CS5590/490-Python/Deep Learning

DL- ASSIGNMENT-1

By: Syed Jawad Hussain Shah

Student ID: 16117985

Class ID: 46

**Author:** Syed Jawad Hussain Shah

**Introduction:**

Logistic regression is a statistical model that classifies the data into two categories or classes. This model can be applied to different data sets where a linear boundary can be obtained between data values such that data points belongs to only one side of the boundary. TensorFlow is open source library that uses data flow graphs to compute numerical problems. Within TensorFlow, TensorBoard is a composition of web applications used to analyzed and inspect tensor graphs and run.

**Objective:**

The objective of this lab task is to get familiar with logistic regression and its implementation in python by using TensorFlow library. The features of this lab include to write a program to implement a logistic regression model on a certain data for classification in python using TensorFlow library. The graphs of this program were written in an event file and were viewed in the TensorBoard by writing a command in the python terminal. The last part of the task was to change the hyperparameter of the initial program the compare the difference between the accuracy of the two approaches.

**Approaches/Methods:**

The approach used in this program is logistic regression. The data used is taken from Mixed National Institute of Standards and Technology (MNIST)database. This database contains images for the purpose of training several imaging systems. The data was composed of images of hand written digits. This image data was loaded using TensorFlow characteristic TF learn. Once the data was loaded, it was kind of divided into training set, test set and validation set. After that, Logistic model was constructed. The creation of logistic regression model pretty much same as liner regression model practiced in class ICP last week.

Once the logistic regression model was created, it was applied on MNIST data and the accuracy of the model was measured. The lost function was also calculated by calculating the mean square error (Y – Y\_Predict).

**Workflow:**

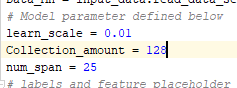
1. MNIST data was imported



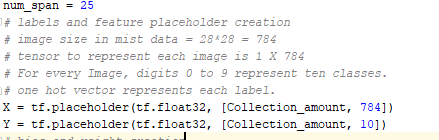
1. Data was read by using TensorFlow Learn’s built in function.



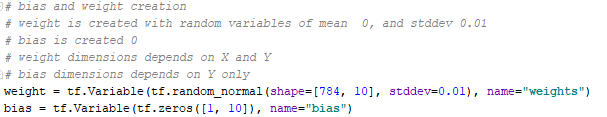
1. Parameters were defined for the model



1. Placeholders creation for features and labels. The image in MNIST is 784 (28\*28) so for that reason 1\*784 tensor is used to represent image. For every Image, digits 0 to 9 represent ten classes and one hot vector represents each label.



1. Bias and weight was created. weight is created with random variables of mean 0, and standard deviation of 0.01 and bias is initialized to 0. The weight dimensions depend on X and Y and the bias dimensions depends on Y only.

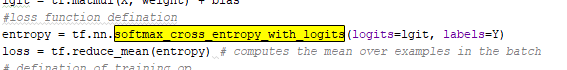


1. The value of Y was predicted by using X, weight and bias, the model predicts the label for a given image with the help of softmax layer. The possibility of digits were represented with tensor of size ten.



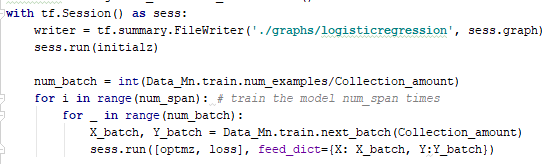
1. Defining a loss function by using softmax cross entropy with logits. Mean cross entropy was computed and softmax is intermally applied.

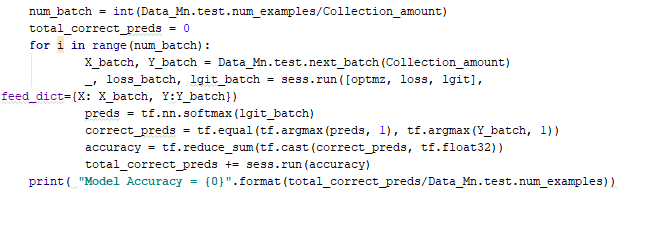
|  |  |
| --- | --- |
|  |  |



1. Defining op training with learning scale of 0.01 for cost minimization by using gradient descent.







**Datasets:**

The data used is taken from Mixed National Institute of Standards and Technology (MNIST)database. This database contains images for the purpose of training several imaging systems. The data was composed of images of hand written digits. The image is 784, 28 X 28 pixels.

**Parameters:**

The parameters used are as following:

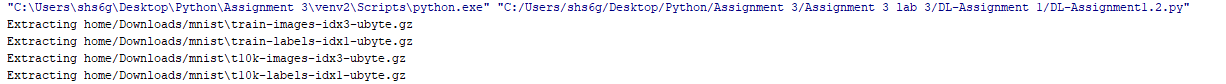
learn\_scale = 0.01 #(Learning rate)  
Collection\_amount = 128 # ( batch size)  
num\_span = 25 # (number of epochs)

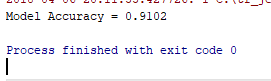
**Evaluation and discussion:**

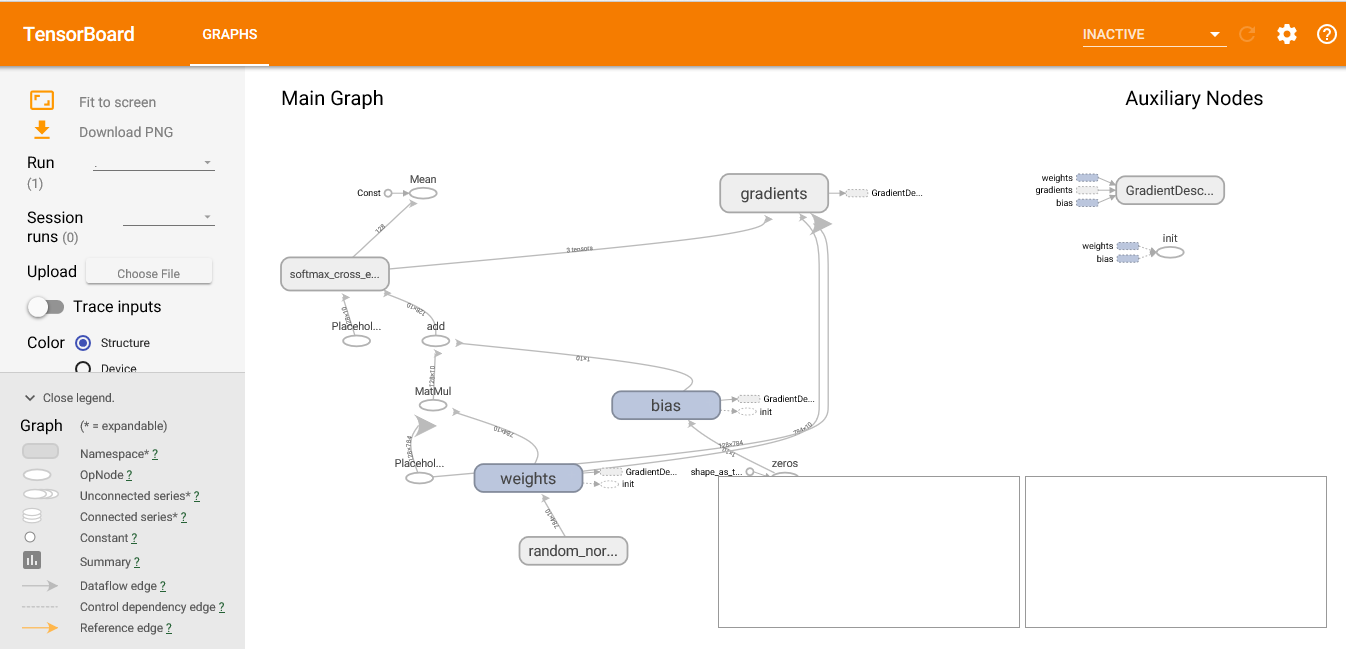
Logistic regression is generally used for 2 class classification. it is based on Maximum likelihood estimation. Logistic regression estimates probability immediately and conditionally. It works like ordinary least squares regression but on the logit of the dependent variable and is not very sensitive to outliers. The model accuracy was 91% which shows that this approach is quite reliable and can be used to classify more of the similar data. When the number of turns (epochs) was put less than 10, the accuracy fell to below than 90 %, which means that model performs better with more repetitions.

**Conclusion:**

The model accuracy was 91% which shows that this approach is quite reliable and can be used to classify more of the similar data. However, when the number of turns (epochs) was put less than 10 to change the hyperparameters, the accuracy fell to below than 90 %, which means that model performs better with more repetitions. The graphs of the model can be seen on TensorBoard. The screenshots of TensorBoard Graph is pasted along with results are pasted below:







**References**

<https://opensource.com/article/17/11/intro-tensorflow>

<https://learningtensorflow.com/Visualisation/>