CSCI 4100 Assignment 12

Boliang Yang 661541863

1. Neural Networks and Backpropagation
2. For identity:

The forward propagation is

The backpropagation is ,

,

For sigmoid transformation:

The forward propagation is

The backpropagation is ,

,

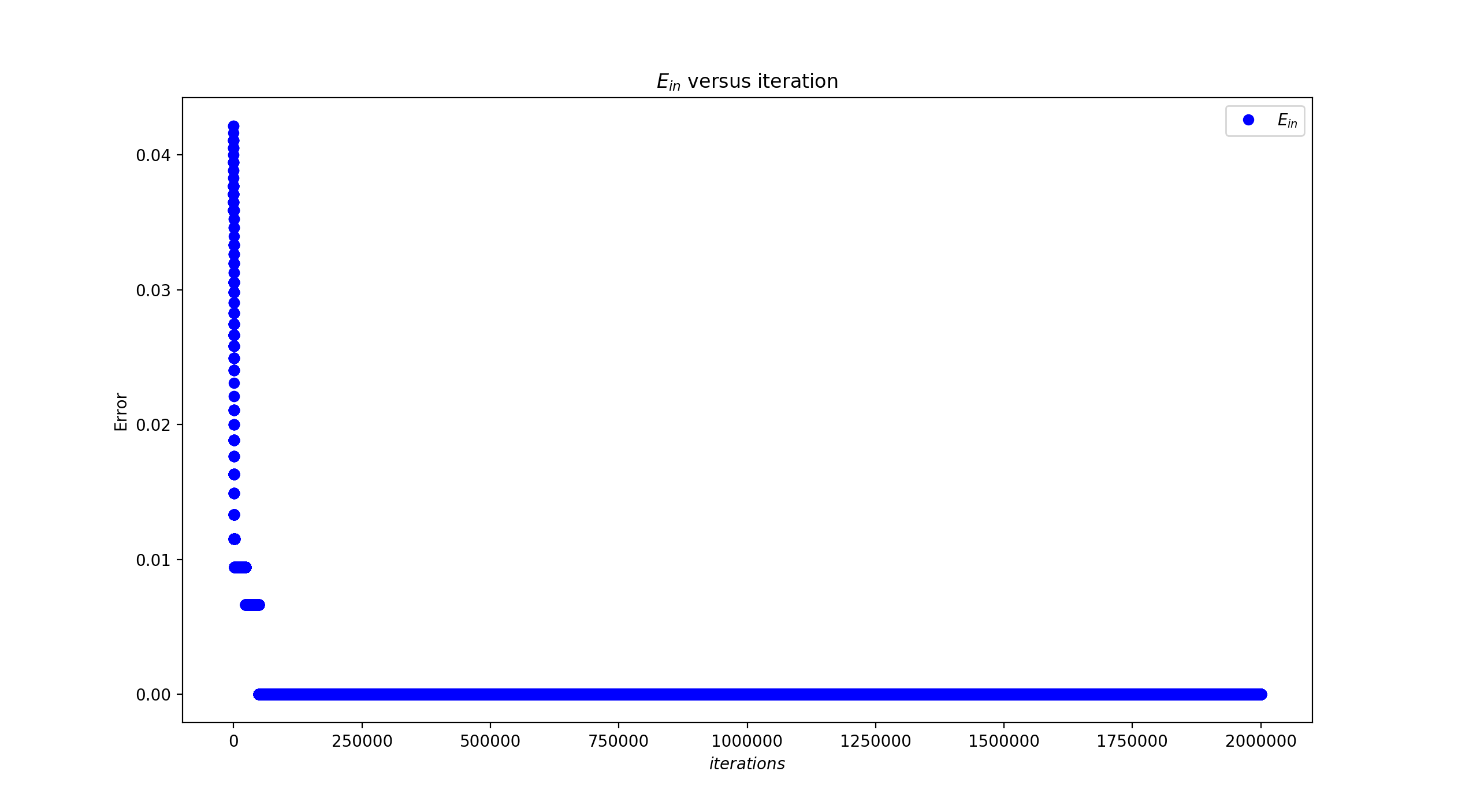
1. For identity:

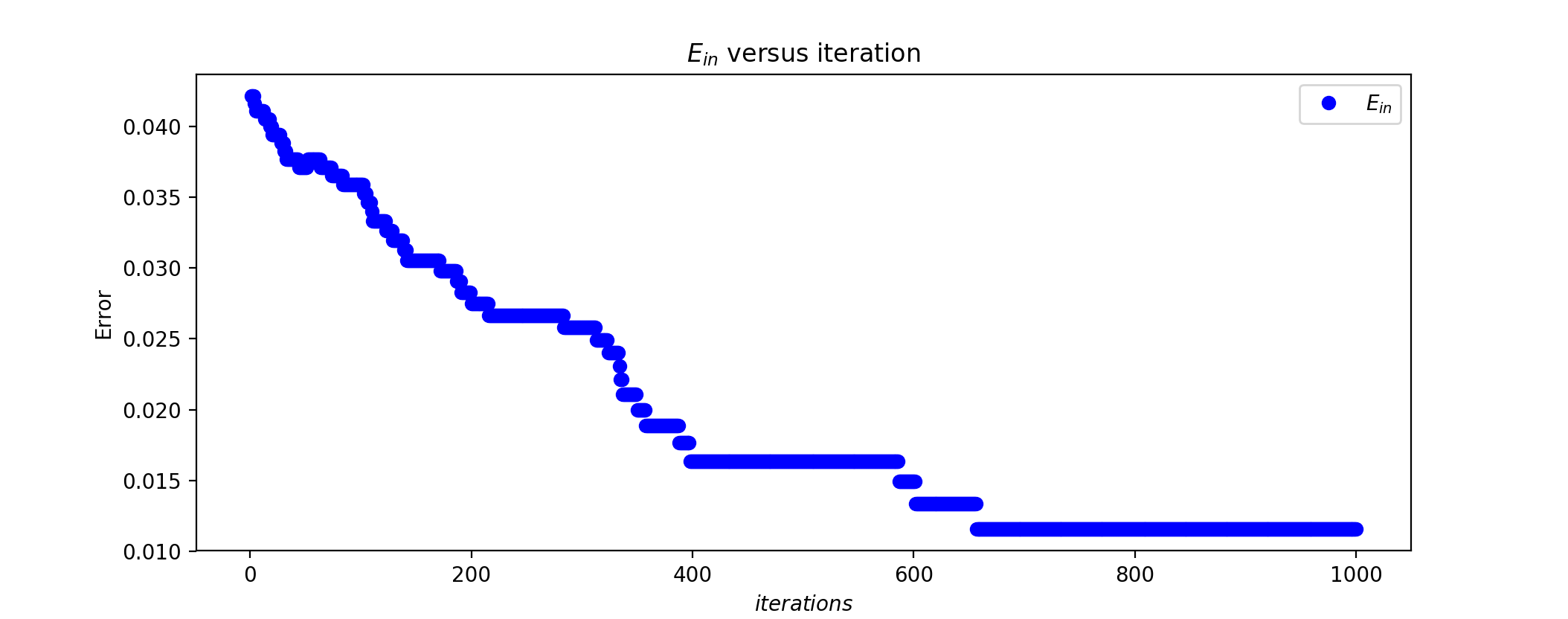
,

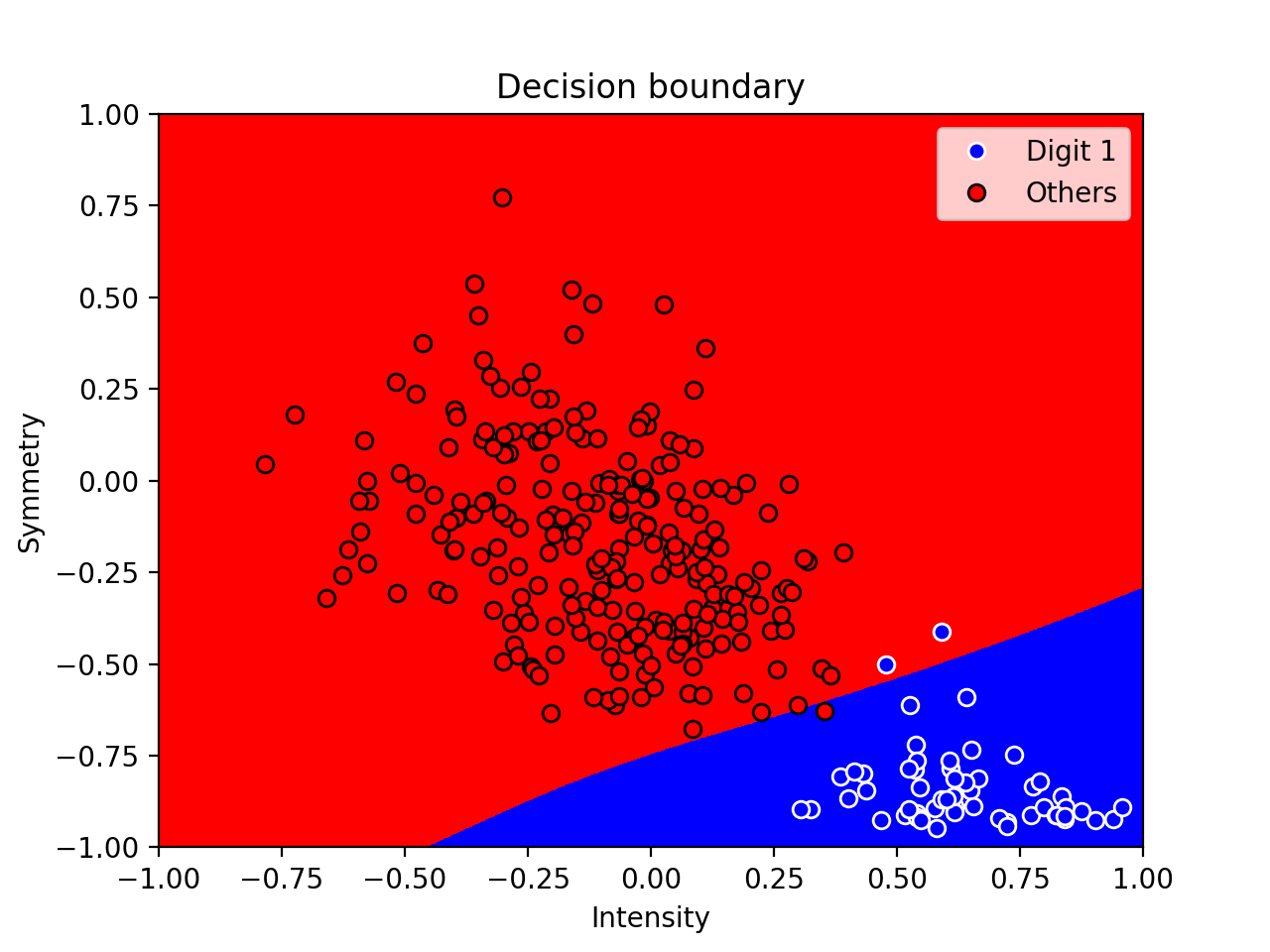
For sigmoid transformation:

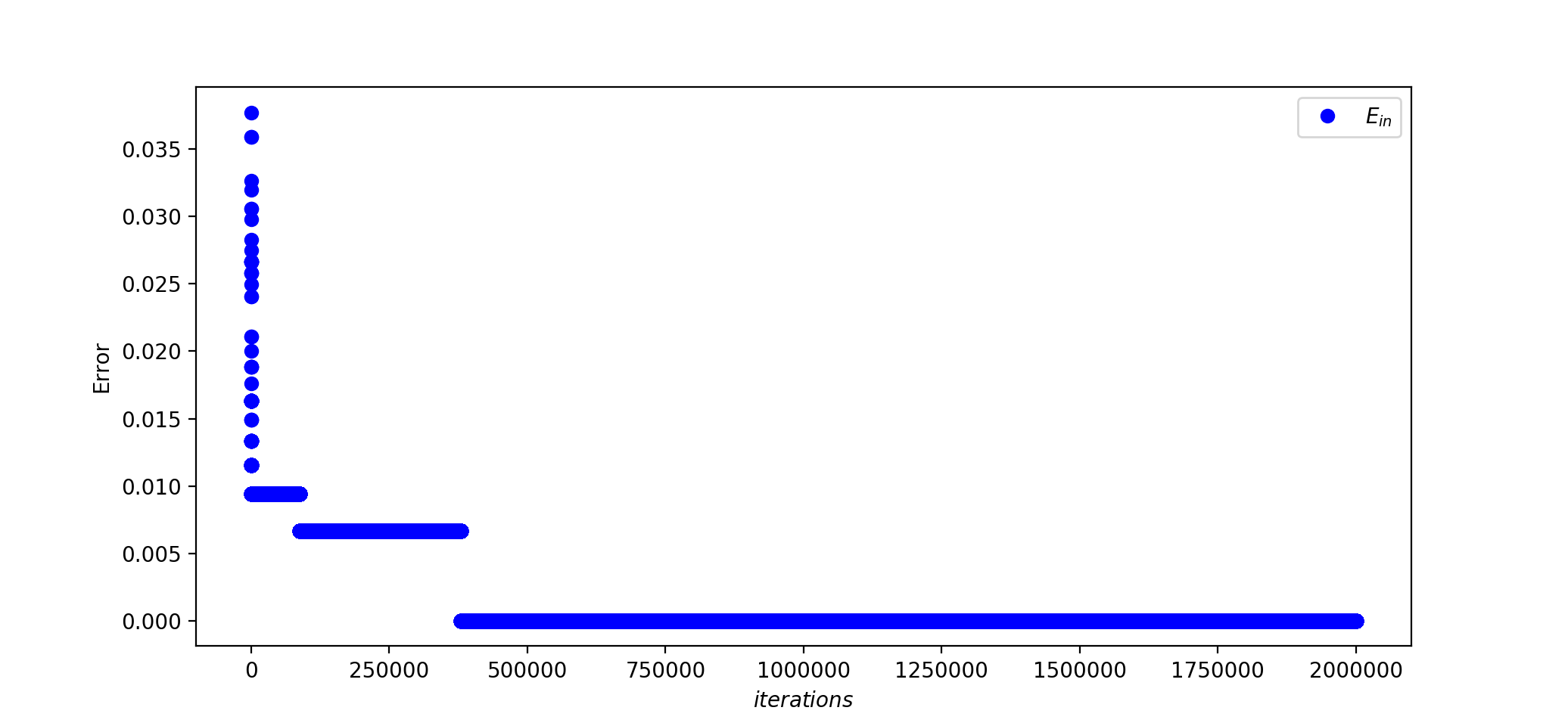
,

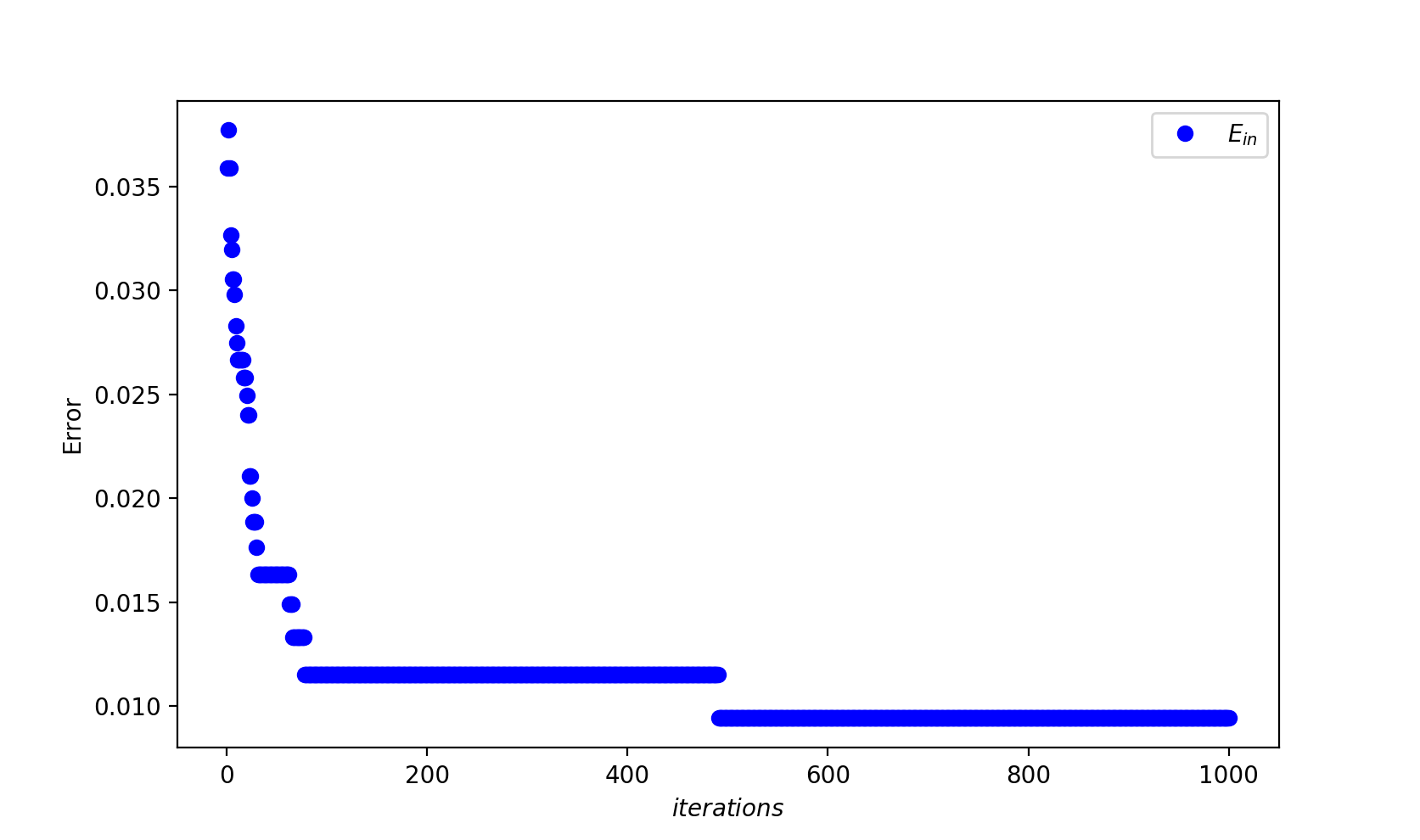
The results are the same as the previous results.

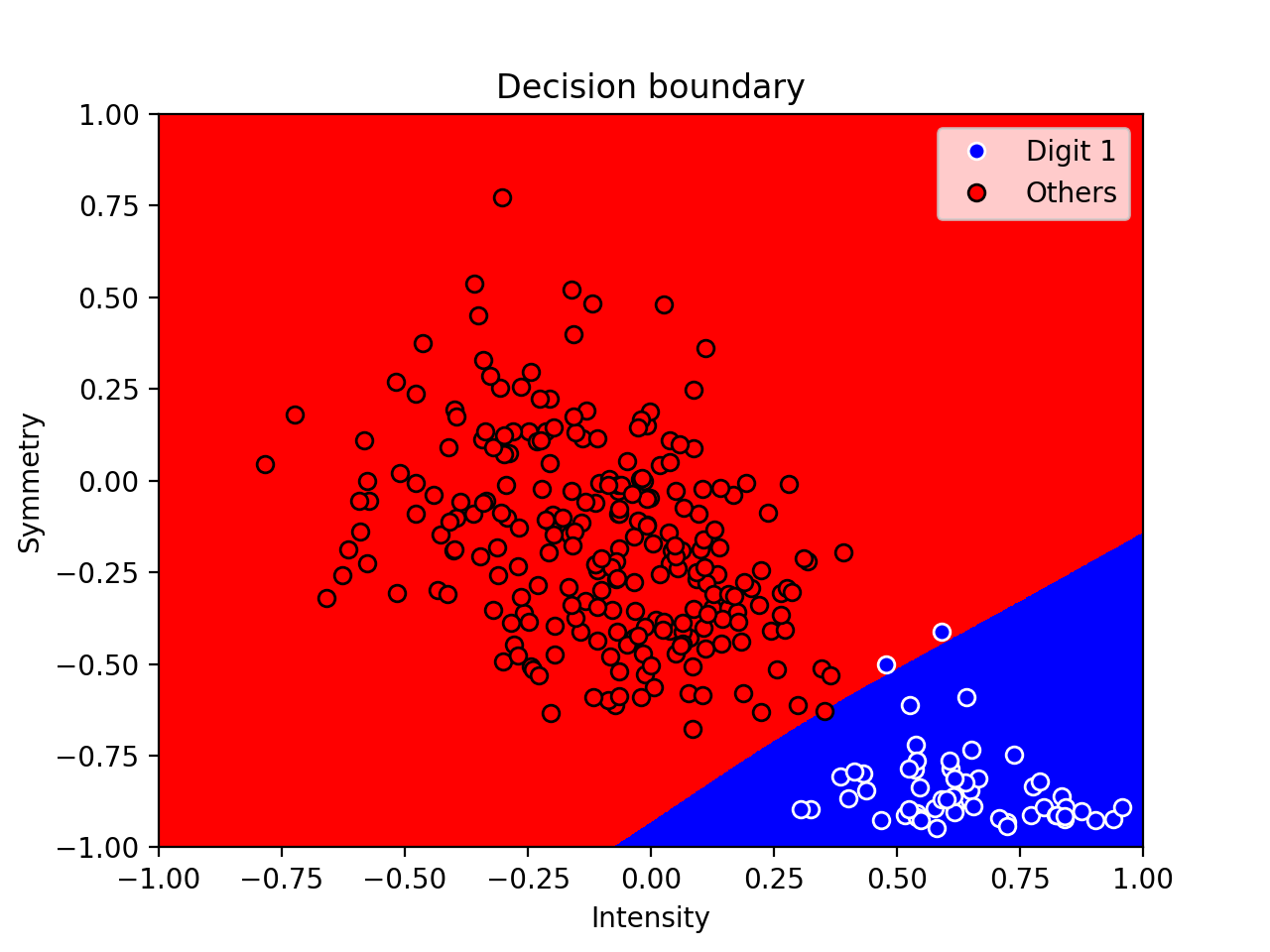
1. Neural Network for Digits
2. The plot for iterations:

And here is plot for 1000 iterations: it converges at about 650 iterations.

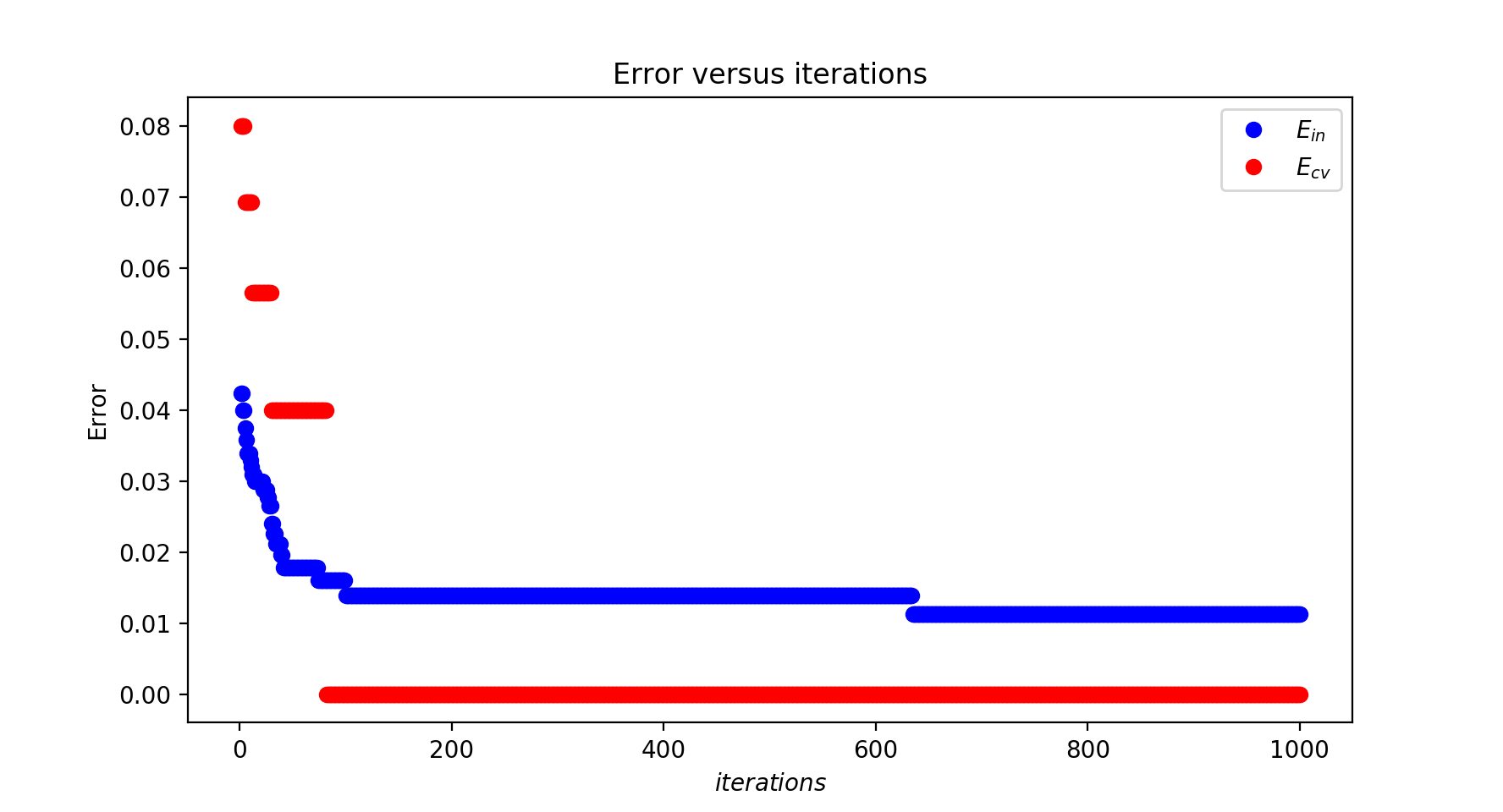
The plot for decision boundary for the resulting classifier:

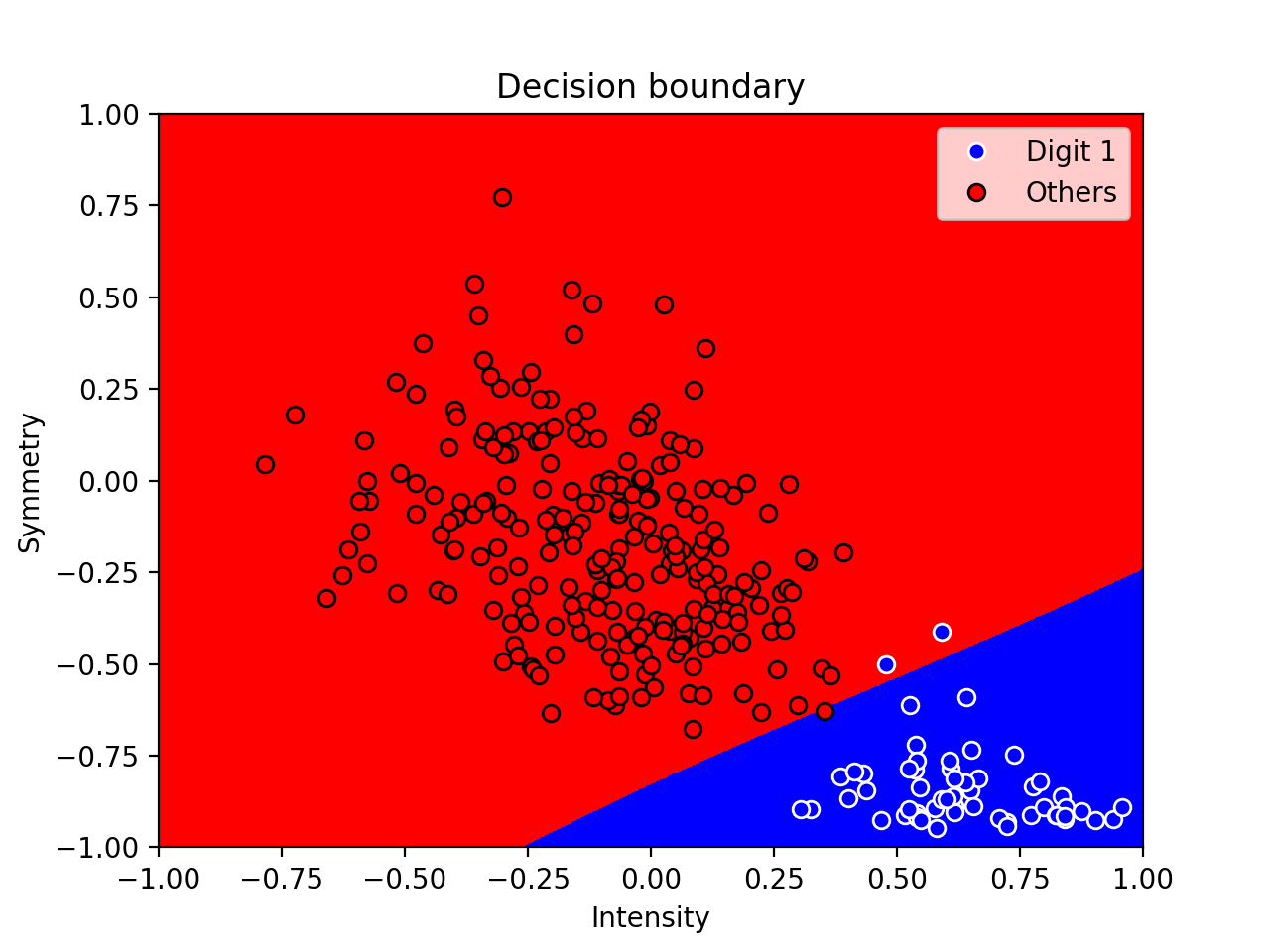
1. With weight decay, here is the plot for iterations:

The plot for 1000 iterations: it converges at about 500 iterations.



The plot for decision boundary for the resulting classifier:

1. Use early stopping with validation, here is the plot: the minimum validation error is at about 50 iterations.



The plot for decision boundary for the resulting classifier:

1. Support Vector Machines
2. We solve optimization problem in order to get optimal separating hyperplane:

subject to:

Since , then the problem becomes:

subject to: and

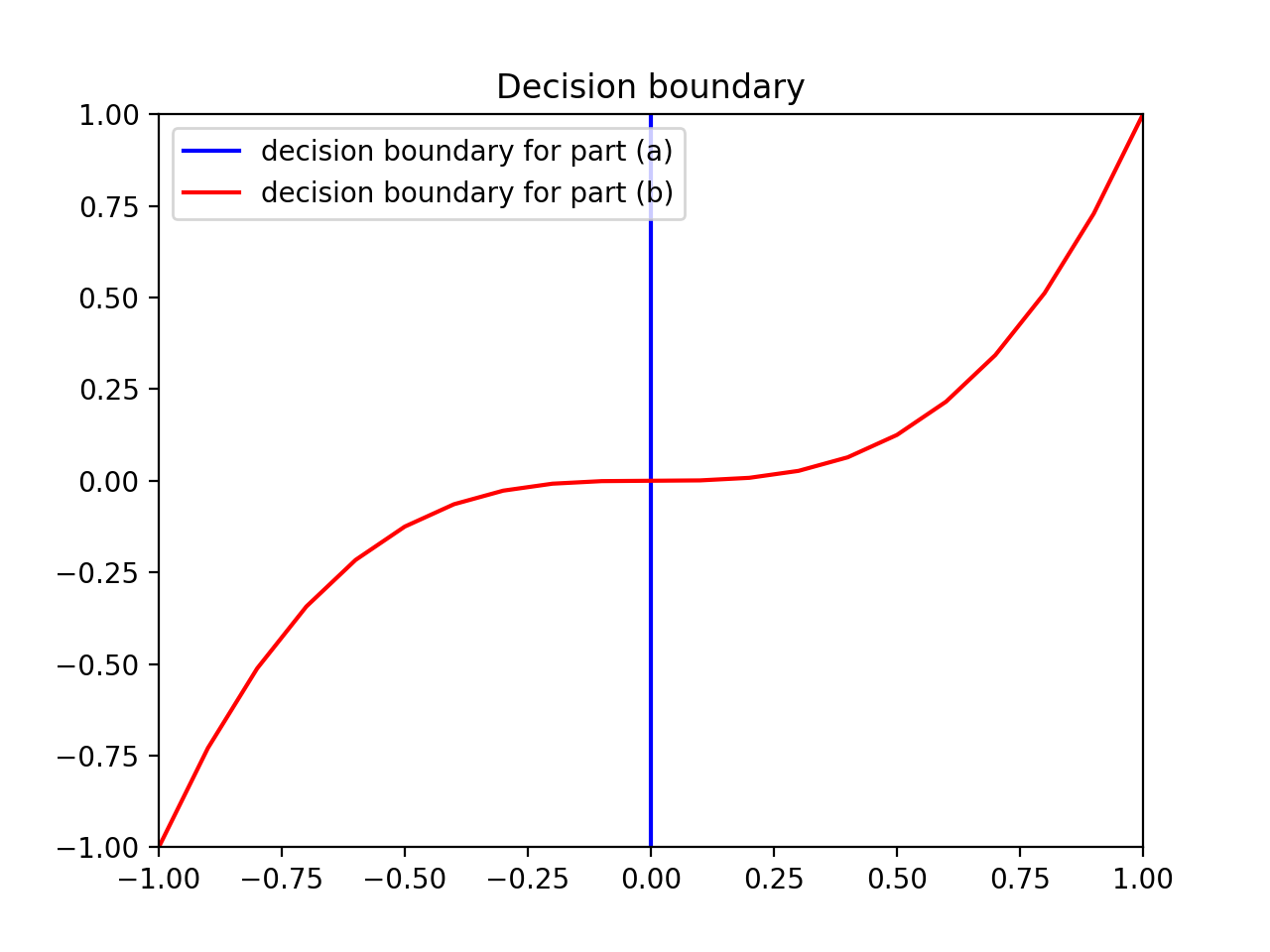
We get . Thus, the optimal hyperplane is . Since the equation for plane is , and the line segment equation is , then the hyperplane is perpendicular to the line segment.

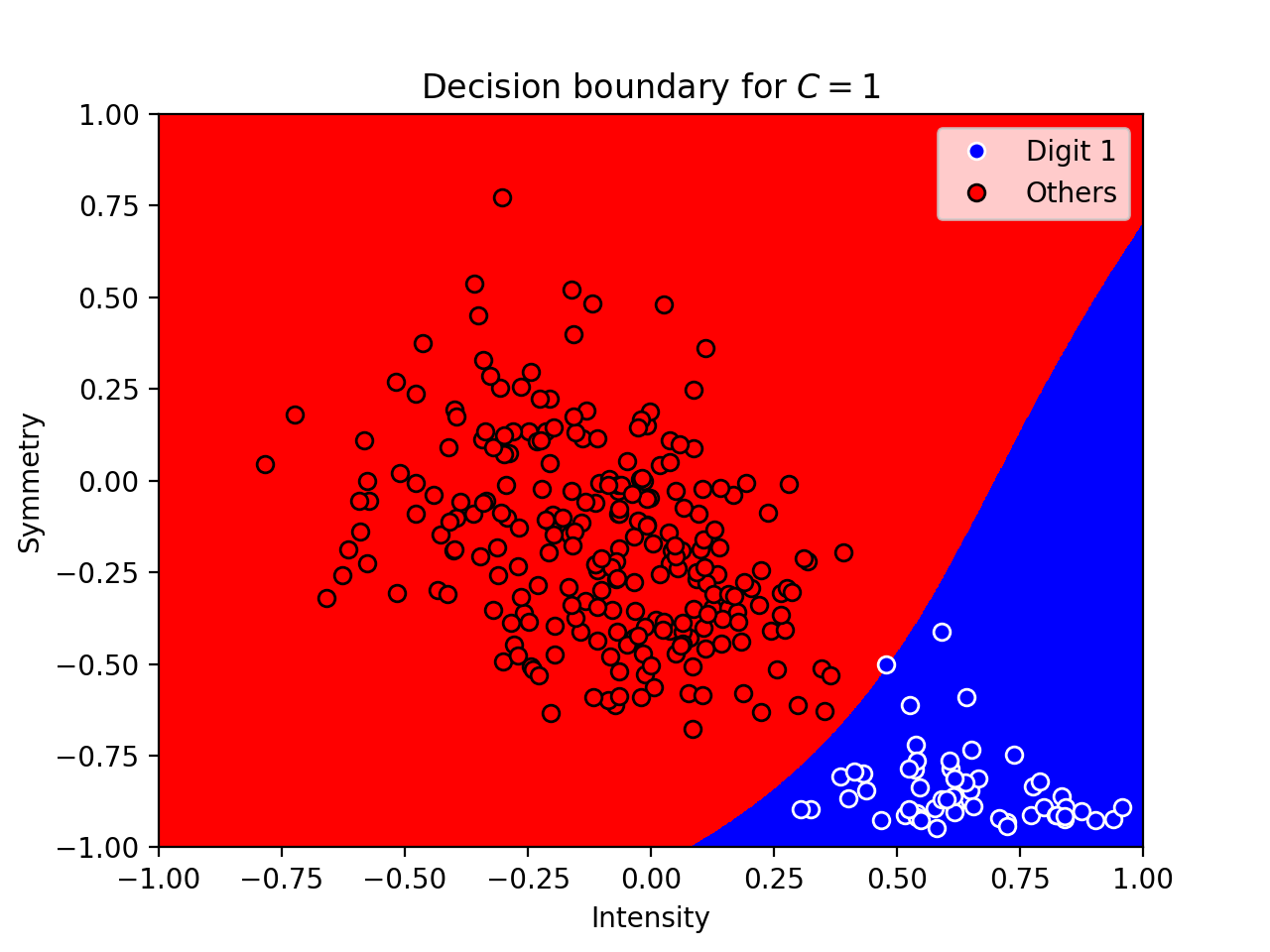
1. i. For , we have ;

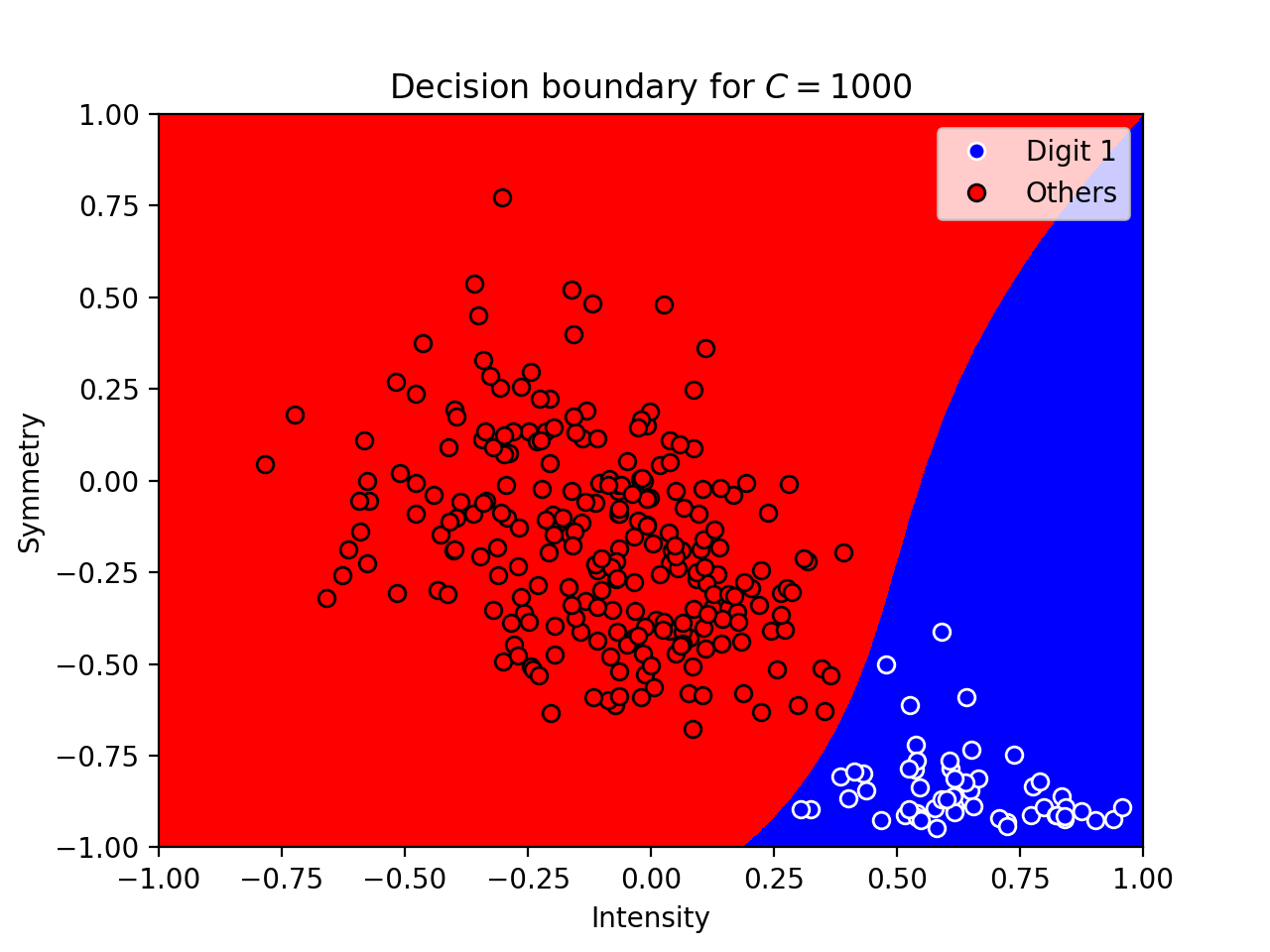
For , we have .

ii. We need to solve

subject to: and

 We get . Thus, the optimal hyperplane is .

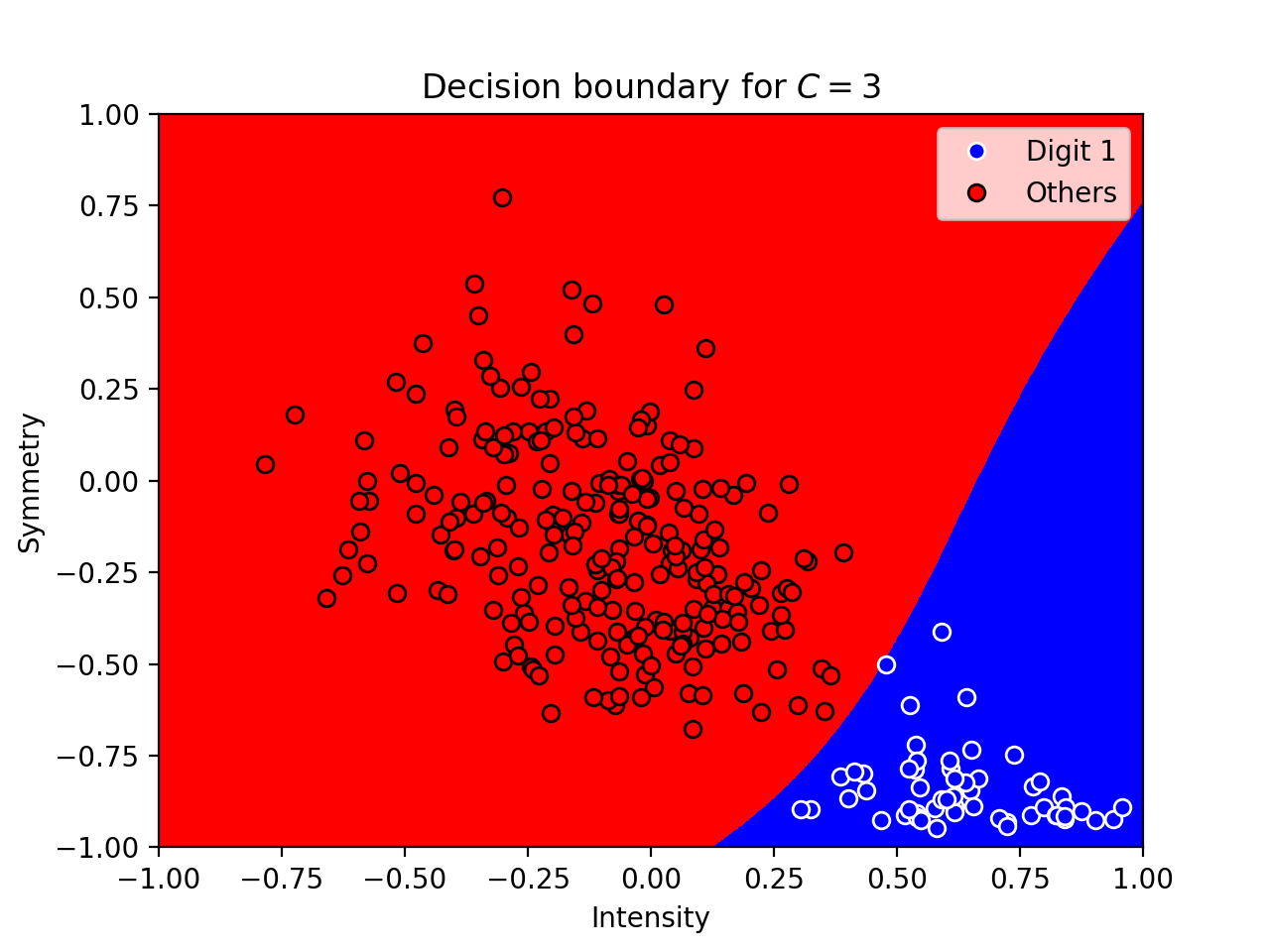
1. The plot of the decision boundary: -1 on the left and +1 on the right
2. Since , after transformation, we have , then
3. Since , then in X-space, .
4. SVM with digits data
5. I choose C = 1 as a small C:

I choose C = 1000 as a large C:

1. As C gets larger, the complexity of the decision boundary becomes larger. For example, in the second graph where C is larger, the top-right region is occupied by “blue” region.
2. Grid of values for C and cross validation errors:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

When C = 3, is the smallest. And here, .



1. Compare Methods: Linear, k-NN, RBF-network, Neural Network, SVM

The final test errors are:

for Linear model with 8th order polynomial transform, ;

for k-NN rule, ;

for RBF-network, ;

for Neural network with early stopping, ;

for SVM with 8th order polynomial kernel, .

We can see that SVM with 8th order polynomial kernel gives us the smallest test error.

It also has a fast running time. Thus, SVM with 8th order polynomial kernel should be the best choice.

For Neural network and RBF-network, the running time is fast and both have small

test errors. Therefore, these two can be the second choice.

However, k-NN is the slowest among these five algorithms and requires the largest

memory space. Thus, k-NN is not a good choice.

The regularized linear model with 8th order polynomial transform is not a good choice

as well. Not only it has the largest test error, but it also has a higher order polynomial transform which results in an increase in complexity.