Homework 01

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

We would be requesting for one late day. It would be counted as Ira Deshmukh's late day.

HOMEWORK 01

Question 01 : Perceptron Algorithm Analysis

Given: $\|w_{OPT}\|^2 = 1$

$y_i (w_{OPT}^T x_i) \geq \gamma$ for some $\gamma > 0$

$\|x_i\| \leq R$

$y_i \in \{-1, 1\}$

1.1 $w_{k+1}^T w_{OPT} = (w_k + y_j x_j)^T w_{OPT}$ [j is the misclassified point]

$= w_k^T w_{OPT} + y_j (w_{OPT}^T x_j)$

$\geq w_k^T w_{OPT} + \gamma$

1.2 $\|w_{k+1}\|^2 = \|w_k + y_j x_j\|^2$

$= \|w_k\|^2 + \|y_j\|^2 \|x_j\|^2 + 2(y_j)(w_{ok}^T x_j)$

$\leq \|w_k\|^2 + \|x_j\|^2$ [$\|y_j\|^2 = 1$]

$\leq \|w_k\|^2 + R^2$

1.3 Assuming $w_0 = 0$

Then for any value of k from 1.1 and 1.2, we get:

$w_k^T w_{OPT} \geq w_{k-1}^T w_{OPT} + \gamma$

$\|w_k\|^2 \leq \|w_{k-1}\|^2 + R^2$

By using cauchy-schwartz inequality, we get:

$w_{k+1}^T w_{OPT} \leq \|w_{k+1}\| \|w_{OPT}\|$

$w_{k+1}^T w_{OPT} \leq \|w_{k+1}\|$

$\therefore \|w_k\| \geq w_{k-1}^T w_{OPT} + \gamma$

Then by the method of induction with M being total mistakes:

$\|w_{k+1}\| \geq M\gamma$

$\|w_{k+1}\|^2 \leq MR^2$

Therefore, by combining the above equations, we get:

$$\gamma M \leq \|W_{k+1}\| \leq R\sqrt{m}$$

1.4 From 1.3, we can see that

$$\gamma m \leq R\sqrt{m}$$

$$\therefore \gamma^2 m^2 \leq R^2 m.$$

$$\therefore m \leq \frac{R^2}{\gamma^2}$$

Question 02: Logistic Regression Analysis

Given: $F(x, A, K, b) = \dfrac{A}{1 + e^{-K(x-b)}}$

2.1 A - represents the odds ratio; it changes the height of the curve in the graph.

K - represents the loss function; it changes the steepness of the curve in the graph.

b - represents the x-intercept; moves along the x-axis in the graph.

2.2 When the value of $A = 1$ and $K \geq 0$. ~~logisti~~ $F(x, A, K, b)$ behaves as a CDF.

2.3 At the decision boundary $+K(x-b) = 0$

$$\therefore F(x, A, K, b) = \frac{1}{1 + e^0}$$

$$= \frac{1}{1+1} = \frac{1}{2} = 0.5$$

Thus, setting $F(x, A, k, b) \geq 0.5$ minimizes the classification ~~error~~. error.

2.4 $$\nabla F = \frac{-w^T e^{-w^T x+b}}{(1+e^{-w^T x+b})^2}$$

Thus, the direction of gradient at any point would be.

(a) parallel to w

(e) orthogonal to w.

$$F(x, w, b) = \frac{1}{2} = \frac{1}{1+e^{-w^T x+b}}$$

$$1+e^{-w^T x+b} = 2$$

$$e^{-w^T x+b} = 1$$

$$-w^T x+b = 0$$

$$x = \frac{|b|}{\|w\|}$$

∴ Distance of $F(x) = \frac{1}{2}$ from origin is $\frac{|b|}{\|w\|}$

Level set of function is always constant. Therefore, the level sets of $F(x, w, b)$ is $|b|$ or $\frac{\|w\|}{4}$

3)

3.1) From the question, let's assume algorithm E returns the smallest rectangle (C_S) enclosing all positive examples in the training set. Now as we assume realizability and the returned rectangle by E is the smallest rectangle which bounds all positive examples, so that implies all the negative examples are outside the smallest rectangle. Therefore we conclude that E is empirical risk minimizer.

3.2) We have been given a bad training data set (S') with the risk of predictor $L_{(D,f)}(f_{S'}^{ERM}) \geq 0.5$

Now as we know

$$R(f_{S'}^{ERM}) = \sum \text{Prob}(S')\, l(f_S^{ERM}, y)) \geq 0.5$$

As you can see the risk is more that clearly means the $\sum \text{prob}(S')$ is non-zero.
Therefore, we conclude that there is a non-zero probability of seeing a bad training set.

3.3) Let's assume a fix distribution D over X and let B^* be the rectangle that gives the true labels which are enclosed by it and everything outside B^* be negative labels. Let f be the hypothesis which is related to B^* on some training data S. Let's denote B_S as the rectangle returned by the algorithm E from the previous question and f_S^{ERM} denotes the corresponding hypothesis.

- Now as E finds the smallest rectangle enclosing all the positive labels that implies $B_S \subseteq B^*$
- Therefore, $L_{(D,f)}(f_S^{ERM}) = D(B^* \setminus B_S)$

which is equal to $D(\bigcup_{i \in \{1,2,3,4\}} B_i)$

where B_i are the rectangles as given in hint with probability mass distribution of $\varepsilon/4$.

$\therefore D(B_i) \leq \varepsilon/4$ for each $i \in \{1,2,3,4\}$

$\therefore$ By applying union bound

$$L_{(D,f)}(f_S^{ERM}) \leq \sum_{i=1}^{4} D(B_i) \leq \varepsilon$$

$\therefore$ if $L_{(D,f)}(f_S^{ERM}) > \varepsilon \Rightarrow D(B_i) > \varepsilon/4$ — (1)

for some $i \in \{1,2,3,4\}$.

Let's denote $I(S)$ as the set of indices i in $\{1,2,3,4\}$ such that (1) is true.

Now if $i \in I(S)$

$$\Rightarrow D^n(i \in I(S)) \leq (1-\varepsilon/4)^n$$

$$\therefore D^n(L_{(D,f)}(f_S^{ERM}) > \varepsilon) \leq D^n(I(S) \neq \phi)$$

$\Rightarrow$ union bound on all the indices that belongs to $I(S)$

$$\Rightarrow D^n(I(S) \neq \phi)$$

$$= D^n\left(\bigcup_{i \in \{1,2,3,4\}} \{i \in I(S)\}\right)$$

$$\leq \sum_{i \in \{1,2,3,4\}} D^n(i \in I(S))$$

$$\leq \sum_{i \in \{1,2,3,4\}} \varepsilon \quad (1-\varepsilon/4)^n$$

$$= 4(1-\varepsilon/4)^n \Rightarrow \leq 4e^{-n\varepsilon/4}$$

3.4) Let's assume there is d dimension.

∴ The We define the classifier $f_{(a_1,b_1,a_2,b_2 \ldots a_d,b_d)}$

by:-

$$f_{(a_1,b_1,\ldots a_d,b_d)}(x_1,\ldots x_d) = \begin{cases} 1 & \text{if } \forall i \in [d],\ a_i \le x_i \le b_i \\ -1 & \text{otherwise} \end{cases}$$

The function class for all the axis aligned rectangle in $\mathbb{R}^d$ is:-

$$\mathcal{F}^d_{rec} = \{ f_{(a_1,b_1,\ldots a_d,b_d)} : \forall i \in [d],\ a_i \le b_i \}.$$

Now as we have d dimensions we require 2d rectangle strips with probability mass distribution of $\frac{\varepsilon}{2d}$

$$\Rightarrow D(B_i) \le \frac{\varepsilon}{2d} \quad \text{for each } i \in \{1, \ldots d\}$$

∴ By applying union bound

$$L_{(D,f)}(f_S^{ERM}) \le \sum_{i=1}^{2d} D(B_i) \le \varepsilon$$

If $L_{(D,f)}(S^{ERM}) > \varepsilon \Rightarrow D(B_i) \geq \frac{\varepsilon}{2d}$ — (1)

for some $i \in \{1, \ldots 2d\}$

Let's denote $I(S)$ as the set of indices i in $\{1, \ldots 2d\}$ such that (1) is true.

Now if $i \in I(S)$

$$\Rightarrow D^n(i \in I(S)) \leq \left(1 - \frac{\varepsilon}{2d}\right)^n$$

$$\therefore D^n(L_{(D,f)}(f_S^{ERM}) > \varepsilon) \leq D^n(I(S) \neq \emptyset)$$

$$= D^n\left(\bigcup_{i \in \{1 \ldots 2d\}} \{i \in I(S)\}\right)$$

$$\leq \sum_{i \in \{1, 2 \ldots 2d\}} D^n(i \in I(S))$$

$$\leq \sum_{i \in \{1, 2 \ldots 2d\}} \left(1 - \frac{\varepsilon}{2d}\right)^n$$

$$= 2d\left(1 - \frac{\varepsilon}{2d}\right)^n$$

$$\leq 2de^{-\frac{n\varepsilon}{2d}}$$

Question 04:

4.1 Error rate of the k nearest neighbor algorithm in the validation set when k = 4 using Euclidean distance.

The validation error rate is 0.3466666666666667 in Problem Set 4.1

4.2 error rate of your k nearest neighbor algorithm in the validation set for k = 4 using Euclidean distance when data is using (1) Normalized featured vector, and (2) Min-max scaling featured vector.

The validation error rate is 0.37333333333333335 in Problem Set 4.2 when using normalization

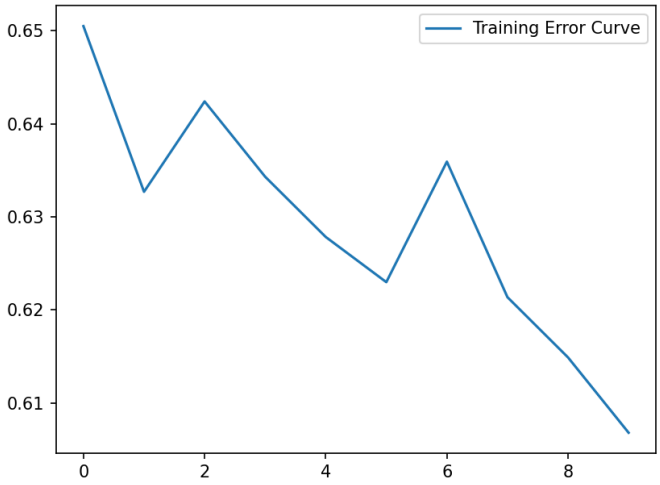
The validation error rate is 0.3466666666666667 in Problem Set 4.2 when using minmax_scaling.

4.3 error rate of your k nearest neighbor algorithm in the validation set for k = 4 using cosine distance for original data.

The validation error rate is 0.3333333333333333 in Problem Set 1.3, which use cosine distance.

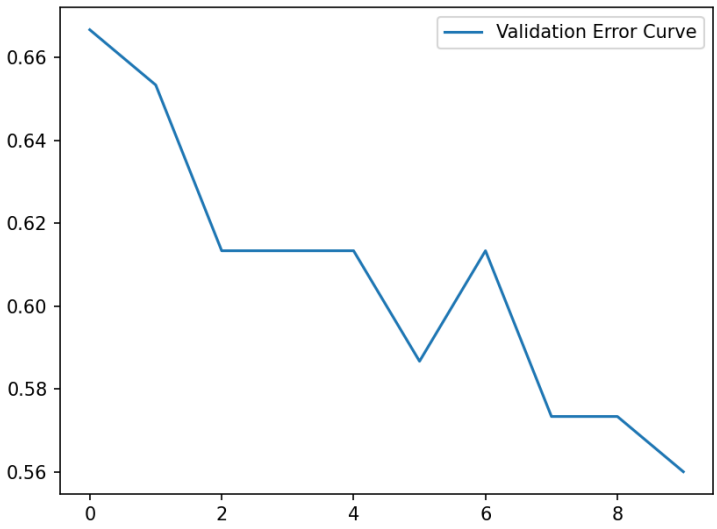
4.4

(1) Report and draw a curve based on the error rate of your model on the training set for each k. What do you observe?



As you can see, the more neighbors you use, the more accurate the segmentation. However, as we increase the K value until reaching N (the total number of data points), we seriously risk overfitting our model, leaving it unable to generalize well on unseen observations.

(2) Report and draw a curve based on the error rate of your model on the validation set for each k. What is your best k?



The best k is 12

(3) What do you observe by comparing the difference between the two curves?

Using the validation set for cross-validation we use hyper-parameters to improve the accuracy of our KNN algorithm and to find the best fit of k for our test set. From the training set we get the value of k as 10 but after further using hyper parameters we find the optimal value of k as 12.

(4) What's the final test set error rate you get using your best-k?

Using the best k, the final test error rate is 0.21333333333333335

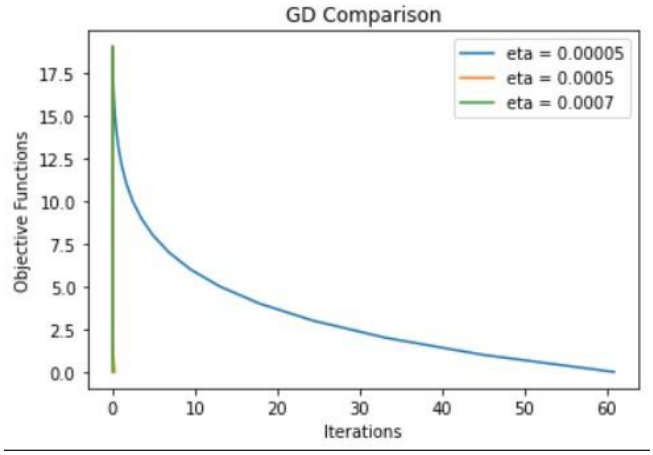
(5) Comment on these results.

Using cross validation gives the model an opportunity to test on multiple splits so as to get a optimal solution to perform well on unseen data. This was observed in the above algorithm as well when finding the best value for k to test the data on.

Question 05:

5.1 The gap in the training and test objective function values is known as the generalization gap. This gap tells us how good a algorithm is at generalizing. If the gap is small that means it is better at generalizing which ultimately means that the algo will perform better on unknown data. In our solution the gap is very less that means it is good in generalizing.

5.2 The step size is a hyperparameter which plays crucial part in the convergence of GD. If the step size value is very small, then it will take more iterations to reach convergence and if it is too big then it might oscillate and finally converge or in worst case scenario it may diverge. eta = 0.00005 is the best step size because as you see in the graph the cost is decreasing with a greater number of iterations that means it is converging. For the other values the cost is constant which means the gradient is oscillating and not reaching convergence.




```
Eta = 0.00005 : [array([[60.85292417]]), array([[44.96513737]]), array([[33.14380214]]), array([[24.36857646]]), array([[17.86189683]]), array([[13.0441753]]), array([[9.48341908]]),
array([[6.85759207]]), array([[4.92661064]]), array([[3.51149649]]), array([[2.47884852]]), array([[1.72927008]]), array([[1.18873969]]), array([[0.80217355]]), array([[0.52862094]]),
array([[0.33767718]]), array([[0.20680462]]), array([[0.11933153]]), array([[0.06295702]]), array([[0.02863405]])]
Eta = 0.0005 : [array([[0.2875961]]), array([[0.12432707]]), array([[0.07793518]]), array([[0.0692548]]), array([[0.0660106]]), array([[0.06457423]]), array([[0.06279057]]),
array([[0.06305358]]), array([[0.06229013]]), array([[0.06260506]]), array([[0.06224824]]), array([[0.06244357]]), array([[0.0622768]]), array([[0.06237903]]), array([[0.06230152]]),
array([[0.06235146]]), array([[0.06231571]]), array([[0.06233928]]), array([[0.06232289]]), array([[0.06233382]])]
Eta = 0.0007 : [array([[0.06232334]]), array([[0.06233802]]), array([[0.06231724]]), array([[0.06234576]]), array([[0.06230697]]), array([[0.06235957]]), array([[0.06228837]]),
array([[0.06238464]]), array([[0.06225462]]), array([[0.06243015]]), array([[0.06219342]]), array([[0.06251274]]), array([[0.06208251]]), array([[0.06266261]]), array([[0.0618817]]),
array([[0.06293473]]), array([[0.06151857]]), array([[0.06342944]]), array([[0.06086315]]), array([[0.06433097]])]
```

5.3 The step size is a hyperparameter which plays crucial part in the convergence of SGD. In our case step size with large value is giving us the result. For SGD the time taken to converge is inversely propotional to step size. In SGD the convergence is noisy as you can see in the graph but the path of convergence does not matter only the final destination to minima matters.

Learning rate of 0.00005 gives the best result for GD.

Learning rate of 0.00005 gives the best result for SGD.

For this data distribution GD performs well as the amount of data is less.

The best final objective function values has very large difference because SGD performs better when the data is large and it would help to reduce the computational calculations.

For GD : Number of Iterations*Number of data points (for our case : 20*1000)

For SGD : Number of Iterations (i.e 1000)

For GD : step size = 0.00005 | objective function value = 0.02863405

For SGD : step size = 0.00005 | objective function value = 3.48929453

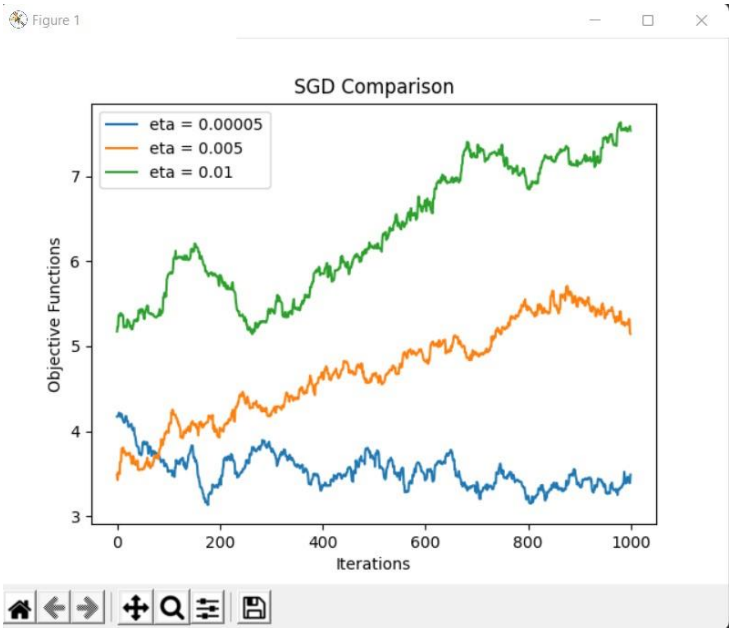


Figure 1
SGD Comparison
eta = 0.00005
eta = 0.005
eta = 0.01
Objective Functions
Iterations
3
4
5
6
7
0
200
400
600
800
1000