DEEP NEURAL NETWORK APPROXIMATION VIA FUNCTION COMPOSITIONS

ZHANG SHIJUN

(B.Sc., Wuhan University, China)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MATHEMATICS
NATIONAL UNIVERSITY OF SINGAPORE
2020

Supervisors:
Professor Shen Zuowei, Main Supervisor
Assistant Professor Yang Haizhao, Co-Supervisor

Examiners:
Associate Professor Ji Hui
Assistant Professor Li Qianxiao
Professor Cai Jianfeng, Hong Kong University of Science and Technology

To my family

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Zhang Shijun

Zhang Shijun

December 30, 2020

Acknowledgments

This dissertation would never be completed without the guidance of my supervisors, the help from my friends, and the support from my family.

First, I would like to express my sincere gratitude to my main supervisor professor Shen Zuowei, for his immense knowledge and research experience, and his guidance through each stage of my Ph.D. studies. His research philosophy plays a key role in defining the path of my research. During our discussions, I have gained a lot of knowledge and skills, especially the way of thinking and doing research. Without his constructive comments and suggestions, I would hardly complete my research work and this dissertation. It is my honor and pleasure to work with such an outstanding main supervisor.

Next, I would like to acknowledge my co-supervisor assistant professor Yang Haizhao, for his patience and enthusiasm, and his continuous support for my Ph.D. studies and related research. He convincingly guided and encouraged me to be professional. He has taught me a lot of things, including writing a research paper professionally, expressing my own opinions regarding a research problem precisely and clearly, *etc.* They benefited me a lot during my Ph.D. studies. Without his insightful feedback, I would hardly complete my research work and this dissertation. I am so lucky to do research with such a wonderful co-supervisor.

Besides my supervisors, I am also immensely grateful to associate professor Ji Hui, all members of the NUS Wavelet group, and all my classmates in the same student office as me. The numerous discussions with them in the group seminars or the student office helped me improve my knowledge in the research topics and develop a lot of skills for computer programming.

Finally, I would like to thank my family for their encouragement and understanding. My grandparents and parents raised me and supported me in achieving my pursuits. They kept me going on and this dissertation would not have been possibly finished without their input.

Contents

Summary

This dissertation is a summary of our previous papers [38, 52, 53, 54], focusing on the approximation theory of neural networks. We provide (nearly optimal) approximation error estimates in terms of the width and depth when constructing ReLU ($\max\{0, x\}$) networks, via the idea of function compositions, to uniformly approximate polynomials, (Hölder) continuous functions, and smooth functions on a hypercube $[0, 1]^d$. The optimality is discussed via studying the connection between the approximation error and VC-dimension.[①] To overcome the limitation of ReLU networks that (nearly) exponential approximation errors[②] hold only for polynomials among all function spaces considered, we introduce new networks built with either Floor ($\lfloor x \rfloor$) or ReLU as the activation function in each neuron. We call such networks Floor-ReLU networks. It is proved by construction that nearly exponential approximation errors can be attained when using Floor-ReLU networks to approximate (Hölder) continuous functions on $[0, 1]^d$. See Table 1.1 for a summary.

Chapter 1 is the introduction of this dissertation, including our main contributions and the literature review.

①See the definition of VC-dimension in Section 4.4.

②Throughout this dissertation, "exponential (approximation) error(s)" means "(approximation) error(s) with exponential decay", similar to [21, 46, 54].

Chapter 2 is the preliminary chapter. In this chapter, we present the notations used throughout this dissertation, discuss the architecture of neural networks, and provide the general ideas of constructing neural networks to approximate given functions.

In Chapter 3, we prove several basic results of ReLU networks, which will be used in later chapters. The chapter consists of four parts. The first part investigates representing shallow ReLU networks by deep ones with a similar number of neurons. Part 2 discusses the width and depth power of ReLU networks to fit points. The third part looks at the approximation in a small region if a ReLU network approximates the target function well except for this small region. That is, we modify this network to let it approximate the target function uniformly well on the whole region. The final part deals with the approximation of step functions by ReLU networks.

Chapter 4 focuses on the approximation by ReLU networks and is divided into four parts. The first part aims to construct ReLU networks to approximate general polynomials on $[0,1]^d$ with exponential approximation errors. The second and third parts provide the detailed constructions of ReLU networks for approximating (Hölder) continuous functions and smooth functions on $[0,1]^d$ with (nearly optimal) approximation errors, respectively. The final part looks at the optimality of the approximation by ReLU networks via studying the connection between the approximation error and VC-dimension.

Chapter 5 aims to reveal the approximation power of Floor-ReLU networks. We provide nearly exponential approximation error estimates when constructing Floor-ReLU networks with fixed architectures[③] to uniformly approximate (Hölder) continuous functions on $[0,1]^d$. In other words, the approximation errors are improved from polynomial ones to nearly exponential ones by adding a simple activation function (Floor) to ReLU networks.

Chapter 6 concludes this dissertation with a short discussion.

[③]A Floor-ReLU network with a fixed architecture means all the components of this network architecture is determined except for the values of the parameters. In particular, the choice of activation functions (Floor or ReLU) in each neuron is independent of the target function.

List of Tables

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

Chapter 1

Introduction

Deep neural networks have made significant impacts in many fields of computer science and engineering, especially for large-scale and high-dimensional learning problems. Well-designed neural network architectures, efficient training algorithms, and high-performance computing technologies have made neural-network-based methods very successful in a great number of real applications. Especially in supervised learning, *e.g.*, image classification and objective detection, the great advantages of neural-network-based methods have been demonstrated over traditional learning methods. Understanding the approximation capacity of deep neural networks has become a key question for revealing the power of deep learning. A large number of experiments in real applications have shown the large capacity of deep neural networks from many empirical perspectives, drawing a great deal of attention to the theoretical foundation of the approximation theory of deep neural networks.

In particular, there are three main directions in the error analysis of the approximation theory of neural networks: the **approximation error** estimate, the **optimization error** estimate, and the **generalization error** estimate. See [38,54] for the introduction of these three error estimates. This dissertation concentrates on the approximation error estimate for neural networks. To this end, we need to solve three fundamental problems listed below.

Problem 1: How do we construct a neural network to approximate a function in a given space?

Problem 2: Is there an error estimate for the approximation in Problem 1 in terms of the size of networks, characterized by either the number of parameters or the width and depth simultaneously?

Problem 3: If an error estimate exists in Problem 2, is this error estimate (nearly) optimal for the given function space?

This dissertation solves these three problems for several function spaces. See Table 1.1 for a summary of the main results in this dissertation, focusing on designing neural networks to approximate functions in several given function spaces.

1.1 Contributions

The main contribution of this dissertation is to provide (nearly optimal) approximation error estimates in terms of the width and depth when constructing neural networks to uniformly approximate polynomials, Hölder continuous functions of order $\alpha \in (0,1]$ with a Hölder constant $\lambda > 0$ (Hölder$([0,1]^d, \alpha, \lambda)$), continuous functions ($C([0,1]^d)$), and smooth functions ($C^s([0,1]^d)$) on $[0,1]^d$. See Table 1.1 for a summary. Note that all approximation errors in Table 1.1 hold for **arbitrary** $N, L \in \mathbb{N}^+$ and on $[0,1]^d$ **uniformly**. All constants in $\mathcal{O}(\cdot)$ are **explicitly** estimated in this dissertation, and $\omega_f(\cdot)$ is the modulus of continuity of f defined by $\omega_f(r) = \sup\{|f(\boldsymbol{x}) - f(\boldsymbol{y})| : \|\boldsymbol{x} - \boldsymbol{y}\|_2 \le r,\ \boldsymbol{x}, \boldsymbol{y} \in [0,1]^d\}$.

Table 1.1: A summary of the main results in this dissertation, aiming to design neural networks to approximate functions in several function spaces.

	target function	activation function	width	depth (#hidden-layer)	approximation error	optimality
Lemma 4.2	$f(x) = x^2$	ReLU	$3N$	L	N^{-L}	
Theorem 4.1	polynomial $f(\boldsymbol{x}) = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$	ReLU	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{O}(N^{-L})$	
Corollary 4.7	$f \in$ Hölder$([0,1]^d, \alpha, \lambda)$	ReLU	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{O}(\lambda N^{-2\alpha/d} L^{-2\alpha/d})$	nearly optimal in N and L, see Section 4.4.1
Theorem 4.6	$f \in C([0,1]^d)$	ReLU	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathcal{O}\big(\omega_f(N^{-2/d} L^{-2/d})\big)$	
Theorem 4.11	$f \in C^s([0,1]^d)$, $s \in \mathbb{N}^+$	ReLU	$\mathcal{O}\big(N \ln(N+1)\big)$	$\mathcal{O}\big(L \ln(L+1)\big)$	$\mathcal{O}(\|f\|_{C^s} N^{-2s/d} L^{-2s/d})$	nearly optimal in N and L, see Section 4.4.2
Corollary 5.3	$f \in$ Hölder$([0,1]^d, \alpha, \lambda)$	Floor and ReLU	$\max\{d, 5N+13\}$	$64dL + 3$	$3\lambda d^{\alpha/2} N^{-\alpha\sqrt{L}}$	
Theorem 5.1	$f \in C([0,1]^d)$	Floor and ReLU	$\max\{d, 5N+13\}$	$64dL + 3$	$\omega_f(\sqrt{d}\, N^{-\sqrt{L}}) + 2\omega_f(\sqrt{d}) N^{-\sqrt{L}}$	

We would like to point out that most results in Table 1.1 can be generalized from $[0,1]^d$ to any compact set $E \subseteq \mathbb{R}^d$. Such a generalization is mainly based on two key ideas: 1) an affine linear map $\mathcal{L}_{a,b}(\boldsymbol{x}) = (b-a)\boldsymbol{x} + a$ with proper $a, b \in \mathbb{R}$ satisfying $E \subseteq [a,b]^d$; 2) the function extension (*e.g.*, see Lemma 4.2 of [53] for the extension of continuous functions).

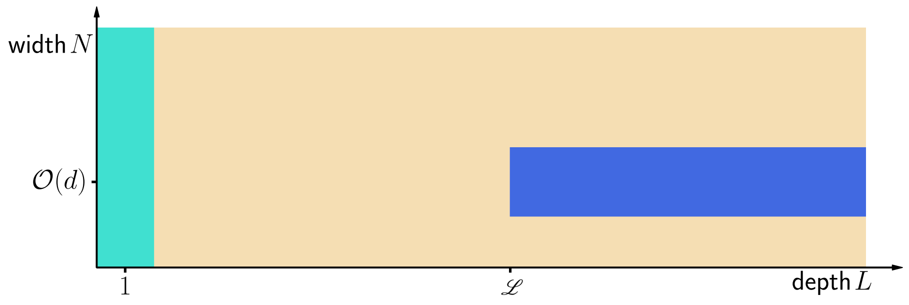


Figure 1.1: A sketch of most existing results and new results in this dissertation. $\mathscr{L}$ represents a sufficiently large unknown number. Most existing results (*e.g.*, [18, 23, 35, 39, 57, 59, 60]) are applicable in the areas in 🟦 or 🟦, while our results are suitable for almost all areas characterized by 🟨.

As far as we know, most existing works focus on either one-hidden-layer networks (visualized by the region in 🟦 in Figure 1.1), or very deep networks with a constant width (visualized by the region in 🟦 in Figure 1.1). Meanwhile, these works only provide asymptotic[①] approximation errors in terms of the number of parameters, which are valid for particular network architectures. They are unable to give approximation error estimates for other network architectures with the same number of parameters. To overcome this, we provide general approximation error characterizations with explicit formulas for the prefactors, in terms of the width and depth simultaneously (visualized by the region in 🟨 in Figure 1.1), which is of more practical interest in real applications and requires innovative proofs. This gives us much more freedom to design neural networks for a good approximation and we can always give an error estimate via the width and depth no matter what network architecture is given, though the error estimate may not be optimal for

[①] "Asymptotic" means the approximation error is described via big O notation $\mathcal{O}(\cdot)$ without an explicit formula for the prefactor.

unusual architectures. In fact, many results in previous works can be regarded as the corollaries of this dissertation.

Problem 1 and 2 are completely solved by providing approximation error estimates in terms of the width and depth simultaneously. For Problem 3, we use VC-dimension to show our approximation error estimates are nearly optimal for the Hölder continuous function space (Hölder$([0,1]^d, \alpha, \lambda)$ and the smooth function space ($C^s([0,1]^d)$). The optimality becomes insignificant if (nearly) exponential approximation errors are attained.

Table 1.1 (Theorem 4.1) shows that ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ are able to approximate d-dimensional polynomials on $[0,1]^d$ within an error $\mathcal{O}(N^{-L})$. This reveals the power of depth in ReLU networks for approximating polynomials, from function compositions. Generally speaking, such an approximation error is the best (up to constants) what we can expect since ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ are continuous piecewise linear functions with at most $\mathcal{O}(N)^{\mathcal{O}(L)}$ linear pieces. The starting point of a good approximation of functions is to approximate polynomials with high accuracy. In classical approximation theory, the approximation power of a lot of numerical schemes depends on the degree of polynomials that can be locally reproduced. Being able to approximate polynomials with an exponential error plays a vital role in the approximation power of deep ReLU networks. It is interesting to study whether there are any other function spaces with a reasonable size, besides the polynomial space, having an exponential error when approximated by neural networks.

In particular, we introduce new networks built with either Floor ($\lfloor x \rfloor$) or ReLU ($\max\{0, x\}$) as the activation function in each neuron. We call such networks Floor-ReLU networks. It is proved by construction that nearly exponential approximation errors can be attained when using Floor-ReLU networks with fixed architectures to approximate Hölder continuous functions and general continuous functions on $[0,1]^d$. As shown in Table 1.1, approximation errors are improved from polynomial ones to nearly exponential ones by adding a simple activation function (Floor) to ReLU networks. This reveals the power of deep Floor-ReLU network architectures. As we shall see later, the idea of function compositions is the most significant cornerstone of the proofs for the results listed in Table 1.1. Finally, we would like to remark that the architecture of the final Floor-ReLU network is independent of the target function f. That is, only the values of the parameters rely on the target function f. In particular, the choice of activation functions (Floor or ReLU) in each neuron is independent of the target function f.

1.2 Related work

This dissertation is a summary of our previous papers [38,52,53,54], focusing on the approximation error estimate for neural networks. Thus, all the contents of this dissertation focus on three main problems, Problem 1, 2, and 3. In the following, only the previous works related to them are reviewed.

The approximation theory of neural networks has been an active research topic in the past few decades. Previously, as a special kind of ridge function approximation, shallow neural networks with one hidden layer and various activation functions (*e.g.*, wavelets pursuits [12,41], adaptive splines [19,49], radial basis functions [10,18,23, 47,57], sigmoid functions [8,13,14,15,24,32,33,37,40]) were widely discussed and admit good approximation properties, *e.g.*, the universal approximation property [16,24,25], overcoming the curse of dimensionality [3], and providing attractive approximation errors in nonlinear approximation [12,18,19,23,41,49,57].

The introduction of deep neural networks with more than one hidden layers has made significant impacts in many fields in computer science and engineering including computer vision [31] and natural language processing [1]. New scientific computing tools based on deep networks have also emerged and facilitated large-scale and high-dimensional problems that were impractical previously [20,22]. The design of deep ReLU networks and high-performance computing technologies are the key of such a revolution. These breakthroughs have stimulated broad research topics from different points of views to study the power of deep ReLU networks, *e.g.*, in terms of combinatorics [44], topology [7], Vapnik-Chervonenkis (VC) dimension [4,5,51], fat-shattering dimension [2,29], information theory [48], classical approximation theory [3,16,25,38,52,53,54,59], optimization [27,28,45], *etc*.

Particularly in approximation theory, non-quantitative and asymptotic approximation errors of ReLU networks have been proposed for various types of functions. For example, smooth functions [21,34,38,39,58], piecewise smooth functions [48], band-limited functions [43], continuous functions [53,59]. However, to the best of our knowledge, existing theories [17,21,34,39,42,43,48,55,58,59] can only provide implicit formulas. In other words, the approximation error contains an unknown prefactor, or they work only for sufficiently large network size. For example, an approximation error $c_d L^{-2\alpha/d}$ for Lipschitz continuous functions on $[0,1]^d$ is estimated in [59] via a narrow and deep ReLU network with L hidden layers, where c_d is an unknown number depending on d. For another example, the existence of a ReLU network with a constant width and W parameters is constructed in [60] to approximate a smooth function in $C^s([0,1]^d)$ within an error $c_{s,d}W^{-2s/d}(\ln W)^{2s/d}$, where $c_{s,d}$ is still an unknown number depending on s and d. Generally, most of these works can be divided into two cases: 1) networks with varying width and only one hidden layer [18,23,35,57] (visualized by the region in 🟦 in Figure 1.1); 2) networks with a fixed width of $\mathcal{O}(d)$ and a varying depth larger than an unknown number $\mathscr{L}$ [39,59,60] (represented by the region in 🟦 in Figure 1.1).

Almost all works listed above answer Problem 1 and 2 for given activation functions and special function spaces. Most of them estimate the approximation error in terms of the number of parameters. In other words, their approximation errors are only valid for very special network architectures, such as very deep but very narrow networks, complicated networks generated by compositing shallow-wide sub-networks and deep-narrow sub-networks, *etc*., while our approximation error estimates in this dissertation are valid for arbitrary width and depth up to absolute constants. It means the shape of our network architectures is a rectangle with free choice of width (the maximum width of networks) and length (the depth of networks), which is of more practical interest in real applications and requires innovative proofs.

Finally, let us turn to Problem 3. A typical method characterizing optimality in the approximation theory of neural networks is essentially to study the connection between the approximation error and VC-dimension [38,52,53,58,59,60]. Of course, this method relies on the VC-dimension upper bound given in [4]. In this dissertation, we adopt this method with several modifications to simplify the proof. As we shall see later in Section 4.4, the optimality is discussed for two function spaces: 1) the Hölder function space (see Section 4.4.1); 2) the smooth function space (see Section 4.4.2).

This page is intentionally left blank.

Chapter 2

Preliminaries

Before moving to the main body of this dissertation, we first introduce the preliminaries related to this dissertation including notations used throughout this dissertation, the architecture of neural networks, and the general ideas of the approximation by neural networks.

2.1 Notations

For convenience, we present all notations used throughout this dissertation in this section. Several notations used only in a particular section are not presented here.

2.1.1 Basic notations

Basic notations are listed below.

- Let $\mathbb{Z}$ and $\mathbb{R}$ denote the set of integers and real numbers, respectively.
- Let $\mathbb{N}$ denote the set of natural numbers and $\mathbb{N}^+$ denote the set containing all positive integers, *i.e.*, $\mathbb{N} = \{0, 1, 2, \cdots\}$ and $\mathbb{N}^+ = \{1, 2, 3, \cdots\}$.
- Matrices are denoted by bold uppercase letters, *e.g.*, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a real matrix of size $m \times n$, and $\boldsymbol{A}^T$ denotes the transpose of $\boldsymbol{A}$. Correspondingly, $\boldsymbol{A}(i, j)$

is the (i, j)-th entry of $\boldsymbol{A}$; $\boldsymbol{A}(:, j)$ is the j-th column of $\boldsymbol{A}$; $\boldsymbol{A}(i, :)$ is the i-th row of $\boldsymbol{A}$.

- Vectors are denoted as bold lowercase letters, *e.g.*,

$$\boldsymbol{v} = [v_1, \cdots, v_d]^T = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix}$$

is a column vector of size d and $\boldsymbol{v}(i)$ is the i-th element of $\boldsymbol{v}$. For simplicity, a vector $\boldsymbol{v} \in \mathbb{R}^d$ can also be denoted by $\boldsymbol{v} = (v_1, \cdots, v_d)$.

- By convention, "[" and "]" are used to partition matrices (vectors) into blocks, e.g., a matrix $\boldsymbol{A}$ can be partitioned into $\boldsymbol{A} = \left[\begin{smallmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{smallmatrix}\right]$ and a row vector $\boldsymbol{v}$ can be denoted by $\boldsymbol{v} = [v_1, v_2, \cdots, v_d] \in \mathbb{R}^d$.

- We say a map (transform) $\boldsymbol{\mathcal{L}} : \mathbb{R}^m \to \mathbb{R}^n$ is **affine linear** if there exist $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ and $\boldsymbol{b} \in \mathbb{R}^n$ such that $\boldsymbol{\mathcal{L}}(\boldsymbol{x}) = \boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b}$ for any $\boldsymbol{x} \in \mathbb{R}^m$. In particular, an affine linear map is also called a **linear** function in the case $n = 1$.

- For a real number $p \in [1, \infty)$, the p-norm (or ℓ^p-norm) of a vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d$ is defined by

$$\|\boldsymbol{x}\|_p := \big(|x_1|^p + |x_2|^p + \cdots + |x_d|^p\big)^{1/p}.$$

- A d-dimensional multi-index is a d-tuple $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_d) \in \mathbb{N}^d$. Several related notations are listed below.

  * $\|\boldsymbol{\alpha}\|_1 = |\alpha_1| + |\alpha_2| + \cdots + |\alpha_d|$;
  * $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$, where $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$;
  * $\boldsymbol{\alpha}! = \alpha_1!\alpha_2! \cdots \alpha_d!$;
  * $\partial^{\boldsymbol{\alpha}} = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} \cdots \frac{\partial^{\alpha_d}}{\partial x_d^{\alpha_d}}$.

- Let $\mathcal{O}(\cdot)$ denote the big O notation. That is, for any $\boldsymbol{n} \in \mathbb{N}^d$ and functions f and g, $f(\boldsymbol{n}) = \mathcal{O}(g(\boldsymbol{n}))$ means that there exist $C > 0$ and $\boldsymbol{n}_0 \in \mathbb{N}^d$ independent of $\boldsymbol{n}$, f, and g such that $f(\boldsymbol{n}) \le Cg(\boldsymbol{n})$ when $\boldsymbol{n}(i) \ge \boldsymbol{n}_0(i)$ for all i.

- The floor function (Floor) is defined as $\lfloor x \rfloor := \max\{n : n \le x,\ n \in \mathbb{Z}\}$ for any $x \in \mathbb{R}$. $\lfloor \boldsymbol{x} \rfloor$ means applying $\lfloor \cdot \rfloor$ elementwisely to $\boldsymbol{x}$. Similarly, the ceiling function (Ceiling) is defined as $\lceil x \rceil := \min\{n : n \ge x,\ n \in \mathbb{Z}\}$ for any $x \in \mathbb{R}$.

- Similar to "min" and "max", let $\mathrm{mid}(x_1, x_2, x_3)$ denote the middle value of **three** inputs x_1, x_2, and x_3.[1]

- 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 $\mathrm{bin}\, 0.\theta_1\theta_2\cdots\theta_L$ to denote the L-term binary representation of θ, *i.e.*, $\mathrm{bin}\, 0.\theta_1\theta_2\cdots\theta_L := \sum_{\ell=1}^{L} \theta_\ell 2^{-\ell}$.

- Let $\text{Hölder}([0, 1]^d, \alpha, \lambda)$ denote the space of Hölder continuous functions of order $\alpha \in (0, 1]$ on $[0, 1]^d$ with a Hölder constant $\lambda > 0$. To be precise, each function f of $\text{Hölder}([0, 1]^d, \alpha, \lambda)$ satisfies

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \le \lambda \|\boldsymbol{x} - \boldsymbol{y}\|_2^{\alpha}, \quad \text{for any } \boldsymbol{x}, \boldsymbol{y} \in [0, 1]^d.$$

- Given $E \subseteq \mathbb{R}^d$, let $C^s(E)$ denote the set containing all functions, all k-th order partial derivatives of which exist and are continuous on E for any $k \in \mathbb{N}$ with $0 \le k \le s$. In particular, $C^0(E)$, also denoted by $C(E)$, is the set of continuous functions on E. For the case $s = \infty$, $C^\infty(E) = \cap_{s=0}^{\infty} C^s(E)$. The C^s-norm is defined by

$$\|f\|_{C^s(E)} := \max\big\{ \|\partial^{\boldsymbol{\alpha}} f\|_{L^\infty(E)} : \boldsymbol{\alpha} \in \mathbb{N}^d \text{ with } \|\boldsymbol{\alpha}\|_1 \le s \big\}.$$

Generally, E is assigned as $[0, 1]^d$ in this dissertation. In particular, the closed

[1] Note that "mid" can be defined via $\mathrm{mid}(x_1, x_2, x_3) = x_1 + x_2 + x_3 - \max(x_1, x_2, x_3) - \min(x_1, x_2, x_3)$, which can be implemented by a ReLU network with width 14 and depth 2, as shown in Lemma 3.8.

unit ball of $C^s([0,1]^d)$ is denoted by

$$C_u^s([0,1]^d) := \big\{ f \in C^s([0,1]^d) : \|f\|_{C^s([0,1]^d)} \le 1 \big\}.$$

- The modulus of continuity of a continuous function $f \in C([0,1]^d)$ is defined by

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

  Clearly, $\omega_f(nr) \le n\omega_f(r)$ for any $n \in \mathbb{N}^+$ and $r \ge 0$.

2.1.2 Set notations

All set notations used in this dissertation can be found below.

- The Lebesgue measure of a measurable set $S \in \mathbb{R}^d$ is denoted by $\mu(S)$.
- Let $|S|$ denote the size of a finite set S, *i.e.*, the number of all elements in S.
- The set difference of two sets A and B is denoted by $A \backslash B := \{x : x \in A,\ x \notin B\}$.
- For a set of numbers A and a real number x, $A - x := \{y - x : y \in A\}$.
- Let 1_S be the characteristic function on a set S, *i.e.*, 1_S is equal to 1 on S and 0 outside S. S can be simply described by one or more conditions, *e.g.*, $1_{\{n \le m\}}$ is equal to 1 if $n \le m$ and 0 if $n > m$.
- Let $\mathcal{B}(\boldsymbol{x}, r) \subseteq \mathbb{R}^d$ denote the closed ball, in ℓ^2-norm, with a center $\boldsymbol{x} \subseteq \mathbb{R}^d$ and a radius r, *i.e.*,

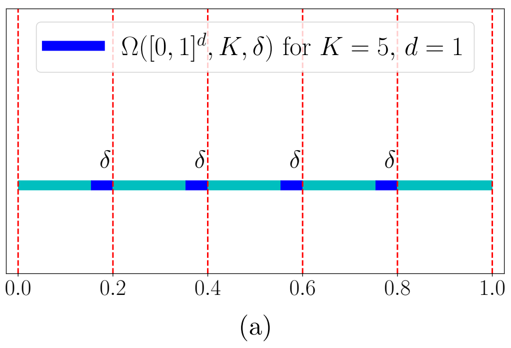
$$\mathcal{B}(\boldsymbol{x}, r) := \big\{ \boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{y}\|_2 \le r \big\}.$$

- 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

$$\Omega([0,1]^d, K, \delta) := \bigcup_{i=1}^{d} \Big\{ \boldsymbol{x} = (x_1, \cdots, x_d) \in [0,1]^d : x_i \in \cup_{k=1}^{K-1} (\tfrac{k}{K} - \delta, \tfrac{k}{K}) \Big\}. \quad (2.1)$$

In particular, $\Omega([0,1]^d, K, \delta) = \emptyset$ if $K = 1$. See Figure 2.1 for two examples of trifling regions.



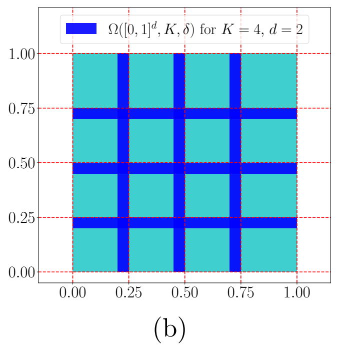


Figure 2.1: Two examples of trifling regions. (a) $K = 5, d = 1$. (b) $K = 4, d = 2$.

2.1.3 Neural network notations

We list neural network notations as follows.

- 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 $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$.

- The expression "a network (architecture) with width N and depth L" means

  $*$ The maximum width of this network (architecture) for all **hidden** layers is no more than N.

  $*$ The number of **hidden** layers of this network (architecture) is no more than L.

- The expression "a (vector-valued) function is implemented by a network (architecture)" means, by specifying the parameters as proper real numbers, this network (architecture) has the same output as this function for each input.

- We use "$\mathcal{NN}$" as "functions implemented by ReLU neural networks" for short and use Python-type notations to specify a class of functions implemented by ReLU networks with several conditions. To be precise, we use $\mathcal{NN}(\mathrm{c}_1;\ \mathrm{c}_2;\ \cdots;\ \mathrm{c}_m)$ to denote the function set containing all functions implemented by ReLU network architectures 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 maximum width of all hidden layers (width), the number of hidden layers (depth), the width in each hidden layer (widthvec), the total number of parameters (#parameter), *etc.* For example, if $\phi \in \mathcal{NN}(\#\mathrm{input} = 2;\ \mathrm{widthvec} = [100, 100];\ \#\mathrm{output} = 1)$, then ϕ is a function satisfying the following conditions.

  ∗ ϕ is a two-dimensional function that maps from $\mathbb{R}^2$ to $\mathbb{R}$.

  ∗ ϕ can be implemented by a two-hidden-layer ReLU network that the number of neurons in each hidden layer is 100.

  We would like to point out that each element of $\mathcal{NN}(\mathrm{c}_1;\ \mathrm{c}_2;\ \cdots;\ \mathrm{c}_m)$ is a continuous piecewise linear function.

2.2 Architecture of neural networks

There are a large number of types of neural network architectures, *e.g.*, convolution neural networks (CNN), recurrent neural networks (RNN), and generative adversarial networks (GAN), variational auto encoders (VAE), residual networks (ResNet), *etc.* This dissertation focuses on feed-forward fully connected neural networks. If there are no special instructions, "feed-forward fully connected neural network(s)" is abbreviated to "network(s)" throughout this dissertation. In this section, we will describe the architecture of networks mathematically and intuitively in

Section 2.2.1 and study the compositions and combinations of network architectures in Section 2.2.2.

2.2.1 Descriptions

First, we use mathematical formulas to describe network architectures. Assume $\varrho_1, \cdots, \varrho_r$ are one-dimensional functions. Let $N_0 = d$, $N_{L+1} \in \mathbb{N}^+$, and N_ℓ be the number of neurons in ℓ-th hidden layer of a network with activation functions $\varrho_1, \cdots, \varrho_r$ for $\ell = 1, 2, \cdots, L$, then the architecture of this network with input $\boldsymbol{x}$ and output $\boldsymbol{\phi}(\boldsymbol{x})$ can be described as

$$\boldsymbol{x} = \widetilde{\boldsymbol{h}}_0 \xrightarrow{\boldsymbol{W}_0,\, \boldsymbol{b}_0} \boldsymbol{h}_1 \xrightarrow{\varrho_1, \cdots, \varrho_r} \widetilde{\boldsymbol{h}}_1 \quad \cdots \quad \xrightarrow{\boldsymbol{W}_{L-1},\, \boldsymbol{b}_{L-1}} \boldsymbol{h}_L \xrightarrow{\varrho_1, \cdots, \varrho_r} \widetilde{\boldsymbol{h}}_L \xrightarrow{\boldsymbol{W}_L,\, \boldsymbol{b}_L} \boldsymbol{h}_{L+1} = \boldsymbol{\phi}(\boldsymbol{x}), \tag{2.2}$$

where $\boldsymbol{W}_\ell \in \mathbb{R}^{N_{\ell+1} \times N_\ell}$, $\boldsymbol{b}_\ell \in \mathbb{R}^{N_{\ell+1}}$,

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

and

$$\widetilde{\boldsymbol{h}}_{\ell,n} \in \big\{\varrho_1(\boldsymbol{h}_{\ell,n}), \cdots, \varrho_r(\boldsymbol{h}_{\ell,n})\big\}, \qquad \text{for } \ell = 1, 2, \cdots, L \text{ and } n = 1, 2, \cdots, N_\ell,$$

where $\boldsymbol{h}_\ell = (\boldsymbol{h}_{\ell,1}, \cdots, \boldsymbol{h}_{\ell,N_\ell})$, $\widetilde{\boldsymbol{h}}_\ell = (\widetilde{\boldsymbol{h}}_{\ell,1}, \cdots, \widetilde{\boldsymbol{h}}_{\ell,N_\ell})$ for each $\ell = 1, 2, \cdots, L$, and $\mathcal{L}_\ell$ is an affine linear map given by $\mathcal{L}_\ell(\boldsymbol{z}) := \boldsymbol{W}_\ell \cdot \boldsymbol{z} + \boldsymbol{b}_\ell$ for each $\ell = 0, 1, 2, \cdots, L$.

The most common type of activation function is the rectified linear unit (ReLU), denoted by σ in this dissertation. We remark that using the ReLU activation function is not much different from using any other **continuous piecewise linear** activation function with finitely many linear pieces. In fact, if we let $\widetilde{\sigma}$ be a continuous piecewise linear activation function with finitely many linear pieces, then we can always replace a network, using one of $\{\sigma, \widetilde{\sigma}\}$ as activation function, by another network having the other activation function in $\{\sigma, \widetilde{\sigma}\}$ while only increasing the width and depth by absolute constant factors.

The networks with only ReLU activation function, *i.e.*, $\varrho_1 = \cdots = \varrho_r = \sigma$ in Equation (2.2), are called ReLU networks. In this case, the set of functions implemented by the architecture in Equation (2.2) is exactly $\mathcal{NN}(\#\text{input} = d;\ \text{widthvec} = [N_1, N_2, \cdots, N_L])$. Moreover, the (vector-valued) function $\boldsymbol{\phi}$ implemented by the network in the Equation (2.2) can also be represented in a compositive manner by

$$\boldsymbol{\phi} = \mathcal{L}_L \circ \sigma \circ \mathcal{L}_{L-1} \circ \sigma \circ \ \cdots \ \circ \mathcal{L}_2 \circ \sigma \circ \mathcal{L}_1 \circ \sigma \circ \mathcal{L}_0.$$

In particular, if $r = 2$, $\varrho_1 = \sigma$, and $\varrho_2(x) = \lfloor x \rfloor$ for any $x \in \mathbb{R}$, the network described by Equation (2.2) is a Floor-ReLU network. We will discuss more details of Floor-ReLU networks in Chapter 5.

To visualize the network architecture, we take ReLU networks as examples. Figure 2.2 provides an example of a ReLU network with width 4 and depth 3. Note that the affine linear transform and the activation function are contained in a single neuron in Figure 2.2. To make the architecture of a ReLU network more clear, we put the affine linear transform and the activation function into different neurons in another example shown in Figure 2.3.

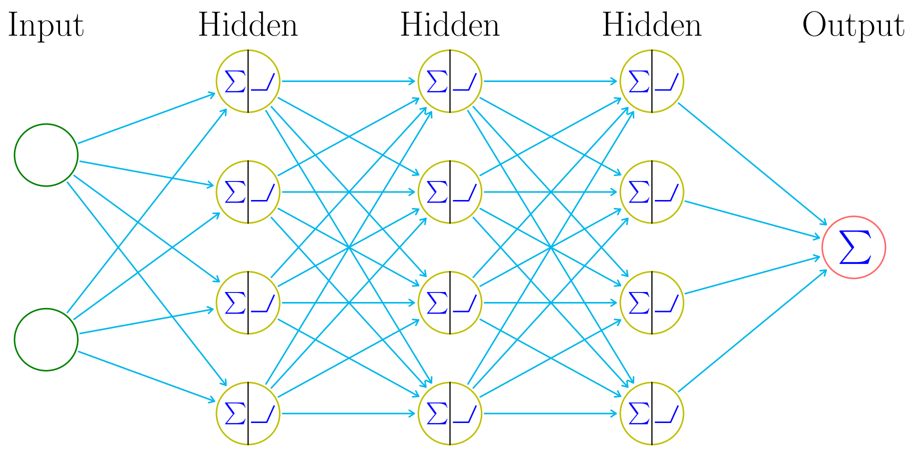
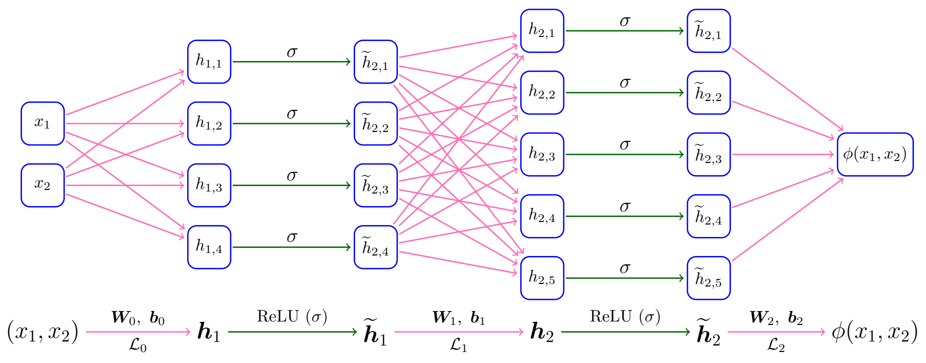


Figure 2.2: An example of a ReLU network with width 4 and depth 3. This network has two neurons in the input layer, one neuron in the output layer, and four neurons in each hidden layer.



x_1
x_2
$h_{1,1}$
$h_{1,2}$
$h_{1,3}$
$h_{1,4}$
σ
$\widetilde{h}_{2,1}$
$\widetilde{h}_{2,2}$
$\widetilde{h}_{2,3}$
$\widetilde{h}_{2,4}$
$h_{2,1}$
$h_{2,2}$
$h_{2,3}$
$h_{2,4}$
$h_{2,5}$
$\phi(x_1,x_2)$
$(x_1,x_2) \xrightarrow[\mathcal{L}_0]{\boldsymbol{W}_0,\ \boldsymbol{b}_0} \boldsymbol{h}_1 \xrightarrow{\text{ReLU }(\sigma)} \widetilde{\boldsymbol{h}}_1 \xrightarrow[\mathcal{L}_1]{\boldsymbol{W}_1,\ \boldsymbol{b}_1} \boldsymbol{h}_2 \xrightarrow{\text{ReLU }(\sigma)} \widetilde{\boldsymbol{h}}_2 \xrightarrow[\mathcal{L}_2]{\boldsymbol{W}_2,\ \boldsymbol{b}_2} \phi(x_1,x_2)$

Figure 2.3: A detailed example of a ReLU network with two inputs x_1, x_2 and an output $\phi(x_1, x_2)$. Here, $\boldsymbol{h}_1 = (h_{1,1}, h_{1,2}, h_{1,3}, h_{1,4})$, $\boldsymbol{h}_2 = (h_{2,1}, h_{2,2}, h_{2,3}, h_{2,4}, h_{2,5})$, $\widetilde{\boldsymbol{h}}_1 = \sigma(\boldsymbol{h}_1) = (\widetilde{h}_{1,1}, \widetilde{h}_{1,2}, \widetilde{h}_{1,3}, \widetilde{h}_{1,4})$, and $\widetilde{\boldsymbol{h}}_2 = \sigma(\boldsymbol{h}_2) = (\widetilde{h}_{2,1}, \widetilde{h}_{2,2}, \widetilde{h}_{2,3}, \widetilde{h}_{2,4}, \widetilde{h}_{2,5})$.

2.2.2 Compositions and combinations

We use a lemma below to describe the compositions and combinations of ReLU network architectures.

Lemma 2.1. *The following three statements hold.*

(i) *For any $N, L, d_1, d_2, d_3, d_4 \in \mathbb{N}^+$, assume that $\mathcal{L}_1 : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ and $\mathcal{L}_2 : \mathbb{R}^{d_3} \to \mathbb{R}^{d_4}$ are two affine linear maps, and $\boldsymbol{\Phi} \in \mathcal{NN}(\#\text{input} = d_2;\ \text{width} \le N;\ \text{depth} \le L;\ \#\text{output} = d_3)$. Then*

$$\boldsymbol{\Phi} \circ \mathcal{L}_1 \in \mathcal{NN}(\#\text{input} = d_1;\ \text{width} \le N;\ \text{depth} \le L;\ \#\text{output} = d_3)$$

and

$$\mathcal{L}_2 \circ \boldsymbol{\Phi} \in \mathcal{NN}(\#\text{input} = d_2;\ \text{width} \le N;\ \text{depth} \le L;\ \#\text{output} = d_4).$$

(ii) *For any $N_1, N_2, L_1, L_2, d_1, d_2, d_3 \in \mathbb{N}^+$, if $\boldsymbol{\Phi}_1 \in \mathcal{NN}(\#\text{input} = d_1;\ \text{width} \le N_1;\ \text{depth} \le L_1;\ \#\text{output} = d_2)$ and $\boldsymbol{\Phi}_2 \in \mathcal{NN}(\#\text{input} = d_2;\ \text{width} \le N_2;\ \text{depth} \le L_2;\ \#\text{output} = d_3)$, then $\boldsymbol{\Phi}_2 \circ \boldsymbol{\Phi}_1$ is in*

$$\mathcal{NN}(\#\text{input} = d_1;\ \text{width} \le \max\{N_1, N_2\};\ \text{depth} \le L_1 + L_2;\ \#\text{output} = d_3).$$

(iii) For any $N_1, N_2, L_1, L_2, d \in \mathbb{N}^+$ and $a, b, c \in \mathbb{R}$ with $N_1 \geq 2$ and $N_2 \geq 2$, if $\phi_1 \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \leq N_1;\ \text{depth} \leq L_1;\ \#\text{output} = 1)$ and $\phi_2 \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \leq N_2;\ \text{depth} \leq L_2;\ \#\text{output} = 1)$, then $a\phi_1 + b\phi_2 + c$ is in

$$\mathcal{NN}(\#\text{input} = d;\ \text{width} \leq N_1 + N_2;\ \text{depth} \leq \max\{L_1, L_2\};\ \#\text{output} = 1)$$

Proof. Let first prove Part (i). The case $L = 1$ is trivial, we consider $L \geq 2$ below. Since $\mathbf{\Phi} \in \mathcal{NN}(\text{width} \leq N;\ \text{depth} \leq L)$, there exist two affine linear maps $\widehat{\mathcal{L}}_1, \widehat{\mathcal{L}}_2$ and $\mathbf{\Psi}_1, \mathbf{\Psi}_2 \in \mathcal{NN}(\text{width} \leq N;\ \text{depth} \leq L - 1)$ such that

$$\mathbf{\Phi} = \mathbf{\Psi}_1 \circ \sigma \circ \widehat{\mathcal{L}}_1, \quad \text{and} \quad \mathbf{\Phi} = \widehat{\mathcal{L}}_2 \circ \sigma \circ \mathbf{\Psi}_2.$$

Therefore,

$$\mathbf{\Phi} \circ \mathcal{L}_1 = \mathbf{\Psi}_1 \circ \sigma \circ \widehat{\mathcal{L}}_1 \circ \mathcal{L}_1 = \mathbf{\Psi}_1 \circ \sigma \circ \widetilde{\mathcal{L}}_1, \quad \text{and} \quad \mathcal{L}_2 \circ \mathbf{\Phi} = \mathcal{L}_2 \circ \widehat{\mathcal{L}}_2 \circ \sigma \circ \mathbf{\Psi}_2 = \widetilde{\mathcal{L}}_2 \circ \sigma \circ \mathbf{\Psi}_2,$$

where $\widetilde{\mathcal{L}}_1 = \widehat{\mathcal{L}}_1 \circ \mathcal{L}_1$ and $\widetilde{\mathcal{L}}_2 = \mathcal{L}_2 \circ \widehat{\mathcal{L}}_2$ are two new affine linear maps, implying

$$\mathbf{\Phi} \circ \mathcal{L}_1 \in \mathcal{NN}(\#\text{input} = d_1;\ \text{width} \leq N;\ \text{depth} \leq L;\ \#\text{output} = d_3)$$

and

$$\mathcal{L}_2 \circ \mathbf{\Phi} \in \mathcal{NN}(\#\text{input} = d_2;\ \text{width} \leq N;\ \text{depth} \leq L;\ \#\text{output} = d_4).$$

Next, let us focus on Part (ii). The case $L_1 = 1$ or $L_2 = 1$ is trivial, so we assume $L_1 \geq 2$ and $L_2 \geq 2$ below. Since $\mathbf{\Phi}_1 \in \mathcal{NN}(\text{width} \leq N_1;\ \text{depth} \leq L_1)$ and $\mathbf{\Phi}_2 \in \mathcal{NN}(\text{width} \leq N_2;\ \text{depth} \leq L_2)$, there exist $\mathbf{\Psi}_1 \in \mathcal{NN}(\text{width} \leq N_1;\ \text{depth} \leq L_1 - 1)$ and $\mathbf{\Psi}_2 \in \mathcal{NN}(\text{width} \leq N_2;\ \text{depth} \leq L_2 - 1)$ such that

$$\mathbf{\Phi}_1 = \mathcal{L}_1 \circ \sigma \circ \mathbf{\Psi}_1, \quad \text{and} \quad \mathbf{\Phi}_2 = \mathbf{\Psi}_2 \circ \sigma \circ \mathcal{L}_2,$$

where $\mathcal{L}_1, \mathcal{L}_2$ are two affine linear maps. Therefore,

$$\mathbf{\Phi}_2 \circ \mathbf{\Phi}_1 = \mathbf{\Psi}_2 \circ \sigma \circ \mathcal{L}_2 \circ \mathcal{L}_1 \circ \sigma \circ \mathbf{\Psi}_1 = \mathbf{\Psi}_2 \circ \sigma \circ \mathcal{L} \circ \sigma \circ \mathbf{\Psi}_1,$$

where $\mathcal{L} = \mathcal{L}_2 \circ \mathcal{L}_1$ is a new affine linear map. Thus, $\mathbf{\Phi}_2 \circ \mathbf{\Phi}_1$ can be implemented by a ReLU network with width $\max\{N_1, N_2\}$ and depth $(L_1-1)+1+1+(L_2-1) = L_1+L_2$, implying $\mathbf{\Phi}_2 \circ \mathbf{\Phi}_1$ is in

$$\mathcal{NN}(\#\text{input} = d_1;\ \text{width} \le \max\{N_1, N_2\};\ \text{depth} \le L_1 + L_2;\ \#\text{output} = d_3).$$

Finally, let us consider Part (iii). Let ι denote the one-dimensional identity map. As shown in Figure 2.4, ι can be understood as an implementation of a ReLU network with an arbitrary number of hidden layers and width 2. Thus, for $j = 1, 2$, $\iota \circ \phi_j$ can

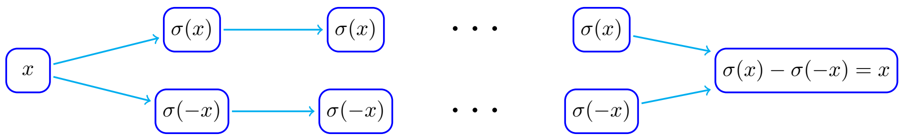


Figure 2.4: An illustration of the implementation of the identity map by a ReLU network based on the fact $\sigma \circ \sigma = \sigma$.

be regarded as an implementation of ReLU network with $\max\{L_1, L_2\}$ hidden layers and width $\max\{N_j, 2\} = N_j$. By placing the two networks implementing $\iota \circ \phi_1$ and $\iota \circ \phi_2$ in parallel (share the inputs), we have

$$\mathbf{\Phi} \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \le N_1 + N_2;\ \text{depth} \le \max\{L_1, L_2\};\ \#\text{output} = 2),$$

where $\mathbf{\Phi} : \mathbb{R}^d \to \mathbb{R}^2$ is defined by $\mathbf{\Phi}(\boldsymbol{x}) = \big(\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x})\big)$ for any $\boldsymbol{x} \in \mathbb{R}^d$. Define an affine linear map $\mathcal{L} : \mathbb{R}^2 \to \mathbb{R}$ via $\mathcal{L}(x, y) = ax + by + c$. By Part (i), we have $a\phi_1 + b\phi_2 + c = \mathcal{L} \circ \mathbf{\Phi} \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \le N_1 + N_2;\ \text{depth} \le \max\{L_1, L_2\};\ \#\text{output} = 1)$. So we finish the proof. $\square$

2.3 General ideas of approximation by networks

In this section, we discuss the general ideas of approximation by networks. Universal approximation theorem shows that one-hidden-layer networks can approximate continuous functions arbitrarily well on $[0,1]^d$ as long as the network size is large enough. However, it is non-trivial to characterize the approximation error in terms of the width and depth simultaneously as we will do in later chapters. Thus, let us discuss the general ideas to warm up the later constructions and proofs.

2.3.1 ReLU networks

First, let us consider the approximation by ReLU networks. To illustrate the general ideas, we take continuous functions as examples. The ideas of smooth functions are similar by applying Taylor expansion, as we shall see later in Section 4.3. To approximate a continuous function f on $[0,1]^d$, we essentially construct a piecewise constant function via function compositions. However, piecewise constant functions cannot be implemented by ReLU networks because of their discontinuity. To overcome this, we introduce the trifling region $\Omega([0,1]^d, K, \delta)$, defined in Equation (2.1), and construct ReLU networks to implement **almost** piecewise linear functions to approximate the target functions outside the trifling region. For the sake of clarity, we divide the main ideas into four steps. See Figure 2.6 for an illustration.

1. Normalize f as $\widetilde{f}$, partition $[0,1]^d$ into a union of sub-cubes[②] $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$ and the trifling region $\Omega([0,1]^d, K, \delta)$, and let $\boldsymbol{x}_{\boldsymbol{\beta}}$ denote the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm, where $K \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{3K}]$ are two numbers determined later. See Figure 2.5 for illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

2. Construct a sub-network to implement a vector-valued function $\boldsymbol{\Phi}_1 : \mathbb{R}^d \to \mathbb{R}^d$ 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 $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

[②] For simplicity, we abbreviate (d-dimensional) hypercube to cube.

3. Construct a sub-network to implement a function $\phi_2 : \mathbb{R}^d \to \mathbb{R}$ mapping the index $\boldsymbol{\beta}$ approximately to $f(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}$. Then $\phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, implying $\widetilde{\phi} := \phi_2 \circ \boldsymbol{\Phi}_1$ approximate $\widetilde{f}$ within an error $\mathcal{O}(\omega_f(1/K))$ outside the trifling region.

4. Re-scale and shift $\widetilde{\phi}$ to obtain a function ϕ approximating f well outside the trifling region. Then modify ϕ to let it approximate f uniformly well on $[0,1]^d$ and determine the network architecture implementing the modified function ϕ.

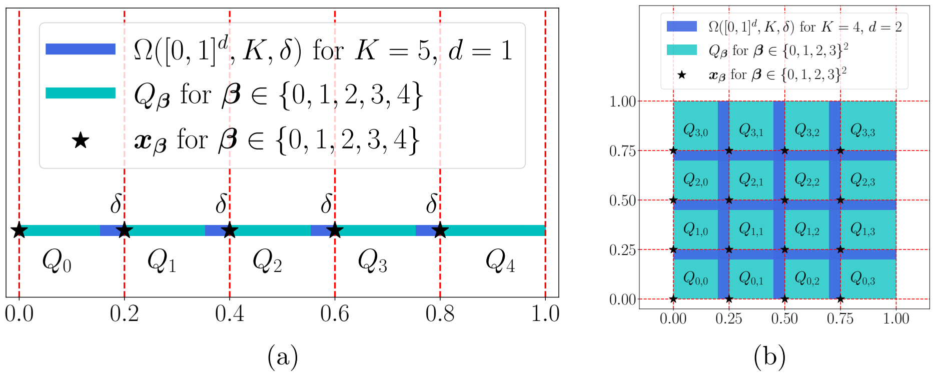

$\Omega([0,1]^d, K, \delta)$ for $K = 5$, $d = 1$
$Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3,4\}$
$\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3,4\}$
δ
Q_0 Q_1 Q_2 Q_3 Q_4
0.0 0.2 0.4 0.6 0.8 1.0
(a)
$\Omega([0,1]^d, K, \delta)$ for $K = 4$, $d = 2$
$Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3\}^2$
$\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3\}^2$
$Q_{3,0}$ $Q_{3,1}$ $Q_{3,2}$ $Q_{3,3}$
$Q_{2,0}$ $Q_{2,1}$ $Q_{2,2}$ $Q_{2,3}$
$Q_{1,0}$ $Q_{1,1}$ $Q_{1,2}$ $Q_{1,3}$
$Q_{0,0}$ $Q_{0,1}$ $Q_{0,2}$ $Q_{0,3}$
0.00 0.25 0.50 0.75 1.00
(b)

Figure 2.5: Illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. (a) $K = 5$ and $d = 1$. (b) $K = 4$ and $d = 2$.

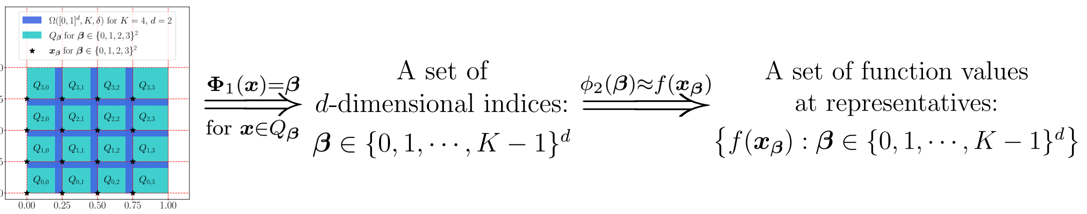

$\boldsymbol{\Phi}_1(\boldsymbol{x})=\boldsymbol{\beta}$
for $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$
A set of
d-dimensional indices:
$\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$
$\phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}})$
A set of function values
at representatives:
$\{f(\boldsymbol{x}_{\boldsymbol{\beta}}) : \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d\}$

Figure 2.6: An illustration of the main ideas of constructing $\phi = \phi_2 \circ \boldsymbol{\Phi}_1$. Note that $\phi \approx f$ on $[0,1]^d \backslash \Omega([0,1]^d, K, \delta)$, since $\phi(\boldsymbol{x}) = \phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx f(\boldsymbol{x}_{\boldsymbol{\beta}})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$.

The first step is straightforward. The construction of $\boldsymbol{\Phi}_1$ in Step 2 is trivial if the network size is large enough. To control the width and depth of the network implementing $\boldsymbol{\Phi}_1$, we establish a theorem, Theorem 3.12 in Section 3.4, to help construct $\boldsymbol{\Phi}_1$. Assume ϕ_1 is the one-dimensional step function attained by Theorem 3.12, then we can attain $\mathbf{\Phi}_1$ via defining

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

See Figure 2.7 for an illustration.

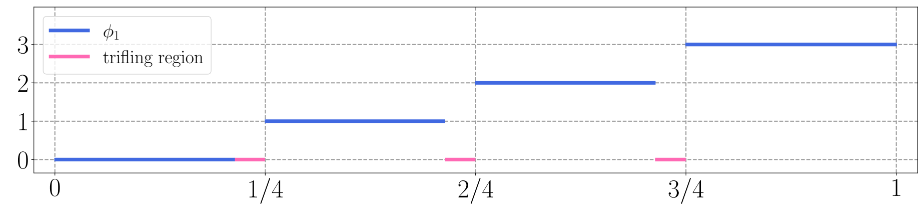


Figure 2.7: An example of a step function for the case $K = 4$ and $d = 1$. We do not need to care about the values of ϕ_1 in the trifling region while constructing a ReLU network to implement ϕ_1.

Step 3 is the core step. We would like to point out that we only need to let ϕ_2 map $\boldsymbol{\beta}$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ within an error $\mathcal{O}(\omega_f(1/K))$ for each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$ when constructing ϕ_2 in Step 3. In other words, it is not necessary to care about the values of ϕ_2 outside the set of points $\{0, 1, \cdots, K-1\}^d$, which plays a key role in constructing a ReLU network to implement ϕ_2 in Step 3. Thus, with $\mathbf{\Phi}_1$ in hand, a function approximation problem is converted to a point fitting problem for ϕ_2 via the idea of function compositions $(\phi_2 \circ \mathbf{\Phi}_1)$,[3] which reveals the power of function compositions in some sense. However, designing a network to solve such a point fitting problem is still a challenging task due to the limitation of the width and depth of the target network. To simplify the construction of a ReLU network solving this point fitting problem, we investigate the width power (Theorem 3.2) and the depth power (Theorem 3.4) of ReLU networks to fit a collection of points in Section 3.2.1 and 3.2.2, respectively. Then we can construct the desired ReLU network by combining these two properties together.

The final step is pretty technical, since ϕ may oscillate greatly in the trifling region. To overcome this, we use two main ideas: "horizontal shift" and "middle

[3] Solving a point fitting problem is to design a function to fit a collection of points $\{(\boldsymbol{x}_i, y_i)\}_i$ in $\mathbb{R}^{d+1}$, namely, the target function takes the value close to y_i at the location $\boldsymbol{x}_i$.

value". For example, if g approximates a one-dimensional continuous function f well except for an interval in $\mathbb{R}$ with a small length δ, then

$$\operatorname{mid}\big(g(x-\delta), g(x), g(x+\delta)\big)$$

can approximate f well on the whole domain $\mathbb{R}$, where $\operatorname{mid}(\cdot,\cdot,\cdot)$ is a function returning the middle value of three inputs. See Section 3.3 for more details.

2.3.2 Floor-ReLU networks

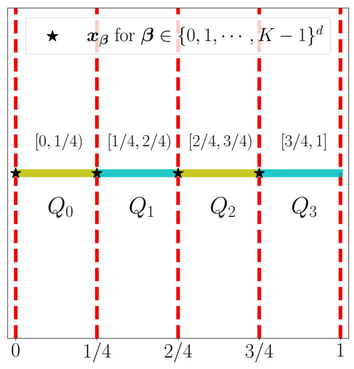
Next, let us discuss the general ideas of the approximation by Floor-ReLU networks, which are similar to those of ReLU networks except for the trifling region. Since Floor-ReLU networks can approximate step functions uniformly well on $[0,1]^d$, we do not need to introduce the trifling region again. The main ideas can be divided into four steps as follows.

1. Normalize f as $\widetilde{f}$ satisfying $\widetilde{f}(\boldsymbol{x}) \in [0,1]$ for any $\boldsymbol{x} \in [0,1]^d$, partition $[0,1]^d$ into a set of non-overlapping cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$, and denote $\boldsymbol{x}_{\boldsymbol{\beta}}$ as the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm, where K is an integer determined later. See Figure 2.8 for the illustrations of $Q_{\boldsymbol{\beta}}$ and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

2. Construct a Floor-ReLU sub-network to implement a vector-valued function $\boldsymbol{\Phi}_1 : \mathbb{R}^d \to \mathbb{R}^d$ projecting the whole cube $Q_{\boldsymbol{\beta}}$ to the index $\boldsymbol{\beta}$ for each $\boldsymbol{\beta}$, *i.e.*, $\boldsymbol{\Phi}_1(\boldsymbol{x}) = \boldsymbol{\beta}$ for all $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

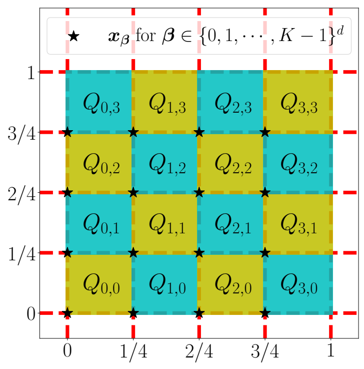
3. Construct a Floor-ReLU sub-network to implement a function $\phi_2 : \mathbb{R}^d \to \mathbb{R}$ mapping $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$, *i.e.*, $\phi_2(\boldsymbol{\beta}) \approx \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}$. Then $\phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$, implying $\widetilde{\phi} := \phi_2 \circ \boldsymbol{\Phi}_1$ approximates $\widetilde{f}$ within an error $\mathcal{O}(\omega_f(1/K))$ on $[0,1]^d$.

4. Re-scale and shift $\widetilde{\phi}$ to obtain the desired function ϕ approximating f well and determine the final Floor-ReLU network to implement ϕ.



★ $\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$
$[0,1/4)$
$[1/4,2/4)$
$[2/4,3/4)$
$[3/4,1]$
Q_0
Q_1
Q_2
Q_3
0
1/4
2/4
3/4
1

(a)



★ $\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$
$Q_{0,3}$ $Q_{1,3}$ $Q_{2,3}$ $Q_{3,3}$
$Q_{0,2}$ $Q_{1,2}$ $Q_{2,2}$ $Q_{3,2}$
$Q_{0,1}$ $Q_{1,1}$ $Q_{2,1}$ $Q_{3,1}$
$Q_{0,0}$ $Q_{1,0}$ $Q_{2,0}$ $Q_{3,0}$
0
1/4
2/4
3/4
1

(b)

Figure 2.8: Illustrations of $Q_{\boldsymbol{\beta}}$ and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. (a) $K = 4,\ d = 1$. (b) $K = 4,\ d = 2$.

The implementations of Step 1, 2, and 4 are straightforward. Step 3 is the core step. Similar to ReLU networks, we only need to solve a point fitting problem due to the power of function compositions. It is still a highly technical problem. Thus, we introduce a proposition, Proposition 5.6, to help implement this step. As we shall see later in Section 5.3, the key idea of proving Proposition 5.6 is the modified "bit extraction" technique derived from [5].

Finnaly, we would like to point out that the key reason Floor-ReLU networks can attain much better approximation errors than those of ReLU networks is that Floor ($\lfloor x \rfloor$) has infinite (constant) pieces, while ReLU ($\max\{0,x\}$) has only two (linear) pieces. Thus, roughly speaking, one Floor activation function can do what many ReLU activation functions do in our construction. For this reason, compared to ReLU networks, Floor-ReLU networks attain significantly better approximation errors.

Chapter 3

Basic results of ReLU networks

In this chapter, we introduce several basic results of ReLU networks, which will be used in the later chapters.

3.1 Wide networks to deep ones

Generally, it is easier to construct shallow and wide sub-networks to meet the requirements during designing the final network. To control the width of the final network, we consider representing wide and shallow networks by deep and narrow ones. To this end, we establish a theorem, Theorem 3.1 below, to convert wide networks with two hidden layers to deep and narrow ones.

Theorem 3.1. *For any $N, L, d \in \mathbb{N}^+$, it holds that*

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

This theorem shows that if a function ϕ can be implemented by a two-hidden-layer ReLU network that the first and second hidden layers have N and NL neurons, respectively, then there exists a new ReLU network with width $2N + 2$ and depth $L + 1$ to implement ϕ.

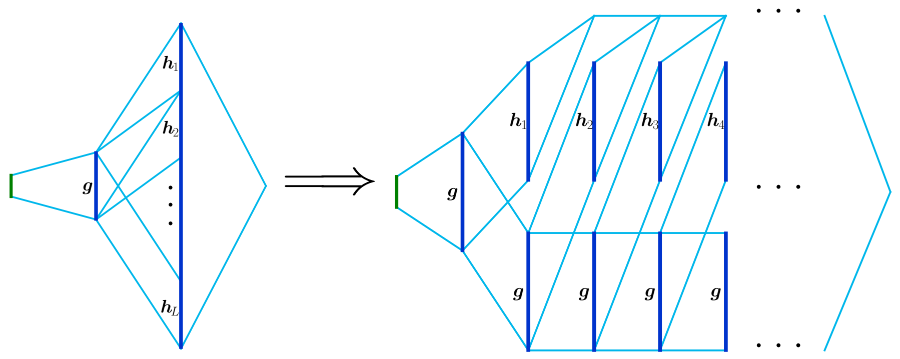


Figure 3.1: An illustration of the main idea of proving Theorem 3.1

The key idea to prove Theorem 3.1 is to re-assemble sub-networks in the shallower network in the left of Figure 3.1 to form a deeper one with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ on the right of Figure 3.1.

Proof of Theorem 3.1. For any $\phi \in \mathcal{NN}(\#\text{input} = d;\ \text{widthvec} = [N, NL];\ \#\text{output} = 1)$, ϕ can be implemented by a ReLU network described as

$$\boldsymbol{x} \xrightarrow[\sigma]{\boldsymbol{W}_0,\ \boldsymbol{b}_0} \boldsymbol{g} \xrightarrow[\sigma]{\boldsymbol{W}_1,\ \boldsymbol{b}_1} \boldsymbol{h} \xrightarrow{\boldsymbol{W}_2,\ \boldsymbol{b}_2} \phi(\boldsymbol{x}),$$

where $\boldsymbol{g}$ and $\boldsymbol{h}$ are the output of the first and second hidden layers, respectively. That is,

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

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

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

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

$\boldsymbol{W}_{2,\ell} \in \mathbb{R}^{1\times N}$ for $\ell = 1, 2, \cdots, L$. Then, for $\ell = 1, 2, \cdots, L$, we have

$$\boldsymbol{h}_\ell = \sigma(\boldsymbol{W}_{1,\ell} \cdot \boldsymbol{g} + \boldsymbol{b}_{1,\ell}) \quad \text{and} \quad \phi(\boldsymbol{x}) = \boldsymbol{W}_2 \cdot \boldsymbol{h} + \boldsymbol{b}_2 = \sum_{j=1}^{L} \boldsymbol{W}_{2,j} \cdot \boldsymbol{h}_j + \boldsymbol{b}_2. \tag{3.1}$$

Define

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

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

$$s_\ell = s_{\ell-1} + \boldsymbol{W}_{2,\ell} \cdot \boldsymbol{h}_\ell, \qquad \text{for } \ell = 1, 2, \cdots, L. \tag{3.2}$$

Hence, it is easy to check that ϕ can be also implemented by the deep network shown in Figure 3.2. Clearly, the network architecture in Figure 3.2 is with width $2N+2$

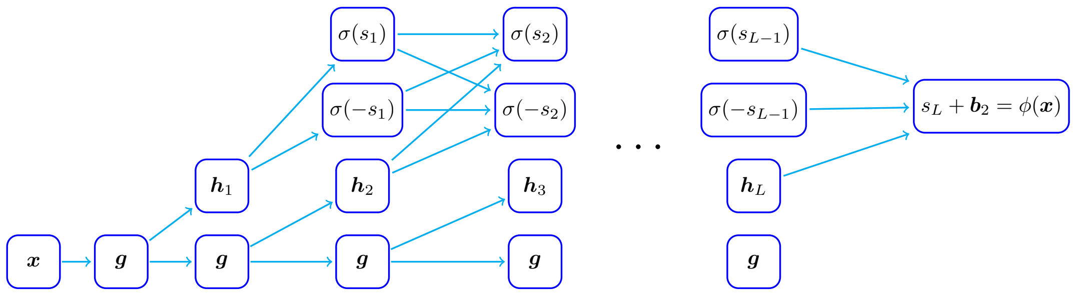


Figure 3.2: An illustration of the desired network implementing ϕ based on Equation (3.1) and (3.2), and the fact $x = \sigma(x) - \sigma(-x)$ for any $x \in \mathbb{R}$.[1]

and depth $L+1$. So we finish the proof. □

3.2 Power of networks to fit points

As mentioned earlier in Section 2.3, we need to construct a ReLU sub-networks with the desired width and depth to solve a point fitting problem. To this end,

[1] In this figure, we omit ReLU (σ) for a neuron if its output is non-negative without ReLU. Such a simplification will be applied to similar figures in the rest of this dissertation.

we discuss the power of ReLU networks to fit points from two perspectives: 1) the width power of ReLU networks to fit points in Section 3.2.1; 2) the depth power of ReLU networks to fit points in Section 3.2.2.

3.2.1 Width power of networks to fit points

Let us first discuss the width power of ReLU network to fit points. Roughly speaking, we would like to minimize the width by fixing the depth when constructing ReLU networks to fit a given number of points. In fact, we prove in Theorem 3.2 that a function $\phi \in \mathcal{NN}(\#\text{input} = 1;\ \text{widthvec} = [2m, 2n+1];\ \#\text{output} = 1)$ can fit $m(n+1)+1$ points in $\mathbb{R}^2$ with several conditions.

Theorem 3.2. *For any $m, n \in \mathbb{N}^+$, given any $m(n+1)+1$ samples $(x_i, y_i) \in \mathbb{R}^2$ with $x_0 < x_1 < x_2 < \cdots < x_{m(n+1)}$ and $y_i \geq 0$ for $i = 0, 1, \cdots, m(n+1)$, there exists $\phi \in \mathcal{NN}(\#\text{input} = 1;\ \text{widthvec} = [2m, 2n+1];\ \#\text{output} = 1)$ satisfying the following three conditions.*

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

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

(iii) ϕ is bounded by a constant determined by m, n, x_i, y_i for $i = 0, 1, \cdots, m(n+1)$. To be exact,

$$\sup_{x \in [x_0, x_{m(n+1)}]} |\phi(x)| \leq C \max_{i \in \{0,1,\cdots,m(n+1)\}} y_i,$$

where

$$C = 1 + \prod_{k=1}^{n} \left(1 + \max\left\{\frac{x_{j(n+1)+n} - x_{j(n+1)+k-1}}{x_{j(n+1)+k} - x_{j(n+1)+k-1}} : j = 0, 1, \cdots, m-1\right\}\right).$$

We would like to point out that ϕ may not be linear on an interval $[x_{i-1}, x_i]$ for some $i \in \{j(n+1) : j = 1, 2, \cdots, m\}$. So ϕ may oscillate greatly in the region

$$\bigcup_{i \in \{j(n+1) : j = 1, 2, \cdots, m\}} [x_{i-1}, x_i],$$

which is called the "don't-care" region in the proof of Theorem 3.2. However, we are able to choose the values of $x_0, x_1, \cdots, x_{m(n+1)}$ properly to make the "don't-care" region small enough, the idea of which is similar to that of the trifling region defined in Equation (2.1).

Before proving Theorem 3.2, let us first study the properties of ReLU networks with only one hidden layer to warm up in Lemma 3.3 below. Recall that for a continuous piecewise linear function $f(x)$, the x values where the slope changes are typically called **breakpoints**.

Lemma 3.3. *Suppose* $\phi \in \mathrm{NN}(\#\mathrm{input} = 1;\ \mathrm{widthvec} = [N];\ \#\mathrm{output} = 1)$ *can be implemented by a ReLU network architecture*

$$x \xrightarrow{\boldsymbol{W}_0,\ \boldsymbol{b}_0} \boldsymbol{h} \xrightarrow{\sigma} \widetilde{\boldsymbol{h}} \xrightarrow{\boldsymbol{W}_1,\ \boldsymbol{b}_1} \phi(x).$$

That is, $\phi(x)$ is a function determined by $\boldsymbol{W}_0$, $\boldsymbol{b}_0$, $\boldsymbol{W}_1$, and $\boldsymbol{b}_1$. Given a sequence of strictly increasing numbers x_0, $x_1, \cdots$, x_N, set $\boldsymbol{W}_0 = (1, 1, \cdots, 1) \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{b}_0 = (-x_0, -x_1, \cdots, -x_{N-1}) \in \mathbb{R}^N$. Then we have

*(i) The breakpoints of ϕ are exactly $x_0, x_1, \cdots, x_N$ on the interval $[x_0, x_N]$*②*;*

(ii) For any sequence $(y_i)_{i=0}^N$, we are able to choose $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ properly such that $\phi(x_i) = y_i$ for $i = 0, 1, \cdots, N$ and ϕ is linear on each interval $[x_i, x_{i+1}]$ for $i = 0, 1, \cdots, N-1$.

Part (i) in Lemma 3.3 is straightforward. The existence in Part (ii) is equivalent to the existence of a solution for a non-singular system of linear equations, which is left for the reader.

With Lemma 3.3 in hand, we are ready to prove Theorem 3.2.

Proof of Theorem 3.2. For any $\phi \in \mathcal{NN}(\#\mathrm{input} = 1;\ \mathrm{widthvec} = [2m, 2n{+}1];\ \#\mathrm{output} =$

②We only consider the interval $[x_0, x_N]$ and hence x_0 and x_N are treated as breakpoints. $\phi(x)$ might not have a real breakpoint in a small open neighborhood of x_0 or x_N.

1), ϕ can be implemented by the following ReLU network architecture

$$x \xrightarrow{\boldsymbol{W}_0,\ \boldsymbol{b}_0} \boldsymbol{h} \xrightarrow{\sigma} \widetilde{\boldsymbol{h}} \xrightarrow{\boldsymbol{W}_1,\ \boldsymbol{b}_1} \boldsymbol{g} \xrightarrow{\sigma} \widetilde{\boldsymbol{g}} \xrightarrow{\boldsymbol{W}_2,\ \boldsymbol{b}_2} \phi(x). \tag{3.3}$$

Clearly, $\phi(x)$ is a function determined by $\boldsymbol{W}_0, \boldsymbol{b}_0, \boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2$. So our goal is to choose $\boldsymbol{W}_0, \boldsymbol{b}_0, \boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2$ properly in order to make Condition (i)-(iii) true.

Note that $\boldsymbol{g} = \boldsymbol{g}(x)$ is a vector-valued function mapping $x \in \mathbb{R}$ to $\boldsymbol{g}(x) \in \mathbb{R}^{2n+1}$ and determined by $\boldsymbol{W}_0, \boldsymbol{b}_0, \boldsymbol{W}_1, \boldsymbol{b}_1$. Hence each entry of $\boldsymbol{g}(x)$ itself is a function implemented by a sub-network with one hidden layer. Denote $\boldsymbol{g} = (g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-)$, then $\{g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-\} \subseteq \mathcal{NN}(\#\text{input} = 1;\ \text{widthvec} = [2m];\ \#\text{output} = 1)$. See Figure 3.3 for an illustration of $\boldsymbol{g} = (g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-)$ and $\widetilde{\boldsymbol{g}} = \sigma(\boldsymbol{g}) = (\widetilde{g}_0, \widetilde{g}_1^+, \widetilde{g}_1^-, \cdots, \widetilde{g}_n^+, \widetilde{g}_n^-)$ for the case $m = n = 2$. Our proof of Theorem 3.2 is mainly

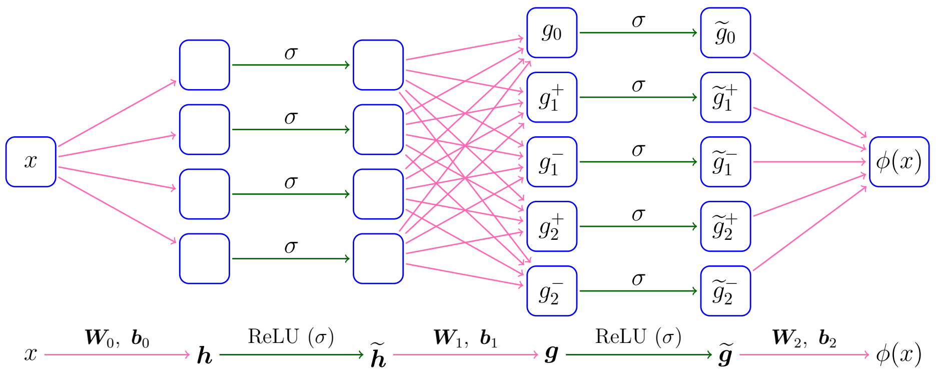


Figure 3.3: An illustration of $\boldsymbol{g} = (g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-)$ and $\widetilde{\boldsymbol{g}} = \sigma(\boldsymbol{g}) = (\widetilde{g}_0, \widetilde{g}_1^+, \widetilde{g}_1^-, \cdots, \widetilde{g}_n^+, \widetilde{g}_n^-)$ for the case $m = n = 2$.

based on the repeated applications of Lemma 3.3 to determine $\boldsymbol{W}_0, \boldsymbol{b}_0, \boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2$ such that Conditions (i)-(iii) hold.

To simplify the notations, we define two index sets $\mathcal{I}_1(m,n)$ and $\mathcal{I}_2(m,n)$ for any $m, n \in \mathbb{N}^+$ as

$$\mathcal{I}_1(m,n) := \{j(n+1) : j = 1, 2, \cdots, m\}$$

and

$$\mathcal{I}_2(m,n) := \mathcal{I}_1(m,n) \cup \big(\mathcal{I}_1(m,n) - 1\big) \cup \{0\},$$

where $\mathcal{I}_1(m,n)-1 = \big\{k-1 : k \in \mathcal{I}_1(m,n)\big\}$. For example, $\mathcal{I}_1(4,4) = \{5,10,15,20\}$ and $\mathcal{I}_2(4,4) = \{0,\ 4,5,\ 9,10,\ 14,15,\ 19,20\}$.

Step 1: Determine $\boldsymbol{W}_0$ and $\boldsymbol{b}_0$.

Clearly, the index set $\mathcal{I}_2(m,n)$ has $2m+1$ elements. Convert the point set $\big\{x_i : i \in \mathcal{I}_2(m,n)\big\}$ in ascending order to a vector $\boldsymbol{\xi} = (\xi_0, \xi_1, \cdots, \xi_{2m}) \in \mathbb{R}^{2m+1}$. Then set $\boldsymbol{W}_0 = (1,1,\cdots,1) \in \mathbb{R}^{2m\times 1}$ and $\boldsymbol{b}_0 = \big(-\xi_0, -\xi_1, \cdots, -\xi_{2m-1}\big) \in \mathbb{R}^{2m}$. Note that $\xi_{2m} = x_{m(n+1)}$ is the right endpoint of the interval $[x_0, x_{m(n+1)}]$. By Lemma 3.3 (set $N = 2m$ therein), we have

- All functions in $\{g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-\}$ have the same set of breakpoints

$$\{\xi_j : j = 0,1,\cdots,2m\} = \big\{x_i : i \in \mathcal{I}_2(m,n)\big\},$$

  that is, each function in $\{g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-\}$ is linear between any two adjacent points of $\big\{x_i : i \in \mathcal{I}_2(m,n)\big\}$, no matter what $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ are.

- We are able to identify $\boldsymbol{W}_1 \in \mathbb{R}^{(2n+1)\times 2m}$ and $\boldsymbol{b}_1 \in \mathbb{R}^{2n+1}$ to freely determine the values of each function in $\{g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-\}$ at all points of $\big\{x_i : i \in \mathcal{I}_2(m,n)\big\}$.

Step 2: Determine $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$.

This is the key step of the proof. Our ultimate goal is to set up

$$\boldsymbol{g} = (g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-)$$

by determining $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ such that, after a nonlinear activation function (ReLU), there exists a linear combination in the last step of our network (specified by $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$ as shown in Equation (3.3)) that can generate a desired function $\phi(x)$ matching the sample points $\{(x_i, y_i)\}_{0\le i\le m(n+1)}$. In the previous step, we have determined the breakpoints of $\{g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-\}$ by setting up $\boldsymbol{W}_0$ and $\boldsymbol{b}_0$; in this step, we will identify $\boldsymbol{W}_1 \in \mathbb{R}^{(2n+1)\times 2m}$ and $\boldsymbol{b}_1 \in \mathbb{R}^{2n+1}$ to fully determine $\{g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-\}$. This will be conducted in two sub-steps.

Step 2.1: Set up.

Let $f_0(x)$ be a continuous piecewise linear function defined on $[0,1]$ satisfying

- $f_0(x_i) = y_i$ for all $i \in \{0, 1, \cdots, m(n+1)\}$.
- f_0 is linear between any two adjacent points of $\big\{x_i : i \in \{0, 1, \cdots, m(n+1)\}\big\}$.

Note that $\big\{x_i : i \in \mathcal{I}_2(m,n)\big\}$ is the set of breakpoints of g_0. By Lemma 3.3 and the setting of Step 1, we are able to choose $\boldsymbol{W}_1(1,:)$ and $\boldsymbol{b}_1(1)$ properly such that $g_0(x_i) = f_0(x_i)$ for all $i \in \mathcal{I}_2(m,n)$ and g_0 is linear between any two adjacent points of $\big\{x_i : i \in \mathcal{I}_2(m,n)\big\}$.

We would like to inductively construct a sequence of f_k for all $k \in \{1, 2, \cdots, n+1\}$ satisfying

- $f_k(x_i) = 0$ for all $i \in \cup_{\ell=0}^{k-1}(\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\}$.
- f_k is linear on each interval $[x_{i-1}, x_i]$ for all $i \notin \mathcal{I}_1(m,n)$.

As we shall see later in Step 3, the construction of the final function ϕ is mainly based on f_{n+1}.

First, let us consider the case $k = 1$. Define $f_1 := f_0 - \widetilde{g}_0$, where $\widetilde{g}_0 = \sigma(g_0) = g_0$ as shown in Equation (3.3), since g_0 is positive by the construction of Lemma 3.3. Note that

$$(\mathcal{I}_1(m,n) - n - 1) \cup \{m(n+1)\} \;=\; \{j(n+1) : j = 0, 1, \cdots, m\} \;\subseteq\; \mathcal{I}_2(m,n).$$

Then we have

- $f_1(x_i) = f_0(x_i) - \widetilde{g}_0(x_i) = 0$ for all $i \in (\mathcal{I}_1(m,n) - n - 1) \cup \{m(n+1)\}$.
- f_1 is linear on each interval $[x_{i-1}, x_i]$ for all i.

Thus, the desired f_1 has been constructed. See Figure 3.4 (a) for an illustration of f_0, f_1, and g_0.

Step 2.2: Mathematical induction.

The initialization of the mathematical induction, f_1, has been constructed in Step 2.1. Hence, it is enough to show how to proceed with an arbitrary k. See Figure 3.4 (b)-(d) for the illustration of the first two induction steps.

Now assume f_k is defined for some $k \in \{1, 2, \cdots, n\}$, we need to construct f_{k+1} satisfying similar conditions as follows.

- $f_{k+1}(x_i) = 0$ for all $i \in \cup_{\ell=0}^{k}(\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\}$.
- f_{k+1} is linear on each interval $[x_{i-1}, x_i]$ for all $i \notin \mathcal{I}_1(m,n)$.

Then we shall determine

$$\boldsymbol{W}_1(2k,:), \quad \boldsymbol{b}_1(2k), \quad \boldsymbol{W}_1(2k+1,:), \quad \text{and} \quad \boldsymbol{b}_1(2k+1)$$

to completely specify g_k^+ and g_k^-, which in turn can determine f_{k+1}. This induction process can be further divided into four sub-steps.

Step 2.2.1: Define index sets.

Define

$$\Lambda_k^+(m,n) := \{j : f_k(x_{j(n+1)+k}) \geq 0,\ 0 \leq j < m\}$$

and

$$\Lambda_k^-(m,n) := \{j : f_k(x_{j(n+1)+k}) < 0,\ 0 \leq j < m\}.$$

Clearly, $\Lambda_k^+(m,n) \cup \Lambda_k^-(m,n) = \{0, 1, \cdots, m-1\}$. Recall that g_k^+ and g_k^- are two continuous piecewise linear functions with the same set of breakpoints $\big\{x_i : i \in \mathcal{I}_2(m,n)\big\}$. We will use $\Lambda_k^+(m,n)$ and $\Lambda_k^-(m,n)$ to generate $2m+1$ samples in

$$\Big\{(x,y) \in \mathbb{R}^2 : x \in \big\{x_i : i \in \mathcal{I}_2(m,n)\big\}\Big\}$$

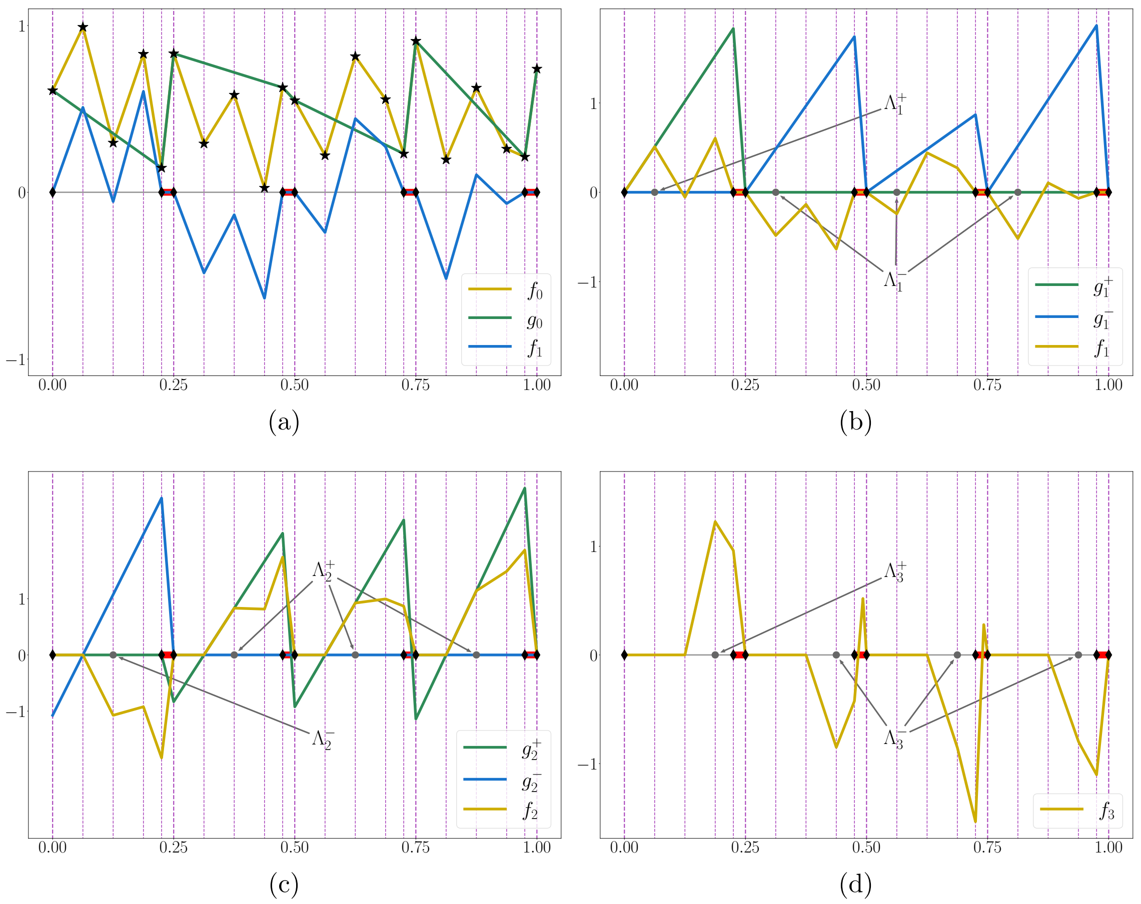


Figure 3.4: Illustrations of the proof of Theorem 3.2, especially Step 2 of the proof, when $m = n = 4$, with the "don't-care" region $\cup_{i\in\mathcal{I}_1(m,n)}[x_{i-1}, x_i]$ in red. $g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-$ share the same set of breakpoints $\{x_i : i \in \mathcal{I}_2(m,n)\}$ marked with black "diamonds". Λ_k^+ and Λ_k^- are short of $\Lambda_k^+(m,n)$ and $\Lambda_k^-(m,n)$, respectively. (a) Given samples $\{(x_i, y_i) : i = 0, 1, \cdots, m(n+1)\}$ marked with black "stars", let $f_0(x)$ be the continuous piecewise linear function fitting these samples, construct g_0 such that $f_1 = f_0 - \sigma(g_0)$ is closer to 0 than f_0 in a larger subset of the "important" region. (b) Construct g_1^+ and g_1^- such that $f_2 = f_1 - \sigma(g_1^+) + \sigma(g_1^-)$ is closer to 0 than f_1 in a larger subset of the "important" region. (c) Construct g_2^+ and g_2^- such that $f_3 = f_2 - \sigma(g_2^+) + \sigma(g_2^-)$ is closer to 0 than f_2 in a larger subset of the "important" region. (d) The visualization of f_3, which is 0 in the "important" areas that have been processed and may remain large near the "don't-care" region. f_k will decay quickly to 0 outside the "don't-care" region as k increases.

to **fully determine** g_k^+ and g_k^- by identifying $\boldsymbol{W}_1(2k,:)$, $\boldsymbol{b}_1(2k)$, $\boldsymbol{W}_1(2k+1,:)$, and $\boldsymbol{b}_1(2k+1)$ in the following steps.

Step 2.2.2: Determine $\boldsymbol{W}_2(2k,:)$ and $\boldsymbol{b}_2(2k)$.

By Lemma 3.3 and the setting of Step 1, we can choose $\boldsymbol{W}_2(2k,:)$ and $\boldsymbol{b}_2(2k)$ to fully determine g_k^+ such that each $g_k^+(x_i)$ matches a specific value for all $i \in \mathcal{I}_2(m,n)$. Note that $\mathcal{I}_2(m,n)$ is the union of three sets: $\{m(n+1)\}$,

$$\left\{j(n+1) : j \in \Lambda_k^+(m,n) \cup \Lambda_k^-(m,n)\right\},$$

and

$$\left\{j(n+1)+n : j \in \Lambda_k^+(m,n) \cup \Lambda_k^-(m,n)\right\}.$$

The values of $\left\{g_k^+(x_i) : i \in \mathcal{I}_2(m,n)\right\}$ are specified as as follows.

- If $j \in \Lambda_k^+(m,n)$, specify the values of $g_k^+(x_{j(n+1)})$ and $g_k^+(x_{j(n+1)+n})$ such that

  $$g_k^+(x_{j(n+1)+k-1}) = 0 \quad \text{and} \quad g_k^+(x_{j(n+1)+k}) = f_k(x_{j(n+1)+k}) \geq 0.$$

  The existence of these values fulfilling the requirements above comes from the fact that g_k^+ is linear on the interval $[x_{j(n+1)}, x_{j(n+1)+n}]$ and g_k^+ only depends on the values of $g_k^+(x_{j(n+1)+k-1})$ and $g_k^+(x_{j(n+1)+k})$ on $[x_{j(n+1)}, x_{j(n+1)+n}]$. See Figure 3.5 for an illustration. Now it is easy to verify that $\widetilde{g}_k^+ = \sigma(g_k^+)$ satisfies

  ∗ $\widetilde{g}_k^+(x_{j(n+1)+\ell}) = 0$ for $\ell = 0, 1, \cdots, k-1$ and

  $$\widetilde{g}_k^+(x_{j(n+1)+k}) = f_k(x_{j(n+1)+k}) \geq 0.$$

  ∗ $\widetilde{g}_k^+$ is linear on each interval $[x_{j(n+1)+\ell}, x_{j(n+1)+\ell+1}]$ for all $\ell \in \{0, 1, \cdots, n-1\}$.

- If $j \in \Lambda_k^-(m,n)$, let $g_k^+(x_{j(n+1)}) = g_k^+(x_{j(n+1)+n}) = 0$. Then $\widetilde{g}_k^+ = \sigma(g_k^+) = 0$ on the interval $[x_{j(n+1)}, x_{j(n+1)+n}]$.

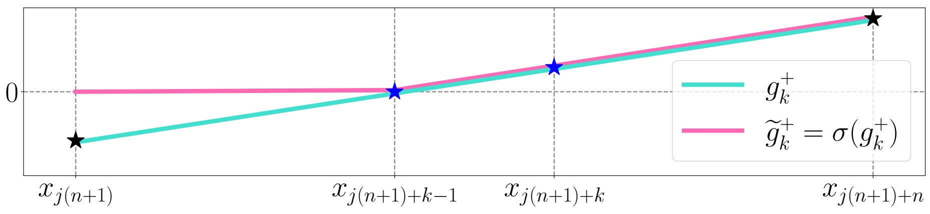


Figure 3.5: An illustration of g_k^+ and $\widetilde{g}_k^+ = \sigma(g_k^+)$. To design f_{k+1} with $f_{k+1}(x_{j(n+1)+k}) = 0$, we shall specify the y-coordinates of two blue "stars" as $f_k(x_{j(n+1)+k-1}) = 0$ and $f_k(x_{j(n+1)+k}) \geq 0$, respectively. Four "stars" should be kept in a straight line. Thus, two blue "stars" determine two black "stars", which in turn determine g_k^+ on $[x_{j(n+1)}, x_{j(n+1)+n}]$ since the x-coordinates of two black "stars" are two adjacent breakpoints of g_k^+. By doing so, we have $f_k - \widetilde{g}_k^+ = 0$ at $x_{j(n+1)+\ell}$ for $\ell = 0, 1, \cdots, k$, which is a big step forward in constructing f_{k+1}.

- Finally, specify the value of g_k^+ at $x_{m(n+1)}$ as 0.

Step 2.2.3: Determine $\boldsymbol{W}_2(2k+1, :)$ and $\boldsymbol{b}_2(2k+1)$.

Similarly, we choose $\boldsymbol{W}_2(2k+1, :)$ and $\boldsymbol{b}_2(2k+1)$ such that g_k^- matches specific values as follows.

- If $j \in \Lambda_k^-(m,n)$, specify the values of $g_k^-(x_{j(n+1)})$ and $g_k^-(x_{j(n+1)+n})$ such that

$$g_k^-(x_{j(n+1)+k-1}) = 0 \quad \text{and} \quad g_k^-(x_{j(n+1)+k}) = -f_k(x_{j(n+1)+k}) > 0.$$

  Then $\widetilde{g}_k^- = \sigma(g_k^-)$ satisfies

  * $\widetilde{g}_k^-(x_{j(n+1)+\ell}) = 0$ for $\ell = 0, 1, \cdots, k-1$ and

$$\widetilde{g}_k^-(x_{j(n+1)+k}) = -f_k(x_{j(n+1)+k}) > 0.$$

  * $\widetilde{g}_k^-$ is linear on each interval $[x_{j(n+1)+\ell}, x_{j(n+1)+\ell+1}]$ for all $\ell \in \{0, 1, \cdots, n-1\}$.

- If $j \in \Lambda_k^+(m,n)$, let $g_k^-(x_{j(n+1)}) = g_k^-(x_{j(n+)+n}) = 0$. Then $\widetilde{g}_k^- = \sigma(g_k^-) = 0$ on the interval $[x_{j(n+1)}, x_{j(n+1)+n}]$.

- Finally, specify the value of g_k^- at $x_{m(n+1)}$ as 0.

Step 2.2.4: Construct f_{k+1} from g_k^+ and g_k^-.

For the sake of clarity, the properties of g_k^+ and g_k^- constructed in Step 2.2.3 are summarized below.

- $\widetilde{g}_k^+(x_i) = \widetilde{g}_k^-(x_i) = 0$ for all $i \in \cup_{\ell=0}^{k-1}(\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\}$.
- If $j \in \Lambda_k^+(m,n)$, $\widetilde{g}_k^+(x_{j(n+1)+k}) = f_k(x_{j(n+1)+k}) \geq 0$ and $\widetilde{g}_k^-(x_{j(n+1)+k}) = 0$.
- If $j \in \Lambda_k^-(m,n)$, $\widetilde{g}_k^-(x_{j(n+1)+k}) = -f_k(x_{j(n+1)+k}) > 0$ and $\widetilde{g}_k^+(x_{j(n+1)+k}) = 0$.
- $\widetilde{g}_k^+$ and $\widetilde{g}_k^-$ are linear on each interval $[x_{j(n+1)+\ell}, x_{j(n+1)+\ell+1}]$ for each $\ell \in \{0, 1, \cdots, n-1\}$ and each $j \in \Lambda_k^+(m,n) \cup \Lambda_k^-(m,n) = \{0, 1, \cdots, m-1\}$. In other words, $\widetilde{g}_k^+$ and $\widetilde{g}_k^-$ are linear on each interval $[x_{i-1}, x_i]$ for all $i \notin \{j(n+1) : j = 1, 2, \cdots, m\} = \mathcal{I}_1(m,n)$.

See Figure 3.4 (a)-(c) for the illustration of g_0, g_1^+, g_1^-, g_2^+, and g_2^-, and to verify their properties as listed above. By the induction hypothesis, we have

- $f_k(x_i) = 0$ for all $i \in \cup_{\ell=0}^{k-1}(\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\}$.
- f_k is linear on each interval $[x_{i-1}, x_i]$ for all $i \notin \mathcal{I}_1(m,n)$.

Thus, $f_k(x_i) - \widetilde{g}_k^+(x_i) + \widetilde{g}_k^-(x_i) = 0$ for all i in

$$\Big(\cup_{\ell=0}^{k-1} (\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\}\Big) \bigcup \Big\{ j(n+1) + k : j \in \Lambda_k^+(m,n) \cup \Lambda_k^-(m,n) \Big\}$$
$$= \cup_{\ell=0}^{k}(\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\},$$

where the equality comes from the fact $\Lambda_k^+(m,n) \cup \Lambda_k^-(m,n) = \{0, 1, \cdots, m-1\}$.

Therefore, by defining

$$f_{k+1} \coloneqq f_k - \widetilde{g}_k^+ + \widetilde{g}_k^-,$$

we have

- $f_{k+1}(x_i) = 0$ for all $i \in \cup_{\ell=0}^{k}(\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\}$.

- f_{k+1} is linear on each interval $[x_{i-1}, x_i]$ for all $i \notin \mathcal{I}_1(m,n)$.

See Figure 3.4 (b)-(d) for the illustration of f_1, f_2, and f_3, and to verify their properties as listed just above. This finishes the mathematical induction process. As we can imagine based on Figure 3.4, when k increases, the support of f_k shrinks to the "don't-care" region.

Step 3: Determine $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$.

With the special vector-valued function $\boldsymbol{g} = (g_0, g_1^+, g_1^-, \cdots, g_n^+, g_n^-)$ constructed in Step 2, we are able to specify $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$ to generate a desired $\phi(x)$ matching the samples $\{(x_i, y_i)\}_{0 \le i \le m(n+1)}$.

In fact, we can simply set $\boldsymbol{W}_2 = (1,\ 1, -1,\ 1, -1, \cdots,\ 1, -1) \in \mathbb{R}^{1\times(2n+1)}$ and $\boldsymbol{b}_2 = 0$, which finishes the implementation of $\phi = \widetilde{g}_0 + \sum_{\ell=1}^{n} \widetilde{g}_\ell^+ - \sum_{\ell=1}^{n} \widetilde{g}_\ell^-$. The rest of the proof is to verify the properties of ϕ. By the principle of mathematical induction, we have

- $f_{n+1} = f_1 + \sum_{\ell=1}^{n}(\widetilde{g}_\ell^- - \widetilde{g}_\ell^+) = f_0 - \widetilde{g}_0 - \sum_{\ell=1}^{n} \widetilde{g}_\ell^+ + \sum_{\ell=1}^{n} \widetilde{g}_\ell^- = f_0 - \phi$.

- $f_{n+1}(x_i) = 0$ for all i in

$$\cup_{\ell=0}^{n}(\mathcal{I}_1(m,n) - n - 1 + \ell) \cup \{m(n+1)\} \ = \ \{0, 1, \cdots, m(n+1)\}.$$

- f_{n+1} is linear on each interval $[x_{i-1}, x_i]$ for all $i \notin \mathcal{I}_1(m,n)$.

Hence, $\phi = \widetilde{g}_0 + \sum_{\ell=1}^{n} \widetilde{g}_\ell^+ - \sum_{\ell=1}^{n} \widetilde{g}_\ell^- = f_0 - f_{n+1}$. Then

$$\phi(x_i) = f_0(x_i) - f_{n+1}(x_i) = y_i, \quad \text{for all } i \in \{0, 1, \cdots, m(n+1)\},$$

which verifies Condition (i), and $\phi = f_0 - f_{n+1}$ is linear on each interval $[x_{i-1}, x_i]$ for $i \notin \mathcal{I}_1(m,n)$, which verifies Condition (ii). It remains to check that ϕ satisfies Condition (iii).

By the definition of $f_1 = f_0 - \widetilde{g}_0$, we have

$$-\max_{i\in\{0,1,\cdots,m(n+1)\}} y_i \le -\widetilde{g}_0(x) \le f_0(x) - \widetilde{g}_0(x) \le f_0(x) \le \max_{i\in\{0,1,\cdots,m(n+1)\}} y_i,$$

for any $x \in [x_0,\, x_{m(n+1)}]$, implying

$$\sup_{x\in[x_0,\, x_{m(n+1)}]} |f_1(x)| \le \max_{i\in\{0,1,\cdots,m(n+1)\}} y_i.$$

By the induction process in Step 2, for any $k \in \{1, 2, \cdots, n\}$, it holds that

$$\sup_{x\in[x_0,\, x_{m(n+1)}]} |\widetilde{g}_k^+(x)| \le C_k(m,n) \sup_{x\in[x_0,\, x_{m(n+1)}]} |f_k(x)|$$

and

$$\sup_{x\in[x_0,\, x_{m(n+1)}]} |\widetilde{g}_k^-(x)| \le C_k(m,n) \sup_{x\in[x_0,\, x_{m(n+1)}]} |f_k(x)|,$$

where

$$C_k(m,n) := \max\left\{\frac{x_{j(n+1)+n} - x_{j(n+1)+k-1}}{x_{j(n+1)+k} - x_{j(n+1)+k-1}} : j = 0, 1, \cdots, m-1\right\}.$$

Since either $\widetilde{g}_k^+(x)$ or $\widetilde{g}_k^-(x)$ is equal to 0 for any $x \in [x_0,\, x_{m(n+1)}]$, we have

$$\sup_{x\in[x_0,\, x_{m(n+1)}]} |\widetilde{g}_k^+(x) - \widetilde{g}_k^-(x)| \le C_k(m,n) \sup_{x\in[x_0,\, x_{m(n+1)}]} |f_k(x)|.$$

It follows from $f_{k+1} = f_k - \widetilde{g}_k^+ + \widetilde{g}_k^-$ that, for any $k \in \{1, 2, \cdots, n\}$,

$$\begin{aligned}\sup_{x\in[x_0,\, x_{m(n+1)}]} |f_{k+1}(x)| &\le \sup_{x\in[x_0,\, x_{m(n+1)}]} |\widetilde{g}_k^+(x) - \widetilde{g}_k^-(x)| + \sup_{x\in[x_0,\, x_{m(n+1)}]} |f_k(x)| \\ &\le \big(C_k(m,n) + 1\big) \sup_{x\in[x_0,\, x_{m(n+1)}]} |f_k(x)|.\end{aligned}$$

Hence,

$$
\begin{aligned}
\sup_{x\in[x_0,x_{m(n+1)}]} |f_{n+1}(x)| &\le \left(\prod_{k=1}^{n} \big(C_k(m,n)+1\big)\right) \sup_{x\in[x_0,x_{m(n+1)}]} |f_1(x)| \\
&\le \left(\prod_{k=1}^{n} \big(C_k(m,n)+1\big)\right) \max_{i\in\{0,1,\cdots,m(n+1)\}} y_i.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\sup_{x\in[x_0,x_{m(n+1)}]} |\phi(x)| &\le \sup_{x\in[x_0,x_{m(n+1)}]} |f_0(x)| + \sup_{x\in[x_0,x_{m(n+1)}]} |f_{n+1}(x)| \\
&\le \left(1+\prod_{k=1}^{n} \big(C_k(m,n)+1\big)\right) \max_{i\in\{0,1,\cdots,m(n+1)\}} y_i \\
&:= C \max_{i\in\{0,1,\cdots,m(n+1)\}} y_i,
\end{aligned}
$$

where

$$
C = 1+\prod_{k=1}^{n}\left(1+\max\left\{\frac{x_{j(n+1)+n}-x_{j(n+1)+k-1}}{x_{j(n+1)+k}-x_{j(n+1)+k-1}} : j=0,1,\cdots,m-1\right\}\right).
$$

So we finish the proof. □

3.2.2 Depth power of networks to fit points

Next, let us discuss the depth power of ReLU networks to fit points. Roughly speaking, we would like to minimize the depth by fixing the width while constructing ReLU networks to fit a given number of points. In fact, we prove in Theorem 3.4 that a function $\phi \in \mathcal{NN}(\#\text{input}=1;\ \text{width}\le 8N+6;\ \text{depth}\le 5L+7;\ \#\text{output}=1)$ can fit N^2L^2 points in $\mathbb{R}^2$ with several conditions.

Theorem 3.4. *For any $N,L\in\mathbb{N}^+$ and any $\theta_i\in\{0,1\}$ for $i=0,1,\cdots,N^2L^2-1$, there exists a function ϕ implemented by a ReLU network with width $8N+6$ and depth $5L+7$ such that*

$$
\phi(i)=\theta_i,\quad \text{for } i=0,1,\cdots,N^2L^2-1.
$$

We would like to remark that the key idea in the proof of Theorem 3.4 is the "bit extraction" technique in [5], which allows us to store L bits in a binary number $\text{bin}0.\theta_1\theta_2\cdots\theta_L$ and extract each bit θ_i. The extraction operator can be efficiently carried out via a deep ReLU network architecture, demonstrating the power of depth.

Next, we introduce Theorem 3.5, a variant of Theorem 3.4, which is easier to prove and can deduce 3.4 simply. Theorem 3.4 and 3.5 characterize the depth power of ReLU networks. Both of them will be used in the later chapters.

Theorem 3.5. *For any $N, L \in \mathbb{N}^+$, any $\theta_{m,\ell} \in \{0,1\}$ for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$, where $M = N^2L$, there exists a function ϕ implemented by a ReLU network with width $4N+3$ and depth $3L+3$ such that*

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

We denote $M = N^2L$ in Theorem 3.5 because it is roughly the number of parameters. The choice of outputting $\sum_{j=0}^{\ell} \theta_{m,j}$ rather than $\theta_{m,\ell}$ not only guarantees the proof of Theorem 3.4 but also simplifies the construction of ReLU networks to approximate continuous functions in $C([0,1]^d)$ in Section 4.2.

Theorem 3.5 will be proven later in this section. Let us first prove Theorem 3.4 based on Theorem 3.5.

Proof of Theorem 3.4. The case $L = 1$ is clear. We assume $L \geq 2$ below.

Denote $M = N^2L$, then $N^2L^2 = ML$. For each $i \in \{0, 1, \cdots, N^2L^2 - 1\}$, there exists a unique representation $i = mL + \ell$ for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$. Thus, we can define, for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$,

$$a_{m,\ell} := \theta_i, \quad \text{where } i = mL + \ell.$$

Then, for each $m \in \{0, 1, \cdots, M-1\}$, we set $b_{m,0} = 0$ and $b_{m,\ell} = a_{m,\ell-1}$ for $\ell = 1, \cdots, L-1$.

By Theorem 3.5, there exist $\phi_1, \phi_2 \in \mathcal{NN}(\text{width} \le 4N+3;\ \text{depth} \le 3L+3)$ such that

$$\phi_1(m,\ell) = \sum_{j=0}^{\ell} a_{m,j} \quad \text{and} \quad \phi_2(m,\ell) = \sum_{j=0}^{\ell} b_{m,j},$$

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

We consider the sample set

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

Its size is $2M + 1 = N \cdot \big((2NL-1)+1\big) + 1$. By Theorem 3.2 (set $m = N$ and $n = 2NL - 1$ therein), there exists

$$\begin{aligned}\psi &\in \mathcal{NN}(\text{widthvec} = [2N, 2(2NL-1)+1]) \\ &= \mathcal{NN}(\text{widthvec} = [2N, 4NL-1])\end{aligned}$$

such that

- $\psi(ML) = M$ and $\psi(mL) = \psi\big((m+1)L-1\big) = m$ for $m = 0, 1, \cdots, M-1$.
- ψ is linear on each interval $[mL, (m+1)L-1]$ for $m = 0, 1, \cdots, M-1$.

It follows that

$$\psi(x) = m, \quad \text{if } x \in [mL, (m+1)L-1], \quad \text{for } m = 0, 1, \cdots, M-1,$$

implying

$$\psi(mL+\ell) = m \quad \text{for } m = 0, 1, \cdots, M-1 \text{ and } \ell = 0, 1, \cdots, L-1.$$

For $i = 0, 1, \cdots, N^2L^2 - 1$, by representing $i = mL + \ell$ for $m = 0, 1, \cdots, M-1$

and $\ell = 0, 1, \cdots, L-1$, we have $\psi(i) = \psi(mL+\ell) = m$ and $i - L\psi(i) = \ell$, deducing

$$\begin{aligned} &\phi_1\big(\psi(i), i - L\psi(i)\big) - \phi_2\big(\psi(i), i - L\psi(i)\big) \\ &= \phi_1(m,\ell) - \phi_2(m,\ell) = \sum_{j=0}^{\ell} a_{m,j} - \sum_{j=0}^{\ell} b_{m,j} \\ &= \sum_{j=0}^{\ell} a_{m,j} - \sum_{j=1}^{\ell} a_{m,j-1} - b_0 = a_{m,\ell} = \theta_i. \end{aligned} \tag{3.4}$$

Therefore, the desired function ϕ can be implemented by the network architecture described in Figure 3.6.

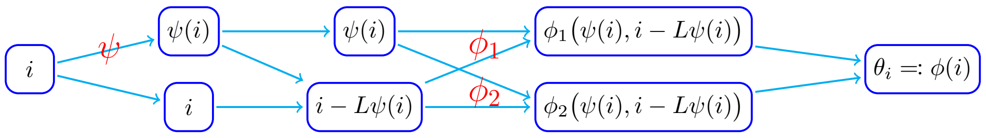


Figure 3.6: An illustration of the network architecture implementing the desired function ϕ based on Equation (3.4) for $i = 0, 1, \cdots, N^2L^2 - 1$.[3]

Note that

$$\phi_1, \phi_2 \in \mathcal{NN}(\text{width} \le 4N+3;\ \text{depth} \le 3L+3).$$

By Theorem 3.1,

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

Hence, the network architecture shown in Figure 3.6 is with width

$$\max\{4L+2+1, 2(4L+3)\} = 8N+6$$

[3]In this figure, "ψ", "ϕ_1", and "ϕ_2" and cyan arrows ("$\longrightarrow$") adjacent to them represent the ReLU networks implementing themselves. We use similar notations in the rest of this dissertation. For example, "$\xrightarrow{\phi}$" means the network architecture that implements a function $\phi : \mathbb{R}^2 \to \mathbb{R}$.

and depth

$$(2L+1)+2+(3L+3)+1=5L+7,$$

implying $\phi \in \mathcal{NN}(\text{width} \le 8N+6;\ \text{depth} \le 5L+7)$. So we finish the proof. □

It remains to prove Theorem 3.5, which relies on the "bit extraction" technique introduced in [5]. We modify this technique to extract the sum of many bits rather than one bit and this modification can be summarized in Lemma 3.6 below.

Lemma 3.6 (Bit extraction). *For any $L \in \mathbb{N}^+$, there exists a function ϕ in*

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

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

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

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

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

and

$$\mathcal{T}(x) \coloneqq \begin{cases} 1, & x \ge 0, \\ 0, & x < 0. \end{cases}$$

Then we have

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

and

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

We would like to point out that, by above two iteration equations, we can iteratively get $\xi_1, \theta_1, \xi_2, \theta_2, \cdots, \xi_L, \theta_L$ when $\xi_1 = \text{bin}\,0.\theta_1\theta_2\cdots\theta_L$ is given. Based on this idea, the rest proof can be divided into three steps.

Step 1: Simplify two iteration equations.

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

$$\begin{aligned} \theta_j = \mathcal{T}(\xi_j - 1/2) &= \sigma\big((\xi_j - 1/2)/\delta + 1\big) - \sigma\big((\xi_j - 1/2)/\delta\big) \\ &= \sigma\big(\mathcal{L}(\xi_j) + 1\big) - \sigma\big(\mathcal{L}(\xi_j)\big), \end{aligned} \tag{3.5}$$

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

$$\xi_{j+1} = 2\xi_j - \theta_j = 2\xi_j - \sigma\big(\mathcal{L}(\xi_j) + 1\big) + \sigma\big(\mathcal{L}(\xi_j)\big). \tag{3.6}$$

Step 2: Design a ReLU network to output $\sum_{j=1}^{\ell} \theta_j$.

It is easy to design a ReLU network to output $\theta_1, \theta_2, \cdots, \theta_L$ by Equation (3.5) and (3.6) when using $\xi_1 = \text{bin}\,0.\theta_1\theta_2 \cdots \theta_L$ as the input. However, it is highly non-trivial to construct a ReLU network to output $\sum_{j=1}^{\ell} \theta_j$ with another input ℓ, since many operations like multiplication and comparison are not allowed in designing ReLU networks.

Now let us establish a formula to represent $\sum_{j=1}^{\ell} \theta_j$ in a form of a ReLU network. Recall two facts: 1) $x_1 x_2 = \sigma(x_1 + x_2 - 1)$ for any $x_1, x_2 \in \{0, 1\}$; 2) $\mathcal{T}(n) = \sigma(n + 1) - \sigma(n)$ for any integer n. Thus, for $\ell = 1, 2, \cdots, L$, we have

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

To simplify the notations, we define

$$z_{\ell,j} \coloneqq \sigma\big(\theta_j + \sigma(\ell - j + 1) - \sigma(\ell - j) - 1\big), \tag{3.7}$$

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

$$\sum_{j=1}^{\ell} \theta_j = \sum_{j=1}^{L} z_{\ell,j}, \quad \text{for } \ell = 1, 2, \cdots, L. \tag{3.8}$$

With Equation (3.5), (3.6), (3.7), and (3.8) in hand, it is easy to construct a function ϕ implemented by a ReLU network with the desired width and depth outputting $\sum_{j=1}^{\ell} \theta_j = \sum_{j=1}^{L} z_{\ell,j}$ for the given input $(\xi_1, \ell) = (\text{bin}\,0.\theta_1\theta_2\cdots\theta_L, \ell)$ for $\ell \in \{1, 2, \cdots, L\}$ and $\theta_1, \theta_2, \cdots, \theta_L \in \{0, 1\}$. The detailed construction is shown in Figure 3.7. It is easy to verify by Figure 3.7 that

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

So we finish the proof. □

With Lemma 3.6 in hand, we are ready to prove Theorem 3.5.

Proof of Theorem 3.5. Define

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

Consider the sample set $\{(m, y_m) : m = 0, 1, \cdots, M\}$, whose size is $M + 1 = N\big((NL - 1) + 1\big) + 1$. By Theorem 3.2 (set $m = N$ and $n = NL - 1$ therein), there exists

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

such that

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

By Lemma 3.6, there exists

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

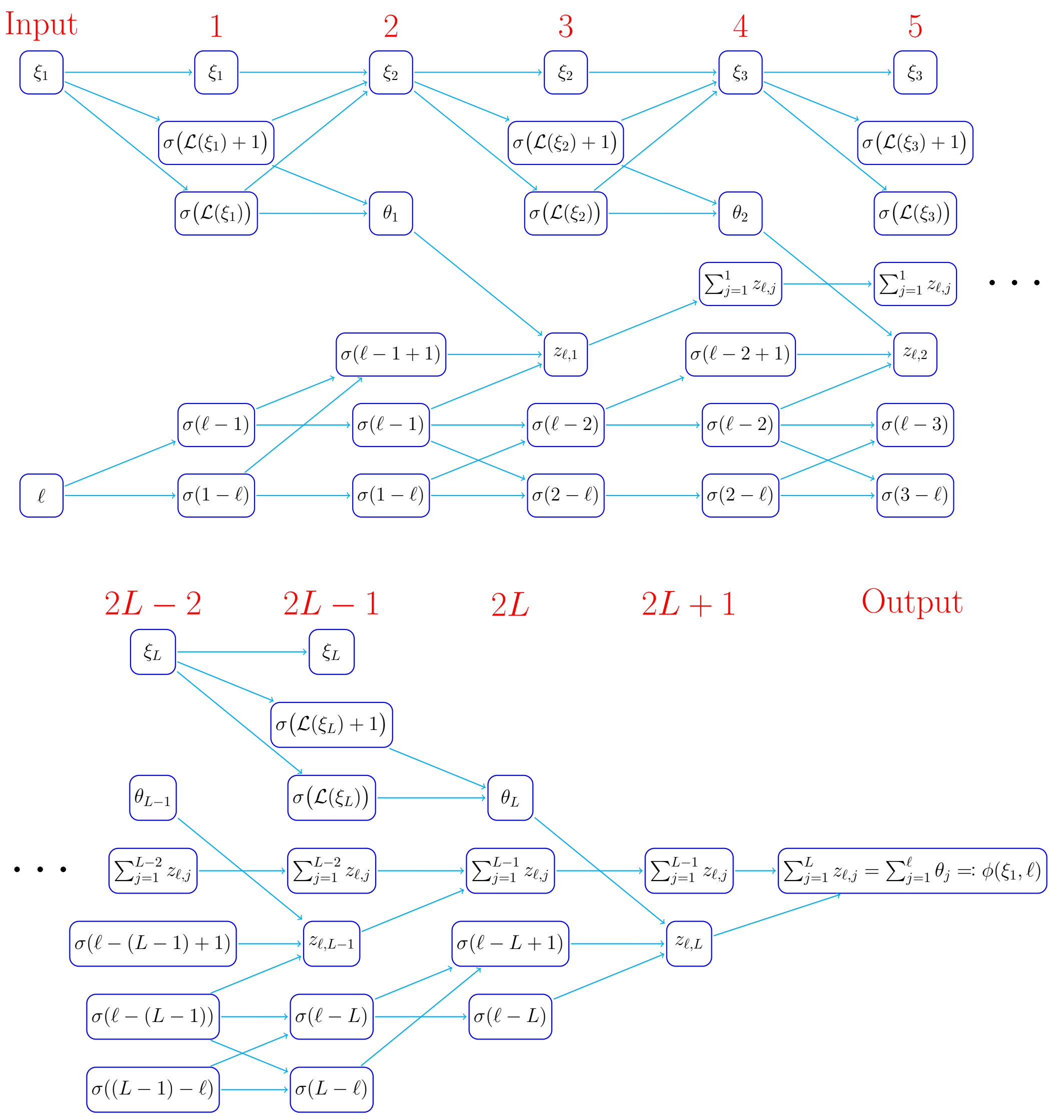


Figure 3.7: An illustration of the target ReLU network implementing ϕ to output $\sum_{j=1}^{L} z_{j,\ell} = \sum_{j=1}^{\ell} \theta_j = \phi(\xi_1, \ell)$ for the given input $(\xi_1, \ell) = (\text{bin}\,0.\theta_1\theta_2\cdots\theta_L, \ell)$ for $\ell \in \{1, 2, \cdots, L\}$ and $\theta_1, \theta_2, \cdots, \theta_L \in \{0, 1\}$. The construction is mainly based on Equation (3.5), (3.6), (3.7), and (3.8). The red numbers above the architecture indicate the order of hidden layers and every two adjacent layers builds a whole iteration step. We output both $\sigma(\ell - j)$ and $\sigma(j - \ell)$ in a hidden layer because we can get the value $\ell - j$ in the next hidden layer because of the fact $x = \sigma(x) - \sigma(-x)$ for any $x \in \mathbb{R}$. We omit ReLU (σ) for a neuron if its output is non-negative without ReLU. Note that all parameters of this network are determined by Equation (3.5), (3.6), (3.7), and (3.8), which are valid no matter what $\theta_1, \theta_2, \cdots, \theta_L \in \{0, 1\}$ are. Thus, the desired function ϕ implemented by this network is independent of $\theta_1, \theta_2, \cdots, \theta_L \in \{0, 1\}$.

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

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

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

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

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

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

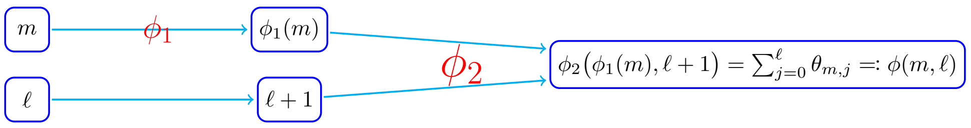

m
ϕ_1
$\phi_1(m)$
ℓ
$\ell+1$
ϕ_2
$\phi_2\big(\phi_1(m), \ell+1\big) = \sum_{j=0}^{\ell} \theta_{m,j} =: \phi(m,\ell)$

Figure 3.8: An illustration of the network architecture implementing the desired function ϕ based on ϕ_1 and ϕ_2 for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$.

Hence, the desired function ϕ can be implemented by the network shown in Figure 3.8. By Theorem 3.1, $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2NL-1]) \subseteq \mathcal{NN}(\text{width} \le 4N+2;\ \text{depth} \le L+1)$. Recall that $\phi_2 \in \mathcal{NN}(\text{width} \le 7;\ \text{depth} \le 2L+1)$. Therefore, the network in Figure 3.8 is with width $\max\{(4N+2)+1, 7\} = 4N+3$ and depth $(L+1)+1+(2L+1) = 3L+3$. So we finish the proof. □

3.3 Approximation in the trifling region

As mentioned earlier in Section 2.3, we need to modify a ReLU network to let it approximate the target function f uniformly well on the whole region $[0,1]^d$, if this ReLU network approximates f well outside the trifling region $\Omega([0,1]^d, K, \delta)$ defined in Equation (2.1).

Theorem 3.7. *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 $\widetilde{\phi}$ is a function implemented by a ReLU network with width N and depth L. If*

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

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

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

Intuitively speaking, Theorem 3.7 shows that: If a function $\widetilde{\phi}$, implemented by a ReLU network, approximates f well except for the trifling region, then we can modify $\widetilde{\phi}$ to get ϕ, implemented by a new ReLU network with similar width and depth to the old one, to approximate f uniformly well on the whole domain. For example, if $\widetilde{\phi}$ approximates a one-dimensional continuous function f well except for an interval in $\mathbb{R}$ with a small length δ, then $\mathrm{mid}\big(\widetilde{\phi}(x-\delta), \widetilde{\phi}(x), \widetilde{\phi}(x+\delta)\big)$ can approximate f well on the whole domain, where $\mathrm{mid}(\cdot,\cdot,\cdot)$ is a function returning the middle value of three inputs and can be implemented via a ReLU network as shown in Lemma 3.8.

Lemma 3.8. *The middle value function $\mathrm{mid}(x_1, x_2, x_3)$ can be implemented by a ReLU network with width 14 and depth 2.*

Proof. Recall the fact

$$x = \sigma(x) - \sigma(-x) \quad \text{and} \quad |x| = \sigma(x) + \sigma(-x), \quad \text{for any } x \in \mathbb{R}. \tag{3.9}$$

Therefore,

$$\begin{aligned} \max(x,y) &= \frac{x+y+|x-y|}{2} \\ &= \tfrac{1}{2}\sigma(x+y) - \tfrac{1}{2}\sigma(-x-y) + \tfrac{1}{2}\sigma(x-y) + \tfrac{1}{2}\sigma(-x+y), \end{aligned} \tag{3.10}$$

for any $x, y \in \mathbb{R}$. Thus, $\max(x_1, x_2, x_3)$ can be implemented by the network shown in Figure 3.9.

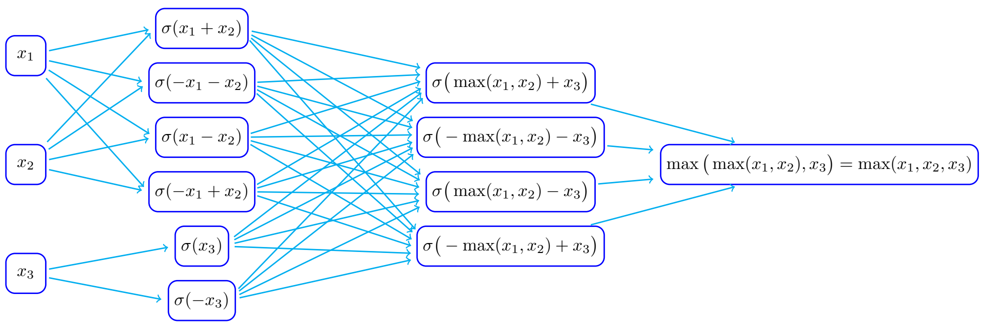

x_1
x_2
x_3
$\sigma(x_1 + x_2)$
$\sigma(-x_1 - x_2)$
$\sigma(x_1 - x_2)$
$\sigma(-x_1 + x_2)$
$\sigma(x_3)$
$\sigma(-x_3)$
$\sigma(\max(x_1, x_2) + x_3)$
$\sigma(-\max(x_1, x_2) - x_3)$
$\sigma(\max(x_1, x_2) - x_3)$
$\sigma(-\max(x_1, x_2) + x_3)$
$\max(\max(x_1, x_2), x_3) = \max(x_1, x_2, x_3)$

Figure 3.9: An illustration of the network architecture implementing $\max(x_1, x_2, x_3)$ based on Equation (3.9) and (3.10).

Clearly,

$$\max(x_1, x_2, x_3) \in \mathcal{NN}(\#\text{input} = 3;\ \text{widthvec} = [6, 4]).$$

Similarly, we have

$$\min(x_1, x_2, x_3) \in \mathcal{NN}(\#\text{input} = 3;\ \text{widthvec} = [6, 4]).$$

It is easy to check that

$$\begin{aligned} &\text{mid}(x_1, x_2, x_3) \\ &= x_1 + x_2 + x_3 - \max(x_1, x_2, x_3) - \min(x_1, x_2, x_3) \\ &= \sigma(x_1 + x_2 + x_3) - \sigma(-x_1 - x_2 - x_3) - \max(x_1, x_2, x_3) - \min(x_1, x_2, x_3). \end{aligned}$$

Hence,

$$\text{mid}(x_1, x_2, x_3) \in \mathcal{NN}(\#\text{input} = 3;\ \text{widthvec} = [14, 10]).$$

That means $\text{mid}(x_1, x_2, x_3)$ can be implemented by a ReLU network with width 14 and depth 2. So we finish the proof. □

The next lemma shows a simple but useful property of the $\mathrm{mid}(x_1, x_2, x_3)$ function that helps to exclude poor approximation in the trifling region.

Lemma 3.9. *For any $\varepsilon > 0$, if at least two of $\{x_1, x_2, x_3\}$ are in $\mathcal{B}(y, \varepsilon)$, then* $\mathrm{mid}(x_1, x_2, x_3) \in \mathcal{B}(y, \varepsilon)$.

Proof. Without loss of generality, we may assume $x_1, x_2 \in \mathcal{B}(y, \varepsilon)$ and $x_1 \le x_2$. Then the proof can be divided into three cases.

- If $x_3 < x_1$, then $\mathrm{mid}(x_1, x_2, x_3) = x_1 \in \mathcal{B}(y, \varepsilon)$.
- If $x_1 \le x_3 \le x_2$, then $\mathrm{mid}(x_1, x_2, x_3) = x_3 \in \mathcal{B}(y, \varepsilon)$ since
$$y - \varepsilon \le x_1 \le x_3 \le x_2 \le y + \varepsilon.$$
- If $x_2 < x_3$, then $\mathrm{mid}(x_1, x_2, x_3) = x_2 \in \mathcal{B}(y, \varepsilon)$.

So we finish the proof. □

Next, given a function g approximating f well on [0,1] except for the trifling region, Lemma 3.10 below shows how to use the $\mathrm{mid}(x_1, x_2, x_3)$ function to construct a new function ϕ uniformly approximating f well on $[0, 1]$, leveraging the useful property of $\mathrm{mid}(x_1, x_2, x_3)$ in Lemma 3.9.

Lemma 3.10. *Given any $\varepsilon > 0$, $K \in \mathbb{N}^+$, and $\delta \in (0, \frac{1}{3K}]$, assume f is a continuous function in $C([0, 1])$ and $g : \mathbb{R} \to \mathbb{R}$ is a general function satisfying*

$$|g(x) - f(x)| \le \varepsilon, \ \text{i.e.,} \ f(x) \in \mathcal{B}\big(g(x), \varepsilon\big), \tag{3.11}$$

for any $x \in [0, 1]\backslash\Omega([0, 1], K, \delta)$. *Then*

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

where

$$\phi(x) := \mathrm{mid}\big(g(x - \delta), g(x), g(x + \delta)\big), \quad \text{for any } x \in \mathbb{R}.$$

Proof. Divide $[0,1]$ into K small intervals denoted by $Q_k = [\frac{k}{K}, \frac{k+1}{K}]$ for $k = 0, 1, \cdots, K-1$. For each k, we further partition Q_k into four small closed intervals as shown in Figure 3.10. To be exact,

$$Q_k = Q_{k,1} \cup Q_{k,2} \cup Q_{k,3} \cup Q_{k,4},$$

where $Q_{k,1} = [\frac{k}{K}, \frac{k}{K} + \delta]$, $Q_{k,2} = [\frac{k}{K} + \delta, \frac{k+1}{K} - 2\delta]$, $Q_{k,3} = [\frac{k+1}{K} - 2\delta, \frac{k+1}{K} - \delta]$, and $Q_{k,4} = [\frac{k+1}{K} - \delta, \frac{k+1}{K}]$.

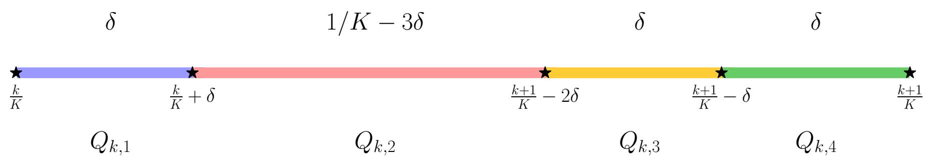


Figure 3.10: An illustration of $Q_{k,i}$ for $i = 1, 2, 3, 4$.

Recall that $\Omega([0,1], K, \delta)$ is the trifling region defined in Equation (2.1). Clearly, $Q_{K-1,4} \subseteq [0,1] \backslash \Omega([0,1], K, \delta)$ and $Q_{k,i} \subseteq [0,1] \backslash \Omega([0,1], K, \delta)$ for $k = 0, 1, \cdots, K-1$ and $i = 1, 2, 3$.

To estimate the difference between $\phi(x)$ and $f(x)$, we consider the following four cases of x in $[0,1]$ for any $k \in \{0, 1, \cdots, K-1\}$.

Case 1: $x \in Q_{k,1}$.

If $x \in Q_{k,1}$, then $x \in [0,1] \backslash \Omega([0,1], K, \delta)$ and

$$x + \delta \in Q_{k,2} \cup Q_{k,3} \subseteq [0,1] \backslash \Omega([0,1], K, \delta).$$

It follows from Equation (3.11) that

$$g(x) \in \mathcal{B}\big(f(x), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big)$$

and

$$g(x + \delta) \in \mathcal{B}\big(f(x + \delta), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

By Lemma 3.9, we get

$$\mathrm{mid}\big(g(x-\delta), g(x), g(x+\delta)\big) \in \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

Case 2: $x \in Q_{k,2}$.

If $x \in Q_{k,2}$, then $x-\delta, x, x+\delta \in [0,1]\backslash\Omega([0,1], K, \delta)$. It follows from Equation (3.11) that

$$g(x-\delta) \in \mathcal{B}\big(f(x-\delta), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big),$$

$$g(x) \in \mathcal{B}\big(f(x), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big),$$

and

$$g(x+\delta) \in \mathcal{B}\big(f(x+\delta), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

Then, by Lemma 3.9, we have

$$\mathrm{mid}\big(g(x-\delta), g(x), g(x+\delta)\big) \in \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

Case 3: $x \in Q_{k,3}$.

If $x \in Q_{k,3}$, then $x \in [0,1]\backslash\Omega([0,1], K, \delta)$ and

$$x-\delta \in Q_{k,1} \cup Q_{k,2} \subseteq [0,1]\backslash\Omega([0,1], K, \delta).$$

It follows from Equation (3.11) that

$$g(x) \in \mathcal{B}\big(f(x), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big)$$

and

$$g(x-\delta) \in \mathcal{B}\big(f(x-\delta), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

By Lemma 3.9, we get

$$\operatorname{mid}\big(g(x-\delta), g(x), g(x+\delta)\big) \in \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

Case 4: $x \in Q_{k,4}$.

If $x \in Q_{k,4}$, we can divide this case into two sub-cases.

- If $k \in \{0, 1, \cdots, K-2\}$, then $x - \delta \in Q_{k,3} \in [0,1]\backslash\Omega([0,1], K, \delta)$ and $x + \delta \in Q_{k+1,1} \subseteq [0,1]\backslash\Omega([0,1], K, \delta)$. It follows from Equation (3.11) that

$$g(x-\delta) \in \mathcal{B}\big(f(x-\delta), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big)$$

  and

$$g(x+\delta) \in \mathcal{B}\big(f(x+\delta), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

  By Lemma 3.9, we get

$$\operatorname{mid}\big(g(x-\delta), g(x), g(x+\delta)\big) \in \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

- If $k = K-1$, then $x \in Q_{k,4} = Q_{K-1,4} \subseteq [0,1]\backslash\Omega([0,1], K, \delta)$ and $x - \delta \in Q_{k,3} \subseteq [0,1]\backslash\Omega([0,1], K, \delta)$. It follows from Equation (3.11) that

$$g(x) \in \mathcal{B}\big(f(x), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big)$$

  and

$$g(x-\delta) \in \mathcal{B}\big(f(x-\delta), \varepsilon\big) \subseteq \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

  By Lemma 3.9, we get

$$\operatorname{mid}\big(g(x-\delta), g(x), g(x+\delta)\big) \in \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big).$$

Since $[0,1] = \cup_{k=0}^{K-1}\Big(\cup_{i=1}^{4} Q_{k,i}\Big)$, we have

$$\mathrm{mid}\big(g(x-\delta), g(x), g(x+\delta)\big) \in \mathcal{B}\big(f(x), \varepsilon + \omega_f(\delta)\big), \quad \text{for any } x \in [0,1].$$

Recall that $\phi(x) = \mathrm{mid}\big(g(x-\delta), g(x), g(x+\delta)\big)$. Then we have

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

So we finish the proof. $\square$

The next lemma below is an analog of Lemma 3.10 for the multidimensional case.

Lemma 3.11. *Given any* $\varepsilon > 0$, $K \in \mathbb{N}^+$, *and* $\delta \in (0, \frac{1}{3K}]$, *assume* f *is a continuous function in* $C([0,1]^d)$ *and* $g : \mathbb{R}^d \to \mathbb{R}$ *is a general function satisfying*

$$|g(\boldsymbol{x}) - f(\boldsymbol{x})| \le \varepsilon, \text{ i.e., } f(\boldsymbol{x}) \in \mathcal{B}\big(g(\boldsymbol{x}), \varepsilon\big), \tag{3.12}$$

for any $\boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta)$. *Then*

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

where $\phi := \phi_d$ *is defined by induction through*

$$\phi_{i+1}(\boldsymbol{x}) := \mathrm{mid}\big(\phi_i(\boldsymbol{x} - \delta \boldsymbol{e}_{i+1}), \phi_i(\boldsymbol{x}), \phi_i(\boldsymbol{x} + \delta \boldsymbol{e}_{i+1})\big), \quad \text{for } i = 0, 1, \cdots, d-1, \tag{3.13}$$

where ϕ_0 *is equal to* g *and* $\{\boldsymbol{e}_i\}_{i=1}^d$ *is the standard basis in* $\mathbb{R}^d$.

Proof. For $\ell = 0, 1, \cdots, d$, we define

$$E_\ell := \left\{ \boldsymbol{x} = (x_1, x_2, \cdots, x_d) : x_i \in \left\{ \begin{smallmatrix} [0,1], & \text{if } i \le \ell, \\ [0,1]\backslash\Omega([0,1],K,\delta), & \text{if } i > \ell \end{smallmatrix} \right. \right\}.$$

Clearly, $E_0 = [0,1]^d \backslash \Omega([0,1]^d, K, \delta)$ and $E_d = [0,1]^d$. See Figure 3.11 for the illustrations of E_ℓ for $\ell = 0, 1, \cdots, d$ when $K = 4$ and $d = 2$.

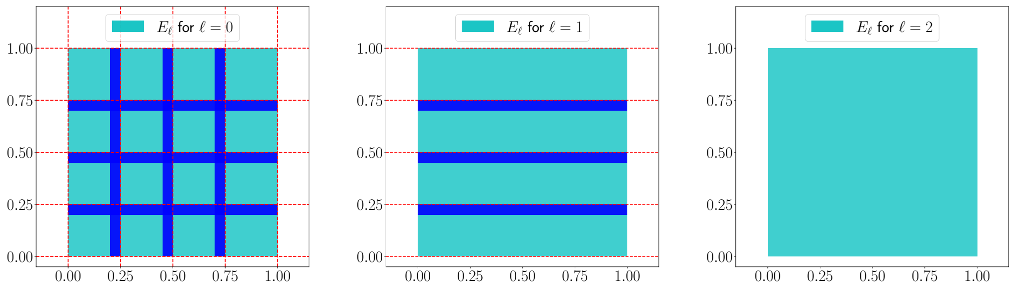


Figure 3.11: Illustrations of E_ℓ for $\ell = 0, 1, 2$ when $K = 4$ and $d = 2$.

We would like to construct a sequence of functions $\phi_0, \phi_1, \cdots, \phi_d$ by induction, based on the iteration equation (3.13), such that, for each $\ell \in \{0, 1, \cdots, d\}$,

$$\phi_\ell(\boldsymbol{x}) \in \mathcal{B}\big(f(\boldsymbol{x}), \varepsilon + \ell \cdot \omega_f(\delta)\big), \quad \text{for any } \boldsymbol{x} \in E_\ell. \tag{3.14}$$

Let us first consider the case $\ell = 0$. Note that $[0,1]^d \backslash \Omega([0,1]^d, K, \delta) = E_0$. Then, by Equation (3.12), we have

$$\phi_0(\boldsymbol{x}) = g(\boldsymbol{x}) \in \mathcal{B}\big(f(\boldsymbol{x}), \varepsilon\big), \quad \text{for any } \boldsymbol{x} \in E_0.$$

That is, Equation (3.14) is true for $\ell = 0$.

Now assume Equation (3.14) is true for $\ell = i \in \{0, 1, \cdots, d-1\}$. We will prove that it also holds for $\ell = i + 1$. By the induction hypothesis, we have

$$\phi_i(x_1, \cdots, x_i, t, x_{i+2}, \cdots, x_d) \in \mathcal{B}\Big(f(x_1, \cdots, x_i, t, x_{i+2}, \cdots, x_d), \varepsilon + i \cdot \omega_f(\delta)\Big), \tag{3.15}$$

for any $x_1, \cdots, x_i \in [0,1]$ and $t, x_{i+2}, \cdots, x_d \in [0,1] \backslash \Omega([0,1], K, \delta)$.

Fix $x_1, \cdots, x_i \in [0,1]$ and $x_{i+2}, \cdots, x_d \in [0,1] \backslash \Omega([0,1], K, \delta)$, and denote

$$\boldsymbol{x}^{[i]} := (x_1, \cdots, x_i,\ x_{i+2}, \cdots, x_d) \in \mathbb{R}^{d-1}.$$

Then define

$$\psi_{\boldsymbol{x}^{[i]}}(t) \coloneqq \phi_i(x_1, \cdots, x_i,\ t,\ x_{i+2}, \cdots, x_d), \qquad \text{for any } t \in \mathbb{R},$$

and

$$f_{\boldsymbol{x}^{[i]}}(t) \coloneqq f(x_1, \cdots, x_i,\ t,\ x_{i+2}, \cdots, x_d), \qquad \text{for any } t \in \mathbb{R}.$$

It follows from Equation (3.15) that

$$\psi_{\boldsymbol{x}^{[i]}}(t) \in \mathcal{B}\big(f_{\boldsymbol{x}^{[i]}}(t), \varepsilon + i \cdot \omega_f(\delta)\big), \quad \text{for any } t \in [0,1]\backslash\Omega([0,1], K, \delta).$$

Then by Lemma 3.10 (set $g = \psi_{\boldsymbol{x}^{[i]}}$ and $f = f_{\boldsymbol{x}^{[i]}}$ therein), we get, for any $t \in [0,1]$,

$$\begin{aligned}\text{mid}\Big(\psi_{\boldsymbol{x}^{[i]}}(t-\delta),\ \psi_{\boldsymbol{x}^{[i]}}(t),\ \psi_{\boldsymbol{x}^{[i]}}(t+\delta)\Big) &\in \mathcal{B}\Big(f_{\boldsymbol{x}^{[i]}}(t),\ \ \varepsilon + i \cdot \omega_f(\delta) + \omega_{f_{\boldsymbol{x}^{[i]}}}(\delta)\Big)\\ &\subseteq \mathcal{B}\Big(f_{\boldsymbol{x}^{[i]}}(t),\ \ \varepsilon + (i+1)\omega_f(\delta)\Big).\end{aligned}$$

That is, for any $x_{i+1} = t \in [0,1]$,

$$\begin{aligned}&\text{mid}\Big(\phi_i(x_1, \cdots, x_i,\ x_{i+1}-\delta,\ x_{i+2}, \cdots, x_d), \phi_i(x_1, \cdots, x_d), \phi_i(x_1, \cdots, x_i,\ x_{i+1}+\delta,\ x_{i+2}, \cdots, x_d)\Big)\\ &\in \mathcal{B}\Big(f(x_1, \cdots, x_d), \varepsilon + (i+1)\omega_f(\delta)\Big).\end{aligned}$$

Note that $x_1, \cdots, x_i \in [0,1]$ and $x_{i+2}, \cdots, x_d \in [0,1]\backslash\Omega([0,1], K, \delta)$ are arbitrary. Thus, for any $\boldsymbol{x} \in E_{i+1}$,

$$\text{mid}\big(\phi_i(\boldsymbol{x} - \delta\boldsymbol{e}_{i+1}), \phi_i(\boldsymbol{x}), \phi_i(\boldsymbol{x} + \delta\boldsymbol{e}_{i+1})\big) \in \mathcal{B}\big(f(\boldsymbol{x}), \varepsilon + (i+1)\omega_f(\delta)\big),$$

implying

$$\phi_{i+1}(\boldsymbol{x}) \in \mathcal{B}\big(f(\boldsymbol{x}), \varepsilon + (i+1)\omega_f(\delta)\big), \quad \text{for any } \boldsymbol{x} \in E_{i+1}.$$

So Equation (3.14) holds for $\ell = i+1$, which means we finish the process of mathematical induction.

By the principle of induction, we have

$$\phi(\boldsymbol{x}) := \phi_d(\boldsymbol{x}) \in \mathcal{B}\big(f(\boldsymbol{x}), \varepsilon + d\cdot\omega_f(\delta)\big), \quad \text{for any } \boldsymbol{x} \in E_d = [0,1]^d.$$

Therefore,

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

which means we finish the proof. □

Now we are ready to prove Theorem 3.7.

Proof of Theorem 3.7. Set $\phi_0 = \widetilde{\phi}$ and define ϕ_i for $i \in \{1, 2, \cdots, d\}$ by induction through

$$\phi_{i+1}(\boldsymbol{x}) := \mathrm{mid}\big(\phi_i(\boldsymbol{x} - \delta\boldsymbol{e}_{i+1}), \phi_i(\boldsymbol{x}), \phi_i(\boldsymbol{x} + \delta\boldsymbol{e}_{i+1})\big), \quad \text{for } i = 0, 1, \cdots, d-1,$$

where $\{\boldsymbol{e}_i\}_{i=1}^d$ is the standard basis in $\mathbb{R}^d$. Then by Lemma 3.11 with $\phi = \phi_d$, we have

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

It remains to determine the network architecture implementing $\phi = \phi_d$.

Define a vector-valued function $\boldsymbol{\Phi}_0 : \mathbb{R}^d \to \mathbb{R}^3$ as

$$\boldsymbol{\Phi}_0(\boldsymbol{x}) := \big(\phi_0(\boldsymbol{x} - \delta\boldsymbol{e}_1),\ \phi_0(\boldsymbol{x}),\ \phi_0(\boldsymbol{x} + \delta\boldsymbol{e}_1)\big), \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^d.$$

Note that $\phi_0 = \widetilde{\phi} \in \mathcal{NN}(\text{width} \le N;\ \text{depth} \le L)$. Hence, $\phi_0(\cdot - \delta\boldsymbol{e}_1)$, $\phi_0(\cdot)$, and $\phi_0(\cdot + \delta\boldsymbol{e}_1)$ can be generated by three networks with the same number of hidden layers, and these three networks are all with width N and depth L. Therefore, by putting these three networks in parallel (share the inputs), we have $\boldsymbol{\Phi}_0 \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \le 3N;\ \text{depth} \le L;\ \#\text{output} = 3)$.

Recall that $\mathrm{mid}(\cdot,\cdot,\cdot) \in \mathcal{NN}(\text{width} \le 14;\ \text{depth} \le 2)$ by Lemma 3.8. Therefore, by Lemma 2.1 (ii),

$$\phi_1 = \min(\cdot,\cdot,\cdot)\circ \boldsymbol{\Phi}_0 \in \mathcal{NN}\Big(\text{width} \le \max\{3N, 14\} \le 3(N+4);\ \text{depth} \le L+2\Big)$$

Similarly, by the iterative formula

$$\phi_{i+1}(\boldsymbol{x}) \coloneqq \text{mid}\big(\phi_i(\boldsymbol{x}-\delta\boldsymbol{e}_{i+1}), \phi_i(\boldsymbol{x}), \phi_i(\boldsymbol{x}+\delta\boldsymbol{e}_{i+1})\big), \quad \text{for } i = 0, 1, \cdots, d-1,$$

it is easy to verify

$$\phi = \phi_d \in \mathcal{NN}\Big(\text{width} \le 3^d(N+4);\ \text{depth} \le L+2d\Big).$$

So we finish the proof. □

3.4 Approximation of step functions

As mentioned earlier in Section 2.3, we need to construct a ReLU sub-network to project a cube to a point. We only need to approximate one-dimensional step functions, because in the multidimensional case we can simply set

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \big(\phi(x_1), \phi(x_2), \cdots, \phi(x_d)\big), \quad \text{for any } \boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d,$$

where ϕ is a one-dimensional step function. The theorem below, Theorem 3.12, shows that ReLU networks with width $\mathcal{O}(N^{1/d})$ and depth $\mathcal{O}(L)$ can implement one-dimensional step functions with $\mathcal{O}(K) = \mathcal{O}(N^{2/d}L^{2/d})$ "steps" outside the trifling region for any $d \in \mathbb{N}^+$. See Figure 3.12 for an example.

Theorem 3.12. *For any $N, L, d \in \mathbb{N}^+$ and $\delta \in (0, \frac{1}{3K}]$ with $K = \lfloor N^{1/d}\rfloor^2 \lfloor L^{2/d}\rfloor$, there exists a one-dimensional function ϕ implemented by a ReLU network with*

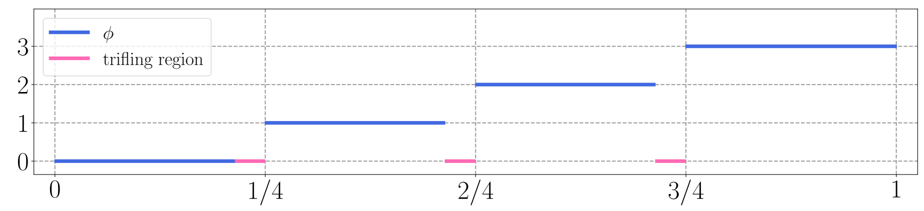


Figure 3.12: An example of a step function for the case $K=4$ and $d=1$. We do not need to care about the values of ϕ in the trifling region while constructing a ReLU network to implement ϕ.

width $4\lfloor N^{1/d}\rfloor+3$ *and depth* $4L+5$ *such that*

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

The setting $K=\lfloor N^{1/d}\rfloor^2\lfloor L^{2/d}\rfloor=\mathcal{O}(N^{2/d}L^{2/d})$ is not neat here, but it is very convenient for later use. Now, let us present the detailed proof of Theorem 3.12.

Proof of Theorem 3.12. We divide the proof into two cases: $d=1$ and $d\ge 2$.

Case $1:\ d=1$.

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

$$\big\{(1,M-1),(2,0)\big\}\cup\big\{(\tfrac{m}{M},m):m=0,1,\cdots,M-1\big\}$$
$$\cup\big\{(\tfrac{m+1}{M}-\delta,m):m=0,1,\cdots,M-2\big\}.$$

Its size is $2M+1=N\cdot\big((2NL-1)+1\big)+1$. By Theorem 3.2 (set $m=N$ and $n=2NL-1$ therein), there exists

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

such that

- $\phi_1(\frac{M-1}{M})=\phi_1(1)=M-1$ and $\phi_1(\frac{m}{M})=\phi_1(\frac{m+1}{M}-\delta)=m$ for $m=0,1,\cdots,M-$

2.

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

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

$$\phi_1(x) = m, \quad \text{for any } x \in [\tfrac{m}{M}, \tfrac{m+1}{M} - \delta \cdot 1_{\{m \le M-2\}}]. \tag{3.16}$$

Now consider the another sample set

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

Its size is $2L+1 = 1 \cdot \big((2L-1)+1\big)+1$. By Theorem 3.2 (set $m = 1$ and $n = 2L-1$ therein), there exists

$$\begin{aligned}\phi_2 &\in \mathcal{NN}(\text{widthvec} = [2, 2(2L-1)+1]) \\ &= \mathcal{NN}(\text{widthvec} = [2, 4L-1])\end{aligned}$$

such that

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

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

$$\phi_2(x - \tfrac{m}{M}) = \ell, \quad \text{for any } x \in [\tfrac{mL+\ell}{ML}, \tfrac{mL+\ell+1}{ML} - \delta \cdot 1_{\{\ell \le L-2\}}]. \tag{3.17}$$

$K = ML$ implies any $k \in \{0, 1, \cdots, K-1\}$ can be unique represented by $k = mL + \ell$ for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$. Then the desired function ϕ can be implemented by a ReLU network shown in Figure 3.13.

Clearly,

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

$$\begin{tikzcd}
 & \boxed{\phi_1(x)=m} \arrow[r] \arrow[ddr] & \boxed{m} \arrow[r] & \boxed{m} \arrow[dr] & \\
\boxed{x} \arrow[ur, "\phi_1"] \arrow[dr] & & & & \boxed{mL+\ell=k=:\phi(x)} \\
 & \boxed{x} \arrow[r] & \boxed{x-\frac{m}{M}} \arrow[r, "\phi_2"] & \boxed{\phi_2(x-\frac{m}{M})=\ell} \arrow[ur] &
\end{tikzcd}$$

Figure 3.13: An illustration of the network architecture implementing ϕ based on Equation (3.16) and (3.17) for $x \in [\frac{k}{K}, \frac{k}{K} - \delta \cdot 1_{\{k \le K-2\}}] = [\frac{mL+\ell}{ML}, \frac{mL+\ell+1}{ML} - \delta \cdot 1_{\{m \le M-2 \text{ or } \ell \le L-2\}}]$, where $k = mL+\ell$ for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$.

By Theorem 3.1, we have

$$\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 4NL-1]) \subseteq \mathcal{NN}(\text{width} \le 4N+2;\ \text{depth} \le 2L+1)$$

and

$$\phi_2 \in \mathcal{NN}(\text{widthvec} = [2, 4L-1]) \subseteq \mathcal{NN}(\text{width} \le 6;\ \text{depth} \le 2L+1),$$

implying by Figure 3.13 that ϕ can be implemented by a ReLU network with width

$$\max\{4N+2+1, 6+1\} = 4N+3$$

and depth

$$(2L+1) + 2 + (2L+1) + 1 = 4L+5.$$

So we finish the proof for the case $d = 1$.

Case 2: $d \ge 2$.

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

$$\big\{(1, K-1), (2, 0)\big\} \cup \big\{(\tfrac{k}{K}, k) : k = 0, 1, \cdots, K-1\big\} \\ \cup \big\{(\tfrac{k+1}{K} - \delta, k) : k = 0, 1, \cdots, K-2\big\}.$$

Its size is

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

By Theorem 3.2 (set $m = \lfloor N^{1/d} \rfloor$ and $n = 2\lfloor N^{1/d} \rfloor \lfloor L^{2/d} \rfloor - 1$ therein), there exists

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

such that

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

Then, for $k = 0, 1, \cdots, K - 1$, we have

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

By Theorem 3.1,

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

Thus, we finish the proof for the case $d \ge 2$. $\square$

This page is intentionally left blank.

Chapter 4

Approximation by ReLU networks

4.1 Approximation of polynomials

In this section, we show how to construct a ReLU network to approximate a multidimensional polynomial $P(\boldsymbol{x})$.

4.1.1 Main theorem

For simplicity, we may assume a polynomial $P(\boldsymbol{x})$ has only one term with coefficient one, namely, $P(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$ for some $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_d) \in \mathbb{N}^d$. As shown in the following theorem, Theorem 4.1, ReLU networks can uniformly approximate polynomials on $[0,1]^d$ with exponential errors.

Theorem 4.1. *Given any $k \in \mathbb{N}^+$ and $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_d) \in \mathbb{N}^d$, assume $P(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$ with $\|\boldsymbol{\alpha}\|_1 \le k$. For any $N, L \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU network with width $9(N+1)+k-1$ and depth $7k^2L$ such that*

$$|\phi(\boldsymbol{x}) - P(\boldsymbol{x})| \le 9k(N+1)^{-7kL}, \quad \text{for any } \boldsymbol{x} \in [0,1]^d.$$

The choice of depth $7k^2L$ is not neat here, but it is convenient for later use. Theorem 4.1 shows that ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ are able to approximate polynomials on $[0,1]^d$ within an error $\mathcal{O}(N^{-L})$. This reveals the power of depth in ReLU networks for approximating polynomials, from function compositions. Theorem 4.1 can be generalized to the case of polynomials defined on an arbitrary hypercube $[a,b]^d$ by scaling. To prove Theorem 4.1, we will construct ReLU networks to approximate polynomials following the four steps below.

- $f(x) = x^2$. We approximate $f(x) = x^2$ by the combinations and compositions of "sawtooth" functions as shown in Figure 4.1 and 4.2.
- $f(x,y) = xy$. To approximate $f(x,y) = xy$, we use the result of the previous step and the fact $xy = 2\big((\frac{x+y}{2})^2 - (\frac{x}{2})^2 - (\frac{y}{2})^2\big)$.
- $f(x_1, x_2, \cdots, x_k) = x_1 x_2 \cdots x_k$. We approximate $f(x_1, x_2, \cdots, x_k) = x_1 x_2 \cdots x_k$ via mathematical induction based on the result of the previous step.
- A general polynomial $P(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$ with $\|\boldsymbol{\alpha}\|_1 \le k \in \mathbb{N}^+$. $P(\boldsymbol{x})$ can be written as $P(\boldsymbol{x}) = z_1 z_2 \cdots z_k$, where $\boldsymbol{z} = (z_1, z_2, \cdots, z_k)$ is a vector with each of its entries equal to 1 or an entry of $\boldsymbol{x}$. Then use the result of the previous step.

4.1.2 Approximation of x^2

Let us show how to approximate $f(x) = x^2$ by linear combinations of "sawtooth" functions, which can be easily implemented by ReLU networks. The idea of using "sawtooth" functions (see Figure 4.1) was first raised in [58] for approximating x^2 using networks with depth $\mathcal{O}(L)$ and a constant width, and achieving an error $\mathcal{O}(2^{-L})$. Our construction is different to and more general than that in [58], working for ReLU networks of width $3N$ and depth L for any $N, L \in \mathbb{N}^+$, and achieving an error N^{-L} as shown in Lemma 4.2.

Lemma 4.2. *For any $N, L \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU network with width $3N$ and depth L such that*

$$0 \le \phi(x) - x^2 \le N^{-L}, \quad \text{for any } x \in [0,1].$$

Proof. Define a set of "sawtooth" functions $T_i : [0,1] \to [0,1]$ by induction as follows. Let

$$T_1(x) = \begin{cases} 2x, & \text{if } x \in [0, \frac{1}{2}], \\ 2(1-x), & \text{if } x \in (\frac{1}{2}, 1] \end{cases}$$

and

$$T_i = T_{i-1} \circ T_1, \quad \text{for } i = 2, 3, 4, \cdots.$$

It is easy to check that T_i has 2^{i-1} "sawteeth" and

$$T_{m+n} = T_m \circ T_n, \quad \text{for any } m, n \in \mathbb{N}^+.$$

See Figure 4.1 for illustrations of T_i for $i = 1, 2, 3, 4$.

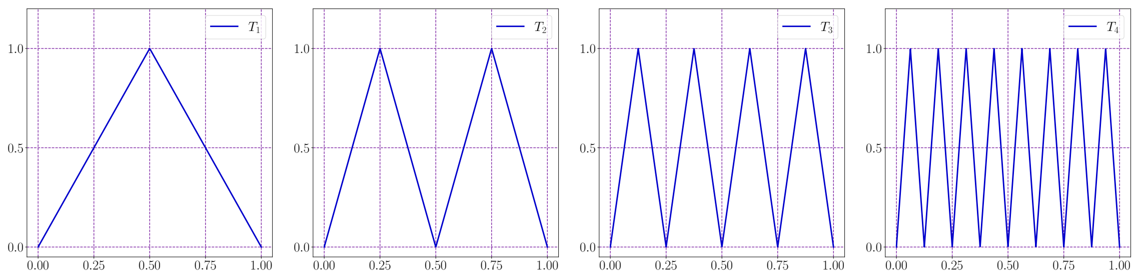


Figure 4.1: Examples of "sawtooth" functions T_1, T_2, T_3, and T_4.

Define piecewise linear functions $f_s : [0,1] \to [0,1]$ for $s \in \mathbb{N}^+$ satisfying the following two requirements (see Figure 4.2 for several examples of f_s).

- $f_s(x) = x^2$ for any $x \in \{\frac{j}{2^s} : j = 0, 1, 2, \cdots, 2^s\}$.
- $f_s(x)$ is linear between any two adjacent points of $\{\frac{j}{2^s} : j = 0, 1, 2, \cdots, 2^s\}$.

Recall the fact

$$0 \le tx_1^2 + (1-t)x_2^2 - \Big(tx_1 + (1-t)x_2\Big)^2 \le \frac{(x_2 - x_1)^2}{4}, \quad \text{for any } t, x_1, x_2 \in [0,1].$$

Thus, we have

$$0 \le f_s(x) - x^2 \le \frac{(2^{-s})^2}{4} = 2^{-2(s+1)}, \quad \text{for any } x \in [0,1] \text{ and } s \in \mathbb{N}^+. \tag{4.1}$$

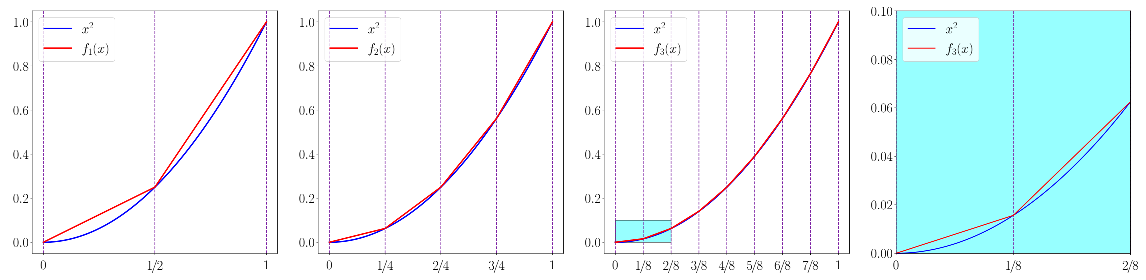


Figure 4.2: Illustrations of f_1, f_2, and f_3 for approximating x^2.

Note that $f_{i-1}(x) = f_i(x) = x^2$ for each $x \in \{\frac{j}{2^{i-1}} : j = 0, 1, 2, \cdots, 2^{i-1}\}$ and the graph of $f_{i-1} - f_i$ is a symmetric "sawtooth" between any two adjacent points of $\{\frac{j}{2^{i-1}} : j = 0, 1, 2, \cdots, 2^{i-1}\}$. See Figure 4.3 for illustrations.

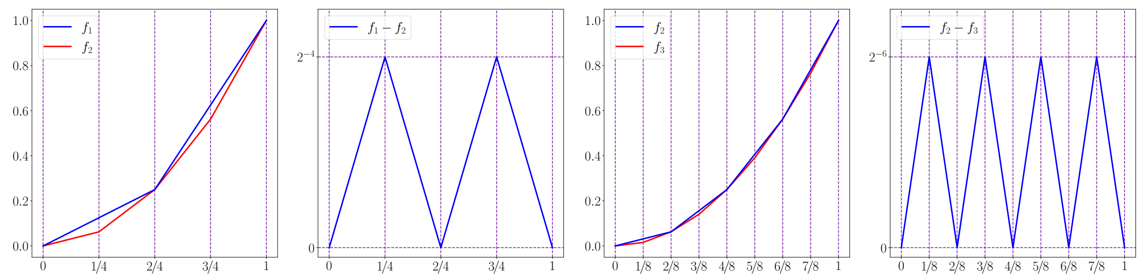


Figure 4.3: Illustrations of $f_1 - f_2$ and $f_2 - f_3$.

Thus, it is easy to verify

$$f_{i-1}(x) - f_i(x) = \frac{T_i(x)}{2^{2i}}, \quad \text{for any } x \in [0,1] \text{ and } i = 2, 3, 4, \cdots.$$

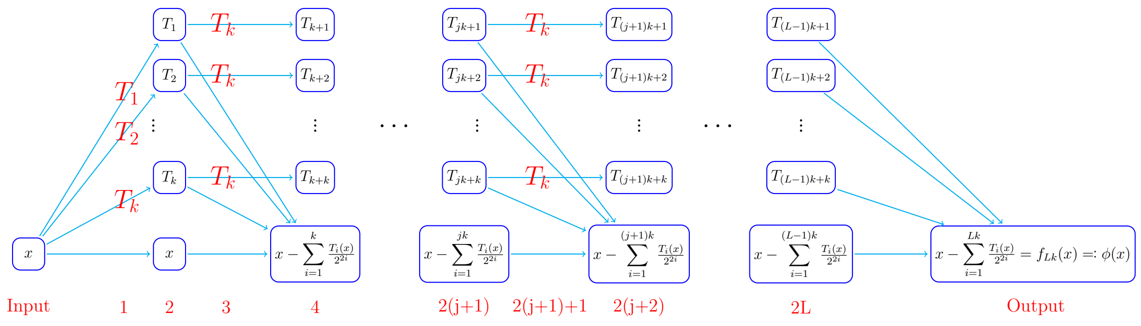
Therefore, for any $x \in [0,1]$ and $s \in \mathbb{N}^+$, we have

$$f_s(x) = f_1(x) + \sum_{i=2}^{s}(f_i - f_{i-1}) = x - (x - f_1(x)) - \sum_{i=2}^{s} \frac{T_i(x)}{2^{2i}} = x - \sum_{i=1}^{s} \frac{T_i(x)}{2^{2i}}.$$

Given any $N \in \mathbb{N}^+$, there exists a unique $k \in \mathbb{N}^+$ such that

$$(k-1)2^{k-1} + 1 \le N \le k2^k.$$

For this k, we can construct a ReLU network as shown in Figure 4.4 to implement a function $\phi = f_{Lk}$ approximating x^2 well. Note that T_i has 2^{i-1} "sawteeth" and hence has $2^i + 1$ breakpoints including the endpoints for any $i \in \mathbb{N}^+$. Then, by Lemma 3.3, T_i can be implemented by a one-hidden-layer ReLU network with width 2^i. Therefore, the network in Figure 4.4 has width $k2^k + 1 \le 3N$ ① and depth $2L$.



T_1
T_2
T_k
x
T_{k+1}
T_{k+2}
T_{k+k}
$x - \sum_{i=1}^{k} \frac{T_i(x)}{2^{2i}}$
T_{jk+1}
T_{jk+2}
T_{jk+k}
$x - \sum_{i=1}^{jk} \frac{T_i(x)}{2^{2i}}$
$T_{(j+1)k+1}$
$T_{(j+1)k+2}$
$T_{(j+1)k+k}$
$x - \sum_{i=1}^{(j+1)k} \frac{T_i(x)}{2^{2i}}$
$T_{(L-1)k+1}$
$T_{(L-1)k+2}$
$T_{(L-1)k+k}$
$x - \sum_{i=1}^{(L-1)k} \frac{T_i(x)}{2^{2i}}$
$x - \sum_{i=1}^{Lk} \frac{T_i(x)}{2^{2i}} = f_{Lk}(x) =: \phi(x)$
Input
1
2
3
4
2(j+1)
2(j+1)+1
2(j+2)
2L
Output

Figure 4.4: An illustration of the target network architecture for approximating x^2 on $x \in [0, 1]$. T_i can be implemented by a one-hidden-layer ReLU network with width 2^i for $i = 1, 2, \cdots, k$. The red numbers below the architecture indicate the order of hidden layers.

As shown in Figure 4.4, all neurons in (2ℓ)-th hidden layer of the network have the identify function as their activation functions for $\ell = 1, 2, \cdots, L$. Thus, the network in Figure 4.4 can be interpreted as a ReLU network with width $3N$ and depth L. In fact, if all activation functions in a certain hidden layer are identity maps, the depth can be reduced by one via combining adjacent two affine linear transforms into one, the idea of which is similar to that of Lemma 2.1. For example, suppose $\boldsymbol{W}_1 \in \mathbb{R}^{N_2 \times N_1}$, $\boldsymbol{b}_1 \in \mathbb{R}^{N_2}$, $\boldsymbol{W}_2 \in \mathbb{R}^{N_3 \times N_2}$, $\boldsymbol{b}_2 \in \mathbb{R}^{N_3}$, ϱ is an identity map that can be applied to vectors or matrices elementwisely, and $\mathcal{L}_i$ is an affine linear map given by $\mathcal{L}_i(\boldsymbol{x}) = \boldsymbol{W}_i \cdot \boldsymbol{x} + \boldsymbol{b}_i$ for $i = 1, 2$. Then, we have

$$\mathcal{L}_2 \circ \varrho \circ \mathcal{L}_1(\boldsymbol{x}) = \boldsymbol{W}_2 \varrho(\boldsymbol{W}_1 \cdot \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2 = \boldsymbol{W}_3 \cdot \boldsymbol{x} + \boldsymbol{b}_3, \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^{N_1},$$

where $\boldsymbol{W}_3 = \boldsymbol{W}_2 \cdot \boldsymbol{W}_1 \in \mathbb{R}^{N_3 \times N_1}$, $\boldsymbol{b}_3 = \boldsymbol{W}_2 \cdot \boldsymbol{b}_1 + \boldsymbol{b}_2 \in \mathbb{R}^{N_3}$.

①This inequality is clear for $k = 1, 2, 3, 4$. In the case $k \ge 5$, we have $k2^k + 1 \le \frac{k2^k+1}{N} N \le \frac{(k+1)2^k}{(k-1)2^{k-1}} N \le 2\frac{k+1}{k-1} N \le 3N$.

It remains to estimate the approximation error of $\phi(x) \approx x^2$. By Equation (4.1), for any $x \in [0,1]$, we have

$$0 \le \phi(x) - x^2 = f_{Lk}(x) - x^2 \le 2^{-2(Lk+1)} \le \left(2^{2k}\right)^{-L} \le N^{-L},$$

where the last inequality comes from $N \le k2^k \le 2^{2k}$. So we finish the proof. □

4.1.3 Approximation of $x_1x_2\cdots x_k$

We have constructed a ReLU network to approximate $f(x) = x^2$. By the fact $xy = 2\big((\frac{x+y}{2})^2 - (\frac{x}{2})^2 - (\frac{y}{2})^2\big)$, it is easy to construct a new ReLU network to approximate $f(x,y) = xy$ as follows.

Lemma 4.3. *For any $N, L \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU network with width $9N$ and depth L such that*

$$|\phi(x,y) - xy| \le 6N^{-L}, \quad \text{for any } x, y \in [0,1].$$

Proof. By Lemma 4.2, there exists a function ψ implemented by a ReLU network with width $3N$ and depth L such that

$$|\psi(x) - x^2| \le N^{-L}, \quad \text{for any } x \in [0,1].$$

Recall the fact

$$xy = 2\big((\tfrac{x+y}{2})^2 - (\tfrac{x}{2})^2 - (\tfrac{y}{2})^2\big), \quad \text{for any } x, y \in \mathbb{R}.$$

The target function ϕ is defined as

$$\phi(x,y) := 2\big(\psi(\tfrac{x+y}{2}) - \psi(\tfrac{x}{2}) - \psi(\tfrac{y}{2})\big), \quad \text{for any } x, y \in \mathbb{R}. \tag{4.2}$$

Then ϕ can be implemented by the network architecture in Figure 4.5.

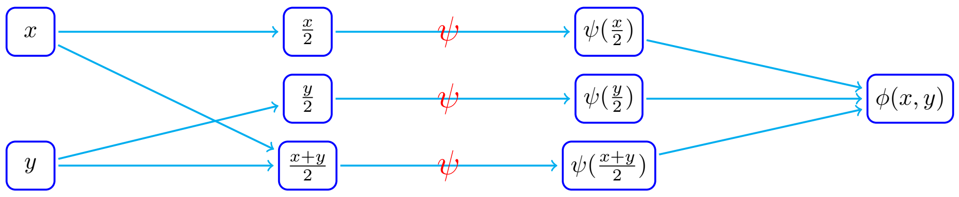


Figure 4.5: An illustration of the network architecture implementing ϕ for approximating xy on $[0,1]^2$.

It follows from $\psi \in \mathcal{NN}(\text{width} \le 3N;\ \text{depth} \le L)$ that the network in Figure 4.5 is with width $9N$ and depth $L+2$. Similar to the discussion in the proof of Lemma 4.2, the network in Figure 4.5 can be interpreted as a ReLU network with width $9N$ and depth L, since two of hidden layers have the identify map as their activation functions. Moreover, for any $x, y \in [0,1]$,

$$\begin{aligned}|\phi(x,y) - xy| &= \left|2\big(\psi(\tfrac{x+y}{2}) - \psi(\tfrac{x}{2}) - \psi(\tfrac{y}{2})\big) - 2\big((\tfrac{x+y}{2})^2 - (\tfrac{x}{2})^2 - (\tfrac{y}{2})^2\big)\right| \\ &\le 2\left|\psi(\tfrac{x+y}{2}) - (\tfrac{x+y}{2})^2\right| + 2\left|\psi(\tfrac{x}{2}) - (\tfrac{x}{2})^2\right| + 2\left|\psi(\tfrac{y}{2}) - (\tfrac{y}{2})^2\right| \le 6N^{-L}.\end{aligned}$$

Therefore, we have finished the proof. □

We would like to remark that we can also use Lemma 2.1 to verify the function ϕ defined in Equation (4.2) can be implemented by a ReLU network with width $9N$ and depth L, since $\psi \in \mathcal{NN}(\text{width} \le 3N;\ \text{depth} \le L)$.

Now let us show how to construct a ReLU network to approximate $f(x,y) = xy$ on $[a,b]^2$ with arbitrary $a < b$, *i.e.*, a rescaled version of Lemma 4.3.

Lemma 4.4. *For any $N, L \in \mathbb{N}^+$ and $a, b \in \mathbb{R}$ with $a < b$, there exists a function ϕ implemented by a ReLU network with width $9N+1$ and depth L such that*

$$|\phi(x,y) - xy| \le 6(b-a)^2 N^{-L}, \quad \text{for any } x, y \in [a,b].$$

Proof. By Lemma 4.3, there exists a function ψ implemented by a ReLU network with width $9N$ and depth L such that

$$|\psi(\widetilde{x}, \widetilde{y}) - \widetilde{x}\widetilde{y}| \le 6N^{-L}, \quad \text{for any } \widetilde{x}, \widetilde{y} \in [0,1].$$

Given any $x, y \in [a,b]$, by setting $\widetilde{x} = \frac{x-a}{b-a}$ and $\widetilde{y} = \frac{y-a}{b-a}$, we have $\widetilde{x}, \widetilde{y} \in [0,1]$, implying

$$\left|\psi(\tfrac{x-a}{b-a}, \tfrac{y-a}{b-a}) - \tfrac{x-a}{b-a}\tfrac{y-a}{b-a}\right| \le 6N^{-L}, \quad \text{for any } x, y \in [a,b].$$

It follows that, for any $x, y \in [a,b]$,

$$\left|(b-a)^2\psi(\tfrac{x-a}{b-a}, \tfrac{y-a}{b-a}) + a(x+y) - a^2 - xy\right| \le 6(b-a)^2N^{-L}. \tag{4.3}$$

Define, for any $x, y \in \mathbb{R}$,

$$\phi(x,y) \coloneqq (b-a)^2\psi(\tfrac{x-a}{b-a}, \tfrac{y-a}{b-a}) + a\cdot\sigma(x+y+2|a|) - a^2 - 2a|a|.$$

Then ϕ can be implemented by the network architecture in Figure 4.6.

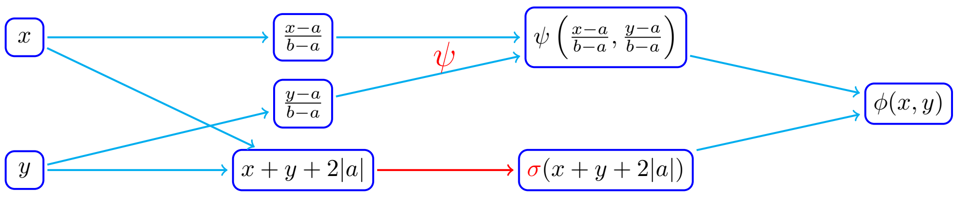


Figure 4.6: An illustration of the network architecture implementing ϕ for approximating xy on $[a,b]^2$. Two of hidden layers has the identify function as their activation functions, since the red "σ" comes from the red arrow "$\longrightarrow$", where the red arrow "$\longrightarrow$" represents a ReLU network with width 1 and depth $L \ge 1$.

If follows from $\psi \in \mathcal{NN}(\text{width} \le 9N;\ \text{depth} \le L)$ that the network in Figure 4.6 is with width $9N+1$ and depth $L+2$. Similar to the discussion in the proof of Lemma 4.2, the network in Figure 4.6 can be interpreted as a ReLU network with width $9N+1$ and depth L, since two of hidden layers have the identify function as their activation functions.

Note that $x+y+2|a| \ge 0$ for any $x, y \in [a,b]$, implying

$$\phi(x,y) = (b-a)^2\psi(\tfrac{x-a}{b-a}, \tfrac{y-a}{b-a}) + a(x+y) - a^2, \quad \text{for any } x, y \in [a,b].$$

Hence, by Equation (4.3), we have

$$\big|\phi(x,y) - xy\big| \le 6(b-a)^2 N^{-L}, \quad \text{for any } x, y \in [a,b].$$

So we finish the proof. □

The next lemma constructs a ReLU network to approximate a multivariable function $f(x_1, x_2, \cdots, x_k) = x_1 x_2 \cdots x_k$ on $[0,1]^k$.

Lemma 4.5. *For any $N, L, k \in \mathbb{N}^+$ with $k \ge 2$, there exists a function ϕ implemented by a ReLU network with width $9(N+1)+k-1$ and depth $7kL(k-1)$ such that*

$$|\phi(\boldsymbol{x}) - x_1 x_2 \cdots x_k| \le 9(k-1)(N+1)^{-7kL},$$

for any $\boldsymbol{x} = (x_1, x_2, \cdots, x_k) \in [0,1]^k$.

Proof. By Lemma 4.4, there exists a function ϕ_1 implemented by a ReLU network with width $9(N+1)+1$ and depth $7kL$ such that

$$|\phi_1(x,y) - xy| \le 6(1.2)^2(N+1)^{-7kL} \le 9(N+1)^{-7kL}, \quad \text{for any } x, y \in [-0.1, 1.1]. \quad (4.4)$$

This equation means the case $k = 2$ is clear. We may assume $k \ge 3$ below. We would like to construct a sequence of functions $\phi_i : [0,1]^{i+1} \to [0,1]$ for any $i \in \{1, 2, \cdots, k-1\}$ by induction such that

(i) $\phi_i \in \mathcal{NN}(\text{width} \le 9(N+1)+i;\ \text{depth} \le 7kLi)$ for any $i \in \{1, 2, \cdots, k-1\}$.

(ii) For any $i \in \{1, 2, \cdots, k-1\}$ and $x_1, x_2, \cdots, x_{i+1} \in [0,1]$, it holds that

$$|\phi_i(x_1, x_2, \cdots, x_{i+1}) - x_1 x_2 \cdots x_{i+1}| \le 9i(N+1)^{-7kL}. \quad (4.5)$$

First, let us consider the case $i = 1$. It is obvious that the two required conditions are true: 1) $9(N+1)+i = 9(N+1)+1$ and $7kLi = 7kL$ in the case $i = 1$; 2) Equation (4.4) implies Equation (4.5) for $i = 1$.

Now assume ϕ_i has been defined for some $i \in \{1, 2, \cdots, k-2\}$, we define

$$\phi_{i+1}(x_1, x_2, \cdots, x_{i+2}) \coloneqq \phi_1\big(\phi_i(x_1, x_2, \cdots, x_{i+1}), \sigma(x_{i+2})\big), \tag{4.6}$$

for any $x_1, x_2, \cdots, x_{i+2} \in \mathbb{R}$. By the induction hypothesis, we have

$$\phi_i \in \mathcal{NN}(\text{width} \le 9(N+1) + i;\ \text{depth} \le 7kLi)$$

and

$$|\phi_i(x_1, x_2, \cdots, x_{i+1}) - x_1 x_2 \cdots x_{i+1}| \le 9i(N+1)^{-7kL}, \tag{4.7}$$

for any $x_1, x_2, \cdots, x_{i+1} \in [0,1]$. Clearly, $\phi_1 \in \mathcal{NN}(\text{width} \le 9(N+1) + 1;\ \text{depth} \le 7kL)$. Then ϕ_{i+1}, defined in Equation (4.6), can be implemented via a ReLU network with width

$$\max\{9(N+1) + i + 1, 9(N+1) + 1\} = 9(N+1) + (i+1)$$

and depth $7kLi + 7kL = 7kL(i+1)$.

Note that $9i(N+1)^{-7kL} \le 9k2^{-7k} \le 0.1$ for any $N, L, k \in \mathbb{N}^+$ and $i \in \{1, 2, \cdots, k\}$. It follows from Equation (4.7) that

$$\phi_i(x_1, x_2, \cdots, x_{i+1}) \in [-0.1, 1.1], \quad \text{for any } x_1, x_2, \cdots, x_{i+1} \in [0,1].$$

Therefore, by Equation (4.4) and (4.7), we have

$$\begin{aligned}
&\big|\phi_{i+1}(x_1, \cdots, x_{i+2}) - x_1 x_2 \cdots x_{i+2}\big| = \Big|\phi_1\big(\phi_i(x_1, \cdots, x_{i+1}), \sigma(x_{i+2})\big) - x_1 x_2 \cdots x_{i+2}\Big| \\
&\le \Big|\phi_1\big(\phi_i(x_1, \cdots, x_{i+1}), x_{i+2}\big) - \phi_i(x_1, \cdots, x_{i+1}) x_{i+2}\Big| + \big|\phi_i(x_1, \cdots, x_{i+1}) x_{i+2} - x_1 x_2 \cdots x_{i+2}\big| \\
&\le 9(N+1)^{-7kL} + 9i(N+1)^{-7kL} = 9(i+1)(N+1)^{-7kL},
\end{aligned}$$

for any $x_1, x_2, \cdots, x_{i+2} \in [0,1]$, which means we finish the process of induction.

Now let $\phi \coloneqq \phi_{k-1}$, by the principle of induction, we have

$$\phi = \phi_{k-1} \in \mathcal{NN}\big(\text{width} \le 9(N+1)+k-1;\ \text{depth} \le 7kL(k-1)\big)$$

and

$$\begin{aligned}|\phi(x_1,x_2,\cdots,x_k) - x_1x_2\cdots x_k| &= |\phi_{k-1}(x_1,x_2,\cdots,x_k) - x_1x_2\cdots x_k| \\ &\le 9(k-1)(N+1)^{-7kL},\end{aligned}$$

for any $x_1,x_2,\cdots,x_k \in [0,1]$. So we finish the proof. $\square$

4.1.4 Proof of main theorem

With Lemma 4.5 in hand, we are ready to prove Theorem 4.1 for approximating general multivariable polynomials by ReLU networks.

Proof of Theorem 4.1. The case $k=1$ is trivial, so we assume $k \ge 2$ below. Set $\widetilde{k} = \|\boldsymbol{\alpha}\|_1 \le k$, denote $\boldsymbol{\alpha} = (\alpha_1,\alpha_2,\cdots,\alpha_d)$, and let $(z_1,z_2,\cdots,z_{\widetilde{k}}) \in \mathbb{R}^{\widetilde{k}}$ be the vector satisfying

$$z_\ell = x_j, \quad \text{if } \sum_{i=1}^{j-1}\alpha_i < \ell \le \sum_{i=1}^{j}\alpha_i, \quad \text{for } j=1,2,\cdots,d.$$

That is,

$$(z_1,z_2,\cdots,z_{\widetilde{k}}) = \big(\overbrace{x_1,\cdots,x_1}^{\alpha_1\ \text{times}},\ \overbrace{x_2,\cdots,x_2}^{\alpha_2\ \text{times}},\ \cdots,\ \overbrace{x_d,\cdots,x_d}^{\alpha_d\ \text{times}}\big) \in \mathbb{R}^{\widetilde{k}}.$$

Then we have $P(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_d^{\alpha_d} = z_1z_2\cdots z_{\widetilde{k}}$.

We construct the target ReLU network in two steps. First, there exists an affine linear map $\boldsymbol{\mathcal{L}} : \mathbb{R}^d \to \mathbb{R}^k$ that duplicates $\boldsymbol{x}$ to form a new vector

$$(z_1,z_2,\cdots,z_{\widetilde{k}},\ 1,\cdots,1) \in \mathbb{R}^k,$$

i.e., $\mathcal{L}(\boldsymbol{x}) = (z_1, z_2, \cdots, z_{\widetilde{k}},\ 1, \cdots, 1) \in \mathbb{R}^k$ for any $\boldsymbol{x} \in \mathbb{R}^d$. Second, by Lemma 4.5, there exists a function $\psi : \mathbb{R}^k \to \mathbb{R}$ implemented by a ReLU network with width $9(N+1)+k-1$ and depth $7kL(k-1)$ such that ψ maps $(z_1, z_2, \cdots, z_{\widetilde{k}},\ 1, \cdots, 1) \in \mathbb{R}^k$ to $z_1 z_2 \cdots z_{\widetilde{k}}$ within an error $9(k-1)(N+1)^{-7kL}$ for any $z_1, z_2, \cdots, z_{\widetilde{k}} \in [0,1]$.

Hence, we can construct our final target function via $\phi := \psi \circ \mathcal{L}$. Then by Lemma 2.1 (i), ϕ can implemented by a ReLU network with width $9(N+1)+k-1$ and depth $7kL(k-1) \le 7k^2L$, and

$$\begin{aligned} |\phi(\boldsymbol{x}) - P(\boldsymbol{x})| = |\phi(\boldsymbol{x}) - \boldsymbol{x}^{\boldsymbol{\alpha}}| &= |\psi \circ \mathcal{L}(\boldsymbol{x}) - x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha}| \\ &= |\psi(z_1, z_2, \cdots, z_{\widetilde{k}}, 1, \cdots, 1) - z_1 z_2 \cdots z_{\widetilde{k}}| \\ &\le 9(k-1)(N+1)^{-7kL} \le 9k(N+1)^{-7kL}, \end{aligned}$$

for any $x_1, x_2, \cdots, x_d \in [0,1]$. So we finish the proof. □

4.2 Approximation of continuous functions

In this section, let us focus on constructing ReLU networks to approximate continuous functions on $[0,1]^d$.

4.2.1 Main theorem and its proof

Theorem 4.6 below shows that ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ can approximate $f \in C([0,1]^d)$ with an approximation error $19\sqrt{d}\,\omega_f(N^{-2/d}L^{-2/d})$.

Theorem 4.6. *Given a continuous function $f \in C([0,1]^d)$, for any $N, L \in \mathbb{N}^+$ and $p \in [1,\infty]$, there exists a function ϕ implemented by a ReLU network with width $C_1 \max\{d\lfloor N^{1/d}\rfloor,\ N+1\}$ and depth $12L + C_2$ such that*

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

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

The approximation error becomes $19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d}$ when Theorem 4.6 is applied to a function $f \in \text{Hölder}([0,1]^d, \alpha, \lambda)$ as shown in the corollary below, where $\text{Hölder}([0,1]^d, \alpha, \lambda)$ is the space of Hölder continuous functions of order $\alpha \in (0,1]$ with a Hölder constant $\lambda > 0$.

Corollary 4.7. *Given a function $f \in \text{Hölder}([0,1]^d, \alpha, \lambda)$, for any $N, L \in \mathbb{N}^+$ and $p \in [1,\infty]$, there exists a function ϕ implemented by a ReLU network with width $C_1 \max\{d\lfloor N^{1/d}\rfloor,\ N+1\}$ and depth $12L + C_2$ such that*

$$\|\phi - f\|_{L^p([0,1]^d)} \le 19\sqrt{d}\lambda N^{-2\alpha/d}L^{-2\alpha/d},$$

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

The next question is: How much we can improve the approximation error in Theorem 4.6 and Corollary 4.7? In fact, for the Hölder continuous function space, the approximation error in Corollary 4.7 is nearly optimal based on VC-dimension as we shall see later in Section 4.4.

To prove Theorem 4.6, we introduce a theorem below, Theorem 4.8, to construct ReLU networks to uniformly approximate continuous functions outside the trifling region, which can deduce Theorem 4.6 easily by applying Theorem 3.7.

Theorem 4.8. *Given a continuous function $f \in C([0,1]^d)$, for any $N, L \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU network with width $\max\{4d\lfloor N^{1/d}\rfloor + 3d,\ 12N + 8\}$ and depth $12L + 14$ such that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \le |f(\mathbf{0})| + \omega_f(\sqrt{d})$ and*

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

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

Now we are ready to prove Theorem 4.6 by assuming Theorem 4.8 is true, which will be proved later in this section.

Proof of Theorem 4.6. Let us first consider the case $p \in [1,\infty)$. We may assume f is not a constant function since it is a trivial case. Then $\omega_f(r) > 0$ for any $r > 0$.

Set $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ and choose a small $\delta \in (0, \frac{1}{3K}]$ such that

$$\begin{aligned} K d\delta\big(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d})\big)^p &= \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor d\delta\big(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d})\big)^p \\ &\le \big(\omega_f(N^{-2/d}L^{-2/d})\big)^p. \end{aligned} \tag{4.8}$$

By Theorem 4.8, there exists a function ϕ implemented by a ReLU network with width

$$\max\big\{4d\lfloor N^{1/d} \rfloor + 3d,\ 12N + 8\big\} \le 12 \max\big\{d\lfloor N^{1/d} \rfloor,\ N + 1\big\}$$

and depth $12L + 14$ such that $\|\phi\|_{L^\infty(\mathbb{R}^d)} \le |f(\mathbf{0})| + \omega_f(\sqrt{d})$ and

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

Note that $\mu(\Omega([0,1]^d, K, \delta)) \le K d\delta$ and $\|f\|_{L^\infty([0,1]^d)} \le |f(\mathbf{0})| + \omega_f(\sqrt{d})$. Then, by Equation (4.8), we have

$$\begin{aligned} \|f - \phi\|^p_{L^p([0,1]^d)} &= \int_{\Omega([0,1]^d, K, \delta)} |f(\boldsymbol{x}) - \phi(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x} + \int_{[0,1]^d \backslash \Omega([0,1]^d, K, \delta)} |f(\boldsymbol{x}) - \phi(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x} \\ &\le K d\delta\big(2|f(\mathbf{0})| + 2\omega_f(\sqrt{d})\big)^p + \big(18\sqrt{d}\,\omega_f(N^{-2/d}L^{-2/d})\big)^p \\ &\le \big(\omega_f(N^{-2/d}L^{-2/d})\big)^p + \big(18\sqrt{d}\,\omega_f(N^{-2/d}L^{-2/d})\big)^p \\ &\le \big(19\sqrt{d}\,\omega_f(N^{-2/d}L^{-2/d})\big)^p. \end{aligned}$$

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

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

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

By Theorem 4.8, there exists a function implemented $\widetilde{\phi}$ by a ReLU network with width $\max\big\{4d\lfloor N^{1/d} \rfloor + 3d,\ 12N + 8\big\}$ and depth $12L + 14$ such that

$$|f(\boldsymbol{x}) - \widetilde{\phi}(\boldsymbol{x})| \le 18\sqrt{d}\,\omega_f(N^{-2/d}L^{-2/d}) =: \varepsilon, \quad \text{for } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta).$$

By Theorem 3.7, there exists a function ϕ implemented by a ReLU network with width

$$3^d\Big(\max\big\{4d\lfloor N^{1/d}\rfloor+3d,\,12N+8\big\}+4\Big)\le 3^{d+3}\max\big\{d\lfloor N^{1/d}\rfloor,\,N+1\big\}$$

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

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

So we finish the proof. □

It remains to prove Theorem 4.8. To this end, we establish a proposition below to warm up the proof of Theorem 4.8.

Proposition 4.9. *For any $\varepsilon>0$ and arbitrary $N,L,J\in\mathbb{N}^+$ with $J\le N^2L^2$, given J samples $y_j\ge 0$ for $j=0,1,\cdots,J-1$ with*

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

Then there exists $\phi\in\mathcal{NN}(\#\text{input}=1;\ \text{width}\le 12N{+}8;\ \text{depth}\le 4L{+}9;\ \#\text{output}=1)$ 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* $x\in\mathbb{R}$.

4.2.2 Proof of auxiliary theorem

We essentially construct an almost piecewise constant function implemented by a ReLU network with with $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ to approximate the target continuous function f on $[0,1]^d$. We assume f is not a constant 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 $M = N^2L$, $K = \lfloor N^{1/d}\rfloor^2 \lfloor L^{2/d}\rfloor$, and δ be an arbitrary number in $(0, \frac{1}{3K}]$. The proof can be divided into four steps as follows.

1. Divide $[0,1]^d$ into a union of sub-cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}$ 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 for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. See Figure 4.7 for the illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

2. Construct a sub-network to implement a vector-valued function $\boldsymbol{\Phi}_1$ projecting the whole cube $Q_{\boldsymbol{\beta}}$ to the d-dimensional index $\boldsymbol{\beta}$ for each $\boldsymbol{\beta}$, *i.e.*, $\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{\beta}$ for all $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

3. Construct a sub-network to implement a function ϕ_2 mapping the index $\boldsymbol{\beta}$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}$. This step can be further divided into three sub-steps.

   3.1. Construct an affine linear map $\psi_1 : \mathbb{R}^d \to \mathbb{R}$ bijectively mapping the index set $\{0,1,\cdots,K-1\}^d$ to an auxiliary set $\mathcal{A}_1 \subseteq \big\{\frac{j}{2K^d} : j = 0,1,\cdots,2K^d\big\}$ defined later. See Figure 4.8 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 two conditions.

   - Assign the value of g at $\psi(\boldsymbol{\beta}) \in \mathcal{A}_1$ as $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$, *i.e.*, $g \circ \psi_1(\boldsymbol{\beta}) = \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.
   - The values of g at breakpoints in $\mathcal{A}_2 \cup \{1\}$ are properly assigned for applying Proposition 4.9.

   3.3. Apply Proposition 4.9 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 $\phi_2 = \psi_2 \circ \psi_1$ satisfies $\phi_2(\boldsymbol{\beta}) = \psi_2 \circ \psi_1(\boldsymbol{\beta}) \approx g \circ \psi_1(\boldsymbol{\beta}) = \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

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

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 $\boldsymbol{x}_{\boldsymbol{\beta}} \coloneqq \boldsymbol{\beta}/K$ and

$$Q_{\boldsymbol{\beta}} \coloneqq \Big\{\boldsymbol{x}=(x_1,\cdots,x_d)\in[0,1]^d : x_i\in[\tfrac{\beta_i}{K},\tfrac{\beta_i+1}{K}-\delta\cdot 1_{\{\beta_i\le K-2\}}] \text{ for } i=1,\cdots,d\Big\}$$

for each $\boldsymbol{\beta}=(\beta_1,\beta_2,\cdots,\beta_d)\in\{0,1,\cdots,K-1\}^d$. 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

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

See Figure 4.7 for illustrations of $\Omega([0,1]^d,K,\delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$.

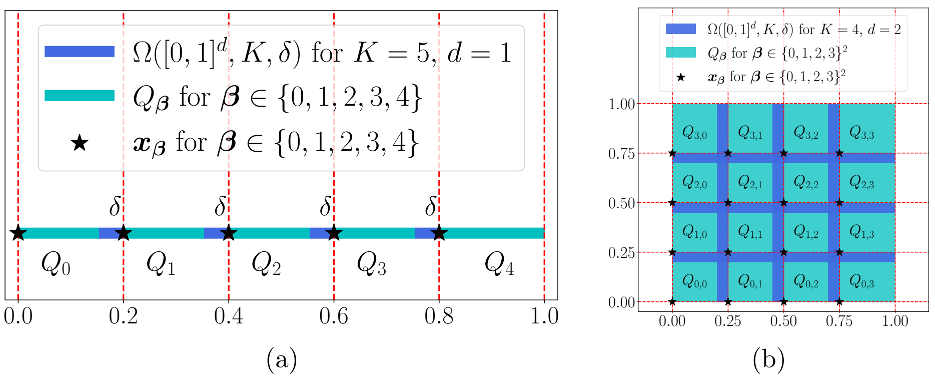

$\Omega([0,1]^d,K,\delta)$ for $K=5$, $d=1$
$Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}\in\{0,1,2,3,4\}$
$\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}\in\{0,1,2,3,4\}$
δ
Q_0
Q_1
Q_2
Q_3
Q_4
0.0
0.2
0.4
0.6
0.8
1.0
(a)
$\Omega([0,1]^d,K,\delta)$ for $K=4$, $d=2$
$Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}\in\{0,1,2,3\}^2$
$\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}\in\{0,1,2,3\}^2$
$Q_{3,0}$ $Q_{3,1}$ $Q_{3,2}$ $Q_{3,3}$
$Q_{2,0}$ $Q_{2,1}$ $Q_{2,2}$ $Q_{2,3}$
$Q_{1,0}$ $Q_{1,1}$ $Q_{1,2}$ $Q_{1,3}$
$Q_{0,0}$ $Q_{0,1}$ $Q_{0,2}$ $Q_{0,3}$
0.00
0.25
0.50
0.75
1.00
(b)

Figure 4.7: Illustrations of $\Omega([0,1]^d,K,\delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$. (a) $K=5$ and $d=1$. (b) $K=4$ and $d=2$.

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

By Theorem 3.12, there exists $\phi_1\in\mathcal{NN}(\text{width}\le 4\lfloor N^{1/d}\rfloor+3;\ \text{depth}\le 4L+5)$ and

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

By defining

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

we have $\boldsymbol{\Phi}_1(\boldsymbol{x}) = \big(\phi_1(x_1), \phi_1(x_2), \cdots, \phi_1(x_d)\big) = \boldsymbol{\beta}$ for all $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$.

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 4.9. To meet the requirements of applying Proposition 4.9, we first define two auxiliary sets $\mathcal{A}_1$ and $\mathcal{A}_2$ as

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

and

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

Clearly, $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\} = \{\frac{j}{2K^d} : j = 0, 1, \cdots, 2K^d\}$ and $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$. See Figure 4.8 for an illustration of $\mathcal{A}_1$ and $\mathcal{A}_2$. Next, we 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

$$\psi_1(\boldsymbol{x}) \coloneqq \frac{x_d}{2K^d} + \sum_{i=1}^{d-1} \frac{x_i}{K^i}, \quad \text{for any } \boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d. \tag{4.9}$$

Then ψ_1 is an affine linear function bijectively mapping the index set $\{0, 1, \cdots, K-$

$1\}^d$ to

$$\begin{aligned}&\left\{\frac{\beta_d}{2K^d}+\sum_{i=1}^{d-1}\frac{\beta_i}{K^i}:\boldsymbol{\beta}=(\beta_1,\cdots,\beta_d)\in\{0,1,\cdots,K-1\}^d\right\}\\ =&\left\{\frac{i}{K^{d-1}}+\frac{k}{2K^d}:i=0,1,\cdots,K^{d-1}-1\quad\text{and}\quad k=0,1,\cdots,K-1\right\}=\mathcal{A}_1.\end{aligned}$$

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 4.9.

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

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

$$g\big(\psi_1(\boldsymbol{\beta})\big)=\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}}),\quad\text{for any }\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d.\tag{4.10}$$

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

- 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 4.9. Note that

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

  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 be 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 4.8 for an example) and satisfies

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

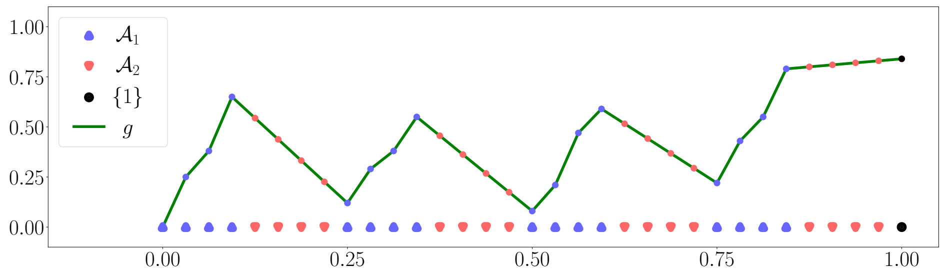


Figure 4.8: An illustration of $\mathcal{A}_1$, $\mathcal{A}_2$, $\{1\}$, and g for the case $d = 2$ and $K = 4$.

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 $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{1\}$.

Since $2K^d = 2\big(\lfloor N^{1/d}\rfloor^2 \lfloor L^{2/d}\rfloor\big)^d \le 2\big(N^2L^2\big) \le N^2\widetilde{L}^2$, where $\widetilde{L} = 2L$, by Proposition 4.9 (set $y_j = g(\frac{j}{2K^d})$ and $\varepsilon = \omega_f(\frac{\sqrt{d}}{K}) > 0$ therein), there exists $\widetilde{\psi}_2 \in \mathcal{NN}(\text{width} \le 12N + 8;\ \text{depth} \le 4\widetilde{L} + 9) = \mathcal{NN}(\text{width} \le 12N + 8;\ \text{depth} \le 8L + 9)$ 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

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

Define $\psi_2(x) := \widetilde{\psi}_2(2K^d x)$ for any $x \in \mathbb{R}$. Then, we have $\psi_2 \in \mathcal{NN}(\text{width} \le 12N + 8;\ \text{depth} \le 8L + 9)$ by Lemma 2.1 (i),

$$0 \le \psi_2(x) = \widetilde{\psi}_2(2K^d x) \le 2\omega_f(\sqrt{d}), \quad \text{for any } x \in \mathbb{R}, \tag{4.11}$$

and

$$|\psi_2(\tfrac{j}{2K^d}) - g(\tfrac{j}{2K^d})| = |\widetilde{\psi}_2(j) - g(\tfrac{j}{2K^d})| \le \omega_f(\tfrac{\sqrt{d}}{K}), \tag{4.12}$$

for $j = 0, 1, \cdots, 2K^d - 1$.

The desired function ϕ_2 can be defined as $\phi_2 := \psi_2 \circ \psi_1$. Note that $\psi_1 : \mathbb{R}^d \to \mathbb{R}$ is an affine linear map and $\psi_2 \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 12N+8;\ \text{depth} \le 8L+9)$. Thus, by Lemma 2.1 (i), $\phi_2 = \psi_2 \circ \psi_1 \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \le 12N+8;\ \text{depth} \le 8L+9)$. By Equation (4.10) and (4.12), we have

$$|\phi_2(\boldsymbol{\beta}) - \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})| = |\psi_2(\psi_1(\boldsymbol{\beta})) - g(\psi_1(\boldsymbol{\beta}))| \le \omega_f(\tfrac{\sqrt{d}}{K}), \tag{4.13}$$

for any $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. Equation (4.11) and $\phi_2 = \psi_2 \circ \psi_1$ implies

$$0 \le \phi_2(\boldsymbol{x}) \le 2\omega_f(\sqrt{d}), \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^d. \tag{4.14}$$

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

Define $\phi := \phi_2 \circ \boldsymbol{\Phi}_1 + f(0) - \omega_f(\sqrt{d})$.

$$\phi_1 \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 4\lfloor N^{1/d} \rfloor + 3;\ \text{depth} \le 4L+5;\ \#\text{output} = 1),$$

implies

$$\boldsymbol{\Phi}_1 \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \le 4d\lfloor N^{1/d} \rfloor + 3d;\ \text{depth} \le 4L+5;\ \#\text{output} = d).$$

Note that $\phi_2 \in \mathcal{NN}(\#\text{input} = d;\ \text{width} \le 12N+8;\ \text{depth} \le 8L+9)$. Thus, by Lemma 2.1, $\phi = \phi_2 \circ \boldsymbol{\Phi}_1 + f(\boldsymbol{0}) - \omega_f(\sqrt{d})$ is in

$$\mathcal{NN}(\text{width} \le \max\{4d\lfloor N^{1/d} \rfloor + 3d, 12N+8\};\ \text{depth} \le (4L+5)+(8L+9) = 12L+14).$$

Finally, let us estimate the approximation error. Recall that $f = \widetilde{f} + f(\boldsymbol{0}) - \omega_f(\sqrt{d})$. By Equation (4.13), for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, we have

$$
\begin{aligned}
|f(\boldsymbol{x})-\phi(\boldsymbol{x})| &= |\widetilde{f}(\boldsymbol{x})-\phi_2\circ\boldsymbol{\Phi}_1(\boldsymbol{x})| = |\widetilde{f}(\boldsymbol{x})-\phi_2(\boldsymbol{\beta})| \\
&\le |\widetilde{f}(\boldsymbol{x})-\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})| + |\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})-\phi_2(\boldsymbol{\beta})| \\
&\le \omega_f(\tfrac{\sqrt{d}}{K}) + \omega_f(\tfrac{\sqrt{d}}{K}) \le 2\omega_f(8\sqrt{d}N^{-2/d}L^{-2/d}),
\end{aligned}
$$

where the last inequality comes from the fact $K=\lfloor N^{1/d}\rfloor^2\lfloor L^{2/d}\rfloor \ge \frac{N^{2/d}L^{2/d}}{8}$ for any $N,L\in\mathbb{N}^+$. Recall the fact $\omega_f(nr)\le n\omega_f(r)$ for any $n\in\mathbb{N}^+$ and $r\in[0,\infty)$. Therefore, for any $\boldsymbol{x}\in\cup_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}Q_{\boldsymbol{\beta}}=[0,1]^d\backslash\Omega([0,1]^d,K,\delta)$, we have

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

It remains to show the upper bound of ϕ. By Equation (4.14) and $\phi=\phi_2\circ\boldsymbol{\Phi}_1+f(\boldsymbol{0})-\omega_f(\sqrt{d})$, we have $\|\phi\|_{L^\infty(\mathbb{R}^d)}\le|f(\boldsymbol{0})|+\omega_f(\sqrt{d})$. Thus, we finish the proof.

4.2.3 Proof of key proposition for auxiliary theorem

Let us prove Proposition 4.9 to end Section 4.2. We apply Theorem 3.5 to prove Lemma 4.10 below, which is a key intermediate conclusion to prove Proposition 4.9.

Lemma 4.10. *For any $\varepsilon>0$ and $N,L\in\mathbb{N}^+$, denote $M=N^2L$ and assume $y_{m,\ell}\ge 0$ for $m=0,1,\cdots,M-1$ and $\ell=0,1,\cdots,L-1$ are samples with*

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

Then there exists $\phi\in\mathcal{NN}(\#\text{input}=2;\ \text{width}\le 12N+8;\ \text{depth}\le 3L+6;\ \#\text{output}=1)$ such that

(i) $|\phi(m,\ell)-y_{m,\ell}|\le\varepsilon$ for $m=0,1,\cdots,M-1$ and $\ell=0,1,\cdots,L-1$.

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

Proof. Define

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

We will construct a function implemented by a ReLU network to map the index (m, ℓ) to $a_{m,\ell}\varepsilon = \lfloor y_{m,\ell}/\varepsilon \rfloor \varepsilon \approx y_{m,\ell}$ for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$.

Define $b_{m,0} \coloneqq 0$ and $b_{m,\ell} \coloneqq a_{m,\ell} - a_{m,\ell-1}$ for $m = 0, 1, \cdots, M-1$ and $\ell = 1, \cdots, L-1$. Since $|y_{m,\ell} - y_{m,\ell-1}| \le \varepsilon$ for all m and ℓ, we have

$$b_{m,\ell} = a_{m,\ell} - a_{m,\ell-1} = \lfloor y_{m,\ell}/\varepsilon \rfloor - \lfloor y_{m,\ell-1}/\varepsilon \rfloor \in \{-1, 0, 1\}.$$

Hence, there exist $c_{m,\ell}$ and $d_{m,\ell} \in \{0, 1\}$ such that $b_{m,\ell} = c_{m,\ell} - d_{m,\ell}$, which implies

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

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

Consider the sample set

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

Its size is $M + 1 = N \cdot \big((NL-1) + 1\big) + 1$. By Theorem 3.2 (set $m = N$ and $n = NL - 1$ 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 Theorem 3.5, there exist $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \le 4N+3;\ \text{depth} \le 3L+3)$

such that

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

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

$$a_{m,\ell} = a_{m,0} + \sum_{j=0}^{\ell} c_{m,j} - \sum_{j=0}^{\ell} d_{m,j} = \psi_1(m) + \psi_2(m,\ell) - \psi_3(m,\ell), \tag{4.15}$$

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

Define

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

Recall that, for any $x_1, x_2 \in \mathbb{R}$, we have

$$\min\{x_1, x_2\} = \tfrac{x_1+x_2-|x_1-x_2|}{2} = \tfrac{\sigma(x_1+x_2)-\sigma(-x_1-x_2)-\sigma(x_1-x_2)-\sigma(-x_1+x_2)}{2}. \tag{4.16}$$

Then the desired function can be implemented by the composition of two sub-networks shown in Figure 4.9.

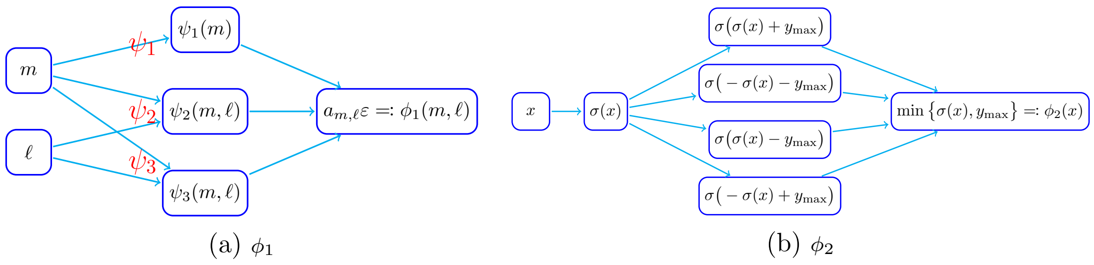

m
ℓ
ψ_1
ψ_2
ψ_3
$\psi_1(m)$
$\psi_2(m,\ell)$
$\psi_3(m,\ell)$
$a_{m,\ell}\varepsilon =: \phi_1(m,\ell)$
(a) ϕ_1
x
$\sigma(x)$
$\sigma(\sigma(x)+y_{\max})$
$\sigma(-\sigma(x)-y_{\max})$
$\sigma(\sigma(x)-y_{\max})$
$\sigma(-\sigma(x)+y_{\max})$
$\min\{\sigma(x), y_{\max}\} =: \phi_2(x)$
(b) ϕ_2

Figure 4.9: Illustrations of two sub-network architectures for implementing the desired function $\phi = \phi_2 \circ \phi_1$ based on Equation (4.15) and (4.16) for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$.

By Theorem 3.1, $\psi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 2NL-1]) \subseteq \mathcal{NN}(\text{width} \le 4N+2; \text{depth} \le L+1)$. Note that $\psi_2, \psi_3 \in \mathcal{NN}(\text{width} \le 4N+3;\ \text{depth} \le 3L+3)$. Thus, $\phi_1 \in \mathcal{NN}\big(\text{width} \le (4N+2)+2(4N+3) = 12N+8;\ \text{depth} \le (3L+3)+1 = 3L+4\big)$ as shown in Figure 4.9. It is clear that $\phi_2 \in \mathcal{NN}(\text{width} \le 4;\ \text{depth} \le 2)$, implying

$\phi = \phi_2 \circ \phi_1 \in \mathcal{NN}\big(\text{width} \le 12N + 8;\ \text{depth} \le (3L+4) + 2 = 3L + 6\big)$ by Lemma 2.1 (ii).

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

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

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

With Lemma 4.10 in hand, we are ready to prove Proposition 4.9.

Proof of Proposition 4.9. Let $M = N^2 L$, then we may assume $J = ML$ since we can set $y_{J-1} = y_J = y_{J+1} = \cdots = y_{ML-1}$ if $J < ML$.

Consider the sample set

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

whose size is $2M + 1 = N \cdot \big((2NL - 1) + 1\big) + 1$. By Theorem 3.2 (set $m = N$ and $n = NL - 1$ therein), there exist

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

such that

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

It follows that

$$\phi_1(j) = m, \quad \text{and} \quad j - L\phi_1(j) = \ell, \quad \text{where } j = mL + \ell, \tag{4.17}$$

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

Note that any number j in $\{0, 1, \ldots, J-1\}$ can be uniquely indexed as $j = mL+\ell$ for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$. So we can denote $y_j = y_{mL+\ell}$ by $y_{m,\ell}$. Then by Lemma 4.10, there exists $\phi_2 \in \mathcal{NN}(\text{width} \le 12N+8;\ \text{depth} \le 3L+6)$ such that

$$|\phi_2(m,\ell) - y_{m,\ell}| \le \varepsilon, \quad \text{for } m = 0, 1, \cdots, M-1 \text{ and } \ell = 0, 1, \cdots, L-1, \tag{4.18}$$

and

$$0 \le \phi_2(x_1, x_2) \le y_{\max}, \quad \text{for any } x_1, x_2 \in \mathbb{R}, \tag{4.19}$$

where $y_{\max} := \max\{y_{m,\ell} : m = 0, 1, \cdots, M-1 \text{ and } \ell = 0, 1, \cdots, L-1\} = \max\{y_j : j = 0, 1, \cdots, ML-1\}$. Then the desired function ϕ can be implemented by the network in Figure 4.10.

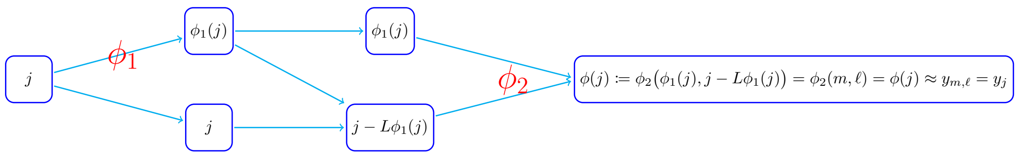

j
ϕ_1
$\phi_1(j)$
j
$\phi_1(j)$
$j - L\phi_1(j)$
ϕ_2
$\phi(j) := \phi_2(\phi_1(j), j - L\phi_1(j)) = \phi_2(m,\ell) = \phi(j) \approx y_{m,\ell} = y_j$

Figure 4.10: An illustration of the network architecture implementing the desired function ϕ based on Equation (4.17). The index $j \in \{0, 1, \cdots, ML-1\}$ is unique represented by $j = mL + \ell$ for $m = 0, 1, \cdots, M-1$ and $\ell = 0, 1, \cdots, L-1$.

Note that $\phi_1 \in \mathcal{NN}(\text{widthvec} = [2N, 4NL-1]) \subseteq \mathcal{NN}(\text{width} \le 8N+2; \text{depth} \le L+1)$ by Theorem 3.1 and $\phi_2 \in \mathcal{NN}(\text{width} \le 12N+8;\ \text{depth} \le 3L+6)$. So $\phi \in \mathcal{NN}(\text{width} \le \max\{8N+2+1, 12N+8\} = 12N+8;\ \text{depth} \le (L+1)+2+(3L+6) = 4L+9)$ as shown in Figure 4.10.

By Equation (4.19) and Figure 4.10, we have

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

Represent $j \in \{0, 1, \cdots, ML-1\}$ via $j = mL + \ell$ for $m = 0, 1, \cdots, M-1$ and

$\ell = 0, 1, \cdots, L-1$, then we have, by Equation (4.18),

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

So we finish the proof. □

We would like to remark that the key idea in the proof of Proposition 4.9 is the "bit extraction" technique, which allows us to store L bits in a binary number $\text{bin}\, 0.\theta_1\theta_2 \cdots \theta_L$ 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.

4.3 Approximation of smooth functions

In Section 4.2, we show that the approximation of a function $f \in C([0,1])^d$, by ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$, admits an approximation error $19\sqrt{d}\,\omega_f(N^{-2/d}L^{-2/d})$ in the L^p-norm for $p \in [1, \infty]$. The next question is whether the smoothness of functions can improve the approximation error. In this section, we investigate the approximation of smooth functions by ReLU networks.

4.3.1 Main theorem and its proof

Theorem 4.11 below shows that ReLU networks with width $\mathcal{O}(N \ln N)$ and depth $\mathcal{O}(L \ln L)$ can approximate a function $f \in C^s([0,1]^d)$ with a nearly optimal approximation error $\mathcal{O}(\|f\|_{C^s([0,1]^d)} N^{-2s/d} L^{-2s/d})$. See Section 4.4.2 for the optimality discussion.

Theorem 4.11. *Given a smooth function $f \in C^s([0,1]^d)$ with $s \in \mathbb{N}^+$, for any $N, L \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU network with width $C_1(N+2)\log_2(8N)$ and depth $C_2(L+2)\log_2(4L) + 2d$ such that*

$$\|\phi - f\|_{L^\infty([0,1]^d)} \le C_3 \|f\|_{C^s([0,1]^d)} N^{-2s/d} L^{-2s/d},$$

where $C_1 = 17s^{d+1}3^d d$, $C_2 = 18s^2$, *and* $C_3 = 85(s+1)^d 8^s$.

As we can see from Theorem 4.11, the smoothness improves the approximation error in N and L. However, we would like to remark that the improved approximation error is at the price of much larger constants.

In Theorem 4.11, the logarithmic terms in width and depth can be further reduced if the approximation error is weaken. Note that for any $\widetilde{N}, \widetilde{L} \in \mathbb{N}^+$ with

$$\widetilde{N} \geq C_1(1+2)\log_2(8) = 17s^{d+1}3^{d+2}d \quad \text{and} \quad \widetilde{L} \geq C_2(1+2)\log_2(4)+2d = 108s^2+2d,$$

there exist $N, L \in \mathbb{N}^+$ such that

$$C_1(N+2)\log_2(8N) \leq \widetilde{N} < C_1\big((N+1)+2\big)\log_2\big(8(N+1)\big)$$

and

$$C_2(L+2)\log_2(4L)+2d \leq \widetilde{L} < C_2\big((L+1)+2\big)\log_2\big(4(L+1)\big)+2d.$$

It follows that

$$N \geq \frac{N+3}{4} > \frac{\widetilde{N}}{4C_1\log_2(8N+8)} \geq \frac{\widetilde{N}}{68s^{d+1}3^d d\log_2(8\widetilde{N}+8)}$$

and

$$L \geq \frac{L+3}{4} > \frac{\widetilde{L}-2d}{4C_2\log_2(4L+4)} \geq \frac{\widetilde{L}-2d}{72s^2\log_2(4\widetilde{L}+4)}.$$

Thus, we have an immediate corollary.

Corollary 4.12. *Given a function* $f \in C^s([0,1]^d)$ *with* $s \in \mathbb{N}^+$, *for any* $\widetilde{N}, \widetilde{L} \in \mathbb{N}^+$, *there exist a function* ϕ *implemented by a ReLU network with width* $\widetilde{N}$ *and depth* $\widetilde{L}$ *such that*

$$\|\phi - f\|_{L^\infty([0,1]^d)} \leq \widetilde{C}_1\|f\|_{C^s([0,1]^d)}\Big(\tfrac{\widetilde{N}}{\widetilde{C}_2\log_2(8\widetilde{N}+8)}\Big)^{-2s/d}\Big(\tfrac{\widetilde{L}-2d}{\widetilde{C}_3\log_2(4\widetilde{L}+4)}\Big)^{-2s/d},$$

for any $\widetilde{N} \geq 17s^{d+1}3^{d+2}d$ *and* $\widetilde{L} \geq 108s^2 + 2d$*, where* $\widetilde{C}_1 = 85(s+1)^d 8^s$*,* $\widetilde{C}_2 = 68s^{d+1}3^d d$*, and* $\widetilde{C}_3 = 72s^2$*.*

To prove Theorem 4.11, we first introduce Theorem 4.13, a simplified version of Theorem 4.11 ignoring the approximation error in the trifling region $\Omega([0,1]^d, K, \delta)$. Then Theorem 4.11 can be easily proved by combining Theorem 3.7 and 4.13 together. Recall that $C_u^s([0,1]^d)$ is the closed unit ball of $C^s([0,1]^d)$.

Theorem 4.13. *Given a smooth function* $f \in C_u^s([0,1]^d)$*, for any* $N, L \in \mathbb{N}^+$*, there exists a function* ϕ *implemented by ReLU network with width* $16s^{d+1}d(N+2)\log_2(8N)$ *and depth* $18s^2(L+2)\log_2(4L)$ *such that*

$$|\phi(\boldsymbol{x}) - f(\boldsymbol{x})| \leq 84(s+1)^d 8^s N^{-2s/d} L^{-2s/d}, \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta),$$

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

Theorem 4.13 will be proved in Section 4.3.3. By assuming Theorem 4.13 is true, we can prove Theorem 4.11 based on Theorem 3.7.

Proof of Theorem 4.11. We may assume $\|f\|_{C^s([0,1]^d)} > 0$ since $\|f\|_{C^s([0,1]^d)} = 0$ is a trivial case. Define $\widetilde{f} := \frac{f}{\|f\|_{C^s([0,1]^d)}} \in C_u^s([0,1]^d)$, set $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$, and choose a small $\delta \in (0, \frac{1}{3K}]$ such that

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

By Theorem 4.13, there exists a function $\widehat{\phi}$ implemented by a ReLU network with width $16s^{d+1}d(N+2)\log_2(8N)$ and depth $18s^2(L+2)\log_2(4L)$ such that

$$|\widehat{\phi}(\boldsymbol{x}) - \widetilde{f}(\boldsymbol{x})| \leq 84(s+1)^d 8^s N^{-2s/d} L^{-2s/d}, \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta),$$

By Theorem 3.7, there exists a new function $\widetilde{\phi}$ implemented by a ReLU network with width

$$3^d\Big(16s^{d+1}d(N+2)\log_2(8N)+4\Big) \le 17s^{d+1}3^d d(N+2)\log_2(8N)$$

and depth $18s^2(L+2)\log_2(4L)+2d$ such that

$$\begin{aligned}\|\widetilde{\phi}-\widetilde{f}\|_{L^\infty([0,1]^d)} &\le 84(s+1)^d 8^s N^{-2s/d}L^{-2s/d} + d\cdot\omega_f(\delta)\\ &\le 85(s+1)^d 8^s N^{-2s/d}L^{-2s/d}.\end{aligned}$$

Finally, set $\phi = \|f\|_{C^s([0,1]^d)}\cdot\widetilde{\phi}$, then

$$\begin{aligned}\|\phi-f\|_{L^\infty([0,1]^d)} &= \|f\|_{C^s([0,1]^d)}\cdot\|\widetilde{f}-\widetilde{\phi}\|_{L^\infty([0,1]^d)}\\ &\le 85(s+1)^d 8^s\|f\|_{C^s([0,1]^d)}N^{-2s/d}L^{-2s/d},\end{aligned}$$

and ϕ can also be implemented by a ReLU network with width $17s^{d+1}3^d d(N+2)\log_2(8N)$ and depth $18s^2(L+2)\log_2(4L)+2d$. So we finish the proof. □

It remains to prove Theorem 4.13, a weaker version of Theorem 4.11 targeting a ReLU network constructed to approximate a smooth function outside the trifling region. We discuss the ideas of the proof in Section 4.3.2 and give the detailed proof in Section 4.3.3.

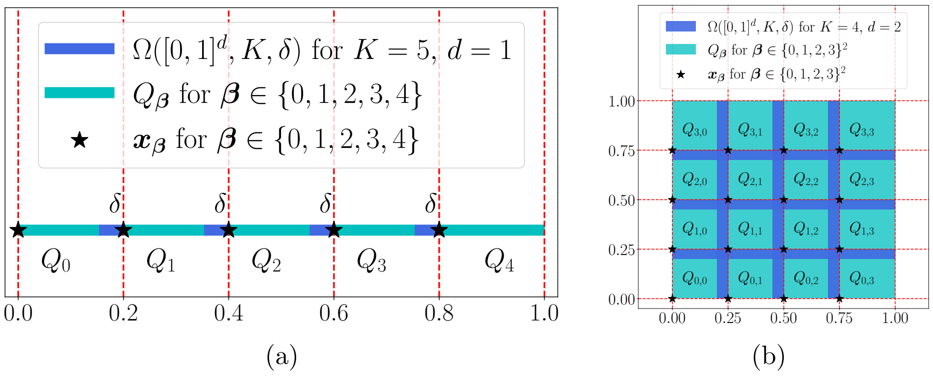
4.3.2 Ideas of proving auxiliary theorem

Set $K=\mathcal{O}(N^{2/d}L^{2/d})$ and let $\Omega([0,1]^d,K,\delta)$ partition $[0,1]^d$ into K^d cubes $Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$. In particular, for each $\boldsymbol{\beta}=(\beta_1,\beta_2,\cdots,\beta_d)\in\{0,1,\cdots,K-1\}^d$, we define $\boldsymbol{x}_{\boldsymbol{\beta}}\coloneqq\boldsymbol{\beta}/K$ and

$$Q_{\boldsymbol{\beta}}=\Big\{\boldsymbol{x}=(x_1,x_2,\cdots,x_d): x_i\in[\tfrac{\beta_i}{K},\tfrac{\beta_i+1}{K}-\delta\cdot 1_{\{\beta_i\le K-2\}}]\text{ for } i=1,2,\cdots,d\Big\}.$$

Clearly, $[0,1]^d=\Omega([0,1]^d,K,\delta)\bigcup\big(\cup_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}Q_{\boldsymbol{\beta}}\big)$ and $\boldsymbol{x}_{\boldsymbol{\beta}}$ is the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm. See Figure 4.11 for the illustrations of $\Omega([0,1]^d,K,\delta)$,

$Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.



$\Omega([0,1]^d, K, \delta)$ for $K=5$, $d=1$
$Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3,4\}$
$\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3,4\}$
δ
Q_0
Q_1
Q_2
Q_3
Q_4
0.0
0.2
0.4
0.6
0.8
1.0
(a)
$\Omega([0,1]^d, K, \delta)$ for $K=4$, $d=2$
$Q_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3\}^2$
$\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,2,3\}^2$
$Q_{3,0}$ $Q_{3,1}$ $Q_{3,2}$ $Q_{3,3}$
$Q_{2,0}$ $Q_{2,1}$ $Q_{2,2}$ $Q_{2,3}$
$Q_{1,0}$ $Q_{1,1}$ $Q_{1,2}$ $Q_{1,3}$
$Q_{0,0}$ $Q_{0,1}$ $Q_{0,2}$ $Q_{0,3}$
0.00
0.25
0.50
0.75
1.00
(b)

Figure 4.11: Illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. (a) $K=5$ and $d=1$. (b) $K=4$ and $d=2$.

For any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$, there exists $\xi_{\boldsymbol{x}} \in (0,1)$ such that

$$f(\boldsymbol{x}) = \underbrace{\sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} \tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}}}_{\mathscr{T}_1} + \underbrace{\sum_{\|\boldsymbol{\alpha}\|_1 = s} \tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\beta}}+\xi_{\boldsymbol{x}}\boldsymbol{h})}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}}}_{\mathscr{T}_2} =: \mathscr{T}_1 + \mathscr{T}_2,$$ [②]

where $\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{\beta}} = \boldsymbol{x} - \boldsymbol{\beta}/K$. It is clear that the magnitude of $\mathscr{T}_2$ is bounded by $\mathcal{O}(K^{-s}) = \mathcal{O}(N^{-2s/d}L^{-2s/d})$. So we only need to construct a function in $\mathcal{NN}\big(\text{width} \le \mathcal{O}(N \ln N);\ \text{depth} \le \mathcal{O}(L \ln L)\big)$ to approximate

$$\mathscr{T}_1 = \sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} \tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}}$$

within an error $\mathcal{O}(N^{-2s/d}L^{-2s/d})$. To approximate $\mathscr{T}_1$ well by ReLU networks, we need three key steps as follows.

- Construct a ReLU network to implement a vector-valued function $\boldsymbol{\Psi}: \mathbb{R}^d \to \mathbb{R}^d$ projecting the whole cube $Q_{\boldsymbol{\beta}}$ to the point $\boldsymbol{x}_{\boldsymbol{\beta}} = \frac{\boldsymbol{\beta}}{K}$, *i.e.*, $\boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

- Construct a ReLU network to implement a function $P_{\boldsymbol{\alpha}}: \mathbb{R}^d \to \mathbb{R}$ approximating the polynomial $\boldsymbol{h}^{\boldsymbol{\alpha}}$ for each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \le s-1$.

[②] $\sum_{\|\boldsymbol{\alpha}\|_1=s}$ is short for $\sum_{\|\boldsymbol{\alpha}\|_1=s,\, \boldsymbol{\alpha}\in\mathbb{N}^d}$. The same notation is used throughout this dissertation.

- Construct a ReLU network to implement a function $\phi_{\boldsymbol{\alpha}} : \mathbb{R}^d \to \mathbb{R}$ approximating $\partial^{\boldsymbol{\alpha}} f$ via solving a point fitting problem, *i.e.*, $\phi_{\boldsymbol{\alpha}}$ should fit $\partial^{\boldsymbol{\alpha}} f$ well at all points in $\big\{\boldsymbol{x}_{\boldsymbol{\beta}} : \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d\big\}$ for each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \le s-1$. That is, for each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \le s-1$, we need to design $\phi_{\boldsymbol{\alpha}}$ to make the following equation true.

$$\big|\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\beta}})\big| \le \mathcal{O}(N^{-2s/d}L^{-2s/d}), \quad \text{for any } \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d.$$

Note that the first and second steps are done by Theorem 3.12 and 4.1, respectively. We will establish a proposition for the last step, which will be applied to support the construction of the desired ReLU networks. Its proof will be available later in Section 4.3.4. In fact, we can construct ReLU networks with width $\mathcal{O}(sN \ln N)$ and depth $\mathcal{O}(L \ln L)$ to fit $\mathcal{O}(N^2L^2)$ points with an error $\mathcal{O}(N^{-2s}L^{-2s}) \le \mathcal{O}(N^{-2s/d}L^{-2s/d})$ as shown in Proposition 4.14 below.

Proposition 4.14. *Given any $N, L, s \in \mathbb{N}^+$ and $\xi_i \in [0,1]$ for $i = 0, 1, \cdots, N^2L^2-1$, there exists $\phi \in \mathcal{NN}(\#\text{input} = 1;\ \text{width} \le 16s(N+1)\log_2(8N);\ \text{depth} \le 5(L+2)\log_2(4L);\ \#\text{output} = 1)$ such that*

(i) $|\phi(i) - \xi_i| \le N^{-2s}L^{-2s}$ *for* $i = 0, 1, \cdots, N^2L^2-1$.

(ii) $0 \le \phi(x) \le 1$ *for any* $x \in \mathbb{R}$.

The proof of Proposition 4.14 can be found in Section 4.3.4. Finally, let us summarize the main ideas of proving Theorem 4.13 in Table 4.1. See the detailed proof in Section 4.3.3.

4.3.3 Proof of auxiliary theorem

According to the key ideas of proving Theorem 4.13 we summarized in Section 4.3.2, we are ready to present the detailed proof.

Proof of Theorem 4.13. The detailed proof can be divided into three steps as follows.

Table 4.1: Key ideas of approximating a smooth function. Note that $\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$.

target function	function implemented by network	width	depth	approximation error
step function	$\boldsymbol{\Psi}(\boldsymbol{x})$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	no error outside $\Omega([0,1]^d, K, \delta)$
$x_1 x_2$	$\varphi(x_1, x_2)$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathscr{E}_1 = 216(N+1)^{-2s(L+1)}$
$\boldsymbol{h}^{\boldsymbol{\alpha}}$	$P_{\boldsymbol{\alpha}}(\boldsymbol{h})$	$\mathcal{O}(N)$	$\mathcal{O}(L)$	$\mathscr{E}_2 = 9s(N+1)^{-7sL}$
$\partial^{\boldsymbol{\alpha}} f(\boldsymbol{\Psi}(\boldsymbol{x}))$	$\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))$	$\mathcal{O}(N \ln N)$	$\mathcal{O}(L \ln L)$	$\mathscr{E}_3 = 2N^{-2s}L^{-2s}$
$\sum_{\Vert\boldsymbol{\alpha}\Vert \le s-1} \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}}$	$\sum_{\Vert\boldsymbol{\alpha}\Vert \le s-1} \varphi\Big(\frac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\Big)$	$\mathcal{O}(N \ln N)$	$\mathcal{O}(L \ln L)$	$\mathcal{O}(\mathscr{E}_1 + \mathscr{E}_2 + \mathscr{E}_3)$
$f(\boldsymbol{x})$	$\phi(\boldsymbol{x}) \coloneqq \sum_{\Vert\boldsymbol{\alpha}\Vert \le s-1} \varphi\Big(\frac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}\big(\boldsymbol{x} - \boldsymbol{\Psi}(\boldsymbol{x})\big)\Big)$	$\mathcal{O}(N \ln N)$	$\mathcal{O}(L \ln L)$	$\mathcal{O}(\Vert\boldsymbol{h}\Vert_2^{-s} + \mathscr{E}_1 + \mathscr{E}_2 + \mathscr{E}_3)$ $\le \mathcal{O}(K^{-s}) = \mathcal{O}(N^{-2s/d}L^{-2s/d})$

Step 1: Set up.

Set $K = \lfloor N^{1/d} \rfloor^2 \lfloor L^{2/d} \rfloor$ and let $\Omega([0,1]^d, K, \delta)$ partition $[0,1]^d$ into K^d cubes $Q_{\boldsymbol{\beta}}$ for each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. In particular, for each $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d) \in \{0, 1, \cdots, K-1\}^d$, we define $\boldsymbol{x}_{\boldsymbol{\beta}} \coloneqq \boldsymbol{\beta}/K$ and

$$Q_{\boldsymbol{\beta}} \coloneqq \Big\{ \boldsymbol{x} = (x_1, x_2, \cdots, x_d) : x_i \in [\tfrac{\beta_i}{K}, \tfrac{\beta_i+1}{K} - \delta \cdot 1_{\{\beta_i \le K-2\}}] \text{ for } i = 1, 2, \cdots, d \Big\}.$$

Clearly, $[0,1]^d = \Omega([0,1]^d, K, \delta) \bigcup \big(\cup_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d} Q_{\boldsymbol{\beta}} \big)$ and $\boldsymbol{x}_{\boldsymbol{\beta}}$ is the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\Vert \cdot \Vert_1$ norm. See Figure 4.11 for the illustrations of $\Omega([0,1]^d, K, \delta)$, $Q_{\boldsymbol{\beta}}$, and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$.

By Theorem 3.12, there exists $\psi \in \mathcal{NN}(\text{width} \le 4N+3;\ \text{depth} \le 4L+5)$ such that

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

Then, for each $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, $\psi(x_i) = \beta_i$ for all $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ for $i = 1, 2, \cdots, d$.

Define

$$\boldsymbol{\Psi}(\boldsymbol{x}) \coloneqq \big(\psi(x_1), \psi(x_2), \cdots, \psi(x_d)\big)/K, \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^d,$$

then

$$\boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{\beta}/K = \boldsymbol{x}_{\boldsymbol{\beta}}, \quad \text{if } \boldsymbol{x} \in Q_{\boldsymbol{\beta}}, \quad \text{for any } \boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d.$$

For any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$, by the Taylor expansion, there exists $\xi_{\boldsymbol{x}} \in (0,1)$ such that

$$f(\boldsymbol{x}) = \sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} \tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}} + \sum_{\|\boldsymbol{\alpha}\|_1 = s} \tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{\Psi}(\boldsymbol{x}) + \xi_{\boldsymbol{x}} \boldsymbol{h})}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}}, \quad \text{where } \boldsymbol{h} = \boldsymbol{x} - \boldsymbol{\Psi}(\boldsymbol{x}).$$

Step 2: Construct the desired function ϕ.

By Lemma 4.4, there exists $\varphi \in \mathcal{NN}\big(\text{width} \le 9(N+1)+1;\ \text{depth} \le 2s(L+1)\big)$ such that

$$|\varphi(x_1,x_2) - x_1 x_2| \le 216(N+1)^{-2s(L+1)} =: \mathscr{E}_1, \quad \text{for any } x_1, x_2 \in [-3,3]. \tag{4.20}$$

For each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \le s$, by Theorem 4.1, there exists $P_{\boldsymbol{\alpha}}$ in

$$\mathcal{NN}\big(\text{width} \le 9(N+1)+s-1;\ \text{depth} \le 7s^2 L\big)$$

such that

$$|P_\alpha(\boldsymbol{x}) - \boldsymbol{x}^{\boldsymbol{\alpha}}| \le 9s(N+1)^{-7sL} =: \mathscr{E}_2, \quad \text{for any } \boldsymbol{x} \in [0,1]^d. \tag{4.21}$$

For each $i = 0,1,\cdots,K^d-1$, define

$$\boldsymbol{\eta}(i) = (\eta_1, \eta_2, \cdots, \eta_d) \in \{0,1,\cdots,K-1\}^d$$

such that $\sum_{j=1}^{d} \eta_j K^{j-1} = i$. Such a map $\boldsymbol{\eta}$ is a bijection from $\{0,1,\cdots,K^d-1\}$ to

$\{0,1,\cdots,K-1\}^d$. For each $\boldsymbol{\alpha}\in\mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1\le s-1$, define

$$\xi_{\boldsymbol{\alpha},i}=\big(\partial^{\boldsymbol{\alpha}}f(\tfrac{\boldsymbol{\eta}(i)}{K})+1\big)/2,\quad \text{for any } i\in\{0,1,\cdots,K^d-1\}.$$

Note that $K^d=\big(\lfloor N^{1/d}\rfloor^2\lfloor L^{2/d}\rfloor\big)^d\le N^2L^2$ and $\xi_{\boldsymbol{\alpha},i}=\big(\partial^{\boldsymbol{\alpha}}f(\frac{\boldsymbol{\eta}(i)}{K})+1\big)/2\in[0,1]$ for $i=0,1,\cdots,K^d-1$. By Proposition 4.14, there exists

$$\widetilde{\phi}_{\boldsymbol{\alpha}}\in\mathcal{NN}\big(\text{width}\le 16s(N+1)\log_2(8N);\ \text{depth}\le 5(L+2)\log_2(4L)\big)$$

such that, for each $\boldsymbol{\alpha}\in\mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1\le s-1$, we have

$$|\widetilde{\phi}_{\boldsymbol{\alpha}}(i)-\xi_{\boldsymbol{\alpha},i}|\le N^{-2s}L^{-2s},\quad \text{for } i=0,1,\cdots,K^d-1.$$

For each $\boldsymbol{\alpha}\in\mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1\le s-1$, define

$$\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}):=2\widetilde{\phi}_{\boldsymbol{\alpha}}\Big(\sum_{j=1}^d x_jK^{j-1}\Big)-1,\quad \text{for any } \boldsymbol{x}=(x_1,x_2,\cdots,x_d)\in\mathbb{R}^d,$$

which implies by Lemma 2.1 that

$$\phi_{\boldsymbol{\alpha}}\in\mathcal{NN}\big(\text{width}\le 16s(N+1)\log_2(8N);\ \text{depth}\le 5(L+2)\log_2(4L)\big).$$

Then, for each $\boldsymbol{\eta}=\boldsymbol{\eta}(i)=(\eta_1,\eta_2,\cdots,\eta_d)\in\{0,1,\cdots,K-1\}^d$ corresponding to $i=\sum_{j=1}^d\eta_jK^{j-1}\in\{0,1,\cdots,K^d-1\}$ and each $\boldsymbol{\alpha}\in\mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1\le s-1$, we have

$$\begin{aligned}\big|\phi_{\boldsymbol{\alpha}}(\tfrac{\boldsymbol{\eta}}{K})-\partial^{\boldsymbol{\alpha}}f(\tfrac{\boldsymbol{\eta}}{K})\big|&=\Big|2\widetilde{\phi}_{\boldsymbol{\alpha}}\Big(\sum_{j=1}^d\eta_jK^{j-1}\Big)-1-(2\xi_{\boldsymbol{\alpha},i}-1)\Big|\\&=2|\widetilde{\phi}_{\boldsymbol{\alpha}}(i)-\xi_{\boldsymbol{\alpha},i}|\le 2N^{-2s}L^{-2s}.\end{aligned}$$

Thus, for each $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$, we have

$$\big|\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}})-\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})\big|=\big|\phi_{\boldsymbol{\alpha}}(\tfrac{\boldsymbol{\beta}}{K})-\partial^{\boldsymbol{\alpha}}f(\tfrac{\boldsymbol{\beta}}{K})\big|\le 2N^{-2s}L^{-2s}=:\mathscr{E}_3. \tag{4.22}$$

Now we can construct the target function ϕ as

$$\phi(\boldsymbol{x}) := \sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} \varphi\Big(\tfrac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}\big(\boldsymbol{x}-\boldsymbol{\Psi}(\boldsymbol{x})\big)\Big), \quad \text{for any } \boldsymbol{x} \in \mathbb{R}^d. \tag{4.23}$$

Step 3: Estimate approximation error.

Fix $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, let us estimate the approximation error for a **fixed** $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$. Recall that $\boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{x}_{\boldsymbol{\beta}}$ and $\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{\beta}}$. Then, it is easy to verify that $|f(\boldsymbol{x}) - \phi(\boldsymbol{x})|$ is bounded by

$$\Bigg|\sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} \tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} + \sum_{\|\boldsymbol{\alpha}\|_1 = s} \tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{\Psi}(\boldsymbol{x})+\xi_{\boldsymbol{x}}\boldsymbol{h})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} - \sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} \varphi\Big(\tfrac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}\big(\boldsymbol{x}-\boldsymbol{\Psi}(\boldsymbol{x})\big)\Big)\Bigg|$$

$$\le \underbrace{\sum_{\|\boldsymbol{\alpha}\|_1 = s} \Big|\tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\beta}+\xi_{\boldsymbol{x}}\boldsymbol{h})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}}\Big|}_{\mathscr{I}_1} + \underbrace{\sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} \Big|\tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\beta})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} - \varphi\big(\tfrac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\beta})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big)\Big|}_{\mathscr{I}_2} =: \mathscr{I}_1 + \mathscr{I}_2.$$

Recall that

$$\sum_{\|\boldsymbol{\alpha}\|_1 = s} 1 = \Big|\big\{\boldsymbol{\alpha} \in \mathbb{N}^d : \|\boldsymbol{\alpha}\|_1 = s\big\}\Big| \le (s+1)^{d-1} \text{ [3]}$$

and

$$\sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} 1 = \sum_{i=0}^{s-1}\Bigg(\sum_{\|\boldsymbol{\alpha}\|_1 = i} 1\Bigg) \le \sum_{i=0}^{s-1}(i+1)^{d-1} \le s \cdot (s-1+1)^{d-1} = s^d.$$

For the first part $\mathscr{I}_1$, we have

$$\mathscr{I}_1 = \sum_{\|\boldsymbol{\alpha}\|_1 = s} \Big|\tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\beta}+\xi_{\boldsymbol{x}}\boldsymbol{h})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}}\Big| \le \sum_{\|\boldsymbol{\alpha}\|_1 = s} \Big|\tfrac{1}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}}\Big| \le (s+1)^{d-1}K^{-s}.$$

[3] In fact, we have $\big|\{\boldsymbol{\alpha} \in \mathbb{N}^d : \|\boldsymbol{\alpha}\|_1 = s\}\big| = \binom{s+d-1}{d-1}$, implying $(s/d+1)^{d-1} \le \sum_{\|\boldsymbol{\alpha}\|_1 = s} 1 \le (s+1)^{d-1}$. Thus, the lower bound of the estimate is still exponentially large in d. To the best of our knowledge, we cannot avoid a constant prefactor that is exponentially large in d when Taylor expansion is used in the analysis.

Now let us estimate the second part $\mathscr{I}_2$ as follows.

$$\begin{aligned}\mathscr{I}_2 &= \sum_{\|\boldsymbol{\alpha}\|_1\le s-1}\Big|\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} - \varphi\big(\tfrac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big)\Big| \\ &\le \underbrace{\sum_{\|\boldsymbol{\alpha}\|_1\le s-1}\Big|\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} - \varphi\big(\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big)\Big|}_{\mathscr{I}_{2,1}} + \underbrace{\sum_{\|\boldsymbol{\alpha}\|_1\le s-1}\Big|\varphi\big(\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big) - \varphi\big(\tfrac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big)\Big|}_{\mathscr{I}_{2,2}} \\ &=: \mathscr{I}_{2,1} + \mathscr{I}_{2,2}.\end{aligned}$$

Note that $\mathscr{E}_2 = 9s(N+1)^{-7sL} \le 9s(2)^{-7s} \le 2$. By Equation (4.21), it easy to verify that $P_{\boldsymbol{\alpha}}(\boldsymbol{h}) \in [-2,3] \subseteq [-3,3]$ for each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \le s-1$. Clearly, $\boldsymbol{h} \in [0,1]^d$ and $\frac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!} \in [-1,1] \subseteq [-3,3]$ for each $\boldsymbol{\alpha}$. Then, by Equation (4.20) and (4.21), we have

$$\begin{aligned}\mathscr{I}_{2,1} &= \sum_{\|\boldsymbol{\alpha}\|_1\le s-1}\Big|\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} - \varphi\big(\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big)\Big| \\ &\le \sum_{\|\boldsymbol{\alpha}\|_1\le s-1}\bigg(\Big|\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} - \tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}P_{\boldsymbol{\alpha}}(\boldsymbol{h})\Big| + \underbrace{\Big|\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}P_{\boldsymbol{\alpha}}(\boldsymbol{h}) - \varphi\big(\tfrac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big)\Big|}_{\le \mathscr{E}_1 \text{ by Eq. (4.20)}}\bigg) \\ &\le \sum_{\|\boldsymbol{\alpha}\|_1\le s-1}\Big(\tfrac{1}{\boldsymbol{\alpha}!}\underbrace{\big|\boldsymbol{h}^{\boldsymbol{\alpha}} - P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big|}_{\le \mathscr{E}_2 \text{ by Eq. (4.21)}} + \mathscr{E}_1\Big) \le \sum_{\|\boldsymbol{\alpha}\|_1\le s-1}(\mathscr{E}_2+\mathscr{E}_1) \le s^d(\mathscr{E}_1+\mathscr{E}_2).\end{aligned}$$

To estimate $\mathscr{I}_{2,2}$, we need the following fact derived from Equation (4.20):

$$\begin{aligned}|\varphi(x_1,x_2)-\varphi(\widetilde{x}_1,x_2)| &\le \underbrace{|\varphi(x_1,x_2)-x_1x_2|}_{\le \mathscr{E}_1 \text{ by Eq. (4.20)}} + \underbrace{|\varphi(\widetilde{x}_1,x_2)-\widetilde{x}_1x_2|}_{\le \mathscr{E}_1 \text{ by Eq. (4.20)}} + |x_1x_2-\widetilde{x}_1x_2| \\ &\le 2\mathscr{E}_1 + 3|x_1-\widetilde{x}_1|,\end{aligned} \tag{4.24}$$

for any $x_1, \widetilde{x}_1, x_2 \in [-3,3]$.

Since $\mathscr{E}_3 = 2N^{-2s}L^{-2s} \le 2$ and $\frac{\partial^{\boldsymbol{\alpha}}f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!} \in [-1,1]$ for each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \le s-1$, we have $\frac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!} \in [-3,3]$ by Equation (4.22). Recall that $P_{\boldsymbol{\alpha}}(\boldsymbol{h}) \in [-3,3]$.

Then, by Equation (4.22) and (4.24), we have

$$\begin{aligned}\mathscr{I}_{2,2} &= \sum_{\|\boldsymbol{\alpha}\|_1\le s-1} \underbrace{\Big|\varphi\big(\tfrac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big) - \varphi\big(\tfrac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}})}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}(\boldsymbol{h})\big)\Big|}_{\le 2\mathscr{E}_1 + \frac{3}{\boldsymbol{\alpha}!}|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\beta}}) - \phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}})| \text{ by Eq. } (4.24)} \\ &\le \sum_{\|\boldsymbol{\alpha}\|_1\le s-1} \Big(2\mathscr{E}_1 + \tfrac{3}{\boldsymbol{\alpha}!}\underbrace{\big|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\beta}}) - \phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{\boldsymbol{\beta}})\big|}_{\le \mathscr{E}_3 \text{ by Eq. } (4.22)}\Big) \le \sum_{\|\boldsymbol{\alpha}\|_1\le s-1} (2\mathscr{E}_1 + 3\mathscr{E}_3) \le s^d(2\mathscr{E}_1 + 3\mathscr{E}_3).\end{aligned}$$

Therefore, we have

$$\begin{aligned}|f(\boldsymbol{x}) - \phi(\boldsymbol{x})| \le \mathscr{I}_1 + \mathscr{I}_2 \le \mathscr{I}_1 + \mathscr{I}_{2,1} + \mathscr{I}_{2,2} & \\ &\le (s+1)^{d-1}K^{-s} + s^d(\mathscr{E}_1 + \mathscr{E}_2) + s^d(2\mathscr{E}_1 + 3\mathscr{E}_3) \\ &\le (s+1)^d(K^{-s} + 3\mathscr{E}_1 + \mathscr{E}_2 + 3\mathscr{E}_3).\end{aligned}$$

Recall the fact $[0,1]^d\backslash\Omega([0,1]^d, K, \delta) = \cup_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d} Q_{\boldsymbol{\beta}}$. Since $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ and $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ are arbitrary, we have,

$$|f(\boldsymbol{x}) - \phi(\boldsymbol{x})| \le (s+1)^d(K^{-s} + 3\mathscr{E}_1 + \mathscr{E}_2 + 3\mathscr{E}_3),$$

for any $\boldsymbol{x} \in [0,1]^d\backslash\Omega([0,1]^d, K, \delta)$. Note that $K = \lfloor N^{1/d}\rfloor^2\lfloor L^{2/d}\rfloor \ge \frac{N^{2/d}L^{2/d}}{8}$ and

$$(N+1)^{-7sL} \le (N+1)^{-2s(L+1)} \le (N+1)^{-2s}2^{-2sL} \le N^{-2s}L^{-2s}.$$

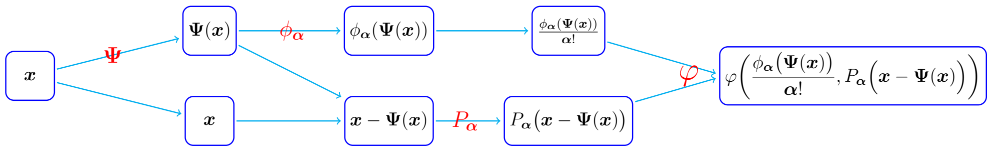
Then we have

$$\begin{aligned}&(s+1)^d(K^{-s} + 3\mathscr{E}_1 + \mathscr{E}_2 + 3\mathscr{E}_3) \\ &= (s+1)^d\Big(K^{-s} + 648(N+1)^{-2s(L+1)} + 9s(N+1)^{-7sL} + 6N^{-2s}L^{-2s}\Big) \\ &\le (s+1)^d\Big(8^sN^{-2s/d}L^{-2s/d} + (654+9s)N^{-2s}L^{-2s}\Big) \\ &\le (s+1)^d(8^s + 654 + 9s)N^{-2s/d}L^{-2s/d} \le 84(s+1)^d8^sN^{-2s/d}L^{-2s/d}.\end{aligned}$$

It remains to estimate the width and depth of the network implementing ϕ.

Recall that, for each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \le s-1$,

$$\begin{cases} \boldsymbol{\Psi} \in \mathcal{NN}\big(\text{width} \le d(4N+3);\ \text{depth} \le 4L+5\big), \\ \phi_{\boldsymbol{\alpha}} \in \mathcal{NN}\big(\text{width} \le 16s(N+1)\log_2(8N);\ \text{depth} \le 5(L+2)\log_2(4L)\big), \\ P_{\boldsymbol{\alpha}} \in \mathcal{NN}\big(\text{width} \le 9(N+1)+s-1;\ \text{depth} \le 7s^2L\big), \\ \varphi \in \mathcal{NN}\big(\text{width} \le 9N+10;\ \text{depth} \le 2s(L+1)\big). \end{cases}$$



$\boldsymbol{x}$
$\boldsymbol{\Psi}$
$\boldsymbol{\Psi}(\boldsymbol{x})$
$\phi_{\boldsymbol{\alpha}}$
$\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))$
$\frac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}$
$\boldsymbol{x}$
$\boldsymbol{x}-\boldsymbol{\Psi}(\boldsymbol{x})$
$P_{\boldsymbol{\alpha}}$
$P_{\boldsymbol{\alpha}}(\boldsymbol{x}-\boldsymbol{\Psi}(\boldsymbol{x}))$
φ
$\varphi\Big(\frac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}\big(\boldsymbol{x}-\boldsymbol{\Psi}(\boldsymbol{x})\big)\Big)$

Figure 4.12: An illustration of the sub-network architecture implementing $\varphi\Big(\frac{\phi_{\boldsymbol{\alpha}}(\boldsymbol{\Psi}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}, P_{\boldsymbol{\alpha}}\big(\boldsymbol{x}-\boldsymbol{\Psi}(\boldsymbol{x})\big)\Big)$ for each $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\| \le s-1$ when $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

By Equation (4.23) and Figure 4.12, it easy to verify ϕ can be implemented by a ReLU network with width

$$\begin{aligned} \sum_{\|\boldsymbol{\alpha}\|_1 \le s-1} 16sd(N+2)\log_2(8N) &\le s^d \cdot 16sd(N+2)\log_2(8N) \\ &= 16s^{d+1}d(N+2)\log_2(8N) \end{aligned}$$

and depth

$$(4L+5) \ + \ 5(L+2)\log_2(4L) \ + \ 7s^2L \ + \ 2s(L+1) \ + \ 3 \le 18s^2(L+2)\log_2(4L).$$

So we finish the proof. □

4.3.4 Proof of key proposition for auxiliary theorem

Let us discuss the construction of ReLU networks to fit a collection of points in $\mathbb{R}^2$. It is trivial to fit n points via one-hidden-layer ReLU networks with $\mathcal{O}(n)$ parameters. However, to prove Proposition 4.14, we need to fit n points with much fewer parameters, which is the main difficulty of our proof. Our proof below is mainly based on the "bit extraction" technique and the idea of function compositions.

Proof of Proposition 4.14. Set $J = \lceil 2s\log_2(NL+1)\rceil \in \mathbb{N}^+$. For each $\xi_i \in [0,1]$, there exist $\xi_{i,1}, \xi_{i,2}, \cdots, \xi_{i,J} \in \{0,1\}$ such that

$$\left|\xi_i - \mathrm{bin}\,0.\xi_{i,1}\xi_{i,2}\cdots\xi_{i,J}\right| \le 2^{-J}, \quad \text{for } i = 0,1,\cdots,N^2L^2-1.$$

By Theorem 3.4, there exist

$$\phi_1, \phi_2, \cdots, \phi_J \in \mathcal{NN}(\text{width} \le 8N+6;\ \text{depth} \le 5L+7)$$

such that

$$\phi_j(i) = \xi_{i,j}, \quad \text{for } i = 0,1,\cdots,N^2L^2-1 \text{ and } \ j = 1,2,\cdots,J.$$

It follows that, for $i = 0,1,\cdots,N^2L^2-1$,

$$\begin{aligned} \Big|\sum_{j=1}^{J} 2^{-j}\phi_j(i) - \xi_i\Big| &= \Big|\sum_{j=1}^{J} 2^{-j}\xi_{i,j} - \xi_i\Big| \\ &= \left|\mathrm{bin}\,0.\xi_{i,1}\xi_{i,2}\cdots\xi_{i,J} - \xi_i\right| \le 2^{-J} \le N^{-2s}L^{-2s}, \end{aligned} \tag{4.25}$$

where the last inequality comes from

$$2^{-J} = 2^{-\lceil 2s\log_2(NL+1)\rceil} \le 2^{-2s\log_2(NL+1)} = (NL+1)^{-2s} \le N^{-2s}L^{-2s}.$$

Recall that

$$\begin{aligned} J = \lceil 2s\log_2(NL+1)\rceil &\le 2s\big(1+\log_2(NL+1)\big) \le 2s\big(1+\log_2(2N)+\log_2 L\big) \\ &\le 2s\big(1+\log_2(2N)\big)\big(1+\log_2 L\big) \le 2s\lceil\log_2(4N)\rceil\lceil\log_2(2L)\rceil, \end{aligned}$$

and $\phi_j \in \mathcal{NN}(\text{width} \le 8N+6;\ \text{depth} \le 5L+7)$ for each j. Then one could use the network architecture in Figure 4.13 to implement a function $\widetilde{\phi}$ such that

$$\widetilde{\phi}(i) = \sum_{j=1}^{J} 2^{-j}\phi_j(i), \quad \text{for } i = 0, 1, \cdots, N^2L^2 - 1.$$

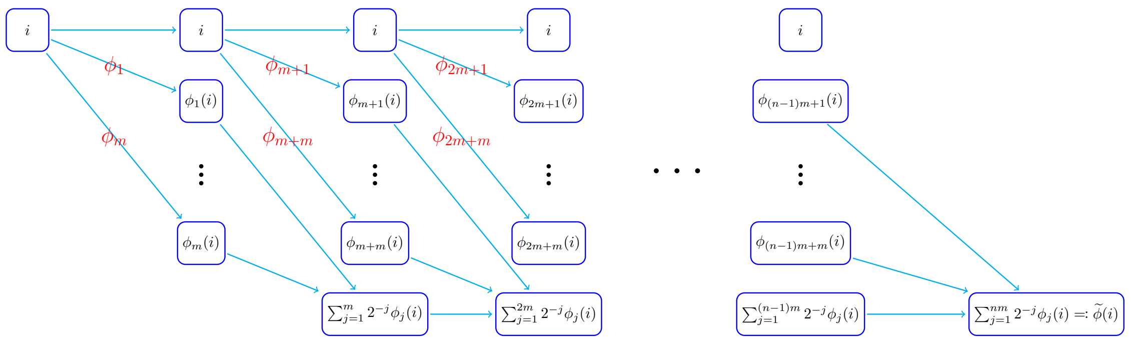


Figure 4.13: An illustration of the network architecture implementing $\widetilde{\phi}(i) = \sum_{j=1}^{J} 2^{-j}\phi_j(i)$ for any $i \in \{0, 1, \cdots, N^2L^2 - 1\}$. We assume $J = mn$, where $m = 2s\lceil \log_2(4N)\rceil$ and $n = \lceil \log_2(2L)\rceil$, since we can set $\phi_{J+1} = \cdots = \phi_{nm} = 0$ if $J < nm$.

Clearly, the network architecture in Figure 4.13 is with width

$$\begin{aligned}(8N+6)m + (1+m+1) &= (8N+6)2s\lceil \log_2(4N)\rceil + \big(2s\lceil \log_2(4N)\rceil + 2\big)\\ &\le 16s(N+1)\log_2(8N)\end{aligned}$$

and depth

$$\big((5L+7)+1\big)n = \big((5L+7)+1\big)\lceil \log_2(2L)\rceil \le (5N+8)\log_2(4L).$$

Finally, we define

$$\phi(x) := \min\big\{\sigma\big(\widetilde{\phi}(x)\big), 1\big\} = \min\big\{\max\{0, \widetilde{\phi}(x)\}, 1\big\}, \quad \text{for any } x \in \mathbb{R}.$$

See Figure 4.14 for the network architecture implementing ϕ. Then $0 \le \phi(x) \le 1$ for any $x \in \mathbb{R}$ and ϕ can be implemented by a ReLU network with width $16s(N+1)\log_2(8N)$ and depth $(5L+8)\log_2(4L) + 3 \le 5(L+2)\log_2(4L)$.

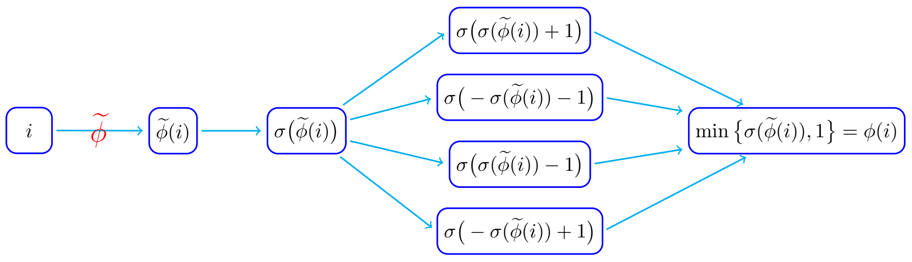

i
$\widetilde{\phi}$
$\widetilde{\phi}(i)$
$\sigma(\widetilde{\phi}(i))$
$\sigma(\sigma(\widetilde{\phi}(i))+1)$
$\sigma(-\sigma(\widetilde{\phi}(i))-1)$
$\sigma(\sigma(\widetilde{\phi}(i))-1)$
$\sigma(-\sigma(\widetilde{\phi}(i))+1)$
$\min\{\sigma(\widetilde{\phi}(i)),1\}=\phi(i)$

Figure 4.14: An illustration of the network architecture implementing the desired function ϕ for $i=0,1,\cdots,N^2L^2-1$, based on the fact $\min\{x_1,x_2\}=\frac{x_1+x_2-|x_1-x_2|}{2}=\frac{\sigma(x_1+x_2)-\sigma(-x_1-x_2)-\sigma(x_1-x_2)-\sigma(-x_1+x_2)}{2}$.

Note that

$$\widetilde{\phi}(i)=\sum_{j=1}^{J}2^{-j}\phi_j(i)=\sum_{j=1}^{J}2^{-j}\xi_{i,j}=\mathrm{bin}\,0.\xi_{i,1}\xi_{i,2}\cdots\xi_{i,J}\in[0,1],$$

for $i=0,1,\cdots,N^2L^2-1$, implying

$$|\phi(i)-\xi_i|=\left|\min\left\{\sigma\big(\widetilde{\phi}(i)\big),1\right\}-\xi_i\right|=|\widetilde{\phi}(i)-\xi_i|=\left|\sum_{j=1}^{J}2^{-j}\xi_{i,j}-\xi_i\right|\le N^{-2s}L^{-2s},$$

where the last inequality comes from Equation (4.25). So the proof is complete. □

4.4 Optimality of approximation by networks

In this section, we will study the best possible approximation errors for several function spaces approximated by ReLU networks. To this end, we adopt the method in [38, 52, 53, 58, 59, 60] via studying the connection between the approximation error and VC-dimension. Thus, let us first present the definitions of VC-dimension and related concepts.

Definition 4.15 (Growth function, VC-dimension, Shattering). Let H be a class of functions mapping from a general domain $\mathcal{X}$ to $\{0,1\}$. For any $m\in\mathbb{N}^+$, we define the growth function of H as

$$\Pi_H(m) \coloneqq \max_{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\in\mathcal{X}} \left|\left\{\left[h(\boldsymbol{x}_1),h(\boldsymbol{x}_2),\cdots,h(\boldsymbol{x}_m)\right]\in\{0,1\}^m : h\in H\right\}\right|,$$

where $|S|$ denotes the size of a set S.

We say H shatters the set $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}\subseteq\mathcal{X}$ if

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

The Vapnik-Chervonenkis (VC) dimension of H, denoted by $\mathrm{VCDim}(H)$, is the size of the largest shattered set, namely, the largest m such that $\Pi_H(m) = 2^m$. By convention, $\mathrm{VCDim}(H)=\infty$ if $\Pi_H(m)=2^m$ for all $m\in\mathbb{N}^+$.

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 $\mathrm{VCDim}(\mathscr{F})\coloneqq\mathrm{VCDim}(\mathcal{T}\circ\mathscr{F})$, where

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

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).

Definition 4.16. Let $Q(\boldsymbol{x}_0,\eta)\subseteq\mathbb{R}^d$ denote the closed cube with center $\boldsymbol{x}_0$ and sidelength η. For any cube $Q=Q(\boldsymbol{x}_0,\eta)$, rQ denote the closed cube satisfying two conditions: 1) rQ has the same center as Q; 2) the sidelength of rQ is equal to the multiplication of r and that of Q.

4.4.1 Hölder continuous functions

Let us first consider the Hölder continuous function space $\text{Hölder}([0,1]^d,\alpha,\lambda)$. Without loss of generality, we assume $\lambda=1$. Theorem 4.17 below shows that the best possible approximation error of functions in $\text{Hölder}([0,1]^d,\alpha,1)$ approximated by functions in $\mathscr{F}$ is bounded by a formula characterized by $\mathrm{VCDim}(\mathscr{F})$.

Theorem 4.17. *Given any $\varepsilon \in (0, 2/9)$ and a function set $\mathscr{F}$ with all elements defined on $[0,1]^d$, if*

$$\inf_{\phi\in\mathscr{F}} \|\phi - f\|_{L^\infty([0,1]^d)} \le \varepsilon, \quad \text{for any } f \in \text{Hölder}([0,1]^d, \alpha, 1), \tag{4.26}$$

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

This theorem investigates the connection between VC-dimension of $\mathscr{F}$ and the approximation errors of functions in $\text{Hölder}([0,1]^d, \alpha, 1)$ approximated by elements of $\mathscr{F}$. In other words, the best possible approximation error is controlled by $\text{VCDim}(\mathscr{F})^{-\alpha/d}/9$. A typical application of this theorem is to prove the optimality of approximation errors when using ReLU networks to approximate functions in $\text{Hölder}([0,1]^d, \alpha, 1)$. It is shown in [4] that VC-dimension of ReLU networks with a fixed architecture with W parameters and L layers has an upper bound $\mathcal{O}(WL\ln W)$. It follows that VC-dimension of ReLU networks with width N and depth L is bounded by $\mathcal{O}(N^2L\cdot L\cdot\ln(N^2L)) \le \mathcal{O}(N^2L^2\ln(NL))$. That is, $\text{VCDim}(\mathscr{F}) \le \mathcal{O}(N^2L^2\ln(NL))$, where

$$\mathscr{F} = \mathcal{NN}(\#\text{input} = d;\ \text{width} \le N;\ \text{depth} \le L;\ \#\text{output} = 1).$$

We 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

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

for any $N, L \in \mathbb{N}^+$. Then, by Theorem 4.17 and Corollary 4.7, we have

$$\underbrace{C_1(\alpha,d)\cdot\Big(N^2L^2\ln(NL)\Big)^{-\alpha/d} \le}_{\text{implied by Theorem 4.17}} \mathcal{E}_{\alpha,d}(N,L) \underbrace{\le\ C_2(\alpha,d)\cdot\Big(N^2L^2\Big)^{-\alpha/d}}_{\text{shown in Corollary 4.7}},$$

where $C_1(\alpha, d)$ and $C_2(\alpha, d)$ are two positive constants determined by α and d, and $C_2(\alpha, d)$ can be **explicitly** represented. Therefore, the approximation error in

Corollary 4.7 is nearly optimal.

Now let us present the detailed proof of Theorem 4.17.

Proof of Theorem 4.17. 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 (4.26) 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.

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

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

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

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

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

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 (4.26), 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 measurable 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

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

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

$$|\phi_\chi(\boldsymbol{x}) - f_\chi(\boldsymbol{x})| \le \tfrac{82}{81}\varepsilon, \quad \text{for any } \chi \in \mathscr{B} \text{ and } \boldsymbol{x} \in [0,1]^d \backslash \mathcal{H}. \tag{4.27}$$

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 $\boldsymbol{x} \in \frac{1}{10}Q_{\boldsymbol{\beta}}$,

$$|f_\chi(\boldsymbol{x})| = \zeta_{Q_{\boldsymbol{\beta}}}(\boldsymbol{x}) \ge \tfrac{9}{10}\zeta_{Q_{\boldsymbol{\beta}}}(\boldsymbol{x}_{Q_{\boldsymbol{\beta}}}) = \tfrac{9}{10}\big(\tfrac{1}{2\lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor}\big)^\alpha/2 \ge \tfrac{81}{80}\varepsilon, \tag{4.28}$$

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

If follows from $\mu\big((\frac{1}{10}Q_{\boldsymbol{\beta}})\backslash\mathcal{H}\big) > 0$ that $(\frac{1}{10}Q_{\boldsymbol{\beta}})\backslash\mathcal{H}$ is not empty for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$. Thus, by Equation (4.27) and (4.28), for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-$

$1\}^d$ and each $\chi \in \mathscr{B}$, there exists $\boldsymbol{x}_{\boldsymbol{\beta}} \in (\frac{1}{10}Q_{\boldsymbol{\beta}})\backslash\mathcal{H}$ such that

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

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

$$\mathrm{VCDim}(\mathscr{F}) \geq \mathrm{VCDim}\big(\{\phi_\chi : \chi \in \mathscr{B}\}\big) \geq K^d = \lfloor (9\varepsilon/2)^{-1/\alpha} \rfloor^d \geq (9\varepsilon)^{-d/\alpha}, \quad (4.29)$$

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

4.4.2 Smooth functions

Next, let us consider another function space $C_u^s([0,1]^d)$, which is the closed unit ball of the smooth function space $C^s([0,1]^d)$. Theorem 4.18 below shows that the best possible approximation error of functions in $C_u^s([0,1]^d)$ approximated by functions in $\mathscr{F}$ is bounded by a formula characterized by $\mathrm{VCDim}(\mathscr{F})$.

Theorem 4.18. *Given any $s, d \in \mathbb{N}^+$, there exists a small positive constant $C_{s,d}$ determined by s and d such that: For any $\varepsilon \in \big(0,\, (2^d C_{s,d})^{s/d}\big]$ and a function set $\mathscr{F}$ with all elements defined on $[0,1]^d$, if*

$$\inf_{\phi \in \mathscr{F}} \|\phi - f\|_{L^\infty([0,1]^d)} \leq \varepsilon, \quad \text{for any } f \in C_u^s([0,1]^d), \quad (4.30)$$

then $\mathrm{VCDim}(\mathscr{F}) \geq C_{s,d}\,\varepsilon^{-d/s}$. [4]

This theorem demonstrates the connection between VC-dimension of $\mathscr{F}$ and the approximation error using elements of $\mathscr{F}$ to approximate functions in $C_u^s([0,1]^d)$. To be precise, the best possible approximation error is controlled by $\mathcal{O}\big(\mathrm{VCDim}(\mathscr{F})^{-s/d}\big)$.

[4] In fact, $C_{s,d}$ can be expressed by s and d with a **explicitly** formula as we remark in the proof of this theorem. However, the formula may be very complicated.

A typical application of this theorem is to prove the optimality of approximation errors when using ReLU networks to approximate functions in $C_u^s([0,1]^d)$. It is shown in [4] that VC-dimension of ReLU networks with a fixed architecture with W parameters and L layers has an upper bound $\mathcal{O}(WL\ln W)$. It follows that VC-dimension of ReLU networks with width N and depth L is bounded by $\mathcal{O}(N^2L\cdot L\cdot\ln(N^2L))\le\mathcal{O}(N^2L^2\ln(NL))$. That is, $\text{VCDim}(\mathscr{F})\le\mathcal{O}(N^2L^2\ln(NL))$, where

$$\mathscr{F}=\mathcal{NN}(\#\text{input}=d;\ \text{width}\le N;\ \text{depth}\le L;\ \#\text{output}=1).$$

We denote the best approximation error of functions in $C_u^s([0,1]^d)$ approximated by ReLU networks with width N and depth L as

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

for any $N,L\in\mathbb{N}^+$, where $C_u^s([0,1]^d)$ is the closed unit ball of $C^s([0,1]^d)$. Then, by Theorem 4.18 and Corollary 4.12, we have

$$\underbrace{C_1(s,d)\cdot\Big(N^2L^2\ln(NL)\Big)^{-s/d}\le}_{\text{implied by Theorem 4.18}}\ \mathcal{E}_{s,d}(N,L)\ \underbrace{\le\ C_2(s,d)\cdot\Big(\tfrac{N^2L^2}{(\ln N\ln L)^2}\Big)^{-s/d}}_{\text{shown in Corollary 4.12}},$$

where $C_1(s,d)$ and $C_2(s,d)$ are two positive constants in s and d, and $C_2(s,d)$ can be **explicitly** expressed. Therefore, the approximation errors in Theorem 4.11 and Corollary 4.12 are nearly optimal.

Now let us present the detailed proof of Theorem 4.18.

Proof of Theorem 4.18. To find a subset of $\mathscr{F}$ shattering $\mathcal{O}(\varepsilon^{-d/s})$ points in $[0,1]^d$, we divide the proof into two steps.

- Construct $\{f_\chi:\chi\in\mathscr{B}\}\subseteq C_u^s([0,1]^d)$ that scatters $\mathcal{O}(\varepsilon^{-d/s})$ 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 (4.30) such that $\{\phi_\chi : \chi \in \mathscr{B}\} \subseteq \mathscr{F}$ also shatters $\mathcal{O}(\varepsilon^{-d/s})$ points.

The details of these two steps can be found below.

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

Let $K = \mathcal{O}(\varepsilon^{-1/s})$ be a positive integer determined later and divide $[0,1]^d$ into K^d sub-cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta}}$ as follows.

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

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

There exists a "bump function" $\widetilde{g} \in C^\infty(\mathbb{R}^d)$ such that $\widetilde{g}(\mathbf{0}) = 1$ and $\widetilde{g}(\boldsymbol{x}) = 0$ for $\|\boldsymbol{x}\|_2 \geq 1/3$. For example, we can define $\widetilde{g}$ as

$$\widetilde{g}(\boldsymbol{x}) \coloneqq \begin{cases} \exp\big(\frac{1}{\|3\boldsymbol{x}\|_2^2 - 1} + 1\big), & \text{if } \|\boldsymbol{x}\|_2 < 1/3, \\ 0, & \text{otherwise,} \end{cases}$$

where $\exp(x) = e^x$ for any $x \in \mathbb{R}$ and $e \approx 2.7$ is the natural logarithmic base. Then, we have $g \coloneqq \widetilde{g}/\widetilde{C}_{s,d} \in C_u^s([0,1]^d)$ by setting $\widetilde{C}_{s,d} \coloneqq \|\widetilde{g}\|_{C^s([0,1]^d)}$.

Define

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

and

$$g_{\boldsymbol{\beta}} \coloneqq K^{-s} g\big(K(\boldsymbol{x} - \boldsymbol{x}_{Q_{\boldsymbol{\beta}}})\big), \qquad \text{for each } \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d,$$

where $\boldsymbol{x}_{Q_{\boldsymbol{\beta}}}$ is the center of $Q_{\boldsymbol{\beta}}$. Then, we have

$$\{\boldsymbol{x} : g_{\boldsymbol{\beta}}(\boldsymbol{x}) \neq 0\} \subseteq \mathcal{B}\big(\boldsymbol{x}_{Q_{\boldsymbol{\beta}}}, \tfrac{1}{3K}\big) \subseteq \tfrac{2}{3} Q_{\boldsymbol{\beta}}, \quad \text{for each } \boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d.$$

Next, for each $\chi \in \mathscr{B}$, we can define f_χ via

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

Then $f_\chi \in C_u^s([0,1]^d)$ for each $\chi \in \mathscr{B}$, since it satisfies the following two conditions.

- By the definition of $g_{\boldsymbol{\beta}}$ and χ, we have

$$\{\boldsymbol{x} : \chi(\boldsymbol{\beta}) g_{\boldsymbol{\beta}}(\boldsymbol{x}) \neq 0\} \subseteq \tfrac{2}{3} Q_{\boldsymbol{\beta}}, \quad \text{for each } \boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d.$$

- For any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$, $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$, and $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \leq s$,

$$\partial^{\boldsymbol{\alpha}} f_\chi(\boldsymbol{x}) = \chi(\boldsymbol{\beta}) \partial^{\boldsymbol{\alpha}} g_{\boldsymbol{\beta}}(\boldsymbol{x}) = \chi(\boldsymbol{\beta}) K^{-s} K^{\|\alpha\|_1} \partial^{\boldsymbol{\alpha}} g\big(K(\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{\beta}})\big),$$

  implying $|\partial^{\boldsymbol{\alpha}} f_\chi(\boldsymbol{x})| = \big|K^{-(s-\|\alpha\|_1)} \partial^{\boldsymbol{\alpha}} g\big(K(\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{\beta}})\big)\big| \leq 1$.

It is easy to check that $\{f_\chi : \chi \in \mathscr{B}\} \subseteq C_u^s([0,1]^d)$ 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 $\mathcal{O}(\varepsilon^{-d/s})$ points.

By Equation (4.30), for each $\chi \in \mathscr{B}$, there exists $\phi_\chi \in \mathscr{F}$ such that

$$\|\phi_\chi - f_\chi\|_{L^\infty([0,1]^d)} \leq \varepsilon + \varepsilon/2.$$

Let $\mu(\cdot)$ denote the Lebesgue measure of a measurable 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

$$|\phi_\chi(\boldsymbol{x}) - f_\chi(\boldsymbol{x})| \leq \tfrac{3}{2}\varepsilon, \quad \text{for any } \boldsymbol{x} \in [0,1]^d \backslash \mathcal{H}_\chi.$$

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

$$|\phi_\chi(\boldsymbol{x}) - f_\chi(\boldsymbol{x})| \leq \tfrac{3}{2}\varepsilon, \quad \text{for any } \chi \in \mathscr{B} \text{ and } \boldsymbol{x} \in [0,1]^d \backslash \mathcal{H}. \tag{4.31}$$

Clearly, there exists $r \in (0,1)$ such that

$$g_{\boldsymbol{\beta}}(\boldsymbol{x}) \geq \tfrac{1}{2} g_{\boldsymbol{\beta}}(\boldsymbol{x}_{Q_{\boldsymbol{\beta}}}), \quad \text{for any } \boldsymbol{x} \in rQ_{\boldsymbol{\beta}},$$

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

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

$$|f_\chi(\boldsymbol{x}_{\boldsymbol{\beta}})| = g_{\boldsymbol{\beta}}(\boldsymbol{x}_{\boldsymbol{\beta}}) \geq \tfrac{1}{2} g_{\boldsymbol{\beta}}(\boldsymbol{x}_{Q_{\boldsymbol{\beta}}}) = \tfrac{1}{2} K^{-s} g(\boldsymbol{0}) = \tfrac{1}{2} K^{-s}/\widetilde{C}_{s,d} \geq 2\varepsilon, \tag{4.32}$$

where the last inequality is attained by setting $K = \lfloor (4\varepsilon \widetilde{C}_{s,d})^{-1/s} \rfloor$. Since our proof is invalid when $K = 0$, it is necessary to guarantee $K = \lfloor (4\varepsilon \widetilde{C}_{s,d})^{-1/s} \rfloor \geq 1$, which will be verified later.

By Equation (4.31) and (4.32), we have, for each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ and each $\chi \in \mathscr{B}$,

$$|f_\chi(\boldsymbol{x}_{\boldsymbol{\beta}})| \geq 2\varepsilon > \tfrac{3}{2}\varepsilon \geq |f_\chi(\boldsymbol{x}_{\boldsymbol{\beta}}) - \phi_\chi(\boldsymbol{x}_{\boldsymbol{\beta}})|.$$

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

$$\text{VCDim}(\mathscr{F}) \geq \text{VCDim}\big(\{\phi_\chi : \chi \in \mathscr{B}\}\big) \geq K^d = \lfloor (4\varepsilon \widetilde{C}_{s,d})^{-1/s} \rfloor^d \geq 2^{-d}(4\varepsilon \widetilde{C}_{s,d})^{-d/s},$$

where the last inequality comes from the fact $\lfloor x \rfloor \geq x/2$ for any $x \in [1,\infty)$.

Finally, set

$$C_{s,d} = 2^{-d}(4\widetilde{C}_{s,d})^{-d/s} = 2^{-d}\big(4\|\widetilde{g}\|_{C^s([0,1]^d)}\big)^{-d/s}.$$

This means $C_{s,d}$ can be computed by an explicit mathematical formula based on the function $\widetilde{g}$ defined previously. Moreover, we have

$$\mathrm{VCDim}(\mathscr{F}) \geq 2^{-d}(4\varepsilon\widetilde{C}_{s,d})^{-d/s} = C_{s,d}\,\varepsilon^{-d/s}$$

and

$$K = \lfloor (4\varepsilon\widetilde{C}_{s,d})^{-1/s} \rfloor = \lfloor \varepsilon^{-1/s}(2^d C_{s,d})^{1/d} \rfloor \geq 1,$$

where the last inequality comes from $\varepsilon \in (0, (2^d C_{s,d})^{s/d}]$. So we finish the proof. $\square$

Chapter 5

Approximation by Floor-ReLU networks

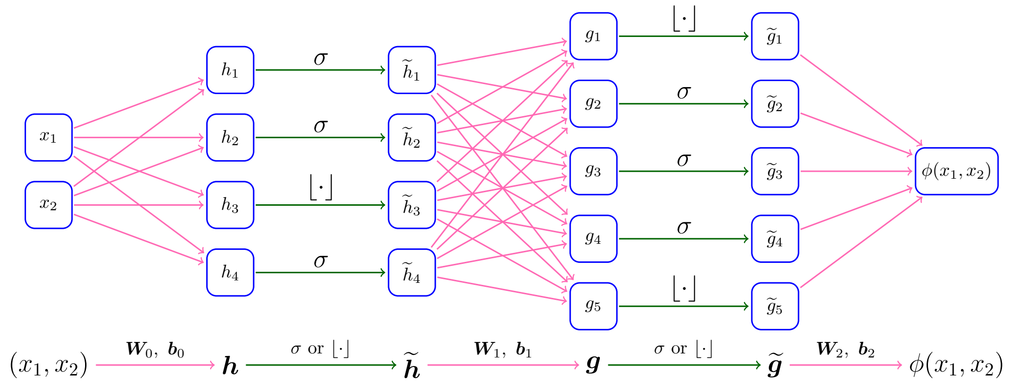
As shown in Section 4.1, an exponential approximation error $O(N^{-L})$ can be achieved when using ReLU networks with width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$ to approximate polynomials on $[0,1]^d$. But such an exponential error is not true for general function spaces as discussed in Section 4.4. The limitation of ReLU networks motivates us to explore other types of network architectures to admit (nearly) exponential approximation errors.

In particular, we introduce new networks built with either Floor ($\lfloor x \rfloor$) or ReLU ($\max\{0,x\}$) as the activation function[①] in each neuron. We call such networks Floor-ReLU networks. See Figure 5.1 for an example. We will prove in this chapter that Floor-ReLU networks with fixed architectures can attain nearly exponential approximation errors for approximating (Hölder) continuous functions on $[0,1]^d$.

5.1 Main theorem and its proof

In Theorem 5.1 below, we show by construction that Floor-ReLU networks, with fixed architectures, with width $\max\{d,\, 5N+13\}$ and depth $64dL+3$ can uniformly approximate an arbitrary continuous function f on $[0,1]^d$ with a nearly exponential

[①]Our results can be easily generalized to Ceiling-ReLU networks, namely, feed-forward fully connected neural networks with either Ceiling ($\lceil x \rceil$) or ReLU ($\max\{0,x\}$) as the activation function in each neuron.



x_1
x_2
h_1
h_2
h_3
h_4
σ
σ
$\lfloor\cdot\rfloor$
σ
$\widetilde{h}_1$
$\widetilde{h}_2$
$\widetilde{h}_3$
$\widetilde{h}_4$
g_1
g_2
g_3
g_4
g_5
$\lfloor\cdot\rfloor$
σ
σ
σ
$\lfloor\cdot\rfloor$
$\widetilde{g}_1$
$\widetilde{g}_2$
$\widetilde{g}_3$
$\widetilde{g}_4$
$\widetilde{g}_5$
$\phi(x_1,x_2)$
$(x_1,x_2) \xrightarrow{\boldsymbol{W}_0,\ \boldsymbol{b}_0} \boldsymbol{h} \xrightarrow{\sigma \text{ or } \lfloor\cdot\rfloor} \widetilde{\boldsymbol{h}} \xrightarrow{\boldsymbol{W}_1,\ \boldsymbol{b}_1} \boldsymbol{g} \xrightarrow{\sigma \text{ or } \lfloor\cdot\rfloor} \widetilde{\boldsymbol{g}} \xrightarrow{\boldsymbol{W}_2,\ \boldsymbol{b}_2} \phi(x_1,x_2)$

Figure 5.1: An example of a Floor-ReLU network with width 5 and depth 2.

approximation error $\omega_f(\sqrt{d}\,N^{-\sqrt{L}})+2\omega_f(\sqrt{d})N^{-\sqrt{L}}$.

Theorem 5.1. *Given any $N, L, d \in \mathbb{N}^+$, there exists a fixed Floor-ReLU network architecture with width* $\max\{d,\,5N+13\}$ *and depth* $64dL+3$ *such that: For any continuous function $f \in C([0,1]^d)$, there exists a function ϕ, implemented by this Floor-ReLU network architecture with proper parameters, satisfying*

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

With Theorem 5.1, we have an immediate corollary.

Corollary 5.2. *Given any $\bar{N}, \bar{L}, d \in \mathbb{N}^+$, there exists a fixed Floor-ReLU network architecture with width $\bar{N}$ and depth $\bar{L}$ such that: For any continuous function $f \in C([0,1]^d)$, there exists a function ϕ, implemented by this Floor-ReLU network architecture with proper parameters, satisfying*

$$|\phi(\boldsymbol{x})-f(\boldsymbol{x})| \le \omega_f\Big(\sqrt{d}\,\big\lfloor\tfrac{\bar{N}-13}{5}\big\rfloor^{-\sqrt{\lfloor\frac{\bar{L}-3}{64d}\rfloor}}\Big)+2\omega_f(\sqrt{d})\big\lfloor\tfrac{\bar{N}-13}{5}\big\rfloor^{-\sqrt{\lfloor\frac{\bar{L}-3}{64d}\rfloor}},$$

for any $\boldsymbol{x}\in[0,1]^d$ and $\bar{N},\bar{L}\in\mathbb{N}^+$ with $\bar{N}\ge\max\{d,18\}$ and $\bar{L}\ge 64d+3$.

We would like to remark that the Floor-ReLU network architectures in Theorem 5.1 and Corollary 5.2 are independent of the target function f. That is, only the values of the parameters rely on the target function f. In particular, the choice of activation functions (Floor or ReLU) in each neuron is independent of the target function f.

In Theorem 5.1, the error in $\omega_f(\sqrt{d}\,N^{-\sqrt{L}})$ implicitly depends on N and L through the modulus of continuity of f, while the error in $2\omega_f(\sqrt{d})N^{-\sqrt{L}}$ is explicit in N and L. Simplifying the implicit approximation error to make it explicitly depending on N and L is challenging in general. However, if f is a Hölder continuous function on $[0,1]^d$ of order $\alpha \in (0,1]$ with a constant λ, i.e., $f \in \text{Hölder}([0,1]^d, \alpha, \lambda)$, then we have

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \le \lambda \|\boldsymbol{x} - \boldsymbol{y}\|_2^\alpha, \quad \text{for any } \boldsymbol{x}, \boldsymbol{y} \in [0,1]^d, \tag{5.1}$$

implying $\omega_f(r) \le \lambda r^\alpha$ for any $r \ge 0$. Therefore, in the case of Hölder continuous functions, the approximation error is simplified to $3\lambda d^{\alpha/2} N^{-\alpha\sqrt{L}}$ as shown in the following corollary. In the special case of Lipschitz continuous functions with a Lipschitz constant λ, the approximation error is simplified to $3\lambda\sqrt{d}N^{-\sqrt{L}}$.

Corollary 5.3. *Given any $N, L, d \in \mathbb{N}^+$, there exist a fixed Floor-ReLU network architecture with width* $\max\{d, 5N+13\}$ *and depth* $64dL+3$ *such that: For any function* $f \in \text{Hölder}([0,1]^d, \alpha, \lambda)$, *there exists a function ϕ, implemented by this Floor-ReLU network architecture with proper parameters, satisfying*

$$|\phi(\boldsymbol{x}) - f(\boldsymbol{x})| \le 3\lambda d^{\alpha/2} N^{-\alpha\sqrt{L}}, \quad \text{for any } \boldsymbol{x} \in [0,1]^d.$$

First, Theorem 5.1 and Corollary 5.3 show that the approximation capacity of deep networks for continuous functions can be nearly exponentially improved by increasing the network depth, and the approximation error can be explicitly characterized in terms of the width $\mathcal{O}(N)$ and depth $\mathcal{O}(L)$. Second, this new class of networks overcomes the curse of dimensionality in the approximation power when the modulus of continuity is moderate, since the approximation order is essentially $\omega_f(\sqrt{d}N^{-\sqrt{L}})$. Finally, applying piecewise constant and integer-valued functions as activation functions and integer numbers as parameters has been explored in the study of quantized neural networks [6, 26, 61] with efficient training algorithms for low computational complexity [56]. The floor function ($\lfloor x \rfloor$) is a piecewise constant function and can be easily implemented numerically at very little cost. Hence, the evaluation of the proposed network could be efficiently implemented in practical computation. Though there might not be an existing optimization algorithm to identify an approximant with an approximation error, Theorem 5.1 and Corollary 5.3 can provide an expected accuracy before a learning task and how much the current optimization algorithms could be improved. Designing an efficient optimization algorithm for Floor-ReLU networks will be left as future work with several possible directions discussed later.

In particular, we let $N = 2$ and $L = W$ in Theorem 5.1, then the width is $\max\{d, 23\}$, the depth is $64dW+3$, and the total number of parameters is bounded by $\mathcal{O}\left(\max\{d^2, 23^2\}(64dW+3)\right) = \mathcal{O}(W)$. Therefore, we can prove Corollary 5.4 below stating that Floor-ReLU networks can provide a nearly exponential approximation error in terms of the number of parameters.

Corollary 5.4. *Given any $W, d \in \mathbb{N}^+$, there exists a fixed Floor-ReLU network architecture with $\mathcal{O}(W)$ parameters, width $\max\{d,\, 23\}$, and depth $64dW+3$, such that: For any continuous function $f \in C([0,1]^d)$, there exists a function ϕ, implemented by this Floor-ReLU network architecture with proper parameters, satisfying*

$$|\phi(\boldsymbol{x}) - f(\boldsymbol{x})| \le \omega_f(\sqrt{d}\,2^{-\sqrt{W}}) + 2\omega_f(\sqrt{d})2^{-\sqrt{W}}, \quad \text{for any } \boldsymbol{x} \in [0,1]^d.$$

We would like to point out that the derivative of Floor is zero almost everywhere, which may lead to the failure of the backpropagation algorithm. To overcome this, we propose three possible methods as follows.

- First, we can consider gradient-free optimization methods, *e.g.*, particle swarm optimization [30], consensus-based optimization [11, 50].

- The second method is to apply optimization algorithms for quantized networks that also have piecewise constant activation functions [6, 9, 26, 36, 56, 61]. For example, an empirical method is to use a straight through estimator (STE) by setting the incoming gradients to the activation function equal to its outgoing gradients, disregarding the derivative of the activation function itself.

- The final method is to use the linear combination of ReLU and Floor, *i.e.*, $p\sigma(x)+(1-p)\lfloor x \rfloor$ for $p \in (0,1)$, to replace Floor ($\lfloor x \rfloor$) to avoid zero derivative. Similar to Theorem 5.1 and Corollary 5.3, the nearly exponential errors can be attained with the new activation functions (σ and $p\sigma(x)+(1-p)\lfloor x \rfloor$).

The proof of Theorem 5.1 is an immediate result of Theorem 5.5 below.

Theorem 5.5. *Given a continuous function $f \in C([0,1]^d)$, for any $N, L \in \mathbb{N}^+$, there exists a function ϕ implemented by a Floor-ReLU network, with a fixed architecture independent of f, with width* $\max\{d,\, 2N^2+5N\}$ *and depth* $7dL^2+3$ *such that*

$$|\phi(\boldsymbol{x}) - f(\boldsymbol{x})| \le \omega_f(\sqrt{d}\, N^{-L}) + 2\omega_f(\sqrt{d})2^{-NL}, \quad \text{for any } \boldsymbol{x} \in [0,1]^d.$$

Theorem 5.5 will be proved later in this section. Now let us prove Theorem 5.1 based on Theorem 5.5.

Proof of Theorem 5.1. Given any $N, L \in \mathbb{N}^+$, there exist $\widetilde{N}, \widetilde{L} \in \mathbb{N}^+$ with $\widetilde{N} \ge 2$ and $\widetilde{L} \ge 3$ such that

$$(\widetilde{N}-1)^2 \le N < \widetilde{N}^2 \quad \text{and} \quad (\widetilde{L}-1)^2 \le 4L < \widetilde{L}^2.$$

By Theorem 5.5, there exists a function ϕ implemented by a Floor-ReLU network, with a fixed architecture independent of f, with width $\max\{d,\, 2\widetilde{N}^2+5\widetilde{N}\}$ and depth $7d\widetilde{L}^2+3$ such that

$$|\phi(\boldsymbol{x}) - f(\boldsymbol{x})| \le \omega_f(\sqrt{d}\, \widetilde{N}^{-\widetilde{L}}) + 2\omega_f(\sqrt{d})2^{-\widetilde{N}\widetilde{L}}, \quad \text{for any } \boldsymbol{x} \in [0,1]^d.$$

Note that

$$2^{-\widetilde{N}\widetilde{L}} \le \widetilde{N}^{-\widetilde{L}} = (\widetilde{N}^2)^{-\frac{1}{2}\sqrt{\widetilde{L}^2}} \le N^{-\frac{1}{2}\sqrt{4L}} \le N^{-\sqrt{L}}.$$

Then we have

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

For $\widetilde{N}, \widetilde{L} \in \mathbb{N}^+$ with $\widetilde{N} \ge 2$ and $\widetilde{L} \ge 3$, we have

$$2\widetilde{N}^2 + 5\widetilde{N} \le 5(\widetilde{N}-1)^2 + 13 \le 5N + 13 \quad \text{and} \quad 7\widetilde{L}^2 \le 16(\widetilde{L}-1)^2 \le 64L.$$

Therefore, ϕ can be implemented by a Floor-ReLU network, with a fixed architecture independent of f, with width $\max\{d,\, 2\widetilde{N}^2 + 5\widetilde{N}\} \le \max\{d,\, 5N + 13\}$ and depth $7d\widetilde{L}^2 + 3 \le 64dL + 3$, as desired. So we finish the proof. □

5.2 Proof of auxiliary theorem

To prove Theorem 5.5, we first present the general ideas of the proof. Shortly speaking, we construct piecewise constant functions implemented by Floor-ReLU networks to approximate continuous functions on $[0,1]^d$. There are four key steps in our construction.

1. Normalize f as $\widetilde{f}$ satisfying $\widetilde{f}(\boldsymbol{x}) \in [0,1]$ for any $\boldsymbol{x} \in [0,1]^d$, divide $[0,1]^d$ into a set of non-overlapping cubes $\{Q_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d}$, and denote $\boldsymbol{x}_{\boldsymbol{\beta}}$ as the vertex of $Q_{\boldsymbol{\beta}}$ with minimum $\|\cdot\|_1$ norm, where K is an integer determined later. See Figure 5.2 for the illustrations of $Q_{\boldsymbol{\beta}}$ and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

2. Construct a Floor-ReLU sub-network to implement a vector-valued function $\boldsymbol{\Phi}_1 : \mathbb{R}^d \to \mathbb{R}^d$ projecting the whole cube $Q_{\boldsymbol{\beta}}$ to the index $\boldsymbol{\beta}$ for each $\boldsymbol{\beta}$, *i.e.*, $\boldsymbol{\Phi}_1(\boldsymbol{x}) = \boldsymbol{\beta}$ for all $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$.

3. Construct a Floor-ReLU sub-network to implement a function $\phi_2 : \mathbb{R}^d \to \mathbb{R}$ mapping $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for each $\boldsymbol{\beta}$, *i.e.*, $\phi_2(\boldsymbol{\beta}) \approx \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$. Then $\phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \phi_2(\boldsymbol{\beta}) \approx \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$ for any $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each

$\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, implying $\widetilde{\phi} := \phi_2 \circ \boldsymbol{\Phi}_1$ approximates $\widetilde{f}$ within an error $\mathcal{O}(\omega_f(1/K))$ on $[0,1]^d$.

4. Re-scale and shift $\widetilde{\phi}$ to obtain the desired function ϕ approximating f well and determine the final Floor-ReLU network to implement ϕ.

It is not difficult to construct Floor-ReLU networks with the desired width and depth to implement $\boldsymbol{\Phi}_1$. The most technical part is the construction of a Floor-ReLU network with the desired width and depth implementing ϕ_2, which needs the following proposition based on the "bit extraction" technique introduced in [5].

Proposition 5.6. *Given any $N, L \in \mathbb{N}^+$ and arbitrary $\theta_m \in \{0,1\}$ for $m = 1, 2, \cdots, N^L$, there exists a function ϕ implemented by a Floor-ReLU network, with a fixed architecture independent of $\theta_m \in \{0,1\}$ for $m = 1, 2, \cdots, N^L$, with width $2N+2$ and depth $7L-2$ such that*

$$\phi(m) = \theta_m, \quad \text{for } m = 1, 2, \cdots, N^L.$$

The proof of this proposition is presented in Section 5.3. It is easy to prove that the VC-dimension of Floor-ReLU networks with a constant width and depth $\mathcal{O}(L)$ has a lower bound 2^L. Such a lower bound is much larger than $\mathcal{O}(L^2)$, which is a VC-dimension upper bound of ReLU networks with the same width and depth due to Theorem 8 of [4]. This means Floor-ReLU networks are much more powerful than ReLU networks from the perspective of VC-dimension.

Now let us give the detailed proof of Theorem 5.5 as follows.

Proof of Theorem 5.5. The proof consists of four steps.

Step 1: Set up.

Assume f is not a constant function since it is a trivial case. Then $\omega_f(r) > 0$ for any $r > 0$. Clearly, $|f(\boldsymbol{x}) - f(\boldsymbol{0})| \le \omega_f(\sqrt{d})$ for any $\boldsymbol{x} \in [0,1]^d$. Define

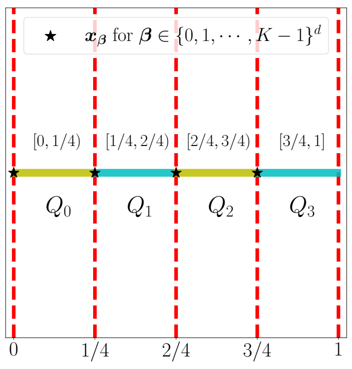
$$\widetilde{f} := \frac{f - f(\boldsymbol{0}) + \omega_f(\sqrt{d})}{2\omega_f(\sqrt{d})}. \tag{5.2}$$

It follows that $\widetilde{f}(\boldsymbol{x}) \in [0,1]$ for any $\boldsymbol{x} \in [0,1]^d$.

Set $K = N^L$, $E_{K-1} = [\frac{K-1}{K}, 1]$, and $E_k = [\frac{k}{K}, \frac{k+1}{K})$ for $k = 0, 1, \cdots, K-2$. Define

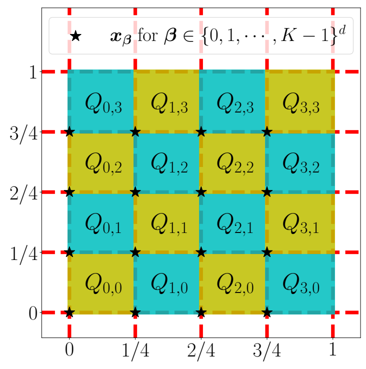
$$Q_{\boldsymbol{\beta}} \coloneqq \Big\{\boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d : x_j \in E_{\beta_j} \text{ for } j = 1, 2, \cdots, d\Big\},$$

for any $\boldsymbol{\beta} = (\beta_1, \beta_2 \cdots, \beta_d) \in \{0, 1, \cdots, K-1\}^d$. See Figure 5.2 for the examples of $Q_{\boldsymbol{\beta}}$ and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$ with for $K = 4$ and $d = 1, 2$.



★ $\boldsymbol{x}_{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$
[0, 1/4)
[1/4, 2/4)
[2/4, 3/4)
[3/4, 1]
Q_0
Q_1
Q_2
Q_3
0
1/4
2/4
3/4
1

(a)



(b)

Figure 5.2: Illustrations of $Q_{\boldsymbol{\beta}}$ and $\boldsymbol{x}_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$. (a) $K = 4,\ d = 1$. (b) $K = 4,\ d = 2$.

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

Define a step function ϕ_1 as

$$\phi_1(x) \coloneqq \big\lfloor -\sigma(-Kx + K - 1) + K - 1 \big\rfloor, \quad \text{for any } x \in \mathbb{R}.$$[②]

See Figure 5.3 for an illustration of ϕ_1 when $K = 4$. It follows from the definition of ϕ_1 that

$$\phi_1(x) = k, \quad \text{if } x \in E_k, \quad \text{for } k = 0, 1, \cdots, K-1.$$

Define

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

②If we just define $\phi_1(x) = \lfloor Kx \rfloor$, then $\phi_1(1) = K \neq K-1$ even though $1 \in E_{K-1}$.

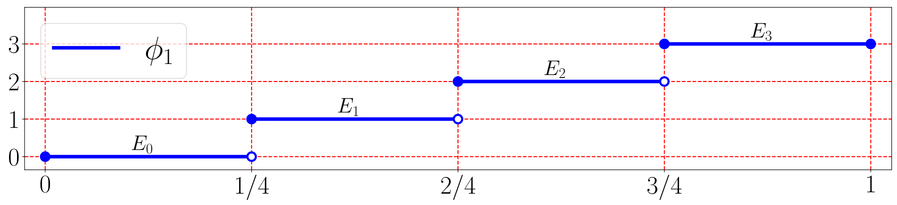


Figure 5.3: An illustration of ϕ_1 on $[0,1]$ for the case $K=4$.

Clearly, we have, for all $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and each $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$,

$$\boldsymbol{\Phi}_1(\boldsymbol{x}) = \big(\phi_1(x_1), \phi_1(x_2), \cdots, \phi_1(x_d)\big) = (\beta_1, \beta_2, \cdots, \beta_d) = \boldsymbol{\beta}.$$

Step 3: Construct ϕ_2 mapping $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ approximately to $\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})$.

Using the idea of K-ary representation, we define

$$\psi_1(\boldsymbol{x}) := 1 + \sum_{j=1}^{d} x_j K^{j-1}, \quad \text{for any } \boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d.$$

It follows that ψ_1 is a bijection from $\{0,1,\cdots,K-1\}^d$ to $\{1,2,\cdots,K^d\}$.

Given any $i \in \{1,2,\cdots,K^d\}$, there exists a unique $\boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d$ such that $i = \psi_1(\boldsymbol{\beta})$. Then define

$$\xi_i := \widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}}) \in [0,1], \quad \text{for } i = \psi_1(\boldsymbol{\beta}) \text{ and } \boldsymbol{\beta} \in \{0,1,\cdots,K-1\}^d,$$

where $\widetilde{f}$ is the normalization of f defined in Equation (5.2). It follows that there exists $\xi_{i,j} \in \{0,1\}$ for $j = 1,2,\cdots,NL$ such that

$$|\xi_i - \text{bin}\,0.\xi_{i,1}\xi_{i,2}\cdots,\xi_{i,NL}| \le 2^{-NL}, \quad \text{for } i = 1,2,\cdots,K^d.$$

By $K^d = (N^L)^d = N^{dL}$ and Proposition 5.6, there exists a function $\psi_{2,j}$ implemented by a Floor-ReLU network, with a fixed architecture independent of $\xi_{i,j}$ for all i, with width $2N+2$ and depth $7dL-2$, for each $j=1,2,\cdots,NL$, such that

$$\psi_{2,j}(i)=\xi_{i,j},\quad \text{for } i=1,2,\cdots,K^d.$$

Define

$$\psi_2 \coloneqq \sum_{j=1}^{NL} 2^{-j}\psi_{2,j} \quad \text{and} \quad \phi_2 \coloneqq \psi_2\circ\psi_1.$$

Then we have, for $i=\psi_1(\boldsymbol{\beta})$ and $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$,

$$\begin{aligned}|\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})-\phi_2(\boldsymbol{\beta})| = |\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})-\psi_2(\psi_1(\boldsymbol{\beta}))| = |\xi_i-\psi_2(i)| &= |\xi_i-\sum_{j=1}^{NL}2^{-j}\psi_{2,j}(i)| \\ &= |\xi_i-\text{bin}\,0.\xi_{i,1}\xi_{i,2}\cdots\xi_{i,NL}|\le 2^{-NL}.\end{aligned} \tag{5.3}$$

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

Define $\widetilde{\phi}\coloneqq\phi_2\circ\boldsymbol{\Phi}_1$, i.e., for any $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)\in\mathbb{R}^d$,

$$\widetilde{\phi}(\boldsymbol{x})=\phi_2\circ\boldsymbol{\Phi}_1(\boldsymbol{x})=\phi_2\big(\phi_1(x_1),\phi_1(x_2),\cdots,\phi_1(x_d)\big).$$

Note that $|\boldsymbol{x}-\boldsymbol{x}_{\boldsymbol{\beta}}|\le\frac{\sqrt{d}}{K}$ for any $\boldsymbol{x}\in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$. Then we have, for any $\boldsymbol{x}\in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$,

$$\begin{aligned}|\widetilde{f}(\boldsymbol{x})-\widetilde{\phi}(\boldsymbol{x})| &\le |\widetilde{f}(\boldsymbol{x})-\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})|+|\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})-\widetilde{\phi}(\boldsymbol{x})| \\ &\le \omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K})+|\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})-\phi_2(\boldsymbol{\Phi}_1(\boldsymbol{x}))| \\ &\le \omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K})+|\widetilde{f}(\boldsymbol{x}_{\boldsymbol{\beta}})-\phi_2(\boldsymbol{\beta})|\le\omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K})+2^{-NL},\end{aligned}$$

where the last inequality comes from Equation (5.3).

Note that $[0,1]^d=\cup_{\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d}Q_{\boldsymbol{\beta}}$. Since $\boldsymbol{x}\in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}\in\{0,1,\cdots,K-1\}^d$ are arbitrary, we have

$$|\widetilde{f}(\boldsymbol{x})-\widetilde{\phi}(\boldsymbol{x})|\le\omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K})+2^{-NL},\quad \text{for any } \boldsymbol{x}\in[0,1]^d.$$

Define

$$\phi := 2\omega_f(\sqrt{d})\widetilde{\phi} + f(\mathbf{0}) - \omega_f(\sqrt{d}).$$

By $K = N^L$ and $\omega_f(r) = 2\omega_f(\sqrt{d}) \cdot \omega_{\widetilde{f}}(r)$ for any $r \geq 0$, we have, for any $\boldsymbol{x} \in [0,1]^d$,

$$\begin{aligned} |f(\boldsymbol{x}) - \phi(\boldsymbol{x})| = 2\omega_f(\sqrt{d})\big|\widetilde{f}(\boldsymbol{x}) - \widetilde{\phi}(\boldsymbol{x})\big| &\leq 2\omega_f(\sqrt{d})\Big(\omega_{\widetilde{f}}(\tfrac{\sqrt{d}}{K}) + 2^{-NL}\Big) \\ &\leq \omega_f(\tfrac{\sqrt{d}}{K}) + 2\omega_f(\sqrt{d})2^{-NL} \\ &\leq \omega_f(\sqrt{d}\,N^{-L}) + 2\omega_f(\sqrt{d})2^{-NL}. \end{aligned}$$

It remains to determine the width and depth of the Floor-ReLU network implementing ϕ. Clearly, ϕ_2 can be implemented by the architecture in Figure 5.4.

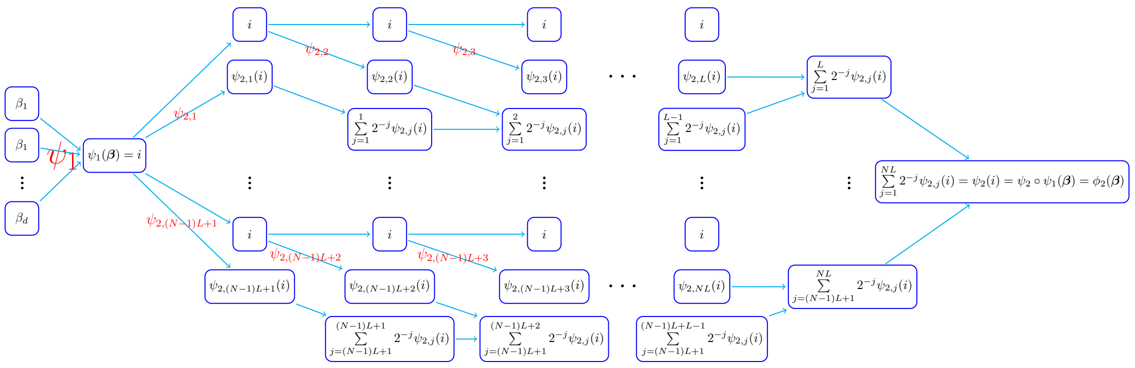

β_1
β_1
β_d
ψ_1
$\psi_1(\boldsymbol{\beta}) = i$
$\psi_{2,1}$
$\psi_{2,2}$
$\psi_{2,3}$
$\psi_{2,(N-1)L+1}$
$\psi_{2,(N-1)L+2}$
$\psi_{2,(N-1)L+3}$
i
$\psi_{2,1}(i)$
$\psi_{2,2}(i)$
$\psi_{2,3}(i)$
$\psi_{2,L}(i)$
$\sum_{j=1}^{1} 2^{-j}\psi_{2,j}(i)$
$\sum_{j=1}^{2} 2^{-j}\psi_{2,j}(i)$
$\sum_{j=1}^{L-1} 2^{-j}\psi_{2,j}(i)$
$\sum_{j=1}^{L} 2^{-j}\psi_{2,j}(i)$
$\sum_{j=1}^{NL} 2^{-j}\psi_{2,j}(i) = \psi_2(i) = \psi_2 \circ \psi_1(\boldsymbol{\beta}) = \phi_2(\boldsymbol{\beta})$
$\psi_{2,(N-1)L+1}(i)$
$\psi_{2,(N-1)L+2}(i)$
$\psi_{2,(N-1)L+3}(i)$
$\psi_{2,NL}(i)$
$\sum_{j=(N-1)L+1}^{(N-1)L+1} 2^{-j}\psi_{2,j}(i)$
$\sum_{j=(N-1)L+1}^{(N-1)L+2} 2^{-j}\psi_{2,j}(i)$
$\sum_{j=(N-1)L+1}^{(N-1)L+L-1} 2^{-j}\psi_{2,j}(i)$
$\sum_{j=(N-1)L+1}^{NL} 2^{-j}\psi_{2,j}(i)$

Figure 5.4: An illustration of the desired network architecture implementing $\phi_2 = \psi_2 \circ \psi_1$ for any input $\boldsymbol{\beta} \in \{0, 1, \cdots, K-1\}^d$, where $i = \psi_1(\boldsymbol{\beta})$.

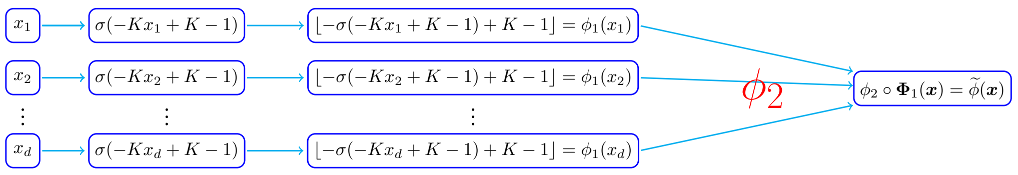
As we can see from Figure 5.4, ϕ_2 can be implemented by a Floor-ReLU network with width

$$N\Big((2N+2)+3\Big) = 2N^2 + 5N$$

and depth

$$1 + L\Big((7dL-2)+1\Big) + 1 = L(7dL-1) + 2.$$

Note that, for each j, $\psi_{2,j}$ is implemented by a Floor-ReLU network, with a fixed architecture independent of $\xi_{i,j}$ that is essentially determined by the target function f for all i. Thus, the Floor-ReLU network implementing ϕ_2 has a fixed architecture independent of f, as shown in Figure 5.4.



x_1
x_2
x_d
$\sigma(-Kx_1+K-1)$
$\sigma(-Kx_2+K-1)$
$\sigma(-Kx_d+K-1)$
$\lfloor -\sigma(-Kx_1+K-1)+K-1 \rfloor = \phi_1(x_1)$
$\lfloor -\sigma(-Kx_2+K-1)+K-1 \rfloor = \phi_1(x_2)$
$\lfloor -\sigma(-Kx_d+K-1)+K-1 \rfloor = \phi_1(x_d)$
ϕ_2
$\phi_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{x}) = \widetilde{\phi}(\boldsymbol{x})$

Figure 5.5: An illustration of the network architecture implementing $\widetilde{\phi} = \phi_2 \circ \boldsymbol{\Phi}_1$ for any $\boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in [0,1]^d$.

Note that ϕ is defined via re-scaling and shifting $\widetilde{\phi}$. As shown in Figure 5.5, ϕ and $\widetilde{\phi}$ can be implemented by a Floor-ReLU network, with a fixed architecture independent of f, with width $\max\{d,\, 2N^2+5N\}$ and depth $2 + L(7dL-1) + 2 \le 7dL^2 + 3$. So we finish the proof. □

5.3 Proof of key proposition for auxiliary theorem

The proof of Proposition 5.6 mainly relies on the "bit extraction" technique. As we shall see later, our key idea is to apply the Floor activation function to make "bit extraction" more powerful to reduce network sizes. In particular, Floor-ReLU networks can extract much more bits than ReLU networks with the same network size.

Let us first establish a basic lemma to extract $1/N$ of total bits stored in a new binary number from an input binary number.

Lemma 5.7. *Given any $J, N \in \mathbb{N}^+$, there exists a function $\phi : \mathbb{R}^2 \to \mathbb{R}$ implemented by a Floor-ReLU network with width $2N$ and depth 4 such that, for any $\theta_j \in \{0,1\}$, $j = 1, 2, \cdots, NJ$, we have*

$$\phi(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{NJ},\, n) = \text{bin}\,0.\theta_{(n-1)J+1}\theta_{(n-1)J+2}\cdots\theta_{nJ}, \quad \text{for } n = 1, 2, \cdots, N.$$

Proof. Given any $\theta_j \in \{0,1\}$ for $j = 1, 2, \cdots, NJ$, denote

$$s = \text{bin}\,0.\theta_1\theta_2\cdots\theta_{NJ} \quad \text{and} \quad s_n = \text{bin}\,0.\theta_{(n-1)J+1}\theta_{(n-1)J+2}\cdots\theta_{nJ},$$

for $n = 1, 2, \cdots, N$.

Then our goal is to construct a function $\phi : \mathbb{R}^2 \to \mathbb{R}$ implemented by a Floor-ReLU network with the desired width and depth that satisfies

$$\phi(s, n) = s_n, \quad \text{for } n = 1, 2, \cdots, N.$$

Based on the properties of the binary representation, it is easy to check that

$$s_n = \lfloor 2^{nJ} s \rfloor / 2^J - \lfloor 2^{(n-1)J} s \rfloor, \quad \text{for } n = 1, 2, \cdots, N. \tag{5.4}$$

With formulas to return $s_1, s_2, \cdots, s_N$, it is still technical to construct a network outputting s_n for a given index $n \in \{1, 2, \cdots, N\}$.

Set $\delta = 2^{-J}$ and define g (see Figure 5.6) as

$$g(x) \coloneqq \sigma\big(\sigma(x) - \sigma(\tfrac{x+\delta-1}{\delta})\big), \quad \text{for any } x \in \mathbb{R}.$$

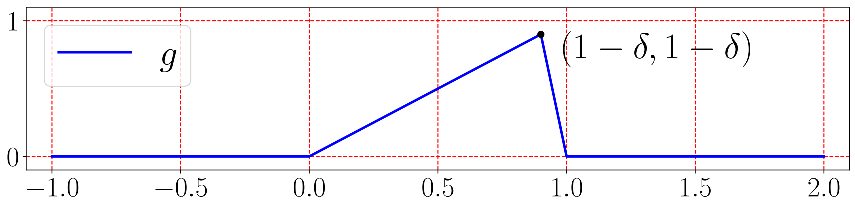


Figure 5.6: An illustration of $g(x) = \sigma\big(\sigma(x) - \sigma(\frac{x+\delta-1}{\delta})\big)$.

Since $s_n \in [0, 1-\delta]$ for $n = 1, 2, \cdots, N$, we have

$$s_n = \sum_{k=1}^{N} g(s_k + k - n), \quad \text{for } n = 1, 2, \cdots, N. \tag{5.5}$$

As shown in Figure 5.7, the desired function ϕ can be implemented by a Floor-ReLU network with width $2N$ and depth 4. Moreover, it holds that

$$\phi(s, n) = s_n, \quad \text{for } n = 1, 2, \cdots, N.$$

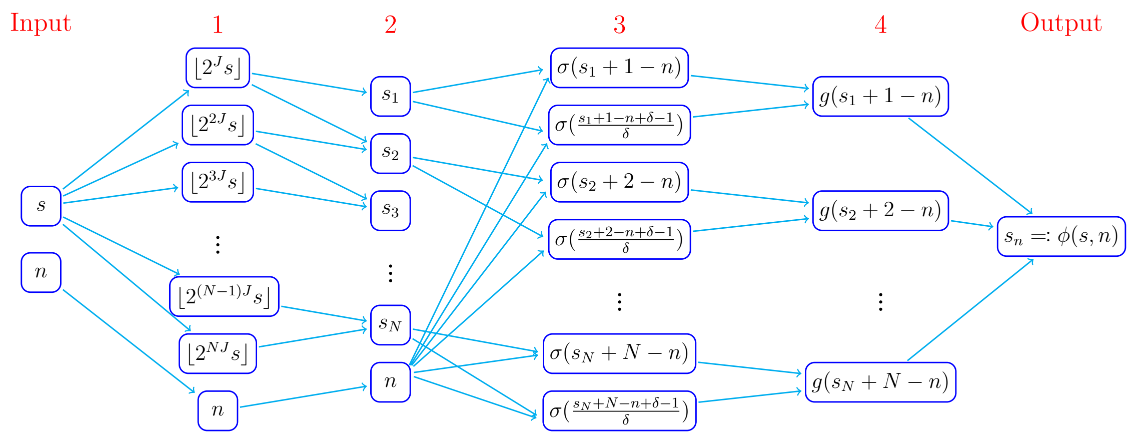

Input
1
2
3
4
Output
s
n
$\lfloor 2^J s\rfloor$
$\lfloor 2^{2J} s\rfloor$
$\lfloor 2^{3J} s\rfloor$
$\lfloor 2^{(N-1)J} s\rfloor$
$\lfloor 2^{NJ} s\rfloor$
n
s_1
s_2
s_3
s_N
n
$\sigma(s_1+1-n)$
$\sigma(\frac{s_1+1-n+\delta-1}{\delta})$
$\sigma(s_2+2-n)$
$\sigma(\frac{s_2+2-n+\delta-1}{\delta})$
$\sigma(s_N+N-n)$
$\sigma(\frac{s_N+N-n+\delta-1}{\delta})$
$g(s_1+1-n)$
$g(s_2+2-n)$
$g(s_N+N-n)$
$s_n =: \phi(s,n)$

Figure 5.7: An illustration of the Floor-ReLU network implementing the desired function ϕ based on Equation (5.4) and (5.5). All parameters in this network are essentially determined by Equation (5.4) and (5.5), which are valid no matter what $\theta_1, \cdots, \theta_{NJ} \in \{0,1\}$ are. Thus, the desired function ϕ implemented by this network is independent of $\theta_1, \cdots, \theta_{NJ} \in \{0,1\}$.

So we finish the proof. $\square$

The next lemma constructs a Floor-ReLU network that can extract any bit from a binary number according to a specific index.

Lemma 5.8. *Given any $N, L \in \mathbb{N}^+$, there exists a function $\phi : \mathbb{R}^2 \to \mathbb{R}$ implemented by a Floor-ReLU network with width $2N+2$ and depth $7L-3$ such that, for any $\theta_m \in \{0,1\}$, $m = 1, 2, \cdots, N^L$, we have*

$$\phi(\text{bin}\, 0.\theta_1\theta_2\cdots\theta_{N^L},\, m) = \theta_m, \quad \text{for } m = 1, 2, \cdots, N^L.$$

Proof. The proof is based on repeated applications of Lemma 5.7. To be exact, we construct a sequence of functions $\phi_1, \phi_2, \cdots, \phi_L$ implemented by Floor-ReLU networks by induction to satisfy the following two conditions for each $\ell \in \{1, 2, \cdots, L\}$.

(i) $\phi_\ell : \mathbb{R}^2 \to \mathbb{R}$ can be implemented by a Floor-ReLU network with width $2N+2$ and depth $7\ell - 3$.

(ii) For any $\theta_m \in \{0,1\}$, $m = 1, 2, \cdots, N^\ell$, we have

$$\phi_\ell(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^\ell},\, m) = \text{bin}\,0.\theta_m, \quad \text{for } m = 1, 2, \cdots, N^\ell.$$

First, consider the case $\ell = 1$. By Lemma 5.7 (set $J = 1$ therein), there exists a function ϕ_1 implemented by a Floor-ReLU network with width $2N \le 2N + 2$ and depth $4 = 7 - 3$ such that, for any $\theta_m \in \{0,1\}$, $m = 1, 2, \cdots, N$, we have

$$\phi_1(\text{bin}\,0.\theta_1\theta_2\cdots\theta_N,\, m) = \text{bin}\,0.\theta_m, \quad \text{for } m = 1, 2, \cdots, N.$$

It follows that Condition (i) and (ii) hold for $\ell = 1$.

Next, assume Condition (i) and (ii) hold for $\ell = k$. We would like to construct ϕ_{k+1} to make Condition (i) and (ii) true for $\ell = k+1$. By Lemma 5.7 (set $J = N^k$ therein), there exists a function ψ implemented by a Floor-ReLU network with width $2N$ and depth 4 such that, for any $\theta_m \in \{0,1\}$, $m = 1, 2, \cdots, N^{k+1}$, we have

$$\psi(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^{k+1}},\, n) = \text{bin}\,0.\theta_{(n-1)N^k+1}\theta_{(n-1)N^k+2}\cdots\theta_{(n-1)N^k+N^k}, \tag{5.6}$$

for $n = 1, 2, \cdots, N$. By the induction hypothesis, we have

- $\phi_k : \mathbb{R}^2 \to \mathbb{R}$ can be implemented by a Floor-ReLU network with width $2N+2$ and depth $7k - 3$.

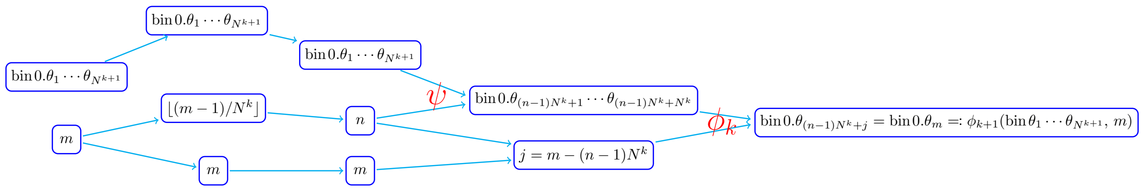
- For any $\theta_j \in \{0,1\}$, $j = 1, 2, \cdots, N^k$, we have

$$\phi_k(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^k},\, j) = \text{bin}\,0.\theta_j, \quad \text{for } j = 1, 2, \cdots, N^k. \tag{5.7}$$

Given any $m \in \{1, 2, \cdots, N^{k+1}\}$, there exist $n \in \{1, 2, \cdots, N\}$ and $j \in \{1, 2, \cdots, N^k\}$ such that $m = (n-1)N^k + j$, and such n, j can be obtained by

$$n = \lfloor (m-1)/N^k \rfloor + 1 \quad \text{and} \quad j = m - (n-1)N^k. \tag{5.8}$$

Then the desired architecture of the Floor-ReLU network implementing ϕ_{k+1} is shown in Figure 5.8.



$\mathrm{bin}\, 0.\theta_1 \cdots \theta_{N^{k+1}}$
$\mathrm{bin}\, 0.\theta_1 \cdots \theta_{N^{k+1}}$
$\mathrm{bin}\, 0.\theta_1 \cdots \theta_{N^{k+1}}$
m
$\lfloor (m-1)/N^k \rfloor$
m
n
m
ψ
$\mathrm{bin}\, 0.\theta_{(n-1)N^k+1} \cdots \theta_{(n-1)N^k+N^k}$
$j = m-(n-1)N^k$
ϕ_k
$\mathrm{bin}\, 0.\theta_{(n-1)N^k+j} = \mathrm{bin}\, 0.\theta_m =: \phi_{k+1}(\mathrm{bin}\, \theta_1 \cdots \theta_{N^{k+1}},\, m)$

Figure 5.8: An illustration of the Floor-ReLU network architecture implementing ϕ_{k+1}, based on Equation (5.6), (5.7), and (5.8) for any $\theta_m \in \{0,1\}$ and $m \in \{1, 2, \cdots, N^{k+1}\}$.

Note that ψ can be implemented by a Floor-ReLU network with width $2N$ and depth 4. Then the desired network implementing ϕ_{k+1} is shown in Figure 5.8. Moreover, we have

- $\phi_{k+1} : \mathbb{R}^2 \to \mathbb{R}$ can be implemented by a Floor-ReLU network with width $2N+2$ and depth $2+4+1+(7k-3) = 7(k+1)-3$, which implies Condition (i) for $\ell = k+1$.

- For any $\theta_m \in \{0,1\}$, $m = 1, 2, \cdots, N^{k+1}$, we have

$$\phi_{k+1}(\mathrm{bin}\, 0.\theta_1\theta_2 \cdots \theta_{N^{k+1}},\, m) = \mathrm{bin}\, 0.\theta_m, \quad \text{for } m = 1, 2, \cdots, N^{k+1}.$$

  That is, Condition (ii) holds for $\ell = k+1$.

So we finish the process of induction.

By the principle of induction, there exists a function $\phi_L : \mathbb{R}^2 \to \mathbb{R}$ such that

- ϕ_L can be implemented by a Floor-ReLU network with width $2N+2$ and depth $7L-3$.

- For any $\theta_m \in \{0,1\}$, $m = 1, 2, \cdots, N^L$, we have

$$\phi_L(\mathrm{bin}\, 0.\theta_1\theta_2 \cdots \theta_{N^L},\, m) = \mathrm{bin}\, 0.\theta_m, \quad \text{for } m = 1, 2, \cdots, N^L.$$

Finally, define $\phi \coloneqq 2\phi_L$. Then ϕ can also be implemented by a Floor-ReLU network with width $2N{+}2$ and depth $7L{-}3$. Moreover, for any $\theta_m \in \{0,1\}$, $m = 1,2,\cdots,N^L$, we have

$$\phi(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^L},\, m) = 2\cdot\phi_L(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^L},\, m) = 2\cdot\text{bin}\,0.\theta_m = \theta_m,$$

for $m = 1,2,\cdots,N^L$. So we finish the proof. □

With Lemma 5.8 in hand, we are ready to prove Proposition 5.6.

Proof of Proposition 5.6. By Lemma 5.8, there exists a function $\widetilde{\phi} : \mathbb{R}^2 \to \mathbb{R}$ implemented by a Floor-ReLU network with width $2N+2$ and depth $7L-3$ such that, for any $z_m \in \{0,1\}$, $m = 1,2,\cdots,N^L$, we have

$$\widetilde{\phi}(\text{bin}\,0.z_1z_2\cdots z_{N^L},\, m) = z_m, \quad \text{for } m = 1,2,\cdots,N^L.$$

Based on $\theta_m \in \{0,1\}$ for $m = 1,2,\cdots,N^L$ given in Proposition 5.6, we define the final function ϕ as

$$\phi(x) \coloneqq \widetilde{\phi}\big(\sigma(x\cdot 0 + \text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^L}), \sigma(x)\big), \quad \text{where } \sigma(x) = \max\{0,x\}.$$

Clearly, ϕ can be implemented by a Floor-ReLU network, with a fixed architecture independent of $\theta_m \in \{0,1\}$ for $m = 1,2,\cdots,N^L$, with width $2N+2$ and depth $1+(7L-3) = 7L-2$. In fact, only one parameter $(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^L})$ of the network implementing ϕ relies on $\theta_m \in \{0,1\}$ for $m = 1,2,\cdots,N^L$. Moreover, we have, for any $m \in \{1,2,\cdots,N^L\}$,

$$\phi(m) = \widetilde{\phi}\big(\sigma(m\cdot 0 + \text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^L}), \sigma(m)\big) = \widetilde{\phi}(\text{bin}\,0.\theta_1\theta_2\cdots\theta_{N^L}, m) = \theta_m.$$

So we finish the proof. □

We shall point out that only the properties of Floor on $[0,\infty)$ are used in our proof. Thus, the Floor can be replaced by the truncation function that can be easily implemented by truncating the decimal part.

Finally, we would like to remark that the key reason Floor-ReLU networks can attain much better approximation errors than ReLU networks is that Floor has infinite (constant) pieces, while ReLU has only two (linear) pieces. Thus, roughly speaking, one Floor activation function can do what many ReLU activation functions do in our construction. For this reason, compared to ReLU networks, Floor-ReLU networks attain significantly better approximation errors. In fact, one may replace Floor by other activation functions with "many pieces". For example, it is shown in [60] that ReLU/Sin-activated networks can also attain nearly exponential approximation errors.

Chapter 6

Conclusion

This dissertation aims to study the approximation power of ReLU networks and Floor-ReLU networks. Based on the idea of function compositions, we construct ReLU networks to uniformly approximate polynomials, Hölder continuous functions, general continuous functions, and smooth functions on a d-dimensional hypercube $[0,1]^d$ with (nearly optimal) approximation errors. All the approximation error estimates are characterized by the width and depth simultaneously and have explicit formulas for the prefactors. Meanwhile, the optimality of the approximation by ReLU networks is discussed via studying the connection between the approximation error and VC-dimension.

To overcome the limitation of ReLU networks that (nearly) exponential approximation errors can only be attained for polynomials among all function spaces considered, we introduce Floor-ReLU networks. It is proved by construction that nearly exponential approximation errors can be attained when using Floor-ReLU networks with fixed architectures to uniformly approximate (Hölder) continuous functions on $[0,1]^d$. In other words, approximation errors are improved from polynomial ones to nearly exponential ones via adding a simple activation function (Floor) to ReLU networks. The optimality of the approximation by Floor-ReLU networks is not discussed due to the nearly exponential approximation errors. All these results stated above completely solve the three problems (Problem 1, 2, and 3) listed in Chapter 1 for certain function spaces.

Finally, we would like to remark that our analysis is only for the feed-forward fully connected neural networks with two types of activation functions: ReLU and Floor. It would be an interesting direction to generalize our results to neural networks with other architectures (*e.g.*, convolutional neural networks and residual networks) and activation functions (*e.g.*, tanh and sigmoid functions). These will be left as future work.

Bibliography

[1] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, Oct 2014.

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

[3] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993.

[4] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[5] P. L. Bartlett, V. Maiorov, and R. Meir. Almost linear VC-dimension bounds for piecewise polynomial networks. *Neural Computation*, 10(8):2159–2173, 1998.

[6] Y. Bengio, N. Léonard, and A. C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

[7] 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(8):1553–1565, Aug 2014.

[8] E. K. Blum and L. K. Li. Approximation theory and feedforward networks. *Neural Networks*, 4(4):511–515, 1991.

[9] Y. Boo, S. Shin, and W. Sung. Quantized neural networks: Characterization and holistic optimization. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6, 2020.

[10] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems 2*, pages 321–355, 1988.

[11] J. A. Carrillo, S. Jin, L. Li, and Y. Zhu. A consensus-based global optimization method for high dimensional machine learning problems. *arXiv e-prints*, page arXiv:1909.09249, Sep 2019.

[12] S. Chen and D. Donoho. Basis pursuit. In *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 41–44, Oct 1994.

[13] D. Costarelli and A. R. Sambucini. Saturation classes for max-product neural network operators activated by sigmoidal functions. *Results in Mathematics*, 72(3):1555–1569, 2017.

[14] D. Costarelli and A. R. Sambucini. Approximation results in Orlicz spaces for sequences of Kantorovich max-product neural network operators. *Results in Mathematics*, 73(1):1–15, 2018.

[15] D. Costarelli and G. Vinti. Convergence for a family of neural network operators in Orlicz spaces. *Mathematische Nachrichten*, 290(2-3):226–235, 2017.

[16] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[17] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (deep) ReLU networks. *arXiv e-prints*, page arXiv:1905.02199, May 2019.

[18] R. DEVORE and A. RON. Approximation using scattered shifts of a multivariate function. *Transactions of the American Mathematical Society*, 362(12):6205–6229, 2010.

[19] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[20] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, Dec 2017.

[21] W. E and Q. Wang. Exponential convergence of the deep neural network approximation for analytic functions. *Science China Mathematics*, 61:1733–1740, 2018.

[22] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[23] T. Hangelbroek and A. Ron. Nonlinear approximation using gaussian kernels. *Journal of Functional Analysis*, 259(1):203–219, 2010.

[24] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[25] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[26] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(1):6869–6898, Jan 2017.

[27] K. Kawaguchi. Deep learning without poor local minima. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 586–594. Curran Associates, Inc., 2016.

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

[29] M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, Jun 1994.

[30] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[32] V. Kůrková. Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, 5(3):501–506, 1992.

[33] G. Lewicki and G. Marino. Approximation of functions of finite variation by superpositions of a sigmoidal function. *Applied Mathematics Letters*, 17(10):1147–1152, 2004.

[34] S. Liang and R. Srikant. Why deep neural networks for function approximation? *arXiv e-prints*, page arXiv:1610.04161, Oct 2016.

[35] S. Lin, X. Liu, Y. Rong, and Z. Xu. Almost optimal estimates for approximation and learning by radial basis function networks. *Machine Learning*, 95(2):147–164, May 2014.

[36] Y. Lin, M. Lei, and L. Niu. Optimization strategies in quantized neural networks: A review. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 385–390, 2019.

[37] B. Llanas and F. Sainz. Constructive approximate interpolation by neural networks. *Journal of Computational and Applied Mathematics*, 188(2):283–308, 2006.

[38] J. Lu, Z. Shen, H. Yang, and S. Zhang. Deep network approximation for smooth functions. *arXiv e-prints*, page arXiv:2001.03040, Jan 2020.

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

[40] V. Maiorov and A. Pinkus. Lower bounds for approximation by MLP neural networks. *Neurocomputing*, 25(1):81–91, 1999.

[41] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, Dec 1993.

[42] H. Montanelli and Q. Du. New error bounds for deep ReLU networks using sparse grids. *SIAM Journal on Mathematics of Data Science*, 1(1):78–92, 2019.

[43] H. Montanelli, H. Yang, and Q. Du. Deep ReLU networks overcome the curse of dimensionality for bandlimited functions. *arXiv e-prints*, page arXiv:1903.00735, Mar 2019.

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

[45] Q. Nguyen and M. Hein. The loss surface of deep and wide neural networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2603–2612, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[46] J. A. A. Opschoor, C. Schwab, and J. Zech. Exponential ReLU DNN expression of holomorphic maps in high dimension. Technical Report 2019-35, Seminar for Applied Mathematics, ETH Zürich, Switzerland., 2019.

[47] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, Jun 1991.

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

[49] P. Petrushev. Multivariate n-term rational and piecewise polynomial approximation. *Journal of Approximation Theory*, 121(1):158–197, 2003.

[50] R. Pinnau, C. Totzeck, O. Tse, and S. Martin. A consensus-based model for global optimization and its mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 27(01):183–204, 2017.

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

[52] Z. Shen, H. Yang, and S. Zhang. Nonlinear approximation via compositions. *Neural Networks*, 119:74–84, 2019.

[53] Z. Shen, H. Yang, and S. Zhang. Deep network approximation characterized by number of neurons. *Communications in Computational Physics*, 28(5):1768–1811, 2020.

[54] 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*, page arXiv:2006.12231, Jun 2020.

[55] T. Suzuki. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

[56] P. Wang, Q. Hu, Y. Zhang, C. Zhang, Y. Liu, and J. Cheng. Two-step quantization for low-bit neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4376–4384, 2018.

[57] T. F. Xie and F. L. Cao. The rate of approximation of gaussian radial basis neural networks in continuous function space. *Acta Mathematica Sinica, English Series*, 29(2):295–302, Feb 2013.

[58] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

[59] D. Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 639–649. PMLR, Jul 2018.

[60] D. Yarotsky and A. Zhevnerchuk. The phase diagram of approximation rates for deep neural networks. *arXiv e-prints*, page arXiv:1906.09477, Jun 2019.

[61] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin. Understanding straight-through estimator in training activation quantized neural nets. *arXiv e-prints*, page arXiv:1903.05662, Mar 2019.