

NONLINEAR APPROXIMATION VIA COMPOSITIONS*

ZUOWEI SHEN[†], HAIZHAO YANG[‡], AND SHIJUN ZHANG[§]

Abstract. Given a function dictionary \mathcal{D} and an approximation budget $N \in \mathbb{N}^+$, nonlinear approximation seeks the linear combination of the best N terms $\{T_n\}_{1 \leq n \leq N} \subseteq \mathcal{D}$ to approximate a given function f with the minimum approximation error

$$\varepsilon_{L,f} := \min_{\{g_n\} \subseteq \mathbb{R}, \{T_n\} \subseteq \mathcal{D}} \left\| f(\mathbf{x}) - \sum_{n=1}^N g_n T_n(\mathbf{x}) \right\|.$$

Motivated by recent success of deep learning, we propose dictionaries with functions in a form of compositions, i.e.,

$$T(\mathbf{x}) = T^{(L)} \circ T^{(L-1)} \circ \dots \circ T^{(1)}(\mathbf{x})$$

for all $T \in \mathcal{D}$, and implement T using ReLU feed-forward neural networks (FNNs) with L hidden layers. We further quantify the improvement of the best N -term approximation rate in terms of N when L is increased from 1 to 2 or 3 to show the power of compositions. In the case when $L > 3$, our analysis shows that increasing L cannot improve the approximation rate in terms of N .

In particular, for any function f on $[0, 1]$, regardless of its smoothness and even the continuity, if f can be approximated using a dictionary when $L = 1$ with the best N -term approximation rate $\varepsilon_{L,f} = \mathcal{O}(N^{-\eta})$, we show that dictionaries with $L = 2$ can improve the best N -term approximation rate to $\varepsilon_{L,f} = \mathcal{O}(N^{-2\eta})$. We also show that for Hölder continuous functions of order α on $[0, 1]^d$, the application of a dictionary with $L = 3$ in nonlinear approximation can achieve an essentially tight best N -term approximation rate $\varepsilon_{L,f} = \mathcal{O}(N^{-2\alpha/d})$. Finally, we show that dictionaries consisting of wide FNNs with a few hidden layers are more attractive in terms of computational efficiency than dictionaries with narrow and very deep FNNs for approximating Hölder continuous functions if the number of computer cores is larger than N in parallel computing.

Key words. Deep Neural Networks, ReLU Activation Function, Nonlinear Approximation, Function Composition, Hölder Continuity, Parallel Computing.

1. Introduction. For non-smooth and high-dimensional function approximation, a favorable technique popularized in recent decades is the nonlinear approximation (DeVore, 1998) that does not limit the approximants to come from linear spaces, obtaining sparser representation, cheaper computation, and more robust estimation, and therein emerged the bloom of many breakthroughs in applied mathematics and computer science (e.g., wavelet analysis (Daubechies, 1992), dictionary learning (Tariyal, Majumdar, Singh, & Vatsa, 2016), data compression and denoising (Jiang, 1996; Joutsensalo, 1994), adaptive pursuit (Davis, 1994; Ohlsson, Yang, Dong, & Sastry, 2013), compressed sensing (Donoho, 2006; Candes & Wakin, 2008)).

Typically, nonlinear approximation is a two-stage algorithm that designs a good redundant nonlinear dictionary, \mathcal{D} , in its first stage, and identifies the optimal approximant as a linear combination of N elements of \mathcal{D} in the second stage:

$$f(\mathbf{x}) \approx g \circ T(\mathbf{x}) := \sum_{n=1}^N g_n T_n(\mathbf{x}), \tag{1.1}$$

where $f(\mathbf{x})$ is the target function in a Hilbert space \mathcal{H} associated with a norm denoted as $\|\cdot\|_*$, $\{T_n\} \subseteq \mathcal{D} \subseteq \mathcal{H}$, T is a nonlinear map from \mathbb{R}^d to \mathbb{R}^N with the n -th coordinate being T_n , and g is a linear map from \mathbb{R}^N to \mathcal{H} with the n -th coordinate being $g_n \in \mathbb{R}$. The nonlinear approximation seeks g and T such that

$$\{\{T_n\}, \{g_n\}\} = \arg \min_{\{g_n\} \subseteq \mathbb{R}, \{T_n\} \subseteq \mathcal{D}} \left\| f(\mathbf{x}) - \sum_{n=1}^N g_n T_n(\mathbf{x}) \right\|_*, \tag{1.2}$$

which is also called the best N -term approximation. One remarkable approach of nonlinear approximation is based on one-hidden-layer neural networks that give simple and elegant bases of the form $T(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$,

*

Funding: H. Yang was supported by the startup of the Department of Mathematics at the National University of Singapore.

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

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

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

where $\mathbf{W}\mathbf{x} + \mathbf{b}$ is a linear transform in \mathbf{x} with the transformation matrix \mathbf{W} (named as the weight matrix) and a shifting vector \mathbf{b} (called bias), and σ is a nonlinear function (called the activation function). The approximation

$$f(\mathbf{x}) \approx \sum_{n=1}^N g_n T_n(\mathbf{x}) = \sum_{n=1}^N g_n \sigma(\mathbf{W}_n \mathbf{x} + \mathbf{b}_n)$$

includes wavelets pursuits (Mallat & Zhang, 1993; Chen & Donoho, 1994), adaptive splines (DeVore, 1998; Petrushev, 2003), radial basis functions (DEVORE & RON, 2010; Hangelbroek & Ron, 2010; Xie & Cao, 2013), sigmoidal neural networks (Llanas & Sainz, 2006; Lewicki & Marino, 2004; Costarelli & Vinti, 2017; Costarelli & Sambucini, 2017; Costarelli & Vinti, 2018; Cybenko, 1989; Hornik, Stinchcombe, & White, 1989; Barron, 1993), etc. For functions in Besov spaces with smoothness s , (DEVORE & RON, 2010; Hangelbroek & Ron, 2010) constructed an $\mathcal{O}(N^{-s/d})$ ^① approximation that is almost optimal (Lin, Liu, Rong, & Xu, 2014) and the smoothness cannot be reduced generally (Hangelbroek & Ron, 2010). For Hölder continuous functions of order 1 on $[0, 1]^d$, (Xie & Cao, 2013) essentially constructed an $\mathcal{O}(N^{-\frac{1}{2d}})$ approximation, which is far from the lower bound $\mathcal{O}(N^{-2/d})$ as we shall prove in this paper. Achieving the optimal approximation rate of general continuous functions in constructive approximation, especially in high dimensional spaces, remains an unsolved challenging problem.

1.1. Problem Statement. ReLU FNNs have been proved to be a powerful tool in many fields from various points of view (Montufar, Pascanu, Cho, & Bengio, 2014; Bianchini & Scarselli, 2014; Bartlett, Maiorov, & Meir, 1998; Sakurai, 1999; Harvey, Liaw, & Mehrabian, 2017; Kearns & Schapire, 1994; Anthony & Bartlett, 2009; Petersen & Voigtlaender, 2018), which motivates us to tackle the open problem above via function compositions in the nonlinear approximation using deep ReLU FNNs, i.e.,

$$f(\mathbf{x}) \approx g \circ T^{(L)} \circ T^{(L-1)} \circ \dots \circ T^{(1)}(\mathbf{x}), \quad (1.3)$$

where $T^{(i)}(\mathbf{x}) = \sigma(\mathbf{W}_i \mathbf{x} + \mathbf{b}_i)$ with $\mathbf{W}_i \in \mathbb{R}^{N_i \times N_{i-1}}$, $\mathbf{b}_i \in \mathbb{R}^{N_i}$ for $i = 1, \dots, L$, σ is the ReLU activation function, and f is a Hölder continuous function. For the convenience of analysis, we consider $N_i = N$ for $i = 1, \dots, L$. Let \mathcal{D}_L be the dictionary consisting of ReLU FNNs $g \circ T^{(L)} \circ T^{(L-1)} \circ \dots \circ T^{(1)}(\mathbf{x})$ with width N and depth L . To identify the optimal FNN to approximate $f(\mathbf{x})$, it is sufficient to solve the following optimization problem

$$\phi^* = \arg \min_{\phi \in \mathcal{D}_L} \|f - \phi\|_*. \quad (1.4)$$

The fundamental limit of nonlinear approximation via the proposed dictionary is essentially determined by the approximation power of function compositions in (1.3), which gives a performance guarantee of the minimizer in (1.4). Since function compositions are implemented via ReLU FNNs, the remaining problem is to quantify the approximation capacity of deep ReLU FNNs, especially their ability to improve the best N -term approximation rate in N for any fixed L defined as

$$\varepsilon_{L,f}(N) = \min_{\phi \in \mathcal{D}_L} \|f - \phi\|_*. \quad (1.5)$$

Function compositions can significantly enrich the dictionary of nonlinear approximation and this idea was not considered in the literature previously due to the expensive computation of function compositions in solving the minimization problem in (1.4). Fortunately, recent development of efficient algorithms for optimization with compositions (e.g., backpropagation techniques (Werbos, 1975; Fukushima, 1980; Rumelhart, McClelland, Group, & University of California, 1986) and parallel computing techniques (Scherer, Müller, & Behnke, 2010; Cireşan, Meier, Masci, Gambardella, & Schmidhuber, 2011)) makes it possible to explore the proposed dictionary in this paper. Furthermore, with advanced optimization algorithms (Duchi, Hazan, & Singer, 2011; Johnson & Zhang, 2013; Kingma & Ba, 2014), good local minima of (1.4) can be identified efficiently (Kawaguchi, 2016; Nguyen & Hein, 2017; Kawaguchi & Bengio, 2018).

^①In this paper, we use the big $\mathcal{O}(\cdot)$ notation when we only care about the scaling in terms of the variables inside (\cdot) and the prefactor outside (\cdot) is independent of these variables.

81 **1.2. Related work and contribution.** The main goal in the remaining article is to quantify the best
 82 N -term approximation rate $\varepsilon_{L,f}(N)$ defined in (1.5) for ReLU FNNs in the dictionary \mathcal{D}_L with a fixed depth
 83 L when f is a Hölder continuous function. This topic is related to several existing approximation theories in
 84 the literature, but none of these existing works can be applied to answer the problem addressed in this paper.

85 First of all, this paper identifies explicit formulas for the best N -term approximation rate

$$86 \quad \varepsilon_{L,f}(N) \leq \begin{cases} 2\nu N^{-2\alpha}, & \text{when } L \geq 2 \text{ and } d = 1, \\ 2(2\sqrt{d})^\alpha \nu N^{-2\alpha/d}, & \text{when } L \geq 3 \text{ and } d > 1, \end{cases} \quad (1.6)$$

87 for any $N \in \mathbb{N}^+$ and a Hölder continuous function f of order α with a constant ν , while existing theories
 88 (Lu, Pu, Wang, Hu, & Wang, 2017; Montanelli, Yang, & Du, 2019; Yarotsky, 2017; Liang & Srikant, 2016;
 89 Montanelli & Du, 2017; Suzuki, 2019; Petersen & Voigtlaender, 2018; Yarotsky, 2018; E & Wang, 2018) can
 90 only provide implicit formulas in the sense that the approximation error contains an unknown prefactor and
 91 work only for sufficiently large N or L larger than some unknown numbers. For example, the approximation
 92 rate in (Yarotsky, 2018) via a narrow and deep ReLU FNN is $c(d)L^{-2\alpha/d}$ with $c(d)$ unknown and for L larger
 93 than a sufficiently large unknown number \mathcal{L} ; the approximation rate in (Yarotsky, 2018) via a wide and shallow
 94 ReLU FNN is $c(d)N^{-\alpha/d}$ with $c(d)$ unknown and for N larger than a sufficiently large unknown number \mathcal{N} .
 95 For another example, given an approximation error ε , (Petersen & Voigtlaender, 2018) proved the existence
 96 of a ReLU FNN with a constant but still unknown number of layers approximating a C^β function within the
 97 target error. Similarly, given the ε error, (Montanelli & Du, 2017; Montanelli et al., 2019; E & Wang, 2018)
 98 estimate the scaling of the network size in ε and the scaling contains unknown prefactors. Given an arbitrary
 99 L and N , no existing work can provide an explicit formula for the approximation error to guide practical
 100 network design, e.g., to guarantee whether the network is large enough to meet the accuracy requirement.
 101 This paper provides such formulas for the first time and in fact the bound in these formulas is asymptotically
 102 tight as we shall prove later.

103 Second, our target functions are Hölder continuous, while most of existing works aim for a smaller function
 104 space with certain smoothness, e.g. functions in $C^\alpha([0, 1]^d)$ with $\alpha \geq 1$ (Lu et al., 2017; Liang & Srikant,
 105 2016; Yarotsky, 2017; E & Wang, 2018), band-limited functions (Montanelli et al., 2019), Korobov spaces
 106 (Montanelli & Du, 2017), or Besov spaces (Suzuki, 2019). To the best of our knowledge, there is only one
 107 existing article (Yarotsky, 2018) concerning the approximation power of deep ReLU FNNs for $C([0, 1]^d)$.
 108 However, the conclusion of (Yarotsky, 2018) only works for ReLU FNNs with a fixed width $2d + 10$ and a
 109 sufficiently large L , instead of a fixed L and an arbitrary N as required in the nonlinear approximation (see
 110 Figure 1 for the comparison of the conclusion of (Yarotsky, 2018) and this paper).

111 As we can see in Figure 1, the improvement of the best N -term approximation rate in terms of N when
 112 L is increased from 1 to 2 or 3 is significant, which shows the power of depth in ReLU FNNs. However, in the
 113 case when $L > 3$, our analysis shows that increasing L cannot improve the approximation rate in terms of N .
 114 As an interesting corollary of our analysis, for any function f on $[0, 1]$, regardless of its smoothness and even
 115 the continuity, if f can be approximated using using functions in \mathcal{D}_1 with the best N -term approximation
 116 rate $\varepsilon_{L,f} = \mathcal{O}(N^{-\eta})$, we show that functions in \mathcal{D}_2 can improve the best N -term approximation rate to
 117 $\varepsilon_{L,f} = \mathcal{O}(N^{-2\eta})$. Extending this conclusion for a general d dimensional function is challenging and we leave it
 118 as future work.

119 From the point of view of analysis techniques, this paper introduce new analysis methods merely based
 120 on the structure of FNNs, while existing works (Lu et al., 2017; Montanelli et al., 2019; Yarotsky, 2017; Liang
 121 & Srikant, 2016; Montanelli & Du, 2017; Suzuki, 2019; Petersen & Voigtlaender, 2018; Yarotsky, 2018; E
 122 & Wang, 2018) rely on constructing FNNs to approximate traditional basis in approximation theory, e.g.,
 123 polynomials, splines, and sparse grids, which are used to approximate smooth functions.

124 Finally, we analyze the approximation efficiency of neural networks in parallel computing, a very important
 125 point of view that was not paid attention to in the literature. In most applications, the efficiency of deep
 126 learning computation highly relies on parallel computation. We show that a narrow and very deep neural
 127 network is inefficient if its approximation rate is not exponentially better than wide and shallower networks.

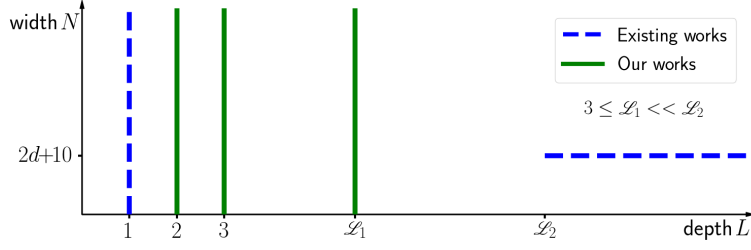


FIG. 1. A comparison of existing works and our contribution on the approximation power of ReLU FNNs for Hölder continuous functions of order α . Existing results in two cases: 1) $\mathcal{O}(N^{-\alpha/d})$ approximation rate for ReLU FNNs with depth $L = 1$ and width N ; 2) $\mathcal{O}(L^{-2\alpha/d})$ approximation rate for ReLU FNNs with depth L larger than a sufficiently large unknown number \mathcal{L}_2 and width $2d + 10$. Our contribution: $\mathcal{O}(N^{-2\alpha/d})$ approximation rate for ReLU FNNs width depth $L \geq 3$ and width N in the case of $d > 1$ ($L \geq 2$ in the case of $d = 1$).

Hence, neural networks with $\mathcal{O}(1)$ layers are more attractive in modern computational platforms, considering the computational efficiency per training iteration in parallel computing platforms. Our conclusion does not conflict with the current state-of-the-art deep learning research since most of these successful deep neural networks have a depth that is asymptotically $\mathcal{O}(1)$ relative to the width.

1.3. Organization. The rest of the paper is organized as follows. Section 2 summarizes the notations throughout this paper. Section 3 presents the main theorems while Section 4 shows numerical tests in parallel computing to support the claims in this paper. Finally, Section 5 concludes this paper with a short discussion.

2. Preliminaries. For the purpose of convenience, we present notations and elementary lemmas used throughout this paper as follows.

2.1. Notations.

- Matrices are denoted by bold uppercase letters, e.g., $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a real matrix of size $m \times n$, and \mathbf{A}^T denotes the transpose of \mathbf{A} . Correspondingly, $\mathbf{A}(i, j)$ is the (i, j) -th entry of \mathbf{A} ; $\mathbf{A}(:, j)$ is the j -th column of \mathbf{A} ; $\mathbf{A}(i, :)$ is the i -th row of \mathbf{A} .
- Vectors are denoted as bold lowercase letters, e.g., $\mathbf{v} \in \mathbb{R}^n$ is a column vector of size n and $\mathbf{v}(i)$ is the i -th element of \mathbf{v} . $\mathbf{v} = [v_1, \dots, v_n]^T = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$ are vectors consisting of numbers $\{v_i\}$ with $\mathbf{v}(i) = v_i$.
- The Lebesgue measure is denoted as $\mu(\cdot)$.
- The set difference of A and B is denoted by $A \setminus B$. A^c denotes $[0, 1]^d \setminus A$ for any $A \subset [0, 1]^d$.
- For a set of numbers A , and a number x , $A - x := \{y - x : y \in A\}$.
- For any $\xi \in \mathbb{R}$, let $\lfloor \xi \rfloor := \max\{i : i \leq \xi, i \in \mathbb{Z}\}$ and $\lceil \xi \rceil := \min\{i : i \geq \xi, i \in \mathbb{Z}\}$.
- Assume $\mathbf{n} \in \mathbb{N}^n$, then $f(\mathbf{n}) = \mathcal{O}(g(\mathbf{n}))$ means that there exists positive C independent of \mathbf{n} , f , and g such that $f(\mathbf{n}) \leq Cg(\mathbf{n})$ when $\mathbf{n}(i)$ goes to $+\infty$ for all i .
- Define $\text{Lip}(\nu, \alpha, d)$ as the class of functions g defined on $[0, 1]^d$ satisfying the uniformly Lipschitz property of order α with a Lipschitz constant $\nu > 0$.
- Let $\text{CPL}(N)$ be the set of continuous piecewise linear functions with $N - 1$ pieces mapping $[0, 1]$ to \mathbb{R} . The endpoints of each linear piece are called “breakpoints” in this paper.
- Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ denote the rectified linear unit (ReLU), i.e. $\sigma(x) = \max\{0, x\}$. With the abuse of notations, we define $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^d$ as $\sigma(\mathbf{x}) = \begin{bmatrix} \max\{0, x_1\} \\ \vdots \\ \max\{0, x_d\} \end{bmatrix}$ for any $\mathbf{x} = [x_1, \dots, x_d]^T \in \mathbb{R}^d$.
- We will use NN as a ReLU neural network for short and use Python-type notations to specify a class of NN’s, e.g., $\text{NN}(c_1; c_2; \dots; c_m)$ is a set of ReLU FNN’s satisfying m conditions given by $\{c_i\}_{1 \leq i \leq m}$, each of which may specify the number of inputs (#input), the total number of nodes in all hidden

layers (#node), the number of hidden layers (#layer), the total number of parameters (#parameter), and the width in each hidden layer (widthvec), the maximum width of all hidden layers (maxwidth) etc. For example, $\text{NN}(\#\text{input} = 2; \text{widthvec} = [100, 100])$ is a set of NN's ϕ satisfying:

- ϕ maps from \mathbb{R}^2 to \mathbb{R} .
- ϕ has two hidden layers and the number of nodes in each hidden layer is 100.
- $[n]^L$ is short for $[n, n, \dots, n] \in \mathbb{N}^L$. For example,

$$\text{NN}(\#\text{input} = d; \text{widthvec} = [100, 100, 100]) = \text{NN}(\#\text{input} = d; \text{widthvec} = [100]^3).$$

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

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

where $\mathbf{W}_i \in \mathbb{R}^{N_i \times N_{i-1}}$ and $\mathbf{b}_i \in \mathbb{R}^{N_i}$ are the weight matrix and the bias vector in the i -th linear transform in ϕ , respectively, i.e., $\mathbf{h}_i := \mathbf{W}_i \tilde{\mathbf{h}}_{i-1} + \mathbf{b}_i$ for $i = 1, 2, \dots, L+1$ and $\tilde{\mathbf{h}}_i = \sigma(\mathbf{h}_i)$ for $i = 1, 2, \dots, L$.

2.2. Lemmas. Let us study the properties of ReLU FNNs with only one hidden layer to warm up in Lemma 2.1 below. It indicates that $\text{CPL}(N+1) = \text{NN}(\#\text{input} = 1; \text{widthvec} = [N])$ for any $N \in \mathbb{N}^+$.

LEMMA 2.1. *Suppose $\phi \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [N])$ with an architecture:*

$$x \xrightarrow{\mathbf{W}_1, \mathbf{b}_1} \mathbf{h} \xrightarrow{\sigma} \tilde{\mathbf{h}} \xrightarrow{\mathbf{W}_2, \mathbf{b}_2} \phi(x).$$

Then ϕ is a continuous piecewise linear function. Let $\mathbf{W}_1 = [1, 1, \dots, 1]^T \in \mathbb{R}^{N \times 1}$, then we have:

- (1) Given a sequence of strictly increasing numbers x_0, x_1, \dots, x_N , there exists $\mathbf{b}_1 \in \mathbb{R}^N$ independent of \mathbf{W}_2 and \mathbf{b}_2 such that the break points of ϕ are exactly x_0, \dots, x_N on the interval $[x_0, x_N]$ ^②.
- (2) Suppose $\{x_i\}_{i \in \{0, 1, \dots, N\}}$ and \mathbf{b}_1 are given in (1). Given any sequence $\{y_i\}_{i \in \{0, 1, \dots, N\}}$, there exist \mathbf{W}_2 and \mathbf{b}_2 such that $\phi(x_i) = y_i$ for $i = 0, 1, \dots, N$ and ϕ is linear on $[x_i, x_{i+1}]$ for $i = 0, 1, \dots, N-1$.

Part (1) in Lemma 2.1 follows by setting $\mathbf{b}_1 = [-x_0, -x_1, \dots, -x_{N-1}]^T$. The existence in Part (2) is equivalent to the existence of a solution of linear equations, which is left for the reader. Next, we study the properties of ReLU FNNs with two hidden layers. In fact, we can show that the closure of $\text{NN}(\#\text{input} = 1; \text{widthvec} = [2m, 2n+1])$ contains $\text{CPL}(mn+1)$ for any $m, n \in \mathbb{N}^+$, where the closure is in the sense of L^p -norm for any $p \in [1, \infty)$. The proof of this property relies on the following lemma.

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

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

Proof. For simplicity of notation, we define $I_0(m, n) := \{0, 1, \dots, m(n+1)\}$, $I_1(m, n) := \{(n+1)j : j = 1, 2, \dots, m\}$, and $I_2(m, n) := I_0(m, n) \setminus I_1(m, n)$ for any $m, n \in \mathbb{N}^+$. Since $\phi \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [2m, 2n+1])$, the architecture of ϕ is

$$x \xrightarrow{\mathbf{W}_1, \mathbf{b}_1} \mathbf{h} \xrightarrow{\sigma} \tilde{\mathbf{h}} \xrightarrow{\mathbf{W}_2, \mathbf{b}_2} \mathbf{g} \xrightarrow{\sigma} \tilde{\mathbf{g}} \xrightarrow{\mathbf{W}_3, \mathbf{b}_3} \phi(x). \quad (2.1)$$

Note that \mathbf{g} maps $x \in \mathbb{R}$ to $\mathbf{g}(x) \in \mathbb{R}^{2n+1}$ and hence each entry of $\mathbf{g}(x)$ itself is a sub-network with one hidden layer. Denote $\mathbf{g} = [g_0, g_1^+, g_1^-, \dots, g_n^+, g_n^-]^T$, then $\{g_0, g_1^+, g_1^-, \dots, g_n^+, g_n^-\} \subseteq \text{NN}(\#\text{input} = 1; \text{widthvec} = [2m])$. Our

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

196 proof of Lemma 2.2 is mainly based on the repeated applications of Lemma 2.1 to determine parameters of
 197 $\phi(x)$ such that Conditions (1) to (3) hold.

198 **Step 1:** Determine \mathbf{W}_1 and \mathbf{b}_1 .

199 By Lemma 2.1, $\exists \mathbf{W}_1 = [1, 1, \dots, 1]^T \in \mathbb{R}^{2m \times 1}$ and $\mathbf{b}_1 \in \mathbb{R}^{2m}$ such that sub-networks in $\{g_0, g_1^+, g_1^-, \dots, g_n^+, g_n^-\}$
 200 have the same set of break points: $\{x_i : i \in I_1(m, n) \cup (I_1(m, n) - 1) \cup \{0\}\}$, no matter what \mathbf{W}_2 and \mathbf{b}_2 are.

201 **Step 2:** Determine \mathbf{W}_2 and \mathbf{b}_2 .

202 This is the key step of the proof. Our ultimate goal is to set up $\mathbf{g} = [g_0, g_1^+, g_1^-, \dots, g_n^+, g_n^-]^T$ such that, after
 203 a nonlinear activation function, there exists a linear combination in the last step of our network (specified by
 204 \mathbf{W}_3 and \mathbf{b}_3 as shown in (2.1)) that can generate a desired $\phi(x)$ matching the sample points $\{(x_i, y_i)\}_{0 \leq i \leq m(n+1)}$.
 205 In the previous step, we have determined the break points of $\{g_0, g_1^+, g_1^-, \dots, g_n^+, g_n^-\}$ by setting up \mathbf{W}_1 and \mathbf{b}_1 ;
 206 in this step, we will identify $\mathbf{W}_2 \in \mathbb{R}^{(2n+1) \times 2m}$ and $\mathbf{b}_2 \in \mathbb{R}^{2n+1}$ to fully determine $\{g_0, g_1^+, g_1^-, \dots, g_n^+, g_n^-\}$. This
 207 will be conducted in two sub-steps.

208 **Step 2.1:** Set up.

209 Suppose $f_0(x)$ is a continuous piecewise linear function defined on $[0, 1]$ fitting the given samples $f_0(x_i) =$
 210 y_i for $i \in I_0(m, n)$, and f_0 is linear between any two adjacent points of $\{x_i : i \in I_0(m, n)\}$. We are able to
 211 choose $\mathbf{W}_2(1, \cdot)$ and $\mathbf{b}_2(1)$ such that $g_0(x_i) = f_0(x_i)$ for $i \in I_1(m, n) \cup (I_1(m, n) - 1) \cup \{0\}$ by Lemma 2.1, since
 212 there are $2m + 1$ points in $I_1(m, n) \cup (I_1(m, n) - 1) \cup \{0\}$. Define $f_1 := f_0 - \tilde{g}_0$, where $\tilde{g}_0 = \sigma(g_0) = g_0$ as shown in
 213 Equation (2.1), since g_0 is positive by the construction of Lemma 2.1. Then we have $f_1(x_i) = f_0(x_i) - \tilde{g}_0(x_i) = 0$
 214 for $i \in (I_1(m, n) - n - 1) \cup \{m(n + 1)\}$. See Figure 2 (a) for an illustration of f_0 , f_1 , and g_0 .

215 **Step 2.2:** Mathematical induction.

216 For each $k \in \{1, 2, \dots, n\}$, given f_k , we determine $\mathbf{W}_2(2k, \cdot)$, $\mathbf{b}_2(2k)$, $\mathbf{W}_2(2k + 1, \cdot)$, and $\mathbf{b}_2(2k + 1)$, to
 217 completely specify g_k^+ and g_k^- , which in turn can determine f_{k+1} . Hence, it is only enough to show how to
 218 proceed with an arbitrary k , since the initialization of the induction, f_1 , has been constructed in Step 2.1. See
 219 Figure 2 (b)-(d) for the illustration of the first two induction steps. We recursively rely on the fact of f_k that

- 220 • $f_k(x_i) = 0$ for $i \in \cup_{\ell=0}^{k-1} (I_1(m, n) - n - 1 + \ell) \cup \{m(n + 1)\}$,
- 221 • f_k is linear on each interval $[x_{i-1}, x_i]$ for $i \in I_2(m, n) \setminus \{0\}$,

222 to construct f_{k+1} satisfying similar conditions as follows:

- 223 • $f_{k+1}(x_i) = 0$ for $i \in \cup_{\ell=0}^k (I_1(m, n) - n - 1 + \ell) \cup \{m(n + 1)\}$,
- 224 • f_{k+1} is linear on each interval $[x_{i-1}, x_i]$ for $i \in I_2(m, n) \setminus \{0\}$.

225 The induction process for $\mathbf{W}_2(2k, \cdot)$, $\mathbf{b}_2(2k)$, $\mathbf{W}_2(2k + 1, \cdot)$, $\mathbf{b}_2(2k + 1)$, and f_{k+1} can be divided into four parts.

226 **Step 2.2.1:** Define index sets.

227 Let $\Lambda_k^+ = \{j : f_k(x_{j(n+1)+k}) \geq 0, 0 \leq j < m\}$ and $\Lambda_k^- = \{j : f_k(x_{j(n+1)+k}) < 0, 0 \leq j < m\}$. The cardinality
 228 of $\Lambda_k^+ \cup \Lambda_k^-$ is m . We will use Λ_k^+ and Λ_k^- to generate $2m + 1$ samples to determine CPL functions $g_k^+(x)$ and
 229 $g_k^-(x)$ in the next step.

230 **Step 2.2.2:** Determine $\mathbf{W}_2(2k, \cdot)$ and $\mathbf{b}_2(2k)$.

231 By Lemma 2.1, we can choose $\mathbf{W}_2(2k, \cdot)$ and $\mathbf{b}_2(2k)$ to fully determine $g_k^+(x)$ such that each $g_k^+(x_i)$
 232 matches a specific value for $i \in (I_1(m, n) - n - 1) \cup (I_1(m, n) - 1) \cup \{m(n + 1)\}$. The values of $\{g_k^+(x_i) : i \in$
 233 $(I_1(m, n) - n - 1) \cup (I_1(m, n) - 1) \cup \{m(n + 1)\}\}$ are specified as:

- 234 • If $j \in \Lambda_k^+$, specify the values of $g_k^+(x_{j(n+1)})$ and $g_k^+(x_{j(n+1)+n})$ such that $g_k^+(x_{j(n+1)+k-1}) = 0$ and
 235 $g_k^+(x_{j(n+1)+k}) = f_k(x_{j(n+1)+k})$. The existence of these values fulfilling the requirements above comes
 236 from the fact that $g_k^+(x)$ is linear on the interval $[x_{j(n+1)}, x_{j(n+1)+n}]$ and $g_k^+(x)$ only depends on
 237 the values of $g_k^+(x_{j(n+1)+k-1})$ and $g_k^+(x_{j(n+1)+k})$ on $[x_{j(n+1)}, x_{j(n+1)+n}]$. Now it is easy to verify that
 238 $\tilde{g}_k^+(x) := \sigma(g_k^+(x))$ satisfies $\tilde{g}_k^+(x_{j(n+1)+k}) = f_k(x_{j(n+1)+k}) \geq 0$ and $\tilde{g}_k^+(x_{j(n+1)+\ell}) = 0$ for $\ell = 0, 1, \dots, k - 1$,
 239 and \tilde{g}_k^+ is linear on each interval $[x_{j(n+1)+\ell}, x_{j(n+1)+\ell+1}]$ for $\ell = 0, 1, \dots, n - 1$.

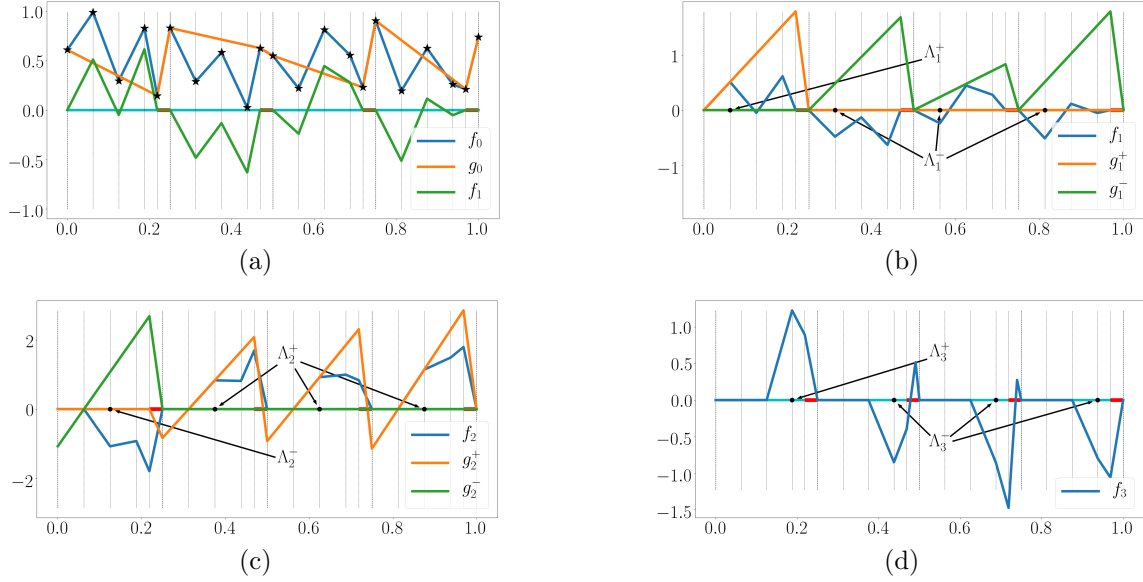


FIG. 2. Illustrations of the proof of Lemma 2.2, especially Step 2 of the proof, when $m = n = 4$, with the “don’t-care” region in red. (a) Given samples $\{(x_i, y_i) : i = 0, 1, \dots, m(n+1)\}$ marked with “star” signs, suppose $f_0(x)$ is a CPL function fitting the samples, construct g_0 such that $f_1 = f_0 - \sigma(g_0)$ is closer to 0 than f_0 in the L^∞ sense. (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 the L^∞ sense in a 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 the L^∞ sense 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 outside the “don’t-care” region as k increases.

- 240 • If $j \in \Lambda_k^-$, let $g_k^+(x_{j(n+1)}) = g_k^+(x_{j(n+1)+n}) = 0$. Then $\tilde{g}_k^+(x) = 0$ on the interval $[x_{j(n+1)}, x_{j(n+1)+n}]$.
- 241 • Finally, specify the value of $g_k^+(x)$ at $x = x_{m(n+1)}$ as 0.

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

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

- 244 • If $j \in \Lambda_k^-$, specify the values of $g_k^-(x_{j(n+1)})$ and $g_k^-(x_{j(n+1)+n})$ such that $g_k^-(x_{j(n+1)+k-1}) = 0$ and
- 245 $g_k^-(x_{j(n+1)+k}) = -f_k(x_{j(n+1)+k})$. Then $\tilde{g}_k^-(x) := \sigma(g_k^-(x))$ satisfies $\tilde{g}_k^-(x_{j(n+1)+k}) = -f_k(x_{j(n+1)+k}) > 0$
- 246 and $\tilde{g}_k^-(x_{j(n+1)+\ell}) = 0$ for $\ell = 0, 1, \dots, k-1$, and $\tilde{g}_k^-(x)$ is linear on each interval $[x_{j(n+1)+\ell}, x_{j(n+1)+\ell+1}]$
- 247 for $\ell = 0, 1, \dots, n-1$.
- 248 • If $j \in \Lambda_k^+$, let $g_k^-(x_{j(n+1)}) = g_k^-(x_{j(n+1)+n}) = 0$. Then $\tilde{g}_k^-(x) = 0$ on the interval $[x_{j(n+1)}, x_{j(n+1)+n}]$.
- 249 • Finally, specify the value of $g_k^-(x)$ at $x = x_{m(n+1)}$ as 0.

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

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

- 252 (1) $f_k(x_i) = \tilde{g}_k^+(x_i) = \tilde{g}_k^-(x_i) = 0$ for $i \in \cup_{\ell=0}^{k-1} (I_1(m, n) - n - 1 + \ell) \cup \{m(n+1)\}$;
- 253 (2) If $j \in \Lambda_k^+$, $\tilde{g}_k^+(x_{j(n+1)+k}) = f_k(x_{j(n+1)+k}) \geq 0$ and $\tilde{g}_k^-(x_{j(n+1)+k}) = 0$;
- 254 (3) If $j \in \Lambda_k^-$, $\tilde{g}_k^+(x_{j(n+1)+k}) = -f_k(x_{j(n+1)+k}) > 0$ and $\tilde{g}_k^+(x_{j(n+1)+k}) = 0$;
- 255 (4) \tilde{g}_k^+ and \tilde{g}_k^- are linear on each interval $[x_{j(n+1)+\ell}, x_{j(n+1)+\ell+1}]$ for $\ell = 0, 1, \dots, n-1$, $j \in \Lambda_k^+ \cup \Lambda_k^- = \{0, 1, \dots, m-1\}$.

256 In other words, \tilde{g}_k^+ and \tilde{g}_k^- are linear on each interval $[x_{i-1}, x_i]$ for $i \in I_2(m, n) \setminus \{0\}$.

257 See Figure 2 (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.

258 Note that $\Lambda_k^+ \cup \Lambda_k^- = \{0, 1, \dots, m-1\}$, so $f_k(x_i) - \tilde{g}_k^+(x_i) + \tilde{g}_k^-(x_i) = 0$ for $i \in \cup_{\ell=0}^k (I_1(m, n) - n - 1 + \ell) \cup \{m(n+1)\}$.

259 Now we define $f_{k+1} := f_k - \tilde{g}_k^+ + \tilde{g}_k^-$, then

- 260 • $f_{k+1}(x_i) = 0$ for $i \in \cup_{\ell=0}^k (I_1(m, n) - n - 1 + \ell) \cup \{m(n+1)\}$;

261 • f_{k+1} is linear on each interval $[x_{i-1}, x_i]$ for $i \in I_2(m, n) \setminus \{0\}$.
 262 See Figure 2 (b)-(d) for the illustration of f_1 , f_2 , and f_3 , and to verify their properties as listed just above.
 263 This finishes the mathematical induction process. As we can imagine based on Figure 2, when k increases,
 264 the support of f_k shrinks to the “don’t-care” region.

265 **Step 3:** Determine \mathbf{W}_3 and \mathbf{b}_3 .

266 With the special vector function $\mathbf{g} = [g_0, g_1^+, g_1^-, \dots, g_n^+, g_n^-]^T$ constructed in Step 2, we are able to specify \mathbf{W}_3
 267 and \mathbf{b}_3 to generate a desired $\phi(x)$ with a well-controlled L^∞ -norm matching the samples $\{(x_i, y_i)\}_{0 \leq i \leq m(n+1)}$.

268 In fact, we can simply set $\mathbf{W}_3 = [1, 1, -1, 1, -1, \dots, 1, -1] \in \mathbb{R}^{1 \times (2n+1)}$ and $\mathbf{b}_3 = 0$, which finishes the con-
 269 struction of $\phi(x)$. The rest of the proof is to verify the properties of $\phi(x)$. Note that $\phi = \tilde{g}_0 + \sum_{\ell=1}^n \tilde{g}_\ell^+ - \sum_{\ell=1}^n \tilde{g}_\ell^-$.
 270 By the mathematical induction, we have:

- 271 • $f_{n+1} = f_0 - \tilde{g}_0 - \sum_{\ell=1}^n \tilde{g}_\ell^+ + \sum_{\ell=1}^n \tilde{g}_\ell^-$;
- 272 • $f_{n+1}(x_i) = 0$ for $i \in \cup_{\ell=0}^n (I_1(m, n) - n - 1 + \ell) \cup \{m(n+1)\} = I_0(m, n)$;
- 273 • f_{n+1} is linear on each interval $[x_{i-1}, x_i]$ for $i \in I_2(m, n) \setminus \{0\}$.

274 Hence, $\phi = \tilde{g}_0 + \sum_{\ell=1}^n \tilde{g}_\ell^+ - \sum_{\ell=1}^n \tilde{g}_\ell^- = f_0 - f_{n+1}$. Then ϕ satisfies Conditions (1) and (2) of this lemma. It remains
 275 to check that ϕ satisfies Condition (3).

276 By the definition of f_1 , we have

$$277 \quad \sup_{x \in [x_0, x_{m(n+1)}]} |f_1(x)| \leq 2 \max\{y_i : i \in I_0(m, n)\}. \quad (2.2)$$

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

$$279 \quad \sup_{x \in [x_0, x_{m(n+1)}]} |\tilde{g}_k^+(x)| \leq \frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} \sup_{x \in [x_0, x_{m(n+1)}]} |f_k(x)| \quad (2.3)$$

280 and

$$281 \quad \sup_{x \in [x_0, x_{m(n+1)}]} |\tilde{g}_k^-(x)| \leq \frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} \sup_{x \in [x_0, x_{m(n+1)}]} |f_k(x)|. \quad (2.4)$$

282 Since either $\tilde{g}_k^+(x)$ or $\tilde{g}_k^-(x)$ is equal to 0 on $[0, 1]$, we have

$$283 \quad \sup_{x \in [x_0, x_{m(n+1)}]} |\tilde{g}_k^+(x) - \tilde{g}_k^-(x)| \leq \frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} \sup_{x \in [x_0, x_{m(n+1)}]} |f_k(x)|. \quad (2.5)$$

284 Note that $f_{k+1} = f_k - \tilde{g}_k^+ + \tilde{g}_k^-$, which means

$$285 \quad \begin{aligned} \sup_{x \in [x_0, x_{m(n+1)}]} |f_{k+1}(x)| &\leq \sup_{x \in [x_0, x_{m(n+1)}]} |\tilde{g}_k^+(x) - \tilde{g}_k^-(x)| + \sup_{x \in [x_0, x_{m(n+1)}]} |f_k(x)| \\ &\leq \left(\frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} + 1 \right) \sup_{x \in [x_0, x_{m(n+1)}]} |f_k(x)| \end{aligned} \quad (2.6)$$

286 for $k \in \{1, 2, \dots, n\}$. Then we have

$$287 \quad \sup_{x \in [x_0, x_{m(n+1)}]} |f_{n+1}(x)| \leq 2 \max_{i \in I_0(m, n)} y_i \prod_{k=1}^n \left(\frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} + 1 \right).$$

288 Hence

$$289 \quad \begin{aligned} \sup_{x \in [x_0, x_{m(n+1)}]} |\phi(x)| &\leq \sup_{x \in [x_0, x_{m(n+1)}]} |f_0(x)| + \sup_{x \in [x_0, x_{m(n+1)}]} |f_{n+1}(x)| \\ &\leq 3 \max_{i \in I_0(m, n)} y_i \prod_{k=1}^n \left(\frac{\max\{x_{j(n+1)+n} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}}{\min\{x_{j(n+1)+k} - x_{j(n+1)+k-1} : j=0, 1, \dots, m-1\}} + 1 \right). \end{aligned}$$

290 So, we finish the proof. \square

291 **3. Main Results.** We present our main results in this section. First, we quantitatively prove an achiev-
 292 able approximation rate in the N -term nonlinear approximation by construction, i.e., the lower bound of
 293 the approximation rate. Second, we show a lower bound of the approximation rate asymptotically, i.e., no
 294 approximant exists asymptotically following the approximation rate. Finally, we discuss the efficiency of the
 295 nonlinear approximation considering the approximation rate and parallel computing in FNNs together.

296 **3.1. Quantitative Achievable Approximation Rate.**

297 **THEOREM 3.1.** *For any $N \in \mathbb{N}^+$ and $f \in \text{Lip}(\nu, \alpha, d)$ with $\alpha \in (0, 1]$, we have^③:*

298 (1) *If $d = 1$, $\exists \phi \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 2N + 1])$ such that*

299
$$\|\phi - f\|_{L^1([0,1])} \leq 2\nu N^{-2\alpha}, \quad \text{for any } N \in \mathbb{N}^+;$$

300 (2) *If $d > 1$, $\exists \phi \in \text{NN}(\#\text{input} = d; \text{widthvec} = [2d\lfloor N^{2/d} \rfloor, 2N + 2, 2N + 3])$ such that*

301
$$\|\phi - f\|_{L^1([0,1]^d)} \leq 2(2\sqrt{d})^\alpha \nu N^{-2\alpha/d}, \quad \text{for any } N \in \mathbb{N}^+.$$

302 *Proof.* Without loss of generality, we assume $f(0) = 0$ and $\nu = 1$.

303 **Step 1:** The case $d = 1$.

304 Given any $f \in \text{Lip}(\nu, \alpha, d)$ and $N \in \mathbb{N}^+$, we know $|f(x)| \leq 1$ for any $x \in [0, 1]$ since $f(0) = 0$ and $\nu = 1$. Set
 305 $\bar{f} = f + 1 \geq 0$, then $0 \leq \bar{f}(x) \leq 2$ for any $x \in [0, 1]$. Let $X = \{\frac{i}{N^2} : i = 0, 1, \dots, N^2\} \cup \{\frac{i}{N} - \delta : i = 1, 2, \dots, N\}$,
 306 where δ is a sufficiently small positive number depending on N , and satisfying (3.2). Let us order X as
 307 $x_0 < x_1 < \dots < x_{N(N+1)}$. By Lemma 2.2, given the set of samples $\{(x_i, \bar{f}(x_i)) : i \in \{0, 1, \dots, N(N+1)\}\}$, there
 308 exists $\phi \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 2N + 1])$ such that

- 309 • $\phi(x_i) = \bar{f}(x_i)$ for $i = 0, 1, \dots, N(N+1)$;
- 310 • ϕ is linear on each interval $[x_{i-1}, x_i]$ for $i \notin \{(N+1)j : j = 1, 2, \dots, N\}$;
- 311 • ϕ has an upper bound estimation: $\sup\{\phi(x) : x \in [0, 1]\} \leq 6(N+1)!$.

312 It follows that

313
$$|\bar{f}(x) - \phi(x)| \leq (x_i - x_{i-1})^\alpha \leq N^{-2\alpha}, \quad \text{if } x \in [x_{i-1}, x_i], \quad \text{for } i \notin \{(N+1)j : j = 1, 2, \dots, N\}.$$

314 Define $H_0 = \cup_{i \in \{(N+1)j : j=1, 2, \dots, N\}} [x_{i-1}, x_i]$, then

315
$$|\bar{f}(x) - \phi(x)| \leq N^{-2\alpha}, \quad \text{for any } x \in [0, 1] \setminus H_0, \quad (3.1)$$

316 by the fact that $\bar{f} \in \text{Lip}(1, \alpha, d)$ and points in X are equispaced. By $\mu(H_0) \leq N\delta$, it follows that

317
$$\begin{aligned} \|\bar{f} - \phi\|_{L^1([0,1])} &= \int_{H_0} |\bar{f}(x) - \phi(x)| dx + \int_{[0,1] \setminus H_0} |\bar{f}(x) - \phi(x)| dx \\ &\leq N\delta(2 + 6(N+1)!) + N^2(N^{-2\alpha})N^{-2} \leq 2N^{-2\alpha}, \end{aligned}$$

318 where the last inequality comes from the fact δ is small enough satisfying

319
$$N\delta(2 + 6(N+1)!) \leq N^{-2\alpha}. \quad (3.2)$$

320 Note that $f - (\phi - 1) = f + 1 - \phi = \bar{f} - \phi$. Hence, $\phi - 1 \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [2N, 2N + 1])$ and
 321 $\|f - (\phi - 1)\| \leq 2N^{-2\alpha}$. So, we finish the proof for the case $d = 1$.

322 **Step 2:** The case $d > 1$.

323 The main idea is to project the d -dimensional problem into a one-dimensional one and use the results
 324 proved above. For any $N \in \mathbb{N}^+$, let $n = \lfloor N^{2/d} \rfloor$ and δ be a sufficiently small positive number depending on N

^③It is easy to generalize the results in Theorem 3.1 and Corollary 3.2 from L^1 to L^p -norm for $p \in [1, \infty)$ since $\mu([0, 1]^d) = 1$.

and d , and satisfying (3.8). We will divide the d -dimensional cube into n^d small non-overlapping sub-cubes (see Figure 3 for an illustration when $d = 3$ and $n = 3$), each of which is associated with a representative point, e.g., a vertex of the sub-cube. Due to the continuity, the target function f can be represented by their values at the representative points. We project these representatives to one-dimensional samples via a ReLU FNN ψ and construct a ReLU FNN $\bar{\phi}$ to fit them. Finally, the ReLU FNN ϕ on the d -dimensional space approximating f can be constructed by $\phi = \bar{\phi} \circ \psi$. The precise construction can be found below.

By Lemma 2.1, there exists $\psi_0 \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [2n])$ such that

- $\psi_0(1) = n - 1$, and $\psi_0(\frac{i}{n}) = \psi_0(\frac{i+1}{n} - \delta) = i$ for $i = 0, 1, \dots, n - 1$;
- ψ_0 is linear between any two adjacent points of $\{\frac{i}{n} : i = 0, 1, \dots, n\} \cup \{\frac{i}{n} - \delta : i = 1, 2, \dots, n\}$.

Define the projection map^④ ψ by

$$\psi(\mathbf{x}) = \sum_{i=1}^d \frac{1}{n^i} \psi_0(x_i), \quad \text{for } \mathbf{x} = [x_1, x_2, \dots, x_d]^T \in [0, 1]^d. \quad (3.3)$$

Note that $\psi \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [2dn])$. Given $f \in \text{Lip}(\nu, \alpha, d)$, then $|f(\mathbf{x})| \leq \sqrt{d}$ for $\mathbf{x} \in [0, 1]^d$ since $f(0) = 0$, $\nu = 1$, and $\alpha \in (0, 1]$. Define $\bar{f} = f + \sqrt{d}$, then $0 \leq \bar{f}(\mathbf{x}) \leq 2\sqrt{d}$ for $\mathbf{x} \in [0, 1]^d$. Hence, we have

$$\left\{ \left(\sum_{i=1}^d \frac{\theta_i}{n^i}, \bar{f}\left(\frac{\boldsymbol{\theta}}{n}\right) \right) : \boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d]^T \in \{0, 1, \dots, n-1\}^d \right\} \cup \{(1, 0)\}$$

as a set of $n^d + 1$ samples of a one-dimensional function. By Lemma 2.1, $\exists \bar{\phi} \in \text{NN}(\#\text{input} = 1; \text{widthvec} = [2\lceil n^{d/2} \rceil, 2\lceil n^{d/2} \rceil + 1])$ such that

$$\bar{\phi}\left(\sum_{i=1}^d \frac{\theta_i}{n^i}\right) = \bar{f}\left(\frac{\boldsymbol{\theta}}{n}\right), \quad \text{for } \boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d]^T \in \{0, 1, \dots, n-1\}^d, \quad (3.4)$$

and

$$\sup_{t \in [0, 1]} |\bar{\phi}(t)| \leq 6\sqrt{d}(\lceil n^{d/2} \rceil + 1)!. \quad (3.5)$$

Since the range of ψ on $[0, 1]^d$ is a subset of $[0, 1]$, $\exists \phi \in \text{NN}(\#\text{input} = d; \text{widthvec} = [2nd, 2\lceil n^{d/2} \rceil, 2\lceil n^{d/2} \rceil + 1])$ defined via $\phi(\mathbf{x}) = \bar{\phi} \circ \psi(\mathbf{x})$ for $\mathbf{x} \in [0, 1]^d$ such that

$$\sup_{\mathbf{x} \in [0, 1]^d} |\phi(\mathbf{x})| \leq 6\sqrt{d}(\lceil n^{d/2} \rceil + 1)!. \quad (3.6)$$

Define $H_1 = \cup_{j=1}^d \{\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in [0, 1]^d : x_j \in \cup_{i=1}^n [\frac{i}{n} - \delta, \frac{i}{n}]\}$, which separates the d -dimensional cube into n^d important sub-cubes as illustrated in Figure 3. To index these d -dimensional smaller sub-cubes, define $Q_{\boldsymbol{\theta}} = \{\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in [0, 1]^d : x_i \in [\frac{\theta_i}{n}, \frac{\theta_i+1}{n} - \delta], i = 1, 2, \dots, d\}$ for each d -dimensional index $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d]^T \in \{0, 1, \dots, n-1\}^d$. By (3.3), (3.4), and the definition of ψ_0 , for any $\mathbf{x} = [x_1, x_2, \dots, x_d]^T \in Q_{\boldsymbol{\theta}}$, we have $\phi(\mathbf{x}) = \bar{\phi}(\psi(\mathbf{x})) = \bar{\phi}\left(\sum_{i=1}^d \frac{1}{n^i} \psi_0(x_i)\right) = \bar{\phi}\left(\sum_{i=1}^d \frac{1}{n^i} \theta_i\right) = \bar{f}\left(\frac{\boldsymbol{\theta}}{n}\right)$. Then

$$|\bar{f}(\mathbf{x}) - \phi(\mathbf{x})| = |\bar{f}(\mathbf{x}) - \bar{f}\left(\frac{\boldsymbol{\theta}}{n}\right)| \leq (\sqrt{d}/n)^\alpha, \quad \text{for any } \mathbf{x} \in Q_{\boldsymbol{\theta}}. \quad (3.7)$$

^④The idea constructing such ψ comes from the binary representation.



FIG. 3. An illustration of H_1 and n^d small non-overlapping sub-cubes that H_1 separates when $n = 3$. (a) When $d = 2$, H_1 in blue separates $[0, 1]^2$ into $n^d = 9$ sub-cubes. (b) When $d = 3$, H_1 (no color) separates $[0, 1]^3$ into $n^d = 27$ sub-cubes in red.

353 Because $\mu(H_1) \leq dn\delta$, $[0, 1]^d = \cup_{\theta \in \{0, 1, \dots, n-1\}^d} Q_\theta \cup H_1$, (3.6), and (3.7), we have

$$\begin{aligned}
 \|\bar{f} - \phi\|_{L^1([0,1]^d)} &= \int_{H_1} |\bar{f} - \phi| d\mathbf{x} + \int_{[0,1]^d \setminus H_1} |\bar{f} - \phi| d\mathbf{x} \\
 &\leq \mu(H_1) (2\sqrt{d} + 6\sqrt{d}(\lceil n^{d/2} \rceil + 1)!) + \sum_{\theta \in \{0, 1, \dots, n-1\}^d} \int_{Q_\theta} |\bar{f} - \phi| d\mathbf{x} \\
 354 &\leq 2n\delta d\sqrt{d} (1 + 3(\lceil n^{d/2} \rceil + 1)!) + \sum_{\theta \in \{0, 1, \dots, n-1\}^d} (\sqrt{d}/n)^\alpha \mu(Q_\theta) \\
 &\leq 2d^{\alpha/2} n^{-\alpha},
 \end{aligned}$$

355 where the last inequality comes from the fact that δ is small enough such that

$$356 \quad 2n\delta d\sqrt{d} (1 + 3(\lceil n^{d/2} \rceil + 1)!) \leq d^{\alpha/2} n^{-\alpha}. \quad (3.8)$$

357 Note that $f - (\phi - \sqrt{d}) = \bar{f} - \phi$. Hence, $\phi - \sqrt{d} \in \text{NN}(\#\text{input} = d; \text{widthvec} = [2nd, 2\lceil n^{d/2} \rceil, 2\lceil n^{d/2} \rceil + 1])$
 358 and $\|f - (\phi - \sqrt{d})\|_{L^1([0,1]^d)} \leq 2d^{\alpha/2} n^{-\alpha}$. Since $n = \lfloor N^{2/d} \rfloor$, we have $\lceil n^{d/2} \rceil \leq N + 1$. Therefore,

$$359 \quad \phi - \sqrt{d} \in \text{NN}(\#\text{input} = d; \text{widthvec} = [2d\lfloor N^{2/d} \rfloor, 2N + 2, 2N + 3])$$

360 and

$$361 \quad \|f - (\phi - \sqrt{d})\|_{L^1([0,1]^d)} \leq 2d^{\alpha/2} n^{-\alpha} = 2d^{\alpha/2} \lfloor N^{2/d} \rfloor^{-\alpha} \leq 2d^{\alpha/2} (N^{2/d}/2)^{-\alpha} = 2(2\sqrt{d})^\alpha N^{-2\alpha/d},$$

362 where the second inequality comes from the fact $\lfloor x \rfloor \geq \frac{x}{2}$ for $x \in [1, \infty)$. So, we finish the proof when $d > 1$. \square

363 Theorem 3.1 shows that for $f \in \text{Lip}(\nu, \alpha, d)$ ReLU FNNs with two or three function compositions can
 364 achieve the approximation rate $\mathcal{O}(\nu N^{-2\alpha/d})$. Following the same proof as in Theorem 3.1, we can show that:

365 COROLLARY 3.2. $\forall m, n \in \mathbb{N}^+$, the closure of $\text{NN}(\#\text{input} = 1; \text{widthvec} = [2m, 2n+1])$ contains $\text{CPL}(mn+$
 366 $1)$ in the sense of L^1 -norm.

367 An immediate implication of Corollary 3.2 is that, for any function f on $[0, 1]$, if f can be approximated
 368 via one-hidden-layer ReLU FNNs with an approximation rate $\mathcal{O}(N^{-\eta})$ for any $\eta > 0$, the rate can be improved
 369 to $\mathcal{O}(N^{-2\eta})$ via one more composition.

370 **3.2. Asymptotic Unachievable Approximation Rate.** In Section 3.1, we have analyzed the approx-
 371 imation capacity of ReLU FNNs in the nonlinear approximation for general continuous functions by construc-
 372 tion. In this section, we will show that the construction in Section 3.1 is essentially and asymptotically tight
 373 via showing the approximation lower bound in Theorem 3.3 below.

374 **THEOREM 3.3.** *For any $L \in \mathbb{N}^+$, $\rho > 0$, and $C > 0$, $\exists f \in \text{Lip}(\nu, \alpha, d)$ with $\alpha \in (0, 1]$, for all $N_0 > 0$, there
 375 exists $N \geq N_0$ such that*

$$376 \quad \inf_{\phi \in \text{NN}(\#\text{input}=d; \text{maxwidth} \leq N; \#\text{layer} \leq L)} \|\phi - f\|_{L^\infty([0,1]^d)} \geq C\nu N^{-(2\alpha/d+\rho)}.$$

377 The proof of Theorem 3.3 relies on the nearly-tight VC-dimension bounds of ReLU FNNs given in (Harvey
 378 et al., 2017) and is similar to Theorem 4 of (Yarotsky, 2017). Hence, we only sketch out its proof and a complete
 379 proof can be found in (Zhang, n.d.).

380 *Proof.* We will prove this theorem by contradiction. Assuming that Theorem 3.3 is not true, we can show
 381 the following claim, which will lead to a contradiction in the end.

382 **CLAIM 3.4.** *There exist $L \in \mathbb{N}^+$, $\rho > 0$, and $C > 0$, $\forall f \in \text{Lip}(\nu, \alpha, d)$ with $\alpha \in (0, 1]$, then $\exists N_0 > 0$, for all
 383 $N \geq N_0$, there exists $\phi \in \text{NN}(\#\text{input} = d; \text{maxwidth} \leq N; \#\text{layer} \leq L)$ such that*

$$384 \quad \|f - \phi\|_{L^\infty([0,1]^d)} \leq C\nu N^{-(2\alpha/d+\rho)}.$$

385 If this claim is true, then we have a better approximation rate. So we need to disprove this claim in order to
 386 prove Theorem 3.3.

387 Without loss of generality, we assume $\nu = 1$; in the case of $\nu \neq 1$, the proof is similar by rescaling
 388 $f \in \text{Lip}(\nu, \alpha, d)$ and FNNs with ν . Let us denote the VC dimension of a function set \mathcal{F} by $\text{VCDim}(\mathcal{F})$. By
 389 (Harvey et al., 2017), there exists $C_1 > 0$ such that

$$390 \quad \begin{aligned} & \text{VCDim}(\text{NN}(\#\text{input} = d; \text{maxwidth} \leq N; \#\text{layer} \leq L)) \\ & \leq C_1((LN + d + 2)(N + 1))L \ln((LN + d + 2)(N + 1)) := b_u, \end{aligned}$$

391 which comes from the fact the number of parameter of a ReLU FNN in $\text{NN}(\#\text{input} = d; \text{maxwidth} \leq$
 392 $N; \#\text{layer} \leq L)$ is less than $(LN + d + 2)(N + 1)$.

393 One can estimate a lower bound of

$$394 \quad \text{VCDim}(\text{NN}(\#\text{input} = d; \text{maxwidth} \leq N; \#\text{layer} \leq L))$$

395 using Claim 3.4, and this lower bound can be $b_\ell := \lfloor N^{2/d+\rho/(2\alpha)} \rfloor^d$, which is asymptotically larger than

$$396 \quad b_u := C_1((LN + d + 2)(N + 1))L \ln((LN + d + 2)(N + 1)) = \mathcal{O}(N^2 \ln N),$$

397 leading to a contradiction that disproves the assumption that ‘‘Theorem 3.3 is not true’’. \square

398 Theorem 3.1 shows that the N -term approximation rate via two or three-hidden-layer ReLU FNNs can
 399 achieve $\mathcal{O}(N^{-2\alpha/d})$, while Theorem 3.3 shows that the rate cannot be improved to $\mathcal{O}(N^{-(2\alpha/d+\rho)})$ for any
 400 $\rho > 0$. It was conjectured in the literature that function compositions can improve the approximation capacity
 401 exponentially. For general continuous functions, Theorem 3.3 shows that this conjecture is not true, i.e., if
 402 the depth of the composition is $L = \mathcal{O}(1)$, the approximation rate cannot be better than $\mathcal{O}(N^{-2\alpha/d})$, not to
 403 mention $\mathcal{O}(N^{-L\alpha/d})$, which implies that adding one more layer cannot improve the approximation rate when
 404 N is large and $L > 2$.

405 Following the same proof as in Theorem 3.3, we have the following corollary, which shows that the result
 406 in Corollary 3.2 cannot be improved.

407 **COROLLARY 3.5.** *$\forall \rho > 0, C > 0$ and $L \in \mathbb{N}^+$, $\exists N_0(\rho, C, L)$ such that for any integer $N \geq N_0$, $\text{CPL}(CN^{2+\rho})$
 408 is not contained in the closure of $\text{NN}(\#\text{input} = 1; \text{maxwidth} \leq N; \#\text{layer} \leq L)$ in the sense of L^∞ -norm.*

409 **3.3. Approximation and Computation Efficiency in Parallel Computing.** In this section, we
 410 will discuss the efficiency of the N -term approximation via ReLU FNNs in parallel computing. This is of
 411 more practical interest than the optimal approximation rate purely based on the number of parameters of the
 412 nonlinear approximation since it is impractical to use FNNs without parallel computing in real applications.
 413 Without loss of generality, we assume $\nu = 1$, $N \gg 1$, and $d \gg 1$.

414 Let us summarize standard statistics of the time and memory complexity in parallel computing (Kumar,
 415 2002) in one training iteration of ReLU FNNs with $\mathcal{O}(N)$ width and $\mathcal{O}(L)$ depth using m computing cores
 416 and $\mathcal{O}(1)$ training data samples per iteration. Let $T_s(N, L, m)$ and $T_d(N, L, m)$ denote the time complexity
 417 in shared and distributed memory parallel computing, respectively. Similarly, $M_s(N, L, m)$ and $M_d(N, L, m)$
 418 are the memory complexity for shared and distributed memory, respectively. $M_s(N, L, m)$ is the total memory
 419 requirement; while $M_d(N, L, m)$ is the memory requirement per computing core. Then

$$420 \quad T_s(N, L, m) = \begin{cases} \mathcal{O}(L(N^2/m + \ln \frac{m}{N})), & m \in [1, N^2], \\ \mathcal{O}(L \ln N), & m \in (N^2, \infty); \end{cases} \quad (3.9)$$

$$421 \quad T_d(N, L, m) = \begin{cases} \mathcal{O}(L(N^2/m + t_s \ln m + \frac{t_w N}{\sqrt{m}} \ln m)), & m \in [1, N^2], \\ \mathcal{O}(L \ln N), & m \in (N^2, \infty); \end{cases} \quad (3.10)$$

$$422 \quad M_s(N, L, m) = \mathcal{O}(LN^2), \quad \text{for all } m \in \mathbb{N}^+; \quad (3.11)$$

423 and

$$424 \quad M_d(N, L, m) = \mathcal{O}(LN^2/m + 1), \quad \text{for all } m \in \mathbb{N}^+, \quad (3.12)$$

425 where t_s and t_w are the ‘‘start-up time’’ and ‘‘per-word transfer time’’ in the data communication between
 426 different computing cores, respectively (see (Kumar, 2002) for a detailed introduction).

427 In real applications, a most frequently asked question would be: given a target function $f \in \text{Lip}(\nu, \alpha, d)$, a
 428 target approximation accuracy ε , and a certain amount of computational resources, e.g., m computer proces-
 429 sors, assuming the computer memory is enough, what is a good choice of FNN architecture we should use to
 430 reduce the running time of our computers? Certainly, the answer depends on the number of processors m and
 431 ideally we hope to increase m by a factor of r to reduce the time (and memory in the distributed environment)
 432 complexity by the same factor r , which is the scalability of parallel computing.

433 We answer the question raised just above using FNN architectures that almost have a uniform width
 434 since the optimal approximation theory of very deep FNNs (Yarotsky, 2018) and this manuscript both utilize
 435 a nearly uniform width. Combining the theory in (Yarotsky, 2018) and ours, we summarize several statistics of
 436 ReLU FNNs in parallel computing in Table 1 and 2 when FNNs nearly have the same approximation accuracy.
 437 For shared memory parallel computing, from Table 1 we see that: if computing resources are enough, shallower
 438 FNNs with $\mathcal{O}(1)$ hidden layers require less and even exponentially less running time than very deep FNNs;
 439 if computing resources are limited, shallower FNNs might not be applicable or are slower, and hence very
 440 deep FNNs are a good choice. For distributed memory parallel computing, the conclusion is almost the same
 441 by Table 2, except that the memory limit is not an issue for shallower FNNs if the number of processors is
 442 large enough. In sum, if the approximation rate of very deep FNNs is not exponentially better than shallower
 443 FNNs, very deep FNNs are less efficient than shallower FNNs theoretically if computing resources are enough.

444 **4. Numerical Experiments.** In this section, we provide two sets of numerical experiments to compare
 445 different ReLU FNNs using shared memory GPU parallel computing. The numerical results for distributed
 446 memory parallel computing would be similar. All numerical tests were conducted using Tensorflow and an
 447 NVIDIA P6000 GPU with 3840 CUDA parallel processing cores.

448 Since it is difficult to generate target functions $f \in \text{Lip}(\nu, \alpha, d)$ with fixed ν and α , we cannot directly
 449 verify the nonlinear approximation rate, but we are able to observe numerical evidence close to our theoretical
 450 conclusions. Furthermore, we are able to verify the running time estimates in Section 3.3 and show that, to
 451 achieve the same theoretical approximation rate, shallow FNNs are more efficient than very deep FNNs.
 452
 453

TABLE 1

The comparison of approximation and computation efficiency of different ReLU FNNs in shared memory parallel computing with m processors when FNNs nearly have the same approximation accuracy. The analysis is asymptotic in N and is optimal up to a log factor ($N \gg d \gg 1$); “running time” in this table is the time spent on each training step with $\mathcal{O}(1)$ training samples.

	NN(width = $[2d\lfloor N^{2/d} \rfloor, 2N, 2N]$)	NN(width = $[N]^L$)	NN(width = $[2d + 10]^N$)
accuracy ε	$\mathcal{O}(\sqrt{d}N^{-2\alpha/d})$	$\mathcal{O}(C(d, L)N^{-2\alpha/d})$	$\mathcal{O}(C(d)N^{-2\alpha/d})$
number of weights	$\mathcal{O}(N^2)$	$\mathcal{O}(LN^2)$	$\mathcal{O}(d^2N)$
number of nodes	$\mathcal{O}(N)$	$\mathcal{O}(LN)$	$\mathcal{O}(dN)$
running time for $m \in [1, (2d + 10)^2]$	$\mathcal{O}(N^2/m)$	$\mathcal{O}(LN^2/m)$	$\mathcal{O}(N(d^2/m + \ln \frac{m}{d}))$
running time for $m \in ((2d + 10)^2, N^2]$	$\mathcal{O}(N^2/m + \ln \frac{m}{N})$	$\mathcal{O}(L(N^2/m + \ln \frac{m}{N}))$	$\mathcal{O}(N \ln d)$
running time for $m \in (N^2, \infty)$	$\mathcal{O}(\ln N)$	$\mathcal{O}(L \ln N)$	$\mathcal{O}(N \ln d)$
total memory	$\mathcal{O}(N^2)$	$\mathcal{O}(LN^2)$	$\mathcal{O}(d^2N)$

TABLE 2

The comparison of approximation and computation efficiency of different ReLU FNNs in distributed memory parallel computing with m processors when FNNs nearly have the same approximation accuracy. The analysis is asymptotic in N and is optimal up to a log factor ($N \gg d \gg 1$); “running time” in this table is the time spent on each training step with $\mathcal{O}(1)$ training samples.

	NN(width = $[2d\lfloor N^{2/d} \rfloor, 2N, 2N]$)	NN(width = $[N]^L$)	NN(width = $[2d + 10]^N$)
accuracy ε	$\mathcal{O}(\sqrt{d}N^{-2\alpha/d})$	$\mathcal{O}(C(d, L)N^{-2\alpha/d})$	$\mathcal{O}(C(d)N^{-2\alpha/d})$
number of weights	$\mathcal{O}(N^2)$	$\mathcal{O}(LN^2)$	$\mathcal{O}(d^2N)$
number of nodes	$\mathcal{O}(N)$	$\mathcal{O}(LN)$	$\mathcal{O}(dN)$
running time for $m \in [1, (2d + 10)^2]$	$\mathcal{O}(N^2/m + t_s \ln m + \frac{t_w N}{\sqrt{m}} \ln m)$	$\mathcal{O}(L(N^2/m + t_s \ln m + \frac{t_w N}{\sqrt{m}} \ln m))$	$\mathcal{O}(N(d^2/m + t_s \ln m + \frac{t_w d}{\sqrt{m}} \ln m))$
running time for $m \in ((2d + 10)^2, N^2]$	$\mathcal{O}(N^2/m + t_s \ln m + \frac{t_w N}{\sqrt{m}} \ln m)$	$\mathcal{O}(L(N^2/m + t_s \ln m + \frac{t_w N}{\sqrt{m}} \ln m))$	$\mathcal{O}(N \ln d)$
running time for $m \in (N^2, \infty)$	$\mathcal{O}(\ln N)$	$\mathcal{O}(L \ln N)$	$\mathcal{O}(N \ln d)$
memory per processor	$\mathcal{O}(N^2/m + 1)$	$\mathcal{O}(LN^2/m + 1)$	$\mathcal{O}(d^2N/m + 1)$

454 In our numerical tests, we generate 50 random smooth functions as our target functions using the algorithm
 455 in (Filip, Javeed, & Trefethen, 2018) with a wavelength parameter $\lambda = 0.1$ and an amplitude parameter $\sqrt{(2/\lambda)}$
 456 therein. These target functions are uniformly sampled with 20000 ordered points $\{x_i\}$ in $[0, 1]$ to form a data
 457 set. The training data set consists of samples with odd indices i 's, while the test data set consists of samples
 458 with even indices i 's. The loss function is defined as the mean square error between the target function and
 459 the FNN approximant evaluated on training sample points. The ADAM algorithm (Kingma & Ba, 2014)
 460 with a decreasing learning rate from 0.005 to 0.0005, a batch size 10000, and a maximum number of epochs
 461 20000, is applied to minimize the mean square error. The minimization is randomly initialized by the “normal
 462 initialization method”^⑤. The test error is defined as the mean square error evaluated on test sample points.
 463 The training and test data sets are essentially the same in our numerical test since we aim at studying the
 464 approximation power of FNNs instead of the generalization capacity of FNNs. Note that due to the high
 465 non-convexity of the optimization problem, there might be chances such that the minimizers we found are bad
 466 local minimizers. Hence, we compute the average test error of the best 40 tests among the total 50 tests of
 467 each architecture.

468 To observe numerical phenomena in terms of N -term nonlinear approximation, in the first set of numerical
 469 experiments, we use two types of FNNs to obtain approximants: the first type has $L = \mathcal{O}(1)$ layers with different
 470 sizes of width N ; the second type has a fixed width $N = 12$ with different numbers of layers L . Numerical
 471 results are summarized in Table 3. To observe numerical phenomena in terms of the number of parameters
 472 in FNNs, in the second set of numerical experiments, we use FNNs with the same number of parameters but
 473 different sizes of width N and different numbers of layers L . Numerical results are summarized in Table 4.

474 By the last columns of Table 3, we verified that as long as the number of computing cores m is larger
 475 than or equal to N^2 , the running time per iteration of FNNs with $\mathcal{O}(1)$ layers is $\mathcal{O}(\ln N)$, while the running

^⑤See <https://medium.com/prateekvishnu/xavier-and-he-normal-he-et-al-initialization-8e3d7a087528>.

TABLE 3

Comparison between NN(#input = 1; widthvec = $[N]^L$) and NN(#input = 1; widthvec = $[12]^N$) for $N = 32, 64, 128$ and $L = 2, 4, 8$. "Time" in this table is the total running time spent on 20000 training steps with training batch size 10000, and the unit is second(s).

N	depth	width	test error	improvement ratio	#parameter	time
32	2	32	8.06×10^{-2}	–	1153	3.09×10^1
32	4	32	3.98×10^{-4}	–	3265	3.82×10^1
32	8	32	1.50×10^{-5}	–	7489	5.60×10^1
32	32	12	1.29×10^{-3}	–	4873	1.27×10^2
64	2	64	2.51×10^{-2}	3.21	4353	3.45×10^1
64	4	64	4.27×10^{-5}	9.32	12673	5.00×10^1
64	8	64	2.01×10^{-6}	7.46	29313	7.91×10^1
64	64	12	1.16×10^{-1}	0.01	9865	2.37×10^2
128	2	128	2.04×10^{-3}	12.3	16897	5.03×10^1
128	4	128	1.05×10^{-5}	4.07	49921	8.21×10^1
128	8	128	1.47×10^{-6}	1.37	115969	1.41×10^2
128	128	12	3.17×10^{-1}	0.37	19849	4.47×10^2

time per iteration of FNNs with $\mathcal{O}(N)$ layers and $\mathcal{O}(1)$ width is $\mathcal{O}(N)$. By the last columns of Table 4, we see that when the number of parameters is the same, very deep FNNs requires much more running time per iteration than shallower FNNs and the difference becomes more significant when the number of parameters increases. Hence, very deep FNNs are much less efficient than shallower FNNs in parallel computing.

Besides, by Table 3 and 4, the test error of very deep FNNs cannot be improved if the depth is increased and the error even becomes larger when depth is larger. However, when the number of layers is fixed, increasing width can reduce the test error. More quantitatively, we define the *improvement ratio* of an FNN with width N and depth L in Table 3 as the ratio of the test error of an FNN in NN(#input = 1; widthvec = $[N/2]^L$) (or NN(#input = 1; widthvec = $[N]^{L/2}$)) over the test error of the current FNN in NN(#input = 1; widthvec = $[N]^L$). Similarly, the improvement ratio of an FNN with a number of parameters W in Table 4 is defined as the ratio of the test error of an FNN with the same type of architecture and a number of parameters $W/2$ over the test error of the current FNN. According to the improvement ratio in Table 3 and 4, when $L = \mathcal{O}(1)$, the numerical approximation rate in terms of N is in a range between 2 to 4. Due to the high non-convexity of the deep learning optimization and the difficulty to generate target functions of the same class with a fixed order α and constant ν , we cannot accurately verify the approximation rate. But the statistics of the improvement ratio can roughly reflect the approximation rate and the numerical results stand in line with our theoretical analysis.

5. Conclusions. We studied the approximation and computation efficiency of nonlinear approximation via compositions, especially when the dictionary \mathcal{D}_L consists of deep ReLU FNNs with width N and depth L . Our main goal is to quantify the best N -term approximation rate $\varepsilon_{L,f}(N)$ for the dictionary \mathcal{D}_L when f is a Hölder continuous function. This topic is related to several existing approximation theories in the literature, but they cannot be applied to answer our problem. By introducing new analysis techniques that are merely based on the FNN structure instead of traditional approximation basis functions used in existing work, we have established a new theory to address our problem.

In particular, for any function f on $[0, 1]$, regardless of its smoothness and even the continuity, if f can be approximated using a dictionary when $L = 1$ with the best N -term approximation rate $\varepsilon_{L,f} = \mathcal{O}(N^{-\eta})$, we showed that dictionaries with $L = 2$ can improve the best N -term approximation rate to $\varepsilon_{L,f} = \mathcal{O}(N^{-2\eta})$. We also showed that for Hölder continuous functions of order α on $[0, 1]^d$, the application of a dictionary with $L = 3$

TABLE 4

Comparison between shallow FNNs and deep FNNs when the total number of parameters (#parameter) is fixed. “Time” in this table is the total running time spent on 20000 training steps with training batch size 10000, and the unit is second(s).

#parameter	depth	width	test error	improvement ratio	time
5038	2	69	1.13×10^{-2}	–	3.84×10^1
5041	4	40	1.65×10^{-4}	–	3.80×10^1
4993	8	26	1.69×10^{-5}	–	5.07×10^1
5029	33	12	4.77×10^{-3}	–	1.28×10^2
9997	2	98	4.69×10^{-3}	2.41	4.40×10^1
10090	4	57	7.69×10^{-5}	2.14	4.67×10^1
9954	8	37	7.43×10^{-6}	2.27	5.92×10^1
10021	65	12	2.80×10^{-1}	0.02	2.31×10^2
19878	2	139	1.43×10^{-3}	3.28	5.18×10^1
20170	4	81	2.30×10^{-5}	3.34	6.26×10^1
20194	8	53	2.97×10^{-6}	2.50	7.08×10^1
20005	129	12	3.17×10^{-1}	0.88	4.30×10^2

504 in nonlinear approximation can achieve an essentially tight best N -term approximation rate $\varepsilon_{L,f} = \mathcal{O}(N^{-2\alpha/d})$,
 505 and increasing L further cannot improve the approximation rate in N . Finally, we showed that dictionaries
 506 consisting of wide FNNs with a few hidden layers are more attractive in terms of computational efficiency than
 507 dictionaries with narrow and very deep FNNs for approximating Hölder continuous functions if the number
 508 of computer cores is larger than N in parallel computing.

509 Our results were based on the L^1 -norm in the analysis of constructive approximations in Section 3.1;
 510 while we used L^∞ -norm in the unachievable approximation rate in Section 3.2. In fact, we can define a new
 511 norm that is weaker than the L^∞ -norm and stronger than the L^1 -norm such that all theorems in this paper
 512 hold in the same norm. The analysis based on the new norm can be found in (Zhang, n.d.) and our future
 513 work, and it shows that the approximation rate in this paper is tight in the same norm. Finally, although the
 514 current result is only valid for Hölder continuous functions, it is easy to generalize it to general continuous
 515 functions by introducing the moduli of continuity, which is also left as future work.

516 **Acknowledgments.** H. Yang thanks the Department of Mathematics at the National University of Singa-
 517 pore for the startup grant, the Ministry of Education in Singapore for the grant MOE2018-T2-2-147, NVIDIA
 518 Corporation for the donation of the Titan Xp GPU, and NSCC (“The computational work for this article was
 519 partially performed on resources of the National Supercomputing Centre, Singapore (<https://www.nscg.sg>)”,
 520 n.d.) for extra GPU resource. We would like to acknowledge the editor and reviewers for their help in improving
 521 this manuscript.

522 References.

- 523 Anthony, M., & Bartlett, P. L. (2009). *Neural network learning: Theoretical foundations* (1st ed.). New York,
 524 NY, USA: Cambridge University Press.
- 525 Barron, A. R. (1993, May). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE*
 526 *Transactions on Information Theory*, 39(3), 930-945. doi: 10.1109/18.256500
- 527 Bartlett, P., Maiorov, V., & Meir, R. (1998). Almost linear VC dimension bounds for piecewise polynomial
 528 networks. *Neural Computation*, 10, 217–3.
- 529 Bianchini, M., & Scarselli, F. (2014, Aug). On the complexity of neural network classifiers: A comparison
 530 between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*,
 531 25(8), 1553-1565. doi: 10.1109/TNNLS.2013.2293637

- 532 Candes, E. J., & Wakin, M. B. (2008, March). An introduction to compressive sampling. *IEEE Signal*
533 *Processing Magazine*, 25(2), 21-30. doi: 10.1109/MSP.2007.914731
- 534 Chen, S., & Donoho, D. (1994, Oct). Basis pursuit. In *Proceedings of 1994 28th asilomar conference on*
535 *signals, systems and computers* (Vol. 1, p. 41-44 vol.1). doi: 10.1109/ACSSC.1994.471413
- 536 Cireřan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance
537 convolutional neural networks for image classification. In *Proceedings of the twenty-second international*
538 *joint conference on artificial intelligence - volume volume two* (pp. 1237–1242). AAAI Press. Retrieved
539 from <http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-210>
- 540 The computational work for this article was partially performed on resources of the national supercomputing
541 centre, singapore (<https://www.nscg.sg>). (n.d.).
- 542 Costarelli, D., & Sambucini, A. R. (2017). Saturation classes for max-product neural network operators
543 activated by sigmoidal functions. *Results in Mathematics*, 72(3), 1555 - 1569. doi: 10.1007/s00025-017
544 -0692-6
- 545 Costarelli, D., & Vinti, G. (2017). Convergence for a family of neural network operators in orlicz spaces.
546 *Mathematische Nachrichten*, 290(2-3), 226-235. Retrieved from [https://onlinelibrary.wiley.com/](https://onlinelibrary.wiley.com/doi/abs/10.1002/mana.201600006)
547 [doi/abs/10.1002/mana.201600006](https://onlinelibrary.wiley.com/doi/abs/10.1002/mana.201600006)
- 548 Costarelli, D., & Vinti, G. (2018). Approximation results in orlicz spaces for sequences of kantorovich max-
549 product neural network operators. *Results in Mathematics*, 73(1), 1 - 15. doi: 10.1007/s00025-018-0799
550 -4
- 551 Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *MCSS*, 2, 303-314.
- 552 Daubechies, I. (1992). *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics. Retrieved
553 from <https://epubs.siam.org/doi/abs/10.1137/1.9781611970104>
- 554 Davis, G. (1994). *Adaptive nonlinear approximations*.
- 555 DEVORE, R., & RON, A. (2010). Approximation using scattered shifts of a multivariate function.
556 *Transactions of the American Mathematical Society*, 362(12), 6205–6229. Retrieved from [http://](http://www.jstor.org/stable/40997201)
557 www.jstor.org/stable/40997201
- 558 DeVore, R. A. (1998). Nonlinear approximation. *Acta Numerica*, 7, 51–150. doi: 10.1017/S0962492900002816
- 559 Donoho, D. L. (2006, April). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 1289-
560 1306. doi: 10.1109/TIT.2006.871582
- 561 Duchi, J., Hazan, E., & Singer, Y. (2011, July). Adaptive subgradient methods for online learning and
562 stochastic optimization. *J. Mach. Learn. Res.*, 12, 2121–2159. Retrieved from [http://dl.acm.org/](http://dl.acm.org/citation.cfm?id=1953048.2021068)
563 [citation.cfm?id=1953048.2021068](http://dl.acm.org/citation.cfm?id=1953048.2021068)
- 564 E, W., & Wang, Q. (2018). Exponential convergence of the deep neural network approximation for analytic
565 functions. *CoRR*, *abs/1807.00297*. Retrieved from <http://arxiv.org/abs/1807.00297>
- 566 Filip, S.-I., Javeed, A., & Trefethen, L. N. (2018, December). *Smooth random functions, random ODEs,*
567 *and Gaussian processes*. Retrieved from <https://hal.inria.fr/hal-01944992> (To appear in SIAM
568 Review.)
- 569 Fukushima, K. (1980, Apr 01). Neocognitron: A self-organizing neural network model for a mechanism of
570 pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202. Retrieved
571 from <https://doi.org/10.1007/BF00344251>
- 572 Hangelbroek, T., & Ron, A. (2010). Nonlinear approximation using gaussian kernels. *Journal of Functional*
573 *Analysis*, 259(1), 203 - 219. Retrieved from [http://www.sciencedirect.com/science/article/pii/](http://www.sciencedirect.com/science/article/pii/S0022123610000467)
574 [S0022123610000467](http://www.sciencedirect.com/science/article/pii/S0022123610000467)
- 575 Harvey, N., Liaw, C., & Mehrabian, A. (2017, 07–10 Jul). Nearly-tight VC-dimension bounds for piecewise
576 linear neural networks. In S. Kale & O. Shamir (Eds.), *Proceedings of the 2017 conference on learning*
577 *theory* (Vol. 65, pp. 1064–1068). Amsterdam, Netherlands: PMLR. Retrieved from [http://proceedings](http://proceedings.mlr.press/v65/harvey17a.html)
578 [.mlr.press/v65/harvey17a.html](http://proceedings.mlr.press/v65/harvey17a.html)
- 579 Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approxi-
580 mators. *Neural Networks*, 2(5), 359 - 366. Retrieved from [http://www.sciencedirect.com/science/](http://www.sciencedirect.com/science/article/pii/0893608089900208)
581 [article/pii/0893608089900208](http://www.sciencedirect.com/science/article/pii/0893608089900208)

- 582 Jiang, J. (1996). Design of neural networks for lossless data compression. *Optical Engineering*, 35, 35 - 35 -
583 7. Retrieved from <https://doi.org/10.1117/1.600614>
- 584 Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction.
585 In *Proceedings of the 26th international conference on neural information processing systems - volume*
586 *1* (pp. 315–323). USA: Curran Associates Inc. Retrieved from [http://dl.acm.org/citation.cfm?id=](http://dl.acm.org/citation.cfm?id=2999611.2999647)
587 [2999611.2999647](http://dl.acm.org/citation.cfm?id=2999611.2999647)
- 588 Joutsensalo, J. (1994, June). Nonlinear data compression and representation by combining self-organizing map
589 and subspace rule. In *Proceedings of 1994 ieee international conference on neural networks (icnn'94)*
590 (Vol. 2, p. 637-640 vol.2). doi: 10.1109/ICNN.1994.374249
- 591 Kawaguchi, K. (2016). Deep learning without poor local minima. In D. D. Lee, M. Sugiyama, U. V. Luxburg,
592 I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems 29* (pp. 586–594). Cur-
593 ran Associates, Inc. Retrieved from [http://papers.nips.cc/paper/6112-deep-learning-without](http://papers.nips.cc/paper/6112-deep-learning-without-poor-local-minima.pdf)
594 [-poor-local-minima.pdf](http://papers.nips.cc/paper/6112-deep-learning-without-poor-local-minima.pdf)
- 595 Kawaguchi, K., & Bengio, Y. (2018). Depth with nonlinearity creates no bad local minima in resnets. Retrieved
596 from <https://arxiv.org/abs/1810.09038>
- 597 Kearns, M. J., & Schapire, R. E. (1994, June). Efficient distribution-free learning of probabilistic concepts.
598 *J. Comput. Syst. Sci.*, 48(3), 464–497. Retrieved from [http://dx.doi.org/10.1016/S0022-0000\(05\)](http://dx.doi.org/10.1016/S0022-0000(05)80062-5)
599 [80062-5](http://dx.doi.org/10.1016/S0022-0000(05)80062-5)
- 600 Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.
601 Retrieved from <http://arxiv.org/abs/1412.6980>
- 602 Kumar, V. (2002). *Introduction to parallel computing* (2nd ed.). Boston, MA, USA: Addison-Wesley Longman
603 Publishing Co., Inc.
- 604 Lewicki, G., & Marino, G. (2004). Approximation of functions of finite variation by superpositions of a
605 sigmoidal function. *Applied Mathematics Letters*, 17(10), 1147 - 1152. Retrieved from [http://www](http://www.sciencedirect.com/science/article/pii/S089396590481694X)
606 [.sciencedirect.com/science/article/pii/S089396590481694X](http://www.sciencedirect.com/science/article/pii/S089396590481694X)
- 607 Liang, S., & Srikant, R. (2016). Why deep neural networks? *CoRR*, *abs/1610.04161*. Retrieved from
608 <http://arxiv.org/abs/1610.04161>
- 609 Lin, S., Liu, X., Rong, Y., & Xu, Z. (2014, May 01). Almost optimal estimates for approximation and learning
610 by radial basis function networks. *Machine Learning*, 95(2), 147–164. Retrieved from [https://doi.org/](https://doi.org/10.1007/s10994-013-5406-z)
611 [10.1007/s10994-013-5406-z](https://doi.org/10.1007/s10994-013-5406-z) doi: 10.1007/s10994-013-5406-z
- 612 Llanas, B., & Sainz, F. (2006). Constructive approximate interpolation by neural networks. *Journal of Compu-*
613 *tational and Applied Mathematics*, 188(2), 283 - 308. Retrieved from [http://www.sciencedirect.com/](http://www.sciencedirect.com/science/article/pii/S0377042705002566)
614 [science/article/pii/S0377042705002566](http://www.sciencedirect.com/science/article/pii/S0377042705002566)
- 615 Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The expressive power of neural networks: A view from
616 the width. In (Vol. *abs/1709.02540*). Retrieved from <http://arxiv.org/abs/1709.02540>
- 617 Mallat, S. G., & Zhang, Z. (1993, Dec). Matching pursuits with time-frequency dictionaries. *IEEE Transactions*
618 *on Signal Processing*, 41(12), 3397-3415. doi: 10.1109/78.258082
- 619 Montanelli, H., & Du, Q. (2017). New error bounds for deep networks using sparse grids. Retrieved from
620 <https://arxiv.org/abs/1712.08688>
- 621 Montanelli, H., Yang, H., & Du, Q. (2019). Deep relu networks overcome the curse of dimensionality for
622 bandlimited functions. Retrieved from <https://arxiv.org/abs/1903.00735v1>
- 623 Montufar, G. F., Pascanu, R., Cho, K., & Bengio, Y. (2014). On the number of linear regions of deep
624 neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger
625 (Eds.), *Advances in neural information processing systems 27* (pp. 2924–2932). Curran Associates,
626 Inc. Retrieved from [http://papers.nips.cc/paper/5422-on-the-number-of-linear-regions-of](http://papers.nips.cc/paper/5422-on-the-number-of-linear-regions-of-deep-neural-networks.pdf)
627 [-deep-neural-networks.pdf](http://papers.nips.cc/paper/5422-on-the-number-of-linear-regions-of-deep-neural-networks.pdf)
- 628 Nguyen, Q. N., & Hein, M. (2017). The loss surface of deep and wide neural networks. *CoRR*, *abs/1704.08045*.
629 Retrieved from <http://arxiv.org/abs/1704.08045>
- 630 Ohlsson, H., Yang, A. Y., Dong, R., & Sastry, S. S. (2013, Nov). Nonlinear basis pursuit. In *2013 asilomar*
631 *conference on signals, systems and computers* (p. 115-119). doi: 10.1109/ACSSC.2013.6810285

- 632 Petersen, P., & Voigtlaender, F. (2018). Optimal approximation of piecewise smooth functions using deep
633 ReLU neural networks. *Neural Networks*, 108, 296 - 330. Retrieved from [http://www.sciencedirect](http://www.sciencedirect.com/science/article/pii/S0893608018302454)
634 [.com/science/article/pii/S0893608018302454](http://www.sciencedirect.com/science/article/pii/S0893608018302454)
- 635 Petrushev, P. (2003). Multivariate n-term rational and piecewise polynomial approximation. *Journal of*
636 *Approximation Theory*, 121(1), 158 - 197. Retrieved from [http://www.sciencedirect.com/science/](http://www.sciencedirect.com/science/article/pii/S0021904502000606)
637 [article/pii/S0021904502000606](http://www.sciencedirect.com/science/article/pii/S0021904502000606) doi: [https://doi.org/10.1016/S0021-9045\(02\)00060-6](https://doi.org/10.1016/S0021-9045(02)00060-6)
- 638 Rumelhart, D., McClelland, J., Group, P. R., & University of California, S. D. P. R. G. (1986). *Psychological*
639 *and biological models* (No. v. 2). MIT Press. Retrieved from [https://books.google.com.sg/books](https://books.google.com.sg/books?id=davmLgzusB8C)
640 [?id=davmLgzusB8C](https://books.google.com.sg/books?id=davmLgzusB8C)
- 641 Sakurai, A. (1999). Tight bounds for the VC-dimension of piecewise polynomial networks. In *Advances in*
642 *neural information processing systems* (pp. 323–329). Neural information processing systems foundation.
- 643 Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures
644 for object recognition. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial neural networks –*
645 *icann 2010* (pp. 92–101). Berlin, Heidelberg: Springer Berlin Heidelberg.
- 646 Suzuki, T. (2019). Adaptivity of deep reLU network for learning in besov and mixed smooth besov spaces: op-
647 timal rate and curse of dimensionality. In *International conference on learning representations*. Retrieved
648 from <https://openreview.net/forum?id=H1ebTsActm>
- 649 Tariyal, S., Majumdar, A., Singh, R., & Vatsa, M. (2016). Greedy deep dictionary learning. *CoRR*,
650 [abs/1602.00203](http://arxiv.org/abs/1602.00203). Retrieved from <http://arxiv.org/abs/1602.00203>
- 651 Werbos, P. (1975). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*.
652 Harvard University. Retrieved from <https://books.google.com.sg/books?id=z81XmgEACAAJ>
- 653 Xie, T. F., & Cao, F. L. (2013, Feb 01). The rate of approximation of gaussian radial basis neural networks
654 in continuous function space. *Acta Mathematica Sinica, English Series*, 29(2), 295–302. Retrieved from
655 <https://doi.org/10.1007/s10114-012-1369-4>
- 656 Yarotsky, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94, 103
657 - 114. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0893608017301545>
- 658 Yarotsky, D. (2018, 06–09 Jul). Optimal approximation of continuous functions by very deep ReLU networks.
659 In S. Bubeck, V. Perchet, & P. Rigollet (Eds.), *Proceedings of the 31st conference on learning theory*
660 (Vol. 75, pp. 639–649). PMLR. Retrieved from [http://proceedings.mlr.press/v75/yarotsky18a](http://proceedings.mlr.press/v75/yarotsky18a.html)
661 [.html](http://proceedings.mlr.press/v75/yarotsky18a.html)
- 662 Zhang, S. (n.d.). Approximation theories for deep neural networks via function compositions. *Ph.D. Thesis*
663 *submitted to the National University of Singapore*.