CHAPTER 6
CLASSIFICATION

6.1 PROLOGUE

Classification is one of the most frequently encountered decision making tasks of human activity. A classification problem occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object. Many problems in business, science, industry, and medicine can be treated as classification problems. Neural networks have emerged as an important tool for classification. The recent vast research activities in neural classification have established that neural networks are a promising alternative to various conventional classification methods.

The advantage of neural networks lies in the following theoretical aspects. First, neural networks are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Second, they are universal functional approximates in that neural networks can approximate any function with arbitrary accuracy. Third, neural networks are nonlinear models, which makes them flexible in modeling real world complex relationships. Finally, neural networks are able to estimate the posterior probabilities, which provide the basis for establishing classification rule and performing statistical analysis [197, 198].

6.2 NEURAL NETWORKS

Neural networks (NN) can learn various tasks from training examples; classify phenomena, and model nonlinear relationships. However, the primary features that are of
concern in the design of the network are problem specific. Despite the availability of some guidelines, it would be helpful to have a computational procedure in this aspect, especially for the optimum design of an NN. The gradient descent algorithms have reported difficulties in learning the topology of the networks whose weights they optimize.

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. In the classification stage, two classifiers based on supervised machine learning have been developed. The first classifier based on feed forward artificial neural network (FF-ANN) and the second classifier based on Back-Propagation Neural Network.

Perceptrons is a special case of feed forward networks with only input and output nodes. Three main Perceptron learning algorithms are covered: mistake bound Perceptron algorithm, Perceptron training rule and the Delta rule. The delta rule uses gradient descent, which makes it easy to compute what changes are needed to optimize the network. The backpropagation-learning algorithm is widely used for multi-layer feed forward network. This uses gradient descent as well. Bayesian learning is based on statistics and knowledge of prior statistics to classify or predict. Bayes Theorem is central to Bayesian learning.
Artificial Neural Networks (ANNs) allow for learning using highly parallel series of simple units and are suited for data that is noisy and vector based. Backpropagation is a learning algorithm for multi-layered feed forward networks that uses the sigmoid function. Hidden layer is the set of nodes that are not input or output units. Learning rate is a value greater than 0 but less than 1, this is used so that the weights on the links do not change to quickly, or the ANN might never converge onto the optimal solution.

Linearly separable function is a function where if plotted in a n-dimensional plane, the negative and positive examples of the function can be totally separated using a straight plane across the space. Multi-layer Feed Forward Networks is a network with at least one unit that is not output or input, where the direction of data flow is in only one direction. Perceptrons is a network with no units that are output or input, where the direction of data flow is in only one direction. Supervised learning is known as all learning algorithms where the known targets are used to adjust the network.

Target is the expected output of the input. This is used to calculate the error. Threshold function is to decide whether a unit should fire or not. Typically 1 is exceeding the threshold and 0 or –1 otherwise. Units/Nodes are the simple elements of an ANN; they take in input from other nodes or training data, sum up the data and apply a threshold function to decide what output to send. Weighted links are Connects units together, conceptually shows the strength of the bond between two units.
The classifier employed in this thesis is a three-layer Back Propagation Neural Network hybrid with the Artificial Bee Colony Optimization Algorithm. The backpropagation neural network optimizes the net for correct responses to the training input data set. More than one hidden layer may be beneficial for some applications, but one hidden layer is sufficient if enough hidden neurons are used.

6.3 BACKPROPAGATION ALGORITHM

The most widely used neural-network learning method is the BP algorithm. Learning in a neural network involves modifying the weights and biases of the network in order to minimize a cost function. The cost function always includes an error term a measure of how close the network's predictions are to the class labels for the examples in the training set. Additionally, it may include a complexity term that reacts a prior distribution over the values that the parameters can take.

The backpropagation algorithm can be implemented in two different modes: on-line mode and batch mode. In the on-line mode the error function is calculated after the presentation of each input pattern and the error signal is propagated back through the network modifying the weights before the presentation of the next pattern. This error function is usually the Mean Square Error (MSE) of the difference between the desired and the actual responses of the network over all the output units. Then the new weights remain fixed and a new pattern is presented to the network and this process continuous until all the patterns have been presented to the network. The presentation of all the patterns is usually called one epoch or single iteration. In practice many epochs are needed before the error becomes acceptably small.
In the batch mode the error signal is calculated for each input pattern and the weights are modified ever time the input pattern has been presented. Then the error function is calculated as the sum of the individual MSE errors for each pattern and the weights are accordingly modified (all in a single step for all the patterns) before the next iteration. Single layer perceptron provided a powerful solution to the problems, which are linearly separable. Multi-layer Perceptrons were considered to be difficult for training. Error back propagation algorithm (backpropagation algorithm) was effective solution to train multi layer perceptrons based on error correction learning. In the forward pass outputs are computed and in the backward pass weights are updated/corrected based on the errors. The development of the back-propagation algorithm is a landmark in neural networks in that it provides a computationally efficient method for the training of multilayer perceptrons.

6.4 BPN HYBRID WITH ARTIFICIAL BEE COLONY OPTIMIZATION ALGORITHM (BPN-ABCO)

Initially the selected features from the feature selection algorithms are normalized between [0,1]. That is each value in the feature set is divided by the maximum value from the set. These normalized values are assigned to the input neurons. The number of hidden neurons is equal to the number of input neurons. And only one output neuron. Initial weights are extracted using the ABCO algorithm as follows:
In weight extraction, N random numbers are generated with d number of digits. Where, N is the total number of neurons in the BPN. The weights are extracted from the population of random numbers to determine the fitness values. The actual weight \( w_k \) is given by:

\[
   w_k = \left\{ \frac{c \cdot \left[ x_{kd+2} \cdot 10^{d-2} + x_{kd+3} \cdot 10^{d-3} + \ldots + x_{(k+1)d} \right]}{10^{d-2}} \right\}
\]

(6.1)

where \( c = 1 \), if \( 5 \leq x_{kd+1} \leq 9 \), else \( c = -1 \), and \( k \) represents the population. The weights are extracted for each string in the population. The fitness values is calculated as defined below:

\[
   F = \frac{1}{E},
\]

(6.2)

Where, \( E = \sqrt{\left( \frac{E_1 + E_2 + \ldots + E_m}{m} \right)} \). Where, \( m \) – is the total number of training patterns, and \( E_1, E_2, \ldots, E_m \) are the errors for each pattern, i.e., \( E_i = (T_i - O_i)^2 \), where \( T_i \) is the desired output, and \( O_i \) is the actual result of the output layer.

Thus the fitness value is calculated for a single population. Like this M populations are generated and their fitness values are calculated. The optimum fitness value is selected using ABCO algorithm. All the minimum fitness values are replaced with the maximum fitness values. Now the weights are updated with the new fitness value and the training is performed again. This procedure is repeated until the error from the backpropagation network is less than the tolerance value. Figure 6.1 show the flowchart for BPN-ABCO Classifier and the figure 6.2 shows the Algorithm for BPN-ABCO Classifier.
Extract the features from the segmented image

Normalize the feature values from 0 to 1

Assign the feature values to input neurons

Extract the weights using ABCO

Calculate the output from hidden (S1) and output (S2) neurons

Calculate the error

Update the weights using ABCO

Perform the steps till the target output is equal to the desired output.

Figure 6.1 Flowchart for BPN-ABCO Classifier
1. Repeat for M times
   
a. Generate N random numbers with d number of digits
   
b. Extract the weights from the random numbers as:
   
c. \( w_k = \left\{ c \times \left[ x_{kd+2} \times 10^{d-2} + x_{kd+3} \times 10^{d-3} + \ldots + x_{(k+1)d} \right] \right\} / 10^{d-2}, \)
   
d. where c=1, if \( 5 \leq x_{kd+1} \leq 9 \), else c=-1
   
e. Calculate the output at the hidden layer, \( S_1 = 1 / (1 + e^{-\lambda x}) \), where \( \lambda = 1 \), and \( x = \Sigma_i w_{ih} k_i \)
   
f. Calculate the output at the output layer, \( S_2 = 1 / (1 + e^{-\lambda x}) \), where \( \lambda = 1 \), and \( x = \Sigma_i w_{ho} S_i \)
   
g. Find the error, \( E_i = (T_i - O_i)^2 \)
   
h. Repeat the steps (e) – (g) for all the training patterns.
   
i. Calculate the Mean Square error, \( E = \sqrt{\left( E_1 + E_2 + \ldots + E_m \right) / m} \),
   
j. Calculate the fitness values for each weight \( F_k = 1 / E \)

2. \( P_i \leftarrow F_k \), initial population for ABCO algorithm

3. The optimum fitness value \( F_{max} \) is found out using ABCO algorithm

4. The minimum fitness values are replaced with \( F_{max} \).

5. Repeat the procedure of weight extraction and fitness value calculation for the updated weights.

6. Repeat the steps (1)-(6) till the error becomes less than the tolerance value.

---

Figure 6.2 Algorithm for BPN-ABCO Classifier
The output from the each hidden neuron is calculated using the sigmoid function, \( S_1 = \frac{1}{1+e^{-\lambda x}} \), where \( \lambda=1 \), and \( x = \sum w_{ih} k_i \), where \( w_{ih} \) is the weight assigned between input and hidden layer, and \( k \) is the input value. The output from the output layer is calculated using the sigmoid function, \( S_2 = \frac{1}{1+e^{-\lambda x}} \), where \( \lambda=1 \), and \( x = \sum w_{ho} S_i \), where \( w_{ho} \) is the weight assigned between hidden and output layer, and \( S_i \) is the output value from hidden neurons. \( S_2 \) is subtracted from the desired output. The network is trained to produce a 0.9 output value for positive ROI (malignant) and 0.1 output value for a negative ROI (benign). The classification performance was studied by the ten-fold validation method and the results were analyzed by using ROC analysis. Figure 6.3 shows the snapshot for mammogram image classification using BPN-ABCO.

![Figure 6.3: snapshot for the mammogram image classification using BPN-ABCO.](image-url)
6.5 SUMMARY

A three-layer Back Propagation Neural network is used for classification. The values of the features available in the reduced feature set, constructed from the feature selection algorithm is normalized and given as input to the classifier. For each testing image, the output is calculated using sigmoid function. The error is calculated between the actual output and the target output. Based on this error value the weights are propagated to reduce the error value.

Thus the classifier was trained to produce the output value 0.9 for malignant images, 0.5 output values for benign images and 0.1 for normal images. The performances of all possible combinations between bilateral subtraction and feature extraction methods were evaluated using this ROC analysis.