A Novel Learning Framework of CMAC With Area-time Credit Assignment and Grey Learning Rate

Po-Lun Chang[1]
[1] Department of Electrical Engineering
National Taiwan University of Science and Technology
Tel:02-29476559
Fax:02-86686892
dirty@mail.lhu.edu.tw

Horng-Lin Shieh[2]
[2]Department of Electrical Engineering
St. John's University
Associate Professor
Tel:02-28013131-6146
Fax:02-28013973
shieh@mail.sju.edu.tw

Ying-Kuei Yang[3]
[3]Department of Electrical Engineering
National Taiwan University of Science and Technology
Associate Professor
Tel: 02-27376681
Fax: 02-27376699
ykyang@mouse.ee.ntust.edu.tw

Abstract

The advantages of CMAC neural network are fast learning convergence, capable of mapping nonlinear functions quickly due to its local generalization of weight updating, simple architecture, easily processing and hardware implementation. In the training phase, the disadvantage of some CMAC models with a larger fixed learning rate is the unstable phenomenon. The smaller learning rate would cause slower convergence speed. In the aspect, we propose grey learning rate for training phase. We incorporate the grey relational analysis with the number of training iterations to get an adaptive learning rate for better response. In addition, a serious problem of learning interference reduces learning speed and accuracy. The idea is that the error correcting must be proportional to the inverse of learning times and trained input area for the addressed hypercube. An area-time credit assignment adopts the idea to provide fast and accurate learning effects. This paper proposes a novel learning framework of CMAC for the best performance. From the simulation results, it is clear that the proposed algorithm provides more accurate and fast convergence.

Keywords: CMAC, Learning interference, grey relational analysis, area-time credit assignment

摘要

小腦模型具有快速學習收斂，快捷區域性更新加權值對應非線性函數的能力，簡單的結構及容易處理與硬體實現的優點。在學習訓練階段，有些的CMAC 模型使用固定較大的學習率具有不穩定現象的缺點，使用較小的學習率則會造成比較慢的收斂速度。因此在所有學習階段，我們提出灰色理論動態調整學習率的方式，採用灰關聯分析結合訓練的次數以得到適當的學習率及較佳的響應。除此之外，存在學習干擾的問題會明顯減少學習的速度與準確度，利用學習信賴度具有與超立方塊的學習次數及已學習區域成反比例的關係，我們提出學習區域-次數加權值信賴分配的方式來達到快速且準確的學習效果。本論文提出新的 CMAC 學習架構來產生最佳的學習效能，由實際模擬結果可知，本文提出的演算法確實可提供 CMAC 模型更快速與準確的學習特性。

關鍵詞：小腦模型，學習干擾，灰關聯分析，學習區域-次數加權值信賴分配。

1. Introduction

Cerebellar Model Articulation Controller(CMAC) was first provided by J.S. Albus in 1975[1][2]. The CMAC model has good capability of local generalization and simple structure for hardware implementation [3].

In the relational theories of CMAC, Thompson and Kwon presented that neighborhood sequential training and random training methods were two training policies to accelerate the learning rate and accuracy of CMAC model[4]; In the improved architecture of conventional CMAC, Yeh and Lu performed grey relational analysis and adaptive quantization about learning space of CMAC [5][11]; Lee丶Chen and Lu presented a hierarchical CMAC that utilized hierarchical architecture and non-uniformly quantized input space for the application of classifier[6]. Lin and Chiang had proposed the convergent features of CMAC [7].

In the related research of learning rate and accuracy, Su丶Tao丶Hung[8][10] and Lu丶Chang[9] all made use of credit assignment to reform the learning strategy of CMAC. They performed credit assignment to distribute learning errors and enhanced the training speed of conventional CMAC. Su丶Tao and Hung presented a solution of credit assignment in accordance to learning times, it could accelerate the learning speed but result in the unstable phenomenon. Lu and Chang considered credit assignment to the mapping hyper cubes of input states and neighborhood states in each training time for increasing learning speed and accuracy but got only little improvement.

In the training phase of conventional CMAC, the learning errors were dispatched to mapping memory cells of input states by uniform distribution. There were existent effects of *learning interference* to reduce learning rate and accuracy in the training phase [1][4][8][9]. Our goal of research is to reduce the influence of *learning interference* and expects to reach the best learning speed and output accuracy. In this paper, we propose a novel learning framework that merges the concepts of area-time credit assignment and grey learning rate in the CMAC model that is really made to reach better performance. In the training phase, we compute the grey relational coefficients for each input state after current iteration and consider the number of learning iterations to get an appropriate grey learning rate for better response. The idea of area-time credit assignment is adopted to provide fast and accurate learning by distributing credits proportional to the inverse of learning times and trained input area of the addressed hypercube. Such a learning scheme can reach expected target in our paper.

The rest of this paper is organized as follows. In Section 2, basic concepts of CMAC model is briefly introduced as part of the proposed approach. In Section 3, a novel learning framework of CMAC model is thoroughly discussed. Experimental simulation results are presented in Section 4. Finally, conclusions are described in Section 5.

2. Basic Concepts of CMAC model

The basic structure of conventional CMAC model is shown in figure 1. The CMAC model defines a series of mapping procedures and repeats the training process to achieve learning objective. The first, it needs to determine the domain of learning space, then quantizes the learning space into many discrete states that are input states of the CMAC model and are represented to a set **S** in figure 1. Each input state is mapped from indexed memory **A** to corresponding real memory cells **W** that store the information of input states and are summed into actual output value. If the halting condition is not true then take the error of actual value and desired output value to update the contents of mapping memory cells **W** by uniform distribution. Repeat the above training phase to reach correct output values.

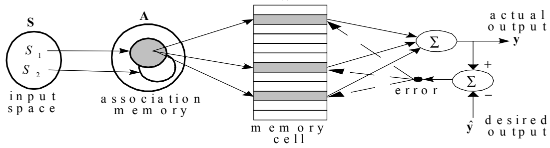

Figure1. Basic structure of CMAC model

An instance of two dimensional CMAC is shown in Figure 2. The input variables S1 and S2 are quantized into 21 units that are named as *elements* from number 0 to number 20. The width of each element is the resolution of quantization. The discrete elements are input states in the axis. A *block* is composed of 5 elements. There are 21 elements that are divided into 5 blocks including 4 complete blocks {A, B, C, D} and 1 residue block {E} in the S1 axis. The L1 layer is formed with blocks {A, B, C, D, E}. The L2 layer could be formed with the L1 layer by shifting one element, and by the way would reach 5 layers. The same art of composition is in the S2 axis. The mapping block is a hyper cube in the S1 axis and S2 axis. Total hyper cubes are real memory cells that store relational information about input states.

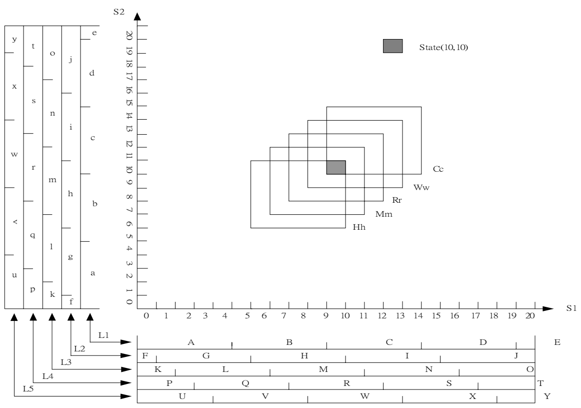

Figure2. The memory cells of two dimensional CMAC

If there are Nh hyper cubes and each input state is made use of Ne hyper cubes, then the actual output is shown in the equation (1). The y_s is the actual output of input states, $\mathbf{a}_s^T$ is indexed vector, and $\mathbf{w}$ is memory cell vector.

$$y_s = [a_{s,1}, a_{s,2}, \cdots, a_{s,N_h}] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N_h} \end{bmatrix} = \mathbf{a}_s^T \mathbf{w} = \sum_{j=1}^{N_h} a_{s,j} w_j \qquad (1)$$

CMAC is one class of supervised learning neural network models. In learning phase, the error of actual and expected output value is accorded to regulate and train the memory cells of CMAC by uniformly updated. The relation is shown in the equation (2).

$$w_j^{(i)} = w_j^{(i-1)} + \alpha \cdot a_{s,j} \cdot \frac{(\hat{y}_s - \mathbf{a}_s^T \mathbf{w}^{(i-1)})}{Ne} \qquad (2)$$

In the above equation(2), s is an input state, $w_j^{(i)}$ is the hyper cube of number *j* in the training time of number i, $a_{s,j}$ is an index of input state s and hyper cube of number *j*, $(\hat{y}_s - \mathbf{a}_s^T \mathbf{w}^{(i-1)})$ is learning error, α is learning rate, and each input state is made use of Ne hyper cubes.

In the structure of CMAC model, the hyper cubes are shared by many relational input states, and neighborhood states share more hyper cubes. There are existent effects of *learning interference* to reduce learning speed and accuracy in the training phase [1][4][8][9][10][11]. The smooth functions have less influence of *learning interference* but other nonlinear functions have important influence.

3. A Novel Learning Framework

The goal of our research is to reduce the influence of learning interference and expects to increase learning speed and accuracy. We apply the concepts of credit assignment[8][10] and combine the adaptive regulation of learning rate in accordance to grey relational analysis of input states and the number of training iterations in the CMAC model that can really reach better performance than the past CMAC models.

3.1. Grey relational analysis

Grey relational analysis [11] is a kind of similarity measure method. This analysis scheme is essentially believed to have captured the relationship between the main factor and the other reference factors in a given system. Due to this feature, grey relational analysis can be applied to relate the reference sequence to the comparison sequences, which can show some degrees of satisfaction to the reference one, such that the best comparison sequence can be determined. Hence, Grey relational analysis is a simple approach to analyze the similarity of the considered sequences. Assume that a single reference sequence is given by $x_0 = \{x_0(1), x_0(2), ..., x_0(n)\}$ and m comparison sequences are $x_i = \{x_i(1), x_i(2), ..., x_i(n)\}$, $i = 1,2,...,$ m, then the ***grey relational coefficient*** between x_0 and x_i at the k-th state is defined as follows.

$$GRC\ (x_0(k), x_i(k)) = \frac{\Delta_{\min} + \xi \cdot \Delta_{\max}}{\Delta_{0i}(k) + \xi \cdot \Delta_{\max}} \tag{3}$$

where $GRC\,(x_0(k), x_i(k))$ is termed the grey relational coefficient, $\Delta_{0i}(k) = |x_0(k) - x_i(k)|$, $\xi \in (0,1]$ is the distinguishing coefficient to control the resolution between $\Delta_{\max}$ and $\Delta_{\min}$, $\Delta_{\max} = \max_i \max_k \Delta_{0i}(k)$, and $\Delta_{\min} = \min_i \min_k \Delta_{0i}(k)$.

By this definition, each ***grey relational coefficient*** will satisfy $0 < GRC\ (x_0(k), x_i(k)) \leq 1$.

3.2. Grey learning rate

After continuous training phase, the contents of hyper cubes could be approximate to correct value. If the learning rate of CMAC is set to larger value, then CMAC could be fast convergence, lower accuracy and in the unstable phenomenon. If the learning rate of CMAC is set to smaller value, then CMAC could be slower convergence and better accuracy. As previous demonstrations, we investigate a new adaptive technique for learning rate on the basis of grey relational analysis in our paper.

Assume that the input space is partitioned into n states and the desired function $\hat{y}$ to be learned is known, then the desired output of a specific input state, say s_k, $k = 1,2,...,n$, can be mathematically expressed as $\hat{y}(k)$. Gathering these n desired outputs can form a reference sequence as follows. $\hat{Y} = \{\hat{y}(1), \hat{y}(2), ..., \hat{y}(n)\}$. To analyze the grey relation between the desired outputs and the actual one, the comparison sequence here is generated by the actual output of CMAC at every state, denoted as follows. $Y = \{y(1), y(2), ..., y(n)\}$. According to equation (3), the grey relational coefficient at state s_k for a single comparison sequence can be easily rewritten as

$$GRC\ (y(k), \hat{y}(k)) = \frac{\Delta_{\min} + \xi \cdot \Delta_{\max}}{\Delta_y(k) + \xi \cdot \Delta_{\max}} \tag{4}$$

Where $\Delta_y(k) = |y(k) - \hat{y}(k)|$, $\Delta_{\max} = \max_k \Delta_y(k)$, $\Delta_{\min} = \min_k \Delta_y(k)$, and $0 < GRC\ (y(k), \hat{y}(k)) \leq 1$.

The grey relational coefficient, as defined in equation (4), can not only reflect the relation between the reference sequence and the comparison one at state s_k, but also show the degrees of output error. That is to say, the smaller the output error is, the larger the grey relational coefficient is. The main feature using the grey relational coefficient to stand for the degree of variation of the output error is that it can be easily computed at the end of each of learning iterations. After a complete learning iteration, the error information is examined. If the results are unsatisfied, the learning rate will be adjusted based on the learned information. The central idea of the proposed adaptive strategy for learning rate is demonstrated as follows.

The actual outputs of all states can be easily obtained after a complete learning iteration. Therefore, we can utilize equation (4) to compute corresponding grey relational coefficients. In the learning phase, the errors of hyper cubes are also related to the inverse of training iterations. This paper presents the adaptive regulation of learning rate in accordance to training iterations and grey relational coefficients in the CMAC model. In the i-th training iteration, the grey learning rate at state s_k is defined as

$$grey\_\alpha(i,k) = i^{-GRC(i-1,k)} \tag{5}$$

Where $GRC(i-1,k)$ is termed the grey relational coefficient at state s_k in the *(i-1)*-th training iteration. Initially, $GRC(0,k)=1$ and $grey\_\alpha(1,k)=1$. The value of equation (5) is depicted in the Table 1.

Table 1. The variation of $grey\_\alpha(i,k)$

$GRC(i-1,k)$ / i	0.1	0.3	0.5	0.7	1.0
2	0.93	0.81	0.71	0.62	0.5
3	0.9	0.72	0.58	0.46	0.33
4	0.87	0.66	0.5	0.38	0.25
5	0.85	0.62	0.45	0.32	0.2

3.3. Area-time Credit Assignment

A hypercube that includes more input states would be more influence of *learning interference*. In order to avoid such effects, the distribution of errors into the addressed hyper cubes must be proportional to the creditability of those hyper cubes. The important information that can be used is how many times the hyper cubes have been updated [8][10]. In our paper also consider the trained proportion of input states in the hypercube at current iteration. In a training iteration, the feature is that the more input states in the current hyper cube have been trained, the more accurate the stored value is. Hence, the times of updating for hyper cubes and the trained proportion of input states in the hypercube can be merged as the creditability of those hyper cubes.

We define $area\_time\_credit(j)$ as the follow equation

$$area\_time\_credit(j) = \frac{(f(j)\times area(j)+1)^{-1}}{\sum_{c=1}^{m}(f(c)\times area(c)+1)^{-1}} \quad (6)$$

Where $f(j)$ is the learning times of the *j*-th hyper cube, and *m* is the number of addressed hyper cubes for a state. Notice, that $f(j)\times area(j)$ must include one to prevent dividing by zero.

$$area(j) = \frac{the\ trained\ input\ states + 1}{total\ states\ of\ the\ jth\ hypercube} \quad (7)$$

Where $area(j)$ is the trained proportion of input states in the *j*-th hypercube at present iteration.

Using the above idea, our paper modifies the weight updating by area-time credit. It is shown that equation (2) is rewritten as

$$w_j^{(i)} = w_j^{(i-1)} + \alpha \cdot a_{s,j} \cdot (\hat{y}_s - a_s^T w^{(i-1)}) \cdot area\_time\_credit(j) \quad (8)$$

It is noted that the credit assignment scheme has obvious effects in the early stage of learning.

3.4. A Novel Learning Framework

The equation (9) depicts area-time credit assignment and grey learning rate for updating hyper cubes. It changes from the learning rate α of equation (8) to $grey\_\alpha(i,k)$ of equation (5).

$$w_j^{(i)} = w_j^{(i-1)} + grey\_\alpha(i,k) \cdot a_{s,j} \cdot (\hat{y}_s - a_s^T w^{(i-1)}) \cdot area\_time\_credit(j) \quad (9)$$

Such a learning scheme is termed the *Area-Time-Credit CMAC*. It is noted that the proposed learning framework has fast and accurate effects in the stage of training.

4. Simulation Results

In this section, the experimental results show the performance of different CMAC models using two examples with various functions. *Time-Credit CMAC* is from references [8][10]. *Area-Time-Credit CMAC* is the proposed approach of our paper.

Example1. Objective function is $\sin x / x$. The distinguishing coefficient $\xi = 0.1$. CMAC architecture: Input states: 50. A state: 3 memory cells. Training cycle: 10 times. Learning comparison: root mean square error (RMSE).

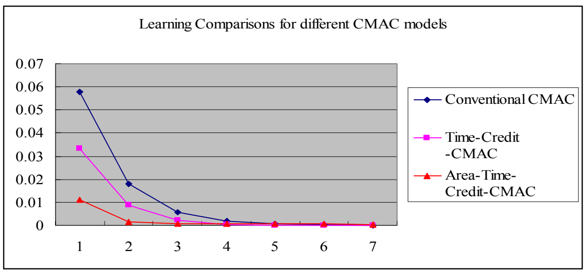

Figure3. Learning Comparisons for different CMAC models

Example2. Objective function is $z(x,y) = (x^2 - y^2)\sin 5x$ for $-1 \le x \le 1$, $-1 \le y \le 1$

The distinguishing coefficient $\xi = 0.1$ .CMAC architecture: 57 elements in the axis. Input states: 3249(57*57). A block: 7 elements. Hyper cubes: 567(7*9*9). Training cycle : 10 times. Learning comparison: root mean square error (RMSE).

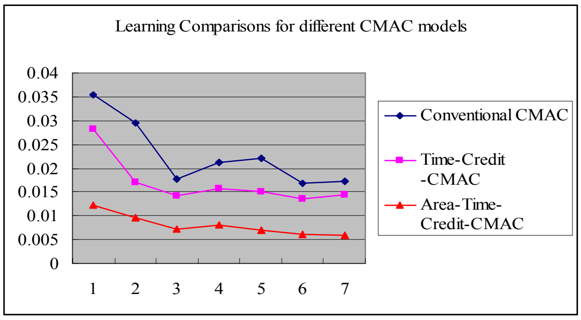

Figure4. Learning Comparisons for different CMAC models

From the above simulations, it is clear that the results can converge very fast to the neighborhood of desired values and accurate enough, even in one or two cycles.

5. Conclusions

This paper proposes an improved strategy which is a novel learning framework of *CMAC* model by incorporating credit dispatching error to memory cells with grey learning rate in accordance to the number of training iterations, grey relational analysis, learning times and trained area of the addressed hypercube. Above test results indicate that the developed method can provide more accurate and good convergence. We present a validly new scheme for CMAC model.

6. References

[1] J. S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)", *ASME J. Dynamic Systems, Measurement, Control*, pp.220-227, 1975.

[2] J. S. Albus, "Data Storage in the Cerebellar Model Articulation Controller (CMAC)", *ASME J. Dynamic Systems, Measurement, Control,* pp.228-233, 1975.

[3] W. T. Miller, F. H. Glanz and L. G. Kraft, "CMAC: An Associative Neural Network Alternative to Backpropagation", *Proceedings of the IEEE* , vol.78 , no.10 , pp.1561 – 1567, 1990

[4] David E. Thompson and Sunggyu Kwon, "Neighborhood Sequential and Random Training Techniques for CMAC", *IEEE Transaction on Neural Networks*, vol.6, no.1, pp.196-202 1995

[5] M. F. Yeh and H. C. Lu, "On-Line Adaptive Quantization Input Space in CMAC Neural Network", *IEEE International Conference on Systems, Man and Cybernetics*, vol.4, 2002

[6] H. M. Lee and C. M. Chen, "Self-Organizing HCMAC Neural-Network Classifier", *IEEE Transaction on Neural Networks*, vol.14, no.1, pp.15-27, 2003

[7] C. S. Lin and C. T. Chiang, "Learning Convergence of CMAC Technique", *IEEE Transaction on Neural Networks*, vol.8, no.6, pp.1281-1292, 1997

[8] S. F. Su, T. Tao, and T. H. Hung, "Credit Assigned CMAC and Its Application to Online Learning Robust Controllers", *IEEE Transaction on System, Man, and Cybernetics-Part B: Cybernetics*, vol.33, no.2, pp.202-213, 2003

[9] H. C. Lu and J. C. Chang, "Enhance the Performance of CMAC Neural Network via Fuzzy Theory and Credit Apportionment", *IEEE Proceedings of the 2002 International Joint Conference on Neural Networks*, vol.1, pp.715-720, 2002

[10] Shun-Feng Su, Zne-Jung Lee, and Yan-Ping Wang, "Robust and Fast Learning for Fuzzy Cerebellar Model Articulation Controllers", *IEEE Transaction on System, Man, and Cybernetics-Part B: Cybernetics*, vol.36, no.1, pp.203-208, 2006

[11] Ming-Feng Yeh and Kuang-Chiung Chang, "A Self-Organizing CMAC Network With Grey Credit Assignment", *IEEE Transaction on System, Man, and Cybernetics-Part B: Cybernetics*, vol.36, no.3, pp.623-635, 2006