Applied Soft Computing 11 (2011) 5400–5410

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Grey adaptive growing CMAC network

Ming-Feng Yeh*, Min-Shyang Leu

Department of Electrical Engineering, Lunghwa University of Science and Technology, 33306 Taoyuan, Taiwan

ARTICLE INFO

Article history:
Received 20 July 2009
Received in revised form 31 August 2010
Accepted 2 May 2011
Available online 20 May 2011

Keywords:
Cerebellar model articulation controller
Grey relational analysis
Tree structure
Adaptive quantization

ABSTRACT

This study attempts to develop a grey adaptive growing cerebellar model articulation controller (CMAC) network, which is constructed by connecting several 1D Albus' CMACs as a two-level tree structure. Even though the target function is unknown in advance, grey relational analysis still can analyze the learning performance between the network outputs and the target values. According to the result of grey relational analysis, the proposed adaptive growing mechanism could determine whether a specific region covered by a state or a CMAC needs to be repartitioned or not. By this way, not only the number of 1D CMACs but also the number of states could be gradually increased during the learning process. And then the purpose of self-organizing input space can be attained. In addition, the linear interpolation scheme is applied to calculate the network output and for simultaneously improving the learning performance and the generalization ability. Simulation results show that the proposed network not only has the adaptive quantization ability, but also can achieve a better learning accuracy and a good generalization ability with less memory requirement.

1. Introduction

Cerebellar model articulation controller (CMAC) network, proposed by Albus [1,2], is a kind of supervised neural network inspired by the human cerebellum. Unlike multilayer perceptron networks, CMAC network learns input–output mappings based on the premise that similar inputs should produce similar outputs. Hence it stores information locally in a way that the learned information can be easily recalled but use less memories. Owing to its fast learning speed, good generalization ability and ease of implementation by hardware, the CMAC network has been successfully applied in the control problem [3–6].

The basis function of CMAC network can be categorized into two main types. One is the constant basis function [1]. The other is the non-constant differentiable basis function such as Gaussian function or B-Spline function [7]. CMAC network with constant basis function is also termed Albus' or conventional CMAC, while with differentiable basis function is simply termed differentiable CMAC hereafter. Not only Albus' CMAC but also differentiable CMAC has two major limitations: enormous memory requirement for solving high-dimensional problem and difficult in selecting the memory structure parameters [8]. To reduce enormous memory requirement, Lin and Li [9] presented a tree-type CMAC structure composed of a set of submodules constructed by several two-dimensional (2D) Albus' CMACs. Each 2D CMAC in a submodule has a subset of the system inputs as its input variables. The network output is the sum of the outputs from a set of submodules. Lee et al. [8] proposed a supervised hierarchical CMAC (HCMAC) network composed of 2D differentiable CMACs in the structure of full binary tree topology to reduce the memory requirement. Rather than using 2D CMACs in the network structure, Hung and Jan [10] developed a macro structure CMAC (MS_CMAC) by connecting several 1D conventional CMACs as a tree structure to reduce the computational complexity in multidimensional CMAC. Further studies of MS_CMAC network can be found in [11,12]. On the other hand, the input space quantization of a CMAC network will affect its memory structure parameters. Since the clustering methods could reduce the memory requirement for high dimensional problem, several researcheres proposed the feasibility of applying clustering techniques to obtain adaptive resolution for input space quantization [8,13–16].

The literatures stated in above can solve at least one of two major limitations of CMAC network. In general, those CMACs with the ability to self-construct or self-organize input space are the ones with differentiable basis function [8,13–16]. In other words, most of Albus' CMACs are lack of such a ability. This gives rise to the motivation to develop a self-organizing approach for Albus' CMAC. However, since the output of CMAC network is always constant, the derivative information of input and output variables cannot be preserved. As a result, the gradient-descent learning rule cannot be applied to update the memory contents of Albus' CMAC and to self-organize the input space. Based on grey relational analysis [17], this study attempts to propose a new kind of 2D CMAC network, termed the grey adaptive growing CMAC network (GAG-CMAC), to

* Corresponding author. Tel.: +886 2 82093211x5501; fax: +886 2 82099728.
E-mail address: mfyeh@mail.lhu.edu.tw (M.-F. Yeh).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.05.019

simultaneously achieve the purpose of self-organizing input space and overcome those two major limitations. Generally speaking, the differentiable CMAC has a better learning accuracy and generalization ability than Albus' CMAC [7]. The memory requirement of the former, however, is larger than that of the latter. Herein the memory requirement is defined as the number of memory elements to store all adjustable parameters such as the memory contents of CMAC network and the mean and variance of each Gaussian basis function. In order to construct the proposed network with less memory requirement, it is on the basis of Albus' CMAC. Hereafter, for abbreviation, this study uses CMAC instead of Albus' CMAC network unless it is necessary to be specified. Besides, this study also applies the linear interpolation scheme to the recall process of CMAC network so as to reduce both the learning error and the generalization error caused by quantization.

The proposed GAG-CMAC network is constructed by several 1D CMACs as MS_CMAC network, but its learning process (also called the adaptive growing process) contains two adaptive growing stages. The first one is the state growth stage which is used to adaptively quantize the domain of one direction, say *x*-axis. Different 1D CMACs may have different quantization results. The second one is the CMAC growth stage which is used to adaptively quantize the domain of another direction, say *y*-axis. In addition, this study also applies grey relational analysis [17] into such two growing mechanisms. Even though the target function is unknown in advance, grey relational analysis still can analyze the learning performance between the network outputs and the target values. Then the relational grade corresponding to a specific region covered by a state or a CMAC is employed to evaluate whether that region needs to be repartitioned or not. By this way, not only the number of 1D CMACs but also the number of states could be gradually increased during the learning process. With the help of these two adaptive growing stages, the purpose of self-organizing input space could be attained. Note that this study only focuses on the 2D problem because the multidimensional problems can be solved by use of the hierarchical structure as HCMAC network [8].

The remainder of this paper is organized as follows. Section 2 briefly represents some background material such as grey relational analysis and CMAC networks. The proposed GAG-CMAC network is given in Section 3. Section 4 shows the simulation results of two examples. Finally, Section 5 contains some conclusions of this study.

2. Grey relational analysis and CMAC networks

2.1. Grey relational analysis

Grey relational analysis is a similarity measure for finite sequences with incomplete information [17]. For a given reference vector and a given set of comparative vectors, grey relational analysis can be used to determine the relational grade between the reference and each element in the given set. Then the most similar vector in the comparative set to the reference can be found by further analyzing the resultant relational grades. In other words, grey relational analysis can be viewed as a measure of similarity for vectors of finite dimensions.

Denote the reference sequence by $\mathbf{v}^{(r)} = (v_1^{(r)}, v_2^{(r)}, \ldots, v_n^{(r)})$ and the ith comparative sequence by $\mathbf{v}^{(i)} = (v_1^{(i)}, v_2^{(i)}, \ldots, v_n^{(i)})$, $i = 1, 2, \ldots, m$. Then the grey relational coefficient between $\mathbf{v}^{(r)}$ and $\mathbf{v}^{(i)}$ at the jth component, $j = 1, 2, \ldots, n$, is defined as follows:

$$r(v_j^{(r)}, v_j^{(i)}) = \frac{\Delta_{\min} + \xi\Delta_{\max}}{\Delta_{ij} + \xi\Delta_{\max}}, \tag{1}$$

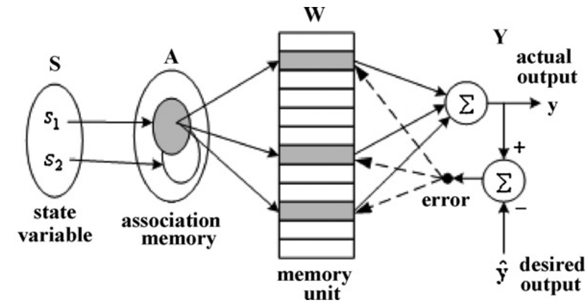

Fig. 1. Architecture of CMAC network.

where $\Delta_{ij} = \left| v_j^{(r)} - v_j^{(i)} \right|$, $\Delta_{\max} = \max_i \max_j \Delta_{ij}$, $\Delta_{\min} = \min_i \min_j \Delta_{ij}$, and $\xi \in (0,1]$ is the distinguishing coefficient controlling the resolution between $\Delta_{\max}$ and $\Delta_{\min}$. The corresponding grey relational grade is

$$g(\mathbf{v}^{(r)}, \mathbf{v}^{(i)}) = \sum_{j=1}^{n} w_j r(v_j^{(r)}, v_j^{(i)}), \tag{2}$$

where $w_j \geq 0$ is the weighting factor satisfying $\sum_{j=1}^{n} w_j = 1$. Normally, we select it as $w_j = 1/n$ for all j. The best comparative sequence to the reference is the one with the largest relational grade. Note that we can always guarantee that $0 < r(v_j^{(r)}, v_j^{(i)}) \leq 1$ and $0 < g(\mathbf{v}^{(r)}, \mathbf{v}^{(i)}) \leq 1$.

2.2. Albus' CMAC network

In a CMAC network, each state variable is quantized and the problem space is divided into discrete states [1,2]. A vector of quantized input values specifies a discrete state and is used to generate addresses for retrieving information from memory elements for this state. The basic structure of CMAC is depicted in Fig. 1. In the figure, the association memory **A** is obtained from the input space **S**, the associated data stored in the memory cell **W** are yielded in accordance with each input state. The CMAC sums the mapped data up as its output and feeds the error between the actual and desired outputs back to the memory cell equally. Assume that the input space **S** is quantized into N_s states and every state utilizes N_e memory units to store the corresponding memory contents. Furthermore, assume that the number of memory units is N_{mem}. Owing to Albus's CMAC network without using the Gaussian function as its basis function, a memory unit consists of one and only one memory element which is to store the memory content of the network. That is to say, N_{mem} is equal to the number of memory elements. The stored data y_k (the actual output of the CMAC) for the state s_k is the sum of stored contents of all addressed memory units and can be expressed as

$$y_k = \mathbf{a}_k \mathbf{w} = \sum_{l=1}^{N_{mem}} a_{k,l} w_l, \tag{3}$$

where w_l, $l = 1, 2, \ldots, N_{mem}$, is the memory content of the lth memory unit, $\mathbf{w} = [w_1, w_2, \ldots, w_{N_{mem}}]^T$, $a_{k,l}$ is the association index indicating whether the lth memory unit is addressed by the state s_k or not, and $\mathbf{a}_k = [a_{k,1}, a_{k,2}, \ldots, a_{k,N_{mem}}]$. Since each state addresses exactly N_e memory units, only those addressed $a_{k,l}$'s are 1, and the others are 0.

The CMAC uses a supervised learning method to adjust the memory contents during each learning cycle. Its updating rule can be described as

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \frac{\eta}{N_e} \mathbf{a}_k^T [\hat{y}_k - \mathbf{a}_k \mathbf{w}(t)], \quad t = 1, 2, 3, \ldots \tag{4}$$