**CS5590/490 Python-DeepLearning**

**Deep-Learning**

**LAB**

**Assignment-2**

**Submitted by: Shajee Uddin Zain Mohammed**

**Student ID: 16230179**

**Class ID: 28**

**Task:**

**1. Implement the Logistic regression with new data set which is not used in class.**

**2. Show the graph in TensorBoard.**

**3. Change the hyperparameter and compare the result.**

**Introduction:**

In this lab assignment we have implemented the logistic regression on the sample stock and calculated the models accuracy.

Logistic regression is a predictive analysis and is used to describe the relationship between one dependent binary variable and more, ordinal, and interval.

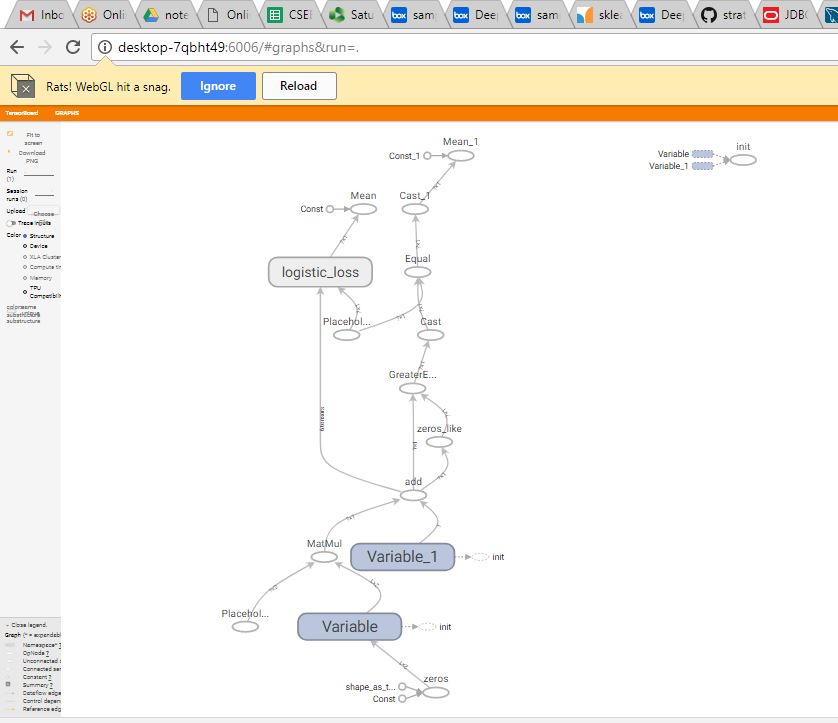
**Objectives:**

To implement the logistic regression and represent it with the tensorboard.

Approaches/Methods

We read the data from the file and divide it into the training and test data set using logistic regression and the output is shown in binary as 0 and 1.

**Workflow:**



**Datasets:**

Textbook contains the section, book and sheet datasets.

**Parameters:**

Learning rate

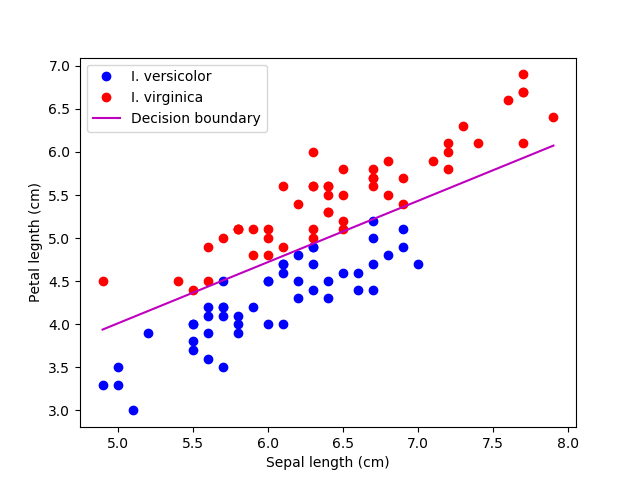
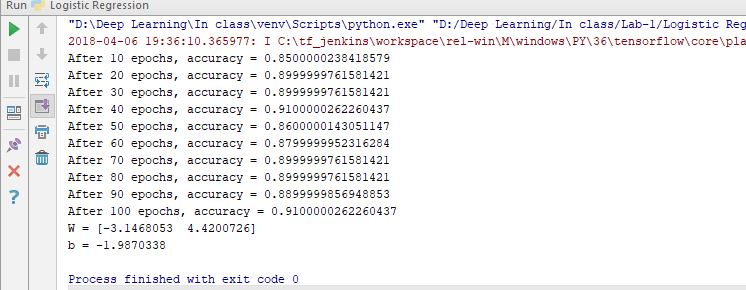
Epochs

Weight and bias

**Evaluation & Discussion**

**Code:**

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** tensorflow **as** tf  
  
*# Step 1: read in data from the file*text = np.loadtxt(**'Samplestck.txt'**, skiprows=1)  
book = text[:,:2]  
sheet = text[:,2:] *#Creating matrix  
# Step 2: create placeholders for input a and label z*a = tf.placeholder(tf.float32, [**None**, 2])  
z = tf.placeholder(tf.float32, [**None**, 1])  
  
*# Step 3: create weight and bias, initialized to 0*w = tf.Variable(tf.ones([2, 1]))  
b = tf.Variable([1.0])  
  
*# Step 4: build model to predict Y*logits = a \* w + b  
  
*# Step 5: use the square error as the loss function*loss = tf.nn.sigmoid\_cross\_entropy\_with\_logits(labels=z, logits=(a \* w + b))  
loss = tf.reduce\_mean(loss)  
  
*# Step 6:Calculating the Accuracy*predictoper = tf.greater\_equal(logits, tf.zeros\_like(logits))  
correctoper = tf.equal(tf.cast(predictoper, tf.float32), z)  
accuracyoper = tf.reduce\_mean(tf.cast(correctoper, tf.float32))  
**with** tf.Session() **as** sess:  
*# Step 7: initialize the necessary variables, in this case, w and b* sess.run(tf.global\_variables\_initializer())  
  
 writer = tf.summary.FileWriter(**'./graphs/log\_reg'**, sess.graph)  
learng\_rt = 0.01  
epochs= 100  
optimizer = tf.train.GradientDescentOptimizer(learng\_rt)  
train\_op = optimizer.minimize(loss)  
  
*#using session run*sess = tf.Session()  
sess.run(tf.global\_variables\_initializer())  
  
*# Using random number generator*np.random.seed(0)  
  
**for** epoch **in** range(epochs):  
 *# each point is once present in random order* index = np.random.permutation(text.shape[0])  
 **for** i **in** index:  
 feed\_dictionary = {a: book[i:i+1], z: sheet[i:i+1]}  
 sess.run(train\_op, feed\_dictionary)  
  
 **if** (epoch+1) % 10 == 0:  
 feed\_dictionary = {a: book, z: sheet}  
 accy = sess.run(accuracyoper, feed\_dictionary)  
 print(**"After {} epochs, accuracy = {}"**.format(epoch+1, accy))  
  
*# Print the result*Weigh\_val, bais\_val = sess.run([w, b])  
Weigh\_val = Weigh\_val[:,0]  
bais\_val = bais\_val[0]  
print(**"w ="**, Weigh\_val)  
print(**"b ="**, bais\_val)  
  
**def** predict(a\_):  
 **return** 1 \* sess.run(predictoper, {a: a\_})  
  
*#predictions Models*labels = predict(book)[:,0]  
  
*#indices for two species*index\_0, = np.where(labels == 0)  
index\_1, = np.where(labels == 1)  
  
*# Plotting*plt.plot(book[index\_0,0], book[index\_0,1], **'bo'**, label=**'I. versicolor'**)  
plt.plot(book[index\_1,0], book[index\_1,1], **'ro'**, label=**'I. virginica'**)  
  
*# Plotting the hyperplane*x\_sep = np.linspace(book[:,0].min(), book[:,0].max())  
y\_sep = (-bais\_val - Weigh\_val[0]\*x\_sep) / Weigh\_val[1]  
plt.plot(x\_sep, y\_sep, **'m'**, label=**"Decision boundary"**)  
  
plt.legend()  
  
plt.xlabel(**"Sepal length (cm)"**)  
plt.ylabel(**"Petal legnth (cm)"**)  
  
plt.show()



**Conclusion:**

When we change the hyper parameters the accuracy of the model is changing since the logistic regression is a predictive analysis, it is used to describe data and to explain the relationship between one dependent binary variable and one or more, interval or independent variables