**CEN-535/435 Artificial Intelligence**

**Homework 2**

**Instructions**

In this homework you will use Python to implement a simple artificial neural network and be able to train a simple convolution neural network on a hand written digit dataset called MNIST. We have posted the numpy based implementation code on Blackboard. To run it, you need to install some dependency libraries like numpy, scikit-image, matplotlib. To setup the environment, you should use pip with the following command.

pip install numpy scikit-image matplotlib scikit-learn

The homework should be done in Python (Python 3.X is recommended). Include a **documentation** (in .doc or .pdf format) that describes the complete steps of your solutions for each problem including: (1) The answer to any questions posed, (2) any results (include print outs or terminal outputs), and (3) the name of any Python module(s) and function(s) (Numpy, Matplotlib, etc.) you used. Discussion is allowed, but you must submit your own write-up and list your collaborators.

Zip all your documents with the **source code** (.py files, if there are self-implemented modules, organize them in different folders), **documentation**, and **data** (image files, text, etc.) into a file called **[Lastname\_FirstInitial\_Homework2.zip]** and upload it to the Blackboard prior to the due date.

Make sure your code works when you send it out. Please use as less third-party libraries as possible cause it is easy for instructors to reproduce your results. If you used the libraries that requires installation, please list the dependencies and its version in your ***README*** file or using pip to generate the dependency list like pip freeze > requirements.txt.

For late assignments, you will receive 10% off per day on any assignment handed in late up to a week. However, after a week on any given homework you will receive no credit for the assignment. So please start your assignment ASAP.

**Homework Description**

The source code *ANN\_LeNet\_MNIST\_demo.py* gives you three choices to build a neural network: *‘TwoLayerNet’, ‘ThreeLayerNet’, ‘LeNet5’*. You should be able to run the program by giving specific parameter to the program like:

python ANN\_LeNet\_MNIST\_demo.py -model TwoLayerNet -iter 10000 -opti SGD

**Data preparation**

You should first download the MNIST dataset (<http://yann.lecun.com/exdb/mnist/>) using this piece of code in MNIST\_util.py

download\_mnist()

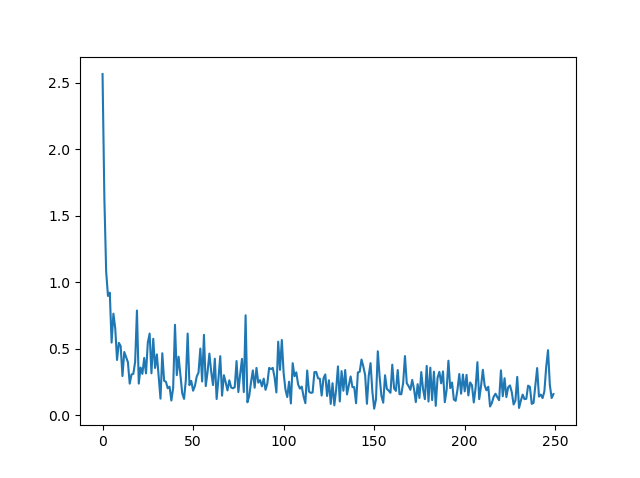
save\_mnist()

After this step, you should see **mnist.pkl** in your folder which is the training and testing set you will use in this homework. You can use get\_MNIST\_train\_viz\_pngs() and get\_MNIST\_test\_viz\_pngs() to visualize these hand written characters.

Artificial Neural Networks

In ANN.py, we have provided the code for model TwoLayerNet. Which is a two layer artificial neural network with two layers of fully connected layers and one ReLU activation layer. The *\_\_init\_\_* function defines all the layers that would be used in the model and initialize them if the weights are given. The *forward* function takes in input X (which is the hand written image in our task), and return the output of the neural network (which is a ten digits vector representing the probability of 10 digits given the input sample). The *get\_params* and *set\_params* return or set parameters of the neural network. Note that every FC layer is followed by an activation layer (ReLU in TwoLayerNet model) except the final FC layer.

In this homework, you should try to implement a three layer artificial neural network -ThreeLayerNet based one TwoLayerNet. Note that since you have three layers in the network, there should be three FC (fully connected) layers and two activation layers (use ReLU). The H2 in *\_\_init\_\_* function is the output dimension of the second FC layer and the input dimension of the third FC layer.

1. Train the TwoLayerNet with default parameter setting, set iteration=25000, try both optimizers ‘SGD’ and ‘SGDMomentum‘, report the training and testing accuracy. Plot the training loss over iterations like the below image. (5 points)

Training loss over iterations. X-axis represents the training iterations/100, y is the training loss.

1. Implement the ThreeLayerNet by yourself. Set H1=300, H2=100, and train the ThreeLayerNet model 25000 iterations. Report accuracy on both training and testing dataset and draw the loss figure. (10 points)
2. Train LeNet5 model 2000 iterations, report the accuracy on testing dataset. (5 points)