Neural Computing and Applications (2019) 31:4613–4628
https://doi.org/10.1007/s00521-018-3573-3

S.I. : EMERGENCE IN HUMAN-LIKE INTELLIGENCE TOWARDS CYBER-PHYSICAL SYSTEMS

Constructive function approximation by neural networks with optimized activation functions and fixed weights

Feng-Jun Li[1]

Received: 28 February 2018 / Accepted: 30 May 2018 / Published online: 9 June 2018

Abstract
Our purpose in this paper is to construct three types of single-hidden layer feed-forward neural networks (FNNs) with optimized piecewise linear activation functions and fixed weights and to present the ideal upper and lower bound estimations on the approximation accuracy of the FNNs, for continuous function defined on bounded intervals. We also prove these three types of single-hidden layer FNNs can interpolate any bounded and measurable functions. Our approach compared with existing methods does not require training. Our conclusions not only uncover the inherent properties of approximation of the FNNs, but also reveal the latent relationship among the precision of approximation, the number of hidden units and the smoothness of the target function. Finally, we demonstrate some numerical results that show good agreement with theory.

Keywords FNN · Approximation · Upper bound · Lower bound

1 Introduction

Since it is wide and latent learning and expression capabilities, neural networks have been widely used in the real world. It includes almost all fields of natural science and part of social science [1–6]. It is well known that the most widely used neural networks is the feed-forward neural networks (FNNs). Many practical problems related to FNNs application, such as in pattern recognition, information processing, engineering technology, computer science, and systems control, can be converted into the ones of learning (or approximating) multivariate functions by the FNNs with optimized activation functions, for which an extensive study on approximation by FNNs has been carried out in a huge topic [7–14].

In recent years, interpolation (approximation with zero error, namely, exact approximation) by FNNs has been a hot spot of research in theory and application of FNNs and its generalization, attracting the attention of scholars all over the world [15–22].

The most widely used and studied neural networks are maybe the FNNs with one hidden layer. The fundamental element of a neural network is known as a "neuron" or a "unit." Neurons are arranged in layers. A FNN with one hidden layer consists of three layers: input layer, hidden layer and output layer. A sketch map of a FNN is exhibited in Fig. 1.

A three-layer FNN with d input units, m hidden units and one output unit is mathematically represented as the following form

$$N(\mathbf{x}) := \sum_{i=1}^{m} c_i \sigma\left(\sum_{j=1}^{d} w_{ij} x_j + \theta_i\right), \mathbf{x} = (x_1, x_2, \ldots, x_d) \in R^d, \quad d \geq 1, \tag{1.1}$$

where $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{id})^T \in R^d$ are connection weights of the unit i in the hidden layer with the input units, $c_i \in R$ are the connection strengths of unit i with the output unit, $\theta_i \in R$ are the thresholds and σ is the activation function. The activation function be usually considered as sigmoid style, namely, it satisfies $\lim_{x \to +\infty} \sigma(x) = 1$ and

✉ Feng-Jun Li
fjli@nxu.edu.cn

1 School of Mathematics and Statistics, Ningxia University, Yinchuan 750021, China

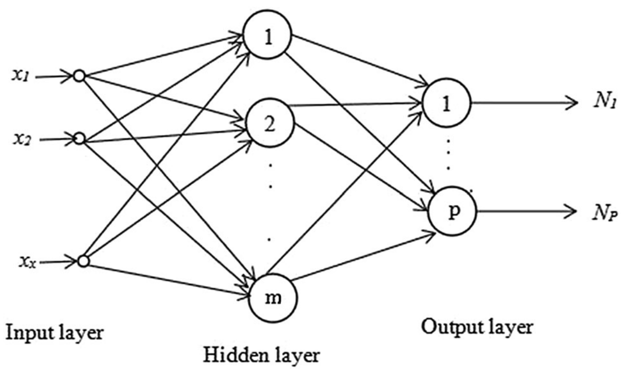


Fig. 1 The architecture of a single hidden layer FNN with n input neurons, m hidden neurons and p output neurons

$\lim_{x\to-\infty}\sigma(x)=0$. Equation (1.1) can be further shown in vector pattern as

$$N(\mathbf{x}) := \sum_{i=1}^{m} c_i\sigma(\mathbf{w}_i\cdot\mathbf{x}+\theta_i), \quad \mathbf{x}\in R^d, \quad d\geq 1.$$

We will study the following set of functions

$$\mathbb{N}_{n+1}^{d}(\phi) := \left\{N(\mathbf{x})=\sum_{j=0}^{n} c_j\phi(\mathbf{w}_j\cdot\mathbf{x}+b_j), \mathbf{w}_j\in R^d, c_j, b_j\in R\right\},$$

where $\mathbf{w}_j\cdot\mathbf{x}$ expresses the ordinary dot product of R^d and ϕ is a function of R to itself. We call three-layer FNNs with one hidden layer to be the elements of $\mathbb{N}_{n+1}^{d}(\phi)$, and also call unit to each summand of $N(\mathbf{x})$.

Function approximation by FNNs (1.1) has been extensively studied in the past years with a variety of important results involving density or complexity (see, e.g., [1–14]). The density problem is to determine the requirements under which any function can be approximated by a three-layer FNN with arbitrary precision. The complexity problem is to ascertain the relationship between the smoothness of the approximated function and the lost necessary to attain an approximation with a desired accuracy, which is nearly equivalent to the problem of the metric of approximation [10]. In essence, the problem of density is qualitative research, while the complexity is a quantitative study. Up to now, all kinds of density and complexity outcomes on approximation of the functions by the FNNs (1.1) in the set $\mathbb{N}_{n+1}^{d}(\phi)$ were given by using different approaches for more or less general situations (for instance [10, 23–30] and references therein). However, in previous papers [1–30, 30–41], the weights and thresholds in FNNs vary such that the results are very difficult to be applied in reality.

Let $S=\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_n\}\subset R^d$ be a set of mutually different vectors, $\{y_i, i=0,1,\ldots,n\}$ be a set of real numbers and $(\mathbf{x}_0,y_0),(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)$, be a group of ordered pairs. We know that the FNNs (1) $N: R^d\to R$ is an interpolation of these ordered pairs if $N(\mathbf{x}_i)=y_i, i=0,1,\ldots,n$.

As we know, three-layer FNNs with at most $n+1$ summands (components of $\mathbb{N}_{n+1,\varphi}^{d}$) can learn $n+1$ different samples $(\mathbf{x}_i,y_i)$ with zero error (exact approximation), and the weights $\mathbf{w}_j$ and thresholds b_j can be selected "almost" arbitrarily. Two main types of proofs of this conclusion have been provided. One is analysis mode, which can be founded in [3, 4, 16, 17, 30, 31]. Another is algebraic form, which has constructive features as given in [4, 6, 18–21, 31–33, 42]. Other direct methods of finding one weight are more difficult and burdensome [4, 18]. From the process of the proof in these references, we can see that it is basically invalid, since almost all the algebraic methods and other direct approaches need to solve a $(n+1)\times(n+1)$ matrix or its inverse matrix. Particularly, when the number of units is large.

2 Description of problems

The problems considered in this paper are as follows: In previous studies ([1–30, 30–41] about the density or complexity of approximation, interpolation), the weights and thresholds in FNNs vary and so the theoretical results are very difficult to be applied in approximate calculation and other aspects. In order to make them easy in application, Ismailov [43] studied function approximation by FNNs with weights varying on a finite set of directions. Nageswara [44] considered learning a function f by using feed-forward sigmoid networks with a single hidden layer and bounded weights. For any continuous function on a compact subset of R, Chui and Li [11] established a density result by FNNs with a sigmoidal function having integer weights and thresholds. Ito [12] proved that the FNNs with sigmoidal functions having unit weights can approximate any continuous function to arbitrary precision on a compact subset of R.

What if the weights and thresholds in FNNs are fixed? Are these kinds of FNNs possible to approximate arbitrary continuous functions in this case? For all we know, this question was first solved by Hahm and Hong [45]. They showed that a FNNs with a sigmoidal activation function and fixed weights can approximate any function to arbitrary precision in C_0 on R. Unfortunately, the findings mentioned above almost are qualitative in feature [45]. Actually, from the application point of view, however, the quantitative study of FNNs approximation is more useful.

The so-called quantitative research is the upper and lower bounds estimations of the neural network approximation ability. (If the upper and lower bounds estimations have the same order, then we call the order of the bounds as the essential order of approximation [46]). In the past ten years, Xu has led the team to create a precedent in the field of this research, and has made a series of important theoretical results, which lays some key foundations for further research on the complexity of the neural network approximation [46–53]. Certainly, these important results are more inclined to theoretical research and are not easy to practice in reality, because the inner universal functions are highly non-smooth or incomputable [47, 52], or infinite differentiability [46, 47, 51, 53], or the number of hidden neurons is exceedingly large [46, 50, 52], or the weights and thresholds in FNNs are variable [51–53].

Based on the above systematical analysis, the following problems arise naturally:

Problem 2.1 Can we fix the weights and thresholds in FNNs and provide the quantitative study of their approximation ability to make them easy in theory or application?

Problem 2.2 How the approximation capability of a FNN with the fixed weights and thresholds is related to the topology of the network? Loosely speaking, how many hidden units are required in order for this network to reach a predetermined approximation precision?

Problem 2.3 Is there a way to get the weights and the thresholds of an exact FNN approximation without training? In other words, is there an effective method to find the fixed weights and thresholds in FNN to satisfy the interpolation conditions?

The purpose of this paper is to solve all problems mentioned above by constructing three types of FNNs with optimized activation functions and fixed weights and thresholds and establishing the quantitative approximation theorems. In the following, optimized activation functions and three types of FNNs are defined and then, in Sect. 4, some approximation and interpolation results are obtained. In Sect. 5, applying the theoretical results obtained in this paper, we demonstrate some numerical approximation and interpolation results that show good agreement with theoretical results. Finally, we summarize the paper and foresee problems for the further study.

3 Optimized activation function and constructed FNNs

In this section, we wish to study the activation function usually used in the literature. In fact, a neuron cannot stay excited or inhibited indefinitely. Therefore, in this work, we appoint there exist excitement and inhibition. Based on this hypothesis, we define triangular and trapezoidal units. By using these neurons, we can provide many activation functions, which indicate naturally why the neural network has the ability of uniform approximation. In what follows, we appoint that $C[a, b]$ is the set of all continuous functions $f : [a, b] \to R$ defined on the bounded interval $[a, b]$.

Let $\sigma : R \to [0, c]$ be the ramp function defined by

$$\sigma(x) := \begin{cases} 0, & x \leq -\mu_0, \\ c, & x \geq \mu_0, \\ \dfrac{x+\mu_0}{2\mu_0}c, & -\mu_0 < x < \mu_0, \end{cases} \quad c \in R^+, \quad 0 < \mu_0 \leq \frac{1}{2}. \tag{3.1}$$

Figure 2 exhibits the ramp function that defined by Eq. (3.1).

Remark 3.1 The ramp transfer function defined above is an example of sigmoidal activation function when $c = 1$. If $c = 1$, and $\mu_0 = 1/2$, it is the same as in [22].

We are now trying to construct a new function φ_1 by using the ramp functions. We define

$$\varphi_1(x) := \sigma(x+\mu_0) - \sigma(x-\mu_0) = \begin{cases} 0, & |x| \geq 2\mu_0, \\ \left(1 - \dfrac{1}{2\mu_0}|x|\right)c, & |x| < 2\mu_0, \end{cases} \quad c \in R^+, \quad 0 < \mu_0 \leq \frac{1}{2}. \tag{3.2}$$

Thus, Eq. (3.2) illustrates the triangle function (see Fig. 3).

Figures 2 and 3 exhibit the activation functions with unbounded excitement or unbounded inhibition. In fact, from a biological point of view, excitement and inhibition

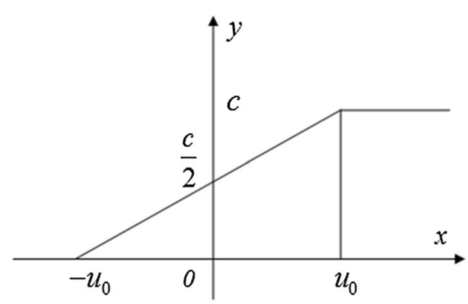


Fig. 2 Ramp activation function

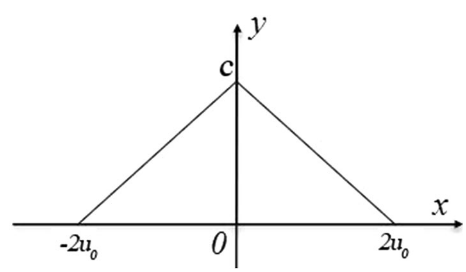


Fig. 3 Triangle activation function

should not happen abruptly. There should have balance (buffer) zones for both excitement and inhibition. For that reason, we can define a more reasonable and nonnegative activation function as follows:

$$\varphi_2(x) := \sigma(x+2\mu_0) - \sigma(x-2\mu_0) = \begin{cases} 0, & |x| \geq 2\mu_0, \\ \left(1-\frac{1}{\mu_0}|x|\right)c, & \mu_0 < |x| < 2\mu_0, c \in R^+, \quad 0<\mu_0 \leq \frac{1}{2}, \\ c, & |x| \leq \mu_0, \end{cases} \tag{3.3}$$

which optimizes the ramp activation function and the triangle activation function. We now see that Eq. (3.3) shows a trapezoidal activation function, which is illustrated in Fig. 4.

Remark 3.2 The ramp activation functions, triangle activation functions and trapezoidal activation functions are all piecewise linear activation functions.

The triangle functions $\varphi_1(x)$ and the trapezoidal functions $\varphi_2(x)$ have the following helpful properties:

P_1 : Both $\varphi_1(x)$ and $\varphi_2(x)$ are even functions;
P_2 : Both $\varphi_1(x)$ and $\varphi_2(x)$ are non-decreasing for $x<0$ and non-increasing for $x>0$;
P_3 : $\mathrm{Supp}(\varphi_j) \subseteq [-c, c], j = 1, 2$.

We now consider the uniform space nodes $x_k = a + kh, k = 0, 1, \ldots, n$, on the interval $[a, d]$, where $h = \frac{d-a}{n} = \frac{2(b-a)}{n}, b = \frac{d+a}{n}$.

Now, we are able to construct three types of FNNs based on piecewise line activation functions $\varphi_1(x)$ and $\varphi_2(x)$ above defined.

Definition 3.1 If $n \in N^+$ and $f : [a, d] \to R$ is a bounded and measurable function, then we construct the FNNs with optimized piecewise line activation functions $\varphi_1(x)$ and $\varphi_2(x)$ as follows:

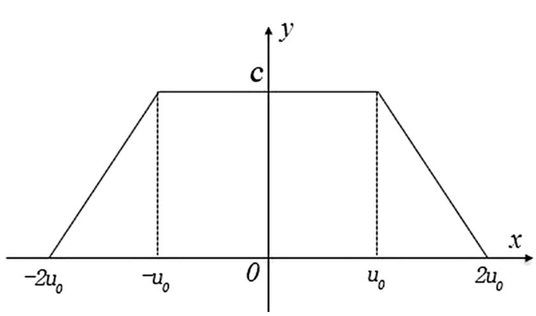


Fig. 4 Trapezoidal activation function

$$N_{n,j}(f, x) := \frac{\sum_{k=0}^{n} f(x_k)\varphi_j\left(\frac{n}{2(b-a)}x - \frac{n}{2(b-a)}x_k\right)}{\sum_{k=0}^{n} \varphi_j\left(\frac{n}{2(b-a)}x - \frac{n}{2(b-a)}x_k\right)}, \quad x \in [a, d], \quad j = 1, 2. \tag{3.4}$$

Remark 3.3 Actually, in [22], such similar neural network, are called "interpolation neural network operators" and have been introduced and studied in case of $\varphi_1(x)$ only with parameter $c = 1$, i.e., when σ is a sigmoidal function, and with the parameter $\mu_0 = 1/2$, the step $h = \frac{b-a}{n}$.

Certainly, the FNNs can be established by using other types of sigmoidal functions as activation function. Denoted by

$$M_s(x) := \frac{1}{(s-1)!}\sum_{i=0}^{s}(-1)^i\binom{s}{i}(\mu_0 s + x - i)_+^{s-1}, \quad x \in R,$$

the well-known B-splines of order $s \in N^+$ [54]. Here and hereafter, the function $(x)_+ := \max\{x, 0\}$ represents the positive past of x. The functions M_s have compact support with $\mathrm{supp}(M_s) \subseteq [-s/2, s/2]$ for arbitrary $s \in N^+$. Only if $\mu_0 = 1/2$, the $M_s(x)$ are the well-known central B-splines. We recall the definition of other kinds of sigmoidal functions $\sigma_{M_s}(x)$, firstly introduced in [34].

$$\sigma_{M_s}(x) := \int_{-\infty}^{x} M_s(t)\mathrm{d}t, \quad x \in R.$$

We can easily find that $\sigma_{M_1}(x)$ accords exactly with the piecewise linear function $\sigma(x)$. Now, we can define the following nonnegative activation functions:

$$\varphi_s(x) := \sigma_{M_s}(x+\mu_0) - \sigma_{M_s}(x-\mu_0), \quad x \in R, \tag{3.5}$$

for any $s \in N^+$. Similarly to the case of the piecewise linear functions φ_1 and φ_2, the function $\varphi_s(x)$ possesses the following properties:

P_4: $\varphi_s(x)$ is an even function;
P_5: $\varphi_s(x)$ is non-decreasing for $x<0$ and non-increasing for $x \geq 0$;
P_6: $\mathrm{supp}(\varphi_s) \subseteq [-K_s, K_s] := [-\mu_0(s+1), \mu_0(s+1)]$, and $\varphi_s\left(\frac{K_s}{2}\right) = \varphi_s\left(\frac{\mu_0(s+1)}{2}\right) > 0$.

Definition 3.2 For any bounded and measurable function $f : [a, d] \to R$, the approximator: FNNs with activation function φ_s are defined by

$$N_{n,s}(f, x) := \frac{\sum_{k=0}^{n} f(x_k)\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)}{\sum_{k=0}^{n} \varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)}, \quad x \in [a, d]. \tag{3.6}$$

Remark 3.4 In fact, when $s = 1$, the FNNs defined in (3.6) degenerate to those recalled in (3.4). Also the neural networks $N_{n,s}(f,x)$ have been originally defined and studied in [22] in case of $\mu_0 = 1/2$, and the step $h = \frac{b-a}{n}$.

4 Theoretical results

In the current note, the following quantitative approximation results for the family of FNNs $N_{n,j}, j = 1, 2$ and $N_{n,s}, s \in N^+$ can be proved. For every $r \in N^+$, $\delta > 0$, and any function $f \in C[a,d]$, we now recall the well-known definition of rth order modulus of smoothness, and the Lipschitzian function class of f as follows [55]:

$$\omega_r(f,\delta) := \sup_{a \le x, x+rt \le b, |t| \le \delta} |\Delta_t^r f(x)|,$$

where $\Delta_t^r f(x) := \Delta_t^1 \Delta_t^{r-1} f(x)$, and $\Delta_t^1 f(x) := f(x+t) - f(x)$. Note that, when $r = 1$, we have $\omega_1(f,\delta) = \omega(f,\delta)$. Namely, the first order modulus of smoothness of f is the same as modulus of continuity of f.

$$\mathrm{Lip(f,\alpha)_2 := \{f | \omega_2(f,\delta) \le M\delta^\alpha, \alpha \in (0,2],}$$
$$\text{M is a positive constant}\}.$$

The rth order modulus of smoothness possesses the following helpful properties:

1. $\omega_r(f,\delta)$ is a monotonically increasing continuous function about δ, and $\omega_r(f,0) = 0$;
2. If $0 \le s < r$, then $\omega_r(f,\delta) \le 2^{r-s}\omega_s(f,\delta)$;
3. If $0 < \delta < \eta$, then $0 < \omega_r(f,\eta) - \omega_r(f,\delta) \le 2^r r\omega(f,\eta-\delta)$, and $\eta^{-r}\omega_r(f,\eta) \le 2^r\delta^{-r}\omega_r(f,\eta-\delta)$;
4. $\omega_r(f,p\delta) \le p^r\omega_r(f,\delta)$ for any $p \in N^+$;
5. $\omega_r(f,q\delta) \le \omega_r(f,[q+1]\delta) \le (q+1)^r\omega_s(f,\delta)$ for arbitrary non-natural number $q > 0$;
6. If f has rth order continuous derivatives, then $\omega_r(f,\delta) \le \delta^r ||f^{(r)}||_C$, and $\omega_{r+s}(f,\delta) \le \delta^r\omega_s(f^{(r)},\delta)$.

4.1 Upper bound of approximation

For any continuous functions defined on $[a,d]$, the following upper bound estimations theorem about the quantitative research on the approximation of FNNs can be proved.

we now give the main results of this subsection.

Theorem 4.1 *Let $f \in C[a,d]$ be fixed. Then*

$$||N_{n,1}(f,x) - f(x)||_\infty \le 4\omega_2\left(f, \frac{b-a}{n}\right), \quad \forall n \in N^+, \tag{4.1}$$

$$||N_{n,2}(f,x) - f(x)||_\infty \le 4\omega_2\left(f, \frac{b-a}{n}\right), \quad \forall n \in N^+, \tag{4.2}$$

$$||N_{n,s}(f,x) - f(x)||_\infty \le \frac{2}{\varphi_s(\frac{K_s}{2})}\omega_2\left(f, \frac{b-a}{n}\right), \quad \forall n, s \in N^+. \tag{4.3}$$

Remark 4.1 The Theorem 4.1 is the positive theorem of approximation of the three kinds of FNNs. These approximation upper bounds are inspired to the results originally in [22], in case of $\varphi_1(x)$, with $c = 1$, $\mu_0 = 1/2$, and the step $h = \frac{b-a}{n}$, and for $\varphi_2(x)$ with $\mu_0 = 1/2$, the step $h = \frac{b-a}{n}$. Moreover, Eqs. (4.1) and (4.3) obviously deepen the results proved in [22] (For instance, if $g(x) = x^n, n \in N$ is a polynomial, then $\omega(g,t) = \bigcirc(t)$, while $\omega_2(g,t) = \bigcirc(t^2)$.), and Eq. (4.2) represents a completely new result.

Proof In fact, we find that for each $x \in [a,d]$, by using P_1, and then, we obtain

$$\sum_{k=0}^{n} \varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right) = \sum_{k=0}^{n} \varphi_j\left(\frac{n|x-x_k|}{2(b-a)}\right) \ge \varphi_j\left(\frac{n|x-x_i|}{2(b-a)}\right), \quad j = 1, 2, \tag{4.4}$$

where $i \in \{0, 1, \ldots, n\}$ satisfies $|x - x_i| \le \mu_0 h$. Thus,

$$\frac{n|x-x_i|}{2(b-a)} \le \frac{n\mu_0 h}{2(b-a)} = \mu_0, \tag{4.5}$$

By Eqs. (4.4), (4.5) and P_2, we have

$$\sum_{k=0}^{n} \varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right) \ge \varphi_j\left(\frac{n|x-x_i|}{2(b-a)}\right) \ge \varphi_j(\mu_0) = \begin{cases} \frac{1}{2}c, & j = 1, \\ c, & j = 2. \end{cases} \tag{4.6}$$

In addition, for any bounded and measurable function $f : [a,d] \to R$, we obtain

$$|N_{n,j}(f,x)| \le ||f||_\infty \frac{\sum_{k=0}^{n} \varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right)}{\sum_{k=0}^{n} \varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right)} = ||f||_\infty < +\infty, \quad j = 1, 2,$$

for each $x \in [a,d]$, where $||f||_\infty := \sup_{x \in [a,d]} |f(x)|$.

For each $x \in [a,d]$ and by (4.6), we know that

$$\begin{aligned} &\left|N_{n,1}(f,x) - f(x)\right| \\ &= \frac{\left|\sum_{k=0}^{n} f(x_k)\varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right) - f(x)\sum_{k=0}^{n}\varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right)\right|}{\sum_{k=0}^{n}\varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right)} \\ &\le \frac{2}{c}\left|\sum_{k=0}^{n} f(x_k)\varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right) - f(x)\sum_{k=0}^{n}\varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right)\right| \\ &\le \frac{2}{c}\sum_{k=0}^{n}|f(x_k) - f(x)|\varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right), \end{aligned}$$

for every fixed $n \in N^+$. We now choose $i \in \{0, 1, \ldots, n-1\}$ such that $x_i \le x \le x_{i+1}$, then

$$\begin{aligned}\left|N_{n,1}(f,x) - f(x)\right| &\le \frac{2}{c}\left[\sum_{k=0,k\neq i,i+1}^{n} |f(x_k) - f(x)|\varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right)\right.\\ &\quad + |f(x_i) - f(x)|\varphi_1\left(\frac{n(x-x_i)}{2(b-a)}\right)\\ &\quad \left. + |f(x_{i+1}) - f(x)|\varphi_1\left(\frac{n(x-x_{i+1})}{2(b-a)}\right)\right]\\ &=: \frac{2}{c}[I_1 + I_2 + I_3].\end{aligned}$$

Obviously, for $k \neq i, i+1$, we obtain $\frac{n|x-x_k|}{2(b-a)} \ge \frac{nh}{2(b-a)} = 1$, then by the properties P_1, P_2 and P_3, we have

$$\begin{aligned}0 \le \varphi_1\left(\frac{n(x-x_k)}{2(b-a)}\right) = \varphi_1\left(\frac{n|x-x_k|}{2(b-a)}\right) \le \varphi_1\left(\frac{nh}{2(b-a)}\right)\\ = \varphi_1(1) = 0,\end{aligned}$$

which implies, $I_1 = 0$. Since $|x_i - x| \le h$ and $|x_{i+1} - x| \le h$, we can find

$$|f(x_i) - f(x)| \le \omega_2\big(f, \frac{b-a}{n}\big).$$

Similarly,

$$|f(x_{i+1}) - f(x)| \le \omega_2\left(f, \frac{b-a}{n}\right).$$

Finally, we obtain

$$I_1 + I_2 + I_3 = I_2 + I_3 \le 2c\omega_2\left(f, \frac{b-a}{n}\right).$$

With this, the proof of (4.1) in Theorem 4.1 is completed. Applying the same method used in proof of (4.1) in Theorem 4.1, we can prove (4.2) in Theorem 4.1. We omit the details. Next, we will prove (4.3) in Theorem 4.1.

Employing the technique similar to that adopted in Eqs. (4.4)–(4.6), it is easily to prove that the above FNNs (3.6) are well defined for arbitrary $n \in N^+$ and

$$\sum_{k=0}^{n} \varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right) \ge \varphi_s\left(\frac{K_s}{2}\right) > 0.$$

Now, for each $x \in [a, d]$, and $f \in C[a, d]$, there exists $i \in \{0, 1, 2, \ldots, n-1\}$ such that $x_i \le x \le x_{i+1}$, and then we have

$$\begin{aligned}&\left|N_{n,s}(f,x) - f(x)\right|\\ &\quad = \frac{\left|\sum_{k=0}^{n} f(x_k)\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right) - f(x)\sum_{k=0}^{n}\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)\right|}{\sum_{k=0}^{n}\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)}\\ &\quad \le \frac{1}{\varphi_s(K_s/2)}\left|\sum_{k=0}^{n} f(x_k)\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)\right.\\ &\quad \left. - f(x)\sum_{k=0}^{n}\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)\right|\\ &\quad \le \frac{1}{\varphi_s(K_s/2)}\sum_{k=0}^{n}|f(x_k) - f(x)|\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)\\ &= \frac{1}{\varphi_s(K_s/2)}\left[\sum_{k=0,k\neq i,i+1}^{n}|f(x_k) - f(x)|\varphi_s\left(K_s\frac{n(x-x_k)}{2(b-a)}\right)\right.\\ &\quad + |f(x_i) - f(x)|\varphi_s\left(K_s\frac{n(x-x_i)}{2(b-a)}\right)\\ &\quad \left. + |f(x_{i+1}) - f(x)|\varphi_s\left(K_s\frac{n(x-x_{i+1})}{2(b-a)}\right)\right]\\ &=: \frac{1}{\varphi_s(K_s/2)}[J_1 + J_2 + J_3].\end{aligned}$$

The addends J_1, J_2 and J_3 can be handled as made in the proof of (4.1) in Theorem 4.1, and then, (4.3) in Theorem 1 follows immediately. This completes the proof of the Theorem 4.1.

4.2 Lower bound of approximation

Until now, many results about density and upper bound estimations on approximation of the functions by the FNNs (1.1) in the set $\mathbb{N}_{n+1}^{d}(\phi)$ were given by many researchers [2, 9, 23, 24, 26–33, 54–65]. In fact, because the established upper bound estimations can only control one side of the approximation error, the estimation results might be too loose to perfectly reflect the approximation capability of the FNNs. Naturally, in order to characterize the approximation ability of FNNs more precisely, besides upper bound estimation, a lower bound estimation that reflects the worst approximation precision of the network is still a question that is worth studying. Therefore, we emphasize that the results of this subsection are completely new.

We now give the main results of this subsection.

Theorem 4.2 *Let $f \in C[a, d]$ be fixed. Then*

$$\omega_2\left(f, \frac{b-a}{n}\right) \le \frac{C}{n}\sum_{i=1}^{n}||N_{i,j}(f,x) - f(x)||_\infty, \quad j = 1, 2, n \in N^+, \tag{4.7}$$

$$\omega_2\left(f, \frac{b-a}{n}\right) \le \frac{C}{n}\sum_{i=1}^{n}||N_{i,s}(f,x) - f(x)||_\infty, \quad \forall s \in N^+. \tag{4.8}$$

Here and hereafter, C that appears in different situations may be different but all are positive constants independent of n and f.

Remark 4.2 We emphasize that the results of this subsection are completely new. The Theorem 4.2 is the converse theorem of approximation of the three kinds of FNNs. These conclusions reveal three lower bound estimations on approximation precision of these FNNs, which means that the average of these FNNs over the number of hidden neurons is lower controlled by the second order modulus of smoothness of function f.

In order to prove Theorem 4.2, we will use the famous Bernstein polynomial and the extended Bernstein polynomial as two basic tools. Let $f \in C[0,1]$, the sequence of Bernstein polynomials for $f(x)$ is defined by [66]

$$B_n(f,x) := \sum_{k=0}^{n} f\left(\frac{k}{n}\right)\binom{n}{k} x^k (1-x)^{n-k}, \quad x \in [0,1].$$

Similarly, if $f \in C[a,d]$, then we define the extended Bernstein polynomials for $f(x)$ as follows

$$\mathrm{EB}_n(f,x) := \sum_{k=0}^{n} f\left(a + \frac{k}{n}(d-a)\right)\binom{n}{k}\left(\frac{x-a}{d-a}\right)^k \left(1 - \frac{x-a}{d-a}\right)^{n-k}, \quad x \in [a,d].$$

For nearly half a century, the Bernstein polynomial and its improvement have attracted much interest, and a great number of interesting results to the classical Bernstein polynomial have been obtained [66–69].

The following fundamental result on classical Bernstein polynomial is well known [55]. In order to facilitate readers, we give the lemmas as follows:

Lemma 4.1 ([55]) *Let* $f \in C[0,1]$. *Then there is positive constant* C *such* that

$$w_2\left(f;\frac{1}{n}\right) \le \frac{C}{n}\sum_{k=1}^{n} ||B_k(f,x) - f(x)||_\infty.$$

According to the Lemma 4.1, we can easily get the following Lemma 4.2.

Lemma 4.2 *If* $f \in C[a,d]$. *Then there is positive constant* C *such that*

$$w_2\left(f;\frac{b-a}{n}\right) \le w_2\left(f;\frac{d-a}{n}\right) \le \frac{C}{n}\sum_{k=1}^{n} ||\mathrm{EB}_k(f,x) - f(x)||_\infty.$$

Lemma 4.3 ([45]) *Let* $f \in C[a,d]$. *If* σ *is a bounded measurable sigmoidal function on* R. *Then, for any* $\epsilon > 0$, *there is a neural network* $N_n(x)$ *of the form* (1.1), *such that*

$$|N_n(x) - f(x)| < \epsilon,$$

where $N(x) = \sum_{i=1}^{n} c_i\sigma(w_i x + \theta), c_i, w_i, \theta \in R$.

We now are to prove Theorem 4.2. First, we demand an equivalent description of the extended Bernstein operator.

$$\begin{aligned}
\mathrm{EB}_n(f,x) &= \sum_{k=0}^{n} f\left(a + \frac{k}{n}(d-a)\right)\binom{n}{k}\left(\frac{x-a}{d-a}\right)^k\left(1 - \frac{x-a}{d-a}\right)^{n-k} \\
&= \sum_{k=0}^{n} f\left(a + \frac{k}{n}(d-a)\right)\binom{n}{k}\left(\frac{x-a}{d-a}\right)^k \\
&\quad \times \left(1 + C_{n-k}^{1}\left(-\frac{x-a}{d-a}\right) + \cdots + C_{n-k}^{i}\left(-\frac{x-a}{d-a}\right)^i + \cdots + \left(-\frac{x-a}{d-a}\right)^{n-k}\right) \\
&= \sum_{k=0}^{n} f\left(a + \frac{k}{n}(d-a)\right)\binom{n}{k}\sum_{i=0}^{n-k}(-1)^i\frac{(n-k)!}{i!(n-k-i)!}\left(\frac{x-a}{d-a}\right)^{i+k} \\
&= \sum_{k=0}^{n}\sum_{i=0}^{n-k} f\left(a + \frac{k}{n}(d-a)\right)\binom{n}{k}(-1)^i\frac{(n-k)!}{i!(n-k-i)!}\left(\frac{x-a}{d-a}\right)^{i+k} \\
&= \sum_{k=0}^{n}\sum_{i=0}^{n-k} d_{i,k}\left(\frac{x-a}{d-a}\right)^{i+k}, \quad x \in [a,d],
\end{aligned} \tag{4.9}$$

where $d_{i,k} = (-1)^i f(a + \frac{k}{n}(d-a))\binom{n}{k}\frac{(n-k)!}{i!(n-k-i)!}$.

Second, let r be a fixed integer and $P_r(x) = a_r x^r, x \in [a,d]$ be a univariate polynomial of degree r. By Lemma 4.3, we know that there is a FNN of the form (1.1) the number of whose hidden units is not less than $(r+1)$ such that

$|N_n(x) - P_r(x)| < \epsilon.$

We find that in (4.9) each term is a univariate polynomial of x with order $i+k(i+k \leq n)$; therefore, it can be approximated arbitrarily well by a FNN of the form

$$N_{i+k+1}(x) = \sum_{l=1}^{K_{i+k}} c_{l,i+k}\sigma(w_{l,i+k}x+\theta), \quad K_{i+k+1} \geq i+k+1. \tag{4.10}$$

Because $B_n(f,x)$ and $\mathrm{EB}_n(f,x)$ can approximate f, the following FNNs

$$\sum_{k=0}^{n}\sum_{i=0}^{n-k} d_{i,k}\sum_{l=1}^{K_{i+k}} c_{l,i+k}\sigma(w_{l,i+k}x+\theta), \quad c_{l,i+k}, w_{l,i+k} \in R,$$
$$K_{i+k} \geq i+k+1$$

can approximate f to any accuracy. In consequence, the networks (4.10) will be the FNN models we propose to use in this subsection.

At the third stage, according to Lemma 4.3, the polynomial $x^{i+k}(i+k \leq n)$ can be approximated by a network having the following form

$$N_{K_{i+k}} = \sum_{l=1}^{K_{i+k}} c_{l,i+k}\sigma(w_{l,i+k}x+\theta), \quad c_{l,i+k}, w_{l,i+k} \in R,$$
$$K_{i+k} \geq i+k+1, \tag{4.11}$$

with approximation error

$$|N_{K_{i+k}} - x^{i+k}| < \epsilon. \tag{4.12}$$

Equations (4.11) and (4.12) imply

$$\begin{aligned}
||N_m(f,x) - \mathrm{EB}_m(f,x)||_\infty &= \left|\left|\sum_{k=0}^{m}\sum_{i=0}^{m-k} d_{i,k}\left\{x^{i+k} - \sum_{l=1}^{K_{i+k}} c_{l,i+k}\sigma(\omega_{l,i+k}x+\theta)\right\}\right|\right|_\infty \\
&= \sum_{k=0}^{m}\sum_{i=0}^{m-k}|d_{i,k}|\max_{x\in[a,b]}\left|x^{i+k} - N_{K_{i+k}}\right| \\
&\leq \epsilon\sum_{k=0}^{m}\sum_{i=0}^{m-k}|d_{i,k}|.
\end{aligned}$$

Next, taking $f(x_k) = \sum_{i=0}^{m-k}\sum_{l=1}^{K_{i+k}} d_{i,k}c_{l,i+k}$, $\varphi_j = \sigma$, $\frac{n}{2(b-a)} = \omega_{l,i+k}$, and $\theta = \frac{n}{2(b-a)}x_k$ in (3.4), and by (4.6), we have

$$\begin{aligned}
\left|\left|N_{m,j}(f,x) - \mathrm{EB}_m(f,x)\right|\right|_\infty &= \left|\left|\frac{\sum_{k=0}^{m} f(x_k)\varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right)}{\sum_{k=0}^{m}\varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right)} - \mathrm{EB}_m f(x)\right|\right|_\infty \\
&\leq \frac{2}{c}\left|\left|\sum_{k=0}^{m} f(x_k)\varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right) - \mathrm{EB}_m(f,x)\right|\right|_\infty \\
&\leq C\left|\left|\sum_{k=0}^{m} f(x_k)\varphi_j\left(\frac{n(x-x_k)}{2(b-a)}\right) - \mathrm{EB}_m(f,x)\right|\right|_\infty \\
&= C\left|\left|\sum_{k=0}^{m}\sum_{i=0}^{m-k} d_{i,k}\sum_{l=1}^{K_{i+k}} c_{l,i+k}\sigma(w_{l,i+k}x+\theta) - \mathrm{EB}_m(f,x)\right|\right|_\infty \\
&= C||N_m(f,x) - \mathrm{EB}_m(f,x)||_\infty \\
&\leq C\epsilon\sum_{k=0}^{m}\sum_{i=0}^{m-k}|d_{i,k}|.
\end{aligned}$$

Finally, for the constructed FNN

$$\sum_{k=0}^{n}\sum_{i=0}^{n-k} d_{i,k}\sum_{l=1}^{K_{i+k}} c_{l,i+k}\sigma(w_{l,i+k}x+\theta)c_{l,i+k}, w_{l,i+k} \in R,$$
$$K_{i+k} \geq i+k+1,$$

we obtain the lower bound estimation of $||N_{n,j} - f||_\infty, j = 1,2$ as follows:

$$\begin{aligned}
\omega_2\left(f, \frac{b-a}{n}\right) &\leq \frac{C}{n}\sum_{k=1}^{n}||\mathrm{EB}_k(f,x) - f(x)||_\infty \\
&\leq \frac{C}{n}\sum_{k=1}^{n}\{||\mathrm{EB}_k(f,x) - N_{k,j}(f,x)||_\infty + ||N_{k,j}(f,x) - f(x)||_\infty\} \\
&\leq \frac{C}{n}\sum_{k=1}^{n}||N_{k,j}(f,x) - f(x)||_\infty + \frac{C\epsilon}{n}\sum_{k=0}^{n}\sum_{i=0}^{n-k}\sum_{l=1}^{K_{i+k}}|d_{i,k}|.
\end{aligned}$$

Letting ϵ tend to zero, it then follows that

$$\omega_2\left(f,\frac{b-a}{n}\right)\le \frac{C}{n}\sum_{k=1}^{n}||N_{k,j}(f,x)-f(x)||_\infty, j=1,2.$$

Thus the first part of the Theorem 4.2 is proved. The second part of the Theorem 4.2 follows immediately by the same arguments used in the proof process of the first part of the Theorem 4.2. In order not to repeat, we omit the details.

4.3 Essential order of approximation

If the arithmetic means on the right sides of in Eqs. (4.1), (4.2), (4.7) and (4.3), (4.8) can be substituted by $||N_{n,j}(f,\cdot)-f(\cdot)||_\infty, j=1,2$ and $||N_{n,s}(f,\cdot)-f(\cdot)||_\infty$, respectively, then by Theorems 4.1 and 4.2, we obtain that the upper and lower bound estimations of approximation by the FNNs, $N_{n,j}(f), j=1,2$ and $F_{n,s}(f)$, become identical as the second order modulus of smoothness of approximated function f. Namely,

$$\omega_2\left(f,\frac{b-a}{n}\right)\sim ||N_{n,j}(f,\cdot)-f(\cdot)||_\infty,\quad j=1,2,$$

and

$$\omega_2\left(f,\frac{b-a}{n}\right)\sim ||N_{n,s}(f,\cdot)-f(\cdot)||_\infty,\quad s\in N^+.$$

Unfortunately, up to now, we cannot answer these problems for arbitrary function classes. Happily, we can solve them when the approximated function f belonging to the class of second order Lipschitz $\alpha(0<\alpha\le 2)$. From this point of view, the following Theorem 4.3 can be drawn directly by combining the Theorem 4.1 with the Theorem 4.2.

Theorem 4.3 *Let $f\in C[a,d]$ be fixed. Then*

$$||N_{n,j}(f,x)-f(x)||_\infty = \bigcirc(n^{-\alpha})\Leftrightarrow f\in \mathrm{Lip}(\alpha)_2,\quad \text{for } j=1,2.$$
$$||N_{n,s}(f,x)-f(x)||_\infty = \bigcirc(n^{-\alpha})\Leftrightarrow f\in \mathrm{Lip}(\alpha)_2,\quad \text{for } s\in N^+.$$

Remark 4.3 We emphasize that the results of this subsection are completely new. The Theorem 4.3 points out that the inherent approximation order of these three kinds of FNNs is $O(n^{-\alpha})$. Thus, the approximation capability of the three kinds of FNNs is thoroughly decided by the smoothness of approximated functions. That is to say, the better the properties of the approximated function is, the higher the precision of approximation is. But the maximal precision of approximation cannot outperform $O(n^{-\alpha})$.

Remark 4.4 Theorems 4.1, 4.2 and 4.3 are three affirmative answers to the Problems 2.1 and 2.2. The connection weights and the thresholds are, respectively, equal to $\frac{n}{2(b-a)}$ and $\frac{-nx_k}{2(b-a)}$ in the FNNs $N_{n,1}$ and $N_{n,2}$. Moreover, the connection weights and the thresholds in the FNNs $N_{n,s}$ are equal to $K_s\frac{n}{2(b-a)}=\mu_0(s+1)\frac{n}{2(b-a)}$ and $-K_s\frac{nx_k}{2(b-a)}=-\mu_0(s+1)\frac{nx_k}{2(b-a)}$. Because $a,b\in R,\mu_0\in R^+$ are all constants, $s\in N^+$ is the positive integer, and $n\in N^+$ is the number of hidden neurons, we can find the connection weights and the thresholds in the FNNs to satisfy the approximation conditions and do not need to train. It gives the quantitative researches on approximation precision of these FNNs and characterizes the implicit relationship among the precision of approximation, the number of hidden neurons and the smoothness of the approximated function.

4.4 Interpolation

In the current note, the following interpolation (exact approximation) results for the family of FNNs $N_{n,j}, j=1,2$ and $N_{n,s}, s\in N^+$ can be proved.

Theorem 4.4 *If $f:[a,d]\to R$ be a bounded and measurable function and $n\in N^+$, then*

$$N_{n,1}(f,x_i)=f(x_i),\quad \text{for every } i=0,1,\ldots,n, \tag{4.13}$$

$$N_{n,2}(f,x_i)=f(x_i),\quad \text{for every } i=0,1,\ldots,n, \tag{4.14}$$

$$N_{n,s}(f,x_i)=f(x_i),\quad \text{for every } i=0,1,\ldots,n, \text{and } s\in N^+. \tag{4.15}$$

Remark 4.5 The results in Theorem 4.4 are inspired to the results originally in [22], in case of $\varphi_1(x)$, with $c=1$, $\mu_0=1/2$, and the step $h=\frac{b-a}{n}$, and for $\varphi_2(x)$ with $\mu_0=1/2$, and the step $h=\frac{b-a}{n}$. Moreover, Eqs. (4.13) and (4.15) represent a slight extension of the results proved in [22], and Eq. (4.14) represent a completely new result.

Proof Let $i\in\{0,1,\ldots,n\}$ be fixed. If $k=i$, then we obtain

$$\varphi_j\left(\frac{n(x_i-x_k)}{2(b-a)}\right)=\varphi_j(0)=c,\quad j=1,2.$$

While, if $k\ne i$, then we have

$$\frac{n|x_i-x_k|}{2(b-a)}\ge\frac{nh}{2(b-a)}=1.$$

Thus, by using $0<\mu_0\le\frac{1}{2}$ and the properties P_1 and P_2, we obtain that

$$0=\varphi_j(2\mu_0)=\varphi_j(1)\ge\varphi_j\left(\frac{n|x_i-x_k|}{2(b-a)}\right)$$
$$=\varphi_j\left(\frac{n(x_i-x_k)}{2(b-a)}\right)\ge 0,\quad j=1,2.$$

Therefore, we get

$$\varphi_j\left(\frac{n(x_i-x_k)}{2(b-a)}\right)=\begin{cases}c, & i=k,\\ 0, & i\ne k,\end{cases}\quad j=1,2$$

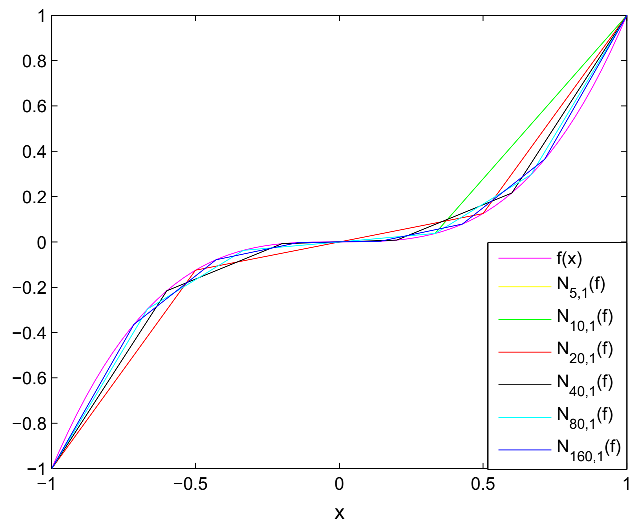


Fig. 5 Target function $f(x) = x^3$ and FNNs: $N_{5,1}(f)$ and $N_{10,1}(f)$ on the interval $[-1,1]$

for each $i, k = 0, 1, \ldots, n$, and this implies that

$$N_{n,j}(f, x_i) = \frac{f(x_i)\varphi_j\left(\frac{n(x_i - x_k)}{2(b-a)}\right)}{\varphi_j\left(\frac{n(x_i - x_k)}{2(b-a)}\right)} = f(x_i), \quad j = 1, 2,$$

for any $i \in \{0, 1, \ldots, \}$. Thus the first part of the Theorem 4.4 is proved.

Next, we will prove the second part of the Theorem 4.4. If $i \neq k$, $i, k = 0, 1, \ldots, n$, then we have $|x_i - x_k| \geq h$, and $K_s \frac{n|x - x_k|}{2(b-a)} \geq K_s$. By the properties of φ_s it turns out that

$$\varphi_s\left(K_s \frac{n(x_i - x_k)}{2(b-a)}\right) = \varphi_s\left(K_s \frac{n|x_i - x_k)|}{2(b-a)}\right) = 0.$$

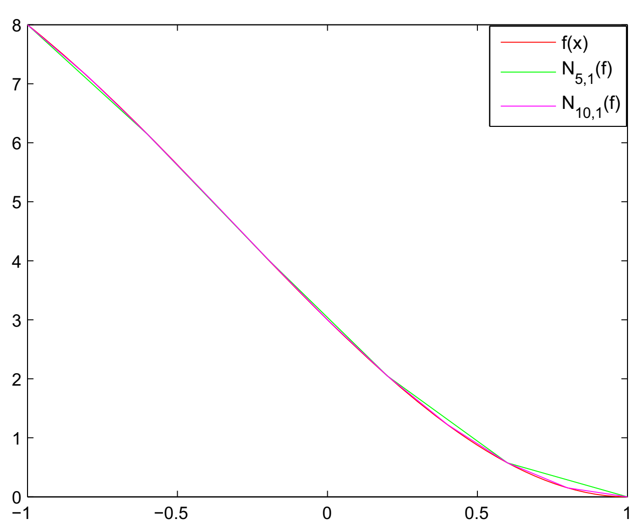


Fig. 6 Target function $f(x) = x^3 + x^2 - 5x + 3$ and FNNs: $N_{5,1}(f)$ and $N_{10,1}(f)$ on the interval $[-1,1]$

Fig. 7 Target function $f(x) = x^3 + x^2 - 5x + 3$ and FNNs: $N_{5,1}(f)$ and $N_{10,1}(f)$ on the neighborhoods of the interpolation points. (*Note*: Fig. 7 is obtained from Fig. 6, which has been magnified to some extent. The interpolation points of $N_{5,1}(f)$ and $N_{10,1}(f)$ are respectively as -1, -0.6, -0.2, 0.2, 0.6, 1 and -1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8 1). **a** The right neighborhood of the point -1. **b** The neighborhood of the point -0.8. **c** The neighborhood of the point -0.6. **d** The neighborhood of the point -0.4. **e** The neighborhood of the point -0.2. **f** The neighborhood of the point 0. **g** The neighborhood of the point 0.2. **h** The neighborhood of the point 0.4. **i** The neighborhood of the point 0.6. **j** The neighborhood of the point 0.8. **k** The left neighborhood of the point 1

Thus the second part of the Theorem 4.4 follows immediately by the same arguments used in the first part of the Theorem 4.4. For the sake of brevity, we omit the details.

Remark 4.6 The Theorem 4.4 shows the interpolation results of these FNNs with the fixed weights and the thresholds, which is the sublimation of the Theorems 4.1 and 4.2 and answers the Problem 2.3 successfully.

5 Numerical results

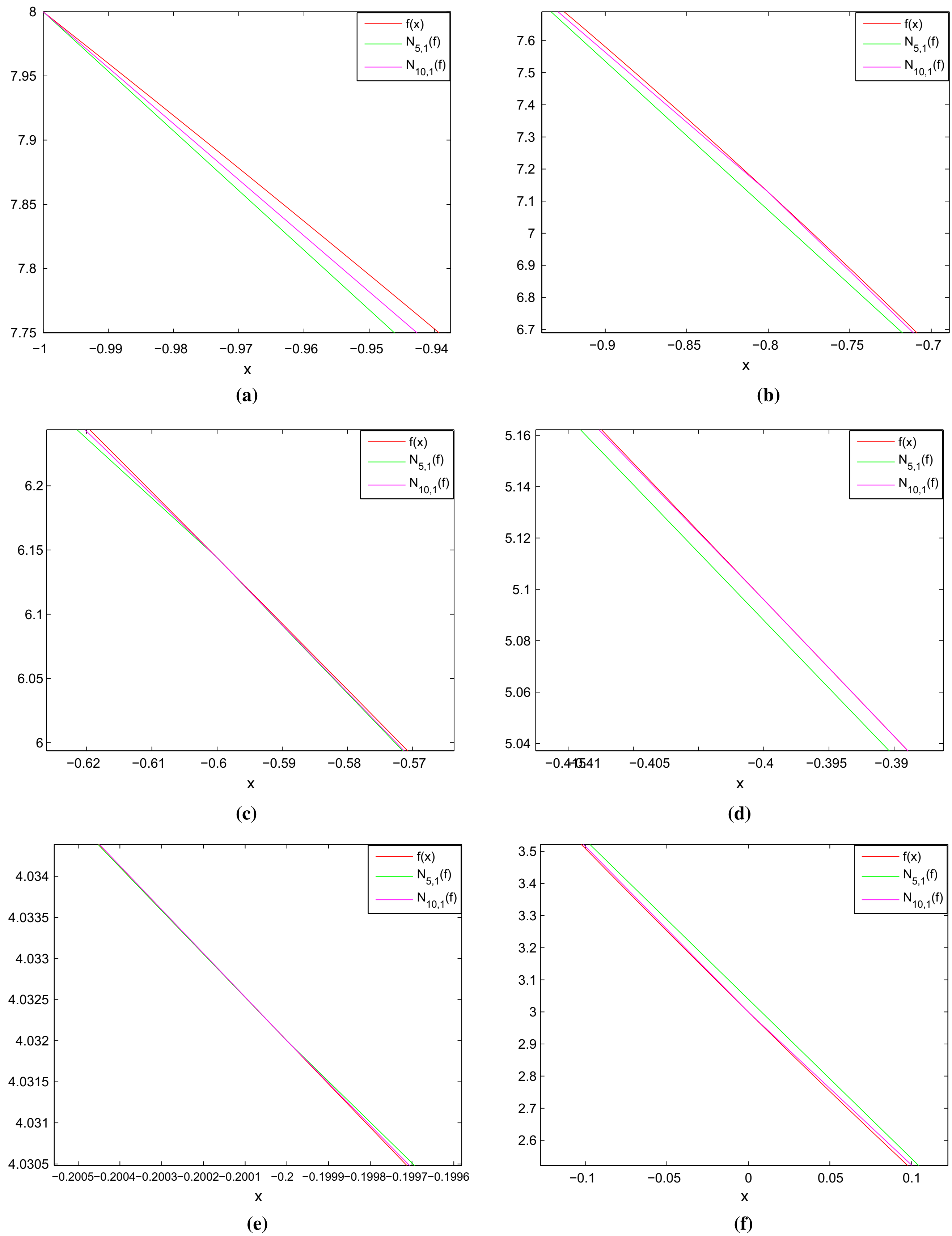
Because continuous functions on bounded interval are normally considered as target functions in engineering and other applications [1, 4–6, 14, 32], we now only focus on a numerical approximation to a continuous function on bounded interval. In Theorems 4.1, 4.2, 4.3 and 4.4, we show that any continuous function on bounded interval $[a, b]$ can be approximated an FNNs with an optimized piecewise line activation function and fixed weights. We show our theoretical results using different activation functions and illustrate the error bound of FNNs approximation. All computations are done in Matalab 7.0.

Example 5.1 We choose a continuous function $f(x) = x^3$ as the target function and research the FNNs with optimized piecewise linear functions and fixed weights (defined by (3.4)) approximation to $f(x)$ on the bounded interval $[a, d]$.

By Eqs. (3.2) and (3.3), and letting $c = 1$, we have

$$\varphi_1\left(\frac{n(x - x_k)}{2(b-a)}\right) = \begin{cases} 0, & |x - x_k| \geq \frac{4(b-a)}{n}\mu_0, \\ 1 - \frac{1}{2\mu_0}\left|\frac{n}{2(b-a)}(x-a) - k\right|, & |x - x_k| < \frac{4(b-a)}{n}\mu_0, \end{cases}$$

and


f(x)
N5,1(f)
N10,1(f)
x
(a)
(b)
(c)
(d)
(e)
(f)

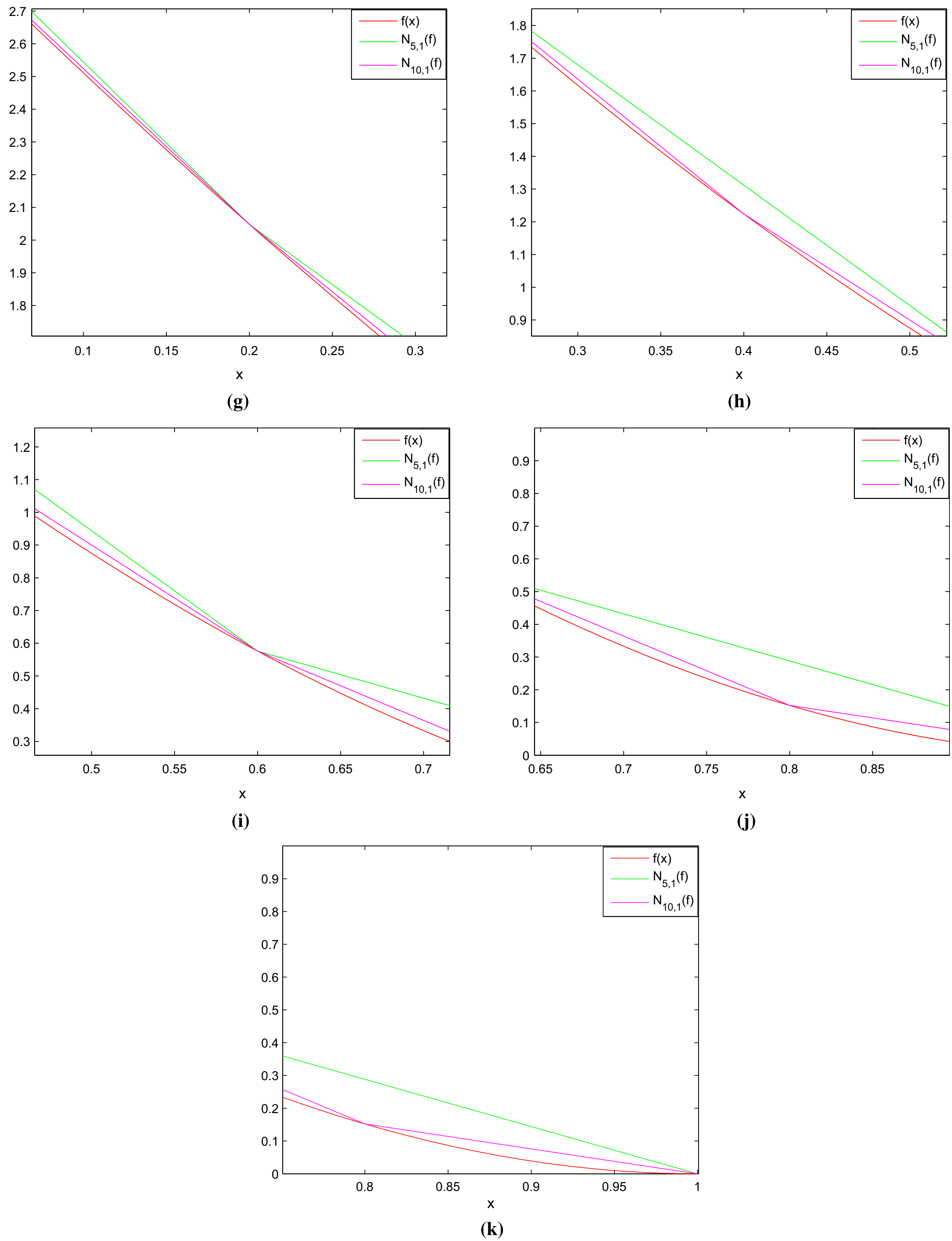


Fig. 7 continued

$$\varphi_2\left(\frac{n(x-x_k)}{2(b-a)}\right)=\begin{cases}0, & |x-x_k|\geq\frac{4(b-a)}{n}\mu_0,\\ 1-\frac{1}{\mu_0}\left|\frac{n}{2(b-a)}(x-a)-k\right|, & \frac{2(b-a)}{n}\mu_0<|x-x_k|<\frac{4(b-a)}{n}\mu_0,\\ 1, & |x-x_k|\leq\frac{2(b-a)}{n}\mu_0.\end{cases}$$

Then, the FNNs $N_{n,1}$ and $N_{n,2}$ are simply reduced to

$$N_{n,1}(f)=\begin{cases}0, & |x-x_k|\geq\frac{4(b-a)}{n}\mu_0,\\ \frac{\sum_{k=0}^{n}f(x_k)\left(1-\frac{1}{2\mu_0}\left|\frac{n}{2(b-a)}(x-a)-k\right|\right)}{\sum_{k=0}^{n}\left(1-\frac{1}{2\mu_0}\left|\frac{n}{2(b-a)}(x-a)-k\right|\right)}, & |x-x_k|<\frac{4(b-a)}{n}\mu_0,\end{cases}\tag{5.1}$$

and

$$N_{n,2}(f)=\begin{cases}0, & |x-x_k|\geq\frac{4(b-a)}{n}\mu_0,\\ \frac{\sum_{k=0}^{n}f(x_k)\left(1-\frac{1}{\mu_0}\left|\frac{n}{2(b-a)}(x-a)-k\right|\right)}{\sum_{k=0}^{n}\left(1-\frac{1}{\mu_0}\left|\frac{n}{2(b-a)}(x-a)-k\right|\right)}, & \frac{2(b-a)}{n}\mu_0<|x-x_k|<\frac{4(b-a)}{n}\mu_0,\\ \left|\frac{\sum_{k=0}^{n}f(x_k)}{n+1}-f(x)\right|, & |x-x_k|\leq\frac{2(b-a)}{n}\mu_0.\end{cases}\tag{5.2}$$

where x_k, a, b, and μ_0 are defined exactly the same as before.

From Eqs. (5.1) and (5.2), we know that the connection weights from the input layer to the hidden layer and from the hidden layer to the output layer, and the thresholds are, respectively, equal to $\frac{n}{2(b-a)}$, $f(x_k)$ and $\frac{-nx_k}{2(b-a)}$ in the FNNs $N_{n,1}$ and $N_{n,2}$. Because $a,b\in R,f(x_k),\mu_0\in R^+$ are all constants, $s\in N^+$ is the positive integer, and $n\in N^+$ is the number of hidden neurons, so we can find that the weights and the thresholds in the FNNs $N_{n,1}$ and $N_{n,2}$ are all fixed constants. Therefore, the error of approximation is

$$\left|N_{n,1}(f,x)-f(x)\right|=\begin{cases}|f(x)|, & |x-x_k|\geq\frac{4(b-a)}{n}\mu_0,\\ \left|\frac{\sum_{k=0}^{n}f(x_k)\left(1-\frac{1}{2\mu_0}\left|\frac{n}{2(b-a)}(x-a)-k\right|\right)}{\sum_{k=0}^{n}\left(1-\frac{1}{2\mu_0}\left|\frac{n}{2(b-a)}(x-a)-k\right|\right)}-f(x)\right|, & |x-x_k|<\frac{4(b-a)}{n}\mu_0,\end{cases}\tag{5.3}$$

and

$$\left|N_{n,2}(f,x)-f(x)\right| = \begin{cases} |f(x)|, & |x-x_k| \geq \dfrac{4(b-a)}{n}\mu_0, \\ \left|\dfrac{\sum_{k=0}^{n} f(x_k)\left(1-\dfrac{1}{\mu_0}\left|\dfrac{n}{2(b-a)}(x-a)-k\right|\right)}{\sum_{k=0}^{n}\left(1-\dfrac{1}{\mu_0}\left|\dfrac{n}{2(b-a)}(x-a)-k\right|\right)} - f(x)\right|, & \dfrac{2(b-a)}{n}\mu_0 < |x-x_k| < \dfrac{4(b-a)}{n}\mu_0, \\ |1-f(x)|, & |x-x_k| \leq \dfrac{2(b-a)}{n}\mu_0. \end{cases} \tag{5.4}$$

We let $a=-1, d=1$, and $\mu_0=\frac{1}{4}$ in Eqs. (5.3) and (5.4). The following Fig. 5 shows a numerical result on $[-1,1]$ for $n=5,10,20,40,80,160$.

Example 5.2 We also choose a continuous function $f(x)=x^3+x^2-5x+3$ as the target function and research the FNNs with optimized piecewise linear functions and fixed weights (defined by (3.4)) approximation to $f(x)$ on the bounded interval $[a, d]$.

We also let $a=-1, d=1$, and $\mu_0=\frac{1}{4}$ in Eqs. (5.3) and (5.4). The following Figs. 6 and 7 provides some numerical results on the neighborhoods of the interpolation points for $n=5,10$. (*Note*: Figure 7 is obtained from Fig. 6, which has been magnified to some extent. The interpolation points of $N_{5,1}(f)$ and $N_{10,1}(f)$ are respectively as − 1.0, − 0.6, − 0.2, 0.2, 0.6, 1.0 and − 1.0, − 0.8, − 0.6, − 0.4, − 0.2, 0, 0.2, 0.4, 0.6, 0.8, 1.0).

Example 5.3 We choose nonnegative density functions φ_s (defined by Eq. (3.5)) as an activation function. We can compute the theoretical error bound like the Example 5.1. We omit the graphs of the nonnegative density functions neural network $N_{n,s}(f,x)$ since the corresponding graphs are almost the same as Figs. 6 and 7.

6 Conclusions and prospects

We have discussed the approximation of FNNs from the mathematical view in this paper. First, the optimized piecewise linear activation functions representations and structures of the three types of FNNs with fixed weights are constructed and discussed completely. Second, the ideal upper bound, lower bound and essential order of approximation precision of these FNNs for continuous function defined on bounded intervals are provided. Third, the interpolation results of these FNNs proved in this paper show that the representation errors made by these FNNs on the elements belonging to the training set are null. In other words, this implies that we can obtain the weights and the thresholds of an exact neural approximation without train. Finally, we also demonstrate some numerical results of examples that show the effectiveness of the method used in this paper. Our conclusions not only further characterize the intrinsic property of approximation of these FNNs, but also reveal the implicit relationship among the precision of approximation, the number of hidden units and the smoothness of the target function.

We wrap up this paper with the following prospects:

(a) Although we give the essential approximation order of the constructed three kinds of FNNs, this is only for the Lipschitzian function class of approximated f. This means that the essential approximation order of the three FNNs for other function class of approximated f are worth to further study.

(b) It is interesting and significant to extend the main theories in this paper to multivariate functions.

Clearly solving these two problems is not easy, but it is very important and valuable.

Acknowledgements Funded by “Major Innovation Projects for Building First-calss Universities in China’s Western Region” ZKZD2017009.

Compliance with ethical standards

Conflict of interest I declare that I have no financial and personal relationships with other people or organizations that can inappropriately influence my work; there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. Cybenko G (1989) Approximation by superpositions of sigmoidal function. Math Control Signals Syst 2:303–314
2. Chen TP (1994) Approximation problems in system identification with neural networks. Sci China Ser A 24:1–7

3. Ito Y (1996) Nonlinearity creates linear independence. Adv Comput Math 5:189–203
4. Barhen J, Cogswell R, Protopopescu V (2000) Single iteration training algorithm for multilayer feedforward neural networks. Neural Process Lett 11:113–129
5. Ito Y, Saito K (1996) Superposition of linearly independent functions and finite mappings by neural networks. Math Sci 21:27–33
6. Ganjefar S, Tofighi M (2015) Single-hidden-layer fuzzy recurrent wavelet neural network: applications to function approximation and system identification. Inf Sci 294:269–285
7. Chen TP, Chen H (1995) Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to a dynamic system. IEEE Trans Neural Netw 6:911–917
8. Yoshifusa I (1991) Approximation of functions on a compact set by finite sums of sigmoid function without scaling. Neural Netw 4:817–826
9. Sartori MA, Antsaklis PJ (1991) A simple method to derive bounds on the size and to train multilayer neural networks. IEEE Trans Neural Netw 24:467–471
10. Hahm N, Hong BI (1999) Extension of localized approximation by neural networks. Bull Austral Math Soc 59:121–131
11. Chui CK, Li X (1992) Approximation by ridge functions and neural networks with one hidden Iayer. J Approx Theory 70:131–141
12. Ito Y (1991) Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory. Neural Netw 4:385–394
13. Funahashi K (1989) On the approximate realization of continuous mapping by neural network. Neural Netw 2:183–192
14. Quéau LM, Kimiaei M, Randolph MF (2015) An analysis of global robust stability of delayed dynamical neural networks. Eng Struct 92:172–185
15. Li FJ (2011) Interpolation and convergence of Bernstein–B*é*zier coefficients. Acta Math Sin Engl Ser 27:1769–1782
16. Cao FL, Lin SB, Xu ZB (2010) Approximation capability of interpolation neural networks. Neurocomputing 74:457–460
17. Sontag ED (1992) Feedforward nets for interpolation and classification. J Comp Syst Sci 45:20–48
18. Li X (2002) Interpolation by ridge polynomials and its application in neural networks. J Comput Appl Math 144:197–209
19. Llanas B, Sainz FJ (2006) Constructive approximate interpolation by neural networks. J Comput Appl Math 188:283–308
20. Llanas B, Lantarón S (2007) Hermite interpolation by neural networks. Appl Math Comput 191:429–4398
21. Li HX, Lee ES (2003) Interpolation functions of Feedforward neural networks. Comput Math Appl 46:1861–1874
22. Costarelli D (2014) Interpolation by neural network operators activated by ramp functions. J Math Anal Appl 419:574–582
23. Stinchcombe MB (1999) Neural network approximation of continuous functionals and continuous functions on compactifications. Neural Netw 12:467–477
24. Konovalov VN, Leviatan D, Maiorov VE (2009) Approximation of Sobolev classes by polynomials and ridge functions. J Approx Theory 159:97–108
25. Anastassiou GA (2002) Univariate sigmoidal neural network approximation. J Comput Anal Appl 14:659–690
26. Anastassiou GA (2011) Multivariate sigmoidal neural network approximation. Neural Netw 24:378–386
27. Anastassiou GA (2011) Multivariate hyperbolic tangent neural network approximation. Comput Math Appl 61:809–821
28. Cao F, Chen Z (2009) The approximation operators with sigmoidal functions. Comput Math Appl 58:758–765
29. Cao F, Chen Z (2012) The construction and approximation of a class of neural networks operators with ramp functions. J Comput Anal Appl 14:101–112
30. Cheang GHL (2010) Approximation with neural networks activated by ramp sigmoids. J Approx Theory 162:1450–1465
31. Pinkus A (1999) Approximation theory of the MLP model in neural networks. Acta Numer 14:143–195
32. Shristava Y, Dasgupta S (1990) Neural networks for exact matching of functions on a discrete domain. In: Proceedings of the 29th IEEE conference on decision and control. Honolulu, pp 1719–1724
33. Huang GB, Babri HA (1998) Feedforward neural networks with arbitrary bounded nonlinear activation functions. IEEE Trans Neural Netw 9:224–229
34. Costarelli D, Spigler R (2015) Approximation by series of sigmoidal functions with application to neural networks. Ann Math Puraed Appl 194:289–306
35. Anastassiou GA (1997) Rate of convergence of some neural network operators to the unit-univariate case. J Math Anal Appl 212:237–262
36. Anastassiou GA (2011) Intelligent systems: approximation by artificial neural networks. In: Intelligent systems reference library, vol 19, Springer, Berlin
37. Cao F, Chen Z (2016) Scattered data approximation by neural networks operators. Neurocomputing 190:237–242
38. Cao F, Chen Z (2015) The construction and approximation of feedforward neural network with hyperbolic tangent function. Appl Math A J Chin Univ 30:151–162
39. Cardaliaguet P, Euvrard G (1992) Approximation of a function and its derivative with a neural network. Neurak Netw 5:207–220
40. Costarelli D, Spigler R (2013) Multivariate neural network operators with sigmoidal activation functions. Neural Netw 48:72–77
41. Costarelli D, Spigler R (2014) Convergence of a family of neural network operators of the Kantorovich type. J Approx Theory 185:80–90
42. Tamura S, Tateishi M (1997) Capabilities of a four-layered feedforward neural network. IEEE Trans Neural Netw 8:251–255
43. Ismailov VE (2012) Approximation by neural networks with weights varying on a finite set of directions. J Math Anal Appl 389:72–83
44. Nageswara SVR (1999) Simple sample bound for feedforward sigmoid networks with bounded weights. Neuralcomputing 29:115–122
45. Hahm N, Hong BI (2004) An approximation by neural networks with a fixed weight. Comput Math Appl 47:1897–1903
46. Xu ZB, Cao FL (2004) The essential order of approximation for neural networks. Sci China 47:97–112
47. Xu ZB, Cao FL (2005) Simultaneous Lp-approximation order for neural networks. Neural Netw 18:914–923
48. Xu ZB, Wang JJ (2006) The essential order of approximation for nearly exponential type neural networks. Sci China 49:446–460
49. Lin SB, Cao FL, Xu ZB (2011) Essential rate for approximation by spherical neural networks. Neural Netw 24:752–758
50. Wang JJ, Xu ZB (2010) New study on neural networks: the essential order of approximation. Neural Netw 23:618–624
51. Li FJ, Xu ZB (2008) The estimation of simultaneous approximation order for neural networks. Chaos, Solitons Fractals 36:572–580
52. Li FJ, Xu ZB, Zhou YT (2008) The essential order of approximation for Suzuki's neural networks. Neurocomputing 71:3525–3533
53. Li FJ, Xu ZB (2007) The essential order of simultaneous approximation for neural networks. Appl Math Comput 194:120–127
54. Butzer PL, Nessel RJ (1971) Fourier analysis and approximation. Pure Appl Math. Academic Press, New York
55. Ditzian Z, Totik V (1987) Moduli of smoothness SSCM. Springer, Berlin

56. Suzuki S (1998) Constructive function approximation by three layer artificial neural networks. Neural Netw 11:1049–1058
57. Chui CK, Li X, Mhaskar HN (2006) Limitations of the approximation capabilities of neural networks with one hidden layer. Adv Comput Math 5:233–243
58. Ismailov VE (2014) On the approximation by neural networks with bounded number of neurons in hidden layers. J Math Anal Appl 417:963–969
59. Arteaga C, Marrero I (2015) Wiener's tauberian theorems for the FourierCBessel transformation and uniform approximation by RBF networks of Delsarte translates. J Math Anal Appl 431:482–493
60. Costarelli D, Spigler R (2013) Approximation results for neural network operators activated by sigmoidal functions. Neural Netw 44:101–106
61. Yang S, Ting TO, Man KL, Gua SU (2013) Investigation of neural networks for function approximation. Proc Comput Sci 17:586–594
62. Anastassiou GA (2012) Fractional neural network approximation. Comput Math Appl 64:1655–1676
63. Bordignon F, Gomide F (2014) Uninorm based evolving neural networks and approximation capabilities. Neurocomputing 127:13–20
64. Vukovi N, Miljkovi Z (2013) A growing and pruning sequential learning algorithm of hyper basis function neural network for function approximation. Neural Netw 46:210–226
65. Costarelli D (2015) Neural network operators: Constructive interpolation of multivariate functions. Neural Netw 67:28–36
66. Anastassiou GA (2011) Univariate hyperbolic tangent neural network approximation. Math Comput Modell 53:1111–1132
67. Ditzian Z (1989) Best polynomial approximation and Bernstein polynomial approximation on a simplex. Proc Kon Nederl Akad Wetensch 92:243–256
68. Berens H, Xu Y (1991) K-moduli, moduli of smoothness, and Bernstein polynomials on a simplex. Indag Mathem NS 4:411–421
69. Phillips GM (2003) Interpolation and approximation by polynomials. Springer, New York