Lab Assignment-1

Akshay Kumar Reddy G

Class Id: 11 (16231179)

Introduction:

In this assignment we implemented Logistic Regression on Iris dataset and computed the accuracy of this model.

Logistic regression is predictive analysis which is conducted when the dependent variable is binary so that we can categorize the data. This technique is used to describe the relationship between one dependent binary variable and one or more independent variables.

We initially load the iris dataset from a .txt file with the help of numpy, define the X_data and Y_data. And next to fit in the data into the model we will define x, y placeholders.

Next we initialize the model parameters Weight(W) and bias(b) with values as zero. Then we define our model, in this as it is a Logistic Regression we defined model as:

```
logits = tf.matmul(x, W) + b
```

We define the function to calculate the loss and accuracy of our model. To calculate the accuracy we take the values of predicted and output values and compute the accuracy values.

In this model we have taken the number epochs to be 100 and learning rate as 0.01 and then to optimize the model using Gradient Descent Optimizer.

We initialize the session and run it, we then define the function to minimize the loss for every epoch in the model and finally we predict the results of dependent variable.

```
def predict(x_):
                       return 1 * sess.run(predict_op, {x: x_})
#Model predict
                  labels = predict(X_data)[:,0]
```

Finally we collect the data, results and label them on axes and plot the result of our model here we get our data plotted in two categories.

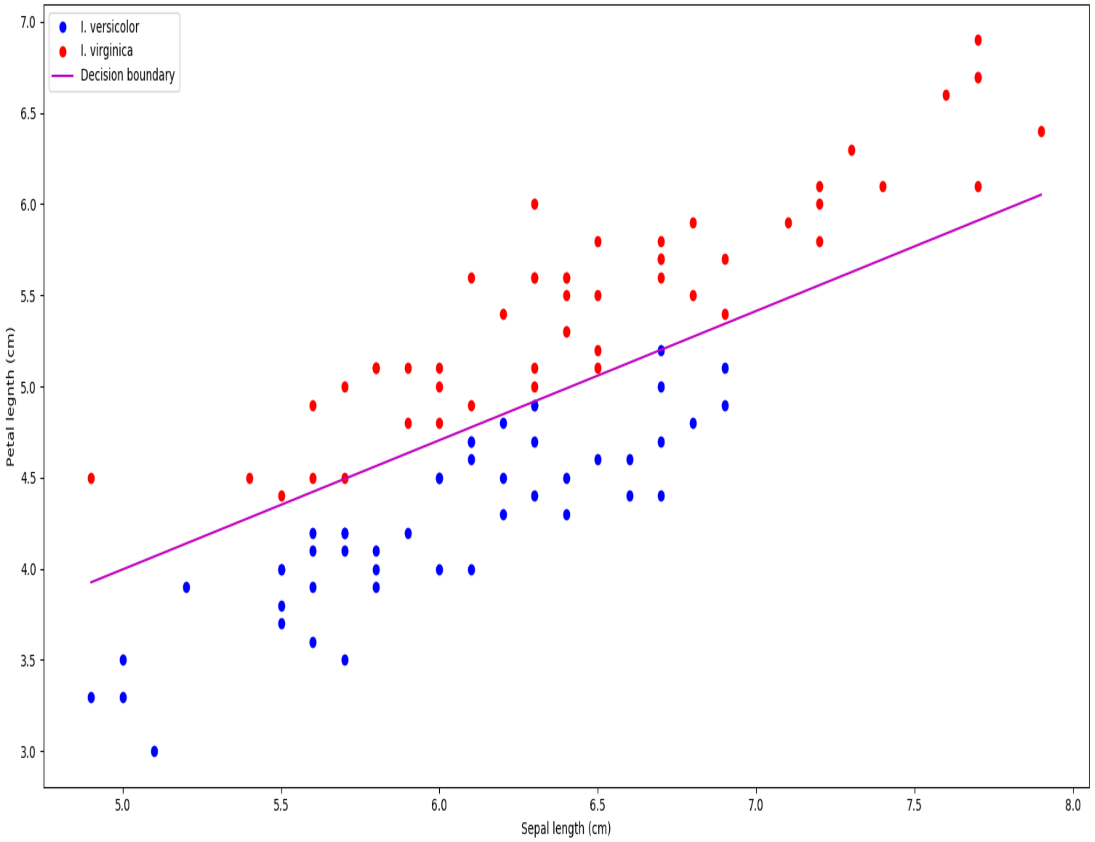
Parameters in our Model:

Learning rate: 0.01, No.of epochs: 100.

Weight W = tf.Variable(tf.zeros([2,1]))

Bias b = tf.Variable([0.0])

Graph(result):



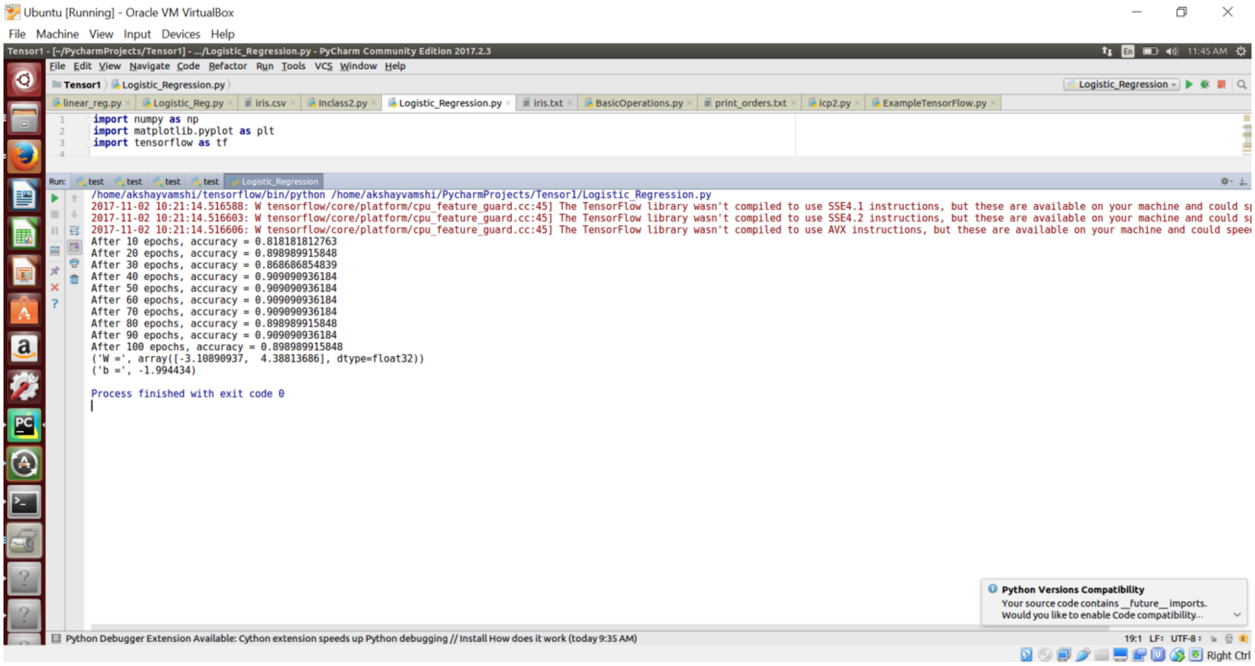
Conclusion:

From results we can see that our model achieved accuracy of 90% after 40 epochs and remains consistent. Final accuracy after calculating for 100 epochs is 89.89% with model parameters as:

Weight W = ([-3.10890937, 4.38813686], dtype=float32)

Bias b = -1.994434.

Output:



Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Tensor1 - [~/PycharmProjects/Tensor1] - .../Logistic_Regression.py - PyCharm Community Edition 2017.2.3
11:45 AM
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Tensor1 Logistic_Regression.py
Logistic_Regression
linear_reg.py Logistic_Reg.py iris.csv Inclass2.py Logistic_Regression.py iris.txt BasicOperations.py print_orders.txt icp2.py ExampleTensorFlow.py
Run: test test test test Logistic_Regression

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
```

```
/home/akshayvamshi/tensorflow/bin/python /home/akshayvamshi/PycharmProjects/Tensor1/Logistic_Regression.py
2017-11-02 10:21:14.516588: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could sp
2017-11-02 10:21:14.516603: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could sp
2017-11-02 10:21:14.516606: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could spee
After 10 epochs, accuracy = 0.818181812763
After 20 epochs, accuracy = 0.898989915848
After 30 epochs, accuracy = 0.868686854839
After 40 epochs, accuracy = 0.909090936184
After 50 epochs, accuracy = 0.909090936184
After 60 epochs, accuracy = 0.909090936184
After 70 epochs, accuracy = 0.909090936184
After 80 epochs, accuracy = 0.898989915848
After 90 epochs, accuracy = 0.909090936184
After 100 epochs, accuracy = 0.898989915848
('W =', array([-3.10890937,  4.38813686], dtype=float32))
('b =', -1.994434)

Process finished with exit code 0
```

Python Versions Compatibility
Your source code contains __future__ imports.
Would you like to enable Code compatibility...

Python Debugger Extension Available: Cython extension speeds up Python debugging // Install How does it work (today 9:35 AM)