HW 1 – Machine Learning Introduction

Overview of Machine Learning, Learning Theory, Simple Linear Models

First Submission Due: Monday, September 25th, 11:59pm Pacific Time

Revision Due Dates will be updated after the grades are released

Task 1

Modified version of exercise 1.3 from the textbook on the PLA weight update.

$$w(t+1) = w(t) + y(t)x(t) \quad (1.3)$$

"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] Show that $y(t)w^T(t)x(t) < 0$

Hint: $x(t)$ is misclassified by $w(t)$.

Since x(t) is misclassified, y(t) and y'(t) must have opposite signs.

$y'(t)y(t) < 0$

Since $y'(t) = \text{sign}(w^T(t)x(t))$,

$y(t)\ \text{sign}(w^T(t)x(t)) < 0$

If $y(t) = +1 \rightarrow \text{sign}(w^T(t)x(t)) = -1 \rightarrow w^T(t)x(t) < 0$

So, $y(t)\ w^T(t)x(t) < 0$ is proved.

If $y(t) = -1 \rightarrow \text{sign}(w^T(t)x(t)) = +1 \rightarrow w^T(t)x(t) > 0$

So, $y(t)\ w^T(t)x(t) < 0$ is proved.

Mahima, AgumbeSuresh (2023). 2-ML-concepts_tools-overview. 2-ML-concepts_tools-overview.pdf: FA23: CMPE-257 Sec 02 - Machine Learning (instructure.com)

[LP 2] Show that for w_{final}, the final hypothesis generated by the Perceptron Learning Algorithm, assuming that the data is linearly separable, $y_n w^T{}_{final} x_f > 0$ for all data points x_f.

Assume data is linearly separable means there are some weight vector w such that all data points x is correctly classified, so

sign $(w^T{}_{final} x_f) = y_n$ where y_n is either +1 or -1

$y_n w^T{}_{final} x_f > 0$ is proved.

Mahima, AgumbeSuresh (2023). 2-ML-concepts_tools-overview. 2-ML-concepts_tools-overview.pdf: FA23: CMPE-257 Sec 02 - Machine Learning (instructure.com)

[HP] Show that $y(t)w^T(t+1)x(t) > y(t)w^T(t)x(t)$. 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".

$w(t+1) = w(t) + y(t)x(t)$

Put the above update the weight rule into the equation below:

$y(t)w^T(t+1)x(t) = y(t)[w(t) + y(t)x(t)]^T x(t)$

$= y(t)w^T(t)x(t) + y(t)^2 x(t)^T x(t)$

Since we know that y(t) is either +1 or -1, we can apply to the equation where $y(t)^2 = 1$.

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

From the equation above, we know that $x(t)^T x(t)$ is the dot product of both x(t) so which means it will always be positive unless x(t) is zero vector.

$y(t)w^T(t+1)x(t) > y(t)w^T(t)x(t)$ is proved.

Since we know that x(t) is misclassified by w(t), we will have $y(t)w^T(t)x(t) < 0$. The w(t +1) update will increases $y(t)w^T(t)x(t)$ towards a positive place. So, if this weight update is able to move the $y(t)w^T(t)x(t)$ above zero which is "in the right direction", then the point will be correctly classified.

Mahima, AgumbeSuresh (2023). 2-ML-concepts_tools-overview. 2-ML-concepts_tools-overview.pdf: FA23: CMPE-257 Sec 02 - Machine Learning (instructure.com)

Task 2

Modified version of problem 1.2 from the textbook.

[LP 1] The perceptron in two dimensions is defined as $h(x) = \text{sign}(w^T x)$, where $w = [w_0, w_1, w_2]^T$ and $x = [1, x_1, x_2]^T$. Compare the linear separator to the traditional equation of a line, i.e., $x_2 = mx_1 + c$. Specifically, what is m and c in terms of w_0, w_1, and w_2?

$W_0 (1) + W_1 (x_1) + W_2 (x_2) = 0$ compared with $x_2 = mx_1 + c$.

We will get $x_2 = -\frac{w1}{w2} x_1 - \frac{w0}{w2}$

So, $m = -\frac{w1}{w2}$, $c = -\frac{w0}{w2}$

[LP 2] Without using a program, with pen and paper, draw the linear separator and the regions where $h(x) = +1$ and $h(x) = -1$ when:

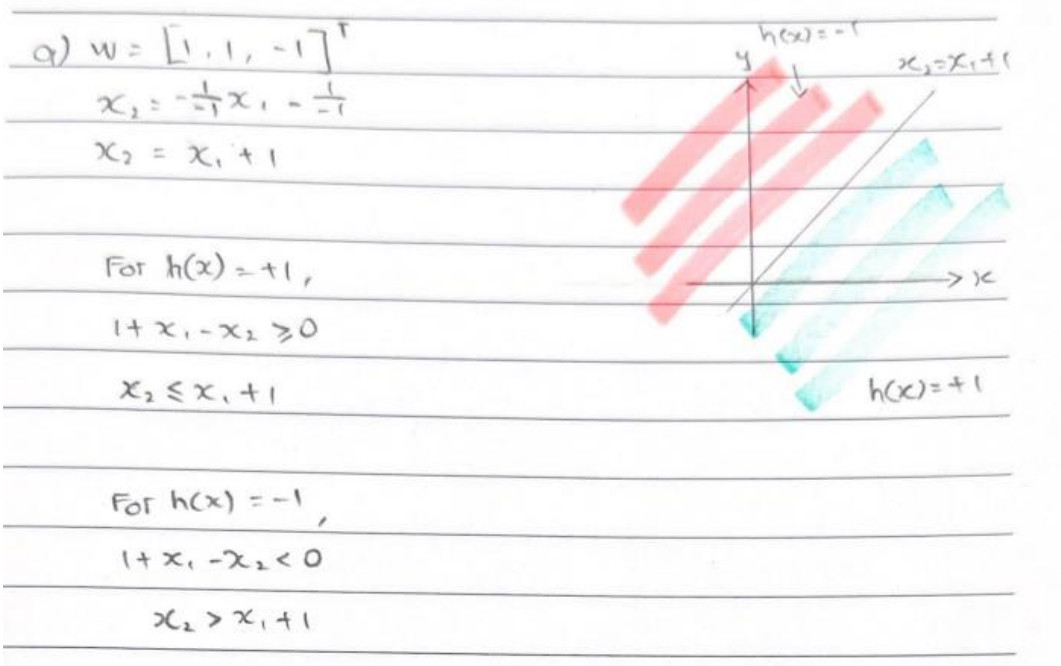
a) $w = [1\ 1 - 1]^T$

$1 + x_1 - x_2 = 0$

$x_2 = x_1 + 1$

For $h(x) = +1$, $1 + x_1 - x_2 \geq 0$, $x_2 \leq x_1 + 1$

For $h(x) = -1$, $1 + x_1 - x_2 < 0$, $x_2 > x_1 + 1$



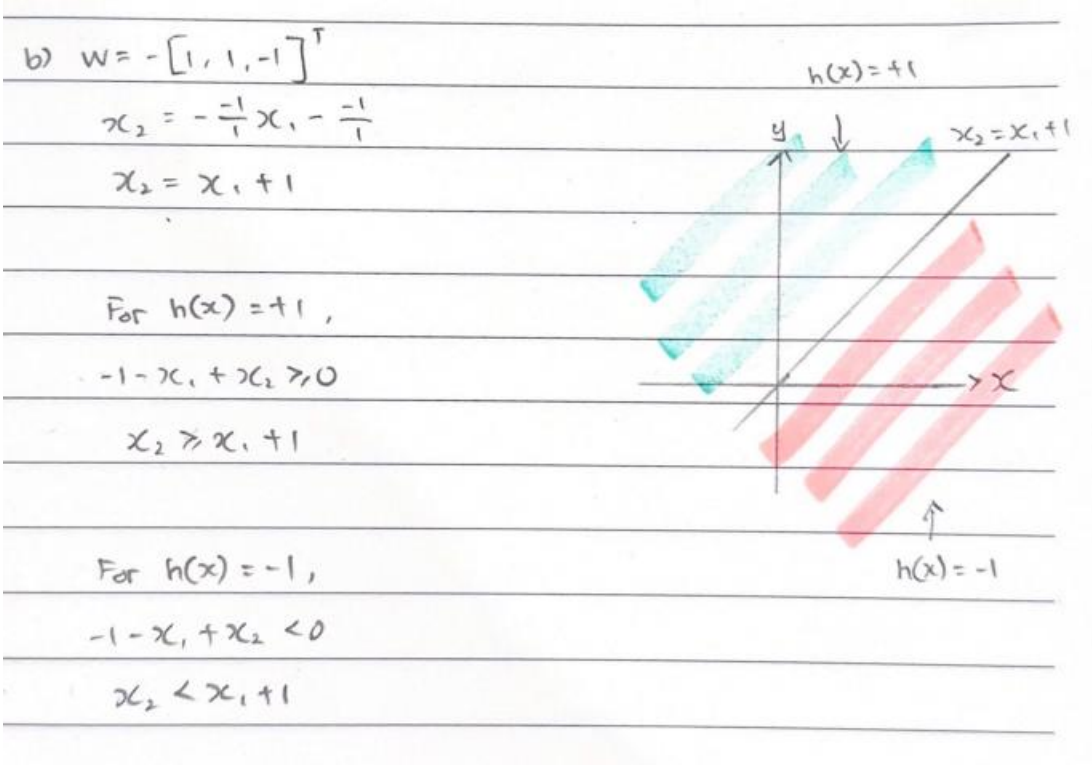
b) $w = -[1\ 1 - 1]^T$

$-1 - x_1 + x_2 = 0$

$x_2 = x_1 + 1$

For h(x) = +1, $-1 - x_1 + x_2 \geq 0$, $x_2 \geq x_1 + 1$

For h(x) = -1, $-1 - x_1 + x_2 < 0$, $x_2 < x_1 + 1$



The difference between the two graphs is from the sign of the weights.

In (a), the weights push the positive region below the line, while in (b), the negative weights push the positive region above the line.

Task 3

[HP] Problem 1.5 from the textbook. The code to generate the synthetic dataset is provided in the attached file. You are recommended to try writing your own code to generate the dataset. Feel free to use the provided code (with citation).

Problem 1.5 The perceptron learning algorithm works like this: In each iteration t, pick a random $(\mathbf{x}(t), y(t))$ and compute the 'signal' $s(t) = \mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t)$. If $y(t) \cdot s(t) \leq 0$, update $\mathbf{w}$ by

$$\mathbf{w}(t+1) \longleftarrow \mathbf{w}(t) + y(t) \cdot \mathbf{x}(t) \ ;$$

One may argue that this algorithm does not take the 'closeness' between $s(t)$ and $y(t)$ into consideration. Let's look at another perceptron learning algorithm: In each iteration, pick a random $(\mathbf{x}(t), y(t))$ and compute $s(t)$. If $y(t) \cdot s(t) \leq 1$, update $\mathbf{w}$ by

$$\mathbf{w}(t+1) \longleftarrow \mathbf{w}(t) + \eta \cdot (y(t) \quad s(t)) \cdot \mathbf{x}(t) \ ,$$

where η is a constant. That is, if $s(t)$ agrees with $y(t)$ well (their product is > 1), the algorithm does nothing. On the other hand, if $s(t)$ is further from $y(t)$, the algorithm changes $\mathbf{w}(t)$ more. In this problem, you are asked to implement this algorithm and study its performance.

(a) Generate a training data set of size 100 similar to that used in Exercise 1.4. Generate a test data set of size $10,000$ from the same process. To get g, run the algorithm above with $\eta = 100$ on the training data set, until a maximum of $1,000$ updates has been reached. Plot the training data set, the target function f, and the final hypothesis g on the same figure. Report the error on the test set.

(b) Use the data set in (a) and redo everything with $\eta = 1$.

(c) Use the data set in (a) and redo everything with $\eta = 0.01$.

(d) Use the data set in (a) and redo everything with $\eta = 0.0001$.

(e) Compare the results that you get from (a) to (d).

The comparison may be qualitative, rather than quantitative.

Task 3 answers are in the synthetic Data Generator.ipynb.

CMPE-257-Fall23-Jeffrey-Ong/Synthetic_Data_Generator.ipynb at homework-1 · kwanqing/CMPE-257-Fall23-Jeffrey-Ong (github.com)

(e)

A high n might result in a higher test error when the model overfits the training data while a low n might result in a higher test error as well when the model underfits the data due to insufficient updates.

For n = 100, There are some errors occured like when running the code with n = 100:

overflow encountered in double_scalars w += n * (y - s) * x

invalid value encountered in add w += n * (y - s) * x

When we use a large value for n (like 100), the weight updates in the Perceptron Learning Algorithm can become very large, especially if the dot product of s and y are of opposite signs. This can cause the weights to grow exponentially which leads to overflow.

For n = 1, I am getting errors as well. It is most likely because the features in the dataset have large values which cause large weight updates that potentially lead to overflow. Next, when the data is not linearly separable, the PLA will keep updating the weights to find a separating hyperplane which doesn't exist and cause the weights to grow large.

For n = 0.01, The algorithm will take more iterations to converge compared to n=1. Smaller updates make the algorithm less susceptible to noise in the data and the final hypothesis g is more stable across runs compared to n=1.

For n = 0.0001, The algorithm takes a very long time to converge, and it is very stable and less sensitive to noise because the updates are so small that the algorithm is not fit the data well within a reasonable number of iterations.

Task 4

[HP] Work on the digits dataset to classify handwritten digits using the provided Digits.zip on Canvas. ZipDigits.train is the training dataset and ZipDigits.test is the test dataset. The first column in the dataset is the digit label, and the next 256 columns are values between -1 and 1 representing a grayscale image.

You will need to preprocess the data to get these labels. Take some time to familiarize yourself with the dataset. We will work on this dataset throughout the course. Do the following preprocessing steps for both training and testing datasets:

Filter the dataset to include only digits labeled as '1' and '5'. Convert the labels into labels for binary classification, i.e., '1' and '-1'.

Extract intensity and symmetry features on the dataset as discussed in the class. You may use your own mathematical definition of the two features. In the written part of the submission, include the definition you used.

For the training dataset, plot a 2D scatter plot with the two features you extracted. Use different colors and/or markers for the different classes. Submit the plot on the written part of your submission.

Task 4 answers are in the task_4.ipynb.

CMPE-257-Fall23-Jeffrey-Ong/Task_4.ipynb at homework-1 · kwanqing/CMPE-257-Fall23-Jeffrey-Ong (github.com)

To extract the intensity and symmetry features from the dataset, I used the link below as a reference.

Intensity is the average of all the pixels values in the image. So, we find the mean values of it.

$$\text{Intensity} = \frac{\sum_{i=1}^{256} pixel_i}{256}$$

Symmetry is the negative of the average absolute difference between the pixels on either side of the vertical centerline of the image.

$$\text{Symmetry} = -\frac{1}{128}\sum_{i=1}^{128} |pixel_i - pixel_{257-i}|$$

reference: SrijoniChakra/Digits-data-classification-using-intensity-and-symmetry-with-neural-network (github.com)

Submission instructions

As mentioned in class activity 1, we will use GitHub to share code and track progress in this class. If you have not already done so, create a GitHub private repository for the course and name it "CMPE257-Fall23-FirstName-LastName". Add the following users to the repository: mahima-as and rbpravin. You are also welcome to add the instructor and ISA to your Colab notebooks.

In your GitHub repo, create a branch called *homework-1*. Frequently commit your code and make sure it is shared with the instructor and ISA. Include a link to the GitHub repository in your submission pdf.

Specifications

All four tasks have components labeled [LP] and [HP]. If you complete ALL the LP components satisfactorily, you will receive a grade of "low pass" on the homework. If you complete ALL the LP components and at least 2/3 HP components satisfactorily, you will receive a grade of "high pass". If you do not meet the criteria for a "low pass", the submission will be marked as "revision needed".

Note the following statements from the syllabus:

If a student receives a "low pass" or "revision needed" grade, the student may revise and resubmit their homework assignment by using one "token".

For homework assignments, if the student fails to submit their assignment by the posted deadline, their submission will receive a grade of "revision needed". If they fail to submit the assignment by the revision deadline, the submission will receive a grade of "fail".

At most two tokens may be used for the one-day deadline extensions (one token for each one-day extension), including the revision deadlines. Tokens will be automatically removed from your wallet if you submit late and/or resubmit.

VERY IMPORTANT: Include ALL the references you used for this assignment, including names of classmates you discuss with. Failure to cite your sources counts as an act of academic dishonesty and will be taken seriously without zero tolerance. You will automatically receive a "fail" grade in the homework and further serious penalties may be imposed.

NOTE: You can look for help on the Internet but refrain from referencing too much. Please cite all your sources in your submission.

When you submit your assignment, you automatically agree to the following statement. If you do not agree, it is your responsibility to provide the reason.

"*I affirm that I have neither given nor received unauthorized help in completing this homework. I am not aware of others receiving such help. I have cited all the sources in the solution file.*"