2005년도 대한전기학회 하계학술대회 논문집 2005. 7. 18 - 20

비선형 시스템의 동정을 위한 안정한 웨이블릿 기반 퍼지 뉴럴 네트워크

°오준섭*, 박진배*, 최윤호**
*연세대학교 전기전자공학과, **경기대학교 전자공학부

Stable Wavelet Based Fuzzy Neural Network for the Identification of Nonlinear Systems

°Joon Seop Oh*, Jin Bae Park*, Yoon Ho Choi**
*Dept. of Electrical & Electronic Eng., Yonsei Univ., **School of Electronic Eng., Kyonggi Univ.

Abstract - In this paper, we present the structure of fuzzy neural network(FNN) based on wavelet function, and apply this network structure to the identification of nonlinear systems. For adjusting the shape of membership function and the connection weights, the parameter learning method based on the gradient descent scheme is adopted. And an approach that uses adaptive learning rates is driven via a Lyapunov stability analysis to guarantee the fast convergence. Finally, to verify the efficiency of our network structure, we compare the identification performance of proposed wavelet based fuzzy neural network(WFNN) with those of the FNN, the wavelet fuzzy model(WFM) and the wavelet neural network(WNN) through the computer simulation.

1. Introduction

When researchers want to find the model of a system mathematically, the differential equation has been widely used. However, there is so much nonlinearity and a number of time constrains in realistic systems that the accurate differential equation can hardly be obtained. Though the comparatively precise model is acquired, the model efficiency is decreased by model approximation. In order to solve this problem, intelligent techniques, based on neural networks and fuzzy logic, have also been developed for system identification[1]-[3]. Even though these intelligent modeling strategies have shown their effectiveness, especially for nonlinear systems, they have certain drawbacks derived from their own characteristics. Therefore, for the identification of nonlinear system, we designed a fuzzy neural network(FNN) structure based on wavelet that merges these advantages of neural network, fuzzy model and wavelet transform[4]. The basic idea of wavelet based fuzzy neural network(WFNN) is to realize the process of fuzzy reasoning of WFM by the structure of a neural network and to make the parameters of fuzzy reasoning be expressed by the connection weights of a neural network. For adjusting the shape of membership function and the connection weights, the parameter learning method based on the gradient descent(GD) scheme is adopted. And an approach that uses adaptive learning rates is driven via a Lyapunov stability analysis to guarantee the fast convergence. Finally, to verify the efficiency of our network structure, we compare the identification performance of proposed WFNN with those of the FNN, the WFM and the wavelet neural network(WNN) through the computer simulation.

2. WaveletBasedFuzzyNeuralNetwork

In our WFNN structure[4], the network output $\hat{y}_c$ is calculated as follows:

$$\hat{y}_c = \sum_{n=1}^{N} a_{nc} x_n + \sum_{j=1}^{R} B_{jc} \Phi_j \tag{1}$$

In our network structure, the network weight set, $\boldsymbol{\gamma} = \{\mathbf{a}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{m}\}$, is tuned to minimize the identification error via the GD method. In order to apply the GD method, the squared error function is defined as follows:

$$J = \frac{1}{2}((y_{r1} - \hat{y}_1)^2 + (y_{r2} - \hat{y}_2)^2 + \cdots + (y_{rC} - \hat{y}_C)^2, \tag{2}$$

where, $\hat{\mathbf{Y}} = [\hat{y}_1\, \hat{y}_2 \cdots \hat{y}_C]$ are the output values of WFNN and $\mathbf{Y}_r = [y_{r1}\, y_{r2} \cdots y_{rC}]$ are the desired values. Using the GD method, the weight set, $\boldsymbol{\gamma} = \{\mathbf{a}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{m}\}$, can be tuned as follows:

$$\begin{aligned} \gamma_p(k+1) &= \gamma_p(k) + \Delta\gamma_p(k) = \gamma_p(k) - \eta \frac{\partial J}{\partial \gamma_p(k)} \\ &= \gamma_p(k) - \eta \frac{\partial J}{\partial \hat{\mathbf{Y}}} \frac{\partial \hat{\mathbf{Y}}}{\partial \gamma_p(k)} = \gamma_p(k) + \eta \cdot \mathbf{E} \cdot \hat{\upsilon}_p, \end{aligned} \tag{3}$$

where, $\mathbf{E} = [(y_{r1} - \hat{y}_1)(y_{r2} - \hat{y}_2) \cdots (y_{rC} - \hat{y}_C)]$ and subscript p denotes each network weight. And η is called the learning rate. The gradient set of WFNN output $\hat{\mathbf{Y}}$ with respect to weight set is calculated as in Eq. (4) and each gradient of WFNN output $\hat{y}$ with respect to each weight is presented as in Eq. (5) to Eq. (7):

$$\hat{\upsilon}_p = \frac{\partial \hat{\mathbf{Y}}}{\partial \gamma_p(k)} = [\hat{\upsilon}_a\ \hat{\upsilon}_\omega\ \hat{\upsilon}_m\ \hat{\upsilon}_d] = \left[\frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{a}(k)}\ \frac{\partial \hat{\mathbf{Y}}}{\partial \boldsymbol{\omega}(k)}\ \frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{m}(k)}\ \frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{d}(k)}\right], \tag{4}$$

$$\hat{\upsilon}_{a_{nc}} = \frac{\partial \hat{y}_c}{\partial a_{nc}(k)} = x_n, \tag{5}$$

$$\hat{\upsilon}_{\omega_{jc}} = \frac{\partial \hat{y}_c}{\partial \omega_{jc}(k)} = \frac{\partial \sum_{j=1}^{R} y_{jc}}{\partial \omega_{jc}(k)} = \frac{\Phi_j}{\sum_{j=1}^{R} I_{D_j}}, \tag{6}$$

$$\begin{aligned} \hat{\upsilon}_{m_{k_n n}, d_{k_n n}} &= \frac{\partial \hat{y}_c}{\partial m_{k_n n}, d_{k_n n}(k)} = \frac{\partial \left(\sum_{j=1}^{H} B_{jc} \Phi_j \right)}{\partial m_{k_n n}, d_{k_n n}(k)} \\ &= \sum_{h=1}^{H} \left(\omega_{jc} \left(\frac{\dot{NUM}(m_{k_n n}, d_{k_n n})}{DEN} - \frac{\dot{DEN}(m_{k_n n}, d_{k_n n}) NUM}{DEN^2} \right) \right)_h, \end{aligned} \tag{7}$$

where, $H = \frac{\prod_{k=1}^{N} K_k}{K_N}$, $NUM = \Phi_j$, $DEN = \sum_{j=1}^{R} I_{D_j}$,

$$NUM(m_{k_n n}) = \frac{\partial z_{k_n n}}{\partial m_{k_n n}} \frac{\partial NUM}{\partial z_{k_n n}} = -\frac{1}{d_{k_n n}} \left(\frac{\prod_{n=1}^{N} \phi_{k_n n}(z_{k_n n})}{\phi_{k_n n}(z_{k_n n})} \left(\left(O_{A_{k_n n}}^2 - 1\right) \exp\left(-\frac{1}{2} O_{A_{k_n n}}^2\right)\right)\right),$$

$$\dot{DEN}(m_{k_n n}) = \frac{\partial z_{k_n n}}{\partial m_{k_n n}} \frac{\partial DEN}{\partial z_{k_n n}} = -\frac{1}{d_{k_n n}} \sum_{h=1}^{H} \left(\frac{\prod_{n=1}^{N} O_{C_{k_n n}}}{O_{C_{k_n n}}} \left(-O_{A_{k_n n}} \exp\left(-\frac{1}{2} O_{A_{k_n n}}^2\right)\right)\right)_h,$$

$$\dot{NUM}(d_{k_n n}) = \frac{\partial z_{k_n n}}{\partial d_{k_n n}} \frac{\partial NUM}{\partial z_{k_n n}} = -\frac{O_{A_{k_n n}}^2}{d_{k_n n}} \left(\frac{\prod_{n=1}^{N} \phi_{k_n n}(z_{k_n n})}{\phi_{k_n n}(z_{k_n n})} \left(\left(O_{A_{k_n n}}^2 - 1\right) \exp\left(-\frac{1}{2} O_{A_{k_n n}}^2\right)\right)\right),$$

$$\dot{DEN}(d_{k_n n}) = \frac{\partial z_{k_n n}}{\partial d_{k_n n}} \frac{\partial DEN}{\partial z_{k_n n}} = -\frac{O_{A_{k_n n}}^2}{d_{k_n n}} \sum_{h=1}^{H} \left(\frac{\prod_{n=1}^{N} O_{C_{k_n n}}}{O_{C_{k_n n}}} \left(-O_{A_{k_n n}} \exp\left(-\frac{1}{2} O_{A_{k_n n}}^2\right)\right)\right)_h.$$

3. Stability Analysis

In the update rule of Eq. (3), selection of the values for the learning rate η has the significant effect on the network performance. Generally, if η is too big, the model is unstable. And for the small η, although the convergence is guaranteed, the speed is very slow. Therefore, In order to train the WFNN effectively, adaptive learning rates, which guarantee the fast convergence and stability, must be derived. In this section, the specific learning rates for the type of network weights are derived based on the convergence analysis of a discrete type Lyapunov function.

Theorem 1: Let $\eta_{p,c}$ be the learning rate for the output $\hat{y}_c$ influenced by weight vector $\boldsymbol{\gamma}_p$ of the WFNN. $\mathbf{G}_{p,c}(k)$ and $\mathbf{G}_{p,c,\max}(k)$ are defined as $\mathbf{G}_{p,c}(k) = \frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}$ and $\mathbf{G}_{p,c,\max}(k) \equiv \max_k \|\mathbf{G}_{p,c}(k)\|$, respectively, and $\|\cdot\|$ is the Euclidean norm in $\Re^n$. Here, subscript p and c denote each weight and output, respectively. Then the convergence is guaranteed if $\eta_{p,c}$ is chosen as follows:

$$0 < \eta_{p,c} < \frac{2}{\mathbf{G}_{p,c,\max}^2(k)}. \tag{8}$$

Proof:

In this analysis, a discrete type Lyapunov function is selected as

$$V(k) = \frac{1}{2} e_c^2(k), \tag{9}$$

where, $e_c(k)$ is the difference between the desired value $y_{rc}(k)$ and the output value $\hat{y}_c(k)$. Then, the change of Lyapunov function is obtained by

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}\left[e_c^2(k+1) - e_c^2(k)\right], \tag{10}$$

where, $e_c(k+1) = e_c(k) + \Delta e_c(k) \approx e_c(k) + \left[\frac{\partial e_c(k)}{\partial \gamma_p(k)}\right] \Delta \gamma_p(k)$.

In the update rule of Eq. (3), $\Delta\gamma_p(k)$ is defined as

$$\Delta \gamma_p(k) = -\eta_{p,c} \frac{\partial J}{\partial \gamma_p(k)} = \eta_{p,c} \left[\frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}\right]^T e_c(k), \tag{11}$$

and the error difference can be represented by

$$\Delta e_c(k) \approx \left[\frac{\partial e_c(k)}{\partial \gamma_p(k)}\right] \Delta \gamma_p(k) = \left[\frac{\partial e_c(k)}{\partial \gamma_p(k)}\right] \eta_{p,c} \left[\frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}\right]^T e_c(k)$$
$$= -\eta_{p,c} \left\|\frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}\right\|^2 e_c(k). \tag{12}$$

From Eqs. (10) - (12), $\Delta V(k)$ can be represented as

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}\left[(e_c(k) + \Delta e_c(k))^2 - e_c^2(k)\right]$$
$$= \Delta e_c(k)\left[e_c(k) + \frac{1}{2}\Delta e_c(k)\right]$$
$$= -\eta_{p,c} \left\|\frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}\right\|^2 e_c(k) \left[e_c(k) - \frac{1}{2}\eta_{p,c} \left\|\frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}\right\|^2 e_c(k)\right]$$
$$= -e_c^2(k)\left[\eta_{p,c} \left\|\frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}\right\|^2 - \frac{1}{2}\eta_{p,c}^2 \left\|\frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}\right\|^4\right] \tag{13}$$
$$= -e_c^2(k)\rho(k).$$

Let us define $\mathbf{G}_{p,c}(k)$ and $\mathbf{G}_{p,c,\max}(k)$ as $\mathbf{G}_{p,c}(k) = \frac{\partial \hat{y}_c(k)}{\partial \gamma_p(k)}$ and $\mathbf{G}_{p,c,\max}(k) \equiv \max_k \|\mathbf{G}_{p,c}(k)\|$, respectively. Since

$$\rho(k) = \eta_{p,c} \|\mathbf{G}_{p,c}(k)\|^2 \left[1 - \frac{1}{2}\eta_{p,c} \|\mathbf{G}_{p,c}(k)\|^2\right]$$
$$= \eta_{p,c} \|\mathbf{G}_{p,c}(k)\|^2 \left[1 - \frac{1}{2} \frac{\eta_{p,c} \mathbf{G}_{p,c,\max}^2(k) \|\mathbf{G}_{p,c}(k)\|^2}{\mathbf{G}_{p,c,\max}^2(k)}\right]$$
$$\geq \eta_{p,c} \|\mathbf{G}_{p,c}(k)\|^2 \left[1 - \frac{1}{2}\eta_{p,c} \mathbf{G}_{p,c,\max}^2(k)\right] > 0, \tag{14}$$

we obtain

$$0 < \eta_{p,c} < \frac{2}{\mathbf{G}_{p,c,\max}^2(k)}. \tag{15}$$

■ Q.E.D

Remark 1: The convergence is guaranteed as long as Eq. (14) is satisfied, i.e.:

$$\eta_{p,c}\left[1 - \frac{1}{2}\eta_{p,c} \mathbf{G}_{p,c,\max}^2(k)\right] > 0. \tag{16}$$

The maximum learning rate, which guarantees the fast convergence, can be obtained as $\eta_{p,c} \mathbf{G}_{p,c,\max}^2(k) = 1$, i.e.:

$$\eta_{p,c,\max} = \frac{1}{\mathbf{G}_{p,c,\max}^2(k)}, \tag{17}$$

which is the half of the upper limit.

Theorem 2: Let $\eta_{p,c} = \{\eta_{a,c}, \eta_{\omega,c}, \eta_{m,c}, \eta_{d,c}\}$ be the learning rate set for the weight set, $\gamma = \{\mathbf{a}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{m}\}$, of WFNN and $\mathbf{G}_{p,c}(k)$ is defined as the gradient set, $\left\{\frac{\partial \hat{y}_c(k)}{\partial \mathbf{a}(k)}, \frac{\partial \hat{y}_c(k)}{\partial \boldsymbol{\omega}(k)}, \frac{\partial \hat{y}_c(k)}{\partial \mathbf{m}(k)}, \frac{\partial \hat{y}_c(k)}{\partial \mathbf{d}(k)}\right\}$, of WFNN output $\hat{y}_c$ with respect to weight set. Then the convergence is guaranteed if $\eta_{p,c}$ is chosen as

$$(a)\quad 0 < \eta_{a,c} < \frac{2}{N|x_n|_{\max}^2}, \quad (b)\quad 0 < \eta_{\omega,c} < \frac{2}{R^2 |O_{B_j}|_{\max}^2},$$

$$(c)\quad 0 < \eta_{m,c} < \frac{2}{\sqrt{C}\left| H|\omega_{jc}|_{\max}^2 \left(\frac{|DEN| + \sqrt{H}}{|d_{k_n n}|_{\min} |DEN|^2}\right)^2 \right|_{\max}},$$

$$(d)\quad 0 < \eta_{d,c} < \frac{2}{\sqrt{C}\left| H|\omega_{jc}|_{\max}^2 \left(\frac{|O_{A,k_n n}^2|_{\max} \left(|DEN| + \sqrt{H}\right)}{|d_{k_n n}|_{\min} |DEN|^2}\right)^2 \right|_{\max}}. \tag{18}$$

Proof : (a)

Let us define $\mathbf{G}_{a,c,\max}(k)$ as $\mathbf{G}_{a,c,\max}(k) \equiv \max_k \|\mathbf{G}_{a,c}(k)\|$. Then from Eq. (15), we obtain $0 < \eta_{a,c} < \frac{2}{\mathbf{G}^2_{a,c,\max}(k)}$. And from the definition of Theorem 1, the maximum condition can be obtained as

$$\max_k \|\mathbf{G}_{a,c}(k)\| = \max_k \left\| \frac{\partial \hat{y}_c(k)}{\partial a_{nc}(k)} \right\| = \max_k \|\mathbf{x}\| \le \sqrt{N} |x_n|_{\max}$$

Thus $\mathbf{G}^2_{a,c,\max}(k) = N |x_n|^2_{\max}$, where x_n is the n- th input value of FWNN and N is the number of input. ■ Q.E.D

4. Simulation Results

In this simulation, we consider the Duffing system that is the representative chaotic system. The state equation of Duffing system is as follows:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ a_1 x(t) - x^3(t) + a_2 y(t) + b\cos(\omega t) \end{bmatrix}, \qquad (19)$$

where typically $a_1 = 1.1$, $a_2 = 0.4$, $b = 2.1$, $\omega = 1.8$. In this simulation, the inputs of the identification model are the present and previous values of $x(t)$. Here, the initial state is $[x(0), y(0)]^T = [1\ 0]^T$. And the initial values of network weight are randomly determined and sampling time is 0.05sec. The simulation environments and identification results are as shown in Table 1. Figure 1 shows the desired output, the WFNN model output and the error between these outputs, respectively. And Fig. 2 represents the adaptive learning rates for the fast convergence and stability.

Table 1. The simulation environment and results

	MF of each input	Wavelet (Rule number)	Parameter	Learning rate	MSE
Our WFNN	3	9	33	**Adaptively** (initial value : 0.1)	**0.0002329**
Our WFNN	3	9	33	Experimentally fixed : 0.09	0.0007112
WFM	9	9	45	Experimentally fixed : 0.015	0.002931
FNN	5	25	47	Experimentally fixed : 0.15	0.2685
WNN	*	10	52	Experimentally fixed : 0.02	0.001763

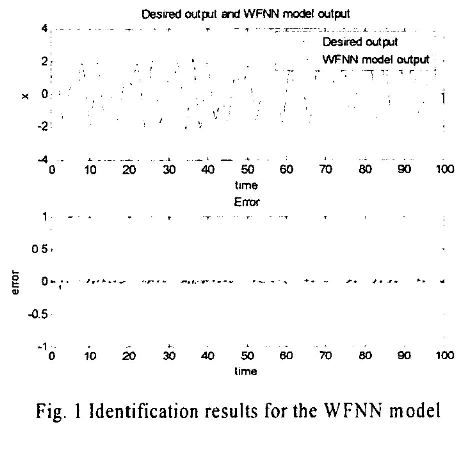

Fig. 1 Identification results for the WFNN model

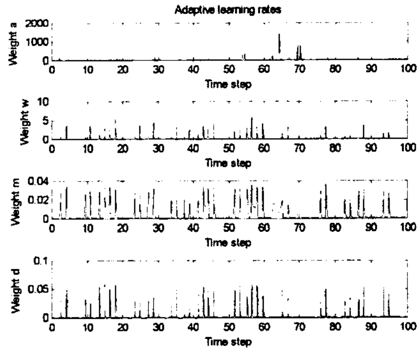

Fig. 2 Adaptive learning rates for the WFNN weights

As a result, if the identification error is increased then the learning rates are increased too for the fast convergence, and if it is decreased then the learning rates are decreased for the accurate identification performance. In our experiments, we use the mean squared error(MSE) as the identification error for comparison of performance. From Fig. 1 and Table 1, we confirm that the identification performance of our WFNN model is better than those of other network models.

5. Conclusions

In this paper, we have proposed a FNN structure based on wavelet transform, which merges the advantages of neural network, fuzzy model and wavelet. And an approach that uses adaptive learning rates was driven via a Lyapunov stability analysis to guarantee the fast convergence. As a result, if the identification error was increased then the learning rates were also increased for the fast convergence, and if it was decreased then the learning rates were decreased for the accurate identification. Therefore, the learning rates were adaptively determined to rapidly minimize error between our WFNN output and the desired output. As a final result, we have confirmed that the identification performance of our WFNN model is better than those of other network models.

References

[1] R. A. Felix and E. N. Sanchez, "Chaos identification using variable structure recurrent neural networks, " in Proc. of the 9th Int. Conf. on Neural Information, vol. 1, 2002, pp. 262-266.

[2] T. Hasegawa, S. Horikawa, T. Furuhashi and Y. Uchikawa, "On design of adaptive fuzzy neural networks and description of its dynamical behavior," Fuzzy Sets and Systems, vol. 71, no. 1, pp. 3-23, 1995.

[3] C. K. Lin and S. D. Wang, "Fuzzy modeling using wavelet transforms," Electronics Letters, vol. 32, Issue 24, 1996, pp. 2255-2256.

[4] J. S. Oh and Y. H. Choi, "Path Tracking Control Using a Wavelet Based Fuzzy Neural Network for Mobile Robots", International Journal of Fuzzy Logic and Intelligent Systems, Vol. 4, No. 1, pp. 111-118, 2004/06