## Neural Network and Fuzzy Systems
## Course Work 1

# Breast Cancer Detection Using Neural Network



**Muhammad Tayyab UOB (14031218)**
The Department of Computer Science. Namal College Mianwali
(An Associate College of University of Bradford)

December 3, 2017

# Contents



## 0.1 Introduction

Breast cancer is the 23 percent of all the cancers which have been diagnosed in the women.

Breast cancer which is also called a localized diseased is currently affecting more than 61 percent of the people having first stage cancer. Approximately one in eight women will die because of cancer. Even though the survival rates have been increased but clinicians on the other hand need the proper tools to classify risk for cancer patients. A number of approaches have been used to detect the risk of breast cancer. For example SVM classifier technique [1].

Neural networks is one of the best approach for predicting the breast cancer. In this report, a problem has been describe where breast cancer data have been given and a neural network need to develop with best hypothesis, at the end after training on training data the neural network will be able to classify whether someone has breast cancer or not.

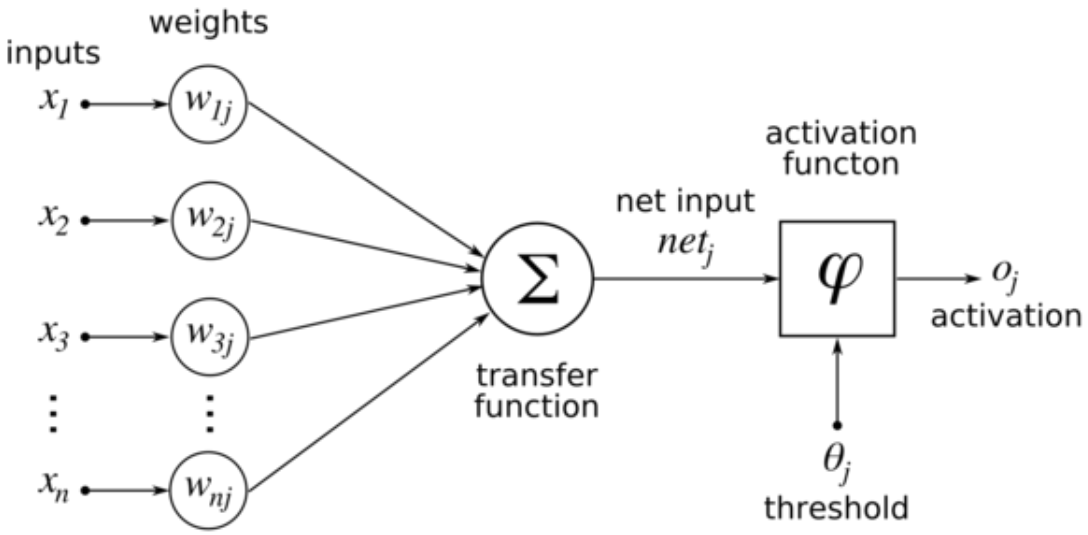
## 0.2 Background

An artificial neural network is an information processing paradigm that is highly inspired by the biological nervous system. The inventor of the first neurocomputer, Dr. Robert Hecht-Nielsen,

defines a neural network as "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs" [2]. The human brain is composed of 86 billion of nerves cells call neurons.

Following is the model of an artificial neuron.

### 0.2.1 Neuron



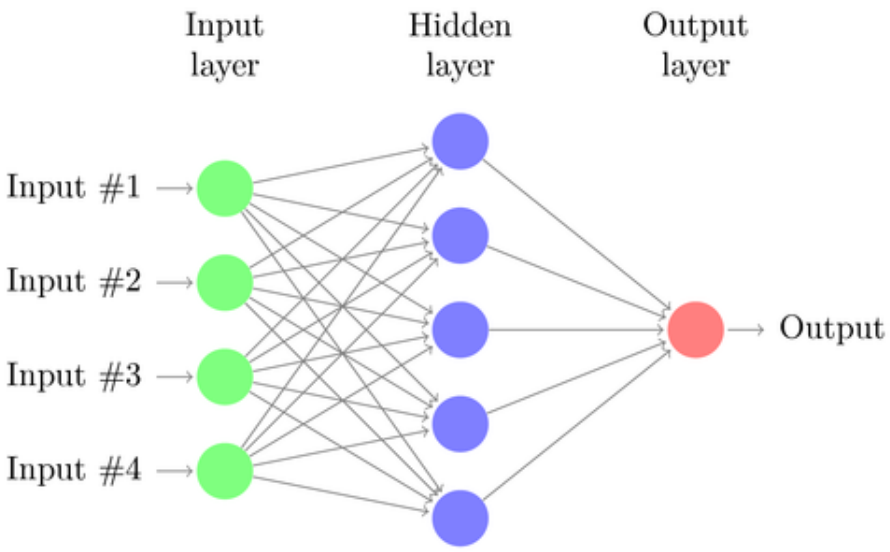

Inputs to the network are represented by the x1 mathematical symbol, Xn. Each of these inputs are multiplied by a connection weights, Wn.

Sum = W1X1 + …….. + WnXn

These products are simply summed and fed through the transfer function f(), to generate a result and then output. The activation function decide the output based on its type. There are a type of activation functions, for example binary threshold function which does binary classification. Others are like, piecewise and sigmoid Gaussian etc.

## 0.3 Artificial Neural Network

A typical neural network consists of a number of neurons which are called units arranged in a series of layers.




There are three major layers.

### 0.3.1 Input Layer

It contains those neurons which receive inputs from outside world on which the network will learn.

### 0.3.2 Output Layer

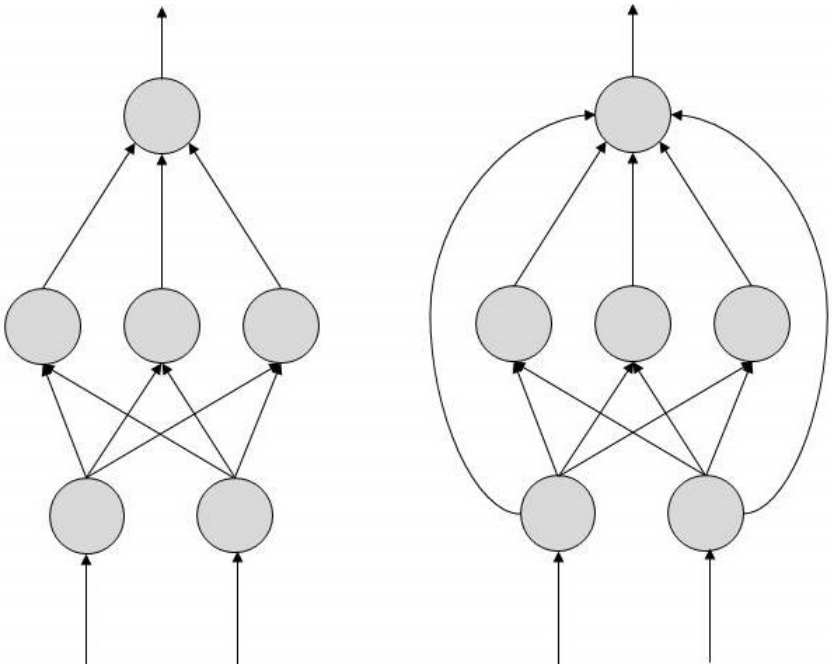
It contains those units (neurons) which response to the information about how its learn any task.

### 0.3.3 Hidden Layer

These neurons are in between input and output layers. The job of hidden layer is to transform the input into something that output unit can use in some way. There are two types of neural networks [3].

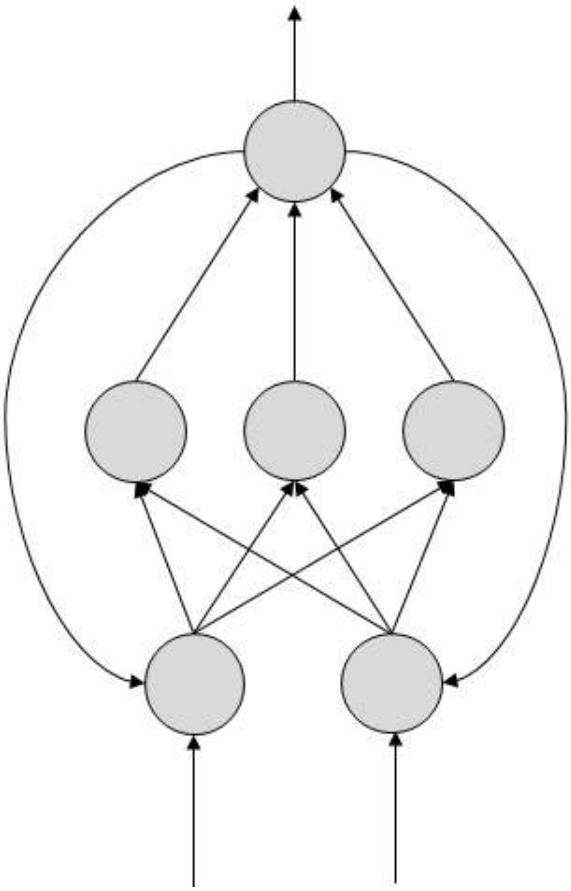
## 0.4 FeedForward Network

In the encourage forward system the stream of data is unidirectional. A unit sends data to other unit from which it doesn't get any data. There are no criticism circles. They are utilized as a part of example age/acknowledgment/grouping. They have settled information sources and yields [2].



## 0.5 FeedBack Network

In feedback network, feedback loops are allowed. They are used in content addressable memories. Here is the figure for feedback network [2].



As it is earlier said that neural network architecture is mainly inspired by the human brain. A brain is some what learns by the previous experience. As long as brain gets older it stores more and more information into it. So on the basis of some stored knowledge it takes decisions. Similarly neural network trained on the existing knowledge. In neural network there are weights which are used for learning. Slightly change in the weights caused the neural network to learn something. Neural network changes its weights on the basis of following formula.

$weight(x+1) = weight(x) * (learningrate) * (desired_output - actual_output) * (bias)$

From the above equation it is clear that if the desired output and actual output is equal then there will be no learning.

## 0.6 Main Part

In this section I will explain the methodology which I have used to develop the neural network and its justification. This section also include the complete procedure from gathering data to the complete processing and training the data.

### 0.6.1 Data Gathering

I have collected data from the UCI Machine Learning data set repository. There are eleven column in total in the data set, first column is the id of the patients and nine column consist of the attributes

of the disease or symptoms, the last column indicates the label class. There are two types of values in last column 2 and 4. 2 indicates begin and 4 indicates malignant. Total values in the whole data set are 699 including some missing values.

### 0.6.2 Pre-Processing

The data was in .txt file. I brought the data into Matlab manually by converting the .txt file into .m file. For this I had to assigned it to a variable and put square brackets around it. The data needed some modifications before sending it to the neural network for training. As the first column of the data indicating patients id which is use for the neural network training, so I dropped it intentionally. Then I removed some missing values in the data set. In the data set '?' was indicating the missing values. First I find '?' in the data using Control+F in the data set and replaced it with 0.

### 0.6.3 Creating and Training Network

After some needed modifications when data set was ready a neural network was build using newff function in the Matlab along with setting different attributes like learning rate, goal, epochs, activation function etc. Firstly, neural network was build then sent it to train function with training data in order to train the network. After training, then sent to the SIM function with testing data for testing the network.

### 0.6.4 Post Processing

So far I have done training and testing. Now a function to measure the accuracy of the neural network. Following is the formula used to measure the accuracy.

$Performance = (count_of_testedResult_matched/count_of_actualOutput) * 100;$

It gives the performance in terms of percentage.

## 0.7 Experimental Results and analysis

I have made some hypothesis and then verified these hypothesis by experimental results.

### 0.7.1 Hypothesis

#### Hypothesis 1

"As long as increase the training data the neural network should increase the learning ability. So that it could give better results".

First hypothesis has been made about certain parameters.

epochs = 50
Validation checks = 10
Hidden Layers = 3

These are the generic parameters. The accuracy by the neural network on different percentages of testing and training data is as follows.

| Training Data | Testing Data | Accuracy |
|---|---|---|
| 10 percent | 90 percent | 96.9106 |
| 20 percent | 80 percent | 97.2578 |
| 30 percent | 70 percent | 96.8815 |
| 40 percent | 60 percent | 97.0732 |
| 50 percent | 50 percent | 97.9532 |
| 60 percent | 40 percent | 99.0244 |
| 70 percent | 30 percent | 99.0244 |
| 80 percent | 20 percent | 99.270 |
| 90 percent | 10 percent | 100.0 |

This shows that as long as I increase the training data and decrease the testing data, it shows better results. It seems that the prediction of cancer by neural network will be high when the training data is highest. So that the neural network could train itself with its best.

**Hypothesis 2**

As the process of the learning of the neural network is slow. I just have to change the one of its parameters to see the slighter improvements that's what the science is which I believe. If I change the epochs from 50 100 then there will be more iterations for the neural network to reach it's goal. Let's see the performance. The parameters for the hypothesis are as:

epochs = 100
Validation checks = 10
Hidden Layers = 3

| Training Data | Testing Data | Accuracy |
|---|---|---|
| 10 percent | 90 percent | 95.2846 |
| 20 percent | 80 percent | 95.9781 |
| 30 percent | 70 percent | 97.0772 |
| 40 percent | 60 percent | 97.5610 |
| 50 percent | 50 percent | 97.9532 |
| 60 percent | 40 percent | 98.1752 |
| 70 percent | 30 percent | 98.0488 |
| 80 percent | 20 percent | 99.2701 |
| 90 percent | 10 percent | 100.0 |

If I compare the performance of these results with the previous experimental results. It seems if I change the epochs from 50 to 100 the accuracy decreased at first then gradually increased to 100 percent.

**Hypothesis 3**

If I increase the validation checks from 10 to 20 the neural network will have more opportunities to learn. The parameters for this hypothesis are as follows.

epochs = 50
Validation checks = 20
Hidden Layers = 3

| Training Data | Testing Data | Accuracy |
|---|---|---|
| 10 percent | 90 percent | 95.6098 |
| 20 percent | 80 percent | 96.8921 |
| 30 percent | 70 percent | 97.4948 |
| 40 percent | 60 percent | 97.5610 |
| 50 percent | 50 percent | 98.2456 |
| 60 percent | 40 percent | 97.4453 |
| 70 percent | 30 percent | 99.0244 |
| 80 percent | 20 percent | 99.2701 |
| 90 percent | 10 percent | 100 |

The results are quite unexpected I can see a slight change in the performance. At the beginning when the training data is low the performance is 95. As long as the training data increases the accuracy increased. This experimental is more accurate then the previous one when the epochs was 10.

**Conclusion**

Neural networks are already being used in the classification of breast cancer. The purpose of this work was to come up with new techniques which can improve the performance and accuracy of neural network to classify breast cancer. There are infinite numbers of parameters which we can test in order to get better accuracy but here we focused on only three; data distribution, validation checks and epochs. By increasing the training data performance get better. But there were no improvements by increasing the number of validation checks and epochs.

# Bibliography

[1] Yi. Anna Rejani and D. Selvi, “Early Detection of Breast Cancer Using Svm Classifier Technique,” Int. J. Comput. Sci. Eng., vol. 1, no. 3, pp. 127–130, 2009.

[2] R. Bapi and S. McCabe, ”Artificial intelligence and neural networks: Steps toward principled integration”, Neural Networks, vol. 9, no. 3, pp. 545-548, 1996.

[3] R. Bapi and S. McCabe, ”Artificial intelligence and neural networks: Steps toward principled integration”, Neural Networks, vol. 9, no. 3, pp. 545-548, 1996.