JACET Journal of Advances in Computer Engineering and Technology, 1(1) 2015

Two Novel Learning Algorithms for CMAC Neural Network Basis on Changeable Learning Rate

Nazal Modhej[1], Mohammad Teshnehlab[2], Mashallah Abbasi Dezfouli[3]

Received (19-9-2011)
Accepted (22-11-2011)

***Abstract* - Cerebellar Model Articulation Controller Neural Network is a computational model of cerebellum which acts as a lookup table. The advantages of CMAC are fast learning convergence, and capability of mapping nonlinear functions due to its local generalization of weight updating, single structure and easy processing. In the training phase, the disadvantage of some CMAC models is unstable phenomenon or slower convergence speed due to larger fixed or smaller fixed learning rate respectively. The present research deals with offering two solutions for this problem. The original idea of the present research is using changeable learning rate at each state of training phase in the CMAC model. The first algorithm deals with a new learning rate based on reviation of learning rate. The second algorithm deals with number of training iteration and performance learning, with respect to this fact that error is compatible with inverse training time. Simulation results show that this algorithms have faster convergence and better performance in comparison to conventional CMAC model in all training cycles .**

***Index Terms* - CMAC, Learning rate, Training iteration, and Learning performance.**

1- Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Khouzestan-Iran (modhej@siau.ac.ir).
2- K. N.Toosi University of Technology (teshnehlab@eetd.kntu.ac.ir).
3- Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Khouzestan-Iran (m.abbasi@khouzestan.srbau.ac.ir).

I. Introduction

Cerebellar Model Articulation Controller (CMAC) was first provided by J.S. Albus in 1975[1],[2]. CMAC is originated from the biological cerebellum and it can be explained as its simple model. CMAC is an associative memory feedforward Neural Network with fast convergence in learning and output computation. CMAC has good capability of local generalization and simple structure for hardware implementation [5], [7], [8].

Several researches were conducted on different techniques of CMAC training. In [11] credit assignment is implemented using grey relational analysis that is an essential issue in grey system theory and can be imagined as a similarity measure for finite sequences.

A novel learning framework of CMAC via grey-area-time credit apportionment and grey learning rate is presented in [5]. Since the network training is basically optimizing technique, and CMAC network is for a given set of inputs, linear mapping, it can be shown that convex optimization procedure can be used to obtain optimal network weights. Reliability and simplicity of such algorithms in addition to the existence of a very large number of professional and shareware solvers are just some of the positive aspects of convex optimization approach [3]. Reinforcement learning control based on two-CMAC structure is explained in [4]. Reference [6] introduces active training for CMAC based on the active deformable model theory. Reference [15] introduces Tikhonov training for CMAC founded on Tikhonov regularization. Both are aimed at the problem of spars training sets.

As it was always mentioned, learning rate plays key role in training phase. In case of using large value for learning rate, then convergence is increased; however, instability is raised and in case of using small value for learning rate, then convergence is decreased. Whereas fixed learning rate is not regarded as suitable issue in training phase of CMAC model, the present research deals with offering methods for training phase of CMAC model that applies changeable learning rate. In so doing, two algorithms were proposed. The first one introduced a method for reaching a novel learning rate based on reviation of learning rate in each state of training phase of CMAC model. In the second algorithm, a new technique for learning rate was investigated on the basis of the learning performance and the number of training iterations. It was believed that such a learning schemes could reach a better performance than the learning method of conventional CMAC.

The reminder of this paper is organized as follows: Basic concept of CMAC neural network is briefly introduced in section 2. The novel learning frameworks of CMAC are thoroughly discussed in section 3. Simulation results are presented in section 4. And finally, conclusions are described in section 5.

Π. Structure of CMAC neural network

CMAC is a supervised learning structure that has fast learning ability and good local generalization capability for approximating nonlinear functions [5] [12] [13] [14].

In a CMAC network,each state variable is quatized and the problem space is divided into discrete states. A vector of quantized input values specifies a discrete state and is used to generate address for retrieving information from memory for this state [5].

The basic structure of conventional CMAC model is shown in fig. 1.

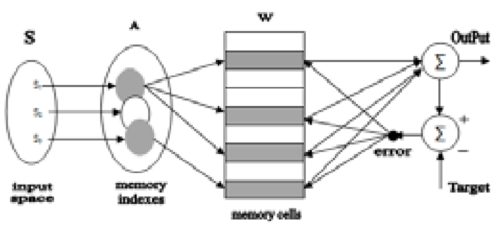

Fig. 1. Basic structure of CMAC model

The CMAC model defines a series of mapping procedures and repeats the training process to achieve learning objective. It needs to determine the domain of learning space, then quantizes the input space into many discrete states (S in fig. 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 [5].

The actual output, at state sk, is calculated as follows.

$$y(k) = a^T(k)\mathbf{W} = \sum_{j=1}^{Ne} a_j(k)W_j \qquad (1)$$

Where $a^T(k)$ is the indexed vector, W is the memory cell, and *Ne* is the number of hypercubes.

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

$$W_{new} = W_{old} + \frac{\alpha}{N_e} a(k)(y_d(k) - a^T(k)W_{old}) \qquad (2)$$

Where $y_d(k) - a^T(k)W_{old}$ is the learning error, and α is the learning rate that is $0 < \alpha < 1$.

An instance of two-dimensional CMAC is shown in fig. 2. The input variables s1 and s2 are quanized into 17 units that are named elements from number 0 to number 16. The discrete elements are input space in the axis. A block is composed of 4 elements. There are 17 elements that are devided 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 4 layers. The same scheme of composition is in the s2 axis. An exampler input state (7,8) activates hypercubes Bc, Hh, Mm, and Rr respectively in association memory layers as shown in fig. 2. Each memory cell considered to have a memory content and the output is computed as the sum of the contents of the activated hypercubes. This style of hypercube forming is called standard or diagonal addressing. Other addressing schemes are discussed in [9].

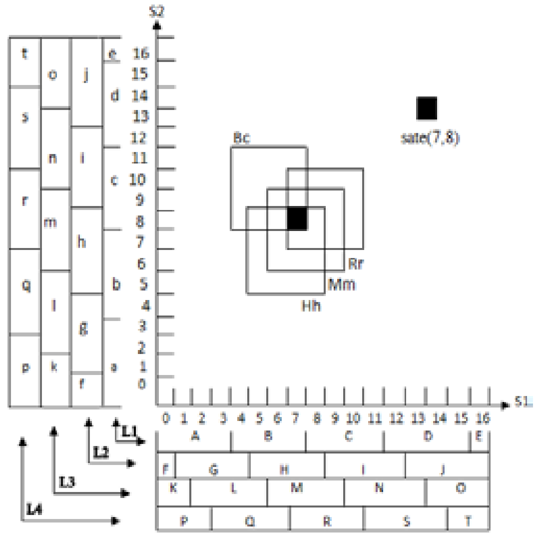

Fig. 2. The memory cells of two dimensional CMAC

III. Novel learning algorithms

As it was already explained, CMAC applies simple equation for updating the memory cells of the input states. In the training phase of conventional CMAC, the learning errors were dispatched to the mapping memory cells of input states by uniform distribution [5]. Goal of the present research is to reduce the influence of learning interference and expects to increase learning spead and accuracy. Two methods were applied for this purpose. The first algorithm deals with a novel learning method based on reviation of learning rate at any state of training phase, which expects to result in improving performance of learning. In the learning phase, the errors of hyper cubes are related to the inverse of training epoch. The second algorithm presents the adaptive regulation of learning rate in accordance to training epoch and learning performance for obtaining changeable learning rate and consequently better performance.

1. The learning algorithm with novel changeable learning rate for CMAC model

As it was explained, fixed learning rate in training phase causes instability or slow convergence. 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. In this section a changeable learning rate based on reviation of learning rate at any state of training phase, is offered. Reviation of learning rate at any state is calculated and these reviation are applied for calculating the novel learning rate, then this new learning rate is applied for updating the memory cells of CMAC in the training phase.

The reviation of the learning rate are computed with descending gradient through following formula:

$$\Delta\alpha = -\xi\partial E/\partial\alpha = -\partial(E/\partial y)(\partial y/\partial net)(\partial net/\partial\alpha)$$
$$.-\xi(-e(i))(1)(e(i-1))/Ne = \xi e(i)e(i-1)/Ne. \qquad (3)$$

In which *e(i)* and *e(i-1)* show the error in training iteration *(i)* and *(i-1)* respectively, and $0 < \xi < 1$.

Consequently, in the *i-th* training iteration, the learning rate at state s_k is defined as follows:

$$\alpha_1^i(k) = \alpha_1^{i-1}(k) + (\xi e^i(k)e^{i-1}(k)/Ne) \qquad (4)$$

Where $\alpha_1^i(k)$ and $\alpha_1^{i-1}(k)$ stand for the learning rate of state s_k, in training iteration *i* and *(i-1)* respectively.

By using such learning rate it is possible to update weights of CMAC model as follows:

$$w^i(k) = w^{i-1}(k) + \alpha_1^i(k)e^i(k)/Ne$$

In order to describe aforesaid equation an example is offered in following table. By having assumed amount of {0.0900, 0.0030, 0.0102, 0.0050} for relation of amounts that are obtained for learning rate with (4), are summarized at this table.

TABLE I : The variation of $\alpha_1^i(k)$

$\alpha_1^i(k)$ \ $\xi e^i(k)e^{i-1}(k)/Ne$	0.0900	0.0030	0.0102	0.0050
i=2	0.6800	0.5060	0.5204	0.5100
i=3	0.7700	0.5090	0.5306	0.5150
i=4	0.8600	0.5120	0.5408	0.5200
i=5	0.9500	0.5150	0.5510	0.5250

It would be found that the proposed learning framework has fast and accurate effects in all cycles of training phase from later simulation results.

The learning procedure is summarized as follows:

1) Initializes the memory contents and necessary parameters.

2) Determine $\xi e(i)e(i-1)/Ne$ *and* $\alpha_1^i(k)$.

3) Update the memory contents by the new learning rate.
4) Put i=i+1, and go to step 2 if assigned termination criterion is not satisfied.

2. Learning algorithm with novel hybrid learning rate for CMAC

As previous demonstrations, in the learning phase, the errors of hyper cubes are related to the inverse of training epoch. In this section a new adaptive technique for learning rate in accordance to training epoch and learning performance, is offered.

The learning performance equation, *J(k)*, is defined as[2]:

$$J(k) = 1/2((y(k) - y_d(k))^2 / Ne) \quad (5)$$

In the above equation, *y(k)* is actual output and $y_d(k)$ is desired output of state s_k.

In *i-th* training iteration, the novel learning rate, at state s_k is defined as:

$$\alpha_2^i(k) = i^{-J^{i-1}(k)} \quad (6)$$

Where $\alpha_2^i(k)$ stands for the learning rate of state s_k in the training time of i and $J^{i-1}(k)$ stands for the learning performance of CMAC model at state s_k in the training time of *(i-1)*.

By using such learning rate it is possible to update weights of CMAC model as follows:

$$w^i(k) = w^{i-1}(k) + \alpha_2^i(k) e^i(k) / Ne$$

In order to describe aforesaid equation an example is offered in following table. By having assumed amount of {0.1500, 0.5000, 0.6700, 0.9000} for relation of $J^{i-1}(k)$ amounts that are obtained for learning rate with (6), are summarized at this table.

TABLE Π : The variation of $\alpha_2^i(k)$

$\alpha_2^i(k)$ \ $J^{i-1}(k)$	0.1500	0.5000	0.6700	0.9000
i=2	0.9013	0.7071	0.6285	0.5359
i=3	0.8481	0.5774	0.4790	0.3720
i=4	0.8123	0.5000	0.3950	0.2872
i=5	0.7855	0.4472	0.3402	0.2349

The learning procedure is summarized as follows:

1) Initializes the memory contents and necessary parameters.
2) Determine $J^{i-1}(k)$ and $\alpha_2^i(k)$
3) Update the memory contents by the new learning rate.
4) Put i=i+1, and go to step 2 if assigned termination criterion is not satisfied.

IV. Simulation Results

The experimental results show the performance of different CMAC models using three examples with various functions. *Changeable-Rate(1)CMAC* and *Changeable-Rate(2)CMAC* are the proposed approaches of this paper respectively. CMAC network with input states of 40, generalization parameter of 10, block size of 4, and initial learning rate of 0.5 is used. Training cycles are 20 times. In addition for better performance, ξ equals to 0.1

Example1.

Target function is: $f(x,y) = (x-y)/\sqrt{x+y}$, where $-1 \le x \le 1$, $-1 \le y \le 9$

Fig. 3 and fig. 4 show comparing error and response time for the offered algorithms and conventional CMAC model.

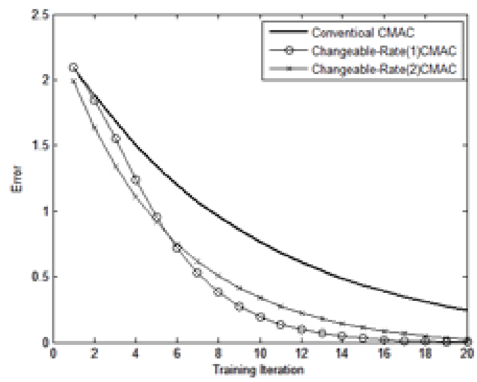

Fig. 3. Error Comparison for different CMAC models.

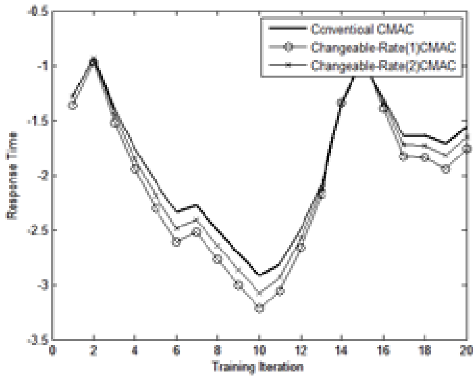

Fig.4. Response Time Comparison for different CMAC models.

Exampl2.

Target function is:

$$f(x,y)=\sin(x)\cos(y)+2\sin(y^2),$$

where $-1<x\le 1, -1<y\le 1$

Error and response time for conventional CMAC and proposed algorithms are shown in fig.5 and fig.6 respectively.

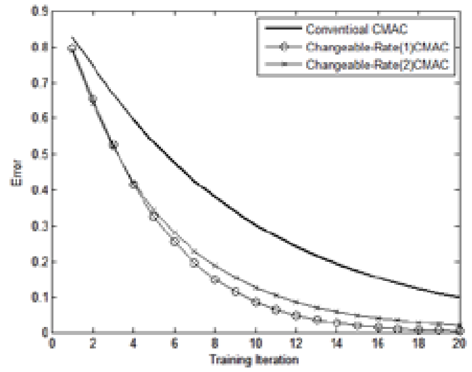

Fig. 5. Error Comparison for different CMAC models.

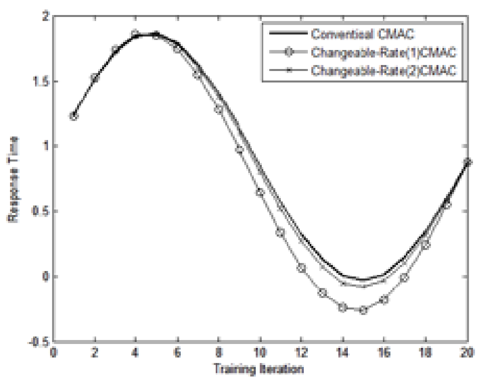

Fig.6. Response Time Comparison for different CMAC models.

Exampl3.

Target function is: $f(x,y)=(3x-y)\sin(2x),$

where $-1<x\le 1, -1<y\le 1$

Comparing error and response time of conventional CMAC and the proposed algorithms are shown at fig. 7 and fig. 8 respectively.

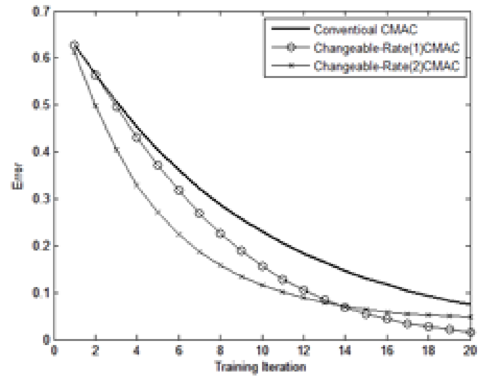

Fig.7. Error Comparison for different CMAC models.

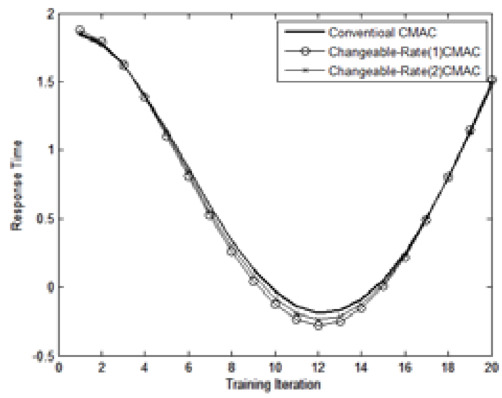

Fig.8. Response Time Comparison for different CMAC models.

As it is indicated from simulation results, the algorithms that are offered at the present research, have very fast convergence and show better performance in comparison to conventional CMAC model

V. Conclusion

Two new methods were offered in the present research with the goal of improving training phase of CMAC model. Whereas fixed learning rate is not suitable in training phase of CMAC model, this paper is offered algorithms whit changeable learning rate at each state of training phase in CMAC model. One of the two applied algorithms presented a novel learning rate based on the gradient reviation of learning rate, and in the second one, learning performance and the number of training iterations were combined with the aim of obtaining changeable learning rate and consequently, a better performance. Test results indicated that these approaches could provide very fast convergence and more accuracy in all training phases of CMAC neural network.

REFERENCES

[1] J.S.Albus, "A New Approach to Manipulator Control: theCerebellar Model Articulation Controller(CMAC)," ASME J.Dynamics Systems, Measurment, Control, pp. 220-227, 19755.

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

[3] Mato Baotic, Ivan Petrovic,Nedjeljko Peric,"Convex Optimizatio in Trainning of CMAC Neural Networks," Automation: Journal for control, Measurement, Electronics, Computing and Communicatios,Vol.42. No. 3-4,2001.

[4] Li Xin, Chen Wei, Chen Mei, Hefei,"Reinforcement Learning Controlbased on TWO-CMAC structure," in Conf. Rec. 2009 IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics,pp. 116-121.

[5] PO-LUN CHANG, YING-KUEI YANG, HORNG-LIN SHIEH, "A novel learning framework of CMAC via Grey-area-time credit apportionment and Grey learning rate," in Conf. Rec. 2008 IEEE Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunmming, pp. 3096-3101.

[6] Luis Weuaga, "Active Training on the CMAC Neuarl Network," Neural Networks,2004. Proceedings. 2004 IEEE International Joint Conf. pp. 855-860. Vol. 2.

[7] Kamran Mohajeri, Ghasem Pishehvar, Mohammad Seifi,"CMAC Neural Networks Structure," in Conf. Rec. 2009 IEEE. Computer Intelligence in Robotics an Autonation (IRA), pp. 39-45.

[8] 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.

[9] S. H. Land, D. A. Handelman, and J. J. Gelfand, "Theory and development of higher-order CMAC neural networks," IEEE Control Syst. Mag., Vol. 12, no. 2, pp. 23-30, Apr. 1992.

[10] Francisco J. Gonzalez-Serrano, Anibal R. Figueiras-Vidal, and Antonio Artes-Rodiguez, "Generalizing CMAC Architecture and Training," IEEE Transactions on Neural Networks, vol. 9, no. 6, 1998.

[11] Ming-Feng Yeh and Kuang-Chiung Chang," A Self-Organizing CMAC Network With Gray Credit Assignment, IEEE Transactions on systems, man, and cybernetics part B:CYBERNETICS, vol. 36, no. 3, June 2006, pp. 623-635.

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

[13] H.M. Lee and C. M. Chen, "Self-Organizing HCMAC Neural-Network classifier," IEEE Transaction on Neural Networks, vol.8, no.6, pp. 1281-1292, 1997.

[14] 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.

[15] Luis Weruaga and Barbara Kieslinger, "Tikhonov Training of the CMAC Neural Networks, vol. 17, no. 3, pp. 613-622, 2006.