CS 5590 – Python and Deep learning

Deep Learning

Lab Assignment-1

Submitted By

Vamsi Krishna Challa

16242688

**Introduction:**

This lab assignment is very much about implementing logistic regression and depicting it using tensor flow. We are given a task to perform logistic regression on a dataset that is not used in class. Therefore, I have considered using mnist data set for this purpose. We were also asked to perform logistic regression after changing multiple hyper parameters, which was very much associated with loss, the resultant data is analysed. The accuracy of system in all the cases is noted down and analysis is done using that data.

**Objective:**

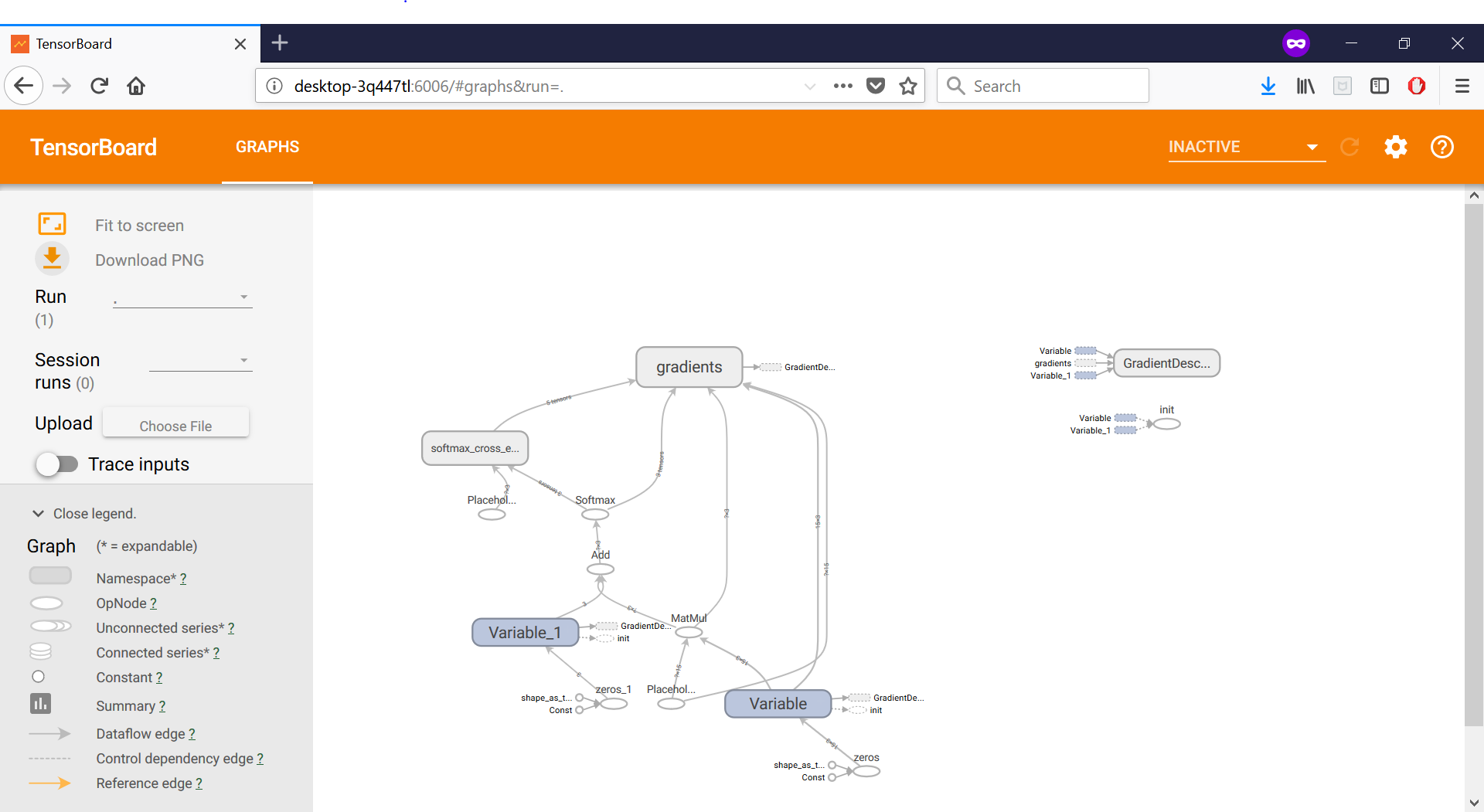
* Our primary objective is to implement Logistic regression by studying the necessary equations
* Not only implementing logistic regression generally and calculating accuracy, we need to implement logistic regression using tensor flow.
* We need to gain knowledge about tensor flow and its features for this implementation purpose.

**Approaches and methods:**

* In this implementation, we implement logistic regression multiple times for multiple hyper parameters.
* This program compares the results of logistic regression with change of various parameters.
* Here we consider mnist data set, which is dataset that contains images with hand written numbers in it.
* Each picture is of 28\*28pixel length and data in the pictures vary from 0-9.
* In various deep learning researches this dataset is used for testing purposes.
* This is a huge dataset with large number of images ranging around 60,000.
* Initially in this program, we train the system with mnist data set and load the pixel data to the program.
* After creating placeholders for data variables, we load the data into the variables. We set up various parameters that are needed to perform logistic regression.
* We train the data using a basic normalisation function y = mx +b.

**Workflow:**

* The work flow that is implemented in this program is depicted on the tensorboard.



**Dataset:**

* As mentioned earlier it is a large dataset with around 60,000 hand written images that contain numbers varying from 0-9.
* Every image is of the same size in this dataset, i.e., 28\*28 pixels. In simple terms each picture is projected using 784 pixels of data.

**Configuration:**

* Software used: Pycharm 3.6.
* We use web browser tool to attain the tensor board image that is generated over the local host data.

**Parameters:**

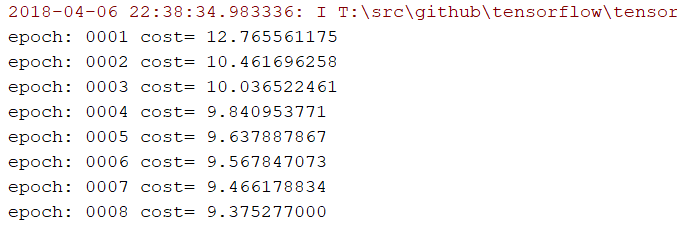
* Learning parameter: Gradient descent optimize which projects, which projects the depth of each step. Larger the learning parameters, we may be affected highly with accuracy. When we use higher learning values, we might get smaller accuracies. When we use smaller learning values, we may get larger accuracies.
* Training Epoch size: the number of training iterations.
* Step value: the gap between the pre-interval step and the post-interval step.

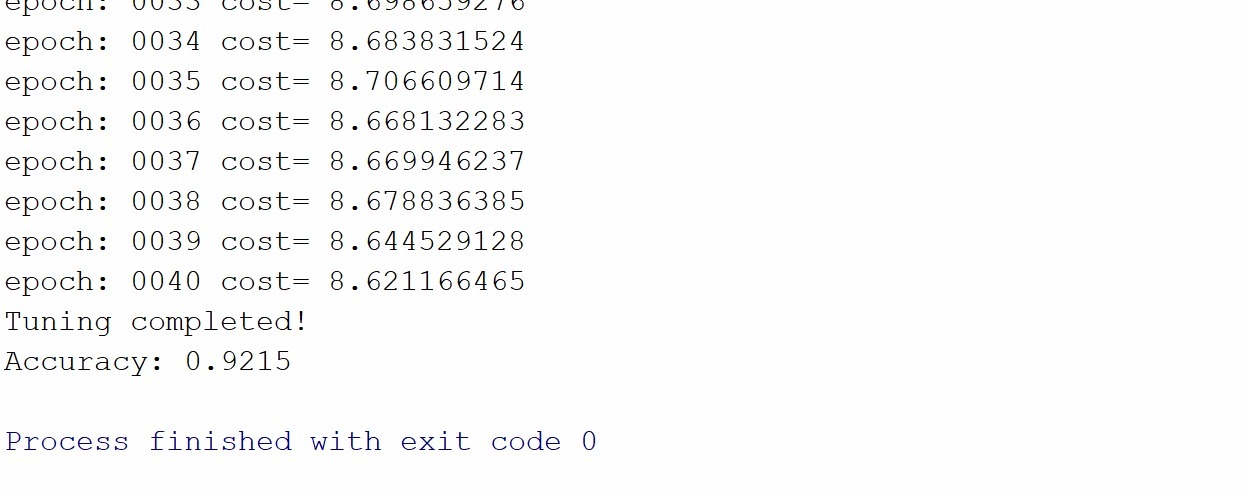
**Evaluation and discussion:**

* We train model using mnist dataset on the placeholder variables x and y using the pixel data and the numbers 0-9.

Initially I considered following as parameters:

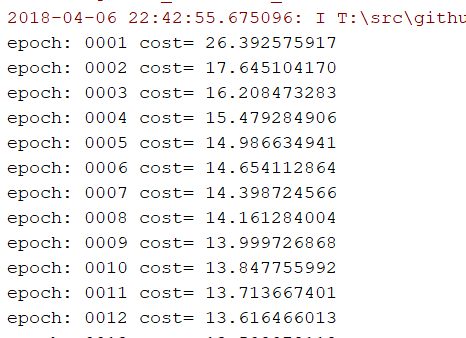
learnrate = 0.01  
trainiter = 40  
bsize = 50  
dstep = 1

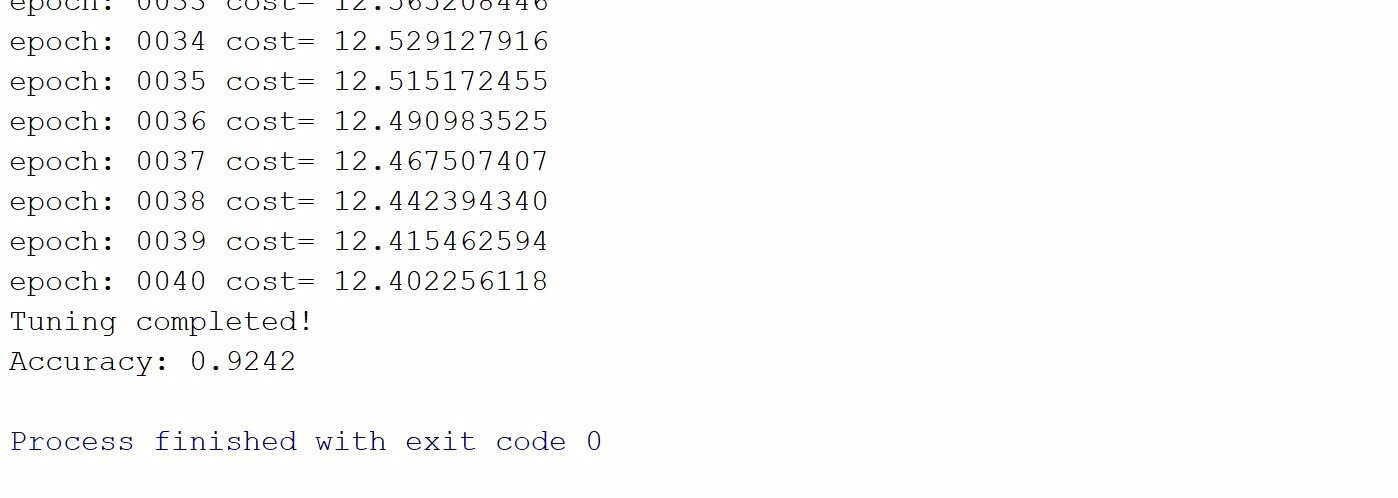




**Next I have set parameters to**

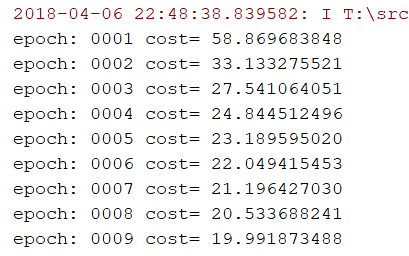
learnrate = 0.001  
trainiter = 40  
bsize = 50  
dstep = 1

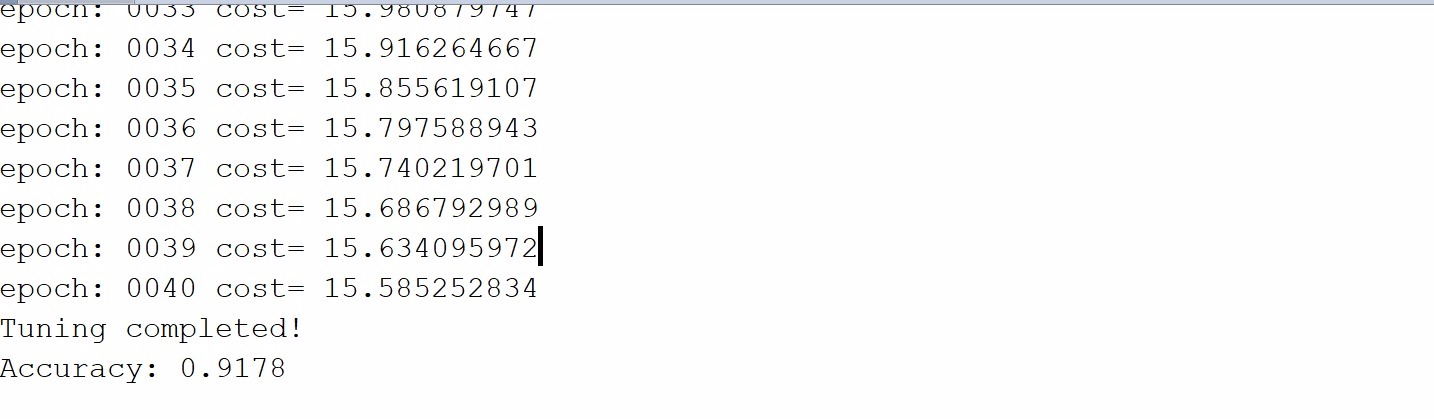




**Next I have set parameters to :**

learnrate = 0.0001  
trainiter = 40  
bsize = 50  
dstep = 1





**Conclusion:**

Here we performed logistic regression over mnist dataset, by changing various parameters in the data.

* When we are increasing learning rate, by keeping epoch and batch numbers at constant, we see that accuracy is increasing.
* In contrast to it, when we keep both learn rate and epoch at constant and increase the batch size, we see that accuracy of the model decreasing.
* The primary limitation of the tensor flow is that, we will not be able to efficiently implement logistic regression in it.
* When we tend towards, the complicated and tough neural networks, then tensor flow works very well.