

Research Article

Initialization by a Novel Clustering for Wavelet Neural Network as Time Series Predictor

Rong Cheng,^{1,2} Hongping Hu,¹ Xiuhui Tan,^{1,2} and Yanping Bai¹

¹*School of Science, North University of China, Shanxi, Taiyuan 030051, China*

²*School of Information and Communication Engineering, North University of China, Shanxi, Taiyuan 030051, China*

Correspondence should be addressed to Yanping Bai; baiyp666@163.com

Received 22 September 2014; Revised 1 April 2015; Accepted 7 April 2015

Academic Editor: Cheng-Jian Lin

Copyright © 2015 Rong Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The architecture and parameter initialization of wavelet neural network are discussed and a novel initialization method is proposed. The new approach can be regarded as a dynamic clustering procedure which will derive the neuron number as well as the initial value of translation and dilation parameters according to the input patterns and the activating wavelets functions. Three simulation examples are given to examine the performance of our method as well as Zhang's heuristic initialization approach. The results show that the new approach not only can decide the WNN structure automatically, but also provides superior initial parameter values that make the optimization process more stable and quickly.

1. Introduction

An artificial neural network (ANN) is a highly parallel distributed network of connected processing units called neurons. Due to their fascinating characteristics of robustness, fault tolerance, adaptive learning ability, and massive parallel processing capabilities, ANNs possess the capability of learning from examples with both linear and nonlinear relationships between the input and output signals, which makes them a popular tool for time series prediction [1, 2], feature extraction [3, 4], pattern recognition [5, 6], and classification [7, 8]. However, ANNs have limited ability to characterize local features, such as discontinuities in curvature, jumps in value, or other edges.

Instead of using common sigmoid activation functions, the wavelet neural network (WNN) employing nonlinear wavelet basis functions [9, 10], which are localized in both the time space and frequency space, has been developed as an alternative approach to nonlinear fitting problem. It has been proven that families of wavelet frames are universal approximators [11], which give a theoretical basis to their use in the framework of function approximation and process modeling.

There are two different WNN architectures: one type has fixed wavelet bases possessing fixed dilation and translation

parameters (WNN-Type1). In this one only the output layer weights are adjustable. Another type has the variable wavelet base whose dilation and translation parameters and output layer weights are adjustable (WNN-Type2). Several WNN models have been proposed in the literatures. In [12], a four-layer self-constructing wavelet network (SCWN) controller for nonlinear systems control is described and the orthogonal wavelet functions are adopted as its node functions. In [13], a local linear wavelet neural network (LLWNN) is presented whose connection weights between the hidden layer and output layer of conventional WNN are replaced by a local linear model. In [14], a model of multiwavelet-based neural networks is proposed. The structure of this network is similar to that of the wavelet network, except that the orthonormal scaling functions are replaced by orthonormal multiscaling functions.

A time series is a sequence of observations taken sequentially in time [15]. Time series prediction 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 prediction models. Besides the well-known linear models such as moving average, exponential smoothing, and the autoregressive integrated moving average, nonlinear models including artificial neural

network, wavelet neural network, and fuzzy system models also become the well-established time series models. In this paper, the wavelet neural network (WNN) is used as the time series predictor, and the detailed research works are described subsequently.

We adopt WNN-Type2 with adjustable translation and dilation parameters and multiplication form of multidimensional wavelets as the nonlinear model for time series prediction in this paper. Key problems in designing of this type of WNN consist of determining WNN architecture, initializing the translation and dilation vectors, and choosing learning algorithm that can be effectively used for training the WNN. This study mainly focuses on the first two points. In the practical applications, the number of hidden neurons which determines the structure of the network is often set by experience or the time-consuming trial-and-error tests, and the initial values of parameters are often set randomly. Due to the rapidly vanishing property of wavelet functions, the random initialization scheme to the dilation and translation parameters may cause the wavelets' effective response regions out of interest which makes the learning performance very instable. So it is inadvisable to adopt random initialization scheme for dilations and translations in WNN. In [9], Zhang proposes a heuristic initialization procedure which considers the interesting domain of input patterns. But, in its implementation, the wavelet functions used in WNN are not considered, and the resolution reduced gradually according to an established rule which does not take full consideration of sample distribution.

In the present paper, inspired by the localization character of wavelet functions and considering the multiplication form of multidimensional wavelets in the hidden neuron for multi-variable inputs, we present a novel initialization approach by the help of a new clustering method for WNN. This approach can determine the unit number of hidden layer and initialize the translation and dilation vectors simultaneously. After performing the training process by gradient descent method, we can see that, besides the capability of neuron number determination, WNN with our initialization method gives more satisfactory and stable results for time series prediction compared to Zhang's heuristic initialization method which is used for this model in some literatures [9, 16, 17].

The paper is organized as follows. A brief review of wavelet and wavelet-based function approximation is given in Section 2, followed by the introduction of the architecture of wavelet neural network in Section 3. The detailed description of the clustering based initialization approach and the training algorithm are given in Sections 4 and 5. Three simulation experiments on time series prediction problems and the comparison results with Zhang's heuristic initialization method are presented in Section 6. Finally, some conclusions are drawn in the last section.

2. Wavelet-Based Function Approximation

Wavelets in the following form,

$$\psi_{a,b} = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right), \quad (a, b \in \mathbb{R}, a \neq 0), \quad (1)$$

are a family of functions generated from one single function $\psi(x)$ by the operation of dilation and translation. $\psi(x) \in L^2(\mathbb{R})$ is called a mother wavelet function that satisfies the admissibility condition:

$$C_\psi = \int_0^{+\infty} \frac{|\widehat{\psi}(\omega)|^2}{\omega} d\omega < +\infty, \quad (2)$$

where $\widehat{\psi}(\omega)$ is the Fourier transform of $\psi(x)$ [11, 18].

Grossmann and Morlet [19] proved that any function $f(x)$ in $L^2(\mathbb{R})$ can be represented by

$$f(x) = \frac{1}{C_\psi} \iint Wf(a, b) |a|^{-1/2} \psi\left(\frac{x-b}{a}\right) \frac{1}{a^2} da db, \quad (3)$$

where $Wf(a, b)$ given by

$$Wf(a, b) = |a|^{-1/2} \int_{-\infty}^{+\infty} \psi\left(\frac{x-b}{a}\right) f(x) dx \quad (4)$$

is the continuous wavelet transform of $f(x)$.

Superior to conventional Fourier transform, the wavelet transform (WT) in its continuous form provides a flexible time-frequency window, which narrows when observing high frequency phenomena and widens when analyzing low frequency behavior. Thus, time resolution becomes arbitrarily good at high frequencies, while the frequency resolution becomes arbitrarily good at low frequencies. This kind of analysis is suitable for signals composed of high frequency components with short duration and low frequency components with long duration, which is often the case in practical situations.

As the parameters a and b are the continuous values, the resulting continuous wavelet transform (CWT) is a very redundant representation and impracticable also. This impracticability is the result of the redundancy. Therefore, the scale and shift parameters are evaluated on a discrete grid of time-scale leading to a discrete set of continuous wavelet functions:

$$\psi_{a_i, b_i} = |a_i|^{-1/2} \psi\left(\frac{x-b_i}{a_i}\right), \quad a_i, b_i \in \mathbb{R}, a_i \neq 0. \quad (5)$$

The continuous inverse wavelet transform (3) is discretized as

$$f(x) = \sum_i w_i |a_i|^{-1/2} \psi\left(\frac{x-b_i}{a_i}\right). \quad (6)$$

If there exist two constants $c > 0$ and $C < +\infty$ such that, for any $f(x)$ in $L^2(\mathbb{R})$, the following inequalities hold:

$$c \|f\|^2 \leq \sum_i |\langle \psi_{a_i, b_i}, f \rangle|^2 \leq C \|f\|^2, \quad (7)$$

where $\|f\|$ denotes the norm of function $f(x)$ and $\langle f, g \rangle$ denotes the inner product of functions f and g , and the family $\{\psi_{a_i, b_i}\}$ is said to be a frame of $L^2(\mathbb{R})$. It has been proved that families of wavelet frames of $L^2(\mathbb{R})$ are universal approximators.

Inspired by the wavelet decomposition of $f(x) \in L^2(\mathbb{R})$ in (6) and a single hidden layer network model, Zhang and Benveniste [9] had developed a new neural network model, namely, wavelet neural network (WNN).

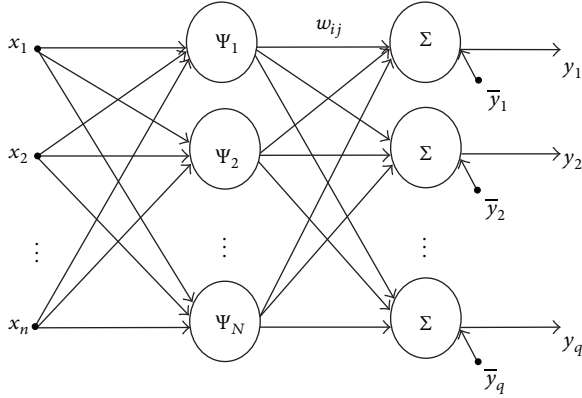


FIGURE 1: Architecture of wavelet neural network.

3. Architecture of Wavelet Neural Network

A brief review of wavelet decomposition theory has been given in Section 2, where functions with univariable were concerned. For the modeling of multivariable processes, multidimensional wavelets must be defined. In the present work, multidimensional wavelets are defined as the multiplication of single-dimensional wavelet functions:

$$\Psi_j(\mathbf{x}) = \psi\left(\frac{\mathbf{x} - \mathbf{b}_j}{\mathbf{a}_j}\right) = \prod_{k=1}^n \psi\left(\frac{x_k - b_{jk}}{a_{jk}}\right), \quad j = 1, 2, \dots, N, \quad (8)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is the input vector and $\mathbf{b}_j = (b_{j1}, b_{j2}, \dots, b_{jn})$ and $\mathbf{a}_j = (a_{j1}, a_{j2}, \dots, a_{jn})$ are the translation and dilation vectors, respectively.

Generalized from radial basis function neural network, WNN is in fact a feed-forward neural network with one hidden layer, wavelet functions as activation functions in the hidden nodes, and a linear output layer. As a result, the network output $\mathbf{y} = (y_1, y_2, \dots, y_q)^T$ is computed as

$$y_i = \sum_{j=1}^N w_{ij} \Psi_j + \bar{y}_i, \quad i = 1, 2, \dots, q, \quad (9)$$

where $\mathbf{w} = (w_{ij})$ and $\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_q)$ define the connecting weights and the bias terms between the hidden layer and the output layer, respectively. N is the number of units in hidden layer. These wavelet neurons are usually referred to as wavelons. The architecture of a WNN is illustrated in Figure 1.

4. Initialization Approach of Wavelet Neural Network

Before training the WNN, some factors should be determined in advance, which are the number of wavelons and initial value of parameters (a_{jk} , b_{jk} , w_{ij} , and \bar{y}_i). The former is fixed once the structure of network was determined, while the latter is adjusted by the training algorithm. All these factors are crucial for the performance of network in simulating the real model. In this section, a brief description of wavelet

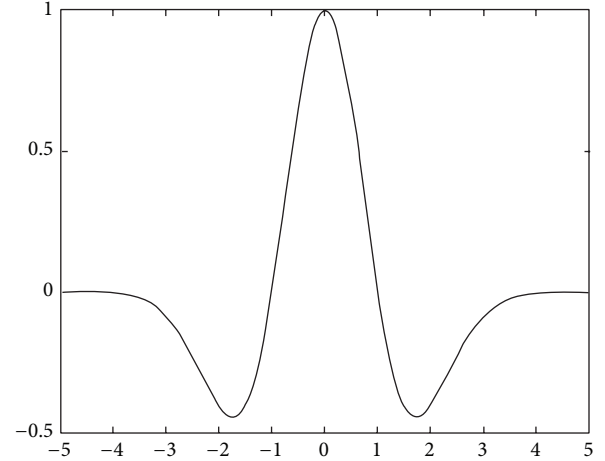


FIGURE 2: Mexican Hat wavelet.

window is presented firstly, and then a novel initialization method based on the dynamic clustering is proposed, which could provide the number of hidden neurons and the initial values of translation and dilation parameters at the same time.

4.1. Wavelet Window in Time Domain. A mother wavelet function $\psi(x)$ defined by (2) will have sufficient decay, which can be considered as “local response.” In other words, $\psi(x)$ is a window with center in μ and radius σ in time domain, which can be computed by [20]

$$\begin{aligned} \mu &= \int_{-\infty}^{+\infty} x |\psi(x)|^2 dx, \\ \sigma^2 &= \int_{-\infty}^{+\infty} (x - \mu)^2 |\psi(x)|^2 dx. \end{aligned} \quad (10)$$

As a result, its translated and dilated version $\psi_{a,b} = \psi((x - b)/a)$ will be concentrated in the region of $[b + a\mu - a\sigma, b + a\mu + a\sigma]$ in the time domain.

In this paper, the Mexican Hat wavelet function with symmetric graph (Figure 2) is employed, which is given by the following equation:

$$\psi_M(x) = (1 - x)^2 \cdot \exp\left(\frac{-x^2}{2}\right). \quad (11)$$

From (10), the center and radius of Mexican Hat wavelet window in the time domain can be derived as

$$\mu_M = 0, \quad \sigma_M = 1.08012345. \quad (12)$$

4.2. Initialization by a Novel Clustering Approach for WNN. The structure of our network is illustrated in Figure 1. Suppose the input data for network training are M vectors with n components: $\{\mathbf{x}_l = (x_{l1}, x_{l2}, \dots, x_{ln})^T, l = 1, 2, \dots, M\}$. The procedure comprises the following steps:

- (1) Create the first cluster $C_1 = \{\mathbf{x}_1\}$ with cluster mean $\mathbf{m}_1 = \mathbf{x}_1$ and dimensional radius $\mathbf{r}_1 = \mathbf{0}$. Set the number of clusters $J = 1$.

(2) Put $l = 2$.

(3) Compute the distance vectors $\mathbf{D}_i = |\mathbf{x}_l - \mathbf{m}_i| = (d_{i1}, d_{i2}, \dots, d_{in})^T, i = 1, 2, \dots, J$.

If there are some $\mathbf{D}_{i_s} \in \{\mathbf{D}_i\}_{i=1}^J$ ($s = 1, 2, \dots, S$), such that, $\forall j \in \{1, 2, \dots, n\}, d_{i_s j} \leq \max(r_{i_s j}, \text{th}_j)$ and $\text{Arg min}_k (\sum_{j=1}^n d_{kj}) = K$, then $\mathbf{x}_l \in C_K$. The vector $\mathbf{th} = (\text{th}_1, \text{th}_2, \dots, \text{th}_n)^T$ is a threshold vector which is set in advance of executing the algorithm. The cluster mean will be reset as

$$\mathbf{m}_K = \frac{1}{|C_K|} \sum_k \mathbf{x}_k \quad (13)$$

and dimensional radius $\mathbf{r}_K = (r_{K1}, r_{K2}, \dots, r_{Kn})$ will be reset as

$$r_{Kj} = \max_k |m_{Kj} - x_{kj}|, \quad j = 1, 2, \dots, n, \quad (14)$$

where $|C_K|$ is the cardinal number of C_K and $\mathbf{x}_k = (x_{k1}, x_{k2}, \dots, x_{kn})^T$ are the patterns that belong to C_K .

Else, the number of clusters becomes $J = J + 1$; create the J th cluster $C_J = \{\mathbf{x}_j\}$ with cluster mean $\mathbf{m}_J = \mathbf{x}_j$ and dimensional radius $\mathbf{r}_J = \mathbf{0}$.

(4) Put $l = l + 1$. If $l > M$, then stop; otherwise go to step (3).

Remark 1. (i) Vector $\mathbf{th} = (\text{th}_1, \text{th}_2, \dots, \text{th}_n)^T$ in the above procedure is crucial to the clustering result. Large elements of \mathbf{th} will lead to a coarse partition, namely, a small J , whereas \mathbf{th} with small value will lead to a large J . In practice, a reasonable \mathbf{th} should be determined by the input patterns. In our experiments, we prefer to adopt vector \mathbf{th} as in formula (15) to control the cluster scale, which offers moderate results in most times. Consider

$$\text{th}_j = \sqrt{\text{var}(\bar{\mathbf{x}}_j)}, \quad j = 1, 2, \dots, n, \quad (15)$$

where $\bar{\mathbf{x}}_j = (x_{1j}, x_{2j}, \dots, x_{Mj})$.

(ii) The conditions “there are some $\mathbf{D}_{i_s} \in \{\mathbf{D}_i\}_{i=1}^J$ ($s = 1, 2, \dots, S$) such that, $\forall j \in \{1, 2, \dots, n\}, d_{i_s j} \leq \max(r_{i_s j}, \text{th}_j)$ ” in step (3) are derived from “local response” property of activation wavelet functions $\Psi(\mathbf{x}) = \prod_{k=1}^n \psi((x_k - b_k)/a_k)$ in the wavelons. As a result, patterns satisfying that each feature activates corresponding 1-D wavelet function will be identified as a class.

(iii) After the clustering procedure of (1)–(4), the corresponding results help us to determine the number of wavelons in WNN as $N = J$ and the initial value of translation and dilation vectors as

$$\mathbf{b}_j = \mathbf{m}_j^T, \quad (16)$$

$$\mathbf{a}_j = \frac{\beta}{\sigma} \mathbf{r}_j^T. \quad (17)$$

β in (17) is a relaxation parameter which satisfies $\beta \geq 1$; σ is the window radius of wavelet function $\psi(x)$.

(iv) In order to avoid the dilation parameters being zeros, the radius vector of the cluster with single element should be

redefined. The minimum value strategy is employed which can be described as $r_{l'j} = \min_l(r_{lj}), (|C_{l'}| = 1, |C_l| > 1)$.

(v) The connecting weights $\mathbf{w} = (w_{ij})$ between the hidden layer and the output layer will be randomly initialized in the region $[-1, 1]$, and the bias term $\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_q)$ initialized as the mean vector of input patterns.

5. Training Algorithm

Gradient descent method is implemented for training the WNN in this paper. Parameters $(a_{jk}, b_{jk}, w_{ij}, \bar{y}_i)$ are adjusted in the opposite direction of the gradient such that the objective function in (18) of the model should be minimized. Consider

$$E = \frac{1}{M} \sum_{l=1}^M \|\mathbf{y}_l - \mathbf{f}_l\|^2 = \frac{1}{M} \sum_{l=1}^M \sum_{i=1}^q (y_{li} - f_{li})^2, \quad (18)$$

where \mathbf{y}_l is the output of network and \mathbf{f}_l is the desired output.

The corrections applied to parameters a_{jk}, b_{jk}, w_{ij} , and \bar{y}_i are shown as follows:

$$\begin{aligned} \Delta \bar{y}_i &= -\gamma_1 \frac{\partial E}{\partial \bar{y}_i} = -\gamma_1 \frac{2}{M} \sum_{l=1}^M (y_{li} - f_{li}), \\ \Delta w_{ij} &= -\gamma_2 \frac{\partial E}{\partial w_{ij}} = -\gamma_2 \frac{2}{M} \sum_{l=1}^M (y_{li} - f_{li}) \Psi_j(\mathbf{x}_l), \\ \Delta a_{jk} &= -\gamma_3 \frac{\partial E}{\partial a_{jk}} = -\gamma_3 \frac{2}{M} \sum_{l=1}^M \sum_{i=1}^q (y_{li} - f_{li}) w_{ij} \frac{\partial \Psi_j(\mathbf{x}_l)}{\partial a_{jk}}, \\ \Delta b_{jk} &= -\gamma_4 \frac{\partial E}{\partial b_{jk}} = -\gamma_4 \frac{2}{M} \sum_{l=1}^M \sum_{i=1}^q (y_{li} - f_{li}) w_{ij} \frac{\partial \Psi_j(\mathbf{x}_l)}{\partial b_{jk}}, \end{aligned} \quad (19)$$

where $\Psi_j(\mathbf{x}_l) = \prod_{k=1}^n \psi((x_{lk} - m_{jk})/d_{jk}), \partial \Psi_j(\mathbf{x}_l)/\partial a_{jk} = F(\mathbf{x}_l)(x_{lk} - b_{jk})(-1/a_{jk}^2), \partial \Psi_j(\mathbf{x}_l)/\partial b_{jk} = F(\mathbf{x}_l)(-1/a_{jk})$, and $F(\mathbf{x}_l) = \psi'((x_{lk} - b_{jk})/a_{jk}) \prod_{i=1, i \neq k}^n \psi((x_{li} - b_{ji})/a_{ji})$. $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ are the learning rates which should be set on the basis of specific experiment.

6. Simulation Examples

In this section, WNN model with two different initialization schemes is applied to three time series prediction problems, namely, the prediction of Mackey-Glass, Box-Jenkins, and traffic volume time series. The performance of WNN with the clustering based initialization approach (WNN-CIA) described in Section 4 is compared to Zhang’s heuristic initialization approach (WNN-HIA) in each simulation.

Because the architecture of WNN-HIA must be decided in advance, in order to compare directly, we adopt the same architecture with WNN-CIA in the experiments. Relaxation parameter β in (17) of WNN-CIA is set as 2.5 in all simulations and the Mexican Hat function defined in (11) is employed as the wavelet function in the hidden neurons of all models. Root mean square error (RMSE) given by (20) of the training/testing set is used as index for comparing

TABLE 1: Comparison results of WNN with two initialization methods for Mackey-Glass time series.

Model	Structure	Number of parameters	Mean of RMSE		Std. of RMSE	
			Training	Testing	Training	Testing
WNN-CIA	4-9-1	82	0.01398	0.01377	0.00188	0.00203
	4-8-1	73	0.02915	0.03073	0.00976	0.01079
WNN-HIA [9]	4-9-1	82	0.02789	0.02912	0.00899	0.00953
	4-10-1	91	0.02906	0.03059	0.01097	0.01174
	4-11-1	100	0.02700	0.02819	0.00871	0.00940

TABLE 2: Some test results of different models for Mackey-Glass time series.

Method	RMSE for testing set
Cascade correlation NN	0.06
Back-propagation NN	0.02
Sixth-order polynomial	0.04
Linear prediction method	0.55
Product T-norm [22]	0.0907
Genetic algorithm and fuzzy system [23]	0.049

performances of WNN with different initialization schemes. Consider

$$\text{RMSE} = \sqrt{E} = \sqrt{\frac{1}{M} \sum_{l=1}^M \sum_{i=1}^q (y_{li} - f_{li})^2}. \quad (20)$$

6.1. Prediction of Mackey-Glass Time Series. The Mackey-Glass chaotic time series is generated from the following delay differential equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t). \quad (21)$$

Here we predict the $x(t+6)$ using the input variables $x(t)$, $x(t-6)$, $x(t-12)$, and $x(t-18)$. Parameters in (21) are set as $a = 0.2$, $b = 0.1$, $\tau = 17$, and $x(0) = 1.2$ which make the equation show chaotic behavior. One thousand input-output data points are extracted from the Mackey-Glass time series $x(t)$, where $t = 118$ to $t = 1117$. The first 500 data pairs of the series are used as training data, while the remaining 500 data pairs are used to validate the proposed network. After performing the proposed clustering based initialization method proposed in Section 4.2, we get that the number of wavelons is $N = 9$.

For the performance comparison of WNN-CIA with WNN-HIA, some different architectures are employed for WNN-HIA. Table 1 shows the mean and standard deviation (std.) of RMSE for training and testing data obtained when 100 runs were performed by each model. The models are trained for 500 epochs in each run. Some results of different models for testing set are shown in Table 2. The RMSE reduction curve during training and testing of gradient descent algorithm corresponding to the best WNN-CIA model is drawn in Figure 3. Figures 4 and 5 show the prediction output of the best WNN-CIA model and the corresponding

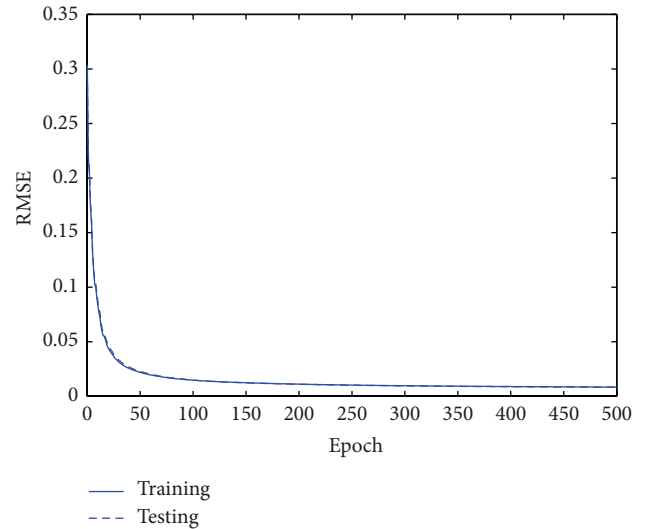


FIGURE 3: RMSE values obtained during training and testing for Mackey-Glass time series.

prediction error for training and testing data with the training and testing RMSE as 0.0080 and 0.0078.

From Table 1, it can be seen that the performance of WNN with structure and initial parameters derived by the proposed initialization approach is much better than that of WNN-HIA, even when more parameters are employed in the model.

6.2. Prediction of Box-Jenkins Time Series. The gas furnace data of Box and Jenkins (1970), that is, Box-Jenkins time series, 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 algorithms. During the process, the portion of methane was randomly changed, keeping a constant gas flow rate. 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.

In this section, the data set used consists of 292 consecutive values of methane at time $(t-4)$ and CO_2 produced in a furnace at time $(t-1)$ as input variables, with the produced CO_2 at time (t) as an output variable. Namely, variables $u(t-4)$ and $y(t-1)$ are used to predict $y(t)$. The data are partitioned in 200 data points as a training set and the remaining 92 points as a testing set for testing the performance of the proposed network. After performing

TABLE 3: Comparison results of WNN with two initialization methods for Box-Jenkins time series.

Model	Structure	Number of parameters	Mean of RMSE		Std. of RMSE	
			Training	Testing	Training	Testing
WNN-CIA	2-8-1	41	0.02023	0.05900	0.00086	0.00952
	2-7-1	36	0.02108	0.06220	0.00121	0.01516
WNN-HIA [9]	2-8-1	41	0.02110	0.05964	0.00172	0.01420
	2-9-1	46	0.02068	0.05616	0.00099	0.01339
	2-10-1	51	0.02064	0.05579	0.00115	0.01064

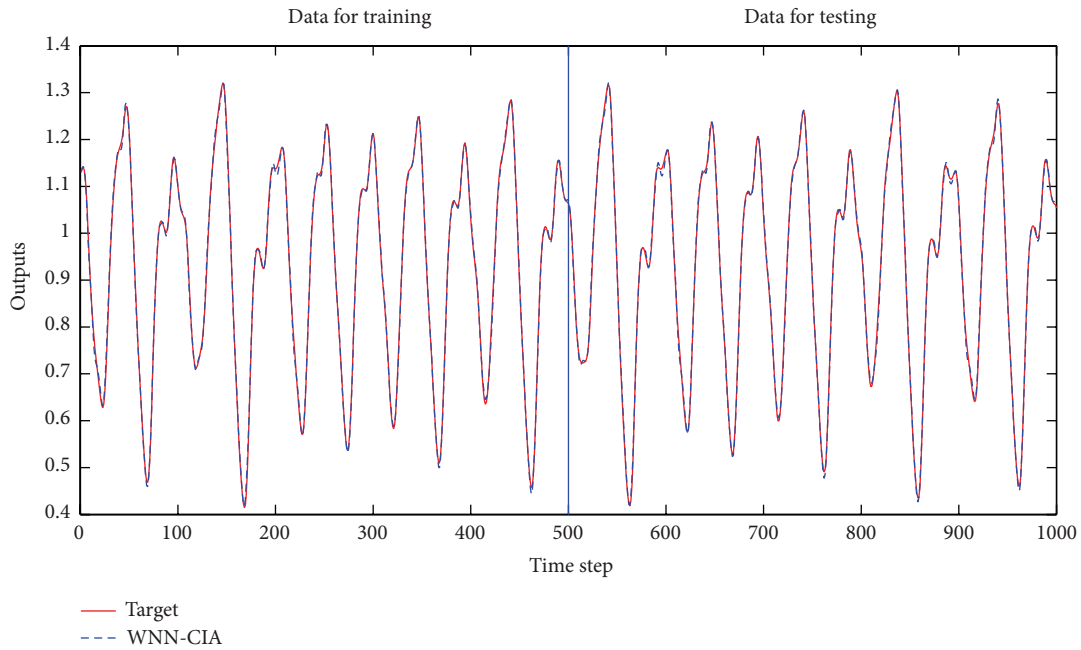


FIGURE 4: Prediction results for Mackey-Glass time series by WNN-CIA.

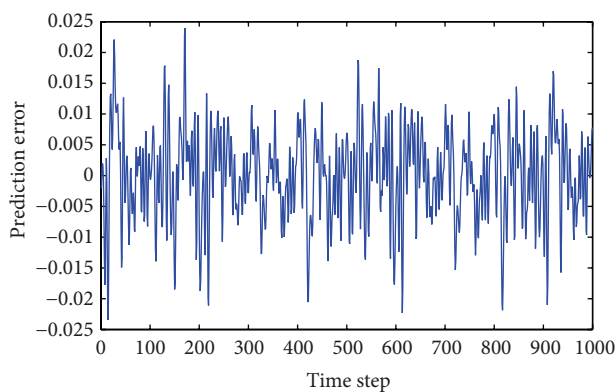


FIGURE 5: Prediction errors for Mackey-Glass time series by WNN-CIA.

the initialization method of WNN proposed in Section 4.2, we get the number of wavelons $N = 8$.

As is done in Section 6.1, different architectures are employed for WNN-HIA for comparison with WNN-CIA whose structure and initial parameters are derived by

TABLE 4: Some test results of different models for Box-Jenkins time series.

Method	Inputs	RMSE for testing set
Surmann's model [24]	2	0.400
Lee's model [25]	2	0.638
Lin's model [26]	5	0.511
Nie's model [27]	4	0.412
ANFIS model [28]	2	0.085
FuNN model [29]	2	0.071

the proposed approach. Table 3 shows the mean and standard deviation of RMSE for training and testing data obtained when 100 runs were performed by each model. The models are also trained for 500 epochs in each run. Table 4 shows some test results of different models. The RMSE reduction curve during training and testing of gradient descent algorithm corresponding to the best WNN-CIA model is drawn in Figure 6. Figures 7 and 8 show the prediction output of the best WNN-CIA model and the corresponding prediction error for training and testing data with the training and testing RMSE as 0.0186 and 0.0348.

TABLE 5: Comparison results of WNN with two initialization methods for traffic volume time series.

Model	Structure	Number of parameters	Mean of RMSE		Std. of RMSE	
			Training	Testing	Training	Testing
WNN-CIA	6-17-1	222	0.03201	0.05343	0.00462	0.01237
	6-16-1	209	0.03996	0.05692	0.01162	0.02125
WNN-HIA [9]	6-17-1	222	0.03868	0.05561	0.01109	0.02266
	6-18-1	235	0.03761	0.05373	0.01065	0.02035
	6-19-1	248	0.03796	0.05567	0.01248	0.02312

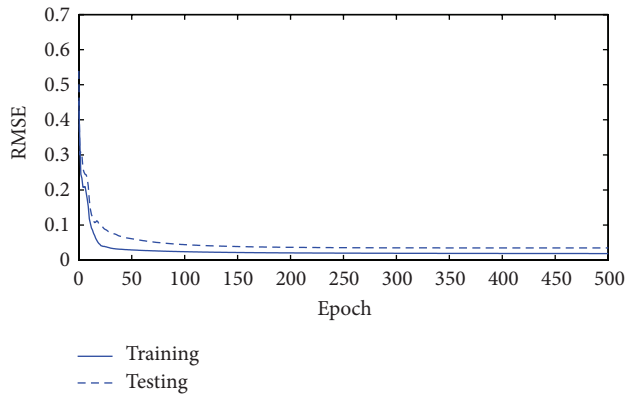


FIGURE 6: RMSE values obtained during training and testing for Box-Jenkins time series.

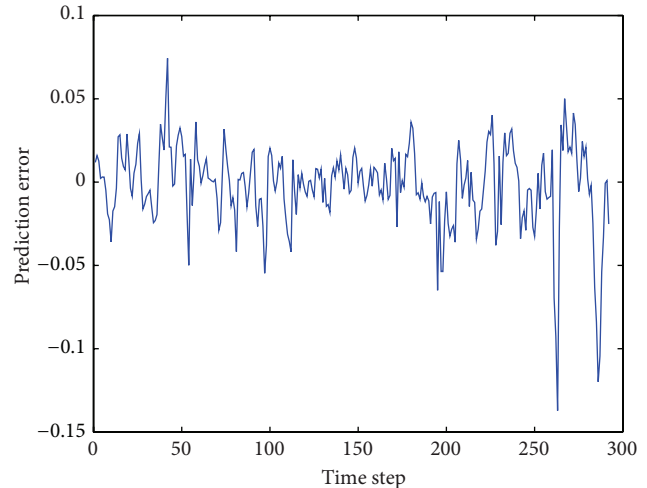


FIGURE 8: Prediction errors for Box-Jenkins time series by WNN-CIA.

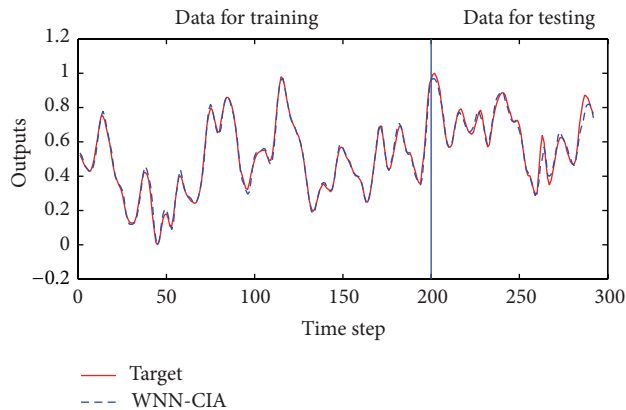


FIGURE 7: Prediction results for Box-Jenkins time series by WNN-CIA.

From the data in Table 3, we can see that WNN-CIA outperforms WNN-HIA when the same architectures are employed. When more parameters are employed to WNN-HIA, the performances of WNN-HIA gradually improve. However, WNN-CIA model can make a more stable performance than all WNN-HIA models in the experiments. In order to further examine the effectiveness of the proposed method, simulation experiments of a real-world example, traffic volume time series prediction, are carried out.

6.3. Prediction of the Traffic Volume Time Series (A Real-World Example). Chen in [21] implemented the neural network

time series models for traffic volume forecasting. In this section, the data of hourly traffic volume for station 5 from [21], which were collected on IR 271 and IR 90 in Cuyahoga County, are used as the real-world time series to examine the performance of WNN-CIA as well as WNN-HIA. There are 105 volume data points collected from June 4, 4:00 pm, to June 8, 12:00 pm, for training purposes, with the remaining 9 data points collected from 1:00 am to 9:00 am on June 9 reserved for model accuracy checking. This is a one-step forecasting with 6 anterior data points as input vector. Data normalizing is done to transfer values of the raw time series into the numbers in interval $[0, 1]$. After performing the initialization method of WNN proposed in Section 4.1, we get the number of wavelons $N = 17$.

Some same and different architectures are employed for WNN-HIA for comparison with WNN-CIA. After 100 experiments with 500 epochs in each run, Table 5 shows the mean and standard deviation of RMSE for training and testing data for two WNN models with different initialization methods. Test results of different models are shown in Table 6. The RMSE reduction curve during training and testing of gradient descent algorithm corresponding to the best WNN-CIA model is drawn in Figure 9. Figures 10 and 11 show the prediction output of the best WNN-CIA model and the corresponding prediction error for training and testing data with the training and testing RMSE as 0.0233 and 0.0335.

TABLE 6: Some test results of different models for traffic volume time series.

Method	Inputs	RMSE for testing set
Linear prediction method	6	0.3090
Sixth-order polynomial	6	0.1381
Back-propagation NN [21]	6	0.1244
Back-propagation NN [21]	12	0.1220
Radial basis function NN	6	0.0743
Elman NN	6	0.1041

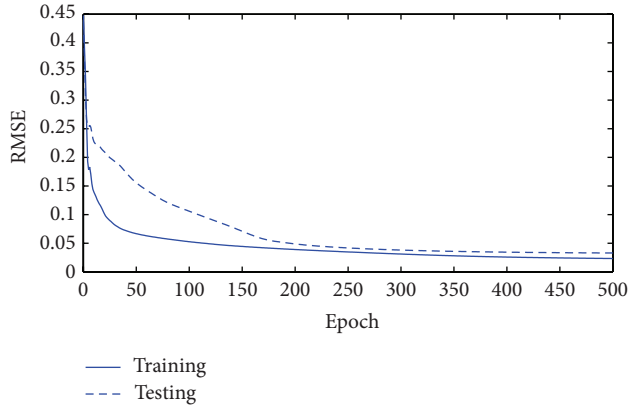


FIGURE 9: RMSE values obtained during training and testing for traffic volume time series.

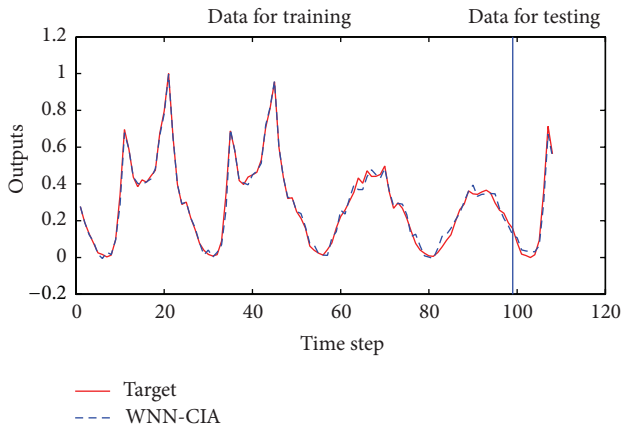


FIGURE 10: Prediction results for traffic volume time series by WNN-CIA.

From Table 5, we can see that the performance of WNN with the proposed clustering based initialization procedure is also superior to that with heuristic initialization approach even when more parameters are employed in WNN-HIA. It demonstrates again the validity of our methods.

7. Conclusion

In this paper, a novel initialization procedure for WNN as time series predictor is proposed, which behaves as

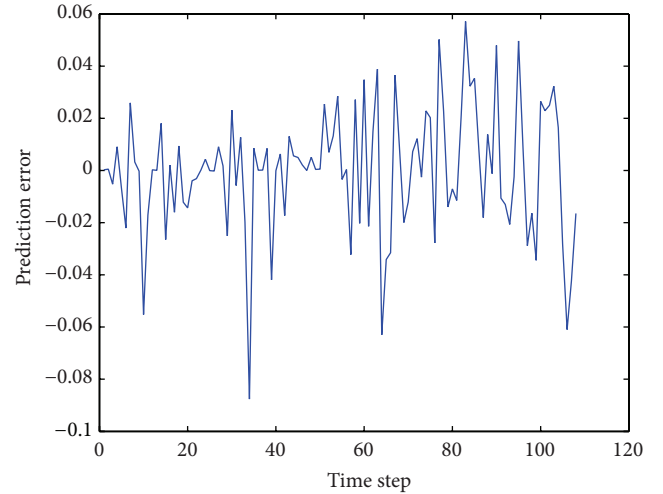


FIGURE 11: Prediction errors for traffic volume time series by WNN-CIA.

a dimensional clustering procession. Taking account of the distribution of input patterns and the local response property of wavelet functions, the input patterns can be dynamically classified by the proposed approach. And then the architecture as well as the initial values of translation and dilation parameters of WNN model can be determined accordingly. Simulation results demonstrate that, besides the capability of neuron number determination, WNN with our initialization method can provide satisfactory and stable results for time series prediction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are thankful that the research is supported by the National Science Foundation of China (61275120) and Tian Yuan Special Foundation (11426207).

References

- [1] R. J. Frank, N. Davey, and S. P. Hunt, "Time series prediction and neural networks," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 31, no. 1-3, pp. 91-103, 2001.
- [2] U. Lotric and A. Dobnikar, "Predicting time series using neural networks with wavelet-based denoising layers," *Neural Computing and Applications*, vol. 14, no. 1, pp. 11-17, 2005.
- [3] R. Setiono and H. Liu, "Feature extraction via neural networks," in *Feature Extraction, Construction and Selection*, pp. 191-204, Springer US, 1998.
- [4] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279-2301, 2002.
- [5] C. M. Bishop, *Neural Networks for Pattern Recognition*, The Clarendon Press, Oxford, UK, 1995.

- [6] M. Balasubramanian, S. Palanivel, and V. Ramalingam, "Real time face and mouth recognition using radial basis function neural networks," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6879–6888, 2009.
- [7] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000.
- [8] V. N. Ghate and S. V. Dudul, "Optimal MLP neural network classifier for fault detection of three phase induction motor," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3468–3481, 2010.
- [9] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 889–898, 1992.
- [10] J. Zhang, G. G. Walter, Y. Miao, and W. N. W. Lee, "Wavelet neural networks for function learning," *IEEE Transactions on Signal Processing*, vol. 43, no. 6, pp. 1485–1497, 1995.
- [11] I. Daubechies, *Ten Lectures on Wavelets*, vol. 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1992.
- [12] C. J. Lin, "Nonlinear systems control using self-constructing wavelet networks," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 71–79, 2009.
- [13] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, no. 4–6, pp. 449–465, 2006.
- [14] L. Jiao, J. Pan, and Y. Fang, "Multiwavelet neural network and its approximation properties," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1060–1066, 2001.
- [15] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, John Wiley & Sons, 2013.
- [16] D. Veitch, *Wavelet Neural Networks and Their Application in the Study of Dynamical Systems*, University of York, 2005.
- [17] R. Liu, *Research on Computational Intelligence-Based Structural Reliability Design Optimization*, Jilin University, 2006.
- [18] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
- [19] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAM Journal on Mathematical Analysis*, vol. 15, no. 4, pp. 723–736, 1984.
- [20] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
- [21] J. Chen, *Characterization and Implementation of Neural Network Time Series Models for Traffic Volume Forecasting*, University of Toledo, Toledo, Ohio, USA, 1997.
- [22] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414–1427, 1992.
- [23] D. Kim and C. Kim, "Forecasting time series with genetic fuzzy predictor ensemble," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 4, pp. 523–535, 1997.
- [24] H. Surmann, A. Kanstein, and K. Goser, "Self-organizing and genetic algorithms for an automatic design of fuzzy control and decision systems," in *Proceedings of the 1st European Congress on Fuzzy and Intelligent Technologies (EUFIT '93)*, Aachen, Germany, 1993.
- [25] Y. C. Lee, C. Hwang, and Y. P. Shih, "A combined approach to fuzzy model identification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 5, pp. 736–744, 1994.
- [26] Y. Lin and G. A. Cunningham III, "A new approach to fuzzy-neural system modeling," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 190–198, 1995.
- [27] J. Nie, "Constructing fuzzy model by self-organizing counter-propagation network," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 6, pp. 963–970, 1995.
- [28] J. S. R. Jang and C. T. Sun, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, 1996.
- [29] N. K. Kasabov, J. Kim, M. J. Watts, and A. R. Gray, "FuNN/2—a fuzzy neural network architecture for adaptive learning and knowledge acquisition," *Information Sciences*, vol. 101, no. 3–4, pp. 155–175, 1997.