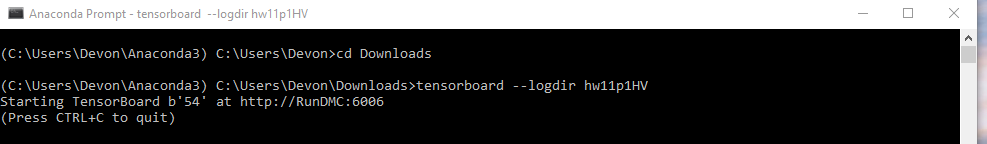
## HU Extension Assignment 11 E63 Big Data Analytics

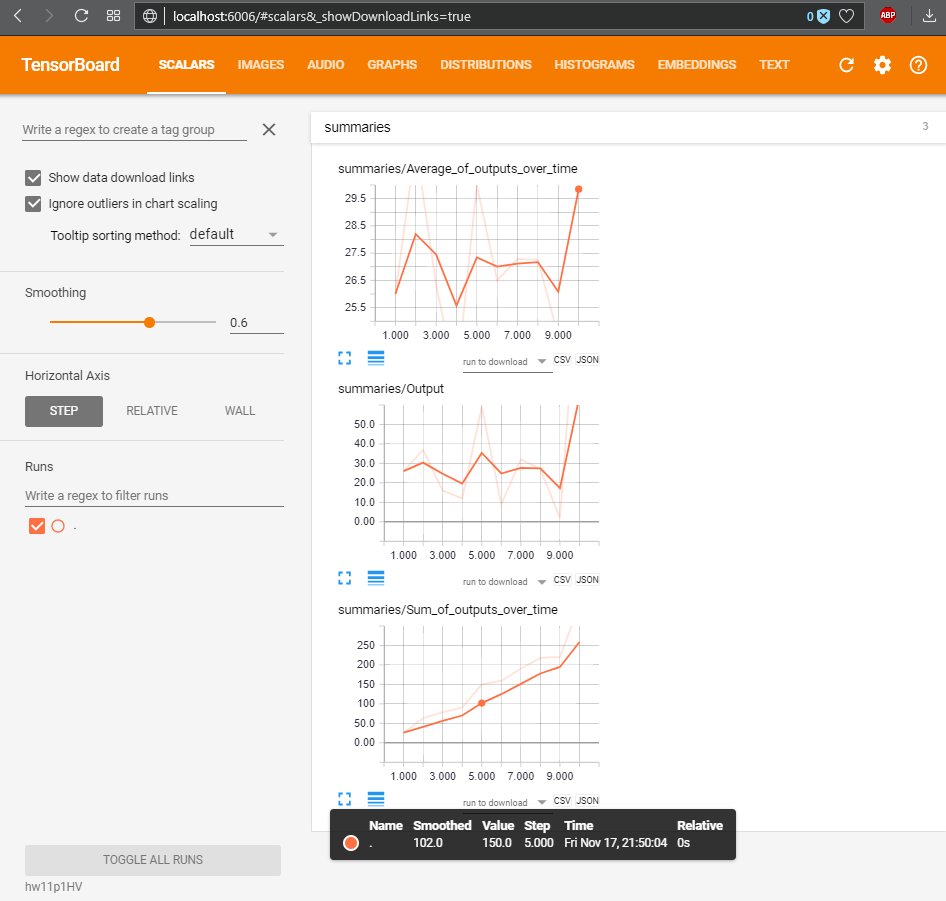
### **Student: Huynh Vo** Due by 4:00 PM EST on Saturday, 11/18/2017

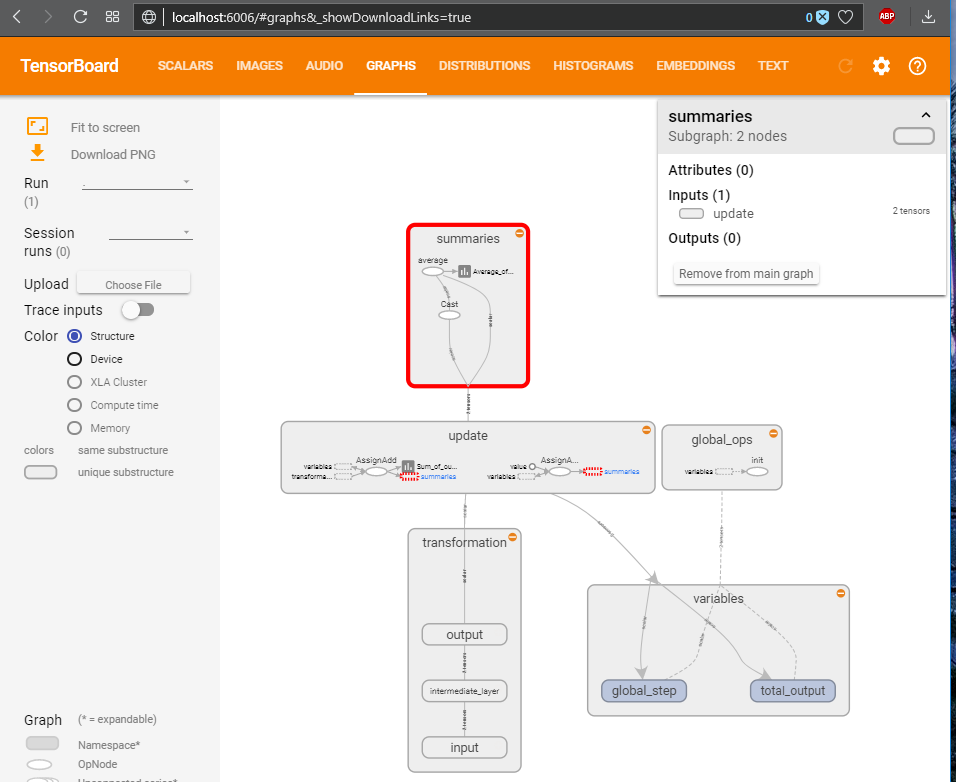
You are welcome to implement TensorFlow problems in this problem set in any of supported languages.

**Problem 1.** Consider provided Jupyter notebook Summaries and NameScopes.ipynb. Add one more output summary. For example, calculate the rolling mean of all one dimensional tensors passed as arguments of run\_graph function. Provide working notebook and images of your graphs and calculated summaries. In the Word document presented as your solution provide snippets of additional or modified code. **(15%)**



**Below is original graphs before I add more out summary**





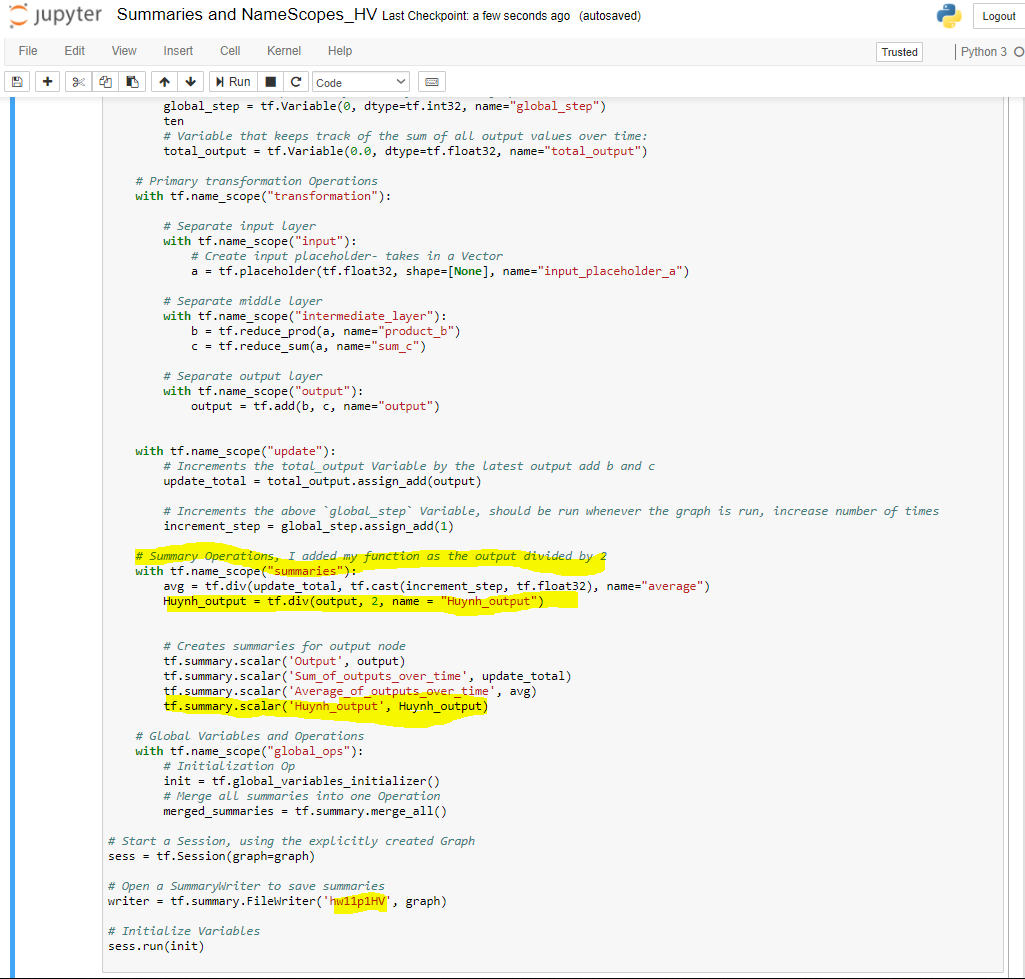
**#Below is my code in Summaries and NameScopes\_HV.ipynb (I also attached this file in Canvas):**

#Separate output layer:

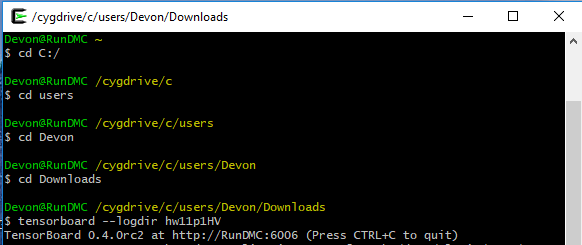
Huynh\_output = tf.div(output, 2, name = "Huynh\_output")

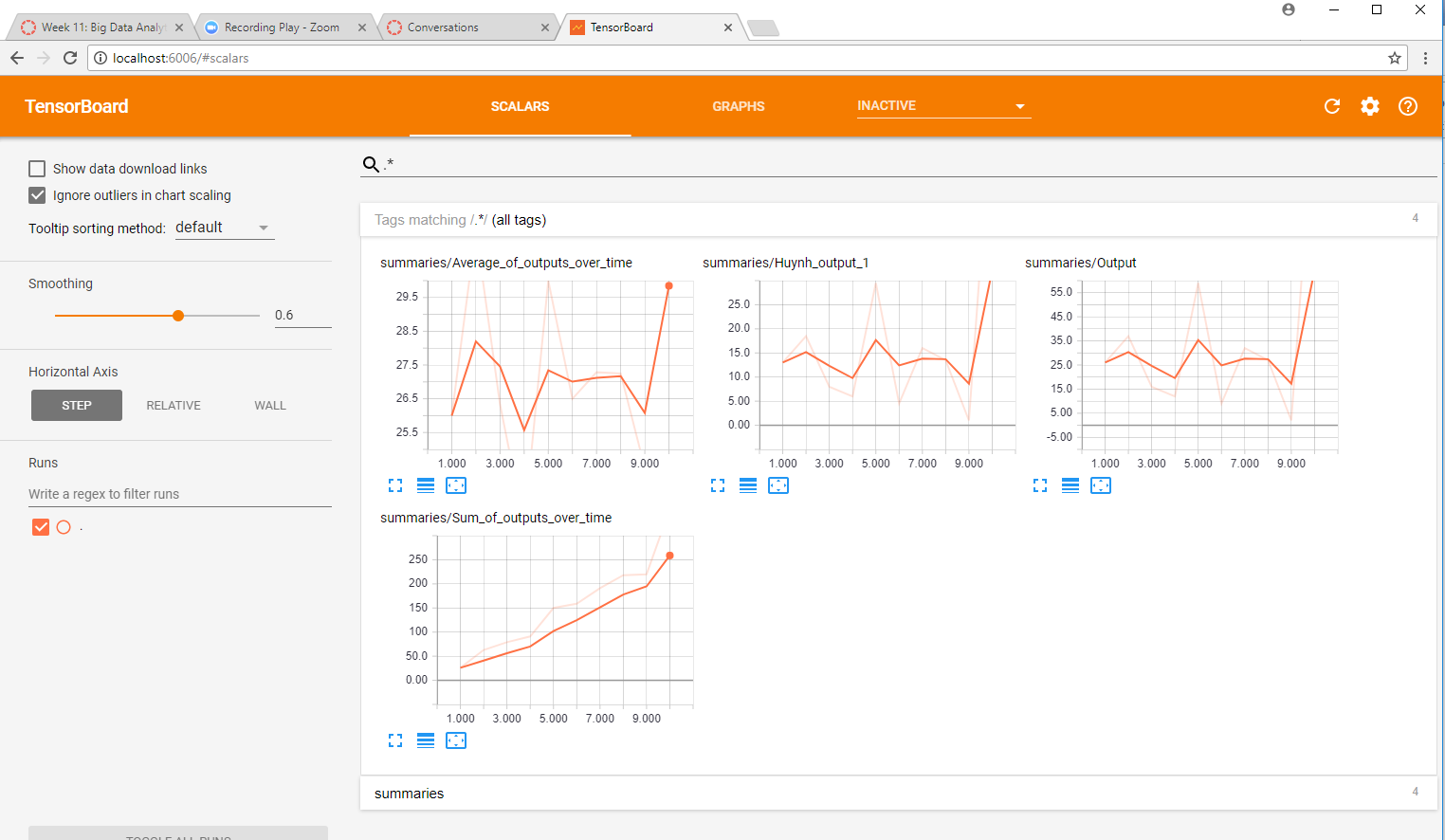
# Creates summaries for output node

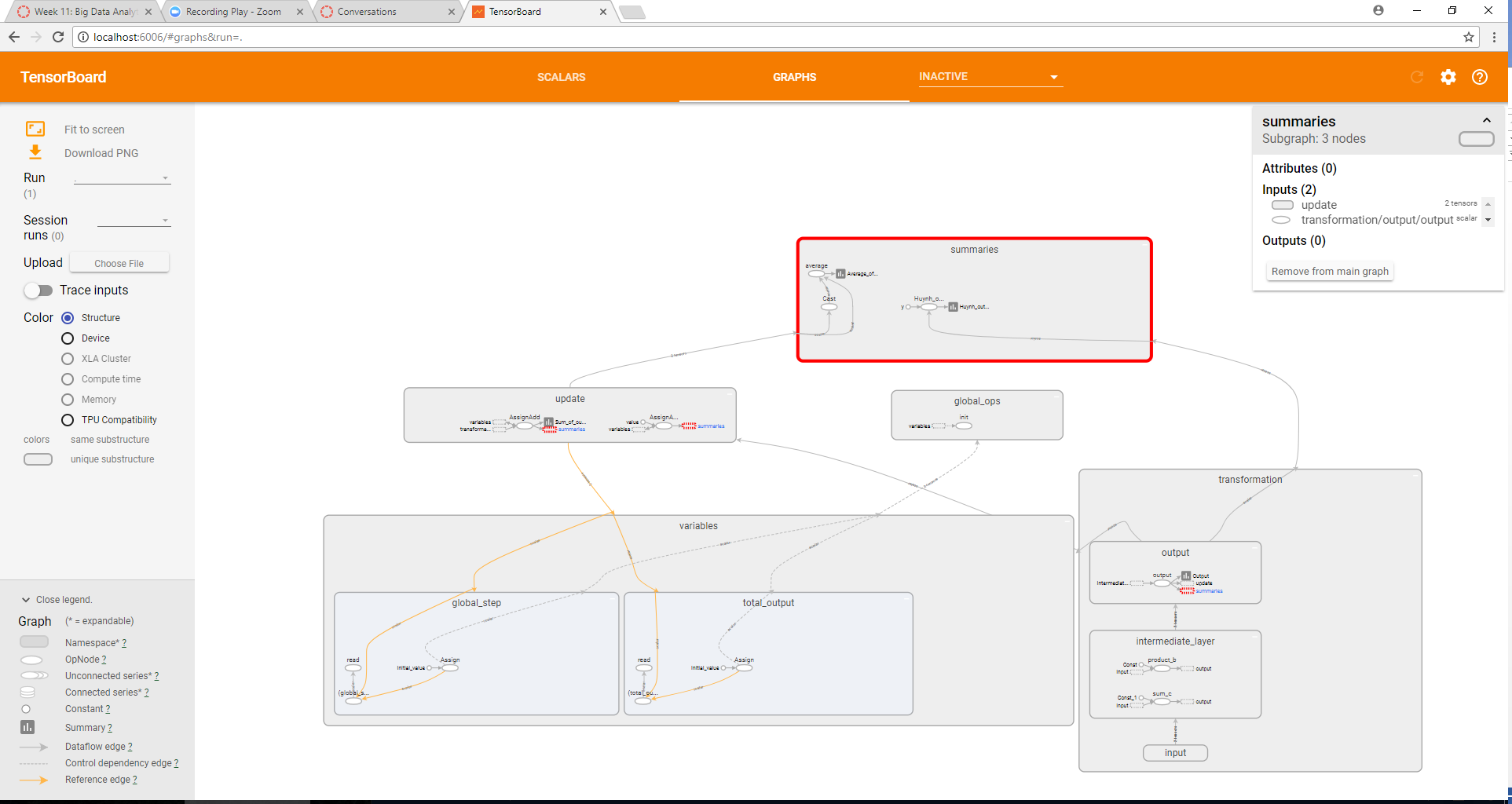
tf.summary.scalar('Huynh\_output', Huynh\_output)



**# After running the code, I go to Cygwin because in Anaconda prompt, my graphs were not shown. Please zoom in the graph if you would like to see details, thanks!**







**Problem 2**. Consider the attached file logistic\_regression\_mnist.py. Search through TensorFlow API documentation and the Internet and describe for us what is the meaning and purpose of functions used in step 5 and step 6. Demonstrate that you can run the code successfully. Fetch for us the TensorBoard Graph. Vary parameter batch\_size through values: 8, 64, 128, 256 and report and plot changes in the execution time and accuracy. Keep other parameters the same as in the original program. Similarly, vary parameter learning\_rate through values 0.001, 0.005, 0.01, 0.02 and 0.05. Report and plot changes in the execution time and accuracy. **(25%)**

Step 5: define loss function

Use cross entropy of softmax of logits as the loss function

**entropy = tf.nn.softmax\_cross\_entropy\_with\_logits(logits=logits,labels=Y, name='loss')**

**loss = tf.reduce\_mean(entropy) # computes the mean over all the examples in the batch**

Step 6: define training op

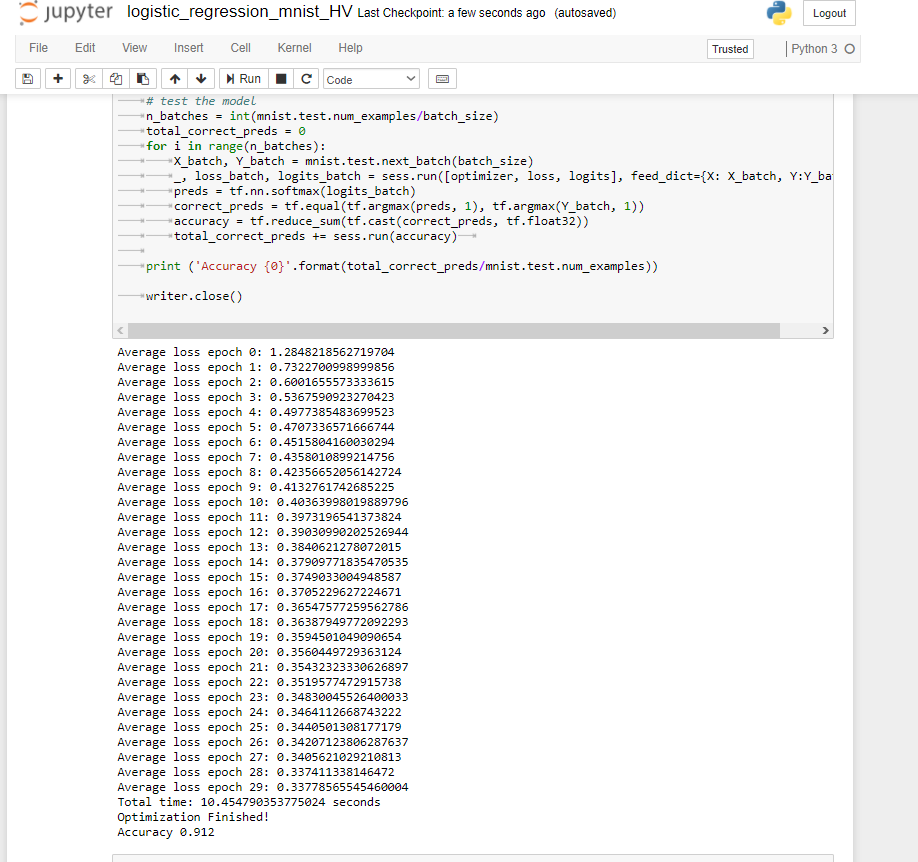
Using gradient descent with learning rate of 0.01 to minimize loss

**optimizer = tf.train.GradientDescentOptimizer(learning\_rate).minimize(loss)**

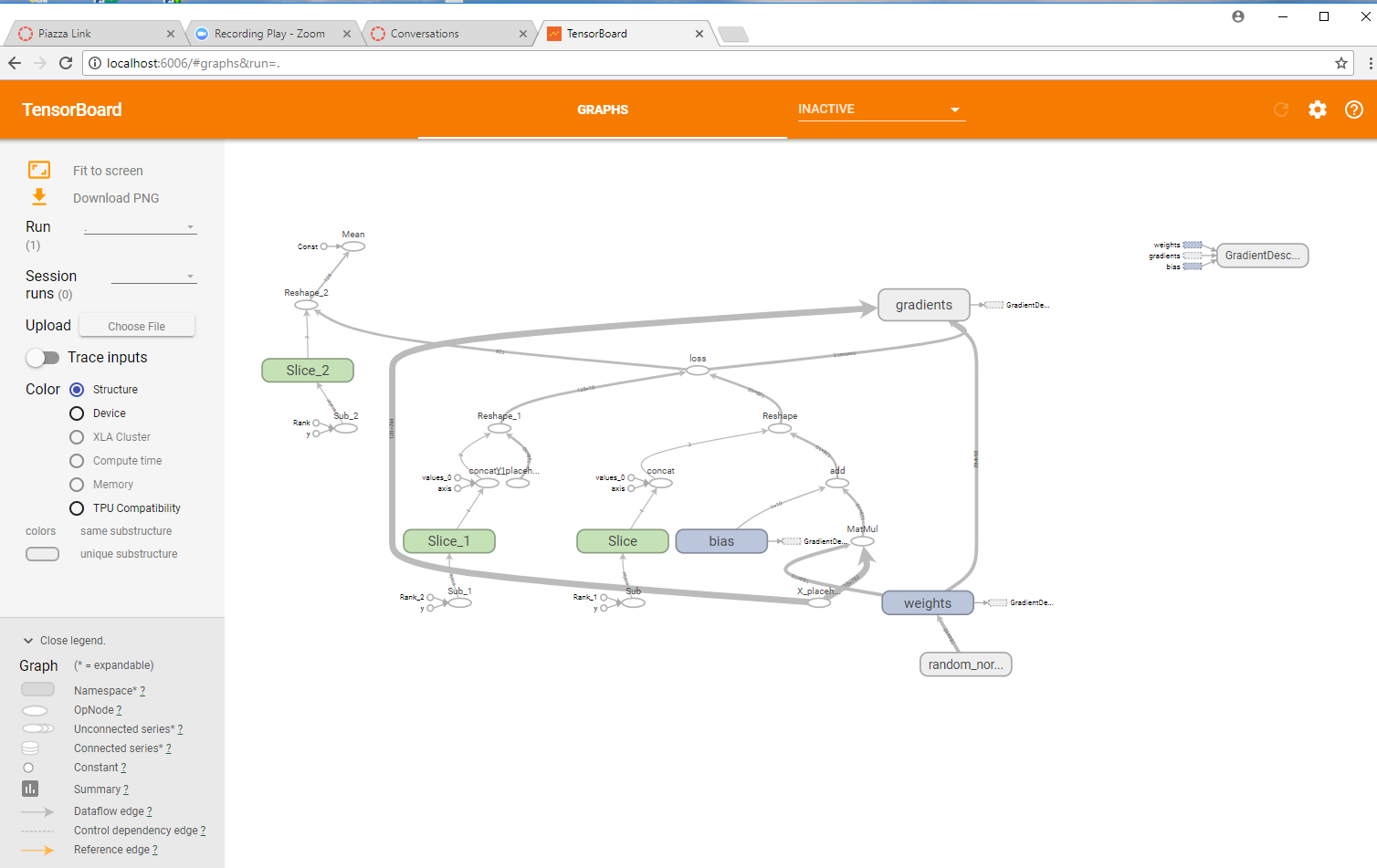
|  |  |
| --- | --- |
| Function | Description |
| tf.nn.softmax\_cross\_entropy\_with\_logits() | Measures the probability error in discrete classification tasks in which the classes are mutually exclusive (each entry is in exactly one class). #This version of softmax function lets you work with training sets containing the probabilities of each example to belong to every class. The final output of the model will always be one single class value, this version supports more flexible training data compared to the sparse version. |
| loss = tf.reduce\_mean() | Compute the mean over all the examples in the batch |
| tf.train.GradientDescentOptimizer().minimize() | Gradient is the direction of the steepest slope. We choose directions randomly and looking for actual value. Gradient descent algorithm is used for optimizing the model parameters. Its output vector indicates the direction of maximum growth for the loss function. Its like an arrow that will indicate in every point of the function where you should move to increase its value (learning\_rate is in this case). |

**# Demonstrate that you can run the code successfully:**

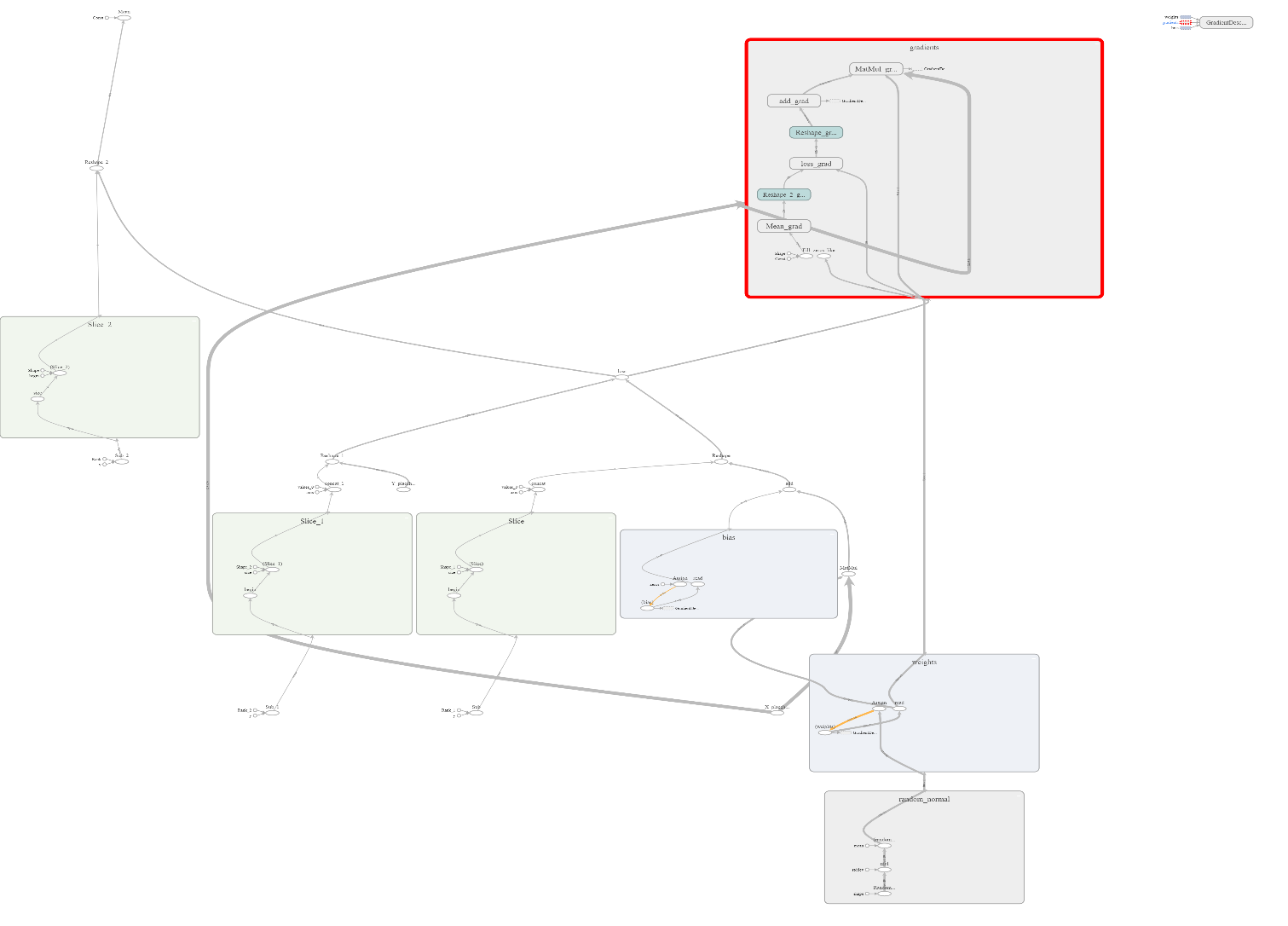
**Please also see attachment “logistic\_regression\_mnist\_HV.ipynb” in Canvas**



# Fetch for us the TensorBoard Graph:

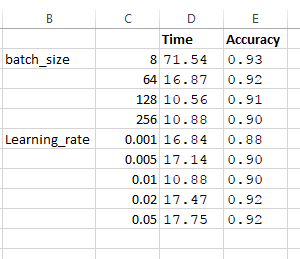


# More zoomed in:

****

**#** Vary parameter batch\_size through values: 8, 64, 128, 256 and report and plot changes in the execution time and accuracy. Also, vary parameter learning\_rate through values 0.001, 0.005, 0.01, 0.02 and 0.05 and report and plot changes in the execution time and accuracy:

**Please see logistic\_regression\_mnist\_HV.ipynb (attached in canvas) for results**



**Problem 3**. Fetch Iris Dataset from <https://archive.ics.uci.edu/ml/datasets/Iris> and make attached Python script, softmax\_irises.py work. You might have to upgrade the script to TF 1.x API. Generate TensorBoard graph of the process and use scalar summary to presenting variation of the loss function during the training process. Report the results of the evaluation process. **(35%)**

**Please see attachment softmax\_irises\_HV.ipynb , I tried my best.**

**Problem 4.** Analyze all relevant and non-obvious individual steps in the script, softwmax\_irises.py by examining their inputs and outputs. When convenient, use existing Iris Dataset. When convenient, you are welcome to provide your own inputs. Please examine and describe actions of functions and operations within those functions:

* combine\_inputs(), line 13
* inference(), line 17
* read\_csv(), line 25
  + decode\_csv() line 34
  + train.shuffle\_batch(), line 37
* inputs(), line 43
  + label\_number = tf.to\_int32(…), line 49
  + features = tf.transpose(..), line 57
* evaluate(), line 67
  + predicted = tf.cast(tf.arg\_max(inference(X), 1).., line 69
  + tf.reduce\_mean(tf.cast(tf.equal(predicted,Y),.,line 71
* threads = tf.train.start\_queue\_runners(sess=sess, coord=coord).., line 85

Please describe the effect of every function or command by providing an illustrative input and output set of values and well as a brief narrative. Please rely on TensorFlow API as much as possible. **(%25)**

|  |  |
| --- | --- |
| Function | Description |
| combine\_inputs(), line 13 | define a function X(W^T)+b. This is a product of 2 vectors and a bias. Combine\_inputs is a named argument, comebine inputs into a single output |
| inference(), line 17 | compute inference model over data X and return result. Tensorflow contains an embedded activation of the softmax function. |
| read\_csv(), line 25 | read in the file irisdata. Queues the filneame for an input, then reads the text in file one line at a time, then puts them in a "key, value" format stings |
| decode\_csv() line 34 | decode\_csv will convert a Tensor from type string (the text line) in a tuple of tensor columns with the specified defaults, which also sets the data type for each column |
| train.shuffle\_batch(), line 37 | train/adjust model parameters. Creates batches by randomly shuffling tensors batch actually reads the file and loads "batch\_size" rows in a single tensor |
| inputs(), line 43 | read/generate input training data X and expected outputs Y |
| label\_number = tf.to\_int32(…), line 49 | convert class names to a 0 based class index. label\_number tells what number is of that label, individually numbering labels creates an index of the different Iris flower types to be identified. I changed tf.pack to tf.stack |
| features = tf.transpose(..), line 57 | based on label\_number, we pack all the features that we care about in a single matrix; we then transpose to have a matrix with one example per row and one feature per column. |
| evaluate(), line 67 | evaluate the resulting trained model |
| predicted = tf.cast(tf.arg\_max(inference(X), 1).., line 69 | cast a tensor to a new type in this case int32. Argmax returns the index with the largest value across axes of a tensor doing all of this to get a predicted value |
| tf.reduce\_mean(tf.cast(tf.equal(predicted,Y),.,line 71 | checking to see if the prediction is accurate and outputting the result |
| threads = tf.train.start\_queue\_runners(sess=sess, coord=coord).., line 85 | starts all queue runners collected in the graph. It returns the list of all threads. |

Overview: I think my knowledge of tensorflow is still not strong. There are a lot of confusions still for me.

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of all essential command and snippets of results into the Word document with explanations of the purpose of those commands. We cannot retype text that is in JPG images. Please, always submit a separate copy of the original, working scripts and/or class files you used. Sometimes we need to run your code and retyping is too costly. Please include in your MS Word document only relevant portions of the console output or output files. Sometime either console output or the result file is too long and including it into the MS Word document makes that document too hard to read. PLEASE DO NOT EMBED files into your MS Word document. For issues and comments visit the class Discussion Board.