Paper

The Local Linear Adaptive Wavelet Neural Network with Hybrid EP/Gradient Algorithm and Its Application to Nonlinear Dynamic System Identification

Student Member Ting Wang (Chiba University)
Member Yasuo Sugai (Chiba University)

Wavelet neural networks are networks employing nonlinear wavelet basis functions as the activation functions of the neurons. This paper presents a new type of wavelet-based neural network: the local linear adaptive wavelet neural network. A hybrid evolutionary programming and gradient descent algorithm is introduced to the learning of the proposed network. The local linear models which are used in some neuro-fuzzy systems are introduced as powerful weights instead of straightforward weights employed in the previous wavelet neural networks. Training is performed by using the evolutionary programming algorithm at first to search a good region in the parameter space and then employing the gradient descent algorithm to find a near optimal solution in that region. The experiments on a number of nonlinear dynamic system identification problems indicates that the proposed network with the hybrid EP/Gradient algorithm can successfully identify and describe the input/output relationship for an unknown complex system with a small number of wavelet basis functions and compared favorably to the traditional neural networks with the sigmoid activation functions and the previous wavelet neural networks with straightforward weights.

Keywords:
local linear adaptive wavelet neural network, evolutionary programming, gradient descent algorithm, nonlinear dynamic system identification

1. Introduction

Due to (i)the ability to learn from the experiences, (ii)generalization for untrained inputs, and (iii)the capability to approximate to arbitrary specified accuracy given sufficient number of neurons, artificial neural networks(NN) have been established as a general nonlinear fitting tool to develop models from observed data, or to learn maps between input and output spaces [1][2]. The marked characteristics are the same as those of interest to researchers in the areas of dynamical system control, signal processing, system identification and many other fields [3][4]. Recently, in stead of using the common sigmoid activation functions, by employing nonlinear wavelet basis functions(called wavelets) which are localized in both the time space and frequency space, the wavelet neural network(WNN) has been developed as an alternative approach to nonlinear fitting problems. Since Zhang and Benveniste first introduced wavelets to a three layer feedforward neural network in (5), several researches on this kind of neural network have been done [6]~[9]. The output of a WNN is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \Psi_i(\boldsymbol{x}) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (1)$$

where Ψ_i is the wavelet activation function of the i-th unit of the hidden layer and w_i is the weight connecting the i-th unit of the hidden layer to the output layer unit. With various wavelets Ψ used as activation functions and gradient descent based training algorithms, all of these previous studies have successfully demonstrated the power of WNNs when employed to approximate nonlinear functions [6]~[9]. However, a large number of basis function units has to be employed and the gradient descent based algorithms have some limitations as pointed in the studies of the back-propagation learning algorithms [7].

In this paper, we introduce local linear models whose good performances have been shown in some neuro-fuzzy systems [10], as powerful weights to an adaptive wavelet neural network(AWNN). The local linear models connect the hidden layer with the output layer instead of the previous straightforward weights w_i in (1), therefore this type of wavelet neural network is called a local linear adaptive wavelet neural network(LLAWNN) [11]. Moreover, a hybrid evolutionary programming and gradient descent algorithm is employed to train this LLAWNN. Evolutionary programming(EP) algorithm is used first to locate a good region in the parameter space and then gradient descent algorithm, the local search procedure, is employed to find a near optimal solution in that region. Due to WNNs have already shown the power on nonlinear function approximation problem which is a fundamental problem in the areas such as signal processing, system identification and many other complex engineering fields, the LLAWNN here is employed to the application of nonlinear dynamic system identification.

The paper is organized as follows. The topology of the LLAWNN network is introduced in section 2. The hybrid evolutionary programming and gradient descent algorithm is described in section 3. The experiments on nonlinear dynamic system identification problems are described in section 4. Finally, conclusions are derived in the last section.

2. The Local Linear Adaptive Wavelet Neural Network

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

$$\Psi = \{\Psi_i = |\boldsymbol{a}_i|^{-\frac{1}{2}}\psi(\frac{\boldsymbol{x}-\boldsymbol{b}_i}{\boldsymbol{a}_i}) : \boldsymbol{a}_i, \boldsymbol{b}_i \in \boldsymbol{R}, i \in Z\} \quad (2)$$

$$\boldsymbol{x} = (x_1, \cdots, x_N)$$

$$\boldsymbol{a}_i = (a_{i1}, \cdots, a_{iN})$$

$$\boldsymbol{b}_i = (b_{i1}, \cdots, b_{iN})$$

is a family of functions generated from one single function $\psi(\boldsymbol{x})$ by the operation of dilations and translations. $\psi(\boldsymbol{x})$, which is localized in both the time space and frequency space, is usually called a mother wavelet and the parameters $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are called the scale parameter and translation parameter, respectively [12].

With Eq.(2), the output of a WNN given by (1) can be rewrote to

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \Psi_i(\boldsymbol{x}) = \sum_{i=1}^{M} w_i |\boldsymbol{a}_i|^{-\frac{1}{2}}\psi(\frac{\boldsymbol{x}-\boldsymbol{b}_i}{\boldsymbol{a}_i}) \quad \cdots\cdots (3)$$

It is obviously, the localization of the i-th unit of the hidden layer is determined by the scale parameter $\boldsymbol{a}_i$ and the translation parameter $\boldsymbol{b}_i$. According to the previous researches, the two parameters can either be predetermined based upon the wavelet transformation theory [6]~[8] or be determined by training [9]. Therefore, an adaptive wavelet neural network is a WNN of which not only the connection weights w_i's, but also the scale parameters $\boldsymbol{a}_i$'s and the translation parameters $\boldsymbol{b}_i$'s in (3) are determined by some training algorithms based on the given training data set [5] [9]. "Adaptive" means the hidden layer units can adapt their receptive fields to the distribution of the input vectors during the training process.

Due to the network architecture and the localized validity of the wavelets, the AWNN (3) can be viewed as a kind of standard basis function network. The output of a standard basis function network given as :

$$y = \sum_{i=1}^{M} w_i \Phi_i(\boldsymbol{x}) \cdots\cdots (4)$$

is a weighted linear combination of many locally active non-linear basis functions $\Phi_i (i = 1, \cdots, M)$, where w_i is the associated weight with Φ_i [13]. In the AWNN (3), it is obviously that the wavelets Ψ are the corresponding locally active non-linear basis functions.

It is well known 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 [14]. 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 (piecewise constant models are integrated by their associated localized basis functions), a large number of basis function units has to be employed to approximate a given system. As reported in the previous research (7), a shortcoming of the wavelet neural network also shared by the RBF network and other basis function networks, 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 to use too many hidden units, we propose an alternative type of wavelet neural network. Its output of the k-th unit in the output layer is given as :

$$y_k = \sum_{i=1}^{M} (w_{i,0} + w_{i,1}x_1 + \cdots + w_{i,N}x_N)\Psi_i(\boldsymbol{x}) \quad (5)$$

$$\boldsymbol{x} = (x_1, \cdots, x_N)$$

where, instead of the straightforward weight w_i (piecewise constant model) in (3), the following linear model

$$u_i = w_{i,0} + w_{i,1}x_1 + \cdots + w_{i,N}x_N \cdots\cdots (6)$$

is introduced as a powerful representation of weights. Because the activities of the linear models u_i's$(i = 1, \cdots, M)$ are determined by the associated locally active wavelet functions Ψ_i's$(i = 1, \cdots, M)$, u_i is locally valid so that it is called a local linear model.

The idea of introducing local linear models to wavelet neural network is inspired by the researches of some neuro-fuzzy systems , where the local linear neuro-fuzzy model described as :

$$y = \sum_{j=1}^{M} (w_{j,0} + w_{j,1}x_1 + \cdots + w_{j,N}x_N)\Phi_j(\boldsymbol{x})$$

$$\boldsymbol{x} = (x_1, \cdots, x_N) \cdots\cdots (7)$$

has been studied and shown good performances [10] [15]. The following given Gaussian-based functions $\Phi_j (j = 1, \cdots, M)$ are the basis functions used in these neuro-fuzzy systems to control the activity of the local linear models.

$$\Phi_j = D_j / \sum_{j=1}^{M} D_j \cdots\cdots (8)$$

where

$$D_j = \exp(-d(\boldsymbol{x}; \boldsymbol{c}_j, \sigma_j)) \cdots\cdots (9)$$

identification.

The paper is organized as follows. The topology of the LLAWNN network is introduced in section 2. The hybrid evolutionary programming and gradient descent algorithm is described in section 3. The experiments on nonlinear dynamic system identification problems are described in section 4. Finally, conclusions are derived in the last section.

2. The Local Linear Adaptive Wavelet Neural Network

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

$$\Psi = \{\Psi_i = |\boldsymbol{a}_i|^{-\frac{1}{2}}\psi(\frac{\boldsymbol{x}-\boldsymbol{b}_i}{\boldsymbol{a}_i}) : \boldsymbol{a}_i, \boldsymbol{b}_i \in \boldsymbol{R}, i \in Z\} \quad (2)$$

$$\boldsymbol{x} = (x_1, \cdots, x_N)$$

$$\boldsymbol{a}_i = (a_{i1}, \cdots, a_{iN})$$

$$\boldsymbol{b}_i = (b_{i1}, \cdots, b_{iN})$$

is a family of functions generated from one single function $\psi(\boldsymbol{x})$ by the operation of dilations and translations. $\psi(\boldsymbol{x})$, which is localized in both the time space and frequency space, is usually called a mother wavelet and the parameters $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are called the scale parameter and translation parameter, respectively [12].

With Eq.(2), the output of a WNN given by (1) can be rewrote to

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \Psi_i(\boldsymbol{x})$$
$$= \sum_{i=1}^{M} w_i |\boldsymbol{a}_i|^{-\frac{1}{2}}\psi(\frac{\boldsymbol{x}-\boldsymbol{b}_i}{\boldsymbol{a}_i}) \quad \cdots\cdots (3)$$

It is obviously, the localization of the i-th unit of the hidden layer is determined by the scale parameter $\boldsymbol{a}_i$ and the translation parameter $\boldsymbol{b}_i$. According to the previous researches, the two parameters can either be predetermined based upon the wavelet transformation theory [6]~[8] or be determined by training [9]. Therefore, an adaptive wavelet neural network is a WNN of which not only the connection weights w_i's, but also the scale parameters $\boldsymbol{a}_i$'s and the translation parameters $\boldsymbol{b}_i$'s in (3) are determined by some training algorithms based on the given training data set [5] [9]. "Adaptive" means the hidden layer units can adapt their receptive fields to the distribution of the input vectors during the training process.

Due to the network architecture and the localized validity of the wavelets, the AWNN (3) can be viewed as a kind of standard basis function network. The output of a standard basis function network given as :

$$y = \sum_{i=1}^{M} w_i \Phi_i(\boldsymbol{x}) \cdots\cdots (4)$$

is a weighted linear combination of many locally active non-linear basis functions $\Phi_i (i = 1, \cdots, M)$, where w_i is the associated weight with Φ_i [13]. In the AWNN (3), it is obviously that the wavelets Ψ are the corresponding locally active non-linear basis functions.

It is well known 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 [14]. 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 (piecewise constant models are integrated by their associated localized basis functions), a large number of basis function units has to be employed to approximate a given system. As reported in the previous research (7), a shortcoming of the wavelet neural network also shared by the RBF network and other basis function networks, 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 to use too many hidden units, we propose an alternative type of wavelet neural network. Its output of the k-th unit in the output layer is given as :

$$y_k = \sum_{i=1}^{M} (w_{i,0} + w_{i,1}x_1 + \cdots + w_{i,N}x_N)\Psi_i(\boldsymbol{x}) \quad (5)$$
$$\boldsymbol{x} = (x_1, \cdots, x_N)$$

where, instead of the straightforward weight w_i (piecewise constant model) in (3), the following linear model

$$u_i = w_{i,0} + w_{i,1}x_1 + \cdots + w_{i,N}x_N \cdots\cdots (6)$$

is introduced as a powerful representation of weights. Because the activities of the linear models u_i's$(i = 1, \cdots, M)$ are determined by the associated locally active wavelet functions Ψ_i's$(i = 1, \cdots, M)$, u_i is locally valid so that it is called a local linear model.

The idea of introducing local linear models to wavelet neural network is inspired by the researches of some neuro-fuzzy systems , where the local linear neuro-fuzzy model described as :

$$y = \sum_{j=1}^{M} (w_{j,0} + w_{j,1}x_1 + \cdots + w_{j,N}x_N)\Phi_j(\boldsymbol{x})$$
$$\boldsymbol{x} = (x_1, \cdots, x_N) \cdots\cdots (7)$$

has been studied and shown good performances [10] [15]. The following given Gaussian-based functions $\Phi_j (j = 1, \cdots, M)$ are the basis functions used in these neuro-fuzzy systems to control the activity of the local linear models.

$$\Phi_j = D_j / \sum_{j=1}^{M} D_j \cdots\cdots (8)$$

where

$$D_j = \exp(-d(\boldsymbol{x}; \boldsymbol{c}_j, \sigma_j)) \cdots\cdots (9)$$

in which

$$d(\boldsymbol{x};\boldsymbol{c}_j,\sigma_j)=\left\|\frac{\boldsymbol{x}-\boldsymbol{c}_j}{\sigma_j}\right\|^2 \cdots\cdots\cdots\cdots\cdots\cdots (10)$$

Similar to the fact that Eg.(7) is called the local linear neuro-fuzzy model, we call the proposed network (5) the local linear adaptive wavelet neural network [11]. Its structure is shown in Fig.1.

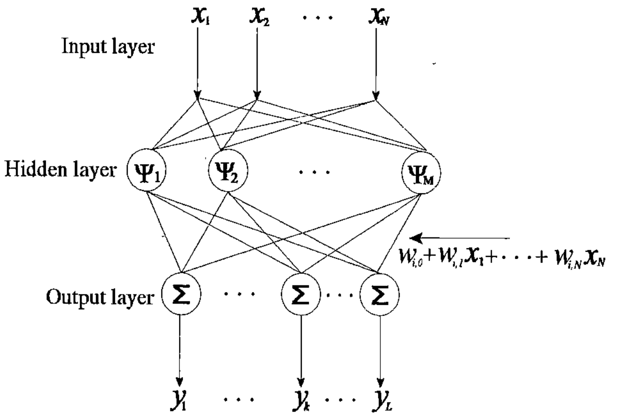


Fig. 1. Network structure

It has a feedforward structure consisting of a single hidden layer. The hidden layer performs nonlinear transformation via the activation functions which are the wavelets and the weights connecting the hidden layer units to the output layer units are the local linear models (6). The working 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. It means that this structure has the advantages inherent to the local nature of the wavelet basis functions while, by employing the more powerful local models (6) associated with those locally basis functions, it is not requiring as many basis functions as before to achieve the desired accuracy.

As mentioned above, various wavelets have been used as activation functions. In this study, for a problem with N inputs, the mother wavelet $\psi(\boldsymbol{x})$ with which wavelets are generated by the operation of dilations and translations, is given as follows :

$$\psi(\boldsymbol{x})=\prod_{n=1}^{N}\psi(x_n) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (11)$$

this is the most frequently chosen scheme to generate a multidimensional wavelet function by the tensor product of a one dimensional wavelet function. The basic one dimensional mother wavelet $\psi(x)$ we used is described as :

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

Therefore, with Eq.(2), wavelets used as basis functions which has N inputs, can be generated from (11) as

$$\Psi_i(\boldsymbol{x})=\prod_{n=1}^{N}|a_{in}|^{-\frac{1}{2}}\psi\left(\frac{x_n-b_{in}}{a_{in}}\right). \cdots\cdots\cdots (13)$$

3. Hybrid EP/Gradient Training Algorithm

Given a set of training data T_P,

$$T_P=\{(\boldsymbol{x}_i,\boldsymbol{f}(\boldsymbol{x}_i)),i=1,\cdots,P\},\cdots\cdots\cdots (14)$$

where $\boldsymbol{x}_i$, $\boldsymbol{f}(\boldsymbol{x}_i)$ are the input vector and the corresponding output vector, the training of the network can be formulated as minimization of an error goal function, such as the mean square error between the target and the actual outputs over all training examples by iteratively adjusting the parameters $\boldsymbol{w}_i$, $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ in (5). Various gradient descent based training algorithms have shown their effectiveness in previous studies on WNN. However, problems such as local minimum or sensitivity to initial conditions are still remained due to the nature of the gradient descent [5]. On the other hand, including genetic algorithm(GA), evolutionary programming(EP) and evolution strategies(ES), evolutionary computation based on the genetic process of biological organisms has been used effectively in the training of the traditional neural networks as a global searching method [17]~[19]. However most evolutionary algorithms(EAs) are rather inefficient in fine-tuned local search [20]. To take advantages of both methods to train the proposed LLAWNN, we introduce a hybrid evolutionary programming and gradient descent algorithm. EP algorithm is used first to locate a good region in the parameter space and then gradient descent algorithm, the local search procedure, is employed to find a near optimal solution in that region. Successful results of hybrid training have been reported in some researches of the training of the traditional neural networks where GAs were used to search for a near optimal set of initial connection weights and then BP was used to perform local search from these initial weights [22][23].

A. Evolutionary Programming

Evolutionary programming, in contrast to GA, is based on the assumption that evolution optimizes the behavior of an instance, but not the underlying genetic code. Therefore, the primary search operator in EP is mutation. Gaussian mutation has been the most commonly used mutation operator, but Cauchy mutation and other mutation operators can also be used. In this study, we employ the Gaussian mutation operator as the primary search operator. Similar to the studies of applying EP to the evolution of the traditional NN's connection weights [20][21], the procedure of evolving the adjustable parameters in the proposed LLAWNN (5) is given as follows :

(1) Generate an initial population of L individuals at random and set *generation* = 1. Each individual is a pair of real valued vectors $(\boldsymbol{\theta}_i,\boldsymbol{\eta}_i)(i=1,\cdots,L)$, where vector $\boldsymbol{\theta}$ is the collection of all of

the parameters $\boldsymbol{w}_i$, $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ in (5) that should be adjusted, $\boldsymbol{\eta}_i$'s are variance vectors for Gaussian mutations.

(2) Evaluate the fitness score for each individual of the population by the inverse of the following defined error function

$$E(\boldsymbol{\theta}) = \frac{1}{P}\sum_{p=1}^{P}(\boldsymbol{f}_p - \boldsymbol{Y}_p)^2, \cdots\cdots\cdots\cdots \quad (15)$$

where $\boldsymbol{f}_p$ and $\boldsymbol{Y}_p$ are the teaching signal and the output of the network (5), respectively.

(3) A single offspring $(\boldsymbol{\theta}_i', \boldsymbol{\eta}_i')(i = 1, \cdots, L)$ is created from each individual $(\boldsymbol{\theta}_i, \boldsymbol{\eta}_i)(i = 1, \cdots, L)$ as the following :

$$\eta_{i,j}' = \eta_{i,j}\exp(\frac{1}{\sqrt{2\sqrt{n}}}N(0,1) + \frac{1}{\sqrt{2n}}N_j(0,1)) \quad (16)$$

$$\theta_{i,j}' = \theta_{i,j} + \eta_{i,j}'N_j(0,1), \cdots\cdots\cdots\cdots\cdots\cdots \quad (17)$$

where $\theta_{i,j}$, $\theta_{i,j}'$, $\eta_{i,j}$, and $\eta_{i,j}'$ are the j-th component of the vectors $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_i'$, $\boldsymbol{\eta}_i$, and $\boldsymbol{\eta}_i'$, respectively. n is the total number of the component in $\boldsymbol{\theta}$. In fact, here n is the total number of the adjustable parameters in the proposed network (5). $N(0,1)$ denotes a normally distributed one dimensional random number with mean 0 and variance 1. $N_i(0,1)$ indicates that the random number is generated anew for each component j.

(4) Calculate the fitness of each offspring as the same as step 2.

(5) Perform pairwise comparison over the union of parents $(\boldsymbol{\theta}_i, \boldsymbol{\eta}_i)(i = 1, \cdots, L)$ and offspring $(\boldsymbol{\theta}_i', \boldsymbol{\eta}_i')(i = 1, \cdots, L)$. For each individual, q opponents are chosen uniformly at random from all of the parents and offspring. For each comparison, if the individual's fitness is no smaller than the opponent's, it receives a "win".

(6) Select L individuals out of all of the parents and the offspring which have most wins to be the parents of the next generation.

(7) Stop if the terminating criterion is satisfied; otherwise set *generation* = *generation* + 1 and go to step 3.

B. Gradient Descent Algorithm

Given the objective function to be minimized as

$$E'(\boldsymbol{\theta}) = \frac{1}{2}\sum_{p=1}^{P}(\boldsymbol{f}_p - \boldsymbol{Y}_p)^2, \cdots\cdots\cdots\cdots\cdots\cdots \quad (18)$$

where as the same as in EP, $\boldsymbol{\theta}$ is the collection of all of the parameters $\boldsymbol{w}_i$, $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ in (5) that should be adjusted. $\boldsymbol{f}_p$ and $\boldsymbol{Y}_p$ are the teaching signal and the output of the network (5), respectively. Then $\boldsymbol{\theta}$ is updated from step t to the next step as follows :

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \Delta\boldsymbol{\theta}(t), \cdots\cdots\cdots\cdots\cdots\cdots \quad (19)$$

$$\Delta\boldsymbol{\theta}(t) = -\eta\frac{\partial E'}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}(t)} + \mu\Delta\boldsymbol{\theta}(t-1), \cdots\cdots \quad (20)$$

where the learning rate η scales the stepsize and μ is the momentum term added to make the training process more stable.

Combining the EP's global search ability with the gradient descent's local search ability, the flow chat of the hybrid EP/Gradient training algorithm is illustrated in Fig.2.

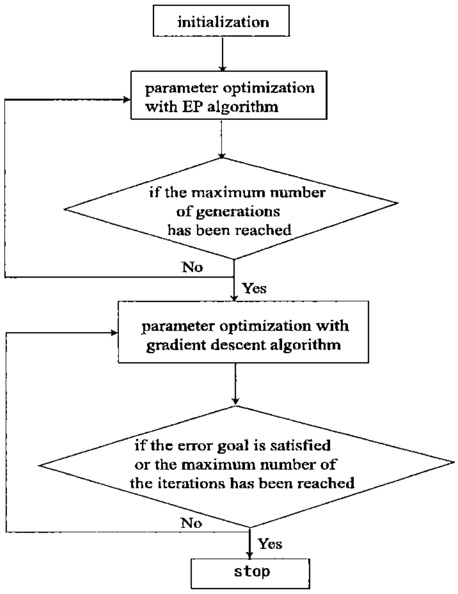


Fig. 2. Flow chat of the hybrid EP/Gradient algorithm

4. Experimental Results

The system identification, which is an important prerequisite for a successful analysis and controller design, is a process aimed to identify and describe the input/output relationship for unknown systems. Because of the nonlinear nature of most of the processes encountered in engineering applications, the problem of nonlinear system identification using neural networks has been extensively researched [(24)~(26)]. The model of a discrete-time dynamic nonlinear plant can be described as :

$$y_p(k+1) = f[y_p(k), y_p(k-1), \cdots, y_p(k-n+1); u(k), u(k-1), \cdots, u(k-m+1)] \quad (21)$$

where $u(k)$ and $y_p(k)$ represent the input and output of the plant at the k-th time instant. m and n are the order of u and y_p, $m \leq n$.

In this section the ability of the LLAWNN to identify the nonlinear dynamics of various plants has been tested in simulation. We show the scheme of nonlinear system identification with the LLAWNN in Fig.3.

The performance index used is the mean square error(MSE):

$$\text{MSE} = \frac{1}{NT}\sum_{k=1}^{NT}(\hat{y}_p(k) - y_p(k))^2 \cdots\cdots\cdots\cdots \quad (22)$$

where $y_p(k)$ and $\hat{y}_p(k)$ are the desired output and the

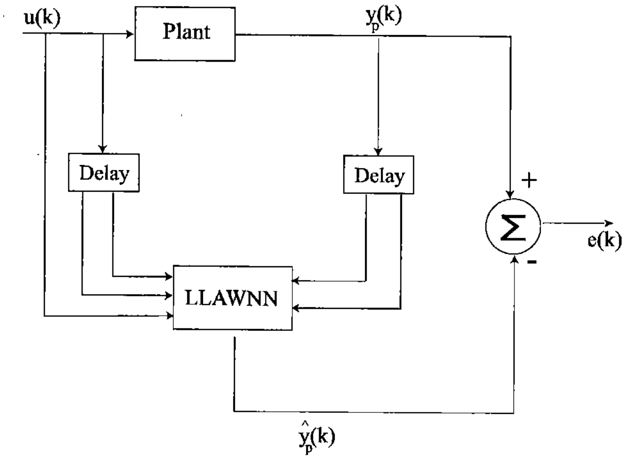


Fig. 3. System identification using LLAWNN

network output of the plant at time k, respectively. NT represents the number of data in the testing data.

A. Example 1

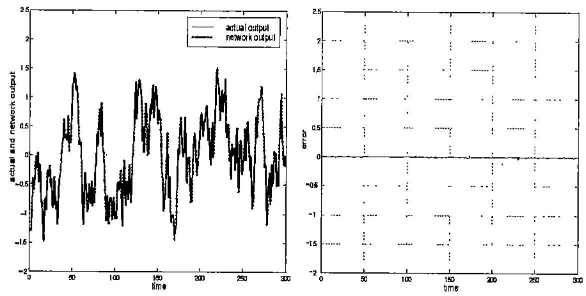
This example is a first-order plant which is assumed to be of the form

$$y_p(k+1) = f[y_p(k), u(k)]. \cdots\cdots\cdots\cdots\cdots\cdots \quad (23)$$

The unknown function f has the form

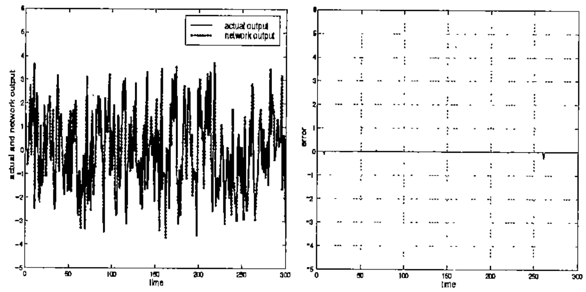
$$f[x_1, x_2] = \sin(x_1) + x_2 * (5 + \cos(x_1 * x_2)) \quad (24)$$

The identification of this plant is reported in Narendra (25) where the linear model and traditional neural network model were employed. In our study, the LLAWNN used for this first order plant identification problem has a $\{2-12-1\}$ structure. The two inputs are $y_p(k)$ and $u(k)$. $u(k)$ is a uniformly distributed random sequence. Similar to (25), three different sets of experiments were carried out with $|u(k)| \leq 0.1, 0.5$, and 1 to study the relative performance of the network with the increasing amplitudes of input and output. In all of the three cases, the number of data used for the training was 1000 and the testing was undertaken by the following 300 points. The algorithm parameters are set as follows : L(population size) = 20, q(tournament size) = 10, and after 200 generations of the evolutionary programming training, the gradient descent algorithm were performed with η(learning rate)= 0.05 and μ(momentum term) = 0.75. For each case of $u(k)$, 10 experiments with different initializations had been done and the averaged MSE of the testing data are 0.000032, 0.00036 and 0.0045 respectively. The identification results with the three input signal $u(k)$ are illustrated in Fig.4~6. The corresponding MSE of each figure is 0.000031(Fig.4), 0.00034(Fig.5) and 0.0041(Fig.6), which is the nearest one to the average of each $u(k)$. It is clear that LLAWNN performed well for the identification of all the three cases. With the $\{2-12-1\}$ structure, the total number of the adjustable parameters in the LLAWNN is 84, where in (25), the traditional neural network employed and shown effectiveness for the identification of the plant has two hidden layers with 20 units in the first hidden layer and 10 units in the second hidden layer.



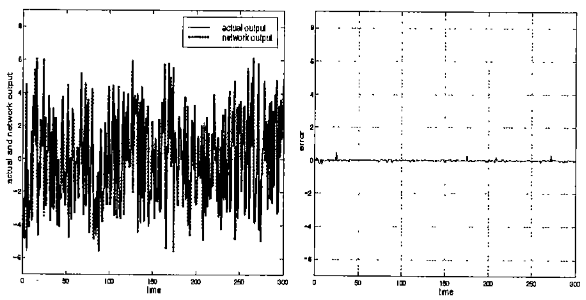
Network and actual output Error of the network output

Fig. 4. Result of $|u(k)| \leq 0.1$



Network and actual output Error of the network output

Fig. 5. Result of $|u(k)| \leq 0.5$



Network and actual output Error of the network output

Fig. 6. Result of $|u(k)| \leq 1$

Table 1. Identification results of $|u| \leq 0.1$

	network structure	number of parameters	training iterations
LLAWNN	2-12-1	84	472
AWNN	2-21-1	105	1803

To further illustrate the performance of the LLAWNN comparing with a previous WNN, we consider the first case(i.e., $|u| \leq 0.1$) in more detail. The compared WNN is the above described network (1) with the same wavelets of the LLAWNN as basis functions. It has straightforward weights and is trained with the gradient descent algorithm. Given a targeted MSE$\leq 2\times10^{-4}$ for the training data, performance comparison results of the two networks are given in Tbl.1. The training iterations of the both networks are the iterations of the gradient descent algorithm and the results are the average of 10 different experiments. According to our preparatory experiment, the calculation time of 200 generations of EP algorithm is as long as about 800 iterations of the gradient descent algorithm. Therefore, it can be seen that the hybrid algorithm made the training of the LLAWNN to converge to the desired error tolerance with few iterations. Moreover, in terms of employing the local linear models (6) as the powerful weights associated with the locally active basis functions instead of the straightforward weights w_i, the LLAWNN can learn the input/output relationship for an unknown dynamic nonlinear system with a smaller number of basis functions with sufficient accuracy.

B. Example 2

The plant is expressed as

$$x_1(k+1) = x_2(k), \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (25)$$

$$x_2(k+1) = \frac{3u(k)x_1(k)x_2(k) + 0.5e^{u(k)}}{1 + x_1^2(k) + x_2^2(k)} \cdots \quad (26)$$

Let

$$y_p(k+1) = x_2(k+1), \cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (27)$$

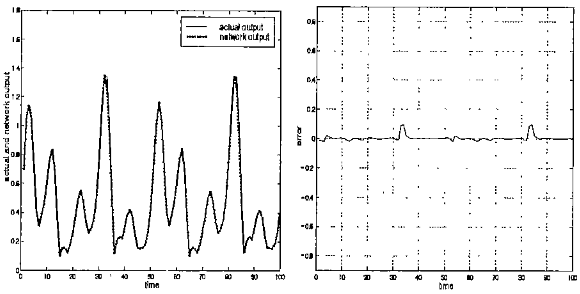
therefore, using the above three equations, rewrite the plant to be identified as :

$$\begin{aligned} y_p(k+1) &= f[y_p(k), y_p(k-1), u(k)] \\ &= \frac{3u(k)y_p(k-1)y_p(k) + 0.5e^{u(k)}}{1 + y_p^2(k-1) + y_p^2(k)}. \end{aligned} \quad (28)$$

Moreover, according to the study (26), the input signal $u(k)$ was a uniformly distributed random variable over $[-1, 1]$ for the identification process. After training, the network's performance was tested using the following input signal :

$$u(k) = 0.5\sin(2\pi k/10) + 0.5\sin(2\pi k/25). \cdots \quad (29)$$

The identification of this plant was carried out using a LLAWNN with the structure $\{3-7-1\}$. Therefore the total number of the adjustable parameters in the LLAWNN is 70. The number of the training data is 500 and the testing data is 100 points. With the following algorithm parameters : L(population size) $= 20$, q(tournament size) $= 10$, η(learning rate)$= 0.05$ and μ(momentum term) $= 0.75$, for 10 different experiments, after 200 generations of EP algorithm and averaged 2081 iterations of gradient descent algorithm, the MSE of the training data was less than 2×10^{-4}. The output of both the network and the actual model with the testing input $u(k)$ are shown in Fig.7 where the MSE of the testing data is 4.05×10^{-4}. This figure indicated that the LLAWNN could successfully identify the complex characteristics of the dynamic nonlinear plant. Here the identification performed with the traditional multilayer neural network in (26) is given as a comparison. The traditional multilayer neural network structure is $\{3-17-1\}$. There are a total of 86 weight parameters and bias parameters to be adjusted. Training was performed with the standard BP algorithm, a modified fast algorithm and a self-learning algorithm employed Kalman filtering. After 4105 iterations and 13467 iterations, both the self-learning algorithm and the modified fast algorithm has a MSE of less than 2×10^{-4}. But with the standard BP algorithm the MSE of the training data was around 0.008 even after 20000 iterations. It is clear that the LLAWNN performed better than the traditional multilayer neural network whether on the number of adjustable parameters or on the training convergence rate.



Network and actual output Error of the network output

Fig. 7. Result of the example 2

C. Example 3

The nonlinear plant employed here as example 3 is taken from (27) in which the identification of this dynamic system with previous WNNs has been discussed. It is assumed to be of the following form :

$$y_p(k+1) = f[y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)] \quad (30)$$

where the unknown function f has the form

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1x_2x_3x_5(x_3-1) + x_4}{1 + x_2^2 + x_3^2}. \quad (31)$$

The input signal $u(k)$ for the identification process is generated by:

$$u(k) = \begin{cases} \sin(\frac{2\pi k}{250}) & k < 500 \\ 0.8\sin(\frac{2\pi k}{250}) + 0.2\sin(\frac{2\pi k}{25}) & k \geq 500 \end{cases} \quad (32)$$

and after training, the network's performance was tested using the following input signal :

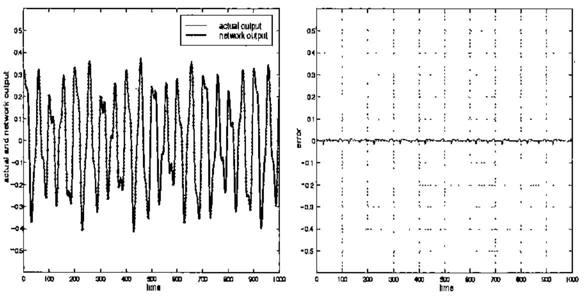
$$u(k) = 0.3\sin(\frac{k\pi}{25}) + 0.1\sin(\frac{k\pi}{32}) + 0.1\sin(\frac{k\pi}{10}). \quad (33)$$

Here, 1000 points of training data and testing data are generated, respectively.

Identification results of the proposed LLAWNN and previous WNNs are shown in Tbl.2. MSE is calculated with the testing input signal $u(k)$. The training details of the LLAWNN are the same as the above two experiments. The two previous WNN in Tbl.2 are adaptive wavelet neural networks with straightforward weights and the probabilistic incremental program evolutionary algorithm. The only difference of the two previous AWNN is that different wavelets were used as activation functions. Similar to the above experiments, it can be seen that the LLAWNN could learn the input/output relationship well for an unknown dynamic nonlinear system with a smaller number of basis functions compared with previous WNNs. Fig.8 showed the output of both the LLAWNN and the actual system with the testing data.

Table 2. Identification results of example 3

	network structure	number of parameters	MSE
LLAWNN	5-8-1	128	0.000041
AWNN1	5-19-1	209	0.000121
AWNN2	5-15-1	165	0.000372



Network and actual output   Error of the network output

Fig. 8. Result of the example 3

5. Conclusion

In this paper, we proposed a new type of wavelet-based neural network: the local linear adaptive wavelet neural network with hybrid evolutionary programming and gradient descent algorithm. The basic idea is to replace the straightforward weights of previous wavelet neural networks by introducing the local linear models as powerful weights. This was inspired by the studies of some neuro-fuzzy systems. 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. Training of the introduced network is performed by a hybrid EP/Gradient algorithm. Evolutionary programming algorithm is used first to locate a good region in the parameter space and then gradient descent algorithm, the local search procedure, is employed to find a near optimal solution in that region. Several tests on the nonlinear dynamic system identification problem indicated that the local linear adaptive wavelet neural network with the hybrid EP/Gradient training algorithm could successfully identify and describe the input/output relationship for an unknown complex system with a small number of wavelet basis functions and compared favorably to the traditional neural networks with sigmoid activation functions and the previous wavelet neural networks with straightforward weights.

For future work, it would be interested in the optimization of the network architecture and other practical applications of the network.

(Manuscript received July, 25, 2001)

References

(1) G. Gybenko, "Approximation by superposition of a sigmoidal function", Mathematics of Control, Signals and Systems, Vol.2, pp.303-314, 1989

(2) K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators", Neural Networks, vol.2, pp.359-366, 1989

(3) S. Chen and S.A. Billings, "Nonlinear system identification using neural networks", Int. Journal of Control, Vol.51, No.6, pp.1191-1214, 1990

(4) M.J. Willis et al., "Artificial neural network in process estimation and control", Automatica, Vol.28, No.6, pp.1181-1187, 1992

(5) Q. Zhang and A. Benveniste, "Wavelet networks", IEEE Trans. on Neural Networks, Vol.3, No.6, pp.889-898, 1992

(6) Y.C. Pati and P.S. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations", IEEE.Trans. on Neural Networks, Vol.4, No.1, pp.77-85, 1993

(7) J. Zhang, G.G. Walter, Y.B. Miao and W.N.W. Lee, "Wavelet neural networks for function learning", IEEE Trans. on Signal Processing, Vol.4, No.6, pp.1485-1497, 1995

(8) T. Yamakawa, E. Uchino and T. Samatsu, "Wavelet neural networks employing over-complete number of compactly supported non-orthogonal wavelets and their applications", Proc. the 1994 IEEE Int. Conf. Neural Networks, Orland, Florida, pp.1391-1396, 1994

(9) S. Kadambe and P. Srinivasan, "Applications of adaptive wavelets for speech", Optical Engineering, Vol.33, No.7, pp.2204-2211, 1994

(10) M. Fischer, O. Nelles and R. Isermann, "Adaptive predictive control of a heat exchanger based on a fuzzy model", Control Engineering Practice, Vol.6, pp.259-269, 1998

(11) T. Wang and Y. Sugai, "A local linear adaptive wavelet neural network", the Trans. of the Institute of Electrical Engineers of Japan, Vol.122-C, No.2, pp.277-284, 2002

(12) Charles k. Chui, An introduction to wavelets, Academic Press, 1992

(13) T. Poggio, F. Girosi, "Networks for approximation and learning", Proceedings on the IEEE, Vol.78, No.9, 1990

(14) R. Murray-Smith, "A local model network approach to nonlinear modeling", Ph.D. Thesis, University of Strathclyde, UK., 1994

(15) B. Foss, T.A. Johansen, "On local and fuzzy modeling", Proc. 3rd Int. Industrial Fuzzy Control and Intelligent Systems, Houston, Texas, 1993

(16) T.A. Johansen and B.A. Foss, "Nonlinear local model representation for adaptive systems", Proceeding of the Int. Conf. on Intelligent Control and Instrumentation, Vol.22, pp.677-682, 1992

(17) B. Yoon, D.J. Holmes and G. Langholz, "Efficient genetic algorithm for training layered feedforward neural networks", Information Sciences, Vol.76, No.1-2, pp.67-85, 1994

(18) X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks", IEEE Trans. on Neural Networks, Vol.8, No.3, pp.694-713, 1997

(19) G.W. Greenwood, "Training partially recurrent neural networks using evolutionary strategies", IEEE Trans. on Speech and Audio Processing, Vol.5, No.2, pp.192-194, 1997

(20) X. Yao, "Evolving artificial neural networks", Proceedings of the IEEE, Vol.87, No.9, pp.1423-1447, 1999

(21) V.W. Porto, D.B. Fogel and L.J. Fogel, "Alterative neu-

ral network training methods", IEEE Expert, Vol.10, No.3, pp.16-22, 1995
(22) A. Likartsis, I. Vlachavas and L.H. Tsoukalas, "New hybrid neural-genetic methodology for improving learning", Proc. of the 9*th* IEEE International Conference on Tools with Artificial Intelligence, pp.32-36, 1997
(23) S.W. Lee, "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.18, No.6, pp.648-652, 1996
(24) K.S.Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", IEEE Tran. on Neural Networks, Vol.1, No.1, pp.4-27, 1990
(25) K.S. Narendra, "Adaptive control using neural networks and approximate models", IEEE Tran. on Neural Networks, Vol.8, No.3, pp.475-485, 1997
(26) K.N. Lou, "A new system identification technique using Kalman filtering and multilayer neural networks", Artificial Intelligence in Engineering, Vol.10, pp.1-8, 1996
(27) Y.H. Chen and S. Kawaji, "Evolving Wavelet Neural Networks for System Identification", Proc. of the 18*th* Annual Conference of the Robotics and Machtronics of Japan, 1A1-29-039, pp.1-6, 2000

Ting Wang (Student Member) received the M.S. degree in electronic engineering from the Dalian University of Technology, P.R.China, in 1996. Presently, she is a doctoral student of the Graduate School of Science and Technology, Chiba University. Her research interests include neural networks and applications of wavelet transform theory.

Yasuo Sugai (Member) received the D.Eng. degree from the Tokyo Institute of Technology in 1985. He is an associate professor, department of urban environmental systems, faculty of engineering, Chiba university. His research areas are the neural networks, the emergent computation, the optimization engineering, and their applications. He is a member of IEICE, IPSJ and SICE.