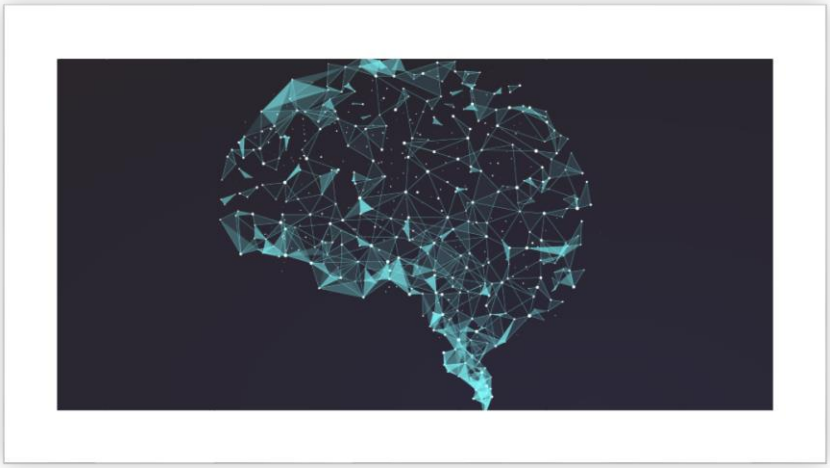

Breast Cancer Classification Using Neural Network Approach

Hamza Rehan | 15026430 | November 28, 2018

Table of Contents

Introduction

Breast cancer is the most common type of cancer in women and its causes are yet to be found by the doctors. Breast cancer occurs when cells in breast grow out of control and forms into a group of cells called tumor. Breast cancer can grow into other part of body through blood vessels which is called as metastasis. [1]

Medical centers all over the world maintain the database of their patients, causes of their diseases and medication methods to conduct a research. Researchers are now using the power of Ai in the field of healthcare to solve complex problems. Until now computer machines have made accurate predictions and classifications of different diseases. This powerful field of Computer Science have made it possible for the dumb machines to detect, classify and predict different diseases.

We have to develop a neural network for the classification of breast cancer data which is provided by Dr. William H. Wolberg from University of Wisconsin Hospitals Madison, Wisconsin, USA.

Background

Breast cancer classification

Breast classification is a great challenge for researchers and scientists. Machine learning techniques have revolutionized the process of diagnoses of the breast cancer [].Here we are identifying two classes of cancer, Malignant and benign. Malignant is cancerous and vice versa. A tumor is malignant (cancerous) if it has the ability to spread to the other part of body and start infecting other body tissues. Benign (non-cancerous) tumor doesn't have the ability to infect other tissues. [1]

Neural network models

Neural networks models are inspired by the neuron networks in human body. These models are algorithms who learn by detecting the patterns and relationships in datasets.

We can view any neural network as set of layers with each layer consisting of different numbers of neurons and each neuron performing a particular task. Layers are defined as follows:

1. Input layer: neurons on Input layer takes initial data into the system and multiply them with weights. [6]
2. Hidden layer: Hidden layer is between input and output layer which takes the weighted input and gives an output using the activation function. [5]
3. Output layer: The neurons on output layer gives the output after processing the data in hidden layers.

Neural network can have a single layer and multi layers. A single layer neural network is known as perceptron.

A general diagram of single and multi-layer neural networks are as follows.

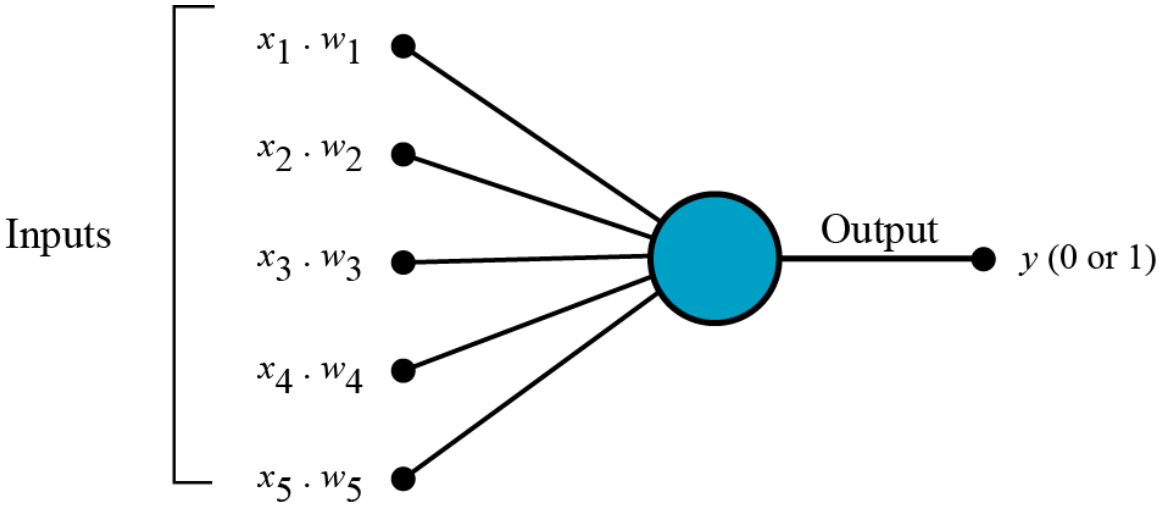

Figure1: A single layer neural network. [7]

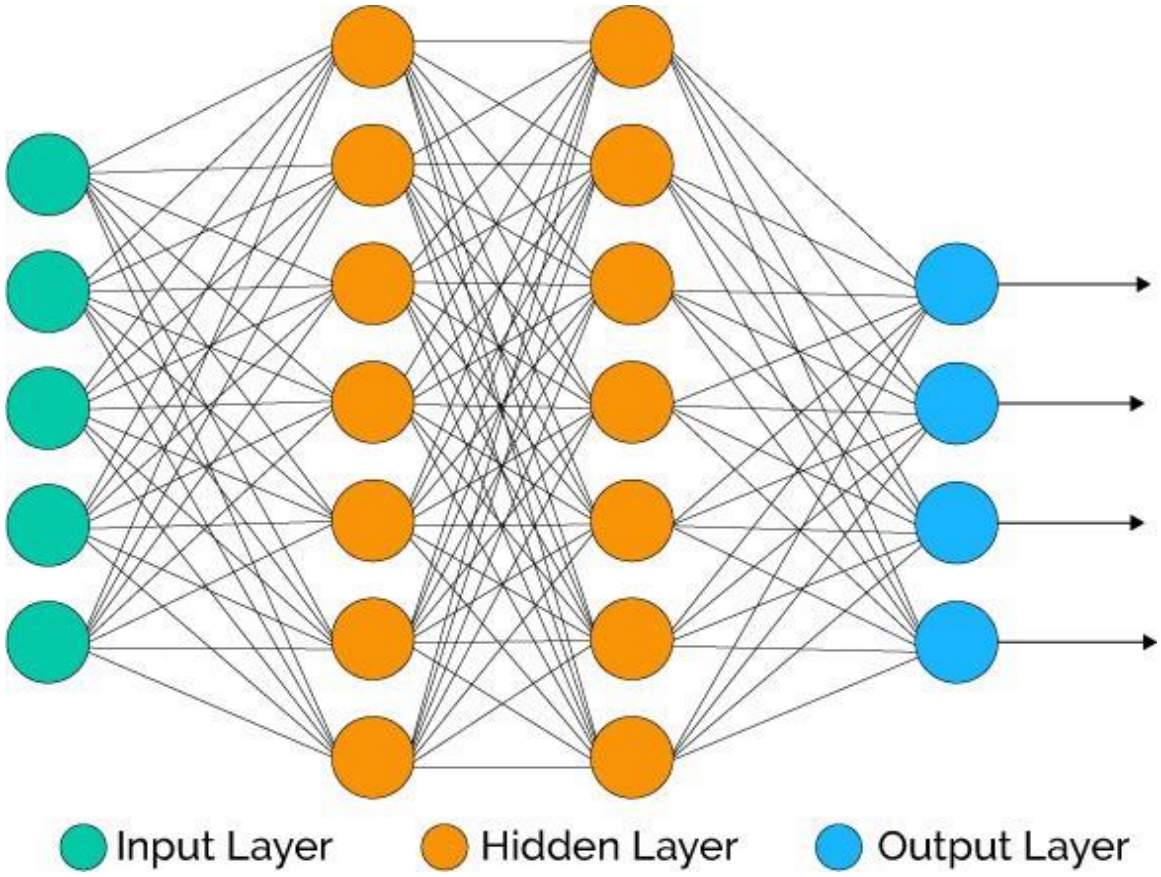

Figure2: A multi-layer neural network. [7]

Single layer neural networks is the simplest type of neural network. Single layer neural network can also be considered as a part of a multi-layer neural network where information travels only in one direction. Single layer neural network can perform a binary classification. [8]

Multi-layer neural networks are more complex and contains hidden layers. The system contains different hidden layers linked with each other using activation function which activates/fires neurons. [8]

Neural networks can be used for classification, pattern matching, prediction and modelling.

Main part:

Preprocessing steps on provided data:

Before we start developing a neural network we must ready few things which includes,

1. Wisconsin Breast Cancer Data by Dr. Williams H. Walberg (physician) University of Wisconsin Hospitals.[9]
2. MATLAB R2015a
   We need to preprocess the data in order to use it for proper training of our neural network. Following steps are
3. The provided data consists of a total 699 rows from which 16 values are missing and they all were in column no 7. I replaced the missing values with '?' and then replaced those '?' with the mean of column 7 which is 3 because csv read command doesn't read '?'.
4. In our dataset first column indicates id value, other 9 column's contains the actual data and in last column we have actual output so in our training input data we will exclude column 1 and 11.

Experimental Results and Analysis:

	Run Configurations				
	Training Data Size	Epochs	Hidden layers	Learning rate	Initial weights
	100	1000	2	0.7	1

0	**Hypothesis**
	N/A
	Accuracy
	93.3222
	Conclusion
	This is test experiment to make base of our experiments

1	**Run Configurations**				
	Training Data Size	Epochs	Hidden layers	Learning rate	Initial weights
	500	1000	2	0.7	1
	Hypothesis				
	In previous test experiment we have small amount of training data. With increases amount of training data it's expected that the accuracy will improve except only in one condition if the additional data is noisy. Now I'm training my model on 71.5% data and testing on the remaining 29.5% data.				
	Accuracy				
	98.4925				
	Conclusion				

	The hypothesis results to be true and accuracy has improved a lot. This is because now our model has been trained on a good amount of data.

2	**Run Configurations**					
	Training Data Size	Epochs	Hidden layers	Learning rate	Initial weights	Accuracy
	500	100	2	0.1	1	96.9849
	500	100	2	0.3	1	97.4874
	500	100	2	0.7	1	98.9950
	500	100	2	0.8	1	22.1106
	Hypothesis					
	In this hypothesis it's expected that by setting the learning rate close to 0 and 1 can significantly decrease the accuracy.					
	Conclusion					
	The hypothesis results to be true as we can see clearly in the above data that learning rate close to 0 and 1 decrease the accuracy because learning rate factor close to 0 means that it makes the weight shift slowly and closer to 1 means it makes the weight shift too fast so we have to set the learning rate between 0 and 1.					

	Run Configurations

3	Training Data Size	Epochs	Hidden layers	Learning rate	Initial weights	Accuracy
	500	100	2	0.7	1	98.9950
	500	200	2	0.3	1	98.4925
	500	700	2	0.1	1	98.4925
	Hypothesis					
	In this hypothesis it's expected that if you lower the learning rate then by increasing the epochs you can maintain the accuracy.					
	Conclusion					
	We can say that even if we lower the learning rate and take it close to 0 we can still maintain the accuracy by increasing the value of epochs. Why is that so? We can see in the next hypothesis.					

4	**Run Configurations**					
	Training Data Size	Epochs	Hidden layers	Learning rate	Initial weights	Accuracy
	500	5	2	0.7	1	97.4874
	500	100	2	0.3	1	98.9950
	500	500	2	0.1	1	99.4975
	Hypothesis					

	In this hypothesis it's expected that if you increase the epoch value then the accuracy should improve.
	Conclusion
	Our hypothesis results to be true as the accuracy has been improved notably by increasing the epoch value because the more no of time you will pass your data in a system the more accurately it will check. The only disadvantage is it will take more time.

Bibliography and citations:

1. https://www.appliedradiology.com/articles/a-promising-future-for-ai-in-breast-cancer-screening
2. https://www.maurerfoundation.org/a-brief-history-of-breast-cancer/
3. https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9
4. https://www.techopedia.com/definition/33263/output-layer-neural-networks
5. https://www.techopedia.com/definition/33264/hidden-layer-neural-networks
6. https://www.techopedia.com/definition/33262/input-layer-neural-networks
7. https://towardsdatascience.com/machine-learning-fundamentals-ii-neural-networks-f1e7b2cb3eef
8. https://www.techopedia.com/definition/33268/multi-layer-neural-network
9. http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29