

FCMAC-BYY: Fuzzy CMAC Using Bayesian Ying-Yang Learning

M.N Nguyen, D.Shi, Senior Member IEEE, C.Quek

Abstract—As an associative memory neural network model, the Cerebellar Model Articulation Controller (CMAC) has attractive properties of fast learning and simple computation, but its rigid structure makes it difficult to approximate certain functions. This research attempts to construct a novel neural fuzzy CMAC, in which Bayesian Ying-Yang (BYY) learning is introduced to determine the optimal fuzzy sets, and Truth Value Restriction (TVR) inference scheme is subsequently employed to derive the truth-values of the rule weights of implication rules. The BYY is motivated from the famous Chinese ancient Ying-Yang philosophy: everything in the universe can be viewed as a product of a constant conflict between opposites – Ying and Yang, a perfect status is reached when Ying and Yang achieves harmony. The proposed FCMAC-BYY enjoys the following advantages: Firstly, it has a higher generalization ability because the fuzzy rule sets are systematically optimized by BYY; Secondly, it reduces the memory requirement of the network by a significant degree as compared to the original CMAC; Thirdly, it provides intuitive fuzzy logic-reasoning and has clear semantic meanings. The experimental results on some benchmark datasets show that the proposed FCMAC-BYY outperforms the existing representative techniques in the research literature.

Index Terms—Bayesian Ying-Yang learning, CMAC, fuzzy rule set, neural networks, TVR.

I. INTRODUCTION

THE Cerebellar Model Articulation Controller (CMAC) [1, 2], is a type of associative memory neural network that models how a human cerebellum would take inputs, organize its memory and compute outputs. CMAC is a table-lookup module that represents complex and non-linear functions. The associative mapping built into CMAC assures local generalization: similar inputs produce similar outputs while distant inputs produce nearly independent outputs.

The CMAC system has the advantages of fast learning, simple computation, local generalization, and can be realized by specialized high-speed hardware. The application of CMAC can be found in many areas such as robotic control, signal processing, and pattern recognition [3-5]. However, the original CMAC model of Albus has two major disadvantages: the memory requirement grows exponentially with respect to

the number of input variables and difficulty in selecting the memory structure parameters [6, 7]. It is not very efficient in terms of data storage and modeling of problem space. Furthermore, as a trained CMAC is a black box, it is not possible to extract structural knowledge from the system or to incorporate domain expert prior knowledge.

To address the above problems, some researchers have incorporated fuzzy logic into CMAC to obtain a new fuzzy neural system model which is called Fuzzy CMAC, or FCMAC [8-10]. Firstly, the use of fuzzy sets as the input clusters, rather than the crisp sets in the original CMAC, can greatly alleviate the memory requirement. Secondly, fuzzy CMAC can provide a human-like thinking ability which is essential to involve expert-knowledge.

As we know, fuzzy neural networks possess the advantages of both neural networks and fuzzy systems, in the former: learning abilities, optimization characteristics and connectionist structures; and in the latter: human-like thinking and ease of incorporating expert knowledge [11]. Hence, fuzzy CMAC has the learning capabilities of neural networks and the advantages of fuzzy systems which make it more robust, highly intuitive and easily comprehended. Moreover, fuzzy CMAC has the capability of acquiring and incorporating human knowledge into the systems, as well as the capability of processing information based on fuzzy inference rules.

Typically, the fuzzification phase in a fuzzy neural network is fulfilled by clustering the training data in each dimension independently. All the existing clustering algorithms can be divided into two groups: clustering with or without pre-specified cluster number [12-15]. However, all of these conventional methods apply “one way” clustering, that is, they consider only either the forward path of mapping the input data into the clusters, or, the backward path of learning the clusters from the input data. Therefore, they cannot guarantee to achieve an optimal cluster architecture for the input data.

Fuzzy ART [15] introduced by Carpenter et al. is self-organized clustering, which groups a given set of input patterns into some categories. One of the characteristics of Fuzzy ART is the use of vigilance parameter to determine the number of clustered groups depending only on the similarity between all input patterns. Another one is the adjustment of the weight vector of clusters through the learning procedure. However, the number of generated clusters is very sensitive to the predefined vigilance threshold value. Moreover, the sequence of the input patterns also affects the clustering results.

Manuscript received July 18, 2005; revised December 1 2005.

The authors are with the Centre for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: asdmshi@ntu.edu.sg).

Similar to Fuzzy ART, Discrete incremental clustering (DIC) [13] proposed by Tung and Quek has the characteristics of noise tolerance and does not require prior knowledge of the number of clusters present in the training data set. It used the same initial information to create a new cluster for every dimension, but the data distribution is different from dimension to dimension, as a result, DIC cannot obtain an optimal number of clusters.

Lee, Chen and Fu proposed a self-organizing HCMAC neural network [16] which combines a self-organizing input space module and a binary hierarchical CMAC neural network. However, the self-organizing input space module based on Shannon's entropy measure and the golden-section search method cannot guarantee to obtain an optimal input space quantization. More over, the rigid structure of the binary hierarchical CMAC neural network is another disadvantage.

Fuzzification is actually equivalent to identifying the underlying distribution of each dimension of a finite size of the training patterns, so that we can apply it to the unknown data. This research aims to determine the optimal number of fuzzy sets and form clusters in the fuzzification phase to achieve higher generalization ability. In this paper, we propose a novel fuzzy CMAC using Bayesian Ying-Yang learning, hereafter referred to as *FCMAC-BYY*. Bayesian Ying-Yang learning (BYY) [17, 18] is based on the harmony of two representations: the mapping of the input data x into an inner representation cluster y , and the generation of the input data x from an inner representation cluster y . This characteristic allows the proposed fuzzifier to derive the optimal clusters from the input training data. In the inference phase of FCMAC-BYY, the Truth-value restriction (TVR) inference scheme is used to derive the truth-values of the rule weights from the truth-value of the antecedents. BYY, together with TVR, provides the FCMAC-BYY system with an optimal fuzzy rule set, and a consistent rule base, a strong theoretical foundation, more logical and intuitive to the human reasoning process.

The structure of this paper is outlined as follows. The structure of FCMAC-BYY is described in the next section. In section III, a novel Bayesian Ying-Yang fuzzification is proposed. Section IV introduces truth-value restriction inference scheme. Experiments and the detailed analyses are presented in Section V, followed by the conclusions in Section VI.

II. FCMAC-BYY STRUCTURE

The main difference between the FCMAC and the original CMAC is that the association layer in the FCMAC is the rule layer and each associative cell represents a fuzzy rule that links input cluster to the output cluster, so that the input data is first fuzzified into fuzzy clusters before it is fed into the system. As shown in Fig. 1, the FCMAC-BYY neural network can be viewed as a 5-layer hierarchical structure as follows:

(1) *Input Layer X*. This is the layer where the input X is obtained from the raw data (or using sensors for hardware

realization).

(2) *Fuzzified Sensor F*. In this layer, the novel Bayesian Ying-Yang fuzzification is conducted on the input training data set to obtain fuzzy labels. Each neuron (sensor) in this layer represents a particular cluster and a group of sensors is associated with the input variables. Contrary to the binary outputs of the traditional CMAC, the outputs of these sensors are real numbers from 0 to 1, which correspond to their membership values.

(3) *Association Layer A*. This layer is the rule layer and each association cell represents a fuzzy rule. In the case of the lack of available memory, this layer is considered as a conceptual/logical memory space. The AND operation is carried out to ensure that a cell is activated only when all the inputs to it are fired. The weight of fuzzy rules is derived by the truth value restriction scheme in this research.

(4) *Post association Layer P*. To address the problem of a large memory size required in Layer A, it can be mapped to a physical memory space P . This is done by either a linear mapping or hashing [9]. The logical OR operation makes any cell in this layer fired if any of its connected inputs is activated.

(5) *Output Layer O*. This layer is fully connected to layer P . The defuzzification center of area (COA) method [19] is used to compute the output of the structure. The output of FCMAC-BYY is derived by the following equation:

$$y_o = \frac{\sum_{p=1}^P \omega_p \times w_p}{\sum_{p=1}^P \omega_p} \quad \text{for } p=1, 2, \dots, P \quad (1)$$

where ω_p refers to the total matching degree of the antecedent, and w_p is the weight of the p th fuzzy inference rule. The detailed fuzzy inference scheme will be described in Section IV.

[Fig.1 Here]

This research is mainly focused on the fuzzification phase and the fuzzy inference scheme. We proposed a fuzzifier using the novel BYY learning approach to specify the number of fuzzy sets and form clusters in the fuzzification phase. Truth-value restriction (TVR) inference scheme described in section IV is subsequently used to derive the truth-values of the rule weights.

First of all, for each dimension, the proposed BYY

fuzzification in the next section is conducted on the input training data set to identify the fuzzy clusters. Each fuzzy cluster is represented by a neuron in the Fuzzified Sensor Layer. Each combination of the fuzzy cluster in this layer becomes a neuron in the next layer, the Association Layer.

As illustrated in Fig. 2, two dimensions are considered with three fuzzy clusters represented in each dimension in the sensor layer. There are nine combinations of these fuzzy clusters; it means that there are nine cells in the association layer. Each cell represents a fuzzy rule. However, a cell is activated only when all its corresponding fuzzy clusters are activated by the input data. In Fig. 2, only four neurons (w_{12} , w_{22} , w_{13} , w_{23}) are activated by the input data X . The strength of activation or the membership value of the input data and the fuzzy cluster depends on the Euclidean distance between them. In other words, the closer the data point to cluster the higher the membership value.

[Fig.2 Here]

III. FUZZIFICATION USING BAYESIAN YING YANG LEARNING

Constructing a neural network is actually equivalent to finding a solution, y , to represent the input data, x . Treating both x and y as random processes, the joint distribution $p(x,y)$ can be calculated by either of these two formulae:

$$p(x, y) = p(y | x)p(x), \quad (2a)$$

$$p(x, y) = p(x | y)p(y), \quad (2b)$$

However, the result of Equation (2a) is not equal to that of Equation (2b) unless y is the optimal solution. Notice that x and y are dialectical: Firstly, x is visible but y is invisible. Secondly, x decides y in training but y decides x in running. This interesting phenomenon fits well with the famous Chinese ancient Ying-Yang philosophy: First, everything in the universe can be viewed as a product of a constant conflict between opposites – Ying and Yang, with Ying referring to negative, female and invisible, whereas Yang referring to positive, male and visible. Second, the optimal status is reached if Ying and Yang achieves harmony.

In this section, the fuzzifier using Bayesian Ying-Yang [17, 18] is employed to automatically construct the fuzzy set for each dimension. As shown in Fig. 3, the BYY fuzzification

system considers two complementary representation of the joint distribution of input pattern x and fuzzy cluster y .

The first one is the forward/training model $p(y) = \int p(y | x)p(x)dx$. It is called a Yang/(visible) model which focuses on the mapping function of the visible input data x into an invisible cluster representation y via a forward propagation distribution $p(y|x)$. In this process, the input data x are visible and be considered as known, whereas the clusters y are invisible and are considered as unknown. By this model, the given input data is transferred into unknown fuzzy clusters. The process is regarded as an unsupervised learning process.

The second one is the backward/running model $p(x) = \int p(x | y)p(y)dy$. It is called a Ying/(invisible) model which focuses on the generation function of the visible input data x from an invisible cluster representation y via a backward propagation distribution $p(x/y)$. In this process, clusters y are visible and be considered as known, whereas the input data x are invisible and are considered as unknown. By this model, the input data is generated from the constructed fuzzy clusters. The process is regarded as a supervised learning process.

[Fig.3 Here]

Under the principle of Ying-Yang harmony, the difference between the two Bayesian representations in (2a) and (2b) should be minimized. Thus, the trade-off between the forward/training model and the backward/running model is optimized. It means that the input data is well mapped into the clusters and at the same time the clusters also adequately cover the input data space. Therefore, the entire BYY fuzzification system is of the least complexity. In other words, the proposed FCMAC-BYY has the highest generalization ability when the harmony of two Ying and Yang models is achieved.

In this research, the Gaussian membership function is used, and the joint probability density of the j th dimension that consists of K^j Gaussians is described by (3).

$$p(x_i^j, \Theta^j) = \sum_{y=1}^{K^j} \alpha_y^j G(x_i^j, m_y^j, \sigma_y^j) \quad (3)$$

and the Gaussian density function is given as follows:

$$G(x_i^j, m_y^j, \sigma_y^j) = \exp \left[-\frac{1}{2} \left(\frac{x_i^j - m_y^j}{\sigma_y^j} \right)^2 \right] \quad (4)$$

where x^j is the input data of the j th dimension, $\Theta^j = \{\theta_y^j \equiv \alpha_y^j, m_y^j, \sigma_y^j\}_{y=1}^{K^j}$ is a set of finite mixture model parameter, α_y^j is the prior probability, m_y^j and σ_y^j refer to the mean value and the width of the y th cluster in the j th dimension. These parameters can be estimated by the maximum likelihood (ML) learning with the EM algorithm [20, 21] based on a given data set $D = \{x_i\}_{i=1}^N$.

Fuzzification using BYY involves two phases, namely: parameter learning and cluster number selection [22]. Parameter learning does the task of determining all the unknown parameter Θ^j for a specific value of cluster K^j . Then, a cluster number selection is made to select the optimal number of cluster K^{j*} from a collection of specific BYY systems with different value of cluster K^j .

In the first phase, the Kullback-Leibler divergence [23] is used to evaluate the difference of the joint probability between Ying and Yang:

$$\min_{\Theta} KL(\Theta) = \int \int P(y|x)P(x) \ln \frac{P(y|x)P(x)}{P(x|y)P(y)} dx dy \quad (5)$$

The minimization of the above Kullback-Leibler divergence will produce the optimal parameter Θ^{j*} at each value of cluster K^j . The learning procedure can be implemented by the following iterative Expectation-Maximization (EM) algorithm:

E-step:

$$P(y^j | x_i^j) = \frac{\alpha_y^j G(x_i^j, m_y^j, \sigma_y^j)}{\sum_{y=1}^{K^j} \alpha_y^j G(x_i^j, m_y^j, \sigma_y^j)} \quad (6)$$

M-step:

$$\alpha_y^{j(new)} = \frac{1}{N} \sum_{i=1}^N \frac{\alpha_y^j G(x_i^j, m_y^j, \sigma_y^j)}{\sum_{y=1}^{K^j} \alpha_y^j G(x_i^j, m_y^j, \sigma_y^j)} = \frac{1}{N} \sum_{i=1}^N P(y^j | x_i^j) \quad (7)$$

$$m_y^{j(new)} = \frac{\sum_{i=1}^N P(y^j | x_i^j) x_i^j}{\sum_{i=1}^N P(y^j | x_i^j)} = \frac{1}{\alpha_y^j N} \sum_{i=1}^N P(y^j | x_i^j) x_i^j \quad (8)$$

$$\sigma_y^{j(new)} = \frac{1}{\alpha_y^j N} \sum_{i=1}^N P(y^j | x_i^j) (x_i^j - m_y^j)^2 + (h^j)^2 \quad (9)$$

where h^j is the smoothing parameter, and be updated as

follows:

$$h_{new}^j = h_{old}^j + \eta g(h_{old}^j), \quad (10)$$

where η is a step length constant and

$$g(h_{old}^j) = \frac{1}{h_{old}^j} - h_{old}^j \sum_{j=1}^{K^j} \frac{\alpha_y^j}{\sigma_y^j} - \frac{\sum_{i=1}^N \sum_{l=1}^N \gamma_{il}^j (x_i^j - x_l^j)^2}{(h_{old}^j)^3} \quad (11)$$

with

$$\gamma_{il}^j = \frac{\exp \left[-\frac{1}{2} \left(\frac{x_i^j - x_l^j}{h_{old}^j} \right)^2 \right]}{\sum_{i=1}^N \sum_{l=1}^N \exp \left[-\frac{1}{2} \left(\frac{x_i^j - x_l^j}{h_{old}^j} \right)^2 \right]} \quad (12)$$

In the second phase, the optimal number of fuzzy cluster K^{j*} is determined by a cluster number selection [17] as follows:

$$K^{j*} = \arg \min_{K^j} J(K^j) \quad (13)$$

$$J(K^j) = \sum_{y=1}^{K^j} \left(\frac{1}{2} \alpha_y^{j*} \ln \sigma_y^{j*} + \frac{1}{2} \frac{(h^j)^2}{(\sigma_y^{j*})^2} - \alpha_y^{j*} \ln \alpha_y^{j*} \right) \quad (14)$$

where θ_y^{j*} and h^j are the results of the parameter learning in the first phase.

In practical, for each dimension j , we start with $K^j=1$, estimate the parameter Θ^j by the EM algorithm based on the given training data, and compute $J(K^j)$. Then, we proceed to $K^j \rightarrow K^j+1$, and compute $J(K^j)$ again. After we gather a series of $J(K^j)$, the optimal cluster number, K^{j*} , is selected from the one with minimal $J(K^j)$.

IV. TRUTH VALUE RESTRICTION INFERENCE SCHEME

Truth value restriction method [24, 25] offers a consistent rule base, a strong theoretical foundation and is more logical and intuitive to the human reasoning process as compared to other alternative techniques such as compositional rule of inference scheme (CRI) [26, 27], Approximate Analogical Reasoning Schema (AARS) [28, 29].

Truth-value restriction uses implication rules to derive the truth-values of the consequents from the truth-value of the antecedents. In the TVR methodology, the degree to which the actual given value of A' of a variable x agrees with the antecedent value A in the proposition "IF x is A THEN y is B " is represented as a fuzzy subset of truth space $\tau_{AA'}$. This fuzzy subset of truth space, known as truth-value restriction, is used in a fuzzy deduction process to determine the corresponding restriction on the truth-value of the proposition "y is B " $\tau_{BB'}$. The latter truth value restriction is then 'inverted', which means that a fuzzy proposition "y is B " in the Y universe of discourse is found such that its agreement with "y is B " is

equal to the truth value restriction derived by the fuzzy inference process as Fig. 4.

[Fig.4 Here]

The fuzzy subset $\tau_{AA'}$ of the truth space is given as follows:

$$\tau_{AA'}(a) = \begin{cases} \sup_x \{\mu_{A'}(x) | x \in \mu_A^{-1}(a)\}, & \mu_A^{-1}(a) \neq \phi \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

where μ_A and $\mu_{A'}$ are the respective membership functions of the fuzzy sets A and A' defined on the universe of discourse X ; x denotes a value in X ; a is the membership value of x in fuzzy set A ; $\mu_A^{-1}(a)$ is the set of values of x in X that take membership value a in the fuzzy set A ; and ϕ denotes the empty set or null set.

The truth-value $\tau_{BB'}$ of the consequent “ y is B ” can be computed using (16):

$$\tau_{BB'}(b) = \sup_a \{m_I[\tau_{AA'}(a), I(a, b)]\} \quad (16)$$

where m_I is the forward reasoning function (usually a T-norm operation); and I is the implication rule.

The truth-value $\tau_{BB'}$ is subsequently “inverted” to determine the inferred conclusion. That is, a fuzzy proposition “ y is B' ” is computed using the *truth function modification* (TFM) process such that the degree of possibility that the proposition “ y is B ” is true given “ y is B' ” is described by the truth-value $\tau_{BB'}$. That is, the derivation of the fuzzy set B' from the truth-value $\tau_{BB'}$ is performed using (17):

$$\begin{aligned} B' &= \tau_{BB'} \circ B \\ \Rightarrow \underline{b'} &= \tau_{BB'}(\underline{b}) \\ &\mu_{B'}(y) \quad \mu_B(y) \\ \Rightarrow \mu_{B'}(y) &= \tau_{BB'}(\mu_B(y)) \end{aligned} \quad (17)$$

where b and b' are the respective truth-values of the propositions “ y is B ” and “ y is B' ”; and μ_B and $\mu_{B'}$ are the membership functions of the fuzzy sets B and B' respectively.

In the proposed model, the computed truth-values of the antecedents can be effectively propagated in the hybrid structure of a neural fuzzy system. The truth-value of the proposition in the antecedent is calculated and allowed to

propagate through the network. It is this value that is used to compute the proposition in the consequent. This treatment makes the TVR a viable inference scheme for implementation in a neural fuzzy system. Given the input-output data $(x^o, y^o) = (x_1^o, x_2^o, \dots, x_n^o, y^o)$, the simplified fuzzy inference rules are shown by the following implications:

$$\begin{aligned} \text{Rule}^p &= \text{If } x_1 \text{ is } A_1^{k^1}, x_2 \text{ is } A_2^{k^2}, \dots, x_n \text{ is } A_n^{k^n} \\ &\text{Then } y \text{ is } w_p \end{aligned} \quad (18)$$

where $p=1, 2, \dots, P$, x_j is the j th input variable, y is an output variable, P is the number of fuzzy rules and w_p is the weight of the p th fuzzy inference rule. While $A_j^{k^j}$ is the k th fuzzy cluster in the j th dimension, $k^j=1, 2, \dots, K^j$. The total fuzzy cluster number K^j of each dimension is obtained by BYY fuzzification. Given the input data x^o , the total matching degree ω_p of the antecedent of the p th rule using TVR inference:

$$\omega_p = \prod_{j=1}^n \mu_{A_j^{k^j}}(x_j^o), \quad \text{for } j=1, 2, \dots, n \quad (19)$$

where $\mu_{A_j^{k^j}}(x_j^o)$ is a membership value of the input data x to the k th cluster in the j th dimension.

V. EXPERIMENTAL RESULTS

We are now in a position to compare the performance of FCMAC-BYY with other representative fuzzy neural networks like Falcon-ART, Falcon-MART and GenSoFNN-CRI(S). Our experiments were conducted on three benchmark data sets: Iris data, two- spiral problem and bank failure classification. The hardware configuration for our experiments is: CPU = Intel Pentium IV 2.2GHz, operating system = Microsoft Window 2000, memory available = 512 Mbytes.

A. Iris Classification

The Fisher’s Iris data set [30] consists of 150 instances of Iris flowers belonging to three classes; namely: Sentosa (class 1), Virginica (class 2) and Versicolor (class 3). There are 50 instances for each of the three classes. Each instance of the flowers consists of four physical attributes. They are sepal length, sepal width, petal length, and petal width. Fig. 5 shows the data distribution of these four dimensions in Iris data, and we can see that they are quite different from dimension to dimension.

The training set consists of 35% of the data, and there are 17 instances from each of the three classes. The test set contains the remaining 65% of the data points and consists of 99 data points. The training and testing sets are randomly

generated. The experiment results are cross validated by using three different groups of training and test sets. They are denoted as CV1, CV2, and CV3. The desired outputs of the network are 0, 1 and 2 for classes 1, 2 and 3 respectively.

[Fig.5 Here]

Experimental results are recorded in term of two classes: *True Acceptance* and *False*. The *True Acceptance (TA)* refers to the total number of correctly classified instances, while the *False* refers to all misclassified instances. The *False class* is subdivided into *False Acceptance class1 (FA₁)*, *False Acceptance class2 (FA₂)*, *False Acceptance class3 (FA₃)*, i.e. wrong instances classified into class 1, class 2, class 3, respectively, and *False Rejection (FR)*, i.e. unclassified instances. The formulas are given as follows:

$$TA = \frac{N_{ins_corr}}{N_{ins_total}} \times 100\% \quad FA_1 = \frac{N_{class1_wrong}}{N_{mis_ins}} \times 100\%$$

$$FA_2 = \frac{N_{class2_wrong}}{N_{mis_ins}} \times 100\% \quad FA_3 = \frac{N_{class3_wrong}}{N_{mis_ins}} \times 100\%$$

$$FR = \frac{N_{un_class}}{N_{mis_ins}} \times 100\%$$

Here, N_{ins_corr} refers to the number of correct instances classified,

N_{ins_total} refers to the total instances,

N_{class1_wrong} , N_{class2_wrong} , N_{class3_wrong} , N_{un_class} refer to the wrong instances classified into class 1, class 2, class 3, and unclassified instances respectively,

N_{mis_ins} refers to the number of wrong instances classified.

[Table I Here]

From Table I, one can observe that the *False Acceptance class1* is almost zero, while the *False Acceptance class2* and *False Acceptance class3* have high values. It means that most of the wrong instances are classified into classes 2 and 3. This is consistent with the claim made by some authors that while

class 1 is well separated from the other two, classes 2 and 3 have significant degree of overlap [31].

In Table II, average classification rate is used to evaluate the performance of the FCMAC-BYY across the three data groups. The experimental results of the proposed FCMAC-BYY network for Iris classification are compared against FCMAC using DIC [13]. Both models use Gaussian distribution as the membership functions.

[Table II Here]

Compared against the results of the FCMAC using DIC, the FCMAC-BYY achieves higher accuracy with significantly fewer fuzzy rules or neurons. Table II indicates that the self-organizing BYY requires less fuzzy clusters than DIC. For example, for CV1, the self-organizing BYY derives 2:1:2:3 fuzzy clusters for the four dimensional input data respectively, while DIC derives 3:3:6:4 fuzzy clusters respectively. In the third dimension, DIC requires 6 or 7 fuzzy clusters, whereas 2 clusters identified by the self-organizing BYY are sufficient. In other words, the self-organizing BYY significantly simplifies the network with higher generalization ability by judiciously optimizing fuzzy rule set by the Ying-Yang approach.

The above results can be explained as follows. Like most of the clustering techniques, DIC is very sensitive to the initial values. DIC simply applies the same values to initialize new clusters for all dimensions; however the data distributions are different from dimension to dimension as shown in Fig. 5. This drawback of DIC leads to more clusters. BYY fuzzification determines the optimal centroids and widths for each particular number of clusters, and the optimal cluster number for each dimension is then computed by (13). However, BYY fuzzification demands a higher computation cost for such an optimization.

Furthermore, BYY needs more computation time for each epoch than other methods due to its optimization. From Table III, we can see that the average execution time of FCMAC-BYY is higher than FCMAC-DIC; however it is much smaller than that of Falcon-ART. Hence, the computational cost of BYY is still acceptable.

[Table III Here]

Further comparison of the proposed FCMAC-BYY with other architectures is listed in Table III. These architectures include variations of Falcon network, namely Falcon-ART [32], Falcon-MART [33] and GenSoFNN-CRI(S) [13]. The average classification rate for FCMAC-BYY is 96.60%, which is the highest amongst different architectures. In addition, FCMAC-BYY has a small number of epochs and a less complexity structure due to the optimal clusters obtained by BYY classification.

B. Two-Spiral Problem

The two-spiral problem was proposed by K.J. Land on the connectionist mailing list as an interesting benchmark task for neural networks [34]. The problem involves the correct classification of two intertwined spirals. For the evaluation of the proposed FCMAC-BYY, the training set is the standard spiral data consisting of 194 points, with 97 points for each spiral as shown in Fig. 6. The test set consists of two dense spirals with 385 points each and is generated using (20) to (22).

[Fig.6 Here]

The formulation for the generation of the two-spiral training data is given as:

$$\text{spiral 1: } \begin{cases} x = \gamma \cos \theta \\ y = \gamma \sin \theta \end{cases} \quad (20)$$

$$\text{spiral 2: } \begin{cases} x = -\gamma \cos \theta \\ y = -\gamma \sin \theta \end{cases} \quad (21)$$

where

$$\gamma = \frac{1}{\pi} \left(\theta + \frac{2}{\pi} \right), \quad \theta = \frac{k\pi}{16}, \quad k = 0, 1, 2, \dots, 96 \quad (22)$$

Lang and Witbrock [34] reported that a conventional feed-forward back-propagation neural network could not resolve this problem. Hence, they employed back-propagation neural networks with shortcut connections as pathways for providing information to all parts of the network to eliminate the problem of attenuated error signals when back-propagating through the layers. The proposed network is a special network with a 2-5-5-5-1 structure that has shortcut connections, with each node being connected to all nodes in all subsequent layers. With one additional bias weight for each node, it has 138 trainable weights. When the weights of the specialized network are updated using vanilla back-propagation, training

required an average of 20,000 training epochs.

Carpenter et al. [35] also evaluated the fuzzy ARTMAP system with 2-spiral data set. They used the most stringent criteria to train the fuzzy ARTMAP to obtain 100% classification for the dense spiral. As a result, the fuzzy ARTMAP creates 194 ART categories for the standard 2-spiral data set that contains 194 points, which essentially degenerate to a pure one-one memory recall.

Tung and Quek [13] proposed a discrete incremental clustering (DIC) technique based GenSoFNN-CRI(S) and reported that is able to achieve 100% average classification rate for both standard and dense spirals with 23 fuzzy sets in each of the two input dimensions. A total of 156 fuzzy rules are created.

[Table IV Here]

Table IV shows the comparison between different methods. Using the spiral data as training set, FCMAC-BYY achieved 100% classification for both training set and test set with 20 fuzzy clusters. Whereas, ARTMAP needs 194 points and GenSoFNN-CRI(S) requires 23 fuzzy sets.

C. Bank failure classification

The financial variables (covariates) used in the bank failure prediction application are extracted from the Call Reports, which are downloaded from the website of Federal Reserve Bank, Chicago [36, 37]. The observation period of the survived banks consists of 21 years from January 1980 to December 2000. There are 702 failed banks and 2933 survived banks over the observation period, leading to a total of 3635 banks. Based on statistical investigation, only nine variables shown in Table V are selected according to their significance and correlation.

[Table V Here]

Three different scenarios of experiments are conducted:

- ❖ Bank failure classification based on the last available financial record.
- ❖ Bank failure prediction using financial records one year prior to the last one.
- ❖ Bank failure prediction using financial records two years prior to the last one.

In each scenario, the data set is split into different training sets and test sets. The training sets contain 20% of the data while the test sets contain the remaining 80%. There are five cross-validation groups, denoted as CV1, CV2, CV3, CV4 and CV5 respectively. The original data set is initially split into two pools: failed and survived banks. For each cross-validation group, the training set is randomly selected from two pools so that the number of survived and failed bank is equal. It is called a “balance” training scenario. The training sets of the five groups are mutually exclusive.

One output is used to differentiate between failed and survived banks. Failed banks are denoted with output “0” while survived (non-failing) banks are identified by output “1”. The Ying-Yang FCMAC network is subsequently used to model the inherent relationships between the financial covariates and their impact on the financial solvency of the respective banks. The Ying-Yang FCMAC network is trained using the data instances in the training set and the modeling capability of the trained network is subsequently evaluated using the test set. The simulation is repeated for all the five cross-validation groups. The classification threshold (to discern between failed and survived (non-failing) banks based on the nine input financial covariates) is varied to obtain the receiver-operating-characteristic (ROC) curves depicted in Fig. 7. Type I error is defined as the error of classifying a failed bank as a survived (non-failing) one whereas Type II error is the classification of a non-failing bank as a failed bank. The EER line denotes the case of equal error rates (EER), where the Type I equals Type II errors.

[Fig.7 Here]

From Fig. 7, one can observe that the Ying-Yang FCMAC outperforms to FCMAC-DIC on all three scenarios. The average classification rate for the failed banks of the FCMAC-DIC using the last financial statements is about 90% and it goes down to around 85% with statements obtained one year prior to the last record and subsequently to about 81% with financial statements two years prior to the last available record. Whereas, the values of Ying-Yang FCMAC only deteriorate from 94% in the first scenario to around 92% in the second scenario and subsequently to about 91% in the last scenario.

Table VI shows the comparison on bank failure classification of GenSoFNN-CRI(S) [37], FCMAC-DIC and FCMAC-BYY. One can observe that the FCMAC-BYY has superior performance to both GenSoFNN-CRI(S) and

FCMAC-DIC. Noticing that, the classification rate degrades with respect to the prediction period. The longer the prediction period, the less accurate are the classification and prediction result. However, FCMAC-BYY is less sensitive to the prediction period than the others. Its average classification rate decreases from 94.53% of the last record to 91.71% of the 2 years prior. Whereas, the average classification rate of GenSoFNN-CRI(S) drops from 90.86% of the last record to 72.19% of the 2 years prior.

[Table VI Here]

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a novel fuzzy FCMAC using Bayesian Ying-Yang learning named FCMAC-BYY is proposed. The experiments are conducted on three benchmark datasets and we compared the classification performance of our proposed model with those of representative neural fuzzy systems such as FALCON-ART, Falcon-MART, and GENSOFNN-CRI(S). The experimental results suggest that our proposed model is able to achieve the same accuracy but has simpler structure with higher generalization capability.

The advantage of FCMAC-BYY is accrued from its fuzzification technique using Bayesian Ying-Yang learning, which obtains Gaussian clusters from a set of raw training data as fuzzy rules. The BYY requires no prior knowledge of the number of clusters and initial information, and provides CMAC network with a concise fuzzy rule base. Another advantage of FCMAC-BYY is the truth-value restriction inference scheme, which provides FCMAC with an intuitive fuzzy logic-reasoning framework. This feature is particularly important in some areas such as financial analysis, medical diagnosis when the domain expert must be involved to analyze the causality. This integration and the realization of the BYY within a logical fuzzy framework is a significant step in the generation of simple and elegant fuzzy neural framework with localized recall and it is able to achieve a high level of generalization using a concise description.

Our future work includes the investigation of possibility-based Ying-Yang learning. On the one hand, the current probability-based BYY cannot provide semantic meaning. On the other hand, in modeling uncertain judgment such as fuzzification, it seems natural not to wish to rigidify the relationship between the indications and those that weigh against them. In this respect, the notion of probability seems less flexible than that of possibility.

APPENDIX

The Kullback Leibler distance (KL-distance) is a natural distance function from a "true" probability distribution, p , to a "target" probability distribution, q . It can be interpreted as the expected extra message-length per datum due to using a code based on the wrong (target) distribution compared to using a code based on the true distribution [23].

For discrete (not necessarily finite) probability distributions, $p=\{p_1, \dots, p_n\}$ and $q=\{q_1, \dots, q_n\}$, the KL-distance is defined to be

$$KL(p,q) = \sum_i p_i \ln \frac{p_i}{q_i}$$

For continuous probability densities, the sum is replaced by an integral as follows:

$$KL(p,q) = \int p(x) \ln \frac{p(x)}{q(x)} dx$$

ACKNOWLEDGMENT

The authors would like to thank Professor Lei Xu for his advice on Bayesian Ying-Yang learning theory.

REFERENCES

- [1] J. S. Albus, "Data storage in the cerebellar model articulation controller (CMAC)," *Transaction of the ASME, Dynamic Systems Measurement and Control*, vol. 97, no. 3, pp. 228-233, 1975.
- [2] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Transaction of the ASME, Dynamic Systems Measurement and Control*, vol. 97, no. 3, pp. 220-227, 1975.
- [3] J. Hu, J. Pratt, and G. Pratt, "Stable adaptive control of a bipedal walking; Robot with CMAC neural networks," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1050-1056, 1999.
- [4] W. T. Miller and F. H. Glanz, "CMAC: An associative neural network alternative to backpropagation," *Proceedings of the IEEE, Special Issue on Neural Networks, II*, vol. 78, pp. 1561-1567, October, 1990.
- [5] F. H. Glanz, W. T. Miller, and L. G. Kraft, "An overview of the CMAC neural network," *IEEE Conference on Neural Networks for Ocean Engineering, Washington, DC*, pp. 301-308, 1991.
- [6] J. Hu and F. Pratt, "Self-organizing CMAC neural networks and adaptive dynamic control," *IEEE International Conference on Intelligent Control, Cambridge, MA*, pp. 15-17, September, 1999.
- [7] A. Menozzi and M.-Y. Chow, "On the training of a multi-resolution CMAC neural network," *Proceedings of IECon'97, New Orleans, LA*, vol. 3, pp. 1201-1205, 1997.
- [8] J. Ozawa, I. Hayashi, and N. Wakami, "Formulation of CMAC-Fuzzy system," *IEEE international Conference on Fuzzy Systems, San Diego, CA*, pp. 1179-1186, 1992.
- [9] M. N. Nguyen, D. Shi, and C. Quek, "Self-Organizing Gaussian fuzzy CMAC with Truth Value Restriction," *Proceedings of IEEE International Conference of Information Technology and Applications (ICITA), Sydney, Australia*, 2005.
- [10] H. Xu, C. M. Kwan, L. Haves, and J. D. Pryor, "Real-time adaptive on-line traffic incident detection," *Fuzzy Sets and System*, pp. 173-183, 1998.
- [11] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*. England; New York: John Wiley, Chichester, 1997.
- [12] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press, 1981.
- [13] W. L. Tung and C. Quek, "GenSoFNN: A Generic Self-Organizing Fuzzy Neural Network," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, September 2002.
- [14] G. Leng, G. Prasad, and T. M. McGinnity, "An on-line algorithm for creating self-organizing fuzzy neural networks," *Neural Networks*, vol. 17, pp. 1477-1493, 2004.
- [15] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759-771, 1991.
- [16] H. M. Lee, C. M. Chen, and Y. F. Lu, "A self-organizing HCMAC neural-network classifier," *IEEE Transactions on neural networks*, vol. 14, no.1, pp. 15-27, 2003.
- [17] L. Xu, "BYY harmony learning, structural RPCL, and topological self-organizing on mixture models," *Neural Networks*, vol. 15, pp. 1125-1151, 2002.
- [18] L. Xu, "Advances on BYY harmony learning: information theoretic perspective, generalized projection geometry, and independent factor autodetermination," *IEEE Transactions on Neural Networks*, vol. 15, pp. 885-902, 2004.
- [19] E. S. Lee and Q. Zhu, *Fuzzy and Evidence Reasoning*: Physica-Verlag, 1995.
- [20] N. M. Laird, A. P. Dempster, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society*, vol. B39, pp. 1-38, 1977.
- [21] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the em algorithm," *SIAM Rev.*, vol. 26, pp. 195-239, 1984.
- [22] L. Xu, "Bayesian Ying Yang Learning (II): A New Mechanism for Model Selection and Regularization," *Intelligent Technologies for Information Analysis, N. Zhong and J. Liu (eds), Springer*, pp. 661-706, 2004.
- [23] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 2(1), pp. 79-86, March, 1951.
- [24] C. T. Lin and C. S. G. Lee, "A Neuro-Fuzzy Synergism to Intelligent Systems," *Neural Fuzzy Systems, Upper Saddle River, NJ: Prentice-Hall*, 1996.
- [25] C. Quek and R. W. Zhou, "POPFNN: A Pseudo Outer-Product Based Fuzzy Neural Network," *IEEE Transaction on Neural Networks*, vol. 9, pp. 1569-1581, 1996.
- [26] K. K. Ang, C. Quek, and M. Pasquier, "POPFNN-CRI(S): Pseudo Outer Product based Fuzzy Neural Network using the Compositional Rule of Inference and Singleton Fuzzifier," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 33, 2003.
- [27] L. A. Zadeh, "Calculus of fuzzy restrictions," *Fuzzy sets and Their Applications to Cognitive and Decision Processes, Edition: New York: Academic*, pp. 1-39, 1975.
- [28] Turksen, I. B. and Z. Zhong, "An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets," *Fuzzy Sets System*, vol. 34, pp. 323-346, 1990.
- [29] C. Quek and R. W. Zhou, "POPFNN-AARS(S): A Pseudo Outer-Product Based Fuzzy Neural Network," *IEEE Transaction on Systems, Man & Cybernetics*, vol. 29, pp. 859-870, 1999.
- [30] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann Eugenics 7, Part II*, pp. 179-188, 1936.
- [31] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
- [32] C. J. Lin and C. T. Lin, "An ART-Based Fuzzy Adaptive Learning Control Network," *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 477-496, 1997.
- [33] C. Quek and W. L. Tung, "A novel approach to the derivation of fuzzy membership functions using the Falcon-MART architecture," *Pattern Recognition Letters*, vol. 22, pp. 941-958, 2001.
- [34] K. J. Lang and M. J. Witbrock, "Learning to tell two spirals apart," presented at Proceeding of the 1988 Connectionist Models Summer School, Carnegie Mellon Universit, 1998.
- [35] C. A. Carpenter, S. Grossberg, and et, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps," *IEEE Transactions on Neural Networks*, vol. 3, pp. 698-713, 1992.
- [36] "Repository for bank data (Online). Available: Federal Reserve Bank of Chicago. URL <http://www.chicagofed.org>."
- [37] W. L. Tung, C. Quek, and P. Cheng, "GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures," *Neural Networks*, vol. 17, pp. 567-587, 2004.



Minh Nhut Nguyen received the BEng degree and M.Phil degree in Computer Engineering from Ho Chi Minh City University of Technology, Vietnam in 2001 and 2005 respectively. He is currently doing a Ph. D. at School of Computer Engineering, Nanyang Technological University, Singapore. His research

interests include machine learning, fuzzy sets theory, pattern recognition and neural networks.



Daming Shi (M'02-SM'04) received the PhD degree in mechanical control from Harbin Institute of Technology, China, and the PhD degree in computer science from University of Southampton, United Kingdom.

He has been serving as an Assistant Professor in Nanyang Technological University in Singapore since 2002. His current research interests include machine learning, medical image processing, pattern recognition and neural networks. Dr. Shi is a co-chair of the technical committee on Intelligent Internet System, IEEE Systems, Man and Cybernetics Society.



C. Quek received the B.Sc. degree in electrical and electronics engineering and the Ph.D. degree in intelligent control from Heriot Watt University, Edinburgh, Scotland.

He is an associate professor and a member of the Centre for Computational Intelligence (C^2I), formerly the Intelligent Systems Laboratory, School of Computer Engineering, Nanyang Technological University. His research interests include intelligent control, intelligent architectures, AI in education, neural networks, fuzzy systems, fuzzy rule-based systems, and genetic algorithms, neurocognitive computational architectures, semantic learning memory systems.

TABLE I
EXPERIMENTAL RESULTS OF FCMAC-BYY ON IRIS CLASSIFICATION

Experiment t	(TA)	FA₁	FA₂%	FA₃%	FR%
CV1	96.67	0.00	44.44	11.11	44.44
CV2	96.67	0.00	57.14	42.85	0.00
CV3	97.33	0.00	44.44	44.44	11.11

TABLE II
COMPARISON OF NUMBER OF CLUSTERS OBTAINED BY BYY AND DIC

	FCMAC-BYY		FCMAC using DIC	
	Classification rate%	Clusters numbers of four dimensions	Classification rate%	Clusters numbers of four dimensions
CV1	97.97	$2 \times 1 \times 2 \times 3 = 12$	96.6	$3 \times 3 \times 6 \times 4 = 216$
CV2	95.96	$2 \times 1 \times 2 \times 3 = 12$	96	$3 \times 4 \times 6 \times 3 = 216$
CV3	95.96	$2 \times 1 \times 2 \times 3 = 12$	95	$2 \times 2 \times 7 \times 3 = 84$

TABLE III
 COMPARISON OF TRAINING EPOCHS, AVERAGE CLASSIFICATION RATE AND
 CPU TIMINGS OF VARIOUS NETWORKS ON IRIS DATASET

Networks	Training epochs	Average classification rate (%)	CPU time (s)
Falcon-ART	1000	75.76	98.75 \pm 8.57
Falcon-MART	11	94.95	1.46 \pm 0.39
FCMAC-DIC	20	95.87	0.32 \pm 0.08
GenSoFNN-CRI(S)	25	96.03	3.91 \pm 1.24
FCMAC-BYY	10	96.60	6.72 \pm 1.36

TABLE IV
CLASSIFICATION RESULTS IN 2-SPIRAL PROBLEM

Networks	Training Set (194 points)	Test Set (770 points)
Lang's 2-5-5-5-1 structure	100%	92.80%
Fuzzy ARTMAP	100%	100%
GenSoFNN-CRI(S)	100%	100%
FCMAC-DIC	100%	99.87%
FCMAC-BYY	100%	100%

TABLE V
DEFINITION OF COVARIATES (NUMBERS IN BRACKETS ARE THE
IDENTIFICATION OF THE DATA ELEMENTS FROM THE CALL REPORTS)

CAMEL category	Covariates
Capital adequacy	CAPADE Average total equity capital (3210) / average total assets (2170)
Asset (loan) quality	OLAQLY Average (accumulated) loan loss allowance (3123) / average total loans and leases, gross (1400) PROBLO Average (accumulated) loans 90 + days late (1407) / average total loans and leases, gross (1400) PLAQLY (Annual) loan loss provisions (4230) / average total loans and leases, gross (1400)
Management	NIEOIN Non-interest expense (4093) / operating income (4000)
Earnings	NINMAR Total interest income (4107) - interest expense (4073) / average total assets (2170) ROE Net income (after tax) (4340) + applicable income taxes (4302) / average total equity capital (3210)
Liquidity	LIQUID Average cash (0010) + average federal funds sold (1350) / average total deposits (2200) + average fed funds purchased (2800) + average banks' liability on acceptances (2920) + average other liabilities (2930)
Miscellaneous	GROWLA Total loans and leases, gross (1400) _t - total loans and leases, gross (1400) _{t-1} / total loans and leases, gross (1400) _{t+1}

TABLE VI
 COMPARISON ON BANK FAILURE CLASSIFICATION WITH 9 INPUTS OF
 GENSoFNN-CRI(S) (CRI), FCMAC-DIC(DIC) AND FCMAC-BYY(BYY)

	Last record			1 year prior			2 years prior		
	CRI (%)	DIC (%)	BYY (%)	CRI (%)	DIC (%)	BYY (%)	CRI (%)	DIC (%)	BYY (%)
CV1	92.92	90.61	93.95	82.26	90.08	93.23	68.29	82.34	91.73
CV2	90.62	92.27	94.96	81.13	83.22	92.77	68.29	82.18	91.81
CV3	94.37	91.63	94.48	87.55	89.10	92.64	81.95	83.93	91.19
CV4	86.04	89.80	94.64	76.6	86.38	93.53	74.15	79.77	92.06
CV5	95.21	90.01	94.60	86.79	85.77	92.77	68.29	77.92	92.14
Average Classification rate (%)	91.83	90.86	94.53	82.87	86.91	92.99	72.19	81.23	91.71

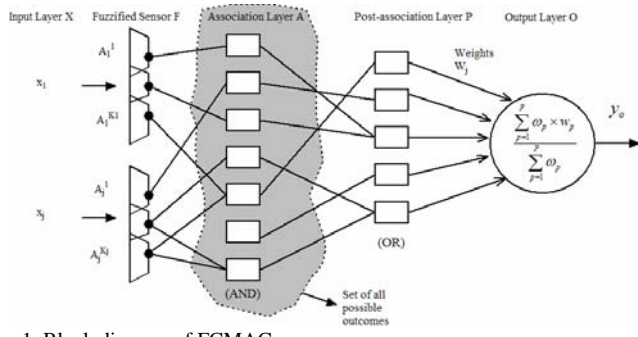


Fig. 1. Block diagram of FCMAC

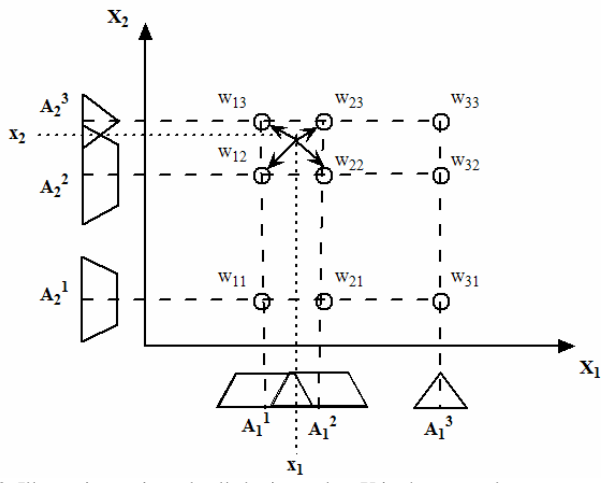


Fig. 2. Illustration activated cells by input data X in the sensor layer.

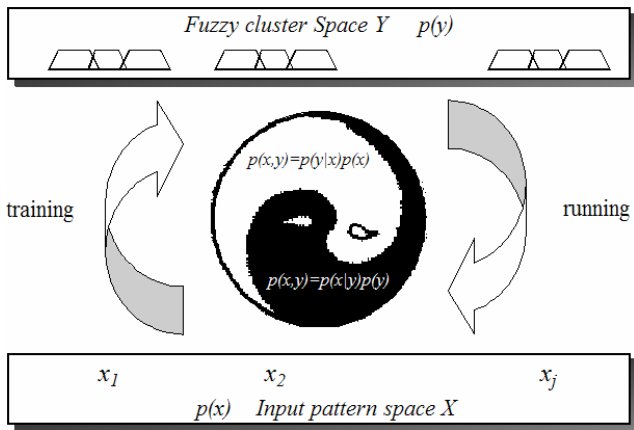


Fig. 3. Neural network construction VS Bayesian Ying-Yang harmony

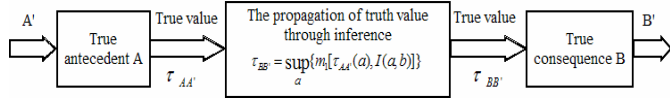


Fig. 4. Truth value restriction method in fuzzy inference

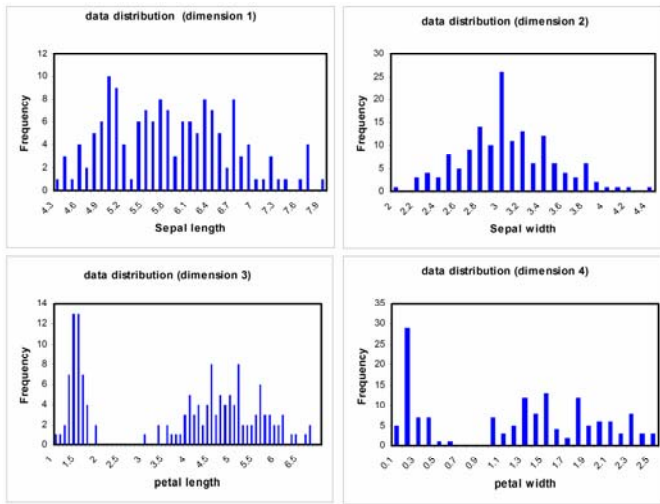


Fig. 5. Data distribution of the four dimensions in Iris data

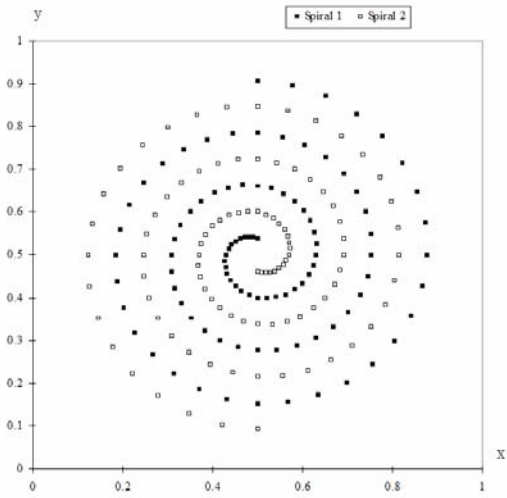


Fig. 6. Training data for the two-spiral problem

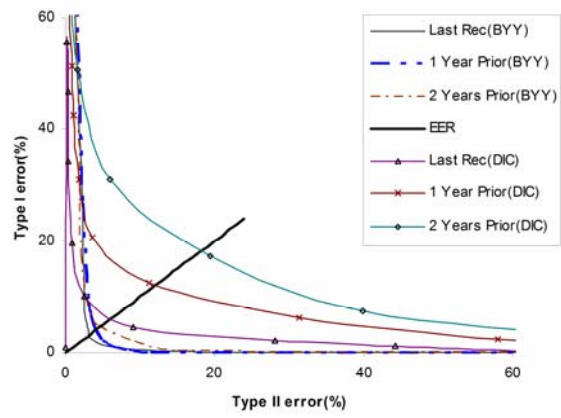


Fig. 7. ROC curves of bank failure classification results on three scenarios using YING-YANG FCMAC (BYY) and FCMAC-DIC (DIC)