Proceedings of the 29th Annual International
Conference of the IEEE EMBS
Cité Internationale, Lyon, France
August 23-26, 2007.

ALGORITHM FOR CLASSIFYING ARRHYTHMIA USING EXTREME LEARNING MACHINE AND PRINCIPAL COMPONENT ANALYSIS

Jinkwon Kim, Hangsik Shin, Yonwook Lee, Myoungho Lee

Abstract— **In this paper, we developed the novel algorithm for cardiac arrhythmia classification. Until now, back propagation neural network (BPNN) was frequently used for these tasks. However, general gradient based learning method is far slower than what is required for their application. The proposed algorithm adapts Extreme Learning Machine(ELM) that has the advantage of very fast learning speed and high accuracy. In this paper, we classify beats into normal beat, left bundle branch block beat, right bundle branch block beat, premature ventricular contraction, atrial premature beat, paced beat, and ventricular escape beat. Experimental results show that we can obtain 97.45 % in average accuracy, 97.44 % in average sensitivity, 98.46 % in average specificity, and 2.423 seconds in learning time.**

I. INTRODUCTION

For past decades, the arrhythmia classification algorithms have been developed. There are researches about methods for classifying arrhythmia by statistical method[1] and fuzzy[2]. And neural network algorithm is frequently used for these tasks[3,4]. Back propagation neural network(BPNN) is the learning algorithm for neural network, but that is based on gradient based learning algorithm, so BPNN is generally very slow due to improper learning steps or easily converge to local minima[5].

ELM is a novel learning algorithm for Single hidden Layer Feedforward Neural networks (SLFNs) which is a kind of artificial neural network. That has the advantage of vary fast learning speed and high accuracy[5]. Preceding researches reported ELM can learn thousands of times faster than conventional popular learning algorithms for feed forward neural networks like BPNN without loss for accuracy. Because these advantages, ELM has many possibilities to be adapted many applications.

In this research, we developed novel arrhythmia classification algorithm using principal component analysis(PCA) and ELM. We intended to classify 7 ECG beat types, including normal rhythm and 6 arrhythmias. And we evaluated accuracy for classifying arrhythmias and learning time through simulation.

J. Kim is with Department of Electrical and Electronic Engineering, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul, Korea (e-mail: jinkwon-mailbox@yonsei.ac.kr)
H. Shin is with Department of Electrical and Electronic Engineering, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul, Korea (e-mail: glority@yonsei.ac.kr)
Y. Lee is with School of Electrical and Electronic Engineering, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul, Korea (e-mail: superbaduki@nate.com)
M. Lee is with Department of Electrical and Electronic Engineering, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul, Korea (corresponding author to provide phone: +82-2-2123-4946; fax: +82-2-312-2770; e-mail: mhlee@yonsei.ac.kr).
This work is supported in part by MIC & IITA through IT Leading R&D Support Project (Project No. 2005-S-093)

II. STRUCTURE OF ALGORITHM

Fig. 1 shows the proposed algorithm uses three kinds of information which are R peak amplitude, instance RR interval and R peak morphology data. Those data are obtained from MIT-BIH arrhythmia database. From MIT-BIH arrhythmia database, we extracted seven ECG beat types which are normal, left bundle branch block(LBBB), right bundle branch block(RBBB), premature ventricular contraction(PVC), atrial premature beat(APB), paced beat, and ventricular escape beat. First, we select 500ms(180 samples) signals around beat annotation because most of the information for classification lie around R peak. The R peak amplitude information is calculated by extracting mean value from R peak value. And instance RR interval means time interval between beat annotations. Dimension of R peak morphology signal is reduced by PCA, and then is used for feature vector of ELM.

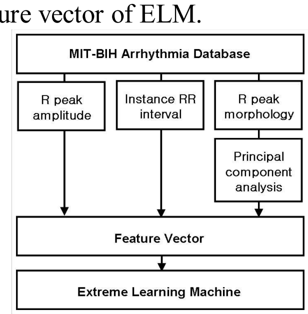

Fig. 1 Block diagram of proposed algorithm

A. Principal Component Analysis

PCA is a kind of method for reducing data dimension with small loss. When d, dimension of original data, is reduced to d' by PCA, we can represent the error $J_{d'}$ as the sum of mean square error (see eq. (1)):

$$J_{d'} = \sum_{k=1}^{n} \left\| (\mathbf{m} + \sum_{i=1}^{d'} a_{ki}\mathbf{e}_i) - \mathbf{x}_k \right\|^2 \quad (1)$$

To minimize $J_{d'}$, we substitute covariance matrix $\mathbf{S}$ into eq. (1). Then we can get minimum value of $\mathbf{e}_i$ using Lagrange multiplier λ_i:

Covariance matrix: $\mathbf{S}=\sum_{k=1}^{n}(\mathbf{x}_k-\mathbf{m})(\mathbf{x}_k-\mathbf{m})^t$

$$u=\mathbf{e}_i{}^t\mathbf{S}\mathbf{e}_i-\lambda_i(\mathbf{e}_i{}^t\mathbf{e}_i-1)$$

$$\mathbf{S}\mathbf{e}_i=\lambda_i\mathbf{e}_i$$

When $\mathbf{e}_i$ is eigenvector of $\mathbf{S}$, $J_{d'}$ can reach to the minimum. Because covariance matrix $\mathbf{S}$ is real and symmetric matrix, that's eigenvector is mutually uncorrelated and orthogonal. So, we can get down dimensional space spanned with eigenvectors ordered by eigenvalues with minimum loss.

B. Extreme Learning Machine

ELM which overcomes defects of BPNN is a novel learning algorithm for SLFN. BPNN is a generally used learning algorithm for artificial neural network among gradient based algorithm. Gradient based algorithm adjusts weights between neurons from output layer to input layer. Though this process there exists dependency between input weights and output weights. Some researches have been shown that SLFNs having including N hidden neurons with randomly chosen input weights can learn N distinct patterns with arbitrarily small error[6]. ELM is based on this result and has learning process with arbitrary chosen input weights and biases of hidden neurons[5]. For approximation of SLFNs, when we have N arbitrary distinct samples ($\mathbf{x}_i,\mathbf{t}_i$), we can model SLFNs as eq. (2),

$$\sum_{i=1}^{N_h}\boldsymbol{\beta}_i\, g(\mathbf{w}_i\cdot\mathbf{x}_j+b_i)=\mathbf{t}_j,\qquad j=1,\cdots,N \tag{2}$$

Where $\mathbf{x}_j=[x_{j1},x_{j2},\cdots,x_{jn}]^T$, $\mathbf{t}_j=[t_{j1},t_{j2},\cdots,t_{jm}]^T$ represents i th input vector and output vector, b_i is bias of i th hidden neuron, $\mathbf{w}_i=[w_{i1},w_{i2},\cdots,w_{in}]^T$ input weight vector connecting i th hidden neuron to input layer, $\boldsymbol{\beta}_i=[\beta_{i1},\beta_{i2},\cdots,\beta_{im}]^T$ output weight vector connecting i th hidden neuron to output layer, and SLFNs have N_h hidden neurons and activation function $g(\cdot)$. The eq. (2) is represented by matrix equation as:

$$\mathbf{H}\cdot\boldsymbol{\beta}=\mathbf{T} \tag{3}$$

$$\mathbf{H}(\mathbf{w}_1,\cdots,\mathbf{w}_{N_h},b_1,\cdots,b_{N_h},\mathbf{x}_1,\cdots,\mathbf{x}_{N_h})=\begin{bmatrix} g(\mathbf{w}_1\cdot\mathbf{x}_1+b_1) & \cdots & g(\mathbf{w}_{N_h}\cdot\mathbf{x}_1+b_{N_h}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1\cdot\mathbf{x}_N+b_1) & \cdots & g(\mathbf{w}_{N_h}\cdot\mathbf{x}_1+b_{N_h}) \end{bmatrix}_{N\times N_h}$$

$$\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_1{}^T\\ \vdots \\ \boldsymbol{\beta}_{N_h}{}^T\end{bmatrix} \text{ and } \mathbf{T}=\begin{bmatrix}\mathbf{t}_1{}^T\\ \vdots \\ \mathbf{t}_{N_h}{}^T\end{bmatrix}$$

Each component of $\mathbf{H}$ represent output of hidden layer. When input weights $\mathbf{w}_i$ and biases b_i of hidden neuron are invariable, $\mathbf{H}$ is determined with input vector $\mathbf{x}_j$. In that case SLFNs are linear system. So, In case of $\mathbf{H}$ has inverse matrix, we can get $\boldsymbol{\beta}$ through $\mathbf{H}^{-1}\cdot\mathbf{T}$. But generally number of samples is greater than number of hidden neurons, $\mathbf{H}$ is a nonsquare matrix and there may not exist $\mathbf{H}^{-1}$. The optimal output weights $\hat{\boldsymbol{\beta}}$ guarantee minimum difference between $\mathbf{H}\boldsymbol{\beta}$ and $\mathbf{T}$ as:

$$\left\|\mathbf{H}\hat{\boldsymbol{\beta}}-\mathbf{T}\right\|=\left\|\mathbf{H}\,\mathbf{H}^{\dagger}\mathbf{T}-\mathbf{T}\right\|=\min_{\boldsymbol{\beta}}\left\|\mathbf{H}\boldsymbol{\beta}-\mathbf{Y}\right\| \tag{4}$$

Using Moor-Penrose generalized inverse $\mathbf{H}^{\dagger}$ we can get minimum norm least-squares solution of (4). That case has the optimum value of $\hat{\boldsymbol{\beta}}$ [5].

The process of ELM for SLFMs learning algorithm is expressed below:

1. Choose arbitrary value for input weights $\mathbf{w}_i$ and biases b_i of hidden neurons.
2. Calculate hidden layer output matrix $\mathbf{H}$.
3. Obtain the optimal $\hat{\boldsymbol{\beta}}$ using $\hat{\boldsymbol{\beta}}=\mathbf{H}^{\dagger}\mathbf{T}$.

Because learning process of ELM randomly chooses the input weights and analytically determines the output weights of SLFNs, there are no iteration processes and extremely smaller learning time of ELM than one of BPNN.

C. experimental method

For evaluation of proposed algorithm we choose 7 beat types among MIT-BIH arrhythmia database. We choose all beat types which exist over 100 beats at one data file. And training and test data set are formed of date pool which contains 100 beats from each data file. In the event we get 3450 beats as Training and test data set. The composition of data set is presented in Table 1.

Morphology data is extracted mean value then divided by standard deviation to normalization. Normalized morphology data processed by PCA include feature vector with R peak information and instance RR interval. We evaluate proposed algorithm with variable number of principal components and number of hidden neurons to get optimal performance of proposed algorithm.

TABLE 1

The composition of data set (Normal : N, LBBB : L, RBBB : R, PVC : V, APB : A, Paced beat : P, Ventricular escape beat : E)

Type	MIT-BIH data file	Learning	Test
N	100, 101, 103, 105, 106, 108, 112, 113, 115, 116, 117, 119, 121, 122, 123, 200, 201, 202, 203, 205, 208, 209, 210, 212, 213, 215, 217, 219, 220, 221, 222, 223, 228, 230, 231, 233, 234	50	50
L	109, 111, 207, 214	50	50
R	118, 124, 212, 231, 232	50	50
V	106, 116, 119, 200, 201, 203, 207, 208, 210, 213, 214, 215, 217, 221, 223, 228, 233	50	50
A	207, 209, 222, 232	50	50
P	107	50	50
E	207	50	50
Total		3450	3450

III. RESULT

For making template data of PCA we randomly select two samples each of data file included above dataset. So we can use 138 samples as template data. The results of PCA are represented in Table 2. The training data set and test data set are projected into space made up with principal component of template data.

TABLE 2
Contribution of principal components

PC	Relative Contribution (%)	Density Contribution (%)
1	46.61124	46.61124
2	22.96032	69.57156
3	9.617746	79.18931
5	3.153082	89.2179
10	0.646041	96.81905
20	0.091612	99.31434
30	0.026796	99.7451
40	0.009994	99.90025

The classifying performance is evaluated by the following three criteria:

$$Sensitivity(ST) = \frac{TP}{TP + FN} \times 100$$

$$Specificity(SF) = \frac{TN}{TN + FP} \times 100$$

$$Accuracy(AC) = \frac{(TP + TN)}{(TP + FN + TN + FP)} \times 100$$

Refer to fig 2, which illustrate the relationship among number of principal components, accuracy, sensitivity and specificity. In this simulation we vary number of principal components from 1 to 40 with 300 hidden neurons.

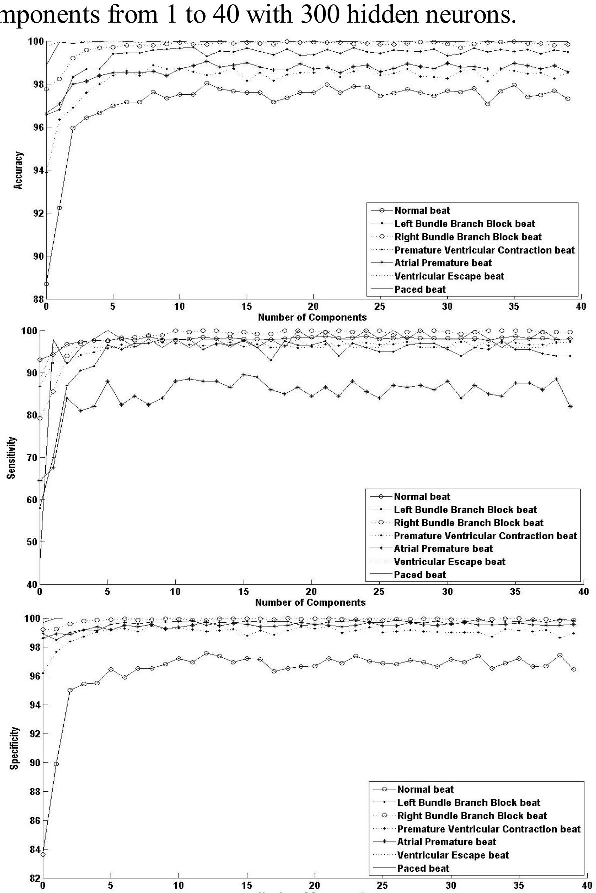

Fig. 2 Evaluation results of proposed algorithm in aspects of number of principal components

The result shows that there is no more improvement after 10 components that have 96% average accuracy. And the maximum average accuracy is shown at 22 components.

The relationship among number of hidden neurons, accuracy, sensitivity and specificity are shown in fig 3. In this simulation we vary number of hidden neuron from 1 to 300 with 22 components.

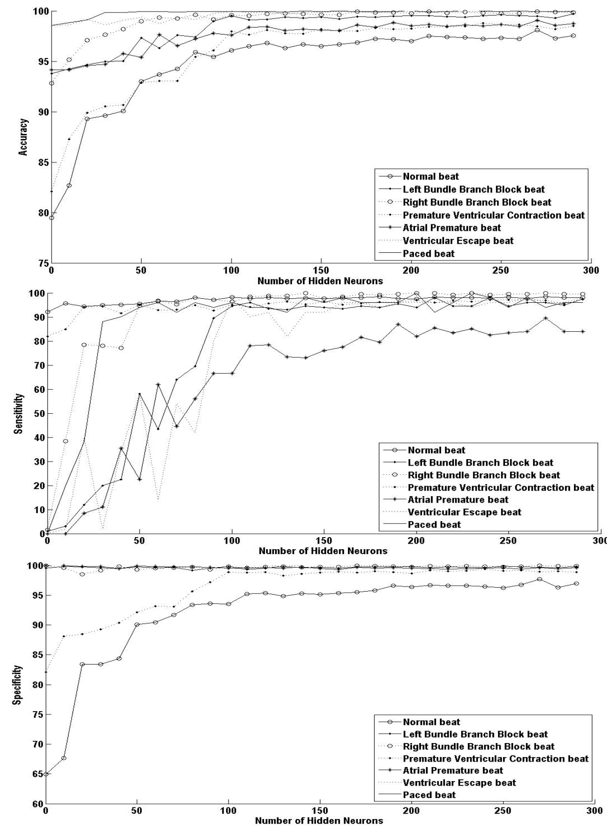

Fig. 3 Evaluation results of proposed algorithm in aspects of number of hidden neurons

The result shows that there is no more improvement after 170 hidden neurons that have 96% average accuracy. And the maximum average accuracy is shown at 280 hidden neurons. The optimal result is shown when using 280 hidden neurons and 22 principal components. Simulation results at that case are listed in table 3. The interesting point of this result is that sensitivity of APB is 89.5 % which is rather low performance compared with one of other beat types.

TABLE 3
Results of proposed algorithm with 280 hidden neurons and 22 components (Normal : N, LBBB : L, RBBB : R, PVC : V, APB : A, Paced beat : P, Ventricular escape beat : E)

actual / estimated	N	L	R	V	A	P	E	Total
N	1822	2	0	16	17	1	1	1859
L	3	192	0	6	1	0	0	202
R	2	0	250	0	0	0	0	252
V	15	6	0	824	3	1	2	851
A	8	0	0	4	179	0	0	191
P	0	0	0	0	0	48	0	48
E	0	0	0	0	0	0	47	47

Total	1850	200	250	850	200	50	50	
Accuracy, Sensitivity, Specificity (%)								
	N	L	R	V	A	P	E	Avg.
Accuracy	98.12	99.48	99.94	98.46	99.04	99.94	99.91	97.45
Sensitivity	98.49	96.00	100	96.94	89.50	96.00	94.00	97.44
Specificity	97.69	99.69	99.94	98.96	99.63	100	100	98.46

We evaluated proposed algorithms learning time with algorithm using BPNN. And simulation is performed in Matlab 7.1 with AMD Athlon 64 Processor 3000+ 1.80 GHz, 1 GB RAM. In table 4, learning speed of proposed algorithm is faster thousands of times then algorithm using BPNN. Algorithm using BPNN has very low sensitivity performance. Algorithm using BPNN converged at local minima with our feature vector.

TABLE 4
Evaluation of learning time among algorithm using ELM or BPNN

Classifier	Learning rate	Number of training iteration	Learning time (sec)	Accuracy
ELM	-	-	2.423	97.45
BPNN	0.0001	2000	2154.734	93.22
BPNN	0.0002	2000	2284.844	88.35
BPNN	0.0001	3000	4187.391	93.68

IV. Discussion

In this paper we evaluated a novel algorithm for classifying arrhythmia using ELM in aspects of accuracy, sensitivity, specificity, and learning time. As a result we obtain 97.45 % in average accuracy, 97.44 % in average sensitivity, 98.46 % in average specificity, and 2.423 seconds in learning time. A summary of the preceding results obtained for arrhythmic beat classification by other methods is shown in Table 5.

TABLE 5
Results of preceding research
* BSS-Fourier[10] didn't use MIT-BIH arrhythmia database as training and test data set

Method	Number of beat types (Training Data Set/Test Data Set)	Accuracy
MOE [7]	4 (about 10000/49260)	94.0 %
FTNN [8]	3 (540/250)	98.0 %
Fhyb-HOSA[9]	7 (4035/3150)	96.06 %
BSS-Fourier[10]	5 (320/160)*	85.04 %
DWT-NN[11]	13 (30293/75130)	96.79 %
ICA-BPNN[4]	8 (4900/4900)	98.37 %
Proposed Algorithm	7 (3450/3450)	97.45 %

V. Conclusion

We developed a novel algorithm for classifying arrhythmia using ELM and evaluated that in the phase of classifying performance and learning speed. Proposed algorithm has remarkable learning speed and classifying performance. But sensitivity of APB is rather lower than other beat types. Because APB have ectopic pacemaker in atrium, R peak morphology of APC is resemble to Normal one. Our feature vector can't manage this problem. We plan to solve this defect in future research.

Acknowledgment

This work is supported in part by MIC & IITA through IT Leading R&D Support Project (Project No. 2006-S-093)

References

[1] Coast D.A., Stern R.M., Cano G.G., Briller S.A.,“An approach to cardiac arrhythmia analysis using hidden Markov models”, Biomedical Engineering, IEEE Transactions on, Volume 37, Issue 9, Page(s):826 - 836, Sept. 1990

[2] Ham F.M. Soowhan Han,“Classification of cardiac arrhythmias using fuzzy ARTMAP”, Biomedical Engineering, IEEE Transactions on Volume 43, Issue 4, Page(s):425 - 429, April 1996

[3] Inan O. T., Giovangrandi L., Kovacs G. T. A., “Robust Neural -Network -Based Classification of Premature Ventricular Contractions Using Wavelet Transform and Timing Interval Features” Biomedical Engineering, IEEE Transactions on, Volume 53, Issue 12, Part 1, Page(s):2507 - 2515, Dec. 2006.

[4] Sung-Nien Yu, Kuan-To Chou, “Combining Independent Component Analysis and Backpropagation Neural Network for ECG Beat Classification”, Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE, Page(s):3090 - 3093, Aug. 2006.

[5] Guang-Bin Huang, Qin-Yu Zhu and Chee-Kheong Siew, "Extreme learning machine: Theory and applications" Neurocomputing, Volume 70, Issues 1-3, Pages 489-501, December 2006.

[6] S. Tamura and M. Tateichi, “Capabilities of a four-layered feedforward neural network: Four layers versus three”, IEEE Transactions on Neural Networks, vol. 8, no. 2, pp/ 251~255, 1997.

[7] Y. H. Hu, S. Palreddy, and W. J. Tompkins, “A Patient-Adaptable ECG beat classifier using a mixture of experts approach,” IEEE Trans. Biomed. Eng., Vol. 44, No. 9, pp. 891-900., 1997.

[8] K. Minami, H. Nakajima, and T. Toyoshima, “Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network,” IEEE Trans. on Biomed. Eng., Vol. 46, No. 2, pp. 179-185 1999.

[9] S. Osowski, , and T. H. Linh, “ECG beat recognition using fuzzy hybrid neural network,” IEEE Trans. Biomed. Eng., Vol. 48, pp. 1265-1271, 2001.

[10] M.I. Owis, A.-B.M. Youssef, and Y.M. Kadah,“Characterization of ECG signals based on blind source separation,” Medical & Biological Engineering & Computing, Vol. 40 , pp.557-564, 2002.

[11] G. K. Prasad, and J. S. Sahambi, “Classification of ECG arrhythmias using multi-resolution analysis and neural networks,” Proc. IEEE Conf. on Convergent Technologies (Tecon2003), Bangalore, India, Vol. 1, pp. 227-231, 2003.