EECS 542 W17 Problems (2/3)

Due: 2/10/2017, Friday, 3:00PM

Jason J. Corso, jjcorso@umich.edu
Abhishek Venkataraman, abhven@umich.edu

Directions – *These are problems to help you ground the reading for the week. You should attempt them **individually** before class (submission due by class-time) and then you will work through them with your group in class on 2/10. These are to stimulate discussion and are not graded beyond your effort (full credit for some submission that shows you have worked on them). Expected time-load is two hours; you might not be able to complete the problem within this time. That is fine. Make an attempt to work these through to the best of your ability. After class and before the following Monday at 11:59am (afternoon), you need to write up a short reflection (max half-page) that describes how your individual implementation was different than that after the group work, and how your understanding of the material was changed based on the group work*

Dataset: MNIST Database[1]. Hand written digits

MNIST is a set of hand written digits. Each image is a 28x28 px grayscale image with exactly one digit. For this problem, you will be working only the digits 0 and 1.

Problem Definition:

- **Objective:** Implement a binary classifier using Artificial Neural Network. Given an image, classify it based on whether a '0' or '1' is contained in it.
- **Training:** Use the images on the train folder to teach your network. The files are already sorted as 0 or 1. Use the file location to get the label.
- **Testing/ Evaluation:** Classify the images on the test folder. The files are already sorted as 0 or 1. Use the file location as ground truth. Plot your classification vs ground truth.
- **Programming details:** You may implement this in any programming language[2] of your choice. Implement a fully connected layer and an activation function. You will have to train the network using the forward pass and back propagation (use a loss function similar to the one shown in class). You may not use any neural network library/package. However, you may use packages such as numpy, mathplot, etc.
- **Network Architecture:** You are free to choose what kind of network you want to use. However, do remember that you will be running this on your computer. Hence, you will might want to have a simple network(just activation and FC layer), that does not have too many parameters to learn.
- **Take ways:** (i) Design your own neural network. (ii) Understand how your gradient calculation and back-propagation works.

[1] Source: http://yann.lecun.com/exdb/mnist/. You may download the complete dataset from here

[2] However, we would prefer, C/C++, python or MATLAB since we can help you in case you get stuck.