Homework 02

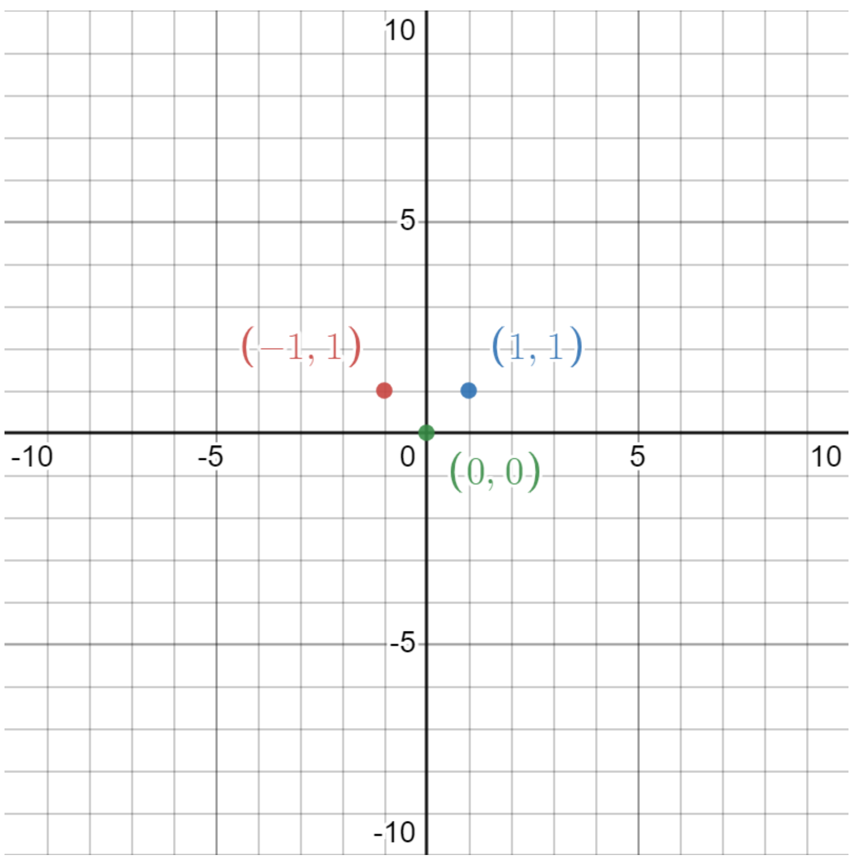
Group Members: Ira Deshmukh (3648692173) and Prem Tibadiya (6351018440)

Question 01:

1.1 No, the given data points cannot be linearly separated. As seen in the graph, the two points labelled -1 are on either side of the datapoint labelled as 1. To successfully classify them separately we would require a non-linear classifier.

1.2 $\phi(x_1) = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ $\phi(x_2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $\phi(x_3) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$\therefore$ New plot:



Yes, as seen from the graph above, the three datapoints can now be easily separated by a linear hyperplane.

1.3 Gram Matrix $= K(x_i, x_j)$

$$= \phi(x_i)^T \phi(x_j)^T$$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1.4 Primal Formulation:

$$\min_{w, b, \varepsilon_i} \; C \sum_i \varepsilon_i + \frac{1}{2} \|w\|_2^2$$

subject to: $y_i(w^T \phi(x_i) + b) \geq 1 - \varepsilon_i$

$$\varepsilon_i \geq 0$$

$$w^* = \sum \alpha_i^* y_i \phi(x_i)$$

$$= \alpha_1(-1)\begin{pmatrix} -1 \\ 1 \end{pmatrix} + \alpha_2(-1)\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \alpha_3(1)\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$= \alpha_1\begin{pmatrix} +1 \\ -1 \end{pmatrix} + \alpha_2\begin{pmatrix} -1 \\ -1 \end{pmatrix} + \alpha_3(0)$$

$$= \begin{pmatrix} \alpha_1 - \alpha_2 \\ -\alpha_1 - \alpha_2 \end{pmatrix}$$

$$b^* = y_i - w^{*T} \phi(x_1)$$

$$= (-1) - \begin{pmatrix} \alpha_1 - \alpha_2 \\ -\alpha_1 - \alpha_2 \end{pmatrix}^T \begin{pmatrix} -1 \\ 1 \end{pmatrix} = (-1) - (\alpha_1 - \alpha_2 \quad -\alpha_1 - \alpha_2)\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$= (-1) - (-\alpha_1 + \alpha_2) - (-\alpha_1 - \alpha_2)$$

$$= -1 + \alpha_1 - \alpha_2 + \alpha_1 + \alpha_2$$

$$= -1 + 2\alpha_1$$

Dual Formulation:

$$\max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2} \sum_{ij} y_i y_j \alpha_i \alpha_j \phi(x_i)^T \phi(x_j)$$

subject to:

$$\sum_i \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C$$

1.5 $$\max_{\alpha_i} \left[(\alpha_1 + \alpha_2 + \alpha_3) + (-1)\left(\frac{1}{2}\right)\left(2\alpha_1^2 + 2\alpha_2^2\right) \right]$$

$$= \max_{\alpha_i} \left(\alpha_1 + \alpha_2 + \alpha_3 - \alpha_1^2 - \alpha_2^2 \right)$$

subject to: $\sum_i \alpha_i y_i = 0$

$$-\alpha_1 - \alpha_2 + \alpha_3 = 0 \longrightarrow \alpha_3 = \alpha_1 + \alpha_2$$

$$\therefore \max_{\alpha_i} \left(2\alpha_1 + 2\alpha_2 - \alpha_1^2 - \alpha_2^2 \right) = L$$

$$\therefore \frac{\partial L}{\partial \alpha_1} = 2 - 2\alpha_1 = 0 \qquad \frac{\partial L}{\partial \alpha_2} = 2 - 2\alpha_2 = 0$$

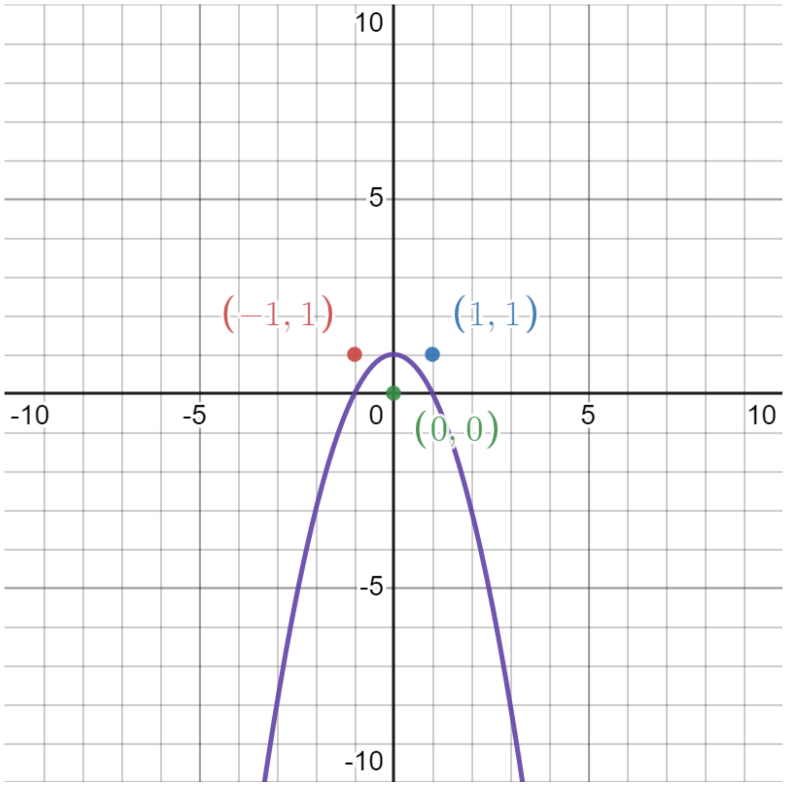
$$\therefore \alpha_1 = 1 \qquad \therefore \alpha_2 = 1$$

$$\therefore \alpha_3 = \alpha_1 + \alpha_2 = 2$$

$$\therefore w^* = \begin{pmatrix} \alpha_1 - \alpha_2 \\ -\alpha_1 - \alpha_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

$$b^* = -1 + 2\alpha_1 = -1 + 2(1) = 1$$

1.6. $\therefore$ The plot for decision boundary in two dimensional space:



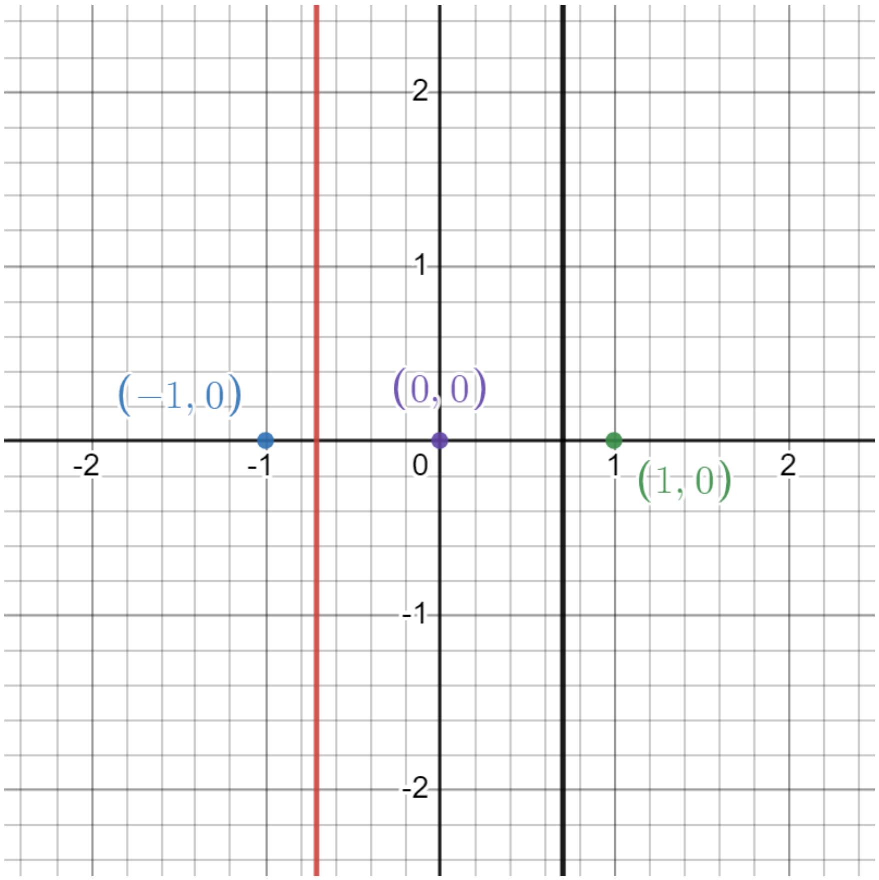
The plot for decision boundary in one dimensional space :

$$(0 \quad -2)\begin{pmatrix} x \\ x^2 \end{pmatrix} + b = 0$$

$$-2x^2 + 1 = 0$$

$$x^2 = 1/2$$

$$\therefore \quad x = 1/\pm\sqrt{2}$$



Question 02:

To prove : $K(x, x') = K_1(x, x') K_2(x, x')$

$$f' K f = \sum_{ij} f' K(x, x') f$$

$f', f \in R^n$.

$$= \sum f' K_1(x, x') K_2(x, x') f$$

$$= \sum f(K_1(x, x')) (f' K_2(x, x') f)$$

$$= \sum K_1(x, x') K_3(x, x')$$

We have assumed here that $K(x, x') = f(x) K(x, x') f(x')$.

This corresponds to the kernel function being positive and semi-definite. Thus then the product matrix of $K_1(x, x')$ & $K_3(x, x')$ is positive semi-definite.

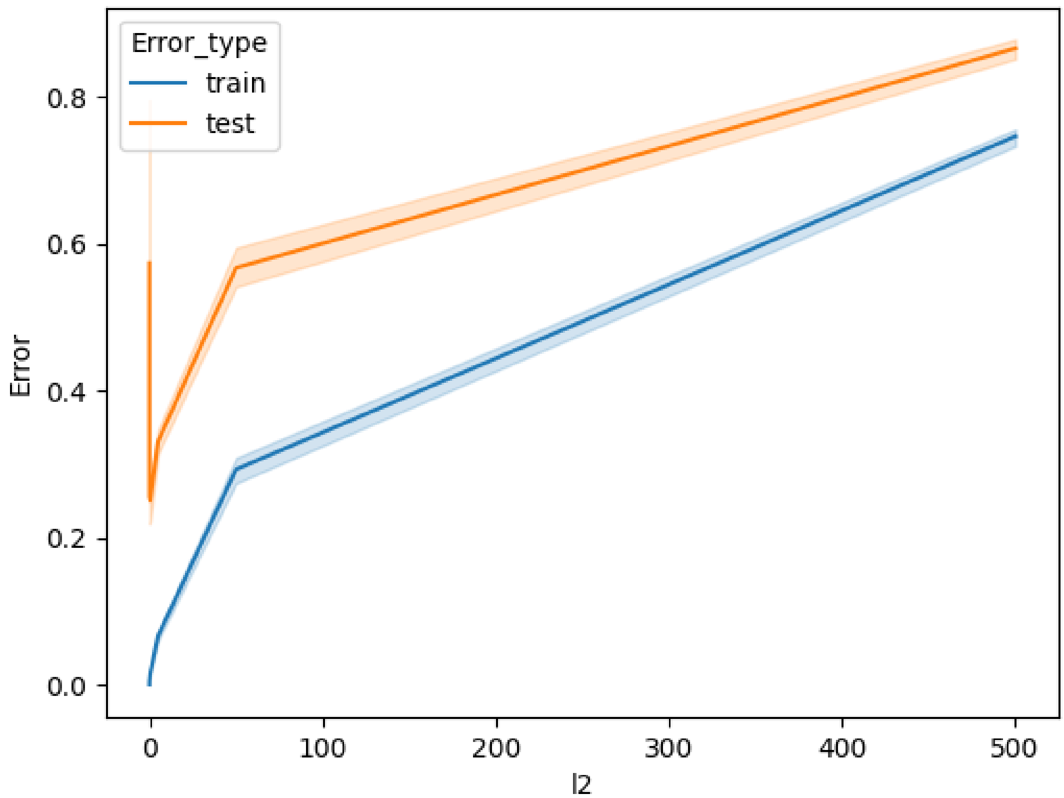
Question 03:

3.1

Average of train errors: 3.6969477125629436e-14

Average of test errors: 1.8773581801213912

3.2



The plot has a considerable error which tends to increase over iterations. When compared to the problem 3.1, it would be beneficial to avoid the issue of overfitting by increasing it's generalization.

3.3

Average of train error for alpha 5e-05 : 0.049700587986781404

Average of test error for alpha 5e-05 : 0.05134781929070856 4

Average of train error for alpha 0.0005 : 0.10075199097749454

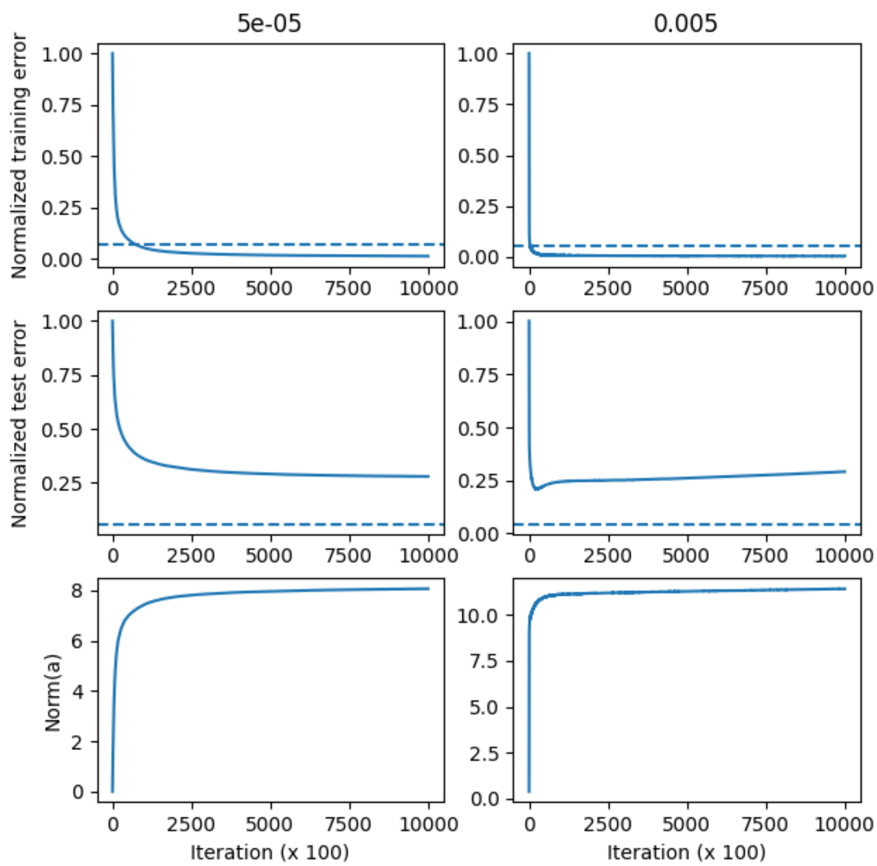
Average of test error for alpha 0.0005 : 0.10100192478806476

Average of train error for alpha 0.005 : 0.14978963427407158

Average of test error for alpha 0.005 : 0.15093456606806163

SGD pushes towards overfitting the model, which when compared to the L2 normalization does not. The training and testing error in SGD is quite less as compared to the problem in 3.1. This is dues to the properties of a SGD model.

3.4

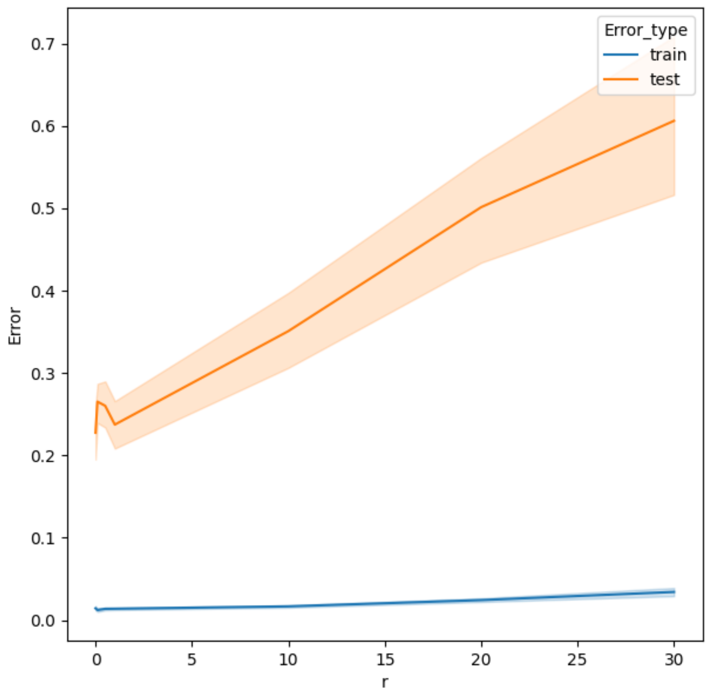


The plot does correspond to the intuition that a learning algorithm starts to overfit when the training errors becomes too small. This in turn reduces the model's ability to generalize. Thus, with every step size, the generalization is reducing.

L2 Regularization shrinks all the weights to small values, preventing the model from learning any complex concept with respect to any node/feature, thereby preventing overfitting. This is necessary because SGD starts overfitting to the model as the training error starts reducing.

3.5

In comparison to 3.2, it seems that the model is overfitting for the training data set.



Question 04:

4.1

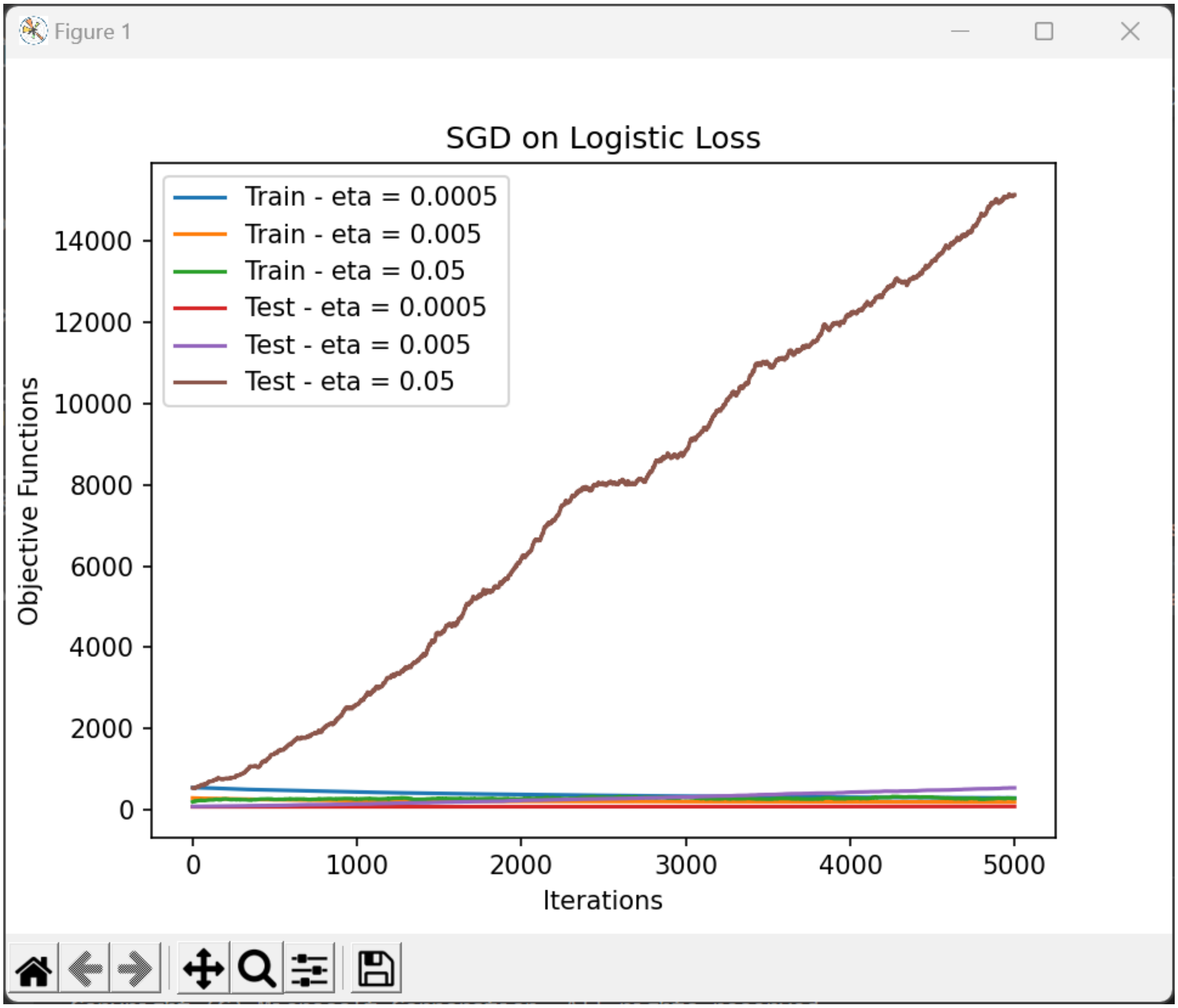
As you can see for eta=0.05 on test data the value of objective function keeps on increasing which means the step size is too large and SGD is not converging.

But if you plot individually for train and test data the value of objective function keeps on decreasing for eta=0.0005 & 0.005. The best result we get is for eta = 0.005 where the objective function value is least after 5000 iterations on both the train and test data.

Cost of Objective Function at 5000 iteration for eta = 0.0005 is 0.535

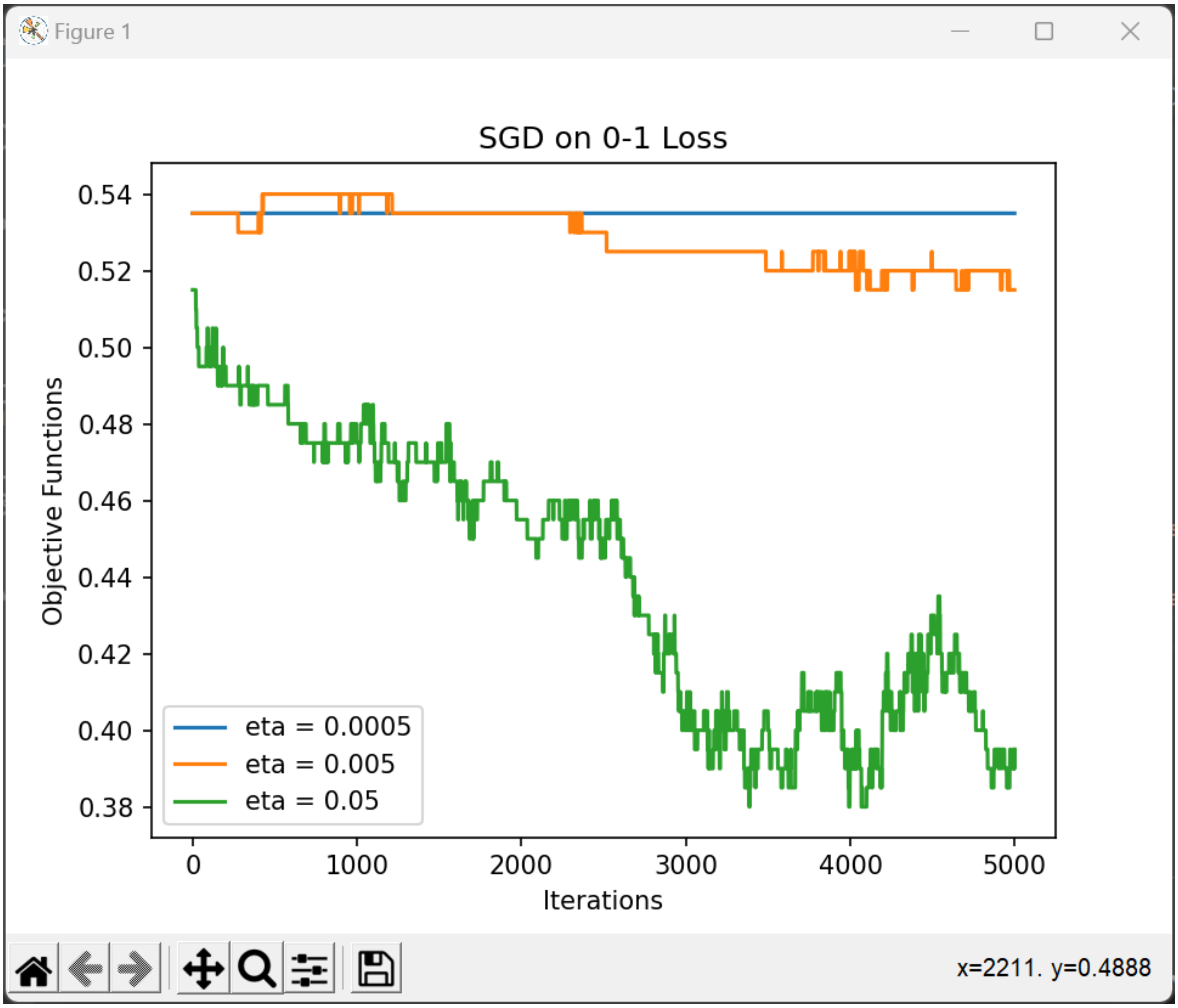
Cost of Objective Function at 5000 iteration for eta = 0.005 is 0.515

Cost of Objective Function at 5000 iteration for eta = 0.05 is 0.395



4.2

The lowest value of objective function 0.395 for eta = 0.05



4.3

As you can see the loss value for 0-1 loss is way less then logistic loss but the surrogate loss indicates the goodness of the classifier.

As minimizing the 0-1 loss is hard, we use the surrogate losses to calculate the loss as these losses can minimized (as you can see in the plot).

Homework-2

Problem 5:

5.1) No, linear classifiers does not do well on the dataset MOON and CIRCLES because they are not linearly separable. so, linear classifiers would be a bad fit. SVM with RBF kernel does pretty good on these datasets as you can the training and test accuracy is high then the rest classifiers.

5.2) As we use lower values of C like $C = 2.5 \times 10^{-3}, 2.5 \times 10^{-4}$ the training and test accuracy decreases to nearly 50% and as we increase the value of C to say 2.5, 25, 250, 2500, 25000 & 250000. The training accuracy remains constant. but the test accuracy increases a little. ~~For smaller~~ values of C As we keep on increasing the value of C the slope of classifier keeps on changing till 2.5 and then it remains constant but for higher values like 25000 it changes a little.

5.3) As we reduce the value of c to 1×10^{-3} the training and test accuracy both decreases for all the three dataset. But as we increases the value of c to {1, 10, 100} the training and test accuracy both increases, infact the training accuracy is 100% in some cases. But as we keep on increasing the value of c the training data is increasing but the test data is not. This is because it is overfitting. As we increase the value of c the decision boundary is getting more complex.

5.4) yes, regularization strength is ~~gett~~ helping us in our testing accuracy as we know the regularization is done for test data. As we ~~increase~~ decrease the value of c the training and test accuracy increases but after some value the test accuracy is not which means it is overfitting so now in this case regularization strength is helping to increase the accuracy ~~of~~ on test data for same value of c if we increase the reg_lamba value the test accuracy increases.