Applied Mathematics & Information Sciences
*An International Journal*

http://dx.doi.org/10.12785/amis/080334

# An Approach for Fuzzy Modeling based on Self-Organizing Feature Maps Neural Network

*Ching-Yi Chen*[1,*], *Jen-Shiun Chiang*[2], *Kuang-Yuan Chen*[2], *Ta-Kang Liu*[2] *and Ching-Chang Wong*[2]

[1] Department of Information and Telecommunications Engineering, Ming Chuan University, Taoyuan County 333, Taiwan
[2] Department of Electrical Engineering, Tamkang University, New Taipei City, 251, Taiwan

Received: 5 Jun. 2013, Revised: 10 Oct. 2013, Accepted: 11 Oct. 2013
Published online: 1 May. 2014

**Abstract:** Exploration of large and high-dimensional data sets is one of the main problems in data analysis. Self-organizing feature maps (SOFM) is a powerful technique for clustering analysis and data mining. Competitive learning in the SOFM training process focuses on finding a neuron that its weight vector is most similar to that of an input vector. SOFM can be used to map large data sets to a simpler, usually one or two-dimensional topological structure. In this paper, we present a new approach to acquisition of initial fuzzy rules using SOFM learning algorithm, not only for its vector feature, but also for its topological. In general, fuzzy modeling requires two stages: structure identification and parameter learning. First, the algorithm partitions the input space into some local regions by using SOFM, then it determines the decision boundaries for local input regions, and finally, based on the decision boundaries, it learns the fuzzy rule for each local region by recursive least squares algorithm. The simulation results show that the proposed method can provide good model structure for fuzzy modeling and has high computing efficiency.

**Keywords:** Competitive learning, SOFM, Fuzzy modeling

## 1 Introduction

When fuzzy inference system is used for system modeling, more number of rules used will theoretically help to construct more complicated system, but in the mean time, it will also increase the computing loading. Therefore, when we are creating fuzzy system, we have to induce first the input space with similar output value and describe it with fuzzy set, by doing so, complicated system can be simplified through the description of the input space, that is, through the setup of several fuzzy rules, this complicated system can be described, however, how to induce the input space with similar output characteristic is one of the important topics [1,2,3,4].

Data analysis is an algorithm using statistics and machine learning. It finds out through inspiration way the hidden knowledge and rule from massive data to generate the meaningful clustering result. Competitive learning is in its nature a Self-organization learning method. It can find out similar feature, rule or relation from unlabelled patterns, and then these patterns with common features are assigned into the same group [5,6]. Generally speaking, under the condition of lacking expert's experience, the trial of using merely input-output data to setup fuzzy system is usually through two stages of design procedures of structure identification and parameter identification; take the fuzzy modeling design as example, the space gathered by similar type of samples can form a fuzzy block and be mapped to a fuzzy rule, hence, the automatic data classification characteristic of Self-organization learning method is very suitable to be used in the design job of structure identification of fuzzy system.

In this article, competitive learning neural network is going to be introduced for fuzzy rule extraction. Through the output characteristic of Self-organizing Feature Maps Neural Network (SOFM) [7], the inference rule of the input data can be obtained, and the related parameter value of membership function of fuzzy rule can be obtained, finally, the structural identification of fuzzy inference system can be completed.

* Corresponding author e-mail: chingyi@mail.mcu.edu.tw

## 2 Using the Topological Network of SOFM to Realize the Structural Identification of Fuzzy System

SOFM is a neural network based on "competitive learning", that is, the neurons of output layer will compete with each other so as to get activated opportunity [7,8]. Generally, in competitive learning neural network, the winner will be selected from competitive phase, and the weight vector of the winner will be adjusted in the reward phase, which is as shown in equation (1)–(2). In Fig. 1, we have used geometrical method to describe the weight adjustment behavior.

$$W_j(t+1) = W_j(t) + \eta(x_i - W_j), if\, j = j^*, \quad (1)$$

Where

$$j^* = \arg\min_j \| x_i - W_j \|, j = 1,2,\ldots,\Phi \quad (2)$$

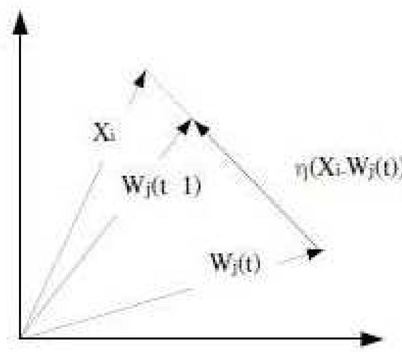


**Fig. 1:** Weight vector adjustment of the winning neuron.

However, the difference between SOFM and general competitive learning neural network is that in the competitive method, co-learning is adopted between the winning neurons and neurons which are the neighborhoods, but for general competitive learning neural network, it adopts "winner-take-it-all", which has architecture as in Fig. 2. In other words, in SOFM, after the competition, not only the winning neurons have the chance to learn, the neurons which are the neighborhoods can also have chance to learn. Fig. 3 is two-dimensional lattice formed by SOFM output neurons, wherein the neighborhood relation between neuron and adjacent neurons is usually described by rectangular or hexagonal topology. Each neuron represents a set of n dimensional weight vector, and n is the dimension of the input data vector. The neighborhood range of SOFM is set up at maximum in the beginning, and it even includes all the neurons, then as the time passes by, the neighborhood area range is gradually reduced. Fig. 4 explains the situation that SOFM winning neuron and neighborhood neuron learns from the input vector.

SOFM, through feature map way and through nonlinear projection method, converts input vector of any dimension into the low dimension matrix space formed by neurons, the neurons of the output layer then follow the current input vector to compete with each other so as to get the chance to adjust the weight vector, finally, based on the feature of the input vector, meaningful topological structure is formed to be displayed in the output space.

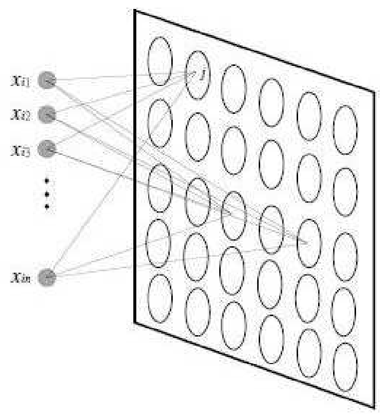


**Fig. 2:** SOFM architecture

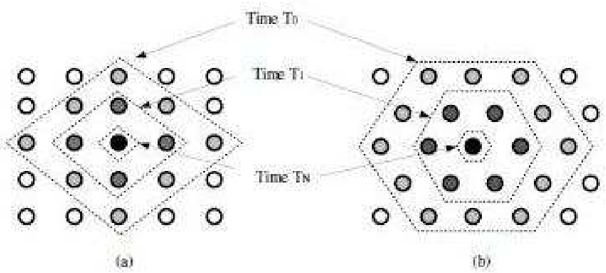


**Fig. 3:** Neighborhood function. (a) Rectangular topology; (b) Hexagonal topology.

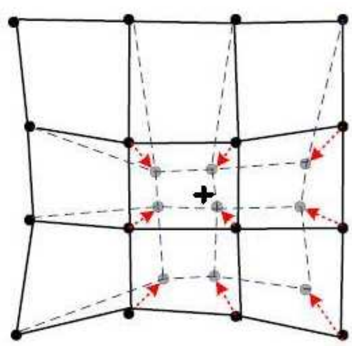

**Fig. 4:** The phenomenon of neighborhood learning of neuron of output layer.

In order to meet better the biological view point, the so-called improved type SOFM usually uses Gaussian function to decide the strength that SOFM neighborhood function is activated [8]:

$$h_{j,j^*} = \exp\left(-\frac{d_{j,j^*}^2}{2\sigma^2}\right) \tag{3}$$

Here $d_{j,j^*}$ represents the side linking distance between $j$th neuron and winning neuron $j^*$. In addition, the effective width $\sigma(t)$ and learning parameter $\eta(t)$ of the neighborhood function is set up respectively as:

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_1}\right) \tag{4}$$

and

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_2}\right) \tag{5}$$

Here the constant $\sigma_0$ and learning parameter $\eta_0$ is set up by the initial value, and $\tau_1$ and $\tau_2$ are constants. Therefore, we can induce the improved SOFM algorithm as in the followings:

1. *Initialize weights to some small, random values*
2. *Repeat until convergence*
   (a) *Select the next input vector $x_i$ from the data set*
      i. *Find the unit $W_{j^*}$ that best matches the input vector $x_i$*

$$\| x_i - W_{j^*} \| = \min_j \| x_i - W_j \|, j = 1, 2, \ldots, \Phi$$

      ii. *Update the weights of the winner $W_{j*}$ and all its neighbors $W_k$*

$$W_k = W_k + \eta(t) \cdot h_{j^*k}(t) \cdot (x_i - W_k)$$

   (b) *Decrease the learning rate $\eta$ $(t)$*
   (c) *Decrease neighborhood size $\sigma$ $(t)$*

In order to describe the space mapping characteristic of SOFM and its capability to be applied in restructuring system that is to be identified, we have taken first one-dimensional SOFM to perform verification. First, for the Gaussian function defined in the range from $-2$ to 2 as defined in equation (6) is generated with enough two dimensional data set (Fig. 5(a)), meanwhile, network structures with respectively 14, 10 and 8 output neurons are created to perform the learning, and the obtained topological structure is as shown in Fig. 5(b)–(d). Not only the output of neurons performs pretty in good order, but also they can strongly reflect the input space as distributed by the input vectors.

$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \sigma = 1.2 \tag{6}$$

If we expand the above problems into $[-2,2] \times [-2,2]$ 3D curved surface, as shown in equation (7) and Fig. 6(a), the trained and resulted two dimensional SOFM output is then as shown in Fig. 6(b)–(d). From the characteristic of the output neurons, it can be found that the allocation of the weights of neurons can indeed reflect the topological structure of the input vectors.

$$z = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2+y^2}{2\sigma^2}}, \sigma = 1.2 \tag{7}$$

From the above example, the SOFM neighborhood learning concept can find out topology network that is enough to represent the data clustering characteristic. Meanwhile, we can also use SOFM output triangular or square grid to describe the curved surface as formed by 3D data set in the example. Therefore, this paper is going to be based on the output network topological structure of SOFM to describe the space distribution characteristics of the input data set to be expressed. Meanwhile, it will be used to complete the design procedure of the structure identification of fuzzy system.

## 3 The Parameter Identification of Fuzzy System

This research has adopted the learning rules as proposed by Wong and Chen to perform the parameter fine tuning of the consequent part of fuzzy system [1,9]. So that the setup system can more accurately approach the behavior of the system to be identified. For N input-output data of system to be identified $x_i, i = 1, 2, \ldots, N$, wherein $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ is n dimensional data point, $(x_{i1}, x_{i2}, \ldots, x_{i(n-1)})$ is the input of $i$th input-output data, and $x_{in}$ is its corresponding output. If our obtained SOFM output is $l \times s$ neuron matrix, that is, $\Phi = l \times s$, and the weight vector is defined as $W_r = (W_{r1}, W_{r2}, \ldots, W_{rn})$, $r = 1, 2, \ldots, \Phi$, then each neuron will be mapped to a fuzzy rule:

*Rule m : If x is $S_r$ then*

$$y_r = C_{r0} + \sum_{j=1}^{n-1} c_{rj} x_j, r = 1, 2, \ldots, \Phi \tag{8}$$

where

$$S_r(x) = \exp\left(-\frac{\sum_{k=1}^{n-1}(x_k - W_{rk})^2}{2\omega_r^2}\right),$$

and

$$\omega_r = \sqrt{\frac{-\sum_{k=1}^{n-1}(x_{rk}^* - W_{rk})^2}{2\ln(\alpha)}} \tag{9}$$

where $\alpha$ is adjustable parameter and we adopt $\alpha = 0.2$ in this paper. $x_r^* = (x_{r1}^*, x_{r2}^*, \ldots, x_{r(n-1)}^*)$ is classified into the data cluster formed by $r$th neuron, and it is the input variable of the farthest data of the weight vector $W_r$ of $r$th

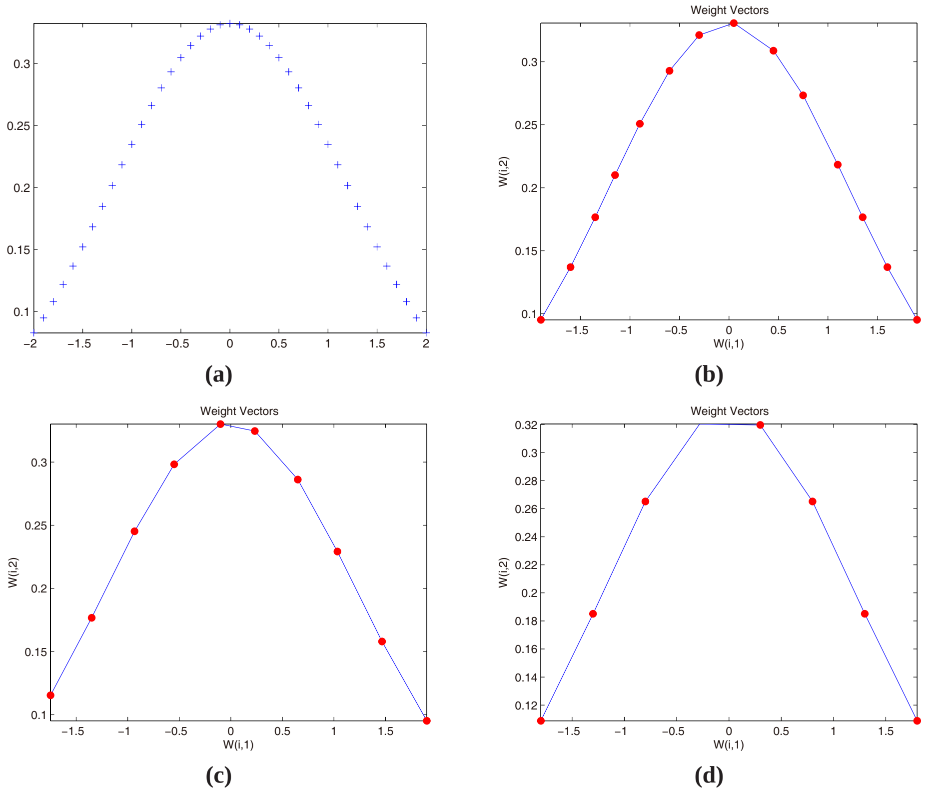


**Fig. 5:** (a) 2D Gaussian distribution data set, (b) [1×14] one dimensional SOFM output, (c) [1×10] one dimensional SOFM output, (d) [1×8] one dimensional SOFM output.

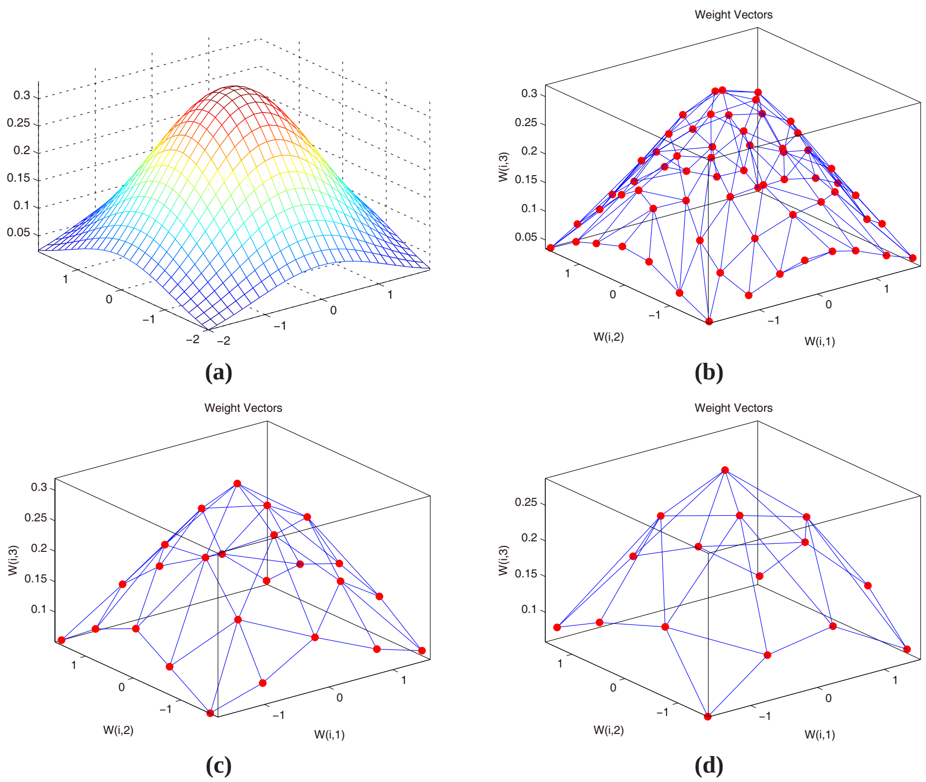


**Fig. 6:** (a) 3D Gaussian distribution data set, (b) [8×8] two dimensional SOFM output, (c) [5×5] two dimensional SOFM output, (d) [4×4] two dimensional SOFM output.

**Table 1:** The parameter values obtained by the proposed method ($l = 5$, $s = 6$).

| $\gamma$ | $W_{\gamma 1}$ | $W_{\gamma 2}$ | $W_{\gamma 3}$ | $\omega_\gamma$ | $c_{\gamma 0}$ | $c_{\gamma 1}$ | $c_{\gamma 2}$ |
|---|---|---|---|---|---|---|---|
| 1 | 7.3786 | −7.4501 | −0.0003 | 2.4207 | 0.1664 | −0.0092 | 0.0092 |
| 2 | 7.1345 | −5.1170 | 0.0026 | 1.6572 | −0.2760 | 0.0070 | −0.0371 |
| 3 | 7.2720 | −1.1944 | −0.0028 | 1.8752 | 0.6762 | −0.0746 | −0.0066 |
| 4 | 7.5379 | 2.3483 | −0.0025 | 1.9274 | 0.5722 | −0.0522 | −0.0639 |
| 5 | 7.6122 | 5.9236 | 0.0032 | 2.1860 | 0.1198 | 0.0249 | −0.0585 |
| 6 | 4.9346 | −6.6654 | −0.0070 | 1.7831 | −0.1514 | −0.0253 | −0.0275 |
| 7 | 4.9965 | −3.0878 | 0.0298 | 1.1764 | −0.6287 | −0.0318 | −0.2149 |
| 8 | 5.1362 | 0.5952 | 0.0393 | 1.2050 | −0.6123 | 0.0265 | 0.0282 |
| 9 | 5.4995 | 5.0548 | 0.0005 | 1.3385 | 0.4731 | −0.1207 | 0.0528 |
| 10 | 5.9073 | 7.4731 | 0.0014 | 2.0687 | −0.2726 | 0.0494 | 0.0063 |
| 11 | 1.4425 | −7.5176 | 0.0019 | 1.2812 | 1.6332 | −0.1064 | 0.1451 |
| 12 | 1.3970 | −5.2671 | 0.0421 | 1.8816 | −0.2929 | 0.1880 | 0.0780 |
| 13 | 1.5541 | −0.9431 | 0.1589 | 1.1779 | 1.0617 | −0.2932 | 0.0748 |
| 14 | 1.8630 | 2.9271 | 0.1075 | 1.2185 | 0.4434 | −0.0107 | −0.1325 |
| 15 | 2.3774 | 6.6717 | 0.0057 | 1.8866 | −0.3284 | 0.0017 | 0.0428 |
| 16 | −1.6263 | −6.7174 | 0.0065 | 1.9055 | 0.7891 | 0.0597 | 0.0674 |
| 17 | −1.4974 | −3.0941 | 0.1140 | 1.2229 | 0.6965 | 0.0758 | 0.1756 |
| 18 | −1.3024 | 0.7619 | 0.1648 | 1.2988 | 1.0306 | 0.2508 | −0.0996 |
| 19 | −0.9974 | 5.0669 | 0.0493 | 1.1800 | −0.5084 | −0.1767 | −0.0298 |
| 20 | −0.9008 | 7.4299 | 0.0021 | 1.8987 | 1.3193 | −0.0038 | −0.1371 |
| 21 | −5.2360 | −7.6152 | 0.0022 | 1.8756 | −0.3032 | 0.0124 | −0.0379 |
| 22 | −5.1646 | −5.3972 | 0.0011 | 1.1262 | 0.3268 | 0.0017 | 0.0449 |
| 23 | −4.9564 | −0.9947 | 0.0534 | 1.1810 | −0.5338 | −0.0007 | −0.1344 |
| 24 | −4.7788 | 2.6879 | 0.0384 | 1.2264 | −0.0625 | 0.1497 | 0.2452 |
| 25 | −4.6880 | 6.3980 | −0.0086 | 2.3625 | −0.7773 | −0.0428 | 0.0640 |
| 26 | −7.3559 | −6.5546 | 0.0023 | 2.0384 | 0.3443 | 0.0172 | 0.0180 |
| 27 | −7.4486 | −3.2503 | −0.0009 | 1.7594 | 0.0368 | −0.0093 | 0.0504 |
| 28 | −7.1316 | 0.6558 | −0.0019 | 1.8559 | 0.9353 | 0.1017 | 0.0078 |
| 29 | −6.9705 | 4.7708 | 0.0014 | 1.5911 | 0.5222 | 0.0716 | −0.0047 |
| 30 | −7.2252 | 7.2283 | 0.0001 | 2.6332 | 0.2155 | −0.0017 | −0.0283 |

neuron. In order to let fuzzy system approach the system to be identified, the least square error function is set up as in Refs. [1]:

$$E = \sum_{i=1}^{N} (y_i - y_i^d)^2 \tag{10}$$

Through the input output behavior of continuous learning of recursive least-squares algorithm, we can then obtain the final parameter set [1,9,10,11,12,13].

## 4 Simulation Results

In order to describe the effectiveness of this method, we have used a two dimensional input and single output nonlinear system to perform the experiment [14,15]:

$$y = \mathrm{sin}\,c(x_1,x_2) = \frac{\sin(x_1)\sin(x_2)}{x_1 x_2} \tag{11}$$

Where $x_1$ and $x_2$ are input variables of $[-10,10] \times [-10,10]$, $y$ is the output of nonlinear system. We have done even grid partition on the input space so as to generate $30 \times 30 = 900$ input-output training data. Fig. 7 shows the input-output data of nonlinear system, and

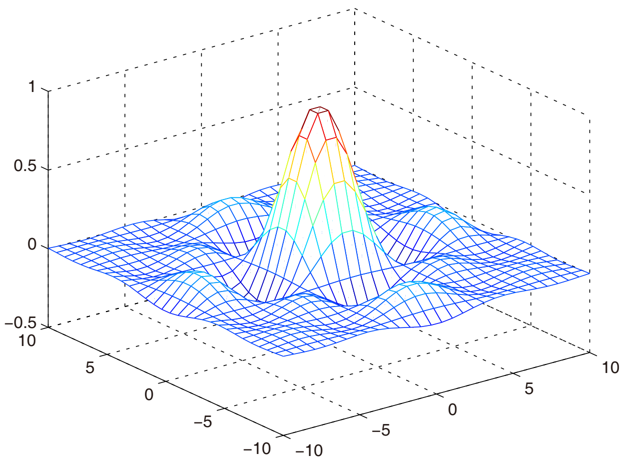


**Fig. 7:** The input output data of nonlinear system.

Fig. 8(a) is the [5×6] two dimensional topological network of the output of SOFM, meanwhile, triangular lattice is used to display the curved surface characteristic as formed by the original data point. Fig. 8(b) is the obtained identified result by using SOFM output neurons, and finally, the fuzzy inference output result and error

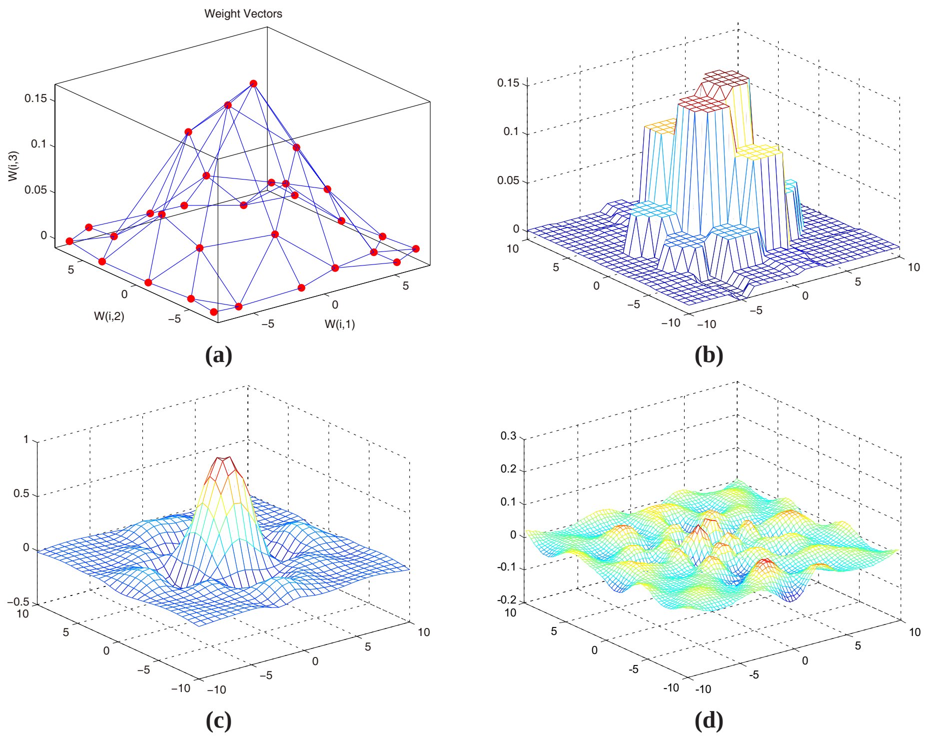


**Fig. 8:** $Sinc(x_1,x_2)$ nonlinear system identification ($l = 5$, $s = 6$). (a) SOFM output ($\Phi = [5\times6]$), (b) nonlinear system structural identification, (c) fuzzy inference system output result, (d) error curved surface drawing ($MSE = 2.9658\times10^{-4}$).

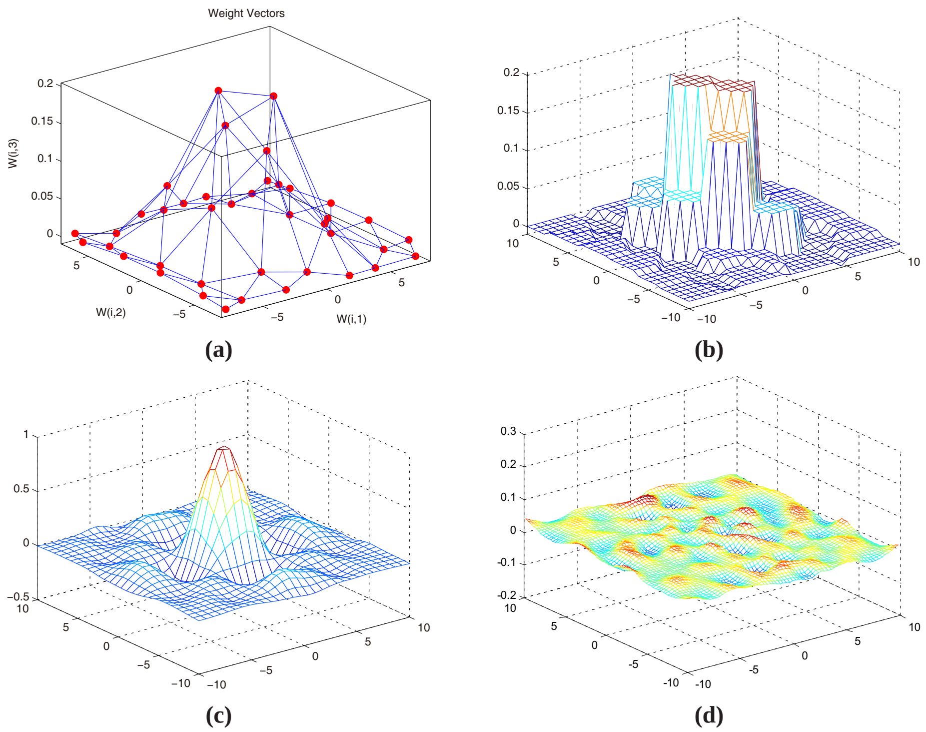


**Fig. 9:** $Sinc(x_1,x_2)$ nonlinear system identification ($l = 5$, $s = 8$). (a) SOFM output ($\Phi = [5\times8]$), (b) nonlinear system structural identification, (c) fuzzy inference system output result, (d) error curved surface drawing ($MSE = 7.8906\times10^{-5}$).

**Table 2:** The parameter values obtained by the proposed method ($l = 5$, $s = 8$).

| $\gamma$ | $W_{\gamma 1}$ | $W_{\gamma 2}$ | $W_{\gamma 3}$ | $\omega_\gamma$ | $c_{\gamma 0}$ | $c_{\gamma 1}$ | $c_{\gamma 2}$ |
|---|---|---|---|---|---|---|---|
| 1 | 8.7656 | 3.1537 | 0.0070 | 2.4531 | −0.4129 | 0.0317 | 0.0457 |
| 2 | 8.2620 | 1.4023 | 0.0124 | 1.7272 | −1.0768 | 0.0693 | 0.0818 |
| 3 | 7.7524 | −1.8153 | 0.0075 | 1.4379 | 0.2101 | 0.0075 | 0.0917 |
| 4 | 7.6892 | −5.0186 | 0.0003 | 1.3828 | −0.2679 | 0.0410 | 0.0221 |
| 5 | 7.3252 | −7.4728 | 0.0000 | 1.5157 | 0.2228 | −0.0120 | 0.0120 |
| 6 | 7.6228 | 4.9249 | 0.0020 | 2.1072 | 0.1705 | −0.0327 | −0.0032 |
| 7 | 6.5691 | 0.9464 | −0.0044 | 1.5240 | −2.3006 | 0.4189 | 0.0342 |
| 8 | 5.8014 | −2.5813 | −0.0127 | 1.2814 | −0.7665 | 0.0987 | −0.0466 |
| 9 | 5.4719 | −6.2355 | −0.0021 | 0.9053 | −0.1240 | 0.0412 | 0.0157 |
| 10 | 5.2496 | −8.0686 | 0.0036 | 1.0882 | −0.0873 | −0.0153 | −0.0154 |
| 11 | 6.7749 | 7.3667 | 0.0008 | 1.6410 | 0.1201 | −0.0014 | −0.0110 |
| 12 | 5.4821 | 4.8302 | 0.0036 | 1.0740 | 0.5953 | −0.0570 | −0.0463 |
| 13 | 3.6319 | 0.0682 | 0.0625 | 1.0564 | −0.3587 | 0.0231 | 0.0678 |
| 14 | 2.6786 | −3.7697 | 0.0477 | 1.8291 | −0.1544 | 0.1016 | 0.0302 |
| 15 | 2.0597 | −7.0063 | −0.0034 | 1.0252 | 0.3706 | −0.0581 | 0.0229 |
| 16 | 3.8855 | 6.6972 | 0.0029 | 1.0958 | 0.0132 | −0.0082 | −0.0057 |
| 17 | 2.2416 | 2.8730 | 0.1310 | 1.6826 | −0.3418 | 0.0836 | 0.0190 |
| 18 | 0.8790 | −1.1718 | 0.1955 | 1.6319 | 0.9675 | −0.1295 | 0.2325 |
| 19 | −0.2197 | −5.4034 | 0.0545 | 1.0813 | −0.5764 | −0.0029 | −0.0807 |
| 20 | −0.7438 | −7.6746 | 0.0022 | 0.9583 | 0.8455 | −0.0096 | 0.0883 |
| 21 | 0.9466 | 7.6728 | 0.0021 | 1.1217 | 0.7304 | −0.0011 | −0.0776 |
| 22 | 0.4033 | 5.4383 | 0.0514 | 1.1317 | −0.4430 | −0.0133 | 0.0517 |
| 23 | −0.8037 | 1.2271 | 0.1923 | 1.7685 | 1.4289 | 0.1902 | −0.3976 |
| 24 | −2.0816 | −2.8088 | 0.1347 | 1.2796 | −0.0375 | −0.1126 | 0.0968 |
| 25 | −3.5849 | −6.7538 | 0.0030 | 1.6802 | −0.1005 | −0.0202 | −0.0041 |
| 26 | −1.9720 | 7.0137 | −0.0037 | 1.0813 | 0.2598 | 0.0395 | −0.0181 |
| 27 | −2.6438 | 3.7589 | 0.0450 | 1.0429 | 0.0897 | −0.0471 | −0.0383 |
| 28 | −3.5833 | −0.1087 | 0.0605 | 1.8708 | 0.2625 | 0.3451 | 0.1037 |
| 29 | −5.1497 | −4.9113 | 0.0035 | 1.6093 | 0.3291 | 0.0022 | 0.0516 |
| 30 | −6.3399 | −7.5557 | 0.0012 | 1.0009 | −0.1974 | −0.0439 | 0.0107 |
| 31 | −5.2806 | 8.0180 | 0.0034 | 1.1321 | −0.0609 | 0.0106 | 0.0105 |
| 32 | −5.5174 | 6.1378 | −0.0020 | 0.9985 | 0.2349 | −0.0089 | −0.0423 |
| 33 | −5.8974 | 2.3402 | −0.0114 | 1.1294 | −0.0972 | 0.0402 | 0.0804 |
| 34 | −6.4751 | −1.2680 | −0.0041 | 2.3207 | 2.2468 | 0.2487 | 0.2039 |
| 35 | −7.3471 | −5.3383 | 0.0008 | 2.0511 | 0.1806 | −0.0018 | 0.0189 |
| 36 | −7.3893 | 7.2655 | 0.0002 | 1.5457 | 0.1747 | 0.0126 | −0.0065 |
| 37 | −7.7585 | 4.6430 | 0.0029 | 1.4439 | −0.4386 | −0.0417 | 0.0055 |
| 38 | −7.8204 | 1.2293 | 0.0106 | 1.6056 | 0.6083 | 0.0654 | −0.0270 |
| 39 | −8.2370 | −2.2271 | 0.0103 | 1.3740 | 0.4665 | 0.0059 | 0.0792 |
| 40 | −8.7829 | −4.3916 | −0.0001 | 1.3189 | −0.2967 | −0.0643 | 0.0413 |

curved surface figure is as shown in Fig. 8(c)–(d). The parameter values obtained by the SOFM and recursive least-squares method are listed in Table 1.

For the same example, if we change the output layer of SOFM into [5×8] matrix, the triangular network curved surface still can effectively display the distribution characteristic of the original data (Fig. 9(a)), and the system structure identification and fuzzy inference system output result is as shown in Fig. 9(b)–(c), and the final error curved surface is displayed in Fig. 9(d). The parameter values obtained by the SOFM and recursive least-squares method are listed in Table 2. Therefore, the SOFM output matrix size is enough to affect the approaching effect of curved surface reconstruction.

## 5 Conclusions

This paper has introduced a structural identification method for using competitive learning network to implement fuzzy system modeling procedure. Since SOFM can, through feature mapping method and nonlinear projection method, convert the high dimensional input vectors into low dimensional matrix space as formed by output neurons and follow the characteristic of the input vectors to form meaningful topological network, hence, if it is applied in structural identification, it can effectively explore the cluster distribution status of the input data set. In the paper, we have used the topological network sent out from SOFM to get the inference rule of the input data and to generate meaningful fuzzy rule to reconstruct the behavior of the system to be identified, and finally, the structural identification job of the fuzzy system is then completed. After getting the initial fuzzy system, we can then use the input-output data as the training samples to further fine-tune the parameter value of fuzzy inference system so that it can more accurately meet the behavior of the system to be identified. From the simulation result, it can be seen that the method proposed by this paper can indeed, with only known input-output data, effectively extract the fuzzy rules and parameters of the membership functions. The system modeling job can be achieved finally.

## References

[1] C. C. Chen , Design of Fuzzy Systems based on Partitioning Input Spaces. Ph. D. Thesis, University of Tamkang, (2000).
[2] M. Delgado, A. F. Gomez-Skarmeta, and F. Martin, A fuzzy clustering-based rapid prototyping for fuzzy rule-based modeling. IEEE Transactions Fuzzy Systems, **5**, 223-232 (1997).
[3] R. R. Yager and D. P. Filev, Approximate clustering via the mountain method. IEEE Transactions System, Man and Cybernetics, **24**, 1279-1284 (1994).
[4] R. R. Yager and D.P. Filev, Essentials of Fuzzy Modeling and Control. John Wiley, New York, (1994).
[5] S. Haykin, Neural Networks and Learning Machines, third edition. Prentice Hall, New Jersey, (2009).
[6] W. J. Hwang, F. J. Lin, S. C. Liao, and J. H. Huang, Efficient pipelined architecture for competitive learning. Journal of Parallel and Distributed Computing, **71**, 236-244 (2011).
[7] T. Kohonen, Self-Organization and Associative Memory, third edition. Springer-Verlag Berlin Heidelberg, New York, (1988).
[8] M. A. Kraaijveld, J. Mao, and A. K. Jain, A nonlinear projection method based on Kohonen's topology preserving maps. IEEE Transactions on Neural Network, **6**, 548-559 (1995).
[9] C. C. Wong and C. C. Chen. A Clustering-Based Method for Fuzzy Modeling. IEICE Transactions Information and System, **E82-D**, 1058-1065 (1999).
[10] C. C. Chen, A PSO-Based Method for Extracting Fuzzy Rules Directly from Numerical Data. Cybernetics and Systems: An International Journal, **37**, 707-723 (2006).
[11] L. X. Wang, A course in fuzzy systems and control. Prentice Hall, New Jersey, (1997).
[12] C. C. Wong and C. C. Chen, A GA-based method for constructing fuzzy systems directly from numerical data. IEEE Transactions on Systems, Man and Cybernetics, **30**, 905-911 (2000).
[13] J. S. Jang, C. T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, New Jersey, (1997).
[14] J. S. Jang, ANFIS: Adaptive-network-fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics, **23**, 665-685 (1993).
[15] J. S. Jang, C. T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing. Prentice Hall, New Jersey, (1997).

**Ching-Yi Chen** received his Ph.D. degree from Tamkang University, New Taipei City, Taiwan, in 2006. He joined the Department of Information and Telecommunications, Ming Chuan University in 2007 and is now an assistant professor. His main research interests include swarm intelligence, pattern recognition, and embedded systems.

**Jen-Shiun Chiang** received the B.S. degree in electronics engineering from Tamkang University, New Taipei City, Taiwan, in 1983, the M.S. degree in electrical engineering from University of Idaho, Moscow Idaho, USA, in 1988, and the Ph.D. degree in the electrical engineering from Texas A&M University, College Station Texas, USA, in 1992. He joined the faculty member of the Department of Electrical Engineering at Tamkang University in 1992 as an associate professor. Currently he is a professor at the department. Dr. Chiang's research interest includes digital signal processing for VLSI architecture, architecture for image data compressing, SOC design, analog to digital data conversion, and low power circuit design.

**Kuang-Yuan Chen** received the B.S. and MS degree in electrical engineering from Tamkang University, New Taipei City, Taiwan, in 1991 and 1997, respectively. He is currently a senior engineer at ITE Tech. Inc., Hsin-Chu, Taiwan, focused on flash storage controller implementation and application. His research interests include digital circuit design, testing, embedded system and fuzzy system.

**Ta-Kang Liu** received the B.S. and MS degree in electrical engineering from Tamkang University, New Taipei City, Taiwan, in 1991 and 1997, respectively. He joined the R&D Center, King Yuan Electronics Company, Hsin-Chu, Taiwan, in 1998 and focused on logic tester and burn-in system development. His current research interests include digital circuit design, testing, neural networks and fuzzy systems.

**Ching-Chang Wong** was born in Chiayi, Taiwan. He received the B.S. degree in electronic engineering from Tamkang University, New Taipei City, Taiwan, in 1984. He received the M.S. and Ph.D. degrees in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1986 and 1989, respectively. He is currently a professor with the Department of Electrical Engineering, Tamkang University. His research interests include fuzzy systems, grey systems, genetic algorithms, and intelligent control.