Neural Networks Lab Manual 4/4

From the Single Unit to MNIST — A Process-Over-Product Sequence

---

Lab 4 — MNIST: Learning at Scale (33/40 pts)

Idea

Real data introduces what toy problems cannot: generalization, and the discipline required to measure it honestly.

Architecture

Use ReLU in the hidden layers and a softmax output layer. Set-up your network so that you can have an arbitrary number of nodes per hidden layer. If you want, you may set it up to have more than one hidden layer. Ensure you have a procedure that tells the accuracy of the network, i.e. the number of errors. Your network will make a prediction of which digit it saw, if it is the right digit, great, if not it is an error.

Tasks

A. Write code that reads in the training data and test data. Here is the MNIST training data. An explanation of it can be found here. Here are some hints for curating the data.

B. Here is a set procedures that visualize the data after you read it in properly. Here is the PixelGrid.java file that goes with it. Paste the visualization of the first three images from the training set in the lab manual.

C. Experiment with the following values of the hyper-parameters.

1. Size of hidden layer: less than 50, 50-500, larger than 500.
2. Initial random weights are all positive, all negative and both, positive and negative.
3. Learning rates: 0.3, 0.5, 0.7.
4. Additional curation of the data. Right now, your data probably consists of a grey-scale image. Try removing some of the precision of the data.

Based on your experiments, what are the values of the hyperparameters that ensure the fewest number of epochs to successfully train the network? By successful, we mean at most 3% wrong on the test data. The training data accuracy should be 99%. [15 pts] At first experiment to get a good set-up that results in the desired accuracy with the minimum number of epochs. [10 pts] Then conduct experiments for each of (1)-(4) to determine their impact on training. Briefly explain the behavior of the network for each of (1) – (4).

D. [3 pts] Once your network works well, train it deliberately too long. Record the training and testing accuracy in a table or diagram. Explain the behavior.

E. [5 pts] Feed the trained network random noise, or an image that is not a digit at all. Predict the output first. Observe that softmax still assigns a confident class — because it must produce a distribution over exactly ten options, there is no “none of the above.” Try to be creative. Produce and attach a visualization of your input. Report your observations. Do this for three interesting inputs.

Extra credit — Label shuffling

[7 pts] Using your best set of hyper-parameters, randomly permute the training labels so that image and label are unrelated, then train the same network with no other changes. Predict first: can a network fit data with

no learnable structure? Run about three experiments. Feel free to vary the number of epochs. Record the accuracy of the training and test sets in a table. Explain what happens.