Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 69 (2006) 449–465
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Time-series prediction using a local linear wavelet neural network

Yuehui Chen[a,*], Bo Yang[a,b], Jiwen Dong[a,b]

[a]*School of Information Science and Engineering, Jinan University, 250022 Jinan, PR China*
[b]*State Key Laboratory of Advanced Technology for Materials Synthesis and Processing, Wuhan University of Science and Technology, Wuhan, PR China*

Received 11 July 2004; received in revised form 25 January 2005; accepted 7 February 2005
Available online 19 April 2005
Communicated by T. Heskes

Abstract

A local linear wavelet neural network (LLWNN) is presented in this paper. The difference of the network with conventional wavelet neural network (WNN) is that the connection weights between the hidden layer and output layer of conventional WNN are replaced by a local linear model. A hybrid training algorithm of particle swarm optimization (PSO) with diversity learning and gradient descent method is introduced for training the LLWNN. Simulation results for the prediction of time-series show the feasibility and effectiveness of the proposed method.

Keywords: Local linear wavelet neural network; Particle swarm optimization algorithm; Gradient descent algorithm; Time-series prediction

1. Introduction

Recently, in stead of using common sigmoid activation functions, the wavelet neural network (WNN) employing nonlinear wavelet basis functions (named

*Corresponding author.
E-mail addresses: yhchen@ujn.edu.cn (Y. Chen), yangbo@ujn.edu.cn (B. Yang), csmaster@ujn.edu.cn (J. Dong).

0925-2312/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.02.006

wavelets), which are localized in both the time space and frequency space, has been developed as an alternative approach to nonlinear fitting problem [31,36]. Two key problems in designing of WNN are how to determine WNN architecture and what learning algorithm can be effectively used for training the WNN [2]. These problems are related to determine an optimal WNN architecture, to arrange the windows of wavelets, and to find the proper orthogonal or nonorthogonal wavelet basis. Curse of dimensionality is a mainly unsolved problem in WNN theory which brings some difficulties in applying a WNN to high-dimension problems.

The basis function neural networks are a class of neural networks, in which the output of the network is a weighted sum of a number of basis functions. The usually used basis functions include Gaussian radial basis functions, B-spline basis functions, wavelet basis functions and some neurofuzzy basis functions [3,12].

The particle swarm optimization (PSO) is a population based optimization method first proposed by Kennedy and Eberhart [14]. Some of the attractive features of the PSO include ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems. Some example applications include neural network training [7,28,29,6] and function minimization [23,24].

Time-series forecasting is an important research and application area. Much effort has been devoted over the past several decades to the development and improvement of time series forecasting models. Well established time series models include: (1) linear models, e.g., moving average, exponential smoothing and the autoregressive integrated moving average (ARIMA); (2) nonlinear models, e.g., neural network models and fuzzy system models [15,19]; and (3) the combination of linear and nonlinear models [35].

In this paper, a local linear wavelet neural network (LLWNN) is proposed, in which the connection weights between the hidden layer units and output units are replaced by a local linear model. The usually used learning algorithm for WNN is gradient descent method. But its disadvantages are slow convergence speed and easy stay at local minimum. A combination approach of PSO with adaptive diversity learning and gradient descent method is proposed for training the LLWNN. Simulation results for time-series prediction problems show the effectiveness of the proposed method. The main contributions of this paper are (1) the LLWNN providing a more parsimonious interpolation in high-dimension spaces when modelling samples are sparse; (2) a novel hybrid training algorithm for WNN and LLWNN was proposed.

The paper is organized as follows. The LLWNN is introduced in Section 2. A hybrid learning algorithm for training LLWNN is described in Section 3. The experiments on time-series prediction problems are given in Section 4. A short discussion is given in Section 5. Finally, concluding remarks are derived in the last section.

2. Local linear wavelet neural network

In terms of wavelet transformation theory, wavelets in the following form:

$$\Psi = \{\Psi_i = |\mathbf{a_i}|^{-1/2}\psi\left(\frac{\mathbf{x}-\mathbf{b_i}}{\mathbf{a_i}}\right): \mathbf{a_i}, \mathbf{b_i} \in R^n, i \in Z\}, \tag{1}$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n),$$

$$\mathbf{a_i} = (a_{i1}, a_{i2}, \ldots, a_{in}),$$

$$\mathbf{b_i} = (b_{i1}, b_{i2}, \ldots, b_{in}),$$

are a family of functions generated from one single function $\psi(x)$ by the operation of dilation and translation. $\psi(x)$, which is localized in both the time space and the frequency space, is called a mother wavelet and the parameters a_i and b_i are named the scale and translation parameters, respectively. The $\mathbf{x}$ represents inputs to the WNN model.

In the standard form of WNN, the output of a WNN is given by

$$f(x) = \sum_{i=1}^{M} \omega_i \Psi_i(x) = \sum_{i=1}^{M} \omega_i |a_i|^{-1/2}\psi\left(\frac{x-b_i}{a_i}\right), \tag{2}$$

where ψ_i is the wavelet activation function of ith unit of the hidden layer and ω_i is the weight connecting the ith unit of the hidden layer to the output layer unit. Note that for the n-dimensional input space, the multivariate wavelet basis function can be calculated by the tensor product of n single wavelet basis functions as follows

$$\psi(x) = \prod_{i=1}^{n} \psi(x_i). \tag{3}$$

Obviously, the localization of the ith units of the hidden layer is determined by the scale parameter a_i and the translation parameter b_i. According to the previous researches, the two parameters can either be predetermined based upon the wavelet transformation theory or be determined by a training algorithm. Note that the above WNN is a kind of basis function neural network in the sense of that the wavelets consists of the basis functions.

Note that an intrinsic feature of the basis function networks is the localized activation of the hidden layer units, so that the connection weights associated with the units can be viewed as locally accurate piecewise constant models whose validity for a given input is indicated by the activation functions. Compared to the multilayer perceptron neural network, this local capacity provides some advantages such as the learning efficiency and the structure transparency. However, the problem of basis function networks is also led by it. Due to the crudeness of the local approximation, a large number of basis function units have to be employed to approximate a given system. A shortcoming of the WNN is that for higher dimensional problems many hidden layer units are needed.

In order to take advantage of the local capacity of the wavelet basis functions while not having too many hidden units, here we propose an alternative type of WNN. The architecture of the proposed LLWNN is shown in Fig. 1. Its output in the output layer is given by

$$y = \sum_{i=1}^{M} (\omega_{i0} + \omega_{i1}x_1 + \cdots + \omega_{in}x_n)\Psi_i(x)$$
$$= \sum_{i=1}^{M} (\omega_{i0} + \omega_{i1}x_1 + \cdots + \omega_{in}x_n)|a_i|^{-1/2}\psi\left(\frac{x - b_i}{a_i}\right), \quad (4)$$

where $x = [x_1, x_2, \ldots, x_n]$. Instead of the straightforward weight ω_i (piecewise constant model), a linear model

$$v_i = \omega_{i0} + \omega_{i1}x_1 + \cdots + \omega_{in}x_n \quad (5)$$

is introduced. The activities of the linear models v_i $(i = 1, 2, \ldots, M)$ are determined by the associated locally active wavelet functions $\psi_i(x)$ $(i = 1, 2, \ldots, M)$, thus v_i is only locally significant. The motivations for introducing the local linear models into a WNN are as follows: (1) local linear models have been studied in some neuro-fuzzy systems and shown good performances [9,8]; and (2) local linear models should provide a more parsimonious interpolation in high-dimension spaces when modelling samples are sparse.

The scale and translation parameters and local linear model parameters are randomly initialized at the beginning and are optimized by a hybrid learning algorithm discussed in the following section.

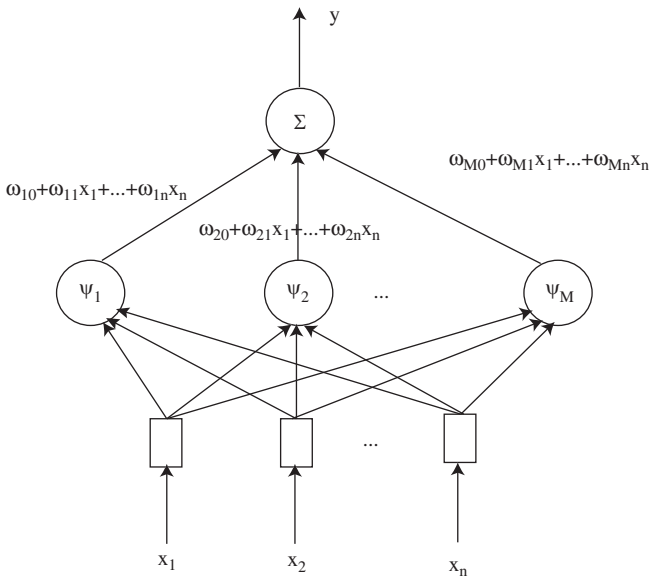

Fig. 1. A local linear wavelet neural network.

3. A hybrid learning algorithm

3.1. The basic PSO model

The PSO [14,34] conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\mathbf{z}_i$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\mathbf{v}_i$. At each time step, a function f_i representing a quality measure is calculated by using $\mathbf{z}_i$ as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector $\mathbf{p}_\mathrm{i}$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\mathbf{p}_\mathrm{g}$. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle.

At each time step t, by using the individual best position, $\mathbf{p}_\mathrm{i}$, and the global best position, $\mathbf{p}_\mathrm{g}(t)$, a new velocity for particle i is updated by [13]

$$\mathbf{v}_i(t+1) = \chi(\alpha\mathbf{v}_i(t) + c_1\phi_1(\mathbf{p}_\mathrm{i}(t) - \mathbf{z}_i(t)) + c_2\phi_2(\mathbf{p}_\mathrm{g}(t) - \mathbf{z}_i(t))), \tag{6}$$

where χ and α are real numbers. The parameter χ controls the magnitude of $\mathbf{v}$, whereas the inertia weight α weights the magnitude of the old velocity $\mathbf{v}_i(t)$ in the calculation of the new velocity $\mathbf{v}_i(t+1)$. c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. Changing velocity this way enables the particle i to search around its individual best position, $\mathbf{p}_\mathrm{i}$, and global best position, $\mathbf{p}_\mathrm{g}$. Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathbf{z}_i(t+1) = \mathbf{z}_i(t) + \mathbf{v}_i(t+1). \tag{7}$$

The parameter $\mathbf{z}_i$ enhances searching ability by controlling the balance between local and global exploration in the problem search space both for EA and PSO. In the next subsection, a detailed method for improving PSO search ability with a diversity search strategy is introduced.

3.2. ADLPSO: PSO model with adaptive diversity learning

The main motivation for improving the particles with diversity operator in search space is that the particles in the basic PSO tend to cluster too closely. When an optimum (local or global) is found by one particle the other particles will be drawn towards it. If all particles end up in this optimum, they will stay at this optimum without much chance to escape. This simply happens because of the way the basic PSO works. If the identified optimum is only local it would be advantageous to let some of the particles explore other areas of the search space to fine turning the solution.

In our ADLPSO model, we tried to increase the diversity in each generation by using a specific probability density function (PDF):

$$p(x) = \begin{cases} (1-q)\beta e^{\beta x} & \text{if } x \leqslant 0, \\ q\beta e^{-\beta x} & \text{if } x > 0, \end{cases} \tag{8}$$

where adjustable parameters $q \in [0,1]$ and β are used to control the range and direction of the diversification search. Two example graphs of the PDF are shown in Fig. 2 (left and right). It can be seen that the smaller the β, the larger the local search range; the larger the q, the higher the search probability in positive direction. $q = 0.5$ means that there is same search probability in positive and in negative direction.

By solving the above PDF inversely, a random diversity search vector $\mathbf{d} = [d_1, d_2, \ldots, d_N]$, which belongs to the proposed PDF in Eq. (8), can be obtained as follows:

$$d_i = \begin{cases} \dfrac{1}{\beta}\ln\left(\dfrac{r_i}{1-q_i}\right) & \text{if}\left(0 < r_i \leqslant 1-q_i\right), \\ -\dfrac{1}{\beta}\ln\left(\dfrac{1-r_i}{q_i}\right) & \text{if}\left(1-q_i \leqslant r_i < 1\right), \end{cases} \tag{9}$$

where r_i is a random real number uniformly distributed at [0,1], $q_i = 0.5$ for our experiments, $i = 1, 2, \ldots, N$ and N is the number of particles whose turning strategy is following the diversity rule.

In each generation, after the fitness values of all the particles are calculated, the top 70% best-forming ones are marked and form the first group. Another 30% particles will enhance themselves by using diversity rule as follows:

$$\mathbf{z}(t+1) = \mathbf{z}(t) + \mathbf{d}(t+1), \tag{10}$$

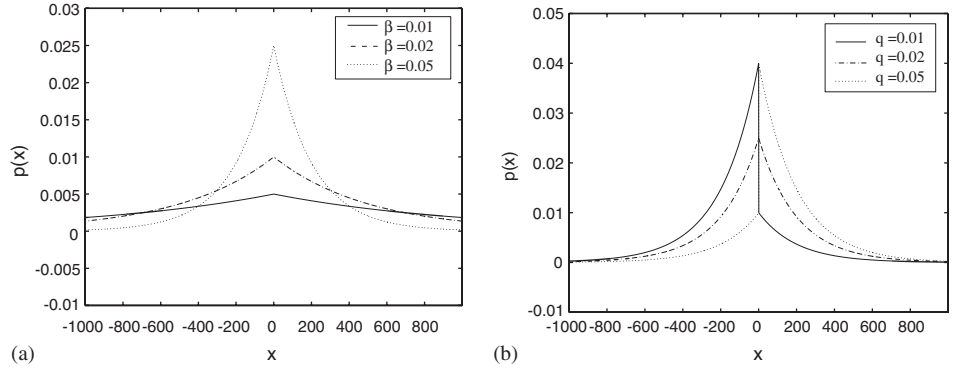

Fig. 2. A specific probability distribution function in which the shape of the function depends on the parameters β and q. (a) The larger the β, the smaller the local search range. (b) The larger the q, the higher the search probability in positive direction.

where $\mathbf{z}(t+1)$ represents the model free parameters at generation $t+1$ and each element of the vector $\mathbf{d}(t+1)$ takes the same form as shown in Eq. (9). The particles in first group will enhance themselves based on their own private cognition and global social interactions with each other (this means that the global best position, $\mathbf{p}_g$, is taken from the total population), and evolve by using Eqs. (6) and (7). It should be noted that the control parameter β in Eq. (9) is adjusted according to the search process with iteration steps as follows:

$$\beta(t+1) = \frac{(\beta(t) - \beta(0))(MAXITER\text{-}iter)}{MAXITER} + \beta(0). \tag{11}$$

The process is shown in Fig. 3, in which with the decrease of β in the initial stage of the search, the local search range becomes larger quickly and then it stayed with a fixed value.

3.3. Combination of ADLPSO and gradient descent algorithms for training LLWNN

Gradient descent-based training algorithms have shown their effectiveness in previous studies on WNN. However, problems such as local minimum, sensitivity to initial conditions and unstable property are still remained due to the nature of the gradient descent [36]. Evolutionary algorithms including PSO are global search algorithms but suffer from fine-tuning inefficiency [32].

To take advantage of gradient descent and ADLPSO methods to train the proposed LLWNN, an alternative training method is proposed in this study. The ADLPSO algorithm is used first to locate a good region in the search space and

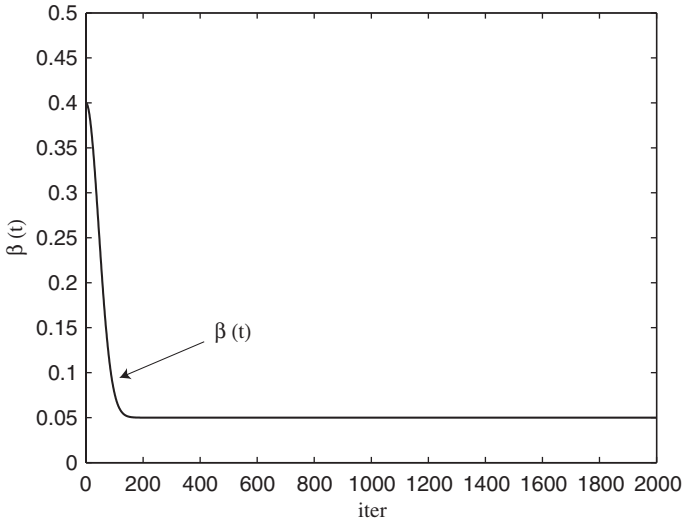

Fig. 3. The variation process of the control parameter β with maximum iteration number 2000 and $\beta(0) = 0.05$, $\beta(1) = 0.4$.

then a gradient descent search algorithm is employed to fine turn the optimal solution.

Before describing details of the algorithm for training LLWNN, the issue of coding is presented. Coding concerns the way the weights, dilation and translation parameters of LLWNN are represented by individuals or particles. A float point coding scheme is adopted here. For LLWNN coding, suppose there are M nodes in hidden layer and n input variables, then the total number of parameters to be coded is $(2n+n+1)*M=(3n+1)M$. The coding of a LLWNN into an individual or particle is as follows:

$$|a_{11}b_{11}\ldots a_{1n}b_{1n}\omega_{10}\omega_{11}\ldots\omega_{1n}|a_{21}b_{21}\ldots a_{2n}b_{2n}\omega_{20}\omega_{21}\ldots\omega_{2n}|\ldots$$
$$|a_{n1}b_{n1}\ldots a_{nn}b_{nn}\omega_{n0}\omega_{n1}\ldots\omega_{nn}|$$

The simple loop of the proposed training algorithm for LLWNN is as follows.

*S*1: Initialization. Initial population is generated randomly. The learning parameters, such as c_1, c_2, $\beta(0)$ and $\beta(1)$ in ADLPSO, and learning rate and momentum in BP should be assigned in advance.

*S*2: Parameter optimization with ADLPSO algorithm.

*S*3: If maximum number of generations is reached or no better parameter vector is found for a significantly long time (100 steps), then go to step S4; otherwise goto step S2.

*S*4: Parameter optimization with gradient descent algorithm.

*S*5: If the satisfactory solution is found then stop; otherwise goto step S4.

4. Experiments

The developed LLWNN model is applied here in conjunction with two time-series prediction problems: the Box–Jenkins time series and the Mackey–Glass time series. Well-known benchmark examples are used for the sake of an easy comparison with existing models.

In this work, the mother wavelet (Eq. (12)) is used for both experiments.

$$\psi(x)=-x\exp\left(-\frac{x^2}{2}\right). \tag{12}$$

The objective function used is the root mean square error (*RMSE*),

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_1^i-y_2^i)^2}, \tag{13}$$

where y_1^i and y_2^i denote the target output and model output, respectively.

The parameters used for both experiments in optimization of LLWNN and WNN are listed in Table 1. In addition, all experiments were performed using an 2.0 GHz processor with 512 MB of RAM.

Table 1
Parameter settings

Population size	50
c_1 and c_2	2.0
α	0.7
χ	0.8
$\beta(0)$	0.05
$\beta(1)$	0.4
Learning rate	0.05
Momentum term	0.75

4.1. Application to Box–Jenkins time series

In this section, the proposed LLWNN model is applied to the gas furnace data (series J) prediction problem [1]. The data set was recorded from a combustion process of a methane–air mixture. It is well known and frequently used as a benchmark example for testing identification and prediction algorithms. The data set consists of 296 pairs of input–output measurements. The input $u(t)$ is the gas flow into the furnace and the output $y(t)$ is the CO_2 concentration in outlet gas. The sampling interval is 9 s.

Following previous researchers [16] in order to make a meaningful comparison, the inputs of the prediction model are selected as $u(t-4)$ and $y(t-1)$, and the output is $y(t)$.

We simulated the following four cases: (1) the gradient descent algorithm was employed to train a WNN model with network architecture {2-8-1}; (2) the proposed hybrid learning algorithm was employed to train a WNN model with same network architecture; (3) the gradient descent algorithm (with momentum) was employed to train a LLWNN model with network architecture {2-8-1}; and (4) the proposed hybrid learning algorithm was employed to train a LLWNN model with same network architecture. For each of the cases, the data were partitioned in 200 data points as a training set, and the remaining 92 points as a test set for testing the performance of the evolved model. In order to remove the effects of the initial values of free parameters on the final results, for each of the cases, 20 experiments (runs) were performed with randomly set initial parameters. Each of the models was trained for 3000 epochs. Over 20 runs, the average RMSE for cases 1, 2, 3, and 4 on test data set are 0.048, 0.042, 0.034 and 0.025, respectively. The maximum and minimum deviations in RMSE are 0.033 and 0.014. For the best results, a comparison of LLWNN and WNN with gradient descent algorithm and the new hybrid technique is shown in Table 2. Table 3 shows the comparison of test results of different models for Box–Jenkins data prediction problem. A comparison has been made to show the actual time-series, the output of the best LLWNN model and the prediction error using the hybrid training algorithm for training and test data sets (see Fig. 4). Testing the methods with different number of hidden units and comparing only the best results got in cases of LLWNN and WNN are shown in Fig. 5.

Table 2
Comparison of LLWNN and WNN for Box–Jenkins time-series prediction

Model	Structure	Para.	Training time (s)	RMSE training	RMSE testing
WNN+gradient	2-8-1	40	134	0.08831	0.09000
WNN+hybrid	2-8-1	40	107	0.08485	0.08831
LLWNN+gradient	2-8-1	56	153	0.01581	0.01643
LLWNN+hybrid	2-8-1	56	125	0.01095	0.013784

Table 3
Comparison of test results of different models for Box–Jenkins data prediction problem

Model name	Inputs	RMSE
ARMA [1]	5	0.843
Tong's model [27]	2	0.685
Pedrycz's model [21]	2	0.566
Xu's model [33]	2	0.573
Sugeno's model [25]	2	0.596
Surmann's model [26]	2	0.400
Lee's model [17]	2	0.638
Lin's model [18]	5	0.511
Nie's model [20]	4	0.412
ANFIS model [10]	2	0.085
FuNN model [11]	2	0.071
HyFIS model [16]	2	0.042
Neural tree model [4]	2	0.026
WNN+gradient	2	0.084
WNN+hybrid	2	0.081
LLWNN+gradient	2	0.017
LLWNN+hybrid	2	0.013

It is clear that the LLWNN has more accuracy than the conventional WNN though it holds a few more free parameters. Meanwhile, the simulation results demonstrated that the new hybrid training algorithm is efficient than the conventional gradient learning algorithm.

4.2. Application to Mackey–Glass time-series

The chaotic Mackey–Glass differential delay equation is recognized as a benchmark problem that has been used and reported by a number of researchers for comparing the learning and generalization ability of different models. The Mackey–Glass chaotic time series is generated from the following equation:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t), \tag{14}$$

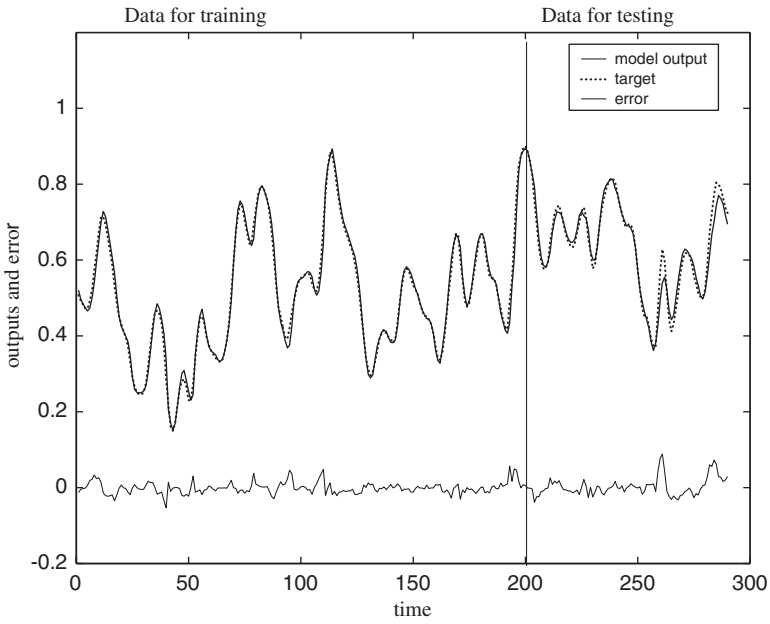

Fig. 4. The time series data, output of the LLWNN and the prediction error for training and test samples using the hybrid training algorithm.

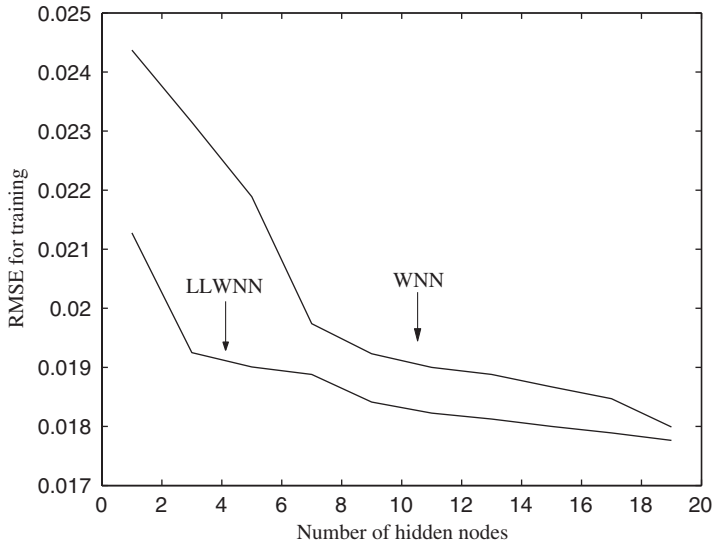

Fig. 5. The performance of LLWNN and WNN in RMSE for training data set with different hidden units.

where $\tau > 17$; the equation shows chaotic behavior. To make the comparison with earlier work fair, we predict the $x(t+6)$ using the input variables $x(t)$, $x(t-6)$, $x(t-12)$ and $x(t-18)$, respectively.

We simulated the following four cases: (1) the gradient descent algorithm was employed to train a WNN model with network architecture {4-10-1}; (2) the proposed hybrid learning algorithm was employed to train a WNN model with same network architecture; (3) the gradient descent algorithm (with momentum) was employed to train a LLWNN model with network architecture {4-10-1}; and (4) the proposed hybrid learning algorithm was employed to train a LLWNN model with same network architecture. For each of the cases, 1000 sample points are used in our study. The first 500 data pairs of the series were used as training data, while the remaining 500 were used to validate the model identified. In order to remove the effects of the initial values of free parameters on the final results, for each of the cases, 20 experiments (runs) were performed with randomly set initial parameters. Each of the models was trained for 3000 epochs. Over 20 runs, the average RMSE for cases 1, 2, 3, and 4 on test data set are 0.0089, 0.071, 0.049 and 0.042, respectively. The maximum and minimum deviations in RMSE are 0.0028 and 0.012. For the best results, a comparison of LLWNN and WNN with gradient descent algorithm and the new hybrid technique is shown in Table 4. Table 5 shows the comparison of test results of different models for the chaotic Mackey–Glass prediction problem. A comparison has been made to show the actual time-series, the output of the best LLWNN model and the prediction error using the hybrid training algorithm for training and test data sets (see Fig. 6). Testing the methods with different number of hidden units and comparing only the best results got in cases of LLWNN and WNN are shown in Fig. 7.

It is also clear that the LLWNN has more accuracy than the conventional WNN though it holds a few more free parameters. Meanwhile, the simulation results demonstrated that the new hybrid training algorithm is efficient than the conventional gradient learning algorithm.

From above simulation results, it can be seen that the proposed LLWNN model with new hybrid technique works well for generating prediction models of time series.

Table 4
Comparison of LLWNN and WNN for Mackey–Glass time-series prediction

Model	Structure	Para.	Training time (s)	RMSE training	RMSE testing
WNN+gradient	4-10-1	90	168	0.0067	0.0071
WNN+hybrid	4-10-1	90	157	0.0056	0.0059
LLWNN+gradient	4-10-1	110	123	0.0038	0.0041
LLWNN+hybrid	4-10-1	110	114	0.0033	0.0036

Table 5
Comparison of test results of different models for the Mackey–Glass time-series problem

Method	Prediction error (RMSE)
Auto-regressive model	0.19
Cascade correlation NN	0.06
Back-propagation NN	0.02
Sixth-order polynomial	0.04
Linear prediction method	0.55
ANFIS and fuzzy system [10]	0.007
Wang et al. [30] Product T-norm	0.0907
Classical RBF (with 23 neurons) [5]	0.0114
PG-RBF network [22]	0.0028
Genetic algorithm and fuzzy system [15]	0.049
Neural tree [4]	0.0069
WNN+gradient	0.0071
WNN+hybrid	0.0059
LLWNN+gradient	0.0041
LLWNN+hybrid	0.0036

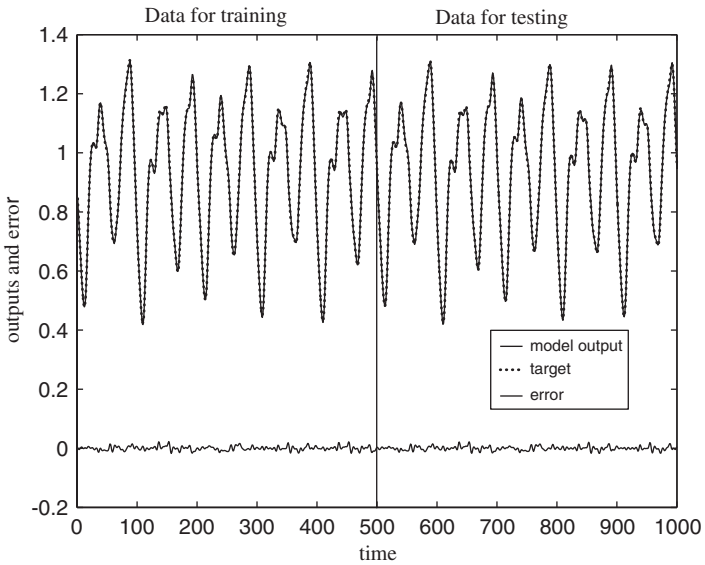

Fig. 6. The actual time series data, output of the LLWNN model and the prediction error for training and test samples using the hybrid training algorithm.

5. Discussion

With various wavelets used as activation functions and gradient descent based training algorithms, the success has been demonstrated by the previous studies in

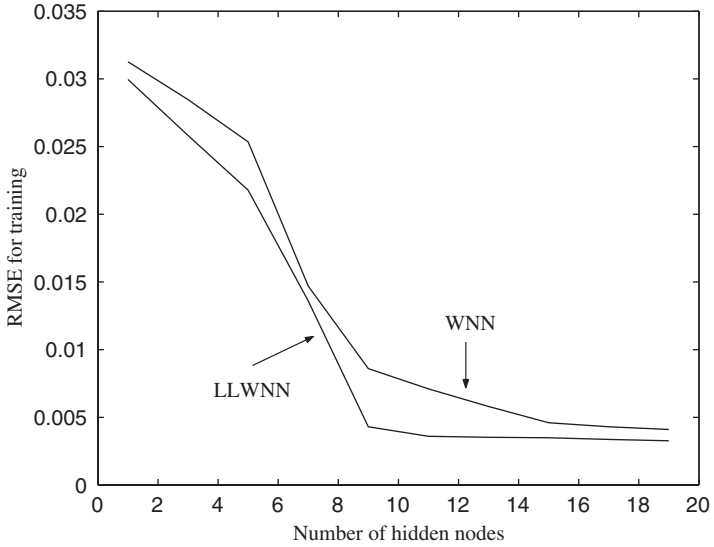

Fig. 7. The performance of LLWNN and WNN in RMSE for training data set with different hidden units.

conventional WNN [36]. However, a large number of basis function units should be carefully determined for a high dimensional nonlinear problem.

The proposed LLWNN experiments demonstrate that only a few of wavelet basis functions is needed for a given approximation/prediction problem with sufficient accuracy. This is because the local linear models provide more power than a constant weight. Moreover, the dilation and translation parameters of LLWNN are randomly generated and optimized without predetermination.

The hybrid training algorithm of the ADLPSO and gradient descent method made the parameter searching process more effective. Namely, the ADLPSO can quickly and globally position the global minimum of objective function and then the gradient descent algorithm gives accurate position of the minimum.

6. Conclusion

In this paper, a LLWNN was proposed. The characteristic of the network is that the straightforward weight is replaced by a local linear model. The working process of the proposed network can be viewed as to decompose the complex, nonlinear system into a set of locally active submodels, then smoothly integrate those submodels by their associated wavelet basis functions. One advantage of the proposed method is that it needs only smaller wavelets for a given problem than the common WNN. A fast and hybrid training algorithm ADLPSO is also introduced

for training the LLWNN. Simulation results for time-series prediction problem showed the effectiveness of the proposed approach.

Acknowledgements

This research was partially supported the National High Technology Development Program of China (863 Program) under contract number 2002AA4Z3240, and The Provincial Science and Technology Development Program of Shandong under Contract no SDSP2004-0720-03.

References

[1] G.E.P. Box, Time series analysis, forecasting and control, San Francisco, Holden Day, 1970.
[2] Y.H. Chen, et al., Evolving wavelet neural networks for system identification, Proceeding of International Conference on Electrical Engineering, 2000, pp. 279–282.
[3] Y.H. Chen, et al., Evolving the basis function neural networks for system identification, Int. J. Adv. Comput. Intell. 5 (4) (2001) 229–238.
[4] Y.H. Chen, et al., Nonlinear system modelling via optimal design of neural trees, Int. J. Neural Sys. 14 (2) (2004) 125–137.
[5] K.B. Cho, et al., Radial basis function based adaptive fuzzy systems their application to system identification and prediction, Fuzzy Sets Syst. 83 (1995) 325–339.
[6] R.C. Eberhart, X. Hu, Human tremor analysis using particle swarm optimization, Proceedings of the Congress on Evolutionary Computation, 1999, pp. 1927–1930.
[7] A.P. Engelbrecht, A. Ismail, Training product unit neural networks, Stability Control: Theory Appl. 2 (1–2) (1999) 59–74.
[8] B. Fischer, O. Nelles, R. Isermann, Adaptive predictive control of a heat exchanger based on a fuzzy model, Control Eng. Practice 6 (1998) 259–269.
[9] B. Foss, T.A. Johansen, On local and fuzzy modelling, Proceedings of 3rd International Industrial Fuzzy Control and Intelligent Systems, 1993, pp. 134–139.
[10] J.S.R. Jang, et al., Neuro-fuzzy and Soft Computing: a Computational Approach to Learning and Machine Intelligence, Prentice-Hall, Upper Saddle River, NJ, 1997.
[11] N.K. Kasabov, et al., FuNN/2-A fuzzy neural network architecture for adaptive learning and knowledge acquisition, Inform. Sci. 101 (1997) 155–175.
[12] S. Kawaji, Y.H. Chen, Evolving Neurofuzzy system by hybrid soft computing approaches for system identification, Int. J. Adv. Comput. Intel. 5 (4) (2001) 229–238.
[13] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1999, pp. 1931–1938.
[14] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proc. IEEE Int. Conf. Neural Networks IV (1995) 1942–1948.
[15] D. Kim, et al., Forecasting time series with genetic fuzzy predictor ensembles, IEEE Trans. Fuzzy Syst. 5 (1997) 523–535.
[16] J. Kim, et al., HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems, Neural Networks 12 (1999) 1301–1319.
[17] C.C. Lee, et al., A combined approach to fuzzy model identification, IEEE Trans. Syst. Man Cybernet. 24 (1994) 736–744.
[18] Y. Lin, et al., A new approach to fuzzy-neural system modelling, IEEE Trans. Fuzzy Syst. 3 (1995) 190–198.

[19] K.S. Narendra, et al., Adaptive control using neural networks and approximation models, IEEE Trans. Neural Networks 8 (3) (1997) 475–485.
[20] J. Nie, Constructing fuzzy model by self-organising counterpropagation network, IEEE Trans. Syst. Man Cybernet. 25 (1995) 963–970.
[21] W. Pedtycz, An identification algorithm in fuzzy relational systems, Fuzzy Sets Syst. 13 (1984) 153–167.
[22] I. Rojas, et al., Time series analysis using normalized PG-RBF network with regression weights, Neurocomputing 42 (2002) 167–285.
[23] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, IEEE International Conference of Evolutionary Computation, May 1998, pp. 367–372.
[24] Y. Shi, R.C. Eberhart. Empirical study of particle swarm optimization, Proceedings of the Congress on Evolutionary Computation, 1999, pp. 1945–1949.
[25] M. Sugeno, et al., Linguistic modelling based on numerical data, Proceedings of the IFSA'91, 1991, pp. 234–247.
[26] H. Surmann, et al., Self-organising and genetic algorithm for an automatic design of fuzzy control and decision systems, Proceedings of the FUFIT's 93, 1993, pp. 1079–1104.
[27] R.M. Tong, The evaluation of fuzzy models derived from experimental data, Fuzzy Sets Syst. 4 (1980) 1–12.
[28] F. Van den Berg, Particle swarm weight initialization in multi-layer perceptron artificial neural networks, in: D. Sha (ed.), Development and Practice of AI Techniques, Proceedings of the ICAI i '99, International Conference on Artificial Intelligence, Durban, September 1999, pp. 41–45.
[29] F. Van den Berg, A.P. Engelbrecht, Cooperative learning in neural networks using particle swarm optimizers, S. Afr. Comp. J. (2000), pp. 84–90.
[30] L.X. Wang, et al., Generating fuzzy rules by learning from examples, IEEE Trans. Syst. Man Cybernet. 22 (1992) 1414–1427.
[31] T. Wang, et al., A wavelet neural network for the approximation of nonlinear multivariable function, Trans. Inst. Electr. Eng. C 102-C (2000) 185–193.
[32] Y. Xin, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423–1447.
[33] C.W. Xu, Fuzzy model identification and self-learning for dynamic systems, IEEE Trans. Syst. Man Cybernet. 17 (1987) 683–689.
[34] H. Yoshida, et al., A particle swarm optimization for reactive power and voltage control considering voltage security assessment, IEEE Trans. Power Syst. 15 (4) (2000).
[35] G.P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing 50 (2003) 159–175.
[36] Q. Zhang, A. Benveniste, Wavelet Networks, IEEE Trans. Neural Networks 3 (6) (1992) 889–898.

Yuehui Chen was born in 1964. He received his B.Sc. degree in mathematics/automatics from the Shandong University of China in 1985, and Ph.D. degree in electrical engineering from the Kumamoto University of Japan in 2001. During 2001–2003, he had worked as the Senior Researcher of the Memory-Tech Corporation at Tokyo. Since 2003 he has been a member at the Faculty of Electrical Engineering in Jinan University, where he is currently head of the Laboratory of Computational Intelligence. His research interests include evolutionary computation, neural networks, fuzzy systems, hybrid computational intelligence and their applications in time-series prediction, system identification and intelligent control. He is the author and co-author of more than 60 papers. Professor Yuehui Chen is a member of IEEE, the IEEE Systems, Man and Cybernetics Society and the Computational Intelligence Society. He is also a member of the editorial boards of several technical journals and a member of the program committee of several international conferences.

Bo Yang is a professor and vice-president of Jinan University, Jinan, China. He is the Director of the Provincial Key Laboratory of Information and Control Engineering, and also acts as the Associate Director of Shandong Computer Federation, and Member of the Technical Committee of Intelligent Control of Chinese Association of Automation. His main research interests include computer networks, artificial intelligence, machine learning, knowledge discovery, and data mining. He has published numerous papers and gotten some of important scientific awards in this area.

Jiwen Dong received his B.E. and M.E. degrees in Computer Science and Automatics from the Wuhan University and the Wuhan University of Science and Technology, China, in 1985 and 1995, respectively. He is currently a Ph.D. student at the Wuhan University of Science and Technology. His research interests include neural networks, fuzzy systems, evolutionary algorithms and image processing. He is currently an associate professor and vice president of the School of Information Science and Engineering of Jinan University.