Programming exercise: Feed-forward nets forward propagation with MNIST data

In this exercise you will work with classifying MNIST images of handwritten digits. The image data can be found in week6data.mat in Matlab-format. Python can read matlab files using the scipy.io functions loadmat. This data consists of the images X, a 5000x400 matrix with 5000 training images with true labels given in y. Each image is originally 20x20 pixels, but converted into vectors of length 400.

The labels y contains labels 1-10, where 10 represent '0', and the other classes represent the respective digits.

Use the script week6.py as the main program. It will load the data, and visualize 100 random training images.

In this exercise you will use a pretrained net to classify X and compute the classification accuracy and the cost function. The weights can be found in 'week6weights.mat' and represent a network with 1 hidden layer with 25 nodes, and 10 output nodes for the classes 1-10.

Use the sigmoid activation function in this exercise.

1. Predict the class labels by the function nnetpredcit(Theta1, Theta2, X)
   Use a one-vs-all classifier for each of the 10 classes.
   Implemented correctly, you should see an accuracy of 0.9752 (97.5% correct)

   To see which images are correctly classified, classify one image at the time and display the true and predicted class labels.

2. Implement the function nnCostFunction(Theta1, Theta2, input_layer_size, hidden_layer_size, numLabels, X, y, lval)
   Remember to append 1-s to X and the a-vector of the hidden layer.
   X is the entire training data set, y the true labels, and lval the regularization parameter lambda. Use sigmoid activations.
   With lval=0 (no regularization), the cost should be about 0.2876.
   Remember to avoid regularizing the bias terms.
   With lval=1, the cost should be about 0.3837.