

Interpolation and rates of convergence for a class of neural networks [☆]

Feilong Cao ^{a,*}, Yongquan Zhang ^a, Ze-Rong He ^b

^a Department of Information and Mathematics Sciences, China Jiliang University, Hangzhou 310018, Zhejiang Province, PR China

^b Institute of Operational Research and Cybernetics, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang Province, PR China

Received 2 February 2007; received in revised form 30 January 2008; accepted 1 February 2008

Available online 26 February 2008

Abstract

This paper presents a type of feedforward neural networks (FNNs), which can be used to approximately interpolate, with arbitrary precision, any set of distinct data in multidimensional Euclidean spaces. They can also uniformly approximate any continuous functions of one variable or two variables. By using the modulus of continuity of function as metric, the rates of convergence of approximate interpolation networks are estimated, and two Jackson-type inequalities are established.

© 2008 Elsevier Inc. All rights reserved.

Keywords: Neural networks; Interpolation; Approximation; Estimate of error

1. Background and motivation

Throughout this paper, we use the following definitions and notations, in which \mathbb{N} and \mathbb{R} denote the natural numbers and the set of real numbers, respectively, and for any positive integer d , \mathbb{R}^d denotes the d dimensional Euclidean space. Let $S = \{x_0, x_1, \dots, x_n\} \subset \mathbb{R}^d$ denote a set of distinct vectors, and $\{f_0, f_1, \dots, f_n\} \subset \mathbb{R}$ a set of real numbers. Then

$$\{(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)\}, \quad (1)$$

is called a set of interpolation sample, and $\{x_i\}_{i=0}^n$ is called a node system of interpolation.

If there exists a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that

$$f(x_i) = f_i, \quad i = 0, 1, \dots, n,$$

then we say that the function f is an exact interpolation of sample set (1). If there exists a function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ such that

[☆] Supported by the National Natural Science Foundation of China (Nos. 60473034 and 10771048).

* Corresponding author. Fax: +86 571 86835737.

E-mail addresses: fcao@263.net (F. Cao), zrhe@hdu.edu.cn (Z.-R. He).

$$|g(x_i) - f_i| < \varepsilon, \quad i = 0, 1, \dots, n,$$

for positive real number ε , then we call g an ε -approximate interpolation of sample set (1). According to common usage, a sigmoidal function σ is defined in \mathbb{R} with the following properties:

$$\lim_{t \rightarrow +\infty} \sigma(t) = 1 \quad \text{and} \quad \lim_{t \rightarrow -\infty} \sigma(t) = 0. \quad (2)$$

Let $\phi_i : \mathbb{R}^d \rightarrow \mathbb{R}, i = 0, 1, \dots, n$, be a family of real function, then we define

$$N(x) = \sum_{i=0}^n c_i \phi_i(x), \quad c_i \in \mathbb{R},$$

and the set

$$\mathcal{N}_{n+1, \phi}^d = \left\{ N(x) : N(x) = \sum_{i=0}^n c_i \phi_i(x), \quad c_i \in \mathbb{R} \right\}. \quad (3)$$

Clearly, $N(x)$ can be understood as a model of FNNs. For approximation and interpolation of FNNs, there exist three basic problems as follows.

Problem 1 (Density). Let D be a compact subset of \mathbb{R}^d and $\mathcal{N}_{n+1, \phi}^d$ the subspace of $C(D)$ defined by (3). Is $\mathcal{N}_{n+1, \phi}^d$ dense in $C(D)$?

Problem 2 (Interpolation). Suppose $x_0, x_1, \dots, x_n \in \mathbb{R}^d$ and data f_0, f_1, \dots, f_n are prescribed. Does there exist a unique set of scalars c_0, c_1, \dots, c_n such that $\sum_{i=0}^n c_i \phi_i(x_j) = f_j, j = 0, 1, \dots, n$?

Problem 3 (Convergence rates). When the answer to Problem 1 is affirmative, we would like some estimations of the convergence rate ϵ

$$\epsilon = |f(x) - N(x)|, \quad f \in C(D), \quad N(x) \in \mathcal{N}_{n+1, \phi}^d.$$

We shall focus on the Problems 2 and 3 for a class of FNNs in \mathbb{R}^d in this paper.

Neural computation research, together with related areas in approximation theory, has developed powerful methods for approximating continuous and integrable functions on compact subsets of \mathbb{R}^d since 1980s. Most approximation schemes using three layered FNNs have been studied (e.g. [1–17]). In such schemes, function approximation capabilities critically depend on the activation function nature of the hidden layer.

On the other hand, the interpolation problem of neural networks, as a very active direction, has drawn great attention. Several proofs on the fact that single hidden layer FNNs with at most $n + 1$ neurons can learn $n + 1$ distinct samples $(x_i, f_i) (i = 0, 1, 2, \dots, n)$ with zero error (exact interpolation) have been proposed in [18–21]. Ito and Saito [19] proved that if the activation function is continuous and nondecreasing sigmoidal function, then the interpolation can be made with inner weights $w_j \in \mathbb{S}^{d-1}$, where \mathbb{S}^{d-1} is the unit sphere in \mathbb{R}^d . In [20] Pinkus proved the same result but ϕ only needs to be continuous in \mathbb{R} and not a polynomial. Shrivatava and Dasgupta [27] gave a proof for sigmoidal activation function $\phi(x) = \frac{1}{1+e^{-x}}$. However, it is more difficult to solve the exact interpolation networks. So, ones turn to the study of the approximate interpolation neural networks, which first were used in [21] as a tool to study the exact interpolation networks. It was proved in [21] that if arbitrary precision approximate interpolation exists in a linear space of functions, then an exact interpolation can be obtained in that space. Furthermore, the fact “If ϕ is sigmoidal, continuous and there exists a point c such that $\phi'(c) \neq 0$, then an interpolation problem with $2n + 1$ samples can be approximated with arbitrary precision by a net with $n + 1$ neurons” was given. Recently, Llanas and Sainz [22] studied the existence and construction of ε -approximate interpolation networks. They first considered that the activation function ϕ is a nondecreasing sigmoidal function satisfying the condition (2) and gave a new and quantitative proof of the fact that $n + 1$ hidden neurons can learn $n + 1$ distinct samples with zero error. Then, they introduced the approximate interpolation networks, which do not require training and can approximately interpolate an arbitrary set of distinct samples. However, it is natural to raise the following two questions:

- (1) Can we replace the sigmoidal nondecreasing function used in [22] by other activation functions?
- (2) Can we estimate the errors of approximation for constructed networks?

The main purpose of this paper is to give an affirmative answer to these questions. We first introduce a class of activation function $g_j : \mathbb{R}^d \rightarrow \mathbb{R}$, defined by

$$g_j(x) = g_j(x, A) = \frac{e^{-A\rho(x, x_j)}}{\sum_{i=0}^n e^{-A\rho(x, x_i)}}, \quad j = 0, 1, \dots, n,$$

where x_0, x_1, \dots, x_n are the data in \mathbb{R}^d , $\rho(a, b) = \|a - b\|_2$ denotes the Euclidean distance between the points a and b in \mathbb{R}^d , and $A > 0$ is parameter. Furthermore, we define the linear combination of $g_j(x, A)$ as

$$N(x) = \sum_{j=0}^n c_j g_j(x, A).$$

Clearly, $N(x)$ can be understood to be a FNN with four layers: the first layer is the input layer, the input is $x(x \in \mathbb{R}^d)$; the second layer is processing layer for computing values $\rho(x, x_i), i = 0, 1, \dots, n$, between input x and the prototypical input points x_i , and it is as the input of the third layer that contains $n + 1$ neurons, $g_j(x, A)$ is activation function of the j th neuron; the fourth layer is output layer, the output is $N(x)$.

In fact, the sigmoidal function $\phi(x) = \frac{1}{1+e^{-x}}$, which usually be used to activation function in the hidden layer of neural networks, is a logistic model. This model is an important one and has been widely used in biology, demography and so on (see [23,24]). Naturally, the functions

$$\Phi_j(x) = \frac{e^{f_j(x)}}{\sum_{i=0}^n e^{f_i(x)}}, \quad j = 0, 1, 2, \dots, n,$$

can be regarded as a multi-class generalization of the logistic model (see section 10.6 in [25]), which also was used to a regression model for the case of multi-class in the classification problems. Although the functions $g_j(x)$ are not sigmoidal, they possess some properties that common sigmoidal functions can not own, such as

$$0 < g_j(x) \leq 1, \quad j = 0, 1, 2, \dots; \quad \sum_{j=0}^n g_j(x) = 1.$$

On the other hand, it follows from their structures that $g_j(x)$ contain the information of the interpolation samples. The second layer of the network composed of $g_j(x)$ can be regarded as the processing layer and the input of the third layer, which is more convenient to the study of network interpolations. These form our main motivation to introduce functions $g_j(x)$ as activation functions in hidden layer of networks.

This paper is organized as follows. In Section 2 we state the main result of this paper. Section 3 gives a new constructive proof for exact interpolation networks. In Section 4 we introduce a type of approximate networks and estimate the error between the approximate networks and the exact interpolation networks. Section 5 provides the estimates of convergence rates of the approximate networks approximating continuous functions of one variable or two variables. Two numerical examples for illustrating the theoretical results are given in Section 6, and conclusions are drawn in the final section.

2. The main results

We aim to construct exact and approximate interpolation networks by using a class of activation function, and to give some estimations of convergence rate ϵ .

The first result, Theorem 1 in Section 3, shows that there exists unique exact interpolation network with activation function $g_j(x)$ for a given sample. It is clear that our proof of the existence is simpler than previous ones of networks consist of sigmoidal functions (see [22,18–20,27,21]). Introduction of activation functions $g_j(x)$ is partially motivated by the considerations above. Meanwhile, the first question has the affirmative answer.

The second main result is on the estimations of convergence rate of approximate interpolation networks, which are discussed in Section 5. The approximate interpolation network defined in the article (see Section

4) is actually a linear combination of $g_j(x)$, which can be treated as a positive linear operator from $C(K)$ to $C(K)$. This situation is helpful to the investigation of approximation properties of the approximate interpolation network. By using the modulus of continuity as metric, two Jackson-type estimations are established. Thus, the answer for the second question raised in Section 1 is “yes”. These quantitative estimations of approximation of interpolation networks should be a new study. Most of existing studies on interpolation networks, such as [22,18–20,27,21], did not relate to the estimates of convergence rate. Up to now, we have not found results on estimations of convergence rates of interpolation network yet. However, from the viewpoint of application, the quantitative estimation of convergence rate is more important.

The obtained estimations in Section 5, which reveal, to some extent, the relation between the convergence rates and the topological structure of the networks. These also construct a basic to study the complexity of interpolation networks approximation.

3. Construction of interpolation networks

For $S = \{x_0, x_1, \dots, x_n\} \subset \mathbb{R}^d, A > 0$, we define

$$g_j(x, A) = \frac{e^{-A\rho(x, x_j)}}{\sum_{i=0}^n e^{-A\rho(x, x_i)}}, \quad j = 0, 1, \dots, n, \quad x \in \mathbb{R}^d.$$

Let K be a compact subset of \mathbb{R}^d , and $S = \{x_0, x_1, \dots, x_n\} \subset K$ a set of distinct vectors. For the sample (1), we shall find a function $N(x)$ in

$$\mathcal{N}_{n+1, g}^d = \left\{ N(x) : N(x) = \sum_{j=0}^n c_j g_j(x, A), c_j \in \mathbb{R} \right\},$$

such that

$$N(x_i) = f_i, \quad i = 0, 1, \dots, n, \tag{4}$$

that is

$$\sum_{j=0}^n c_j g_j(x_i, A) = f_i, \quad i = 0, 1, \dots, n,$$

or, in vectorial form

$$\mathbf{MC} = \mathbf{Y}, \tag{5}$$

where

$$\mathbf{M} = \begin{bmatrix} g_0(x_0, A) & g_1(x_0, A) & \cdots & g_n(x_0, A) \\ g_0(x_1, A) & g_1(x_1, A) & \cdots & g_n(x_1, A) \\ \vdots & \vdots & \cdots & \vdots \\ g_0(x_n, A) & g_1(x_n, A) & \cdots & g_n(x_n, A) \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + \sum_{i=1}^n e^{-A\rho(x_0, x_i)}} & \frac{e^{-A\rho(x_0, x_1)}}{1 + \sum_{i=1}^n e^{-A\rho(x_0, x_i)}} & \cdots & \frac{e^{-A\rho(x_0, x_n)}}{1 + \sum_{i=1}^n e^{-A\rho(x_0, x_i)}} \\ \frac{e^{-A\rho(x_1, x_0)}}{1 + \sum_{i=0, i \neq 1}^n e^{-A\rho(x_1, x_i)}} & \frac{1}{1 + \sum_{i=0, i \neq 1}^n e^{-A\rho(x_1, x_i)}} & \cdots & \frac{e^{-A\rho(x_1, x_n)}}{1 + \sum_{i=0, i \neq 1}^n e^{-A\rho(x_1, x_i)}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{e^{-A\rho(x_n, x_0)}}{1 + \sum_{i=0}^{n-1} e^{-A\rho(x_n, x_i)}} & \frac{e^{-A\rho(x_n, x_1)}}{1 + \sum_{i=0}^{n-1} e^{-A\rho(x_n, x_i)}} & \cdots & \frac{1}{1 + \sum_{i=0}^{n-1} e^{-A\rho(x_n, x_i)}} \end{bmatrix},$$

and

$$\mathbf{C} = [c_0, c_1, \dots, c_n]^T, \quad \mathbf{Y} = [f_0, f_1, \dots, f_n]^T.$$

If the n degree square matrix $\mathbf{S} = (s_{ij})_{i,j=1}^n$ satisfies

$$|s_{ii}| > \sum_{j=1, j \neq i}^n |s_{ij}|, \quad i = 1, 2, \dots, n,$$

then \mathbf{S} is said to be strictly diagonally dominant. Clearly, strictly diagonally dominant matrices are invertible.

We can now state

Theorem 1. *There exists a real number $A^* > 0$, such that if $A > A^*$, then the matrix \mathbf{M} is invertible.*

Proof. Noting that the elements of \mathbf{M} are all positive and the sum of each row is equal to 1, to show the matrix \mathbf{M} is strictly diagonally dominant, we only need to prove

$$g_{jj} = \frac{1}{1 + \sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)}} > \frac{1}{2}, \quad j = 0, 1, \dots, n,$$

i.e.

$$\sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)} < 1.$$

Let

$$t = \min_{i \neq j} \{\rho(x_i, x_j)\}, \quad i, j = 0, 1, \dots, n,$$

since $S = \{x_0, x_1, \dots, x_n\}$ is a set of distinct vectors in \mathbb{R}^d , we have $t > 0$, and

$$\sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)} \leq ne^{-At}.$$

So, putting $A^* = \frac{\ln n}{t}$, we get

$$\sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)} \leq ne^{-At} < 1,$$

for $A > A^*$. Thus, the matrix \mathbf{M} is invertible. The proof of [Theorem 1](#) is completed.

From [Theorem 1](#) and Grama’s rule, we see that the solution of (5) for $(c_0, c_1, \dots, c_n)^T$ is unique. Thus, for the activation function g_j there exists an exact interpolation neural network with the condition (4). \square

4. The relation between approximate interpolation networks and exact interpolation networks

Suppose that $S = \{x_0, x_1, \dots, x_n\} \subset K$, where K is a compact subset of \mathbb{R}^d , and f is continuous function defined on K . Let $f(x_i) = f_i, i = 0, 1, \dots, n$, for the sample of interpolation (1), we define

$$N_a(x) = \sum_{j=0}^n f_j g_j(x, A).$$

This neural network will be called an approximate interpolation network for the interpolation sample set

$$\{(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)\}.$$

We shall prove in this section that the network is arbitrarily near the corresponding exact interpolation network.

Let us now introduce some previous definitions and a lemma used in [26].

In \mathbb{C}^{n+1} we introduce the norm

$$\|\mathbf{Z}\|_1 = |z_0| + |z_1| + \dots + |z_n|,$$

where $\mathbf{Z} = (z_0, z_1, \dots, z_n), |z| = \sqrt{z\bar{z}}$ (\bar{z} is the conjugate of complex number z).

In the space of complex $(n + 1) \times (n + 1)$ matrices the induced matrix norm is

$$\|\mathbf{B}\|_1 = \max_{i=0,1,\dots,n} \left(\sum_{j=0}^n |b_{ij}| \right).$$

Lemma 1 (see [26] or [22]). Let \mathbf{B} and $\mathbf{B} + \Delta\mathbf{B}$ be complex, invertible matrices. Let \mathbf{Z} , and $\mathbf{Z} + \Delta\mathbf{Z}$ be vectors such that

$$\mathbf{B}\mathbf{Z} = \mathbf{b}, \quad (\mathbf{B} + \Delta\mathbf{B})(\mathbf{Z} + \Delta\mathbf{Z}) = \mathbf{b},$$

then

$$\frac{\|\Delta\mathbf{Z}\|_1}{\|\mathbf{Z}\|_1} \leq \text{Cond}_1(\mathbf{B}) \frac{\|\Delta\mathbf{B}\|_1}{\|\mathbf{B}\|_1} \left(\frac{1}{1 - \text{Cond}_1(\mathbf{B}) \frac{\|\Delta\mathbf{B}\|_1}{\|\mathbf{B}\|_1}} \right),$$

where $\text{Cond}_1(\mathbf{B}) = \|\mathbf{B}\|_1 \|\mathbf{B}^{-1}\|_1$ is the condition number of \mathbf{B} .

Theorem 2. For the sample set of interpolation (1), and $t = \min_{i \neq j, 0 \leq i, j \leq n} \{\rho(x_i, x_j)\}$, there exists a positive real number A_1 , such that if $A > A_1$, then

$$|N_e(x, A) - N_a(x, A)| \leq \frac{2ne^{-At}}{1 - 2ne^{-At}} \sum_{j=0}^n |f_j|,$$

for all $x \in K$.

Proof. We shall consider that $A > A^*$ ($A^* = \frac{\ln n}{t}$ given in the last section).

The exact interpolation networks can be written as

$$N_e(x, A) = \sum_{j=0}^n c_j^* g_j(x, A),$$

and the approximate interpolation networks can be written as

$$N_a(x, A) = \sum_{j=0}^n c_j g_j(x, A).$$

The coefficients c_j^* and c_j are, respectively, solutions of the systems

$$\mathbf{M}\mathbf{C}^* = \mathbf{f} \quad \text{and} \quad \mathbf{U}\mathbf{C} = \mathbf{f},$$

where

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix},$$

and

$$\mathbf{C}^* = [c_0^*, c_1^*, \dots, c_n^*]^T, \quad \mathbf{C} = [c_0, c_1, \dots, c_n]^T, \quad \mathbf{f} = [f_0, f_1, \dots, f_n]^T.$$

Let $\mathbf{B} = \mathbf{U}$, $\mathbf{Z} = \mathbf{C}$, $\mathbf{M} = \mathbf{B} + \Delta\mathbf{B}$, $\mathbf{Z} + \Delta\mathbf{Z} = \mathbf{C}^*$, $\mathbf{b} = \mathbf{f}$, we then obtain from Lemma 1 that

$$\frac{\|\mathbf{C}^* - \mathbf{C}\|_1}{\|\mathbf{C}\|_1} \leq \frac{\|\mathbf{U}^{-1}\|_1 \|\mathbf{M} - \mathbf{U}\|_1}{1 - \|\mathbf{U}^{-1}\|_1 \|\mathbf{M} - \mathbf{U}\|_1}.$$

So, we only need to estimate $\|\mathbf{U}^{-1}\|_1 \|\mathbf{M} - \mathbf{U}\|_1$. Noting the expression of every element in matrix $\mathbf{M} = (m_{ij})_{i,j=0}^n$ and $t = \min_{i \neq j, 0 \leq i, j \leq n} \{\rho(x_i, x_j)\} > 0$, we have for $A > A^*$,

$$|m_{jj} - 1| = |g_{jj} - 1| = \left| \frac{1}{1 + \sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)}} - 1 \right| = \frac{\sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)}}{1 + \sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)}} \leq \sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)} \leq ne^{-At},$$

$j = 0, 1, \dots, n$.

For $i \neq j$, we have

$$|m_{ij}| = \frac{e^{-A\rho(x_j, x_i)}}{1 + \sum_{i=0, i \neq j}^n e^{-A\rho(x_j, x_i)}} \leq e^{-A\rho(x_j, x_i)} \leq e^{-At}.$$

So

$$\|\mathbf{M} - \mathbf{U}\|_1 \leq ne^{-At} + ne^{-At} = 2ne^{-At}.$$

Recalling $\|\mathbf{U}^{-1}\|_1 = 1$, we thus obtain

$$\|\mathbf{U}^{-1}\|_1 \|\mathbf{M} - \mathbf{U}\|_1 \leq 2ne^{-At}.$$

There exists a number A' such that if $A > A'$ then $2ne^{-At} < 1$. Considering that the function $g(x) = \frac{x}{1-x}$ is strictly increasing on $(-\infty, 1)$, one has

$$\|\mathbf{C} - \mathbf{C}^*\|_1 \leq \frac{2ne^{-At}}{1 - 2ne^{-At}} \|\mathbf{C}\|_1 = \frac{2ne^{-At}}{1 - 2ne^{-At}} \|\mathbf{f}\|_1.$$

On the other hand

$$|N_e(x, A) - N_a(x, A)| = \left| \sum_{j=0}^n (c_j^* - c_j) g_j(x, A) \right| \leq \sum_{j=0}^n |c_j^* - c_j| = \|\mathbf{C}^* - \mathbf{C}\|_1.$$

Finally, taking $A > A_1 = \max\{A', A^*\}$, we obtain

$$|N_e(x, A) - N_a(x, A)| \leq \frac{2ne^{-At}}{1 - 2ne^{-At}} \sum_{j=0}^n |f_j|.$$

The proof of [Theorem 2](#) is complete. \square

5. Uniform approximation by approximate interpolation networks

We first consider one dimension case. Let f be a continuous function defined on $[a, b]$. The modulus of continuity of function f on $[a, b]$ is defined for $t > 0$ by

$$\omega(f, t) = \sup_{|h| \leq t} \max_{x, x+h \in [a, b]} |f(x+h) - f(x)|.$$

The modulus of continuity is usually considered as the measure of the smoothness of function and the approximation error in approximation theory. The function f is called Lipschitz α ($0 < \alpha \leq 1$) continuous and is written as $f \in \text{Lip}_{C(f)}(\alpha)$, if there exists a constant $C(f)$ such that $\omega(f, \delta) \leq C(f)\delta^\alpha$, where $C(f)$ denotes the positive constant depending only on f .

Divide $[a, b]$ into n equal segments, each has length of $\frac{b-a}{n}$ and let $a = x_0 < x_1 < x_2 \cdots < x_n = b$. Set

$$f_j = f(x_j) = f\left(a + \frac{b-a}{n}j\right),$$

and $A = A(n)$ depend on n . The approximate interpolation networks are defined by

$$N_a(x, A) = N_a(x, A(n)) = \sum_{j=0}^n f_j \frac{e^{-A(n)|x-x_j|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}}.$$

We can now state

Theorem 3. *Let f be continuous on $[a, b]$. Then there exists $A^* > 0$, such that*

$$|f(x) - N_a(x, A(n))| \leq 2\omega\left(f, \frac{b-a}{n}\right) + 2Mne^{-n},$$

for $A(n) > A^*$ and for all $x \in [a, b]$, where $M = \max_{x \in [a, b]} |f(x)|$.

Proof. For $x \in [a, b]$, there exists $J \in \mathbb{N}, 0 \leq J \leq n - 1$, such that $x \in [x_J, x_{J+1}]$, then

$$\begin{aligned} |f(x) - N_a(x, A(n))| &= \left| \sum_{j=0}^n f(x) \frac{e^{-A(n)|x-x_j|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} - \sum_{j=0}^n f_j \frac{e^{-A(n)|x-x_j|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} \right| = \left| \sum_{j=0}^n (f(x) - f_j) \frac{e^{-A(n)|x-x_j|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} \right| \\ &\leq |f(x) - f_J| \frac{e^{-A(n)|x-x_J|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} + |f(x) - f_{J+1}| \frac{e^{-A(n)|x-x_{J+1}|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} \\ &\quad + \sum_{j=0}^{J-1} |f(x) - f_j| \frac{e^{-A(n)|x-x_j|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} + \sum_{j=J+2}^n |f(x) - f_j| \frac{e^{-A(n)|x-x_j|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} \\ &= I_1 + I_2 + I_3 + I_4. \end{aligned}$$

Clearly, we have

$$I_1 \leq \omega\left(f, \frac{b-a}{n}\right) \frac{e^{-A(n)|x-x_J|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} \leq \omega\left(f, \frac{b-a}{n}\right),$$

and

$$I_2 \leq \omega\left(f, \frac{b-a}{n}\right).$$

For $x \in [x_J, x_{J+1}]$ and all $j = 0, 1, 2, \dots, J - 1$, there holds $|x - x_j| - |x - x_J| \geq \frac{b-a}{n}$. Also, $|x - x_j| - |x - x_{J+1}| \geq \frac{b-a}{n}$ is valid for all $j = J + 2, J + 3, \dots, n$, and $x \in [x_J, x_{J+1}]$. Hence

$$I_3 \leq 2M \sum_{j=0}^{J-1} \frac{e^{-A(n)|x-x_j|}}{\sum_{i=0}^n e^{-A(n)|x-x_i|}} \leq 2M \sum_{j=0}^{J-1} e^{-A(n)(|x-x_j|-|x-x_J|)} \leq 2MJ e^{-A(n)\frac{b-a}{n}},$$

and

$$I_4 \leq 2M \sum_{j=J+2}^n e^{-A(n)(|x-x_j|-|x-x_{J+1}|)} \leq 2M(n - J - 1)e^{-A(n)\frac{b-a}{n}}.$$

Letting $A^* = \frac{n^2}{b-a}$, one has for $A(n) > A^*$

$$I_3 \leq 2MJ e^{-n} \quad \text{and} \quad I_4 \leq 2M(n - J - 1)e^{-n}.$$

Combining with the estimates of I_1, I_2, I_3 , and I_4 , we have

$$|f(x) - N_a(x, A(n))| \leq 2\omega\left(f, \frac{b-a}{n}\right) + 2Mne^{-n}.$$

The proof of Theorem 3 is completed. \square

In the following, we shall discuss two dimensions case. Let f be continuous on $[a, b] \times [c, d]$. The modulus of continuity of f is defined by

$$\omega(f, t) = \sup_{\|H\|_2 \leq t} \max_{X, X+H \in [a,b] \times [c,d]} |f(X+H) - f(X)|.$$

Consider uniform partitions $\{x_0, x_1, \dots, x_n\}$ of $[a, b]$ and $\{y_0, y_1, \dots, y_n\}$ of $[c, d]$, respectively, that is, $x_j = a + \frac{b-a}{n}j$ and $y_j = c + \frac{d-c}{n}j, j = 0, 1, \dots, n$. The vectors $X_i, i = 0, 1, \dots, (n+1)^2 - 1$, and the corresponding values $f(X_i)$ are defined by

$$\begin{aligned} f(X_0) &= f(x_0, y_0) & f(X_1) &= f(x_1, y_0) & \cdots & f(X_n) &= f(x_n, y_0), \\ f(X_{n+1}) &= f(x_0, y_1) & f(X_{n+2}) &= f(x_1, y_1) & \cdots & f(X_{2n+1}) &= f(x_n, y_1), \\ &\vdots & &\vdots & & &\vdots \\ f(X_{n^2+n+1}) &= f(x_0, y_n) & f(X_{n^2+n+2}) &= f(x_1, y_n) & \cdots & f(X_{(n+1)^2-1}) &= f(x_n, y_n). \end{aligned}$$

Let $A = A(n)$ depend on n , we can define the approximate interpolation networks as

$$N_a(X, A(n)) = \sum_{j=0}^{(n+1)^2-1} f(X_j) \frac{e^{-A(n)\|X-X_j\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}}.$$

We can now state

Theorem 4. *Let f be continuous on $[a, b] \times [c, d]$. Then there exists A^* such that*

$$|f(X) - N_a(X, A(n))| \leq 4\omega\left(f, \frac{\sqrt{(b-a)^2 + (d-c)^2}}{n}\right) + 2M((n+1)^2 - 4)e^{-n},$$

for $A(n) > A^*$ and for all $X \in [a, b] \times [c, d]$, where $M = \max_{X \in [a,b] \times [c,d]} |f(X)|$.

Proof. For $X \in [a, b] \times [c, d]$, from the partitions we can find a small rectangle Δ in $[a, b] \times [c, d]$, such that $X \in \Delta$. Namely, there exists a $J \in \mathbb{N}$, such that the four vertexes of Δ are X_J, X_{J+1}, X_{J+n} , and X_{J+n+1} . So

$$\begin{aligned} \left| f(X) - \sum_{j=0}^{(n+1)^2-1} f(X_j) \frac{e^{-A(n)\|X-X_j\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}} \right| &\leq \sum_{j=0}^{(n+1)^2-1} |f(X) - f(X_j)| \frac{e^{-A(n)\|X-X_j\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}} \\ &= |f(X) - f(X_J)| \frac{e^{-A(n)\|X-X_J\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}} \\ &\quad + |f(X) - f(X_{J+1})| \frac{e^{-A(n)\|X-X_{J+1}\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}} \\ &\quad + |f(X) - f(X_{J+n})| \frac{e^{-A(n)\|X-X_{J+n}\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}} \\ &\quad + |f(X) - f(X_{J+n+1})| \frac{e^{-A(n)\|X-X_{J+n+1}\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}} \\ &\quad + \sum_{j=0, j \neq J, J+1, J+n, J+n+1}^{(n+1)^2-1} |f(X) - f(X_j)| \frac{e^{-A(n)\|X-X_j\|_2}}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)\|X-X_i\|_2}} \\ &= I_1 + I_2 + I_3 + I_4 + I_5. \end{aligned}$$

It is clear to see

$$\begin{aligned} I_1 &\leq \omega\left(f, \frac{\sqrt{(b-a)^2 + (d-c)^2}}{n}\right), & I_2 &\leq \omega\left(f, \frac{\sqrt{(b-a)^2 + (d-c)^2}}{n}\right), \\ I_3 &\leq \omega\left(f, \frac{\sqrt{(b-a)^2 + (d-c)^2}}{n}\right), & \text{and } I_4 &\leq \omega\left(f, \frac{\sqrt{(b-a)^2 + (d-c)^2}}{n}\right). \end{aligned}$$

For I_5 , we have

$$\begin{aligned}
 I_5 &\leq 2M \sum_{j=0, j \neq J, J+1, J+n, J+n+1}^{(n+1)^2-1} \frac{1}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)(\|X-X_i\|_2 - \|X-X_j\|_2)}} \\
 &\leq 2M \sum_{j=0}^{J-1} \frac{1}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)(\|X-X_i\|_2 - \|X-X_j\|_2)}} + 2M \sum_{i=J+2}^{J+n-1} \frac{1}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)(\|X-X_i\|_2 - \|X-X_j\|_2)}} \\
 &\quad + 2M \sum_{j=J+n+2}^{(n+1)^2-1} \frac{1}{\sum_{i=0}^{(n+1)^2-1} e^{-A(n)(\|X-X_i\|_2 - \|X-X_j\|_2)}} \\
 &\leq 2M \sum_{j=0}^{J-1} e^{-A(n)(\|X-X_j\|_2 - \|X-X_J\|_2)} + 2M \sum_{j=J+2}^{J+n-1} e^{-A(n)(\|X-X_j\|_2 - \|X-X_{J+1}\|_2)} + 2M \sum_{j=J+n+2}^{(n+1)^2-1} e^{-A(n)(\|X-X_j\|_2 - \|X-X_{J+n+1}\|_2)} \\
 &= I_5^{(1)} + I_5^{(2)} + I_5^{(3)}.
 \end{aligned}$$

Next, we only give the detail of estimating $I_5^{(1)}$ because the estimates of $I_5^{(2)}$ and $I_5^{(3)}$ are similar. For $0 \leq j \leq J - 1$, we have

$$\|X - X_j\|_2 \geq \min\{\|X - X_{J-1}\|_2, \|X - X_{J-n}\|_2, \|X - X_{J-n+1}\|_2\}.$$

Case a. If

$$\|X - X_{J-1}\|_2 = \min\{\|X - X_{J-1}\|_2, \|X - X_{J-n}\|_2, \|X - X_{J-n+1}\|_2\},$$

then

$$\begin{aligned}
 \|X - X_j\|_2 - \|X - X_J\|_2 &\geq \|X - X_{J-1}\|_2 - \|X - X_J\|_2 = \frac{\|X - X_{J-1}\|_2^2 - \|X - X_J\|_2^2}{\|X - X_{J-1}\|_2 + \|X - X_J\|_2} \\
 &\geq \frac{\left(\frac{b-a}{n}\right)^2}{\|X_{J+n+1} - X_{J-1}\|_2 + \|X_{J+n+1} - X_J\|_2} \\
 &= \frac{\left(\frac{b-a}{n}\right)^2}{\left(\left(\frac{2(b-a)}{n}\right)^2 + \left(\frac{d-c}{n}\right)^2\right)^{1/2} + \left(\left(\frac{b-a}{n}\right)^2 + \left(\frac{d-c}{n}\right)^2\right)^{1/2}} \geq \frac{\left(\frac{b-a}{n}\right)^2}{\frac{3(b-a)}{n} + \frac{2(d-c)}{n}} \\
 &= \frac{(b-a)^2}{n(3(b-a) + 2(d-c))}.
 \end{aligned}$$

Case b. If

$$\|X - X_{J-n}\|_2 = \min\{\|X - X_{J-1}\|_2, \|X - X_{J-n}\|_2, \|X - X_{J-n+1}\|_2\},$$

then we similarly obtain

$$\begin{aligned}
 \|X - X_j\|_2 - \|X - X_J\|_2 &\geq \|X - X_{J-n}\|_2 - \|X - X_J\|_2 = \frac{\|X - X_{J-n}\|_2^2 - \|X - X_J\|_2^2}{\|X - X_{J-n}\|_2 + \|X - X_J\|_2} \\
 &\geq \frac{\left(\frac{d-c}{n}\right)^2}{\|X_{J+n+1} - X_{J-n}\|_2 + \|X_{J+n+1} - X_J\|_2} \geq \frac{\left(\frac{d-c}{n}\right)^2}{\frac{3(d-c)}{n} + \frac{2(b-a)}{n}} \\
 &= \frac{(d-c)^2}{n(3(d-c) + 2(b-a))}.
 \end{aligned}$$

Case c. If

$$\|X - X_{J-n+1}\|_2 = \min\{\|X - X_{J-1}\|_2, \|X - X_{J-n}\|_2, \|X - X_{J-n+1}\|_2\},$$

then

$$\begin{aligned} \|X - X_j\|_2 - \|X - X_J\|_2 &\geq \|X - X_{J-n+1}\|_2 - \|X - X_J\|_2 = \frac{\|X - X_{J-n+1}\|_2^2 - \|X - X_J\|_2^2}{\|X - X_{J-n+1}\|_2 + \|X - X_J\|_2} \\ &\geq \frac{(d - c)^2}{n(3(d - c) + 2(b - a))}. \end{aligned}$$

Combining with three cases above and letting

$$\mathcal{D}(n, a, b, c, d) = \min \left\{ \frac{(b - a)^2}{n(3(b - a) + 2(d - c))}, \frac{(d - c)^2}{n(3(d - c) + 2(b - a))} \right\},$$

we have for $0 \leq j \leq J - 1$

$$\|X - X_j\|_2 - \|X - X_J\|_2 \geq \mathcal{D}(n, a, b, c, d).$$

Similarly, we have for $J + n < j \leq (n + 1)^2 - 1$

$$\|X - X_j\|_2 - \|X - X_{J+n}\|_2 \geq \mathcal{D}(n, a, b, c, d).$$

For $J + 2 \leq j \leq J + n - 1$, noting that

$$\|X - X_j\|_2 \geq \min\{\|X - X_{J+2}\|_2, \|X - X_{J+n-1}\|_2\},$$

we also have

$$\|X - X_j\|_2 - \|X - X_{J+1}\|_2 \geq \frac{(b - a)^2}{n(3(b - a) + 2(d - c))} \geq \mathcal{D}(n, a, b, c, d).$$

Table 1
Approximation error for target function $\sin x$

n	$\sup_{x \in [0, \pi]} N_n(x) - \sin x $	$\frac{2\pi}{n} + 2ne^{-n}$
5	0.1415	1.7173
10	0.1023	0.7003
30	0.0446	0.2163
50	0.0279	0.1282

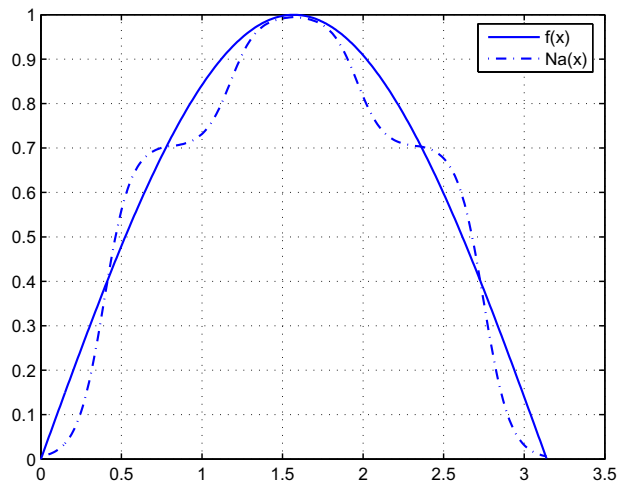


Fig. 1. $n = 5$ case.

Therefore, taking $A^* = \frac{n}{\mathcal{D}(n,a,b,c,d)}$, we obtain for $A(n) > A^*$

$$I_5^{(1)} \leq 2MJe^{-n}, \quad I_5^{(2)} \leq 2M(n-2)e^{-n}, \quad \text{and} \quad I_5^{(3)} \leq 2M((n+1)^2 - J - n - 2)e^{-n},$$

which implies $I_5 \leq 2M((n+1)^2 - 4)e^{-n}$. Combining with the estimates of $I_i (i = 1, 2, 3, 4, 5)$, one has

$$|f(X) - N_a(X, A(n))| \leq 4\omega \left(f, \frac{\sqrt{(b-a)^2 + (d-c)^2}}{n} \right) + 2M((n+1)^2 - 4)e^{-n}.$$

This completes the proof. \square

6. Numerical examples

In this section, we present some numerical experiments to demonstrate the validity of the obtained results and suggest the error bound of neural network approximation. All computations were done in Matlab.

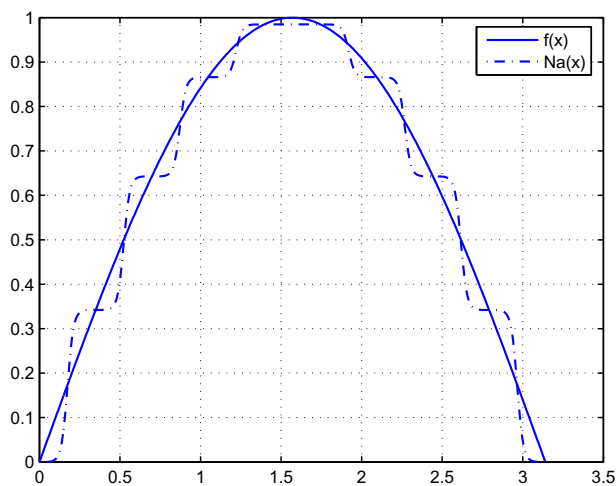


Fig. 2. $n = 10$ case.

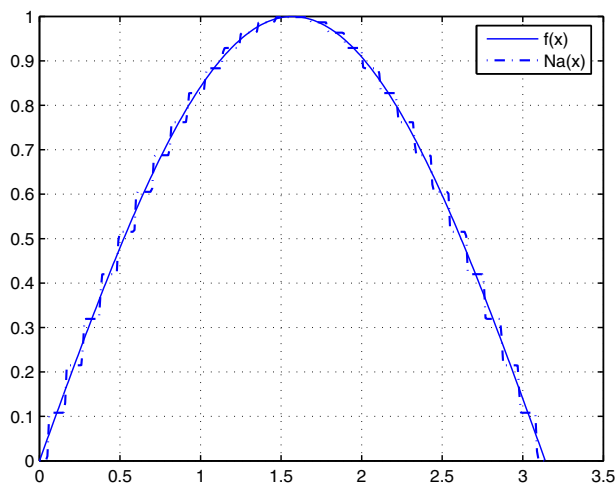


Fig. 3. $n = 30$ case.

First, we select two functions $f_1(x) = \sin x, x \in [0, \pi]$ and $f_2(x) = \sqrt{x}, x \in [0, 5]$ as the target functions and investigate the approximate interpolation networks with activation function $g_j(x)$ approximation to $f_1(x)$ and $f_2(x)$ over the compact interval $[0, \pi]$ and $[0, 5]$, respectively.

Clearly, $f_1 \in \text{Lip}_1(1), f_2 \in \text{Lip}_1(1/2)$, and $M_1 = \max_{x \in [0, \pi]} |\sin x| = 1, M_2 = \max_{x \in [0, 5]} |\sqrt{x}| = \sqrt{5}$. From Theorem 3, it follows that

$$\sup_{x \in [0, \pi]} |N_a(x) - \sin x| \leq \frac{2\pi}{n} + 2ne^{-n},$$

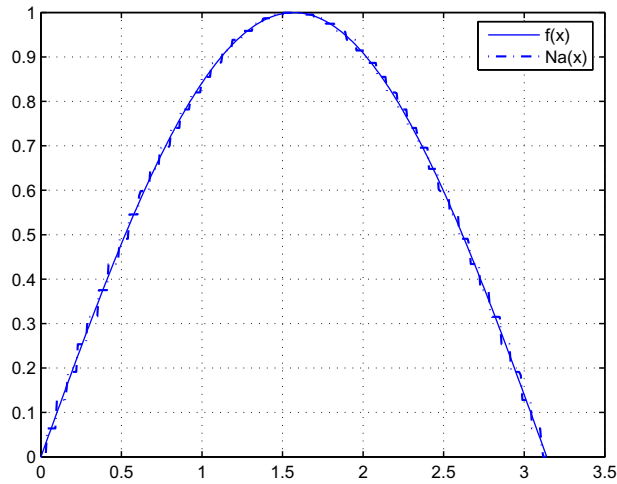


Fig. 4. $n = 50$ case.

Table 2
Approximation error for target function \sqrt{x}

n	$\sup_{x \in [0, 5]} N_a(x) - \sqrt{x} $	$2\sqrt{5}(\frac{2}{n})^{1/2} + 2\sqrt{5}ne^{-n}$
5	0.4903	5.3276
10	0.4012	3.3383
30	0.2629	1.8570
50	0.2000	1.4286

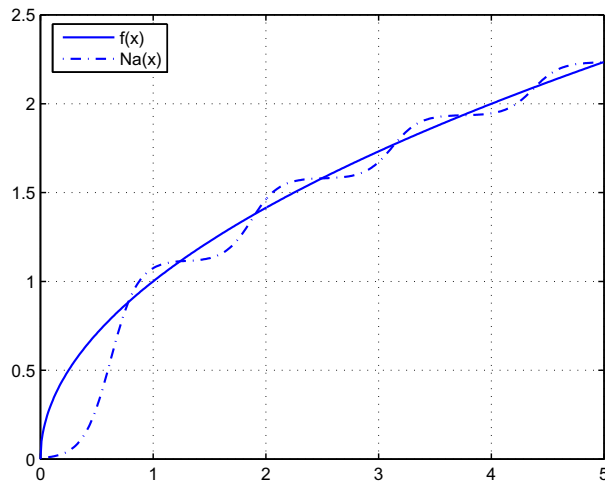
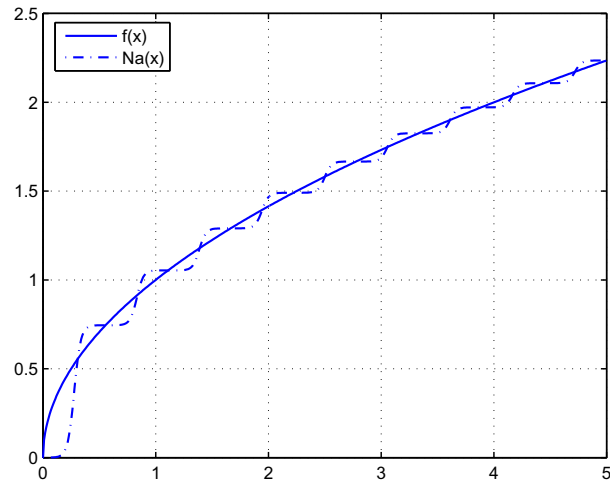
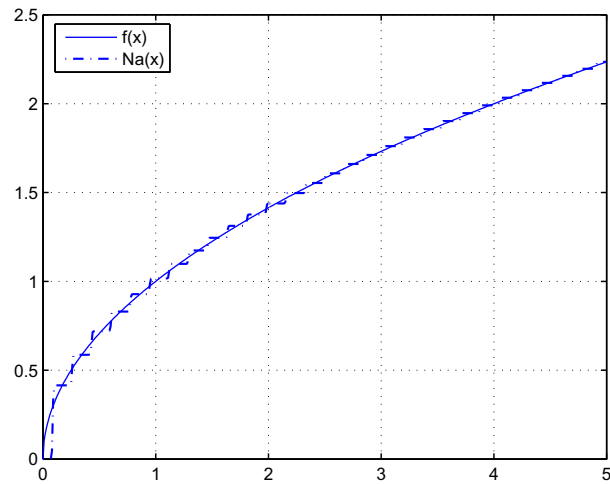
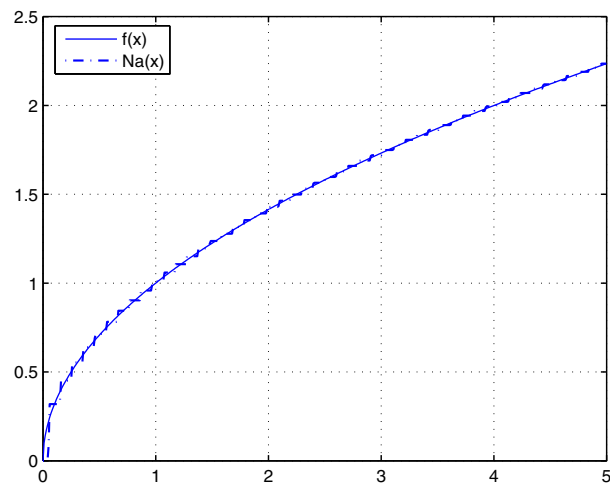


Fig. 5. $n = 5$ case.

Fig. 6. $n = 10$ case.Fig. 7. $n = 30$ case.Fig. 8. $n = 50$ case.

and

$$\sup_{x \in [0,5]} |N_n(x) - \sqrt{x}| \leq 2\sqrt{5} \left(\frac{5}{n}\right)^{1/2} + 2\sqrt{5}ne^{-n}.$$

The following Table 1 shows some approximation errors of the target function $\sin x$, and Figs. 1–4 then demonstrate that the target function is well approximated. The Table 2, Figs. 5–8 show the corresponding ones of the other target function \sqrt{x} .

7. Conclusion

In this paper, we have studied the interpolation and rates of convergence for a class of FNNs. In general, the activation function used in the hidden layer of FNNs is sigmoidal, so the interpolation networks are difficult to be constructed, and the rates of convergence of interpolation networks are also not easy to be estimate. To make them facile, we introduce a type of activation functions, which possess some properties that common sigmoidal functions can not own. Therefore, it is much easier to construct exact and approximate interpolation networks. The error between the exact interpolation network and approximate interpolation network is also good estimated. Using the modulus of continuity of function as a metric we give the estimates of the rate of convergence of approximating continuous functions by approximate interpolation networks. Our methods of proof are constructive. The numerical approximation are in good agreement with theoretical results of error estimation. Our thought lead us to study the “complexity” problem by interpolation network with the activation function $g_j(x)$ in the future.

Acknowledgement

The authors are enormously grateful to the referees for their many valuable comments and suggestions.

References

- [1] G. Cybenko, Approximation by superpositions of sigmoidal function, *Math. Contr. Signals Syst.* 2 (1989) 303–314.
- [2] K.I. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks* 2 (1989) 183–192.
- [3] C.K. Chui, X. Li, Approximation by ridge functions and neural networks with one hidden layer, *J. Approx. Theory* 70 (1992) 131–141.
- [4] H.N. Mhaskar, C.A. Micchelli, Approximation by superposition of a sigmoidal function, *Adv. Appl. Math.* 13 (1992) 350–373.
- [5] M. Leshno, V.Y. Lin, A. Pinks, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, *Neural Networks* 6 (1993) 861–867.
- [6] T.P. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to a dynamic system, *IEEE Trans. Neural Networks* 6 (1995) 911–917.
- [7] T.P. Chen, H. Chen, R.W. Liu, Approximation capability in $C(R^n)$ by multilayer feedforward networks and related problems, *IEEE Trans. Neural Networks* 6 (1995) 25–30.
- [8] T. Nguyen-Thien, T. Tran-Cong, Approximation of functions and their derivatives: a neural network implementation with applications, *Appl. Math. Modell.* 23 (1999) 687–704.
- [9] B. Martin, N. Andreas, Analysis of Tikhonov regularization for function approximation by neural networks, *Neural Networks* 16 (2003) 79–90.
- [10] N. Mai-Duy, T. Tran-Cong, Approximation of function and its derivatives using radial basis function networks, *Appl. Math. Modell.* 27 (2003) 197–220.
- [11] H.N. Mhaskar, When is approximation by Gaussian networks necessarily a linear process? *Neural Networks* 17 (2004) 989–1001.
- [12] Z.B. Xu, F.L. Cao, The essential order of approximation for neural networks, *Sci. China (Series F)* 47 (2004) 97–112.
- [13] Z.B. Xu, F.L. Cao, Simultaneous L^p -approximation order for neural networks, *Neural Networks* 18 (2005) 914–923.
- [14] S. Ferrari, R.F. Stengel, Smooth function approximation using neural networks, *IEEE Trans. Neural Networks* 16 (2005) 24–38.
- [15] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* 70 (2006) 489–501.
- [16] V. Maiorov, Approximation by neural networks and learning theory, *J. Complex.* 22 (2006) 102–117.
- [17] F.L. Cao, T.F. Xie, Z.B. Xu, The estimate for approximation error of neural networks: A constructive approach, *Neurocomputing* 71 (2008) 626–630.
- [18] Y. Ito, Independence of unscaled basis functions and finite mappings by neural networks, *Math. Sci.* 26 (2001) 117–126.
- [19] Y. Ito, K. Saito, Superposition of linearly independent functions and finite mappings by neural networks, *Math. Sci.* 21 (1996) 27–33.
- [20] A. Pinkus, Approximation theory of the MLP model in neural networks, *Acta Numer.* (1999) 143–195.

- [21] E.D. Sontag, Feedforward nets for interpolation and classification, *J. Comp. Syst. Sci.* 45 (1992) 20–48.
- [22] B. Llanas, F.J. Sainz, Constructive approximate interpolation by neural networks, *J. Comput. Appl. Math.* 188 (2006) 283–308.
- [23] F. Brauer, C. Castillo-Chavez, *Mathematical Models in Population Biology and Epidemiology*, Springer-Verlag, New York, 2001, pp. 8–9.
- [24] N. Hritonenko, Y. Yatsenko, *Mathematical Modeling in Economics, Ecology and the Environment*, Science Press, Beijing, 2006, pp. 92–93, Reprint.
- [25] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, New York, 2001, pp. 50–240.
- [26] R. Kress, *Numerical Analysis*, Springer-Verlag, New York, 1998.
- [27] Y. Shrivatava, S. Dasgupta, Neural networks for exact matching of functions on a discrete domain, in: *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, 1990, pp. 1719–1724.