Optimization and Implementation of Self-Organization Fuzzy Neutral Network Control Algorithm

Zeng Xiao-hui[1,2,*] and Shi Yi-bing[1]

[1]*School of Automation Engineering, UESTC, 610000, China*

[2]*School of Communication Engineering, CUIT, 610000, China*

Abstract: Traditional fuzzy neural network tends to be used for parameters identification and the network structure is separated by grids, so there are obvious defects in this mode of control design. This paper introduces a flexibly model of self-organizing fuzzy neural network according to specific network structure. We analyze its structure and containing parameters and propose an improved nearest neighbor clustering algorithm first for the predicting model of online identification. For parameter optimization, the parameter value acquired at self-organizing learning phase is adopted as the initial value of supervised learning. Then it adopts BP algorithm to adjust the parameter to optimal value based on the same training set, so as to acquire the final model of FNN. The experiments demonstrate that our algorithm can solve the predicting problems of nonlinear system with constraints, and the range and changing rate of control signal. It shows rapid computing speed, better stability and strong anti-disturbance capacity. It is also verified to be suitable for actual engineering control environments.

Keywords: SOFNN, İdentification, Adaptive parameter, Membership function, Clustering.

1. INTRODUCTION

The appearance of fuzzy neural network (FNN) control aims to adjust various information to connect control the methods of different forms, to be closely and mutually related comprehendingly. Self-feedback process refers to constantly contrast between given signals and output signals to produce deviation signals, when regulating the fuzzy neural network control [1,2]. Then this deviation signal is applied to control and reduce difference and keep stable state. When output signal is basically consistent with given signal and output signal satisfies the requirement, internal information flow of the whole system will keep stable by feedback and its internal structure is in stable status. If the information of feedback factor has contained some new contents, which has difference from original status, FNN control system can be supplemented, modified and regulated to some extent by self-feedback to ensure that the whole system is normally operated in dynamism. Self-adaptive ability refers to depend on sufficient information exchange and feedback between systems to realize the balance of the whole system, without relying on external compulsory power. This adaptive ability is the aim of self-organized system and is a purposeful activity [3]. It indicates that FNN control system performs self-renewal through realizing self-adaptive function to form new ordered structure. In addition, based on exterior environment change, it regulates its structure and changes its function to be adaptive to the change of environment. Three self-organization functions of automorphism, self-feedback and self-adaptation have their various characteristics in FNN control [4-6]. However, they are not isolated but mutually related and interacted. Only when these three functions are comprehensively acted, can self-organization function of fuzzy neural network control be displayed. Traditional PI and fuzzy controller control rely on fuzzy inference to realize simple control and to imitate humans' decision behaviors. They do not rely on object model too much and show advantages such as perfect adaptability [7, 8]. However, it is difficult to make artificial extraction for control rules of complex object. The controlled object information is not fully implemented to cause reduction of control accuracy. Since algorithm is rather simple, it cannot be adapted to parameter change of the controlled object to rely on humans' priori experience. Self-Organization FNN (SOFNN) is a kind of FNN whose owns self-learning ability in structure and parameters [9]. It learns from the structure as well as parameters in neural network during nonlinear system control and on-line regulates structure of fuzzy neural network, and produces fuzzy control rules and regulates control rule parameters to change the control performance.

Since the structure of FNN is difficult to be determined [10], Akhbardeh proposes fuzzy adaptive resonance theory (ART) and inferior methods in [11] to train the structure and parameters of fuzzy adaptive learning control network. With the increasing of input data, this strategy can divide input/output space online, regulate membership function, find out appropriate fuzzy logic rules and dynamically delete redundant rules. Akhbardeh also presents a novel ANFIS framework-based neural fuzzy system. More specifically, he proposes a concurrent clustering algorithm to establish new rule base and optimize this rule base with back-propagation method in practical environment. GU and Zhang [12] combine the vector isolation and adaptive resonance theory as clustering algorithm of FNN structure identification. Oentaryo and Pasquier [13] propose a new FNN system-Generic Self-Organizing GSOFNN which applies discrete enhanced clustering algorithm to learn fuzzy neural network structure. The rule base of GSOFNN is constant and compact because it has a set of mechanisms to delete redundant rules. Based on previous jobs, we can discover that online clustering algorithm of structure learning is very important. Effectively dividing input/output data can provide a relatively accurate initial fuzzy neural network structure, which helps to the convergence of network. Therefore, various clustering algorithms are used to divide input/output space such as Kohonen hierarchical clustering algorithms, hyperbox method, k-means, and fuzzy C-mean method [14].

By the study and generation of above problems, this paper adopts an improved clustering algorithm to identify the structure of FNN and to learn parameters of FNN with back-propagation (BP) method. This clustering algorithm can divide the input data online and gradually update clustering and organize FNN structure by itself. It does not need to know the input data distribution and all rules can be created by itself. With input data increasing, the rule number will automatically increase and the fuzzy neural network after training does not have contradictory rules. By simulation example and actual cases of control engineer, it is proved that our SOFNN method is effective and shoes advantages such as simple structure, high identification accuracy, etc.

2. STRUCTURE OPTIMIZATION

The SOFNN controller is a neural network-based self-organization fuzzy controller. This controller does not need to determine fuzzy control rules in advance. Its control rules and membership functions are operated by a neural network but the adjustment of rule revision and membership function parameters is realized by BP learning algorithm. Through structure and parameter learning of neural network in the process of nonlinear system control, it can regulate structure of fuzzy neural network online, produce fuzzy control rule and regulate control rule parameters to change control performance. The controller in SOFNN is realized by neural network, which has a hierarchical structure [15]. The bottom layer is input layer, the middle is hidden layer and the top is output layer. Its information is gradually transferred upward from input layer to output layer. Then, according to error property index, from output node, gradient descent algorithm is applied in each layer of network to modify the connection weight and adjust the membership function parameters by reverse channel. It does not need to adopt the scale factors K_e, K_{ee} or K_u, but it needs to use center and width of membership function as well as connecting weight coefficients between networks, etc.

Classical FNN structure divisions have obvious defects: one is that the spatial division needs to be provided in advance based on experience. The network structure keeps stable during learning and it is easy to cause lacking and redundancy of fuzzy rules; The other one is, for complex system, the fuzzy rule numbers are in exponential growth to bring dimension disaster with increasing input variable numbers.

To solve the problems mentioned above, this paper first introduces a novel SOFNN proposed in reference [16], to study its optimized structure and parameter. Without loss of generality, we consider a multi-input single-output fuzzy system. The system contains fuzzy inference rules and adopts zero order T-S fuzzy inference computing model. pieces of the rules are: R_k: If x_1 is A_1^k AND…AND x_n is A_n^k. Then y_k is w_k. $x=\{x_i\}, i=1,2,...n$ is input variable and is conclusion variable. A_n^k is the membership function of input variable and it takes Gauss function as the form of the following equation:

$$\mu_{ik}(x_i)=\exp\{-(x_i-c_{ij})^2/\delta_{ij}^2\} \tag{1}$$

μ_{ik} is k_{th} membership function of variable x_i. c_{ij} and δ_{ij} denote the central value and width of Gauss function. n is the number of input variables and L is the number of membership functions of each input variable. The topology of SOFNN in this paper is depicted as Fig. (**1**). Its structure determines the amount of fuzzy rules and membership functions. Weight parameter determines the expression of rules and the shape of membership function.

The first layer is input layer. The neurons of this layer are directly connected with input variables and the number of nodes equal to that of input variables; The second layer is fuzzed layer. Assuming x_1 and x_2 are two input variables and they adopts several fuzzy variables. A_i is language variable of fuzzy set corresponding to node input value. Then the output of node is the membership of fuzzy set corresponding to itself:

$$O_i^2=\mu_{A_i}(x) \tag{2}$$

The membership function has many forms, such as Bell-shaped membership:

$$\mu_{A_i}(x)=\frac{1}{[(\frac{x-c_i}{a_i})^2]^{b_i}} \tag{3}$$

$\{a_i, b_i, c_i\}$ are antecedent parameters whose values determine the form of $\mu(\bullet)$.

In the third layer, i_{th} node denotes the ratio between the trigger intensity of i_{th} rule and all the trigger intensities, which presents the normalization. The output of this layer is normalized trigger intensity and is described as

$$O_i^3=\bar{w}_i=\frac{w_i}{w_1+w_2} \tag{4}$$

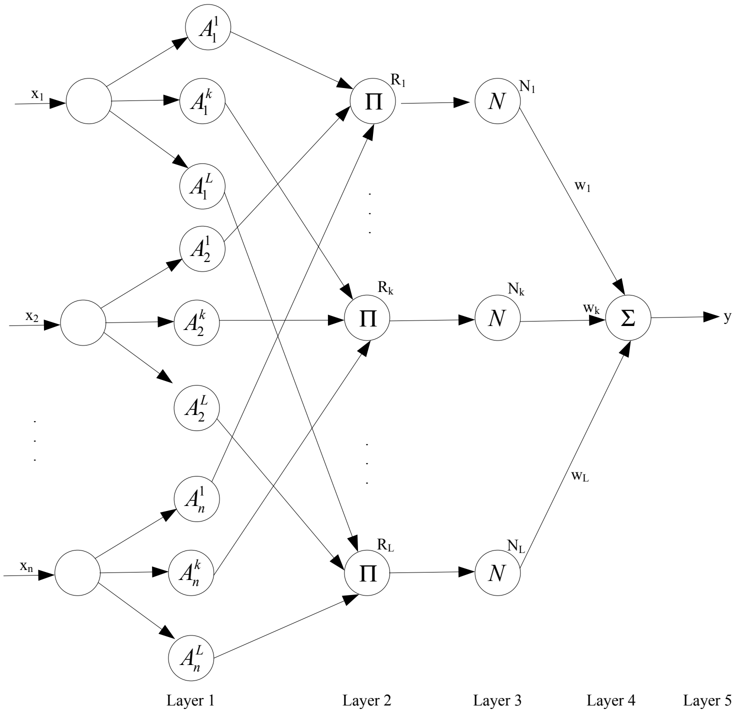


Fig. (1). Structure of self-organization fuzzy neural network.

Each node of the fourth layer is adaptive node and its output is

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i) \tag{5}$$

$\{p_i, q_i, r_i\}$ is parameters set which denotes the inferior parameters. The output of the fifth layer is total output value and its ambiguity resolution adopts gravity method:

$$O_i^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{6}$$

This adaptive FNN equals to a fuzzy inference system with T-S rules in function [17]. Its optimization is adopting the existing learning algorithms and absorbing the advantage ANN and FS, offsetting their defects simultaneously. When its structure is determined, the follows will discuss the problems on structure identification and parameters adjusting.

3. LEARNING METHOD OF SOFNN

The design of fuzzy neural network contains structure identification and parameter identification [18]. Structure identification contains input space division and fuzzy rule number determination. For traditional FNN, universe number of input variable division needs to be given in advance by experience. However, for different problems objects, they lack an efficient guiding principle so it is very difficult to obtain the best division, which usually causes deficiency and redundancy of fuzzy rule numbers. Meanwhile, for complicated multi-input variable systems, traditional division method will generate bottleneck of dimension disaster. Then the SOFNNN design in this paper is divided into two parts: structural self-organization design of FNN and parameter optimization learning.

3.1. Analysis and Improvement of System Identification Model

We assume the fuzzy rule library is composed of the following fuzzy rules [19]. $R_u^{(l)}$: If x_1 is A_1^l and ,…, and x_n is A_n^l, then y is B^l. A_i^l and B^l are fuzzy sets on $U_j \subset R$ and $V \subset R$, $x = (x_1, x_2, ..., x_n)^T \in U$ and $y \in l$ denote the input and output language variables of fuzzy system. M is the number of rules in library, that is , $l = 1, 2, ..., M$.

Given input and output pairs (x_0^l, y_0^l), if we want to construct a fuzzy system $f(x)$, it can fit *N* pairs of data to any accuracy. It means for any $\varepsilon > 0$, there is $| f(x_0^l) - y_0^l) | < \varepsilon$. Then the FNN is established as equation 7:

$$f(x) = \frac{\sum_{l=1}^{N} y_0^l \exp(-\frac{|x - x_0^l|^2}{\sigma^2})}{\sum_{l=1}^{N} \exp(-\frac{|x - x_0^l|^2}{\sigma^2})} \tag{7}$$

The above equation denotes the FNN actually is constructed by *N* rules. $\mu_{A_i^l}(x_i) = \exp(-\frac{|x - x_0^l|^2}{\sigma^2})$ and the center of B^l is y_0^l. The fuzzy system adopts product inference machine, single-valued fuzzy controller, center mean defuzzifier and Gauss membership function. This equation also shows that the optimal FNN system is applying a rule to an input and output pair. But if the number of pairs is large, this system is no more a actual system. For large samples, we need different clustering technique to group the input and output pairs, to make some group of data correspond to one rule.

Among the clustering algorithms, the nearest neighbor-clustering algorithm is the simplest clustering algorithm. In this algorithm, the first data is taken as the clustering center of the first group at first. Next, if the distance between one data and clustering center is smaller than one predictive value, this data is put in this group. That is, the clustering center in this group should be the most approximate to this data; Otherwise, this data is set as clustering center of a new group. When the nearest neighbor-clustering algorithm is used to design FNN, if each input and output data pair establishes one clustering center, the designed FNN is changed into the optimized fuzzy system. Since the optimized fuzzy system means that one rule corresponds to one input and output data pair, the nearest neighbor-clustering algorithm is considerer as that one rule is corresponding to one group of input and output data pair. Each input and output data pair may produce one new clustering, so the numbers of designed rules in fuzzy system is constantly changing during the process of design.

It needs to set a radius r when using the nearest clustering method to design fuzzy system. The number of rules is up to input point distribution and radius of input-output data pairs, r determines the complexity of fuzzy system. In practice, the method of trial and error can be used to find out an appropriate radius to ensure that the involved fuzzy system is simple and convinced. This paper proposes an online identification system model of improved nearest clustering method. This algorithm contains two stages: initial off-line identification and online identification, which are described as the follows:

At the offline phase, we collect N pairs of input and output data (x^l, y^l), $l = 1, 2, .., N$. x^l is input vector of model and y^l is output vector of model; The radius r is initialized and the nearest neighbor clustering method is add to SOFNN system as equation 7.

At the online phase, when we get a group of new input and output data (x^{new}, y^{new}), we will compute the Euclidean distance of it with the other N groups of data pairs (x^l, y^l). If the flowing inequality does not stand, we believe (x^{new}, y^{new}) is similar to (x^l, y^l).

$$\| (x^{new}, y^{new}) - (x^l, y^l) \| < \varepsilon \tag{8}$$

ε is a very small value and $1 \le \varepsilon \le r$, $1 \le l \le N$. If no input and output data pair is similar to (x^{new}, y^{new}), the model will be modified as the follows:

Compute the Euclidean distance of and clustering centers. Assuming the shortest distance is $\| x^{new} - x_c^k \|, (1 \le k \le M)$, it means that x_c^k is the nearest neighbor clustering of x^{new}. M is the number of clustering centers, x_c^l $(l = 1, 2, ..., M)$ is clustering center. A^l, B^l are computed value.

(1) If $\| x^{new} - x_c^l \| > r$, is a new clustering center. Let $x_c^{M+1} = x^{new}$, $A^{M+1} = y^{new}$, $B^{M+1} = 1$, then $M = M + 1$.

(2) If $\| x^{new} - x_c^l \| \le r$, $A^l = A^l + y^{new}$ let and $B^l = B^l + 1$.

So the above can be used to make online identification of predicting models.

3.2. Adaptive Parameter Optimization

When the structure of fuzzy neural network is determined, we will take the parameters acquired at self-organization phase as initial value of supervised learning of the network. Since it is based on the same training set, we adopt the improved error back propagation algorithm to adjust the parameters to be optimal. Then the final model of self-organization of FNN can be determined.

We choose any input sample x_p from set S and the output of network is

$$y = \frac{\sum_{k=1}^{L} \mu_k(x_p) w_k}{\sum_{k=1}^{L} \mu_k(x_p)} \tag{9}$$

The error function is

$$E = (y - Y^2) / 2 \tag{10}$$

In above equations, y denotes the actual output of P_{th} sample, Y denotes the expected output. This paper introduces an adaptive learning efficiency adjusting BP algorithm as the learning algorithm of FNN. Its process is: when the directions of two iterations of gradient are the same, it means the descent is too slow, so the step can be doubled; when the directions of two iterations of gradient are different, it means the descent is fast, so the step is halved.

In our FNN model, there are four types of parameters to be identified: mean value C_{ij} of Gauss function and width of δ_{ij}; and the weight value of the third layer w_{kj}

Then the derivations of these parameters are:

$$\begin{aligned} c_{ij}(n+1) - c_{ij}(n) &= -\eta(n) \frac{\partial E}{\partial c_{ij}(n)} = -\eta(n) \frac{\partial E}{\partial y} \frac{\partial y}{\partial c_{ij}(n)} = -\eta(n)(y - Y) \frac{\partial y}{\partial c_{ij}(n)} \\ &= -\eta(n)(y - Y)[\alpha_j w_j(n) \prod_{k=1, k \ne j}^{\mu} w_{1k} O_k^2 \exp[-\sum_{i=1}^{r} \frac{(x_i - c_{ij}(n))^2}{\delta_{ij}^2(n)}] \frac{2(x_i - c_{ij}(n))}{\delta_{ij}^2(n)} \\ &+ (1 - \alpha_j) w_j(n) w_{2j} \exp[-\sum_{i=1}^{r} \frac{(x_i - c_{ij}(n))^2}{\delta_{ij}^2(n)}] \frac{2(x_i - c_{ij}(n))}{\delta_{ij}^2(n)} \end{aligned} \tag{11}$$

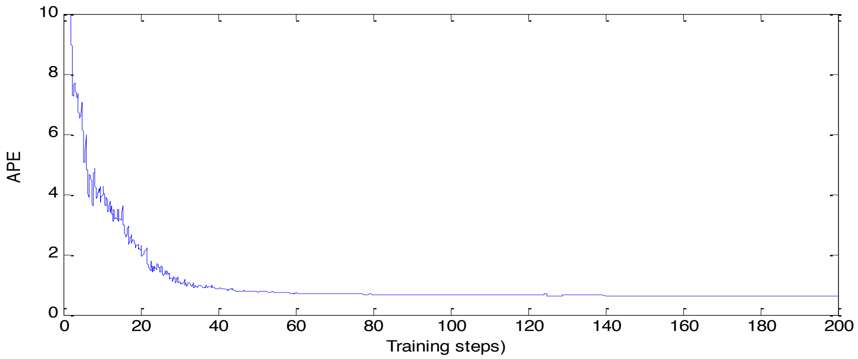


Fig. (2). APE error curve.

Table 1. Performance comparison of training models.

Algorithm	APE$_{tr}$	APE$_{te}$	Number of Parameters	Training Data Scale	Testing Data Scale
FNN	4.69	5.2	--	20	20
OLS	2.44	2.55	66	206	124
GSOFNN	2.01	1.53	64	206	124
SOFNN	0.57	2.24	35	206	124

$$\delta_{ij}(n+1)-\delta_{ij}(n)=-\eta(n)\frac{\partial E}{\partial\delta_{ij}(n)}=-\eta(n)\frac{\partial E}{\partial y}\frac{\partial y}{\partial\delta_{ij}(n)}=-\eta(n)(y-Y)\frac{\partial y}{\partial\delta_{ij}(n)}$$
$$=-\eta(n)(y-Y)[\alpha_j w_j(n)\prod_{k=1,k\neq j}^{\mu} w_{1k}O_k^2\exp[-\sum_{i=1}^{r}\frac{(x_i-c_{ij}(n))^2}{\delta_{ij}^2(n)}]\frac{2(x_i-c_{ij}(n))}{\delta_{ij}^2(n)}$$
$$+(1-\alpha_j)w_j(n)w_{2j}\exp[-\sum_{i=1}^{r}\frac{(x_i-c_{ij}(n))^2}{\delta_{ij}^2(n)}]\frac{2(x_i-c_{ij}(n))}{\delta_{ij}^2(n)} \tag{12}$$

$$w_{kj}(n+1)-w_{kj}(n)=-\eta(n)\frac{\partial E}{\partial w_{kj}(n)}=-\eta(n)\frac{\partial E}{\partial y}\frac{\partial y}{\partial w_{kj}(n)}=-\eta(n)\frac{\partial E}{\partial y}(y-Y)\frac{\partial y}{\partial w_{kj}(n)} \tag{13}$$
$$=\begin{cases}-\eta(n)(y-Y)w_j(n)\alpha_j O_{\Pi}^2/w_{1j}(n) & k=1\\ -\eta(n)(y-Y)w_j(n)(1-\alpha_j)O_j^2 & k=2\end{cases}$$

The second layer completes the membership function and synthesizes the fuzzy membership. At the fourth layer, we compute the network output with centralized defusing method. From equation 11, 12 and 13 we can see such adjustment is conducive for initialization and optimization of the network. Then the adaptive parameters for training are summarized as:

$$\begin{cases} c_{ij}(n+1)-c_{ij}(n)=-\eta(n)\dfrac{\partial E}{\partial c_{ij}(n)}, \\ \eta(n)=2^{\lambda}\eta(n-1), \lambda=sign[\dfrac{\partial E}{\partial c_{ij}(n)}\dfrac{\partial E}{\partial c_{ij}(n-1)}] \\ \delta_{ij}(n+1)-\delta_{ij}(n)=-\eta(n)\dfrac{\partial E}{\partial\delta_{ij}(n)}, \\ \eta(n)=2^{\lambda}\eta(n-1), \lambda=sign[\dfrac{\partial E}{\partial\delta_{ij}(n)}\dfrac{\partial E}{\partial\delta_{ij}(n-1)}] \\ w_{kj}(n+1)-w_{kj}(n)=-\eta(n)\dfrac{\partial E}{\partial w_{kj}(n)}, \\ \eta(n)=2^{\lambda}\eta(n-1), \lambda=sign[\dfrac{\partial E}{\partial w_{kj}(n)}\dfrac{\partial E}{\partial w_{kj}(n-1)}] \end{cases} \tag{14}$$

4. EXPERIMENTAL RESULTS

4.1. Simulations

To verify the effectiveness of SOFNN algorithm proposed in this paper. We provide a simulation of static function approximation to test its function in system identification. For convenient contrast we choose a case of classic three-input nonlinear function approximation [20]:

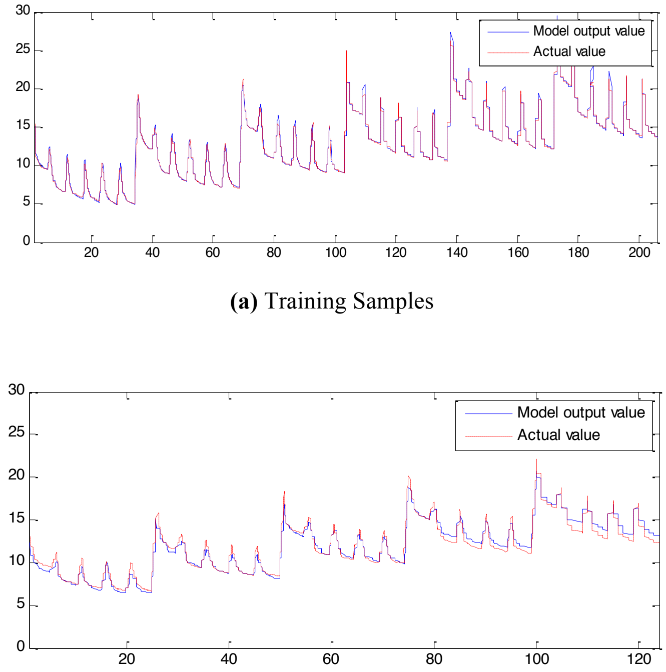
$$f=(1+x^{0.5}+y^{-1}+z^{-1.5})^2 \tag{15}$$

There are 206 training data and 124 test data from the input space $[1,6]\times[1,6]\times[1,6]$ and $[1.5,5.5]\times[1.5,5.5]\times[1.5,5.5]$, and they are sampled averagely. After the learning of SOFNN, the fuzzy neural network includes 5 fuzzy rules. At the optimal learning phase of network, we adopt the average percentage error [21] as performance indicator:

$$APE=\frac{1}{P}\sum_{p=1}^{P}\frac{|y-Y|}{|Y+|} \tag{16}$$

P is the number of data pairs. Fig. (**2**) depicts the training sample error curve after 200 steps of training. The actual value and model output value of training samples and test samples are shown in Fig. (**3**). It can be seen that the trained network acquires higher accuracy and generalization ability.

Table **1** list the test results of improved SOFNN model and other early results. From it we can see, with the same data scale, the parameters needed to be identified of SOFNN are obviously less than that of other similar models [22, 23]. It also can acquire higher training accuracy with the same training steps. "--" denotes the result is not lists in original



(a) Training Samples

(b) Test Samples

Fig. (3). Comparison of sample actual value and model value.

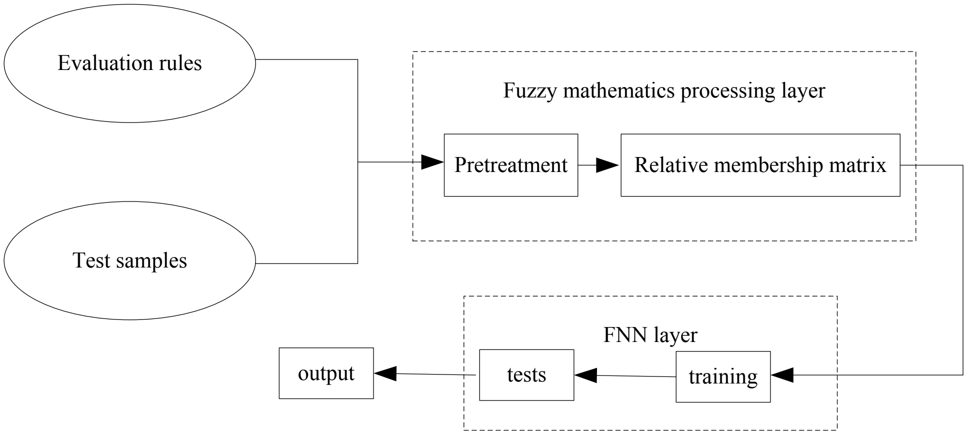


Fig. (4). Water quality evaluation process with SOFNN.

reference. APE_{tr} and APE_{te} denote the APE value of training data and test data.

4.2. Implementation on Water Quality Evaluation System

The evaluation process on water quality with SOFNN is shown as Fig. (**4**). Water quality evaluation is based on national water environment quality standard. The method proposed in this paper is used to construct learning samples and object output, regulate smoothing parameter on radial basis function network layer and construct trained network. The evaluation process is evaluating the supervised samples and learning samples to obtain the largest response output. Radial basis layer output is vectors to express the similarity degree between models. The competitive layer outputs learning mode with the largest probability by obtaining contributive rate of measured mode on each learning mode.

In data pretreatment, since the measured water quality parameters for water quality analysis process has different units, it may lead to that various input and output variable sizes in network have large difference on numerical value. Direct use of original measurement data for calculation may result in information loss and unstable numerical value calculation. Thus, it needs standardizing procession on each variable, so that the means of all variables after treatment are 0 and their standard deviations are 1, which eliminates the effects brought by different dimensions and magnitude of various variables. Its mathematical expression is:

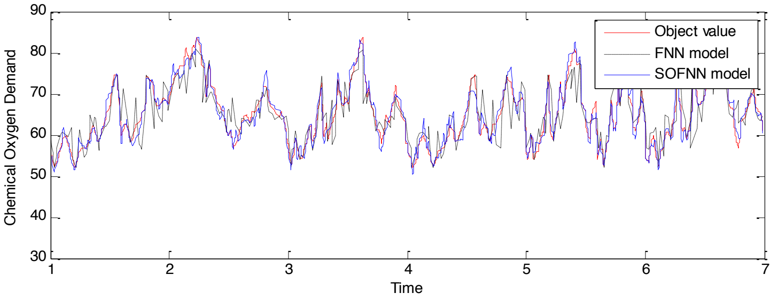


Fig. (5). Actual value of training samples and output of models.

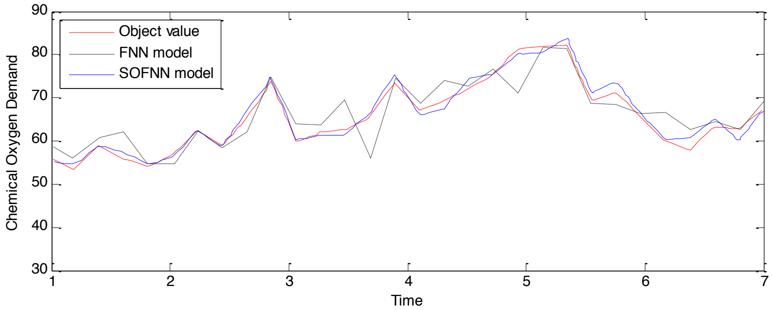


Fig. (6). Actual value of testing samples and output of models.

$$x(i,j)_{new} = \frac{x(i,j)_{old} - \overline{x}(i)}{S(i)}, \qquad (17)$$
$$i = 1,2,...,n,\ j = 1,2,...,p$$

$x(i,j)_{new}$ and $x(i,j)_{old}$ respectively denote the standardized data samples and original data samples. They represent the j_{th} sample of i_{th} variable. $\overline{x}(i)$ and $S(i)$ denote the mean and standard deviation of i_{th} variable. n and p denote the number of variables and samples.

The SOFNN model in this paper is used to establish water quality prediction model. It is compared to the prediction model established by traditional FNN. After the training of neural network, practical value of training sample, model output value, practical value of measured sample and output value of model are respectively shown in Fig. (**5**) and Fig. (**6**). From the curves and data shown in figures, we can see that the training accuracy and generalization ability of SOFNN model are superior to traditional FNN model. The water quality evaluation and prediction model established by SOFNN model in this paper effectively simulates the operation of water quality evaluation. That is, it has perfect generalization ability.

CONCLUSION

Based on the study of FNN and clustering algorithms, this paper proposes an improved nearest clustering method to identify the structure of FNN and adopts back propagation method to adjust the parameters. This kind of clustering algorithm can divide input data online, gradually update clusters and organize structure of fuzzy neural network by itself. We do not need to initialize the distribution of input data. All rules are generated by themselves, that is, with increasing of input data, the rules will automatically increase. So the generated FNN does not have contradictory rules. Our research proves that fuzzy neural network-based supervised clustering algorithm can be used to self-organize the structure learning of FNN. Latter simulations and applications have shown that our algorithm can effectively identify nonlinear dynamic system and it has effective generalization ability. So it has perfect theory developing space. However, the convergence, stability, robustness, etc, of this algorithm should be strictly proved in mathematical theory. So it needs further research on improving the training speed and stability.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] W. Pedrycz, M. Reformat, and C.W. Han, "Cascade architectures of fuzzy neural networks", *Fuzzy Optimization and Decision Making*, vol.12, no.1, pp.78-85, 2004.

[2] W. Xie, and Y. Qian, "A review of fuzzy neural networks", *Shenzhen University Journal*, vol.16, no.2, pp.22-28, 1999.

[3] G. Leng, G. Prasad, and T. M. McGinnity, "An on-line algorithm for creating self-organizing fuzzy neural networks", *Neural Networks*, vol.17, no.10, pp.1477-1493, 2004.

[4] D. Coyle, T. M. McGinnity, and G. Prasad, "A multi-class brain-computer interface with SOFNN-based prediction preprocessing", In: *Proceedings of the International Joint Conference on Neural Networks*, Hong Kong, China, 2008, pp.3696-3703.

[5] G. Leng, T. M. McGinnity, and G. Prasad, "Design for self-organizing fuzzy neural networks based on genetic algorithms", *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 755-765, 2006.

[6] X. L. Xu, Y. J. Lei, and W. B. Xie, "Self-organising intuitionistic fuzzy neural networks based on UKF", *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 38, no. 3, pp. 638-645, 2010.

[7] Z. Chen, M. Sun, and Z. Yuan, "PI adaptive controller based on neuro-fuzzy networks", *Journal of East China University of Science and Technology*, vol. 28, no. s1, pp. 45-48, 2002.

[8] D. Coyle, G. Prasad, and T.M. McGinnity, "Faster self-organizing fuzzy neural network training and a hyperparameter analysis for a brain-computer interface", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1458-1471, 2009.

[9] D. Coyle, G. Prasad, and M. McGinnity, "A self-organising fuzzy neural network with locally recurrent self-adaptive synapses", In: *Proceedings of IEEE Workshop on Evolving and Adaptive Intelligent Systems,* Paris, France, 2011, pp. 1-8.

[10] C. H. Kao, C. F. Hsu, and H. Don, "Design of an adaptive self-organizing fuzzy neural network controller for uncertain nonlinear chaotic systems", *Neural Computing and Applications*, vol. 21, no. 6, pp. 1243-1253, 2012.

[11] A. Akhbardeh, and A. Värri, "Novel supervised fuzzy adaptive resonance theory (SF-ART) neural network for pattern recognition", In: *Proceesings of IEEE International Workshop on Intelligent Signal Processing,* Faro, Portugal*,* 2005, pp. 149-154.

[12] J.G. Gu, W. Zhang, Z. W. Wang, "Method of machine learning based on partitioned clustering and fuzzy neural network", *Journal of System Simulation*, vol. 29, no. 23, pp. 5581-5586, 2007.

[13] R. J. Oentaryo, M. Pasquier, and C. Quek, "GenSoFNN-Yager: A novel brain-inspired generic self-organizing neuro-fuzzy system realizing Yager inference", *Expert Systems with Applications*, vol. 35, no. 4, pp. 1825-1840, 2008.

[14] R. Veroneze, A. R. Gonçalves, and F. J. Von Zuben, "A multi-objective analysis of adaptive clustering algorithms for the definition of RBF neural network centers in regression problems", *Lecture Notes in Computer Science*, vol. 74, no. 35, pp. 127-134, 2012.

[15] C. De, and A.T. F. De, B.N. Barbosa Gibson, "Partitioning fuzzy C-means clustering algorithms for interval-valued data based on city-block distances", *Proceedings of Brazilian Conference on Intelligent Systems*, Fortaleza, Ceara, Brazil, pp.113-118, 2013.

[16] X. Rui, and W. Yun, "Application of self-adaptive fuzzy neural networks in wastewater treatment", *Modern Electronic Technique*, vol. 12, no. 4, pp. 58-61, 2012.

[17] G. Leng, G. Prasad, and T.M. McGinnity, "A new approach to generate a self-organizing fuzzy neural network model", In: *Proceedings of the IEEE International Conference on Systems*, Academia Sinica, 2002, pp. 348-353,.

[18] Y.F. Peng, and C. F. Hsu, "Identification-based chaos control *via* backstepping design using self-organizing fuzzy neural networks", *Chaos, Solitons and Fractals*, vol. 41, no. 3, pp. 1377-1389, 2009.

[19] D. Lin, and X. Wang, "Self-organizing adaptive fuzzy neural control for the synchronization of uncertain chaotic systems with random-varying parameters", *Neurocomputing*, vol. 74, no. 12, pp. 2241-2249, 2011.

[20] J. Li, and Y. Zhang, "Nonlinear function approximation based on BP neural net and implementation on SAS", *Modern Computer*, vol. 8, no. 3, pp. 7-11, 2013.

[21] Y. Zhang, and Z. Ren, "SVM-based forecasting methods for software fault", *Journal of Chinese Computer Systems*, vol. 31, no. 7, pp. 1380-1384, 2010.

[22] N. Wang, E. M. Joo, and X. Meng, "A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks", *Neurocomputing*, vol. 72, no. 16, pp. 3818-3829, 2009.

[23] C. F. Hsu, "Self-organizing adaptive fuzzy neural control for a class of nonlinear systems", *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 1232-1241, 2007.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014