**CS549 (Topic in AI)**

**Final Report**

**On**

**Heart Disease Diagnosis Expert System**

**Submitted by**

**Bhushan Bamane (916-56-0463)**

# **Topic: Heart Disease Diagnosis Expert System using feed forward back propagation algorithm.**

# **Problem Definition:**

**What is the problem?**

The problem here is I am going to detect Heart Disease and its severity level using diagnosis attributes present in the dataset. To classify the input data into appropriate output i.e. whether a person has disease/not. I am using Feed-forward back propagation algorithm. The data set for training/testing is taken from UCI machine learning repository to diagnose the disease. After consideration of appropriate parameters/activation function while building up the neural network - I reached the classification accuracy for heart disease. The objective of this project is demonstrating the capabilities of Artificial Neural Network implementation in detection of Heart diseases from symptoms available.

**Background:**

An Artificial Neural Network is computational models that are inspired by the human nervous system that are capable of machine learning and pattern recognition. An ANN can also be defined as the inter-connected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation. An ANN is a powerful data modelling tool that can capture and represent complex input or output relationships. The traditional back propagation neural network algorithm is widely used in solving many practical problems. The back propagation neural network learns by calculating the errors of the output layer to find the errors in the hidden layers.

**What do you want to do?**

I want to build a system that detects the Heart disease on basis of the various symptoms of the person. The system should detect whether the person is healthy or unhealthy. First, I would like to train the ANN/system with the set of training data sets and then I would test the system with the Test data sets. The ANN I am building uses feed forwardback propagation algorithm to update the weights.

# Data Set:

**Describe the data set that you use?**

The data set used for our system is set of values of symptoms of an individual with his age. The datasets available are converted into 0’s and 1’s using the normalization process and thereafter provided to neural network to recognize and detect heart disease. To normalize the original input, I am using datasets available and using MIN-MAX normalization method. The number of attributes/symptoms taken for any individual gives the number of input units of the artificial neural network. The weights of the neuron are adjusted till the error rate decreases and these weights are updated using the feed forward back-propagation algorithm. There are five output units in within the datasets which determines the severity level of Heart disease of an individual which ranges from 0 to 4.

I have used following data sets:

<https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data>

**Describe the input and output data?**

**Input Data:** Inputs are the various attribute and tests results of an individual health. Below are shown thirteen set of attributes of each person available. These attributes values are converted into appropriate 0’s and 1’s using MIN-MAX normalization so that they can be applied as an actual input to the neural network.

**TABLE LISTING ALL ATTRIBUTES:**

|  |  |
| --- | --- |
| Diagnosis Attributes | Highest Value |
| Age | 77 |
| Sex  M =1, F = 0 | 1 |
| Chest pain:  --Value 1:typical angina  --Value 2: atypical anginal  --Value 3: non-anginal pain  --Value 4: asymptotic | 4 |
| Resting Blood Pressure | 200 |
| Serum Cholesterol (mg/dl) | 564 |
| Fasting Blood Sugar  1=true; 0=false | 1 |
| Resting Electrocardiographic Result  --Value 0: normal  --Value 1:having ST-T wave abnormality (T wave inversions and/or ST)  --Value 2:showing probable or definite left ventricular Hypertrophy by Estes’ criteria | 2 |
| Maximum Heart Rate Achieved | 202 |
| Exercise Induced Angina 1=yes;0=no | 1 |
| Old Peak | 6.2 |
| The slope of the peak Exercise ST segment  --Value 1: up sloping  --Value 2: flat  --Value 3:down sloping | 3 |
| Number of Major Vessels  (0-3) | 3 |
| Thal: 3 = normal; 6 = fixed defect; 7 = reversible defect | 7 |

**Output Data:**

Based on input attributes value, Neural network determines the severity level of heart disease.

Output classification is as follows:

|  |  |  |
| --- | --- | --- |
| Target Output | Multiclass classification | Heart disease type |
| 0 | 1   0    0    0     0 | Healthy |
| 1 | 0   1    0    0     0 | Unhealthy - 1 |
| 2 | 0   0    1    0     0 | Unhealthy - 2 |
| 3 | 0   0    0    1     0 | Unhealthy – 3 |
| 4 | 0   0    0    0     1 | Serious |

System Flow:

Divide UCI machine learning dataset into Training data set (90%) and testing data set (10%)

Normalization of training and testing data set

Train neural network using training data set, by applying feed forward back propagation algorithm.

Test neural network using test data.

Display neural network statistics, test data output with accuracy.

**Implementation:**

The system is developed in java platform.

Module details:

1)      **ReadFiles.java** – This module reads the given test file  and training file and stores the resultant attribute in matrix.

2)      **Normalization.java** – This module performs Min-Max normalization on input attribute obtain from module 1.

3)      **Backprapogation.java** – In this module I have actually implemented Feed Forward Backpropagation algorithm.

**Steps:**

**Step one**

1. Initialization of the connection Weights as small random numbers.
2. Feeding the input to the input units: *Inputp=(Ip1, Ip2, …, Ipi)*, where Inputp is the pth training example.
3. Computation of the net values (Iighted sum) of the hidden units  is the Iight from the input unit *i* to the hidden unit *j*.
4. Computation of the output values of the hidden units  is the output function of the hidden unit *j*.
5. Computation of the net values (Iighted sum) of the output units  is the Iight from the hidden unit *j* to the output unit *k*.
6. Computation of the output values of the output units  is the output function of the output unit *k*.

**Step two**

1. Computation of the error terms of the output units. The error is computed from the desired output (*Tpk*) and the actual output (*Opk*) of the corresponding output unit *k* and the corresponding example *p*. From this error and the derivative of the output function, *dpk* is computed by  When the output function is logistic function, 
2. Computation of the error terms of the hidden units. Because I don’t have target outputs for the hidden units, the estimated error of the hidden unit *j* is computed by 
3. Updating the connection Weights between the hidden units and the output units by  is the learning rate and *t* is a time step.
4. Updating the connection Weights between the input units and the hidden units by



4)      **Train.java** – Once the neural network is trained using training data set, test data is given as input to the neural network for testing.  Test data accuracy is calculated.

5)      **Home.java** - This user interface module gives all statistical details about the system performance.

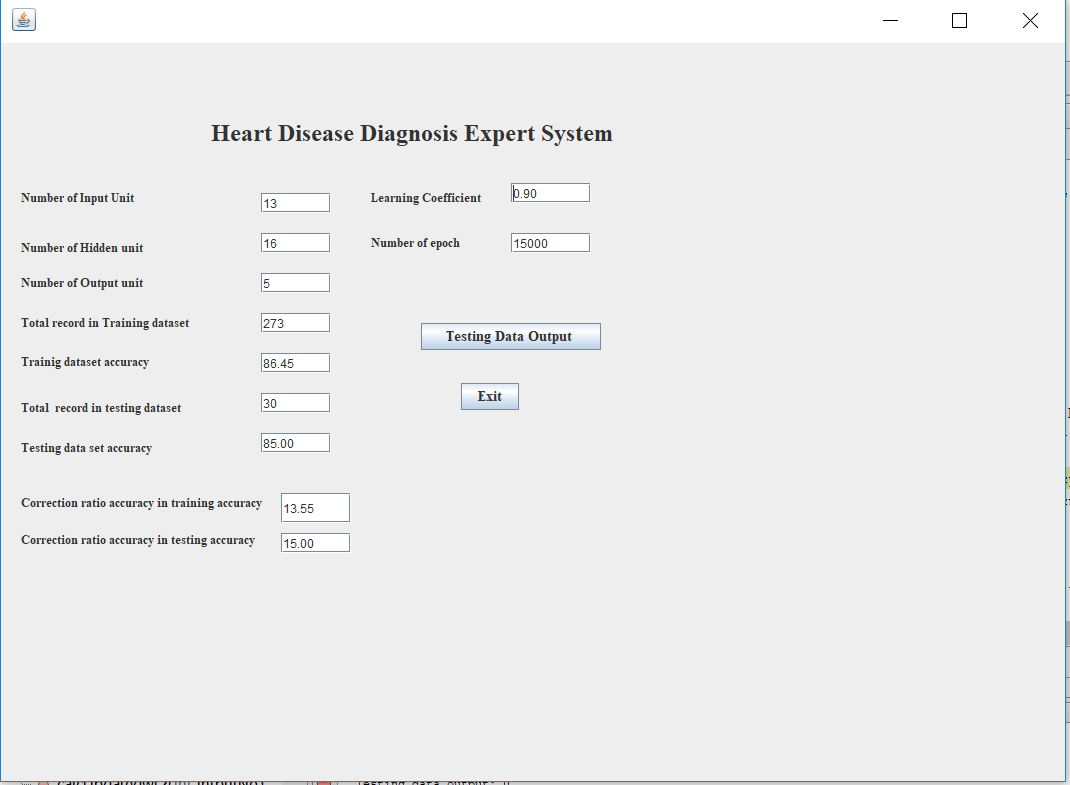
6)        **Result.java** – This user interface module represents all test data evaluation with all input attributes and resultant output.

**Experimental Results:**

**Result.java:**

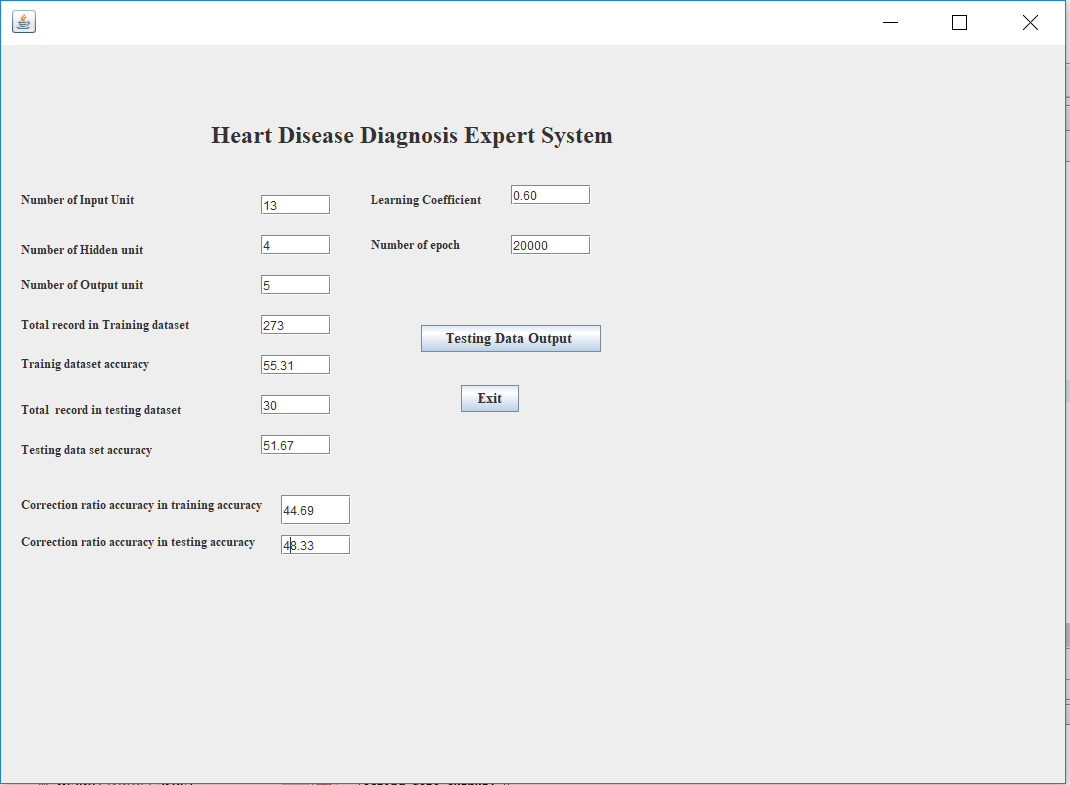
Case1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No of input Units | No of hidden Units | No of output Units | Total record in training  dataset | Total record in testing  dataset | No of epochs | Learning Coefficient | Accuracy of Training dataset | Accuracy of Testing dataset |
| 13 | 16 | 5 | 273 | 30 | 15000 | 0.9 | 86.45% | 85% |



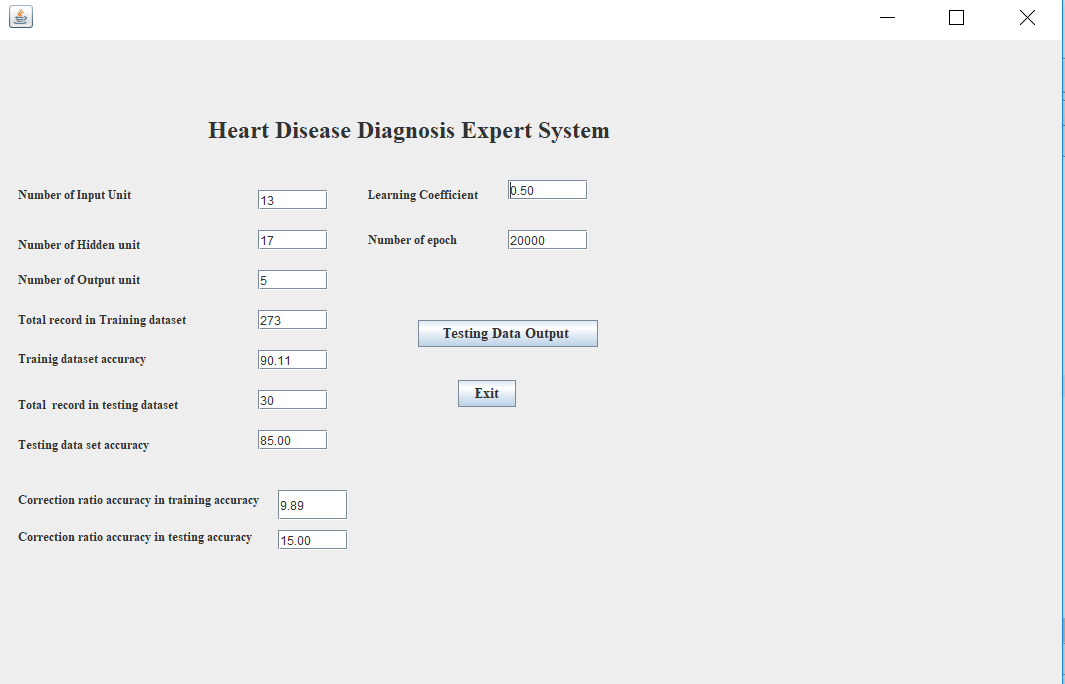
Case2:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No of input Units | No of hidden Units | No of output Units | Total record in training  dataset | Total record in testing  dataset | No of epochs | Learning Coefficient | Accuracy of Training dataset | Accuracy of Testing dataset |
| 13 | 4 | 5 | 273 | 30 | 20000 | 0.6 | 55.31% | 51.67% |



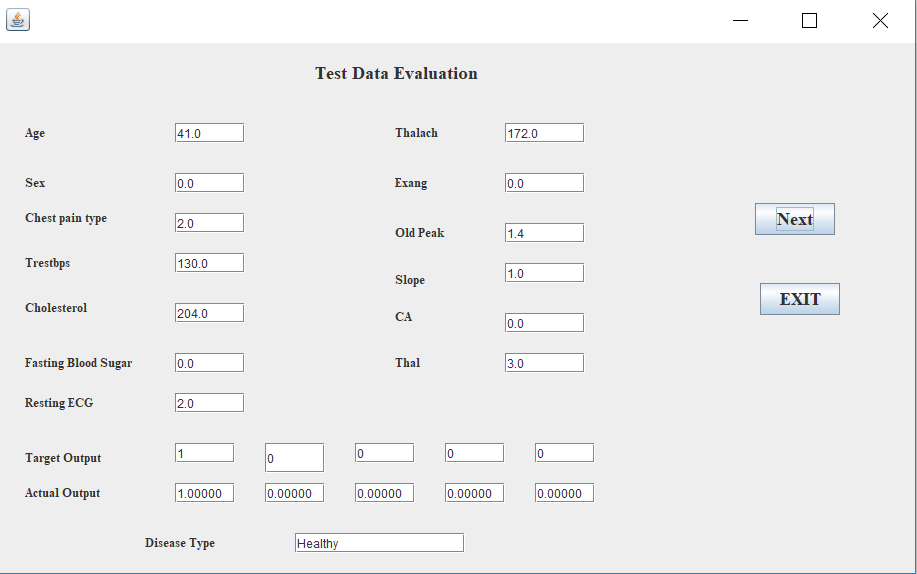
Case3:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No of input Units | No of hidden Units | No of output Units | Total record in training  dataset | Total record in testing  dataset | No of epochs | Learning Coefficient | Accuracy of Training dataset | Accuracy of Testing dataset |
| 13 | 17 | 5 | 273 | 30 | 20000 | 0.5 | 90.11% | 85% |



**Data Evaluation:**

This shows all the diagnosis attributes of test data sets available and gives the result achieved through the neural network.



**Conclusions:**

The system I have designed recognizes the Heart Disease. I have used feed forward back propagation learning algorithm to test the model based on their ability to diagnosis Heart Disease. By considering appropriate activation function for hidden layer and 17 neurons in hidden layer, I can reach to the classification accuracy for heart disease to 85%. Artificial Neural Network should significant results in heart disease diagnosis.

**Ideas for future work:**

Making this system web based where by importing files with proper data can get you the report online.

**References:**

* <https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data>
* <https://archive.ics.uci.edu/ml/datasets.html>
* Class Notes (For feed forward and back propagation algorithm logic).