Namal College Mianwali

Breast Cancer Classification Using Neural Network approach

A well designed approach of development of breast cancer classifier

NAME: Murad khateeb
UOB#: 14031274

NOVEMBER 22, 2017

Contents

INTRODUCTION

The breast cancer is considered amongst the most fatal type of cancers which causes an alarming number of causalities around the world. According to WHO the number of deaths caused by breast cancer is increasing at an alarming rate and will reach to 12 million by 2030 (George *et al.*, 2012). Various historical findings had revealed that the breast cancer had been there as long as humans (*A Brief History of Breast Cancer*, no date). It commonly effects the females and rarely a male. The exact information on what causes the breast cancer is yet unknown, but there have been various myths and theories on it including the imbalances in various hormones in the body, the growth of unwanted cells, number of children, the breast feeding duration etc. It had been treated by various traditional and spiritual methods (*A Brief History of Breast Cancer*, no date).

The importance of breast cancer and its cure had lead people to work in the diagnostic and treatment of this fatal disease. The best cure of this disease seems to be its early detection as it becomes very much curable if detected at an early stage. The recent research and techniques in data analysis have urged the medical organizations to keep the records of the symptoms, medication and the treatment results of their patients, which later can be very much helpful to extract meaningful information by applying the powerful data mining algorithms (Raad, Kalakech and Ayache, no date).

There are various medical procedures for the detection and cure of the breast cancer. These medical procedures are time taking, expensive and require skilled examiners and the chances of error are relatively higher. There is another efficient way of detecting the breast cancer which is to develop intelligent machines using the techniques of artificial intelligence to classify the data based on the previously provided data and its results (Mandal and Banerjee, 2015). We are using the Wisconsin Breast Cancer Database (*UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set*, no date) to train the neural network which later will be able to classify the input data into either benign or malignant type of breast based on its supervised training.

BACKGROUND

Breast Cancer Classification

Classification is the process of placing an object or item into the right category based on its characteristics. A classifier performs certain checks on the set of input values and places the object into the class it belongs to. In our case we have used an artificial neural network as a classifier which will take a features vector extracted from the input data and will respond with a category to which the input object belongs (Tino, Benuskova and Sperduti, 1997). The input data here is taken from the slides provided by a medical procedure called as the **find needle aspiration (FNA)**. In the process of FNA, a small sample of suspicious tissues is examined to classify or detect the type of cancer (Mendoza *et al.*, 2011). The result of the FNA is a set of various samples critical for the detection of breast cancer. Our job is to classify the input data into either benign or malignant types of cancer using an artificial neural network.

Neural Network

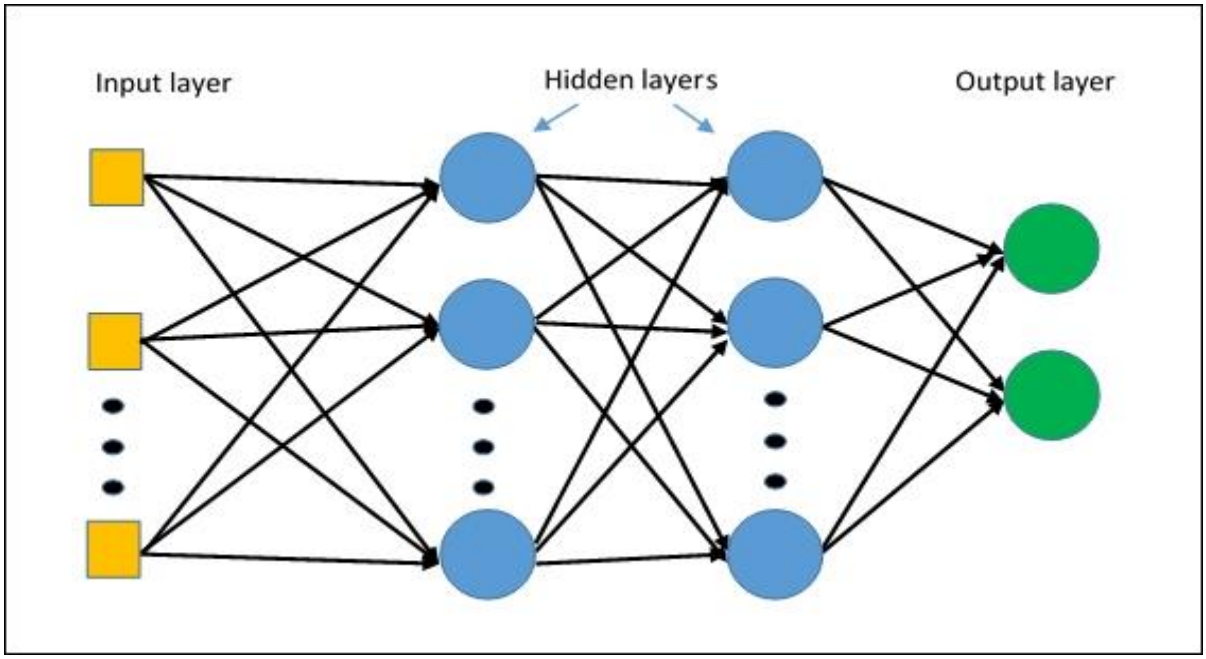
Pattern matching is a technique in which the set of input is mapped to output based or a certain pattern. Neural networks are considered as the best problem-solving agent when the problems are related to pattern matching. In our case a neural network acting as a classifier will place the objects into the category it belongs by matching the patterns found in the input data. Following are some techniques and NN models that have been used for the classification purposes. Based on the working principles and architecture, the neural networks are mainly classified into the following two categories.

1. Feedforward Neural Network
2. Recurrent Neural Network

The connections between the various layers and units in a neural network defines the category of neural network it belongs to. If the layers are connected in a linear way it will be a feedforward neural network and if the connections form a circle, it will be categorized as a recurrent neural network.

Multilayer Perceptron (MLP)

A multilayer perceptron is a type of feedforward neural networks. Other than the input and the output layers, it also has the one or more hidden layers. Every neuron from the nth layer is connected to every neuron in (n+1)th layer. This network is also called as a supervised neural network, because it needs the set of both the input and the output data in order to learn during the training procedure. The data from the input vector is multiplied by the interconnection weights and are propagated to the next layer for further processing and the process is repeated unless we reach at the output layer (*What are Neural Networks & Predictive Data Analytics?*, no date). The trained network then can be used to classify the untrained data based on the prier training.



A MLP with 2 hidden layers

Radial bases function (RBF)

The radial bases function is considered as an alternative to MLP. It uses the sigmoid activation function. This kind of neural network is considered useful for classification and modelling purposes. Since we are using the MLP we may not need to go into the details of the RBF.

Back-propagation algorithm

MLP and many other neural networks use back-propagation technique for training purposes. In a back propagated neural network the resultant output of the neural network is compared with the desired output and an error factor is computed. The error is then propagated back to the neural network and the weights are adjusted such that the error minimizes (Raad, Kalakech and Ayache, no date). Repeating the process makes the resultant output closer to the desired output.

Neural Network as a breast cancer classifier

Since there exist a major role of pattern matching in classifying the breast cancer data and the neural networks are considered best in pattern matching so a lot of researches has used neural networks to classify the breast cancer. We are using a multi-layered, feedforward neural network that uses the back-propagation mechanism for learning. The 9 defining features obtained from the Wisconsin Breast Cancer Dataset will be given as an input vector to the neural network. The output of these input vectors is also thrown to the neural network in order to insure the supervised learning of the neural network. An error is calculated with each iteration and the weights are adjusted accordingly in order to be able to classify the unknown data. The neural network will classify the input data into either benign or malignant based on the training of the neural network.

MAIN PART

Pre processing

Before we start developing the neural network for the classification of the breast cancer, we must be having a few things ready. These includes,

1. Wisconsin Breast Cancer Database's data
2. Matlab R2015a

We need to refine the data received from the WBCD in order to use it for the training of the neural networks. We have to go through the following steps in order to prepare the data as required by the neural network.

1. The data from the WBCD consists of a total of 699 values. Out of those 699 values 16 are the samples where any one of the attributes is missing and it was replaced with '?'. We need to take care of that so that we don't end up producing erroneous results. One approach could have been of replacing '?' with the mean value of that whole column, this could have been a good approach but we had removed all the 16 values containing '?' just to avoid any kind of adulteration in the actual data.
2. The WBCD data has been classified into two kinds of data namely benign (2) and malignant (4). Throughout our experimentation with the neural network we have divided the training data in 50% of both the classes, where it was possible.
3. We have sorted the whole data on the bases of the last column of the WBCD, which is the class of the breast cancer to which the corresponding sample data belongs. We did this because it will be easier to insure that we have 50% data from both the classes in the training data of the neural network.

System Architecture

We are using a multilayer feedforward neural network which uses the technique of back-propagation for the classification of the breast cancer.

Training

1. First of all the data for the training and testing of the neural network had to be prepared, we selected 50% data from each class to be given as input and output data for the training of the neural network. We have prepared the remaining data for the testing input and testing output data for the purpose of testing the network. The testing output data will be compared with the output generated by the neural network when was fed with the testing input data.

2. In the next step, we have used the Matlab function 'newff' to create a feedforward, backpropagation neural network. The number of hidden layers and hidden layer neurons and the names of various learning and training functions have also been sent as arguments to the function.
3. We set the values of few of the 'net.trainParams' in order to refine the training of the neural network, finally we used the Matlab's built in 'train' function which had trained the network on the input and output data given to it as arguments.

Testing

Now that our network has been trained, we will test the data by giving it the input data and then comparing the generated output with the actual output against the input data.

1. We passed the testing input data to the trained network which had generated the output based on its training.
2. After converting the output generated by the neural network, we have compared the two matrices of the output data, one of them was generated from the original data and the second one of has been generated by the neural network.

Accuracy Calculation

We used the following formulae to calculate the percentage accuracy of the neural network.

$$\text{Accuracy} = \frac{\begin{matrix}Number\ of\ the\ matches\ between \\ the\ calcualted\ and\ the\ orignal\ output\end{matrix}}{Total\ number\ of\ the\ testing\ data} \text{ X } 100$$

EXPERIMENTAL RESULTS AND ANALYSIS

We have developed the neural network and now we will test this neural network in order to be able to better understand the behaviour of this neural network. Initially, we will start with random values of different variables and training parameters.

Initial Experiment

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
300	100	0.1	20

Results

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
96.3446	8	11	Performance goal met

Experiment 1

Hypothesis

"If you decrease the value of number of epochs in the run configuration, the neural network will have lesser sessions of training so it ultimately should have a lesser accuracy then it currently is having".

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
300	50	0.1	20

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
96.3446	8	11	Performance goal met

Result Analysis

By decreasing the number of epochs by almost half the performance didn't decrease, but one thing that should be noted here is the reason for the termination of the training process which says "Performance goal met". It means that the extra number of epochs are not doing any good in the training of the neural network. The number of epochs it takes to meet the goal is '11' so let's decrease the number of epochs bellow 11.

Experiment 2

Hypothesis

"If you decrease the value of number of epochs below 11 in the same configuration, the goal of the training will not be met by then and the NN will have lesser sessions of training so it ultimately should have a lesser accuracy then it currently is having".

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
300	10	0.1	20

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
95.303	7	10	Maximum epochs reached

Result Analysis

The accuracy has decreased as expected by decreasing the number of epochs such that the performance goal does not meet. When I kept on decreasing the number of epochs I found that there was not any considerable decrease of accuracy. It seems that the training data we set is not enough

for the neural network to train, so I guess we should keep the number of epochs same and increase the amount of training data.

Experiment 3

Hypothesis

"Start from the same configurations as were in "experiment 1". If we increase the amount of training data. The neural network will have much more instances to train on and the accuracy should increase. The training time should also increase".

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
500	100	0.1	20

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
99.45	09	08	Performance goal met

Result Analysis

As hypothesized, the accuracy of the neural network has increased. The time taken has also increased but that seems legit as the number of input data has increased. Now that we know that by increasing the training input data, the accuracy of the network can be increased, let's see if by decreasing the training input data, the accuracy of the network decreases or not.

Experiment 4

Hypothesis

"By decreasing the training input data to a significant amount, the accuracy and the time taken by the resultant system should decrease".

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
100	100	0.1	20

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
93.82	05	9	Performance goal met

Result Analysis

There has been seen a significant decrease in the accuracy and the time taken by the system, but the rate at which it decreased was very slow. It was like '3%' decrease in '200' of the decrease in training

data. Let us try to figure out, what will the rate of decrease, if we start further decreasing the training data. The current rate of decrease in accuracy with a decrease in training data is “1.5%/100”.

Experiment 5

Hypothesis

“By further decreasing the training input data, the rate of decrease in accuracy will increase“.

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
50	100	0.1	20

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
89.88	01	9	Performance goal met

Result Analysis

Just like before, the accuracy and time has been decreased with the decrease in training data but the rate at which it decrease has increased. Now the rate at which the accuracy decreases with the decrease in number of training input data is “8%/100”. It can easily be concluded from the above experiments that the accuracy and the time taken by the neural network during its training decreases with the decrease in number of training input data but the rate at which the accuracy decreases, increases with the decreasing number of input data.

Now that we know the behaviour of the neural network on the increase or the decrease of the training input data, lets us see how does the goal effect the training of the neural network.

Experiment 6

Hypothesis

“By decreasing the goal and making it closer to ‘0’, the neural network will try to refine its training and the accuracy of the neural network should increase and the time taken by the neural network for its training should increase“.

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
50	100	0.01	20

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
88.62	03	24	Validation stop

Result Analysis

Unlike our hypothesis, the accuracy of the neural network has decreased, but the interesting thing to note here is that the neural network has stopped its training because of the max number of failures of the learning algorithm has been reached. I tried to increase the validation check but it seems that the training had stuck into something and it always terminates due to reaching to the "max_fail", so let's repeat the procedure by increasing the training input data to 100.

Experiment 6 (Again)

Hypothesis

"By decreasing the goal and making it closer to '0', the neural network will try to refine its training and the accuracy of the neural network should increase and the time taken by the neural network for its training should increase".

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail
100	100	0.01	20

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
95.57	24	100	Maximum epochs reached

Result Analysis

The accuracy of the system has increased so has the time taken, both of which make complete sense. Hence the more accurate the neural network will be if you set a more specific and closer to the actual goal. The time will also increase as neural network will utilize its available number of epochs to reach the goal.

Experiment 7

We have been experimenting with a neural network which was set to have only one hidden layer which had 10 neurons in it. Let's change the number of hidden neurons to see how it effects the performance of the neural network.

Hypothesis

"The decrease in the number of hidden layer neurons will result in the decrease in accuracy of the trained neural network".

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail	Hidden layer neurons
100	100	0.01	20	02

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
94.34	10	43	Performance goal met

Result Analysis

The accuracy of the system has decrease a bit with the decrease in the number of hidden layer neurons, but this time the performance goal met and the time taken by the neural network for its training decreases which clearly mean that the greater the number of hidden layer neuron, the greater will be the complexity, more training time will be required and the more refined training of the neural network will happen. Almost the same kind of behaviour had been seen with having different number of hidden layers.

Experiment 8

Hypothesis

"Keeping in view our evolved understanding about our neural network, we can say that the following starting configuration will result in the best performance of this neural network".

Configurations

Training Data Count	Number of Epochs	Goal	Validation Check max_fail	Hidden layer neurons
500	100	0.095	20	10

Resultant Accuracy

Accuracy (%)	Time taken (sec)	Number of epochs utilized	Remarks
99.45	12	09	Performance goal met

Result Analysis

This is only one of the configurations where it produces the best results. There may even exist certain configurations where the accuracy further increases.

CONCLUSION

We have developed a multi-layered feedforward neural network which uses backpropagation for learning purpose and it had been used as a breast cancer classifier. We trained and tested the neural network on various variables and tried to understand the general behaviour of neural network, when fed with different variation of data. Finally, based on our evolved view about the neural network we came up with a configuration that results in almost the best performance of the neural network.

REFERENCES

A Brief History of Breast Cancer (no date). Available at: https://www.maurerfoundation.org/a-brief-history-of-breast-cancer/ (Accessed: 1 December 2017).

George, Y. M. *et al.* (2012) 'Breast Fine Needle Tumor Classification using Neural Networks', 9(5), pp. 247–256.

Mandal, S. and Banerjee, I. (2015) 'Cancer Classification Using Neural Network', *International Journal of Emerging Engineering Research and Technology*, 3(7), pp. 172–178.

Mendoza, P. *et al.* (2011) 'Fine Needle Aspiration Cytology of the Breast: The Nonmalignant Categories', *Pathology Research International*, 2011, pp. 1–8. doi: 10.4061/2011/547580.

Raad, A., Kalakech, A. and Ayache, M. (no date) 'Breast Cancer Classification Using Neural Network Approach: Mlp and Rbf', *Networks*, 7(8), p. 9. Available at: http://www.acit2k.org/ACIT/2012Proceedings/13233.pdf.

Tino, P., Benuskova, L. and Sperduti, A. (1997) 'Artificial Neu', 8(3), pp. 455–472.

UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set (no date). Available at: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 (Accessed: 1 December 2017).

What are Neural Networks & Predictive Data Analytics? (no date). Available at: http://www.neurosolutions.com/products/ns/whatisNN.html (Accessed: 1 December 2017).

Multilayered perceptron. Available at: https://www.safaribooksonline.com/library/view/getting-started-with/9781786468574/ch04s04.html (Accessed: 1 December 2017).