

# Quiz Submissions - Quiz 1



## Attempt 1

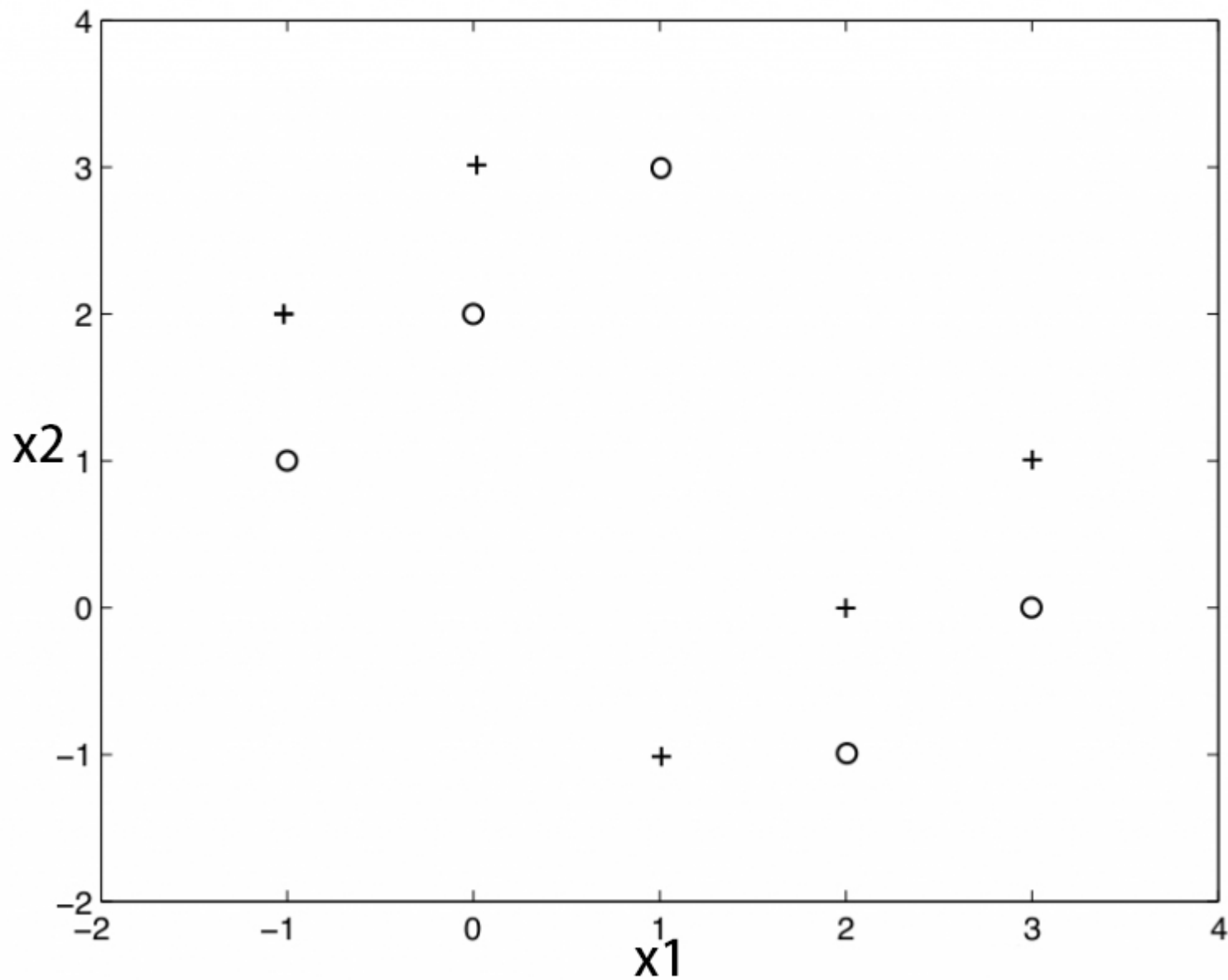
## Submission View

K-NN

### Question 1

1 / 1 point

For the following classification problem, '+' and '-' represent 2 classes. Using 1-NN method for classification, what's the accuracy of leave-one-out Cross Validation:



✓ ☒ 0%

☐ 100%

☐ 0% to 100%

☐ None of the above

▼ [Hide Feedback](#)

Using 1-NN method and leave-one-out Cross Validation, obviously, for each center point, the closest point is different from center point, thus the accuracy is 0%

## Question 2

1 / 1 point

K-NN is a linear classifier.

☐ True

✓ ☒ False

## Linear regression

## Question 3

0 / 1 point

Suppose you are optimizing a linear regression function using the closed form approach on a dataset with  $m$  features (including the bias term) and  $n$  training examples. What is the time complexity of running leave-one-out cross validation on this training set?



$$O(nm^3 + n^2m^2)$$

☐

$$O(nm^2 + n^2m^2)$$

☐

$$O(n^2m^3 + n^3m^2)$$

☒

$$O(m^3 + nm^2)$$

▼ Hide Feedback

The complexity of computing the closed form solution is

$$O(m^3 + nm^2)$$

, as discussed in class. Since we are doing leave-one-out cross-validation, we need to run the entire optimization process  $n$  times, which gives us the final complexity value.

#### Question 4

1 / 1 point

Suppose there are  $m = 17$  training data with  $n = 4$  features (bias term is not included), the number of targets is  $D' = 4$ . Linear Least Squares solution is given by  $\mathbf{W}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$ , the dimensions of  $\mathbf{W}^*$ ,  $\mathbf{X}$ ,  $\mathbf{Y}$  are:

☐

$$\mathbf{W}^* \quad 4 * 4 \quad \mathbf{X} \quad 17 * 4 \quad \mathbf{Y} \quad 17 * 4$$



$$\mathbf{W}^* \quad 5 * 4 \quad \mathbf{X} \quad 17 * 5 \quad \mathbf{Y} \quad 17 * 4$$



$$\mathbf{W}^* \quad 5 * 5 \quad \mathbf{X} \quad 17 * 5 \quad \mathbf{Y} \quad 17 * 5$$



$$\mathbf{W}^* \quad 4 * 5 \quad \mathbf{X} \quad 17 * 4 \quad \mathbf{Y} \quad 17 * 5$$

▼ Hide Feedback

As bias term should be included in linear regression, there are  $(4+1)=5$  features in total for 17 data points. Matrix  $\mathbf{X}$  is  $17 * 5$ . For this Multiple targets linear regression, we have  $D' = 4$  targets, thus  $\mathbf{Y}$   $17 * 4$  with  $\mathbf{W}^*$   $5 * 4$ .

## Logistic regression

### Question 5

0 / 1 point

Suppose that we have the following logistic prediction model:

$$P(y = 1 | \mathbf{x}) = \frac{1}{1 + e^{-0.9 + x_1 + 2x_2}}$$

Compute the predicted logit for the point

$$\mathbf{x} = [2, 1]$$

Answer:

0.0431 ✖ (-3.1)

▼ Hide Feedback

For a generic logistic regression model

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-a}}$$

the logit is equal to the "a" term (i.e., the logit is product of the linear weights and the input features).

### Question 6

0 / 1 point

For logistic function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

.

If we have  $\sigma(x_0) = 0.7$ ,

what is the derivative of the logistic function at point  $x_0$ ?

Answer:

0.2417 ✖ (0.21)

▼ Hide Feedback

For logistic function  $\sigma(x)$ , the derivative is  $\sigma(x)(1 - \sigma(x))$ , thus we can compute it easily.

### Question 7

1 / 1 point

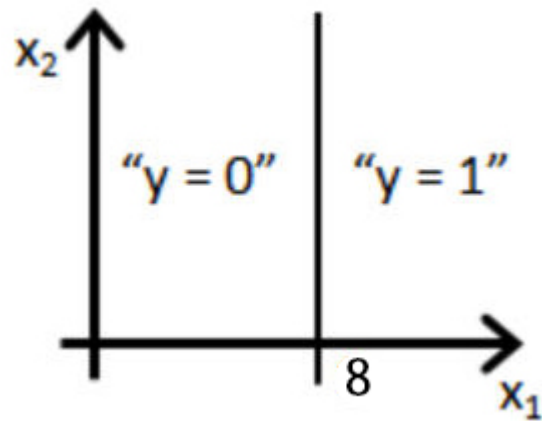
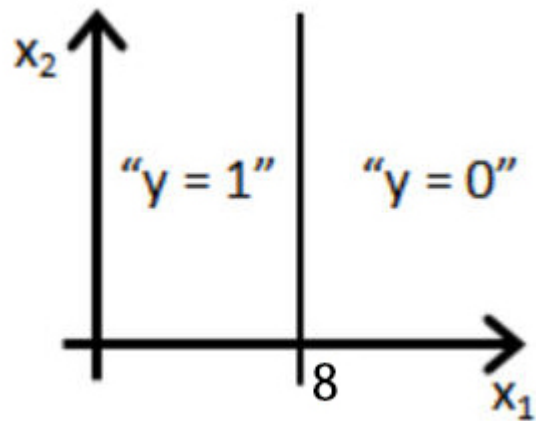
Consider the following logistic regression model for binary classification problem:

$$P(y = 1|x) = \sigma(w_0 + w_1x_1 + w_2x_2)$$

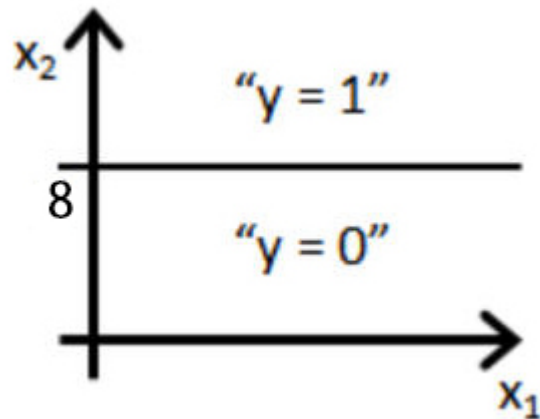
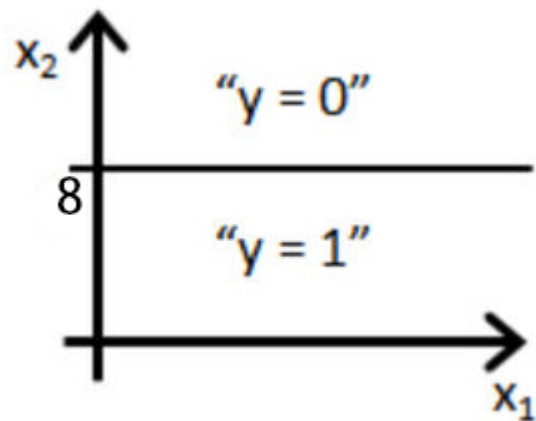
where  $w_0 = 8$   $w_1 = 0$   $w_2 = -1$ .

Which one is the correct decision boundary?









▼ Hide Feedback

When  $w^T x \geq 0$ , the prediction is 1, thus the decision boundary for  $y = 1$  is

$$\begin{aligned} 8 - x_2 &\geq 0 \\ x_2 &\leq 8 \end{aligned}$$

## Question 8

1 / 1 point

For the 3-class Softmax classification model we have weights:

$$w_1 = [1, 2, 3]$$

$$w_2 = [3, 2, 1]$$

$$w_3 = [2, 3, 1]$$

For data  $x = [2, 3, 1]$ , the predicted category is:

☐ 1☐ 2☒ 3☐ Hide Feedback

For data  $x = [2, 3, 1]$ , we have

$$z_1 = 1 * 2 + 2 * 3 + 3 * 1 = 11$$

$$z_2 = 3 * 2 + 2 * 3 + 1 * 1 = 13$$

$$z_3 = 2 * 2 + 3 * 3 + 1 * 1 = 14$$

,

with

$$\hat{y}_1 = \frac{e^{11}}{e^{11} + e^{13} + e^{14}} = 0.0351$$

$$\hat{y}_2 = \frac{e^{13}}{e^{11} + e^{13} + e^{14}} = 0.2595$$

$$\hat{y}_3 = \frac{e^{14}}{e^{11} + e^{13} + e^{14}} = 0.7054$$

.

Thus the predicted category is 3.

---

**Attempt Score:**  5 / 8 - 62.5 %

**Overall Grade (highest attempt):**  5 / 8 - 62.5 %

Done

# Quiz Submissions - Quiz 2



## Attempt 1

## Submission View

### Naive Bayes

#### Question 1

**1 / 1 point**

Suppose you have two **binary** classification datasets: Dataset A has  $m$  binary features and Dataset B has  $m$  continuous (i.e., real-valued) features. You plan to run "Bernoulli Naive Bayes" (i.e., Naive Bayes with binary features) on Dataset A and Gaussian Naive Bayes on Dataset B. Which dataset/model requires more parameters to learn?

- ☒ Gaussian Naive Bayes requires more parameters
- ☐ Binary Naive Bayes requires more parameters
- ☐ They require the same number of parameters
- ☐ Not enough information

▼ [Hide Feedback](#)

Both methods require that you learn the class distribution  $P(c)$ , so there is no difference there. Since the output is binary, estimating  $P(c)$  requires a single parameter (i.e., we need to estimate  $P(c=1)$  which gives the estimated prior class probability that  $c$  is equal to 1).

For Binary Naive Bayes, we also need to estimate  $P(x|y=1)$  and  $P(x|y=0)$  for each feature. Each of these estimates is a single parameter value, so we need to estimate  $2m+1$  parameters in total for the binary case.

In the Gaussian Naive Bayes case, we also need to estimate  $P(x|y=1)$  and  $P(x|y=0)$  for each feature. In this case, we estimate  $P(x|y=k)$  as a Gaussian and we need to learn the mean and variance for each feature, which requires 2 parameters to learn. Thus in total we have  $(2 \text{ classes}) * (2 \text{ parameters to learn the conditional distribution of each feature for each class}) * (m \text{ features}) + (1 \text{ parameter to learn the marginal likelihood of the target class}) = 4m+1$  parameters.

## Question 2

**1 / 1 point**

Naive Bayes classifier is a method when the posterior probability and the class conditional probability are known where prior probability will depend only on evidence and class-conditional probability.

- ☐ True
- ✓ ☒ False

▼ [Hide Feedback](#)

Naive Bayes classifier is a method when the prior probability is known. Then the posterior probability will depend only on evidence and class-conditional probability.

## Question 3

**0 / 1 point**

Consider we monitored Ben's class attendance and the weather for 14 days, aiming to predict based on the weather, if he is more likely to come to the class or not. We noted four different weather conditions (sky, temperature, humidity, wind) for those days, and counted for each condition the number of times he attended or skipped the class. In total, he attended 9 times and skipped 5 times during our observations. The first four tables below give the exact counts per each weather condition.

Assuming these weather conditions are conditionally independent of each other given Ben's attendance, and that today we have a cold and windy but sunny day with high humidity, what is more likely to happen?

**Attendance counts:**

Sky	yes	no	Temp.	yes	no	Hum.	yes	no	Wind	yes	no
Sunny	2	3	Hot	2	2	High	3	4	No	6	2
Cloudy	4	0	Warm	4	2	Normal	6	1	Windy	3	3
Rainy	3	2	Cold	3	1						

✖ ☒ Ben comes to the class

➡ ☐ Ben skips the class

☐ Both are equally likely

▼ [Hide Feedback](#)

Denote the event as  $E=[E1,E2,E3,E4]$ , by Bayes' rule we have

$$P(yes|E) = \frac{P(E|yes)P(yes)}{P(E)}$$

$$P(no|E) = \frac{P(E|no)P(no)}{P(E)}$$

As they are independent:

$$P(E|yes) = P(E1|yes)P(E2|yes)P(E3|yes)P(E4|yes)$$

and we have

$$P(\text{yes}|E)P(E) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$$

$$P(\text{no}|E)P(E) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206$$

$$P(\text{yes}|E) \leq P(\text{no}|E)$$

Therefore it is more likely that Ben would not go to the class.

#### Question 4

0 / 1 point

Consider aforementioned problem that:

Consider we monitored Ben's class attendance and the weather for 14 days, aiming to predict based on the weather, if he is more likely to come to the class or not. We noted four different weather conditions (sky, temperature, humidity, wind) for those days, and counted for each condition the number of times he attended or skipped the class. In total, he attended 9 times and skipped 5 times during our observations. The first four tables below give the exact counts per each weather condition. Assuming these weather conditions are conditionally independent given Ben's attendance, we want to compute the probability of having a sunny cold windy day with high humidity based on our 14 days of observations, what is that probability?

**Attendance counts:**

Sky			Temp.			Hum.			Wind		
	yes	no		yes	no		yes	no		yes	no
Sunny	2	3	Hot	2	2	High	3	4	No	6	2
Cloudy	4	0	Warm	4	2	Normal	6	1	Windy	3	3
Rainy	3	2	Cold	3	1						



✖ ☒ 0.0219

➡ ☐ 0.0259

☐ 0.1125

☐ 0.001

▼ [Hide Feedback](#)

Let us denote today's weather condition as event  $E=[E1,E2,E3,E4]$ , we want to compute  $P(E)$  by using these Bayes' rules as:

$$P(yes|E) = \frac{P(E|yes)P(yes)}{P(E)}$$

$$P(no|E) = \frac{P(E|no)P(no)}{P(E)}$$

Then we can write

$$\begin{aligned}
P(E) &= (P(\text{yes}|E) + P(\text{no}|E))P(E) \\
&= P(E|\text{yes})P(\text{yes}) + P(E|\text{no})P(\text{no}) \\
&= P(E1|\text{yes})P(E2|\text{yes})P(E3|\text{yes})P(E4|\text{yes})P(\text{yes}) \\
&\quad + P(E1|\text{no})P(E2|\text{no})P(E3|\text{no})P(E4|\text{no})P(\text{no}) \\
&= 0.0259
\end{aligned}$$

Note that the assumption of Naive Bayes is conditionally independent but independent, therefore this is not correct:

$$P(E) = P(E_1)P(E_2)P(E_3)P(E_4) = 5/14 * 4/14 * 7/14 * 6/14 = 0.0219$$

### Question 5

0.33333333 / 1 point

Please select all statements that are correct.

(Multiple answers might be correct, please select all that are correct, grade is given by number of the right selections minus the wrong selections)

- ➡ ☒ ☒ logistic regression is a discriminative classifier and learns the conditional probability of target given features:  $p(y|x)$
- ➡ ☒ ☒ naive bayes is a generative classifier and learns the joint distribution of target and features:  $p(x,y)$
- ✗ ☒ k-nearest neighbour is a discriminative classifier and learns the posterior probability of targets given the features:  $p(y|x)$

▼ Hide Feedback

k-nearest neighbor finds the decision boundaries directly without considering probabilities.

## Regularization

## Question 6

1 / 1 point

A data scientist has designed a new regularizer called a “quartic non-center regularizer” that adds the following penalty to a loss function, based on the weight vector  $\mathbf{w}$ :

$$J_{\text{reg}}(\mathbf{w}) = J(\mathbf{w}) + \sum_{j=1}^m (w_j - 1)^4$$

where  $J(\mathbf{w})$  denotes the unregularized error for a model and  $w_j$  denotes the  $j$ 'th entry in the parameter vector  $\mathbf{w}$ .

Which of the following would correspond to the gradient of the error for logistic regression with this regularization?



$$\left[ \sum_{i=1}^n \mathbf{x}_i (y_i - \sigma(\mathbf{w}^\top \mathbf{x}_i)) \right] + \sum_{j=1}^m 4(w_j - 1)^3$$



$$\sum_{j=1}^m (w_j - 1)^3 + \sum_{i=1}^n \mathbf{x}_i (y_i - \sigma(\mathbf{w}^\top \mathbf{x}_i))$$



$$\left[ \sum_{i=1}^n \mathbf{x}_i (y_i - \sigma(\mathbf{w}^\top \mathbf{x}_i)) \right] + \sum_{j=1}^m 4w_j^3$$



$$\sum_{i=1}^n \mathbf{x}_i (y_i - \mathbf{x}_i \sigma(\mathbf{w}^\top \mathbf{x}_i)) + 4(w_i - 1)^3$$

▼ Hide Feedback

To obtain the error of the gradient, we need to take the usual gradient of the error for logistic (see, e.g., Lecture 4 slides 5.8) and add the gradient of the quartic regularizing term:

$$\nabla_{\mathbf{w}} \left( \sum_{j=1}^m (w_j - 1)^4 \right) = \sum_{j=1}^m 4(w_j - 1)^3$$

### Question 7

1 / 1 point

Choose the correct statement about Ridge regression:

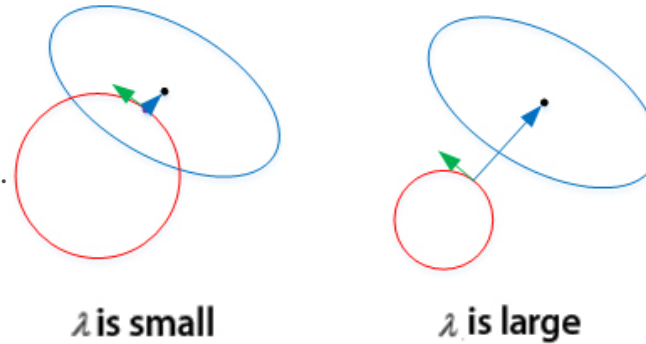
(Multiple answers might be correct, please select all that are correct, grade is given by number of the right selections minus the wrong selections)

- ✓ ☒ If  $\lambda = 0$ , it equals to linear regression.
- ✓ ☐ If  $\lambda = 0$ , it doesn't equal to linear regression.
- ✓ ☒ If  $\lambda = +\infty$ , the weights  $w$  are small, close to 0.
- ✓ ☐ If  $\lambda = +\infty$ , the weights  $w$  are large, close to  $\infty$ .

▼ Hide Feedback

If  $\lambda = 0$ , it equals to linear regression as there is no regulation term and only left is sum of squared error.

If  $\lambda = +\infty$ , it would have high strength of regularization.



$\lambda$  is small

$\lambda$  is large

### Question 8

0 / 1 point

After applying a regularization penalty in linear regression, you find that some of the coefficients of  $\mathcal{W}$  are zeroed out. Which of the following penalties might have been used?

- ☐ L0 norm
- ☒ L1 norm
- ☐ L2 norm
- ☐ L0 norm or L1 norm
- ☐ any of the above

▼ [Hide Feedback](#)

L0 norm penalizes the number of non-zero features and L1 produces sparse solutions. Thus both of them would likely produce 0 coefficients.

**Question 9****1 / 1 point**

To avoid overfitting we want to

(Multiple answers might be correct, please select all that are correct, grade is given by number of the right selections minus the wrong selections)

- ✓ ☒ use cross-validation when tuning hyperparameters
- ✓ ☒ add more samples to the dataset
- ✓ ☒ regularize the model's weights
- ✓ ☐ add more features to the dataset

▼ [Hide Feedback](#)

As discussed in the class, the cross-validation and regularization both avoid overfitting. Having more samples also makes overfitting harder since our sampled dataset represents the true data distribution more closely. Adding more features on the other hand, makes the model more complex and increases the chance of overfitting.

**Question 10****1 / 1 point**

After modelling, we find the bias is high in our model, we should

- ☐ make the model simpler
- ✓ ☒ make the model more complex

▼ [Hide Feedback](#)

If the model has high bias, it means that the model is too simple. To make the model more expressive, we can for example add more non-linear basis to the feature space.

---

**Attempt Score:**  6.33 / 10 - 63.33 %

**Overall Grade (highest attempt):**  6.33 / 10 - 63.33 %

Done

# Quiz Submissions - Quiz 3



## Attempt 1

## Submission View

### Regularization

### Question 1

**1 / 1 point**

Suppose you are given the following training inputs  $\mathbf{X} = \begin{bmatrix} -3 \\ 5 \\ 4 \end{bmatrix}$  and  $\mathbf{Y} = \begin{bmatrix} -10 \\ 20 \\ 20 \end{bmatrix}$ ,

Which of the following is closest to the linear regression parameter  $w$ ?

☐ 2

☐ 4.1

☒ 4.2



☐ 8.2

▼ Hide Feedback

The solution to the linear regression parameter is computed as following:

$$w = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = \frac{1}{(-3)^2 + 5^2 + 4^2} (30 + 100 + 80) = 4.2$$

**Question 2****1 / 1 point**

Using the same data as above  $\mathbf{X} = \begin{bmatrix} -3 \\ 5 \\ 4 \end{bmatrix}$  and  $\mathbf{Y} = \begin{bmatrix} -10 \\ 20 \\ 20 \end{bmatrix}$ , assuming a ridge penalty  $\lambda = 50$

, the optimal parameter  $w_2$  of ridge regression is?

☐ 4.2✓ ☒ 2.1☐ 8.4

☐ 2.2

▼ Hide Feedback

The optimal solution to the ridge regression parameter is computed as following:

$$w = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{Y} = \frac{1}{(-3)^2 + 5^2 + 4^2 + 50} (30 + 100 + 80) = 2.1$$

## Gradient descent

### Question 3

0 / 1 point

Please select all statements that are correct about methods to get the optimal parameter of the Linear regression.

(Multiple answers might be correct, please select all that are correct, grade is given by number of the right selections minus the wrong selections)

→ ✗ ☐ The closed formed solution method (**Normal Equation**) doesn't need learning rate.

→ ✓ ☒ When there are many features, the closed formed solution method is slow.

✓ ☐ When there are many features, the Gradient descent method is slow.

➡ ✓ ☒ The closed formed solution method (**Normal Equation**) doesn't need Iterative training.

▼ [Hide Feedback](#)

The closed formed solution of Linear regression is given by  $w = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$ . Obviously, there is no learning rate and Iterative training. When there are many features, the dimension of  $\mathbf{X}$  is large, and computing the inverse  $\mathbf{X}^T \mathbf{X}^{-1}$

is costly. Under this condition, the Gradient descent method is better.

#### Question 4

1 / 1 point

Gradient descent will update a set of parameters in an iterative manner to minimize an error function while Stochastic gradient descent will not.

- ☐ True
- ✓ ☒ False

▼ [Hide Feedback](#)

Both gradient descent (GD) and stochastic gradient descent (SGD) will update a set of parameters in an iterative manner to minimize an error function.

## Question 5

1 / 1 point

Match the correct descriptions about the gradient methods.

✓ \_\_\_2\_\_\_ Training with mini batch but needs to have a smaller learning rate over time to guarantee convergence.

✓ \_\_\_3\_\_\_ Keeps an exponentially decaying average of past gradients, similar to momentum.

✓ \_\_\_1\_\_\_ Training with all data and converges to a local minima but maybe slow.

✓ \_\_\_4\_\_\_ Use different learning rate for each parameter.

1. Gradient Descent (GD)

2. Stochastic gradient descent(SGD)

3. Adaptive Moment Estimation (Adam)

4. Adaptive gradient (Adagrad )

## Question 6

0.714285714 / 1 point

Select all the convex functions:

(Multiple answers might be correct, please select all that are correct, grade is given by number of the right selections minus the wrong selections)



$$f(x) = x^2$$



$$f(x) = x^3$$



$$f(x) = |x|$$



$$f(x) = x^3(x \geq 0)$$



$$f(x) = x^3(x < 0)$$



$$f(x) = \sqrt{x}(x > 0)$$



$$f(x) = \log(x)$$

▼ Hide Feedback

Obviously,

$$f(x) = x^2$$

,

$$f(x) = |x|$$

are convex functions.

$$f(x) = x^3$$

is not convex, while

$$f(x) = x^3 (x \geq 0)$$

is convex and

$$f(x) = x^3 (x < 0)$$

is Concave function.

$$f(x) = \sqrt{x} (x > 0)$$

and

$$f(x) = \log(x)$$

are monotonically increasing but not convex.

### Question 7

0.2 / 1 point

Select all sequences that satisfy Robbins Monro condition:

(Multiple answers might be correct, please select all that are correct, grade is given by number of the right selections minus the wrong selections)



$$\alpha^{\{t\}} = \frac{1}{t^{0.51}}$$



$$\alpha^{\{t\}} = \frac{1}{t^{0.48}}$$



$$\alpha^{\{t\}} = \frac{1}{t^{0.51}} + \frac{1}{t^{0.48}}$$



$$\alpha^{\{t\}} = \frac{1}{t^{0.51}} \times \frac{1}{t^{0.48}}$$



$$\alpha^{\{t\}} = \frac{1}{1.01^t}$$

▼ Hide Feedback

The p-series

$$\sum_1^{\infty} \frac{1}{t^p}$$

converges if  $p > 1$  and diverges for  $p \leq 1$ , thus if p-series satisfies Robbins Monro condition,  $p \leq 1$  and  $2p > 1$ , which is  $0.5 < p \leq 1$ .

Series  $\left(\frac{1}{t^{0.51}} + \frac{1}{t^{0.48}}\right)^2 = \frac{1}{t^{1.02}} + \frac{2}{t^{0.99}} + \frac{1}{t^{0.96}}$  diverges.

Series

$$\alpha^{\{t\}} = \frac{1}{1.01^t}$$

converges but  $(\alpha^{\{t\}})^2 = \frac{1}{1.02^t}$  also converges.

### Question 8

1 / 1 point

Using the same data as above  $\mathbf{X} = \begin{bmatrix} -3 \\ 5 \\ 4 \end{bmatrix}$  and  $\mathbf{Y} = \begin{bmatrix} -10 \\ 20 \\ 20 \end{bmatrix}$  where the model is linear

regression and the initial weight  $w^{\{0\}} = 4$ . Suppose we use Momentum method for full-batch GD where the momentum  $\beta = 0.9$  and the learning rate  $\alpha = 0.1$ , the next step weight  $w^{\{1\}}$  is ( Assume  $\Delta w^{\{0\}} = 0$ ):

☐ 3.9

☒ 4.1



✖ ☒ 4.8

☐ 4.2

▼ Hide Feedback

$$\nabla J(w^{\{0\}}) = \mathbf{X}^T (\hat{\mathbf{Y}} - \mathbf{Y}) = [-3, 5, 4] \left( \begin{bmatrix} -12 \\ 20 \\ 16 \end{bmatrix} - \begin{bmatrix} -10 \\ 20 \\ 20 \end{bmatrix} \right) = -10$$

$$\Delta w^{\{1\}} = 0.9 \Delta w^{\{0\}} + 0.1 \nabla J(w^{\{0\}}) = -1$$

$$w^{\{1\}} = w^{\{0\}} - 0.1 \Delta w^{\{1\}} = 4 + 0.1 = 4.1$$

## Evaluation

### Question 9

0 / 1 point

Suppose you have picked the hyper-parameter  $\alpha$  for a model using 10-fold cross validation. The best way to pick a final model to use and estimate its error is to

☐ pick the best of the 10 models you built; use its error estimate on its validation set

- ☐ pick the best of the 10 models you built; but report the average CV error estimate as the true error estimate
- ✗ ☒ average all of the 10 models you got; use the average CV error as its error estimate
- ➡ ☐ Train a new model on the full data set used for CV, using your best  $\alpha$ , report the error on the unseen test set

▼ [Hide Feedback](#)

Once the hyper-parameters are selected, we can use the whole train-validation set for training then report final error on the test set.

---

**Attempt Score:** 5.91 / 9 - 65.71 %

**Overall Grade (highest attempt):** 5.91 / 9 - 65.71 %

Done

# Quiz Submissions - Quiz 4



## Attempt 1

## Submission View

### SVM

#### Question 1

2 / 2 points

Consider following data points:

$x=-3, y=-1$

$x=-1, y=-1,$

$x=2, y=1,$

SVM with  $z = wx + w_0$  is trained to separate these data points, what the optimal  $w$  and  $w_0$  would be:

(2 points)

✓ ☒  $w=0.66, w_0=-0.33$

☐  $w=0, w_0=1$

☐  $w=0, w_0=0$

☐  $w=0.66, w_0=0.33$

▼ [Hide Feedback](#)

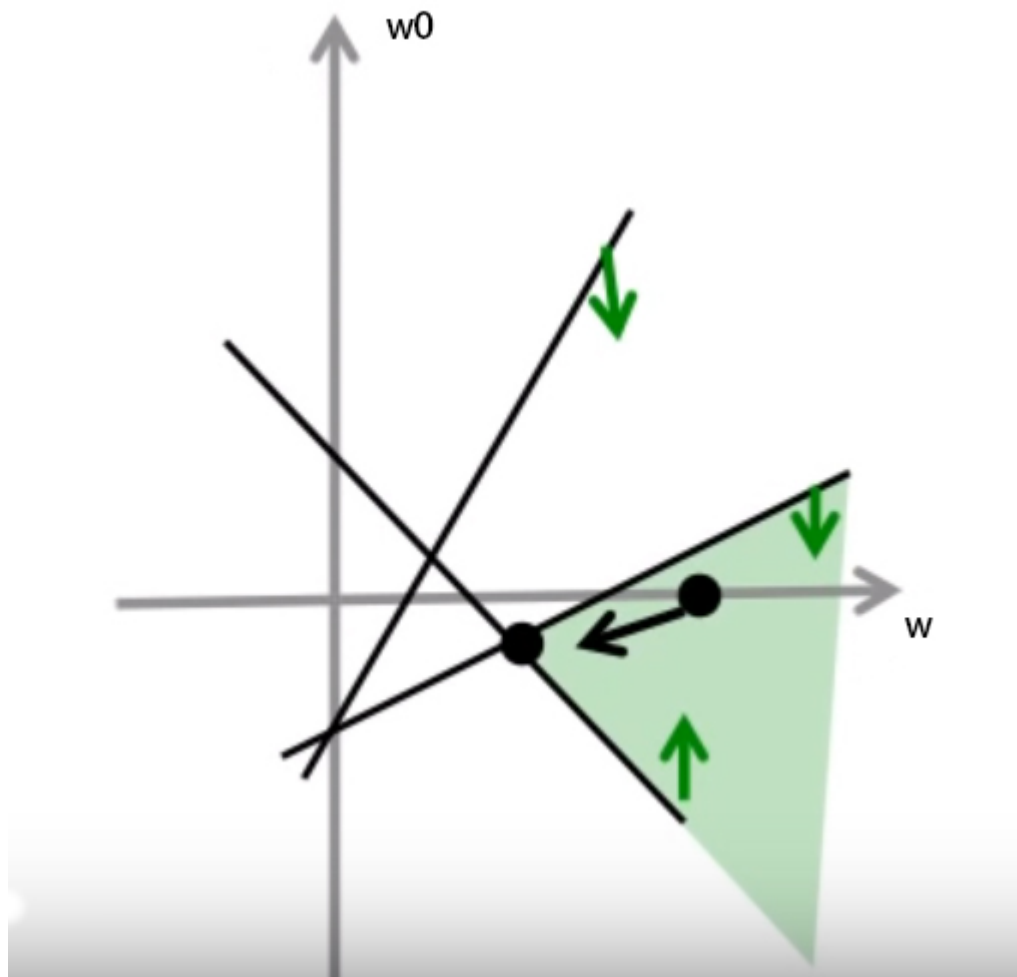
The margin constrains are:

$$-1(-3w+w_0)>1$$

$$-1(-w+w_0)>1$$

$$2w+w_0>+1$$

min  $\|w\|$ , we have  $w=0.66$  and  $w_0=-0.33$



## Question 2

0 / 1 point

Consider  $f$  the soft SVM optimization problem. Suppose we increase the value of  $\gamma$  (slides 8.2), which means that we increase the weight of the SVM hinge loss in the optimization (i.e., increasing  $\gamma$  means we pay a larger cost for misclassifying training points). Which of the following statements is most applicable in this setting:

- ☐ Increasing  $\gamma$  will tend to improve the model's accuracy on the development/test set.
- ➔ ☐ Increasing  $\gamma$  will tend to improve the model's accuracy on the training set.
- ☐ Increasing  $\gamma$  will tend to decrease the variance of the model and decrease the bias.
- ✗ ☒ Increasing  $\gamma$  will tend to decrease the variance of the model and increase the bias.

▼ Hide Feedback

Increasing  $\gamma$  increases the penalty for misclassifying training points, which means that increasing  $\gamma$  will generally lead to higher accuracy on the training set but also a risk of overfitting (i.e., increasing  $\gamma$  will generally lead to more variance and less bias in the learned solution).

### Question 3

1 / 1 point

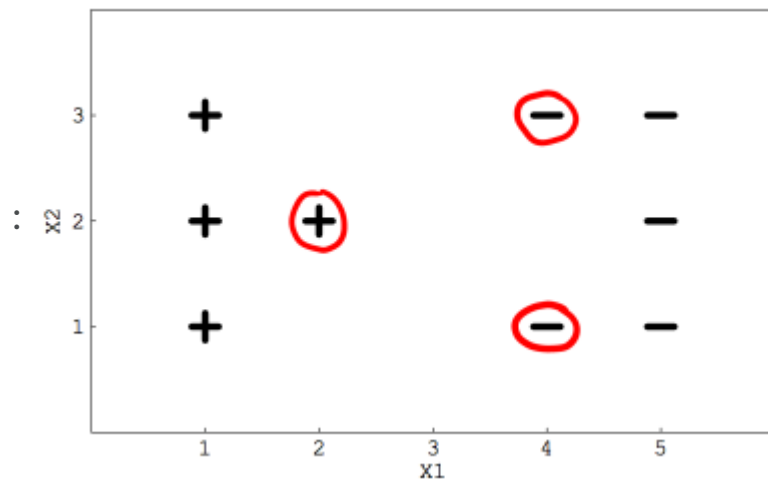
Match the similarities and differences of the SVM and Logistic regression, choose if they are belong to SVM, Logistic regression or both:

- |  |                             |
|--|-----------------------------|
| ✓ __2__ Max-margin classification          | 1. Both                     |
| ✓ __1__ Discriminative model               | 2. SVM only                 |
| ✓ __1__ Is a kind of classification method | 3. Logistic regression only |

- ✓ \_\_\_3\_\_\_ Maximum likelihood estimation
- ✓ \_\_\_1\_\_\_ Linear model (without kernel basis)
- ✓ \_\_\_3\_\_\_ All data points have same influence on the model

**Question 4****1 / 1 point**

Consider following data: if SVM is applied to this binary classification problem: if we move any red point, the decision boundary would change (non-parallel to the lines formed by them).



- ✓ ☒ True
- ☐ False

▼ [Hide Feedback](#)

These points are support vectors thus if we move any of them, the decision boundary would change.

## Perceptron

### Question 5

1 / 1 point

The steps of Perception are:

- 1 Randomly initialize the weight of the perceptron
- 2 Go to the next batch of the dataset
- 3 If the predicted value and the output are not consistent, adjust the weight
- 4 Calculate the output value for an input sample

☐ 1,2,3,4

☐ 4,3,2,1

☐ 3,1,2,4

✓ ☒ 1,4,3,2

### Question 6

1 / 1 point



For the Perceptron, given initial weight  $\mathbf{w}_0 = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}$ , and two datapoints  $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ ,  
 $y_1 = 1$  and

$\mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ ,  $y_2 = 1$ . After two steps of optimization by SGD ( $\alpha = 1$ ), what will be the weight  $\mathbf{w}_2$ :

✓ ☒ [2,4,0]

☐ [3,6,1]

☐ [1,2,-1]

☐ [2,4,-2]

▼ [Hide Feedback](#)

First data

$$\hat{y}_1 = \mathbf{w}_0 \mathbf{x}_1 = \begin{bmatrix} 1 & 2 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = -4 < 0$$

Thus we need to update

$$\mathbf{w}_1 = \mathbf{w}_0 + \alpha y_1 \mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix} + 1 * 1 * \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix}$$

For the second data we have

$$\hat{y}_2 = \mathbf{w}_1 \mathbf{x}_2 = \begin{bmatrix} 2 & 4 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} > 0$$

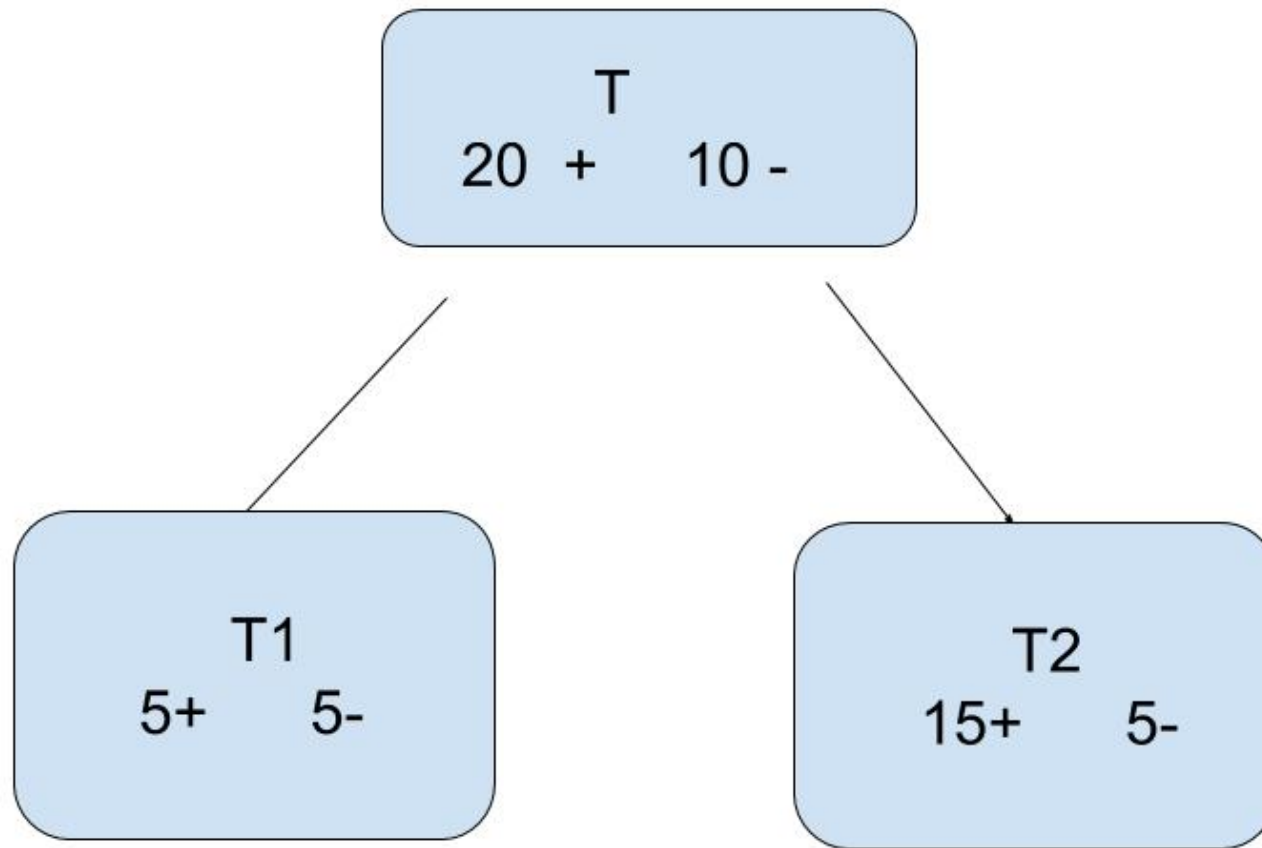
, then we do not need to update.

## Decision Tree

### Question 7

1 / 1 point

Consider following partition where classes are + and -, the classification cost (misclassification rate) is:



☐

$$\frac{1}{2}$$

☒

$$\frac{1}{3}$$

☐

$$\frac{1}{6}$$

☐

$$\frac{1}{4}$$

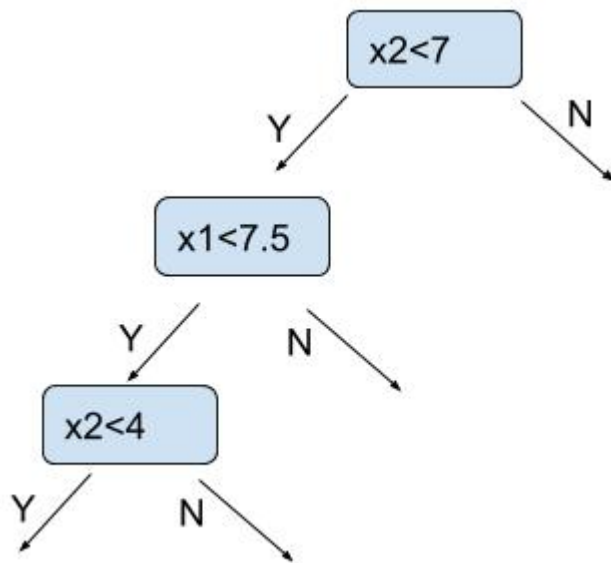
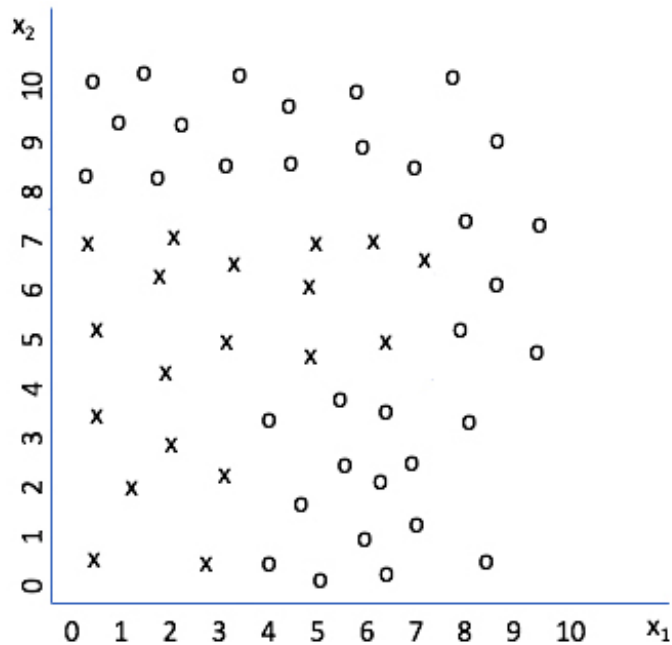
▼ Hide Feedback

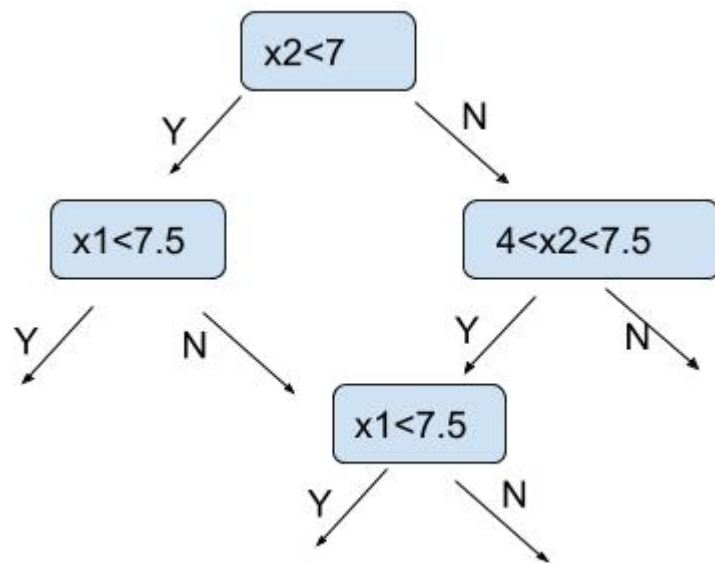
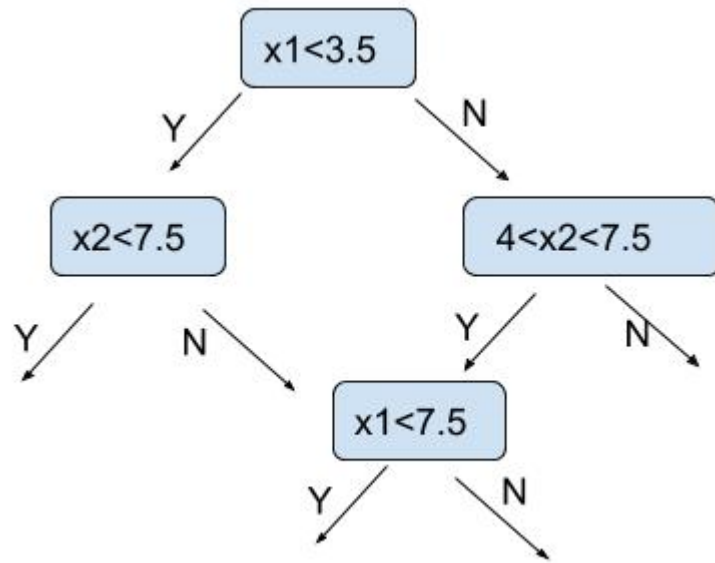
$$\frac{10}{30} * \frac{5}{10} + \frac{20}{30} * \frac{5}{20} = \frac{1}{3}$$

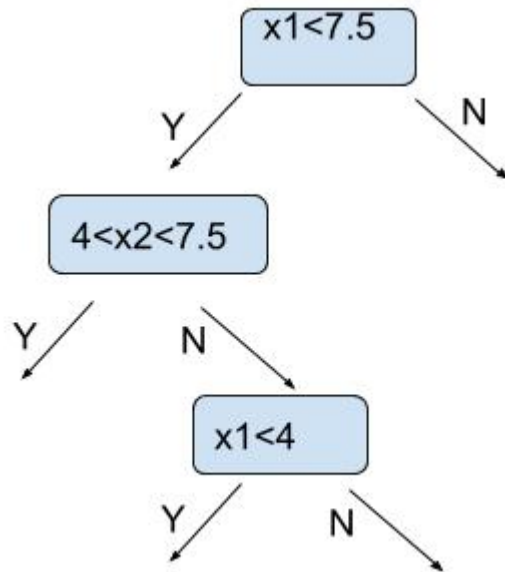
### Question 8

1 / 1 point

For the following data, which tree could achieve 100% accuracy on classification:

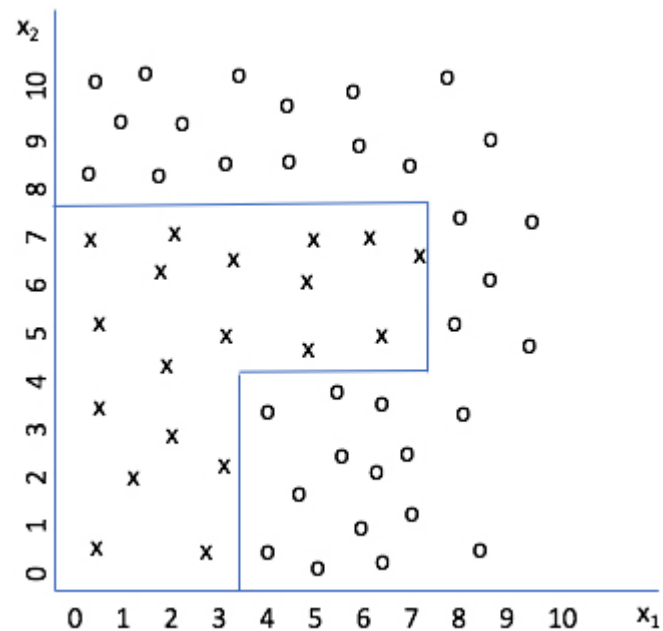






▼ Hide Feedback

Only tree 1 could achieve 100% accuracy and the region is splited as:

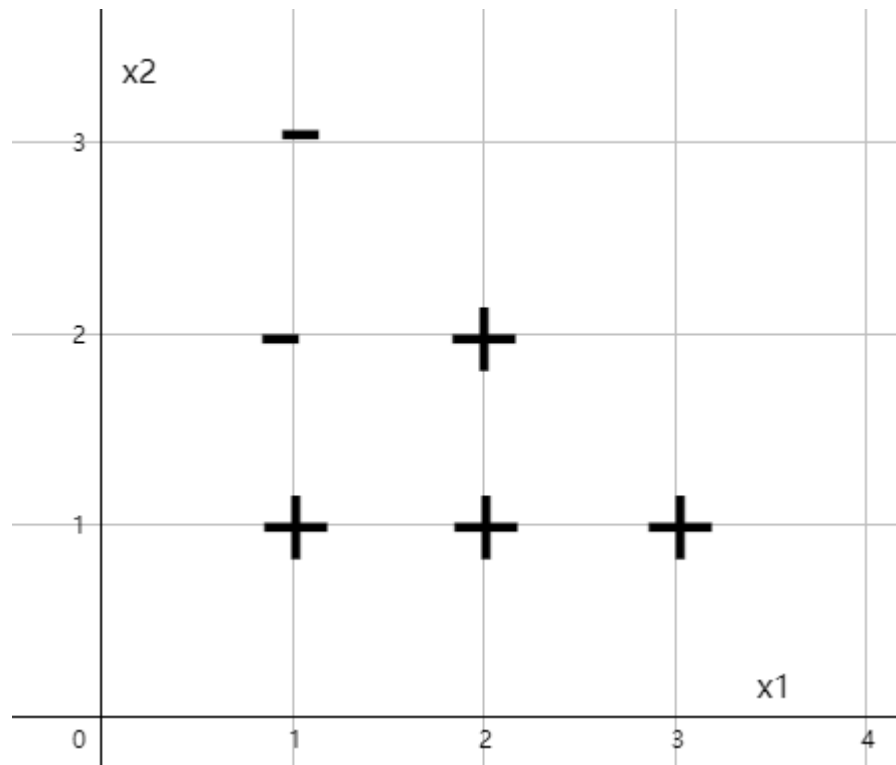


### Question 9

1 / 1 point

For the following data, using Greedy heuristic to choose the tests and misclassification loss, which of these decisions could be the first one picked?





☐  $x_1 < 1.5$

☐  $x_2 < 2.5$

☐  $x_2 < 1.5$

✓ ☒ All of above

▼ [Hide Feedback](#)

We can easily check for all the decision above, the misclassification rate is  $1/6$ , thus they have same misclassification rate.

---

**Attempt Score:**  9 / 10 - 90 %

**Overall Grade (highest attempt):**  9 / 10 - 90 %

Done

# Quiz Submissions - Quiz 5



## Attempt 1

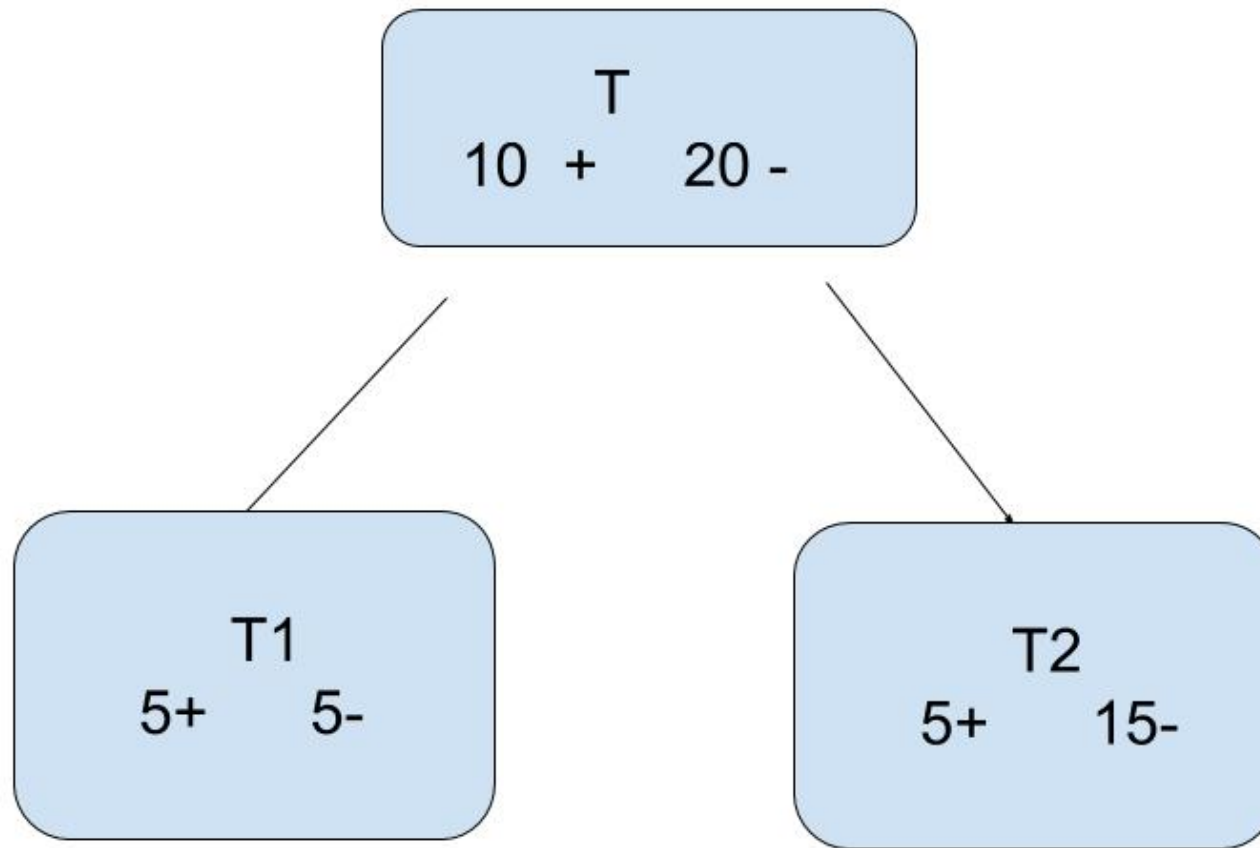
## Submission View

### Decision Tree

### Question 1

**1 / 1 point**

Consider following decision tree where the numbers denote how many samples of each class (+ or -) are in that node. Calculate the total entropy cost of this tree, when we use base 2 for the logarithm:



✓ ☒ 0.8742

☐ 0.6059

☐ 0.2632

☐ 0.1234

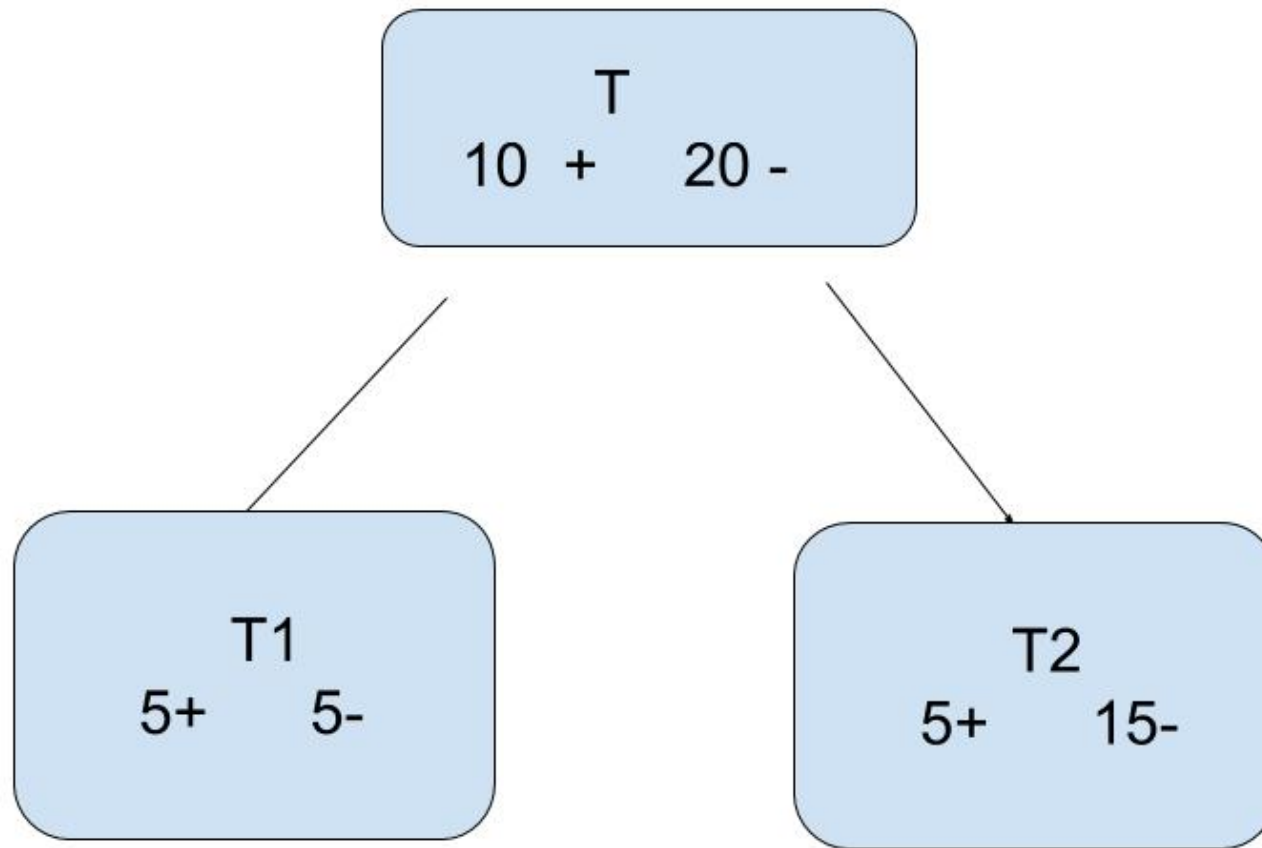
▼ Hide Feedback

$$\frac{10}{30} \left( -\frac{5}{10} \log\left(\frac{5}{10}\right) - \frac{5}{10} \log\left(\frac{5}{10}\right) \right) + \frac{20}{30} \left( -\frac{5}{20} \log\left(\frac{5}{20}\right) - \frac{15}{20} \log\left(\frac{15}{20}\right) \right) = 0.8742$$

## Question 2

1 / 1 point

Consider the same tree as previous question, what would be the Gini indexes of node T1 and T2, respectively:



✓ ☒ 0.5, 0.375

☐ 0.444, 0.5

☐ 0.444, 0.375

☐ 0.375, 0.5

▼ Hide Feedback

$$Gini(T1) = 1 - \left(\frac{5}{5+5}\right)^2 - \left(\frac{5}{5+5}\right)^2 = 0.5$$

$$Gini(T2) = 1 - \left(\frac{5}{5+15}\right)^2 - \left(\frac{15}{5+15}\right)^2 = 0.375$$

### Question 3

1 / 1 point

Select all that are true (multiselect).

✓ ☒ Trees are better than SVMs in handling mixed type input

✓ ☒ Trees are more robust than SVMs to outliers in the input space

- ✓ ☒ Trees are computationally more scalable than SVMs
- ✓ ☐ Trees have better predictive power compared to SVMs

▼ [Hide Feedback](#)

See HTF page 351

## Bootstrap, Bagging and Boosting

### Question 4

1 / 1 point

The purpose of Random forest is to reduce the bias.

- ☐ True
- ✓ ☒ False

▼ [Hide Feedback](#)

Random forest is based on bagging which is a variance reduction techniques, not a bias reduction technique.

### Question 5

1 / 1 point



AdaBoost minimizes an exponential loss function.

- ✓ ☒ True  
☐ False

▼ [Hide Feedback](#)

See slides 8.1

### Question 6

1 / 1 point

The coefficients assigned to the classifiers assembled by AdaBoost are non-negative if weak learners are better than random .

- ✓ ☒ True  
☐ False

▼ [Hide Feedback](#)

See slides 8.3

### Question 7

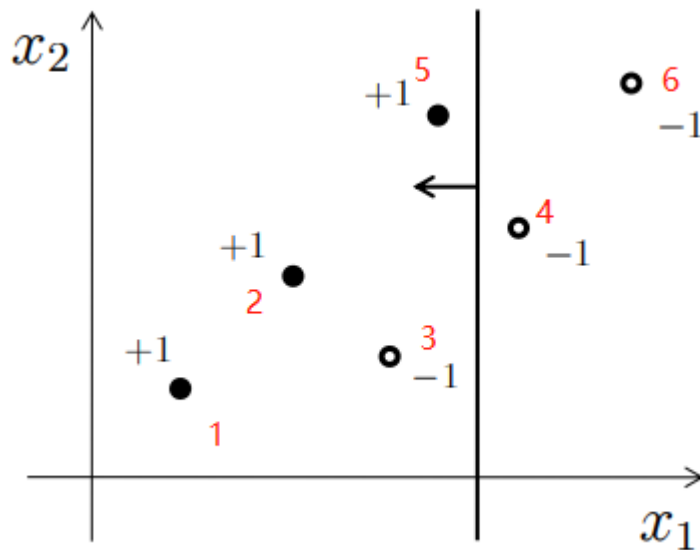
0 / 1 point

Consider following decision stumps with the AdaBoost algorithm :

$$f(x) = \text{sign}\left(\sum_t w^{\{t\}} \phi(x; s^{\{t\}})\right)$$

.

A stump predicts binary  $\pm 1$  values which depends on only one coordinate value (the split point). The line in the figure is the decision boundary of the first classifier fitted at the first iteration of the Adaboost algorithm, where the arrow indicates the positive side where the stump (predicts +1). In this algorithm, what is  $w^{\{t=1\}}$ , the weight assigned to this classifier at the first iteration?



✖ ☒ 1.1610

➡ ☐ 0.8047

☐ 0.3495

☐ 0.6931

▼ Hide Feedback

The error rate for this simple weak classifier is

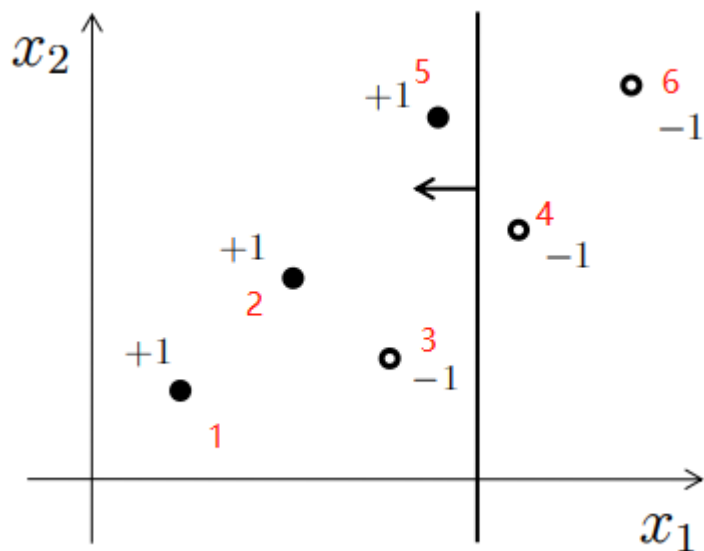
$$l^{\{t=1\}} = \frac{1}{6}$$

then the weight is

$$w^{\{1\}} = \frac{1}{2} \ln \frac{1 - l^{\{1\}}}{l^{\{1\}}} = \frac{1}{2} \ln \frac{1 - \frac{1}{6}}{\frac{1}{6}} = 0.8047$$

**Question 8****1 / 1 point**

Consider the previous setting, which point's weight will increase as a result of incorporating the first stump? this point will be more important in the subsequent iteration.



☐ 4 and 6

☐ 3 and 4

✓ ☒ 3

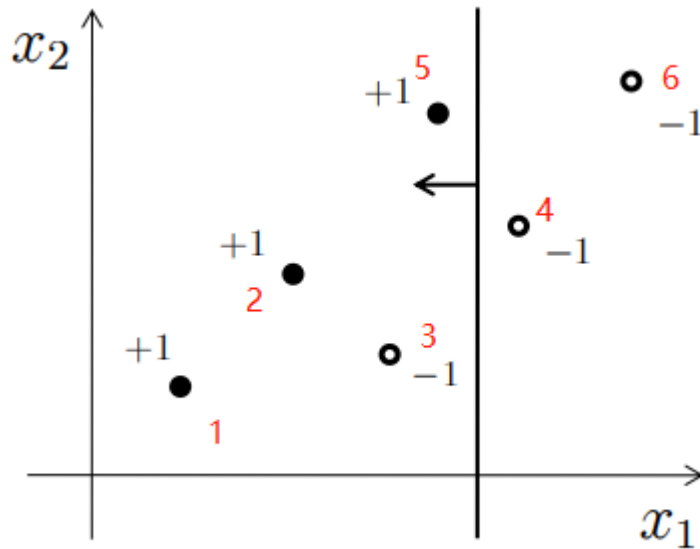
☐ 5

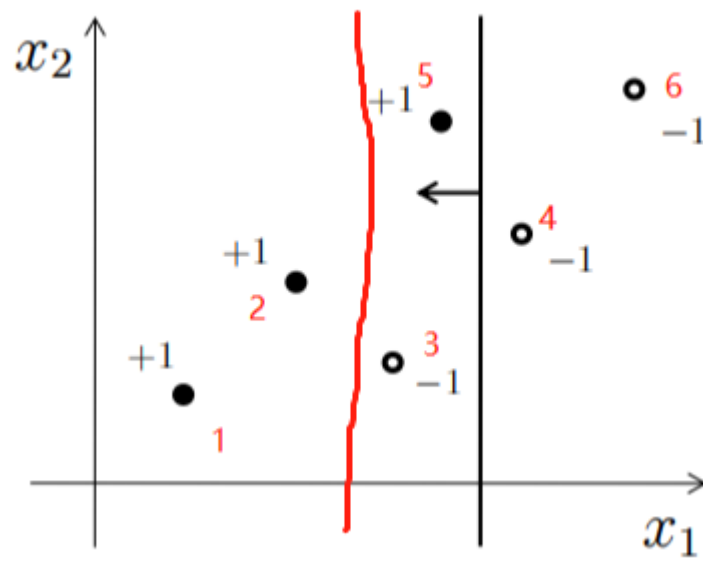
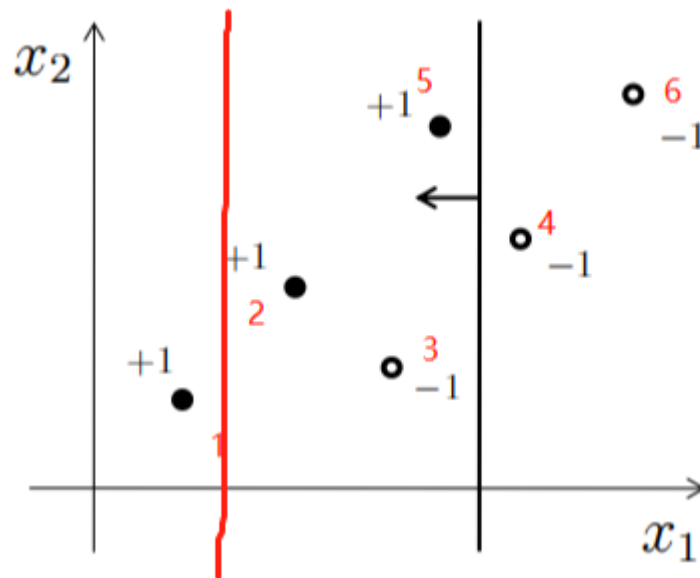
▼ [Hide Feedback](#)

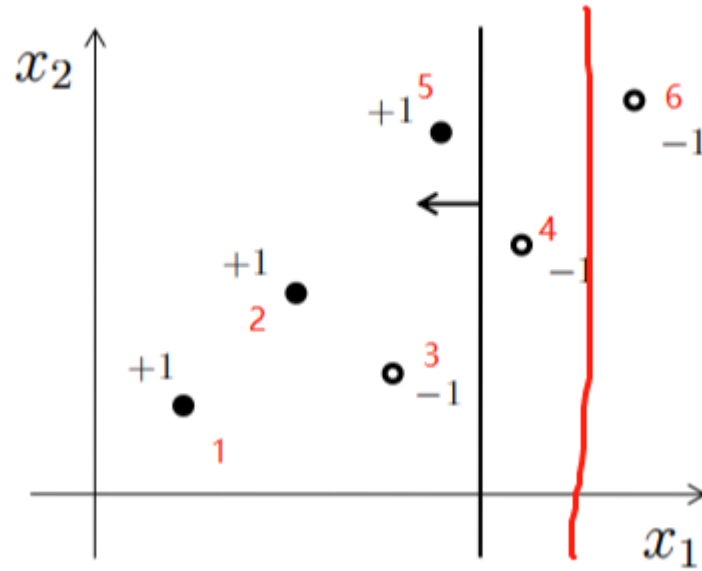
Point 3 is misclassified, thus would have higher weight next run.

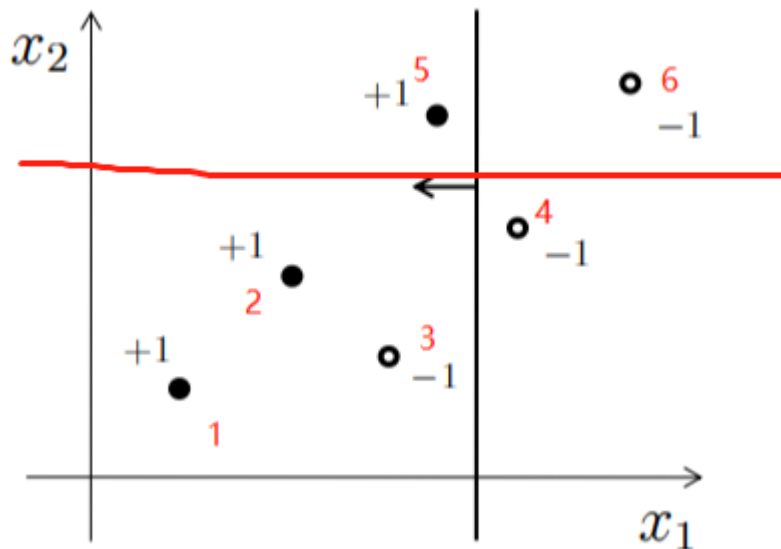
**Question 9****1 / 1 point**

Consider the previous setting again, which one would more likely be a decision boundary of the classifier fitted in the next (second) iteration of AdaBoost algorithm?









▼ Hide Feedback

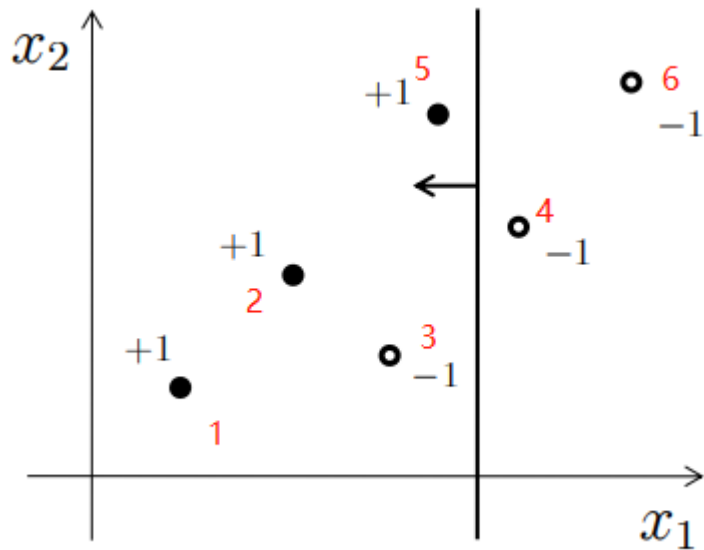
Choice 2 would have better performance as it could split 3 "-" points out and only 1 "+" is misclassified (note that all points other than 3 have the same weight since they have been correctly classified in the first iteration).

## Question 10

1 / 1 point

Consider, yet again, the same setting. After we have the  $w^{\{2\}}$  for the second iteration, the relation between  $w^{\{1\}}$  and  $w^{\{2\}}$  is:





- ☐  $w^{\{1\}} > w^{\{2\}}$
- ☐  $w^{\{1\}} = w^{\{2\}}$
- ☒  $w^{\{1\}} < w^{\{2\}}$
- ☐ Not sure

▼ Hide Feedback

The point that the second stump misclassifies will have a smaller relative weight ( $1/(6 \cdot 25)$ ) since it is classified correctly by the first stump

---

**Attempt Score:**  9 / 9 - 100 %

**Overall Grade (highest attempt):**  9 / 9 - 100 %

Done

# Quiz Submissions - Quiz 6



## Attempt 1

## Submission View

### Multilayer Perceptron

#### Question 1

**1 / 1 point**

Suppose you use the activation function X in a hidden layer in a neural network. Given a input on a particular neuron, you will get an output of -0.01. Which of the following activation functions might be X ?

- ☐ ReLU
- ☒ tanh
- ☐ Sigmoid
- ☐ All of above

▼ [Hide Feedback](#)

Only tanh function could have negative value.

## Question 2

1 / 1 point

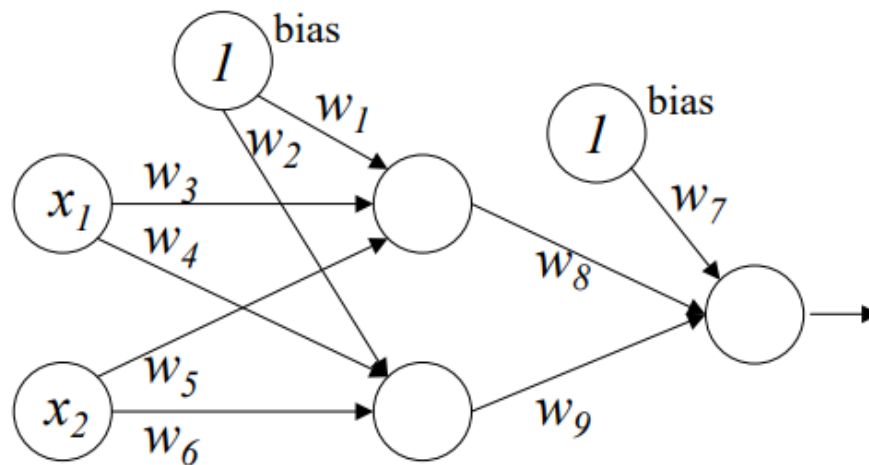
Consider a neural net for a binary classification which has one hidden layer as shown in the figure. We use a linear activation function  $h(z) = cz$  at hidden units and a sigmoid activation function  $g(z) = \frac{1}{1+e^{-z}}$  at the output unit to learn the function for

$$P(y = 1|x, w)$$

where  $x = [x_1, x_2]$  and

$$w = [w_1, w_2, \dots, w_9]$$

. Which one is the correct final classification boundary?



$$w_7 + cw_8w_1 + cw_9w_2 + (cw_8w_3 + cw_9w_4)x_1 + (cw_8w_5 + cw_9w_6)x_2 = 0$$



$$w_7 + (cw_8w_3 + cw_9w_4 + w_1w_8)x_1 + (cw_8w_5 + cw_9w_6 + w_9w_2)x_2 = 0$$

☐

$$w_7 + (cw_8w_3 + cw_9w_4)x_1 + (cw_8w_5 + cw_9w_6)x_2 = 0$$

☐

$$(cw_8w_3 + cw_9w_4)x_1 + (cw_8w_5 + cw_9w_6)x_2 = 0$$

▼ Hide Feedback

The output of this net is

$$g(w_7 + cw_8w_1 + cw_9w_2 + (cw_8w_3 + cw_9w_4)x_1 + (cw_8w_5 + cw_9w_6)x_2)$$

$$= \frac{1}{1 + \exp(-(w_7 + cw_8w_1 + cw_9w_2 + (cw_8w_3 + cw_9w_4)x_1 + (cw_8w_5 + cw_9w_6)x_2))}$$

Thus, the decision boundary is

$$w_7 + cw_8w_1 + cw_9w_2 + (cw_8w_3 + cw_9w_4)x_1 + (cw_8w_5 + cw_9w_6)x_2 = 0$$

### Question 3

1 / 1 point

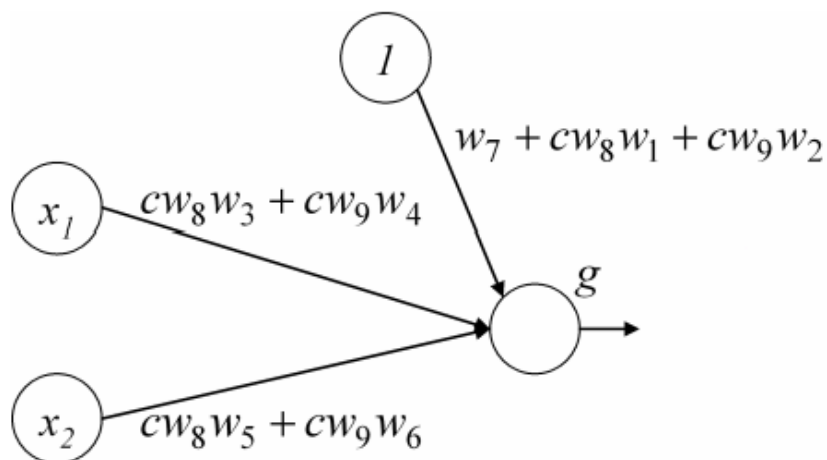
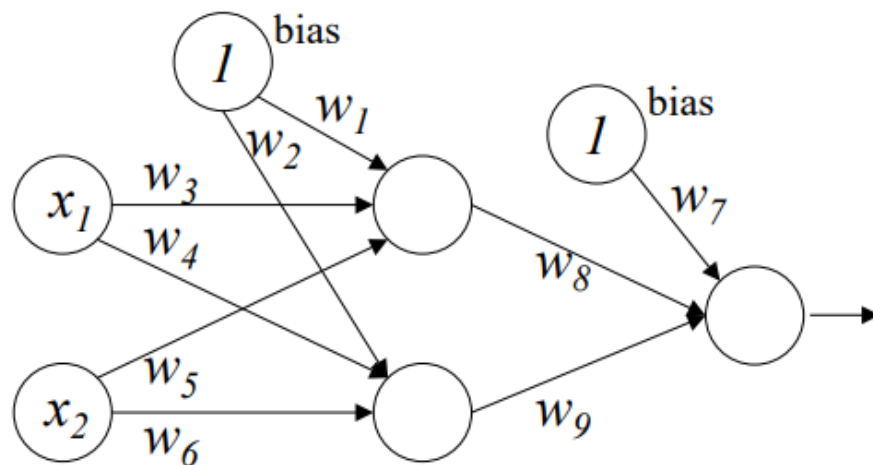
Consider a neural net for a binary classification which has one hidden layer as shown in the figure. We use a linear activation function  $h(z) = cz$  at hidden units and a sigmoid activation function  $g(z) = \frac{1}{1+e^{-z}}$  at the output unit to learn the function for

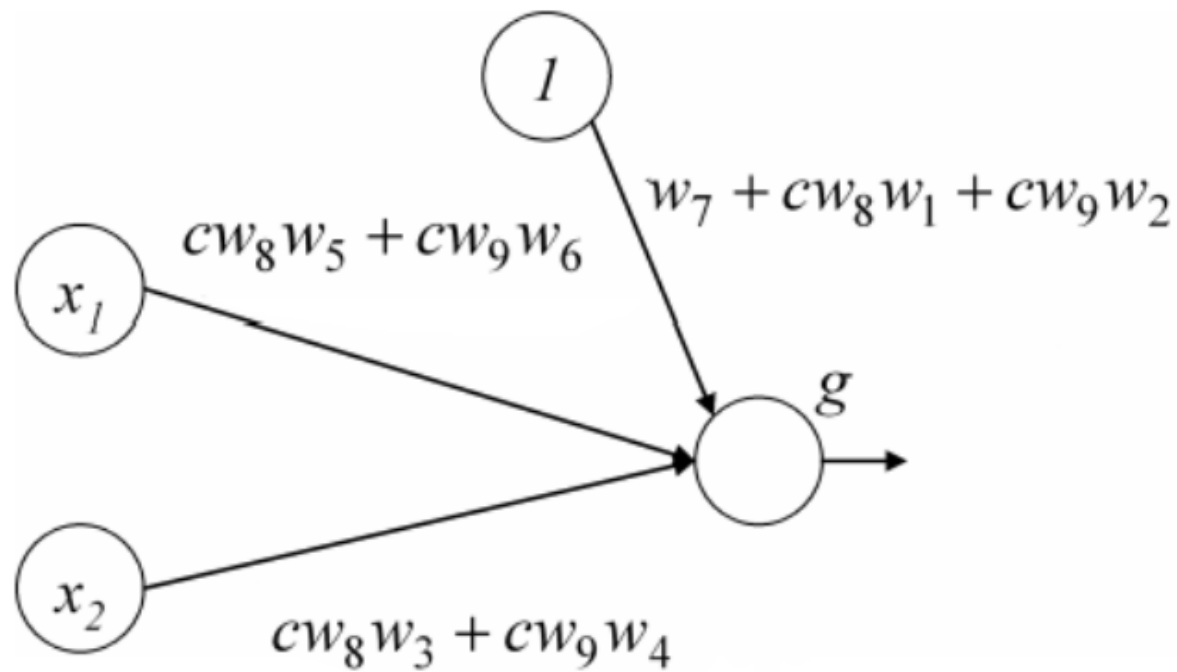
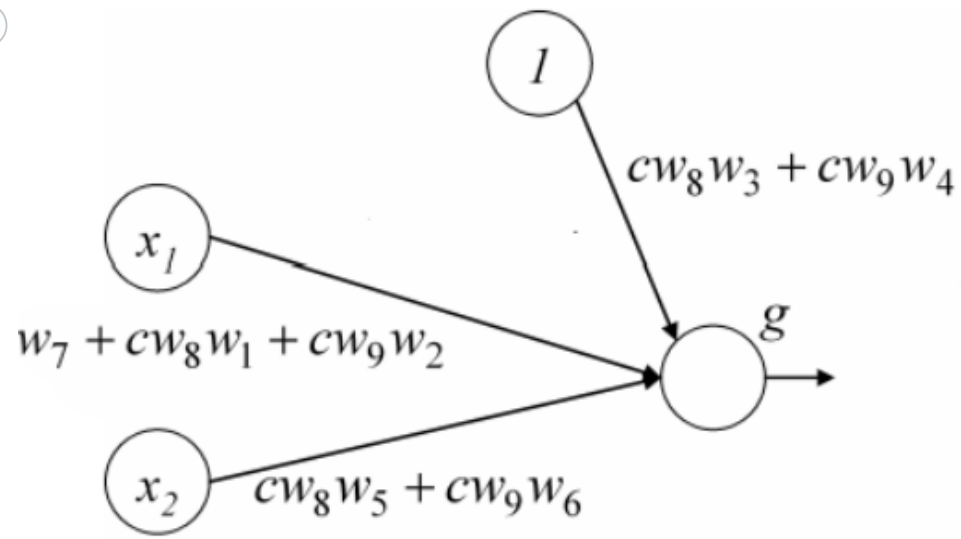
$$P(y = 1|x, w)$$

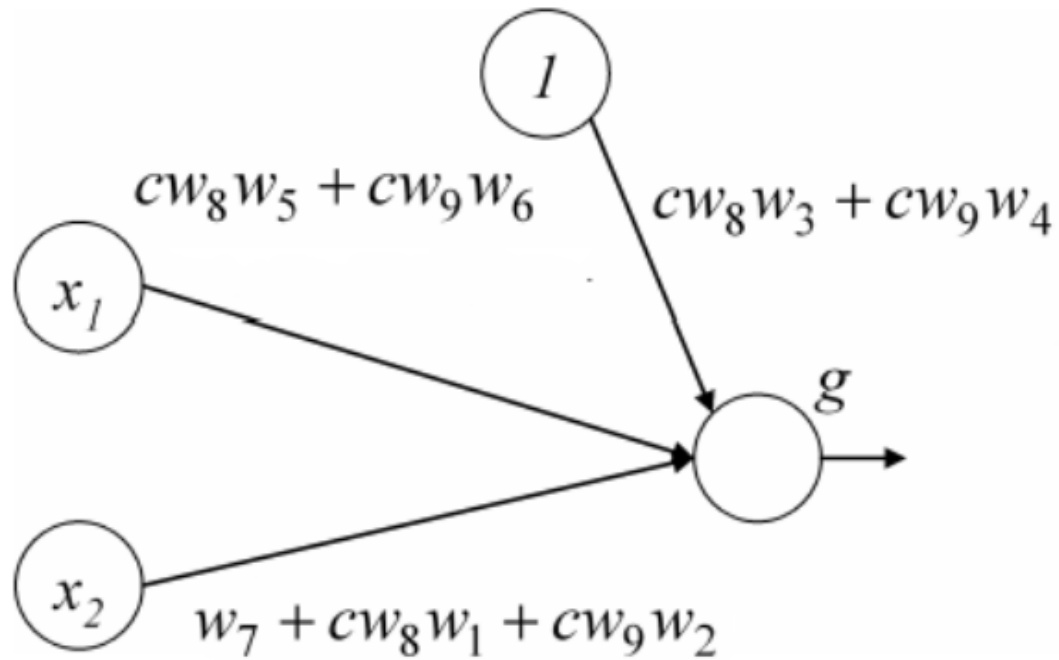
where  $\mathbf{x} = [x_1, x_2]$  and

$$\mathbf{w} = [w_1, w_2, \dots, w_9]$$

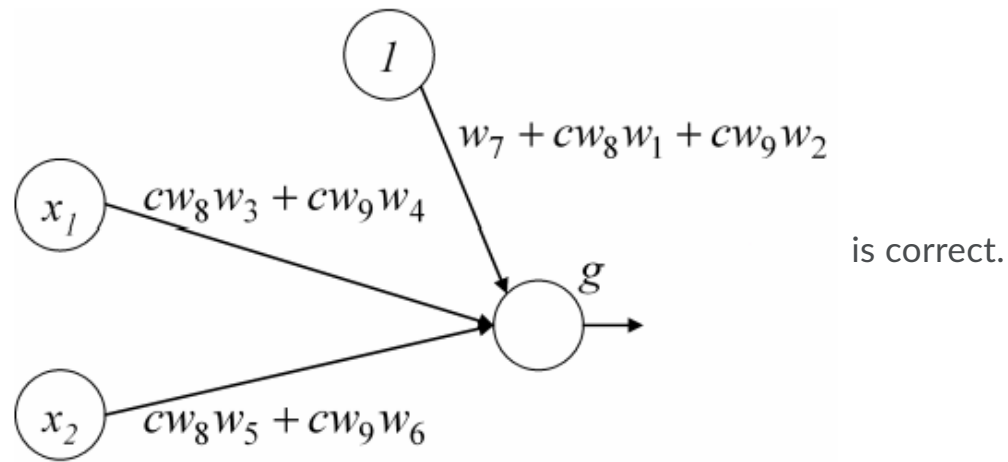
. Which one is equivalent to the given neural net?







▼ Hide Feedback





**Question 4****1 / 1 point**

Any multi-layered neural net with linear activation functions at hidden layers can be represented as a neural net without any hidden layer.

- ✓ ☒ True  
☐ False

**Question 5****1 / 1 point**

Which of the following algorithms can be constructed using neural networks:

1. KNN,
2. Linear regression
3. Logistic regression

- ☐ 1 and 2  
✓ ☒ 2 and 3  
☐ 1, 2 and 3  
☐ None

**Question 6****1 / 1 point**

A two layer neural network with linear activation functions is essentially a weighted combination of linear separators, trained on a given dataset; the boosting algorithm built on linear separators also finds a combination of linear separators, therefore these two algorithms will give the same result.

- ☐ True
- ✓ ☒ False

▼ [Hide Feedback](#)

A two layer neural network with linear activation would result in linear regression (single model) while boosting may not as it is ensemble of multiple linear separators.

### Question 7

1 / 1 point

True or False: Suppose we train any 5-hidden-layer neural network with sigmoid activation functions; then there exists a 3-hidden-layer neural network that can approximate the 5-hidden-layer network to an arbitrary accuracy.

- ✓ ☒ True
- ☐ False

▼ [Hide Feedback](#)

As discussed in Lecture 12 (e.g., slide 7.2) a single hidden layer neural network is a universal function approximator. As a consequence any neural network with more than 2 layers is also a universal function approximator (e.g., if we just make the final layer the identity function, then it becomes a 2-hidden-layer neural network, which we know is a universal function approximator). Thus, a 3-layer neural network is also a universal function approximator and any 5-layer neural network can be approximated by some 3-layer neural network.

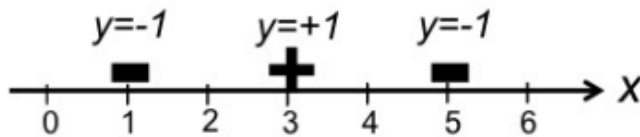
### Still Boosting

### Question 8

1 / 1 point

## [Boosting 1]

Consider following classification problem, What is the initial weight that is assigned to each data point in Adaboost algorithm?



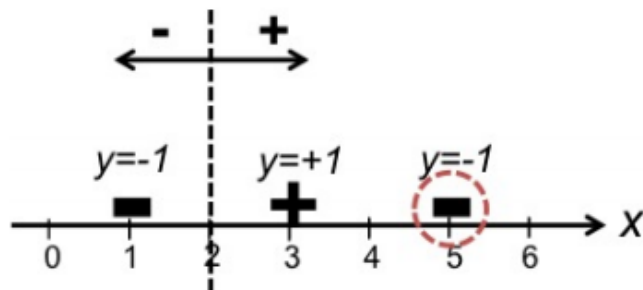
- ☐ 1
- ☐ 3
- ☒ 1/3
- ☐ 1/2

## Question 9

1 / 1 point

## [Boosting 2]

Consider following classification problem, in Adaboost algorithm if the first decision is made as following the red circled point's weight would



✓ ☒ Increase

☐ Decrease

▼ Hide Feedback

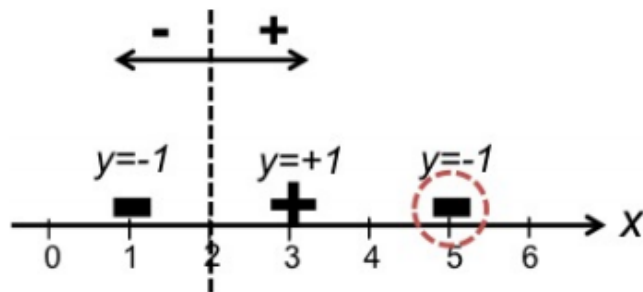
This points is misclassified thus the weight increases.

### Question 10

1 / 1 point

[Boosting 3]

Consider following classification problem, in Adaboost algorithm if the first decision is made as following the weight of points are classified correctly and the weight that is classified incorrectly are:



☐ 0.4714, 0.2357

✓ ☒ 0.2357, 0.4714

☐ 0.25, 0.5

☐ 0.5,0.25☐ Hide Feedback

$$l = \frac{1}{3}$$

$$w = \frac{1}{2} \log(2) = \log(\sqrt{2})$$

$$q(\text{correct}) = 1/3 * \exp(-\log(\sqrt{2})) = \frac{1}{3\sqrt{2}} = 0.23$$

$$q(\text{incorrect}) = 1/3 * \exp(\log(\sqrt{2})) = \frac{\sqrt{2}}{3} = 0.47$$

**Question 11****1 / 1 point**

In gradient boosting, if the input matrix is  $N \times D$ , the parameter vector  $f$  has a dimension equal to:

☒ N☐ D☐  $N \times D$ ☐ 1

---

**Attempt Score:**  11 / 11 - 100 %

**Overall Grade (highest attempt):**  11 / 11 - 100 %

Done

# Quiz Submissions - Quiz 7



## Attempt 1

## Submission View

### Question 1

1 / 1 point

Suppose we have a 2-hidden-layer neural network with the following characteristics:

- The input feature dimension is 30, and the inputs are denoted  $\mathbf{x}$ .
- The first hidden layer has dimension 25, and the vector(output) of hidden units is denoted  $\mathbf{h}^{(1)}$ . The weight matrix at this hidden layer is denoted  $\mathbf{W}^{(1)}$  and the bias vector  $\mathbf{b}^{(1)}$ .
- The second hidden layer has dimension 10, and the vector(output) of hidden units is denoted  $\mathbf{h}^{(2)}$ . The weight matrix at this hidden layer is denoted  $\mathbf{W}^{(2)}$  and the bias vector  $\mathbf{b}^{(2)}$ .
- The output layer is a binary classification task using the standard cross-entropy loss.
- Sigmoid activations are used in every layer and there is no regularization applied.

Given a single training example,  $\mathbf{x}$ , what would be the derivative of the weights  $\mathbf{w}_{\text{out}}$  at the output layer?



$$\frac{\partial \text{Err}(\mathbf{w}_{\text{out}})}{\partial \mathbf{w}_{\text{out}}} = \mathbf{x}(\sigma(\mathbf{w}_{\text{out}}^\top \mathbf{h}^{(2)} + b_{\text{out}}) - y)$$



$$\frac{\partial Err(\mathbf{w}_{out})}{\partial \mathbf{w}_{out}} = \mathbf{h}^{(2)} (\sigma(\mathbf{w}_{out}^\top \mathbf{h}^{(2)} + b_{out}) - y)$$



$$\begin{aligned} & \frac{\partial Err(\mathbf{w}_{out})}{\partial \mathbf{w}_{out}} \\ &= \mathbf{x} (\sigma(\mathbf{w}_{out}^\top \sigma(\mathbf{W}^{(2)} \sigma(\mathbf{W}^{(1)} \mathbf{x} + b^{(1)}) + b^{(2)}) + b_{out}) - y) \end{aligned}$$

☐ Not enough information is given.

▼ Hide Feedback

Note that most of the details provided are not necessary. Since we are using a sigmoid activation function at all layers and we are performing binary classification with a cross-entropy loss, the final output layer is identical to a logistic regression classifier that takes  $\mathbf{h}^{(2)}$  as input. Thus, the derivative for the weight function is identical to the derivative computed in Lecture 5, Slide 5.8, except we replace  $\mathbf{x}$  with  $\mathbf{h}^{(2)}$ .

## Question 2

1 / 1 point

The back-propagation algorithm learns a globally optimal neural network with hidden layers.

☐ True

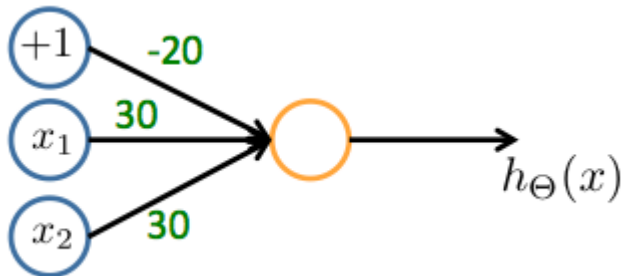
✓ ☒ False



## Question 3

1 / 1 point

Given following net where  $x_1, x_2 \in 0, 1$  and  $h_\theta(x)$  is sigmoid function, this net calculates (approximately) which of the following logical functions?



✓ ☒ OR

☐ AND

☐ XOR

☐ NAND

▼ [Hide Feedback](#)

We could easily compute that

$$\begin{array}{llll}
 x_1 = x_2 = 1 & h > 0 & - > 1 \\
 x_1 = 1 & x_2 = 0 & h > 0 & - > 1 \\
 x_1 = 0 & x_2 = 1 & h > 0 & - > 1 \\
 x_1 = 0 & x_2 = 0 & h < 0 & - > 0
 \end{array}$$

**Question 4****1 / 1 point**

Given following net where the learning parameters(weights) are

$$\Theta_1 = \begin{bmatrix} 1 & 1 & 2.4 \\ 1 & 1.7 & 3.2 \end{bmatrix}$$

and

$$\Theta_2 = [1 \quad 0.3 \quad -1.2]$$

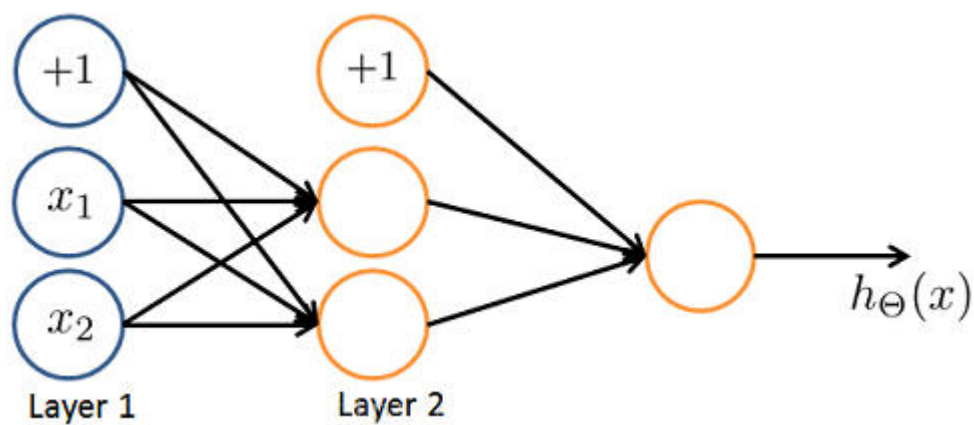
for the first and second layer (from left to right), if we exchange the parameters into

$$\Theta_1 = \begin{bmatrix} 1 & 1.7 & 3.2 \\ 1 & 1 & 2.4 \end{bmatrix}$$

and

$$\Theta_2 = [1 \quad -1.2 \quad 0.3]$$

, the output  $h_{\theta}(x)$  would :



- ☐ Increase
- ☐ Decrease
- ✓ ☒ Stay same
- ☐ None of above

▼ Hide Feedback

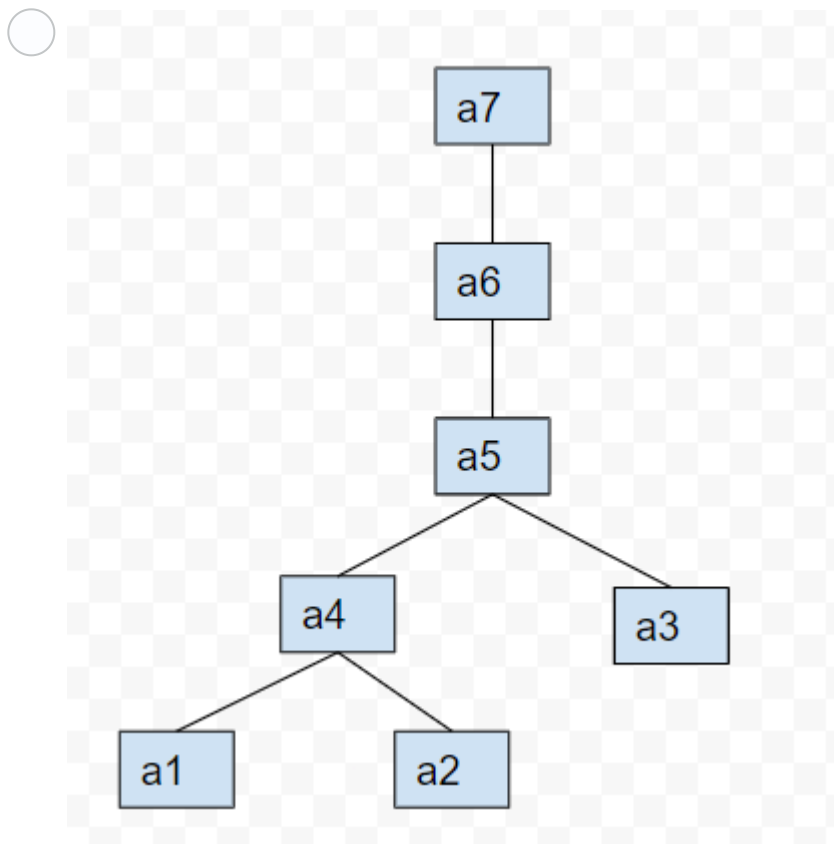
If we changed the matrix in this way, it's equivalent to change the position of neurals in the second layers, thus the output would not change.

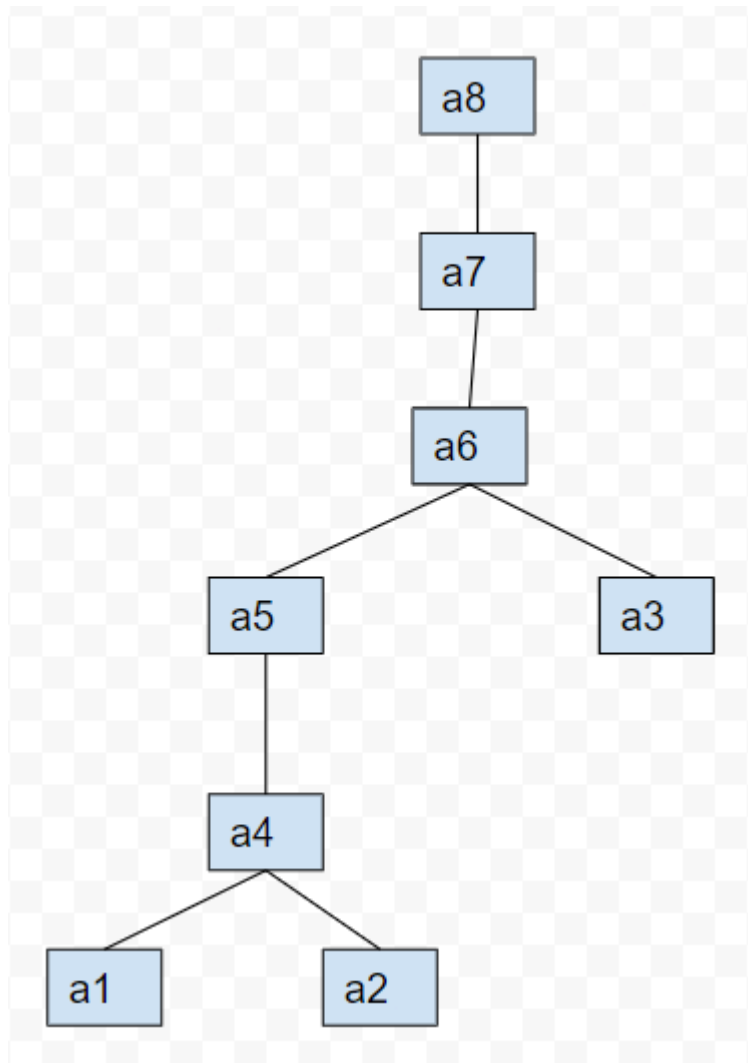
**Question 5****1 / 1 point**

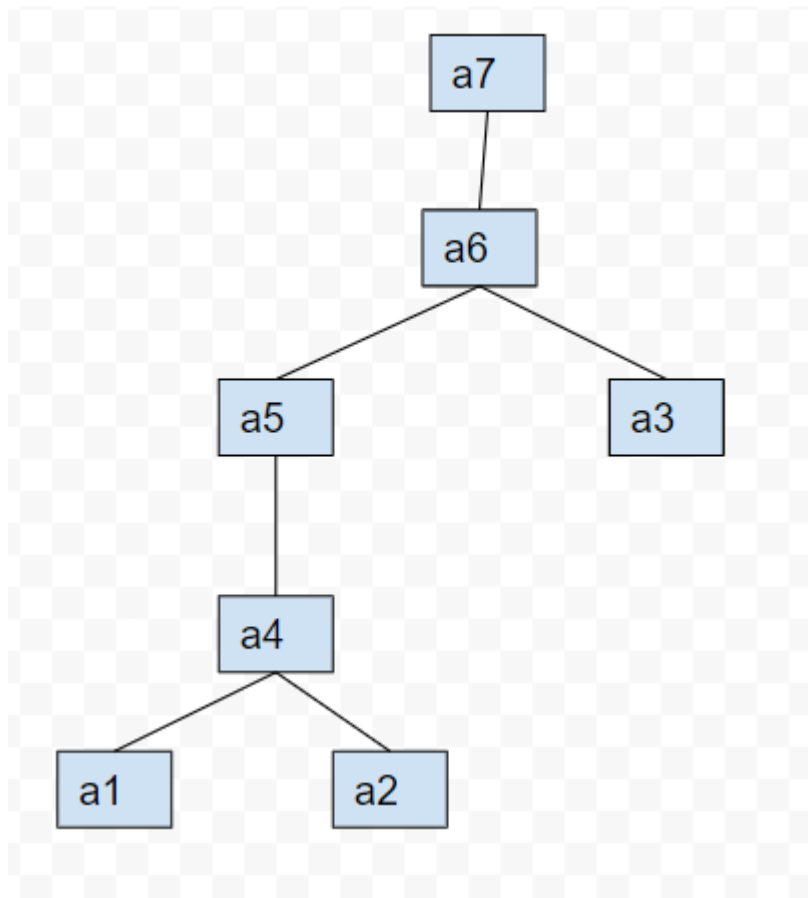
[ Computational graph 1] Consider the model

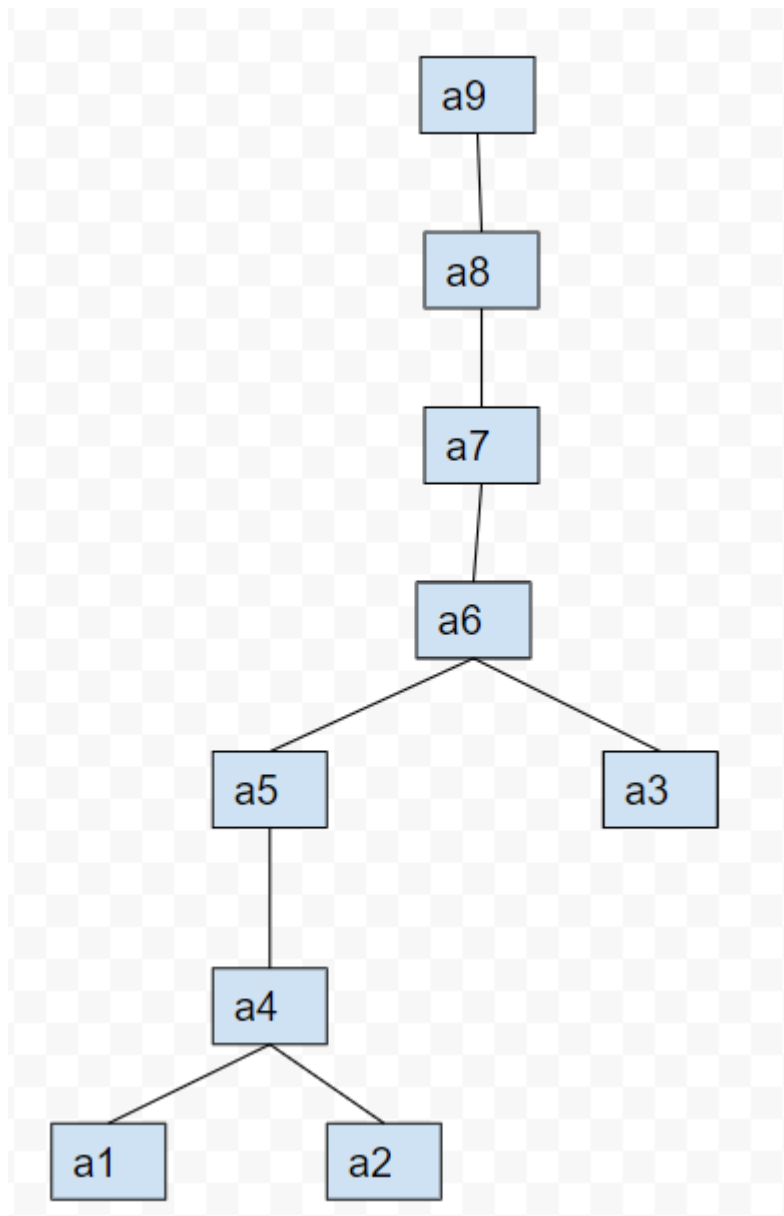
$$L = \frac{1}{2} (\sigma(wx) - y)^2$$

, the computational graph is (Only use simple operations , see, i.e., slides 7.3):









▼ Hide Feedback

See next question.

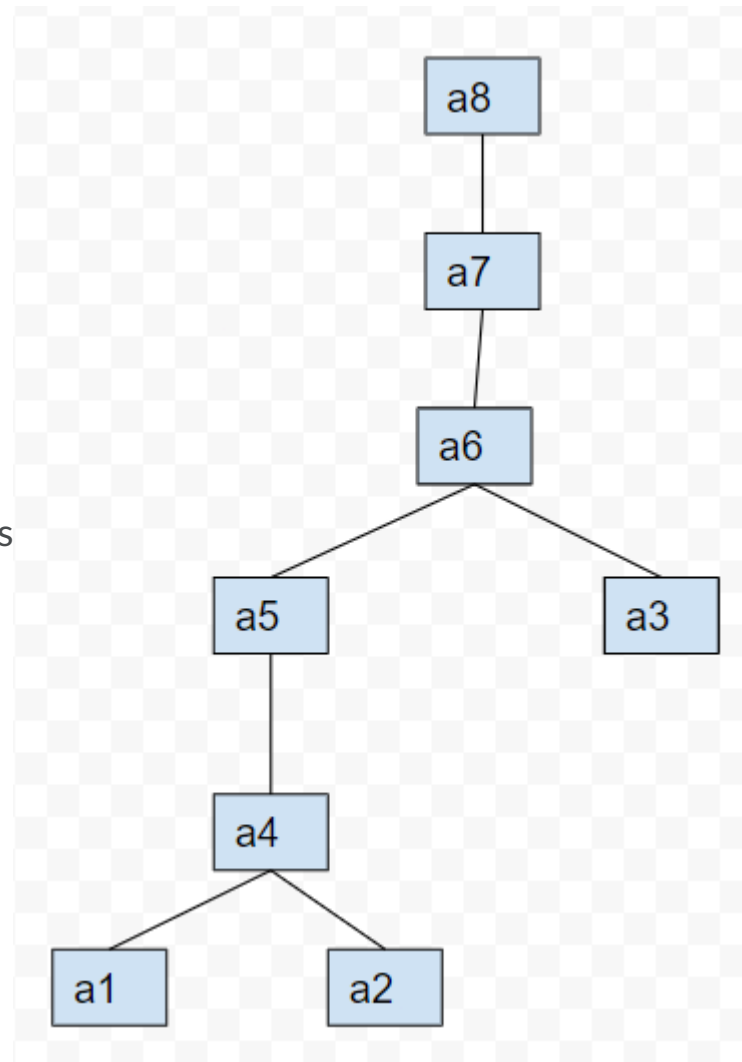
**Question 6****1 / 1 point**

[ Computational graph 2] Consider the model

$$L = \frac{1}{2}(\sigma(wx) - y)^2$$



, the computational graph is



where we take

$$a_8 = 0.5a_7$$

$$a_7 = (a_6)^2$$

$$a_6 = a_5 - a_3$$

$$a_5 = \sigma(a_4)$$

$$a_4 = a_1 * a_2$$

$$a_3 = y$$

$$a_2 = x$$

$$a_1 = w$$

Compute it in forward model where  $\dot{\square} = \frac{\partial \square}{\partial w}$ ,  $\dot{a}_7$  is

☐

$$\dot{a}_7 = 2 * a_6 * \dot{a}_6 = 2 * (\sigma(a_4) - a_4) * a_6 * (1 - a_6) * \dot{a}_4$$

☒

$$\dot{a}_7 = 2 * a_6 * \dot{a}_6 = 2 * (\sigma(a_4) - a_3) * a_5 * (1 - a_5) * \dot{a}_4$$

☐

$$\dot{a}_7 = 2 * a_6 * \dot{a}_6 = 2 * (\sigma(a_1 a_3) - a_3) * a_5 * (1 - a_5) * \dot{a}_4$$

☐

$$\dot{a}_7 = 2 * a_6 * \dot{a}_6 = 2 * (\sigma(a_4) - a_3) * a_5 * (1 - a_5) * \dot{a}_5$$

▼ Hide Feedback

$$\dot{a}_1 = 1$$

$$\dot{a}_2 = 0$$

$$\dot{a}_3 = 0$$

$$\dot{a}_4 = a_1 * \dot{a}_2 + a_2 * \dot{a}_1 = x$$

$$\dot{a}_5 = a_5 * (1 - a_5) * \dot{a}_4$$

$$\dot{a}_6 = \dot{a}_5 - \dot{a}_3 = a_5 * (1 - a_5) * \dot{a}_4$$

$$\dot{a}_7 = 2 * a_6 * \dot{a}_6 = 2 * (\sigma(a_4) - a_3) * a_5 * (1 - a_5) * \dot{a}_4$$

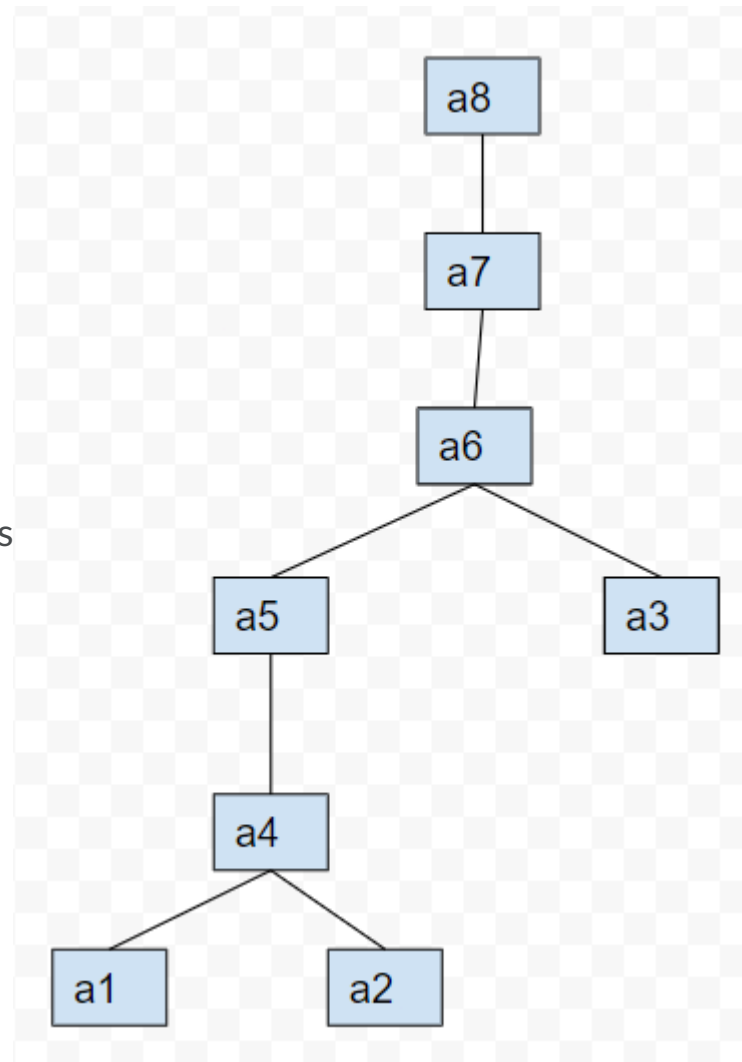
### Question 7

1 / 1 point

[ Computational graph 3][Right minus wrong ]Consider the model

$$L = \frac{1}{2}(\sigma(wx) - y)^2$$

, the computational graph is



where we take

$$a_8 = 0.5a_7$$

$$a_7 = (a_6)^2$$

$$a_6 = a_5 - a_3$$

$$a_5 = \sigma(a_4)$$

$$a_4 = a_1 * a_2$$

$$a_3 = y$$

$$a_2 = x$$

$$a_1 = w$$

Compute it in reverse model where  $\bar{\square} = \frac{\partial a_8}{\partial \square}$ ,  $\bar{a}_1$  is



$$\bar{a}_1 = x * \sigma(a_4) * (1 - \sigma(a_4)) * \bar{a}_5$$



$$\bar{a}_1 = x * \sigma(a_4) * (1 - \sigma(a_4)) * \bar{a}_3$$



$$\bar{a}_1 = x * \sigma(a_5) * (1 - \sigma(a_5)) * \bar{a}_5$$



$$\bar{a}_1 = x * \sigma(a_4) * (1 - \sigma(a_4)) * \bar{a}_7$$



$$\bar{a}_1 = -x * \sigma(a_4) * (1 - \sigma(a_4)) * \bar{a}_3$$



$$\bar{a}_1 = x * \sigma(a_4) * (1 - \sigma(a_4)) * \bar{a}_6$$

▼ Hide Feedback

$$\bar{a}_8 = 1$$

$$\bar{a}_7 = 0.5$$

$$\bar{a}_6 = a_6$$

$$\bar{a}_3 = -\bar{a}_6$$

$$\bar{a}_5 = \bar{a}_6 = a_6$$

$$\bar{a}_4 = \sigma(a_4) * (1 - \sigma(a_4)) * \bar{a}_5$$

$$\bar{a}_1 = x * \bar{a}_4 = x * \sigma(a_4) * (1 - \sigma(a_4)) * \bar{a}_5$$

$$= x * \sigma(a_4) * (1 - \sigma(a_4)) * a_6$$

$$= -x * \sigma(a_4) * (1 - \sigma(a_4)) * a_3$$

### Question 8

1 / 1 point

[CNN simple computation] Given input image is of size 227x227x3, The first convolutional layer has 96 kernels of size 11x11x3. The stride is 4 and padding is 3 (in each direction i.e., 230x230x3 in total), the output image of this first bank of convolutional layers is:

☐ 57x57x96

☐ 55x55x96☒ 56x56x96☐ 55x55x97

▼ Hide Feedback

the total padding is  $2 \times 3 = 6$  thus the dimension is

$$\left\lfloor \frac{227 + 2 * 3 - 11}{4} + 1 \right\rfloor = 56$$

and the 96 is number of kernels.

### Question 9



0 / 1 point

[CNN advanced computation]

You are training two neural network models on a binary image classification dataset with 10x10 greyscale images.

- The first model is a feedforward neural network with two hidden layers. The first hidden layer has a dimension of 50 and the second has a dimension of 10.
- The second model is a convolutional neural network. It has one convolutional layer with 2x2 filters. There are 75 different convolutional filters applied in this layer (i.e., the "depth" or "number of channels" in this layer is 75) with stride 1 and no zero padding.

Which model has more parameters?

-  ☒ The feedforward neural network.
-  ☐ The convolutional neural network.
- ☐ They have the same number of parameters.

▼ Hide Feedback

For the FFNN, we have that the first layer has  $100 \times 50 + 1$  parameters (i.e., the input is the flattened image of dimension  $10 \times 10 = 100$ , the hidden dimension is 50, and there is the bias term). The second layer has  $50 \times 10 + 1$  parameters (i.e., 50 dimensional input and 10 dimensional output, with the bias term). And finally, the binary classification output layer has  $10 \times 1 + 1$  parameters (i.e., 10 dimensional input and 1 dimensional output, plus the bias term). Thus, in total, the FFNN has  $100 \times 50 + 1 + 50 \times 10 + 1 + 10 + 1 = 5513$  parameters.

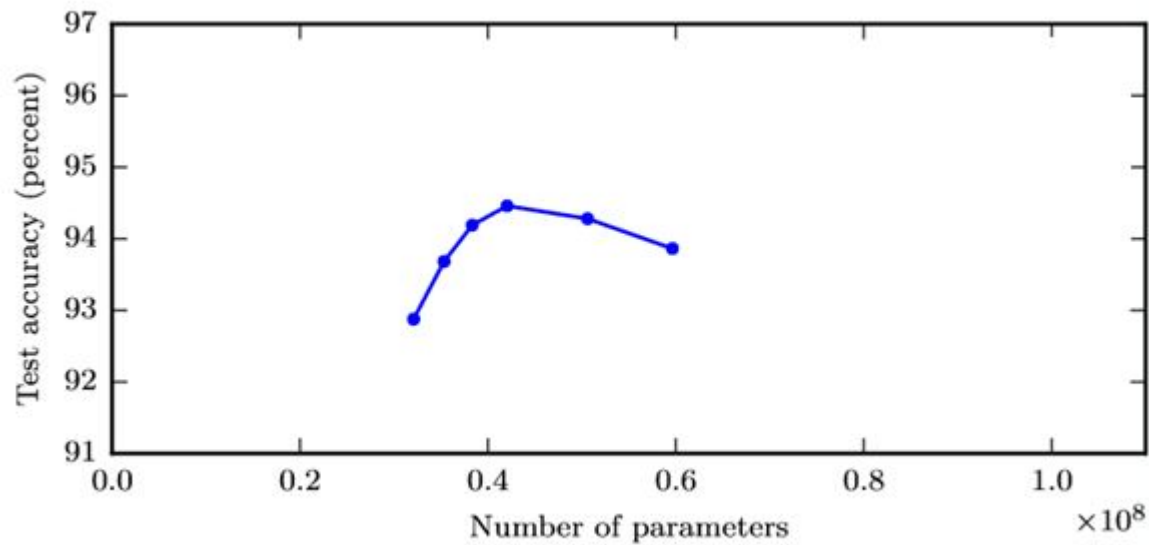
For the CNN, we have 75  $2 \times 2$  convolutional filters, each with  $2 \times 2 + 1 = 5$  parameters (including the bias term). Thus in total there are  $75 \times 5 = 375$  parameters in the first convolutional layer. Since there is no zero padding, the output of this first layer is dimension  $9 \times 9 \times 75 = 6075$  (i.e., there are 75 different channels since we are applying 75 different convolutions with  $(10-2)/1+1$  size for output). Thus, in total the CNN has  $6075 + 375 = 6450$  parameters, which is more than the FFNN.

## Question 10

1 / 1 point

The below graph shows the accuracy of a trained 3-layer convolutional neural network vs the number of parameters (i.e. number of feature kernels).





The trend suggests that as you increase the width of a neural network, the accuracy increases till a certain threshold value, and then starts decreasing.

What could be the possible reason for this decrease?

- ☐ Even if number of kernels increase, only few of them are used for prediction.
- ☐ As the number of kernels increase, the predictive power of neural network decrease.
- ☒ As the number of kernels increase, they start to correlate with each other which in turn helps overfitting.
- ☐ None of these

▼ Hide Feedback

The possible reason could be kernel correlation when the number of kernels increases.

**Question 11****1 / 1 point**

[Convolution computation example] For the following input and kernel in convolutional layer, if we add 0-padding with size 1 to the input, what is the output image.

input image:

0	1	2
3	4	5
6	7	8

kernel,

Kernel

0	1
2	3



43	25
37	19



19	25
37	43



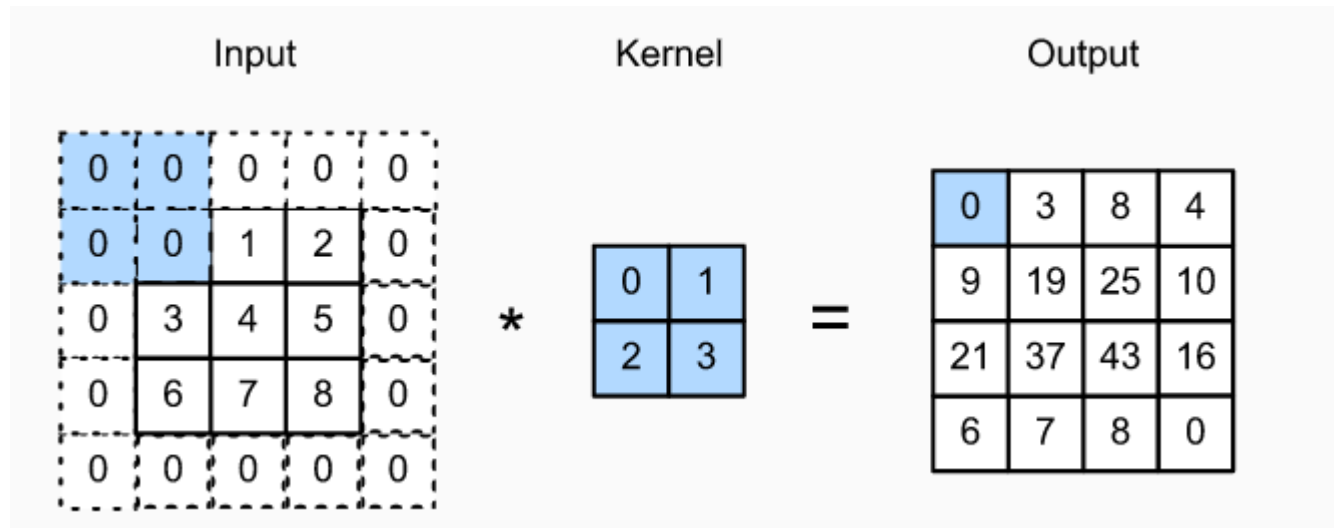
19	25	10
37	43	16
7	8	0



0	3	8	4
9	19	25	10
21	37	43	16
6	7	8	0



Hide Feedback



Attempt Score: 10 / 11 - 90.91 %

Overall Grade (highest attempt): 10 / 11 - 90.91 %

Done

# Quiz Submissions - Quiz 8



## Attempt 1

## Submission View

### Question 1

1 / 1 point

[K-means]

Suppose you are running simple K-means with  $K=2$  on the following set of two-dimensional data points:

- $\mathbf{x}_1 = [0, 2]$
- $\mathbf{x}_2 = [2, 0]$
- $\mathbf{x}_3 = [-1, -1]$
- $\mathbf{x}_4 = [-1, 0]$
- $\mathbf{x}_5 = [2, 2]$

The initial guesses for the 2 centroids are:

- $\mu_1 = [-2, -2]$

•

$$\mu_2 = [2, 2]$$

What would be the new centroids after one iteration of the expectation maximization algorithm (i.e., after one update to the estimated centroids)? **Assume that you are using an L1 distance function.**



$$\mu_1 = [1.33, 1.33], \mu_2 = [-1, -0.5]$$



$$\mu_1 = [0.33, 0.33], \mu_2 = [-0.5, -0.5]$$



$$\mu_1 = [0, 1.33], \mu_2 = [-1, -0.5]$$



$$\mu_1 = [0.33, 0.33], \mu_2 = [-1.5, -1.5]$$

▼ [Hide Feedback](#)

In the first iteration, the first, second, and fifth points are assigned to the first cluster, and the third and fourth points are assigned to the second cluster.

Using these assignments we can recompute the means as:

$$\mu_1 = \frac{1}{3}([0, 2] + [2, 0] + [2, 2]) = [1.33, 1.33]$$

$$\mu_2 = \frac{1}{2}([-1, -1] + [-1, 0]) = [-1, -0.5]$$

**Question 2****1 / 1 point**

[K-means, Unsupervised learning, Right minus wrong]

For which of the following problems, K-means clustering may be a suitable algorithm?

- ✓ ☒ Given a database of user information, users are automatically grouped into different market groups.
- ✓ ☒ Based the sales data in the supermarket, find out which products can form a combination (for example, often buy together), so they should be placed on the same shelf.
- ✓ ☐ Based on historical weather records, predict the rainfall tomorrow.
- ✓ ☐ Given the sales data in the supermarket, estimate the future sales of these products.
- ✓ ☐ Given many emails, determine whether they are spam or non-spam.

**Question 3****1 / 1 point**

[K-means, Right minus wrong ] Which of them are correct?

- ✓ ☒ If we are concerned that K-means would get stuck in local optimal, one way to solve this problem is to try to use multiple random initializations.

- ✓ ☐ Since K-Means is an unsupervised learning algorithm, it cannot overfit the data, so it is better to have more clusters as possible.
- ✓ ☐ No matter how we choose the initialization of the cluster center, K-means will always give the same result.
- ✓ ☐ Once an example is assigned to a specific cluster center, it will never be reassigned to a different cluster center.
- ✓ ☒ When we initialize the cluster centers, it's better to choose the centers whose distance between of them are further.

**Question 4****1 / 1 point**

Can SVD and PCA produce the same projection result?

- ✓ ☒ True
- ☐ False

▼ [Hide Feedback](#)

Yes. When the data has a zero mean vector, otherwise you have to center the data first before taking SVD.

**Question 5****1 / 1 point**



[K-means] Let a configuration of the k means algorithm correspond to the k way partition (on the set of instances to be clustered) generated by the clustering at the end of each iteration. Is it possible for the k-means algorithm to revisit a configuration?

☐ Possible

✓ ☒ Impossible

▼ [Hide Feedback](#)

Since the k means algorithm converges if the k way partition does not change in successive iterations, thus the k way partition has to change after every iteration. As the mean squared error monotonically decreases it is thus impossible to revisit a configuration. Thus eventually the k means algorithm will run out of configurations, and converge.

### Question 6

1 / 1 point

[PCA 1] Given 3 data points in 2-d space, (1,1), (2,2), (3,3), what is the first principle component?

✓ ☒

$(1/\sqrt{2}, 1/\sqrt{2})$

☐

$(1, 1)$

☐

$(-1, -1)$

 $(2, 2)$ 

▼ Hide Feedback

The direction is  $(1,1)$  and normalized to

$$(1/\sqrt{2}, 1/\sqrt{2})$$

.

### Question 7

1 / 1 point

[PCA 2] For same data,  $(1,1), (2,2), (3,3)$ . If we want to project the original data points into 1-d space by principle component you choose, what is the variance of the projected data?

☐ 0

✓ ☒ 4/3

☐ 2/3

☐ 1

▼ Hide Feedback

The data becomes  $-\sqrt{2}, 0, \sqrt{2}$  (after centered) and we can easily get the mean is  $2\sqrt{2}$  with variance  $4/3$ .

**Question 8****1 / 1 point**

[PCA 3] For the projected data in the previous question, now if we represent them in the original 2-d space, what is the reconstruction error?

✓ ☒ 0

☐ 2/3

☐ 4/3

☐ 1

**Question 9****1 / 1 point**

[PCA 4, advanced] You are given a design matrix

$$X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$$

, Compute the covariance matrix for the sample points. (HINT: X is not centered)



$$\begin{bmatrix} 82 & -80 \\ -80 & 82 \end{bmatrix}$$



$$\begin{bmatrix} 98 & -72 \\ -72 & 86 \end{bmatrix}$$



$$\begin{bmatrix} 41 & -40 & -41 & 40 \\ -40 & 41 & 40 & -41 \\ -41 & 40 & 41 & -40 \\ 40 & -41 & -40 & 41 \end{bmatrix}$$



$$\begin{bmatrix} 36 & 16 \\ 9 & 25 \\ 4 & 36 \\ 49 & 9 \end{bmatrix}$$

▼ Hide Feedback

The mean of first column is 2 and the second row is 1, Thus the centered data matrix is

$$Y = \begin{bmatrix} 4 & -5 \\ -5 & 4 \\ -4 & 5 \\ 5 & -4 \end{bmatrix}$$

The covariance matrix is

$$Y^T Y = \begin{bmatrix} 82 & -80 \\ -80 & 82 \end{bmatrix}$$

### Question 10

1 / 1 point

[PCA 5, advanced] You are given a design matrix

$$X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$$

, Compute the corresponding eigenvalues of covariance matrix for the sample points. ( If you graph the points, you can probably guess the eigenvectors )

✓ ☒ 2, 162

☐ 19.75, 164.2

☐ 4, 82

▼ [Hide Feedback](#)

We can easily get the eigenvectors

$$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

with eigenvalue 2

and

$$\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

with eigenvalue 162

### Question 11

1 / 1 point

[PCA 6, advanced] You are given a design matrix

$$X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$$

.

Suppose we use PCA to project the sample points onto a one-dimensional space. What one-dimensional subspace are we projecting onto? (Note: eigenvalue is the square of the corresponding singular value )

☐

$$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

☒

$$\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

☐

$$\begin{bmatrix} 1/\sqrt{2} & 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

▼ [Hide Feedback](#)

We would choose the one with larger singular value/eigenvalue, and the space is

$$\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

### Question 12

1 / 1 point

[PCA 7, advanced] You are given a design matrix

$$X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$$

.

Suppose we use PCA to project the sample points onto a one-dimensional space. What  $(6, -4)$  is projected to? (Hint: Be careful centered or not?)



$$\frac{10}{\sqrt{2}}$$





$$\frac{9}{\sqrt{2}}$$



$$\frac{1}{\sqrt{2}}$$



$$\frac{2}{\sqrt{2}}$$

▼ Hide Feedback

$$\begin{bmatrix} 6 & -4 \end{bmatrix} * \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} = \frac{10}{\sqrt{2}}$$

This is projection operation. Here we do not need to centre it again.

### Question 13

1 / 1 point

Instead of using all features, reduce the training data down to k-dimensions with PCA, and use the PCA projections as the only features. If we increase k, generally, the variance of the data would:

- ✓ ☒ larger
- ☐ smaller
- ☐ both are possible

**Question 14****1 / 1 point**

[Right minus wrong] Both PCA and Lasso can be used for feature selection. Which of the following statements are true?

- ✓ ☒ Lasso selects a subset (not necessarily a strict subset) of the original features
- ✓ ☐ PCA and Lasso both allow you to specify how many features are chosen
- ✓ ☒ PCA produces features that are linear combinations of the original features

**Question 15****1 / 1 point**

[Right minus Wrong ] Suppose we are given data comprising points of several classes. Each class has a different probability distribution from which the sample points are drawn. We do not have the class labels. We use k-means clustering to try to guess the classes. Which of the following circumstances would undermine its effectiveness?

- ✓ ☐ Some of the classes are not normally distributed
- ✓ ☐ The variance of each distribution is small in all directions
- ✓ ☒ Each class has the same mean

✓ ☒ You choose  $k = n$ , the number of sample points

---

**Attempt Score:** ☐ 15 / 15 - 100 %

**Overall Grade (highest attempt):** ☐ 15 / 15 - 100 %

Done