International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 22 (2018) pp. 16088-16092

Neural Network and Genetic Algorithm Based ECG Beat Classification

Anwar Al-Shrouf, **Ahmad Khaleel AlOmari,**

Department of Biomedical Equipment Technology,
Prince Sattam Bin Abdul-Aziz University, Al Kharj, Saudi Arabia.

Abstract

This paper proposes a potential cascaded neural network algorithm for ECG Holter system beat classification. The system works in real time and is capable of recognizing up to 20 artificial QRS templates. The parallelism of neural network increases the efficiency of computations. An error signal derived from differences between predictor and testing signal is used in the classification. A neural network is used to generate linear predictions for signals. Another neural network generates error signals measured between predictions taken from the first neural network and testing signal. A fifth neural network does the classifications utilizing the error signals instead of complex raw signal. Three Hermite functions are used in generating testing signals with and without noise. Proper thresholding for the error signals is essential for classifier immunity. The result is a compact, online, efficient, and hardware realizable signals classifier that uses minimal compressed error signal.

Keywords: Neural Networks, ECG, Self-Organizing, Maps, Genetic Algorithm,, Arrhythmia.

1. INTRODUCTION

The early ECG analysis is one of the most valuable cardiac tests. It helps in early arrhythmia diagnosis.

Holter systems can recognize up to 40 different morphologies according to templates created individually for each patient; this creates a significant computational load. In this paper we propose a method that used linear prediction coefficients in finding a three-state error template. A simple neural trained with the genetic algorithm (GA) is used to predict the signal directly after a short phase of training [1]. Then another network connected in series with the first network is used to measure the difference between the original Signal and the predicted signal. The output, which is made as a template consisting of 1, 0,-1 trits is stored in another neural network using unsupervised learning algorithm.

Each time the unsupervised neural network is subjected to a new template, it either classifies it to an existing class or it considers it as distinguished new pattern and, consequently, adds new class. Eventually, the number of classes will reach its maximum possible value. Keeping in mind the parallel nature of neural networks and the fact that linear prediction coefficients can be used as constants for different beats and different patients, then this method is an efficient online technique that can be on real time to help in providing crucial medical decisions. A lot of studies had previously been done on linear predictions and patterns classification [2] and [3], but cascading neural networks and using error ternary patterns help in data reduction, cuts processing effort, and increases noise immunity. In this paper, and as a first step we use Hermite functions to model ECG signals that are used for testing the classifier, however, real ECG signals are used in building the predictor. Future work will include using testing signals from arrhythmia database.

2. LINEAR PREDICTION METHOD (LPM)

The actual ECG signal y (i) can be estimated by another sequence $\check{y}(i)$ determined by unique set of predictors coefficients and P (i) sample signals. That is

$$\check{y}(j) = \sum_{k=1}^{p} a(k)y(j-k) \qquad (1)$$

Where a(k) is the k^{th} linear predictive coefficient. Upper part of Figure (2) shows a two-layer neural network(NN) topology (input and hidden layers) for ECG linear predictions, We employed genetic algorithm (GA) to solve quickly for the coefficients. The GA is a massively parallel algorithm used for global optimization.

Table 1: (Two parts) First two LPM coefficients for 10 beats from 5 different ECG records

Record	**N13**		**N14**		**N15**		**V74**		**V77**	
Beat	a(1)	a(2)	a(1)	a(2)	a(1)	a(2)	a(1)	a(2)	a(1)	a(2)
1	-1.88	0.76	-1.83	0.92	-1.86	0.87	-1.80	0.91	-1.80	0.91
2	-1.96	0.77	-1.82	0.71	-1.97	0.86	-1.90	0.81	-1.95	0.83
3	-1.88	0.96	-1.83	0.95	-1.89	0.78	-1.85	0.96	-1.81	0.93
4	-1.83	0.87	-1.85	0.94	-1.92	0.94	-1.74	0.85	-1.74	0.85
5	-1.87	0.86	-1.84	0.93	-1.44	0.38	-1.85	0.95	-1.85	0.95
6	-1.79	0.82	-1.85	0.93	-1.72	0.63	-1.85	0.94	-1.76	0.85
7	-1.72	0.81	-1.85	0.93	-1.85	0.88	-1.85	0.94	-1.85	0.94
8	-1.82	0.91	-1.83	0.91	-1.79	0.89	-1.86	0.98	-1.86	0.98
9	-1.90	0.79	-1.86	0.94	-1.75	0.83	-1.89	0.97	-1.86	0.97
10	-1.73	0.83	-1.95	0.84	-1.81	0.70	-1.81	0.92	-1.78	0.86

The GA can quickly train the single neuron neural network as described in [4], minimizing the residual error signal e(i) to almost zero. e(i) is given by:

$$e(i) = y(i) - \check{y}(i) = y(i) + \sum_{k=1}^{p} a(k)y(j-k) \qquad (2)$$

a(k) is a prediction error coefficient and it is equal to one at k = 0

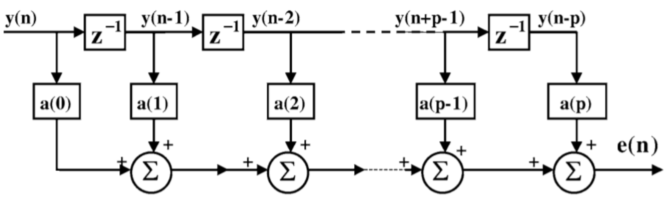

Figure 1: P-Order linear prediction

To calculate the mean squared value of error M (e (i)) we use

$$M = \mathit{Mean\ squared\ Value}\left[\left(y(i) + \sum_{k=1}^{p} a(k)y(j-k)\right)^2\right] \qquad (3)$$

Here, M denotes square of prediction error. The optimal coefficients a (0)…a (p) could be achieved when M is minimum. This result could be obtained by solving the following normal equation;

$$\begin{bmatrix} R(0) & \cdots & R(p-1) \\ \vdots & \ddots & \vdots \\ R(p-1) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a(1) \\ \vdots \\ a(p) \end{bmatrix} = \begin{bmatrix} \hat{R}(1) \\ \vdots \\ \hat{R}(p) \end{bmatrix} \qquad (4)$$

The above equation is symmetric and Toeplitz and can be solved for the linear predictive coefficients a(i) by using Levinson-Dubrin's algorithm. By applying Levinson-Durbin's recursive procedure for optimal production filter coefficient in $p^2 + O(p)$ operations [23], noting that the optimal order of prediction has to determine first. One of the important decisions that usually have to be made in the linear prediction method is the determination of the optimal order of prediction. According to our assumption, that is the minimum value of *p* is adequate for compression purpose, as usually done in most ECG signals were the residual flatness criterion is used [24]. In this study, it is considered p=2 for most ECG signal used.

The key of the Levinson-Durbin's solution method model of 250 points was used and training was repeated until the residual error was minimized to preset measure. The process is repeated for every record until the list of shown LPM coefficients was generated. In our predictor NN we used a two weights (two coefficients) single. The key to the Levinson-Durbin's method solution, that exploits the Toeplitz property of the matrix, is to proceed recursively, beginning with a predictor of order one, and then to increase the order recursively using lower-order solutions to obtain the solution to the next higher Levinson-Durbin's algorithm .

The values of weights for the NN are the values of the linear predictive coefficients. First, we had to verify whether LPM coefficients had to be individually computed for each ECG beat. 5 records was randomly picked from the AHA arrhythmia database and the coefficients for 10 beats were computed Values of the coefficients are shown in Table 1. The results for order of P= 2 is presented. The second order prediction was chosen based on the studies of [5], [9] and [6] which reported that the second order prediction is sufficient for ECG signal classification. For this reason P=2 was considered in generating predicted signals.

Observing the coefficients a (l) and a(2) values, it's obvious that there is a great similarity between the values of coefficients for different beats for a single patient (record). Therefore, it is enough to keep these coefficients constant at some average value during beat classification.

During training of the predictor NN, a reference model of 250 points was used and training was repeated until residual error was minimized to preset measure. The process is repeated for every record until the list of shown LPM coefficients was generated. In our predictor NN, a two weights (two coefficients) single neuron NN was used. The first weight was the average of a(1)'s and the second weight was the average of a(2). The aim here is to build some reference model for normal signals that can create a minimal error signal. The independent output of the reference model itself is not our first concern.

3. THE CLASSIFIER DESIGN

The second stage is building the neural network that outputs the ternary vector that carries information about the class of the ECG beat. The NN for this part is a single neuron, NN with an activation function of three levels (-1, 0, +1). This network measures the difference between the ECG under classification and the associated predicted signal every five steps, then this error signal which is much smaller than original ECG signal is used as a blueprint for the ECG signal. This blueprint is a self-organizing neural network that uses 20 neurons in a (4X5) topology. This type of networks is known as a Self-Organizing Maps (SOM) and was originally proposed by Kohonen [7]. This type of networks has the ability to force adjacent neurons in the feature map (network) to respond to similar feature inputs. Figure (2) shows a diagram explaining the architecture of this all-neural classifier.

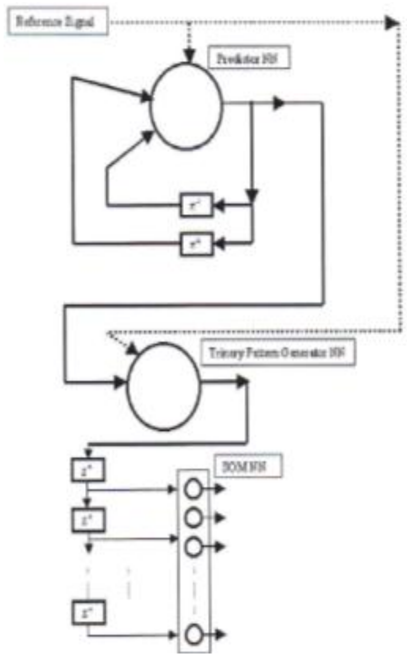

Figure 2: Cascaded neural network classifier.

The Hermite functions are used to model QRS complexes. This was originally proposed by Sormo et al. [5]. Although this model does not reflect optimum modeling of a real-life ECG with all the noise and irregularities associated with it, it offers excellent testing signals before a move toward real ECG is made. Based on our observations on this artificial model we study the learning capabilities and sensitivity of our classifier. We generated a series of 40 QRS shapes using linear combinations of the first five Hermite functions $\Phi_0, \Phi_1 \ldots \Phi_5$:

$$ecg(t) = \begin{cases} \frac{N-2j}{N}\Phi_0(t,\sigma) + \frac{2j}{N}\Phi_1(t,\sigma) & 0 \le j < 10 \\ \frac{N-j}{N}\Phi_0(t,\sigma) + \frac{2N-2j}{N}\Phi_1(t,\sigma) & 10 \le j < 20 \\ \frac{N-2j}{N}\Phi_0(t,\sigma) + \frac{2j}{N}\Phi_1(t,\sigma) + \frac{j}{N}\Phi_2(t,\sigma) + \frac{1.2(N-j)}{N}\Phi_3(t,\sigma) + \frac{0.4(N-j)}{N}\Phi_4(t,\sigma) & 20 \le j < 30 \\ \frac{N-2j}{N}\Phi_0(t,\sigma) + \frac{2j}{N}\Phi_1(t,\sigma) + \frac{j}{N}\Phi_2(t,\sigma) + \frac{0.4(N-j)}{N}\Phi_3(t,\sigma) + \frac{1.2(N-j)}{N}\Phi_4(t,\sigma) & 30 \le j < 40 \end{cases} \quad (5)$$

The first five Hermite functions are shown in Figure (3).

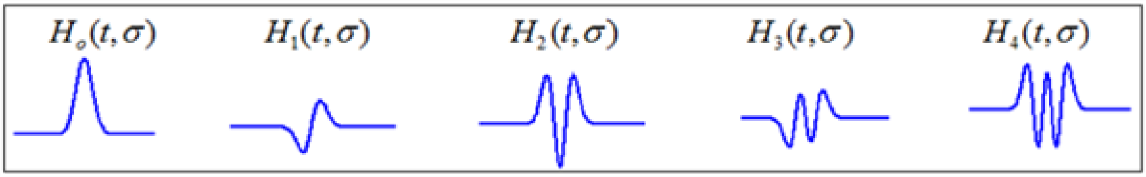

Figure 3: First five Hermite functions

N was equal to 40. With those templates we could fluently change its morphology from mono phasic to biphasic and from biphasic to three-phasic. Figure (4) show 40 typical artificial ECG signals. The ternary error signals were generated after every 5 consecutive samples from the artificial ECG signal (Hermite function Signal).

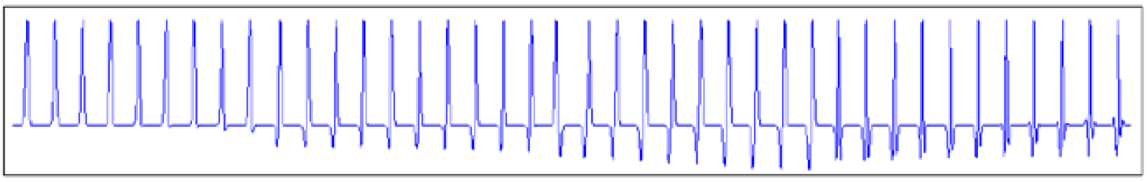

Figure 4: 40 QRS shapes using linear combinations of the first five Hermite

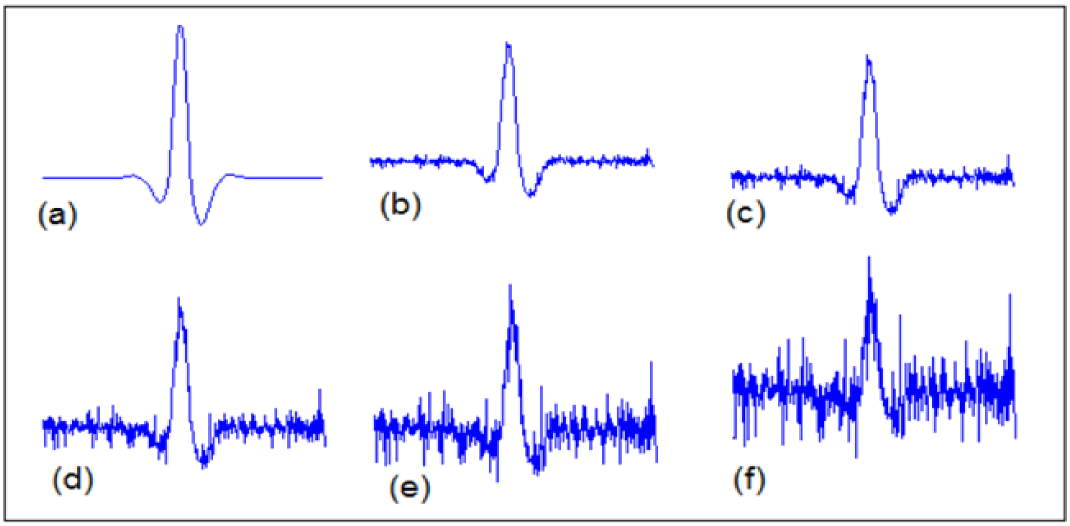

Figure 5: Generated ECG signal with different SNR levels.

4. TESTING ROBUSTNESS OF CLASSIFIER

Further testing of the classifier was done by adding white noise to the artificial ECG signal at the input of the reference signal. This was done only at retrievals without any new learning phases. The added noise generated signal-to-noise ratio with levels from 20 to 0 dB.

Figure (5) shows artificially generated signal, where (5a) represents a noiseless generated signal, and (5b), (5c), (5d), (5e) and (5f) represent the signal after adding white noise with SNR 20 dB, 15 dB, 10 dB, 5 dB and 0 dB, respectively.

The classifier did 100% classification with SNR down to 15 dB. Results are depicted in table:

Table 3: Percentage of classification by SOM under different SNR

SNR	*20dB*	*15dB*	*0dB*	*5dB*	*0dB*
Correct class	100%	100%	85%	72%	58%

In Table (3), figures (4) and (5) shows the artificial ECG beats with worst case added noise, i.e., SNR is 0dB. We did not go below the 0dB SNR in our simulations.

5. CONCLUSIONS

The simulations show that the 40 ternary templates of Hermite function were properly classified by the SOM into 20 distinct clusters. Learning phase took only 200 iterations, while retrievals were prompt and online. Although H(n) and H(n + 1/ - 1) shapes were similar, the created ternary templates were sensitive enough for the MOP to classify them into different classes. Setting the thresholding value was crucial in creating distinct templates. The use of ternary templates helped in fine-tuning cluster classes, while using real values directly requires another stage of "vector quantization" [8]. Testing same classifier with noisy reference signals showed robustness of that classifier; this is due to the ability of pattern generator network to tolerate abrupt changes in the reference signal values. The pattern generator network transforms the reference signal into a brief primitive message enough to reflect the class of the signal and disregarding to some extent unnecessary details. The SOM network at the last stage reacts to the ternary pattern signal in parallel settling to a single neuron out of the 20 possible neurons in the map. Each neuron in the SOM is supposed to represent a class.

Learning continues until each artificial ECG (Hermite) is associated with a single neuron. Online retrievals with noisy signal showed excellent classification rate. After these encouraging results, future work includes testing the robustness of our classifier with noisy real-life ECG's.

REFERENCES

[1] D.E.Golclhcrg,"Genetic: Algorithms in Search Optimization, and Machine Learning", Reading, MA: Addison-Wesley, 1989.

[2] Z.Frankicwicz and A.Shrouf,"ECG Beat Classification Using Linear Prediction Error Signal," Proceedings of Medical Informatics, Vienna, Austria,1991.

[3] B.Eiscnstein and R.Vac:caro,"Feature Extraction by System Identification," IEEE Trans. on Systems, Man, and Cybernetirs, 12(1):42-50,1982.

[4] RA.Alm Zitar, M.H.Hassoun,"Neuroontrollers trained with rules extracted by a genetic assisted reinforcement learning system", IEEE Trans. on Neural Networks, 6(4)(1995), 859-879.

[5] L.Sornmo, P.Bmjesson, M.Nygards, O.Pahlrn," A Method for Evaluation of QRS Shape Features Using a Mathematical Model for ECG" ,IEEE Trans. on Biomedical Engineering, 28 (10),p713-717, 1981.

[6] K.Lin and W.Chang,"QRS Feature Extraction Using Linear Prediction," IEEE Trans. on Biomedical Engineering, 36 (10):1050-1055, 1989.

[7] T.Kohonen," Self-organizing maps: Optimization approaches, in Artificial Neural Networks,T. Kol10-nen, K.Makisara, O.Simula, and J.Kanga, eds., pp. 1147-1156. IEEE, New York, 1993.

[8] Y. Hu, S. Palreclcly, and W. J. Tompkins," A Patient-Adaptable ECG Beat Classifier Using a Mixture of Experts Approach," IEEE Trans. on Biomedical Engineering, 44(9): 891-900, 1997.

[9] M. Javadi, S. A. A. A. Arani, A. Sajedin, and R. Ebrahimpour, “Classification of ECG arrhythmia by a modular neural network based on Mixture of Experts and Negatively Correlated Learning,” Biomed. Signal Process. Control, 8(3):289–296, 2013.

[10] R. Oweis and B. Al-Tabbaa, “QRS Detection and Heart Rate Variability Analysis: A Survey,” Biomed. Sci. Eng., 2(1):3–34, 2014.

[11] C.-C. Chiu, T.-H. Lin, and B.-Y. Liau, “Using Correlation Coefficient in ECG Waveform for Arrhythmia Detection,” Biomed. Eng. Appl. Basis Commun., 17:147–152, 2005.

[12] A. Al-shrouf, “Noise-Immune ECG Classifier Using Wavelet Transform and Neural Networks,” Int. J. Eng. Adv. Technol., 5(1), pp. 87–92, 2015.

[13] M. K. Das and S. Ari, “ECG Beats Classification Using Mixture of Features,” Int. Sch. Research Not., 2014:1–12, 2014.

[14] Y. Xia, Q. Gao, and Q. Ye, “Classification of gait rhythm signals between patients with neurodegenerative diseases and normal subjects: Experiments with statistical features and different classification models,” Biomed. Signal Process. Control, 18:254–262, 2015.

[15] V. D. Awandekar, R. Mane, A. N. Cheeran, and P. Rani, “Cardiac Arrhythmia Detection By ECG Feature Extraction,” 3(2)327–332, 2013.