Electrical Power and Energy Systems 64 (2015) 671–678

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

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

A novel approach to fuzzy wavelet neural network modeling and optimization

Rong Cheng [a,b], Yanping Bai [a,*]

[a] School of Science, North University of China, Shanxi, Taiyuan 030051, People's Republic of China
[b] School of Information and Communication Engineering, North University of China, Shanxi, Taiyuan 030051, People's Republic of China

ARTICLE INFO

Article history:
Received 8 April 2014
Received in revised form 15 July 2014
Accepted 19 July 2014

Keywords:
Fuzzy logic
Wavelet neural network
Particle swarm optimization
Gradient descent algorithm
System identification

ABSTRACT

In this paper, an efficient approach of combining Takagi–Sugeno–Kang fuzzy system with wavelet based neural network is presented. The model replaces the constant or a linear function of inputs in conclusion part of traditional TSK fuzzy model with wavelet neural network (WNN), thus each rule uses fuzzy set to separate the input space into subspaces spanned by different wavelet functions. For finding the optimal values for parameters of our proposed fuzzy wavelet neural network (proposed-FWNN), a hybrid learning algorithm integrating an improved particle swarm optimization (PSO) and gradient descent algorithm is employed. The two-layer inline-PSO process is proposed in this paper, whose adjustment scheme is more fitting the consequent pattern learning based gradient descent optimization and will locate a good region in the search space. Simulation examples are given to test the efficiency of proposed-FWNN model for identification of the dynamic plants. It is seen that our modeling and optimization approach results in a better performance.

Introduction

Recently, the concepts of neural network, fuzzy logic, wavelet technology have got a lot of attention by researchers. In the field of artificial intelligence, neural networks have been widely used on account of its ability of nonlinear approximation and advantage of easy realization [1–3]. Due to the ability of wavelet transformation for revealing the property of function in localize region, different types of wavelet neural network (WNN) which combine wavelets with neural networks have been proposed [4–6]. In WNN, the wavelets were introduced as activation functions of the hidden neurons in traditional feedforward neural networks with a linear output neuron. There are two different WNN architectures: one type has fixed wavelet bases possessing fixed dilation and translation parameters. 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. Several WNN models have been proposed in the literatures. In [7], 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 [8], 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 [9], 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.

Fuzzy logic systems are often used to deal with complex nonlinear systems with ill-defined conditions and uncertain factors [10,11]. Compared with the difficulty to understand the meaning associated with each neuron and each weight in the neural network, fuzzy logic uses linguistic terms and the structure of if-then rules. The traditional Takagi–Sugeno–Kang (TSK) fuzzy model consist of a set of rules, and each rule uses fuzzy set to separate the input space into local fuzzy regions. As the form of combining the benefits of neural network and fuzzy systems, fuzzy neural networks have emerged as a powerful approach to solving many problems [12–15]. In [12], a fuzzy modeling method using fuzzy neural networks with the backpropagation algorithm is presented which can identify the fuzzy model of a nonlinear system automatically. In [13], a hybrid neural fuzzy inference system (HyFIS) for building and optimizing fuzzy models is proposed which is applied to an on-line incremental adaptive learning for the prediction and control of nonlinear dynamical systems. In [14], the adaptive neural fuzzy inference system is used as classifier of fault in power distribution system and makes good performance. In [15], a

* Corresponding author.
E-mail addresses: chengro@163.com (R. Cheng), baiyp666@163.com (Y. Bai).

http://dx.doi.org/10.1016/j.ijepes.2014.07.067
0142-0615/

self-organizing complex neuro-fuzzy system is presented and applied to the problem of time series forecasting.

Taking account of the neural networks' self learning ability, fuzzy logic's handling uncertainty capacity and wavelet transforms' analyzing local details superiority, several researchers have made a synthesis model of a fuzzy wavelet neural network (FWNN) [16–20]. In [16], a fuzzy wavelet network is proposed for approximating arbitrary nonlinear functions. Each rule of network corresponding to a sub-wavelet neural network consists of single-scaling wavelets. In [17], a dynamic time-delay fuzzy wavelet neural network model is presented for nonparametric identification of structures using the nonlinear autoregressive moving average with exogenous inputs approach. In [18], the proposed fuzzy wavelet neural network is used for the identification and the control of the dynamic plants. Each rule in FWNN includes a wavelet function in the consequent part of the rule and multi-dimensional wavelet functions are the summation form. In [19], the genetic algorithm (GA) approach is used to adjust the FWNN parameters of dilation, translation, weights, and membership functions. In [20], a hybrid adaptive wavelet-neuro-fuzzy system is proposed using wavelets as membership functions in the antecedent part as well as the activation functions in the consequent part of fuzzy rules.

Inspired by social behavior of bird flocking or fish schooling, particle swarm optimization (PSO) was firstly introduced by Kennedy and Eberhart in 1995 [21,22]. After several years of development, PSO and its modified forms have evolved as an important branch of stochastic techniques to explore the search space for optimization, which have been successfully applied in many areas: function optimization, artificial network training, fuzzy system control and so on [23–27]. The attractive features of PSO include ease of implementation and the fact that no gradient information is required. However, PSO does exhibits some disadvantages: like other heuristic algorithms, it takes more calculation time than traditional gradient descent method when you want to achieve similar accuracy; it sometimes is easy to be trapped in local optima, and the convergence rate decreased considerably in the later period of evolution; when reaching a near optimal solution, the algorithm stops optimizing, and thus the accuracy the algorithm can achieve is limited.

In this paper, a novel fuzzy wavelet neural network is proposed, which use the concepts of fuzzy logic in combination with WNN. In our proposed fuzzy wavelet neural network (proposed-FWNN), each fuzzy rule corresponds to a WNN consisting of several wavelets with adjustable translation and dilation parameters. In the aspect of optimizing the proposed-FWNN, in order to avoid the trial-and-error process and the impact coming from random initialization, we adopt a hybrid learning algorithm. Firstly, an inline-PSO algorithm is proposed to find a relative good initial value of adjustable parameters. The inline-PSO shows a faster convergence than the basic PSO and the updating scheme of the velocity and position of particles are more coordinating with the following pattern learning based gradient descent algorithm. Secondly, the gradient descent algorithm is adopted to adjust parameters in the proposed-FWNN. For getting a more reasonable model, the performance criterion of training and testing signals during learning are both investigated.

The rest of the paper is organized as follows. The proposed-FWNN is introduced in Section "The proposed fuzzy wavelet neural network modeling". A hybrid learning algorithm for training proposed-FWNN is described in Section "Hybrid learning algorithm to optimize the proposed-FWNN". In Section "Simulation examples", two simulation examples of system identification are given to demonstrate the better performance of proposed-FWNN. Finally, a brief conclusion is drawn in Section "Conclusion".

The proposed fuzzy wavelet neural network modeling

Motivated by the reason stated in Section "Introduction", we present a novel type of fuzzy wavelet-based model. TSK fuzzy models are employed to describe the proposed-FWNN by some fuzzy rules, and WNNs consisting of several wavelets with adjustable translation and dilation parameters form the consequent parts of each fuzzy rules.

Takagi–Sugeno–Kang fuzzy system

In a TSK fuzzy model, the domain interval of each input is separated into fuzzy regions and each region shows a membership function in the IF part of the fuzzy rules. A constant or a linear function of inputs is used in the THEN part of the rules. That is, the IF-THEN rules are as follows [10]:

$$\begin{aligned} R_k&: \text{IF } x_1 \text{ is } A_{k1} \text{ AND } x_2 \text{ is } A_{k2} \text{ AND } \cdots \text{ AND } x_n \text{ is } A_{kn} \\ &\text{THEN } y_k = a_{k0} + a_{k1}x_1 + \cdots + a_{kn}x_n, \end{aligned} \tag{1}$$

where R_k represents the kth fuzzy inference rule, x_j and A_{kj} are fuzzy variables and fuzzy sets with membership functions.

The fuzzy membership functions of A_{kj} are Gaussian function defined by (2):

$$\mu_{kj}(x_j) = \exp\left(-\left(\frac{x_j - c_{kj}}{\sigma_{kj}}\right)^2\right), \tag{2}$$

where c_{kj} denote the centers and σ_{kj} denote the standard deviation for membership function associated with rule k. The output of TSK fuzzy system with M rules is aggregated as (3):

$$y = \frac{\sum_{k=1}^{M} y_k \prod_{j=1}^{n} \mu_{kj}(x_j)}{\sum_{k=1}^{M} \prod_{j=1}^{n} \mu_{kj}(x_j)}. \tag{3}$$

The TSK fuzzy model is based on a fuzzy partition of input space and it can be viewed as expansion of a piecewise linear partition.

Wavelet neural network

Wavelets in the following form:

$$\psi_{a,b} = |a|^{-1/2}\psi\left(\frac{x-b}{a}\right), (a, b \in R, a \neq 0) \tag{4}$$

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

$$C_\psi = \int_0^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty, \tag{5}$$

where $\hat{\psi}(\omega)$ is the Fourier transform of $\psi(x)$ [28,29].

Grossmann and Morlet [30] proved that any function $f(x)$ in $L^2(R)$ can be represented by (6):

$$f(x) = \frac{1}{C_\psi} \iint Wf(a,b)|a|^{-1/2}\psi\left(\frac{x-b}{a}\right)\frac{1}{a^2} dadb, \tag{6}$$

where $Wf(a,b)$ given by (7):

$$Wf(a,b) = |a|^{-1/2} \int_{-\infty}^{+\infty} \psi\left(\frac{x-b}{a}\right) f(x) dx \tag{7}$$

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,

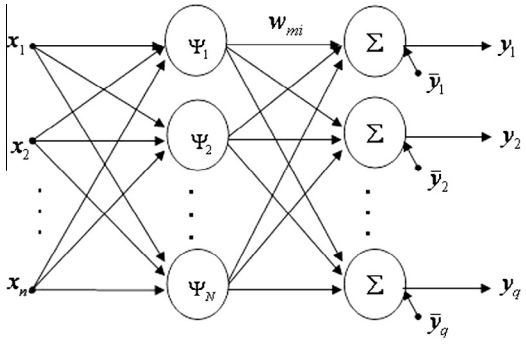

Fig. 1. Structure of WNN.

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.

Inspired by the wavelet decomposition of $f(x) \in L^2(R)$ in (6) and a single hidden layer network model, Zhang and Benveniste [5] had developed a new neural network model, namely, wavelet neural network (WNN). The structure of a WNN is shown in Fig. 1. For the modeling of multivariable processes here, multi-dimensional wavelets must be defined. In the present work, multi-dimensional wavelets are defined as (8):

$$\Psi_i(\mathbf{x}) = \prod_{j=1}^{n} \psi\left(\frac{x_j - b_{ij}}{a_{ij}}\right), \quad i = 1,2,\ldots,N, \tag{8}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ is the input vector, $\mathbf{b}_i = (b_{ij})$ and $\mathbf{a}_i = (a_{ij})$ are the translation and dilation vectors, respectively. The network output $\mathbf{y} = (y_1, y_2, \ldots, y_q)^{\mathrm{T}}$ is computed as (9):

$$y_m = \sum_{i=1}^{N} w_{mi}\Psi_i + \bar{y}_m, \quad m = 1,2,\ldots,q, \tag{9}$$

where $\mathbf{w} = (w_{mi})$ and $\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_q)$ defines 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. It is easy to see that the WNN is in fact a feed-forward neural network with one hidden layer where wavelet functions act as activation functions in the hidden nodes and a linear output layer.

Structure of the proposed-FWNN

As mentioned earlier, in most fuzzy models, the approximated function is a linear combination of the input variables plus a constant term. These systems do not have localizability and their convergence is generally slow. In the case of modeling of complex nonlinear processes, the lack of full mapping capabilities of the consequent part of these fuzzy systems will show more obviously. In this paper, we propose a new network using WNN (rather than linear functions) in (9) as the consequent part of TSK fuzzy models, which can achieve the desired accuracy with less number of rules compared with other fuzzy models (simulation results are illustrated in Section "Simulation examples"). For a convenient expression, the proposed-FWNN with one output node is considered here.

The fuzzy rules of the proposed-FWNN are of the following form:

R_k: IF x_1 is A_{k1} AND x_2 is A_{k2} AND ... AND x_n is A_{kn}

THEN $Y_k = \sum_{i=1}^{N_k} w_i^k \Psi_i^k + \bar{y}_k$,

where $x_1, x_2, \ldots, x_n$ are the input variables, $Y_1, Y_2, \ldots, Y_M$ are the output variables, A_{kj} is the kth fuzzy set with Gaussian membership functions defined in (2). Conclusion parts of the rules contain different WNNs with N_k wavelet activating functions

$$\Psi_i^k = \prod_{j=1}^{n} \psi_{ij}^k = \prod_{j=1}^{n} \psi\left(\frac{x_j - b_{ij}^k}{a_{ij}^k}\right) \tag{10}$$

in the hidden neurons. w_i^k and $\bar{y}_k$ are the connecting weights and the bias terms.

The structure of the proposed-FWNN is depicted in Fig. 2. It includes 7 layers. Nodes in layer 1 pass the input signals to the second layer. Nodes in layer 2 act as the membership functions in the IF part of the rules. So we called it the fuzzification layer. The fuzzification neuron performs the mappings from crisp value x_j into fuzzy set A_{kj} , with degree $\mu_{kj}(x_j)$ as in (2). In layer 3, each node represents one fuzzy rule. AND operator is used to calculate the output signals showed as (11):

$$O_k = O_k^{(3)} = \prod_{j=1}^{n} \mu_{kj}(x_j), \quad (k = 1,2,\ldots,M). \tag{11}$$

Layer 4 accepts the variables $x_1, x_2, \ldots, x_n$ as input signals, which consists of M wavelet neural networks and each network

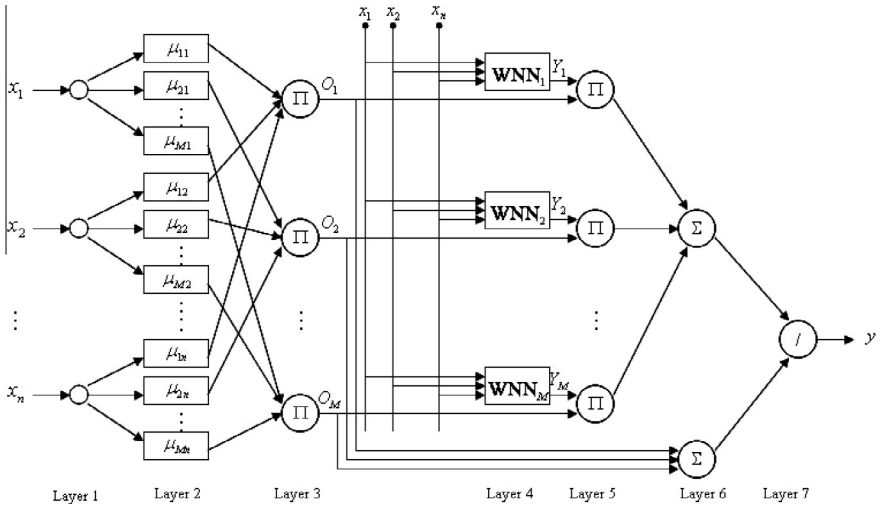

Fig. 2. Structure of proposed-FWNN.

corresponds to a consequent part of a fuzzy rule. The number of wavelet neurons in WNN corresponding to the kth fuzzy rule is N_k , and the outputs of layer 4 are calculated as (12):

$$Y_k = O_k^{(4)} = \sum_{i=1}^{N_k} w_i^k \Psi_i^k + \bar{y}_k. \tag{12}$$

The Defuzzification inference is described in layer 5–7. In layer 5, the output signals of layer 3 are multiplied by the output signals of layer 4. Two neurons in layer 6 perform as the summation operator to the output signals from layer 5 and layer 3 respectively. And the output neuron in layer 7 calculates the quotient which shows the contribution of each WNN to the final output of proposed-FWNN. The specific formulas are described by (13)–(15):

$$O_k^{(5)} = O_k^{(3)} \cdot O_k^{(4)} = O_k \cdot Y_k. \tag{13}$$

$$O_1^{(6)} = \sum_{k=1}^{M} O_k^{(5)}, \quad O_2^{(6)} = \sum_{k=1}^{M} O_k^{(3)}. \tag{14}$$

$$y = O^{(7)} = \frac{O_1^{(6)}}{O_2^{(6)}} = \frac{\sum_{k=1}^{M} O_k Y_k}{\sum_{k=1}^{M} O_k}. \tag{15}$$

The structure of our proposed-FWNN is different from the FWNN structures designed in some existing literatures. In [16,31] for example, each rule of the FWNN corresponds to a sub-wavelet neural network consists of single-scaling wavelets. While the translation and dilation parameters in our WNN in the consequent part of fuzzy rule are all adjustable. In [18], the multidimensional wavelet function is the summation form, while we adopt the multiplication form that is employed in much more cases. Moreover, in out models, the input space is made a fuzzy partition to "multidimensional" sub-space with several wavelet bases. While in [18], the partition is to "unidimensional" sub-space in nature. That is to say, our proposed-FWNN have more freedom in design and may employ less fuzzy rules to achieve certain accuracy.

Hybrid learning algorithm to optimize the proposed-FWNN

The parameter vector in proposed-FWNN to be updated is $\Theta = (c_{kj}, \sigma_{kj}, b_{ij}^k, a_{ij}^k, w_i^k, \bar{y}_k)$ consisting of the center parameters c_{kj} and standard deviation parameters σ_{kj} of the membership functions in the layer 2; translation parameters b_{ij}^k , dilation parameters a_{ij}^k of wavelet functions, weight parameters w_i^k and bias $\bar{y}_k$ in the layer 4. Here $k = 1, 2, \ldots, M$, $i = 1, 2, \ldots, N_k$, $j = 1, 2, \ldots, n$ are the subscripts corresponding to the fuzzy rule, wavelet neurons and input signals respectively. In this paper, all parameters are adjusted by a hybrid learning algorithm consisting of initialization by a proposed inline particle swarm optimization (inline-PSO) and adjustment by the gradient descent algorithm (GDA).

Basic PSO algorithm

PSO is an evolutionary computation technique mimicking the behavior of flying birds and their means of information exchange. In PSO, each single solution is a particle in the search space. All of particles have fitness values that are evaluated by the fitness unction to be optimized, and have velocities that direct the flying of the particles. Similar to other population based evolutionary algorithms, PSO as an optimization tool can solve a variety of difficult optimization problems. The system is initialized with a population of random solutions and searches for optima by updating generations.

In the basic PSO algorithm, the velocity and position updating rule is given by (16) and (17) [21,22]:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (pbest_{id}^k - x_{id}^k) + c_2 r_2 (gbest_d^k - x_{id}^k), \tag{16}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}, \tag{17}$$

where c_1 and c_2 are positive constants named acceleration coefficients. r_1 and r_2 are two independent random numbers uniformly distributed in the range $[0, 1]$. ω is the inertia weight factor which can be a constant or a variable of iteration. Empirical studies of PSO with inertia weight have shown that a relatively large ω have more global search ability while a relatively small ω results in a faster convergence. $pbest_{id}^k$ is the best previous position along the dth dimension of particle i in the iteration k and $gbest_d^k$ is the best previous position among all the particles along the dth dimension in iteration k.

Inline-PSO algorithm of proposed-FWNN

As an optimization algorithm, basic PSO and its improved schemes has already been applied in many areas, such as function optimization, artificial neural network training, pattern classification and fuzzy system control. In consideration of the PSO's disadvantage of taking more calculation time for a relatively good accuracy, in this paper, we would like to combine it with gradient descent algorithm to train the proposed-FWNN. That is to say, a "reasonable" PSO with relatively small population size and less number of iterations is performed to get a good initialization of proposed-FWNN, and then a gradient descent algorithm is used to get the final values of parameters which make the model achieving the satisfactory solution.

The advantages of the hybrid learning algorithm are obvious. Firstly, it brings more stable training process than that employing only one optimization algorithm (PSO or GDA), which shows stronger dependent on the random factors of training. Secondly, as mentioned in some literatures, the particles may shows the "similarity" phenomenon in optimization procedure, which makes the convergence of PSO very slow. Combining with the gradient descent algorithm can speed the convergence of the training process.

In this subsection, taking into account of the adjustment scheme of parameters in pattern learning based gradient descent, which update the parameter after each measurement $(\mathbf{x}_l, f_l)$,$(l = 1, 2, \ldots, K)$ in the opposite direction of the gradient of function defined by (18):

$$E(\Theta, \mathbf{x}_l, f_l) = \frac{1}{2}(y_l(\Theta, \mathbf{x}_l) - f_l)^2, \tag{18}$$

a modified PSO named inline-PSO algorithm containing two fitness functions is proposed. y_l in (18) refers to the output of network based on input $\mathbf{x}_l$ and parameter vector Θ. Its optimization scheme is constituted by two layers. In the outer layer, particles' velocity and position are updated based on fitness values calculated by the performance index of root mean square error (RMSE) given by (19):

$$\mathrm{RMSE}(\Theta, \tilde{\mathbf{x}}, \mathbf{f}) = \sqrt{\frac{1}{K}\sum_{l=1}^{K}(y_l - f_l)^2}. \tag{19}$$

where $\tilde{\mathbf{x}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}, \mathbf{f} = \{f_1, f_2, \ldots, f_K\}$. In the inner layer, particles' velocity and position are updated based on fitness values calculated by (18).

The detailed inline-PSO approach is as follows:

Step 1: Initialization. Set the population size (*psize*) and the termination iterative number (*Maxgen*) of evolution. Initialize velocity and position of particles randomly. Let the iterative number $k = 1$.

Step 2: Calculate the fitness value of each particle by (19) and represent it by $fitness_i$,$(i = 1, 2, \ldots, psize)$. Find the best

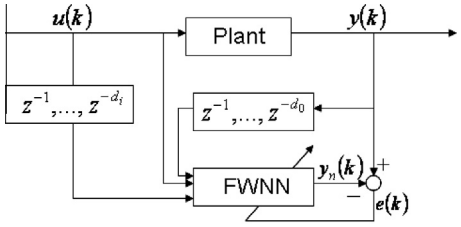

Fig. 3. Structure of identification scheme.

previous position $\mathbf{pbest}_i^k$ of particle i and choose the best previous position $\mathbf{gbest}^k$ among all the particles.

Step 3: Update the velocity and position of particles by (16) and (17).

S3.1: Sort the input vectors randomly and record the new input sequence as $\{\hat{\mathbf{x}}_l\}_{l=1}^K$. Let $l=1$.

S3.2: Calculate the fitness value $(E(\Theta,\hat{\mathbf{x}}_l,f_l))$ of each particle by (18) and represent it by $Infitness_i$, $(i=1,2,\ldots,psize)$. Find the best previous position $\mathbf{Inpbest}_i^k$ of particle i and choose the best previous position $\mathbf{Ingbest}^k$ among all the particles.

S3.3: Update the velocity and position of particles by

$$v_{id}^{k+1}=\omega v_{id}^k+c_1r_1(Inpbest_{id}^k-x_{id}^k)+c_2r_2(Ingbest_d^k-x_{id}^k),$$
$$x_{id}^{k+1}=x_{id}^k+v_{id}^{k+1},$$

S3.4: Let $l=l+1$. If $l<K+1$, go to S3.2; otherwise go to Step 4.

Step 4: Let $k=k+1$. Calculate the fitness value $fitniess_i$ of each particle and find the best previous position $\mathbf{pbest}_i^k$ of particle i and choose the best previous position $\mathbf{gbest}^k$ among all the particles. If $k>Maxgen$, stop and pick up $\mathbf{gbest}^k$; otherwise go to Step 3.

Remarks.

1. Inertia weight factor ω here can be a constant or a variable of iteration. In this paper, we prefer to a variable that linear decrease with the number of iterations which has been used in some literatures [32,33] and showed a satisfactory performance. The linear decreasing weight is as follows:

$$\omega(k)=\omega_{start}-(\omega_{start}-\omega_{end})*k/Maxgen, \tag{20}$$

where $\omega_{start}=0.9$ and $\omega_{end}=0.4$. For reducing the calculation time,less input–output pairs can adopted for updating particles' velocity and position in step 3 if the number of training dada is large.

2. After same iteration steps, the particles of inline-PSO will reach a less fitness value compared with the basic PSO, which can be illustrated in the simulation examples in Section "Simulation examples". And due to the two-layer adjustment scheme, two advantages are brought. Firstly, particles in population can maintain a certain degree of diversity accompanied with narrowing search space around the global best position. Secondly, parameters in particles of the inline-PSO can fit the consequent gradient descent optimization well and reach the optimal solution quickly.
3. The inline-PSO algorithm is used to locate a good region in the search space and then a gradient descent search algorithm is employed to fine tune the optimal solution as the following subsection.

Gradient descent algorithm following the inline-PSO of proposed-FWNN

Gradient descent method is implemented for training the proposed-FWNN after parameter initialization by inline-PSO. Parameters $\Theta=(c_{kj},\sigma_{kj},b_{ij}^k,a_{ij}^k,w_i^k,\bar{y}_k)$ are adjusted in the opposite direction of the gradient of the objective function defined by (21), which is based on each pattern presentation:

$$E(\Theta,\mathbf{x},y)=\frac{1}{2}(y-f)^2, \tag{21}$$

and the adjusting formulas are:

$$\Theta(t+1)=\Theta(t)+\Delta\Theta,$$
$$\Delta\Theta=\left(-\gamma_c\frac{\partial E}{\partial c_{kj}},-\gamma_\sigma\frac{\partial E}{\partial\sigma_{kj}},-\gamma_b\frac{\partial E}{\partial b_{ij}^k},-\gamma_a\frac{\partial E}{\partial a_{ij}^k},-\gamma_w\frac{\partial E}{\partial w_i^k},-\gamma_{\bar{y}}\frac{\partial E}{\partial\bar{y}_k}\right), \tag{22}$$

where γ is the learning rate. The values of derivatives in (22) can be calculated by the following formulas (23)–(28):

$$\frac{\partial E}{\partial c_{kj}}=(y-f)\cdot\frac{\partial y}{\partial O_k}\cdot\frac{\partial O_k}{\partial\mu_{kj}}\cdot\frac{\partial\mu_{kj}}{\partial c_{kj}}=(y-f)\cdot\frac{\partial y}{\partial O_k}\cdot\frac{\partial O_k}{\partial\mu_{kj}}\cdot\mu_{kj}\cdot\frac{2(x_j-c_{kj})}{\sigma_{kj}^2}, \tag{23}$$

$$\frac{\partial E}{\partial\sigma_{kj}}=(y-f)\cdot\frac{\partial y}{\partial O_k}\cdot\frac{\partial O_k}{\partial\mu_{kj}}\cdot\frac{\partial\mu_{kj}}{\partial\sigma_{kj}}=(y-f)\cdot\frac{\partial y}{\partial O_k}\cdot\frac{\partial O_k}{\partial\mu_{kj}}\cdot\mu_{kj}\cdot\frac{(x_j-c_{kj})^2}{\sigma_{kj}^3}, \tag{24}$$

where $\frac{\partial y}{\partial O_k}=\frac{Y_k\cdot\sum_{k'=1}^M O_{k'}+\sum_{k'=1}^M O_{k'}Y_{k'}}{\left(\sum_{k'=1}^M O_{k'}\right)^2}$, $\frac{\partial O_k}{\partial\mu_{kj}}=\prod_{l=1}^{j-1}\mu_{kl}\prod_{l=j+1}^{n}\mu_{kl}$.

$$\frac{\partial E}{\partial b_{ij}^k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}\cdot\frac{\partial Y_k}{\partial\psi_{ij}^k}\cdot\frac{\partial\psi_{ij}^k}{\partial b_{ij}^k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}\cdot\frac{\partial Y_k}{\partial\psi_{ij}^k}\cdot\psi'\cdot\frac{-1}{a_{ij}^k}, \tag{25}$$

$$\frac{\partial E}{\partial a_{ij}^k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}\cdot\frac{\partial Y_k}{\partial\psi_{ij}^k}\cdot\frac{\partial\psi_{ij}^k}{\partial a_{ij}^k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}\cdot\frac{\partial Y_k}{\partial\psi_{ij}^k}\cdot\psi'\cdot\frac{-(x_j-b_{ij}^k)}{(a_{ij}^k)^2}, \tag{26}$$

$$\frac{\partial E}{\partial w_i^k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}\cdot\frac{\partial Y_k}{\partial w_i^k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}\cdot\Psi_i^k, \tag{27}$$

$$\frac{\partial E}{\partial\bar{y}_k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}\cdot\frac{\partial Y_k}{\partial\bar{y}_k}=(y-f)\cdot\frac{\partial y}{\partial Y_k}, \tag{28}$$

where

$\frac{\partial y}{\partial Y_k}=\frac{O_k}{\sum_{k'=1}^M O_{k'}}$, $\frac{\partial Y_k}{\partial\psi_{ij}^k}=w_i^k\prod_{l=1}^{j-1}\psi_{il}^k\prod_{l=j+1}^{n}\psi_{il}^k$, $k=1,2,\ldots,M$, $i=1,2,\ldots,N_k$, $j=1,2,\ldots,n$.

Simulation examples

In order to evaluate the performance of proposed-FWNN, two simulation studies of nonlinear system identifications are carried out employing the proposed model and learning algorithm. The structure of the identification scheme with proposed-FWNN is shown in Fig. 3. The inputs of the model are delayed values of control signal $u(k)$ and plant output $y(k)$. Here, the problem is to find such values of parameters of proposed-FWNN by using them in the system for all input values of $u(k)$ the difference between plant output $y(k)$ and network output $y_n(k)$ will be minimum. In all experiments here, Gauss wavelet functions are adopted as the activation functions in proposed-FWNN.

Example 1

In this example, the plant to be identified is described as

$$y(k)=0.72y(k-1)+0.025y(k-2)u(k-2)+0.01u^2(k-3)+0.2u(k-4). \tag{29}$$

This plant is the same as that used in [34,18]. The current output of the plant depends on two previous outputs and four previous

inputs. However, we only adopt the current state of system and the control signal feeding into the proposed-FWNN as inputs. Input signals used for training proposed-FWNN are same as in literatures [34,18], which is an independent and identically distributed uniform sequence over $[-2,2]$ for about half of the 900 time steps and a sinusoid given by $1.05sin(\pi k/45)$ for the remaining time. To see the identification result, the following input signal is adopt for test:

$$u(k)=\begin{cases} sin(\pi k/25), & k<250 \\ 1.0, & 250\leqslant k<500 \\ -1.0, & 500\leqslant k<750 \\ 0.3sin(\pi k/25)+0.1sin(\pi k/32) & \\ \quad +0.6sin(\pi k/10), & 750\leqslant k<1000. \end{cases} \tag{30}$$

Two rules ($M=2$) and two wavelet neurons ($N_k=2, k=1,2$) in each WNN are employed in our experiment. So the number of adjustable parameters is $N=30$. The hybrid learning algorithm stated in Section "Hybrid learning algorithm to optimize the proposed-FWNN" is used to train the proposed-FWNN. The optimization results of inline-PSO and basic PSO are compared and illustrated in Fig. 4. The population size $psize=20$ and the termination iterative number $Maxgen=50$, which are set relatively small to shorten the runtime and lest overtraining of the training signal which may lead to narrow search space of testing signal. The acceleration coefficients c_1 and c_2 are set to 2, and the linear decreasing inertia weight defined by (20) are adopted here. As shown in Fig. 4, the convergence of inline-PSO is faster than that of PSO, which reach the fitness values of 0.0250 and 0.0380 respectively.

The RMSE reduction curve during training and testing of gradient descent algorithm in the consequent part of hybrid learning scheme is illustrated in Fig. 5. And the actual and predicted outputs of the plant for the test signal are drawn in Fig. 6. As shown in Fig. 5, due to the suitable initialization of network in the stage of two-layer inline-PSO, whose adjustment scheme is coordinate with following gradient descent, the RMSE values can reduce smoothly with iteration when small learning rates are adopted. The dashed line of RMSE during testing indicates the rationality of the model.

The RMSE values of proposed-FWNN-based identification system for training and testing data are illustrated in Table 1, which give the results of other models as well. From Table 1, it can be seen that the proposed-FWNN model shows better performance than the other models in this system identification problem even though fewer fuzzy rules are employed.

Example 2

In this example, the second-order nonlinear plant to be identified is described as

$$y(k+1)=f(y(k),y(k-1),y(k-2),u(k),u(k-1)), \tag{31}$$

where

$$f(x_1,x_2,x_3,x_4,x_5)=\frac{x_1x_2x_3x_5(x_3-1)+x_4}{1+x_3^2+x_2^2}. \tag{32}$$

The identification of the same plant are considered in [34,35,18]. The current output of the plant depends on three previous output values and two previous input values. However, we only adopt the current state of system and the control signal feeding into the proposed-FWNN as inputs. The testing signal is adopted as in (30) too. The number of fuzzy rules and wavelet neurons in each WNN are same as example 1 and thus the number of adjustable parameters is $N=30$ too. The hybrid learning algorithm stated in Section "Hybrid learning algorithm to optimize the proposed-FWNN" is used to train the proposed-FWNN. The acceleration coefficients

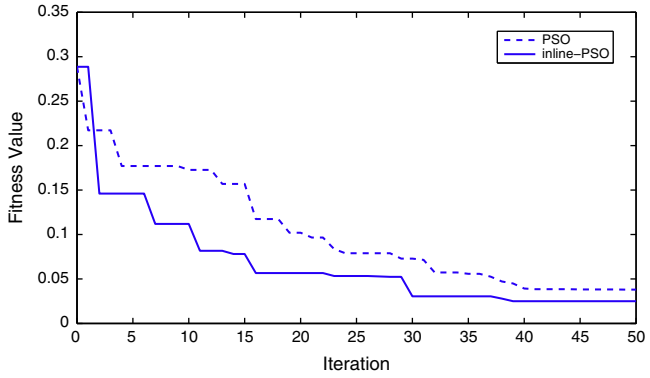

Fig. 4. Fitness values obtained during iteration of two PSO methods for Example 1.

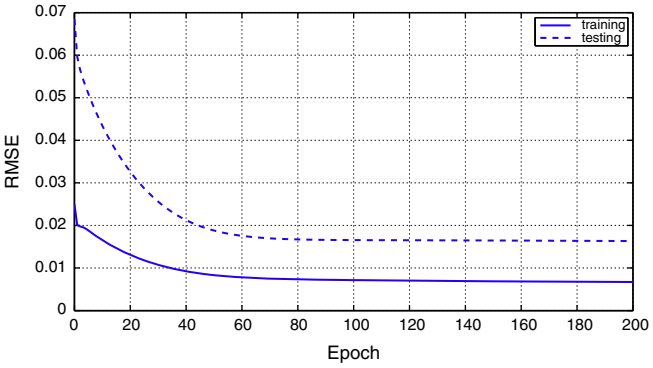

Fig. 5. RMSE values obtained during training and testing for Example 1.

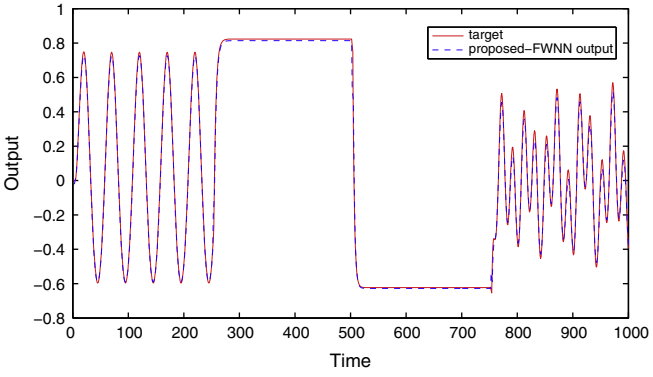

Fig. 6. Results of identification for Example 1.

Table 1
comparison of simulation results of different models for Example 1.

Model	Number of fuzzy rules	Number of parameters	RMSE of training	RMSE of testing
ERNN [36]	–	54	0.036	0.078
RSONFIN [35]	–	49	0.03	0.06
TRFN-S [34]	3	33	0.0067	0.0313
FWNN [18]	3	27	0.019736	0.022609
FWNN [18]	5	43	0.018713	0.020169
proposed-FWNN	2	30	0.0067	0.0163

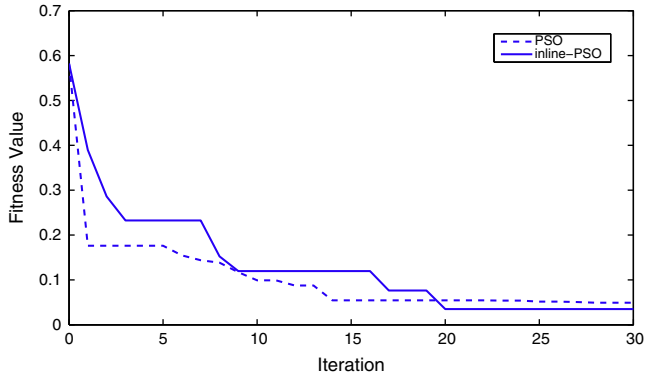

Fig. 7. Fitness values obtained during iteration of two PSO methods for Example 2.

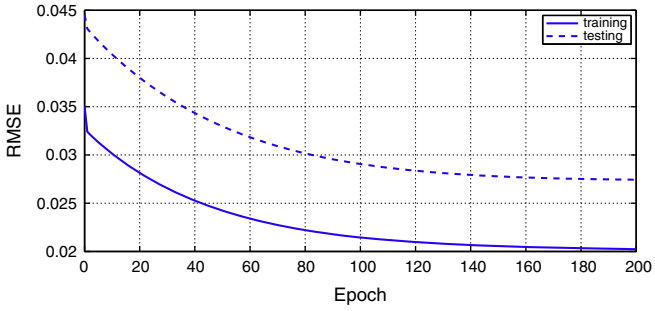

Fig. 8. RMSE values obtained during training and testing for Example 2.

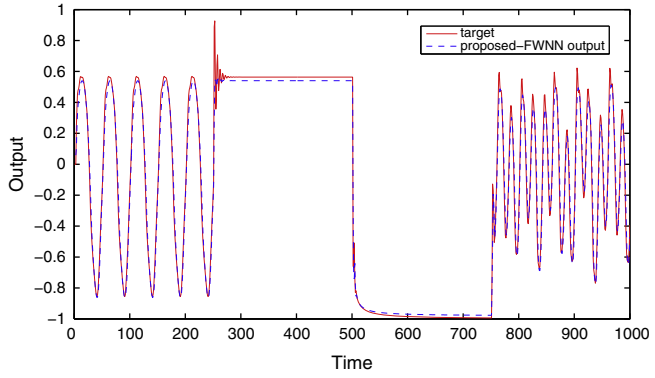

Fig. 9. Results of identification for Example 2.

Table 2
Comparison of simulation results of different models for Example 2.

Model	Number of fuzzy rules	Number of parameters	RMSE of training	RMSE of testing
RFNN [37]	16	112	0.0114	0.0575
RSONFIN [35]	–	36	0.0248	0.0780
TRFN-S [34]	3	33	0.0084	0.0346
FWNN [18]	3	27	0.029179	0.031212
FWNN [18]	5	43	0.028232	0.030125
proposed-FWNN	2	30	0.0202	0.0274

in inline-PSO and basic PSO are $c_1 = c_2 = 2$, and the inertia weight ω is linear decreasing during iteration. The population size $psize = 20$ and the termination iterative number $Maxgen = 30$, which are set relatively small for the same reason sated in the previous example. The optimization results of inline-PSO and basic PSO are compared and illustrated in Fig. 7, which shows obviously that the convergence of inline-PSO is faster than that of PSO, which reach the fitness values of 0.0350 and 0.0492 respectively.

Fig. 8 displays the RMSE reduction curve during training and testing of gradient descent algorithm in the consequent part of hybrid learning scheme. And the actual and predicted outputs of the plant for the test signal are drawn in Fig. 9. Table 2 compares the RMSE values of proposed-FWNN-based identification system with other approaches reported in the other literatures. As it is seen, the proposed-FWNN models are again successful in identification than the other models even though fewer fuzzy rules are employed.

Conclusion

In this paper, a novel approach to fuzzy wavelet neural network modeling as well as optimization algorithm is proposed. In the structure design aspect of proposed-FWNN, the TSK fuzzy logic is combined with wavelet neural network which derive a fuzzy partition of input space into different wavelet based subspaces. In the training aspect of proposed-FWNN, a hybrid training algorithm combining a so called inline-PSO and gradient-based algorithm is proposed. The proposed inline-PSO finds the optimal solution by adjusting parameters after each measurement as well as that contains all vectors, which corresponds to the adjustment scheme of the consequent gradient descent algorithm and could make a faster convergence. In both two simulation examples of system identification, proposed-FWNN gets a better performance than other models, despite fewer rules and smaller number of parameters.

Acknowledgment

The authors are thankful that the research is supported by the National Science Foundations of China (61275120).

References

[1] Hecht-Nielsen Robert. Theory of the backpropagation neural network. In: International Joint Conference on Neural Networks, 1989. IJCNN. IEEE; 1989.
[2] Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. IEEE Trans Neural Networks 1990;1(1):4–27.
[3] Azoff EM. Neural network time series forecasting of financial markets. John Wiley & Sons; 1994.
[4] Zhang J, Walter GG, Miao Y, et al. Wavelet neural networks for function learning. IEEE Trans Signal Process 1995;43(6):1485–97.
[5] Zhang Q, Benveniste A. Wavelet networks. IEEE Trans Neural Networks 1992;3(6):889–98.
[6] Pati YC, Krishnaprasad PS. Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. IEEE Trans Neural Networks 1993;4(1):73–85.
[7] Lin CJ. Nonlinear systems control using self-constructing wavelet networks. Appl Soft Comput 2009;9(1):71–9.
[8] Chen Y, Yang B, Dong J. Time-series prediction using a local linear wavelet neural network. Neurocomputing 2006;69(4):449–65.
[9] Jiao L, Pan J, Fang Y. Multiwavelet neural network and its approximation properties. IEEE Trans Neural Networks 2001;12(5):1060–6.
[10] Wang LX. A course in fuzzy systems and control. Prentice-Hall International, Inc.; 1997.
[11] Mendel JM. Fuzzy logic systems for engineering: a tutorial. Proc IEEE 1995;83(3):345–77.
[12] Horikawa S, Furuhashi T, Uchikawa Y. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm. IEEE Trans Neural Networks 1992;3(5):801–6.
[13] Kim J, Kasabov N. HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. Neural Networks 1999;12(9):1301–19.
[14] Zhang J, He ZY, Lin S, et al. An ANFIS-based fault classification approach in power distribution system. Int J Electr Power Energy Syst 2013;49:243–52.

[15] Li CS, Chiang TW, Yeh LC. A novel self-organizing complex neuro-fuzzy approach to the problem of time series forecasting. Neurocomputing 2013;99:467–76.
[16] Ho DWC, Zhang PA, Xu J. Fuzzy wavelet networks for function learning. IEEE Trans Fuzzy Syst 2001;9(1):200–11.
[17] Adeli H, Jiang X. Dynamic fuzzy wavelet neural network model for structural system identification. J Struct Eng 2006;132(1):102–11.
[18] Abiyev RH, Kaynak O. Fuzzy wavelet neural networks for identification and control of dynamic plants a novel structure and a comparative study. IEEE Trans Ind Electron 2008;55(8):3133–40.
[19] Tzeng ST. Design of fuzzy wavelet neural networks using the GA approach for function approximation and system identification. Fuzzy Sets Syst 2010;161(19):2585–96.
[20] Bodyanskiy Y, Vynokurova O. Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification. Inform Sci 2013;220:170–9.
[21] Kennedy J, Eberhart R. Particle swarm optimization. Proc IEEE Int Conf Neural Networks 1995;4(2):1942–8.
[22] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, vol. 1. 1995. p. 39–43.
[23] Park JB, Lee KS, Shin JR, et al. A particle swarm optimization for economic dispatch with nonsmooth cost functions. IEEE Trans Power Syst 2005;20(1):34–42.
[24] Eberhart RC, Shi Y. Tracking and optimizing dynamic systems with particle swarms. Proceedings of the 2001 congress on evolutionary computation, 2001, vol. 1. IEEE; 2001. p. 94–100.
[25] Kamyab GR, Fotuhi-Firuzabad M, Rashidinejad M. A PSO based approach for multi-stage transmission expansion planning in electricity markets. Int J Electrical Power Energy Syst 2014;54:91–100.
[26] Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 2004;8(3):256–79.
[27] Juang CF. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans Syst Man Cybern Part B: Cybern 2004;34(2):997–1006.
[28] Daubechies I. Ten lectures on wavelets. Philadelphia: Society for Industrial and Applied Mathematics; 1992.
[29] Mallat SG. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans Pattern Anal Mach Intell 1989;11(7):674–93.
[30] Grossmann A, Morlet J. Decomposition of Hardy functions into square integrable wavelets of constant shape. SIAM J Math Anal 1984;15(4):723–36.
[31] Lin CJ, Chin CC, Lee CL. A wavelet-based neuro-fuzzy system and its applications. Proceedings of the international joint conference on neural networks, vol. 3. Springer; 2003.
[32] Shi Y, Eberhart R. A modified particle swarm optimizer. In: The 1998 IEEE international conference on evolutionary computation proceedings, 1998. IEEE world congress on computational intelligence. IEEE; 1998. p. 69–73.
[33] Shi Y, Eberhart RC. Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99, vol. 3. IEEE; 1999.
[34] Juang CF. A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. IEEE Trans Fuzzy Syst 2002;10(2):155–70.
[35] Juang CF, Lin CT. A recurrent self-organizing neural fuzzy inference network. IEEE Trans Neural Networks 1999;10(4):828–45.
[36] Elman JL. Finding structure in time. Cognitive Sci 1990;14(2):179–211.
[37] Lee CH, Teng CC. Identification and control of dynamic systems using recurrent fuzzy neural networks. IEEE Trans Fuzzy Syst 2000;8(4):349–66.