1000-9825/2000/11(11)1440-11 ©2000 Journal of Software 软件学报 Vol. 11, No. 11

Theoretical Results on Learning Convergence of Generalized Fuzzy CMAC*

WANG Shi-tong[1,2], Baldwin, J.F.[2], Martin, T.P.[2]

[1](Department of Computer Science, Eastchina Shipbuilding Institute, Zhengjiang 212003, China)

[2](Advanced Computing Research Center, Bristol University, UK)

E-mail: zjstwang@public.zj.js.cn

http://www.zj.js.cn

Received November 25, 1998; accepted August 23, 1999

Abstract: In this paper, a generalized fuzzy CMAC (cerebellar model articulation controller) is presented, the sufficient condition of the learning of the generalized fuzzy CMAC is derived, and finally the learning convergence of the generalized fuzzy CMAC to the least square error is proved. The results provide a mathematical foundation for the generalized fuzzy CMAC's wide applications.

Key words: CMAC (cerebellar model articulation controller); learning convergence; fuzzy set; learning rule

Two decades ago, Dr. Albus presented a unique neural network model called CMAC (cerebellar model articulation controller)[1], based on a model of the human memory and neuro muscular control system. A CMAC network as a controller has the capability to learn an unknown nonlinear mapping by examples, and to reproduce multiple outputs in response to multiple inputs. Because of its table look-up mechanism and its hash-code based mapping structure, CMAC is able to cope with high-dimensional input/output applications without severely deteriorating their processing speed and performance. Recently, Dr. C. T. Chiang *et al.* proved the learning of CMAC can converge[5].

Fuzzy set theory was initially proposed by L. A. Zadeh as a tool to model the imprecision that is inherent in human reasoning, especially when dealing with complexity. This theory has been used in widespread application areas. One of the important applications is FLC (fuzzy logic controller). FLC is easy to design, especially in cases where the control laws are nonlinear and the system is complex.

As mentioned above, CMAC and FLC can been used as controllers. The advantages of CMAC over FLS are as follows.

(1) There are very efficient learning laws to update the values of weights based on experience and examples.

(2) There is a random mapping mechanism to reduce the physical memory requirement for multiple input and high-resolution applications.

(3) There exist efficient input encoding schemes for high-dimensional input vectors.

The advantages of FLC over CMAC are as follows.

(1) It is possible to interpret the implication of weight values using linguistic labels.

---

* Project is supported by the British Royal Society. **WANG Shi-tong** was born in 1964. He is the head and professor of Department of Computer Science at the Eastchina Shipbuilding Institute. His research interests include artificial intelligence, fuzzy systems, neural networks, etc. **Baldwin, J.F.** is the head and professor of Advanced Computer Research Center of Bristol University. **Martin, T.P.** is the reader of Advanced Computer Research Center of Bristol University.

(2) The membership functions and the firing strengths contain additional information as to how close the input vector is to each linguistic variable. Therefore, the number of input space partitions may be smaller to achieve the same generalization and output smoothness.

(3) The fuzzy rules can take a variety of forms while only numeric values can be associated with CMAC associative memory locations.

(4) There are many approaches to construct a fuzzy control knowledge base, using expert's experience and knowledge.

Fuzzy CMAC combines the advantages of both CMAC and FLC, and is obtaining greater attention and being used in applications. The architecture of often-used fuzzy CMACs is shown in Fig. 1.

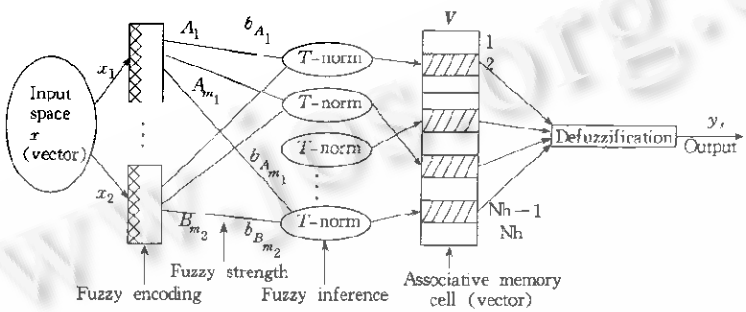

Fig. 1 Architecture of often-used fuzzy CMAC

As shown in Fig. 1, fuzzy CMAC inherits the preferred features of arbitrary function approximation, learning and parallel processing from the original CMAC neural network, and the capability of acquiring and incorporating human knowledge into a system and the capability of processing information based on fuzzy inference rules from fuzzy logic. The combination of neural network and fuzzy logic yields an advanced intelligent system architecture.

At the input stage, the fuzzy CMAC uses the fuzzification method of FLC as its input encoding scheme. Fuzzy rules can be assigned to each associative memory cell. These rules may not necessarily have a crisp consequent part. The output generation uses a centroid defuzzification approach which sums up the weighted outputs of the activated rules based on the firing strengths b_i. The overall mapping function of a fuzzy CMAC can be formalized as

$$y_s(x)=\frac{\sum_{p=1}^{M} b_p v_p}{\sum_{p=1}^{M} b_p} \tag{1}$$

where $x=[x-1,x-2,\ldots,x-N]^T$ is the input vector; v_p, $p=1,2,\ldots,M$, are weights of the network; $M=j_1$, if $N=1$; $M=\sum_{i=2}^{N}(j_-1)\prod_{i=1}^{i-1}m_{i-1}+j_1$ for $N>1$, $i=1,2,\ldots N$, and m_j is the number of knot points (see Fig. 1) on the ith input dimension. $j_i=1,2,\ldots,m_i$, $b_p=\mu_{1,j1}(x-1)*\mu_{2,j2}(x-2)*\ldots*\mu_{N,jN}(x-N)$, where ji (including the following similar representations) denotes j_i.

For most fuzzy CMACs, an often-used learning rule of weights is based on least square error and BP-learning rule. Thus, an important problem occurs: Can such learning rule converge? Up to date, although there are many successful applications, only one paper deals with the theoretical approximation problem for fuzzy CMAC with a special fuzzy membership function[6]. The learning convergence problems are as yet to be solved. This paper firstly generalizes the above fuzzy CMAC, then answers this problem. Our analysis results show that generalized fuzzy CMACs have the feature of learning convergence. This conclusion is very important, and it provides the solid mathematical foundation for wide applications.

1 Generalized Fuzzy CMAC and Its Learning Rule

In this section, we will describe the generalized fuzzy CMAC and its learning rule.

Definition 1. By an &-operation we mean a continuous function $f_{\&}:[0,1]*[0,1]\rightarrow[0,1]$ that satisfies the following four properties:

(1) $f_{\&}(0,0)=f_{\&}(0,1)=f_{\&}(1,0)=0, f_{\&}(1,1)=1$;

(2) $f_{\&}(a,b)=f_{\&}(b,a)$ for all a and b,

(3) $f_{\&}(a,b)<=a$ for all a and b,

(4) if $a>0$ and $b>0$ then $f_{\&}(a,b)>0$.

Obviously, $f_{\&}$ is an extension of triangular norm T.

Definition 2. Suppose $\mu_{i,ji}(x-i)$ is a fuzzy membership function, we define the generalized fuzzy basis function $b_p(x)$ as

$$b_p(x)=f_{\&}(\mu_{1,j1}(x-1),(f_{\&}(\mu_{2,j2}(x-2),(f_{\&}\ldots f_{\&}(\mu_{N-1jN-1}(x-(N-1)),\mu_{N,jN}(x-N)))))$$

Remark 1. There are $N*\sum_{j=1}^{N}m_j$ fuzzy membership functions in fuzzy encoding, as shown in Fig. 1. For Fig. 1, we can get $Nh=\sum_{i=1}^{N}m_i$ generalized fuzzy basis functions, i.e, $p=1,2,\ldots,Nh$.

Remark 2. The generalized fuzzy basis function here is quite different from the often used fuzzy basis function, just like

$$\mu_{i,ji}(x-i)\Big/\sum_{p=1}^{Nh}b_p$$

For generalized fuzzy basis functions, it is very easy to prove the following theorem.

Theorem 1. &-operation is the extension of triangle T norm; a generalized fuzzy basis function is a bounded fuzzy basis function.

Let $\boldsymbol{a}_s^T=[a_{s,1},a_{s,2},\ldots,a_{s,Nh}]$ denote a selection vector of generalized fuzzy basis functions which has M_1. Formula (1) can be represented as:

$$y_s=\boldsymbol{a}_s^T B(x_s)\boldsymbol{v}\Big/\sum_{i=1}^{Nh}=a_{s,i}b_i$$

$$=[a_{s,1},a_{s,2},\ldots,a_{s,Nh}]\begin{bmatrix} b_1(x_s) & 0 & 0 & 0 \\ 0 & b_2(x_s) & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & b_{Nh}(x_s)\end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_{Nh}\end{bmatrix}\Big/\sum_{i=1}^{Nh}a_{s,i}b_i \tag{2}$$

where

$$B(x_s)=\begin{bmatrix} b_1(x_s) & 0 & 0 & 0 \\ 0 & b_2(x_s) & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & b_{Nh}(x_s)\end{bmatrix},\ \boldsymbol{v}=\begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_{Nh}\end{bmatrix}$$

We call Eq. (2) a generalized fuzzy CMAC. It is obvious that the fuzzy CMAC shown in Fig. 1 is a special case of a generalized fuzzy CMAC. Generalized fuzzy CMAC represents a family of fuzzy CMAC.

Remark 3. It should be pointed out that we can not define $a_{s,i}b_i\Big/\sum_{i=1}^{Nh}a_{s,i}b_i$ as a basis function or its variant. This is because $a_{s,i}$ is a component of selection vector which is not deterministic with the execution of fuzzy

CMAC. This sufficiently shows that the generalized fuzzy CMAC is not a simple fuzzy generalization of CMAC, and that we must view it as a new model to be investigated.

Suppose Y_s is the actual output corresponding to input x_s, we choose the least square error function:

$$E=(Y_s-y_s)^2/2 \tag{3}$$

then the learning rule of the generalized fuzzy CMAC is as follows:

$$\Delta v_k=-\alpha/M*\frac{\partial E}{\partial v_k}=-\alpha/M*\frac{\partial E}{\partial w_k}\frac{\partial w_k}{\partial v_k}=\alpha/M*(Y_s-\boldsymbol{a}_s^T\boldsymbol{w}(x_s))\frac{a_{s,k}b_k(x_s)}{\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)} \tag{4}$$

where $\boldsymbol{w}(x_s)=[w_1,w_2,\dots,w_{Nh}]^T=\frac{B(x_s)\boldsymbol{v}}{\sum_{i=1}^{NH}a_{s,i}b_i(x_s)}$; α/M is the learning rate; M is defined in the above section.

2 Condition of Learning Convergence of Generalized Fuzzy CMAC

In this section, we will prove that a generalized fuzzy CMAC with the above learning rule can converge to a limit cycle. Suppose a set of N_s training data is repeatedly presented to the learning rule of a generalized fuzzy CMAC, $\boldsymbol{v}_s^{(i)}$ is used to indicate the vector of weights before the sth sample is presented in the ith iteration of learning. If the number of elements in the weight vector $\boldsymbol{v}$ is less than the number of training samples, generally speaking, during the training the vector $\boldsymbol{v}$ will change between one sample and the next, i.e, $\boldsymbol{v}_s^{(i)}\neq\boldsymbol{v}_s^{(i+1)}$. However, we will prove that $\boldsymbol{v}_s^{(i)}$ will eventually equal $\boldsymbol{v}_s^{(i+1)}$ if i approaches infinity. This shows that the vector will repeat after a cycle of training.

With the above learning rule, we can rewrite it as follows:

$$\begin{aligned}
\boldsymbol{v}_s^{(i)}&=\boldsymbol{v}_{s-1}^{(i)}+\Delta\boldsymbol{v}_{s-1}^{(i)}\\
&=\boldsymbol{v}_{s-1}^{(i)}+\alpha/M*(Y_{s-1}-\boldsymbol{a}_{s-1}^{(T)}\boldsymbol{w}_{s-1}^{(i)})\begin{bmatrix}a_{s-1,1}b_1(x_{s-1})\\a_{s-1,2}b_2(x_{s-1})\\\cdots\\a_{s-1,Nh}b_{Nh}(x_{s-1})\end{bmatrix}\Big/\sum_{i=1}^{Nh}a_{s-1}b_i(x_{s-1})\\
&=\boldsymbol{v}_{s-1}^{(i)}+\alpha/M\Big(Y_{s-1}-\boldsymbol{a}_{s-1}^{(T)}B(x_{s-1})\boldsymbol{v}_{s-1}^{(i)}\Big/\sum_{i=1}^{Nh}\boldsymbol{a}_{s-1,i}\boldsymbol{b}_i(\boldsymbol{x}_{s-1})\Big)\boldsymbol{B}(\boldsymbol{x}_{s-1})\boldsymbol{a}_{s-1}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\\
&=\boldsymbol{v}_{s-1}^{(i)}+\alpha/M\Big(B(x_{s-1})\boldsymbol{a}_{s-1}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)\Big(Y_{s-1}-\boldsymbol{a}_{s-1}^{(T)}B(x_{s-1})\boldsymbol{v}_{s-1}^{(i)}\Big)\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_o(x_{s-1})\Big)
\end{aligned} \tag{5}$$

By using the above Eq. (5), we can calculate the difference in vector v_s between two consecutive iterations i and $i+1$ as follows:

$$\begin{aligned}
D\boldsymbol{v}_s^{(i)}&=\boldsymbol{v}_s^{(i+1)}-\boldsymbol{v}_s^{(i)}=\boldsymbol{v}_{s-1}^{(i+1)}+\Delta\boldsymbol{v}_{s-1}^{(i+1)}-(\boldsymbol{v}_{s-1}^{(i)}-\Delta\boldsymbol{v}_{s-1}^{(i)})\\
&=D\boldsymbol{v}_{s-1}^{(i)}+\alpha/M*\Big[B(x_{s-1})\boldsymbol{a}_{s-1}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big](Y_{s-1}-\boldsymbol{a}_{s-1}^{(T)}B(x_{s-1})\boldsymbol{v}_{s-1}^{(i+1)})\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})-\\
&\quad\alpha/M*\Big[B(x_{s-1})\boldsymbol{a}_{s-1}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big](Y_{s-1}-\boldsymbol{a}_{s-1}^{(T)}B(x_{s-1})\boldsymbol{v}_{s-1}^{(i)})\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\\
&=D\boldsymbol{v}_{s-1}^{(i)}-\alpha/M*\Big[B(x_{s-1})\boldsymbol{a}_{s-1}\boldsymbol{a}_{s-1}^{(T)}B(x_{s-1})\Big/\Big(\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)^2\Big]*(\boldsymbol{v}_{s-1}^{(i+1)}-\boldsymbol{v}_{s-1}^{(i)})\\
&=\Big(\boldsymbol{I}-\alpha/M*B(x_{s-1})\boldsymbol{a}^{s-1}\boldsymbol{a}_{s-1}^{(T)}B(x_{s-1})\Big/\Big(\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)^2\Big]\Big)*D\boldsymbol{v}_{s-1}^{(i)}
\end{aligned} \tag{6}$$

We define $D\boldsymbol{v}_0^{(i)}=D\boldsymbol{v}_{N_s}^{(i)}$ and $\boldsymbol{a}_{N_s}=\boldsymbol{a}_0$ while $s=1$. We also define

$$\boldsymbol{E}_s(x_s)=\boldsymbol{I}-\alpha/M*B(x_{s-1})\boldsymbol{a}_{s-1}\boldsymbol{a}_{s-1}^{(T)}B(x_{s-1})\Big/\Big(\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)^2 \tag{7}$$

With these notations, the proof of convergence of generalized fuzzy CMAC to a limit cycle is to show that $D\boldsymbol{v}_s^{(i)}$ will become a null vector for all s when i approaches infinity.

In terms of Eqs. (6) and (7), we have

$$D\boldsymbol{v}_s^{(i)} = (\boldsymbol{E}_{s-1}\boldsymbol{E}_{s-2}\ldots\boldsymbol{E}_1\boldsymbol{E}_N\ldots\boldsymbol{E}_s)D\boldsymbol{v}_s^{(i-1)} = (\boldsymbol{E}_{s-1}\boldsymbol{E}_{s-2}\ldots\boldsymbol{E}_1\boldsymbol{E}_N\ldots\boldsymbol{E}_s)^i D\boldsymbol{v}_s^{(0)} \tag{8}$$

Let

$$\boldsymbol{G}_s = \boldsymbol{E}_{s-1}\boldsymbol{E}_{s-2}\ldots\boldsymbol{E}_1\boldsymbol{E}_N\ldots\boldsymbol{E}_s \tag{9}$$

thus, Eq. (8) becomes

$$D\boldsymbol{v}_s^{(i)} = (\boldsymbol{G}_s)^i D\boldsymbol{v}_s^{(0)} \tag{10}$$

Theorem 2. The matrix $\boldsymbol{E}_s$ has the following properties:

(1) $\boldsymbol{E}_s$ is a symmetric matrix.

(2) Let e_{ij} be the elements of the matrix $\boldsymbol{E}_s$, then

$$e_{ii} = \begin{cases} 1 & \text{if the } i\text{th element of } \boldsymbol{a}_s \text{ is } 0 \\ 1-\alpha/M * b_i(x_s) * b_i(x_s) * b_i(x_s) \Big/ \left(\sum_{i=1}^{Nh} a_{s,i} b_i(x_s)\right)^2 & \text{if the } i\text{th element of } \boldsymbol{a}_s \text{ is } 1 \end{cases}$$

$$e_{ij} = \begin{cases} 1 & \text{if the } i\text{th or } j\text{th element of } \boldsymbol{a}_s \text{ is } 0 \\ -\alpha/M * b_i(x_s) * b_j(x_s) * b_i(x_s) \Big/ \left(\sum_{i=1}^{Nh} a_{s,i} b_i(x_s)\right)^2 & \text{if the } i\text{th and } j\text{th element of } \boldsymbol{a}_s \text{ are 1 and } i \neq j \end{cases}$$

Proof. This theorem can be directly obtained from the definition of $\boldsymbol{E}_s$.

In terms of the above analysis, we may represent $\boldsymbol{E}_s$ as

$$\boldsymbol{E}_s = \begin{bmatrix} 1 & \ldots & 0 & \ldots & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & & \vdots & & & \vdots & & & & \vdots & & \vdots \\ 0 & \ldots & e_{pp} & 0 & \ldots & e_{pq} & 0 & \ldots & e_{pr} & 0 & \ldots & 0 \\ & & 0 & 1 & & 0 & & & 0 & & & 0 \\ \vdots & & \vdots & & & \vdots & & & \vdots & & & \\ 0 & \ldots & e_{qp} & 0 & \ldots & e_{qq} & 0 & \ldots & e_{qr} & 0 & \ldots & 0 \\ & & 0 & & & 0 & 1 & & 0 & & & \\ \vdots & & \vdots & & & \vdots & & & \vdots & & & \\ 0 & \ldots & e_{rp} & 0 & \ldots & e_{rq} & 0 & \ldots & e_{rr} & 0 & \ldots & \vdots \\ \vdots & & \vdots & & & \vdots & & & \vdots & & & \\ 0 & \ldots & 0 & \ldots & \ldots & 0 & \ldots & \ldots & 0 & \ldots & \ldots & 1 \end{bmatrix}$$

Theorem 3. Let S_s be a set indicating the positions of those non-zero e_{ij} elements in $\boldsymbol{E}_s$. If α is positive and less than

$$2M\left[\left(\sum_{i=1}^{Nh} a_{s,i} b_i(x_s)\right)^2\right] \Big/ \sum_{i \in S_s} b_i(x_s) * b_i(x_s)$$

for any matrix $\boldsymbol{M}_s$ whose elements are m_{ij}, the norm of row vector i of $\boldsymbol{M}_s \cdot \boldsymbol{E}_s$ will not be greater than that of row vector i of $\boldsymbol{M}_s$.

Proof. Let $\boldsymbol{Q} = \boldsymbol{M}_s \cdot \boldsymbol{E}_s$ and $\boldsymbol{S}_s = \{p, q, \ldots, r\}$ be a set indicating the positions of those non-zero e_{ij} elements in $\boldsymbol{E}_s$. The subscript s denotes that $\boldsymbol{S}_s$ is the set for $\boldsymbol{E}_s$ and x_s. For simplicity, we use b_i to denote $b_i(x_s)$ in the following proof. With the above definitions, the norm can be calculated as follows.

$$\| \text{row } i \text{ of } \boldsymbol{Q} \|^2 = \sum_{j=1}^{Nh} q_{ij} * q_{ij} = (m_{ip}e_{pp} + m_{iq}e_{qp} + \ldots + m_{ir}e_{rp})^2 + (m_{ip}e_{pq} + m_{iq}e_{qq} + \ldots + m_{ir}e_{rq})^2 +$$

$$(m_{ip}e_{pr} + m_{iq}e_{qr} + \ldots + m_{ir}e_{rr})^2 + \sum_{j=1, j\in S_s}^{Nh} m_{ij}m_{ij}$$

$$= \left(m_{ip} - \alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \sum_{l\in S_s} m_{il}b_lb_p \right)^2 + \left(m_{iq} - \alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \sum_{l\in S_s} m_{il}b_lb_q \right)^2 +$$

$$\left(m_{ir} - \alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \sum_{l\in S_s} m_{il}b_lb_r \right)^2 + \sum_{j=1, j\in S_s}^{Nh} m_{ij}m_{ij}$$

$$= \sum_{l\in S_s} m_{il}m_{il} - 2a/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 (m_{ip}b_p + m_{iq}b_q + \ldots + m_{ir}b_r) \sum_{l\in S_s} m_{il}b_l +$$

$$\left(\alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right)^2 (b_pb_p + b_qb_q + \ldots + b_rb_r) \left(\sum_{l\in S_s} m_{il}b_l \right)^2 + \sum_{j=1, j\in S_s}^{Nh} m_{ij}m_{ij}$$

$$= \sum_{i=1}^{Nh} m_{ij}m_{ij} + \left[\left(\alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right)^2 \sum_{l\in S_s} b_lb_l - 2\alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right] \left(\sum_{l\in S_s} m_{il}b_l \right)^2$$

Since $\left(\sum_{l\in S_s} m_{il}b_l \right)^2 \geqslant 0$, if

$$\left[\left(\alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right)^2 \sum_{l\in S_s} b_lb_l - 2\alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right] < 0$$

then

$$\| \text{row } i \text{ of } \boldsymbol{Q} \|^2 \leqslant \sum_{i=1}^{Nh} m_{ij}m_{ij}$$

To make

$$\left[\left(\alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right)^2 \sum_{l\in S_s} b_lb_l - 2\alpha/M \Big/ \left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right] < 0$$

we must have

$$0 < \alpha < 2M\left[\left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right] \Big/ \sum_{l\in S_s} b_l(x_s) * b_l(x_s) \tag{11}$$

Theorem 4. If α satisfies

$$0 < \alpha < \min 2M\left[\left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right] \Big/ \sum_{l\in S_S} b_l(x_t)b_l(x_t) \text{ for all } t = tt \in \{1,2,\ldots,N_s\}, \tag{12}$$

then the norm of row i of $\boldsymbol{M} \cdot \boldsymbol{G}_s$ will not be greater than that of row i of $\boldsymbol{M}$.

Theorem 5. If

$$0 < \alpha < 2M\left[\left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right] \Big/ \sum_{l\in S_s} b_lb_l$$

and the norm of row vector i of $\boldsymbol{M} \cdot \boldsymbol{E}_s$ equals that row i of $\boldsymbol{M}$, then

$$\left(\sum_{l\in S_s} m_{il}b_l \right)^2 = 0$$

Theorems 4 and 5 can be easily derived from Theorem 3.

Theorem 6. If

$$0 < \alpha < \min 2M\left[\left(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s) \right)^2 \right] \Big/ \sum_{l\in Stt} b_l(x_t)b_l(x_t) \quad \text{for all } t = tt \in \{1,2,\ldots,N_s\},$$

then $\lim_{k\to\infty} \|$ any row of $\boldsymbol{G}_s^k \|$ is bounded and equals $\lim_{k\to\infty} \|$ the same row of $\boldsymbol{G}_s^k\boldsymbol{G}_s \|$.

Proof. In terms of Theorem 3,

$$\|\text{any row of } \boldsymbol{G}_{k_s}\| \geqslant \|\text{any row of } \boldsymbol{G}_s^k\boldsymbol{E}_{s-1}\| \geqslant$$
$$\|\text{any row of } (\boldsymbol{G}_s^k\boldsymbol{E}_{s-1})\boldsymbol{E}_{s-2}\| \ldots \geqslant \|\text{any row of } (\boldsymbol{G}_s^k\boldsymbol{E}_{s-1},\ldots,\boldsymbol{E}_{s+1})\boldsymbol{E}_s\| = \|\text{any row of } \boldsymbol{G}_s^k\boldsymbol{G}_s\|$$

From the above derivation, we can know that the norm of any row in $\boldsymbol{G}_s^k$, when k increases, will never increase and is bounded. Since the norm must be positive or zero, we can further obtain:

$$\lim_{k\to\infty}\|\text{any row of } \boldsymbol{G}_s^k\| = \lim_{k\to\infty}\|\text{any row of } \boldsymbol{G}^k\boldsymbol{G}_s\| \tag{13}$$

Theorem 7. If

$$0<\alpha<\min 2M\left[\left(\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)\right)^2\right]\Big/\sum_{l\in S_{tt}}b_l(x_t)b_l(x_t) \quad \text{for all } t=tt\in\{1,2,\ldots,N_t\},$$

as k approaches infinity, the limit of $\boldsymbol{G}_s^k\boldsymbol{E}_j$ equals the limit of $\boldsymbol{G}_s^k$ for $j=1,2,\ldots,N$, and $\boldsymbol{G}_s^k$ converges to a null or constant matrix.

Proof. In terms of Theorem 6, we have

$$\lim_{k\to\infty}\|\text{any row of } \boldsymbol{G}_s^k\| = \lim_{k\to\infty}\|\text{any row of } \boldsymbol{G}_s^k\boldsymbol{E}_{s-1}\|$$
$$= \lim_{k\to\infty}\|\text{any row of } (\boldsymbol{G}_s^k\boldsymbol{E}_{s-1})\boldsymbol{E}_{s-2}\| \ldots = \lim_{k\to\infty}\|\text{any row of } (\boldsymbol{G}_s^k\boldsymbol{E}_{s-1},\ldots,\boldsymbol{E}_{s+1})\boldsymbol{E}_s\|$$
$$= \lim_{k\to\infty}\|\text{any row of } \boldsymbol{G}_s^k\boldsymbol{G}_s\|$$

We will use the above equalities to prove that when k approaches infinity, $\boldsymbol{G}_s^k\boldsymbol{E}_{s-1}=\boldsymbol{G}_s^k, \boldsymbol{G}_s^k\boldsymbol{E}_{s-1}\boldsymbol{E}_{s-2}=\boldsymbol{G}_s^k,\ldots$, etc. Let us first examine the explicit expression for $\boldsymbol{G}_s^k\boldsymbol{E}_{s-1}$:

$$\boldsymbol{G}_s^k\boldsymbol{E}_{s-1}=\begin{bmatrix} g_{11}(k) & \ldots & \ldots & \ldots & g_{1Nh}(k) \\ \vdots & & & & \vdots \\ g_{ik}(k) & \ldots & g_{ii}(k) & \ldots & g_{iNh}(k) \\ \vdots & & & & \vdots \\ g_{NkNh}(k) & \ldots & \ldots & \ldots & g_{NkNh}(k) \end{bmatrix}\times$$

$$\begin{bmatrix} 1 & \ldots & 0 & \ldots & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & & \vdots & & & \vdots & & & & \vdots & & \vdots \\ 0 & \ldots & e_{pp} & 0 & \ldots & e_{pp} & 0 & \ldots & e_{pr} & 0 & \ldots & 0 \\ & & 0 & 1 & & 0 & 0 & & 0 & 0 & & \vdots \\ & \vdots & & \vdots & & & \vdots & & & \vdots & & \\ 0 & \ldots & e_{qp} & 0 & \ldots & e_{qq} & 0 & \ldots & e_{qr} & 0 & \ldots & 0 \\ & & 0 & & & 0 & 1 & & 0 & 0 & & \\ \vdots & & \vdots & & & \vdots & & & & \vdots & & \vdots \\ 0 & \ldots & e_{rp} & 0 & \ldots & e_{rq} & 0 & \ldots & e_{rr} & 0 & \ldots & \vdots \\ \vdots & & \vdots & & & \vdots & & & & \vdots & & \vdots \\ 0 & \ldots & 0 & \ldots & \ldots & 0 & \ldots & \ldots & 0 & \ldots & \ldots & 1 \end{bmatrix}$$

The ith row can be expressed as:

$$[g_{i1}(k),g_{i2}(k),g_{i3}(k),\ldots,(g_{ip}(k)e_{pp}+g_{iq}(k)e_{pq}+\ldots+g_{ir}(k)e_{pr}),\ldots,g_{ij}(k),\ldots] \tag{14}$$

In terms of Theorem 2, ipth element of Eq. (14) can be expressed as

$$g_{ip}(k)\left(1-\alpha/M\Big/\left(\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)\right)^2 * b_p^2(x_{s-1})\right)-\alpha/M\Big/\left(\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)\right)^2 * \left(\sum_{l\in S_s,l\neq p}g_{il}(k)b_l(x_{s-1})b_p(x_{s-1})\right)$$
$$=g_{ip}(k)-\alpha/M\Big/\left(\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)\right)^2 * b_p(x_{s-1})\left(\sum_{l\in S_s}g_{il}(k)b_l(x_{s-1})\right) \tag{15}$$

where $S_S=\{p,q,\ldots,r\}$ indicates the positions of non-zero e_{ij} elements for $\boldsymbol{E}_{s-1}$.

We have known that $\lim_{k\to\infty}\|\text{row } i \text{ of } \boldsymbol{G}_s^k\boldsymbol{E}_{s-1}\| = \lim_{k\to\infty}\|\text{row } i \text{ of } \boldsymbol{G}_s^k\|$. In terms of Theorem 5, we must have $\lim_{k\to\infty}\sum_{l\in S_s} g_{il}(k)b_l(x_{s-1})=0$. Consequently, the pth element of row i of $G_s^k E_{s-1}$ equals $g_{ip}(k)$ when k approaches infinity. We can give similar proofs for the qth element, rth element, ..., etc, in the ith row. In other words, row i will remain unchanged. Similarly, we can also prove other rows will remain unchanged for $G_s^k E_{s-1}$. Thus, the limit of $G_s^k E_{s-1}$ will equal the limit of G_s^k as k approaches infinity.

Continuing this procedure, we will prove:

$$G_s^k E_{s-1} E_{s-2} = G_s^k E_{s-2}$$

$$\ldots$$

$$G_s^k G_s = G_s^k$$

when k approaches infinity.

We know that $\lim_{k\to\infty} G_s^k$ exists and equals a null or constant matrix. This completes the proof of this theorem.

Theorem 8. If

$$0<\alpha<\min 2M\Big[\Big(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s)\Big)^2\Big]\Big/\sum_{l\in S_{tt}} b_l(x_t)b_l(x_t) \text{ for all } t=tt\in\{1,2,\ldots,N_s\}$$

then $\lim_{k\to\infty} G_s^k B(x_j)a_j$ will equal zero, $j=1,2,\ldots,N_s$.

Proof. According to Theorems 5 and 7, if $\lim_{k\to\infty} g_i^T$ is the ith row of $\lim_{k\to\infty} G_s^k$, then

$$\lim_{k\to\infty}\sum_{l\in S_j} g_{il}(k)b_l(x_j)=0 \text{ for } j=1,2,\ldots,N_s \tag{16}$$

where S_j indicates the positions of non-zero e_{ij} elements for E_j.

In terms of Eq. (16), we have

$$\lim_{k\to\infty} G_s^k B(x_j)a_j=\Big[\lim_{k\to\infty}\sum_{l\in S_j} g_{1l}(k)b_l(x_j), \lim_{k\to\infty}\sum_{l\in S_j} g_{2l}(k)b_l(x_j),\ldots \lim_{k\to\infty}\sum_{l\in S_j} g_{pl}(k)b_l(x_j)\Big]^T=0$$

for $j=1,2,\ldots,N_s$. Thus, $G_s^k D v_s^{(0)}$ will become null as k approaches infinity. Hence, this theorem holds.

With Theorems 2~8, we instantly obtain the following important theorem about the condition of the convergence of generalized fuzzy CMAC.

Theorem 9. If

$$0<\alpha<\min 2M\Big[\Big(\sum_{i=1}^{Nh} a_{s,i}b_i(x_s)\Big)^2\Big]\Big/\sum_{l\in S_{tt}} b_l(x_t)b_l(x_t) \text{ for all } t=tt\in\{1,2,\ldots,N_s\}$$

then the learning of generalized fuzzy CMAC will converge to a limit cycle.

3 Proof of the Learning Convergence to Least Square Error

In this section, with the following Theorem 10, we will prove that the learning of a generalized fuzzy CMAC will eventually converge to least square error.

Theorem 10. The learning of generalized fuzzy CMAC results in a least square error if the number of iterations approaches infinity and the learning rate approaches zero.

Proof. In terms of Eq. (6), we can derive the weight as follows:

$$v_s^{(i+1)}=Dv_s^{(i)}+v_s^{(i)}=Dv_s^{(i)}+Dv_s^{(i-1)}+v_s^{(i-1)}\ldots=\sum_{k=0}^{i} Dv_s^{(i)}+v_s^{(0)}$$

In terms of Eq. (10), we can further obtain:

$$\begin{aligned}
v_s^{(i+1)} &= \sum_{k=0}^{i} G_s^k(\Delta v_s^{(0)}+\Delta v_{s+1}^{(0)}+\ldots+\Delta v_{Ns}^{(0)}+\Delta v_1^{(1)}+\ldots+\Delta v_{s-1}^{(1)})+v_s^{(0)}\\
&= \sum_{k=0}^{i} G_s^k\alpha/M * \Big[B(x_s)a_s\Big/\sum_{i=1}^{Nh}a_{s,i}b(x_s) * \Big(Y_s-a_s^TB(x_s)v_s^{(0)}\Big/\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)\Big)+\ldots+\\
&B(x_{Ns})a_{Ns}\Big/\sum_{i=1}^{Nh}a_{Ns,i}b_i(x_{Ns}) * \Big(Y_{Ns}-a_{Ns}^TB(x_{Ns})v_{Ns}^{(0)}\Big/\sum_{i=1}^{Nh}a_{Ns,i}b_i(x_{Ns})\Big)+\\
&B(x_1)a1\Big/\sum_{i=1}^{Nh}a_{1,i}b_i(x_1) * \Big(Y_1-a_1^TB(x_1)v_1^{(1)}\Big/\sum_{i=1}^{Nh}a_{1,i}b_i(x_1)\Big)+\ldots+\\
&B(x_{s-1})a_{s-1}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1}) * \Big(Y_{s-1}-a_{s-1}^TB(x_{s-1})v_{s-1}^{(1)}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)\Big]+\\
&v_1^{(0)}+\alpha/M * \Big[B(x_1)a_1\Big/\sum_{i=1}^{Nh}a_{1,i}b_i(x_1) * \Big(Y_1-a_1^TB(x_1)v_1^{(0)}\Big/\sum_{i=1}^{Nh}a_{1,i}b_i(x_1)\Big)+\ldots+\\
&B(x_{s-1})a_{s-1}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1}) * \Big(Y_{s-1}-a_{s-1}^TB(x_{s-1})v_{s-1}^{(0)}\Big/\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)\Big]\\
&= \sum_{k=0}^{i} G_s^k\alpha/M * \Big[\Big(B(x_1)a_1Y_1\Big/\sum_{i=1}^{Nh}a_{1,i}b_i(x_1)+\ldots+\\
&B(x_{Ns})a_{Ns}Y_{Ns}\Big/\sum_{i=1}^{Nh}a_{Ns,i}b_i(x_{Ns})\Big)-\Big(B(x_1)a_1a_1^TB(x_1)v_1^{(1)}\Big/\Big(\sum_{i=1}^{Nh}a_{1,i}b_i(x_1)\Big)^2+\ldots+\\
&B(x_{s-1})a_{s-1}a_{s-1}^TB(x_{s-1})v_{s-1}^{(1)}\Big/\Big(\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)^2+\\
&B(x_s)a_sa_s^TB(x_s)v_s^{(0)}\Big/\Big(\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)\Big)^2+B(x_{Ns})a_{Ns}a_{Ns}^TB(x_{Ns})v_{Ns}^{(0)}\Big/\Big(\sum_{i=1}^{Nh}a_{Ns,i}b_i(x_{Ns})\Big)^2\Big]+v_1^{(0)}+O(\alpha)
\end{aligned} \tag{17}$$

Furthermore, for $j=1,2,\ldots,s-1$, we have

$$\begin{aligned}
B(x_j)a_ja_j^TB(x_j)v_j^{(1)} &= B(x_j)a_ja_j^TB(x_j)(v_{j-1}^{(1)}+\Delta v_{j-1}^{(1)})\\
&= B(x_j)a_ja_j^TB(x_j)(v_1^{(0)}+\Delta v_1^{(0)}+\ldots+\Delta v_{Ni}^{(0)}+\Delta v_1^{(1)}\ldots+\Delta v_{j-1}^{(0)})
\end{aligned}$$

In terms of Eq. (4), we have

$$B(x_j)a_ja_j^TB(x_j)v_j^{(1)}=B(x_j)a_ja_j^TB(x_j)v_1^{(0)}+B(x_j)a_ja_j^TB(x_j) * \alpha/M * \Big[a_1\Big/\sum_{i=1}^{Nh}a_{1,i}b_i(x_1) *$$

$$(Y_1-a_1^Tv_1^{(0)})-\ldots+a_{j-1}\Big/\sum_{i=1}^{Nh}a_{j-1,i}b_i(x_{j-1}) * (Y_{j-1}-a_{j-1}^Tv_{j-1}^{(1)})\Big]$$

Obviously, the above formula can be expressed as:

$$B(x_j)a_ja_j^TB(x_j)v_j^{(1)}=B(x_j)a_ja_j^TB(x_j)v_1^{(0)}+Z_j * \alpha \tag{18}$$

where Z_j is bounded.

Similarly, we have

$$B(x_k)a_ka_k^TB(x_k)v_k^{(0)}=B(x_k)a_ka_k^TB(x_k)v_1^{(0)}+Z_k * \alpha \tag{19}$$

where Z_k is bounded.

Suppose

$$A^T=\Big[B(x_1)a_1\Big/\sum_{i=1}^{Nh}a_{1,i}b_i(x_1),B(x_2)a_2\Big/\sum_{i=1}^{Nh}a_{2,i}b_i(x_2),\ldots,B(x_{Ns})\,a_{Ns}\Big/\sum_{i=1}^{Nh}a_{Ns,i}b_i(x_{Ns})\Big]$$

$$Y=[Y_1,Y_2,\ldots,Y_{Ns}]$$

then, we can further simplify Eq. (17) into

$$v_s^{(i+1)}=\sum_{k=0}^{i}G_s^k\alpha/M*\Big[A^TY-B(x_1)a_1a_1^TB(x_1)v_1^{(0)}\Big/\Big(\sum_{i=1}^{Nh}(a_{1,i}b_i(x_1)\Big)^2+\ldots+$$

$$B(x_{Ns})a_{Ns}a_{Ns}^TB(x_{Ns})v_1^{(0)}\Big/\Big(\sum_{i=1}^{Nh}a_{Ns,i}b_i(x_{Ns})\Big)^2-\alpha\sum_{j=1}^{Ns}Z_j\Big]+v_1^{(0)}+O(\alpha)$$

$$=\sum_{k=0}^{i}G_s^k\alpha/M*[A^TY-A^TAv_1^{(0)}-O(\alpha)]+v_1^{(0)}+O(\alpha)$$

$$\lim_{\alpha\to0}\lim_{i\to\infty}v_s^{(i+1)}=\lim_{\alpha\to0}\lim_{i\to\infty}\Big\{\sum_{k=0}^{i}G_s^k\alpha/M*[A^TY-A^TAv_1^{(0)}-O(\alpha)]+v_1^{(0)}+O(\alpha)\Big\}$$
$$=\lim_{\alpha\to0}\{(I-G)^{-1}\alpha/M*[A^TY-A^TAv_1^{(0)}-O(\alpha)]+v_1^{(0)}+O(\alpha)\} \quad (20)$$

In terms of the definitions of G_s and E_s, we have

$$G_s=E_{s-1}E_{s-2}\ldots E_1E_{Ns}\ldots E_s=(I-\alpha/M*B(x_{s-1})a_{s-1}a_{s-1}^TB(x_{s-1}))\Big/\Big(\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)^2*\ldots*$$

$$(I-\alpha/M*B(x_s)a_sa_s^TB(x_s))\Big/\Big(\sum_{i=1}^{Nh}a_{s,i}b_i(x_s)\Big)^2$$

$$=I-\alpha/M*\Big[B(x_{s-1})a_{s-1}a_{s-1}^TB(x_{s-1})\Big/\Big(\sum_{i=1}^{Nh}a_{s-1,i}b_i(x_{s-1})\Big)^2+\ldots+ \quad (21)$$

$$B(x_{Ns})a_{Ns}a_{Ns}^TB(x_{Ns})\Big/\Big(\sum_{i=1}^{Nh}a_{Ns,i}b_i(x_{Ns})\Big)^2\Big]+O(\alpha*\alpha)$$

$$=I-\alpha/M*A^TA+O(\alpha*\alpha)$$

Substituting Eq. (21) into (20) results in

$$\lim_{\alpha\to0}\lim_{i\to\infty}v_s^{(i+1)}=\lim_{\alpha\to0}\{[\alpha/M*A^TA-O(\alpha*\alpha)]^{-1}\alpha/M*[A^TY-A^TAv_1^{(0)}-O(\alpha)]\}+v_1^{(0)}$$
$$=\lim_{\alpha\to0}\{[A^TA-O(\alpha)]^{-1}*[A^TY-A^TAv_1^{(0)}-O(\alpha)]\}+v_1^{(0)} \quad (22)$$
$$=[A^TA]^{-1}A^TY-[A^TA]-A^TAv_1^{(0)}+v_1^{(0)}=[A^TA]^{-1}A^TY$$

Eq. (22) shows that the final learning result of generalized fuzzy CMAC equals the one obtained by having $v=[A^TA]^{-1}A^TY$, which gives the least square error. Thus, the most important theorem in this paper has been proved.

4 Conclusion

In this paper, we generalize the fuzzy CMAC, and then present a generalized fuzzy CMAC, which represents a family of fuzzy CMAC. After defining the learning rule based on BP rule, we prove that the learning of new generalized fuzzy CMAC converges to the least square error if the number of iterations approaches infinity and the learning rate approaches 0.

Further research work is required in order to speed up this convergence. How can we revise the learning rule so that the generalized fuzzy CMAC can linearly converge to the least square error. This work is in progress.

References:

[1] Albus, J. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). Journal of Dynamic Systems, Measurement, and Control Transactions of ASME, 1975,(97):200～227.

[2] Glanz, F. *et al.* An Overview of CMAC Neural Network. In: Proceedings of Conference Neural Networks for Ocean Engeering, Washington, 1991. 301～308.

[3] Lane, S. *et al.* Development of high-order CMAC neural network. IEEE Control Systems Magazine, 1992,8(12):73～82.

[4] Lee, S. *et al*. A Gaussian potential neural network with hierarchically self-organizing learning. Journal of Neural Networks, 1991,4(1):207~224.

[5] Wong, Y. *et al*. Learning convergence of CMAC. Journal of IEEE Neural Networks, 1992,2(1):115~121.

[6] Chiang, C. T. *et al*. Fuzzy Systems and Fuzzy Neural Networks. Prentice-Hall, Inc., 1997.

[7] Lin C S *et al*. Learning convergence of CMAC technique. Journal of IEEE Neural Networks, 1997,8(6):1032~1042.

[8] Xu H, Kwan, C. M. Real-time adaptive on-line traffic incident detection. Fuzzy Sets and Systems, 1993,93(2):173~183.

[9] Wang Shi-tong. Neural Fuzzy Systems and Their Applications. Beijing: Publishing House of Beijing University of Aeronautics and Astronautics, 1998.

关于广义模糊 CMAC 学习收敛性的理论结果

王士同[1,2], Baldwin, J.F.[2], Martin, T.P.[2]

[1](华东船舶工业学院 计算机系,江苏 镇江 212003)

[2](英国 Bristol 大学 高级计算研究中心,英国)

摘要: 提出了广义模糊 CMAC(cerebellar model articulation controller)神经网络,并导出了其学习的充分条件.最后,证明了广义模糊 CMAC 在平方误差意义下的学习收敛性.研究结果为广义模糊 CMAC 的广泛应用提供了基础.

关键词: CMAC(cerebellar model articulation controller);学习收敛;模糊泛集合;学习规则

中图法分类号: TP183　　**文献标识码**: A