HOMEWORK 3
DECISION TREES, K-NN, PERCEPTRON, REGRESSION

10-301 / 10-601 INTRODUCTION TO MACHINE LEARNING (SUMMER 2022)
https://www.cs.cmu.edu/~hchai2/courses/10601/

OUT: Tuesday, May 31
DUE: Tuesday, June 7 at 1:00 PM
TAs: Sana, Brendon, Ayush, Boyang (Jack), Chu

Summary It's time to practice what you've learned! In this assignment, you will answer questions on topics we've covered in class so far, including Decision Trees, K-Nearest Neighbors, Perceptron, and Linear Regression. This assignment consists of a written component split into four sections, one for each topic. These questions are designed to test your understanding of the theoretical and mathematical concepts related to each topic. For each topic, you will also apply your understanding of the concept to the related ideas such as overfitting, error rates, and model selection. This homework is designed to help you apply what you've learned and solve a few concrete problems.

START HERE: Instructions

- **Collaboration Policy**: Please read the collaboration policy here: https://www.cs.cmu.edu/~hchai2/courses/10601/#Syllabus
- **Late Submission Policy:** See the late submission policy here: https://www.cs.cmu.edu/~hchai2/courses/10601/#Syllabus
- **Submitting your work:** You will use Gradescope to submit answers to all questions. Please follow instructions at the end of this PDF to correctly submit all your code to Gradescope.
  - **Written:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, please use the provided template. Submissions can be handwritten onto the template, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in LaTeX. Each derivation/proof should be completed in the boxes provided. You are responsible for ensuring that your submission contains exactly the same number of pages and the same alignment as our PDF template. If you do not follow the template, your assignment may not be graded correctly by our AI assisted grader.

Instructions for Specific Problem Types

For "Select One" questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

- ● Henry Chai
- ○ Marie Curie
- ○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

- ● Henry Chai
- ○ Marie Curie
- ~~●~~ Noam Chomsky

For "Select all that apply" questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

- ■ Stephen Hawking
- ■ Albert Einstein
- ■ Isaac Newton
- □ I don't know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

- ■ Stephen Hawking
- ■ Albert Einstein
- ■ Isaac Newton
- ~~■~~ I don't know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-601	10-~~6~~301

1 Latex Bonus Point (1 points)

1. (1 point) **Select one:** Did you use Latex for the entire written portion of this homework?

   ○ Yes

   ● No

2 Decision Trees (Revisited) (14 points)

1. Consider the following 4×4 checkerboard pattern. Suppose our goal is to perfectly classify the range shown such that all black regions are labeled as $+1$ and all white regions are labeled as -1. Let the horizontal axis denote feature x_1 and vertical axis denote feature x_2.

   NOTE: If a point is on the border or corner of a region, it has the same label as the region that is above it and/or to its right. For example, $(1, 0)$ has label $+1$, $(1, 1)$ has label -1, and $(1, 1.5)$ has label -1.

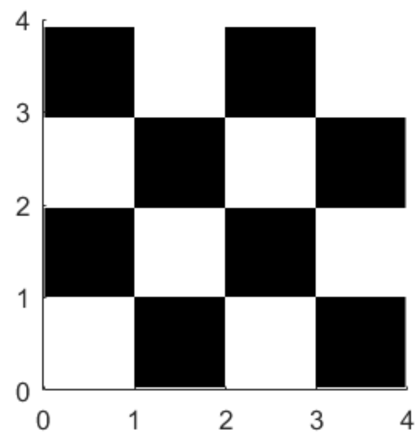


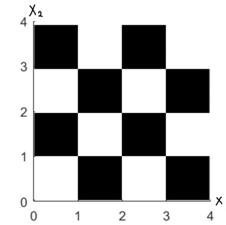
Figure 1: Checkerboard Pattern

   (a) (2 points) What is the minimum depth of a binary decision tree that perfectly classifies the colored regions in Figure 1, using features that only inspect either x_1 or x_2, but not both x_1 and x_2?

   Your Answer

   3

(b) (2 points) What is the minimum depth of a binary decision tree that perfectly classifies the colored regions in Figure 1, using features of the form $x_1 < c$ or of the form $x_2 < c$? Different features may use different values of c.



Your Answer

4

(c) (2 points) What is the minimum depth of a binary decision tree that perfectly classifies the colored regions in Figure 1, using any features of x_1, x_2, or both?

Your Answer

1

2. Consider the graph below showing accuracy plotted against tree size for a decision tree which has been pruned back to the vertical (red) line. Assume all of our labeled data for this task was randomly divided into a training dataset D_{train}, a validation data set D_{val}, and a test dataset D_{test}. The full tree was trained only on D_{train}, then reduced-error pruning was applied using D_{val}.

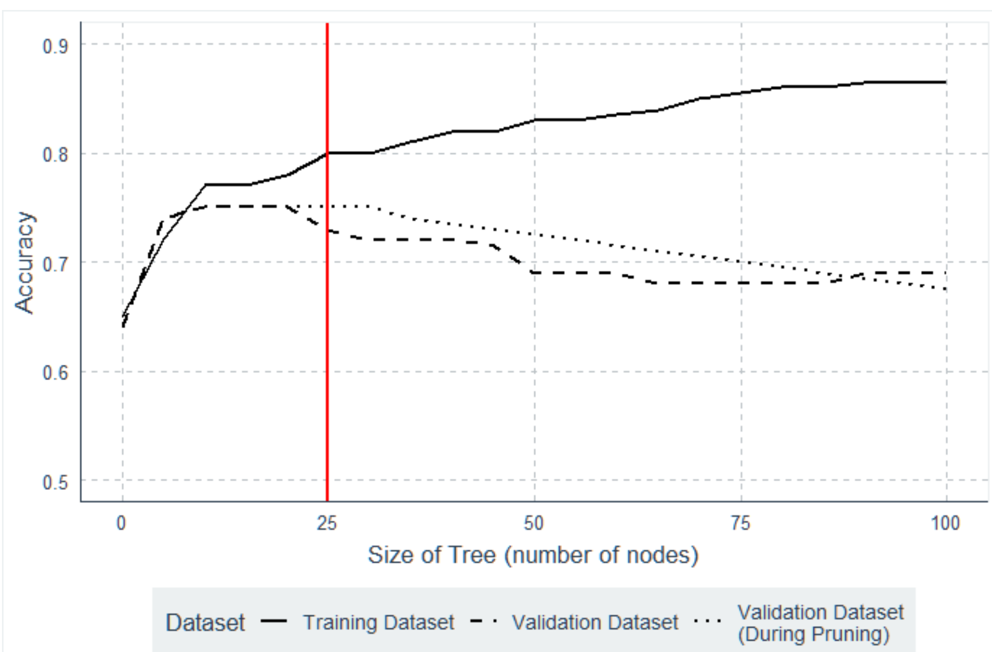


Figure 2: Accuracy vs Decision Tree Size for varying datasets

(a) (1 point) **Select one:** Refer to Figure 2. Note that D_{test} was not used for training or pruning. When the size of the pruned tree is at **25 nodes**, what is its accuracy on D_{test} likely to be?

- ○ Slightly higher than the pruned tree's accuracy on D_{val}
- ○ The same as the pruned tree's accuracy on D_{val}
- ● Slightly lower than the pruned tree's accuracy on D_{val}

(b) (1 point) **Select one:** Which of the following gives us the best approximation of the true error?

- ○ Error corresponding to training data D_{train}
- ○ Error corresponding to validation data D_{val}
- ● Error corresponding to test data D_{test}

3. (2 points) **Select all that apply:** Which of the following are valid ways to avoid overfitting?

- ☐ Decrease the training set size.
- ■ Set a threshold for a minimum number of examples required to split at an internal node.
- ■ Prune the tree so that cross-validation error is minimized.
- ☐ Maximize the tree depth.
- ☐ None of the above.

4. Consider a binary classification problem using 1-nearest neighbors with the Euclidean distance metric. We have N 1-dimensional training points $x^{(1)}, x^{(2)}, \ldots x^{(N)}$ and corresponding labels $y^{(1)}, y^{(2)}, \ldots y^{(N)}$, with $x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \{0, 1\}$.

   Assume the points $x^{(1)}, x^{(2)}, \ldots x^{(N)}$ are in ascending order by value. If there are ties during the 1-NN algorithm, we break ties by choosing the label corresponding to the $x^{(i)}$ with lower value.

   (a) (2 points) **Select one:** Is it possible to build a decision tree that behaves exactly the same as the 1-nearest neighbor classifier? Assume that the decision at each node takes the form of "$x \leq t$ or $x > t$," where $t \in \mathbb{R}$.

   ● Yes

   ○ No

   If your answer is yes, please explain how you will construct the decision tree. If your answer is no, explain why it's not possible.

   Your answer:

   the decision tree would be constructed with nodes where t is the mean value of two consecutive $x^{(i)}$ values of the set. Thus, the first node would check if $x \leq (x^{(1)} + x^{(2)})/2$. If the inequality is true, the predicted label would be the label of the lower $x^{(i)}$, in this case $x^{(1)}$, and if the inequality is false, the next node would check if $x \leq (x^{(2)} + x^{(3)})/2$. This pattern would continue until $x^{(n)}$

   (b) (2 points) Let's add a dimension! Now assume the training points are 2-dimensional where $\mathbf{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}\right) \in \mathbb{R}^2$ and the decision at each node takes the form of "$x_j \leq t$ or $x_j > t$," where $t \in \mathbb{R}$ and $j \in \{1, 2\}$. Give an example with at most 3 training points for which it isn't possible to build a decision tree that behaves exactly the same as a 1-nearest neighbor classifier.

   Your answer:

x_1	x_2	y
2	2	0
2	2	1

3 k-Nearest Neighbors (20 points)

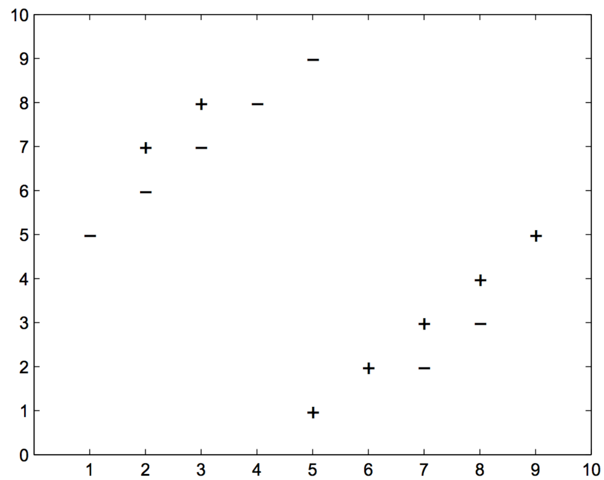


Figure 3: k-NN Dataset

1. Consider a k-nearest neighbors (k-NN) binary classifier which assigns the class of a test point to be the class of the majority of the k-nearest neighbors, according to the Euclidean distance metric. Assume that ties are broken by selecting one of the labels uniformly at random.

   (a) (2 points) Using Figure 3 shown above to train the classifier and choosing $k = 6$, what is the classification error on the training set? **Round to 4 decimal places.**

   Your answer:

   0.429

(b) (2 points) **Select all that apply:** Let's say that we have a new test point (not present in our training data) $\mathbf{x}^{\text{new}} = [3, 9]^T$ that we would like to apply our k-NN classifier to, as seen in figure 4.

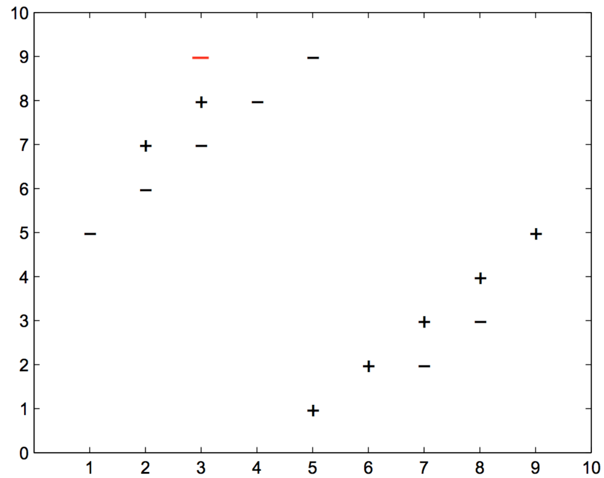


Figure 4: k-NN Dataset with Test Point

For which values of k is this test point correctly classified by the k-NN algorithm?

☐ $k = 1$

■ $k = 5$

■ $k = 9$

☐ $k = 12$

☐ None of the above

2. **Select one:** Assume we have large labeled dataset that is randomly divided into a training set and a test set, and we would like to classify points in the test set using a k-NN classifier.

(a) (1 point) In order to minimize the classification error on this test set, we should always choose the value of k which minimizes the training set error.

○ True

● False

(b) (2 points) **Select one:** Instead of choosing the hyper-parameters by merely minimizing the training set error, we instead consider splitting the training-all data set into a training and a validation data set, and choosing the hyper-parameters that lead to lower validation error. Is choosing hyper-parameters based on validation error better than choosing hyper-parameters based on training error?

- ● Yes, lowering validation error instead of training error is better because lowering training error will not help generalize our model and may lead to overfitting.
- ○ Yes, lowering validation error is better for the model because cross-validation guarantees a better test error.
- ○ No, lowering training error instead of validation error is better because lowering validation error will not help generalize our model and may lead to overfitting.
- ○ No, lowering training error is better for the model because we have to learn the training set as well as possible to guarantee the best possible test error.

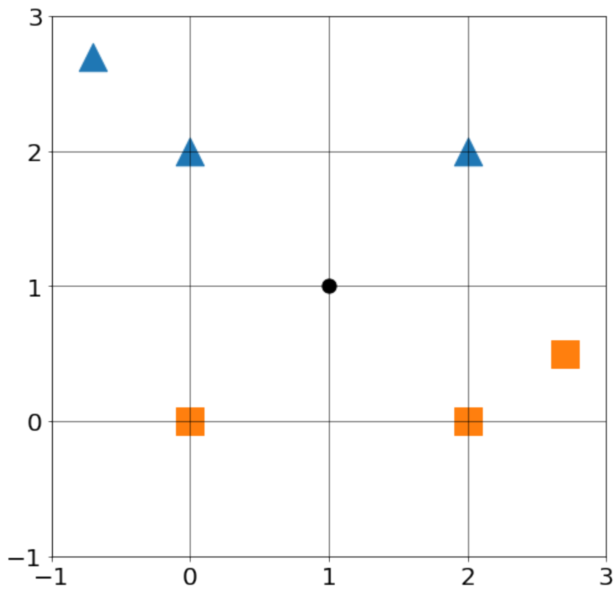
(c) (2 points) **Select one:** Your friend Sally suggests that instead of splitting the original training set into separate training and validation sets, we should instead use the test set as the validation data for choosing hyper-parameters. Is this a good idea? Justify your opinion with no more than 3 sentences.

- ○ Yes
- ● No

Your answer:

I do not think this is a good idea because using the test set to train the model can lead to overfitting.

3. (2 points) **Select all that apply:** Consider a binary k-NN classifier where $k = 4$ and the two labels are “triangle” and “square”. Consider classifying a new point $\mathbf{x} = (1, 1)$, where two of the $\mathbf{x}$’s nearest neighbors are labeled “triangle” and two are labeled “square” as shown below.



Which of the following methods can be used to break ties or avoid ties on this dataset?

- □ Assign $\mathbf{x}$ the label of its nearest neighbor
- ■ Flip a coin to randomly assign a label to $\mathbf{x}$ (from the labels of its 4 closest points)
- □ Use $k = 3$ instead
- ■ Use $k = 5$ instead
- □ None of the above.

4. (3 points) **Select all that apply:** Please select all that apply about k-NN in the following options.

- ■ A larger k gives a smoother decision boundary.
- ■ To reduce the impact of noise or outliers in our data, we should increase the value k.
- □ If we make k too large, we could end up overfitting the data.
- ■ We can use cross-validation to help us select the value of k.
- □ We should never select the k that minimizes the error on the validation dataset.
- □ None of the above.

5. Consider the following data concerning the relationship between academic performance and salary after graduation. High school GPA and university GPA are two numerical predictors and salary is the numerical target. Note that salary is measured in thousands of dollars per year.

Student ID	High School GPA	University GPA	Salary
1	2.7	3.4	45
2	3.1	3.4	92
3	4.0	4.0	132
4	3.7	2.0	164
5	3.8	3.1	2300
6	3.4	2.6	68
7	3.7	3.2	unknown

(a) (2 points) Among Students 1 to 6, who is the nearest neighbor to Student 7, using Euclidean distance? Answer the Student ID only.

Your answer:

5

(b) (2 points) Now, our task is to predict the salary Student 7 earns after graduation. We apply k-NN to this regression problem: the prediction for the numerical target (salary in this example) is equal to the average of salaries for the top k nearest neighbors. If $k = 3$, what is our prediction for Student 7's salary? Be sure to use the same unit of measure (thousands of dollars per year) as the table above.
Round your answer to the nearest integer.

Your answer:

820

(c) (2 points) **Select all that apply:** Suppose that the first 6 students shown above are only a subset of your full training data set, which consists of 10,000 students. We apply k-NN regression using Euclidean distance to this problem and we define training loss on this full data set to be the mean squared error (MSE) of salary. Now consider the possible consequences of modifying the data in various ways. Which of the following changes **could** have an effect on training loss on the full data set as measured by mean squared error (MSE) of salary?

- ■ Rescaling only "High School GPA" to be a percentage of 4.0
- ■ Rescaling only "University GPA" to be a percentage of 4.0
- □ Rescaling both "High School GPA" and "University GPA", so that each is a percentage of 4.0
- □ None of the above.

4 Perceptron (19 points)

1. (1 point) **Select one:** Consider running the online perceptron algorithm on some sequence of examples S (an example is a data point and its label). Let S' be the same set of examples as S, but presented in a different order.

   True or False: The online perceptron algorithm is guaranteed to make the same number of mistakes on S as it does on S'.

   ○ True

   ● False

2. (3 points) **Select all that apply:** Suppose we have a perceptron whose inputs are 2-dimensional vectors and each feature vector component is either 0 or 1, i.e., $x_i \in \{0, 1\}$. The prediction function is $y = \text{sign}(w_1 x_1 + w_2 x_2 + b)$, and

$$\text{sign}(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

   Which of the following functions can be implemented with the above perceptron? That is, for which of the following functions does there exist a set of parameters w, b that correctly define the function.

   ■ AND function, i.e., the function that evaluates to 1 if and only if all inputs are 1, and 0 otherwise.

   ■ OR function, i.e., the function that evaluates to 1 if and only if at least one of the inputs are 1, and 0 otherwise.

   □ XOR function, i.e., the function that evaluates to 1 if and only if the inputs are not all the same. For example

$$\text{XOR}(1, 0) = 1, \text{ but } \text{XOR}(1, 1) = 0.$$

   □ None of the above.

3. (2 points) **Select one:** Suppose we have a dataset $\{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^M$, $y^{(i)} \in \{+1, -1\}$. We would like to apply the perceptron algorithm on this dataset. Assume there is no intercept term. How many parameter values is the perceptron algorithm learning?

   ○ N

   ○ $N \times M$

   ● M

4. (2 points) **Select one:** The following table shows a dataset and the number of times each point is misclassified during a run of the perceptron algorithm. What is the separating plane w found by the algorithm, i.e. $w = [b, w_1, w_2, w_3]$? Assume that the initial weights are all zero.

x_1	x_2	x_3	y	Times Misclassified
2	1	5	1	10
5	3	3	1	5
1	6	2	1	8
7	2	1	-1	2
3	2	6	-1	3

○ $[18, 25, 14, 34]$

● $[18, 30, 63, 61]$

○ $[16, 56, 18, 47]$

○ $[18, 52, 19, 47]$

5. (2 points) **Select all that apply:** Please select the correct statement(s) about the mistake bound of the perceptron algorithm.

□ If the minimum distance from any data point to the separating hyperplane is increased, without any other change to the data points, the mistake bound will also increase.

■ If the whole dataset is shifted away from origin, then the mistake bound will also increase.

□ If the pair-wise distance between data points is increased, i.e. the data is scaled by some constant value, then the mistake bound will also increase.

□ The mistake bound is linearly inverse-proportional to the minimum distance of any data point to the separating hyperplane of the data.

□ None of the above.

6. (2 points) **Select one:** Suppose we have data whose elements are of the form $[x_1, x_2]$, where $x_1 - x_2 = 0$. We do not know the label for each element. Suppose the perceptron algorithm starts with $\boldsymbol{\theta} = [5, 2]$; which of the following values will $\boldsymbol{\theta}$ never take on in the process of running the perceptron algorithm on the data?

○ $[2, -1]$

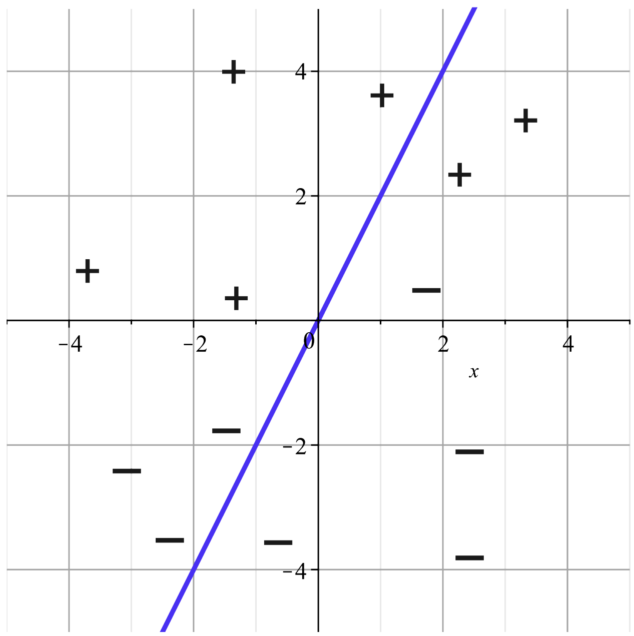
○ $[8, 5]$

○ $[0, -3]$

● $[-4, -6]$

7. (2 points) **Select all that apply:** Consider the linear decision boundary below and the dataset shown. Which of the following are valid weights $\boldsymbol{\theta}$ and their corresponding error on this dataset? (Note: Assume the decision boundary is fixed and does not change while evaluating error.)

- ■ $\boldsymbol{\theta} = [-2, 1]$, error = 5/13
- ■ $\boldsymbol{\theta} = [2, -1]$, error = 8/13
- □ $\boldsymbol{\theta} = [2, -1]$, error = 5/13
- □ $\boldsymbol{\theta} = [-2, 1]$, error = 8/13
- □ None of the above.



8. The following problem will walk you through an application of the Perceptron Mistake Bound. The following table shows a linearly separable dataset, and your task will be to determine the mistake bound for the dataset.

NOTE: The proof of the perceptron mistake bound, as presented in this class, requires that the optimal linear separator passes through the origin. Therefore, make sure that you are projecting the data into **3-dimensional space** in this problem.

x_1	x_2	y
-3	3	1
-1	-3	-1
-2	-3	-1
1	2	1
2	-1	1

(a) (2 points) Compute the radius R of the 3-dimensional "ball" centered at the origin that bounds the data points.
Round to 4 decimal places after the decimal point.

Radius:

4.3589

(b) (2 points) Assume that the linear separator with the largest margin is given by

$$\boldsymbol{\theta}^{*T} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = 0, \text{, where } \boldsymbol{\theta}^* = \begin{bmatrix} 8 \\ 4 \\ 5 \end{bmatrix}$$

Now, compute the margin of the dataset.
Round to 4 decimal places after the decimal point.

Margin:

1.0735

(c) (1 point) Based on the above values, what is the theoretical perceptron mistake bound for this dataset, given this linear separator?
Round to 4 decimal places after the decimal point.

Mistake Bound:

16.4873

5 Linear Regression (20 points)

1. Consider the following dataset:

x	1.0	3.0	6.0	7.0	8.0
y	1.0	1.0	0.0	4.0	2.0

Let $\boldsymbol{x}$ be the vector of datapoints and $\boldsymbol{y}$ be the label vector. Here, we are fitting the data using gradient descent, and our objective function is $\frac{1}{N}\sum_{i=1}^{N}(wx_i + b - y_i)^2$ where N is the number of data points, w is the weight, and b is the intercept.

Note: Showing your work is optional, but it is recommended to help us understand where any misconceptions may occur. Only your numerical answer in the left box will be graded.

(a) (2 points) If we initialize the weight as 4.0 and intercept as 1.0, what is the gradient of the loss function with respect to the weight w, calculated over all the data points, in the first step of the gradient descent update?
Round to 4 decimal places after the decimal point.

Gradient:

245.2

Work:

$$\frac{dy}{dx} = \frac{2}{N}\sum_{i=1}^{N}(wx_i + b - y_i)x_i$$

$$= \frac{2}{5}\cdot((4\cdot1+1-1)\cdot1 + (4\cdot3+1-1)\cdot3 + (4\cdot6+1-0)\cdot6 + (4\cdot7+1-4)\cdot7 + (4\cdot8+1-2)\cdot8)$$

$$= \frac{2}{5}\cdot(4 + 36 + 150 + 175 + 248) = \frac{2}{5}(613) = 245.2$$

(b) (2 points) With the weight and intercept initialization given above and the learning rate 0.01, perform one step of gradient descent on the data. Fill in the following blanks with the value of the weight and the value of the intercept after this step.
Round to 4 decimal places after the decimal point.

Weight:

1.548

Work:

$w \mathrel{-}= \nabla\cdot0.01$

$w = 4 - 245.2\cdot0.01$

Intercept:

-1.452

Work:

$b \mathrel{-}= \nabla\cdot0.01$

$b = 1 - 245.2\cdot0.01$

2. Consider a dataset $\mathcal{D}_1 = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$. Assume the linear regression model that minimizes the mean-squared error on $\mathcal{D}_1$ is $y = w_1 x + b_1$.

   (a) (2 points) **Select one:** Now, suppose we have the dataset $\mathcal{D}_2 = \{(x^{(1)} + \alpha,\ y^{(1)} + \beta), \ldots, (x^{(N)} + \alpha,\ y^{(N)} + \beta)\}$ where $\alpha > 0, \beta > 0$ and $w_1\alpha \neq \beta$. Assume the linear regression model that minimizes the mean-squared error on $\mathcal{D}_2$ is $y = w_2 x + b_2$. Select the correct statement about w_1, w_2, b_1, b_2 below. Note that the statement should hold no matter what values α, β take on within the specified constraints.

   ○ $w_1 = w_2, b_1 = b_2$

   ○ $w_1 \neq w_2, b_1 = b_2$

   ● $w_1 = w_2, b_1 \neq b_2$

   ○ $w_1 \neq w_2, b_1 \neq b_2$

   (b) (2 points) We decide to ask a friend to analyze $\mathcal{D}_1$; however, he makes a mistake by duplicating a subset of the rows in $\mathcal{D}_1$. Explain why the linear regression parameters that minimize mean-squared error on the duplicated data may differ from the parameters learned on $\mathcal{D}_1$, i.e. w_1 and b_1.

   Your answer:

   the additional subset of rows would increase the MSE and thus change w_1 and b_1

3. We wish to learn a linear regression model on the dataset $\mathcal{D} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$ where $\boldsymbol{x} \in \mathbb{R}^k$. For this question, define the *log-cosh loss* as

$$\ell(\hat{y}, y) = \log(\cosh(\hat{y} - y))$$

   In particular, for a given point $\boldsymbol{x}^{(i)}$, the log-cosh loss of a model with parameters $\boldsymbol{\theta}$ is

$$J^{(i)}(\boldsymbol{\theta}) = \log\left(\cosh\left(\boldsymbol{\theta}^T \boldsymbol{x}^{(i)} - y^{(i)}\right)\right)$$

   We are interested in minimizing loss over our training data, so we minimize the average log-cosh loss over all points in $\mathcal{D}$.

   (a) (2 points) What is the objective $J(\boldsymbol{\theta})$ in this setting?

   Your answer:

$$J(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \log(\cosh(\Theta^T x^{(i)} - y^{(i)}))$$

(b) (3 points) What is the partial derivative of $J^{(i)}(\boldsymbol{\theta})$ with respect to the j^{th} parameter, θ_j? It may be helpful to know that $\frac{d}{dx}\cosh(x) = \sinh(x)$ and that $\frac{\sinh(x)}{\cosh(x)} = \tanh(x)$.

Your answer:

$\tanh(\theta^T x^{(i)} - y^{(i)}) \cdot x_j$

(c) (2 points) What is the gradient of $J^{(i)}(\boldsymbol{\theta})$ with respect to the entire parameter vector $\boldsymbol{\theta}$?

Your answer:

$\vec{x}^{(i)} \tanh(\theta^T x^{(i)} - y^{(i)})$

(d) (2 points) To find the optimal parameter vector $\boldsymbol{\theta}^*$ that minimizes $J(\boldsymbol{\theta})$, we again decide to use gradient descent. Write pseudocode that performs gradient descent for one iteration. Set learning rate $\alpha = 0.1$ and initialize $\boldsymbol{\theta}$ to be the zero vector. You may use `gradient[i]` as a variable that contains your answer to part (c) in your pseudocode. Limit your answer to 10 lines.

Your Answer

```
initiate sum_gradient as 0
loop i in range of N:
      sum_gradient -= gradient[i]
      exit loop
divide sum_gradient by N
theta = sum_gradient * 0.1   # α = 0.1
return theta
```

4. We would like to fit a linear regression model to the dataset

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(1)}, y^{(1)}\right), \left(\mathbf{x}^{(2)}, y^{(2)}\right), \cdots, \left(\mathbf{x}^{(N)}, y^{(N)}\right) \right\}$$

with $\mathbf{x}^{(i)} \in \mathbb{R}^M$ by minimizing the ordinary least square (OLS) objective function:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - \sum_{j=1}^{M} w_j x_j^{(i)} \right)^2$$

Specifically, we solve for each linear regression coefficient $w_k, 1 \leq k \leq M$, by deriving an expression for w_k from the critical point $\frac{\partial J(\mathbf{w})}{\partial w_k} = 0$.

(a) (2 points) **Select one:** What is the expression for each w_k in terms of the dataset $\mathcal{D}$ and coefficients $w_1, \cdots, w_{k-1}, w_{k+1}, \cdots, w_M$?

- ● $w_k = \frac{\sum_{i=1}^{N} x_k^{(i)} \left(y^{(i)} - \sum_{j=1, j \neq k}^{M} w_j x_j^{(i)}\right)}{\sum_{i=1}^{N} \left(x_k^{(i)} y^{(i)}\right)^2}$
- ○ $w_k = \frac{\sum_{i=1}^{N} x_k^{(i)} \left(y^{(i)} - \sum_{j=1, j \neq k}^{M} w_j x_j^{(i)}\right)}{\sum_{i=1}^{N} \left(y^{(i)}\right)^2}$
- ○ $w_k = \frac{\sum_{i=1}^{N} x_k^{(i)} \left(y^{(i)} - \sum_{j=1, j \neq k}^{M} w_j x_j^{(i)}\right)}{\sum_{i=1}^{N} \left(x_k^{(i)}\right)^2}$
- ○ $w_k = \sum_{i=1}^{N} x_k^{(i)} \left(y^{(i)} - \sum_{j=1}^{M} w_j x_j^{(i)}\right)$

(b) (1 point) **Select one:** How many coefficients are estimated in the process given above? How many equations are used to solve for these coefficients?

- ● M coefficients, M equations
- ○ M coefficients, N equations
- ○ N coefficients, M equations
- ○ N coefficients, N equations

6 Collaboration Questions

After you have completed all other components of this assignment, report your answers to these questions regarding the collaboration policy. Details of the policy can be found here.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details.
2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details.
3. Did you find or come across code that implements any part of this assignment? If so, include full details.

Your Answer

1) no
2) no
3) no