Multiresolution Neural Networks Based on Immune Particle Swarm Algorithm

Li Ying[1,2] and Deng Zhidong[1,⋆]

[1] Department of Computer Science and Technology State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100084
{liyingqh, michael}@tsinghua.edu.cn
[2] College of Computer Science and Technology Jilin University, Changchun 130021, P.R. China

Abstract. Inspired by the theory of multiresolution analysis (MRA) of wavelets and artificial neural networks, a multiresolution neural network (MRNN) for approximating arbitrary nonlinear functions is proposed in this paper. MRNN consists of a scaling function neural network (SNN) and a set of sub-wavelet neural networks, in which each sub-neural network can capture the specific approximation behavior (global and local) at different resolution of the approximated function. The structure of MRNN has explicit physical meaning, which indeed embodies the spirit of multiresolution analysis of wavelets. A hierarchical construction algorithm is designed to gradually approximate unknown complex nonlinear relationship between input data and output data from coarse resolution to fine resolution. Furthermore, A new algorithm based on immune particle swarm optimization (IPSO) is proposed to train MRNN. To illustrate the effectiveness of our proposed MRNN, experiments are carried out with different kinds of wavelets from orthonormal wavelets to prewavelets. Simulation results show that MRNN provides accurate approximation and good generalization.

1 Introduction

Artificial neural networks (ANN) and wavelet theory become popular tools in various applications such as engineering problems, pattern recognition and nonlinear system control. Incorporating ANN with wavelet theory, wavelet neural networks (WNN) was first proposed by Zhang and Benveniste [1] to approximate nonlinear functions. WNNs are feedforward neural networks with one hidden layer, where wavelets were introduced as activation functions of the hidden neurons instead of the usual sigmoid functions. As a result of the excellent properties of wavelet theory and the adaptive learning ability of ANN, WNN can make an remarkable improvement for some complex nonlinear system control and identification. Consequently, WNN was received considerable attention [2,3,4,5,6,12].

⋆ Correspondence author. This work was supported by the National Science Foundation of China under Grant No. 60321002.

L. Jiao et al. (Eds.): ICNC 2006, Part I, LNCS 4221, pp. 97–106, 2006.

However, in general the hidden layer of WNN consists of only wavelet nodes and the common used wavelets are Mexican hat function. Multiresolution analysis, the main characteristic of wavelet theory, has still not been efficiently taken into account. In addition, the number of the hidden layer nodes and efficient learning algorithms of WNN are still challenging problems. The most used learning algorithm is backpropagation based on gradient descent,quasi-Newton, Levenburg-Marquardt and conjugate gradient techniques, which easily leads to a local minimum especially for multimodal function optimization problem partially attributed to the lack of a stochastic component in the training procedure. Particle swarm optimization (PSO) in [8,9] is a stochastic population-based optimization technique. The simplicity of implementation and weak dependence on the optimized model of PSO make it a popular tool for a wide range of optimization problems.

In this paper, a multiresolution neural network (MRNN) is proposed, which consists of a scaling function neural network (SNN) and a set of sub-wavelet neural networks (sub-WNN), in which each sub-neural network can capture the specific approximation behavior (global and local information) at different resolution of the approximated function. Such sub-WNN have a specific division of work. The adjusted parameters of MRNN include linear connective weights and nonlinear parameters: dilation and translation, which have explicit physical meaning, i.e., resolution and domain of signification (defined by the training set). The structure of MRNN is designed according to the theory of multiresolution analysis, which plays a significant role in wavelet analysis and approximation of a given function. To determine the structure of MRNN, we give a hierarchical construction algorithm to gradually approximate the unknown function from global to local behavior. In addition, a new learning algorithm referred as immune particle swarm optimization (IPSO) is proposed to train MRNN. In this approach, the introduction of immune behavior contributes to increase the diversity of particles and improve the speed and quality of convergence. For validating the effectiveness of our proposed MRNN, we take different kinds of wavelets from orthonormal wavelets to prewavelets. Experiments were carried out on different types of functions such as discontinuous, continuous and infinitely differentiable functions. Simulation results show that MRNN provides accurate approximation and good generalization. In addition, our proposed learning algorithm based on IPSO can efficiently reduce the probability of trapping in the local minimum.

This paper is organized as follows. The MRNN is introduced in section 2. In section 3, the hierarchical construction algorithm is described,and the training algorithm based on IPSO is presented to update the unknown parameters of MRNN. Finally, simulation examples are utilized to illustrate the good performance of the method in section 4.

2 Multiresolution Neural Networks

The main characteristics of wavelets are their excellent properties of time-frequency localization and multi-resolution. The wavelet transform can capture high frequency (local) behavior and can focus on any detail of the observed object through modulating the scale parameters, while the scale function captures the low frequency (global behavior). In this sense wavelets can be referred to as a mathematical microscope. Firstly, we give a simplified review of wavelets and multiresolution analysis of $L^2(\mathbb{R})$.

A series sequence $V_j, j \in \mathbb{Z}$ of closed subspaces in $L^2(\mathbb{R})$ is called a multiresolution analysis (MRA) if the following holds:
(1) $\cdots \subset V_j \subset V_{j+1} \cdots$
(2) $f(\cdot) \in V_j$ if and only if $f(2\cdot) \in V_{j+1}$
(3) $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$, $\overline{\bigcap_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$
(4) There exists a function $\phi \in V_0$, such that its integer shift $\{\phi(\cdot - k), k \in \mathbb{Z}\}$ form a Riesz basis of V_0.

It is noted that ϕ is a scaling function. Let $\psi(x)$ be the associated wavelet function. For $j \in \mathbb{Z}$, $\{\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k), k \in \mathbb{Z}\}$ forms a Riesz basis of V_j, $\{\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k), k \in \mathbb{Z}\}$ forms a Riesz basis of W_j, where W_j is the orthogonal complement of V_j in V_{j+1}, i.e. $V_{j+1} = V_j \oplus W_j$, where V_j and W_j are referred to as scaling space and detail space (wavelet space) respectively. Obviously,$L^2(\mathbb{R}) = V_J \bigcup_{j=J}^{\infty} \oplus W_j$, and $L^2(\mathbb{R}) = \bigcup_{j=-\infty}^{\infty} \oplus W_j$.

For any function $f(x) \in L^2(\mathbb{R})$ could be written

$$f(x) = \sum_{k\in\mathbb{Z}} c_{J,k}\phi_{J,k}(x) + \sum_{j\geq J}^{\infty} \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(x)$$

as a combination of global information and local information at different resolution levels. On the other hand, $f(x)$ could also be represented as

$$f(x) = \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(x)$$

as a linear combination of wavelets at different resolutions levels, i.e. the combination of all local information at different resolution levels. The above equations indeed embody the concept of multiresolution analysis and imply the ideal of hierarchical and successive approximation, which is the motivation of our work.

Inspired by the theory of multiresolution analysis, we present a new wavelet neural networks model, called multiresolution neural networks (MRNN). The MRNN consists of a scaling function neural network (SNN) and a set of sub-wavelet neural networks (WNN). In SNN, scaling function at a specific resolution are introduced as active functions, while the active functions of each sub-WNN correspond to wavelet functions at different resolution. The unknown parameters of MRNN include linear connective weights and nonlinear parameters: dilation and translation parameters. The architecture of the proposed MRNN is shown in Fig 1. The formulation of MRNN is given as follows:

$$\hat{y}(x) = \sum_{i=0}^{C} \hat{y}^i(x)$$
$$\hat{y}^0(x) = \sum_{j=1}^{N^0} w_{0,j}\phi(a_0 x - b_{0,j}), y^i(x) = \sum_{i=1}^{C} \sum_{j=1}^{N^i} w_{i,j}\psi(a_i x - b_{i,j})$$

Here $\hat{y}^0(x)$ represents the output of SNN and $\hat{y}^i(x)$ is the output of the i-th sub-wavelet neural network,i.e. the output of WNN-i. Note that C is the number of sub-wavelet neural networks, i.e. the number of different resolution level of wavelets. N^0 denotes the number of neurons of SNN, and $N^i, 1 \le i \le C$, is the number of WNN-i. The number of whole neurons of NRNN is $N = \sum_{i=0}^{C} N^i$. $a_i, b_{i,j}$ and $w_{i,j} \in \mathbb{R}, 1 \le i \le N^i, 0 \le i \le C$ are dilation parameters, transition parameters and linear output connective weights, which need to be adjusted to fit the known data optimally. Dilation parameters have explicit meaning, i.e., resolution, which satisfies $a_i > 0$ and $a_0 \le a_1 < a_2 \cdots < a_C$. The number of unknown parameters to be adjusted sums up to $C + 2N$. The other parameters including C and $N^i, i = 1, \cdots, C$, are called structure parameters. In general, such structure parameters must be settled in advance. But in this paper, due to the hierarchical learning ability of our proposed MRNN, structure parameters can be partially determined in the training process.

For SNN: $\{\phi(a_0 x - b_{0,j}), j = 1, 2, \cdots, N^0\}$,are taken as active functions, which capture the global behavior of the approximated function at the a_0 coarse resolution level.

For WNN-i: $\{\psi(a_i x - b_{i,j}), j = 1, 2, \cdots, N^i\}$,are taken as active functions of the i-th sub-wavelet neural networks, which capture the local behavior of approximated function at the a_i finer resolution level. And with an increase of i, the WNN-i focus on the more local behavior of the unknown function. The adaption of scaling function can rapidly implement a global approximation and probably obtain good performance with a relative small model size.

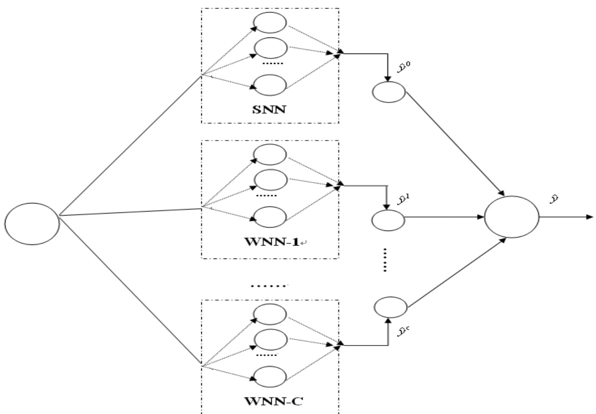

Fig. 1. Architecture of Multiresolution Neural Networks: a scaling function neural netwok and C sub-wavelet neural networks (WNN-1,$\cdots$,WNN-C)

The most commonly used wavelets are continuous wavelets, Daubecies wavelets and spline wavelets and so on. For the general WNN, the most frequently adopted active functions are continuous wavelets such as Morlet, Mexican Hat and Gaussian derivatives wavelets, which do not exist scaling functions. Daubechies compactly supported orthonormal wavelets are provided by scaling and wavelets filters, while they have no analytic formula. In this paper, the cascade algorithm i.e. the repeated single-level inverse wavelet transform, is employed to approximate the corresponding scale function and wavelet function, which are adopted as active functions in our proposed MRNN. Considering that such functions are the approximated form, an learning algorithm based on gradient descent seems unreasonable. A training algorithm not demanding gradient information will be described in the next. In addition,pre-wavelets are also considered as candidates for active functions of our MRNN. Compared with orthogonal wavelets, pre-wavelets have more freedom degrees and potential advantages. Let $N_m(x)$ be the spline function of m-th order supported on $[0,m]$, then a m-th order pre-wavelet corresponding to $N_m(x)$ is defined as : $\psi_m(x) = \frac{1}{2^{m-1}} \sum_{n=0}^{3m-2} (-1)^n [\sum_{j=0}^{m} \begin{pmatrix} m \\ j \end{pmatrix} N_{2m}(n-j+1)] N_m(2x-n)$.

Given a training set $S : S = \{(x_1, y_1), \cdots, (x_L, y_L)\}, \quad x_i, y_i \in \mathbb{R}$ where L is the total number of training samples and (x_i, y_i) is input-output pair. Denote the input sample vector $\mathbf{X} = [\mathbf{x_1}, \cdots, \mathbf{x_L}]^{\mathbf{t}} \in \mathbb{R}^{\mathbf{L}}$ and output sample vector $\mathbf{Y} = [\mathbf{y_1}, \cdots, \mathbf{y_L}]^{\mathbf{t}} \in \mathbb{R}^{\mathbf{L}}$, where 't' represents the transpose.

For a training set S, there must exit an unknown function $f : \mathbb{R} \to \mathbb{R}$, s.t.

$$y_i = f(x_i), 1 \le i \le L$$

The network deals with finding a function $\tilde{f}$ that can approximate the unknown function f based on the training set S. The next problems are how to determine the structure parameters and train MRNN based on the given training set S so that the error between the output of MRNN and $\mathbf{Y}$ is minimal.

3 Hierarchical Training Algorithm Based on Immune Particle Swarm Optimization

3.1 Hierarchical Training Strategy

According to the multiresolution property of our proposed neural networks, we employ a hierarchical learning strategy at the start of SNN, which is the coarsest resolution. The concrete steps are given in the following:

(1). Determine the scaling function and the associated wavelet and give an expected error index. Let $[A, B]$ and $[C, D]$ be the main range of scaling function and mother function respectivley. Note that, the main range refer to the support for the compactly supported scaling function and wavelet , while for a non compactly supported functions, due to the rapid vanishing of wavelets, roughly take a range where the value of non compactly supported functions is usually bigger. Let $I_1, I_2]$ be the significant domain of the input vectors.

(2). First train SNN-the coarsest resolution level. The dilation parameter a_0 of SNN is random in a reasonable range. For covering the significant domain of the input vectors, SNN may demand more neurons (translation parameters), because the support of active functions at resolution a_0 becomes smaller when a relative larger a_0 is taken. In detail, the translation parameters belong to the rang of $[-B + a_0 I_1, -A + a_0 I_2]$. According to the width of $[-B + a_0 I_1, -A +$

$a_0 I_2]$, a ratio is designed to determine the number of neurons N^0, then we randomly chosen the translation parameters in $[-B + a_0 I_1, -A + a_0 I_2]$. The linear connective weights are initialized with small random values, taken from a normal distribution with zero mean. Now we compute the current error for such initialization method. Repeating the above process several times, we chose a good trade-off between the error index, the size of SNN and resolution. After we determine the structure parameters, our proposed new training algorithm based on immune particle swarm optimization is used to update the dilation, translation parameters and linear weights until the current error reduces very slowly or obtains a certain admissible error range.

(3). After training SNN, a sub-WNN (WNN-1) is added to approximate the difference between the expected output vector and the estimated output of SNN. Define $D^1 = \{d_j^1, j = 1, \cdots, L\}$ as the expected output of WNN-1, where $d_j^1 = Y_j - \hat{y}^0(x_j)$. Similar to the above, the resolution of WNN-1 a_1 needs to be randomly chosen in a relative small reasonable range larger than a_0. Then the translation parameters are initialized from the range $[-D + a_1 I_1, -C + a_1 I_2]$, in which the number of neurons of WNN-1 is also determined using a width of $[-D + a_1 I_1, -C + a_1 I_2]$ and the comparison of several trial results for reference. Then the unknown parameters of WNN-1 are trained based on the current input data X and expected output data D^1 using the IPSO algorithm.

(4). While the approximation error of WNN-1 dose not yet reach the expected error, repeat step (3) to gradually add WNN-i+1 until the expected error is reached. Note that the expected output vector of WNN-i+1 is the difference between the expected output vector WNN-i and the estimated output vector of WNN-i. Here, denote WNN-0 as SNN. In addition, the dilation parameter of WNN-i+1 needs to be larger than that of WNN-i.

From the above hierarchical training strategy, the approximation of an unknown function is gradually performed from coarse resolution to fine resolution, then from fine resolution to finer resolution. This process gradually focuses on different details at different resolutions of the approximated function. Each subneural network of MRNN has a definite division of work. Additionally through this strategy, the complexity of the training problem of MRNN is greatly reduced, MRNN can be rapidly and effectively constructed.

3.2 Training Algorithm Based on Immune Particle Swarm Optimization

The particle swarm optimization method mimics the animal social behaviors such as schools of fish, flocks of birds etc and mimics the way they find food sources. PSO has gained much attention and wide applications in different optimization problems. PSO has the ability to escape from local minima traps due to its stochastic nature. Assume that a particle swarm contains M particles. Denote Z_i and V_i as the position and velocity of the i-th particle.In fact the position of each particle is a potential solution. Define f_i as the fitness value of the i-th particle, which is used to evaluate the goodness of a position. P_i represents the best position of the i-th particle so far. P_g is the best position of all the particles in a particle swarm so far. The velocity and position of each particle are updated based on the following equations:

$$\begin{aligned} V_i(k+1) &= wV_i(k) + c_1r_1(P_i - x) + c_2r_2(P_g - Z_i) \\ Z_i(k+1) &= Z_i(k) + Z_i(k+1), \quad i = 1, 2, \cdots, M \end{aligned} \tag{1}$$

where w is inertial weight, c_1 and c_2 are constriction factors, r_1 and r_2 are random numbers in $[0, 1]$, and k represents the iteration step. The empirical investigations shows the importance of decreasing the value of w from a higher valve (usually 0.9 to 0.4) during the searching process.

In optimization processing of PSO, premature convergence would often take place especially for the case of an uneven solution population distribution. In this paper, we propose two kinds of mechanism to improve the the diversity of particles and reduce the premature convergence.

One mechanism is the combination of an immune system and PSO. Biological research on the immune system indicates that the immune principle can provide inspiration for improvement of the performance of evolution computations [11]. Immune behavior is beneficial for retaining diversity, which can effectively avoid premature convergence and improve the search speed and quality of the evolution algorithm based on populations. Inspired by the diversity of biological immune systems, we introduce a diversity retaining mechanism based on concentration and immune memory for PSO.

On the other hand, we find another efficient mechanism for diversity, called a forgetfulness mechanism. The basic ideal is that we impose each particle and the swarm a slight amnesia when the particle swarm trends to premature convergence. This mechanism gives the particles a new chance based on existing better conditions. Firstly, we define a merit τ to reflect the convergent status of the particle swarm as follows:

$$\tau = \frac{\sigma_f^2}{max_{1 \leq j \leq M}\{(f_j - \bar{f})^2\}}, \quad \sigma_f^2 = \frac{1}{M}\sum_{i=1}^{M}(f_i - \bar{f})^2, \quad \bar{f} = \frac{1}{M}\sum_{i=1}^{M} f_i \tag{2}$$

Here M is the size of the population. $\bar{f}$ is the average fitness of all particles, and σ_f^2 is the covariance of fitness. For a given small threshold, if τ is less than this threshold and the expected solution have not been reached, then we think that this particle swarm tends to premature convergence. Under this condition, we impose each particle and the swarm a slight amnesia. Each particle forgets its historical best position and considers the current position as its best position. Similarly, the swarm does not remember its historical global best position and choose the best position from the current positions of all particles. The implementation of our algorithm can be categorized into the following steps:

Step-1. Define the solution space and a fitness function and determine c_1, c_2, the size of particles - M and the size of the immune memory library - U.

Step-2. Initialize randomly the position and velocity of each particle.

Step-3. Proceed PSO under a given iteration steps (100 steps), and store the best position P_g in the immune memory library at each iteration. Then evaluate the convergence status using (2). If the convergence status is good, continue *Step-3* until the expected error is reached. Otherwise, go to *Step-4*.

Step-4. Introduce the forgetfulness mechanism: the particle swarm and each particle all forget their historical best position.

Step-5. Build a candidate particle set including 2M particles, where M particles are obtained in terms of the update formulation (1), while the remaining M particles are generated randomly.

Step-6. Compute each candidate particle concentration d_i and selection probability s_i, which are defined as : $d_i = (\sum_{j=1}^{2M} |f_i - f_j|)^{-1}, s_i = d_i / \sum_{j=1}^{2M} d_j), \quad i = 1, 2, \cdots, 2M$.

Step-7. Undate the particle swarm: chose M particles with large selection probabilities from the candidate set and replace them with lower fitness among the chosen M particles by the particles in the immune memory library, then turn to *Step-3*.

4 Numerical Experiments

In this section, for evaluating the performance of the network, we take the following merits defined in [1] as a criterion to compare the performance of various methods.

$$J = \sqrt{(\sum_{l=1}^{N} y_l - \hat{y}_l)/(\sum_{l=1}^{N} y_l - \bar{y})}, \bar{y} = \frac{1}{N}\sum_{l=1}^{N} y_l$$

where y_l is the desired output and $\hat{y}_l$ is the estimated output from the constructed neural networks.

Experiments are carried out on different types of functions such as discontinuous , continuous and infinitely differentiable in order to to demonstrate the validity of the presented MRNN. We respectively selected spline wavelets with order 3 and Daubechies wavelet with support width 2N-1 and vanishing moment N. The approximated functions from $f_1(x)$ to $f_2(x)$ are given in the following:

$$f_1(x) = \begin{cases} -2.186x - 12.864, -10 \le x < -2 \\ 4.246x, -2 \le x < 0 \\ 10e^{-0.05x-0.5} sin[(0.03x + 0.7)x], 0 \le x \le 10 \end{cases}$$

$$f_2(x) = \begin{cases} 0, & -10 \le x < -7.5 \\ 0.2x + 1.5, & -7.5 \le x < -5 \\ 0.5, & -5 \le x < -2.5 \\ 0.2x + 1, & -2.5 \le x < 0 \\ -0.2x + 1, & 0 \le x < 2.5 \\ 0.5, & 2.5 \le x < 5 \\ -0.2x + 1.5, & 5 \le x < 7.5 \\ 0, & 7.5 \le x \le 10 \end{cases} \qquad f_2(x) = \begin{cases} 0, & -10 \le x < -7.5 \\ 1/3, & -7.5 \le x < -5 \\ 2/3, & -5 \le x < -2.5 \\ 1, & -2.5 \le x < 2.5 \\ 1/3, & 2.5 \le x < 5 \\ 0, & 7.5 \le x \le 10 \end{cases}$$

$$f_4(x) = sin(6\pi/20(x + 10)), \qquad -10 \le x \le 10$$

For each approximated function, we sampled 200 points distributed uniformly over $[-10, 10]$ as training data to construct and train the associated MRNN. The test data was obtained through uniformly sampled data of 200 points. Each simulation is computed on uniformly sampled test data of 200 points. In the following, we show the maximal performance index and minimal performance index among nine simulations to evaluate the performance of our MRNN. The simulation results of the approximated functions can be seen in Table I. From Fig 2 and Fig 3 we show the good approximation performance of our MRNN for $f_1(x), f_2(x), f_3(x)$ and $f_4(x)$. Additionally, in order to evaluate the performance of our MRNN, for $f_i(x)$, we make a comparison between our MRNN and other works [1,12]. The Mexican Hat function are taken as the wavelet function in [1,12], whereas we take the B-spline wavelet with order 4 as the active function. The comparised results are given in Table II. It should be noticed that the performance of MRNN shown in Table II is the average of the nine simulations.

Table 1. Simulation results of our MRNN

Example	parameters number	Maximal performance index	Minimal performance index
$f_1(x)$	23	0.040297	0.021052
$f_2(x)$	13	0.00010712	8.6695e-005
$f_3(x)$	22	0.060172	2.1689e-005
$f_4(x)$	17	0.023313	0.033459

Table 2. Comparison of our MRNN with others work for $f_1(x)$

Method	Number of unknown parameters	Performance index
MRNN-prewavelets	23	0.03259
WNN [1]	22	0.05057
WNN [12]	23	0.0480

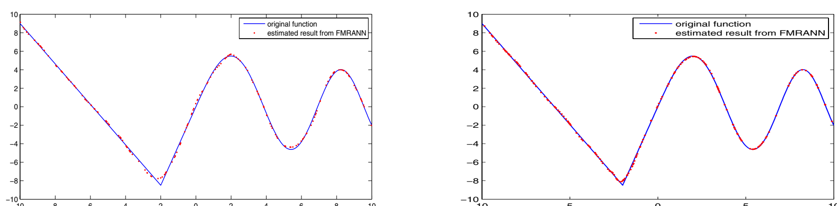

Fig. 2. For $f_1(x)$: at left is a comparison between the original function (solid line) and the estimated result (dotted point) from MRNN with the maximal performance index 0.040297 during the nine simulations, at the right is the comparison with the minimal performance index 0.021052

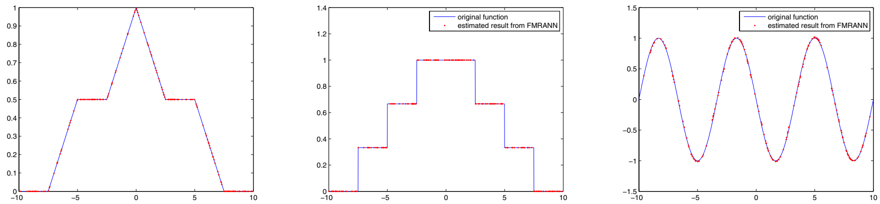

Fig. 3. the comparison between the original function (solid line) and the estimated result (dotted point) from MRNN: at the left is the comparison of $f_2(x)$ with the minimal performance index 8.6695e-005 during the nine simulations, at the middle is the comparison of $f_3(x)$ with the maximal performance index 0.06060172, at the right is the comparison of $f_4(x)$ with the maximal performance index 0.023313

References

1. Q. H. Zhang and A. Benveniste, Wavelet networks, IEEE Trans. Neural Netw., vol. 3, no. 6, pp. 889-898, Nov. 1992.
2. Y. C. Pati and P. S. Krishnaprasad, Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformation, IEEE Trans. Neural Networks, vol. 4, pp. 73-85, 1993.
3. J. Zhang, G. G. Walter, and W. N. W. Lee, Wavelet neural networks for function learning, IEEE Trans. Signal Processing, vol. 43, pp. 1485-1497, June 1995.
4. Q. Zhang, Using wavelet networks in nonparametric estimation, IEEE Trans. Neural Networks, vol. 8, pp. 227-236, Mar. 1997. Beijing, P.R. China, June 1999, pp. 451-456.
5. F. Alonge, F. Dippolito, S. Mantione, and F. M. Raimondi, A new method for optimal synthesis of wavelet-based neural networks suitable for identification purposes, in Proc. 14th IFAC, Beijing, P.R. China, June 1999, pp. 445-450.
6. X. Li, Z. Wang, L. Xu, and J. Liu, Combined construction of wavelet neural networks for nonlinear system modeling, in Proc. 14th IFAC, Beijing, P.R. China, June 1999, pp. 451-456.
7. J. Chen and D. D. Bruns, WaveARX neural network development for system identification using a systematic design synthesis, Ind. Eng. Chem. Res., vol. 34, pp. 4420-4435, 1995.
8. R. C. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in Proc. 6th Int. Symp. Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39-43.
9. R. C. Eberhart, P. Simpson, and R. Dobbins, Computational Intelligence PC Tools: Academic, 1996, ch. 6, pp. 212-226.
10. R. C. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, Evolutionary Prograinniing VII: Proc. 71h Ann. Conf. on Evolutionary Prograriirnirig Conf., San Diego, CA. Berlin: Springer-Verlag 1998.
11. Dipankar Dusgupta, Artificial Immnue Systems and their a pplications, Springer-Verlag, Berlin Heidelberg, 1999.
12. J. Chen and D. D. Bruns, WaveARX neural network development for system identification using a systematic design synthesis, Ind. Eng. Chem. Res., vol. 34, pp. 4420-4435, 1995.