6.036: Midterm Exam, Fall 2021

Solutions

- This is a closed book exam. One page (8 1/2 in. by 11 in.) of notes, front and back, is permitted. Computers, phones, and other electronics are not permitted.
- You have 2 hours.
- The problems are not necessarily in any order of difficulty.
- Write all your answers in the places provided. If you run out of room for an answer, indicate that you are continuing your answer, use the provided blank page at the end, and mark clearly what question is being continued.
- If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.

Name: _____

Kerberos (MIT username):_____

Question:	1	2	3	4	5	6	7	8	9	10	11	Total
Points:	9	9	10	6	6	8	15	12	10	5	10	100
Score:												

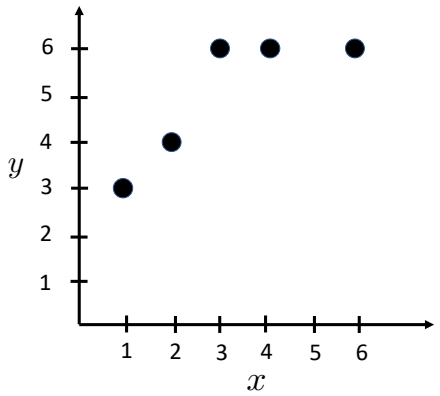
Name: _____

Beatriz and mysteries of regression

1. (9 points) Recall that ridge regression is a special case of a general recipe for constructing ML objectives,

$$J(\Theta) = \left(\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(x^{(i)}; \Theta), y^{(i)}) \right) + \lambda \mathcal{R}(\Theta),$$

where the hypothesis is $h(x^{(i)}; \Theta) = \theta^T x^{(i)} + \theta_0$, the loss is $\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$ (where $\hat{y}$ is the prediction, y the observed value), and the regularizer is $\mathcal{R}(\Theta) = ||\theta||^2$ (λ always assumed to be ≥ 0). Consider the following 1-D data set:



(a) What is the mean-squared error (MSE) on this data for the hypothesis $h(x^{(i)}) = 2x^{(i)}$?

Solution: MSE $= \frac{1}{5}(1^2 + 0^2 + 0^2 + 2^2 + 6^2)$ or $\frac{41}{5}$.

(b) Beatriz decides that for her application, small errors in the predicted y-values are irrelevant, and so she designs a new loss function $\mathcal{L}_{tol}(\hat{y}, y)$ which is 0 if $y - 2 \leq \hat{y} \leq y + 2$, and $(|y - \hat{y}| - 2)^2$ otherwise. In words, Loss(guess, actual) is 0 if guess is within 2 units of actual and the difference minus 2, squared, if guess is at least 2 units away from actual. What is the average loss using $\mathcal{L}_{tol}$ on the same data set as the previous question, assuming again the hypothesis $h(x^{(i)}) = 2x^{(i)}$?

Solution: We have

$$\mathcal{L}_{tol}(\hat{y}, y) = \begin{cases} 0, & \text{if } y - 2 \leq \hat{y} \leq y + 2 \\ (|y - y| - 2)^2, & \text{otherwise} \end{cases}$$

Therefore,

$$\frac{1}{5} \sum_{i=1}^{5} \mathcal{L}_{tol}(h(x^{(i)}), y^{(i)}) = \frac{1}{5}(0^2 + 0^2 + 0^2 + 0^2 + 4^2), \quad (1)$$

$$= \frac{16}{5}. \quad (2)$$

(c) In reviewing her 6.036 notes, Beatriz wonders why the regularizer shouldn't instead be $\mathcal{R}(\Theta) = -||\theta||^2$. Explain why this is this a bad idea.

Name: _____

Solution: This is a bad idea because we know that $\lambda \geq 0$ and for an optimization problem where we are looking to minimize the objective, the term $-\lambda||\theta||^2$ can be made to be arbitrarily large and negative (by setting θ to be larger and larger without any constraints).

Name: _____

2. (9 points) Consider the following data set with 4-dimensional data points (recall that each column represents one data point):

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 1.1 & 1.9 & 3.1 \end{bmatrix}$$

We perform ridge regression with a linear hypothesis class and no constant offset, i.e. $h(x^{(i)}; \Theta) = \theta^T x^{(i)}$.

(a) What is an optimal θ^* and its mean-squared error (MSE) for a minimizer of the ridge regression objective with $\lambda = 0$, on this data? (Note, θ^* may not be unique with $\lambda = 0$.)

Solution: There are many solutions because the number of features ($d = 4$) is larger than the number of data points ($n = 3$). The system of equations is:

$$\theta^T X = Y$$

One possible solution is:

$$\theta^* = [1, 0.1, -0.1, 0.1]^T$$

This can be seen by inspection, because Y is essentially just the first feature in X (the top row), with some corrections. The corrections are given by the one-hot encodings provided by the other features in X.

The MSE of this solution is zero.

(b) As λ becomes very large, what will the MSE be of the θ^* that minimizes the ridge regression objective? It is OK to leave unsimplified, e.g. 5^2.

Solution: As λ becomes very large, $\hat{\theta}$ will become smaller and smaller. Eventually, $\hat{\theta} = 0$. This would lead to

$$\text{MSE} = \frac{1}{3}(1.1^2 + 1.9^2 + 3.1^2) = \frac{1}{3}(14.43) = 4.81.$$

(c) Each one of the following parameter vectors was obtained by minimizing the ridge regression objective with $\lambda = .01$, 1, and 100. Which was which? (We rounded to 3 decimals.)

$\theta = [0.789, 0.078, 0.081, 0.183]^T$

Solution: $\lambda = 1$

$\theta = [0.045, 0.004, 0.006, 0.010]^T$

Solution: $\lambda = 100$

$\theta = [0.945, 0.151, 0.010, 0.258]^T$

Solution: $\lambda = .01$

Name: _____

Trial separation

Let's look at linear separability and linear classification.

3. (10 points) Linear separability.

   (a) Consider the following $n = 4$ data set with 4-dimensional data points (recall that each column represents one data point):

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} +1 & +1 & -1 & -1 \end{bmatrix}$$

   Is the data linearly separable? If yes, please provide a classifier θ, θ_0 that correctly classifies the data. If no, please explain why not.

   Solution: Yes, the data is separable. The first feature (the top row) of X completely determines Y. One solution is $\theta^T = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \theta_0 = -.5$.

   For each of the following True/False questions, please provide a brief explanation following your answer.

   (b) If we take any linearly separable data set and *add* a new feature, it is still guaranteed to be linearly separable.

   √ **True** ◯ False

   Solution: Assume that model had d features and it was separable. When adding the new feature, the associated weight θ_{d+1} can be set to 0. That would keep the data separable, as before.

   (c) If we take any linearly separable data set and *remove* a feature, it is still guaranteed to be linearly separable.

   ◯ True √ **False**

   Solution: Assume that the data was separable due to just one feature. The data set provided in part (a) is one such case where, for two of the data points, the first feature is the only differentiation between the positive and negative labels. Now assume that we remove the first feature in that example. The resulting data would no longer be separable (there are data points with identical features but different labels).

(d) If we take any data set that is not linearly separable and *remove* a feature, it is still guaranteed to not be linearly separable.

√ **True** ◯ False

Solution: Removing a feature would be the same as setting the weight associated with the removed feature to be = 0 in the original data set. However, since the original data set was not linearly separable, there is no setting of weights (including one with the removed feature's weight set to 0) that correctly classifies all of the data.

(e) If we take any data set that is not linearly separable and remove a *data point*, it is still guaranteed to not be linearly separable.

◯ True √ **False**

Solution: Consider the following datapoints and associated labels:

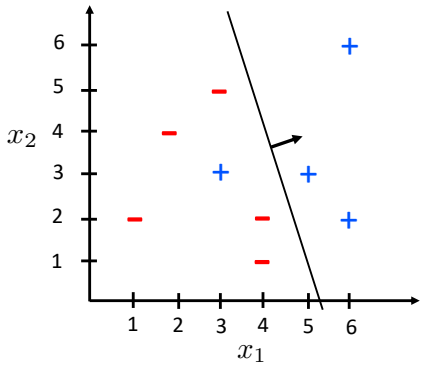
$$(1, 1) : +1,$$
$$(1, -1) : -1,$$
$$(-1, -1) : +1,$$
$$(-1, 1) : -1$$

This data set is not linearly separable (remember XOR?). Imagine we remove the datapoint $(1, 1)$. Now, the remaining data is linearly separable.

4. (6 points) Consider the data set shown in the box below.

(a) Draw a hyperplane that obtains the smallest training error (i.e., highest accuracy). Be sure to also draw the normal vector.

Solution:



(b) Suppose we remove data points ($x_1 = 3$, $x_2 = 3$) and ($x_1 = 4$, $x_2 = 2$). And let us say that two hypotheses are considered different if there exists a test point (i.e., not necessarily from the data set shown) that they would classify differently.

How many different hypotheses are there that obtain zero training error? Explain your answer.

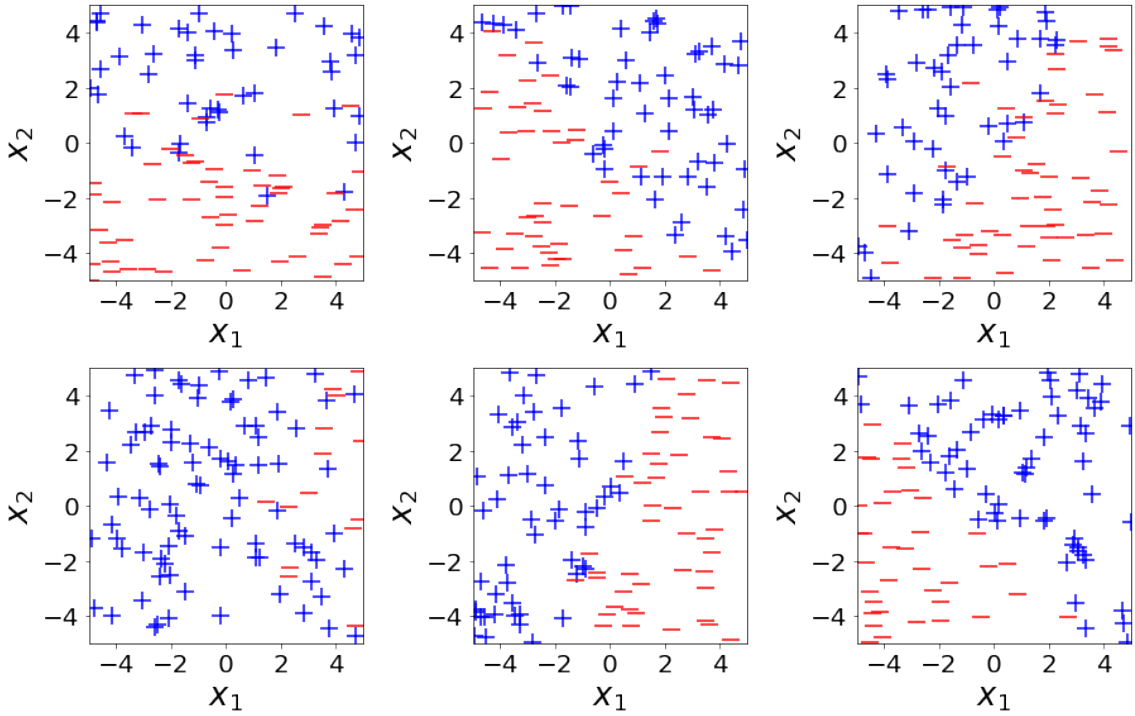
Solution: There is an infinite number of hypotheses that can obtain zero training error because the dataset is now linearly separable. These correspond to the infinite number of straight lines that can lie between the positively labeled data point $(5, 3)$ and the two negatively labeled data points $(4, 1), (3, 5)$.

Name: _____

Logistic mix-up

5. (6 points) Below we show six different data sets in 2-D. We learned an unregularized linear logistic classifier for each of them. But they all got mixed up! Please help us match each set of parameters to the data set they came from. (Each set of parameters is used exactly once.)

(a) (b) (c)



(d) (e) (f)

$\theta_1 = -6, \theta_2 = 2, \theta_0 = 0$
◯ (a) ◯ (b) ◯ (c) ◯ (d) √ **(e)** ◯ (f)

$\theta_1 = -1, \theta_2 = 1, \theta_0 = 0$
◯ (a) ◯ (b) √ **(c)** ◯ (d) ◯ (e) ◯ (f)

$\theta_1 = 0, \theta_2 = 1, \theta_0 = 0$
√ **(a)** ◯ (b) ◯ (c) ◯ (d) ◯ (e) ◯ (f)

$\theta_1 = 10, \theta_2 = 10, \theta_0 = 10$
◯ (a) ◯ (b) ◯ (c) ◯ (d) ◯ (e) √ **(f)**

$\theta_1 = -1, \theta_2 = 0, \theta_0 = 4$
◯ (a) ◯ (b) ◯ (c) √ **(d)** ◯ (e) ◯ (f)

$\theta_1 = 1, \theta_2 = 1, \theta_0 = 1$
◯ (a) √ **(b)** ◯ (c) ◯ (d) ◯ (e) ◯ (f)

Name: _____

6. (8 points) Beatriz used logistic regression on a data set derived from people living in Framingham, MA to learn a linear logistic classifier $\sigma(\theta^T x + \theta_0)$ giving the probability that an adult with features x will develop heart disease in the next decade.

   Her friend, John, would like to use the same logistic regression classifier (i.e., the θ^* and θ_0^* learned by Beatriz) to make predictions for people living in Norway. However, he notices that heart disease is much less common in Norway and thinks that the model may need to be adjusted to account for this.

   (a) Consider a specific patient with feature vector x. How could John adjust θ_0, relative to the θ_0^* learned by Beatriz, so as to make smaller the probability of this patient developing heart disease?

   Solution: Assuming all other parameters remain the same, John would need to make θ_0 smaller, i.e. $\theta_0 < \theta_0^*$.

   (b) John realizes that choosing the right value of θ_0 is tricky since he doesn't have access to any labeled data from Norway. John tells Beatriz that he only plans to use the model to find the 10% of individuals with highest probability of developing heart disease so that he can closely follow them and make sure they are tested appropriately.

   "Aha!", says Beatriz. "In that case, any value of θ_0 would suffice, and you can simply make use of my original linear logistic classifier!" Explain why Beatriz is right.

   Solution: Since σ is a monotonic function and θ_0 is a constant that does not depend on x, the ranking of patients according to $\sigma(\theta^T x + \theta_0)$ is the same no matter what the value of θ_0 is.

   Side remark: one could also multiply θ and θ_0 by any constant strictly greater than 0 and we would still get the same ranking.

Name: _____

Machine Grading

7. (15 points) Prof. Regu LaRisashun has just joined the 6.036 team, and they are excited to help teach students about machine learning. In particular. Prof. Regu (as they are fondly called) wants to try reducing stress by eliminating the final exam. They believe that nanoquizzes and homeworks should be sufficient to predict exam performance.

   Specifically, Prof. Regu takes the homework and nanoquiz grades (x), and runs a linear regression with hypothesis $\hat{y} = \theta^T x + \theta_0$ to make predictions ($\hat{y}$) for students' midterm grades. They minimize an objective function with just mean square error between the predicted and actual midterm grades (y). Data from 70% of the students are used for training, and the remaining 30% for evaluating the model.

   The initial results do not look so good, but Prof. Regu understands that this often happens with a simple linear model, and it can help a great deal to model and encode features more thoughtfully. Prof. Regu thus writes a problem for the midterm exam, asking students to help make the final exam unnecessary, by exploring five specific ideas.

   (a) Majors

   Prof. Regu notices that some students find the homework questions harder than other students, and believes this could be due to what students have studied in their other classes. Specifically, Prof. Regu notices that EECS majors seem to do better on homeworks than Physics majors. Fortunately, at MIT students' majors are conveniently coded up as a number (e.g. 1 = Civil engineering, 2 = Mech. Eng., 6 = EECS, 8 = Physics, 15 = Management, etc.) so Prof. Regu enters this number for each student as a new feature for the model.

   Is this a good idea? Explain why or why not. If not, what better way might you encode students' majors for the model?

   Solution: This not a good idea.

   Using course numbers (integers) as features implies an ordering between majors (which is incorrect). Using one-hot encoding is a better option.

   (b) Programming experience

   Looking more deeply, Prof. Regu notices that the coding questions seem to be very strong predictors of exam grades, but only if students' prior experience with python programming is taken into consideration. Prof. Regu obtains data from an initial survey students filled out at the start of the semester, where they were asked to check one box on this question:

   What is your level of python programming experience?
   ( ) None ( ) Beginner ( ) Experienced ( ) Expert

   How should data from this question best be encoded for Prof. Regu's model?

   Solution: Prof Regu should use Thermometer encoding to encode programming experience, because the items are ordered, but there is no definitive notion of distance between the items.

An integer or other real number encoding would preserve order, but improperly impose a specific notion of distance.

(c) Name of students

In a fit of exhaustion, after too many days filled with administrative Zoom meetings, Prof. Regu notices that students' exam grades are highly correlated with their names. They decide to one-hot encode each of the names of the $\sim$ 500 students, and use these new features in their model.

How good would you expect the model to perform with this approach? Discuss both the test error and the training error.

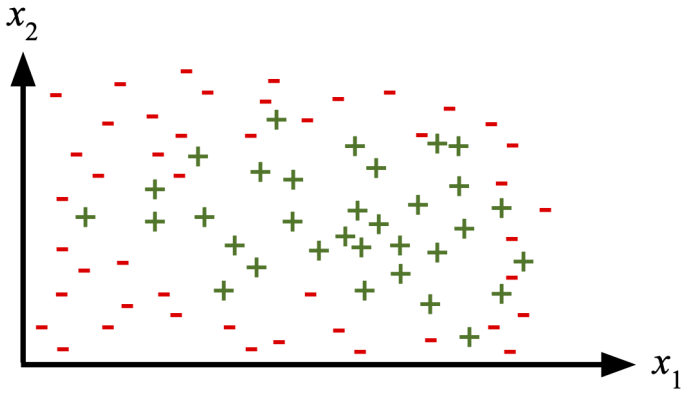
Solution: The resulting model will not perform well. The one-hot encoding will lead to zero or very low training error, because each name will become matched to the corresponding student grade. This is overfitting. However, the test data will have names which were unseen in the training data, and thus predictions for these new names will be poor. Thus, test error will be very high.

(d) Time on task

A kind colleague at the Harvard Graduate School of Education tells Prof. Regu about an interesting experiment: apparently, the Educational Testing Service (which administers major tests like the GRE) is looking at the amount of time students take to answer questions, as a measure for students' understanding of the material. The idea is that a more skilled student should be able to answer questions faster than a less skilled one.

Inspired by this idea, Prof. Regu mines data about how long students are taking to complete 6.036 nanoquizzes (x_1) and homeworks (x_2). Prof. Regu also changes approach: instead of predicting exam grades, Prof. Regu just tries to predict whether the student passes ($y = 1$) or fails ($y = 0$) the midterm exam based on just these x_1 and x_2 data. They employ linear logistic regression, with hypothesis $\hat{y} = \sigma(\theta^T x + \theta_0)$.

However, this model performs poorly! Prof. Regu plots the data to try and understand why, and sees this (+ indicates $y = 1$, and $-$ indicates $y = 0$):



Apparently, while it is the case that students who take a long time on nanoquizzes and homeworks indeed tend not to pass the exam, students who take a very short amount of time also tend not to pass! (How are some students able to finish entire homework assign-

ments in just a few minutes?) Prof. Regu decides to try to fix the model to accommodate this peculiar behavior, by employing a feature transform $\phi(x)$, and using the hypothesis $\hat{y} = \sigma(\theta^T \phi(x) + \theta_0)$.

Specify a mathematical function $\phi(x)$ which substantially improves the training error for these data:

Solution:

This is much like the example given in lecture, about predicting heart attacks based on blood pressure and heartbeat rate, where the idea is to allow the model to identify an elliptical region of the form $(ax_1 - b)^2 + (cx_2 - d)^2$. We want the center of this region to be determined by model parameters, and how to do this can be seen by expanding the expression, giving $a^2x_1^2 - 2abx_1 + b^2 + c^2x_2^2 - 2cdx_2 + d^2$. So if our model were of the form $\theta_1 x_1^2 + \theta_2 x_1 + \theta_3 x_2^2 + \theta_4 x_2 + \theta_5$, then it should fit the data well.

Thus, recognizing that the dimension of the output feature space can be larger than the input, let's just include all the polynomial terms needed for such a model:

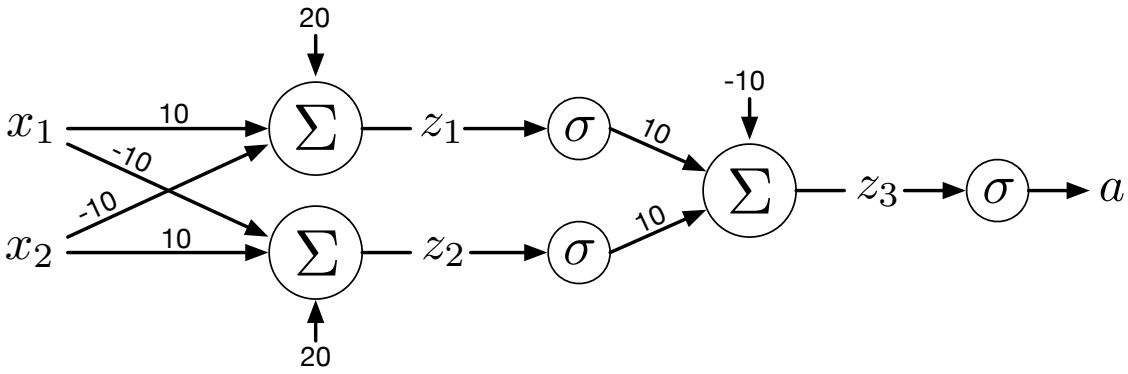
$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}. \tag{3}$$

Here, we also include an $x_1 x_2$ term, to allow for possible rotation of the axes of the ellipse.

Name: _____

Love at first spike

8. (12 points) Dr. Ne Twork is on the verge of a major discovery, and needs your help. She has identified two neurotransmitter chemicals in the brain, which are the key to a person falling instantly in love with another. Moreover, after years of functional magnetic resonance imaging studies, she believes the essence of the mechanism is the following network of neurons:



where she has written weights above and below arrows, included offset inputs, used σ to indicate a sigmoid activation function, and labeled three intermediate pre-activation values (z_1, z_2, and z_3), following 6.036 conventions. The inputs x_1 and x_2 are real numbers, representing concentrations of the two key neurotransmitter chemicals.

These neurons generate $a \approx 1$ (a "spike") only when input neurotransmitter concentrations x_1 and x_2 have the correct relationship, and otherwise, $a \approx 0$.

Let's help Dr. Ne Twork by working out for what values of x_1 and x_2 will the output a be predominantly high (value 1), versus being predominantly low (value 0). We can do this step-by-step.

(a) Give mathematical formulas for z_1 and z_2 as a functions of x_1 and x_2:

Solution:

$$z_1 = 10(x_1 - x_2 + 2),$$
$$z_2 = 10(-x_1 + x_2 + 2)$$

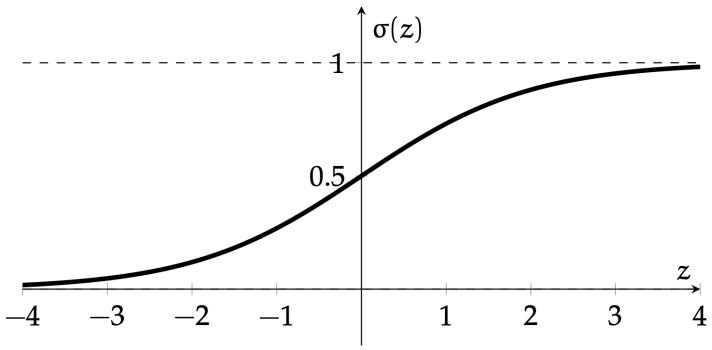
(b) Give a mathematical formula for z_3 in terms of z_1 and z_2:

Solution:

$$z_3 = 10(\sigma(z_1) + \sigma(z_2) - 1)$$

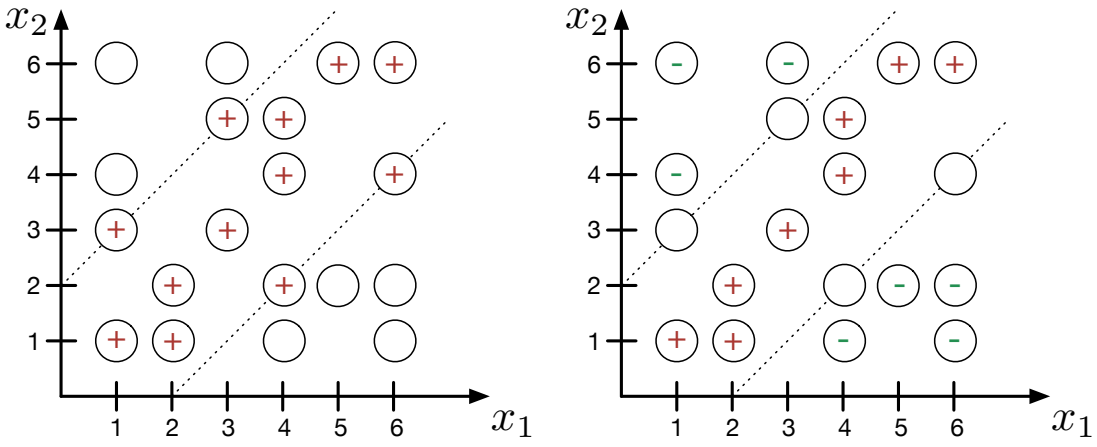
(c) Sketch a plot of a vs z_3; label the ticks on the axes with values of your choice:

Solution:
The plot of the sigmoid function where $a = \sigma(z_3)$. The plot crosses the y-axis at $z_3 = 0$.



(d) Complete the following plot of the output a, at various discrete values of x_1 and x_2, by filling in each circle with either a + (if a is close to 1) or a − (if a is close to 0). Leave a circle empty if $a = 1/2$.

Solution:
See the plot on the left, below. In addition to filling in the circles, two dotted lines are drawn in this plot; these lines separate the $a = 1$ region from the $a = 1/2$ regions:



Note that the specification of this problem has a bug. Ideally, a would go to 0, not to 1/2, when $|x_1 - x_2| >> 2$. If the offset bias going into the second Σ unit had been -15 instead of -10, then the plot would have been instead the one shown on the right.

Name: _____

(e) Complete the following sentence to provide the (pheromone-logical?) conclusion to Dr. Ne Twork's research paper about love at first spike:

Solution:
The neural network outputs a positive signal when the two neurotransmitter chemical concentrations x_1 and x_2 are within $\sim$2 of each other, i.e. $|x_1 - x_2| \lessapprox 2$.

Name: _____

Descent into code

9. (10 points) Sto Chastic is a student taking 6.036 this semester, and he prepared dilligently for the midterm exam. Unfortunately, his carefully prepared one-page of notes got eaten by a shredder, and now he needs your help derandomizing lines to answer the two questions below.

   The available lines (each prefaced with a letter, as an identifier) are:

   A: `n = y.shape[1]`
   B: `d = y.shape[0]`
   C: `j = np.random.randint(n)`
   D: `j = np.random.randint(d)`
   E: `Xj = X[j:j+1, :]`
   F: `Xj = X[:, j:j+1]`
   G: `yj = y[j:j+1, :]`
   H: `yj = y[:, j:j+1]`
   I: `th = th0`
   J: `th = th - step_size_fn(k) * dJ(Xj, yj, th)`
   K: `th = th + step_size_fn(k) * dJ(Xj, yj, th)`
   L: `th = th - step_size_fn(k) * dJ(th)`
   M: `th = th + step_size_fn(k) * dJ(th)`

   (a) Fill in the blanks below, to give correct python code implementing gradient descent as a function `gd(dJ, th0, step_size_fn, num_steps)` which takes as arguments

   - `dJ`: a function which takes as input the vector of model parameters `th`, and outputs the gradient $dJ/d\theta$ of the objective function J at $\theta =$ `th`.
   - `th0`: an initial value of model parameter vector θ, a column vector.
   - `step_size_fn`: a function that is given the iteration index (an integer) and returns a step size parameter.
   - `num_steps`: the number of iterations to perform

   The `gd` function should return the value of the model parameter vector at the final step.

   Fill in each blank with one letter (**A**, **B**, ...), corresponding to one of the available lines listed above, from Sto Chastic's notes.

```
1. def gd(dJ, th0, step_size_fn, num_steps):

2.       I: th = th0
3.     for k in range(num_steps):

4.         L: th = th - step_size_fn(k) * dJ(th)
5.     return th
```

Name: _____

(b) Fill in the blanks below, to give correct python code implementing *stochastic* gradient descent as a function `sgd(X, y, dJ, th0, step_size_fn, num_steps)` which takes as arguments

- `X`: a standard $d \times n$ data array
- `y`: a standard $1 \times n$ row vector of labels
- `dJ`: a function which takes as input a data point (column vector), a label (1×1), and a vector of model parameters `th`, and outputs the gradient $dJ/d\theta$ of the objective function J for the given data point and label evaluated at the given model parameters.
- `th0`: an initial value of model parameter vector θ, a column vector.
- `step_size_fn`: a function that is given the iteration index (an integer) and returns a step size parameter.
- `num_steps`: the number of iterations to perform

The `sgd` function should return the value of the model parameter vector at the final step.

Fill in each blank with one letter (**A**, **B**, ...), corresponding to one of the available lines listed above, from Sto Chastic's notes.

```
1. def sgd(X, y, dJ, th0, step_size_fn, num_steps):
2.     th = th0

3.     A: n = y.shape[1]
4.     for k in range(num_steps):

5.         C: j = np.random.randint(n)

6.         F: Xj = X[:, j:j+1]

7.         H: yj = y[:, j:j+1]

8.         J: th = th - step_size_fn(k) * dJ(Xj, yj, w)
9.     return th
```

Name: _____

10. (5 points) Mark each of the following statements as true or false, and provide a correct – and brief – explanation for the validity of each of your answers:

(a) The purpose of `step_size_fn` is to allow step sizes to increase with iteration index `k`, so that `sgd` and `gd` can converge faster.

◯ True √ **False**

Solution: The purpose of `step_size_fn` is to allow step sizes to get **smaller**, so that `sgd` converges.

(b) If $\theta =$ `th0` were luckily at a local **minimum** of the objective function, then the output of gradient descent `gd` would always be θ, independent of `num_steps`.

√ **True** ◯ False

Solution: At a local minimum, `dJ(th)` $= dJ/d\theta = 0$, so for `gd` the update rule `th = th - step_size_fn(k) * dJ(th)` would reduce to `th = th`

(c) If $\theta =$ `th0` were luckily at a local **maximum** of the objective function, then the output of gradient descent `gd` would always be θ, independent of `num_steps`.

√ **True** ◯ False

Solution: At a local maximum, `dJ(th)` $= dJ/d\theta = 0$, so for `gd` the update rule `th = th - step_size_fn(k) * dJ(th)` would reduce to `th = th`

(d) If $\theta =$ `th0` were luckily at a local **minimum** of the objective function, then the output of stochastic gradient descent `sgd` would always be θ, independent of `num_steps`.

◯ True √ **False**

Solution: For SGD, the update rule `th = th - step_size_fn(k) * dJ(Xj, yj, th)` means that the gradient is dependent on which datapoint is chosen. In general, this gradient will be nonzero for some datapoints, even if the gradient is zero when averaged over all datapoints.

(e) The gradient produced by `dJ` is a d-dimensional vector which points in the direction which maximizes the objective function.

√ **True** ◯ False

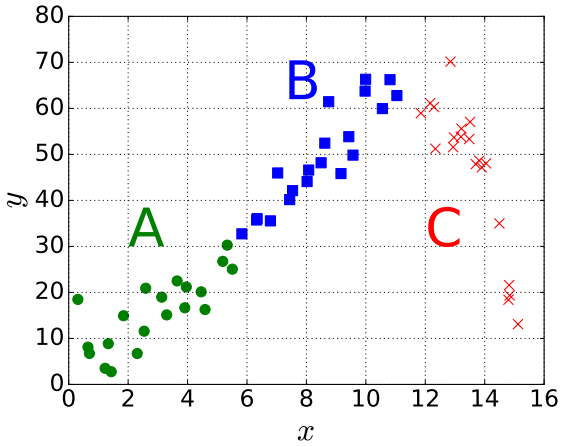
Solution: $dJ/d\theta$ points θ in the direction of increasing J.

Name: _____

ML is from Mars, Validation is...

11. (10 points) It's 2030, and MIT's Subsurface Ice eXplorer (SIX) instrument has just sent back exciting data giving the concentration of water ice y at depth x beneath the surface of the north pole on Mars!

    Your task, as one of the mission specialists (back on Earth), is to figure out what hypothesis best models the data, which look like this:



    Due to how the SIX sampling drill works, the datapoints shown in this plot come from three disjoint subsets:

    A: depth $x = 0$ to x around 6 (circles)

    B: depth x around 6 to x around 12 (squares)

    C: depth x approximately above 12 (the symbol $\times$)

    And as an ML expert, you know that while you may train your model on one subset of data, you should test it on a different subset of data.

    (a) Suppose your hypothesis is that ice concentration is linearly related to depth, i.e. $y = \theta x + \theta_0$. You employ mean square error (MSE) for the objective function, and use dataset A for training, and dataset B for testing (since they are conveniently disjoint!). Let us say that that MSE below 30 is LOW, and MSE above 100 is HIGH. Judging from the above plot, will the MSE for training be LOW or HIGH? How about for testing? Explain why.

    Solution:
    Training Error: LOW.
    Testing Error: LOW.

    Both errors are LOW because training on dataset A should produce a straight line which fits both A and B very well.

    (b) Continuing with the hypothesis that ice concentration is linearly related to depth, you now employ datasets A and B (combined) for training, and dataset C for testing. Judging

from the above plot, will the MSE for training be LOW or HIGH? How about for testing? Are your choices for training and testing datasets good ones? Explain.

Solution:
Training Error: LOW.
Testing Error: HIGH.

Training error will be LOW because training on dataset A and B should produce a straight line which fits both A and B very well. However, extrapolating forward the straight line produced will not be a good fit for dataset C leading to a HIGH testing MSE.

Are these choices for training and testing error good ones? If we are trying to model all of the data (i.e. the data in subsets A, B, *and* C), the union of subsets A and B is not representative; it misses out on the behavior in subset C. Similarly, subset C is not representative; it misses out on the behavior in subsets A&B. A better choice of training data would be one that has points from every subset; similarly, a better choice of testing data would have points from every subset.

(c) Realizing that Mars is unlikely to be a snowball of ice (although it's possible Earth once was!), you switch to a family of hypotheses with nonlinear feature transforms, $y = \theta^T \phi_k(x) + \theta_0$, where $\phi_k(x)$ is a vector of polynomials up to order k. Can you think of any good way to evaluate what order k is the best to choose? Explain.

Solution:
Training Set: randomly select data points from across all three datasets (A, B, C). A good percentage could be 80% data for training.

Testing Set: use the remaining 20% points not chosen for training to be part of the test set.

The reason one would want to choose randomly from across all datasets is because the data for training and for testing should come from the same sample distribution, even if they are disjoint datapoints.

Alternatively, use cross-validation. With cross-validation, you could use all the data for training then determine the best k by minimizing the error output by cross-validation. This would mean no need for a single separate test set.