CM146: Introduction to Machine Learning **Fall 2022**

Midterm

Nov. 3^{rd}, 2022

- This is an open book exam – you can use course materials posted at BruinLearn as your reference. Please do not access **any other material** during the exam. Discussion is strictly prohibited.
- Calculator is allowed.
- This exam booklet has a total of 100 points.
- The exam period is from 10:00am–11:50am 11/3.
- No late submissions will be accepted for any reason. Therefore, make sure to submit well before the deadline! If you face a technical issue, contact us in Piazza (or email) immediately.
- You can ask **private** questions in Piazza, but only clarification questions will be answered. Please don't discuss the exam with anyone except the instructor/TA.
- If you think the question is ambiguous, please write down your explanations or comments at the end of the exam (i.e., Question 7). The regrading will be only based on the submitted answers and explanations. However we may not be able to offer partial credit.
- Note that we cannot provide customized rubrics. Also, as we have already broken down some big questions into sub-questions, we will not provide further partial credits for some sub-questions. In some cases, if you make mistake on earlier sub-questions, it may lead to a wrong answer for the following sub-questions. The rubrics are already designed with that in mind.
- For numerical questions, we ask you to round your answer to the nearest 2 decimal places and write in the format of X.XX or -X.XX). For example, if your answer is 1 (or 0.995), please write down 1.00.

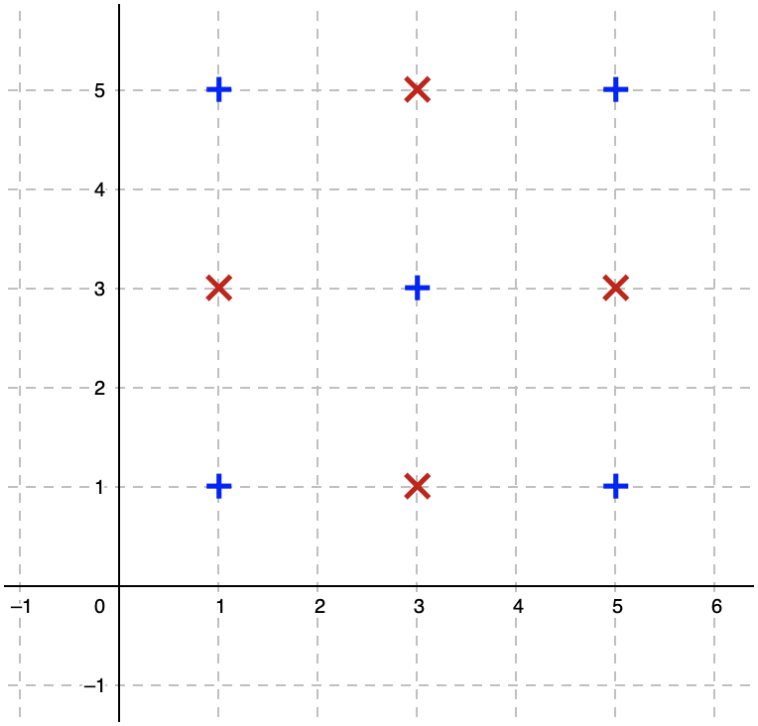
Good Luck!

Name and ID: (2 Point)

Name		/0
KNN		/12
Decision Tree		/22
Margin		/19
Maximum Likelihood Estimation		/22
Perceptron		/12
Multiple Choices and short answer		/13
Total		/100

KNN [12 points]

(a) Consider the following training dataset, what is the leave-one-out validation accuracy (i.e., accuracy computed using 9-fold cross-validation) for the following classifier?



i. [4 points] 1-Nearest Neighbor with Euclidean distance
The accuracy will be 0. For any point, the nearest neighbour is a point of the opposite polarity.

ii. [4 points] 3-Nearest Neighbor with Euclidean distance
The accuracy will be 0. For any point, the two nearest neighbours (out of three) are points of the opposite polarity.

iii. [4 points] 5-Nearest Neighbor with Euclidean distance
The accuracy will be 4/9=0.44. For corner points, 3/5 nearest neighbours are points of the same polarity. For any other point, 3 or more neighbours are points of opposite polarity.

Decision Tree [22 points]

When training deep neural networks, out-of-memory errors often happen, depending on factors such as model size, batch size, and the quality of implementations. We will use the following dataset to learn a decision tree that predicts if an out-of-memory error will happen, based on 3 attributes (batch size, network depth, and the implementation version).

Batch size	Depth	Implementation	Out-of-memory?
small	Deep	A	No
small	Shallow	B	No
small	Medium	B	Yes
large	Shallow	A	No
large	Medium	A	Yes
large	Shallow	B	Yes
large	Deep	B	Yes

In case that more than one attribute has equal information gain, the priority of choosing the attributes is ordered as Batch size > Depth > Implementation.

You may use the following formula.

Entropy:

$$H(p) = -(p \log_2 p + (1-p) \log_2 (1-p))$$

$$H(0) = 0; H(1/2) = 1; H(1/3) = 0.933; H(1/4) = 0.8;$$
$$H(1/5) = 0.7; H(1/7) = 0.571; H(3/7) = 0.971$$

Please round your answer to 2 decimal places.

Use log with base 2 for the cross entropy. Take $\log_2 3 = 1.6$, $\log_2 5 = 2.3$, and $\log_2 7 = 2.8$, if needed.

1. [6 points] What is the entropy of $\mathcal{H}$(out-of-memory)?

$$\mathcal{H}(\text{out-of-memory}) = H(3/7) = 0.971$$

2. [6 points] What is the information gain if we partition the data on the attribute *Implementation*?

$$0.971 - \left(\frac{3}{7}H(1/3) + \frac{4}{7}H(1/4)\right) = 0.971 - \frac{3}{7} \times 0.933 - \frac{4}{7} \times 0.8 = 0.114$$

3. [6 points] Suppose we learn a decision tree by the ID3 algorithm. What is the attribute used for the first split?

$$\text{Gain(batch size)} = \text{Gain(implementation)} = 0.114$$

$$\text{Gain(depth)} = 0.971 - \left(\frac{3}{7}H(1/3) + \frac{2}{7}H(0) + \frac{2}{7}H(1/2)\right) = 0.971 - 0.6856 = 0.2854$$

Therefore, use *depth*.

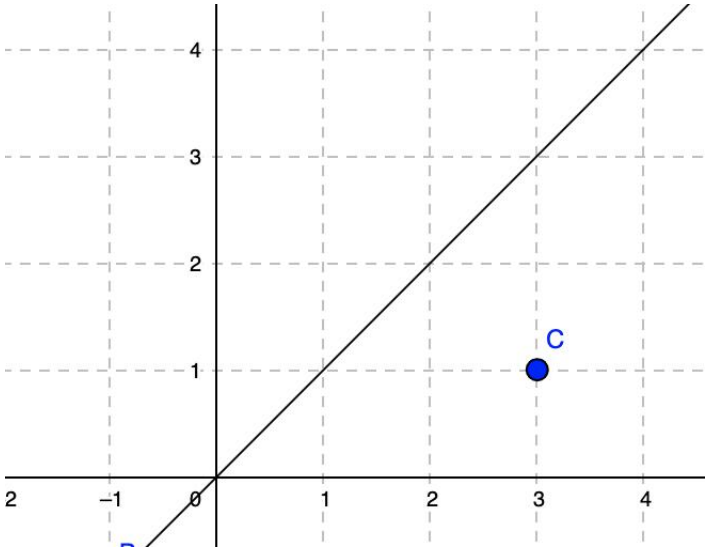
4. [4 points] For the learned decision tree, what is the prediction for an input with: *Batch size*=**small**, *Depth*=**Medium**, *Implementation*=**B**?

Somehow students are seeing two different versions of the problem:

- If depth is *deep*, the prediction is *no*, as batch size is used in the second split and the prediction is the same as the label of the 1st example.
- If depth is *medium*, the prediction is *yes*, as all the examples with *medium* depth are labeled as *yes*.

Margin [19 points]

1. [5 points] Consider the following data point and the hyperplane. What is the



distance between the point C to the line in L2 (euclidean) distance? Round your answer to 2 decimal places.

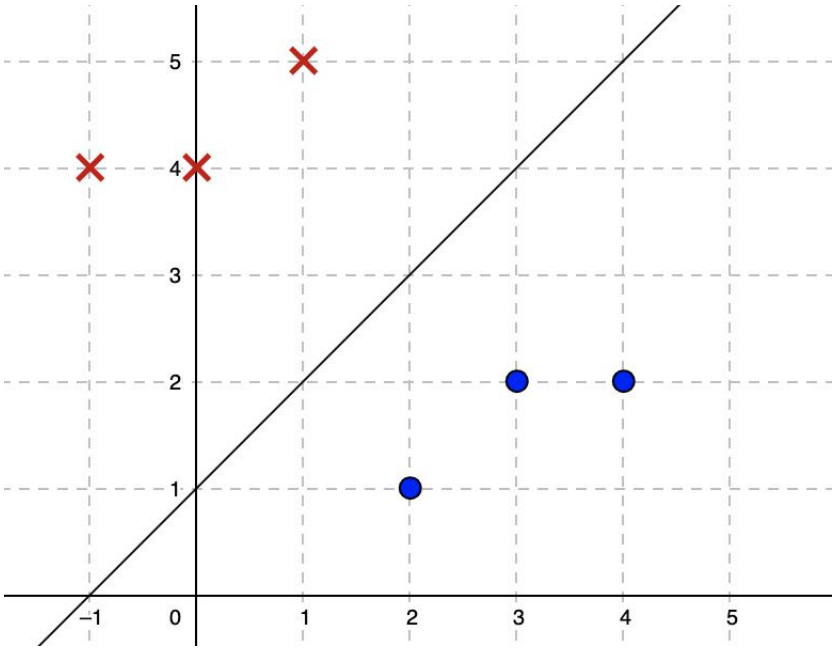
1.41

2. [5 points] Follow the previous question. What is the distance between the point C to the line in L1 (manhattan) distance? Round your answer to 2 decimal places.

2.00

3. [5 points] What is the margin of the hyperplane in L2 (euclidean) distance? Round your answer to 2 decimal places.

1.41



4. [4 points] Follow the previous question. Is there another hyperplane with a larger margin in L2 (euclidean)?

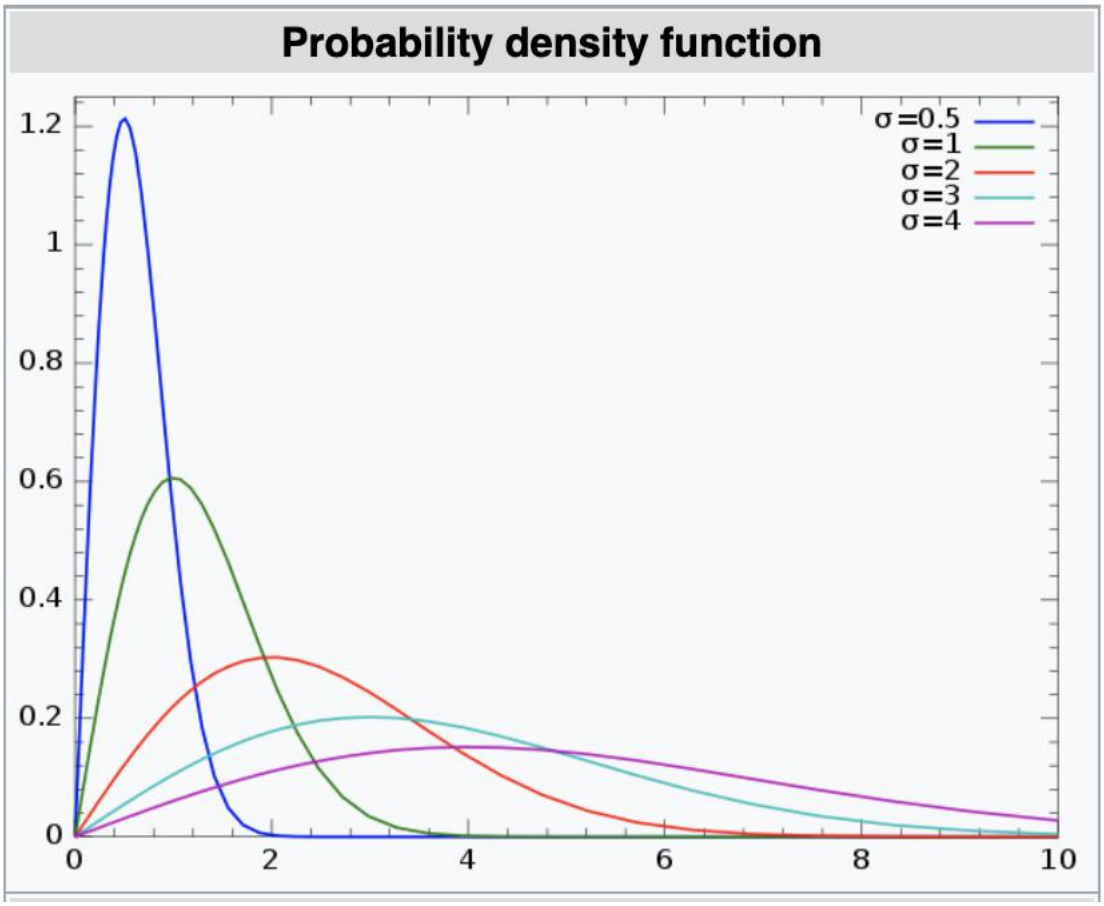
Yes. It's $\frac{5}{4}\sqrt{2}$

Maximum Likelihood Estimation [22 points]

In the following questions, we will explore how to estimate the lifetime of a capacitor. The age of a capacitor depends on various factors and we have encoded them into numerical values: x. Given a bunch of capacitors drawn i.i.d. from the underlying distribution, their age and the factors, we create a training data set $\{(x_i, y_i)\}_{i=1}^N$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. We ask some of our friends from the Statistics department and they suggest that we model the capacitor age using a Rayleigh distribution with parameter $\sigma = w^T x$ as follows:

$$P(Y = y|\sigma) = \frac{y}{\sigma^2} \cdot exp\left(-\frac{y^2}{2\sigma^2}\right) \quad \text{where} \quad exp(y) = e^y$$

That is the probability the lifetime y of a capacitor x follows the Rayleigh distribution with $\sigma = w^T x$, $w \in \mathbb{R}^d$ is the weight vector we can learn.



1. [4 points] We first consider giving a learned w. How we predict the most likely lifetime of a capacitor.

   Rayleigh distribution with parameter σ have the following property:

$$\arg\max_y \frac{y}{\sigma^2} \cdot \exp\left(-\frac{y^2}{2\sigma^2}\right) = \sigma$$

That is the peak of the Rayleigh distribution $P(Y = y|\sigma)$ is when $y = \sigma$.

Based on this property, consider the feature vector extracted for a capacitor is $x = [1, 3, 1]$ and the learned weight vector $w = [1, 2, 1]$. What is the most likely lifetime y of the capacitor?

We estimate $\sigma = w^T x = 8$ and the peak (most likely) occurs when $y = \sigma$. Thus, most likely lifetime $y = 8$.

2. [4 points] Now, let's discuss how to learn w based on the training data set $\{(x_i, y_i)\}_{i=1}^N$. We start with considering one data point (x_i, y_i).

   Based on our assumption, the distribution of the lifetime of a capacitor follow a Rayleigh distribution with parameter $\sigma = w^T x$. What is the probability $P(Y = y_i|x_i, w)$ of that the lifetime of a capacitor x_i is y_i based on the assumption.

   i. $P(y_i|x_i, w) = \frac{y_i}{x_i^2} \cdot exp\left(-\frac{y_i^2}{2x_i^2}\right)$

   ii. $P(y_i|x_i, w) = \frac{y_i}{w^T x_i} \cdot exp\left(-\frac{y_i^2}{2w^T x_i}\right)$

   iii. $P(y_i|x_i, w) = \frac{y_i}{(w^T x_i)^2} \cdot exp\left(-\frac{y_i^2}{2(w^T x_i)^2}\right)$

   iv. $P(y_i|x_i, w) = \log(y_i) - \log(w^T x_i) - \frac{y_i^2}{2w^T x_i}$

   The answer is (C). We mainly substitute $\sigma = w^T x_i$ in the equation for $P(Y = y_i|\sigma)$

3. [4 points] Given the dataset $\{(x_i, y_i)\}_{i=1}^N$, what is the log-likelihood of the model w?

   i. $\mathcal{LL}(w) = constant + \sum_{i=1}^N \left(-log(w^T x_i) - \frac{y_i^2}{2w^T x_i}\right)$

   ii. $\mathcal{LL}(w) = constant + \sum_{i=1}^N \left(-2log(w^T x_i) - \frac{y_i^2}{2(w^T x_i)^2}\right)$

   iii. $\mathcal{LL}(w) = constant + \prod_{i=1}^N \frac{y_i}{(w^T x_i)^2} \cdot exp\left(-\frac{y_i^2}{2(w^T x_i)^2}\right)$

   iv. $\mathcal{LL}(w) = constant + \prod_{i=1}^N \left(-log(w^T x_i) - \frac{y_i^2}{2w^T x_i}\right)$

   The answer is (B). We multiply the likelihood for each datapoint (from previous part) and take the log.

4. [4 points] How we can obtain w?

   - We can maximize the log-likelihood in previous question.
   - We can minimize the log-likelihood in previous question

   The answer is (A). We want to find the best w which maximizes the likelihood of observing these datapoints.

5. [6 points] How we can solve the optimization problem? (selected all correct options)

   - The optimization problem can be solved by SGD.

- The optimization problem can be solved by GD.

The answer is (A), (B). We can use both the methods to solve the current optimization problem.

Perceptron [12 points] Consider the following binary classification dataset with 3 features:

x_1	x_2	x_3	y
1	1	2	1
1	2	0	1
4	0	0	-1
2	0	0	-1
0	0	1	1

Consider training a Perceptron model $y = sgn(w^T x + b)$ with w and b are initialized with 0.

Note that you can augment b into w using the trick we discussed in class.

1. [4 points] After running the Perceptron model over these five data points **once**, what is w and b?

Using $\text{sgn}(0) = -1$: $\mathbf{w} = [-3, 1, 2]$, $b = 0$.

Using $\text{sgn}(0) = 1$: $\mathbf{w} = [-4, 0, 1]$, $b = 0$. Both are considered correct.

2. [4 points] Given a test data point [-1, -3, 1], what is the model prediction y.

The model prediction is $\text{sgn}(\mathbf{w}^\top \mathbf{x} + b) = \text{sgn}(2) = 1$.

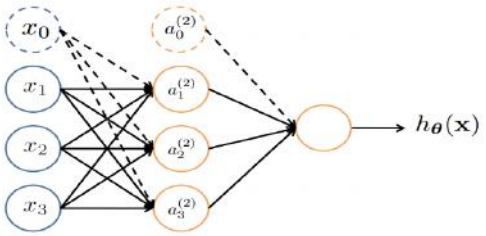
3. [4 points] If we allow the Perceptron model to run over these five data points **until it converges**, what is the final model w and b?

Using $\text{sgn}(0) = -1$: $\mathbf{w} = [-2, 3, 2]$, $b = 1$.

Using $\text{sgn}(0) = 1$: $\mathbf{w} = [-3, 1, 3]$, $b = 1$. Both are considered correct.

Multiple choices and short answer [12 points]

- **Neural Networks** [6 points] Consider the following neural network we discussed in the class. For a binary classification problem, the model make prediction on x based on the sign of $h_\Theta(x)$. That is the model predicts positive if $h_\Theta(x) > 0$; otherwise it predicts negative.



$a_i^{(j)}$ = "activation" of unit i in layer j

$\Theta^{(j)}$ = weight matrix controlling function mapping from layer j to layer $j+1$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

If the activation function $g(z) = \beta z$, where β is a non-zero constant.

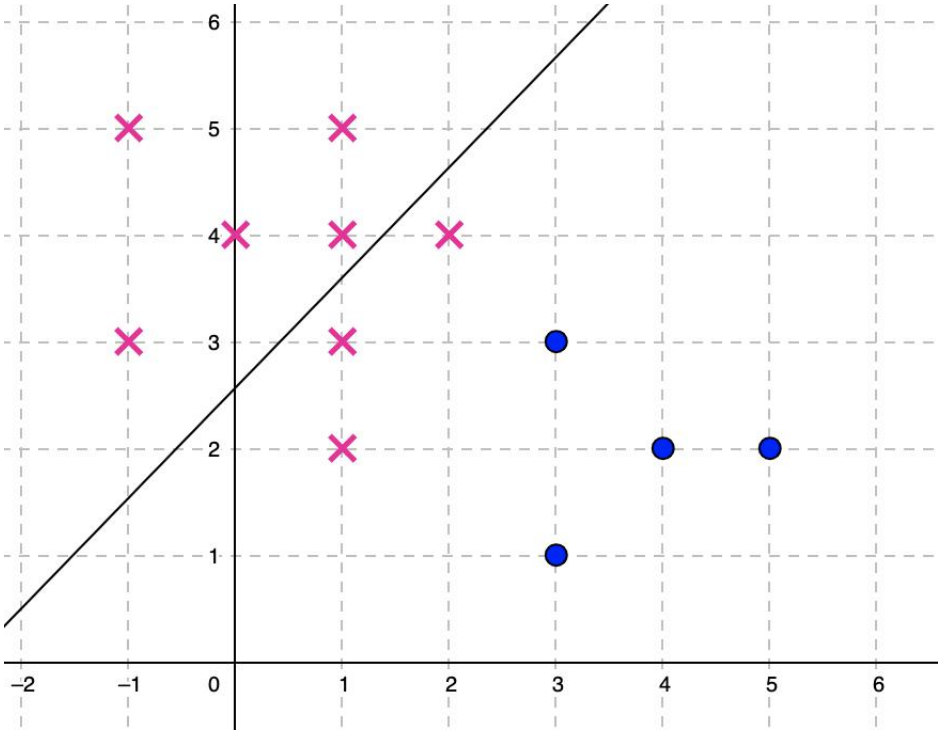
Which of the following is True?

  - If data are linearly separable, we can find a set of parameter Θ such that $sgn(h_\Theta(x))$ separates all positive and negative data.
  - If data are linearly separable, there is no any assignment of parameter Θ such that $sgn(h_\Theta(x))$ separates all positive and negative data.
  - If data are **not** linearly separable, we can find a set of parameter Θ such that $sgn(h_\Theta(x))$ separates all positive and negative data.
  - If data are **not** linearly separable, there is no any assignment of parameter Θ such that $sgn(h_\Theta(x))$ separates all positive and negative data.

Correct Answers: (A), (D)

Since the activation function used in the hidden layer $g(z)$ is a linear function in z, $h_\Theta(x)$ will behave like a linear function and only be able to separate data which is linearly separable.

- [3 points] Given a set of positive points (blue circle) and negative points (red cross) as training data. We train a logistic regression model using SGD with 100 iterations. The results are shown in the following figure



Which of the following statement is true?

  - The model is overfitting.
  - The model is underfitting.
  - The model has 0 training error

Correct Answer: (B)
We can clearly see that the model makes mistakes (for some negative points) and classify all points correctly despite the points being linearly separable.

- [4 points] Provide one possible reason why the model does not do well in one sentence.

Correct Answers: Low number of training epochs, low/high learning rate, randomness in selection of datapoints due to SGD's algorithm.