Homework 3

Problem 1.
(a) The zip file for HW3 contains the PYTHON files
`Run_Michael_Nielsen_MNIST_program_Network2.py`
`network2.py`
`MNIST_Loader.py`
Save the output to a `.txt` file.

In the above `Run` file note the command `net.save('MNIST_CrossEntropy.json').` Look at the definition of `save` in the `network2.py` which shows that this command saves the network's size, weights, biases, and cost in the file `MNIST_CrossEntropy.json`.

Also in the above `Run` file note that the `validation_data` (not `test_data`) is being used to check the classification accuracy after each epoch.

(b) Restart the kernel and then run `Run_Network2_test_data_2.py`.
Save the output to a `.txt` file.

Note this program loads the test data, then it loads `network2` with the **already trained** parameters by the command `net = network2.load('MNIST_CrossEntropy.json').` Finally it checks the classification accuracy of the network on the test data by the command `net.accuracy(test_data, convert=False))`. Look at the definition of `accuracy` in the `network2.py` file.

The point here is to show how to save the network parameters to a `json` file and then later (after restarting the kernel) be able to use the *trained* network to check its classification accuracy on test data.

Hand in the .txt files from parts (a) and (b).

Problem 2.
(a) Use the same files as in Problem 1. Keep $\eta = 0.1$, $\lambda = 5$, and a batch size of 10. By varying the number of nodes in the hidden layer(s), the number of layers, and the number of epochs, what is the best accuracy on the test data you can find. Save the output from the run that gives the best accuracy to a `.txt` file.

(b) Restart the kernel and then run `Run_Network2_test_data_2.py`. Save the output to a `.txt` file.

Hand in the outputs from parts (a) and (b).