Article ID: 1000-8152(2001)03-0395-06

Fault Diagnosis of Nonlinear Systems Based on Modular Fuzzy Neural Networks *

ZHAO Xiang and XIAO Deyun

(Department of Automation, Tsinghua University · Beijing, 100084, P. R. China)

Abstract: A new approach to fault diagnosis based on modular fuzzy neural networks for nonlinear systems is proposed. Firstly, the measurement space has been divided into several subspaces by using fuzzy c-means clustering. Secondly, according to the requirements of fuzzy rules, the subspaces have been fitted by local BP network respectively. Lastly, the characteristics between fault outputs and measuring inputs in different subspaces have been obtained by processing off-line learning. Testing shows the network has good generalization performance and can distinctly improve the speediness, robustness and validity of fault diagnosis in nonlinear systems.

Key words: fault diagnosis; fuzzy neural network; cluster analysis; fuzzy c-means clustering

Document code: A

基于模块化模糊神经网络的非线性系统故障诊断

赵 翔 萧德云

(清华大学自动化系·北京, 100084)

摘要: 提出一种基于模块化模糊神经网络的非线性系统故障诊断新方法. 该方法先使用模糊 c-均值聚类法对测量空间进行模块分割, 再利用模糊 IF-THEN 规则对分割后的子空间分别采用局部 BP 模型进行逼近. 最后, 通过离线学习获得不同子空间故障输出与测量输入的非线性动力特性. 试验表明该网络具有良好的泛化性能, 可显著提高非线性系统故障检测的快速性、鲁棒性及准确率.

关键词: 故障诊断; 模糊神经网络; 聚类分析; 模糊 c-均值聚类

1 Introduction

The fuzzy neural network, which is a merged form of fuzzy system and neural network, can combine their merits and overcome their demerits. In this network, the neural network provides the connectionist structure and learning abilities to the fuzzy logic system, and the fuzzy logic system provides a structuctural framework with high-level fuzzy IF-THEN rule thinking and reasoning for the neural network. In addition, several simple sub-networks are cooperated under a given criterion to deal with a large-scale problem, which is called modular neural network. Its sub-networks, or modules are easily constructed, and their weights and structures can be separately trained in their own way. In this paper, taking account of the validity of modular neural network and fuzzy neural network respectively[1,2], a new approach to fault diagnosis based on modular fuzzy neural network for nonlinear systems is proposed. Firstly, the measurement space has been divided into several subspaces by using fuzzy c-means clustering (FCM). Secondly, according to the requirements of fuzzy rules, the subspaes have been fitted by local BP network respectively. Lastly, the characteristics between fault outputs and measuring inputs in different subspaces have been obtained by processing off-line learning.

2 Fuzzy neural networks for fault diagnosis

2.1 Modular splitting based on cluster analysis

Assume that $(X_i, Y_i, i = 1, 2, \cdots)$ are record pairs

* Foundation item: supported by Special Scientific Research Foundation for Doctoral Subject of Colleges and Universities in China (5108060) and National High Technology R&D Program (863/CIMS) Foundation of China(9845-010).

Received date: 1999-11-22; Revised date: 2001-01-10.

of a system, $X_i = (x_{i1}, x_{i2}, \cdots, x_{im})^T$ and $Y_i = (y_{i1}, y_{i2}, \cdots, y_{in})^T$ separately represent measuring vector and corresponding fault fuzzy state vector obtained by surveying the record data. The problem of fault diagnosis is that it performs a reliable map of measuring vectors into corresponding fuzzy state vectors by means of learning and associating with a large amount of record pairs under a given criterion. In order to simplify the mapping relationship, it is necessary to perform a modular splitting of the measure space. In this paper, a splitting method based on fuzzy clustering is used[3].

Consider a finite set of vectors $X = \{X_1, X_2, \cdots, X_n\}$ as vectors of an m-dimensional Euclidean space $\mathbb{R}^m$, in which $X_j \in \mathbb{R}^m, j = 1, 2, \cdots, n$. The purpose is to perform a partition of the collection of vectors into c fuzzy sets with respect to a given criterion, where c is a given number of clusters. A partition matrix can express the result of fuzzy clustering

$$U = [u_{ij}]_{i=1,\cdots,c, j=1,\cdots,n}, \tag{1}$$

where $u_{ij} \in [0,1]$ is a numeric value and expresses the degree to which the vector X_j belongs to the i th cluster. However, there are two additional constraints on the value of u_{ij}. That is

$$\begin{cases} \sum_{i=1}^{c} u_{ij} = 1, \text{ for all } j = 1, 2, \cdots, n; \\ 0 < \sum_{j=1}^{n} u_{ij} < n, \text{ for all } i = 1, 2, \cdots, c. \end{cases} \tag{2}$$

A general form of the objective function is

$$J(u_{ij}, V_k) = \sum_{i=1}^{c} \sum_{j=1}^{n} \sum_{k=1}^{c} g[w(X_i), u_{ij}] d(X_j, V_k), \tag{3}$$

where $w(X_i)$ is a prior weight for each X_i and $d(X_j, V_k)$ is the degree of dissimilarity between the data X_j and the supplemental element V_k, which can be considered the central vector of the kth cluster. The degree of dissimilarity is defined as

$$\begin{cases} d(X_j, V_k) \geqslant 0, \\ d(X_j, V_k) = d(V_k, X_j). \end{cases} \tag{4}$$

Obviously, it is a concept weaker than distance measures. With the above background, fuzzy clustering can be precisely formulated as an optimization problem

$$\begin{cases} \min J(u_{ij}, V_k), i, k = 1, 2, \cdots, c;\ j = 1, 2, \cdots, n, \\ \text{s.t. Eq. (2).} \end{cases} \tag{5}$$

One of the widely used clustering methods based on Eq. (5) is the FCM algorithm developed by Bezdek, of which objective function is

$$J(u_{ij}, V_k) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \| X_j - V_i \|^2, \ m > 1, \tag{6}$$

where m is called the exponential weight which influences the degree of fuzziness of the membership or partition matrix. To solve this minimization problem, we first differentiate the objective function in Eq. (6) with respect to V_i (for fixed $u_{ij}, i = 1, 2, \cdots, c,\ j = 1, 2, \cdots, n$) and to u_{ij} (for fixed V_i, $i = 1, 2, \cdots, c$) and apply the conditions of Eq. (2), obtaining

$$V_i = \frac{1}{\sum_{j=1}^{n} (u_{ij})^m} \sum_{j=1}^{n} (u_{ij})^m X_j, \ i = 1, 2, \cdots, c, \tag{7}$$

$$u_{ij} = \frac{(1/\| X_j - V_k \|^2)^{1/m-1}}{\sum_{k=1}^{c} (1/\| X_j - V_k \|^2)^{1/m-1}}, \tag{8}$$

$$i = 1, 2, \cdots, c;\ j = 1, 2, \cdots, n.$$

The system described by Eqs. (7) and (8) cannot be solved analytically. However, the FCM algorithm provides an iterative approach to approximating the minimum of the objective function starting from a given position. This algorithm is summarized in the following:

Step 1 Select a number of clusters $c (2 \leqslant c \leqslant n)$ and expoential weight $m (1.5 \leqslant m \leqslant 30)$. Choose initial partition matrix $U^{(0)}$ and a termination criterion ε. Set the iteration index $l = 0$.

Step 2 Calculate the fuzzy cluster centers $\{V_i^{(l)} \mid i = 1, \cdots, c\}$ by using $U^{(l)}$ and Eq. (7).

Step 3 Calculate the new partition matrix $U^{(l+1)}$ by using $\{V_i^{(l)} \mid i = 1, \cdots, c\}$ and Eq. (8).

Step 4 Calculate $\Delta = \| U^{(l+1)} - U^{(l)} \| = \max_{i,j} | u_{ij}^{(l+1)} - u_{ij}^{(l)} |$. If $\Delta > \epsilon$, then set $l = l + 1$ and go to step 2; otherwise, stop looping.

An important question for the FCM algorithm is how to determine the "correct" number of clusters, c. Some scalar measures of partitioning fuzziness have been used as synthetic indices to point out the most plausible number of clusters in the data set.

1) Partitioning entropy:

$$H(U, c) = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{c} | u_{ij} \ln u_{ij} |;$$

2) Partitioning coefficient:

$$F(U,c) = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^{2};$$

3) Proportion exponent:

$$P(U,c) = -\ln\left\{\prod_{j=1}^{n}\left[\sum_{k=1}^{u_j^{-1}}(-1)^{k+1}\binom{c}{k}(1-ku_j)^{c-1}\right]\right\},$$

where $u_j = \max_{1\leqslant i\leqslant c} u_{ij}$ and u_j^{-1} = greatest integer $\leqslant (1/u_j)$.

We adopt the following heuristic rules for selecting the best partitioning number c. They are

$$\min_{c=2}^{n-1}\{\min_{U\in\Omega_c}[H(U,c)]\}, \max_{c=2}^{n-1}\{\max_{U\in\Omega_c}[F(U,c)]\},$$

or

$$\max_{c=2}^{n-1}\{\max_{U\in\Omega_c}[P(U,c)]\},$$

where Ω_c is the set of all optimal solutions to a given c.

Divide the record pairs $(X_i, Y_i), i = 1,2,\cdots$ into two parts, of which one is used for training netwrok (set the number to be N_t), the other is used for checking and evaluating network (set the number to be N_c). It is easy to cluster the training samples into r clusters and realize the modular splitting to the measure space. For expressing covenience, assume that $\mathbb{R}^s (s = 1,2,\cdots,r)$ represents clusters and N_s represents the number of samples in $\mathbb{R}^s$.

2.2 Model of modular fuzzy neural networks

The basic idea of using fuzzy neural networks to realize fault diagnosis is to implement the membership functions in the preconditions as well as the inference function in the consequence by proper topology structure and learning rules. One scheme for generalization of the fault diagnosis using fuzzy neural networks is the neural network-driven fuzzy reasoning (NDF). That is

Rule s: IF $X \in \mathbb{R}^s$, THEN $Y_s = NN_s(X)$, $s = 1,2,\cdots$. (9)

where $X = (x_1, x_2, \cdots, x_m)^{\mathrm{T}}$ represents a measuring vector, and $\mathbb{R}^s$ represents a fuzzy subspace formed after modular splitting or represents a fuzzy cluster while FCM is used, and $NN_s(X)$ denotes a structure of model function characterized by a neural network. The number of measure variables employed in $NN_s(X)$ is determined by a method for selecting the optimum model described later. The neural network-driven fuzzy reasoning system has been derived in Fig.1.

The above structure includes $r + 1$ BP networks, where NN is the neural network that determines the conjoint membership functions of the preconditions of all rules, and $NN_1 \sim NN_r$ are the neural networks that determine the detecting values for each rule.

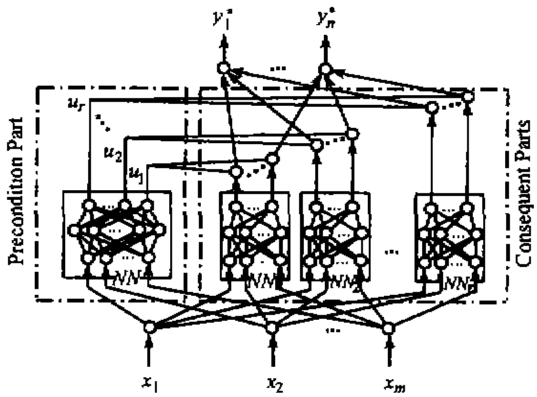

Fig. 1 Model structure of NDF reasoning system

2.3 Learn algorithm of fuzzy neural networks

Meaning for the fuzzy neural network model as showed in Fig.1, we must process the following tasks. Firstly, complete the modular splitting to measurement space and determine the number of fuzzy rules. Secondly, construct a neural network used to calculate the conjoint membership functions of measuring vectors in preconditions. Lastly, construct several neural networks used to map the nonlinear relationship in consequence.

As mentioned before, $X_i(x_{i1}, x_{i2}, \cdots, x_{im})^{\mathrm{T}}$ represents the system's measuring vector and $Y_i = (y_{i1}, y_{i2}, \cdots, y_{in})^{\mathrm{T}}$ represents the corresponding fuzzy state vector. Furthermore, each element of vector Y_i corresponds to a type of fault pattern. The number of fuzzy rules can be determined by means of processing FCM to measuring data. Assume that teaching signal adopts the fuzzy state vector, and transfer function adopts a sigmoid function. Thus, the design procedure can be described in the following steps.

Step 1 Divide the record pairs $(X_i, Y_i), i = 1,2,\cdots,N$ into two parts arbitrarily. One is used to train neural networks (set the number to be N_t) and the other is used to check or evaluate the performance of neural networks (set the number to be N_c).

Step 2 Divide the training samples into r classes of $\mathbb{R}^s$ by FCM, where $s = 1,2,\cdots,r$. The training samples for each $\mathbb{R}^s$ are expressed by $(X_i^s, Y_i^s), i = 1,2,\cdots,N_s$, provided that N_s is the number of training samples for $\mathbb{R}^s$.

Step 3 Train NN corresponding to the preconditions of all rules. For each measuring vector of training samples $X_i \in \mathbb{R}^s$, define a vector $U_i = (u_{i1}, u_{i2}, \cdots, u_{ir})^T$ such that $u_{ik} = 1$ for $k = s$ and $u_{ik} = 0$ for $k \neq s$. The NN with m input nodes and r output nodes is then trained on the input-output pairs (X_i, U_i), $i = 1,2,\cdots, N_t$. Hence, the NN becomes capable of inferring the conjoint membership functions $\hat{u}_{is}$ of each checking spample X_i to $\mathbb{R}^s$. The membership function of the IF part is thus defined as the fitness value $\hat{u}_{is}$ that is the output of the learned NN, that is,

$$\mu_{\mathbb{R}^s}(X_i) = \hat{u}_{is},\ i = 1,2,\cdots,N; s = 1,2,\cdots,r. \tag{10}$$

Step 4 Train $NN_1 \sim NN_r$ corresponding to the THEN part of the rules. The training samples (X_i^s, Y_i^s), or $x_{i1}^s, x_{i2}^s, \cdots, x_{im}^s$ and $y_{i1}^s, y_{i2}^s, \cdots, y_{in}^s (i = 1,2,\cdots, N_s)$, are assigned to the input and output of the NN_s. The training of NN_s is conducted so that the detecting values can be inferred. The checking samples $x_{i1}, x_{i2}, \cdots, x_{im} (i = 1,2,\cdots,N_c)$ are substituted into the obtained NN_s to obtain the sum E_m^s of the squared error

$$E_m^s = \sum_{i=1}^{N_c} \sum_{j=1}^{n} |y_{ij} - \mu_{sj}(X_i)\mu_{\mathbb{R}^s}(X_i)|^2, \tag{11}$$

where the estimated $\mu_{sj}(X_i) (j = 1,2,\cdots,n)$ is obtained as the output of NN_s.

Step 5 Simplify the THEN parts by a backward elimination method. Among the m measure variables of NN_s, on variable x_p is arbitrarily eliminated, and the NN_s is trained again by using the traning samples as in Step 4. The squared error E_{m-1}^{sp} of the output value of the s^{th} rule in the case of eliminating x_p can be estimated by using checking samples

$$E_{m-1}^{sp} = \sum_{i=1}^{N_c} \sum_{j=1}^{n} |y_{ij} - \mu_{sj}(\hat{X}_i)\mu_{\mathbb{R}^s}(\hat{X}_i)|^2, \quad p = 1,2,\cdots,m. \tag{12}$$

where $\hat{X}_i = (x_{i,1}, \cdots, x_{i,p-1}, x_{i,p+1}, \cdots, x_{i,m})^T$. Comparing Eqs. (11) and (12), if $E_m^s > E_{m-1}^{sp}$, then the significance of the eliminated measure variables x_p can be considered minimal and x_p can be discarded. The same operations are carried out for the remaining $(m-1)$ input variables. The elimination process is repeated until $E_m^s > E_{m-1}^{sp}$ will not hold for any remaining measure variables. The model that gives the minimum E^s value is the best NN_s.

Step 6 Decide on final output. The following equation can derive the final value

$$y_{ij}^* = \frac{\sum_{s=1}^{r} \mu_{\mathbb{R}^s}(X_i)\mu_{sj}(X_i)}{\sum_{s=1}^{r} \mu_{\mathbb{R}^s}(X_i)},$$

$$i = 1,2,\cdots,N;\ j = 1,2,\cdots,n.$$

3 Application example

The fuzzy neural network model described in the previous sections has been implemented in a chemical process[4] shown in Fig. 2 for the off-line test.

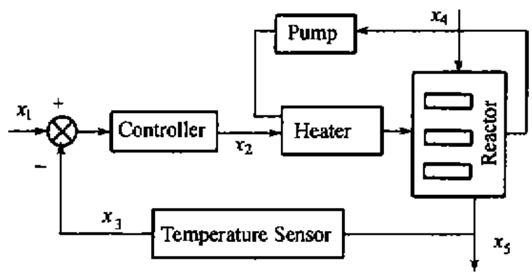

Fig. 2 Chemical process used to test

According to the actual situation in this chemical process, four types of faults may occur during running.

Fault 1: The performance of catalyst may be degenerated (the occurring membership function of this fault is represented by y_1).

Fault 2: The surfaces of heat exchanger in reactor may be jammed (the occurring membership function of this fault is represented by y_2).

Fault 3: The surfaces of heater may the jammed (the occurring membership function of this fault is represented by y_3).

Fault 4: The connecting pipeline may be jammed in local sites (the occurring membership function of this fault is represented by y_4).

The actual data acquired at the test site are as follows: the set point signal x_1, the output signal of controller x_2, the outlet temperature of heater x_3, the rate of inlet flow x_4, and the outlet consistency x_5. The fault state vectors $(y_1 \sim y_4)$ of the process are obtained by means of researching workday monitored records and investigating spot operators. These data pairs (see Table 1) reflect the relationships between measuring parameters and corresponding fault modes.

Table 1 Data pairs of the process

	No.	x_1(mV)	x_2(mV)	x_3(K)	x_4(m^3/h)	x_5(mol/m^3)	y_1	y_2	y_3	y_4
Training data	1	802.6	301.874	1197.99	464.907	4489.817	0.15	0.86	0.48	0.03
	2	682.21	234.003	1317.789	289.942	5002.078	0.05	0.71	0.79	0.1
	3	842.73	243.12	878.526	454.909	6418.329	0.93	0.22	0.14	0.15
	4	1163.77	244.133	1357.722	349.93	7322.319	0.89	0.24	0.48	0.11
	5	682.21	220.834	1158.057	239.952	8015.378	0.56	0.62	0.21	0.09
	6	521.69	290.731	638.928	234.953	6990.856	0.13	0.04	0.68	0.02
	7	1003.25	259.328	519.129	224.955	6689.526	0.14	0.2	0.71	0.07
	8	1003.25	293.77	798.66	294.941	5454.073	0.93	0.22	0.14	0.03
	9	1123.64	262.367	1317.789	394.921	4128.221	0.11	0.81	0.07	0.08
	10	1043.38	293.77	319.464	464.907	2320.241	0.91	0.77	0.14	0.04
	11	922.99	278.575	598.995	514.897	2079.177	0.13	0.94	0.2	0.11
	12	1003.25	287.692	479.196	324.935	4580.216	0.89	0.15	0.24	0.08
	13	1203.9	286.679	439.263	429.914	3013.3	0.16	0.31	0.74	0.13
	14	762.47	230.964	718.794	429.914	2621.571	0.14	0.48	0.77	0.17
	15	601.95	281.614	638.928	489.902	2259.975	0.91	0.25	0.33	0.02
	16	682.21	262.367	878.526	499.9	3766.625	0.46	0.17	0.67	0.06
	17	441.43	281.614	998.325	549.89	5333.541	0.52	0.44	0.4	0.22
	18	762.47	232.99	1118.124	429.914	6448.462	0.59	0.55	0.67	0.25
	19	922.99	224.886	998.325	239.952	8015.378	0.2	0.9	0.63	0.04
	20	441.43	290.731	1517.454	409.918	8648.171	0.09	0.15	0.27	0.73
Checking data	21	1003.25	317.069	798.66	429.914	8045.511	0.51	0.55	0.09	0.07
	22	1003.25	313.017	638.928	269.946	6990.856	0.66	0.25	0.31	0.01
	23	1203.9	308.965	678.861	254.949	5755.403	0.14	0.06	0.57	0.06
	24	1203.9	294.783	838.593	354.929	4278.886	0.49	0.22	0.16	0.1
	25	1003.25	296.809	718.794	369.926	3766.625	0.57	0.18	0.11	0.03
	26	963.12	292.757	878.526	469.906	2802.369	0.64	0.13	0.24	0.09
	27	1003.25	304.913	838.593	469.906	2862.635	0.6	0.09	0.19	0.01
	28	1003.25	295.796	838.593	454.909	3887.157	0.46	0.47	0.03	0.04
	29	842.73	261.354	1158.057	499.9	4821.28	0.54	0.17	0.62	0.14
	30	642.08	231.977	1197.99	499.9	5936.201	0.52	0.71	0.47	0.12
	31	1203.9	224.886	798.66	249.95	6870.324	0.13	0.59	0.05	0.03
	32	401.3	262.367	1317.789	499.9	8497.506	0.17	0.22	0.15	0.85
	33	1123.64	310.991	838.593	319.936	7412.718	0.49	0.7	0.2	0.07
	34	922.99	304.913	1038.258	294.941	6689.526	0.23	0.42	0.74	0.1
	35	1003.25	273.51	798.66	334.933	5393.807	0.83	0.26	0.16	0.11
	36	1605.2	284.653	598.995	374.925	4309.019	0.2	0.14	0.67	0.02

The following steps are performed one by one.

Step 1 The clustered results are shown in Table 2. Therefore, the fuzzy system can be expressed in three rules.

Table 2 Clustered results of training samples

Rule	The sequence number of training samples
1	No.1, No.9, No.10, No.11, No.12, No.15
2	No.2, No.3, No.4, No.5, No.6, No.16, No.19, No.20
3	No.7, No.8, No.13, No.14, No.17, No.18

Step 2 Separtely train NN and NN_1, NN_2, NN_3 about 1500 epochs. Each network has two hidden layers, and the number of nodes in each hidden layer is determined as 5 and 3 by trial-and-error. The following rules are obtained

Rule 1: IF $X = (x_1, x_2, x_3, x_4, x_5)^T \in \mathbb{R}^1$,

THEN $Y = (y_1, y_2, y_3, y_4)^T = NN_1(x_1, x_2, x_4, x_5)$;

Rule 2: IF $X = (x_1, x_2, x_3, x_4, x_5)^T \in \mathbb{R}^2$,

THEN $Y = (y_1, y_2, y_3, y_4)^T = NN_2(x_1, x_2, x_3, x_4, x_5)$;

Rule 3: IF $X = (x_1, x_2, x_3, x_4, x_5)^T \in \mathbb{R}^3$,

THEN $Y = (y_1, y_2, y_3, y_4)^T = NN_3(x_1, x_3, x_4, x_5)$.

According to the above three rules, we input all the record pairs into diagnostic model and obtain the detecting results as shown in Fig. 3. In order to test the ability of cancellation to measure noises, we have superimposed white noises to each measure variables. In Fig. 3, we only present the detecting results of the membership function y_1, which tallies with the monitored results. There are similar detecting results for the other membership function (y_2, y_3 and y_4).

Fig. 3 illustrates the actual output sequence and the model output sequence for both the training samples (from No. 1 to No. 20) and checking samples (from No. 21 to No. 36). The diagnostic model has been trained with only training samples, it still maps the checking samples very good. Obviously, the diagnostic model based on the fuzzy neural network has perfect generalization performance and tracking ability.

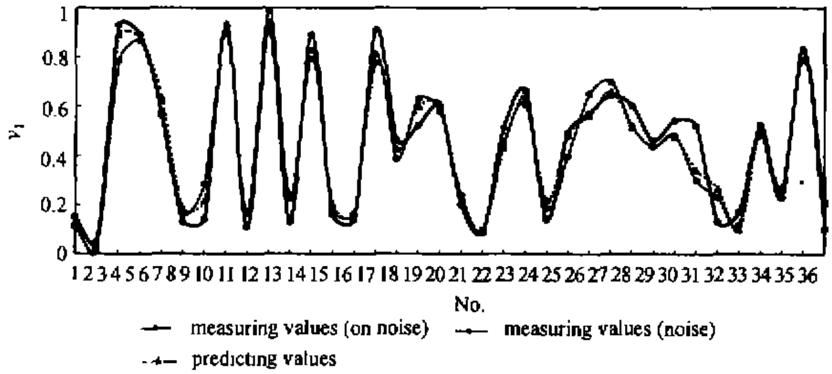

Fig. 3 Actual output and model output of fault membership function

4 Conclusions and further discussions

The simulation results show that the proposed structure and principle of fuzzy neural network for nonlinear systems are feasible. They can be used to reailze online fault diagnosis in real time. The method has strong robust for measuring noises, and can effectively improve the correctness and speediness of fault diagnosis. However, the training samples obtained by past record pairs are limited and cannot cover the whole working conditions of the detecting systems. In view of the localizations of sample spaces and time-varying properties of system parameters, the off-line learning and on-line learning should be incorporated in complex application in order to improve the diagnostic robustness.

References

[1] Dong Xuanmin, Qiu Lihua and Wang Zhanlin. Adaptive nonlinear filter neural network based and its applications in on-line fault diagnose [J]. Control and Decision, 1997, 12(1): 78 – 82

[2] Jang J S R. ANFIS: Adaptive-network-based fuzzy inference system [J]. IEEE Trans. on System, Man and Cybernetics, 1993, 23(3): 714 – 723

[3] Lin Chin-Teng and Georgelee C S. Neural Fuzzy Systems [M]. New Jersey: Prentice-Hall International, Inc., 1996

[4] Wang Yunji and Tu Jian. Neural Networks Controller [M]. Beijing: Pubilsher of Mechanical Industry, 1988

本文作者简介

ZHAO Xiang Born in 1973, received the Ph.D. degree in 2000. Works in Logistics Engineering University of the People's Liberation Army and acts now as a visiting scholar in Department of Automation, Tsinghua University. Research interests include intelligent fault diagnosis based on neural network, information fusion, evolution computing and so on.

XIAO Deyun Born in 1945, graduated from Tsinghua University. Now be is a professor and Ph.D. advisor in Department of Automation, Tsinghua University. The main area of interest and expertise is identification modeling, fault diagnosis, multi-sensor fusion, computer control, CIMS in process industries and so on.