New Mexico State University
Klipsch School of Electrical Engineering

EE565 Pattern Recognition and Machine Learning
Fall 2016 – Project #5
Due: 5:00pm Thu. Oct. 27

Name: _____

Grade: _____

Project

The goal of this project is to gain familiarity with a Convolutional Neural Network (CNN) and use it to build a system that can recognize handwritten digits {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. This technology is also known as optical character recognition (OCR). The United States Postal Service (USPS) uses OCR at its processing facilities to sort and direct mail based on zip codes. For additional background information see

`http://en.wikipedia.org/wiki/Handwriting_recognition`

We will utilize the MNIST database of handwritten digits which consists of a training set of 60,000 examples and a test set of 10,000 examples. Information about this dataset, evaluation protocols, and classifier results can be found at [1]. MNIST raw image data has also been transformed into a MAT-file format and can be found at [2]. Finally, there are many tutorials on this "classic" project [3].

If you a developing in MATLAB, you may wish to use one of several libraries for CNNs or something you find on your own [4–6]. If you are developing in Python, you may wish to use one of several libraries for CNNs [7–9] or something you find on your own.

The project consists of developing a code which implements a training stage and test stage. The **training stage** reads in the training inputs $\{\mathbf{x}_n\}$ and associated 1-of-K coded target vectors $\{\mathbf{t}_n\}$ which are fed to a CNN in order to train the system via back propagation. The **test stage** reads in the test input $\mathbf{x}'$, forward propagates it through the CNN, and obtains an output vector $\mathbf{y}$. The maximum element in $\mathbf{y}$ decides the class or in this case, the digit. The decisions are used to determine the confusion matrix and overall classification error rate.

Your grade will not entirely be based on accuracy but this will, of course, be very important. Most important, however, is your design approach, code, simulation, and *reporting*.

Report

Please submit a hardcopy report detailing your handwritten digit recognition system including commentary, tables, and plots. Please `mailto:pdeleon@nmsu.edu` a zip file containing your code for this project and resulting CNN model for Problem 2. Do not include any MNIST data. It goes without saying that all code should run "out-of-the-box" (no hard-coded paths).

Notes

Students are encouraged to discuss detailed, technical aspects with each other and Prof. De Leon. However, students must write all other required codes on an *individual* basis.

1 Preliminary

a (10 points) Recreate Fig. 5.3 in the textbook using a simple (not convolutional), two-layer neural network (NN). Due to random initialization of the network weights, your values for the outputs

of the hidden units may not be the same as those in the plots.

b (5 points) Print an example of each digit (0-9) in the MNIST training data as shown in Fig. 1 to convince yourself the data are images of handwritten digits.

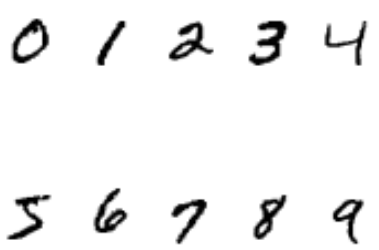

Figure 1: First example of each MNIST training digit

2 Handwritten Digit Recognition

(85 points) Based on the complete set of 60,000 training images and 10,000 test images, provide the confusion matrix as well as overall classification error rate for your NN-based handwritten digit recognition system. The i, j-th element of the confusion matrix should be the percentage of times digit i was classified as digit j.

Students should be prepared to re-run their codes during office hours if submitted error rates are less than 3%. Points will be assigned as follows and will assign points as follows.

Table 1: Point assignments for error rates in handwritten digit recognition code.

$< 3\%$	85	13-20%	63
3-7%	78	$> 20\%$	55
7-13%	70		

References

[1] (2016). [Online]. Available: http://yann.lecun.com/exdb/mnist/

[2] (2016). [Online]. Available: http://cs.nyu.edu/~roweis/data/mnist_all.mat

[3] (2016). [Online]. Available: http://neuralnetworksanddeeplearning.com/chap6.html

[4] (2016). [Online]. Available: https://www.mathworks.com/help/nnet/convolutional-neural-networks.html

[5] (2016). [Online]. Available: http://www.vlfeat.org/matconvnet/

[6] (2016). [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/ExerciseConvolutionalNeuralNetwork/

[7] (2016). [Online]. Available: http://www.pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/

[8] (2016). [Online]. Available: https://www.tensorflow.org/versions/r0.11/tutorials/deep_cnn/index.html/

[9] (2016). [Online]. Available: http://deeplearning.net/tutorial/lenet.html