CLASIFICATION OF BREAST CANCER USING NEURAL NETWORK

Atiq uz Zaman
UOB:15026391
Submitted to: Mr. Junaid Akhter

Abstract

Artificial Neural Network is an information processing paragon which is inspired by the human's brain neuron. This document demonstrates the methodology of "Artificial Neural Network" to classify the breast cancer data. Results were acquired by using back propagation Neural Network algorithms and to get the high accuracy, different hypothesis were taken by changing the algorithms and their parameters. Finally, high accuracy (99% -100%) was reported by using training function "trainbr" with the different combination of training parameters.

Contents

1. Introduction

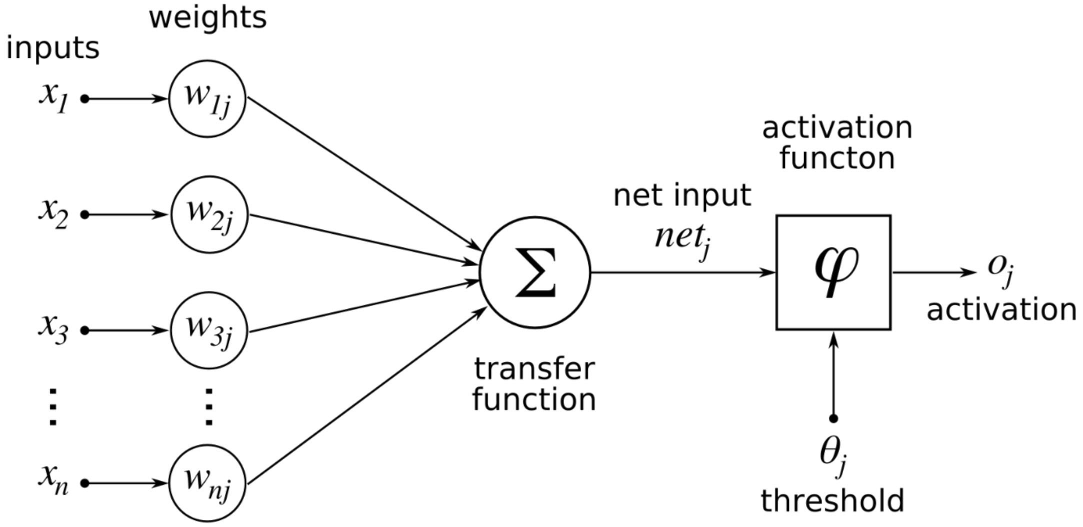
Classification is a technique and a process to classify the given data into something meaningful results. Neural Network uses a classification technique to solve many problems in different fields like medicine and engineering. In the medical field "Breast cancer is the most commonly occurring cancer in women and the second most common cancer overall. There are over 2 million new cases in 2018" [1]. This report provides a comprehensive explanation of classification of Breast Cancer using Neural Network technique. A dataset of Breast Cancer was taken from UCI Machine Learning dataset repository and tested and analyzed by NN. Breast Cancer has two type of data set, one is malignant (cancerous) and other is benign (not a cancerous).

2. Background

Breast cancer is the most dangerous cancer in women in developed and less-developed countries. In early days it was not recognized as cancer and it was just as a breast pain but later on it was detected by mammography. Now Scientists have classified breast cancer based on stage, gene expression, grade and type. The purpose of classification is to select the best treatment but many treatments are for a specific type and are very hostile which can cause verse effects. That's why breast cancer treatment algorithms are reliable and provide the best corroboration available [2].

Artificial Neural Network was constructed by the inspiration of the human brain neurons and it works like a biological neuron. Although it is straightforward in functionality but is very useful to solve our daily problems like classification, processing and much more.

Figure 1



[3]

Neural Network consists of inputs, hidden layers and outputs as shown in fig1. Inputs depend upon given data. Initially, weights can be set randomly. The impulse of a neuron is computed by the multiplication of weights and inputs. Transform function transforms this weighted sum to the activation function. Now according to the output, weights are reinitialized by the learning algorithms and training functions.

3. Main part

3.1 preprocessing

First of all, the dataset was taken from the UCI Machine Learning dataset repository. There were 11 columns and 699 rows. Sixteen instances were missing and there is '?' in those places, which were replaced by zeros manually. The first column represents the case ID (ignored because we didn't need this) and next 9 columns (input data) are symptoms data. The last column represents the output/desired values where 2 represents the benign and 4 represent the malignant.

3.2 Preprocessing for neural network

To train the data using feedforwardnet function, some preprocessing is required on given data, like all zeros were replaced with mean values of the 7th column. Data ware categorized into two ways, first input data and second output data. In output column two integers values were there, 2 represent the benign and 4 represent the malignant. Now problem was that, if we want good accuracy and performance then we have to get the equal number of input data and out data of both malignant and benign to pass the net function. To solve this issue, assign all benign values to a benign variable and all malignant values to a malignant variable. In the next step, we can get half rows from benign and half from malignant and combine them to make training input, similarly for training output we get corresponding values from the output column. Same procedure followed to get testing input and output. Now all our prerequisites are ready to train or test the breast cancer data.

3.3 System Architecture

A very common and efficient Artificial Neural Network called "Feed Forward Neural Network" is used for the classification of Breast cancer with the help of back propagation algorithms. Initially, a neural network is created by feedforwardnet and is stored in a variable called net. I used "traincgb" as a training function that updates weights and bias values according to the conjugate gradient backpropagation with Powell-Beale restarts. "traincgb" sets the network trainFun property. [4] Training occurs according to traincgb training parameters, shown below:

- Number of neurons are 45
- Epochs to train are 10000
- Show training window is true
- Maximum validation failures are 6
- Scale factor that determines sufficient reduction is 0.0001
- Scale factor that determines sufficiently large step size is 0.1
- Initial step size in interval location step is 0.01
- Parameter to avoid small reductions in performance is set to 0.1

Now data can be trained by passing arguments (net, input and output). Now by using sim function, we can simulate the trained net on testing data. After 30 iterations, mean accuracy was obtained which is 99.6%.

4. Experimental Results and analysis

All experiments were tested based on hypothesis, like changing the training functions and training parameters. Although by Matlab documentation, “traincgb” was the best training function for cancer data set but according to my hypothesis trainbr give the best accuracy and performance with the following parameters.

- net = feedforwardnet (13,'trainbr');
- epochs = 10
- lr = 0.00001
- trainParam.show = 25
- goal = 0.0
- max_fail=9
- net = setwb(net,rand(55));

4.1 Hypothesis#1 Effects of training functions

Hypothesis: It is difficult to decide which training function/training algorithm will be the best for a particular problem. It depends on many factors, including the complexity, training set, weights, biases in the network, error goal and whether the network is being used for pattern recognition or function approximation. “Trainbr” is the best training function with respect to accuracy and performance [6].

Based on the above hypothesis, experiments were performed not only using “trainbr” but also using other training functions and desired results are listed below:

Results

Training Function	Performance (%)
traincgb	98.9
trainbfg	95.1
*train*lm	98.2
traincgf	95.1
trainoss	94.3
traingdx	96.5
trainbr	99.6-100
trainrp	97.1
trainscg	97.6

Table 1: Effects of training functions

Analysis: Trainbr (Bayesian regularization backpropagation) updates the weights and biases according to Powell optimization and it minimizes the combination of square errors and weights while other training functions are not able to find the best combination. That's why it performs best than others.

4.2 Hypothesis#2 Effects of Number of Neurons in Hidden layer

Hypothesis: Increasing the number of neurons in hidden layer, increases the accuracy and vice versa.

Results:

No's of neurons on hidden layer	Performance	Accuracy (%)
1	1.4	95.13
3	2.04	96.21
5	3.1	98.81
10	2.1	98.92
12	2.2	99.1
15	1.20	95
20	2.6	96.00
30	0.5	93.3

Table 2 Effects of Number of Neurons in Hidden layer

Analysis:

Above results show that, as we increase the number of neurons, performance and accuracy increases accordingly but after some points when our neurons are more than 15 both performance and accuracy decreases. So Accuracy increases to 12 neurons then decrease.

4.3 Hypothesis#3 Effect of data distribution among training and testing

Hypothesis: Increasing the training data increases the accuracy. By increasing the training data, accuracy increases but after 80% training and 20% testing data accuracy decreases due to the unequal distribution of data.

Results:

Sr.	Training data (%)	Testing data (%)	Accuracy (%)
1	10	90	95.82
2	20	80	95.35
3	30	70	96.18
4	40	60	97.31
5	50	50	98.44
6	60	40	98.52
7	70	30	99.04
8	80	20	95
9	90	10	98.5

Analysis:

If we increase the training data and decrease the testing data, then accuracy increases until training data reaches to 70 percent. When our training data is 80% and above then our accuracy decreases and this result shows that 70/30 is the best combination to train the neural network because our feedforward neural network gets more possible cases. So large data of training improve the accuracy of the network. As our weights are randomly initialized for individuals, so sometimes it does not give the expected result but after taking 10-20 iteration and taking mean, it will give the pattern of expected result which is illustrated in the table.

4.4 Hypothesis#4 Effect of learning rate

Hypothesis: Learning rate is very important in neural network training because it can increase or decrease the performance and accuracy of a neural network. To get a better performance and accuracy learning rate should be small. Leaning rate also increases the training speed which is also important from some perspective.

Results:

Learning Rate	Performance	Accuracy (%)
0.0001	2.19	93.6
0.001	1.02	95.5
0.01	1.82	94.02
1	1.45	94.6
2	0.67	95.7
5	2.15	95.6
10	1.56	95.01

Analysis:

From the above table, results show that lesser the learning rate higher the performance and accuracy. The reason is that if the learning rate is high, weights proceed too far in right direction, which makes the network less accurate and accuracy will suppress.

5. Conclusion

It is concluded that various factors can affect the accuracy and performance like data distribution, number of neurons, learning rate and training functions. Hypothesis shows that data distribution is the key factor and its best ration is 70/30 which give the 99.6% best accuracy and performance. Similarly, learning rate is also very important which not only affect accuracy but also affect the training speed too. However, classification performance and accuracy of "traincgb" was also better but "trainbr" with one hidden layer, 13 neurons and learning rate of 0.0001 can be used to accomplish the best accuracy.

6. Bibliography

[1] F. J. S. I. S. R. Bray F, "Breast Cancer Statistics," 2018. [Online]. Available: https://www.wcrf.org/dietandcancer/cancer-trends/breast-cancer-statistics.

[2] W. Foundation, "Breast cancer classification," Wikipedia, 11 11 2018. [Online]. Available: https://en.wikipedia.org/wiki/Breast_cancer_classification. [Accessed 1 12 2018].

[3] A. Castrounis, "Artificial Intelligence, Deep Learning, and Neural Networks, Explained," KDnuggets, 10 2016. [Online]. Available: https://www.kdnuggets.com/2016/10/artificial-intelligence-deep-learning-neural-networks-explained.html. [Accessed 9 12 2018].

[4] T. M. inc, "traincgb," MathWorks, 2018. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/traincgb.html?searchHighlight=net.trainparam&s_tid=doc_srchtitle. [Accessed 1 12 2018].

[5] W.-P. C. Der-Ming Liou, "Applying Data Mining for the Analysis of Breast Cancer Data," LnkSpinger, 5 11 2014. [Online]. Available: https://link.springer.com/protocol/10.1007/978-1-4939-1985-7_12. [Accessed 1 12 2018].

[6] Abraham, measuring system design, 2005.

[7] M. inc, "Choose a Multilayer Neural Network Training Function," MathWorks, 2018. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/choose-a-multilayer-neural-network-training-function.html?searchHighlight=which%20training%20function%20best&s_tid=doc_srchtitle. [Accessed 1 12 2018].

[8] A. Abraham, Artificial neural networks, 2005.