

**Breast cancer Classification using Neural Networks**

Sanaullah Khan

14031251

Neural Network and Fuzzy System

Course work 1

Instructor Dr. Junaid Akhtar

Namal college Mianwali

Abstract:

This report is a written description of a system “breast cancer classifier” built on matlab tool using artificial neural networks. This system classifies the breast cancer in the form of benign and malignant.

Introduction:

Breast cancer is a deadly disease that develops from breast tissues. According to the U.S. Breast Cancer Statistics “About 1 in 8 U.S. women (about 12%) will develop invasive breast cancer over the course of her lifetime.”(1).

Now to avoid this disease from getting deadly is diagnosing it in the early stages. One way to diagnose early is making diagnosis easy. So how we can make diagnosis easy? The answer is we can use technology. We can use AI techniques to build algorithms that can detect breast cancer. So I am using artificial neural network to build a system that can mimic the natural way of diagnosing the breast cancer. Basically it is a classifier which classifies that either the patient is diagnosed with the benign form or the patient is diagnosed with the malignant form.

Background:

By using ANN (artificial neural networks) system has been made and research work has been done for the diagnosis of breast cancer. WILLIAM H. WOLBERG and OLVI L. MANGASARIAN in their research paper “Multisurface method of pattern separation for medical diagnosis applied to breast cytology” presents Multisurface pattern separation. It is a mathematical method for distinguishing between elements of two pattern set. Also their method is applicable to other medical diagnostic and decision making problems (2). Htet Thazin Tike Thein and Khin Mo Mo Tun in their research paper “AN APPROACH FOR BREAST CANCER DIAGNOSIS CLASSIFICATION USING NEURAL NETWORK” proposed an island based model to overcome the problem of local minima. They faced issues like longer training time and lower classification accuracy (3).

Work Methodology:

My work methodology is the following:

1. Firstly I replaced the commas “,” with spaces and converted the given breast cancer data into matrix by placing the brackets around it and assigned it to a variable named “a” and saved into a CW1.m file which is compatible for matlab.
2. The data contains 16 cases with missing attributes. As they are small portion of the overall data so I have ignored these cases by deleting them.
3. I have split the data into training and testing because I want to test and train the artificial neural network on different parts of the data as it is recommended to test the network on unseen data.
4. So I have divided the data by creating separate matrices for training and testing.
5. Two matrices are trainInputData and trainTargetData for the training of artificial neural network.
6. Two matrices are testInputData and testTargetData for the testing of artificial neural network.
7. I have used a built-in function of matlab “newff” to train the artificial neural network. This “newff” function creates a feed forward back propagation network.
8. I have passed the matrices which I created for training to the “newff” function to train the network.
9. In the arguments of “newff” function I have used the “tansig” as an activation function. Tansig function according to the documentations of matlab “converts an artificial neural networks layer input into layers output”. I have used “trainr” function as a training function. This trainr function according to the documentations of matlab “trains a network with weights and bias learning rules with incremental updates after each presentation of an input”. I have used “learngd” which is the gradient descent weight/bias learning function. Also I have used “mse” mean squared error performance function as an argument of “newff” function.
10. I have tested the “testinputData” matrix which I created earlier for the testing and compared the predicted results with the “testTargetData” matrix which I also created earlier and which is the matrix of actual results from the given data. In the end I have calculated the accuracy that is how many time the system have predicted the actual output by the following formula:

(No of correctly predicted values\*100)/size of “testTargetData”

Experimental Results and Analysis:

I have made some hypothesis and tested them. Their results and analysis are following:

Hypothesis#1):

If I split the data for training and testing in such a way that the training data is major portion of the overall data and testing is a small portion of overall data then I can achieve the higher accuracy. This is because when artificial neural network will train on major part of the data it will cover majority of the cases and variations and its learning will be better. So in this way higher accuracy can be achieved.

Following are the experiments and results based on this hypothesis

1. Activation function == “tansig”
2. Training function == “trainr”
3. Learning function == “learngd”

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Data** | **Testing Data** | **No of neurons** | **Accuracy** |
| 30 % | 70 % | 20 | 95.824 |
| 50 % | 50 % | 20 | 97.3684 |
| 70 % | 30 % | 20 | 98.5366 |
| 80 % | 20 % | 20 | 98.5401 |

Analysis:

This hypothesis was truly validated by above experiments and its results. This is because when you train the network on major part of the whole data it covers majority of the cases and it learns more effectively with more cases. So training the network with higher ration of data can achieve higher accuracy.

Hypothesis#2):

As the no of neurons used in hidden layer plays an important role in achieving the goal and minimizing the error. So by changing the no of neurons can affect the rate of accuracy.

Following are the experiments and results based on this hypothesis

1. Activation function == “tansig”
2. Training function == “trainr”
3. Learning function == “learngd”

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Data** | **Testing Data** | **No. of neurons** | **Accuracy** |
| 80 % | 20 % | 3 | 95.6204 |
| 80 % | 20 % | 5 | 97.0803 |
| 80 % | 20 % | 10 | 98.5401 |
| 80 % | 20 % | 15 | 98.5401 |
| 80 % | 20 % | 20 | 98.5401 |
| 80 % | 20 % | 25 | 98.5401 |

Analysis:

Above experiments and results tell us that neurons after a certain limit does not affect the rate of accuracy.

Hypothesis#3):

Hidden layer means a cut that classifies the outputs. So the general conception is by increasing the hidden layers will achieve the higher accuracy

Following are the experiments and results based on this hypothesis

1. Activation function == “tansig”
2. Training function == “trainr”
3. Learning function == “learngd”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training Data** | **Testing Data** | **No. of neurons** | **Hidden Layers** | **Accuracy** |
| 80 % | 20 % | 20 | 1 | 98.5401 |
| 80 % | 20 % | 20 | 2 | 98.5401 |
| 80 % | 20 % | 20 | 3 | 97.8102 |
| 80 % | 20 % | 20 | 4 | 89.7810 |

Analysis:

By increasing the hidden layers instead of achieving the higher accuracy it lowers the accuracy rate. Because only one hidden layer is enough for the linearly separable data. If the data is linearly separable and you are increasing the hidden layers then it will make things more complicated and will lower the accuracy.

Conclusion and Future Work:

After doing above experiments I have concluded that if you will increase the ratio of target data against target data, it will increase the rate of accuracy. Also I have observed that by increasing the no of neurons increase the rate of accuracy but after a certain limit there is no change in the accuracy. This is because you can achieve the accuracy up to a certain limit, so after that by increasing the no of neurons may lower your accuracy rate but it cannot increase your accuracy rate. Also I have concluded that this data is linearly separable because one hidden layer is enough as one layer achieves the higher accuracy rate. By increasing the hidden layers accuracy rate has been lowered.

Here I have only focused on three factors that are data distribution, no of neurons and no of hidden layers. But in future we can explore other factors like using a different neural network and then using different activation functions. Similarly for training and learning different functions can be used.

References:

1) <http://www.breastcancer.org/symptoms/understand_bc/statistics>

2) Wolberg, W. and Mangasarian, O. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87(23), pp.9193-9196.

3) Tike Thein, H. and Mo Tun, K. (2015). An Approach for Breast Cancer Diagnosis Classification Using Neural Network. *Advanced Computing: An International Journal*, 6(1), pp.1-11.