Learning Algorithm for Training CMAC by Using Reinforcement Learning and Comparative Discount Rate

Nazal Modhej: Department of Computer Engineering, Soosangerd Branch, Islamic Azad University, Khouzestan-Iran (modhej@siau.ac.ir).
Jamil Neisi: Khoramshahr Branch, Islamic Azad University, Khouzestan-Iran(d_modhej@siau.ac.ir)

Paper Reference Number: 07-04-0415
Name of the Presenter: Nazal Modhej

Abstract:

CMAC is a calculation model based on human cerebellum and is offered as a simple model and may be observed as a lookup table. CMAC due to high efficiency has great application in the field of modeling and control; therefore, requirement for methods to accelerate more exact learning process have made relookupers to use more diverse learning algorithms. In the present article a new algorithm for obtaining more accelerate convergence and therefore less error is offered that operates based on reinforcement learning algorithm. Whereas fixed discount rate in reinforcement learning algorithm is not suitable, a new algorithm based on discount rate of variable is offered in the present article that is applied for training CMAC. Results of simulation show that the recommended algorithm in comparison to contractual CMAC considerably decreases error.

Key Words: Network, CMAC, Training Phase, System Identification, Convergence Speed, Reinforcement Learning, Discount Rate

1.Introduction:

CMAC was first time offered by J.S Albus in the year 1975 [1,2]. CMAC is a calculation model based on human cerebellum and is offered as a simple model and may be imagined as lookup table. This table has high learning speed [3] and its prompt convergence is guaranteed [4]. This network applies from very simple algorithm in order to train network weights and in order to obtain higher convergence from one hand and higher exactness and stability on the other hand, applies from several algorithms for training in this network. Whereas CMAC is interested for high learning speed, several improved learning plans are offered specially for linear control [5]. Credit is determined by using related grey analysis that is regarded as fundamental issue at grey system theory and maybe regarded as specified sequence. In addition it is used from convex optimization for offering more prompt learning rate at CMAC network [6]. In [7] it is used from variable learning rate for obtaining to higher convergence speed from one hand and higher exactness and stability on the other hand [11]. A new training method is offered based on active theory model. Reinforcement learning is a method that operates based on reward and punishment i.e. learning through exchange with environment for obtaining a specified goal [9, 10]. This learning is used at CMAC network [8]. In addition special type of this learning called Q reinforcement learning is studied at CMAC network [12]. Whereas fixed discount rate in reinforcement learning is not suitable, in the present research a learning algorithm based on discount rate of variable is studied. Goal of the present research is decreasing effect of fixed discount rate and obtaining to higher learning speed and high exactness. Therefore, a new discount rate is offered for allocating reward at any state of CMAC through reinforcement learning algorithm. Using such method at training CMAC is regarded as a type of system identification that directs us toward our goal. Structure of the present article is as follows: section 3 deals with reinforcement learning, section 4 and section 5 deal with framework of new algorithm and system identification respectively. Section 6 deals with specific simulation and finally conclusion is offered in section 7.

2. Structure of CMAC:

CMAC network is a set of writing procedures and repeats training process for obtaining to goal of learning. CMAC network has associated memory that applies from following formula for describing non-linear function of $y=f_N(k,w)$: X is input vector, A is associated vector and N is number of input. Concenptual structure of CMAC contractual network is offered at figure 1.

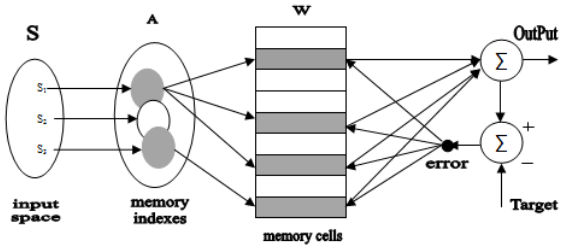

Fig. 1. Basic structure of CMAC model

In this network first of all the domain of learning is determined. Then learning atmosphere is written into several discrete modes that are regarded as input mode for CMAC model. Each of the discrete atmospheres are written from index memory A into real cells of W memory that save information of input mode and are summed for determining level of output. If suitable condition is not satisfied and the output is not equal to expected signal, level of error is calculated and on this basis the weights of network are balanced. This process repeats until network is convergence toward expected signal. In CMAC network all weights are not balanced in the process of training, rather some of them are updated at each state; in which, this number may be higher than total weights of network. These advantages of CMAC results in considerable increase at convergence speed, since total weight vector at each state do not require calculation for being updated. Sample double dimensional CMAC is shown at figure 2. The input variables as S1, S2, are discrete toward unit 21 and are called from 0 to 21. Several elements form a block. In this figure each block is consist of 5 elements; therefore, 21 elements are divided into 5 blocks that are including: 4 complete blocks A,B,C,D and single element block E and layer 2 that is formed from these 5 blocks. A similar design is obtained at S2 line. The aforesaid written blocks are called cube that are in fact real memory cells; in which, they save input information of memory. For example, input mode of (8, 7), activates cubes Bc, Hh, Mm, Rr and Vx in associated memory layer at figure 2. As it was already mentioned each cell is regarded as content of a memory and its output is obtained through sum of cell content.

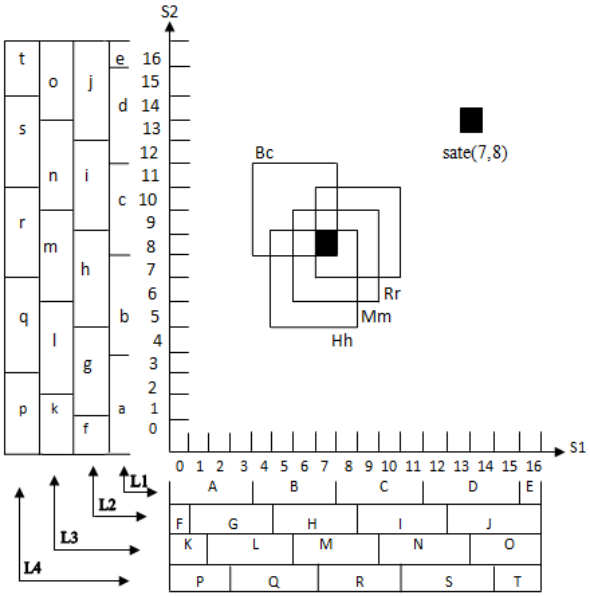

Fig. 2. The memory cells of two dimensional CMAC

Output of this network is calculated whereas follows:

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

K is specified input mode; M is volume of memory that equals to number of cubes and W is weight of vector i.e. saved information at memory. If place of memory j is covered by one of the cubes for input s then aj (s) equals to 1 otherwise it equals to 0. The updating process is very simple and obtains whereas follows:

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

$e=y_d(k)-a^T(s)w$ is error for sample and $y_d(k)$ is expected signal. Error is multiplied at a(s) to cover weight at place of memory by input cubes. α is learning rate so that $0<\alpha<1$,N_e are number of layer for each input.

3.Reinforcement Learning:

Reinforcement learning is learning at different situations with the goal of increasing reward signal; in which, this process was modeled based on Markov Decision Process (MDP) [9, 10]. Within framework of this learning, the factor and environment are exchanged together through separate time steps as $t=0,1, \ldots,\infty$. At each step, the factor is transferred from a new mode i.e.

from $s \epsilon S$ in which S is set of possible modes. Factor results in transferring possibility from $a \epsilon A(s_t)$ in which A(st) is set of accessible operations under St mode. This formula is called π_t and is defined through relation 3:

$$\pi_t(s,a) = Pr\ (a_t = a | s_t = s) \qquad (3)$$

environment transfers factor into new mode of s_{t+1} and reward of $r_{t+1} \epsilon R$ is allocated to it. Goal of this algorithm is maximizing set of rewards that is determined through equation 4.

$$R_t = (r_{t+1}, \gamma r_{t+2}, \gamma^2 r_{t+2}, \dots) = \sum_{k=0}^{\infty} (\gamma^k r_{t+k+1}) \qquad (4)$$

In this formula γ is parameter for discount rate and $0 \leq \gamma \leq 1$. In fact discount rate determines current value of future reward. Real value of reward that is received as k state later equals to γ^{k-1}

4.Description of New Algorithm:

By calculating reward of each mode, at any state of learning CMAC, weights are updated according to relation 5 [8].

$$w_i(k+1) = w_i(k) + \alpha R(k)\, e_i(k)/Ne \qquad (5)$$

R(k) in this relation shows cumulative reward (Puteman 1994) that is determined as follows:

$$R(k) = r(k) + \gamma y(k) - y(k-1) \qquad (6)$$

$y^i(k)$ ، $y^{i-1}(k)$ in this relation are output of network and $r_0(k)$ is either reward or punishment of network based on its error. In case of having many errors, the network receives punishment and therefore fewer error results in allocating reward signal to network. As it was already mentioned the reason of using γ is that effect of reward at initial steps in the process of training is higher than reward at final states; since, response of network convergence is more suitable at initial steps. Therefore, by observing discount rate of γ and reward that is receives at k^{th} iteration is γ^{k-1} worse than reward that is promptly receives. Fixed discount rate is not suitable and if this rate is observed very great, the factor receives very serious future reward and in case of observing this rate very small, the reward of initial state is promptly received for factor. A new method for obtaining to discount rate of variable in order to train weight of CMAC for reinforcement learning is offered in the present article. By using gradian relations, the discount rate of variable is calculated as follows:

$$\Delta\gamma = -\xi\frac{\partial E}{\partial \gamma} = -\frac{\xi\partial E}{\partial y}.\frac{\partial y}{\partial w}.\frac{\partial w}{\partial \gamma}$$

$$= -\xi\big(-e(k)\big).(1).\left(\frac{e(k-1)}{Ne}\right)(y(k)-y(k-1))$$

or

$$\Delta\gamma = \xi e(k)e(k-1)(y(k)-y(k-1))/Ne$$

Therefore parameter of learning rate of variables at each state is calculated as follows:

$$\gamma(k+1) = \gamma(k) + \xi e(k)e(k-1)(y(k)-y(k-1))/Ne \qquad (7)$$

by calculating discount rate according to the above equation, reward of each state R (k) at each state of learning phase is determined as follows:

$$R^{*}(k) = r(k) + \gamma(k)\big(y(k)-y(k-1)\big) \qquad (8)$$

The only difference of abovementioned equation with equation 6 in relation to discount rate is that in equation 6, this parameter is fixed for all steps, meanwhile in equation 8 this parameter is variable at each state. It is obvious that by using this new relation for reward, weights of CMAC network at each state of learning phase is updated as follows:

$$w_i(k+1) = w_i(k) + \alpha R^{*}(k)e_i(k)/Ne$$

Using such method for training CMAC results in increasing speed and exactness for all training cycles. The learning procedure is summarized as figure 3.

1) initialize the memory contents and necessary parameters.
2) determine r(k), γ(k) and R*(k).
3) update the memory contents with the new parameters.
4) i=i+1 and go to step 2 if assigned termination criterion is not Satisfied.

figure3: Learning Procedure with the novel learning framework

The new learning algorithm is a type of system identification that is explained in the next section.

5.System Identification:

Models of real systems have special importance at virtual systems. Models are useful for system analysis or obtaining better understand in relation to system. Modeling and identifying non-linear dynamic systems is among most important problems of engineering; since, non-linear processing due to lack of sharing, have several unique properties. The most important of each non-linear system is ability of describing great class of different structural systems. For more simplicity only one process is observed for determining output. Quality of model is calculated by using error function between processing real output and mode output and this error is used for balancing parameters of model. Input of process and model are common: $v=[v_1,v_2,\ldots,v_p]$ output of both sections is compared for calculating error signal e that is used to balance model. It is to be noted that output of process may influence by n noise according to figure 4.

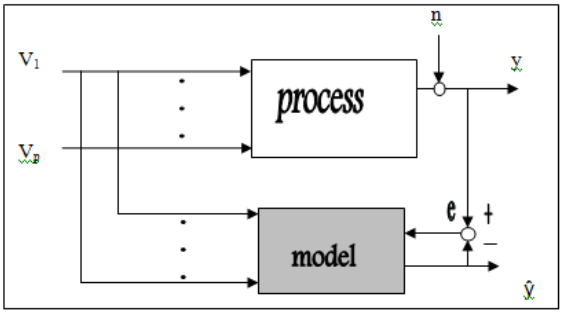

Fig4.non-linear block identifier

Figure 5 shows process of system identification. Complexity and requirement of system to previous knowledge is decreased from step 1 to 7; meanwhile, using knowledge experiences is increased. Description for detail of these steps is offered in [14].

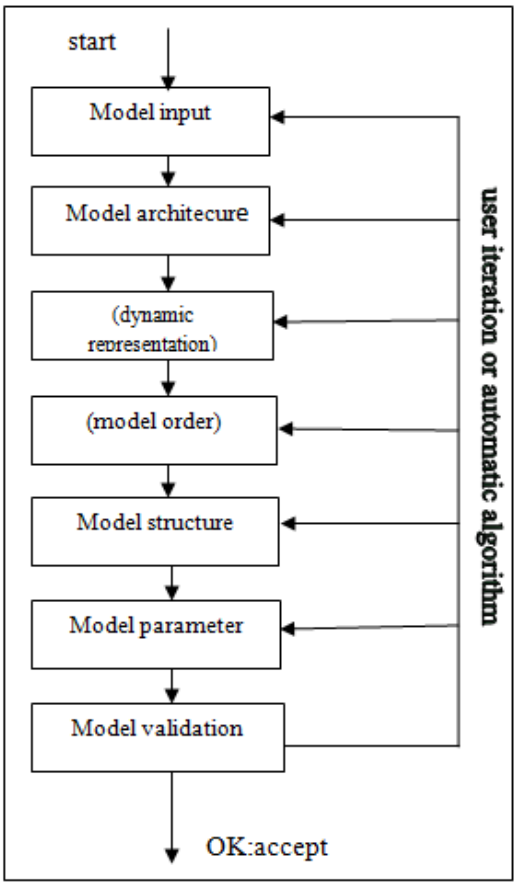

Fig5.system identifier cycle

In order to show efficiency of this new algorithm, it is simulated and its results are compared with contractual CMAC network. For simulation it is used from Matlab 7.6.0.324(R2008a) software in this research. In order to test new learning algorithm, 3 types of functions were studied and results of simulation with respect to error at network is completely described in next section. Extension parameter is observed as 10 and size of cube is observed as 4. Learning rate equals to 0.5 and 20 couples of input-output data were used to test this network and training courses were repeated for 20 times.

Sample of First Function:

$$f(x_1,x_2)=\sin(x_1)\cos(x_1)+2\sin(x\hat{}2)$$

$$-1 < x_1 \leq 1, -1 < x_2 \leq 1$$

Figures 6 and 7 show comparing error, time of responding to new algorithm and CMAC algorithm; in which, in this simulation the process of system identification was better in new algorithm.

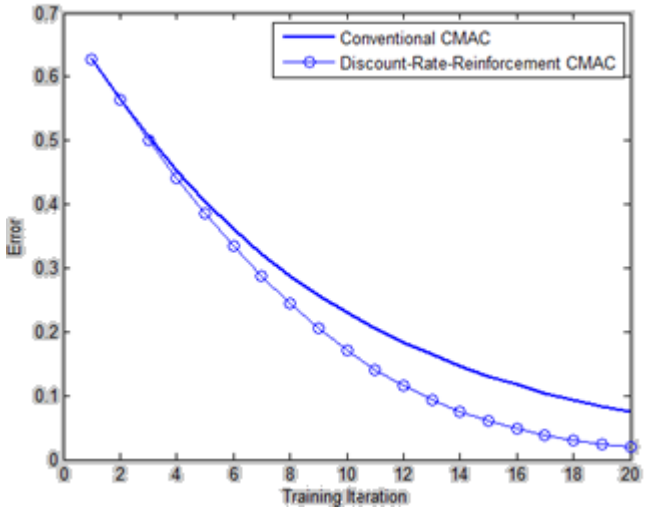

Fig. 6 Error Comparison for different CMAC models.

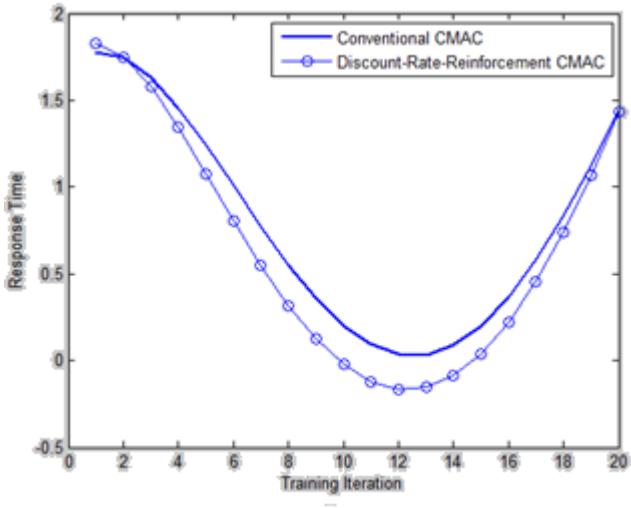

Fig. 7. Response Time Comparison for different CMAC models.

Sample of Second Function:

$$f(x,y) = (x-y)/\sqrt{(x+y)}$$

$$-1 \le x \le 1, 1 \le y \le 9$$

Figures 8 and 9 show error and time of response to new learning algorithm toward CMAC

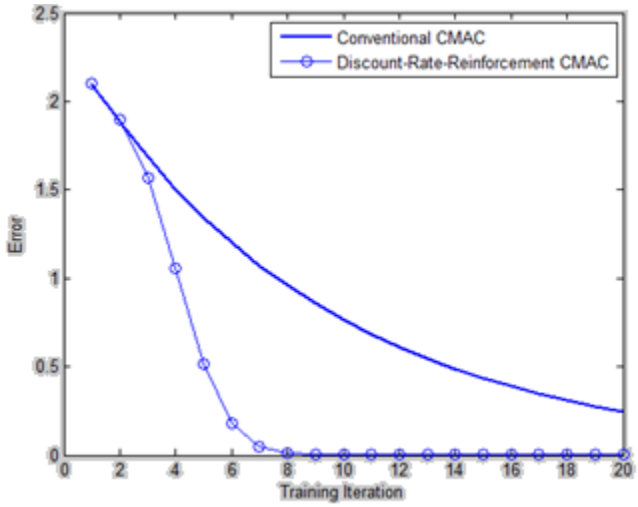

Fig. 8 Error Comparison for different CMAC models.

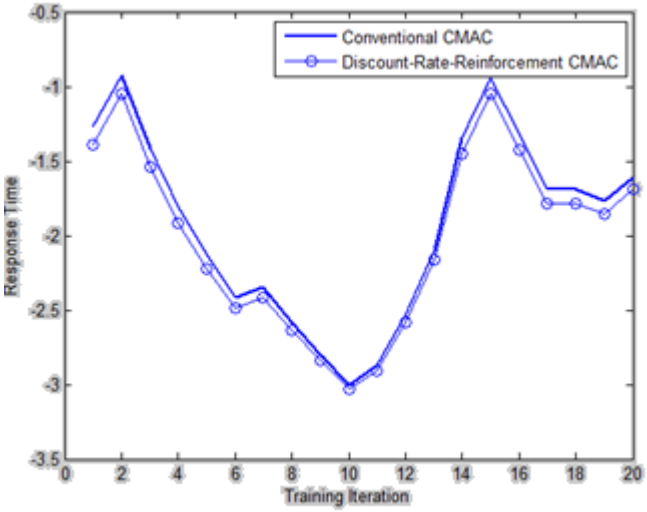

Fig. 9. Response Time Comparison for different CMAC models.

As it is obvious in these 2 simulations, results of all sample functions show that network error in case of using new learning algorithm is considerably decreased in comparison to CMAC learning algorithm.

6) Conclusion:

In this research it was used from a method to obtain higher speed at CMAC network. This algorithm applies from mixture of reinforcement learning algorithm and discount rate of variable. Reinforcement learning operates based on reward and punishment system at each state. If network is close to expected signal it may receive a positive number as reward, otherwise it may receive a negative number as punishment. Factor called discount rate is observed for reward and punishment system; in which, effect of reward on initial states in comparison to final states of training is more considerable. In this research it was used from concept of effect rate of variable for reinforcement learning and results of simulation showed that reinforcement learning algorithm by having discount rate in comparison to common learning method called CMAC, considerably decreases level of error.

REFERENCES

[1]-Albus J.S., "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," Trans. of the ASME: Journal of Dynamic Systems, Measurement, and Control, pp.220-227,1975.

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

[3] Glanz F.H. & Miller W.T., "Shape Recognition Using a CMAC base learning system," SPIE Conference on Robotics and Intelligent Systems, Vol. 848*, pp.* $294-298$*,* 1987*.*

[4] Parks P.C., Militzer J., "Convergence Properties of Associative Memory Storage for Learning Control System," Automat, Remote Contr. Vol. 50*, pp.* $254-286$*,* 1989*.*

[5] 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.

[6]Mato Baotic, Ivan Petrovic,Nedjeljko Peric,"Convex Optimizatio in Trainning of CMAC Neural Networks", AUTOMATIKA 42(2001) 3-4, 151-157.

[7]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", Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunmming, 2008.

[8] Li Xin, Chen Wei, Chen Mei, Hefei,"Reinforcement Learning Controlbased on TWO-CMAC structure",International Conference on Intelligent Human-Machine Systems and Cybernetics,2009.

[9]PatryK A. Laurent, "The emergence of saliency and novelty responses from reinforcement Learning principles", Neural Networks 21(2008) 1493-1499

[10]K.H. Quai, Chai Quek, "Maximum reward reinforcement learning: A non-cumulative reward criterion", Expert Systems with applications 31 (2006) 351-359

[11] Luis Weuaga, "Active Training on the CMAC Neuarl Network", 0-7803-8359-1/04/$20.00, 2004IEEE.

[12] Hisashi Handa, "Detecting of Critical Situations by CMAC+Q-Learning for PacMan Agent", Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan, March 26-29, 2009.

[13] 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, NOVEMBER 1998.

[14] Oliver Nelles, 'Nonlinear System Identification:From Classical Approaches To Neural Networka and Fuzzy Models'.