Breast Cancer Diagnoses Using Neural Network

By
Muhammad Adnan
14031260

Contents

Abstract:

The ability of a software to self-direct itself was the idea that led the world to think of 'autonomous systems'. The sole purpose of developing such systems was to make the software self-operative and self-dependent so as to minimize or remove human interaction. Identification of breast cancer is an open study area which can be solved by such technologies. One of the approaches to solving this is using Artificial Neural Networks as It is one of the most intelligent tools designed for automation. The network is usually fed with a predefined amount of training data using which the network familiarizes itself with the problem and its possible solutions and when it is used to operate on a different set of data, it tries to give the best possible result from the previously learned data. Different types of neural networks have been used and the best result on a new case has been near 93% to 98% accuracy.

Introduction:

Breast Cancer is one of the most dangerous diseases in the world and its been increasing day by day. As in United Kingdom, every one woman out of eight has this disease. Previously it was find in those women who was above 50 but recently there were some cases in which its detected that some younger women have also attacked by it. Keeping this in mind, most of the good researches have given some best solutions and one of them is Neural Network.

Here, I am trying to explain how does Neural Network work for this issue, solution of this classification issue and I will test and analyze different hypothesis regarding the accuracy of solution.

Background:

Neural Network is created by the inspiration of human mind. It consists of three parts.

- Input
- Hidden Layers(Neurons)
- Output

Different values are given to an Input and these inputs are given to transfer functions to output some numerical values. It is based on connected layers called neurons. Different number of Neurons are used together to produce different type of results. Neural network used different type of algorithms to make a network like **traingd**, **transig** etc. It uses different types of training Networks like I will use feedforward network which is as follows:

Inputs Layer ⟶ Hidden Layers ⟶ Output Layer

- # Main Part:

1. Data Gathering:

Collected data from Wisconsin Diagnostic Breast Cancer Website and saved it as Data.m. The data which is being used here consists of 699 rows and 11 columns while 1st column is for Id's and last one is telling that either the person of respective id has the symptoms of Breast Cancer or not. If there is 2 in last column means Benign while if 4 then its Malignant. There were some values missing and I replaced these values with '0'. Then I used this file in another file named CW.m where I am doing my code.

2. Preprocessing:

Firstly ignored first and last column and stored remaining data into a variable named Input. Separated last column and saved in a variable named output. Then took the mean of input and saved these values in the place of '0' which I replaced with '?'. After that I separated Benign and Malignant data. Then divided Benign and Malignant data into 70, 70 percent which I will train and remaining 30 percent of data will be tested on trained network.

3. Training and Testing Network:

Here I made I separated train and test data of benign and malignant. I separately made benign training and testing input as well as output and did same for malignant. After that I combined both benign and malignant data using **vertcat**, which concatenate the values of these both variables. Then I made a network using **feedforwardnet** where I am providing two hidden layers and an activation function **traingd**.

After that, I trained that data using **train** which is taking a network which I prepared above, an input and a target output.

4. Post Processing:

I checked the accuracy of my testing data by using **for** loop where I am taking value from 1 to the end value and compared the output of testing data with the result data after training data. Here I used a count variable which actually is starting by '0' and increase count value '1' whenever finds the same value in testing and resulting data. Then I checked the percentage accuracy of the training data by using count value divided by length of the output of testing data and then multiplying it by 100.

- # Experimental Results and Analysis:

After designing a working a Neural Network, I did different hypothesis to test, analyze and understand the working of my Neural Network and its effects of different parameters on results. Some are stated as follows:

1. Hypothesis No.1: Ratio Changes effects on Testing and Training Data:

Here I changed the Training and Testing Percentage of data and run my Network. It was found that as the percentage of training data increases the accuracy of the Network

increases while the best ratio is when we use 30% data for testing and 70% data for training because it gives us most accurate results. Results that are produced by my Neural Network with different ratio are as follows:

1.1. Results:

Serial No.	Training Data%	Testing Data%	Accuracy%
1	10	90	83.13
2	20	80	88.19
3	30	70	91.22
4	40	60	93.09
5	50	50	94.26
6	60	40	91.39
7	70	30	93.30
8	80	20	95.71
9	90	10	91.42

1.2. Analysis:

I have trained the network with different values, iterating from 10 to 90 percent. As much data I'm training, results are getting better. But the ideal position is 70 percent training and 30 percent testing because as much as training data is increasing accuracy is getting better. But if we increase training data more than 70 percent, the essence of Neural Network gone.

2. Hypothesis No.2: Number of Neurons on 1 Layer:

Increasing the Number of Neurons will increase the accuracy. Because as if we see there is a task for one man and he completes in 10 days but if we increase the number of men, definitely that task will be easy and will complete early. Same thing happens here if we increase the number of neurons the accuracy of Neural Network will increase and vice versa.

2.1. Results:

Serial No.	Number of Neurons	Accuracy%
1	10	88.51
2	20	90.90
3	30	89.95
4	40	87.55
5	50	86.12

6	60	87.08

2.2. Analysis:

The above results are not as clear as sometimes accuracy is increasing and then decreasing but whenever we increase Neurons the time of training data changes, as by increasing number of Neurons will take much time to complete than lower number of neurons.

3. Hypothesis No.3: Different Training Network (newff):

'newff' takes an input, target output, Number of neurons and some activation functions. I checked it on my network and accuracy results were following:

3.1. Results:

Serial No.	Number of Neurons	Accuracy%
1	10	97.12
2	20	98.56
3	30	97.60
4	40	98.08
5	50	97.60
6	60	96.65

3.2. Analysis:

This training function takes a lot time to train but its accuracy is higher than 'feedforwardnet'. Here error percentage is low. Here what I have noticed is that if the number of neurons is lower network will train slowly but when the number of neurons will be higher it will be trained a lit bit faster than lowers.

4. Hypothesis No.4: Changing the Number of Hidden Layers:

Here I am going to run my Network on same input and I will change the number of hidden layers and will notice the accuracy of the network. My training data here will be 70% and testing data will be 30%. My training network is feedforward network.

Serial No.	Number of Neurons	Accuracy%
1	2(5,5)	92.34
2	2(5,10)	95.69
3	2(10,10)	96.17

4.1. Results:

4	3(5,5,5)	96.65
5	3(5,5,10)	95.21
6	3(5,10,10)	97.60

4.2. Analysis:

As the number of hidden layers increases the data trains so fast and the accuracy increases.

Conclusion:

The findings, in this report, are in light of above mentioned hypotheses and their test and few other experiments that were done during the lab sections. Neural networks are being used by different people to solve this problem and many other, each of them has his/her own results, most of the times, different from others. It's because, there is a long list of parameters, and different nature of different problems. What I learned from this whole exercise is, how neural networks work and how this technique can be applied to different problems. Fine data distribution, like 70-30 percentage ratio, in this problem, produces good results. Lower learning rate produces good results but if it is too lower, then performance issues will occur. Same goes for the number of neurons in hidden layer. The other point that I learned during other tests, not mentioned in this report, is that uniform data as input produces more accurate results. However, many test can be executed regarding different parameters and provided data, and can be figured out the best answers.

References:

http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29