CS4610/CS5335: Homework 5

Out: 4/10/19, Due: 4/18/19

Please turn in this homework to the TAs via email by the due date. Q1, Q2, and Q3 should be submitted in the form of a set of three files named Q1.m, Q2.m, and Q3.m. All this should be zipped up into a single file and emailed to me.

Have a look at the accompanying zip file. Stub files for Q1.m, Q2.m, and Q3.m are provided to you. You should implement each of these. Once implemented, you should be able to run "hw5(X)" in order to run code for question "X". hw5.m is given to you and should not need to be modified. The only thing you need to do is to insert code into the stub functions in Q1.m, Q2.m, and Q3.m. NOT ALL OF THE STUB FUNCTIONS IN Q1.m, Q2.m, and Q3.m NEED TO BE MODIFIED. Please see the code.

PA Q1: In this question, I have created a convolutional neural network that can be trained to classify MNIST digits. However, my current code runs rather slowly. Please change two lines in the definition of "layers" in Q1.m to speed up learning. When you're done, it should get to near 99% accuracy the the end of the second epoch. It might help to have a look at the Matlab docs on MNIST classification at `https://www.mathworks.com/help/nnet/examples/create-simple-deep-learning-network-for-classification.html`.

PA Q2: In this question, you must modify the network you designed in Q1.m to handle English letter classification instead of digit classification. There are 26 letters in the English alphabet. The letters have exactly the same image size as do the MNIST digits: 28x28 grayscale. Your answer to

this question should be a "Layers" structure analogous to what I provided in Q1.m.

PA Q3: In this question, we will speed up learning for letter classification by finetuning a network that was pretrained on the MNIST digit classification task. In pretraining/finetuning, we train a network on one task (in this case, the MNIST digit classification) and then copy most of the learned weights to a new network that is used to solve a new task (in this case, letter classification). Your answer to this question should actually *do* MNIST classification using your code from Q1 and then copy the appropriate weights to a new network. For more information on how to do this with Matlab, see the documentation at: `https://www.mathworks.com/help/nnet/examples/transfer-learning-using-alexnet.html`.