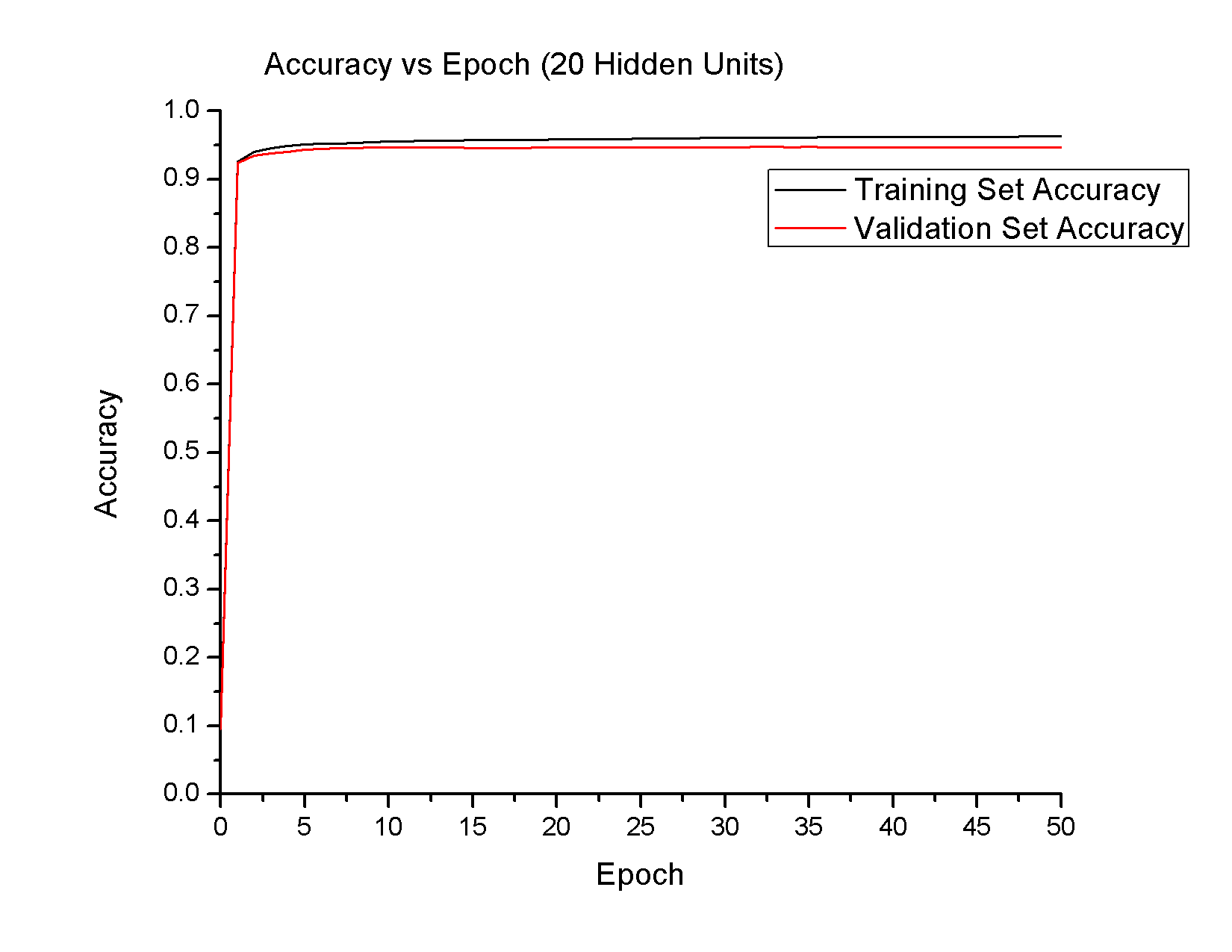
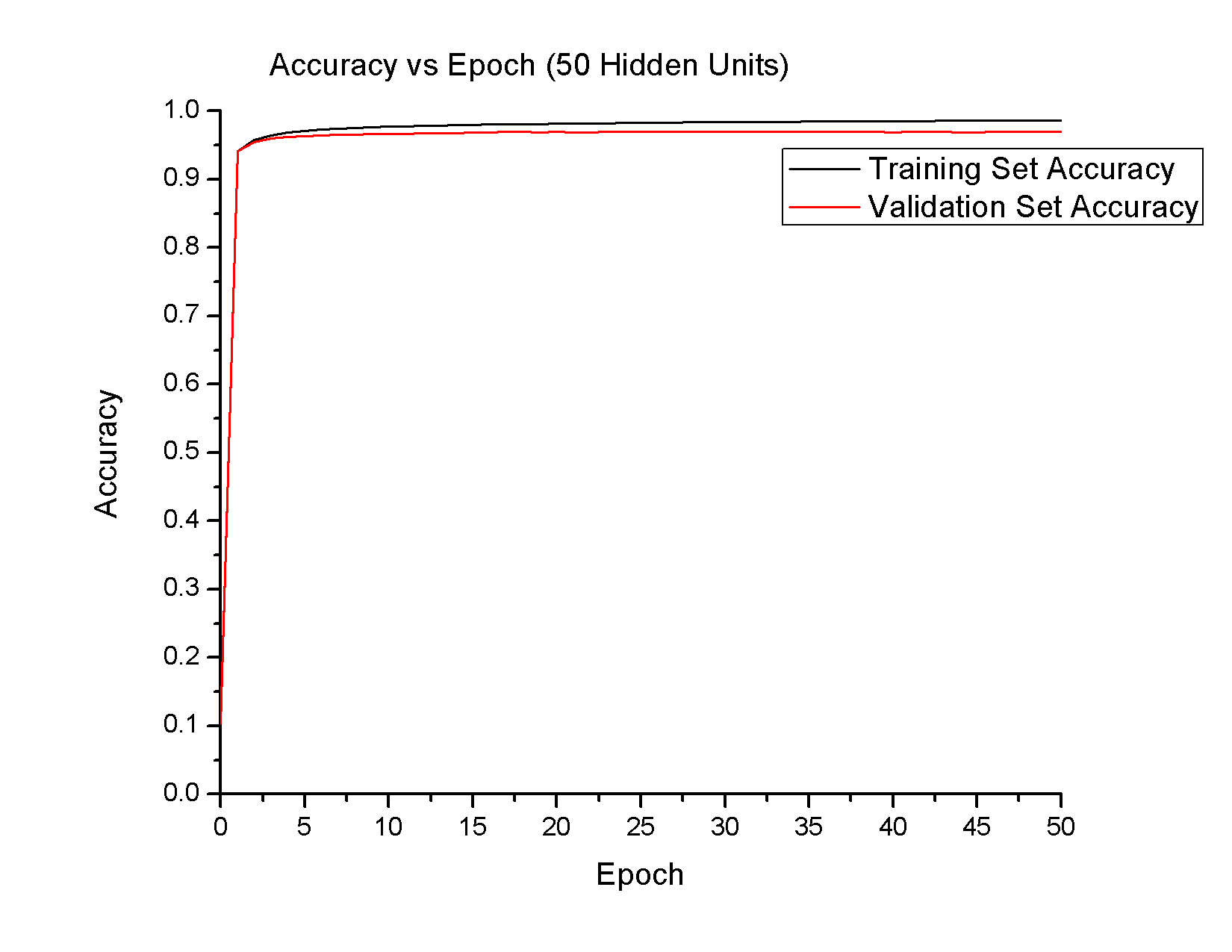
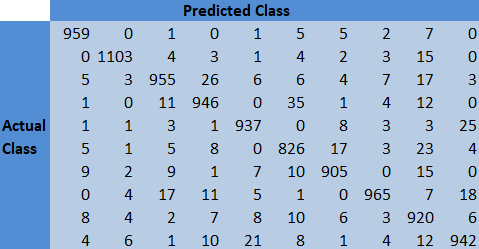
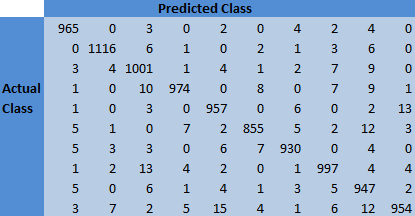
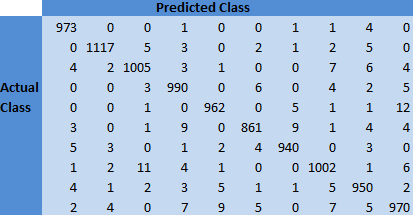
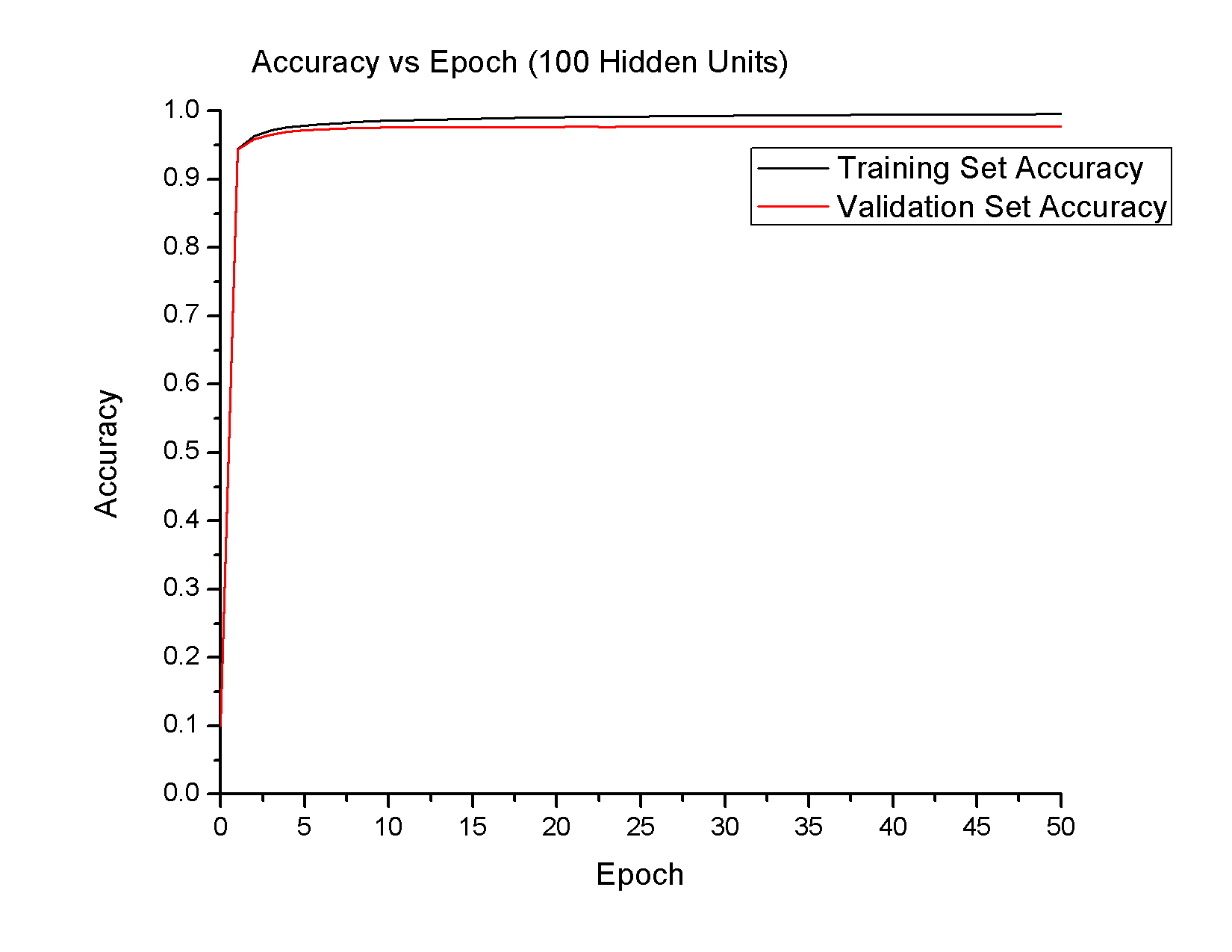
Austen Hsiao ID:985647212

CS545 Assignment 2

**Experiment 1: Vary the number of hidden units**

The numbers of hidden units were varied using 20, 50, and 100 units in the hidden layer. The accuracies for each number of hidden units are given below with its confusion matrix to the right.

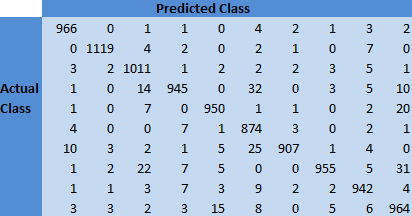
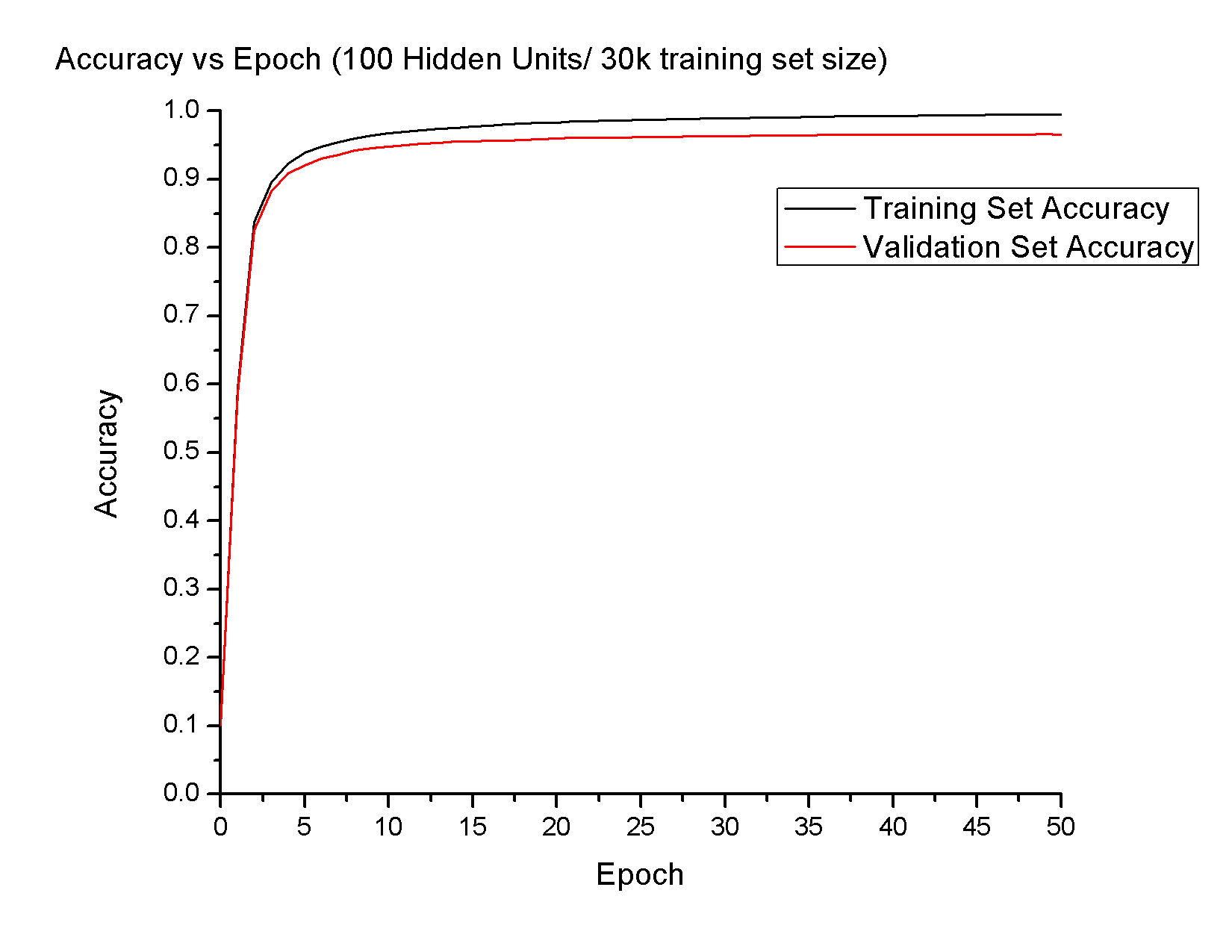


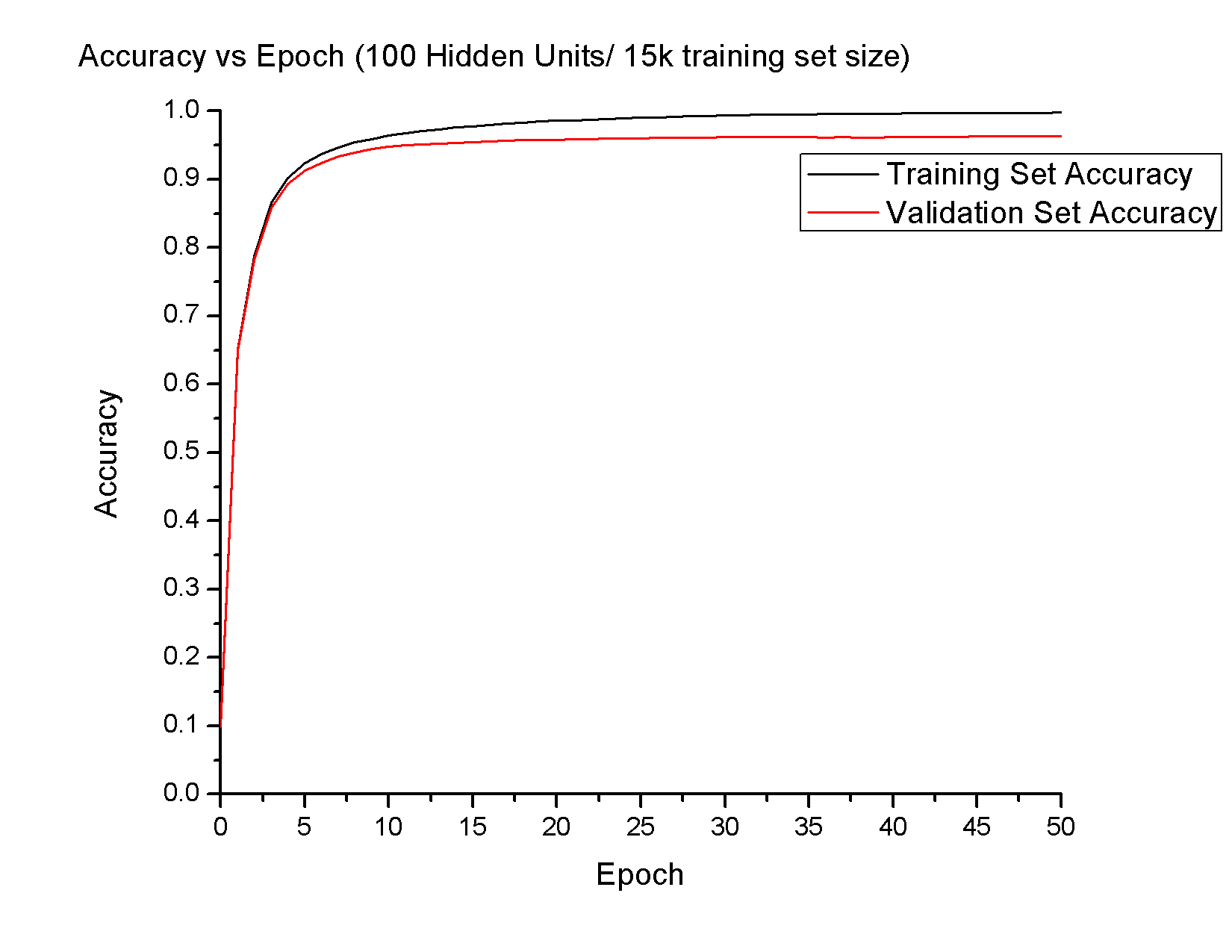
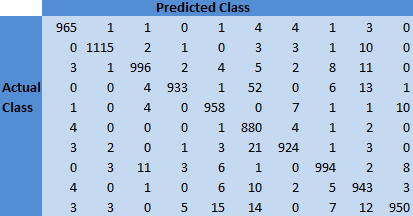


1. An increase in the number of hidden units increased the final accuracy. The accuracy of the validation set in the network with 20, 50, and 100 hidden units resulted in an accuracy of 94.6, 96.9, and 97.7 percent, respectively.
2. **In this assignment, for all experiments, I define the point of convergence as when the next iteration provides an accuracy within +/- 0.0001 of the first accuracy**. The network with 20 hidden units converged within 34 epochs. With 50 hidden units, the data converged within 26 epochs. With 100 hidden units, the data converged within 22 epochs. The most notable difference in all three trials was after the first epoch—which increased accuracy from ~10% to ~95%. It seems as though increasing the number of hidden units caused faster learning.
3. There are no clear signs of overfitting when looking at the graph. However, since the network with 100 hidden units converged so much quicker than the other 2 networks, this could be a subtle sign of overfitting. Faster convergence means that the network has quickly learned our training set. The implication is that it may have learned too quickly to be able to generalize the data trend. In general, I would expect to see divergence between training and test data and/or oscillations as a clear sign of overfitting.
4. Compared to the results from assignment 1, the accuracies achieved in this assignment were much higher (~10 percentage points) and with less signs of overfitting. In assignment 1, we had some signs of overfitting— oscillations as well as divergence between training and validation sets were present. Adding a single hidden layer has produced a better model to predict hand-written digits.

**Experiment 2: Vary the number of training examples**

The training set was reduced to ¼ and ½ of the original size (60,000) and two networks were trained with 100 hidden units. Training data was ensured to be balanced. Accuracy and confusion matrices are given below.



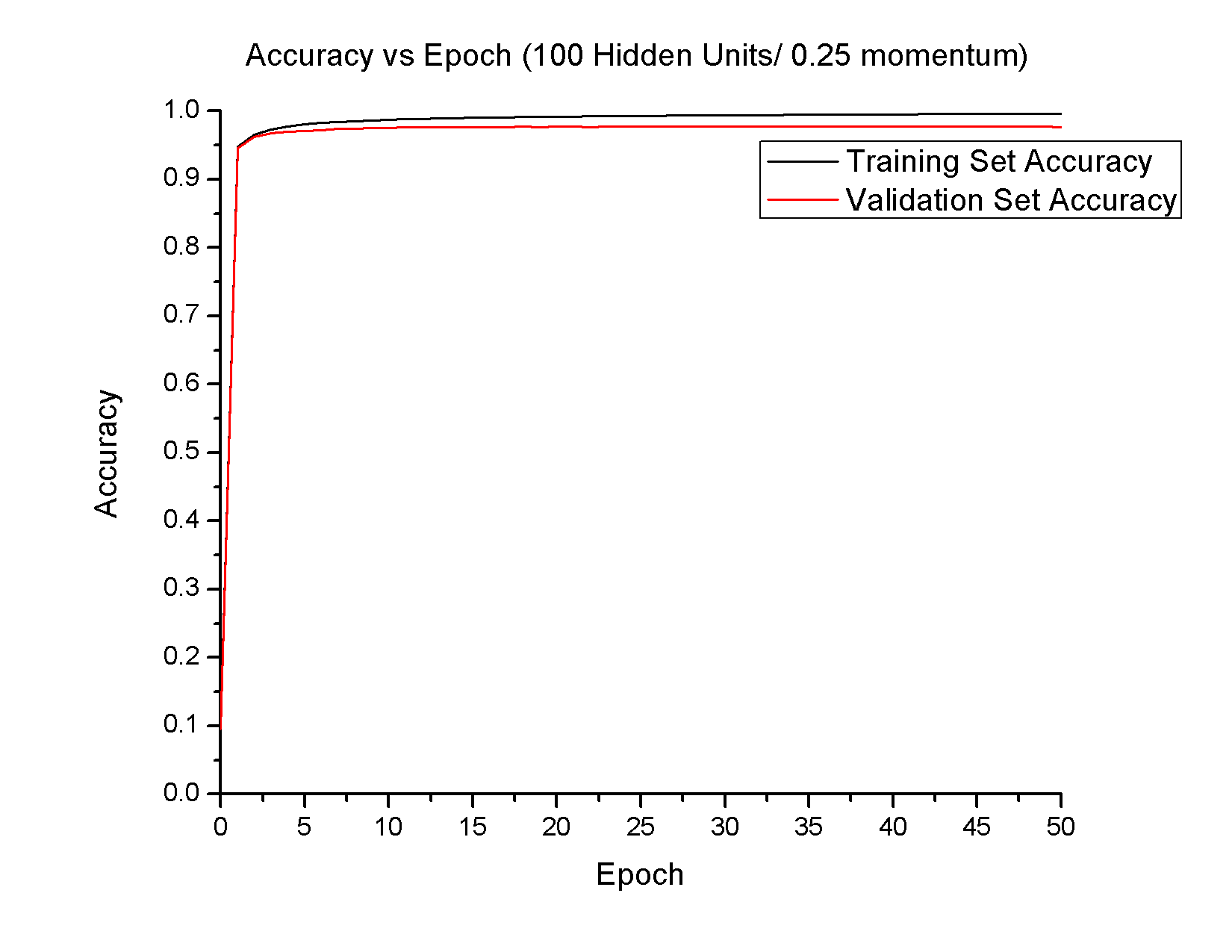


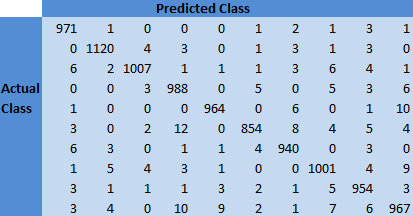
1. The accuracy for the 15k training set and 30k training sets were 96.3 and 96.6 percent, respectively. This difference is not very large, considering the runtime for each epoch for the 30k training set took approximately double the time of the 15k set.
2. It took 28 epochs for the 15k sized set to converge and 33 epochs for the larger, 35k, set. This is logical, as the smaller set has less data.
3. As in the first experiment, I saw no evidence that the network overfit the training data. Since the time to converge increased, this is also a further sign that we would see less overfitting.

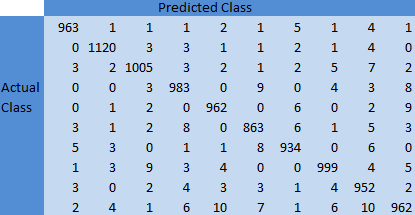
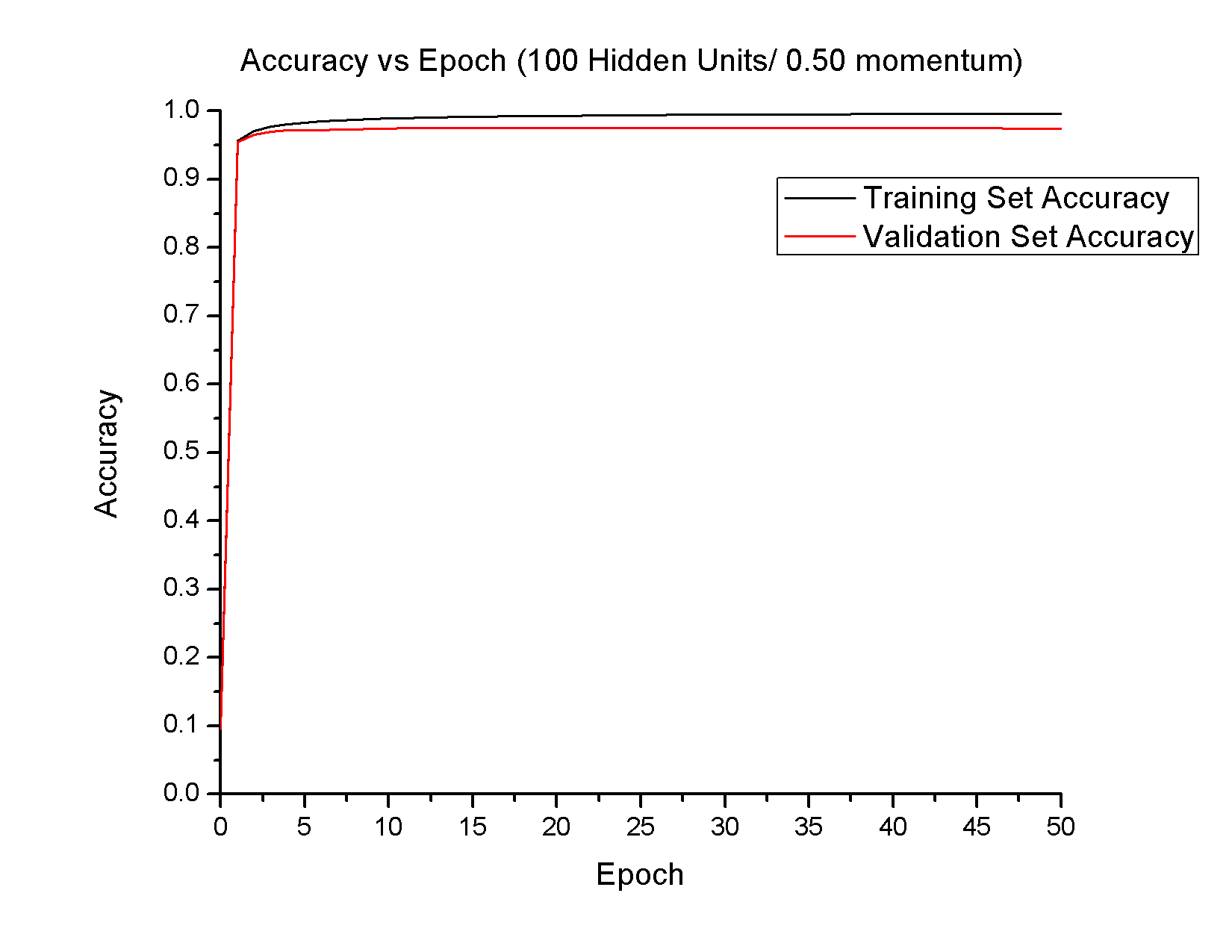
The most notable difference between using the smaller training sets and the full training set was in the run time. With my code, it took approximately 180 seconds for each epoch with the full set, 60 seconds with the 30k set, and 30 seconds for the 15k set. Between the two smaller training sets, the results of the 30k training set suggest better modeling of the data.

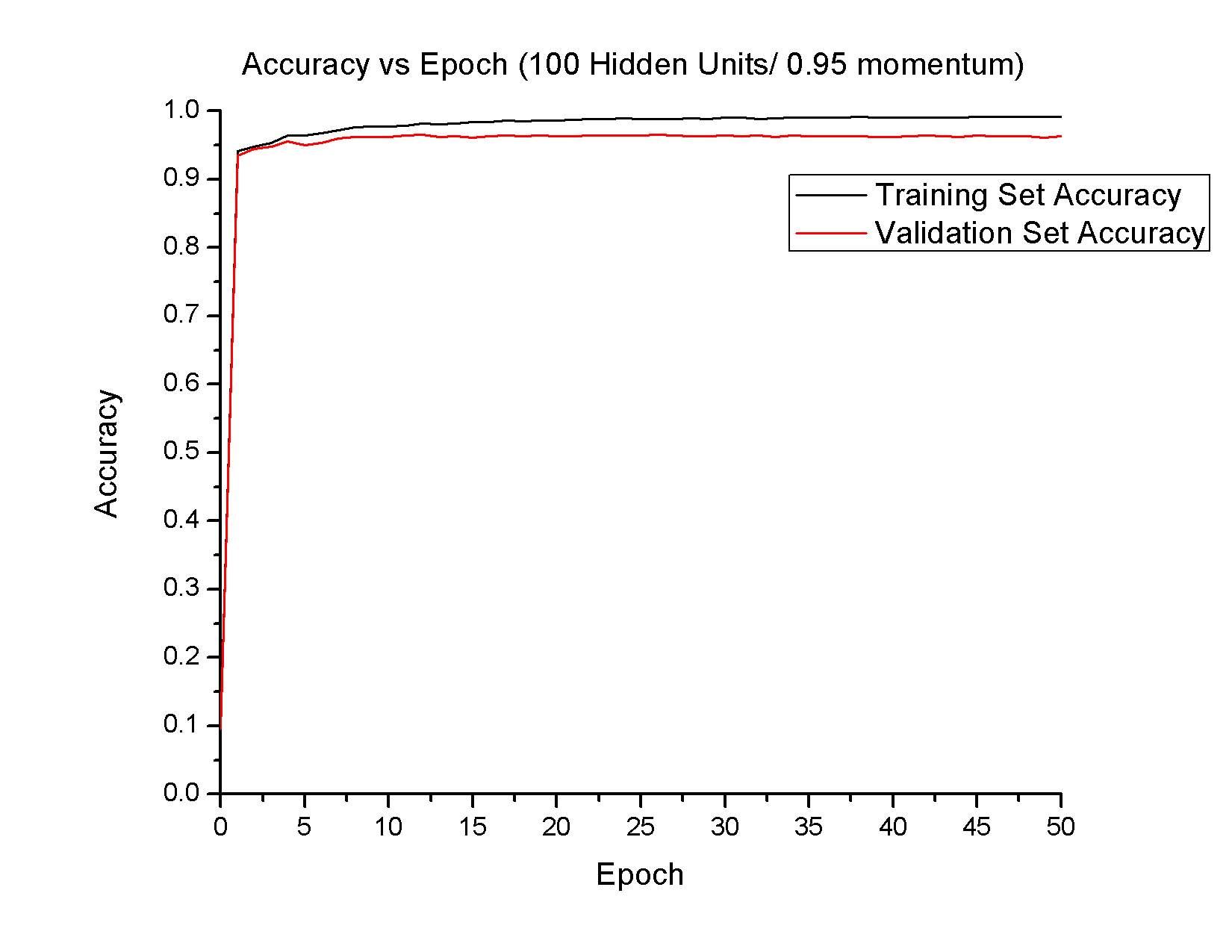
**Experiment 3: Vary the momentum value**

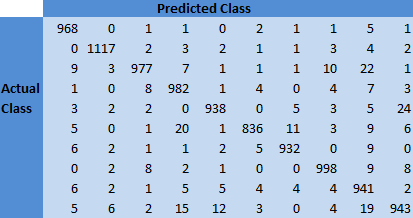
Trials were conducted with 100 hidden units and a momentum value of 0.25, 0.5, and 0.95. The momentum is added to the Δw term when training and can help reduce the number of epochs to converge. The difficulty in using a non-zero momentum value is that the Δw values must be stored. Accuracies and confusion matrices for each momentum value are given below.









1. The accuracy was not affected by different momentum values. All three trials had final accuracies of 96.58% with the test data.
2. The trial with a momentum of 0.25 converged within 24 epochs, 0.5 converged within 19 epochs, and 0.95 converged within 30 epochs. In general, it seems that increasing the momentum value decreased the amount of epochs required for convergence.
3. With a momentum value of 0.95, there are clear oscillations in the test data. This is a tell-tale sign of overfitting and may be caused by the training algorithm changing values too quickly. The result is a well fit training set but poorly fit test set. The convergence within 30 epochs may be attributed to randomness, as the accuracies continue to oscillate well after this point. There is also a greater delta between the final training and validation accuracies when compared to that of the graphs with momentum values of 0.025 and 0.5, which can be seen visually on the graph.