

Military Technical College
Kobry Elkobbah,
Cairo, Egypt



2nd International Conference
on Electrical Engineering
ICEENG 99

A CARDIOLOGY APPLICATION USING AN ARTIFICIAL NEURAL NETWORK MODEL

M. Shokr* , M. Shaarawy** , and H. Abo Elsoud***.

Egyptian Armed Forces

ABSTRACT

In this paper, a Neural Network Model is designed for the classification of normal and abnormal electrocardiography (ECG) signals. Linear Prediction Coding (LPC) is used for extracting the features of the signals generated from each patient. The features of the signals are applied as inputs to train and test the Neural Network. Different Neural Network architectures investigated in order to achieve a better performance. Test results show that, the classification accuracy of the network can reach 98 %.

KEY WORDS

Artificial Neural Networks (ANN), Electrocardiograph (ECG), Back Propagation Network (BPN), Linear Prediction Coding (LPC).

1. INTRODUCTION

The use of Neural Network systems in ECG signal analysis offers several advantages over conventional techniques. First, a Neural Network can perform the necessary transformation and clustering operations automatically and simultaneously. Second, a Neural Network is able to recognize complex and nonlinear groups in the hyperspace. This is a distinct advantage over many conventional techniques. Third, a Neural Network is massively parallel in nature and can easily operate in real-time scenarios [1].

Neural Networks models have been used in the domain of cardiology such as myocardial infraction prediction [2], [3], ECG classification [4], [5], detection and recognition of abnormal electrocardiograms [6], [7], model and diagnosis cardiovascular systems [8], training of ECG signals [9], and analysis of serial cardiac enzyme data [10], [11].

In this paper, we study the effect of changing the architecture of a proposed Neural Network model applied to cardiology. There are two main stages: feature extraction and classification. In the first stage, Linear Prediction Coding (LPC) is used for feature extraction where its linear predictive coefficients act as features for the

* Dr., Dept. of Computer & Operation Research, Military Technical Collage.

** Ass. Prof. Dr., Dept. of Computer & Operation Research, Military Technical Collage.

*** Eng., Egyptian Armed Forces.

measured ECG signals [12]. In the second stage, the Back Propagation Neural Network (BPN) is used as a tool to learn the cases, and based upon the extracted linear prediction coefficients it can classify the type of the ECG's signals.

Linear Prediction Coding (LPC) has been successfully used by others for ECG feature extraction [12], QRS feature extraction [13], Computer based feature extraction and diagnosis of special electronic signals [14], speech recognition [15], and myoelectric signal analysis [16]. The use of LPC coefficients increases the discrimination capability of the Neural Network classifier without drastically increasing the Neural Network architecture complexity [1].

The feedforward, Back Propagation Network (BPN) is the most widely network used for classification. It is the most popular, effective, and easy to learn model for complex, multilayered networks. This network is used more than all other combined. It is used in many different types of applications. It has spawned a large class of network types with many different topologies and training methods. Its greatest strength is in non-linear solutions to ill-defined problems [17].

2. THE APPLICATION PROBLEM

2.1 Nature

The ECG is an important tool for providing information about the heart. It can provide evidence to support a diagnosis, and therefore management, of abnormal cardiac rhythms. Conventional ECG diagnosis depends on the experience of a doctor, therefore, it may suffer from errors. A solution for this problem is to use a Neural Network system to implement the diagnosis process. This helps specialists to reach their final diagnosis decision in a faster, easier, and more accurate way.

2.2 Data Collection

The ECG, however, is a nonlinear signal generated from the human body. The electrical signal from the heart is detected as the surface of the body through five electrodes, which are joined to the ECG recorder by wires. One electrode is attached to each limb and one is held by suction to the front of the chest and moved to different positions. Using a data acquisition card, a group of ECG signals for different patients have been collected and converted to digital signals, then applied as inputs to the proposed Neural Network model.

2.3 Input Measurements

The input of the network is the Linear Predictive Coding (LPC) coefficients resulted from the preprocessing of the collected ECG signals as mentioned in 3.1.

2.4 Output Decision

The proposed model has the ability to distinguish between normal and abnormal persons. Also, it classifies the type of the disease whether it is Myocardial Infraction, Hyper Kalenia, Digitalis Toxicity, or Angina Pectoris.

3. THE PROPOSED SYSTEM

Figure 1 illustrates the block diagram of the proposed system.

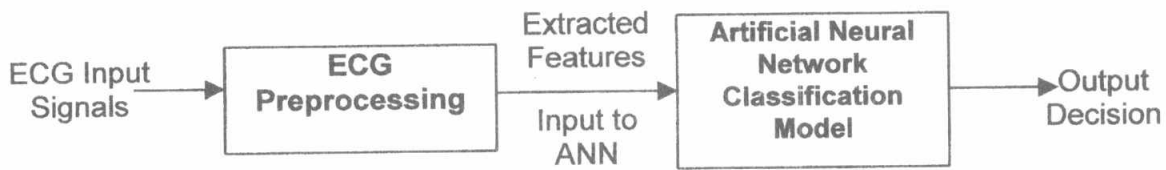


Fig. 1 : Block diagram of the proposed system.

3.1 ECG Preprocessing (Feature Extraction)

The main objective of the preprocessing is to extract a minimal set of parameters which adequately represents each ECG signal without sacrificing classification performance of the ANN classifier. Using a minimal set of ECG features allows for a reduction in the size of the ANN, which is desirable for real-time implementation of the classifier. These ECG feature parameters are used as training and test inputs to the ANN classification model [1].

The basic idea behind the linear prediction method in ECG signal analysis is that a sample ECG data can be approximated as a linear combination of past ECG samples. The structure of this concept is illustrated in figure 2.

The actual ECG sequence $s(i)$ can be approximated by another sequence $\hat{s}(i)$ which is determined by a unique set of predictor coefficients and the past P samples $s(i)$ [13]. That is,

$$\hat{s}(i) = \sum_{k=1}^P a(k) * s(i - k) \quad (1)$$

where $a(k)$ is k th linear predictive coefficients (LPC). These coefficients are used as weighting factors in a linear combiner as shown in figure 2.

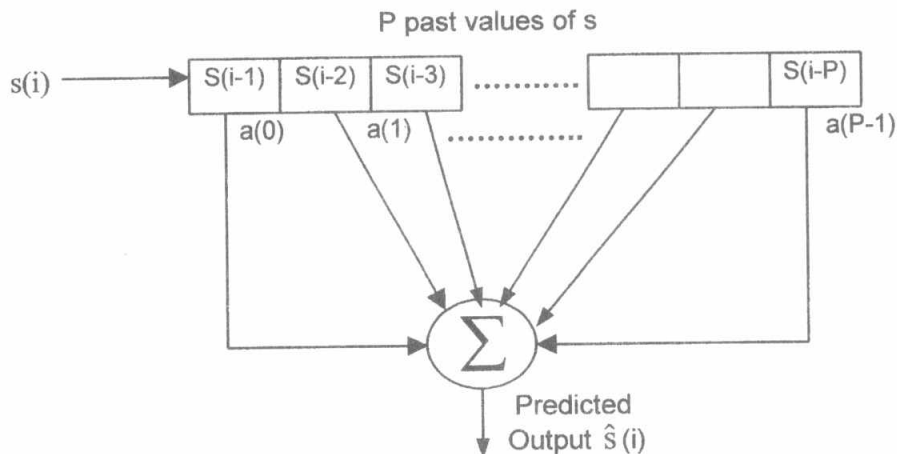


Fig. 2 : Pth order linear predictive mechanism

Table 1 shows the changes in the linear predictive coefficients sets for prediction order P for $P=5$ to $P=9$ coefficients. In figure 3, the ECG signal and its linear prediction coefficients for various order are compared.

Table 1 The results of linear prediction coefficients

P=5 LPC coefficients	P=6 LPC coefficients	P=7 LPC coefficients	P=8 LPC coefficients	P=9 LPC coefficients
a(0) = 1.0	a(0) = 1.0	a(0) = 1.0	a(0) = 1.0	a(0) = 1.0
a(1) = -0.8210	a(1) = -0.9509	a(1) = -0.9638	a(1) = -0.9630	a(1) = -0.9627
a(2) = -0.5341	a(2) = -0.4440	a(2) = -0.4185	a(2) = -0.4265	a(2) = -0.4263
a(3) = 0.2939	a(3) = 0.1950	a(3) = 0.1917	a(3) = 0.2055	a(3) = 0.2003
a(4) = -0.2677	a(4) = -0.0881	a(4) = -0.0805	a(4) = -0.0823	a(4) = -0.0734
a(5) = 0.3861	a(5) = 0.6623	a(5) = 0.6452	a(5) = 0.6494	a(5) = 0.6482
	a(6) = -0.3365	a(6) = -0.3731	a(6) = -0.3821	a(6) = -0.3792
		a(7) = 0.0385	a(7) = 0.0177	a(7) = 0.0119
			a(8) = 0.0215	a(8) = 0.0083
				a(9) = 0.0137

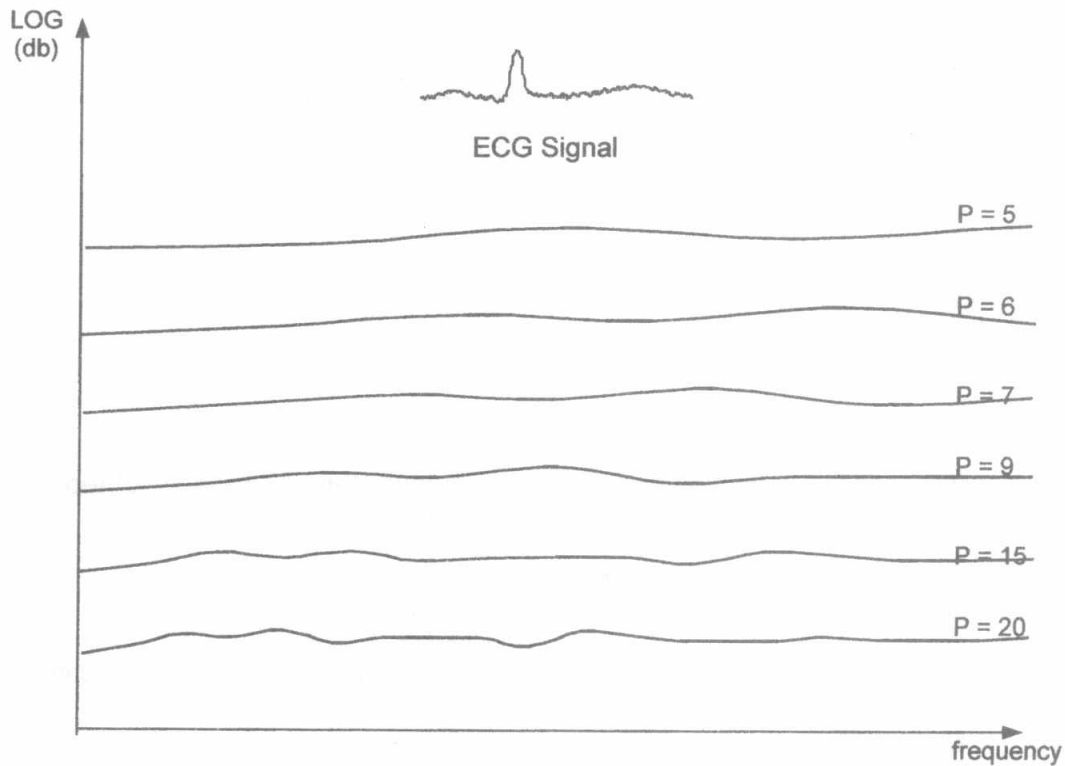


Fig. 3 : ECG sampled for several values of predictor order P

3.2 ANN Classification Model

A three-layer, feed-forward Back Propagation Neural Network model is created. The creation performed using Neural Works Professional II/PLUS format [18].

3.2.1 Network architecture

The architecture of the proposed network is achieved by determining the number of processing elements (PEs) in each layer, number of hidden layers, and the layer parameters (Learning Rule, Transfer Function, Noise Function, Summation Function, and Error Function) for each layer.

The weights associated to the interconnections among the neurons represent the implicit knowledge contained in the network. This knowledge is the result of training,

which is performed until reaching the desired response. High performance network is achieved, when it passes the test examples with acceptable accuracy.

3.2.2 Initial network architecture

A three-layer, feed-forward Back Propagation model is created, consisting of an input layer, one hidden layer, and an output layer with the number of PEs in the input layer=5, in the hidden layer=5, and in the output layer=5, as shown in figure 4. Each unit in the network is connected to all units in the layers above its own (fully interconnected). Each connection has a positive or negative weight associated with it. The layer parameters are as follows :-

3.2.2.1 Learning rule

Delta rule as a learning rule.

3.2.2.2 Summation function

Weighted sum, which is the traditional sum of the effective inputs .

$$\text{net} = \sum_{i=1}^n W_i X_i \quad (2)$$

where :

W_i : the weight.

X_i : the input.

net : the output of the summation function.

3.2.2.3 Transfer function

Each unit in the hidden layer and the output layer thresholds a weighted sum of its inputs to get an output by applying a Sigmoid function defined by :

$$f(\text{net}) = \frac{1}{(1 + e^{-k\text{net}})} \quad (3)$$

where k is a positive constant that controls the spread of the sigmoid function.

3.2.2.4 Noise function

Uniform distributed noise function is used as a noise function.

3.2.2.5 Error function

Standard error function is used. It does not perform any transformation for the error. It is defined by :

$$E = \sum_j T_j O_j \quad (4)$$

where :

O_j : the j^{th} component of the actual output vector.

T_j : the j^{th} component of the desired output vector

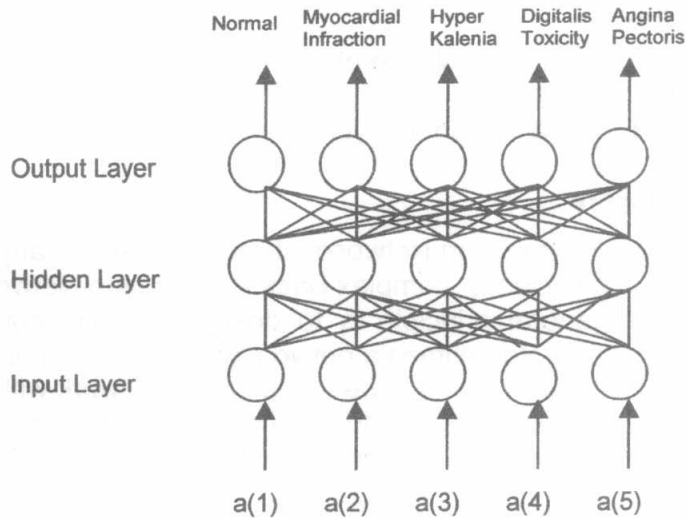


Fig. 4 : Initial model architecture

The initial architecture of the proposed model is changed, to reach suitable architectures that improves the performance, by changing :-

1. The number of processing elements in the input layer, i.e., changing the number of linear predictive coding (LPC) coefficients.
2. The number of hidden layers.
3. The number of processing elements in the hidden layers.
4. The layer parameters of the hidden layer, and the output layer.

3.2.3 Input layer design

The input layer of the network has a number of processing elements which is equal to the number of Linear Predictive Coding (LPC) coefficients. The LPC coefficients are the result of the preprocessing of the acquired ECG signals [12]. The suitable number of LPC coefficients has been chosen according to the minimum Root Mean Square error (RMS). LPC has been taken equal to 5, 6, 7, 9, 12, 15, and 20 coefficients respectively. Figure 5 shows networks with different number of LPC coefficients and their corresponding root mean square error. It has been noticed that, the error degraded gracefully as the number of LPC coefficients increases. It has been found that there is no remarkable error reduction after using 15 LPC coefficients. So, only 15 LPC coefficients have been chosen [12].

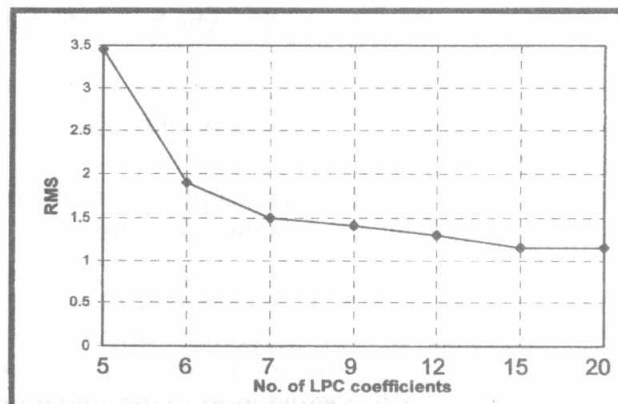


Figure 5 : The suitable number of LPC coefficients

3.2.4 Hidden layers design

The design of the hidden layer means the choice of :

- 1- Number of hidden layers
- 2- Number of PEs in each hidden layer

3.2.4.1 Number of hidden layers

Recent researches into Back Propagation Networks shows that almost any function can be synthesized using a sufficiently complex Back Propagation Network with a single hidden layer. A good general principle is to design the least complicated network that provides good results. Sometimes a network with two hidden layers and fewer total PEs can do just as well as a more complicated network with a single hidden layer. Having less PEs has some performance advantages, in particular, requiring less memory and providing faster convergence [18].

We design the proposed model for the number of hidden layers by using none hidden layers, one hidden layer, and two hidden layers, and test the RMS at each case as shown in figure 6. It was found that the RMS decreases when using one hidden layer, so, one hidden layer is the best choice for the number of hidden layers.

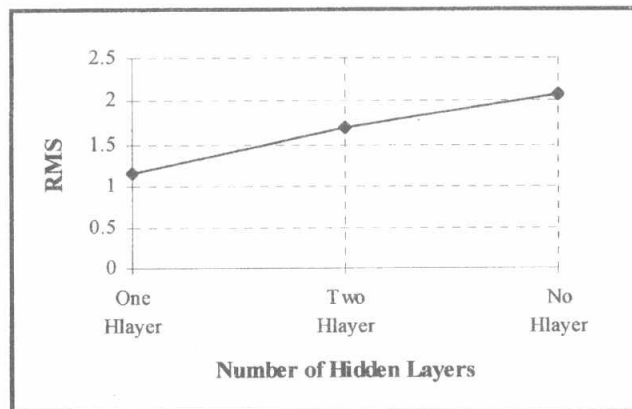


Figure 6 : The suitable number of hidden layers leading to a minimum RMS

3.2.4.2 Number of PEs on the hidden layer

The design of the number of processing elements in the hidden layer follows the following algorithm [19]:

- Step1** : Start with a number of neurons which is definitely too small .
- Step2** : Train and test the network
- Step3** : Record its performance
- Step4** : Then slightly increase the number of hidden neurons , and train and test again.
- Step5** : Repeat 1-4 until the error is acceptably small , or no significant improvement is noticed , whichever come first.

The number of neurons in the hidden layer has been chosen according to minimum root mean square error. We start with 6 neurons and increase this number until 15 neurons. Figure 7 shows different networks and their corresponding root mean square error. It was found that 7 PEs would lead to the minimum RMS error.

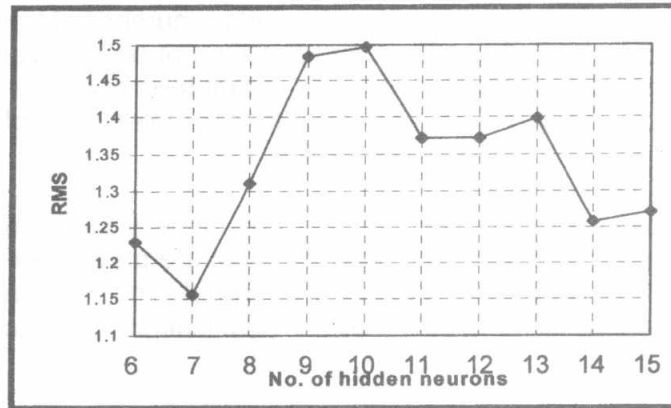


Figure 7 : The suitable number of hidden neurons leading to a minimum RMS

3.2.5 Output layer design

The last layer, output layer, has five PEs which corresponding to the five Output classes (Normal, Myocardial Infraction, Hyper Kalenia, Digitalis Toxicity, and Angina Pectoris).

3.2.6 Training technique

The aim of training an ANN is to adjust the weights and the bias. Training is performed by a training algorithm. There are two approaches to train an ANN Supervised Training, and Unsupervised Training.

The proposed model is a Back Propagation model that uses a supervised training technique. In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network.

3.2.7 Training examples

There are five groups of examples to train and test the network which correspond to the Normal, Myocardial Infraction, Hyper Kalenia, Digitalis Toxicity, and Angina Pectoris cases. The available number of training examples, eighty five, has been divided into fifty training examples, and thirty five test examples.

The train and test files have been written in PROFESSIONS II/PLUS format [18], each row in the file consists of a number of LP coefficients, forms the input vector, paired with the desired target vector. The input vector is separated from the target vector by the "&" sign. The number of LP coefficients that were tried are 5, 6, 7, 9, 12, 15, and 20 coefficients.

3.2.8 System performance

Neural network does not always give the exact result you desire. It is for that reason ANNs used in applications where humans are also unable to be always right [18]. In this study, the network performance is analyzed during both the training and test phase.

During the training phase, the network needs a number of training cycle to meet the convergence criterion, this number gives an idea about Convergence Time (Number of Cycles). Low convergence time means that the network is trained over the set of examples, and it understands the trained examples completely up to the desired error level.

The test phase is one way of determining how will the network has learned, and perform. During the test phase, the test cases are presented to the network and the network provides output. Based on this output and the known desired output, the Root mean Square error (RMS), and classification accuracy of the network are measured.

3.2.8.1 Root Mean Square error (RMS)

The root mean square error adds up the squares of the errors for each PE in the output layer, divides by the number of PEs in the output layer to obtain an average, and then takes the square root of that average. Hence the name root mean square. The squaring of the errors get rid of the sign of the error, but increase the magnitude. The square root removes the increase of the magnitude resulting from the squaring operation, leaving the absolute value.

$$RMS = \sum_{i=1}^P \sqrt{\frac{\sum_{j=1}^N (T_j - O_j)^2}{N}} \quad (5)$$

Where :

- O_j : the j^{th} component of the actual output vector.
- T_j : the j^{th} component of the desired output vector.
- N : number of PEs in the output layer.
- P : number of patients (test examples).

3.2.8.2 Classification accuracy

Reject a true / Accept a false (type I / type II) errors are one of the common aspects for any system. Type I error represents False Reject Rate (FRR) which is the error rate when the system rejects a true case, while type II error represents False Accept Rate (FAR) which is the error rate when the system accepts a false case [20].

In this paper, the classification accuracy of the system represents the true rate when the system accepts a true case, i.e., for a number of patients equal to P the FRR, FAR, and accuracy are defined by :

$$FRR = \frac{FR}{P} \quad (6)$$

$$FAR = \frac{FA}{P} \quad (7)$$

$$Accuracy = [1 - (FRR + FAR)] * 100 \% \quad (8)$$

Where :

- FR : the number of false reject patients.
- FA : the number of false accept patients.

3.2.9 Layer parameters

There are five major components, which make up an artificial neuron, Learning Rule, Transfer Function, Noise Function, Summation Function, and Error Function. These components are valid whether the neuron is used for input, output, or is in one of the hidden layers. In the initial proposed model the layer parameters are as follows :- Learning Rule : Delta Rule, Transfer Function : Sigmoid, Summation Function : Sum, Error Function : Standard, and Noise Function : Uniform.

3.2.10 Changing layer parameters of the proposed model:

The purpose of changing the layer parameters is to reach a suitable architecture that improves the performance (higher Accuracy, smaller Root Mean Square error, and less Convergence Time). Changing layer parameters is achieved by changing learning rule, transfer function, summation function, noise function, and error function.

3.2.10.1 Learning rule

The Learning Rule specifies how connection weights are changed during the learning process. The purpose of the learning rule is to modify the variable connection weights on the inputs of each processing element (PE) according to some algorithms. For the Back Propagation Neural Network there are three learning rules :- the Delta Rule, the Cumulative Delta Rule, and the Normalized Cumulative Delta Rule

3.2.10.2 Summation function

The first step in a processing element's operation is to compute the weighted sum of all inputs. Available summation functions are :- Weighted sum, and Cumulative Summation.

3.2.10.3 Transfer function

The transfer function is a non-linear function that transfers the internally generated sum of each processing element to a potential value. The result of the summation function is transformed to a working output through an algorithmic process known as a transfer function. Available transfer functions are :- Sigmoid, and Tangent Hyperbolic (TanH).

3.2.10.4 Noise function

The noise function is normally referred to as "temperature" of the artificial neurons. Available noise functions are :- Uniform, Gaussian, and None (Noise=0).

3.2.10.5 Error function

In most learning networks the difference between the current output and the desired output is calculated. This raw error is then transformed by the error function to match a particular network architecture. Available error functions are :- Standard, Quadratic, Cubic, and Tolerant.

Table 2 represents some of the examined models and their corresponding layer parameters. The examination process starts from an initial model referred to as

Model 1. By changing the layer parameters other models are obtained. Table 2 contains 32 of these models.

Table 2 : Some Models and their corresponding layer parameters

Network	Layer Parameters				
	Learning Rule	Transfer Function	Noise Function	Error Function	Sum. Function
Model 1	Delta	Sigmoid	Uniform	Standard	Sum
Model 2	Cum	Sigmoid	Uniform	Standard	Sum
Model 3	Norm	Sigmoid	Uniform	Standard	Sum
Model 4	Delta	TanH	Uniform	Standard	Sum
Model 5	Cum	TanH	Uniform	Standard	Sum
Model 6	Norm	TanH	Uniform	Standard	Sum
Model 7	Delta	Sigmoid	Gaussian	Standard	Sum
Model 8	Delta	Sigmoid	None	Standard	Sum
Model 9	Cum	Sigmoid	Gaussian	Standard	Sum
Model 10	Cum	Sigmoid	None	Standard	Sum
Model 11	Norm	Sigmoid	Gaussian	Standard	Sum
Model 12	Norm	Sigmoid	None	Standard	Sum
Model 13	Delta	TanH	Gaussian	Standard	Sum
Model 14	Delta	TanH	None	Standard	Sum
Model 15	Cum	TanH	Gaussian	Standard	Sum
Model 16	Cum	TanH	None	Standard	Sum
Model 17	Norm	TanH	Gaussian	Standard	Sum
Model 18	Norm	TanH	None	Standard	Sum
Model 19	Delta	Sigmoid	Uniform	Quadratic	Sum
Model 20	Delta	Sigmoid	Uniform	Cubic	Sum
Model 21	Delta	Sigmoid	Uniform	Tolerant	Sum
Model 22	Cum	Sigmoid	Uniform	Quadratic	Sum
Model 23	Cum	Sigmoid	Uniform	Cubic	Sum
Model 24	Cum	Sigmoid	Uniform	Tolerant	Sum
Model 25	Norm	Sigmoid	Uniform	Quadratic	Sum
Model 26	Norm	Sigmoid	Uniform	Cubic	Sum
Model 27	Norm	Sigmoid	Uniform	Tolerant	Sum
Model 28	Delta	TanH	Uniform	Quadratic	Sum
Model 29	Delta	Sigmoid	Gaussian	Quadratic	Sum
Model 30	Delta	TanH	None	Cubic	Sum
Model 31	Cum	TanH	Gaussian	Cubic	Sum
Model 32	Delta	Sigmoid	Uniform	Standard	Cum Sum

4. EXPERIMENTAL RESULTS

The experimental results presented in this study, for classification of normal and abnormal ECG signals, were based on data for eighty five patients. Linear predictive coding (LPC) coefficients extracted from each ECG signal, are utilized to train and test the proposed neural network classifier. The network was written in the PROFESSIONS II/PLUS format [18], and implemented on a PC Pentium 233 MHz Computer.

A three-layer, feed-forward Back Propagation model was created. It has an initial architecture consisting of an input layer, one hidden layer, and an output layer. The number of PEs in the input layer=5, the number of PEs in the hidden layer=5, and the number of PEs in the output layer=5. The layer parameters are :- Delta rule as a learning rule, Sigmoid as a transfer function, Weighted sum as a summation function, Uniform as a noise function, and Standard as an error function.

The performance of the network was evaluated by three different parameters: accuracy, root mean square error (RMS), and number of cycles (convergence time). Based on try and error technique, the architecture of the initial model has been changed to reach a suitable architecture that improves the performance. It has been noticed that :-

1. The RMS error degraded gracefully as the number of LPC coefficients increases. There is no remarkable error reduction after using 15 LPC coefficients. So, only 15 LPC coefficients have been chosen.
2. The RMS error decreases when using one hidden layer, so, one hidden layer is the best choice for the number of hidden layers.
3. The number of PEs in the hidden layer leads to a minimum RMS error is equal to 7 PEs.

The results of changing the layer parameters are summarized in table 3.

Table 3 : Some Models and their Performance

Network	RMS	Accuracy	No. of Cycles
Model 1	1.155864	98 %	70473
Model 2	1.206985	98 %	134674
Model 3	2.018764	94 %	490972
Model 4	1.693075	96 %	19323
Model 5	1.381727	98 %	65565
Model 6	1.633462	94 %	188526
Model 7	1.173183	98 %	52537
Model 8	1.155864	98 %	70311
Model 9	1.213655	98 %	185311
Model 10	1.260578	96 %	172341
Model 11	1.840695	92 %	660069
Model 12	1.431283	96 %	640375
Model 13	1.772098	94 %	13534
Model 14	2.093762	94 %	20560
Model 15	1.644504	92 %	53406
Model 16	1.565251	94 %	33320
Model 17	1.244459	98 %	173031
Model 18	1.412616	94 %	200311
Model 19	1.636765	94 %	25875
Model 20	2.28209	94 %	25795
Model 21	1.189504	98 %	56031
Model 22	1.735934	94 %	55350
Model 23	2.30464	92 %	62058
Model 24	1.148306	98 %	186042
Model 25	2.00747	94 %	188432
Model 26	2.590554	94 %	188406

Model 27	2.019781	94 %	521309
Model 28	1.529237	96 %	8712
Model 29	1.538362	98 %	21820
Model 30	1.89918	98 %	9056
Model 31	1.983665	98 %	16144
Model 32	-	-	No Convergence

Figure 8 shows that, the layer parameters of Models 1, 2, 5, 7, 8, 9, 10, 17, 21, 24, 29, 30 and 31 lead to a higher accuracy.

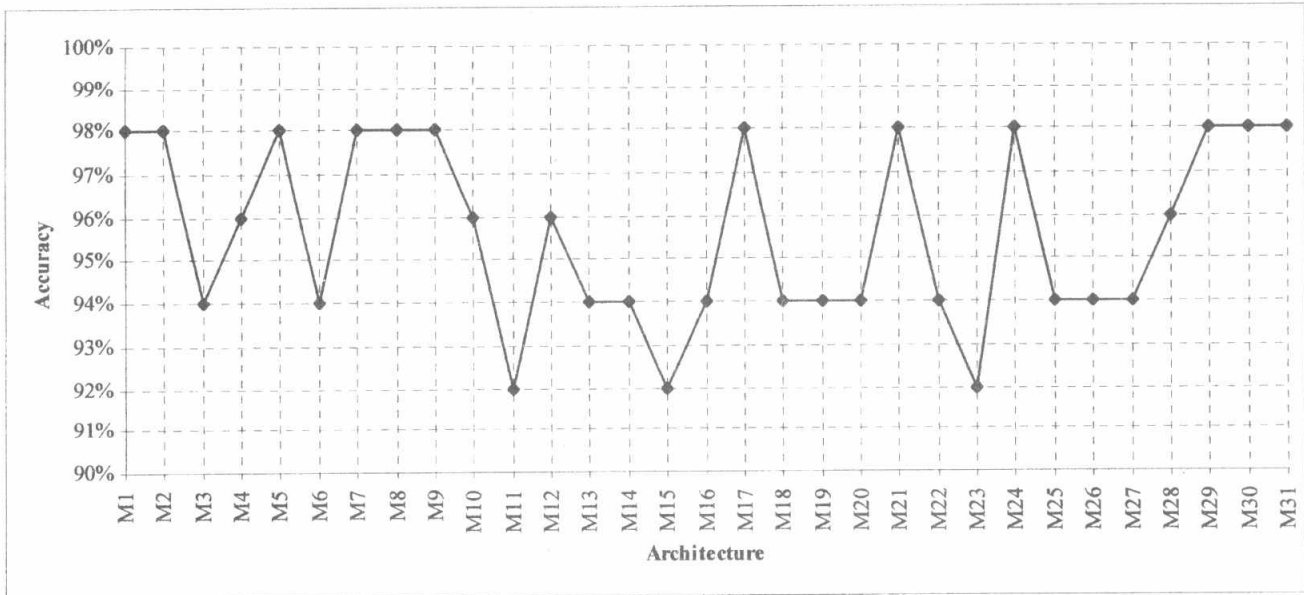


Fig. 8 : Effect of changing Architecture on Accuracy

Figure 9 shows that, the layer parameters of Model 28, and 30 lead to a less convergence time (number of cycles).

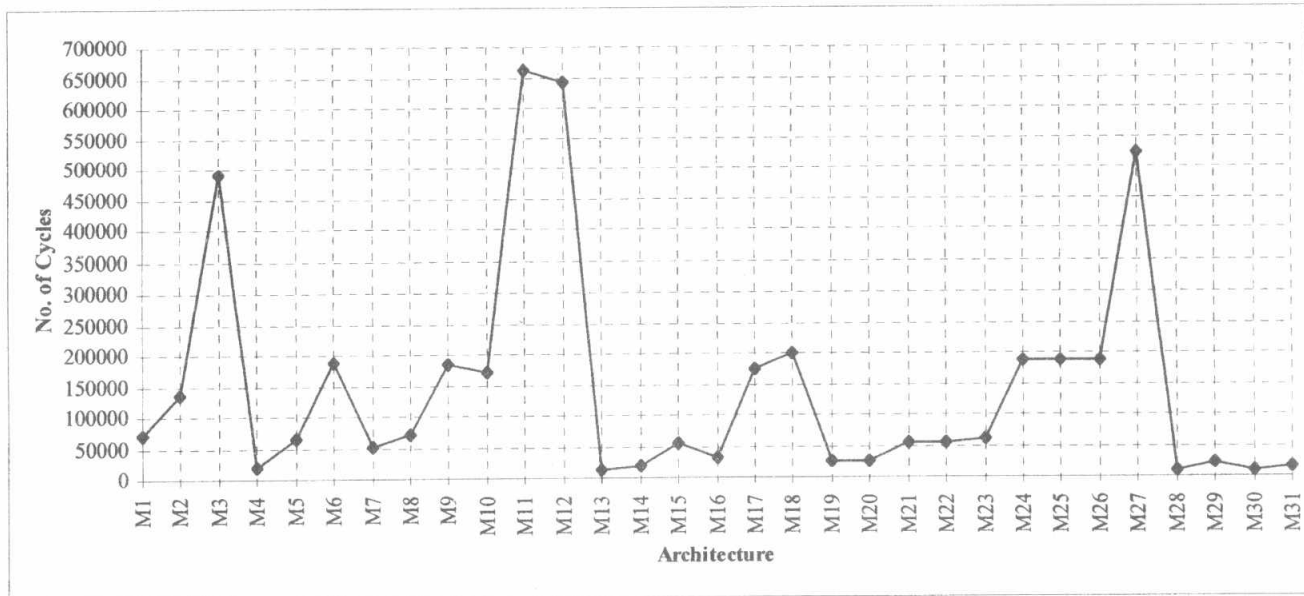


Fig. 9 : Effect of changing Architecture on No. of Cycles

Figure 10 shows that, the layer parameters of Models 1, 7, 8, 21, and 24 lead to a less RMS.

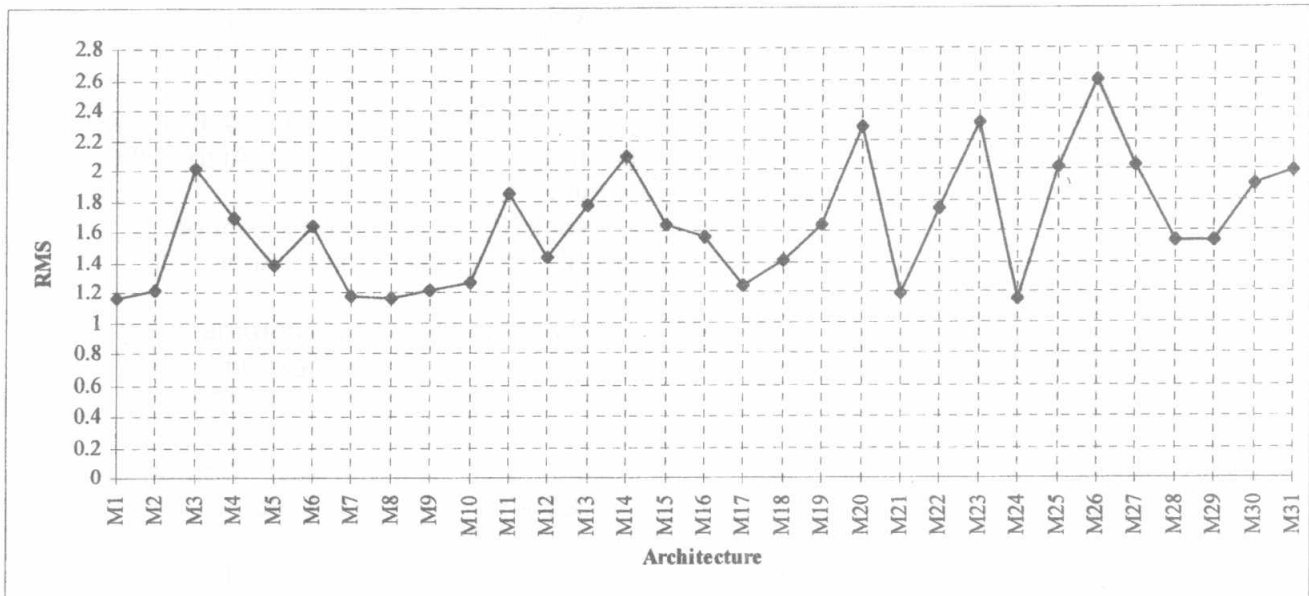


Fig. 10 : Effect of changing Architecture on RMS

5. CONCLUSION

A reliable classification methodology to distinguish between normal and abnormal ECG signals, based on neural network architecture, is presented in this study. The features that were used for the classification task reduce the size of the neural network classifier with acceptable discrimination ability for possible real-time ECG diagnosis. The architecture of the designed cardiovascular neural network is changed to improve its performance, it was noticed that the total performance (Accuracy, RMS, Convergence time) of the network is better when using :-

- 1-Fifteen linear predictive coding (LPC) coefficients to represent the number of processing elements in the input layer.
- 2-The number of hidden layers is equal to one hidden layer.
- 3-The number of processing elements in the hidden layer is equal to 7 PEs.
- 4-The layer parameters, Delta Rule as a learning rule, Sigmoid as a transfer function, Weighted sum as a summation function, Gaussian as a noise function, and Quadratic or Cubic as an error function.

For the future work, There is still an extra work that can be done in the developed system, by acquiring more extra ordinary examples from patients, train the network over it to increase its performance, and adding new cases to the system to increase its capability.

REFERENCES

- [1] Fredric M. Ham., Soowhan Han., "Classification of Cardiac Arrhythmias Using Fuzzy ARTMAP". IEEE Transactions on Biomedical Eng., Vol. 43, No. 4, April 1996.
- [2] Baxt, W.G., "Use of Artificial Neural Networks for the diagnose of myocardial Infraction." Ann. Inter. Med., 115: 843-848, 199.

- [3] Browner, W. S., "myocardial infraction prediction by Artificial Neural Networks." Ann. Inter. Med., 116(8): 701-2; discussion 702, 1992.
- [4] Bortolan, G., Degani, R., Willems, J.L., "Design of Neural Networks for classification of electrocardiography (ECG) signals." Proceed. 12th Ann, Conf., IEEE Engineer. Med. Biol. Soc., Philadelphia, PA, USA, 1990.
- [5] Bortolan, G., Degani, R., Willems, J.L., "ECG classification with Neural Networks and cluster analysis." In: Computers in cardiology 1991, Murray, A.; Arzbaecher, R. (Ed.), IEEE Computer Society, Los Alamitos, CA, 177-180, 1991.
- [6] Cheung, J. Y; Hull Jr., S.S., Yeo, C.Y.S.; Kohli, P. "Recognition of Abnormal ECG Signals By a Neural Network Approach" . Int. J. Microcomp. Appl., 10(2) : 48 - 53, 1991.
- [7] Cheung, J. Y; Hull Jr., S.S., "Detection of abnormal electrocardiograms using a Neural Network Approach" Proceed. Ann. Int. Conf. IEEE Engineering in Medicine and Biology Society, Vol. 11, Part 6/6, pp. 2015-2016 Nov., 1989.
- [8] Kangas - Lars - J, Keller - Paul - E, Allan - Paul - A. "Artificial Neural Networks (ANNs) to model and diagnosis cardiovascular systems." Northcon - Conference Record. IEEE, Piscataway, NJ, USA, 95CB35792. p 78 -83. 1995.
- [9] Xue, Q.; Hen, Y.; Tompkins, W. J. "Training of ECG Signals in Neural Network Pattern Recognition." Proceed. 12th. Ann. Conf. IEEE Eng. Med. Biol. Soc., Philadelphia, PA, USA, 1990.
- [10] Furling, James W., Milton E. Dupuy, James A. Heinsimer, "Neural Networks analysis of serial cardiac enzyme data." Am J Clin Pathol 96(1): 134 -141, 1991.
- [11] Furlong, J. W., Dupuy, M. E., Heinsimer, J. A. "Neural Networks analysis of serial cardiac enzyme data." A clinical application of artificial machine intelligence. Am J Clin Pathol 96(1): 134 -141, 1991.
- [12] E. Pietka, "Feature Extraction in Computerized Approach to the ECG Analysis". Pattern Recognition. Vol. 24, pp. 139-146, 1991.
- [13] Kang-Ping Lin and Walter H. Chang "QRS Feature Extraction Using Linear Prediction", IEEE Transaction On Biomedical Engineering, Vol. 36, No. 10, Oct. 1989.
- [14] M. A. Farghaly, "Computer Based feature Extraction and Diagnosis of Special Electronic Signals", Ph D. Thesis, The Military Technical College, Cairo, Egypt, Oct., 1991.
- [15] C. Shim, B. Espinoza-Varas, and J. Y. Cheung. "A PC Based Neural Network for Recognition of Different Syllables Using LPC Coefficients Difference", in Proc. IJCNN, Vol. II, pp.185-190, June, 1990.
- [16] M. F. Kelly, P. A. Parker, and R. N. Scott, "The Application of Neural Network to Myoelectric Signal Analysis : A Preliminary Study", IEEE Transaction On Biomedical Engineering, Vol. 37, No. 3, pp. 221-230, Mar. 1990.
- [17] Anil K. Jain, Jianchang Maco, and K. M. Mohiuddin. "Artificial Neural Networks (ANNs) : A Tutorial" . pp. 31 - 44 . March 1996.
- [18] Neural Ware Neural Computing, authored by Neural Ware, Inc. employees for their Neural Works Professional II/PLUS ANN Development Software, Pittsburgh, PA, 1991.
- [19] Antognetti, P. and Milutinovic, V., "Neural Networks : Concepts, Applications, and Implementations (Eds.)". Volumes I-IV, Prentice Hall, Englewood Cliffs, NJ., 1991.
- [20] A. Salem, M. Shaarawy, and M. Elfeky., "Handwritten Signature Verification Systems", Proc. Int.