Breast cancer detection using neural network

Neural Network and Fuzzy System

Course work 1

Muhammad Ramzan Shahid Khan

14031243

Abstract:

The ability of a software to self-direct itself was the idea that led the world to think of 'autonomous systems'. Identification of breast cancer is an open study area which can be solved by such technologies. One of the mostly used techniques is Artificial Neural Networks as it is mostly used for automation system. The neural network are trained on specific data and then it is used for identification of unseen data since it is learned on the trained data so it can give best possible results. Many neural networks are already trained on breast cancer data and used for the identification of cancer on the basis of learned data. Most of the networks has given 97 to 99% accuracy. I developed neural network which is trained and then used on data for identification of cancer and different experiments and results are obtained.

1. Introduction

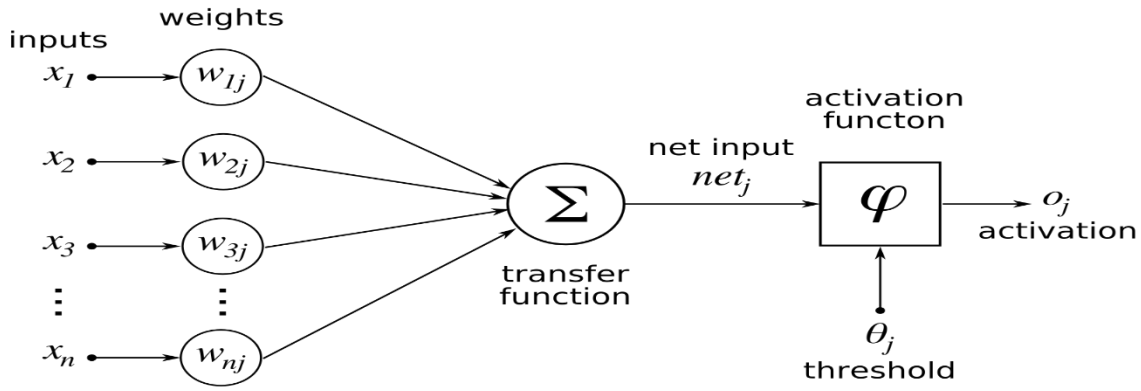
"Breast cancer affects one in eight women during their lives" [1]. It is the second cause of the death among women [2]. This is widely spreading disease which needs to be tackled. Many efforts have been made to discover such kind of disease. There are many methods and ways to discover or predict that if a woman has breast cancer or not. But neural networks can also be used for detection of similar diseases. This report describes a problem where breast cancer data is available and there is a need to develop a neural network which, after training on training data, can classify whether someone has breast cancer or not.

2. Background and Related work

Neural Network is inspired by human brain. A neural network is consist of many interconnected neurons.

a. Neuron

Following figure show the structure of neuron.

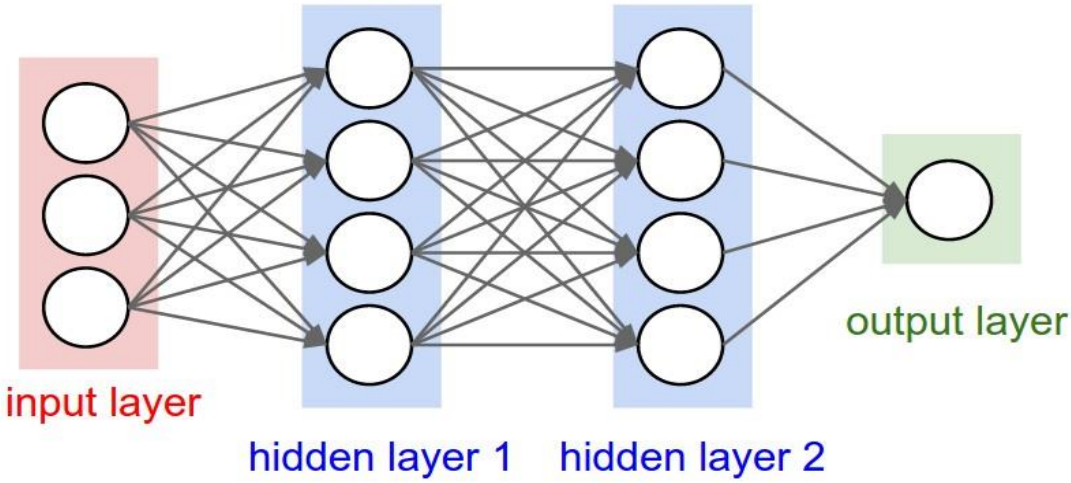


[3]

A basic building block of neural network is Neuron. It has several inputs and an output. Specific weights are assigned to inputs. Learning takes place by changing the weights. Inputs are multiplied with weights and then transfer functions sum them all, activation functions give the output bases on the type of activation functions because there are many types of activation functions. Like, binary threshold is a function which does binary classification. Activation function as 'tansig' which gives a bipolar value. Similarly, there are many more like piecewise, sigmoid, Gaussion etc.

b. Neural Network

Following figure shows the architecture of neural network.



[4]

A neural network consists of three layers. It has one input layer, one output layer and between input and output layers, are hidden layers which can be one to many layers. Each layer has some neuron which are interconnected through weights. There are some types of neural networks but here I will be using feedforward neural network.

c. Feedforward Neural Network

In feedforward network, data flow in forward direction, in unidirectional. Inputs are given to nodes and their output send forward to other nodes. In this network, there is no feedback loop. Following figure shows the architecture of feedforward network.

3. Data Processing

This section contains the methodology which I used for Data processing and developing neural network and training it on this data.

a. Data Gathering

I used Dataset available at UCI Machine learning dataset repository. This dataset has 699 instances and 11 columns. 1st column is the ID, 9 columns are the attribute of disease and last column is the class attribute which has two values 2 and 4: 2 indicates benign and 4 indicates malignant. In this dataset, 16 values were missing.

b. Pre Processing

Since the dataset is contained some missing valued so first manually replaced the 16 missing values represented by "?" is replaced with 0. Then dataset is saved in variable in Matlab. Since the ID column is useless so this column is dropped.

The missing values are replaced with the meanValue of 6th column of input_data. These missing values were in 6th column of the input data.

Mean of 6th column is taken and missing values are replaced in the input data through this code is following:

```
meanValue = round(mean(input_data(:,6)));
input_data(input_data(:,6)==0) = meanValue;
```

In the given data set missing values were indicated with "?" so firstly

I replaced "?" with 0 manually in note editor through finding with ctrl+F and then finally 0 is replaced with the mean Value of 6th column of input data in code. Programmatically, first it is taken the mean of 6th column of input data and round it, then replaced the 0 with this mean value of 6th column.

Since output data which is class attribute has values 2 and 4. So output data is also processed because transfer functions which are used like tansig has range [-1 1] and logsig [0 1].So I indicated 2 with -1 and 4 with 1.

Then I divided input_data and output_data into 4 parts, training_data, training_target, testing_data, testing_target.

Then inserted the values 0 and 1 in the training_target and testing_target so that result obtained from the trained network is compared with these values which is in the form of 0 and 1 values.

So training_target and testing_target data contains the values of 0 and 1.

c. Training Neural Network

Since the Data is preprocessed so now neural network is build using function newff in matlab then its different parameters are set like input_data, training_target, hidden layers, transfer functions, training functions, learning rate, and epochs. This network uses back propagation training function and gradient descent learning function mainly.

Then this build network is trained by train function and after training the network, this is tested on the testing_data and saved values in the result variable.

d. Post Data Processing

Since after testing the network, calculated the accuracy of the network through following:

First check the accuracy by using minus function on the testing_target data and result data obtained from the network. Sum the values which are saved in accuracy variable that are equal to 0 and stored in the variable match.

And values of 1 are saved in variable mismatch.

Then accuracy and percentage of accuracy is calculated through this formula.

```
accuracy = minus(testing_target',result);
match = sum(accuracy(:)==0);
mismatch=sum(accuracy(:)==1);
accuracy_percentage = (match/length(testing_target)*100);
disp(accuracy_percentage);
```

4. Experimental Results And Analysis

a. Hypothesis one

Hypothesis: First hypothesis is that if I increase the number of hidden layers, then learning time and also the performance should increase. This hypothesis is only on the change of hidden layers and others expects are not included. Hidden layers has nodes so hidden layers have different nodes and increase in nodes should also improve the performance.

Analysis and experiment:

I did experiments on this hypothesis that how the change in hidden layers and nodes effect the performance. As I did experiment on the given data of breast cancer on different hidden layers. It is cleared from experiments that surprisingly increase in number of hidden layers and nodes, decrease the accuracy. It gives best result on one

hidden layers and in the next more hidden layers and nodes, the accuracy is less than the 1 hidden layers.

Following table shows the results of different experiments.

No. of Hidden layers*(nodes)	Learning rate	Transfer function	Training function	Data train+ test	Accuracy
1*(5)	0.02	tansig	Trainr	50x50	98.5714
2*(10)	0.02	tansig	Trainr	//	98
{1*(10),2*(10)}	0.02	tansig	Trainr	//	98
{1*(10),2*(20), 3*(20)}	0.02	tansig	Trainr	//	97.7143
{1*(10),2*(20),3*(20),4*(20)}	0.02	tansig	Trainr	//	97.1429
{1*(10),2*(20),3*(20),4*(20),5*(20)}	0.02	tansig	Trainr	//	96
{1*(10),2*(20),3*(20),4*(20),5*(20),6*(20)}	0.02	tansig	Trainr	//	96.8571
{1*(10),2*(20),3*(20),4*(20),5*(20),6*(20)}	0.02	tansig	Trainr	//	97.7143
{1*(10),2*(20),3*(20),4*(20),5*(20),6*(20),7*(20)}	0.02	tansig	trainr	//	96.5714

b. Hypothesis Two

This hypothesis is on the training and testing data that how it effect on the performance of neural network and how we can achieve more accuracy through changing the training and testing data.

Hypothesis:

The hypothesis is that ideally, if we increase the training data for neural network, its accuracy should also increase. Because in this way neural network can learn more on more data and should give more accurate result on testing. If the training

Analysis and Experiments:

Here I have done 9 experiments on different training and testing data. From the analysis of these experiments, it is cleared that neural network gives best accuracy on more training data. Experiments on 10 percent training data gave 93% accuracy and on 60-90 percent training data, it gave 98 to 100 percent.

So by experiments it is observed that as we increased the training data, neural network give more accurate result.

Following table shows the experimental results:

Hidden layers(nodes)	Training data	Training Data rows[2,4]	Testing Data	Testing Data Rows[2,4]	Accuracy
3*(10)	10%	[46,24]=70	90%	[412,217]=629	93.1638
//	20%	[92,48]=140	80%	[366,193]=559	95.1641
//	30%	[138,72]=210	70%	[320,169]=489	96.9325
//	40%	[183,97]=280	60%	[275,144]=419	97.1360
//	50%	[229,121]=350	50%	[229,120]=349	97.4212
//	60%	[275,145]=420	40%	[183,96]=279	98.2079
//	70%	[321,169]=490	30%	[137,72]=209	98.5646
//	80%	[367,193]=560	20%	[91,48]=139	99.2806
//	90%	[413,217]=630	10%	[45,24]=69	100

c. Hypothesis Three

This hypothesis is about the effect the of the activation function, learning rate and other parameters for neural network.

Hypothesis:

This hypothesis on the activation function and training functions. The hypothesis is that different activations functions should give different results because they have different types like tansig give values in range [-1 1] and logsig in range [0 1].

The tansig and logsig are opposite to each other so they should give differ results.

Analysis and Experiments:

Experiments shows that if we use tansig or purelin for output and hidden layers. As by experiments, it is shown that tansig and purelin give best results but logsig give less accurate. And training functions like 'trainrb' also increase the time performance.

Hidden layers	Activation functions	Training function	Learning rate	Accuracy
2	2*tansig,tansig	Trainr	0.02	97.7077
2	2*tansig,logsig	Trainr	0.02	34.3840
2	2*logsig,logsig	Trainr	0.02	34.3840
2	2*logsig,tansig	trainrb	0.02	97.9943
2	2*logsing,purelin	Trainr	0.02	97.7077
2	2*purelin,purelin	trainr	0.02	96.2751
2	2*purelin,logsig	trainr	0.02	34.3840

5. Conclusions:

Neural network are mostly being used in classification of breast cancer data analysis. The main goal of this work and experiments was that using neural network technique try to improve the accuracy result of breast cancer. There are many ways to improve the accuracy of neural network through parameters and weights changing etc. Through Experiments in this work, it is observed that training data and activation functions improve the accuracy but more hidden layers did not give more accurate results.

6. References

[1]: Medlineplus.gov. (2017). *Breast Cancer | Breast Cancer Symptoms | MedlinePlus*. [online] Available at: https://medlineplus.gov/breastcancer.html [Accessed 3 Dec. 2017].

[2]. Utomo, C. P., Kardiana, A., & Yuliwulandari, R. (2014). Breast Cancer Diagnosis using Artificial Neural Networks with Extreme Learning Techniques. *International Journal of Advanced Research in Artificial Intelligence(IJARAI), 3*(7), 10–14. Retrieved from http://thesai.org/Publications/ViewPaper?Volume=3&Issue=7&Code=IJARAI&SerialNo=3
http://doi.org/10.14569/ijarai.2014.030703

[3]: Innoarchitech.com. (2017). *Cite a Website - Cite This For Me*. [online] Available at: https://www.innoarchitech.com/artificial-intelligence-deep-learning-neural-networks-explained/ [Accessed 3 Dec. 2017].

[4]: Cs231n.github.io. (2017). *CS231n Convolutional Neural Networks for Visual Recognition*. [online] Available at: http://cs231n.github.io/neural-networks-1/ [Accessed 3 Dec. 2017].

.