CLUSTERING ECG COMPLEXES USING SELF-ORGANIZING CMAC

CHENG WEN¹, CHIH-HUNG HUANG², MING-FENG YEH¹, KUANG-CHIUNG CHANG¹

¹Department of Electrical Engineering,

²Department of Information Management, Lunghwa University of Science and Technology,
Taoyuan, Taiwan

ABSTRACT

In the study, we apply a self-organizing cerebellar model articulation controller (SOCMAC) network to cluster QRS complexes. Forty-four out of 48 records from the MIT-BIH Arrhythmia Database are used to test the SOCMAC network. The SOCMAC network is an unsupervised and nonlinear learning method. For each record, under fixed initial memory contents, we search optimal parameters in the updating rule of the network for achieving the best classification accuracy. The entire classification accuracy, with the optimal parameters, is 99.41%, which is higher than existing results. The testing data are divided into two groups: noisy and clean. The classification accuracy of each group differs not much. Therefore, we can conclude that the SOCMAC shows a certain degree of robustness against noise.

Biomed Eng Appl Basis Comm, 2006(December); 18: 312-318.

Keywords: ECG classifier; QRS complex; SOCMAC; gray relational analysis; gray credit assignment

1. INTRODUCTION

The electrocardiogram (ECG) is the record of variation of bioelectric potential with respect to time as the human heart beats. Due to the ease of recording ECG in a noninvasive manner, ECG plays an important role in patient monitoring and diagnosis today. However, a large amount of data may be obtained in the case of monitoring ECG signals of a patient continuously during 24 hours. In this case, typically, there could be more than 100,000 heartbeats per channel for analysis. Since visual analysis of such

amount of ECG data is time-consuming, many computer-based methods for automated ECG analysis have been studied for a long time. ECG pattern recognition is a difficult problem even with the aid of a computer because ECG signals differ significantly even for the same type and for the same patient.

Several approaches to automatic classification of heartbeats have been proposed in the literature. They include statistical [1] and syntactic [2] approaches. Artificial neural networks were also employed to exploit their natural ability in pattern recognition tasks for successful classification of ECG beats [3-9]. The MLP is one of the most popular artificial neural networks used as the ECG classifier. In [3], the principal characteristics of the ECG signal were extracted by using the principal component analysis technique and then were applied to an MLP for classification. Minami et al. [4] used Fourier transform to extract features of QRS complexes and used an

Received: July 26, 2006; Accepted: Sep. 26, 2006

Correspondence: Kuang-Chiung Chang, Professor
Department of Electrical Engineering, Lunghwa
University of Science and Technology, Taoyuan, Taiwan
33306

E-mail: kcchang@mail.lhu.edu.tw

MLP with 5 input and 2 output cells to classify the features. Self-organizing maps have also been applied for classifying ECG beats [5-6]. In [5], self-organizing maps were used for designing a customized beat classifier to a specific patient and a global beat classifier, and then the two classifiers were combined by the mixture-of-experts principle. A 5×5 self-organizing map was employed for clustering ECG beats into 25 clusters in [6], in which the QRS complexes were represented by the orthogonal Hermite functions. In [7], a multichannel ART2 neural network was proposed for the classification of two-channel ECG signals. Recent years have seen a growing number of combined neural network approaches to ECG classification. The neuro-fuzzy network has been used to design the ECG classifier [8]. The neuro-fuzzy network is composed of two subnetworks connected in cascade: the fuzzy self-organizing layer performing the preclassification task and the following MLP working as the final classifier. In [9], the MLPs were used at the first level and the second level for the implementation of the combined neural network.

In this paper, the proposed ECG beat classification method is based on the self-organizing cerebellar model articulation controller (SOCMAC) network [10], which was proposed by our research group. The SOCMAC combines the structure of the cerebellar model articulation controller (CMAC) network into the Kohonen layer of the self-organizing map (SOM) so that it can perform the function of an SOM and can distribute the learning error into the memory contents of all addressed hypercubes as a CMAC. Therefore, this new type of neural network has the features of both the SOM and the CMAC. Furthermore, a credit assignment approach, which uses the inverse of gray relational grades of the memory contents of all addressed hypercubes as the credibility of those hypercubes, has also been introduced in the updating rule to speed up the overall learning process.

The remainder of this paper is organized as follows. Section II presents some background material. The methodology for ECG beat classification and simulation results are shown in Section III and Section IV, respectively. Finally, Section V contains some conclusions.

2. PRELIMINARIES

In this section, we briefly introduce the network used for designing the proposed ECG classifier. Because the structure of SOCMAC is the same as the CMAC, we introduce the CMAC first and then the SOCMAC.

2.1 CMAC

Fig. 1. Block division of CMAC network for a two-variable case.

In a CMAC network, each state variable is quantized and the problem space is divided into discrete states [11-12]. 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. Fig. 1 illustrates a simple block division of the CMAC scheme for a two-variable case. This simple example has two state variables with each quantized into three discrete regions, called blocks, in each layer. Areas, such as *Bb*, *Ee*, and *Ih*, formed by quantized regions are called hypercubes. Only the blocks on the same layer can be combined to form a hypercube. In Fig. 1, it is clear that each state is covered by three different hypercubes, one from each layer. For example, three hypercubes, *Bb*, *Ee*, and *Ih*, are addressed by the state located on (5,4). The CMAC associates each hypercube to a physical memory address. Information for a discrete state is distributively stored in memory elements associated with the hypercubes being addressed. Therefore, information for that state is stored in the memory elements associated with these three hypercubes. In Fig. 1, each state has its own coordinates. For example, *i* and *j* are the coordinates of the state $s(i, j)$. Hereafter, for abbreviation, we use the notation s_k instead of $s(i, j)$ unless it is necessary to be specified.

Assume that information for a quantized state is distributively stored in N_e memory elements and that N_h represents the entire memory size ($N_h > N_e$). 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 hypercubes and can be expressed as

$$y_k = \sum_{h=1}^{N_h} a_{k,h} v_h, \quad (1)$$

where v_h , $h=1,2,\dots, N_h$, is the memory content of the h th hypercube and $a_{k,h}$ is the index indicating whether the h th hypercube is addressed by the state s_k . Since each state addresses exactly N_e hypercubes, only those addressed $a_{k,h}$'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

$$v_h(t+1) = v_h(t) + \frac{\eta}{N_e} a_{k,h} [\hat{y}_k - \sum_{h=1}^{N_h} a_{k,h} v_h(t)], \quad (2)$$

$$t = 1, 2, \dots,$$

where $v_h(t+1)$ is the stored value of the h th hypercube at time $t+1$, $v_h(t)$ is the one at previous time t , η is the learning rate, $\hat{y}_k$ is the desired value for the state s_k , and $\hat{y}_k - \sum_{h=1}^{N_h} a_{k,h} v_h(t)$ is the error for the state s_k . Note that only the addressed hypercubes are updated.

2.2 SOCMAC

The input space of the CMAC can be viewed as a topological structure similar to the output layer of the SOM. If the output of a state is in an n -dimensional vector form, then each state in such a CMAC input space can play the role of an output neuron of the SOM, and the output vector of that state is as the corresponding connection weight of the SOM. To make the SOCMAC perform the behavior of the SOM, every memory content has to be an n -dimensional vector, not a scalar. By this way, the output of a specific state is also an n -dimensional vector that is consistent with the connection weight of the SOM. Denote the memory content of the h th hypercube of the SOCMAC by $\mathbf{w}_h = (w_{h1}, w_{h2}, \dots, w_{hn})$, where $h=1,2,\dots,N_h$. Then, the output of the state s_k , denoted by $\mathbf{y}_k$, is

$$\mathbf{y}_k = \sum_{h=1}^{N_h} c_{k,h} \mathbf{w}_h, \quad (3)$$

where $c_{k,h}$ is the index indicating whether the h th hypercube is addressed by the state s_k . As stated in the above, only those addressed $c_{k,h}$'s are 1, and the others are 0. Suppose that s_1 - and s_2 - axes of SOCMAC input space are quantized into m_x and m_y elements, respectively, where an element represents a small interval in each axis. Then there are $m_x \times m_y$ states in the input space of the SOCMAC, and the output of each state can be obtained by using (3). When a training data, say $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is presented to the SOCMAC, the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}_k$, denoted by $d(\mathbf{x}, \mathbf{y}_k)$, is

$$d(\mathbf{x}, \mathbf{y}_k) = \|\mathbf{x} - \mathbf{y}_k\|, \quad (4)$$

where $k \in K_s = \{1, 2, \dots, m_x \times m_y\}$. Then the winning state is selected as the one whose output has the largest similarity measurement to the input $\mathbf{x}$. Thus, the winning state, s_{k^*} , satisfies

$$d(\mathbf{x}, \mathbf{y}_{k^*}) = \|\mathbf{x} - \mathbf{y}_{k^*}\| = \min_{k \in K_s} \|\mathbf{x} - \mathbf{y}_k\|. \quad (5)$$

Similar to (2), the updating rule for the memory contents of the winning state is

$$\mathbf{w}_h(t+1) = \mathbf{w}_h(t) + \frac{\eta(t)}{N} c_{k^*,h} [\mathbf{x} - \sum_{h=1}^{N_h} c_{k^*,h} \mathbf{w}_h(t)], \quad (6)$$

$$t = 1, 2, \dots,$$

where $\mathbf{w}_h(t+1)$ is the stored vector of the h th hypercube at time $t+1$, $\mathbf{w}_h(t)$ is the one at previous time t , $\eta(t)$ is the learning rate at time t , $c_{k^*,h}$ is the index indicating whether the h th hypercube is addressed by the winning state, and $\mathbf{x} - \sum_{h=1}^{N_h} c_{k^*,h} \mathbf{w}_h(t)$ represents the learning error for the winning state. The updating rule has two features: (1) it is in an unsupervised learning manner, and (2) the learning error is equally distributed into the associated memory elements of the winning state. Hence, the updating rule in (6) possesses both of characteristics of the SOM and the CMAC.

2.3. SOCMAC with Gray Credit Assignment

In the updating rule (6), the learning error is equally distributed into all addressed hypercubes ignoring the credibility of those hypercubes. In other words, equation (6) cannot determine which memory content is more responsible for the current error. If each input data is viewed as the reference while the memory contents of all addressed hypercubes are regarded as the comparative ones in gray relational analysis [13-14], then the corresponding relational grades can be obtained by using

$$g(\mathbf{x}, \mathbf{w}_i) = \sum_{j=1}^n [\alpha_j \cdot r(x_j, w_{ij})], \quad (7)$$

where $\alpha_j \geq 0$ is the weighting factor satisfying $\sum_{j=1}^n \alpha_j = 1$, and the gray relational coefficient $r(x_j, w_{ij})$ is defined as

$$r(x_j, w_{ij}) = \frac{\Delta_{\min} + \xi \cdot \Delta_{\max}}{\Delta_{ij} + \xi \cdot \Delta_{\max}}, \quad j \in J = \{1, 2, \dots, n\}, \quad (8)$$

$$\text{where } \Delta_{ij} = |x_j - w_{ij}|, \quad \Delta_{\max} = \max_{i \in I} \max_{j \in J} \Delta_{ij},$$

$\Delta_{\min} = \min_{i \in I} \min_{j \in J} \Delta_{ij}$, and $\xi \in (0, 1]$ is the distinguishing coefficient controlling the resolution of $r(x_j, w_{ij})$. The assumption made here is that the larger the relational grade, the more accurate the memory content and, consequently, the less the modification needed to the

memory content. Hence, the relational grades between the input data and the memory contents of all addressed hypercubes can be viewed as the credibility of those hypercubes. This credit assignment approach is termed the gray credit assignment.

Under the above assumption, (6) can be rewritten as

$$\mathbf{w}_h(t+1) = \mathbf{w}_h(t) + \eta(t)c_{k^*,h}f_h(t)[\mathbf{x} - \sum_{h=1}^{N_e} c_{k^*,h}\mathbf{w}_h(t)],$$

$$t = 1, 2, \dots, \quad (9)$$

where $f_h(t)$ is the credibility of the h th hypercube at time t , which is calculated by

$$f_h(t) = \frac{\hat{g}_h(t)^{-1}}{\sum_{h^*=1}^{N_e} g(\mathbf{x}, \mathbf{w}_{h^*}(t))^{-1}}, \quad (10)$$

where $\mathbf{w}_{h^*}(t)$, $h^*=1,2,\dots,N_e$, denotes the memory content of the hypercube addressed by the winning state s_{k^*} at time t , $g(\mathbf{x}, \mathbf{w}_{h^*}(t))$ is the gray relational grade between $\mathbf{x}$ and $\mathbf{w}_{h^*}(t)$ obtained by using (7), and $\hat{g}_h(t) = 0$ if $c_{k^*,h} = 0$; $\hat{g}_h(t) = g(\mathbf{x}, \mathbf{w}_{h^*}(t))$, otherwise. Note that since each state addresses exactly N_e hypercubes, the computation of gray relational coefficient in (8) only takes N_e comparative $\mathbf{w}_{h^*}(t)$'s into consideration. The credibility $f_h(t)$ in (10) is well defined since the gray relational grade satisfies $0 < g(\mathbf{x}, \mathbf{w}_{h^*}(t)) \leq 1$.

The idea of the proposed updating algorithm is that the error correcting is proportional to the inverse of the gray relational grade of the addressed hypercubes. With this modification, the learning errors can be appropriately distributed into the addressed hypercubes according to the credibility of hypercubes. With the gray credit assignment, the overall learning procedure for the SOCMAC is described as follows:

Overall Learning Procedure

- (1) Initialize the memory contents and necessary parameters.
- (2) Present an input training vector, $\mathbf{x}(t)$, to the SOCMAC.
- (3) Calculate all state outputs by using (3).
- (4) Determine the winning state by using (5).
- (5) Update the memory contents by using (9).
- (6) If every input training data is presented to the SOCMAC, then go to step 7; otherwise, go to step 2.
- (7) Reduce the learning rate by a small and fixed amount.
- (8) Go to step 2 if the selected termination criterion is not satisfied.

3. ECG DATABASE

The MIT/BIH arrhythmia database [15], sampled

at 360 Hz with 11 bits resolution, is used in this study because it contains a wide variety of QRS complex morphologies and different types of noise and artifacts. This database consists of 48 records. Each recording has a duration of 30 minutes and includes two channels labeled by the cardiologists. Excluding records containing the paced beats (102, 104, 107, and 217), we use 44 records of the MIT/BIH database. Raw data from channel 1 (with some beats missing) are used in this experiment. Since we use the QRS part of the signal to identify the type of ECG beat, the "x" beats only containing the P wave are regarded as unclassifiable beats.

The AAMI recommended practice [16] has provided a protocol for a reproducible test with realistic clinical requirements, emphasizing record-by-record presentation of results that estimate an ECG classifier's ability to detect events of clinical significance. The beats in the MIT/BIH database are of several different beat types. In this study, the AAMI recommended practice is used to combine the MIT/BIH beat types into 4 categories [5]. The detail is shown in Table I. In which, category V contains ventricular premature beats, ventricular flutter beats, and ventricular escape beats, category F contains fusion of ventricular and normal beats, category Q contains blocked APC and unclassifiable beats category, and N contains the other types of beats.

Table I. Labels used for representing beat types in this study

MIT Label	Beat Type	Our Label
N	Normal beat	N
L	Left bundle branch block	N
R	Right bundle branch block	N
A	Atrial premature beat	N
a	Aberrated arial premature beat	N
J	Nodal (junctional) premature beat	N
S	Supraventricular premature beat	N
V	Ventricular premature beats	V
F	Fusion of ventricular and normal beat	F
!	Ventricular flutter beat	V
e	Atrial escape beat	N
j	Nodal (junctional) escape beat	N
E	Ventricular escape beat	V
x	Non-conducted P-wave (blocked APC)	Q
Q	Unclassified beat	Q

This study does not investigate the problem of QRS detection, instead the position of annotation labels provided with the MIT/BIH Database is used to identify the R wave (highest point of the ECG beat). The size of QRS is 73 samples (about 200 ms) with the R wave at the center. Only the morphology of QRS is used in this study. To test the robustness of the proposed method to high frequency noise and power line interference, we do not use any data-preprocessing procedure such as filtering. However, the mean value is subtracted from each beat to remove the baseline wander [5].

4. EXPERIMENTAL RESULTS

A SOCMAC network was used as an ECG classifier to cluster the beats in each one of the selected 44 records into 4 categories. Because there are 14 MIT beat types in Table I and QRS morphologies differ significantly in nature even for the same beat type and for the same patient, we use 25 clusters for collecting all possible types of QRS morphology. As a consequence, the number of states in the SOCMAC input space is 25, i.e., $m_x=m_y=5$. The possible generalization parameter N_e is 2, 3, or 4 for implementing this network. Different generalization parameters result in different SOCMAC structures. With $N_e=2$ each layer of the block contains 3 blocks and the entire memory size is 18. For the case that $N_e=3$, there are two layers containing 2 blocks and one layer containing 3 blocks. Hence the corresponding memory size is 17. Since each layer contains only 2 blocks for $N_e=4$ the entire memory size is 16. In this experiment, we select $N_e=2$.

Clustering ECG complexes was done record by record. From the updating rule in (9), we see that the unsupervised learning of the SOCMAC is nonlinear and depends on initial weights, initial learning rate, and distinguishing coefficient. For each record, the initial weights $\mathbf{w}_h(1)$ are uniformly selected from that record according to the distribution of beat types. Under the initial weights, for searching the best result of each record in classification accuracy, the initial learning rate $\eta(1)$ starts from 0.85 and increases by 0.01 until it reaches to 0.93, and the distinguishing coefficient ξ starts from 0.70 and increases by 0.05 until it reaches to 0.90. Thus, the best classification result of each record is selected from 45 tests. The termination criteria for the learning process is that the mean square error of two successive outputs in (3) is less than 0.1 or the number of learning epochs is larger than 500. All programs coded by Matlab[®] version R12 were executed by a personal computer with Pentium[®] 4 2.66-GHz processor and 512 MB RAM.

Fig. 2. Beats from Record 233 assigned to different clusters.

Figure 2 shows the best beat assignment of the 5×5 state outputs of the SOCMAC for Record 233. Each box in this figure represents a cluster of beats. Also indicated is the contents in terms of 3 categories of beats: normal beats (N), ventricular premature beats (V), and fusion of ventricular and normal beats (F). From this figure, we see that the SOCMAC uses only 22 states to cluster this 30 min record and leaves 3 empty clusters. Here, an empty cluster is a state of the SOCMAC containing nothing but its initial weights when the learning process is completed. In this case, 12 F and 2 V beats that end up in clusters where their kind are not dominant. Note that the beats plotted in the boxes are the original beats without removing their baseline wander.

Table II. Parameters achieving the best result

Record	ξ	$\eta(1)$	Record	ξ	$\eta(1)$	Record	ξ	$\eta(1)$	Record	ξ	$\eta(1)$
100	0.70	0.90	115	0.90	0.90	202	0.85	0.91	219	0.70	0.89
101	0.85	0.89	116	0.85	0.91	203	0.75	0.88	220	0.90	0.85
103	0.80	0.93	117	0.90	0.90	205	0.80	0.90	221	0.75	0.90
105	0.80	0.92	118	0.90	0.91	207	0.90	0.90	222	0.90	0.88
106	0.80	0.86	119	0.75	0.89	208	0.80	0.91	223	0.80	0.91
108	0.85	0.87	121	0.90	0.93	209	0.80	0.92	228	0.90	0.92
109	0.85	0.92	122	0.80	0.88	210	0.70	0.86	230	0.70	0.85
111	0.90	0.91	123	0.85	0.92	212	0.90	0.86	231	0.70	0.90
112	0.85	0.91	124	0.90	0.92	213	0.75	0.92	232	0.90	0.89
113	0.75	0.87	200	0.70	0.87	214	0.70	0.86	233	0.75	0.89
114	0.90	0.89	201	0.75	0.92	215	0.85	0.92	234	0.80	0.91

When the learning procedure is completed, we can use the annotation file provided by the MIT/BIH database to identify the type of each ECG beat. In this study, the label of a cluster is defined as the type of dominant beats, and a beat is defined misclassified if it ends up a cluster where the label of the cluster is different one. Table II contains parameters of the updating rule that achieve the best result in classification accuracy for each record. Table III lists the dataset for each record divided into the 4 different categories, clustering results including number of clusters used for each category, and classification accuracy. Because that there are some empty clusters in each record, the number of total clusters for the 44 records is not 1,100. From this table, we see that the majority of beats are normal type (91.63%) and, consequently, the number of clusters dominated by normal beats is high (71.22% of total resulting clusters).

Table III. Datasets and learning results

Record	Beats in Categories				Clusters in Categories				Accuracy
	N	V	F	Q	N	V	F	Q	
100	2270	1	--	--	18	1	--	--	100.00
101*	1862	--	--	2	12	--	--	2	100.00
103*	2084	--	--	--	21	--	--	--	100.00
105*	2526	41	--	5	21	1	--	1	99.77
106	1507	520	--	--	5	16	--	--	99.70
108*	1744	16	2	11	18	2	--	--	98.82
109	2491	38	2	--	13	5	--	--	99.84
111*	2123	1	--	--	19	1	--	--	100.00
112*	2539	--	--	--	21	--	--	--	100.00
113	1794	--	--	--	24	--	--	--	100.00
114*	1832	43	4	--	18	2	--	--	99.79
115*	1952	--	--	--	24	--	--	--	100.00
116	2302	109	--	--	13	7	--	--	99.96
117*	1534	--	--	--	20	--	--	--	100.00
118*	2261	16	--	10	14	2	--	2	99.91
119	1543	444	--	--	11	4	--	--	100.00
121*	1862	1	--	--	19	1	--	--	100.00
122	2476	--	--	--	22	--	--	--	100.00
123	1514	3	--	--	18	3	--	--	100.00
124	1567	47	5	--	9	6	2	--	99.81
200*	1772	826	2	--	7	17	--	--	98.88
201*	1763	198	2	37	9	3	--	5	99.65
202	2116	19	1	--	15	3	1	--	100.00
203*	2531	444	1	4	14	10	--	--	97.11
205	2574	71	11	--	8	6	2	--	99.66
207*	1649	682	--	--	9	15	--	--	98.37
208*	1587	992	372	2	11	5	5	--	96.92
209	3003	1	--	--	13	1	--	--	100.00
210*	2443	195	10	--	10	10	--	--	98.87
212*	2747	--	--	--	23	--	--	--	100.00
213	2668	220	362	--	8	9	7	--	96.09
214*	2001	256	1	2	5	15	--	--	99.69
215*	3198	164	1	--	6	15	--	--	99.94
219	2089	64	1	133	8	6	--	5	97.38
220	2047	--	--	--	22	--	--	--	100.00
221*	2031	396	--	--	9	11	--	--	100.00
222*	2482	--	--	--	22	--	--	--	100.00
223*	2118	473	14	--	6	12	--	--	98.31
228*	1691	362	--	--	10	13	--	--	99.85
230*	2254	1	--	--	19	1	--	--	100.00
231	1569	2	--	2	7	2	--	2	100.00
232*	1780	--	--	--	18	--	--	--	100.00
233	2236	830	11	--	10	12	--	--	99.55
234*	2750	3	--	--	17	2	--	--	100.00
Total	92882	7479	802	208	626	219	17	17	99.41
		101381				879			

* This record is noisy.

With the optimal parameters listed in Table II, the SOCMAC network based ECG classifier achieves an overall classification accuracy of 99.41%. Moreover, 21 out of 44 records have a classification accuracy of 100%. The worst two cases occur on Records 208 and 213, which still have classification accuracies of 96.92% and 96.09%, respectively. The common feature of these two records is that they contain lots of F beats. The F beat is a mixture between V (ventricular) and N (normal) beats, as defined in Table I, and is hardly classified correctly. It is interesting to know the classification results of the 4 different beats. Table IV compares them. It can be seen that the classification accuracies of F and Q beats are relatively low when compared with N or V beat. However, they are still better than the results given in [6], where 75.1% for category F and 6.1% for category Q were reported.

Table IV. Classification results of different categories

Category	No. of Testing Beats	No. of Correctly Classified Beats	Accuracy
N	92,882	92,770	99.88
V	7,479	7,262	97.10
F	802	619	77.18
Q	208	126	60.58

Table V. Comparison of classification results of clean and noisy records

Record	No. of Testing Beats	No. of Correctly Classified Beats	Accuracy
Noisy	62,708	62,338	99.41
Clean	38,663	38,439	99.42

Table V compares the classification results of clean and noisy records. From this table we see that the classification results differ not much. Without any data pre-processing steps, the simulation results show that the proposed ECG beat classifier is robust against noise and achieves a good classification result.

Finally, we compare the obtained results with the results presented in the literature by using different networks such as multilayer perceptron (MLP) [3-4], self-organizing map (SOM) [5-6], multichannel ART (MART) [7], and combined neural network for ECG beat classification [8-9]. Table VI summarizes the results of ECG beat classification using different networks. Column 3 of this table shows number of categories used in those works, and the last column shows their classification results. It should be noted that in those works, though the used ECG data all came from the MIT/BIH Database, different number of categories, different records, different amount of data, and different signal conditioning methods were used.

Thus, it is really hard to compare the results in a fair way.

Table VI. Comparative results of the ECG beat classification

Reference	Method	No. of Categories	Acc (%)
Proposed method	SOCMAC	4	99.41
[3]	MLP	3	94.09
[4]	MLP	3	98
[5]	SOM	4	94
[6]	SOM	16	98.5
[7]	MART	2-4	96.6
[8]	Fuzzy c means + MLP	4	93.5
[9]	MLP + MLP	4	96.94

5. CONCLUSIONS

A SOCMAC network based ECG classifier has been proposed. The MIT/BIH database was used for the simulation. For each selected record, we search the optimal parameters for achieving the best classification accuracy. As a result, the classifier achieves an overall classification accuracy of 99.41%. The simulation results also show that the proposed classifier is also robust against noise.

ACKNOWLEDGMENT

This work was supported by the National Science Council, Taiwan, Republic of China, under Grant NSC 94-2213-E-262-005.

REFERENCES

- De Chazal P, O'Dwyer M, and Ropella KM: Automatic classification of heartbeats using ECG morphology and heartbeat interval features, *IEEE Trans. Biomed. Eng.* 2004; 51: 1196-1206.
- Tümer MB, Belfore LA II, and Ropella KM: A syntactic methodology for automatic diagnosis by analysis of continuous time measurements using hierarchical signal representations, *IEEE Trans. Syst., Man, Cybern., B.* 2003; 33: 951-965
- Vargas F, Lettinn D, de Castro MCF, and Macarthy M: Electrocardiogram pattern recognition by means of MLP network and PCA: A case study on equal amount of input signal types, in *Proc. of the VII Brazilian Symposium on Neural Networks*, 2002.
- Minami K, Nakajima H, and Toyoshima T: Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network, *IEEE Trans. Biomed. Eng.* 1999; 46: 179-185.
- Hu YH, Palreddy S, and Tompkins WJ: A patient-adaptable ECG beat classifier using a mixture of experts approach, *IEEE Trans. Biomed. Eng.* 1997; 44: 891-900.
- Lagerholm M, Peterson C, Braccini G, Edenbrandt L, and Sörnmo L: Clustering ECG complexes using Hermite functions and self-organizing maps, *IEEE Trans. Biomed. Eng.* 2000; 47: 838-848.
- Fernández-Delgado M, and Barro Ameneiro S: MART: A multichannel ART-based neural network, *IEEE Trans. Neural Netw.* 1998; 9: 139-150.
- Engin M: ECG beat classification using neuro-fuzzy network, *Pattern Recogn. Lett.* 2004; 25: 1715-1722.
- Güler, İ., and Übeyli ED: ECG beat classifier designed by combined neural network model, *Pattern Recogn.* 2005; 38: 199-208.
- Yeh MF and Chang KC: A self-organizing CMAC network with gray credit assignment, *IEEE Trans. Syst., Man, Cybern., B.* 2006; 36: 623-635.
- Albus J S: Data storage in the cerebellar model articulation controller (CMAC), *J. Dyn. Syst., Meas., Contr., Trans. ASME* 1975; 97: 228-233.
- Albus JS: *Brains, Behavior and Robotics*, New York: McGraw-Hill, 1981.
- Deng JL: Introduction to gray system theory, *J. Gray Syst.* 1989; 1: 1-24.
- Deng JL: Control problems of gray systems, *Syst. Contr. Lett.* 1982; 5: 288-294.
- MIT/BIH database distributor, Beth Israel Hospital, Biomedical Engineering, Division available in the address: <http://ecg.mit.edu/>, USA, 1979.
- Recommended Practice for Testing and Reporting Performance Results of Ventricular Arrhythmia Detection Algorithms, Association for the Advancement of Medical Instrumentation, 1987.