

STAT 775 Homework #5 - April 5, 2016

Timothy Sweet and Luke Fraser

This report describes the method and results of two classifiers used to classify prostate data in the prostate cancer dataset

(<http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/prostate.info.txt>). A neural network is used to classify the data

The rest of this report is structured as follows. Section 1 describes the neural network approach, results, and evaluation. Section 2 lists the source code and instructions for executing the code.

1 - Neural Network Approach, Results, and Evaluation

The neural network is implemented in Python, taking advantage of the NumPy and NumExprs libraries to speed up processing. Note that the built-in neural network is *not* used.

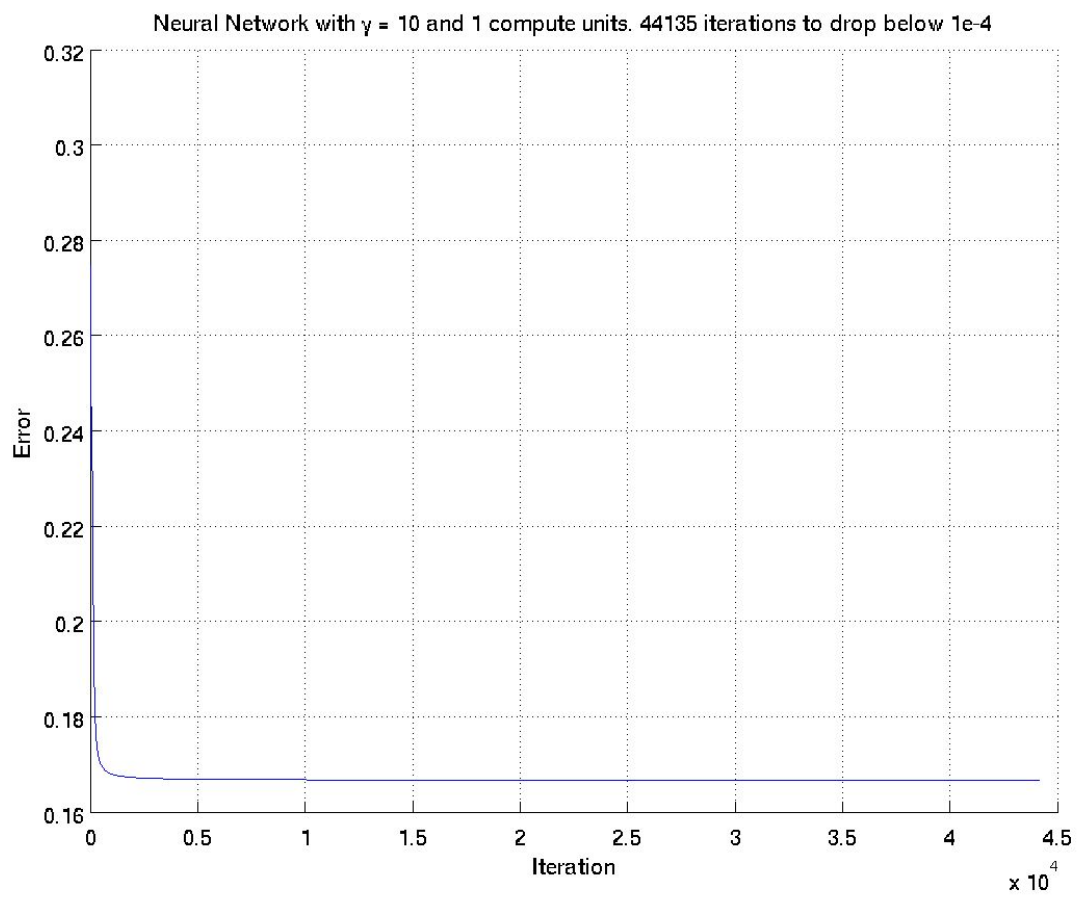
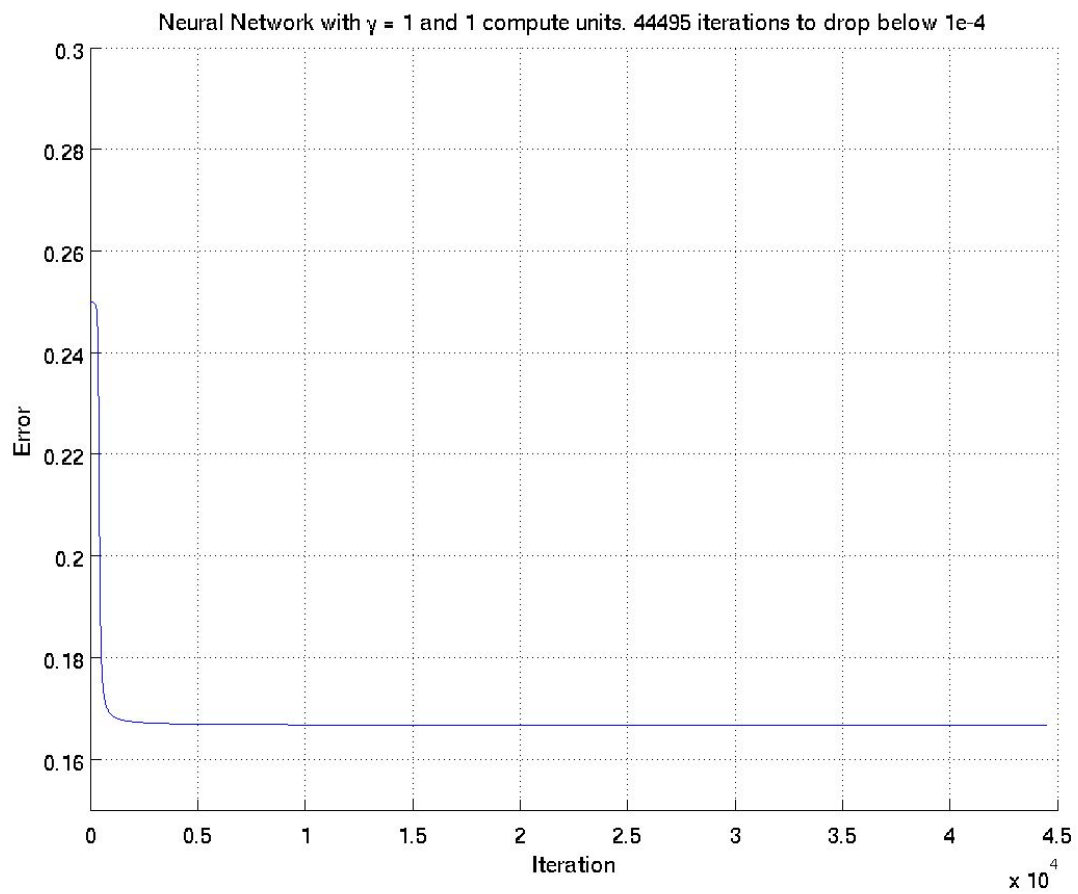
The graphs below show results for the XOR data set for various numbers of compute units (i.e. nodes in a hidden layer) and different values for gamma (i.e. the scale applied to the weights on each iteration). In each case, the neural network is run until the error is less than $10e-4$ (somewhat arbitrarily chosen). The number of iterations required to reach that threshold is provided. A simple gamma is applied to each weight.

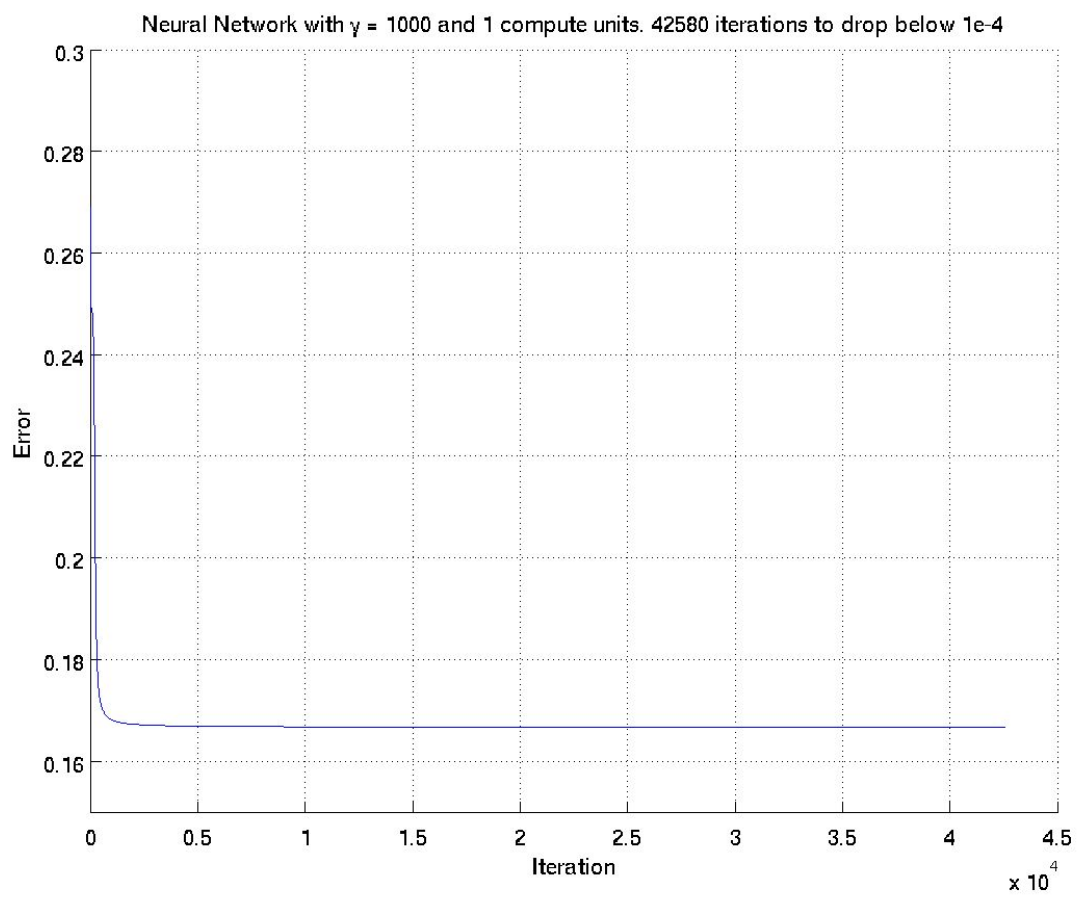
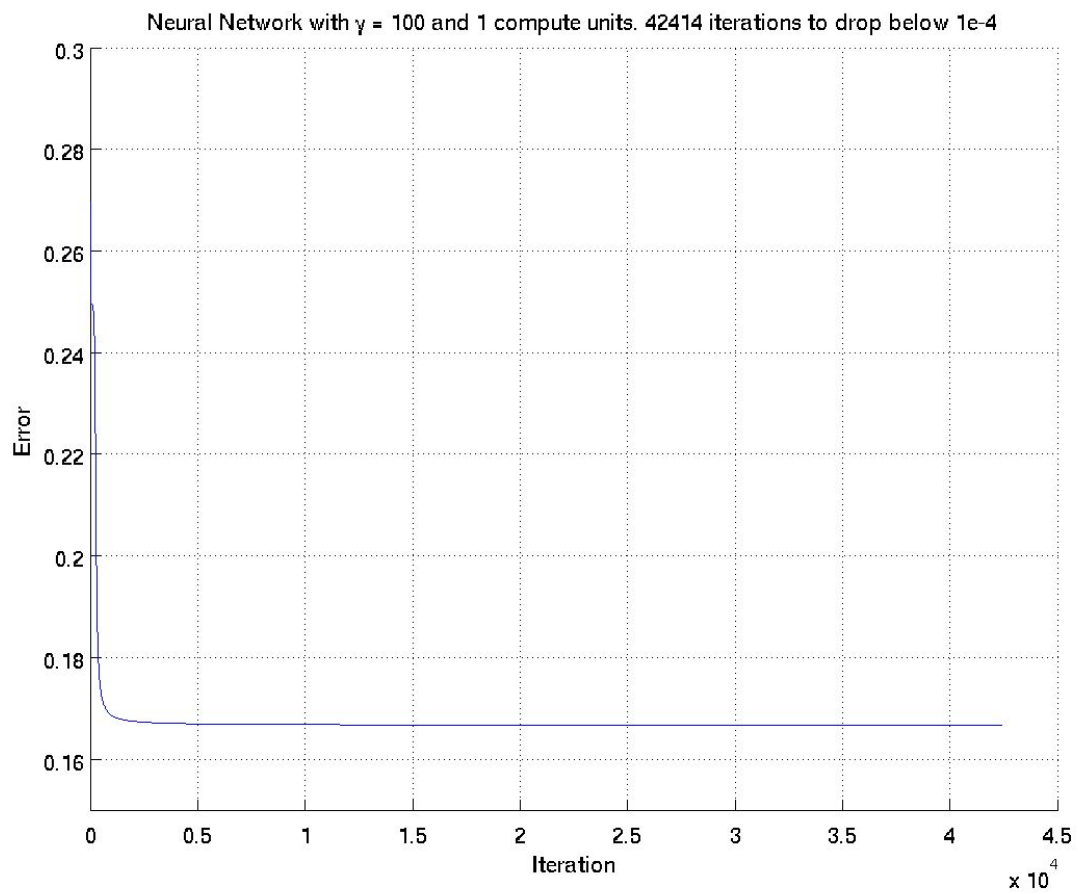
For the case where one compute unit was used, the classification was halted prematurely when the error reached approximately 0.16 as the network never improves much beyond this value. This behavior is expected with only a single compute unit. The best (least number of iterations) performance was seen with the $\gamma=10$ and 10 compute units. This likely represents a good balance of convergence speed with number of compute units. The result is also affected by random weight selection.

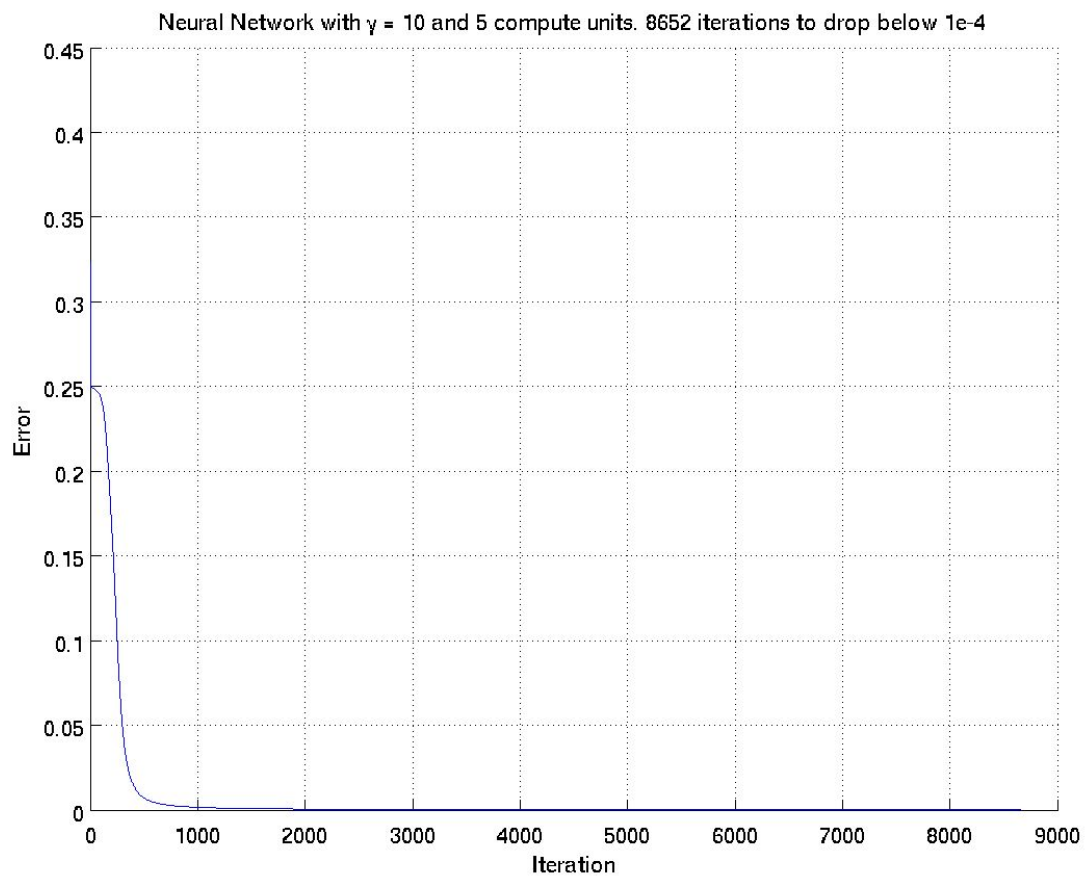
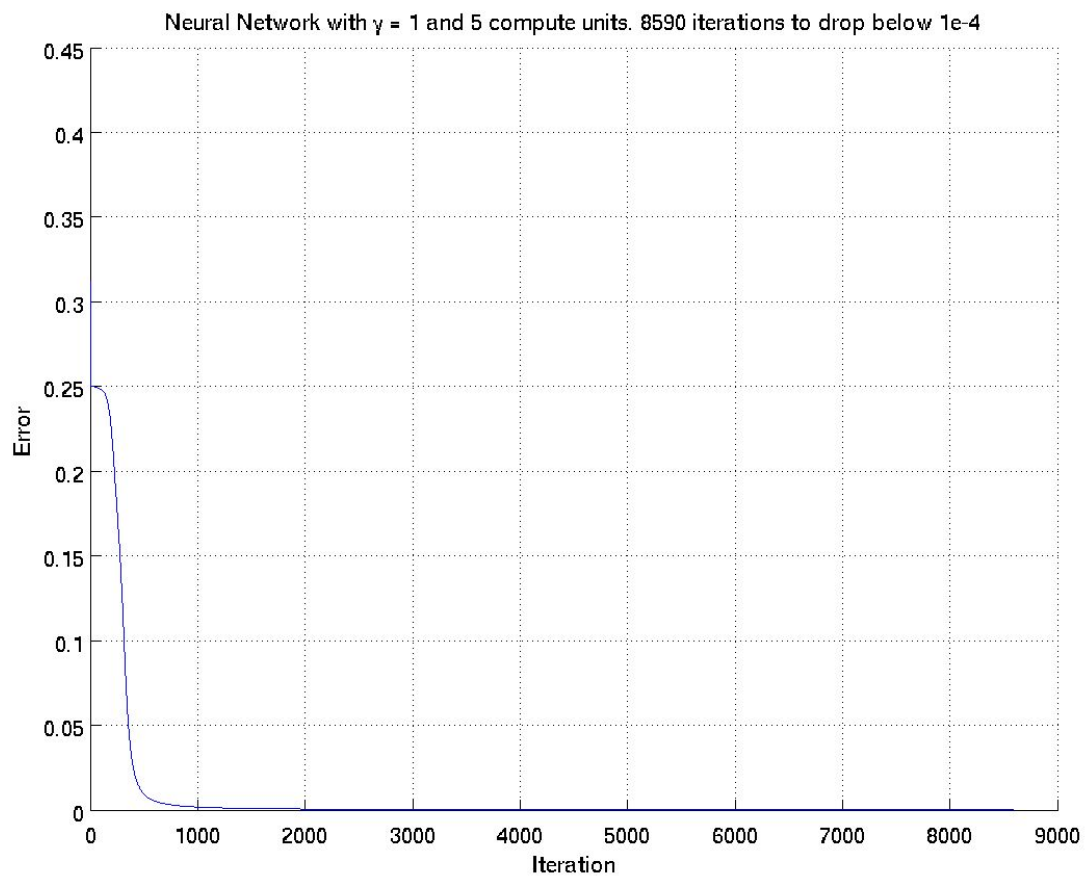
In general, more compute units improved the performance of the detector on the XOR dataset.

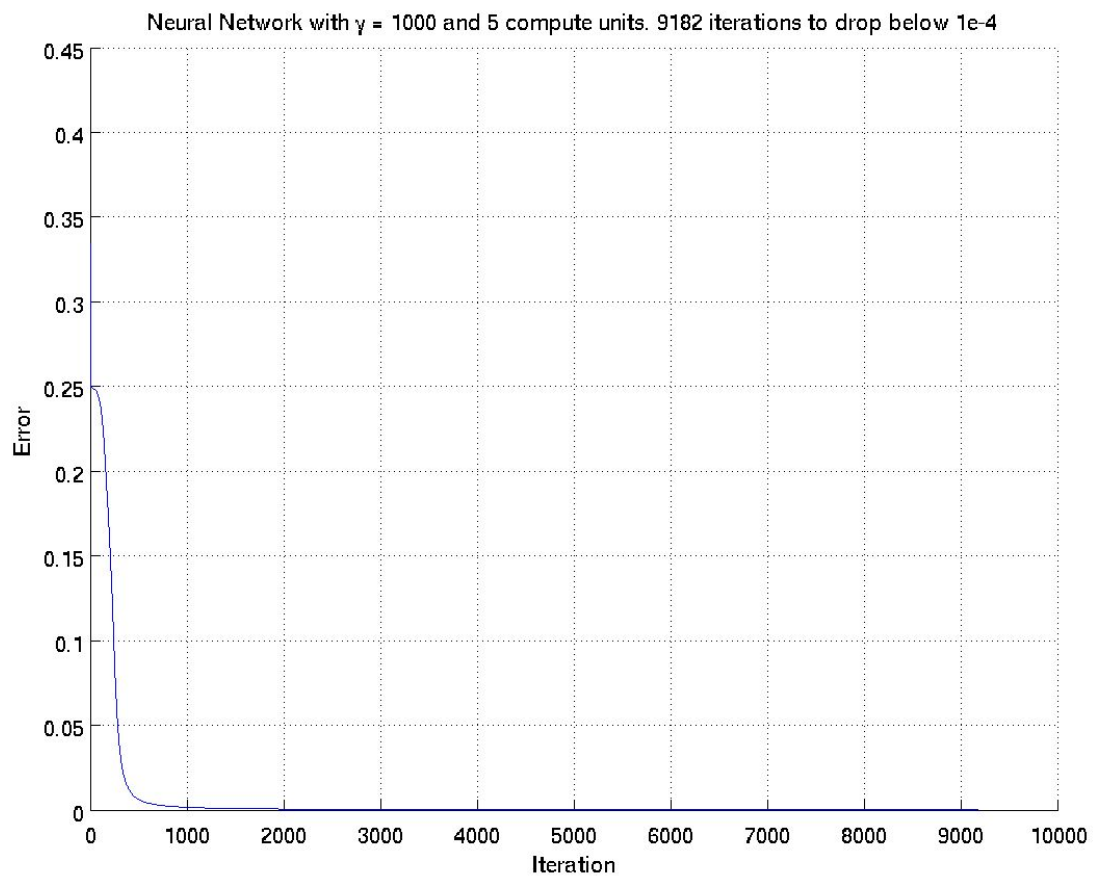
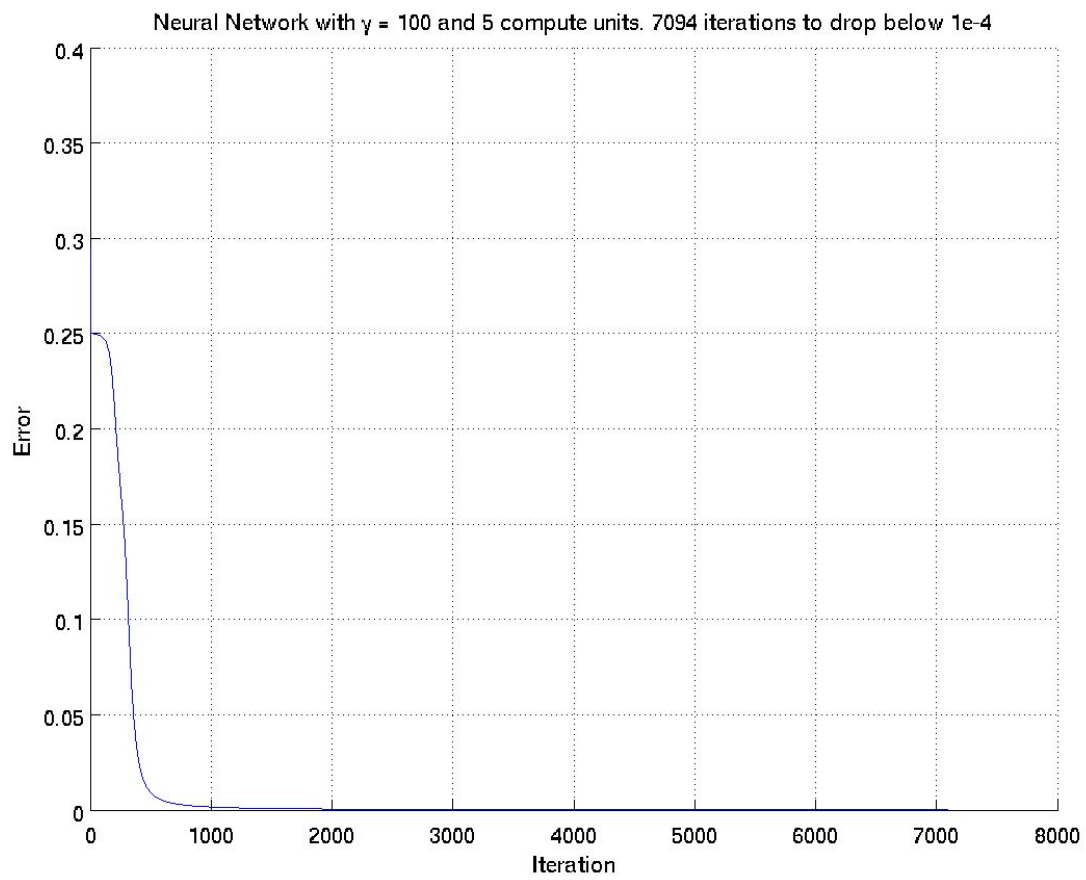
The following table describes the number of iterations performed for each gamma/compute unit pair:

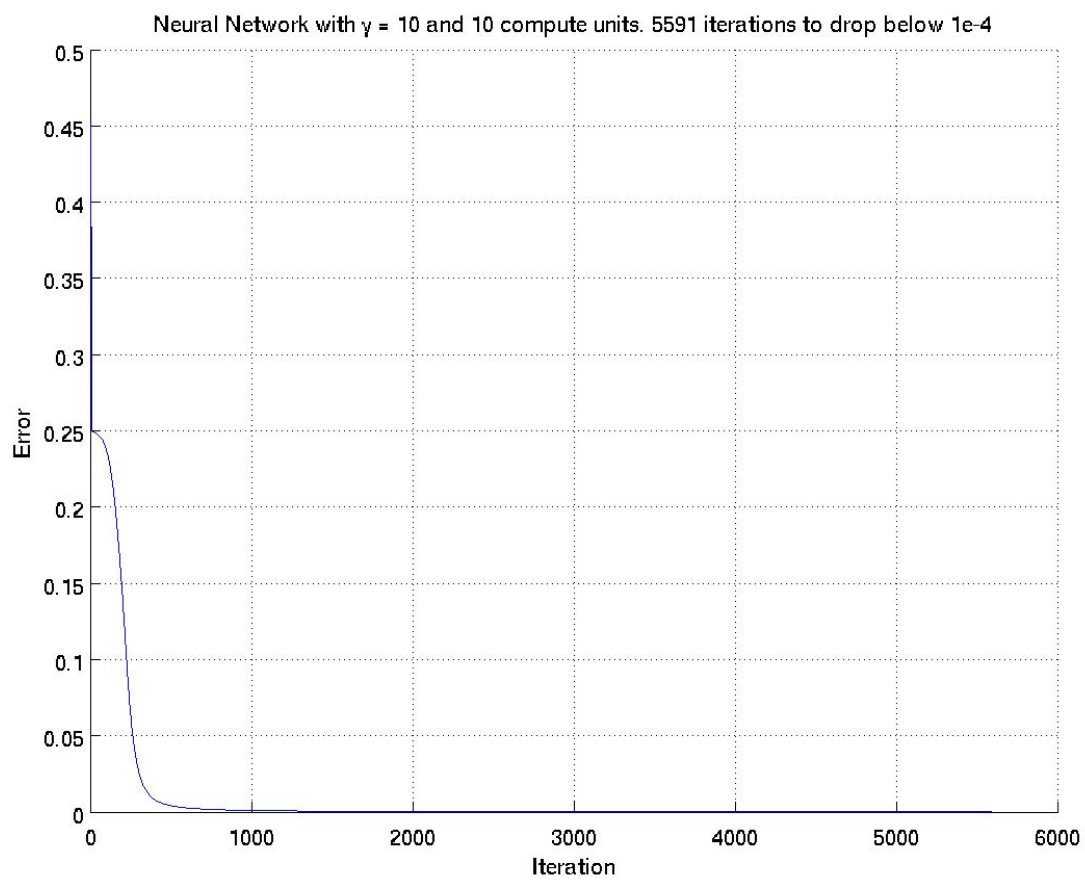
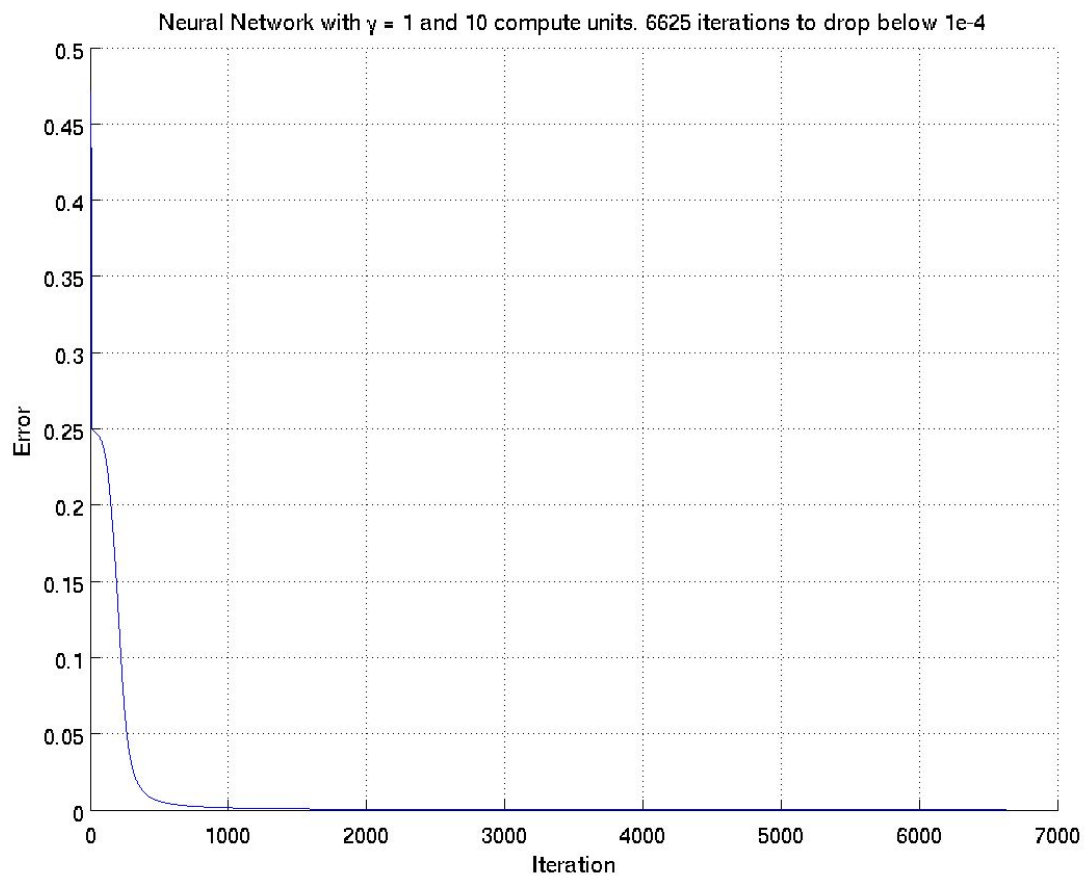
Number of Iterations	Trial (first number=num compute units, second number=gamma)
44495	./xor_1e-4/result_1_1.000000.csv
42580	./xor_1e-4/result_1_1000.000000.csv
8590	./xor_1e-4/result_5_1.000000.csv
8652	./xor_1e-4/result_5_10.000000.csv
6625	./xor_1e-4/result_10_1.000000.csv
44135	./xor_1e-4/result_1_10.000000.csv
6285	./xor_1e-4/result_10_100.000000.csv
9182	./xor_1e-4/result_5_1000.000000.csv
7094	./xor_1e-4/result_5_100.000000.csv
5591	./xor_1e-4/result_10_10.000000.csv
42414	./xor_1e-4/result_1_100.000000.csv
7036	./xor_1e-4/result_10_1000.000000.csv
232679	Total

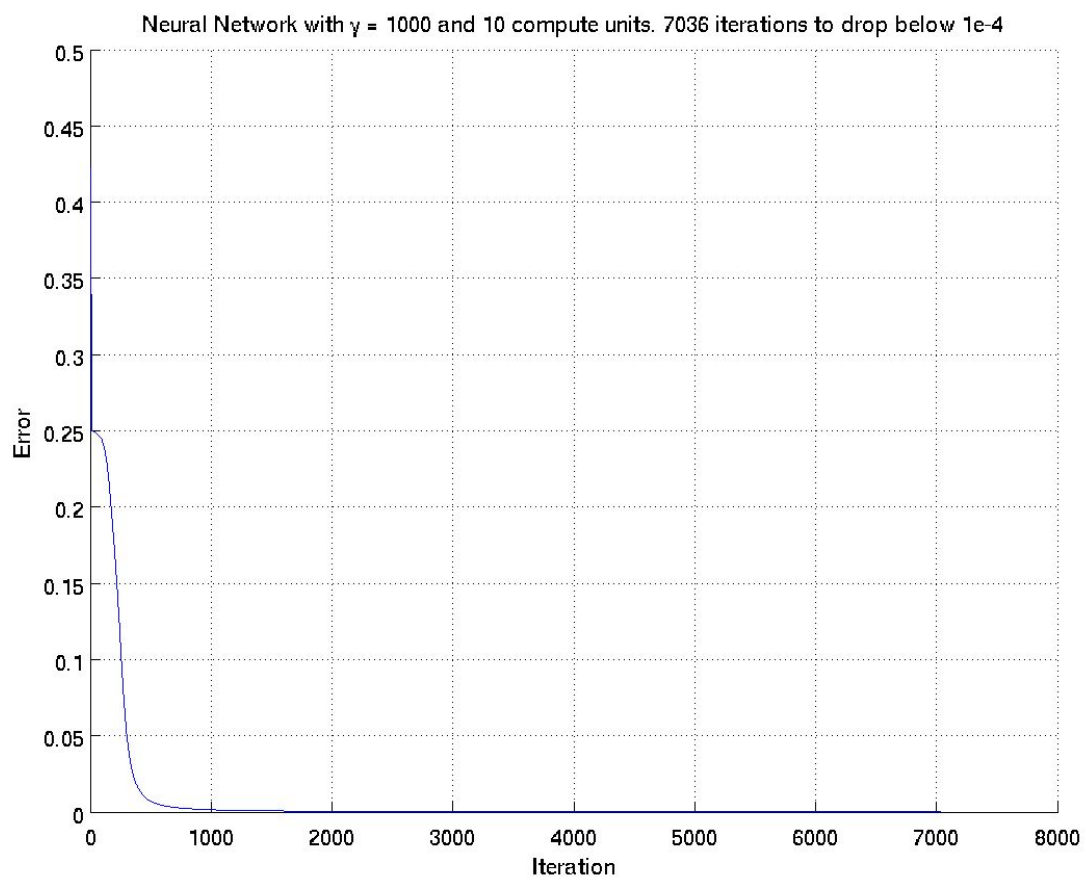
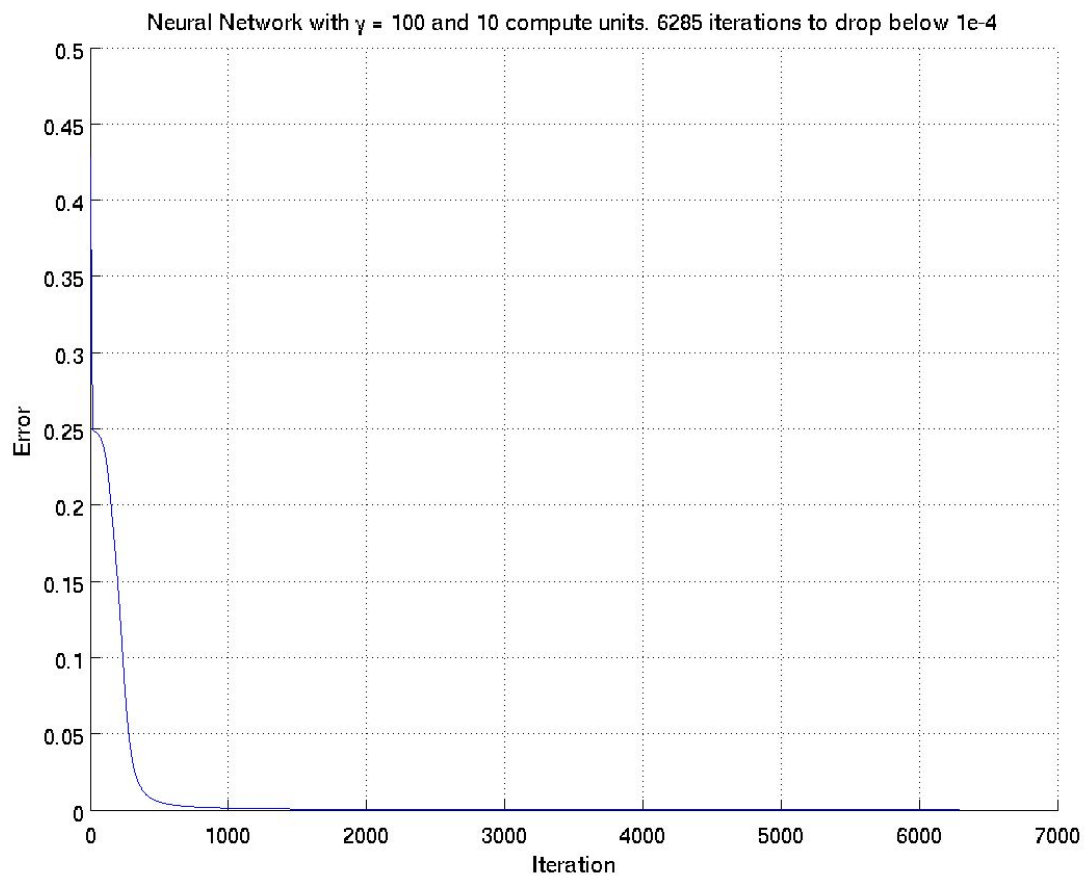












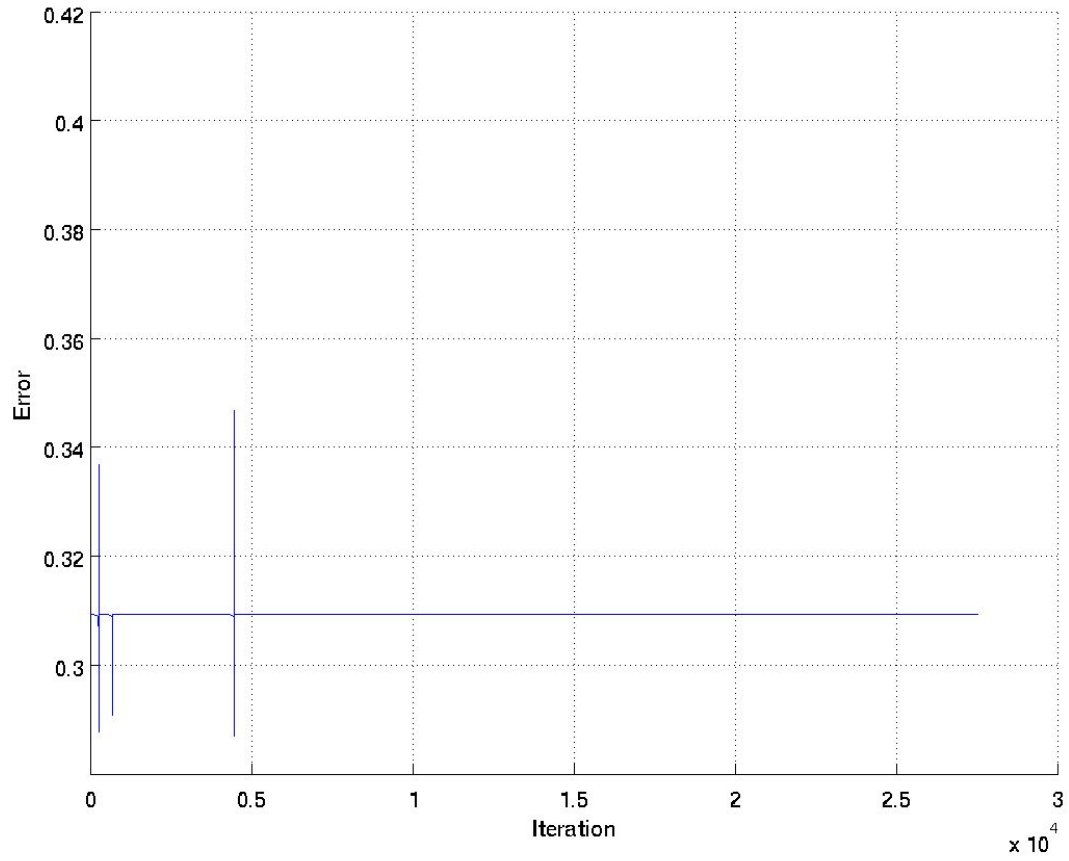
The prostate data set contains 97 data samples. Of those samples, there are 67 “has cancer” samples and 30 “does not have cancer” samples. The neural network is trained on all the samples. Similar charts are provided for this data set. It is noted that, without implementing a proper method to adjust gamma during runtime, and this data set our classifier either operated too slowly to move anywhere for small values of gamma, or skipped past a minimum for large values of gamma. Thus the minimum of the error does not occur at the last iteration. In order to address this issue, gamma should be adjusted during runtime as a function of the network’s gradient.

The minimum error computed was 0.2604 for gamma 1000 and 10 compute units, during 20905 iterations. The minimum was actually found early in the data set, however due to the gamma issues described above the minimum was bypassed and the neural network became essentially stuck on a plateau of the sigmoid function.

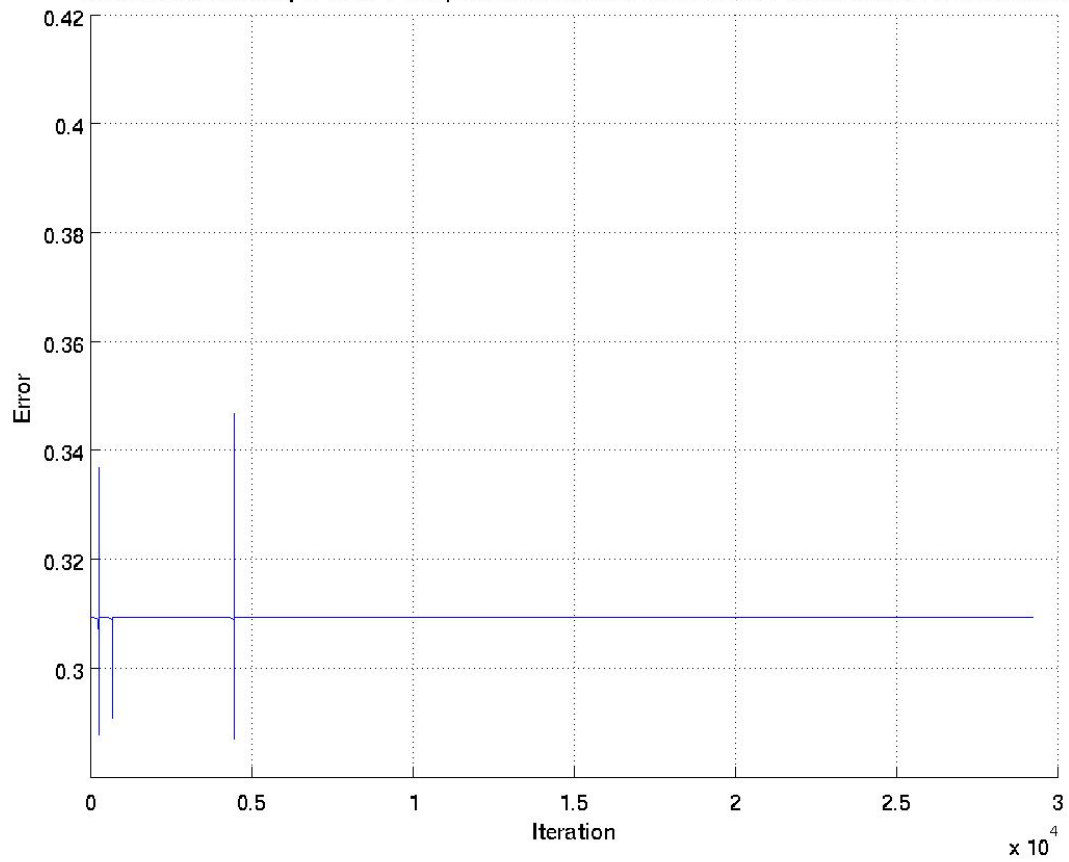
The following table describes the number of iterations performed for each gamma/compute unit pair:

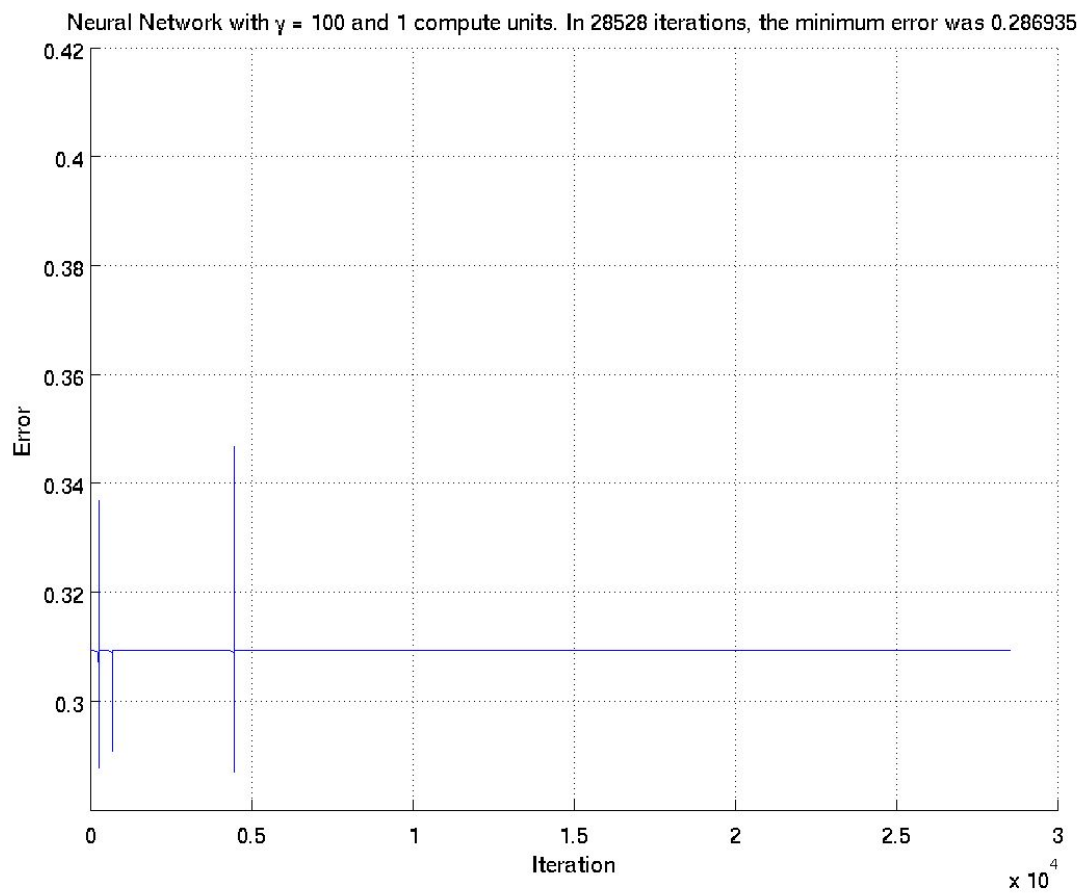
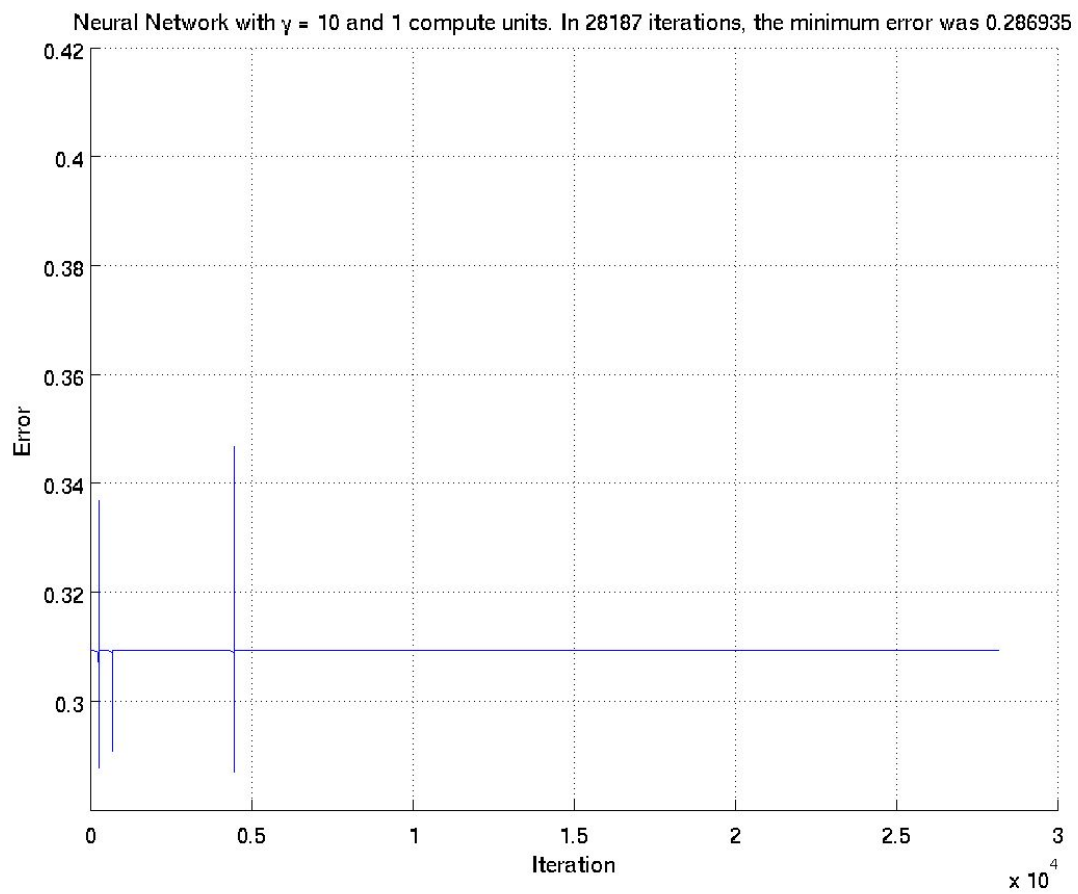
Number of Iterations	Trial (first number=num compute units, second number=gamma)
29245	./result_1_1.000000.csv
12314	./result_10_0.100000.csv
30222	./result_1_1000.000000.csv
12876	./result_5_1.000000.csv
19926	./result_5_10.000000.csv
12444	./result_10_1.000000.csv
27520	./result_1_0.100000.csv
28187	./result_1_10.000000.csv
12487	./result_10_100.000000.csv
10487	./result_5_1000.000000.csv
10661	./result_5_100.000000.csv
12861	./result_5_0.100000.csv
12401	./result_10_10.000000.csv
28528	./result_1_100.000000.csv
20905	./result_10_1000.000000.csv
281064	Total

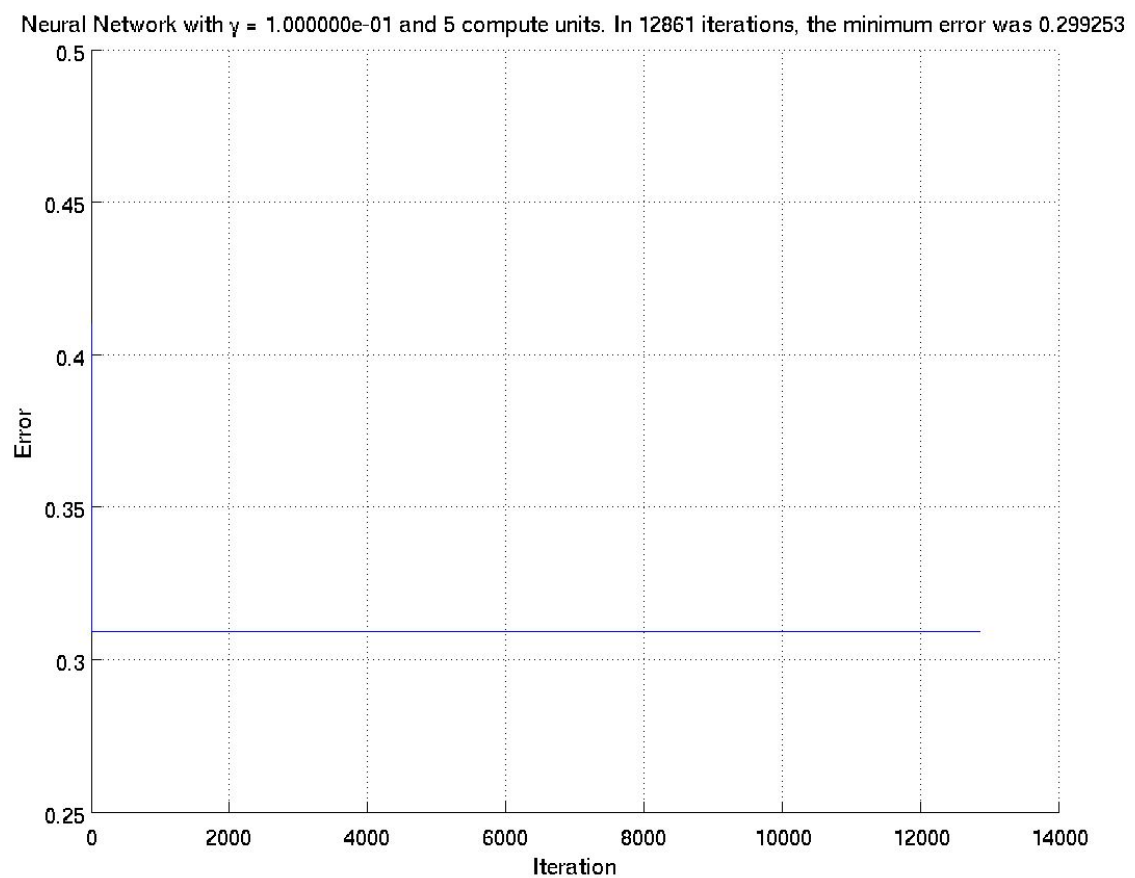
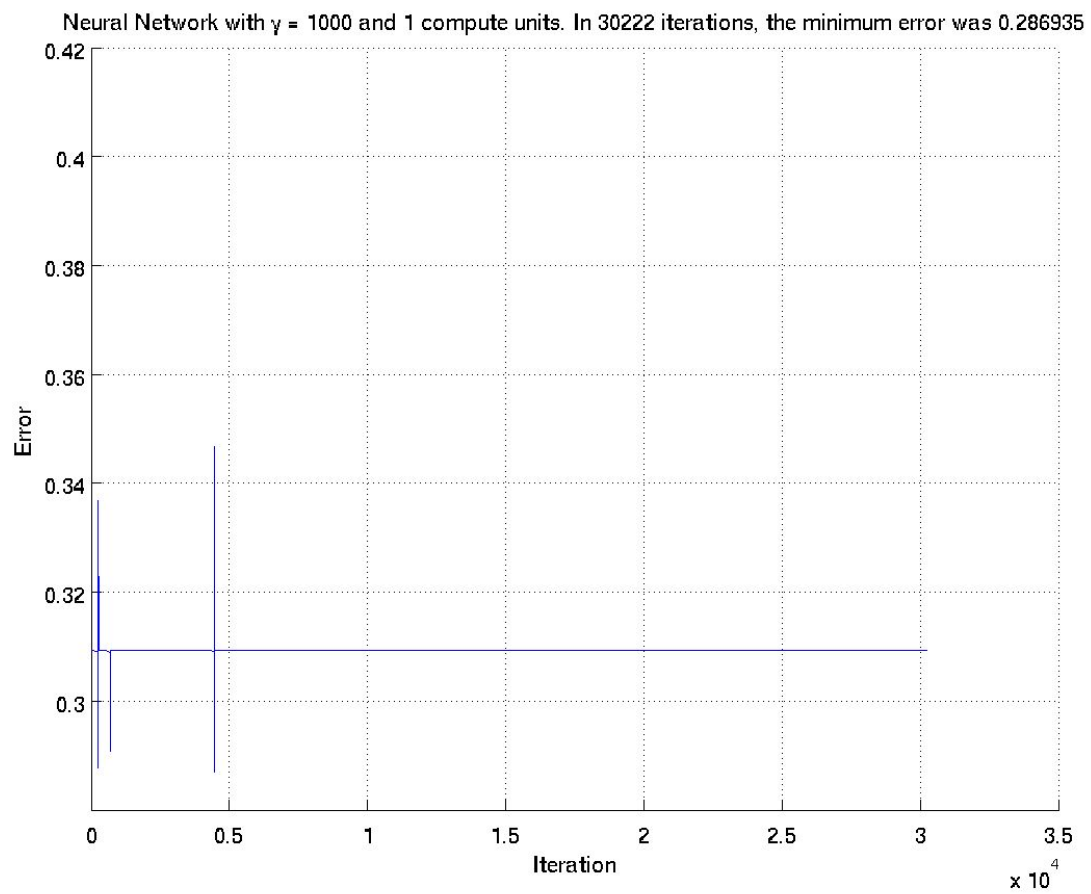
Neural Network with $\gamma = 1.000000\text{e-}01$ and 1 compute units. In 27520 iterations, the minimum error was 0.286935

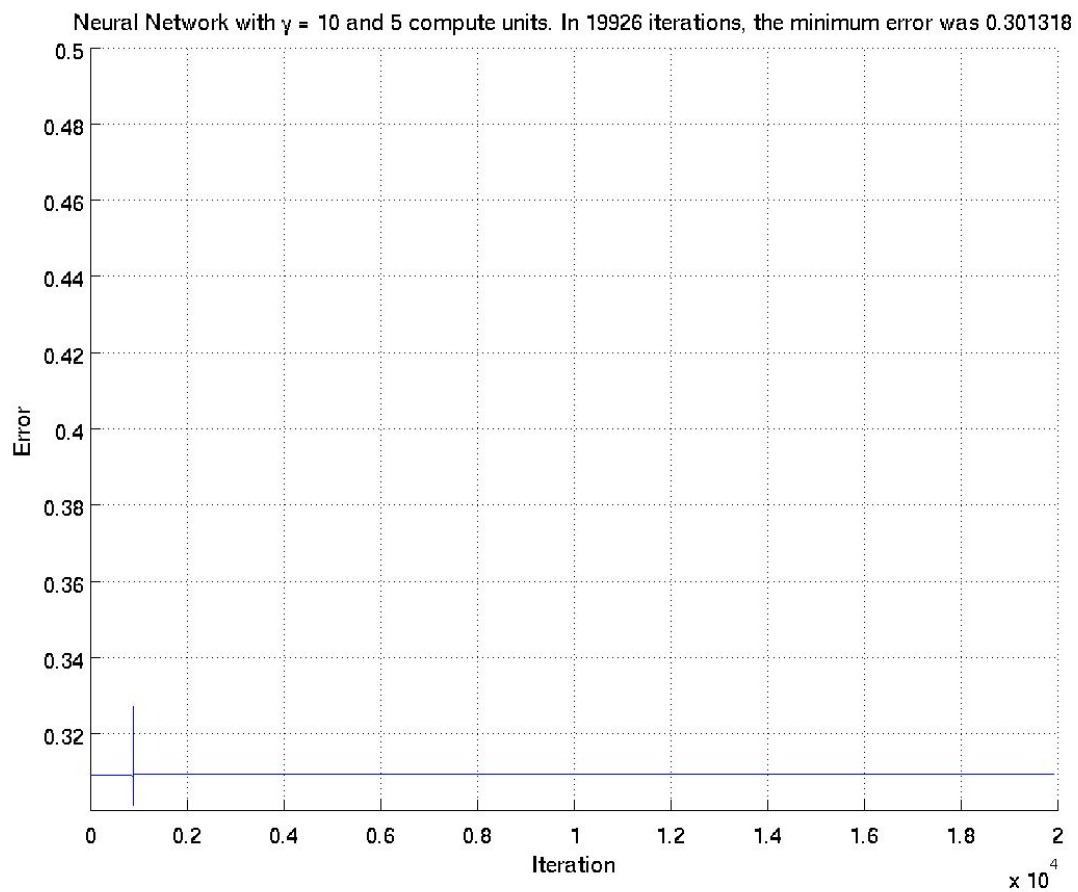
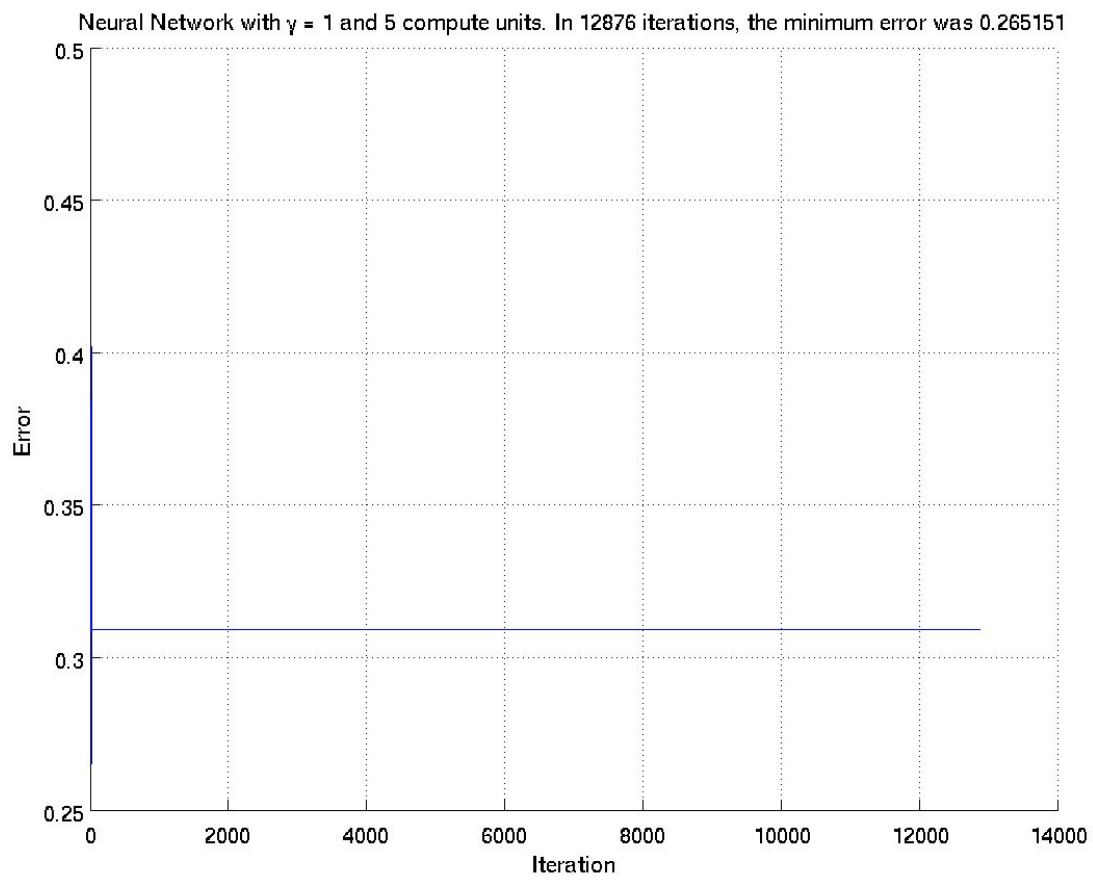


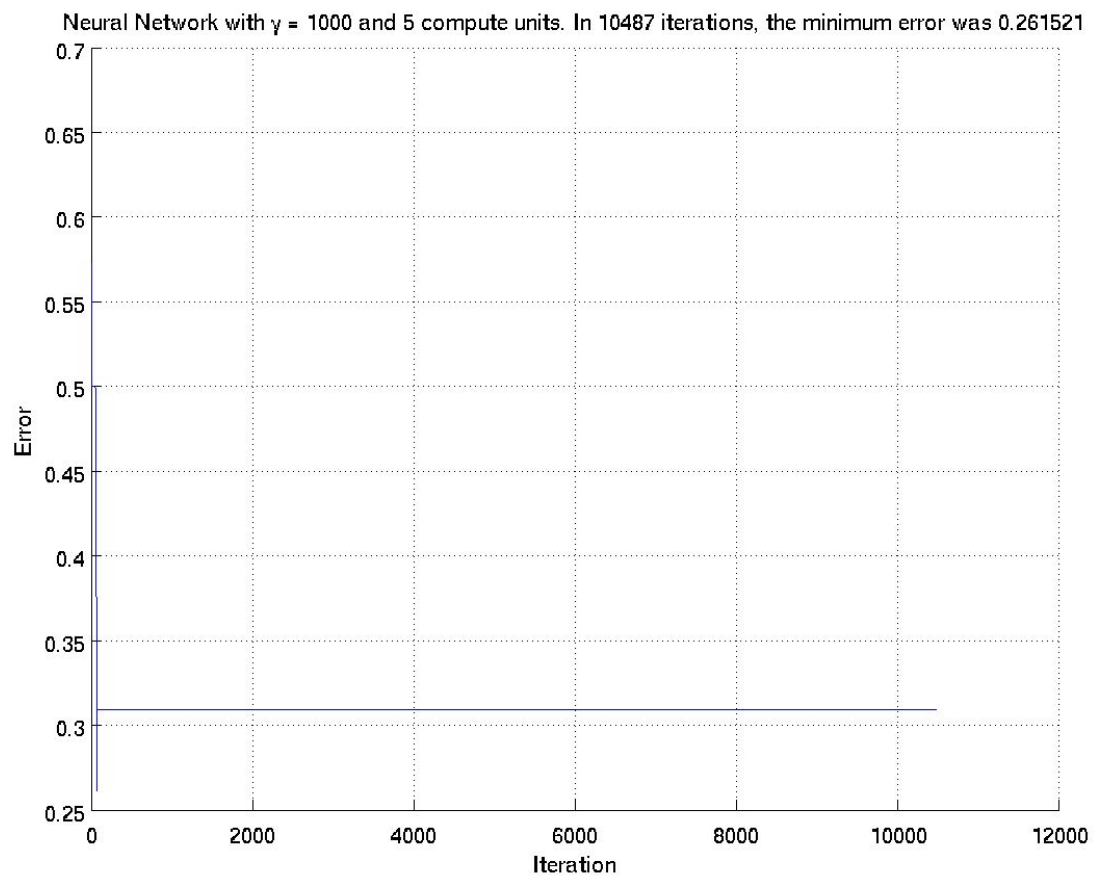
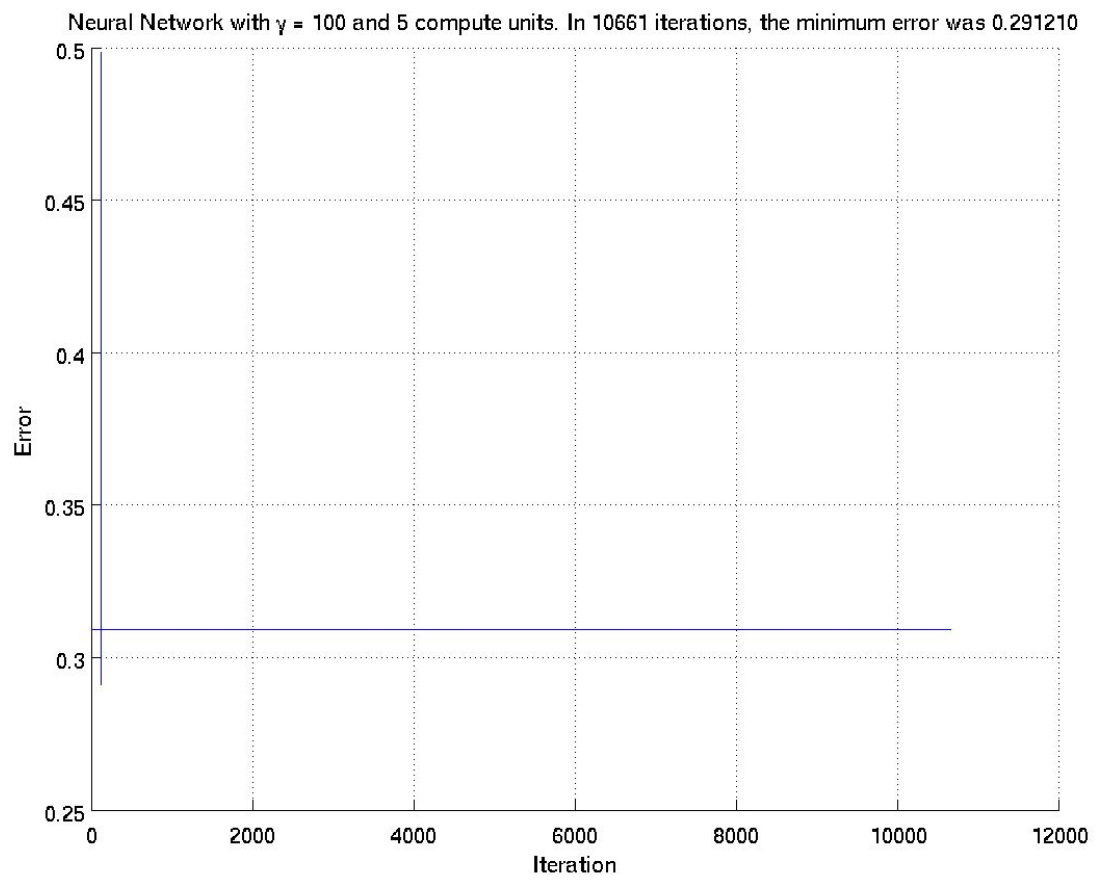
Neural Network with $\gamma = 1$ and 1 compute units. In 29245 iterations, the minimum error was 0.286935



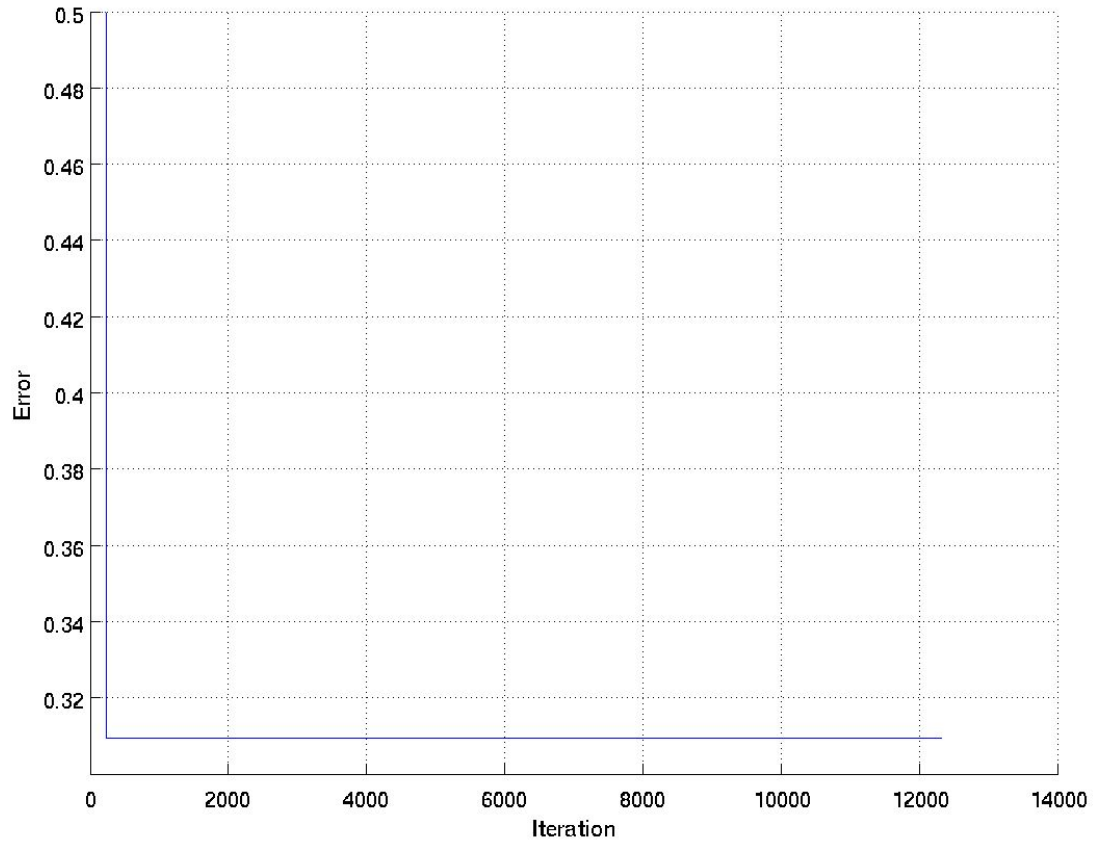




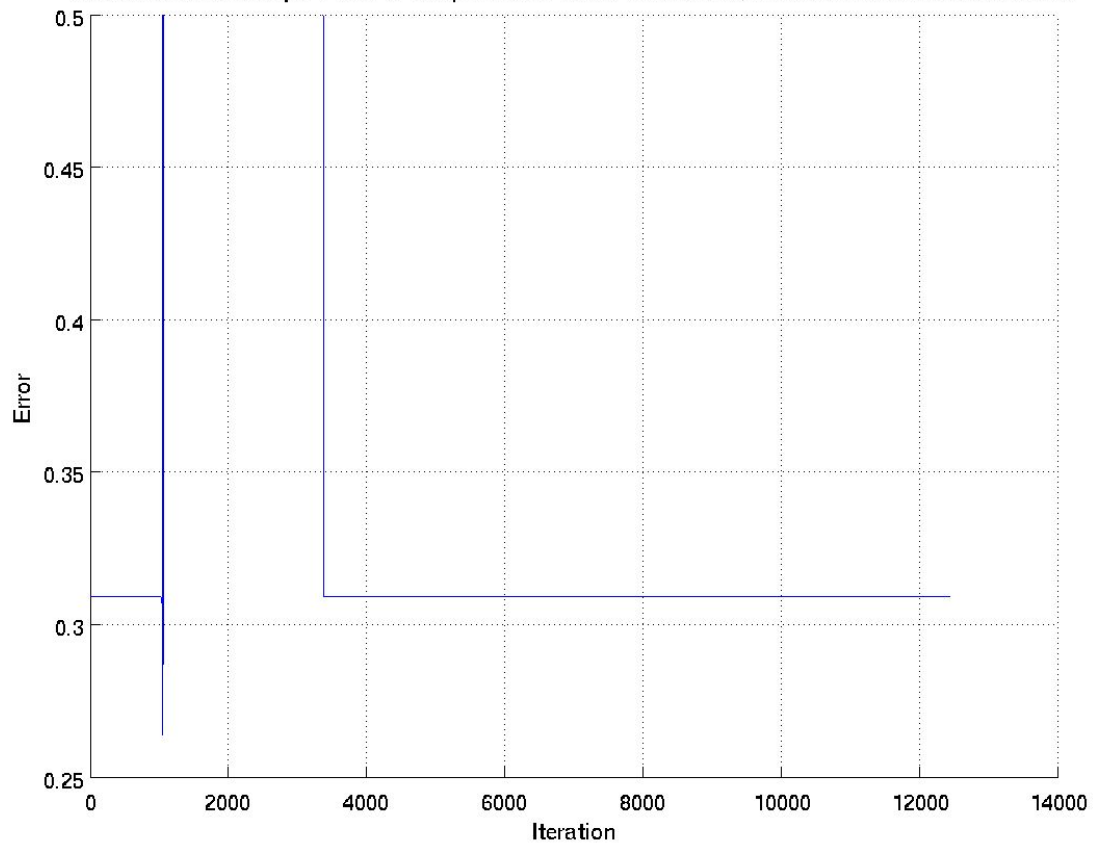


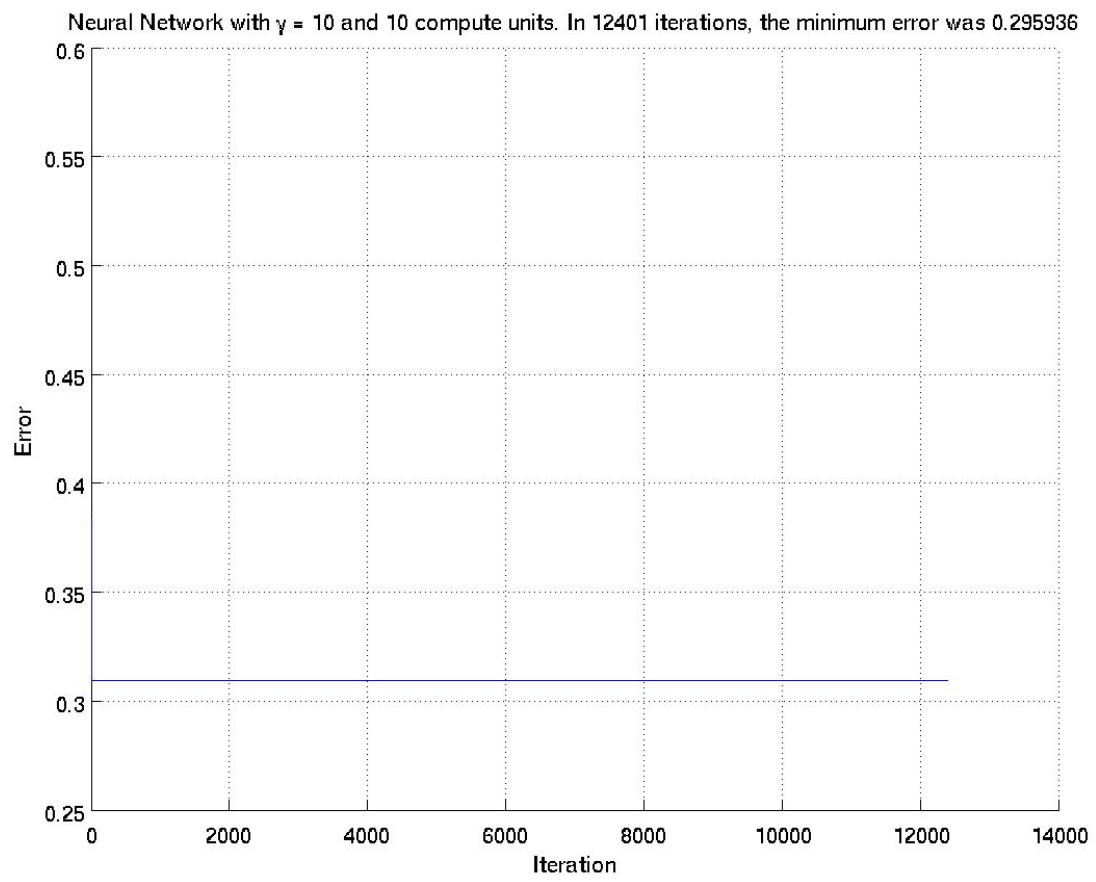


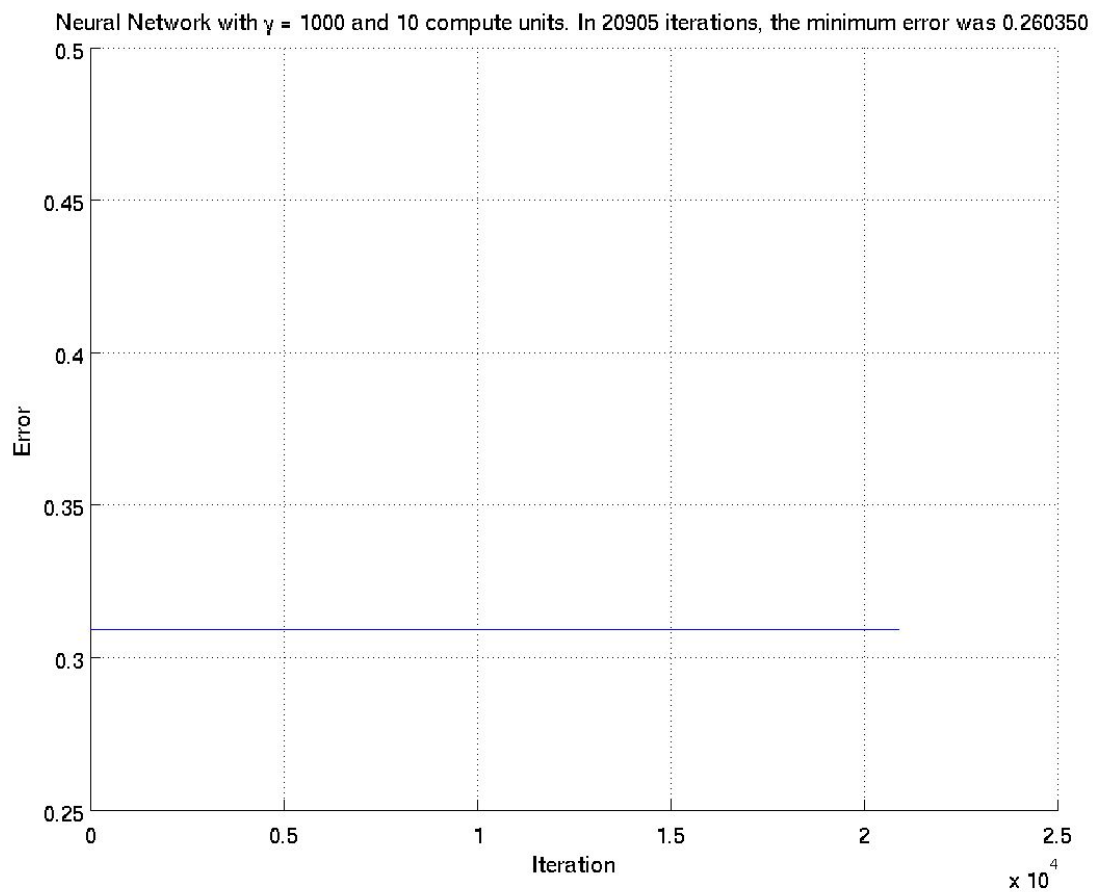
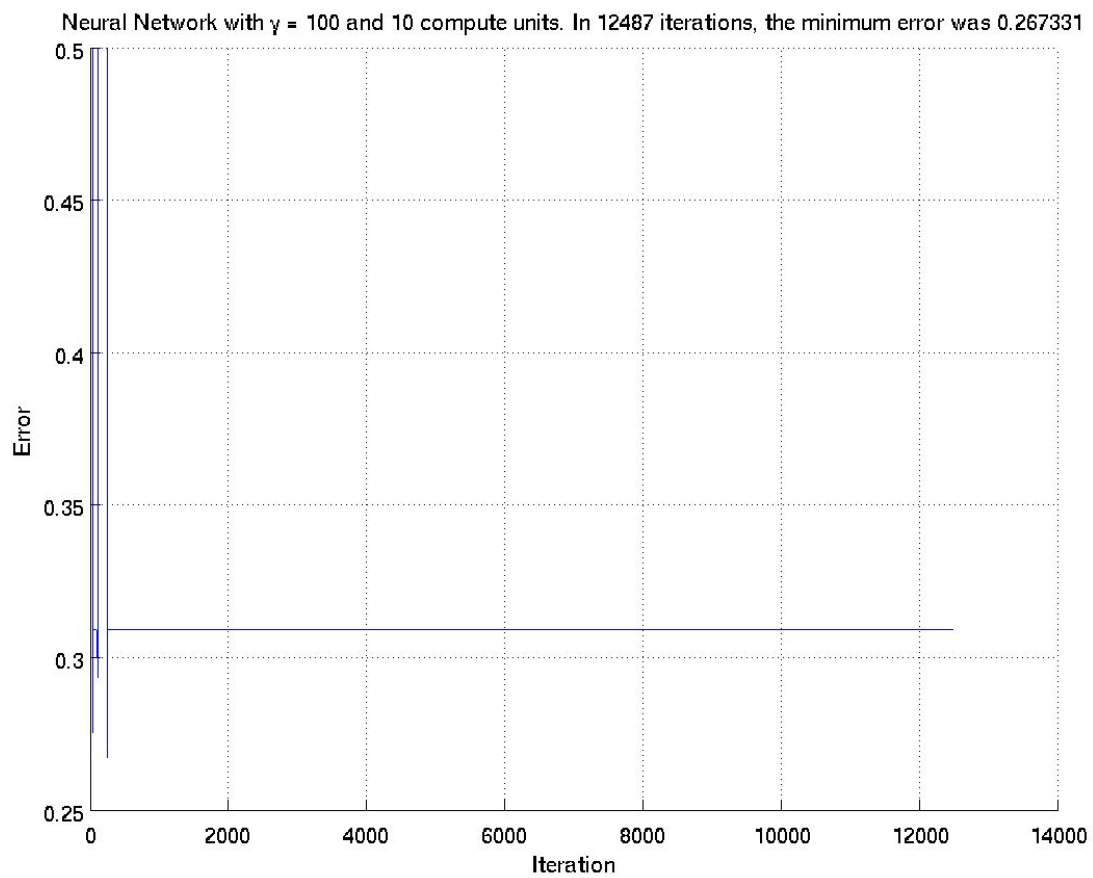
Neural Network with $\gamma = 1.000000\text{e-}01$ and 10 compute units. In 12314 iterations, the minimum error was 0.309278



Neural Network with $\gamma = 1$ and 10 compute units. In 12444 iterations, the minimum error was 0.263910







In total, 513846 iterations of the neural network were performed to accumulate this data, taking approximately one hour (about 10,000 iterations per minute).

2 - Software Code Listing for the Neural Network Classifier

A single python program implements the classifier, with command line parameters varying the behavior of the program. The following parameters are implemented:

```
python neural_network.py dataset.csv num_compute_units
```

Where dataset.csv is the prostate dataset or similarly formatted XOR dataset, and num_compute_units is an integer greater than or equal to 1 indicating the number of compute units in the hidden layer

The program prints to a file one line for each iteration of the classifier, and halts when the error drops below $10e-4$ or when stopped by the user.

Note that in this code, an attempt was made at implementing an improved gamma function, however the functionality was not completed prior to generating the data above. The associated code is included for completeness.