# Use of Self Organizing Map to Obtain ECG Data Templates for Its Compression and Reconstruction

Akhil Ranjan Garg, PhD Dept. of Electrical Engg. , JNVU, Jodhpur (Raj) Mridul Kumar Mathur, PhD Dept of Comp. Sc., LMCST, Jodhpur (Raj.) Megha Singh I.I.T., Jodhpur

# ABSTRACT

An electrocardiogram (ECG) is a recording of electrical impulses generated by the electrical activity of the heart and is used as a diagnostic tool to analyze various heart diseases. For economical storage and fast transmission over low-bandwidth channels, ECG data need to be compressed. For the efficient compression of ECG signals, the topology preservation feature of self-organizing maps (SOM) is used. It is observed that a compression ratio up to 1:20 can be achieved with a very low-percentage root-mean-square difference, i.e. below 1.6, by creating templates of ECG patterns in the form of weight vectors of neurons. The templates obtained in this manner are then used to reconstruct the ECG signal. This analysis shows that the reconstructed signal is perfectly matched to the original signal.

# **Keywords**

Self-organizing maps; Compression; Winning neuron; Neighborhood function; Templates

# 1. INTRODUCTION

The electrical potential generated by the heart can be recorded by applying electrodes to various locations on the body surface and then connecting them to a recording apparatus [1, 2]. This recorded signal is called an electrocardiogram (ECG) and is often used as an important diagnostic tool to ascertain the functioning of a heart. There are situations in which ECG data are recorded at one location and analysed at a different location. Further, there is an increasing need to store the recorded ECG signal for future use. Therefore, there is a requirement for efficient and economical use of memory or the transmitting channel, which can be accomplished by compressing the ECG signal.

In this work, the utility of a self-organising map (SOM) for the compression and visualisation of ECG signals is explored. An SOM is a special class of artificial neural networks which has the ability to extract the most important statistical characteristics of the input space [3, 4, 5]. One of the biggest advantages of an SOM is that it can find a set of prototype/templates which are sufficient to represent the entire original input space. Using these characteristics of an SOM, it is shown that ECG data can be successfully compressed [5, 6]. As a first step, different patterns are constructed from the original ECG signal to compress the ECG signal. These patterns are then used to train the SOM neural network. After completion of the training process, it is found that the neurons of the network become tuned in such a way that their weight vectors are the representative prototype/templates of the patterns used to train the network. These prototypes contain the most informative content of the ECG signal, and the redundancy present in the signal is removed. The reconstructed ECG signal using these templates demonstrates that a higher CR and low values of the PRD can

be obtained by proper selection of the various parameters used for training the SOM.

# 2. MATERIAL AND METHODS

# 2.1 Source of the ECG Signal

The ECG signal used in this work is two-lead ECG data taken from the MIT-BIH Arrhythmia Database [7]. A single patient record has been used, which is digitised at 360 samples per second per channel with 11-bit resolution over the 10-mV range.

# 2.2 Data Pre-Processing

To remove 50 Hz power-line interference noise and noisy portion of the signal, the data have been pre-processed by passing through stages containing a low-pass filter and high-pass filter devised by Pan and Tompkins [8, 9]. Thereafter, the filtering process, the filtered data are normalised for compatibility with further stages of analysis. A linear normalisation method is used and each normalised value of the filtered ECG signal is obtained as follows:

$$x_{normalised} = \left(x_{original} - x_{\min}\right) / \left(x_{\max} - x_{\min}\right)_{(1)}$$

where  $x_{original}$  is the original value of the input data, and

 $x_{\min}$  and  $x_{\max}$  are the minimum and maximum values in

the complete ECG signal, respectively.  $X_{normalised}$  will contain values in the range from 0 to 1. Thereafter, this normalised signal is used to construct input patterns for the SOM.

# 2.3 Formation of Input Patterns for Training

To extract similarity between input patterns, the input data are formed by allowing overlapping between them. This overlapping is obtained by a repeating signal in the input data with an offset. Various sets of input patterns for training can be formed in the following manner:

1st input pattern:

$$x(1) = [x(t), x(t+1), x(t+2), \dots, x(t+z)]$$
(2)

2nd input pattern:

$$x(2) = [x(t+n), x(t+n+1), \dots, x(t+n+z)]$$
(3)

and so on. Here, n is the size of the offset which is taken as unity to construct the patterns and to avoid loss of information. To determine the size of the window, we considered that the duration of the normal QRS is 0.06 to 0.10 s (60–100 ms).

### 2.4 Selection of the Window Size

The original data taken from MIT-BIH Arrhythmia Database have been sampled at 360 Hz, which means each sample takes 1/360 = 2.77 ms time to occur. Therefore, the number of samples needed for a complete QRS region should vary between  $0.06 \times 360 = 21$  to  $0.10 \times 360 = 36$ , i.e. the window size should lie between 21 to 36 to include the complete QRS region in an input pattern. In this work, training and analysis are performed with window sizes of 8, 16, 32, and 64.

# 2.5 Formation of Input Patterns for Testing

In this phase, the value of n is taken to be equal to the window size. Depending on the window size Z, the test signal is broken into 'P' input vectors, each consisting of Z elements.

The input patterns for testing can be formed using following mathematical expressions:

1st input pattern

$$x(1) = [x(t), x(t+1), x(t+2), \dots, x(t+z)]$$
(4)

2nd Input pattern:

 $x(2) = [x(t+Z), x(t+Z+1), x(t+Z+2), \dots, x(t+2z)]$ (5)

and so on.

The Z-dimensional test data are given to the trained network of neurons with the same number of elements in their weight vectors.

# 3. MAP FORMATION

### **3.1 SOM Algorithm**

The network in this work is a two-layered structure in which one layer is an input layer having a single dimension, and the other layer is the output layer having two dimensions. The input layer is activated from the input data, expressed as

$$x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$$
(6)

where n is its dimension equal to the window size. The output layer consists of neurons or nodes arranged in a twodimensional lattice structure. These weight vectors are represented as

$$w_{j} = \left(w_{j1}, w_{j2}, ..., w_{jn}\right)^{T} \in \mathbb{R}^{n}$$
 (7)

where j is the coordinate of the neuron in the two-dimensional output space. The weight vectors of each neuron are initially assigned some random values from range [0, 1]. The steps for the formation of an SOM proposed by Kohonen [3, 4, 5] are as follows.

# Step 1: Determination of the winning neuron or best matching unit (BMU) and its location

Any input pattern picked up randomly from the input space of patterns to the network. Thereafter, the winning neuron or BMU is the neuron having minimum Euclidean distance (ED)  $d_i(x)$  is determined by

$$d_{j}(x) = \left\| x - w_{j} \right\| \qquad \text{for } j=1,\dots,L \tag{8}$$

where x is the input vector given by equation (6), and  $w_j$  is the weight vector of the j<sup>th</sup> neuron. If c(x) is used to represent the index of the BMU, then c(x) = j, where j is the index corresponding to the BMU.

#### Step 2: Determination of the amount of cooperation

As a part of the SOM learning mechanism, weight adaptation is utilised for the weights of the BMU and its neighbouring neurons. This property of the SOM is also called cooperation among neurons and helps in topological preservation of information among the patterns, i.e. nearby neurons tend to have a preference for a similar pattern type [4]. The amount of cooperation among neighbouring neurons is calculated using the distance-dependent function given by

$$e_{j,c(x)}(t) = \exp\left(\frac{-d_{j,c}^2}{2\sigma^2(t)}\right)$$
(9)

where t is the iteration number,  $d_{j,c(x)}$  given by ||c(x) - j||is the ED in the spatial domain between the c(x) location of the BMU and the j location of another neuron, and  $\sigma(t)$  is the effective width of the neighbourhood around the winning neuron and is also an iteration-dependent quantity.

## Step 3: Weight updating

The updated weights of the neurons are determined by following equation:

$$w_{j}(t+1) = w_{j}(t) + \eta(t)e_{j,c(x)}(t)(x(t) - w_{k}(t)) \quad (10)$$

The learning rate parameter  $\eta(n)$  is an iteration-dependent quantity in the map and is made to decay exponentially with the iteration.

In Step 4, the above steps are iterated until the stopping criterion is met. The iteration is stopped when  $\eta(n)$  is reduced to 0.01.

### **3.2 SOM Parameters**

When designing the SOM, the parameters which are of paramount importance are (i) the size of the map, (ii) the size of the input patterns or weight vector, (iii) the learning rate parameter, and (iv) the width of the neighbourhood function.

The size of the map is one of the most important parameters, as it determines how many neurons represent the input data space. Further, the size of the map configures the number of neurons for clustering different types of information contained in the input patterns. In this work, the different map sizes of 5x5, 7x7, 9x9, 11x11, 13x13, 15x15, 17x17, 19x19, and 21x21.

The dimension of the input data should be chosen in such a manner that each template in the map forms visually understandable data for that dimension. This dimension is equal to the window size.

The learning rate parameter defines how fast the network learns the input data. The capability of learning in the map depends on the learning rate parameter [3, 4, 5] given by

$$\eta(i) = \eta_0 \exp\left(-\frac{i}{\tau_2}\right) \tag{11}$$

As per the definition, the learning rate is a time-dependent quantity which also depends on the initial learning rate and the value of the time constant  $\tau_2$ . In the present work the initial value of the learning rate parameter is taken to be unity for the network to learn fast. The time constant  $\tau_2$  is chosen experimentally.

The effective width of the neighbourhood function [3, 4, 5] is given as:

$$\sigma(i) = \sigma_0 \exp\left(-\frac{i}{\tau_1}\right) \tag{12.1}$$

$$\tau_1 = k/\log(\sigma_0) \tag{12.2}$$

The rate at which the neighbourhood width decays exponentially relies on the value of  $\tau_1$  in equation (12.2), where  $\sigma_0$  the initial neighbourhood width, and K is is a constant. The neighbourhood width depends on its initial value, set to be equal to the radius of the map; the value of K is chosen experimentally.

# 3.3 Methodology to Test the Effectiveness of SOM in Compressing the ECG Signal

After the completion of training process the effectiveness of the SOM in compressing the ECG signal is tested using the procedure as mentioned below:

### Step 1: Storage of Weights/Templates in a File

The weights of each neuron of the SOM taken column wise are stored sequentially in a file named A. Thus there are N x N rows in the file with each row consisting of p elements, where p is the length of the weight vector.

# Step 2: Determination of Location of Winning Neuron for each Input Pattern:

Using the pre-processed ECG signal the input patterns are constructed with an appropriate window size using the method as described in section 2.5. These input patterns are stored in a file B in sequential manner. Each of these input patterns are taken one by one and applied as input to SOM. Out of N x N neurons for each of these input patterns one of the neuron will be the winning neuron. The location (r, c) of this winning is stored in file C in sequential manner (column wise), where r and c are the row and column number of winning neuron.

# Step 3: Reconstruction of ECG Signal Using the Data in File A and C

The location of winning neuron along with its representative template for each input pattern is stored in file C and A respectively. Using this information the new ECG signal is obtained by using the following algorithm.

- i. Initialise i = 1.
- ii. Read the values of the (i)<sup>th</sup> and (i+1)<sup>th</sup> element from file C and store them into variables r and c respectively.
- iii. Calculate s = [(r \* N) N + c], where N x N is the dimension of the map.

- iv. Copy the elements of the <sup>sth</sup> row from file A and store it in the file D column wise.
- v. Set i = i + 2.
- vi. Repeat Steps (ii–v) until the end of file C.
- vii. Exit

File D will contain the reconstructed ECG signal.

### **3.4 Performance Evaluation**

To examine the effect of varying parameters on the performance of the algorithm to map the input data and to authenticate the reconstruction of the original data from the map, PRDs[10] and CR [10] is utilized as performance measures. The definition of the PRD is given by following equation:

$$PRD = \sqrt{\frac{\sum_{n=1}^{N} \left(x[n] - \hat{x}[n]^{2}\right)}{\sum_{n=1}^{N} \left(x[n]^{2}\right)}} X100$$
(13)

where x[n] and x[n] are the original and reconstructed signals, respectively; and N its length.

The CR is the ratio of the original file size to the compressed file size, given as follows:

$$CR = \frac{\text{Original File Size}}{\text{Compressed File Size}}$$
(14)

For better reconstruction of the original signal, the values of the PRD should be very low and CR should be high. Many variations exist regarding the parameters used in the equation of the SOM algorithm. In this work, we show experimental results for variations in two important parameters of the algorithm, the width of the neighbourhood function and the learning rate parameter, which directly depend on the values

of K and  $\, au_2^{}$  , respectively.

# 4. EXPERIMENTAL RESULTS AND OBSERVATIONS

On the basis of the methods described in Section 3, the map is trained and tested for different window and map sizes, and the signal is then reconstructed to evaluate the performance of the proposed method. Before training, the suitable values of the map parameters explained in section 3.2 are selected. Their effect on map formation and calculation of the PRD will be analysed one by one.

# 4.0.1 Effect of the Time Constant $\tau_2$ , on the Learning Rate Parameter

As described in Section 3.2, the learning rate parameter is a time-dependent quantity which decays exponentially with time. To analyse the effect of  $\tau_2$ , on  $\eta(n)$ ,  $\tau_2$  was varied between 10,000 to 100,000 with the number of iterations fixed at 700,000. The iterative process was automatically terminated if the learning rate reached a value of 0.001. It was observed that 69,078 and 4, 83,543 number of iterations was needed for  $\eta$  to reach to 0.001 for  $\tau_2$  equals to 10000 and 100000 respectively.

It can be observed that a larger CPU time and a greater

number of iterations are needed for  $\eta$  to reach 0.001 as  $\tau_2$  increases. Therefore, the network learns at a fast rate for a

lower value of  $\tau_2$ , but the network requires more time to learn as  $\tau_2$  increases. However, we cannot choose a suitable value for  $\tau_2$  from this observation alone; therefore, we have used the

PRD to determine a suitable value for  $\tau_2$ , as described in next section.

# 4.0.2 Selection of the Suitable Values of $\tau_2$ and value of constant (K)

To determine the effects of  $\tau_2$  and K, a  $7 \times 7$  map was trained with 50,000 sampled ECG data. The map is tested with the same input for a window size of 32. For training the SOM, the input pattern was selected randomly from the input data space; however, the testing input patterns are fed into the SOM in a sequential manner. K was varied in the range of

8,000 to 13,000, and  $\tau_2$  was varied in the range of 10,000 to

100,000 to select suitable values of K and  $\tau_2$ . It is observed

that the lowest PRD of 2.5502 is obtained for K and  $\tau_2$  equal to 10,000 and 60,000, respectively. Therefore, these values for

K and  $\tau_2$  are considered to be the most suitable and were used for further analysis and experimentation.

### 4.1 Training of the SOM with an ECG Signal

For training the map after choosing the map and window size, the weight vector of each neuron was initialised. For initialising the weight vector, every element of the vector was assigned a randomly generated value in the range of 0 to 1. Figure 1 shows the initially obtained weight vectors for each neuron for a map size of  $7 \times 7$ . As can be seen from this figure, each weight-vector pattern does not contain any useful information, i.e. the patterns of the weight vectors are not structured.



Fig 1: Randomly initialised weight vectors in a 7 × 7 map obtained before training having window size of 32

The SOM is trained with the same set of input patterns as in Section 2.3 for fixed values of K = 10,000 and  $\tau_2 = 60,000$ until the leaning rate reaches 0.001. Figure 2 shows these values plotted using MATLAB. Figure 2 contains 49 subfigures, where each subfigure corresponds to the weight vectors after the completion of training for each neuron in a  $7 \times 7$  map with 32 window-sized input patterns. Each subfigure can be considered to be a representative template of some part of the ECG signal. From this figure, it can also be observed that the nearby patterns/templates are similar. This confirms the visualisation property of the SOM. The weight vectors of some neurons represent the QRS region, whereas other weight vectors represent the non-QRS portions including the isoelectric potential portions of the ECG signal, the P T waves



#### Fig 2: Weight vectors of neurons in a $7 \times 7$ mapobtained after training with input pattern having window s size of 32

#### 4.1.1 Effect of Map Sizes

To observe the role of the map size and to obtain a suitable map size, the SOM was trained for different map sizes of  $9 \times 9$  and  $11 \times 11$ . As the size of the map increases, the maximum number of neurons available for learning increase, which in turn allows the input data to map to a greater number of neurons. Figures 3 and 4 show the obtained maps for map sizes of  $9 \times 9$  and  $11 \times 11$  for a 32-point window size, respectively.



Fig 3: Weight vectors of neurons in a 9 × 9 map obtained after training with input pattern having window size of 32

	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	0.5	0.5	0.5	0.5 <u>0</u> 0 20 40	0.5	0.5	0.5	0.5	0.5	0.5	0.5
/ector	0.5 0 20 40	0.5	0.5	0.5 0 20 40	0.5	0.5	0.5	0.5	0.5 0 20 40	0.5	0.5
Weight \	0.5	0.5 0 20 40	0.5	0.5 1	0.5	0.5	0.5	0.5	0.5 0 20 40	0.5	0.5
ment of	0.5 0 20 40	0.5 1 10 20 40	0.5 0 20 40	0.5 0 20 40	0.5	0.5	0.5	0.5	0.5	0.5	0.5 0 20 40
each ele	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
ding to e	0.5	0.5	0.5	0.5 0 20 40	0.5 0 20 40	0.5	0.5	0.5	0.5	0.5	0.5
orrespon	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Value Co	0.5	0.5	0.5	0.5	0.5 0 20 40	0.5	0.5 0 20 40	0.5	0.5	0.5	0.5
-	0.5	0.5	0.5	0.5	0.5	0.5 20.40	0.5	0.5	0.5	0.5	0.5
	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Number of elements in each Neuron

### Fig 4: Weight vectors of neurons in a 11 × 11 map obtained after training with input pattern having window size of 32

From Figures 3 and 4, it can be observed that nearby neurons tend to have a similar structure for their weight vectors. An increase in the map size improves the distinguishing capability of the SOM for information. However, the increase in the map size does not always play a positive role, as the uniqueness in the templates degrades, while redundancy begins after a certain limit. This is due to the fact that input patterns with a slight difference do not map to the same neuron but to nearby neurons.

### 4.1.2 Effect of Window Size

In an ECG curve, more than 32 data points are required to represent a complete QRS complex. Therefore, we need to increase the window size to capture a complete waveform (P, QRS, or T wave) in the weight vectors of a single neuron. With a proper window size, better visualisation can be achieved.



Fig 5: Weight vectors of neurons in a  $9 \times 9$  map obtained after training with input pattern having window size of 16

	0.5 0 100	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Corresponding to each element of Weight Vector	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	0.5 0 100	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	0.5 0 50 100	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	0.5 0 50 100	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Value	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
			N	umber of e	elements i	in each Ne	euron		



Figures 5 and 6 show the weight vectors of neurons for a map size of  $9 \times 9$  with window sizes of 16, 64, respectively. From Figure 5, we cannot determine any portion of the ECG waveform with a window size of 16, whereas a portion of the ECG waveform having either a QRS or non-QRS region is visible in the subfigures in Figure 6 with a window size of 64. Therefore, the window size directly affects the visualisation property in terms the classification of the waveform.

# 4.2 Calculation of the PRD for Different Map and Window Sizes

*4.3.1 Testing on Untrained (Unknown) Data* For the same trained neurons of the map, unknown ECG data containing 600,000 samples (one complete ECG lead) from which the trained data were taken were tested for all map and window sizes. The number of input patterns P in the tested signal for various window sizes of 8, 16, 32, and 64 are 600,000/8 = 75,000; 600,000/16 = 37,500; 600,000/32 = 18,750; and 600,000/64 = 9,375, respectively. Again, K and  $\tau_2$  are fixed at 10,000 and 60,000, respectively. The values of the PRD for the reconstructed signal obtained from these tested signals for different map sizes and window sizes are listed in Table 1. From the table 1, it can be seen that the lowest PRD of 1.2600 is achieved for a map size of  $21 \times 21$  and window size of 8, this can be termed as best case whereas the highest PRD of 4.5237 is achieved for a map size of 5x5 and window size of 64; this can be termed as worst case.

Map size	Window size							
		8	16	32	64			
5×5	PRD	2.4113	3.0339	3.5548	4.5237			
	CR	19.64	40.46	72.25	114.50			
7 × 7	PRD	1.9197	2.2695	2.6868	3.1600			
	CR	19.514 4	39.4099	66.92	91.95			
9 × 9	PRD	1.7250	2.1533	2.4249	2.8738			
	CR	19.98	98.411	60.69	72.25			
11 × 11	PRD	1.6623	1.9647	2.3151	2.6534			
	CR	17.34	36.125	52.7739	54.675			
1212	PRD	1.5936	1.8372	2.1801	2.5100			
15 × 15	CR	15.84	30.807	44.2992 7	43.35			
1.5 1.5	PRD	1.4724	1.7898	2.0519	2.2519			
15 × 15	CR	16.630	27.967	37.9312	34.87			
17 ~ 17	PRD	1.4379	1.6865	2.9004	2.1154			
1/×1/	CR	15.364	16.49	26.7356	29.3188			
19 × 19	PRD	1.4095	1.6225	1.8455	2.0779			
	CR	14.213 1	24.8729	31.2835	24.1792			
21 × 21	PRD	1.2600	1.5616	1.7885	1.9211			
	CR	13.983	25.0785	27.7123	20.5728			

Table 1 Calculated value of the PRD and CR for different map and window sizes for unknown data

An overlapped original and reconstructed signal for worst case for lead I is shown in figure 7. As can be seen from this figure that the QRS region of reconstructed signal overlaps the QRS region of original signal i.e. the QRS region has a proper reconstruction whereas in Non-QRS portion of reconstructed signal most of curve is flat and P and T wave forms are missing. Therefore, we can say that map size of 5x5 with window size of 64 is not suitable for proper reconstruction of ECG signal.



#### Figure 7: Overlapped Original and reconstructed signals for a map size of 5 × 5 and a window size of 64 [PRD = 4.4937 and CR = 31.2381]

Figure 8 shows a graph of the original and reconstructed signals with 1,000 data points, where the lowest PRD is achieved in testing the unknown data, i.e. 628,404 samples for a map size of  $21 \times 21$  and a window size of 8. Using the weight vector for these map and window sizes, a reconstructed signal is obtained.





From Figure 8, the reconstructed signal completely overlaps the original signal; therefore, it can be concluded that the signal can be compressed and reconstructed by properly choosing map and window sizes such that there is hardly any difference between the original and reconstructed signals. Therefore we can say that as the PRD decreases the quality of reconstructed signal improves. Therefore, a map size of 21x21 with window size of 8 is the most suitable combination for a very good reconstructed signal quality amongst all the combinations.

# 5. CONCLUSION

In this paper, the topology preservation property of self organizing map is used as a data compression and visualization tool to compress and visualize ECG signals. In this work the input pattern of filtered and normalized ECG signal are formed. Various parameters that affect learning are varied for a particular map size and the suitable parameters for learning are chosen amongst all the combination of various parameters where the lowest PRD is achieved. These input patterns are afterwards trained and tested on different window and map sizes for visualization and compression with fixed parameters. After training weight vectors of neurons of SOM become representative templates to cluster the various portions of ECG signal i.e. QRS and non QRS region (P, T, U wave and isoelectric line). It is observed that as the window size increases more portion of ECG wave form can be captured in the template.

It is also observed that for small map sizes there are similar type of templates present in neighborhood and cannot distinguish the information properly. But with very large map size uniqueness in templates decreases and number of neuron increases so CR decreases. Therefore a proper map size should be chosen for better visualization and compression. Quality of reconstructed signal is analyzed on PRD and CR.

Therefore, it is concluded that rather than transmitting the whole ECG signal only these weight vectors of neurons (templates) can be transmitted to remote location at a lower bandwidth with in very less time and that can be easily reconstructed for clinical diagnosis for the experts.

### 6. REFERENCES

- U. R. Acharya, J. S. Suri, J. A. E. Spaan, and S. M. Krishnan, Advances in Cardiac Signal Processing, Springer, 2007.
- [2] D. C. Reddy, Cardiological Signal Processing in Biomedical Signal Processing: Principles and Techniques, 12th Ed. ,New Delhi, India, Chapter 7,Tata McGraw–Hill, 2009.
- [3] T. Kohonen, Self Organizing Maps, 3rd Edition, Springer-Verlag, 2001.
- [4] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Pearson Education, 2004, pp 465-500.
- [5] Th. Villmann, Ralf Der, Michael Herrmann and Th. Martinetz, Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement", IEEE transactions on neural networks, vol. 8, no. 2, march 1997.
- [6] Chung-Chian Hsu, "Generalizing Self-Organizing Map for Categorical Data", IEEE Transactions On Neural Networks, Vol. 17, No. 2, March 2006
- [7] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23) (2000) E215-E220.
- [8] J. Pan and W. J. Tompkins, A Real-Time QRS Detection Algorithm, IEEE Transactions on Biomedical Engineering, BME-32(3) (1985) 230-236, http://dx.doi.org/10.1109/TBME.1985.325532.
- [9] S. M. S. Jalaleddine, C. G. Hutchens, R. D. Strattan, and W. A. Coberley, ECG Data Compression Techniques - A Unified Approach, IEEE Transactions on Biomedical Engineering, 37(4) (1990) 329-343, http://dx.doi.org/10.1109/10.52340.
- [10] M. Blanco-Velasco, F. Cruz-Roldán, J. I. Godino-Llorente, J. Blanco-Velasco, C. Armiens-Aparicio, and F. López-Ferreras On the Use of PRD and CR Parameters for ECG Compression International Journal on Medical Engineering & Physics 27(6) (2005) 798-802, http://dx.doi.org/10.1016/j.medengphy.2005.02.007.