HW 3: TensorFlow
CSCI 5525: Machine Learning

Most of the problems here are relatively straight forward and constitute learning exercises--their purpose is to get you acquainted with the TensorFlow API. Do them from scratch and you'll learn a lot. I want you to get more familiar with the TensorFlow documentation that you can find at https://www.tensorflow.org/api_docs/python/.

Problem #1 (20 points) is contained in the "OpsProblems.py" file. Complete the code in template form and submit the file.

Problem #2 (20 points) Complete a basic logistic regression analysis of the MNIST dataset by finishing the code in TFLogRegStarter.py and demonstrating it works by showing test performance. Please submit output from TensorBoard, the completed template file and test error.

Problem #3 (20 points) Complete a convnet analysis of the MNIST dataset by finishing the code in ConvNetTemplate.py and demonstrating it works by showing test performance. Please submit output from TensorBoard, the completed template file and test error.

The remaining problems are more open-ended and less structured. These problems don't have starter code. The problems have several possibilities for you to choose from. Some options are easier than others, and we will take that into account when graded. You'll be graded on both functionality and style, e.g. how elegant your code is.
We are more interested in the process than the results, so you will get some points even if you don't arrive at the results you want to, as long as you explain the problem you encountered along the way and how you tackled them.

Problem 4 (40 points): Improving Logistic regression

CHOOSE between Task #1 or task #2
Task #1: Vanilla logistic regression gets about~90% on the MNIST dataset, which is unacceptable. The dataset is basically solved and state of the art models reach accuracies above 99%. You can use whatever loss functions, optimizers, even models that you want, as long as your model is built in TensorFlow. Save your code in the file named ImproveLogReg.py. In the comments, explain what you decided to do, instruction on how to run your code, and report your results. Try to achieve 97%.

Task 2: Logistic regression on the notMNIST dataset

The machine learning community got a bit sick of seeing MNIST pop up everywhere so they created a similar dataset and literally named it notMNIST. Created by Yaroslav Bulatov, a research engineer previously at Google and now at OpenAI, notMNIST is designed to look like the classic MNIST dataset, but less 'clean' and extremely cute. The images are still 28x28 and the there are also 10 labels, representing letters 'A' to 'J'.

Since the format of the notMNIST dataset is not the same as the MNIST dataset, you can't just put the notMNIST dataset in place of the MNIST dataset for your model. David Flanagan is very kind to publish his [script to convert notMNIST to MNIST's format](). Even then, you still can't just call the function input_data from tensorflow.examples.tutorials.mnist to read in data for you. You'll have to read in data yourself, and you'll also have to take care of how to separate your train set from your test set.

Once you have the data ready, you can use the exact model you built for MNIST for notMNIST. Don't freak out if the accuracy is lower for notMNIST. notMNIST is supposed to be harder to train than MNIST.

Problem #5 (Extra Credit 10 points): Train a vanilla LSTM Network on MNIST, treating the digit rows as a time series of vectors. Try to achieve 97%.

Submission format: For problem 3-5, please submit a writeup that describes
1) Your model, includes equations, background references and the logic behind your choices. Also include in your write up any figures you generate.
2) Your Tensorflow code as a self-contained IPython notebook or .py file
3) Screen shots or other output of Tensorboard, including model and training error graphics.