Best k-layer neural network approximations

Lek-Heng Lim, Mateusz Michalek, Yang Qi

Noname manuscript No.
(will be inserted by the editor)

Best k-layer neural network approximations

Lek-Heng Lim · Mateusz Michałek · Yang Qi

Received: December 12, 2019 / Accepted: November 10, 2020

Abstract We show that the empirical risk minimization (ERM) problem for neural networks has no solution in general. Given a training set $s_1, \dots, s_n \in \mathbb{R}^p$ with corresponding responses $t_1, \dots, t_n \in \mathbb{R}^q$, fitting a k-layer neural network $\nu_\theta : \mathbb{R}^p \to \mathbb{R}^q$ involves estimation of the weights $\theta \in \mathbb{R}^m$ via an ERM:

$$\inf_{\theta \in \mathbb{R}^m} \sum_{i=1}^{n} \|t_i - \nu_\theta(s_i)\|_2^2.$$

We show that even for $k = 2$, this infimum is not attainable in general for common activations like ReLU, hyperbolic tangent, and sigmoid functions. In addition, we deduce that if one attempts to minimize such a loss function in the event when its infimum is not attainable, it necessarily results in values of θ diverging to $\pm\infty$. We will show that for smooth activations $\sigma(x) = 1/\big(1 + \exp(-x)\big)$ and $\sigma(x) = \tanh(x)$, such failure to attain an infimum can happen on a positive-measured subset of responses. For the ReLU activation $\sigma(x) = \max(0, x)$, we completely classify cases where the ERM for a best two-layer neural network approximation attains its infimum. In recent applications of neural networks, where overfitting is commonplace, the failure to attain an infimum is avoided by ensuring that the system of equations $t_i = \nu_\theta(s_i)$, $i = 1, \dots, n$, has a solution. For a two-layer ReLU-activated network, we will

L.-H. Lim
Department of Statistics, University of Chicago, Chicago, IL 60637
E-mail: lekheng@uchicago.edu

M. Michałek
Max Planck Institute for Mathematics in the Sciences, 04103 Leipzig, Germany
University of Konstanz, D-78457 Konstanz, Germany
E-mail: Mateusz.michalek@uni-konstanz.de

Y. Qi
INRIA Saclay–Île-de-France, CMAP, École Polytechnique, IP Paris, CNRS, 91128 Palaiseau Cedex, France
E-mail: yang.qi@polytechnique.edu

show when such a system of equations has a solution generically, i.e., when can such a neural network be fitted perfectly with probability one.

Keywords neural network · best approximation · join loci · secant loci

Mathematics Subject Classification (2010) 92B20 · 41A50 · 41A30

1 Introduction

Let $\alpha_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$, $x \mapsto A_i x + b_i$ be an affine function with $A_i \in \mathbb{R}^{d_{i+1} \times d_i}$ and $b_i \in \mathbb{R}^{d_{i+1}}$, $i = 1, \dots, k$. Given any fixed *activation* function $\sigma : \mathbb{R} \to \mathbb{R}$, we will abuse notation slightly by also writing $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ for the function where σ is applied coordinatewise, i.e., $\sigma(x_1, \dots, x_d) = (\sigma(x_1), \dots, \sigma(x_d))$, for any $d \in \mathbb{N}$. Consider a k-layer neural network $\nu : \mathbb{R}^p \to \mathbb{R}^q$,

$$\nu = \alpha_k \circ \sigma \circ \alpha_{k-1} \circ \cdots \circ \sigma \circ \alpha_2 \circ \sigma \circ \alpha_1, \tag{1}$$

obtained from alternately composing σ with affine functions k times. Note that such a function ν is parameterized (and completely determined) by its *weights* $\theta := (A_k, b_k, \dots, A_1, b_1)$ in

$$\Theta := (\mathbb{R}^{d_{k+1} \times d_k} \times \mathbb{R}^{d_{k+1}}) \times \cdots \times (\mathbb{R}^{d_2 \times d_1} \times \mathbb{R}^{d_2}) \cong \mathbb{R}^m. \tag{2}$$

Here and throughout this article,

$$m := \sum_{i=1}^{k} (d_i + 1) d_{i+1} \tag{3}$$

will always denote the number of weights that parameterize ν. In neural networks lingo, the dimension of the ith layer d_i is also called the *number of neurons* in the ith layer. Whenever it is necessary to emphasize the dependence of ν on θ, we will write ν_θ for a k-layer neural network parameterized by $\theta \in \Theta$.

Consider the *function approximation problem* with[1] $d_{k+1} = 1$, i.e., given $\Omega \subseteq \mathbb{R}^{d_1}$ and a target function $f : \Omega \to \mathbb{R}$ in some Banach space $\mathcal{B} = L^p(\Omega)$, $W^{k,p}(\Omega)$, $\mathrm{BMO}(\Omega)$, etc, how well can f be approximated by a neural network $\nu_\theta : \Omega \to \mathbb{R}$ in the Banach space norm $\|\cdot\|_{\mathcal{B}}$? In other words, one is interested in the problem

$$\inf_{\theta \in \Theta} \|f - \nu_\theta\|_{\mathcal{B}}. \tag{4}$$

The most celebrated results along these lines are the universal approximation theorems of Cybenko [5], for sigmoidal activation and L^1-norm, as well as those of Hornik et al. [9,8], for more general activations such as ReLU and L^p-norms, $1 \le p \le \infty$. These results essentially say that the infimum in (4) is zero as long as k is at least two (but with no bound on d_2). In this article, for

[1] Results may be extended to $d_{k+1} > 1$ by applying them coordinatewise, i.e., with $\mathcal{B} \otimes \mathbb{R}^{d_{k+1}}$ in place of $\mathcal{B}$.

simplicity, we will focus on the L^2-norm. Henceforth we denote the dimensions of the first and last layers by

$$p \coloneqq d_1 \qquad \text{and} \qquad q \coloneqq d_{k+1}$$

to avoid the clutter of subscripts.

Traditional studies of neural networks in approximation theory [5,7,9,8] invariably assume that Ω, the domain of the target function f in the problem (4), is an *open subset* of $\mathbb{R}^p$. Nevertheless, any actual training of a neural network involves only finitely many points $s_1, \dots, s_n \in \Omega$ and the values of f on these points: $f(s_1) = t_1, \dots, f(s_n) = t_n$. Therefore, in reality, one only solves problem (4) for a *finite* $\Omega = \{s_1, \dots, s_n\}$ and this becomes a parameter estimation problem commonly called the *empirical risk minimization problem*. Let $s_1, \dots, s_n \in \Omega \subseteq \mathbb{R}^p$ be a sample of n independent, identically distributed observations with corresponding *responses* $t_1, \dots, t_n \in \mathbb{R}^q$. The main computational problem in supervised learning with neural networks is to fit the *training set* $\{(s_i, t_i) \in \mathbb{R}^p \times \mathbb{R}^q : i = 1, \dots, n\}$ with a k-layer neural network $\nu_\theta : \Omega \to \mathbb{R}^q$ so that

$$t_i \approx \nu_\theta(s_i), \quad i = 1, \dots, n,$$

often in the least-squares sense

$$\inf_{\theta \in \Theta} \sum_{i=1}^{n} \|t_i - \nu_\theta(s_i)\|_2^2. \tag{5}$$

The responses are regarded as values of the unknown function $f : \Omega \to \mathbb{R}^q$ to be learned, i.e., $t_i = f(s_i)$, $i = 1, \dots, n$. The hope is that by solving (5) for $\theta^* \in \mathbb{R}^m$, the neural network obtained ν_{θ^*} will approximate f well in the sense of having small generalization errors, i.e., $f(s) \approx \nu_{\theta^*}(s)$ for $s \notin \{s_1, \dots, s_n\}$. This hope has been borne out empirically in spectacular ways [10,12,16].

Observe that the empirical risk estimation problem (5) is simply the function approximation problem (4) for the case when Ω is a finite set equipped with the counting measure. The problem (4) asks how well a given target function can be approximated by a given function class, in this case the class of k-layer σ-activated neural networks. This is an infinite-dimensional problem when Ω is infinite and is usually studied using functional analytic techniques. On the other hand (5) asks how well the approximation is at finitely many sample points, a finite-dimensional problem, and is therefore amenable to techniques in algebraic and differential geometry. We would like to emphasize that the finite-dimensional problem (5) is not any easier than the infinite-dimensional problem (4); they simply require different techniques. In particular, our results do not follow from the results in [3,7,14] for infinite-dimensional spaces — we will have more to say about this in Section 2.

There is a parallel with [15], where we applied methods from algebraic and differential geometry to study the empirical risk minimization problem corresponding to *nonlinear approximation*, i.e., where one seeks to approximate a target function by a sum of k atoms $\varphi_1, \dots, \varphi_k$ from a dictionary D,

$$\inf_{\varphi_i \in D} \|f - \varphi_1 - \varphi_2 - \cdots - \varphi_k\|_{\mathcal{B}}.$$

If we denote the layers of a neural network by $\varphi_i \in L$, then (4) may be written in a form that parallels the above:

$$\inf_{\varphi_i \in L} \|f - \varphi_k \circ \varphi_{k-1} \circ \cdots \circ \varphi_1\|_{\mathcal{B}}.$$

Again our goal is to study the corresponding empirical risk minimization problem, i.e., the approximation problem (5). The first surprise is that this may not always have a solution. For example, take $n = 6$ and $p = q = 2$ with

$$s_1 = \begin{bmatrix}-2\\0\end{bmatrix}, s_2 = \begin{bmatrix}-1\\0\end{bmatrix}, s_3 = \begin{bmatrix}0\\0\end{bmatrix}, s_4 = \begin{bmatrix}1\\0\end{bmatrix}, \quad s_5 = \begin{bmatrix}2\\0\end{bmatrix}, \quad s_6 = \begin{bmatrix}1\\1\end{bmatrix},$$
$$t_1 = \begin{bmatrix}2\\0\end{bmatrix}, \quad t_2 = \begin{bmatrix}1\\0\end{bmatrix}, \quad t_3 = \begin{bmatrix}0\\0\end{bmatrix}, t_4 = \begin{bmatrix}-2\\0\end{bmatrix}, t_5 = \begin{bmatrix}-4\\0\end{bmatrix}, t_6 = \begin{bmatrix}0\\1\end{bmatrix}.$$

For a ReLU-activated two-layer neural network, the approximation problem (5) seeks weights $\theta = (A, b, B, c)$ that attain the infimum over all $A, B \in \mathbb{R}^{2\times 2}$, $b, c \in \mathbb{R}^2$ of the loss function

$$\begin{aligned}
&\left\|\begin{bmatrix}2\\0\end{bmatrix} - \left[B\max\left(A\begin{bmatrix}-2\\0\end{bmatrix}+b, \begin{bmatrix}0\\0\end{bmatrix}\right)+c\right]\right\|^2 + \left\|\begin{bmatrix}1\\0\end{bmatrix} - \left[B\max\left(A\begin{bmatrix}-1\\0\end{bmatrix}+b, \begin{bmatrix}0\\0\end{bmatrix}\right)+c\right]\right\|^2\\
&\quad + \left\|\begin{bmatrix}0\\0\end{bmatrix} - \left[B\max\left(A\begin{bmatrix}0\\0\end{bmatrix}+b, \begin{bmatrix}0\\0\end{bmatrix}\right)+c\right]\right\|^2 + \left\|\begin{bmatrix}-2\\0\end{bmatrix} - \left[B\max\left(A\begin{bmatrix}1\\0\end{bmatrix}+b, \begin{bmatrix}0\\0\end{bmatrix}\right)+c\right]\right\|^2\\
&\quad + \left\|\begin{bmatrix}-4\\0\end{bmatrix} - \left[B\max\left(A\begin{bmatrix}2\\0\end{bmatrix}+b, \begin{bmatrix}0\\0\end{bmatrix}\right)+c\right]\right\|^2 + \left\|\begin{bmatrix}0\\1\end{bmatrix} - \left[B\max\left(A\begin{bmatrix}1\\1\end{bmatrix}+b, \begin{bmatrix}0\\0\end{bmatrix}\right)+c\right]\right\|^2.
\end{aligned}$$

We will see in the proof of Theorem 1 that this has no solution. Any sequence of $\theta = (A, b, B, c)$ chosen so that the loss function converges to its infimum will have $\|\theta\|^2 = \|A\|_F^2 + \|b\|_2^2 + \|B\|_F^2 + \|c\|_2^2$ becoming unbounded — the entries of θ will diverge to $\pm\infty$ in such a way that keeps the loss function bounded and in fact convergent to its infimum.[2]

With a smooth activation like hyperbolic tangent in place of ReLU, we can establish a stronger nonexistence result: In Theorem 5, we show that there is a positive-measured set $U \subseteq (\mathbb{R}^q)^n$ such that for any target values $(t_1, \ldots, t_n) \in U$, the empirical risk estimation problem (5) has no solution.

Whenever one attempts to minimize a function that does not have a minimum (i.e., infimum is not attained), one runs into serious numerical issues. We establish this formally in Proposition 1, showing that the parameters θ must necessarily diverge to $\pm\infty$ when one attempts to minimize (5) in the event when its infimum is not attainable.

Our study here may thus shed some light on a key feature of modern deep neural networks, made possible by the abundance of computing power not available to early adopters like the authors of [5,7,9,8]. *Deep* neural networks are, almost by definition and certainly in practice, heavily overparameterized.

[2] We assume the Euclidean norm $\|\theta\|^2 := \sum_{i=1}^k \|A_i\|_F^2 + \|b_i\|_2^2$ on our parameter space Θ but results in this article are independent of the choice of norms as all norms are equivalent on finite-dimensional spaces, another departure from the infinite-dimensional case in [7,14].

In this case, whatever training data $(s_1, t_1), \dots, (s_n, t_n)$ may be perfectly fitted and in essence one solves the system of *neural network equations*:

$$t_i = \nu_\theta(s_i), \quad i = 1, \dots, n, \tag{6}$$

for a solution $\theta \in \Theta$ and thereby circumvents the issue of whether the infimum in (5) can be attained. This in our view is the reason ill-posedness issues did not prevent the recent success of neural networks. For a two-layer ReLU-activated neural network, we will address the question of whether (6) has a solution generically in Section 5.

A word about our use of the term "ill-posedness" is in order: Recall that a problem is said to be *ill-posed* if a solution either (i) does not exist, (ii) exists but is not unqiue, or (iii) exists and is unique but does not depend continuously on the input parameters of the problem. When we claimed that the problem (5) is ill-posed, it is in the sense of (i), clearly the most serious issue of the three — (ii) and (iii) may be ameliorated with regularization or other strategies but not (i). We use "ill-posedness" in the sense of (i) throughout our article. The work in [13], for example, is about ill-posedness in the sense of (ii).

2 Geometry of empirical risk minimization for neural networks

Given that we are interested in the behavior of ν_θ as a function of weights θ, we will rephrase (5) to put it on more relevant footing. Let the sample $s_1, \dots, s_n \in \mathbb{R}^p$ and responses $t_1, \dots, t_n \in \mathbb{R}^q$ be arbitrary but fixed. Henceforth we will assemble the sample into a *design matrix*,

$$S \coloneqq \begin{bmatrix} s_1^\intercal \\ s_2^\intercal \\ \vdots \\ s_n^\intercal \end{bmatrix} \in \mathbb{R}^{n \times p}, \tag{7}$$

and the corresponding responses into a *response matrix*,

$$T \coloneqq \begin{bmatrix} t_1^\intercal \\ t_2^\intercal \\ \vdots \\ t_n^\intercal \end{bmatrix} \in \mathbb{R}^{n \times q}. \tag{8}$$

Here and for the rest of this article, we use the following numerical linear algebra conventions:

- a vector $a \in \mathbb{R}^n$ will always be regarded as a column vector;
- a row vector will always be denoted $a^\intercal$ for some column vector a;
- a matrix $A \in \mathbb{R}^{n \times p}$ may be denoted either
  - as a list of its column vectors $A = [a_1, \dots, a_p]$, i.e., $a_1, \dots, a_p \in \mathbb{R}^n$ are columns of A;

– or as a list of its row vectors $A = [\alpha_1^\intercal, \dots, \alpha_n^\intercal]^\intercal$, i.e., $\alpha_1, \dots, \alpha_n \in \mathbb{R}^p$ are rows of A.

We will also adopt the convention that treats a *direct sum* of p subspaces (resp. cones) in $\mathbb{R}^n$ as a subspace (resp. cone) of $\mathbb{R}^{n\times p}$: If $V_1, \dots, V_p \subseteq \mathbb{R}^n$ are subspaces (resp. cones), then

$$V_1 \oplus \cdots \oplus V_p := \{[v_1, \dots, v_p] \in \mathbb{R}^{n\times p} : v_1 \in V_1, \dots, v_p \in V_p\}. \tag{9}$$

Let $\nu_\theta : \mathbb{R}^p \to \mathbb{R}^q$ be a k-layer neural network, $k \geq 2$. We define the *weights map* $\psi_k : \Theta \to \mathbb{R}^{n\times q}$ by

$$\psi_k(\theta) = \begin{bmatrix} \nu_\theta(s_1)^\intercal \\ \nu_\theta(s_2)^\intercal \\ \vdots \\ \nu_\theta(s_n)^\intercal \end{bmatrix} \in \mathbb{R}^{n\times q}.$$

In other words, for a fixed sample, ψ_k is ν_θ regarded as a function of the weights θ. The empirical risk minimization problem is (5) rewritten as

$$\inf_{\theta\in\Theta} \|T - \psi_k(\theta)\|_F, \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. We may view (10) as a matrix approximation problem — finding a matrix in

$$\psi_k(\Theta) = \{\psi_k(\theta) \in \mathbb{R}^{n\times q} : \theta \in \Theta\} \tag{11}$$

that is nearest to a given matrix $T \in \mathbb{R}^{n\times q}$.

Definition 1 We will call the set $\psi_k(\Theta)$ the *image of weights* of the k-layer neural network ν_θ and the corresponding problem (10) a *best k-layer neural network approximation problem.*

As we noted in (2), the space of all weights Θ is essentially the Euclidean space $\mathbb{R}^m$ and uninteresting geometrically, but the image of weights $\psi_k(\Theta)$, as we will see in this article, has complicated geometry (e.g., for $k = 2$ and ReLU activation, it is the join locus of a line and the secant locus of a cone — see Theorem 2). In fact, the geometry of the neural network *is* the geometry of $\psi_k(\Theta)$. We expect that it will be pertinent to understand this geometry if one wants to understand neural networks at a deeper level. For one, the nontrivial geometry of $\psi_k(\Theta)$ is the reason that the best k-layer neural network approximation problem, which is to find a point in $\psi_k(\Theta)$ closest to a given $T \in \mathbb{R}^{n\times q}$, lacks a solution in general.

Indeed, the most immediate mathematical issues with the approximation problem (10) are the existence and uniqueness of solutions:

(i) a nearest point may not exist since the set $\psi_k(\Theta)$ may not be a closed subset of $\mathbb{R}^{n\times q}$, i.e., the infimum in (10) may not be attainable;
(ii) even if it exists, the nearest point may not be unique, i.e., the infimum in (10) may be attained by two or more points in $\psi_k(\Theta)$.

As a reminder a problem is said to be *ill-posed* if it lacks existence and uniqueness guarantees. Ill-posedness creates numerous difficulties both logical (what does it mean to find a solution when it does not exist?) and practical (which solution do we find when there are more than one?). In addition, a well-posed problem near an ill-posed one is the very definition of an *ill-conditioned* problem [4], which presents its own set of difficulties. In general, ill-posed problems are not only to be avoided but also delineated to reveal the region of ill-conditioning.

For the function approximation problem (4) with Ω an open subset of $\mathbb{R}^p$, the nonexistence issue of a best neural network approximant is very well-known, dating back to [7], with a series of follow-up works, e.g., [3,14]. But for the case that actually occurs in the training of a neural network, i.e., where Ω is a finite set, (4) becomes (5) or (10) and its well-posedness has never been studied, to the best of our knowledge. Our article seeks to address this gap. The geometry of the set in (11) will play an important role in studying these problems, much like the role played by the geometry of rank-k tensors in [15].

We will show that for many networks, the problem (10) is ill-posed. We have already mentioned an explicit example at the end of Section 1 for the ReLU activation:

$$\sigma_{\max}(x) \coloneqq \max(0, x) \tag{12}$$

where (10) lacks a solution; we will discuss this in detail in Section 4. Perhaps more surprisingly, for the sigmoidal and hyperbolic tangent activation functions:

$$\sigma_{\exp}(x) \coloneqq \frac{1}{1+\exp(-x)} \quad \text{and} \quad \sigma_{\tanh}(x) \coloneqq \tanh(x),$$

we will see in Section 6 that (10) lacks a solution with positive probability, i.e., there exists an open set $U \subseteq \mathbb{R}^{n\times q}$ such that for any $T \in U$ there is no nearest point in $\psi_k(\Theta)$. Similar phenomenon is known for real tensors [6, Section 8].

For neural networks with ReLU activation, we are unable to establish similar "failure with positive probability" results but the geometry of the problem (10) is actually simpler in this case. For two-layer ReLU-activated network, we can completely characterize the geometry of $\psi_2(\Theta)$, which provides us with greater insights as to why (10) generally lacks a solution. We can also determine the dimension of $\psi_2(\Theta)$ in many instances. These will be discussed in Section 5.

The following map will play a key role in this article and we give it a name to facilitate exposition.

Definition 2 (ReLU projection) The map $\sigma_{\max} : \mathbb{R}^d \to \mathbb{R}^d$ where the ReLU activation (12) is applied coordinatewise will be called a *ReLU projection*. For any $\Omega \subseteq \mathbb{R}^d$, $\sigma_{\max}(\Omega) \subseteq \mathbb{R}^d$ will be called a ReLU projection of Ω.

Note that a ReLU projection is a linear projection when restricted to any orthant of $\mathbb{R}^d$.

3 Geometry of a "one-layer neural network"

We start by studying the 'first part' of a two-layer ReLU-activated neural network:

$$\mathbb{R}^p \xrightarrow{\alpha} \mathbb{R}^q \xrightarrow{\sigma_{\max}} \mathbb{R}^q$$

and, slightly abusing terminologies, call this a *one-layer ReLU-activated neural network*. Note that the weights here are $\theta = (A, b) \in \mathbb{R}^{q\times(p+1)} = \Theta$ with $A \in \mathbb{R}^{q\times p}$, $b \in \mathbb{R}^q$ that define the affine map $\alpha(x) = Ax + b$.

Let the sample $S = [s_1^\intercal, \dots, s_n^\intercal]^\intercal \in \mathbb{R}^{n\times p}$ be fixed. Define the weight map $\psi_1 : \Theta \to \mathbb{R}^{n\times q}$ by

$$\psi_1(A, b) = \sigma_{\max}\left(\begin{bmatrix} As_1 + b \\ \vdots \\ As_n + b \end{bmatrix}\right), \tag{13}$$

where $\sigma_{\max}$ is applied coordinatewise.

Recall that a *cone* $C \subseteq \mathbb{R}^d$ is simply a set invariant under scaling by positive scalars, i.e., if $x \in C$, then $\lambda x \in C$ for all $\lambda > 0$. Note that we do not assume that a cone has to be convex; in particular, in our article the dimension of a cone C refers to its dimension as a semialgebraic set.

Definition 3 (ReLU cone) Let $S = [s_1^\intercal, \dots, s_n^\intercal]^\intercal \in \mathbb{R}^{n\times p}$. The *ReLU cone* of S is the set

$$\mathrm{C}_{\max}(S) \coloneqq \left\{ \sigma_{\max}\left(\begin{bmatrix} a^\intercal s_1 + b \\ \vdots \\ a^\intercal s_n + b \end{bmatrix}\right) \in \mathbb{R}^n : a \in \mathbb{R}^p,\ b \in \mathbb{R} \right\}.$$

The ReLU cone is clearly a cone. Such cones will form the building blocks for the image of weights $\psi_k(\Theta)$. In fact, it is easy to see that $\mathrm{C}_{\max}(S)$ is exactly $\psi_1(\Theta)$ in case when $q = 1$. The next lemma describes the geometry of ReLU cones in greater detail.

Lemma 1 *Let $S = [s_1^\intercal, \dots, s_n^\intercal]^\intercal \in \mathbb{R}^{n\times p}$.*

(i) *The set $\mathrm{C}_{\max}(S)$ is always a closed pointed cone of dimension*

$$\dim \mathrm{C}_{\max}(S) = \operatorname{rank}[S, \mathbb{1}]. \tag{14}$$

Here $[S, \mathbb{1}] \in \mathbb{R}^{n\times(p+1)}$ is augmented with an extra column $\mathbb{1} \coloneqq [1, \dots, 1]^\intercal \in \mathbb{R}^n$, the vector of all ones.

(ii) *A set $C \subseteq \mathbb{R}^n$ is a ReLU cone if and only if it is a ReLU projection of some linear subspace in $\mathbb{R}^n$ containing the vector $\mathbb{1}$.*

Proof Consider the map in (13) with $q = 1$. Then $\psi_1 : \Theta \to \mathbb{R}^n$ is given by a composition of the linear map

$$\Theta \to \mathbb{R}^n, \quad \begin{bmatrix} a \\ b \end{bmatrix} \mapsto \begin{bmatrix} s_1^\intercal a + b \\ \vdots \\ s_n^\intercal a + b \end{bmatrix} = [S, \mathbb{1}] \begin{bmatrix} a \\ b \end{bmatrix},$$

whose image is a linear space L of dimension $\operatorname{rank}[S, \mathbb{1}]$ containing $\mathbb{1}$, and the ReLU projection $\sigma_{\max} : \mathbb{R}^n \to \mathbb{R}^n$. Since $\mathrm{C}_{\max}(S) = \psi_1(\Theta)$, it is a ReLU projection of a linear subspace in $\mathbb{R}^n$, which is clearly a closed pointed cone.

For its dimension, note that a ReLU projection is a linear projection on each quadrant and thus cannot increase dimension. On the other hand, since $\mathbb{1} \in L$, we know that L intersects the interior of the nonnegative quadrant on which $\sigma_{\max}$ is the identity; thus $\sigma_{\max}$ preserves dimension and we have (14).

Conversely, a ReLU projection of any linear space L may be realized as $\mathrm{C}_{\max}(S)$ for some choice of S — just choose S so that the image of the matrix $[S, \mathbb{1}]$ is L. □

It follows from Lemma 1 that the image of weights of a one-layer ReLU neural network has the geometry of a direct sum of q closed pointed cones. Recall our convention for direct sum in (9).

Corollary 1 *Consider the one-layer ReLU-activated neural network*

$$\mathbb{R}^p \xrightarrow{\alpha_1} \mathbb{R}^q \xrightarrow{\sigma_{\max}} \mathbb{R}^q.$$

Let $S \in \mathbb{R}^{n\times p}$. Then $\psi_1(\Theta) \subseteq \mathbb{R}^{n\times q}$ has the structure of a direct sum of q copies of $\mathrm{C}_{\max}(S) \subseteq \mathbb{R}^n$. More precisely,

$$\psi_1(\Theta) = \{[v_1, \dots, v_q] \in \mathbb{R}^{n\times q} : v_1, \dots, v_q \in \mathrm{C}_{\max}(S)\}. \tag{15}$$

In particular, $\psi_1(\Theta)$ is a closed pointed cone of dimension $q \cdot \operatorname{rank}[S, \mathbb{1}]$ in $\mathbb{R}^{n\times q}$.

Proof Each row of the matrix α_1 can be identified with the affine map defined in Lemma 1. Then the conclusion follows by Lemma 1. □

Given Corollary 1, one might perhaps think that a two-layer neural network

$$\mathbb{R}^{d_1} \xrightarrow{\alpha_1} \mathbb{R}^{d_2} \xrightarrow{\sigma_{\max}} \mathbb{R}^{d_2} \xrightarrow{\alpha_2} \mathbb{R}^{d_3}$$

would also have a closed image of weights $\psi_2(\Theta)$. This turns out to be false. We will show that the image $\psi_2(\Theta)$ may not be closed.

As a side remark, note that Definition 3 and Lemma 1 are peculiar to the ReLU activation. For smooth activations like $\sigma_{\exp}$ and $\sigma_{\tanh}$, the image of weights $\psi_k(\Theta)$ is almost never a cone and Lemma 1 does not hold in multiple ways.

4 Ill-posedness of best k-layer neural network approximation

The $k = 2$ case is the simplest and yet already nontrivial in that it has the universal approximation property, as we mentioned earlier. The main content of the next theorem is in its proof, which is constructive and furnishes an explicit example of a function that does not have a best approximation by a two-layer neural network. The reader is reminded that even training a two-layer neural network is already an NP-hard problem [1,2].

Theorem 1 (Ill-posedness of neural network approximation I) *The best two-layer ReLU-activated neural network approximation problem*

$$\inf_{\theta\in\Theta} \|T - \psi_2(\theta)\|_F \tag{16}$$

is ill-posed, i.e., the infimum in (16) cannot be attained in general.

Proof We will construct an explicit two-layer ReLU-activated network whose image of weights is not closed. Let $d_1 = d_2 = d_3 = 2$. For the two-layer ReLU-activated network

$$\mathbb{R}^2 \xrightarrow{\alpha_1} \mathbb{R}^2 \xrightarrow{\sigma_{\max}} \mathbb{R}^2 \xrightarrow{\alpha_2} \mathbb{R}^2,$$

the weights take the form

$$\theta = (A_1, b_1, A_2, b_2) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^2 \times \mathbb{R}^{2\times 2} \times \mathbb{R}^2 = \Theta \cong \mathbb{R}^{12},$$

i.e., $m = 12$. Consider a sample S of size $n = 6$ given by

$$s_i = \begin{bmatrix} i-3 \\ 0 \end{bmatrix}, \quad i = 1, \dots, 5, \quad s_6 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

that is

$$S = \begin{bmatrix} -2 & 0 \\ -1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ 1 & 1 \end{bmatrix} \in \mathbb{R}^{6\times 2}.$$

Thus the weight map $\psi_2 : \Theta \to \mathbb{R}^{6\times 2}$, or, more precisely,

$$\psi_2 : \mathbb{R}^{2\times 2} \times \mathbb{R}^2 \times \mathbb{R}^{2\times 2} \times \mathbb{R}^2 \to \mathbb{R}^{6\times 2},$$

is given by

$$\psi_2(\theta) = \begin{bmatrix} \nu_\theta(s_1)^\intercal \\ \vdots \\ \nu_\theta(s_6)^\intercal \end{bmatrix} = \begin{bmatrix} \big(A_2 \max(A_1 s_1 + b_1, 0) + b_2\big)^\intercal \\ \vdots \\ \big(A_2 \max(A_1 s_6 + b_1, 0) + b_2\big)^\intercal \end{bmatrix} \in \mathbb{R}^{6\times 2}.$$

We claim that the image of weights $\psi_2(\Theta)$ is not closed — the point

$$T = \begin{bmatrix} 2 & 0 \\ 1 & 0 \\ 0 & 0 \\ -2 & 0 \\ -4 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{6\times 2} \tag{17}$$

is in the closure of $\psi_2(\Theta)$ but not in $\psi_2(\Theta)$. Therefore for this choice of T, the infimum in (16) is zero but is never attainable by any point in $\psi_2(\Theta)$.

We will first prove that T is in the closure of $\psi_2(\Theta)$: Consider a sequence of affine transformations $\alpha_1^{(k)}$, $k = 1, 2, \ldots$, defined by

$$\alpha_1^{(k)}(s_i) = (i-3)\begin{bmatrix}-1\\1\end{bmatrix}, \quad i = 1, \ldots, 5, \quad \alpha_1^{(k)}(s_6) = \begin{bmatrix}2k\\k\end{bmatrix},$$

and set

$$\alpha_2^{(k)}\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \begin{bmatrix}x-2y\\\frac{1}{k}y\end{bmatrix}.$$

The sequence of two-layer neural networks,

$$\nu^{(k)} = \alpha_2^{(k)} \circ \sigma_{\max} \circ \alpha_1^{(k)}, \quad k \in \mathbb{N},$$

have weights given by

$$\theta_k = \left(\begin{bmatrix}-1 & 2k+1\\1 & k-1\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & -2\\0 & \frac{1}{k}\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}\right) \tag{18}$$

and that

$$\lim_{k\to\infty} \psi_2(\theta_k) = T.$$

This shows that T in (17) is indeed in the closure of $\psi_2(\Theta)$.

We next show by contradiction that $T \notin \psi_2(\Theta)$. Write $T = [t_1^\intercal, \ldots, t_6^\intercal]^\intercal \in \mathbb{R}^{6\times 2}$ where $t_1, \ldots, t_6 \in \mathbb{R}^2$ are as in (17). Suppose $T \in \psi_2(\Theta)$. Then there exist some affine maps $\beta_1, \beta_2 : \mathbb{R}^2 \to \mathbb{R}^2$ such that

$$t_i = \beta_2 \circ \sigma_{\max} \circ \beta_1(s_i), \quad i = 1, \ldots, 6.$$

As t_1, t_3, t_6 are affinely independent, β_2 has to be an affine isomorphism. Hence the five points

$$\sigma_{\max}(\beta_1(s_1)), \ldots, \sigma_{\max}(\beta_1(s_5)) \tag{19}$$

have to lie on a line in $\mathbb{R}^2$. Also, note that

$$\beta_1(s_1), \ldots, \beta_1(s_5) \tag{20}$$

lie on a (different) line in $\mathbb{R}^2$ since β_1 is an affine homomorphism. The five points in (20) have to be in the same quadrant, otherwise the points in (19) could not lie on a line or have successive distances $\delta, \delta, 2\delta, 2\delta$ for some $\delta > 0$. Note that $\sigma_{\max} : \mathbb{R}^2 \to \mathbb{R}^2$ is identity in the first quadrant, projection to the y-axis in the second, projection to the point $(0, 0)$ in the third, and projection to the x-axis in the fourth. So $\sigma_{\max}$ takes points with equal successive distances in the first, second, fourth quadrant to points with equal successive distances in the same quadrant; and it takes all points in the third quadrant to the origin. Hence $\sigma_{\max}$ cannot take the five colinear points in (20), with equal successive distances, to the five colinear points in (19), with successive distances $\delta, \delta, 2\delta, 2\delta$. This yields the required contradiction. □

We would like to emphasize that the example constructed in the proof of Theorem 1, which is about the nonclosedness of the image of weights within a finite-dimensional space of response matrices, differs from examples in [7,14], which are about the nonclosedness of the class of neural network within an infinite-dimensional space of target functions. Another difference is that here we have considered the ReLU activation $\sigma_{\max}$ as opposed to the hyperbolic tangent activation $\sigma_{\exp}$ in [7,14]. The 'neural networks' studied in [3] differ from standard usage [5,7,9,8,14] in the sense of (1); they are defined as a linear combination of neural networks whose weights are *fixed in advanced* and approximation is in the sense of finding the coefficients of such linear combinations. As such the results in [3] are irrelevant to our discussions.

As we pointed out at the end of Section 1 and as the reader might also have observed in the proof of Theorem 1, the sequence of weights θ_j in (18) contain entries that become unbounded as $j \to \infty$. This is not peculiar to the sequence we chose in (18); by the same discussion in [6, Section 4.3], this will always be the case:

Proposition 1 *If the infimum in* (10) *is not attainable, then any sequence of weights* $\theta_j \in \Theta$ *with*

$$\lim_{j\to\infty} \|T - \psi_k(\theta_j)\|_F = \inf_{\theta\in\Theta} \|T - \psi_k(\theta)\|_F$$

must be unbounded, i.e.,

$$\limsup_{j\to\infty} \|\theta_j\| = \infty.$$

This holds regardless of the choice of activation and number of layers.

The implication of Proposition 1 is that if one attempts to forcibly fit a neural network to target values T where (16) does not have a solution, then it simply results in the parameters θ blowing up to $\pm\infty$.

5 Generic solvability of the neural network equations

As we mentioned at the end of Section 1, modern deep neural networks avoids the issue that (10) may lack an infimum by overfitting. In which case the relevant problem is no longer one of approximation but becomes one of solving a system of what we called neural network equations (6), rewritten here as

$$\psi_k(\theta) = T. \tag{21}$$

Whether (21) has a solution is not determined by $\dim \Theta$ but by $\dim \psi_k(\Theta)$. If the dimension of the image of weights equals nq, i.e., the dimension of the ambient space $\mathbb{R}^{n\times q}$ in which T lies, then (21) has a solution generically. In this section, we will provide various expressions for $\dim \psi_k(\Theta)$ for $k = 2$ and the ReLU activation.

There is a deeper geometrical explanation behind Theorem 1. Let $d \in \mathbb{N}$. The *join locus* of $X_1, \dots, X_r \subseteq \mathbb{R}^d$ is the set

$$\operatorname{Join}(X_1, \dots, X_r) \coloneqq \{\lambda_1 x_1 + \cdots + \lambda_r x_r \in \mathbb{R}^d : x_i \in X_i,\ \lambda_i \in \mathbb{R},\ i = 1, \dots, r\}. \quad (22)$$

A special case is when $X_1 = \cdots = X_r = X$ and in which case the join locus is called the rth *secant locus*

$$\Sigma_r^\circ(X) = \{\lambda_1 x_1 + \cdots + \lambda_r x_r \in \mathbb{R}^d : x_i \in X,\ \lambda_i \in \mathbb{R},\ i = 1, \dots, r\}.$$

An example of a join locus is the set of "sparse-plus-low-rank" matrices [15, Section 8.1]; an example of a rth secant locus is the set of rank-r tensors [15, Section 7].

From a geometrical perspective, we will next show that for $k = 2$ and $q = 1$ the set $\psi_2(\Theta)$ has the structure of a join locus. Join loci are known in general to be nonclosed [18]. For this reason, the ill-posedness of the best k-layer neural network problem is not unlike that of the best rank-r approximation problem for tensors, which is a consequence of the nonclosedness of the secant loci of the Segre variety [6].

We shall begin with the case of a two-layer neural network with one-dimensional output, i.e., $q = 1$. In this case, $\psi_2(\Theta) \subseteq \mathbb{R}^n$ and we can describe its geometry very precisely. The more general case where $q > 1$ will be in Theorem 4.

Theorem 2 (Geometry of two-layer neural network I) *Consider the two-layer ReLU-activated network with p-dimensional inputs and d neurons in the hidden layer:*

$$\mathbb{R}^p \xrightarrow{\alpha_1} \mathbb{R}^d \xrightarrow{\sigma_{\max}} \mathbb{R}^d \xrightarrow{\alpha_2} \mathbb{R}. \quad (23)$$

The image of weights is given by

$$\psi_2(\Theta) = \operatorname{Join}\big(\Sigma_d^\circ(\mathrm{C}_{\max}(S)), \operatorname{span}\{\mathbb{1}\}\big)$$

where

$$\Sigma_d^\circ\big(\mathrm{C}_{\max}(S)\big) \coloneqq \bigcup_{y_1, \dots, y_d \in \mathrm{C}_{\max}(S)} \operatorname{span}\{y_1, \dots, y_d\} \subseteq \mathbb{R}^n$$

is the dth secant locus of the ReLU cone and $\operatorname{span}\{\mathbb{1}\}$ *is the one-dimensional subspace spanned by $\mathbb{1} \in \mathbb{R}^n$.*

Proof Let $\alpha_1(z) = A_1 z + b$, where

$$A_1 = \begin{bmatrix} a_1^\intercal \\ \vdots \\ a_d^\intercal \end{bmatrix} \in \mathbb{R}^{d \times p} \quad \text{and} \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_d \end{bmatrix} \in \mathbb{R}^d.$$

Let $\alpha_2(z) = A_2^\intercal z + \lambda$, where $A_2^\intercal = (c_1, \dots, c_d) \in \mathbb{R}^d$. Then $x = (x_1, \dots, x_n)^\intercal \in \psi_2(\Theta)$ if and only if

$$\begin{cases} x_1 = c_1\sigma(s_1^\intercal a_1 + b_1) + \cdots + c_d\sigma(s_1^\intercal a_d + b_d) + \lambda, \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ x_n = c_1\sigma(s_n^\intercal a_1 + b_1) + \cdots + c_d\sigma(s_n^\intercal a_d + b_d) + \lambda. \end{cases} \tag{24}$$

For $i = 1, \dots, d$, define the vector $y_i = (\sigma(s_1^\intercal a_i + b_i), \dots, \sigma(s_n^\intercal a_i + b_i))^\intercal \in \mathbb{R}^n$, which belongs to $\mathrm{C}_{\max}(S)$ by Corollary 1. Thus, (24) is equivalent to

$$x = c_1 y_1 + \cdots + c_d y_d + \lambda \mathbb{1} \tag{25}$$

for some $c_1, \dots, c_d \in \mathbb{R}$. By definition of secant locus, (25) is equivalent to the statement

$$x \in \Sigma_d^\circ\big(\mathrm{C}_{\max}(S)\big) + \lambda \mathbb{1},$$

which completes the proof. □

From a practical point of view, we are most interested in basic *topological* issues like whether the image of weights $\psi_k(\Theta)$ is closed or not, since this affects the solvability of (16). However, the *geometrical* description of $\psi_2(\Theta)$ in Theorem 2 will allow us to deduce bounds on its dimension. Note that the dimension of the space of weights Θ as in (2) is just m as in (14) but this is not the true dimension of the neural network, which should instead be that of the image of weights $\psi_k(\Theta)$.

In general, even for $k = 2$, it will be difficult to obtain the exact dimension of $\psi_2(\Theta)$ for an arbitrary two-layer network (23). In the next corollary, we deduce from Theorem 2 an upper bound dependent on the sample $S \in \mathbb{R}^{n\times p}$ and another independent of it.

Corollary 2 *For the two-layer ReLU-activated network* (23)*, we have*

$$\dim \psi_2(\Theta) \le d(\operatorname{rank}[S, \mathbb{1}]) + 1,$$

and in particular,

$$\dim \psi_2(\Theta) \le \min\big(d(\min(p, n) + 1) + 1, pn\big).$$

When n is sufficiently large and the observations $s_1, \dots, s_n$ are sufficiently general,[3] we may deduce a more precise value of $\dim \psi_2(\Theta)$. Before describing our results, we introduce several notations. For any index set $I \subseteq \{1, \dots, n\}$ and sample $S \in \mathbb{R}^{n\times p}$, we write

$$\begin{aligned} \mathbb{R}^n_I &:= \{x \in \mathbb{R}^n : x_i = 0 \text{ if } i \notin I \text{ and } x_j > 0 \text{ if } j \in I\}, \\ F_I(S) &:= \mathrm{C}_{\max}(S) \cap \mathbb{R}^n_I. \end{aligned}$$

Note that $F_\varnothing(S) = \{0\}$, $\mathbb{1} \in F_{\{1,\dots,n\}}(S)$, and $\mathrm{C}_{\max}(S)$ may be expressed as

$$\mathrm{C}_{\max}(S) = F_{I_1}(S) \cup \cdots \cup F_{I_\ell}(S), \tag{26}$$

for some index sets $I_1, \dots, I_\ell \subseteq \{1, \dots, n\}$ and $\ell \in \mathbb{N}$ minimum.

[3] Here and in Lemma 2 and Corollary 3, 'general' is used in the sense of algebraic geometry: A property is general if the set of points that does not have it is contained in a Zariski closed subset that is not the whole space.

Lemma 2 *Given a general $x \in \mathbb{R}^n$ and any integer k, $1 \le k \le n$, there is a k-element subset $I \subseteq \{1, \dots, n\}$ and a $\lambda \in \mathbb{R}$ such that $\sigma(\lambda \mathbb{1} + x) \in \mathbb{R}^n_I$.*

Proof Let $x = (x_1, \dots, x_n)^\intercal \in \mathbb{R}^n$ be general. Without loss of generality, we may assume its coordinates are in ascending order $x_1 < \cdots < x_n$. For any k with $1 \le k \le n$, choose λ so that $x_{n-k} < \lambda < x_{n-k+1}$ where we set $x_0 = -\infty$. Then $\sigma(u - \lambda \mathbb{1}) \in \mathbb{R}_{\{n-k+1,\dots,n\}}$. □

Lemma 3 *Let $n \ge p+1$. There is a nonempty open subset of vectors $v_1, \dots, v_p \in \mathbb{R}^n$ such that for any $p+1 \le k \le n$, there are a k-element subset $I \subseteq \{1, \dots, n\}$ and $\lambda_1, \dots, \lambda_p, \mu \in \mathbb{R}$ where*

$$\sigma(\lambda_1 \mathbb{1} + v_1), \dots, \sigma(\lambda_p \mathbb{1} + v_p), \sigma(\mu \mathbb{1} + v_1) \in \mathbb{R}^n_I$$

are linearly independent.

Proof For each $i = 1, \dots, p$, we choose general $v_i = (v_{i,1}, \dots, v_{i,n})^\intercal \in \mathbb{R}^n$ so that

$$v_{i,1} < \cdots < v_{i,n}.$$

For any fixed k with $p + 1 \le k \le n$, by Lemma 2, we can find $\lambda_1, \dots, \lambda_p, \mu \in \mathbb{R}$ such that $\sigma(v_i - \lambda_i \mathbb{1}) \in \mathbb{R}_{\{n-k+1,\dots,n\}}$, $i = 1, \dots, n$, and $\sigma(v_1 - \mu \mathbb{1}) \in \mathbb{R}_{\{n-k+1,\dots,n\}}$. By the generality of v_i's, the vectors $\sigma(v_1 - \lambda_1 \mathbb{1}), \dots, \sigma(v_p - \lambda_p \mathbb{1}), \sigma(v_1 - \mu \mathbb{1})$ are linearly independent. □

We are now ready to state our main result on the dimension of the image of weights of a two-layer ReLU-activated neural network.

Theorem 3 (Dimension of two-layer neural network I) *Let $n \ge d(p + 1) + 1$ where p is the dimension of the input and d is the dimension of the hidden layer. Then there is a nonempty open subset of samples $S \in \mathbb{R}^{n \times p}$ such that the image of weights for the two-layer ReLU-activated network (23) has dimension*

$$\dim \psi_2(\Theta) = d(p + 1) + 1. \tag{27}$$

Proof The rows of $S = [s_1^\intercal, \dots, s_n^\intercal]^\intercal \in \mathbb{R}^{n \times p}$ are the n samples $s_1, \dots, s_n \in \mathbb{R}^p$. In this case it will be more convenient to consider the columns of S, which we will denote by $v_1, \dots, v_p \in \mathbb{R}^n$. Denote the coordinates by $v_i = (v_{i,1}, \dots, v_{i,n})^\intercal$, $i = 1, \dots, p$. Consider the nonempty open subset

$$U \coloneqq \{S = [v_1, \dots, v_p] \in \mathbb{R}^{n \times p} : v_{i,1} < \cdots < v_{i,n},\ i = 1, \dots, p\}. \tag{28}$$

Define the index sets $J_i \subseteq \{1, \dots, n\}$ by

$$J_i \coloneqq \{n - i(p + 1) + 1, \dots, n\}, \qquad i = 1, \dots, d.$$

By Lemma 3,

$$\dim F_{J_i}(S) = \operatorname{rank}[S, \mathbb{1}] = p + 1, \qquad i = 1, \dots, d.$$

When $S \in U$ is sufficiently general,

$$\operatorname{span} F_{J_1}(S) + \cdots + \operatorname{span} F_{J_d}(S) = \operatorname{span} F_{J_1}(S) \oplus \cdots \oplus \operatorname{span} F_{J_d}(S). \quad (29)$$

Given any $I \subseteq \{1, \dots, n\}$ with $F_I(S) \neq \varnothing$, we have that for any $x, y \in F_I(S)$ and any $a, b > 0$, $ax + by \in F_I(S)$. This implies that

$$\dim F_I(S) = \dim \operatorname{span} F_I(S) = \dim \Sigma_r^\circ\big(F_I(S)\big) \quad (30)$$

for any $r \in \mathbb{N}$. Let $I_1, \dots, I_\ell \subseteq \{1, \dots, n\}$ and $\ell \in \mathbb{N}$ be chosen as in (26). Then

$$\Sigma_d^\circ\big(\mathrm{C}_{\max}(S)\big) = \bigcup_{1 \le i_1 \le \cdots \le i_d \le \ell} \operatorname{Join}\big(F_{I_{i_1}}(S), \dots, F_{I_{i_d}}(S)\big).$$

Now choose $I_{i_1} = J_1, \dots, I_{i_d} = J_d$. By (29) and (30),

$$\dim \operatorname{Join}\big(F_{J_1}(S), \dots, F_{J_d}(S)\big) = \sum_{i=1}^{d} \dim F_{J_i}(S).$$

Therefore

$$\begin{aligned} \dim \Sigma_d^\circ\big(\mathrm{C}_{\max}(S)\big) &= \dim \operatorname{Join}\big(F_{J_1}(S), \dots, F_{J_d}(S)\big) + \dim \operatorname{span}\{\mathbb{1}\} \\ &= d \operatorname{rank}[S, \mathbb{1}] + 1, \end{aligned}$$

which gives us (27). □

A consequence of Theorem 3 is that the dimension formula (27) holds for any general sample $s_1, \dots, s_n \in \mathbb{R}^p$ when n is sufficiently large.

Corollary 3 (Dimension of two-layer neural network II) *Let $n \gg pd$. Then for general $S \in \mathbb{R}^{n \times p}$, the image of weights for the two-layer ReLU-activated network (23) has dimension*

$$\dim \psi_2(\Theta) = d(p+1) + 1.$$

Proof Let the notations be as in the proof of Theorem 3. When n is sufficiently large, we can find a subset

$$I = \{i_1, \dots, i_{d(p+1)+1}\} \subseteq \{1, \dots, n\}$$

such that either

$$v_{j,i_1} < \cdots < v_{j,i_{d(p+1)+1}} \qquad \text{or} \qquad v_{j,i_1} > \cdots > v_{j,i_{d(p+1)+1}}$$

for each $j = 1, \dots, p$. The conclusion then follows from Theorem 3. □

For deeper networks one may have $m \gg \dim \psi_k(\Theta)$ even for $n \gg 0$. Consider a k-layer network with one neuron in every layer, i.e.,

$$d_1 = d_2 = \cdots = d_k = d_{k+1} = 1.$$

For any samples $s_1, \dots, s_n \in \mathbb{R}$, we may assume $s_1 \le \cdots \le s_n$ without loss of generality. Then the image of weights $\psi_k(\Theta) \subseteq \mathbb{R}^n$ may be described as follows: a point $x = (x_1, \dots, x_n)^\intercal \in \psi_k(\Theta)$ if and only if

$$x_1 = \cdots = x_\ell \le \cdots \le x_{\ell+\ell'} = \cdots = x_n$$

and

$$x_{\ell+1}, \dots, x_{\ell+\ell'-1} \text{ are the affine images of } s_{\ell+1}, \dots, s_{\ell+\ell'-1}.$$

In particular, as soon as $k \ge 3$ the image of weights $\psi_k(\Theta)$ does not change and its dimension remains constant for any $n \ge 6$.

We next address the case where $q > 1$. One might think that by the $q = 1$ case in Theorem 2 and "one-layer" case in Corollary 1, the image of weights $\psi_2(\Theta) \subseteq \mathbb{R}^{n\times q}$ in this case is simply the direct sum of q copies of $\operatorname{Join}\big(\Sigma_d^\circ(\mathrm{C}_{\max}(S)), \operatorname{span}\{\mathbb{1}\}\big)$. It is in fact only a subset of that, i.e.,

$$\psi_2(\Theta) \subseteq \big\{[x_1, \dots, x_q] \in \mathbb{R}^{n\times q} : x_1, \dots, x_d \in \operatorname{Join}\big(\Sigma_d^\circ(\mathrm{C}_{\max}(S)), \operatorname{span}\{\mathbb{1}\}\big)\big\}$$

but equality does not in general hold.

Theorem 4 (Geometry of two-layer neural network II) *Consider the two-layer ReLU-activated network with p-dimensional inputs, d neurons in the hidden layer, and q-dimensional outputs:*

$$\mathbb{R}^p \xrightarrow{\alpha_1} \mathbb{R}^d \xrightarrow{\sigma_{\max}} \mathbb{R}^d \xrightarrow{\alpha_2} \mathbb{R}^q. \tag{31}$$

The image of weights is given by

$$\psi_2(\Theta) = \big\{[x_1, \dots, x_q] \in \mathbb{R}^{n\times q} : \textit{there exist } y_1, \dots, y_d \in \mathrm{C}_{\max}(S) \\ \textit{such that } x_i \in \operatorname{span}\{\mathbb{1}, y_1, \dots, y_d\},\ i = 1, \dots, d\big\}.$$

Proof Let $X = [x_1, \dots, x_q] \in \psi_2(\Theta) \subseteq \mathbb{R}^{n\times q}$. Suppose that the affine map $\alpha_2 : \mathbb{R}^d \to \mathbb{R}^q$ is given by $\alpha_2(x) = Ax + b$ where $A = [a_1, \dots, a_q] \in \mathbb{R}^{d\times q}$ and $b = (b_1, \dots, b_q)^\intercal \in \mathbb{R}^q$. Then each x_i is realized as in (25) in the proof of Theorem 2. Therefore we conclude that $[x_1, \dots, x_q] \in \psi_2(\Theta)$ if and only if there exist $y_1, \dots, y_d \in \mathrm{C}_{\max}(S)$ with

$$x_i = b_i \mathbb{1} + \sum_{j=1}^{d} a_{ij} y_j, \qquad i = 1, \dots, q,$$

for some $b_i, a_{ij} \in \mathbb{R}$, $i = 1, \dots, n$, $j = 1, \dots, d$. □

With Theorem 4, we may deduce analogues of (part of) Theorem 3 and Corollary 3 for the case $q > 1$. The proofs are similar to those of Theorem 3 and Corollary 3.

Corollary 4 (Dimension of two-layer neural network III) *The image of weights of the two-layer ReLU-activated network* (31) *with p-dimensional inputs, d neurons in the hidden layer, and q-dimensional output has dimension*

$$\dim \psi_2(\Theta) = (q + \operatorname{rank}[S, \mathbb{1}])d + q.$$

If the sample size n is sufficiently large, then for general $S \in \mathbb{R}^{n\times p}$, the dimension is

$$\dim \psi_2(\Theta) = (p + q + 1)d + q.$$

Note that by (14),

$$\dim \Theta = (p + 1)d + (d + 1)q = \dim \psi_2(\Theta)$$

in the latter case of Corollary 4, as we expect.

Note that when $\dim \psi_2(\Theta) = \dim \mathbb{R}^{n\times q}$, $\psi_2(\Theta)$ becomes a dense set in $\mathbb{R}^{n\times q}$. Thus the expressions for $\dim \psi_2(\Theta)$ in Theorem 3, Corollaries 3 and 4 allows one to ascertain when the neural network equations (21) have a solution generically, namely, when $\dim \psi_2(\Theta) = nq$.

6 Smooth activations

For smooth activation like sigmoidal and hyperbolic tangent, we expect the geometry of the image of weights to be considerably more difficult to describe. Nevertheless when it comes to the ill-posedness of the best k-layer neural network problem (10), it is easy to deduce not only that there is a $T \in \mathbb{R}^{n\times q}$ such that (10) does not attain its infimum, but that there is a positive-measured set of such T's.

The phenomenon is already visible in the one-dimensional case $p = q = 1$ and can be readily extended to arbitrary p and q. Take the sigmoidal activation $\sigma_{\exp}(x) = 1/\big(1 + \exp(-x)\big)$. Let $n = 1$. So the sample S and response matrix T are both in $\mathbb{R}^{1\times 1} = \mathbb{R}$. Suppose $S \neq 0$. Then for a $\sigma_{\exp}$-activated k-layer neural network of arbitrary $k \in \mathbb{N}$,

$$\psi_k(\Theta) = (0, 1).$$

Therefore any $T \geq 1$ or $T \leq 0$ will not have a best approximation by points in $\psi_k(\Theta)$. The same argument works for the hyperbolic tangent activation $\sigma_{\tanh}(x) = \tanh(x)$ or indeed any activation σ whose range is a proper open interval. In this sense, the ReLU activation $\sigma_{\max}$ is special in that its range is not an open interval.

To show that the $n = 1$ assumption above is not the cause of the ill-posedness, we provide a more complicated example with $n = 3$. Again we will keep $p = q = 1$ and let

$$s_1 = 0, \quad s_2 = 1, \quad s_3 = 2; \quad t_1 = 0, \quad t_2 = 2, \quad t_3 = 1.$$

Consider a $k = 2$ layer neural network with hyperbolic tangent activation

$$\mathbb{R} \xrightarrow{\alpha_1} \mathbb{R} \xrightarrow{\sigma_{\tanh}} \mathbb{R} \xrightarrow{\alpha_2} \mathbb{R}. \tag{32}$$

Note that its weights take the form

$$\theta = (a, b, c, d) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \cong \mathbb{R}^4,$$

and thus $\Theta = \mathbb{R}^4$. It is also straightforward to see that

$$\psi_2(\Theta) = \left\{ \begin{bmatrix} c\dfrac{e^b - e^{-b}}{e^b + e^{-b}} + d \\ c\dfrac{e^{a+b} - e^{-a-b}}{e^{a+b} + e^{-a-b}} + d \\ c\dfrac{e^{2a+b} - e^{-2a-b}}{e^{2a+b} + e^{-2a-b}} + d \end{bmatrix} \in \mathbb{R}^3 : a, b, c, d \in \mathbb{R} \right\}.$$

For $\varepsilon > 0$, consider the open set of response matrices

$$U(\varepsilon) \coloneqq \left\{ \begin{bmatrix} t_1' \\ t_2' \\ t_3' \end{bmatrix} \in \mathbb{R}^3 : |t_1 - t_1'| \leq \varepsilon,\ |t_2 - t_2'| \leq \varepsilon,\ |t_3 - t_3'| \leq \varepsilon \right\}.$$

We claim that for ε small enough, any response matrix $T' = (t_1', t_2', t_3')^\intercal \in U(\varepsilon)$ will not have a best approximation in $\psi_2(\Theta)$.

Any best approximation of $T = (0, 2, 1)^\intercal$ in the closure of $\psi_2(\Theta)$ must take of the form $(0, y, y)^\intercal$ for some $y \in [1, 2]$. On the other hand, $(0, y, y)^\intercal \notin \psi_2(\Theta)$ for any $y \in [1, 2]$ and thus T does not have a best approximation in $\psi_2(\Theta)$. Similarly, for small $\varepsilon > 0$ and $T' = (t_1', t_2', t_3')^\intercal \in U$, a best approximation of T' in the closure of $\psi_2(\Theta)$ must take the form $(t_1', y, y)^\intercal$ for some $y \in [t_3', t_2']$. Since $(t_1', y, y)^\intercal \notin \psi_2(\Theta)$ for any $y \in [t_3', t_2']$, T' has no best approximation in $\psi_2(\Theta)$. Thus for small enough $\varepsilon > 0$, the infimum in (10) is unattainable for any $T \in U(\varepsilon)$, a nonempty open set.

Theorem 5 (Ill-posedness of neural network approximation II) *Let $n \geq 3$, $p \geq 1$, $q = 1$, and $k \geq 2$. Then there exists a positive-measured set $U \subseteq \mathbb{R}^{n \times q}$ and some $S \in \mathbb{R}^{n \times p}$ such that the best k-layer neural network approximation problem (10) with hyperbolic tangent activation $\sigma_{\tanh}$ does not attain its infimum for any $T \in U$.*

Proof The discussion preceding the theorem gives an explicit example for $n = 3$, $p = q = 1$, and $k = 2$. It remains to show that the values of n, p, k can be extended arbitrarily. Firstly, observe that the assumption $p = 1$ is totally unnecessary since we may embed $\mathbb{R}^1 \subseteq \mathbb{R}^p$ — our example works as long as the domain of the neural network contains a line. Secondly, it is also trivial to replace $n = 3$ by any $n > 3$, setting $s_i = s_1$ for all $i > 3$. Thirdly, we may extend the number of layers $k = 2$ to arbitrary $k > 2$, keeping all $d_i = 1$, as neither the affine transformation nor tanh can nontrivially change the order of points. Finally, under these assumptions, we note that the set S may be chosen to have positive measure as in the example constructed above. □

Note that the implicit assumption $d_i = 1$ may not be omitted from our construction. Indeed, if we allow $d_i > 1$, then for $n = 3$ any function may be fitted perfectly without any error. In particular, the infimum is always attained.

We leave open the question as to whether Theorem 5 holds for the ReLU activation $\sigma_{\max}$ or for outputs of dimension $q > 1$. Despite our best efforts, we are unable to construct an example nor show that such an example cannot possibly exist.

7 Concluding remarks

This article studies the best k-layer neural network approximation from the perspective of our earlier work [15], where we studied similar issues for the best k-term approximation. An important departure from [15] is that a neural network is not an algebraic object because the most common activation functions $\sigma_{\max}$, $\sigma_{\tanh}$, $\sigma_{\exp}$ are not polynomials; thus the algebraic techniques in [15] do not apply in our study here and are relevant at best only through analogy.

Nevertheless, by the Stone–Weierstrass theorem continuous functions may be uniformly approximated by polynomials. This suggests that it might perhaps be fruitful to study "algebraic neural networks," i.e., where the activation function σ is a polynomial function. This will allow us to apply the full machinery of algebraic geometry to deduce information about the image of weights $\psi_k(\Theta)$ on the one hand and to extend the field of interest from $\mathbb{R}$ to $\mathbb{C}$ on the other. In fact one of the consequences of our results in [15] is that for an algebraic neural network over $\mathbb{C}$, i.e., $\Theta = \mathbb{C}^m$, any response matrix $T \in \mathbb{C}^{n\times q}$ will almost always have a unique best approximation in $\psi_k(\mathbb{C}^m)$, i.e., the approximation problem (10) attains its infimum with probability one.

Furthermore, from our perspective, the most basic questions about neural network approximations are the ones that we studied in this article but questions like:

generic dimension for neural networks: for a general $S \in \mathbb{R}^{n\times p}$ with $n \gg 0$, what is the dimension of $\psi_k(\Theta)$?

generic rank for neural networks: what is the smallest value of $k \in \mathbb{N}$ such that $\psi_k(\Theta)$ is a dense set in $\mathbb{R}^{n\times q}$?

These are certainly the questions that one would first try to answer about various types of tensor ranks [11] or tensor networks [17] but as far as we know, they have never been studied for neural networks. We leave these as directions for potential future work.

Acknowledgements The work of YQ and LHL is supported by DARPA D15AP00109 and NSF IIS 1546413. In addition LHL acknowledges support from a DARPA Director's Fellowship and the Eckhardt Faculty Fund. MM would like to thank Guido Montufar for helpful discussions. Special thanks go to the anonymous referees for their careful reading and numerous invaluable suggestions that significantly enriched the content of this article.

References

1. Blum, A., Rivest, R.L.: Training a 3-node neural network is NP-complete (extended abstract). In: Proceedings of the 1988 Workshop on Computational Learning Theory (Cambridge, MA, 1988), pp. 9–18. Morgan Kaufmann, San Mateo, CA (1989)
2. Blum, A.L., Rivest, R.L.: Training a 3-node neural network is NP-complete. In: Machine learning: from theory to applications, *Lecture Notes in Comput. Sci.*, vol. 661, pp. 9–28. Springer, Berlin (1993)
3. Burger, M., Engl, H.W.: Training neural networks with noisy data as an ill-posed problem. Adv. Comput. Math. **13**(4), 335–354 (2000)
4. Bürgisser, P., Cucker, F.: Condition: The geometry of numerical algorithms, *Grundlehren der Mathematischen Wissenschaften*, vol. 349. Springer, Heidelberg (2013)
5. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signals Systems **2**(4), 303–314 (1989)
6. De Silva, V., Lim, L.H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J. Matrix Anal. Appl. **30**(3), 1084–1127 (2008)
7. Girosi, F., Poggio, T.: Networks and the best approximation property. Biol. Cybernet. **63**(3), 169–176 (1990)
8. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Netw. **4**(2), 251–257 (1991)
9. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
11. Landsberg, J.M.: Tensors: geometry and applications, *Graduate Studies in Mathematics*, vol. 128. American Mathematical Society, Providence, RI (2012)
12. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
13. O'Leary-Roseberry, T., Ghattas, O.: Ill-posedness and optimization geometry for nonlinear neural network training. arXiv:2002.02882 (2020)
14. Petersen, P., Raslan, M., Voigtlaender, F.: Topological properties of the set of functions generated by neural networks of fixed size. arXiv:1806.08459 (2018)
15. Qi, Y., Michał ek, M., Lim, L.H.: Complex best r-term approximations almost always exist in finite dimensions. Appl. Comput. Harmon. Anal. **49**(1), 180–207 (2020)
16. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nature **529**, 484–489 (2016)
17. Ye, K., Lim, L.H.: Tensor network ranks. arXiv:1801.02662 (2018)
18. Zak, F.L.: Tangents and secants of algebraic varieties, *Translations of Mathematical Monographs*, vol. 127. American Mathematical Society, Providence, RI (1993)