Breast Cancer classification using Neural Network Approach

An approach to come up with an accurate and time efficient trained Neural Network for classifying the breast cancer.

Umer Awais Malik - 14031198
NAMAL COLLEGE, MIANWALI.

Introduction

One of the main reasons of Breast Cancer is the growth of **malignant cells** in breast. Other than skin cancer, it is the most common type of cancer in the women that is affecting a number of ladies across the globe and the battle against cancer is a long way from finished. This disease mainly occurs after the age of 40-50. Throughout the ages, nobody has found exactly what causes the breast cancer but the Ancient Greeks, for example, believed that imbalances of bodily humors (fluids, especially black bile) were responsible for breast cancer [6]. But still the work is ongoing to reach the roots of this problem.

This disease has been around for as long as humans but initially, it wasn't cured/handled properly. As the time progressed and the scientists and researchers started putting in some valuable efforts, they started to find its remedies. Early remedies for Breast Cancer were aimed at offering temporary relief or prolonging life rather than attempting to cure the disease but as the work in this field progressed, the breast cancer treatments started to become common, once the scientists began to establish the relationship between breast cancer and genes. They started viewing it as a localized disease which could be removed before it spreads [6].

The process of breast cancer detection is carried out by the help of a technique called mammography. The actual test is called a **mammogram**. **Mammography** (also called **mastography**) is the process of using low-energy X-rays (usually around 30 kVp) to examine the human breast for diagnosis and screening. The goal of mammography is the early detection of breast cancer, typically through detection of characteristic masses or micro calcifications [7] [8] [9].

All of the work on breast cancer classification conducted as yet results in making accurate predictions and those too by computer machines. Computer machines are dumb; how could they make predictions? Artificial Intelligence and data mining play a major role in this regard. Medication propels on all fronts to improve the care of patients and to overcome this illness has been done consistently and to provide assistance to the medical, robust and reliable diagnosis, neural networks can be a powerful tool for distributed diagnosis [3]. Recently, the neural network has become a popular tool in the classification of Cancer Dataset [1] [2] [4] [5]. This is particularly due to its ability to represent the behavior of linear or nonlinear functions multidimensional and complex [11].

Coming on to the given problem, we have breast cancer data which was collected from the University of Wisconsin Hospitals, Madisons from Dr. William H. Wolberg and we have Neural Networks from Artificial Intelligence domain. We have to come up with a trained Neural Network which gives us a better/accurate classification rate.

Background

Breast Cancer Classification

Breast cancer is mainly classified into two types, that is, **Benign** (Non-Cancerous) and **Malignant** (Cancerous). Unlike cancer tumors, a non-cancerous tumor (Benign) is unable to spread throughout the body effecting different parts of the body, rather it responds positively to a treatment and can be cured. But such a tumor can be genuinely threatening, if it is effecting a primary nerve, a main artery or compresses brain matter [10].On the other hand, a cancerous (Malignant) tumor is a type of tumor that

has the ability to metastasize (spread) to various parts of the body and invade/effect the surrounding tissues. Malignant tumors are formed from abnormal cells that are highly unstable and travel via blood stream, circulatory system and lymphatic system [10].

Neural Network Models

Neural networks are basically some computer programs designed to simulate the way human brain processes information. They are trained though experiences, that is, by detecting different patterns and relationships in the data. A neural network "is formed from hundreds of **single units**, **artificial neurons** or **processing elements (PE),** connected with coefficients **(weights)**, which constitute the neural structure and are organized in layers. The power of neural computations comes from **connecting neurons** in a network. Each PE has **weighted inputs, transfer function and one output**. The behavior of a neural network is determined by the transfer functions of its neurons, by the learning rule and by the architecture itself. The weights are the adjustable parameters and, in that sense, a neural network is a parameterized system. The weighed sum of the inputs constitutes the activation of the neuron. The activation signal is passed through transfer function to produce a single output of the neuron. Transfer function introduces non-linearity to the network. During training, the inter-unit connections are optimized until the error in predictions is minimized and the network reaches the **specified level of accuracy**. Once the network is trained and tested it can be given new input information to predict the output" [12].

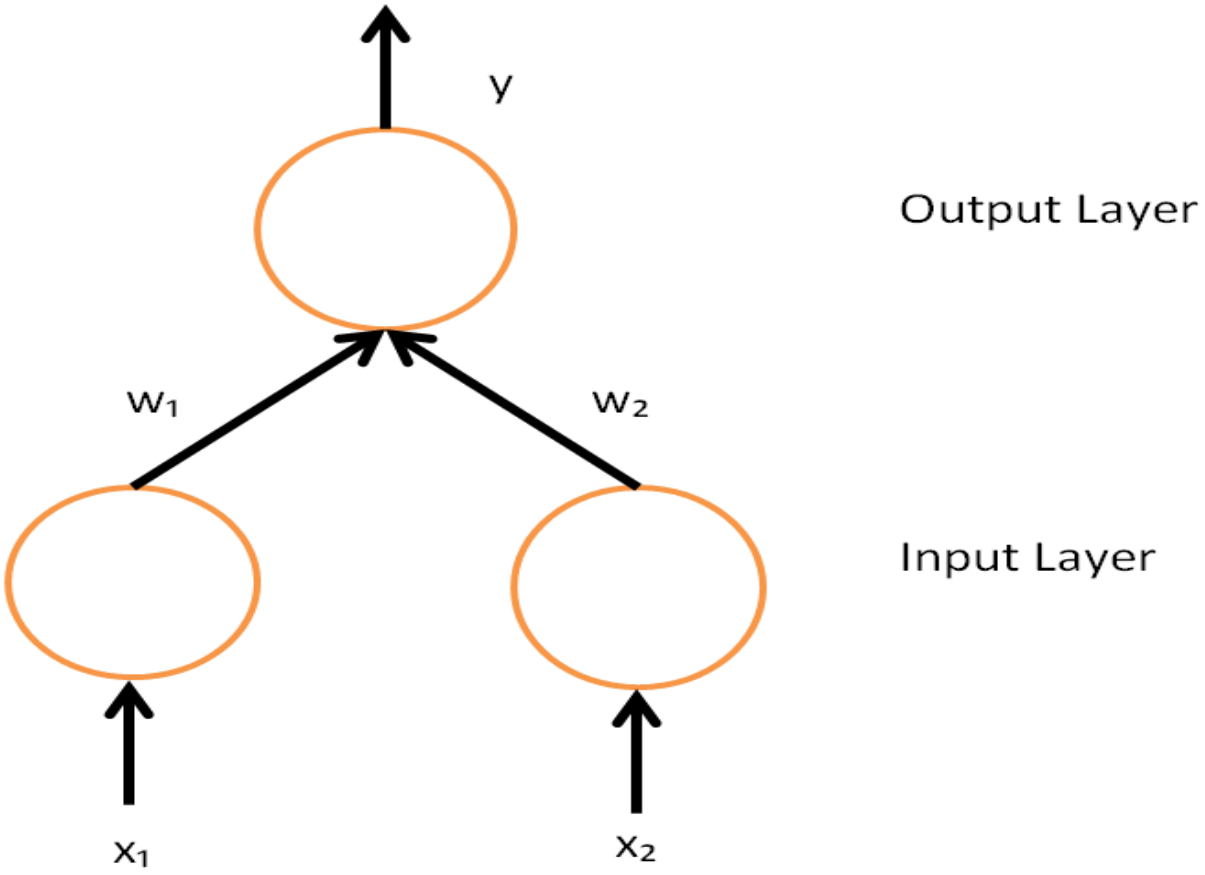
A neural network can be regarded as a **network of neurons** organized in layers defined below:

- **Input Layer:** This layer takes inputs and attaches some coefficients known as weights to the given inputs.

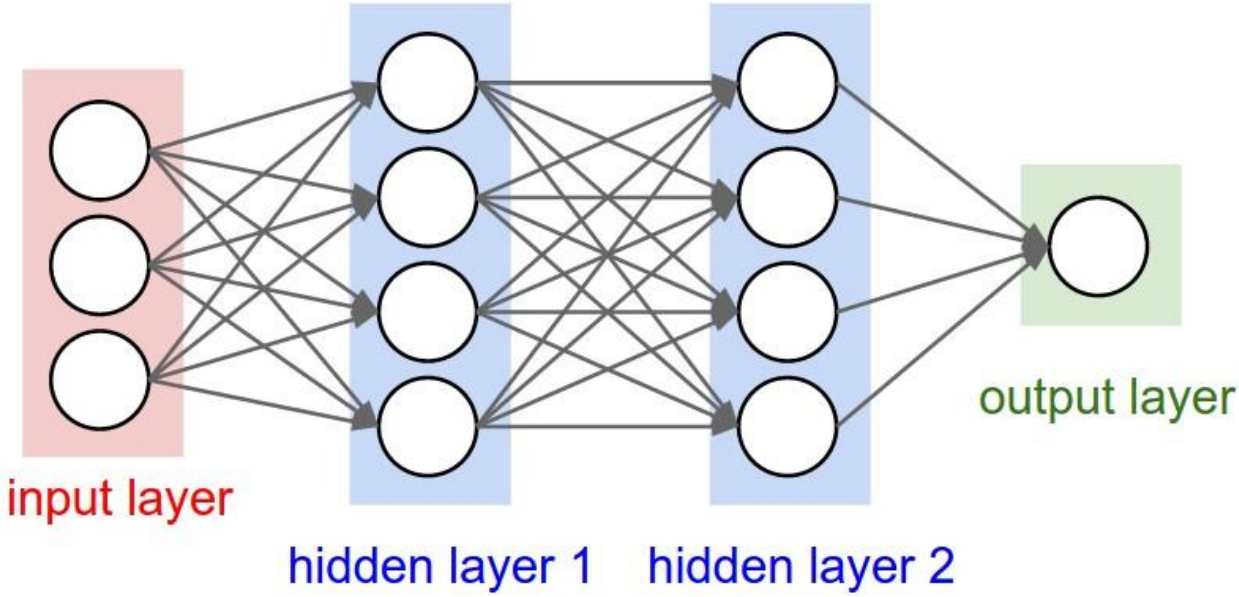
- **Hidden Layer:** This layer introduces non-linearity to the neural network and improves the performance of the network. A neural network can also be linear, having no hidden layers.

- **Output Layer:** As the name indicates, this layer generates the results on the basis of weighted inputs and hidden layers.

Shown below are the general diagrams of linear and non-linear neural networks.



: - **Network without Hidden Layers**



: - **Network with Hidden Layers**

Some Basic Concepts:

The development tool used to solve the given problem is named Matlab. It is a very effective and powerful development tool and provides a range of built-in feasibilities to the user. In order to test the solution I have suggested, one must have Matlab tool installed on his machine and should have a better know how of programming in Matlab.

To demonstrate a better understanding of how the problem was approached and solved, some useful concepts used in the solution are described below:

I. **Import Data from a Text File:**
The dataset used for experimentation was obtained from University of Wisconsin Hospitals, Madisons from Dr. William H. Wolberg. The data was stored in a text file and to load that text file in our program, 'importdata' method was used. This method is used to import data from the file. It receives an argument of type string which is the name of the file and returns a multidimensional array. In order to get a better understanding of this method or any built-in method in Matlab, use the "help" command.

II. **Array Manipulation:**
   a. Array creation in Matlab is the same as it is in some other programming languages and tools, that is, creating a variable and then assigning a set of values to it. For example, **List = [1, 2, 3, 4, 5, 6].**
   b. Indexing in Matlab is slightly different from normal indexing convention. Unlike other indexing convention where indexing starts from 0 (zero), indexing in Matlab starts from 1 and List (1) will return me the first element of the list, that is, 1.

   b. Getting a range/group of values is also very different in Matlab. When we need a specific values from a specific row and column, we give the row number and column number separated by a comma but if we need some specific values from a column than we define the range of required values. For example, **List1 = List (1, 2:5).** The range is defined by giving the starting and ending indexes separated by colon, which represents the range. In the example mentioned above, List1 will have values [2, 3, 4, 5].

III. **Creating, Setting, Training, and Testing Neural Network:**

- **Creating a Neural Network with newff method:**

  Following command is used to create network, which can be configured.

  net = newff (Input_Data, Targeted_Data, Hidden_Layers, TF, BTF, BLF, PF);

  where [13]:

  Input_Data is the training dataset on which we want to train our network.

  Targeted_Data is the resultant/output of Input_Data on which we want to map our given outputs.

  Hidden_Layers is a variable quantity and can be altered according to our requirement. It defines how many hidden layers to create and its function is to introduces non-linearity to the neural

network and improve the performance of the network. A neural network can also be linear, having no hidden layers.

TF is a Transfer Function. We set it to {'tansig' 'tansig'}.

BTF is Backpropagation Training Function. We set it to 'trainr'.

BLF is Backpropagation Learning Factor. We set it to 'learngd'.

PF is Performance Function. We set it to 'mse'.

- **Setting Parameters of the network, returned by newff:**

  Out of all the parameters returned by newwff function, following are the ones which we will be altering throughout the training of the network. That is,

  net.trainParam.epochs is used to set maximum numbers of epochs/iterations to train the network [14].

  net.trainParam.goal is used to set the performance goal [14].

  net.trainParam.max_fail is used to set maximum validation failures [14].

- **Training the Configured Network:**
  The following function is used to train the network, that is,

  net = train(net, Input_Data, Targeted_Data);

  This function returns a trained network which can be used for testing different testing datasets, according to our requirement.

- **Testing the trained network:**

  The function used to test the trained network is given below. It returns a set of outputs corresponding to the set of inputs given as **Input_Data**.

  results = net(Testing_Data);

  Testing_Data is the dataset on which we want to test our neural network.

Methodology

Pre-processing:

There were four things (Pre-processes) we did before proceeding to our method which are listed below:

- Removing 16 Values from our Dataset.
- Sorting.
- Equal distribution of each category.

- Data Patches.

Removing 16 values from Our Dataset:

In our given dataset, there were sixteen (16) values which contained a question mark (?) instead of an integral value of a symptom. We had three options to consider, that is, either take the mean of the column and then replace the question mark with the mean value, place an integer other than 1-10 or the third one was to remove all those rows having a question mark in them. We decided to choose the third option and removed all the value having a question mark, so that we don't have to face any problem during our training and testing of the neural network on the given dataset. So, after removing these values, we had a total of 683(rows) values to consider for training and testing.

Sorting:

Last column of every row has a value, that is, either 2 or 4, representing the state of cancer (either benign or malignant). Out of 683 values (rows) in our dataset, we had 239 rows having 4 in the last column and 444 (rows) had value 2 in their last column. We decided to sort the data on the basis of last column so that we can separate out 2's and 4's and all the same values (either 2 or 4) would now be present one after the other. This convention would help us in deriving the method/ technique we followed in training our neural network.

Equal Distribution of Each Category:

We decided to use the same amount of values for 2's and 4's while training the data. Our understanding is to train the same amount of data of both the symptoms.

Data Patches:

We have divided the dataset into different chunks of data having the same amount values of 2's and 4's. We would be training our neural network by giving these chunks of data as training data.

Our Method

As the details of what we will be changing and the conventions we will be using as we move along are described above. The flow of our method along with the alterations in different parameters will be described in this part.

The first thing we did was to remove the rows containing question marks and then sort the data. The reason of sorting was to make the access to the same values, that is, either 2's or 4's, easy. Once the data was sorted, we removed the first and last column from every row and the only thing we were left with in our dataset were the symptoms. The data patches we made had to have equal number of values of both the categories (2's and 4's). As the data was sorted so we decided to take the first half of the patch from the start and the other half of the patch from the bottom. The patch taken from the start only contained those samples which had the value 2, in their eleventh (11^{th}) column and the other half of the patch which was taken from the bottom (starting from the last value and then moving up in the data) only contained the samples with value 4 in their eleventh column. We concatenated both halves of the path to make up a single patch of training data having equal amount of values for both the states of cancer. The eleventh column was saved separately so that we could compare the trained values with the original values to find the precision of our neural network. As the neural network was returning values in floating points, so we decided to place a threshold, that is, the floating point values less than 3 to be converted to 2 and those greater than 3 to be converted to 4. In this way all the resulting values were either 2's or 4's.

This convention helped us in comparing the trained values with the original values and in determining the precision and accuracy of our neural network. Other than all of what has been described so far, we also altered the hidden layers, Param.epochs/number of iterations, Param.max_fail/maximum validation failures and Paramgoal/performance goal to check and increase the precision of our neural network.

Experiment and Analysis

0. This is a trail experiment with following settings

Data_Size = 250, Epochs = 50, Goal = 0.05, Maxfail = 50.

Results: The experiment completed in 39 seconds and resulted in an accuracy of 97.22%.

1. Our random experiment turned out to give a very good accuracy. Let's try to improve the previous settings so that we may get some better results. For now, I believe that Data_Size would matter the most in experiments. **Accuracy will increase as the Data_Size is increased**.

Data_Size = 350, Epochs = 50, Goal = 0.05, Maxfail = 50.

Results: The experiment completed in 56 seconds and resulted in an accuracy of 94.895. **Our hypothesis failed** which means data is not the only factor to improve the performance.

2. If data is not the only factor of improvement in the accuracy, let's change the Epochs and Maxfail which were not completely reached in previous experiments. **This is for sure that they will not affect our performance.**

Data_Size = 350, Epochs = 25, Goal = 0.05, Maxfail = 25.

Results: The experiment completed in 30 seconds and resulted in an accuracy of 96.09%. This is not what we hypothesize and our hypothesis failed, as it is clearly seen that Epochs and Maxfail affected the performance and time efficiency.

3. Epochs and Maxfail affected out performance. 2nd experiment showed that data does not affect the performance as such, so we will change the Goal this time and we will set it to an increased value and this is acknowledged that with greater goal, performance will be improved and the time efficiency can be decreased.

Data_Size = 350,
Epochs = 25,
Goal = 0.09,
Maxfail = 25.

Results: The experiment completed in 26 seconds and resulted in an accuracy of 96.09%. Our accuracy did not increase and the time efficiency has increased, so our hypothesis failed somehow. Time efficiency increased because our Goal met earlier in 23 Epochs completions.

4. Previous experiments have improved the time efficiency and now we will try to improve our accuracy. **With such an increased Goal, increased Data_Size will increase our performance.**

Data_Size =450,
Epochs = 25,
Goal = 0.09,
Maxfail = 25.

Results: The experiment completed in 22 seconds and resulted in an increased accuracy of 98.28%. We hypothesized the performance of this experiment correctly and we also got the increased time efficiency. This means with an increased Goal and Data_Size, we can get a better performance.

5. As stated in previous experiment, we will increase both Epochs and Data_Size in the experiment and our performance will definitely increase.

Data_Size = 550,
Epochs = 25,
Goal = 0.15,
Maxfail = 25.

Results: The experiment completed in 1 second and resulted in a little increased accuracy of 98.89%. We are right about the increment of Goal and Data_Size and we can conclude that performance of our network is directly proportional to Goal and Data_Size.

Conclusion

We can hereby conclude that our performance depends on Goal and Data_Size and somehow a little dependent on Epochs and Maxfail. We have come up with a trained network which has an accuracy of 98.89% and is very time efficient. So the best configuration of our trained network is in the 5^{th} experiment.

References

[1] A. Cichocki and R. Unbehauen, '' Neural Networks for optimisation and signal processing,'' J. Wiley, Sons Ltd. And B.G. Teubner, Stuttgart, 1993.

[2] Abdelaal Ahmed, Mohamed Medhat and FarouqWaelMuhamed, "Using data mining for assessing diagnosis of breast cnacer," in Proc. International multiconference on computer science and information Technology, 2010, pp. 11-17.

[3] Arun George Eapen, master thesis, Application of Data Mining in Medical Applications, Waterloo, Ontario, Canada, 2004.

[4] BellaachiaAbdelghani and ErhanGuven, "Predicting Breast Cancer Survivability using Data Mining Techniques," Ninth Workshop on Mining Scientific and Engineering Datasets in conjunction with the Sixth SIAM International Conference on Data Mining," 2006.

[5] Burke H. B. Et al, "Artificial Neural Networks Improve the Accuracy of Cancer Survival Prediction", Cancer, 1997, vol.79, pp.857-862.

[6] Brechon, S. (2017). *A Brief History of Breast Cancer*. [online] Maurer Foundation. Available at: https://www.maurerfoundation.org/a-brief-history-of-breast-cancer/ [Accessed 28 Nov. 2017].

[7] Anon, (2017). [online] Available at: https://medical-dictionary.thefreedictionary.com/mammography [Accessed 30 Nov. 2017].

[8] Cdc.gov. (2017). *CDC - What Is a Mammogram? - Breast Cancer*. [online] Available at: https://www.cdc.gov/cancer/breast/basic_info/mammograms.htm [Accessed 30 Nov. 2017].

[9] (ACR), R. (2017). *Mammography (Mammogram)*. [online] Radiologyinfo.org. Available at: https://www.radiologyinfo.org/en/info.cfm?pg=mammo [Accessed 30 Nov. 2017].

[10] Bollinger, Ty. (2016). Benign and Malignant: What is the difference? . [online] The Truth About Cancer. Available at: https://thetruthaboutcancer.com/benign-malignant- tumors-difference/ [Accessed 27 Nov. 2017].

[11] Acit2k.org. (n.p.). Cite a Website - Cite This For Me. [online] Available at: http://www.acit2k.org/ACIT/2012Proceedings/13233.pdf [Accessed 3 Dec. 2017].

[12] Agatonovic-Kustrin, S. and Beresford, R. (2017). *Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research*.

[13] Radio.feld.cvut.cz. (n.p.). newff (Neural Network Toolbox). [online] Available at: http://radio.feld.cvut.cz/matlab/toolbox/nnet/newff.html [Accessed 3 Dec. 2017].

[14] Mathworks.com. (n.d.). Levenberg-Marquardt backpropagation - MATLAB trainlm - MathWorks United Kingdom. [online] Available at: https://www.mathworks.com/help/nnet/ref/trainlm.html [Accessed 3 Dec. 2017].