AI Programming: Assignment 5

Eirik Vågeskar

November 24, 2015

This report describes five artificial neural networks (ANNs) constructed using the Theano framework. These networks are used to classify hand-written numbers from the MNIST data set. The networks will be compared by their performance, error rates, and running times.

1 The Networks

The networks vary in size from 1 to 3 layers. They all use the sigmoid function for activation in all layers and the binary cross entropy function for measuring the errors.

The networks' hidden layer topologies are as follows, starting with the layer closest to the input layer:

1. 100
2. 200
3. 100, 100
4. 200, 200
5. 200, 150, 100

2 Comparison

The following comparisons are based on averages of twenty fresh runs of each ANN type. Each run consisted of training for 20 epochs followed by a test of the network on the MNIST testing and training sets. Note: For the (200, 200) network, the testing was paused in the middle of one of the runs. This pause resulted in a high running time. This data point is excluded from the calculation of the average running time.

The networks all used an adaptive learning rate, $lr = \max(\frac{1}{\text{epoch number}}, lr_{min})$, where $lr_{min} = 0.1$ and epoch number starting at 1.

The raw data is attached as five text files, each with a title corresponding to its network's topology.

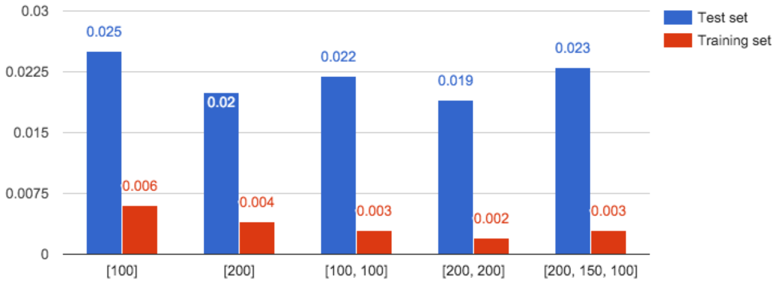

Figure 1: Performance of ANNs on test and training sets

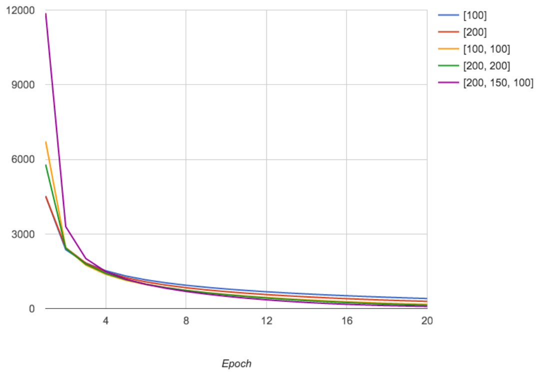

Figure 2: Error rates during training

2.1 Average Performance on Training and Test Sets

As one can see in Fig. 1, when networks have an equal number of hidden layers, a larger number of nodes in each layer seems to be the deciding factor in how the networks perform on the test sets.

The deciding factor in performance on the training set seems to be the number of layers. This may stem from overspecialization in the network.

2.2 Average errors during training of ANNs

As can be seen in Fig. 2, the errors of the output layer converge at about the same rates. The networks start at quite a high error rate, and end up at a relatively low error rate. The more complex networks have a lower error rate at the end of training, which may be a sign of overspecialization.

2.3 Running time

A comparison of the average time for training and testing each ANN can be seen in Fig. 3. As one would expect, networks with larger weight matrices have a longer running time.

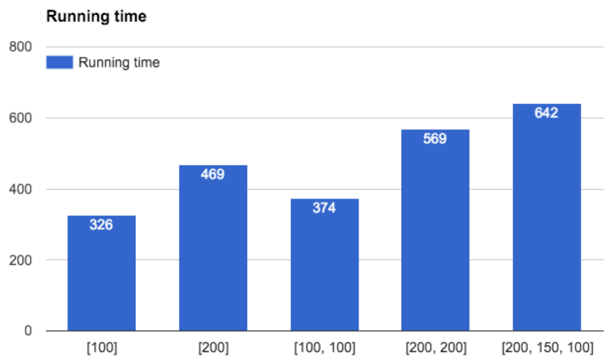

Figure 3: Average training and testing time in seconds for the ANNs

This also seems to pay off, as the larger running time results in a lower error rate on the test set, as can be seen in Fig. 1.

2.4 The Selected Network

The network with the topology (200, 200) has been selected to be run on the demo day. Its performance on the test data is comparable to that of the network with a (200) topology, and it makes only half the amount of mistakes on its training data. This justifies its slightly increased running time.

3 Discussion

The selected networks vary only in their topology. This was done for one theoretical and one practical reason. The theoretical reason was that varying only one variable demonstrates more clearly the effect of that variable. The second reason was that the network was already able to perform to the satisfaction of the evaluation criteria, and performance was significantly worsened when using the hyperbolic tangent as a transfer function.

The combination of sigmoid activation and binary cross entropy error was selected because the sigmoid function yielded good results during the test trials and that the binary cross entropy function was recommended for classification tasks in the material found during research. (Mean squared error was recommended for tasks involving regression and estimation of continuous values.)

Two examples of material that recommends the binary cross entropy function: Blog post by James D. McCaffrey on Cross-Entropy vs Mean Squared or Classification Error, `https://archive.is/GjvVJ`, and a discussion of the matter on Reddit's Machine Learning page, `https://archive.is/b7ThD`.