



[7]

Breast cancer classification using Neural Network Approach

Name# Muhammad Ali Khalid

UoB# 14031211

Module Instructor# Dr. Junaid Akhtar

Contents

Breast cancer classification using Neural Network Approach	1
1. Introduction:	3
2. Background:	3
3. Main Part:	3
3.1. Data Gathering:.....	4
3.2. Pre-processing:.....	4
3.3. Creating and Training Network:.....	4
3.4. Post Processing:	4
4. Experimental Results and Analysis:	5
4.1. Generic Parameters:	5
4.2. Hypothesis 1:.....	5
4.2.1. Training and Results old:.....	5
4.2.2. Training and Results new:	5
4.2.3. Analysis:	6
4.3. Hypothesis 2:.....	6
4.3.1. Training and Results:.....	6
4.3.2. Analysis:	7
4.4. Hypothesis 3:.....	7
4.4.1. Training and Results:.....	7
4.4.2. Analysis:	8
4.5. Hypothesis 4:.....	8
4.5.1. Training and Results:.....	8
4.5.2. Analysis:	8
4.6. Hypothesis 5:.....	9
4.6.1. Training and Results:.....	9
4.6.2. Analysis:	10
5. Some Observations:	10
6. Conclusion:.....	10
7. Bibliography:	10

1. Introduction:

“Breast cancer is a malignant tumor that develops when cells in the breast tissue divide and grow without the normal controls on cell death and cell division [2,3]. It is the most common cancer among women [1].”[1]

Breast cancer is very common and life taking disease in women. “Breast cancer is the second leading cause of cancer death in women”[2]

“Although breast cancer is the second leading cause of cancer death in women, the survival rate is high. With early diagnosis, 97% of women survive for 5 years or more [3,10].”[1]

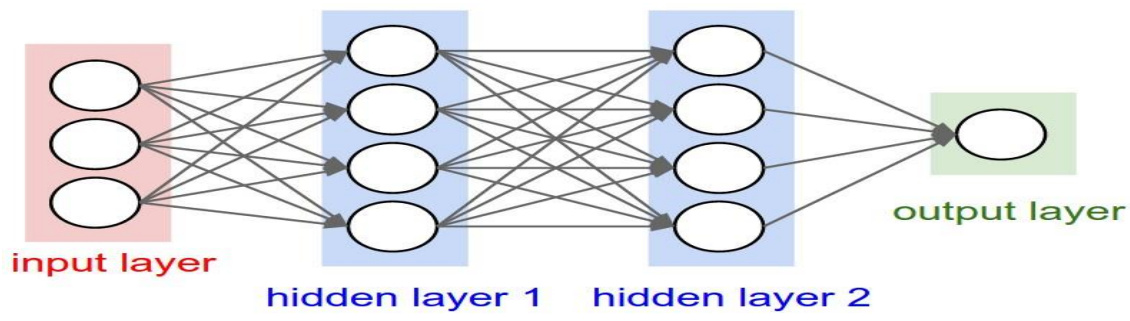
There are many different research works and efforts are in pipe line and a lot of systems which are successfully deployed for the detection of breast cancer in women. In this report I have projected that how neural networks could be more helpful in this field.

2. Background:

The idea of bringing Neural Networks in computer field is inspired by the actual complex neural network within human brain. “The human brain is arguably one of the most exciting products of evolution on Earth.”[3]. “The beginning of Neurocomputing is often taken to be the research article of McCulloch and Pitts [10] published in 1943, which showed that even simple types of neural networks could, in principle, compute any arithmetic or logical function, was widely read and had great influence. Other researchers, principally Norbert Wiener and von Neumann, wrote a book and research paper [11, 12] in which the suggestion was made that the research into the design of brain-like or brain-inspired computers might be interesting.” [4]

3. Main Part:

Neural networks are best in this way that I could easily train the system and then test different cases on it and visualize the bigger picture of the whole system.[5]




Here I am going to tell about the methodology which I followed to develop neural network and will also justify it. Here I would also highlight the each step undertaken from gathering procedural data to processing and training so that if someone likes to implement this whole thing, he could easily reverse engineer it.

3.1. Data Gathering:

I used a dataset provided at UCI Machine Learning dataset repository. There are 11 columns in the given dataset where 1st column is identification number of patient and then the next 9 columns are different symptoms of breast cancer and the last column has only two values (2 and 4); where 2 means benign and 4 means malignant. There are total 699 entries. There are 16 rows where some data is missing and mark of ? is there.

3.2. Pre-processing:

Data loaded to Matlab manually by converting the extension .txt file into .m file and putting **square bracket** around it and assigning it to a variable. Data needed preprocessing before sending it to the neural network for training. The **patient id** in data was useless so the first column of data is dropped intentionally. Then there were some missing values in 16 rows for which I decided to remove those rows because I was unable to decide from taking Mean, fitting random values or neighbor values what would be the best substitute for them. 

The last attribute of data set also required processing because I was intended to use transfer function 'tansig' which has range from -1 to 1 but output values in last column were 2 or 4 which are **out of range**, so in order to bring them in range, 2 was represented with 0 and 4 with 1. Following piece of code is used to replace all 2 and 4 with 0 and 1.

```
outputData(find(outputData == 2)) = 0;  
outputData(find(outputData == 4)) = 1;
```

3.3. Creating and Training Network:

After all preprocessing when data set was ready a neural network was build using **newff** function(feedforward) in matlab, then it's different attributes were set like learning rate, goal, epochs, activation function etc. Once network architecture was built, it was sent to train function with training data in order to train the network. After training, it was then sent to '**sim**' function of matlab with testing data to test the network.

3.4. Post Processing:

After training and testing a function was written to measure the accuracy of network. Following formula was used to calculate the accuracy.

```
accuracy = (counterlength(netOutput)) * 100;
```

And then printing the accuracy by rounding of the results using the following piece of code.

```
fprintf('Accuracy = %i%%' , round(accuracy));
```

4. Experimental Results and Analysis:

4.1. Generic Parameters:

Following are the generic parameters which would replace the hypothesis parameters after getting results.

Hidden Layers: 20
Epochs: 100
Validation Checks: 10
Goal: 0.01

4.2. Hypothesis 1:

First thing first, I assume that with the increase in number of hidden layers, the neural network will learn more. So I doubled the number of hidden layers from 20 to 40 so that to gain more accuracy.

4.2.1. Training and Results old:

The previous results gained on 20 hidden layers are shown first table, and then in next table the results are shown where hidden layers are 40.

Hidden Layers: 20

Epochs: 100

Validation Checks: 10

Goal: 0.01

Training Data	Testing Data	Iterations	Accuracy
10%	90%	22	96%
20%	80%	75	97%
30%	70%	40	96%
40%	60%	43	97%
50%	50%	15	99%
60%	40%	35	97%
70%	30%	26	100%

4.2.2. Training and Results new:

Hidden Layers: 40

Epochs: 100

Validation Checks: 10

Goal: 0.01

Training Data	Testing Data	Iterations	Accuracy
10%	90%	9	96%
20%	80%	19	97%
30%	70%	10	97%
40%	60%	23	98%
50%	50%	8	99%
60%	40%	16	98%
70%	30%	9	99%

4.2.3. Analysis:

After comparing the results of old version when hidden layers were 20, and when hidden layers becomes 40, I noticed that in all cases this hypothesis works and gave better or equivalent accuracy except the last one where accuracy dropped one percent which force me to get back from my hypothesis and make next generation of hypothesis. It should have given the more or equivalent accuracy as it was giving before but it didn't. I am unable to visualize the solution for this strange result.

4.3. Hypothesis 2:

As previous assumption was wrong (not fully) which leads me to make new hypothesis. This time I generated hypothesis that with the increase in number of Epochs, the neural network would be able to make more iterations as a result of which it would output more accuracy.

So I doubled the Epochs from 100 to 200.

4.3.1. Training and Results:

Hidden Layers: 20

Epochs: 200

Validation Checks: 10

Goal: 0.01

Training Data	Testing Data	Iterations	Accuracy
10%	90%	12	97%
20%	80%	40	97%

30%	70%	30	97%
40%	60%	36	97%
50%	50%	13	98%
60%	40%	17	97%
70%	30%	30	99%

4.3.2. Analysis:

The above results are based on my hypothesis which are very diverse as in **two** cases, the accuracy is decreased, in **three** cases the accuracy remains same and in **two** cases the accuracy increased as well when compare to the output of generic parameters. This is very complex situation where results are totally different and there is no pattern in them. This results the same way as our brain do which is uncertainty. As artificial neural networks were made to work same way as our brain works. Also the number of iterations in every case decreases except one.

4.4. Hypothesis 3:

Giving another try and this time I assume that with the decrease in validation checks, the probability of accuracy is more because with the decrease in validation checks the probability of wrong answers also decreases. So I decreased the validation checks from 10 to 06 which are default checks.

4.4.1. Training and Results:

Hidden Layers: 20

Epochs: 100

Validation Checks: 06

Goal: 0.01

Training Data	Testing Data	Iterations	Accuracy
10%	90%	30	96%
20%	80%	13	97%
30%	70%	17	98%
40%	60%	07	98%
50%	50%	23	98%
60%	40%	14	99%
70%	30%	32	99%

4.4.2. Analysis:

The number of iterations are peculiar as well as accuracy is uncertain because it again doesn't have a pattern (which shouldn't be) as his ancestor above. Some places the accuracy decreases, some cases remains same and in some cases increases which are fine because this kind of thing I experienced before in above hypothesis. Now let's see what kind of results come in next generation of hypothesis.

4.5. Hypothesis 4:

After doing above experiments based on different hypothesis, I will now train and test my neural network accuracy on the hypothesis that if I decrease the Goal parameter, it will increase the accuracy of my neural network because it will now try to reach the goal where error percentage is lower than before. I converted the goal from 0.01 to 0.001.

4.5.1. Training and Results:

Hidden Layers: 20

Epochs: 100

Validation Checks: 10

Goal: 0.001

Training Data	Testing Data	Iterations	Accuracy
10%	90%	32	97%
20%	80%	32	97%
30%	70%	57	97%
40%	60%	74	96%
50%	50%	13	98%
60%	40%	23	98%
70%	30%	25	99%

4.5.2. Analysis:

Again I got very mixed results and the accuracy remain uncertain on each test. I think this is because of the weights which are randomly generated each of the time. This problem could be tested if I save the weights after training and then again use them for next generation but I think this would shatter the very basic building block of neural network which is uncertainty, spontaneity and non-linearity.

4.6. Hypothesis 5:

The more I train neural network on more data, the better will be the accuracy. This means that with the increase in training data and decrease in testing data, the accuracy will be better.

4.6.1. Training and Results:

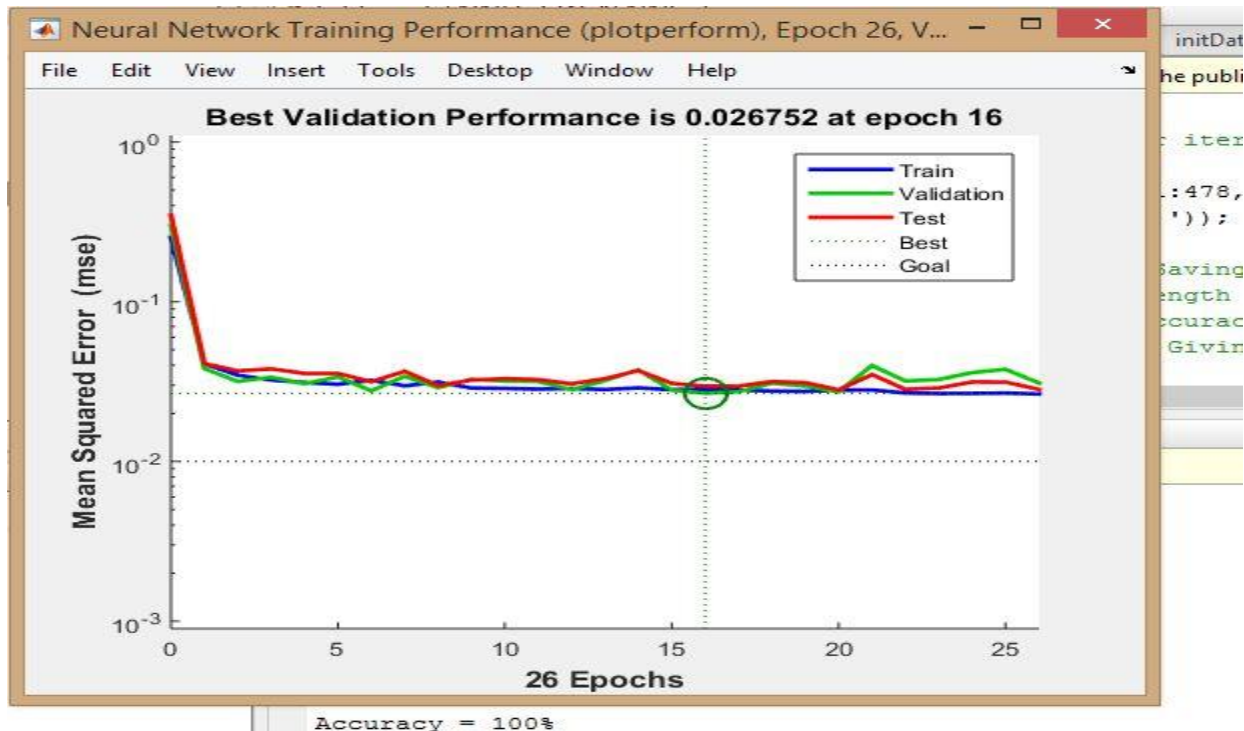
Hidden Layers: 20

Epochs: 100

Validation Checks: 10

Goal: 0.01

Training Data	Testing Data	Iterations	Accuracy
10%	90%	22	96%
20%	80%	75	97%
30%	70%	40	96%
40%	60%	43	97%
50%	50%	15	99%
60%	40%	35	97%
70%	30%	26	100%



4.6.2. Analysis:

This hypothesis **works**. As it could be seen that with the increase in training data and decrease in testing data, the accuracy on majority of the cases becomes better. The ideal case is 100 percent accuracy which on this hypothesis is achieved as shown in above results. The 100 percent accuracy is achieved when training data is 70% and testing data is 30%. The iterations remain a bit peculiar which I think are because of random weights generated which put affect on number of iterations.

5. Some Observations:

- When I increase the number of validation checks, it increases some number of iterations but it gives little bit more accuracy as compare to default validation checks which is 6.
- When I run the program for first time on validation checks = 10, it takes more iterations but in second run without clearing variables, it takes less iterations. And in third round a lot less iterations are taken and accuracy is also improved after second go.
- When training data is 70% and Testing Data is 30%, then 100% accuracy is achieved. This much accuracy is also achieved in other hypothesis but not in first row.
- On generic parameters and setting Training data to 70% and Testing Data to 30%, When I change the **Transfer function** from tansig to logsig, the Accuracy doesn't goes beyond to 22% from which I concluded that I should stick to tansig because it gives much more accuracy than logsig.

6. Conclusion:

This is not the first effort where neural networks are used to detect the breast cancer. Many systems were build and deployed and working efficiently. People still don't believe that the machines could be more accurate and especially when it comes to medical field.[6] Here I tried to produce more efficiency than before, so I changed different parameters and analyzed them on their results. Neural network best works on 5th hypothesis where training data is 70% and testing data is 30%. On this division of data, neural network gives ideal results which is 100 percent accuracy.

7. Bibliography:

- [1] D. Delen, G. Walker, and A. Kadam, "Predicting breast cancer survivability: A comparison of three data mining methods," *Artif. Intell. Med.*, vol. 34, no. 2, pp. 113–127, 2005.
- [2] Susan G. Komen Organization, "What is Breast Cancer?," 2014.
- [3] P. Tino, L. Benuskova, and A. Sperduti, "Artificial Neu," vol. 8, no. 3, pp. 455–472, 1997.
- [4] N. Yadav, A. Yadav, and M. Kumar, "An Introduction to Neural Network Methods for Differential Equations," pp. 13–16, 2015.
- [5] B. Murphy, A. Wakefield, and J. Friedman, "Best Practices for Verification, Validation, and Test in

Model- Based Design," *MathWorks, Inc.*, no. HCSS+, 2008.

[6] H. A. Abbass, "An evolutionary artificial neural networks approach for breast cancer diagnosis," *Artif. Intell. Med.*, vol. 25, no. 3, pp. 265–281, 2002.

[7]"breast cancer sign - Google Search", *Google.com.pk*, 2017. [Online]. Available: https://www.google.com.pk/search?biw=1366&bih=662&tbm=isch&sa=1&ei=NDMkWo3xH4r7vgThiJq4CA&q=breast+cancer+sign&oq=breast+cancer+sign&gs_l=psy-ab.3..0l10.86948.88966.0.89220.8.7.0.0.0.0.443.785.3-1j1.2.0....0...1c.1.64.psy-ab..6.2.784...0i67k1.0.hCVf9UBDtjg#imgsrc=e8y5yP0rgom8pM: [Accessed: 03- Dec- 2017].

[8]"neural network - Google Search", *Google.com.pk*, 2017. [Online]. Available: https://www.google.com.pk/search?q=neural+network&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjMmoTHrO7XAhWMO48KHfvQCsQQ_AUICigB&biw=1366&bih=662#imgsrc=EXoHkcJRnWRI3M: [Accessed: 03- Dec- 2017].