**Exercise: Neural Network Computation**

**Overview:**

In this exercise, you will implement the backpropagation algorithm for neural networks and apply it to the task of hand-written digit recognition.

**List of Files for this assignment:**

ex4.py - script that steps you through the exercise

data1.mat - Training set of hand-written digits

weights.mat - Neural network parameters for exercise 4

helperFunctions.py - script that contains functions performing various tasks. The functions are:

* randInitializeWeights: Function for randomly initializing weights
* debugInitializeWeights: Function for initializing weights to fixed values. Only to be used for debugging
* computeNumericalGradient: Numerically compute gradients
* checkNNGradients: Function to help check your gradients
* displayData: Function to help visualize the dataset

ex4modules.py - the script that needs to be filled in by you for this assignment. It contains the following functions:

* sigmoidGradient: Computes the gradient of the sigmoid function
* nnCostFunction: Computes the cost value and gradient vector of the neural network.
* predict: Neural network prediction function

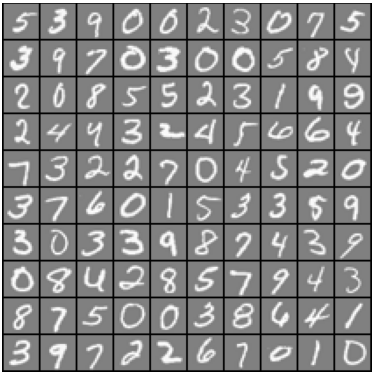
**What you should do:**

Note that the three functions declared in ex4modules.py are not yet implemented. Your task is to implement the three functions by filling in “YOUR CODE HERE” sections. After you have finished filling in your code, activate Miniconda, change directory to where your ex4.py is located, then type in following command and press Enter:

|  |
| --- |
| python ex4.py |

This will run the ex4.py Python script. During the execution, you will see output text results in console and graphic results in a separate window. If your implementation is correct, the graphic results will be similar to what is shown on the “Sample Results” section of this instructions sheet.

**Sample Results:**

****