

Deep Network Approximation Characterized by Number of Neurons*

Zuowei Shen[†] Haizhao Yang[‡] Shijun Zhang[§]

July 31, 2020

Abstract

This paper quantitatively characterizes the approximation power of deep feed-forward neural networks (FNNs) in terms of the number of neurons. It is shown by construction that ReLU FNNs with width $C_1 \max\{d\lfloor N^{1/d} \rfloor, N+1\}$ and depth $12L+C_2$ can approximate an arbitrary Hölder continuous function of order α with a constant λ on $[0, 1]^d$ with a nearly tight approximation rate $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$ measured in L^p -norm for any given $N, L \in \mathbb{N}^+$ and $p \in [1, \infty]$. The constructive approximation is a corollary of a more general result for an arbitrary continuous function f in terms of its modulus of continuity $\omega_f(\cdot)$. In particular, the approximation rate of ReLU FNNs with width $C_1 \max\{d\lfloor N^{1/d} \rfloor, N+1\}$ and depth $12L+C_2$ for a general continuous function f is $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$. We also extend our analysis to f on irregular domains or those localized in an ε -neighborhood of a $d_{\mathcal{M}}$ -dimensional smooth manifold $\mathcal{M} \subseteq [0, 1]^d$ with $d_{\mathcal{M}} \ll d$. Especially, in the case of an essentially low-dimensional domain, we show an approximation rate $2\omega_f(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon) + 19\sqrt{d}\omega_f(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta})$ for ReLU FNNs to approximate f in the ε -neighborhood, where $d_\delta = \mathcal{O}(d_{\mathcal{M}}\frac{\ln(d/\delta)}{\delta^2})$ for any given $\delta \in (0, 1)$ as a relative error for a projection to approximate an isometry when we project \mathcal{M} to a d_δ -dimensional domain.

Key words. Deep ReLU Neural Networks, Hölder Continuity, Modulus of Continuity, Approximation Theory, Low-Dimensional Manifold, Parallel Computing.

1 Introduction

The approximation theory of neural networks has been an active research topic in the past few decades. Previously, as a special kind of ridge function approximation, shallow neural networks with one hidden layer and various activation functions (e.g., wavelets pursuits [10, 46], adaptive splines [19, 55], radial basis functions [8, 18, 25, 53, 65], sigmoid

*Submitted to the editors DATE.



[†]Department of Mathematics, National University of Singapore (matzuows@nus.edu.sg).

[‡]Department of Mathematics, Purdue University (haizhao@purdue.edu).

[§]Department of Mathematics, National University of Singapore (zhangshijun@u.nus.edu).


functions [7, 13–15, 29, 37, 38, 41, 45]) were widely discussed and admit good approximation properties, e.g., the universal approximation property [16, 29, 30], lessening the curse of dimensionality [4, 21, 22], and providing attractive approximation rate in nonlinear approximation [10, 18, 19, 25, 46, 55, 65].

The introduction of deep networks with more than one hidden layers has made significant impacts in many fields in computer science and engineering including computer vision [35] and natural language processing [1]. New scientific computing tools based on deep networks have also emerged and facilitated large-scale and high-dimensional problems that were impractical previously [20, 24]. The design of deep ReLU FNNs is the key of such a revolution. These breakthroughs have stimulated broad research topics from different points of views to study the power of deep ReLU FNNs, e.g. in terms of combinatorics [51], topology [6], Vapnik-Chervonenkis (VC) dimension [5, 27, 58], fat-shattering dimension [2, 34], information theory [54], classical approximation theory [4, 16, 30, 62, 67], optimization [32, 33, 52] etc.

Particularly in approximation theory, **non-quantitative and asymptotic** approximation rates of ReLU FNNs have been proposed for various types of functions. For example, smooth functions [23, 39, 43, 66], piecewise smooth functions [54], band-limited functions [50], continuous functions [67], solutions to partial differential equations [31]. However, to the best of our knowledge, existing theories [17, 23, 39, 43, 48, 50, 54, 63, 66, 67] can only provide implicit formulas in the sense that the approximation error contains an unknown prefactor, or work only for sufficiently large N and L larger than some unknown numbers. For example, [67] estimated an approximation rate $c(d)L^{-2\alpha/d}$ via a narrow and deep ReLU FNN, where $c(d)$ is an unknown number depending on d , and L is required to be larger than a sufficiently large unknown number \mathcal{L} . For another example, given an approximation error ε , [54] proved the existence of a ReLU FNN with a constant but still unknown number of layers approximating a C^β function within the target error. These works can be divided into two cases: 1) FNNs with varying width and only one hidden layer [18, 25, 40, 65] (visualized by the region in  in Figure 1); 2) FNNs with a fixed width of $\mathcal{O}(d)$ and a varying depth larger than an unknown number \mathcal{L} [44, 67] (represented by the region in  in Figure 1).

As far as we know, the first **quantitative and non-asymptotic** approximation rate of deep ReLU FNNs was obtained in [62]. Specifically, [62] identified an explicit formulas of the approximation rate

$$\begin{cases} 2\lambda N^{-2\alpha}, & \text{when } L \geq 2 \text{ and } d = 1, \\ 2(2\sqrt{d})^\alpha \lambda N^{-2\alpha/d}, & \text{when } L \geq 3 \text{ and } d \geq 2, \end{cases} \quad (1.1)$$

for ReLU FNNs with an arbitrary width $N \in \mathbb{N}^+$ and a fixed depth $L \in \mathbb{N}^+$ to approximate a Hölder continuous function f of order α with a Hölder constant λ (visualized in the region shown by  in Figure 1). The approximation rate $\mathcal{O}(N^{-2\alpha/d})$ is tight in terms of N and increasing L cannot improve the approximation rate in N . The success of deep FNNs in a broad range of applications has motivated a well-known conjecture that the depth L has an important role in improving the approximation power of deep FNNs. In particular, a very important question in practice would be, given an arbitrary L and N , what is the explicit formula to characterize the approximation error so as to see whether the network is large enough to meet the accuracy requirement. Due to the highly

nonlinear structure of deep FNNs, it is still a challenging open problem to characterize N and L simultaneously in the approximation rate.

To answer the question just above, we establish the first framework that is able to quantify the approximation power of deep ReLU FNNs essentially with arbitrary width N and depth L , achieving a nearly optimal approximation rate, $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$, for continuous functions $f \in C([0, 1]^d)$. Our result is based on new analysis techniques merely based on the structure of FNNs and a modified bit extraction technique inspired by [5], instead of designing FNNs to approximate traditional approximation basis like polynomials and splines as in the existing literature [26, 39, 43, 48, 49, 54, 56, 57, 60, 63, 66, 67]. The approximation rate obtained here admits an explicit formula to compute the prefactor when $\omega_f(\cdot)$ is known. For example, in the case of Hölder continuous functions of order α with a Hölder constant λ (denoted as the class $\text{Höld}([0, 1]^d, \alpha, \lambda)$), $\omega_f(r) \leq \lambda r^\alpha$ for $r \geq 0$, resulting in the approximation rate $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$ as mentioned previously. As a consequence, existing works for the function class $C([0, 1]^d)$ are special cases of our result (see Figure 1 for a comparison).

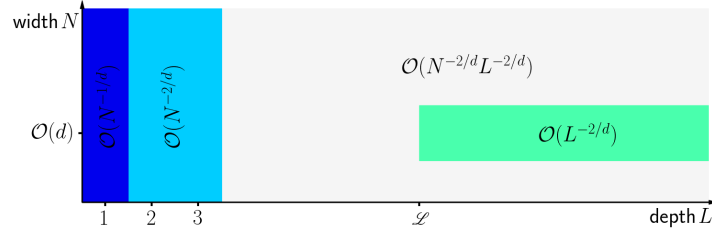


Figure 1: A summary of existing and our new results on the approximation rate of ReLU FNNs for continuous functions. Existing results [18, 25, 40, 44, 62, 65, 67] are applicable in the areas in , , and ; our new result is suitable for almost all areas when $L \geq 2$.

Our key contributions can be summarized as follows.

1. Lower bound: We provide a quantitative and non-asymptotic approximation rate $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$ in terms of width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ for functions in $C([0, 1]^d)$ in Theorem 1.1.
2. Upper bound: Through the nearly tight VC-dimension Bounds of ReLU FNNs [27], we show that the approximation rate $19\sqrt{d}\omega_f(N^{-2\alpha/d}L^{-2\alpha/d})$ in terms of N and L is nearly optimal for $\text{Höld}([0, 1]^d, \alpha, \lambda)$ in Theorem 2.3.
3. The approximation rate in terms of the width and depth in this paper is more generic and useful than the one characterized by the number of nonzero parameters denoted as W in the literature. First, the characterization in terms of width and depth implies the one in terms of W , while it is not true the other way around. Second, our theory can provide practical guidance for choosing network sizes in realistic applications while theories in terms of W cannot tell how large a network should be to guarantee a target accuracy, since there are too many networks of different sizes sharing the same number of parameters but with different accuracies.

103 4. Finally, three aspects of neural networks in practice are discussed: 1) neural net-
 104 work approximation in a high-dimensional irregular domain; 2) neural network
 105 approximation in the case of a low-dimensional data structure; 3) the optimal
 106 ReLU FNN in parallel computation.

107 Our main result, Theorem 1.1 below, shows that ReLU FNNs with width $\mathcal{O}(N)$
 108 and depth $\mathcal{O}(L)$ can approximate f with an approximation rate $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$,
 109 where $\omega_f(\cdot)$ is the modulus of continuity of f defined via

$$110 \quad \omega_f(r) := \sup \{ |f(\mathbf{x}) - f(\mathbf{y})| : \mathbf{x}, \mathbf{y} \in [0, 1]^d, \|\mathbf{x} - \mathbf{y}\|_2 \leq r \}, \quad \text{for any } r \geq 0.$$

111 **Theorem 1.1.** *Given $f \in C([0, 1]^d)$, for any $L \in \mathbb{N}^+$, $N \in \mathbb{N}^+$, and $p \in [1, \infty]$, there exists*
 112 *a function ϕ implemented by a ReLU FNN with width $C_1 \max \{ d \lfloor N^{1/d} \rfloor, N+1 \}$ and depth*
 113 *$12L + C_2$ such that*

$$114 \quad \|f - \phi\|_{L^p([0,1]^d)} \leq 19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}),$$

115 where $C_1 = 12$ and $C_2 = 14$ if $p \in [1, \infty)$; $C_1 = 3^{d+3}$ and $C_2 = 14 + 2d$ if $p = \infty$.

116 When Theorem 1.1 is applied to $f \in \text{Höld}([0, 1]^d, \alpha, \lambda)$, the approximation rate is
 117 $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$, because $\omega_f(r) \leq \lambda r^\alpha$ for any $r \geq 0$. An immediate question following
 118 the constructive approximation is how much we can improve the approximation rate. In
 119 fact, the approximation rate of $f \in \text{Höld}([0, 1]^d, \alpha, \lambda)$ is asymptotically tight based on
 120 VC-dimension as we shall see later.

121 In most real applications of neural networks, though the target function f is defined
 122 in a high-dimensional domain, e.g., $[0, 1]^d$, where d could be tens of thousands or even
 123 millions, only the approximation error of f in a neighborhood of a $d_{\mathcal{M}}$ -dimensional
 124 manifold \mathcal{M} with $d_{\mathcal{M}} \ll d$ is concerned. Hence, we extend Theorem 1.1 to the case
 125 when the domain of f is localized in an ε -neighborhood of a compact $d_{\mathcal{M}}$ -dimensional
 126 Riemannian submanifold $\mathcal{M} \subseteq [0, 1]^d$ having condition number $1/\tau$, volume V , and
 127 geodesic covering regularity \mathcal{R} . The ε -neighborhood is defined as

$$128 \quad \mathcal{M}_\varepsilon := \{ \mathbf{x} \in [0, 1]^d : \inf \{ \|\mathbf{x} - \mathbf{y}\|_2 : \mathbf{y} \in \mathcal{M} \} \leq \varepsilon \}, \quad \text{for } \varepsilon \in (0, 1). \quad (1.2)$$

129 Let $d_\delta = \mathcal{O}\left(\frac{d_{\mathcal{M}} \ln(dV\mathcal{R}\tau^{-1}\delta^{-1})}{\delta^2}\right) = \mathcal{O}\left(d_{\mathcal{M}} \frac{\ln(d/\delta)}{\delta^2}\right)$ be an integer for any $\delta \in (0, 1)$ such that

130 $d_{\mathcal{M}} \leq d_\delta \leq d$. We show an approximation rate $2\omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right)$
 131 for ReLU FNNs to pointwisely approximate f on \mathcal{M}_ε . The key ideas of the proof is the
 132 application of Theorem 3.1 in [3], which provides a nearly isometric projection $\mathbf{A} \in \mathbb{R}^{d_\delta \times d}$
 133 that maps points in $\mathcal{M} \subseteq [0, 1]^d$ to a d_δ -dimensional domain with

$$134 \quad (1 - \delta)|\mathbf{x}_1 - \mathbf{x}_2| \leq |\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| \leq (1 + \delta)|\mathbf{x}_1 - \mathbf{x}_2| \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M},$$

135 and the application of Theorem 1.1 in this paper, which constructs the desired ReLU
 136 FNN with a size depending on d_δ instead of d to lessen the curse of dimensionality.
 137 When δ is closer to 1, d_δ is closer to $d_{\mathcal{M}}$ but the isometric property of the projection is
 138 weakened; when δ is closer to 0, the isometric property becomes better but d_δ could be
 139 larger than d , in which case we can simply enforce $d_\delta = d$ and choose the identity map
 140 as the projection. Hence, $\delta \in (0, 1)$ is a parameter to make a balance between isometry
 141 and dimension reduction.

Theorem 1.2. *Let f be a continuous function on $[0, 1]^d$ and $\mathcal{M} \subseteq [0, 1]^d$ be a compact $d_{\mathcal{M}}$ -dimensional Riemannian submanifold. For any $N \in \mathbb{N}^+$, $L \in \mathbb{N}^+$, $\varepsilon \in (0, 1)$, and $\delta \in (0, 1)$, there exists a function ϕ implemented by a ReLU FNN with width $3^{d_{\delta}+3} \max\{d_{\delta} \lfloor N^{1/d_{\delta}} \rfloor, N+1\}$ and depth $12L + 14 + 2d_{\delta}$ such that*

$$|f(\mathbf{x}) - \phi(\mathbf{x})| \leq 2\omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_{\delta}}} + 2\varepsilon\right) + 19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_{\delta}}}N^{-2/d_{\delta}}L^{-2/d_{\delta}}\right), \quad (1.3)$$

for any $\mathbf{x} \in \mathcal{M}_{\varepsilon}$, where $\mathcal{M}_{\varepsilon}$ is defined in Equation (1.2)

The approximation rate of deep neural networks for functions defined precisely on low-dimensional smooth manifolds has been studied in [61] for C^2 functions and in [9, 11] for Lipschitz continuous functions. Considering that it might be more reasonable to assume data located in a small neighborhood of low-dimensional smooth manifold in real applications, we introduce the ε -neighborhood of the manifold \mathcal{M} in Theorem 1.2. In general, existing results are again asymptotic and they cannot be applied to estimate the approximation accuracy of a ReLU FNN with arbitrarily given width N and depth L , since there is no explicit formula without unknown constants to specify the exact error bound. For example, [9] provides an approximation rate $c_1(NL)^{-c_2/d_{\delta}}$ with unknown constants (e.g., c_1 and c_2) and requires NL greater than an unknown large number. The demand of an explicit error estimation motivates Theorem 1.2 in this paper. When data are concentrating around \mathcal{M} , ε is very small and the dominant term of the approximation error in (1.3) is $19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_{\delta}}}N^{-2/d_{\delta}}L^{-2/d_{\delta}}\right)$ implying that the approximation via deep ReLU FNNs can lessen the curse of dimensionality.

The analysis above provides a general guide for selecting the width and depth of ReLU FNNs to approximate continuous functions, especially when the computation is conducted with parallel computing, which is usually the case in real applications [12, 59]. As we shall see later, when the approximation accuracy and the parallel computing efficiency are considered together, very deep FNNs become less attractive than those with $\mathcal{O}(1)$ depth.

The approximation theories in this paper assume that the target function f is fully accessible, making it possible to estimate the approximation error and identify an asymptotically optimal ReLU FNN with a given budget of neurons to minimize the approximation error. In real applications, usually only a limited number of possibly noisy observations of f is available, resulting in a regression problem in statistics. In the latter case, the problem is usually formulated in a stochastic setting with randomly generated noisy observations and the regression error contains mainly two components: bias and variance. The bias is the difference of the expectation of an estimated function and its ground truth f . The approximation theories in this paper play an important role in characterizing the power of neural networks when they are applied to solve regression problems by providing a lower bound of the regression bias.

The rest of this paper is organized as follows. We first prove Theorem 1.1 and show its optimality in Section 2 when assuming Theorem 2.1 is true. Next, Theorem 2.1 is proved in Section 3. In Section 4, three aspects of neural networks in practice will be discussed: 1) neural network approximation in a high-dimensional irregular domain; 2) neural network approximation in the case of a low-dimensional data structure; 3) the optimal ReLU FNN in parallel computation. Finally, Section 5 concludes this paper with a short discussion.

2 Approximation of continuous functions

In this section, we prove Theorem 1.1 and discuss its optimality when assume Theorem 2.1 is true. Notations throughout the proof will be summarized in Section 2.1.

2.1 Notations

Let us summarize all basic notations used in this paper as follows.

- Matrices are denoted by bold uppercase letters. For instance, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a real matrix of size $m \times n$, and \mathbf{A}^T denotes the transpose of \mathbf{A} .
- Vectors are denoted as bold lowercase letters. For example, $\mathbf{v} \in \mathbb{R}^n$ is a column vector of size n . Correspondingly, $\mathbf{v}(i)$ is the i -th element of \mathbf{v} . $\mathbf{v} = (v_1, \dots, v_n) = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$ are vectors consisting of numbers $\{v_i\}_i$ with $\mathbf{v}(i) = v_i$.
- Let $\mu(\cdot)$ be the Lebesgue measure.
- Let 1_S be the characteristic function on a set S , i.e., 1_S equals to 1 on S and 0 outside of S .
- The set difference of two sets A and B is denoted by $A \setminus B := \{x : x \in A, x \notin B\}$.
- For any $\xi \in \mathbb{R}$, let $\lfloor \xi \rfloor := \max\{i : i \leq \xi, i \in \mathbb{Z}\}$ and $\lceil \xi \rceil := \min\{i : i \geq \xi, i \in \mathbb{Z}\}$.
- Assume $\mathbf{n} \in \mathbb{N}^n$, then $f(\mathbf{n}) = \mathcal{O}(g(\mathbf{n}))$ means that there exists positive C independent of \mathbf{n} , f , and g such that $f(\mathbf{n}) \leq Cg(\mathbf{n})$ when all entries of \mathbf{n} go to $+\infty$.
- Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ denote the rectified linear unit (ReLU), i.e. $\sigma(x) = \max\{0, x\}$. With the abuse of notations, we define $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^d$ as $\sigma(\mathbf{x}) = \begin{bmatrix} \max\{0, x_1\} \\ \vdots \\ \max\{0, x_d\} \end{bmatrix}$ for any $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$.
- Given $K \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{K})$, define a trifling region $\Omega([0, 1]^d, K, \delta)$ of $[0, 1]^d$ as

$$\Omega([0, 1]^d, K, \delta) := \bigcup_{i=1}^d \left\{ \mathbf{x} = (x_1, x_2, \dots, x_d) : x_i \in \cup_{k=1}^{K-1} \left(\frac{k}{K} - \delta, \frac{k}{K} \right) \right\}. \quad (2.1)$$

In particular, $\Omega([0, 1]^d, K, \delta) = \emptyset$ if $K = 1$. See Figure 2 for two examples of trifling regions.

- Let $\text{Höld}([0, 1]^d, \alpha, \lambda)$ be the set containing all Hölder continuous functions of order $\alpha \in (0, 1]$ with a Hölder constant λ .
- We will use \mathcal{NN} to denote a function implemented by a ReLU FNN for short and use Python-type notations to specify a class of functions implemented by ReLU FNNs with several conditions, e.g., $\mathcal{NN}(c_1; c_2; \dots; c_m)$ is a set of functions implemented by ReLU FNNs satisfying m conditions given by $\{c_i\}_{1 \leq i \leq m}$, each of which may

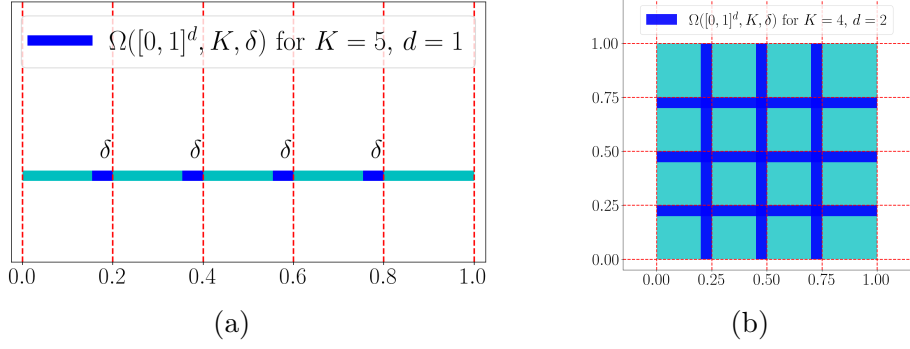


Figure 2: Two examples of trifling regions. (a) $K = 5, d = 1$. (b) $K = 4, d = 2$.

specify the number of inputs ($\#input$), the number of outputs ($\#output$), the total number of nodes in all hidden layers ($\#neuron$), the number of hidden layers (depth), the number of total parameters ($\#parameter$), and the width in each hidden layer ($widthvec$), the maximum width of all hidden layers ($width$), etc. For example, if $\phi \in \mathcal{NN}(\#input = 2; widthvec = [100, 100]; \#output = 1)$, then ϕ is a functions satisfies

- ϕ maps from \mathbb{R}^2 to \mathbb{R} .
- ϕ can be implemented by a ReLU FNN with two hidden layers and the number of nodes in each hidden layer is 100.

- $[n]^L$ is short for $[n, n, \dots, n] \in \mathbb{N}^L$. For example,

$$\mathcal{NN}(\#input = d; widthvec = [100, 100]) = \mathcal{NN}(\#input = d; widthvec = [100]^2).$$

- For $\phi \in \mathcal{NN}(\#input = d; widthvec = [N_1, N_2, \dots, N_L]; \#output = 1)$, if we define $N_0 = d$ and $N_{L+1} = 1$, then the architecture of ϕ can be briefly described as follows:

$$\mathbf{x} = \tilde{\mathbf{h}}_0 \xrightarrow{\mathbf{W}_0, \mathbf{b}_0} \mathbf{h}_1 \xrightarrow{\sigma} \tilde{\mathbf{h}}_1 \dots \xrightarrow{\mathbf{W}_{L-1}, \mathbf{b}_{L-1}} \mathbf{h}_L \xrightarrow{\sigma} \tilde{\mathbf{h}}_L \xrightarrow{\mathbf{W}_L, \mathbf{b}_L} \mathbf{h}_{L+1} = \phi(\mathbf{x}),$$

where $\mathbf{W}_i \in \mathbb{R}^{N_{i+1} \times N_i}$ and $\mathbf{b}_i \in \mathbb{R}^{N_{i+1}}$ are the weight matrix and the bias vector in the i -th linear transform \mathcal{L}_i in ϕ , respectively, i.e.,

$$\mathbf{h}_{i+1} := \mathbf{W}_i \tilde{\mathbf{h}}_i + \mathbf{b}_i =: \mathcal{L}_i(\tilde{\mathbf{h}}_i), \quad \text{for } i = 0, 1, \dots, L,$$

and

$$\tilde{\mathbf{h}}_i = \sigma(\mathbf{h}_i), \quad \text{for } i = 1, \dots, L.$$

In particular, ϕ can be represented in a form of function compositions as follows

$$\phi = \mathcal{L}_L \circ \sigma \circ \mathcal{L}_{L-1} \circ \sigma \circ \dots \circ \sigma \circ \mathcal{L}_1 \circ \sigma \circ \mathcal{L}_0,$$

which has been illustrated in Figure 3.

- The expression “an FNN with width N and depth L ” means

- The maximum width of this FNN for all hidden layers is no more than N .

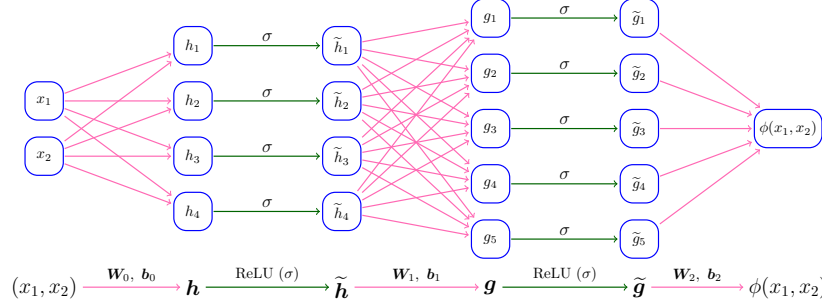


Figure 3: An example of ReLU networks with width 5 and depth 2.

– The number of hidden layers of this FNN is no more than L .

- For $\theta \in [0, 1)$, suppose its binary representation is $\theta = \sum_{\ell=1}^{\infty} \theta_{\ell} 2^{-\ell}$ with $\theta_{\ell} \in \{0, 1\}$, we introduce a special notation $\text{bin}0.\theta_1\theta_2\cdots\theta_L$ to denote the L -term binary representation of θ , i.e., $\sum_{\ell=1}^L \theta_{\ell} 2^{-\ell}$.

2.2 Proof of Theorem 1.1

We essentially construct piecewise constant functions to approximate continuous functions in the proof. However, it is impossible to construct a piecewise constant function via ReLU FNNs due to the continuity of ReLU FNNs. Thus, we introduce the trifling region $\Omega([0, 1]^d, K, \delta)$, defined in Equation (2.1), and use ReLU FNNs to implement piecewise constant functions outside of the trifling region. To prove Theorem 1.1, we first establish a theorem showing how to construct ReLU FNNs to pointwisely approximate continuous functions except the trifling region.

Theorem 2.1. *Given $f \in C([0, 1]^d)$, for any $L \in \mathbb{N}^+$ and $N \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU FNN with width $\max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\}$ and depth $12L + 14$ such that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$ and*

$$|f(\mathbf{x}) - \phi(\mathbf{x})| \leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}), \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

where $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ and δ is an arbitrary number in $(0, \frac{1}{3K}]$.

With Theorem 2.1 that will be proved in Section 3, we can easily prove Theorem 1.1 for the case $p \in [1, \infty)$. In the early version of this paper, which focuses on continuous functions as target functions, we only considered the case $p \in [1, \infty)$ since it was challenging to control the approximation error in the trifling region. Later in [42] when we considered smooth functions as target functions, we invented a technique that can handle the error in the trifling region as in the lemma below. Therefore, we are now able to control the approximation error for $p = \infty$. The results in this paper are for continuous functions, to which the results in [42] are not applicable; the results in [42] characterize how the smoothness of target functions helps to enhance the approximation capacity of ReLU FNNs, which is not addressed in this paper. It is interesting to point out that the approximation rate $\mathcal{O}(N^{-2/d}L^{-2/d})$ for continuous functions in this paper is even better than the rate $\mathcal{O}((\frac{N}{\ln N})^{-2/d}(\frac{L}{\ln L})^{-2/d})$ for functions in $C^1([0, 1]^d)$ in [42].

269 **Lemma 2.2** (Theorem 2.1 of [42]). Given $\varepsilon > 0$, $N, L, K \in \mathbb{N}^+$, and $\delta \in (0, \frac{1}{3K}]$, assume
 270 $f \in C([0, 1]^d)$ and $\tilde{\phi}$ can be implemented by a ReLU FNN with width N and depth L . If
 271 $|f(\mathbf{x}) - \tilde{\phi}(\mathbf{x})| \leq \varepsilon$, for any $\mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$,

272 then there exists a function ϕ implemented by a new ReLU FNN with width $3^d(N + 4)$
 273 and depth $L + 2d$ such that

$$274 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

275 Now we are ready to prove Theorem 1.1 by assuming Theorem 2.1 is true, which
 276 will be proved later in Section 3.2.

277 *Proof of Theorem 1.1.* Let us first consider the case $p \in [1, \infty)$. We may assume f is
 278 not a constant function since it is a trivial case. Then $\omega_f(r) > 0$ for any $r > 0$. Set
 279 $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ and choose a small $\delta \in (0, \frac{1}{3K}]$ such that

$$280 \quad \begin{aligned} Kd\delta(|f(\mathbf{0})| + \omega_f(\sqrt{d}))^p &= \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor d\delta(|f(\mathbf{0})| + \omega_f(\sqrt{d}))^p \\ &\leq (\omega_f(N^{-2/d}L^{-2/d}))^p. \end{aligned}$$

281 By Theorem 2.1, there exists a function ϕ implemented by a ReLU FNN with width

$$282 \quad \max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\} \leq 12 \max\{d\lfloor N^{1/d} \rfloor, N + 1\}$$

283 and depth $12L + 14$ such that

$$284 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}), \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

285 It follows from $\mu(\Omega([0, 1]^d, K, \delta)) \leq Kd\delta$ that

$$\begin{aligned} \|f - \phi\|_{L^p([0, 1]^d)}^p &= \int_{\Omega([0, 1]^d, K, \delta)} |f(\mathbf{x}) - \phi(\mathbf{x})|^p d\mathbf{x} + \int_{[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)} |f(\mathbf{x}) - \phi(\mathbf{x})|^p d\mathbf{x} \\ &\leq Kd\delta(|f(\mathbf{0})| + \omega_f(\sqrt{d}))^p + (18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}))^p \\ 286 \quad &\leq (\omega_f(N^{-2/d}L^{-2/d}))^p + (18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}))^p \\ &\leq (19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}))^p. \end{aligned}$$

287 Hence, $\|f - \phi\|_{L^p([0, 1]^d)} \leq 19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$.

288 Next, let us discuss the case $p = \infty$. Set $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ and choose a small
 289 $\delta \in (0, \frac{1}{3K}]$ such that

$$290 \quad d \cdot \omega_f(\delta) \leq \omega_f(N^{-2/d}L^{-2/d}).$$

291 By Theorem 2.1, there exists a function $\tilde{\phi}$ implemented by a ReLU FNN with width
 292 $\max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\}$ and depth $12L + 14$ such that

$$293 \quad |f(\mathbf{x}) - \tilde{\phi}(\mathbf{x})| \leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}) := \varepsilon, \quad \text{for } \mathbf{x} \in [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta),$$

294 By Lemma 2.2, there exists a function ϕ implemented by a ReLU FNN with width

$$295 \quad 3^d(\max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N + 8\} + 4) \leq 3^{d+3} \max\{d\lfloor N^{1/d} \rfloor, N + 1\}$$

296 and depth $12L + 14 + 2d$ such that

$$297 \quad |f(\mathbf{x}) - \phi(\mathbf{x})| \leq \varepsilon + d \cdot \omega_f(\delta) \leq 19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}), \quad \text{for any } \mathbf{x} \in [0, 1]^d.$$

298 So we finish the proof. □

2.3 Optimality of Theorem 1.1

This section will show that the approximation rate in Theorem 1.1 is nearly tight and there is no room to improve for the function class $\text{Höld}([0, 1]^d, \alpha, \lambda)$. Theorem 2.3 below shows that the approximation rate $\mathcal{O}(\omega_f(N^{-(2/d+\rho)}L^{-(2/d+\rho)}))$ for any $\rho > 0$ is unachievable, implying the approximation rate in Theorem 1.1 is nearly tight for the function class $\text{Höld}([0, 1]^d, \alpha, \lambda)$.

Theorem 2.3. *Given any $\rho > 0$ and $C > 0$, there exists $f \in \text{Höld}([0, 1]^d, \alpha, \lambda)$ such that, for any $J_0 > 0$, there exist $N, L \in \mathbb{N}$ with $NL \geq J_0$ satisfying*

$$\inf_{\phi \in \mathcal{NN}(\# \text{input}=d; \text{width} \leq N; \text{depth} \leq L)} \|\phi - f\|_{L^\infty([0, 1]^d)} \geq C\lambda N^{-(2\alpha/d+\rho)} L^{-(2\alpha/d+\rho)}.$$

In fact, we can show a stronger result than Theorem 2.3. Under the same conditions as in Theorem 2.3, for any $\mathcal{H} \in [0, 1]^d$ with $\mu(\mathcal{H}) \leq 2^{-(d+K^{d+1})}K^{-d}$, where $K = \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor$, it can be proved that

$$\inf_{\phi \in \mathcal{NN}(\# \text{input}=d; \text{width} \leq N; \text{depth} \leq L)} \|\phi - f\|_{L^\infty([0, 1]^d \setminus \mathcal{H})} \geq C\lambda N^{-(2\alpha/d+\rho)} L^{-(2\alpha/d+\rho)}. \quad (2.2)$$

We will prove (2.2) by contradiction, then Theorem 2.3 holds as a consequence. Assuming Equation (2.2) is false, we have the following claim.

Claim 2.4. *There exist $\rho > 0$ and $C > 0$ such that given any $f \in \text{Höld}([0, 1]^d, \alpha, \lambda)$, there exists $J_0 = J_0(\rho, C, f) > 0$ such that, for any $N, L \in \mathbb{N}$ with $NL \geq J_0$, there exist $\phi \in \mathcal{NN}(\# \text{input} = d; \text{width} \leq N; \text{depth} \leq L)$ and $\mathcal{H} \in [0, 1]^d$ with $\mu(\mathcal{H}) \leq 2^{-(d+K^{d+1})}K^{-d}$, where $K = \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor$, satisfying*

$$\|f - \phi\|_{L^\infty([0, 1]^d \setminus \mathcal{H})} \leq C\lambda N^{-(2\alpha/d+\rho)} L^{-(2\alpha/d+\rho)}.$$

Now let us disprove this claim to show Theorem 2.3 and Equation (2.2) are true.

Disproof of Claim 2.4. Without the loss of generality, we assume $\lambda = 1$; in the case of $\lambda \neq 1$, the proof is similar. We will disprove Claim 2.4 using the VC dimension. Recall that the VC dimension of a class of functions is defined as the cardinality of the largest set of points that this class of functions can shatter. Denote the VC dimension of a function set \mathcal{F} by $\text{VCDim}(\mathcal{F})$. By [27] and the fact

$$\mathcal{NN}(\text{width} \leq N; \text{depth} \leq L) \subseteq \mathcal{NN}(\# \text{parameter} \leq (LN + d + 2)(N + 1)),$$

there exists $C_1 > 0$ such that

$$\begin{aligned} & \text{VCDim}(\mathcal{NN}(\# \text{input} = d; \text{width} \leq N; \text{depth} \leq L)) \\ & \leq C_1(LN + d + 2)(N + 1)L \ln((LN + d + 2)(N + 1)) \\ & := b_u(N, L). \end{aligned} \quad (2.3)$$

Then we will use Claim 2.4 to estimate a lower bound of

$$\text{VCDim}(\mathcal{NN}(\# \text{input} = d; \text{width} \leq N; \text{depth} \leq L)), \quad (2.4)$$

330 and this lower bound is asymptotically larger than $b_u(N, L)$, which leads to a contradic-
 331 tion.

332 More precisely, we will construct $\{f_\chi : \chi \in \mathcal{B}\} \subseteq \text{Höld}([0, 1]^d, \alpha, 1)$, which can
 333 shatter $b_\ell(N, L) := K^d$ points, where \mathcal{B} is a set defined later. Then by Claim 2.4,
 334 there exists $\{\phi_\chi : \chi \in \mathcal{B}\}$ such that this set can shatter $b_\ell(N, L)$ points. Finally,
 335 $b_\ell(N, L) = K^d = \lfloor (NL)^{2/d+\rho/(2\alpha)} \rfloor^d$ is asymptotically larger than $b_u(N, L) = C_1(LN +$
 336 $d + 2)(N + 1)L \ln((LN + d + 2)(N + 1))$, which leads to a contradiction. More details can
 337 be found below.

338 **Step 1:** Construct $\{f_\chi : \chi \in \mathcal{B}\} \subseteq \text{Höld}([0, 1]^d, \alpha, \lambda)$ that scatters $b_\ell(N, L)$ points.

339 Divide $[0, 1]^d$ into K^d non-overlapping sub-cubes $\{Q_\beta\}_\beta$ as follows:

$$340 \quad Q_\beta := \{\mathbf{x} = (x_1, x_2, \dots, x_d) \in [0, 1]^d : x_i \in [\frac{\beta_i-1}{K}, \frac{\beta_i}{K}], \ i = 1, 2, \dots, d\},$$

341 for any index vector $\beta = (\beta_1, \beta_2, \dots, \beta_d) \in \{1, 2, \dots, K\}^d$.

342 Let $Q(\mathbf{x}_0, \eta) \subseteq [0, 1]^d$ be a hypercube, whose center and sidelength are \mathbf{x}_0 and η ,
 343 respectively. Then we define a function ζ_Q on $[0, 1]^d$ corresponding to $Q = Q(\mathbf{x}_0, \eta) \subseteq$
 344 $[0, 1]^d$ such that:

- 345 • $\zeta_Q(\mathbf{x}_0) = (\eta/2)^\alpha/2$;
- 346 • $\zeta_Q(\mathbf{x}) = 0$ for any $\mathbf{x} \notin Q \setminus \partial Q$, where ∂Q is the boundary of Q ;
- 347 • ζ_Q is linear on the line that connects \mathbf{x}_0 and \mathbf{x} , for any $\mathbf{x} \in \partial Q$.

348 Define

$$349 \quad \mathcal{B} := \{\chi : \chi \text{ is a map from } \{1, 2, \dots, K\}^d \text{ to } \{-1, 1\}\}.$$

350 For each $\chi \in \mathcal{B}$, we define

$$351 \quad f_\chi(\mathbf{x}) := \sum_{\beta \in \{1, 2, \dots, K\}^d} \chi(\beta) \zeta_{Q_\beta}(\mathbf{x}),$$

352 where $\zeta_{Q_\beta}(\mathbf{x})$ is the associated function introduced just above. It is easy to check that
 353 $\{f_\chi : \chi \in \mathcal{B}\} \subseteq \text{Höld}([0, 1]^d, \alpha, 1)$ can shatter $b_\ell(N, L) = K^d$ points.

354 **Step 2:** Construct $\{\phi_\chi : \chi \in \mathcal{B}\}$ that scatters $b_\ell(N, L)$ points.

355 By Claim 2.4, there exist $\rho > 0$ and $C_2 > 0$ such that, for any $f_\chi \in \{f_\chi : \chi \in \mathcal{B}\}$ there
 356 exists $J_\chi > 0$ such that for all $N, L \in \mathbb{N}$ with $NL \geq J_\chi$, there exist $\phi_\chi \in \mathcal{NN}(\#input =$
 357 d ; width $\leq N$; depth $\leq L)$ and \mathcal{H}_χ with $\mu(\mathcal{H}_\chi) \leq 2^{-(d+K^d+1)}K^{-d}$ such that

$$358 \quad |f_\chi(\mathbf{x}) - \phi_\chi(\mathbf{x})| \leq C_2(NL)^{-\alpha(2/d+\rho/\alpha)}, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}_\chi.$$

359 Set $\mathcal{H} = \cup_{\chi \in \mathcal{B}} \mathcal{H}_\chi$ and $J_1 = \max_{\chi \in \mathcal{B}} J_\chi$. Then it holds that

$$360 \quad \mu(\mathcal{H}) \leq 2^{K^d} 2^{-(d+K^d+1)} K^{-d} = (2K)^{-d}/2. \quad (2.5)$$

361 It follows that for all $\chi \in \mathcal{B}$ and $N, L \in \mathbb{N}$ with $NL \geq J_1$, we have

$$362 \quad |f_\chi(\mathbf{x}) - \phi_\chi(\mathbf{x})| \leq C_2(NL)^{-\alpha(2/d+\rho/\alpha)}, \quad \text{for any } \mathbf{x} \in [0, 1]^d \setminus \mathcal{H}. \quad (2.6)$$

For each index vector $\beta \in \{1, 2, \dots, K\}^d$ and any $\mathbf{x} \in \frac{1}{2}Q_\beta$, where $\frac{1}{2}Q_\beta$ denotes the cube whose sidelength is half of that of Q_β sharing the same center of Q_β , since Q_β has a sidelength $\frac{1}{K}$, we have

$$|f_\chi(\mathbf{x})| = |\zeta_{Q_\beta}(\mathbf{x})| \geq |\zeta_{Q_\beta}(\mathbf{x}_{Q_\beta})|/2 = \left(\frac{1}{2K}\right)^\alpha / 4, \quad (2.7)$$

where \mathbf{x}_{Q_β} is the center of Q_β . For fixed d , α , and ρ , there exists $J_2 > 0$ large enough such that, for any $N, L \in \mathbb{N}$ with $NL \geq J_2$, we have

$$\frac{1}{2^{2+\alpha}} [(NL)^{2/d+\rho/(2\alpha)}]^{-\alpha} > C_2 (NL)^{-\alpha(2/d+\rho/\alpha)}. \quad (2.8)$$

By Equation (2.5), for any $\beta \in \{1, 2, \dots, K\}^d$, we have

$$\mu(\mathcal{H}) \leq (2K)^{-d}/2 < (2K)^{-d} = \mu(\frac{1}{2}Q_\beta),$$

which means $(\frac{1}{2}Q_\beta) \cap ([0, 1]^d \setminus \mathcal{H})$ is not empty. Therefore, there exists $\mathbf{x}_\beta \in (\frac{1}{2}Q_\beta) \cap ([0, 1]^d \setminus \mathcal{H})$ for each $\beta \in \{1, 2, \dots, K\}^d$ such that

$$|f_\chi(\mathbf{x}_\beta)| \geq \frac{1}{2^{2+\alpha}} [(NL)^{2/d+\rho/(2\alpha)}]^{-\alpha} > C_2 (NL)^{-\alpha(2/d+\rho/\alpha)} \geq |f_\chi(\mathbf{x}_\beta) - \phi_\chi(\mathbf{x}_\beta)|,$$

where the first, the second, and the last inequalities come from (2.7), (2.8), and (2.6), respectively. In other words, for any $\chi \in \mathcal{B}$ and $\beta \in \{1, 2, \dots, K\}^d$, $f_\chi(\mathbf{x}_\beta)$ and $\phi_\chi(\mathbf{x}_\beta)$ have the same sign. Then $\{\phi_\chi : \chi \in \mathcal{B}\}$ shatters $\{\mathbf{x}_\beta : \beta \in \{1, 2, \dots, K\}^d\}$ since $\{f_\chi : \chi \in \mathcal{B}\}$ shatters $\{\mathbf{x}_\beta : \beta \in \{1, 2, \dots, K\}^d\}$ as discussed in Step 1. Hence,

$$\text{VCDim}(\{\phi_\chi : \chi \in \mathcal{B}\}) \geq K^d = b_\ell(N, L). \quad (2.9)$$

Step 3: Contradiction.

By Equation (2.3) and (2.9), for any $N, L \in \mathbb{N}$ with $NL \geq J_0 = \max\{J_1, J_2\}$, we have

$$b_\ell(N, L) \leq \text{VCDim}(\{\phi_\chi : \chi \in \mathcal{B}\}) \leq \text{VCDim}(\mathcal{NN}(\#\text{input} = d; \text{width} \leq N; \text{depth} \leq L)) \leq b_u(N, L),$$

implying that

$$[(NL)^{2/d+\rho/(2\alpha)}]^d \leq C_1 (LN + d + 2)(N + 1)L \ln((LN + d + 2)(N + 1)),$$

which is a contradiction for sufficiently large $N, L \in \mathbb{N}$. So we finish the proof. \square

By Theorem 2.3, for any $\rho > 0$, the approximation rate cannot be better than $\mathcal{O}(N^{-(2\alpha/d+\rho)} L^{-(2/\alpha+\rho)})$, if we use FNNs in $\mathcal{NN}(\#\text{input} = d; \text{width} \leq N; \text{depth} \leq L)$ to approximate functions in $\text{Höld}([0, 1]^d, \alpha, \lambda)$. By a similar argument, we can show that the approximation rate cannot be $\mathcal{O}(N^{-2\alpha/d} L^{-(2/\alpha+\rho)})$ nor $\mathcal{O}(N^{-(2\alpha/d+\rho)} L^{-2\alpha/d})$. Hence, the approximation rate in Theorem 1.1 is nearly tight.

3 Proof of Theorem 2.1

In this section, we will prove Theorem 2.1. We first present the key ideas in Section 3.1. Based on two propositions in Section 3.1, the detailed proof is presented in Section 3.2. Finally, the proofs of two propositions in Section 3.1 can be found in Section 3.3 and 3.4.

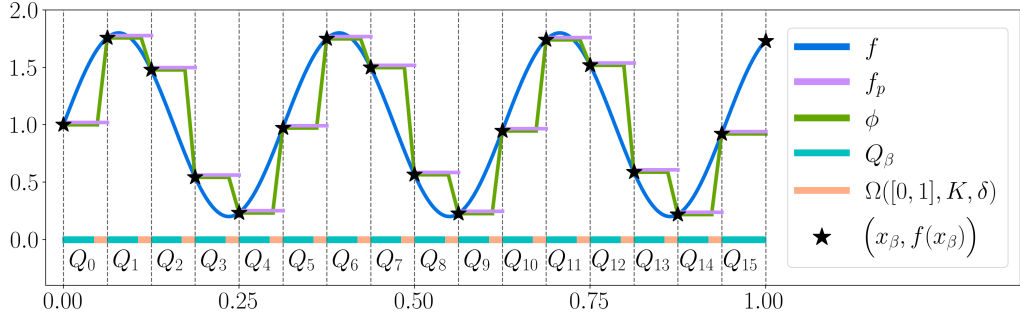


Figure 4: An illustration of f , f_p , ϕ , x_β , Q_β , and the trifling region $\Omega([0, 1]^d, K, \delta)$ in the one-dimensional case for $\beta \in \{0, 1, \dots, K-1\}^d$, where $K = N^2 L^2$ $d = 1$ with $N = 2$ and $L = 2$. f is the target function; f_p is the piecewise constant function approximating f ; ϕ is a function, implemented by a ReLU FNN, approximating f ; and x_β is a representative of Q_β . The measure of the trifling region $\Omega([0, 1]^d, K, \delta)$ can be arbitrarily small as we shall see in the proof of Theorem 1.1.

3.1 Key idea of proving Theorem 2.1

We will show that an almost piecewise constant function ϕ implemented by a ReLU FNN is enough to achieve the desired approximation rate in Theorem 1.1. Given an arbitrary $f \in C([0, 1]^d)$, we introduce a piecewise constant function $f_p \approx f$ serving as an intermediate approximant in our construction in the sense that

$$f \approx f_p \text{ on } [0, 1]^d, \quad \text{and} \quad f_p \approx \phi \text{ on } [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta).$$

The approximation in $f \approx f_p$ is a simple and standard technique in constructive approximation. For example, given arbitrary N and L , uniformly partition $[0, 1]^d$ into $\mathcal{O}(N^2 L^2)$ pieces and define f_p using this partition. Then the approximation error of $f_p \approx f$ scales like $\mathcal{O}(N^{-2/d} L^{-2/d})$. We will address the approximation in $f_p \approx \phi$ with the same error scaling and a limited budget of the FNN size, e.g., $\mathcal{O}(NL)$ neurons, based on the fact that f_p can be approximately implemented by a ReLU FNN in $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$, where $\Omega([0, 1]^d, K, \delta)$ is the trifling region near the discontinuous locations of f_p with an arbitrarily small Lebesgue measure (see Figure 4 for an illustration). The introduction of the trifling region is to ease the construction of a deep ReLU FNN to implement the desired ϕ , which is a piecewise linear and continuous function, to approximate the discontinuous function f_p by removing the difficulty near discontinuous points, essentially smoothing f_p by restricting the approximation domain in $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$.

Now let us discuss the detailed steps of construction. First, divide $[0, 1]^d$ into a union of important regions $\{Q_\beta\}_\beta$ and the trifling region $\Omega([0, 1]^d, K, \delta)$, where each Q_β is associated with a representative $\mathbf{x}_\beta \in Q_\beta$ such that $f(\mathbf{x}_\beta) = f_p(\mathbf{x}_\beta)$ for each index vector $\beta \in \{0, 1, \dots, K-1\}^d$, where $K = \mathcal{O}(N^{2/d} L^{2/d})$ is the partition number per dimension (see Figure 6 for examples for $d = 1$ and $d = 2$). Next, we design a vector function $\Phi_1(\mathbf{x})$ constructed via $\Phi_1(\mathbf{x}) = (\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d))$ to project the whole cube Q_β to a d -dimensional index β for each β , where each one-dimensional function ϕ_1 is a step function implemented by a ReLU FNN. The final step is to solve a point fitting problem. To be precise, we construct a function implemented by a ReLU FNN to map β approximately to $f_p(\beta) = f(\beta)$. Then $\phi_2 \circ \Phi_1(\mathbf{x}) = \phi_2(\beta) \approx f_p(\mathbf{x}_\beta) = f(\mathbf{x}_\beta)$

424 for any $\mathbf{x} \in Q_\beta$ and each β , implying $\phi := \phi_2 \circ \Phi_1 \approx f_p \approx f$ on $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$.
 425 We would like to point out that we only need to care about the values of ϕ_2 at a set of
 426 points $\{0, 1, \dots, K-1\}^d$ during construction, since we design $\phi = \phi_2 \circ \Phi_1$ via a function
 427 composition as shown in Figure 5. That means we are unnecessary to care about the
 428 values of ϕ_2 outside of the set of points $\{0, 1, \dots, K-1\}^d$, which is a key point while
 429 designing a ReLU FNN to implement ϕ_2 as we shall see later. This reveals the power of
 430 function compositions in a sense.

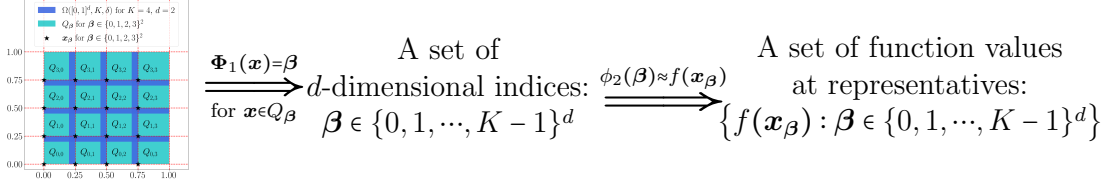


Figure 5: An illustration of the desired function $\phi = \phi_2 \circ \Phi_1$. Note that $\phi \approx f$ on $[0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$, since $\phi(\mathbf{x}) = \phi_2 \circ \Phi_1(\mathbf{x}) = \phi_2(\beta) \approx f(\mathbf{x}_\beta)$ for any $\mathbf{x} \in Q_\beta$ and each $\beta \in \{0, 1, \dots, K-1\}^d$.

431 Finally, we discuss how to implement Φ_1 and ϕ_2 by deep ReLU FNNs with width
 432 $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ using two propositions as we shall prove in Section 3.3 and 3.4
 433 later. We first construct a ReLU FNN with desired width and depth by Proposition 3.1
 434 to implement a one-dimensional step function ϕ_1 . Then Φ_1 can be attained via defining

$$435 \quad \Phi_1(\mathbf{x}) = (\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)), \quad \text{for any } \mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d.$$

436 **Proposition 3.1.** *For any $N, L, d \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{3K}]$ with $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$, there*
 437 *exists a one-dimensional function ϕ implemented by a ReLU FNN with width $4\lfloor N^{1/d} \rfloor + 3$*
 438 *and depth $4L + 5$ such that*

$$439 \quad \phi(x) = k, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}}\right] \text{ for } k = 0, 1, \dots, K-1.$$

440 The construction of ϕ_2 is a direct result of Proposition 3.2 below, the proof of which
 441 relies on the bit extraction technique in [5].

442 **Proposition 3.2.** *Given any $\varepsilon > 0$ and arbitrary $N, L, J \in \mathbb{N}^+$ with $J \leq N^2 L^2$, assume*
 443 *$\{y_j \geq 0 : j = 0, 1, \dots, J-1\}$ is a sample set with $|y_j - y_{j-1}| \leq \varepsilon$ for $j = 1, 2, \dots, J-1$. Then*
 444 *there exists $\phi \in \mathcal{NN}(\#input = 1; \text{width} \leq 12N + 8; \text{depth} \leq 4L + 9)$ such that*

$$445 \quad (i) \quad |\phi(j) - y_j| \leq \varepsilon \text{ for } j = 0, 1, \dots, J-1;$$

$$446 \quad (ii) \quad 0 \leq \phi(x) \leq \max\{y_j : j = 0, 1, \dots, J-1\} \text{ for any } x \in \mathbb{R}.$$

447 With the above propositions ready, let us prove Theorem 2.1 in Section 3.2. We
 448 further assume that $\omega_f(r) > 0$ for any $r > 0$, excluding a simple case when f is a constant
 449 function.

3.2 Proof of Theorem 2.1

We essentially construct an almost piecewise constant function implemented by a ReLU FNN with $\mathcal{O}(NL)$ neurons to approximate f . It is clear that $|f(\mathbf{x}) - f(\mathbf{0})| \leq \omega_f(\sqrt{d})$ for any $\mathbf{x} \in [0, 1]^d$. Define $\tilde{f} = f - f(\mathbf{0}) + \omega_f(\sqrt{d})$, then $0 \leq \tilde{f}(\mathbf{x}) \leq 2\omega_f(\sqrt{d})$ for any $\mathbf{x} \in [0, 1]^d$. Let $M = N^2L$, $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$, and δ be an arbitrary number in $(0, \frac{1}{3K}]$.

The proof can be divided into four steps as follows:

1. Divide $[0, 1]^d$ into a union of sub-cubes $\{Q_\beta\}_{\beta \in \{0, 1, \dots, K-1\}^d}$ and the trifling region $\Omega([0, 1]^d, K, \delta)$, and denote \mathbf{x}_β as the vertex of Q_β with minimum $\|\cdot\|_1$ norm
2. Construct a sub-network to implement a vector function Φ_1 projecting the whole cube Q_β to the d -dimensional index β for each β , i.e., $\Phi_1(\mathbf{x}) = \beta$ for all $\mathbf{x} \in Q_\beta$;
3. Construct a sub-network to implement a function ϕ_2 mapping the index β approximately to $\tilde{f}(\mathbf{x}_\beta)$. This core step can be further divided into three sub-steps:
 - 3.1. Construct a sub-network to implement ψ_1 bijectively mapping the index set $\{0, 1, \dots, K-1\}^d$ to an auxiliary set $\mathcal{A}_1 \subseteq \{\frac{j}{2K^d} : j = 0, 1, \dots, 2K^d\}$ defined later, see Figure 7 for a illustration;
 - 3.2. Determine a continuous piecewise linear function g with a set of breakpoints $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$ satisfying: 1) assign the values of g at breakpoints in \mathcal{A}_1 based on $\{\tilde{f}(\mathbf{x}_\beta)\}_\beta$, i.e., $g \circ \psi_1(\beta) = \tilde{f}(\mathbf{x}_\beta)$; 2) assign the values of g at breakpoints in $\mathcal{A}_2 \cup \{1\}$ to reduce the variation of g for applying Proposition 3.2;
 - 3.3. Apply Proposition 3.2 to construct a sub-network to implement a function ψ_2 approximating g well on $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$. Then the desired function ϕ_2 is given by $\phi_2 = \psi_2 \circ \psi_1$ satisfying $\phi_2(\beta) = \psi_2 \circ \psi_1(\beta) \approx g \circ \psi_1(\beta) = \tilde{f}(\mathbf{x}_\beta)$;
4. Construct the final target network to implement the desired function ϕ such that $\phi(\mathbf{x}) = \phi_2 \circ \Phi_1(\mathbf{x}) + f(\mathbf{0}) - \omega_f(\sqrt{d}) \approx \tilde{f}(\mathbf{x}_\beta) + f(\mathbf{0}) - \omega_f(\sqrt{d}) = f(\mathbf{x}_\beta)$ for $\mathbf{x} \in Q_\beta$.

The details of these steps can be found below.

Step 1: Divide $[0, 1]^d$ into $\{Q_\beta\}_{\beta \in \{0, 1, \dots, K-1\}^d}$ and $\Omega([0, 1]^d, K, \delta)$.

Define $\mathbf{x}_\beta := \beta/K$ and

$$Q_\beta = \left\{ \mathbf{x} := (x_1, \dots, x_d) \in [0, 1]^d : x_i \in \left[\frac{\beta_i}{K}, \frac{\beta_i+1}{K} - \delta \cdot 1_{\{\beta_i \leq K-2\}} \right], i = 1, \dots, d \right\}$$

for each d -dimensional index $\beta = (\beta_1, \beta_2, \dots, \beta_d) \in \{0, 1, \dots, K-1\}^d$. Recall that $\Omega([0, 1]^d, K, \delta)$ is the trifling region defined in Equation (2.1). Apparently, \mathbf{x}_β is the vertex of Q_β with minimum $\|\cdot\|_1$ norm and

$$[0, 1]^d = \left(\cup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_\beta \right) \cup \Omega([0, 1]^d, K, \delta),$$

see Figure 6 for illustrations.

Step 2: Construct Φ_1 mapping $\mathbf{x} \in Q_\beta$ to β .

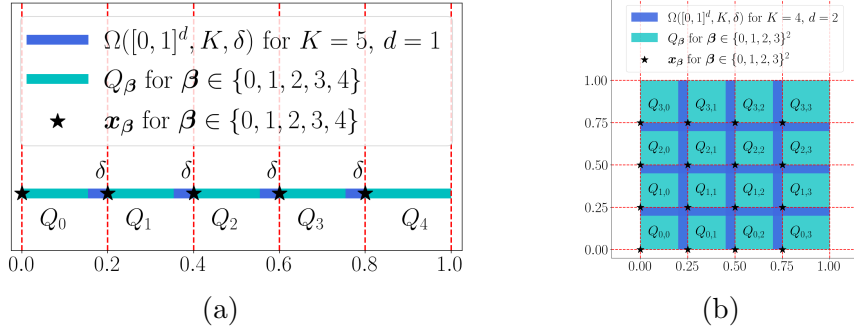


Figure 6: Illustrations of $\Omega([0, 1]^d, K, \delta)$, Q_β , and \mathbf{x}_β for $\beta \in \{0, 1, \dots, K-1\}^d$. (a) $K = 5$ and $d = 1$. (b) $K = 4$ and $d = 2$.

By Proposition 3.1, there exists $\phi_1 \in \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 3; \text{depth} \leq 4L + 5)$ such that

$$\phi_1(x) = \frac{k}{K}, \quad \text{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}}\right] \text{ for } k = 0, 1, \dots, K-1.$$

It follows that $\phi_1(x_i) = \beta_i$ if $\mathbf{x} = (x_1, x_2, \dots, x_d) \in Q_\beta$ for each $\beta = (\beta_1, \beta_2, \dots, \beta_d)$.

By defining

$$\Phi_1(\mathbf{x}) := (\phi_1(x_1), \phi_1(x_2), \dots, \phi_1(x_d)), \quad \text{for any } \mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d,$$

we have $\Phi_1(\mathbf{x}) = \beta$ if $\mathbf{x} \in Q_\beta$ for $\beta \in \{0, 1, \dots, K-1\}^d$.

Step 3: Construct ϕ_2 mapping β approximately to $\tilde{f}(\mathbf{x}_\beta)$.

The construction of the sub-network implementing ϕ_2 is essentially based on Proposition 3.2. To meet the requirements of applying Proposition 3.2, we first define two auxiliary set \mathcal{A}_1 and \mathcal{A}_2 as

$$\mathcal{A}_1 := \left\{ \frac{i}{K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \dots, K^{d-1} \text{ and } k = 0, 1, \dots, K-1 \right\}$$

and

$$\mathcal{A}_2 := \left\{ \frac{i}{K^{d-1}} + \frac{1}{2K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \dots, K^{d-1} \text{ and } k = 0, 1, \dots, K-1 \right\}.$$

Clearly, $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\} = \left\{ \frac{j}{2K^d} : j = 0, 1, \dots, 2K^d \right\}$ and $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$. See Figure 6 for an illustration of \mathcal{A}_1 and \mathcal{A}_2 . Next, we further divide this step into three sub-steps.

Step 3.1: Construct ψ_1 bijectively mapping $\{0, 1, \dots, K-1\}^d$ to \mathcal{A}_1 .

Inspired by the binary representation, we define

$$\psi_1(\mathbf{x}) := \frac{x_d}{2K^d} + \sum_{i=1}^{d-1} \frac{x_i}{K^i}, \quad \text{for any } \mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d. \quad (3.1)$$

Then ψ_1 is a linear function bijectively mapping the index set $\{0, 1, \dots, K-1\}^d$ to

$$\begin{aligned} & \left\{ \frac{\beta_d}{2K^d} + \sum_{i=1}^{d-1} \frac{\beta_i}{K^i} : \beta \in \{0, 1, \dots, K-1\}^d \right\} \\ & = \left\{ \frac{i}{K^{d-1}} + \frac{k}{2K^d} : i = 0, 1, \dots, K^{d-1} \text{ and } k = 0, 1, \dots, K-1 \right\} = \mathcal{A}_1. \end{aligned}$$

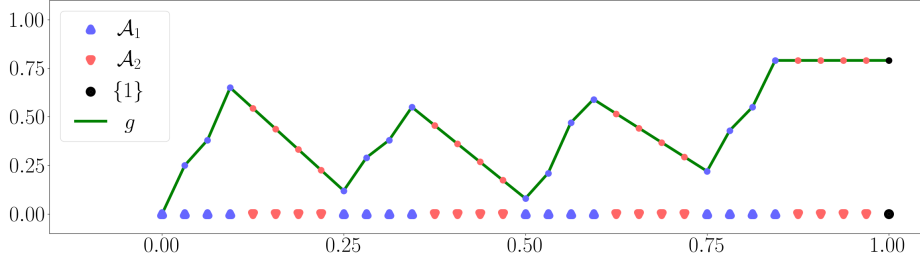


Figure 7: An illustration of \mathcal{A}_1 , \mathcal{A}_2 , $\{1\}$, and g for $d = 2$ and $K = 4$.

Step 3.2: Construct g to satisfy $g \circ \psi_1(\beta) = \tilde{f}(\mathbf{x}_\beta)$ and to meet the requirements of applying Proposition 3.2.

Let $g : [0, 1] \rightarrow \mathbb{R}$ be a continuous piecewise linear function with a set of breakpoints $\{\frac{j}{2K^d} : j = 0, 1, \dots, 2K^d\} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$ and the values of g at these breakpoints satisfy the following properties:

- The values of g at the breakpoints in \mathcal{A}_1 are set as

$$g(\psi_1(\beta)) = \tilde{f}(\mathbf{x}_\beta), \quad \text{for any } \beta \in \{0, 1, \dots, K-1\}^d; \quad (3.2)$$

- At the breakpoint 1, let $g(1) = \tilde{f}(\mathbf{1})$, where $\mathbf{1} = (1, 1, \dots, 1) \in \mathbb{R}^d$;
- The values of g at the breakpoints in $\mathcal{A}_2 \cup \{1\}$ are assigned to reduce the variation of g , which is a requirement of applying Proposition 3.2. Note that

$$\left\{ \frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}} \right\} \subseteq \mathcal{A}_1 \cup \{1\}, \quad \text{for } i = 1, 2, \dots, K^{d-1},$$

implying the values of g at $\frac{i}{K^{d-1}} - \frac{K+1}{2K^d}$ and $\frac{i}{K^{d-1}}$ have been assigned for $i = 1, 2, \dots, K^{d-1}$. Thus, the values of g at the breakpoints in \mathcal{A}_2 can be successfully assigned by letting g linear on each interval $[\frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}}]$ for $i = 1, 2, \dots, K^{d-1}$, since $\mathcal{A}_2 \subseteq \cup_{i=1}^{K^{d-1}} [\frac{i}{K^{d-1}} - \frac{K+1}{2K^d}, \frac{i}{K^{d-1}}]$.

Apparently, such a function g exists (see Figure 7 for an example) and satisfies

$$\left| g\left(\frac{j}{2K^d}\right) - g\left(\frac{j-1}{2K^d}\right) \right| \leq \max \left\{ \omega_f\left(\frac{1}{K}\right), \omega_f(\sqrt{d})/K \right\} \leq \omega_f\left(\frac{\sqrt{d}}{K}\right), \quad \text{for } j = 1, 2, \dots, 2K^d,$$

and

$$0 \leq g\left(\frac{j}{2K^d}\right) \leq 2\omega_f(\sqrt{d}), \quad \text{for } j = 0, 1, \dots, 2K^d.$$

Step 3.3: Construct ψ_2 approximating g well on $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$.

Since $2K^d = 2(\lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor) \leq 2(N^2 L^2) \leq N^2 \tilde{L}^2$, where $\tilde{L} = 2L$, by Proposition 3.2, there exists $\tilde{\psi}_2 \in \mathcal{NN}(\#input = 1; \text{width} \leq 12N + 8; \text{depth} \leq 4\tilde{L} + 9) = \mathcal{NN}(\#input = 1; \text{width} \leq 12N + 8; \text{depth} \leq 8L + 9)$ such that

$$|\tilde{\psi}_2(j) - g\left(\frac{j}{2K^d}\right)| \leq \omega_f\left(\frac{\sqrt{d}}{K}\right), \quad \text{for } j = 0, 1, \dots, 2K^d - 1,$$

530 and

$$531 \quad 0 \leq \tilde{\psi}_2(x) \leq \max\{g(\frac{j}{2K^d}) : j = 0, 1, \dots, 2K^d - 1\} \leq 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}.$$

532 By defining $\psi_2(x) := \tilde{\psi}_2(2K^d x)$ for any $x \in \mathbb{R}$, we have $\psi_2 = \psi_2 \circ \psi_1 \in \mathcal{NN}(\#input =$
 533 $1; \text{ width} \leq 12N + 8; \text{ depth} \leq 8L + 9),$

$$534 \quad 0 \leq \psi_2(x) = \tilde{\psi}_2(2K^d x) \leq 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}, \quad (3.3)$$

535 and

$$536 \quad |\psi_2(\frac{j}{2K^d}) - g(\frac{j}{2K^d})| = |\tilde{\psi}_2(j) - g(\frac{j}{2K^d})| \leq \omega_f(\frac{\sqrt{d}}{K}), \quad \text{for } j = 0, 1, \dots, 2K^d - 1. \quad (3.4)$$

537 Let us end Step 3 by defining the desired function ϕ_2 as $\phi_2 := \psi_2 \circ \psi_1$. Note that $\psi_1 : \mathbb{R}^d \rightarrow \mathbb{R}$ is a linear function and $\psi_2 \in \mathcal{NN}(\#input = 1; \text{ width} \leq 12N + 8; \text{ depth} \leq 8L + 9)$.
 538 Thus, $\phi_2 \in \mathcal{NN}(\#input = d; \text{ width} \leq 12N + 8; \text{ depth} \leq 8L + 9)$. By Equation (3.2) and
 539 (3.4), we have

$$541 \quad |\phi_2(\beta) - \tilde{f}(\mathbf{x}_\beta)| = |\psi_2(\psi_1(\beta)) - g(\psi_1(\beta))| \leq \omega_f(\frac{\sqrt{d}}{K}), \quad (3.5)$$

542 for any $\beta \in \{0, 1, \dots, K - 1\}^d$. Equation (3.3) and $\phi_2 = \psi_2 \circ \psi_1$ implies

$$543 \quad 0 \leq \phi_2(\mathbf{x}) \leq 2\omega_f(\sqrt{d}), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d. \quad (3.6)$$

544 **Step 4:** Construct the final network to implement the desired function ϕ .

545 Define $\phi := \phi_2 \circ \Phi_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$. Since $\phi_1 \in \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 3; \text{ depth} \leq$
 546 $4L + 5)$, we have $\Phi_1 \in \mathcal{NN}(\#input = d; \text{ width} \leq 4d\lfloor N^{1/d} \rfloor + 3d; \text{ depth} \leq 4L + 5; \#output =$
 547 $d)$. Note that $\phi_2 \in \mathcal{NN}(\#input = d; \text{ width} \leq 12N + 8; \text{ depth} \leq 8L + 9)$. Thus, $\phi =$
 548 $\phi_2 \circ \Phi_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$ is in

$$549 \quad \mathcal{NN}(\text{width} \leq \max\{4d\lfloor N^{1/d} \rfloor + 5d, 12N + 8\}; \text{ depth} \leq (4L + 5) + (8L + 9) = 12L + 14).$$

550 Now let us estimate the approximation error. Note that $f = \tilde{f} + f(\mathbf{0}) - \omega_f(\sqrt{d})$. By
 551 Equation (3.5), for any $\mathbf{x} \in Q_\beta$ and $\beta \in \{0, 1, \dots, K - 1\}^d$, we have

$$\begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &= |\tilde{f}(\mathbf{x}) - \phi_2(\Phi_1(\mathbf{x}))| = |\tilde{f}(\mathbf{x}) - \phi_2(\beta)| \\ &\leq |\tilde{f}(\mathbf{x}) - \tilde{f}(\mathbf{x}_\beta)| + |\tilde{f}(\mathbf{x}_\beta) - \phi_2(\beta)| \\ &\leq \omega_f(\frac{\sqrt{d}}{K}) + \omega_f(\frac{\sqrt{d}}{K}) \leq 2\omega_f(8\sqrt{d}N^{-2/d}L^{-2/d}), \end{aligned}$$

553 where the last inequality comes from the fact $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor \geq \frac{N^{2/d}L^{2/d}}{8}$ for any $N, L \in$
 554 \mathbb{N}^+ . Recall the fact $\omega_f(nr) \leq n\omega_f(r)$ for any $n \in \mathbb{N}^+$ and $r \in [0, \infty)$. Therefore, for any
 555 $\mathbf{x} \in \cup_{\beta \in \{0, 1, \dots, K-1\}^d} Q_\beta \subseteq [0, 1]^d \setminus \Omega([0, 1]^d, K, \delta)$, we have

$$\begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &\leq 2\omega_f(8\sqrt{d}N^{-2/d}L^{-2/d}) \leq 2\lceil 8\sqrt{d} \rceil \omega_f(N^{-2/d}L^{-2/d}) \\ &\leq 18\sqrt{d}\omega_f(N^{-2/d}L^{-2/d}). \end{aligned}$$

557 It remains to show the upper bound of ϕ . By Equation (3.6) and $\phi = \phi_2 \circ \Phi_1 + f(\mathbf{0}) -$
 558 $\omega_f(\sqrt{d})$, it holds that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \leq |f(\mathbf{0})| + \omega_f(\sqrt{d})$. Thus, we finish the proof.

3.3 Proof of Proposition 3.1

Lemma 3.3. For any $N_1, N_2 \in \mathbb{N}^+$, given $N_1(N_2 + 1) + 1$ samples $(x_i, y_i) \in \mathbb{R}^2$ with $x_0 < x_1 < \dots < x_{N_1(N_2+1)}$ and $y_i \geq 0$ for $i = 0, 1, \dots, N_1(N_2+1)$, there exists $\phi \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2N_1, 2N_2 + 1]; \#output = 1)$ satisfying the following conditions.

- (i) $\phi(x_i) = y_i$ for $i = 0, 1, \dots, N_1(N_2 + 1)$;
- (ii) ϕ is linear on each interval $[x_{i-1}, x_i]$ for $i \notin \{(N_2 + 1)j : j = 1, 2, \dots, N_1\}$.

In fact, Proposition 3.3 is a part of Lemma 2.2 in [62]. For the purpose of being self-contained, we present it as follows.

Lemma (Lemma 2.2 of [62]). For any $m, n \in \mathbb{N}^+$, given any $m(n+1) + 1$ samples $(x_i, y_i) \in \mathbb{R}^2$ with $x_0 < x_1 < x_2 < \dots < x_{m(n+1)}$ and $y_i \geq 0$ for $i = 0, 1, \dots, m(n+1)$, there exists $\phi \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2m, 2n + 1]; \#output = 1)$ satisfying the following conditions.

- (i) $\phi(x_i) = y_i$ for $i = 0, 1, \dots, m(n+1)$;
- (ii) ϕ is linear on each interval $[x_{i-1}, x_i]$ for $i \notin \{(n+1)j : j = 1, 2, \dots, m\}$;
- (iii) $\sup_{x \in [x_0, x_{m(n+1)}]} |\phi(x)| \leq 3 \max_{i \in \{0, 1, \dots, m(n+1)\}} y_i \prod_{k=1}^n \left(1 + \frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} \right)$.

Lemma 3.4. Given any $N, L, d \in \mathbb{N}^+$, it holds that

$$\begin{aligned} & \mathcal{NN}(\#input = d; \text{widthvec} = [N, NL]; \#output = 1) \\ & \subseteq \mathcal{NN}(\#input = d; \text{width} \leq 2N + 2; \text{depth} \leq L + 1; \#output = 1). \end{aligned}$$

Proof. The key idea to prove Proposition 3.4 is to re-assemble $\mathcal{O}(L)$ sub-FNNs in the shallower FNN in the left of Figure 8 to form a deeper one with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ on the right of Figure 8.

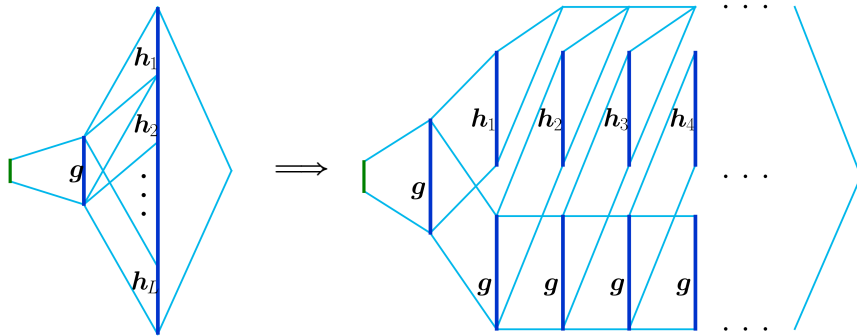


Figure 8: An illustration of the main idea to prove Lemma 3.4.

For any $\phi \in \mathcal{NN}(\#input = d; \text{widthvec} = [N, NL]; \#output = 1)$, ϕ can be implemented by a ReLU FNN described as

$$x \xrightarrow[\sigma]{W_0, b_0} g \xrightarrow[\sigma]{W_1, b_1} h \xrightarrow{W_2, b_2} \phi(x),$$

where \mathbf{g} and \mathbf{h} are the output of the first hidden layer and the second hidden layer, respectively. Note that

$$\mathbf{g} = \sigma(\mathbf{W}_0 \mathbf{x} + \mathbf{b}_0), \quad \mathbf{h} = \sigma(\mathbf{W}_1 \mathbf{g} + \mathbf{b}_1), \quad \text{and} \quad \phi(\mathbf{x}) = \mathbf{W}_2 \mathbf{h} + \mathbf{b}_2. \quad (3.7)$$

We can evenly divide $\mathbf{h} \in \mathbb{R}^{NL}$, $\mathbf{b}_1 \in \mathbb{R}^{NL}$, $\mathbf{W}_1 \in \mathbb{R}^{NL \times N}$, and $\mathbf{W}_2 \in \mathbb{R}^{1 \times NL}$ into L parts as follows:

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_L \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} \mathbf{b}_{1,1} \\ \mathbf{b}_{1,2} \\ \vdots \\ \mathbf{b}_{1,L} \end{bmatrix}, \quad \mathbf{W}_1 = \begin{bmatrix} \mathbf{W}_{1,1} \\ \mathbf{W}_{1,2} \\ \vdots \\ \mathbf{W}_{1,L} \end{bmatrix},$$

and $\mathbf{W}_2 = [\mathbf{W}_{2,1}, \mathbf{W}_{2,2}, \dots, \mathbf{W}_{2,L}]$, where $\mathbf{h}_\ell, \mathbf{b}_{1,\ell} \in \mathbb{R}^N$, $\mathbf{W}_{1,\ell} \in \mathbb{R}^{N \times N}$, and $\mathbf{W}_{2,\ell} \in \mathbb{R}^{1 \times N}$ for $\ell = 1, 2, \dots, L$.

Define

$$s_0 := 0, \quad \text{and} \quad s_\ell := \sum_{j=1}^{\ell} \mathbf{W}_{2,j} \cdot \mathbf{h}_j, \quad \text{for } \ell = 1, 2, \dots, L.$$

Then $\phi(\mathbf{x}) = \mathbf{W}_2 \cdot \mathbf{h} + \mathbf{b}_2 = s_L + \mathbf{b}_2$ and

$$s_\ell = s_{\ell-1} + \mathbf{W}_{2,\ell} \cdot \mathbf{h}_\ell, \quad \text{for } \ell = 1, 2, \dots, L. \quad (3.8)$$

Hence, it is easy to check that ϕ can be also implemented by the deep network shown in Figure 9. It is clear that the network has the architecture of Figure 9 is with width

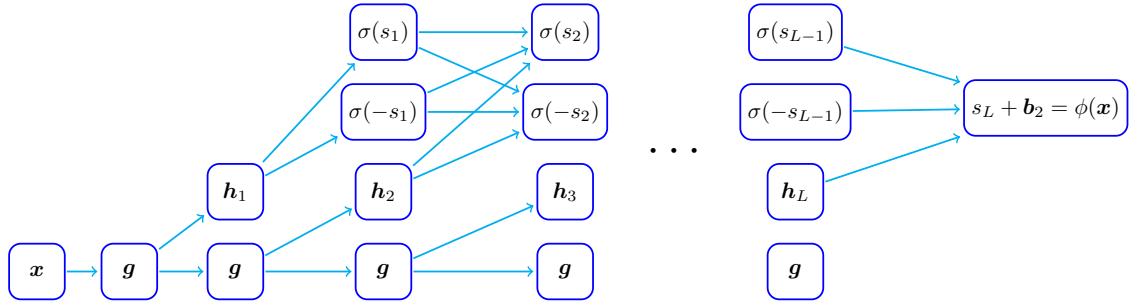


Figure 9: A illustration of the desired network based on Equation (3.7) and (3.8), and the fact $x = \sigma(x) - \sigma(-x)$ for any $x \in \mathbb{R}$. We omit the activation function (σ) if the input is non-negative.

2N + 2 and depth L + 1. So, we finish the proof. \square

With Lemma 3.3 and 3.4 in hand, we are ready to present the detailed proof of Proposition 3.1.

Proof of Proposition 3.1. We divide the proof into two cases: $d = 1$ and $d \geq 2$.

Case 1: $d = 1$.

In this case, $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor = N^2 L^2$. Denote $M = N^2 L$ and consider the sample set

$$\{(1, M-1), (2, 0)\} \cup \left\{ \left(\frac{m}{M}, m \right) : m = 0, 1, \dots, M-1 \right\} \cup \left\{ \left(\frac{m+1}{M} - \delta, m \right) : m = 0, 1, \dots, M-2 \right\}.$$

Its size is $2M + 1 = N \cdot ((2NL - 1) + 1) + 1$. By Lemma 3.3 (set $N_1 = N$ and $N_2 = 2NL - 1$ therein), there exists $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2(2NL - 1) + 1]) = \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1])$ such that

- $\phi_1(\frac{M-1}{M}) = \phi_1(1) = M - 1$ and $\phi_1(\frac{m}{M}) = \phi_1(\frac{m+1}{M} - \delta) = m$ for $m = 0, 1, \dots, M - 2$;
- ϕ_1 is linear on $[\frac{M-1}{M}, 1]$ and each interval $[\frac{m}{M}, \frac{m+1}{M} - \delta]$ for $m = 0, 1, \dots, M - 2$.

Then

$$\phi_1(x) = m, \quad \text{if } x \in [\frac{m}{M}, \frac{m+1}{M} - \delta \cdot 1_{\{m \leq M-2\}}], \quad \text{for } m = 0, 1, \dots, M - 1. \quad (3.9)$$

Now consider the another sample set

$$\{(\frac{1}{M}, L - 1), (2, 0)\} \cup \{(\frac{\ell}{ML}, \ell) : \ell = 0, 1, \dots, L - 1\} \cup \{(\frac{\ell+1}{ML} - \delta, \ell) : \ell = 0, 1, \dots, L - 2\}.$$

Its size is $2L + 1 = 1 \cdot ((2L - 1) + 1) + 1$. By Lemma 3.3 (set $N_1 = 1$ and $N_2 = 2L - 1$ therein), there exists $\phi_2 \in \mathcal{NN}(\text{widthvec} = [2, 2(2L - 1) + 1]) = \mathcal{NN}(\text{widthvec} = [2, 4L - 1])$ such that

- $\phi_2(\frac{L-1}{ML}) = \phi_2(\frac{1}{M}) = L - 1$ and $\phi_2(\frac{\ell}{ML}) = \phi_2(\frac{\ell+1}{ML} - \delta) = \ell$ for $\ell = 0, 1, \dots, L - 2$;
- ϕ_2 is linear on $[\frac{L-1}{ML}, \frac{1}{M}]$ and each interval $[\frac{\ell}{ML}, \frac{\ell+1}{ML} - \delta]$ for $\ell = 0, 1, \dots, L - 2$.

It follows that, for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$,

$$\phi_2(x - \frac{m}{M}) = \ell, \quad \text{for } x \in [\frac{mL+\ell}{ML}, \frac{mL+\ell+1}{ML} - \delta \cdot 1_{\{\ell \leq L-2\}}]. \quad (3.10)$$

The fact $K = ML$ implies each $k \in \{0, 1, \dots, K - 1\}$ can be unique represented by $k = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. Then the desired function ϕ can be implemented by a ReLU FNN shown in Figure 10. Clearly,

$$\phi(x) = k, \quad \text{if } x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot 1_{\{k \leq K-2\}}] \text{ for } k \in \{0, 1, \dots, K - 1\}.$$

By Lemma 3.4, $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 4NL - 1]) \subseteq \mathcal{NN}(\text{width} \leq 4N + 2; \text{depth} \leq 2L + 1)$ and $\phi_2 \in \mathcal{NN}(\text{widthvec} = [2, 4L - 1]) \subseteq \mathcal{NN}(\text{width} \leq 6; \text{depth} \leq 2L + 1)$, implying $\phi \in \mathcal{NN}(\text{width} \leq \max\{4N + 2 + 1, 6 + 1\} = 4N + 3; \text{depth} \leq (2L + 1) + 2 + (2L + 1) + 1 = 4L + 5)$. So we finish the proof for the case $d = 1$

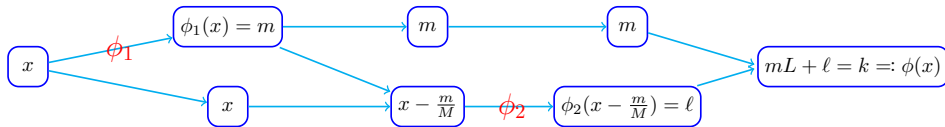


Figure 10: An illustration of the ReLU FNN implementing ϕ based on Equation (3.9) and (3.10) with $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot 1_{\{k \leq K-2\}}] = [\frac{mL+\ell}{ML}, \frac{mL+\ell+1}{ML} - \delta \cdot 1_{\{m \leq M-2 \text{ or } \ell \leq L-2\}}]$, where $k = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. “ ϕ_1 ” and “ ϕ_2 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

627

628 **Case 2:** $d \geq 2$.

Now we consider the case when $d \geq 2$. Consider the sample set

$$\{(1, K-1), (2, 0)\} \cup \left\{ \left(\frac{k}{K}, \frac{k}{K} \right) : k = 0, 1, \dots, K-1 \right\} \cup \left\{ \left(\frac{k+1}{K} - \delta, k \right) : k = 0, 1, \dots, K-2 \right\},$$

whose size is $2K+1 = \lfloor N^{1/d} \rfloor ((2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1) + 1) + 1$. By Lemma 3.3 (set $N_1 = \lfloor N^{1/d} \rfloor$ and $N_2 = 2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1$ therein), there exists ϕ in

$$\begin{aligned} & \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 2(2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1) + 1]) \\ & \subseteq \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1]) \end{aligned}$$

such that

- $\phi(\frac{K-1}{K}) = \phi(1) = K-1$, and $\phi(\frac{k}{K}) = \phi(\frac{k+1}{K} - \delta) = k$ for $k = 0, 1, \dots, K-2$;
- ϕ is linear on $[\frac{K-1}{K}, 1]$ and each interval $[\frac{k}{K}, \frac{k+1}{K} - \delta]$ for $k = 0, 1, \dots, K-2$.

Then

$$\phi(x) = k, \quad \text{if } x \in [\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k \leq K-2\}}] \text{ for } k = 0, 1, \dots, K-1.$$

By Lemma 3.4,

$$\begin{aligned} & \phi \in \mathcal{NN}(\text{widthvec} = [2\lfloor N^{1/d} \rfloor, 4\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1]) \\ & = \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 2; \text{depth} \leq 2\lfloor L^{2/d} \rfloor + 1) \\ & \subseteq \mathcal{NN}(\text{width} \leq 4\lfloor N^{1/d} \rfloor + 3; \text{depth} \leq 4L + 5). \end{aligned}$$

which means we finish the proof for the case $d \geq 2$. □

3.4 Proof of Proposition 3.2

The proof of Proposition 3.2 is based on the bit extraction technique in [5, 27]. In fact, we modify this technique to extract the sum of many bits rather than one bit and this modification can be summarized in Lemma 3.5 and 3.6 below.

Lemma 3.5. *For any $L \in \mathbb{N}^+$, there exists a function ϕ in*

$$\mathcal{NN}(\#input = 2; \text{width} \leq 7; \text{depth} \leq 2L + 1; \#output = 1)$$

such that, for any $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$, we have

$$\phi(\text{bin}0.\theta_1\theta_2\cdots\theta_L, \ell) = \sum_{j=1}^{\ell} \theta_j, \quad \text{for } \ell = 1, 2, \dots, L.$$

Proof. Given $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$, define

$$\xi_j := \text{bin}0.\theta_j\theta_{j+1}\cdots\theta_L, \quad \text{for } j = 1, 2, \dots, L$$

and

$$\mathcal{T}(x) := \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

654 Then we have

$$655 \quad \theta_j = \mathcal{T}(\xi_j - 1/2), \quad \text{for } j = 1, 2, \dots, L,$$

656 and

$$657 \quad \xi_{j+1} = 2\xi_j - \theta_j, \quad \text{for } j = 1, 2, \dots, L-1.$$

658 I would like to point out that, by above two iteration equations, we can iteratively get
 659 $\xi_1, \theta_1, \xi_2, \theta_2, \dots, \xi_L, \theta_L$ when ξ_1 is given. Based on this iteration idea, the rest proof can be
 660 divided into three steps.

661 **Step 1:** Simplify the iteration equations.

662 Note that $\mathcal{T}(x) = \sigma(x/\delta + 1) - \sigma(x/\delta)$ for any $x \notin (-\delta, 0)$. By setting $\delta = 1/2 - \sum_{j=2}^L 2^{-j} =$
 663 2^{-L} , we have $\xi_j - 1/2 \notin (-\delta, 0)$ for all j , implying

$$664 \quad \begin{aligned} \theta_j &= \mathcal{T}(\xi_j - 1/2) = \sigma((\xi_j - 1/2)/\delta + 1) - \sigma((\xi_j - 1/2)/\delta) \\ &= \sigma(\mathcal{L}(\xi_j) + 1) - \sigma(\mathcal{L}(\xi_j)), \end{aligned} \quad (3.11)$$

665 for $j = 1, 2, \dots, L$, where \mathcal{L} is the linear map given by $\mathcal{L}(x) = (x - 1/2)/\delta$. It follows that,
 666 for $j = 1, 2, \dots, L-1$,

$$667 \quad \xi_{j+1} = 2\xi_j - \theta_j = 2\xi_j - \sigma(\mathcal{L}(\xi_j) + 1) + \sigma(\mathcal{L}(\xi_j)). \quad (3.12)$$

668 **Step 2:** Design a ReLU FNN to output $\sum_{j=1}^L \theta_j$.

669 It is easy to design a ReLU FNN to output $\theta_1, \theta_2, \dots, \theta_L$ by Equation (3.11) and
 670 (3.12) when using $\xi_1 = \text{bin}0.\theta_1\theta_2\dots\theta_L$ as the input. However, it is highly non-trivial to
 671 construct a ReLU FNN to output $\sum_{j=1}^L \theta_j$ with another input ℓ , since many operations
 672 like multiplication and comparison are not allowed in designing ReLU FNNs.

673 Now let us establish a formula to represent $\sum_{j=1}^L \theta_j$ in a form of a ReLU FNN as
 674 follows:

675 The fact that $x_1x_2 = \sigma(x_1 + x_2 - 1)$ for any $x_1, x_2 \in \{0, 1\}$ implies

$$676 \quad \begin{aligned} \sum_{j=1}^L \theta_j &= \sum_{j=1}^L \theta_j \mathcal{T}(\ell - j) = \sum_{j=1}^L \sigma(\theta_j + \mathcal{T}(\ell - j) - 1) \\ &= \sum_{j=1}^L \sigma(\theta_j + \sigma(\ell - j + 1) - \sigma(\ell - j) - 1), \end{aligned}$$

677 for $\ell = 1, 2, \dots, L$, where the last equality comes from the fact $\mathcal{T}(n) = \sigma(n + 1) - \sigma(n)$ for
 678 any integer n .

679 To simplify the notations, we define

$$680 \quad z_{\ell,j} := \sigma(\theta_j + \sigma(\ell - j + 1) - \sigma(\ell - j) - 1), \quad (3.13)$$

681 for $\ell = 1, 2, \dots, L$ and $j = 1, 2, \dots, L$. Then,

$$682 \quad \sum_{j=1}^L \theta_j = \sum_{j=1}^L z_{\ell,j}, \quad \text{for } \ell = 1, 2, \dots, L. \quad (3.14)$$

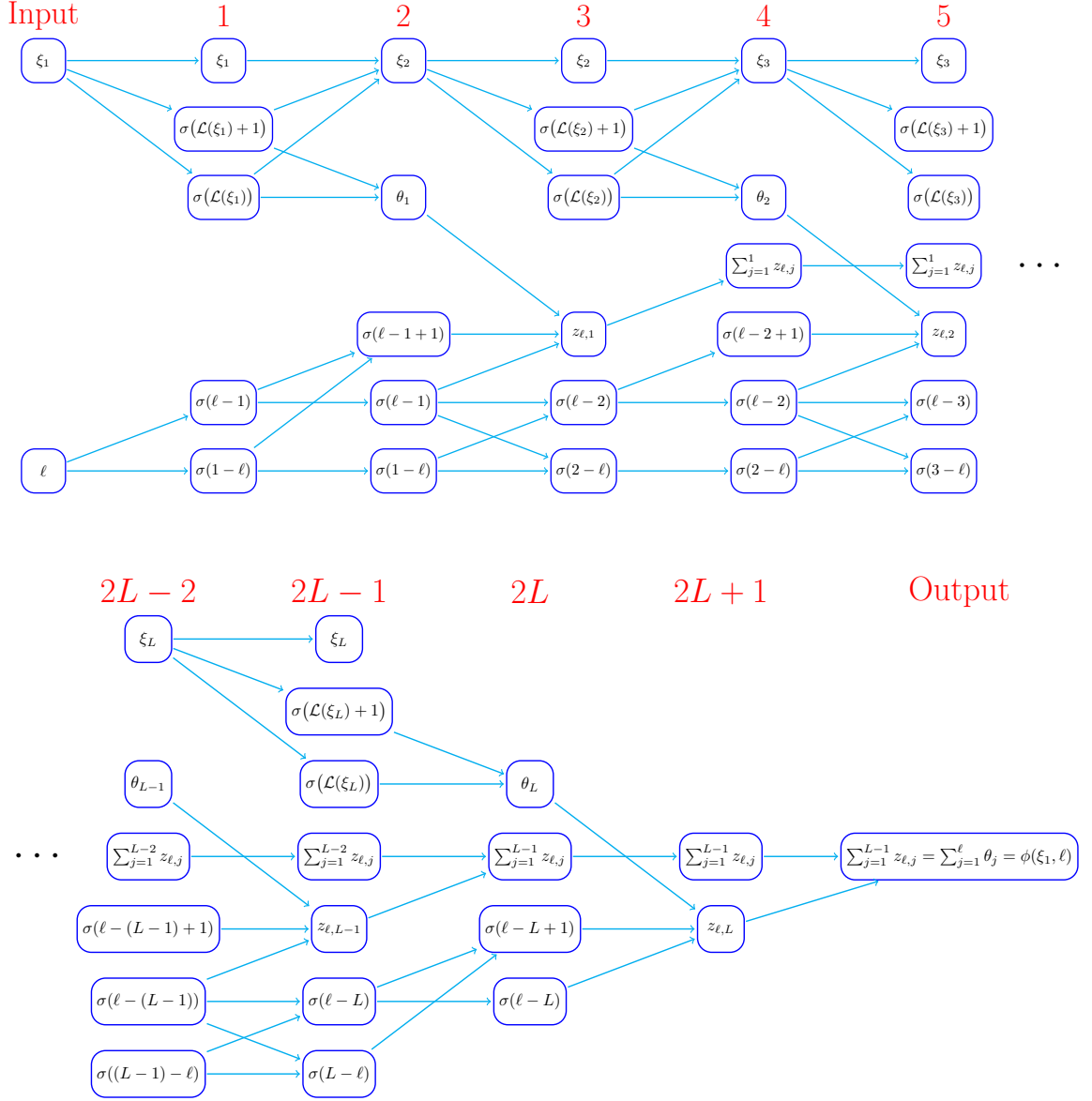


Figure 11: A illustration of the target ReLU FNN implementing ϕ to output $\sum_{j=1}^L z_{j,\ell} = \sum_{j=1}^L \theta_j = \phi(\xi_1, \ell)$ given the input $(\xi_1, \ell) = (\text{bin}0.\theta_1\theta_2\cdots\theta_L, \ell)$ for $\ell \in \{1, 2, \dots, L\}$ and $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$. The construction is mainly based on Equation (3.11), (3.12), (3.13), and (3.14). The numbers above the architecture indicate the order of hidden layers. It builds a whole iteration step for every two layers. We output both $\sigma(\ell - j)$ and $\sigma(j - \ell)$ in the hidden layers for $j = 1, 2, \dots, L$ because of the fact $x = \sigma(x) - \sigma(-x)$ for any $x \in \mathbb{R}$. We omit the activation function (σ) if the input of a neuron is non-negative. Note that the parameters are independent of $\theta_1, \theta_2, \dots, \theta_L$ when using this network architecture to implement the desired function ϕ .

With Equation (3.11), (3.12), (3.14), and (3.13) in hand, it is easy to construct a function ϕ implemented by a ReLU FNN with the desired width and depth outputting $\sum_{j=1}^{\ell} \theta_j = \sum_{j=1}^L z_{\ell,j}$ given the input $(\xi_1, \ell) = (\text{bin}0.\theta_1\theta_2\cdots\theta_L, \ell)$ for $\ell \in \{1, 2, \dots, L\}$ and $\theta_1, \theta_2, \dots, \theta_L \in \{0, 1\}$. The details of construction is shown in Figure 11. Clearly, the network in Figure 11 is with width 7 and depth $2L + 1$, which implies

$$\phi \in \mathcal{NN}(\#input = 2; \text{width} \leq 7; \text{depth} \leq 2L + 1; \#output = 1).$$

So we finish the proof. \square

Next, we introduce Lemma 3.6 as an advanced version of Lemma 3.5.

Lemma 3.6. *For any $N, L \in \mathbb{N}^+$, any $\theta_{m,\ell} \in \{0, 1\}$ for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$, where $M = N^2L$, there exists a function ϕ implemented by a ReLU FNN with width $4N + 3$ and depth $3L + 3$ such that*

$$\phi(m, \ell) = \sum_{j=0}^{\ell} \theta_{m,j}, \quad \text{for } m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1.$$

Proof. Define

$$y_m := \text{bin}0.\theta_{m,0}\theta_{m,1}\cdots\theta_{m,L-1}, \quad \text{for } m = 0, 1, \dots, M-1.$$

Consider the sample set $\{(m, y_m) : m = 0, 1, \dots, M\}$, whose cardinality is $M+1 = N((NL-1) + 1) + 1$. By Lemma 3.3 (set $N_1 = N$ and $N_2 = NL - 1$ therein), there exists

$$\begin{aligned} \phi_1 &\in \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 2(NL-1) + 1]) \\ &= \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 2NL-1]) \end{aligned}$$

such that

$$\phi_1(m) = y_m, \quad \text{for } m = 0, 1, \dots, M-1.$$

By Lemma 3.5, there exists

$$\phi_2 \in \mathcal{NN}(\#input = 2; \text{width} \leq 7; \text{depth} \leq 2L + 1)$$

such that, for any $\xi_1, \xi_2, \dots, \xi_L \in \{0, 1\}$, we have

$$\phi_2(\text{bin}0.\xi_1\xi_2\cdots\xi_L, \ell) = \sum_{j=1}^{\ell} \xi_j, \quad \text{for } \ell = 1, 2, \dots, L.$$

It follows that, for any $\xi_0, \xi_1, \dots, \xi_{L-1} \in \{0, 1\}$, we have

$$\phi_2(\text{bin}0.\xi_0\xi_1\cdots\xi_{L-1}, \ell + 1) = \sum_{j=0}^{\ell} \xi_j, \quad \text{for } \ell = 0, 1, \dots, L-1.$$

Thus, for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$, we have

$$\phi_2(\phi_1(m), \ell + 1) = \phi_2(y_m, \ell + 1) = \phi_2(0.\theta_{m,0}\theta_{m,1}\cdots\theta_{m,L-1}, \ell + 1) = \sum_{j=0}^{\ell} \theta_{m,j}.$$

Hence, the desired function ϕ can be implemented by the network shown in Figure 12. By Lemma 3.4, $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2NL-1]) \subseteq \mathcal{NN}(\text{width} \leq 4N + 2; \text{depth} \leq L + 1)$, implying the network in Figure 12 is with width $\max\{(4N + 2) + 1, 7\} = 4N + 3$ and depth $(2L + 1) + 1 + (L + 1) = 3L + 3$. So we finish the proof. \square

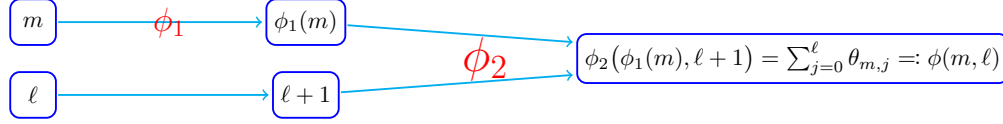


Figure 12: A illustration of the network implementing the desired function ϕ . “ ϕ_1 ” and “ ϕ_2 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

Next, we apply Lemma 3.6 to prove Lemma 3.7 below, which is a key intermediate conclusion to prove Proposition 3.2.

Lemma 3.7. For any $\varepsilon > 0$, $L, N \in \mathbb{N}^+$, denote $M = N^2 L$ and assume $\{y_{m,\ell} \geq 0 : m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1\}$ is sample set with

$$|y_{m,\ell} - y_{m,\ell-1}| \leq \varepsilon, \quad \text{for } m = 0, 1, \dots, M-1 \text{ and } \ell = 1, 2, \dots, L-1.$$

Then there exists $\phi \in \mathcal{NN}(\#input = 2; \text{width} \leq 12N + 8; \text{depth} \leq 3L + 6)$ such that

(i) $|\phi(m, \ell) - y_{m,\ell}| \leq \varepsilon$, for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$;

(ii) $0 \leq \phi(x_1, x_2) \leq \max\{y_{m,\ell} : m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1\}$, for any $x_1, x_2 \in \mathbb{R}$.

Proof. Define

$$a_{m,\ell} := \lfloor y_{m,\ell} / \varepsilon \rfloor, \quad \text{for } m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1.$$

We will construct a function implemented by a ReLU FNN to map the index (m, ℓ) to $a_{m,\ell} \varepsilon$ for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$.

Define $b_{m,0} := 0$ and $b_{m,\ell} := a_{m,\ell} - a_{m,\ell-1}$ for $m = 0, 1, \dots, M-1$ and $\ell = 1, \dots, L-1$. Since $|y_{m,\ell} - y_{m,\ell-1}| \leq \varepsilon$ for all m and ℓ , we have $b_{m,\ell} \in \{-1, 0, 1\}$. Hence, there exist $c_{m,\ell}$ and $d_{m,\ell} \in \{0, 1\}$ such that $b_{m,\ell} = c_{m,\ell} - d_{m,\ell}$, which implies

$$\begin{aligned} a_{m,\ell} &= a_{m,0} + \sum_{j=1}^{\ell} (a_{m,j} - a_{m,j-1}) = a_{m,0} + \sum_{j=1}^{\ell} b_{m,j} = a_{m,0} + \sum_{j=0}^{\ell} b_{m,j} \\ &= a_{m,0} + \sum_{j=0}^{\ell} c_{m,j} - \sum_{j=0}^{\ell} d_{m,j}. \end{aligned}$$

for $m = 0, 1, \dots, M-1$ and $\ell = 1, \dots, L-1$.

For the sample set $\{(m, a_{m,0}) : m = 0, 1, \dots, M-1\} \cup \{(M, 0)\}$, whose size is $M+1 = N \cdot ((NL-1)+1) + 1$, by Lemma 3.3 (set $N_1 = N$ and $N_2 = NL-1$ therein), there exists $\psi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2(NL-1)+1]) = \mathcal{NN}(\text{widthvec} = [2N, 2NL-1])$ such that

$$\psi_1(m) = a_{m,0}, \quad \text{for } m = 0, 1, \dots, M-1.$$

By Lemma 3.6, there exist $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \leq 4N + 3; \text{depth} \leq 3L + 3)$ such that

$$\psi_2(m, \ell) = \sum_{j=0}^{\ell} c_{m,j} \quad \text{and} \quad \psi_3(m, \ell) = \sum_{j=0}^{\ell} d_{m,j},$$

for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$. Hence, it holds that

$$a_{m,\ell} = a_{m,0} + \sum_{j=0}^{\ell} c_{m,j} - \sum_{j=0}^{\ell} d_{m,j} = \psi_1(m) + \psi_2(m, \ell) - \psi_3(m, \ell), \quad (3.15)$$

for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$.

Define

$$y_{\max} := \max\{y_{m,\ell} : m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1\}.$$

Then the desired function can be implemented by two sub-networks shown in Figure 13.

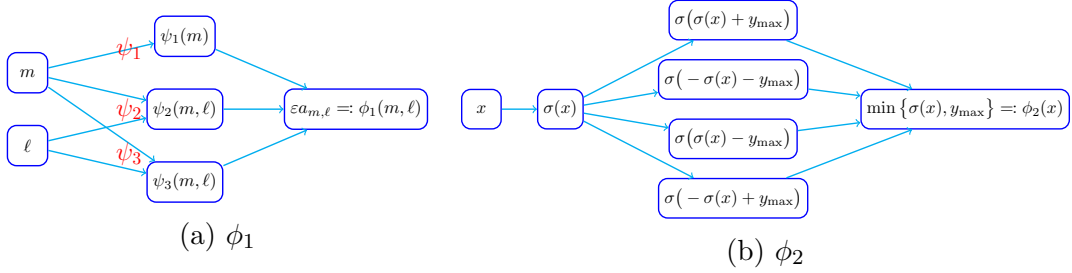


Figure 13: Illustrations of two sub-networks implementing the desired function $\phi = \phi_2 \circ \phi_1$ based Equation (3.15) and the fact $\min\{x_1, x_2\} = \frac{x_1 + x_2 - |x_1 - x_2|}{2} = \frac{\sigma(x_1 + x_2) - \sigma(-x_1 - x_2) - \sigma(x_1 - x_2) - \sigma(-x_1 + x_2)}{2}$. y_{\max} is given by $\max\{y_{m,\ell} : m = 0, 1, \dots, M-1 \text{ and } \ell = 0, 1, \dots, L-1\}$. “ ψ_1 ”, “ ψ_2 ”, and “ ψ_3 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

By Lemma 3.4, $\psi_1 \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 2NL - 1]) \subseteq \mathcal{NN}(\#input = 1; \text{width} \leq 4N + 2; \text{depth} \leq L + 1)$. Note that $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \leq 4N + 3; \text{depth} \leq 3L + 3)$. Thus, $\phi_1 \in \mathcal{NN}(\text{width} \leq (4N + 2) + 2(4N + 3) = 12N + 8; \text{depth} \leq (3L + 3) + 1 = 3L + 4)$ as shown in Figure 13. And it is clear that $\phi_2 \in \mathcal{NN}(\text{width} \leq 4; \text{depth} \leq 2)$, implying $\phi = \phi_2 \circ \phi_1 \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq (3L + 4) + 2 = 3L + 6)$.

Clearly, $0 \leq \phi(x_1, x_2) \leq y_{\max}$ for any $x_1, x_2 \in \mathbb{R}$, since $\phi(x_1, x_2) = \phi_2 \circ \phi_1(x_1, x_2) = \max\{\sigma(\phi_1(x_1, x_2)), y_{\max}\}$.

Note that $0 \leq \varepsilon a_{m,\ell} = \varepsilon \lfloor y_{m,\ell} / \varepsilon \rfloor \leq y_{\max}$. Then we have $\phi(m, \ell) = \phi_2 \circ \phi_1(m, \ell) = \phi_2(\varepsilon a_{m,\ell}) = \max\{\sigma(\varepsilon a_{m,\ell}), y_{\max}\} = \varepsilon a_{m,\ell}$. Therefore,

$$|\phi(m, \ell) - y_{m,\ell}| = |\varepsilon a_{m,\ell} - y_{m,\ell}| = \left| \lfloor y_{m,\ell} / \varepsilon \rfloor \varepsilon - y_{m,\ell} \right| \leq \varepsilon,$$

for $m = 0, 1, \dots, M-1$ and $\ell = 0, 1, \dots, L-1$. Hence, we finish the proof. \square

Finally, we apply Lemma 3.7 to prove Proposition 3.2.

Proof of Proposition 3.2. Let $M = N^2L$, then we may assume $J = ML$ since we can set $y_{J-1} = y_J = y_{J+1} = \dots = y_{ML-1}$ if $J < ML$.

For the sample set

$$\{(mL, m) : m = 0, 1, \dots, M\} \cup \{(mL + L - 1, m) : m = 0, 1, \dots, M - 1\},$$

whose size is $2M + 1 = N \cdot ((2NL - 1) + 1) + 1$, by Lemma 3.3 (set $N_1 = N$ and $N_2 = NL - 1$ therein), there exist $\phi_1 \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 2(2NL - 1) + 1]) = \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 4NL - 1])$ such that

- $\phi_1(ML) = M$ and $\phi_1(mL) = \phi_1(mL + L - 1) = m$ for $m = 0, 1, \dots, M - 1$;
- ϕ_1 is linear on each interval $[mL, mL + L - 1]$ for $m = 0, 1, \dots, M - 1$.

It follows that

$$\phi_1(j) = m, \quad \text{and} \quad j - L\phi_1(j) = \ell, \quad \text{where } j = mL + \ell, \quad (3.16)$$

for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$.

Note that any number j in $\{0, 1, \dots, J - 1\}$ can be uniquely indexed as $j = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. So we can denote $y_j = y_{mL + \ell}$ as $y_{m, \ell}$. Then by Lemma 3.7, there exists $\phi_2 \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq 3L + 6)$ such that

$$|\phi_2(m, \ell) - y_{m, \ell}| \leq \varepsilon, \quad \text{for } m = 0, 1, \dots, M - 1 \text{ and } \ell = 0, 1, \dots, L - 1, \quad (3.17)$$

and

$$0 \leq \phi_2(x_1, x_2) \leq y_{\max}, \quad \text{for any } x_1, x_2 \in \mathbb{R}, \quad (3.18)$$

where $y_{\max} := \max\{y_{m, \ell} : m = 0, 1, \dots, M - 1 \text{ and } \ell = 0, 1, \dots, L - 1\} = \max\{y_j : j = 0, 1, \dots, ML - 1\}$.

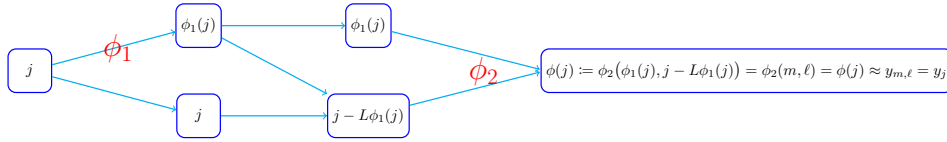


Figure 14: A illustration of the ReLU FNN implementing the desired function ϕ based Equation (3.16). The index $j \in \{0, 1, \dots, ML - 1\}$ is unique represented by $j = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$. “ ϕ_1 ” and “ ϕ_2 ” near “ \rightarrow ” represent the respective ReLU FNN implementing itself. We omit the activation function ReLU if the input of a neuron is non-negative.

Note that $\phi_1 \in \mathcal{NN}(\#input = 1; \text{widthvec} = [2N, 4NL - 1]) \subseteq \mathcal{NN}(\#input = 1; \text{width} \leq 8N + 2; \text{depth} \leq L + 1)$ by Lemma 3.4 and $\phi_2 \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq 3L + 6)$. So $\phi \in \mathcal{NN}(\text{width} \leq 12N + 8; \text{depth} \leq (L + 1) + 2 + (3L + 6) = 4L + 9)$ as shown in Figure 14.

Equation (3.18) implies

$$0 \leq \phi(x) \leq y_{\max}, \quad \text{for any } x \in \mathbb{R},$$

Since ϕ is given by $\phi(x) = \phi_2(\phi_1(x), x - L\phi_1(x))$.

Represent $j \in \{0, 1, \dots, ML - 1\}$ via $j = mL + \ell$ for $m = 0, 1, \dots, M - 1$ and $\ell = 0, 1, \dots, L - 1$, then we have, by Equation (3.17),

$$|\phi(j) - y_j| = |\phi_2(\phi_1(j), j - L\phi_1(j)) - y_j| = |\phi_2(m, \ell) - y_{m, \ell}| \leq \varepsilon.$$

So we finish the proof. \square

We would like to remark that the key idea in the proof of Proposition 3.2 is the bit extraction technique in Lemma 3.5, which allows us to store L bits in a binary number $\text{bin}0.x_1x_2\dots x_L$ and extract each bit x_i . The extraction operator can be efficiently carried out via a deep ReLU neural network demonstrating the power of depth.

4 Neural networks in practice

This section is concerned with neural networks in practice, e.g., approximating functions defined on irregular domains or domains with a low-dimensional structure, and neural network computation in parallel computing.

4.1 Approximation on irregular domain

In this section, we consider approximating continuous functions defined on irregular domains by deep ReLU FNNs. The construction is through extending the target function to a cubic domain, applying Theorem 1.1, and finally restricting the constructed FNN back to the irregular domain.

Given any uniformly continuous and real-valued function f defined on a metric space S with a metric $d_S(\cdot, \cdot)$, we define the (optimal) modulus of continuity of f on a subset $E \subseteq S$ as

$$\omega_f^E(r) := \sup\{|f(\mathbf{x}_1) - f(\mathbf{x}_2)| : d_S(\mathbf{x}_1, \mathbf{x}_2) \leq r, \mathbf{x}_1, \mathbf{x}_2 \in E\}, \quad \text{for any } r \geq 0.$$

For the purpose of consistency and simplicity, $\omega_f(\cdot)$ is short of $\omega_f^{[0,1]^d}(\cdot)$.

First, let us present two lemmas for (approximately) extending (almost) continuous functions on E to (almost) continuous functions on S . These lemmas are similar to the well-known results for extending Lipschitz or differentiable functions in [47, 64]. We generalize these results to a broader class of functions required in the proof of Theorem 4.3.

Lemma 4.1 (Approximate Extension of Almost-Continuous Functions). *Assume S is a metric space with a metric $d_S(\cdot, \cdot)$ and $\omega : [0, \infty) \rightarrow [0, \infty)$ is an increasing function with*

$$\omega(r_1 + r_2) \leq \omega(r_1) + \omega(r_2), \quad \text{for any } r_1, r_2 \in [0, \infty). \quad (4.1)$$

Let f be a real-valued function defined on a subset $E \subseteq S$ and satisfy

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2) + \Delta), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in E, \quad (4.2)$$

where Δ is a positive constant independent of f . Then there exists a function g defined on S such that

$$0 \leq f(\mathbf{x}) - g(\mathbf{x}) \leq \omega(\Delta), \quad \text{for any } \mathbf{x} \in E$$

and

$$|g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in S.$$

In Lemma 4.1, g is an approximate extension of f defined on E to a new domain S with an approximation error $\omega(\Delta)$. In a special case when $\Delta = 0$ and $\omega(0) = 0$, g is an exact extension of f .

Proof of Lemma 4.1. Define

$$g(\mathbf{x}) := \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}) + \Delta)).$$

824 By Equation (4.2), we have $f(\mathbf{x}_1) - \omega(d_S(\mathbf{x}_1, \mathbf{x}_2) + \Delta) \leq f(\mathbf{x}_2)$ for any $\mathbf{x}_1, \mathbf{x}_2 \in E$. It
 825 holds that $g(\mathbf{x}) \leq f(\mathbf{x})$ for any $\mathbf{x} \in E$. Together with

$$826 \quad g(\mathbf{x}) = \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}) + \Delta)) \geq f(\mathbf{x}) - \omega(d_S(\mathbf{x}, \mathbf{x}) + \Delta) = f(\mathbf{x}) - \omega(\Delta),$$

827 for any $\mathbf{x} \in E$, it follows that $0 \leq f(\mathbf{x}) - g(\mathbf{x}) \leq \omega(\Delta)$ for any $\mathbf{x} \in E$. By Equation (4.1)
 828 and the fact

$$829 \quad \sup_{\mathbf{z} \in E} f_1(\mathbf{z}) - \sup_{\mathbf{z} \in E} f_2(\mathbf{z}) \leq \sup_{\mathbf{z} \in E} (f_1(\mathbf{z}) - f_2(\mathbf{z})), \quad \text{for any functions } f_1, f_2,$$

830 we have

$$\begin{aligned} g(\mathbf{x}_1) - g(\mathbf{x}_2) &= \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}_1))) - \sup_{\mathbf{z} \in E} (f(\mathbf{z}) - \omega(d_S(\mathbf{z}, \mathbf{x}_2))) \\ &\leq \sup_{\mathbf{z} \in E} (\omega(d_S(\mathbf{z}, \mathbf{x}_1)) - \omega(d_S(\mathbf{z}, \mathbf{x}_2))) \\ 831 \quad &\leq \sup_{\mathbf{z} \in E} \omega(d_S(\mathbf{z}, \mathbf{x}_1) - d_S(\mathbf{z}, \mathbf{x}_2)) \\ &\leq \sup_{\mathbf{z} \in E} \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)) = \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)), \end{aligned}$$

832 for any $\mathbf{x}_1, \mathbf{x}_2 \in S$. Similarly, we have $g(\mathbf{x}_2) - g(\mathbf{x}_1) \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2))$, which implies

$$833 \quad |g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega(d_S(\mathbf{x}_1, \mathbf{x}_2)).$$

834 So we finish the proof. □

835 Next, we introduce a lemma below for extending continuous functions defined on
 836 $E \subseteq S$ to continuous functions defined on S preserving the modulus of continuity.

837 **Lemma 4.2** (Extension of Continuous Functions). *Suppose f is a uniformly continuous*
 838 *function defined on a subset $E \subseteq S$, where S is a metric space with a metric $d_S(\cdot, \cdot)$, then*
 839 *there exists a uniformly continuous function g on S such that $f(x) = g(x)$ for $x \in E$ and*
 840 *$\omega_f^E(r) = \omega_g^S(r)$ for any $r \geq 0$.*

841 *Proof.* By the application of Lemma 4.1 with $\omega(r) = \omega_f^E(r)$ for $r \geq 0$ and $\Delta = 0$, we know
 842 that there exists $g : S \rightarrow \mathbb{R}$ such that

$$843 \quad 0 \leq f(\mathbf{x}) - g(\mathbf{x}) \leq \omega_f^E(\Delta) = 0, \quad \text{for any } \mathbf{x} \in E,$$

844 and

$$845 \quad |g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega_f^E(d_S(\mathbf{x}_1, \mathbf{x}_2)), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in S.$$

846 The equation above and the uniform continuity of f imply that g is uniformly continuous.
 847 It also follows that

$$848 \quad f(\mathbf{x}) = g(\mathbf{x}), \quad \text{for any } \mathbf{x} \in E, \quad \text{and} \quad \omega_g^S(r) \leq \omega_f^E(r), \quad \text{for any } r \geq 0,$$

849 since $\omega_g^S(\cdot)$ is the optimal modulus of continuity of g . Note that $\omega_f^E(\cdot)$ is the optimal
 850 modulus of continuity of f and

$$851 \quad |f(\mathbf{x}_1) - f(\mathbf{x}_2)| = |g(\mathbf{x}_1) - g(\mathbf{x}_2)| \leq \omega_g^S(d_S(\mathbf{x}_1, \mathbf{x}_2)), \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in E.$$

852 Hence, $\omega_f^E(r) \leq \omega_g^S(r)$ for all $r \geq 0$, which implies $\omega_f^E(r) = \omega_g^S(r)$ since we have proved
 853 that $\omega_g^S(r) \leq \omega_f^E(r)$ for all $r \geq 0$. So we finish the proof. □

Now we are ready to introduce and prove the main theorem of this section, which extends Theorem 1.1 to an irregular domain as follows.

Theorem 4.3. *Let f be a uniformly continuous function defined on $E \subseteq [-R, R]^d$. For arbitrary $L \in \mathbb{N}^+$ and $N \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU FNN with width $3^{d+3} \max\{d \lfloor N^{1/d} \rfloor, N+1\}$ and depth $12L+14+2d$ such that*

$$\|f - \phi\|_{L^\infty(E)} \leq 19\sqrt{d}\omega_f^E(2R\sqrt{d}N^{-2/d}L^{-2/d}).$$

Proof. By Lemma 4.2, f can be extended to \mathbb{R}^d such that

$$\omega_f^{\mathbb{R}^d}(r) = \omega_f^E(r), \quad \text{for any } r \geq 0.$$

Define

$$\tilde{f}(\mathbf{x}) := f(2R\mathbf{x} - R), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d.$$

It follows that

$$\omega_{\tilde{f}}^{\mathbb{R}^d}(r) = \omega_f^{\mathbb{R}^d}(2Rr) = \omega_f^E(2Rr), \quad \text{for any } r \geq 0. \quad (4.3)$$

By Theorem 1.1, there exists a function $\tilde{\phi}$ implemented by a ReLU FNN with width $3^{d+3} \max\{d \lfloor N^{1/d} \rfloor, N+1\}$ and depth $12L+14+2d$ such that

$$\|\tilde{f} - \tilde{\phi}\|_{L^\infty([0,1]^d)} \leq 19\sqrt{d}\omega_{\tilde{f}}^{\mathbb{R}^d}(N^{-2/d}L^{-2/d}).$$

Define

$$\phi(\mathbf{x}) := \tilde{\phi}\left(\frac{1}{2R}\mathbf{x} + \frac{1}{2}\right), \quad \text{for any } \mathbf{x} \in \mathbb{R}^d.$$

Then, by Equation (4.3), for any $\mathbf{x} \in E \subseteq [-R, R]^d$, we have

$$\begin{aligned} |f(\mathbf{x}) - \phi(\mathbf{x})| &= |\tilde{f}\left(\frac{1}{2R}\mathbf{x} + \frac{1}{2}\right) - \tilde{\phi}\left(\frac{1}{2R}\mathbf{x} + \frac{1}{2}\right)| \leq \|\tilde{f} - \tilde{\phi}\|_{L^\infty([0,1]^d)} \\ &\leq 19\sqrt{d}\omega_{\tilde{f}}^{\mathbb{R}^d}(N^{-2/d}L^{-2/d}) = 19\sqrt{d}\omega_f^E(2RN^{-2/d}L^{-2/d}), \end{aligned}$$

which implies

$$\|f - \phi\|_{L^\infty(E)} \leq 19\sqrt{d}\omega_f^E(2RN^{-2/d}L^{-2/d}).$$

So we finish the proof. \square

4.2 Approximation in a neighborhood of a low-dimensional manifold

In this section, we study neural network approximation of functions defined in a neighborhood of a low-dimensional manifold and prove Theorem 1.2 in this setting. Let us first introduce Theorem 4.4 which is required to prove Theorem 1.2.

Theorem 4.4 (Theorem 3.1 of [3]). *Let \mathcal{M} be a compact $d_{\mathcal{M}}$ -dimensional Riemannian submanifold of \mathbb{R}^d having condition number $1/\tau$, volume V , and geodesic covering regularity \mathcal{R} . Fix $\delta \in (0, 1)$ and $\gamma \in (0, 1)$. Let $\mathbf{A} = \sqrt{\frac{d}{d_\delta}}\Phi$, where $\Phi \in \mathbb{R}^{d_\delta \times d}$ is a random orthoprojector with*

$$d_\delta = \mathcal{O}\left(\frac{d_{\mathcal{M}} \ln(dV\mathcal{R}\tau^{-1}\delta^{-1}) \ln(1/\gamma)}{\delta^2}\right).$$

If $d_\delta \leq d$, then with probability at least $1 - \gamma$, the following statement holds: For every $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M}$,

$$(1 - \delta)|\mathbf{x}_1 - \mathbf{x}_2| \leq |\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| \leq (1 + \delta)|\mathbf{x}_1 - \mathbf{x}_2|.$$

Theorem 4.4 shows the existence of a linear projector $\mathbf{A} \in \mathbb{R}^{d_\delta \times d}$ that maps a low-dimensional manifold in a high-dimensional space to a low-dimensional space nearly preserving distance. With this projection \mathbf{A} available, we can prove Theorem 1.2 via constructing a ReLU FNN defined in the low-dimensional space using Theorem 4.3 and hence the curse of dimensionality is lessened. The ideas of the proof are summarized in the following Table 1.

In Table 1 and the detailed proof later, we introduce a new notation $\mathcal{SL}(E)$ for any compact set $E \subseteq \mathbb{R}^d$ as the “smallest” element of E . Specifically, $\mathcal{SL}(E)$ is defined as the unique point in $\cap_{k=1}^d E_k$, where

$$E_k := \{\mathbf{x} \in E_{k-1} : x_k = s_k\}, \quad s_k := \inf\{x_k : (x_1, x_2, \dots, x_d) \in E_{k-1}\}, \quad \text{for } k = 1, 2, \dots, d,$$

and $E_0 = E$. The compactness of E ensures that $\cap_{k=1}^d E_k$ is in fact one point belonging to E . The introduction of $\mathcal{SL}(\cdot)$ uniquely formulates a low-dimensional function \tilde{f} representing a high-dimensional function f defined on \mathcal{M}_ε by

$$\tilde{f}(\mathbf{y}) := f(\mathbf{x}_\mathbf{y}), \quad \text{where } \mathbf{x}_\mathbf{y} = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}), \quad \text{for any } \mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}.$$

As we shall see later, such a definition of \tilde{f} is reasonable because $\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}$ is contained in a small ball of radius $\mathcal{O}(\varepsilon)$ for any $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon)$. There are many other alternative ways to define $\mathcal{SL}(\cdot)$ as long as the definition ensures that $\mathcal{SL}(E)$ contains only one element. For example, $\mathcal{SL}(E)$ can be defined as any arbitrary point in E . For another example, $\mathbf{y} \in \mathbf{A}(\mathcal{M})$ cannot guarantee $\mathbf{x}_\mathbf{y} = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}) \in \mathcal{M}$ in the current definition, but in practice we can choose $\mathcal{SL}(\{\mathbf{x} \in \mathcal{M} : \mathbf{A}\mathbf{x} = \mathbf{y}\})$ as $\mathbf{x}_\mathbf{y}$ to ensure that $\mathbf{x}_\mathbf{y} \in \mathcal{M}$, which might be beneficial for potential applications.

Table 1: Main steps of the proof of Theorem 1.2. Step 1: dimension reduction via the nearly isometric projection operator \mathbf{A} provided by Theorem 4.4 to obtain an “equivalent” function \tilde{f} of f in a low-dimensional domain using $\mathbf{x}_\mathbf{y} = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\})$. Step 2: construct a ReLU FNN to implement $\tilde{\phi} \approx \tilde{f}$ by Theorem 4.3. Step 3: define a ReLU FNN to implement ϕ in the original high-dimensional domain via the projection \mathbf{A} . Step 4: verify that the approximation error of $\phi \approx f$ satisfies our requirement.

$f(\mathbf{x})$ for $\mathbf{x} \in \mathcal{M}_\varepsilon \subseteq [0, 1]^d$	Step 4 \approx	$\phi(\mathbf{x}) := \tilde{\phi}(\mathbf{A}\mathbf{x})$ for $\mathbf{x} \in \mathcal{M}_\varepsilon \subseteq [0, 1]^d$
Step 1 $\Downarrow \mathbf{x}_\mathbf{y} = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\})$		Step 3 $\Downarrow \mathbf{y} = \mathbf{A}\mathbf{x}$
$\tilde{f}(\mathbf{y}) := f(\mathbf{x}_\mathbf{y})$ for $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}$	Step 2 \approx	$\tilde{\phi}(\mathbf{y})$ for $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}$

Now we are ready to prove Theorem 1.2.

Proof of Theorem 1.2. By Theorem 4.4, there exists a matrix $\mathbf{A} \in \mathbb{R}^{d_\delta \times d}$ such that

$$\mathbf{A}\mathbf{A}^T = \frac{d}{d_\delta} \mathbf{I}_{d_\delta}, \quad (4.4)$$

where \mathbf{I}_{d_δ} is an identity matrix of size $d_\delta \times d_\delta$, and

$$(1 - \delta)|\mathbf{x}_1 - \mathbf{x}_2| \leq |\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| \leq (1 + \delta)|\mathbf{x}_1 - \mathbf{x}_2|, \quad \text{for any } \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M}. \quad (4.5)$$

Given any $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon)$, then $\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}$ is a nonzero compact set. Let $\mathbf{x}_y = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\})$, then we define \tilde{f} on $\mathbf{A}(\mathcal{M}_\varepsilon)$ as $\tilde{f}(\mathbf{y}) = f(\mathbf{x}_y)$.

For any $\mathbf{y}_1, \mathbf{y}_2 \in \mathbf{A}(\mathcal{M}_\varepsilon)$, let $\mathbf{x}_i = \mathcal{SL}(\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}_i\})$, then $\mathbf{x}_i \in \mathcal{M}_\varepsilon$ for $i = 1, 2$. By the definition of \mathcal{M}_ε , there exist $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 \in \mathcal{M}$ such that $|\tilde{\mathbf{x}}_i - \mathbf{x}_i| \leq \varepsilon$ for $i = 1, 2$. It follows that

$$|\tilde{f}(\mathbf{y}_1) - \tilde{f}(\mathbf{y}_2)| = |f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq \omega_f(|\mathbf{x}_1 - \mathbf{x}_2|) \leq \omega_f(|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2| + 2\varepsilon) \leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{A}\tilde{\mathbf{x}}_1 - \mathbf{A}\tilde{\mathbf{x}}_2| + 2\varepsilon\right),$$

where the last inequality comes from Equation (4.5). By the triangular inequality, we have

$$\begin{aligned} |\tilde{f}(\mathbf{y}_1) - \tilde{f}(\mathbf{y}_2)| &\leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| + \frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\tilde{\mathbf{x}}_1| + \frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_2 - \mathbf{A}\tilde{\mathbf{x}}_2| + 2\varepsilon\right) \\ &\leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2| + \frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) \\ &\leq \omega_f\left(\frac{1}{1-\delta}|\mathbf{y}_1 - \mathbf{y}_2| + \frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right). \end{aligned}$$

Set $\omega(r) = \omega_f\left(\frac{1}{1-\delta}r\right)$ for any $r \geq 0$ and $\Delta = 2\varepsilon\sqrt{\frac{d}{d_\delta}} + 2\varepsilon(1-\delta)$, then

$$|\tilde{f}(\mathbf{y}_1) - \tilde{f}(\mathbf{y}_2)| \leq \omega(|\mathbf{y}_1 - \mathbf{y}_2| + \Delta), \quad \text{for any } \mathbf{y}_1, \mathbf{y}_2 \in \mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbb{R}^{d_\delta}.$$

By Lemma 4.1, there exists \tilde{g} defined on \mathbb{R}^{d_δ} such that

$$|\tilde{g}(\mathbf{y}) - \tilde{f}(\mathbf{y})| \leq \omega(\Delta) = \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right), \quad \text{for any } \mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon), \quad (4.6)$$

and

$$|\tilde{g}(\mathbf{y}_1) - \tilde{g}(\mathbf{y}_2)| \leq \omega(|\mathbf{y}_1 - \mathbf{y}_2|) = \omega_f\left(\frac{1}{1-\delta}|\mathbf{y}_1 - \mathbf{y}_2|\right), \quad \text{for any } \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^{d_\delta}.$$

It follows that

$$\omega_{\tilde{g}}^{\mathbb{R}^{d_\delta}}(r) \leq \omega_f\left(\frac{r}{1-\delta}\right), \quad \text{for any } r \geq 0. \quad (4.7)$$

By Equation (4.4) and the definition of \mathcal{M}_ε in Equation (1.2), it is easy to check that

$$\mathbf{A}(\mathcal{M}_\varepsilon) \subseteq \mathbf{A}([0, 1]^d) \subseteq [-\sqrt{\frac{d}{d_\delta}}, \sqrt{\frac{d}{d_\delta}}]^{d_\delta}.$$

By the application of Theorem 4.3 with $E = [-\sqrt{\frac{d}{d_\delta}}, \sqrt{\frac{d}{d_\delta}}]^{d_\delta}$, there exists a function $\tilde{\phi}$ implemented by a ReLU FNN with width $3^{d_\delta+3} \max\{d_\delta \lfloor N^{1/d_\delta} \rfloor, N+1\}$ and depth $12L + 14 + 2d_\delta$ such that

$$\|\tilde{g} - \tilde{\phi}\|_{L^\infty(E)} \leq 19\sqrt{d}\omega_g^E\left(2\sqrt{\frac{d}{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right). \quad (4.8)$$

Define $\phi := \tilde{\phi} \circ \mathbf{A}$, i.e., $\phi(\mathbf{x}) := \tilde{\phi}(\mathbf{A}\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^d$. Then ϕ is also a ReLU FNN with width $3^{d_\delta+3} \max\{d_\delta \lfloor N^{1/d_\delta} \rfloor, N+1\}$ and depth $12L + 14 + 2d_\delta$.

For any $\mathbf{x} \in \mathcal{M}_\varepsilon$, set $\mathbf{y} = \mathbf{A}\mathbf{x}$ and $\mathbf{x}_y = \mathcal{SL}(\{\mathbf{z} \in \mathbb{R}^d : \mathbf{A}\mathbf{z} = \mathbf{y}\})$, there exist $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_y \in \mathcal{M}$ such that $|\tilde{\mathbf{x}} - \mathbf{x}| \leq \varepsilon$ and $|\tilde{\mathbf{x}}_y - \mathbf{x}_y| \leq \varepsilon$. It follows from Equation (4.5) that

$$\begin{aligned} |\mathbf{x} - \mathbf{x}_y| &\leq |\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_y| + 2\varepsilon \leq \frac{1}{1-\delta}|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{A}\tilde{\mathbf{x}}_y| + 2\varepsilon \\ &\leq \frac{1}{1-\delta}(|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{A}\mathbf{x}| + |\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}_y| + |\mathbf{A}\mathbf{x}_y - \mathbf{A}\tilde{\mathbf{x}}_y|) + 2\varepsilon \\ &= \frac{1}{1-\delta}(|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{A}\mathbf{x}| + |\mathbf{A}\mathbf{x}_y - \mathbf{A}\tilde{\mathbf{x}}_y|) + 2\varepsilon \leq \frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon. \end{aligned} \quad (4.9)$$

944 In fact, the above equation implies that $\{\mathbf{x} \in \mathcal{M}_\varepsilon : \mathbf{A}\mathbf{x} = \mathbf{y}\}$ is contained in a small ball
 945 of radius $\mathcal{O}(\varepsilon)$ for $\mathbf{y} \in \mathbf{A}(\mathcal{M}_\varepsilon)$ as we mentioned previously.

946 Together with Equation (4.6), (4.7), (4.8), and (4.9), we have, for any $\mathbf{x} \in \mathcal{M}_\varepsilon$,

$$\begin{aligned}
 |f(\mathbf{x}) - \phi(\mathbf{x})| &\leq |f(\mathbf{x}) - f(\mathbf{x}_\mathbf{y})| + |f(\mathbf{x}_\mathbf{y}) - \phi(\mathbf{x})| \\
 &\leq \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + |\tilde{f}(\mathbf{y}) - \tilde{\phi}(\mathbf{y})| \\
 947 &\leq \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + |\tilde{f}(\mathbf{y}) - \tilde{g}(\mathbf{y})| + |\tilde{g}(\mathbf{y}) - \tilde{\phi}(\mathbf{y})| \\
 &\leq \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + \omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_{\tilde{g}}^E(2\sqrt{\frac{d}{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}) \\
 &\leq 2\omega_f\left(\frac{2\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}} + 2\varepsilon\right) + 19\sqrt{d}\omega_f\left(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta}\right).
 \end{aligned}$$

948 Hence, we have finished the proof of this theorem. \square

949 It is worth emphasizing that the approximation error

$$950 \quad \mathcal{O}\left(\omega_f(\mathcal{O}(\varepsilon)) + \omega_f(\mathcal{O}(N^{-2/d_\delta}L^{-2/d_\delta}))\right)$$

951 in Theorem 1.2 is equal to $\mathcal{O}\left(\omega_f(\mathcal{O}(N^{-2/d_\delta}L^{-2/d_\delta}))\right)$ when $\varepsilon = \mathcal{O}(N^{-2/d_\delta}L^{-2/d_\delta})$.

952 The application of Theorem 4.4 and the proof of Theorem 1.2 in fact inspire an
 953 efficient two-step algorithm for high-dimensional learning problems: in the first step,
 954 high-dimensional data are projected to a low-dimensional space via a random projection;
 955 in the second step, a deep learning algorithm is applied to learn from the low-dimensional
 956 data. By Theorem 4.4 and 1.2, the deep learning algorithm in the low-dimensional space
 957 can still provide good results with a high probability.

958 4.3 Optimal ReLU FNN structure in parallel computing

959 In this section, we show how to select the best ReLU FNN to approximate functions
 960 in $\text{Höld}([0, 1]^d, \alpha, \lambda)$ on a d -dimensional cube, if the approximation error ε and the
 961 number of parallel computing cores (processors) p are given. We choose the best ReLU
 962 FNN by minimizing the time complexity in each training iteration. The analysis in this
 963 section is valid up to a constant prefactor.

964 Assume $\phi_\theta \in \mathcal{NN}(\#\text{input} = d; \text{widthvec} = [N]^L)$, $N, L \in \mathbb{N}^+$, where θ is the vector
 965 including all parameters of ϕ_θ . By the basic knowledge of parallel computing (see [36]
 966 for more details), we have the following Table 2.

Table 2: Time complexity of one training iteration for an FNN of width N and depth L .

Number of cores p	Time Complexity	
	Evaluating $\phi_\theta(\mathbf{x})$	Evaluating $\frac{\partial \phi_\theta(\mathbf{x})}{\partial \theta}$
$p \in [1, N]$	$\mathcal{O}(N^2 L/p)$	$\mathcal{O}(N^2 L/p)$
$p \in (N, N^2]$	$\mathcal{O}(L(N^2/p + \ln \frac{p}{N}))$	$\mathcal{O}(L(N^2/p + \ln \frac{p}{N}))$
$p \in (N^2, \infty)$	$\mathcal{O}(L \ln N)$	$\mathcal{O}(L \ln N)$

For the sake of simplicity, we assume that the training batch size is $\mathcal{O}(1)$. Denote the time complexity of each training iteration as $T(n, L)$, then

$$T(N, L) = \begin{cases} \mathcal{O}(N^2 L/p), & p \in [1, N], \\ \mathcal{O}(L(N^2/p + \ln \frac{p}{N})), & p \in (N, N^2], \\ \mathcal{O}(L \ln N), & p \in (N^2, \infty). \end{cases}$$

Theorem 1.1 and 2.3 imply that the approximation error ε is essentially $\mathcal{O}((NL)^{-2\alpha/d})$. Hence, we can get the optimal size of ReLU FNNs via the optimization problem below:

$$\begin{aligned} (N_{\text{opt}}, L_{\text{opt}}) &= \arg \min_{N, L} T(N, L) \\ \text{subject to } &\begin{cases} \varepsilon = \mathcal{O}((NL)^{-2\alpha/d}), \\ N, L, p \in N^+. \end{cases} \end{aligned} \quad (4.10)$$

To simplify the discussion, we have the following assumptions:

- Dropping the notation $\mathcal{O}(\cdot)$ sometimes while assuming asymptotic analysis with the abuse of notations.
- N , L , and p are allowed to be real numbers.
- We denote $\varepsilon = (NL)^{-2\alpha/d}$ since the approximation rate $\mathcal{O}((NL)^{-2\alpha/d})$ is both attainable and nearly optimal.

With $\varepsilon = (NL)^{-2\alpha/d}$, we have

$$\begin{aligned} \bar{T}(N, L) &:= \begin{cases} N^2 L/p, & p \in [1, N], \\ L(N^2/p + \ln \frac{p}{N}), & p \in (N, N^2], \\ L(1 + \ln N), & p \in [N^2, \infty), \end{cases} \\ &= \begin{cases} N\varepsilon^{-d/(2\alpha)}/p, & N \in [p, \infty), \\ N\varepsilon^{-d/(2\alpha)}/p + \frac{1}{N}\varepsilon^{-d/(2\alpha)} \ln \frac{p}{N}, & N \in [\sqrt{p}, p], \\ \frac{1+\ln N}{N}\varepsilon^{-d/(2\alpha)}, & N \in [1, \sqrt{p}). \end{cases} \end{aligned} \quad (4.11)$$

Then we get $T(N, L) = \mathcal{O}(\bar{T}(N, L))$. Therefore, the optimization problem in Equation (4.10) can be simplified to

$$\begin{aligned} (N_{\text{opt}}, L_{\text{opt}}) &= \arg \min_{N, L} \bar{T}(N, L) \\ \text{subject to } &\begin{cases} \varepsilon = (NL)^{-2\alpha/d}, \\ N, L, p \in [1, \infty). \end{cases} \end{aligned} \quad (4.12)$$

By Equation (4.11), $\bar{T}(N, L)$ is independent of L on the condition that $\varepsilon = (NL)^{-2\alpha/d}$. Therefore, we may denote $\bar{T}(N, L)$ by $\bar{T}(N)$. Now we consider two cases: the case $p = \mathcal{O}(1)$ and the case $p \gg \mathcal{O}(1)$.

Case 1: The case $p = \mathcal{O}(1)$.

It is clear that $\bar{T}(N)$ is increasing in N when $N \in [p, \infty)$ by Equation (4.11). Together with $p = \mathcal{O}(1)$, then $\mathcal{O}(\sqrt{p}) = \mathcal{O}(p) = \mathcal{O}(1)$. Therefore, $N_{\text{opt}} = \mathcal{O}(1)$ and

$L_{\text{opt}} = \mathcal{O}(\varepsilon^{-d/(2\alpha)})$. Note that we regard d as a constant ($\mathcal{O}(1)$) in above analysis, N_{opt} should be $\mathcal{O}(d)$ in fact.

Case 2: The case $p \gg \mathcal{O}(1)$.

Since $\varepsilon = (NL)^{-2\alpha/d}$, we have $N \leq \varepsilon^{-d/(2\alpha)}$. We only need to consider the monotonicity of $\bar{T}(N)$ on $[1, \varepsilon^{-d/(2\alpha)}]$. Together with Equation (4.11), this case can be divided into two sub-cases: the sub-case $\sqrt{p} \leq \varepsilon^{-d/(2\alpha)}$ and the sub-case $\sqrt{p} > \varepsilon^{-d/(2\alpha)}$.

Case 2.1: The sub-case $\sqrt{p} > \varepsilon^{-d/(2\alpha)}$.

$\sqrt{p} > \varepsilon^{-d/(2\alpha)}$ implies $[1, \varepsilon^{-d/(2\alpha)}] \subseteq [1, \sqrt{p}]$. Hence, $\bar{T}(N)$ is decreasing in N on $[1, \varepsilon^{-d/(2\alpha)}]$. It follows that $N_{\text{opt}} = \mathcal{O}(\varepsilon^{-d/(2\alpha)})$ and that $L_{\text{opt}} = \mathcal{O}(1)$.

Case 2.2: The sub-case $\sqrt{p} \leq \varepsilon^{-d/(2\alpha)}$.

For this sub-case, N_{opt} and N_{opt} are hard to estimate. However, we can give a rough range of N_{opt} . Since $\bar{T}(N)$ is decreasing in N on $[1, \sqrt{p}]$ and increasing in N on $[p, \infty)$, the minimum of $\bar{T}(N)$ is achieved on $[\sqrt{p}, p]$. Hence, $N_{\text{opt}} \in [\mathcal{O}(\sqrt{p}), \mathcal{O}(p)] \cap [\mathcal{O}(\sqrt{p}), \mathcal{O}(\varepsilon^{-d/(2\alpha)})]$ and $L_{\text{opt}} = \mathcal{O}(\varepsilon^{-d/(2\alpha)}/N_{\text{opt}})$.

5 Conclusion and future work

This paper aims at a quantitative and optimal approximation rate of ReLU FNNs in terms of both width and depth simultaneously to approximate continuous functions. It was shown that ReLU FNNs with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ can approximate an arbitrary continuous function on a d -dimensional cube with an approximation rate $19\sqrt{d}\omega_f(N^{-2/d}L^{-2/d})$. In particular, when f is a Hölder continuous function of order α with a Hölder constant λ , the approximation rate is $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$ and it is nearly asymptotically tight. We also extended our analysis to the case when the domain of f is irregular and showed the same approximation rate. In practical applications, it is usually believed that real data are sampled from an ε -neighborhood of a $d_{\mathcal{M}}$ -dimensional smooth manifold $\mathcal{M} \subseteq [0, 1]^d$ with $d_{\mathcal{M}} \ll d$. In the case of an essentially low-dimensional domain, we show an approximation rate $2\omega_f(\frac{4\varepsilon}{1-\delta}\sqrt{\frac{d}{d_\delta}}) + 19\sqrt{d}\omega_f(\frac{2\sqrt{d}}{(1-\delta)\sqrt{d_\delta}}N^{-2/d_\delta}L^{-2/d_\delta})$ for ReLU FNNs to approximate f in the ε -neighborhood, $d_\delta = \mathcal{O}(d_{\mathcal{M}}\frac{\ln(d/\delta)}{\delta^2})$ for any given $\delta \in (0, 1)$.

Besides, we studied how to select the best ReLU FNN to approximate continuous function in parallel computing. In particular, ReLU FNNs with depth $\mathcal{O}(1)$ are the best choices if the number of parallel computing cores p is sufficiently large. ReLU FNNs with width $\mathcal{O}(d)$ are best choices if $p = \mathcal{O}(1)$. The width of best ReLU FNNs is between $\mathcal{O}(\sqrt{p})$ and $\mathcal{O}(p)$ if p is moderate.

We would like to remark that our analysis was based on the fully connected feed-forward neural networks and the ReLU activation function. It would be very interesting to generalize our conclusions to neural networks with other types of architectures (e.g., convolutional neural networks) and activation functions (e.g., tanh and sigmoid functions). Besides, if identity maps are allowed in the construction of neural networks as in the residual networks [28], the size of FNNs in our construction can be further optimized. Finally, the proposed analysis could be generalized to other function spaces with explicit

formulas to characterize the approximation error. These will be left as future work.

Acknowledgments

H. Yang was partially supported by National Science Foundation under the grant award 1945029.

References

- [1] O. ABDEL-HAMID, A. MOHAMED, H. JIANG, L. DENG, G. PENN, AND D. YU, *Convolutional neural networks for speech recognition*, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22 (2014), pp. 1533–1545.
- [2] M. ANTHONY AND P. L. BARTLETT, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, New York, NY, USA, 1st ed., 2009.
- [3] R. G. BARANIUK AND M. B. WAKIN, *Random projections of smooth manifolds*, Foundations of Computational Mathematics, 9 (2009), pp. 51–77.
- [4] A. R. BARRON, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Transactions on Information Theory, 39 (1993), pp. 930–945.
- [5] P. BARTLETT, V. MAIOROV, AND R. MEIR, *Almost linear VC dimension bounds for piecewise polynomial networks*, Neural Computation, 10 (1998), pp. 217–3.
- [6] M. BIANCHINI AND F. SCARSELLI, *On the complexity of neural network classifiers: A comparison between shallow and deep architectures*, IEEE Transactions on Neural Networks and Learning Systems, 25 (2014), pp. 1553–1565.
- [7] E. K. BLUM AND L. K. LI, *Approximation theory and feedforward networks*, Neural Networks, 4 (1991), pp. 511 – 515.
- [8] D. S. BROOMHEAD AND D. LOWE, *Multivariable Functional Interpolation and Adaptive Networks*, Complex Systems 2, (1988), pp. 321–355.
- [9] J. CAI, D. LI, J. SUN, AND K. WANG, *Enhanced expressive power and fast training of neural networks by random projections*, CoRR, abs/1811.09054 (2018).
- [10] S. CHEN AND D. DONOHO, *Basis pursuit*, in Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers, vol. 1, Oct 1994, pp. 41–44 vol.1.
- [11] C. K. CHUI, S.-B. LIN, AND D.-X. ZHOU, *Construction of neural networks for realization of localized deep learning*, Frontiers in Applied Mathematics and Statistics, 4 (2018), p. 14.
- [12] D. C. CIREŞAN, U. MEIER, J. MASCI, L. M. GAMBARDELLA, AND J. SCHMID-HUBER, *Flexible, high performance convolutional neural networks for image classification*, in Proceedings of the Twenty-Second International Joint Conference on

- Artificial Intelligence - Volume Volume Two, IJCAI'11, AAAI Press, 2011, pp. 1237–1242.
- [13] D. COSTARELLI AND A. R. SAMBUCINI, *Saturation classes for max-product neural network operators activated by sigmoidal functions*, Results in Mathematics, 72 (2017), pp. 1555 – 1569.
- [14] D. COSTARELLI AND G. VINTI, *Convergence for a family of neural network operators in orlicz spaces*, Mathematische Nachrichten, 290 (2017), pp. 226–235.
- [15] —, *Approximation results in orlicz spaces for sequences of kantorovich max-product neural network operators*, Results in Mathematics, 73 (2018), pp. 1 – 15.
- [16] G. CYBENKO, *Approximation by superpositions of a sigmoidal function*, MCSS, 2 (1989), pp. 303–314.
- [17] I. DAUBECHIES, R. DEVORE, S. FOUCART, B. HANIN, AND G. PETROVA, *Non-linear approximation and (deep) relu networks*, vol. abs/1905.02199, 2019.
- [18] R. DEVORE AND A. RON, *Approximation using scattered shifts of a multivariate function*, Transactions of the American Mathematical Society, 362 (2010), pp. 6205–6229.
- [19] R. A. DEVORE, *Nonlinear approximation*, Acta Numerica, 7 (1998), p. 51150.
- [20] W. E, J. HAN, AND A. JENTZEN, *Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations*, Communications in Mathematics and Statistics, 5 (2017), pp. 349–380.
- [21] W. E, C. MA, AND Q. WANG, *A priori estimates of the population risk for residual networks*, ArXiv, abs/1903.02154 (2019).
- [22] W. E, C. MA, AND L. WU, *A priori estimates of the population risk for two-layer neural networks*, Communications in Mathematical Sciences, 17 (2019), pp. 1407 – 1425.
- [23] W. E AND Q. WANG, *Exponential convergence of the deep neural network approximation for analytic functions*, CoRR, abs/1807.00297 (2018).
- [24] J. HAN, A. JENTZEN, AND W. E, *Solving high-dimensional partial differential equations using deep learning*, Proceedings of the National Academy of Sciences, 115 (2018), pp. 8505–8510.
- [25] T. HANGELBROEK AND A. RON, *Nonlinear approximation using gaussian kernels*, Journal of Functional Analysis, 259 (2010), pp. 203 – 219.
- [26] B. HANIN AND M. SELLKE, *Approximating continuous functions by ReLU nets of minimal width*, (2017).

- [27] N. HARVEY, C. LIAW, AND A. MEHRABIAN, *Nearly-tight VC-dimension bounds for piecewise linear neural networks*, in Proceedings of the 2017 Conference on Learning Theory, S. Kale and O. Shamir, eds., vol. 65 of Proceedings of Machine Learning Research, Amsterdam, Netherlands, 07–10 Jul 2017, PMLR, pp. 1064–1068.
- [28] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 770–778.
- [29] K. HORNIK, *Approximation capabilities of multilayer feedforward networks*, Neural Networks, 4 (1991), pp. 251 – 257.
- [30] K. HORNIK, M. STINCHCOMBE, AND H. WHITE, *Multilayer feedforward networks are universal approximators*, Neural Networks, 2 (1989), pp. 359 – 366.
- [31] M. HUTZENTHALER, A. JENTZEN, T. KRUSE, AND T. A. NGUYEN, *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, (2020).
- [32] K. KAWAGUCHI, *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.
- [33] K. KAWAGUCHI AND Y. BENGIO, *Depth with nonlinearity creates no bad local minima in resnets*, (2018).
- [34] M. J. KEARNS AND R. E. SCHAPIRE, *Efficient distribution-free learning of probabilistic concepts*, J. Comput. Syst. Sci., 48 (1994), pp. 464–497.
- [35] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *Imagenet classification with deep convolutional neural networks*, in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds., Curran Associates, Inc., 2012, pp. 1097–1105.
- [36] V. KUMAR, *Introduction to Parallel Computing*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd ed., 2002.
- [37] V. KŮRKOVÁ, *Kolmogorov’s theorem and multilayer neural networks*, Neural Networks, 5 (1992), pp. 501 – 506.
- [38] G. LEWICKI AND G. MARINO, *Approximation of functions of finite variation by superpositions of a sigmoidal function*, Applied Mathematics Letters, 17 (2004), pp. 1147 – 1152.
- [39] S. LIANG AND R. SRIKANT, *Why deep neural networks?*, CoRR, abs/1610.04161 (2016).
- [40] S. LIN, X. LIU, Y. RONG, AND Z. XU, *Almost optimal estimates for approximation and learning by radial basis function networks*, Machine Learning, 95 (2014), pp. 147–164.

- 1136 [41] B. LLANAS AND F. SAINZ, *Constructive approximate interpolation by neural net-*
1137 *works*, Journal of Computational and Applied Mathematics, 188 (2006), pp. 283 –
1138 308.
- 1139 [42] J. LU, Z. SHEN, H. YANG, AND S. ZHANG, *Deep Network Approximation for*
1140 *Smooth Functions*, arXiv e-prints, (2020), p. arXiv:2001.03040.
- 1141 [43] Z. LU, H. PU, F. WANG, Z. HU, AND L. WANG, *The expressive power of neural*
1142 *networks: A view from the width*, in Advances in Neural Information Processing
1143 Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-
1144 wanathan, and R. Garnett, eds., Curran Associates, Inc., 2017, pp. 6231–6239.
- 1145 [44] Z. LU, H. PU, F. WANG, Z. HU, AND L. WANG, *The expressive power of neural*
1146 *networks: A view from the width*, CoRR, abs/1709.02540 (2017).
- 1147 [45] V. MAIOROV AND A. PINKUS, *Lower bounds for approximation by mlp neural*
1148 *networks*, Neurocomputing, 25 (1999), pp. 81 – 91.
- 1149 [46] S. G. MALLAT AND Z. ZHANG, *Matching pursuits with time-frequency dictionaries*,
1150 IEEE Transactions on Signal Processing, 41 (1993), pp. 3397–3415.
- 1151 [47] E. J. MCSHANE, *Extension of range of functions*, Bull. Amer. Math. Soc., 40
1152 (1934), pp. 837–842.
- 1153 [48] H. MONTANELLI AND Q. DU, *New error bounds for deep relu networks using sparse*
1154 *grids*, SIAM Journal on Mathematics of Data Science, 1 (2019), pp. 78–92.
- 1155 [49] H. MONTANELLI AND H. YANG, *Error bounds for deep relu networks using the*
1156 *kolmogorovarnold superposition theorem*, Neural Networks, 129 (2020), pp. 1 – 6.
- 1157 [50] H. MONTANELLI, H. YANG, AND Q. DU, *Deep relu networks overcome the curse of*
1158 *dimensionality for bandlimited functions*, Journal of Computational Mathematics,
1159 (to appear).
- 1160 [51] G. F. MONTUFAR, R. PASCANU, K. CHO, AND Y. BENGIO, *On the number of*
1161 *linear regions of deep neural networks*, in Advances in Neural Information Processing
1162 Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q.
1163 Weinberger, eds., Curran Associates, Inc., 2014, pp. 2924–2932.
- 1164 [52] Q. N. NGUYEN AND M. HEIN, *The loss surface of deep and wide neural networks*,
1165 CoRR, abs/1704.08045 (2017).
- 1166 [53] J. PARK AND I. W. SANDBERG, *Universal approximation using radial-basis-*
1167 *function networks*, Neural Computation, 3 (1991), pp. 246–257.
- 1168 [54] P. PETERSEN AND F. VOIGTLAENDER, *Optimal approximation of piecewise smooth*
1169 *functions using deep ReLU neural networks*, Neural Networks, 108 (2018), pp. 296
1170 – 330.
- 1171 [55] P. PETRUSHEV, *Multivariate n -term rational and piecewise polynomial approxima-*
1172 *tion*, Journal of Approximation Theory, 121 (2003), pp. 158 – 197.

- 1173 [56] D. ROLNICK AND M. TEGMARK, *The power of deeper networks for expressing*
1174 *natural functions*, CoRR, abs/1705.05502 (2017).
- 1175 [57] I. SAFRAN AND O. SHAMIR, *Depth-width tradeoffs in approximating natural func-*
1176 *tions with neural networks*, in Proceedings of the 34th International Conference on
1177 Machine Learning, D. Precup and Y. W. Teh, eds., vol. 70 of Proceedings of Machine
1178 Learning Research, International Convention Centre, Sydney, Australia, 06–11 Aug
1179 2017, PMLR, pp. 2979–2987.
- 1180 [58] A. SAKURAI, *Tight bounds for the VC-dimension of piecewise polynomial networks*,
1181 in Advances in Neural Information Processing Systems, Neural information process-
1182 ing systems foundation, 1999, pp. 323–329.
- 1183 [59] D. SCHERER, A. MÜLLER, AND S. BEHNKE, *Evaluation of pooling operations in*
1184 *convolutional architectures for object recognition*, in Artificial Neural Networks –
1185 ICANN 2010, K. Diamantaras, W. Duch, and L. S. Iliadis, eds., Berlin, Heidelberg,
1186 2010, Springer Berlin Heidelberg, pp. 92–101.
- 1187 [60] J. SCHMIDT-HIEBER, *Nonparametric regression using deep neural networks with*
1188 *ReLU activation function*, (2017).
- 1189 [61] U. SHAHAM, A. CLONINGER, AND R. R. COIFMAN, *Provable approximation prop-*
1190 *erties for deep neural networks*, Applied and Computational Harmonic Analysis, 44
1191 (2018), pp. 537 – 557.
- 1192 [62] Z. SHEN, H. YANG, AND S. ZHANG, *Nonlinear approximation via compositions*,
1193 Neural Networks, 119 (2019), pp. 74 – 84.
- 1194 [63] T. SUZUKI, *Adaptivity of deep reLU network for learning in besov and mixed smooth*
1195 *besov spaces: optimal rate and curse of dimensionality*, in International Conference
1196 on Learning Representations, 2019.
- 1197 [64] H. WHITNEY, *Analytic extensions of differentiable functions defined in closed sets*,
1198 Transactions of the American Mathematical Society, 36 (1934), pp. 63–89.
- 1199 [65] T. F. XIE AND F. L. CAO, *The rate of approximation of gaussian radial basis*
1200 *neural networks in continuous function space*, Acta Mathematica Sinica, English
1201 Series, 29 (2013), pp. 295–302.
- 1202 [66] D. YAROTSKY, *Error bounds for approximations with deep ReLU networks*, Neural
1203 Networks, 94 (2017), pp. 103 – 114.
- 1204 [67] D. YAROTSKY, *Optimal approximation of continuous functions by very deep ReLU*
1205 *networks*, in Proceedings of the 31st Conference On Learning Theory, S. Bubeck,
1206 V. Perchet, and P. Rigollet, eds., vol. 75 of Proceedings of Machine Learning Re-
1207 search, PMLR, 06–09 Jul 2018, pp. 639–649.