第 29 卷 第 6 期 自 动 化 学 报 Vol. 29, No. 6
2003 年 11 月 ACTA AUTOMATICA SINICA Nov., 2003

Generalized Fuzzy Neural Network[1)]

YU You-Ling XU Li-Hong WU Qi-Di
(*Department of Control Theory and Control Engineering, Tongji University, Shanghai* 200092)
(E-mail: yuyouling@yahoo.com)

Abstract Based on the intrinsic physical background of nonlinear system, a system identification model is derived from the inherent systematic characteristics, a priori knowledge and experiences. And then, the GFNN (generalized fuzzy neural network) is put forwand, the GFNN approximation theorem is proved. The structure-self-organization and parameter-self-learning algorithm is proposed, which can automatically and simultaneously deal with the process of the system structure identification and parameter self-learning under predefined precision, so that the novel on-line structure self-organization of GFNN is realized. Simulation shows the nonlinear approximation abilities of GFNN, especially for identification of slow time-varying plant. The GFNN is a successful integrated algorithm of fuzzy logic and neural network.

Key words Fuzzy neural network, identification, approximation theorem, time-varying nonlinear, structure-self-organization, parameter-self-learning

1 Introduction

Artificial neural network and fuzzy logic system are two successful methods for the modeling and identification of nonlinear system. Up to now, many exciting improvements have been made and some novel solutions have been provided for nonlinear problem resolving. But for problems about the intrinsic nonlinear characteristics of the plant, such as network structure, information interface, parameter initials, convergent speed of the optimization algorithms, *etc.*, no satisfying solutions can be found.

Fuzzy neural network is the integration of fuzzy logic system and artificial neural network. But to be an effective fuzzy neural network, it must at least have the following features: reasonable network knowledge structure, extensible knowledge base, intelligent and wide information interface, and so on.

2 GFNN neuron & GFNN network structure

2.1 GFNN neuron

The fundamental component of GFNN is the neuron which shows a relationship between the input and output and implicates an inference mechanism. So GFNN neuron should possess the generalized network structure, the information storage and processing abilities. Specifically, the neuron can process different input information, such as linguistic information and data information, Gaussian random information, etc. Its inference mechanism results in a wide selection range of activation functions, so many membership functions can be adopted as the activation function of the neuron. In its relatively simple network structure, GFNN neurons in different layers take different roles and show different characteristics. A typical GFNN neuron is shown in Fig. 1. Its input-output function is

$$w = f(g-\theta) = \begin{cases} g, & g \geqslant \theta \\ 0, & g < \theta \end{cases} \tag{1}$$

where $g=g(\boldsymbol{X},\boldsymbol{\Phi})$ is the inference function, $\boldsymbol{\Phi}$ is the stored information parametric vec-

1) Supported by National "973" Key Project of P. R. China(2002CB312200)
Received January 05, 2001; in revised form December 23, 2002
收稿日期 2001-01-05; 收修改稿件日期 2002-11-23

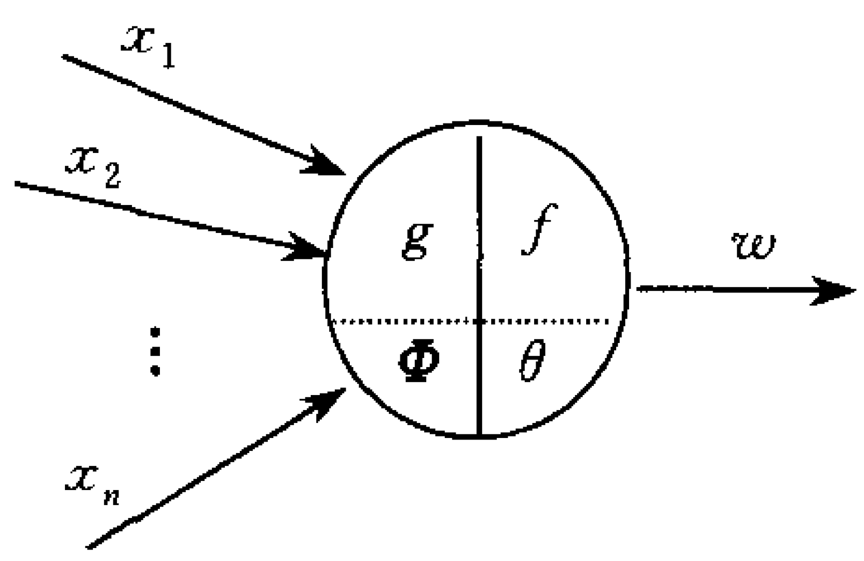

Fig. 1 GFNN neuron

tor, $\boldsymbol{X}=(x_1, x_2, \cdots, x_n)$ is the input vector, θ is the threshold and $f(\cdot)$ is the activation function. Obviously, the neuron produces its output based on its intrinsic inference mechanism, such as Gaussian inference function.

2.2 GFNN network structure

2.2.1 Network structure

GFNN is a 3-layer feed-forward network that consists of three layer neurons (Fig. 2, MISO case), i.e., the input layer, hidden layer and output layer. These layers are corresponding to the three phases of FLS (Fuzzy Logic System) respectively and the key parts of FLS, the fuzzy inference and the fuzzy rule-base, are integrated into the hidden layer. The inference mechanism behaves as the inference function of the hidden neurons. So the hidden neurons are corresponding to the fuzzy rules and the number of hidden neurons is equal to that of fuzzy rules.

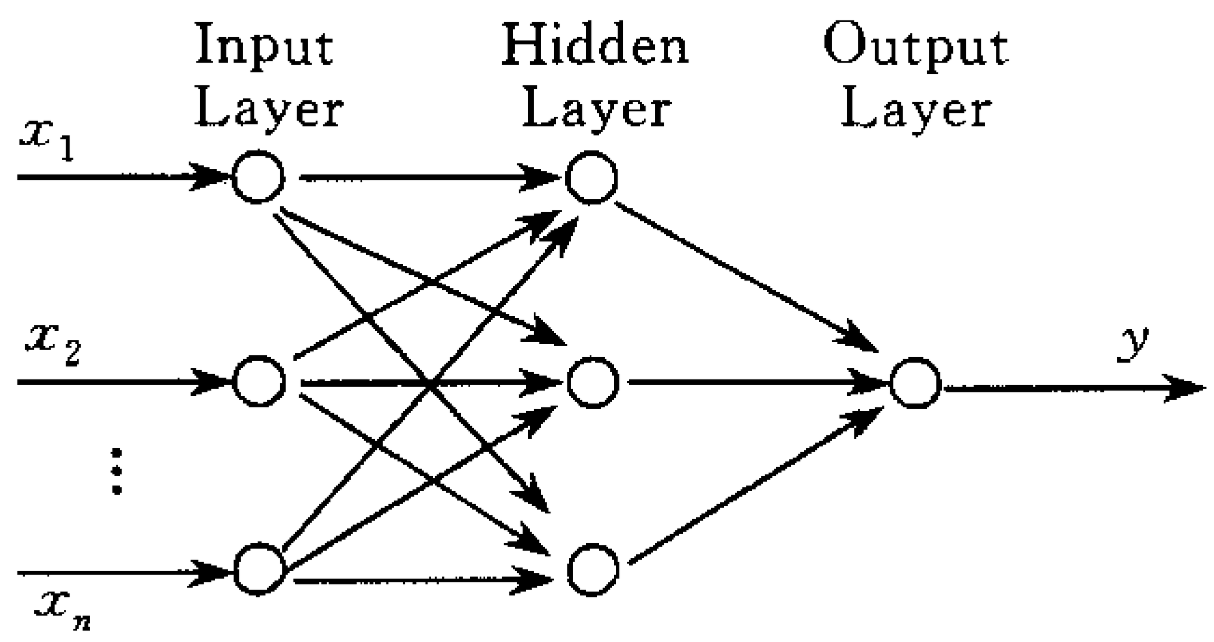

Fig. 2 GFNN 3-layer network structure

1) Information match degree (IMD)

Definition 1. For an input $\boldsymbol{X}$, the Information Match Degree (IMD) corresponding to the jth rule is

$$\lambda_j(X) = g_j(X, \Phi_j) \tag{2}$$

The physical meaning of $\lambda_j(\boldsymbol{X})$ is very clear and it shows the match level between the input and the jth rule. For the activation of the input, it shows the inner response level of the jth hidden neuron.

2) Information believable level (IBL)

Definition 2. For an input X, only the hidden neuron whose output is greater than a certain level can influence the neurons in the output layer. This level is defined as Information Believable Level (IBL), represented by θ_j. Actually, the IBL is equivalent to the threshold of the traditional neuron. So the output of the jth neuron is

$$z_j = f(g_j(\boldsymbol{X}, \Phi_j) - \theta_j) = \begin{cases} g_j(\boldsymbol{X}, \Phi_j), & g_j \geq \theta_j \\ 0, & g_j < \theta_j \end{cases} \tag{3}$$

2.2.2 Initialization of network parameters

Based on the above definitions and the physical background of FLS, the initialization process of the neuron in each layer is shown as below.

1) Input layer

For the singleton fuzzy generator, set $\boldsymbol{\Phi}=0, \theta=0, g(\boldsymbol{X})=\boldsymbol{X}$.

2) Hidden layer

Corresponding to a fuzzy rule, each hidden neuron has two main functions: information storage and fuzzy inference.

a) Information storage

The adjustable vector Φ_j of the jth neuron is corresponding to the parameter vector of the jth rule.

b) Fuzzy inference

The inference function of the jth neuron is corresponding to the fuzzy inference principle and acts as the fuzzy membership function. For product inference principle, we can get $g_j(\boldsymbol{X},\Phi_j)=\prod_{i=1}^{n}u_{ij}(x_i)$, where $u_{ij}(x_i)$ is the input membership function for the input x_i, Φ_j can be initialized in the same way. For simplicity, we can set $\theta_j=\theta^*$ (θ_j can be set by self-learning or commonly predefined as $0\leqslant\theta^*<1$). So the output of the hidden neuron is

$$z_j = f(g_j(\boldsymbol{X},\Phi_j)-\theta^*) = \begin{cases} g_j(\boldsymbol{X},\Phi_j), & g_j \geqslant \theta^* \\ 0, & g_j < \theta^* \end{cases} \tag{4}$$

3) Output layer

This layer is corresponding to the fuzzy defuzzier phase. Choosing the barycenter defuzzier methods with $\theta=0$, the process is shown as below. There exist two cases.

a) If there exists $z_j>0$, $j=1,\cdots,M(t)$ in the hidden layer, that is, there exist $M(t)$ valid outputs and $M(t)$ hidden neurons which satisfy $g_j(\boldsymbol{X},\Phi_j)>\theta^*$, then the output of the output layer is $y=g(\boldsymbol{Z},\boldsymbol{\Phi}_z)=\sum_{j=1}^{M(t)}v_jz_j'$, where $z_j'=z_j/\sum_{k=1}^{M(t)}z_k$, $\boldsymbol{Z}=(z_1,z_2,\cdots,z_{M(t)})$ is the valid output vector of the hidden layer, v_j is the center position of the jth neuron, $\boldsymbol{\Phi}_z=(v_1,v_2,\cdots,v_{M(t)})$ is the information parametric vector of the output neuron, $M(t)$ is the number of the hidden neurons which have valid output at time t.

b) If for all hidden neurons, $z_j=0$, $j=1,\cdots,N(t)$, where $N(t)$ is the total number of hidden neurons, that is to say, there are no valid outputs and the outputs of all hidden neurons are less than the IBL, then instead of the equation $z_j'=z_j/\sum_{k=1}^{M(t)}z_k$ we have $z_i'=\max_{j=1}^{N(t)}(g_j(X,\Phi_j))$. That is, we use the output of the hidden neuron which has the maximum IBL as the input for the output layer. If this neuron is the ith one, then the output of the output layer is $y=v_i$.

Comment: v_i is equivalent to the output connection weight of the traditional neural network. We think that this parameter should be integrated into the neuron based on the understanding of the bionic neuron.

3 GFNN approximation theorem

We will prove this theorem through constructed proof techniques.

3.1 GFNN approximation theorem in 1-dimension space

Theorem 1. Assume $f(x)$ is the continuous function defined in $(-\infty,\infty)$, and that there exist $\lim_{x\to-\infty}f(x)=A$ and $\lim_{x\to\infty}f(x)=B$, where A, B are constant. For any $\varepsilon>0$ and $x\in(-\infty,\infty)$, there exists a GFNN function $h(x)$ satisfying

$$|f(x)-h(x)|<\varepsilon \tag{5}$$

Proof.

From assumption, for any $\varepsilon>0$, we can find $M,N>0$ which satisfy

1) if $x>M$, there exists $|f(x)-B|<\varepsilon/3$;

2) if $x<-M$, there exists $|f(x)-A|<\varepsilon/3$;

3) if $|x'|\leqslant M$, $|x''|\leqslant M$ and $|x'-x''|<1/N$, there exists $|f(x')-f(x'')|<\varepsilon/3$.

Now we divide $[-M,M]$ into $2MN$ sections and the length of each section is $1/N$. Set

$$-M = x_0 < x_1 < \cdots < x_{MN} = 0 < x_{MN+1} < \cdots < x_{2MN} = M$$

We know that if $x' \in (x_i, x_{i+1}), x'' \in (x_i, x_{i+1}), i=0,1,\cdots,2MN-1$, there should be $|f(x')-f(x'')|<\varepsilon/3$. Let t_j be the point in section $[x_j, x_{j+1}]$ at which the maximum membership value exists. Choose a reasonable membership function $g_j(X,\Phi_j)$ and a neuron threshold θ_j in this section, which satisfy $f(x_j)=f(x_{j+1})=\theta_j$, and the output of hidden neuron in this section is

$$z_j = f(g_j(x,\Phi_j)-\theta_j) = \begin{cases} g_j(x,\Phi_j), & x \in [x_j, x_{j+1}] \\ 0, & x \notin [x_j, x_{j+1}] \end{cases}$$

Simultaneously, select the output neuron as a singleton, that is, the output set contain only one point $f(t_j)$ which satisfies $\mu_O(f(t_j))=1$. So when the membership value is 1, $v_j = f(t_j)$.

So from the definitions of GFNN, for $x \in (-\infty,\infty)$, we can prove this theorem in the three sections respectively.

1) If $x \in [-M,M]$, then setting$x \in [x_k, x_{k+1}], k=0, 1, \cdots, 2MN-1$, we have $\begin{cases} z_k = f(g_k(x,\Phi_k)-\theta_k) = g_k(x,\Phi_k) \\ z_j = 0 \quad j \neq k \end{cases}$. Since $|t_k - x| < 1/N$, $y = h(x) = v_k = f(t_k)$, and therefore there exists $|f(x)-h(x)| = |f(x)-f(t_k)| < \varepsilon/3 < \varepsilon$.

2) If $x > M$, there exists $g_{2MN-1}(x,\Phi_{2MN-1}) = \max\limits_{j=0,\cdots,2MN-1}(g_j(x,\Phi_j))$, hence $y=h(x) = v_{2MN-1} = f(t_{2MN-1})$ and $|f(t_{2MN-1})-f(M)|<\varepsilon/3$, $|f(M)-B|<\varepsilon/3$, $|f(x)-B|<\varepsilon/3$, so $|f(x)-h(x)| = |f(x)-f(t_{2MN-1})|<\varepsilon$.

3) If $x < -M$, there exists $g_0(x,\Phi_0) = \max\limits_{j=0,\cdots,2MN-1}(g_j(x,\Phi_j))$, hence$y=h(x)=v_0= f(t_0)$ and $|f(t_0)-f(-M)|<\varepsilon/3$, $|f(-M)-A|<\varepsilon/3$, $|f(x)-A|<\varepsilon/3$, so $|f(x)-h(x)| = |f(x)-f(t_0)|<\varepsilon$.

This theorem is proved. □

Corollary. Assuming $f(x)$ is a continuous function defined in a limited section $[a,b]$, for any $\varepsilon>0$ and all $x \in [a,b]$, there exists a 1-dimensional GFNN $h(x)$ which satisfies

$$|f(x) - h(x)| < \varepsilon \tag{6}$$

That is, $h(x)$ is dense in $C[a,b]$.

3.2 GFNN approximation theorem in multi-dimensional space

Theorem 2. If $g(x): R^1 \to R^1$ satisfies $g \in L^P_{loc}$ and there exists a GFNN $h(x)$ which is dense in each $L^p[a,b]$, then for $x \in K$, $h(x)$ is dense in $L^p(K)$, where K is a compact set defined in R^n.

Proof.

From reasonable shift and transform, we can limit space K to be $K \in [-1,1]^n$. For any $\varepsilon>0$, exists $\delta>0$ which satisfies

$$\| f_\delta(x) - f(x) \|_{L^p(K)} < \varepsilon/2 \tag{7}$$

Extending $f_\delta(x)$ in section$[0,1]^n$, we have

$$h_\delta(x_1,\cdots,x_n) = \begin{cases} f_\delta(x_1,\cdots,x_n), & (x_1,\cdots,x_n) \in [0,1]^n \\ f_\delta(x_1,\cdots,-x_k,\cdots,x_n), & (x_1,\cdots,x_n) \notin [0,1]^n \end{cases}.$$

For all $x \in [-1,1]^n$, we can find $R>0$ which satisfies $\left| \sum\limits_{|m| \leqslant R} \left(1 - \frac{|m|^2}{R^2}\right)^\alpha c_m(h_\delta) e^{i\pi m \cdot x} - h_\delta(x) \right| < \frac{\varepsilon}{4}$. This means

$$\left\| \sum_{|m| \leqslant R} \left(1 - \frac{|m|^2}{R^2}\right)^\alpha c_m(h_\delta) e^{i\pi m \cdot x} - h_\delta(x) \right\|_{L^p(K)} < \frac{\varepsilon}{4} \tag{8}$$

From the definition of the Fourier coefficients and the even functional characteristics of

$h_\delta(x_1,\cdots,x_n)$, then exist real numbers $d_{m_1,\cdots,m_n}$ which satisfy

$$\left\| \sum_{|m|^2 \leqslant R} d_{m_1,\cdots,m_n} \cos\pi(m_1 x_1 + \cdots + m_n x_n) - h_\delta(x) \right\|_{L^p(K)} < \frac{\varepsilon}{4} \tag{9}$$

Obviously, for any $x \in [-1,1]^n$, there exists $\cos(u) \in L^p[-\sqrt{n}\pi R, \sqrt{n}\pi R]$. From the corollary of Theorem 1 and the assumption of this theorem, we can find a GFNN $h'(u)$, i. e., $h(x)$, which satisfies $\|h'(u)-\cos(u)\|_{L^p[-\sqrt{n}\pi R,\sqrt{n}\pi R]} < \varepsilon/4L$, where $L = \sum_{|m| \leqslant R} |d_{m_1,\cdots,m_n}|$.

Thereby, $\|h(x)-\cos(\pi m \cdot x)\|_{L^p(K)} < \frac{\varepsilon}{4L}$. Substituting it into formula (9), we have

$$\|h(x) - h_\delta(x)\|_{L^p(K)} < \varepsilon/2 \tag{10}$$

From formulae (7) and (10), there exists

$$\|f(x) - h(x)\|_{L^p(K)} < \varepsilon$$

The Theorem 2 is proved (relative lemmas seen in reference 5). □

Theorem 3. If a GFNN $h(x)$ constructed in $g(x): R^1 \to R^1$ is dense in $C(\bar{R}^1)$, then for all $x \in R^n$, $h(x)$ is dense in $C(\bar{R}^n)$.

Proof. We know $h(x) \in C_0(R^n)$. From Theorem 2, any continuous function in $C(\bar{R}^n)$ can be approximated by a GFNN respectively. So we can draw the conclusion that $h(x) \in C(\bar{R}^n)$.

The Theorem 3 is proved (relative lemmas seen in reference 5). □

Hence, the system constructed by GFNN can approximate any continuous nonlinear function in $C(\bar{R}^n)$ with any predefined precision. More importantly, the GFNN activation functions are bounded, but not necessarily continuous and this provides a wide choosing range for the GFNN activation functions.

4 The network self-organization and parameter self-learning algorithms of GFNN

Based on the GFNN approximation theorem, for a continuous nonlinear function $y = h(X)$ defined in a close set $K \subset R^n$ and any $\varepsilon > 0$, we can finally find a GFNN by the self-organization and self-learning process, and its output $\hat{y} = \hat{h}(X)$ satisfies $|e| < \varepsilon (e = y - \hat{y})$.

In the identification process of GFNN, the network structure self-organization and parameter self-learning process are going forward simultaneously. Starting from the initial structure, the network self-learning process begins. When a hidden neuron is added or expurgated, the parameters of each neuron will be calibrated by self-learning algorithm to meet the best approximation performance. Now, the algorithm that can automatically adjust the network structure based on the background of the singleton fuzzy generator, product fuzzy inference principle, Gauss membership function and the barycenter defuzzification method is described below as an example.

4.1 Initialization of the GFNN

At time t_0, assume there is a nonlinear system $y = h(\boldsymbol{X})$, and that there always exist $M(t_0)$ ($M(t_0) \geqslant 0$) IF-THEN linguistic fuzzy rules. Based on these $M(t_0)$ rules, we can construct the initial 3-layer feed-forward GFNN with $M(t_0)$ hidden neurons. And the adjustable parameter vector of the jth neuron is $\boldsymbol{\Phi}_j = (\bar{\boldsymbol{X}}_j, \sigma_j)$ $(j = 1, 2, \cdots, M(t_0))$, where $\bar{\boldsymbol{X}}_j = (\bar{x}_{1j}, \bar{x}_{2j}, \cdots, \bar{x}_{nj})$ and $\sigma_j = (\sigma_{1j}, \sigma_{2j}, \cdots, \sigma_{nj})$ are the center position and width vector of the precondition of the fuzzy rules, respectively. The adjustable parameter vector of output layer $\boldsymbol{\Phi}_z = (v_1, v_2, \cdots, v_{M(t_0)})$ is the center position vector of the post-condition of the corresponding fuzzy rules.

4.2 The network self-organization algorithms

1) Introduction

First we set a match degree criterion λ^* in the hidden layer. If the IMDs between the input and hidden neurons are all less than λ^* , then the amount of hidden neurons is inadequate and new neurons should be constructed. If the IMD of the input and a hidden neuron is less than the threshold of this neuron, that is, the information believable level θ^* (IBL), we think this neuron has no effect on the output layer. The expurgation of neurons is processed according to the predefined precision and the match degree criterion, that is, if the distance of the center position of the post-condition of two rules is in the range of the systematic error and the IMD between the two rules is greater than the match degree criterion, then one of the two rules should be expurgated. The process can be done regularly. Based on the automatic addition and expurgation of the hidden neurons, the novel GFNN network on-line self-organization is realized. For the parameters of the hidden neurons, it can be adjusted by the self-learning algorithm, such as BP algorithm or the others. The network self-organization and parameter self-learning are done automatically in the identification process.

2) Network self-learning algorithm

Considering the predefined precision ε, the match degree criterion λ^* and the IBL θ^*, and based on the background of fuzzy logic system, this algorithms is shown as below.

A) Add new hidden neuron

Let the identified network be GFNN($N(t-1)$) at time $t-1$, that is, the network has $N(t)$ hidden neurons at this time. Assume the new training data pair is $(\boldsymbol{X}^{(t)}, y_t)$, where $\boldsymbol{X}^{(t)} = (x_1^{(t)}, x_2^{(t)}, \cdots, x_n^{(t)})$ and $y_t = h(\boldsymbol{X}^{(t)})$, and the output of GFNN is $GFNN(X^{(t)})$. There are two cases.

First: $|GFNN(\boldsymbol{X}^{(t)}) - y_t| < \varepsilon$. This shows that for the current input, the output of GFNN meets the identification precision and no new neuron is necessary to add.

Second: $|GFNN(\boldsymbol{X}^{(t)}) - y_t| > \varepsilon$. This shows that after many adjust steps the system still can not meet the identification requirements. So this case shows that the system structure is inadequate or the predefined parameters are not reasonable, such as the match degree criterion λ^*, IBL θ^*. Therefore, whether a new neuron is added will be according to the detailed circumstance.

a. There exists a number i $(1 < i < N(t))$ such that for the input X, the ith neuron satisfies

$$g_i(\boldsymbol{X}, \Phi_i) = \max_j^{N(t)}(g_j(\boldsymbol{X}, \Phi_j)) > \lambda^*$$

That is, the ith hidden neuron has output but its output has not enough contributions to the total system output. This implicates the match degree criterion λ^* and IBL θ^* are small and each hidden neuron has an output of a wider range, so the system can not derive the accurate output. We can adjust this parameter with many reasonable algorithms.

b. There does not exist a number $i(1 < i < N(t))$ such that for the input $\boldsymbol{X}$, the output of ith neuron satisfies $g_i(\boldsymbol{X}, \Phi_i) > \lambda^*$, i. e., $\max_{1 \leqslant i \leqslant N(t)} g_j(\boldsymbol{X}, \Phi_j) < \lambda^*$. At this circumstance, we think the amount of hidden neurons is inadequate and a new neuron O_t should be added. This neuron comes with the stored parameter vector $\Phi_t = (\bar{\boldsymbol{X}}_t, \sigma_t)$ which satisfies $\begin{cases} \bar{\boldsymbol{X}}_t = \boldsymbol{X}^{(t)} \\ \sigma_t = \sigma_0 \end{cases}$, $v_t = y_t$, where σ_0 is the minimum initial width vector of the specified rule and v_t is output parameter of output neuron.

B) The expurgation of the invalid neuron

The expurgation of neurons is processed according to the predefined precision and the match degree criterion, that is, if the distance of the center position of the post-condition of two rules is in the range of the systematic error and the IMD between the two rules is greater than the match degree criterion, then one of the two rules is redundant and should be expurgated. The process can be done regularly and will influence all hidden neurons.

In all hidden neurons, two of them are selected, O_1 and O_2 their stored parameters are

$\Phi_1=(\bar{X}_1,\sigma_1)$, $\Phi_2=(\bar{X}_2,\sigma_2)$. And the output of the corresponding output layer are v_1, v_2, respectively. If $|v_1-v_2|<\varepsilon$ and $|g_1(\bar{X}_2,\Phi_1)|>\lambda^*$ are satisfied, then one of these two neurons is redundant and is expurgated. Go on with this process repeatedly till all hidden neurons are processed.

3) Network parameter self-learning algorithms

The purpose of parameter self-learning is to reduce the system error to the predefined range by monitoring the error variation. Assume at time t, the error between the output $\hat{y}_t$ and the plant output y is $|e_t|>\varepsilon$. The post-propagation learning algorithm is applied, and the parameter $v_j^{(t-1)}$ of the output layer and the parameter vector $\Phi_j^{(t-1)}$ of the hidden layer are adjusted respectively until $|e_t|<\varepsilon$ is satisfied. The information stored parameters $v_j^{(t)}$ and $\Phi_j^{(t)}$ are derived and the process is advanced to time $t+1$. In the process, the common algorithm is BP algorithm.

4.3 The principle for termination of system identification

In a long period T_0, if the number of hidden neurons does not vary and the error between the output of GFNN, $\hat{y}$, and the plant output satisfies $|e|<\varepsilon$, the structure self-organization and parameter self-learning algorithm will terminate automatically. That is to say, all invalid neurons have been deleted and the system architecture is on the stable. Since GFNN is an on-line identification algorithm, GFNN will converge to a stable state through trainings under the predefined precision when the plant is not varying. When the plant varies, the unsatisfied precision will awake the identification process and the process will continue till the termination conditions are all satisfied.

5 The on-line identification and analysis of the dynamic time-varying system

Assume the plant satisfies the difference equation $y(k+1)=0.3y(k)+0.6y(k-1)+g(u(k))$, and that the unkown part is

$$g(u(k))=\begin{cases}\cos(u(k)), & k<900\\ 0.6\sin(\pi u(k))+0.3\sin(3\pi u(k))+0.1\sin(5\pi u(k)), & 900\leqslant k<1600\\ \dfrac{\cos(u(k))+\sin(u(k))}{2}, & k\geqslant 1600\end{cases}$$

From the plant model, the structure of the plant model varies at time 900 and 1600. We apply GFNN to identify the structure of the plant. We adopt the difference equation based on the series-parallel connection model $\hat{y}(k+1)=0.3y(k)+0.6y(k-1)+\hat{f}(u(k))$. In GFNN, the simulation parameters are set as follows: the system identification precision $\varepsilon=10^{-8}$, Information Match Degree criterion $\lambda^*=0.7$, the minimum Information Believable Level $\theta^*=0.4$, the neuron function is Gaussian function and the initial width parameter $\sigma_0=0.8$, and the initial number of the random selected hidden neurons $M(t_0)=10$. The simulation results are shown in Fig. 3 and the identification error is shown in Fig. 4.

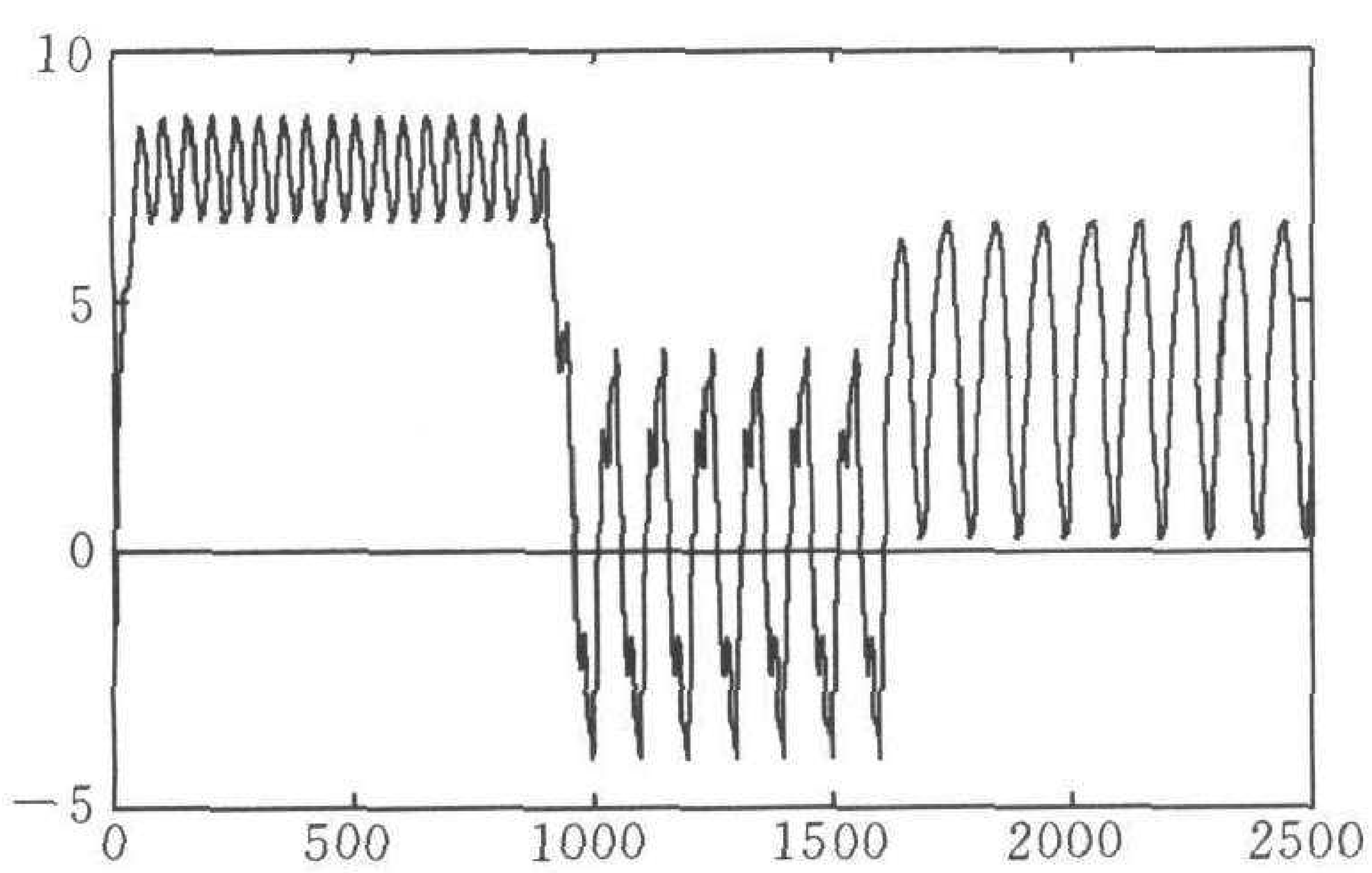

Fig. 3 The identification result

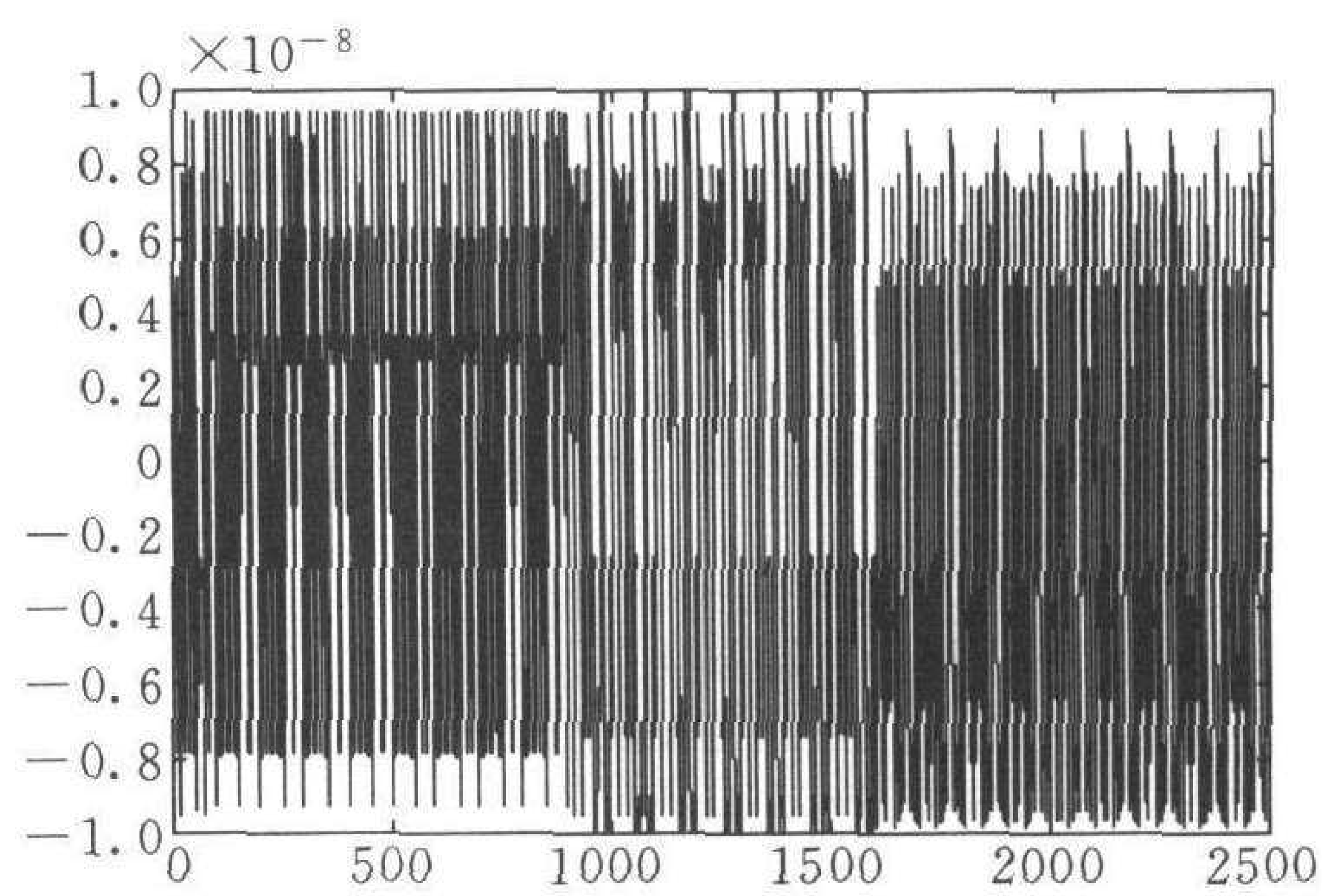

Fig. 4 The identification error

Simulation results

When the system converges, the number of hidden neurons under predefined precision is 59. In 0~900, the system converges at time 130, the number of hidden neurons is 57, In 900~1600, it converges at time 980 with 59 neurons, and after time 1600, it converges at time 1679 with 59 neurons. The maximum identification error after convergence is 9.9213$\times 10^{-9}$(from time 1679).

The analysis of simulation

It shows that GFNN can identify the dynamic time-varying system on-line and can process automatic structure adjustment with parameter self-learning for the structure varying system with high dynamic trace speed and can automatically terminate the identification under predefined precision.

6 Conclusions

This paper extends the Chen approximation theorem and the range of the activation function selection. It shows that any membership function with distance-measure characteristics can be merged into neural network. From the theorem proof, the GFNN functional expression is a simple constructed function. GFNN has the following features.

1) GFNN can realize novel on-line network structure self-organization and this method has general meaning. And the problems of the predefined network structure in neural network are also resolved.

2) GFNN can be used for on-line identification of the time-varying nonlinear system because of its features of neuron expurgation, simple 3-layer structure, small computation and so on. Moreover, GFNN has the background of fuzzy system and the parameters have the explicit physical meaning. So GFNN can be easily realized, on the other hand, the initial values of the adjustable parameters are also easy to setup and these are helpful to improve the convergence speed and the global optimization performance.

3) GFNN has a wide information interface for structural linguistic information and data sample information. This feature makes GFNN a generalized model. The neuron of GFNN has information storage and inference functions as that of the biological neurons, and it provides this simple 3-layer network structure more powerful functions.

4) GFNN can produce many kinds of network structure for the problem resolving based on the fuzzy logical systems composed of many kinds of fuzzy generator, fuzzy inference rules, defuzzier. So it is one of the generalized methods.

Thereby, GFNN is successful integration of fuzzy logical system and artificial neural network. It is shown that fuzzy logical system, artificial neural network and the other outstanding optimization methods will be integrated on a higher layer and compose an integrated intelligent system.

References

1 Wang Yao-Nan. Intelligent Control Systems: Fuzzy Logic, Expert System, Neural Network Control. Changsha: Hunan University Press, 1996 (in Chinese)

2 Bart Kosko. Fuzzy Engineering. Prentice Hall, 1999

3 Xu Li-Hong, Yu You-Ling, Wu Qi-Di. General fuzzy neural network: Basic structure, algorithms and its applications. In: Proceedings of the 14th IFAC (International Federation of Automatic Control) World Congress, Beijing, 1999

4 Jiao Li-Cheng. Theory of Neural Network System. Xi'an: Press of University of Xi'an Electronics & Science Technology, 1990 (in Chinese)

5 Chen Tian-Ping, Chen Hong, Liu Ruey-Wen. Approximation capability in $C(\bar{R}^n)$ by multilayer feedforward networks and related problems. *IEEE Transactions on Neural Networks*, 1995, **6**(1): 126~132

6 Lin Chin-Teng, George Lee C S. Reinforcement structure/parameter learning for neural-network-based fuzzy logic control system. *IEEE Transactions on Fuzzy Systems*, 1994, **2**(1): 46~63

7 Xu Li-Hong, Yu You-Ling, Wu Qi-Di, Shi Jin-Hui. General fuzzy neural network: Theory and applications. In: Pro-

ceedings of 1999 IEEE International Fuzzy Systems Conference Proceedings, Fuzzy-IEEE'99, Seoul, Korea, 1999

YU You-Ling Postdoctor in the Department of Control Theory and Control Engineering at Tongji University, Shanghai. His research interests include intelligent control theory and its application, fuzzy neural network, embedded control system, and design of analog/digital integrated circuits.

XU Li-Hong Professor in the Department of Control Theory and Control Engineering at Tongji University. His research interests include intelligent control theory and its application, fuzzy neural network, and agriculture engineering.

WU Qi-Di Professor in the Department of Control Theory and Control Engineering at Tongji University. Her research interests include intelligent control theory and its application, fuzzy neural network, design of analog/digital integrated circuits, and management information engineering.

广义模糊神经网络

余有灵　徐立鸿　吴启迪

（同济大学信息与控制工程系　上海　200092）

（E-mail：yuyouling@yahoo.com）

摘　要　从非线性系统本身的物理背景出发，根据系统本身的内在特性、先验知识和经验建立系统辨识模型，提出了广义模糊神经网络（GFNN）．文中证明了 GFNN 的函数逼近定理，并据此提出了 GFNN 的结构自组织和参数自学习算法．GFNN 在预设的辨识精度下能自动辨识系统的网络结构以及进行参数自学习，实现 GFNN 网络结构的真正在线自组织．仿真结果表明，对于慢时变非线性对象，GFNN 表现出了很强的非线性逼近能力，是模糊逻辑系统与人工神经网络两类方法的比较成功的融合．

关键词　模糊神经网络，辨识，逼近定理，时变非线性，结构自组织，参数自学习

中图分类号　TP18