CS838 Lab 2

Sek Cheong

March 12, 2017

1 Introduction

The purpose of this experiment was to design a neural network with single hidden layer to predict protein secondary structures. The data set we used in this experiment was the Molecular Biology (Protein Secondary Structure) Data Set obtained from the UC Irvine Machine Learning Repository. We used convolution with the window of 17 amino for the input train the network with the middle amino secondary structure. The overall accuracy of our network was about 62% obtained in less than 5 epochs on average.

2 The Data Set

The data set from UC Irvine was composed of two files one for training and one for testing. We combine these two files into a single file. The combined file was then split into training set, tuning set, and test set. The test set was a collection of every 5th example in combined file. The tune set was a collection of every 6th example in the combined file. The remaining examples in the combined file were used as the training set.

3 The Experiment

We used all the examples in the training set for each epoch in training of the network. The early stop criteria was set on the difference of accuracy on the tuning set of previous epoch and the current epoch. When the difference is greater or equal than 0.001 we stop the training.

$$\epsilon = accuracy(tune, epoch - 1) - accuracy(tune, epoch) >= 0.001 \qquad (1)$$

We tested the network with various number of hidden units and learning rate. We also tested the network unregularized and regularized with learning rate, momentum, and weight decay. Following were our findings.

3.1 Unregularized

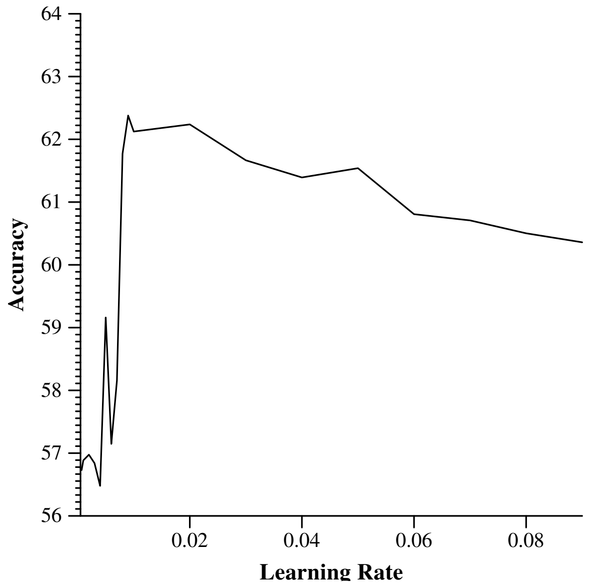
We used the following setting for our network for this experiment:

$$HU = 9, \eta = 0.01, \epsilon = 0.001, \alpha = 0, \lambda = 0, MaxEpoch = 100$$

Amount the experiments the unregularized network (with other setting being equal) gave as the lowest accuracy. We were only able to obtain on average 56.7% accuracy.

3.2 Learning Rate

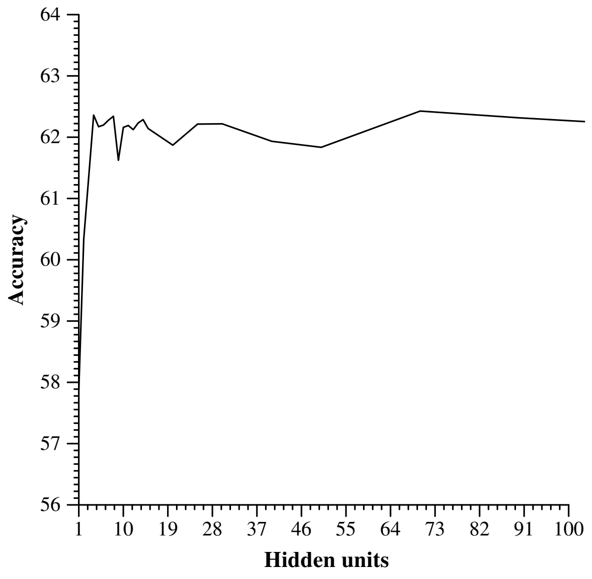
The learning rate η contribute greatly to the overall accuracy of the network prediction. We experimented η from 0.00050 to 0.09. The following graph shows the η vs. accuracy:



To make the comparison meaningful, we used the same setting as in 3.1, we only change the value of η for each run. As you can see from the graph, setting η too small or too large could adversely affect network accuracy. In out case the optimal η was somewhere between 0.008 and 0.009.

3.3 Number of Hidden Units

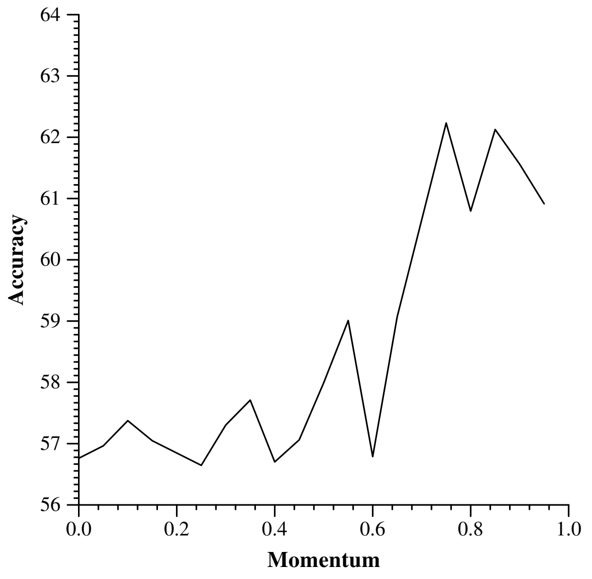
The number of hidden units also plays a significant role in the network accuracy. If the network have to few hidden units the network will not have the desired capacity to be trained to predict the give sample distribution. If the network has too many hidden units overfitting is likely to occur.



In out case the optimal number of hidden units was between 7 and 10 units. When the number of hidden units increase the network accuracy plateaued and steadily dropping lower.

3.4 Momentum

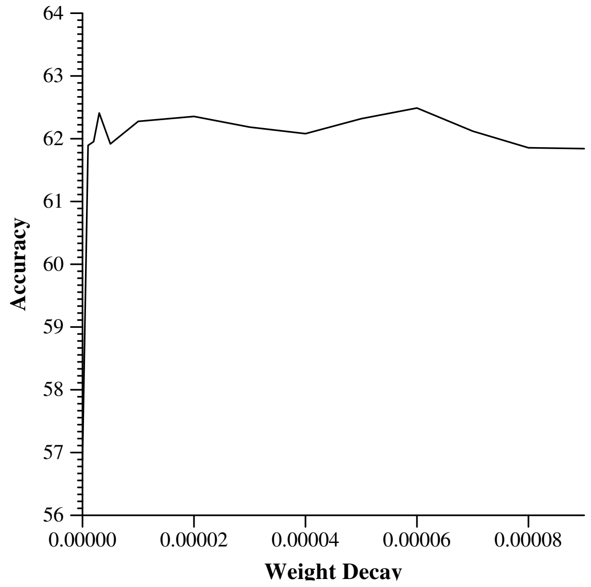
The momentum α also has a significant affect on the accuracy of the network. The appropriate momentum range is $0 < \alpha < 1$. The empirical result shows that larger momentum with smaller η gives yields pretty good result.



In out experiment we set the momentum somewhere between 0.75 to 0.95 yield the best result.

3.5 Weight Decay

The weight decay λ improves the accuracy to some degree. The result suggested that once λ reached 0.00006 the accuracy was in steady decline.



The weight decay was supposed to improve the accuracy by regularize the weight complexity of the network. When combined with the momentum α it did improve accuracy but not by much, however, the training time was noticeably reduced.

4 Conclusion

Our neural network performed pretty well and could achieve accuracy around 62% consistently. The number of hidden units could significant affect the performance of the network. Too few hidden units, the network accuracy would suffer and too many hidden units would not improve network accuracy or even reduce the accuracy. The training time is in polynomial time proportion to the number of hidden units. Therefore, it is important to choose the right amount of hidden units to get the best balanced result. Our best setting is as following:

$$HU = 9, \eta = 0.009, \epsilon = 0.001, \alpha = 0.90, \lambda = 0.0006, MaxEpoch = 150$$

A network with properly tuned learning rate, momentum, and weight decay could significantly out perform networks without any regularization. The weight decay did not seem to be very useful in out experiment.