Unfavorable structural properties of the set of neural networks with fixed architecture

Philipp Petersen
Mathematical Institute
University of Oxford
Woodstock Road, Oxford
OX2 6GG, UK
E-Mail: Philipp.Petersen@maths.ox.ac.uk

Mones Raslan
Institut für Mathematik
TU Berlin
Straße des 17. Juni 136
10623 Berlin, Germany
E-Mail: raslan@math.tu-berlin.de

Felix Voigtlaender
Wissenschaftliches Rechnen
KU Eichstätt–Ingolstadt
Ostenstraße 26
85072 Eichstätt, Germany
E-Mail: felix.voigtlaender@ku.de

***Abstract*—In this note, we present a variety of results from the recent paper [1] in which the structural properties of the set of functions that can be implemented by neural networks with a fixed architecture have been studied. As it turns out, this set has many unfavorable properties: It is highly non-convex, except possibly for a few uncommon activation functions. Additionally, the set is not closed with respect to L^p-norms, $0 < p < \infty$, for all frequently used activation functions, and also not closed with respect to the L^∞-norm for all practically-used activation functions except for the (parametric) ReLU. Finally, the function that maps a family of parameters to the function computed by the associated network is not inverse stable for every practically used activation function. Overall, our findings identify potential causes for issues in the optimization of neural networks such as no guaranteed or very slow convergence and the explosion of parameters.**

I. Introduction

The term *deep learning* [2] describes a variety of machine learning algorithms based on the employment of *neural networks* which have first been introduced in [3]. Although these methods work extremely well for a huge variety of applications (such as speech recognition or image classification), a thorough understanding of their success is still in its infancy.

One very active research area lies in the examination of the mathematical properties of neural networks, among them the investigation of their approximation properties. The first result in this direction is given by the universal approximation theorem [4] which states that every continuous function defined on a compact set can be approximated arbitrarily well by a two layer neural network if one does not impose any restriction on its width. Subsequent works such as [5], [6], [7] and the references therein study the expressiveness of neural networks for more specific function classes with a particular focus on the tradeoff *network size vs. approximation accuracy*. Moreover, papers such as [8], [7] as well as the references therein examine under which circumstances deep neural networks have a higher expressivity than shallow ones. Although these papers provide significant insight about the capabilities of neural networks, two potential problems of such an approach in view of practical applications can be identified:

On the one hand, all results finally reduce the underlying approximation problem to a classical approximation problem employing polynomial-, wavelet-, or spline approximation thereby assuming that the functions which are to be approximated are contained in some classical approximation space such as a Hölder space, a Sobolev or a Besov space. In contrast, in many applications such as classification tasks such an assumption is not realistic.

On the other hand, the vast majority of approximation theoretical results is of an asymptotic type in the sense that for many functions, as the approximation accuracy converges to zero, the size of the approximating neural network needs to explode. However, as is common in many deep learning methods, one a priori fixes a neural network architecture and only adapts the parameters of the neural network during the training process. Hence, in view of practical applications, a more thorough study of neural networks having a prescribed size is more appropriate.

Although there exist works focussing on different issues related to the architecture of neural networks (such as [9] and the references therein), to the best of our knowledge the *intrinsic structure* of the set of functions generated by neural networks with a fixed architecture has first been studied in [1]. The goal of this manuscript is to present a selection of particularly interesting results as well as their consequences.

We first state the basic notions of neural networks in

Section II with a particular focus on the distinction between a neural network as a *collection of weights* and the *realization* of a neural network as a function. In Section III we recapitulate some statements about the shape of the set of neural network realizations which imply that this set is structured in a significantly different way than most classical approximation spaces. This observation implies that the reduction to classical approximation problems is not sufficient in order to ultimately explain the efficiency of neural networks. Afterwards, we study the closedness of the set of realizations with respect to $L^p,\ 0 < p \leq \infty$ in Section IV, whereas in Section V we concentrate on an analysis of the properties of the map which takes a collection of weights as an input and returns the corresponding realization. These results offer possible explanations for some phenomena frequently observed in practice when optimizing a neural network such as very slow convergence, no convergence at all and exploding network weights.

II. Basic Notions of Neural Networks

In order to state our results, we will distinguish between a *neural network* as a set of weights and the associated function, referred to as its *realization*. To explain this distinction, we first give the following definition of a neural network.

Definition 1. *[7] Let $L \in \mathbb{N}$ be a* number of layers *and $d = N_0 \in \mathbb{N}$ be an* input dimension. *Moreover, let $N_L := 1$ be the* output dimension. *For $N_1, \dots, N_{L-1} \in \mathbb{N}$, we say that a family $\Phi = (W_\ell)_{\ell=1}^L$ of affine linear maps $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ is a* neural network. *We call $S := (d, N_1, \dots, N_L)$ the* architecture *of Φ; furthermore $L = L(S)$ is the* number of layers of *S.*

Now we turn our attention to the definition of the realization of a neural network as a function.

Definition 2. *[7] Let $\Phi = (W_\ell)_{\ell=1}^L$ be a neural network, $\varrho : \mathbb{R} \to \mathbb{R}$ be an* activation function *and $\Omega \subset \mathbb{R}^d$. The* Ω-realization of a neural network *$\Phi = (W_\ell)_{\ell=1}^L$ is the function*

$$\begin{aligned} \mathrm{R}_\varrho^\Omega(\Phi) &: \Omega \to \mathbb{R}, \\ &x \mapsto W_L(\varrho(W_{L-1}(\dots \varrho(W_1(x))))) \,, \end{aligned}$$

where $\varrho(y) := (\varrho(y_1), ..., \varrho(y_m))$ for $y = (y_1, ..., y_m) \in \mathbb{R}^m$.

In the remainder of this manuscript we will always assume $\Omega \subset \mathbb{R}^d$ to be compact with nonempty interior and without any isolated points. We denote by $C(\Omega)$ the Banach space of all *continuous functions defined on Ω with values in $\mathbb{R}$ equipped with the supremum norm* $\|\cdot\|_{\sup}$, which, on $C(\Omega)$, coincides with the L^∞-norm.

In the upcoming sections, we study structural properties of sets of realizations of neural networks with a *fixed architecture* and we denote by $\mathcal{RNN}_\varrho^\Omega(S)$ the *set of Ω-realizations of neural networks with architecture S and activation function ϱ.*

The definition of networks and realizations from above is sufficiently precise so that we can state our results. For proofs and precise calculations we refer to [1].

In principle, the activation function ϱ can be chosen arbitrarily. However, in the framework of deep learning, a variety of activation functions have been identified which turned out to work well in practice. We have listed some of the most common activation functions which we refer to throughout this note in Table I. We emphasize that all activation functions listed below are *globally* Lipschitz continuous functions, whereas many results in [1] remain valid for *locally* Lipschitz continuous functions.

III. Shape of the set of realizations

It is not hard to see that the set of all neural networks with a fixed architecture can be turned into a finite-dimensional vector space. The goal of this section is to argue that the set of corresponding Ω-realizations behaves in a considerably different way for every activation function listed in Table I. First of all, we are able to show that under very mild assumptions imposed on ϱ, which are fulfilled by any of the activation functions given in Table I, the set $\mathcal{RNN}_\varrho^\Omega(S)$ contains *infinitely* many linearly independent functions. Secondly, one observes that for a given architecture S and an arbitrary activation function ϱ, the set $\mathcal{RNN}_\varrho^\Omega(S)$ is *star-shaped*, that is, there exists a *center* $f \in \mathcal{RNN}_\varrho^\Omega(S)$, i.e. for all $g \in \mathcal{RNN}_\varrho^\Omega(S)$, also

$$\{\lambda f + (1-\lambda) g \colon \lambda \in [0,1]\} \subset \mathcal{RNN}_\varrho^\Omega(S) \,.$$

However, for locally Lipschitz continuous activation functions ϱ the set $\mathcal{RNN}_\varrho^\Omega(S)$ has a *finite* number of linearly independent centers. Combining all of the above arguments, we obtain our first negative result about the structure of $\mathcal{RNN}_\varrho^\Omega(S)$.

Theorem 3. *[1, Corollary 3.6.] Let S be an architecture and $\varrho : \mathbb{R} \to \mathbb{R}$ be one of the activation functions given in Table I. Then $\mathcal{RNN}_\varrho^\Omega(S)$ is* not *convex.*

Additionally, it has been demonstrated that for a large class of activation functions (including the ReLU, the tanh and the sigmoid activation function), the set $\mathcal{RNN}_\varrho^\Omega(S)$ turns out to be *highly* non-convex in the

TABLE I
COMMONLY-USED ACTIVATION FUNCTIONS

Name	Given by
rectified linear unit (ReLU)	$\max\{0, x\}$
a-parametric ReLU	$\max\{ax, x\}$ for some $a \geq 0$, $a \neq 1$
exponential linear unit	$x \cdot \chi_{x \geq 0}(x) + (\exp(x) - 1) \cdot \chi_{x<0}(x)$
softsign	$\frac{x}{1+\lvert x \rvert}$
a-inverse square root linear unit	$x \cdot \chi_{x \geq 0}(x) + \frac{x}{\sqrt{1+ax^2}} \cdot \chi_{x<0}(x)$ for $a > 0$
a-inverse square root unit	$\frac{x}{\sqrt{1+ax^2}}$ for $a > 0$
sigmoid / logistic	$\frac{1}{1+\exp(-x)}$
tanh	$\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$
arctan	$\arctan(x)$
softplus	$\ln(1 + \exp(x))$

sense that for every $r \in [0, \infty)$, the set of functions having uniform distance less than r to any function in $\mathcal{RNN}_\varrho^\Omega(S)$ is not convex. This nonconvexity is undesirable, since in the classical statistical learning setting [10], the hypothesis space is often assumed to be convex, and because for non-convex hypothesis spaces, as pointed out in [10, Chapter 7], the learning problem is considerably harder. Moreover, in recent years also neural network based approaches towards the numerical solution of PDEs have gained attention (see for instance [11], [12] and the references therein). In this context, practitioners are primarily interested in the realization of a neural network rather than the corresponding weights and an optimization task is performed over a non-convex set hampering a possible convergence analysis of the underlying algorithm.

IV. (NON-)CLOSEDNESS OF THE SET OF REALIZATIONS

In this section, for any fixed architecture S, we examine the closedness of the set $\mathcal{RNN}_\varrho^\Omega(S)$ with respect to the topologies on $L^p(\Omega)$, $p \in (0, \infty]$. In fact, the following second negative result about the structure of $\mathcal{RNN}_\varrho^\Omega(S)$ holds.

Theorem 4. *[1, Subsection 4.1./Subsection 4.2.] Let S be a neural network architecture and $\varrho : \mathbb{R} \to \mathbb{R}$. Under very general assumptions imposed on ϱ which are satisfied by all activation functions listed in Table I, the set $\mathcal{RNN}_\varrho^\Omega(S)$ is* not *closed in $L^p(\Omega)$ for any $p \in (0, \infty)$.*

In addition, if $\varrho : \mathbb{R} \to \mathbb{R}$ is one of the activation functions given in Table I except for the ReLU or the a-parametric ReLU*, the set $\mathcal{RNN}_\varrho^\Omega(S)$ is not closed in $C(\Omega)$.*

Concerning the (a-parametric) ReLU, we do not know for general architectures S whether closedness of the set $\mathcal{RNN}_\varrho^\Omega(S)$ holds with respect to the topology on $C(\Omega)$. However, we were able to show the following result for the special case $L(S) = 2$.

Theorem 5. *[1, Theorem 4.10.] Let S be a neural network architecture with $L(S) = 2$, let $a \geq 0$ and let $\varrho : \mathbb{R} \to \mathbb{R}$ be given by the a-parametric ReLU. Then the set $\mathcal{RNN}_\varrho^\Omega(S)$ is closed in $C(\Omega)$.*

We now describe two disadvantageous consequences of the non-closedness of $\mathcal{RNN}_\varrho^\Omega(S)$ which also might serve as an argument to use the (a-parametric) ReLU as the underlying activation function, since these problems, at least for two-layered neural networks, do not occur, if the underlying optimization is done over $C(\Omega)$ or $L^\infty(\Omega)$. Firstly, one frequently aims at minimizing a loss function over $\mathcal{RNN}_\varrho^\Omega(S)$. In case the error between a neural network realization and a target function f is measured with respect to the L^p-norm, the non-closedness of $\mathcal{RNN}_\varrho^\Omega(S)$ implies that such a problem does not have a solution for every f in the sense that f does not have a best approximation in $\mathcal{RNN}_\varrho^\Omega(S)$. Secondly, we additionally show that for an arbitrary but fixed $C > 0$ the set

$$\big\{\mathrm{R}_\varrho^\Omega(\Phi) : \ \Phi = (W_\ell)_{\ell=1}^L \text{ has architecture } S \text{ and } \\ W_\ell = A_\ell(\cdot) + b_\ell \text{ with } \|A_\ell\| + \|b_\ell\| \leq C\}$$

of realizations of neural networks with a fixed architecture and all affine linear maps bounded in a suitable norm, *is* closed in $L^p(\Omega)$ for all $p \in (0, \infty]$. This observation implies that if f lies in the L^p-closure of $\mathcal{RNN}_\varrho^\Omega(S)$, but not in $\mathcal{RNN}_\varrho^\Omega(S)$ itself, for any sequence of networks $(\Phi_n)_n$ with architecture S and $\|f - \mathrm{R}_\varrho^\Omega(\Phi_n)\|_{L^p} \to 0$ as $n \to \infty$, the weights of the networks Φ_n cannot remain uniformly bounded. This phenomenon might explain numerical instabilites and exploding weights in practical optimization algorithms.

V. Failure of inverse stability of the realization map

For our final negative result, we study the *inverse stability* of the realization mapping $\mathrm{R}_\varrho^\Omega$ from Definition 2, which maps a family of neural network parameters to its realization. Even though this mapping turns out to be continuous (i.e. it is *forward stable*) from the finite dimensional parameter space to $L^p(\Omega)$ for any $p \in (0, \infty]$, it is *not* inverse stable. To be more precise, the following statement which is applicable to any of the activations given in Table I is true.

Theorem 6. *[1, Theorem 5.2.] Let $\varrho : \mathbb{R} \to \mathbb{R}$ be Lipschitz continuous, but not affine-linear. Moreover, let S be a neural network architecture with $L(S) \geq 2$ and $N_1 \geq 3$. Then there is a sequence $(\Phi_n)_n$ of neural networks with architecture S and*

(i) $\left\|\mathrm{R}_\varrho^\Omega(\Phi_n)\right\|_{\sup} \to 0$, as $n \to \infty$,

(ii) for any sequence $(\Psi_n)_n$ of neural networks with architecture S and $\mathrm{R}_\varrho^\Omega(\Phi_n) = \mathrm{R}_\varrho^\Omega(\Psi_n)$, the weights of $(\Psi_n)_n$ cannot remain uniformly bounded.

Rephrasing the statement of Theorem 6, we observe that it is *not* always possible for two realizations that are very close in the uniform norm to find corresponding neural networks whose weights have small distance.

The missing inverse stability implies yet another consequence which might explain a common phenomenon when optimizing the weights of a neural network. Considering a standard regression task in which the weights of a neural network are updated using a (stochastic) gradient descent, it might occur that at some point the underlying loss function applied to the *realization* of the neural networks returns a small error, although the associated *weights* are far away from these of the target function. This might lead to very slow convergence of the underlying optimization algorithm or, in extreme cases, to no convergence at all.

VI. Future Work

Despite the unfavorable structure of the set of functions generated by neural networks with fixed architecture, it is not immediately clear whether the same results remain valid if one considers special architectures such as those of convolutional neural networks. However, based on an equivalence between convolutional and fully-connected networks established in [13], we expect that the situation will not change significantly. Moreover, properties like closedness or inverse stability are heavily dependent on the norm which induces the underlying topology. We anticipate a higher chance to obtain closed sets of neural networks and inverse stable parametrizations, if we consider Sobolev-type norms, which, in the context of deep learning, have already been used in [14].

Acknowledgements

P.P. and M.R. were supported by the DFG Collaborative Research Center TRR 109 "Discretization in Geometry and Dynamics". P.P is supported by a DFG Research Fellowship "Shearlet-based energy functionals for anisotropic phase-field methods". M.R. is supported by the Berlin Mathematical School.

References

[1] P. Petersen, M. Raslan, and F. Voigtlaender, "Topological properties of the set of functions generated by neural networks of fixed size," *arXiv preprint arXiv:1806.08459*, 2018.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] W. McCulloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signal*, vol. 2, no. 4, pp. 303–314, 1989.

[5] D. Yarotsky, "Error bounds for approximations with deep ReLU networks," *Neural Netw.*, vol. 94, pp. 103–114, 2017.

[6] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen, "Optimal approximation with sparsely connected deep neural networks," *SIAM J. Math. Data Sci., to appear*, 2019.

[7] P. Petersen and F. Voigtlaender, "Optimal approximation of piecewise smooth functions using deep ReLU neural networks," *Neural Netw.*, vol. 108, pp. 296–330, 2018.

[8] I. Safran and O. Shamir, "Depth-width tradeoffs in approximating natural functions with neural networks," in *ICML*, vol. 70, 2017, pp. 2979–2987.

[9] L. Venturi, A. Bandeira, and J. Bruna, "Neural networks with finite intrinsic dimension have no spurious valleys," *arXiv preprint arXiv:1802.06384*, 2018.

[10] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bull. Am. Math. Soc.*, vol. 39, pp. 1–49, 2002.

[11] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 987–1000, 1998.

[12] W. E, J. Han, and A. Jentzen, "Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations," *Commun. Math. Stat.*, vol. 5, no. 4, pp. 349–380, 2017.

[13] P. Petersen and F. Voigtlaender, "Equivalence of approximation by convolutional neural networks and fully-connected networks," *arXiv preprint arXiv:1809.00973*, 2018.

[14] W. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, "Sobolev training for neural networks," in *NeurIPS*, 2017.