## -------------------------------------------------------------------------------------------------------

## CS5542 - Lab Assignment 4 – Lab Report

## Sub team # 1-2 | Avni Mehta (Class Id: 11) | Raji Muppala (Class Id: 14)

## -------------------------------------------------------------------------------------------------------

**Task 1 Objective:**

Implement Linear Regression for project Dataset. Plot training cost using matplotlib in python. Perform cross-validation (optional)

**Input:**

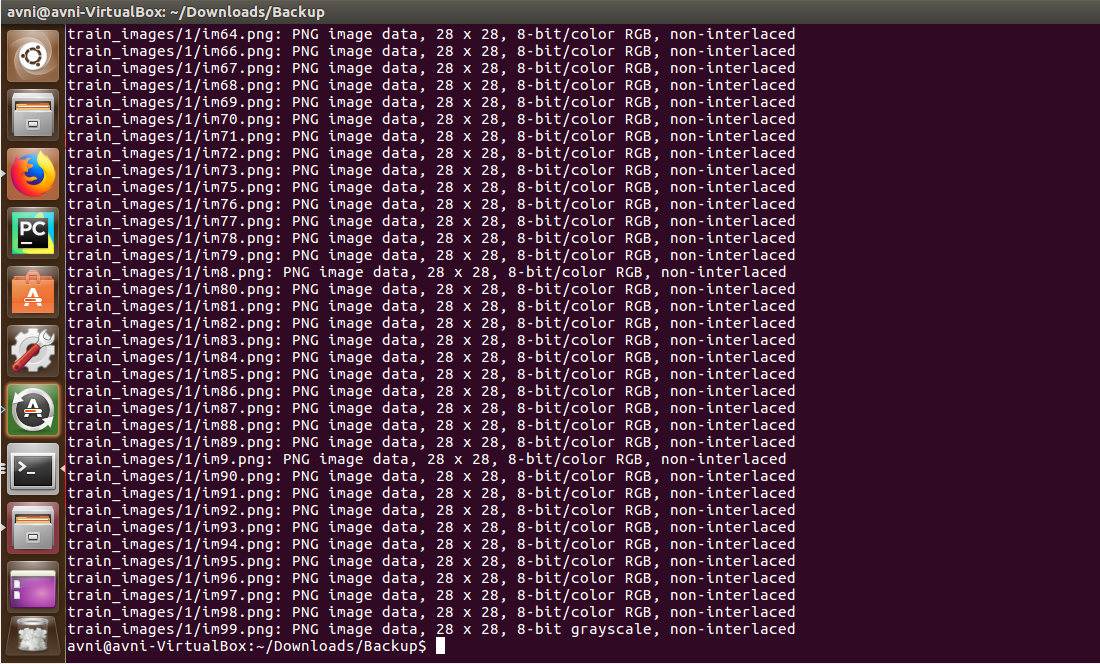
The dataset for our project is MS COCO dataset. COCO is a large-scale object detection and segment detection dataset. We have picked interior domain. The categories in this domain is kitchen, bathroom, bedroom, living room. For linear regression, we use only two categories: kitchen and bedroom.

**Data Pre-processing:**

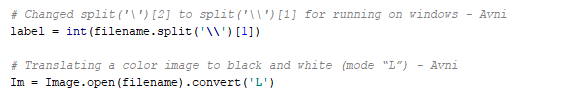
Explanation:

In order to perform the Linear Regression and Softmax on our project dataset, we had to perform the following pre-processing steps:

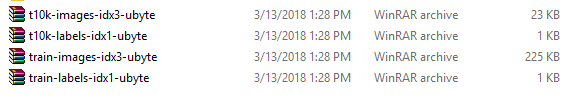
* Resize all the training and test images to 28x28 pixels.



* Change the image colour from RGB to black and white.



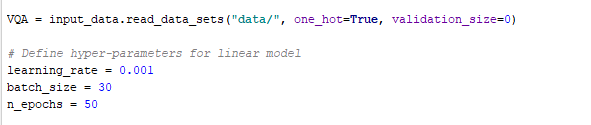
* Convert the resized images to MNIST format. Output is:



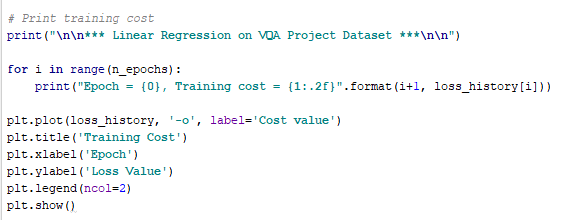
**Code:**

Explanation:

* The code for linear regression has been partly adapted from [blog](http://www.xiaoliangbai.com/2017/02/01/tensorflow-applying-linear-regression-on-mnist-dataset) for applying linear regression for MNIST dataset.
* We read the dataset using tensorflow input\_data library.
* The hyperparameters set are learning rate = 0.01, batch\_size = 30, n\_epochs = 50

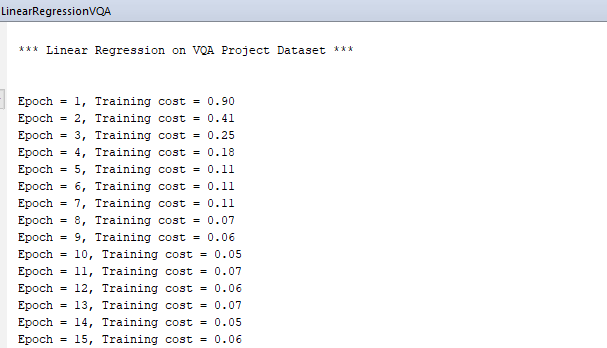


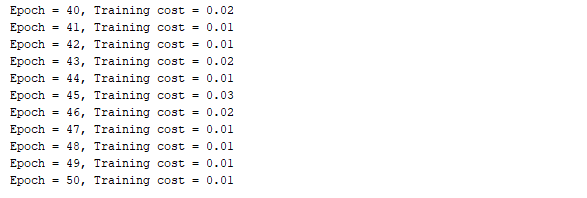
* The training is performed and training cost is calculated. The training cost is plotted on the graph using matplotlib.



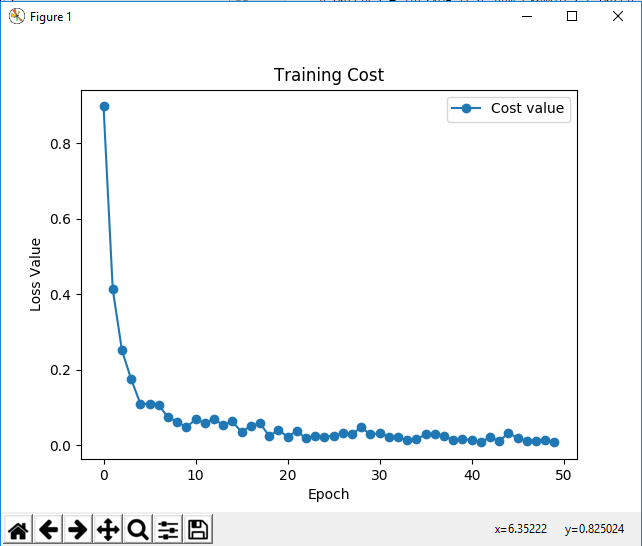
**Output:**

Training cost printed per epoch. For 1st epoch, the cost was 0.9, while for the 50th epoch the cost is 0.01. This model is good for the given data.

****

****

Plot for training cost.



**Task 2 Objective:**

Implement Softmax classification for the project dataset. Report accuracy. Show visualizations on Tensorboard.

**Input:**

The dataset for our project is MS COCO dataset. COCO is a large-scale object detection and segment detection dataset. We have picked interior domain. The categories in this domain is kitchen, bathroom, bedroom, living room. For Softmax classification, we use all four categories. The train-test split is 90-10.

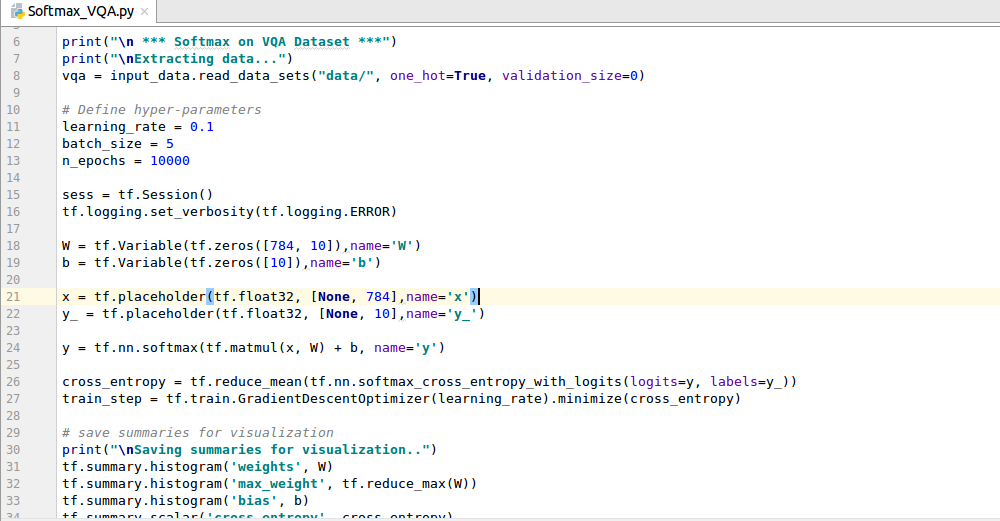
The accuracy for image classification has to be compared with Spark, Clarifai API and Softmax. We have limited our number of train images per class to 50, because our system throws memory error while running Random Forest on Spark and Clarifai API. In order to make a fair comparison, we have used the same data set for all three models.

**Code:**

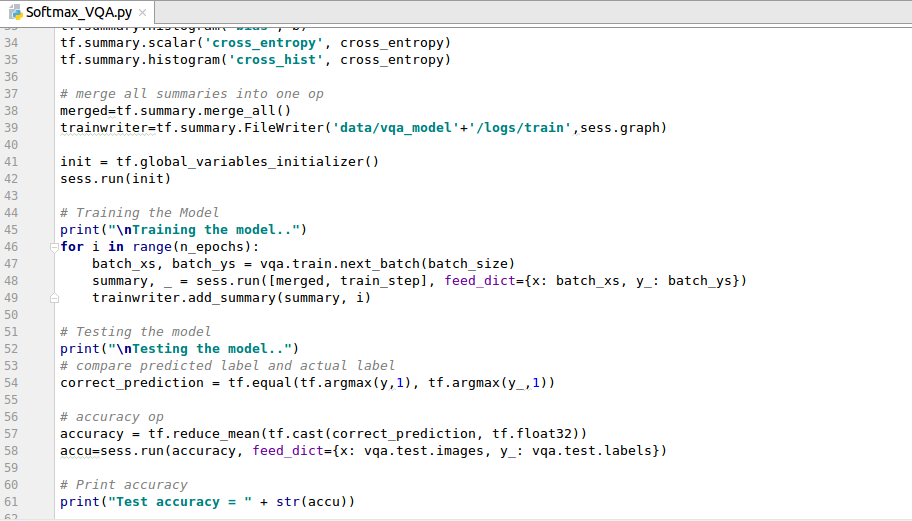
a. Reading the project dataset using MNIST library methods. Setting the hyperparameters

b. Creating placeholders for data and variables for weights and bias. Evaluating the score using y = Wx + b. Cross entropy is used with Softmax.

c. Saving the summaries for visualization



d. Training the model in batches. Testing the model based on accuracy.

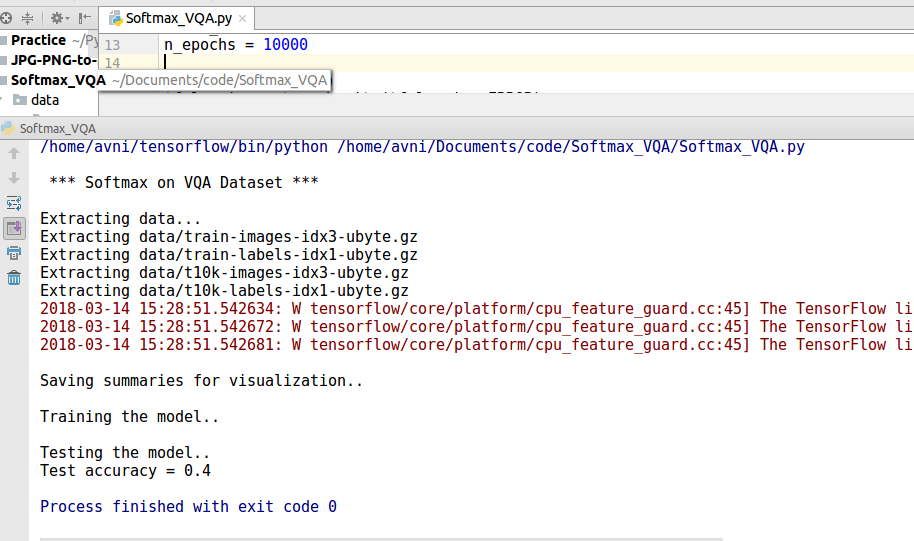


**Output:**

**Output Observation**:

We have achieved a low accuracy of 40% for this classifier. This is because of the following reasons:

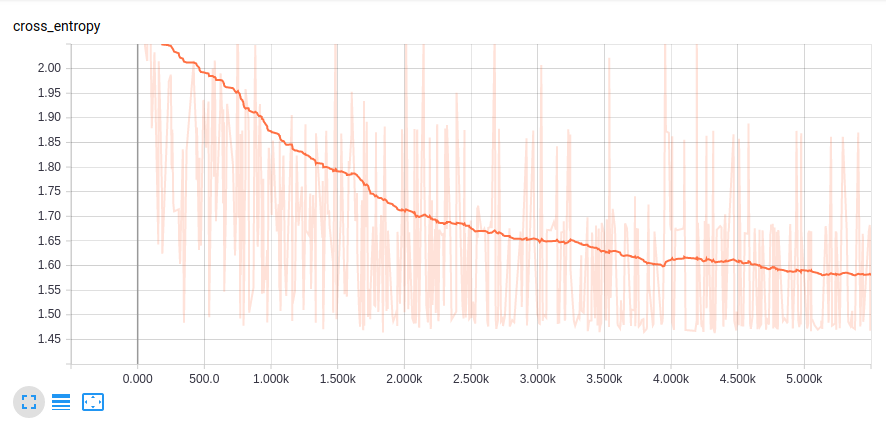
1. Very small dataset (50 train images per class)
2. Image size reduced to 28x28 pixel and color changed to black and white. This caused lot of information loss from the data.
3. Even though we are doing deep learning, Softmax is not the best classifier for image classification. A deep network with CNN layers would produce a much higher accuracy for the same data.
4. We have increased the no of epochs to 10000 which gave us 40% accuracy. For 1000-5000 epochs we had lower 33.33% accuracy.



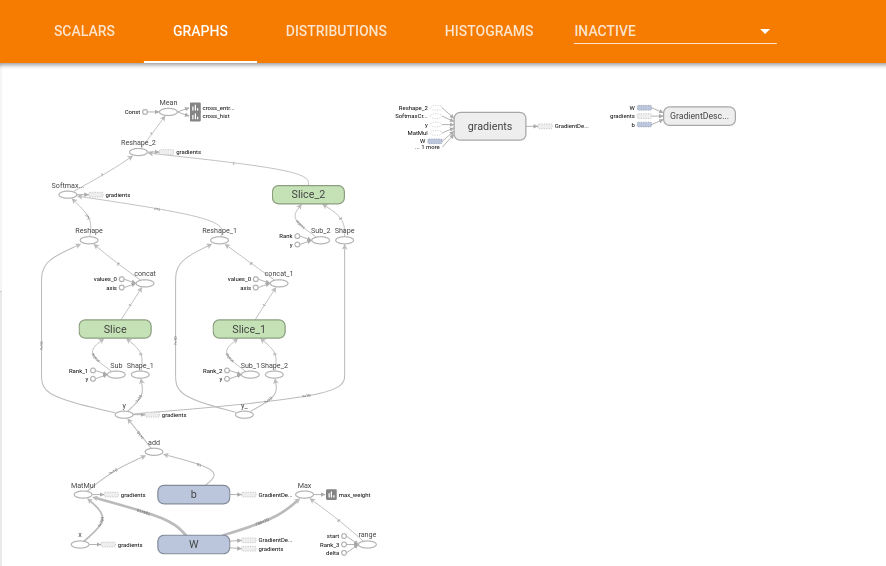
**Tensorboard Visualization**:

We have the following visualization:

Cross Entropy: The cross entropy has decreased very slowly from approx. 2.05 in first epoch to approx. 1.60 in the 10000th epoch.



Graph: The graph visualization helps to understand the complicated tensorflow computation.



Histogram: The histogram shows how the distribution of tensors like weight, bias, cross\_entropy has changed over time.

