**Sravani Konujula**

**16230172**

**ID :- 23**

**Deep Learning**

**LAB Assignment-1**

Introduction

Logistic Regression which is used to be performed on the variables which are binary (0 and 1). It is a predictive analysis.

The main difference between logistic and linear is

Logistic is binary classification where the result will be projected inform of discreet valued output.

Linear regression is defined as continuous model

Objectives

The main of this regression is for finding the absolute model to portray relation between binary dependent variable and group of ratio-level independent variables.

For instance, for a given x if f(x)>threshold classify it to be 1 else classify it to be 0

Let’s consider a set with brain tumor as training data and use it as the input and to figure out the result either be a benine or malignant tumor. Hence the result will be 1 or 0.

* Implementing LR in tensor flow
* Creating graph
* Comparing results

Approaches/Methods

Sigmoid Function in the logistic regression is defined as mapping between inputs to predicated outputs.



X is independent variables

Y\_Predicted is target

Workflow

The procedure is

* Choosing Frequency and Summary statistics
* Considering In-variate points
* Loading the wanted dataset.
* Split the data as training and testing data
* Initialize both data and target for optimal results
* Begin the plot with obtained results
* In addition to this model will be estimated.
* In the next step evolution is required.
* It is followed by graph plotting.
* Finally, It is concluded as tabulated results

Datasets

Data set taken to proceed with logistic regression is “load\_breast\_cancer()”

It is with 569 rows and 30 columns which are to be loaded for the further procedure .

Configuration

Pycharm

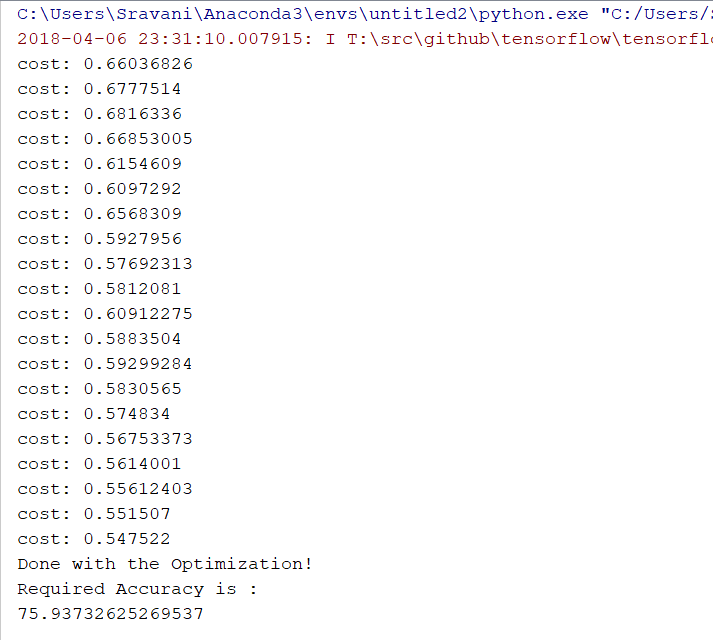
Python: 2.7.13

Parameters

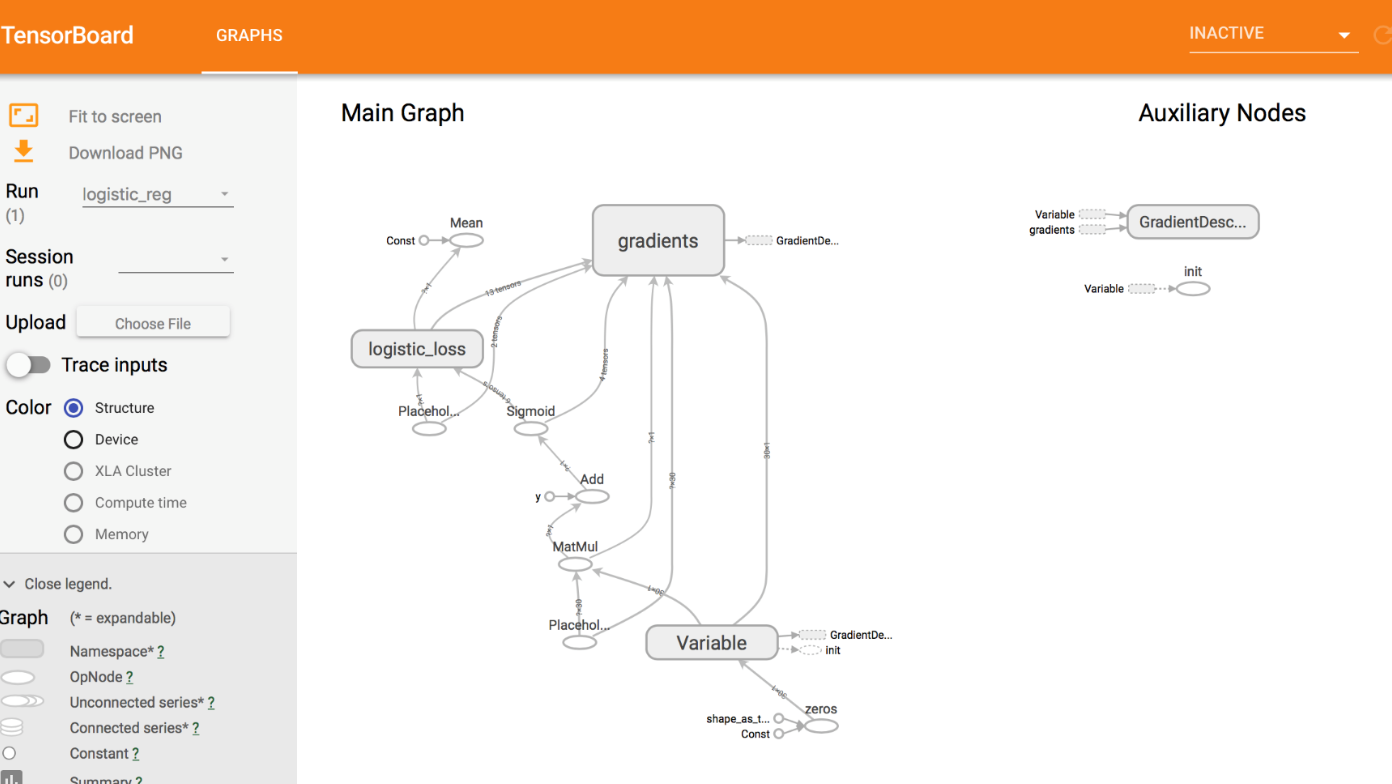
Learn Rate = 0.001  
Training count is = 200

Evaluation & Discussion

* Code snippet
* **import** tensorflow **as** tensef1  
  **import** numpy **as** np  
  **from** sklearn.datasets **import** load\_breast\_cancer  
  **from** Q1 **import** pred1  
  *# Considering a data set and loading it*dset1 = load\_breast\_cancer()  
  data = dset1.data  
  labels = dset1.target  
    
  *#taking the labels in array format  
  #569 rows*labels = np.array(labels).reshape(569,1)  
    
  *#Constructing a placeholderds for 30 columns*X1 = tensef1.placeholder(tensef1.float32, shape=[**None**, 30])  
  y1 = tensef1.placeholder(tensef1.float32, shape=[**None**, 1])  
  *# goes with random values.*tensef1.set\_random\_seed(1)  
  *#weights getting started*W3 = tensef1.Variable(tensef1.zeros([30, 1]))  
  b1 = 0  
  *#operational functions*Activity1 = tensef1.nn.sigmoid(tensef1.add(tensef1.matmul(X1, W3), b1))  
    
  *#It is for loss*loSS1 = tensef1.reduce\_mean(tensef1.nn.sigmoid\_cross\_entropy\_with\_logits(logits=Activity1, labels=y1))  
  *#optimixer for gradient descent*optimaL1 = tensef1.train.GradientDescentOptimizer(learning\_rate=0.0001).minimize(loSS1)  
  *# Declaring global variables*init = tensef1.global\_variables\_initializer()  
  *# Session method declaration***with** tensef1.Session() **as** sess:  
   sess.run(init)  
   writer = tensef1.summary.FileWriter(**"./graphs/logistic\_reg"**, sess.graph)  
  *#considering range for 200 values* **for** k **in** range(200):  
    
   \_, acc1 = sess.run([optimaL1, loSS1], feed\_dict={X1:data, y1:labels})  
    
   **if** k%10==0:  
   print(**"cost: "** + str(acc1))  
    
   writer.close()  
   parameters = sess.run(W3)  
  print(**"Done with the Optimization!"**)  
  outs = pred1(data, parameters) *#Calling the function  
  #Calculating accuracy score for the regression model*A=format(100 - np.mean(np.abs(outs - labels)) \* 100)  
  print(**"Required Accuracy is :"**)  
  print(A)
* Output



* Tensor Graph



If the learning rate is 0.0001 the accuracy is 75.93732625269537 and if it is learning rate is 0.001 the accuracy is 37.25834821372352

If range is 200 then accuracy is 75.93732625269537 and If 450 then accuracy is 37.25834821372352

Conclusion:

* As the Learning rate increases the Accuracy also gets decreases

So Learning rate and Accuracy are inversely proportional

* As the Range increases the Accuracy gets decreased

References:

https://www.statisticssolutions.com/what-is-logistic-regression/