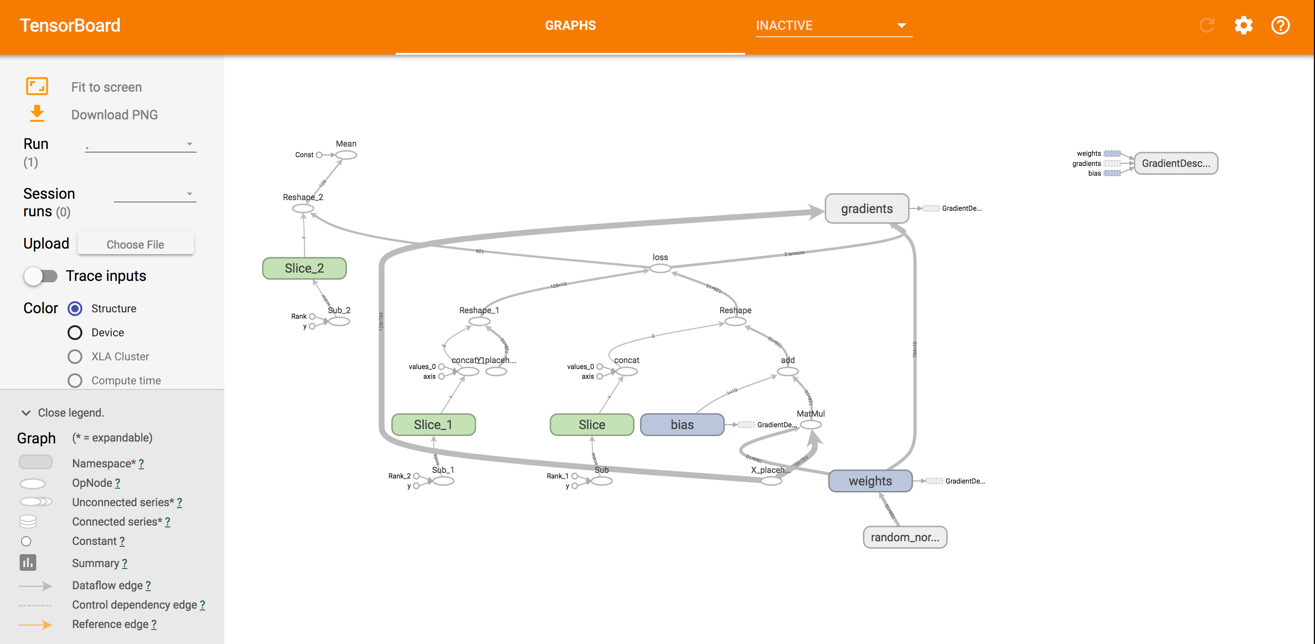
## HU Extension Assignment 11 E63 Big Data Analytics

### Handed out: 11/10/2017 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 summay. 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%)**

**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%)**

****

**BatchSize Time (**seconds**) Accuracy**

**256** 12.366281032562256 0.904

**128** 18.631927013397217 0.9113

**64** 24.737818002700806 0.917

**8** 105.58878779411316 0.9251

**Learning Rate Time (**seconds**) Accuracy**

**.001** 20.745763063430786 0.8567

**.005** 20.092628240585327 0.893

**.01** 16.71730494499206 0.904

**.02** 18.73343586921692 0.9111

**.05** 9.41432499885559 0.9173

**step 5**

We are optimizing our model by minimizing how wrong we are. With our labels in one hot encoded label, we compare these with the class probabilities predicted by the model. We use the cross\_entropy to measure this. We are punishing the model for more for less accurate predictions.

**step 6**

TensorFlow provides built-in optimizers to take advantage of the loss function above. Gradient descent will slowly push our weights toward better results.

 We are iteratively moving in the direction of steepest descent as defined by the negative of the gradient, to see where the cost function decreases.The size of the steps is defined by learning rate

**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%)**

**bin** python3 tf\_upgrade.py --infile iris-fix.py --outfile iris-fix-upgrade.py

TensorFlow 1.0 Upgrade Script

-----------------------------

Converted 1 files

Detected 0 errors that require attention

--------------------------------------------------------------------------------

Make sure to read the detailed log 'report.txt'

Upgraded TF and made changes to various functions.

**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
  + We are trying to carry out Logistic\_Regression
  + For this we have to combine the input like tf.mul(X, W) + b
    - Here W is weight and b is bias
  + In order to feed the function with the multiple dimensions, or features from the examples of our training datasets, we need to combine them into a single value
* inference(), line 17
  + inference calls the softmax activation function here since we are doing multiple classification logistic regression model for IRIS. The calculated value so, should be in between 0 and 1 and sum of all should be equal to 1.
* read\_csv(), line 25
* Based on what tensorflow docs say , this is how we read data from csv
  + (Ref:https://www.tensorflow.org/versions/r1.0/programmers\_guide/reading\_data)
  + decode\_csv() line 34
    - This creates tuple of tensor columns from input Strings
  + train.shuffle\_batch(), line 37
    - batch reads file and loads "batch\_size" rows in a single tensor
    - We use single filename queue
* inputs(), line 43
  + - features and labels defined
    - Make input ready by putting all features we care about into a single matrix and transpose to have a matrix with one example per row and one feature per column.
  + label\_number = tf.to\_int32(…), line 49
    - tf.pack:Packs the list of tensors in values into a tensor with rank one higher than each tensor in values, by packing them along the axis dimension.
    - tf.argmax:Returns the index with the largest value across axes of a tensor.
    - That’s how we are converting class names to a 0 based class index.
  + features = tf.transpose(..), line 57
    - transpose to have a matrix with one example per row and one feature per column.
* evaluate(), line 67
  + - After training, we apply an evaluation phase; where we execute the inference against a different set data to which we also have the expected output, and evaluate the loss for it
  + predicted = tf.cast(tf.arg\_max(inference(X), 1).., line 69
  + tf.reduce\_mean(tf.cast(tf.equal(predicted,Y),.,line 71
    - Computes the mean of elements across dimensions of a tensor.
    - Reduces input\_tensor along the dimensions given in axis.
* threads = tf.train.start\_queue\_runners(sess=sess, coord=coord).., line 85
  + It just starts threads for all queue runners collected in the graph. It returns the list of all threads.

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)**

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.