Chapter 3

Artificial Neural Network(ANN) for Classification

Contents

Artificial Neural Network (ANN) has provided first step towards the Artificial Intelligence. Due to its effectiveness in classification, it is widely used in medical field to diagnose diseases. In this chapter, ANN is used with Levenberg -Marquardt optimization algorithm in back propagation learning rule to updates the network weights. Classification accuracy is measured using various statistical measures. Despite its drawback to stuck algorithm in local optimum, a very good classification accuracy is obtained. In section 3.1, a general introduction about ANN is given.

Learning of ANN is discussed in section 3.2. Experimental set up and results are given in section 3.3 and summary is presented in section 3.4.

3.1 Introduction

Artificial Neural Networks(ANNs) are very powerful technique compared to traditional statistical techniques. They provide more accurate results of prediction compared to regression model of statistics. An ANN is a parallelly distributed information processing structure. It is built up with processing units (Neurons) interconnected via connections (Dendrites) and each processing unit has a single output connection (Axon). The information is completely local. i.e. the information is stored in the processing unit's local memory.

A major task of medical science is to diagnose diseases properly. ANNs are most widely and successfully used soft computing technique to diagnose diseases. They are playing a vital role in classification because of their capacity to learn and store knowledge.

ANN is analogue to the human brain which is made up of neurons. A neuron receives input signals through dendrites from surrounding neurons. These signals are processed in the cell body of the neuron and then transmitted through the axon to other neuron which is called output terminal [143].

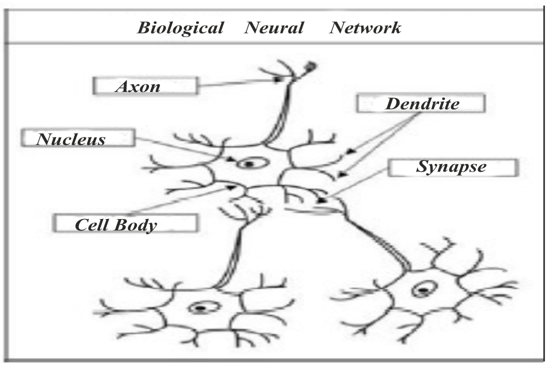

Figure 3.1: Biological Neural Network

ANN simulate this process, in which input signals are input variables which are storing information about the patterns. These input variables are weighted according to their importance which is simulating the working of dendrites of the biological neural networks. These weighted signals are summed in hidden layers and processed by activation functions such as sigmoid function, radial basis function, sine, cosine, exponential function etc.

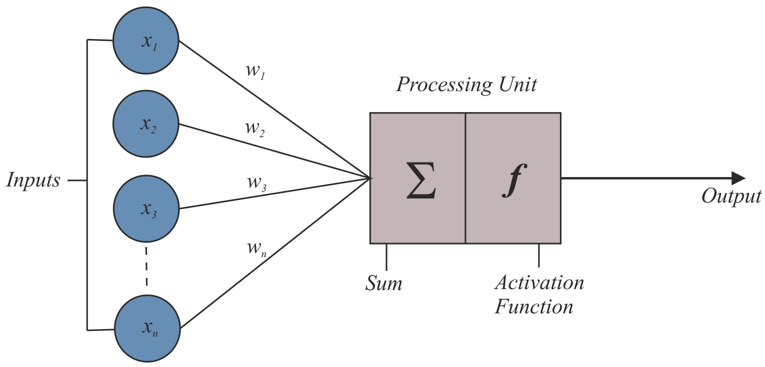

Figure 3.2: Artificial Neural Network

In 1943 McCulloch and Pitts had created a simple model of neural network [86]. They described how neurons in brain might work and show how to compute any arithmetic or logical functions using simple neural network model. The first learning algorithm, called Perceptron learning rule for classification of linearly separable data, was developed by Rosenblatt in 1958, which is a single layer network (Network contains only input and output layer is called single layer network) [110]. But, ANN becomes more popular from 1986 with multiple layered neural networks (see figure 3.3) (Multilayer Network consist of input layer, output layer and one or more hidden layer) with Back Propagation learning rule which can classify non linear data [113].

Amato *et al.* discussed various applications of ANNs in medical diagnosis and represented ANNs as more powerful and reliable tool [3]. Bapko *et al.* used ANN in diagnosis of various skin diseases and achieved 90% success [7]. Mehdy *et al.* have shown that ANNs plays an important role in diagnosis of breast cancer [87]. ABCD rule (An algorithm which differentiate benign from malignant melanocytic tumors)has become a standard practice by many dermatologists. Okuboyejo *et al.* have characterized the ABCD rule into quantitative attributes measured by image analysis and implore texture analysis technique with Gabor wavelet (to make scale,

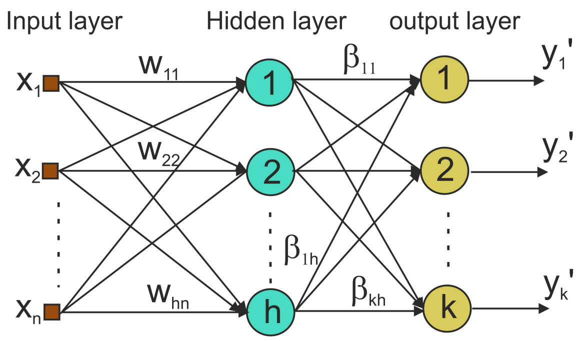

Figure 3.3: Single Hidden Layer Artificial Neural Network

translation and rotation invariant) in order to automate the classification process. Then lesions are classified as benign or malignant using Multilayer Perceptron Classifier (MLP) [93]. Maghsoudi *et al.* have used ANN for diagnosis and prediction of oral diseases such as Lichen Planus, Leukoplakia and Squamous cell carcinoma [85]. Filimon *et al.* have discussed how ANNs can be used successfully for quick diagnosis of some diseases with very similar features [37].

This chapter aims to diagnose some common skin diseases like Bacterial Infection, Fungal infection, Scabies and Eczema by using ANNs as classification technique. The algorithm and learning method used to classify above mentioned diseases are discussed in the next section.

3.2 Learning using ANN

3.2.1 Learning Methodology

The learning method is discussed for one hidden layer, which can be extended to more than one hidden layers.
Let there be m training samples and where each input sample is $\mathbf{x} \in \mathbb{R}^n$ and let $\mathbf{y} \in \mathbb{R}^k$ be the actual output and $\mathbf{y}^* \in \mathbb{R}^k$ be the network output.

The aim is to find the set of weights which minimizes the error between actual output y_l and network's calculated output y_l^*, $l = 1, 2, ..., k$ defined as:

$$E = \sum_{l=1}^{k} e_l^2 = \frac{1}{2}\sum_{l=1}^{k}(y_l - y_l^*)^2, \tag{3.2.1}$$

where the network output (y_1^*, y_2^*,...,y_k^*) is calculated in the following way:

1. Assign initial weights $\{w_{j1}, w_{j2}, ..., w_{jn}\}$, $j = 1, 2, ..., h$ from each input nodes to j^{th} hidden node where n is the number of input nodes, h is the number of hidden nodes.
2. The input of j^{th} hidden node is calculated as: $net_j = \sum_{i=1}^{n} x_i w_{ji}$, $j = 1, 2, ..., h$.
3. The output of j^{th} hidden node is given by $z_j = g_h\left(\sum_{i=1}^{n} x_i w_{ji}\right)$. Here $g_h(x)$ is some activation function. We use Sigmoid function in our study which is defined as:

$$g(x) = \frac{1}{(1 + e^{-x})} \tag{3.2.2}$$

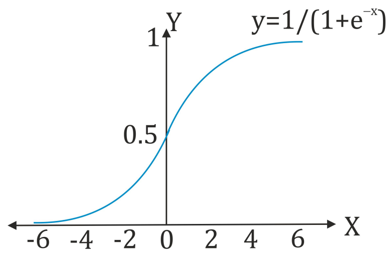

Figure 3.4: Sigmoid Function

4. Let $\{w_{j1}, w_{j2}, ..., w_{jk}\}$ be the weight connecting the j^{th} hidden node and the l^{th} output nodes, where $l = 1, 2, ..., k$ and $j = 1, 2, ..., h$.
5. Calculate the network output $\mathbf{y}_l^*$ as:

$$y_l^* = g_o\left(\sum_{j=0}^{h} w_{lj} z_j\right) \tag{3.2.3}$$

where,

- z_j be the output of j^{th} hidden node.

- w_{lj} be the weight connecting j^{th} hidden node to l^{th} output node.
- g_o is the activation function. Here sigmoid function is consider.
- y_l^*: output of l^{th} node of output layer.

6. After calculating output of the each node of output layer, using forward propagation phase, calculate network predicted output error E using the formula (3.2.1).
7. If network error E is greater than the tolerance, start Back propagation phase and modify weights of the network to minimize the error E.
8. We have used Levenberg-Marquard optimization algorithm, which is discussed in section (3.2.2.), to minimize the network error E by updating weights of the network.

3.2.2 The Levenberg-Marquard algorithm

In this study Levenberg-Marquard(LM) algorithm is used to minimize the network error. LM is a general least squares optimization method. It has more potential to converge to an optimal solution compare to the most popular steepest descent optimization technique. LM is the combination of Steepest descent and the Newton methods. The basic Back propagation algorithm uses steepest descent method or negative gradient direction to update weights. We know that any function increases most rapidly in the direction of gradient, so to minimize the error E (equation 3.2.1) the negative of the gradient direction is chosen. Though steepest descent method uses negative of gradient to update the weights, it's convergence rate is not always faster [29]. The steepest descent find optimum when design vector is away from optimum point and Newton method gives fastest convergence when weight vector is closest to the optimum value. LM algorithm has advantage of both methods.
It is a quadratic convergent method in which no need to calculate Hessian matrix. The Hessian matrix can be approximated by

$$H = J^T J$$

and the gradient is computed as

$$g = J^T E$$

where, J is the Jacobian containing the first order partial derivatives of the network error E (equation 3.2.1) with respect to the weights and biases. E is the vector of network error.
In LM weights are updated as [115]:

$$w_{k+1} = w_k - [J^T J + \eta I]^{-1} J^T E$$

where η is the learning rate.

3.3 Experimental set up and results

Artificial Neural Network is applied to the Dataset-I (Appendix-A) to diagnose four common skin diseases viz., Bacterial Infection, Fungal Infection, Eczema and Scabies. To evaluate the performance of the classifier, the Dataset is randomly divided into 80-20% and 70-30% partitions, i.e., 80% and 70% data for training and 20% and 30% for testing respectively.
The Neural Network is designed and implemented in MATLAB using Neural Network toolbox. In ANN the popular Back propagation (BP) learning algorithm is used where gradients can be computed efficiently by propagation from the output to the input. The network is created using newff() matlab in-built function (Appendix-A). Activation function is Sigmoid function and training is done using Levenberg-Marquardt algorithm. Network is trained using one hidden layer and two hidden layers. When network is built up using single hidden layer, 10 nodes are used in the hidden layer, and when network is built up using two hidden layers, 20 nodes are used in the 1^{st} layer and 10 nodes are used in the 2^{nd} hidden layer (figure 3.5). Classification accuracies displayed in the (Table 3.1)are average of 50 trials.

Accuracy of multiclass problem is measured by two different measures of accuracy:

1. By finding ratio of correct prediction to the total prediction made (refer definition 2.4.3). From confusion matrix it can be found using,

$$\text{Accuracy} = \frac{\text{sum of diagonal elements}}{\text{Total of all elements}}$$

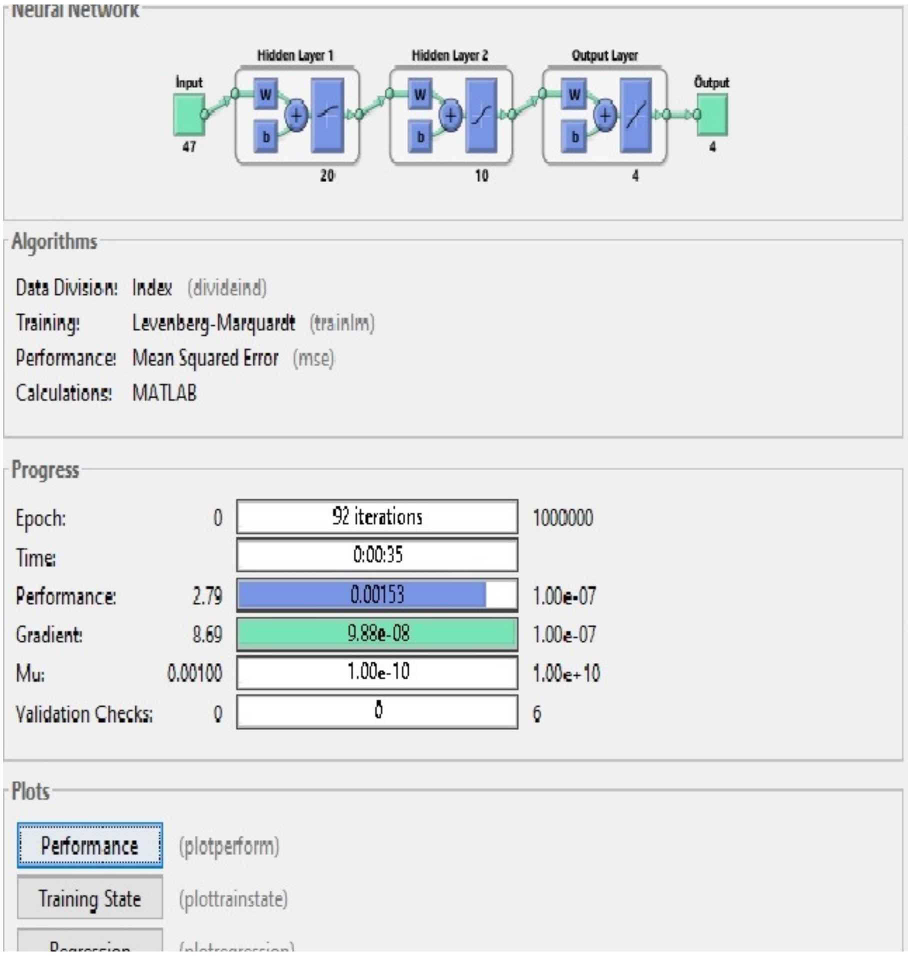

Figure 3.5: Neural Network with Two Hidden Layers

2. By finding F-Score (refer definition 2.4.6)which is the harmonic mean of Precision (refere definition 2.4.4) and Recall (refer definition 2.4.5)

$$\text{Precision} = \frac{\text{sum of diagonal elements}}{\text{sum of column element of the particular class}}$$

$$\text{Recall} = \frac{\text{sum of diagonal elements}}{\text{sum of row elements of the particular class}}$$

The following confusion matrix is generated using neural network tool box.

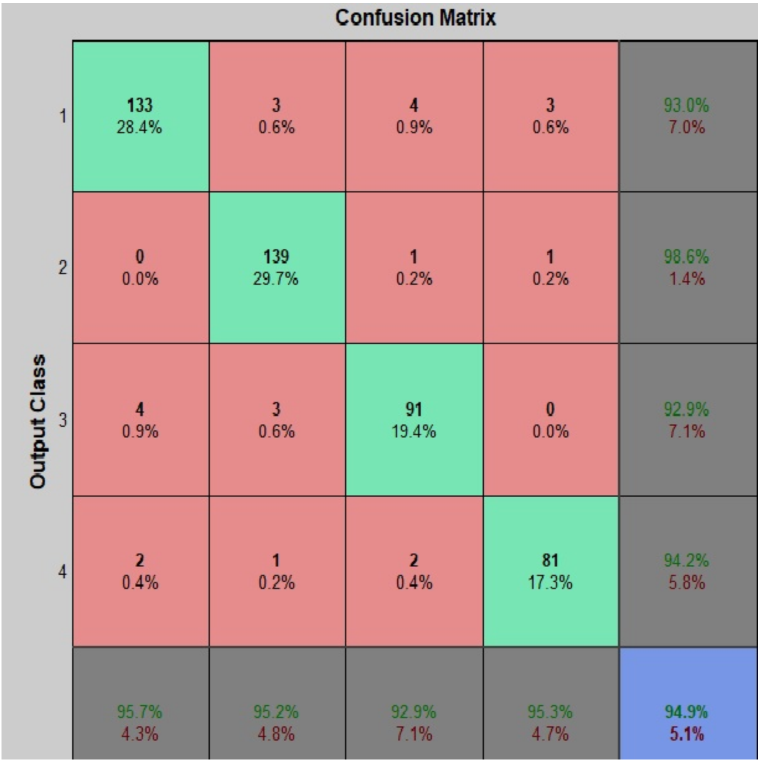

Figure 3.6: Confusion Matrix

Table 3.1: Performance Results using ANN

	70%-30% data partition		80%-20% data partition	
	Accuracy	F-Score	Accuracy	F-Score
1-Hidden Layer	95.82%	93.59%	96.20%	94.19%
2-Hidden Layers	96.23%	94.23%	97.17%	95.70%

3.4 Summary

In this chapter ANN with Back Propogation learning rule is used to diagnose four commonly observed dermatological diseases of Dataset-I discussed in Appendix-A. To minimize the network error, Levenberg-Marquard(LM) optimization algorithm is used to update weights. Back Propagration algorithm may face several issues like local minima, proper choice of learning rate and over fitting etc even though from this study it has been found that good classification accuracy can be achieved for the Dataset under study. The Dataset-I is imbalanced therefore along with the normal accuracy measure, other good statistical measure of accuracy for imbalance data namely F-score is used to find the classification accuracy. From the observations it may be inferred that the performance of ANN is giving very good classification accuracy with two hidden layers, where 20 and 10 hidden nodes are used in 1^{st} and 2^{nd} layer respectively.