Lab Assignment 3
ECSE440L
Convolutional Neural Networks

Due: Feb. 16 (11:59pm), 2020

Submit an electronic copy of your assignment via LMS. No late submission! Be sure to include your name and student ID with your assignment. Please hand in your code and detailed answer to the questions.

For this question, you will experiment with fully connected neural networks and convolutional neural networks, using the Keras open source package. Keras is one of the simplest deep learning packages that serves as a wrapper on top of TensorFlow, CNTK and Theano. Preliminary steps:
_ Download and install Keras from https://keras.io/ . A CPU installation is sufficient for this assignment.
_ Click on "Getting Started" and read the "Guide to the Sequential Model".
_ Download the file cifar10_cnn.py from the example folder https://github.com/keras-team/keras/tree/master/examples.

Answer the following questions by modifying the code in cifar10_cnn.py.

a) Compare the accuracy of the convolutional neural network in the file cifar10_cnn.py on the cifar10 dataset to the accuracy of simple dense neural networks with 0, 1, 2, 3 and 4 hidden layers of 512 rectified linear units each.
Modify the code in cifar10 cnn.py to obtain simple dense neural networks with 0, 1, 2, 3 and 4 hidden layers of 512 rectified linear units (with a dropout rate of 0.5). Produce a graph that contains 6 curves (one for the convolutional neural net and one for each dense neural net of 0-4 hidden layers). The y-axis is the test (validation) accuracy and the x-axis are the number of epochs (# of passes through the training set). Produce curves for the first 10 epochs. Although 10 epochs are not sufficient to reach convergence, it is sufficient to see the trend. Explain the results (i.e., why some models perform better or worse than other models).

b) Compare the accuracy achieved by rectified linear units and sigmoid units in the convolutional neural network in cifar10 cnn.py. Modify the code in cifar10 cnn.py to use sigmoid units. Produce a graph that contains 2 curves (one for rectified linear units and another one for sigmoid units). The y-axis is the test (validation) accuracy and the x-axis is the number of epochs (# of passes through the training set). Produce curves for the first 10 epochs.

Although 10 epochs are not sufficient to reach convergence, it is sufficient to see the trend. Explain the results (i.e., why did one model perform better than the other model). No need to submit your code since the modifications are simple.

c) Compare the accuracy achieved with and without drop out as well as with and without data augmentation in the convolutional neural network in cifar10 cnn.py. Modify the code in cifar10 cnn.py to turn on and off dropout as well as data augmentation. Produce two graphs (one for

training accuracy and the other one for test accuracy) that each contain 4 curves (with and without dropout as well as with and without data augmentation). The y-axis
is the accuracy (i.e., train or test/validation accuracy) and the x- axis is the number of epochs (# of passes through the training set). Produce curves for as many epochs as you can up to 100 epochs. Explain the results (i.e., why did some models perform better or worse than other models and are the results consistent with the theory). No marks will be deducted for doing less than 100 epochs, however make sure to explain what you expect to see in the curves as the number of epochs reaches 100.