First and Second Order Training Algorithms for Artificial Neural Networks to Detect the Cardiac State

Sanjit K. Dash
Department of ECE
Raajdhani Engineering College, Bhubaneswar, Odisha, India

G. Sasibhushana Rao
Department of ECE
College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India

Abstract- In this paper two minimization methods for training feedforward networks with backpropagation are discussed. Feedforward network training is a special case of functional minimization, where no explicit model of the data is assumed. Due to the high dimensionality of the data, linearization of the training problem using orthogonal basis function is one of the options. The focus is functional minimization on a different basis. Two different methods, one based on local gradient and the other on Hessian matrix are discussed. MIT-BIH arrhythmia datasets are used to detect six different beats to know the cardiac state. The beats are Normal (N) beat, Left Bundle Branch Block (L) beat, Right Bundle Branch Block(R) beat, premature ventricular contraction (V) beat, paced (PA) beat and fusion of paced and normal (f) beat.

Keywords – Backpropagation algorithm, Line search, conjugate gradient algorithm, ECG arrhythmia.

I. INTRODUCTION

For feed-forward neural network having differentiable activation functions, there exists a powerful and computationally efficient method[1]-[3]. This method is known as error backpropagation, finds the derivatives of an error function with respect to the weights and biases of the network. These derivatives play a central role in the majority of the training algorithms for multi-layer networks. The simplest backpropagation technique involves steepest descent. The backpropagation technique works in two stages. In the first stage the error function is propagated backward through the network to evaluate the derivatives such as Jacobian and Hessian matrices, whereas in a second stage the weight adjustment using the calculated derivatives and different optimization schemes is taken. This particular type of training of the neural network is known as performance learning, apart from other methods for training the network, such as associative learning and competitive learning. There are several optimization schemes other than simple steepest descent such as conjugate gradient, scaled conjugate gradients, Newton's method, Quasi-Newton methods, Limited memory quasi-Newton methods and Levenberg-Marquardt algorithm. This work is limited to two optimization schemes, i.e. steepest descent and conjugate gradients.

ECG gives the first hand information about the health of the heart. Deviation in the electrical conduction in the heart from the normal conduction, known as arrhythmia, is reflected in the ECG. Automatic arrhythmia detection is necessary as manual verification of ECG for long period is a tedious job. In the past a number of authors have developed different techniques to detect arrhythmias. In [4] hermitian bias function and K-nearest neighbor(Knn) is used for classification of arrhythmias. In[5]the authors used principal component analysis(PCA) in the hybrid multilayered perceptron network (HMLP).Dual tree complex wavelet transform(DTCWT)is used in[6] by Thomas, Das & Ari to classify five different types of ECG beats.Fuzzy classifier is used in [7]by the authors in the first stage with an accuracy of 93.34% and then improved to 98.64% in the second stage by applying genetic algorithm. In [8] PCA is applied to the statistical feature extracted from the spectral correlation of ECG data and then support vector machine(SVM) is applied to classify the five different ECG beats with an accuracy of 98.60%. In [9]the accuracy of ECG classification is 96.2%± 3.4% using SVM. In [10]] Yu and Chou applied Independent Component

Analysis(ICA) & neural network for classifying eight different ECG beats with maximum accuracy of 98.37%.In [11]and[12] Dash & Rao applied balanced training datasets and optimized the hidden layer neurons for better accuracy and quick convergance of aritificial neural network. In [13] Wigner-Ville transformation is used to classify the four different classes of arrhythmia beats. In this paper we have analyzed in details the principles behind the steepest descent and conjugate gradient optimization techniques. Six different classes of arrhythmia beats were classified using these two methods to verify their performance.

The rest of the paper is arranged as follows. Section II explains gradient, Hessian matrix, 1st and 2nd order conditions for optimization, characteristics of quadratic function, line search, steepest descent and conjugate gradient algorithms.Section III discusses the classification result. Section IV gives the conclusion.

II. METHODS

2.1 Analysis of Performance Surface

The error(performance) surface $\mathbf{E(w)}$ of the neural network is a function of all weights and biases $\mathbf{w}$, which is a vector. The Taylor's series expansion of $\mathbf{E(w)} = \mathbf{E}(w_1, w, \dots .. w_n)$ at $\mathbf{w} = \mathbf{w}^*$ is

$$\mathbf{E(w)} = \mathbf{E(w^*)} + [\nabla \mathbf{E(w)}^T|_{\mathbf{w=w^*}}](\mathbf{w} - \mathbf{w^*}) + \frac{1}{2}(\mathbf{w} - \mathbf{w^*})^T[\nabla^2 \mathbf{E(w)}|_{\mathbf{w=w^*}}](\mathbf{w} - \mathbf{w^*}) + \cdots \quad (1)$$

Where $\nabla \mathbf{E(w)}$ is the gradient, and is defined as

$$\nabla \mathbf{E(w)} = \left[\frac{\partial}{\partial w_1}\mathbf{E(w)}, \frac{\partial}{\partial w_2}\mathbf{E(w)}, \dots \frac{\partial}{\partial w_n}\mathbf{E(w)}\right]^T \quad (2)$$

and $\nabla^2 \mathbf{E(w)}$ is the Hessian matrix defined as

$$\nabla^2 \mathbf{E(w)} = \begin{bmatrix} \frac{\partial^2}{\partial {w_1}^2}\mathbf{E(w)} & \frac{\partial^2}{\partial w_1 \partial w_2}\mathbf{E(w)} & \cdots & \frac{\partial^2}{\partial w_1 \partial w_n}\mathbf{E(w)} \\ \frac{\partial^2}{\partial w_2 \partial w_1}\mathbf{E(w)} & \frac{\partial^2}{\partial {w_2}^2}\mathbf{E(w)} & \cdots & \frac{\partial^2}{\partial w_2 \partial w_n}\mathbf{E(w)} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2}{\partial w_n \partial w_1}\mathbf{E(w)} & \frac{\partial^2}{\partial w_n \partial w_2}\mathbf{E(w)} & \cdots & \frac{\partial^2}{\partial {w_n}^2}\mathbf{E(w)} \end{bmatrix} \quad (3)$$

The directional derivative of the performance index $\mathbf{E(w)}$ along the vector $\mathbf{d}$ can be computed from

$$\frac{\mathbf{d}^T \nabla \mathbf{E(w)}}{\|\mathbf{d}\|} \quad (4)$$

The second derivative along $\mathbf{d}$ can also be computed from

$$\frac{\mathbf{d}^T \nabla^2 \mathbf{E(w)}\, \mathbf{d}}{\|\mathbf{d}\|^2} = \frac{\mathbf{d}^T \mathbf{H} \mathbf{d}}{\|\mathbf{d}\|^2} \quad (5)$$

1. The point $\mathbf{w^*}$ is a strong minimum of $\mathbf{E(w)}$ if for all $\Delta\mathbf{w}$, $\mathbf{E(w^*)} < \mathbf{E(w^* + \Delta w)}$.
2. The point $\mathbf{w^*}$ is a unique global minimum of $\mathbf{E(w)}$ if for all $\Delta\mathbf{w} \neq 0$, $\mathbf{E(w^*)} < \mathbf{E(w^* + \Delta w)}$.
3. The point $\mathbf{w^*}$ is weak minimum of $\mathbf{E(w)}$ if for all $\Delta\mathbf{w}$, $\mathbf{E(w)} \leq \mathbf{E(w^* + \Delta w)}$.

2.2 Necessary Conditions For Optimality-

After defining the optimum (minimum) point, the conditions necessary for such point is to be identified.

2.2.1 First-order Condition-

Gradient must be zero at the minimum point i.e.

$$\nabla E(\mathbf{w})|_{w=w^*} = 0 \tag{6}$$

Proof-

Putting $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}^*$ in (1) and taking $\|\Delta\mathbf{w}\| \ll 1$ Taylor's series expansion (after neglecting the 2nd and higher order term of $\Delta\mathbf{w}$) is approximated as

$$E(\mathbf{w}) = E(\mathbf{w}^*) + [\nabla E(\mathbf{w})^T|_{w=w^*}]\,\Delta\mathbf{w}$$

As $\mathbf{w}^*$ is the minimum point of the function E, any point $\mathbf{w}$ other than $\mathbf{w} = \mathbf{w}^*$, function $E(\mathbf{w}) \geq E(\mathbf{w}^*)$.

This is possible if $[\nabla E(\mathbf{w})^T|_{w=w^*}]\,\Delta\mathbf{w} \geq 0$.

If $\Delta\mathbf{w} \neq 0$ then $E(\mathbf{w}) = E(\mathbf{w}^* + \Delta\mathbf{w}) \geq E(\mathbf{w}^*)$

But $E(\mathbf{w}) = E(\mathbf{w}^* - \Delta\mathbf{w}) \leq E(\mathbf{w}^*)$ which is a contradiction. Hence, for any $\Delta\mathbf{w}$, $[\nabla E(\mathbf{w})^T|_{w=w^*}]\,\Delta\mathbf{w} = 0$. This is true only when $[\nabla E(\mathbf{w})|_{w=w^*}] = 0$ that is gradient must be zero at the minimum point. Any point $\mathbf{w} = \mathbf{w}_1$ that satisfies $[\nabla E(\mathbf{w})|_{w=w_1}] = 0$ is known as stationary point. $\mathbf{w}^*$ is also a stationary point.

2.2.2 Second-order Condition-

The *sufficient condition* for a strong minimum to exist is that the Hessian matrix must be positive definite. The minimum is strong if from the Taylor series the second order term

$$\nabla^2 E(\mathbf{w})|_{w=w^*} = 0 \tag{7}$$

but the third order term $\nabla^3 E(\mathbf{w})|_{w=w^*} > 0$. *Therefore the second order necessary condition for strong minimum is that the Hessian matrix must be positive semidefinite.*

Since the gradient of $E(\mathbf{w})$ is zero at stationary point $\mathbf{w}^*$, the Taylor series expansion is

$$E(\mathbf{w}^*+\Delta\mathbf{w}) = E(\mathbf{w}^*) + \tfrac{1}{2}\Delta\mathbf{w}^T[\nabla^2 E(\mathbf{w})|_{w=w^*}]\Delta\mathbf{w} + \cdots$$

For small $\|\Delta\mathbf{w}\|$, $E(\mathbf{w})$ can be approximated by the first two terms neglecting the 3rd and higher order terms of $\Delta\mathbf{w}$. Thus, strong minimum at $\mathbf{w}^*$ exists if

$$\Delta\mathbf{w}^T[\nabla^2 E(\mathbf{w})|_{w=w^*}]\Delta\mathbf{w} > 0.$$

This is possible if the Hessian matrix $\mathbf{H} = \nabla^2 E(\mathbf{w})|_{w=w^*}$ is positive definite.

$\mathbf{H}$ is positive definite only when $\mathbf{z}^T\mathbf{H}\mathbf{z} > 0$ where $\mathbf{z}$ is any vector. $\mathbf{H}$ is positive definite if all of its eigenvalues are positive. Similarly $\mathbf{H}$ is positive semidefinite if all of its eigenvalues are nonnegative.

For positive semidefinite $\mathbf{z}^T\mathbf{H}\mathbf{z} \geq 0$.

The necessary conditions for $\mathbf{w}^*$ *to be minimum, strong/weak of* $\mathrm{E}(\mathbf{w})$ *are*

$$\nabla \mathrm{E}(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*} = 0$$

and $\mathbf{H} = \nabla^2 \mathrm{E}(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*} \geq 0$ *(positive semidefinite)*

The sufficient conditions for $\mathbf{w}^*$ *to be minimum, strong/weak of* $\mathrm{E}(\mathbf{w})$ *are*

$$\nabla \mathrm{E}(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*} = 0$$

and $\mathbf{H} = \nabla^2 \mathrm{E}(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*} > 0$ *(positive definite)*

2.3 Quadratic Functions-

The general form of a quadratic function is

$$\mathrm{E}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{H}\mathbf{w} + \mathbf{b}^{\mathrm{T}}\mathbf{w} + \mathrm{E}_0 \tag{8}$$

where matrix $\mathbf{H}$ is symmetric.(if not symmetric it can be replaced by a symmetric matrix that produces the same $\mathrm{E}(\mathbf{w})$)
From equation (8) we can compute

$$\nabla \mathrm{E}(\mathbf{w}) = \mathbf{H}\mathbf{w} + \mathbf{b} \tag{9}$$

and

$$\nabla^2 \mathrm{E}(\mathbf{w}) = \mathbf{H} \tag{10}$$

Now define vector $\mathbf{d}$ as

$$\mathbf{d} = \mathbf{B}\mathbf{c} \tag{11}$$

where $\mathbf{B}$ is the matrix of eigenvectors of the Hessian matrix $\mathbf{H}$.

$$\mathbf{H} = \mathbf{B}\Lambda\mathbf{B}^{\mathrm{T}} \tag{12}$$

and $$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

where λ_i is the eigenvalue of Hessian matrix $\mathbf{H}$.
With equations (5) (11) & (12) we can rewrite equation (5) as

$$\frac{\mathbf{d}^{\mathrm{T}}\mathbf{H}\mathbf{d}}{\|\mathbf{d}\|^2} = \frac{\mathbf{c}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}(\mathbf{B}\Lambda\mathbf{B}^{\mathrm{T}})\mathbf{B}\mathbf{c}}{\mathbf{c}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{c}} = \frac{\mathbf{c}^{\mathrm{T}}\Lambda\mathbf{c}}{\mathbf{c}^{\mathrm{T}}\mathbf{c}} = \frac{\sum_{i=1}^{n}\lambda_i c_i^2}{\sum_{i=1}^{n} c_i^2} \tag{13}$$

The above equation tells that this second derivative is just a weighted average of the eigenvalues, hence lies between the maximum and minimum of the eigenvalues.

$$\lambda_{\min} \leq \frac{\mathbf{d}^{\mathrm{T}}\mathbf{H}\mathbf{d}}{\|\mathbf{d}\|^2} \leq \lambda_{\max} \tag{14}$$

Substituting $\mathbf{z}_{\max}$ (eigenvector corresponding to $\lambda_{\max}$) for $\mathbf{d}$ in Eq.(13) we obtain

$$\frac{\mathbf{z}_{\max}{}^{\mathrm{T}}\mathbf{H}\mathbf{z}_{\max}}{\|\mathbf{z}_{\max}\|^2} = \frac{\sum_{i=1}^{n}\lambda_i c_i^2}{\sum_{i=1}^{n} c_i^2} = \lambda_{\max} \tag{15}$$

The above equation shows that the maximum second derivative occurs in the direction of the eigenvector corresponding to the largest eigenvalue.(In each of the eigenvector directions, the second derivatives are equal to the corresponding eigenvalue. In other directions in the second derivative is the weighted average of the eigenvalues.) The eigenvalues are the second derivatives in the directions of the eigenvectors. The eigenvectors are known as the principal axes of the function contours as shown in Fig 1. In case the contour is an ellipse (Hessian matrix is 2x2) the minimum curvature will occur in the direction of first eigenvector z_1. This means that we will cross the contour lines more slowly in this direction. The maximum curvature will occur in the direction of 2nd eigenvector z_2. Therefore, we will cross the contour lines more quickly, in this direction.

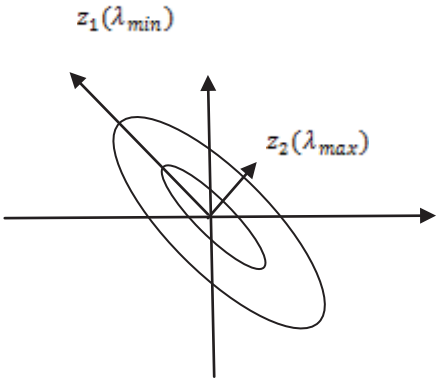

Fig.1 Contour of 2X2 Hessian matrix with principal axes.

The followings are the characteristics of the quadratic function.

1. *If all the eigenvalues of Hessian matrix are positive, then the function will have a single strong minimum.*
2. *If all the eigenvalues of Hessian matrix are negative, then the function will have a single strong maximum.*
3. *If some of the eigenvalues of the Hessian matrix are positive and others are negative, then the function will have a single saddle point.*
4. *If all the eigenvalues are nonnegative, with some are zero, then the function will either have a weak minimum or will have no stationary points.*
5. *If all the eigenvalues are nonpositive, with some are zero, then the function will either have a weak maximum or will have no stationary points.*

We have already investigated the performance surface. Now we will develop algorithms to search the performance space and locate the minimum points of the surface $E(\mathbf{w})$. (By optimizing the weights and biases for a given neural network) If $\mathbf{w}_0$ is the initial guess this can be updated the equation

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{d}_k \quad (16)$$

$$\Delta \mathbf{w}_k = \mathbf{w}_{k+1} - \mathbf{w}_k = \alpha_k \mathbf{d}_k \quad (17)$$

Where $\mathbf{d}_k$ is the search direction and α_k is the learning rate.

2.4 Line Search

Some search direction in the weight space to find the minimum of the error function along that direction is known as *Line Search.* In gradient descent the direction at each step is given by the local negative gradient of the error function (weight) and the step size is determined by learning rate.

Let $\mathbf{w}(\tau)$ be the weight vector and $\mathbf{d}(\tau)$ be the search direction at step τ in the weight space. The next value of weight vector $\mathbf{w}(\tau + 1)$ along the search direction is given as

$$\mathbf{w}(\tau + 1) = \mathbf{w}(\tau) + \alpha(\tau)\mathbf{d}(\tau) \quad (18)$$

Where $\alpha(\tau)$ is the parameter to minimize

$$E(\alpha) = E\{\mathbf{w}(\tau) + \alpha(\tau)\mathbf{d}(\tau)\} \quad (19)$$

$\alpha(\tau)$ may be different for different step or constant (α). In steepest descent algorithm α is constant. For varying α at each step may speed up the convergence.

2.5 Steepest Descent-

While updating the guess for the optimum point using Eq.(16) at each iteration the performance index decreases i.e. $E(\mathbf{w}_{k+1}) < E(\mathbf{w}_k)$. Now we have to choose the direction of $\mathbf{d}_k$ so that we will move downhill of the performance surface. Consider the first order Taylor's series expansion of $E(\mathbf{w})$ about the old guess $\mathbf{w}_k$.

$$E(\mathbf{w}_{k+1}) = E(\mathbf{w}_k + \Delta\mathbf{w}_k) \approx E(\mathbf{w}_k) + \mathbf{g}_k^T\Delta\mathbf{w}_k \quad (20)$$

Where $$\mathbf{g}_k \equiv \nabla E(\mathbf{w}_k)|_{w=w_k} \quad (21)$$

$E(\mathbf{w}_{k+1})$ to be less than $E(\mathbf{w}_k)$, $\mathbf{g}_k^T\Delta\mathbf{w}_k$ must be negative, i.e.

$$\mathbf{g}_k^T\Delta\mathbf{w}_k = \alpha_k\mathbf{g}_k^T\mathbf{d}_k < 0 \quad (22)$$

For small learning rate α_k which is positive $\mathbf{g}_k^T\mathbf{d}_k$ must be less than zero.

Hence $$\mathbf{d}_k = -\mathbf{g}_k \quad (23)$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k\mathbf{g}_k \quad (24)$$

2.6 Conjugate Gradient Algorithm

The derivation of the steepest descent algorithm is based on first order Taylor's series expansion, whereas the conjugate gradient algorithm is based on the second order Taylor's series. When the other second order algorithm, Newton's method[14] converges in finite number of iterations ,if the error surface is quadratic only. It needs to calculate and store the second order derivatives, which is impractical when the parameters are large. This is true with neural networks, where the number of weights can be several hundreds to thousands. In this case a method that requires a first derivative only, but still has quadratic termination is desirable. The steepest descent algorithm with linear search at each iteration, the search directions at consecutive iterations are orthogonal. For a quadratic function with elliptical contours this produces a zig-zag trajectory of short steps which are not the best choice. Conjugate directions search guarantees quadratic termination.

A set of vectors $\{\mathbf{d}_k\}$ is mutually conjugate with respect to a positive definite Hessian matrix $\mathbf{H}$ if and only

if $$\mathbf{d}_k{}^T\mathbf{H}\mathbf{d}_j = 0 \quad \forall \ k \neq j \quad (25)$$

With orthogonal vectors, there are an infinite number of mutually conjugate sets of vectors in the given n-dimensional space. One set of conjugate vectors are the eigenvectors of $\mathbf{H}$. If $\{\lambda_1, \lambda_2, \ldots \lambda_n\}$ and $\{\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_n\}$ are the eigenvalues and eigenvectors respectively of the Hessian matrix $\mathbf{H}$, then

$$\mathbf{z}_k{}^T\mathbf{H}\mathbf{z}_j = \lambda_j\mathbf{z}_k{}^T\mathbf{z}_j = 0 \quad \forall \ k \neq j \quad (26)$$

The above equation shows that eigenvectors of a symmetric matrix are mutually orthogonal and conjugate. Though the quadratic function can be minimized exactly by searching along the eigenvectors of matrix$\mathbf{H}$, practically it will not help much as in each iteration the $\mathbf{H}$ matrix is to be calculated. Development of an algorithm without calculating

Hessian matrix is described. The conjugacy condition is restated without using Heesian matrix. Consider the quadratic error function

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} + \mathbf{b}^T\mathbf{w} + E_o$$

Where the gradient $\nabla E(\mathbf{w}) = \mathbf{H}\mathbf{w} + \mathbf{b} = \mathbf{g}$ Hessian matrix $\nabla^2 E(\mathbf{w}) = \mathbf{H}$

Change in the gradient $\Delta\mathbf{g}_k = \mathbf{g}_{k+1} - \mathbf{g}_k = \mathbf{H}\Delta\mathbf{w}_k$

As $\Delta\mathbf{w}_k = \mathbf{w}_{k+1} - \mathbf{w}_k = \alpha_k\mathbf{d}_k$ and α_k is chosen to minimize $E(\mathbf{w})$ in the direction of $\mathbf{d}_k$.

The conjugacy condition in (25) is now restated as

$$\alpha_k\mathbf{d}_k^T\mathbf{H}\mathbf{d}_j = \Delta\mathbf{w}_k^T\mathbf{H}\mathbf{d}_j = \Delta\mathbf{g}_k^T\mathbf{d}_j = 0 \quad \forall \ k \neq j \qquad (27)$$

Now the conjugacy condition is restated as the changes in the gradient at successive iterations instead of the Hessian matrix.The search directions are conjugate if they are orthogonal to the changes in the gradient.

Algorithm-

1. *The first search direction* $\mathbf{d}_0$*, is negative of the gradient* $\mathbf{g}_0$.
2. *The other search directions* $\mathbf{d}_{k+1}$ *is a vector orthogonal to* $\Delta\mathbf{g}_k$ *where* $k \geq 1$*. Using a procedure similar to Gram-Schmidt orthogonalization* [15] $\mathbf{d}_k = -\mathbf{g}_k + \beta_k\mathbf{d}_{k-1}$.
3. *The coefficient* β_k*, for the Polak-Ribiere algorithm is* $\beta_k = \frac{\mathbf{g}_{k+1}^T(\mathbf{g}_{k+1}-\mathbf{g}_k)}{\mathbf{g}_k^T\mathbf{g}_k}$

III. RESULTS AND DISCUSSION

MIT-BIH arrhythmia data [16] were used for this work. Total 998 arrhythmia beats from all six classes were considered for classification both by steepest descent backpropagation neural network (SDNN) and complex conjugate backpropagation neural network (CGNN). The network was trained by 90 patterns, taking 15 patterns from each class. Three layer neural network with 35 neurons in the hidden layer was used. The binary neural network was used for this multiclass classification. In SDNN variable learning rate was used for early convergence. The network was trained in 1000 iterations. In CGDN convergence was very fast. The binary networks were converged in 75,14,104,103,16 and 60 iterations only. The confusion matrix for both the techniques is given in table-1. Three performance parameters were considered for evaluation of the classifiers. The terms used in evaluating the system is defined as

TP: true positive, TN: true negative FP: false positive, FN: false negative

$$P_c = TP + FN \ \& \ N_c = FP + TN$$

$$\text{sensitivity} = \frac{TP}{TP + FN} \qquad (28)$$

$$\text{specificity} = \frac{TN}{TN + FP} \qquad (29)$$

$$\text{accuracy} = \frac{TP + TN}{P_c + N_c} \qquad (30)$$

Assessment matrix in table-2 shows the values of different parameters, for SDNN & CGNN. Accuracy in case of CGNN is more than 99.00% for all the classes, whereas sensitivity and specificity are 100% for two classes.

Table -1. Confusion Matrix of SDNN & CGNN

	SDNN							CGNN					
	N	L	R	V	PA	f		N	L	R	V	PA	f
N	703	0	0	1	0	12		712	0	0	0	0	4
L	0	87	0	0	0	1		0	86	0	0	0	2
R	0	0	74	0	0	0		0	0	74	0	0	0
V	0	3	1	39	0	1		1	2	4	36	0	1
PA	0	0	0	0	62	2		0	0	0	0	64	0
f	1	1	0	0	0	10		0	1	0	0	0	11

Table 2. Assessment Matrix of SDNN & CGNN

	SDNN				CGNN		
	Sensitivity	Specificity	Accuracy		Sensitivity	Specificity	Accuracy
N	0.9818	0.9965	0.9860		0.9944	0.9965	0.9950
L	0.9886	0.9956	0.9950		0.9773	0.9967	0.9950
R	1.0000	0.9989	0.9990		1.0000	0.9957	0.9960
V	0.8864	0.9990	0.9940		0.8182	1.0000	0.9920
PA	0.9688	1.0000	0.9980		1.0000	1.0000	1.0000
f	0.8333	0.9838	0.9820		0.9167	0.9929	0.9920

IV. CONCLUSION

The principles of both the optimization techniques are discussed in details. From the analysis, it was found that conjugate gradient was developed for early convergence, which is the main drawback of the steepest descent method. In the experiment, CGNN was converged in just 14 iterations the lowest to highest iterations of 104. The overall performance of CGNN is better than SDNN. Further improvement in conjugate gradient method can be done by judiciously selecting the initial direction.

REFERENCES

[1] C.M.Bishop, Neural Network for Pattern Recognition, Clarendon Press, Oxford, 1995.

[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd ed., Wiley, New York, 2001.

[3] D.Nguyen,B.Widrow, "Improving the learning Speed of 2-LayerNeural Networks by choosing Initial Values of the Adaptive Weights", IEEE International Conference on Neural Networks, Vol-3, pages.21-26, 17-21,June 1990,San Diego, CA,USA.

[4] S.Karimifard,A.Ahmadian, "Morphological Heart Arrhythmia Detection Using Hermitian Basis Functions and KNN classifier", Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA, Aug 30-Sept 3,2006,pp.1367-1370.

[5] A.I.Amirudden,M.S.A.Megat,M.FSaaid,A.H.Jahidin & Z.H.Noor, "Feature Reduction and Arrhythmia Classification via hybrid Multilayered Perceptron Network", IEEE 3rd International Conference on System Engineering and Technology,19-20 Aug 2013,pp.290-294, Shah Alam, Malaysia.

[6] M.Thomas,M.Das & S.Ari, "Automatic ECG arrhythmia classification using dual tree complex wavelet based features", International Journal of Electronics and Communications, Vol.69, Issue 4, April 2015, pp.715-721.

[7] M.H. Vafaie, M. Ataei, H.R. Koofigar, **"**Heart diseases prediction based on ECG signals classification using a genetic fuzzy system and dynamical model of ECG signals", Biomedical Signal Processing and Control, Vol. 14, November 2014,pp.291-296.

[8] A.F.Khalaf, M. I. Owis, I. A. Yassine, "A novel technique for cardiac arrhythmia classification using spectral correlation and support vector machine", Vol.42,Nov 2015,pp.8361-8368.

[9] Q.Li,C.Rajagopalann,G.D.Clifford, "Ventricular Fibrillation and Tachycardia Classification Using a Machine Learning Approach", IEEE Transaction on Biomedical Engineering,Vol.61, NO.6, June 2014,pp.1607-1613.

[10] S.N.Yu & K.Chou, "Combining Independent Component Analysis and Bacpropagation Neural Network for ECG Beat Classification", proceedings of the 28th IEEE EMBS Annual International Conference, New York city, USA, Aug 30- Sep 3,2006.

[11] Sanjit K. Dash and G.Sasibhusana Rao, Robust Multiclass ECG Arrhythmia Detection Using Balanced Trained Neural Network, IEEE International Conference on Electrical, Electronics & Optimization Techniques(ICEEOT-2016),3rd -5th March 2016, Chennai, Tamilnadu, India.

[12] Sanjit K.Dash and G.Sasibhusan Rao, **"**Optimized Neural Network for Improved Electrocardiogram Beat classification", 6th IEEE International Advanced Computing connference(IACC-2016),27th-28th Feb 2016, Bhimavaram, AP, India.

[13] Sanjit K.Dash and G.Sasibhusan Rao, "Arrhythmia Detection Using Wigner-Ville Distribution Based Neural Network", Procedia Computer Science 85 (2016) 806 – 811. www.sciencedirect.com .

[14] R.Battiti, First and second order methods for learning between steepest descent and Newton's method,Neural Computation, Vol.4, No.2, 1992, pp.141-166.

[15] E. Scales, *Introduction to Non-Linear Optimization*, New York: Springer-Verlag, 1985.

[16] MIT-BIH Arrhythmia Database-Physionet,https://www.physionet.org/..../database/.