2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Medical Diagnosis Applications Using a Novel Interactively Recurrent Self-evolving Fuzzy CMAC Model

Jyun-Guo Wang
National Cheng Kung University
Inst. of Comput. and Commun. Eng.
Tainan city, Taiwan
jyunguo.wang@gmail.com

Shen-Chuan Tai
National Cheng Kung University
Inst. of Comput. and Commun. Eng.
Tainan city, Taiwan
sctai@mail.ncku.edu.tw

Cheng-Jian Lin
National Chin-Yi Univ. of Technol.
Dept. of Comput. Sci. and Inf. Eng.
Taichung city, Taiwan
cjlin@ncut.edu.tw

***Abstract*—In this paper, a recurrent self-evolving Fuzzy Cerebellar Model Articulation Controller (FCMAC) model for classification problems is developed, namely the interactively recurrent self-evolving fuzzy Cerebellar Model Articulation Controller (IRSFCMAC). The interactively recurrent structure in an IRSFCMAC is formed as external loops and internal feedbacks by feeding the rule firing strength to itself and others rules. The IRSFCMAC learning starts with an empty rule base and all of rules are generated and learned online, through a simultaneous structure and parameter learning, while the relative parameters are learned through a gradient descent algorithm. The proposed IRSFCMAC is tested by the four benchmarked classification problems and compared with the well-known traditional FCMAC. Experimental results show that the proposed IRSFCMAC model enhanced classification performance results, in terms of accuracy and RMSE.**

Keywords—interactively recurrent self-evolving fuzzy Cerebellar Model Articulation Controller; gradient descent algorithm.

I. INTRODUCTION

For classification system processing, practical problems are generally encountered in a variety of areas, such as control, pattern recognition, medicine imaging, and signal processing. Recently, artificial neural networks (ANN) and fuzzy theory have becomes popular in classification behavior applications [1], [2] in order to improve and validate the effectiveness and efficiency of classifications. Being a still-popular approach in ANN research [3], [4], the Cerebellar Model Articulation Controller (CMAC) network applies the structure and function of the cerebellum of a human brain, in the sense that it is a local network, i.e., for a given input vector, only a few of the network nodes (or hypercube cells) will be active and will effectively contribute to the corresponding network output. In addition, the internal mapping structure is built in such a way that it implements, for each CMAC memory locations, one linear parametric equation of the model input variance. Using this technique, the input space is quantized into discrete states as well as larger size overlapped areas called hypercube. Each hypercube covers many discrete states and is assigned a memory cell that stores information for it. The CMAC model can be viewed as a basis function network that uses plateau basis functions. To compute the output of the model for given input data point, only those basis functions assigned to the hypercube covering the input data point are needed, is characterized by a local weight updating scheme which exhibits the advantages of fast learning ability and good generalization capability.

Although the conventional CMAC can learn much faster than multilayer feed-forward neural networks do, the learning effects are still not good enough for online learning systems [5], larger required computing memory [6], relatively poor ability of function approximation [7], [8] and difficulty of adaptively selecting structural parameters [9], [10]. Therefore, in order to overcome these mentioned drawbacks, early studies have strived for the improvement of the CMAC models topology structure [11], the selection of learning parameters [12], and convergence property of CMAC [13]. In recent years, many researchers embedded fuzzy inference (membership functions) capability into the conventional CMAC model to tackle the above-remarked disadvantages. Thus, a fuzzy CMAC model, called FCMAC, is fully exploited the advantages of fuzzy set theory and the local generalization feature of the CMAC model [14]-[16]. A non-constant differentiable basis function (i.e., Gaussian basis function) is used to model the receptive field functions and fuzzy weights. Many learning algorithms are proposed for the automatic construction of the FCMAC model during the learning procedure. However, these methods still have some drawbacks in FCMAC, such as not able to find the global optimum solution. Therefore, in order to overcome all above-mentioned drawbacks, this paper proposes a novel interactively recurrent self-evolving FCMAC (IRSFCMAC), which interactively construct the consequent part and through the structural strategy to strengthen the search abilities for both the local and global solutions. The major contributions of the proposed IRSFCMAC can be summarized as follows. (1) A novel

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

recurrent structure with interaction feedback incorporates the advantages of local and global feedbacks. Local source is not sufficient to represent the necessary information (i.e., a rule gets feedback from itself only). The global feedback in the proposed network means that the necessary information is obtained from itself and the other fuzzy rules. (2) Many studies [17], [18] have considered only the past states in recurrent structure, which is insufficient under the assumption without referring to current states. As the result, we the proposed model depends on current states along with previous states.

In the rest of this paper, the IRSFCMAC model and the corresponding on-line learning algorithm are proposed respectively in Sections II and III. In Section IV, the conducted experimental results for four benchmarked classification applications are exhibited and discussed. Final, conclusions are drawn in Section V.

II. Structure of the Proposed IRSFCMAC Model

The general concept of the IRSFCMAC model is illustrated in Fig. 1, which consists of the input space partition, association memory selection, interactively recurrent, and defuzzification. Similar to the conventional CMAC model, the IRSFCMAC model approximates a nonlinear function $y = f(x)$ by using two primary mappings $S(x)$ and $P(\alpha)$. These two mappings are realized by fuzzy operations. The function $S(x)$ maps each point x in the input space onto an association vector $\alpha = S(x) \in A$ that has N_L nonzero elements ($N_L < N_A$). Here, $0 \le \alpha \le 1$ for all components in α $(\alpha = \alpha_1, \alpha_2, ..., \alpha_n)$, is derived from the composition of the receptive field functions and sensory inputs. Different from the traditional CMAC model, several hypercube are addressed by the input state x that hypercube value is calculated by product operation through the strength of the receptive field functions for each input state. In the IRSFCMAC model, the input space is an s-dimensional, s is the number of input variables, and the Gaussian basis function is applied as the receptive field functions and the fuzzy weight functions for learning. Association memory space is a N_D-dimensional. N_D is the user-defined number of the hypercube. In addition, we added an interactively recurrent method in the consequent parts, and defuzzification represents one-dimensional output space. In the following paragraphs, the detailed functionality of each layer is fully presented.

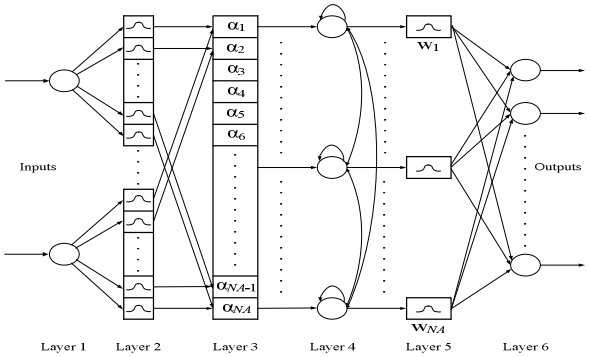

Fig. 1. Structure of the proposed IRSFCMAC model.

For clear understanding of the mathematical formulation of each node and layer, the functional relationships between each layer of the six-layered proposed model are further presented. The net input to the ith node in layer l is represented as $u_i^{(l)}$ and the output value is represents as $O_i^{(l)}$.

Layer 1 (input layer): The inputs are crisp values and $\vec{x} = (x_1, x_2, ..., x_n)$ are fed as inputs to this layer, while the corresponding outputs are computed as

$$O_i^{(1)} = u_i^{(1)}, \text{ and } u_i^{(1)} = x_i. \tag{1}$$

Layer 2 (fuzzification layer): Each node in this layer is in fact a Gaussian membership function. For the ith fuzzy set A_j^i on the input variable x_i , i=1,…,n, a Gaussian membership function is defined by refer to (2), i.e.,

$$O_{ij}^{(2)} = \exp\left(-\frac{[u_i^{(1)} - m_{ij}]^2}{\sigma_{ij}^2}\right), \text{ and } u_i^{(2)} = O_i^{(1)} \tag{2}$$

where m_{ij} and σ_{ij} are respectively the mean and variance of the Gaussian membership function of the jth term of the ith input variable x_i.

Layer 3 (spatial firing layer): Within this layer each node receives one-dimensional membership degree of the associated rule from the nodes of a set in layer 2. In other words, each node represents one fuzzy rule that computes the firing strength in this layer. For the obtained spatial firing strength α_j , each node performs a fuzzy meet operation on inputs, as it receives from the previous layer via an algebraic product operation. As a result, the output function of each inference node can be computed as,

$$\alpha_j = O_j^{(3)} = \prod_i u_{ij}^{(3)}, \text{ and } u_{ij}^{(3)} = O_{ij}^{(2)} \tag{3}$$

where the $\prod_i u_{ij}^{(3)}$ of a rule node represents the firing strength of its corresponding rule.

Layer 4 (temporal firing layer): Each node here is a recurrent rule node, which formulates an internal feedback (self-loop) and external interaction feedback loop. The output of a recurrent rule node is a temporal firing strength that depends on both the current spatial and the previous temporal firing strengths. The temporal firing strength is a linear combination function and is expressed as

$$O_j^{(4)} = \sum_{k=1} (\lambda_{kj}^q \cdot O_k^{(4)}(t-1)) + (1-\gamma_j^q) \cdot u_j^{(4)}, \text{ and } u_j^{(4)} = O_j^{(3)} \tag{4}$$

where $\gamma_j^q = \sum_{k=1}^{M} \lambda_{kj}^q$ and $\lambda_{kj}^q = \frac{R_{kj}^q}{M}$ $(0 \le R_{kj}^q \le 1)$ is the rule interaction weight between itself and other rules, and M is

the total number of current rules. The recurrent weights λ_{jk}^{q} determine the compromised ratio between the current and previous inputs to the network outputs.

Layer 5 (consequent layer): Each node is an optional node by way of fuzzy weight in this layer. Each fuzzy weight is inferred to produce a partial fuzzy output by applying the value of its corresponding association memory selection vector as $O_j^{(4)}$ matching degree. The formula expressed as

$$O_j^{(5)} = O_j^{(4)} w_j^m w_j^\sigma \tag{5}$$

Layer 6 (output layer): The partial fuzzy output is defuzzified into a scalar output by the centroid of area (COA) approach. Then the actual output y is derived as follows

$$y = \frac{\sum_{j=1}^{NL} O_j^{(4)} w_j^m w_j^\sigma}{\sum_{j=1}^{NL} O_j^{(4)} w_j^\sigma} \tag{6}$$

where w_j^m denotes the mean value of the fuzzy weights, w_j^σ the variance value of the fuzzy weights, and N_L the number of the hypercube cells.

III. Learning Algorithms of the Proposed IRSFCMAC Model

This section presents an online learning algorithm for constructing the IRSFCMAC model. The proposed learning algorithm comprises both the structural and parametric learning phases. Fig. 2 displays the flow diagram of the learning scheme for the proposed IRSFCMAC model. Firstly, the structure learning scheme is used to decide the proper input space partition. The self-constructing input space partition is based on the degree measure to appropriately determine the various distributions of the input training data. In other words, structure learning is used to determine whether a new rule should be added to satisfy the fuzzy partitioning of input variables. Secondly, the parameter learning scheme is based on supervised learning algorithms. The backpropagation algorithm minimizes a given cost function by adjusting the linked weights in the consequent part and the parameters of the membership functions. The proposed model is created dynamically and automatically in the learning process about receiving the on-line incoming training data by performing the structure and parameter learning processes. Initially, there are nodes in the network except the input-output nodes, i.e., there are no nodes in the IRSFCMAC model. The nodes are created automatically as learning proceeds, upon the reception of online incoming training data in the structure and parameter learning processes. For the initial system, the values of the turning parameters w_j^m and w_j^σ are generated randomly, where m is generated by incoming training data and σ is generated by a proper value. The rest of this section details the structural and parametric learning phases, respectively.

A. Structure learning phase

In general, structure learning aims to determine whether a new rule should be extracted from the training data and to resolve the number of fuzzy sets in the universe of discourse of each input variable, since one cluster in the input space corresponds to one potential fuzzy logic rule, in which m_{ij} and σ_{ij} represent the mean and variance of that cluster, respectively. Therefore, we can use the product operation of the firing strength obtained directly from $\prod_i u_{ij}^{(3)}$ as the degree measure

$$\alpha_j = \prod_i u_{ij}^{(3)} \tag{7}$$

where u_{ij} represents the degree of the firing strength of the input vector for $i=1,2,...,N_D$ and the association vector $\alpha_j \in [0,1]$. Equation (8) states the criterion of the degree measure for generating a new hypercube cell of new incoming data, i.e., to find the maximum degree $\alpha_{\max}$

$$\alpha_{\max} = \max_{1 \le j \le N_L} \alpha_j \tag{8}$$

where N_L is the number of nonzero elements of association vector. If $\alpha_{\max} \le \overline{\alpha}$, then a new hypercube cell is generated. Here, $\overline{\alpha} \in [0,1]$ is a prespecified threshold that should decay during the learning process in order to limit the size of the proposed IRSFCMAC model.

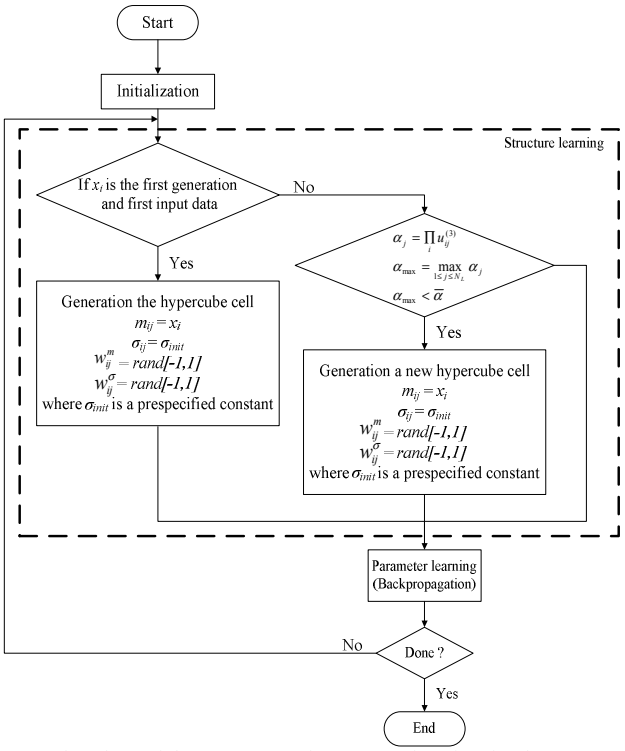

Fig. 2. Flowchart of the structure and parameter learning for the proposed IRSFCMAC model

B. Parameter learning phase

The network enters the parameter learning phase to adjust the free parameters of the network optimally based on the training data. There are five parameters need to be tuned, i.e., m_{ij}, σ_{ij}, w_j^m, w_j^σ, and R_{jk}^q. The total number of these free parameters for the multi-input single-output IRSFCMAC model is $2N_L(N_D+1)$, where N_D and N_L respectively denote the amounts of inputs and hypercube cells. Since the learning process generally involves determining the minimum of a given cost function, the gradient of the cost function is firstly computed and the parameters are correspondingly adjusted with negative gradient. The backpropagation algorithm is adopted for this supervised learning method to modify these parameters. When the single-output case is considered for clarity, the goal to minimize the cost function E is defined as,

$$E(t) = \frac{1}{2}[y^d(t) - y(t)]^2 \tag{9}$$

where $y^d(t)$ denotes the desired output and $y(t)$ the model output for each discrete time t. When the backpropagation learning algorithm is applied, the weighted vector of the proposed IRSFCMAC model is regulated such that the error defined in refer to (9) can be less than the desired threshold value after a given number of training cycles. The well-known backpropagation learning algorithm may be written briefly as follows, i.e.,

$$W(t+1) = W(t) + \Delta W(t) = W(t) + \left(-\eta \frac{\partial E(t)}{\partial W(t)}\right) \tag{10}$$

where η and W represent the learning rate and the free parameters of the proposed model, individually. Then, the gradient of error function $E(\cdot)$ in refer to (9) with respect to an arbitrary weight vector W is calculated by

$$\frac{\partial E(t)}{\partial W} = e(t)\frac{\partial y(t)}{\partial W} \tag{11}$$

By the chain rule yield the error term recursive applications for each layer and the parameter in the corresponding layers are adjusted. Therefore, we used the gradient concept of the backpropagation algorithm to tune the antecedent and consequent parameters of the IRSFCMAC model. The gradient descent algorithm is performed once for each piece of incoming datum.

By using a gradient descent algorithm for the updated recurrent weights, we have

$$\lambda_{kj}^q(t+1) = \lambda_{kj}^q(t) + \Delta\lambda_{kj}^q(t) \tag{12}$$

and

$$\begin{aligned}\Delta\lambda_{kj}^q(t) &= -\eta\frac{\partial E}{\partial \lambda_{kj}^q} \\ &= -\eta\cdot\frac{\partial E}{\partial y}\cdot\frac{\partial y}{\partial O_j^{(4)}}\cdot\frac{\partial O_j^{(4)}}{\partial \lambda_{kj}^q} \\ &= -\eta\cdot e\cdot\left[\frac{w_j^m\cdot(w_j^m - y)}{\sum_{j=1}^{NL} O_j^{(4)} w_j^\sigma}\right]\cdot(O_k^{(4)}(t-1) - \alpha_j)\end{aligned} \tag{13}$$

where η, between 0 and 1, is the learning rate of the recurrent λ for the fuzzy weight functions, and e denotes the error between the desired output and actual output, i.e., $e = y^d - y$.

The fuzzy weight parameter w_j^m and w_j^σ cells are updated according to the following equations, i.e.,

$$w_j^m(t+1) = w_j^m(t) + \Delta w_j^m(t) \tag{14}$$

and

$$w_j^\sigma(t+1) = w_j^\sigma(t) + \Delta w_j^\sigma(t) \tag{15}$$

where j denotes the jth fuzzy weight cell for $j=1,2,...,N_L$, w_j^m denotes the mean of the fuzzy weights and w_j^σ the variance of the fuzzy weights. The elements of the fuzzy weights are updated by the following amounts, i.e.,

$$\begin{aligned}\Delta w_j^m(t) &= -\eta\cdot\frac{\partial E}{\partial y}\cdot\frac{\partial y}{\partial w_j^m} \\ &= -\eta\cdot e\cdot\frac{O_j^{(4)} w_j^\sigma}{\sum_{j=1}^{NL} O_j^{(4)} w_j^\sigma}\end{aligned} \tag{16}$$

and

$$\begin{aligned}\Delta w_j^\sigma &= -\eta\cdot\frac{\partial E}{\partial y}\cdot\frac{\partial y}{\partial w_j^\sigma} \\ &= -\eta\cdot e\cdot\left(\frac{O_j^{(4)}\cdot(w_j^m - y)}{\sum_{j=1}^{NL} O_j^{(4)} w_j^\sigma}\right)\end{aligned} \tag{17}$$

where η is the learning rate of the mean and the variance for the fuzzy weight functions between 0 and 1.

The antecedent part of parameters m_{ij} and σ_{ij} are computed via the following equations, i.e.,

$$m_{ij}(t+1) = m_{ij}(t) + \Delta m_{ij}(t) \tag{18}$$

and

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \Delta\sigma_{ij}(t) \tag{19}$$

where i represents the ith input dimension for $i=1,2,...,n$, m_{ij} the mean of the receptive field functions and σ_{ij} is the variance of the receptive field functions. The parameters of the receptive field functions are calculated by

$$\begin{aligned}\Delta m_{ij} &= -\eta \cdot \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial O_j^{(4)}} \cdot \frac{\partial O_j^{(4)}}{\partial O_j^{(3)}} \cdot \frac{\partial O_j^{(3)}}{\partial O_j^{(2)}} \cdot \frac{\partial O_j^{(2)}}{\partial m_{ij}} \\ &= -\eta \cdot e \cdot \left(\frac{w_j^{\sigma} \cdot (w_j^m - y)}{\sum_{j=1}^{NL} O_j^{(4)} w_j^{\sigma}} \right) \cdot (1-\gamma_j^q) \cdot \alpha_j \cdot \frac{2(u_i^{(1)} - m_{ij})}{\sigma_{ij}^2}\end{aligned} \tag{20}$$

and

$$\begin{aligned}\Delta\sigma_{ij} &= -\eta \cdot \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial O_j^{(4)}} \cdot \frac{\partial O_j^{(4)}}{\partial O_j^{(3)}} \cdot \frac{\partial O_j^{(3)}}{\partial O_j^{(2)}} \cdot \frac{\partial O_j^{(2)}}{\partial \sigma_{ij}} \\ &= -\eta \cdot e \cdot \left(\frac{w_j^{\sigma} \cdot (w_j^m - y)}{\sum_{j=1}^{NL} O_j^{(4)} w_j^{\sigma}} \right) \cdot (1-\gamma_j^q) \cdot \alpha_j \cdot \frac{2(u_i^{(1)} - m_{ij})^2}{\sigma_{ij}^3}\end{aligned} \tag{21}$$

where η is the learning rate of the mean and the variance for the receptive field functions, individually. It is worthy to mention that all the above stated formulas belong to the case of a multi-input single-output system. If one wants to perform multi-input multi-output system the cost function E should be rewritten as

$$E = \frac{1}{2k} \sum_{k=1}^{k} (y_k^d(t) - y_k(t))^2 \tag{22}$$

where k is the number of output for $k=1,2,...,k$ and then the tasks are to update those free parameters, such as $m_{ij}, \sigma_{ij}, w_j^m, w_j^{\sigma}$ and λ_{kj}^q.

IV. Experimental Results

In order to demonstrate the classification performance of the proposed IRSFCMAC model, two well-known benchmarked classification problems, i.e., Breast Cancer and Thyroid, are chosen in this paper. All of the two datasets are from the (UCI) machine learning repository, University of California, Irvine. In addition, the Thyroid classification is low-dimensional problem, while the Breast Cancer problem is high-dimensional one.

A. Classification of Thyroid data

The 215-pattern Thyroid data set is a three-class problem (i.e., normal, hyper and hypo) with 150, 35 and 30 instances, respectively.

Each pattern contains five features, and they are T3-resin uptake test, total serum thyroxin, total serum triiodothyronine, Basal thyroid-stimulating hormone (TSH) and Maximal absolute difference of TSH value, where Total serum thyroxin is measured by the isotopic displacement method, Total serum triiodothyronine and Basal TSH by radioimmuno assay, and Maximal absolute difference of TSH value after injection of 200 micro grams of thyrotropin-releasing hormone as compared to the basal value. Since the data take half unevenly, we have taken randomly 107 patterns from 215 patterns as training data and the rest 108 for testing. In this classification, the hypercube of our proposed IRSFCMAC model is automatically produced, while the number of generations is set to 5000 times and initial parameters η = 0.01, $\overline{\alpha} = 10^{-6}$, which is identical to the initial settings for the compared FCMAC model. In addition, the output y of the proposed IRSFCMAC here is defined by following classification rules, i.e.,

$$\text{Thyroid} = \begin{cases} Normal, & \text{if} \quad y \le 1.5 \\ Hyper, & \text{if} \quad 1.5 < y \le 2.5 \\ Hypo, & \text{if} \quad y > 2.5 \end{cases} \tag{24}$$

The 10-run averaged results for learning curves of the traditional FCMAC and IRSFCMAC are exhibited in Fig. 3.

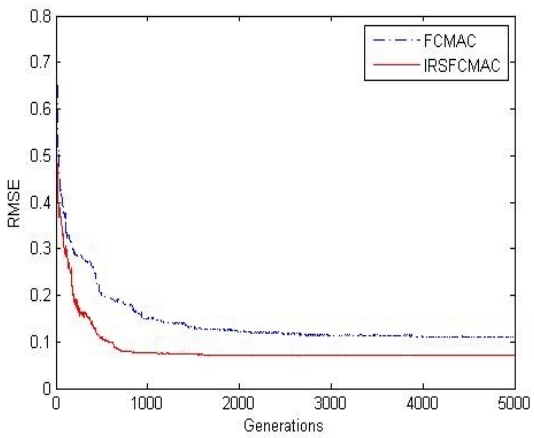

Fig. 3. 10-run averaged learning curves (the Thyroid data)

Table I presents the RMSE compared results, with the best, mean, and worst of the 10 independent runs, while the detailed outcomes for these 10 runs are illustrated within Table II. Furthermore, from the testing results presented in Table III it can be showed that our proposed method granted averaged better testing accuracy than the FCMAC method.

TABLE I. RMSE comparison in training (the Thyroid data).

Models	Items		
	Mean RMSE	***Best RMSE***	***Worst RMSE***
FCMAC	0.1228	0.1108	0.1367
IRSFCMAC	0.0785	0.0709	0.0826

TABLE II. Numbers of hypercube cells and accuracy rates for 10 independent runs (the Thyroid data).

Models	Experiments				
	1	*2*	*3*	*4*	*5*
Hypercube cell (FCMAC)	8	10	8	8	8
Hypercube cell (IRSFCMAC)	6	5	4	6	5
Accuracy (%) (FCMAC)	91.6667	87.9630	81.4815	86.1111	80.5556
Accuracy (%) (IRSFCMAC)	94.4444	92.5926	94.4444	93.5185	91.6667
Models	**Experiments**				
	6	*7*	*8*	*9*	*10*
Hypercube cell (FCMAC)	7	9	7	7	10
Hypercube cell (IRSFCMAC)	5	5	3	3	3
Accuracy (%) (FCMAC)	75.0000	87.9630	87.9630	95.3704	95.2963
Accuracy (%) (IRSFCMAC)	95.3704	93.5185	94.4444	95.3704	91.6667

TABLE III. Accuracy comparison in testing (the Thyroid data).

Models	Items		
	Mean	*Best*	*Worst*
FCMAC	87.1371%	95.3704%	75.0000%
IRSFCMAC	93.7037%	95.3704%	91.6667%

B. Classification of Breast Cancer data

The second classification experiment is about the 699-pattern Breast Cancer data set, consisting of nine features, i.e., Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli and Mitoses. In other words, the Breast Cancer classification is one of the well-known multi-dimension problem and used here to verify whether the proposed model can be effectively applied. In this classification, the initial parameters η = 0.01, $\overline{\alpha} = 10^{-6}$. In this paper, the partition of the training and testing data sets of the Breast Cancer data are 349 and 350, respectively. The output y of the IRSFCMAC model is defined by the following classification rules, i.e.,

$$\text{BreastCancer} = \begin{cases} \textit{Benign}, & \textit{if} \quad y \le 0.5 \\ \textit{Malignant}, & \textit{if} \quad \textit{otherwise} \end{cases} \tag{25}$$

Other settings of this application are identical to the second experiments, while the averaged learning curves are displayed as in Fig. 4.

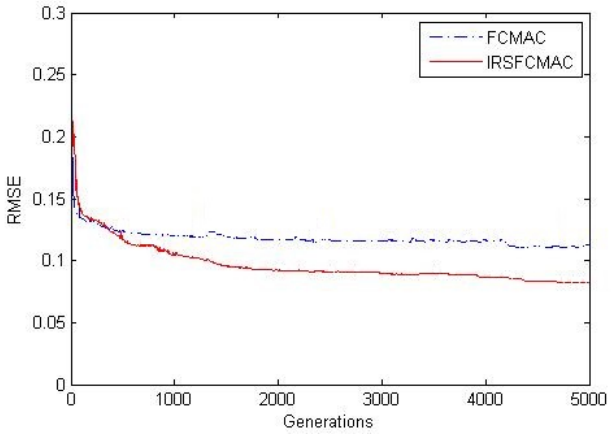

Fig. 4. 10-run averaged learning curves (the Breast Cancer data)

Tables IV and V respectively show the conducted 10-run RMSE results, numbers of the hypercube cells and accuracy rates. According to these two tables our proposed method grants the better averaged testing accuracy than the FCMAC method, as same as Table VI exhibits.

TABLE IV. RMSE comparison in training (the Breast Cancer data).

Models	Items		
	Mean RMSE	*Best RMSE*	*Worst RMSE*
FCMAC	0.1215	0.1128	0.1372
IRSFCMAC	0.0902	0.0826	0.0965

TABLE V. Numbers of hypercube cells and accuracy rates for 10 independent runs (the Breast Cancer data).

Models	Experiments				
	1	*2*	*3*	*4*	*5*
Hypercube cell (FCMAC)	7	6	8	6	7
Hypercube cell (IRSFCMAC)	4	4	5	4	3
Accuracy (%) (FCMAC)	95.1429	97.1429	94.8571	96.2857	96.5714
Accuracy (%) (IRSFCMAC)	96.2857	97.1429	95.7143	96.5714	96.8571
Models	**Experiments**				
	6	*7*	*8*	*9*	*10*
Hypercube cell (FCMAC)	6	6	5	5	6
Hypercube cell (IRSFCMAC)	3	3	4	3	4
Accuracy (%) (FCMAC)	96.2857	93.7143	94.8571	95.4286	96.0000
Accuracy (%) (IRSFCMAC)	96.2857	94.0000	94.8571	95.4286	96.8571

TABLE VI. Accuracy comparison in testing (the Breast Cancer data).

Models	Items		
	Mean	*Best*	*Worst*
FCMAC	95.6016%	97.1429%	93.7143%
IRSFCMAC	96.0000%	97.1429%	94.0000%

V. CONCLUSIONS

In this paper, we propose a novel interactively recurrent self-constructing Fuzzy CMAC (IRSFCMAC) model. The proposed approach applies a non-constant differentiable Gaussian basis function to model the hypercube structure and the corresponding fuzzy weights, which interactively construct the consequent part and through the structure strategy to strengthen the search abilities for both the local and global solutions. An on-line learning algorithm for consisting of structure and parameter learning schemes is presented as well. The degree measure is used to determine the proper input space partition size in the structure learning scheme and based on supervise gradient-descent method developed in parameter learning scheme. The simulated classification examples provide the positive evidence of the effectiveness of the proposed model. Experimental Results also indicated that the proposed IRSFCMAC model has a higher average recognition rate and lower memory requirement than the traditional FCMAC model.

REFERENCES

[1] J. Wang and B. Liu, "A Study on Emotion Classification of Image Based on BP Neural Network," Proc. Int'l Conf. Information Science and Management Engineering, vol. 1, pp. 100-104, 2010.

[2] Z. Xiong, K. Chen, C. Gu, Y. Liang, and F. Yu, "An algorithm of image classification based on BP neural network," Proc. IEEE Int'l Conf. Computer Science and Automation Engineering, vol. 1, pp. 523-526, 2012.

[3] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," Trans. ASME J. Dyn. Syst. Meas. Contr., pp. 220-227, 1975.

[4] J. S. Albus, "Data storage in the cerebellar model articulation controller (CMAC)," Trans. ASME J. Dyn. Syst. Meas., pp. 228-233, Sept. 1975.

[5] S. F. Su, T. Tao, and T. H. Hung, "Credit assigned CMAC and its application to online learning robust controllers," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 33, no. 3, pp. 202-213, Jun. 2003.

[6] Y. G. Leu, C. M. Hong, Z. R. Chen and J. H. Liao, "Compact cerebellar model articulation controller for ultrasonic motors," Int'l J. Innovative Computing, Information and Control (IJICIC), vol. 6, no. 12, pp. 5539-5552, 2010.

[7] J. Wu and F. Pratt, "Self-organizing CMAC neural networks and adaptive dynamic control," Proc. IEEE Int. Symp. Intell. Contr./Intell. Syst. Semiotics, Cambridge, MA, pp. 259-265, 1999.

[8] S. Commuri and F. L. Lewis, "CMAC neural networks for control of nonlinear dynamical systems: structure, stability, and passivity," Automatics, vol. 33, no. 4, pp. 635-641, 1997.

[9] K. S. Hwang and C. S. Lin, "Smooth trajectory tracking of three-link robot: a self-organizing CMAC approach," IEEE Trans. Systems, Man, Cybern., Part B, vol. 28, no. 5, pp. 680-692, 1998.

[10] H. M. Lee, C. M. Chen and Y. F. Lu, "A self-organizing HCMAC neural-network classifier," IEEE Trans. Neural Networks, vol. 14, no. 1, pp. 15-27, 2003.

[11] C. S. Lin and C. K. Li, "A new neural network structure composed of small CMACs," Proc. IEEE Conf. Neural Systems, pp. 1777-1783, 1996.

[12] S. H. Lane and J. Militzer, "A comparison of five algorithm for the training of CMAC memories for learning control systems," Int. Fed. Automat. Contr., vol. 28, no. 5, pp. 1027-1035, 1992.

[13] C. S. Lin and C. T. Chiang, "Learning convergence of CMAC Technique," IEEE Trans. Neural Netw., vol. 8, no. 6, pp. 1281-1292, Nov. 1997.

[14] J. S. Ker, C. C. Hsu, Y. H. Huo and B. D. Liu, "A fuzzy CMAC model for color reproduction," Fuzzy Sets and Systems, vol. 91, pp. 53-68, 1997.

[15] K. Zhang and F. Qian, "Fuzzy CMAC and its application," Proc. 3rd World Congress on Intelligent Control and Automation, Hefei, P.R. China, pp. 944-947, 2000.

[16] C. Guo, Z. Ye, Z. Sun, P. Sarkar and M. Jamshidi, "A hybrid fuzzy cerebellar model articulation controller based autonomous controller," Comp. Elec. Eng., vol. 28, pp. 1-16, 2002.

[17] J. B. Theocharis, "A high-order recurrent neuro-fuzzy system with internal dynamics: application to the adaptive noise cancellation," Fuzzy Sets and Syst., vol. 157, no. 4, pp. 471-500, Feb. 2006.

[18] D. G. Stavrakoudis and J. B. Theocharis, "A recurrent fuzzy neural network for adaptive speech prediction," Proc. IEEE Int'l Conf. Syst., Man, Cybern., Montreal, QC, Canada, pp. 2056-2061, Oct. 2007.