6.390: Midterm Exam, Spring 2024

Solutions

- This is a closed book exam. One page (8 1/2 in. by 11 in.) of notes, front and back, are permitted. Calculators are not permitted.
- The total exam time is 2 hours.
- The problems are not necessarily in any order of difficulty.
- Record all your answers in the places provided. If you run out of room for an answer, continue on a blank page and mark it clearly.
- If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.
- If you absolutely *have* to ask a question, come to the front.
- **Write your name on every piece of paper.**

Name: _____ MIT Email: _____

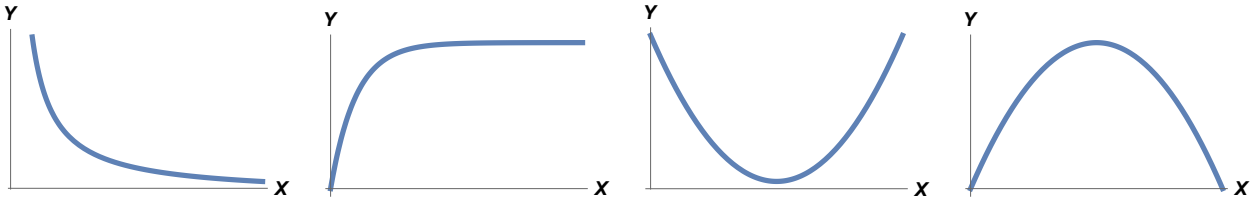
Question	Points	Score
1	12	
2	19	
3	22	
4	20	
5	9	
6	18	
Total:	100	

Name: _____

Trendy Pics

1. (12 points) Consider just the general shape of the following plots.

   For each of the following possible interpretations of the quantities being plotted on the X and Y axes, indicate which of the plots would *most typically* be the result, or indicate "none" if none are appropriate. Provide a one-sentence justification for each answer.



(A) (B) (C) (D)

   Assume all quantities other than X are held constant during the experiment. Error quantities reported are averages over the data set they are being reported on.

   Any given plot may appear more than once in the answers.

   (a) X axis: Number of training examples; Y axis: Test error.

   Solution: A. With more training data, we are able to find a better predictor.

   (b) X axis: Number of training examples; Y axis: Training error.

   Solution: B. It's easy to fit a small amount of data exactly; harder as we get more data.

   (c) X axis: Order of polynomial features; Y axis: Test error.

   Solution: C. If we do not have enough features, we may underfit; If we have too many features, we may overfit.

   (d) X axis: Order of polynomial features; Y axis: Cross-validation error.

   Solution: C. Same as test set error

Name: _____

Regression or Repetition?

2. (19 points) Suppose that we are given a small dataset and we would like to learn the parameters of a linear regressor hypothesis taking the form $h(x) = \theta^\top x + \theta_0$ for fitting the data.

   (a) Consider the following dataset $\mathcal{D}_1$ containing three data points (in feature-label pairs):

x	y
-4	15
2	-3
-1	0

   Suppose that we would like to minimize:

$$J_1(\theta, \theta_0; \mathcal{D}_1) = \frac{1}{n} \sum_{i=1}^{n} (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2.$$

   i. For J_1, we know that there exist θ^* and θ_0^* that minimizes it. Can we find θ^* via the analytical solution formula? (No need for justification.)

> **Solution:**
> Yes.

   ii. Suppose we know that for J_1, one set of minimizing parameters has $\theta_0^* = 1$. What is the corresponding unknown θ^*?

> **Solution:**
>
> $$\begin{aligned} J_1 &= (-4\theta + 1 - 15)^2 + (2\theta + 1 + 3)^2 + (-\theta + 1)^2 \\ &= (-4\theta - 14)^2 + (2\theta + 4)^2 + (\theta - 1)^2 \\ &= (16\theta^2 + 4 \times 14 \times 2\theta + (14)^2) + \left(4\theta^2 + 16\theta + 16\right) + (\theta^2 - 2\theta + 1) \\ &= (16 + 4 + 1)\theta^2 + (4 \times 14 \times 2 + 16 - 2)\theta + \text{ some constant} \\ &= 21\theta^2 + 126\theta + \text{ some constant} \end{aligned}$$
>
> So
>
> $$\theta^* = -\frac{126}{21 \times 2} = -3$$

   iii. What is J_1^*, the minimum value achievable of J_1?

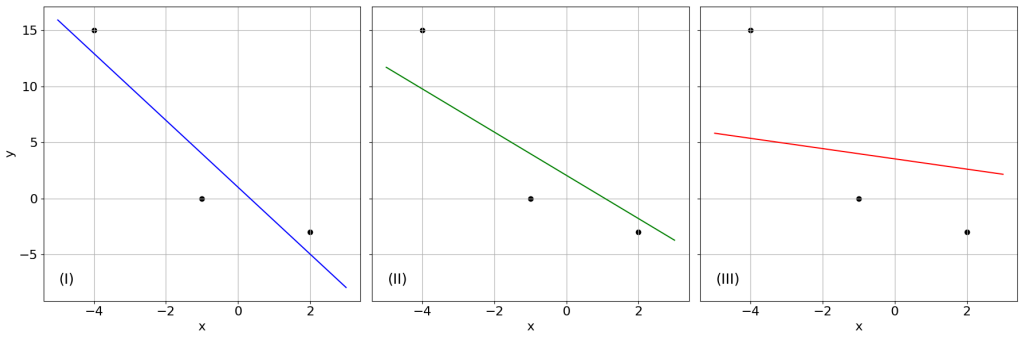
> **Solution:**
>
> $$J_1^* = \frac{(12 + 1 - 15)^2 + (-6 + 1 + 3)^2 + (3 + 1)^2}{3} = \frac{4 + 4 + 16}{3} = 8$$

(b) Suppose instead of J_1, we try to minimize:

$$J_2(\theta, \theta_0; \mathcal{D}_1, \lambda) = \frac{1}{n} \sum_{i=1}^{n} (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda \|\theta\|^2.$$

with $\lambda = 0.1$, 10, and 100, respectively. Identify the λ used to generate plot (III).

Solution: For plot (III), we used $\lambda = 100$.



(c) Suppose we add a second feature for each of the three datapoints in $\mathcal{D}_1$. In other words, consider the new dataset $\mathcal{D}_2$:

x_1	x_2	y
-4	-8	15
2	4	-3
-1	-2	0

Suppose that we would like to minimize:

$$J_3(\theta, \theta_0; \mathcal{D}_2) = \frac{1}{n} \sum_{i=1}^{n} (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2.$$

i. For J_3, we also know that there exist θ^* and θ_0^* that minimizes it. Can we find θ^* via the analytical solution formula? If yes, provide such θ^*, if no, briefly justify why not.

Solution:
No. The two features are linearly dependent. Hence the $\tilde{X}^T \tilde{X}$ will not be invertible.

ii. Compare J_3^*, the minimum value achievable of J_3, with J_1^*.
Which option below is true? Briefly justify your choice.

Solution:
$J_3^* = J_1^*$. Note that in $\mathcal{D}_2$, the 2nd feature is linearly dependent on the 1st feature. Intuitively, this means the 2nd feature gives us nothing additional to learn from towards a linear hypothesis, i.e. the 2nd feature is a redundant feature.
Algebraically, we can write out explicitly J_3 and J_1 to see their connection. Suppose, for the sake of notational cleanness, that we let the set of parameters for J_1 be α and β:

$$J_1 = \left[(-4\alpha + \beta - 15)^2 + (2\alpha + \beta + 3)^2 + (-\alpha + \beta)^2\right]/3$$

and let the parameters for J_3 be θ and θ_0:

$$J_3 = \left[(-4\theta_1 + 8\theta_2 + \theta_0 - 15)^2 + (2\theta_1 + 4\theta_2 + \theta_0 + 3)^2 + (-\theta_1 - 2\theta_2 + \theta_0)^2\right]/3$$

Then, we realize that by letting $\theta_1 + 2\theta_2 = \alpha$ and $\theta_0 = \beta$, any value achievable by J_1 is achievable by J_3, and vise-versa.
Hence $J_1^* = J_3^*$.

Name: _____

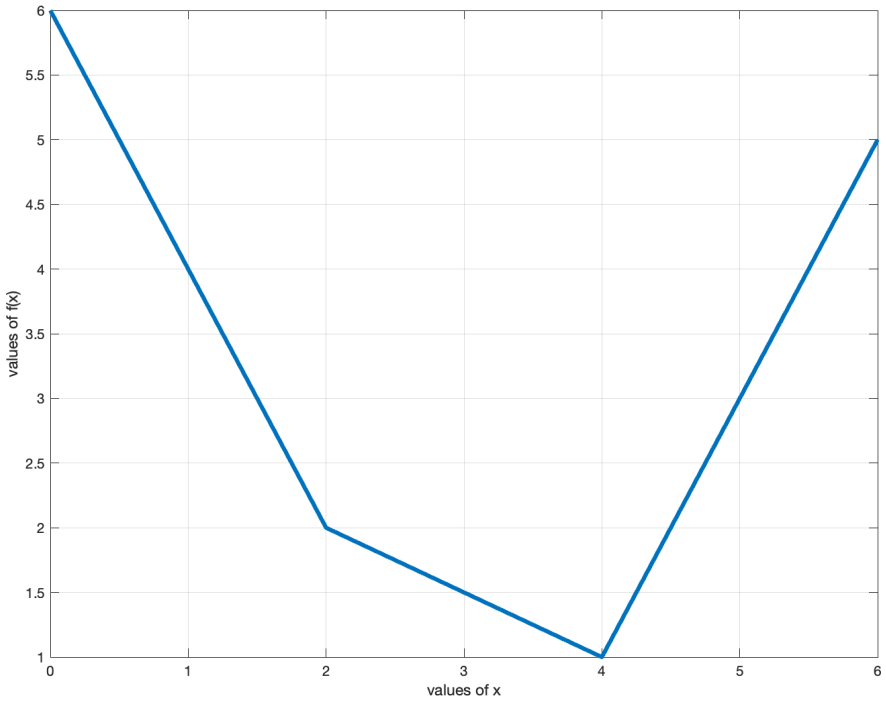
Gradient Descent in Pictures

3. (22 points) John is using standard gradient descent iterations

$$x^{(k+1)} = x^{(k)} - \eta \nabla f(x^{(k)}) \qquad (k = 0, 1, 2, \dots)$$

on a variety of functions f.

(a) First, John applies gradient descent to a piecewise-linear function $f : \mathbb{R} \to \mathbb{R}$ with the (partial) graph shown on the figure below:



At points $x = 2$ and $x = 4$, John uses $\nabla f(2) = -0.5$ and $\nabla f(4) = 0$.

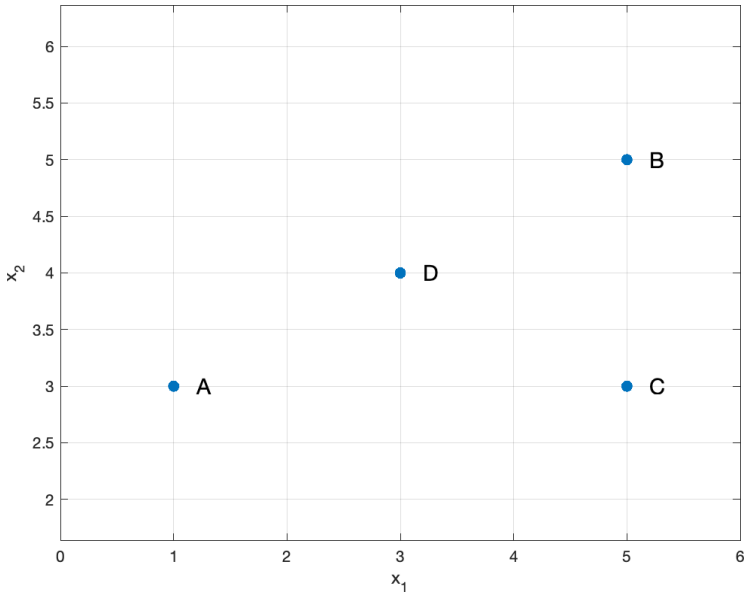
i. Starting from the initial guess $x^{(0)} = 5$, and using step size $\eta = 1$, what will be the values of $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$?

Solution: Answer: $x^{(1)} = 3$, $x^{(2)} = 3.5$, $x^{(3)} = 4$. **Reasoning:** by inspection of the graph, $\nabla f(x^{(0)}) = 2$, hence $x^{(1)} = 5 - 1 \cdot 2 = 3$. Next, $\nabla f(x^{(1)}) = -0.5$, hence $x^{(2)} = 3 - 1 \cdot (-0.5) = 3.5$. Finally, $\nabla f(x^{(2)}) = -0.5$, hence $x^{(3)} = 3.5 - 1 \cdot (-0.5) = 4$.

ii. John discovers that, starting with $x^{(0)} = 1$, there are many values of $\eta > 0$ for which the gradient descent iterations produce oscillations of period 2 within the range $(0, 6)$ (i.e., $x^{(k+2)} = x^{(k)} \in (0, 6)$ for all $k = 0, 1, 2, \dots$). Find all such values of η.

Solution: Answer: $\eta \in (1.5, 2.5)$. **Reasoning:** equality $x^{(k+2)} = x^{(k)}$ requires $\nabla f(x^{(k)}) = -\nabla f(x^{(k+1)})$. Since $\nabla f(x^{(0)}) = -2$, we need $\nabla f(x^{(1)}) = 2$, which means $x^{(1)} = x^{(0)} - \eta \nabla f(x^{(0)}) = 1 + 2\eta \in (4, 6)$. Hence $\eta \in (1.5, 2.5)$.

(b) After mastering one-dimensional optimization, John applies gradient descent to a smooth function $f : \mathbb{R}^2 \to \mathbb{R}$, with $\eta = 0.1$, resulting in the sequence of points $x^{(0)} = A$, $x^{(1)} = B$, $x^{(2)} = C$, $x^{(3)} = x^{(4)} = D$ shown on the plot below:



i. Find $\nabla f(x^{(0)})$, $\nabla f(x^{(1)})$, $\nabla f(x^{(2)})$, and $\nabla f(x^{(3)})$.

Solution: $\nabla f(x^{(k)}) = \eta^{-1}(x^{(k)} - x^{(k+1)})$, hence

$$\nabla f(x^{(0)}) = \begin{bmatrix} -40 \\ -20 \end{bmatrix}, \nabla f(x^{(1)}) = \begin{bmatrix} 0 \\ 20 \end{bmatrix}, \nabla f(x^{(2)}) = \begin{bmatrix} 20 \\ -10 \end{bmatrix}, \nabla f(x^{(3)}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

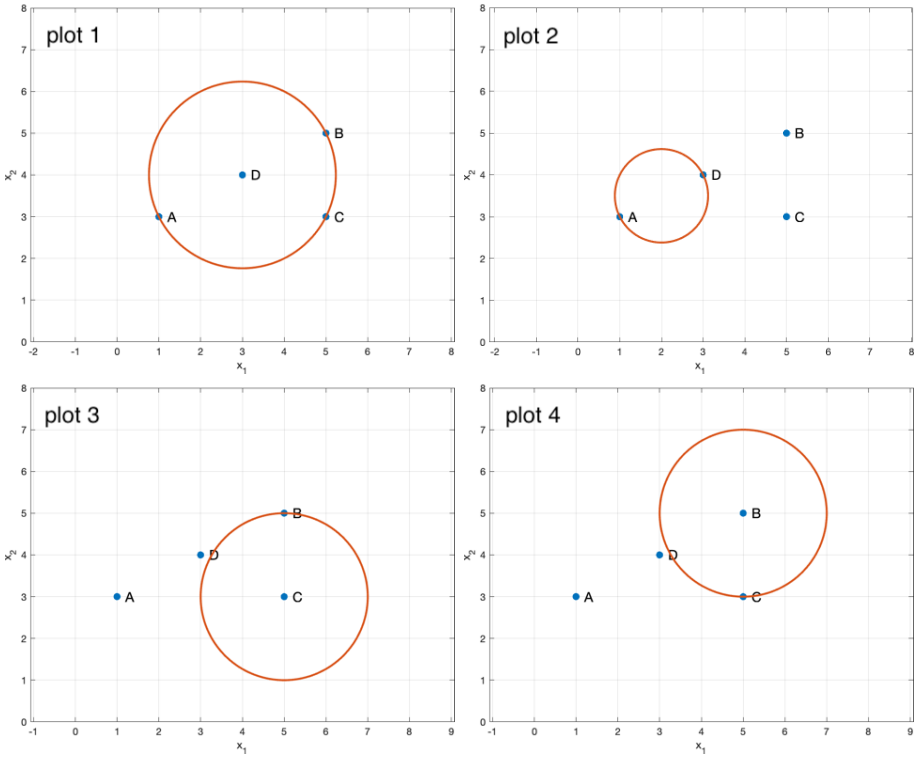
ii. Is this statement true or false: "given the information provided, the point D **must** be a global minimum of function f"? Briefly justify your choice.

> **Solution:**
> False. The only thing we know is that $\nabla f(D) = 0$. With the info provided, the point D could be the global *maximum* of f!

iii. A level set of a function h is said to be the set of points x for which $h(x) = c$ for some constant scalar c. (For example, for function $h(x) = x_1^2 + 2x_2^2$, its level sets at various c levels are concentric ellipses.)

The plots 1-4 below show some circles. Given the sequence we get from gradient descent on f, which of these circles can *possibly* be a level set of function f? Choose all that apply; and provide a short justification.

Hint: for any point p, think about the relation between the direction of $\nabla f(p)$ and the level set passing through p.



> **Solution:**
> plots 2 and 3.
>
> Since the gradient vector's components are the partial derivatives of the function with respect to each of its variables, by definition, gradient vector indicates how fast the function changes in each coordinate direction. In other words, the gradient vector must point towards the direction of maximum increase of the function's

value.

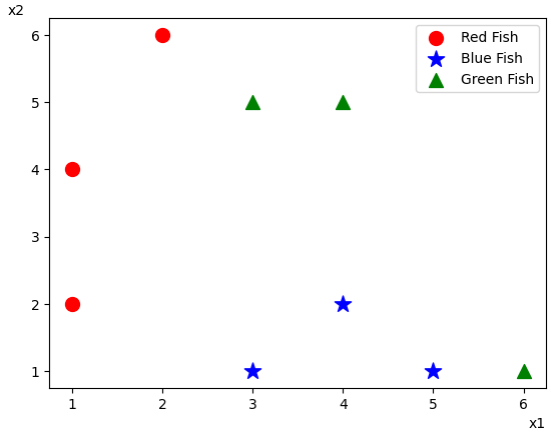
Now consider level sets. By definition, there is no change in the function's value for all points on a level set.

Combined with the fact above about gradient vectors, we must have that gradient vectors must be perpendicular to tangent line of a levelset.

Name: _____

Classi-fish-cation

4. (20 points) Allen, Bonnie, and Clive are restoring a polluted river in their town and the fish are starting to come back! There are three different species of fish: Red, Blue, and Green. They have setup a sensor in the river which collects 2D data (that is, two features for each data point) from the fish that swim by. After a day, they get the dataset plotted below.



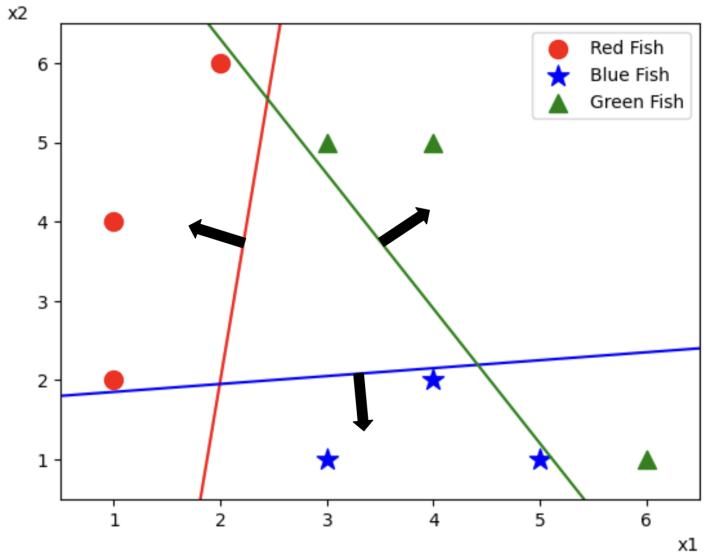
The friends would like to use this dataset to to learn a fish classifier. However, each friend wants to try a different multi-class classification approach.

(a) Allen thinks One-vs-All (OVA) classification is the best approach.

Recall that, in OVA, we train one binary classifier for every class. For instance, towards training for the "Red" OVA classifier, we use the three data points at $[1, 2], [1, 4]$, and $[2, 6]$ as positives ("red"), and all other six data points as negatives ("not red"). Similarly, we can train for the "Green" classifier and "Blue" classifier. And at prediction time, we return the label that the classifiers are most confident about.

Allen runs his classifier and gets the output below, with the classifiers unlabeled.

i. On the plot below, use arrows to clearly draw the normal direction for each of the classifiers (Red, Green, and Blue).



Solution:

ii. What is the classification accuracy of Allen's approach on the training dataset? (You may leave your answer in fractional form, if applicable.) Briefly show your reasoning.

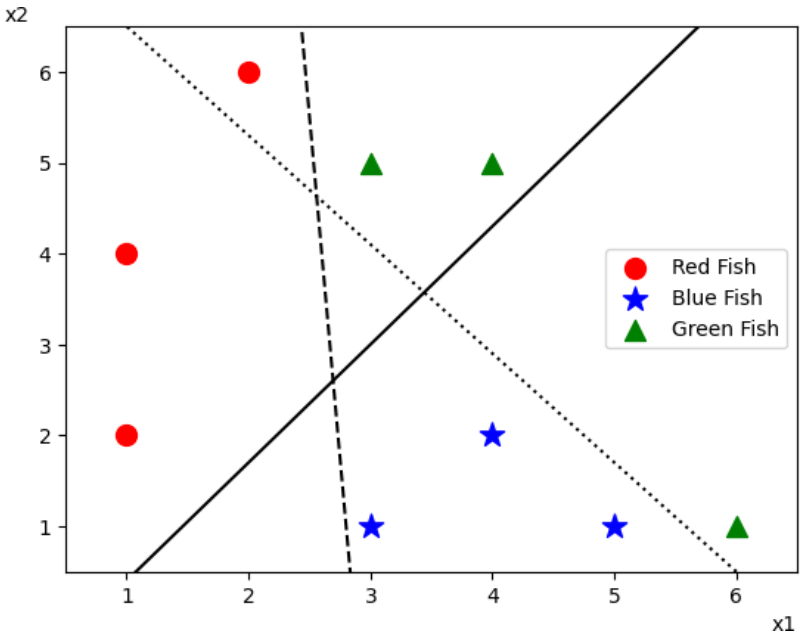
Solution: 8/9. All data points are correctly classified except for the green point at (6, 1). This data point is on the positive side of both the blue classifier and the green classifier, but is farther away from the blue classifier than the green. This tells us that our confidence that this point is blue (or otherwise, its output on the sigmoid of this classifier) is higher than it is for green, which is incorrect. All other data points are only on the positive side of their correct classifier, and therefore we must have the highest confidence that they are each the appropriate class.

(b) Bonnie thinks One-vs-One (OVO) classification is superior.

Recall that in OVO, we train one binary classifier for every pair of classes. For instance, towards training for the "Red-Green" OVO classifier, we use the three data points at $[1, 2], [1, 4]$, and $[2, 6]$ as positives ("red"), and the three data points at $[3, 5], [4, 5]$, and $[6, 1]$ as negatives ("green", or the "not-red"). Similarly, we can train for the "Blue-Red" classifier (with "Blue" being the positive class and "Red" being the negative), and "Blue-Green" classifier (with "Blue" being the positive class and "Green" being the negative). At prediction time, each classifier returns a single binary prediction, and the class with the most votes overall will be our final prediction.

Bonnie runs OVO and gets the Blue-Red, Blue-Green, Red-Green classifiers plotted below.

Name: _____



i. Match the lines with "Blue-Red", "Blue-Green", "Red-Green" classifiers.

Solution:
..... (dotted line): Blue-Green

– – – (dashed line): Red-Green

——— (solid line): Blue-Red

ii. What is the classification accuracy of Bonnie's approach on the training dataset? (You may leave your answer in fractional form, if applicable.) Briefly show your reasoning.

Solution: 1. Here, each data point correctly gets at least two votes for the appropriate class.

iii. Consider the Blue-Red classifier. Which of the options below would be a plausible θ and θ_0 values that could lead to this classifier? (Recall that for this Blue-Red classifier, the "Blue" class is the **positive** class.)

Choose all that apply and provide a one-sentence justification.

Solution:
$\theta = [1.3, -1]$, $\theta_0 = -0.9$. The clearest way to see this is based on the normal, which should be pointing toward the class Blue in the classifier Blue-Red (note that this is the convention we have defined here for this style of labeling, which is not necessarily the same as what is in the homework). This normal is positive in x_1 and negative in x_2. The option $\theta = [1.3, -1]$, $\theta_0 = -0.9$ is the only one which satisfies this.

iv. Is it possible that the Blue-Red classifier results from a set of θ and θ_0 values that is different than the options given in the previous part?

If yes, provide one such set of θ and θ_0 values. If not, provide a one-sentence justification.

Solution: Yes. Any positive scaling of the correct option given in the previous part.
E.g., $\theta = 2 * [1.3, -1]$, $\theta_0 = 2 * (-0.9)$

(c) Finally, Clive argues that a direct multi-class classification (using softmax and NLLM loss) is better than the binary classifiers from both OVA and OVO. Assume Clive encodes the true labels as 3×1 one-hot vectors corresponding to each class in the order

$$y = \begin{bmatrix} \text{Red} \\ \text{Blue} \\ \text{Green} \end{bmatrix}$$

i. Is the following a possible softmax output from Clive's classifier? Why or why not?

$$g = \begin{bmatrix} 0.30 \\ 0.25 \\ 0.41 \end{bmatrix}$$

Solution: No. Entries do not sum up to 1.

ii. Let's look at a single data point $[2, 4]$ whose true label is Red. Suppose for this data point, Clive gets the following softmax output,

$$g = \begin{bmatrix} 0.46 \\ 0.23 \\ 0.31 \end{bmatrix}$$

What is the loss $\mathcal{L}_{\text{NLLM}}$ incurred by this data point?

Recall that $\mathcal{L}_{\text{NLLM}}(g, y) = -\sum_{k=1}^{K} y_k \log(g_k)$. Also, for reference, $\log(0.46) = -0.78$, $\log(0.23) = -1.47$, $\log(0.31) = -1.17$.

Solution:

$$y = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

Depending on if the true label is Red: $\mathcal{L}_{\text{NLLM}}(g, y) = -((1 * -0.78) + (0 * -1.47) + (0 * -1.17)) = 0.78$
Or if the prompt says label is Blue: $\mathcal{L}_{\text{NLLM}}(g, y) = -((0 * -0.78) + (1 * -1.47) + (0 * -1.17)) = 1.47$

Name: _____

Banana Madness

5. (9 points) From 2016-2019, the "Banana Man" (that is, an undergraduate student in a banana costume) could be seen running through some of the popular lectures on campus. There was no warning as to when or where the Banana Man would show up (for example, you might be mid-derivation in 18.01 and then, BAM!, a large banana is sprinting behind your professor with a shout–"That's bananas!").

   New student Tyler is a big fan, and is interested in seeing if he can figure out how the Banana Man chose when and where to appear in class.

   (a) First, Tyler interviews former students about what they can remember about the Banana Man encounters, in order to collect some data.

   For each of the following, suggest an appropriate feature encoding, and provide the dimension of the feature encoding.

   i. Each interviewee remembers the class they were in when they had an encounter. It was always one of {3.091, 5.111, 18.01, 18.02, or 6.100A}.

   Solution: This is a categorical variable, and there is no relationship between each class. We can use a one-hot vector of size 5, in which only a single class is marked at a time.

   ii. Each interviewee also roughly remembers when an encounter happened: either 1) in the beginning of class, 2) the middle of class, or 3) the end of class.

   Additionally, it was well-known on campus that the Banana Man favored the following times in descending order: middle of class (preferred most!), beginning of class (less preferred), and end of class (least preferred). However, the exact magnitude of the preference between them was unknown.

   Solution: Since we know a-priori that the Banana Man had established preferences on the section of class he would appear during, we might try a thermometer encoding with a vector of size 3 with the end of class being the lowest (or [1, 0, 0]), beginning of class being in the middle (or [1, 1, 0]) and the middle of class being the highest (or [1, 1, 1]). Therefore, we have built-in a relationship between these sections of class, even though we don't know the exact numerical relationship between them. (Technically, we could also argue that this is categorical and simply use a one-hot vector, in which that relationship may still be learned. But the prior knowledge of preference ordering would be lost).

iii. Tyler can also find out the number of students who were present in lecture for each encounter. This might range from zero students showing up (during busy weeks), to a maximum of 566 students attending (full capacity in the largest lecture hall, 26-100).

> **Solution:** This is a continuous numerical feature with a known range. We ideally want our values to be in an appropriate range, [0, 1]. Both standardization and normalization could be acceptable answers here.

(b) Using all of this data, Tyler now wants to make predictions about how likely it might've been to see the Banana Man.

He wants to design a simple neural network for the task. What should the dimension of Tyler's *output* prediction be (that is, the shape of this output)? What would be a good choice of activation function on the output layer? Briefly justify your answer.

> **Solution:** Tyler wants a probability, so this should be a scalar number output in the range 0 to 1. "Scalar" or size 1 are acceptable. Tyler wants a single probability, so he should use a sigmoid activation. Note that while there are many cases where softmax is a helpful function for probabilities, and is equivalent to sigmoid when $n = 2$, softmax cannot be used here as the probability on a single output would always be 1.

Name: _____

Neural Not/And/Ors?

6. (18 points) Neural networks have the capability to express a wide variety of functions – an attribute that has been critical in their success. In this question, we will show that neural networks have the ability to represent *any* logical (Boolean) formula, even when limited to having only one hidden layer!

   Let us assume the following setup:

   - **Inputs**: $x^{(i)} \in \{0,1\}^m$ are binary vectors of length m. (That is, a data point $x^{(i)}$ is m-dimensional, and each of the m entries is either a 0 or a 1.)
   - **Activation function**: The step function, i.e., $f(z) = 1$ if $z > 0$ and 0 otherwise. (We would not actually want to use this activation function when training a neural network due to issues with zero gradients, but are using it here for simplicity.)
   - **Outputs**: $y^{(i)} \in \{0,1\}$, where 0 indicates FALSE and 1 indicates TRUE.

   (a) Suppose $m = 2$, i.e., the input is a pair of binary values. Suppose we have a neural network with *no hidden units* and just a single output unit, i.e., $y = f(w^\top x + w_0)$ is the entire neural network.

   i. Convince yourself of this fact: we can represent Boolean OR by letting $w = \begin{bmatrix} 2 & 2 \end{bmatrix}^\top$ and $w_0 = -1$.

   Now, what is a set of values for w and w_0 that would allow us to represent Boolean AND? (See the logic tables below for reference.)

OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Solution: One option is $w = \begin{bmatrix} 2 & 2 \end{bmatrix}^\top$ and $w_0 = -3$.

ii. What is a set of values for w and w_0 for a unit that would allow us to represent the Boolean NAND operation – which is ((NOT x_1) OR (NOT x_2))?

Hint: $(1 - x_i)$ is equivalent to (NOT x_i). Can you plug this in to the "OR unit" from the previous part, and rearrange to get the desired expression?

Solution: One approach is to let $w_i = 2$ if x_i is not negated, and $w_i = -2$ if x_i is negated. We then let w_0 equal -1 plus two times the number of variables that are negated (so, $w_0 = -1$ if neither are negated, 1 if one is negated, and 3 if both are negated).

As an example, let us see how this works for computing the Boolean NAND operation, which can be expressed as ((NOT x_1) OR (NOT x_2)). Let $w^{\text{OR}} = \begin{bmatrix} 2 & 2 \end{bmatrix}^\top$ and $w_0^{\text{OR}} = -1$ be the weights and offset of our "OR unit." We can compute NAND as follows:

$$(w^{\text{OR}})^\top \begin{bmatrix} 1 - x_1 \\ 1 - x_2 \end{bmatrix} + w_0^{\text{OR}} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}^\top \begin{bmatrix} 1 - x_1 \\ 1 - x_2 \end{bmatrix} - 1 = (2 - 2x_1) + (2 - 2x_2) - 1 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}^\top x + 3.$$

Thus, $w = \begin{bmatrix} -2 & -2 \end{bmatrix}^\top$ and $w_0 = 3$ are potential weights and offset for a "NAND unit."

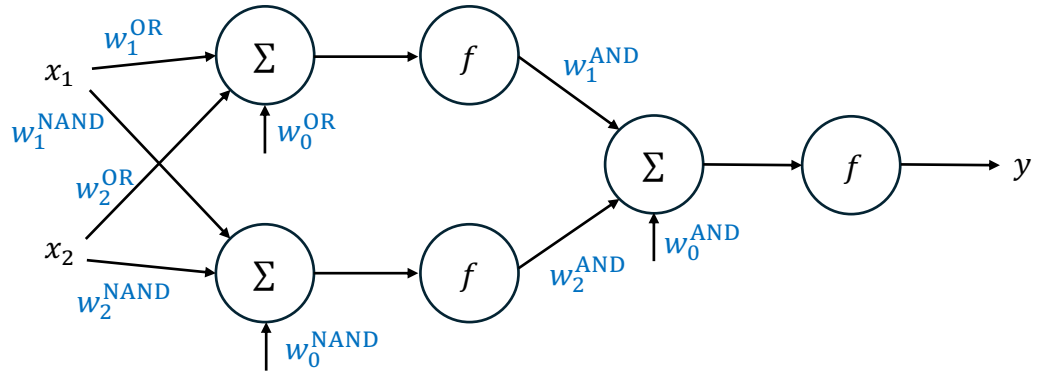
(b) Naturally, we wonder if a similar design approach could be used to construct out our favorite (well, at least most-frequently visited) Boolean expression: XOR.

It turns out that we can't express Boolean XOR using a neural network with no hidden layer. However, we *can* represent XOR using a neural network with one hidden layer.

Note for $m = 2$, XOR can be written as (x_1 OR x_2) AND ((NOT x_1) OR (NOT x_2)).

Let $(w_1^{\text{OR}}, w_2^{\text{OR}}, w_0^{\text{OR}})$, $(w_1^{\text{AND}}, w_2^{\text{AND}}, w_0^{\text{AND}})$, and $(w_1^{\text{NAND}}, w_2^{\text{NAND}}, w_0^{\text{NAND}})$ be the weights/offsets of our OR, AND, and NAND units, respectively. Label the arrows of the neural network below using these weights/offsets, so that the network represents Boolean XOR.

Solution:



(c) That was cool! In fact, it turns out that *any* Boolean function can be represented by a neural network with a single hidden layer.

This relies on the following facts, which hold even for $m > 2$:

- We can construct a one-unit neural network representing any logical expression that is a combination of ORs and NOTs (as you showed for $m = 2$ above).
- Any Boolean formula can be represented in what is called *conjunctive normal form* - that is, the "AND" of multiple clauses containing only ORs and NOTs. A few examples of formulas in conjunctive normal form are given below for illustration:
  - $(x_1$ OR $x_2)$ AND $((\text{NOT } x_1)$ OR $(\text{NOT } x_2))$
  - $(x_1$ OR $(\text{NOT } x_2)$ OR $x_3)$ AND $(x_2$ OR $x_4)$
  - $(x_2$ OR $x_5)$ AND $(x_1$ OR x_3 OR $x_4)$ AND $((\text{NOT } x_2)$ OR $x_3)$

i. Briefly describe in words: Given an arbitrary formula in conjunctive normal form, how would you represent it as a neural network with one hidden layer, using a combination of "OR/NOT clause units" and "AND units"?

> **Solution:** For each clause in the conjunctive normal form formula, construct an appropriate "OR/NOT clause unit" representing that clause and put it in the hidden layer. Connect the appropriate inputs to each "OR/NOT clause unit," and feed the outputs of all these units to a single "AND unit" in the output layer.

ii. If a single hidden layer is enough to represent all Boolean functions, why would we ever want to use multiple hidden layers?

Hint: Note that there are 2^m potential clauses that could be included in a conjunctive normal form formula.

> **Solution:** You may need an exponential number of hidden units to represent your formula. That's a lot of units! Using more hidden layers may allow us to get an equally expressive neural network using fewer units overall.