Multilayer networks trained with backpropagation

Train a two layer network (one input layer, two hidden layers, one output layer) to classify hand written digits (the famous MNIST dataset). Use the matlab/python code and the dataset that are available on the website. The parameters are explained in the .m file. If the computer is slow, it is possible to reduce the size of the dataset.

- Choose the size of the two hidden layers, the learning rate and the number of epochs to get a cross-validated performance of at least 96% (4% error). A performance of 98% is not too difficult to achieve. Notice that matlab gives the performance in terms of cross-entropy. Write a few lines of code to compute the performance in terms of number of patterns correctly classified. The initial parameters in the code on the website work but are not optimal.

- Once the network works, plot the training and the test error as a function of the number of epochs.

- Variability from run to run: Run the program 10 times, and plot the training and test errors for all the runs on the same plot. Determine the minimum and the maximum number of iterations needed to achieve a 96% performance. The variability that will be observed is one of obstacles to study deep networks.

- Choose the size of the two hidden layers to minimize the training error and to maximize the test error (overfitting situation). Plot the errors as a function of the number of epochs.