MSML 603: Project #2 Description **Fall 2024**

Submission Instructions: Please submit all the materials, including codes, in a single file (a single tar file is preferred). Along with the codes, you are required to submit a short report that summarizes (i) the results that you obtained, which include (a) average testing accuracy from 5 runs for each network, and (ii) any lessons that you learned while working on the architecture design and parameter selection. In the codes, add all the necessary comments so that the grader can follow what you are doing. Otherwise, if the grader does not understand your codes, you may unnecessarily lose points. Finally, please include a `README.txt` file explaining how to run your code.

For this project, you will use (a) a feedforward neural network with fully connected layers and (b) a convolutional neural network to design classifiers using `MNIST` dataset. The data for the project is provided in four separate files in the folder `Files/PROJECT/Project2`.

- `train-images.idx3-ubyte` contains 60,000 training images of handwritten digits. Each sample is a 28×28 pixel image.
- `train-labels.idx1-ubyte` has the labels for the 60,000 training images (a number in $\{0, 1, \ldots, 9\}$ for each image).
- `t10k-images.idx3-ubyte` contains 10,000 images for testing.
- `t10k-labels.idx1-ubyte` provides the labels for the 10,000 images for testing.

For the project you can use any programming language and built-in packages to set up a neural network and train it using the provided training data. However, the neural network trained by your code should achieve testing accuracy of 0.95 with high probability. Note that each time you train a neural network, you will likely obtain a different neural network as the end of the training.

1. The feedforward neural network with fully connected layers should have at least two hidden layers. You can determine the number of nodes in each hidden layers and the activation functions.
2. The convolutional neural network should include at least two convolutional layers and two fully connected layers. The size of the filter and the use of pooling or stride is optional but encouraged. You can design your own architecture or use the architecture discussed in [1].

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.