



Available online at www.sciencedirect.com



Decision Support Systems 42 (2006) 89–103

Decision Support
Systems

www.elsevier.com/locate/dsw

Part family formation through fuzzy ART2 neural network

R.J. Kuo ^{*}, Y.T. Su, C.Y. Chiu, Kai-Ying Chen, F.C. Tien

*Department of Industrial Engineering and Management, National Taipei University of Technology, No. 1, Section 3,
Chung-Hsiao East Road, Taipei, Taiwan 106, Province of China*

Received 5 February 2004; received in revised form 15 October 2004; accepted 18 October 2004

Available online 15 December 2004

Abstract

In order to overcome some unavoidable factors, like shift of the part, that influence the crisp neural networks' recognition, the present study is dedicated in developing a novel fuzzy neural network (FNN), which integrates both the fuzzy set theory and adaptive resonance theory 2 (ART2) neural network for grouping the parts into several families based on the image captured from the vision sensor. The proposed network posses the fuzzy inputs as well as the fuzzy weights. The model evaluation results showed that the proposed fuzzy neural network is able to provide more accurate results compared to the fuzzy self-organizing feature maps (SOM) neural network [R.J. Kuo, S.S. Chi, P.W. Teng, Generalized part family formation through fuzzy self-organizing feature map neural network, International Journal of Computers in Industrial Engineering, 40 (2001b) 79–100] and fuzzy c-means algorithm.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Group technology; Fuzzy set theory; Fuzzy neural network; ART2 neural network

1. Introduction

In flexible manufacturing system (FMS), group technology (GT) has been widely applied, since it allows design and manufacturing to take advantage of similarities between parts. A design engineer facing the task of developing a new part can use GT code or an image of the part to determine whether similar parts exist in a computer-aided design (CAD) database. Basically, the parts in the same family should

have both similar design features and similar manufacturing features. In addition, the parts in the same family usually need similar machining. Therefore, the implementation of GT could decrease the complexity of the design process and further shorten the design life cycle as well as the manufacturing life cycle, which is the basis of cell formation.

Recently, artificial neural networks (ANNs) have also been employed in GT since they have shown very promising results in areas of control and pattern recognition [36,37]. Among them, most have considered only crisp input data instead of fuzzy data. However, environmental conditions, like shift and noise, can always decrease the recognition accuracy if

* Corresponding author. Tel.: +886 2 27712171; fax: +886 2 27317168.

E-mail address: rjkuo@ntut.edu.tw (R.J. Kuo).

crisp data are used. Thus, fuzzy set theory, which has been successfully applied in pattern recognition and control [29], is utilized in combination with the ANNs. To an extent, the present study attempts to develop an intelligent GT system which consists of five components: (1) image acquisition, (2) image processing, (3) feature extraction, (4) pattern recognition, and (5) parts clustering, wherein the first, second, and third components intend to extract the fuzzy features from the captured image, while the fourth component clusters the parts with fuzzy features into several families. A fuzzy adaptive resonance theory 2 (ART2) neural network is proposed to solve the parts clustering problem in the fifth component. It is able to quickly and objectively group the parts. The network is based on the ART2 neural network [3]. However, two main differences are that the proposed network processes the fuzzy inputs as well as the fuzzy weights. The ART2 neural network can solve that the general clustering neural networks cannot cluster automatically and objectively (like self-organizing feature map [SOM] neural network).

The model evaluation results for applying the parts from Ref. [17] showed that the proposed fuzzy ART2 is better than fuzzy c-means and fuzzy SOM [25] as considering both the shift of the parts and noise. The remainder of this paper is organized as follows. Section 2 provides some necessary background information, while the proposed approach is presented in Section 3. Section 4 summarizes the evaluation results and discussion. Finally, the concluding remarks are made in Section 5.

2. Background

This section briefly reviews the applications of ANNs in GT. In addition, the fuzzy neural networks (FNNs) are also discussed.

2.1. Applications of artificial neural networks in GT

The basic idea of GT is to decompose the manufacturing system into various subsystems. This can decrease the machining time and increase the manufacturing flexibility. Several automated GT coding systems applying ANNs have been presented in the last few years [10]. Kaparthi and Suresh [19,20] have

applied the ANNs for classification and coding for rotational parts using a three-digit part description, whereas Liao and Lee [32] developed an automated GT coding and part family forming system that comprises an adaptive resonance theory (ART1) neural network and a feature-based CAD system. Awwal and Karim [1] applied a Hopfield neural network to recognize the shapes of part in the form of binary images. Four part shapes were used to train a neural network, and nine partial input shapes were provided to test the recognition capability of network. It was found that the tested shapes were identified correctly. In addition, Karmarthi et al. [17] have utilized a feed-forward neural network with back-propagation learning algorithm for the retrieval of the part data, while Chung and Kusiak [7] classified the machine parts based on their geometry using a feed-forward neural network with a back-propagation learning algorithm. Furthermore, Caudell et al. [4] have demonstrated the feasibility of training an ART1 neural network first to classify the cluster designs into families and then to recall the family when presented a similar design. This can dramatically decrease the design life cycle by avoiding duplication of the design efforts. Kuo et al. [25] integrated SOM neural network and fuzzy set theory to develop a fuzzy SOM system. They used this system to cluster GT parts through CDD images captured. For more references, the reader can refer to Chen and Cheng [6], Ham et al. [10], Harish and Gu [11], Kao and Moon [18], Lee et al. [30], Lee and Wang [31], and Moon and Chi [38] for more detailed information.

2.2. Fuzzy neural network

The ANNs [36] and fuzzy model [45,29] have been applied in many application areas, each pairing its own merits and disadvantages. Therefore, how to successfully combine these two approaches, ANNs and fuzzy modeling, has become a very potential research area.

Generally, the traditional fuzzy system mentioned above is based on experts' knowledge. However, it is not very objective. Besides, it is very difficult to acquire robust knowledge and find available human experts [15]. Recently, the ANN's learning algorithm has been applied to improve the performance of fuzzy system and has been shown to be a new and promising approach. Takagi and Hayashi [42] have introduced a feed-forward ANN into the fuzzy inference wherein an

ANN represents a rule, while all membership functions are represented by only one ANN. The algorithm is divided into three major parts: (1) the partition of inference rules, (2) the identification of IF parts, and (3) the identification of THEN parts. Since each rule and all the membership functions are represented by different ANNs, they are trained separately. In other words, the parameters cannot be updated concurrently.

Jang [14,15] and Jang and Sun [16] have proposed a method which transforms the fuzzy inference system into a functional equivalent adaptive network and then employs the EBP-type algorithm to update the premise parameters and the least square method to identify the consequence parameters. Meanwhile, Fukuda and Shibata [8], Shibata et al. [41], and Wang and Mendel [43] have also presented similar methods. Moreover, Nakayama et al. [39] have proposed a so-called FNN, which has a special structure for realizing a fuzzy inference system wherein each membership function consists of one or two sigmoid functions for each inference rules. Owing to lack of membership function setup procedure, the rule determination and the membership function setup are decided by the so-called experts where the decision is very subjective. Lin and Lee [33] have proposed a so-called neural-network-based fuzzy logic control system (NN-FLCS) wherein they introduced the low-level learning power of neural networks in fuzzy logic system and provided a high-level human-understandable meaning to normal connectionist architecture. In addition, Kuo and Cohen [27] have also introduced a feed-forward ANN into the fuzzy inference represented by Takagi's fuzzy modeling and applied it to a multisensor integration, whereas Buckley and Hayashi [2] have surveyed recent findings on learning algorithm and applications of FNNs. Furthermore, Buckley and Hayashi have also introduced several methods in the error back-propagation learning algorithms.

The abovementioned FNNs are only appropriate for digital data. However, the expert's knowledge is always of fuzzy type. Thus, some researchers have attempted to address the problem. Ishibuchi et al. [12,13] have proposed the learning methods of neural networks to utilize not only the digital data but also the expert knowledge represented by fuzzy if-then rules. Kuo and Xue [26] have proposed a novel FNN whose inputs, outputs, and weights are all asymmetric Gaussian functions. The learning algorithm is an

EBP-type learning procedure. Kuo et al. continued to improve the proposed FNN by combining the genetic algorithm [24,23]. Kuo et al. also presented a fuzzy unsupervised neural network, fuzzy SOM [25,21,22,44] for grouping the parts. In addition, Lin and Lu [34] and Lin [35] have also presented an FNN capable of handling both fuzzy inputs and outputs.

3. Methodology

Section 2 has presented the relevance of GT as well as some necessary information. The proposed clustering scheme is presented in more detail in this section. Fig. 1 illustrates the flowchart of the parts clustering scheme based on the fuzzy adaptive resonance theory 2 (fuzzy ART2) neural network. It consists of five components: (1) image acquisition, (2) image processing, (3) feature extraction, (4) pattern recognition, and (5) parts clustering. The following sections present a more detailed discussion of each component.

3.1. Image acquisition

The main purpose of this setup is to capture the part image by using the charge-coupled-device (CCD) camera. In order to reveal the influence of different kinds of shifts and noise, the corresponding images are generated.

3.2. Image processing

Since the image from the vision system may be distorted because of environmental conditions, this

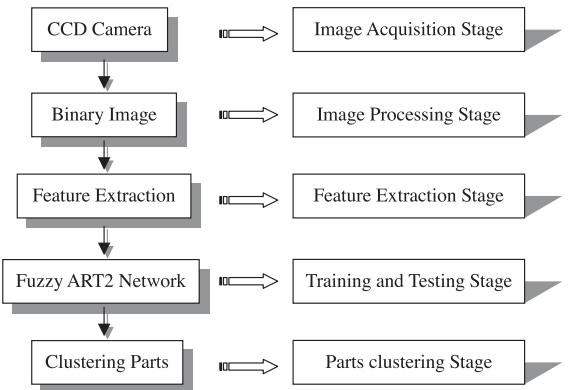


Fig. 1. The flow chart of the part clustering scheme.

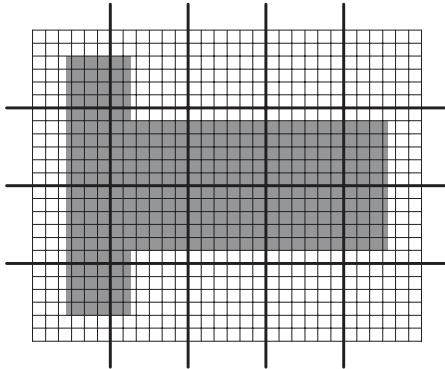


Fig. 2. Segmentation of image.

step mainly employs the image processing technique to improve the quality of captured images. This can lead to better image information. The present study applied the threshold method [9], which is one of the most frequently applied image segmentation techniques. The main concept of this technique is to transform the image brightness to binary values, which represent the brightness and the darkness, respectively. If the brightness value is over the threshold value, then it is 0; otherwise, it is 1. During the binary process, the image gray distribution histogram reveals that the image consists of two components, the object and the background gray models. In order to extract the object from the background, it is necessary to specify a threshold value, λ^* , which can separate the object from the background models. If the corresponding value of pixel (x,y) , whose gray value is $g(x,y)$, is larger than λ^* , then this pixel is the object pixel; otherwise, it is background pixel. The formula is as follows:

$$g(x,y) = \begin{cases} b_0, & g(x,y) > \lambda^* \\ b_1, & g(x,y) \leq \lambda^* \end{cases} \quad (1)$$

where b_0 and b_1 represent the object and the background's gray values, respectively. Thus, we can have

an image which only processes two values, 0 or 1. Besides, it is much easier to recognize the object and the background. Thus, the only thing to do is to find the pixels of the object, if the main purpose is to discuss the object's image.

3.3. Feature extraction

Before implementing the fuzzy ART2 neural network, the features are first extracted from the image. The main concept is to cut the acquired image into several blocks. For instance, the binary image (30×24) is cut into 20 (5×4) blocks. Thus, each block consists of 36 (6×6) pixels, as shown in Fig. 2, and represents a feature of the image, which is calculated by summing the binary values of pixels inside the block. However, due to the geometry of the part, it is difficult to determine the membership of some pixels, as illustrated in Fig. 3. Thus, the Fuzzy ART2 neural network developed in this study can overcome the situation mentioned earlier. In other words, the features are fuzzy instead of crisp; for instance, the total value of the circled block in Fig. 3 is between 15 and 20. Thus, the feature extraction procedures can be listed as follows based on the abovementioned concept.

- Step 1. Segment the image into several blocks.
- Step 2. Calculate the fuzzy interval for each block.
- Step 3. Normalize the interval in $[0,1]$. In Fig. 3, the normalized interval is $[0.4117, 0.5556]$.
- Step 4. Determine the average index. The average index is determined by the geometric mean of pessimistic (l) and optimistic (u) indices. If the pessimistic index is equal to 0, then

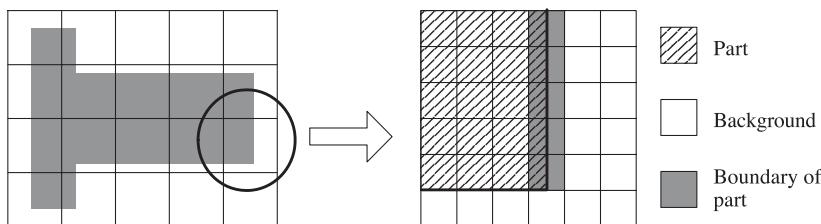


Fig. 3. Processing of fuzzy blocks.

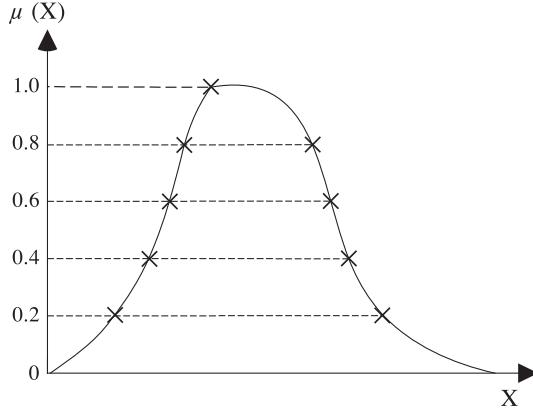


Fig. 4. Asymmetric bell-shaped fuzzy number.

apply arithmetic mean. The calculation procedure is as follows:

$$\mu_{optimal} = \begin{cases} (l \times u)^{1/2}, & \text{if } l \neq 0 \\ (l + u)/2, & \text{if } l = 0 \end{cases} \quad (2)$$

This average index (μ) is the one with the membership value of 1. Thus, we can determine the triangular fuzzy number $\tilde{A} = (l, \mu, u)$.

Step 5. Transform the data.

The present study employs the asymmetric bell-shaped fuzzy number, as shown in Fig. 4, since it can accelerate the convergence of the network [26]. The asymmetric bell-shaped fuzzy number, $\tilde{B} = (\mu, \sigma^L, \sigma^R)_{L-R}$, is determined by the pessimistic index and optimistic index of the fuzzy number $\tilde{A} = (l^*, \mu^*, u^*)$, as shown in Step 4. The formulation is as follows:

$$\mu = \text{average index(membership value is 1)} \quad (3)$$

$$\sigma^L = \frac{\mu^* - l^*}{3}, \quad (4)$$

$$\sigma^R = \frac{u^* - \mu^*}{3}, \quad (5)$$

$$\tilde{B}(x) = \begin{cases} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma^L}\right)^2\right) & , x < \mu \\ 1 & , x = \mu \\ \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma^R}\right)^2\right) & , \text{otherwise} \end{cases} \quad (6)$$

where μ , σ^L , and σ^R represent the mean of fuzzy number \tilde{B} , left width and right width, respectively. In summary, the main objective of image processing and transformation is to obtain the fuzzy input data for the fuzzy ART2 neural network.

3.4. Pattern recognition (FNN)

After the features of the part image have been extracted, the proposed FNN called the fuzzy ART2 neural network is employed to automatically cluster the parts. Most of the FNNs proposed in the literatures are supervised and they only handle the actual real inputs and outputs. Although Lin [35], Ishibuchi et al. [13], Kuo and Xue [26], and Kuo et al. [24] have presented the FNNs with fuzzy inputs, weights, and outputs, yet they are all supervised networks. However, for the purpose of clustering, supervised networks are not feasible. Although the unsupervised neural network proposed in [25] is fuzzy SOM, it yet needs visual examination to determine the number of clusters. The proposed fuzzy ART2 does not have the abovementioned shortcomings. Like Kuo et al.'s previous works, both the fuzzy inputs and weights are all asymmetric fuzzy numbers defined as $\tilde{A} = (\mu, \sigma^L, \sigma^R)_{L-R}$ and

$$\tilde{A}(x) = \begin{cases} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma^L}\right)^2\right) & , x < \mu \\ 1 & , x = \mu \\ \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma^R}\right)^2\right) & , \text{otherwise} \end{cases} \quad (7)$$

where μ , σ^L , and σ^R represent the mean, left width, and right width, respectively. Since the input vectors and connection weight vectors of the fuzzy ART2 neural network are fuzzified, the addition, multiplication, and nonlinear mapping of fuzzy number numbers are necessary for defining the proposed network.

3.4.1. Operations of fuzzy numbers

The fuzzy operations are defined as follows:

$$\mu_{\tilde{X}+\tilde{Y}}(z) = \max\{\mu_{\tilde{X}}(x) \wedge \mu_{\tilde{Y}}(y) | z = x + y\} \quad (8)$$

$$\mu_{\tilde{X} \cdot \tilde{Y}}(z) = \max\{\mu_{\tilde{X}}(x) \wedge \mu_{\tilde{Y}}(y) | z = x \cdot y\} \quad (9)$$

$$\mu_{f(\overline{Net})}(z) = \max\{\mu_{\overline{Net}}(x) | z = f(x)\} \quad (10)$$

where \tilde{X} , \tilde{Y} , and \tilde{Z} are all fuzzy numbers, $\mu(\cdot)$ denotes the membership function of each fuzzy number, and \wedge is the minimum operator. The α -cut of the fuzzy numbers is \tilde{X} which is defined as:

$$\tilde{X}[\alpha] = \{x | \mu_{\tilde{X}} \geq \alpha, x \in \mathcal{R}\} \quad \text{for } 0 < \alpha \leq 1 \quad (11)$$

After α -cutting the fuzzy number, the above equation can be rewritten as:

$$\tilde{X}[\alpha] = [\bar{X}[\alpha]^L, \bar{X}[\alpha]^U] \quad (12)$$

where $\bar{X}[\alpha]^L$ and $\bar{X}[\alpha]^U$ are the upper and the lower bounds of the α -level set. In addition, the corresponding operators are summarized in the following equations.

$$\begin{aligned} \tilde{X}[\alpha] + \tilde{Y}[\alpha] &= [\bar{X}[\alpha]^L, \bar{X}[\alpha]^U] + [\bar{Y}[\alpha]^L, \bar{Y}[\alpha]^U] \\ &= [\bar{X}[\alpha]^L + \bar{Y}[\alpha]^L, \bar{X}[\alpha]^U + \bar{Y}[\alpha]^U] \end{aligned} \quad (13)$$

$$\begin{aligned} \tilde{X}[\alpha] - \tilde{Y}[\alpha] &= [\bar{X}[\alpha]^L, \bar{X}[\alpha]^U] - [\bar{Y}[\alpha]^L, \bar{Y}[\alpha]^U] \\ &= [\bar{X}[\alpha]^L, \bar{X}[\alpha]^U] + [-\bar{Y}[\alpha]^U, -\bar{Y}[\alpha]^L] \\ &= [\bar{X}[\alpha]^L - \bar{Y}[\alpha]^U, \bar{X}[\alpha]^U - \bar{Y}[\alpha]^L] \end{aligned} \quad (14)$$

$$\begin{aligned} \tilde{X}[\alpha] \cdot \tilde{Y}[\alpha] &= [\bar{X}[\alpha]^L, \bar{X}[\alpha]^U] \cdot [\bar{Y}[\alpha]^L, \bar{Y}[\alpha]^U] \\ &= [\min\{\bar{X}[\alpha]^L \cdot \bar{Y}[\alpha]^L, \bar{X}[\alpha]^L \cdot \bar{Y}[\alpha]^U, \\ &\quad \bar{X}[\alpha]^U \cdot \bar{Y}[\alpha]^L, \bar{X}[\alpha]^U \cdot \bar{Y}[\alpha]^U\}, \\ &\quad \max\{\bar{X}[\alpha]^L \cdot \bar{Y}[\alpha]^L, \bar{X}[\alpha]^L \cdot \bar{Y}[\alpha]^U, \bar{X}[\alpha]^U \\ &\quad \cdot \bar{Y}[\alpha]^L, \bar{X}[\alpha]^U \cdot \bar{Y}[\alpha]^U\}] \end{aligned} \quad (15)$$

$$\begin{aligned} \tilde{X}[\alpha]/\tilde{Y}[\alpha] &= [\bar{X}[\alpha]^L, \bar{X}[\alpha]^U] / [\bar{Y}[\alpha]^L, \bar{Y}[\alpha]^U] \\ &= [\min\{\bar{X}[\alpha]^L / \bar{Y}[\alpha]^L, \bar{X}[\alpha]^L / \bar{Y}[\alpha]^U, \\ &\quad \bar{X}[\alpha]^U / \bar{Y}[\alpha]^L / \bar{X}[\alpha]^U / \bar{Y}[\alpha]^U\}, \\ &\quad \max\{\bar{X}[\alpha]^L / \bar{Y}[\alpha]^L, \bar{X}[\alpha]^L / \bar{Y}[\alpha]^U, \\ &\quad \bar{X}[\alpha]^U / \bar{Y}[\alpha]^L, \bar{X}[\alpha]^U / \bar{Y}[\alpha]^U\}] \end{aligned} \quad (16)$$

$$\begin{aligned} f(\overline{Net}[\alpha]) &= f([\overline{Net}[\alpha]^L, \overline{Net}[\alpha]^U]) \\ &= [f(\overline{Net}[\alpha]^L), f(\overline{Net}[\alpha]^U)] \end{aligned} \quad (17)$$

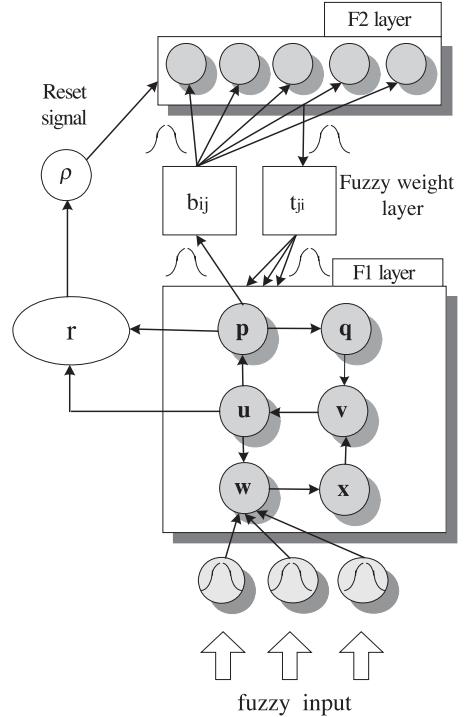


Fig. 5. The structure of fuzzy ART2 neural network.

Thus, under the assumption of $0 \leq \bar{Y}[\alpha]^L \leq \bar{Y}[\alpha]^U$, $\tilde{X}[\alpha] \cdot \tilde{Y}[\alpha]$ can be rewritten as:

$$\begin{aligned} &[\min\{\bar{X}[\alpha]^L \cdot \bar{Y}[\alpha]^L, \bar{X}[\alpha]^L \cdot \bar{Y}[\alpha]^U\}, \\ &\quad \max\{\bar{X}[\alpha]^U \cdot \bar{Y}[\alpha]^L, \bar{X}[\alpha]^U \cdot \bar{Y}[\alpha]^U\}] \end{aligned} \quad (18)$$

3.4.2. Network structure

Fig. 5 presents the framework of fuzzy ART2 neural network. The input and output relation of the proposed fuzzy ART2 neural network is defined by extension principle and can be written as follows:

3.4.2.1. Input layer (F1 layer)

$$\tilde{O}_{pi} = \tilde{X}_{pi}, i = 1, 2, \dots, n. \quad (19)$$

The F1 layer consists of six types of units (the w , x , u , v , p , and q units):

$$\tilde{w}_{kpi}, i = 1, 2, \dots, n \quad (20)$$

$$\tilde{x}_{kpi}, i = 1, 2, \dots, n \quad (21)$$

$$\tilde{u}_{kpi}, i = 1, 2, \dots, n \quad (22)$$

$$\tilde{v}_{kpi}, i = 1, 2, \dots, n \quad (23)$$

$$\tilde{p}_{kpi}, i = 1, 2, \dots, n \quad (24)$$

$$\tilde{q}_{kpi}, i = 1, 2, \dots, n \quad (25)$$

3.4.2.2. Weight layer. The weight layer consists of two types of weights, down-top and top-down weights.

$$\text{Down-top weight: } \tilde{b}_{ij}, j = 1, 2, \dots, m \quad (26)$$

$$\text{Top-down weight: } \tilde{t}_{ji}, j = 1, 2, \dots, m \quad (27)$$

3.4.2.3. Output layer (F2 layer)

- (1) Calculate the fuzzy vector between the fuzzy weight, down-top, and p unit fuzzy vector of F1 layer for each input node. The fuzzy vector is defined as:

$$\tilde{T}_j = \tilde{p}_{ij} \otimes \tilde{b}_{ij} \quad (28)$$

where \otimes is the fuzzy delete operation.

- (2) Apply the transformation method proposed by Chen and Hwang [5] to defuzzify the fuzzy vector and compute the defuzzified values g_{ij} .
- (3) Choose the winner with the maximum T_j .

$$T_j = \sum_i g_{ij}, \quad (29)$$

$$T_j^* = \max_j \{T_j\} \quad (30)$$

3.4.2.4. Reset or resonance layer. The layer will decide if the inputting fuzzy vector is “reset” or “resonance” through vigilance parameter testing. The check for a reset gives $\|r\|$. However, $(\|r\| + e) < \rho$ for a valid value of ρ (ρ is the vigilance parameter), so the winning cluster unit will be allowed to learn current pattern.

3.4.3. Learning algorithm

For the above equations, the winner unit is calculated for fuzzy inputs and the fuzzy weights. The fuzzy relation of network structure and parameters definition can be found as follows.

3.4.3.1. Parameters definition

- | | |
|----------|--|
| n | Number of input units. |
| m | Number of cluster units. |
| a, b | Fixed weights in the F1 layer. |
| c | Fixed weight used in testing for reset. |
| d | Activation of winning of F2 units. |
| e | Small parameter introduced to prevent division by zero where the norm of a vector is zero. |
| θ | Noise suppression parameter. |
| ρ | Vigilance parameter. |

3.4.3.2. Input layer (F1 layer)

$$\tilde{X}_i[\alpha] = [\bar{X}_i[\alpha]^L, \bar{X}_i[\alpha]^U], i = 1, 2, \dots, n \quad (31)$$

$$\begin{aligned} \vec{X}[\alpha] &= (\tilde{X}_1[\alpha], \dots, \tilde{X}_i[\alpha], \dots, \tilde{X}_n[\alpha]) \\ &= ([\bar{X}_1[\alpha]^L, \bar{X}_1[\alpha]^U], \dots, [\bar{X}_i[\alpha]^L, \bar{X}_i[\alpha]^U], \dots \\ &\quad [\bar{X}_n[\alpha]^L, \bar{X}_n[\alpha]^U]) \end{aligned} \quad (32)$$

Six types of units:

The w unit:

$$\begin{aligned} \vec{w}_i[\alpha] &= (\tilde{X}_1[\alpha], \dots, \tilde{X}_i[\alpha], \dots, \tilde{X}_n[\alpha]) \\ &\quad + a(\tilde{u}_1[\alpha], \dots, \tilde{u}_i[\alpha], \dots, \tilde{u}_n[\alpha]) \\ &= ([\bar{X}_1[\alpha]^L, \bar{X}_1[\alpha]^U], \dots, [\bar{X}_i[\alpha]^L, \bar{X}_i[\alpha]^U], \dots \\ &\quad [\bar{X}_n[\alpha]^L, \bar{X}_n[\alpha]^U]) + ([\bar{a}\bar{u}_1[\alpha]^L, \bar{a}\bar{u}_1[\alpha]^U], \dots, \\ &\quad [\bar{a}\bar{u}_i[\alpha]^L, \bar{a}\bar{u}_i[\alpha]^U], \dots, [\bar{a}\bar{u}_n[\alpha]^L, \bar{a}\bar{u}_n[\alpha]^U]) \\ &= ([\bar{X}_1[\alpha]^L + \bar{a}\bar{u}_1[\alpha]^L, \bar{X}_1[\alpha]^U + \bar{a}\bar{u}_1[\alpha]^U], \dots, \\ &\quad [\bar{X}_i[\alpha]^L + \bar{a}\bar{u}_i[\alpha]^L, \bar{X}_i[\alpha]^U + \bar{a}\bar{u}_i[\alpha]^U], \dots, \\ &\quad [\bar{X}_n[\alpha]^L + \bar{a}\bar{u}_n[\alpha]^L, \bar{X}_n[\alpha]^U + \bar{a}\bar{u}_n[\alpha]^U]) \\ &= ([\bar{w}_1[\alpha]^L, \bar{w}_1[\alpha]^U], \dots, [\bar{w}_i[\alpha]^L, \bar{w}_i[\alpha]^U], \dots \\ &\quad [\bar{w}_n[\alpha]^L, \bar{w}_n[\alpha]^U]) = (\tilde{w}_1[\alpha], \dots, \tilde{w}_i[\alpha], \dots, \tilde{w}_n[\alpha]) \end{aligned} \quad (33)$$

The x unit:

$$\begin{aligned} \vec{x}_i[\alpha] &= \frac{\vec{w}}{e + \|\vec{w}\|} = \frac{[\bar{w}_i[\alpha]^L, \bar{w}_i[\alpha]^U]}{e + \left(\sum_i \{\bar{w}_i[\alpha]^L, \bar{w}_i[\alpha]^U\}^2 \right)^{1/2}} \\ &= \frac{[\bar{w}_1[\alpha]^L, \bar{w}_1[\alpha]^U]}{e + [\bar{W}_i[\alpha]^L, \bar{W}_i[\alpha]^U]} ; \\ \bar{W}[\alpha] &= \sum \bar{w}[\alpha] = \frac{[\bar{w}_i[\alpha]^L, \bar{w}_i[\alpha]^U]}{[e + \bar{W}_i[\alpha]^L, e + \bar{W}_i[\alpha]^U]} \\ &= \left[\frac{\bar{w}_i[\alpha]^L}{e + \bar{W}_i[\alpha]^U}, \frac{\bar{w}_i[\alpha]^U}{e + \bar{W}_i[\alpha]^L} \right] \\ &= (\tilde{x}_1[\alpha], \dots, \tilde{x}_i[\alpha], \dots, \tilde{x}_n[\alpha]) \end{aligned} \quad (34)$$

The u unit:

$$\begin{aligned}
 \tilde{\mathbf{u}}_i[\alpha] &= \frac{\vec{\mathbf{v}}}{e + \|\vec{\mathbf{v}}\|} = \frac{\left[\bar{v}_i[\alpha]^L, \bar{v}_i[\alpha]^U \right]}{e + \left(\sum_i \left\{ \bar{v}_i[\alpha]^L, \bar{v}_i[\alpha]^U \right\}^2 \right)^{1/2}} \\
 &= \frac{\left[\bar{v}_i[\alpha]^L, \bar{v}_i[\alpha]^U \right]}{e + \left[\bar{V}_i[\alpha]^L, \bar{V}_i[\alpha]^U \right]} ; \quad \bar{V}[\alpha] = \sum \bar{v}[\alpha] \\
 &= \frac{\left[\bar{v}_i[\alpha]^L, \bar{v}_i[\alpha]^U \right]}{e + \left[\bar{V}_i[\alpha]^L, e + \bar{V}_i[\alpha]^U \right]} \\
 &= \left[\frac{\bar{v}_i[\alpha]^L}{e + \bar{V}_i[\alpha]^U}, \frac{\bar{v}_i[\alpha]^U}{e + \bar{V}_i[\alpha]^L} \right] \\
 &= (\tilde{\mathbf{u}}_1[\alpha], \dots, \tilde{\mathbf{u}}_i[\alpha], \dots, \tilde{\mathbf{u}}_n[\alpha])
 \end{aligned} \tag{35}$$

The q unit:

$$\begin{aligned}
 \tilde{q}_i[\alpha] &= \frac{\vec{\mathbf{p}}}{e + \|\vec{\mathbf{p}}\|} \\
 &= \frac{\left[\bar{p}_i[\alpha]^L, \bar{p}_i[\alpha]^U \right]}{[e, e] + \left(\sum_i \left\{ \bar{p}_i[\alpha]^L, \bar{p}_i[\alpha]^U \right\}^2 \right)^{1/2}} \\
 &= \frac{\left[\bar{p}_1[\alpha]^L, \bar{p}_1[\alpha]^U \right]}{e + \left[\bar{P}_i[\alpha]^L, \bar{P}_i[\alpha]^U \right]} ; \quad \bar{P}[\alpha] = \sum \bar{p}[\alpha] \\
 &= \frac{\left[\bar{p}_1[\alpha]^L, \bar{p}_1[\alpha]^U \right]}{e + \left[\bar{P}_i[\alpha]^L, e + \bar{P}_i[\alpha]^U \right]} \\
 &= \left[\frac{\bar{p}_i[\alpha]^L}{e + \bar{P}_i[\alpha]^U}, \frac{\bar{p}_i[\alpha]^U}{e + \bar{P}_i[\alpha]^L} \right] \\
 &= (\tilde{q}_1[\alpha], \dots, \tilde{q}_i[\alpha], \dots, \tilde{q}_n[\alpha])
 \end{aligned} \tag{36}$$

The p unit:

$$\begin{aligned}
 \tilde{\mathbf{p}}_i[\alpha] &= (\tilde{\mathbf{u}}_1[\alpha], \dots, \tilde{\mathbf{u}}_i[\alpha], \dots, \tilde{\mathbf{u}}_n[\alpha]) \\
 &\quad + d(\tilde{\mathbf{t}}_1[\alpha], \dots, \tilde{\mathbf{t}}_i[\alpha], \dots, \tilde{\mathbf{t}}_n[\alpha]) \\
 &= \left(\left[\bar{u}_1[\alpha]^L, \bar{u}_1[\alpha]^U \right], \dots, \left[\bar{u}_i[\alpha]^L, \bar{u}_i[\alpha]^U \right], \dots, \right. \\
 &\quad \left. \left[\bar{u}_n[\alpha]^L, \bar{u}_n[\alpha]^U \right] \right) + \left(\left[\bar{d}\tilde{t}_1[\alpha]^L, \bar{d}\tilde{t}_1[\alpha]^U \right], \dots, \right. \\
 &\quad \left. \left[\bar{d}\tilde{t}_i[\alpha]^L, \bar{d}\tilde{t}_i[\alpha]^U \right], \dots, \left[\bar{d}\tilde{t}_n[\alpha]^L, \bar{d}\tilde{t}_n[\alpha]^U \right] \right) \\
 &= \left(\left[\bar{u}_1[\alpha]^L + \bar{d}\tilde{t}_1[\alpha]^L, \bar{u}_1[\alpha]^U + \bar{d}\tilde{t}_1[\alpha]^U \right], \dots, \right. \\
 &\quad \left. \left[\bar{u}_i[\alpha]^L + \bar{d}\tilde{t}_i[\alpha]^L, \bar{u}_i[\alpha]^U + \bar{d}\tilde{t}_i[\alpha]^U \right], \dots, \right. \\
 &\quad \left. \left[\bar{u}_n[\alpha]^L + \bar{d}\tilde{t}_n[\alpha]^L, \bar{u}_n[\alpha]^U + \bar{d}\tilde{t}_n[\alpha]^U \right] \right) \\
 &= \left(\left[\bar{p}_1[\alpha]^L, \bar{p}_1[\alpha]^U \right], \dots, \left[\bar{p}_i[\alpha]^L, \bar{p}_i[\alpha]^U \right], \dots, \right. \\
 &\quad \left. \left[\bar{p}_n[\alpha]^L, \bar{p}_n[\alpha]^U \right] \right) \\
 &= (\tilde{\mathbf{p}}_1[\alpha], \dots, \tilde{\mathbf{p}}_i[\alpha], \dots, \tilde{\mathbf{p}}_n[\alpha])
 \end{aligned} \tag{37}$$

The v unit:

For this unit, apply the transformation method proposed by Chen and Hwang [5] to defuzzify \tilde{x} and \tilde{q} and then test the defuzzified numbers through the activation functions shown below:

$$f(X) = \begin{cases} X & \text{if } X \geq \theta \\ 0 & \text{if } X < \theta \end{cases} \tag{38}$$

$$f(Q) = \begin{cases} Q & \text{if } Q \geq \theta \\ 0 & \text{if } Q < \theta \end{cases} \tag{39}$$

This function treats any signal, which is less than θ as noise and suppresses it (set it to zero and fuzzy numbers to [0,0]).

3.4.3.3. Weight layer

Down – top weight: $\tilde{b}_{ij}[\alpha]$

$$= [\tilde{b}_{ij}[\alpha]^L, \bar{b}_{ij}[\alpha]^U], j = 1, 2, \dots, m \tag{40}$$

Top – down weight: $\tilde{t}_{ji}[\alpha]$

$$= [\tilde{t}_{ji}[\alpha]^L, \bar{t}_{ji}[\alpha]^U], j = 1, 2, \dots, m \tag{41}$$

Transform the asymmetric fuzzy numbers:

$$\begin{cases} \bar{b}_{ij}[\alpha]^L = \mu_{ij} - \sigma_{ij}^L \cdot (-2\ln\alpha)^{1/2}, & \text{if } x \leq \mu \\ \bar{b}_{ij}[\alpha]^U = \mu_{ij} + \sigma_{ij}^R \cdot (-2\ln\alpha)^{1/2}, & \text{if } x > \mu \end{cases} \quad (42)$$

$$\begin{cases} \bar{t}_{ji}[\alpha]^L = \mu_{ji} - \sigma_{ji}^L \cdot (-2\ln\alpha)^{1/2}, & \text{if } x \leq \mu \\ \bar{t}_{ji}[\alpha]^U = \mu_{ji} + \sigma_{ji}^R \cdot (-2\ln\alpha)^{1/2}, & \text{if } x > \mu \end{cases} \quad (43)$$

3.4.3.4. Output layer (F2 layer)

- (1) Calculate the fuzzy vector between the fuzzy weight, down-top, and p unit fuzzy vector of F1 layer for each input node. The fuzzy vector is defined as

$$\begin{aligned} \tilde{T}_j &= \vec{\tilde{p}}_{ij} \otimes \vec{\tilde{b}}_{ij} \\ &= [\bar{p}_{ij}[\alpha]^L, \bar{p}_{ij}[\alpha]^U] \otimes [\bar{b}_{ij}[\alpha]^L, \bar{b}_{ij}[\alpha]^U] \\ &= [[\bar{p}_{ji}[\alpha]^L \cdot \bar{b}_{ij}[\alpha]^L], [\bar{p}_i[\alpha]^U, \bar{b}_{ij}[\alpha]^U]] \\ &= [\bar{T}_j[\alpha]^L, \bar{T}_j[\alpha]^U] \end{aligned} \quad (44)$$

- (2) Apply the transformation method proposed by Chen and Hwang [5] to defuzzify the fuzzy vector and compute the defuzzified values g_{ij} .
- (3) Choose the winner with the maximum T_j .

3.4.3.5. Reset or resonance layer. This layer decides whether the inputted fuzzy vector is “reset” or “resonance” through vigilance parameter testing. The check for a reset gives $\|r\|$.

$$\begin{aligned} \|r\| &= \frac{\|\vec{u} + c\vec{p}\|}{e + \|\vec{u}\| + c\|\vec{p}\|} \\ &= \frac{\|(\bar{u}_i[\alpha], \dots, \bar{u}_n[\alpha]) + c(\bar{p}_i[\alpha], \dots, \bar{p}_n[\alpha])\|}{e + \left(\sum_i \{ \bar{u}_i[\alpha]^L, \bar{u}_i[\alpha]^U \}^2 \right)^{1/2} + c \left(\sum_i \{ \bar{p}_i[\alpha]^L, \bar{p}_i[\alpha]^U \}^2 \right)^{1/2}} \\ &= \left(\sum_i \{ (\bar{u}_i[\alpha]^L + c\bar{p}_i[\alpha]^L, \bar{u}_i[\alpha]^U + c\bar{p}_i[\alpha]^U) \}^2 \right)^{1/2} \\ &= \left[\left(e + \bar{U}_i[\alpha]^L + c\bar{P}_i[\alpha]^L \right), \left[e + \bar{U}_i[\alpha]^U + c\bar{P}_i[\alpha]^U \right] \right] \\ &= \left[\frac{\bar{u}_i + c\bar{p}_i[\alpha]^L}{e + \bar{U}_i[\alpha]^U + c\bar{P}_i[\alpha]^U}, \frac{\bar{u}_i + c\bar{p}_i[\alpha]^U}{e + \bar{U}_i[\alpha]^L + c\bar{P}_i[\alpha]^L} \right] \end{aligned} \quad (45)$$

If the layer is resonant, modify the weights as below:

$$\tilde{b}_{ij} = \frac{\tilde{u}_{ij}}{1-d} = \left[\frac{\bar{u}_{ij}[\alpha]^L}{1-d}, \frac{\bar{u}_{ij}[\alpha]^U}{1-d} \right] \quad (46)$$

$$\tilde{t}_{ji} = \frac{\tilde{u}_{ji}}{1-d} = \left[\frac{\bar{u}_{ji}[\alpha]^L}{1-d}, \frac{\bar{u}_{ji}[\alpha]^U}{1-d} \right] \quad (47)$$

3.4.4. Learning procedures

The learning procedures (Fig. 6) of the fuzzy ART2 neural network are summarized as follows:

- Step 1. Input the fuzzy vector into F1 layer and compute the six units until the u or p unit value is convergent.
- Step 2. Calculate the fuzzy vector between the fuzzy weight, down-top, and p unit fuzzy vector of F1 layer for each input node. Find the maximum value and decide which is the winner.
- Step 3. Input the fuzzy vector into F1 layer and compute the six units until the u unit value is convergent again.
- Step 4. Test the vigilance parameter and then decide whether the layer state is “reset” or “resonance.” If the state is “reset,” set the winner’s T_j to be zero and repeat Step 2 to find the other winner.

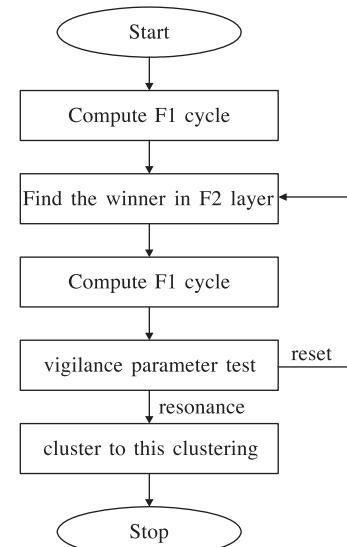


Fig. 6. The fuzzy ART2 neural network learning algorithm.

If all winners do not pass the vigilance parameter test, it is necessary to create a new cluster and add the corresponding weights. If the state is “resonance”, just make the current fuzzy input belong to this cluster and modify the corresponding weights.

4. Model evaluation results and discussion

As presented earlier, the fuzzy ART2 can cluster the parts with fuzzy features into several families. The binary image is cut into 20 blocks. In accord with α -cuts, each fuzzy input has the corresponding interval

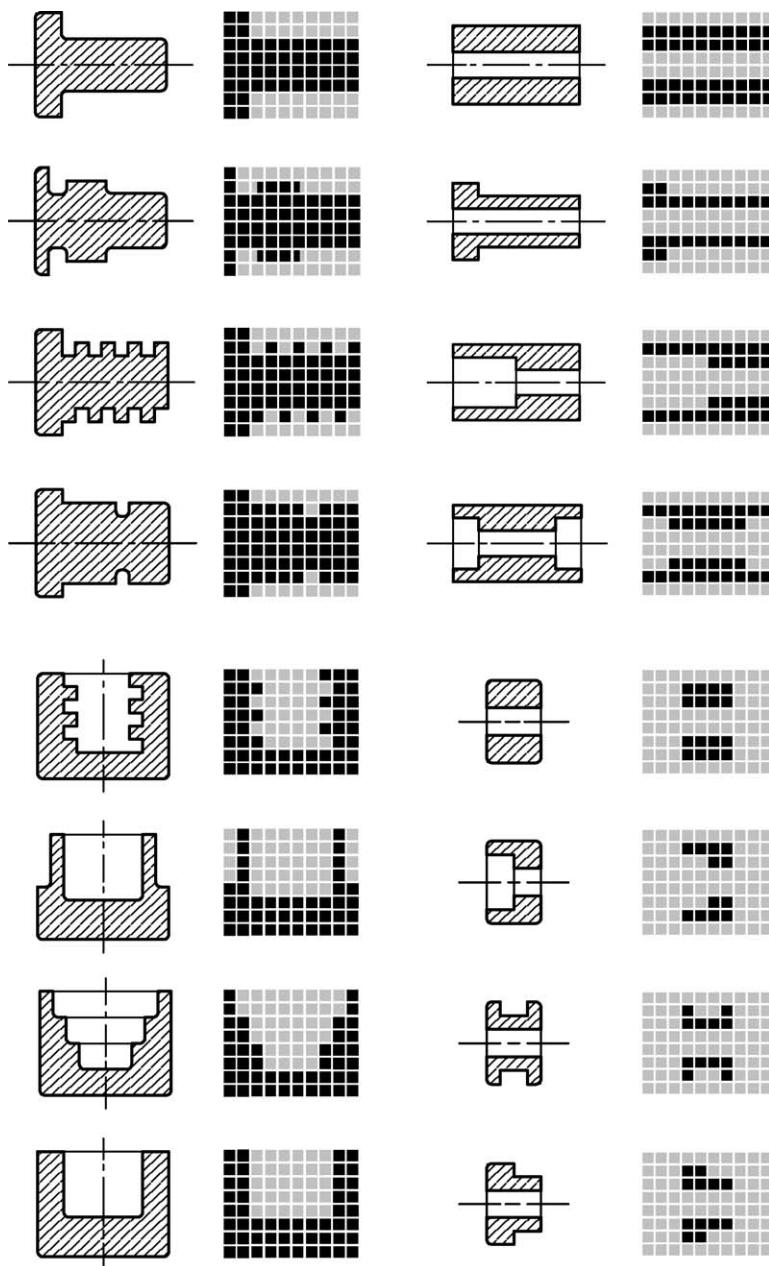


Fig. 7. Sixteen part images.

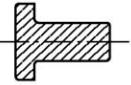
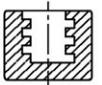
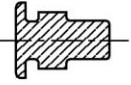
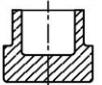
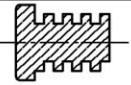
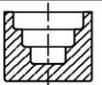
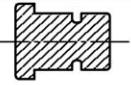
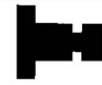
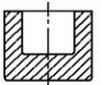
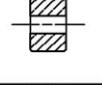
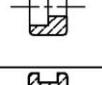
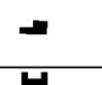
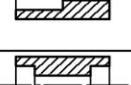
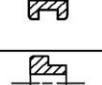
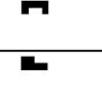
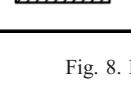
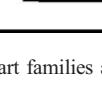
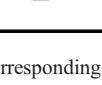
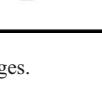
Part Family 1		Part Family 2	
Part	Binary Image	Part	Binary Image
			
			
			
			
Part Family 3		Part Family 4	
Part	Binary Image	Part	Binary Image
			
			
			
			

Fig. 8. Four part families and their corresponding binary images.

for calculation. Finally, the clustering result can be obtained from the output array. The parameter setup is defined as follows:

- (1) Input layer: the number of input nodes is 20, since the image is cut into 20 blocks and the α -cuts levels are 0, 0.2, 0.4, 0.6, and 0.8, respectively.
- (2) The fixed weight in F1 layer parameters: $a=b=0.05$.
- (3) The fixed weight used in testing for reset parameter: $c=0.1$.
- (4) The activation of winning F2 unit parameter: $d=0.9$.
- (5) The noise suppression parameter: $\theta=1 \sqrt{n}=0.16$.

- (6) The vigilance parameter: $\rho=0.94$.
- (7) Initial weight: randomly set up the top-down and down-top weights between $[-0.3, 0.3]$ in accord with the research of Lee [28].

Table 1
Clustering results of fuzzy ART2

α -cut level	$a, b=0.5, d=0.9, c=0.01, \rho=0.94, \theta=0.16$	Accurate number	Error number	Accurate rate (%)
0	823	177	0.823	
0.2	866	134	0.866	
0.4	887	113	0.892	
0.6	892	108	0.884	
0.8	853	147	0.853	

- (8) Stop learning as all input data have been clustered.

Basically, the above parameter setup is determined by several times of testing and references' suggestions.

4.1. Training the parts

The parts used to verify the proposed scheme are adapted from Kamarthi et al. [17], as shown in Fig. 7. The parts are further classified into four families, as illustrated in Fig. 8. For the parts training, the features of the standard parts are first applied to find the best parameter setup. Randomly choose 1000 samples for each α -cuts. The training performance and the accurate rate are listed in Table 1. The accurate rate is defined as:

$$CR\% = \frac{CN}{TN} \times 100\%, \quad (48)$$

where CR, CN, and TN are the correct rate, the number of parts being correctly clustered, and the total number of parts, respectively.

The clustering results indicate that the accurate rate is between 0.8 and 0.9. It represents that the fuzzy ART2 neural network is stable and has higher accuracy.

4.2. The influence of shift on clustering

During the image acquisition, it is very difficult to keep the part exactly in the same position on the working table. Shift of the part is unavoidable.

Table 2
The influence of shift on the part clustering

Rotational angle	Accurate rate (%)		
	Fuzzy c-means	Fuzzy SOM	Fuzzy ART2
0°	80.675	100	100
1°	82.225	100	100
2°	80.669	100	100
3°	81.256	99.780	99.433
4°	82.194	94.103	96.449
5°	82.644	93.750	92.359
6°	81.369	92.612	91.133
7°	81.477	88.051	90.642
8°	83.881	87.500	87.226
9°	83.881	87.119	85.735
10°	86.181	81.250	84.703
Average accurate rate	82.405	93.106	93.425

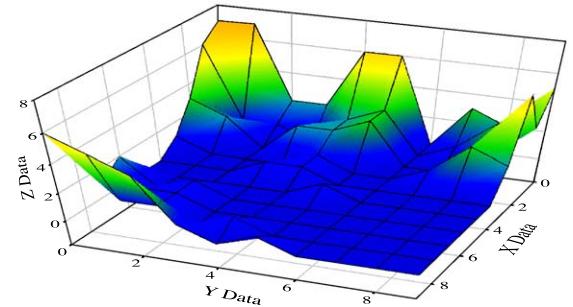


Fig. 9. Clear output map.

However, a small angle of shift may cause the image acquired to be not exactly the same as the training sample. Thus, here, different angles of shift are generated in order to examine the shift. Besides testing the proposed network, the present study also compares fuzzy SOM neural network [25]. Table 2 presents the analytical results.

Table 2 reveals that averagely, both fuzzy SOM and fuzzy ART2 can provide more adequate recognition than fuzzy c-means [40]. If the shifting angle is not very large, these two methods can cluster almost accurately. However, if the shifting angle is very large, say 10°, then their results are not as good as fuzzy c-means. According to the average accurate rate, fuzzy ART2 is better than fuzzy SOM. But fuzzy SOM is based on SOM neural network, it is necessary to make the visual examination for the output array. Some-

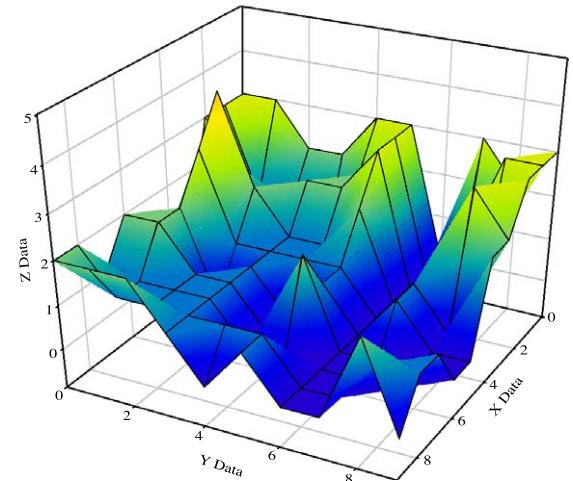


Fig. 10. Not clear output map.

Table 3
The influence of noise on the part clustering

The degree of noise (%)	Accurate rate (%)		
	Fuzzy c-means	Fuzzy SOM	Fuzzy ART2
0	80.675	100	100
5	71.525	98.653	99.422
10	64.574	87.791	90.207
15	53.186	75.082	82.925
20	53.688	62.507	66.687
25	53.131	56.259	59.039
30	53.425	52.254	56.942
Average accurate rate	61.458	76.078	79.317

times it is quite difficult to determine the number of clusters by examining the output array. Such method may create more deviations and is not objective. If the distribution of samples is clear like Fig. 9, it is easy to make the decision. But, the researchers will have difficulty in determining the number of clusters if the output array is like Fig. 10. In addition, fuzzy ART compares similarities of patterns. It can cluster automatically and objectively. It is more reasonable to apply fuzzy ART instead of fuzzy SOM for practical reason.

4.3. The influence of noise on clustering

In order to find out the influence of noise on clustering, it is necessary to create the noisy images for the paper. First, the noise ratio is prespecified. If the noise ratio is n , then define the large random value, L , which is the product of pixel number and noise ratio. Then use a PC to generate the random number in $[1, L]$. If a pixel is selected, change its pixel value.

Table 3 indicates that both fuzzy SOM and fuzzy ART2 are sensitive to noise. As the noise level is not very large, they can cluster well. But if the noise level is very large, they will make bigger bias. According to the average accurate rate, fuzzy ART2 is still better than fuzzy SOM. Besides, like mentioned in the above subsection, fuzzy ART2 can automatically cluster the samples without visual examination.

5. Conclusions

The present study has demonstrated a novel fuzzy neural network, the Fuzzy ART2 neural network, for

clustering parts into several families. The fuzzy ART2 neural network can correctly cluster the parts as Kamarthi et al. [17] specified. Even under the shift and noise conditions, fuzzy ART2 also can have very promising results compared with fuzzy SOM and fuzzy c-means. The testing results also indicate that the network is more stable and accurate. In addition, the Fuzzy ART2 neural network, which is a kind of unsupervised network, does not need a very long training time. It can fit the requirements of the industries. Besides the current application, the fuzzy ART2 neural network has been applied in other areas, e.g., marketing segmentation, and got very promising results.

Acknowledgement

This study is partially supported by the National Science Council of Taiwan Government, Province of China under contract number NSC 91-2213-E-027-010. Her support is appreciated.

References

- [1] A.A.S. Awwal, M.A. Karim, Machine parts recognition using a trinary associative memory, Optical Engineering 28 (5) (1989) 537–543.
- [2] J.J. Buckley, Y. Hayashi, Fuzzy neural networks: a survey, Fuzzy Sets and Systems 66 (1994) 1–13.
- [3] G.A. Carpenter, S. Grossberg, ART2: self-organization of stable category recognition codes for analogue input patterns, Applied Optics 26 (1987) 4919–4930.
- [4] T.P. Caudell, S.D.G. Smith, G.C. Johnson, An application of neural networks to group technology, Applications of Artificial Neural Networks 1490 (1991) 612–621.
- [5] S.J. Chen, C.L. Hwang, Fuzzy Multiple Attribute Decision Making Method and Application, A State-of-the-Art Survey, Springer-Verlag, New York, 1992.
- [6] S.J. Chen, C.S. Cheng, A neural network-based cell formation algorithm in cellular manufacturing, International Journal of Production Research 33 (2) (1995) 293–318.
- [7] Y. Chung, A. Kusiak, Grouping parts with a neural network, Journal of Manufacturing Systems 13 (4) (1994) 262–275.
- [8] T. Fukuda, T. Shibata, Hierarchical intelligent control for robotic motion by using fuzzy, artificial intelligence, and neural network, Proceedings of IJCNN'92, 1992, pp. I-269–I-274.
- [9] R.J. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley Publishing, 1992.
- [10] I. Ham, E.V. Goncalves, C.P. Han, An integrated approach to group technology part family data base design based on

- artificial intelligence technology, CIRP Annals 37 (1) (1988).
- [11] A.R. Harish, P. Gu, Expert self-organizing neural network for the design of cellular manufacturing systems, Journal of Manufacturing Systems 13 (5) (1994) 346–358.
- [12] H. Ishibuchi, H. Okada, R. Fujioka, H. Tanaka, Neural networks that learn from fuzzy if-then rules, IEEE Transactions on Fuzzy Systems FS-1 (2) (1993 May) 85–97.
- [13] H. Ishibuchi, K. Kwon, H. Tanaka, A learning algorithm of fuzzy neural networks with triangular fuzzy weights, Fuzzy Sets and Systems 71 (1995) 277–293.
- [14] J.-S.R. Jang, Fuzzy modeling using generalized neural networks and kalman filter algorithm, Proceedings of Ninth National Conference on Artificial Intelligence, 1991, pp. 762–767.
- [15] J.-S.R. Jang, Fuzzy controller design without domain expert, Proceedings of IEEE International Conference on Fuzzy Systems (1992) 289–296.
- [16] J.-S.R. Jang, C.-T. Sun, Functional equivalence between radial basic function networks and fuzzy inference systems, IEEE Transactions on Neural Networks 4 (1) (1993) 156–159.
- [17] S. Kamarthi, S.T. Kumara, F.S. Yu, I. Ham, Neural networks and their applications in component design data retrieval, Journal of Intelligent Manufacturing (1990) 125–140.
- [18] Y. Kao, Y.B. Moon, A unified group technology implementation using the backpropagation learning rule of neural networks, Computers & Industrial Engineering 20 (1991) 425–437.
- [19] S. Kaparthy, N.C. Suresh, A neural network system for shape-based classification and coding of rotational parts, International Journal of Production Research 29 (9) (1991) 1771–1784.
- [20] S. Kaparthy, N.C. Suresh, Machine-component cell formation in group technology: a neural network approach, International Journal of Production Research 30 (1992) 1353–1367.
- [21] T. Kohonen, Self-Organization and Associative Memory, Springer-Verlag, Berlin, 1988.
- [22] T. Kohonen, An introduction to neural computing, Neural Networks 1 (1) (1988) 3–16.
- [23] R.J. Kuo, P.C. Wu, C.P. Wang, An intelligent sales forecasting system through integration of artificial neural network and fuzzy neural network with fuzzy weight-elimination, Journal of Neural Networks 15 (7) (2002 September) 909–925.
- [24] R.J. Kuo, J.H. Chen, Y.C. Hwang, An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network, Fuzzy Sets and Systems 118/1 (2001) 21–45.
- [25] R.J. Kuo, S.S. Chi, P.W. Teng, Generalized part family formation through fuzzy self-organizing feature map neural network, International Journal of Computers in Industrial Engineering 40 (2001) 79–100.
- [26] R.J. Kuo, K.C. Xue, A decision support system for sales forecasting through fuzzy neural network with asymmetric fuzzy weights, Decision Support Systems 24 (2) (1998 December) 105–126.
- [27] R.J. Kuo, P.H. Cohen, Integration of artificial neural networks and fuzzy modeling for intelligent control of machining, Fuzzy Sets and Systems 98 (1) (1998 August) 15–31.
- [28] D.G. Lee, Preliminary results of applying neural networks to ship image recognition, in: Proceedings of IJCNN, vol. 2, 1989, pp. 576–579.
- [29] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller: Parts I and II, IEEE Transactions on Systems, Man, and Cybernetics SMC-20 (2) (1990) 404–435.
- [30] H. Lee, C.O. Malave, S. Ramachandran, A self-organizing neural network approach for the design of cellular manufacturing systems, Journal of Intelligent Manufacturing 3 (1992) 325–332.
- [31] S. Lee, H.P. Wang, Manufacturing cell formation: a dual-objective simulated annealing approach, International Journal of Advanced Manufacturing Technology 7 (1992) 314–320.
- [32] T.W. Liao, K.S. Lee, Integration of a feature-based CAD system and an ART1 Neural Model for GT coding and part family forming, Computers & Industrial Engineering 26 (1) (1994) 93–104.
- [33] C.T. Lin, C.S.G. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Transactions on Computers C-40 (12) (1991) 1320–1336.
- [34] C.T. Lin, Y.C. Lu, A neural fuzzy system with linguistic teaching signals, IEEE Transactions on Fuzzy Systems 3 (2) (1995) 169–189.
- [35] C.T. Lin, A neural fuzzy control system with structure and parameter learning, Fuzzy Sets and Systems 70 (1995) 183–212.
- [36] R.P. Lippmann, An introduction to computing with neural nets, IEEE ASSP Magazine (1987 April) 4–22.
- [37] E. Masson, Y.J. Wang, Introduction to computation and learning in artificial neural networks, European Journal of Operational Research 47 (1) (1990) 1–28.
- [38] Y.B. Moon, S.C. Chi, Generalized part family formation using neural network techniques, Journal of Manufacturing Systems 11 (1992) 149–159.
- [39] S. Nakayama, S. Horikawa, T. Furuhashi, Y. Uchikawa, Knowledge acquisition of strategy and tactics using fuzzy neural networks, Proceedings of IJCNN'92, 1992, pp. II-751–II-756.
- [40] N.R. Pal, J.C. Bezdek, R.J. Hathaway, Sequential competitive learning and the fuzzy c-means clustering algorithms, Neural Networks 9 (5) (1996) 787–796.
- [41] T. Shibata, T. Fukuda, T. Kosuge, F. Arai, Skill based control by using fuzzy neural network for hierarchical intelligent control, Proceedings of IJCNN'92, 1992, pp. II-81–II-86.
- [42] T. Takagi, I. Hayashi, NN-driven fuzzy reasoning, International Journal of Approximate Reasoning 5 (1991) 191–212.
- [43] L.-X. Wang, J.M. Mendel, Back-propagation fuzzy system as nonlinear dynamic system identifiers, Proceedings of IEEE International Conference on Fuzzy Systems (1992) 1409–1418.
- [44] H. Yang, T.S. Dillon, Convergence of self-organizing neural algorithms, Neural Networks 5 (1992) 485–493.
- [45] L. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (1) (1973 January) 28–44.



Ren-Jieh Kuo received the MS degree in Industrial and Manufacturing Systems Engineering from Iowa State University, Ames, IA, in 1990 and the PhD degree in Industrial and Management Systems Engineering from the Pennsylvania State University, University Park, PA, in 1994. Currently, he is the Professor and Dean in the College of Management, National Taipei University of Technology, Taiwan, Province of China. His research interests include architecture issues of computational intelligence and their applications in electronic business, logistics, and supply chain management.



Yu-Ting Su received the MS degree from National Taipei University of Technology, Taiwan, Province of China. Currently, he is the engineer of AU Optronics Corporation, Taiwan, Province of China. His research interests are in fuzzy neural networks and their applications in management.



Chui-Yu Chiu is an Associate Professor of Industrial Engineering and Management Department at National Taipei University of Technology. His area of teaching and research interests includes fuzzy theory, artificial intelligence applications, and production management. He received a BS degree from Tunghai University, both MS and PhD degrees in Industrial and Systems Engineering from Auburn University.



Kai-Ying Chen is currently an assistant professor in the Department of Industrial Engineering and Management at National Taipei University of Technology, Taiwan, ROC. He received the B.S., M.S., and Ph.D. degrees all in Mechanical Engineering from National Taiwan University. His research interests are in genetic algorithms and industrial automation.



Fang-Chih Tien is currently a professor in the Department of Industrial Engineering and Management at National Taipei University of Technology, Taiwan, Province of China. He received the BS degree in Industrial Engineering from Chung Yuan Christian University, Taiwan, and the MS and PhD degrees in industrial engineering from the University of Missouri-Columbia, USA. His research interests are in the areas of computer vision inspection systems, combinatorial optimization, correspondence problems, and computer-aided inspection.