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.

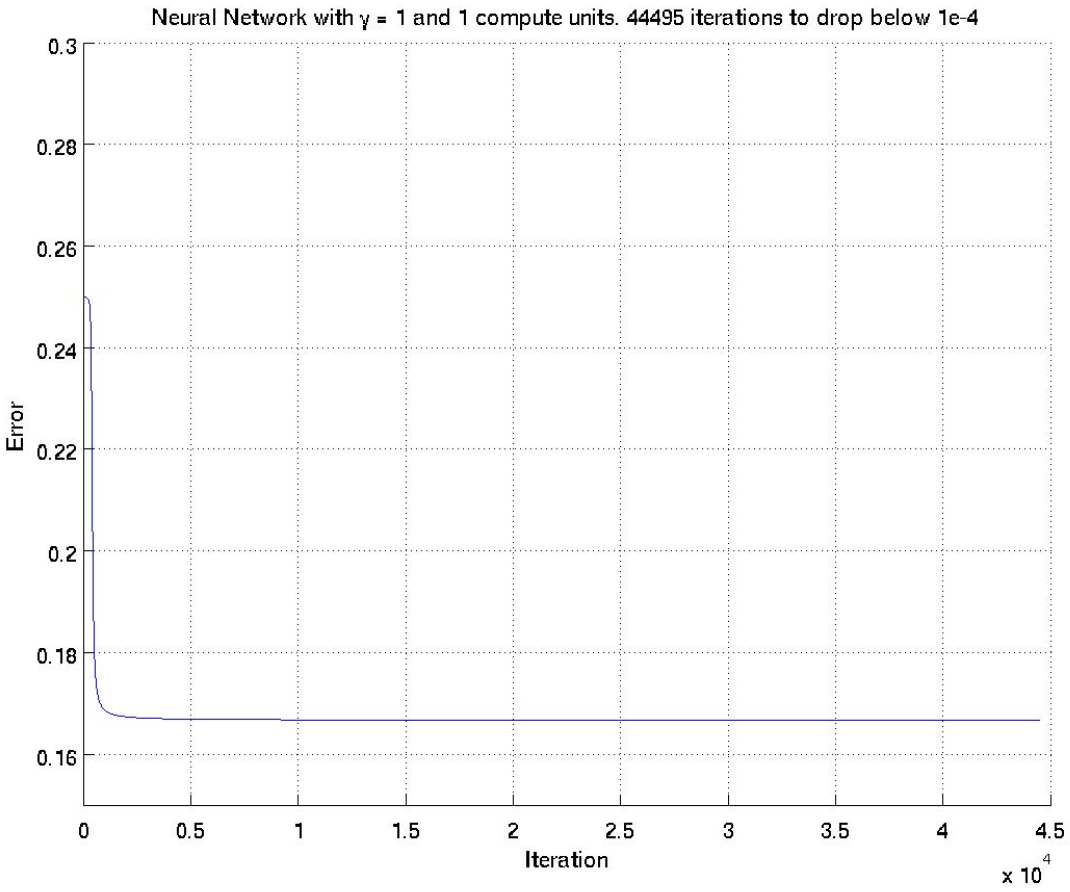
1 - Neural Network Approach, Results, and Evaluation

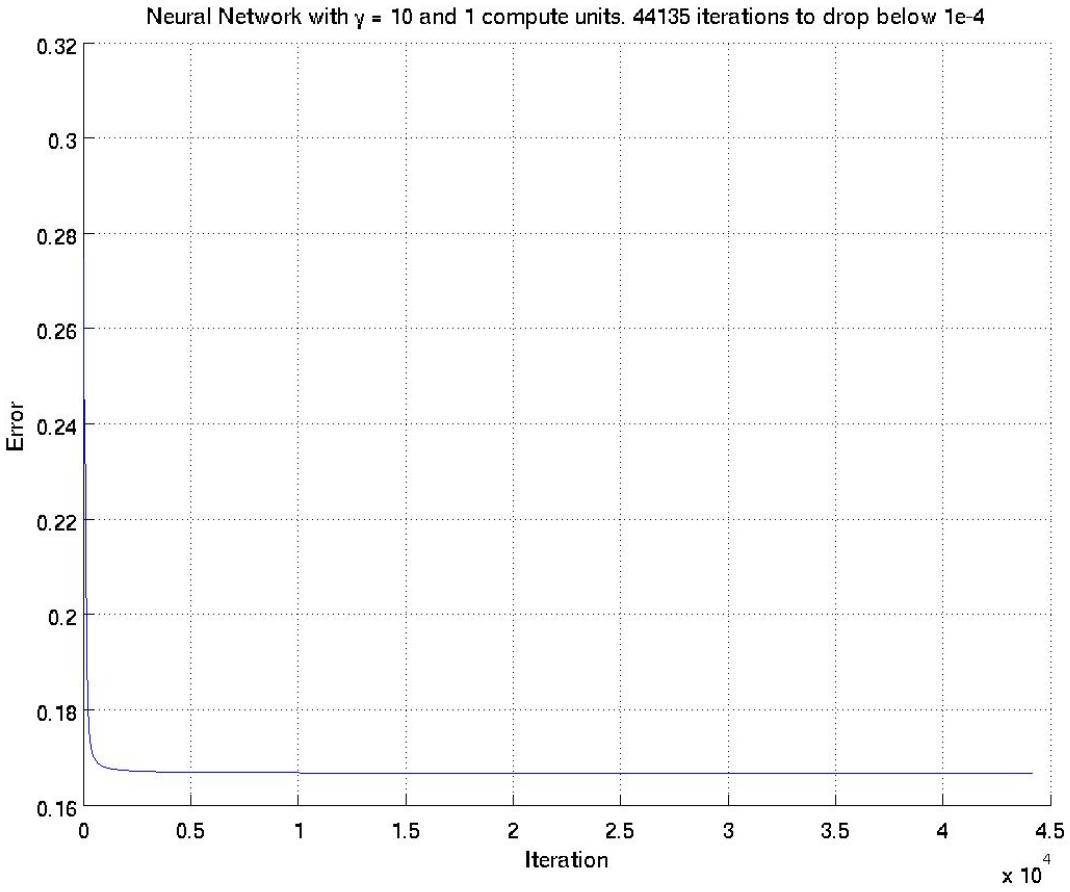
The neural network is implemented in Python, taking advantage of the NumPy and NumExprs libraries to speed up processing. Note that the built-in neural network is *not* used.

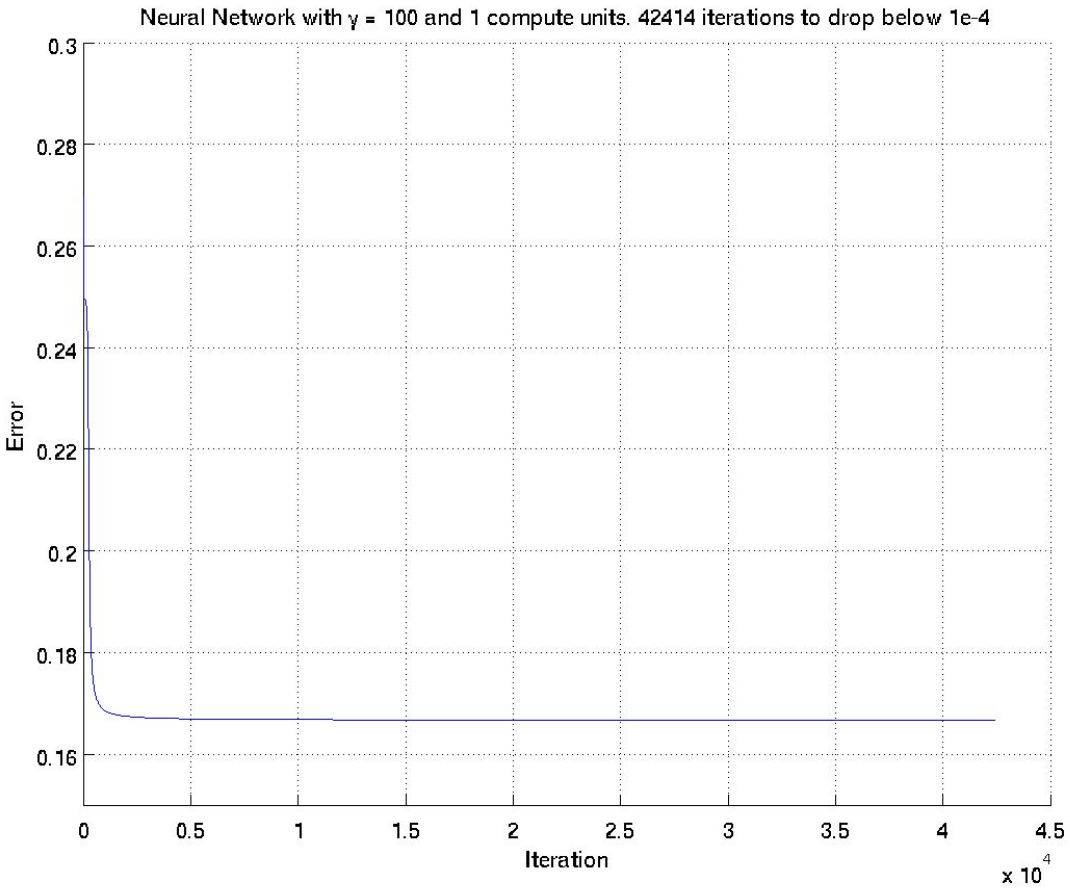
The graphs below show results for the XOR data set for various numbers of compute units (i.e. nodes in a hidden layer) and different values for gamma (i.e. the scale applied to the weights on each iteration). In each case, the neural network is run until the error is less than 10e-4 (somewhat arbitrarily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

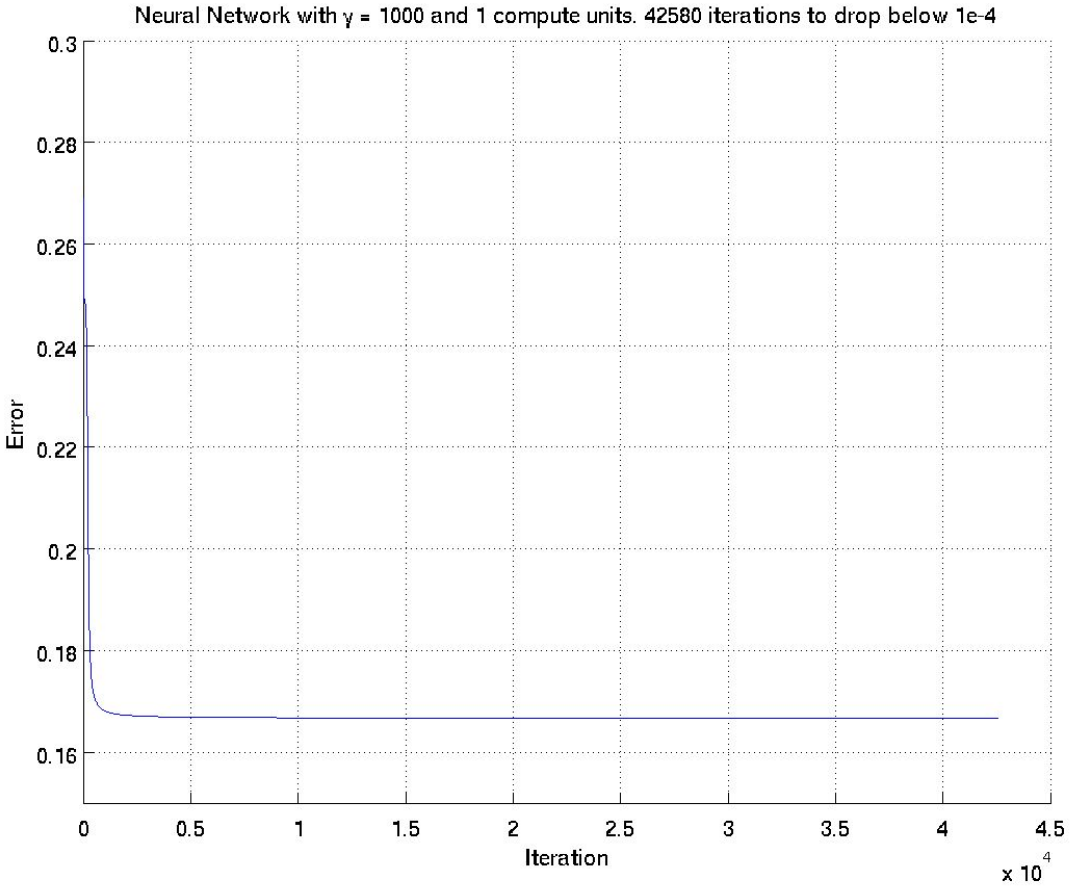
For the case where one compute unit was used, the classification was halted prematurely when the error reached approximately 0.16 as the network never improves much beyond this value. This behavior is expected with only a single compute unit. The best (least number of iterations) performance was seen 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.

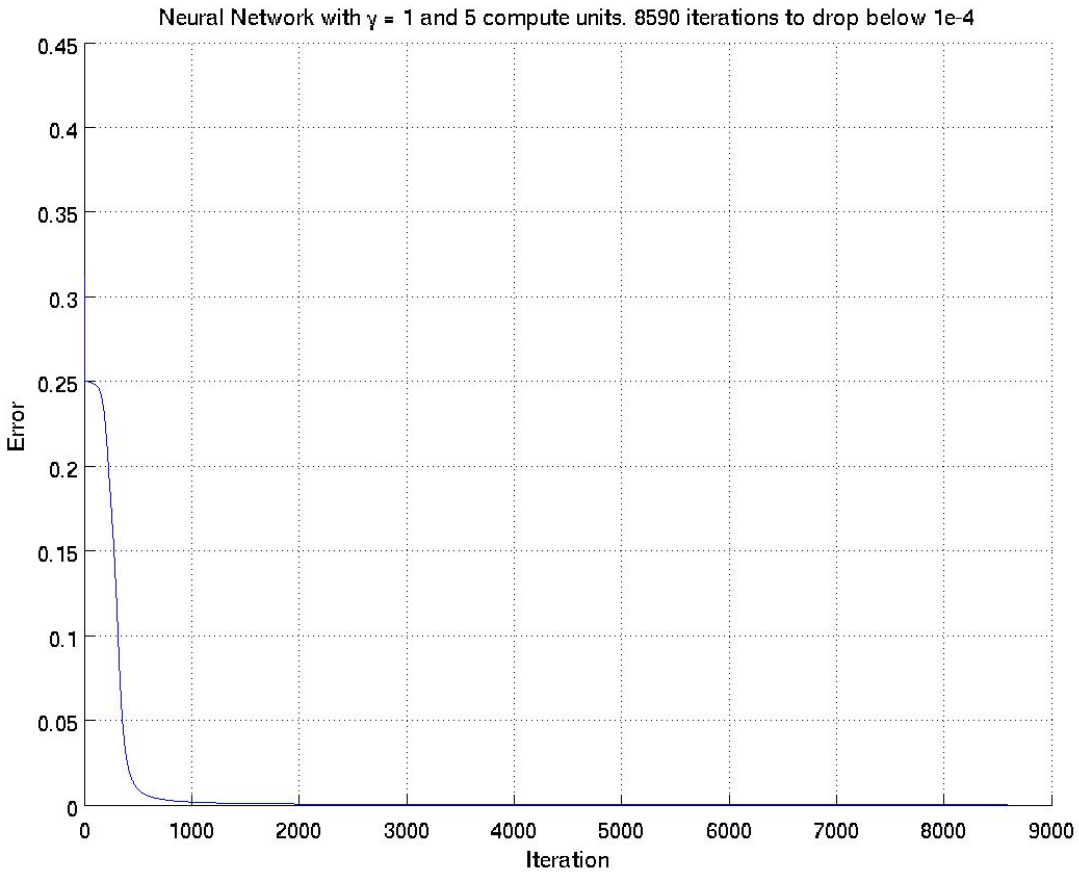
In general, more compute units improved the performance of the detector on the XOR dataset.

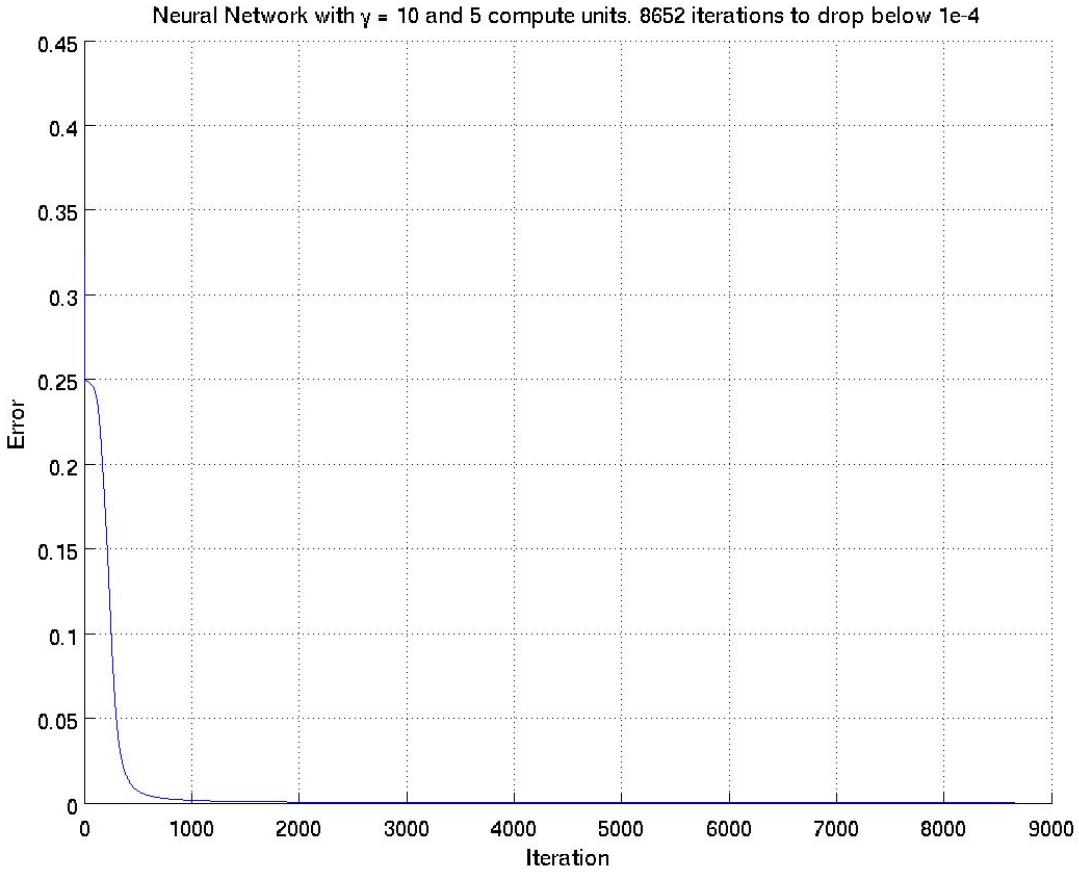

Neural Network with γ = 1 and 1 compute units. 44495 iterations to drop below 1e-4
Error
Iteration
x 10^4
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

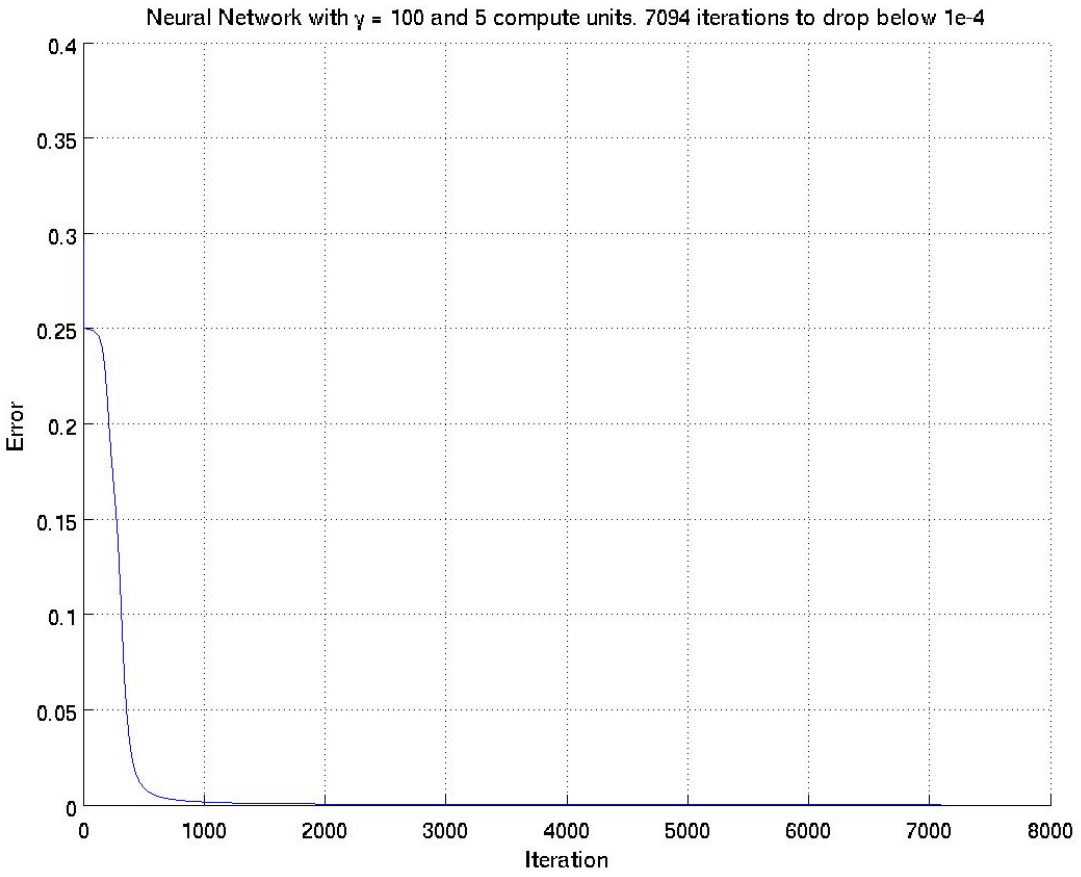

Neural Network with γ = 10 and 1 compute units. 44135 iterations to drop below 1e-4
Error
Iteration
x 10^4
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

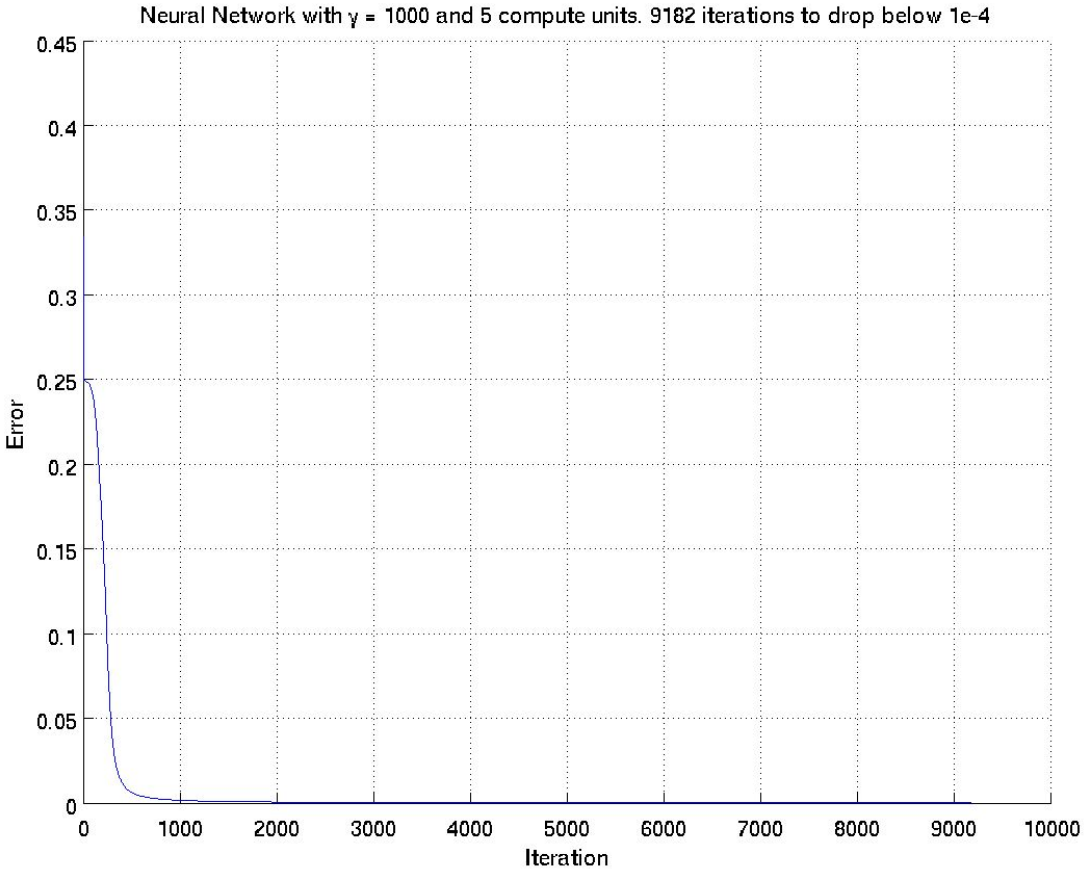

Neural Network with γ = 100 and 1 compute units. 42414 iterations to drop below 1e-4
Error
Iteration
x 10^4
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

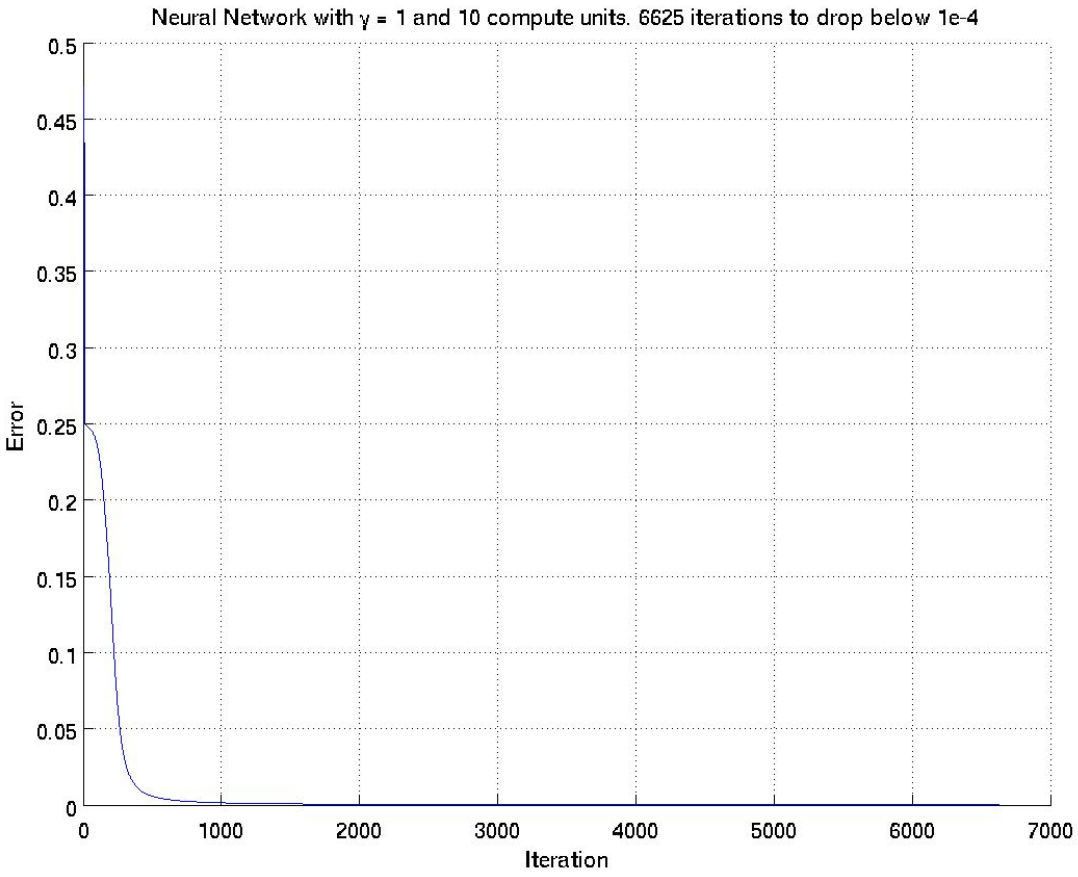

Neural Network with γ = 1000 and 1 compute units. 42580 iterations to drop below 1e-4
Error
Iteration
x 10^4
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

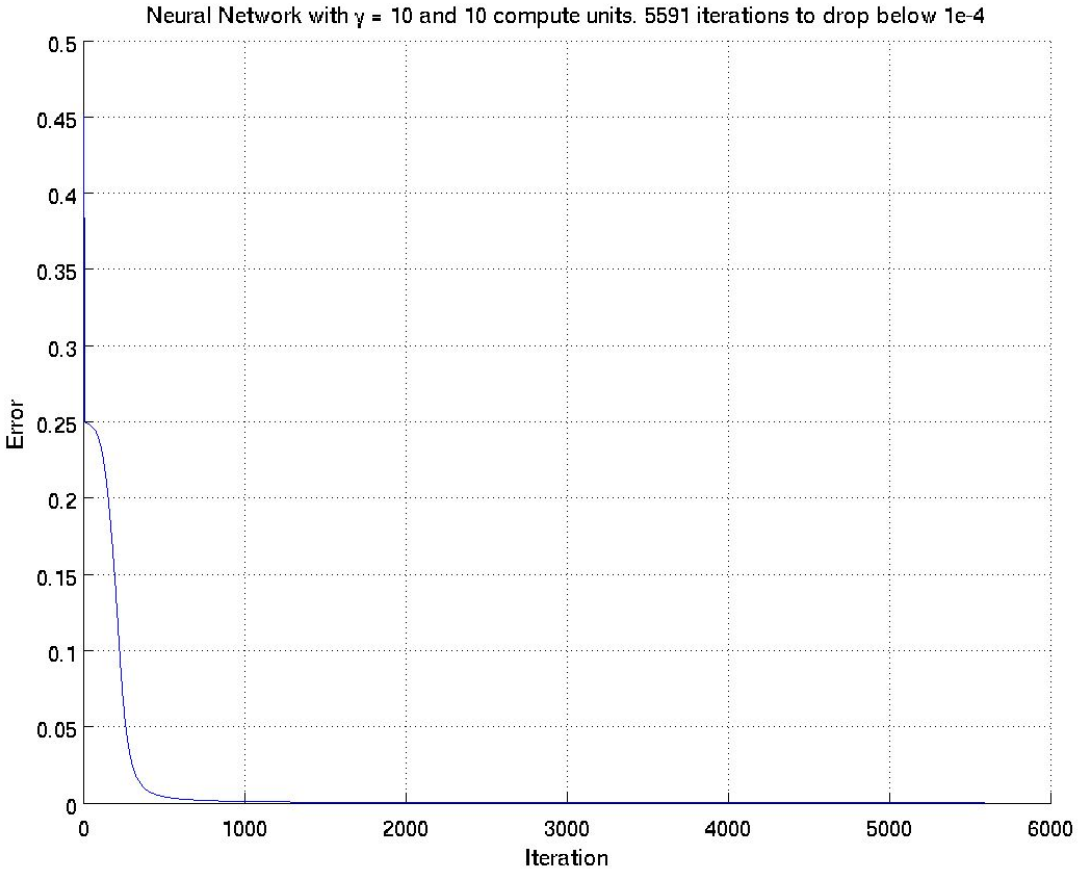

Neural Network with γ = 1 and 5 compute units. 8590 iterations to drop below 1e-4
Error
Iteration
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

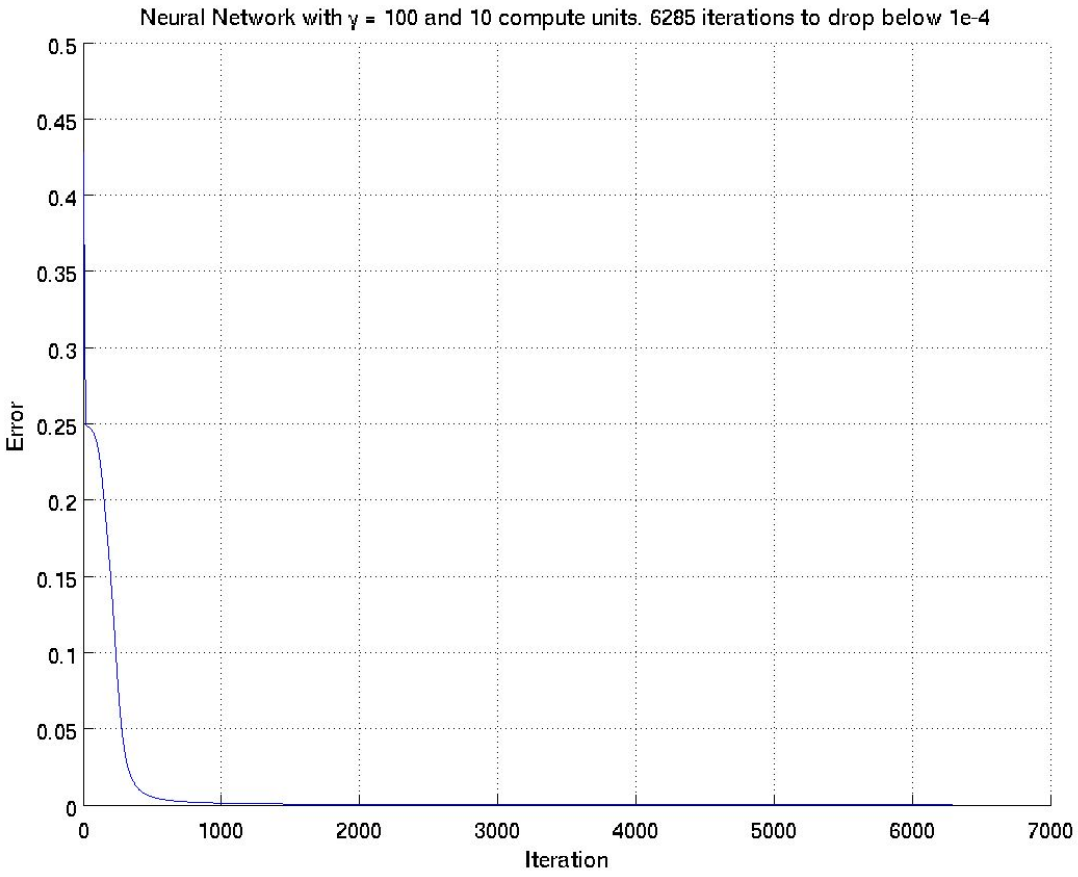

Neural Network with γ = 10 and 5 compute units. 8652 iterations to drop below 1e-4
Error
Iteration
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

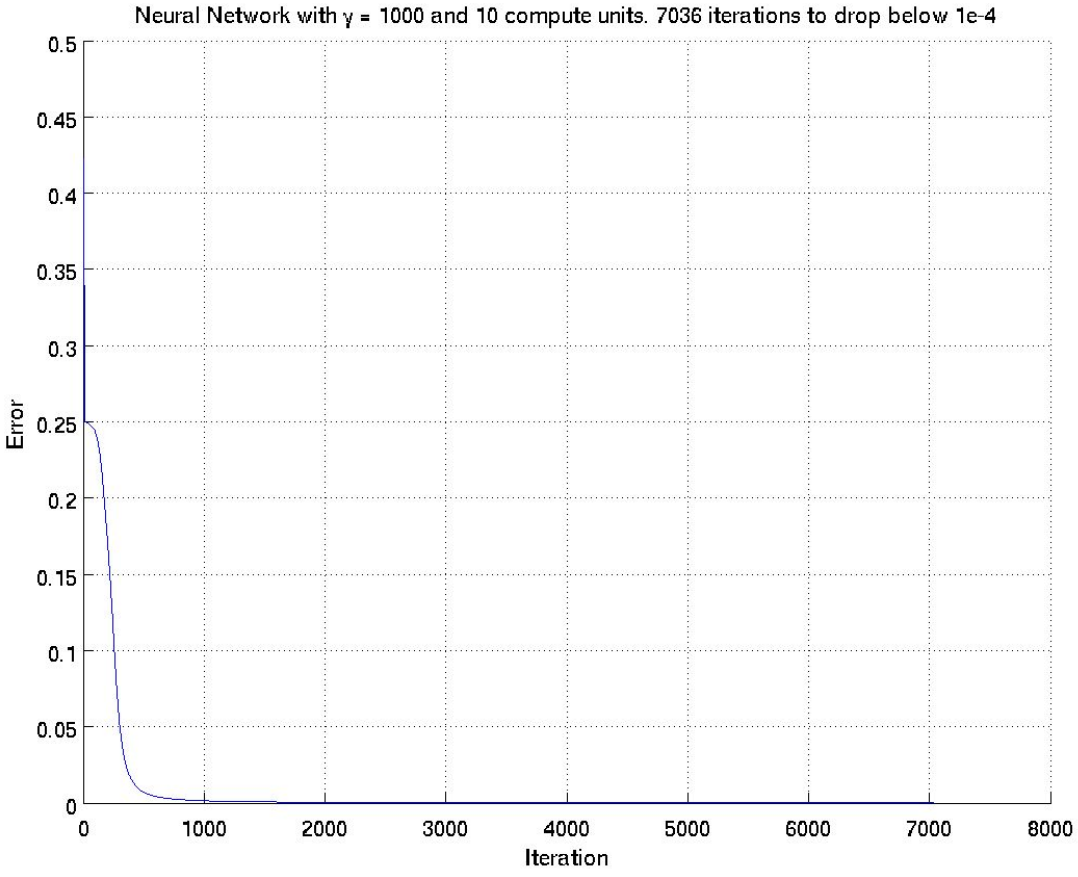

Neural Network with γ = 100 and 5 compute units. 7094 iterations to drop below 1e-4
Error
Iteration
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0
1000
2000
3000
4000
5000
6000
7000
8000


Neural Network with γ = 1000 and 5 compute units. 9182 iterations to drop below 1e-4
Error
Iteration
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Neural Network with γ = 1 and 10 compute units. 6625 iterations to drop below 1e-4
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1000
2000
3000
4000
5000
6000
7000

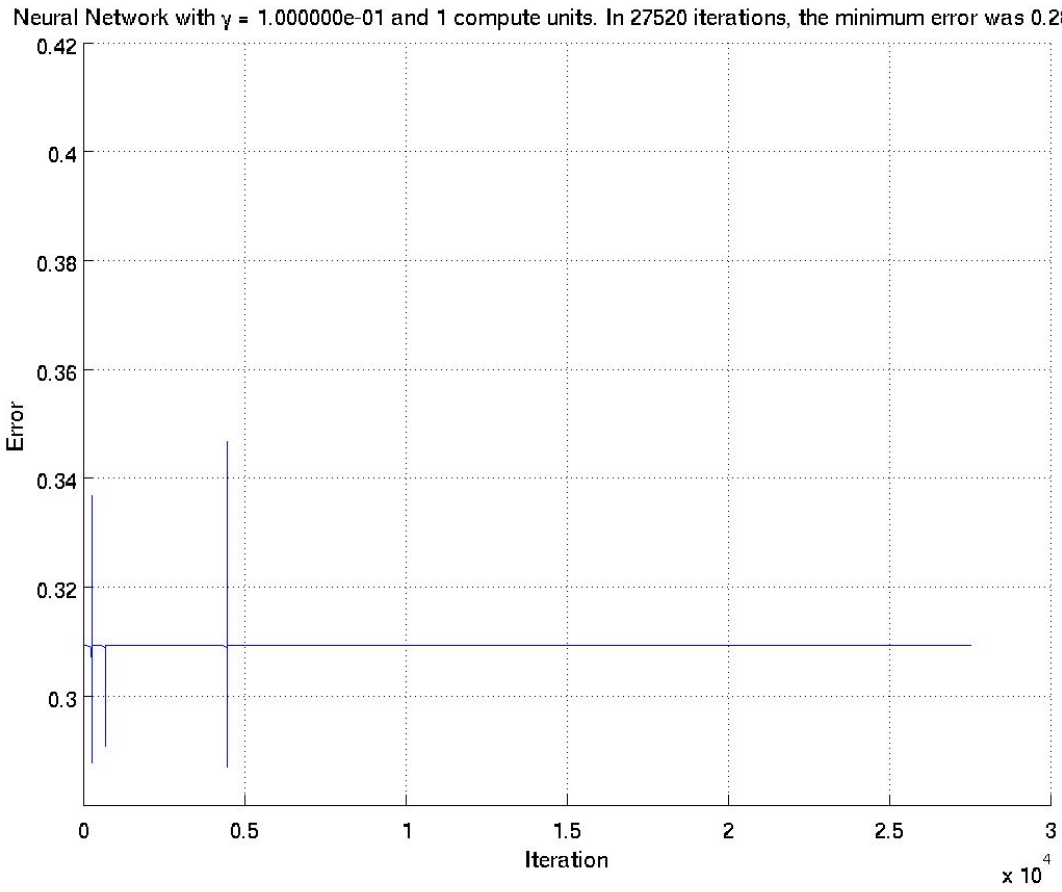

Neural Network with γ = 10 and 10 compute units. 5591 iterations to drop below 1e-4
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1000
2000
3000
4000
5000
6000

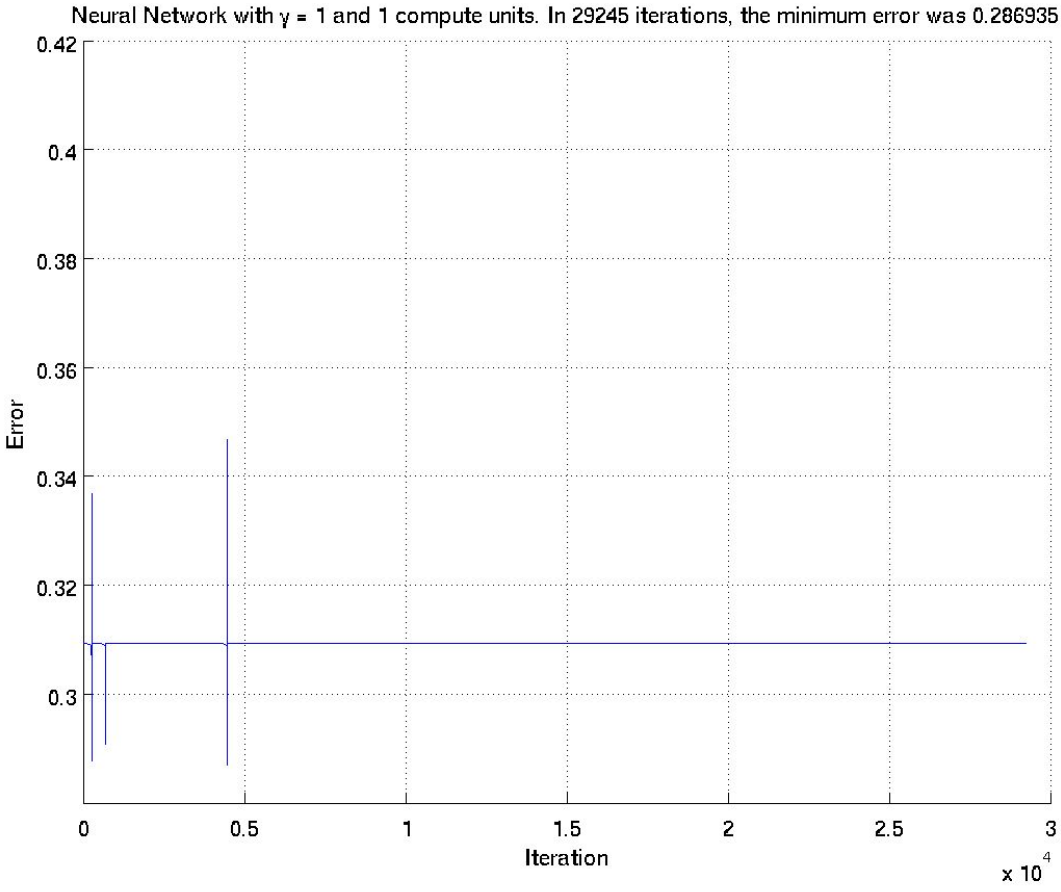

Neural Network with γ = 100 and 10 compute units. 6285 iterations to drop below 1e-4
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1000
2000
3000
4000
5000
6000
7000

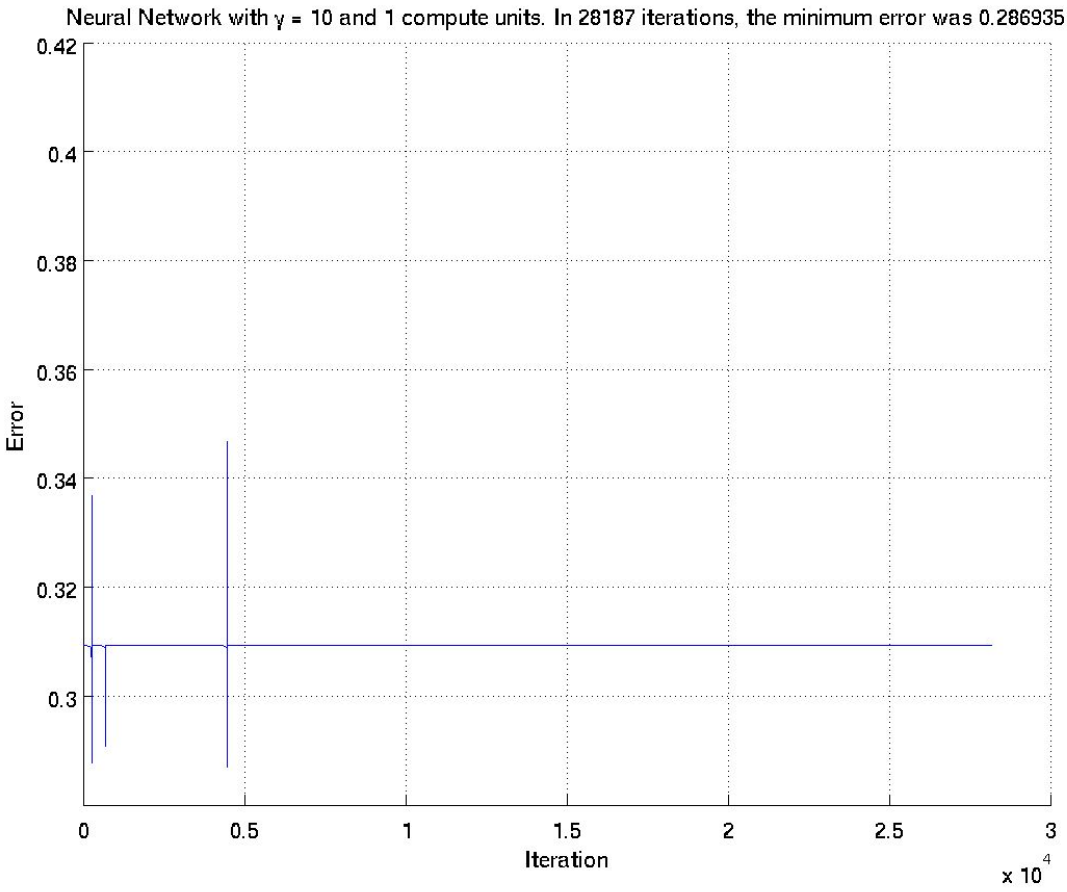

Neural Network with γ = 1000 and 10 compute units. 7036 iterations to drop below 1e-4
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1000
2000
3000
4000
5000
6000
7000
8000

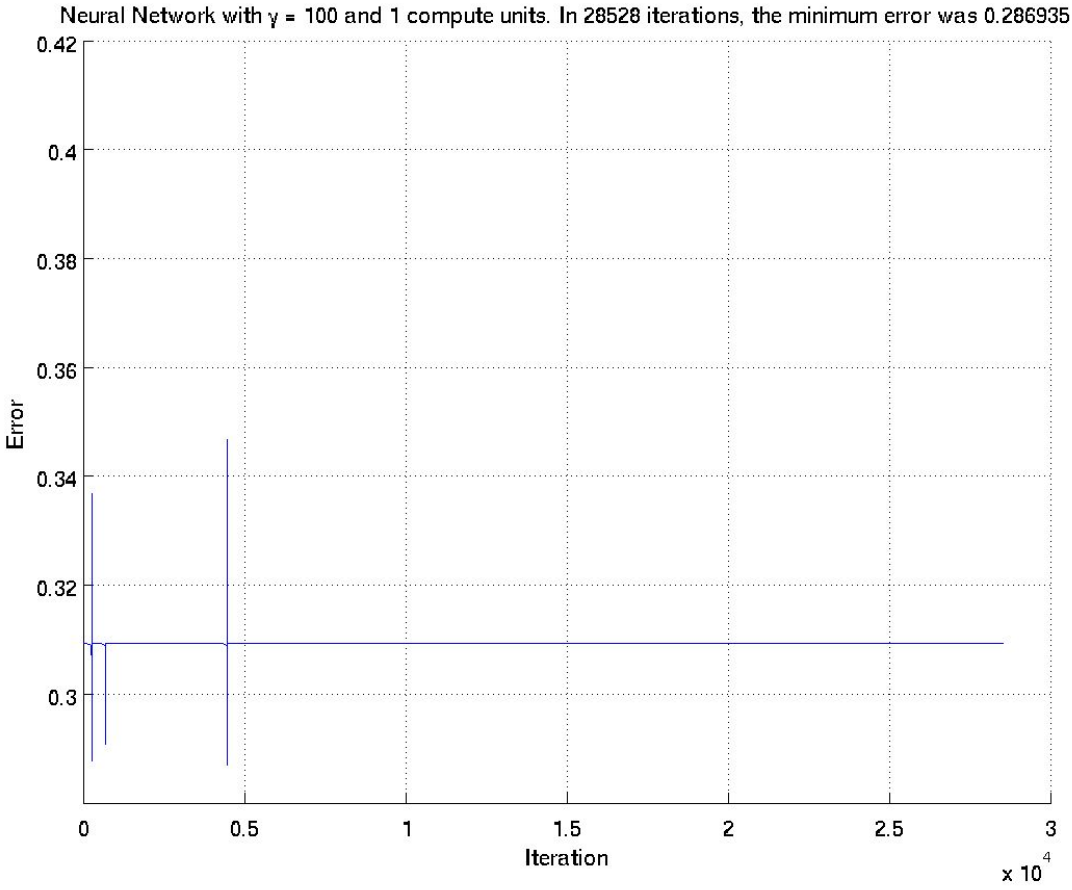
The prostate data set contains 97 data samples. Of those samples, there are 67 “has cancer” samples and 30 “does not have cancer” samples. The neural network is trained on all the samples. Similar charts are provided for this data set. It is noted that, without implementing a proper method to adjust 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.

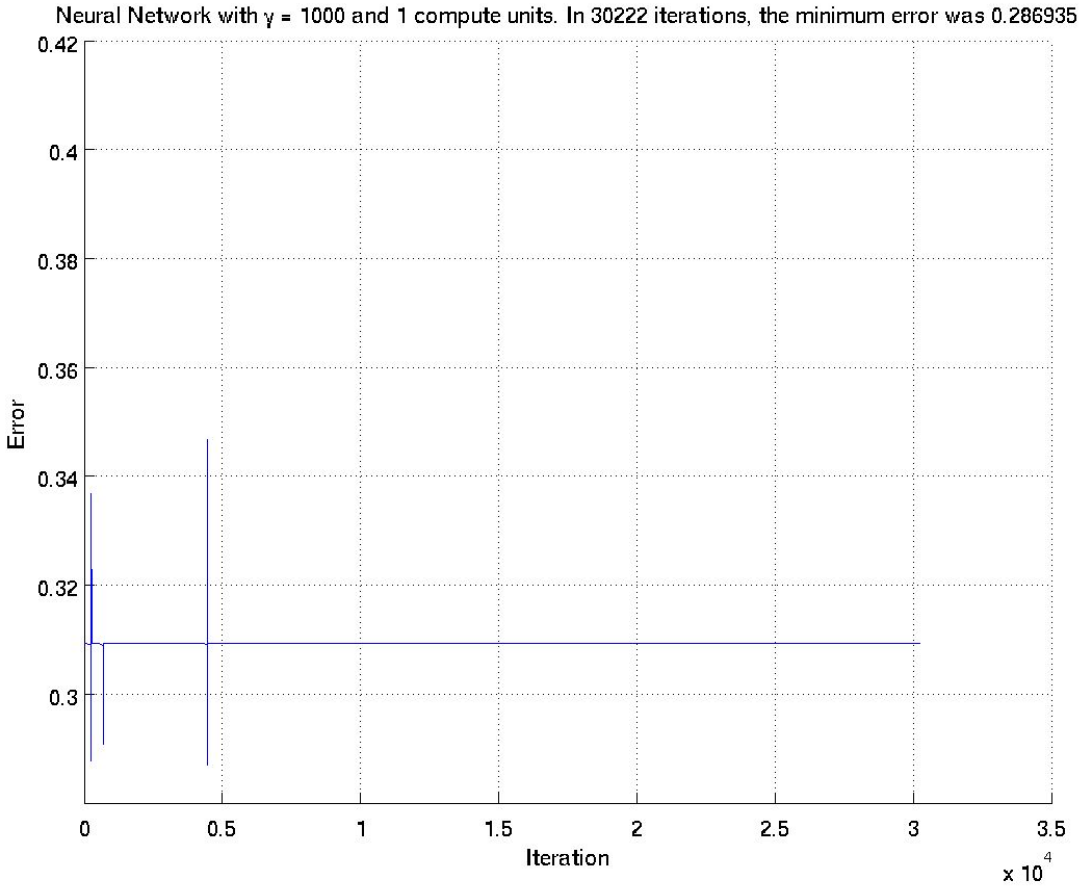
The minimum error computed was 0.2604 for gamma 1000 and 10 compute units, during 20905 iterations. The minimum was actually found early in the data set, however due to the gamma issues described above the minimum was bypassed and the neural network became essentially stuck on a plateau of the sigmoid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

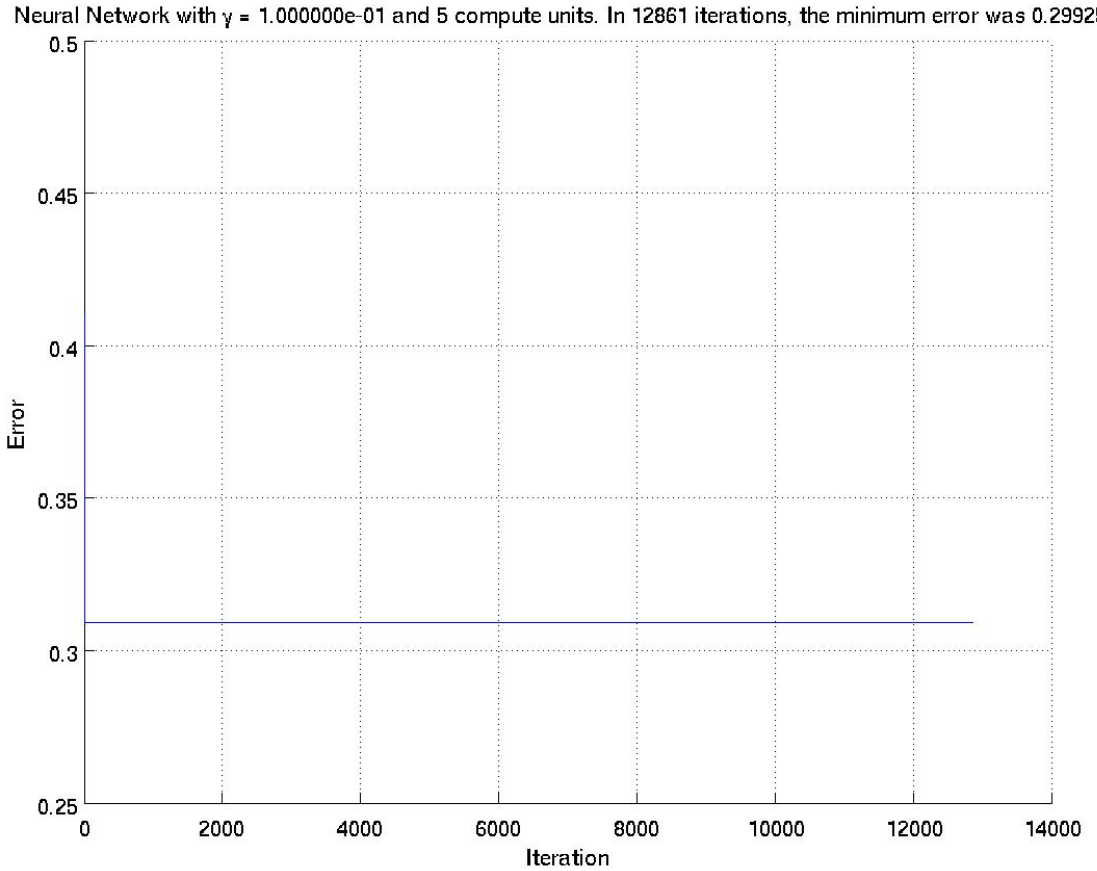

Neural Network with γ = 1.000000e-01 and 1 compute units. In 27520 iterations, the minimum error was 0.286935
Error
Iteration
x 10^4
0.42
0.4
0.38
0.36
0.34
0.32
0.3
0
0.5
1
1.5
2
2.5
3

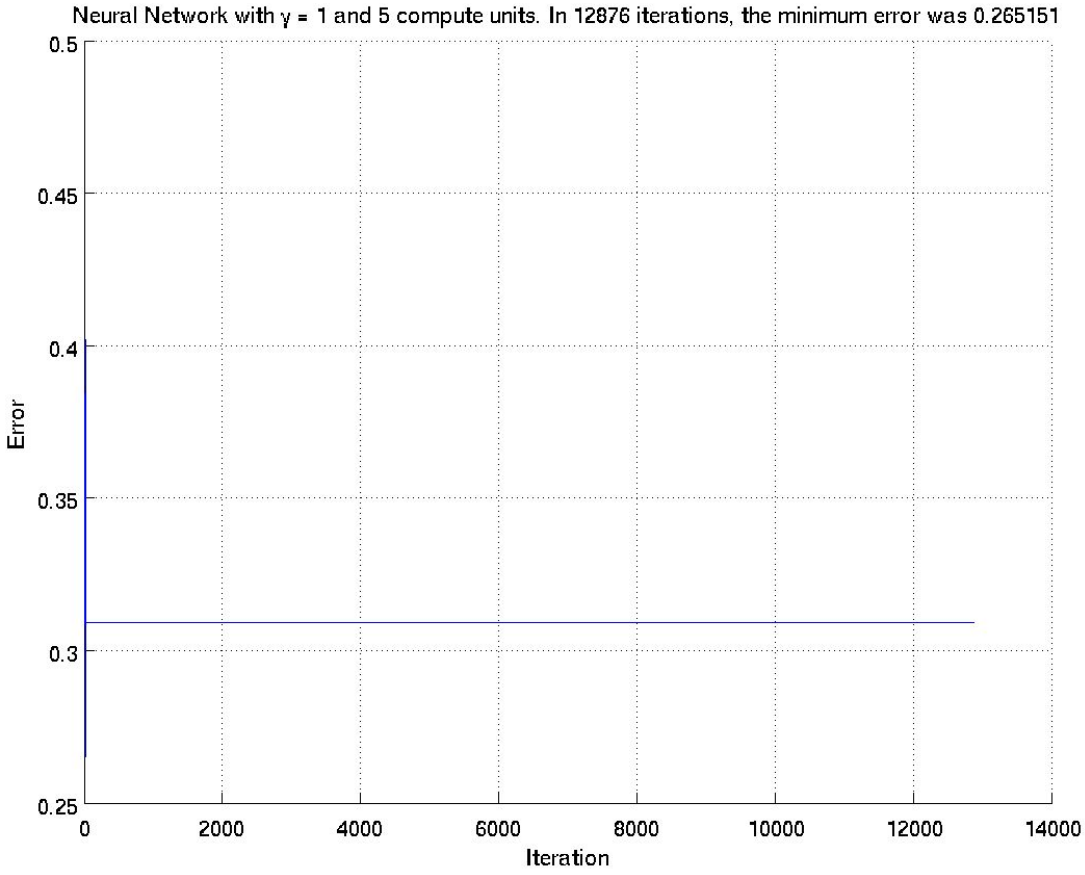

Neural Network with γ = 1 and 1 compute units. In 29245 iterations, the minimum error was 0.286935
Error
Iteration
x 10^4
0.42
0.4
0.38
0.36
0.34
0.32
0.3
0
0.5
1
1.5
2
2.5
3

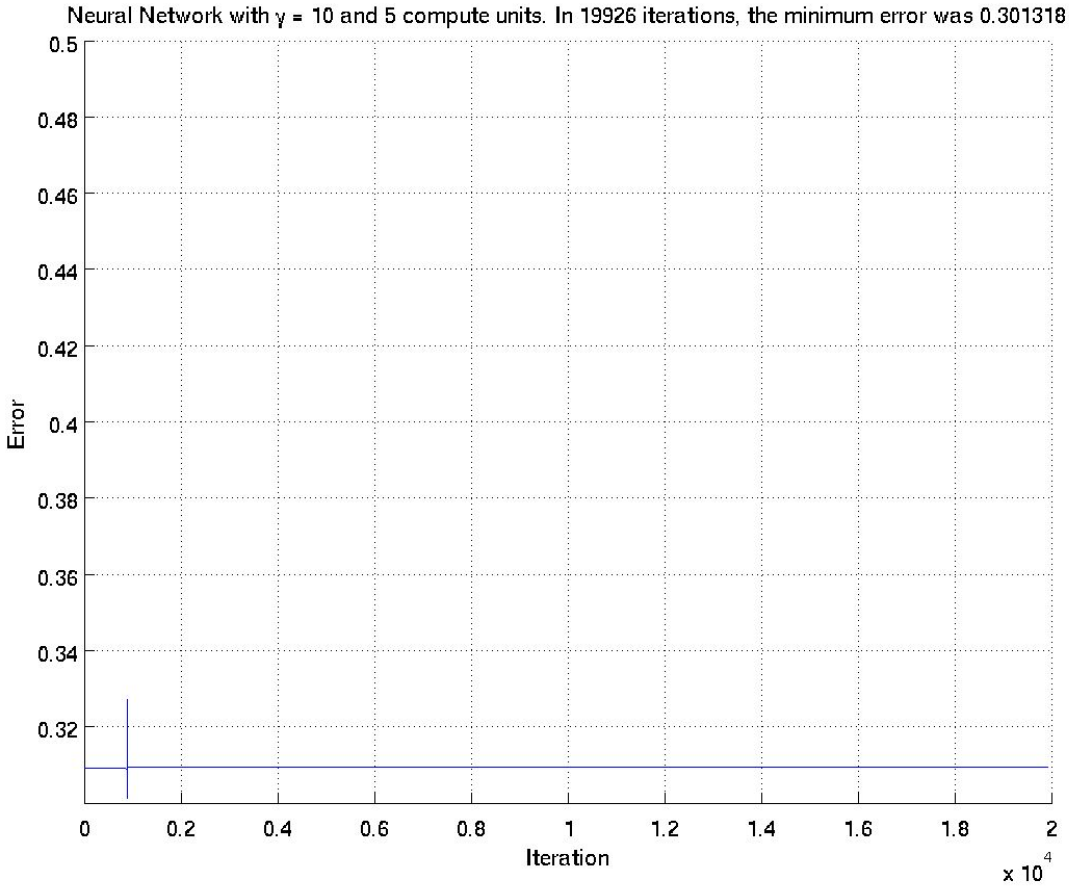

Neural Network with γ = 10 and 1 compute units. In 28187 iterations, the minimum error was 0.286935
Error
Iteration
x 10^4
0.42
0.4
0.38
0.36
0.34
0.32
0.3
0
0.5
1
1.5
2
2.5
3

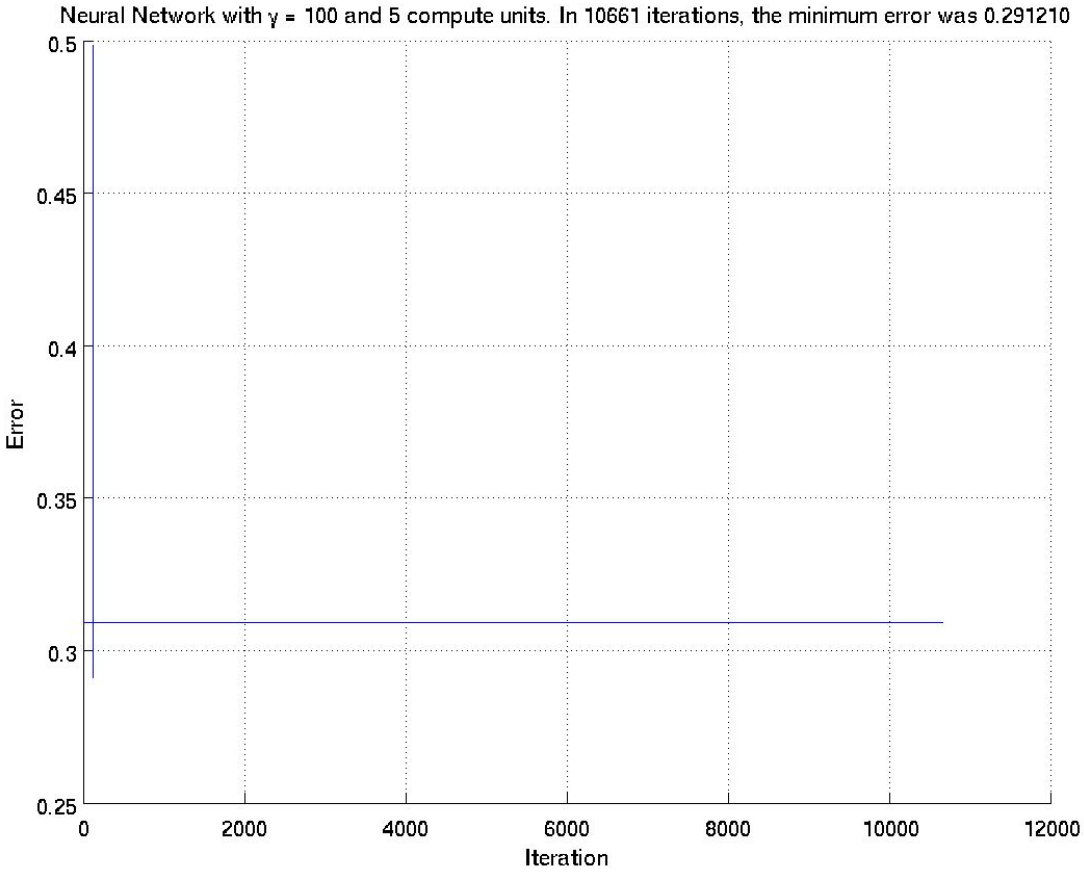

Neural Network with γ = 100 and 1 compute units. In 28528 iterations, the minimum error was 0.286935
Error
Iteration
x 10^4
0.42
0.4
0.38
0.36
0.34
0.32
0.3
0
0.5
1
1.5
2
2.5
3

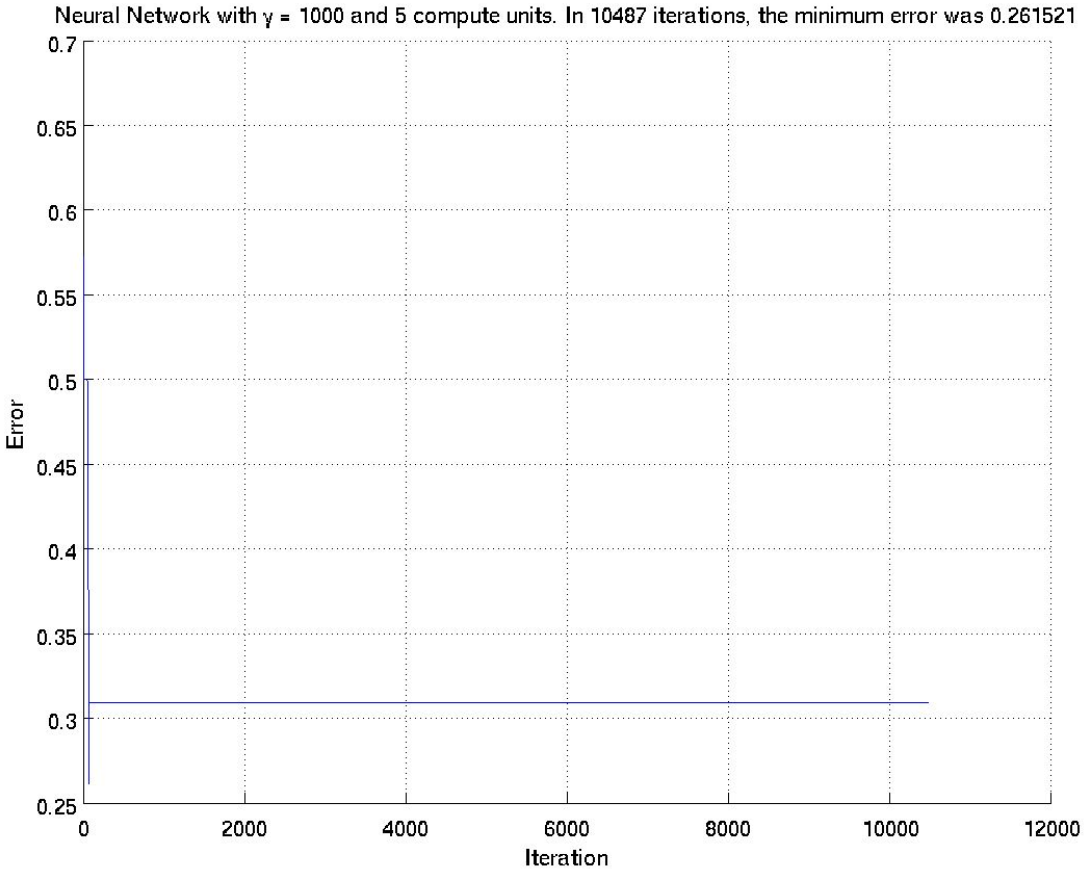

Neural Network with γ = 1000 and 1 compute units. In 30222 iterations, the minimum error was 0.286935
Error
Iteration
x 10^4
0.42
0.4
0.38
0.36
0.34
0.32
0.3
0
0.5
1
1.5
2
2.5
3
3.5

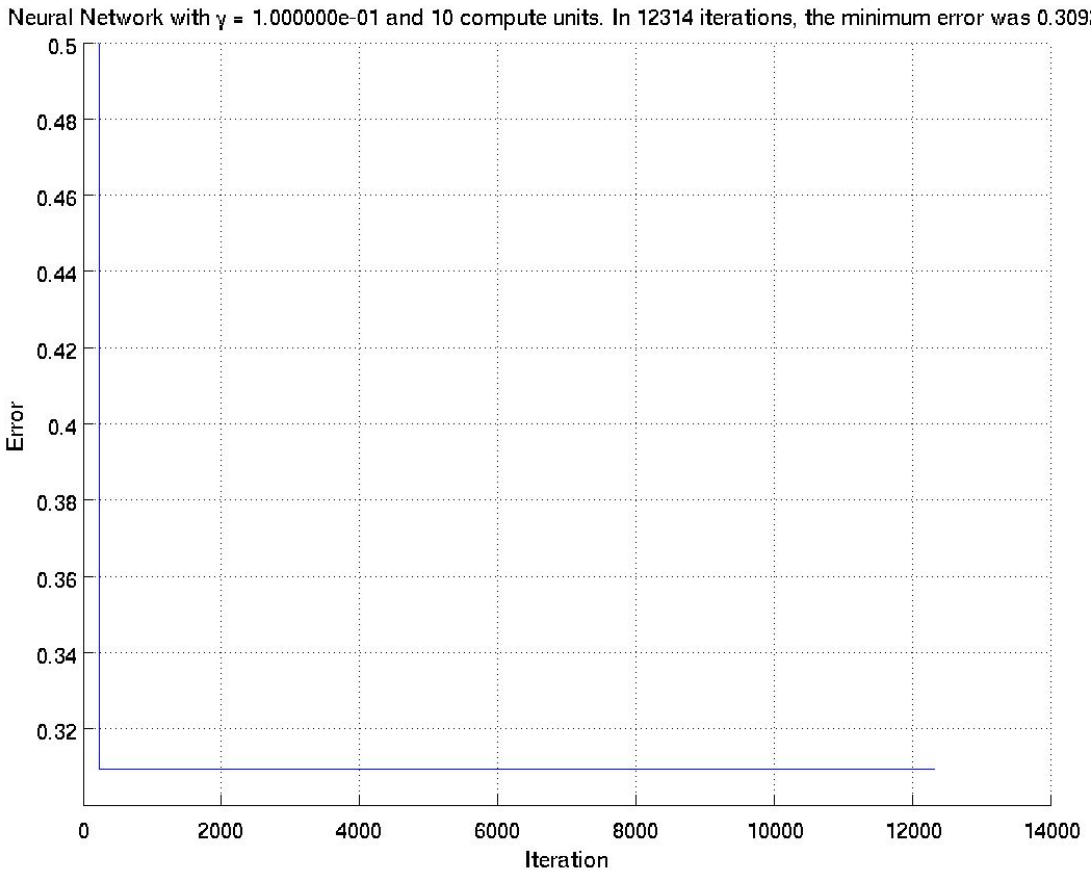

Neural Network with γ = 1.000000e-01 and 5 compute units. In 12861 iterations, the minimum error was 0.299253
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0
2000
4000
6000
8000
10000
12000
14000

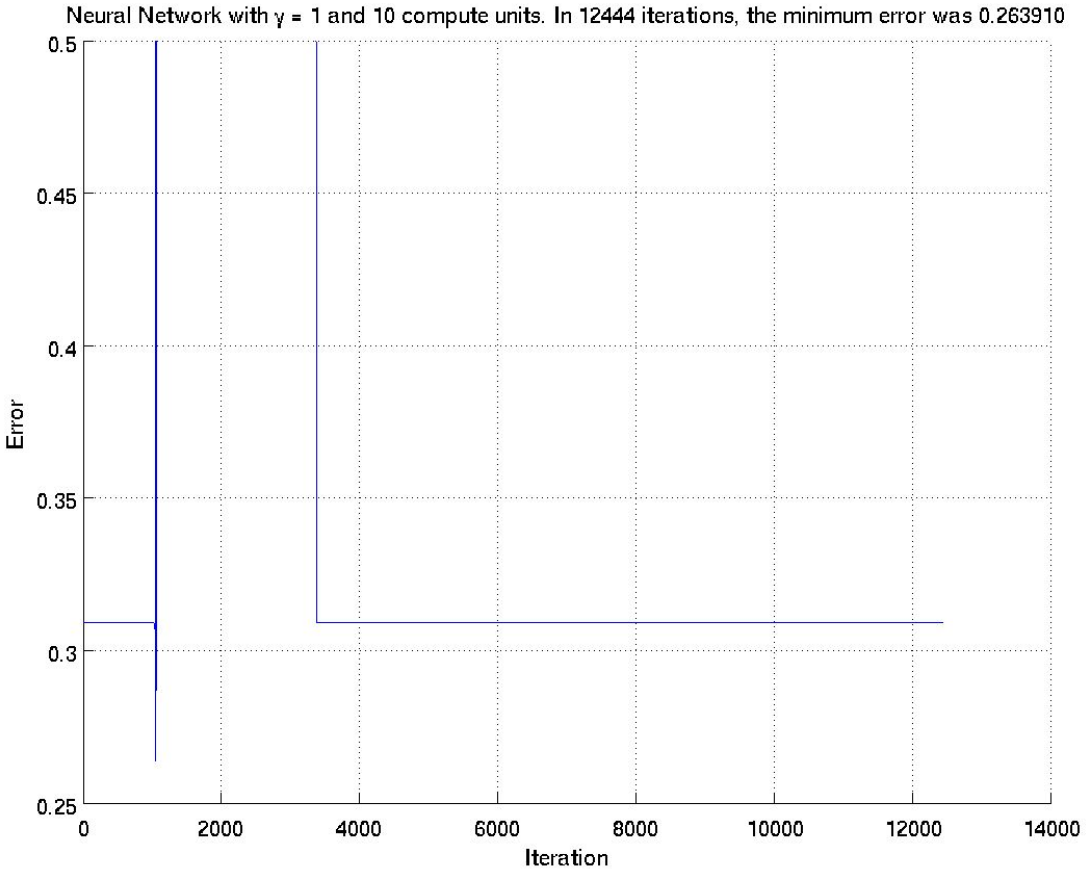

Neural Network with γ = 1 and 5 compute units. In 12876 iterations, the minimum error was 0.265151
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0
2000
4000
6000
8000
10000
12000
14000

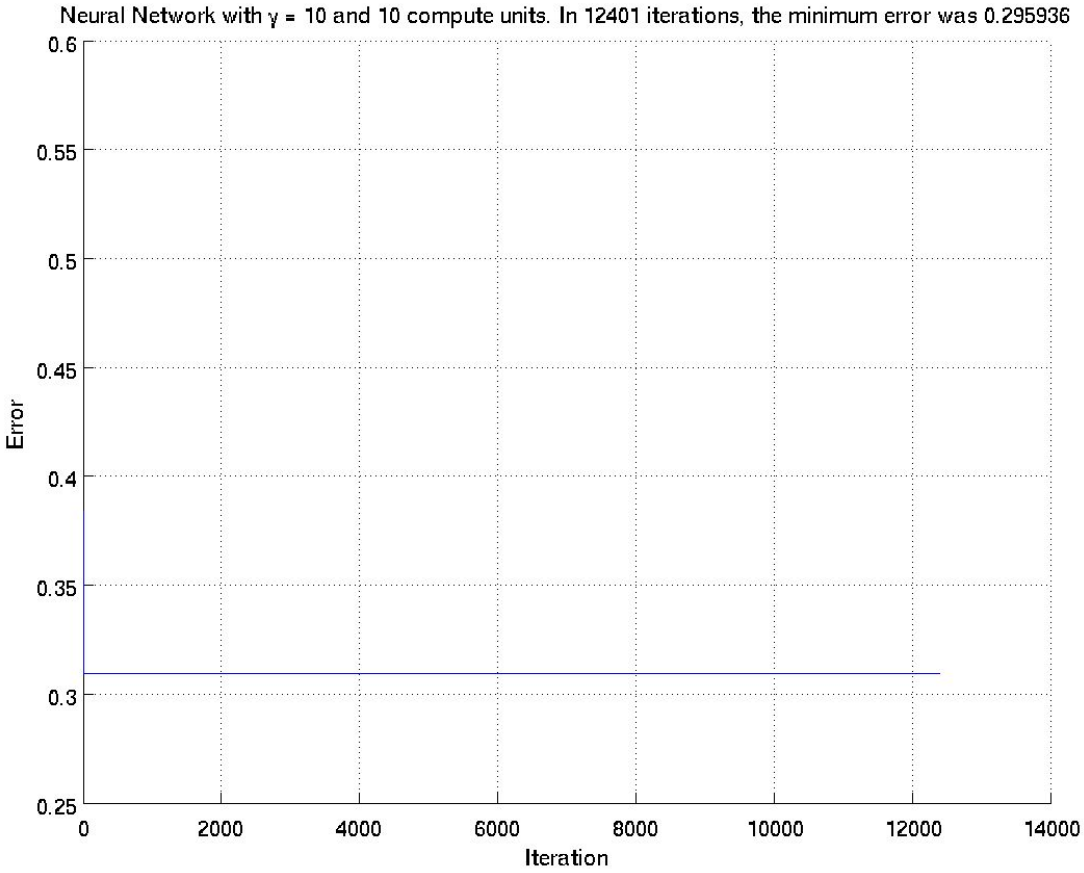

Neural Network with γ = 10 and 5 compute units. In 19926 iterations, the minimum error was 0.301318
Error
Iteration
x 10^4
0.5
0.48
0.46
0.44
0.42
0.4
0.38
0.36
0.34
0.32
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

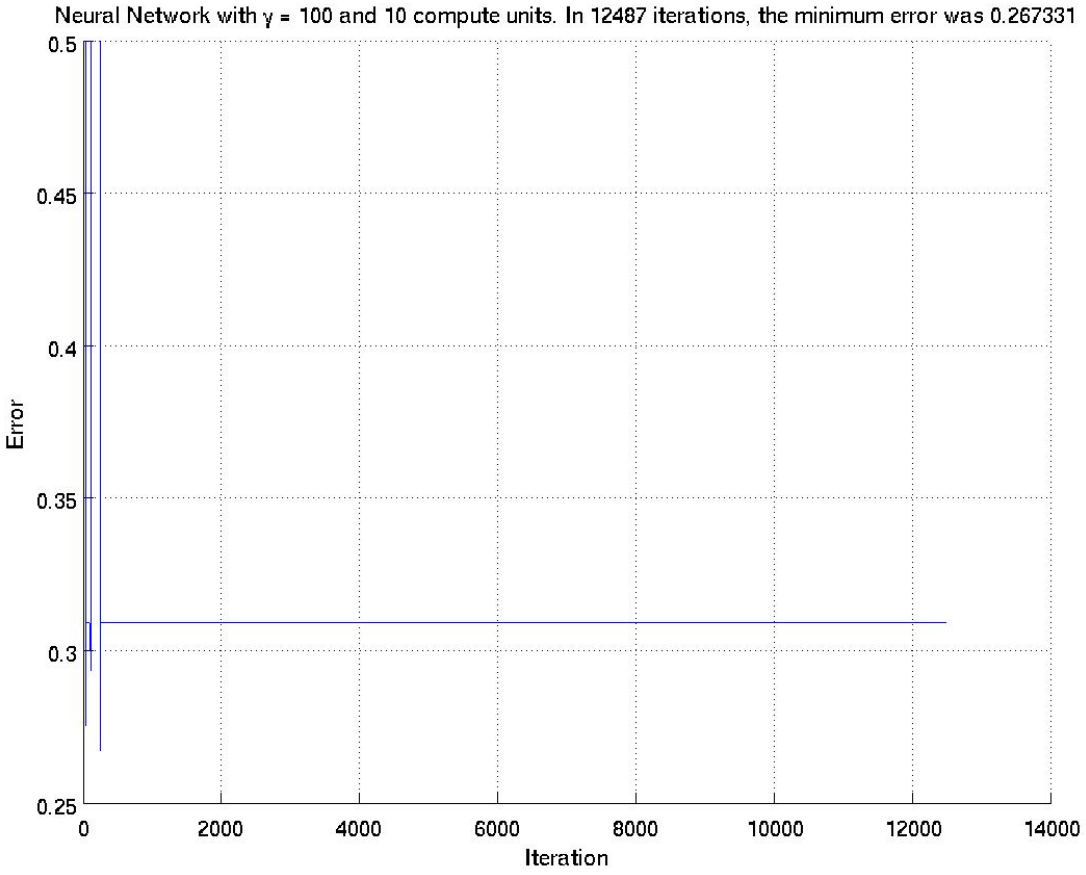

Neural Network with γ = 100 and 5 compute units. In 10661 iterations, the minimum error was 0.291210
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0
2000
4000
6000
8000
10000
12000

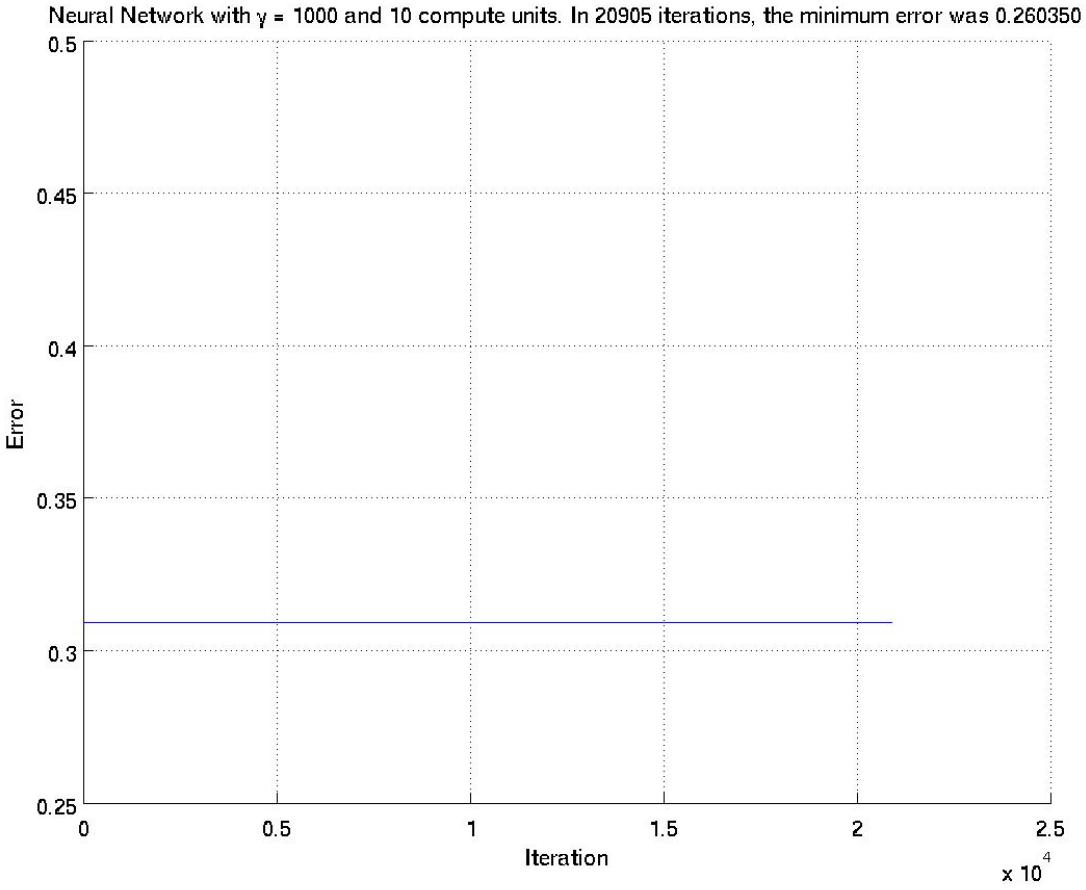

Neural Network with γ = 1000 and 5 compute units. In 10487 iterations, the minimum error was 0.261521
Error
Iteration
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0
2000
4000
6000
8000
10000
12000


Neural Network with γ = 1.000000e-01 and 10 compute units. In 12314 iterations, the minimum error was 0.309278
Error
Iteration
0.5
0.48
0.46
0.44
0.42
0.4
0.38
0.36
0.34
0.32
0
2000
4000
6000
8000
10000
12000
14000


Neural Network with γ = 1 and 10 compute units. In 12444 iterations, the minimum error was 0.263910
Error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0
2000
4000
6000
8000
10000
12000
14000


Neural Network with γ = 10 and 10 compute units. In 12401 iterations, the minimum error was 0.295936
Error
Iteration
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0
2000
4000
6000
8000
10000
12000
14000





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, and 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.