CSCI 4100 Assignment 12

Boliang Yang 661541863

1. Neural Networks and Backpropagation

(a) For identity:

The forward propagation is $x_1 = [1{,}1]$, $s_1 = [0.75{,}0.75]$,

$$x_2 = [1. \quad, 0.635149, 0.635149], h(x) = 0.567574$$

The backpropagation is $\delta_2 = -0.216213$, $\delta_1 = [-0.032247, -0.032247]$,

$$\frac{de}{dW_2} = [-0.216213, -0.137327, -0.137327],$$

$$\frac{de}{dW_1} = \begin{bmatrix} -0.032247 & -0.032247 \\ -0.032247 & -0.032247 \\ -0.032247 & -0.032247 \end{bmatrix}$$

For sigmoid transformation:

The forward propagation is $x_1 = [1{,}1]$, $s_1 = [0.75{,}0.75]$,

$$x_2 = [1. \quad, 0.635149, 0.635149], s_2 = 0.567574, h(x) = 0.513576$$

The backpropagation is $\delta_2 = -0.179062$, $\delta_1 = [-0.026707, -0.026707]$,

$$\frac{de}{dW_2} = [-0.179062, -0.113731, -0.113731],$$

$$\frac{de}{dW_1} = \begin{bmatrix} -0.026707 & -0.026707 \\ -0.026707 & -0.026707 \\ -0.026707 & -0.026707 \end{bmatrix}$$

(b) For identity:

$$\frac{de}{dW_2} = [-0.216213, -0.137327, -0.137327],$$

$$\frac{de}{dW_1} = \begin{bmatrix} -0.032247 & -0.032247 \\ -0.032247 & -0.032247 \\ -0.032247 & -0.032247 \end{bmatrix}$$

For sigmoid transformation:

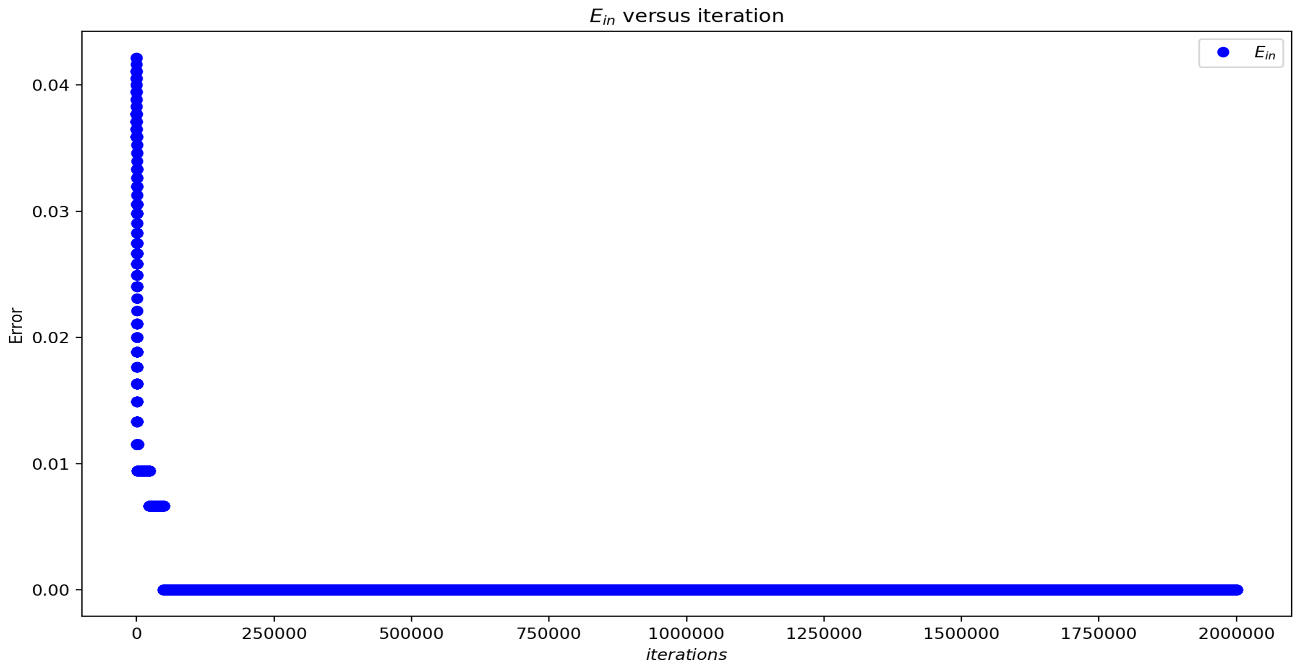
$$\frac{de}{dW_2} = [-0.179062, -0.113731, -0.113731],$$

$$\frac{de}{dW_1} = \begin{bmatrix} -0.026707 & -0.026707 \\ -0.026707 & -0.026707 \\ -0.026707 & -0.026707 \end{bmatrix}$$

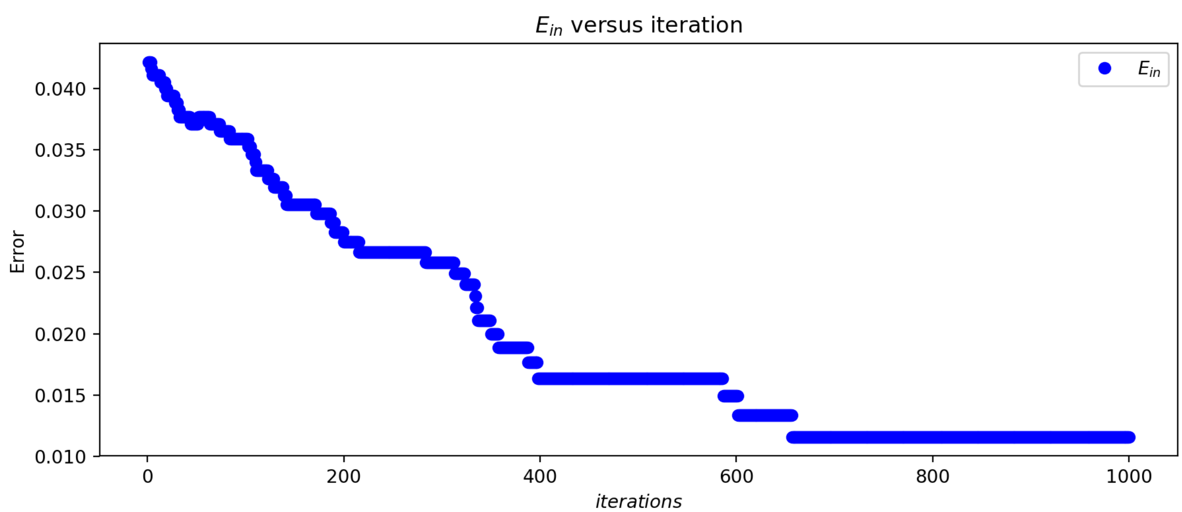
The results are the same as the previous results.

2. Neural Network for Digits

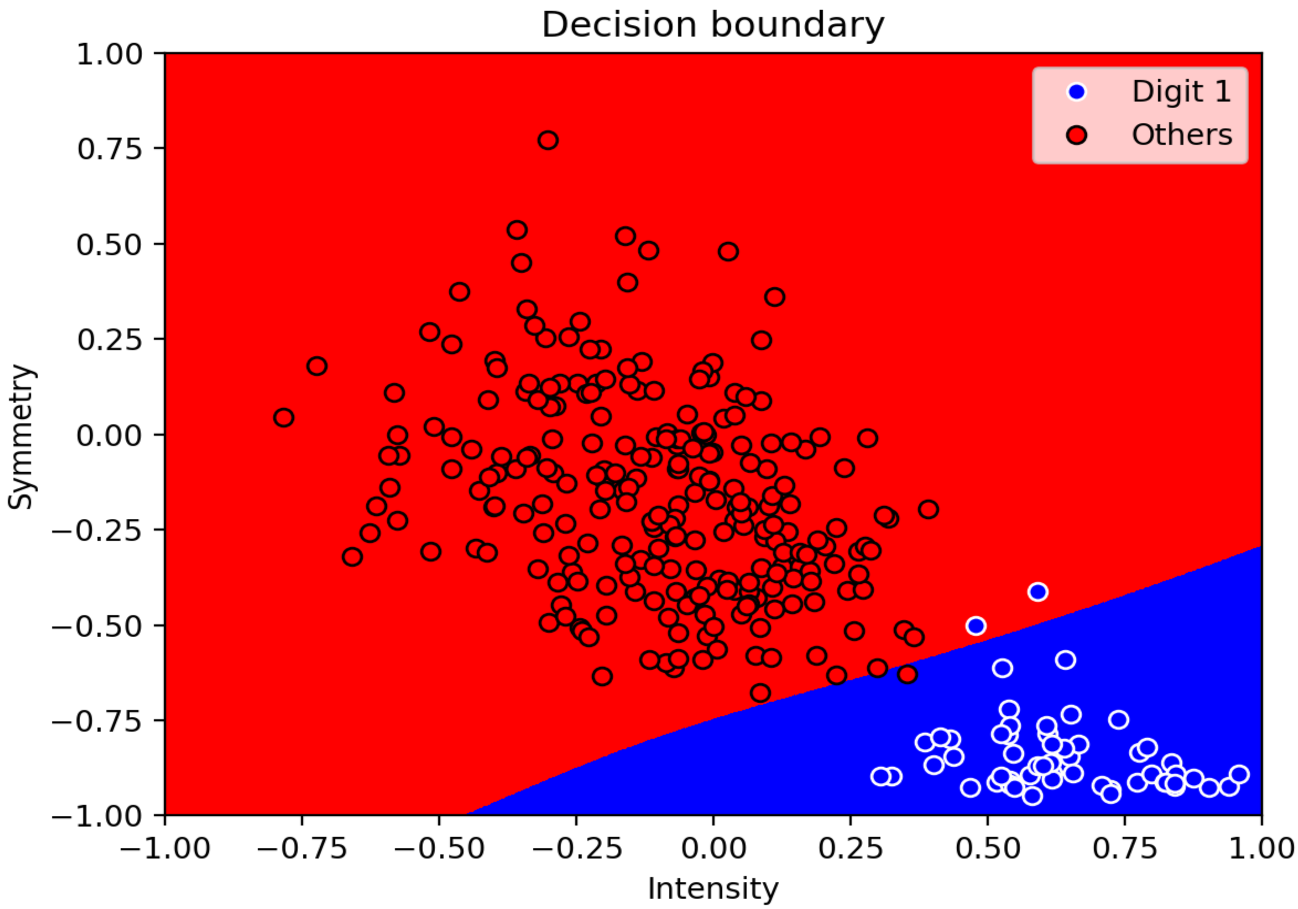
(a) The plot for 2×10^6 iterations:



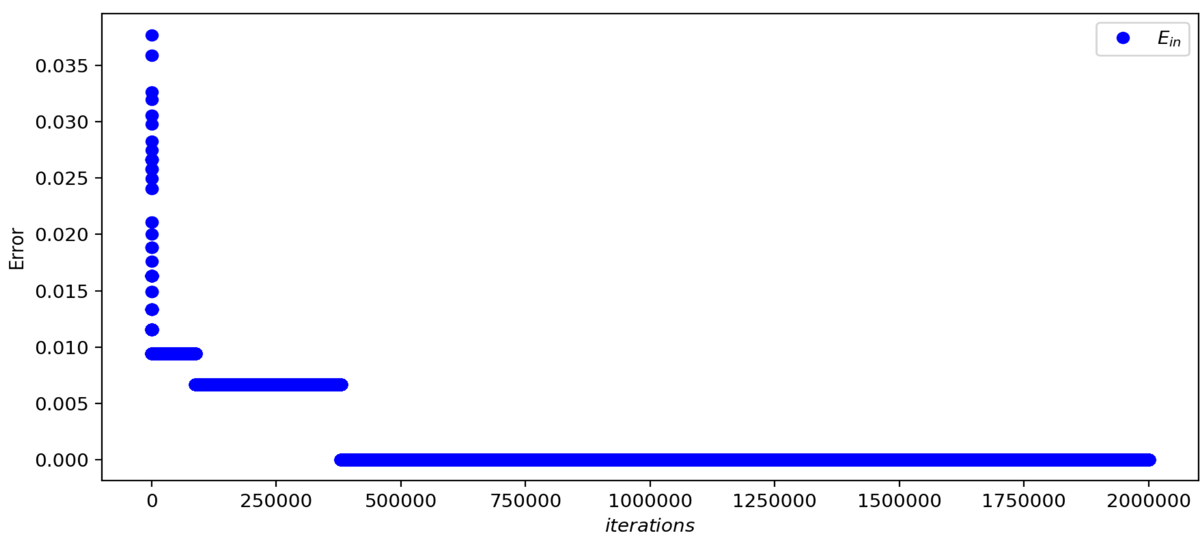
And here is plot for 1000 iterations: it converges at about 650 iterations.



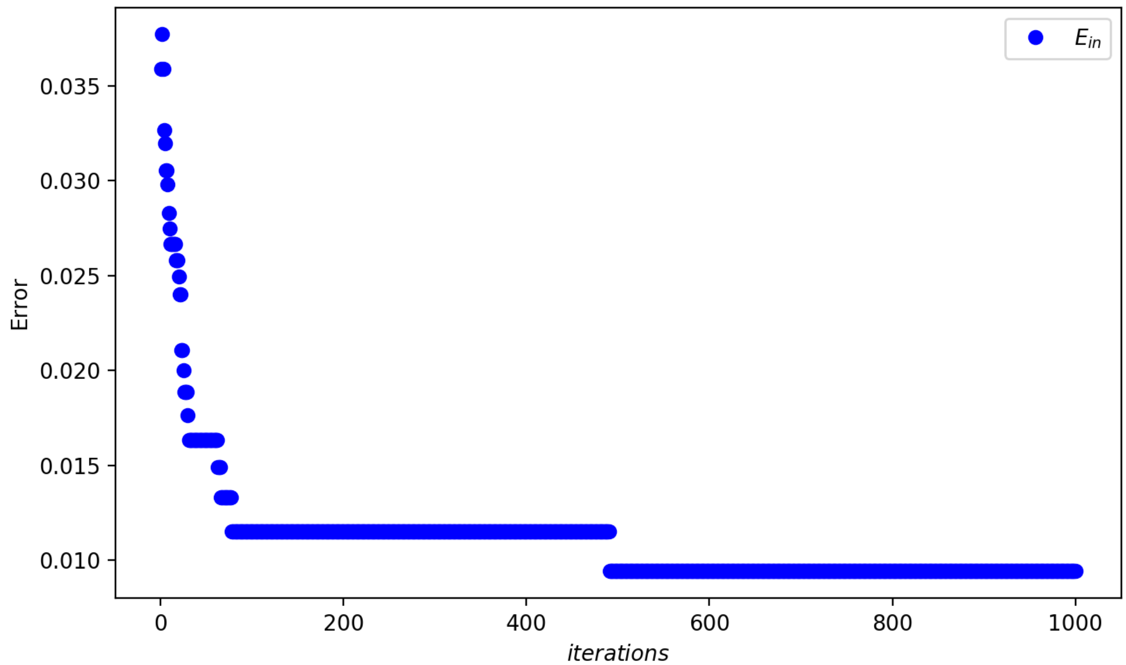
The plot for decision boundary for the resulting classifier:



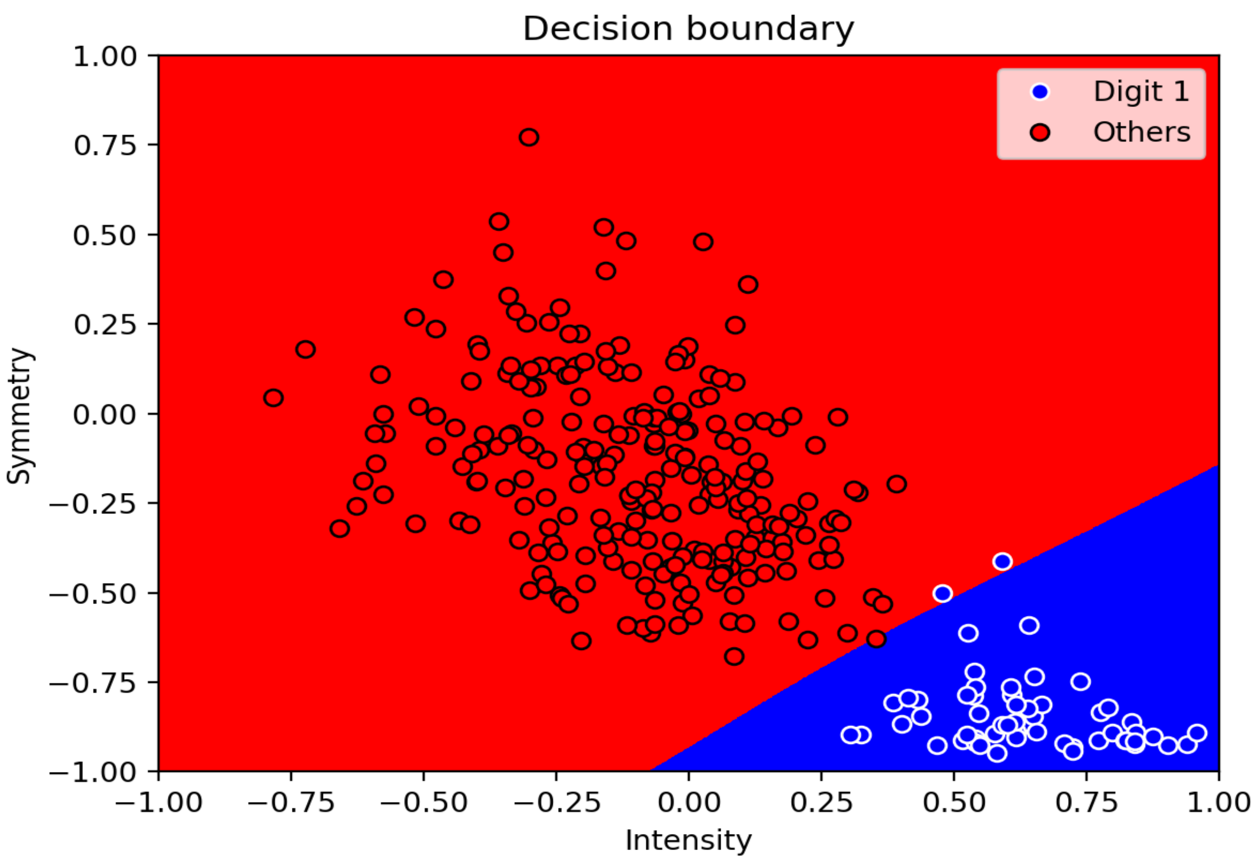
(b) With weight decay, here is the plot for 2×10^6 iterations:



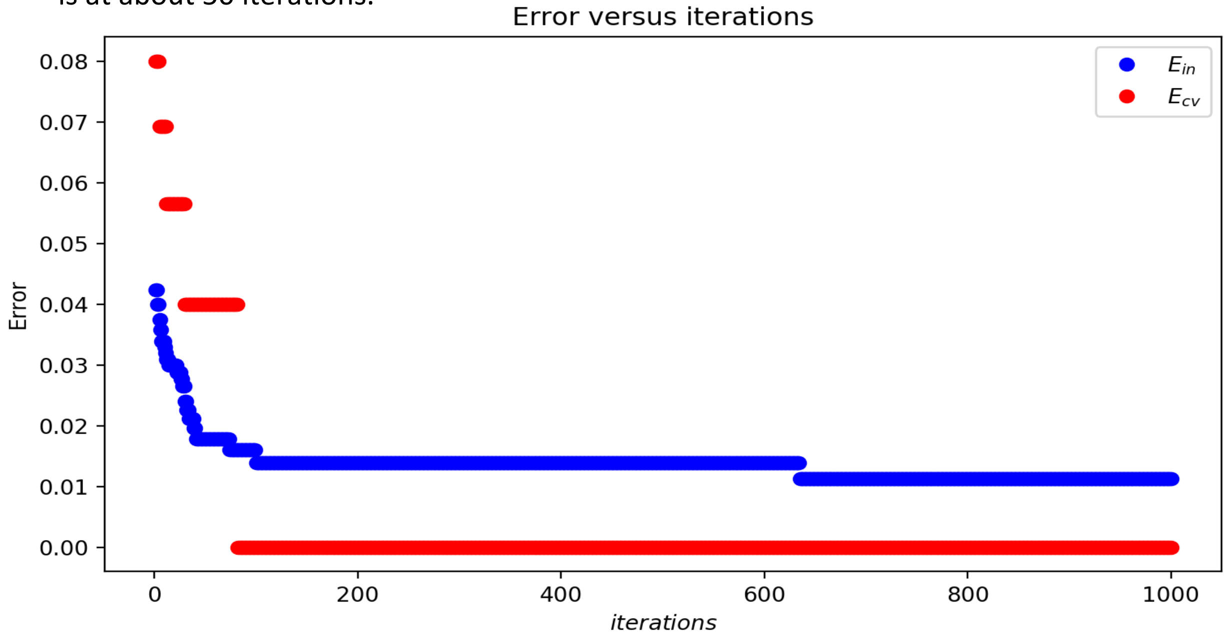
The plot for 1000 iterations: it converges at about 500 iterations.



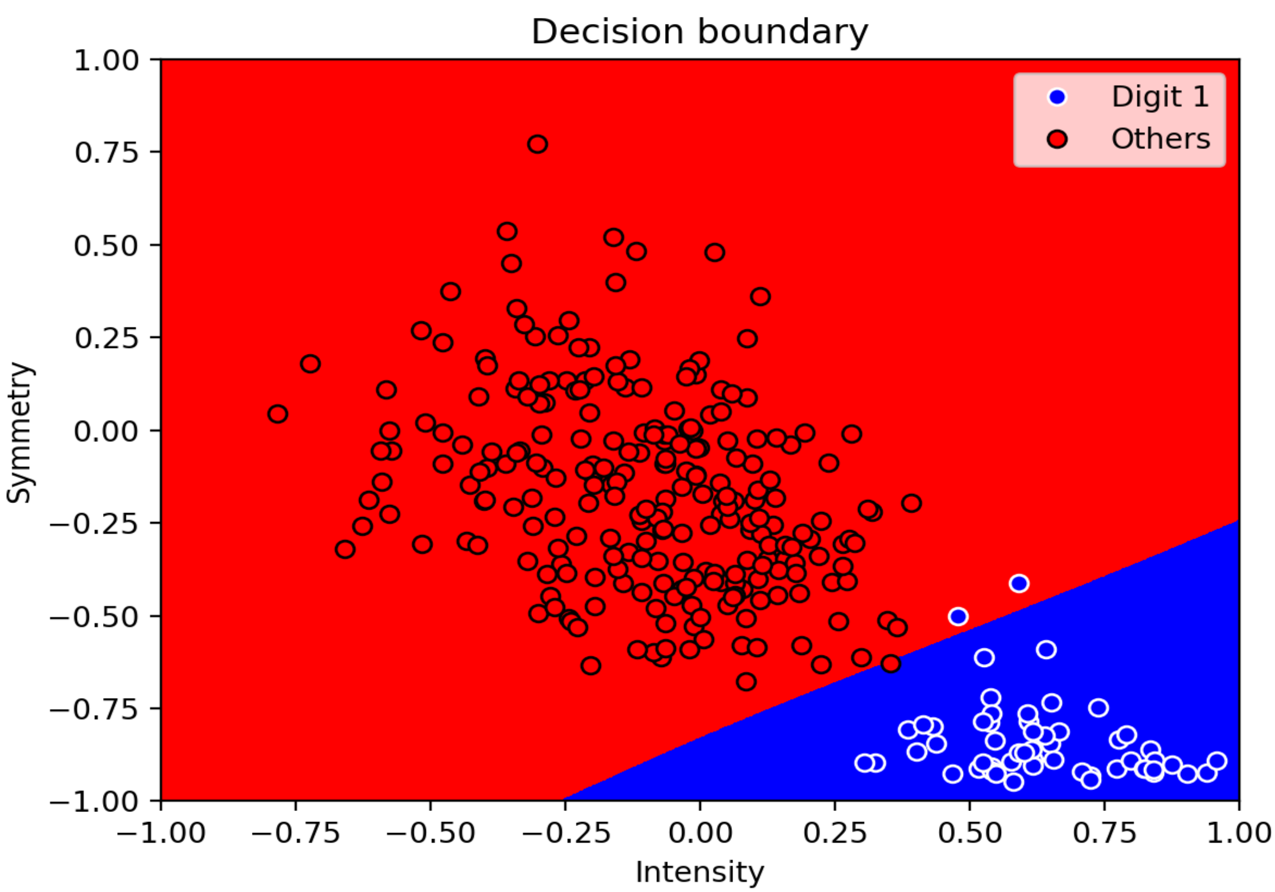
The plot for decision boundary for the resulting classifier:



(c) 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:



3. Support Vector Machines

(a) We solve optimization problem in order to get optimal separating hyperplane:

$$\min_{w,b} \frac{1}{2} w^T w \ \text{ subject to: } \ y_n(w^T x_n + b) \geq 1 \ (n = 1, \dots, N)$$

Since $x_1 = (1,0), y_1 = +1, x_2 = (-1,0), y_2 = -1$, then the problem becomes:

$\min \frac{1}{2}(w_1^2 + w_2^2)$ subject to: $1(w_1 + b) \geq 1$ and $-1(-w_1 + b) \geq 1$

We get $w_1 = 1, w_2 = 0, b = 0$. Thus, the optimal hyperplane is $g(x) = sign\left(\begin{bmatrix}1\\0\end{bmatrix}^T x\right) = sign(x_1)$. Since the equation for plane is $x_1 = 0$, and the line segment equation is $x_2 = 0$, then the hyperplane is perpendicular to the line segment.

(b) i. For $x_1 = (1,0), y_1 = +1$, we have $z_1 = (1,0), y_1 = +1$;

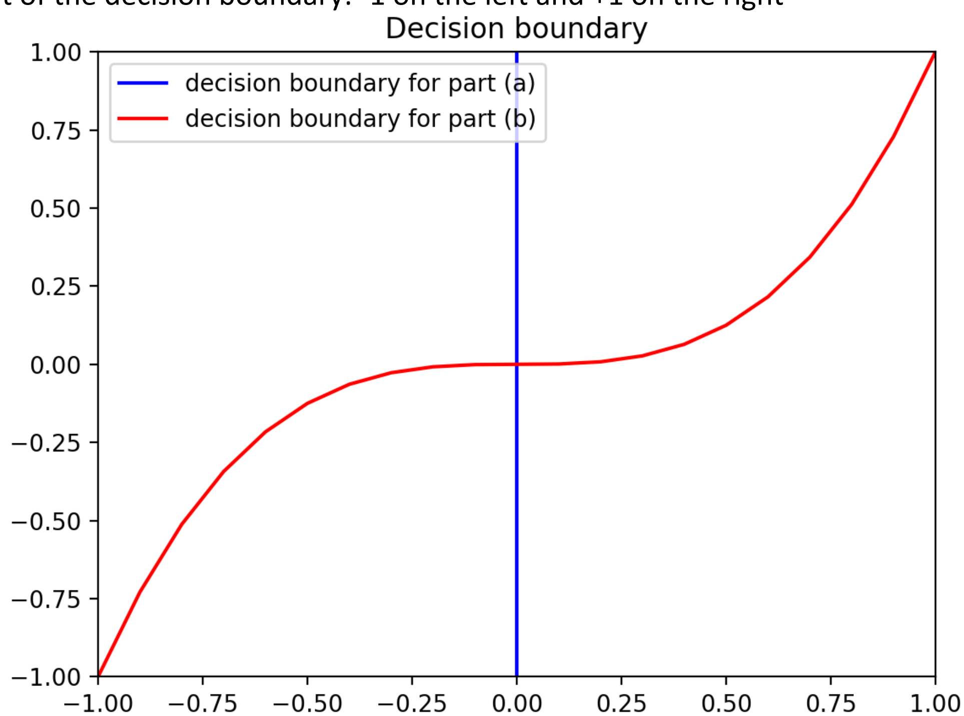
For $x_2 = (-1,0), y_2 = -1$, we have $z_2 = (-1,0), y_2 = -1$.

ii. We need to solve

$\min \frac{1}{2}(w_1^2 + w_2^2)$ subject to: $1(w_1 + b) \geq 1$ and $-1(-w_1 + b) \geq 1$

We get $w_1 = 1, w_2 = 0, b = 0$. Thus, the optimal hyperplane is $g(z) = sign\left(\begin{bmatrix}1\\0\end{bmatrix}^T z\right)$.

(c) The plot of the decision boundary: -1 on the left and +1 on the right

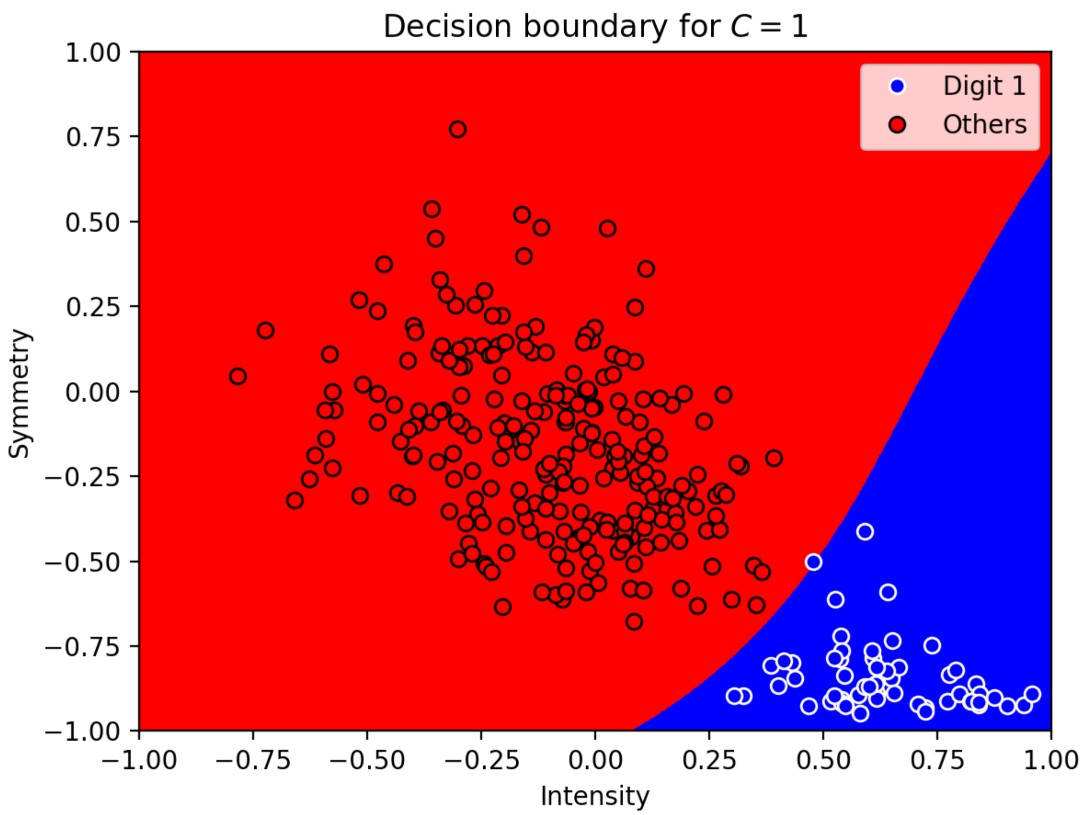


(d) Since $x = (x_1, x_2), y = (y_1, y_2)$, after transformation, we have $z_x = (x_1^3 - x_2, x_1x_2), z_y = (y_1^3 - y_2, y_1y_2)$, then $z_xz_y = (x_1^3 - x_2)(y_1^3 - y_2) + x_1x_2y_1y_2 = x_1^3y_1^3 - x_1^3y_2 - x_2y_1^3 + x_2y_2 + x_1x_2y_1y_2$

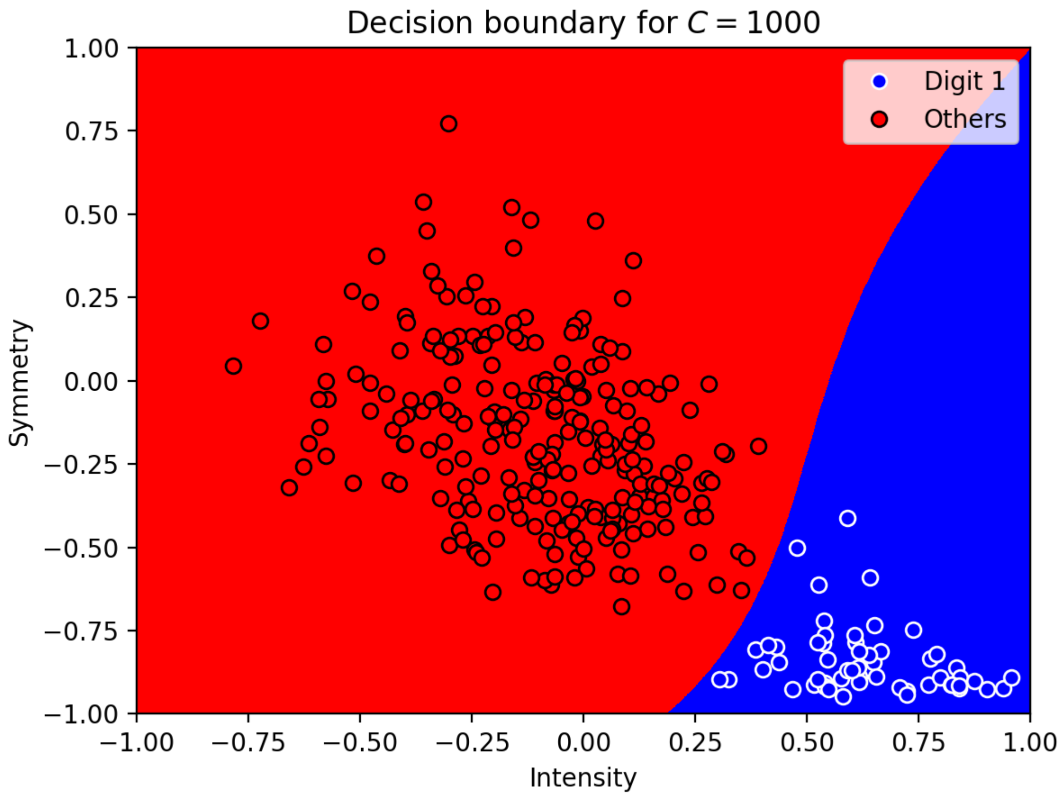
(e) Since $g(z) = sign\left(\begin{bmatrix}1\\0\end{bmatrix}^T z\right)$, then in X-space, $g(x) = sign(x_1^3 - x_2)$.

4. SVM with digits data

(a) I choose C = 1 as a small C:



I choose C = 1000 as a large C:

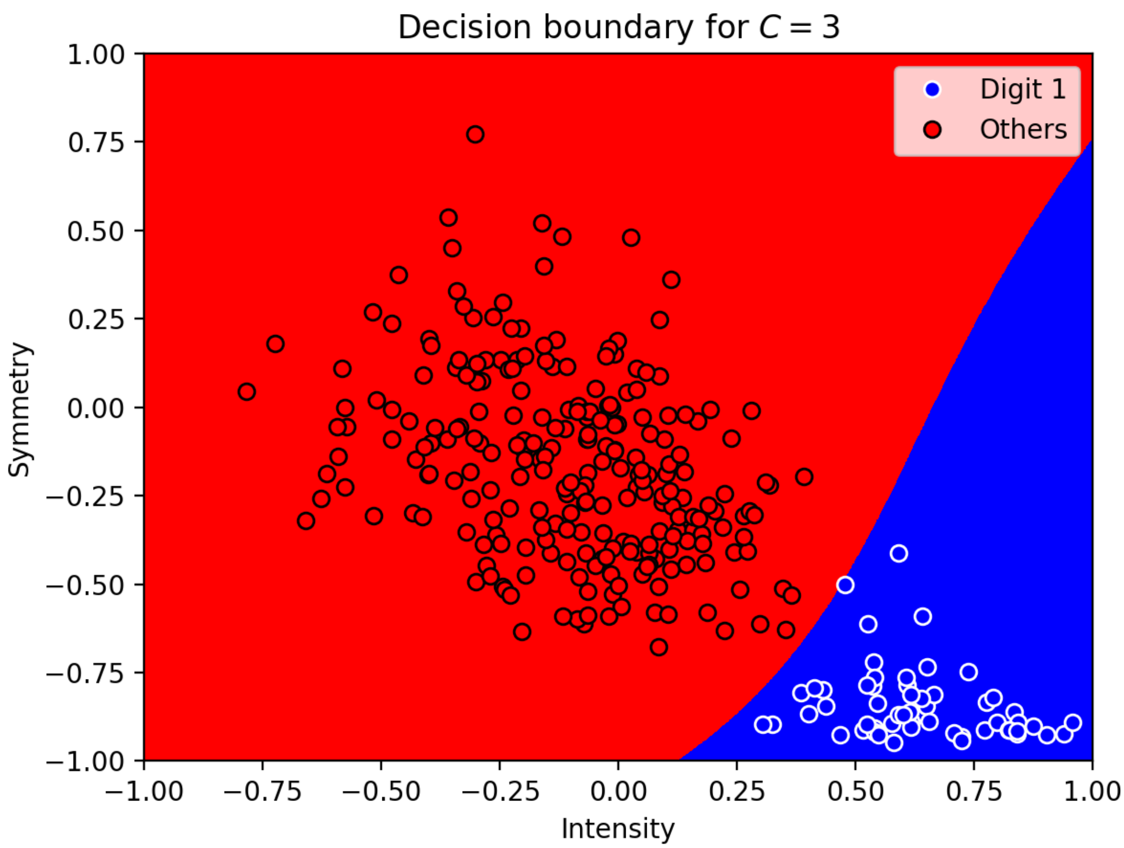


(b) As C gets larger, the complexity of the decision boundary becomes larger. For example, in the second graph where C is larger, the top-right region is occupied by “blue” region.

(c) Grid of values for C and cross validation errors:

C	1	2	3	4	5	6	7	8
E_{cv}	0.003344	0.003344	0.0	0.003344	0.003344	0.003344	0.003344	0.003344
C	9	10	15	20	25	30	40	50
E_{cv}	0.003344	0.003344	0.003344	0.003344	0.003344	0.003344	0.003344	0.003344

When C = 3, E_{cv} is the smallest. And here, $E_{test} = 0.009012$.



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

The final test errors are:

for Linear model with 8th order polynomial transform, $E_{test} = 0.057199$;

for k-NN rule, $E_{test} = 0.009441$;

for RBF-network, $E_{test} = 0.009726$;

for Neural network with early stopping, $E_{test} = 0.009853$;

for SVM with 8th order polynomial kernel, $E_{test} = 0.009012$.

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.