ReLU}}$ *is $8^{(d-1)/2}r^{d(d-1)}$-Lipschitz continuous on $\mathbb{R}^d$,*

(iii) $\mathcal{P}(\phi) \leq \frac{1}{3}10^5 \log_2(d) d^6 \log_2(\varepsilon^{-1})$ *if $r = 1$ and*

(iv) $\mathcal{P}(\phi) \leq \frac{2}{5}10^5 \log_2(r)\log_2(d) d^8 \log_2(\varepsilon^{-1})$ *if $r \geq 2$.*

Proof. Assume without loss of generality that $d \geq 3$, let $\mathcal{F} = \mathcal{F}^{\text{prod}}$ be the product catalog, and suppose $Q = (Q_f)_{f\in\mathcal{F}}$ and $L = (L_f)_{f\in\mathcal{F}}$ are defined as in Example 6.2. Moreover, let $\xi_d \in \mathcal{C}_\mathcal{F}$ be given by

$$\begin{aligned}\xi_d = \big[&\big(\mathrm{id}_{\mathbb{R}^d}, 0, (\mathrm{pr}, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big),\\ &\big(\mathrm{id}_{\mathbb{R}^{d-1}}, 0, (\mathrm{pr}, \mathrm{id}_\mathbb{R}, \dots, \mathrm{id}_\mathbb{R})\big), \dots, \big(\mathrm{id}_{\mathbb{R}^3}, 0, (\mathrm{pr}, \mathrm{id}_\mathbb{R})\big), \big(\mathrm{id}_{\mathbb{R}^2}, 0, \mathrm{pr}\big)\big].\end{aligned}$$

Then $\mathcal{R}^{\xi_d} = f_d$, $\mathcal{D}_{\xi_d} = d-1$, $\mathcal{W}_{\xi_d} = d$ and $\mathcal{T}_{L,\xi_d} = 8^{(d-1)/2} r^{d(d-1)}$. Moreover, $Q^{\xi_d,n} = [-r^d, r^d]^2 \times \mathbb{R}^{d-n-1}$ for all $n \in \{1, \dots, d-1\}$. Hence, the fact that for all $n \in \{1, \dots, d-1\}$ and $x \in [-r, r]^d$ we have $|\prod_{i=1}^n x_i| \leq r^d$ ensures that $[-r, r]^d \subseteq \mathrm{Dom}_{Q,\xi_d}$. Now, we can conclude with Remark 5.3 and Corollary 6.4 using the inequality

$$\log_2\big(\sqrt{d}(d-1)8^{(d-1)/2}\varepsilon^{-1}\big) \leq \tfrac{31}{18} d \log_2(d) \log_2(\varepsilon^{-1})$$

if $r = 1$ and the inequality

$$\log_2\big(\sqrt{d}(d-1)8^{(d-1)/2} r^{d(d-1)}\varepsilon^{-1}\big) \leq \tfrac{67}{54} \log_2(r) d^2 \log_2(d) \log_2(\varepsilon^{-1})$$

if $r \geq 2$, for which we note that $\log_2(d) \geq \frac{3}{2}$ since $d \geq 3$. ∎

In our next result, we show the approximability of ridge functions based on a Lipschitz function.

Proposition 7.7. *Let $K, r, S \in [1, \infty)$, $\theta_d \in [-S, S]^d$, $d \in \mathbb{N}$, and suppose $f\colon \mathbb{R} \to \mathbb{R}$ is K-Lipschitz continuous with $|f(0)| \leq K$. Consider the ridge functions $g_d\colon \mathbb{R}^d \to \mathbb{R}$ given by $g_d(x) = f(\theta_d \cdot x)$. Then for all $d \in \mathbb{N}$ and $\varepsilon \in (0, 1]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\text{ReLU}} \in C(\mathbb{R}^d, \mathbb{R})$ such that*

(i) $\sup_{x\in[-r,r]^d} |g_d(x) - \mathcal{R}^\phi_{\text{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\text{ReLU}}$ *is $\sqrt{d}KS$-Lipschitz continuous on $\mathbb{R}^d$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{1}{7} 10^3 K^2 r S^2 d^6 \varepsilon^{-1}$.

Proof. Let $\mathcal{F} = \mathcal{F}^{\text{Lip}}_K$ be the K-Lipschitz catalog and suppose $Q = (Q_f)_{f\in\mathcal{F}}$ and $L = (L_f)_{f\in\mathcal{F}}$ are defined as in (i) of Example 4.1 (with drS instead of r). Let $\xi_d \in \mathcal{C}_\mathcal{F}$ be given by $\xi_d = (\theta_d^T, 0, f)$, where ^{T} denotes transposition. Then $\mathcal{R}^{\xi_d} = g_d$, $\mathcal{D}_{\xi_d} = 1$, $\mathcal{W}_{\xi_d} = d$ and $\mathcal{T}_{L,\xi_d} \leq \sqrt{d}KS$. Moreover, $[-r, r]^d \subseteq \{x \in \mathbb{R}^d\colon |\theta_d \cdot x| \leq drS\} \subseteq \mathrm{Dom}_{Q,\xi_d}$ by the Cauchy–Schwarz inequality. So the proposition follows from Remark 5.3 and Corollary 5.6. ∎

As our last example, we consider generalized Gaussian radial basis function networks, i.e. weighted sums of the Gaussian function applied to the distance of x to a given vector.

Proposition 7.8. *Let $N \in \mathbb{N}$, $r, S \in [1, \infty)$ with $r + S \geq 5$ as well as $\alpha_1, \dots, \alpha_N \in [0, S]$, $u_1, \dots, u_N \in [-S, S]$ and $v_{d,1}, \dots, v_{d,N} \in [-S, S]^d$, $d \in \mathbb{N}$. Consider generalized Gaussian radial basis function networks $f_d\colon \mathbb{R}^d \to \mathbb{R}$ with N neurons given by $f_d(x) = \sum_{i=1}^N u_i e^{-\alpha_i \|x - v_{d,i}\|^2}$. Then for all $d \in \mathbb{N}$ and $\varepsilon \in (0, (r+S)^{-3}]$, there exists $\phi \in \mathcal{N}$ with $\mathcal{R}^\phi_{\text{ReLU}} \in C(\mathbb{R}^d, \mathbb{R})$ such that*

(i) $\sup_{x\in[-r,r]^d} |f_d(x) - \mathcal{R}^\phi_{\text{ReLU}}(x)| \leq \varepsilon$,

(ii) $\mathcal{R}^\phi_{\text{ReLU}}$ *is $2(r+S)S^2\sqrt{d}N$-Lipschitz on $\mathbb{R}^d$ and*

(iii) $\mathcal{P}(\phi) \leq \frac{1}{6}10^4(r+S)S^2N^{11/2}d^5\varepsilon^{-1}$.

Proof. Let $\mathcal{F} = \mathcal{F}^{\mathrm{RBF}}$ be the Gaussian radial basis function catalog and suppose $Q = (Q_f)_{f\in\mathcal{F}}$ and $L = (L_f)_{f\in\mathcal{F}}$ are defined as in Example 4.4 (with $r+S$ instead of r). For all $d \in \mathbb{N}$, let $U_d \in \mathbb{R}^{dN\times d}$ be the block-matrix $U_d = (\mathrm{id}_{\mathbb{R}^d}, \dots, \mathrm{id}_{\mathbb{R}^d})^T$, $b_d \in \mathbb{R}^{dN}$ the vector $(v_{d,1}, \dots, v_{d,N})^T$, $V_d \in \mathbb{R}^{N\times dN}$ the block-matrix with $\alpha_i(1,\dots,1) \in \mathbb{R}^{1\times d}$ blocks on the i-th entry of the diagonal and 0 entries otherwise and $W \in \mathbb{R}^{1\times N}$ the matrix $(u_1, \dots, u_N)$. Moreover, let $\xi_d \in \mathcal{C}_{\mathcal{F}}$ be given by

$$\big[\big(U_d, -b_d, (\mathrm{sq}, \dots, \mathrm{sq})\big), \big(V_d, 0, (\mathrm{e}, \dots, \mathrm{e})\big), \big(W, 0, \mathrm{id}_{\mathbb{R}}\big)\big].$$

Then $\mathcal{R}^{\xi_d} = f_d$, $\mathcal{D}_{\xi_d} = 3$, $\mathcal{W}_{\xi_d} = dN$ and $\mathcal{T}_{L,\xi_d} \leq 2(r+S)\sqrt{d}NS^2$. Moreover, $Q^{\xi_d,1} = [-(r+S), r+S]^{dN}$, $Q^{\xi_d,2} = [0,\infty)^N$ and $Q^{\xi_d,3} = \mathbb{R}$. It follows that $[-r,r]^d \subseteq \mathrm{Dom}_{Q,\xi_d}$, and the proposition follows from Remark 5.3 and Corollary 5.6. ∎

CHAPTER 3

Non-convergence of stochastic gradient descent in the training of deep neural networks

This chapter is an adaptation of the published article [15].

1. Introduction

The main contribution of this chapter is a demonstration that stochastic gradient descent (SGD) fails to converge for ReLU networks if the number of random initializations does not increase fast enough compared to the size of the network. To illustrate our findings, we present a special case of our main result, Theorem 5.3, in Theorem 1.1 below.

We denote by $d \in \mathbb{N} = \{1, 2, \dots\}$ the dimension of the input domain of the approximation problem. The set $A_d = \bigcup_{D \in \mathbb{N}} (\{d\} \times \mathbb{N}^{D-1} \times \{1\})$ represents all network architectures with input dimension d and output dimension 1. In particular, a vector $a = (a_0, \dots, a_D) \in A_d$ describes the depth D of a network and the number of neurons $a_0, \dots, a_D$ in the different layers. For any such architecture a, the quantity $\mathcal{P}(a) = \sum_{j=1}^{D} a_j(a_{j-1} + 1)$ counts the number of real parameters, that is, the number of weights and biases of a deep neural network (DNN) with architecture a. We consider networks with ReLU activation in the hidden layers and a linear read-out map. That is, the realization function $\mathcal{R}_a^\theta \colon \mathbb{R}^d \to \mathbb{R}$ of a DNN with architecture $a = (a_0, \dots, a_D) \in A_d$ and weights and biases $\theta \in \mathbb{R}^{\mathcal{P}(a)}$ is given by

$$\mathcal{R}_a^\theta = \mathcal{A}_{a_D, a_{D-1}}^{\theta, \sum_{i=1}^{D-1} a_i(a_{i-1}+1)} \circ \rho \circ \mathcal{A}_{a_{D-1}, a_{D-2}}^{\theta, \sum_{i=1}^{D-2} a_i(a_{i-1}+1)} \circ \rho \circ \cdots \circ \mathcal{A}_{a_2, a_1}^{\theta, a_1(a_0+1)} \circ \rho \circ \mathcal{A}_{a_1, a_0}^{\theta, 0},$$

where $\mathcal{A}_{m,n}^{\theta,k} \colon \mathbb{R}^n \to \mathbb{R}^m$ denotes the affine mapping

$$(x_1, \dots, x_n) \mapsto \begin{pmatrix} \theta_{k+1} & \theta_{k+2} & \cdots & \theta_{k+n} \\ \theta_{k+n+1} & \theta_{k+n+2} & \cdots & \theta_{k+2n} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{k+(m-1)n+1} & \theta_{k+(m-1)n+2} & \cdots & \theta_{k+mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} \theta_{k+mn+1} \\ \theta_{k+mn+2} \\ \vdots \\ \theta_{k+mn+m} \end{pmatrix}$$

and $\rho \colon \bigcup_{k \in \mathbb{N}} \mathbb{R}^k \to \bigcup_{k \in \mathbb{N}} \mathbb{R}^k$ is the ReLU function

$$(x_1, \dots, x_k) \mapsto (\max\{x_1, 0\}, \dots, \max\{x_k, 0\}).$$

In the following description of the SGD algorithm, $n \in \mathbb{N}$ is the index of the trajectory, $t \in \mathbb{N}_0$ represents the index of the step along the trajectory, $m \in \mathbb{N}$ denotes the batch size of

the empirical risk, and $a \in A_d$ describes the architecture under consideration. We assume the training data is given by functions $X_j^{n,t}\colon \Omega \to [0,1]^d$ and $Y_j^{n,t}\colon \Omega \to [0,1]$, $j,n,t \in \mathbb{N}_0$, on a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$. In a typical learning problem, $(X_j^{n,t}, Y_j^{n,t})$, $j,n,t \in \mathbb{N}_0$, are i.i.d. random variables. But for Theorem 1.1 to hold, it is enough if $(X_j^{0,0}, Y_j^{0,0})$, $j \in \mathbb{N}_0$, are i.i.d. random variables, whereas $(X_j^{n,t}, Y_j^{n,t})\colon \Omega \to [0,1]^{d+1}$ are arbitrary mappings for $(n,t) \neq (0,0)$. The target function $\mathcal{E}\colon [0,1]^d \to [0,1]$ we are trying to learn is the factorized conditional expectation given ($\mathbb{P}$-a.s.) by $\mathcal{E}(X_0^{0,0}) = \mathbb{E}[Y_0^{0,0}|X_0^{0,0}]$. The empirical risk used for training is

$$\mathcal{L}_{a,m}^{n,t}(\theta) = \frac{1}{m}\sum_{j=1}^{m} \left|\mathfrak{c} \circ \mathcal{R}_a^\theta(X_j^{n,t}) - Y_j^{n,t}\right|^2, \tag{1.1}$$

where we compose the network realization with the clipping function $\mathfrak{c}\colon \mathbb{R} \to \mathbb{R}$, $x \mapsto \max\{0, \min\{x, 1\}\}$. This composition inside the risk is equivalent to a nonlinear read-out map of the network. However, it is more convenient for us to view $\mathfrak{c}$ as part of the risk criterion instead of the network. But this is only a matter of notation. Observe that (1.1) is a supervised learning task with noise since, in general, the best possible least squares approximation of $Y_0^{0,0}$ with a deterministic function of $X_0^{0,0}$ is $\mathcal{E}(X_0^{0,0})$, which is only equal to $Y_0^{0,0}$ in the special case where $Y_0^{0,0}$ is $X_0^{0,0}$-measurable. We let $\mathcal{G}_{a,m}^{n,t}\colon \mathbb{R}^{\mathcal{P}(a)} \times \Omega \to \mathbb{R}^{\mathcal{P}(a)}$ be a function that is equal to the gradient of $\mathcal{L}_{a,m}^{n,t}$ where it exists. The trajectories of the SGD algorithm are given by random variables $\Theta_{a,m}^{n,t}\colon \Omega \to \mathbb{R}^{\mathcal{P}(a)}$ satisfying the defining relation

$$\Theta_{a,m}^{n,t} = \Theta_{a,m}^{n,t-1} - \gamma_t \mathcal{G}_{a,m}^{n,t}(\Theta_{a,m}^{n,t-1})$$

for given step sizes $(\gamma_t)_{t\in\mathbb{N}} \subseteq \mathbb{R}$. Now, we are ready to state the following result, which is a consequence of [69, Theorem 6.5] and Corollary 5.4 below.

Theorem 1.1. *Assume that the target function $\mathcal{E}$ is Lipschitz continuous and that $\mathcal{E}(X_0^{0,0})$ is not $\mathbb{P}$-a.s.-constant. Suppose that, for all $a \in A_d$ and $m \in \mathbb{N}$, the random initializations $\Theta_{a,m}^{n,0}$, $n \in \mathbb{N}$, are independent and uniformly distributed on $[-c,c]^{\mathcal{P}(a)}$, where $c \in [2,\infty)$ is larger than the Lipschitz constant of $\mathcal{E}$. Let $\mathbf{k}_{a,M,N,T}\colon \Omega \to \mathbb{N} \times \mathbb{N}_0$ be random variables satisfying*

$$\mathbf{k}_{a,M,N,T}(\omega) \in \operatorname{argmin}_{(n,t)\in\{1,\dots,N\}\times\{0,\dots,T\},\, \Theta_{a,M}^{n,t}(\omega)\in[-c,c]^{\mathcal{P}(a)}} \mathcal{L}_{a,M}^{0,0}(\Theta_{a,M}^{n,t}(\omega), \omega). \tag{1.2}$$

Then, one has

$$\limsup_{\substack{a=(a_0,\dots,a_D)\in A_d \\ \min\{D,a_1,\dots,a_{D-1}\}\to\infty}} \limsup_{\substack{M,N\in\mathbb{N} \\ \min\{M,N\}\to\infty}} \sup_{T\in\mathbb{N}_0} \mathbb{E}\left[\min\left\{\int_{[0,1]^d} \left|\left(\mathfrak{c}\circ\mathcal{R}_a^{\Theta_{a,M}^{\mathbf{k}_{a,M,N,T}}}\right)(x) - \mathcal{E}(x)\right| \mathbb{P}_{X_0^{0,0}}(dx), 1\right\}\right] = 0 \tag{1.3}$$

and

$$\inf_{N\in\mathbb{N}} \limsup_{\substack{a=(a_0,\dots,a_D)\in A_d \\ \min\{D,a_1,\dots,a_{D-1}\}\to\infty}} \inf_{\substack{M\in\mathbb{N} \\ T\in\mathbb{N}_0}} \mathbb{E}\left[\min\left\{\int_{[0,1]^d} \left|\left(\mathfrak{c}\circ\mathcal{R}_a^{\Theta_{a,M}^{\mathbf{k}_{a,M,N,T}}}\right)(x) - \mathcal{E}(x)\right| \mathbb{P}_{X_0^{0,0}}(dx), 1\right\}\right] > 0. \tag{1.4}$$

The integrals in (1.3) and (1.4) describe the true risk. Note that in Theorem 1.1 the random initializations of the different trajectories are assumed to be independent and uniformly distributed on the hypercube $[-c,c]^{\mathcal{P}(a)}$, but our main result, Theorem 5.3 below, also covers more general cases. The random variable $\mathbf{k}_{a,M,N,T}$ determines the specific trajectory and gradient step among the first N trajectories and T steps which minimize the empirical risk corresponding to the batch size of M. Note that $\mathcal{E}(X_0^{0,0})$ not being a.s.-constant is a weak assumption since it merely means that the learning task is nontrivial. Moreover, the stronger condition that $\mathcal{E}$ must be Lipschitz continuous is made only to ensure the validity of the positive result (1.3), whereas our new contribution (1.4) does not require this. Similarly, we use the clipping function $\mathfrak{c}$ to ensure the validity of (1.3), which in [69] is formulated for networks with clipping function as read-out map.

As outlined in Chapter 1, our arguments are based on an analysis of regions in the parameter space related to "inactive" neurons. In these regions, the realization function is constant in the network parameter and, hence, SGD will not be able to escape these regions. We give precise estimates on the probability that the whole network becomes inactive and deduce that SGD fails to converge if the number of random initializations does not increase fast enough. Note that in (1.4) we take the limit superior over all architectures $(a_0,\dots,a_D)\in A_d$ whose depth D and minimal width $\min\{a_1,\dots,a_{D-1}\}$ both tend to infinity. In particular, to prove (1.4), it is sufficient to construct a single sequence of such architectures over which the limit is positive. For the sequence we use, the depth grows much faster than the maximal width $\max\{a_1,\dots,a_{D-1}\}$. This imbalance between the depth and the width has the effect that the training procedure does not converge.

The remainder of this chapter is organized as follows. In Section 2, we provide an abstract version of the SGD algorithm for training neural networks in a supervised learning framework. Section 3 contains preliminary results on inactive neurons and constant network realization functions. In Section 4, we discuss the consequences of these preliminary results for the convergence of the SGD method, and Section 5 contains our main results, Theorem 5.3 and Corollary 5.4.

2. Mathematical description of the SGD method

In this section, we give a mathematical description of an abstract version of the SGD algorithm for training neural networks in a supervised learning framework. To do that, we slightly generalize the setup of the introduction. We begin with an informal description and give a precise formulation afterwards. First, fix a network architecture $a=(a_0,\dots,a_D)\in A_d$. Let $\mathcal{X}\colon\Omega\to[u,v]^d$ and $\mathcal{B}\colon\Omega\to[\mathfrak{u},\mathfrak{v}]$ be random variables on a probability space $(\Omega,\mathcal{F},\mathbb{P})$, on which the true risk $\mathfrak{L}(\theta)=\mathbb{E}[|(\mathfrak{c}\circ\mathcal{R}_a^\theta)(\mathcal{X})-\mathcal{B}|]$ of a network $\theta\in\mathbb{R}^{\mathcal{P}(a)}$ is based. Here, $\mathfrak{c}\colon\mathbb{R}\to\mathbb{R}$ can be any continuous function, which covers the case of network realizations with nonlinear read-out maps. In the context of the introduction, $\mathcal{B}$ stands for the random variable $\mathcal{E}(X_0^{0,0})$. Throughout, $n\in\mathbb{N}$ will denote the index of the gradient trajectory and $t\in\mathbb{N}_0$ the index of the gradient step. $L^{n,t}$ denotes the empirical risk defined on the space of functions $C(\mathbb{R}^d,\mathbb{R})$. In this general setting, $L^{n,t}$ can be any function from $C(\mathbb{R}^d,\mathbb{R})\times\Omega$ to $\mathbb{R}$, but the specific example we have in mind is

$$L^{n,t}(f)=\frac{1}{m}\sum_{j=1}^{m}\left|f(X_j^{n,t})-Y_j^{n,t}\right|^2$$

for a given batch size $m \in \mathbb{N}$. $\mathcal{L}^{n,t}$ is the empirical risk defined on the space of network parameters, given in terms of $L^{n,t}$ by $\mathcal{L}^{n,t}(\theta) = L^{n,t}(\mathfrak{c} \circ \mathcal{R}_a^\theta)$. Let $\mathcal{G}^{n,t} \colon \mathbb{R}^{\mathcal{P}(a)} \times \Omega \to \mathbb{R}^{\mathcal{P}(a)}$ be a function that agrees with the gradient of $\mathcal{L}^{n,t}$ where it exists. Then, we can introduce the gradient trajectories $\Theta^{n,t} \colon \Omega \to \mathbb{R}^{\mathcal{P}(a)}$ satisfying

$$\Theta^{n,t} = \Theta^{n,t-1} - \gamma_t \mathcal{G}^{n,t}(\Theta^{n,t-1})$$

for given step sizes γ_t. The N random initializations $\Theta^{n,0}$, $n \in \{1, \ldots, N\}$, are assumed to be i.i.d in n and have independent marginals. Lastly, $\mathbf{k} \colon \Omega \to \mathbb{N} \times \mathbb{N}_0$ specifies the output of the algorithm consisting of a pair of indices for a gradient trajectory and a gradient step. The expected true risk is $\mathcal{V} = \mathbb{E}[\min\{\mathfrak{L}(\Theta^{\mathbf{k}}), 1\}]$. In the following, we present the formal algorithm.

Setting 2.1. Let $u, \mathfrak{u} \in \mathbb{R}$, $v \in (u, \infty)$, $\mathfrak{v} \in (\mathfrak{u}, \infty)$, $\mathfrak{c} \in C(\mathbb{R}, \mathbb{R})$, $d, D, N \in \mathbb{N}$, $a = (a_0, \ldots, a_D) \in A_d$, and $(\gamma_t)_{t \in \mathbb{N}} \subseteq \mathbb{R}$. Consider random variables $\mathcal{X} \colon \Omega \to [u, v]^d$ and $\mathcal{B} \colon \Omega \to [\mathfrak{u}, \mathfrak{v}]$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $\mathfrak{L} \colon \mathbb{R}^{\mathcal{P}(a)} \to [0, \infty]$ be given by $\mathfrak{L}(\theta) = \mathbb{E}[|(\mathfrak{c} \circ \mathcal{R}_a^\theta)(\mathcal{X}) - \mathcal{B}|]$. For all $n \in \mathbb{N}$ and $t \in \mathbb{N}_0$, let $L^{n,t}$ be a function from $C(\mathbb{R}^d, \mathbb{R}) \times \Omega$ to $\mathbb{R}$, and denote by $\mathcal{L}^{n,t} \colon \mathbb{R}^{\mathcal{P}(a)} \times \Omega \to \mathbb{R}$ the mapping given by $\mathcal{L}^{n,t}(\theta) = L^{n,t}(\mathfrak{c} \circ \mathcal{R}_a^\theta)$. Let $\mathcal{G}^{n,t} = (\mathcal{G}_1^{n,t}, \ldots, \mathcal{G}_{\mathcal{P}(a)}^{n,t}) \colon \mathbb{R}^{\mathcal{P}(a)} \times \Omega \to \mathbb{R}^{\mathcal{P}(a)}$ be a function satisfying

$$\mathcal{G}_i^{n,t}(\theta, \omega) = \frac{\partial}{\partial \theta_i} \mathcal{L}^{n,t}(\theta, \omega) \tag{2.1}$$

for all $n, t \in \mathbb{N}$, $i \in \{1, \ldots, \mathcal{P}(a)\}$, $\omega \in \Omega$, and

$$\theta \in \left\{ \vartheta = (\vartheta_1, \ldots, \vartheta_{\mathcal{P}(a)}) \in \mathbb{R}^{\mathcal{P}(a)} \colon \begin{array}{l} \mathcal{L}^{n,t}(\vartheta_1, \ldots, \vartheta_{i-1}, (\cdot), \vartheta_{i+1}, \ldots, \vartheta_{\mathcal{P}(a)}, \omega) \\ \text{as a function } \mathbb{R} \to \mathbb{R} \text{ is differentiable at } \vartheta_i. \end{array} \right\}.$$

Let $\Theta^{n,t} = (\Theta_1^{n,t}, \ldots, \Theta_{\mathcal{P}(a)}^{n,t}) \colon \Omega \to \mathbb{R}^{\mathcal{P}(a)}$, $n \in \mathbb{N}$, $t \in \mathbb{N}_0$, be random variables such that $\Theta^{1,0}, \ldots, \Theta^{N,0}$ are i.i.d., $\Theta_1^{1,0}, \ldots, \Theta_{\mathcal{P}(a)}^{1,0}$ are independent, and

$$\Theta^{n,t} = \Theta^{n,t-1} - \gamma_t \mathcal{G}^{n,t}(\Theta^{n,t-1}) \tag{2.2}$$

for all $n, t \in \mathbb{N}$. Let $\mathbf{k} \colon \Omega \to \{1, \ldots, N\} \times \mathbb{N}_0$ be a random variable, and denote $\mathcal{V} = \mathbb{E}[\min\{\mathfrak{L}(\Theta^{\mathbf{k}}), 1\}]$.

Note that, by [69, Lemma 6.2] and Tonelli's theorem, it follows from Setting 2.1 that $\mathfrak{L}(\Theta^{\mathbf{k}}) \colon \Omega \to [0, \infty]$ is measurable and, as a consequence, $\mathcal{V} = \mathbb{E}[\min\{\mathfrak{L}(\Theta^{\mathbf{k}}), 1\}]$ is well-defined.

3. DNNs with constant realization functions

In this section, we study a subset of the parameter space, specified in Definition 3.1 below, for which neurons in a DNN become "inactive", rendering the realization function of the DNN constant. We deduce a few properties for such DNNs in Lemmas 3.2, 3.3, and 3.4 below. The material in this section is related to the findings in [90, 117].

Definition 3.1. Let $D \in \mathbb{N}$ and $a = (a_0, \ldots, a_D) \in \mathbb{N}^{D+1}$. For all $j \in \mathbb{N} \cap (0, D)$, let $\mathfrak{I}_{a,j} \subseteq \mathbb{R}^{\mathcal{P}(a)}$ be the set

$$\mathfrak{I}_{a,j} = \left\{ \theta = (\theta_1, \ldots, \theta_{\mathcal{P}(a)}) \in \mathbb{R}^{\mathcal{P}(a)} \colon \left[\forall\, k \in \mathbb{N} \cap \left(\sum_{i=1}^{j-1} a_i(a_{i-1}+1), \sum_{i=1}^{j} a_i(a_{i-1}+1) \right] \colon \theta_k < 0 \right] \right\},$$

and denote $\mathrm{I}_a = \bigcup_{j \in \mathbb{N} \cap (1,D)} \mathfrak{I}_{a,j}$.

First, we verify that the realization function is constant in both the argument and the network parameter on certain subsets of $\mathfrak{I}_{a,j}$.

Lemma 3.2. *Let $D \in \mathbb{N}$, $j \in \mathbb{N} \cap (1, D)$, $a = (a_0, \ldots, a_D) \in \mathbb{N}^{D+1}$, $\theta = (\theta_1, \ldots, \theta_{\mathcal{P}(a)})$, $\vartheta = (\vartheta_1, \ldots, \vartheta_{\mathcal{P}(a)}) \in \mathfrak{I}_{a,j}$, $x \in \mathbb{R}^{a_0}$, and assume that $\theta_k = \vartheta_k$ for all $k \in \mathbb{N} \cap \big(\sum_{i=1}^{j} a_i(a_{i-1} + 1), \mathcal{P}(a) \big]$. Then $\mathcal{R}_a^\theta(0) = \mathcal{R}_a^\theta(x) = \mathcal{R}_a^\vartheta(x) = \mathcal{R}_a^\vartheta(0)$.*

Proof. For all $k \in \{1, \ldots, D\}$, denote $m_k = \sum_{i=1}^{k} a_i(a_{i-1} + 1)$. Since, by assumption, $\theta, \vartheta \in \mathfrak{I}_{a,j}$, one has for all $k \in \mathbb{N} \cap (m_{j-1}, m_j]$ that $\theta_k < 0$ and $\vartheta_k < 0$. This and $\rho(\mathbb{R}^{a_{j-1}}) = [0, \infty)^{a_{j-1}}$ imply for all $y \in \mathbb{R}^{a_{j-1}}$, $\phi \in \{\theta, \vartheta\}$ that $\mathcal{A}_{a_j, a_{j-1}}^{\phi, m_{j-1}} \circ \rho(y) \in (-\infty, 0]^{a_j}$. This ensures for all $y \in \mathbb{R}^{a_{j-1}}$, $\phi \in \{\theta, \vartheta\}$ that $\rho \circ \mathcal{A}_{a_j, a_{j-1}}^{\phi, m_{j-1}} \circ \rho(y) = 0$. Moreover, the assumption that $\theta_k = \vartheta_k$ for all $k \in \mathbb{N} \cap \big(\sum_{i=1}^{j} a_i(a_{i-1} + 1), \mathcal{P}(a) \big]$ yields $\mathcal{A}_{a_k, a_{k-1}}^{\theta, m_{k-1}} = \mathcal{A}_{a_k, a_{k-1}}^{\vartheta, m_{k-1}}$ for all $k \in \mathbb{N} \cap (j, D]$. This implies that $\mathcal{R}_a^\theta(y) = \mathcal{R}_a^\vartheta(z)$ for all $y, z \in \mathbb{R}^{a_0}$, which completes the proof of Lemma 3.2. ∎

The next lemma shows that networks with parameters in I_a cannot perform better than a constant solution to the learning task.

Lemma 3.3. *Assume Setting 2.1 and let $\theta \in \mathrm{I}_a$. Then $\mathfrak{L}(\theta) \geq \inf_{b \in \mathbb{R}} \mathbb{E}[|b - \mathcal{B}|]$.*

Proof. Let $\zeta \in \Omega$. By Lemma 3.2, one has $\mathcal{R}_a^\theta(x) = \mathcal{R}_a^\theta(0)$ for all $x \in \mathbb{R}^d$. Therefore, we obtain $\mathcal{R}_a^\theta(\mathcal{X}(\omega)) = \mathcal{R}_a^\theta(\mathcal{X}(\zeta))$ for all $\omega \in \Omega$. In particular, $\mathfrak{L}(\theta) = \mathbb{E}[|(\mathfrak{c} \circ \mathcal{R}_a^\theta)(\mathcal{X}(\zeta)) - \mathcal{B}|] \geq \inf_{b \in \mathbb{R}} \mathbb{E}[|b - \mathcal{B}|]$. ∎

Finally, we show that SGD cannot escape from I_a.

Lemma 3.4. *Assume Setting 2.1 and let $n, t \in \mathbb{N}$, $\omega \in \Omega$, $j \in \mathbb{N} \cap (1, D)$. Suppose that $\Theta^{n,0}(\omega) \in \mathfrak{I}_{a,j}$. Then $\Theta^{n,t}(\omega) \in \mathfrak{I}_{a,j}$.*

Proof. Denote $m_0 = \sum_{i=1}^{j-1} a_i(a_{i-1} + 1)$ and $m_1 = \sum_{i=1}^{j} a_i(a_{i-1} + 1)$. We prove by induction that for all $s \in \mathbb{N}_0$ we have $\Theta^{n,s}(\omega) \in \mathfrak{I}_{a,j}$. The case $s = 0$ is true by assumption. Now suppose that $s \in \mathbb{N}_0$ and $\theta = (\theta_1, \ldots, \theta_{\mathcal{P}(a)}) \in \mathbb{R}^{\mathcal{P}(a)}$ satisfy $\theta = \Theta^{n,s}(\omega) \in \mathfrak{I}_{a,j}$. Let $U \subseteq \mathbb{R}^{\mathcal{P}(a)}$ be the set given by $U = \{(\theta_1, \ldots, \theta_{m_0})\} \times (-\infty, 0)^{m_1 - m_0} \times \{(\theta_{m_1+1}, \ldots, \theta_{\mathcal{P}(a)})\}$. Then $\theta \in U \subseteq \mathfrak{I}_{a,j}$. By Lemma 3.2, we have $\mathcal{R}_a^\phi(x) = \mathcal{R}_a^\theta(x)$ for all $\phi \in U$ and $x \in \mathbb{R}^d$. Hence, $\mathcal{L}^{n,s+1}(\phi, \omega) = \mathcal{L}^{n,s+1}(\theta, \omega)$ for all $\phi \in U$ and, as a consequence, $\frac{\partial}{\partial \theta_k} \mathcal{L}^{n,s+1}(\theta, \omega) = 0$ for all $k \in \mathbb{N} \cap (m_0, m_1]$. So, it follows from (2.1), (2.2), and the induction hypothesis that $\Theta^{n,s+1}(\omega) \in \mathfrak{I}_{a,j}$, which completes the proof of Lemma 3.4. ∎

4. Quantitative lower bounds for the SGD method in the training of DNNs

In this section, we establish in Proposition 4.2 below a quantitative lower bound for the error of the SGD method in the training of DNNs.

Lemma 4.1. *Assume Setting 2.1 and suppose $D \geq 3$. For all $j \in \{1, \ldots, D-1\}$, denote $k_j = \sum_{i=1}^{j} a_i(a_{i-1}+1)$, $p = \inf_{i \in \{1,\ldots,\mathcal{P}(a)\}} \mathbb{P}(\Theta_i^{1,0} < 0)$, and $W = \max\{a_1, \ldots, a_{D-1}\}$. Then*

$$\begin{aligned}\mathbb{P}\big(\forall\, n \in \{1,\ldots,N\},\, t \in \mathbb{N}_0 \colon \Theta^{n,t} \in \mathrm{I}_a\big) &= \Bigg[1 - \prod_{j=2}^{D-1}\Big(1 - \prod_{i=1+k_{j-1}}^{k_j} \mathbb{P}(\Theta_i^{1,0} < 0)\Big)\Bigg]^N \\ &\geq \big[1 - (1-p^{W(W+1)})^{D-2}\big]^N.\end{aligned}$$

Proof. It follows from the independence of $\Theta_1^{1,0}, \ldots, \Theta_{\mathcal{P}(a)}^{1,0}$ that

$$\begin{aligned}\mathbb{P}(\Theta^{1,0} \in \mathrm{I}_a) &= \mathbb{P}\big(\exists\, j \in \mathbb{N} \cap (1, D) \colon \forall\, i \in \mathbb{N} \cap (k_{j-1}, k_j] \colon \Theta_i^{1,0} < 0\big) \\ &= 1 - \prod_{j=2}^{D-1}\Big(1 - \prod_{i=1+k_{j-1}}^{k_j} \mathbb{P}(\Theta_i^{1,0} < 0)\Big).\end{aligned}$$

By definition of p and W, the right hand side is greater than or equal to $1-(1-p^{W(W+1)})^{D-2}$. Moreover, Lemma 3.4 and the assumption that $\Theta^{1,0}, \ldots, \Theta^{N,0}$ are i.i.d. yield

$$\begin{aligned}\mathbb{P}\big(\forall\, n \in \{1,\ldots,N\},\, t \in \mathbb{N}_0 \colon \Theta^{n,t} \in \mathrm{I}_a\big) &= \mathbb{P}\big(\forall\, n \in \{1,\ldots,N\} \colon \Theta^{n,0} \in \mathrm{I}_a\big) \\ &= \big(\mathbb{P}(\Theta^{1,0} \in \mathrm{I}_a)\big)^N,\end{aligned}$$

which completes the proof of Lemma 4.1. ∎

Proposition 4.2. *Under the same assumptions as in Lemma 4.1, one has*

$$\begin{aligned}\mathcal{V} = \mathbb{E}[\min\{\mathfrak{L}(\Theta^{\mathbf{k}}), 1\}] &\geq \Bigg[1 - \prod_{j=2}^{D-1}\Big(1 - \prod_{i=1+k_{j-1}}^{k_j} \mathbb{P}(\Theta_i^{1,0} < 0)\Big)\Bigg]^N \min\Big\{\inf_{b \in \mathbb{R}} \mathbb{E}[|b - \mathcal{B}|], 1\Big\} \\ &\geq \big[1 - (1-p^{W(W+1)})^{D-2}\big]^N \min\Big\{\inf_{b \in \mathbb{R}} \mathbb{E}[|b - \mathcal{B}|], 1\Big\}.\end{aligned} \tag{4.1}$$

Proof. Denote $C = \min\{\inf_{b \in \mathbb{R}} \mathbb{E}[|b - \mathcal{B}|], 1\}$ and observe that Lemma 3.3 implies for all $\omega \in \Omega$ with $\Theta^{\mathbf{k}(\omega)}(\omega) \in \mathrm{I}_a$ that $\min\{\mathfrak{L}(\Theta^{\mathbf{k}(\omega)}(\omega)), 1\} \geq C$. Markov's inequality hence ensures that

$$C\,\mathbb{P}(\Theta^{\mathbf{k}} \in \mathrm{I}_a) \leq C\,\mathbb{P}(\min\{\mathfrak{L}(\Theta^{\mathbf{k}}), 1\} \geq C) \leq \mathcal{V}.$$

Combining this with Lemma 4.1 and the fact that $\mathbb{P}(\Theta^{\mathbf{k}} \in \mathrm{I}_a) \geq \mathbb{P}(\forall\, n \in \{1,\ldots,N\},\, t \in \mathbb{N}_0 \colon \Theta^{n,t} \in \mathrm{I}_a)$ establishes (4.1). ∎

Let us briefly discuss how the inequality in Proposition 4.2 relates to prior work in the literature. Fix a depth $D \in \mathbb{N}$ and consider the problem of distributing a given number of neurons among the $D-1$ hidden layers. In order to minimize the chance of starting with an inactive network, one needs to minimize the quantity $1 - \prod_{j=2}^{D-1}(1 - \prod_{i=1+k_{j-1}}^{k_j} \mathbb{P}(\Theta_i^{1,0} < 0))$

from (4.1). Under the assumption that $\mathbb{P}(\Theta_i^{1,0} < 0)$ does not depend on i, this can be achieved by choosing the same number of neurons in each layer.

The effects of initialization and architecture on early training have also been studied in [52, 54]. While [52] investigates the problem of vanishing and exploding gradients, [54] studies two failure modes associated with poor starting conditions. Both find that, given a total number of neurons to spend, distributing them evenly among the hidden layers, yields the best results. This is in line with our findings.

5. Non-convergence of the SGD method in the training of DNNs

In this section, we prove the chapter's main results, Theorem 5.3 and Corollary 5.4. While Theorem 5.3 provides precise quantitative conditions under which SGD does not converge in the training of DNNs, Corollary 5.4 is a qualitative result. To prove them, we need the following elementary result. Throughout, log denotes the natural logarithm.

Lemma 5.1. *Let $D, N, W \in (0, \infty)$ and $\kappa, p \in (0, 1)$ be such that $D \geq |\log(p)|Wp^{-W}$ and $N \leq |\log(\kappa)|(1-p^W)^{1-D}$. Then $[1-(1-p^W)^D]^N \geq \kappa$.*

Proof. Let the functions $f\colon [0,1) \to \mathbb{R}$ and $g\colon [0,1) \to \mathbb{R}$ be given by $f(x) = x + \log(1-x)$ and $g(x) = (1-p^W)^{-1}x + \log(1-x)$. Since $f(0) = 0$ and $f'(x) = 1-(1-x)^{-1} < 0$ for all $x \in (0,1)$, one has $|\log(1-x)|^{-1} < x^{-1}$ for all $x \in (0,1)$. Hence, $D > |\log(p)|W|\log(1-p^W)|^{-1}$, from which it follows that $(1-p^W)^D < p^W$. In addition, $g(0) = 0$ and $g'(x) = (1-p^W)^{-1} - (1-x)^{-1} > 0$ for all $x \in (0, p^W)$, which implies that $|\log(1-x)| < (1-p^W)^{-1}x$ for all $x \in (0, p^W)$. Hence, we deduce from $(1-p^W)^D < p^W$ that $N|\log(1-(1-p^W)^D)| < N(1-p^W)^{D-1} \leq |\log(\kappa)|$, and taking the exponential yields the desired statement. ∎

We proved Proposition 4.2 in the abstract framework of Setting 2.1. For the sake of concreteness, we now return to the setup of the introduction. We quickly recall it below.

Setting 5.2. Let $u, \mathfrak{u} \in \mathbb{R}$, $v \in (u, \infty)$, $\mathfrak{v} \in (\mathfrak{u}, \infty)$, $\mathfrak{c} \in C(\mathbb{R}, \mathbb{R})$, $d \in \mathbb{N}$, and $(\gamma_t)_{t\in\mathbb{N}} \subseteq \mathbb{R}$. Consider functions $X_j^{n,t}\colon \Omega \to [u,v]^d$ and $Y_j^{n,t}\colon \Omega \to [\mathfrak{u}, \mathfrak{v}]$, $j, n, t \in \mathbb{N}_0$, on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that $X_0^{0,0}$ and $Y_0^{0,0}$ are random variables. Let $\mathcal{E}\colon [u,v]^d \to [\mathfrak{u}, \mathfrak{v}]$ be a measurable function such that $\mathbb{P}$-a.s. $\mathcal{E}(X_0^{0,0}) = \mathbb{E}[Y_0^{0,0}|X_0^{0,0}]$. Let $\mathcal{L}_{a,m}^{n,t}\colon \mathbb{R}^{\mathcal{P}(a)} \times \Omega \to \mathbb{R}$, $m \in \mathbb{N}$, $n, t \in \mathbb{N}_0$, $a \in A_d$, be given by

$$\mathcal{L}_{a,m}^{n,t}(\theta) = \frac{1}{m}\sum_{j=1}^{m} \left|(\mathfrak{c} \circ \mathcal{R}_a^\theta)(X_j^{n,t}) - Y_j^{n,t}\right|^2, \tag{5.1}$$

and assume $\mathcal{G}_{a,m}^{n,t} = (\mathcal{G}_{a,m,1}^{n,t}, \ldots, \mathcal{G}_{a,m,\mathcal{P}(a)}^{n,t})\colon \mathbb{R}^{\mathcal{P}(a)} \times \Omega \to \mathbb{R}^{\mathcal{P}(a)}$ are mappings satisfying

$$\mathcal{G}_{a,m,i}^{n,t}(\theta, \omega) = \frac{\partial}{\partial \theta_i}\mathcal{L}_{a,m}^{n,t}(\theta, \omega)$$

for all $m, n, t \in \mathbb{N}$, $a \in A_d$, $i \in \{1, \ldots, \mathcal{P}(a)\}$, $\omega \in \Omega$, and

$$\theta =\in \left\{\vartheta = (\vartheta_1, \ldots, \vartheta_{\mathcal{P}(a)}) \in \mathbb{R}^{\mathcal{P}(a)}\colon \begin{array}{l}\mathcal{L}_{a,m}^{n,t}(\vartheta_1, \ldots, \vartheta_{i-1}, (\cdot), \vartheta_{i+1}, \ldots, \vartheta_{\mathcal{P}(a)}, \omega) \\ \text{as a function } \mathbb{R} \to \mathbb{R} \text{ is differentiable at } \vartheta_i.\end{array}\right\}.$$

Let $\Theta^{n,t}_{a,m} = (\Theta^{n,t}_{a,m,1}, \dots, \Theta^{n,t}_{a,m,\mathcal{P}(a)})\colon \Omega \to \mathbb{R}^{\mathcal{P}(a)}$, $m, n \in \mathbb{N}$, $t \in \mathbb{N}_0$, $a \in A_d$, be random variables such that $\Theta^{n,0}_{a,m}$, $n \in \mathbb{N}$, are i.i.d., $\Theta^{1,0}_{a,m,1}, \dots, \Theta^{1,0}_{a,m,\mathcal{P}(a)}$ are independent for all $m \in \mathbb{N}$, $a \in A_d$, and

$$\Theta^{n,t}_{a,m} = \Theta^{n,t-1}_{a,m} - \gamma_t \mathcal{G}^{n,t}_{a,m}(\Theta^{n,t-1}_{a,m})$$

for all $m, n, t \in \mathbb{N}$, $a \in A_d$. Let $\mathbf{k}_{a,M,N,T}\colon \Omega \to \{1, \dots, N\} \times \mathbb{N}_0$, $M, N \in \mathbb{N}$, $T \in \mathbb{N}_0$, $a \in A_d$, be random variables.

The following is the main result of this chapter.

Theorem 5.3. *Assume Setting 5.2 and fix $M \in \mathbb{N}$. Consider sequences $(D_l, N_l, W_l)_{l \in \mathbb{N}_0} \subseteq \mathbb{N}^3$, $(a^l)_{l \in \mathbb{N}_0} = (a^l_0, \dots, a^l_{D_l})_{l \in \mathbb{N}_0} \subseteq A_d$ and constants $\kappa, p \in (0,1)$ such that, for all $l \in \mathbb{N}_0$, $W_l = \max\{a^l_1, \dots, a^l_{D_l - 1}\}$, $D_l \geq |\log(p)| W_l(W_l+1) p^{-W_l(W_l+1)} + 2$, and $N_l \leq |\log(\kappa)|(1 - p^{W_l(W_l+1)})^{3-D_l}$. Let $\Phi_{l,T}\colon \Omega \to \mathbb{R}^{\mathcal{P}(a^l)}$, $l, T \in \mathbb{N}_0$, be given by $\Phi_{l,T} = \Theta^{\mathbf{k}_{a^l,M,N_l,T}}_{a^l,M}$, and assume that $\inf_{l \in \mathbb{N}_0} \inf_{i \in \{1,\dots,\mathcal{P}(a^l)\}} \mathbb{P}\big(\Theta^{1,0}_{a^l,M,i} < 0\big) \geq p$. Then*

$$\begin{aligned} \liminf_{l \to \infty} \inf_{T \in \mathbb{N}_0} \mathbb{E}\Bigg[\min\Bigg\{ \int_{[u,v]^d} \Big| \Big(\mathfrak{c} \circ \mathcal{R}^{\Phi_{l,T}}_{a^l} \Big)(x) - \mathcal{E}(x) \Big| \, \mathbb{P}_{X^{0,0}_0}(dx), 1 \Bigg\} \Bigg] \\ \geq \kappa \min\Big\{ \inf_{b \in \mathbb{R}} \mathbb{E}\big[|b - \mathcal{E}(X^{0,0}_0)|\big], 1 \Big\}. \end{aligned} \tag{5.2}$$

Proof. Denote $q = \inf_{l \in \mathbb{N}_0} \inf_{i \in \{1,\dots,\mathcal{P}(a^l)\}} \mathbb{P}\big(\Theta^{1,0}_{a^l,M,i} < 0\big)$. By Proposition 4.2, one has, for all $l, T \in \mathbb{N}_0$,

$$\begin{aligned} &\mathbb{E}\Bigg[\min\Bigg\{ \int_{[u,v]^d} \Big| \Big(\mathfrak{c} \circ \mathcal{R}^{\Phi_{l,T}}_{a^l} \Big)(x) - \mathcal{E}(x) \Big| \, \mathbb{P}_{X^{0,0}_0}(dx), 1 \Bigg\} \Bigg] \\ &\geq \big[1 - (1 - q^{W_l(W_l+1)})^{D_l - 2}\big]^{N_l} \min\Big\{ \inf_{b \in \mathbb{R}} \mathbb{E}\big[|b - \mathcal{E}(X^{0,0}_0)|\big], 1 \Big\}. \end{aligned}$$

Moreover, Lemma 5.1 implies that, for all $l \in \mathbb{N}_0$,

$$\big[1 - (1 - q^{W_l(W_l+1)})^{D_l - 2}\big]^{N_l} \geq \big[1 - (1 - p^{W_l(W_l+1)})^{D_l - 2}\big]^{N_l} \geq \kappa,$$

which completes the proof of Theorem 5.3. ∎

Instead of focusing on a single sequence of architectures as in Theorem 5.3, one can instead consider the limit superior over all possible architectures, which we do in Corollary 5.4 below. Note that this allows us to increase the constant κ from (5.2) to 1.

Corollary 5.4. *Assume Setting 5.2 and let $c \in (0, \infty)$. Suppose that $\operatorname{Var}(\mathcal{E}(X^{0,0}_0)) > 0$ and assume that $\Theta^{n,0}_{a,m}$ is uniformly distributed on $[-c, c]^{\mathcal{P}(a)}$ for all $m, n \in \mathbb{N}$, $a \in A_d$. Then*

$$\begin{aligned} \inf_{N \in \mathbb{N}} \limsup_{\substack{a = (a_0, \dots, a_D) \in A_d \\ \min\{D, a_1, \dots, a_{D-1}\} \to \infty}} \inf_{\substack{M \in \mathbb{N} \\ T \in \mathbb{N}_0}} \mathbb{E}\Bigg[\min\Bigg\{ \int_{[u,v]^d} \Big| \Big(\mathfrak{c} \circ \mathcal{R}^{\Theta^{\mathbf{k}_{a,M,N,T}}_{a,M}}_{a} \Big)(x) - \mathcal{E}(x) \Big| \, \mathbb{P}_{X^{0,0}_0}(dx), 1 \Bigg\} \Bigg] \\ \geq \min\Big\{ \inf_{b \in \mathbb{R}} \mathbb{E}\big[|b - \mathcal{E}(X^{0,0}_0)|\big], 1 \Big\} > 0. \end{aligned} \tag{5.3}$$

Proof. First note that $\inf_{M\in\mathbb{N}}\inf_{a\in A_d}\inf_{i\in\{1,\dots,\mathcal{P}(a)\}}\mathbb{P}(\Theta^{1,0}_{a,M,i}<0)=\frac{1}{2}$. So, it follows from Theorem 5.3 that for all $k,N\in\mathbb{N}$, $\kappa\in(0,1)$ there exist $D\in\mathbb{N}$ and $a=(a_0,\dots,a_D)\in A_d$ such that $\min\{D,a_1,\dots,a_{D-1}\}\geq k$ and

$$\inf_{T\in\mathbb{N}_0}\mathbb{E}\Bigg[\min\Bigg\{\int_{[u,v]^d}\big|\big(\mathfrak{c}\circ\mathcal{R}_a^{\Theta^{\mathbf{k}_{a,M,N,T}}_{a,M}}\big)(x)-\mathcal{E}(x)\big|\,\mathbb{P}_{X_0^{0,0}}(dx),1\Bigg\}\Bigg]\\ \geq\kappa\min\Big\{\inf_{b\in\mathbb{R}}\mathbb{E}\big[|b-\mathcal{E}(X_0^{0,0})|\big],1\Big\}$$

for all $M\in\mathbb{N}$. As a result, one has

$$\inf_{N\in\mathbb{N}}\limsup_{\substack{a=(a_0,\dots,a_D)\in A_d\\ \min\{D,a_1,\dots,a_{D-1}\}\to\infty}}\inf_{\substack{M\in\mathbb{N}\\ T\in\mathbb{N}_0}}\mathbb{E}\Bigg[\min\Bigg\{\int_{[u,v]^d}\left|\left(\mathfrak{c}\circ\mathcal{R}_a^{\Theta^{\mathbf{k}_{a,M,N,T}}_{a,M}}\right)(x)-\mathcal{E}(x)\right|\,\mathbb{P}_{X_0^{0,0}}(dx),1\Bigg\}\Bigg]\\ \geq\kappa\min\Big\{\inf_{b\in\mathbb{R}}\mathbb{E}\big[|b-\mathcal{E}(X_0^{0,0})|\big],1\Big\}$$

for all $\kappa\in(0,1)$. Taking the limit $\kappa\uparrow 1$ and noting that the assumption $\mathrm{Var}(\mathcal{E}(X_0^{0,0}))>0$ implies $\inf_{b\in\mathbb{R}}\mathbb{E}\big[|b-\mathcal{E}(X_0^{0,0})|\big]>0$ completes the proof of the corollary. ■

CHAPTER 4

Landscape analysis for shallow neural networks: complete classification of critical points for affine target functions

This chapter is an adaptation of the published article [18].

1. Introduction

In this chapter, we conduct a landscape analysis of the loss surface. Our goal is to understand the occurrence and frequency of critical points of the loss function and knowing their type. We consider shallow networks with (leaky) ReLU or quadratic activation. As mentioned in Chapter 1, we do not impose assumptions on the network model that are not met in practice, but instead focus on special target functions, namely affine ones. In this framework, we provide a complete classification of the critical points of the true loss. We do so by unfolding the combinatorics of the problem, governed by different types of hidden neurons appearing in a network.

Using the classification in this chapter, we are able to derive results about the existence of strictly negative eigenvalues of the Hessian at most of the saddle points (understood in a suitable sense because we have to deal with differentiability issues arising from the (leaky) ReLU activation). This will be important later in Chapter 5.

The remainder of this chapter is organized as follows. The first activation function we consider is the ReLU activation in Section 2. We begin by introducing the relevant notation and definitions, including a new description of the types of hidden neurons that can appear in a ReLU network, in Sections 2.1 and 2.2. The first main result, the classification for ReLU networks, is Theorem 2.4 in Section 2.3. The remainder of Section 2 is dedicated to proving the classification. More precisely, we discuss a few important ingredients for the proof in Section 2.4. Thereafter, Section 2.5 is devoted to the differentiability and regularity properties of the loss function in view of the non-differentiability of the ReLU activation. The heart of the proof is contained in Sections 2.6 and 2.7. Finally, we establish in Section 2.8 a special case of Theorem 2.4 and deduce it in full generality afterwards in Section 2.9. Section 3 is concerned with extending the classification to leaky ReLU, stated as our second main result in Theorem 3.5, which heavily relies on understanding the ReLU case. To conclude, we also classify the critical points for networks with the quadratic activation in our third main result, Theorem 4.1 in Section 4.

2. Classification for ReLU activation

2.1 Notation and formal problem description

For simplicity, we focus on shallow networks with a single input and output neuron. The set of such networks with $N \in \mathbb{N}$ hidden neurons can be parametrized by $\mathbb{R}^{3N+1}$. We begin by describing the problem for the ReLU activation function $x \mapsto \max\{x, 0\}$. We will always write an element $\phi \in \mathbb{R}^{3N+1}$ as $\phi = (w, b, v, c)$, where $w, b, v \in \mathbb{R}^N$ and $c \in \mathbb{R}$. The realization of the network ϕ with ReLU activation is the function $f_\phi \in C(\mathbb{R}, \mathbb{R})$ given by

$$f_\phi(x) = c + \sum_{j=1}^{N} v_j \max\{w_j x + b_j, 0\}.$$

We suppose that the objective is to approximate an affine function on an interval $[T_0, T_1]$ in the L^2-norm. In other words, given $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$ and $T = (T_0, T_1) \in \mathbb{R}^2$, one tries to minimize the loss function $\mathcal{L}_{N,T,\mathcal{A}} \in C(\mathbb{R}^{3N+1}, \mathbb{R})$ given by

$$\mathcal{L}_{N,T,\mathcal{A}}(\phi) = \int_{T_0}^{T_1} (f_\phi(x) - \alpha x - \beta)^2 \, dx.$$

The purpose of the first half of this chapter is to classify the critical points of the loss function $\mathcal{L}_{N,T,\mathcal{A}}$. Since the ReLU function is not differentiable at 0, we work with the generalized gradient $\mathcal{G}_{N,T,\mathcal{A}} \colon \mathbb{R}^{3N+1} \to \mathbb{R}^{3N+1}$ of the loss obtained by taking right-hand partial derivatives;

$$(\mathcal{G}_{N,T,\mathcal{A}}(\phi))_k = \lim_{h \downarrow 0} \frac{\mathcal{L}_{N,T,\mathcal{A}}(\phi + h e_k) - \mathcal{L}_{N,T,\mathcal{A}}(\phi)}{h}$$

for all $k \in \{1, \dots, 3N+1\}$, where e_k is the k^{th} unit vector in $\mathbb{R}^{3N+1}$. The function $\mathcal{G}_{N,T,\mathcal{A}}$ is defined on the entire parameter space $\mathbb{R}^{3N+1}$ and agrees with the gradient of $\mathcal{L}_{N,T,\mathcal{A}}$ if the latter exists. We verify this and study regularity properties of $\mathcal{L}_{N,T,\mathcal{A}}$ more thoroughly in Section 2.5.

Definition 2.1. Let $N \in \mathbb{N}$ and $\mathcal{A}, T \in \mathbb{R}^2$. Then we call $\phi \in \mathbb{R}^{3N+1}$ a critical point of $\mathcal{L}_{N,T,\mathcal{A}}$ if $\mathcal{G}_{N,T,\mathcal{A}}(\phi) = 0$ and a saddle point if it is a critical point but not a local extremum.[1]

It can be shown that if ϕ is a critical point of $\mathcal{L}_{N,T,\mathcal{A}}$, then 0 belongs to the limiting sub-differential of $\mathcal{L}_{N,T,\mathcal{A}}$; see[2] [37, Prop. 2.12]. With Definition 2.1, it is not immediately clear whether all local extrema are critical points. However, we will show that this is the case by demonstrating that local extrema are points of differentiability of the loss function. In particular, Definition 2.1 is well-suited for our purposes. The next notion relates the outer bias, i.e., the coordinate c, to the target function $x \mapsto \alpha x + \beta$.

Definition 2.2. Let $N \in \mathbb{N}$, $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$, $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$, and $T = (T_0, T_1) \in \mathbb{R}^2$. Then we say that ϕ is $(T, \mathcal{A})$-centered if $c = \frac{\alpha}{2}(T_0 + T_1) + \beta$.

To motivate this definition, note that $\frac{\alpha}{2}(T_0+T_1)+\beta$ is the best constant L^2-approximation of the function $[T_0, T_1] \to \mathbb{R}$, $x \mapsto \alpha x + \beta$.

[1] We consider nonstrict local extrema, i.e. ϕ is a local minimum (maximum) of $\mathcal{L}_{N,T,\mathcal{A}}$ if $\mathcal{L}_{N,T,\mathcal{A}}(\phi) \leq \mathcal{L}_{N,T,\mathcal{A}}(\psi)$ $(\geq)$ for all ψ in an open neighborhood of ϕ, allowing equality $\mathcal{L}_{N,T,\mathcal{A}}(\phi) = \mathcal{L}_{N,T,\mathcal{A}}(\psi)$.

[2] In [37], the authors use a different generalization of the gradient, which can be obtained by taking left-hand partial derivatives. However, if $\mathcal{G}_{N,T,\mathcal{A}}$ is zero at some ϕ, then its left-hand analog is also zero at ϕ, so [37, Prop. 2.12] is applicable.

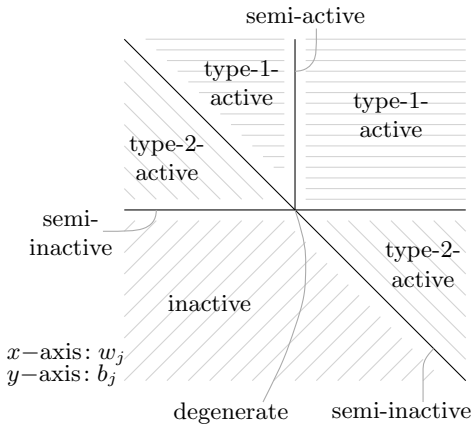

Figure 4.1: Illustration of the notions introduced in Definition 2.3: regions[4] with different types of a hidden neuron as seen in the (w_j, b_j)-plane.

2.2 Different types of hidden neurons

In this section, we introduce a few notions that describe how the different hidden neurons in a network are contributing to the realization function. In the definition below, we introduce sets I_j, which are defined such that $[T_0, T_1]\backslash I_j$ is the interval on which the output of the j^{th} hidden neuron is rendered zero by the ReLU activation.

Definition 2.3. Let $N \in \mathbb{N}$, $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$, $j \in \{1, \dots, N\}$, and $T_0, T_1 \in \mathbb{R}$ such that $T_0 < T_1$. Then, we denote by I_j the set given by $I_j = \{x \in [T_0, T_1] \colon w_j x + b_j \geq 0\}$, we say that the j^{th} hidden neuron of ϕ is

- *flat* if $v_j = 0$,
- *non-flat* if $v_j \neq 0$,
- *inactive* if $I_j = \emptyset$,
- *semi-inactive* if $\#I_j = 1$,
- *semi-active* if $w_j = 0 < b_j$,
- *active* if $w_j \neq 0 < b_j + \max_{k\in\{0,1\}} w_j T_k$,
- *type-1-active* if $w_j \neq 0 \leq b_j + \min_{k\in\{0,1\}} w_j T_k$,
- *type-2-active* if $\emptyset \neq I_j \cap (T_0, T_1) \neq (T_0, T_1)$,
- *degenerate* if $|w_j| + |b_j| = 0$,
- *non-degenerate* if $|w_j| + |b_j| > 0$,

and we say that $t \in \mathbb{R}$ is the breakpoint of the j^{th} hidden neuron of ϕ if $w_j \neq 0 = w_j t + b_j$.

Let us briefly motivate these notions. Every hidden neuron is exactly one of: inactive, semi-inactive, semi-active, active, or degenerate. Moreover, observe that I_j is always an interval.

For an inactive neuron, applying the ReLU activation function yields the constant zero function on $[T_0, T_1]$. The breakpoint t_j might not exist (if $w_j = 0$ and $b_j < 0$), or it might exist and lie outside of $[T_0, T_1]$ with $t_j < T_0$ if $w_j < 0$ and $t_j > T_1$ if $w_j > 0$. Note that inactivity is a stable condition in the sense that a small perturbation of an inactive neuron remains inactive.

Applying the ReLU activation to a semi-inactive neuron also yields the constant zero function on $[T_0, T_1]$. But in this case, a breakpoint must exist and be equal to one of the endpoints T_0, T_1 (which one depends on the sign of w_j similarly to the inactive case). However, a perturbation of a semi-inactive neuron may yield a (semi-)inactive or a type-2-active neuron; see Fig. 4.1. In this sense, semi-inactive neurons are boundary cases.

[4]Fig. 4.1 shows the case $T_0 = 0$, $T_1 = 1$. The general case is obtained by a shear transformation.

The realization of a semi-active neuron is also constant, but not necessarily zero since the corresponding interval I_j is $[T_0, T_1]$. As can be seen from Fig. 4.1, perturbing a semi-active neuron always yields a semi- or type-1-active neuron.

Non-flat active neurons provide a nonconstant contribution to the overall realization function. Note that a hidden neuron is active exactly if it is type-1- or type-2-active. These two types distinguish whether the breakpoint t_j, which exists in either case, lies outside or inside the interval (T_0, T_1) and, hence, whether the contribution of the neuron is affine (corresponding to $I_j = [T_0, T_1]$) or piecewise affine (corresponding to $I_j = [T_0, t_j]$ or $I_j = [t_j, T_1]$). Type-1 and type-2-active neurons both form two connected components in the (w_j, b_j)-plane; see Fig. 4.1. A perturbation of an active neuron remains active.

The case $w_j = 0 = b_j$ is called degenerate because it leads to problems with differentiability. Perturbing a degenerate neuron may yield any of the other types of neurons.

Lastly, a flat neuron also does not contribute to the overall realization, but the reason for this lies between the second and third layer and not between the first and second one, which is why this case deserves a separate notion.

2.3 Classification of the critical points of the loss function

Now, we are ready to provide a classification of the critical points of the loss function.

Theorem 2.4. *Let $N \in \mathbb{N}$, $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$, $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$, and $T = (T_0, T_1) \in \mathbb{R}^2$ satisfy $\alpha \neq 0$ and $0 \leq T_0 < T_1$. Then the following hold:*

(I) *ϕ is not a local maximum of $\mathcal{L}_{N,T,\mathcal{A}}$.*

(II) *If ϕ is a critical point or a local extremum of $\mathcal{L}_{N,T,\mathcal{A}}$, then $\mathcal{L}_{N,T,\mathcal{A}}$ is differentiable at ϕ with gradient $\nabla \mathcal{L}_{N,T,\mathcal{A}}(\phi) = 0$.*

(III) *ϕ is a non-global local minimum of $\mathcal{L}_{N,T,\mathcal{A}}$ if and only if ϕ is $(T, \mathcal{A})$-centered and, for all $j \in \{1, \dots, N\}$, the j^{th} hidden neuron of ϕ is*

- (a) *inactive,*
- (b) *semi-inactive with $I_j = \{T_0\}$ and $\alpha v_j > 0$, or*
- (c) *semi-inactive with $I_j = \{T_1\}$ and $\alpha v_j < 0$.*

(IV) *ϕ is a saddle point of $\mathcal{L}_{N,T,\mathcal{A}}$ if and only if ϕ is $(T, \mathcal{A})$-centered, ϕ does not have any type-1-active neurons, ϕ does not have any non-flat semi-active neurons, ϕ does not have any non-flat degenerate neurons, and exactly one of the following two items holds:*

- (a) *ϕ does not have any type-2-active neurons and there exists $j \in \{1, \dots, N\}$ such that the j^{th} hidden neuron of ϕ is*
  - (i) *flat semi-active,*
  - (ii) *semi-inactive with $I_j = \{T_0\}$ and $\alpha v_j \leq 0$,*
  - (iii) *semi-inactive with $I_j = \{T_1\}$ and $\alpha v_j \geq 0$, or*
  - (iv) *flat degenerate.*
- (b) *There exists $n \in \{2, 4, 6, \dots\}$ such that*

$$\bigcup_{j \in \{1,\dots,N\},\, w_j \neq 0} \left\{ -\frac{b_j}{w_j} \right\} \cap (T_0, T_1) = \bigcup_{i=1}^{n} \left\{ T_0 + \frac{i(T_1 - T_0)}{n+1} \right\}$$

and, for all $j \in \{1, \dots, N\}$, $i \in \{1, \dots, n\}$ *with* $w_j \neq 0 = b_j + w_j(T_0 + \frac{i(T_1 - T_0)}{n+1})$, *it holds that* $\operatorname{sign}(w_j) = (-1)^{i+1}$ *and*

$$\sum_{k \in \{1,\dots,N\},\, w_k \neq 0 = b_k + w_k(T_0 + \frac{i(T_1-T_0)}{n+1})} v_k w_k = \frac{2\alpha}{n+1}.$$

(V) *If* ϕ *is a non-global local minimum of* $\mathcal{L}_{N,T,\mathcal{A}}$ *or a saddle point of* $\mathcal{L}_{N,T,\mathcal{A}}$ *without type-2-active neurons, then* $f_\phi(x) = \frac{\alpha}{2}(T_0 + T_1) + \beta$ *for all* $x \in [T_0, T_1]$.

(VI) *If* ϕ *is a saddle point of* $\mathcal{L}_{N,T,\mathcal{A}}$ *with at least one type-2-active neuron, then there exists* $n \in \{2, 4, 6, \dots\}$ *such that* $n \leq N$ *and*

$$f_\phi(x) = \alpha x + \beta - \frac{(-1)^i \alpha}{n+1}\Big(x - T_0 - \frac{(i + \frac{1}{2})(T_1 - T_0)}{n+1}\Big)$$

for all $i \in \{0, \dots, n\}$, $x \in [T_0 + \frac{i(T_1-T_0)}{n+1}, T_0 + \frac{(i+1)(T_1-T_0)}{n+1}]$.

Theorem 2.4.(IV.b) says that the set of breakpoints of all type-2-active neurons agrees with the set of n equally spaced points $T_0 < q_1 < \cdots < q_n < T_1$. Furthermore, for any type-2-active neuron with breakpoint q_i, the sign of the coordinate w is given by $(-1)^{i+1}$. Lastly, the sum of $v_k w_k$, where k ranges over all type-2-active neurons with breakpoint q_i, is equal to $\frac{2\alpha}{n+1}$. The term $v_k w_k$ is the contribution of the k^{th} hidden neuron to the slope of the realization.

Remark 2.5. Note that, by Theorem 2.4.(II), all local extrema and all critical points of $\mathcal{L}_{N,T,\mathcal{A}}$, which we defined as zeros of $\mathcal{G}_{N,T,\mathcal{A}}$, are actually critical points of $\mathcal{L}_{N,T,\mathcal{A}}$ in the classical sense, i.e. points of differentiability of $\mathcal{L}_{N,T,\mathcal{A}}$ with vanishing gradient. In particular, the classification in Theorem 2.4 turns out to be a classification of the critical points in the classical sense as well.

Remark 2.6. Gradient Descent-type algorithms typically use generalized gradients to train ReLU networks. For instance, they might compute $\mathcal{G}$, its left-hand analog, the average of the two, or quantities obtained by artificially defining the derivative of the ReLU function at 0. For each of these versions, a similar classification of critical points could be derived.

Theorem 2.4.(V) shows that any non-global local minimum has the constant realization $\frac{\alpha}{2}(T_0 + T_1) + \beta$. In particular, there is only one value that the loss function can take at non-global local minima. Similarly, it follows from Theorem 2.4.(VI) that a saddle point can lead to exactly one of $\lfloor N/2 \rfloor + 1$ possible loss values.

Corollary 2.7. *Let* $N \in \mathbb{N}$, $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$, *and* $T = (T_0, T_1) \in \mathbb{R}^2$ *satisfy* $0 \leq T_0 < T_1$, *and assume that* $\phi \in \mathbb{R}^{3N+1}$ *is a critical point of* $\mathcal{L}_{N,T,\mathcal{A}}$. *Then the following hold:*

(i) *If* ϕ *is a non-global local minimum of* $\mathcal{L}_{N,T,\mathcal{A}}$, *then* $\mathcal{L}_{N,T,\mathcal{A}}(\phi) = \frac{1}{12}\alpha^2(T_1 - T_0)^3$.

(ii) *If* ϕ *is a saddle point of* $\mathcal{L}_{N,T,\mathcal{A}}$, *then there exists* $n \in \{0, 2, 4, \dots\}$ *such that* $n \leq N$ *and* $\mathcal{L}_{N,T,\mathcal{A}}(\phi) = \frac{1}{12(n+1)^4}\alpha^2(T_1 - T_0)^3$.

Formally, Corollary 2.7 only follows from Theorem 2.4 for $\alpha \neq 0$. But for $\alpha = 0$ it holds trivially since for constant target functions there exist no critical points other than global minima; see [14].

2.4 Ingredients for the proof of the classification

As a first step, let us provide a simple argument to establish Theorem 2.4.(I).

Lemma 2.8. *Let $N \in \mathbb{N}$, $\mathcal{A} \in \mathbb{R}^2$, and $T = (T_0, T_1) \in \mathbb{R}^2$ satisfy $T_0 < T_1$. Then $\mathcal{L}_{N,T,\mathcal{A}}$ does not have any local maxima.*

Proof. Write $\mathcal{A} = (\alpha, \beta)$. The lemma directly follows from the simple fact that

$$\mathcal{L}_{N,T,\mathcal{A}}(w, b, v, c) = \int_{T_0}^{T_1} \Big(c + \sum_{j=1}^{N} v_j \max\{w_j x + b_j, 0\} - \alpha x - \beta \Big)^2 dx$$

is strictly convex in c. ∎

As a consequence of this lemma, whenever we want to show that a critical point ϕ is a saddle point, it suffices to show that it is not a local minimum, that is, it suffices to show that, in every neighborhood of ϕ, $\mathcal{L}$ attains a value that is below $\mathcal{L}(\phi)$.

Remark 2.9. The previous proof only used linearity of the realization function in the c-coordinate and strict convexity of the square function. In particular, the same argument shows that the square loss never has local maxima regardless of the target function, the activation function, and the architecture of the network.

Let us now provide a sketch of the proofs to come. Instead of proving Theorem 2.4 directly, we first assume that the affine target function is the identity on the interval $[0, 1]$, corresponding to the special case $T_0 = \beta = 0$ and $T_1 = \alpha = 1$ in Theorem 2.4. Afterwards, we will verify that the general case can always be reduced to this one. For convenience of notation, we assume the following convention to hold throughout the remainder of Section 2.

Setting 2.10. Fix $N \in \mathbb{N}$ and denote $\mathcal{L} = \mathcal{L}_{N,(0,1),(1,0)}$ and $\mathcal{G} = \mathcal{G}_{N,(0,1),(1,0)}$. We say that a network $\phi \in \mathbb{R}^{3N+1}$ is centered if it is $((0, 1), (1, 0))$-centered.

The generalized gradient $\mathcal{G}$ was defined in terms of the right-hand partial derivatives of $\mathcal{L}$. These are given by

$$\frac{\partial^+}{\partial w_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} x(f_\phi(x) - x)dx, \qquad \frac{\partial^+}{\partial v_j}\mathcal{L}(\phi) = 2 \int_{I_j} (w_j x + b_j)(f_\phi(x) - x)dx,$$
$$\frac{\partial^+}{\partial b_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} (f_\phi(x) - x)dx, \qquad \frac{\partial^+}{\partial c}\mathcal{L}(\phi) = 2 \int_0^1 (f_\phi(x) - x)dx.$$

Regularity properties of the loss function will be discussed in detail in the next section. We will see then that these right-hand partial derivatives are proper partial derivatives if the j^{th} hidden neuron is flat or non-degenerate. If these partial derivatives are zero, then we encounter the system of equations

$$\begin{aligned} 0 &= 2v_j \int_{I_j} x(f_\phi(x) - x)dx, \\ 0 &= 2v_j \int_{I_j} (f_\phi(x) - x)dx, \\ 0 &= 2 \int_{I_j} (w_j x + b_j)(f_\phi(x) - x)dx, \\ 0 &= 2 \int_0^1 (f_\phi(x) - x)dx, \end{aligned} \tag{2.1}$$

from which we deduce that any non-flat non-degenerate neuron of a critical point or local extremum ϕ satisfies

$$\int_{I_j} (f_\phi(x) - x)dx = 0 = \int_{I_j} x(f_\phi(x) - x)dx. \tag{2.2}$$

This simple observation will be used repeatedly in the proof of Theorem 2.4. Moreover, for a type-1-active neuron (for which $I_j = [0, 1]$), (2.2) is even satisfied if the neuron is flat as can be seen from the third and fourth line of (2.1). Here is an example of how (2.2) can be employed: note that any affine function $f\colon [0, 1] \to \mathbb{R}$ satisfying

$$\int_0^1 (f(x) - x)dx = 0 = \int_0^1 x(f(x) - x)dx \tag{2.3}$$

necessarily equals the identity on $[0, 1]$. Thus, if ϕ is a critical point or local extremum of $\mathcal{L}$ for which f_ϕ is affine and if ϕ admits a type-1-active or non-flat semi-active neuron (so that $I_j = [0, 1]$), then we obtain from (2.2) that ϕ is a global minimum. If f_ϕ is not affine, we will be able to develop similar arguments for each affine piece of f_ϕ. In this case, we will obtain a system of equations from (2.1) that intricately describes the combinatorics of the realization function.

2.5 Differentiability of the loss function

Since the ReLU function is not differentiable at 0, the loss function is not everywhere differentiable. However, a simple argument establishes that $\mathcal{L}$ is differentiable at any of its global minima as the following lemma shows.

Lemma 2.11. *Let $\phi \in \mathbb{R}^{3N+1}$. If $f_\phi(x) = x$ for all $x \in [0, 1]$, then $\mathcal{L}$ is differentiable at ϕ.*

Proof. It is well known that the realization function $\mathbb{R}^{3N+1} \to C([0, 1], \mathbb{R})$, $\phi \mapsto f_\phi|_{[0,1]}$ is locally Lipschitz continuous if $C([0, 1], \mathbb{R})$ is equipped with the supremums norm; see, e.g., [104]. Thus, there is a constant $L > 0$ depending only on N and ϕ with $|f_{\phi+\psi}(x) - f_\phi(x)| \leq L\|\psi\|$ uniformly on $[0, 1]$ for all ψ sufficiently close to ϕ. Then

$$\frac{\mathcal{L}(\phi + \psi) - \mathcal{L}(\phi)}{\|\psi\|} = \frac{1}{\|\psi\|} \int_0^1 (f_{\phi+\psi}(x) - f_\phi(x))^2 dx \leq L^2\|\psi\|,$$

which shows that $\mathcal{L}$ is differentiable at ϕ. ∎

The next result shows that there even are regions in the parameter space where $\mathcal{L}$ is infinitely often differentiable in spite of the ReLU activation.

Lemma 2.12. *The loss function $\mathcal{L}$ is everywhere analytic in (v, c). Moreover, if the j^{th} hidden neuron of $\phi \in \mathbb{R}^{3N+1}$ is inactive, semi-active, or type-1-active with breakpoint neither 0 nor 1 for some $j \in \{1, \dots, N\}$, then $\mathcal{L}$ is also analytic in (w_j, b_j, v, c) in a neighborhood of ϕ, and mixed partial derivatives of any order can be obtained by differentiating under the integral. In particular,*

$$\frac{\partial}{\partial w_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} x(f_\phi(x) - x)dx, \qquad \frac{\partial}{\partial v_j}\mathcal{L}(\phi) = 2\int_{I_j} (w_j x + b_j)(f_\phi(x) - x)dx,$$
$$\frac{\partial}{\partial b_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} (f_\phi(x) - x)dx, \qquad \frac{\partial}{\partial c}\mathcal{L}(\phi) = 2\int_0^1 (f_\phi(x) - x)dx.$$

Proof. For the first part, note that $\mathcal{L}$ is a polynomial in the coordinates (v, c). Secondly, assume that the j^{th} hidden neuron of $\phi^0 \in \mathbb{R}^{3N+1}$ is inactive. Then for all ϕ in a sufficiently small neighborhood of ϕ^0 and all $x \in [0,1]$ we have $\max\{w_j x + b_j, 0\} = 0$. Hence, $\mathcal{L}$ is constant in the coordinates (w_j, b_j) near ϕ^0 and it is a polynomial in (w_j, b_j, v, c). Thirdly, assume that the j^{th} hidden neuron of ϕ^0 is semi-active or type-1-active with breakpoint neither 0 nor 1. Then for all ϕ in a sufficiently small neighborhood of ϕ^0 and all $x \in [0,1]$ we have $\max\{w_j x + b_j, 0\} = w_j x + b_j$. In particular, $\mathcal{L}$ is a polynomial in the coordinates (w_j, b_j, v, c) near ϕ^0. The statement about differentiating under the integral follows from dominated convergence. ∎

In regions of the parameter space not covered by Lemma 2.12, we cannot guarantee as much regularity of the loss function, but we can still hope for differentiability. Indeed, we already noted in the proof of Lemma 2.11 that the realization function $\mathbb{R}^{3N+1} \to C([0,1], \mathbb{R})$, $\phi \mapsto f_\phi|_{[0,1]}$ is locally Lipschitz continuous. So, it follows from Rademacher's theorem that $\mathcal{G}$ is, in fact, equal to the true gradient $\nabla\mathcal{L}$ of $\mathcal{L}$ almost everywhere. In the next result, we obtain insights about the measure-zero set on which $\mathcal{G}$ may not be the true gradient.

Lemma 2.13. *For all $j \in \{1, \dots, N\}$, the right-hand partial derivatives $\partial^+\mathcal{L}(\phi)/\partial w_j$ and $\partial^+\mathcal{L}(\phi)/\partial b_j$ exist everywhere and are given by*

$$\frac{\partial^+}{\partial w_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} x(f_\phi(x) - x)dx \quad \text{and} \quad \frac{\partial^+}{\partial b_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} (f_\phi(x) - x)dx.$$

Moreover, if the j^{th} hidden neuron is flat or non-degenerate, then $\mathcal{L}$ is differentiable in (w_j, b_j, v, c) and, in particular, the right-hand partial derivatives $\partial^+\mathcal{L}(\phi)/\partial w_j$ and $\partial^+\mathcal{L}(\phi)/\partial b_j$ are proper partial derivatives.

Proof. Let $\phi \in \mathbb{R}^{3N+1}$ be arbitrary and denote by ϕ_h, $h = (h^1, h^2) \in \mathbb{R}^2$, the network with the same coordinates as ϕ except in the j^{th} hidden neuron, where ϕ_h has coordinates $w_j + h^1$ and $b_j + h^2$. We use the notation I_j^h for the interval I_j associated to ϕ_h and denote

$$\varepsilon = \mathcal{L}(\phi_h) - \mathcal{L}(\phi) - 2v_j h^1 \int_{I_j} x(f_\phi(x) - x)dx - 2v_j h^2 \int_{I_j} (f_\phi(x) - x)dx.$$

The proof is complete if we can show that ε goes to zero faster than (h^1, h^2). To do that, we estimate the two terms of the last line of

$$\begin{aligned}
\varepsilon &= \int_0^1 (f_{\phi_h}(x) - f_\phi(x))^2 dx \\
&\quad + 2\int_0^1 (f_{\phi_h}(x) - f_\phi(x))(f_\phi(x) - x)dx - 2v_j \int_{I_j} (h^1 x + h^2)(f_\phi(x) - x)dx \\
&= \int_0^1 (f_{\phi_h}(x) - f_\phi(x))^2 dx + 2v_j \int_0^1 (w_j x + b_j + h^1 x + h^2)(f_\phi(x) - x)(\mathbb{1}_{I_j^h}(x) - \mathbb{1}_{I_j}(x))dx.
\end{aligned}$$

To control the first term, we use local Lipschitz continuity of the realization function, which yields a constant $L > 0$ depending only on ϕ so that $|f_{\phi_h}(x) - f_\phi(x)| \leq L(|h^1| + |h^2|)$ uniformly on $[0,1]$ for all sufficiently small h. To estimate the second term, we note that the absolute

value of $\mathbb{1}_{I_j^h} - \mathbb{1}_{I_j}$ is the indicator function of the symmetric difference $I_j \triangle I_j^h$. By definition of these sets, we obtain the bound $|w_j x + b_j| \leq |h^1 x + h^2|$ for any $x \in I_j \triangle I_j^h$. This yields

$$\frac{|\varepsilon|}{|h^1| + |h^2|} \leq L^2(|h^1| + |h^2|) + 4|v_j| \int_0^1 |f_\phi(x) - x| \mathbb{1}_{I_j \triangle I_j^h}(x) dx.$$

The term $L^2(|h^1| + |h^2|)$ vanishes as $h \to 0$. We need to argue that the second term also vanishes as $h \to 0$. If the j^{th} hidden neuron is flat, then the second term is trivially zero. On the other hand, if the j^{th} hidden neuron is non-degenerate, then the Lebesgue measure of $I_j \triangle I_j^h$ tends to zero as $h \to 0$. Thus, in this case, the integral also vanishes as $h \to 0$. If the j^{th} hidden neuron is non-flat degenerate, then we consider the directional derivatives from the right, i.e. with $h^1, h^2 \downarrow 0$. But then $I_j = [0,1] = I_j^h$, so $\mathbb{1}_{I_j \triangle I_j^h}$ is constantly zero. ■

It is well known that a multivariate function is continuously differentiable if it has continuous partial derivatives. The following result is a slight extension for the loss function $\mathcal{L}$.

Lemma 2.14. *The loss function $\mathcal{L}$ is continuously differentiable on the set of networks without degenerate neurons. In addition, $\mathcal{L}$ is differentiable at networks without non-flat degenerate neurons.*

Proof. The preceding two results established existence of all partial derivatives of first order at networks without degenerate neurons. Furthermore, these partial derivatives are continuous in the network parameters. This is clear for (v, c) and it also holds for (w, b) because the endpoints of I_j vary continuously in w_j and b_j as long as not both are zero. This concludes the first statement.

To prove that $\mathcal{L}$ is still differentiable if flat degenerate neurons appear, assume without loss of generality that the first $M \leq N$ hidden neurons of $\phi \in \mathbb{R}^{3N+1}$ are flat degenerate and the remaining $N - M$ hidden neurons are non-degenerate. Denote by $\phi_1 \in \mathbb{R}^{3M+1}$ the network comprised of the first M hidden neurons of ϕ (with zero outer bias) and by $\phi_2 \in \mathbb{R}^{3(N-M)+1}$ the network comprised of the last $N - M$ hidden neurons. We write $\mathcal{L}_{N-M}$ for the loss defined on networks with $N - M$ hidden neurons. Then, for any perturbation $\phi_h = \phi + h \in \mathbb{R}^{3N+1}$ of ϕ with the same decomposition into its first M and last $N - M$ hidden neurons, we can write $f_{\phi_h}(x) = f_{\phi_{1,h}}(x) + f_{\phi_{2,h}}(x)$ and, hence,

$$\mathcal{L}(\phi_h) = \int_0^1 f_{\phi_{1,h}}(x)^2 dx + 2 \int_0^1 f_{\phi_{1,h}}(x)(f_{\phi_{2,h}}(x) - x) dx + \mathcal{L}_{N-M}(\phi_{2,h}).$$

Since the first M hidden neurons of ϕ are flat degenerate, $f_{\phi_{1,h}}(x)$ is given by

$$f_{\phi_{1,h}}(x) = \sum_{j=1}^{M} h_{j+2N} \max\{h_j x + h_{j+N}, 0\}.$$

In particular, $f_{\phi_{1,h}}(x)/\|h\| \to 0$ uniformly in $x \in [0,1]$ as $h \to 0$. Denote by $\tilde{h}$ the last $3(N-M)$ components of h. Since ϕ_2 has only non-degenerate neurons, $\mathcal{L}_{N-M}$ is differentiable at ϕ_2 with some gradient A. Using that the first M hidden neurons of ϕ do not contribute

to its realization and, hence, $\mathcal{L}(\phi) = \mathcal{L}_{N-M}(\phi_2)$, we find

$$\lim_{h\to 0} \frac{\mathcal{L}(\phi_h) - \mathcal{L}(\phi) - A\tilde{h}}{\|h\|} = \lim_{h\to 0} \frac{\mathcal{L}_{N-M}(\phi_{2,h}) - \mathcal{L}_{N-M}(\phi_2) - A\tilde{h}}{\|\tilde{h}\|} \frac{\|\tilde{h}\|}{\|h\|}$$
$$+ \lim_{h\to 0} \frac{1}{\|h\|} \Big(\int_0^1 f_{\phi_{1,h}}(x)^2 dx + 2 \int_0^1 f_{\phi_{1,h}}(x)(f_{\phi_{2,h}}(x) - x)dx \Big) = 0.$$

This proves differentiability of $\mathcal{L}$ at ϕ. ■

So far, we have seen that, in some regions of the parameter space, the loss is differentiable while in others it may not be. In the following, we show that, for type-2-active neurons, one even has twice continuous differentiability.

Lemma 2.15. *Let $i, j \in \{1, \dots, N\}$. If the i^{th} and j^{th} hidden neuron of $\phi \in \mathbb{R}^{3N+1}$ are type-2-active, then $\mathcal{L}$ is twice continuously differentiable in $(w_i, w_j, b_i, b_j, v, c)$ in a neighborhood of ϕ in $\mathbb{R}^{3N+1}$.*

Proof. We established twice continuous differentiability of $\mathcal{L}$ in (v, c) in Lemma 2.12. Suppose the i^{th} and j^{th} hidden neuron of $\phi^0 = (w^0, b^0, v^0, c^0) \in \mathbb{R}^{3N+1}$ are type-2-active. Since a small perturbation of a type-2-active neuron remains type-2-active and since a type-2-active neuron is non-degenerate, it follows from Lemma 2.13 that $\mathcal{L}$ is differentiable in (w_j, b_j) in a neighborhood $U \subseteq \mathbb{R}^{3N+1}$ of ϕ^0 with partial derivatives

$$\frac{\partial}{\partial w_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} x(f_\phi(x) - x)dx \quad \text{and} \quad \frac{\partial}{\partial b_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} (f_\phi(x) - x)dx$$

for any $\phi = (w, b, v, c) \in U$. Because the j^{th} hidden neuron is assumed to be type-2-active, the interval I_j^0 is exactly $[0, t_j^0]$ or $[t_j^0, 1]$ for the breakpoint $t_j^0 = -b_j^0/w_j^0 \in (0, 1)$. Assume $I_j^0 = [0, t_j^0]$ as the other case is dealt with analogously. By shrinking U if necessary, we therefore integrate over $[0, -b_j/w_j]$ in the above partial derivatives for all $\phi = (w, b, v, c) \in U$. In particular, the integration boundaries vary smoothly in (w_j, b_j) in U. So, it follows from Leibniz' rule that these partial derivatives are continuously differentiable with respect to (w_j, b_j). Furthermore, since $t_j = -b_j/w_j$ does not depend on (w_i, b_i, v, c), it follows from dominated convergence that $\partial\mathcal{L}(\phi)/\partial w_j$ and $\partial\mathcal{L}(\phi)/\partial b_j$ are also differentiable with respect to (w_i, b_i, v, c). The mixed partial derivative with respect to w_i and w_j is given by

$$\frac{\partial}{\partial w_i}\frac{\partial}{\partial w_j}\mathcal{L}(\phi) = 2v_j \int_{I_j} x \frac{\partial}{\partial w_i} f_\phi(x)dx = 2v_i v_j \int_{I_i \cap I_j} x^2 dx.$$

That the i^{th} and j^{th} hidden neuron are type-2-active ensures that $\int_{I_i \cap I_j} x^2 dx$ is continuous in (w_i, w_j, b_i, b_j) and, hence, that $\partial^2\mathcal{L}(\phi)/(\partial w_i \partial w_j)$ is continuous in $(w_i, w_j, b_i, b_j, v, c)$. Analogous considerations show that all mixed partial derivatives with respect to w_i, w_j, b_i, b_j, v, c up to second order exist and are continuous. Thus, $\mathcal{L}$ restricted to $(w_i, w_j, b_i, b_j, v, c)$ is twice continuously differentiable in a neighborhood of ϕ^0. ■

Remark 2.16. We mentioned in Remark 2.5 that all critical points and local extrema of $\mathcal{L}$ are actually proper critical points and, hence, the classification actually does not deal with points of non-differentiability. Furthermore, by modifying the Gradient Descent algorithm and the initialization in an appropriate way, one can ensure that the trajectories of the algorithm avoid any points of non-differentiability; see [128] and also the appendix in [20]. Nonetheless, to formally prove the classification, including that all critical points are proper, an extensive regularity analysis of the loss function as done in this section is necessary.

2.6 Critical points of the loss function with affine realization

In this and the next section, we develop the building blocks necessary for proving the main result. The first lemma establishes one direction of the equivalence in Theorem 2.4.(III).

Lemma 2.17. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is centered and all of its hidden neurons satisfy one of the properties* (III.a)-(III.c) *in Theorem 2.4. Then ϕ is a local minimum of $\mathcal{L}$.*

Proof. Denote by $J_0 \subseteq \{1, \dots, N\}$ the set of those hidden neurons of ϕ that satisfy Theorem 2.4.(III.b), and, likewise, denote by $J_1 \subseteq \{1, \dots, N\}$ the set of those hidden neurons of ϕ that satisfy Theorem 2.4.(III.c). Write $\phi = (w^0, b^0, v^0, c^0)$ and consider $\psi = (w, b, v, c) \in U$ in a small neighborhood U of ϕ. Since a small perturbation of an inactive neuron remains inactive, we have for all $\psi \in U$ and every $x \in [0, 1]$ that

$$f_\psi(x) = c + \sum_{j \in J_0 \cup J_1} v_j \max\{w_j x + b_j, 0\}$$

if U is small enough. Moreover, for any $j \in J_0$ and $\psi \in U$, note that $\max\{w_j x + b_j, 0\} = 0$ for all $x \in [1/4, 1]$. Similarly, $\max\{w_j x + b_j, 0\} = 0$ for all $x \in [0, 3/4]$ if $j \in J_1$. Since we also know $v_j^0 > 0$ for all $j \in J_0$ and $v_j^0 < 0$ for all $j \in J_1$, we find that the realization of $\psi \in U$ satisfies

$$f_\psi(x) = \begin{cases} c + \sum_{j \in J_0} v_j \max\{w_j x + b_j, 0\} \geq c & \text{if } x \in [0, 1/4] \\ c & \text{if } x \in [1/4, 3/4] \\ c + \sum_{j \in J_1} v_j \max\{w_j x + b_j, 0\} \leq c & \text{if } x \in [3/4, 1] \end{cases}$$

for sufficiently small U. In particular, it follows that $|f_\psi(x) - x| \geq |c - x|$ for all $x \in [0, 1]$ and, because ϕ is centered, that

$$\mathcal{L}(\psi) \geq \int_0^1 (c - x)^2 dx \geq \int_0^1 (\tfrac{1}{2} - x)^2 dx = \mathcal{L}(\phi).$$

Thus, ϕ is a local minimum. ∎

The proof of the next lemma revolves, for the most part, around the argument (2.3), presented in Section 2.4. The last statement of the lemma paired with Lemma 2.14 shows that saddle points with an affine realization are also points of differentiability of $\mathcal{L}$.

Lemma 2.18. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}$ but not a global minimum and that f_ϕ is affine on $[0, 1]$. Then ϕ is centered and does not have any active or non-flat semi-active neurons, so, in particular, $f_\phi \equiv 1/2$. Moreover, if ϕ is a saddle point, then it also does not have any non-flat degenerate neurons.*

Proof. We know from Lemma 2.13 that $\mathcal{L}$ is differentiable in those coordinates that correspond to non-degenerate neurons and its partial derivatives must vanish at ϕ. Thus, the argument using (2.3) shows that ϕ does not have any type-1-active or non-flat semi-active neurons. If ϕ had a non-flat type-2-active neuron, say the j^{th}, then we could, using the same argument with I_j in place of $[0, 1]$, conclude that $f_\phi(x) = x$ on I_j. But since f_ϕ was assumed to be affine, this could only be true if ϕ were a global minimum. Having no type-1-active or non-flat type-2-active neurons, f_ϕ must be constant. By the fourth equation of (2.1), this constant is $1/2$, so ϕ is centered.

Next, suppose that the j^{th} hidden neuron is flat type-2-active. In particular, $I_j = [0, t_j]$ or $I_j = [t_j, 1]$, where $t_j = -b_j/w_j \in (0,1)$ is the breakpoint. After dividing by $2w_j$, the integral in the third equation of (2.1) evaluates to

$$0 = \int_{I_j} (x - t_j)(\tfrac{1}{2} - x)dx = \left\{ \begin{matrix} -\frac{1}{6}t_j^2(\frac{3}{2} - t_j) & \text{if } I_j = [0, t_j] \\ -\frac{1}{6}(1 - t_j)^2(t_j + \frac{1}{2}) & \text{if } I_j = [t_j, 1] \end{matrix} \right\} \neq 0,$$

yielding a contradiction. Lastly, suppose ϕ is a saddle point. If there were a non-flat degenerate neuron, then $\mathcal{G}(\phi) = 0$ would imply $0 = \int_0^1 x(f_\phi(x) - x)dx$. But since we know that $f_\phi(x) \equiv 1/2$, this cannot be. ■

The next lemma serves as the basis of Theorem 2.4.(IV.a). However, note that we also consider the possibility of a non-flat degenerate neuron, whereas Theorem 2.4.(IV.a.iv) requires the degenerate neuron to be flat. This generalization is needed in the proof of Theorem 2.4.(III), which will be given later by way of contradiction. In addition, Lemma 2.19 shows that non-global local minima with an affine realization cannot have non-flat degenerate neurons and, hence, are points of differentiability of $\mathcal{L}$ by Lemma 2.14. Together with the preceding lemma and Lemmas 2.11 and 2.14, we conclude that all critical points and local extrema with an affine realization are points of differentiability.

Lemma 2.19. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}$ but not a global minimum and that f_ϕ is affine on $[0,1]$. Suppose further that at least one of its hidden neurons satisfies one of the properties (IV.a.i)-(IV.a.iii) in Theorem 2.4 or is degenerate. Then ϕ is a saddle point.*

Proof. Since, by Lemma 2.8, $\mathcal{L}$ cannot have any local maxima, it is enough to show that $\mathcal{L}$ is strictly decreasing along some direction starting from ϕ. First, assume that the j^{th} hidden neuron of ϕ is flat semi-active. Then Lemma 2.12 asserts smoothness of the loss in the coordinates of the j^{th} hidden neuron and

$$\begin{aligned} \frac{\partial}{\partial w_j}\frac{\partial}{\partial w_j}\mathcal{L}(\phi) &= 2v_j \int_0^1 x \frac{\partial}{\partial w_j} f_\phi(x)dx = 0, \\ \frac{\partial}{\partial v_j}\frac{\partial}{\partial w_j}\mathcal{L}(\phi) &= 2v_j \int_0^1 x \frac{\partial}{\partial v_j} f_\phi(x)dx + 2\int_0^1 x(f_\phi(x) - x)dx \\ &= 2\int_0^1 x(f_\phi(x) - x)dx =: R, \\ \frac{\partial}{\partial v_j}\frac{\partial}{\partial v_j}\mathcal{L}(\phi) &= 2\int_0^1 (w_j x + b_j)\frac{\partial}{\partial v_j} f_\phi(x)dx =: S, \end{aligned}$$

where we used that the j^{th} hidden neuron is flat. Since $2\int_0^1 (f_\phi(x) - x)dx = \frac{\partial}{\partial c}\mathcal{L}(\phi) = 0$, we must have $R \neq 0$ for otherwise ϕ would be a global minimum by the argument (2.3). This yields

$$\det \begin{pmatrix} \frac{\partial}{\partial w_j}\frac{\partial}{\partial w_j}\mathcal{L}(\phi) & \frac{\partial}{\partial w_j}\frac{\partial}{\partial v_j}\mathcal{L}(\phi) \\ \frac{\partial}{\partial v_j}\frac{\partial}{\partial w_j}\mathcal{L}(\phi) & \frac{\partial}{\partial v_j}\frac{\partial}{\partial v_j}\mathcal{L}(\phi) \end{pmatrix} = \det \begin{pmatrix} 0 & R \\ R & S \end{pmatrix} = -R^2 < 0.$$

In particular, this matrix must have a strictly negative eigenvalue, and a second order expansion of the loss restricted to (w_j, v_j) shows that $\mathcal{L}$ is strictly decreasing along the direction of an eigenvector associated to this negative eigenvalue.

Next, assume that the j^{th} hidden neuron is semi-inactive with $I_j = \{0\}$ and $v_j \leq 0$ (case one) or that it is degenerate with $v_j \leq 0$ (case two). In either case, note that $b_j = 0$ and consider the perturbation $\phi_s = (w^s, b^s, v^s, c^s)$, $s \in [0,1]$, of $\phi = \phi_0$ given by $w_j^s = w_j - s$, $b_j^s = -sw_j^s$, and $v_j^s = v_j - s$ (all other coordinates coincide with those of ϕ). Note that we have $w_j^s < 0$ and $v_j^s < 0$ for all $s \in (0,1]$ in both cases. For simplicity, denote $a^s = v_j^s w_j^s$. By Lemma 2.18, we already know that ϕ is centered and does not have any active or non-flat semi-active neurons. Thus, for every $s, x \in [0,1]$, we can write

$$f_{\phi_s}(x) = c + v_j^s \max\{w_j^s x + b_j^s, 0\} = c + v_j^s \max\{w_j^s(x-s), 0\} = \tfrac{1}{2} + a^s(x-s)\mathbb{1}_{[0,s]}(x).$$

Using this formula, we have for all $s \in [0,1]$

$$\begin{aligned}
\mathcal{L}(\phi_s) - \mathcal{L}(\phi) &= \int_0^s [a^s(x-s)]^2 dx - \int_0^s 2a^s(x-s)(x-\tfrac{1}{2})dx \\
&= \tfrac{1}{3}a^s(a^s+1)s^3 - \tfrac{1}{2}a^s s^2 \\
&= \begin{cases} -\frac{1}{2}v_j w_j s^2 + \mathcal{O}(s^3) & \text{if } w_j \neq 0 \neq v_j \\ -\frac{1}{2}|v_j + w_j| s^3 + \mathcal{O}(s^4) & \text{if } w_j \neq 0 = v_j \text{ or } w_j = 0 \neq v_j \\ -\frac{1}{2}s^4 + \mathcal{O}(s^5) & \text{if } w_j = 0 = v_j, \end{cases}
\end{aligned}$$

which is strictly negative for small $s > 0$. Hence, ϕ is a saddle point.

Lastly, assume that the j^{th} hidden neuron is semi-inactive with $I_j = \{1\}$ and $v_j \geq 0$ (case one) or that it is degenerate with $v_j > 0$ (case two). This is dealt with the same way as the previous step. Let $\phi_s \in \mathbb{R}^{3N+1}$, $s \in [0,1]$, be given by $w_j^s = w_j + s$, $b_j^s = -(1-s)w_j^s$, and $v_j^s = v_j + s$. This time, we have $w_j^s > 0$ and $a^s = v_j^s w_j^s > 0$ for all $s \in (0,1]$ in both cases. The realization of ϕ_s on $[0,1]$ is given for all $s, x \in [0,1]$ by

$$f_{\phi_s}(x) = c + v_j^s \max\{w_j^s x + b_j^s, 0\} = \tfrac{1}{2} + a^s(x-1+s)\mathbb{1}_{[1-s,1]}(x).$$

Essentially by the same computation as in the previous step,

$$\begin{aligned}
\mathcal{L}(\phi_s) - \mathcal{L}(\phi) &= \tfrac{1}{3}a^s(a^s+1)s^3 - \tfrac{1}{2}a^s s^2 \\
&= \begin{cases} -\frac{1}{2}v_j w_j s^2 + \mathcal{O}(s^3) & \text{if } w_j \neq 0 \neq v_j \\ -\frac{1}{2}(v_j + w_j) s^3 + \mathcal{O}(s^4) & \text{if } w_j \neq 0 = v_j \text{ or } w_j = 0 \neq v_j \\ -\frac{1}{2}s^4 + \mathcal{O}(s^5) & \text{if } w_j = 0 = v_j, \end{cases}
\end{aligned}$$

from which we conclude that ϕ is a saddle point. ■

This finishes the treatment of the affine case, and we now tend to the more involved non-affine case in the next section.

2.7 Critical points of the loss function with non-affine realization

The following lemma is the main tool for this section. It generalizes the argument (2.3) that we presented in Section 2.4; see Lemma 2.20.(vi) below. This lemma captures the combinatorics of piecewise affine functions satisfying conditions of the form (2.2).

Lemma 2.20. *Let $n \in \mathbb{N}_0$, $A_0, \ldots, A_n, B_0, \ldots, B_n, q_0, \ldots, q_{n+1} \in \mathbb{R}$ satisfy $q_0 < \cdots < q_{n+1}$, and consider a function $f \in C([q_0, q_{n+1}], \mathbb{R})$ satisfying for all $i \in \{0, \ldots, n\}$, $x \in [q_i, q_{i+1}]$ that $f(x) = A_i x + B_i$ and $\int_{q_i}^{q_{i+1}} (f(y) - y)dy = 0$. Then*

(i) *we have for all* $i \in \{0, \dots, n\}$ *that*

$$\begin{aligned} A_i - 1 &= (-1)^i \frac{q_1 - q_0}{q_{i+1} - q_i}(A_0 - 1), \\ B_i &= (-1)^{i+1} \frac{q_{i+1} + q_i}{2} \frac{q_1 - q_0}{q_{i+1} - q_i}(A_0 - 1), \end{aligned} \tag{2.4}$$

(ii) *we have* $f = \mathrm{id}_{[q_0, q_{n+1}]} \iff \forall i \in \{0, \dots, n\} \colon A_i = 1,\ B_i = 0$
$\iff \exists i \in \{0, \dots, n\} \colon A_i = 1,\ B_i = 0 \iff \exists i \in \{0, \dots, n\} \colon f|_{[q_i, q_{i+1}]} = \mathrm{id}_{[q_i, q_{i+1}]}$,

(iii) *for all* $i \in \{0, \dots, n\}$ *we have* $\operatorname{sign}(A_i - 1) = (-1)^i \operatorname{sign}(A_0 - 1)$.

If, in addition, $0 = \int_{q_0}^{q_{n+1}} x(f(x) - x)dx$, *then*

(iv) *we have* $0 = (A_0 - 1) \sum_{i=0}^{n} (-1)^i (q_{i+1} - q_i)^2$,

(v) *if* $f \neq \mathrm{id}_{[q_0, q_{n+1}]}$, *then* $0 = \sum_{i=0}^{n} (-1)^{i+1} (q_{i+1} - q_i)^2$,

(vi) *if* $n = 0$, *then* $f = \mathrm{id}_{[q_0, q_1]}$.

Proof. First note that we must have $A_i q_{i+1} + B_i = A_{i+1} q_{i+1} + B_{i+1}$ for all $i \in \{0, \dots, n-1\}$. Moreover, the assumption $0 = \int_{q_i}^{q_{i+1}} (f(x) - x)dx$ is equivalent to $B_i = -\frac{1}{2}(q_{i+1} + q_i)(A_i - 1)$. Combining these yields

$$A_{i+1} - 1 = -\frac{q_{i+1} - q_i}{q_{i+2} - q_{i+1}}(A_i - 1)$$

for all $i \in \{0, \dots, n-1\}$. Induction then proves the formula for $A_i - 1$, and the formula for B_i follows. Lastly, by plugging the formulas for A_i and B_i into $f(x)$, we compute

$$\begin{aligned} \int_{q_0}^{q_{n+1}} x(f(x) - x)dx &= \sum_{i=0}^{n} \int_{q_i}^{q_{i+1}} x((A_i - 1)x + B_i)dx \\ &= \frac{q_1 - q_0}{12}(A_0 - 1) \sum_{i=0}^{n} (-1)^i (q_{i+1} - q_i)^2. \end{aligned}$$

The remaining items follow immediately. ∎

In order to apply this lemma later on, let us verify that our network always satisfies the condition $\int_{q_i}^{q_{i+1}} (f(y) - y)dx = 0$ for suitable choices of q_i and q_{i+1}.

Lemma 2.21. *Suppose* $\phi \in \mathbb{R}^{3N+1}$ *is a critical point or a local extremum of* $\mathcal{L}$ *and denote by* $0 = q_0 < q_1 < \cdots < q_n < q_{n+1} = 1$, *for* $n \in \mathbb{N}_0$, *the roughest partition such that* f_ϕ *is affine on all subintervals* $[q_i, q_{i+1}]$. *Then we have for all* $i \in \{0, \dots, n\}$ *that*

$$\int_{q_i}^{q_{i+1}} (f_\phi(x) - x)dx = 0.$$

Proof. First, note that ϕ must have a non-flat type-2-active neuron whose breakpoint is q_i, for all $i \in \{1, \dots, n\}$. From the fourth line of (2.1), we know that $\int_0^1 (f_\phi(x) - x)dx = 0$. This and the second line of (2.1) imply, for any non-flat type-2-active neuron j,

$$\int_{I_j} (f_\phi(x) - x)dx = 0 = \int_{[0,1] \setminus I_j} (f_\phi(x) - x)dx.$$

Since either $I_j = [0, t_j]$ or $[0, 1] \setminus I_j = [0, t_j]$, it follows that $\int_0^{q_i} (f_\phi(x) - x)dx = 0$, for all $i \in \{0, \dots, n+1\}$. Taking differences of these integrals yields the desired statement. ∎

Next, as a first application of Lemma 2.20, we prove that only global minima can have type-1-active or non-flat semi-active neurons. We already established this in Lemma 2.18 in the affine case, but now we extend it to the non-affine case. The statement from Lemma 2.18 about saddle points not having non-flat degenerate neurons also holds in the non-affine case, but we will not see this until later in Section 2.8.

Lemma 2.22. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}$ but not a global minimum. Then ϕ does not have any type-1-active or non-flat semi-active neurons.*

Proof. For affine f_ϕ, the result has been established in Lemma 2.18. Thus, suppose f_ϕ is not affine on $[0,1]$ and that ϕ has a type-1-active or non-flat semi-active neuron. Denote by $0 = q_0 < q_1 < \cdots < q_n < q_{n+1} = 1$, for $n \in \mathbb{N}$, the roughest partition such that f_ϕ is affine on all subintervals $[q_i, q_{i+1}]$. We know from Lemma 2.21 that $\int_{q_0}^{q_1}(f_\phi(x)-x)dx = 0$, and we claim that also $\int_{q_0}^{q_1} x(f_\phi(x)-x)dx = 0$. To prove this, note that ϕ must have at least one non-flat type-2-active neuron (without loss of generality the first) with breakpoint $-b_1/w_1 = q_1$. Moreover, (2.2) shows that $0 = \int_0^1 x(f_\phi(x)-x)dx$ if applied with the type-1-active or non-flat semi-active neuron. Using this and $\frac{\partial}{\partial w_1}\mathcal{L}(\phi) = 0$, one deduces the claim as in the proof of Lemma 2.21. Hence, we conclude $f_\phi|_{[q_0,q_1]} = \mathrm{id}_{[q_0,q_1]}$ with the argument (2.3). But then we also get $f_\phi = \mathrm{id}_{[q_0,q_{n+1}]}$ by Lemma 2.20.(ii) and Lemma 2.21, yielding a contradiction. ∎

We now turn to the proof of Theorem 2.4.(IV.b). More precisely, we show that critical points and local extrema whose realizations are not affine must take a very specific form. The only degree of freedom of their realization functions is a single parameter varying over the set of even integers in $\{1,\dots,N\}$. Examples of the possible realizations are shown in Fig. 4.2, which illustrates that the degree of freedom is reflected by the number of breakpoints. Once this number is fixed, the shape of the function is uniquely determined: the breakpoints are equally spaced in the interval $[0,1]$, and the slope of the realization on each affine segment alternates between two given values in such a way that the function symmetrically oscillates around the diagonal. In addition, we deduce in Lemma 2.23 that critical points and local extrema can realize these functions only in a very specific way, limited by few combinatorial choices.

Lemma 2.23. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}$ but not a global minimum and that f_ϕ is not affine on $[0,1]$. Denote by $0 = q_0 < q_1 < \cdots < q_n < q_{n+1} = 1$, for $n \in \mathbb{N}$, the roughest partition such that f_ϕ is affine on all subintervals $[q_i, q_{i+1}]$, and denote by $K_i \subseteq \{1,\dots,N\}$ the set of all type-2-active neurons of ϕ whose breakpoint is q_i. Then the following hold:*

(i) *n is even,*
(ii) *$q_i = \frac{i}{n+1}$ for all $i \in \{1,\dots,n\}$,*
(iii) *$-b_j/w_j \in \{q_1,\dots,q_n\}$ for all type-2-active neurons $j \in \{1,\dots,N\}$ of ϕ,*
(iv) *$\mathrm{sign}(w_j) = (-1)^{i+1}$ for all $i \in \{1,\dots,n\}$, $j \in K_i$,*
(v) *$\sum_{j\in K_i} v_j w_j = 2/(n+1)$ for all $i \in \{1,\dots,n\}$,*
(vi) *ϕ is centered,*
(vii) *$f_\phi(x) = x - \frac{(-1)^i}{n+1}\big(x - \frac{i+1/2}{n+1}\big)$ for all $i \in \{0,\dots,n\}$, $x \in [q_i, q_{i+1}]$.*

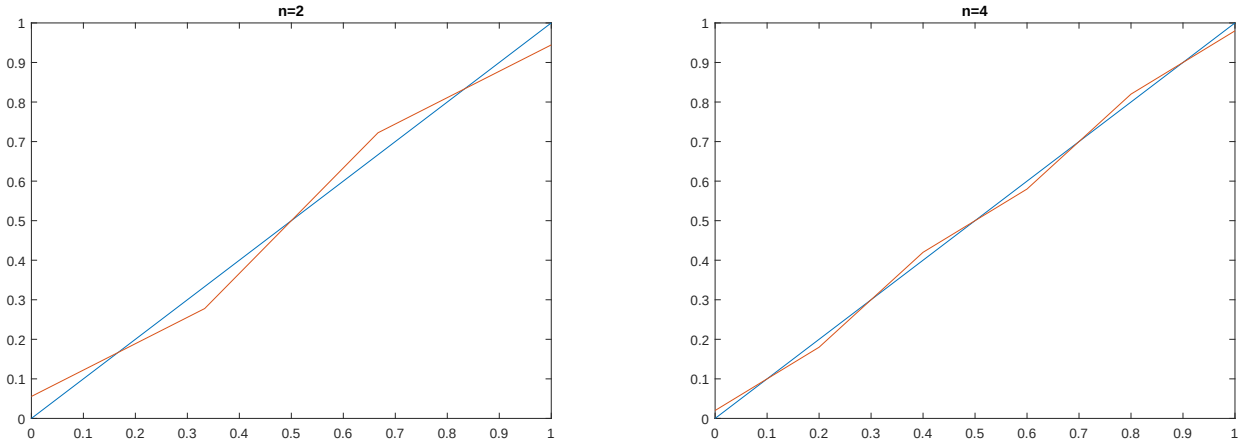

Figure 4.2: Examples of the network realizations (red) in Lemma 2.23 for the cases $n = 2$ and $n = 4$. The blue line is the target function (identity function).

The proof of this lemma requires a successive application of Lemma 2.20. We prove the statements of the lemma in a different order than stated. First of all, Lemma 2.20.(ii) will enforce the correct sign for each w_j, $j \in K_i$. That n is even will be a consequence of these signs. It will also follow from the signs together with Lemma 2.20.(v) that $q_i = \frac{i}{n+1}$. Afterwards, we use the formulas (2.4) from Lemma 2.20 to verify that any type-2-active neuron must have as breakpoint one of $q_1, \dots, q_n$. Once this has been shown, we obtain a more explicit version of those formulas and deduce $\sum_{k \in K_i} v_k w_k = 2/(n+1)$. That f_ϕ takes exactly the form in Lemma 2.23.(vii) is a byproduct of the last derivation, and that ϕ is centered is shown last.

Proof of Lemma 2.23. We begin by noting that none of the sets K_i, $i \in \{1, \dots, n\}$, can be empty. Furthermore, the third equation of (2.1) and Lemma 2.21 imply that (2.2) holds for all neurons in $\bigcup_i K_i$ even if they are flat. Applying Lemma 2.20.(ii), which we can do by Lemma 2.21, ensures that $f_\phi|_{[q_i, q_{i+1}]} \neq \mathrm{id}_{[q_i, q_{i+1}]}$ for all $i \in \{0, \dots, n\}$. In particular, (2.2) and the argument (2.3) show for all $i \in \{1, \dots, n-1\}$ and $j_0 \in K_i$, $j_1 \in K_{i+1}$ that $\mathrm{sign}(w_{j_0}) \neq \mathrm{sign}(w_{j_1})$ for otherwise we would have $I_{j_0} \backslash I_{j_1} = [q_i, q_{i+1}]$ or $I_{j_1} \backslash I_{j_0} = [q_i, q_{i+1}]$ (depending on the sign) and, hence,

$$\int_{q_i}^{q_{i+1}} (f_\phi(x) - x)dx = 0 = \int_{q_i}^{q_{i+1}} x(f_\phi(x) - x)dx.$$

Likewise, we must have $\int_0^{q_1} x(f_\phi(x) - x)dx \neq 0$ and, hence, $w_j > 0$ for any $j \in K_1$. Combining the previous two arguments establishes $\mathrm{sign}(w_j) = (-1)^{i+1}$ for any $i \in \{1, \dots, n\}$, $j \in K_i$. Just like $w_j > 0$ for any $j \in K_1$, we must also have $w_j < 0$ for any $j \in K_n$. Thus, $-1 = \mathrm{sign}(w_j) = (-1)^{n+1}$ for all $j \in K_n$, so n is even. Now that we know the sign of each parameter w_j for neurons $j \in \bigcup_i K_i$, we can use (2.2) again to find that $\int_{q_i}^{q_{i+2}} x(f_\phi(x) - x)dx = 0$ for all $i \in \{0, \dots, n-1\}$. Then Lemma 2.20.(v) (with the partition q_i, q_{i+1}, q_{i+2}) tells us

$$0 = (q_{i+2} - q_{i+1})^2 - (q_{i+1} - q_i)^2.$$

This can only hold for all $i \in \{0, \dots, n-1\}$ if the points $q_1, \dots, q_n$ are equidistributed, which means $q_i = i/(n+1)$. Next, if we denote $f_\phi(x) = A_i x + B_i$ on $[q_i, q_{i+1}]$, then the formulas (2.4)

must hold for all $i \in \{0, \dots, n\}$. Since $q_1, \dots, q_n$ are equidistributed, the formulas simplify to

$$A_i - 1 = (-1)^i (A_0 - 1) \quad \text{and} \quad B_i = (-1)^{i+1} \frac{i + \frac{1}{2}}{n+1} (A_0 - 1) \tag{2.5}$$

for all $i \in \{0, \dots, n\}$. Using (2.5), one can verify that any type-2-active neuron of ϕ must have as breakpoint one of the points $q_1, \dots, q_n$. If this were not the case, say the j^{th} hidden neuron were type-2-active with breakpoint $t_j = -b_j/w_j$, then one could choose $i \in \{0, \dots, n\}$ such that $q_i < t_j < q_{i+1}$. Using (2.2), (2.5), and Lemma 2.21, the integral from the third line of (2.1) reads (after dividing by $2w_j$)

$$\begin{aligned}
&\int_{I_j} (x - t_j)(f_\phi(x) - x) dx \\
&= \int_{[q_i, q_{i+1}] \cap I_j} (x - t_j)(f_\phi(x) - x) dx - \begin{cases} 0 & \text{if } i \text{ is even} \\ \int_{q_i}^{q_{i+1}} x(f_\phi(x) - x) dx & \text{if } i \text{ is odd} \end{cases} \\
&= \begin{cases} \frac{1}{6}(A_0 - 1)(t_j - q_i)^2 (q_{i+1} - t_j + \frac{1}{2(n+1)}) & \text{if } I_j = [0, t_j] \text{ and } i \text{ is even} \\ & \text{or if } I_j = [t_j, 1] \text{ and } i \text{ is odd} \\ \frac{1}{6}(A_0 - 1)(q_{i+1} - t_j)^2 (t_j - q_i + \frac{1}{2(n+1)}) & \text{if } I_j = [0, t_j] \text{ and } i \text{ is odd} \\ & \text{or if } I_j = [t_j, 1] \text{ and } i \text{ is even.} \end{cases}
\end{aligned}$$

So, the partial derivative of $\mathcal{L}$ with respect to v_j does not vanish, yielding a contradiction. This proves that all type-2-active neurons lie in $\bigcup_i K_i$. In particular, we can write

$$A_l = \sum_{\substack{i=1 \\ i \text{ odd}}}^{l} \sum_{j \in K_i} v_j w_j + \sum_{\substack{i=l+1 \\ i \text{ even}}}^{n} \sum_{j \in K_i} v_j w_j$$

for all $l \in \{0, \dots, n\}$ because ϕ does not have any type-1-active neurons by Lemma 2.22. We can combine this formula with (2.5) to find for all $i \in \{0, \dots, n-1\}$

$$-(A_0 - 1) = (-1)^i (A_{i+1} - 1) = (-1)^i (A_i - 1) + \sum_{j \in K_{i+1}} v_j w_j = A_0 - 1 + \sum_{j \in K_{i+1}} v_j w_j.$$

Thus, the quantity $a := \sum_{j \in K_i} v_j w_j$ is independent of $i \in \{1, \dots, n\}$. Consequently, we obtain $A_i = an/2$ for even i (including $i = 0$) and $A_i = a(1 + n/2)$ for odd i. The identity $A_1 - 1 = 1 - A_0$ then forces $a = 2/(n+1)$. That ϕ has to be centered follows from $f_\phi(0) = B_0$. ∎

As our final building block for the proof of Theorem 2.4, we show that the networks from Lemma 2.23 are saddle points of the loss function. To achieve this, we will find a set of coordinates in which $\mathcal{L}$ is twice differentiable and calculate the determinant of the Hessian of $\mathcal{L}$ restricted to these coordinates. It will turn out to be strictly negative, from which it follows that we deal with a saddle point.

Lemma 2.24. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}$ but not a global minimum and that f_ϕ is not affine on $[0, 1]$. Then ϕ is a saddle point of $\mathcal{L}$.*

Proof. Take $n \in \mathbb{N}$ satisfying the assumptions of Lemma 2.23 and let $K_1 \subseteq \{1, \dots, N\}$ denote the set of those type-2-active neurons with breakpoint $1/(n+1)$. Denote by $K_1^- \subseteq K_1$ the set of all those hidden neurons $j \in K_1$ with $v_j < 0$. It may happen that K_1^- is empty. However, the complement $K_1 \backslash K_1^-$ is never empty since $\sum_{j \in K_1} v_j w_j = 2/(n+1)$ and $\operatorname{sign}(w_j) = 1$ for all $j \in K_1$ by Lemma 2.23. Let $j_1 \in K_1$ be any hidden neuron with $v_{j_1} > 0$ and denote by $j_2, \dots, j_l$, for $l \in \{1, \dots, N\}$, an enumeration of K_1^-. Moreover, let $k \in \{1, \dots, N\}$ be any type-2-active neuron with breakpoint $t_k = 2/(n+1)$.

We know from Lemma 2.15 that $\mathcal{L}$ is twice continuously differentiable in the coordinates of type-2-active neurons and in (v, c). We will show that the Hessian H of $\mathcal{L}$ restricted to $(b_{j_1}, \dots, b_{j_l}, v_k, c)$ has a strictly negative determinant.

In order to compute this determinant, we introduce some shorthand notation. For $i \in \{1, \dots, l\}$, denote $\lambda_i = \frac{n+1}{2} v_{j_i} w_{j_i}$ so that $\sum_{i=1}^{l} \lambda_i \leq 1$ by the choice of neurons in the collection $\{j_1, \dots, j_l\}$. Define $\mu = \frac{n+1}{2n}$ and the vectors $u_1 = (v_{j_1}, \dots, v_{j_l})$, $u_2 = (\frac{-1}{4n^2 \mu} w_k, 1)$, and $u = (u_1, u_2)$. Furthermore, let D be the diagonal matrix with entries $-v_{j_i}^2/(4\lambda_i n)$, $i \in \{1, \dots, l\}$, let A be the Hessian of $\mathcal{L}$ restricted to (v_k, c), let $B = \mu A - u_2 u_2^T$, and let E be the diagonal block matrix with blocks D and B. Then $H = \frac{1}{\mu}(E + uu^T)$ and, hence,

$$\det(H) = \mu^{-(l+2)}(1 + u^T E^{-1} u) \det(E)$$

once we verified that E is invertible. We calculate directly

$$\det(A) = \det \begin{pmatrix} \frac{2}{3(n\mu)^3} w_k^2 & \frac{-1}{(n\mu)^2} w_k \\ \frac{-1}{(n\mu)^2} w_k & \frac{n+1}{n\mu} \end{pmatrix} = \frac{2n-1}{3(n\mu)^4} w_k^2 > 0.$$

Next, we compute

$$\Gamma := \frac{1}{\mu} u_2^T A^{-1} u_2 = \frac{32n^2 - 21n + 3}{16n(2n-1)} \in (0, 1). \tag{2.6}$$

Using Γ, we obtain $\det(B) = \mu^2(1-\Gamma)\det(A) > 0$ and $B^{-1} = \frac{1}{\mu} A^{-1} + \frac{1}{\mu^2(1-\Gamma)} A^{-1} u_2 u_2^T A^{-1}$. In particular, E is invertible. Using $u_2^T B^{-1} u_2 = \frac{\Gamma}{1-\Gamma}$, we can write

$$u^T E^{-1} u = u_1^T D^{-1} u_1 + u_2^T B^{-1} u_2 = -4n \sum_{i=1}^{l} \lambda_i + \frac{\Gamma}{1-\Gamma}.$$

The determinant of D is $-(4n)^{-l} \prod_{i=1}^{l} v_{j_i}^2 |\lambda_i|^{-1} < 0$ so that

$$\Delta := -\mu^{-(l+2)}(1-\Gamma)^{-1} \det(D) \det(B)$$

is strictly positive. Summing up, we obtain that the determinant of H is

$$\det(H) = \Delta\Big(4n(1-\Gamma) \sum_{i=1}^{l} \lambda_i - 1\Big).$$

We already mentioned that $\sum_{i=1}^{l} \lambda_i \leq 1$. Finally, we compute $4n(1-\Gamma) = \frac{5n-3}{8n-4} < 1$ to conclude $\det(H) < 0$, which finishes the proof. ∎

We now have constructed all the tools needed to prove Theorem 2.4 in the special case in which the target function is the identity on $[0, 1]$. This will be done in the next section.

2.8 Classification of the critical points if the target function is the identity

In this section, we gather the results of the previous two sections to prove the main theorem in the case where the target function is the identity on $[0,1]$.

Proposition 2.25. *Let $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$. Then the following hold:*

(I) *ϕ is not a local maximum of $\mathcal{L}$.*

(II) *If ϕ is a critical point or a local extremum of $\mathcal{L}$, then $\mathcal{L}$ is differentiable at ϕ with gradient $\nabla\mathcal{L}(\phi) = 0$.*

(III) *ϕ is a non-global local minimum of $\mathcal{L}$ if and only if ϕ is centered and, for all $j \in \{1, \dots, N\}$, the j^{th} hidden neuron of ϕ is*

- (a) *inactive,*
- (b) *semi-inactive with $I_j = \{0\}$ and $v_j > 0$, or*
- (c) *semi-inactive with $I_j = \{1\}$ and $v_j < 0$.*

(IV) *ϕ is a saddle point of $\mathcal{L}$ if and only if ϕ is centered, ϕ does not have any type-1-active neurons, ϕ does not have any non-flat semi-active neurons, ϕ does not have any non-flat degenerate neurons, and exactly one of the following two items holds:*

- (a) *ϕ does not have any type-2-active neurons and there exists $j \in \{1, \dots, N\}$ such that the j^{th} hidden neuron of ϕ is*
  - (i) *flat semi-active,*
  - (ii) *semi-inactive with $I_j = \{0\}$ and $v_j \leq 0$,*
  - (iii) *semi-inactive with $I_j = \{1\}$ and $v_j \geq 0$, or*
  - (iv) *flat degenerate.*
- (b) *There exists $n \in \{2, 4, 6, \dots\}$ such that*

$$\bigcup_{j \in \{1,\dots,N\},\, w_j \neq 0} \left\{ -\frac{b_j}{w_j} \right\} \cap (0,1) = \bigcup_{i=1}^{n} \left\{ \frac{i}{n+1} \right\}$$

and, for all $j \in \{1, \dots, N\}$, $i \in \{1, \dots, n\}$ with $w_j \neq 0 = b_j + \frac{iw_j}{n+1}$, it holds that $\operatorname{sign}(w_j) = (-1)^{i+1}$ and

$$\sum_{k \in \{1,\dots,N\},\, w_k \neq 0 = b_k + \frac{iw_k}{n+1}} v_k w_k = \frac{2}{n+1}.$$

(V) *If ϕ is a non-global local minimum of $\mathcal{L}$ or a saddle point of $\mathcal{L}$ without type-2-active neurons, then $f_\phi(x) = 1/2$ for all $x \in [0,1]$.*

(VI) *If ϕ is a saddle point of $\mathcal{L}$ with at least one type-2-active neuron, then there exists $n \in \{2, 4, 6, \dots\}$ such that $n \leq N$ and, for all $i \in \{0, \dots, n\}$, $x \in [\frac{i}{n+1}, \frac{i+1}{n+1}]$, one has*

$$f_\phi(x) = x - \frac{(-1)^i}{n+1}\Big(x - \frac{i + \frac{1}{2}}{n+1}\Big). \tag{2.7}$$

Proof. Statement (I) follows from Lemma 2.8 and the 'if' part of the 'if and only if' statement in (III) is the content of Lemma 2.17. Moreover, if ϕ is as in (IV.a), then it is a critical point because it satisfies (2.1) and it is a saddle point by Lemma 2.19. Next, denote $q_i = i/(n+1)$ for all $i \in \{0, \dots, n+1\}$. If ϕ is as in (IV.b), then its realization on $[0,1]$ is given by

$$f_\phi(x) = \frac{1}{2} + \frac{2}{n+1} \sum_{i=1}^{n} (-1)^{i+1} \max\{(-1)^{i+1}(x - q_i), 0\}. \tag{2.8}$$

which coincides with the formula (2.7). In particular, we have $\int_{q_i}^{q_{i+1}} (f_\phi(x) - x)dx = 0$ for all $i \in \{0, \dots, n\}$ and $\int_{q_i}^{q_{i+2}} x(f_\phi(x) - x)dx = 0$ for all $i \in \{0, \dots, n-1\}$. The latter asserts that $\int_{q_i}^{1} x(f_\phi(x) - x)dx = 0$ for odd i and $\int_{0}^{q_i} x(f_\phi(x) - x)dx = 0$ for even i. Thus, ϕ satisfies (2.1) and, hence, is a critical point. Furthermore, it is a saddle point by Lemma 2.24. This proves the 'if' part of the 'if and only if' statement in (IV).

Now, suppose ϕ is a non-global local minimum. Then f_ϕ is affine by Lemma 2.24. Lemma 2.18 asserts that ϕ is centered and does not have any active or non-flat semi-active neurons. Furthermore, for each hidden neuron, Lemma 2.19 rules out all possibilities except (III.a)-(III.c). This proves the 'only if' part of (III).

Next, suppose ϕ is a saddle point. If f_ϕ is affine, then ϕ is centered and does not have any active, non-flat semi-active, or non-flat degenerate neurons by Lemma 2.18. If there is no hidden neuron as in (IV.a.i)-(IV.a.iv), then all hidden neurons satisfy one of the conditions in (III.a)-(III.c). But this contradicts Lemma 2.17. This proves (IV.a). If f_ϕ is not affine, then it still does not admit any type-1-active or non-flat semi-active neurons by Lemma 2.22. Moreover, Lemma 2.23 shows that ϕ is centered and its type-2-active neurons satisfy (IV.b). We need to argue that ϕ does not have any non-flat degenerate neurons in this case either. If there were a non-flat degenerate neuron, then $\mathcal{G}(\phi) = 0$ implies $0 = \int_0^1 x(f_\phi(x) - x)dx$. But Lemma 2.20.(v) and Lemma 2.23 ensure that this integral is different from zero. This finishes the proof of the 'only if' part of (IV).

Next, we prove (II). If ϕ is a saddle point, then it does not have any non-flat degenerate neurons by (IV). If ϕ is a non-global local extremum, then (I) and (III) imply that ϕ does not have any non-flat degenerate neurons either. Thus, $\mathcal{L}$ is differentiable at ϕ by Lemma 2.14. If ϕ is a global minimum, then ϕ is point of differentiability by Lemma 2.11.

Statement (V) follows immediately from (III) and (IV.a). The remaining statement (VI) is implied by (IV.b) and (2.8). ∎

2.9 Completion of the proof of Theorem 2.4

In this section, we show that Theorem 2.4 can always be reduced to its special case, Proposition 2.25, by employing a transformation of the parameter space.

Proof of Theorem 2.4. First, we assume that $T = (0,1)$. Consider the transformation $P\colon \mathbb{R}^{3N+1} \to \mathbb{R}^{3N+1}$ of the parameter space given by $P(w,b,v,c) = (w, b, \frac{v}{\alpha}, \frac{c-\beta}{\alpha})$. We then have $\mathcal{L}_{N,T,\mathcal{A}}(\phi) = \alpha^2 \mathcal{L} \circ P(\phi)$ for all $\phi \in \mathbb{R}^{3N+1}$. Since the coordinates w and b remain unchanged and the vector v only gets scaled under the transformation P, the transformation P does not change the types of the hidden neurons. Moreover, a network $\phi \in \mathbb{R}^{3N+1}$ is $(T, \mathcal{A})$-centered if and only if $P(\phi)$ is centered. The map P clearly is a smooth diffeomorphism and, hence, Theorem 2.4 with $T = (0,1)$ is exactly what we obtain from Proposition 2.25 under the transformation P.

Now, we deduce Theorem 2.4 for general T. This time, set $\mathcal{B} = (\alpha(T_1 - T_0), \alpha T_0 + \beta)$ and denote by $Q\colon \mathbb{R}^{3N+1} \to \mathbb{R}^{3N+1}$ the transformation $Q(w, b, v, c) = ((T_1 - T_0)w, T_0 w + b, v, c)$. Then $\mathcal{L}_{N,T,\mathcal{A}}(\phi) = (T_1 - T_0)\mathcal{L}_{N,(0,1),\mathcal{B}} \circ Q(\phi)$ for any $\phi \in \mathbb{R}^{3N+1}$. As above, the transformation Q does not change the types of the hidden neurons. Note for the breakpoints that

$$-\frac{b_j}{w_j} = T_0 + \frac{i(T_1 - T_0)}{n+1} \iff -\frac{T_0 w_j + b_j}{(T_1 - T_0)w_j} = \frac{i}{n+1}.$$

Also, $\phi \in \mathbb{R}^{3N+1}$ is $(T, \mathcal{A})$-centered if and only if $Q(\phi)$ is $((0,1), \mathcal{B})$-centered. Since we have shown the theorem to hold for $T = (0, 1)$, the smooth diffeomorphism Q yields Theorem 2.4 in the general case. ■

3. From ReLU to leaky ReLU

In this section, we attempt to derive Theorem 2.4 for leaky ReLU activation, given by $x \mapsto \max\{x, \gamma x\}$ for a parameter $\gamma \in (0, 1)$. We denote the realization $f_\phi^\gamma \in C(\mathbb{R}, \mathbb{R})$ of a network $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$ with this activation by

$$f_\phi^\gamma(x) = c + \sum_{j=1}^{N} v_j \max\{w_j x + b_j, \gamma(w_j x + b_j)\}.$$

Analogously to the ReLU case, given $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$ and $T = (T_0, T_1) \in \mathbb{R}^2$, the loss function $\mathcal{L}^\gamma_{N,T,\mathcal{A}} \in C(\mathbb{R}^{3N+1}, \mathbb{R})$ is the L^2-loss given by

$$\mathcal{L}^\gamma_{N,T,\mathcal{A}}(\phi) = \int_{T_0}^{T_1} (f_\phi^\gamma(x) - \alpha x - \beta)^2 \, dx.$$

Again, we call a point a critical point of $\mathcal{L}^\gamma_{N,T,\mathcal{A}}$ if it is a zero of the generalized gradient defined by right-hand partial derivatives. The notions about types of neurons remain the same as in Definition 2.3. Strictly speaking, the notions 'inactive' and 'semi-inactive' are no longer suitable for leaky ReLU activation, but it is convenient to stick to the same terminology. We will deduce the classification for leaky ReLU by reducing it to the ReLU case in some instances and deal with other instances directly.

3.1 Partial reduction to the ReLU case

As before, we first consider the special case where the target function is the identity on $[0, 1]$. Let us abbreviate $\mathcal{L}^\gamma = \mathcal{L}^\gamma_{N,(0,1),(1,0)}$ and $\mathcal{L} = \mathcal{L}_{2N,(0,1),(1,0)}$. Let $P\colon \mathbb{R}^{3N+1} \to \mathbb{R}^{6N+1}$ denote the smooth map $P(w, b, v, c) = (w, -w, b, -b, v, -\gamma v, c)$. Then, $f_\phi^\gamma = f_{P(\phi)}$ and $\mathcal{L}^\gamma = \mathcal{L} \circ P$. Hence, if $\mathcal{L}$ is differentiable at $P(\phi)$, then $\mathcal{L}^\gamma$ is differentiable at ϕ, so differentiability properties of $\mathcal{L}$ convert to $\mathcal{L}^\gamma$. The partial derivatives of $\mathcal{L}^\gamma$ at any network ϕ and any

non-degenerate or flat degenerate neuron j are given by

$$\begin{aligned}
\frac{\partial}{\partial w_j}\mathcal{L}^\gamma(\phi) &= \Big(\frac{\partial}{\partial w_j}\mathcal{L}\Big)(P(\phi)) - \Big(\frac{\partial}{\partial w_{j+N}}\mathcal{L}\Big)(P(\phi)),\\
\frac{\partial}{\partial b_j}\mathcal{L}^\gamma(\phi) &= \Big(\frac{\partial}{\partial b_j}\mathcal{L}\Big)(P(\phi)) - \Big(\frac{\partial}{\partial b_{j+N}}\mathcal{L}\Big)(P(\phi)),\\
\frac{\partial}{\partial v_j}\mathcal{L}^\gamma(\phi) &= \Big(\frac{\partial}{\partial v_j}\mathcal{L}\Big)(P(\phi)) - \gamma\Big(\frac{\partial}{\partial v_{j+N}}\mathcal{L}\Big)(P(\phi)),\\
\frac{\partial}{\partial c}\mathcal{L}^\gamma(\phi) &= \Big(\frac{\partial}{\partial c}\mathcal{L}\Big)(P(\phi)).
\end{aligned}$$

We can also write these in explicit formulas. To do so, we complement the notation I_j by the intervals $\hat{I}_j = \{x \in [0,1] \colon w_j x + b_j < 0\} = [0,1]\backslash I_j$. Then,

$$\begin{aligned}
\frac{\partial}{\partial w_j}\mathcal{L}^\gamma(\phi) &= 2v_j \int_{I_j} x(f_\phi^\gamma(x) - x)dx + 2\gamma v_j \int_{\hat{I}_j} x(f_\phi^\gamma(x) - x)dx,\\
\frac{\partial}{\partial b_j}\mathcal{L}^\gamma(\phi) &= 2v_j \int_{I_j} (f_\phi^\gamma(x) - x)dx + 2\gamma v_j \int_{\hat{I}_j} (f_\phi^\gamma(x) - x)dx,\\
\frac{\partial}{\partial v_j}\mathcal{L}^\gamma(\phi) &= 2 \int_{I_j} (w_j x + b_j)(f_\phi^\gamma(x) - x)dx + 2\gamma \int_{\hat{I}_j} (w_j x + b_j)(f_\phi^\gamma(x) - x)dx,\\
\frac{\partial}{\partial c}\mathcal{L}^\gamma(\phi) &= 2 \int_0^1 (f_\phi^\gamma(x) - x)dx.
\end{aligned}$$

This notation allows to treat non-flat degenerate neurons. For such neurons, the right-hand partial derivatives of $\mathcal{L}^\gamma$ are also given by the above formulas. We now show how the reduction to the ReLU case works.

Lemma 3.1. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}^\gamma$ but not a global minimum and that $\int_0^1 x(f_\phi^\gamma(x) - x)dx = 0$. Then all neurons of ϕ are flat semi-active, flat inactive with $w_j = 0$, or flat degenerate.*

Proof. We first show that $P(\phi)$ is a critical point of $\mathcal{L}$ and then apply Theorem 2.4 to $P(\phi)$. Since the partial derivative of $\mathcal{L}^\gamma$ with respect to c exists and must be zero, we have

$$\frac{1}{2}\frac{\partial}{\partial c}\mathcal{L}^\gamma(\phi) = \int_0^1 (f_{P(\phi)}(x) - x)dx = 0 = \int_0^1 x(f_{P(\phi)}(x) - x)dx.$$

This shows that the (right-hand) partial derivatives of $\mathcal{L}$ are zero at $P(\phi)$ with respect to coordinates corresponding to inactive, semi-inactive, semi-active, type-1-active, and degenerate neurons. We need to verify that also partial derivatives of $\mathcal{L}$ with respect to type-2-active neurons vanish at $P(\phi)$. To see this, note that, for a type-2-active neuron j of ϕ, the partial derivative of $\mathcal{L}^\gamma$ with respect to w_j exists at ϕ and

$$0 = \frac{\partial}{\partial w_j}\mathcal{L}^\gamma(\phi) = 2(1-\gamma)v_j \int_{I_j} x(f_\phi^\gamma(x) - x)dx.$$

Thus,

$$\begin{aligned}
0 = 2v_j \int_{I_j} x(f_\phi^\gamma(x) - x)dx &= \Big(\frac{\partial}{\partial w_j}\mathcal{L}\Big)(P(\phi)),\\
0 = -2\gamma v_j \int_{\hat{I}_j} x(f_\phi^\gamma(x) - x)dx &= \Big(\frac{\partial}{\partial w_{j+N}}\mathcal{L}\Big)(P(\phi)),
\end{aligned}$$

and analogously for the coordinates $b_j, b_{j+N}, v_j, v_{j+N}$. This concludes that $P(\phi)$ is a critical point of $\mathcal{L}$. By Theorem 2.4, $P(\phi)$ does not have any type-1-active, non-flat semi-active, or non-flat degenerate neurons. By definition of the map P, it follows that ϕ does not have any type-1-active, non-flat semi-active, or non-flat degenerate neurons, nor does it have any semi-inactive, non-flat inactive, or inactive neurons with $w_j \neq 0$ for otherwise $P(\phi)$ would have one of the former types. Further, by definition of P, any type-2-active neuron of ϕ gives rise to two type-2-active neurons of $P(\phi)$ with the same breakpoint but with opposite signs of the w-coordinate. This is not possible by (IV.b) of Theorem 2.4, so ϕ cannot have any type-2-active neurons. In summary, ϕ can only have flat semi-active, flat degenerate, or flat inactive neurons with $w_j = 0$. ∎

The condition $\int_0^1 x(f_\phi^\gamma(x) - x)dx = 0$ in the previous lemma is easily converted into a condition about existence of certain types of neurons. This is done in the first part of the next lemma. For the second part, we recycle some arguments we learned from the ReLU case.

Lemma 3.2. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}^\gamma$ but not a global minimum. Then all neurons of ϕ are flat semi-active, flat inactive with $w_j = 0$, degenerate, or type-2-active. Moreover, if ϕ does not have any non-flat type-2-active neurons, then ϕ is a saddle point and it also does not have any flat type-2-active or non-flat degenerate neurons.*

Proof. Suppose ϕ had a neuron of a different type than in the first statement of this lemma, say the j^{th}. Note that one of the intervals I_j and $\hat{I}_j$ is empty and the other one is $[0,1]$ (up to possibly a singleton). Since the j^{th} neuron is non-degenerate, $\mathcal{L}^\gamma$ is differentiable with respect to the coordinates of the j^{th} neuron, so $\int_0^1 x(f_\phi^\gamma(x) - x)dx = 0$. This contradicts Lemma 3.1.

The remainder of the proof is similar to the ones of Lemmas 2.18 and 2.19. Assume ϕ does not have any non-flat type-2-active neurons. Then f_ϕ^γ is constant on $[0,1]$, and this constant is $1/2$ since $\frac{\partial}{\partial c}\mathcal{L}^\gamma(\phi) = 0$. We claim that ϕ cannot have any flat type-2-active neurons. Suppose for contradiction the j^{th} neuron were that. Let $\tau = \operatorname{sign}(w_j)$ and $t_j = -b_j/w_j \in (0,1)$. Then $\frac{\partial}{\partial v_j}\mathcal{L}^\gamma(\phi) = 0$ implies

$$\begin{aligned} 0 &= \int_{t_j}^1 (x - t_j)(\tfrac{1}{2} - x)dx + \gamma^\tau \int_0^{t_j} (x - t_j)(\tfrac{1}{2} - x)dx \\ &= \frac{1}{12}\big(-1 + (1 - \gamma^\tau)(3 - 2t_j)t_j^2\big). \end{aligned}$$

But, for any $\gamma, t \in (0,1)$, $\tau \in \{-1, 1\}$, we have $-1 + (1 - \gamma^\tau)(3 - 2t)t^2 < 0$, which is a contradiction. Thus, all neurons of ϕ are flat semi-active, flat inactive with $w_j = 0$, or degenerate. With an argument analogous to the proof of Lemma 2.19, we find that ϕ is a saddle point of $\mathcal{L}^\gamma$. Indeed, if there is a flat semi-active or flat inactive neuron j with $w_j = 0$, then, with $\tau = 1 - \operatorname{sign}(b_j)$,

$$\det \begin{pmatrix} \frac{\partial}{\partial w_j}\frac{\partial}{\partial w_j}\mathcal{L}^\gamma(\phi) & \frac{\partial}{\partial w_j}\frac{\partial}{\partial v_j}\mathcal{L}^\gamma(\phi) \\ \frac{\partial}{\partial v_j}\frac{\partial}{\partial w_j}\mathcal{L}^\gamma(\phi) & \frac{\partial}{\partial v_j}\frac{\partial}{\partial v_j}\mathcal{L}^\gamma(\phi) \end{pmatrix} = -\left(2\gamma^{\tau/2}\int_0^1 x(\tfrac{1}{2} - x)dx\right)^2 = -\frac{1}{36}\gamma^\tau < 0.$$

Instead, if there is a degenerate neuron j, then, for the perturbation ϕ^s, $s \in [0,1]$, in the coordinates of the j^{th} neuron given by $w_j^s = \tau s$, $b_j^s = -\tau s^2$, and $v_j^s = v_j + \tau s$ with $\tau = 1$ if

$v_j \geq 0$ and $\tau = -1$ if $v_j < 0$, we have

$$\begin{aligned}
&\mathcal{L}^\gamma(\phi^s) - \mathcal{L}^\gamma(\phi) \\
&= \frac{1}{6} v_j^s w_j^s \gamma^{(1-\tau)/2} \big(-1 + (1-\gamma^\tau)(3-2s)s^2 \big) + \frac{1}{3} (v_j^s w_j^s)^2 \gamma^{1-\tau} \big((1-s)^3 + \gamma^{2\tau} s^3 \big) \\
&= -\frac{1}{6} s(|v_j| + s) \gamma^{(1-\tau)/2} + \frac{1}{3} |v_j|^2 s^2 \gamma^{1-\tau} + \mathcal{O}(s^3),
\end{aligned}$$

which is strictly negative for small $s > 0$. This concludes that ϕ is a saddle point. In particular, any degenerate neuron j must be flat because

$$0 = \frac{\partial^+}{\partial w_j} \mathcal{L}^\gamma(\phi) = 2 v_j \int_0^1 x(\tfrac{1}{2} - x) dx = -\frac{v_j}{6}.$$

■

We finished dealing with critical points of $\mathcal{L}^\gamma$ that have a constant realization function. In the next section, we find saddle points of $\mathcal{L}^\gamma$ analogous to the ones in Theorem 2.4.(IV.b). For these, we cannot reduce the analysis entirely to the known ReLU case. However, the arguments are analogous to the ones developed in Lemmas 2.23 and 2.24, and we can use a shortcut for small γ by arguing that we approximate the ReLU case in a suitable sense.

3.2 Explicit analysis for leaky ReLU

The following is the analog of Lemma 2.23 in the leaky ReLU case. Informally, one recovers Lemma 2.23 from Lemma 3.3 in the limit $\gamma \to 0$. We will discuss this in more detail after having proved the lemma.

Lemma 3.3. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}^\gamma$ but not a global minimum and that ϕ has a type-2-active neuron. Denote by $0 = q_0 < q_1 < \cdots < q_n < q_{n+1} = 1$, for $n \in \mathbb{N}_0$, the roughest partition such that f_ϕ^γ is affine on all subintervals $[q_i, q_{i+1}]$, and denote by $K_i \subseteq \{1, \dots, N\}$ the set of all type-2-active neurons of ϕ whose breakpoint is q_i. Then $n \geq 1$ and there exists $\sigma \in \{-1, 1\}$ such that, abbreviating*

$$\delta = \gamma^{(1-\sigma)/4} + \gamma^{(1-\sigma(-1)^n)/4} + (n-1)\sqrt{1+\gamma},$$

the following hold:

(i) (a) $q_i = q_1 + \frac{(i-1)(q_n - q_1)}{n-1}$ *for all* $i \in \{2, \dots, n-1\}$,
(b) $q_1 = \delta^{-1} \gamma^{(1-\sigma)/4}$, *and* $q_n = 1 - \delta^{-1} \gamma^{(1-\sigma(-1)^n)/4}$, *and* $q_n - q_1 = \delta^{-1}(n-1)\sqrt{1+\gamma}$,

(ii) $-b_j / w_j \in \{q_1, \dots, q_n\}$ *for all type-2-active neurons* $j \in \{1, \dots, N\}$ *of* ϕ,

(iii) $\operatorname{sign}(w_j) = \sigma(-1)^{i+1}$ *for all* $i \in \{1, \dots, n\}$, $j \in K_i$,

(iv) (a) $\sum_{j \in K_i} v_j w_j = \begin{cases} \gamma^{-1/2} & \text{if } i = 1 = n, \\ \frac{1}{\delta}\big(\frac{1}{\sqrt{1+\gamma}} + \frac{1}{\gamma^{(1-\sigma)/4}}\big) & \text{if } i = 1 \neq n, \\ \frac{1}{\delta}\frac{2}{\sqrt{1+\gamma}} & \text{if } 2 \leq i \leq n-1, \\ \frac{1}{\delta}\big(\frac{1}{\sqrt{1+\gamma}} + \frac{1}{\gamma^{(1-\sigma(-1)^n)/4}}\big) & \text{if } i = n \neq 1, \end{cases}$

(v) ϕ *is centered,*

(vi) $f_\phi^\gamma(x) - x = \dfrac{-\sigma(-1)^i(1-\gamma)}{\delta} \cdot \begin{cases} \frac{x}{\gamma^{(1-\sigma)/4}} - \frac{1}{2\delta} & \text{if } i = 0, \\ \frac{x}{\sqrt{1+\gamma}} - \frac{i-1/2}{\delta} - \frac{\gamma^{(1-\sigma)/4}}{\delta\sqrt{1+\gamma}} & \text{if } 1 \le i \le n-1, \\ \frac{x}{\gamma^{(1-\sigma(-1)^n)/4}} + \frac{1}{2\delta} - \frac{1}{\gamma^{(1-\sigma(-1)^n)/4}} & \text{if } i = n \end{cases}$

for all $i \in \{0, \dots, n\}$, $x \in [q_i, q_{i+1}]$.

Proof. First, note that ϕ must have at least one non-flat type-2-active neuron by Lemma 3.2. For any such neuron j,

$$0 = \frac{1}{2v_j}\frac{\partial}{\partial w_j}\mathcal{L}^\gamma(\phi) = (1-\gamma)\int_{I_j} x(f_\phi^\gamma(x) - x)dx + \gamma\int_0^1 x(f_\phi^\gamma(x) - x)dx,$$

so the two integrals

$$\begin{aligned} \int_{I_j} x(f_\phi^\gamma(x) - x)dx &= \frac{-\gamma}{1-\gamma}\int_0^1 x(f_\phi^\gamma(x) - x)dx, \\ \int_{\hat{I}_j} x(f_\phi^\gamma(x) - x)dx &= \frac{1}{1-\gamma}\int_0^1 x(f_\phi^\gamma(x) - x)dx \end{aligned} \tag{3.1}$$

are independent of the non-flat type-2-active neuron j. Doing the same with the coordinate b_j and using that $2\int_0^1 (f_\phi^\gamma(x) - x)dx = \frac{\partial}{\partial c}\mathcal{L}^\gamma(\phi) = 0$, we find

$$\int_{I_j} (f_\phi^\gamma(x) - x)dx = 0 = \int_{\hat{I}_j} (f_\phi^\gamma(x) - x)dx. \tag{3.2}$$

The function f_ϕ^γ cannot be affine for otherwise we could apply Lemma 2.20 with the partition $0 < t_j < 1$ for the breakpoint t_j of any non-flat type-2-active neuron j and obtain a contradiction with ϕ not being a global minimum. In other words, $n \neq 0$. Moreover, since each K_i, $i \in \{1, \dots, n\}$, must contain a non-flat neuron, we deduce from (3.2) that $\int_{q_i}^{q_{i+1}} (f_\phi^\gamma(x) - x)dx = 0$ for all $i \in \{0, \dots, n\}$. It follows from this and $\frac{\partial}{\partial v}\mathcal{L}^\gamma(\phi) = 0$ that (3.1) holds even for flat neurons $j \in \bigcup_i K_i$. Also, Lemma 2.20 implies that the two integrals in (3.1) are not zero. In particular,

$$\int_{I_j} x(f_\phi^\gamma(x) - x)dx \neq \int_{\hat{I}_j} x(f_\phi^\gamma(x) - x)dx$$

for any $j \in \bigcup_i K_i$ and, hence, $\operatorname{sign}(w_{j_0}) = \operatorname{sign}(w_{j_1})$ if j_0 and j_1 belong to the same set K_i. Furthermore, we find from (3.1) that $\operatorname{sign}(w_{j_0}) \neq \operatorname{sign}(w_{j_1})$ for all $i \in \{1, \dots, n-1\}$ and $j_0 \in K_i$, $j_1 \in K_{i+1}$ by taking differences of the integrals $\int_{I_j} x(f_\phi^\gamma(x) - x)dx$ for different j. This establishes item (iii). Consequently, we obtain from Lemma 2.20.(v) (with the partition q_i, q_{i+1}, q_{i+2}) that

$$0 = (q_{i+2} - q_{i+1})^2 - (q_{i+1} - q_i)^2,$$

for all $i \in \{1, \dots, n-2\}$ (note that we do not obtain this equality for $i = 0$ or $i = n-1$). Thus, the points $q_1, \dots, q_n$ are equidistributed in $[q_1, q_n]$ (but not necessarily in $[0, 1]$), which is exactly item (i.a). Next, we prove item (i.b). To do so, we distinguish between even n and odd n. In the former case, $\operatorname{sign}(w_{j_1}) \neq \operatorname{sign}(w_{j_n})$ for all $j_1 \in K_1$, $j_n \in K_n$ by item (iii) and, hence, by (3.1),

$$\int_0^{q_1} x(f_\phi^\gamma(x) - x)dx = \int_{q_n}^1 x(f_\phi^\gamma(x) - x)dx.$$

Write $f_\phi^\gamma(x) = A_i x + B_i$ on $[q_i, q_{i+1}]$, for all $i \in \{0, \dots, n\}$, so that the formulas in (2.4) hold. We compute

$$\begin{aligned}\frac{1}{12}(A_0 - 1)q_1^3 &= \int_0^{q_1} x(f_\phi^\gamma(x) - x)dx \\ &= \int_{q_n}^1 x(f_\phi^\gamma(x) - x)dx = \frac{(-1)^n}{12}(A_0 - 1)q_1(1 - q_n)^2.\end{aligned}$$

Thus, $q_1 = 1 - q_n$ and, by (i.a),

$$\begin{aligned}\int_0^1 x(f_\phi^\gamma(x) - x)dx &= \frac{1}{12}(A_0 - 1)q_1 \sum_{k=0}^{n}(-1)^k(q_{k+1} - q_k)^2 \\ &= \frac{1}{12}(A_0 - 1)q_1\left(2q_1^2 - \left(\frac{1 - 2q_1}{n - 1}\right)^2\right).\end{aligned}$$

Hence, it follows from (3.1) and item (iii) that

$$q_1^2 = \frac{\sigma\gamma^{(1-\sigma)/2}}{1 - \gamma}\left(2q_1^2 - \left(\frac{1 - 2q_1}{n - 1}\right)^2\right).$$

Solving this as a quadratic equation in q_1 under the constraint $q_1 \in (0, 1/2)$ yields $q_1 = \delta^{-1}\gamma^{(1-\sigma)/4}$. Now, assume n is odd. Recall that $\int_{q_i}^{q_{i+2}} x(f_\phi^\gamma(x) - x)dx = 0$ for all $i \in \{1, \dots, n - 2\}$. In particular, $\int_{q_1}^{q_n} x(f_\phi^\gamma(x) - x)dx = 0$. Note that σ is already determined as the sign of w_j for any $j \in K_1$. The partial derivative with respect to w_j being zero for a non-flat neuron $j \in K_1$ implies

$$\begin{aligned}0 &= \int_{q_n}^1 x(f_\phi^\gamma(x) - x)dx + \gamma^\sigma \int_0^{q_1} x(f_\phi^\gamma(x) - x)dx \\ &= -\frac{1}{12}(A_0 - 1)q_1((1 - q_n)^2 - \gamma^\sigma q_1^2).\end{aligned}$$

Thus, $1 - q_n = \gamma^{\sigma/2}q_1$. From this, the formula for q_1 follows in the case $n = 1$. If $n \neq 1$, then we use that the partial derivative with respect to w_j for a non-flat neuron $j \in K_2$ is zero to calculate

$$\begin{aligned}0 &= \int_{q_2}^1 x(f_\phi^\gamma(x) - x)dx + \gamma^{-\sigma} \int_0^{q_2} x(f_\phi^\gamma(x) - x)dx \\ &= \int_{q_{n-1}}^1 x(f_\phi^\gamma(x) - x)dx + \gamma^{-\sigma} \int_0^{q_2} x(f_\phi^\gamma(x) - x)dx \\ &= \frac{1}{12}(A_0 - 1)q_1\left[\gamma^{-\sigma}q_1^2 - (1 - q_n)^2 + (1 - \gamma^{-\sigma})\left(\frac{q_n - q_1}{n - 1}\right)^2\right].\end{aligned}$$

Using $1 - q_n = \gamma^{\sigma/2}q_1$, the term in the rectangular brackets becomes a quadratic polynomial in q_1, and solving for q_1 leads to $q_1 = \delta^{-1}\gamma^{(1-\sigma)/4}$. This finishes item (i.b). From here on, we no longer treat even n and odd n separately. Next, we show item (ii). Given any type-2-active

neuron $j \in \{1,\dots,N\}$, take $i \in \{0,\dots,n\}$ with $q_i \le t_j \le q_{i+1}$ and denote $\tau = \operatorname{sign}(w_j)$. Then, $\frac{\partial}{\partial v_j}\mathcal{L}(\phi) = 0$ implies

$$\begin{aligned} 0 = &\int_{t_j}^{q_{i+1}} (x - t_j)(f_\phi^\gamma(x) - x)dx + \gamma^\tau \int_{q_i}^{t_j} (x - t_j)(f_\phi^\gamma(x) - x)dx \\ &+ \int_{q_{i+1}}^{1} x(f_\phi^\gamma(x) - x)dx + \gamma^\tau \int_{0}^{q_i} x(f_\phi^\gamma(x) - x)dx. \end{aligned} \tag{3.3}$$

A direct computation with the formulas in (2.4) yields

$$\begin{aligned} &\int_{t_j}^{q_{i+1}} (x - t_j)(f_\phi^\gamma(x) - x)dx + \gamma^\tau \int_{q_i}^{t_j} (x - t_j)(f_\phi^\gamma(x) - x)dx \\ &= \frac{(-1)^i}{12}(A_0 - 1)\frac{q_1}{q_{i+1} - q_i}\Big[(q_{i+1} - q_i)^3 - (1 - \gamma^\tau)(t_j - q_i)^2(3q_{i+1} - 2t_j - q_i)\Big]. \end{aligned} \tag{3.4}$$

Furthermore, if $i \neq 0$ and $\tau = \sigma(-1)^{i+1}$, then

$$\begin{aligned} &\int_{q_{i+1}}^{1} x(f_\phi^\gamma(x) - x)dx + \gamma^\tau \int_{0}^{q_i} x(f_\phi^\gamma(x) - x)dx \\ &= -\int_{q_i}^{q_{i+1}} x(f_\phi^\gamma(x) - x)dx + \int_{q_i}^{1} x(f_\phi^\gamma(x) - x)dx + \gamma^\tau \int_{0}^{q_i} x(f_\phi^\gamma(x) - x)dx \\ &= -\int_{q_i}^{q_{i+1}} x(f_\phi^\gamma(x) - x)dx = -\frac{(-1)^i}{12}(A_0 - 1)q_1(q_{i+1} - q_i)^2, \end{aligned}$$

where the second-last equality is implied by $\frac{\partial}{\partial w_k}\mathcal{L}(\phi) = 0$ for a non-flat type-2-active neuron $k \in K_i$. Similarly, if $i \neq n$ and $\tau = \sigma(-1)^{i+2}$, then

$$\begin{aligned} \int_{q_{i+1}}^{1} x(f_\phi^\gamma(x) - x)dx + \gamma^\tau \int_{0}^{q_i} x(f_\phi^\gamma(x) - x)dx &= -\gamma^\tau \int_{q_i}^{q_{i+1}} x(f_\phi^\gamma(x) - x)dx \\ &= -\gamma^\tau \frac{(-1)^i}{12}(A_0 - 1)q_1(q_{i+1} - q_i)^2. \end{aligned}$$

The remaining cases are $i \in \{0, n\}$ with $\tau = -\sigma$, respectively $\tau = \sigma(-1)^n$, for which

$$\begin{aligned} &\int_{q_{i+1}}^{1} x(f_\phi^\gamma(x) - x)dx + \gamma^\tau \int_{0}^{q_i} x(f_\phi^\gamma(x) - x)dx \\ &= \frac{(-1)^{n-i}}{12}(A_0 - 1)\gamma^{i\tau/n}q_1 \cdot \begin{cases} \gamma^{(n-i)\sigma/n}q_1^2 & \text{if } n \text{ is odd,} \\ q_1^2 - (q_2 - q_1)^2 & \text{if } n \text{ is even.} \end{cases} \end{aligned}$$

In conclusion, we obtain from (3.3) and (3.4) that

$$0 = \begin{cases} -(t_j - q_i)^2(3q_{i+1} - 2t_j - q_i) & \text{if } i \neq 0 \text{ and } \tau = \sigma(-1)^{i+1}, \\ (q_{i+1} - q_i)^3 - (t_j - q_i)^2(3q_{i+1} - 2t_j - q_i) & \text{if } i \neq n \text{ and } \tau = \sigma(-1)^{i+2}, \\ (1 - \gamma^\sigma)q_1^3 - (1 - \gamma^{-\sigma})t_j^2(3q_1 - 2t_j) & \text{if } n \text{ is odd}, i = 0, \text{ and } \tau = -\sigma, \\ (1 + \gamma^\sigma)(1 - q_n)q_1^2 - (t_j - q_n)^2(3 - 2t_j - q_n) & \text{if } n \text{ is odd}, i = n, \text{ and } \tau = -\sigma, \\ 2q_1^3 - q_1(q_2 - q_1)^2 - (1 - \gamma^{-\sigma})t_j^2(3q_1 - 2t_j) & \text{if } n \text{ is even}, i = 0, \text{ and } \tau = -\sigma, \\ (1 + \gamma^\sigma)q_1^3 - \gamma^\sigma q_1(q_2 - q_1)^2 - (1 - \gamma^\sigma)(t_j - q_n)^2(3 - 2t_j - q_n) & \text{if } n \text{ is even}, i = n, \text{ and } \tau = \sigma. \end{cases}$$

In the first case, we must have $t_j = q_i$. In the second case, the term can be rewritten as $(q_{i+1} - t_j)^2(q_{i+1} + 2t_j - 3q_i)$, so we must have $t_j = q_{i+1}$. In the third case, the two summands always have opposite signs, so their difference is always strictly positive or strictly negative but not zero. In the fourth case, the right hand side is lower bounded by $(1 - q_n)q_1^2$, so it cannot be zero. In the fifth case, after plugging in q_1 and q_2, we find that t_j must satisfy

$$0 = \sqrt{\gamma}\gamma^{(1+\sigma)/4} + t_j^2\delta^2(3\gamma^{(1-\sigma)/4} - 2t_j\delta).$$

However, there is no solution t_j to this equation with $t_j \in [0, q_1]$. Lastly, in the sixth case, $1 - t_j$ must satisfy the same equation, which is incompatible with $t_j \in [q_n, 1]$. This proves item (ii). Now, we tend to item (iv). Since $\bigcup_i K_i$ contains all type-2-active neurons of ϕ and there are no type-1-active neurons by Lemma 3.2, we can write the slopes of f_ϕ^γ as

$$A_l = \sum_{i=1}^{l} \gamma^{\frac{1+\sigma(-1)^i}{2}} \sum_{j \in K_i} v_j w_j + \sum_{i=l+1}^{n} \gamma^{\frac{1-\sigma(-1)^i}{2}} \sum_{j \in K_i} v_j w_j, \tag{3.5}$$

for all $l \in \{0, \dots, n\}$, by item (iii). With this, we find, for all $i \in \{1, \dots, n\}$,

$$\begin{aligned} -\frac{q_1}{q_{i+1} - q_i}(A_0 - 1) = (-1)^{i-1}(A_i - 1) &= (-1)^{i-1}(A_{i-1} - 1) + \sigma(1-\gamma)\sum_{j \in K_i} v_j w_j \\ &= \frac{q_1}{q_i - q_{i-1}}(A_0 - 1) + \sigma(1-\gamma)\sum_{j \in K_i} v_j w_j. \end{aligned}$$

Thus, for all $i \in \{1, \dots, n\}$,

$$\sum_{j \in K_i} v_j w_j = \frac{-\sigma}{1-\gamma}(A_0 - 1) q_1 \frac{q_{i+1} - q_{i-1}}{(q_{i+1} - q_i)(q_i - q_{i-1})}.$$

Combining this with the formula (3.5) for A_0 yields

$$\frac{-\sigma(1-\gamma)}{A_0 - 1} = \sigma(1-\gamma) + q_1 \sum_{i=1}^{n} \gamma^{\frac{1-\sigma(-1)^i}{2}} \frac{q_{i+1} - q_{i-1}}{(q_{i+1} - q_i)(q_i - q_{i-1})} = \gamma^{(1-\sigma)/4}\delta. \tag{3.6}$$

Plugging this back into the formula for $\sum_{j \in K_i} v_j w_j$, we obtain for $n = 1$ that $\sum_{j \in K_1} v_j w_j = \gamma^{-1/2}$ and for $n \geq 2$, $i \in \{1, \dots, n\}$ that

$$\begin{aligned} \sum_{j \in K_i} v_j w_j &= \frac{1}{\delta^2} \frac{q_{i+1} - q_{i-1}}{(q_{i+1} - q_i)(q_i - q_{i-1})} \\ &= \begin{cases} \delta^{-1}\big((1+\gamma)^{-1/2} + \gamma^{-(1-\sigma)/4}\big) & \text{if } i = 1, \\ 2\delta^{-1}(1+\gamma)^{-1/2} & \text{if } 2 \leq i \leq n-1, \\ \delta^{-1}\big((1+\gamma)^{-1/2} + \gamma^{-(1-\sigma(-1)^n)/4}\big) & \text{if } i = n. \end{cases} \end{aligned}$$

This establishes item (iv). By the formulas in (2.4) and (3.6),

$$A_i - 1 = \sigma(-1)^{i+1}(1-\gamma)\delta^{-1} \cdot \begin{cases} \gamma^{-(1-\sigma)/4} & \text{if } i = 0, \\ (1+\gamma)^{-1/2} & \text{if } 1 \leq i \leq n-1, \\ \gamma^{-(1-\sigma(-1)^n)/4} & \text{if } i = n \end{cases}$$

and

$$B_i = \frac{1}{2}\sigma(-1)^i(1-\gamma)\delta^{-2} \cdot \begin{cases} 1 & \text{if } i = 0, \\ 2i - 1 + 2\gamma^{(1-\sigma)/4}(1+\gamma)^{-1/2} & \text{if } 1 \leq i \leq n-1, \\ 2\gamma^{-(1-\sigma(-1)^n)/4}\delta - 1 & \text{if } i = n. \end{cases}$$

In particular, item (vi) holds. Lastly, we know from Lemma 3.2 and item (iii) that

$$0 = f_\phi^\gamma(0) - B_0 = c - \sum_{i=1}^{n} \gamma^{\frac{1-\sigma(-1)^i}{2}} q_i \sum_{j \in K_i} v_j w_j - B_0.$$

After plugging in the formulas for B_0, δ, q_i, and $\sum_{j\in K_i} v_j w_j$, a lengthy but straight-forward computation results in $c = 1/2$. Thus, ϕ is centered, which concludes the proof. ■

We make a few remarks about the relationship between the previous and Lemma 2.23. The quantity δ in Lemma 3.3 replaces the factor $n+1$ that appears throughout Lemma 2.23. In the limit $\gamma \to 0$,

$$\delta \to \begin{cases} n & \text{if } n \text{ is odd}, \\ n+1 & \text{if } n \text{ is even and } \sigma = 1, \\ n-1 & \text{if } n \text{ is even and } \sigma = -1. \end{cases}$$

Thus, in order to match Lemma 2.23 with the limit case $\gamma \to 0$ of Lemma 3.3, one would need to apply the former lemma with

$$\tilde{n} = \begin{cases} n-1 & \text{if } n \text{ is odd}, \\ n & \text{if } n \text{ is even and } \sigma = 1, \\ n-2 & \text{if } n \text{ is even and } \sigma = -1 \end{cases}$$

in place of n so that $\delta \to \tilde{n}+1$. One would hope that the quantities in Lemma 3.3 converge to their counterparts from Lemma 2.23 with $\tilde{n}$ as $\gamma \to 0$. Although the number of breakpoints in each lemma is different in most cases (i.e. $n \neq \tilde{n}$), this convergence actually happens: on the one hand, if n is odd and $\sigma = 1$, then $q_n \to 1$ 'degenerates' into the endpoint of the interval $[0,1]$ and only the $(n-1)$-many breakpoints $q_1, \ldots, q_{n-1}$ remain, which converge to $\frac{i}{\tilde{n}+1}$, $i \in \{1, \ldots, \tilde{n}\}$. Similarly, if n is odd and $\sigma = -1$, then $q_1 \to 0$ degenerates and $q_2, \ldots, q_n$ remain and converge to the correct breakpoints $\frac{i}{\tilde{n}+1}$, $i \in \{1, \ldots, \tilde{n}\}$. On the other hand, if n is even and $\sigma = 1$, then none of the breakpoints degenerate and $q_1, \ldots, q_n$ remain and converge. Lastly, if n is even and $\sigma = -1$, then both $q_1 \to 0$ and $q_n \to n$, and we are left with $q_2, \ldots, q_{n-1}$, which converge.

In addition, note that the parity of the w-coordinate of the type-2-active neurons match in each lemma even though these are $\sigma(-1)^{i+1}$ and $(-1)^{i+1}$, respectively. They match because q_1 can only degenerate into 0 if $\sigma = -1$. Lastly, note that the quantities $\sum_{j\in K_i} v_j w_j$ also converge to their counterparts as $\gamma \to 0$.

Lemma 3.4. *Suppose $\phi \in \mathbb{R}^{3N+1}$ is a critical point or a local extremum of $\mathcal{L}^\gamma$ but not a global minimum and that ϕ has a type-2-active neuron. There exists $\gamma_0 \in (0,1]$ depending only on N such that if $\gamma < \gamma_0$, then ϕ is a saddle point of $\mathcal{L}^\gamma$.*

Recall that, in the proof of Lemma 2.24, we studied the Hessian of $\mathcal{L}^\gamma$ restricted to a suitable set of coordinates, taken from type-2-active neurons with breakpoints $\frac{i}{n+1}$, $i \in \{1,2\}$. To prove Lemma 3.4, we proceed analogously, which works for sufficiently small γ by the above observation about Lemmas 2.23 and 3.3. More precisely, if $n \neq 1$ and $\sigma = 1$, then we will be able to work with the same set of coordinates because $q_1 \to \frac{1}{\tilde{n}+1}$ and $q_2 \to \frac{2}{\tilde{n}+1}$. On the other hand, if $n \geq 3$ and $\sigma = -1$, then $q_1 \to 0$ but $q_2 \to \frac{1}{\tilde{n}+1}$ and $q_3 \to \frac{2}{\tilde{n}+1}$. In this case, we will use the analogous set of coordinates with q_2 and q_3 in place of q_1 and q_2. However, the argument does not work if $n = 1$ or if $n = 2$ and $\sigma = -1$ because then $q_1 \to 0$, $q_2 \to 1$, and f_ϕ^γ becomes an affine function as $\gamma \to 0$. We will treat these two cases separately.

Proof of Lemma 3.4. Take n, δ, $q_1, \dots, q_n$, and σ from Lemma 3.3. First, assume $n = 2$ with $\sigma = 1$ or $n \geq 3$. Abbreviate $\tau = (3-\sigma)/2 \in \{1,2\}$. Similar to the proof of Lemma 2.24, let $K_\tau \subseteq \{1, \dots, N\}$ denote the set of those type-2-active neurons with breakpoint q_τ, and let $K_\tau^- \subseteq K_\tau$ be the subset of those neurons $j \in K_\tau$ with $v_j < 0$. Let $j_1 \in K_\tau$ with $v_{j_1} > 0$, which exists since $a := \sum_{j \in K_\tau} v_j w_j > 0$ and $w_j > 0$ for all $j \in K_\tau$, and let $j_2, \dots, j_l$, for $l \in \{1, \dots, N\}$, be an enumeration of K_τ^-. Moreover, let $k \in \{1, \dots, N\}$ be any type-2-active neuron with breakpoint $q_{\tau+1}$. As in the ReLU case, we consider the Hessian H of $\mathcal{L}^\gamma$ restricted to $(b_{j_1}, \dots, b_{j_l}, v_k, c)$.

We again introduce some shorthand notation. For all $i \in \{1, \dots, l\}$, denote $\lambda_i = a^{-1} v_{j_i} w_{j_i}$ so that $\sum_{i=1}^l \lambda_i \leq 1$. Define $\mu = \frac{1}{2}(1 - (1-\gamma^2)q_\tau)^{-1} > 0$ and the vectors $u_1 = (v_{j_1}, \dots, v_{j_l})$,

$$u_2 = \mu \begin{pmatrix} w_k\big(\gamma(1-2q_{\tau+1}) - (1-\gamma)(q_{\tau+1}-q_\tau)^2\big) \\ 2(1-(1-\gamma)q_\tau) \end{pmatrix},$$

and $u = (u_1, u_2)$. Further, let D be the diagonal matrix with entries $-\mu(1-\gamma)^2 v_{j_i}^2/(a\delta^2\lambda_i)$, $i \in \{1, \dots, l\}$, let A be the Hessian of $\mathcal{L}^\gamma$ restricted to (v_k, c), let $B = \mu A - u_2 u_2^T$, and let E be the diagonal block matrix with blocks D and B. Then $H = \frac{1}{\mu}(E + uu^T)$. The matrix A is

$$A = \begin{pmatrix} \frac{2}{3} w_k^2 \big(q_{\tau+1}^3 + \gamma^2(1-q_{\tau+1})^3\big) & -w_k\big(q_{\tau+1}^2 - \gamma(1-q_{\tau+1})^2\big) \\ -w_k\big(q_{\tau+1}^2 - \gamma(1-q_{\tau+1})^2\big) & 2 \end{pmatrix},$$

of which both the determinant and the upper left entry are strictly positive. In particular, A is positive definite and, hence, $\Gamma := \frac{1}{\mu} u_2^T A^{-1} u_2$ is strictly positive. If $\Gamma < 1$, then the same considerations as in the proof of Lemma 2.24 show that B and E are invertible and

$$\begin{aligned} \det(H) &= \mu^{-(l+2)}(1 + u_1^T D^{-1} u_1 + u_2^T B^{-1} u_2) \det(E) \\ &= \Delta\Big(\frac{a}{\mu}\Big(\frac{\delta}{1-\gamma}\Big)^2 (1-\Gamma) \sum_{i=1}^l \lambda_i - 1\Big), \end{aligned}$$

where $\Delta = -\mu^{(l+2)}(1-\Gamma)^{-1} \det(D) \det(B) > 0$. So far, we did not impose any restrictions on γ. To verify that $\Gamma < 1$, we use the limit argument to reduce the calculation to the one we performed in the proof of Lemma 2.24. To this end, we point out that Γ is independent of w_k and that δ, $q_\tau, q_{\tau+1}$, and μ only depend on n and γ. For fixed n, if we let γ tend to zero, then $\delta \to \tilde{n}+1$, $q_\tau \to \frac{1}{\tilde{n}+1}$, $q_{\tau+1} \to \frac{2}{\tilde{n}+1}$, and $\mu \to \frac{\tilde{n}+1}{2\tilde{n}}$, where we take $\tilde{n} = n - 1 + \sigma$ if n is even and $\tilde{n} = n-1$ if n is odd. These limits coincide with the corresponding objects from the proof of Lemma 2.24 with $\tilde{n}$ in place of n as discussed prior to stating Lemma 3.4. The

same goes for the limits of a, u_2, and A. Thus, we find from (2.6) that, for sufficiently small γ,

$$\Gamma \approx \frac{32\tilde{n}^2 - 21\tilde{n} + 3}{16\tilde{n}(2\tilde{n}-1)} < 1 \quad \text{and} \quad \frac{a}{\mu}\Big(\frac{\delta}{1-\gamma}\Big)^2(1-\Gamma) \approx 4\tilde{n}(1-\Gamma) \approx \frac{5\tilde{n}-3}{8\tilde{n}-4} < 1.$$

This concludes the existence of a $\gamma_0 \in (0,1]$ such that if $\gamma < \gamma_0$, then $\det(H) < 0$. This γ_0 depends only on n. Since $n \leq N$, we can shrink γ_0 if necessary so that it depends only on N.

It remains to treat the cases $n = 1$ and $n = 2$ with $\sigma = -1$. Assume $n = 1$. This time, let $j_1 \in \{1, \dots, N\}$ be any type-2-active neuron with $\operatorname{sign}(v_{j_1}) = \sigma$, and let $j_2, \dots, j_l$, for $l \in \{1, \dots, N\}$, be an enumeration of all type-2-active neurons with $\operatorname{sign}(v_{j_1}) = -\sigma$. As before, let $a = \gamma^{-1/2}$, $\lambda_i = a^{-1}v_{j_i}w_{j_i}$, $\mu = \frac{1}{2}\gamma^{-1/2}(1-\sqrt{\gamma}+\gamma)^{-1}$, $D_i = -\mu(1-\gamma)^2 v_{j_i}^2/(a\delta^2\lambda_i)$, and $u_1 = (v_{j_1}, \dots, v_{j_l})$ so that $\sum_{i=1}^{l}\lambda_i \leq 1$ and $\det(D) < 0$. On the other hand, let $u_2 = \sigma\mu\sqrt{\gamma}(1-\gamma)\lambda_1/(\delta^2 v_{j_1})$ and $B = \mu\frac{\partial^2}{\partial v_{j_1}^2}\mathcal{L}^\gamma(\phi) - u_2^2 = \frac{1}{3}\mu^2\gamma\lambda_1^2 v_{j_1}^{-2} > 0$. Then the Hessian of $\mathcal{L}^\gamma$ restricted to the coordinates $(b_{j_1}, \dots, b_{j_l}, v_{j_1})$ is $H = \frac{1}{\mu}(E + uu^T)$, where E is the diagonal block matrix with blocks D and B. Hence,

$$\begin{aligned}\det(H) &= \mu^{-(l+1)}B\det(D)(1 + u_1^T D^{-1}u_1 + u_2^2/B)\\ &= -\mu^{-(l+1)}B\det(D)\frac{4(1-\sqrt{\gamma}+\gamma)}{(1+\sqrt{\gamma})^2}\left(\frac{1}{2}\left(\frac{1+\sqrt{\gamma}}{1-\sqrt{\gamma}}\right)^2\sum_{i=1}^{l}\lambda_i - 1\right).\end{aligned}$$

In particular, $\det(H) < 0$ for sufficiently small γ.

Lastly, assume $n = 2$ and $\sigma = -1$. Similar as in the beginning, let $K_1 \subseteq \{1, \dots, N\}$ denote the set of those type-2-active neurons with breakpoint q_1, and let $K_1^+ \subseteq K_1$ be the subset of those neurons $j \in K_1$ with $v_j > 0$. Let $j_1 \in K_1$ with $v_{j_1} < 0$, which exists since $a = \sum_{j\in K_1} v_j w_j > 0$ and $w_j < 0$ for all $j \in K_1$, and let $j_2, \dots, j_l$, for $l \in \{1, \dots, N\}$, be an enumeration of K_1^+. Further, denote the same shorthand $\lambda_i = a^{-1}v_{j_i}w_{j_i}$ and $u_1 = (v_{j_1}, \dots, v_{j_l})$ but set $\mu = \frac{3}{2}(q_1^3 + \gamma^2 - \gamma^2 q_1^3)^{-1}$ and $D_i = -\mu(1-\gamma)^2 q_1^2 v_{j_i}^2/(a\delta^2\lambda_i)$. Then the Hessian of $\mathcal{L}^\gamma$ restricted to $(w_{j_1}, \dots, w_{j_l})$ is $H = \frac{1}{\mu}(D + u_1 u_1^T)$ with determinant

$$\det(H) = \mu^{-l}(1 + u_1^T D^{-1}u_1)\det(D) = -\mu^{-l}\det(D)\left(\frac{a\delta^2}{\mu(1-\gamma)^2 q_1^2}\sum_{i=1}^{l}\lambda_i - 1\right).$$

By construction, $\sum_{i=1}^{l}\lambda_i \leq 1$ and, by plugging in the formulas for a, q_1, and δ from Lemma 3.3,

$$\frac{a\delta^2}{\mu(1-\gamma)^2 q_1^2} = \frac{2}{3}\frac{\sqrt{1+\gamma}+\sqrt{\gamma}}{(1-\gamma)^2\sqrt{1+\gamma}}(1 + \sqrt{\gamma}\delta^3 - \gamma^2) = \frac{2}{3}\frac{1}{(1-\gamma)^2} + \mathcal{O}(\sqrt{\gamma}).$$

In particular, $\det(H) < 0$ for small γ. ∎

3.3 Classification for leaky ReLU activation

In the following, we state the classification of critical points of the L^2-loss for leaky ReLU networks. It is almost analogous to Theorem 2.4, but the main difference is the absence of non-global local minima. These critical points vanish for leaky ReLU because they were caused solely by dead ReLU neurons.

Theorem 3.5. *Let $N \in \mathbb{N}$, $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$, $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$, and $T = (T_0, T_1) \in \mathbb{R}^2$ satisfy $\alpha \neq 0$ and $0 \leq T_0 < T_1$. Then there exists $\gamma_0 \in (0, 1]$ such that for all $\gamma \in (0, \gamma_0)$ the following hold:*

(I) *ϕ is not a local maximum of $\mathcal{L}^{\gamma}_{N,T,\mathcal{A}}$.*

(II) *If ϕ is a critical point or a local extremum of $\mathcal{L}^{\gamma}_{N,T,\mathcal{A}}$, then $\mathcal{L}^{\gamma}_{N,T,\mathcal{A}}$ is differentiable at ϕ with gradient $\nabla \mathcal{L}^{\gamma}_{N,T,\mathcal{A}}(\phi) = 0$.*

(III) *ϕ is not a non-global local minimum of $\mathcal{L}^{\gamma}_{N,T,\mathcal{A}}$.*

(IV) *ϕ is a saddle point of $\mathcal{L}^{\gamma}_{N,T,\mathcal{A}}$ if and only if ϕ is $(T, \mathcal{A})$-centered, for all $j \in \{1, \dots, N\}$ the j^{th} hidden neuron of ϕ is flat semi-active, flat inactive with $w_j = 0$, flat degenerate, or type-2-active, and exactly one of the following two items holds:*

(a) *ϕ does not have any type-2-active neurons.*

(b) *There exist $\sigma \in \{-1, 1\}$, $n \in \mathbb{N}$ such that if*

$$\delta = \gamma^{(1-\sigma)/4} + \gamma^{(1-\sigma(-1)^n)/4} + (n-1)\sqrt{1+\gamma},$$
$$q_i = T_0 + \frac{T_1 - T_0}{\delta}\big(\gamma^{(1-\sigma)/4} + (i-1)\sqrt{1+\gamma}\big), \qquad i \in \{1, \dots, n\},$$

then $\bigcup_{j \in \{1,\dots,N\},\, w_j \neq 0} \{-b_j/w_j\} = \{q_1, \dots, q_n\}$ and, for all $j \in \{1, \dots, N\}$, $i \in \{1, \dots, n\}$ with $w_j \neq 0 = b_j + w_j q_i$, it holds that $\operatorname{sign}(w_j) = \sigma(-1)^{i+1}$ and

$$\sum_{\substack{k \in \{1,\dots,N\},\\ w_k \neq 0 = b_k + w_k q_i}} v_k w_k = \begin{cases} \frac{\alpha}{\sqrt{\gamma}} & \text{if } i = 1 = n, \\ \frac{\alpha}{\delta}\big(\frac{1}{\sqrt{1+\gamma}} + \frac{1}{\gamma^{(1-\sigma)/4}}\big) & \text{if } i = 1 \neq n, \\ \frac{\alpha}{\delta}\frac{2}{\sqrt{1+\gamma}} & \text{if } 2 \leq i \leq n-1, \\ \frac{\alpha}{\delta}\big(\frac{1}{\sqrt{1+\gamma}} + \frac{1}{\gamma^{(1-\sigma(-1)^n)/4}}\big) & \text{if } i = n \neq 1. \end{cases}$$

(V) *If ϕ is a saddle point of $\mathcal{L}^{\gamma}_{N,T,\mathcal{A}}$ without type-2-active neurons, then $f^{\gamma}_{\phi}(x) = \frac{\alpha}{2}(T_0 + T_1) + \beta$ for all $x \in [T_0, T_1]$.*

(VI) *If ϕ is a saddle point of $\mathcal{L}^{\gamma}_{N,T,\mathcal{A}}$ with at least one type-2-active neuron, then there exist $\sigma \in \{-1, 1\}$, $n \in \mathbb{N}$ such that $n \leq N$ and, for all $i \in \{0, \dots, n\}$, $x \in [q_i, q_{i+1}]$, one has*

$$f^{\gamma}_{\phi}(x) - \alpha x - \beta = \left[\frac{-\sigma(-1)^i(1-\gamma)\alpha}{\delta}\right] \cdot \begin{cases} \frac{x - T_0}{\gamma^{(1-\sigma)/4}} - \frac{T_1 - T_0}{2\delta} & \text{if } i = 0, \\ \frac{x - T_0}{\sqrt{1+\gamma}} - \frac{(i-1/2)(T_1 - T_0)}{\delta} - \frac{\gamma^{(1-\sigma)/4}(T_1 - T_0)}{\delta\sqrt{1+\gamma}} & \text{if } 1 \leq i \leq n-1, \\ \frac{x - T_0}{\gamma^{(1-\sigma(-1)^n)/4}} + \frac{T_1 - T_0}{2\delta} - \frac{T_1 - T_0}{\gamma^{(1-\sigma(-1)^n)/4}} & \text{if } i = n, \end{cases}$$

where δ and $q_1, \dots, q_n$ are the same as in item (IV.b).

Proof. We prove Theorem 3.5 in the special case $\mathcal{A} = (1, 0)$ and $T = (0, 1)$. The general case follows from this the same way as Theorem 2.4 followed from Proposition 2.25 in Section 2.9. The first item is shown in Lemma 2.8; see Remark 2.9.

Suppose ϕ is a critical point or a local extremum of $\mathcal{L}^{\gamma}$ but not a global minimum. By Lemma 3.2, all neurons of ϕ are flat semi-active, flat inactive with $w_j = 0$, degenerate, or type-2-active. If, in addition, ϕ does not have any type-2-active neurons, then it also does

not have any non-flat degenerate neurons, it is a saddle point, and ϕ must be centered since $\frac{\partial}{\partial c}\mathcal{L}^\gamma(\phi) = 0$. If, on the other hand, ϕ has a type-2-active neuron, then ϕ is as in item (IV.b) by Lemma 3.3 apart from potentially having non-flat degenerate neurons, and ϕ is a saddle point by Lemma 3.4. However, a posteriori, ϕ cannot have non-flat degenerate neurons because, by Lemma 3.3.(vi),

$$\int_0^1 x(f_\phi^\gamma(x) - x)dx = -\frac{(1-\gamma)^2}{12\delta^4} < 0,$$

so $\frac{\partial^+}{\partial w_j}\mathcal{L}^\gamma(\phi)$ could not be zero for a non-flat degenerate neuron j. This proves item (III) and the 'only if' part in item (IV). This also implies that any critical point or local extremum of $\mathcal{L}^\gamma$ is a global minimum or does not have any non-flat degenerate neurons. Hence, the relation $\mathcal{L}^\gamma = \mathcal{L} \circ P$ with the smooth map P and the differentiability properties of $\mathcal{L}$ assert item (II).

If ϕ is as in item (IV.a), then it clearly is a critical point of $\mathcal{L}^\gamma$, and it is a saddle point by Lemma 3.2. If ϕ is as in item (IV.b), then f_ϕ^γ is given by the formula in item (VI). We can calculate $\int_{q_i}^{q_{i+1}}(f_\phi^\gamma(x) - x)dx = 0$ for all $i \in \{0, \dots, n\}$ and

$$\int_{q_i}^1 x(f_\phi^\gamma(x) - x)dx + \gamma^{\sigma(-1)^{i+1}} \int_0^{q_i} x(f_\phi^\gamma(x) - x)dx = 0$$

for all $i \in \{1, \dots, n\}$. It follows from this that ϕ is a critical point of $\mathcal{L}^\gamma$, and it is a saddle point by Lemma 3.4. This proves the 'if' part in item (IV). Item (V) is immediate and the last item was implicit in the previous step. ∎

Remark 3.6. The restriction on γ to lie in $(0, \gamma_0)$ is only needed in the proof of Lemma 3.4. All other proofs were carried out for general $\gamma \in (0, 1)$. We believe that, in fact, one can take $\gamma_0 = 1$ in Lemma 3.4 and, hence, that Theorem 3.5 also holds for general $\gamma \in (0, 1)$.

4. Classification for quadratic activation

As the last case, we consider the quadratic activation function. The realization $f_\phi^{\text{quad}} \in C(\mathbb{R}, \mathbb{R})$ of a network $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$ with the quadratic activation is

$$f_\phi^{\text{quad}}(x) = c + \sum_{j=1}^N v_j(w_j x + b_j)^2.$$

Given $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$ and $T = (T_0, T_1) \in \mathbb{R}^2$, the loss function $\mathcal{L}_{N,T,\mathcal{A}}^{\text{quad}}$ is the L^2-loss given by

$$\mathcal{L}_{N,T,\mathcal{A}}^{\text{quad}}(\phi) = \int_{T_0}^{T_1} (f_\phi^{\text{quad}}(x) - \alpha x - \beta)^2 dx.$$

This time, there are no issues with differentiability since $\mathcal{L}_{N,T,\mathcal{A}}^{\text{quad}}$ is infinitely times differentiable, even analytic, everywhere. The classification turns out to be simpler than in the ReLU and leaky ReLU case as there are no local extrema and only saddle points with a constant realization function.

Theorem 4.1. *Let $N \in \mathbb{N}$, $\phi = (w, b, v, c) \in \mathbb{R}^{3N+1}$, $\mathcal{A} = (\alpha, \beta) \in \mathbb{R}^2$, and $T = (T_0, T_1) \in \mathbb{R}^2$ satisfy $\alpha \neq 0$ and $T_0 < T_1$. Then the following hold:*

(I) *ϕ is not a local maximum of $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}$.*

(II) *ϕ is not a non-global local minimum of $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}$.*

(III) *ϕ is a global minimum of $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}$ if and only if $N \geq 2$ and $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) = 0$.*

(IV) *ϕ is a saddle point of $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}$ if and only if ϕ is $(T, \mathcal{A})$-centered and, for all $j \in \{1, \dots, N\}$, the j^{th} hidden neuron of ϕ satisfies $v_j b_j = 0 = w_j$ or $w_j \neq v_j = 0 = b_j + \frac{1}{2}(T_0 + T_1)w_j$.*

(V) *If ϕ is a saddle point of $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}$, then $f^{\text{quad}}_\phi(x) = \frac{\alpha}{2}(T_0 + T_1) + \beta$ for all $x \in [T_0, T_1]$.*

Proof. As for the other activation functions, the first item is shown in Lemma 2.8; see Remark 2.9. Now, suppose ϕ is a critical point of $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}$ and $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) > 0$. Since $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}$ is smooth, we have, for any $j \in \{1, \dots, N\}$,

$$\begin{aligned}
0 &= \frac{\partial}{\partial w_j}\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) = 4v_j \int_{T_0}^{T_1} x(w_j x + b_j)(f^{\text{quad}}_\phi(x) - \alpha x - \beta)dx, \\
0 &= \frac{\partial}{\partial b_j}\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) = 4v_j \int_{T_0}^{T_1} (w_j x + b_j)(f^{\text{quad}}_\phi(x) - \alpha x - \beta)dx, \\
0 &= \frac{\partial}{\partial v_j}\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) = 2 \int_{T_0}^{T_1} (w_j x + b_j)^2(f^{\text{quad}}_\phi(x) - \alpha x - \beta)dx, \\
0 &= \frac{\partial}{\partial c}\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) = 2 \int_{T_0}^{T_1} (f^{\text{quad}}_\phi(x) - \alpha x - \beta)dx.
\end{aligned}$$

Thus, if there exists $j \in \{1, \dots, N\}$ with $v_j \neq 0 \neq w_j$, then $\int_{T_0}^{T_1} x^m(f^{\text{quad}}_\phi(x) - \alpha x - \beta)dx = 0$ for all $m \in \{0, 1, 2\}$. However, note that the zero polynomial is the only polynomial p of degree at most two satisfying $\int_{T_0}^{T_1} x^m p(x)dx = 0$ for all $m \in \{0, 1, 2\}$. Hence, since $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) > 0$, we must have $v_j = 0$ or $w_j = 0$ for all neurons. In particular, f^{quad}_ϕ is constant and $\int_{T_0}^{T_1} x(f^{\text{quad}}_\phi(x) - \alpha x - \beta)dx \neq 0$. Thus, for all j, if $v_j \neq 0$, then $b_j = 0$. So far, we have shown that all neurons must satisfy $v_j = 0$ or $w_j = 0 = b_j$. It follows that ϕ is $(T, \mathcal{A})$-centered. For a neuron j with $w_j \neq 0$ and $t_j = -b_j/w_j$, we have

$$0 = 2 \int_{T_0}^{T_1} (w_j x + b_j)^2(c - \alpha x - \beta)dx = -2\alpha w_j^2 \int_{T_0}^{T_1} (x - t_j)^2(x - \tfrac{T_0+T_1}{2})dx,$$

which is true if and only if $t_j = (T_0 + T_1)/2$. This proves the 'only if' direction in (IV). Next, we show that ϕ must be a saddle point. We will pick a path $\phi_s = (w^s, b^s, v^s, c^s)$, $s \in (-1, 1)$, through $\phi = \phi_0$, which differs only in the coordinates of the first neuron and in

$$c^s = c - v_1^s(b_1^s)^2 - \frac{1}{3}A_s(T_0^2 + T_0T_1 + T_1^2) - B_s(T_0 + T_1),$$

where $A_s = v_1^s(w_1^s)^2$ and $B_s = v_1^s w_1^s b_1^s$. Then,

$$\begin{aligned}
\frac{\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_s) - \mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_0)}{(T_1 - T_0)^3} &= \frac{1}{45}A_s^2(4T_0^2 + 7T_0T_1 + 4T_1^2) + \frac{1}{3}A_sB_s(T_0 + T_1) \\
&\quad + \frac{1}{3}B_s^2 - \frac{\alpha}{6}(A_s(T_0 + T_1) + 2B_s).
\end{aligned}$$

We distinguish three cases. First, if $v_1 = 0 \neq w_1$, then we use $w_1^s = w_1$, $b_1^s = b_1 - sw_1$, and $v_1^s = -\text{sign}(\alpha)s^2$. In this case, $B_s = -\frac{1}{2}A_s(T_0 + T_1) - sA_s$ and, hence,

$$\frac{\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_s) - \mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_0)}{(T_1 - T_0)^3} = -\frac{|\alpha|}{3}w_1^2 s^3 + \mathcal{O}(s^4).$$

This is strictly negative for sufficiently small $s > 0$, so ϕ is a saddle point. Secondly, if $v_1 \neq 0 = w_1$, then we use $w_1^s = s$, $b_1^s = -\frac{1}{2}(T_0 + T_1)s + \text{sign}(\alpha v_1)s^2$, and $v_1^s = v_1$. In this case,

$$\frac{\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_s) - \mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_0)}{(T_1 - T_0)^3} = -\frac{|\alpha|}{3}|v_1|s^3 + \mathcal{O}(s^4).$$

In the last case, namely $v_1 = 0 = w_1$, we use $w_1^s = sb_1^s$, $b_1^s = b_1 + s$, and $v_1^s = \text{sign}(\alpha)s^3(b_1^s)^{-2}$. Then,

$$\frac{\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_s) - \mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi_0)}{T_1 - T_0} = -\frac{|\alpha|}{3}s^4 + \mathcal{O}(s^5).$$

We have shown that if ϕ is a critical point with $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) > 0$, then it is a saddle point. This establishes item (II) and it also implies that if ϕ is a global minimum, then $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) = 0$. The latter is only possible if $N \geq 2$. Conversely, if $N \geq 2$, then there are networks with zero loss, so item (III) holds. If ϕ is $(T, \mathcal{A})$-centered and all of its neurons are as in item (IV), then $\nabla\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) = 0$ and ϕ is a saddle point since clearly $\mathcal{L}^{\text{quad}}_{N,T,\mathcal{A}}(\phi) > 0$. This finishes (IV), and (V) follows. ∎

The conditions in Theorem 4.1.(IV) are equivalent to all neurons being flat semi-active, flat inactive with $w_j = 0$, flat type-2-active with breakpoint $-b_j/w_j = (T_0 + T_1)/2$, or degenerate. However, for the quadratic activation, the notions of in-/active neurons seem no longer appropriate.

Remark 4.2. In Theorem 4.1, the case $N = 1$ of a single neuron is special due to the absence of global minima. The loss can still be arbitrarily small, but there is no network achieving the infimum. Indeed, for all $(w, b) \in \mathbb{R}^2$ with $w \neq 0$,

$$\inf_{(v,c)\in\mathbb{R}^2} \mathcal{L}^{\text{quad}}_{1,T,\mathcal{A}}(w, b, v, c) = \frac{\alpha^2(T_1 - T_0)^3}{12}\left(1 - \frac{60\left(\frac{T_0+T_1}{2} + \frac{b}{w}\right)^2}{(T_1 - T_0)^2 + 60\left(\frac{T_0+T_1}{2} + \frac{b}{w}\right)^2}\right) \xrightarrow[\text{monotone}]{\frac{T_0+T_1}{2}+\frac{b}{w}\to\pm\infty} 0.$$

CHAPTER 5

Gradient descent provably escapes saddle points in the training of shallow ReLU networks

This chapter is an adaptation of the preprint [17].

1. Introduction

As discussed in Chapter 1, in this chapter, we intend to apply a stable manifold theorem to analyze the training of neural networks with gradient descent. The intuition behind this theory becomes clearer if one pictures the linearization of the gradient descent map $f(x) = x - \gamma \nabla \mathcal{L}(x)$ (the function describing one step of the algorithm with step size γ and loss function $\mathcal{L}$) around a saddle point z. Note that z, being a critical point of the loss function, is a fixed point of the gradient descent map. For simplicity of the presentation, assume $z = 0$. The first-order Taylor approximation of f around the origin reads $f(x) \approx f'(0)x = (I - \gamma \nabla^2 \mathcal{L}(0))x$, where I denotes the identity matrix. Therefore, after neglecting the second and the higher-order terms, the behavior of the next step $f(x)$ can be determined by looking at x in the eigenspace decomposition of the matrix $f'(0)$. If the saddle point 0 is strict, then $\nabla^2 \mathcal{L}(0)$ has a strictly negative eigenvalue, so $f'(0)$ has an eigenvalue strictly greater than 1. Thus, there is a direction in the linearization along which we move away from the origin. This means that the only way to actually move towards the origin is if one moves *inside* a so-called center-stable manifold. Loosely speaking, a center-stable manifold is a manifold whose tangent space at the origin is the span E^{cs} of the eigenvectors of $f'(0)$ for eigenvalues of absolute value less than or equal to 1. The span E^{cs} is the center-stable space of the linearization, and a[1] center-stable manifold takes into account the second and the higher-order terms. The final step of the approach consists in showing that the set of initializations, from which the gradient descent trajectory eventually enters this center-stable manifold, has measure zero.

In the argument above, we implicitly use that $f'(0)$ is diagonalizable, which is guaranteed if $I - \gamma \nabla^2 \mathcal{L}(0)$ is a real symmetric matrix. But this requires $\mathcal{L}$ to be twice differentiable at the origin. In the framework of ReLU networks, this regularity is not given for the L^2-loss measured against a given target function. To tackle this problem, we have to modify the gradient descent map and consider $f(x) = x - \gamma \mathcal{G}(x)$, where $\mathcal{G}$ is a modification of $\nabla \mathcal{L}$. The function $\mathcal{G}$ may not arise as the gradient of any scalar-valued function. Therefore, we need to ensure explicitly that $\mathcal{G}'$ is symmetric at the origin so that $f'(0)$ is still diagonalizable.

Another (more restrictive) assumption implicitly used in the above argument is that $f'(0)$

[1] While the linear subspace E^{cs} is unique, a center-stable manifold is not; [118].

is non-degenerate. Indeed, if $f'(0)$ is degenerate, then, for x in the kernel of $f'(0)$, nothing can be said about $f(x)$ without considering the second-order terms. In [26,83,84,98,99], this non-degeneracy assumption is guaranteed to hold by requiring $\nabla\mathcal{L}$ to be globally Lipschitz continuous. Then, $I - \gamma\nabla^2\mathcal{L}(0)$ cannot be degenerate for sufficiently small γ compared to the Lipschitz constant.[2] But global Lipschitz continuity of $\nabla\mathcal{L}$ is a strong assumption and does not hold in the aforementioned framework of ReLU networks (nor for networks with other activation functions). In conclusion, one of the main difficulties on the side of the dynamical systems theory is to provide a variant of the center-stable manifold theorem that works even if $f'(0)$ is degenerate. To this end, we extend a result of [100], which we present in Theorem 1.1. Therein, observe that $f'(x)$ need only be non-degenerate almost everywhere but not necessarily at the saddle points $x \in \mathcal{S}$ of interest.

Theorem 1.1. *Let $d \in \mathbb{N}$, let $\|\cdot\| : \mathbb{R}^d \to \mathbb{R}$ be the standard norm on $\mathbb{R}^d$, let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be a function, let $(X_n^x)_{(n,x)\in\mathbb{N}_0\times\mathbb{R}^d} \subseteq \mathbb{R}^d$ be given by $X_0^x = x$ and $X_{n+1}^x = f(X_n^x)$, let $V \subseteq U \subseteq \mathbb{R}^d$ be open sets, assume that $\mathbb{R}^d\backslash V$ has Lebesgue measure zero, assume $f|_U \in C^1(U,\mathbb{R}^d)$, assume that $U \ni x \mapsto f'(x) \in \mathbb{R}^{d\times d}$ is locally Lipschitz continuous, assume for all $x \in V$ that $\det(f'(x)) \neq 0$, let $\mathcal{S} \subseteq \{x \in U\colon f(x) = x\}$, and assume for all $x \in \mathcal{S}$ that the matrix $f'(x)$ is symmetric and has an eigenvalue whose absolute value is strictly greater than 1. Then, the set $\{x \in \mathbb{R}^d\colon (\exists\, y \in \mathcal{S}\colon \limsup_{n\to\infty} \|X_n^x - y\| = 0)\}$ has Lebesgue measure zero.*

Concerning the ReLU network application, the main challenge is to construct a suitable modification $\mathcal{G}$ of $\nabla\mathcal{L}$ mentioned above and verify that it suits our purpose. Of course, we have to ensure that, upon replacing $\nabla\mathcal{L}$ by its modification $\mathcal{G}$, we do not loose information about the dynamics of the original gradient descent algorithm. To obtain the necessary strictness of (in some sense) most saddle points, we rely on the classification of saddle points from Chapter 4. To apply this classification, we need to restrict our attention to shallow ReLU networks on the L^2-loss with respect to an affine target function. Combining all of the above, we prove in Theorem 3.7 that the gradient descent algorithm almost surely avoids most saddle points in this framework, where almost surely is understood with respect to a random initialization that is absolutely continuous with respect to the Lebesgue measure.

Building more intricately on the classification of critical points from Chapter 4, we proceed to deduce convergence of the algorithm to a global minimum under a suitable initialization as stated in Theorem 1.2 below. Let us explain the notation used in that theorem. A shallow network with N hidden neurons is a collection of weights and biases, represented by a vector $\theta \in \mathbb{R}^{3N+1}$. The realization of such a network is the function $\mathcal{R}_\theta$. The map $\mathcal{L}$ is the squared L^2-loss measured against a target function $\mathfrak{f}$. As $\mathcal{L}$ is not differentiable everywhere, we take $\mathcal{G}$ to be the left gradient of $\mathcal{L}$, that is we take partial directional derivatives from the left. This specific choice is for the sake of the presentation, but in the main body of this chapter $\mathcal{G}$ may take coordinate-wise any values when $\mathcal{L}$ is not differentiable. Finally, $\Theta_{k+1}^{\gamma,\theta} = \Theta_k^{\gamma,\theta} - \gamma\mathcal{G}(\Theta_k^{\gamma,\theta})$ is the gradient descent algorithm with step size γ and initial value θ.

Theorem 1.2. *Let $N \in \mathbb{N}$, $\alpha, \beta \in \mathbb{R}$ satisfy $\alpha < \beta$ and $N/2 \in \mathbb{N}$, for every $\theta =$*

[2] We remark that local Lipschitz continuity at the origin is sufficient to guarantee that $I - \gamma\nabla^2\mathcal{L}(0)$ is non-degenerate for small γ. But we want to study many saddle points in an unbounded set simultaneously, and γ would depend on the local Lipschitz constant around each of those saddle points. Therefore, to guarantee $\gamma \neq 0$, we would need a uniform upper bound on these local Lipschitz constants, which essentially amounts to a global bound.

$(\theta_1, \dots, \theta_{3N+1}) \in \mathbb{R}^{3N+1}$ let $\mathcal{R}_\theta \in C([\alpha, \beta], \mathbb{R})$ be given by

$$\mathcal{R}_\theta(x) = \theta_{3N+1} + \sum_{j=1}^{N} \theta_{2N+j} \max\{\theta_j x + \theta_{N+j}, 0\},$$

let $\mathfrak{f} \in C([\alpha, \beta], \mathbb{R})$ be affine, let $\mathcal{L} \in C(\mathbb{R}^{3N+1}, \mathbb{R})$ be given by $\mathcal{L}(\theta) = \int_\alpha^\beta (\mathcal{R}_\theta(x) - \mathfrak{f}(x))^2 \, dx$, let $\mathcal{G} \colon \mathbb{R}^{3N+1} \to \mathbb{R}^{3N+1}$ be the left gradient of $\mathcal{L}$, and let $(\Theta_k^{\gamma,\theta})_{(k,\gamma,\theta) \in \mathbb{N}_0 \times (0,\infty) \times \mathbb{R}^{3N+1}} \subseteq \mathbb{R}^{3N+1}$ be given by $\Theta_0^{\gamma,\theta} = \theta$ and $\Theta_{k+1}^{\gamma,\theta} = \Theta_k^{\gamma,\theta} - \gamma \mathcal{G}(\Theta_k^{\gamma,\theta})$. Then, for Lebesgue almost all $\gamma \in (0, \infty)$ and Lebesgue almost all

$$\theta \in \left\{ \vartheta \in \mathbb{R}^{3N+1} \colon (\Theta_k^{\gamma,\vartheta})_{k \in \mathbb{N}_0} \text{ is convergent and } \lim_{k \to \infty} \mathcal{L}(\Theta_k^{\gamma,\vartheta}) < \tfrac{[\mathfrak{f}'(\alpha)]^2 (\beta - \alpha)^3}{12(N-1)^4} \right\} \tag{1.1}$$

it holds that $\lim_{k \to \infty} \mathcal{L}(\Theta_k^{\gamma,\theta}) = 0$.

We remark that the conclusion of Theorem 1.2 is void if the target function $\mathfrak{f}$ is constant. In this case, every critical point of $\mathcal{L}$ is a global minimum and there is nothing to prove; [14].

The remainder of this chapter is organized as follows. In Section 2, we state our variant of the center-stable manifold theorem and deduce Theorem 1.1. Section 3 introduces the shallow ReLU network framework and begins with constructing the previously mentioned modification of the gradient of the loss function. Then, Section 3.1 is devoted to almost everywhere non-degeneracy of the Jacobian of the modified gradient descent map, and Section 3.2 deals with strictness of saddle points. In Section 3.3, we deduce Theorem 1.2. Finally, Section 4 contains the proof of the center-stable manifold theorem.

Notation

We denote by $\|\cdot\|$ the Euclidean norm when applied to vectors and the operator norm induced by the Euclidean norm when applied to matrices. Throughout this chapter, we fix a dimension $d \in \mathbb{N}$ and write $I \in \mathbb{R}^{d \times d}$ for the identity matrix. The closed ball around a point $x \in \mathbb{R}^d$ with radius $r \in (0, \infty)$ is denoted $\mathbb{B}_r(x) = \{y \in \mathbb{R}^d \colon \|y - x\| \leq r\}$. A discrete dynamical system is written as follows. For every function $f \colon \mathbb{R}^d \to \mathbb{R}^d$, we denote by $f^k \colon \mathbb{R}^d \to \mathbb{R}^d$, $k \in \mathbb{N}_0$, the functions that satisfy for all $k \in \mathbb{N}_0$ that $f^0 = \mathrm{id}_{\mathbb{R}^d}$ and $f^{k+1} = f \circ f^k$. To describe critical points of a function $\mathcal{L} \colon \mathbb{R}^d \to \mathbb{R}$, we use the following terminology. Local extrema refer to nonstrict local extrema; a point $x \in \mathbb{R}^d$ is called a critical point of $\mathcal{L}$ if $\mathcal{L}$ is differentiable at x with $\nabla \mathcal{L}(x) = 0$; and a critical point is called a saddle point if it is not a local extremum.

2. A center-stable manifold theorem

The core of this section is a variant of the stable manifold theorem. The novelty is that we do not require the dynamical system to be a local diffeomorphism as is the case in the classical formulation [118]. Specifically, the Jacobian may be degenerate at the fixed point under consideration. This comes at the expense of less regularity of the center-stable manifold. Indeed, the graph in Theorem 2.2 is only proved to be Lipschitz-regular. Our variant is an extension of the corresponding statement in [100]. The exact regularity requirement needed is a certain local Lipschitz condition on the remainder term of the first-order Taylor expansion of the dynamical system around a fixed point:

Assumption 2.1. Let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be a function and let $\mathcal{S} \subseteq \{x \in \mathbb{R}^d\colon f(x) = x\}$. Assume for all $z \in \mathcal{S}$ that f is differentiable at z, that the matrix $f'(z) \in \mathbb{R}^{d\times d}$ is diagonalizable over $\mathbb{R}$ and has an eigenvalue of absolute value strictly greater than 1, and that for all $\varepsilon \in (0,\infty)$ there exists $r_\varepsilon \in (0,\infty)$ so that the map $\mathbb{B}_{r_\varepsilon}(z) \to \mathbb{R}^d$, $x \mapsto f(x) - z - f'(z)(x-z)$ is ε-Lipschitz continuous.

For all $z \in \mathcal{S}$, denote by $E_z^{cs} \subseteq \mathbb{R}^d$ the span of those eigenvectors of $f'(z)$ associated with eigenvalues that lie in $[-1,1]$ (the center-stable space) and by $E_z^u \subseteq \mathbb{R}^d$ the span of those eigenvectors of $f'(z)$ associated with eigenvalues that lie in $\mathbb{R}\backslash[-1,1]$ (the unstable space). Then, $\mathbb{R}^d = E_z^{cs} \oplus E_z^u$. Under Assumption 2.1, we have $0 \leq \dim(E_z^{cs}) \leq d-1$ for all $z \in \mathcal{S}$. Now, we can state our version of the center-stable manifold theorem.

Theorem 2.2 (Center-stable Lipschitz manifold)**.** *Let Assumption 2.1 hold and let $z \in \mathcal{S}$. Then, there exists an $r \in (0,\infty)$ and a Lipschitz continuous map $\Psi\colon E_z^{cs} \to E_z^u$ such that*

$$\{x \in \mathbb{R}^d\colon f^k(x) \in \mathbb{B}_r(z) \text{ for all } k \in \mathbb{N}_0\} \subseteq \operatorname{Graph}(\Psi).$$

This theorem states that all those points, whose orbits under the dynamical system remain close to z, lie in the graph of a Lipschitz function, whose domain is a linear space of dimension between 0 and $d-1$. We defer the proof to Section 4. In Theorem 2.2, we considered a single point $z \in \mathcal{S}$. We obtain a statement about all points in $\mathcal{S}$ simultaneously the same way it was done in [83, 99], using second-countability of Euclidean space. For completeness, we repeat the argument to prove Corollary 2.3.

Corollary 2.3. *Let Assumption 2.1 hold. Then, there exists a set $W \subseteq \mathbb{R}^d$ of Lebesgue measure zero such that*

$$\left\{x \in \mathbb{R}^d\colon \lim_{k\to\infty} f^k(x) \in \mathcal{S}\right\} \subseteq \bigcup_{k,n\in\mathbb{N}_0} f^{-k}(W).$$

Proof. By Theorem 2.2, for all $z \in \mathcal{S}$, there exists an open neighborhood $U_z \subseteq \mathbb{R}^d$ of z and a Lipschitz continuous map $\Psi_z\colon E_z^{cs} \to E_z^u$ such that $\{x \in \mathbb{R}^d\colon f^k(x) \in U_z \text{ for all } k \in \mathbb{N}_0\} \subseteq \operatorname{Graph}(\Psi_z)$. Now, $\bigcup_{z\in\mathcal{S}} U_z$ is an open cover of $\mathcal{S}$ and, by second-countability of $\mathbb{R}^d$, there exists a countable subcover $\bigcup_{n\in\mathbb{N}_0} U_{z_n}$. Set $W = \bigcup_{n\in\mathbb{N}_0} \operatorname{Graph}(\Psi_{z_n})$. If $y \in \{x \in \mathbb{R}^d\colon \lim_{k\to\infty} f^k(x) \in \mathcal{S}\}$, then there exist $k,n \in \mathbb{N}_0$ such that for all $m \in \mathbb{N}_0$ we have $f^m(f^k((y)) = f^{m+k}(y) \in U_{z_n}$. Thus, $f^k(y) \in \operatorname{Graph}(\Psi_{z_n})$ and, hence, $y \in f^{-k}(W)$. Lastly, the set W, being a countable union of graphs, has Lebesgue measure zero. ∎

Note that Corollary 2.3 is a statement about the stable set of $\mathcal{S}$ and not its center-stable set. However, the proof of the corollary relies on the center-stable manifolds from Theorem 2.2 and would not work with the stable manifolds.

The goal is to show, under reasonable assumptions, that the set $\{x \in \mathbb{R}^d\colon \lim_{k\to\infty} f^k(x) \in \mathcal{S}\}$ has Lebesgue measure zero. This follows from the previous corollary if we can ensure that preimages of measure zero sets under f^k have themselves measure zero. This is certainly true for local diffeomorphisms (see [83, 99]), but we do not want to exclude the possibility that the dynamical system has a degenerate Jacobian at points in $\mathcal{S}$. Fortunately, it is sufficient to have a non-degenerate Jacobian almost everywhere (but potentially at no point in $\mathcal{S}$), as the next lemma shows.

Lemma 2.4. *Let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be a function and suppose there exists an open set $V \subseteq \mathbb{R}^d$, whose complement has Lebesgue measure zero, such that f is continuously differentiable on V with $\det(f'(x)) \neq 0$ for all $x \in V$. Then, for any set $W \subseteq \mathbb{R}^d$ of Lebesgue measure zero, the set $f^{-1}(W)$ also has Lebesgue measure zero.*

Proof. First, note that f is Lebesgue measurable because it is continuous on a subset of full measure. It suffices to show that the set $f^{-1}(W) \cap V$ has Lebesgue measure zero. By the assumptions, the restriction of f to V is a local C^1-diffeomorphism. This and second-countability of $\mathbb{R}^d$ guarantee the existence of a countable open cover $\bigcup_{n\in\mathbb{N}} D_n = V$ of V such that, for all $n \in \mathbb{N}$, the restriction $f|_{D_n}\colon D_n \to f(D_n)$ is a C^1-diffeomorphism. By the integral transformation theorem, we have, for all $n \in \mathbb{N}$,

$$\int_{f^{-1}(W)\cap D_n} |\det(f'(x))|\, dx = \int_{f(f^{-1}(W)\cap D_n)} dx \leq \int_W dx = 0.$$

Since $\det(f'(x)) \neq 0$ for all $x \in V$, this implies that $f^{-1}(W) \cap D_n$ has Lebesgue measure zero for all $n \in \mathbb{N}$ and, hence, so does $f^{-1}(W) \cap V$. ∎

With Corollary 2.3, we conclude that if Assumption 2.1 holds as well as the assumption of Lemma 2.4, then the stable set $\{x \in \mathbb{R}^d\colon \lim_{k\to\infty} f^k(x) \in \mathcal{S}\}$ of $\mathcal{S}$ has Lebesgue measure zero. We finish this section by applying this result to a class of dynamical systems that includes the gradient descent algorithm, which will be of interest in the next section.

Proposition 2.5. *Let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be a function and suppose there exist open sets $V \subseteq U \subseteq \mathbb{R}^d$, whose complements have Lebesgue measure zero, such that f is continuously differentiable on U with a locally Lipschitz continuous Jacobian, which is non-degenerate on V. Let $\mathcal{S} \subseteq \{x \in U\colon f(x) = x\}$ and assume for all $x \in \mathcal{S}$ that $f'(x)$ is symmetric and has an eigenvalue of absolute value strictly greater than 1. Then, the set $\{x \in \mathbb{R}^d\colon \lim_{k\to\infty} f^k(x) \in \mathcal{S}\}$ has Lebesgue measure zero.*

Proof. The result is immediate from Corollary 2.3 and Lemma 2.4 once we have verified that Assumption 2.1 is fulfilled. Let $z \in \mathcal{S}$ and let $R(x) = f(x) - z - f'(z)(x - z)$ be the remainder term of the first-order Taylor expansion of f around z. We need to verify that for a given $\varepsilon \in (0, \infty)$ we can take $r_\varepsilon \in (0, \infty)$ so small that the restriction of R to $\mathbb{B}_{r_\varepsilon}(z)$ is ε-Lipschitz continuous. Take $r \in (0, \infty)$ so that $\mathbb{B}_r(z) \subseteq U$. For all $x \in \mathbb{B}_r(z)$, note that

$$R(x) = \int_0^1 \big[f'(z + s(x - z)) - f'(z)\big](x - z)\, ds.$$

Denote the Lipschitz constant of f' on $\mathbb{B}_r(z)$ by $L \in [0, \infty)$ Then, for all $x, y \in \mathbb{B}_r(z)$,

$$\begin{aligned}
&\|R(x) - R(y)\| \\
&= \left\| \int_0^1 \big[f'(z + s(x - z)) - f'(z)\big](x - y) + \big[f'(z + s(x - z)) - f'(z + s(y - z))\big](y - z)\, ds \right\| \\
&\leq \int_0^1 Ls \|x - z\| \|x - y\| + Ls \|x - y\| \|y - z\|\, ds \leq Lr \|x - y\|.
\end{aligned}$$

So, given $\varepsilon \in (0, \infty)$, we pick $r_\varepsilon = \min\{r, \varepsilon L^{-1}\}$. ∎

3. Gradient descent for shallow ReLU networks

We now turn to studying shallow ReLU networks. Throughout this section, suppose $d = 3N + 1$ for an $N \in \mathbb{N}$ and fix $\alpha, \beta \in \mathbb{R}$ with $\alpha < \beta$. Then, $\mathbb{R}^d$ represents the space of all shallow networks with N hidden neurons. We will always write a network $\theta \in \mathbb{R}^{3N+1}$ as $\theta = (w, b, v, c)$, where $w, b, v \in \mathbb{R}^N$ and $c \in \mathbb{R}$. The realization of a network θ is the function $\mathcal{R}_\theta \in C(\mathbb{R}, \mathbb{R})$ given by

$$\mathcal{R}_\theta(x) = c + \sum_{j=1}^{N} v_j \max\{w_j x + b_j, 0\}.$$

Fix $\mathfrak{f} \in C([\alpha, \beta], \mathbb{R})$. We denote by $\mathcal{L} \in C(\mathbb{R}^d, \mathbb{R})$ the squared L^2-loss with target function $\mathfrak{f}$, that is

$$\mathcal{L}(\theta) = \int_\alpha^\beta (\mathcal{R}_\theta(x) - \mathfrak{f}(x))^2 \, dx.$$

To discuss regularity properties of the loss function, it is convenient to recall the following definition, which we introduced in Chapter 4 (see Definition 2.3 therein). Motivation and discussion of these notions can be found there.

Definition 3.1. Let $\theta = (w, b, v, c) \in \mathbb{R}^{3N+1}$ and $j \in \{1, \dots, N\}$. Then, we denote by I_j the set given by $I_j = \{x \in [\alpha, \beta] \colon w_j x + b_j \geq 0\}$, we say that the j^{th} hidden neuron of θ is

- *flat* if $v_j = 0$,
- *non-flat* if $v_j \neq 0$,
- *inactive* if $I_j = \emptyset$,
- *semi-inactive* if $\# I_j = 1$,
- *semi-active* if $w_j = 0 < b_j$,
- *active* if $w_j \neq 0 < b_j + \max\{w_j \alpha, w_j \beta\}$,
- *type-1-active* if $w_j \neq 0 \leq b_j + \min\{w_j \alpha, w_j \beta\}$,
- *type-2-active* if $\emptyset \neq I_j \cap (\alpha, \beta) \neq (\alpha, \beta)$,
- *degenerate* if $|w_j| + |b_j| = 0$,
- *non-degenerate* if $|w_j| + |b_j| > 0$,

and we say that $t \in \mathbb{R}$ is the breakpoint of the j^{th} hidden neuron of θ if $w_j \neq 0 = w_j t + b_j$.

We showed in Chapter 4 that $\mathcal{L}$ is differentiable at all coordinates corresponding to non-degenerate or flat degenerate neurons. In general, the loss fails to be differentiable at non-flat degenerate neurons. To apply the dynamical systems theory, we need a function defined on the whole $\mathbb{R}^d$. Thus, we need to work with a generalized gradient of $\mathcal{L}$. There are many different choices for such a generalized gradient. Here, we actually do not specify a choice, but only require that our generalized gradient agrees with partial derivatives of $\mathcal{L}$ coordinate-wise. So, throughout this section, let $\mathcal{G} \colon \mathbb{R}^d \to \mathbb{R}^d$ satisfy for all $\theta \in \mathbb{R}^d$ and all $j \in \{1, \dots, N\}$ such that the j^{th} neuron of θ is non-degenerate or flat degenerate that

$$\mathcal{G}_j(\theta) = \frac{\partial}{\partial w_j}\mathcal{L}(\theta), \quad \mathcal{G}_{N+j}(\theta) = \frac{\partial}{\partial b_j}\mathcal{L}(\theta), \quad \mathcal{G}_{2N+j}(\theta) = \frac{\partial}{\partial v_j}\mathcal{L}(\theta), \quad \mathcal{G}_{3N+1}(\theta) = \frac{\partial}{\partial c}\mathcal{L}(\theta).$$

The map $\mathcal{G}$ may take any values at coordinates of non-flat degenerate neurons. The dynamical system we are interested in is the gradient descent step $f_\gamma(\theta) = \theta - \gamma \mathcal{G}(\theta)$ for some given step size $\gamma \in (0, \infty)$.

One crucial aspect of Theorem 2.2 is that we do not need the dynamical system to be a local diffeomorphism, let alone differentiable everywhere. However, the dynamical system ought to be differentiable at the saddle points of $\mathcal{L}$ we are interested in. Where

$\mathcal{G} = \nabla\mathcal{L}$, differentiability of f_γ means two times differentiability of $\mathcal{L}$. Even though $\mathcal{L}$ is twice differentiable on a set of full measure (see [66]), some of the saddle points lie outside of that full-measure set. More precisely, it is semi-inactive neurons that cause the regularity problems. The resulting nonexistence of the Hessian of $\mathcal{L}$ urges us to work with suitably modified dynamical systems. The idea is to replace entries of $\mathcal{G}$ that correspond to semi-inactive neurons of a given saddle point but to keep the remaining entries as they are. For technical reasons, a prescribed set $J \subseteq \{1, \dots, N\}$ of semi-inactive neurons is split into two subsets J_+ and J_-, each containing those semi-inactive neurons with $w_j > 0$ and $w_j < 0$, respectively. The exact formula for the modified gradient $\mathcal{G}^J$ is given below. The new dynamical system $f_{\gamma,J}(\theta) = \theta - \gamma\mathcal{G}^J(\theta)$ no longer coincides with the original gradient descent f_γ, but we will be able to recover information about the dynamics of f_γ from $f_{\gamma,J}$; see Lemma 3.2 below. Now, let $\mathcal{J}$ be the set

$$\mathcal{J} = \{(J_+, J_-) \colon J_+, J_- \subseteq \{1, \dots, N\} \text{ such that } J_+ \cap J_- = \emptyset\}.$$

For any $J = (J_+, J_-) \in \mathcal{J}$, let $\mathcal{G}^J \colon \mathbb{R}^d \to \mathbb{R}^d$ satisfy for all $\theta \in \mathbb{R}^d$ and $j \in \{1, \dots, N\}$ that $\mathcal{G}^J_{3N+1}(\theta) = \mathcal{G}_{3N+1}(\theta)$ and

$$\begin{aligned}
&\left(\mathcal{G}^J_j, \mathcal{G}^J_{N+j}, \mathcal{G}^J_{2N+j}\right)(\theta) \\
&= \begin{cases}
\left(\mathcal{G}_j, \mathcal{G}_{N+j}, \mathcal{G}_{2N+j}\right)(\theta) & \text{if } j \notin J_+ \cup J_- \text{ or } w_j = 0, \\
2\int_{t_j}^{\beta} (v_j x, v_j, w_j x + b_j)(\mathcal{R}_\theta(x) - \mathfrak{f}(x))\, dx & \text{if } j \in J_+ \text{ and } w_j \neq 0, \\
2\int_{\alpha}^{t_j} (v_j x, v_j, w_j x + b_j)(\mathcal{R}_\theta(x) - \mathfrak{f}(x))\, dx & \text{if } j \in J_- \text{ and } w_j \neq 0.
\end{cases}
\end{aligned}$$

Note that $\mathcal{G}^{(\emptyset,\emptyset)} = \mathcal{G}$. If a neuron $j \in J_+ \cup J_-$ is semi-inactive or type-2-active with the sign of w_j matching the sign in the subscript of $J_\pm$, then $\mathcal{G}^J$ agrees with $\mathcal{G}$ in the coordinates of the j^{th} neuron. Thus, we did not actually change $\mathcal{G}$ at semi-inactive neurons with matching signs, but we changed $\mathcal{G}$ at inactive neurons in a way that $\mathcal{G}^J$ becomes differentiable at semi-inactive neurons (which are neighbored by inactive neurons). In the next lemma, we leverage that the original dynamical system f_γ does not alter coordinates of inactive neurons to show how to infer dynamical information about f_γ from its modifications.

Lemma 3.2. *Let $\mathcal{S} \subseteq \mathbb{R}^d$ and, for all $J = (J_+, J_-) \in \mathcal{J}$, let $\mathcal{S}_J \subseteq \mathcal{S}$ contain all networks $\theta \in \mathcal{S}$ such that J_+ is exactly the set of neurons of θ that are semi-inactive with $w_j > 0$ and J_- is exactly the set of neurons of θ that are semi-inactive with $w_j < 0$. Then, $\mathcal{S} = \bigcup_{J\in\mathcal{J}} \mathcal{S}_J$ and*

$$\left\{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f^k_\gamma(\theta) \in \mathcal{S}\right\} \subseteq \bigcup_{J\in\mathcal{J}} \bigcup_{n\in\mathbb{N}_0} f^{-n}_\gamma\left(\left\{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f^k_{\gamma,J}(\theta) \in \mathcal{S}_J\right\}\right).$$

Proof. That $\mathcal{S} = \bigcup_{J\in\mathcal{J}} \mathcal{S}_J$ is clear. Suppose $\theta_0 \in \{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f^k_\gamma(\theta) \in \mathcal{S}\}$ and let $\theta_\infty \in \mathcal{S}$ be the limit point of $f^k_\gamma(\theta_0)$ as $k \to \infty$. Take $J \in \mathcal{J}$ with $\theta_\infty \in \mathcal{S}_J$ and abbreviate $\theta_k = f^k_\gamma(\theta_0)$. Note that f_γ does not change coordinates of inactive neurons. More precisely, for all $j \in \{1, \dots, N\}$ and $k, n \in \mathbb{N}_0$, if the j^{th} neuron of θ_n is inactive, then θ_n and θ_{n+k} agree in the (w_j, b_j, v_j)-coordinates. Furthermore, any sufficiently small neighborhood of a semi-inactive neuron contains only inactive, semi-inactive, and type-2-active neurons. It follows from this that there exists an $n \in \mathbb{N}_0$ such that for all $k \in \mathbb{N}_0$ and all $j \in J_+ \cup J_-$ the j^{th} neuron of θ_{n+k} is type-2-active or semi-inactive with $\operatorname{sgn}(w_j)$ matching the subscript of $J_\pm$ in both cases. Then, $\mathcal{G}^J(\theta_{n+k}) = \mathcal{G}(\theta_{n+k})$ for all $k \in \mathbb{N}_0$. In particular, $\theta_{n+k} = f^k_{\gamma,J}(\theta_n)$ for all $k \in \mathbb{N}_0$ and, hence, $\theta_n \in \{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f^k_{\gamma,J}(\theta) \in \mathcal{S}_J\}$. ∎

Subsequently, we need to accomplish two objectives. First, we need to verify that the dynamical systems theory (Proposition 2.5) is applicable to each $f_{\gamma,J}$ to deduce that the sets $\{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f_{\gamma,J}^k(\theta) \in \mathcal{S}_J\}$ have zero Lebesgue measure. Secondly, we need to apply Lemma 2.4 to f_γ to conclude with the previous lemma that $\{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f_\gamma^k(\theta) \in \mathcal{S}\}$ also has zero Lebesgue measure.

3.1 Non-degeneracy almost everywhere

In this section, we show that there exists an open subset of $\mathbb{R}^d$ of full measure such that the modified dynamical system exhibits the regularity required by Proposition 2.5 on that subset. For any $J = (J_+, J_-) \in \mathcal{J}$, let $U_0^J \subseteq \mathbb{R}^d$ be the set of all networks without degenerate neurons such that $w_j \neq 0$ for all $j \in J_+ \cup J_-$; let $U_1^J \subseteq \mathbb{R}^d$ be the set of all networks without degenerate neurons such that no neuron in $\{1, \dots, N\} \backslash (J_+ \cup J_-)$ is semi-inactive or type-1-active with breakpoint α or β; and let $U^J = U_0^J \cap U_1^J$. Let $U_\infty \subseteq U^{\emptyset,\emptyset} \cap U^{\{1,\dots,N\},\emptyset}$ be the set of all networks that do not have two distinct type-2-active neurons with the same breakpoint. We remark that $U_\infty \subseteq \mathbb{R}^d$ is open and has full measure.

Lemma 3.3. *Let $J = (J_+, J_-) \in \mathcal{J}$. Then, the following properties hold.*

(i) *$\mathcal{G}^J$ is continuously differentiable on U^J.*

(ii) *The Jacobian $(\mathcal{G}^J)'(\theta)$ is a symmetric matrix for all $\theta \in U^J$ for which for all $\tau \in \{+, -\}$ and $j \in J_\tau$ the j^{th} neuron of θ is semi-inactive with $\operatorname{sign}(w_j) = \tau$.*

(iii) *If $\mathfrak{f}$ is Lipschitz continuous, then the Jacobian of $\mathcal{G}^J$ is locally Lipschitz continuous on U^J.*

(iv) *If $\mathfrak{f}$ is a polynomial, then $\mathcal{G}^J$ is a rational function on U_∞.*

Proof. The set $U_0^{(\emptyset,\emptyset)}$ is the set of all networks without degenerate neurons. For all $j \in \{1, \dots, N\}$, we let $r_j, s_j \colon U_0^{(\emptyset,\emptyset)} \to \mathbb{R}$ be the functions given by

$$r_j(\theta) = \begin{cases} \frac{\alpha+\beta}{2} - \frac{w_j(\beta-\alpha)^2}{2w_j(\alpha+\beta)+4b_j} & \text{if the } j^{th} \text{ neuron of } \theta \text{ is inactive,} \\ \beta & \text{if the } j^{th} \text{ neuron of } \theta \text{ is semi-inactive with } w_j > 0, \\ t_j & \text{if the } j^{th} \text{ neuron of } \theta \text{ is type-2-active with } w_j > 0, \\ \alpha & \text{otherwise} \end{cases}$$

and

$$s_j(\theta) = \begin{cases} r_j(\theta) & \text{if the } j^{th} \text{ neuron of } \theta \text{ is inactive,} \\ \alpha & \text{if the } j^{th} \text{ neuron of } \theta \text{ is semi-inactive with } w_j < 0, \\ t_j & \text{if the } j^{th} \text{ neuron of } \theta \text{ is type-2-active with } w_j < 0, \\ \beta & \text{otherwise.} \end{cases}$$

$r_j(\theta)$ and $s_j(\theta)$ are the endpoints of the interval I_j if the j^{th} neuron of θ is not inactive and $[r_j, s_j]$ is a singleton if it is inactive. Observe that r_j and s_j are locally Lipschitz continuous and, for any connected component V of $U_1^{(\{j\}^c,\emptyset)}$, the restrictions $r_j|_V$ and $s_j|_V$ are rational functions. In particular, r_j and s_j are infinitely often differentiable on $U_1^{(\{j\}^c,\emptyset)}$. Next, we define similar functions $r_j^J, s_j^J \colon U_0^J \to \mathbb{R}$ by

$$r_j^J(\theta) = \begin{cases} r_j(\theta) & \text{if } j \notin J_+ \cup J_-, \\ t_j & \text{if } j \in J_+, \\ \alpha & \text{if } j \in J_-, \end{cases} \qquad s_j^J(\theta) = \begin{cases} s_j(\theta) & \text{if } j \notin J_+ \cup J_-, \\ \beta & \text{if } j \in J_+, \\ t_j & \text{if } j \in J_-. \end{cases}$$

These functions are locally Lipschitz continuous on U_0^J and infinitely often differentiable on U^J because if $j \notin J_+ \cup J_-$, then $U^J \subseteq U_1^{(\{j\}^c,\emptyset)}$. Now, for all $\theta \in U_0^J$, $j \in \{1,\dots,N\}$, $i \in \{0,1,2\}$,

$$\mathcal{G}^J_{iN+j}(\theta) = 2\int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)(\mathcal{R}_\theta(x) - \mathfrak{f}(x))\,dx,$$
$$\mathcal{G}^J_{3N+1}(\theta) = 2\int_{\alpha}^{\beta} (\mathcal{R}_\theta(x) - \mathfrak{f}(x))\,dx.$$

Thus, all partial derivatives of $\mathcal{G}^J$ exist on U^J by the Leibniz integral rule and are given by

$$\begin{aligned}\frac{\partial}{\partial\theta_{i'N+j'}}\mathcal{G}^J_{iN+j}(\theta) &= 2\int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)\Big(\frac{\partial}{\partial\theta_{i'N+j'}} v_{j'}(w_{j'} x + b_{j'})\Big)\mathbb{1}_{[r_{j'},s_{j'}]}(x)\,dx\\ &\quad + 2\int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{i'N+j'}}\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)(\mathcal{R}_\theta(x) - \mathfrak{f}(x))\,dx\\ &\quad + 2\Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)(\mathcal{R}_\theta(x) - \mathfrak{f}(x))\Big|_{x=s_j^J}\Big(\frac{\partial}{\partial\theta_{i'N+j'}} s_j^J\Big)\\ &\quad - 2\Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)(\mathcal{R}_\theta(x) - \mathfrak{f}(x))\Big|_{x=r_j^J}\Big(\frac{\partial}{\partial\theta_{i'N+j'}} r_j^J\Big)\end{aligned}$$

and

$$\begin{aligned}\frac{\partial}{\partial\theta_{3N+1}}\mathcal{G}^J_{iN+j}(\theta) &= 2\int_{r_j^J}^{s_j^J} \frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\,dx,\\ \frac{\partial}{\partial\theta_{iN+j}}\mathcal{G}^J_{3N+1}(\theta) &= 2\int_{r_j}^{s_j} \frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\,dx,\\ \frac{\partial}{\partial\theta_{3N+1}}\mathcal{G}^J_{3N+1}(\theta) &= 2.\end{aligned}$$

In particular, all partial derivatives of $\mathcal{G}^J$ are continuous and, hence, $\mathcal{G}^J$ is continuously differentiable on U^J. This proves (i). Moreover, since r_j, s_j, r_j^J, s_j^J, and $\theta \mapsto \mathcal{R}_\theta$ are locally Lipschitz continuous on U_0^J, it follows from the above formulas that if $\mathfrak{f}$ is Lipschitz, then $(\mathcal{G}^J)'$ is locally Lipschitz on U^J. Next, note the following equality, for all $j, j' \in \{1,\dots,N\}$, $i, i' \in \{0,1,2\}$, $\theta \in U_0^{(\{j,j'\},\emptyset)}$;

$$\Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)\Big|_{x=t_j}\frac{\partial}{\partial\theta_{i'N+j'}} t_j = \Big(\frac{\partial}{\partial\theta_{i'N+j'}} v_{j'}(w_{j'} x + b_{j'})\Big)\Big|_{x=t_{j'}}\frac{\partial}{\partial\theta_{iN+j}} t_{j'}.$$

Therefore, if $\theta \in U^J$ satisfies $r_j^J = r_j$ and $s_j^J = s_j$ for all $j \in \{1,\dots,N\}$, then $(\mathcal{G}^J)'(\theta)$ is symmetric. In particular, this holds for all $\theta \in U^J$ satisfying the conditions of (ii).

Now, suppose $\mathfrak{f}$ is a polynomial. For any $\theta \in U_0^J$, $j \in \{1,\dots,N\}$, $i \in \{0,1,2\}$, we can write

$$\begin{aligned}\mathcal{G}^J_{iN+j}(\theta) &= 2\int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)(c - \mathfrak{f}(x))\,dx\\ &\quad + 2\sum_{n=1}^{N}\int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big) v_n(w_n x + b_n)\mathbb{1}_{[r_n,s_n]}(x)\,dx.\end{aligned}$$

The functions $P_{i,j}\colon \mathbb{R}^d \times [\alpha,\beta] \to \mathbb{R}$ given by

$$\begin{aligned} P_{i,j}(\theta,x) &= \int_0^x \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j y + b_j)\Big)(c - \mathfrak{f}(y))\,dy \\ &= \frac{\partial}{\partial\theta_{iN+j}} \left(\frac{1}{2} v_j w_j c x^2 + v_j b_j c x - v_j(w_j x + b_j)\int_0^x \mathfrak{f}(y)\,dy + v_j w_j \int_0^x \int_0^y \mathfrak{f}(z)\,dz dy\right) \end{aligned}$$

are polynomials. By definition of these functions, for any $\theta \in U_0^J$,

$$\int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)(c - \mathfrak{f}(x))\,dx = P_{i,j}(\theta, s_j^J) - P_{i,j}(\theta, r_j^J).$$

For any $j \in \{1,\dots,N\}$ and any connected component of U_∞, the functions r_j^J and s_j^J equal each other, are constant, or are equal to $t_j = -b_j/w_j$ throughout that entire component. It follows that we can take $q \in \mathbb{N}$ sufficiently large so that

$$\theta \mapsto \left(\prod_{k=1}^N w_k^q\right) \int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big)(c - \mathfrak{f}(x))\,dx = \left(\prod_{k=1}^N w_k^q\right)\big(P_{i,j}(\theta, s_j^J) - P_{i,j}(\theta, r_j^J)\big)$$

is a polynomial on U_∞. The remainder of the proof is similar to the previous step. The functions $P_{i,j,n}\colon \mathbb{R}^d \times [\alpha,\beta] \to \mathbb{R}$ given by

$$\begin{aligned} P_{i,j,n}(\theta,x) &= \int_0^x \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j y + b_j)\Big) v_n(w_n y + b_n)\,dy \\ &= \frac{1}{6}\Big(\frac{\partial}{\partial\theta_{iN+j}} v_j w_j\Big)(2 v_n w_n x^3 + 3 v_n b_n x^2) + \frac{1}{2}\Big(\frac{\partial}{\partial\theta_{iN+j}} v_j b_j\Big)(v_n w_n x^2 + 2 v_n b_n x) \end{aligned}$$

are polynomials. By definition of U_∞, given $j,n \in \{1,\dots,N\}$, if two of the functions r_j^J, s_j^J, r_n, and s_n agree at some network $\theta \in U_\infty$, then they agree on the entire component of U_∞ containing θ or the n^{th} neuron is inactive for all networks in that component. Thus, given any connected component of U_∞, one of the following eight cases holds throughout the entire component:

$$\int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big) v_n(w_n x + b_n)\mathbb{1}_{[r_n,s_n]}(x)\,dx = \begin{cases} P_{i,j,n}(\theta, s_j^J) - P_{i,j,n}(\theta, s_n), \\ P_{i,j,n}(\theta, s_j^J) - P_{i,j,n}(\theta, r_n), \\ P_{i,j,n}(\theta, s_j^J) - P_{i,j,n}(\theta, r_j^J), \\ P_{i,j,n}(\theta, s_n) - P_{i,j,n}(\theta, r_j^J), \\ P_{i,j,n}(\theta, s_n) - P_{i,j,n}(\theta, r_n), \\ P_{i,j,n}(\theta, r_n) - P_{i,j,n}(\theta, s_n), \\ P_{i,j,n}(\theta, r_n) - P_{i,j,n}(\theta, r_j^J), \\ 0. \end{cases}$$

This implies that

$$\theta \mapsto \left(\prod_{k=1}^N w_k^q\right) \int_{r_j^J}^{s_j^J} \Big(\frac{\partial}{\partial\theta_{iN+j}} v_j(w_j x + b_j)\Big) v_n(w_n x + b_n)\mathbb{1}_{[r_n,s_n]}(x)\,dx$$

is a polynomial on U_∞ for a sufficiently large $q \in \mathbb{N}$. We conclude that also the map $\theta \mapsto \mathcal{G}_{iN+j}^J(\theta)\prod_{k=1}^N w_k^q$ is a polynomial on U_∞ for a sufficiently large $q \in \mathbb{N}$. The same argument in a simplified version works for $\mathcal{G}_{3N+1}^J(\theta) = 2\int_\alpha^\beta (\mathcal{R}_\theta(x) - \mathfrak{f}(x))\,dx$. ∎

The last bit of regularity we need to check is the non-degeneracy of the Jacobian $f'_{\gamma,J}$. Lemma 3.3.(iv) enables this.

Lemma 3.4. *If $\mathfrak{f}$ is a polynomial, then for almost all $\gamma \in (0,\infty)$ and for all $J \in \mathcal{J}$ there exists an open set $U_\gamma^J \subseteq U_\infty$ of full measure such that $\det(f'_{\gamma,J}(\theta)) \neq 0$ for all $\theta \in U_\gamma^J$.*

Proof. Fix $J \in \mathcal{J}$. Since $\mathcal{G}^J$ is a rational function on U_∞, there exists a polynomial $p\colon U_\infty \to \mathbb{R}$, which is not constantly zero on any connected component of U_∞, such that $\theta \mapsto p(\theta)\mathcal{G}^J(\theta)$ is a polynomial on U_∞. Since the derivative of a polynomial is still a polynomial, it follows that

$$\theta \mapsto p(\theta)^2(\mathcal{G}^J)'(\theta) = (pp\mathcal{G}^J)'(\theta) - 2p(\theta)p'(\theta)\mathcal{G}^J(\theta)$$

is also a polynomial on U_∞. The differential of $f_{\gamma,J}$ on U_∞ is $f'_{\gamma,J}(\theta) = I - \gamma(\mathcal{G}^J)'(\theta)$. Therefore, the map $P\colon U_\infty \times \mathbb{R} \to \mathbb{R}$ given by $P(\theta,\gamma) = \det(p(\theta)^2 f'_{\gamma,J}(\theta))$ is a polynomial. Moreover, P is not constantly zero on any connected component of $U_\infty \times \mathbb{R}$ because $P(\theta,0) = p(\theta)^2$. In particular, its zero set $P^{-1}(0)$ has Lebesgue measure zero. For every $\gamma \in \mathbb{R}$, denote $Z_\gamma = \{\theta \in U_\infty\colon P(\theta,\gamma) = 0\}$. By Tonelli's theorem,

$$0 = \int_{P^{-1}(0)} d\theta d\gamma = \int_{\mathbb{R}} \int_{Z_\gamma} d\theta d\gamma,$$

from which it follows that Z_γ has zero Lebesgue measure for almost every $\gamma \in \mathbb{R}$. Set $U_\gamma^J = U_\infty \backslash Z_\gamma$. ∎

This concludes the discussion of the regularity requirements. It remains to establish strictness of saddle points of $\mathcal{L}$.

3.2 Strict saddle points

To investigate saddle points of $\mathcal{L}$, it is useful to classify them in terms of their types of neurons. The next result follows from Theorem 2.4 and Corollary 2.7 in Chapter 4.

Proposition 3.5. *Assume $\mathfrak{f}$ is affine but not constant and let $\theta = (w,b,v,c) \in U_0^{(\emptyset,\emptyset)}$ be a critical point of $\mathcal{L}$ that is not a global minimum. Then, the following hold:*

(i) *θ is not a local maximum of $\mathcal{L}$.*

(ii) *θ is a local minimum of $\mathcal{L}$ if and only if $c = \mathfrak{f}(\frac{\alpha+\beta}{2})$ and, for all $j \in \{1,\dots,N\}$, the j^{th} hidden neuron of θ is inactive or semi-inactive with $\mathfrak{f}'(\alpha)v_j w_j < 0$.*

(iii) *θ is a saddle point of $\mathcal{L}$ if and only if $c = \mathfrak{f}(\frac{\alpha+\beta}{2})$, θ does not have any type-1-active or non-flat semi-active neurons, and exactly one of the following two conditions holds:*

  (a) *θ does not have any type-2-active neurons and there exists $j \in \{1,\dots,N\}$ such that the j^{th} hidden neuron of θ is flat semi-active or semi-inactive with $\mathfrak{f}'(\alpha)v_j w_j \geq 0$.*

  (b) *There exists $n \in \{2,4,6,\dots\}$ such that*

$$\bigcup_{j\in\{1,\dots,N\},\, w_j\neq 0} \left\{-\frac{b_j}{w_j}\right\} \cap (\alpha,\beta) = \bigcup_{i=1}^{n} \left\{\alpha + \frac{i(\beta-\alpha)}{n+1}\right\}$$

and, for all $j \in \{1,\dots,N\}$, $i \in \{1,\dots,n\}$ *with* $w_j \neq 0 = b_j + w_j(\alpha + \frac{i(\beta-\alpha)}{n+1})$, *it holds that* $\operatorname{sign}(w_j) = (-1)^{i+1}$ *and*

$$\sum_{k\in\{1,\dots,N\},\, w_k\neq 0 = b_k + w_k(\alpha+\frac{i(\beta-\alpha)}{n+1})} v_k w_k = \frac{2\mathfrak{f}'(\alpha)}{n+1}.$$

(iv) *There exists* $n \in \{0,2,4,\dots\}$ *with* $n \leq N$ *such that* $\mathcal{L}(\theta) = \dfrac{[\mathfrak{f}'(\alpha)]^2(\beta-\alpha)^3}{12(n+1)^4}$ *and*

$$\mathcal{R}_\theta(x) = \mathfrak{f}(x) - \frac{(-1)^i \mathfrak{f}'(\alpha)}{n+1}\Big(x - \alpha - \frac{(i+\frac{1}{2})(\beta-\alpha)}{n+1}\Big)$$

for all $i \in \{0,\dots,n\}$, $x \in [\alpha + \frac{i(\beta-\alpha)}{n+1}, \alpha + \frac{(i+1)(\beta-\alpha)}{n+1}]$.

Now, we can clarify for which saddle points of $\mathcal{L}$ we can establish strictness. For an affine target function $\mathfrak{f}$, let $\mathcal{S} \subseteq U_0^{(\emptyset,\emptyset)}$ be the set of all saddle points of $\mathcal{L}$ that are not solely comprised of inactive neurons and semi-inactive neurons with $\mathfrak{f}'(\alpha)v_j w_j \leq 0$. As in Lemma 3.2, for all $J \in \mathcal{J}$, let $\mathcal{S}_J \subseteq \mathcal{S}$ be the set of networks θ such that J_+ is exactly the set of neurons of θ that are semi-inactive with $w_j > 0$ and J_- is exactly the set of neurons of θ that are semi-inactive with $w_j < 0$.

Recall that $f_{\gamma,J}(\theta) = \theta - \gamma\mathcal{G}^J(\theta)$. Thus, to show that $f'_{\gamma,J}(\theta) = I - \gamma(\mathcal{G}^J)'(\theta)$ has an eigenvalue of absolute value strictly greater than 1, it is sufficient to show that $(\mathcal{G}^J)'(\theta)$ has a strictly negative eigenvalue.

Lemma 3.6. *Assume* $\mathfrak{f}$ *is affine but not constant, and let* $J \in \mathcal{J}$, $\theta \in \mathcal{S}_J$. *Then,* $\mathcal{G}^J(\theta) = 0$ *and the matrix* $(\mathcal{G}^J)'(\theta)$ *has a strictly negative eigenvalue.*

Proof. On the one hand, for all $j \notin J_+ \cup J_-$ and $i \in \{0,1,2\}$, we know that $\mathcal{G}^J_{iN+j}(\theta) = \mathcal{G}_{iN+j}(\theta) = 0$. On the other hand, for all $j \in J_+$, we have that $t_j = \beta$ and, hence, also $\mathcal{G}^J_{iN+j}(\theta) = 0$ for all $i \in \{0,1,2\}$; likewise for $j \in J_-$. This shows that $\mathcal{G}^J(\theta) = 0$.

Proposition 3.5 tells us that θ has no type-1-active neurons, so $\theta \in U^J$ and $\mathcal{G}^J$ is differentiable at θ with symmetric Jacobian $(\mathcal{G}^J)'(\theta)$ by Lemma 3.3. We will conclude the proof by showing that $(\mathcal{G}^J)'(\theta)$ contains a strictly negative principle minor. To this end, we distinguish two cases. First, if $\mathcal{R}_\theta$ is affine on $[\alpha,\beta]$, then θ must have a flat semi-active neuron or a semi-inactive neuron with $\mathfrak{f}'(\alpha)v_j w_j > 0$, by Proposition 3.5 and by the definition of the set $\mathcal{S}$. If θ has a flat semi-active neuron j, then

$$\det\begin{pmatrix} \frac{\partial}{\partial\theta_j}\mathcal{G}^J_j(\theta) & \frac{\partial}{\partial\theta_j}\mathcal{G}^J_{2N+j}(\theta) \\ \frac{\partial}{\partial\theta_{2N+j}}\mathcal{G}^J_j(\theta) & \frac{\partial}{\partial\theta_{2N+j}}\mathcal{G}^J_{2N+j}(\theta)\end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{6}\mathfrak{f}'(\alpha)(\beta-\alpha)^3 \\ -\frac{1}{6}\mathfrak{f}'(\alpha)(\beta-\alpha)^3 & 2b_j^2(\beta-\alpha)\end{pmatrix}$$
$$= -\frac{1}{36}[\mathfrak{f}'(\alpha)]^2(\beta-\alpha)^6 < 0.$$

If θ has a semi-inactive neuron j with $\mathfrak{f}'(\alpha)v_j w_j > 0$, then

$$\frac{\partial}{\partial\theta_{N+j}}\mathcal{G}^J_{N+j}(\theta) = -\mathfrak{f}'(\alpha)\frac{v_j}{w_j}(\beta-\alpha) < 0.$$

Secondly, if $\mathcal{R}_\theta$ is not affine on $[\alpha, \beta]$, then exactly as in the proof[3] of Lemma 2.24 in Chapter 4 we can find a set of coordinates corresponding to type-2-active neurons such that the determinant of the Hessian H of $\mathcal{L}$ restricted to these coordinates is strictly negative. Since this involves only neurons in $\{1, \dots, N\} \backslash (J_+ \cup J_-)$, the matrix $(\mathcal{G}^J)'(\theta)$ contains H as a submatrix. ■

Having established strictness of saddle points, it is now straight-forward to apply Proposition 2.5, which yields the following result.

Theorem 3.7. *Assume* $\mathfrak{f}$ *is affine but not constant. Then, for almost every step size* $\gamma \in (0, \infty)$*, the set* $\{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f_\gamma^k(\theta) \in \mathcal{S}\}$ *has Lebesgue measure zero.*

Proof. By Lemmas 3.3, 3.4, and 3.6, we can apply Proposition 2.5 for almost all $\gamma \in (0, \infty)$ and all $J \in \mathcal{J}$ to the dynamical system $f_{\gamma,J}$ and the set $\mathcal{S}_J$ with $U = U^J$ and $V = U_\gamma^J$ to find that $\{\theta \in \mathbb{R}^d \colon \lim_{k\to\infty} f_{\gamma,J}^k(\theta) \in \mathcal{S}_J\}$ has Lebesgue measure zero. Since $\mathcal{G} = \mathcal{G}^{(\emptyset,\emptyset)}$, Lemmas 3.3 and 3.4 enable us to apply Lemma 2.4 to f_γ so that, together with Lemma 3.2, we obtain the desired result. ■

3.3 Convergence to global minima for suitable initialization

Suppose a trajectory of gradient descent for the loss function $\mathcal{L}$ with affine nonconstant target function $\mathfrak{f}$ converges to a critical point of $\mathcal{L}$. If the gradient descent algorithm was initialized randomly under a probability measure that is absolutely continuous with respect to the Lebesgue measure, then, with probability one, the limit critical point is not a saddle point in $\mathcal{S}$ by Theorem 3.7. Here, $\mathcal{S}$ is the same set of saddle points as specified above Lemma 3.6. We can say more about the limit critical point using Proposition 3.5.(iv). It states that there are only finitely many possibilities for the value of the loss function at its critical points, which we can think of as partitioning the set of all critical points into "layers". In particular, if the loss at the limit critical point is below the threshold $\frac{[\mathfrak{f}'(\alpha)]^2(\beta-\alpha)^3}{12(N+1)^4}$, then this critical point must belong to the first layer, that is it must be a global minimum. In the following, we improve this threshold to the next layer of critical points.

Proposition 3.8. *Assume* $\mathfrak{f}$ *is affine but not constant and that* N *is even. For almost all* $\gamma \in (0, \infty)$ *and almost all*

$$\theta \in \left\{ \vartheta \in \mathbb{R}^d \colon (f_\gamma^k(\vartheta))_{k\in\mathbb{N}_0} \text{ is convergent and } \lim_{k\to\infty} \mathcal{L}\left(f_\gamma^k(\vartheta)\right) < \tfrac{[\mathfrak{f}'(\alpha)]^2(\beta-\alpha)^3}{12(N-1)^4} \right\}, \tag{3.1}$$

it holds that $\lim_{k\to\infty} \mathcal{L}(f_\gamma^k(\theta)) = 0$.

Proof. Let θ_0 be in (3.1) and let $\theta = \lim_{k\to\infty} f_\gamma^k(\theta_0)$. By definition of f_γ, we have

$$\lim_{k\to\infty} \left\| \mathcal{G}(f_\gamma^k(\theta_0)) \right\| = \lim_{k\to\infty} \frac{1}{\gamma} \left\| f_\gamma^k(\theta_0) - f_\gamma^{k+1}(\theta_0) \right\| = 0. \tag{3.2}$$

[3] While Lemma 2.24 in Chapter 4 considers the special case $\alpha = 0$, $\beta = 1$, and $\mathfrak{f}(x) = x$, the arguments work exactly the same way in the general case. In the proof of Lemma 2.24 in Chapter 4, only the following modifications need to be made: the sign condition on v_j in K_1^- becomes $\operatorname{sgn}(v_j) = -\operatorname{sgn}(\mathfrak{f}'(\alpha))$ instead of $v_j < 0$; the sign condition on v_{j_1} becomes $\operatorname{sgn}(v_{j_1}) = \operatorname{sgn}(\mathfrak{f}'(\alpha))$ instead of $v_{j_1} > 0$; and the constants μ and λ_i become $\mu = \frac{n+1}{2n(\beta-\alpha)}$ and $\lambda_i = \frac{n+1}{2\mathfrak{f}'(\alpha)} v_{j_i} w_{j_i}$, respectively.

Let $m \in \{0, \dots, N\}$ be the number of degenerate neurons of θ. Note that $\mathcal{R}_\theta$ cannot be constant on $[\alpha, \beta]$ since

$$\mathcal{L}(\theta) < \frac{1}{12}[\mathfrak{f}'(\alpha)]^2(\beta-\alpha)^3 = \inf_{C\in\mathbb{R}} \int_\alpha^\beta (C - \mathfrak{f}(x))^2 \, dx.$$

Hence, it cannot be that θ has N degenerate neurons. In other words, $m \leq N-1$. Let $\vartheta \in \mathbb{R}^{d-3m}$ be the network obtained from θ by dropping its degenerate neurons. Since the generalized gradient is assumed to agree with the partial derivatives of the loss coordinate-wise when the latter exist, it follows that the generalized gradient defined on $\mathbb{R}^{d-3m}$ is continuous in a neighborhood of ϑ. This and (3.2) show that ϑ is a critical point of the loss function defined on $\mathbb{R}^{d-3m}$. Moreover, since we only removed degenerate neurons, the value of the loss at ϑ is equal to $\mathcal{L}(\theta)$. Proposition 3.5 and the assumption on $\mathcal{L}(\theta)$ imply that ϑ is a global minimum or a saddle point with N type-2-active neurons (because ϑ belongs to the first or second layer of critical points). In the former case, θ is also a global minimum of $\mathcal{L}$ and we are done. In the latter case, $m = 0$, so $\theta = \vartheta$ is a saddle point of $\mathcal{L}$. We already know that all neurons of ϑ are type-2-active neurons, so $\theta \in \mathcal{S}$. Finally, Theorem 3.7 tells us that γ or θ_0 belongs to a set of Lebesgue measure zero. ■

4. Proof of the center-stable manifold theorem

In this section, we present a proof of Theorem 2.2. The structure of the proof follows the appendix of [100] with some modifications. We begin with a lemma needed later on.

4.1 Auxiliary lemma

The following lemma involves the existence of bump functions on balls of radii $r > 0$ with bounds on their derivative independent of r.

Lemma 4.1. *For all $r \in (0, \infty)$, there exists $\rho_r \in C^\infty(\mathbb{R}^d, \mathbb{B}_r(0))$ with support in $\mathbb{B}_r(0)$, which is the identity on $\mathbb{B}_{r/2}(0)$, such that the Frobenius norm of ρ_r' is uniformly bounded by $6\sqrt{d}$, so, in particular, ρ_r is $6\sqrt{d}$-Lipschitz continuous.*

Proof. This can be achieved, for example, by taking a function $\sigma \in C^\infty(\mathbb{R}, [0,1])$ such that σ is 1 on $(-\infty, 1]$, it is 0 on $[4, \infty)$, and $\sigma'(x) \in [-2/3, 0]$ for all $x \in \mathbb{R}$. A possible choice for σ would be $\sigma(x) = e^{3/(x-4)}\left[e^{3/(x-4)} + e^{3/(1-x)}\right]^{-1}$ for $x \in (1,4)$. Then, set $\rho_r(x) = x\sigma\left(4\|x\|^2/r^2\right)$. We estimate the square of the Frobenius norm of $d\rho_r$ by

$$\begin{aligned}
&\sum_{j,k=1}^{d} \left[\frac{\partial}{\partial x_k}\rho_r(x)_j\right]^2 \\
&= d\underbrace{\left[\sigma\left(\frac{4\|x\|^2}{r^2}\right)\right]^2}_{\leq 1} + \frac{16\|x\|^2}{r^2}\sigma\left(\frac{4\|x\|^2}{r^2}\right)\underbrace{\sigma'\left(\frac{4\|x\|^2}{r^2}\right)}_{\leq 0} + \frac{64\|x\|^4}{r^4}\underbrace{\left[\sigma'\left(\frac{4\|x\|^2}{r^2}\right)\right]^2}_{\leq 4/9} \\
&\leq d + \frac{256}{9} < 36d.
\end{aligned}$$

■

4.2 Proof of the theorem in the diagonal case

In this section, we will proof Theorem 2.2 in a special case. Denote $s = \dim(E_z^{cs}) \in \{0, \dots, d-1\}$ and $A = f'(z) \in \mathbb{R}^{d\times d}$. Assume that $z = 0$ and that A is a diagonal matrix with diagonal entries $\lambda_1, \dots, \lambda_d$ such that $\lambda_1, \dots, \lambda_s \in [-1, 1]$ and $\lambda_{s+1}, \dots, \lambda_d \in \mathbb{R}\backslash[-1, 1]$. We deduce the general case in the next section. Denote by $B \in \mathbb{R}^{s\times s}$ the diagonal matrix with diagonal entries $\lambda_1, \dots, \lambda_s$ and by $C \in \mathbb{R}^{(d-s)\times(d-s)}$ the diagonal matrix with diagonal entries $\lambda_{s+1}, \dots, \lambda_d$. Denote projections $\Pi^+ \colon \mathbb{R}^d \to E^{cs} = E_z^{cs}$ and $\Pi^- \colon \mathbb{R}^d \to E^u = E_z^u$ onto the first s coordinates and onto the last $d - s$ coordinates, respectively. For any $x \in \mathbb{R}^d$, we write $x^+ = \Pi^+(x)$ and $x^- = \Pi^-(x)$ so that $x = (x^+, x^-)$. Similarly, we write $g^+ = \Pi^+ \circ g$ and $g^- = \Pi^- \circ g$ for any function $g \colon \mathbb{R}^d \to \mathbb{R}^d$ so that $g(x) = (g^+(x), g^-(x))$. Note that

$$\max\left\{ \left\|x^+\right\|, \left\|x^-\right\| \right\} \le \|x\| \le \left\|x^+\right\| + \left\|x^-\right\|.$$

We use the following convention throughout this proof: we denote by $A^0 \in \mathbb{R}^{d\times d}$ and $B^0 \in \mathbb{R}^{s\times s}$ identity matrices even if one of the entries of the matrices A and B is zero. The matrices $A^j \in \mathbb{R}^{d\times d}$, $j \in \mathbb{N}_0$, split into a center-stable and an unstable component. More precisely, they take on the block form

$$A^j = \begin{pmatrix} B^j & 0 \\ 0 & C^j \end{pmatrix} : E^{cs} \oplus E^u \to E^{cs} \oplus E^u, \quad x \mapsto (B^j x^+, C^j x^-).$$

Denote by $\eta \colon \mathbb{R}^d \to \mathbb{R}^d$ the remainder term of the first-order Taylor expansion of f around 0, that is the map $\eta(x) = f(x) - Ax$. The dynamical system is given for all $k \in \mathbb{N}_0$, $x \in \mathbb{R}^d$ by

$$f^k(x) = A^k x + \sum_{i=1}^{k} A^{k-i} \eta(f^{i-1}(x)),$$

which can easily be shown by induction on k. This can be written in the center-stable and unstable components as

$$\begin{aligned} (f^k)^+(x) &= B^k x^+ + \sum_{i=1}^{k} B^{k-i} \eta^+(f^{i-1}(x)), \\ (f^k)^-(x) &= C^k x^- + \sum_{i=1}^{k} C^{k-i} \eta^-(f^{i-1}(x)). \end{aligned} \tag{4.1}$$

In particular, we obtain

$$x^- = C^{-k}(f^k)^-(x) - \sum_{i=1}^{k} C^{-i} \eta^-(f^{i-1}(x)). \tag{4.2}$$

Next, define $\mu = \min_{j\in\{s+1,\dots,d\}} |\lambda_j| = \|C^{-1}\|^{-1} \in (1, \infty)$ and let $\omega = (\omega_k)_{k\ge0} \subseteq (0, 1]$ be given by $\omega_k = \mu^{-k/2}$. Note that the space $\mathcal{C}_w = \{(x_k)_{k\ge0} \subseteq \mathbb{R}^d \colon \sup_{k\ge0} \omega_k \|x_k\| < \infty\}$ equipped with $\|(x_k)_{k\ge0}\|_w = \sup_{k\ge0} \omega_k \|x_k\|$ is a Banach space since it is isomorphic to the Banach space ℓ^∞ of bounded sequences. If $x = (x_k)_{k\ge0} \in \mathcal{C}_w$, then

$$\left\|C^{-k} x_k^-\right\| \le \left\|C^{-k}\right\| \left\|x_k^-\right\| = \omega_k^2 \left\|x_k^-\right\| \le \omega_k \|x\|_w \xrightarrow{k\to\infty} 0.$$

Let us introduce the orbit map $\mathcal{O}$ to the space of sequences $\mathcal{C} = \{(x_k)_{k\geq 0} \subseteq \mathbb{R}^d\}$;

$$\mathcal{O}\colon \mathbb{R}^d \to \mathcal{C}, \quad x \mapsto \mathcal{O}x = (f^k(x))_{k\geq 0}.$$

If $\mathcal{O}x \in \mathcal{C}_w$, then $\left\|C^{-k}(f^k)^-(x)\right\| \to 0$ as $k \to \infty$, so the partial sums in (4.2) converge in this case. Thus, if $\mathcal{O}x \in \mathcal{C}_w$, then

$$x^- = -\sum_{i=1}^{\infty} C^{-i}\eta^-(f^{i-1}(x)).$$

Plugging this into (4.1) yields

$$(f^k)^-(x) = -\sum_{i=k+1}^{\infty} C^{k-i}\eta^-(f^{i-1}(x)) \tag{4.3}$$

for all $x \in \mathbb{R}^d$ with $\mathcal{O}x \in \mathcal{C}_w$. Let ρ_r be the functions promised by Lemma 4.1 and let r_ε be the radii from Assumption 2.1. Since $\rho_{r_\varepsilon}(\mathbb{R}^d) \subseteq \mathbb{B}_{r_\varepsilon}(0)$ and since η is ε-Lipschitz continuous on $\mathbb{B}_{r_\varepsilon}(0)$ by assumption with $\eta(0) = 0$, we have, for all $x \in \mathcal{C}_w$ and $k \in \mathbb{N}$,

$$\begin{aligned}
&\sum_{i=k+1}^{\infty} \left\|C^{k-i}\eta^-(\rho_{r_\varepsilon}(x_{i-1}))\right\| \\
&\leq \sum_{i=k+1}^{\infty} \left\|C^{k-i}\right\| \left\|\eta^-(\rho_{r_\varepsilon}(x_{i-1})) - \eta^-(\rho_{r_\varepsilon}(0))\right\| \\
&\leq 6\varepsilon\sqrt{d} \sum_{i=k+1}^{\infty} \left\|C^{k-i}\right\| \|x_{i-1}\| = 6\varepsilon\sqrt{d} \sum_{i=k+1}^{\infty} \omega_{i-k}\omega_{i-1} \|x_{i-1}\| \omega_{k-1}^{-1} \\
&\leq 6\varepsilon\sqrt{d} \|x\|_w \omega_{k-1}^{-1} \sum_{i=k+1}^{\infty} \omega_{i-k} = 6\varepsilon\sqrt{d} \|x\|_w \omega_{k-1}^{-1} \frac{\omega_1}{1-\omega_1} < \infty
\end{aligned} \tag{4.4}$$

and for $k = 0$

$$\begin{aligned}
&\sum_{i=1}^{\infty} \left\|C^{-i}\eta^-(\rho_{r_\varepsilon}(x_{i-1}))\right\| \\
&\leq \left\|C^{-1}\eta^-(\rho_{r_\varepsilon}(x_0))\right\| + \left\|C^{-1}\right\| \sum_{i=2}^{\infty} \left\|C^{1-i}\eta^-(\rho_{r_\varepsilon}(x_{i-1}))\right\| \\
&\leq \left\|C^{-1}\right\| 6\varepsilon\sqrt{d} \|x\|_w + \left\|C^{-1}\right\| 6\varepsilon\sqrt{d} \|x\|_w \frac{\omega_1}{1-\omega_1} = 6\varepsilon\sqrt{d} \|x\|_w \frac{\omega_1^2}{1-\omega_1}.
\end{aligned} \tag{4.5}$$

Hence, for all $\varepsilon \in (0,1)$ and $y \in E^{cs}$, the map $T_y^\varepsilon\colon \mathcal{C}_w \to \mathcal{C}$ given by

$$(T_y^\varepsilon x)_k = \begin{pmatrix} B^k y + \sum_{i=1}^{k} B^{k-i}\eta^+(\rho_{r_\varepsilon}(x_{i-1})) \\ -\sum_{i=k+1}^{\infty} C^{k-i}\eta^-(\rho_{r_\varepsilon}(x_{i-1})) \end{pmatrix} \in E^{cs} \oplus E^u$$

for all $k \geq 0$ is well-defined. We write $\mathcal{U}_\varepsilon \subseteq \mathbb{R}^d$ for the set $\mathcal{U}_\varepsilon = \{x \in \mathbb{R}^d\colon f^k(x) \in \mathbb{B}_{r_\varepsilon/2}(0) \text{ for all } k \in \mathbb{N}_0\}$. In (4.1) and (4.3) above, we established that if $x \in \mathcal{U}_\varepsilon$ (which

implies $\mathcal{O}x \in \mathcal{C}_w$), then $\mathcal{O}x$ is a fixed point of $T^\varepsilon_{x^+}$. Since $\|B^j\| \le 1$ for all $j \in \mathbb{N}_0$, we have, for all $x \in \mathcal{C}_w$ and $k \in \mathbb{N}_0$,

$$\begin{aligned}
\omega_k \left\| B^k y + \sum_{i=1}^{k} B^{k-i}\eta^+(\rho_{r_\varepsilon}(x_{i-1})) \right\| &\le \omega_k \|y\| + \omega_k \sum_{i=1}^{k} \left\|\eta^+(\rho_{r_\varepsilon}(x_{i-1}))\right\| \\
&\le \omega_k \|y\| + 6\varepsilon\sqrt{d}\omega_k \sum_{i=1}^{k} \|x_{i-1}\| \\
&= \omega_k \|y\| + 6\varepsilon\sqrt{d} \sum_{i=1}^{k} \omega_{k-i+1}\omega_{i-1} \|x_{i-1}\| \\
&\le \omega_k \|y\| + 6\varepsilon\sqrt{d} \|x\|_w \sum_{i=1}^{k} \omega_{k-i+1} \\
&= \omega_k \|y\| + 6\varepsilon\sqrt{d} \|x\|_w \, \omega_1 \frac{1-\omega_k}{1-\omega_1} \\
&\le \|y\| + 6\varepsilon\sqrt{d} \|x\|_w \frac{\omega_1}{1-\omega_1}.
\end{aligned}$$

Together with (4.4) and (4.5), we obtain, for all $y \in E^{cs}$, $x \in \mathcal{C}_w$, and $k \in \mathbb{N}_0$,

$$\omega_k \left\|(T^\varepsilon_y x)_k\right\| \le \|y\| + 12\varepsilon\sqrt{d} \|x\|_w \frac{\omega_1}{1-\omega_1},$$

so $T^\varepsilon_y(\mathcal{C}_w) \subseteq \mathcal{C}_w$. By essentially the same calculations, we find, for all $x^1, x^2 \in \mathcal{C}_w$ and $k \in \mathbb{N}_0$,

$$\omega_k \left\|(T^\varepsilon_y x^1)_k - (T^\varepsilon_y x^2)_k\right\| \le 12\varepsilon\sqrt{d} \left\|x^1 - x^2\right\|_w \frac{\omega_1}{1-\omega_1}.$$

In other words, the restriction $T^\varepsilon_y \colon \mathcal{C}_w \to \mathcal{C}_w$ is $12\varepsilon\sqrt{d}\omega_1(1-\omega_1)^{-1}$ Lipschitz continuous with respect to $\|\cdot\|_w$. In particular, for all $y \in E^{cs}$ and $\varepsilon \in (0, (1-\omega_1)(12\sqrt{d}\omega_1)^{-1})$, the restriction $T^\varepsilon_y \colon \mathcal{C}_w \to \mathcal{C}_w$ is a contraction. Now, let $\varepsilon = (1-\omega_1)(24\sqrt{d}\omega_1)^{-1}$. By the Banach Fixed Point Theorem, there is a unique fixed point map $\Phi \colon E^{cs} \to \mathcal{C}_w$ specified by $T^\varepsilon_y \Phi(y) = \Phi(y)$. Note that, for any $y_1, y_2 \in E^{cs}$, $x \in \mathcal{C}_w$, and $k \in \mathbb{N}_0$,

$$(T^\varepsilon_{y_1} x - T^\varepsilon_{y_2} x)_k = \begin{pmatrix} B^k(y_1 - y_2) \\ 0 \end{pmatrix} \in E^{cs} \oplus E^u.$$

Thus,

$$\begin{aligned}
\|\Phi(y_1) - \Phi(y_2)\|_w &\le \left\|T^\varepsilon_{y_1}\Phi(y_1) - T^\varepsilon_{y_2}\Phi(y_1)\right\|_w + \left\|T^\varepsilon_{y_2}\Phi(y_1) - T^\varepsilon_{y_2}\Phi(y_2)\right\|_w \\
&\le \|y_1 - y_2\| + \frac{1}{2}\|\Phi(y_1) - \Phi(y_2)\|_w
\end{aligned}$$

and, hence,

$$\|\Phi(y_1) - \Phi(y_2)\|_w \le 2\|y_1 - y_2\|.$$

So, $\Phi \colon E^{cs} \to \mathcal{C}_w$ is Lipschitz continuous. Denote by $\Psi \colon E^{cs} \to E^u$ the map $\Psi(y) = (\Phi(y))^-_0$. Then, for all $y_1, y_2 \in E^{cs}$,

$$\|\Psi(y_1) - \Psi(y_2)\| \le \|\Phi(y_1) - \Phi(y_2)\|_w \le 2\|y_1 - y_2\|,$$

so Ψ is also Lipschitz continuous. We noted above that if $x \in \mathcal{U}_\varepsilon$, then $\mathcal{O}x$ is a fixed point of $T^\varepsilon_{x^+}$. Thus, if $x \in \mathcal{U}_\varepsilon$, then $\mathcal{O}x = \Phi(x^+)$ and $x^- = \Psi(x^+)$. In other words, we have shown that $\mathcal{U}_\varepsilon \subseteq \mathrm{Graph}(\Psi)$. This proves Theorem 2.2 in the diagonal case.

4.3 Proof of the theorem in the general case

In this section, we prove Theorem 2.2 in the general case by reducing it to the special case from the previous section. As before, denote $s = \dim(E_z^{cs})$. Since $f'(z)$ is diagonalizable, there is an invertible matrix $Q \in \mathbb{R}^{d\times d}$ such that $Q^{-1}f'(z)Q$ is a diagonal matrix, of which the first s entries lie in $[-1,1]$ and the last $d-s$ entries lie in $\mathbb{R}\backslash[-1,1]$. Set $\tilde{f}(x) = Q^{-1}f(z+Qx)-Q^{-1}z$. Given $\varepsilon \in (0,\infty)$, set $\delta(\varepsilon) = \varepsilon/(\|Q\|\,\|Q^{-1}\|)$ and $\tilde{r}_\varepsilon = r_{\delta(\varepsilon)}/\|Q\|$, where r_δ are the radii from Assumption 2.1. Then, $\tilde{f}$ and $\tilde{r}_\varepsilon$ satisfy Assumption 2.1 at the point 0. Indeed, if $x, y \in \mathbb{B}_{\tilde{r}_\varepsilon}(0)$, then $z+Qx, z+Qy \in \mathbb{B}_{r_{\delta(\varepsilon)}}(z)$ and

$$\begin{aligned}
&\left\|\tilde{f}(x) - \tilde{f}'(0)x - \big(\tilde{f}(y) - \tilde{f}'(0)y\big)\right\| \\
&= \left\|Q^{-1}\big[f(z+Qx) - z - f'(z)(z+Qx-z) - \big(f(z+Qy) - z - f'(z)(z+Qy-z)\big)\big]\right\| \\
&\le \left\|Q^{-1}\right\|\delta(\varepsilon)\,\|z+Qx-(z+Qy)\| \le \varepsilon.
\end{aligned}$$

By the theorem for the diagonal case, there exist an $\tilde{r} \in (0,\infty)$ and a Lipschitz continuous map $\tilde{\Psi}\colon \tilde{E}^{cs} \to \tilde{E}^u$ such that $\{x \in \mathbb{R}^d\colon \tilde{f}^k(x) \in \mathbb{B}_{\tilde{r}}(0) \text{ for all } k \in \mathbb{N}_0\} \subseteq \operatorname{Graph}(\tilde{\Psi})$. Note that $E_z^{cs} = Q\tilde{E}^{cs}$ and $E_z^u = Q\tilde{E}^u$. Now, set $r = \tilde{r}/\|Q^{-1}\|$. Observe that $\tilde{f}^k(x) = Q^{-1}f^k(z+Qx) - Q^{-1}z$ for all $x \in \mathbb{R}^d$ and $k \in \mathbb{N}_0$. In particular, if $f^k(x) \in \mathbb{B}_r(z)$, then $\tilde{f}^k(Q^{-1}(x-z)) \in \mathbb{B}_{\tilde{r}}(0)$. Thus, if $y \in \{x \in \mathbb{R}^d\colon f^k(x) \in \mathbb{B}_r(z) \text{ for all } k \in \mathbb{N}_0\}$, then $Q^{-1}(y-z) \in \operatorname{Graph}(\tilde{\Psi})$ and, hence, $y \in Q(\operatorname{Graph}(\tilde{\Psi})) + z$. Define $\Psi\colon E_z^{cs} \to E_z^u$ by

$$\Psi(x) = Q\tilde{\Psi}(Q^{-1}(x - \Pi^+(z))) + \Pi^-(z),$$

where $\Pi^+\colon \mathbb{R}^d \to E_z^{cs}$ and $\Pi^-\colon \mathbb{R}^d \to E_z^u$ are the projections given by $\Pi^\pm(x) = Q\tilde{\Pi}^\pm(Q^{-1}x)$. Then, $Q(\operatorname{Graph}(\tilde{\Psi})) + z = \operatorname{Graph}(\Psi)$, which finishes the proof of Theorem 2.2.

CHAPTER 6

Outlook

In this thesis, we considered neural network theory from the approximation and the optimization point of view. In Chapter 2, we constructed a framework for neural networks to approximate functions without the curse of dimensionality. Therein, we relied on what we coined the c-identity requirement (Definition 2.4 in Chapter 2), which is a condition on the activation function, ensuring that neural networks of depth at least two can represent identity functions exactly. It is natural to wonder whether one can relax the c-identity requirement. The answer is 'yes' as long as we restrict ourselves to approximations on compact sets, which we did not do in Chapter 2. In fact, for the approximation on compact sets, the following weak assumption is sufficient: suppose the activation function admits a point of differentiability at which its derivative does not vanish. Then, the c-identity requirement is "asymptotically satisfied" in the sense that we can approximate the identity function on any compact set to any desired accuracy with a shallow network, whose number of neurons is independent of the compact set and the accuracy; [87, 107]. This includes virtually all activation functions except for the (noncontinuous) Heaviside activation; [76]. Furthermore, regardless of the c-identity requirement, any approximation task on compact sets that can be solved with ReLU networks can also be solved with sigmoidal networks up to increasing the number of neurons by a constant factor. This is due to sigmoidal networks with six neurons being able to approximate the ReLU activation function on any compact set to any accuracy. This kind of approximations with an architecture independent of the accuracy, which are known in the approximation of polynomials (see [30, 107]), can be taken further to the approximation of arbitrary polynomial splines.

Another aspect is the depth-width trade-off. Some of the examples in Chapter 2 featured wide networks of limited depth, others featured networks of unlimited depth and width. Using that networks can represent identity functions, one can shift depth and width around. This has been demonstrated for ReLU networks in [53, 55, 91] but continues to hold for all other activation functions that can asymptotically represent identity functions as discussed above. However, depth and width do not contribute equally to the expressiveness of a network; recall [25, 39, 109] from Chapter 1 and also see [13, 27, 91, 105, 129, 130].

In the context of optimization, this thesis presented results based on a landscape analysis of the loss surface of the true loss as a function of the network parameters. Another possibility is to study the loss defined on a function space, which does not take the parameter vector as input but the realization function of the network. One is then interested in how the two different landscapes interact. This raises the question of stability of the realization operator that maps a parameter vector to its realization function. In general, neural networks fail

this inverse stability; [104]. For shallow ReLU networks, by restricting the parameter space and considering realizations in a Sobolev norm, inverse stability can be recovered; [9]. But for ReLU networks, the Sobolev norm is a very strong norm in the sense that the realization operator is no longer continuous. For input and output dimension one, it is possible to drop the Sobolev norm and still obtain a local inverse stability, which is sufficient to relate local minima of the two different landscapes. Trying to establish local inverse stability for higher input dimensions would require a careful study of the tessellation of the input space into the convex polytopes on which the realization function is piecewise linear. This tessellation is tractable for shallow networks, but its complexity can grow exponentially with the depth of the network; [95, 101].

Since training algorithms act on the parameter space and since said local inverse stability only holds on certain subsets, it is more promising to study the landscape in parameter space, which we did in the second part of this thesis. We stress again that the landscape analysis in Chapter 4 was conducted for a fixed number of hidden neurons. In particular, the classification did not rely on an over-parametrization, with which we would enter the regime of the neural tangent kernel or of a many-particle flow; [19, 21, 63]. At the other extreme, previous articles had studied the case of a single neuron; [46, 119]. We used our classification from Chapter 4 to study a convergence property of the gradient descent algorithm in Chapter 5. Inspired by the ideas introduced in [14] and in Chapters 4 and 5, subsequent works deduced other related results about gradient-based algorithms. In the simpler case of constant target functions, the convergence of gradient descent in [14] has been extended to its stochastic analogue in [65] and has been adapted to the setting of deep networks in [61]. For affine target functions, the convergence of gradient descent in Chapter 5 has been extended to its continuous analogue, gradient flows, in [67]. For the discrete algorithm, the range of possible target functions has been broadened to piecewise affine in [66]. Finally, [37, 62, 64] study similar problems as Chapters 4 and 5 for piecewise polynomial target functions.

Bibliography

[1] Absil, P. A., Mahony, R., and Andrews, B. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization 16*, 2 (2005), 531–547.

[2] Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *Proceedings of the 36th International Conference on Machine Learning* (09–15 Jun 2019), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 242–252.

[3] Bah, B., Rauhut, H., Terstiege, U., and Westdickenberg, M. Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. *Information and Inference: A Journal of the IMA* (02 2021).

[4] Baldi, P., and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks 2*, 1 (1989), 53–58.

[5] Barron, A. R. Neural net approximation. *Proceedings of the 7th Yale Workshop on Adaptive and Learning Systems* (1992), 69–72.

[6] Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory 39*, 3 (1993), 930–945.

[7] Barron, A. R. Approximation and estimation bounds for artificial neural networks. *Machine Learning 14*, 1 (Jan 1994), 115–133.

[8] Beck, C., Jentzen, A., and Kuckuck, B. Full error analysis for the training of deep neural networks. *Infinite Dimensional Analysis, Quantum Probability and Related Topics 25*, 02 (2022), 2150020.

[9] Berner, J., Elbrächter, D. M., and Grohs, P. How degenerate is the parametrization of neural networks with the ReLU activation function? In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 7790–7801.

[10] Bhojanapalli, S., Neyshabur, B., and Srebro, N. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems* (2016), D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc.

[11] Bölcskei, H., Grohs, P., Kutyniok, G., and Petersen, P. Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math. Data Sci. 1*, 1 (2019), 8–45.

[12] Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review 60*, 2 (2018), 223–311.

[13] Chatziafratis, V., Nagarajan, S. G., Panageas, I., and Wang, X. Depth-width trade-offs for relu networks via sharkovsky's theorem. *arXiv:1912.04378v1* (2019).

[14] Cheridito, P., Jentzen, A., Riekert, A., and Rossmannek, F. A proof of convergence for gradient descent in the training of artificial neural networks for constant target functions. *Journal of Complexity 72* (2022), 101646.

[15] Cheridito, P., Jentzen, A., and Rossmannek, F. Non-convergence of stochastic gradient descent in the training of deep neural networks. *Journal of Complexity 64* (2021), 101540.

[16] Cheridito, P., Jentzen, A., and Rossmannek, F. Efficient approximation of high-dimensional functions with neural networks. *IEEE Transactions on Neural Networks and Learning Systems 33*, 7 (2022), 3079–3093.

[17] Cheridito, P., Jentzen, A., and Rossmannek, F. Gradient descent provably escapes saddle points in the training of shallow relu networks. *arXiv:2208.02083v1* (2022).

[18] Cheridito, P., Jentzen, A., and Rossmannek, F. Landscape analysis for shallow neural networks: Complete classification of critical points for affine target functions. *Journal of Nonlinear Science 32*, 5 (Jul 2022), 64.

[19] Chizat, L., and Bach, F. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 3036–3046.

[20] Chizat, L., and Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Proceedings of Thirty Third Conference on Learning Theory* (09–12 Jul 2020), J. Abernethy and S. Agarwal, Eds., vol. 125 of *Proceedings of Machine Learning Research*, PMLR, pp. 1305–1338.

[21] Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 2937–2947.

[22] Choromanska, A., Henaff, M., Mathieu, M., Ben Arous, G., and LeCun, Y. The Loss Surfaces of Multilayer Networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* (09–12 May 2015), G. Lebanon and S. V. N. Vishwanathan, Eds., vol. 38 of *Proceedings of Machine Learning Research*, PMLR, pp. 192–204.

[23] Choromanska, A., LeCun, Y., and Ben Arous, G. Open problem: The landscape of the loss surfaces of multilayer networks. In *Proceedings of The 28th Conference on Learning Theory* (03–06 Jul 2015), P. Grünwald, E. Hazan, and S. Kale, Eds., vol. 40 of *Proceedings of Machine Learning Research*, PMLR, pp. 1756–1760.

[24] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems 2*, 4 (1989), 303–314.

[25] Daniely, A. Depth separation for neural networks. In *Proceedings of the 2017 Conference on Learning Theory* (2017), PMLR, pp. 690–696.

[26] Daskalakis, C., and Panageas, I. The limit points of (optimistic) gradient descent in min-max optimization. In *Advances in Neural Information Processing Systems* (2018), S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc.

[27] Daubechies, I., DeVore, R., Foucart, S., Hanin, B., and Petrova, G. Nonlinear approximation and (deep) relu networks. *Constructive Approximation 55*, 1 (Feb 2022), 127–172.

[28] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2933–2941.

[29] Davis, D., Drusvyatskiy, D., Kakade, S., and Lee, J. D. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics 20*, 1 (Feb 2020), 119–154.

[30] De Ryck, T., Lanthaler, S., and Mishra, S. On the approximation of functions by tanh neural networks. *Neural Networks 143* (2021), 732–750.

[31] Donahue, M. J., Darken, C., Gurvits, L., and Sontag, E. Rates of convex approximation in non-Hilbert spaces. *Constructive Approximation 13*, 2 (Jun 1997), 187–220.

[32] Du, S. S., Jin, C., Lee, J. D., Jordan, M. I., Singh, A., and Poczos, B. Gradient descent can take exponential time to escape saddle points. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[33] Du, S. S., and Lee, J. On the power of over-parametrization in neural networks with quadratic activation. In *Proceedings of the 35th International Conference on Machine Learning* (Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018), J. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 1329–1338.

[34] Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations* (2019).

[35] E, W., Han, J., and Jentzen, A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics 5*, 4 (Dec 2017), 349–380.

[36] E, W., Ma, C., and Wu, L. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Science China Mathematics* (2020).

[37] Eberle, S., Jentzen, A., Riekert, A., and Weiss, G. S. Existence, uniqueness, and convergence rates for gradient flows in the training of artificial neural networks with ReLU activation. *arXiv:2108.08106v1* (2021).

[38] Elbrächter, D., Perekrestenko, D., Grohs, P., and Bölcskei, H. Deep neural network approximation theory. *IEEE Transactions on Information Theory 67*, 5 (2021), 2581–2623.

[39] Eldan, R., and Shamir, O. The power of depth for feedforward neural networks. In *29th Annual Conference on Learning Theory* (2016), PMLR, pp. 907–940.

[40] Frankel, P., Garrigos, G., and Peypouquet, J. Splitting methods with variable metric for Kurdyka Łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications 165*, 3 (Jun 2015), 874–900.

[41] Fukumizu, K., and Amari, S.-i. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks 13*, 3 (2000), 317–327.

[42] Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural Networks 2*, 3 (1989), 183–192.

[43] Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle points — online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory* (03–06 Jul 2015), P. Grünwald, E. Hazan, and S. Kale, Eds., vol. 40 of *Proceedings of Machine Learning Research*, PMLR, pp. 797–842.

[44] Ge, R., Jin, C., and Zheng, Y. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *Proceedings of the 34th International Conference on Machine Learning* (06–11 Aug 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1233–1242.

[45] Girosi, F., and Anzellotti, G. Rates of convergence for radial basis functions and neural networks. In *Artificial Neural Networks for Speech and Vision*, R. J. Mammone, Ed. Chapman & Hall, 1993, pp. 97–113.

[46] Goel, S., Kanade, V., Klivans, A., and Thaler, J. Reliably learning the relu in polynomial time. In *Proceedings of the 2017 Conference on Learning Theory* (07–10 Jul 2017), S. Kale and O. Shamir, Eds., vol. 65 of *Proceedings of Machine Learning Research*, PMLR, pp. 1004–1042.

[47] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

[48] Grohs, P., Hornung, F., Jentzen, A., and von Wurstemberger, P. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations. *arXiv:1809.02362v1* (2018). accepted in Mem. Amer. Math. Soc.

[49] Grohs, P., Hornung, F., Jentzen, A., and Zimmermann, P. Space-time error estimates for deep neural network approximations for differential equations. *Advances in Computational Mathematics 49*, 1 (Jan 2023), 4.

[50] Guliyev, N. J., and Ismailov, V. E. Approximation capability of two hidden layer feedforward neural networks with fixed weights. *Neurocomputing 316* (2018), 262–269.

[51] Gurvits, L., and Koiran, P. Approximation and learning of convex superpositions. *Journal of Computer and System Sciences 55*, 1 (1997), 161–170.

[52] Hanin, B. Which neural net architectures give rise to exploding and vanishing gradients? In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 582–591.

[53] Hanin, B. Universal function approximation by deep neural nets with bounded width and ReLU activations. *Mathematics 7*, 10 (2019), 992.

[54] Hanin, B., and Rolnick, D. How to start training: The effect of initialization and architecture. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 571–581.

[55] Hanin, B., and Sellke, M. Approximating continuous functions by ReLU nets of minimal width. *arXiv:1710.11278v2* (2018).

[56] Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks 4*, 2 (1991), 251–257.

[57] Hornik, K. Some new results on neural network approximation. *Neural Networks 6*, 8 (1993), 1069–1072.

[58] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks 2*, 5 (1989), 359–366.

[59] Hornik, K., Stinchcombe, M., and White, H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks 3*, 5 (1990), 551–560.

[60] Hutzenthaler, M., Jentzen, A., Kruse, T., and Nguyen, T. A. A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations. *SN Partial Differential Equations and Applications 1*, 2 (Apr 2020), 10.

[61] Hutzenthaler, M., Jentzen, A., Pohl, K., Riekert, A., and Scarpa, L. Convergence proof for stochastic gradient descent in the training of deep neural networks with relu activation for constant target functions. *arXiv:2112.07369v1* (2021).

[62] Ibragimov, S., Jentzen, A., Kröger, T., and Riekert, A. On the existence of infinitely many realization functions of non-global local minima in the training of artificial neural networks with relu activation. *arXiv:2202.11481v1* (2022).

[63] Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 8571–8580.

[64] Jentzen, A., and Riekert, A. On the existence of global minima and convergence analyses for gradient descent methods in the training of deep neural networks. *Journal of Machine Learning 1*, 2 (2022), 141–246.

[65] Jentzen, A., and Riekert, A. A proof of convergence for stochastic gradient descent in the training of artificial neural networks with relu activation for constant target functions. *Zeitschrift für angewandte Mathematik und Physik 73*, 5 (Aug 2022), 188.

[66] Jentzen, A., and Riekert, A. A proof of convergence for the gradient descent optimization method with random initializations in the training of neural networks with relu activation for piecewise linear target functions. *Journal of Machine Learning Research 23*, 260 (2022), 1–50.

[67] Jentzen, A., and Riekert, A. Convergence analysis for gradient flows in the training of artificial neural networks with relu activation. *Journal of Mathematical Analysis and Applications 517*, 2 (2023), 126601.

[68] Jentzen, A., Salimova, D., and Welti, T. A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients. *Communications in Mathematical Sciences 19*, 5 (2021), 1167–1205.

[69] Jentzen, A., and Welti, T. Overall error analysis for the training of deep neural networks via stochastic gradient descend with random initialisation. *arXiv:2003.01291v1* (2020).

[70] Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning* (06–11 Aug 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1724–1732.

[71] Jones, L. K. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Ann. Statist. 20*, 1 (1992), 608–613.

[72] Kainen, P. C., Kůrková, V., and Sanguineti, M. Complexity of gaussian-radial-basis networks approximating smooth functions. *Journal of Complexity 25*, 1 (2009), 63–74.

[73] Kainen, P. C., Kůrková, V., and Sanguineti, M. Dependence of computational models on input dimension: Tractability of approximation and optimization tasks. *IEEE Transactions on Information Theory 58*, 2 (Feb 2012), 1203–1214.

[74] Kawaguchi, K. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 586–594.

[75] Klusowski, J. M., and Barron, A. R. Approximation by combinations of relu and squared relu ridge functions with ℓ^1 and ℓ^0 controls. *IEEE Transactions on Information Theory 64*, 12 (Dec 2018), 7649–7656.

[76] Kůrková, V. Approximation of functions by perceptron networks with bounded number of hidden units. *Neural Networks 8*, 5 (1995), 745–750.

[77] Kůrková, V. Minimization of error functionals over perceptron networks. *Neural Computation 20*, 1 (Jan 2008), 252–270.

[78] Kůrková, V., Kainen, P. C., and Kreinovich, V. Estimates of the number of hidden units and variation with respect to half-spaces. *Neural Networks 10*, 6 (1997), 1061–1068.

[79] Kůrková, V., and Sanguineti, M. Comparison of worst case errors in linear and neural network approximation. *IEEE Transactions on Information Theory 48*, 1 (Jan 2002), 264–275.

[80] Kůrková, V., and Sanguineti, M. Geometric upper bounds on rates of variable-basis approximation. *IEEE Transactions on Information Theory 54*, 12 (Dec 2008), 5681–5688.

[81] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521*, 7553 (2015), 436–444.

[82] Lee, H., Ge, R., Ma, T., Risteski, A., and Arora, S. On the ability of neural nets to express distributions. In *Proceedings of the 2017 Conference on Learning Theory* (07–10 Jul 2017), S. Kale and O. Shamir, Eds., vol. 65 of *Proceedings of Machine Learning Research*, PMLR, pp. 1271–1296.

[83] Lee, J. D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M. I., and Recht, B. First-order methods almost always avoid strict saddle points. *Mathematical Programming 176*, 1 (Jul 2019), 311–337.

[84] Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent only converges to minimizers. In *29th Annual Conference on Learning Theory* (23–26 Jun 2016), V. Feldman, A. Rakhlin, and O. Shamir, Eds., vol. 49 of *Proceedings of Machine Learning Research*, PMLR, pp. 1246–1257.

[85] Lei, Y., Hu, T., Li, G., and Tang, K. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE Transactions on Neural Networks and Learning Systems* (2019), 1–7.

[86] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks 6*, 6 (1993), 861–867.

[87] Lin, H. W., Tegmark, M., and Rolnick, D. Why does deep and cheap learning work so well? *J. Stat. Phys. 168*, 6 (2017), 1223–1247.

[88] Livni, R., Shalev-Shwartz, S., and Shamir, O. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 855–863.

[89] Lu, J., Shen, Z., Yang, H., and Zhang, S. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis 53*, 5 (2021), 5465–5506.

[90] Lu, L., Shin, Y., Su, Y., and Karniadakis, G. E. Dying ReLU and Initialization: Theory and Numerical Examples. *arXiv:1903.06733v2* (2019).

[91] Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[92] Maiorov, V., and Pinkus, A. Lower bounds for approximation by MLP neural networks. *Neurocomputing 25*, 1 (1999), 81–91.

[93] Mhaskar, H. N. Approximation properties of a multilayered feedforward artificial neural network. *Adv. Comput. Math. 1*, 1 (1993), 61–80.

[94] Mhaskar, H. N., and Micchelli, C. A. Dimension-independent bounds on the degree of approximation by neural networks. *IBM Journal of Research and Development 38*, 3 (May 1994), 277–284.

[95] Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems* (2014), Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc.

[96] Nesterov, Y. *Lectures on Convex Optimization*, 2nd ed., vol. 137 of *Springer Optimization and Its Applications*. Springer, Cham, 2018.

[97] Nguyen, Q., and Hein, M. The loss surface of deep and wide neural networks. In *Proceedings of the 34th International Conference on Machine Learning* (06–11 Aug 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 2603–2612.

[98] O'Neill, M., and Wright, S. J. Behavior of accelerated gradient methods near critical points of nonconvex functions. *Mathematical Programming 176*, 1 (Jul 2019), 403–427.

[99] Panageas, I., and Piliouras, G. Gradient Descent Only Converges to Minimizers: Non-Isolated Critical Points and Invariant Regions. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)* (2017), C. H. Papadimitriou, Ed., vol. 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 2:1–2:12.

[100] Panageas, I., Piliouras, G., and Wang, X. First-order methods almost always avoid saddle points: The case of vanishing step-sizes. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 6474–6483.

[101] Pascanu, R., Montufar, G., and Bengio, Y. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv:1312.6098v5* (2013).

[102] Pemantle, R. Nonconvergence to Unstable Points in Urn Models and Stochastic Approximations. *The Annals of Probability 18*, 2 (1990), 698 – 712.

[103] Pennington, J., and Bahri, Y. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning* (06–11 Aug 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 2798–2806.

[104] Petersen, P., Raslan, M., and Voigtlaender, F. Topological properties of the set of functions generated by neural networks of fixed size. *Foundations of Computational Mathematics* (May 2020).

[105] Petersen, P., and Voigtlaender, F. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks 108* (2018), 296–330.

[106] Pinkus, A. Approximation theory of the MLP model in neural networks. *Acta Numerica 8* (1999), 143–195.

[107] Rolnick, D., and Tegmark, M. The power of deeper networks for expressing natural functions. In *International Conference on Learning Representations* (2018).

[108] Safran, I., and Shamir, O. On the quality of the initial basin in overspecified neural networks. In *Proceedings of The 33rd International Conference on Machine Learning* (20–22 Jun 2016), M. F. Balcan and K. Q. Weinberger, Eds., vol. 48 of *Proceedings of Machine Learning Research*, PMLR, pp. 774–782.

[109] Safran, I., and Shamir, O. Depth-width tradeoffs in approximating natural functions with neural networks. In *Proceedings of the 34th International Conference on Machine Learning* (2017), PMLR, pp. 2979–2987.

[110] Safran, I., and Shamir, O. Spurious local minima are common in two-layer ReLU neural networks. In *Proceedings of the 35th International Conference on Machine Learning* (10–15 Jul 2018), J. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 4433–4441.

[111] Sarao Mannelli, S., Vanden-Eijnden, E., and Zdeborová, L. Optimization and generalization of shallow neural networks with quadratic activation functions. In *Advances in Neural Information Processing Systems* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., pp. 13445–13455.

[112] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks 61* (2015), 85–117.

[113] Schwab, C., and Zech, J. Deep learning in high dimension: neural network expression rates for generalized polynomial chaos expansions in UQ. *Analysis and Applications 17*, 1 (2019), 19–55.

[114] Shalev-Shwartz, S., and Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[115] Shalev-Shwartz, S., Shamir, O., and Shammah, S. Failures of gradient-based deep learning. In *Proceedings of the 34th International Conference on Machine Learning* (06–11 Aug 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 3067–3075.

[116] Shamir, O. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. In *Proceedings of the Thirty-Second Conference on Learning Theory* (25–28 Jun 2019), A. Beygelzimer and D. Hsu, Eds., vol. 99 of *Proceedings of Machine Learning Research*, PMLR, pp. 2691–2713.

[117] Shin, Y., and Karniadakis, G. E. Trainability of ReLU networks and data-dependent initialization. *Journal of Machine Learning for Modeling and Computing 1*, 1 (2020), 39–74.

[118] Shub, M. *Global Stability of Dynamical Systems*, 1st ed. Springer, New York, NY, 1987.

[119] Soltanolkotabi, M. Learning relus via gradient descent. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 2007–2017.

[120] Soltanolkotabi, M., Javanmard, A., and Lee, J. D. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory 65*, 2 (Feb 2019), 742–769.

[121] Soudry, D., and Carmon, Y. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv:1605.08361v2* (2016).

[122] Soudry, D., and Hoffer, E. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv:1702.05777v5* (2017).

[123] Sun, J., Qu, Q., and Wright, J. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Transactions on Information Theory 63*, 2 (2017), 853–884.

[124] Sun, J., Qu, Q., and Wright, J. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics 18*, 5 (Oct 2018), 1131–1198.

[125] Swirszcz, G., Czarnecki, W. M., and Pascanu, R. Local minima in training of neural networks. *arXiv:1611.06310v2* (2016).

[126] Venturi, L., Bandeira, A. S., and Bruna, J. Spurious valleys in one-hidden-layer neural network optimization landscapes. *Journal of Machine Learning Research 20*, 133 (2019), 1–34.

[127] Voigtlaender, F., and Petersen, P. Approximation in $L^p(\mu)$ with deep ReLU neural networks. In *2019 13th International conference on Sampling Theory and Applications (SampTA)* (2019), pp. 1–4.

[128] Wojtowytsch, S. On the convergence of gradient descent training for two-layer relu-networks in the mean field regime. *arXiv:2005.13530v1* (2020).

[129] Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Networks 94* (2017), 103–114.

[130] Yarotsky, D. Optimal approximation of continuous functions by very deep relu networks. In *Proceedings of the 31st Conference On Learning Theory* (06–09 Jul 2018), S. Bubeck, V. Perchet, and P. Rigollet, Eds., vol. 75 of *Proceedings of Machine Learning Research*, PMLR, pp. 639–649.

[131] Zou, D., Cao, Y., Zhou, D., and Gu, Q. Gradient descent optimizes over-parameterized deep ReLU networks. *Machine Learning 109*, 3 (Mar 2020), 467–492.