Bilal Ahmad Khan

Neural Networks are inspired by human brains. Like brain, they have ability to learn and infer the results based on previous trained data. Neural networks gets train on certain input and when an unseen input is passed to these trained networks they judge the output of new input based on previous training. The neural network has become a popular tool in classification of cancer datasets. (Thazin, January2015)

Breast cancer classification using Neural Network Approach

11/20/2017

**Contents**

[**1.** **Introduction** 2](#_Toc500066262)

[**2** **Background** 2](#_Toc500066263)

[2.1 Breast Cancer Dataset: 2](#_Toc500066264)

[2.2 Overview of an artificial neuron: 2](#_Toc500066265)

[2.3 Neural network used: 2](#_Toc500066266)

[2.4 The neural net: 2](#_Toc500066267)

[2.5 Hidden layers and neurons in hidden layers: 2](#_Toc500066268)

[2.6 Training of neural network: 2](#_Toc500066269)

[**3** **Main Part** 3](#_Toc500066270)

[3.1 Preprocessing 3](#_Toc500066271)

[3.1.1 Data Normalization: 3](#_Toc500066272)

[3.2 Main Processing: 3](#_Toc500066273)

[3.2.1 Input and output datasets: 3](#_Toc500066274)

[3.2.2 Creation of network: 4](#_Toc500066275)

[3.2.3 Training the neural network: 4](#_Toc500066276)

[3.2.4 Simulating the trained network: 4](#_Toc500066277)

[**4** **Experimental Results and Analysis** 5](#_Toc500066278)

[4.1 Different ratio of dataset 5](#_Toc500066279)

[4.1.1 Hypothesis: 5](#_Toc500066280)

[4.1.2 Results: 5](#_Toc500066281)

[4.1.3 Analysis: 5](#_Toc500066282)

[4.2 Different activation functions 6](#_Toc500066283)

[4.2.1 Hypothesis: 6](#_Toc500066284)

[4.2.2 Results: 6](#_Toc500066285)

[4.2.3 Analysis: 6](#_Toc500066286)

[4.3 Number of neurons in the hidden layer 6](#_Toc500066287)

[4.3.1 Hypothesis: 6](#_Toc500066288)

[4.3.2 Results: 7](#_Toc500066289)

[4.3.3 Analysis: 7](#_Toc500066290)

[4.4 Conclusion 7](#_Toc500066291)

[**5** **Bibliography and Citations** 7](#_Toc500066292)

# **Introduction**

Breast cancer is a disease which can prove to be incurable and particularly affects women around the globe. There are many different methods to diagnose this disease and Neural Network methodology is one of them. The neural network was invented by neurophysiologist Warren McCulloch and mathematician Walter Pitts in 1943.(“History of Neural Networks Á Pattern recognition Á,” 2015) Neural networks although compute the solution to the problem but their approach is different from traditional computers. Computers work on algorithm approach but neural networks work in the way of human brain works.(Eksteen, n.d.) Due to this approach neural networks can do multiple tasks in a way as humans do.

# **Background**

## Breast Cancer Dataset:

The Publically available database provided by University of Wisconsin hospital, Madison from Dr. William H. Wolberg has been used for the detection of breast cancer using neural networks. There are total 699 instances and 10 attributes plus class attribute. Each instance has one of two possible classes: malignant or benign. According to the class distribution, 34.5% cells are malignant and 65.5% cells are benign.

## Overview of an artificial neuron:

Artificial neurons are the elementary units in an artificial neural network.it is a mathematical model of biological neurons. A neuron takes certain inputs which are multiplied by certain weights (weights could be random or controlled). Then activation function works on the output and plots particular output against certain neuron.

## Neural network used:

Feed-forward back propagation network has been used in the classification of breast cancer.

## The neural net:

A neural net is created which is particularly designed to get certain output with the help of activation functions.

## Hidden layers and neurons in hidden layers:

The neural net created could have multiple hidden layers and multiple neurons in hidden layers. These layers affect the results produced by neural network.

## Training of neural network:

After the creation of the neural network, it is trained on a certain portion of data. In our case, it will be trained on a certain portion of breast cancer data and will be tested on another portion of data to produce results.

# **Main Part**

In this section, I have described the preprocessing, adopted methodology, system architecture and implementation technique to solve this problem.

## Preprocessing

### Data Normalization:

In neural networks, data normalization is one of the important steps. (Thazin, January 2015) Breast cancer dataset was provided in the form of a text file. First of all the data inside the text file has been enclosed in square brackets for array construction and concatenation and capturing values returned by a function. Then the text file was converted into M-file so that we could access the given dataset in MATLAB. There were question marks in the dataset which were replaced by zeros because the (?) Operator works only with a class name, not an object. Finally, all the data (699x11 metrics) has been accessed through MATLAB command window.

## Main Processing:

There were eleven attributes in our dataset in which the first column consists of ID’s and the class attributes were in the last column (11th). To achieve the desired output, I have followed the principle steps for designing my neural network.

### Input and output datasets:

* Input data was separated which comprised of second column to second last column and six hundred and ninety-nine rows.
* Output data covered last column and six hundred and ninety-nine rows.
* It is observed that all the zeros belong to column seven, so we replaced zeros with the mean of column seven because it effects accuracy.
* In the output data, I replaced 4’s and 2’s with 1’s and 0’s respectively because I had to use ‘tansig’ and ‘purelin’ activation functions and they work in the range of -1 to 1.
* Code:

function[inputdata,targetdata,testinput,testtarget] = myProcess(data)

data(data==0) = round(mean(data(:,7)));

inputdata = data(1:490,2:10);

targetdata = data(1:490,11);

testinput = data(491:699,2:10);

testtarget = data(491:699,11);

targetdata(targetdata==4) = 1;

targetdata(targetdata==2) = 0;

testtarget(testtarget==4) = 1;

testtarget(testtarget==2) = 0;

end

### Creation of network:

A neural network has been created using newff (feed-forward backpropagation function). It creates a network with dialog a box and returns an N layer feed-forward backprop network. This network consists of N1 layers. Weights of the first layer are coming from the input and each subsequent layer has weights coming from the previous layer. The last layer is the network output.

Code:

function[net] = myNet(inputdata,targetdata)

net = newff(inputdata',targetdata', 20,{'tansig' 'tansig'},'trainlm','learngd','mse');

net.trainparam.epochs = 300;

net.trainparam.max\_fail = 100;

net.trainparam.lr = 0.02;

net.trainparam.goal = 0.01;

end

### Training the neural network:

Finally the created network (net), extracted input data (inputdata) and the output data (targetdata) is passed to our train function and now we could train our network accordingly.

Code:

function[net] = trainNet(net,inputdata,targetdata)

net = train(net,inputdata',targetdata');

end

### Simulating the trained network:

The function simulates on input data and calculates accuracy in terms of percentage.

Code:

function percentage = simulation(trainedNet,testinput,testtarget)

roundit = round(sim(trainedNet,testinput'));

checkarray = roundit' == testtarget;

ones = find(checkarray == 1);

countones = length(ones);

percentage = countones/length(testtarget)\*100;

fprintf('Accuracy = %i%', round(percentage));

fprintf(' percent');

fprintf('\n');

end

# **Experimental Results and Analysis**

## Different ratio of dataset

Changing the ratio of the dataset for training and testing the neural network will surely affect the accuracy.

### Hypothesis:

When the dataset for training the network increase its accuracy also increases. Following are the accuracies found against the particular ratio of datasets.

|  |  |  |
| --- | --- | --- |
| **Training Data (%)** | **Testing Data (%)** | **Accuracy (%)** |
| 30% | 70% | 96.50% |
| 40% | 60% | 96.42% |
| 50% | 50% | 96.84% |
| 60% | 40% | 97.85% |
| 70% | 30% | 98.57% |
| 80% | 20% | 99.28% |

### Results:

Hence prove that when we increase the ratio of the dataset for training our network its accuracy also increases but in initial three cases accuracies are unclear.so we could make another hypothesis here:

### Analysis:

There could be no equal distribution of malignant cells and benign cells in an initial 30% and 40 % of the dataset. Let’s find the percentage of malignant cells (represented by ‘1’) in the total dataset.

length (find(targetdata == 1))/length(targetdata)\*100

ans = 34.3840

Finding the percentages of malignant cells in the first 30% and 40% of the dataset respectively.

length(find(targetdata == 1))/length(targetdata)\*100

ans = 41.9048

length(find(targetdata == 1))/length(targetdata)\*100

ans = 45

It can be clearly seen that there is no equal distribution of malignant and benign cells in an initial 30% and 40% of the dataset and other than that there could be some data which would not be properly learned, so both above-mentioned facts could affect our accuracy.

## Different activation functions

Changing the activation functions of hidden layer and output layer to see its impact on our accuracy. To achieve these different cases my neural network has been trained on 70% of the dataset and tested on remaining 30% of the dataset.

### Hypothesis:

Different activation functions should deliver output in a different way, keeping the other factors constant.

|  |  |  |
| --- | --- | --- |
| **Activation Function** | **Network Training Function** | **Accuracy (%)** |
| purelin, purelin | Trainlm | 99.52% |
| purelin, tansig | Trainlm | 98.08% |
| tansig, tansig | Trainlm | 97.12% |
| tansig, purelin | Trainlm | 97.60% |

### Results:

Results have shown that activation functions have great effect on accuracy. As we could see that purelin function has produced better results comparatively.

### Analysis:

We know the behavior of an artificial neural network depends on both transfer functions and weights. Purelin is a linear unit and it’s provided output activity is proportional to the total weighted output while tansig is a sigmoid unit and its output varies continuously but not linearly as input changes. (Doc.ic.ac.uk, 2017) During network training each node has given a transfer function to judge its inputs, output nodes produce a result and after comparing results and desired output our network decides take the action again but with better learning. So when different activation functions are applied we got different results.

## Number of neurons in the hidden layer

What will be the effect of a different number of neurons in our neural network? To answer this question lets go for an experiment in which ‘tansig’ and ‘tansig’ have been used as activation functions for both hidden and output layer. Similarly, dataset ratio for training and testing is 70% and 30% respectively.

### Hypothesis:

If there are a number of neurons in the hidden layer then our goal will be reached and in case of less number of neurons we will not reach our goal.

|  |  |  |
| --- | --- | --- |
| Number of neurons | Accuracy (%) | Goal |
| 1 | 99.52% | Not reached |
| 2 | 99.52% | Not reached |
| 4 | 96.65% | Reached |
| 8 | 97.60% | Reached |
| 15 | 98.56% | Reached |
| 30 | 98.56% | Reached |

### Results:

Hence we could experimentally see that when we decreased the number of neurons to certain level we could not reach our goal but when we increased the number of neurons from that certain level we reached our goal.

### Analysis:

It’s because neurons are a vital entity in the learning process as they take particular inputs and produce outputs which then compared with desired outputs and their difference is noted. Again neurons assign better suited different connection weights to certain nodes to produce better results and they repeat this process until the goal is achieved. So we conclude that a number of neurons help neural networks to learn more effectively.

## Conclusion

We have seen that neural networks are helping a human being in every part of life and it is very easy to get desired results using neural networks but with proper study and understanding of neural networks we could get more fruitful outputs. Here we have discussed a particular real-life problem (breast cancer classification) and we observed that neural network allowed us to test different cases that would affect our accuracy. Hence neural networks brought great easiness for us and made our life easy.

# **Bibliography and Citations**

Thazin, H., 2015. AN APPROACH FOR BREAST CANCER DIAGNOSIS. *Advanced Computing: An International Journal (ACIJ),,* 6(1), pp. 1-11.

Eksteen, S. (n.d.). the Use of Artificial Neural Networks To Assist Gis in Decision Making : a Case Study – Predicting the Occurrence of Culicoides Causing African. *Training*.

History of Neural Networks Á Pattern recognition Á. (2015), 13–16. https://doi.org/10.1007/978-94-017-9816-7

Doc.ic.ac.uk. (2017). *Neural Networks*. [Online] Available at: http://www.doc.ic.ac.uk/~nd/surprise\_96/journal/vol4/cs11/report.html#Appendix A - Historical background in detail [Accessed 2 Dec. 2017].