Homework 2: Neural Networks

Naomi Dickerson

February 1, 2016

Description

A two-layer neural network in Python, run on the UCI letter recognition dataset. Hyper-parameters must be set in the code, and sections may be uncommented to produce an optional confusion matrix as output.

Experiment 1

As shown in figure 1 below, there is no evidence that the neural network has overfit to the training data. There is no significant difference between the accuracy of the training set and that of the test set, and both are frequently the same and always within two percentage points of one another.

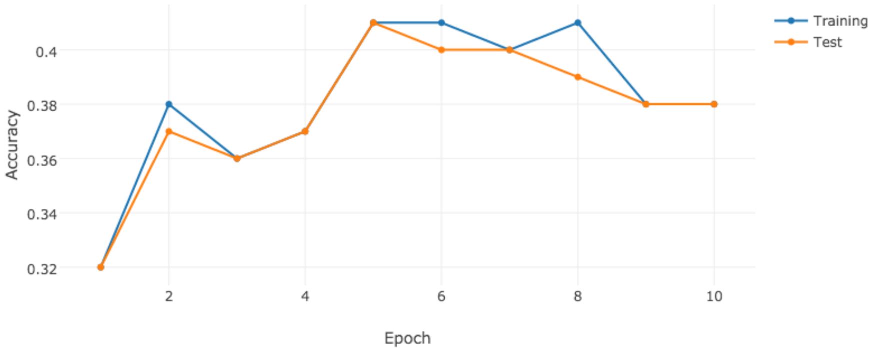

Figure 1: Training and test accuracy with $\eta = 0.3$, $\alpha = 0.3$, n = 4

Experiment 2

Varying η had a significant effect on the accuracy of the neural network. Figure 2 shows that the network with a low learning rate was still climbing in accuracy after 10 epochs, but even then had not yet attained the initial accuracy of the network with a high learning rate. The network with high η levels out almost immediately in accuracy, and of note is the fact that the highest accuracy achieved is still 5% lower than that in experiment 1.

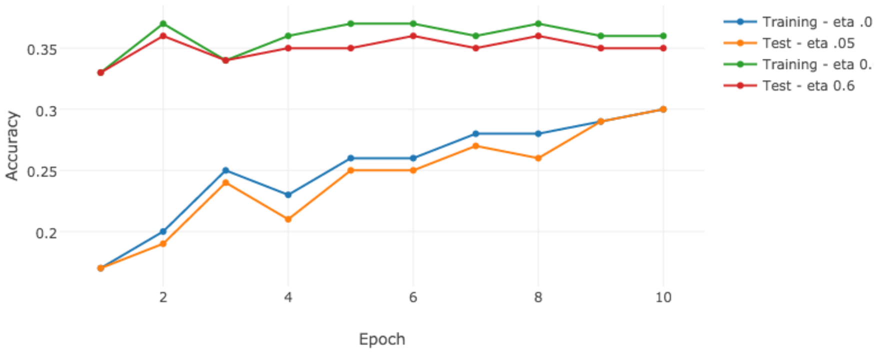

Figure 2: Training and test accuracy with low $\eta = 0.05$, high $\eta = 0.6$, $\alpha = 0.3$, n = 4

Experiment 3

The network with low momentum experienced slightly more pronounced fluctuations in accuracy, but overall varying α did not have a notable effect. At least, within this limited testing framework it would be hard to say that figure 3 is remarkably different. The accuracy wavers between 35-40% after the third epoch. As in the previous two tests, the training and test accuracy were extremely close, within 2 percentage points.

Experiment 4

Training and test accuracy continued to be nearly identical, so accuracies on the training set have been omitted in figure 4. Varying the number of hidden units had the most significant effect on accuracy, so I chose to run tests on a wider range to see if accuracy continued to increase. Even up to 128 hidden units the accuracy improved overall, although the amount of improvement decreased

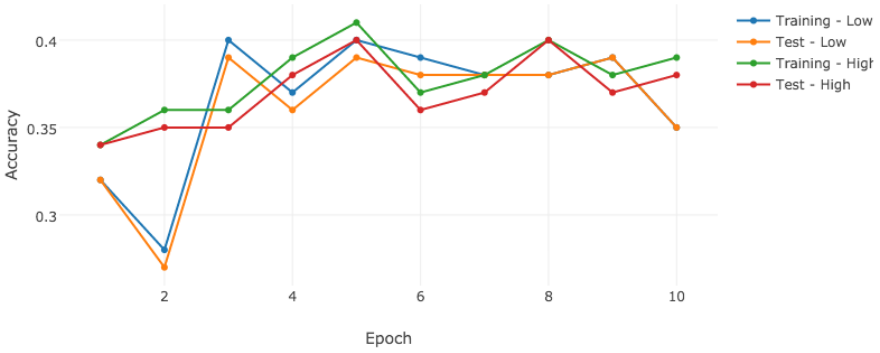

Figure 3: Training and test accuracy with low $\alpha = 0.05$, high $\alpha = 0.6$, $\eta = 0.3$, n = 4

as hidden units increased, and the performance of networks with 64 and 128 hidden units is comparable. The fact that the training and test set accuracies remained nearly identical even at 128 hidden units suggests that this data is 'iid'.

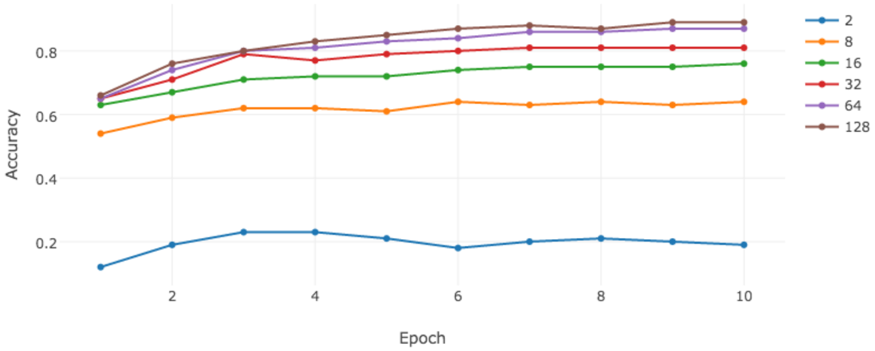

Figure 4: Test accuracy with n = 2-128, $\eta = 0.3$, $\alpha = 0.3$

Notes

While not a part of this assignment, I was curious about the results of my neural network and printed confusion matrices for some of the data. With the initial settings $\eta = 0.3$, $\alpha = 0.3$, n = 4, I noticed that there were several classifiers the network never chose. While it could reliably and accurately classify an 'A', it never classified, either correctly or incorrectly, any of the following:

E,F,G,H,I,N,O,Q,X,Y

One possible hypothesis, supported by the results of figure 4, is that 4 hidden units is simply not enough to fit the data. With 8 hidden units, there were still three classifiers **F,O,X** that were never chosen. Doubling the number of hidden units to 16 finally produced a neural net that was capable of classifying all letters.