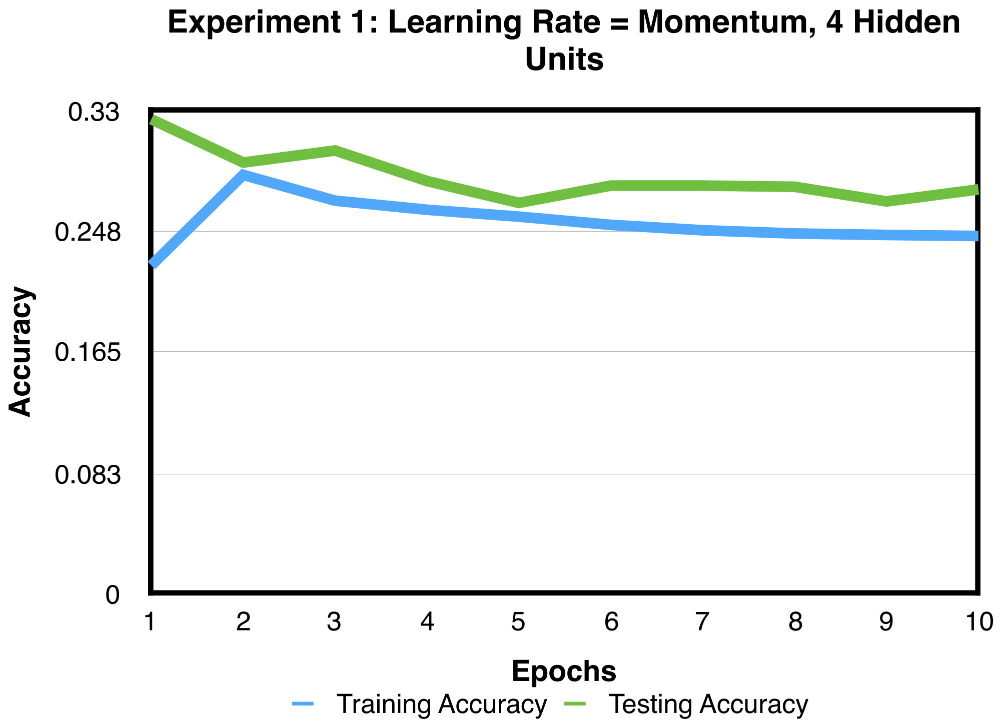
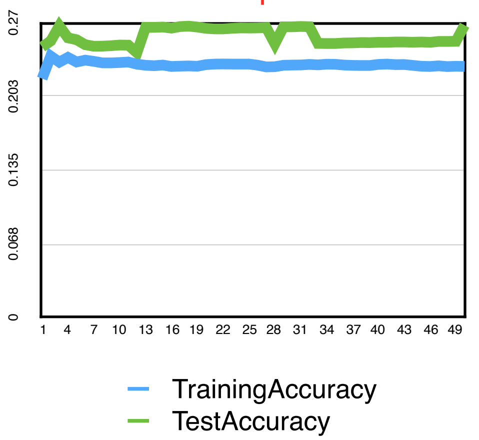
Tyler Sorg
CS445: Machine Learning
Homework 2: Multilayer Neural Networks



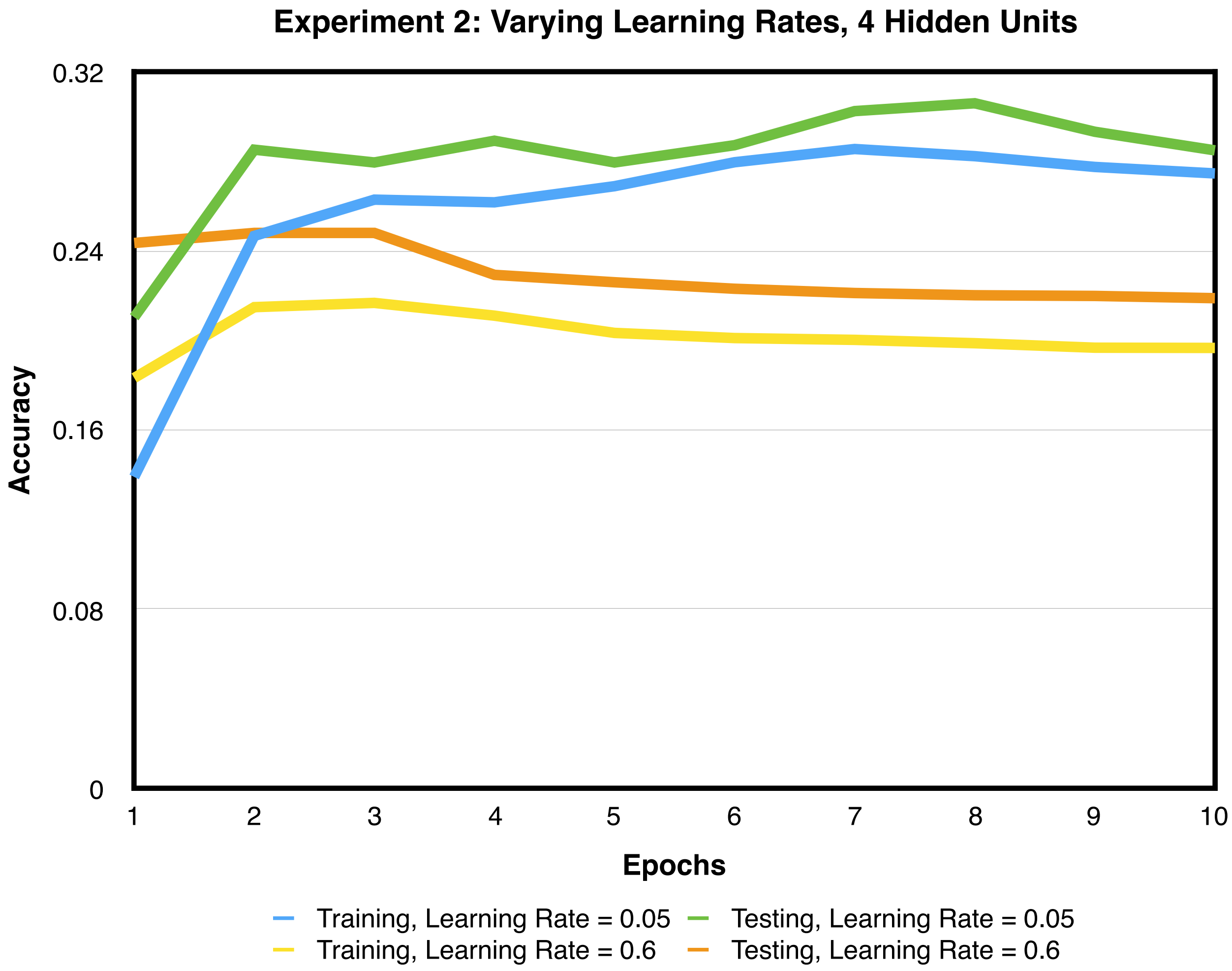
More epochs



Prior to experimenting:
The network was trained on ~10,000 examples and tested on ~10,000 different examples. The training method was back-propagation with stochastic gradient descent, including a momentum term to avoid oscillations when optimizing the weight vectors. Before training, the training and test examples were standardized by scaling the values such that they have zero mean and unit variance along each of the 16 features. Note: The test examples were scaled using the statistics of the training set. The weights between layers of the network were initialized between -0.25 and 0.25.

Description:
The weights between units in each layer were changed after each training example. After each epoch (every training and test example went through the network), the network's accuracy was calculated on the training set and the test set. This process repeated for a maximum of five epochs in most experiments.

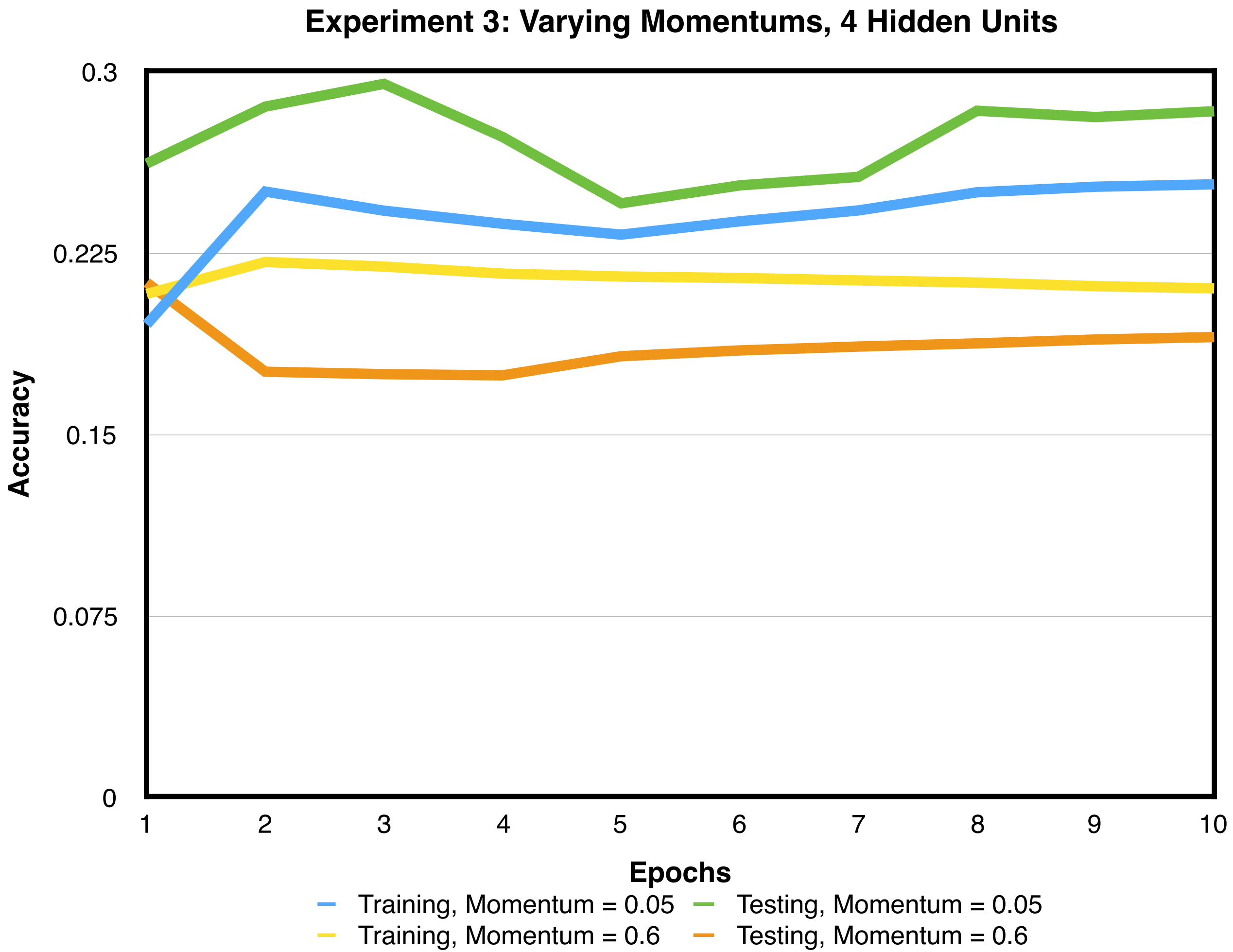
Question: *Is there evidence that your network has overfit to the training data? If so, what is that evidence?*
There does not seem to be evidence of overfitting here because the accuracy after each epoch was higher on the test set. The opposite phenomenon would indicate overfitting, which might happen with a larger hidden layer, or if the network was trained over many more epochs.
Keeping the hyperparameters the same as in experiment 1, but running the network for 50 epochs, still did not lead to apparent overfitting.



Description:
Experiment 1 was repeated twice more, but the learning rate was varied. The **blue** and **green** lines correspond to training the network with a **learning rate of 0.05**. The network was trained a second time with a **high learning rate of 0.6**, with training and test accuracies plotted in **yellow** and **orange**, respectively.

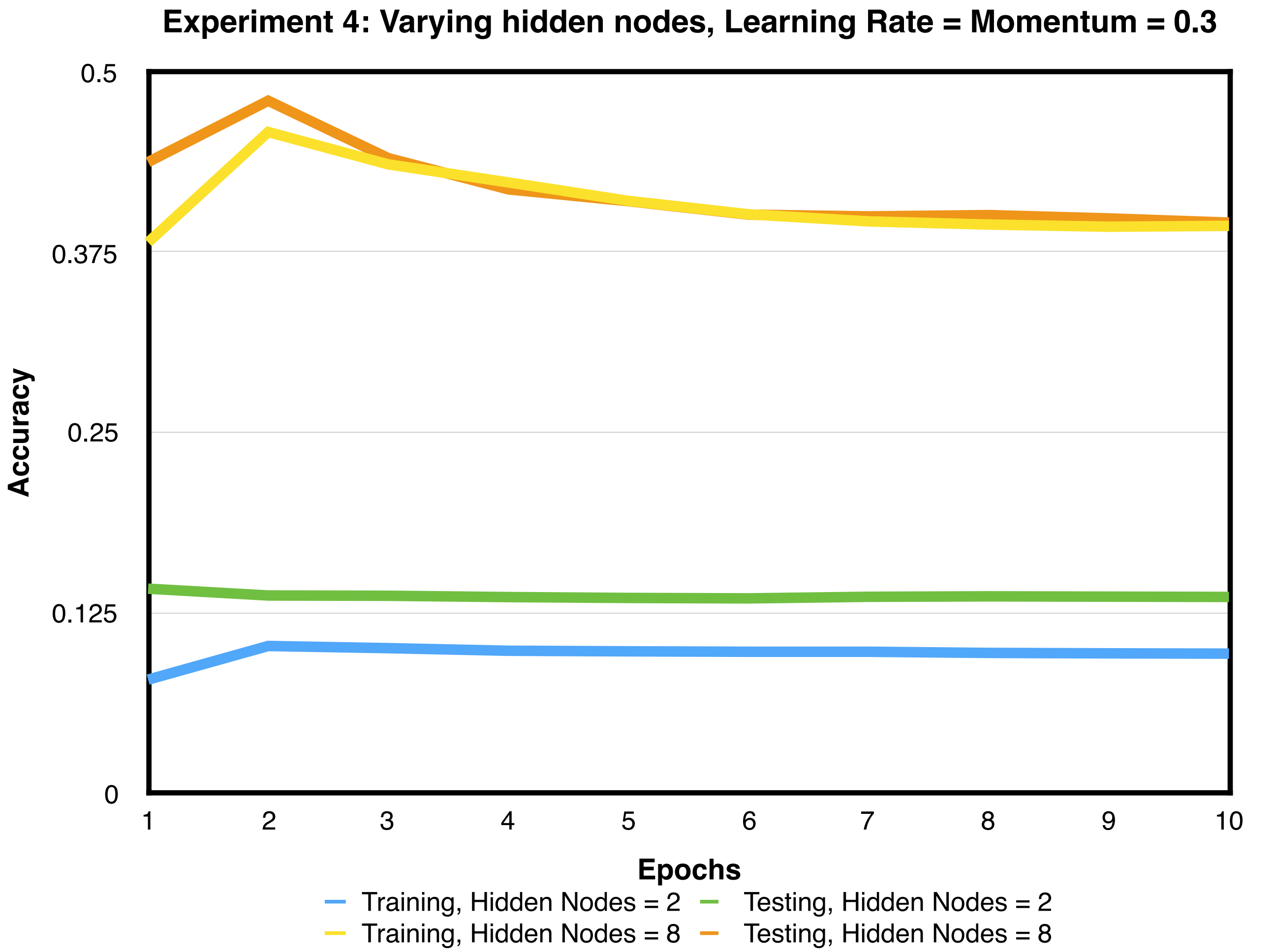
Question: *How does changing the learning rate change your results?*
Although the runtime was not recorded, the higher learning rate coupled with a momentum term was supposed to increase efficiency while not sacrificing accuracy. Low learning rate means that more adjustments have to be made for the optimization to converge, but it is less susceptible to oscillations.

In the experiment, the higher learning rate leads to higher accuracies in the beginning, but as more epochs elapsed, the accuracy decreased while the accuracies of the lower learning rate network increased. Since a large number of epochs leads to overfitting to the training data, perhaps this experiment's results support the use of higher learning rates and momentum?



Description:
The experiments were repeated again twice more, holding the number of hidden nodes and the learning rates constant. This time, the momentum coefficient, α, varied. One trial with **α = 0.05**, and the other with **α = 0.6.** The first trial's accuracies and the second trial's accuracies were plotted in the same color conventions as before.

Question: *How does changing the momentum change your results?*
Changing the momentum had varying effects between trials. In the second trial, it appeared to have caused overfitting, but that phenomenon was not present in the first trial. This was the first experiment where the network's accuracy was noticably better on the training set than the test set (not between trials).


Experiment 4: Varying hidden nodes, Learning Rate = Momentum = 0.3

Description:

In this experiment, all the hyperparameters were held constant except for varying sizes of the hidden layer.

Questions: *How does changing the number of hidden units change your results?*

The accuracy of the network seems mostly dependent on the size of the hidden layer. When the number of hidden units is small, the target function cannot be approximated well. However, when the number of units is high, the network tends to overfit to the training set. The second phenomenon did not present itself in this experiment because eight hidden units is still relatively small.

The accuracies increased to around 65% when the number of hidden units approached 40. Experiments conducted by classmates yielded ~90% training accuracies with 100-200 hidden units.