Deep Learning - HW1

Tidhar Suchard, Danielle Ben Shlomo

Model architecture description and training procedure

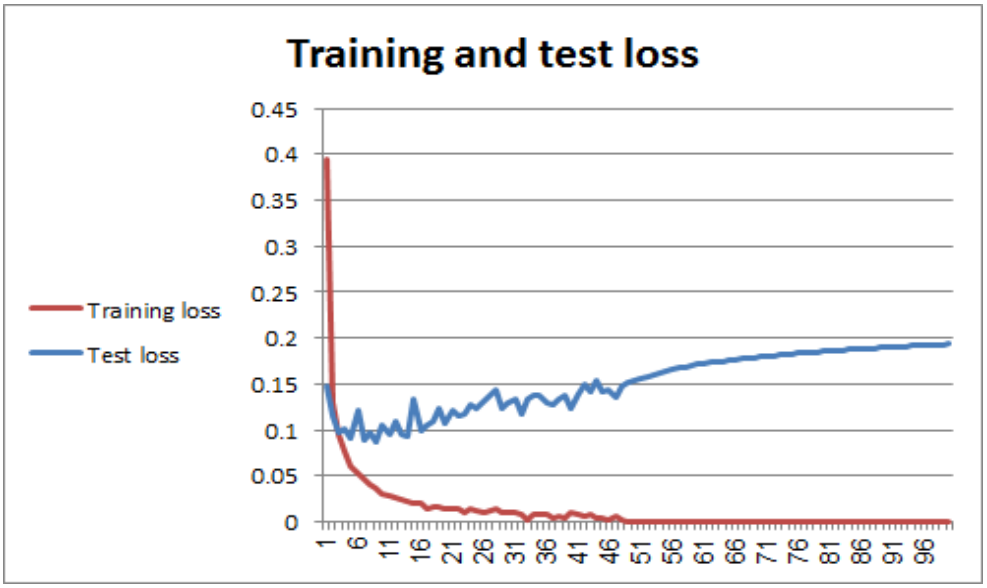
The model we built contains four linear layer with 64 neurons each, as well as a ReLU layer after each of the linear layers. After that there is a linear layers with output size 10, corresponding to the number of different classifications of the digits. At last, we added a LogSoftMax layer, required by the criterion we chose to use: ClassNLLCriterion (Negative Log Likelihood), a criterion used for classifications problems.
When constructing the model architecture, we conducted several experiments with different layers sizes (all linear). At first we tried using layers with size growing exponentially. Since the number of parameters used in the model was limited to 65,000, we could only create a very small number of layers to meet the requirement. As a result, we made attempts to deepen the network by using a lot more layers, but with smaller layer sizes, for example: many layers of 32 neurons. Eventually we found that the best results out of the models we tested were given by the model described above. This means we chose a slightly deeper network than our initial approach, and yet still keeping a significant number of neurons for each layer.

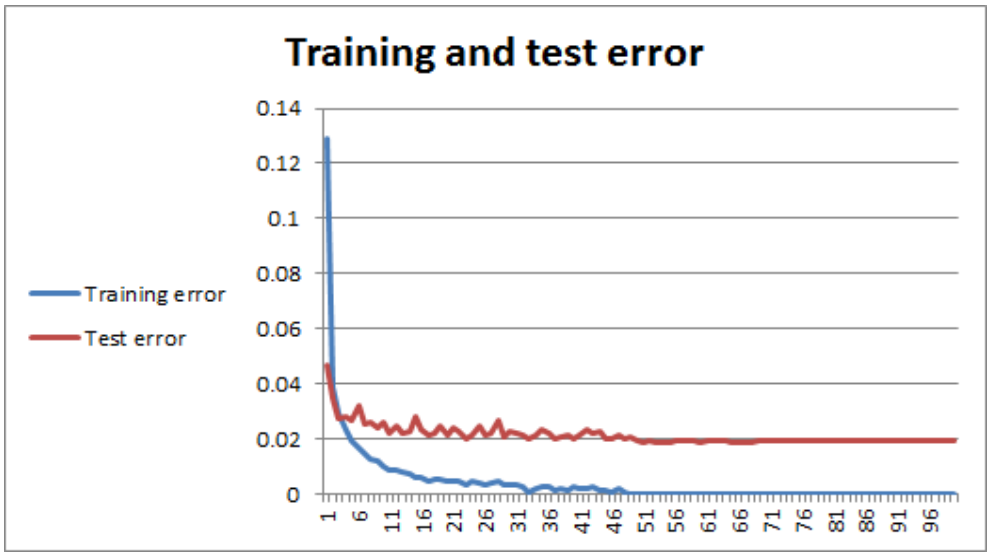
As well as trying to optimize the layer architecture, as part of preparing the data for training we normalized it around 0 to try and further improve our result.

We used 51 epochs to train the network (The minimal error for the test set was given using this number of epochs), and the size of each batch is 32.

Our model contains 63,370 parameters.

Results:





Summary and conclusions:

As we can see, both training loss and training error decrease as the numbers of epochs increase.
We could expect to see the test error increase after it reaches its optimal value as a result of over fitting, contrary to the actual results showed in the graph. A possible reason for that is the number of epochs we used. We can expect to see the function increases if we use more epochs, though it should be mentioned that we attempted to use a larger number of epochs and surprisingly didn't still notice significant rise in the test error.