Proceedings of the 5th World Congress on Intelligent Control
and Automation, June 15-19, 2004, Hangzhou, P.R. China

第五届全球智能控制与自动化大会
2004 年 6 月 15 - 19 日，中国杭州

A Hybrid One-by-One Learning with Incremental Learning Algorithm in CMAC *

Lei Zhang and Qixin Cao
Research Institute of Robotics
Shanghai Jiao Tong University
Shanghai, 200030, China
Zhanglei75@sina.com

Jay Lee
Research Center of Intelligent Maintenance Systems
University of Wisconsin-Milwaukee
Milwaukee, WI 53224, USA
jaylee@uwm.edu

***Abstract* - A one-by-one learning algorithm similar to traditional incremental learning for CMAC is suggested. The convergence property is investigated based on the principles of geometric sequence and iteration theory of linear equations. The sufficient condition for the convergence of the algorithm is the same as that of incremental learning. The performance of two algorithms is compared, then a hybrid one-by-one learning with incremental learning algorithm is proposed. The simulation results about two-dimension function approximation prove the hybrid algorithm has better performance in convergent speed and precision.**

Index Terms* - *one-by-one learning; incremental learning; a hybrid one-by-one learning with incremental learning; CMAC

I. INTRODUCTION

Cerebellar Model Articulation Controller (CMAC) network was first developed in the 1970s by Albus [1] as a control method based on the principles of the cerebellum's behaviour. The main advantages of CMAC against MLP, RBF, etc. networks are it's local generalization, extremely fast learning and easy implementation in software and hardware, so it has been widely used in many fields, especially in real-time control in robotics and other industrial control fields [2-4].

According to the way of error correction, there are two basic schemes in CMAC training. One is incremental learning and the other is batch learning. In batch learning, all the information about the training data points in one cycle should be known, and the range of learning rate is difficult to determine. These limit the usage of this method. According to the way of selecting training samples, incremental learning and batch learning both belong to cyclic learning and all training samples are repeated in many cycles. Kwon [5] developed a random training method in which the training samples are selected in random fashion. David E. [6] pointed out that random training required relatively long training periods to reach a desired performance level, and he suggested a method termed as neighborhood sequential training. In this method, training points that lie outside of the neighborhood of the previous training points are chosen. It is usually impossible if the function to be learned is unknown. So developing general training techniques with better convergent performance for CMAC is important, however, little work has been done.

In this paper, we first present a learning algorithm named as one-by-one learning and investigate its convergence property. Then by comparing one-by-one learning with incremental leaning, a hybrid one-by-one learning with incremental learning algorithm is proposed. The simulation of two-dimension function approximation is carried out to compare the performance of these three algorithms and conclusions are given at last.

II. ONE-BY-ONE LEARNING ALGORITHM

CMAC can be considered as an associative memory network which performs two subsequent mappings: $f: X \rightarrow A$, $g: A \rightarrow P$, where X is a m-dimension input space, A is an association vector, which contains N_A elements. W is the weight vector and P is the output space. The first mapping projects an input space point X_k into a binary associative vector A_k. Only the cells which are activated by X_k is set to "1" and others are set to "0". The second one calculates the output of the network as scalar product of the association vector A_k and the weight vector W:

$$y_k = A_k \cdot W = \sum_{i=1}^{N_A} a_{k,i} w_i \tag{1}$$

where k means the k-th state, the weights are updated as the following

$$W(j+1) = W(j) + \frac{\beta * A_k}{g}(y_{d,k} - y_k) \tag{2}$$

where j means the j-th cycle, β is the learning rate, g is the number of activated association cells (the generalization parameter), and $y_{d,k}$ is the desired output at the k-th sample.

Incremental learning is often used for training CMAC, in which each training sample is presented in turn and the weights are updated at each presentation according to (2). As we can see, the error of every sample is corrected only one time in one cycle. So the actual output is still far away from the desired one. We suggest a learning method named as one-by-one learning. It is also a cyclic learning. The only difference between one-by-one learning and incremental

* This work is supported by NSFC Grant #50128504 to Qixin Cao and Jay Lee

learning is, in the former, the error correction is repeated for each sample until the error is smaller than the defined value aforehand. This repeated error correction may result in longer learning time, but it is found in our study, the convergence speed isn't reduced significantly only if the learning rate isn't too small. By this way, the error of every sample will be reduced to the minimum, so we wish the total error of all samples after each cycle will be reduced also. But through detailed analysis in section Ⅳ, we drew an interesting conclusion, and which enlightened us to get a hybrid one-by-one learning with incremental learning algorithm.

Ⅲ. The Convergence Of One-by-One Learning

In this section, we will discuss the convergence property of one-by-one learning. Suppose that W is set to a null vector before learning begins. $(X_1, y_{d,1})$ is the first training sample. The defined permitted error is e_p. When X_1 is presented into CMAC in first cycle. The output will be 0, so the error is $y_{d,1}$, corresponding weighs are updated based on (2). After the first update, the error will turn into $y_{d,1} - \beta * y_{d,1} = y_{d,1} * (1-\beta)$. After the second update, the error will turn into $y_{d,1} - \beta * y_{d,1} - \beta * (1-\beta) * y_{d,1} = y_{d,1} * (1-\beta)^2$, and the rest may be deduced by analogy. The values of different items are listed in table 1 to show this process.

As can be seen from table1, the correction error in each update composes a geometric sequence. The fist item of this geometric sequence is $\beta * y_{d,1}$, and $1-\beta$ is the geometric parameter. According to the principles about geometric sequence, after n times update, the sum of the geometric sequence is

$$S_n = \frac{\beta * y_{d,1} * (1-(1-\beta)^n)}{1-(1-\beta)} \quad (3)$$

It is obvious that if the geometric parameter $1-\beta$ fits inequality,

$$|1-\beta| < 1, \rightarrow \quad 0 < \beta < 2 \quad (4)$$

TABLE I

THE LEARNING PROCESS OF THE SAMPLE $(X_1, y_{d,1})$

Update time	Actual output	Error	Correction error
1	0	$y_{d,1}$	$\beta * y_{d,1}$
2	$\beta * y_{d,1}$	$(1-\beta) * y_{d,1}$	$(1-\beta) * \beta * y_{d,1}$
3	$\beta * y_{d,1} +$ $(1-\beta) * \beta * y_{d,1}$	$(1-\beta)^2 * y_{d,1}$	$(1-\beta)^2 * \beta * y_{d,1}$
…	…	…	…
n+1	$\beta * y_{d,1} +$ $(1-\beta) * \beta * y_{d,1} +$ $\cdots +$ $(1-\beta)^n * \beta * y_{d,1}$	$(1-\beta)^n * y_{d,1}$	$(1-\beta)^n * \beta * y_{d,1}$

The sequence is convergent. If the permitted error e_p is very small, such as e_p=0.0001, after n times update, the error of the sample will be smaller than e_p ($n \rightarrow \infty$ if $e_p \rightarrow 0$, $\beta \neq 1$), then

$$\lim_{n\to\infty} S_n = \lim_{n\to\infty} \frac{\beta * y_{d,1} * (1-(1-\beta)^n)}{1-(1-\beta)} \approx y_{d,1}$$
$$(0 < \beta < 2, \beta \neq 1) \quad (5)$$

It can be seen the sum of this sequence is no other than the initial output error of the sample $(X_1, y_{d,1})$, so the desired output will converge to the actual output. If $\beta = 1$, the error correction is one-shot process. The same conclusion can be drawn from other training samples. In general, we can get theorem 3.1.

THEOREM 3.1 In one-by-one learning algorithm, if $0 < \beta < 2$, the real output of every sample is convergent to the desired one in each cycle.

In the following section, we will investigate the convergence of one-by-one learning after many cycles when $0 < \beta < 2$. Another description of CMAC algorithm proposed by Wong [7] is suitable to analyse the learning convergence, the goal of CMAC learning is to find a set of weights W such that

$$AW = Y \quad (6)$$

$A = [A_1, A_2, \cdots A_N], Y = [y_{d,1}, y_{d,2}, \cdots y_{d,N}]^T$, N is the number of all samples. Wong proved the CMAC learning rule was equivalent to find the solutions of the linear system (7) if the initial weights are set to zero.

$$C\Delta = Y \quad (7)$$

where $C = AA^T$, called as the correlation matrix, $\Delta = [\Delta_1, \Delta_2, \cdots \Delta_N]^T$, more details can be referred to [7]. In incremental learning, $\Delta_i = \sum_l \beta\delta_i^{(l)} / g$, δ_i^l is the output error when the i-th sample is presented in the l-th cycle, but in one-by-one cyclic learning, $\Delta_i = \sum_l \delta_i^{(l)} / g$. The update rule of the l-th cycle can be written as

$$\Delta_i^{l+1} = \Delta_i^l + \frac{1}{g}(y_{d,i} - \sum_{j=1}^{i-1} c_{ij}\Delta_j^{l+1} - \sum_{j=i}^{N} c_{ij}\Delta_j^l) \quad (0 < \beta < 2) \quad (8)$$

It can be seen that (8) is the same as the equation in incremental learning when $\beta = 1$. Wong had proved that when $\beta = 1$, learning scheme expressed by (8) in incremental learning is equivalent to the Gauss Siedel iteration of linear equations (7), then we have lemma 3.1.

LEMMA 3.1 If A is a positive definite matrix, Gauss-Seidel iteration of the linear coupled equations $Ax=b$ is convergent

THEOREM 3.2 If C is a positive definite matrix, and $0 < \beta < 2$, one-by-one learning scheme is convergent.

Proof. It can be easily obtained from theorem 3.1 and lemma 3.1.

IV. A Hybrid One-by-One Learning with Incremental Learning Algorithm

From theorem 3.2, we can see the sufficient condition for the convergence of one-by-one learning is the same as that for traditional incremental learning. When $\beta=1$, one-by-one learning is completely equivalent to incremental learning. By simulations of several functions approximation, we find in one-by-one learning, although the error of every sample is reduced to the minimum after it is presented, the learning of subsequent samples will destroy this precision because of "learning interference". The error of the sample after all the samples have been trained in each cycle is still very large. We also find when $\beta>1$, the total error of all samples after the same cycle in one-by-one learning is smaller than that in incremental learning. However, when $\beta<1$, a contrary conclusion is drawn. The theoretic explanation of this phenomenon is difficult, so qualitative analysis is given.

In one-by-one learning, as for sample ($X_i, y_{d,i}$) in the l-th cycle, the learning interference of subsequent samples is $|\sum_{j=i+1}^{N} c_{ij}\delta_j^l / g|$, c_{ij} is the element in matrix C. In incremental learning, the corresponding item is $|\beta * \sum_{j=i+1}^{N} c_{ij}\delta_j^l / g|$. So it is obvious, if δ_j^l is the same, the learning inference in one-by-one learning is smaller than that in incremental learning when $\beta>1$. So total error of all samples after each cycle is smaller. The contrary conclusion can be obtained by the same way when $\beta<1$. In fact, δ_i^l isn't the same in two learning algorithms, but we can consider that the value of β plays a more important role than the difference of δ_i^l in learning interference.

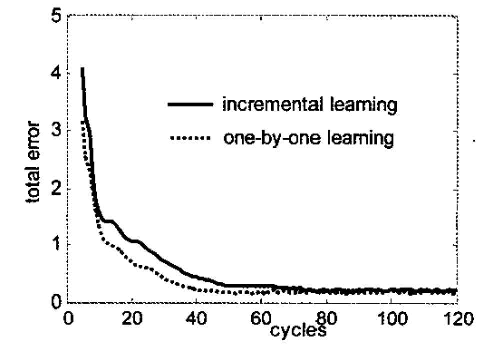
This conclusion enlightens us that, if the learning parameter is adjusted dynamically, a hybrid one-by-one learning with incremental learning algorithm will get much better performance. Dynamic adjusting learning rate has been suggested by many researchers and proved a good way to improve the convergent performance of general CMAC algorithms [8-9]. The hybrid algorithm is realized as follows, first $\beta=\beta_0, (1<\beta_0<2)$, then training samples are learned in order, when cycle >2, if $TE(cycle)<TE(cylce-1)$, $\beta=\beta$, otherwise $\beta=0.8*\beta$, if $\beta\geq 1$, one-by-one learning algorithm is used, when β is reduced to be smaller than 1, incremental learning is used. TE means total error and is defined as $TE=\sum_{i=1}^{N}|e_i|=\sum_{i=1}^{N}|y_{d,i}-y_i|$

V. Simulation

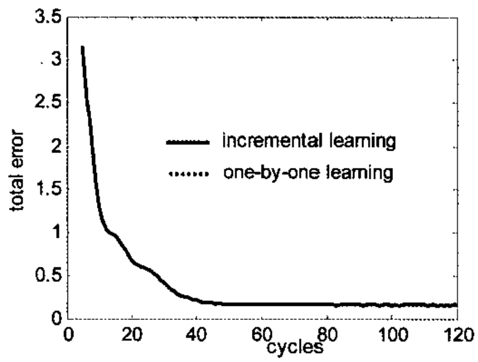
A two-dimension function $z(x,y)$ is used to compare the learning properties of incremental learning, one-by-one learning and the hybrid learning algorithm.

$$z(x,y)=\sin(2\pi x/360)*\cos(2\pi y/360), 0\leq x,y\leq 50 \quad (9)$$

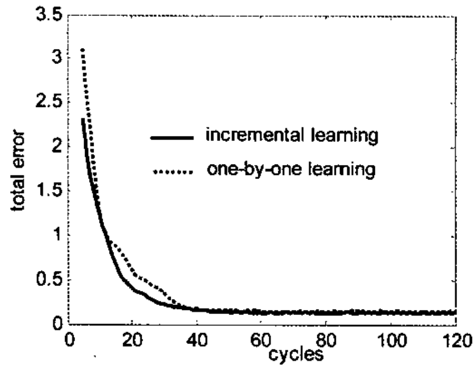
The training samples are $z=z(x,y)$, $x,y\subset[0,4,8,12,\cdots 48]$, and quantization resolutions for x and y are both 2, the generalization parameter $g=8$. Fig.1 (a), (b) and (c) show the total error of all samples versus cycles when $\beta=1.5, \beta=1.0, \beta=0.5$, cycle is from 5 to 120. Total error is defined as in section IV. It can be seen, when $\beta=1$, two error curves overlap completely. When $\beta>1$, one-by-one learning is better than incremental learning in convergent speed and precision. When $\beta<1$, the contrary conclusion is drawn. Fig.1 (d) shows the comparison of three algorithms when the learning rate is adjusted dynamically and the initial value $\beta_0=1.5$. Cycle is from 40 to 120. It can be seen although the convergent speed in one-by-one learning is faster than incremental learning, the convergent precision is worse at last. In the hybrid algorithm, learning rate is becoming smaller than 1 when cycle=46, then one-by-one learning is replaced by incremental learning. It can be seen that the hybrid algorithm has better performance than other two algorithms in convergent speed and precision.



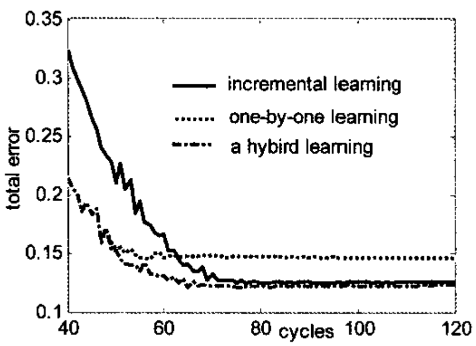
(a) $\beta=1.5$



(b) $\beta=1.0$



(c) $\beta = 0.5$



(d) β is adjusted dynamically

Fig.1 The comparison of convergence property of different algorithms

VI. CONCLUSIONS

We first suggest a one-by-one learning algorithm for CMAC and prove if C is a positive define matrix, and $0 < \beta < 2$, the proposed learning scheme is convergent. This sufficient condition for the convergence is the same as that of traditional incremental learning. By comparing the performance of two algorithms, a hybrid one-by-one learning with incremental learning algorithm is proposed. A two-dimension function approximation is used to compare these three different algorithms and the simulation results prove the hybrid algorithm is better than the others in convergent speed and precision. The hybrid algorithm has no additive limits, so can replace incremental learning and be widely used.

REFERENCES

[1] J. S. Albus, "A new approach to manipulator control; the Cerebellar Model Articulation Controller(CMAC)," *Journal of Dynamic System, Measurement and Control*, pp.220-233, Sept. 1975.

[2] W. T. Miller, P.R. Hewes, and F.H. Glanz, "Real-time dynamic control of an industrial manipulators using a neural network-based learning controllers," *IEEE Trans. Robot. Automat*, vol. 6, no.1, pp.1-9, Feb.1990

[3] C. S. Lin and H. Kim, "CMAC-based adaptive critic learning control," *IEEE Trans. Neural Network*, vol.2, pp. 530-535, Sept.1991.

[4] S. F. Su, Ted Tao and T. H. Hung, "Credit Assigned CMAC and its application to online learning robust controllers," *IEEE Trans. Systems, Man, and Cybernetics-part B: cybernetics*, vol. 33, no.2, pp.202-218, April 2003.

[5] S. Kwon, "An adaptive control system for biological and robotic simulation," Ph. D. dissertation, Dep. Mech. Eng. Louisiana State Univ., Dec. 1990.

[6] D. E. Thompson, "Neighborhood sequential and random training techniques for CAMC," *IEEE Trans. Neural networks*, vol. 6, no.1, pp.196-202, Jan. 1995.

[7] Y. Wong and A. Sideris, "Learning convergence in the cerebellar model articulation controller," *IEEE Trans. Neural Networks*, vol.3, pp.115-121, Jan. 1992.

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

[9] H. C. Lu and J.C. Chang, "Enhance the performance of CMAC neural network via fuzzy theory and credit apportionment," *Neural networks, 2002, IJCNN'02, Proceedings of the 2002 International Joint Conference on*, vol.1, pp.715-720, 12-17 May 2002,