CIS 4190/5190: Applied Machine Learning Fall 2024

Midterm Exam I

Exam Data: October 16, 2024

Name: _____

Penn ID number: _____

Section number (4190 or 5190): _____

Instructions

- Write your name, Penn ID number, and section number on this page.
- Write your answer directly on the exam paper. Consider using a pencil and eraser, which will allow you to easily adjust your answers if needed.
- No digital devices are allowed during the exam (e.g., laptops, tablets, phones, watches). Please ensure that all your devices are turned off or set to silent mode.
- This is 75-minute exam, containing 15 questions with 50 points total.
- Each point should take approximately 1.5 minutes; if you find yourself spending too much time on one problem, move on and come back to it.
- At the end of 75 minutes, you will put down your pens and submit your exam.

Good luck!

1. (1 pt) Which of the following options is the correct order of steps involved in training a neural network? _____.

A. backward pass → compute loss → forward pass

B. forward pass → backward pass → compute loss

C. compute loss → forward pass → backward pass

D. forward pass → compute loss → backward pass

D

2. (1 pt) Which of the following neural networks is the most expressive, i.e., it can represent the most possible functions? As usual, W, W_i, b, b_i represent learnable weights. _____.

A. $\text{ReLU}(Wx + b)$

B. $W_2(\text{ReLU}(W_1x + b_1)) + b_2$

C. $W_2(W_1x + b_1) + b_2$

D. $W_1x + W_2x + W_1W_2x + b_1 + b_2 + b_1b_2$

B

3. (1 pt) Assume we are using a 3-way dataset split (`train`, `val`, and `test`) to train a machine learning model on `train` and tune its hyperparameters on `val`. Suppose we compute the accuracy of the trained network on each split, $\text{Acc}_{\texttt{train}}$, $\text{Acc}_{\texttt{val}}$, $\text{Acc}_{\texttt{test}}$, respectively. Which of the following is most likely to be true? _____.

A. $\text{Acc}_{\texttt{train}} > \text{Acc}_{\texttt{val}} > \text{Acc}_{\texttt{test}}$

B. $\text{Acc}_{\texttt{train}} < \text{Acc}_{\texttt{val}} < \text{Acc}_{\texttt{test}}$

C. $\text{Acc}_{\texttt{train}} = \text{Acc}_{\texttt{val}} = \text{Acc}_{\texttt{test}}$

D. $\text{Acc}_{\texttt{train}} > \text{Acc}_{\texttt{val}} = \text{Acc}_{\texttt{test}}$

A

4. (1 pt) For a k-nearest neighbor (KNN) classifier with Euclidean distance, would its decision boundary change if we multiply all features of each sample by 0.5? _____.

A. Yes

B. No

B

5. (1 pt) In a k-nearest neighbor (KNN) model, what is the effect of choosing a very small value for k, such as k=1? Select all options that apply. _____.

A. It leads to a more robust model that handles noise better.

B. It may result in overfitting, as the model can be sensitive to noise and outliers.

C. It reduces the capacity of the model.

D. It makes the model immune to class imbalance.

B

6. (2 pts) You fit a logistic regression classifier to a dataset using the plain maximum likelihood objective and batch gradient descent (i.e., each gradient update is computed over the entire dataset) with learning rate α. You track changes to the parameters $\|\beta_t - \beta_{t-1}\|_2^2$ over iterations t. You observe that the parameters continue to change indefinitely without convergence. Which of the following would you suspect? Select all options that apply. _____.

A. The learning rate is too small.

B. The learning rate is too large.

C. The training dataset cannot be separated by any function within your function class.

D. The training dataset is perfectly separable within your function class.

B D

7. (2 pts) Alice is trying to grow a decision tree using the gain ratio, which is the information gain divided by split information. For a particular split that she is considering, she finds, to her initial horror that its split information is 0. After taking a moment to think it through, she smiles and declares that this is not a problem. She is right. Can you explain why? (1-2 sentences)
This feature has the same value for all the data points, so it would anyway have zero information gain / be useless for the decision tree.

8. (3 pts) You are training a logistic regression model on a binary classification task, where the features are *Age* and *Income*. The following is the logistic regression equation:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{Age} + \beta_2 \cdot \text{Income})}}$$

After training, you obtain the following coefficients: $\beta_0 = -3$, $\beta_1 = 0.05$, and $\beta_2 = 0.001$. Which of the following statements is **correct** about the relationship between *Age*, *Income*, and the predicted probability? Select all that apply. _____.

A. An increase of 20 years in *Age* results in a greater change in the predicted probability than an increase of \$10,000 in *Income*.

B. The positive sign of β_1 and β_2 indicates that increasing *Age* and *Income* both increase the predicted probability of the outcome being $y = 1$.

C. The effect on the predicted probability of an increase by δ in *Income* is negligible compared to the effect of an increase by the same δ in *Age* because β_1 is much larger than β_2.

D. The intercept $\beta_0 = -3$ suggests that when both *Age* and *Income* are zero, the predicted probability of $y = 1$ is $\approx$50%.

B C

9. (3 pts) Consider a linear regression model with d features (no regularization). Describe how the variance and bias of the model would change (i.e., increase, decrease, or remain unchanged) for each of the following modifications.

A. (1 pt) Randomly drop 50% of the features, resulting in a model with $d/2$ input features. _____.
Answer: bias increases, variance decreases

B. (1 pt) Perform Principal Component Analysis (PCA) on the training dataset, and keep the top d' features (where $d' = d/2$), then use these features as input to the linear regression model. _____.
Answer: bias increases, variance decreases

C. (1 pt) Concatenate the top d' features from PCA with the original d-dimensional features, resulting in a model with $d + d'$ input features. (Ignore any optimization difficulties that arise from this). _____.
Answer: bias and variance remain unchanged

10. (6 pts) Suppose that your function class for a regression problem is $\hat{y} = f_\beta(x) = \beta_0$. You want to minimize the mean squared error. The dataset is $\mathcal{D} = \{(x_i, y_i)\}_{i=1,2,\ldots,N}$.

A. (2 pts) Can you derive an expression for the constant β^* that minimizes the MSE on the training dataset $\frac{1}{N}\sum_i(\hat{y}_i - y_i)^2$? Show your steps.

The function class for the regression problem is given by:

$$\hat{y} = f_\beta(x) = \beta_0$$

The mean squared error (MSE) is defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

Since $\hat{y}_i = \beta_0$ for all i, we can substitute:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\beta_0 - y_i)^2$$

To minimize this expression with respect to β_0, we take the derivative of MSE with respect to β_0:

$$\frac{d}{d\beta_0}\left(\frac{1}{N}\sum_{i=1}^{N}(\beta_0 - y_i)^2\right) = \frac{2}{N}\sum_{i=1}^{N}(\beta_0 - y_i)$$

Setting this derivative equal to zero:

$$\frac{2}{N}\sum_{i=1}^{N}(\beta_0 - y_i) = 0$$

This simplifies to:

$$\sum_{i=1}^{N}\beta_0 = \sum_{i=1}^{N} y_i$$

$$\beta_0 = \frac{1}{N}\sum_{i=1}^{N} y_i = \bar{y}$$

B. (2 pts) You fit a 3-layer neural network to this data by minimizing its MSE. The final training MSE loss is $1.25 \times$ variance $(\{y_i\}_{i=1,2,\ldots,N})$. Can you conclusively say whether this is a "good" fit to the training data? Explain in 1-2 sentences.

Setting β to the label mean as derived in 10A yields MSE = label variance (by definition of variance). If MSE(neural net) > MSE(constant β), the neural net has not fit the data well.

C. (2 pts) For the same function class as in part (A), can you derive an expression for the constant β^+ that minimizes the mean absolute error $\frac{1}{N}\sum_i |\hat{y}_i - y_i|$? Show your steps.

The objective function $\mathcal{L}$ is given by:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}|\beta - y_i|$$

Where:

$$|\beta - y_i| = \begin{cases}(\beta - y_i), & \text{if } y_i < \beta \\ (y_i - \beta), & \text{if } y_i \geq \beta\end{cases}$$

So, the objective function can be rewritten as:

$$\mathcal{L} = \frac{1}{N}\left(\sum_{i:y_i<\beta}(\beta - y_i) + \sum_{i:y_i\geq\beta}(y_i - \beta)\right)$$

Now, to minimize $\mathcal{L}$, take the derivative of $\mathcal{L}$ with respect to β:

$$\frac{\partial\mathcal{L}}{\partial\beta} = \sum_{i:y_i<\beta} 1 + \sum_{i:y_i\geq\beta}(-1)$$

This simplifies to:

$$\frac{\partial \mathcal{L}}{\partial \beta} = 1 \cdot (\text{number of samples where } y_i < \beta) - 1 \cdot (\text{number of samples where } y_i \geq \beta)$$

Setting this derivative to zero:

$$(\text{number of samples where } y_i < \beta) = (\text{number of samples where } y_i \geq \beta)$$

Thus, β must be such that half the samples are less than β and half are greater than or equal to β. This means that β must be the median of the $\{y_i\}$:

$$\beta = \text{median}(y_1, y_2, \ldots, y_N)$$

11. (4 pts) While growing a decision tree, you arrive at a set of samples at a particular node, as shown in Figure 1 on the right. For the purpose of this question, you are considering a split of the data that is indicated by the vertical dashed line. Compute the gain ratio for this split. You may leave expressions in terms of logarithms if the logarithm is not a whole number. You are classifying circles versus stars.

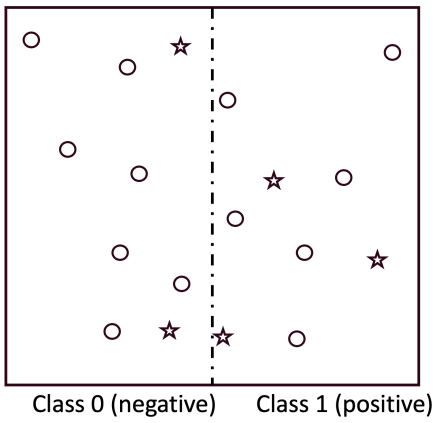


Figure 1: Decision tree.

The Gain Ratio is defined as:

$$\text{Gain Ratio} = \frac{\text{Info Gain}}{\text{Split Info}}$$

We are given the following information:
- Left split: 9 samples (7-, 2+) - Right split: 9 samples (6-, 3+) - Parent data: 18 samples (13-, 5+)

The information gain is calculated as:

$$\text{Info Gain} = H_y(\text{parent}) - \frac{9}{18} H_y(\text{left}) - \frac{9}{18} H_y(\text{right})$$

Where the entropies are:

$$H_y(\text{parent}) = -\frac{13}{18} \log_2 \frac{13}{18} - \frac{5}{18} \log_2 \frac{5}{18}$$

$$H_y(\text{left}) = -\frac{7}{9}\log_2\frac{7}{9} - \frac{2}{9}\log_2\frac{2}{9}$$

$$H_y(\text{right}) = -\frac{6}{9}\log_2\frac{6}{9} - \frac{3}{9}\log_2\frac{3}{9}$$

Substituting these into the Info Gain formula:

$$\text{Info Gain} = \left(-\frac{13}{18}\log_2\frac{13}{18} - \frac{5}{18}\log_2\frac{5}{18}\right) - \left(\frac{9}{18}\left(-\frac{7}{9}\log_2\frac{7}{9} - \frac{2}{9}\log_2\frac{2}{9}\right)\right) - \left(\frac{9}{18}\left(-\frac{6}{9}\log_2\frac{6}{9} - \frac{3}{9}\log_2\frac{3}{9}\right)\right)$$

Next, we calculate the Split Info:

$$\text{Split Info} = -\frac{9}{18}\log_2\frac{9}{18} - \frac{9}{18}\log_2\frac{9}{18}$$

$$\text{Split Info} = 1$$

Thus, the Gain Ratio is:

$$\text{Gain Ratio} = \frac{\text{Info Gain}}{\text{Split Info}} = \text{Info Gain}$$

12. (5 pts) Consider the dataset shown in the Figure 2 on the right, for classifying circles versus stars.

A. (3 pts) Draw the decision boundaries (within this box) for 1-nearest neighbors. Use Euclidean distances as measured on the image.

B. (2 pts) Suppose you are trying to "compress" this training dataset by removing as many data points as possible without altering the decision boundary within this box. Draw visible "x"s over each point that you would remove.

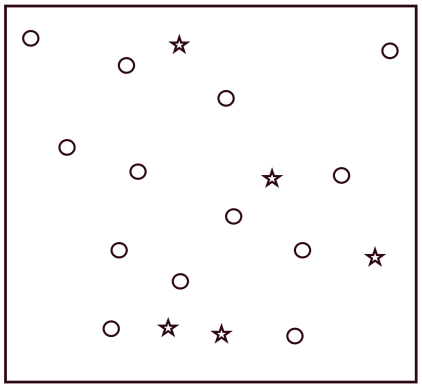


Figure 2: Nearest neighbors.

Answer (figure below):

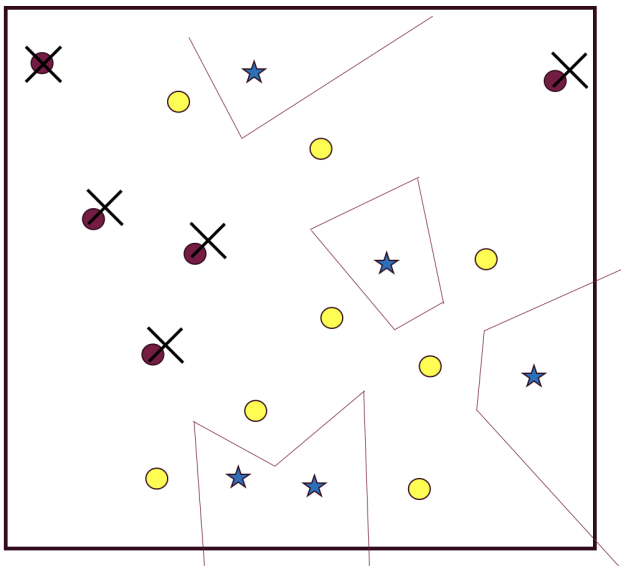


Figure 3: This is the solution graph for Q12

13. (8 pts) Consider a simple neural network model: $f_{\beta,\omega}(x) = \beta \cdot g(\omega_1 x_1 + \omega_2 x_2)$ where:

- $x = [x_1, x_2]$ is the input,
- β is a scalar weight, and ω_1, ω_2 are weights associated with the input features,
- g is the Leaky ReLU activation function, where:

$$g(z) = \begin{cases} z & \text{if } z \geq 0 \\ 0.1z & \text{if } z < 0 \end{cases}$$

- The loss function is $L(f_{\beta,\omega}(x), y) = \frac{1}{2}(f_{\beta,\omega}(x) - y)^2$, where y is the label.

A. (3 pts) Write down the gradients of the loss with respect to the model parameters: $\frac{\partial L}{\partial \beta}$, $\frac{\partial L}{\partial \omega_1}$, and $\frac{\partial L}{\partial \omega_2}$. You can leave your answer in terms g and g', where $g'(z) = \frac{\partial}{\partial z} g(z)$.

Answers:

$$\frac{\partial L}{\partial \beta} = (f_{\beta,\omega}(x) - y) \cdot \sigma(\omega_1 x_1 + \omega_2 x_2)$$

$$\frac{\partial L}{\partial \omega_1} = (f_{\beta,\omega}(x) - y) \cdot \beta \cdot \sigma'(\omega_1 x_1 + \omega_2 x_2) \cdot x_1$$

$$\frac{\partial L}{\partial \omega_2} = (f_{\beta,\omega}(x) - y) \cdot \beta \cdot \sigma'(\omega_1 x_1 + \omega_2 x_2) \cdot x_2$$

B. (2 pts) Find the derivative of the Leaky ReLU activation function $g(z)$ with respect to z, denoted as $g'(z)$. **Answers:**

$$\sigma'(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0.1 & \text{if } z < 0 \end{cases}$$

C. (3 pts) Consider a dataset with a single data point of $x = [5, 4]$ and $y = 0$. Calculate the gradients $\frac{\partial L}{\partial \beta}$, $\frac{\partial L}{\partial \omega_1}$, and $\frac{\partial L}{\partial \omega_2}$ for the following weights: $\beta = 1$, $\omega_1 = -2$, and $\omega_2 = -5$. **Answers:** Compute $z = \omega_1 x_1 + \omega_2 x_2 = (-2) \times 5 + (-5) \times 4 = -30$. Since $z = -30 < 0$, $\sigma(z) = 0.1 \times (-30) = -3$, and $\sigma'(z) = 0.1$. Therefore $f_{\beta,\omega}(x) = \beta \cdot \sigma(z) = 1 \times (-3) = -3$. Additionally, $f_{\beta,\omega}(x) - y = -3 - 0 = -3$. Gradient with respect to β:

$$\frac{\partial L}{\partial \beta} = (-3) \cdot (-3) = 9$$

Gradient with respect to ω_1:

$$\frac{\partial L}{\partial \omega_1} = (-3) \cdot 1 \cdot 0.1 \cdot 5 = -1.5$$

Gradient with respect to ω_2:

$$\frac{\partial L}{\partial \omega_2} = (-3) \cdot 1 \cdot 0.1 \cdot 4 = -1.2$$

Your answer to Question 13:

(Continue your answer to Question 13 on the next page)

Your answer to Question 13 (continued):

14. (6 pts) Consider the dataset shown in the Figure 3 for principal component analysis.

A. (2 pts) Draw the axis corresponding to the first principal component. Use the Figure 5 provided below to annotate your answer for this item (14.A).

B. (1 pts) Draw the axis corresponding to the second principal component. Use the Figure 5 provided below to annotate your answer for this item (14.B).

C. (2 pts) Bob is a data scientist who has forgotten that he must first subtract the mean from all the data before applying PCA (Figure 4). Instead, he directly applies the following equation to calculate the covariance matrix in the PCA procedure.

$$C = \mathbb{E}\left[xx^\top\right] = \mathbb{E}\begin{bmatrix} x_1x_1 & x_1x_2 \\ x_2x_1 & x_2x_2 \end{bmatrix}$$

Draw the axis corresponding to the first component with Bob's faulty PCA. Use the Figure 5 provided below to annotate your answer for this item (14.C).

D. (1 pts) Draw the axis corresponding to the second component with Bob's faulty PCA. Use the Figure 5 provided below to annotate your answer for this item (14.D).

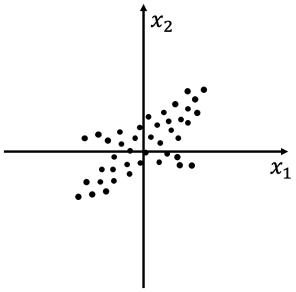


Figure 4: 2D dataset for PCA.

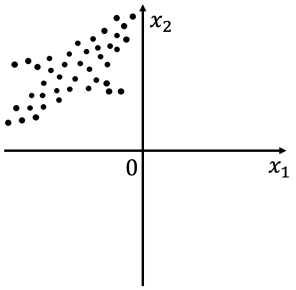
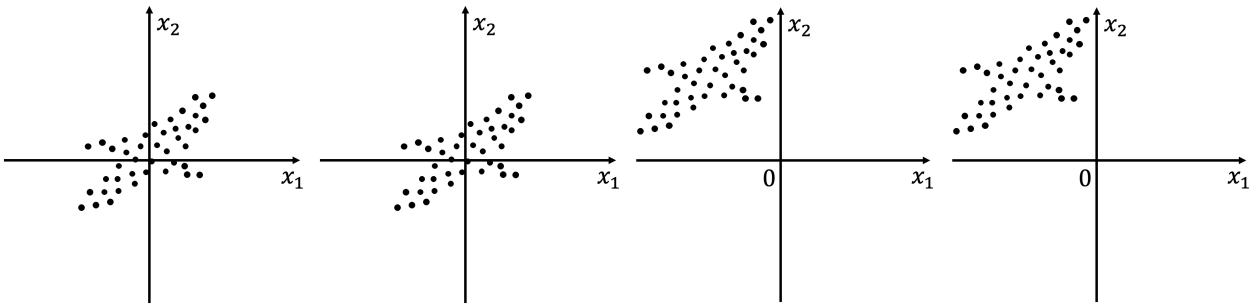


Figure 5: 2D dataset without mean subtraction.

Note the principal components' axes should pass through the origin of the coordinate system.

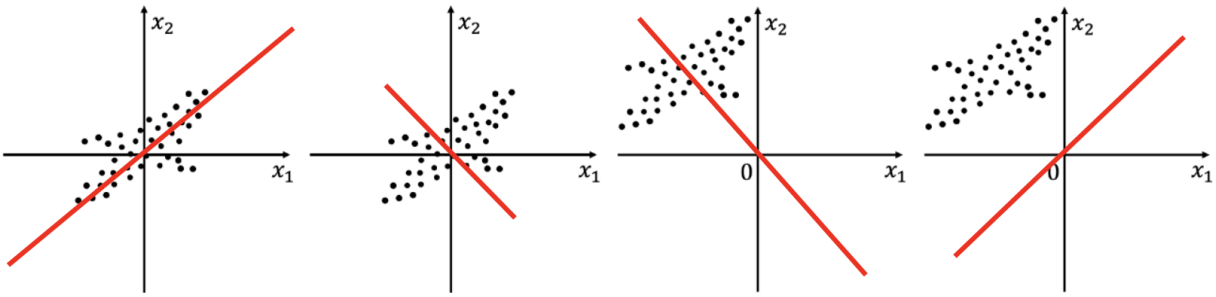


[Answer for 14.A] [Answer for 14.B] [Answer for 14.C] [Answer for 14.D]

Figure 6: Your answers for question 14.

ANSWER:

Orthogonal to the 1st one



[Answer for 14.A] [Answer for 14.B] [Answer for 14.C] [Answer for 14.D]

Figure 5: Your answers for question 14.

15. (6 pts) In class, we explored a dataset where traditional k-means clustering struggles—specifically when all the data points are roughly aligned along two distinct lines (as shown in Figure 6 on the right). Recognizing this limitation, Chris took up this challenge and developed an intriguing variation of k-means, which he calls k-lines clustering. Rather than clustering around k centroids, the k-lines algorithm seeks to identify k lines that better capture the underlying structure of the data. Here's how Chris's k-lines algorithm works:

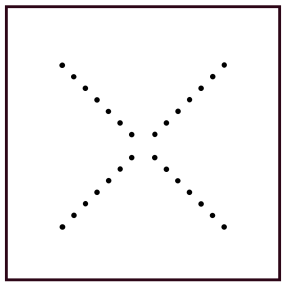


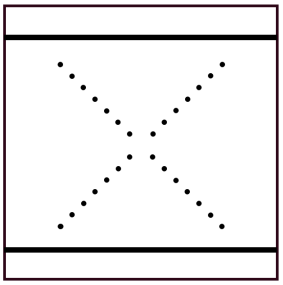
Figure 7: Clustering.

- Randomly initialize k lines.
- Assign each point to the line that it is closest to (i.e., perpendicular distance).
- Perform linear regression (minimizing the mean squared error) to update each line based on the points assigned to it.
- Repeat until convergence.

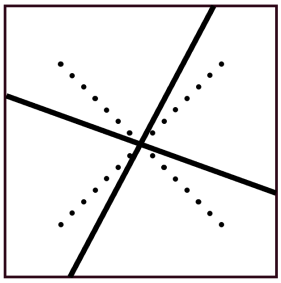
Similar to k-means, the converged results of this k-lines algorithm also depend on the initialization of the lines. Consider the following random initializations of $k = 2$ lines:

A. (3 pts) Two horizontal lines are selected as initialization as shown in Figure 15.A below. Please plot the lines after the first update in Figure 15.A. How many iterations does it take before k-lines converges? _____.

B. (3 pts) Two lines passing through the center of the data plane are selected as initialization as shown in Figure 15.B below. Please plot the lines after the first update in Figure 15.B. How many iterations does it take before k-lines converges? _____.



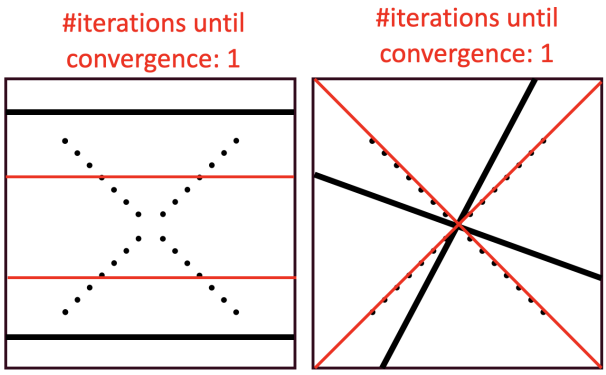
[Answer for 15.A]



[Answer for 15.B]

Figure 8: Your answers for question 15.

Answer:


#iterations until convergence: 1
#iterations until convergence: 1

Extra space

Extra space

Extra space