BREAST CANCER CLASSIFICATION

NEURAL NETWORK APPROACH

ABOUT THE DOCUMENT

This report holds the details about the Neural Network created for the classification of Breast Cancer. The data set for both training and testing is available at UCI Machine Learning dataset repository. This report discusses the problem statement, the method followed, the solution derived, the results gathered and the future work.

Faizan Ullah

Neural Network and Fuzzy System

Contents

Introduction

Cancer can simply define as unbeatable cell division. Our body code (genes) can get mutated by different factors or due to the activation of already inherited mutated genes which in result can cause cancerous cell division. This clumping of mutated cells mold into an accumulated mass that can develop into a tumor.

Breast Cancer happens when the malignant(cancer) cell accumulate in the tissues of the breast. It is a most common cause of death in women. According to National Breast Cancer Foundation

> *On average, every 2 minutes a woman is diagnosed with breast cancer and 1 woman will die of breast cancer every 13 minutes.* ("Breast Cancer Facts - National Breast Cancer Foundation," n.d.)

The severity of this condition had lead people to work on this issue from the early ages. Its early diagnoses can save life because it is curable in early stage (Pan et al., 2017) . Its cure is available, but before curing it should get diagnosed first.

With the advancement of science and data analysis techniques scientist are keeping an eye on the symptoms of their earlier patients in order to diagnose this fatal disease. They store the record of previous patients and formulate a technique to extract useful information without using any lethal medical procedures. This is where artificial neural network comes in the story. Neural Network can be used for any kind of input to output mapping. Classification techniques of neural network collects the weights of different factors and contrive a solution on the supervised learning that has the ability to classify any unknown data given to it in future.

Background

Classification

Classification is the process of categorizing things in classes based on certain attributes. Classification methods require facts upon which it devise a mechanism to classify new data where is belongs. World Health Organization (WHO) classifies breast cancer in to two types, benign (mostly harmless) tumor and malignant (cancerous). A tumor is benign when it is not spreading to other body parts where as a tumor is malignant when it is affecting the other body parts.

We need to train our neural network based on this classification, so it will learn according to the data, what are benign and what are malignant and how to classify both based on any other data.

Neural Network

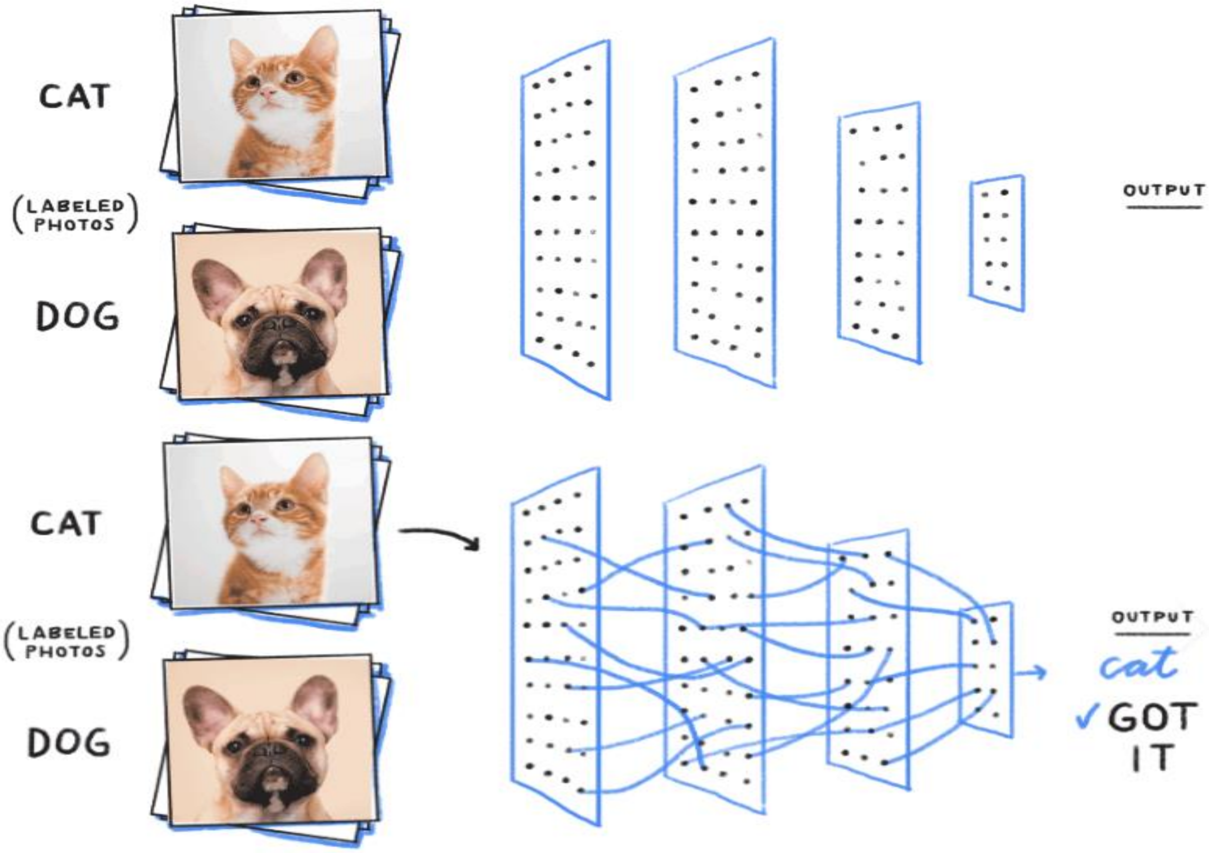
Neural Network is technique of devising pattern with the help of some organized data. Neural networks are considered quite good at this. Considering the inner architecture, we can find two types of NN's.

- Recurrent Neural Network: Where the neurons in the layers are joined in a circular manner.
- Feedforward Neural Network: Where the layers are connected in a liner way.

we can think of it as a composition function where we give different inputs and it gives you some output. Artificial Neural Network consider the architecture of human brain where neurons get input from the previous neuron and after definite process they categorically snap an output that is the input of other neurons and so on until the output layer. Under the hood the structure of neural network is based on the following parts.

- Neurons
- Weights
- Biases
- Activation Function
- Layers

These units are what we need to construct a basic neural network. With the manipulation of these we can increase or decrease the accuracy and complexity of our problem.



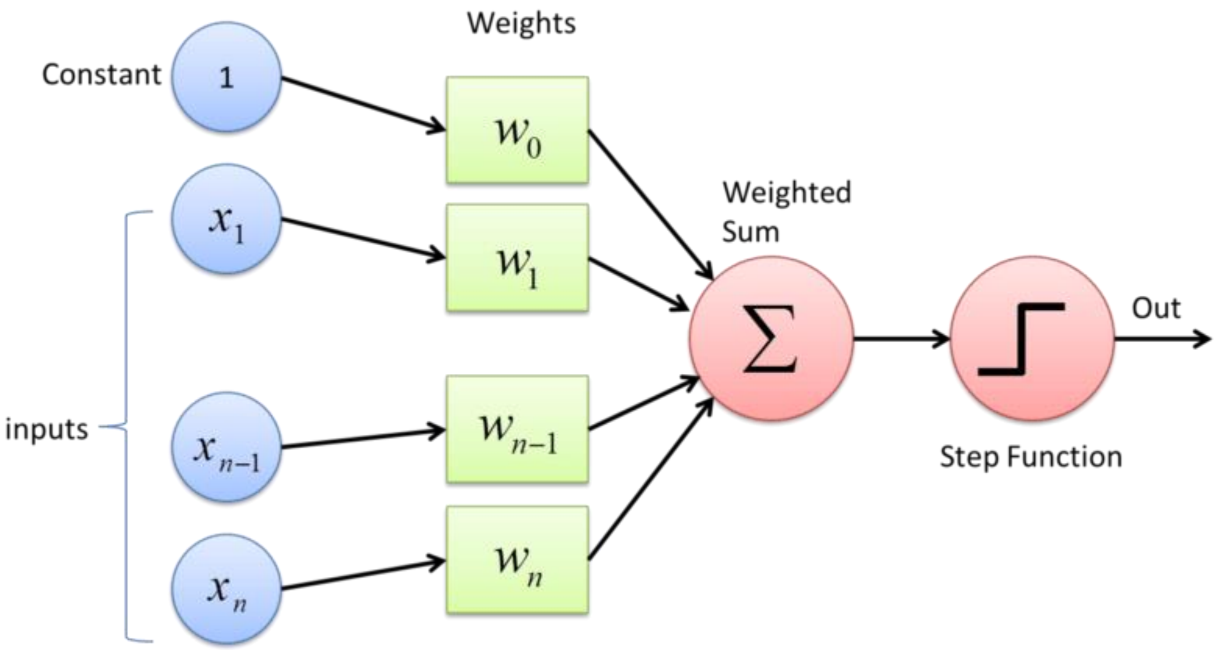
(SAGAR, 2017)

Neurons

The Basic building blocks use to accumulate the inputs with biases to apply the activation function because we need to restrict the output to a certain level. The properties of this unit are related to a function of line.

y = mx + c

Where m is the gradient, and c is y intercept. That's is the reason why it is called as liner Binary classifier, because it divides the data in to two parts.



(Suryansh,n.d.)

A neuron is a single layer neural network and a multi-layer neuron is a called as Neural Networks. Now Consider abundant of these. These multiple outputs are used as inputs to other units and then output is formulated after the final layer of neurons

Weights

There most important piece of the puzzle which is what our neural network is trying to learn along with the biases, so it generalizes the given problem.

Biases/Constant

The biases are considered as the neural network confidence on the amount of value it formulated to add on the set of waits.

Activation Function/Step Function

The function in every neuron that restrict the output in certain range depending on the type of activation function used. The encourage non-linearity in the NN and drastically effect the accuracy.

Types of activation functions are

- Sigmoid
- Tanh
- ReLU
- Leaky ReLU

Layers

Complexity of neural network significantly depends upon the layers it has. Non-Linearity of the output and the number of layers are proportional to each other. Increasing the layers also increase the non-linearity in the output. But unnecessarily increasing the layers for a simple task can drastically increase its complexity and decrease its accuracy.

Leaning of a Neural Network

Neural Network generalize to a problem using an algorithm which mostly is gradient descent. It requires a cost function which we minimize to make NN learn and do minimum mistakes. Along with it we have back prorogation algorithm which propagate the error from the output layer to backwards using the chain rule from calculus.

Applications

Neural networks beside solving classification problems use in identifying patterns, predicting the solution and modeling.

Main Part

To make a Neural Network Classifier we are going to need a tool named as MATLAB. It has a neural network tool box that will help us making neural network of our choice with attributes as needed.

Along with this tool we are going to use Wisconsin Breast Cancer Database (January 8, 1991) to perform supervised learning of our neural network.

About the Data Set

This 'Wisconsin Breast Cancer Database (January 8, 1991)' is obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. Values from 2 to 10 is the attributes used to classify between two possible class: benign and Malignant.

Number of Instances: 699 (as of 15 July 1992)

Number of Attributes: 9 plus the class attribute and id number.

#	Attribute	Domain
1	Sample code number	id number
2	Clump Thickness	1--10
3	Uniformity of Cell Size	1 – 10
4	Uniformity of Cell Shape	1 – 10
5	Marginal Adhesion	1 – 10
6	Single Epithelial Cell Size	1 – 10
7	Bare Nuclei	1 – 10
8	Bland Chromatin	1 - 10
9	Normal Nucleoli	1 - 10
10	Mitoses	1 - 10
11	Class	(2 for benign, 4 for malignant)

Missing attribute values: 16

There are 16 instances in Groups 1 to 6 that have a single missing (i.e., unavailable) attribute value, now denoted by "?".

Class distribution:

- Benign: 458 (65.5%)
- Malignant: 241 (34.5%)

Pre-Processing

To make this data useful we need to apply certain refinement on it. This will make the data suitable for the NEURAL NETWORK as an input.

Step-1 Sorting of the data.

Column 11 has the result showing whether the patient is benign or malignant. Soring with respect to this column will separate these two classes. Splitting the data helps us applying further steps of preprocessing.

Step-2 Adjustment of the NULL value.

According to the data we are lacking in total of 16 values, so to train the NN on this data we need to add a value here. Our approach relies on taking mean of the whole column that has the same class as this missing block i.e. benign or malignant. As the data is sorted so this is not going to be a problem.
Column of benign class is formulating 2 as the mean value and column of benign class is formulating 7. This means that both classes will replace their null value with these results.

Step-3 Formation of training set and validation set

As we have 699 instances in the data and we want to train our neural network to classify new instances. But to test our claimed classifier we must have some data upon which we can test. As we don't have any, so we will separate this data in 70% training set and 30% validation set.

Step-4 Creating Input Data Set

We'll take the data from both classes (benign and malignant) in the proportion of the given testing percentage. So, a new input matrix then gets populated with this.

Step-5 Output Value Mapping Set

Making an output representation more natural to neural network activation function. As benign class is represented by 2 and Malignant by 4. But the sigmoid output is in between 0 and 1 that's why we will develop a new set of matrix that have two columns to map results in binary. Two is represented by (1,0) and four is represented by (0,1). As Final output is also in two columns, we will take max of both and substitute this value with 1 and other value with 0.

The system architecture

Classification of data is achieved by setting up a multi-layer feedforward neural network with back-prorogation algorithm, having one hidden layer having ten neurons which are subjected to change.

Training

Preprocessing is followed by training where we apply the following steps

Feedforward neural network is made by inserting number of neurons and layers as one parameter and training algorithm as other parameter. This will develop a neural network.

Setting up training parameters of the neural network that is just created using ‘trainparam’ method of matlab. We are also setting custom waits because we want to make the neural network take same steps for the same data.

To train it we’ll pass this neural network object to MATLAB’S train method along with input data and output values of these instances because this is a supervised learning area.

Testing

Testing is done by matlab’s sim method that simulate the validation data on the trained net.

Accuracy

We’ll collect the result and compare it with the expected ones. The count of the matched result is stored.

Accuracy is calculated by the following formula.

(Count/total number of validation data)*100

EXPERIMENTAL RESULTS AND ANALYSIS

Neural network is prepared, and it is running just fine. Now we will test it to get the best of it. We will change multiple attributes to understand its functionality better. Let’s start it with random initial set variable.

Experiment Zero -Initial Run.

Training percentage	Training algorithm	Max Epochs	Learning rate	Performance Goal	Hidden layer and Neuron
70%	Traingd	10000	0.2	0.02	1 with 10

Results

Accuracy (%)	Time taken (sec)	Max Epochs Iterate	Perf. Goal Achieved	Remarks
99.0431	20	10000	0.0291	Goal did not get achieved, but accuracy is all we care about and its satisfying.

Experiment 1

Hypothesis

“Number of epochs and learning rate are disproportional. If we increase the number of epochs from 10000 to 100000 we will decrease the learning rate from 0.2 to 0.02. And there will be minimal effect in the accuracy as minimal learning rate means minimal deviation towards error which require multiple iterations on the data to find the true point.

Configurations

Training percentage	Training algorithm	Max Epochs	Learning rate	Performance Goal	Hidden layer and Neuron
70%	Traingd	100000	0.02	0.02	1 with 10

Results

Accuracy (%)	Time taken (sec)	Max Epochs Iterate	Best training Perf. Achieved	Remarks
99.0431	20	100000	0.029106	Accuracy stays the same which confirm my hypothesis.

Let’s try another scenario to confirm this hypothesis. I will proportionally decrease both the learning rate and the epochs(iteration) and I should see the decrease in efficiency.

Configuration

Training percentage	Training algorithm	Max Epochs	Learning rate	Performance Goal	Hidden layer and Neuron
70%	Traingd	1000	0.002	0.02	1 with 10

Results

Accuracy (%)	Time taken (sec)	Max Epochs Iterate	Best training Perf. Achieved	Remarks
94.7368	1	1000	0.0982	Accuracy deceases with decreasing of learning rate proportional to the epochs

Result Analysis

If the learning rate is small, then the epochs must be substantial enough to decrease the error arising. The leaning rate helps in deciding the shift the curve might need to take, with small amount of learning rate the shift will be minimum hence more epochs are required to find the optimal place. Whereas when learning rate is huge then the shift will be enormous and will arise more problem then reducing it. Above results are seconding this explanation.

Experiment 2

Hypothesis

“Your choice of an activation function can increase or decrease the performance of your Neural Network”

Activation function

Sigmoid, tanh, log, RELU etc.

Configurations

Training percentage	Training algorithm	Max Epochs	Learning rate	Performance Goal	Hidden layer and Neuron with activation function
70%	Traingd	10000	0.02	0.02	1 and 10 with “x”

Resultant Accuracy when tansig is used

Accuracy (%)	Time taken (sec)	Max Epochs Iterate	Best training Perf. Achieved	Remarks
99.0431	20	10000	0.0291	It’s the default function giving us maximum accuracy.

Resultant Accuracy when logsig used.

Accuracy (%)	Time taken (sec)	Gradient Achieved	Max Epochs Iterate	Best training Perf. Achieved	Remarks
98.5646	19	0.00508	10000	0.0376	Logsig decreased the efficiency a little but it was somewhat quicker.

Results when poslin (RELU) used.

Accuracy (%)	Time taken (sec)	Gradient Achieved	Max Epochs Iterate	Best training Perf. Achieved	Remarks
98.0861	9	$1.00e^{-5}$ (Max)	4980	0.0400	ReLU activation function was most quick with Maximum gradient under minimum epochs.

Result Analysis

Accuracy and efficiency is changing with the change of activation function as we can see in the above results. Activation function is use to limit the output to a certain range. Choice of activation function effect the non-linearity which in result increase or decrease the efficiency.

Experiment 3

Hypothesis

“Having too many layers for a simple task can unnecessarily increase its complexity and in most cases decrease its accuracy and vice versa”

Configurations

Training percentage	Training algorithm	Max Epochs	least gradient Possible	Learning rate	Performance Goal	Hidden layer and Neuron
70%	Traingd	10000	9.98e^{-06}	0.2	0.02	3 layers with 1000, 100 and 10 neurons respectively.

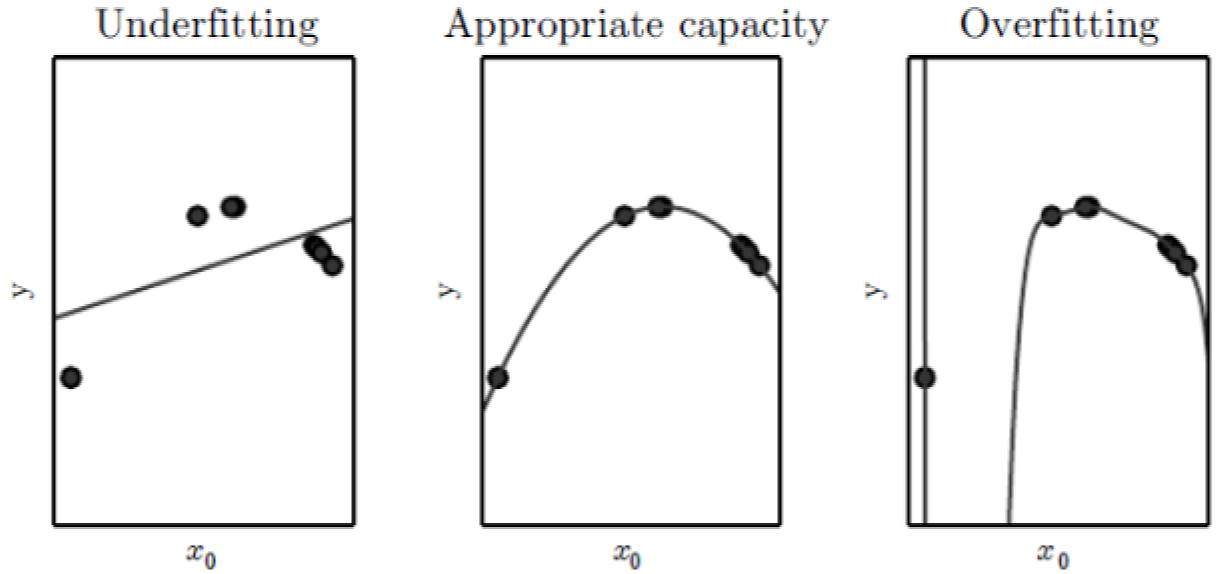
Results

Accuracy (%)	Time taken (min:sec)	Minimum Gradient achieved	Max Epochs Iterate	Best training Perf. Achieved	Remarks
94.736842	1:41	9.98e^{-06}	100000	0.100	Accuracy drops, which second my hypothesis of increasing complexity makes NN overfit on the problem

1.

Result Analysis

Increasing the number of layers, if the problem is not that complex, will surely decrease the amount of accuracy on testing data. This is known as overfitting on the data set. Overfitting means that neural network is only able to generalize on training data set and it won’t able to figure out a solution for any unseen data. This can pictorially explain from the following image.



(Ian, Yoshua, & Aaron, 2016)

This means that data is only trained for the given data and not able to generalize a solution for unseen one.

Bibliography

Breast Cancer Facts - National Breast Cancer Foundation. (n.d.). Retrieved December 3, 2018, from https://www.nationalbreastcancer.org/breast-cancer-facts

Ian, G., Yoshua, B., & Aaron, C. (2016). *Deep Learning*. MIT Press. Retrieved from http://www.deeplearningbook.org/

Pan, H., Gray, R., Braybrooke, J., Davies, C., Taylor, C., McGale, P., … Hayes, D. F. (2017). 20-Year Risks of Breast-Cancer Recurrence after Stopping Endocrine Therapy at 5 Years. *New England Journal of Medicine*, *377*(19), 1836–1846. https://doi.org/10.1056/NEJMoa1701830

SAGAR, S. (2017). What the Hell is Perceptron? – Towards Data Science. Retrieved December 4, 2018, from https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53

Suryansh, S. (n.d.). Neural Networks: All YOU Need to Know – Towards Data Science. Retrieved December 4, 2018, from https://towardsdatascience.com/nns-aynk-c34efe37f15a