12/3/2017

Faisal Imran

Namal college, mianwali

NNFS CW1

Breast cancer classification using Neural Network Approach

# **1 Introduction:**

Breast cancer is the most common cancer in women both in the developed and less developed world. It is estimated that worldwide over 508 000 women died in 2011 due to breast cancer [1]. There are a few signs of breast cancer which can help a person to detect either someone has cancer or not.

Importance of computer in human life can’t be denied, computers are now being used in medical filed to detect, diagnose and suggest medicines using Machine Learning. For this particular problem, we have to train a neural on a given dataset (UCI Machine Learning Repository) so that it can detect the breast cancer.

# **2 Background:**

In the given dataset the total number of instances are 699, having 11 attributes. Attribute no. 1 is the sample code number and no. 11 classifies the case as 2 for Benign and 4 for Malignant Cancer. On this dataset we have to train our neural network.

The founding father of neural networks describes neural networks 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 basic building block of a neural network is a neutron.

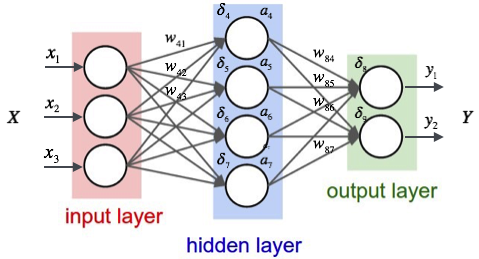
A neural network learns by changing weights, it changes its weights based on this formula.

**W (t+1) = W(t)+(learning rate)\*( Desired Output -Actual output) \*(bias)**

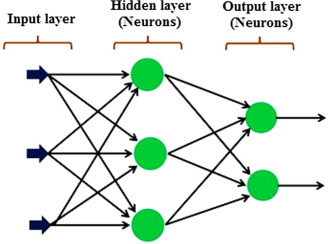
## **2.1 Neural Network**

A neural network consists of different layers input layer, hidden layers (Number of hidden layers may differ depending upon the problem) and the output layer. In these layers the neurons are inter-connected and connected to other layers and they can transmit signals to each other. There are neutrons that can process these signals and again pass them to other neurons. Neurons learns through an example, same as the human learn.

Moreover, architecturally, neural networks are of two types. **1)** Feedforward **2)** Feedback Network

[3]

## **2.2 Feed-forward Network**

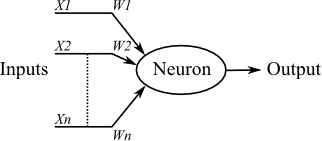
[4]

As it is being manifest from figure and name, in feed-forward networks the signal travels in one direction; from input to output. The output of any layer can’t effect the same layer as there are no feedback loops. In problem like pattern recognition feed-forward neural networks are usually preferred.

## **2.3 Feedback Network**

Interactive and recurrent are the names for feedback network because signals in these networks can travel in both directions, therefore, they can be really complicated. Feedback is defined to occur when the (Full or partial) output of a system is routed back into the input as part of an interactive cause-and-effect process [5]. These networks can change their state until they reach to an equilibrium point but they will change their state when a new set of inputs will be entered.

## **2.4 Neurons**

[6]

Neuron is the building block of artificial neural network, having many inputs and one output. Each neuron works in two modes training mode and using mode. Neurons are trained to fire at a particular point for a particular set of inputs depending upon the strength of input signal. Neuron uses transfer function (to sum the input which are multiplied by weights) and the activation function decides the output. Like a binary activation function can decides either output would 1 or 0.

## **2.5 Activation Function**

Activation is related to a node (Neuron), it will decide the output of a node depending upon the nature of input. Activation function are of different types like Binary function which will decide either input should be 0 or 1 depending upon the input.

# **3 Main Part:**

Main part of the report contain the methodology developed for the given problem.

## **3.1 Data Collection**

According the coursework requirement the data set for breast cancer classification was used from the UCI Machine Learning Repository. As discuss earlier it has 699 instances, having 11 attributes. First attribute is the ID, last attribute is the cancer type (2 for Benign and 4 for Malignant) and all other attributes are the Disease signs. Out of 699 instances, 13 instances have some missing values.

## **3.2 Data Import into MATLAB**

As discuss in above paragraph data were having some instances which were not completed (Missing values). Question marks were there in the missing values columns so MATLAB was unable to load it. There were a few to tackle with this problem.

* Instances could have been removed from the data as they were only 13 instance out of 699
* Question could have been removed with a random value
* Mean value could replace the question mark

For sake of this problem all above mentioned methods were tried and the method of mean value was chosen because accuracy was best while using mean value method. In the mean value method all 699 instances were used as compare to method were missing instances were removed that result into more accuracy as there was more data to learn. Mean value was calculated by this formula

**Mean Value = Sum (7th** **Column) / Total no. of instances**

**3.5** was the value which was calculated by this formula so in the data “**?**” was replaced with 3. By rounding off 3.5 to 4, 4 was checked too but it gave the same accuracy so 3 was kept in the data.

After importing data to MATLAB, data was divided into two component **inputData** and **outputData**. In the input data there are 9 columns and all rows **(inputData = data (: 2, 10))** and outputData have last column and all rows **(outputData = data(:,11)).**

## **3.3 Creation, Training and Testing of Network**

% Creating Network

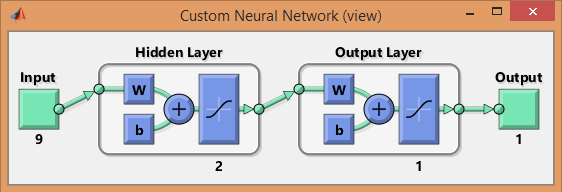
net = newff(inputData',outputData',2, {'tansig' 'tansig'}, 'trainr', 'learngd', 'mse');

net.trainParam.goal = 0.01;

net.trainParam.epochs = 100;

net.trainParam.max\_fail=10;

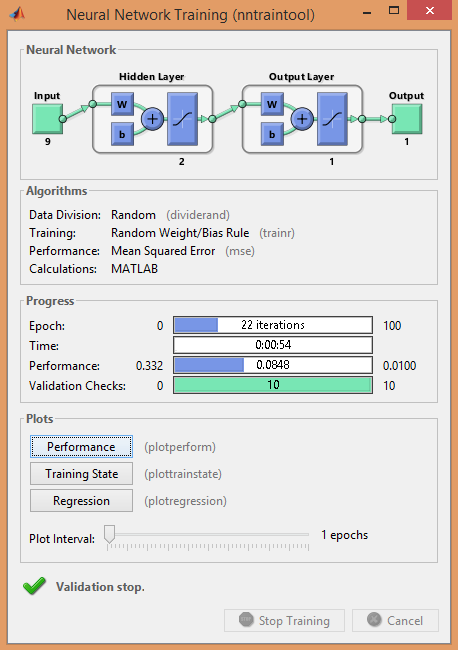
A new neural network was created using newff function, after data was successfully imported into MATLAB memory. Different attributes of this network were set like validation checks, goal and epochs and inputData and outputData was passed as a parameter, moreover, number of hidden layers were defined. The architecture of the network was completed, now it was ready to be trained.



% Training Network

net = train(net, inputData',outputData');

Now the network was passed to the train function so that it can trained.



After training it was tested on the data and gave the result in the form of matrix.

## **3.4 Calculation of Accuracy**

After testing result was compared to the original matrix and error were counted, to calculate the accuracy by following formula.

**Performance = Total No. of Instances - Erroneous Instances / Total No. of Instances \* 100;**

# **4 Experimental Results and Analysis:**

In science, it is common to observe, make hypothesis and then run experiments to see, if hypothesis is right or wrong. Over here we will make a hypothesis on a parameter like Number of hidden layers, data distribution and the validation checks then we shall do experiments to prove it right or wrong.

* **All experiments will be run on a default setting in which no. of hidden layers =1, epochs = 100, validation checks = 100, goal = 0.01 and we are using full data for learning and testing.**

### **Hypothesis No. 1: Increasing the number of hidden layers will bring more accuracy, as it add more neurons, more neurons means more accuracy.**

### **Experiment and Analysis:**

Five different experiments were run for this hypothesis, all other variable were kept constant. Below table shows the results of experiments.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment No.** | **No. of Hidden Layers** | **Accuracy** | **Time taken on Learning** |
| 1 | 1 | 98.42 | 1:12 |
| 2 | 2 | 98.42 | 1:15 |
| 3 | 3 | 98.14 | 51 sec |
| 4 | 10 | 97.71 | 1:31 |
| 5 | 50 | 97.28 | 30 sec |

As there are no hard and fast rules for number of hidden layers, so we did some experiments to check our hypothesis. It can be seen from the table that by increasing number of hidden layers decreases the accuracy but wasn’t dependent upon the number of hidden layers, it is dependent upon the number of iterations and weights.

Our hypothesis wasn’t right so for sake of this problem 1 or 2 hidden layers are enough, as, maximum accuracy was seen for 1 and 2 numbers of hidden layers.

### **Hypothesis no. 2: Increasing number of validation checks will improve accuracy.**

### **Experiment and Analysis:**

Hypothesis was to check whether the number of validation have any effect on accuracy or not. Intuitively, it looks like it might increase the accuracy of system but when different were run on different number of validation sets, accuracy seems to be decreasing but for a large number of validation, neural network took a lot more time than usual.

|  |  |  |
| --- | --- | --- |
| **Experiment no.** | **No. of Validation Checks** | **Accuracy** |
| 1 | 2 | 98.85 |
| 2 | 6 | 98.56 |
| 3 | 10 | 98.48 |
| 4 | 20 | 98.14 |
| 5 | 50 | 97.14 |

As it can be seen from above table that as the number of validation checks keeps on increasing the accuracy of neural network started to decrease fractionally but after a point it became constant, no matter how many validation checks are there.

### **Hypothesis no.3: More training data will result in more accuracy, as neural network will have more examples to train on.**

### **Experiment and Analysis:**

In this hypothesis, we will divide the data into different parts, like we will train the neural network on 10% of the total data and network will be checked on 90% data. Then we will keep on increasing the training data by 10% and will check the neural network on the remaining data. In this way neural network will be checked on the unseen data which is the case in real world. If the neural network gives the good accuracy in that condition then it would a great solution.

* **10% training data, 90% testing data**

In this case training data was from 1-70 and testing data was 71-699 which result in **accuracy 97.45.**

* 20% training data, 80% testing data

In this case training data was from 1-140 and testing data was 141-699 which result in **accuracy 98.2.**

* 30% training data, 70% testing data

In this case training data was from 1-210 and testing data was 211-699 which result in **accuracy** **97.34**.

* 40% training data, 60% testing data

In this case training data was from 1-280 and testing data was 281-699 which result in **accuracy 97.37.**

* 50% training data, 50% testing data

In this case training data was from 1-350 and testing data was 351-699 which result in **accuracy 99.42.**

* 60% training data, 40% testing data

In this case training data was from 1-420 and testing data was 421-699 which result in **accuracy 99.49.**

* 70% training data, 30% testing data

In this case training data was from 1-490 and testing data was 491-699 which result in **accuracy 99.04.**

* 80% training data, 20% testing data

In this case training data was from 1-560 and testing data was 561-699 which result in **accuracy 100.**

* 90% training data, 10% testing data

In this case training data was from 1-630 and testing data was 631-699 which result in **accuracy 100.**

This hypothesis seems to be true as the results are surprising, on less training data the accuracy of system was low but it kept on increasing while training data was increased. As discussed earlier neural network are like human brain, they tend learn by examples, and by increasing data means there is more learning but there could be another possibility that when the learning data as increased, the testing data was decreasing at the same time, this might be the reason behind the 100% accuracy that we have to check a lot instances.

Finally, it can be concluded from the experiments and analysis that there are hard and fast rules for the parameters of a neural network, like no one say that how much hidden layers should be there in a neural network, it purely depends upon the nature of the problem.

# **5 Bibliography and Citations:**

[1]"Breast cancer: prevention and control", *World Health Organization*, 2017. [Online]. Available: http://www.who.int/cancer/detection/breastcancer/en/index1.html. [Accessed: 02- Dec- 2017].

[2]"A Basic Introduction To Neural Networks", *Pages.cs.wisc.edu*, 2017. [Online]. Available: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [Accessed: 02- Dec- 2017].

[3]"Creating a Neural Network from Scratch — TensorFlow for Hackers (Part IV)", *Medium*, 2017. [Online]. Available: https://medium.com/@curiousily/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8. [Accessed: 02- Dec- 2017].

[4]2017. [Online]. Available: https://www.researchgate.net/figure/301344011\_fig13\_Multi-layer-feedforward-network. [Accessed: 02- Dec- 2017].

[5] A. R. Zamir, T.-L. Wu, L. Sun, W. Shen, J. Malik, and S. Savarese, “Feedback Networks,” 2016.

[6]"Chegg.com", *Chegg.com*, 2017. [Online]. Available: http://www.chegg.com/homework-help/definitions/artificial-neural-network-3. [Accessed: 02- Dec- 2017].