Optimal Approximation Rate of ReLU Networks in terms of Width and Depth[*]

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

Abstract

This paper concentrates on the approximation power of deep feed-forward neural networks in terms of width and depth. It is proved by construction that ReLU networks with width $\mathcal{O}\big(\max\{d\lfloor N^{1/d}\rfloor, N+2\}\big)$ and depth $\mathcal{O}(L)$ can approximate a Hölder continuous function on $[0,1]^d$ with an approximation rate $\mathcal{O}\big(\lambda\sqrt{d}(N^2L^2\ln N)^{-\alpha/d}\big)$, where $\alpha\in(0,1]$ and $\lambda>0$ are Hölder order and constant, respectively. Such a rate is optimal up to a constant in terms of width and depth separately, while existing results are only nearly optimal without the logarithmic factor in the approximation rate. More generally, for an arbitrary continuous function f on $[0,1]^d$, the approximation rate becomes $\mathcal{O}\big(\sqrt{d}\,\omega_f\big((N^2L^2\ln N)^{-1/d}\big)\big)$, where $\omega_f(\cdot)$ is the modulus of continuity. We also extend our analysis to any continuous function f on a bounded set. Particularly, if ReLU networks with depth 31 and width $\mathcal{O}(N)$ are used to approximate one-dimensional Lipschitz continuous functions on $[0,1]$ with a Lipschitz constant $\lambda>0$, the approximation rate in terms of the total number of parameters, $W=\mathcal{O}(N^2)$, becomes $\mathcal{O}(\frac{\lambda}{W\ln W})$, which has not been discovered in the literature for fixed-depth ReLU networks.

Key words. Deep ReLU Networks; Optimal Approximation; VC-dimension; Bit Extraction; Hölder Continuity.

1 Introduction

Over the past few decades, the expressiveness of neural networks has been widely studied from many points of view, e.g., in terms of combinatorics [27], topology [4], Vapnik-Chervonenkis (VC) dimension [3,13,31], fat-shattering dimension [1,19], information theory [30], classical approximation theory [2,6,11,16,20,24,32,32,33,34,35,36,41,43], optimization [14, 17, 18, 21, 29]. The error analysis of neural networks consists of three parts: the approximation error, the optimization error, and the generalization error. This paper focuses on the approximation error for ReLU networks.

[*]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).

The approximation errors of feed-forward neural networks with various activation functions have been studied for different types of functions, e.g., smooth functions [9, 22, 24, 25, 40], piecewise smooth functions [30], band-limited functions [26], continuous functions [33, 34, 35, 41]. In the early works of approximation theory for neural networks, the universal approximation theorem [6, 15, 16] without approximation rates showed that there exists a sufficiently large neural network approximating a target function in a certain function space within any given error $\varepsilon > 0$. In particular, it is shown in [23] that the ReLU-activated residual neural network with one-neuron hidden layers is a universal approximator. The universal approximation property for general residual neural networks was proved in [20] via a dynamical system approach.

An asymptotic analysis of the approximation rate in terms of depth is provided in [41, 42] for ReLU networks. To be exact, the nearly optimal approximation rates of ReLU networks with width $\mathcal{O}(d)$ and depth $\mathcal{O}(L)$ for functions in $C([0,1]^d)$ and $C^s([0,1]^d)$ are $\mathcal{O}(\omega_f(L^{-2/d}))$ and $\mathcal{O}((L/\ln L)^{-2s/d})$, respectively. These two papers provide the approximation rate in terms of depth asymptotically for fixed-width networks. A different approach is used in [24, 33] to obtain a quantitative characterization of the approximation rate in terms of width, depth, and smoothness order for continuous and smooth functions.

Particularly, it was shown in [33] that a ReLU network with width $C_1(d)\cdot N$ and depth $C_2(d)\cdot L$ can attain an approximation error $C_3(d)\cdot\omega_f(N^{-2/d}L^{-2/d})$ to approximate a continuous function f on $[0,1]^d$, where $C_1(d)$, $C_2(d)$, and $C_3(d)$ are three constants in d with explicit formulas to specify their values, and $\omega_f(\cdot)$ is the modulus of continuity of $f\in C([0,1]^d)$ defined via

$$\omega_f(r) := \sup\left\{|f(\boldsymbol{x})-f(\boldsymbol{y})| : \boldsymbol{x},\boldsymbol{y}\in[0,1]^d,\ \|\boldsymbol{x}-\boldsymbol{y}\|_2\le r\right\},\quad \text{for any } r\ge 0.$$

Such an approximation rate is optimal in terms of N and L up to a logarithmic term and the corresponding optimal approximation theory is still unavailable. To address this problem, we provide a constructive proof in this paper to show that ReLU networks of width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ can approximate an arbitrary continuous function f on $[0,1]^d$ with an optimal approximation error $\mathcal{O}\left(\sqrt{d}\,\omega_f\big((N^2L^2\ln N)^{-\alpha/d}\big)\right)$ in terms of N and L. As shown by our main result, Theorem 1.1 below, the approximation rate obtained here admits explicit formulas to specify its prefactors when $\omega_f(\cdot)$ is known.

Theorem 1.1. *Given a continuous function $f\in C([0,1]^d)$, for any $N\in\mathbb{N}^+$, $L\in\mathbb{N}^+$, and $p\in[1,\infty]$, there exists a function ϕ implemented by a ReLU network with width $C_1\max\left\{d\lfloor N^{1/d}\rfloor,\, N+2\right\}$ and depth $11L+C_2$ such that*

$$\|f-\phi\|_{L^p([0,1]^d)} \le 131\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big),$$

where $C_1=16$ and $C_2=18$ if $p\in[1,\infty)$; $C_1=3^{d+3}$ and $C_2=18+2d$ if $p=\infty$.

Note that $3^{d+3}\max\left\{d\lfloor N^{1/d}\rfloor,\, N+2\right\}\le 3^{d+3}\max\left\{dN,\,3N\right\}\le 3^{d+4}dN$. Given any $\widetilde{N},\widetilde{L}\in\mathbb{N}^+$ with $\widetilde{N}\ge 3^{d+4}d$ and $\widetilde{L}\ge 29+2d$, there exist $N,L\in\mathbb{N}^+$ such that

$$3^{d+4}dN\le\widetilde{N}<3^{d+4}d(N+1)\quad\text{and}\quad 11L+18+2d\le\widetilde{L}<11(L+1)+18+2d.$$

If follows that

$$N \geq \frac{N+1}{3} > \frac{\widetilde{N}}{3^{d+5}d} \quad \text{and} \quad L \geq \frac{L+1}{2} > \frac{1}{2} \cdot \frac{\widetilde{L}-18-2d}{11} = \frac{\widetilde{L}-18-2d}{22}.$$

Then we have an immediate corollary of Theorem 1.1.

Corollary 1.2. *Given a continuous function $f \in C([0,1]^d)$, for any $\widetilde{N}, \widetilde{L} \in \mathbb{N}^+$ with $\widetilde{N} \geq 3^{d+4}d$ and $\widetilde{L} \geq 29 + 2d$, there exists a function ϕ implemented by a ReLU network with width $\widetilde{N}$ and depth $\widetilde{L}$ such that*

$$\|f - \phi\|_{L^\infty([0,1]^d)} \leq 131\sqrt{d}\,\omega_f\bigg(\Big(\big(\tfrac{\widetilde{N}}{3^{d+5}d}\big)^2\big(\tfrac{\widetilde{L}-18-2d}{22}\big)^2\log_3\big(\tfrac{\widetilde{N}}{3^{d+5}d}+2\big)\Big)^{-1/d}\bigg).$$

As a special case of Theorem 1.1 for explicit error characterization, let us take Hölder continuous functions as an example. Let Hölder$([0,1]^d, \alpha, \lambda)$ denote the space of Hölder continuous functions on $[0,1]^d$ of order $\alpha \in (0,1]$ with a Hölder constant $\lambda > 0$. We have an immediate corollary of Theorem 1.1 as follows.

Corollary 1.3. *Given a Hölder continuous function $f \in \text{Hölder}([0,1]^d, \alpha, \lambda)$, for any $N \in \mathbb{N}^+$, $L \in \mathbb{N}^+$, and $p \in [1,\infty]$, there exists a function ϕ implemented by a ReLU network with width $C_1 \max\{d\lfloor N^{1/d}\rfloor, N+2\}$ and depth $11L + C_2$ such that*

$$\|f - \phi\|_{L^p([0,1]^d)} \leq 131\lambda\sqrt{d}\big(N^2L^2\log_3(N+2)\big)^{-\alpha/d},$$

where $C_1 = 16$ and $C_2 = 18$ if $p \in [1,\infty)$; $C_1 = 3^{d+3}$ and $C_2 = 18 + 2d$ if $p = \infty$.

To better illustrate the importance of our theory, we summarize our key contributions as follows.

(1) Upper bound: We provide a quantitative and non-asymptotic approximation rate $131\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big)$ in terms of width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ for any $f \in C([0,1]^d)$ in Theorem 1.1.

  (1.1) This approximation error analysis can be extended to $f \in C(E)$ for any $E \subseteq [-R,R]^d$ with $R > 0$ as we shall see later in Theorem 2.5.

  (1.2) In the case of one-dimensional Lipschitz continuous functions on $[0,1]$ with a constant $\lambda > 0$, the approximation rate in Theorem 1.1 becomes $\mathcal{O}(\frac{\lambda}{W\ln W})$ for ReLU networks with 31 hidden layers and $\mathcal{O}(W)$ parameters via setting $L = 1$ and $W = \mathcal{O}(N^2)$ therein. To the best of our knowledge, the approximation rate $\mathcal{O}(\frac{\lambda}{W\ln W})$ is better than existing known results using fixed-depth ReLU networks to approximate Lipschitz continuous functions on $[0,1]$.

(2) Lower bound: Through the VC-dimension bounds of ReLU networks given in [13], we show, in Section 2.3, that the approximation rate $131\lambda\sqrt{d}\big(N^2L^2\log_3(N+2)\big)^{-\alpha/d}$ in terms of width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ for Hölder$([0,1]^d,\alpha,\lambda)$ is optimal as follows.

(2.1) When the width is fixed, both the approximation upper and lower bounds take the form of $CL^{-2\alpha/d}$ for a positive constant C.

(2.2) When the depth is fixed, both the approximation upper and lower bounds take the form of $C(N^2 \ln N)^{-\alpha/d}$ for a positive constant C.

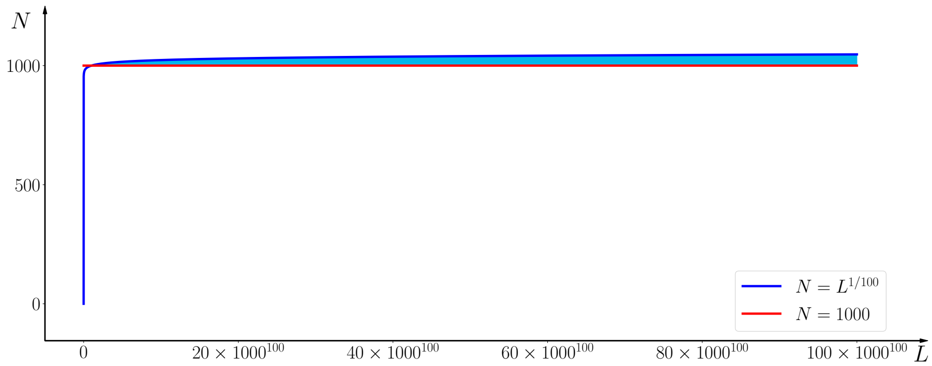


Figure 1: Our rate is optimal in terms of width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ simultaneously except for the region marked in cyan characterized by $\{(N, L) \in \mathbb{N}^2 : C_1 \le N \le L^{C_2}\}$, where $C_i = C_i(\alpha, d)$ for $i = 1, 2$ are two positive constants. This figure is an example for $C_1 = 1000$ and $C_2 = 1/100$.

We would like to point out that if N and L vary simultaneously, the rate is optimal in the N-L plane except for a small region as shown in Figure 1. See Section 2.3 for a detailed discussion. The earlier result in [33] provides a nearly optimal approximation error that has a gap (a logarithmic term) between the lower and upper bounds. It is technically challenging to match the upper bound with the lower bound. Compared to the nearly optimal rate $19\lambda\sqrt{d}N^{-2\alpha/d}L^{-2\alpha/d}$ for Hölder continuous functions in Hölder$([0,1]^d, \alpha, \lambda)$ in [33], this paper achieves the optimal rate $131\lambda\sqrt{d}\big(N^2L^2\log_3(N+2)\big)^{-\alpha/d}$ using more technical and sophisticated construction. For example, a novel bit extraction technique different to that in [3] is proposed, and new ReLU networks are constructed to approximate step functions more efficiently than those in [33]. The optimal result obtained in this paper could also be extended to other functions spaces, leading to better understanding of deep network approximation.

We have obtained the optimal approximation rate for (Hölder) continuous functions approximated by ReLU networks. There are two possible directions to improve the approximation rate or reduce the effect of the curse of dimensionality. The first one is to consider proper target function spaces, e.g., Barron spaces [2, 8, 12, 37], band-limited functions [5, 26], smooth functions [24, 42], and analytic functions [9]. The other direction is to consider neural networks with other activation functions. For example, it is shown in [34] that Floor-ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ admits an approximation rate $\mathcal{O}\big(\omega_f(N^{-\sqrt{L}})\big)$ for continuous function $f \in C([0,1]^d)$ with a prefactor that is only a polynomial of d for Hölder continuous functions. It is shown in [35] that networks with $\mathcal{O}(W)$ parameters using the floor function ($\lfloor x \rfloor$), the exponential function (2^x), and the step function ($\mathbb{1}_{x\ge 0}$) as activation functions can approximate Lipschitz functions with an exponentially small error $\mathcal{O}(2^{-W})$ with a prefactor that is only a polynomial of d. The $(\sin, \mathrm{ReLU})$-activated neural networks are studied in [42] with an approximation error $\mathcal{O}(e^{-c_d\sqrt{W}})$ for Lipschitz functions, where c_d is a constant depending on d.

The error analysis of deep learning is to estimate approximation, generalization, and optimization errors. Here, we give a brief discussion, the interested reader can find more details in [24]. Let $\phi(\boldsymbol{x};\boldsymbol{\theta})$ denote a function computed by a network parameterized with $\boldsymbol{\theta}$. Given a target function f, the final goal is to find the expected risk minimizer

$$\boldsymbol{\theta}_{\mathcal{D}} \coloneqq \arg\min_{\boldsymbol{\theta}} R_{\mathcal{D}}(\boldsymbol{\theta}), \quad \text{where } R_{\mathcal{D}}(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\boldsymbol{x}\sim U(\mathcal{X})}\left[\ell(\phi(\boldsymbol{x};\boldsymbol{\theta}), f(\boldsymbol{x}))\right]$$

with a loss function $\ell(\cdot,\cdot)$ and an unknown data distribution $U(\mathcal{X})$.

In practice, for given samples $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n$, the goal of supervised learning is to identify the empirical risk minimizer

$$\boldsymbol{\theta}_{\mathcal{S}} \coloneqq \arg\min_{\boldsymbol{\theta}} R_{\mathcal{S}}(\boldsymbol{\theta}), \quad \text{where } R_{\mathcal{S}}(\boldsymbol{\theta}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \ell\big(\phi(\boldsymbol{x}_i;\boldsymbol{\theta}), f(\boldsymbol{x}_i)\big).$$

In fact, one could only get a numerical minimizer $\boldsymbol{\theta}_{\mathcal{N}}$ via a numerical optimization method. The discrepancy between the target function f and the learned function $\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}})$ is measured by $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$, which is bounded by

$$R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) \le \underbrace{R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})}_{\text{Approximation error}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}})]}_{\text{Optimization error}} + \underbrace{[R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}})] + [R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}}) - R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\text{Generalization error}}.$$

This paper deals with the approximation error of ReLU networks for continues functions and gives an upper bound of $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$ which is optimal up to a constant. Note that the approximation error analysis given here is independent of data samples and deep learning algorithms. However, the analysis of optimization and generalization errors do depend on data samples, deep learning algorithms, models, etc. For example, refer to [7, 8, 10, 14, 17, 18, 21, 28, 29] for a further understanding of the generalization and optimization errors.

The rest of this paper is organized as follows. In Section 2, we prove Theorem 1.1 by assuming Theorem 2.1 is true, show the optimality of Theorem 1.1, and extend our analysis to continuous functions defined on any bounded set. Next, Theorem 2.1 is proved in Section 3 based on Proposition 3.1 and 3.2, the proofs of which can be found in Section 4. Finally, Section 5 concludes this paper with a short discussion.

2 Theoretical analysis

In this section, we first prove Theorem 1.1 and discuss its optimality. Next, we extend our analysis to general continuous functions defined on any bounded set in $\mathbb{R}^d$. Notations throughout this paper is summarized in Section 2.1.

2.1 Notations

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

- Matrices are denoted by bold uppercase letters. For instance, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a real matrix of size $m \times n$, and $\boldsymbol{A}^T$ denotes the transpose of $\boldsymbol{A}$. Vectors are denoted as bold lowercase letters. For example, $\boldsymbol{v} = [v_1, \cdots, v_d]^T = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} \in \mathbb{R}^d$ is a column vector with $\boldsymbol{v}(i) = v_i$ being the i-th element. Besides, "[" and "]" are used to partition matrices (vectors) into blocks, e.g., $\boldsymbol{A} = \left[\begin{smallmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{smallmatrix}\right]$.

- For any $p \in [1, \infty)$, the p-norm (or ℓ^p-norm) of a vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T \in \mathbb{R}^d$ is defined by
$$\|\boldsymbol{x}\|_p \coloneqq \big(|x_1|^p + |x_2|^p + \cdots + |x_d|^p\big)^{1/p}.$$

- For any $x \in \mathbb{R}$, let $\lfloor x \rfloor \coloneqq \max\{n : n \le x,\ n \in \mathbb{Z}\}$ and $\lceil x \rceil \coloneqq \min\{n : n \ge x,\ n \in \mathbb{Z}\}$.

- Assume $\boldsymbol{n} \in \mathbb{N}^d$, then $f(\boldsymbol{n}) = \mathcal{O}(g(\boldsymbol{n}))$ means that there exists positive C independent of $\boldsymbol{n}$, f, and g such that $f(\boldsymbol{n}) \le Cg(\boldsymbol{n})$ when all entries of $\boldsymbol{n}$ go to $+\infty$.

- For any $\theta \in [0, 1)$, 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., $\text{bin}\,0.\theta_1\theta_2\cdots\theta_L \coloneqq \sum_{\ell=1}^{L} \theta_\ell 2^{-\ell}$.

- Let $\mu(\cdot)$ denote the Lebesgue measure.

- Let $\mathbb{1}_S$ be the characteristic function on a set S, i.e., $\mathbb{1}_S$ is equal to 1 on S and 0 outside S.

- Let $|S|$ denote the size of a set S, i.e., the number of all elements in S.

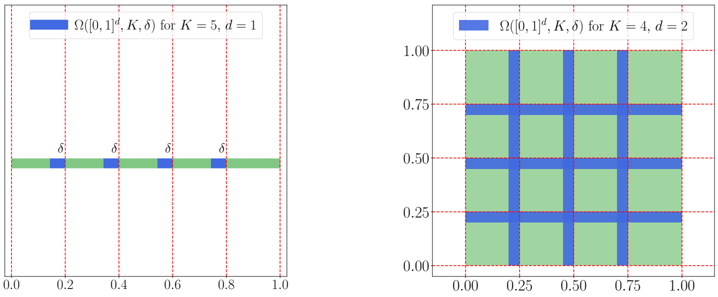
- The set difference of two sets A and B is denoted by $A\backslash B \coloneqq \{x : x \in A,\ x \notin B\}$.

- Given any $K \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{K})$, define a trifling region $\Omega([0,1]^d, K, \delta)$ of $[0,1]^d$ as
$$\Omega([0,1]^d, K, \delta) \coloneqq \bigcup_{j=1}^{d} \Big\{ \boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T \in [0,1]^d : x_j \in \bigcup_{k=1}^{K-1} \big(\tfrac{k}{K} - \delta, \tfrac{k}{K}\big) \Big\}. \tag{2.1}$$
In particular, $\Omega([0,1]^d, K, \delta) = \varnothing$ if $K = 1$. See Figure 2 for two examples of trifling regions.

- Let Hölder$([0,1]^d, \alpha, \lambda)$ denote the space of Hölder continuous functions on $[0,1]^d$ of order $\alpha \in (0, 1]$ with a Hölder constant $\lambda > 0$.

- For a continuous piecewise linear function $f(x)$, the x values where the slope changes are typically called **breakpoints**.

- Let CPwL$(\mathbb{R}, n)$ denote the space that consists of all continuous piecewise linear functions with at most n breakpoints on $\mathbb{R}$.



(a) (b)

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

- Let $\sigma : \mathbb{R} \to \mathbb{R}$ denote the rectified linear unit (ReLU), i.e. $\sigma(x) = \max\{0, x\}$. With a slight abuse of notation, we define $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ as $\sigma(\boldsymbol{x}) = \begin{bmatrix} \max\{0, x_1\} \\ \vdots \\ \max\{0, x_d\} \end{bmatrix}$ for any $\boldsymbol{x} = [x_1, \cdots, x_d]^T \in \mathbb{R}^d$.

- We will use $\mathcal{NN}$ to denote a function implemented by a ReLU network for short and use Python-type notations to specify a class of functions implemented by ReLU networks with several conditions, e.g., $\mathcal{NN}(\mathrm{c}_1;\ \mathrm{c}_2;\ \cdots;\ \mathrm{c}_m)$ is a set of functions implemented by ReLU networks satisfying m conditions given by $\{\mathrm{c}_i\}_{1 \le i \le m}$, each of which may specify the number of inputs (#input), the number of outputs (#output), the total number of neurons in all hidden layers (#neuron), the number of hidden layers (depth), the total number of parameters (#parameter), and the width in each hidden layer (widthvec), the maximum width of all hidden layers (width), etc. For example, if $\phi \in \mathcal{NN}(\#\text{input} = 2;\ \text{widthvec} = [100, 100];\ \#\text{output} = 1)$, then ϕ is a functions satisfies

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

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

$$\boldsymbol{x} = \widetilde{\boldsymbol{h}}_0 \xrightarrow[\mathcal{L}_0]{\boldsymbol{W}_0,\ \boldsymbol{b}_0} \boldsymbol{h}_1 \xrightarrow{\sigma} \widetilde{\boldsymbol{h}}_1 \cdots \xrightarrow[\mathcal{L}_{L-1}]{\boldsymbol{W}_{L-1},\ \boldsymbol{b}_{L-1}} \boldsymbol{h}_L \xrightarrow{\sigma} \widetilde{\boldsymbol{h}}_L \xrightarrow[\mathcal{L}_L]{\boldsymbol{W}_L,\ \boldsymbol{b}_L} \boldsymbol{h}_{L+1} = \phi(\boldsymbol{x}),$$

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

$$\boldsymbol{h}_{i+1} = \boldsymbol{W}_i \cdot \widetilde{\boldsymbol{h}}_i + \boldsymbol{b}_i =: \mathcal{L}_i(\widetilde{\boldsymbol{h}}_i), \quad \text{for } i = 0, 1, \cdots, L,$$

and

$$\widetilde{\boldsymbol{h}}_i = \sigma(\boldsymbol{h}_i), \quad \text{for } i = 1, 2, \cdots, L.$$

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

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

which has been illustrated in Figure 3.

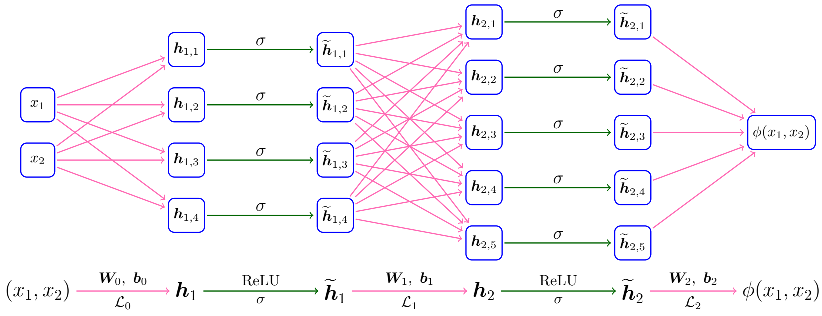


Figure 3: An example of a ReLU network with width 5 and depth 2.

- The expression "a network with width N and depth L" means
  - The maximum width of this network for all **hidden** layers is no more than N.
  - The number of **hidden** layers of this network is no more than L.

2.2 Proof of Theorem 1.1

The key point is to construct piecewise constant functions to approximate continuous functions in the proof. However, it is impossible to construct a piecewise constant function implemented by a ReLU network due to the continuity of ReLU networks. Thus, we introduce the trifling region $\Omega([0,1]^d, K, \delta)$, defined in Equation (2.1), and use ReLU networks to implement piecewise constant functions outside the trifling region. To prove Theorem 1.1, we first introduce a weaker variant of Theorem 1.1, showing how to construct ReLU networks to pointwisely approximate continuous functions except for the trifling region.

Theorem 2.1. *Given a function $f \in C([0,1]^d)$, for any $N \in \mathbb{N}^+$ and $L \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU network with width $\max\{8d\lfloor N^{1/d}\rfloor + 3d,\ 16N + 30\}$ and depth $11L + 18$ such that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \le |f(\boldsymbol{0})| + \omega_f(\sqrt{d})$ and*

$$|f(\boldsymbol{x}) - \phi(\boldsymbol{x})| \le 130\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big), \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta),$$

where $K = \lfloor N^{1/d}\rfloor^2 \lfloor L^{1/d}\rfloor^2 \big\lfloor \lfloor \log_3(N+2)\rfloor^{1/d}\big\rfloor$ 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 $p \in [1, \infty)$. To attain the rate in L^∞-norm, we need to control the approximation error in the trifling region. To this end, we introduce a theorem to deal with the approximation inside the trifling region $\Omega([0,1]^d, K, \delta)$.

Theorem 2.2 (Theorem 3.7 of [43] or Theorem 2.1 of [24])**.** *Given any $\varepsilon > 0$, $N, L, K \in \mathbb{N}^+$, and $\delta \in (0, \frac{1}{3K}]$, assume f is a continuous function in $C([0,1]^d)$ and $\widetilde{\phi}$ can be implemented by a ReLU network with width N and depth L. If*

$$|f(\boldsymbol{x}) - \widetilde{\phi}(\boldsymbol{x})| \le \varepsilon, \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta),$$

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

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

Now we are ready to prove Theorem 1.1 by assuming Theorem 2.1 is true, which will be proved later in Section 3.

Proof of Theorem 1.1. We may assume f is not a constant function since it is a trivial case. Then $\omega_f(r) > 0$ for any $r > 0$. Let us first consider the case $p \in [1, \infty)$. Set $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \big\lfloor \lfloor \log_3(N+2) \rfloor^{1/d} \big\rfloor$ and choose a small $\delta \in (0, \frac{1}{3K}]$ such that

$$\begin{aligned} Kd\delta\big(2|f(\boldsymbol{0})| + 2\omega_f(\sqrt{d})\big)^p &= \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \big\lfloor \lfloor \log_3(N+2) \rfloor^{1/d} \big\rfloor d\delta\big(2|f(\boldsymbol{0})| + 2\omega_f(\sqrt{d})\big)^p \\ &\le \left(\omega_f\Big(\big(N^2 L^2 \log_3(N+2)\big)^{-1/d}\Big)\right)^p. \end{aligned}$$

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

$$\max\big\{8d\lfloor N^{1/d} \rfloor + 3d,\ 16N + 30\big\} \le 16 \max\big\{d\lfloor N^{1/d} \rfloor,\ N + 2\big\}$$

and depth $11L + 18$ such that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \le |f(\boldsymbol{0})| + \omega_f(\sqrt{d})$ and

$$|f(\boldsymbol{x}) - \phi(\boldsymbol{x})| \le 130\sqrt{d}\,\omega_f\Big(\big(N^2 L^2 \log_3(N+2)\big)^{-1/d}\Big), \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta),$$

It follows from $\mu(\Omega([0,1]^d, K, \delta)) \le Kd\delta$ and $\|f\|_{L^\infty([0,1]^d)} \le |f(\boldsymbol{0})| + \omega_f(\sqrt{d})$ that

$$\begin{aligned} \|f - \phi\|^p_{L^p([0,1]^d)} &= \int_{\Omega([0,1]^d, K, \delta)} |f(\boldsymbol{x}) - \phi(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x} + \int_{[0,1]^d \backslash \Omega([0,1]^d, K, \delta)} |f(\boldsymbol{x}) - \phi(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x} \\ &\le Kd\delta\big(2|f(\boldsymbol{0})| + 2\omega_f(\sqrt{d})\big)^p + \left(130\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big)\right)^p \\ &\le \left(\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big)\right)^p + \left(130\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big)\right)^p \\ &\le \left(131\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big)\right)^p. \end{aligned}$$

Hence, $\|f - \phi\|_{L^p([0,1]^d)} \le 131\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big)$.

Next, let us discuss the case $p = \infty$. Set $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \big\lfloor \lfloor \log_3(N+2) \rfloor^{1/d} \big\rfloor$ and choose a small $\delta \in (0, \frac{1}{3K}]$ such that

$$d \cdot \omega_f(\delta) \le \omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big).$$

By Theorem 2.1, there exists a function $\widetilde{\phi}$ implemented by a ReLU network with width $\max\{8d\lfloor N^{1/d}\rfloor+3d,\,16N+30\}$ and depth $11L+18$ such that

$$|f(\boldsymbol{x})-\widetilde{\phi}(\boldsymbol{x})|\le 130\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big)=:\varepsilon,$$

for any $\boldsymbol{x}\in[0,1]^d\backslash\Omega([0,1]^d,K,\delta)$. By Theorem 2.2, there exists a function ϕ implemented by a ReLU network with width

$$3^d\Big(\max\{8d\lfloor N^{1/d}\rfloor+3d,\,16N+30\}+4\Big)\le 3^{d+3}\max\{d\lfloor N^{1/d}\rfloor,\,N+2\}$$

and depth $11L+18+2d$ such that

$$|f(\boldsymbol{x})-\phi(\boldsymbol{x})|\le\varepsilon+d\cdot\omega_f(\delta)\le 131\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big),\quad\text{for any }\boldsymbol{x}\in[0,1]^d.$$

So we finish the proof. □

2.3 Optimality

This section will show that the approximation rates in Theorem 1.1 and Corollary 1.3 are optimal and there is no room to improve for the function class Hölder$([0,1]^d,\alpha,\lambda)$. Therefore, the approximation rate for the whole continuous functions space in terms of width and depth in Theorem 1.1 cannot be improved. A typical method to characterize the optimal approximation theory of neural networks is to study the connection between the approximation error and Vapnik–Chervonenkis (VC) dimension [24, 33, 40, 41, 43]. This method relies on the VC-dimension upper bound given in [13]. In this paper, we adopt this method with several modifications to simplify the proof.

Let us first present the definitions of VC-dimension and related concepts. Let H be a class of functions mapping from a general domain $\mathcal{X}$ to $\{0,1\}$. We say H shatters the set $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}\subseteq\mathcal{X}$ if

$$\Big|\Big\{\big[h(\boldsymbol{x}_1),h(\boldsymbol{x}_2),\cdots,h(\boldsymbol{x}_m)\big]^T\in\{0,1\}^m:h\in H\Big\}\Big|=2^m,$$

where $|\cdot|$ denotes the size of a set. This equation means, given any $\theta_i\in\{0,1\}$ for $i=1,2,\cdots,m$, there exists $h\in H$ such that $h(\boldsymbol{x}_i)=\theta_i$ for all i. For general a function set $\mathscr{F}$ mapping from $\mathcal{X}$ to $\mathbb{R}$, we say $\mathscr{F}$ shatters $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}\subseteq\mathcal{X}$ if $\mathcal{T}\circ\mathscr{F}$ does, where

$$\mathcal{T}(t):=\begin{cases}1, & t\ge 0,\\ 0, & t<0\end{cases}\quad\text{and}\quad \mathcal{T}\circ\mathscr{F}:=\{\mathcal{T}\circ f:f\in\mathscr{F}\}.$$

For any $m\in\mathbb{N}^+$, we define the growth function of H as

$$\Pi_H(m):=\max_{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\in\mathcal{X}}\Big|\Big\{\big[h(\boldsymbol{x}_1),h(\boldsymbol{x}_2),\cdots,h(\boldsymbol{x}_m)\big]^T\in\{0,1\}^m:h\in H\Big\}\Big|.$$

Definition 2.3 (VC-dimension)**.** Let H be a class of functions from $\mathcal{X}$ to $\{0,1\}$. The VC-dimension of H, denoted by VCDim(H), is the size of the largest shattered set, namely,

$$\text{VCDim}(H):=\sup\{m\in\mathbb{N}^+:\Pi_H(m)=2^m\}$$

if $\{m \in \mathbb{N}^+ : \Pi_H(m) = 2^m\}$ is not empty. In the case of $\{m \in \mathbb{N}^+ : \Pi_H(m) = 2^m\} = \varnothing$, we may define $\mathrm{VCDim}(H) = 0$.

Let $\mathscr{F}$ be a class of functions from $\mathcal{X}$ to $\mathbb{R}$. The VC-dimension of $\mathscr{F}$, denoted by $\mathrm{VCDim}(\mathscr{F})$, is defined by $\mathrm{VCDim}(\mathscr{F}) := \mathrm{VCDim}(\mathcal{T} \circ \mathscr{F})$, where

$$\mathcal{T}(t) := \begin{cases} 1, & t \ge 0, \\ 0, & t < 0 \end{cases} \quad \text{and} \quad \mathcal{T} \circ \mathscr{F} := \{\mathcal{T} \circ f : f \in \mathscr{F}\}.$$

In particular, the expression "VC-dimension of a network (architecture)" means the VC-dimension of the function set that consists of all functions implemented by this network (architecture).

We remark that one may also define $\mathrm{VCDim}(\mathscr{F})$ as $\mathrm{VCDim}(\mathscr{F}) := \mathrm{VCDim}(\widetilde{\mathcal{T}} \circ \mathscr{F})$, where

$$\widetilde{\mathcal{T}}(t) := \begin{cases} 1, & t > 0, \\ 0, & t \le 0 \end{cases} \quad \text{and} \quad \widetilde{\mathcal{T}} \circ \mathscr{F} := \{\widetilde{\mathcal{T}} \circ f : f \in \mathscr{F}\}.$$

Note that function spaces generated by networks are closed under linear transformation. Thus, these two definitions of VC-dimension are equivalent.

The theorem below, similar to Theorem 4.17 of [43], reveals the connection between VC-dimension and approximation rate.

Theorem 2.4. *Assume $\mathscr{F}$ is a set of functions mapping from $[0,1]^d$ to $\mathbb{R}$. For any $\varepsilon > 0$, if $\mathrm{VCDim}(\mathscr{F}) \ge 1$ and*

$$\inf_{\phi \in \mathscr{F}} \|\phi - f\|_{L^\infty([0,1]^d)} \le \varepsilon, \quad \text{for any } f \in \text{Hölder}([0,1]^d, \alpha, 1), \tag{2.2}$$

then $\mathrm{VCDim}(\mathscr{F}) \ge (9\varepsilon)^{-d/\alpha}$.

This theorem demonstrates the connection between VC-dimension of $\mathscr{F}$ and the approximation rate using elements of $\mathscr{F}$ to approximate functions in $\text{Hölder}([0,1]^d, \alpha, \lambda)$. To be precise, the VC-dimension of $\mathscr{F}$ determines an approximation rate lower bound $\mathrm{VCDim}(\mathscr{F})^{-\alpha/d}/9$, which is the best possible approximation rate. Denote the best approximation error of functions in $\text{Hölder}([0,1]^d, \alpha, 1)$ approximated by ReLU networks with width N and depth L as

$$\mathcal{E}_{\alpha,d}(N, L) := \sup_{f \in \text{Hölder}([0,1]^d, \alpha, 1)} \left(\inf_{\phi \in \mathcal{NN}(\text{width} \le N;\, \text{depth} \le L)} \|\phi - f\|_{L^\infty([0,1]^d)} \right),$$

We have three remarks listed below.

(i) A large VC-dimension cannot guarantee a good approximation rate. For example, it is easy to verify that

$$\mathrm{VCDim}\Big(\big\{f : f(x) = \cos(ax),\ a \in \mathbb{R}\big\}\Big) = \infty.$$

However, functions in $\big\{f : f(x) = \cos(ax),\ a \in \mathbb{R}\big\}$ cannot approximate Hölder continuous functions well.

(ii) A large VC-dimension is necessary for a good approximation rate, because the best possible approximation rate is controlled by an expression of VC-dimension, as shown in Theorem 2.4. It is shown in Theorem 6 and 8 of [13] that the VC-dimension of ReLU networks has two types of upper bounds: $\mathcal{O}(WL\ln W)$ and $\mathcal{O}(WU)$. Here, W, L, and U are the numbers of parameters, layers, and neurons, respectively. If we let N denote the maximum width of the network, then $W = \mathcal{O}(N^2L)$ and $U = \mathcal{O}(NL)$, implying that

$$WL\ln W = \mathcal{O}\big(N^2L\cdot L\ln(N^2L)\big) = \mathcal{O}\big(N^2L^2\ln(NL)\big)$$

and

$$WU = \mathcal{O}(N^2L\cdot NL) = \mathcal{O}(N^3L^2).$$

If follows that

$$\mathrm{VCDim}\big(\mathcal{NN}(\text{width}\le N;\ \text{depth}\le L)\big) \le \min\Big\{\mathcal{O}\big(N^2L^2\ln(NL)\big), \mathcal{O}(N^3L^2)\Big\},$$

deducing

$$\underbrace{C_1(\alpha,d)\Big(\min\{N^2L^2\ln(NL), N^3L^2\}\Big)^{-\alpha/d} \le \mathcal{E}_{\alpha,d}(N,L)}_{\text{implied by Theorem 2.4}} \underbrace{\le C_2(\alpha,d)\Big(N^2L^2\ln N\Big)^{-\alpha/d}}_{\text{implied by Corollary 1.2 and 1.3}}, \tag{2.3}$$

where $C_1(\alpha,d)$ and $C_2(\alpha,d)$ are two positive constants determined by s,d, and $C_2(s,d)$ can be explicitly expressed.

- When $L = L_0$ is fixed, Equation (2.3) implies

$$C_1(\alpha,d,L_0)(N^2\ln N)^{-\alpha/d} \le \mathcal{E}_{\alpha,d}(N,L_0) \le C_2(\alpha,d,L_0)(N^2\ln N)^{-\alpha/d},$$

  where $C_1(\alpha,d,L_0)$ and $C_2(\alpha,d,L_0)$ are two positive constants determined by α,d,L_0.
- When $N = N_0$ is fixed, Equation (2.3) implies

$$C_1(\alpha,d,N_0)L^{-2\alpha/d} \le \mathcal{E}_{\alpha,d}(N_0,L) \le C_2(\alpha,d,N_0)L^{-2\alpha/d},$$

  where $C_1(\alpha,d,N_0)$ and $C_2(\alpha,d,N_0)$ are two positive constants determined by α,d,N_0.
- It is easy to verify that Equation (2.3) is tight except for the following region

$$\{(N,L)\in\mathbb{N}^2 : C_3(\alpha,d)\le N\le L^{C_4(\alpha,d)}\},$$

  $C_3 = C_3(\alpha,d)$ and $C_4 = C_4(\alpha,d)$ are two positive constants. See Figure 1 for an illustration for the case $C_3 = 1000$ and $C_4 = 1/100$.

Finally, let us present the detailed proof of Theorem 2.4.

Proof of Theorem 2.4. Recall that the VC-dimension of a function set is defined as the size of the largest set of points that this class of functions can shatter. So our goal is to find a subset of $\mathscr{F}$ to shatter $\mathcal{O}(\varepsilon^{-d/\alpha})$ points in $[0,1]^d$, which can be divided into two steps.

- Construct $\{f_\chi : \chi \in \mathscr{B}\} \subseteq \text{Hölder}([0,1]^d, \alpha, 1)$ that scatters $\mathcal{O}(\varepsilon^{-d/\alpha})$ points, where $\mathscr{B}$ is a set defined later.

- Design $\phi_\chi \in \mathscr{F}$, for each $\chi \in \mathscr{B}$, based on f_χ and Equation (2.2) such that $\{\phi_\chi : \chi \in \mathscr{F}\} \subseteq \mathscr{F}$ also shatters $\mathcal{O}(\varepsilon^{-d/\alpha})$ points.

The details of these two steps can be found below.

Step 1: Construct $\{f_\chi : \chi \in \mathscr{B}\} \subseteq \text{Hölder}([0,1]^d, \alpha, 1)$ that scatters $\mathcal{O}(\varepsilon^{-d/\alpha})$ points.

We may assume $\varepsilon \le 2/9$ since the case $\varepsilon > 2/9$ is trivial. In fact, $\varepsilon > 2/9$ implies

$$\text{VCDim}(\mathscr{F}) \ge 1 \ge 1/2 \ge 2^{-d/\alpha} > (9\varepsilon)^{-d/\alpha}.$$

Let $K = \lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor \in \mathbb{N}^+$ and divide $[0,1]^d$ into K^d non-overlapping sub-cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}}$ as follows:

$$Q_{\boldsymbol{\beta}} \coloneqq \Big\{\boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T \in [0,1]^d : x_i \in [\tfrac{\beta_i}{K}, \tfrac{\beta_i+1}{K}],\ i = 1, 2, \cdots, d\Big\},$$

for any index vector $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_d]^T \in \{0, 1, \cdots, K-1\}^d$.

Define a function ζ_Q on $[0,1]^d$ corresponding to $Q = Q(\boldsymbol{x}_0, \eta) \subseteq [0,1]^d$ such that:

- $\zeta_Q(\boldsymbol{x}_0) = (\eta/2)^\alpha/2$;

- $\zeta_Q(\boldsymbol{x}) = 0$ for any $\boldsymbol{x} \notin Q \backslash \partial Q$, where ∂Q is the boundary of Q;

- ζ_Q is linear on the line that connects $\boldsymbol{x}_0$ and $\boldsymbol{x}$ for any $\boldsymbol{x} \in \partial Q$.

Define

$$\mathscr{B} \coloneqq \Big\{\chi : \chi \text{ is a map from } \{0, 1, \cdots, K-1\}^d \text{ to } \{-1, 1\}\Big\}.$$

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

$$f_\chi(\boldsymbol{x}) \coloneqq \sum_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d} \chi(\boldsymbol{\beta}) \zeta_{Q_{\boldsymbol{\beta}}}(\boldsymbol{x}),$$

where $\zeta_{Q_{\boldsymbol{\beta}}}(\boldsymbol{x})$ is the associated function introduced just above. It is easy to check that $\{f_\chi : \chi \in \mathscr{B}\} \subseteq \text{Hölder}([0,1]^d, \alpha, 1)$ can shatter $K^d = \mathcal{O}(\varepsilon^{-d/\alpha})$ points in $[0,1]^d$.

Step 2: Construct $\{\phi_\chi : \chi \in \mathscr{B}\}$ that also scatters $\mathcal{O}(\varepsilon^{-d/\alpha})$ points.

By Equation (2.2), for each $\chi \in \mathscr{B}$, there exists $\phi_\chi \in \mathscr{F}$ such that

$$\|\phi_\chi - f_\chi\|_{L^\infty([0,1]^d)} \le \varepsilon + \varepsilon/81.$$

Let $\mu(\cdot)$ denote the Lebesgue measure of a set. Then, for each $\chi \in \mathscr{B}$, there exists $\mathcal{H}_\chi \subseteq [0,1]^d$ with $\mu(\mathcal{H}_\chi) = 0$ such that

$$|\phi_\chi(\boldsymbol{x}) - f_\chi(\boldsymbol{x})| \le \tfrac{82}{81}\varepsilon, \quad \text{for any } \boldsymbol{x} \in [0,1] \backslash \mathcal{H}_\chi.$$

Set $\mathcal{H} = \cup_{\chi \in \mathscr{B}} \mathcal{H}_\chi$, then we have $\mu(\mathcal{H}) = 0$ and

$$|\phi_\chi(\boldsymbol{x}) - f_\chi(\boldsymbol{x})| \le \tfrac{82}{81}\varepsilon, \quad \text{for any } \chi \in \mathscr{B} \text{ and } \boldsymbol{x} \in [0,1] \backslash \mathcal{H}. \tag{2.4}$$

Since $Q_{\boldsymbol{\beta}}$ has a sidelength $\frac{1}{K} = \frac{1}{\lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor}$, we have, for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ and any $\boldsymbol{x} \in \frac{1}{10} Q_{\boldsymbol{\beta}}$①,

$$|f_\chi(\boldsymbol{x})| = |\zeta_{Q_{\boldsymbol{\beta}}}(\boldsymbol{x})| \ge \tfrac{9}{10} |\zeta_{Q_{\boldsymbol{\beta}}}(\boldsymbol{x}_{Q_{\boldsymbol{\beta}}})| = \tfrac{9}{10} \big(\tfrac{1}{2\lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor}\big)^\alpha / 2 \ge \tfrac{81}{80}\varepsilon, \tag{2.5}$$

where $\boldsymbol{x}_{Q_{\boldsymbol{\beta}}}$ is the center of $Q_{\boldsymbol{\beta}}$.

Note that $(\frac{1}{10} Q_{\boldsymbol{\beta}}) \backslash \mathcal{H}$ is not empty, since $\mu\big((\frac{1}{10} Q_{\boldsymbol{\beta}}) \backslash \mathcal{H}\big) > 0$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Together with Equation (2.4) and (2.5), there exists $\boldsymbol{x}_{\boldsymbol{\beta}} \in (\frac{1}{10} Q_{\boldsymbol{\beta}}) \backslash \mathcal{H}$ such that, for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ and each $\chi \in \mathscr{B}$,

$$|f_\chi(\boldsymbol{x}_{\boldsymbol{\beta}})| \ge \tfrac{81}{80}\varepsilon > \tfrac{82}{81}\varepsilon \ge |f_\chi(\boldsymbol{x}_{\boldsymbol{\beta}}) - \phi_\chi(\boldsymbol{x}_{\boldsymbol{\beta}})|,$$

Hence, $f_\chi(\boldsymbol{x}_{\boldsymbol{\beta}})$ and $\phi_\chi(\boldsymbol{x}_{\boldsymbol{\beta}})$ have the same sign for each $\chi \in \mathscr{B}$ and $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Then $\{\phi_\chi : \chi \in \mathscr{B}\}$ shatters $\big\{\boldsymbol{x}_{\boldsymbol{\beta}} : \boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d\big\}$ since $\{f_\chi : \chi \in \mathscr{B}\}$ shatters $\big\{\boldsymbol{x}_{\boldsymbol{\beta}} : \boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d\big\}$. Therefore,

$$\mathrm{VCDim}(\mathscr{F}) \ge \mathrm{VCDim}\big(\{\phi_\chi : \chi \in \mathscr{B}\}\big) \ge K^d = \lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor^d \ge (9\varepsilon)^{-d/\alpha}, \tag{2.6}$$

where the last inequality comes from the fact $\lfloor x \rfloor \ge x/2 \ge x/(2^{1/\alpha})$ for any $x \in [1,\infty)$ and $\alpha \in (0,1]$. So we finish the proof. □

2.4 Approximation in irregular domain

We extend our analysis to general continuous functions defined on any irregular bounded set in $\mathbb{R}^d$. The key idea is to extend the target function to a hypercube while preserving the modulus of continuity. The extension of continuous (smooth) functions has been widely studied, e.g., [39] for smooth functions and [38] for continuous functions. For simplicity, we use Lemma 4.2 of [33]. The proof can be found therein. For a general set $E \subseteq \mathbb{R}^d$, the modulus of continuity of $f \in C(E)$ is defined via

$$\omega_f^E(r) := \sup\big\{|f(\boldsymbol{x}) - f(\boldsymbol{y})| : \boldsymbol{x}, \boldsymbol{y} \in E,\ \|\boldsymbol{x} - \boldsymbol{y}\|_2 \le r\big\}, \quad \text{for any } r \ge 0.$$

In particular, $\omega_f(\cdot)$ is short of $\omega_f^E(\cdot)$ in the case of $E = [0,1]^d$. Then, Theorem 1.1 can be generalized to $f \in C(E)$ for any bounded set $E \subseteq [-R,R]^d$ with $R > 0$, as shown in the following theorem.

Theorem 2.5. *Given a continuous function $f \in C(E)$ with $E \subseteq [-R,R]^d$ and $R > 0$, for any $N \in \mathbb{N}^+$, $L \in \mathbb{N}^+$, and $p \in [1,\infty]$, there exists a function ϕ implemented by a ReLU network with width $C_1 \max\big\{d\lfloor N^{1/d} \rfloor, N+2\big\}$ and depth $11L + C_2$ such that*

$$\|f - \phi\|_{L^p(E)} \le 131(2R)^{d/p}\sqrt{d}\,\omega_f^E\Big(2R\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big),$$

where $C_1 = 16$ and $C_2 = 18$ if $p \in [1,\infty)$; $C_1 = 3^{d+3}$ and $C_2 = 18 + 2d$ if $p = \infty$.

① $\frac{1}{10} Q_{\boldsymbol{\beta}}$ denotes the closed cube whose sidelength is $1/10$ of that of $Q_{\boldsymbol{\beta}}$ and which shares the same center of $Q_{\boldsymbol{\beta}}$.

Proof. Given any $f \in C(E)$, by Lemma 4.2 of [33] via setting $S = [-R, R]^d$, there exists $g \in C([-R, R]^d)$ such that

- $g(\boldsymbol{x}) = f(\boldsymbol{x})$ for any $\boldsymbol{x} \in E \subseteq S = [-R, R]^d$;
- $\omega_g^S(r) = \omega_f^E(r)$ for any $r \ge 0$.

Define
$$\widetilde{g}(\boldsymbol{x}) \coloneqq g(2R\boldsymbol{x} - R), \quad \text{for any } \boldsymbol{x} \in [0,1]^d.$$
By applying Theorem 1.1 to $\widetilde{g} \in C([0,1]^d)$, there exists a function $\widetilde{\phi}$ implemented by a ReLU network with width $C_1 \max\{d\lfloor N^{1/d}\rfloor, N+2\}$ and depth $11L + C_2$ such that
$$\|\widetilde{\phi} - \widetilde{g}\|_{L^p([0,1]^d)} \le 131\sqrt{d}\,\omega_{\widetilde{g}}\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big),$$
where $C_1 = 16$ and $C_2 = 18$ if $p \in [1,\infty)$; $C_1 = 3^{d+3}$ and $C_2 = 18 + 2d$ if $p = \infty$.

Recall that $f(\boldsymbol{x}) = g(\boldsymbol{x}) = \widetilde{g}(\frac{\boldsymbol{x}+R}{2R})$ for any $\boldsymbol{x} \in E \subseteq S = [-R, R]^d$ and
$$\omega_{\widetilde{g}}(r) = \omega_g^S(2Rr) = \omega_f^E(2Rr), \quad \text{for any } r \ge 0.$$
Define $\phi(\boldsymbol{x}) \coloneqq \widetilde{\phi}(\frac{\boldsymbol{x}+R}{2R}) = \widetilde{\phi} \circ \mathcal{L}(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{R}^d$, where $\mathcal{L}$ is an affine linear map given by $\mathcal{L}(\boldsymbol{x}) = \frac{\boldsymbol{x}+R}{2R}$. Clearly, ϕ can be implemented by a ReLU network with width $C_1 \max\{d\lfloor N^{1/d}\rfloor, N+2\}$ and depth $11L + C_2$, where $C_1 = 16$ and $C_2 = 18$ if $p \in [1,\infty)$; $C_1 = 3^{d+3}$ and $C_2 = 18 + 2d$ if $p = \infty$. Moreover, for any $\boldsymbol{x} \in E \subseteq S = [-R, R]^d$, we have $\frac{\boldsymbol{x}+R}{2R} \in [0,1]^d$, implying
$$\begin{aligned}
\|\phi - f\|_{L^p(E)} &= \|\phi - g\|_{L^p(E)} = \|\widetilde{\phi} \circ \mathcal{L} - \widetilde{g} \circ \mathcal{L}\|_{L^p(E)} \\
&\le \|\widetilde{\phi} \circ \mathcal{L} - \widetilde{g} \circ \mathcal{L}\|_{L^p([-R,R]^d)} = (2R)^{d/p}\|\widetilde{\phi} - \widetilde{g}\|_{L^p([0,1]^d)} \\
&\le 131(2R)^{d/p}\sqrt{d}\,\omega_{\widetilde{g}}\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big) \\
&= 131(2R)^{d/p}\sqrt{d}\,\omega_f^E\Big(2R\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big).
\end{aligned}$$
With the discussion above, we have proved Theorem 2.5. □

3 Proof of Theorem 2.1

We will prove Theorem 2.1 in this section. We first present the key ideas in Section 3.1. The detailed proof is presented in Section 3.3, based on two propositions in Section 3.1, the proofs of which can be found in Section 4.

3.1 Key ideas of proving Theorem 2.1

Given an arbitrary $f \in C([0,1]^d)$, our goal is to construct an almost piecewise constant function ϕ implemented by a ReLU network to approximate f well. To this end, we introduce a piecewise constant function $f_p \approx f$ serving as an intermediate approximant in our construction in the sense that

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

The approximation in $f \approx f_p$ is a simple and standard technique in constructive approximation. The most technical part is to design a ReLU network with the desired width and depth to implement a function ϕ with $\phi \approx f_p$ outside $\Omega([0,1]^d, K, \delta)$. See Figure 4 for an illustration. The introduction of the trifling region is to ease the construction of ϕ, which is a continuous piecewise linear 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 $[0,1]^d \backslash \Omega([0,1]^d, K, \delta)$.

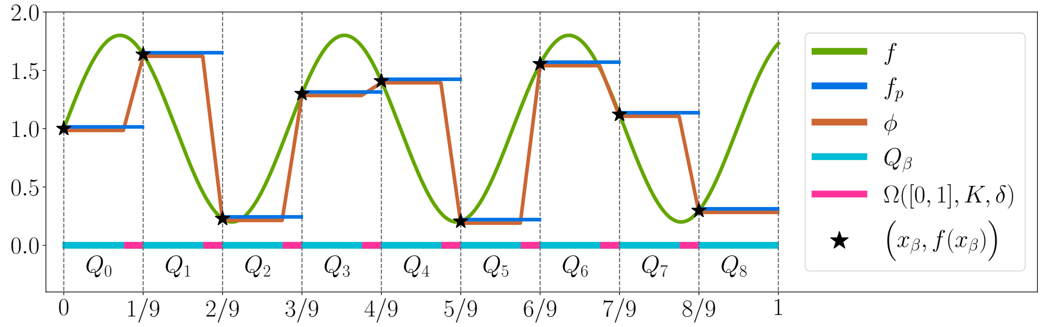


Figure 4: An illustration of f, f_p, ϕ, x_β, Q_β, and the trifling region $\Omega([0,1]^d, K, \delta)$ in the one-dimensional case for $\beta \in \{0, 1, \cdots, K-1\}^d$, where $K = N^2L^2 \log_3(N+2)$ and $d = 1$ with $N = 1$ and $L = 3$. f is the target function; f_p is the piecewise constant function approximating f; ϕ is a function, implemented by a ReLU network, approximating f; and x_β is a representative of Q_β. The measure of $\Omega([0,1]^d, K, \delta)$ can be arbitrarily small as we shall see in the proof of Theorem 1.1.

Now let us discuss the detailed steps of construction.

(1) First, divide $[0,1]^d$ into a union of important regions $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}}$ and the trifling region $\Omega([0,1]^d, K, \delta)$, where each $Q_{\boldsymbol{\beta}}$ is associated with a representative $\boldsymbol{x}_{\boldsymbol{\beta}} \in Q_{\boldsymbol{\beta}}$ such that $f(\boldsymbol{x}_{\boldsymbol{\beta}}) = f_p(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each index vector $\boldsymbol{\beta} \in \{0, 1, \ldots, K-1\}^d$, where $K = \mathcal{O}((N^2L^2 \ln N)^{1/d})$ is the partition number per dimension (see Figure 7 for examples for $d = 1$ and $d = 2$).

(2) Next, we design a vector function $\boldsymbol{\Phi}_1(\boldsymbol{x})$ constructed via

$$\boldsymbol{\Phi}_1(\boldsymbol{x}) = \big[\phi_1(x_1), \phi_1(x_2), \cdots, \phi_1(x_d)\big]^T$$

to project the whole cube $Q_{\boldsymbol{\beta}}$ to a d-dimensional index $\boldsymbol{\beta}$ for each $\boldsymbol{\beta}$, where each one-dimensional function ϕ_1 is a step function implemented by a ReLU network.

(3) The third step is to solve a point fitting problem. To be precise, we construct a function ϕ_2 implemented by a ReLU network to map $\boldsymbol{\beta}$ approximately to $f_p(\boldsymbol{x}_{\boldsymbol{\beta}}) =$

$f(\boldsymbol{x}_{\boldsymbol{\beta}})$. Then $\phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx f_p(\boldsymbol{x}_{\boldsymbol{\beta}}) = f(\boldsymbol{x}_{\boldsymbol{\beta}})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta}$, implying $\phi \coloneqq \phi_2 \circ \boldsymbol{\Phi}_1 \approx f_p \approx f$ on $[0,1]^d \backslash \Omega([0,1]^d, K, \delta)$. We would like to point out that we only need to care about the values of ϕ_2 at a set of points $\{0,1,\cdots,K-1\}^d$ in the construction of ϕ_2 according to our design $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$ as illustrated in Figure 5. Therefore, it is not necessary to care about the values of ϕ_2 sampled outside the set $\{0,1,\cdots,K-1\}^d$, which is a key point to ease the design of a ReLU network to implement ϕ_2 as we shall see later.

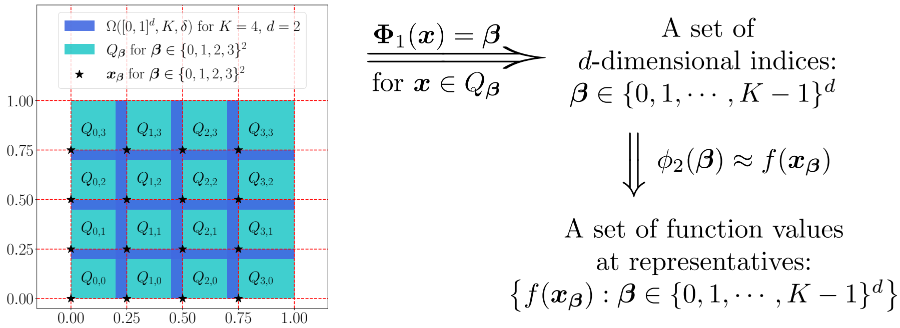


Figure 5: An illustration of the desired function $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$. Note that $\phi \approx f$ on $[0,1]^d \backslash \Omega([0,1]^d, K, \delta)$, since $\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}}) \approx f(\boldsymbol{x})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

We remark that in Figure 5, we have

$$\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \overset{\mathscr{E}_1}{\approx} f(\boldsymbol{x}_{\boldsymbol{\beta}}) \overset{\mathscr{E}_2}{\approx} f(\boldsymbol{x})$$

for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Thus, $\phi - f$ is bounded by $\mathscr{E}_1 + \mathscr{E}_2$ outside the trifling region. Observe that $\mathscr{E}_2$ is bounded by $\omega_f(\sqrt{d}/K)$. As we shall see later in Section 3.3, $\mathscr{E}_1$ can also be bounded by $\omega_f(\sqrt{d}/K)$ by applying Proposition 3.2. Hence, $\phi - f$ is controlled by $2\omega_f(\sqrt{d}/K)$ outside the trifling region, which deduces the desired error.

Finally, we discuss how to implement $\boldsymbol{\Phi}_1$ and ϕ_2 by deep ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ using two propositions as we shall prove in Section 4.2 and 4.3 later. We first show how to construct a ReLU network with the desired width and depth by Proposition 3.1 to implement a one-dimensional step function ϕ_1. Then $\boldsymbol{\Phi}_1$ can be attained via defining

$$\boldsymbol{\Phi}_1(\boldsymbol{x}) = \big[\phi_1(x_1), \phi_1(x_2), \cdots, \phi_1(x_d)\big]^T, \quad \text{for any } \boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T \in \mathbb{R}^d.$$

Proposition 3.1. *For any $N, L, d \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{3K}]$ with*

$$K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor \lfloor n^{1/d} \rfloor, \quad \text{where } n = \lfloor \log_3(N+2) \rfloor,$$

there exists a one-dimensional function ϕ implemented by a ReLU network with width $8\lfloor N^{1/d} \rfloor + 3$ and depth $2\lfloor L^{1/d} \rfloor + 5$ such that

$$\phi(x) = k, \quad \text{if } x \in \big[\tfrac{k}{K}, \tfrac{k+1}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}\big] \text{ for } k = 0, 1, \cdots, K-1.$$

The setting $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor n^{1/d} \rfloor = \mathcal{O}(N^{2/d} L^{2/d} n^{1/d})$ is not neat here, but it is very convenient for later use. The construction of ϕ_2 is a direct result of Proposition 3.2 below, the proof of which relies on the bit extraction technique in [3].

Proposition 3.2. *Given any $\varepsilon > 0$ and arbitrary $N, L, J \in \mathbb{N}^+$ with $J \le N^2 L^2 \lfloor \log_3(N+2) \rfloor$, assume $y_j \ge 0$ for $j = 0, 1, \cdots, J-1$ are samples with*

$$|y_j - y_{j-1}| \le \varepsilon, \quad \text{for } j = 1, 2, \cdots, J-1.$$

Then there exists $\phi \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16N + 30;\ \text{depth} \le 6L + 10;\ \#\text{output} = 1)$ such that

(i) $|\phi(j) - y_j| \le \varepsilon$ for $j = 0, 1, \cdots, J-1$.

(ii) $0 \le \phi(x) \le \max\{y_j : j = 0, 1, \cdots, J-1\}$ for any $x \in \mathbb{R}$.

3.2 Construction of final network

We will discuss the construction of the final network approximating the target function with the same setting as in Section 3.1. There are two main parts: 1) Construct the final network architecture based on Proposition 3.1 and 3.2; 2) Implement the network architectures in Proposition 3.1 and 3.2.

Final network architecture based on Proposition 3.1 and 3.2

By the idea mentioned in Figure 5, the final network architecture can be implemented by Figure 6.

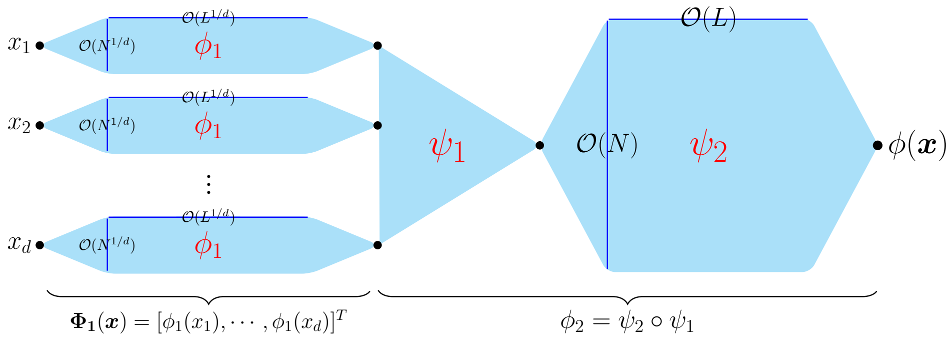


Figure 6: An illustration of the final network architecture with width $\max\{\mathcal{O}(dN^{1/d}), \mathcal{O}(N)\}$ and depth $\mathcal{O}(L)$. $\psi_1 : \mathbb{R}^d \to \mathbb{R}$ is a linear function. ϕ_1 and ψ_2 are implemented via Proposition 3.1 and 3.2, respectively.

Note that ϕ_1 in Figure 6 is a step function mapping $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot \mathbb{1}_{\{k \le K-1\}}]$ to k. It can be easily implemented via Proposition 3.1. Clearly, by defining $\boldsymbol{\Phi}_1(\boldsymbol{x}) = \big[\phi_1(x_1), \phi_1(x_2), \cdots, \phi_1(x_d)\big]^T$, $\boldsymbol{\Phi}_1$ maps $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ to $\boldsymbol{\beta}$.

As shown in Figure 3.1, we need to design a network to compute ϕ_2 mapping $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ approximately to $f(\boldsymbol{x}_{\boldsymbol{\beta}})$. To this end, we first construct a **linear** function $\psi_1 : \mathbb{R}^d \to \mathbb{R}$ mapping $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ to $\mathbb{R}$ for the purpose of converting a d-dimensional point-fitting problem to a one-dimensional one, and then construct a network to compute ψ_2 with $\psi_2(\psi_1(\boldsymbol{\beta})) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}})$ via applying Proposition 3.2. Thus, we have $\phi_2(\boldsymbol{\beta}) \coloneqq \psi_2 \circ \psi(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}})$ as desired.

Network architectures in Proposition 3.1 and 3.2

To prove Proposition 3.1, we need to construct a ReLU network with width $\mathcal{O}(N^{1/d})$ and depth $\mathcal{O}(L^{1/d})$ to compute a step function with $\mathcal{O}\big((N^2L^2\ln N)^{1/d}\big)$ "steps" outside the trifling region. It is easy to construct a ReLU network with $\mathcal{O}(W)$ parameters to compute a step function with W "steps" outside a small region. As we shall see later in Section 4.2, the composition architecture of ReLU networks can help to implement step functions with much more "steps". Refer to Section 4.2 for the detailed proof of Proposition 3.1.

Proposition 3.2 essentially solves a point-fitting problem with $N^2L^2\lfloor\log_3(N+2)\rfloor$ points via a ReLU network with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$. Set $M = N^2L$, $\widehat{L} = L\lfloor\log_3(N+2)\rfloor$, and represent $j \in \{0,1,\cdots,M\widehat{L}-1\}$ via $j = m\widehat{L}+k$, where $m \in \{0,1,\cdots,M-1\}$ and $k \in \{0,1,\cdots,\widehat{L}-1\}$.

Define $a_{m,k} = \lfloor y_{m,k}/\varepsilon\rfloor$ where $y_{m,k} = y_{m\widehat{L}+k}$. Then

$$|a_{m,k}\varepsilon - y_{m,k}| = \big|\lfloor y_{m,k}/\varepsilon\rfloor\varepsilon - y_{m,k}\big| \le \varepsilon.$$

It suffices to prove $\phi(m,k) = a_{m,k}$. The assumption $|y_j - y_{j-1}| < \varepsilon$ implies that $b_{m,k} \coloneqq a_{m,k} - a_{m,k-1} \in \{-1,0,1\}$. Thus, there exist $c_{m,k} \in \{0,1\}$ and $d_{m,k} \in \{0,1\}$ such that $b_{m,k} = c_{m,k} - d_{m,k}$.

Note that

$$a_{m,k} = a_{m,0} + \sum_{j=1}^{k}(a_{m,j} - a_{m,j-1}) = a_{m,0} + \sum_{j=1}^{k} b_{m,j} = a_{m,0} + \sum_{j=1}^{k} c_{m,j} - \sum_{j=1}^{k} d_{m,j}.$$

It is easy to construct a ReLU network with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ ($\mathcal{O}(N^2L)$ parameters in total) to compute ϕ_1 such that $\phi_1(m) = a_{m,0}$. By the bit extraction technique in [3], one could construct $\phi_2, \phi_3 \in \mathcal{NN}(\text{width} \le \mathcal{O}(N);\ \text{depth} \le \mathcal{O}(L))$ such that $\phi_2(m,k) = \sum_{j=1}^{k} c_{m,j}$ and $\phi_3(m,k) = \sum_{j=1}^{k} d_{m,j}$. Thus, $\phi(m,k) \coloneqq \phi_1(m) + \phi_2(m,k) - \phi_3(m,k) = a_{m,k}$ as desired. In order to use the bit extraction technique (two types of bits 0 or 1) to solve the given point-fitting problem, we simplify the target as discussed above. That is,

$$\begin{aligned}\text{positive number } y_{m,k} \longrightarrow \text{integer } a_{m,k} = \lfloor y_{m,k}/\varepsilon\rfloor \longrightarrow\ & b_{m,k} = a_{m,k} - a_{m,k-1} \in \{-1,0,1\}\\ \longrightarrow\ & b_{m,k} = c_{m,k} - d_{m,k} \text{ with } c_{m,k}, d_{m,k-k} \in \{0,1\}.\end{aligned}$$

The detailed proof of Proposition 3.2 can be found in Section 4.3.

3.3 Detailed proof

We essentially construct an almost piecewise constant function implemented by a ReLU network with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ to approximate f. We may assume f is not a constant function since it is a trivial case. Then $\omega_f(r) > 0$ for any $r > 0$. It is clear that $|f(\boldsymbol{x}) - f(\boldsymbol{0})| \le \omega_f(\sqrt{d})$ for any $\boldsymbol{x} \in [0,1]^d$. Define $\widetilde{f} = f - f(\boldsymbol{0}) + \omega_f(\sqrt{d})$, then $0 \le \widetilde{f}(\boldsymbol{x}) \le 2\omega_f(\sqrt{d})$ for any $\boldsymbol{x} \in [0,1]^d$.

Let $M = N^2 L$, $n = \lfloor \log_3(N+2) \rfloor$, $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{1/d} \rfloor^2 \lfloor n^{1/d} \rfloor$, and δ be an arbitrary number in $(0, \frac{1}{3K}]$. The proof can be divided into four steps as follows:

1. Normalize f as $\widetilde{f}$, divide $[0,1]^d$ into a union of sub-cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$ and the trifling region $\Omega([0,1]^d, K, \delta)$, and denote $\boldsymbol{x}_{\boldsymbol{\beta}}$ as the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm;

2. Construct a sub-network to implement a vector function $\boldsymbol{\Phi}_1$ projecting the whole cube $Q_{\boldsymbol{\beta}}$ to the d-dimensional index $\boldsymbol{\beta}$ for each $\boldsymbol{\beta}$, i.e., $\boldsymbol{\Phi}_1(\boldsymbol{x}) = \boldsymbol{\beta}$ for all $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$;

3. Construct a sub-network to implement a function ϕ_2 mapping the index $\boldsymbol{\beta}$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$. 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 $\{0,1,\cdots,K-1\}^d$ to an auxiliary set $\mathcal{A}_1 \subseteq \left\{\frac{j}{2K^d} : j = 0,1,\cdots,2K^d\right\}$ defined later (see Figure 8 for an 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 $\{\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})\}_{\boldsymbol{\beta}}$, i.e., $g \circ \psi_1(\boldsymbol{\beta}) = \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$; 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 $\phi_2(\boldsymbol{\beta}) = \psi_2 \circ \psi_1(\boldsymbol{\beta}) \approx g \circ \psi_1(\boldsymbol{\beta}) = \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$;

4. Construct the final network to implement the desired function ϕ such that $\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) + f(\boldsymbol{0}) - \omega_f(\sqrt{d}) \approx \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}}) + f(\boldsymbol{0}) - \omega_f(\sqrt{d}) = f(\boldsymbol{x}_{\boldsymbol{\beta}}) \approx f(\boldsymbol{x})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

The details of these steps can be found below.

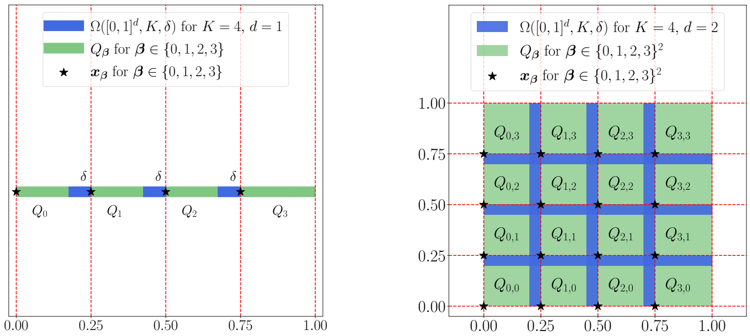
Step 1: Divide $[0,1]^d$ into $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$ and $\Omega([0,1]^d, K, \delta)$.

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

$$Q_{\boldsymbol{\beta}} := \left\{\boldsymbol{x} = [x_1,\cdots,x_d]^T \in [0,1]^d : x_i \in \left[\tfrac{\beta_i}{K}, \tfrac{\beta_i+1}{K} - \delta \cdot \mathbb{1}_{\{\beta_i \le K-2\}}\right],\ i = 1,\cdots,d\right\}$$

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

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



(a) (b)

Figure 7: Illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. (a) $K=4$ and $d=1$. (b) $K=4$ and $d=2$.

see Figure 7 for illustrations.

Step 2: Construct $\boldsymbol{\Phi}_1$ mapping $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ to $\boldsymbol{\beta}$.

By Proposition 3.1, there exists $\phi_1 \in \mathcal{NN}(\text{width} \le 8\lfloor N^{1/d}\rfloor + 3;\ \text{depth} \le 2\lfloor L^{1/d}\rfloor + 5)$ such that

$$\phi_1(x) = k, \quad \text{if } x \in \big[\tfrac{k}{K}, \tfrac{k+1}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}\big] \text{ for } k = 0,1,\cdots,K-1.$$

It follows that $\phi_1(x_i) = \beta_i$ if $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T \in Q_{\boldsymbol{\beta}}$ for each $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_d]^T$.

By defining

$$\boldsymbol{\Phi}_1(\boldsymbol{x}) \coloneqq \big[\phi_1(x_1), \phi_1(x_2), \cdots, \phi_1(x_d)\big]^T, \quad \text{for any } \boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T \in \mathbb{R}^d,$$

we have $\boldsymbol{\Phi}_1(\boldsymbol{x}) = \boldsymbol{\beta}$ if $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

Step 3: Construct ϕ_2 mapping $\boldsymbol{\beta}$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$.

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

$$\mathcal{A}_1 \coloneqq \Big\{\tfrac{i}{K^{d-1}} + \tfrac{k}{2K^d} : i = 0,1,\cdots,K^{d-1}-1 \quad \text{and} \quad k = 0,1,\cdots,K-1\Big\}$$

and

$$\mathcal{A}_2 \coloneqq \Big\{\tfrac{i}{K^{d-1}} + \tfrac{K+k}{2K^d} : i = 0,1,\cdots,K^{d-1}-1 \quad \text{and} \quad k = 0,1,\cdots,K-1\Big\}.$$

Clearly, $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\} = \{\tfrac{j}{2K^d} : j = 0,1,\cdots,2K^d\}$ and $\mathcal{A}_1 \cap \mathcal{A}_2 = \varnothing$. See Figure 7 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 $\{0,1,\cdots,K-1\}^d$ to $\mathcal{A}_1$.

Inspired by the binary representation, we define

$$\psi_1(\boldsymbol{x}) \coloneqq \frac{x_d}{2K^d} + \sum_{i=1}^{d-1} \frac{x_i}{K^i}, \quad \text{for any } \boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T \in \mathbb{R}^d. \tag{3.1}$$

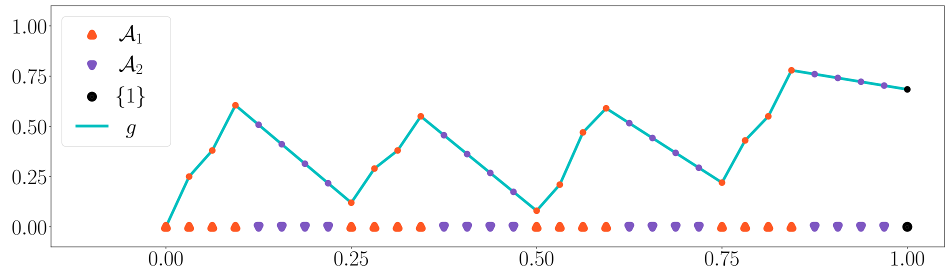


Figure 8: An illustration of $\mathcal{A}_1$, $\mathcal{A}_2$, $\{1\}$, and g for $d = 2$ and $K = 4$.

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

$$\begin{aligned}&\Big\{\tfrac{\beta_d}{2K^d} + \sum_{i=1}^{d-1} \tfrac{\beta_i}{K^i} : \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d\Big\}\\ = &\Big\{\tfrac{i}{K^{d-1}} + \tfrac{k}{2K^d} : i = 0, 1, \cdots, K^{d-1}{-}1 \quad \text{and} \quad k = 0, 1, \cdots, K-1\Big\} = \mathcal{A}_1.\end{aligned}$$

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

Let $g : [0, 1] \to \mathbb{R}$ be a continuous piecewise linear function with a set of breakpoints $\big\{\frac{j}{2K^d} : j = 0, 1, \cdots, 2K^d\big\} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$ 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

$$g(\psi_1(\boldsymbol{\beta})) = \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}}), \quad \text{for any } \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d; \tag{3.2}$$

- At the breakpoint 1, let $g(1) = \widetilde{f}(\mathbf{1})$, where $\mathbf{1} = [1, 1, \cdots, 1]^T \in \mathbb{R}^d$;

- The values of g at the breakpoints in $\mathcal{A}_2$ are assigned to reduce the variation of g, which is a requirement of applying Proposition 3.2. Note that

$$\{\tfrac{i}{K^{d-1}} - \tfrac{K+1}{2K^d},\ \tfrac{i}{K^{d-1}}\} \subseteq \mathcal{A}_1 \cup \{1\}, \quad \text{for } i = 1, 2, \cdots, K^{d-1},$$

  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 $i = 1, 2, \cdots, K^{d-1}$. 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 $i = 1, 2, \cdots, K^{d-1}$, 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 8 for an example) and satisfies

$$\big|g(\tfrac{j}{2K^d}) - g(\tfrac{j-1}{2K^d})\big| \le \max\big\{\omega_f(\tfrac{1}{K}), \omega_f(\sqrt{d})/K\big\} \le \omega_f(\tfrac{\sqrt{d}}{K}), \quad \text{for } j = 1, 2, \cdots, 2K^d,$$

and

$$0 \le g(\tfrac{j}{2K^d}) \le 2\omega_f(\sqrt{d}), \quad \text{for } j = 0, 1, \cdots, 2K^d.$$

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

Note that

$$2K^d = 2\big(\lfloor N^{1/d}\rfloor^2\lfloor L^{1/d}\rfloor^2\lfloor n^{1/d}\rfloor\big)^d \le 2\big(N^2L^2n\big) \le N^2\lceil\sqrt{2}L\rceil^2\lfloor\log_3(N+2)\rfloor.$$

By Proposition 3.2 (set $y_j = g(\frac{j}{2K^2})$ and $\varepsilon = \omega_f(\frac{\sqrt{d}}{K}) > 0$ therein), there exists

$$\widetilde{\psi}_2 \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16N+30;\ \text{depth} \le 6\lceil\sqrt{2}L\rceil + 10;\ \#\text{output} = 1)$$

such that

$$|\widetilde{\psi}_2(j) - g(\tfrac{j}{2K^d})| \le \omega_f(\tfrac{\sqrt{d}}{K}), \quad \text{for } j = 0,1,\cdots,2K^d-1,$$

and

$$0 \le \widetilde{\psi}_2(x) \le \max\{g(\tfrac{j}{2K^d}) : j = 0,1,\cdots,2K^d-1\} \le 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}.$$

By defining $\psi_2(x) \coloneqq \widetilde{\psi}_2(2K^d x)$ for any $x \in \mathbb{R}$, we have $\psi_2 \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16N+30;\ \text{depth} \le 6\lceil\sqrt{2}L\rceil + 10;\ \#\text{output} = 1)$,

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

and

$$|\psi_2(\tfrac{j}{2K^d}) - g(\tfrac{j}{2K^d})| = |\widetilde{\psi}_2(j) - g(\tfrac{j}{2K^d})| \le \omega_f(\tfrac{\sqrt{d}}{K}), \quad \text{for } j = 0,1,\cdots,2K^d-1. \tag{3.4}$$

Let us end Step 3 by defining the desired function ϕ_2 as $\phi_2 \coloneqq \psi_2 \circ \psi_1$. Note that $\psi_1 : \mathbb{R}^d \to \mathbb{R}$ is a linear function and $\psi_2 \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16N+30;\ \text{depth} \le 6\lceil\sqrt{2}L\rceil + 10;\ \#\text{output} = 1)$. Thus, $\phi_2 \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16N+30;\ \text{depth} \le 6\lceil\sqrt{2}L\rceil + 10;\ \#\text{output} = 1)$. By Equation (3.2) and (3.4), we have

$$|\phi_2(\boldsymbol{\beta}) - \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})| = |\psi_2(\psi_1(\boldsymbol{\beta})) - g(\psi_1(\boldsymbol{\beta}))| \le \omega_f(\tfrac{\sqrt{d}}{K}), \tag{3.5}$$

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

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

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

Define $\phi \coloneqq \phi_2 \circ \boldsymbol{\Phi}_1 + f(\boldsymbol{0}) - \omega_f(\sqrt{d})$. Since $\phi_1 \in \mathcal{NN}(\text{width} \le 8\lfloor N^{1/d}\rfloor + 3;\ \text{depth} \le 2\lfloor L^{1/d}\rfloor + 5])$, we have $\boldsymbol{\Phi}_1 \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \le 8d\lfloor N^{1/d}\rfloor + 3d;\ \text{depth} \le 2L + 5;\ \#\text{output} = d)$. If follows from the fact $\lceil\sqrt{2}L\rceil \le \lceil\frac{3}{2}L\rceil \le \frac{3}{2}L + \frac{1}{2}$ that $6\lceil\sqrt{2}L\rceil + 10 \le 9L + 13$, implying

$$\begin{aligned}\phi_2 &\in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16N+30;\ \text{depth} \le 6\lceil\sqrt{2}L\rceil + 10;\ \#\text{output} = 1)\\ &\subseteq \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16N+30;\ \text{depth} \le 9L+13;\ \#\text{output} = 1).\end{aligned}$$

Thus, $\phi = \phi_2 \circ \boldsymbol{\Phi}_1 + f(\boldsymbol{0}) - \omega_f(\sqrt{d})$ is in

$$\mathcal{NN}\big(\text{width} \le \max\{8d\lfloor N^{1/d}\rfloor + 3d, 16N+30\};\ \text{depth} \le (2L+5) + (9L+13) = 11L+18\big).$$

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

$$\begin{aligned}|f(\boldsymbol{x}) - \phi(\boldsymbol{x})| &= |\widetilde{f}(\boldsymbol{x}) - \phi_2(\boldsymbol{\Phi}_1(\boldsymbol{x}))| = |\widetilde{f}(\boldsymbol{x}) - \phi_2(\boldsymbol{\beta})| \\ &\le |\widetilde{f}(\boldsymbol{x}) - \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})| + |\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}}) - \phi_2(\boldsymbol{\beta})| \\ &\le \omega_f(\tfrac{\sqrt{d}}{K}) + \omega_f(\tfrac{\sqrt{d}}{K}) \le 2\omega_f\Big(64\sqrt{d}\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big),\end{aligned}$$

where the last inequality comes from the fact

$$K = \lfloor N^{1/d}\rfloor^2 \lfloor L^{1/d}\rfloor^2 \lfloor n^{1/d}\rfloor \ge \frac{N^{2/d}L^{2/d}n^{1/d}}{32} = \frac{N^{2/d}L^{2/d}\lfloor\log_3(N+2)\rfloor^{1/d}}{32} \ge \frac{(N^2L^2\log_3(N+2))^{1/d}}{64},$$

for any $N, L \in \mathbb{N}^+$. Recall the fact $\omega_f(j\cdot r) \le j\cdot\omega_f(r)$ for any $j \in \mathbb{N}^+$ and $r \in [0,\infty)$. Therefore, for any $\boldsymbol{x} \in \bigcup_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d} Q_{\boldsymbol{\beta}} = [0,1]^d \backslash \Omega([0,1]^d, K, \delta)$, we have

$$\begin{aligned}|f(\boldsymbol{x}) - \phi(\boldsymbol{x})| &\le 2\omega_f\Big(64\sqrt{d}\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big) \\ &\le 2\lceil 64\sqrt{d}\rceil\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big) \\ &\le 130\sqrt{d}\,\omega_f\Big(\big(N^2L^2\log_3(N+2)\big)^{-1/d}\Big).\end{aligned}$$

It remains to show the upper bound of ϕ. By Equation (3.6) and $\phi = \phi_2\circ\boldsymbol{\Phi}_1 + f(\mathbf{0}) - \omega_f(\sqrt{d})$, it holds that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \le |f(\mathbf{0})| + \omega_f(\sqrt{d})$. Thus, we finish the proof.

4 Proofs of propositions in Section 3.1

In this section, we will prove Proposition 3.1 and 3.2. We first introduce several basic results of ReLU networks. Next, we prove these two propositions based on these basic results.

4.1 Basic results of ReLU networks

To simplify the proofs of two propositions in Section 3.1, we introduce three lemmas below, which are basic results of ReLU networks

Lemma 4.1. *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 $x_0 < x_1 < \cdots < x_{N_1(N_2+1)}$ and $y_i \ge 0$ for $i = 0,1,\cdots,N_1(N_2+1)$, there exists $\phi \in \mathcal{NN}(\#\text{input} = 1;\ \text{widthvec} = [2N_1, 2N_2+1];\ \#\text{output} = 1)$ satisfying the following conditions.*

(i) $\phi(x_i) = y_i$ for $i = 0,1,\cdots,N_1(N_2+1)$.

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

Lemma 4.2. *Given any $N, L, d \in \mathbb{N}^+$, it holds that*

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

Lemma 4.3. *For any $n \in \mathbb{N}^+$, it holds that*

$$\mathrm{CPwL}\big(\mathbb{R}, n\big) \subseteq \mathcal{NN}(\#\mathrm{input} = 1;\ \mathrm{widthvec} = [n+1];\ \#\mathrm{output} = 1). \tag{4.1}$$

Lemma 4.1 is a part of Theorem 3.2 in [43] or Lemma 2.2 in [32]. Lemma 4.1 is Theorem 3.1 in [43] or Lemma 3.4 in [32]. It remains to prove Lemma 4.3.

Proof of Lemma 4.3. We use the mathematics induction to prove Equation (4.1). First, consider the case $n = 1$. Given any $f \in \mathrm{CPwL}\big(\mathbb{R}, n\big)$, there exist $a_1, a_2, x_0 \in \mathbb{R}$ such that

$$f(x) = \begin{cases} a_1(x - x_0) + f(x_0), & \text{if } \ x \ge x_0, \\ a_2(x_0 - x) + f(x_0), & \text{if } \ x < x_0. \end{cases}$$

Thus, $f(x) = a_1\sigma(x - x_0) + a_2\sigma(x_0 - x) + f(x_0)$ for any $x \in \mathbb{R}$, implying

$$f \in \mathcal{NN}(\#\mathrm{input} = 1;\ \mathrm{widthvec} = [2];\ \#\mathrm{output} = 1).$$

Thus, Equation (4.1) holds for $n = 1$.

Now assume Equation (4.1) holds for $n = k \in \mathbb{N}^+$, we would like to show it is also true for $n = k+1$. Given any $f \in \mathrm{CPwL}\big(\mathbb{R}, k+1\big)$, we may assume the biggest breakpoint of f is x_0 since it is trivial for the case that f has no breakpoint. Denote the slopes of the linear pieces left and right next to x_0 by a_1 and a_2, respectively. Define

$$\widetilde{f}(x) \coloneqq f(x) - (a_2 - a_1)\sigma(x - x_0), \qquad \text{for any } x \in \mathbb{R}.$$

Then $\widetilde{f}$ has at most k breakpoints. By the induction hypothesis, we have

$$\widetilde{f} \in \mathrm{CPwL}\big(\mathbb{R}, k\big) \subseteq \mathcal{NN}(\#\mathrm{input} = 1;\ \mathrm{widthvec} = [k+1];\ \#\mathrm{output} = 1).$$

Thus, there exist $w_{0,j}, b_{0,j}, w_{1,j}, b_1$ for $j = 1, 2, \cdots, k+1$ such that

$$\widetilde{f}(x) = \sum_{j=1}^{k+1} w_{1,j}\sigma(w_{0,j}x + b_{0,j}) + b_1, \quad \text{for any } x \in \mathbb{R}.$$

Therefore, for any $x \in \mathbb{R}$, we have

$$f(x) = (a_2 - a_1)\sigma(x - x_0) + \widetilde{f}(x) = (a_2 - a_1)\sigma(x - x_0) + \sum_{j=1}^{k} w_{1,j}\sigma(w_{0,j}x + b_{0,j}) + b_1,$$

implying $f \in \mathcal{NN}(\#\mathrm{input} = 1;\ \mathrm{widthvec} = [k+1];\ \#\mathrm{output} = 1)$. Thus, Equation (4.1) holds for $k+1$, which means we finish the induction process. So we complete the proof. $\square$

4.2 Proof of Proposition 3.1

Now, let us present the detailed proof of Proposition 3.1. Denote $K = \widetilde{M} \cdot \widetilde{L}$, where $\widetilde{M} = \lfloor N^{1/d}\rfloor^2 \lfloor L^{1/d}\rfloor$, $n = \lfloor \log_3(N+2)\rfloor$, and $\widetilde{L} = \lfloor L^{1/d}\rfloor \lfloor n^{1/d}\rfloor$. Consider the sample set

$$\begin{aligned}\big\{(1, \widetilde{M} - 1), (2, 0)\big\} &\bigcup \big\{\big(\tfrac{m}{\widetilde{M}}, m\big) : m = 0, 1, \cdots, \widetilde{M} - 1\big\} \\ &\bigcup \big\{\big(\tfrac{m+1}{\widetilde{M}} - \delta, m\big) : m = 0, 1, \cdots, \widetilde{M} - 2\big\}.\end{aligned}$$

Its size is

$$2\widetilde{M}+1 = 2\lfloor N^{1/d}\rfloor^2\lfloor L^{1/d}\rfloor + 1 = \lfloor N^{1/d}\rfloor \cdot \Big(\big(2\lfloor N^{1/d}\rfloor\lfloor L^{1/d}\rfloor - 1\big) + 1 \Big) + 1.$$

By Lemma 4.1 (set $N_1 = \lfloor N^{1/d}\rfloor$ and $N_2 = 2\lfloor N^{1/d}\rfloor\lfloor L^{1/d}\rfloor - 1$ therein), there exists

$$\begin{aligned}\phi_1 &\in \mathcal{NN}\big(\text{widthvec} = \big[2\lfloor N^{1/d}\rfloor, 2(2\lfloor N^{1/d}\rfloor\lfloor L^{1/d}\rfloor - 1) + 1\big]\big) \\ &= \mathcal{NN}\big(\text{widthvec} = \big[2\lfloor N^{1/d}\rfloor, 4\lfloor N^{1/d}\rfloor\lfloor L^{1/d}\rfloor - 1\big]\big)\end{aligned}$$

such that

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

Then, for $m = 0, 1, \cdots, \widetilde{M}-1$, we have

$$\phi_1(x) = m, \quad \text{for any } x \in [\tfrac{m}{\widetilde{M}}, \tfrac{m+1}{\widetilde{M}} - \delta \cdot \mathbb{1}_{\{m \le \widetilde{M}-2\}}]. \tag{4.2}$$

Now consider the another sample set

$$\begin{aligned}\big\{(\tfrac{1}{\widetilde{M}}, \widetilde{L}-1), (2,0)\big\} \bigcup \big\{(\tfrac{\ell}{\widetilde{M}\widetilde{L}}, \ell) : \ell = 0, 1, \cdots, \widetilde{L}-1\big\} \\ \bigcup \big\{(\tfrac{\ell+1}{\widetilde{M}\widetilde{L}} - \delta, \ell) : \ell = 0, 1, \cdots, \widetilde{L}-2\big\}.\end{aligned}$$

Its size is

$$2\widetilde{L}+1 = 2\lfloor L^{1/d}\rfloor\lfloor n^{1/d}\rfloor + 1 = \lfloor n^{1/d}\rfloor \cdot \big((2\lfloor L^{1/d}\rfloor - 1) + 1 \big) + 1.$$

By Lemma 4.1 (set $N_1 = \lfloor n^{1/d}\rfloor$ and $N_2 = 2\lfloor L^{1/d}\rfloor - 1$ therein), there exists

$$\begin{aligned}\phi_2 &\in \mathcal{NN}\big(\text{widthvec} = \big[2\lfloor n^{1/d}\rfloor, 2(2\lfloor L^{1/d}\rfloor - 1) + 1\big]\big) \\ &= \mathcal{NN}\big(\text{widthvec} = \big[2\lfloor n^{1/d}\rfloor, 4\lfloor L^{1/d}\rfloor - 1\big]\big)\end{aligned}$$

such that

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

It follows that, for $m = 0, 1, \cdots, \widetilde{M}-1$ and $\ell = 0, 1, \cdots, \widetilde{L}-1$,

$$\phi_2(x - \tfrac{m}{\widetilde{M}}) = \ell, \quad \text{for any } x \in [\tfrac{m\widetilde{L}+\ell}{\widetilde{M}\widetilde{L}}, \tfrac{m\widetilde{L}+\ell+1}{\widetilde{M}\widetilde{L}} - \delta \cdot \mathbb{1}_{\{\ell \le \widetilde{L}-2\}}]. \tag{4.3}$$

$K = \widetilde{M} \cdot \widetilde{L}$ implies any $k \in \{0, 1, \cdots, K-1\}$ can be unique represented by $k = m\widetilde{L}+\ell$ for $m = 0, 1, \cdots, \widetilde{M}-1$ and $\ell = 0, 1, \cdots, \widetilde{L}-1$. Then the desired function ϕ can be implemented by a ReLU network shown in Figure 9.

Clearly,

$$\phi(x) = k, \quad \text{if } x \in [\tfrac{k}{K}, \tfrac{k}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}], \quad \text{for any } k \in \{0, 1, \cdots, K-1\}.$$

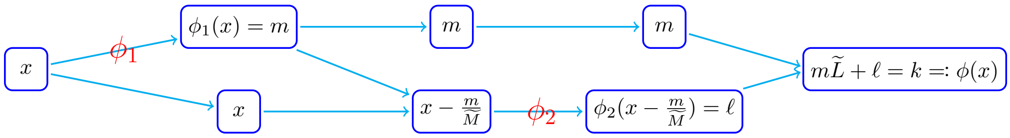


Figure 9: An illustration of the network architecture implementing ϕ based on Equation (4.2) and (4.3) for $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot \mathbb{1}_{\{k \le K-2\}}] = [\frac{mL+\ell}{\widetilde{M}\widetilde{L}}, \frac{mL+\ell+1}{\widetilde{M}\widetilde{L}} - \delta \cdot \mathbb{1}_{\{m \le \widetilde{M}-2 \text{ or } \ell \le \widetilde{L}-2\}}]$, where $k = m\widetilde{L} + \ell$ for $m = 0, 1, \cdots, \widetilde{M} - 1$ and $\ell = 0, 1, \cdots, \widetilde{L} - 1$.

By Lemma 4.2, we have

$$\begin{aligned}\phi_1 &\in \mathcal{NN}\big(\#\text{input} = 1;\ \text{widthvec} = \big[2\lfloor N^{1/d}\rfloor, 4\lfloor N^{1/d}\rfloor\lfloor L^{1/d}\rfloor - 1\big];\ \#\text{output} = 1\big)\\ &\subseteq \mathcal{NN}\big(\#\text{input} = 1;\ \text{width} \le 8\lfloor N^{1/d}\rfloor + 2;\ \text{depth} \le \lfloor L^{1/d}\rfloor + 1;\ \#\text{output} = 1\big)\end{aligned}$$

and

$$\begin{aligned}\phi_2 &\in \mathcal{NN}\big(\#\text{input} = 1;\ \text{widthvec} = \big[2\lfloor n^{1/d}\rfloor, 4\lfloor L^{1/d}\rfloor - 1\big];\ \#\text{output} = 1\big)\\ &\subseteq \mathcal{NN}\big(\#\text{input} = 1;\ \text{width} \le 8\lfloor n^{1/d}\rfloor + 2;\ \text{depth} \le \lfloor L^{1/d}\rfloor + 1;\ \#\text{output} = 1\big).\end{aligned}$$

Recall that $n = \lfloor \log_3(N+2)\rfloor \le N$. It follows from Figure 9 that ϕ can be implemented by a ReLU network with width

$$\max\big\{8\lfloor N^{1/d}\rfloor + 2 + 1, 8\lfloor n^{1/d}\rfloor + 2 + 1\big\} = 8\lfloor N^{1/d}\rfloor + 3$$

and depth

$$(\lfloor L^{1/d}\rfloor + 1) + 2 + (\lfloor L^{1/d}\rfloor + 1) + 1 = 2\lfloor L^{1/d}\rfloor + 5.$$

So we finish the proof.

4.3 Proof of Proposition 3.2

The proof of Proposition 3.2 is based on the bit extraction technique in [3, 13]. 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 4.4 and 4.5 below.

Lemma 4.4. *For any $n \in \mathbb{N}^+$, there exists a function ϕ in*

$$\mathcal{NN}\big(\#\text{input} = 2;\ \text{width} \le (n+1)2^{n+1};\ \text{depth} \le 3;\ \#\text{output} = 1\big)$$

such that: Given any $\theta_j \in \{0,1\}$ for $j = 1, 2, \cdots, n$, we have

$$\phi(\text{bin}\, 0.\theta_1\theta_2\cdots\theta_n,\ i) = \sum_{j=1}^{i} \theta_j, \quad \text{for any } i \in \{0, 1, 2, \cdots, n\}.^{②}$$

Proof. Define $\theta = \text{bin}\, 0.\theta_1\theta_2\cdots\theta_n$. Clearly,

$$\theta_j = \lfloor 2^j\theta\rfloor/2 - \lfloor 2^{j-1}\theta\rfloor, \quad \text{for any } j \in \{1, 2, \cdots, n\}.$$

We shall use a ReLU network to replace $\lfloor\cdot\rfloor$. Let $g \in \text{CPwL}(\mathbb{R}, 2^{n+1} - 2)$ be the function satisfying two conditions:

②By convention, $\sum_{j=n}^{m} a_j = 0$ if $n > m$, no matter what a_j is for each j.

- g matches set of samples

$$\bigcup_{k=0}^{2^n-1} \big\{(k,k),(k+1-\delta,k)\big\}, \quad \text{where } \delta = 2^{-(n+1)};$$

- The breakpoint set of g is

$$\Big(\bigcup_{k=0}^{2^n-1} \{k, k+1-\delta\}\Big) \Big\backslash \Big(\{0\} \cup \{2^n-\delta\}\Big).$$

Then $g(x) = \lfloor x \rfloor$ for any $x \in \bigcup_{k=0}^{2^n-1}[k,k+1-\delta]$. Clearly, $\theta = \text{bin}\, 0.\theta_1\theta_2\cdots\theta_n$ implies

$$2^j\theta \in \bigcup_{k=0}^{2^n-1}[k,k+1-\delta], \quad \text{for any } j \in \{1,2,\cdots,n\}.$$

Thus,

$$\theta_j = \lfloor 2^j\theta \rfloor/2 - \lfloor 2^{j-1}\theta \rfloor = g(2^j\theta)/2 - g(2^{j-1}\theta), \quad \text{for any } j \in \{1,2,\cdots,n\}. \tag{4.4}$$

It is easy to design a ReLU network to output $\theta_1,\theta_2,\cdots,\theta_n$ by Equation (4.4) when using $\theta = \text{bin}\,0.\theta_1\theta_2\cdots\theta_n$ as the input. However, it is highly non-trivial to construct a ReLU network to output $\sum_{j=1}^{i}\theta_j$ with another input i, since many operations like multiplication and comparison are not allowed in designing ReLU networks. Now let us establish a formula to represent $\sum_{j=1}^{i}\theta_j$ in a form of a ReLU network as follows.

Define $\mathcal{T}(n) \coloneqq \sigma(n+1) - \sigma(n) = \left\{\begin{smallmatrix} 1, & n\ge 0, \\ 0, & n<0 \end{smallmatrix}\right.$ for any integer n. Then, by Equation (4.4) and the fact $x_1x_2 = \sigma(x_1+x_2-1)$ for any $x_1,x_2 \in \{0,1\}$, we have, for $i = 0,1,2,\cdots,n$,

$$\begin{aligned} \sum_{j=1}^{i}\theta_j = \sum_{j=1}^{n}\theta_j \cdot \mathcal{T}(i-j) &= \sum_{j=1}^{n}\theta_j \cdot \Big(\sigma(i-j+1) - \sigma(i-j)\Big) \\ &= \sum_{j=1}^{n}\sigma\Big(\theta_j + \sigma(i-j+1) - \sigma(i-j) - 1\Big) \\ &= \sum_{j=1}^{n}\sigma\Big(g(2^j\theta)/2 - g(2^{j-1}\theta) + \sigma(i-j+1) - \sigma(i-j) - 1\Big). \end{aligned}$$

Define

$$z_{i,j} \coloneqq \sigma\Big(g(2^j\theta)/2 - g(2^{j-1}\theta) + \sigma(i-j+1) - \sigma(i-j) - 1\Big), \tag{4.5}$$

for any $i,j \in \{1,2,\cdots,n\}$. Then the goal is to design ϕ satisfying

$$\phi(\theta,i) = \sum_{j=1}^{i}\theta_j = \sum_{j=1}^{n} z_{i,j}, \quad \text{for any } i \in \{0,1,2,\cdots,n\}. \tag{4.6}$$

See Figure 10 for the network architecture implementing the desired function ϕ.

By Lemma 4.3, we have

$$g \in \text{CPwL}(\mathbb{R}, 2^{n+1}-2) \subseteq \mathcal{NN}\Big(\#\text{input} = 1;\ \text{widthvec} = [2^{n+1}-1];\ \#\text{output} = 1\Big),$$

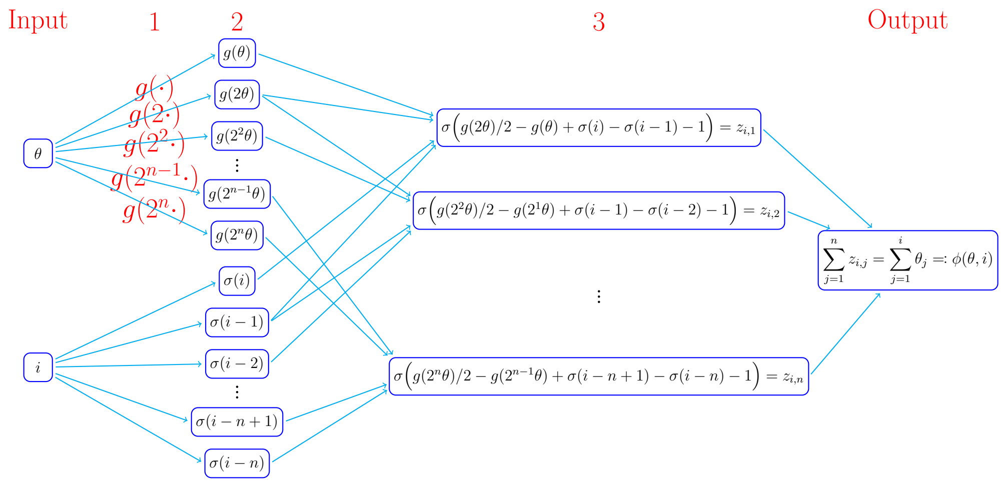


Figure 10: An illustration of the network implementing the desired function ϕ with the input $[\theta, i]^T = [\mathrm{bin}\, 0.\theta_1\theta_2\cdots\theta_n,\ i]^T$ for any $i \in \{0, 1, 2, \cdots, n\}$ and $\theta_1, \theta_2, \cdots, \theta_n \in \{0, 1\}$. $g(2^j\cdot)$ can be implemented by a one-hidden-layer network with width $2^{n+1} - 1$ for each $j \in \{0, 1, \cdots, n\}$. The red numbers above the architecture indicate the order of hidden layers. The network architecture is essentially determined by Equation (4.5) and (4.6), which are valid no matter what $\theta_1, \theta_2, \cdots, \theta_n \in \{0, 1\}$ are. Thus, the desired function ϕ is independent of $\theta_1, \theta_2, \cdots, \theta_n \in \{0, 1\}$. We omit ReLU ($\sigma$) for a neuron if its output is non-negative without ReLU. Such a simplification are applied to similar figures in this paper.

implying

$$g(2^j\cdot) \in \mathrm{CPwL}(\mathbb{R}, 2^{n+1} - 2) \subseteq \mathcal{NN}\big(\#\mathrm{input} = 1;\ \mathrm{widthvec} = [2^{n+1} - 1];\ \#\mathrm{output} = 1\big),$$

for any $j = 0, 1, 2, \cdots, n$. Clearly, the network in Figure 10 has width $(n+1)(2^{n+1} - 1) + (n+1) = (n+1)2^{n+1}$ and depth 3. So we finish the proof. $\square$

Lemma 4.5. *For any $n, L \in \mathbb{N}^+$, there exists a function ϕ in*

$$\mathcal{NN}\big(\#\mathrm{input} = 2;\ \mathrm{width} \le (n+3)2^{n+1} + 4;\ \mathrm{depth} \le 4L + 2;\ \#\mathrm{output} = 1\big)$$

such that: Given any $\theta_j \in \{0, 1\}$ for $j = 1, 2, \cdots, Ln$, we have

$$\phi(\mathrm{bin}\, 0.\theta_1\theta_2\cdots\theta_{Ln},\ k) = \sum_{j=1}^{k} \theta_j, \quad \text{for any } k \in \{1, 2, \cdots, Ln\}.$$

Proof. Let $g_1 \in \mathrm{CPwL}(\mathbb{R}, 2^{n+1} - 2)$ be the function satisfying:

- g_1 matches the set of samples

$$\bigcup_{i=0}^{2^n - 1} \big\{(i, i), (i + 1 - \delta, i)\big\}, \quad \text{where } \delta = 2^{-(Ln+1)}.$$

- The breakpoint set of g_1 is

$$\Bigg(\bigcup_{i=0}^{2^n - 1} \big\{(i, i), (i + 1 - \delta, i)\big\}\Bigg) \Big\backslash \Big(\{0\} \cup \{2^n - \delta\}\Big).$$

Then $g_1(x) = \lfloor x \rfloor$ for any $x \in \bigcup_{i=0}^{2^n-1}[i, i+1-\delta]$. Note that

$$2^n \cdot \mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{Ln} \in \bigcup_{i=0}^{2^n-1}[i, i+1-\delta], \qquad \text{for any } \ell \in \{0,1,\cdots,L-1\}.$$

Thus, for any $\ell \in \{0,1,\cdots,L-1\}$, we have

$$\mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{\ell n+n} = \frac{\lfloor 2^n \cdot \mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{Ln} \rfloor}{2^n} = \frac{g_1(2^n \cdot \mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{Ln})}{2^n}. \tag{4.7}$$

Define $g_2(x) \coloneqq 2^n x - g_1(2^n x)$ for any $x \in \mathbb{R}$. Then $g_2 \in \mathrm{CPwL}(\mathbb{R}, 2^{n+1}-2)$ and

$$\begin{aligned} \mathrm{bin}\,0.\theta_{(\ell+1)n+1}\cdots\theta_{Ln} &= 2^n\Big(\mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{Ln} - \mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{\ell n+n}\Big) \\ = 2^n\Big(\mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{Ln} &- \frac{g_1(2^n \cdot \mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{Ln})}{2^n}\Big) = g_2(\mathrm{bin}\,0.\theta_{\ell n+1}\cdots\theta_{Ln}). \end{aligned} \tag{4.8}$$

By Lemma 4.4, there exists

$$\phi_1 \in \mathcal{NN}\big(\#\text{input} = 2;\ \text{width} \le (n+1)2^{n+1};\ \text{depth} \le 3;\ \#\text{output} = 1\big)$$

such that: For any $\xi_1, \xi_2, \cdots, \xi_n \in \{0,1\}$, we have

$$\phi_1(\mathrm{bin}\,0.\xi_1\xi_2\cdots\xi_n,\ i) = \sum_{j=1}^{i} \xi_j, \qquad \text{for } i = 0,1,2,\cdots,n.$$

It follows that

$$\phi_1(\mathrm{bin}\,0.\theta_{\ell n+1}\theta_{\ell n+2}\cdots\theta_{\ell n+n},\ i) = \sum_{j=1}^{i} \theta_{\ell n+j}, \qquad \text{for } \ell = 0,1,\cdots,L-1 \text{ and } i = 0,1,\cdots,n. \tag{4.9}$$

Define $\phi_{2,\ell}(x) \coloneqq \min\{\sigma(x-\ell n), n\}$ for any $x \in \mathbb{R}$ and $\ell \in \{0,1,\cdots,L-1\}$. For any $k \in \{1,2,\cdots,Ln\}$, there exists $k_1 \in \{0,1,\cdots,L-1\}$ and $k_2 \in \{1,2,\cdots,n\}$ such that $k = k_1 n + k_2$, implying

$$\begin{aligned} \sum_{i=1}^{k} \theta_i = \sum_{i=1}^{k_1 n+k_2} \theta_i &= \sum_{\ell=0}^{k_1-1}\Bigg(\sum_{j=1}^{n}\theta_{\ell n+j}\Bigg) + \sum_{\ell=k_1}^{k_1}\Bigg(\sum_{j=1}^{k_2}\theta_{\ell n+j}\Bigg) + \sum_{\ell=k_1+1}^{L-1}\Bigg(\sum_{j=1}^{0}\theta_{\ell n+j}\Bigg) \\ &= \sum_{\ell=0}^{L-1}\Bigg(\sum_{j=1}^{\min\{\sigma(k-\ell n), n\}}\theta_{\ell n+j}\Bigg) = \sum_{\ell=0}^{L-1}\Bigg(\sum_{j=1}^{\phi_{2,\ell}(k)}\theta_{\ell n+j}\Bigg). \end{aligned} \tag{4.10}$$

Then, the desired function ϕ can be implemented by the network architecture in Figure 11.

By Lemma 4.3, we have

$$g_1, g_2 \in \mathrm{CPwL}(\mathbb{R}, 2^{n+1}-2) \subseteq \mathcal{NN}\big(\#\text{input} = 1;\ \text{widthvec} = [2^{n+1}-1];\ \#\text{output} = 1\big).$$

Recall that $\phi_1 \in \mathcal{NN}\big(\text{width} \le (n+1)2^{n+1};\ \text{depth} \le 3\big)$. As shown in Figure 12, $\phi_{2,\ell}(x) \in \mathcal{NN}(\text{width} \le 4;\ \text{depth} \le 2)$ for $\ell = 0,1,\cdots,L-1$. Therefore, the network in Figure 11 has width

$$(2^{n+1}-1)\ +\ (2^{n+1}-1)\ +\ (n+1)2^{n+1}\ +\ 1\ +\ 4\ +\ 1 = (n+3)2^{n+1}+4$$

and depth

$$2 + L(1+3) = 4L+2.$$

So we finish the proof. $\square$

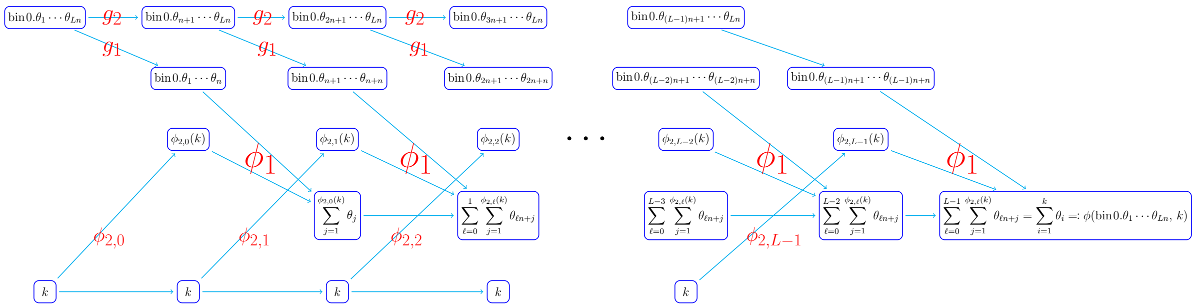


Figure 11: An illustration of the network implementing the desired function ϕ with the input $[\text{bin}0.\theta_1\theta_2\cdots\theta_{Ln},k]^T$ for any $k\in\{1,2,\cdots,Ln\}$ and $\theta_1,\theta_2,\cdots,\theta_{Ln}\in\{0,1\}$. The network architecture is essentially determined by Equation (4.7), (4.8), (4.9), and (4.10), which are valid no matter what $\theta_1,\theta_2,\cdots,\theta_{Ln}\in\{0,1\}$ are. Thus, the desired function ϕ is independent of $\theta_1,\theta_2,\cdots,\theta_{Ln}\in\{0,1\}$. We omit ReLU ($\sigma$) for a neuron if its output is non-negative without ReLU.

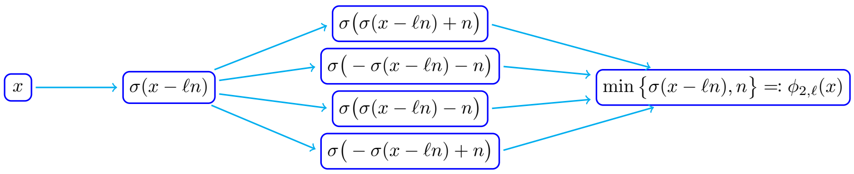


Figure 12: An illustration of the network implementing the desired function $\phi_{2,\ell}$ for each $\ell\in\{0,1,\cdots,L-1\}$, based on $\min\{x,n\}=\frac{1}{2}\big(\sigma(x-n)-\sigma(-x-n)-\sigma(x-n)-\sigma(-x+n)\big)$.

Next, we introduce Lemma 4.6 to map indices to the partial sum of given bits.

Lemma 4.6. *Given any $N,L\in\mathbb{N}^+$ and arbitrary $\theta_{m,k}\in\{0,1\}$ for $m=0,1,\cdots,M-1$ and $k=0,1,\cdots,Ln-1$, where $M=N^2L$ and $n=\lfloor\log_3(N+2)\rfloor$, there exists*

$$\phi\in\mathcal{NN}\big(\#\text{input}=2;\ \text{width}\le 6N+14;\ \text{depth}\le 5L+4;\ \#\text{output}=1\big)$$

such that

$$\phi(m,k)=\sum_{j=0}^{k}\theta_{m,j},\qquad \text{for } m=0,1,\cdots,M-1 \text{ and } k=0,1,\cdots,Ln-1.$$

Proof. Define

$$y_m:=\text{bin}\,0.\theta_{m,0}\theta_{m,1}\cdots\theta_{m,Ln-1},\qquad \text{for } m=0,1,\cdots,M-1.$$

Consider the sample set $\{(m,y_m):m=0,1,\cdots,M\}$, whose cardinality is

$$M+1=N\big((NL-1)+1\big)+1.$$

By Lemma 4.1 (set $N_1=N$ and $N_2=NL-1$ therein), there exists

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

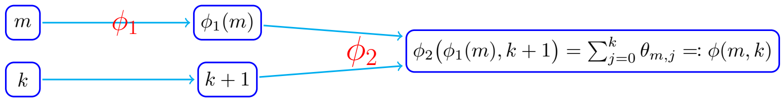


Figure 13: An illustration of the network implementing the desired function ϕ for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$.

such that

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

By Lemma 4.4, there exists

$$\phi_2 \in \mathcal{NN}\big(\#\text{input} = 2;\ \text{width} \le (n+3)2^{n+1} + 4;\ \text{depth} \le 4L + 2;\ \#\text{output} = 1\big)$$

such that, for any $\xi_1, \xi_2, \cdots, \xi_{Ln} \in \{0, 1\}$, we have

$$\phi_2(\text{bin}\, 0.\xi_1\xi_2\cdots\xi_{Ln},\ k) = \sum_{j=1}^{k} \xi_j, \quad \text{for } k = 1, 2, \cdots, Ln.$$

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

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

Thus, for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$, we have

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

Hence, the desired function function ϕ can be implemented by the network shown in Figure 13. By Lemma 4.2, $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2NL-1]) \subseteq \mathcal{NN}(\text{width} \le 4N+2;\ \text{depth} \le L+1)$. It holds that

$$(n+3)2^{n+1} + 4 \le 6 \cdot (3^n) + 2 = 6 \cdot (3^{\lfloor \log_3(N+2) \rfloor}) + 2 \le 6(N+2) + 2 = 6N + 14,$$

implying

$$\begin{aligned} \phi_2 &\in \mathcal{NN}\big(\#\text{input} = 2;\ \text{width} \le (n+3)2^{n+1} + 4;\ \text{depth} \le 4L+2;\ \#\text{output} = 1\big) \\ &\subseteq \mathcal{NN}\big(\#\text{input} = 2;\ \text{width} \le 6N + 14;\ \text{depth} \le 4L+2;\ \#\text{output} = 1\big). \end{aligned}$$

Therefore, the network in Figure 13 is with width $\max\{(4N+2)+1, 6N+14\} = 6N+14$ and depth $(4L+2) + 1 + (L+1) = 5L+4$. So we finish the proof. $\square$

Next, we apply Lemma 4.6 to prove Lemma 4.7 below, which is a key intermediate conclusion to prove Proposition 3.2.

Lemma 4.7. *For any $\varepsilon > 0$ and $N, L \in \mathbb{N}^+$, denote $M = N^2L$ and $n = \lfloor \log_3(N+2) \rfloor$. Assume $y_{m,k} \geq 0$ for $m = 0, 1, \cdots, M-1$* and *$k = 0, 1, \cdots, Ln-1$ are samples with*

$$|y_{m,k} - y_{m,k-1}| \leq \varepsilon, \quad \text{for } m = 0, 1, \cdots, M-1 \quad \text{and} \quad k = 1, 2, \cdots, Ln-1.$$

Then there exists $\phi \in \mathcal{NN}(\#\text{input} = 2;\ \text{width} \leq 16N + 30;\ \text{depth} \leq 5L + 7;\ \#\text{output} = 1)$ such that

(i) $|\phi(m,k) - y_{m,k}| \leq \varepsilon$ for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$;

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

Proof. Define

$$a_{m,k} \coloneqq \lfloor y_{m,k}/\varepsilon \rfloor, \quad \text{for } m = 0, 1, \cdots, M-1 \quad \text{and} \quad k = 0, 1, \cdots, Ln-1.$$

We will construct a function implemented by a ReLU network to map the index (m,k) to $a_{m,k}\varepsilon$ for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$.

Define $b_{m,0} \coloneqq 0$ and $b_{m,k} \coloneqq a_{m,k} - a_{m,k-1}$ for $m = 0, 1, \cdots, M-1$ and $k = 1, 2, \cdots, Ln-1$. Since $|y_{m,k} - y_{m,k-1}| \leq \varepsilon$ for all m and k, we have $b_{m,k} \in \{-1, 0, 1\}$. Hence, there exist $c_{m,k} \in \{0, 1\}$ and $d_{m,k} \in \{0, 1\}$ such that $b_{m,k} = c_{m,k} - d_{m,k}$, which implies

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

for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$.

Consider the sample set

$$\big\{(m, a_{m,0}) : m = 0, 1, \cdots, M-1\big\} \bigcup \{(M, 0)\}.$$

Its size is $M + 1 = N \cdot \big((NL-1)+1\big) + 1$, by Lemma 4.1 (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

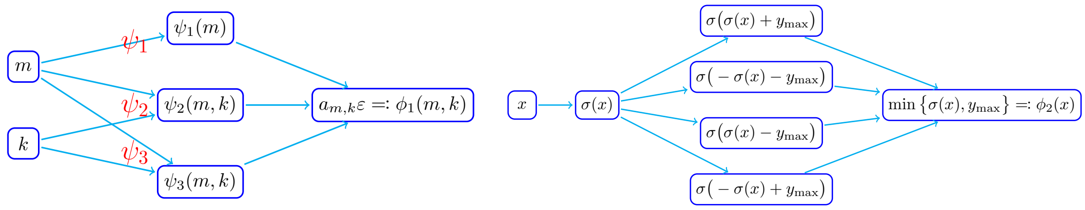
$$\psi_1(m) = a_{m,0}, \quad \text{for } m = 0, 1, \cdots, M-1.$$

By Lemma 4.6, there exist $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \leq 6N + 14;\ \text{depth} \leq 5L + 4)$ such that

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

for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$. Hence, it holds that

$$a_{m,k} = a_{m,0} + \sum_{i=0}^{k} c_{m,i} - \sum_{i=0}^{k} d_{m,i} = \psi_1(m) + \psi_2(m,k) - \psi_3(m,k), \tag{4.11}$$



(a) ϕ_1 (b) ϕ_2

Figure 14: Illustrations of two sub-networks implementing the desired function $\phi = \phi_2 \circ \phi_1$ for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$, based on Equation (4.11) and the fact $\min\{x_1, x_2\} = \frac{x_1+x_2-|x_1-x_2|}{2} = \frac{\sigma(x_1+x_2)-\sigma(-x_1-x_2)-\sigma(x_1-x_2)-\sigma(-x_1+x_2)}{2}$.

for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$.

Define

$$y_{\max} := \max\{y_{m,k} : m = 0, 1, \cdots, M-1 \quad \text{and} \quad k = 0, 1, \cdots, Ln-1\}.$$

Then the desired function can be implemented by two sub-networks shown in Figure 14.

By Lemma 4.2,

$$\begin{aligned} \psi_1 &\in \mathcal{NN}(\#\text{input} = 1;\ \text{widthvec} = [2N, 2NL-1];\ \#\text{output} = 1) \\ &\subseteq \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 4N+2;\ \text{depth} \le L+1;\ \#\text{output} = 1). \end{aligned}$$

Recall that $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \le 6N+14;\ \text{depth} \le 5L+4)$. Thus, $\phi_1 \in \mathcal{NN}(\text{width} \le (4N+2) + 2(6N+14) = 16N+30;\ \text{depth} \le (5L+4)+1 = 5L+5)$ as shown in Figure 14. And it is clear that $\phi_2 \in \mathcal{NN}(\text{width} \le 4;\ \text{depth} \le 2)$, implying $\phi = \phi_2 \circ \phi_1 \in \mathcal{NN}(\text{width} \le 16N+30;\ \text{depth} \le (5L+5)+2 = 5L+7)$.

Clearly, $0 \le \phi(x_1, x_2) \le y_{\max}$ 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 $0 \le a_{m,k}\varepsilon = \lfloor y_{m,k}/\varepsilon \rfloor \varepsilon \le y_{\max}$. Then we have $\phi(m,k) = \phi_2 \circ \phi_1(m,k) = \phi_2(a_{m,k}\varepsilon) = \max\{\sigma(a_{m,k}\varepsilon), y_{\max}\} = a_{m,k}\varepsilon$. Therefore,

$$|\phi(m,k) - y_{m,k}| = |a_{m,k}\varepsilon - y_{m,k}| = \big|\lfloor y_{m,k}/\varepsilon \rfloor \varepsilon - y_{m,k}\big| \le \varepsilon,$$

for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, Ln-1$. Hence, we finish the proof. □

Finally, we apply Lemma 4.7 to prove Proposition 3.2.

Proof of Proposition 3.2. Denote $M = N^2L$, $n = \lfloor \log_3(N+2) \rfloor$, and $\widehat{L} = Ln$. We may assume $J = MLn = M\widehat{L}$ since we can set $y_{J-1} = y_J = y_{J+1} = \cdots = y_{M\widehat{L}-1}$ if $J < M\widehat{L}$.

Consider the sample set

$$\{(m\widehat{L}, m) : m = 0, 1, \cdots, M\} \bigcup \{(m\widehat{L} + \widehat{L} - 1, m) : m = 0, 1, \cdots, M-1\}.$$

Its size is $2M + 1 = N \cdot \big((2NL-1)+1\big) + 1$. By Lemma 4.1 (set $N_1 = N$ and $N_2 = NL-1$ therein), there exist

$$\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2(2NL-1)+1]) = \mathcal{NN}(\text{widthvec} = [2N, 4NL-1])$$

such that

- $\phi_1(M\widehat{L}) = M$ and $\phi_1(m\widehat{L}) = \phi_1(m\widehat{L} + \widehat{L} - 1) = m$ for $m = 0, 1, \cdots, M-1$.
- ϕ_1 is linear on each interval $[m\widehat{L}, m\widehat{L} + \widehat{L} - 1]$ for $m = 0, 1, \cdots, M-1$.

It follows that

$$\phi_1(j) = m, \quad \text{and} \quad j - \widehat{L}\phi_1(j) = k, \quad \text{where } j = m\widehat{L} + k, \tag{4.12}$$

for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, \widehat{L} - 1$.

Note that any number j in $\{0, 1, \cdots, J-1\}$ can be uniquely indexed as $j = m\widehat{L} + k$ for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, \widehat{L} - 1$. So we can denote $y_j = y_{m\widehat{L}+k}$ as $y_{m,k}$. Then by Lemma 4.7, there exists $\phi_2 \in \mathcal{NN}(\text{width} \le 16N + 30;\ \text{depth} \le 5L + 7)$ such that

$$|\phi_2(m,k) - y_{m,k}| \le \varepsilon, \quad \text{for } m = 0, 1, \cdots, M-1 \quad \text{and} \quad k = 0, 1, \cdots, \widehat{L} - 1, \tag{4.13}$$

and

$$0 \le \phi_2(x_1, x_2) \le y_{\max}, \quad \text{for any } x_1, x_2 \in \mathbb{R}, \tag{4.14}$$

where $y_{\max} := \max\{y_{m,k} : m = 0, 1, \cdots, M-1 \text{ and } k = 0, 1, \cdots, \widehat{L} - 1\} = \max\{y_j : j = 0, 1, \cdots, M\widehat{L} - 1\}$.

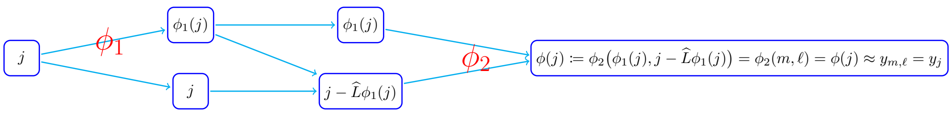


Figure 15: An illustration of the ReLU network implementing the desired function ϕ based Equation (4.12). The index $j \in \{0, 1, \cdots, M\widehat{L} - 1\}$ is unique represented by $j = mL + k$ for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, \widehat{L} - 1$.

By Lemma 4.2,

$$\begin{aligned} \phi_1 &\in \mathcal{NN}(\#\text{input} = 1;\ \text{widthvec} = [2N, 4NL - 1];\ \#\text{output} = 1) \\ &\subseteq \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 8N + 2; \text{depth} \le L + 1;\ \#\text{output} = 1). \end{aligned}$$

Recall that $\phi_2 \in \mathcal{NN}(\text{width} \le 16N + 30;\ \text{depth} \le 5L + 7)$. So $\phi \in \mathcal{NN}(\text{width} \le 16N + 30;\ \text{depth} \le (L+1) + 2 + (5L+7) = 6L + 10)$ as shown in Figure 15.

Equation (4.14) implies

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

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

Represent $j \in \{0, 1, \cdots, M\widehat{L} - 1\}$ via $j = mL + k$ for $m = 0, 1, \cdots, M-1$ and $k = 0, 1, \cdots, \widehat{L} - 1$. Then, by Equation (4.13), we have

$$|\phi(j) - y_j| = |\phi_2\big(\phi_1(j), j - L\phi_1(j)\big) - y_j| = |\phi_2(m,k) - y_{m,k}| \le \varepsilon,$$

for any $j \in \{0, 1, \cdots, M\widehat{L} - 1\} = \{0, 1, \cdots, J-1\}$. So we finish the proof. □

We would like to remark that the key idea in the proof of Proposition 3.2 is the bit extraction technique in Lemma 4.5, which allows us to store Ln bits in a binary number $\text{bin}\, 0.\theta_1\theta_2\cdots\theta_{Ln}$ and extract each bit θ_i. The extraction operator can be efficiently carried out via a deep ReLU neural network demonstrating the power of depth.

5 Conclusion and future work

This paper aims at a quantitative and optimal approximation rate for ReLU networks in terms of the width and depth to approximate continuous functions. It is shown by construction that ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ can approximate an arbitrary continuous function on $[0,1]^d$ with an approximation rate $\mathcal{O}\big(\omega_f\big((N^2L^2\ln N)^{-1/d}\big)\big)$. By connecting the approximation property to VC-dimension, we prove that such a rate is optimal for Hölder continuous functions on $[0,1]^d$ in terms of the width and depth separately, and hence this rate is also optimal for the whole continuous function class. We also extend our analysis to general continuous functions on any bounded set in $\mathbb{R}^d$. 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 extend our conclusions to neural networks with other types of architectures (e.g., convolutional neural networks) and activation functions (e.g., tanh and sigmoid functions).

Acknowledgments

Z. Shen is supported by Tan Chin Tuan Centennial Professorship. H. Yang was partially supported by the US National Science Foundation under award DMS-1945029. S. Zhang is supported by a Postdoctoral Fellowship under NUS ENDOWMENT FUND (EXP WBS) (01 651).

References

[1] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, New York, NY, USA, 1st ed., 2009.

[2] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Transactions on Information Theory, 39 (1993), pp. 930–945.

[3] P. Bartlett, V. Maiorov, and R. Meir, *Almost linear VC dimension bounds for piecewise polynomial networks*, Neural Computation, 10 (1998), pp. 2159–2173.

[4] 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.

[5] L. Chen and C. Wu, *A note on the expressive power of deep rectified linear unit networks in high-dimensional spaces*, Mathematical Methods in the Applied Sciences, 42 (2019), pp. 3400–3404.

[6] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, MCSS, 2 (1989), pp. 303–314.

[7] W. E, C. Ma, and Q. Wang, *A priori estimates of the population risk for residual networks*, ArXiv, abs/1903.02154 (2019).

[8] 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.

[9] W. E and Q. Wang, *Exponential convergence of the deep neural network approximation for analytic functions*, CoRR, abs/1807.00297 (2018).

[10] W. E and S. Wojtowytsch, *A priori estimates for classification problems using neural networks*, arXiv e-prints, (2020), p. arXiv:2009.13500.

[11] W. E and S. Wojtowytsch, *On the banach spaces associated with multi-layer relu networks: Function representation, approximation theory and gradient descent dynamics*, arXiv:2007.15623, (2020).

[12] ——, *Representation formulas and pointwise properties for barron functions*, 2020.

[13] 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.

[14] J. He, X. Jia, J. Xu, L. Zhang, and L. Zhao, *Make ℓ_1 regularization effective in training sparse cnn*, Computational Optimization and Applications, 77 (2020), p. 163–182.

[15] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Networks, 4 (1991), pp. 251 – 257.

[16] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural Networks, 2 (1989), pp. 359 – 366.

[17] 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.

[18] K. Kawaguchi and Y. Bengio, *Depth with nonlinearity creates no bad local minima in resnets*, Neural Networks, 118 (2019), pp. 167–174.

[19] M. J. Kearns and R. E. Schapire, *Efficient distribution-free learning of probabilistic concepts*, J. Comput. Syst. Sci., 48 (1994), pp. 464–497.

[20] Q. Li, T. Lin, and Z. Shen, *Deep learning via dynamical systems: An approximation perspective*, arXiv e-prints, (2019).

[21] Q. Li, C. Tai, and W. E, *Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations*, Journal of Machine Learning Research, 20 (2019), pp. 1–47.

[22] S. Liang and R. Srikant, *Why deep neural networks?*, CoRR, abs/1610.04161 (2016).

[23] H. Lin and S. Jegelka, *Resnet with one-neuron hidden layers is a universal approximator*, in Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., vol. 31, Curran Associates, Inc., 2018.

[24] J. Lu, Z. Shen, H. Yang, and S. Zhang, *Deep network approximation for smooth functions*, arXiv e-prints, (2020), p. arXiv:2001.03040.

[25] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, *The expressive power of neural networks: A view from the width*, in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Curran Associates, Inc., 2017, pp. 6231–6239.

[26] H. Montanelli, H. Yang, and Q. Du, *Deep ReLU networks overcome the curse of dimensionality for bandlimited functions*, Journal of Computational Mathematics, (to appear).

[27] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, *On the number of linear regions of deep neural networks*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Inc., 2014, pp. 2924–2932.

[28] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, *The role of over-parametrization in generalization of neural networks*, in International Conference on Learning Representations, 2019.

[29] Q. N. Nguyen and M. Hein, *The loss surface of deep and wide neural networks*, CoRR, abs/1704.08045 (2017).

[30] P. Petersen and F. Voigtlaender, *Optimal approximation of piecewise smooth functions using deep ReLU neural networks*, Neural Networks, 108 (2018), pp. 296 – 330.

[31] A. Sakurai, *Tight bounds for the VC-dimension of piecewise polynomial networks*, in Advances in Neural Information Processing Systems, Neural information processing systems foundation, 1999, pp. 323–329.

[32] Z. Shen, H. Yang, and S. Zhang, *Nonlinear approximation via compositions*, Neural Networks, 119 (2019), pp. 74 – 84.

[33] ——, *Deep network approximation characterized by number of neurons*, Communications in Computational Physics, 28 (2020), pp. 1768–1811.

[34] Z. Shen, H. Yang, and S. Zhang, *Deep network with approximation error being reciprocal of width to power of square root of depth*, arXiv e-prints, (2020).

[35] ——, *Neural network approximation: Three hidden layers are enough*, arXiv e-prints, (2020).

[36] J. W. Siegel and J. Xu, *Approximation rates for neural networks with general activation functions*, Neural Networks, 128 (2020), pp. 313–321.

[37] ——, *Optimal approximation rates and metric entropy of relu^k and cosine networks*, 2021.

[38] P. Urysohn, *über die mächtigkeit der zusammenhängenden mengen*, Mathematische Annalen, 94 (1925), p. 262–295.

[39] H. Whitney, *Analytic extensions of differentiable functions defined in closed sets*, Transactions of the American Mathematical Society, 36 (1934), pp. 63–89.

[40] D. Yarotsky, *Error bounds for approximations with deep ReLU networks*, Neural Networks, 94 (2017), pp. 103 – 114.

[41] D. Yarotsky, *Optimal approximation of continuous functions by very deep ReLU networks*, in Proceedings of the 31st Conference On Learning Theory, S. Bubeck, V. Perchet, and P. Rigollet, eds., vol. 75 of Proceedings of Machine Learning Research, PMLR, 06–09 Jul 2018, pp. 639–649.

[42] D. Yarotsky and A. Zhevnerchuk, *The phase diagram of approximation rates for deep neural networks*, in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., vol. 33, Curran Associates, Inc., 2020, pp. 13005–13015.

[43] S. Zhang, *Deep neural network approximation via function compositions*, PhD Thesis, National University of Singapore, (2020). URL: https://scholarbank.nus.edu.sg/handle/10635/186064.