Homework 6 – Deep Neural Networks (CS/DS 541, Whitehill, Spring 2020)

1 Implementing a Neural Network with TensorFlow [25 points]

You may complete this task either individually or in teams up to 2 people. In this problem you will train a neural network to classify images from the Fashion MNIST dataset – in contrast to the previous assignment in which the data contained hand-written digits, here you will be recognizing different kinds of clothing. Instead of implementing forward- and backward-propagation yourself using numeric Python, you will rely on the highly optimized TensorFlow package, which implements symbolic differentiation and other nifty features. Your tasks are to do the following:

1. Install TensorFlow on your own machine (see `https://www.tensorflow.org/install/` for instructions). Either the CPU- or (if you have one) GPU-based version is fine for this assignment.

2. Follow the instructions in the following tutorial: `https://www.tensorflow.org/tutorials/keras/basic_classification`. Train a neural network to achieve a **test** accuracy of at least 85.0%. Feel free to vary the number of layers, units per layer, etc. Create a screenshot showing at least 5 gradient updates on the training set, along with a screenshot of your final accuracy on the test set.

3. Follow the instructions in the following Medium blogpost (by Margaret Maynard-Reid): `https://medium.com/tensorflow/hello-deep-learning-fashion-mnist-with-keras-50fcff8cd74a` to train a convolutional neural network (CNN). Train a neural network to achieve a **test** accuracy of at least 90.0%. Feel free to vary the number of convolutional and pooling layers as well as their associated hyperparameters (padding, stride, etc.). Create a screenshot showing at least 5 gradient updates on the training set, along with a screenshot of your final accuracy on the test set.

In addition to your Python code (`homework6_WPIUSERNAME1.py` or `homework6_WPIUSERNAME1_WPIUSERNAME2.py` for teams), create a PDF file (`homework6_WPIUSERNAME1.pdf` or `homework6_WPIUSERNAME1_WPIUSERNAME2.pdf` for teams) containing the screenshots described above. **Please submit both the PDF and Python files in a single Zip file.**

2 Competition: Face Age Estimation [Optional]

In this contest you will be competing with your classmates to win the Fabulous Prize #1. You may participate in the competition in teams of *any* size. However, the Fabulous Prize is fixed in magnitude, so keep that in mind.

Rules:

- You can use the face images and associated ages in `facesAndAges.zip` to train a neural network and to optimize its hyperparameters. How you partition the data is up to you.

- You are permitted to use any other *publicly available* dataset of face images and labels as long as you disclose where they came from.

- You are encouraged to explore different neural network architectures, hyperparameter settings, data augmentation, pre-training, etc.

- You are strongly suggested to use an off-the-shelf neural networks training framework such as TensorFlow or PyTorch.

- Your network will be evaluated on a test set that will be released on Wednesday, March 4 at 8am. These test images are randomly sampled from the same population of face images and labels as were the training data.

- Prior to receiving the test data, you will be required to *freeze* and submit your trained neural network and all associated weights that you will use to estimate the ages of the faces in the test set.
- You are *not* permitted to manually label any of the face images in the test set.
- After receiving the test data, you will have 5 hours to submit your network's estimates for the age of each test face. *In addition*, you must also submit your *estimate* for your guesses' mean-squared error relative to the ages according to ground-truth labels.
- The winner will be announced during class on Wednesday, March 4. The team with the lowest actual MSE wins.