DECEMBER 3, 2017

BREAST CLASSIFICATION

A NEURAL NETWORKS APPROACH

ASIM HAMEED KHAN
14031254
Coursework -1

Table of Contents

Introduction

Breast cancer is the cancer that forms in the cells of breasts. Breast cancer can occur in both men and women but its most common in women[3]. This disease has been widely spreading which needs to be controlled. Many efforts had been made in the medical sciences to predict by looking into symptoms and causes which are responsible for breast cancer. Many methods were discovered to predict the disease. One of which is the neural network classification method in which the data of the patient is collected and trained by the neural network. This report further illustrates how a neural network classifier is used to detect the breast cancer.

Background

Basic Neuron

Artificial neural Network is the paradigm that is inspired the biological nervous system. It is like artificial human nervous system which receives, process and outputs the information in terms of computer science. The following diagram shows the general model of the neural network:

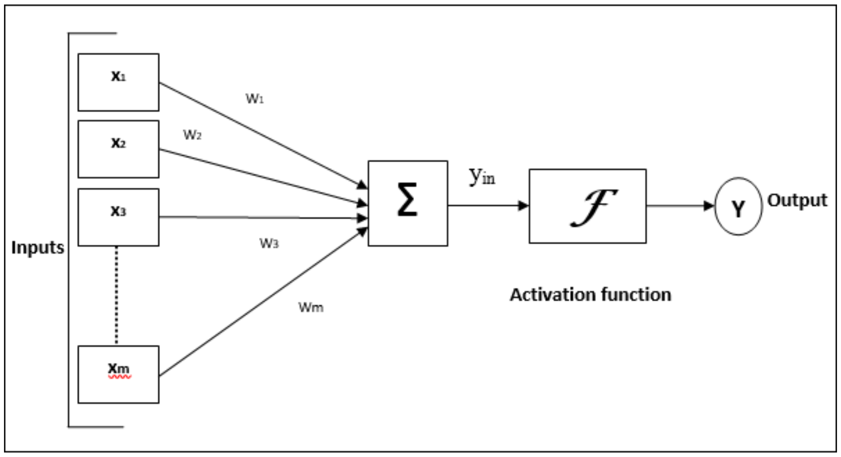

Figure 1 Model [5]

All the inputs are fed from input layer, which are then processed in the hidden layers and the processed data is made available at output layer. Following is the manner in which this concept is laid:

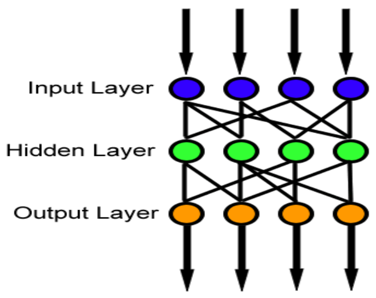

Figure 2 Basic Flow [2]

From the above neural network model, the input can be calculated as follows:

$$Y_{in} = X_1.W_1 + X_2.W_2 + X_3.W_3 \dots X_m.W_m$$

$$\text{i.e. Net Input } Y_{in} = {}_i\sum^m xi.wi$$

The output can be calculated as:

$$Y = F(y_{in})$$

There are two types of neural networks:

Feed Forward Networks

It's a non-recurrent model in which in nodes in a layer are connected to previous layers. They don't have any loopback feature. The input and outputs are fixed and are usually used for classification. They can be single or multiple layered networks. Following diagram shows an example of feedforward network:

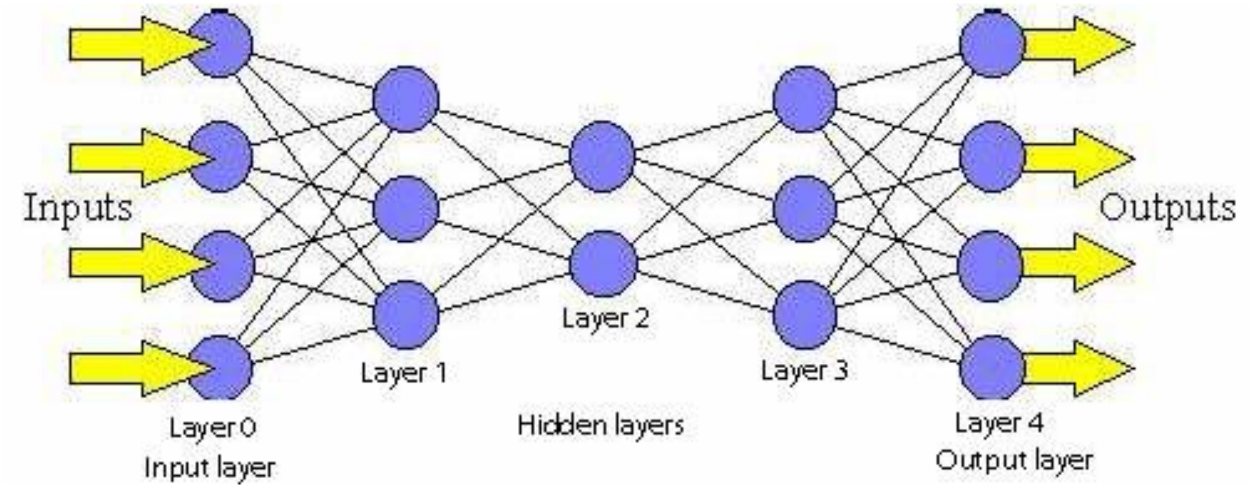

Figure 3 Feed Forward Network [4]

Feedback Network

As the name tells it self that the signal can flow in both directions which means that can continuously change state of equilibrium. An example is shown in diagram:

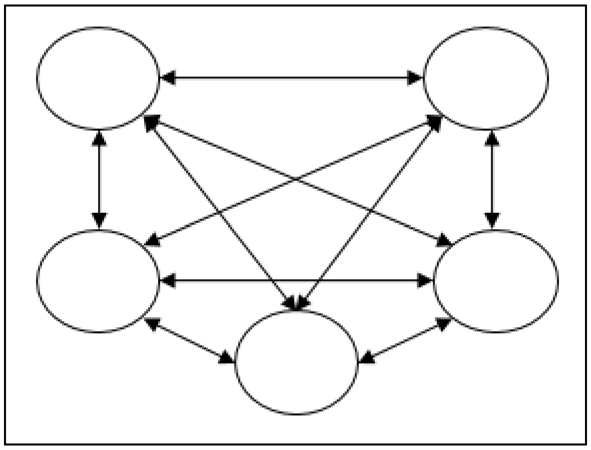

Figure 4 simple feedback network

Main Part

This section shows the methodology to develop a neural network and its justifications. It also contains the information about the procedure to prepare and train the data:

Data Gathering

The dataset used in this report is collected from UCI Machine learning dataset repository. The dataset is publicly available at:
http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

The dataset consists of 699 rows and 11 columns. First columned will be further ignored in this report because it consists of patient ID which is useless to us. The next 9 columns contain the attributes of the disease which will be further used in training of the data. The last column indicates whether the person has benign or malignant cancer. 2 represents benign and 4 represents malignant. There are also some missing values which are represented by “?”.

Pre-Processing

The data was brought into MATLAB by converting .data file into .m which was saved into a variable in MATLAB workspace.

```
data = load('breastCancer.m');
```

Some pre-processing was done to prepare the data for neural network. The patient ID was excluded, 2 to 9 columns were prepared for training and last was used as target.

```
trainingData = data(:,2:10); %EXCLUDING patient ID

classData = data(:,11); %data on which NN will be classified
```

The missing values “?” was earlier replaced with 0 so that we can bring it to MATLAB and itcould be handled by four ways:

- Mean Value
- Neighbor value
- Random Values
- Exclude rows from data

In general, there is no best method for handling such kind of data But, mean values was chosen over others because it can represent our data way.

```
Var = mean(data(:,7))

round(var)
```

The mean value of column 7 of overall data was 3.4635 ~~whose round was replaced with 0~~ in overall data. So, the values was 3 at “?”

The last column also needed pre-processing because it was in the from of 2 or 4 but ~~we to use~~ activation function ‘tansig’ which takes the values in the range of -1 to 1 so 2 was replaced with 0 and 4 with 1.

```
classData(find(classData == 2)) = 0;

classData(find(classData == 4)) = 1;
```

The data was further divided into 70% as training data and 30% as testing data.

```
%training inputs

trainInput = trainingData(1:490,1:9);

trainTarget = classData(1:490,1);


%testing

%testInput = trainingData(490:699,1:9);

%testTarget = classData(490:699,1);
```

Developing Neural Network

After pre-processing all the data is prepared for the further progress. The neural network was created by newff in MATLAB and its parameters were set like epochs, goals, learning rate, validation checks, activation function, training algorithm etc. After building the neural network it was sent to neural network and then it was tested by passing testing data into net.

```
net = newff(trainInput',trainTarget',10, {'tansig' 'tansig'},
'traingdx', 'learngd', 'mse');

net.trainParam.epochs = 1000;
net.trainParam.lr  = 0.01;

net.trainParam.max_fail = 1000;

net.trainParam.goal = 0.01;
```

Post-Processing

After training and testing was done it was time to check the accuracy of the neural network. Following code was used to measure it:

```
error = gsubtract(testTarget',testOutput);

check = find(error ==0);

accuracy = length(check)/length(error) * 100;
```

Experimental Results and Analysis

~~Firstly, the hypothesis about the observation that were observed during initial preparation of the data and the neural network~~. The experiments were carried out to the ~~confirm~~ the hypothesis and their results will be presented later.

Hidden layers

Hypothesis

Increase in number of hidden layers should increase the accuracy, should increase the learning time and ~~performance~~.

Experiment

The number of layers in newff function was changed for experimentation while keeping everything else constant and accuracy was calculated. Readings are shown in the table:

NO. OF HIDDEN LAYERS	**ACCURACY (%)**
1	98.5
2	99.04
3	98.5
4	98.5
5	98.09

Results

It was clear from the experiment that there cannot be a specific/unique number of hidden layer for architecture of every neural network. Here for this problem the optimal number of hidden layer is 2 as it shows the best accuracy from other readings. As we increased the number of layers the time was surprisingly decreasing and it also came with a little decrease in accuracy as we can last three readings.

Learning Rate

Hypothesis

The increase in learning rate increases the accuracy and decreases the time to train.

Experiment

The learning parameter was changed gradually and reading were noted which are shown into the table below:

LEARNING RATE	ACCURACY (%)
0.01	99.5
0.02	98.5
0.03	98.09
0.04	99.04
0.05	98.05

Results

It was surprising to see that the minimal learning rate experiment came with the high accuracy but with the increase in the learning rate, the network took less time to train itself on the data. ~~So~~ ~~it can be said that it was safe to say that for further experiments~~ we should go with learning rate of 0.01 which came with the high accuracy.

Weights and biases

Hypothesis

Every time we run the script and it comes with the different type of accuracy. Sometime it comes with the very good accuracy and sometime it gradually falls. But the hypothesis is that whenever we run script ~~It,~~ every time after training it should come up with higher accuracy.

Experiment

From studies, even a small change in weights can lead to significant change in output. sometimes results may be worse [1]. Therefore, different readings were taken to measure accuracy while saving and without saving weights and biases of the neural network. Following table contains accuracy(%age):

WITHOUT SAVING	WITH SAVING
98.5	97.61
99.04	98.09
98.09	98.5
96.6	98.5
97.14	99.52

Result

According to the readings from above experiment, it is clear that without saving the weights it always gave us random accuracy because sometimes the weights were so good that the accuracy was very high and sometimes it dropped down. After saving the weights it can be seen that there was a gradual increase in the accuracy every time.

Training algorithms

Hypothesis

As we checked in above experiments by changing different learning parameters of the neural network. So, this time, let see what different learning algorithms does. It is believed that some learning algorithms will better than others.

Experiment

Different algorithms were tried to check the performance:

ALGORITHM	PERFROMANCE
TRAINR	Slowest
TRAINGDX	Fast
TRAINGDM	Fast

Result

It was observed that the slowest algorithm took more than 10 min to train for 1000 epochs but the gradient descent algorithms were the most optimal as they had the best accuracy and very less training time. So, as we were using TRAINGDX in previous experiments so it would be optimal to continue with this for further experiments.

Activation function

Hypothesis

As we know from MATLAB documentation that logsig's range is from 0 to 1 and tansig's -1 to 1. So, it was believed that logsig should work better than tansig as we have positive ranges in our data.

Experiment

The activation functions were changed in newff function and the readings were noted in the table below:

ACTIVATION FUNC.	ACCURACY (%)
LOGSIG, LOGSIG	22.38
LOGSIG, TANSIG	96.66
TANSIG, LOGSIG	22.3
TANSIG, TANSIG	98.01

Result

Surprisingly, the lowest accuracy values were the ones that actually came from the activation function logsig when it was one output layer. So, further training the tansig will be used.

Changing the amount of testing and training Data

Hypothesis

If we increase the training data then it should be better for neural network in terms of accuracy. The greater the amount of training data the more it will be good for neural network in terms of input and it should become more and more accurate.

Experiment

The training data was started from 10% and testing data at 90%. With every reading the training data was increased 10% and testing data decreased 10% which would be pretty clear from statistics noted in table below:

Training data (rows)	Testing data (rows)	Accuracy (%)
1 – 70	70 – 699	96.5
1 – 140	140 – 699	96.2
1 – 210	210 – 699	96.93
1 – 280	280 – 699	97.38
1 – 350	350 – 699	97.14
1 – 420	420 – 699	98.5
1 – 490	490 – 699	99.04
1 – 560	560 – 699	95.7
1 – 630	630 – 699	98.5

Results

After analyzing, it was unexpected that changing the amount of training data does not produce the best accuracy which resulted into further assumptions.

Further experiment was connected by checking reversing the order of the data. As this time training started from bottom rows. Further readings are available in the table:

Training data (rows)	Testing data (rows)	Accuracy (%)
630 – 699	1 – 630	77.93
560 – 699	1 – 560	94.82
490 – 699	1 – 490	92.04
420 – 699	1 – 420	93.5
350 – 699	1 – 350	92.00
280 – 699	1 – 280	93.2
210 – 699	1 – 210	91.90
140 – 699	1 – 140	91.4
70 – 699	1 – 70	88.57

Checking in the result also did not provide any of the fact that providing more training data will result into higher accuracy rate.

Conclusion

Neural networks are widely being used in classification problems. The overall purpose of this work was to determine newer system to predict breast cancer. We did several experiments by changing the parameters of the neural network to achieve the best accuracy rate and performance. The experiment that we performed show that many number of hidden layer result with a little decrease in accuracy but also performs in less time. By saving the weights and biases results in better training. Small learning rate results in good accuracy. We had a look at how different training algorithms and their performance. Data distribution for testing and training showed that the increase in training of the data doesn't necessarily improve accuracy.

References

1. Anon, How to improve performance of Neural Networks. Available at: https://d4datascience.wordpress.com/2016/09/29/fbf/ [Accessed December 3, 2017a].
2. Anon, Introduction to Neural Networks. Available at: https://becominghuman.ai/artificial-neuron-networks-basics-introduction-to-neural-networks-3082f1dcca8c [Accessed December 3, 2017b].
3. Staff, M.C., Breast cancer. Available at: https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470 [Accessed December 3, 2017].
4. Standford, Feed Forward Network. Available at: https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/feedforward.html [Accessed December 3, 2017].
5. TutorialPoint, Artificial Neural Network Basic Concepts. Available at: https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm [Accessed December 3, 2017].