Assignment 2

Indian Institute of Information, Kanpur
Department of Computer Science
CS771 Introduction to Machine Learning
Instructor: Purushottam Kar

Submitted: November 1, 2022

Ambati Madhav
160098

Kudupudi Mohan Kumar
22105037

Abhinav Kuruma
22111401

Gnanendra Sri Phani Sai Channamsetty
22105032

Rahul Aggarwal
22111403

Preprint. Under review.

Contents

1 Answer 1 **3**

2 Answer 2 **3**

2.1 Case 1 4

2.2 Case 2 4

3 Answer 3 **4**

4 Answer 4 **5**

5 Answer 5 **5**

5.1 Hyperparameters used in Code(Q4) 5

5.2 Explanation for selection of Hyperparameters 5

5.3 Explanation for initialization of weights and bias term 6

6 Answer 6 **7**

6.1 Convergence Curve 7

7 References **7**

1 Answer 1

As we have to map binary digits 0,1 to signs -1,+1,
We should make a function say, $m : \{0,1\} \to \{-1,+1\}$. Such that

$$m(x) = \begin{cases} -1, & \text{if } x = 1 \\ +1, & \text{if } x = 0 \end{cases}$$

and another function named $f : \{-1,+1\} \to \{0,1\}$ and it should be such that

$$f(x) = \begin{cases} 0, & \text{if } x = +1 \\ 1, & \text{if } x = -1 \end{cases}$$

So, for m the fuction is $m(x) = 1 - 2x$ and for f the function is $f(x) = \frac{1-sign(x)}{2}$.

And, we also observe that f is the inverse of m. Now, let's take an example to see that

$$XOR(b_1, b_2, \ldots, b_n) = f\left(\prod_{i=1}^{n} m(b_i)\right) \tag{1}$$

Assume $b_1 = 0, b_2 = 1, b_3 = 1$

$$\begin{aligned} XOR(0,1,1) &= f(m(0) * m(1) * m(1)) \\ &= f(1 * -1 * -1) && \because m(x) = 1 - 2x \\ &= f(1) \\ &= 0 && \because f(x) = \frac{1 - sign(x)}{2} \end{aligned} \tag{2}$$

Hence, we can implement XOR in this manner.

2 Answer 2

(To prove) To exploit the above result, first give a mathematical proof that for any real numbers (that could be positive, negative, zero) r_1, r_2, . . . ,r_n for any n ∈ N , we always have

$$\prod_{i=1}^{n} sign(r_i) = sign \prod_{i=1}^{n} r_i$$

(Proof) To get the sign of a number we can use the following:

$$sign(r) = \frac{|r|}{r} \tag{3}$$

where,

$$sign(r) = \begin{cases} -1, & \text{if } r < 0 \\ 1, & \text{if } r > 0 \\ 0, & \text{if } r = 0 \end{cases}$$

Using above definition we can say that,

$$\frac{|r_1||r_2|\ldots|r_n|}{r_1 r_2 \ldots r_n} = \frac{|r_1 r_2 \ldots r_n|}{r_1 r_2 \ldots r_n}$$

proving this will prove the theorem.

2.1 Case 1

For, $r_i \neq 0 \ \forall_{i=0}^{n} r_i$

$$
\begin{aligned}
\text{L.H.S } &= \prod_{i=1}^{n} sign(r_1) \\
&= sign(r_1) * sign(r_2) * \cdots * sign(r_n) \\
&= \frac{|r_1|}{r_1} * \frac{|r_2|}{r_2} * \cdots * \frac{|r_n|}{r_n} \\
&= \frac{|r_1||r_2|...|r_n|}{r_1 r_2 ... r_n} \qquad (\because \text{ Product term amount to positive value})
\end{aligned}
\tag{4}
$$

$$sign \text{ of L.H.S depends on } (\frac{1}{r_1 r_2 ... r_n})$$

$$
\begin{aligned}
\text{R.H.S } &= sign \prod_{i=1}^{n} r_i \\
&= sign[(r_1) * (r_2) * \cdots * (r_n)] \\
&= \frac{|r_1 * r_2 * \cdots * r_n|}{r_1 * r_2 * \cdots * r_n} \\
&= \frac{|r_1 r_2 \ldots r_n|}{r_1 r_2 ... r_n} \qquad (\because \text{ The numerator itself is positive})
\end{aligned}
\tag{5}
$$

Let $r_1 * r_2 * \cdots * r_n$ = x,
Then, we get from equation 3 $\frac{|x|}{x}$, and

$$|x| = \begin{cases} -x, & \text{if } x < 0 \\ x, & \text{if } x > 0 \end{cases}$$

So, anyways numerator becomes positive. As it is a modular function. So,

$$sign \text{ of R.H.S depends on } (\frac{1}{r_1 r_2 \ldots r_n})$$

L.H.S=R.H.S

2.2 Case 2

For, $r_i = 0 \ \forall_{i=0}^{n} r_i$

As $sign(0) = 0$. So, we don't care about other values and simply the whole value becomes zero for both L.H.S and R.H.S.

L.H.S=R.H.S
Hence Proved.

3 Answer 3

We have to prove that, we can map 9-Dimensional vector to D-Dimensional vector:

$$R^9 \rightarrow R^D$$

such that, $\forall (\tilde{u}, \tilde{v}, \tilde{w}), \exists w \in R^D$
Mathematically,

$$(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x}) = w^T.\phi(\tilde{x}) \tag{6}$$

$$
\begin{aligned}
L.H.S &= (\tilde{u}^T\tilde{x}).(\tilde{v}^T\tilde{x}).(\tilde{w}^T\tilde{x}) \\
&= (\Sigma_{i=1}^{9}\tilde{u}_i\tilde{x}_i)(\Sigma_{j=1}^{9}\tilde{u}_j\tilde{x}_j)(\Sigma_{k=1}^{9}\tilde{u}_k\tilde{x}_k) \\
&= \Sigma_{i=1}^{9}\Sigma_{j=1}^{9}\Sigma_{k=1}^{9}\,\tilde{u}_i\tilde{u}_j\tilde{u}_k\tilde{x}_i\tilde{x}_j\tilde{x}_k
\end{aligned}
$$

We have to map from $\tilde{x} \to \phi(\tilde{x})$ such that $\tilde{x}$ is a 9-Dimensional and $\phi(\tilde{x})$ is 729 dimension.
So, we get

$\tilde{x} = (\tilde{x}_1\tilde{x}_2\ldots\tilde{x}_9) \to \phi(\tilde{x}) = (\tilde{x}_1\tilde{x}_1\tilde{x}_1, \tilde{x}_1\tilde{x}_1\tilde{x}_2, \ldots, \tilde{x}_1\tilde{x}_1\tilde{x}_9, \tilde{x}_1\tilde{x}_2\tilde{x}_1, \ldots, \tilde{x}_9\tilde{x}_9\tilde{x}_9)$

Now, we get

$$(\tilde{u}^T\tilde{x}).(\tilde{v}^T\tilde{y}).(\tilde{w}^T\tilde{z}) = w^T.\phi(\tilde{x})$$

where, w = $(\tilde{u}_1\tilde{v}_1\tilde{w}_1, \tilde{u}_1\tilde{v}_1\tilde{w}_2, \ldots, \tilde{u}_1\tilde{v}_1\tilde{w}_9, \tilde{u}_1\tilde{v}_2\tilde{w}_1, \ldots\ , \tilde{u}_9\tilde{v}_9\tilde{w}_9)$

and so w is of 729 dimension. So, we proved

$$(\tilde{u}^T\tilde{x}).(\tilde{v}^T\tilde{y}).(\tilde{w}^T\tilde{z}) = w^T.\phi(\tilde{x})$$

i.e we can map 9-Dimensional to 729 Dimensional vector.

where $\exists w \in R^D$ for any $(\tilde{u}, \tilde{v}, \tilde{w})$ such that $\forall \tilde{x} \in R^9$.

4 Answer 4

<Code Link> : https://github.com/phanisai97/mlass1/raw/main/submit.zip

5 Answer 5

5.1 Hyperparameters used in Code(Q4)

There are 3 Hyperparameters in the code submitted

1) Step Length or Learning Rate 'lr'
2) Correction Factor or lambda 'la'
3) Dynamic Epoch '_epoch'

5.2 Explanation for selection of Hyperparameters

1) To obtain optimised values of Step Length and Correction Factor or Lambda, **grid search** has been used.

- Step length 'lr' was iterated in three different ranges of $[0.20e^{-01}\ to\ 0.29e^{-01}], [0.20e^{-02}\ to\ 0.29e^{-02}], [0.20e^{-03}\ to\ 0.29e^{-03}]$
- Correction Factor 'la' was iterated in three different ranges of $[0.20e^{-01}\ to\ 0.29e^{-01}], [0.20e^{-02}\ to\ 0.29e^{-02}], [0.20e^{-03}\ to\ 0.29e^{-03}]$
- Step length value of $0.23e^{-02}$ and Correction Factor value of $0.20e^{-03}$ have been obtained as optimised values from this grid search operation

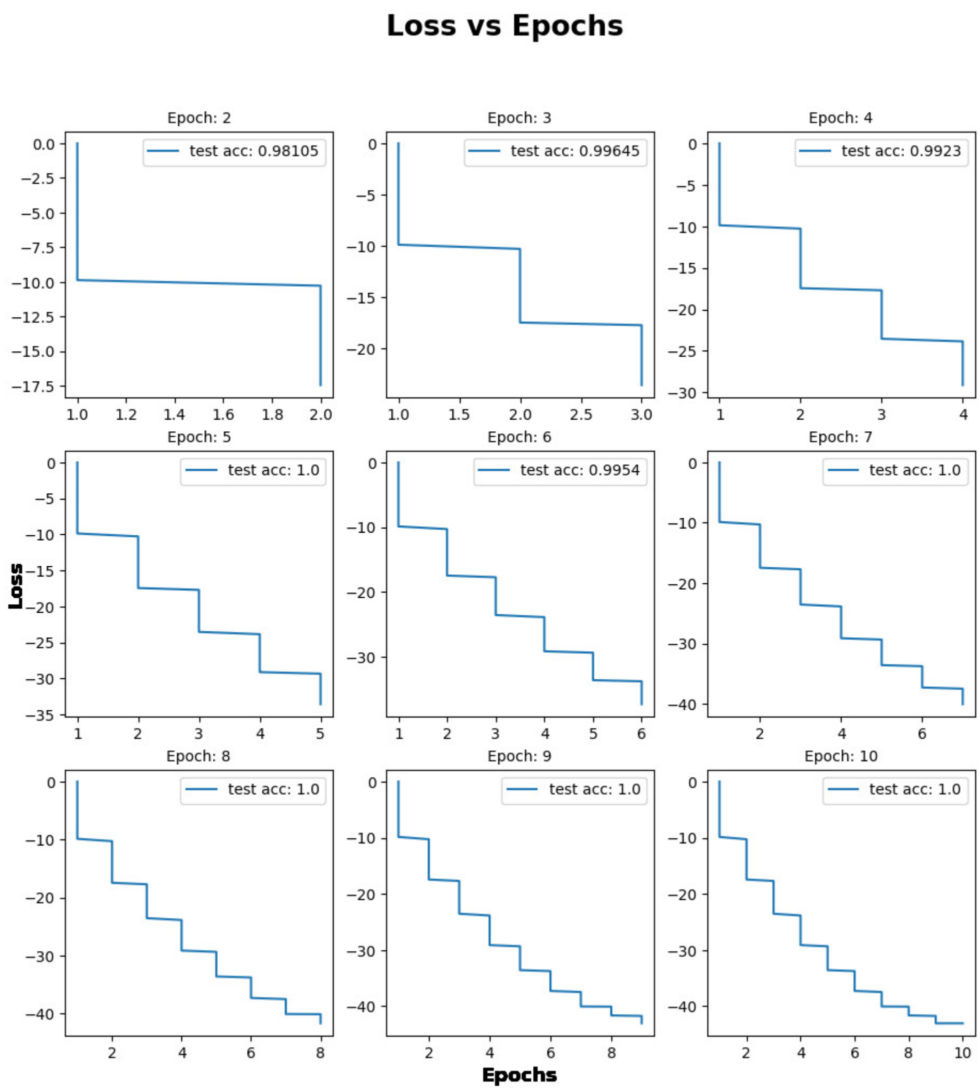
2) To decide the total number of epochs '_epoch', we chose a dynamic epoch which varies based on number of input data points using ceil() function as

$$\_epochs = \lceil(\frac{epochs * 10000}{y.size})\rceil$$

10000 is the size of the train data set with which optimum epochs has been obtained. The epochs gets scaled with the number of train data points , as per the formula.

- To arrive at optimum ‘epochs’ value of 5, a curve was plotted between hinge loss and number of epochs for 10000 train data points, number of epochs varying from 2 to 10. It was found that at epochs 5, model reached maximum of 100 percent accuracy , then decreased in accuracy followed by a subsequent increase in accuracy. It was speculated that model started over fitting beyond epoch 5. Hence, 5 has been selected as ‘epochs’ value.

- Plot between Hinge Loss and Number of Epochs has been included below



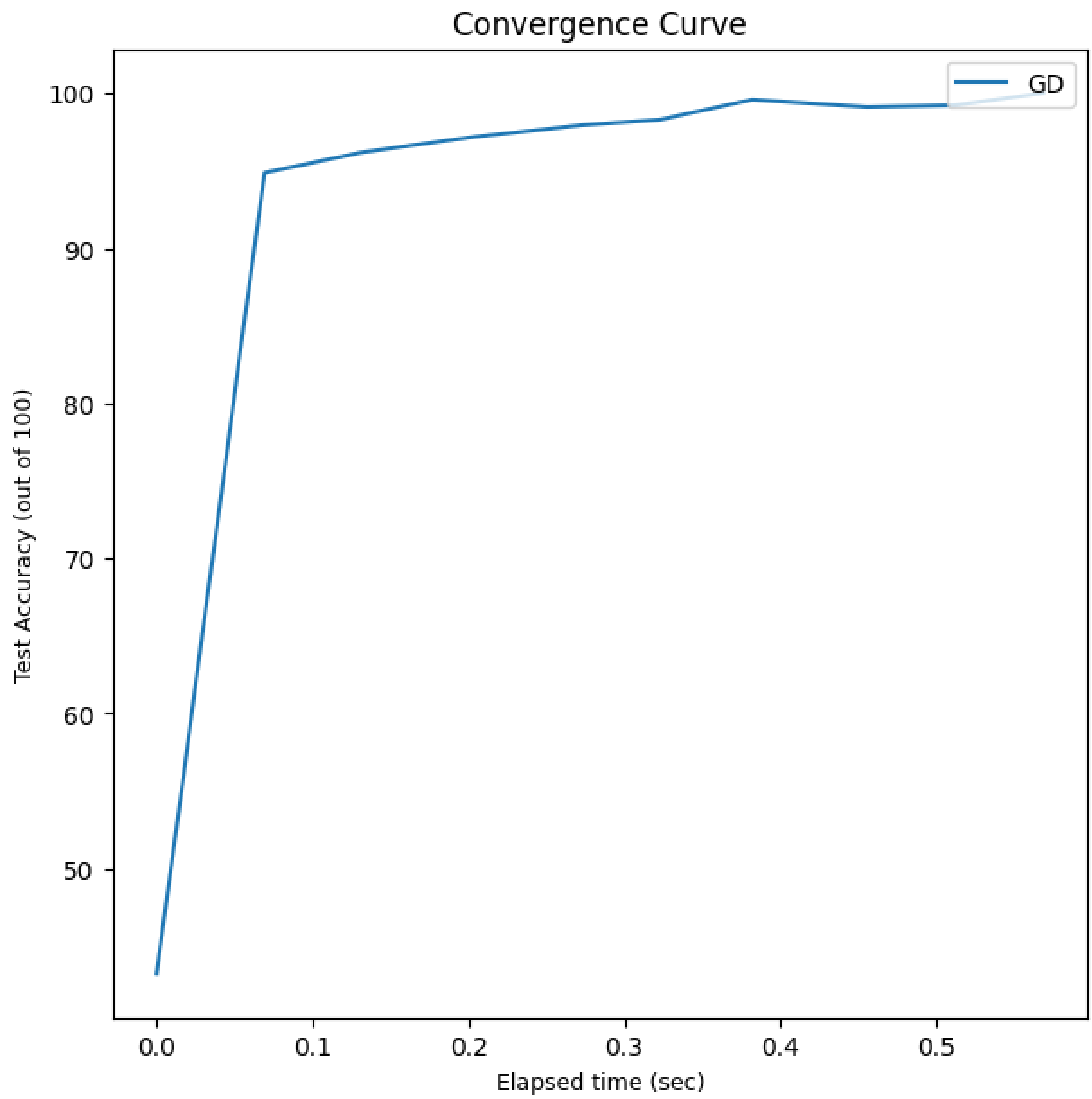
5.3 Explanation for initialization of weights and bias term

Both the Weights and bias term have been initialised with zero as it was found to be the fastest in converging to a optimised solution.

6 Answer 6

6.1 Convergence Curve

Convergence Curve between Time taken and test classificaton Accuracy has been included below.
GD stands for Gradient Descent solver method



7 References

[1] https://programmathically.com/understanding-hinge-loss-and-the-svm-cost-function/

[2] https://web.cse.iitk.ac.in/users/purushot/courses/ml/2022-23-a/discussion.html