Homework #1

RELEASE DATE: 03/09/2023

DUE DATE: 03/30/2023, BEFORE 13:00 on Gradescope

QUESTIONS ARE WELCOMED ON DISCORD.

You will use Gradescope to upload your choices and your scanned/printed solutions. For problems marked with (*), please follow the guidelines on the course website and upload your source code to Gradescope as well. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

This homework set comes with 20 problems and a total of 500 points. For each problem, there is one correct choice. If you choose the correct answer, you get 20 points; if you choose an incorrect answer, you get 0 points. For five of the secretly-selected problems, the TAs will grade your detailed solution in terms of the written explanations and/or code based on how logical/clear your solution is. Each of the five problems graded by the TAs counts as additional 20 points (in addition to the correct/incorrect choices you made). In general, each homework (except homework 0) is of a total of 500 points.

Basics of Machine Learning

1. Which of the following tasks is best suited for machine learning? Choose the best answer; explain your answer.

[a] create a voice assistant that understands and reacts to users' requests

[b] predict whether Schrödinger's cat is alive or dead inside the box

[c] search for the shortest path to exit a maze

[d] generate an image of Zeus that matches his actual facial look

[e] none of the other choices

2. We learned about the Perceptron Learning Algorithm (PLA), which comes with an update rule of

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot 1 \quad .$$

However, the algorithm does not take the scale of updating into consideration and always takes a step (which will be called the learning rate in future classes) of 1 regardless of the value of $y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}$. Then, after updating, $\mathbf{w}_{t+1}^T$ can still be wrong on $(\mathbf{x}_{n(t)}, y_{n(t)})$. Which one of the following update rules ensures that $\mathbf{w}_{t+1}^T$ is correct on $(\mathbf{x}_{n(t)}, y_{n(t)})$? Choose the correct answer; prove your answer.

[a] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot 1126$

[b] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot \left(\frac{-y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}}{\|\mathbf{x}_{n(t)}\|^2}\right)$

[c] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot \left\lceil \frac{-y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}}{\|\mathbf{x}_{n(t)}\|^2} \right\rceil$

[d] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot \left\lfloor \frac{-y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}}{\|\mathbf{x}_{n(t)}\|^2} + 1 \right\rfloor$

[e] none of the other choices

3. Dr. Norman thinks PLA will be highly influenced by very long examples, as $\mathbf{w}_t$ changes drastically if $\|\mathbf{x}_{n(t)}\|$ is large. Hence, she decides to preprocess the training data by normalizing each input vector i.e., $\mathbf{z}_n \leftarrow \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|}$. How does PLA's upper bound on Page 41 of Lecture 1 change with this preprocessing procedure in terms of $\rho_{\mathbf{z}} = \min_n \frac{y_n\mathbf{w}_f^T\mathbf{z}_n}{\|\mathbf{w}_f\|}$? Choose the correct answer; prove your answer.

[a] $\frac{1}{\sqrt{\rho_{\mathbf{z}}}}$

[b] $\frac{1}{\rho_{\mathbf{z}}}$

[c] $\frac{1}{\rho_{\mathbf{z}}^2}$

[d] ∞ (i.e., PLA might never terminate)

[e] none of the other choices

4. Following the previous problem, Dr. Norman would like to compare her method with the original PLA. What is the relationship between the new bound U in the previous problem and the original bound $U_{\text{orig}} = \left(\frac{R}{\rho}\right)^2$? Choose the correct answer; prove your answer.

[a] $U < U_{\text{orig}}$

[b] $U \leq U_{\text{orig}}$

[c] $U = U_{\text{orig}}$

[d] $U \geq U_{\text{orig}}$

[e] none of the other choices

5. In PLA, we define $\rho = \min_n y_n \mathbf{w}_f^T \mathbf{x}_n$, and ρ is often called the **margin** of $\mathbf{w}_f$. Margin also means the minimal distance between $\mathbf{x}_n$ and hyperplane $\mathbf{w}_f$, and will be a main concept when we introduce the Support Vector Machine (SVM) later in this class. Before that, let us play with a variant of PLA, the Perceptron Algorithm using Margins (PAM). The difference between PAM and the original PLA is that PAM updates on $\mathbf{x}_n$ if

$$y_n \mathbf{w}_t^T \mathbf{x}_n < \tau,$$

where τ is the margin we would like to achieve. Please simulate PLA (take sign(0) as +1) and PAM with $\tau = 5$ by checking the following five training examples **orderly** with $\mathbf{w}_0 = \mathbf{0}$, from the first example to the fifth one. Stop the algorithm after checking the fifth example. That is, there is no need to cycle back to the first example. Then, use the resulting perceptron from the algorithm to predict on the test examples. How many test examples are wrongly predicted by PLA but correctly predicted by PAM? Choose the correct answer; list your derivation steps.

Training Examples

$\mathbf{x}$	y
$(1, -2, 2)$	-1
$(1, -1, 2)$	-1
$(1, 2, 0)$	1
$(1, -1, 0)$	-1
$(1, 1, 1)$	1

Test Examples

$\mathbf{x}$	y
$(1, \frac{1}{2}, 2)$	1
$(1, \frac{1}{4}, 1)$	1
$(1, \frac{1}{2}, 0)$	1
$(1, -\frac{1}{2}, 1)$	-1

[a] 4

[b] 3

[c] 2

[d] 1

[e] 0

The Learning Problems

6. On Page 14 of Lecture 1, we introduced the recommender problem, which aims to predict how a viewer would rate a certain movie, say, with real numbers within $[1, 5]$, given existing ratings from some viewers. Which of the following best describes the associated learning problem? Choose the best answer; explain your answer.

[a] supervised regression

[b] supervised multi-class classification

[c] clustering

[d] self-supervised learning

[e] none of the other choices

7. On Page 19 of Lecture 2, we described that the second step of ChatGPT samples a prompt from the dataset, and then asks a (human) labeler to compare several possible outputs from the model. Assume that the labeler is always fed with two possible outputs and needs to select the better one. Which of the following best describes the associated learning problem? Choose the best answer; explain your answer.

[a] outlier detection

[b] binary classification

[c] clustering

[d] self-supervised learning

[e] multilabel classification

Feasibility of Learning

8. As discussed on Page 9 of lecture 3, what we really care about is whether $g \approx f$ *outside* $\mathcal{D}$. For a set of "universe" examples $\mathcal{U}$ with $\mathcal{D} \subset \mathcal{U}$, the error *outside* $\mathcal{D}$ is typically called the Off-Training-Set (OTS) error

$$E_{\text{ots}}(h) = \frac{1}{|\mathcal{U} \setminus \mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{U}\setminus\mathcal{D}} [\![h(\mathbf{x}) \neq y]\!] .$$

Consider $\mathcal{U}$ with 6 examples

$\mathbf{x}$	y
$(1,0)$,	$+1$
$(3,2)$,	$+1$
$(0,2)$,	$+1$
$(2,3)$,	-1
$(2,4)$,	-1
$(3,5)$,	-1

Run the process of choosing any three examples from $\mathcal{U}$ as $\mathcal{D}$, and learn a perceptron hypothesis (say, with PLA, or any of your "human learning" algorithm) to achieve $E_{\text{in}}(g) = 0$ on $\mathcal{D}$. Then, evaluate g outside $\mathcal{D}$. What is the smallest and largest $E_{\text{ots}}(g)$? Choose the correct answer; explain your answer.

[a] $(0, \frac{1}{3})$
[b] $(0, \frac{2}{3})$
[c] $(\frac{2}{3}, 1)$
[d] $(\frac{1}{3}, 1)$
[e] $(0, 1)$

9. Consider $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$ and the target function $f(\mathbf{x}) = -\text{sign}(x_1^2 + x_2^2 - 0.25)$. With two candidate hypothesis $h_1(\mathbf{x}) = -\text{sign}(x_1^2 + x_2^2 - 1)$ and $h_2(\mathbf{x}) = -\text{sign}(|x_1| + |x_2| - 0.5)$. What is the $(E_{\text{out}}(h_1), E_{\text{out}}(h_2))$ subject to a uniform distribution in $[+1, -1] \times [+1, -1]$ that generates $\mathbf{x}$? Choose the correct answer; write down your derivation steps.

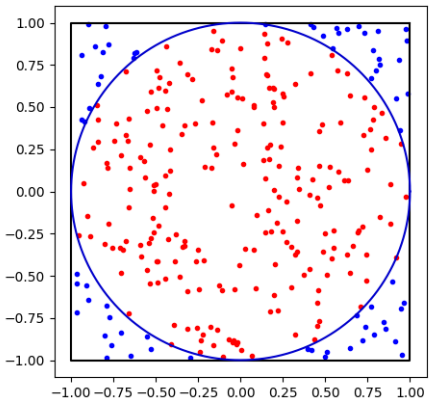
[a] $(\frac{5\pi}{16}, \frac{\pi-2}{8})$
[b] $(\frac{3\pi}{16}, \frac{\pi-2}{8})$
[c] $(\frac{5\pi}{16}, \frac{\pi-2}{16})$
[d] $(\frac{3\pi}{16}, \frac{\pi-2}{16})$
[e] none of the other choices

10. Following the previous problem, when drawing 4 examples independently and uniformly within $[-1, +1] \times [-1, +1]$ as $\mathcal{D}$, what is the probability that we get 4 examples such that $E_{\text{in}}(h_2) = E_{\text{in}}(h_1) = 0$? Choose the closest answer; explain your answer.

Note: This is one of the BAD-data cases where we cannot distinguish the better-E_{out} hypothesis h_2 from the worse hypothesis h_1.

[a] 0.00
[b] 0.01
[c] 0.02
[d] 0.03
[e] 0.04

11. One well-known method to estimate the value of π is by using Monte Carlo simulations. Given a dartboard of size 2×2 as shown in the figure below, the area of the square and the circle are 4 and π respectively. The value of π is then estimated by $4 \times \frac{\text{number of darts landed in the circle}}{\text{total number of darts}}$. Based on the version of Hoeffding's Inequality introduced in our class, which of the following is the smallest number of darts (among the choices) you should throw to ensure your estimation error is within 10^{-2} with probability more than 0.99? Choose the smallest value among the choices as your answer; list your derivation steps.



[a] 26492
[b] 105967
[c] 314159
[d] 423866
[e] none of the other choices

12. Consider a game with M boxes $b_1, b_2, \ldots, b_M$. In each round of the game, you can pick one of the boxes b_m, and get 1 dollar from that box with a fixed probability p_m, independent of other rounds in the game. The box with the largest "reward" probability p_{m^*} is denoted as b_{m^*}. Any box is called ϵ-optimal if its p_m is within ϵ to p_{m^*}. That is,

$$p_m \geq p_{m^*} - \epsilon.$$

The following is an algorithm that guarantees to return an ϵ-optimal box with probability at least $1 - \delta$ if N is large enough.

Algorithm

1. For every box $b_m, m \in 1, ..., M$, open it N times.
2. Let c_m be the total number of coins collected from box b_m.
3. Output $m = \text{argmax}_{m \in \{1,\ldots,M\}} \{c_m\}$

Which of the following N is big enough to return an ϵ-optimal box with probability at least $1 - \delta$? (*Hint: Multiple-bin Hoeffding!*) Choose the correct answer; prove your answer.

[a] $\frac{1}{\epsilon^2} \ln \frac{2M}{\delta}$
[b] $\frac{1}{\epsilon^2} \ln \frac{M}{\delta}$
[c] $\frac{1}{2\epsilon^2} \ln \frac{M}{\delta}$
[d] $\frac{2}{\epsilon^2} \ln \frac{2M}{\delta}$
[e] none of the other choices

Experiments with Perceptron Learning Algorithm

Next, we use an artificial data set to study PLA. The data set with $N = 256$ examples is in

http://www.csie.ntu.edu.tw/~htlin/course/ml23spring/hw1/hw1_train.dat

Each line of the data set contains one $(\mathbf{x}_n, y_n)$ with $\mathbf{x}_n \in \mathbb{R}^{10}$. The first 10 numbers of the line contains the components of $\mathbf{x}_n$ orderly, the last number is y_n. Please initialize your algorithm with $\mathbf{w} = \mathbf{0}$ and take sign(0) as +1.

13. (*) Please first follow Page 29 of Lecture 1, and add $x_0 = 1$ to every $\mathbf{x}_n$. Implement a version of PLA that randomly picks an example $(\mathbf{x}_n, y_n)$ in every iteration, and updates $\mathbf{w}_t$ if and only if $\mathbf{w}_t$ is incorrect on the example. Note that the random picking can be simply implemented *with replacement*—that is, the same example can be picked multiple times, even consecutively. Stop updating and return $\mathbf{w}_t$ as $\mathbf{w}_{\text{PLA}}$ if $\mathbf{w}_t$ is correct consecutively after checking M randomly-picked examples.

Hint: (1) The update procedure described above is equivalent to the procedure of gathering all the incorrect examples first and then randomly picking an example among the incorrect ones. But the description above is usually much easier to implement. (2) The stopping criterion above is a randomized, more efficient implementation of checking whether $\mathbf{w}_t$ makes no mistakes on the data set.

Take $M = N/2$. Repeat your experiment for 1000 times, each with a different random seed. What is the average $E_{\text{in}}(\mathbf{w}_{\text{PLA}})$? Choose the closest value; upload your source code.

[a] 0.01
[b] 0.02
[c] 0.04
[d] 0.08
[e] 0.16

14. (*) Following the previous problem. Take $M = 4N$ instead. Repeat your experiment for 1000 times, each with a different random seed. What is the average $E_{\text{in}}(\mathbf{w}_{\text{PLA}})$? Choose the closest value; upload your source code.

[a] 0.00020
[b] 0.00080
[c] 0.00320
[d] 0.01260
[e] 0.05040

15. (*) Following the previous problem. Take $M = 4N$. When running PLA, record the number of updates (i.e. step 2 of the PLA algorithm, only when encountering an incorrect example). Repeat your experiment for 1000 times, each with a different random seed. What is the **median** number of updates? Choose the closest value; upload your source code.

[a] 50
[b] 100
[c] 200
[d] 400
[e] 800

16. (*) Among all the w_0 (the zero-th component of $\mathbf{w}_{\text{PLA}}$) obtained from the 1000 experiments above, what is the **median**? Choose the closest value; upload your source code.

[a] -40
[b] -20
[c] 0
[d] 20
[e] 40

17. (*) Scale down each $\mathbf{x}_n$ by 2, including scaling each x_0 from 1 to 0.5. Then, run PLA on the scaled examples for 1000 experiments, each with a different random seed. Take $M = 4N$. What is the **median** number of updates? Choose the closest value; upload your source code.

[a] 50
[b] 100
[c] 200
[d] 400
[e] 800

18. (*) Set $x_0 = 0$ to every $\mathbf{x}_n$ instead of $x_0 = 1$, and do not do any scaling. This equivalently means not adding any x_0, and you will get a separating hyperplane that passes the origin. Repeat the 1000 experiments above with $M = 4N$. What is the **median** number of updates? Choose the closest value. Choose the closest value; upload your source code.

[a] 50
[b] 100
[c] 200
[d] 400
[e] 800

19. (*) Set $x_0 = -1$ to every $\mathbf{x}_n$ instead of $x_0 = 1$, and do not do any scaling. Repeat the 1000 experiments above with $M = 4N$. Among all the $w_0 \cdot x_0$ (where w_0 is the zero-th component of $\mathbf{w}_{\text{PLA}}$) values obtained from the 1000 experiments above, what is the **median**? Choose the closest value; upload your source code.

[a] -40
[b] -20
[c] 0
[d] 20
[e] 40

20. (*) Set $x_0 = 0.1126$ to every $\mathbf{x}_n$ instead of $x_0 = 1$, and do not do any scaling. Repeat the 1000 experiments above with $M = 4N$. Among all the $w_0 \cdot x_0$ (where w_0 is the zero-th component of $\mathbf{w}_{\text{PLA}}$) values obtained from the 1000 experiments above, what is the **median**? Choose the closest value; upload your source code.

[a] 0.2
[b] 0.3
[c] 0.4
[d] 0.5
[e] 0.6