Л.Н. Гумилев атындағы Еуразия ұлттық университетінің хабаршысы. Математика. Компьютерлік ғылымдар. Механика сериясы, 2020, том 133, №4, 8-18 беттер
http://bulmathmc.enu.kz, E-mail: vest_math@enu.kz

IRSTI: 27.41.17, 28.23.37

D. Dũng [1], V. K. Nguyen [2], M. X. Thao [3]

[1] *Vietnam National University, Information Technology Institute*
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
[2] *Faculty of Basic Sciences, University of Transport and Communications*
No.3 Cau Giay Street, Lang Thuong Ward, Dong Da District, Hanoi, Vietnam
[3] *Department of Natural Sciences, Hong Duc University*
565 Quang Trung, Thanh Hoa, Vietnam
(E-mail: [1] dinhzung@gmail.com, [2] kiennv@utc.edu.vn, [3] maixuanthao@hdu.edu.vn)

ON COMPUTATION COMPLEXITY OF HIGH-DIMENSIONAL APPROXIMATION BY DEEP ReLU NEURAL NETWORKS

Abstract: We investigate computation complexity of deep ReLU neural networks for approximating functions in Hölder-Nikol'skii spaces of mixed smoothness $H_\infty^\alpha(\mathbb{I}^d)$ on the unit cube $\mathbb{I}^d := [0,1]^d$. For any function $f \in H_\infty^\alpha(\mathbb{I}^d)$, we explicitly construct nonadaptive and adaptive deep ReLU neural networks having an output that approximates f with a prescribed accuracy ε, and prove dimension-dependent bounds for the computation complexity of this approximation, characterized by the size and depth of this deep ReLU neural network, explicitly in d and ε. Our results show the advantage of the adaptive method of approximation by deep ReLU neural networks over nonadaptive one.

Keywords: Deep ReLU neural network; computation complexity; high-dimensional approximation; Hölder-Nikol'skii space of mixed smoothness.

DOI: https://doi.org/10.32523/2616-7182/2020-133-4-8-18

1. Introduction

In recent years, deep neural networks have been successfully applied to a striking variety of Machine Learning problems, including computer vision [14], natural language processing [25], speech recognition and image classification [15]. In approximation theory, there has been a number of interesting papers that address the role of depth and architecture of deep neural networks in approximating sets of functions which have a very special regularity properties such as analytic functions [7, 16], differentiable functions [19, 26], oscillatory functions [12], functions in isotropic Sobolev or Besov spaces [1, 6, 10, 13, 27], functions with dominating mixed smoothness [17, 23] or in approximating solutions to partial differential equations [9, 18, 22], to mention just a few. The main advantage of deep neural networks in approximation functions is that that they can output compositions of functions cheaply and consequently improve the convergence rate of approximation error, see [6, 7, 26]. We refer the reader to recent surveys [12, 20] for concept and results in deep neural network approximation theory.

In the recent paper [4], we have studied the approximation by deep ReLU neural networks, of functions from the Hölder-Zygmund space of mixed smoothness defined on the unit cube $\mathbb{I}^d := [0,1]^d$ when the dimension d may be very large. The approximation error is measured in the norm of the isotropic Sobolev space. For any function f from Hölder-Zygmund space of mixed smoothness, we explicitly construct a deep ReLU neural network having an output that approximates f with a prescribed accuracy ε, and prove tight dimension-dependent estimates of the computation complexity of this approximation, characterized as the size and depth of this deep ReLU neural network, explicitly in d and ε.

As a continuation of this paper the present paper investigates nonadaptive and adaptive high-dimensional approximation by deep ReLU neural networks for functions from Hölder-Nikol'skii spaces of mixed smoothness $H_\infty^\alpha(\mathbb{I}^d)$ on the unit cube $\mathbb{I}^d$. The approximation error is measured in the norm of $L_\infty(\mathbb{I}^d)$. In this context, we pay attention on the computation complexity of the deep ReLU networks, characterized by the size and depth of this deep ReLU neural network, explicitly in d and tolerance ε. A key tool for explicit construction of approximation methods by deep ReLU networks for functions in $H_\infty^\alpha(\mathbb{I}^d)$ is truncations of tensorized Faber series.

The space $H_\infty^\alpha(\mathbb{I}^d)$ of our interest is defined as follows. For univariate functions f on $\mathbb{I} := [0,1]$, the difference operator Δ_h is defined by

$$\Delta_h f(x) := f(x+h) - f(x),$$

for all x and $h \geq 0$ such that $x, x+h \in \mathbb{I}$. If u is a subset of $\{1,\ldots,d\}$, for multivariate functions f on $\mathbb{I}^d$ the mixed difference operator $\Delta_{\boldsymbol{h},u}$ is defined by

$$\Delta_{\boldsymbol{h},u} := \prod_{i\in u} \Delta_{h_i}, \quad \Delta_{\boldsymbol{h},\varnothing} = \mathrm{Id},$$

for all $\boldsymbol{x}$ and $\boldsymbol{h}$ such that $\boldsymbol{x}, \boldsymbol{x}+\boldsymbol{h} \in \mathbb{I}^d$. Here the univariate operator Δ_{h_i} is applied to the univariate function f by considering f as a function of variable x_i with the other variables held fixed. If $0 < \alpha \leq 1$, we introduce the semi-norm $|f|_{H_\infty^\alpha(u)}$ for functions $f \in C(\mathbb{I}^d)$ by

$$|f|_{H_\infty^\alpha(u)} := \sup_{\boldsymbol{h}>0} \prod_{i\in u} h_i^{-\alpha} \|\Delta_{\boldsymbol{h},u}(f)\|_{C(\mathbb{I}^d(\boldsymbol{h},u))}$$

(in particular, $|f|_{H_\infty^\alpha(\varnothing)} = \|f\|_{C(\mathbb{I}^d)}$), where $\mathbb{I}^d(\boldsymbol{h},u) := \{\boldsymbol{x} \in \mathbb{I}^d : x_i + h_i \in \mathbb{I}, i \in u\}$. The Hölder-Nikol'skii space $H_\infty^\alpha(\mathbb{I}^d)$ of mixed smoothness α then is defined as the set of functions $f \in C(\mathbb{I}^d)$ for which the norm

$$\|f\|_{H_\infty^\alpha(\mathbb{I}^d)} := \max_{u\subset\{1,\ldots,d\}} |f|_{H_\infty^\alpha(u)}$$

is finite. From the definition we have that $H_\infty^\alpha(\mathbb{I}^d) \subset C(\mathbb{I}^d)$. Denote by $\mathring{C}(\mathbb{I}^d)$ the set of all functions $f \in C(\mathbb{I}^d)$ vanishing on the boundary $\partial\mathbb{I}^d$ of $\mathbb{I}^d$, i.e., the set of all functions $f \in C(\mathbb{I}^d)$ such that $f(x) = 0$ if $x_j = 0$ or $x_j = 1$ for some index $j \in \{1,\ldots,d\}$. Denote by $\mathring{U}_\infty^{\alpha,d}$ the set of all functions f in the intersection $\mathring{H}_\infty^\alpha(\mathbb{I}^d) := H_\infty^\alpha(\mathbb{I}^d) \cap \mathring{C}(\mathbb{I}^d)$ such that $\|f\|_{H_\infty^\alpha(\mathbb{I}^d)} \leq 1$.

Notation. As usual, $\mathbb{N}$ is the natural numbers, $\mathbb{Z}$ is the integers, $\mathbb{R}$ is the real numbers and $\mathbb{N}_0 := \{s \in \mathbb{Z} : s \geq 0\}$; $\mathbb{N}_{-1} = \mathbb{N}_0 \cup \{-1\}$. The letter d is reserved for the underlying dimension of $\mathbb{R}^d$, $\mathbb{N}^d$, etc. Vectorial quantities are denoted by boldface letters and x_i denotes the ith coordinate of $\boldsymbol{x} \in \mathbb{R}^d$, i.e., $\boldsymbol{x} := (x_1,\ldots,x_d)$. For $\boldsymbol{x} \in \mathbb{R}^d$, we denote $|\boldsymbol{x}|_1 := |x_1| + \ldots + |x_d|$ and $2^{\boldsymbol{x}} := (2^{x_1},\ldots,2^{x_d})$. For $\boldsymbol{k},\boldsymbol{s} \in \mathbb{N}_0^d$, we denote $2^{-\boldsymbol{k}}\boldsymbol{s} := (2^{-k_1}s_1,\ldots,2^{-k_d}s_d)$. Universal constants or constants depending on parameter α are denoted by C or C_α, respectively.

2. Deep ReLU neural networks

In this section we introduce necessary definitions and elementary facts on deep ReLU neural networks. There is a wide variety of neural network architectures and each of them is adapted to specific tasks. We only consider feed-forward deep ReLU neural networks for which only connections between neighboring layers are allowed.

Definition 1. Let $d, L \in \mathbb{N}$, $L \geq 2$, $N_0 = d$, and $N_1,\ldots,N_L \in \mathbb{N}$. Let $\boldsymbol{W}^\ell = (w_{i,j}^\ell)$, $\ell = 1,\ldots,L$, be $N_\ell \times N_{\ell-1}$ matrix, and $\boldsymbol{b}^\ell = (b_j^\ell) \in \mathbb{R}^{N_\ell}$.

- A neural network Φ with input dimension d and L layers is a sequence of matrix-vector tuples

$$\Phi = \big((\boldsymbol{W}^1,\boldsymbol{b}^1),\ldots,(\boldsymbol{W}^L,\boldsymbol{b}^L)\big).$$

  We will use the following terminology.

  – The number of layers $L(\Phi) = L$ is the depth of Φ;

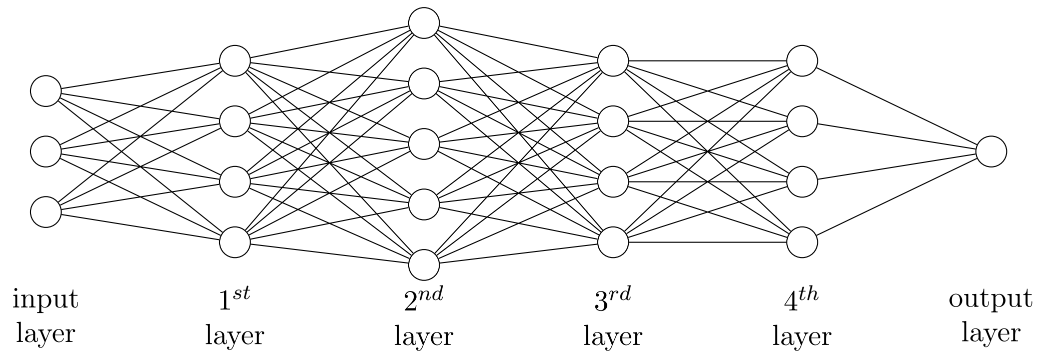

Figure 1 – **The graph associated to a deep neural network with input dimension 3 and 5 layers.**

- $N_w(\Phi) = \max_{\ell=0,\dots,L}\{N_\ell\}$ is the width of Φ; $\boldsymbol{N}(\Phi) = (N_0, N_1, \dots, N_L)$ the dimension of Φ;
- The real numbers $w^\ell_{i,j}$ and b^ℓ_j are edge and node weights of Φ, respectively;
- The number of nonzero weights $w^\ell_{i,j}$ and b^ℓ_j is the size of Φ and denoted by $W(\Phi)$;
- When $L(\Phi) \geq 3$, Φ is called a deep neural network, and otherwise, a shallow neural network.

- A neural network architecture $\mathbb{A}$ with input dimension d and L layers is a neural network
$$\mathbb{A} = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \dots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big),$$
where elements of $\boldsymbol{W}^\ell$ and $\boldsymbol{b}^\ell$, $\ell = 1, \dots, L$, are in $\{0,1\}$.

The above defined deep neural network is sometimes called standard networks to distinguish with networks allowing for connections of neurons in non-neighboring layers. A deep neural network can be visualized in a graph. The graph associated with a deep neural network Φ defined in Definition 1 consists of $L+1$ layers which are numbered from 0 to L. The ℓth layer has N_ℓ nodes which are numbered from 1 to N_ℓ. If $w^\ell_{i,j} \neq 0$, then there is an edge connecting the node j in the layer $\ell - 1$ to the node i in the layer ℓ.

In Figure 1 we illustrate a deep neural network with input dimension 3 and 5 layers.

Definition 2. Given $L \in \mathbb{N}$, $L \geq 2$, and a deep neural network architecture $\mathbb{A} = \big((\overline{\boldsymbol{W}}^1, \overline{\boldsymbol{b}}^1), \dots, (\overline{\boldsymbol{W}}^L, \overline{\boldsymbol{b}}^L)\big)$. We say that a neural network $\Phi = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \dots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big)$ has architecture $\mathbb{A}$ if

- $\boldsymbol{N}(\Phi) = \boldsymbol{N}(\mathbb{A})$
- $\overline{w}^\ell_{i,j} = 0$ implies $w^\ell_{i,j} = 0$, $\overline{b}^\ell_i = 0$ implies $b^\ell_i = 0$ for all $i = 1, \dots, N_\ell$, $j = 1, \dots, N_{\ell-1}$, and $\ell = 1, \dots, L$. Here $\overline{w}^\ell_{i,j}$ are entries of $\overline{\boldsymbol{W}}^\ell$ and $\overline{b}^\ell_i$ are elements of $\overline{\boldsymbol{b}}^\ell$, $\ell = 1, \dots, L$.

For a given deep neural network $\Phi = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \dots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big)$, there exists a unique deep neural network architecture $\mathbb{A} = \big((\overline{\boldsymbol{W}}^1, \overline{\boldsymbol{b}}^1), \dots, (\overline{\boldsymbol{W}}^L, \overline{\boldsymbol{b}}^L)\big)$ such that

- $\boldsymbol{N}(\Phi) = \boldsymbol{N}(\mathbb{A})$
- $\overline{w}^\ell_{i,j} = 0 \iff w^\ell_{i,j} = 0$, $\overline{b}^\ell_i = 0 \iff b^\ell_i = 0$ for all $i = 1, \dots, N_\ell$, $j = 1, \dots, N_{\ell-1}$, and $\ell = 1, \dots, L$.

We call this architecture $\mathbb{A}$ the minimal architecture of Φ (this definition is proper in the sense that any architecture of Φ is also an architecture of $\mathbb{A}$.)

A deep neural network is associated with an activation function which calculates output at each node. The choice of activation function depends on the problem under consideration. In this paper we focus our attention on ReLU activation function defined by $\sigma(t) := \max\{t, 0\}, t \in \mathbb{R}$. We will use the notation $\sigma(\boldsymbol{x}) := (\sigma(x_1), \dots, \sigma(x_d))$ for $\boldsymbol{x} \in \mathbb{R}^d$.

Definition 3. A deep ReLU neural network with input dimension d and L layers is a neural network

$$\Phi = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \ldots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big)$$

in which the following computation scheme is implemented

$$\begin{aligned} &\boldsymbol{z}^0 := \boldsymbol{x} \in \mathbb{R}^d, \\ &\boldsymbol{z}^\ell := \sigma(\boldsymbol{W}^\ell \boldsymbol{z}^{\ell-1} + \boldsymbol{b}^\ell), \quad \ell = 1, \ldots, L-1, \\ &\boldsymbol{z}^L := \boldsymbol{W}^L \boldsymbol{z}^{L-1} + \boldsymbol{b}^L. \end{aligned}$$

We call $\boldsymbol{z}^0$ the input and with an ambiguity denote $\Phi(\boldsymbol{x}) := \boldsymbol{z}^L$ the output of Φ and in some places we identify a deep ReLU neural network with its output.

From the above definition, a deep ReLU neural network Φ is a function mapping from $\mathbb{R}^d$ to $\mathbb{R}^{N_L}$. Several deep ReLU neural networks can be combined to form a larger deep ReLU neural network whose output is a linear combination or composition of outputs of sub-networks. In the following, we introduce parallelization and concatenation, see, e.g., [4] and [19].

Lemma 1 (Parallelization)**.** *Let $N \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ be a bounded set, $\lambda_j \in \mathbb{R}$, $j = 1, \ldots, N$. Let Φ_j, $j = 1, \ldots, N$ be deep ReLU neural networks with input dimension d. Then we can explicitly construct a deep ReLU neural network denoted by Φ so that*

$$\Phi(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \Phi_j(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega.$$

Moreover we have

$$L(\Phi) = \max_{j=1,\ldots,N} \{L(\Phi_j)\} \qquad \text{and} \qquad W(\Phi) \leq 3N \max_{j=1,\ldots,N} W(\Phi_j).$$

The network Φ is called the parallelization network of Φ_j, $j = 1, \ldots, N$.

Lemma 2 (Concatenation)**.** *Let Φ_1 and Φ_2 be two ReLU neural networks such that output layer of Φ_1 has the same dimension as input layer of Φ_2. Then, we can explicitly construct a ReLU neural network Φ such that $\Phi(\boldsymbol{x}) = \Phi_2(\Phi_1(\boldsymbol{x}))$ for $\boldsymbol{x} \in \mathbb{R}^d$. Moreover we have*

$$L(\Phi) = L(\Phi_1) + L(\Phi_2) \qquad \text{and} \qquad W(\Phi) \leq 2W(\Phi_1) + 2W(\Phi_2).$$

The network Φ in this lemma is called the concatenation network of Φ_1 and Φ_2.

3. Tensorized Faber series and interpolation sampling recovery

In this section we recall a decomposition of continuous functions on the unit cube $\mathbb{I}^d$ by tensorized Faber series, interpolation approximation by truncated Faber series.

Let $\varphi(x) = (1 - |x-1|)_+$, $x \in \mathbb{I}$, be the hat function (the piece-wise linear B-spline with knots at $0, 1, 2$), where $x_+ := \max(x, 0)$ for $x \in \mathbb{R}$. For $k \in \mathbb{N}_{-1}$ we define the functions $\varphi_{k,s}$ by

$$\varphi_{k,s}(x) := \varphi(2^{k+1}x - 2s), \quad k \geq 0, \ s \in Z(k) := \{0, 1, \ldots, 2^k - 1\}, \tag{1}$$

and

$$\varphi_{-1,s}(x) := \varphi(x - s + 1), \ s \in Z(-1) := \{0, 1\}. \tag{2}$$

For a univariate function f on $\mathbb{I}$, $k \in \mathbb{N}_{-1}$, and $s \in Z(k)$ we define

$$\lambda_{k,s}(f) \ := -\frac{1}{2}\Delta^2_{2^{-k-1}} f\big(2^{-k}s\big), \ k \geq 0, \quad \lambda_{-1,s}(f) \ := f(s),$$

where

$$\Delta_h^2 f(x) := \ f(x + 2h) - 2f(x+h) + f(x),$$

for all x and $h \geq 0$ such that $x, x+h \in \mathbb{I}$. If $m \in \mathbb{N}_0$ we put

$$R_m(f) := \sum_{k=0}^{m} q_k(f), \qquad q_k(f) := \sum_{s \in Z(k)} \lambda_{k,s}(f) \varphi_{k,s}.$$

For $k \in \mathbb{N}_0$, we define the functions $\varphi^*_{k,s} \in \mathring{C}(\mathbb{I})$ by

$$\varphi^*_{k,s}(x) := \varphi(2^{k+1}x - s + 1), \quad s \in Z_*(k) := \{1, \ldots, 2^{k+1} - 1\},$$

and for $f \in \mathring{C}(\mathbb{I})$ one can check

$$R_m(f) \ = \sum_{s \in Z_*(m)} f(2^{-m-1}s)\varphi^*_{m,s}.$$

Hence $R_m(f) \in \mathring{C}(\mathbb{I})$ interpolates f at the points $2^{-m-1}s$, $s \in Z_*(m)$, that is,

$$R_m(f)(2^{-m-1}s) \ = \ f(2^{-m-1}s), \quad s \in Z_*(m).$$

Put $Z(\boldsymbol{k}) := \times_{j=1}^{d} Z(k_j)$. For $\boldsymbol{k} \in \mathbb{N}^d_{-1}$, $\boldsymbol{s} \in Z(\boldsymbol{k})$, we introduce the tensorized Faber basis by

$$\varphi_{\boldsymbol{k},\boldsymbol{s}}(\boldsymbol{x}) \ := \ \varphi_{k_1,s_1}(x_1) \cdot \ldots \cdot \varphi_{k_d,s_d}(x_d), \quad \boldsymbol{x} \in \mathbb{I}^d. \tag{3}$$

We also define the linear functionals $\lambda_{\boldsymbol{k},\boldsymbol{s}}$ for multivariate function f on $\mathbb{I}^d$, $\boldsymbol{k} \in \mathbb{N}^d_{-1}$, and $\boldsymbol{s} \in Z(\boldsymbol{k})$ by

$$\lambda_{\boldsymbol{k},\boldsymbol{s}}(f) \ := \prod_{j=1}^{d} \lambda_{k_j,s_j}(f),$$

where the univariate functional λ_{k_j,s_j} is applied to the univariate function f by considering f as a function of variable x_j with the other variables held fixed. We have the following lemma.

Lemma 3. *The tensorized Faber system* $\{\varphi_{\boldsymbol{k},\boldsymbol{s}} : \ \boldsymbol{k} \in \mathbb{N}^d_{-1}, \boldsymbol{s} \in Z(\boldsymbol{k})\}$ *is a basis in* $C(\mathbb{I}^d)$. *Moreover, every function* $f \in C(\mathbb{I}^d)$ *can be represented by the Faber series*

$$f \ = \sum_{\boldsymbol{k} \in \mathbb{N}^d_{-1}} q_{\boldsymbol{k}}(f), \qquad q_{\boldsymbol{k}}(f) := \sum_{\boldsymbol{s} \in Z(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f)\varphi_{\boldsymbol{k},\boldsymbol{s}} \tag{4}$$

converging in the norm of $C(\mathbb{I}^d)$.

When $d = 1$, the system (1), (2) and above result goes back to Faber [8]. The decomposition (4) when $d = 2$ and an extension for function spaces with mixed smoothness was obtained independently in [24, Theorem 3.10] and in [2, Section 4]. A generalization for the case $d \geq 2$ and also to B-spline interpolation and quasi-interpolation representation was established in [2, 3].

When $f \in \mathring{U}^{\alpha,d}_{\infty}$, $\lambda_{\boldsymbol{k},\boldsymbol{s}}(f) = 0$ if $k_j = -1$ for some $j \in \{1, \ldots, d\}$, hence we can write

$$f \ = \sum_{\boldsymbol{k} \in \mathbb{N}^d_0} q_{\boldsymbol{k}}(f)$$

with unconditional convergence in $C(\mathbb{I}^d)$, see [24, Theorem 3.13]. In this case it holds the following estimate

$$|\lambda_{\boldsymbol{k},\boldsymbol{s}}(f)| \leq 2^{-\alpha d} 2^{-\alpha|\boldsymbol{k}|_1},$$

for $\boldsymbol{k} \in \mathbb{N}^d_0, \ \boldsymbol{s} \in Z(\boldsymbol{k})$.

For $f \in \mathring{C}(\mathbb{I}^d)$, we define the operator R_m by

$$R_m(f) := \sum_{|\boldsymbol{k}|_1 \leq m} q_{\boldsymbol{k}}(f) \ = \sum_{|\boldsymbol{k}|_1 \leq m} \ \sum_{\boldsymbol{s} \in Z(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f)\varphi_{\boldsymbol{k},\boldsymbol{s}}. \tag{5}$$

The truncated Faber series $R_m(f) \in \mathring{C}(\mathbb{I}^d)$ completely determined by values of f at the points $2^{-\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{s}$, for $(\boldsymbol{k}, \boldsymbol{s}) \in G^d(m)$, where

$$G^d(m) := \{(\boldsymbol{k}, \boldsymbol{s}) : \ |\boldsymbol{k}|_1 \leq m, \ \boldsymbol{s} \in Z_*(\boldsymbol{k})\},$$

$Z_*(\boldsymbol{k}) := \prod_{j=1}^{d} Z_*(k_j)$ and $\boldsymbol{1} = (1, \ldots, 1) \in \mathbb{N}^d$. Moreover, $R_m(f)$ interpolates f at the points $2^{-\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{s}$, for $(\boldsymbol{k}, \boldsymbol{s}) \in G^d(m)$, i.e.,

$$R_m(f)(2^{-\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{s}) \ = \ f(2^{-\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{s}), \quad (\boldsymbol{k}, \boldsymbol{s}) \in G^d(m).$$

The following lemma gives a d-dependent estimate of the approximation error by $R_m(f)$ of $f \in \mathring{U}^{\alpha,d}_\infty$, see [5].

Lemma 4. *Let $d, m \in \mathbb{N}$, and $0 < \alpha \le 1$. Then we have*

$$\sup_{f \in \mathring{U}^{\alpha,d}_\infty} \|f - R_m(f)\|_{L_\infty(\mathbb{I}^d)} \le 2^{-\alpha} B^d \, 2^{-\alpha m} \binom{m+d}{d-1}, \qquad B = (2^\alpha - 1)^{-1}.$$

For univariate functions $f \in \mathring{C}(\mathbb{I})$, let the operator T_k, $k \in \mathbb{N}_0$, be defined by

$$T_k(f) := f - R_{k-1}(f)$$

with the operator R_k defined as in (5) and the convention $R_{-1} := 0$. From this definition we have T_0 is the identity operator. Notice that for $f \in \mathring{U}^{\alpha,1}_\infty$, it holds the inequality $\|T_k(f)\|_{H^\alpha_\infty(\mathbb{I})} \le 2$.

For a multivariate function $f \in \mathring{C}(\mathbb{I}^d)$, the tensor product operator $T_{\boldsymbol{k}}$, $\boldsymbol{k} = (k_1, \ldots, k_d) \in \mathbb{N}^d_0$, is defined by

$$T_{\boldsymbol{k}}(f) := \prod_{j=1}^{d} T_{k_j}(f),$$

where the univariate operator T_{k_j} is applied to the univariate function f by considering f as a function of variable x_j with the other variables held fixed. The following lemma was proved in [5].

Lemma 5. *Let $n, d \in \mathbb{N}$, $\alpha \in (0,1]$, and $f \in \mathring{U}^{\alpha,d}_\infty$. Then $f - R_n(f)$ can be represented in the following special form*

$$f - R_n(f) = \sum_{j=0}^{d-1} \sum_{|\boldsymbol{k}_j|_1 \le n} F_{\boldsymbol{k}_j}, \tag{6}$$

where $F_{\boldsymbol{k}_0} := T_{(n+1)\boldsymbol{e}^1}$ and

$$F_{\boldsymbol{k}_j} := T_{(n+1-|\boldsymbol{k}_j|_1)\boldsymbol{e}^{j+1}}\big(q_{\boldsymbol{k}_j}(f)\big), \quad j = 1, \ldots, d-1.$$

Here $\boldsymbol{k}_j = (k_1, \ldots, k_j, 0, \ldots, 0) \in \mathbb{N}^d_0$ and $\{\boldsymbol{e}^j\}_{j=1,\ldots,d}$ is the standard basis of $\mathbb{R}^d$.

4. Deep ReLU network approximations

In this section, we construct nonadaptive and adaptive methods of approximation by deep ReLU neural networks of functions $f \in \mathring{U}^{\alpha,d}_\infty$. Since the case $d = 1$ was already studied in [1, 6, 10] for nonadaptive method and in [6, 27] for adaptive method, in this paper we focus our attention on the case $d \ge 2$. We first recall a result of approximating tensorized Faber functions $\varphi_{\boldsymbol{k},\boldsymbol{s}}$ by deep ReLU neural networks, see [4].

Lemma 6. *For every dimension $d \ge 2$, $\delta \in (0,1)$ and for the d-variate hat functions $\varphi_{\boldsymbol{k},\boldsymbol{s}}$, $\boldsymbol{k} \in \mathbb{N}^d_0$, $\boldsymbol{s} \in Z(\boldsymbol{k})$, defined as in (3), we can explicitly construct a deep neural network $\Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})$ so that*

$$\big\|\varphi_{\boldsymbol{k},\boldsymbol{s}} - \Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})\big\|_{L_\infty(\mathbb{I}^d)} \le \delta$$

and

$$W(\Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})) \le C d \log(d\delta^{-1}) \qquad \text{and} \qquad L(\Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})) \le C \log d \log(d\delta^{-1}).$$

Moreover, $\operatorname{supp} \Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}}) \subset \operatorname{supp} \varphi_{\boldsymbol{k},\boldsymbol{s}}$.

The above result allows us to construct a deep ReLU network $\Phi_\varepsilon(R_n(f))$ to approximate $R_n(f)$. The network $\Phi_\varepsilon(R_n(f))$ is constructed by parallelization of the networks $\{\Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})\}_{|\boldsymbol{k}|_1 \le n, \, \boldsymbol{s} \in Z(\boldsymbol{k})}$ and has output

$$\Phi_\varepsilon\big(R_n(f)\big)(\boldsymbol{x}) = \sum_{|\boldsymbol{k}|_1 \le n} \sum_{\boldsymbol{s} \in Z(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f) \Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})(\boldsymbol{x}).$$

Lemma 6 and 1 allow us to control number of weights and length of $\Phi_\varepsilon(R_n(f))$. More precisely we have the following.

Lemma 7. *Let $d \in \mathbb{N}$, $d \geq 2$, $n \in \mathbb{N}$, $\alpha \in (0,1]$ and $\varepsilon \in (0,1)$. Then for every $f \in \mathring{U}_\infty^{\alpha,d}$ we can explicitly construct a deep ReLU network $\Phi_\varepsilon(R_n(f))$ of the same architecture $\mathbb{A}_\varepsilon$ so that*

$$\big\|R_n(f) - \Phi_\varepsilon\big(R_n(f)\big)\big\|_{L_\infty(\mathbb{I}^d)} \leq \varepsilon.$$

Moreover, we have

$$W\big(\Phi_\varepsilon(R_n(f))\big) \leq Cd2^n\binom{n+d-1}{d-1}\log(dB^d\varepsilon^{-1}) \tag{7}$$

and

$$L\big(\Phi_\varepsilon(R_n(f))\big) \leq C\log d\log(dB^d\varepsilon^{-1}). \tag{8}$$

The estimates (7) and (8) also hold for $W(\mathbb{A}_\varepsilon)$ and $L(\mathbb{A}_\varepsilon)$ respectively.

We are now in the position to formulate our nonadaptive result. Nonadaptivity means that architecture of approximating deep ReLU neural networks is the same for all $f \in \mathring{U}_\infty^{\alpha,d}$.

Theorem 1. *Let $d \in \mathbb{N}$, $d \geq 2$, and $\alpha \in (0,1]$. Then there is $\varepsilon_0 = \varepsilon_0(d,\alpha) \in (0,1]$ such that for every $\varepsilon \in (0,\varepsilon_0)$ we can explicitly construct a deep neural network architecture $\mathbb{A}_\varepsilon$ with the following property. For every $f \in \mathring{U}_\infty^{\alpha,d}$, we can explicitly construct a deep ReLU neural network $\Phi_\varepsilon(f)$ having the architecture $\mathbb{A}_\varepsilon$ such that*

$$\|f - \Phi_\varepsilon(f)\|_{L_\infty(\mathbb{I}^d)} \leq \varepsilon.$$

Moreover, we have

$$W(\mathbb{A}_\varepsilon) \leq C_\alpha d\left(\frac{K_1^d}{(d-1)!}\right)^{\frac{1}{\alpha}+1}\varepsilon^{-\frac{1}{\alpha}}\log(2\varepsilon^{-1})^{(d-1)(\frac{1}{\alpha}+1)+1} \tag{9}$$

and

$$L(\mathbb{A}_\varepsilon) \leq C\log d\log(2\varepsilon^{-1}),$$

where $K_1 = B^{1/(\alpha+1)}4\alpha^{-1}$ with B given in Lemma 4 and C_α depends only on α.

Let us explain the idea of proving this theorem. Our technique is first to approximate f by its truncation of Faber series $R_n(f)$ and then approximate $R_n(f)$ by a deep ReLU network. With $\varepsilon' = \varepsilon/2$ in Lemma 7 and $\Phi_\varepsilon(f) = \Phi_{\varepsilon'}(R_n(f))$ we have

$$\begin{aligned}\|f - \Phi_\varepsilon(f)\|_{L_\infty(\mathbb{I}^d)} &\leq \|f - R_n(f)\|_{L_\infty(\mathbb{I}^d)} + \|R_n - \Phi_{\varepsilon'}(R_n(f))\|_{L_\infty(\mathbb{I}^d)} \\ &\leq 2^{-\alpha}B^d2^{-\alpha n}\binom{n+d}{d-1} + \frac{\varepsilon}{2}.\end{aligned}$$

Then we choose n such that $2^{-\alpha}B^d2^{-\alpha n}\binom{n+d}{d-1} \leq \frac{\varepsilon}{2}$. With this choice of n, Lemma 7 gives estimates for number of weighs and length of $\Phi_\varepsilon(f)$.

To complete the proof, we notice that $\Phi_\varepsilon(f)$ has the architecture $\mathbb{A}_\varepsilon$ which is defined as the minimal architecture of the deep ReLU neural network Φ_ε obtained by parallelization of the networks $\{\Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})\}_{|\boldsymbol{k}|_1\leq n,\, \boldsymbol{s}\in Z(\boldsymbol{k})}$ with the output

$$\Phi_\varepsilon(\boldsymbol{x}) = \sum_{|\boldsymbol{k}|_1\leq n}\sum_{\boldsymbol{s}\in Z(\boldsymbol{k})}\Phi_\delta(\varphi_{\boldsymbol{k},\boldsymbol{s}})(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{I}^d.$$

The advantage of the above method is that the deep ReLU neural networks are easily constructed and they have the same architecture for all functions in $\mathring{U}_\infty^{\alpha,d}$. Since it uses $R_n(f)$ as a mediate approximation, a disadvantage is that with the same accuracy the computation complexity of deep ReLU networks is not better than that when approximating functions in $\mathring{U}_\infty^{\alpha,d}$ by linear methods. To overcome this disadvantage we develop a technique used in [27] and [6] for the univariate case. This method reduces the computation complexity of the approximating deep ReLU networks comparing with that of the nonadaptive method given in Theorem 1. Our results on adaptive methods are read as follows.

Theorem 2. *Let $d \in \mathbb{N}$, $d \geq 2$, $\alpha \in (0,1]$. Then there is $\varepsilon_0 = \varepsilon_0(d,\alpha) \in (0,1/2]$ such that for every $\varepsilon \in (0,\varepsilon_0)$ and for every $f \in \mathring{U}^{\alpha,d}_\infty$ we can explicitly construct an adaptive deep ReLU neural network $\Phi_\varepsilon(f)$ so that*

$$\|f - \Phi_\varepsilon(f)\|_{L_\infty(\mathbb{I}^d)} \leq \varepsilon.$$

Moreover, we have

$$W(\Phi_\varepsilon(f)) \leq C_\alpha d^2 \left(\frac{K_2^d}{(d-1)!}\right)^{\frac{2}{\alpha}+2} \varepsilon^{-\frac{1}{\alpha}} \left(\log(2\varepsilon^{-1}) \log\log(2\varepsilon^{-1})\right)^{(1+\frac{1}{\alpha})(d-1)} \tag{10}$$

and

$$L(\Phi_\varepsilon(f)) \leq C'_\alpha \varepsilon^{-\frac{1}{d\alpha}} (\log(2\varepsilon^{-1}))^{\frac{d-1-\alpha}{d\alpha}} (\log\log(2\varepsilon^{-1}))^{\frac{(\alpha+1)(d-1)}{d\alpha}},$$

where

$$K_2 := 4(2^{\alpha+3}B)^{\frac{1}{2\alpha+2}} (\alpha^{-1}\log(2\alpha^{-1}))^{1/2}$$

and positive constants C_α, C'_α depend on α only.

Comparing (9) and (10) we find the later estimation improves $\log(2\varepsilon^{-1})$. Notice that terms in (9) and (10) which depend on dimension d only decay as fast as super exponential in d.

We sketch a plan of proof of this theorem. Let $f \in \mathring{U}^{\alpha,d}_\infty$ and $\varepsilon \in (0,\varepsilon_0)$ (with some $\varepsilon_0 \in (0,1)$) be given. Using the writing

$$f \;=\; R_n(f) \;+\; (f - R_n(f)),$$

we explicitly construct deep ReLU neural networks $\Phi_{\varepsilon/2}(R_n(f))$ and $\Phi_{\varepsilon/2}(f - R_n(f))$ to approximate the terms $R_n(f)$ and $f - R_n(f)$ with accuracy $\varepsilon/2$. We then construct a deep ReLU neural network $\Phi_\varepsilon(f)$ as a parallelization of $\Phi_{\varepsilon/2}(R_n(f))$ and $\Phi_{\varepsilon/2}(f - R_n(f))$ with an output

$$\Phi_\varepsilon(f)(\boldsymbol{x}) \;=\; \Phi_{\varepsilon/2}\big(R_n(f)\big)(\boldsymbol{x}) \;+\; \Phi_{\varepsilon/2}\big(f - R_n(f)\big)(\boldsymbol{x}).$$

Then we have

$$\|f - \Phi_\varepsilon(f)\|_{L_\infty(\mathbb{I}^d)} \leq \varepsilon.$$

For construction of the network $\Phi_{\varepsilon/2}(R_n(f))$ we use Lemma 7. To construct a desired deep ReLU neural network $\Phi_{\varepsilon/2}(f - R_n(f))$ our strategy is to employ the special representation in Lemma 5. Using parallelization and concatenation of deep ReLU neural networks in Lemma 1 and 2 we explicitly construct deep ReLU neural networks $\Phi_{\varepsilon'}(F_{\boldsymbol{k}_j})$ to approximate each term $F_{\boldsymbol{k}_j}$ with accuracy ε' in the sum in (6). The network $\Phi_{\varepsilon/2}(f - R_n(f))$ is defined as a parallelization of $\Phi_{\varepsilon'}(F_{\boldsymbol{k}_j})$, $|\boldsymbol{k}_j|_1 \leq n$, $j = 0,\ldots,d-1$ with the output

$$\Phi_{\varepsilon/2}\big(f - R_n(f)\big)(\boldsymbol{x}) := \sum_{j=0}^{d-1} \sum_{|\boldsymbol{k}_j|_1 \leq n} \Phi_{\varepsilon'}\big(F_{\boldsymbol{k}_j}\big)(\boldsymbol{x}).$$

In the next step we chose ε' depending on ε such that

$$\big\|\big(f - R_n(f)\big) - \Phi_{\varepsilon/2}\big(f - R_n(f)\big)\big\|_{L_\infty(\mathbb{I}^d)} \leq \varepsilon/2.$$

Finally, the size and depth of $\Phi_\varepsilon(f)$ are estimated explicitly in d and ε from the estimation of sizes and depths of $\Phi_{\varepsilon/2}(R_n(f))$ and $\Phi_{\varepsilon'}(F_{\boldsymbol{k}_j})$.

5. An application to numerical solving PDEs

In this section, we apply the results on approximation by deep ReLU neural networks in Section 4 for numerical approximation of the solution to elliptic PDEs.

Consider a modeled diffusion elliptic equation

$$-\mathrm{div}(a(\boldsymbol{x})\nabla u(\boldsymbol{x})) \;=\; f(\boldsymbol{x}) \quad \text{in} \quad \mathbb{I}^d, \quad u|_{\partial\mathbb{I}^d} \;=\; 0,$$

with a function f and a diffusion coefficient a having sufficient regularity. Denote by $V := H^1_0(\mathbb{I}^d) = \mathring{W}^1_2(\mathbb{I}^d)$ the energy space. If a satisfies the ellipticity assumption

$$0 < a_{\min} \leq a(\boldsymbol{x}) \leq a_{\max} < \infty, \quad \boldsymbol{x} \in \mathbb{I}^d,$$

by the well-known Lax-Milgram lemma, there exists a unique solution $u \in V$ which satisfies the variational equation

$$\int_{\mathbb{I}^d} a(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \;=\; \int_{\mathbb{I}^d} f(\boldsymbol{x}) v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \quad \forall v \in V.$$

We want to approximate the solution u by deep ReLU neural networks. The approximation error is measured in the norm of $L_\infty(\mathbb{I}^d)$. Assume for the modeled case that a and f have Hölder-Nikol'skii mixed smoothness 1, i.e., $a, f \in H^1_\infty(\mathbb{I}^d)$. Then, the solution u has at least mixed derivatives $\partial^{\boldsymbol{\alpha}} u$ with $\boldsymbol{\alpha} \in \mathbb{N}_0^d$, $\max_{j=1,\dots,d} \alpha_j \le 1$, belonging to $L_2(\mathbb{I}^d)$ [11], and therefore, by embedding for function spaces of mixed smoothness, see [21, Theorem 2.4.1], u belongs to $\mathring{H}^{1/2}_\infty(\mathbb{I}^d)$. For simplicity we assume that $u \in \mathring{U}^{1/2}_\infty$.

For the nonadaptive approximation, according to Theorem 1, for any $\varepsilon > 0$ sufficient small one can explicitly construct a deep neural network architecture $\mathbb{A}_\varepsilon$ independent of f and a, and a deep ReLU neural network $\Phi_\varepsilon(u)$ having the architecture $\mathbb{A}_\varepsilon$ such that

$$\|u - \Phi_\varepsilon(u)\|_{L_\infty(\mathbb{I}^d)} \le \varepsilon,$$

$$W(\mathbb{A}_\varepsilon) \le Cd \left(\frac{K_1^d}{(d-1)!} \right)^3 \varepsilon^{-2} \log(2\varepsilon^{-1})^{3(d-1)+1},$$

and

$$L(\mathbb{A}_\varepsilon) \le C \log d \log(2\varepsilon^{-1}),$$

where $K_1 := 8(\sqrt{2}+1)^{3/2}$.

For the adaptive approximation, according to Theorem 2, for any $\varepsilon > 0$ sufficient small one can explicitly construct an adaptive deep ReLU neural network $\Phi_\varepsilon(u)$ so that

$$\|u - \Phi_\varepsilon(u)\|_{L_\infty(\mathbb{I}^d)} \le \varepsilon,$$

$$W(\Phi_\varepsilon(u)) \le Cd^2 \left(\frac{K_2^d}{(d-1)!} \right)^6 \varepsilon^{-2} \big(\log(2\varepsilon^{-1}) \log\log(2\varepsilon^{-1}) \big)^{3(d-1)},$$

and

$$L(\Phi_\varepsilon(u)) \le C' \varepsilon^{-\frac{2}{d}} (\log(2\varepsilon^{-1}))^{\frac{2d-3}{d}} (\log\log(2\varepsilon^{-1}))^{\frac{3(d-1)}{d}},$$

where $K_2 := 16((2+\sqrt{2}))^{1/3}$.

Acknowledgments. This work is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant No. 102.01-2020.03. A part of this work was done when the authors were working at the Vietnam Institute for Advanced Study in Mathematics (VIASM). They would like to thank the VIASM for providing a fruitful research environment and working condition.

References

1 Ali M., Nouy A. Approximation of smoothness classes by deep ReLU networks // arXiv:2007.15645. –2020.
2 Dũng D. B-spline quasi-interpolant representations and sampling recovery of functions with mixed smoothness // J. Complexity. –2011. – Vol.27, №541. –P. 567.
3 Dũng D. Sampling and cubature on sparse grids based on a B-spline quasi-interpolation // Found. Comp. Math. –2016. –Vol. 16. –P. 1193-1240.
4 Dũng D., Nguyen V. K. Sparse-grid sampling recovery and deep ReLU neural networks in high-dimensional approximation // arxiv.org/abs/2007.08729. –2020.
5 Dũng D., Nguyen V. K. High-dimensional nonlinear approximation by parametric manifolds in Hölder-Nikol'skii spaces of mixed smoothness // arxiv.org/abs/2102.04370. –2021.
6 Daubechies I., DeVore R., Foucart S., Hanin B., Petrova G. Nonlinear approximation and (Deep) ReLU networks // arXiv:1905.02199. –2019.
7 E W., Wang Q. Exponential convergence of the deep neural network approximation for analytic functions // Sci. China Math. –2018. – Vol. 61. –P. 1733-1740.
8 Faber G. Über stetige Funktionen // Math. Ann. –1909. – Vol. 66. –P. 81-94.
9 Geist M., Petersen P., Raslan M., Schneider R., Kutyniok G. Numerical solution of the parametric diffusion equation by deep neural networks // Preprint. –2020.
10 Gribonval R., Kutyniok G., Nielsen M., Voigtlaender F. Approximation spaces of deep neural networks // arXiv:1905.01208. –2019.

11 Griebel M., Knapek S. Optimized general sparse grid approximation spaces for operator equations // Math. Comp. –2009. –Vol. 78. –P. 2223-2257.
12 Grohs P., Perekrestenko D., Elbrachter D., Bolcskei H. Deep neural network approximation theory // arXiv: 1901.02220. –2019.
13 Gühring I., Kutyniok G., Petersen P. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms // Anal. Appl. (Singap.). –2020 –Vol. 18. –P. 803-859.
14 Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. NeurIPS. –2012. –P. 1106-1114.
15 LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. –2015. –Vol. 521. –P. 436-444.
16 Mhaskar H. N. Neural networks for optimal approximation of smooth and analytic functions // Neural Comput. –1996. –Vol. 8. –P. 164-177.
17 Montanelli H., Du Q. New error bounds for deep ReLU networks using sparse grids // SIAM J. Math. Data Sci. –2019. –Vol. 1. –P. 78-92.
18 Opschoor J. A. A., Petersen P. C., Schwab C. Deep ReLU networks and high-order finite element methods // Anal. Appl. (Singap.). –2020. –Vol. 18. –P. 715-770.
19 Petersen P., Voigtlaender F. Optimal approximation of piecewise smooth functions using deep ReLU neural networks // Neural Netw. –2018. –Vol. 108. –P. 296-330.
20 Petersen P. C. Neural network theory. Preprint. –2020.
21 Schmeisser H. Triebel H. Topics in Fourier Analysis and Function Spaces. Chichester, New York, Wiley. –1987.
22 Schwab C. Zech J. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ // Anal. Appl. (Singap.). –2019. –Vol. 17. –P. 19-55.
23 Suzuki T. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality // International Conference on Learning Representations. –2019.
24 Triebel H. Bases in Function Spaces, Sampling, Discrepancy, Numerical Integration. European Math. Soc. Publishing House, Zürich. –2010.
25 Wu Y., Schuster M., Chen Z., Le Q. V., Norouzi M. Google's neural machine translation system: Bridging the gap between human and machine translation // arXiv: 1609.08144. –2016.
26 Yarotsky D. Error bounds for approximations with deep ReLU networks // Neural Netw. –2017. –Vol. 94. –P. 103-114.
27 Yarotsky D. Quantifed advantage of discontinuous weight selection in approximations with deep neural networks // arXiv: 1705.01365. –2017.

Д. Зунг [1] , В. К. Нгуен [2] , М. С. Тао [3]

[1] *Вьетнам ұлттық университеті, Ақпараттық технологиялар институты, Ханой, Вьетнам*
[2] *Көлік және коммуникация университеті, Іргелі ғылымдар факультеті, Ханой, Вьетнам*
[3] *Хонг Дук университеті, Жаратылыстану ғылымдары факультеті, Тханьхоа, Вьетнам*

ReLU терең нейрондық желілері бойынша көп өлшемді жуықтаудың есептеу күрделілігі туралы

Аннотация: Мақалада $\mathbb{I}^d := [0,1]^d$ бірлік кубында аралас тегістікті $H^\alpha_\infty(\mathbb{I}^d)$ Гольдер-Никольский кеңістігіндегі функцияларды жуықтау үшін ReLU терең нейрондық желілерінің есептеу күрделілігі зерттелген. Кез келген $f \in H^\alpha_\infty(\mathbb{I}^d)$ функциясы үшін f-ті берілген ε дәлдікпен жуықтайтын шығыс сигналы бар адаптивті емес және адаптивті ReLU терең нейрондық желілері құрылды және осы ReLU терең нейрондық желісінің өлшемі мен тереңдігі арқылы сипатталатын жуықтаудың есептеу күрделілігінің бағалаулары d және ε шамаларынан тәуелді болатындығы дәлелденеді. Зерттеу нәтижелері ReLU терең нейрондық желілерімен жуықтаудың адаптивті әдісі адаптивті емес әдіске қарағанда артықшылығын көрсетті.

Түйін сөздер: ReLU терең нейрондық желісі; есептеудің күрделілігі; көп өлшемді жуықтау; аралас тегістікті Гёльдер-Никольский кеңістігі.

Д. Зунг [1] , В. К. Нгуен [2] , М. С. Тао [3]

[1] *Институт информационных технологий, Вьетнамский национальный университет, Ханой, Вьетнам*
[2] *Факультет фундаментальных наук Университета транспорта и коммуникаций Ханой, Вьетнам*
[3] *Факультет естественных наук Университета Хонг Дык, Тханьхоа, Вьетнам*

О сложности вычислений многомерной аппроксимации глубокими нейронными сетями ReLU

Аннотация: В статье исследуется вычислительная сложность глубоких нейронных сетей ReLU для аппроксимации функций в пространствах Гёльдера-Никольского со смешанной гладкостью $H^\alpha_\infty(\mathbb{I}^d)$ на единичном кубе $\mathbb{I}^d := [0,1]^d$. Для любой функции $f \in H^\alpha_\infty(\mathbb{I}^d)$, строятся неадаптивные и адаптивные глубокие нейронные сети ReLU, имеющие выходные сигналы, приближающие f с заданной точностью ε, и доказывается, что оценки вычислительной сложности приближения, характеризующиеся размером и глубиной этой глубокой нейронной сети ReLU, зависят от d и ε. Результаты показывают преимущество адаптивного метода аппроксимации глубокими нейронными сетями ReLU над неадаптивным методом.

Ключевые слова: Глубокая нейронная сеть ReLU; вычислительная сложность; многомерное приближение; Пространство Гёльдера-Никольского смешанной гладкости.

References

1 M. Ali and A. Nouy. Approximation of smoothness classes by deep ReLU networks. arXiv:2007.15645 (2020).
2 D. Dũng. B-spline quasi-interpolant representations and sampling recovery of functions with mixed smoothness. J. Complexity, 27(541), 567 (2011).
3 D. Dũng. Sampling and cubature on sparse grids based on a B-spline quasi-interpolation. Found. Comp. Math., 16, 1193-1240 (2016).
4 D. Dũng and V. K. Nguyen. Sparse-grid sampling recovery and deep ReLU neural networks in high-dimensional approximation. arxiv.org/abs/2007.08729 (2020).
5 D. Dũng and V. K. Nguyen. High-dimensional nonlinear approximation by parametric manifolds in Hölder-Nikol'skii spaces of mixed smoothness. arxiv.org/abs/2102.04370 (2021).
6 I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (Deep) ReLU networks. arXiv:1905.02199 (2019).
7 W. E and Q. Wang. Exponential convergence of the deep neural network approximation for analytic functions. Sci. China Math., 61, 1733-1740 (2018).
8 G. Faber. Über stetige Funktionen. Math. Ann., 66, 81-94 (1909).
9 M. Geist, P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok. Numerical solution of the parametric diffusion equation by deep neural networks. Preprint (2020).
10 R. Gribonval, Kutyniok, M. Nielsen, and F. Voigtlaender. Approximation spaces of deep neural networks. arXiv:1905.01208 (2019).
11 M. Griebel and S. Knapek. Optimized general sparse grid approximation spaces for operator equations. Math. Comp., 78, 2223-2257 (2009).
12 P. Grohs, D. Perekrestenko, D. Elbrachter, and H. Bolcskei. Deep neural network approximation theory. arXiv: 1901.02220 (2019).
13 I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. Anal. Appl. (Singap.), 18, 803-859 (2020).
14 A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. NeurIPS, 1106-1114 (2012).
15 Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521, 436-444 (2015).
16 H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. Neural Comput., 8, 164-177 (1996).
17 H. Montanelli and Q. Du. New error bounds for deep ReLU networks using sparse grids. SIAM J. Math. Data Sci., 1, 78-92 (2019).
18 J. A. A. Opschoor, P. C. Petersen, and C. Schwab. Deep ReLU networks and high-order finite element methods. Anal. Appl. (Singap.), 18, 715-770 (2020).
19 P. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. Neural Netw., 108, 296-330 (2018).
20 P. C. Petersen. Neural network theory. Preprint (2020).
21 H. Schmeisser and H. Triebel. Topics in Fourier Analysis and Function Spaces (Chichester, New York, Wiley, 1987).
22 C. Schwab and J. Zech. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ. Anal. Appl. (Singap.), 17, 19-55 (2019).
23 T. Suzuki. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. International Conference on Learning Representations (2019).
24 H. Triebel. Bases in Function Spaces, Sampling, Discrepancy, Numerical Integration. European Math. Soc. Publishing House, Zürich (2010).
25 Y. Wu, M. Schuster, Z. Chen, Q. V. Le, and M. Norouzi. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv: 1609.08144 (2016).
26 D. Yarotsky. Error bounds for approximations with deep ReLU networks. Neural Netw., 94, 103-114 (2017).
27 D. Yarotsky. Quantifed advantage of discontinuous weight selection in approximations with deep neural networks. arXiv: 1705.01365 (2017).

Authors Information:

Dinh Dũng – Dr.Sc., Professor, Vietnam National University, Information Technology Institute, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam.

Van Kien Nguyen – corresponding author, PhD, Faculty of Basic Sciences, University of Transport and Communications, No.3 Cau Giay Street, Lang Thuong Ward, Dong Da District, Hanoi, Vietnam.

Mai Xuan Thao – PhD, Department of Natural Sciences, Hong Duc University, 565 Quang Trung, Thanh Hoa, Vietnam.

Динь Зунг –Ғылым докторы, профессор, Вьетнам ұлттық университеті, Ақпараттық технологиялар институты, Сюан Тхуй, 144, Кау Джай, Ханой, Вьетнам.

Ван Киен Нгуен – корреспонденция үшін автор, PhD, Көлік және коммуникация университеті, Іргелі ғылымдар факультеті, Кау Гиай, 3, Ланг Тхьюнг, район Донг Да, Ханой, Вьетнам.

Май Суан Тао – PhD, Хонг Дук университеті, Жаратылыстану ғылымдары факультеті, Куанг Трунг, 565, Тханьхоа, Вьетнам.

Received 11.12.2020