**BREAST CANCER DIAGNOSIS**

Using NEURAL NETWORKs

Muhammad Jaleel Course Instructor: Dr. Junaid Akhtar

UOB: 13031155

# Introduction

Breast cancer is a widely spreading disease among women. According to different surveys, one out of eight women in the world is affected by breast cancer in her life[1][2]. The essence behind the each department of science is to solve different problems. So, many approaches have been made to diagnose this disease[2]. The methodology, being used by computer scientists, is by training neural network on available data and testing it on known results to measure, this method’s accuracy. This solution classifies whether a patient having such and such symptoms is affected by this disease or not. The solution is very elegant and understandable, and is based simple mathematical functions. In this report, we will have a brief overview of neural networks, solution of this classification problem, and we will test and analyze different hypotheses relating to performance and accuracy of provided solution.

# Background

Using neural networks to solve different problems is different approach than other computational methods. Neural networks are created by the inspiration of human mind. The building block of a neural network is a neuron as human mind has, but this neuron is rather silicon based not biological. A neuron takes one or several inputs and one output that models certain biological properties. Different weights are assigned to inputs, these inputs with their weights are given to transfer function which outputs a numerical value. The output is positive if the value is greater than a specific threshold assigned to this neuron otherwise the value is negative zero or negative depending upon the nature of transfer function. There is a long list of such transfer functions which provide different results on same input data set. The selection of a transfer function depends upon the nature of problem, input data and output results. For example, a transfer function named “tansig” provides a bipolar out ranging in between -1 and 1. There are many other functions that produce outputs in different ranges.

Different number of neurons are used together to produce better results. There can be one or many neuron layers in a single neural network. So the simple structure is input layer, hidden layers and output layer. If the input is provided to input layer and then to hidden layer and then to output layer, such single directional neural network is termed as “Feed Forward Neural Network”. Such a neural network is being used in this solution. However, as human brain can learn from existing knowledge, therefore, there is another type of neural network known as “Feed Back Neural Network” in which there are loops, means next layer neuron’s outputs are provided to previous layer neurons as inputs. Such configurations provide higher accuracy but low performance with respect to time and resources.

Different people related to computer science have tried to provide a better solution[3]. Many of them used the dataset provided by Wisconsin University known as Wisconsin Diagnostic Breast Cancer (WBDC) dataset.

# Main Part

In this section, we will present the solution step by step, from data gathering to training a neural network, obtaining and compiling results and measuring the solution’s accuracy.

## Data Gathering

The dataset being used in this exercise is available at UCI Machine Learning dataset repository[4]. Data file contains 11 columns and 699 rows. First column represents case id that is of no use for us. Next 9 columns provide 9 different symptoms, and the value of each symptom is ranging from 1 to 10. Last column tells that the patient having these 9 symptoms of these values, is benign or malignant (2 for benign and 4 for malignant). Some the values in 6th column are missing and there is ‘?’ on those places.

## Prepressing of Data

First of all, missing values in the data set were manually replaced with zeros, so that all data should have same format. The file was loaded into a matrix in Matlab. Here the data is divided into two parts, input data and target data. First column is ignored because it’s just an id, 2-10 columns are marked as input data and 11th column is saved as target data. The missing values that were replaced by zeros, are now replaced with ***mean*** *(a mathematical function)* of that column, because letting zeros means misrepresentation of data. As we are going to use “tansig” transfer function in initial settings of the neural network, which outputs in range of -1 to 1, so we replaced 4 with 1 and 2 with -1 in target data. After these settings, we divided that input data and target into two other divisions, named as training input data, training target data and testing input data, testing target data. Initially the division percentage is fifty-fifty. Training input and target data is used in training of neural network and testing input and target data is used for testing and measuring the accuracy of results produced by the neural network.

## Creating and Training a Neural Network

In initial settings, a neural network is created by ***newff*** and is stored in a variable called ***net***. Basic settings provided to ***newff*** are following:

* Both transfer functions are ***tansig***
* Number of neurons are ***30***
* Training function is ***trainr***
* Learning function is ***learngd***
* Goal is set to 0.01
* Number of iterations is set to 100
* Number of validation checks is set to 10

The ***net*** object is then passed to ***train*** function along with training input data and training target data, and resulting trained network is stored in a variable called ***trainedNet***. By default, random weights will selected for inputs in initial iteration. After the network is trained, testing input data is passed to trained network and results are saved in a variable.

## Post Processing

After getting results from neural network, it’s time to measure the accuracy of our neural network. Each of the result is compared with the actual result and in case of wrong result error count is maintained. After comparing all results, percentage accuracy is obtained by the following simple formula:

*% accuracy= (error count / total results)\*100;*

# Experimental Results and Analysis

After setting up a working neural network, different hypotheses are made, tested and analyzed to know and understand the working of neural network and effect of different parameters on results. Few hypotheses are described below:

## Ratio among training data and testing data

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. | Training data (%) | Testing data (%) | Accuracy (%) |
| 1 | 10 | 90 | 97.45 |
| 2 | 20 | 80 | 96.23 |
| 3 | 30 | 70 | 95.90 |
| 4 | 40 | 60 | 97.84 |
| 5 | 50 | 50 | 97.98 |
| 6 | 60 | 40 | 98.20 |
| 7 | 70 | 30 | 99.04 |
| 8 | 80 | 20 | 99.28 |
| 9 | 90 | 10 | 98.55 |

If percentage of training data is higher than testing data, neural network will produce more accurate results. More precisely, 70% training data and 30% testing data will produce more accurate results. Now let’s see these results:

As random weights are assigned to inputs in initial iteration of an individual case, so, sometimes, result better and sometimes not. This can be seen in above table, 10% training data is producing better result than 20% training data. However, produced results are expected results except the first three and last one. As training data is increasing, accuracy is increasing. It’s because, higher the training data, lesser the unseen data to test so higher the accuracy and vice versa. Ideally 70% training data and 30% testing data is fine, because if we increase training data more than that, then the essence of neural networks is gone and may be It will not so practical in life.

## Effect of learning rate

Learning rate represents the delta change in graph of sigmoid functions, higher learning rate means higher jump in that graph. So, I think, lower the learning rate higher the accuracy will be and vice versa.

|  |  |  |
| --- | --- | --- |
| Sr. | Learning Rate | Accuracy (%) |
| 1 | 0.2 | 98.08 |
| 2 | 0.1 | 97.61 |
| 3 | 0.05 | 99.04 |
| 4 | 0.03 | 99.17 |
| 5 | 0.01 | 99.08 |

Results are not very clear, but still it can be seen that lower learning rate is increasing our accuracy but it takes more time to produce results. It could be because of lower jumps on curve and more calculations are responsible for such results.

## Number of neurons in hidden layer

Number neurons in hidden layer have crucial role on the resulting output. So, higher the number of neurons in hidden layer, higher the accuracy and vice versa.

|  |  |  |
| --- | --- | --- |
| Sr. | No. of neurons in hidden layer | Accuracy (%) |
| 1 | 10 | 99.04 |
| 2 | 20 | 98.08 |
| 3 | 30 | 97.12 |
| 4 | 40 | 96.17 |
| 5 | 50 | 98.08 |

The results shown above in table are not verifying our hypotheses. The noticeable thing during the execution was time. On higher number of neurons took more time in execution. I executed these experiments more than 5 times the results were more are less similar to the results shown in the table. Maybe, this parameter depends upon the nature of problems. Through hit and trial method, we can figure out the best number for our problem.

Apart from these hypotheses, many other hypotheses have been made and tested, and all of them can’t be explained here in this report.

# 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:

[1]: Breast Cancer: MedlinePlus. 2016. Breast Cancer: MedlinePlus. [ONLINE] Available at: https://medlineplus.gov/breastcancer.html. [Accessed 1 December 2017].

[2]:(<http://www.wcrf.org/int/research-we-fund/continuous-update-project-findings-reports/breast-cancer>).

[3]: Prasetyo, C., Kardiana, A., & Yuliwulandari, R. (2014). Breast Cancer Diagnosis using Artificial Neural Networks with Extreme Learning Techniques. International Journal of Advanced Research in Artificial Intelligence, 3(7). <http://doi.org/10.14569/ijarai.2014.030703>

[4]:(<http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>) [Accessed 21 November 2017]