Mathematical and Computer Modelling 52 (2010) 1674–1681

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Constructive approximate interpolation by neural networks in the metric space☆

Feilong Cao [a,*], Shaobo Lin [b], Zongben Xu [b]

[a] Institute of Metrology and Computational Science, China Jiliang University, Hangzhou 310018, Zhejiang Province, PR China
[b] Institute for Information and System Sciences, Xi'an Jiaotong University, Xi'an 710049, Shannxi Province, PR China

ARTICLE INFO

Article history:
Received 27 March 2010
Received in revised form 22 June 2010
Accepted 22 June 2010

Keywords:
Neural networks
Approximate interpolation
Exact interpolation
Metric space

ABSTRACT

In this paper, we construct two types of feed-forward neural networks (FNNs) which can approximately interpolate, with arbitrary precision, any set of distinct data in the metric space. Firstly, for analytic activation function, an approximate interpolation FNN is constructed in the metric space, and the approximate error for this network is deduced by using Taylor formula. Secondly, for a bounded sigmoidal activation function, exact interpolation and approximate interpolation FNNs are constructed in the metric space. Also the error between the exact and approximate interpolation FNNs is given.

1. Introduction

Let (X, d) be a metric space with distance d, and the interpolation nodes $S = \{x_1, x_2, \ldots, x_n\}$ be n distinct points in X. For $\{y_i : i = 1, 2, \ldots, n\} \subset \mathbb{R}$, we call the set

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{1}$$

a set of interpolation samples. If there exists a feed-forward neural network (FNN), $N_e(x)$, satisfying

$$N_e(x_j) = y_j, \quad j = 1, 2, \ldots, n,$$

then $N_e(x)$ is called an exact interpolation FNN of the sample set (1). If for any fixed $\varepsilon > 0$, there is an FNN, $N_a(x)$, such that

$$|N_a(x_j) - y_j| < \varepsilon, \quad j = 1, 2, \ldots, n$$

then $N_a(x)$ is called an ε-approximate interpolation FNN of the sample set (1).

In applications, FNNs are usually trained by using finite input samples. It is known that n arbitrary distinct samples (x_i, f_i) $(i = 1, 2, \ldots, n)$ can be learned precisely by FNNs with n hidden neurons. Several proofs on the existence of exact interpolation FNNs have been proposed in [1–5]. However, it is difficult to fix all the parameters of the interpolation FNNs. So ones turn to study the approximate interpolation FNNs which were first used in [5] as a tool to study the exact interpolation FNNs. Later, some scholars studied the approximate interpolation FNNs and the relationship between the exact and approximation interpolation FNNs. By now, there have been more results related to this topic. We refer the reader to [6–11].

☆ The research was supported by the National Natural Science Foundation of China (No. 60873206) and the Natural Science Foundation of Zhejiang Province of China (No. Y7080235).

* Corresponding author.
E-mail address: feilongcao@gmail.com (F. Cao).

0895-7177/$ – see front matter Crown Copyright © 2010 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2010.06.035

All of these results mentioned above are related to Euclidean space $\mathbb{R}^d$. However, in many applications, the problem of interpolation often arises in general metric space. In this paper, we focus on the approximation and construction of the exact and approximate interpolation FNNs in the metric space. We will construct two types of approximate interpolation FNNs in metric space, one with analytic activation function and the other with sigmoidal activation function.

The rest of this paper is organized as follows. In the next section, we will construct an approximate interpolation FNN with analytic and non-polynomial activation function. Our construction is based on the Lagrange interpolant on the metric space. In Section 3, we will give a rigorous proof of the existence of exact interpolation FNN with sigmoidal function in the metric space. We consider the approximate interpolation FNN with a bounded sigmoidal activation function in Section 4, where an error estimate between exact and approximate interpolation FNNs will be also given.

2. Approximate interpolation FNN with analytic activation function

Before giving the main result of this section, we first construct the Lagrange interpolant and the Newton interpolant in the metric space. The following Proposition 1 is the main tool of our construction.

Proposition 1. *Suppose that (X, d) is a metric space with distance d, and for every interpolation nodes $\{x_1, x_2, \ldots, x_n\} \subset X$, there is a $\xi \in X$ satisfying $d(\xi, x_i) \neq d(\xi, x_j)$ for $1 \le i < j \le n$. Then for the interpolation sample set (1), there exists*

$$P(x) := \sum_{k=0}^{n-1} C_k (d(\xi, x))^k$$

such that

$$P(x_i) = y_i. \tag{2}$$

Proof. In order to prove (2), it is sufficient to prove that the system of equations

$$\sum_{k=0}^{n-1} C_k (d(\xi, x_i))^k = y_i, \quad 1 \le i \le n \tag{3}$$

is solvable. That is to prove that

$$\begin{pmatrix} 1 & d(\xi, x_1) & (d(\xi, x_1))^2 & \ldots & (d(\xi, x_1))^{n-1} \\ 1 & d(\xi, x_2) & (d(\xi, x_2))^2 & \ldots & (d(\xi, x_2))^{n-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & d(\xi, x_n) & (d(\xi, x_n))^2 & \ldots & (d(\xi, x_n))^{n-1} \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \\ \ldots \\ C_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{pmatrix}$$

is solvable. Noting that the coefficients matrix of the system of equations (3) is a Vandermonde matrix, and $d(\xi, x_i) \neq d(\xi, x_j)$, $i \neq j$, then (3) is solvable. This completes the proof of Proposition 1. □

Based on Proposition 1, we can construct the Lagrange interpolant in the metric space as follows.

Proposition 2. *Under the conditions of Proposition 1, there exists*

$$L(x) := \sum_{i=1}^{n} y_i l_i(x)$$

satisfying

$$L(x_i) = y_i, \quad i = 1, 2, \ldots, n,$$

for the sample set (1), where

$$l_i(x) := \prod_{j=1, j\neq i}^{n} \frac{d(\xi, x) - d(\xi, x_j)}{d(\xi, x_i) - d(\xi, x_j)}. \tag{4}$$

The proof of Proposition 2 is obvious. For the sake of brevity, we omit the details.

In order to construct the Newton interpolant in the metric space, we need introduce the divided difference with respect to ξ recursively, by

$$y_\xi[x_i] = y_i, \quad 1 \le i \le n,$$

$$y_\xi[x_j, \ldots, x_k] = \frac{y_\xi[x_j, \ldots, x_{k-1}] - y_\xi[x_{j+1}, \ldots, x_k]}{d(\xi, x_j) - d(\xi, x_k)}.$$

Then the Newton interpolant in the metric space can be constructed as follows.

Proposition 3. *Under the conditions of Proposition 1, there exists*

$$T(x) := y_\xi[x_1] + \sum_{i=1}^{n-1} y_\xi[x_1, \dots, x_{i+1}](d(\xi, x) - d(\xi, x_1))(d(\xi, x) - d(\xi, x_2)) \cdots (d(\xi, x) - d(\xi, x_i)) \tag{5}$$

such that

$$T(x_i) = y_i, \quad 1 \le i \le n$$

for the interpolation sample set (1).

Proof. Let $t_i = d(\xi, x_i)$, and

$$y[t_i] = y_i, \quad 1 \le i \le n,$$
$$y[t_j, \dots, t_k] = \frac{y[t_j, \dots, t_{k-1}] - y[t_{j+1}, \dots, t_k]}{t_j - t_k}.$$

Then form the Newton interpolation formula of univariate polynomial, it is obvious that there exists

$$g(t) := y[t_1] + \sum_{i=1}^{n-1} y[t_1, \dots, t_{i+1}](t - t_1)(t - t_2) \dots (t - t_i)$$

satisfying

$$g(t_i) = y_i, \quad 1 \le i \le n.$$

Moreover, since

$$y_\xi[x_j, \dots, x_k] = y[t_j, \dots, t_k], \quad 1 \le j \le k \le n,$$

then we obtain (5) immediately. □

Now, we are in a position to give our main result in this section. Suppose that

$$\sup_{x \in X} d(\xi, x) \le b < \infty, \tag{6}$$

and $\sigma : [0, b] \to \mathbb{R}$ is analytic and not a polynomial. Then there exist $[c, d] \subset [0, b]$ and $\beta \in [c, d]$ such that for any $k \ge 0$, $\sigma^{(k)}(\beta) \ne 0$ (see [12]). Thus by using Taylor formula, for arbitrary $\theta_j \in [0, b]$, $j = 0, 1, \dots, n$ and any $h > 0$, there holds

$$\sigma(\theta_j ht + \beta) = \sigma(\beta) + \sigma'(\beta)\theta_j ht + \dots + \frac{\sigma^{(n-1)}(\beta)}{(n-1)!}(\theta_j t)^{n-1} + \frac{1}{(n-1)!}\int_0^{\theta_j ht + \beta} \sigma^{(n)}(s)(\theta_j ht + \beta - s)^{n-1} ds.$$

Therefore

$$t^j = \sum_{k=0}^{n-1} \frac{c_{jk} j!}{h^j \sigma^{(j)}(\beta)} \sigma(\theta_k ht + \beta) - \frac{j!}{h^j (n-1)! \sigma^{(j)}(\beta)} \sum_{k=0}^{n-1} c_{jk} \int_0^{\theta_k ht + \beta} \sigma^{(n)}(s)(\theta_k ht + \beta - s)^{n-1} ds,$$

where $(c_{j0}, \dots, c_{jn-1})$ denotes the $(j+1)$ row of W^{-1} where W^{-1} denotes the inverse matrix of

$$W := \begin{pmatrix} 1 & \theta_0 & \dots & \theta_0^{n-1} \\ 1 & \theta_1 & \dots & \theta_1^{n-1} \\ \dots & \dots & \dots & \dots \\ 1 & \theta_{n-1} & \dots & \theta_{n-1}^{n-1} \end{pmatrix}.$$

If we write

$$M := \max_{t \in [-1,1]} |\sigma^{(n)}(t)| \max_{0 \le j \le n-1} \frac{j!}{|\sigma^{(j)}(\beta)|(n-1)!} \sum_{k=0}^{n-1} |c_{jk}||b\theta_k|^n$$

then from [6, Lemma 3.2], we know that for arbitrary $0 \le j \le n-1$ there holds

$$\left| t^j - \sum_{k=0}^{n-1} \frac{c_{jk} j!}{h^j \sigma^{(j)}(\beta)} \sigma(\theta_k ht + \beta) \right| < M h^{n-j}. \tag{7}$$

Based on this, we can construct the approximate interpolation FNN in the metric space as

$$N_a(x) := \sum_{k=0}^{n-1} d_k \sigma(\theta_k h d(\xi, x) + \beta) := \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} \frac{C_j c_{jk} j!}{h^j \sigma^{(j)}(\beta)} \sigma(\theta_k h d(\xi, x) + \beta),$$

where $\{C_0, C_1, \ldots, C_{n-1}\}$ is the solution to the system of equations (3). Furthermore, $N_a(x)$ can be interpreted as a model of FNN with four layers:

- The first layer is the input layer with the input x ($x \in X$).
- The second one is the pre-handling layer, which transform an input x into the distance between x and ξ, $d(\xi, x)$.
- The third one is the handling layer with n neurons in it.
- The last one is the output layer.

From our construction, we can get the following Theorem 1, which describes the error between $N_a(x)$ and the interpolation sample.

Theorem 1. *Suppose that (X, d) is a metric space with distance d, and for every interpolation nodes $\{x_1, x_2, \ldots, x_n\} \subset X$, there is a $\xi \in X$ satisfying $d(\xi, x_i) \neq d(\xi, x_j)$ for $1 \le i < j \le n$. Suppose further that (6) holds and $\sigma : [0, b] \to \mathbb{R}$ is analytic and not a polynomial. Then for any $h > 0$ and sample set (1) there exists an FNN, $N_a(x)$, such that*

$$|N_a(x_i) - y_i| \le CMh, \tag{8}$$

where $C = \sum_{j=1}^{n-1} |C_j|$, and M and C_j are given as the above.

Proof. From Proposition 1, there exists $P(x) = \sum_{j=0}^{n-1} C_j (d(\xi, x))^k$ such that $P(x_i) = y_i,\ i = 1, 2, \ldots, n$. Then

$$|N_a(x_i) - y_i| = |N_a(x_i) - P(x_i)|.$$

Noting (7) we obtain

$$\begin{aligned} |N_a(x) - P(x)| &= \left| \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} \frac{C_j c_{jk} j!}{h^j \sigma^{(j)}(\beta)} \sigma(\theta_k h d(\xi, x) + \beta) - \sum_{j=0}^{n-1} C_j (d(\xi, x))^j \right| \\ &\le \sum_{j=0}^{n-1} |C_j| \sum_{k=0}^{n-1} \left| \frac{c_{jk} j!}{h^j \sigma^{(j)}(\beta)} \sigma(\theta_k h d(\xi, x) + \beta) - (d(\xi, x))^j \right| \\ &\le \sum_{j=0}^{n-1} |C_j| M h^{n-j} \le CMh. \end{aligned}$$

Therefore

$$|N_a(x_i) - y_i| = |N_a(x_i) - P(x_i)| \le CMh.$$

This finishes the proof of Theorem 1. □

3. The existence of exact interpolation FNN with sigmoidal activation function

If $\sigma : \mathbb{R} \to \mathbb{R}$ satisfies

$$\lim_{t \to -\infty} \sigma(t) = 0, \qquad \lim_{t \to \infty} \sigma(t) = 1,$$

then we call it a sigmoidal function. Assume that σ is a bounded sigmoidal function, $A > 0$, and

$$\delta_\sigma(A) := \sup_{t \ge A} \max \{|1 - \sigma(t)|, |\sigma(-t)|\},$$

then $\delta_\sigma(A)$ is non-increasing with variable A and satisfies

$$\lim_{A \to +\infty} \delta_\sigma(A) = 0.$$

Based on these conditions, we now construct the exact interpolation FNN for the interpolation sample set (1). Let $x'_1 = x_1$ and $\{x'_1, x'_2, \ldots, x'_n\}$ satisfy $d(x'_1, x'_i) \le d(x'_1, x'_j)$ for $i \le j$. Then we rearrange the order of elements of the interpolation sample set (1) as

$$(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_n, y'_n).$$

For the sake of convenience, we also denote the set of $(x'_i, y'_i)(i = 1, 2, \ldots, n)$ as

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}.$$

Then the exact interpolation FNN can be constructed as follows.

$$N_e^A(x) := \sum_{j=1}^{n-1} c_j \sigma\left(-2A \frac{d(x_1, x) - d(x_1, x_j)}{d(x_1, x_{j+1}) - d(x_1, x_j)} + A\right) + c_n \sigma\left(-2A \frac{d(x_1, x) - d(x_1, x_n)}{d(x_1, x_n) - d(x_1, x_{n-1})} + A\right).$$

From the definition, we know that when $i < j$,

$$-2A\frac{d(x_1,x_i)-d(x_1,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A \geq A,$$

when $i = j$,

$$-2A\frac{d(x_1,x_i)-d(x_i,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A = A,$$

when $i = j+1$,

$$-2A\frac{d(x_1,x_i)-d(x_1,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A = -A$$

and when $i > j+1$

$$-2A\frac{d(x_1,x_i)-d(x_1,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A \leq -A.$$

Therefore, by the definition of $\delta_\sigma(A)$ we obtain that

$$1-\sigma\left(-2A\frac{d(x_1,x_i)-d(x_1,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A\right) \leq \delta_\sigma(A), \quad 1 \leq i \leq j, \tag{9}$$

and there holds

$$\sigma\left(-2A\frac{d(x_1,x_i)-d(x_1,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A\right) \leq \delta_\sigma(A), \quad j+1 \leq i \leq n. \tag{10}$$

The following Theorem 2 is the main result of this section, which gives a rigorous proof of the existence of the exact interpolation FNN in the metric space.

Theorem 2. *If σ is a bounded sigmoidal function and*

$$\delta_\sigma(A) < \frac{1}{4n}, \tag{11}$$

then for sample set (1) there exists $\{c_j\}_{j=1}^n \subset \mathbb{R}$ such that $N_e^A(x)$ is an exact interpolation FNN.

Proof. In order to prove Theorem 2, it is sufficient to prove that the following systems of equation with variable $\{c_i\}_{i=1}^n$

$$N_e^A(x_i) = y_i \quad (i = 1, \ldots, n) \tag{12}$$

is solvable under the condition (11). Denote

$$e_{i,j}(A) := \sigma\left(-2A\frac{d(x_1,x_i)-d(x_1,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A\right), \quad i,j = 1,\ldots,n-1,$$
$$e_{i,n}(A) := \sigma\left(-2A\frac{d(x_1,x_i)-d(x_1,x_j)}{d(x_1,x_n)-d(x_1,x_{n-1})}+A\right), \quad i = 1,\ldots,n-1,$$
$$e_{n,j}(A) := \sigma\left(-2A\frac{d(x_1,x_n)-d(x_1,x_j)}{d(x_1,x_{j+1})-d(x_1,x_j)}+A\right), \quad j = 1,\ldots,n-1,$$
$$e_{n,n}(A) := \sigma(A).$$

Then the coefficient matrix of the system of equations (12) can be written as

$$D_n(A) := \begin{vmatrix} e_{11}(A) & e_{12}(A) & \ldots & e_{1n}(A) \\ e_{21}(A) & e_{22}(A) & \ldots & e_{2n}(A) \\ \ldots & \ldots & \ldots & \ldots \\ e_{n1}(A) & e_{n2}(A) & \ldots & e_{nn}(A) \end{vmatrix}.$$

Let

$$d_{ij}(A) = e_{ij}(A) - e_{i+1j}(A), \quad i,j = 1,\ldots,n-1,$$
$$d_{in}(A) = e_{in}(A) - e_{i+1n}(A), \quad i = 1,\ldots,n-1,$$
$$d_{nj}(A) = e_{nj}(A), \quad j = 1,\ldots,n-1, \qquad d_{nn}(A) = e_{nn}(A),$$

then

$$D_n(A) = \begin{vmatrix} d_{11}(A) & d_{12}(A) & \dots & d_{1n}(A) \\ d_{21}(A) & d_{22}(A) & \dots & d_{2n}(A) \\ \dots & \dots & \dots & \dots \\ d_{n1}(A) & d_{n2}(A) & \dots & d_{nn}(A) \end{vmatrix}.$$

Moreover, from (11) and the definition of $\delta_\sigma(A)$ we obtain that if $t \geq A$, then

$$|\sigma(-t)| < \frac{1}{4n}, \qquad |1 - \sigma(t)| < \frac{1}{4n}.$$

Therefore,

$$\begin{aligned} d_{ii}(A) &= \sigma(A) - \sigma(-A) = 1 - (1 - \sigma(A)) - \sigma(-A) \\ &\geq 1 - \frac{1}{4n} - \frac{1}{4n} = 1 - \frac{1}{2n}, \quad 1 \leq i \leq n - 1, \end{aligned}$$

and

$$\begin{aligned} d_{nn}(A) &= \sigma(A) = 1 - (1 - \sigma(A)) \\ &\geq 1 - \frac{1}{4n} \geq 1 - \frac{1}{2n}. \end{aligned}$$

Noting (9) and (10), we get

$$\sum_{j=1, j\neq i}^{n} |d_{ij}| = |e_{ij} - e_{ij+1}| \leq |e_{ij}| + |e_{ij+1}| < n \cdot \frac{1}{2n} = \frac{1}{2}, \quad i = 1, \dots, n.$$

Hence

$$d_{ii}(A) > \sum_{j=1, j\neq i}^{n} |d_{ij}| \quad (i = 1, \dots, n).$$

Then from the strictly diagonally dominant matrices are invertible principle (see [13]), we have

$$D_n(A) \neq 0,$$

which means that the system of equations (12) is solvable. This completes the proof of Theorem 2. □

4. Approximate interpolation FNN with sigmoidal activation function

The inner weights and thresholds of the exact interpolation FNN, $N_e^A(x)$, in Theorem 2 depend on the interpolation nodes, while the coefficients of $N_e^A(x)$ is a solution to the system of equations (12). We have proved that (12) is solvable when A is sufficient large, but it is not easy to work out the solution. Therefore, we turn to consider the approximate interpolation FNN, $N_a^A(x)$:

$$N_a^A(x) := \sum_{j=1}^{n-1} (y_j - y_{j+1}) \sigma \left(-2A \frac{d(x_1, x) - d(x_1, x_j)}{d(x_1, x_{j+1}) - d(x_1, x_j)} + A \right) + y_n \sigma \left(-2A \frac{d(x_1, x) - d(x_1, x_n)}{d(x_1, x_n) - d(x_1, x_{n-1})} + A \right).$$

It is obvious that the exact interpolation FNN, $N_e^A(x)$, and the approximate interpolation FNN, $N_a^A(x)$, differ only in the coefficients. The following Theorem 3 shows the error between $N_e^A(x)$ and $N_a^A(x)$.

Theorem 3. *If σ is a bounded sigmoidal function and (11) holds, then*

$$\left| N_e^A(x) - N_a^A(x) \right| \leq \frac{(2n+1)\delta_\sigma(A)\|\sigma\|}{1 - (2n+1)\delta_\sigma(A)} \left(\sum_{j=1}^{n-1} |y_j - y_{j+1}| + |y_n| \right),$$

where $\|\sigma\| := \sup_{t\in\mathbb{R}} |\sigma(t)|$.

Proof. Since (11) holds, from Theorem 1 we know that (12) is solvable. We denote its solution as $V_c := (c_1, \dots, c_n)$, and the coefficients matrix of (12) as M. Let $V_y = (y_1, \dots, y_n)$, then (12) can be rewritten as

$$MV_c^T = V_y^T, \tag{13}$$

where V^T denotes the transpose of the vector V. Define

$$U := \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 0 & 1 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

A direct computation yields that the inverse matrix of U

$$U^{-1} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

Write

$$M - U = (\alpha_{ij})_{i,j=1}^{n,n},$$

then

$$|\alpha_{ij}| \le \delta_\sigma(A).$$

If we denote

$$U^{-1}(M - U) = (\beta_{ij})_{i,j=1}^{n,n},$$

then

$$|\beta_{ij}| \le 2\delta_\sigma(A), \qquad |\beta_{nj}| \le \delta_\sigma(A) \quad (i = 1, \dots, n-1, j = 1, \dots n). \tag{14}$$

Let

$$V_Y := (y_1 - y_2, \dots, y_{n-1} - y_n, y_n), \qquad \Delta V_c := V_c - V_Y,$$

we have

$$UV_Y^T = V_y^T.$$

Noting (13) we obtain

$$(U + (M - U))(V_Y^T + \Delta V_c^T) = V_y^T.$$

That is

$$U\Delta V_c^T = -(M - U)\Delta V_c^T - (M - U)V_Y^T.$$

Hence

$$\Delta V_c^T = -U^{-1}(M - U)\Delta V_c^T - U^{-1}(M - U)V_Y^T.$$

The last equation together with (14) yields

$$\sum_{i=1}^{n} |\Delta V_{C_i}| \le (2n+1)\delta_\sigma(A) \sum_{i=1}^{n} |\Delta V_{C_i}| + (2n+1)\delta_\sigma(A) \left(\sum_{i=1}^{n_1} |y_i - y_{i+1}| + |y_n| \right).$$

Furthermore, from (11) we know

$$\sum_{i=1}^{n} |\Delta V_{C_i}| \le \frac{(2n+1)\delta_\sigma(A)}{1 - (2n+1)\delta_\sigma(A)} \left(\sum_{j=1}^{n} |y_j - y_{j+1}| + |y_n| \right).$$

Since

$$|N_e^A(x) - N_a^A(x)| \le \sum_{i=1}^{n} |\Delta V_{C_i}| \|\sigma\|,$$

we have

$$|N_e^A(x) - N_a^A(x)| \le \frac{(2n+1)\delta_\sigma(A)\|\sigma\|}{1 - (2n+1)\delta_\sigma(A)} \left(\sum_{j=1}^{n-1} |y_j - y_{j+1}| + |y_n| \right).$$

This finishes the proof of Theorem 3. □

References

[1] Y. Ito, K. Saito, Superposition of linearly independent functions and finite mappings by neural networks, Math. Sci. 21 (1996) 27–33.
[2] Y. Ito, Independence of unscaled basis functions and finite mappings by neural networks, Math. Sci. 26 (2001) 117–126.
[3] A. Pinkus, Approximation theory of the MLP model in neural networks, Acta Numer. (1999) 143–195.
[4] 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.
[5] E.D. Sontag, Feedforward nets for interpolation and classification, J. Comput. System Sci. 45 (1) (1992) 20–48.
[6] X. Li, Interpolation by ridge polynomials and its application in neural networks, J. Comput. Appl. Math. 144 (2002) 197–209.
[7] H.X. Li, E.S. Lee, Interpolation functions of feedfoward neural networks, Comput. Math. Appl. 46 (2003) 1861–1874.
[8] H.X. Li, L.X. Li, J.Y. Wang, Interpolation representation of feedforward neural networks, Math. Comput. Model. 37 (2003) 829–847.
[9] B. Llanas, F.J. Sainz, Constructive approximate interpolation by nerual networks, J. Comput. Appl. Math. 188 (2006) 283–308.
[10] B. Llanas, S. Lantarón, Hermite interpolation by neural networks, Appl. Math. Comput. 191 (2007) 429–439.
[11] F.L. Cao, Y.Q. Zhang, Z.R. He, Interpolation and rates of convergence for a class of neural networks, Appl. Math. Modelling 33 (2009) 1441–1456.
[12] T.F. Xie, F.L. Cao, The ridge function representation of polynomials and an application to neural networks. Acta Math. Sin. Eng. Ser. (2010) (in press).
[13] P. Lascaux, T. Theodor, Analyse Numerique Matricielle Apliquée a l'Art de l'Ingenieur, Masson, Paris, 1986.