HW 1 – Machine Learning Introduction

All code are present in the github repo https://github.com/somesh-bagadiya/CMPE257-Fall23-Somesh-Bagadiya/tree/4646d5f2607c3de5dc4a18cbf236f9d5a8366856/HW1

Task 1:

A modified version of exercise 1.3 from the textbook on the PLA weight update.
w(t + 1) = w(t) + y(t)x(t)
"The weight update rule in (1.3) in the textbook has the nice interpretation that it moves in the direction of classifying x(t) correctly."

LP 1:
[LP 1] Show that y(t) w^T(t) x(t) < O
Hint: x(t) is misclassified by w(t).
Ans.
The y(t) w^T(t) x(t) would be less than zero for a misclassified point because the sign of w^Tx would be the opposite of the target function, if the signs were the same the point wouldn't be a misclassified point.

For example,

x(t) is misclassified point by the vector w(t) then the sign would be -1
The target of the x(t) would be +1

Therefore, (+1) * (-1) < 0

LP 2:
[LP 2] Show that for W'final, the final hypothesis generated by the Perceptron Learning Algorithm,
assuming that the data is linearly separable, $Y_n W^T_{final} X_n > 0$ for all data points xn.
Ans.
Perceptron Algorithm will stop once it correctly classifies all the points in the data set, so if we have a final weight vector i.e. W_{final} This would correctly classify all the points in the data set. y(t) w^T(t) x(t) would be positive it the point is correctly classified.

For Example,

For some x the target is -1, and the sign(w^Tx) would also be -1 as it correctly classifies the point x. Thus, $Y_nW^T_{final}X_n > 0$.

HP:
[HP) Show that y(t)w^T(t + l)x(t) > y(t)w^T(t)x(t). As far as classifying x(t) is concerned, argue that the move from w(t) to w(t + 1) is a move "in the right direction".
Ans.
Show that y(t) w^T(t+1) x(t) > y(t) w^T(t) x(t) - - - - - - - - - - {1}
We know w(t+1) = w(t) + y(t) x(t).

$w^T(t+1) = w^T(t) + y(t)\, x^T(t)$

Put $w^T(t+1)$ in the equation 1

$y(t)\, [w^T(t) + y^T(t)\, x^T(t)]\, x(t) > y(t)\, w^T(t)\, x(t)$

$y(t)\, w^T(t)\, x(t) + [\, y(t)\, x(t)\,]\, [\, y(t)\, x^T(t)] > y(t)\, w^T(t)\, x(t)$

$y(t)\, w^T(t)\, x(t) + [y(t)]^2\, x^T(t)\, x(t) > y(t)\, w^T(t)\, x(t)$

Simplifying this we will get,

$[y(t)]^2\, x^T(t)\, x(t) > 0$

We know y(t) can be either -1 or 1, thus $[y(t)]^2 > 0$.

We also know $x^T(t)\, x(t)$ is a positive definitive matrix; it is greater than 0.

Hence Proved, $y(t)^2\, x^T(t)\, x(t) > 0$ or $y(t)\, w^T(t+1)\, x(t) > y(t)\, w^T(t)\, x(t)$

Theory:

x(t) is a misclassified point and from LP 1 we know that the $y(t)\, w^T(t)\, x(t) < 0$, We will update the weights to classify the point x(t) correctly. The updated weight would be w(t+1), and from LP2 we know that if a weight correctly classifies a point then $y(t)\, w^T(t)'\, x(t) > 0$. So, $y(t)\, w^T(t+1)\, x(t) > 0$. Therefore we can say that $y(t)\, w^T(t+1)\, x(t) > y(t)\, w^T(t)\, x(t)$.

TASK 2:

LP 1:

[LP I] The perceptron in two dimensions is defined as $h(x) = sign(w^Tx)$, where $w = [w0, w1, w2]^T$ and $x = [1, x1, x2]^T$. Compare the linear separator to the traditional equation of a line, i.e., X2 = mX1 + C.

Specifically, what are m and c regarding W0, W1, and W2?

Ans.

$X_2 = mX_1 + C$ - - - - - - - - - - - - - - - - - - - equation of line

$- mX_1 + X_2 - C = 0$

$mX_1 - X_2 + C = 0$

$C + mX_1 - X_2 = 0$

We can assume W_0, W_1, W_2 as the coefficients of the above-mentioned equations.

I.e

$W_0C + W_1X_1 + W_2X_2 = 0$ - - - - - - - - - - - - - - - - - Equation of line in term of W's

$W_2X_2 = - W_1X_1 - W_0C'$

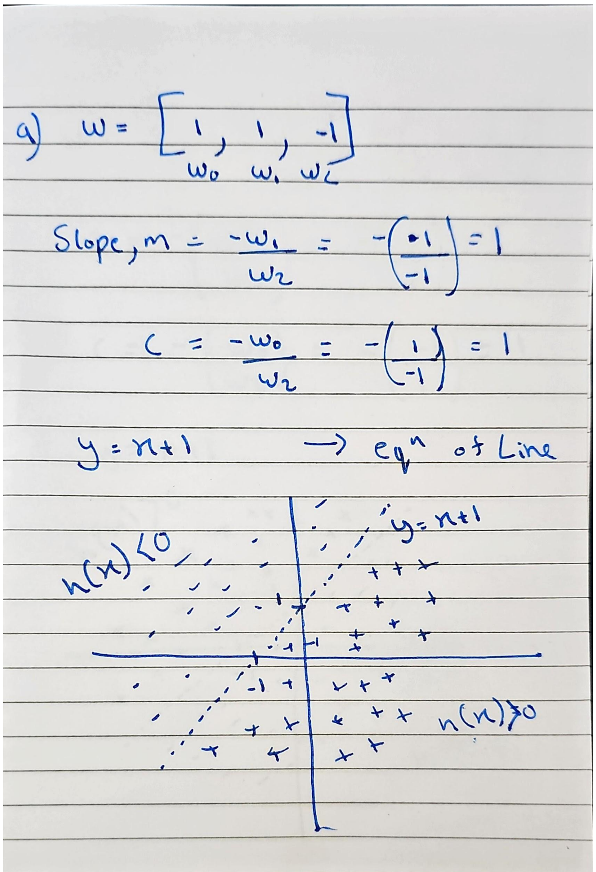
$X_2 = (-W_1 / W_2)X_1 + (-W_0 / W_2)\, C$ - - - - - - - - - - - - - In the same for form as $X_2 = mX_1 + C$

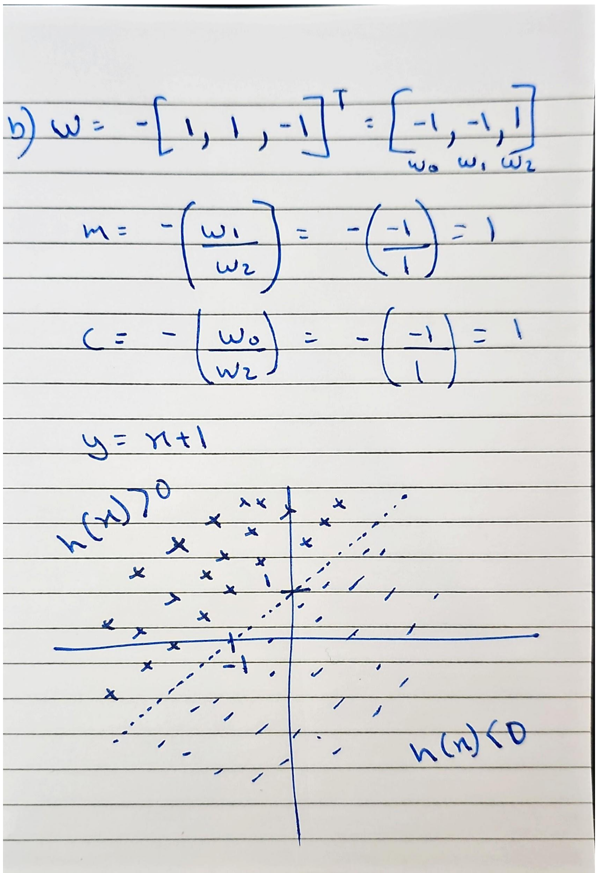
Therefore, $m = (-W_1 / W_2)$ and $C = (-W_0 / W_2)$

LP 2:

[LP 2] Without using a program, with pen and paper, draw the linear separator and the regions where
$h(xx)$ = +1 and $h(xx)$ = −1 when:
a) w = [1 1 − 1]^T
b) w = −[1 1 − 1]^T

Ans:





TASK 3:
Ans.
Refer Code HW1 New Pla.ipynb

Generated a training data of 100 points with final weights of [0.5, 0.3, 0.4].

1. Learning rate is 100: The rate is too high to compute a result
2. Learning rate is 1: The rate is too high to compute a result
3. Learning rate is 0.01: The rate is good enough to generate the result but I got an accuracy of only 56%
4. Learning rate is 0.0001: The rate is good enough to generate the result but I got an accuracy of only 56%

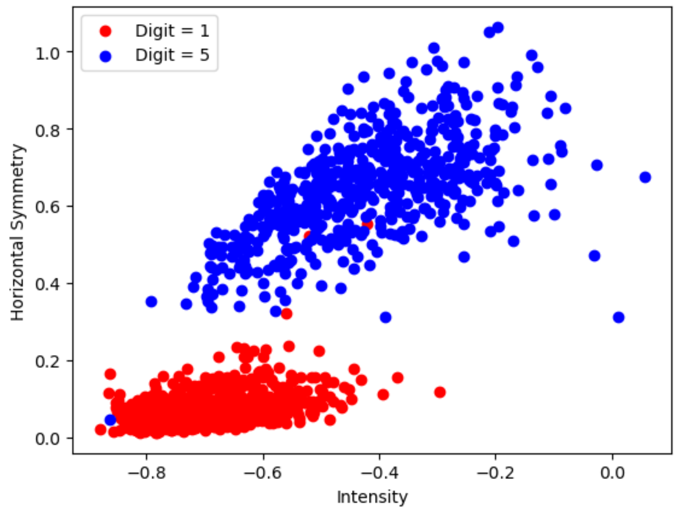
5. Final Learning rate is 0.052: This is the perfect learning rate at which the accuracy of the result is about 97.61%

—--

TASK 4:
Ans.
Refer code HW1 Zip Data.ipynb

In [136]:
```
plt.scatter(intensities_dig1, symmetry_dig1, c="red", label="Digit = {}".format(dig1))
plt.scatter(intensities_dig2, symmetry_dig2, c="blue", label="Digit = {}".format(dig2))

plt.xlabel("Intensity")
plt.ylabel("Horizontal Symmetry")
plt.legend()
plt.show()
```



I have computed the intensities of the pixels by averaging the values and then plotting it on the x axis, and i have used horizontal symmetry to plot it on the y axis.