

Handed out: 07/17/2019

Due by 11:59 PM EST on Tuesday, 07/23/2019

Problem 1. Represent the following expressions in TensorFlow

$$y = 2x^2 + 5$$
$$z = (3 * y - 4)/x$$

Calculate and print the value of y where x is a series of values starting with 1, incremented by 1 and ending at 9. Define variable x with a placeholder so that you could feed different values of x while calculating y . Use `Session.run()` method which is fed different x values through a `fetch_dict(ionary)`. Similarly, calculate and print values of z . Generate a TensorBoard graph of your calculations.

(15%)

Problem 2. Consider 100 points along the straight line in (x, y) plane represented by the linear equation $y = 0.3x + 0.2$. Distribute those points along the line uniformly in the interval between -2.0 and 3.0. To the y coordinate of each point add a random normally distributed value with standard deviation of 1 and mean 0. You have created an artificial set of random measurements. Create a shallow neural network with one layer which will be able to predict y value corresponding to any x value in the above interval. Implement and train the network using TensorFlow API. Report on the accuracy of your model. This is a rather trivial problem and you do not need neural networks to solve it. We are practicing TensorFlow API. This was problem #2 of your Problem set #1. Then we did it with Keras. Now, you are asked to implement this problem in TensorFlow.

(25%)

Problem 3. Using code in the attached file named `graph_and_summaries.py`, demonstrate complex TensorBoard Graphs and Summaries. Capture a few of resulting graphs and summaries. Demonstrate that nested graphs could be expanded and collapsed. Please fix the code if it needs fixing. Provide us with working code. Code is a bit old and you might have to replace older TF calls with the current versions. This code paints several TensorBoard graphs displayed in the notes for Lecture 7.

(10%)

Problem 4. During the lecture on Neural Machine Translation we discussed the content of Jupyter notebook named: `nmt_with_attention.ipynb`. The notebook is from TensorFlow tutorial site which implemented Seq2Seq model for translation between two languages using GRU Encoder-Decoder with Attention. All supporting files are uploaded to the folder of Lecture 8.

Select a data set from the site <http://www.manythings.org>. Our suggestion is to use languages with a large data set, i.e. number of pairs of sentences. Set the number of sentences, the maximum length of sentences and the size of vocabulary to values close to those mentioned in notes. Set the size of the training and testing datasets to the values

close to those mentioned in notes. Set the size of the context vector close to the one used in notes. If your network trains too slowly, reduce those parameters.

- a) Next, train the network and record the resulting BLUE scores.
- b) Use three sentences of length, 5, 8 and 10 and record their translations produced by both versions of your network.
- c) Finally, capture the attention heat maps for those 3 sentences. Examine the heat maps for the existence of any patterns. There might be no obvious patterns in those particular sentences.

(50%)

Please submit working Jupyter notebook and its PDF image. If you can not generate PDF, please use Chrome Print and then Save as PDF. Please comment every operation that