

Question 1:

'Statement 1] Linear regression estimator has the smallest variance among all unbiased estimators. Statement 2] The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are always non-negative.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: Linear regression estimator does not necessarily have the smallest variance among all unbiased estimators. False.

Statement 2: The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are always non-negative. True.

Final answer: C

=====

Answer Option 2:

Statement 1: Linear regression estimator does not necessarily have the smallest variance among all unbiased estimators. Therefore, Statement 1 is false.

Statement 2: The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are not always non-negative. Therefore, Statement 2 is false.

Final answer: D

=====

Answer Option 3:

Statement 1: Linear regression estimator does not always have the smallest variance among all unbiased estimators. So, the statement is False.

Statement 2: The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are always non-negative. This statement is True.

Final answer: C (False, True)

=====

Answer Option 4:

Statement 1: Linear regression estimator does not necessarily have the smallest variance among all unbiased estimators. So, Statement 1 is False.

Statement 2: The coefficients assigned to the classifiers assembled by AdaBoost are not always non-negative. AdaBoost can assign positive and negative weights to the classifiers. So, Statement 2 is False.

Final answer: D

=====

Answer Option 5:

Statement 1: The statement that "Linear regression estimator has the smallest variance among all unbiased estimators" is false. Linear regression is not guaranteed to have the smallest variance among all unbiased estimators.

Statement 2: The statement that "The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are always non-negative" is true. The coefficients assigned to the classifiers in AdaBoost are always non-negative.

Therefore, the correct answer is C. 'True, False'.

=====

Answer Option 6:

Statement 1 is false. While linear regression is an unbiased estimator, it does not necessarily have the smallest variance among all unbiased estimators. Statement 2 is true. The coefficients  $\alpha$  assigned to classifiers assembled by AdaBoost are always non-negative. Final answer: C

=====

Answer Option 7:

Statement 1: Linear regression estimator does not necessarily have the smallest variance among all unbiased estimators. Therefore, Statement 1 is False.

Statement 2: The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are always non-negative. Therefore, Statement 2 is True.

Final answer: C (False, True).

=====

Answer Option 8:

Statement 1: Linear regression estimator does not necessarily have the smallest variance among all unbiased estimators. False.

Statement 2: The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are always non-negative. True.

Final answer: C

=====

Answer Option 9:

Statement 1: "Linear regression estimator has the smallest variance among all unbiased estimators."

This statement is false. While linear regression is an unbiased estimator, it does not necessarily have the smallest variance among all unbiased estimators. There are cases where other estimators can have smaller variance.

Statement 2: "The coefficients  $\alpha$  assigned to the classifiers assembled by AdaBoost are always non-negative."

This statement is true. The AdaBoost algorithm assigns non-negative coefficients ( $\alpha$ ) to the classifiers, ensuring that each classifier contributes positively to the final ensemble prediction.

Final answer: C (True, False)

=====



Question 2:

'Statement 1| RoBERTa pretrains on a corpus that is approximate 10x larger than the corpus BERT pretrained on. Statement 2| ResNeXts in 2018 usually used tanh activation functions.', A. 'True, True', B.

'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. RoBERTa does pretrain on a corpus that is approximately 10 times larger than the corpus BERT pretrained on.

Statement 2 is false. ResNeXts in 2018 did not usually use tanh activation functions. They typically used ReLU activation functions.

Final answer: C

=====

Answer Option 2:

Statement 1: RoBERTa does pretrain on a corpus that is approximately 10x larger than the corpus BERT pretrained on. This statement is true.

Statement 2: ResNeXts in 2018 did not usually use tanh activation functions. They commonly used ReLU activation functions. This statement is false.

Final answer: C

=====

Answer Option 3:

Statement 1 is true. RoBERTa indeed pretrains on a corpus that is approximately 10 times larger than the corpus BERT pretrained on.

Statement 2 is false. ResNeXts in 2018 did not usually use tanh activation functions. Final Answer: C

=====

Answer Option 4:

Statement 1 is true. RoBERTa does pretrain on a corpus that is approximately 10 times larger than the corpus BERT pretrained on.

Statement 2 is false. ResNeXt models in 2018 did not usually use tanh activation functions. They commonly used ReLU (Rectified Linear Unit) activation functions.

Final answer: C

=====

Answer Option 5:

Statement 1: RoBERTa indeed pretrains on a corpus that is approximately 10 times larger than the corpus BERT was pretrained on. This statement is true.

Statement 2: ResNeXts in 2018 did not usually use tanh activation functions. This statement is false.

Therefore, the correct answer is C. 'True, False'.

=====

Answer Option 6:

Statement 1: RoBERTa does indeed pretrain on a corpus that is approximately 10x larger than the corpus BERT pretrained on. This statement is true.

Statement 2: ResNeXts, in 2018 or otherwise, do not usually use tanh activation functions. This statement is false.

Final answer: C

=====

Answer Option 7:

Statement 1: RoBERTa does pretrain on a corpus that is approximately 10 times larger than the corpus BERT is pretrained on. This statement is true.

Statement 2: ResNeXts in 2018 did not usually use tanh activation functions. This statement is false.

Final answer: C

=====

Answer Option 8:

Statement 1: RoBERTa does indeed pretrain on a corpus that is approximately 10 times larger than the corpus BERT pretrained on. So, Statement 1 is true.

Statement 2: ResNeXt models typically use ReLU activation functions rather than tanh activation functions. So, Statement 2 is false.

Final answer: C

=====

Answer Option 9:

Statement 1: RoBERTa does pretrain on a corpus that is approximately 10x larger than the corpus BERT pretrained on. This statement is true.

Statement 2: ResNeXts in 2018 did not typically use tanh activation functions. This statement is false.

Final answer: C

=====

Question 3:

'Statement 1| Support vector machines, like logistic regression models, give a probability distribution over the possible labels given an input example. Statement 2| We would expect the support vectors to remain the same in general as we move from a linear kernel to higher order polynomial kernels.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. Support vector machines do not inherently provide a probability distribution over possible labels like logistic regression models do.

Statement 2 is true. In general, as we move from a linear kernel to higher order polynomial kernels, the support vectors are expected to remain the same. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is false. Support vector machines (SVMs) do not inherently provide a probability distribution over the possible labels. However, methods like Platt scaling can be used to estimate probabilities from SVM outputs.

Statement 2 is false. As we move from a linear kernel to higher order polynomial kernels, the support vectors can change. Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is false. Support vector machines (SVMs) do not inherently provide probability distributions over the possible labels. However, probabilistic SVMs can be constructed by using techniques like Platt scaling or isotonic regression.

Statement 2 is false. As we move from a linear kernel to higher order polynomial kernels, the support vectors can change. The decision boundaries and the support vectors depend on the choice of kernel and its parameters. Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is false. Support vector machines (SVMs) do not directly provide a probability distribution over the possible labels. While there are techniques such as Platt scaling that can be used to estimate probabilities from SVM outputs, it is not inherent to the model.

Statement 2 is false. Moving from a linear kernel to higher order polynomial kernels can cause the support vectors to change. The decision boundary and support vectors depend on the kernel used and the complexity of the data. Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is false. Support vector machines (SVMs) do not inherently provide a probability distribution over the possible labels. However, there are techniques like Platt scaling that can be used to obtain probability estimates from SVM models.

Statement 2 is false. As we move from a linear kernel to higher order polynomial kernels, it is expected that the support vectors may change. The decision boundary and the importance of specific training examples may be different with different kernels.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is False. Unlike logistic regression models, support vector machines (SVMs) do not naturally provide a probability distribution over the possible labels. However, probabilities can be estimated using techniques such as Platt scaling.

Statement 2 is True. In general, the support vectors in SVMs tend to remain the same as we move from a linear kernel to higher order polynomial kernels. Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1 is False. Support vector machines (SVMs) do not inherently give a probability distribution over the possible labels. However, there are methods such as Platt scaling or using a sigmoid function that can be applied to SVMs to obtain probability estimates.

Statement 2 is True. In general, as we move from a linear kernel to higher order polynomial kernels, the support vectors are expected to remain the same. The kernel trick allows us to implicitly map the data into a higher-dimensional feature space without explicitly computing all the features. Final answer: C.

\*\*\*\*\*

Answer Option 8:

Statement 1 is False. Support vector machines (SVMs) do not inherently provide a probability distribution over possible labels like logistic regression models. However, probabilities can be estimated using techniques such as Platt scaling or using an SVM as a component in a larger probabilistic model.

Statement 2 is True. In general, as we move from a linear kernel to higher-order polynomial kernels, the support vectors are expected to remain the same. The kernel trick allows SVMs to operate in a high-dimensional feature space without explicitly computing the transformations, so the support vectors are determined by the geometry of the data rather than the specific kernel used.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: Support vector machines (SVM) do not inherently provide a probability distribution over the possible labels given an input example. However, there are techniques such as Platt scaling that can be applied to SVM to estimate class probabilities. So, Statement 1 is False.

Statement 2: As we move from a linear kernel to higher order polynomial kernels in an SVM, we would not necessarily expect the support vectors to remain the same. The kernel function affects the decision boundary and how the SVM identifies support vectors. So, Statement 2 is False.

Final answer: C

\*\*\*\*\*

Question 4:

'A machine learning problem involves four attributes plus a class. The attributes have 3, 2, 2, and 2 possible values each. The class has 3 possible values. How many maximum possible different examples are there?', A. '12', B. '24', C. '48', D. '72'

=====

Answer Option 1:

To calculate the maximum possible different examples, we need to multiply the number of possible values for each attribute together.

The first attribute has 3 possible values, the second attribute has 2 possible values, the third attribute has 2 possible values, and the fourth attribute has 2 possible values. The class has 3 possible values.

So, the maximum possible different examples would be:  $3 * 2 * 2 * 2 * 3 = 72$ .

Final answer: D

\*\*\*\*\*

Answer Option 2:

To calculate the maximum possible different examples in a machine learning problem, we multiply the number of possible values for each attribute, including the class.

In this case, the first attribute has 3 possible values, the second attribute has 2 possible values, the third attribute has 2 possible values, the fourth attribute has 2 possible values, and the class has 3 possible values.

Therefore, the total number of maximum possible different examples is  $3 * 2 * 2 * 2 * 3 = 72$ .

Final answer: D

\*\*\*\*\*

Answer Option 3:

To determine the maximum possible different examples, we need to multiply the number of possible values for each attribute together, and then multiply that by the number of possible values for the class.

Number of possible examples = (number of values for attribute 1) \* (number of values for attribute 2) \* (number of values for attribute 3) \* (number of values for attribute 4) \* (number of values for the class)

Number of possible examples =  $3 * 2 * 2 * 2 * 3 = 72$

Therefore, the maximum possible different examples is 72. Final answer: D

\*\*\*\*\*

Answer Option 4:

To calculate the maximum possible different examples, we multiply the number of possible values for each attribute together, and then multiply that by the number of possible values for the class. In this case, we have 3 possible values for the first attribute, 2 possible values for the second, 2 possible values for the third, and 2 possible values for the fourth, and 3 possible values for the class.

Therefore, the total number of maximum possible different examples is:  $3 * 2 * 2 * 2 * 3 = 72$ .

Final answer: D

\*\*\*\*\*

Answer Option 5:

To calculate the maximum possible different examples, multiply the number of possible values for each attribute and the number of possible values for the class.

Number of possible examples = (Number of possible values for attribute 1) \* (Number of possible values for attribute 2) \* (Number of possible values for attribute 3) \* (Number of possible values for attribute 4) \* (Number of possible values for the class)

Number of possible examples =  $3 * 2 * 2 * 2 * 3 = 72$

Final answer: D

\*\*\*\*\*

Answer Option 6:

To calculate the maximum possible different examples, we need to multiply the number of possible values for each attribute and the class.

The first attribute has 3 possible values, the second attribute has 2 possible values, the third attribute has 2 possible values, the fourth attribute has 2 possible values, and the class has 3 possible values.

Therefore, the total number of maximum possible different examples is:  $3 * 2 * 2 * 2 * 3 = 72$ .

Final answer: D

\*\*\*\*\*

Answer Option 7:

To calculate the maximum possible different examples, we need to multiply the number of possible values for each attribute and the number of possible values for the class.

Number of possible examples = (number of values for attribute 1) \* (number of values for attribute 2) \* (number of values for attribute 3) \* (number of values for attribute 4) \* (number of values for the class)

Number of possible examples =  $3 * 2 * 2 * 2 * 3 = 72$

Therefore, the maximum possible different examples is 72. Final answer: D

\*\*\*\*\*

Answer Option 8:

To calculate the maximum possible different examples, we multiply the number of possible values for each attribute and the class.

Number of possible values for the first attribute = 3

Number of possible values for the second attribute = 2

Number of possible values for the third attribute = 2

Number of possible values for the fourth attribute = 2

Number of possible values for the class = 3

Total number of maximum possible different examples =  $3 * 2 * 2 * 2 * 3 = 72$

Final Answer: D

\*\*\*\*\*

Answer Option 9:

To determine the maximum possible number of different examples, we need to multiply the number of possible values for each attribute together, and then multiply that by the number of possible values for the class.

Number of possible examples = (Number of values for attribute 1) \* (Number of values for attribute 2) \* (Number of values for attribute 3) \* (Number of values for attribute 4) \* (Number of values for class)

Number of possible examples =  $3 * 2 * 2 * 2 * 3 = 72$

Final answer: D

\*\*\*\*\*

Question 5:

'As of 2020, which architecture is best for classifying high-resolution images?', A. 'convolutional networks', B. 'graph networks', C. 'fully connected networks', D. 'RBF networks'

=====

Answer Option 1:

As of 2020, convolutional networks (A) are considered to be the best architecture for classifying high-resolution images. Convolutional neural networks (CNNs) have been proven to be very effective in image classification tasks due to their ability to capture spatial dependencies and extract meaningful features from the images. Final answer: A

=====

Answer Option 2:

As of 2020, convolutional networks (also known as convolutional neural networks or CNNs) have been widely recognized as the best architecture for classifying high-resolution images. Graph networks, fully connected networks, and RBF networks are not specifically designed for image classification tasks and may not perform as well. Final answer: A

=====

Answer Option 3:

As of 2020, convolutional neural networks (CNNs) have been proven to be highly effective for classifying high-resolution images. CNNs are specifically designed to capture spatial relationships in images, making them the preferred architecture for image classification tasks. Graph networks, fully connected networks, and RBF networks are not specifically tailored for image classification and may not perform as well. Final answer: A

=====

Answer Option 4:

As of 2020, convolutional neural networks (CNNs) are considered the best architecture for classifying high-resolution images. CNNs are specifically designed to efficiently extract features from images, making them highly effective for image classification tasks. Graph networks, fully connected networks, and radial basis function (RBF) networks are not typically used as the primary architecture for high-resolution image classification. Final answer: A

=====

Answer Option 5:

As of 2020, convolutional networks are considered the best architecture for classifying high-resolution images. Convolutional networks, or convolutional neural networks (CNNs), are specifically designed to process and analyze visual data such as images. Graph networks, fully connected networks, and radial basis function (RBF) networks are not specifically tailored for image classification tasks. Final answer:

A

=====

Answer Option 6:

As of 2020, the best architecture for classifying high-resolution images is convolutional neural networks (CNNs). CNNs have proven to be highly effective in image classification tasks due to their ability to learn hierarchical representations of images. Graph networks, fully connected networks, and radial basis function (RBF) networks are not specifically designed for image classification tasks and are not as commonly used for this purpose. Final answer: A

=====

Answer Option 7:

As of 2020, convolutional neural networks (CNNs) are considered the best architecture for classifying high-resolution images. CNNs are specifically designed to effectively extract features from images and have shown superior performance in image classification tasks. Final answer: A

=====

Answer Option 8:

As of 2020, convolutional neural networks (CNNs) are considered the best architecture for classifying high-resolution images. CNNs are specifically designed to handle image data and have achieved remarkable performance in image classification tasks. Graph networks, fully connected networks, and RBF networks are not commonly used for classifying high-resolution images. Final answer: A

=====

Answer Option 9:

As of 2020, convolutional neural networks (CNNs) are considered to be the best architecture for classifying high-resolution images. CNNs have shown remarkable performance in image classification tasks due to their ability to capture spatial hierarchies and local patterns in the data. Final answer: A

=====



Question 6:

'Statement 1] The log-likelihood of the data will always increase through successive iterations of the expectation maximization algorithm. Statement 2] One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. The log-likelihood of the data can either increase or decrease through successive iterations of the expectation-maximization algorithm.

Statement 2 is false. One advantage of Q-learning is that it does not require prior knowledge of how its actions affect the environment. Final answer: B

\*\*\*\*\*

Answer Option 2:

Statement 1 is not always true. The log-likelihood of the data may not always increase through successive iterations of the expectation-maximization algorithm. Statement 2 is also not true. Q-learning does not require prior knowledge of how actions affect the environment. Final answer: B

\*\*\*\*\*

Answer Option 3:

Statement 1: The log-likelihood of the data will always increase through successive iterations of the expectation maximization algorithm.

This statement is not necessarily true. The expectation maximization algorithm aims to maximize the log-likelihood of the observed data, but it is possible to get stuck in local optima or convergence issues where the log-likelihood does not always increase.

Statement 2: One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

This statement is false. Q-learning is a model-free reinforcement learning algorithm that does not require prior knowledge of the environment. It learns the optimal action-value function through an iterative process.

Final answer: B

\*\*\*\*\*

Answer Option 4:

Statement 1: The log-likelihood of the data will always increase through successive iterations of the expectation-maximization algorithm.

This statement is False. The expectation-maximization algorithm aims to maximize the log-likelihood of the data, but there is no guarantee that it will always increase through successive iterations. It may converge to a local maximum instead.

Statement 2: One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

This statement is False. Q-learning is a model-free reinforcement learning algorithm that does not require prior knowledge of how actions affect the environment. It learns from trial and error by interacting with the environment.

Final answer: B

\*\*\*\*\*

Answer Option 5:

Statement 1 is not necessarily true. The log-likelihood of the data may not always increase through successive iterations of the expectation-maximization algorithm. Therefore, statement 1 is false.

Statement 2 is also false. One advantage of Q-learning is that it does not require prior knowledge of how the learner's actions affect the environment. Final answer: B

\*\*\*\*\*

Answer Option 6:

Statement 1: The log-likelihood of the data will always increase through successive iterations of the expectation-maximization (EM) algorithm.

This statement is false. While the EM algorithm aims to maximize the log-likelihood of the data, it is not guaranteed to always increase through successive iterations. The algorithm can sometimes get stuck in local optima or fail to converge to the global optima.

Statement 2: One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

This statement is false. Q-learning is a model-free reinforcement learning algorithm that does not require prior knowledge of the environment. It learns by interacting with the environment and updating its Q-values based on the observed rewards.

Final answer: B

\*\*\*\*\*

Answer Option 7:

Statement 1: The log-likelihood of the data will always increase through successive iterations of the expectation-maximization algorithm.

This statement is not necessarily true. The expectation-maximization algorithm aims to maximize the likelihood of the observed data, but it can sometimes converge to a local optimum instead of the global optimum, causing the log-likelihood to decrease in some iterations. So the correct answer for Statement 1 is: False.

Statement 2: One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

This statement is also not true. Q-learning is a model-free reinforcement learning algorithm that does not require prior knowledge of how actions affect the environment. It learns by interacting with the environment and updating its action-value function based on observed rewards. So the correct answer for Statement 2 is: False.

Final answer: B

\*\*\*\*\*

Answer Option 8:

Statement 1: The log-likelihood of the data will always increase through successive iterations of the expectation-maximization (EM) algorithm.

This is not necessarily true. The EM algorithm aims to maximize the likelihood function, but it can converge to a local maximum instead of the global maximum. Therefore, the log-likelihood may not always increase through successive iterations. Answer: False

Statement 2: One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

This is not true. Q-learning is a reinforcement learning algorithm that does not require prior knowledge of the environment. It uses an exploration-exploitation strategy to learn the optimal action-value function.

Answer: False

Final answer: B

\*\*\*\*\*

Answer Option 9:

Statement 1: The log-likelihood of the data will always increase through successive iterations of the expectation maximization algorithm.

This statement is false. The expectation maximization (EM) algorithm is an iterative method used to estimate parameters in probabilistic models. While it typically aims to maximize the likelihood of the observed data, there is no guarantee that the log-likelihood will always increase in each iteration. In fact, it is possible for the log-likelihood to sometimes decrease before reaching a maximum.

Statement 2: One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

This statement is false. Q-learning is a model-free reinforcement learning algorithm that does not require prior knowledge of the environment. It learns by interacting with the environment, updating its Q-values based on the observed rewards and actions. Q-learning can be used in situations where the learner has no prior knowledge of its actions' effects on the environment.

Final answer: B

\*\*\*\*\*

Question 7:

'Let us say that we have computed the gradient of our cost function and stored it in a vector  $g$ . What is the cost of one gradient descent update given the gradient?', A. ' $O(D)$ ', B. ' $O(N)$ ', C. ' $O(ND)$ ', D.

$O(ND^2)$

=====

Answer Option 1:

The cost of one gradient descent update is proportional to the dimensionality ( $D$ ) of the problem. Therefore, the correct answer is A. ' $O(D)$ '.

\*\*\*\*\*

Answer Option 2:

The cost of one gradient descent update is typically proportional to the number of features ( $D$ ) in the model. Therefore, the correct answer is A: ' $O(D)$ '.

\*\*\*\*\*

Answer Option 3:

The cost of one gradient descent update is proportional to the number of dimensions,  $D$ , and not the number of training examples,  $N$ . Therefore, the correct answer is A. Final answer: A

\*\*\*\*\*

Answer Option 4:

The cost of one gradient descent update is proportional to the number of parameters ( $D$ ) in the model and does not depend on the number of training examples ( $N$ ). Therefore, the correct answer is A. Final answer: A

\*\*\*\*\*

Answer Option 5:

The cost of one gradient descent update, given the gradient stored in a vector  $g$ , is proportional to the number of features ( $D$ ). Since we are updating the parameters for each feature, the time complexity is  $O(D)$ . Final answer: A

\*\*\*\*\*

Answer Option 6:

The cost of one gradient descent update is determined by the number of features ( $D$ ) in the dataset. Since the gradient ( $g$ ) has the same dimensionality as the features, the cost of one gradient descent update is proportional to the number of features ( $D$ ). Therefore, the correct answer is A. Final answer: A

\*\*\*\*\*

Answer Option 7:

The cost of one gradient descent update, given the gradient stored in the vector  $g$ , depends on the number of parameters  $D$  in the model. The time complexity of performing one update is proportional to the number of parameters, so it is  $O(D)$ . Final answer: A

\*\*\*\*\*

Answer Option 8:

The cost of one gradient descent update, given the gradient stored in a vector  $g$ , is  $O(D)$ , where  $D$  is the number of parameters in the model. Each parameter needs to be updated individually based on the corresponding element in the gradient vector. The time complexity scales linearly with the number of parameters. Final answer: A.

\*\*\*\*\*

Answer Option 9:

The cost of one gradient descent update, given the gradient stored in vector  $g$ , would be proportional to the number of dimensions (features) in the data. Therefore, the cost is linear in terms of the number of dimensions. Final answer: A

\*\*\*\*\*

Question 8:

'Statement 1| For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ . Statement 2| Decision tree is learned by minimizing information gain.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true as the probability distribution function  $p(x)$  for a continuous random variable  $x$  should always be between 0 and 1 for all possible values of  $x$ . Statement 2 is false since decision trees are learned by maximizing information gain, not minimizing it. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ .

Statement 2 is false. Decision trees are learned by maximizing information gain, not minimizing it.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ .

Statement 2 is false. Decision tree is learned by minimizing impurity measures such as Gini impurity or entropy, not information gain.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ .

Statement 2 is false. Decision trees are learned by minimizing impurity measures, such as Gini impurity or entropy, rather than information gain.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1: For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ . This statement is true as the probability of any value for a continuous random variable must be between 0 and 1.

Statement 2: Decision tree is learned by minimizing information gain. This statement is false. Decision trees are learned by minimizing impurity measures such as Gini Index or Cross-Entropy.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ . This statement is true because the probability of any event or value for a continuous random variable falls between 0 and 1.

Statement 2: Decision tree is learned by minimizing information gain. This statement is false. Decision trees are learned by minimizing impurity measures such as Gini index or entropy, not information gain.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ . This statement is true because the probability of any value of a continuous random variable falls between 0 and 1.

Statement 2: Decision tree is learned by minimizing information gain. This statement is true because decision trees are typically learned by selecting the splitting criteria that maximizes information gain, which is a measure of the reduction in entropy or impurity.

Final answer: A. True, True

\*\*\*\*\*

Answer Option 8:

The statement "For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ " is true. The probability distribution function represents the probability density of the random variable, and its values must be non-negative and not exceed 1.

The statement "Decision tree is learned by minimizing information gain" is false. Decision trees are learned by various algorithms that aim to maximize information gain or minimize impurity measures such as Gini Index or entropy.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: For a continuous random variable  $x$  and its probability distribution function  $p(x)$ , it holds that  $0 \leq p(x) \leq 1$  for all  $x$ . This statement is true, as the probability of any outcome for a continuous random variable must be between 0 and 1.

Statement 2: Decision tree is learned by minimizing information gain. This statement is true as well. Decision trees are learned by recursively splitting the data based on the feature that maximizes the information gain, which measures how much the entropy of the target variable decreases after the split.

Final answer: A. 'True, True'

\*\*\*\*\*

Question 9:

Consider the Bayesian network given below. How many independent parameters are needed for this Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ ?', A. '2', B. '4', C. '8', D. '16'

=====

Answer Option 1:

For the Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , we need to count the number of independent parameters required.

Starting from the top:

- H: 2 possible values (True or False)
- U: 2 possible values
- P: 2 possible values
- W: 2 possible values

To calculate the number of independent parameters, we multiply the number of possible values for each variable:

$$2 * 2 * 2 * 2 = 16$$

Therefore, the correct answer is D, '16'.

=====

Answer Option 2:

In the given Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , we can calculate the number of independent parameters needed by considering the number of parameters for each node conditioned on its parents.

H has no parents, so it requires 2 parameters (one for each possible value).

U has one parent (H), so it requires  $2 * 2 = 4$  parameters.

P has one parent (U), so it requires  $2 * 2 = 4$  parameters.

W has one parent (P), so it requires  $2 * 2 = 4$  parameters.

Summing up, we get  $2 + 4 + 4 + 4 = 14$  parameters. Final answer: D

=====

Answer Option 3:

In the given Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , each node depends only on its immediate parent. Therefore, the number of independent parameters needed for each node is equal to the number of possible values for that node's parent.

H has no parent, so it requires 2 independent parameters (one for each possible value: true or false).

U has one parent (H), so it requires 2 independent parameters (one for each possible value of U given H: true or false).

P has one parent (U), so it requires 2 independent parameters.

W has one parent (P), so it requires 2 independent parameters.

Therefore, the total number of independent parameters needed for this Bayesian network is  $2 + 2 + 2 + 2 = 8$ . Final answer: C

=====

Answer Option 4:

In the given Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , each node depends only on its immediate parent, and the number of independent parameters needed for each node is equal to the number of possible values of the node's parent.

H has no parents, so it requires 2 independent parameters:  $P(H=\text{true})$  and  $P(H=\text{false})$ .

U has 1 parent (H), so it requires 2 independent parameters:  $P(U=\text{true}|H=\text{true})$  and  $P(U=\text{true}|H=\text{false})$ .

P has 1 parent (U), so it requires 2 independent parameters:  $P(P=\text{true}|U=\text{true})$  and  $P(P=\text{true}|U=\text{false})$ .

W has 1 parent (P), so it requires 2 independent parameters:  $P(W=\text{true}|P=\text{true})$  and  $P(W=\text{true}|P=\text{false})$ .

Therefore, the total number of independent parameters needed for this Bayesian network is  $2 + 2 + 2 + 2 = 8$ . Final answer: C

=====

Answer Option 5:

In a Bayesian network, each variable depends on its parents in the network. The number of independent parameters needed for a variable is equal to the number of possible values for that variable multiplied by the number of possible values for its parents.

In this Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , we have:

- H with 2 possible values (e.g., true or false)
- U with 2 possible values
- P with 2 possible values
- W with 2 possible values

Since each variable only depends on its immediate parents, we have:

- H depends on no parent, so it requires 2 parameters.
- U depends on H, so it requires 2 parameters.
- P depends on U, so it requires 2 parameters.
- W depends on P, so it requires 2 parameters.

Therefore, the total number of independent parameters needed for this Bayesian network is  $2 + 2 + 2 + 2 = 8$ . Final Answer: C

\*\*\*\*\*

Answer Option 6:

In the given Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , we have 4 nodes: H, U, P, and W.

To determine the number of independent parameters needed, we need to consider the number of parameters for each node based on its parents.

H has no parents, so it needs 2 parameters:  $P(H)$  and  $P(\text{not } H)$ .

U has one parent (H), so it needs 4 parameters:  $P(U|H)$  and  $P(U|\text{not } H)$  for each possible value of H.

P has one parent (U), so it needs 4 parameters:  $P(P|U)$  and  $P(P|\text{not } U)$  for each possible value of U.

W has one parent (P), so it needs 4 parameters:  $P(W|P)$  and  $P(W|\text{not } P)$  for each possible value of P.

Therefore, the total number of independent parameters needed for this Bayesian network is  $2 + 4 + 4 + 4 = 14$ .

Final answer: D (16) (as option D is the closest option to the actual answer).

\*\*\*\*\*

Answer Option 7:

In the given Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , each variable has a conditional probability table (CPT).

For variable H, it has one CPT with 2 possibilities: H and  $\neg H$ . So, it requires 2 independent parameters.

For variable U, it has one CPT with 2 possibilities for each parent value:  $U|H$  and  $U|\neg H$ . Since H has 2 possibilities, U requires  $2 \times 2 = 4$  independent parameters.

For variable P, it has one CPT with 2 possibilities for each parent value:  $P|U$  and  $P|\neg U$ . Since U has 2 possibilities, P requires  $2 \times 2 = 4$  independent parameters.

For variable W, it has one CPT with 2 possibilities for each parent value:  $W|P$  and  $W|\neg P$ . Since P has 2 possibilities, W requires  $2 \times 2 = 4$  independent parameters.

Therefore, the total number of independent parameters needed for this Bayesian network is  $2 + 4 + 4 + 4 = 14$ .

Final answer: D. '16'

\*\*\*\*\*

Answer Option 8:

In the given Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , we have the following variables:

H: Hypothesis

U: Unobserved variable

P: Evidence variable

W: Evidence variable

To compute the number of independent parameters needed for this network, we need to consider the conditional probability tables (CPTs) for each variable.

The number of independent parameters for each variable is as follows:

H: Since H has no parents, it requires 2 parameters ( $P(H)$ )

U: U has one parent (H), so it requires 2 parameters for each value of H ( $P(U|H)$ ). Hence, for 2 values of H, we need 4 parameters.

P: P has one parent (U), so it requires 2 parameters for each value of U ( $P(P|U)$ ). Hence, for 2 values of U, we need 4 parameters.

W: W has one parent (P), so it requires 2 parameters for each value of P ( $P(W|P)$ ). Hence, for 2 values of P, we need 4 parameters.

Therefore, the total number of independent parameters needed for this Bayesian network is  $2 + 4 + 4 + 4 = 14$ .

Final answer: D. '16'

\*\*\*\*\*

Answer Option 9:

In the given Bayesian network, there are 4 nodes: H, U, P, and W.

The number of independent parameters needed for a Bayesian network is calculated by considering the number of parameters for each node. For each node, we need to consider the number of possible values it can take and the number of possible values for its parents.

For the given network:

- Node H has no parent nodes, so it requires 2 parameters (assuming it can take 2 possible values).

- Node U has one parent node (H), so it requires 4 parameters (assuming it can take 2 possible values and H can take 2 possible values).

- Node P has one parent node (U), so it requires 4 parameters (assuming it can take 2 possible values and U can take 2 possible values).

- Node W has one parent node (P), so it requires 4 parameters (assuming it can take 2 possible values and P can take 2 possible values).

Adding up the parameters for each node, we have:

$2 + 4 + 4 + 4 = 14$ .

Therefore, the correct answer is not given among the choices.

\*\*\*\*\*

Question 10:

'As the number of training examples goes to infinity, your model trained on that data will have:', A. 'Lower variance', B. 'Higher variance', C. 'Same variance', D. 'None of the above'

=====

Answer Option 1:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. Increasing the number of training examples helps to reduce the variability in the model's predictions and leads to a more stable and reliable model. Final answer: A

\*\*\*\*\*

Answer Option 2:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. More training examples help to reduce the variability in the model's predictions and make it more stable. Final answer: A

\*\*\*\*\*

Answer Option 3:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. This is because with more data, the model becomes more generalizable and less sensitive to individual training examples, leading to lower variance. Final answer: A

\*\*\*\*\*

Answer Option 4:

As the number of training examples goes to infinity, a model trained on that data will have lower variance. This is because with more training examples, the model becomes more generalized and less prone to overfitting the training data. Final answer: A

\*\*\*\*\*

Answer Option 5:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. This is because with a large amount of training data, the model becomes more generalized and less sensitive to individual training examples, resulting in lower variance. Final answer: A

\*\*\*\*\*

Answer Option 6:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. Increasing the number of training examples helps to reduce the variance of the model, leading to more stable and reliable predictions. Final answer: A

\*\*\*\*\*

Answer Option 7:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. This is because having more training examples helps the model generalize better and reduces the impact of individual examples on the overall model. Final answer: A

\*\*\*\*\*

Answer Option 8:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. This is known as the law of large numbers, where the larger the sample size, the closer the sample mean (model prediction) will be to the population mean (true value). Final answer: A

\*\*\*\*\*

Answer Option 9:

As the number of training examples goes to infinity, the model trained on that data will have lower variance. Higher variance typically occurs when the model is trained on a smaller dataset, as it may overfit to the specific examples. With an infinite number of training examples, the model becomes more generalized and less prone to overfitting, resulting in lower variance. Final answer: A

\*\*\*\*\*



Question 11:

'Statement 1] The set of all rectangles in the 2D plane (which includes non axisaligned rectangles) can shatter a set of 5 points. Statement 2] The VC-dimension of k-Nearest Neighbour classifier when  $k = 1$  is infinite.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. The set of all rectangles in the 2D plane cannot shatter a set of 5 points.

Statement 2 is true. The VC-dimension of the k-Nearest Neighbor classifier when  $k = 1$  is infinite.

Final Answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 stating that the set of all rectangles in the 2D plane can shatter a set of 5 points is true according to the VC-dimension theory. Statement 2 stating that the VC-dimension of k-Nearest Neighbour classifier when  $k = 1$  is infinite is false. The VC-dimension of k-Nearest Neighbour classifier when  $k = 1$  is 1. Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1: The set of all rectangles in the 2D plane (including non-axis-aligned rectangles) cannot shatter a set of 5 points. This is because there will always be at least one configuration of 5 points that cannot be completely enclosed by any rectangle.

Statement 2: The VC-dimension of k-Nearest Neighbour classifier when  $k = 1$  is not infinite. The VC-dimension of the k-Nearest Neighbour classifier with any value of  $k$  is equal to  $k+1$ .

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1: The set of all rectangles in the 2D plane can shatter a set of 5 points. This statement is true because it is possible to find arrangements of 5 points that can be shattered by different rectangles.

Statement 2: The VC-dimension of a k-Nearest Neighbour classifier when  $k = 1$  is infinite. This statement is false. The VC-dimension of a k-Nearest Neighbour classifier when  $k = 1$  is not infinite, but rather equal to the dimensionality of the input space.

Therefore, the correct answer is C. 'True, False'

\*\*\*\*\*

Answer Option 5:

The first statement is true. The set of all rectangles in the 2D plane can shatter a set of 5 points, meaning that it is possible to find a rectangle that can perfectly separate any arrangement of 5 points.

The second statement is false. The VC-dimension of k-Nearest Neighbour classifier when  $k = 1$  is not infinite. The VC-dimension of k-Nearest Neighbour is actually  $k+1$ , so when  $k = 1$ , the VC-dimension is 2.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 6:

Statement 1 suggests that the set of all rectangles in the 2D plane can shatter a set of 5 points. This statement is true because rectangles can be adjusted and oriented to perfectly separate any possible labeling of 5 points in the plane.

Statement 2 states that the VC-dimension of the k-Nearest Neighbour classifier when  $k = 1$  is infinite. This statement is false. The VC-dimension of the k-Nearest Neighbour classifier is bounded by 1 when  $k = 1$ , meaning it cannot shatter sets of more than 1 point.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 7:

Statement 1 states that the set of all rectangles in the 2D plane can shatter a set of 5 points. This is true because it is possible to separate any set of 5 points using rectangles of different sizes and orientations.

Statement 2 states that the VC-dimension of k-Nearest Neighbour classifier when  $k = 1$  is infinite. This is false because when  $k = 1$ , the VC-dimension of the k-Nearest Neighbour classifier is actually 1, as it can only shatter 1 point.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 8:

Statement 1: The set of all rectangles in the 2D plane (including non-axis-aligned rectangles) can shatter a set of 5 points.

This statement is true. The VC-dimension of the set of all rectangles in the 2D plane is at least 5, which means it can shatter a set of 5 points.

Statement 2: The VC-dimension of k-Nearest Neighbor classifier when  $k = 1$  is infinite.

This statement is false. The VC-dimension of the k-Nearest Neighbor classifier when  $k = 1$  is not infinite. It is equal to the dimensionality of the feature space.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 9:

Statement 1: The set of all rectangles in the 2D plane (which includes non-axis-aligned rectangles) can shatter a set of 5 points.

This statement is false. The VC-dimension of the set of all rectangles in the 2D plane is 4, which means that it can shatter at most 4 points, not 5.

Statement 2: The VC-dimension of k-Nearest Neighbor classifier when  $k = 1$  is infinite.

This statement is true. When  $k = 1$ , the VC-dimension of the k-Nearest Neighbor classifier is infinite. This is because the classifier can perfectly separate any set of points by simply memorizing and classifying each point individually.

Final answer: C

\*\*\*\*\*

Question 12:

'\_ refers to a model that can neither model the training data nor generalize to new data.', A. 'good fitting', B. 'overfitting', C. 'underfitting', D. 'all of the above'

=====

Answer Option 1:

Underfitting refers to a model that cannot adequately capture the patterns in the training data and also fails to generalize well to new data. Therefore, the correct answer is C. 'underfitting'.

\*\*\*\*\*

Answer Option 2:

Underfitting refers to a model that can neither model the training data nor generalize to new data. Good fitting refers to a model that can accurately model the training data and generalize to new data.

Overfitting refers to a model that models the training data too closely and does not generalize well to new data. Therefore, the correct answer is C. Final answer: C

\*\*\*\*\*

Answer Option 3:

Underfitting refers to a model that can neither model the training data nor generalize to new data. Good fitting refers to a model that accurately models the training data and can generalize to new data.

Overfitting refers to a model that models the training data too closely and does not generalize well to new data. Therefore, the correct answer is C. 'underfitting'.

\*\*\*\*\*

Answer Option 4:

Underfitting refers to a model that cannot effectively model the training data and also fails to generalize to new data. Good fitting refers to a model that adequately captures the patterns in the training data and

can generalize to new data. Overfitting refers to a model that excessively fits the training data but fails to generalize well to new data. Therefore, the correct answer is C: 'underfitting'.

\*\*\*\*\*

Answer Option 5:

Underfitting refers to a model that can neither model the training data nor generalize to new data. Good fitting refers to a model that adequately captures the patterns and relationships in the training data and

generalizes well to new data. Overfitting refers to a model that memorizes the training data too well and fails to generalize to new data. Therefore, the correct answer is C: 'underfitting'.

\*\*\*\*\*

Answer Option 6:

Underfitting refers to a model that can neither model the training data nor generalize to new data. Good fitting refers to a model that successfully captures the patterns in the training data and generalizes well

to new data. Overfitting refers to a model that models the training data too closely, leading to poor generalization to new data. Therefore, the correct answer is C. Final answer: C.

\*\*\*\*\*

Answer Option 7:

Underfitting refers to a model that cannot adequately capture the patterns and relationships in the training data, leading to poor performance on both the training data and new data. Overfitting refers to a

model that fits the training data too closely, capturing noise and irrelevant patterns, but performing poorly on new data. Good fitting refers to a model that strikes a balance between capturing the patterns in

the training data and generalizing well to new data. Therefore, the correct answer is C. 'underfitting'.

\*\*\*\*\*

Answer Option 8:

Underfitting refers to a model that is too simple and cannot capture the underlying patterns in the training data, leading to poor performance on both the training data and new data. Overfitting, on the other

hand, refers to a model that is overly complex and memorizes the training data, resulting in poor generalization to new data. Good fitting refers to a model that strikes a balance between capturing the patterns

in the training data and generalizing well to new data. Therefore, the correct answer is C. 'underfitting'.

\*\*\*\*\*

Answer Option 9:

Underfitting refers to a model that cannot adequately capture the patterns and relationships in the training data, and therefore it also struggles to generalize well to new data. Overfitting, on the other hand,

refers to a model that models the training data too closely and does not generalize well. Good fitting refers to a model that strikes a balance between capturing the patterns in the training data and generalizing

well to new data. Therefore, the correct answer is C. 'underfitting'.

\*\*\*\*\*

Question 13:

'Statement 1] The F1 score can be especially useful for datasets with class high imbalance. Statement 2] The area under the ROC curve is one of the main metrics used to assess anomaly detectors.', A.

'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. The F1 score is a metric that combines precision and recall, making it particularly useful for datasets with class imbalance.

Statement 2 is false. The area under the ROC curve (AUC-ROC) is commonly used to assess classification models, not anomaly detectors. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. The F1 score is a metric that combines precision and recall, making it useful for datasets with class imbalance.

Statement 2 is false. The area under the ROC curve (AUC-ROC) is commonly used to evaluate the performance of binary classifiers, not anomaly detectors. Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. The F1 score is a metric that combines precision and recall, making it particularly useful for datasets with class imbalance.

Statement 2 is false. The area under the ROC curve (AUC-ROC) is commonly used to assess binary classifiers, not anomaly detectors. Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. The F1 score is a metric that combines precision and recall, making it useful for imbalanced datasets where one class may have significantly more instances than the other.

Statement 2 is false. The area under the ROC curve (AUROC) is commonly used to assess binary classification models, not specifically anomaly detectors. Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is true. The F1 score, which considers both precision and recall, can be particularly useful for datasets with class imbalance as it provides a balanced measure of the model's performance.

Statement 2 is false. The area under the ROC curve (AUC-ROC) is commonly used to assess the performance of binary classifiers, not anomaly detectors. Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: The F1 score can indeed be especially useful for datasets with class imbalance since it takes into account both precision and recall, which are crucial when dealing with imbalanced classes.

Statement 2: The area under the ROC curve (AUC-ROC) is commonly used to assess the performance of binary classifiers, including anomaly detectors. Final answer: A. True, True

\*\*\*\*\*

Answer Option 7:

Statement 1: The F1 score can be especially useful for datasets with class high imbalance.

This statement is true because the F1 score takes into account both precision and recall, which are important when dealing with imbalanced datasets.

Statement 2: The area under the ROC curve is one of the main metrics used to assess anomaly detectors.

This statement is also true because the ROC curve and the area under it provide a good measure of the performance of anomaly detectors.

Final answer: A

\*\*\*\*\*

Answer Option 8:

Statement 1 is true. The F1 score is a metric commonly used in classification tasks, and it combines precision and recall. It is particularly useful for datasets with class imbalance as it considers both false positives and false negatives.

Statement 2 is false. The area under the ROC (Receiver Operating Characteristic) curve is commonly used to assess classification models, but not specifically anomaly detectors. Anomaly detection typically uses metrics such as precision, recall, or F1 score to evaluate performance. Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The F1 score can be especially useful for datasets with class high imbalance.

This statement is true. The F1 score takes into account both precision and recall, making it a suitable metric for evaluating models on imbalanced datasets where the class distribution is skewed.

Statement 2: The area under the ROC curve is one of the main metrics used to assess anomaly detectors.

This statement is also true. The area under the ROC curve (AUC-ROC) is commonly used to evaluate anomaly detection models because it provides a measure of the model's ability to discriminate between normal and anomalous instances.

Therefore, the correct answer is A. 'True, True'.

\*\*\*\*\*

Question 14:

'Statement 1] The back-propagation algorithm learns a globally optimal neural network with hidden layers. Statement 2] The VC dimension of a line should be at most 2, since I can find at least one case of 3 points that cannot be shattered by any line.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. The back-propagation algorithm learns a locally optimal neural network with hidden layers, but not necessarily a globally optimal one.

Statement 2 is true. The VC dimension of a line is at most 2, as there exists at least one case of 3 points that cannot be shattered by any line.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: The back-propagation algorithm learns a locally optimal neural network with hidden layers, not necessarily globally optimal.

Statement 2: The VC dimension of a line in 2D is 3, so the statement that at least one case of 3 points cannot be shattered by any line is false.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1: The back-propagation algorithm does not guarantee learning a globally optimal neural network with hidden layers. It can find a locally optimal solution but not necessarily the global optimum. So,

Statement 1 is False.

Statement 2: The VC dimension of a line in 2D is 3. This is because there exist cases where 3 points cannot be shattered by any line. So, Statement 2 is True.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1: The back-propagation algorithm learns a locally optimal neural network with hidden layers, not necessarily a globally optimal one. False.

Statement 2: The VC (Vapnik-Chervonenkis) dimension of a line in 2D space is 3, not 2. It is possible to find at least one case of 3 points that cannot be shattered by any line. True.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1: The back-propagation algorithm does not guarantee to learn a globally optimal neural network with hidden layers. It is a gradient-based optimization algorithm that aims to find a local minimum of the loss function.

Statement 2: The VC (Vapnik-Chervonenkis) dimension of a line is 3, not 2. This means that there exists a set of 3 points that can be shattered by a line.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: The back-propagation algorithm learns a locally optimal neural network with hidden layers, not a globally optimal one. Therefore, Statement 1 is False.

Statement 2: The VC (Vapnik-Chervonenkis) dimension defines the capacity of a learning algorithm to shatter (correctly classify) all possible dichotomies of a given size. The VC dimension of a line is 3, not at most 2, because it can correctly shatter any set of three points in a 2D space. Therefore, Statement 2 is False.

Final Answer: D

\*\*\*\*\*

Answer Option 7:

Statement 1: The back-propagation algorithm learns a locally optimal neural network with hidden layers. This is because the algorithm uses gradient descent to update the weights and biases of the neural network, which can get stuck in local optima.

Statement 2: The VC dimension of a line is 3, not 2. A line can shatter any set of 2 points, but there exists at least one case of 3 points that cannot be shattered by any line. This means that the VC dimension of a line is at least 3.

Therefore, the correct answer is D. 'False, True'.

\*\*\*\*\*

Answer Option 8:

Statement 1: The back-propagation algorithm learns a globally optimal neural network with hidden layers.

This statement is False. The back-propagation algorithm is used to train neural networks, but it does not guarantee finding the globally optimal solution. It is a local optimization algorithm that can find a locally optimal solution based on the provided data and model architecture.

Statement 2: The VC dimension of a line should be at most 2, since I can find at least one case of 3 points that cannot be shattered by any line.

This statement is True. The VC dimension of a line is 2. This means that, with a line, it is possible to shatter any arrangement of 2 points, but not any arrangement of 3 points. Therefore, the VC dimension is at most 2 for a line.

Final Answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The back-propagation algorithm learns a globally optimal neural network with hidden layers.

This statement is False. The back-propagation algorithm is used to train neural networks, but it does not guarantee finding the globally optimal solution. It may converge to a local minimum instead.

Statement 2: The VC dimension of a line should be at most 2, since I can find at least one case of 3 points that cannot be shattered by any line.

This statement is True. The VC dimension of a hypothesis class is the maximum number of points that can be shattered by the class. In the case of a line, it has a VC dimension of at most 2 because there exists at least one case of 3 points that cannot be shattered by any line (e.g., three points forming a triangle that is not a straight line).

Final answer: C

\*\*\*\*\*

Question 15:

'High entropy means that the partitions in classification are', A. 'pure', B. 'not pure', C. 'useful', D. 'useless'

=====

Answer Option 1:

High entropy means that the partitions in classification are not pure. Entropy measures the impurity or disorder of a set of data. A higher entropy value indicates a higher level of uncertainty or impurity in the classification. Final answer: B

=====

Answer Option 2:

High entropy means that the partitions in classification are not pure. Entropy measures the impurity or disorder within a set of data, where high entropy indicates a higher level of uncertainty or mixed classes within the partitions. Final answer: B

=====

Answer Option 3:

High entropy in classification means that the partitions are not pure. Entropy is a measure of impurity in a set of data, and high entropy indicates that the classes within the partitions are mixed or not well separated. Final answer: B

=====

Answer Option 4:

High entropy means that the partitions in classification are not pure. Entropy measures the impurity or disorder of a set of data. A high entropy value indicates that the classes in the partitions are mixed or uncertain. Final answer: B

=====

Answer Option 5:

High entropy means that the partitions in classification are not pure. Entropy is a measure of impurity or disorder in a set of samples, and high entropy indicates that the partitions contain a mixture of different classes. Final answer: B

=====

Answer Option 6:

High entropy means that the partitions in classification are not pure. Entropy measures the impurity or disorder in a set of data, and higher entropy indicates that the partitions are more mixed or less distinct. Final answer: B

=====

Answer Option 7:

High entropy means that the partitions in classification are not pure. Entropy is a measure of randomness or uncertainty in the data. If the entropy is high, it means that the partitions are not well-separated and contain a mixture of different classes. Therefore, the correct answer is B. Final answer: B.

=====

Answer Option 8:

High entropy means that the partitions in a classification problem are not pure. Entropy measures the impurity or disorder in a set of data. So, a higher entropy value indicates that the partitions are not pure and contain a mixture of different classes. Final answer: B

=====

Answer Option 9:

High entropy means that the partitions in classification are not pure. Entropy measures the impurity or disorder in a set of class labels. Higher entropy indicates a higher degree of uncertainty or mixed class labels within the partitions. Final answer: B

=====

Question 16:

'Statement 1] Layer Normalization is used in the original ResNet paper, not Batch Normalization. Statement 2] DCGANs use self-attention to stabilize training.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. The original ResNet paper actually used batch normalization.

Statement 2 is false. DCGANs (Deep Convolutional Generative Adversarial Networks) do not use self-attention to stabilize training. Final Answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is false. The original ResNet paper actually used batch normalization, not layer normalization. Statement 2 is also false. DCGANs (Deep Convolutional Generative Adversarial Networks) do not use self-attention to stabilize training. Final answer: D

\*\*\*\*\*

Answer Option 3:

Statement 1 is False. The original ResNet paper does indeed use Batch Normalization, not Layer Normalization.

Statement 2 is False as well. DCGANs (Deep Convolutional Generative Adversarial Networks) do not use self-attention to stabilize training. Final answer: C

\*\*\*\*\*

Answer Option 4:

The correct answer is D. 'False, True'.

Statement 1 is false because the original ResNet paper actually uses Batch Normalization, not Layer Normalization.

Statement 2 is true because DCGANs (Deep Convolutional Generative Adversarial Networks) do use self-attention as a technique to stabilize training.

\*\*\*\*\*

Answer Option 5:

Statement 1 is false. The original ResNet paper used batch normalization, not layer normalization.

Statement 2 is false. DCGANs (Deep Convolutional Generative Adversarial Networks) do not use self-attention to stabilize training. Instead, they use techniques like batch normalization and alternative loss functions. Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is false. The original ResNet paper actually used Batch Normalization, not Layer Normalization. Statement 2 is also false. DCGANs (Deep Convolutional Generative Adversarial Networks) do not use self-attention to stabilize training. Final answer: D

\*\*\*\*\*

Answer Option 7:

Statement 1 is false. The original ResNet paper actually uses Batch Normalization, not Layer Normalization.

Statement 2 is false as well. DCGANs (Deep Convolutional Generative Adversarial Networks) do not use self-attention to stabilize training. They primarily use convolutional layers and transposed convolutional layers for training stability. Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1 is false. The original ResNet paper actually used Batch Normalization, not Layer Normalization.

Statement 2 is false. DCGANs (Deep Convolutional Generative Adversarial Networks) do not use self-attention to stabilize training. They primarily use techniques like convolutional layers, transposed convolutions, and batch normalization.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: Layer Normalization is used in the original ResNet paper, not Batch Normalization. This statement is False. Batch Normalization is indeed used in the original ResNet paper.

Statement 2: DCGANs use self-attention to stabilize training. This statement is also False. DCGANs typically do not use self-attention; instead, they use techniques like batch normalization and convolutional layers for training stability.

Final answer: D

\*\*\*\*\*



Question 17:

'In building a linear regression model for a particular data set, you observe the coefficient of one of the features having a relatively high negative value. This suggests that', A. 'This feature has a strong effect on the model (should be retained)', B. 'This feature does not have a strong effect on the model (should be ignored)', C. 'It is not possible to comment on the importance of this feature without additional information', D. 'Nothing can be determined.'

=====

Answer Option 1:

A relatively high negative value for the coefficient of a feature in a linear regression model suggests that this feature has a strong negative effect on the model. Therefore, it should be retained as it has a significant impact on the predictions made by the model. Final answer: A

\*\*\*\*\*

Answer Option 2:

If the coefficient of a feature in a linear regression model has a relatively high negative value, it suggests that this feature has a strong negative effect on the model. Therefore, it should be retained and considered important for the model. Final answer: A

\*\*\*\*\*

Answer Option 3:

When the coefficient of a feature in a linear regression model has a relatively high negative value, it suggests that this feature has a strong negative effect on the model. Therefore, it should be retained as it contributes significantly to the prediction. Final answer: A

\*\*\*\*\*

Answer Option 4:

When the coefficient of a feature in a linear regression model has a relatively high negative value, it suggests that this feature has a strong negative effect on the model. Therefore, it should be retained in the model as it provides valuable information for predicting the target variable. Final answer: A

\*\*\*\*\*

Answer Option 5:

In linear regression, the coefficient of a feature represents the impact or effect of that feature on the target variable. A relatively high negative value suggests that the feature has a strong negative effect on the model. Therefore, it should be retained as it contributes significantly to the model's prediction. Final answer: A.

\*\*\*\*\*

Answer Option 6:

If the coefficient of a feature in a linear regression model has a relatively high negative value, it suggests that the feature has a strong negative effect on the model. Therefore, it should be retained as it is contributing significantly to the prediction. Final answer: A

\*\*\*\*\*

Answer Option 7:

If the coefficient of a feature in a linear regression model has a relatively high negative value, it suggests that this feature has a strong negative effect on the model. In other words, this feature has a strong influence on the outcome variable and should be retained in the model. Final answer: A

\*\*\*\*\*

Answer Option 8:

In linear regression, the coefficient of a feature represents the impact or influence of that feature on the target variable. If the coefficient of a feature has a relatively high negative value, it suggests that the feature has a strong negative effect on the model. Therefore, it is important to retain this feature in the model. Final answer: A

\*\*\*\*\*

Answer Option 9:

In building a linear regression model, if a coefficient of a feature has a relatively high negative value, it suggests that the feature has a strong negative effect on the model. This means that as the value of the feature increases, the predicted output will decrease. Therefore, the feature should be retained as it is important for the model. Final answer: A

\*\*\*\*\*

Question 18:

'For a neural network, which one of these structural assumptions is the one that most affects the trade-off between underfitting (i.e. a high bias model) and overfitting (i.e. a high variance model);', A. 'The number of hidden nodes', B. 'The learning rate', C. 'The initial choice of weights', D. 'The use of a constant-term unit input'

=====

Answer Option 1:

The structural assumption that most affects the trade-off between underfitting and overfitting in a neural network is the number of hidden nodes. The number of hidden nodes determines the complexity and capacity of the neural network, which directly impacts its ability to capture and represent patterns in the data. Final answer: A

\*\*\*\*\*

Answer Option 2:

The number of hidden nodes in a neural network is a structural assumption that most affects the trade-off between underfitting and overfitting. The complexity of the neural network, determined by the number of hidden nodes, can greatly impact its ability to capture complex relationships in the data. Final answer: A

\*\*\*\*\*

Answer Option 3:

For a neural network, the number of hidden nodes is the structural assumption that most affects the trade-off between underfitting and overfitting. If the number of hidden nodes is too low, the model may underfit and have high bias. Conversely, if the number of hidden nodes is too high, the model may overfit and have high variance. Final answer: A

\*\*\*\*\*

Answer Option 4:

Among the given options, the structural assumption that most affects the trade-off between underfitting and overfitting in a neural network is the number of hidden nodes. The number of hidden nodes determines the capacity and complexity of the model, and a higher number of hidden nodes can lead to overfitting, while a lower number can result in underfitting. Final answer: A

\*\*\*\*\*

Answer Option 5:

For a neural network, the structural assumption that most affects the trade-off between underfitting and overfitting is the number of hidden nodes. The number of hidden nodes determines the complexity and capacity of the neural network, and having too few hidden nodes may result in underfitting (high bias), while having too many hidden nodes may lead to overfitting (high variance). Final answer: A

\*\*\*\*\*

Answer Option 6:

The structural assumption that most affects the trade-off between underfitting and overfitting in a neural network is the number of hidden nodes. The number of hidden nodes determines the complexity and capacity of the network, which can lead to overfitting if the model becomes too complex or underfitting if the model is too simple. The learning rate, initial choice of weights, and the use of a constant-term unit input can also impact the learning process and performance of the network, but they do not directly affect the underfitting and overfitting trade-off. Final answer: A

\*\*\*\*\*

Answer Option 7:

For a neural network, the number of hidden nodes is the structural assumption that most affects the trade-off between underfitting and overfitting. The number of hidden nodes determines the capacity of the network to learn complex patterns in the data. Fewer hidden nodes may result in underfitting, while more hidden nodes may lead to overfitting. The learning rate, initial choice of weights, and the use of a constant-term unit input can also affect the model's performance, but the number of hidden nodes is the structural assumption that has the most significant impact on the bias-variance trade-off. Final answer:

A

\*\*\*\*\*

Answer Option 8:

The structural assumption that most affects the trade-off between underfitting (high bias) and overfitting (high variance) in a neural network is the number of hidden nodes. The number of hidden nodes determines the complexity and capacity of the model, which can result in underfitting or overfitting based on the amount of information it can capture. The learning rate, initial choice of weights, and the use of a constant-term unit input can also impact the performance of the model, but they do not directly affect the trade-off between underfitting and overfitting. Final answer: A

\*\*\*\*\*

Answer Option 9:

The number of hidden nodes in a neural network is the structural assumption that most affects the trade-off between underfitting (high bias) and overfitting (high variance). The number of hidden nodes determines the complexity and capacity of the model, which can lead to overfitting if there are too many nodes or underfitting if there are too few nodes. The learning rate, initial choice of weights, and the use of a constant-term unit input can also affect the performance of the neural network, but they are not directly related to the trade-off between underfitting and overfitting. Final answer: A

\*\*\*\*\*

Question 19:

'For polynomial regression, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting:', A. 'The polynomial degree', B. 'Whether we learn the weights by matrix inversion or gradient descent', C. 'The assumed variance of the Gaussian noise', D. 'The use of a constant-term unit input'

=====

Answer Option 1:

For polynomial regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the polynomial degree. The higher the degree, the more complex the model becomes, increasing the risk of overfitting. Final answer: A

=====

Answer Option 2:

For polynomial regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the polynomial degree. The higher the degree, the more complex the model becomes, increasing the risk of overfitting. Final answer: A

=====

Answer Option 3:

For polynomial regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the polynomial degree. The higher the degree of the polynomial, the more complex the model becomes, which can lead to overfitting. The choice of polynomial degree is a key factor in finding the right balance between underfitting and overfitting. Final answer: A

=====

Answer Option 4:

For polynomial regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the polynomial degree (option A). The polynomial degree determines the complexity of the model and can lead to underfitting if it is too low or overfitting if it is too high. The other options, such as the method of learning weights or the assumed variance of the Gaussian noise, do not directly influence the trade-off between underfitting and overfitting in the same way as the choice of polynomial degree. Final answer: A

=====

Answer Option 5:

For polynomial regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the polynomial degree (option A). The degree determines the complexity of the model and influences how well it can fit the training data. Increasing the degree can lead to overfitting, while decreasing it can result in underfitting. The other options mentioned (B, C, and D) do not directly impact the trade-off between underfitting and overfitting in polynomial regression. Final answer: A

=====

Answer Option 6:

For polynomial regression, the most important structural assumption that affects the trade-off between underfitting and overfitting is the polynomial degree. The degree determines how flexible the model is in capturing the underlying relationship between the features and the target variable. Final answer: A

=====

Answer Option 7:

The structural assumption that most affects the trade-off between underfitting and overfitting in polynomial regression is the polynomial degree. By increasing the degree, the model becomes more complex and has the potential to overfit the data. Final answer: A

=====

Answer Option 8:

The structural assumption that most affects the trade-off between underfitting and overfitting in polynomial regression is the polynomial degree. The degree of the polynomial determines the complexity of the model and can lead to underfitting or overfitting depending on how well it captures the underlying relationship in the data. Final answer: A

=====

Answer Option 9:

For polynomial regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the polynomial degree. The polynomial degree determines the complexity of the model and how well it can fit the data. Higher degrees can lead to overfitting, while lower degrees can result in underfitting. The other options (B, C, and D) do not directly impact the trade-off between underfitting and overfitting in polynomial regression. Final answer: A

=====

Question 20:

'Statement 1| As of 2020, some models attain greater than 98% accuracy on CIFAR-10. Statement 2| The original ResNets were not optimized with the Adam optimizer.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: As of 2020, some models attain greater than 98% accuracy on CIFAR-10.

This statement is plausible as researchers have achieved high accuracy rates on the CIFAR-10 dataset using deep learning models.

Statement 2: The original ResNets were not optimized with the Adam optimizer.

This statement is false. The original ResNets were not specifically optimized with the Adam optimizer, but rather they were trained using stochastic gradient descent (SGD) with momentum.

Final answer: C

=====

Answer Option 2:

Statement 1| As of 2020, some models attain greater than 98% accuracy on CIFAR-10. This statement is possible as there have been recent advancements in deep learning models achieving high accuracy on CIFAR-10. Statement 2| The original ResNets were not optimized with the Adam optimizer. This statement is false as the original ResNet paper did not specifically mention the Adam optimizer, but it is not accurate to say that they were not optimized with it. Final answer: C

=====

Answer Option 3:

Statement 1: As of 2020, some models attain greater than 98% accuracy on CIFAR-10. - This statement is possible as there have been models that have achieved high accuracy on CIFAR-10 dataset.

Therefore, the statement is true.

Statement 2: The original ResNets were not optimized with the Adam optimizer. - The original ResNets were not specifically optimized or designed with the Adam optimizer in mind. They were commonly optimized using stochastic gradient descent (SGD) with momentum. Therefore, the statement is also true.

Final answer: A. 'True, True'

=====

Answer Option 4:

Statement 1: As of 2020, some models attain greater than 98% accuracy on CIFAR-10.

This statement is plausible as there have been advancements in deep learning models and techniques that have achieved high accuracy on the CIFAR-10 dataset. Therefore, statement 1 is likely to be true.

Statement 2: The original ResNets were not optimized with the Adam optimizer.

The original ResNet architecture was proposed before the Adam optimizer was introduced. It used stochastic gradient descent (SGD) with momentum as the optimization algorithm. Therefore, statement 2 is likely to be true.

Final answer: C

=====

Answer Option 5:

Statement 1| As of 2020, some models attain greater than 98% accuracy on CIFAR-10: This statement is possible as the performance of models can improve over time with advancements in machine learning techniques and computing power. Therefore, it can be considered as true.

Statement 2| The original ResNets were not optimized with the Adam optimizer: This statement is true. The original ResNet architecture was proposed in 2015 before the introduction of the Adam optimizer in 2014. Therefore, it was not optimized using Adam. Final answer: C

=====

Answer Option 6:

Statement 1: As of 2020, some models attain greater than 98% accuracy on CIFAR-10.

This statement is possible as there have been advancements in deep learning models that have achieved high accuracy on the CIFAR-10 dataset. Therefore, the statement is true.

Statement 2: The original ResNets were not optimized with the Adam optimizer.

This statement is also true. The original ResNet models were not optimized with the Adam optimizer as it was not introduced until later. The original ResNets typically used stochastic gradient descent (SGD) as the optimizer. Therefore, the statement is true.

Final answer: C

=====

Answer Option 7:

Statement 1: As of 2020, some models attain greater than 98% accuracy on CIFAR-10.

This statement is plausible because with advances in deep learning architectures and techniques, it is possible for models to achieve high accuracy on CIFAR-10. However, without specific information about the models, it is difficult to determine the exact accuracy achieved. Therefore, statement 1 is likely true.

Statement 2: The original ResNets were not optimized with the Adam optimizer.

This statement is false. The original ResNet paper, published in 2015 by He et al., did not specifically mention the use of the Adam optimizer. However, the Adam optimizer was introduced in 2014, before the publication of the ResNet paper, and has since become a widely used optimization algorithm in deep learning. It is highly probable that ResNets, when implemented and trained by researchers, have been optimized using the Adam optimizer. Therefore, statement 2 is likely false.

Final Answer: C. 'True, False'

\*\*\*\*\*

Answer Option 8:

Statement 1: As of 2020, some models attain greater than 98% accuracy on CIFAR-10. This statement is plausible as there have been significant advancements in deep learning models and techniques, achieving high accuracy on CIFAR-10.

Statement 2: The original ResNets were not optimized with the Adam optimizer. This statement is also plausible as the original ResNet paper introduced the concept of residual learning without specifically mentioning the Adam optimizer.

Final answer: C. 'True, False'

\*\*\*\*\*

Answer Option 9:

Statement 1: As of 2020, some models attain greater than 98% accuracy on CIFAR-10.

This statement is plausible as there have been advancements in deep learning models achieving high accuracy on the CIFAR-10 dataset. Therefore, the statement is likely true.

Statement 2: The original ResNets were not optimized with the Adam optimizer.

This statement is false. The original ResNet paper, "Deep Residual Learning for Image Recognition" by He et al., did not explicitly mention the optimizer used, but it is common practice to use stochastic gradient descent (SGD) with momentum when training neural networks, including ResNets. The Adam optimizer, on the other hand, was introduced in a separate paper by Kingma and Ba in 2014, well after the publication of the original ResNet paper. Therefore, the statement is likely false.

Final answer: C - 'True, False'

\*\*\*\*\*

Question 21:

'The K-means algorithm:', A. 'Requires the dimension of the feature space to be no bigger than the number of samples', B. 'Has the smallest value of the objective function when  $K = 1$ ', C. 'Minimizes the within class variance for a given number of clusters', D. 'Converges to the global optimum if and only if the initial means are chosen as some of the samples themselves'

=====

Answer Option 1:

The K-means algorithm minimizes the within-class variance for a given number of clusters. It does not require the dimension of the feature space to be no bigger than the number of samples, it does not have the smallest value of the objective function when  $K = 1$ , and it does not necessarily converge to the global optimum if the initial means are chosen as some of the samples themselves. Final answer: C

\*\*\*\*\*

Answer Option 2:

The K-means algorithm minimizes the within-class variance for a given number of clusters. It does not require the dimension of the feature space to be no bigger than the number of samples. The smallest value of the objective function occurs when  $K = 1$ . The algorithm may converge to a local optimum, not necessarily the global optimum, regardless of the initial means chosen. Final answer: C

\*\*\*\*\*

Answer Option 3:

The K-means algorithm minimizes the within-cluster variance for a given number of clusters. It does not require the dimension of the feature space to be smaller than the number of samples. The smallest value of the objective function is achieved when  $K = 1$ , but this does not make it the best choice for the algorithm. The convergence to the global optimum does not depend on choosing the initial means as samples themselves. Final answer: C

\*\*\*\*\*

Answer Option 4:

The K-means algorithm minimizes the within-class variance for a given number of clusters. It does not require the dimension of the feature space to be no bigger than the number of samples, it does not necessarily have the smallest value of the objective function when  $K = 1$ , and it does not necessarily converge to the global optimum if the initial means are chosen as some of the samples themselves. Final answer: C

\*\*\*\*\*

Answer Option 5:

The K-means algorithm is an iterative clustering algorithm that aims to partition data points into  $K$  clusters. Among the given options, the correct statement about the K-means algorithm is:

C. 'Minimizes the within-class variance for a given number of clusters.'

The K-means algorithm minimizes the sum of squared distances between each data point and the mean of its assigned cluster. This objective function minimizes the within-class variance, aiming to make the data points within each cluster as similar as possible. Final answer: C

\*\*\*\*\*

Answer Option 6:

The K-means algorithm is an iterative clustering algorithm that aims to partition a dataset into  $K$  clusters. It does not require the dimension of the feature space to be no bigger than the number of samples (A). The smallest value of the objective function is obtained when  $K = 1$  (B) as there is only one cluster. The K-means algorithm minimizes the within-class variance for a given number of clusters (C), and it does not necessarily converge to the global optimum if the initial means are chosen as some of the samples themselves (D). Final answer: C

\*\*\*\*\*

Answer Option 7:

The K-means algorithm does not require the dimension of the feature space to be no bigger than the number of samples, so option A is incorrect. The K-means algorithm does not necessarily have the smallest value of the objective function when  $K = 1$ , so option B is incorrect. The K-means algorithm aims to minimize the within-class variance for a given number of clusters, so option C is correct. The K-means algorithm does not necessarily converge to the global optimum if the initial means are chosen as some of the samples themselves, so option D is incorrect. Final answer: C

\*\*\*\*\*

Answer Option 8:

The K-means algorithm is an iterative clustering algorithm that aims to partition a given dataset into  $K$  distinct clusters. It does not require the dimension of the feature space to be no bigger than the number of samples (option A is incorrect). It also does not necessarily have the smallest value of the objective function when  $K = 1$  (option B is incorrect). The K-means algorithm minimizes the within-cluster variance for a given number of clusters, not within-class variance (option C is incorrect). Additionally, it does not require the initial means to be chosen as some of the samples themselves for convergence to the global optimum (option D is incorrect). Final answer: None of the above.

\*\*\*\*\*

Answer Option 9:

The K-means algorithm is an iterative clustering algorithm. It minimizes the within-cluster variance for a given number of clusters, so the statement 'The K-means algorithm minimizes the within-class variance for a given number of clusters' (C) is correct. The algorithm does not have a strict requirement on the dimension of the feature space in relation to the number of samples, so statement A is incorrect. The smallest value of the objective function is achieved when  $K = 1$ , so statement B is also incorrect. The algorithm does not guarantee convergence to the global optimum if the initial means are chosen as some of the samples themselves, so statement D is incorrect. Final answer: C

\*\*\*\*\*

Question 22:

'Statement 1| VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels. Statement 2| Data-dependent weight initialization procedures were introduced before Batch Normalization.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. VGGNets indeed have convolutional kernels of smaller width and height compared to AlexNet's first-layer kernels.

Statement 2 is false. Data-dependent weight initialization procedures were introduced after the introduction of Batch Normalization.

Final answer: C

=====

Answer Option 2:

Statement 1 is true. VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels.

Statement 2 is false. Data-dependent weight initialization procedures were introduced after Batch Normalization.

Final answer: C

=====

Answer Option 3:

Statement 1: VGGNets do have convolutional kernels of smaller width and height than AlexNet's first-layer kernels. This statement is true.

Statement 2: Data-dependent weight initialization procedures were actually introduced after Batch Normalization. This statement is false.

Final answer: C

=====

Answer Option 4:

Statement 1: VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels.

This statement is false. VGGNet and AlexNet both use convolutional kernels of the same size, which is typically 3x3.

Statement 2: Data-dependent weight initialization procedures were introduced before Batch Normalization.

This statement is true. Data-dependent weight initialization techniques, such as Xavier or He initialization, were introduced before Batch Normalization became popular.

Final answer: C

=====

Answer Option 5:

Statement 1: 'VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels.'

This statement is False. In AlexNet, the first-layer kernels have a smaller width and height compared to VGGNet.

Statement 2: 'Data-dependent weight initialization procedures were introduced before Batch Normalization.'

This statement is True. Data-dependent weight initialization procedures, such as Xavier and He initialization, were introduced before Batch Normalization.

Final answer: C

=====

Answer Option 6:

Statement 1: VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels.

This statement is False. VGGNet and AlexNet both use 3x3 convolutional kernels in their first layers.

Statement 2: Data-dependent weight initialization procedures were introduced before Batch Normalization.

This statement is True. Data-dependent weight initialization procedures, such as Xavier and He initialization, were introduced before the introduction of Batch Normalization.

Final answer: C

=====

Answer Option 7:

Statement 1: VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels.

This statement is false. VGGNet actually has convolutional kernels of larger width and height compared to AlexNet's first-layer kernels.

Statement 2: Data-dependent weight initialization procedures were introduced before Batch Normalization.

This statement is true. Data-dependent weight initialization procedures were introduced before the concept of Batch Normalization.

Final answer: C

=====

Answer Option 8:

Statement 1: VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels. This statement is False. In fact, VGGNets have convolutional kernels of the same size as AlexNet's first-layer kernels, which is typically 3x3.

Statement 2: Data-dependent weight initialization procedures were introduced before Batch Normalization. This statement is True. Data-dependent weight initialization procedures, such as the Glorot initialization and He initialization, were introduced before Batch Normalization became popular in deep learning.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: VGGNets have convolutional kernels of smaller width and height than AlexNet's first-layer kernels.

This statement is false. VGGNets and AlexNet both use convolutional kernels of the same size, typically 3x3.

Statement 2: Data-dependent weight initialization procedures were introduced before Batch Normalization.

This statement is true. Data-dependent weight initialization procedures, such as Xavier or He initialization, were introduced before Batch Normalization.

Final answer: C

\*\*\*\*\*



Question 23:

'What is the rank of the following matrix?  $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ ', A. '0', B. '1', C. '2', D. '3'

=====

Answer Option 1:

The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. In the given matrix A, all rows and columns are linearly dependent, as they are scalar multiples of each other. Therefore, the rank of matrix A is 1. Final answer: B

=====

Answer Option 2:

The given matrix  $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  is a matrix of all ones. Since all rows of the matrix are linearly dependent, the rank of this matrix is 1. Final answer: B

=====

Answer Option 3:

The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. In the given matrix A, all the rows and columns are linearly dependent on each other. Therefore, the rank of matrix A is 1. Final answer: B

=====

Answer Option 4:

The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. In the given matrix A, all rows and columns are linearly dependent as they are multiples of each other.

Hence, the rank of matrix A is 1. Final answer: B

=====

Answer Option 5:

The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. Given the matrix  $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ , all the rows and columns are linearly dependent, meaning they can be expressed as a linear combination of each other. Hence, the rank of matrix A is 1. Final answer: B

=====

Answer Option 6:

The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. In the given matrix  $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ , all rows and columns are linearly dependent, meaning they can be expressed as a linear combination of each other. Therefore, the rank of matrix A is 1. Final answer: B

=====

Answer Option 7:

The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. In this case, the matrix  $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  has all rows and columns identical, and they are linearly dependent. Therefore, the rank of this matrix is 1. Final answer: B

=====

Answer Option 8:

The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. The given matrix  $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  has all rows and columns that are linearly dependent, as each row is a scalar multiple of the other rows. Therefore, the rank of the matrix is 1. Final answer: B

=====

Answer Option 9:

The given matrix  $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  is a 3x3 matrix. The rank of a matrix is the maximum number of linearly independent rows or columns in the matrix. In this case, all the rows and columns in the matrix are linearly dependent since each row is a constant multiple of any other row, and each column is a constant multiple of any other column. Therefore, the rank of matrix A is 1. Final answer: B

=====

Question 24:

'Statement 1] Density estimation (using say, the kernel density estimator) can be used to perform classification. Statement 2] The correspondence between logistic regression and Gaussian Naive Bayes (with identity class covariances) means that there is a one-to-one correspondence between the parameters of the two classifiers.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. Density estimation, such as using the kernel density estimator, is typically used for estimating the underlying probability density function of a dataset, not for classification.

Statement 2 is false. While there are connections between logistic regression and Gaussian Naive Bayes, they are not equivalent and do not have a one-to-one correspondence between their parameters.

Final answer: C

=====

Answer Option 2:

Statement 1: Density estimation using the kernel density estimator is not typically used for classification tasks. It is more commonly used for estimating the probability density function of a continuous random variable. False.

Statement 2: Logistic regression and Gaussian Naive Bayes are different classifiers and do not have a one-to-one correspondence between their parameters. False.

Final answer: C (True, False)

=====

Answer Option 3:

Statement 1 is false. Density estimation using the kernel density estimator is typically used for estimating probability density functions and not for performing classification.

Statement 2 is false. While both logistic regression and Gaussian Naive Bayes are classifiers, there is no one-to-one correspondence between their parameters. They make different assumptions about the underlying data distribution and have different parameterizations.

Final answer: D

=====

Answer Option 4:

Statement 1 is false. Density estimation using kernel density estimator is typically used for estimating probability distribution functions, not for classification.

Statement 2 is false. While logistic regression and Gaussian Naive Bayes are both classification algorithms, there is no direct one-to-one correspondence between their parameters. They are based on different assumptions and have different parameterizations.

Final answer: C

=====

Answer Option 5:

Statement 1: Density estimation using the kernel density estimator is typically used for estimating the probability density function of a continuous variable and is not directly used for classification tasks.

Therefore, the statement is False.

Statement 2: Logistic regression and Gaussian Naive Bayes are two different classification algorithms and have different underlying assumptions and parameter estimation procedures. Therefore, there is no one-to-one correspondence between the parameters of the two classifiers. The statement is False.

Final answer: D

=====

Answer Option 6:

Statement 1 is false. Density estimation, such as using the kernel density estimator, is typically used for estimating probability densities or generating samples, not for classification tasks.

Statement 2 is false. While logistic regression and Gaussian Naive Bayes share some similarities in their underlying assumptions and probabilistic interpretations, there is not a one-to-one correspondence between the parameters of the two classifiers.

Final answer: C (False, False)

=====

Answer Option 7:

Statement 1: Density estimation using the kernel density estimator can be used to perform classification. This statement is False. While density estimation techniques like the kernel density estimator can give probabilities, they are not directly used for classification tasks.

Statement 2: There is a one-to-one correspondence between the parameters of logistic regression and Gaussian Naive Bayes (with identity class covariances). This statement is False. Logistic regression and Gaussian Naive Bayes are different algorithms with different assumptions and parameterizations.

Final answer: C

=====

Answer Option 8:

Statement 1: Density estimation using the kernel density estimator is not typically used for classification. It is commonly employed for estimating the probability density function of a given dataset. Therefore, statement 1 is false.

Statement 2: Logistic regression and Gaussian Naive Bayes are different classification algorithms. While both models are based on the Bayes theorem, they make different assumptions about the data distribution. Logistic regression assumes a linear relationship between the features and the log-odds of the classes, while Gaussian Naive Bayes assumes that the features are conditionally independent given the class labels. Therefore, statement 2 is false.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: Density estimation using methods like the kernel density estimator is not typically used for classification. Density estimation is used to estimate the underlying probability distribution of the data, while classification involves assigning data points to predefined classes.

Statement 2: While both logistic regression and Gaussian Naive Bayes are classifiers, there is no one-to-one correspondence between their parameters. Logistic regression models the probability of a binary outcome using a logistic function, while Gaussian Naive Bayes assumes that the features are conditionally independent given the class.

Final answer: C ('True, False')

\*\*\*\*\*

Question 25:

'Suppose we would like to perform clustering on spatial data such as the geometrical locations of houses. We wish to produce clusters of many different sizes and shapes. Which of the following methods is the most appropriate?', A. 'Decision Trees', B. 'Density-based clustering', C. 'Model-based clustering', D. 'K-means clustering'

=====

Answer Option 1:

If we want to produce clusters of different sizes and shapes, the most appropriate method would be density-based clustering. Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), are capable of identifying clusters of varying densities and shapes based on the density of data points in the vicinity. Decision trees, model-based clustering, and k-means clustering are not specifically designed for handling clusters of different sizes and shapes. Final answer: B

\*\*\*\*\*

Answer Option 2:

In order to produce clusters of different sizes and shapes, the most appropriate method would be density-based clustering. Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), are able to detect clusters of varying densities and shapes based on the density of the data points. This makes them suitable for clustering spatial data like the geometrical locations of houses. Decision trees, model-based clustering, and k-means clustering do not have the flexibility to create clusters of varying sizes and shapes. Final answer: B

\*\*\*\*\*

Answer Option 3:

For clustering spatial data with the objective of producing clusters of different sizes and shapes, the most appropriate method would be density-based clustering. Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), are well-suited for identifying clusters of varying densities and shapes in spatial data. Decision trees, model-based clustering, and k-means clustering are not specifically designed for this type of task. Final answer: B

\*\*\*\*\*

Answer Option 4:

When we want to produce clusters of different sizes and shapes in spatial data, density-based clustering is the most appropriate method. Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), are capable of identifying clusters based on density and can capture clusters of different shapes and sizes. Decision trees, model-based clustering, and k-means clustering do not have the same capability to handle clusters of varying shapes and sizes in spatial data. Final answer: B

\*\*\*\*\*

Answer Option 5:

For clustering spatial data with the goal of producing clusters of various sizes and shapes, the most appropriate method would be density-based clustering. Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), are capable of identifying clusters of arbitrary shapes and sizes based on the density of data points. Decision trees, model-based clustering, and k-means clustering are not specifically designed for handling spatial data and may not be as effective in producing clusters of different sizes and shapes. Final answer: B

\*\*\*\*\*

Answer Option 6:

Density-based clustering is the most appropriate method for clustering spatial data to produce clusters of different sizes and shapes. This is because density-based clustering algorithms, such as DBSCAN, can identify clusters based on the density of data points rather than assuming a specific shape or number of clusters. Decision trees, model-based clustering, and k-means clustering are not specifically designed for clustering spatial data with varying sizes and shapes. Final Answer: B

\*\*\*\*\*

Answer Option 7:

If we want to produce clusters of different sizes and shapes for spatial data, the most appropriate method would be density-based clustering. Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), are well-suited for discovering clusters of arbitrary shapes and sizes based on the density of data points. Decision Trees, Model-based clustering, and K-means clustering are not specifically designed for this purpose. Final answer: B

\*\*\*\*\*

Answer Option 8:

For clustering spatial data where clusters of different sizes and shapes are desired, density-based clustering would be the most appropriate method. Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), can effectively capture clusters of arbitrary shapes and sizes based on the density of points. Decision trees, model-based clustering, and k-means clustering are not specifically designed for clustering spatial data with varying shapes and sizes. Final answer: B

\*\*\*\*\*

Answer Option 9:

If we want to produce clusters of different sizes and shapes for spatial data, the most appropriate method would be density-based clustering. Density-based clustering algorithms like DBSCAN (Density-Based Spatial Clustering of Applications with Noise) are well-suited for identifying clusters of varying densities and shapes in spatial data. Decision trees, model-based clustering, and k-means clustering are not specifically designed for clustering spatial data with varying cluster sizes and shapes. Final answer: B

\*\*\*\*\*

Question 26:

'Statement 1| In AdaBoost weights of the misclassified examples go up by the same multiplicative factor. Statement 2| In AdaBoost, weighted training error  $e_t$  of the  $t$ th weak classifier on training data with weights  $D_t$  tends to increase as a function of  $t$ .' A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: In AdaBoost, the weights of the misclassified examples are typically increased by the same multiplicative factor, which is true. Statement 2: In AdaBoost, the weighted training error of the  $t$ th weak classifier tends to decrease as a function of  $t$ , not increase. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. In AdaBoost, the weights of misclassified examples are increased by the same multiplicative factor to give them more importance in subsequent iterations.

Statement 2 is false. In AdaBoost, the weighted training error of the weak classifier tends to decrease as a function of  $t$ , not increase. The goal of AdaBoost is to minimize the weighted training error.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1: In AdaBoost, weights of the misclassified examples are increased by the same multiplicative factor, which is higher for the classifiers with lower training error. This statement is true.

Statement 2: In AdaBoost, the weighted training error of the  $t$ th weak classifier tends to decrease as a function of  $t$ , not increase. This statement is false.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1: In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor. This statement is true.

Statement 2: In AdaBoost, weighted training error  $e_t$  of the  $t$ th weak classifier on training data with weights  $D_t$  tends to increase as a function of  $t$ . This statement is false.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is true. In AdaBoost, the weights of the misclassified examples are increased by the same multiplicative factor to give them more importance in the subsequent iterations.

Statement 2 is false. In AdaBoost, the weighted training error of the weak classifier tends to decrease as a function of  $t$ , not increase. The objective of AdaBoost is to iteratively train weak classifiers to reduce the overall error on the training data. Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is true. In AdaBoost, the weights of misclassified examples are increased by the same multiplicative factor to give them more importance in subsequent iterations.

Statement 2 is false. In AdaBoost, the weighted training error of the  $t$ th weak classifier tends to decrease as a function of  $t$ . The goal of AdaBoost is to iteratively improve the model's performance by giving more emphasis on misclassified examples in each iteration. Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor. This statement is true.

Statement 2: In AdaBoost, the weighted training error  $e_t$  of the  $t$ th weak classifier on training data with weights  $D_t$  tends to increase as a function of  $t$ . This statement is false. The weighted training error tends to decrease as AdaBoost focuses more on the misclassified examples.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: In AdaBoost, the weights of the misclassified examples are increased, but not necessarily by the same multiplicative factor. The weights are adjusted based on the error rate of the weak classifier.

Statement 2: In AdaBoost, the weighted training error of the  $t$ th weak classifier tends to decrease as a function of  $t$ . This is because AdaBoost assigns higher weights to the misclassified examples, making them more influential in subsequent iterations.

Based on these explanations, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 9:

Statement 1: In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor.

This statement is true. In AdaBoost, the weights of the misclassified examples are increased by the same multiplicative factor to give them higher importance in the next iteration.

Statement 2: In AdaBoost, weighted training error  $e_t$  of the  $t$ th weak classifier on training data with weights  $D_t$  tends to increase as a function of  $t$ .

This statement is false. In AdaBoost, the weighted training error of the  $t$ th weak classifier tends to decrease as a function of  $t$ , as the subsequent weak classifiers are trained to focus more on the samples that were misclassified by the previous weak classifiers.

Final Answer: C

\*\*\*\*\*

Question 27:

'MLE estimates are often undesirable because', A. 'they are biased', B. 'they have high variance', C. 'they are not consistent estimators', D. 'None of the above'

=====

Answer Option 1:

MLE (Maximum Likelihood Estimation) estimates are generally desirable, as they have several desirable properties, such as consistency and asymptotic efficiency. Therefore, the correct answer is D. "None of the above."

=====

Answer Option 2:

MLE (Maximum Likelihood Estimation) estimates are generally considered desirable because they have several desirable properties. They are asymptotically unbiased, consistent estimators, and they achieve the Cramer-Rao lower bound, which means they are efficient. Therefore, the correct answer is D. None of the above.

=====

Answer Option 3:

MLE (Maximum Likelihood Estimation) estimates are generally desirable because they have desirable statistical properties, such as consistency and asymptotic efficiency. They are not biased, have low variance (in large sample sizes), and are consistent estimators. Therefore, the correct answer is D. None of the above.

=====

Answer Option 4:

MLE (Maximum Likelihood Estimation) estimates are not inherently biased or have high variance. However, they can sometimes have limitations such as not being consistent estimators when the sample size is small or when the distribution assumptions are violated. Therefore, the correct answer is C. 'they are not consistent estimators'.

=====

Answer Option 5:

MLE (maximum likelihood estimation) estimates are desirable because they have several desirable properties such as consistency, efficiency, and asymptotic normality. They are not biased, have low variance, and are consistent estimators when certain conditions are met. Therefore, the correct answer is D. None of the above.

=====

Answer Option 6:

MLE (Maximum Likelihood Estimation) estimates are generally desirable in many cases because they have several desirable properties, such as consistency and asymptotic efficiency. However, one potential drawback of MLE estimates is that they can have high variance, especially when the sample size is small. Therefore, the correct answer is B: 'they have high variance'.

=====

Answer Option 7:

Maximum Likelihood Estimation (MLE) is a commonly used method for estimating model parameters. MLE estimates are not inherently biased or have high variance. In fact, MLE estimates are asymptotically consistent, meaning that as the sample size increases, the estimates converge to the true parameter values. Therefore, the correct answer is D. None of the above.

=====

Answer Option 8:

MLE (Maximum Likelihood Estimation) estimates are generally desirable because they have several desirable statistical properties. They are asymptotically unbiased, meaning that as the sample size increases, the estimate converges to the true parameter value. They are also consistent, meaning that as the sample size increases, the estimate becomes more accurate. Additionally, MLE estimates are efficient and achieve the Cramer-Rao lower bound. Therefore, the correct answer is D. 'None of the above'.

=====

Answer Option 9:

MLE (Maximum Likelihood Estimation) estimates are often desirable because they have good statistical properties. They are consistent estimators, meaning that as the sample size increases, the estimated parameters converge to the true population parameters. MLE estimates are also asymptotically efficient, meaning that they achieve the lowest possible variance among all consistent estimators. Therefore, the correct answer is D. 'None of the above'.

=====

Question 28:

'Computational complexity of Gradient descent is,', A. 'linear in D', B. 'linear in N', C. 'polynomial in D', D. 'dependent on the number of iterations'

=====

Answer Option 1:

The computational complexity of Gradient descent is dependent on the number of iterations. Final answer: D

\*\*\*\*\*

Answer Option 2:

The computational complexity of Gradient Descent is dependent on the number of iterations, not on the dimensionality of the problem (D) or the number of training examples (N). Final answer: D

\*\*\*\*\*

Answer Option 3:

The computational complexity of Gradient descent depends on the number of iterations, which is determined by factors such as the learning rate and convergence criteria. It is not directly related to the dimensionality of the data (D) or the number of samples (N). Final answer: D

\*\*\*\*\*

Answer Option 4:

The computational complexity of Gradient Descent is dependent on the number of iterations, not on the dimensions of the dataset (D) or the number of samples (N). The time complexity is typically linear in the number of iterations. Final answer: D

\*\*\*\*\*

Answer Option 5:

The computational complexity of gradient descent is dependent on the number of iterations, as the algorithm iteratively updates the model parameters. It is not specifically linear in D (the number of features) or N (the number of data points) and it is not polynomial in D. Final answer: D

\*\*\*\*\*

Answer Option 6:

The computational complexity of Gradient Descent algorithm is dependent on the number of iterations. It does not have a direct relationship with the dimensionality of the input features (D) or the number of samples (N). Final answer: D

\*\*\*\*\*

Answer Option 7:

The computational complexity of gradient descent is typically linear in the number of iterations, which is often denoted as  $O(\text{iterations})$ . It is not directly dependent on the dimensionality of the problem (D) or the size of the dataset (N). Final answer: D

\*\*\*\*\*

Answer Option 8:

The computational complexity of Gradient Descent is dependent on the number of iterations. It does not have a fixed complexity with respect to the number of features (D) or the number of data points (N).

Final answer: D

\*\*\*\*\*

Answer Option 9:

The computational complexity of gradient descent is dependent on the number of iterations, as indicated in option D. The other options, A, B, and C, do not accurately represent the computational complexity of gradient descent. Final answer: D.

\*\*\*\*\*



Question 29:

'Averaging the output of multiple decision trees helps \_', A. 'Increase bias', B. 'Decrease bias', C. 'Increase variance', D. 'Decrease variance'

=====

Answer Option 1:

Averaging the output of multiple decision trees helps decrease variance. By combining the predictions from multiple trees, the variability in the predictions is reduced, resulting in a more stable and accurate overall prediction. Final answer: D

\*\*\*\*\*

Answer Option 2:

Averaging the output of multiple decision trees helps decrease variance. By combining the predictions of multiple trees, the randomness and error associated with individual trees are reduced, leading to more stable and reliable predictions. Final answer: D

\*\*\*\*\*

Answer Option 3:

Averaging the output of multiple decision trees helps decrease variance. By combining the predictions of multiple trees, the overall variance of the model is reduced, leading to a more stable and reliable prediction. Final answer: D

\*\*\*\*\*

Answer Option 4:

Averaging the output of multiple decision trees helps to decrease variance. By combining the predictions of multiple decision trees, the overall model becomes more robust and less sensitive to individual tree predictions, reducing the variance in the model's predictions. Final answer: D

\*\*\*\*\*

Answer Option 5:

Averaging the output of multiple decision trees helps decrease variance. This is because individual decision trees may have high variance and can overfit the training data. By averaging their predictions, the overall variance of the model decreases, leading to improved generalization performance. Final answer: D

\*\*\*\*\*

Answer Option 6:

Averaging the output of multiple decision trees helps decrease variance. By combining the predictions of multiple trees, the overall variance of the model decreases, leading to improved generalization and reducing the risk of overfitting. Final answer: D

\*\*\*\*\*

Answer Option 7:

Averaging the output of multiple decision trees helps decrease variance. When decision trees are combined through techniques such as bagging or random forests, the variance of the model's predictions tends to decrease, leading to more stable and reliable results. Final answer: D

\*\*\*\*\*

Answer Option 8:

Averaging the output of multiple decision trees helps decrease variance. By combining the predictions of multiple trees, the model becomes more robust and less sensitive to individual trees' predictions, resulting in a lower variance. Final answer: D

\*\*\*\*\*

Answer Option 9:

Averaging the output of multiple decision trees helps decrease variance. By combining the predictions of multiple trees, the overall predictions become more stable and less prone to being influenced by the specific characteristics of individual trees, thereby reducing variance. Final answer: D

\*\*\*\*\*

Question 30:

'The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during', A. 'Best-subset selection', B. 'Forward stepwise selection', C. 'Forward stage wise selection', D. 'All of the above'

=====

Answer Option 1:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during all of the following methods:

Best-subset selection, forward stepwise selection, and forward stage wise selection. Final Answer: D

\*\*\*\*\*

Answer Option 2:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during all three methods:

best-subset selection, forward stepwise selection, and forward stage wise selection. Final answer: D

\*\*\*\*\*

Answer Option 3:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during all of the mentioned

techniques: best-subset selection, forward stepwise selection, and forward stage wise selection. Final answer: D

\*\*\*\*\*

Answer Option 4:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during all of the above methods:

best-subset selection, forward stepwise selection, and forward stage-wise selection. Final answer: D

\*\*\*\*\*

Answer Option 5:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during 'Best-subset selection',

'Forward stepwise selection', and 'Forward stage wise selection'. Final answer: D

\*\*\*\*\*

Answer Option 6:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during all of the mentioned methods:

best-subset selection, forward stepwise selection, and forward stage wise selection. Final answer: D

\*\*\*\*\*

Answer Option 7:

The process of identifying the subset of features during best-subset selection, forward stepwise selection, and forward stage wise selection can lead to different models compared to the model obtained by applying linear regression on the identified subset of features. Therefore, the correct answer is D - 'All of the above'.

\*\*\*\*\*

Answer Option 8:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during forward stepwise selection,

forward stage-wise selection, and best-subset selection. Therefore, the correct answer is D. 'All of the above'.

\*\*\*\*\*

Answer Option 9:

The model obtained by applying linear regression on the identified subset of features may differ from the model obtained at the end of the process of identifying the subset during Best-subset selection,

Forward stepwise selection, and Forward stage wise selection. Therefore, the correct answer is D. 'All of the above'.

\*\*\*\*\*

Question 31:

'Neural networks:', A. 'Optimize a convex objective function', B. 'Can only be trained with stochastic gradient descent', C. 'Can use a mix of different activation functions', D. 'None of the above'

=====

Answer Option 1:

Neural networks can use a mix of different activation functions, such as sigmoid, ReLU, or tanh, to introduce non-linearity and model complex relationships in the data. Therefore, the correct answer is C.

\*\*\*\*\*

Answer Option 2:

Neural networks do not necessarily optimize a convex objective function. They can use various optimization algorithms and are not limited to stochastic gradient descent. Furthermore, neural networks can indeed utilize a mix of different activation functions. Final answer: C.

\*\*\*\*\*

Answer Option 3:

Neural networks do not optimize a convex objective function (A). They can be trained with various optimization algorithms, not just stochastic gradient descent (B). Neural networks can indeed use a mix of different activation functions (C). Final answer: C.

\*\*\*\*\*

Answer Option 4:

Neural networks do not optimize a convex objective function. They can be trained with various optimization algorithms, including stochastic gradient descent, but they are not limited to it. Neural networks can indeed use a mix of different activation functions in their layers. Final answer: C

\*\*\*\*\*

Answer Option 5:

Neural networks do not necessarily optimize a convex objective function. They can use a variety of optimization algorithms, and the convexity of the objective function depends on the specific architecture and problem being solved. Neural networks can also use a mix of different activation functions, such as sigmoid, ReLU, and tanh. Final answer: C

\*\*\*\*\*

Answer Option 6:

Neural networks do not necessarily optimize a convex objective function. They can be trained with various optimization algorithms, such as stochastic gradient descent, but they are not limited to it. Neural networks can indeed use a mix of different activation functions, such as sigmoid, ReLU, and softmax. Final answer: C

\*\*\*\*\*

Answer Option 7:

Neural networks do not necessarily optimize a convex objective function, as the optimization problem can be non-convex. They also do not require the use of stochastic gradient descent as the only training algorithm, although it is commonly used. Neural networks can indeed use a mix of different activation functions, depending on the architecture and problem at hand. Final answer: C

\*\*\*\*\*

Answer Option 8:

Neural networks do not necessarily optimize a convex objective function. They are typically trained using various optimization algorithms, including stochastic gradient descent, but they are not limited to this method alone. Neural networks can indeed use a mix of different activation functions, such as sigmoid, ReLU, and tanh, depending on the architecture and requirements of the model. Final answer: C

\*\*\*\*\*

Answer Option 9:

Neural networks do not necessarily optimize a convex objective function as the shape of the objective function can vary depending on the architecture and complexity of the network. Neural networks can be trained with different optimization algorithms, not just stochastic gradient descent. Neural networks can indeed use a mix of different activation functions, such as sigmoid, ReLU, and tanh, to introduce non-linearity in the network. Final answer: C

\*\*\*\*\*

Question 32:

'Say the incidence of a disease D is about 5 cases per 100 people (i.e.,  $P(D) = 0.05$ ). Let Boolean random variable D mean a patient "has disease D" and let Boolean random variable TP stand for "tests positive." Tests for disease D are known to be very accurate in the sense that the probability of testing positive when you have the disease is 0.99, and the probability of testing negative when you do not have the disease is 0.97. What is  $P(TP)$ , the prior probability of testing positive.', A. '0.0368', B. '0.473', C. '0.078', D. 'None of the above'

=====

Answer Option 1:

To calculate the prior probability of testing positive ( $P(TP)$ ), we need to consider both the incidence of the disease and the accuracy of the test.

Given:

$P(D) = 0.05$  (incidence of the disease)

$P(TP|D) = 0.99$  (probability of testing positive given the disease)

$P(TP|\neg D) = 0.03$  (probability of testing positive given no disease)

We can use Bayes' theorem to calculate  $P(TP)$ :

$P(TP) = P(TP|D) * P(D) + P(TP|\neg D) * P(\neg D)$

$P(\neg D) = 1 - P(D) = 1 - 0.05 = 0.95$

$P(TP) = 0.99 * 0.05 + 0.03 * 0.95$

$= 0.0495 + 0.0285$

$= 0.078$

Therefore, the correct answer is C. '0.078'.

\*\*\*\*\*

Answer Option 2:

The probability of testing positive,  $P(TP)$ , can be calculated using Bayes' theorem.

$P(TP) = P(TP|D) * P(D) + P(TP|\neg D) * P(\neg D)$

Given that the probability of testing positive when you have the disease is 0.99 ( $P(TP|D) = 0.99$ ) and the incidence of the disease is 0.05 ( $P(D) = 0.05$ ), we can calculate:

$P(TP) = 0.99 * 0.05 + (1 - 0.97) * (1 - 0.05)$

$= 0.0495 + (0.03 * 0.95)$

$= 0.0495 + 0.0285$

$= 0.078$

Therefore, the prior probability of testing positive,  $P(TP)$ , is 0.078. Final answer: C.

\*\*\*\*\*

Answer Option 3:

To find the probability of testing positive ( $P(TP)$ ), we need to consider both the cases where the patient has the disease (D) and where the patient does not have the disease ( $\neg D$ ).

Let's calculate the probability of testing positive when the patient has the disease:

$P(TP|D) = 0.99$  (given)

Now, let's calculate the probability of testing positive when the patient does not have the disease:

$P(TP|\neg D) = 1 - P(\text{Negative}|\neg D) = 1 - 0.97 = 0.03$

Finally, we can calculate  $P(TP)$  using the law of total probability:

$P(TP) = P(D) * P(TP|D) + P(\neg D) * P(TP|\neg D)$

$= 0.05 * 0.99 + 0.95 * 0.03$

$= 0.0495 + 0.0285$

$= 0.078$

Therefore, the prior probability of testing positive ( $P(TP)$ ) is 0.078. Final answer: C

\*\*\*\*\*

Answer Option 4:

To find the probability of testing positive, we can apply Bayes' theorem and use the law of total probability.

$P(TP) = P(TP | D) * P(D) + P(TP | \text{not } D) * P(\text{not } D)$

Given that the probability of testing positive when you have the disease is 0.99 ( $P(TP | D) = 0.99$ ) and the incidence of the disease is 0.05 ( $P(D) = 0.05$ ), we can calculate  $P(TP | \text{not } D)$  as  $1 - P(\text{not } TP | \text{not } D)$ ,

where  $P(\text{not } TP | \text{not } D)$  is the probability of testing negative when you do not have the disease. From the given information,  $P(\text{not } TP | \text{not } D) = 0.97$ .

$P(TP) = 0.99 * 0.05 + (1 - 0.97) * (1 - 0.05)$

$= 0.0495 + 0.015$

$= 0.0645$

Therefore, the prior probability of testing positive is 0.0645.

Final answer: D. 'None of the above'

\*\*\*\*\*

Answer Option 5:

To find the probability of testing positive ( $P(TP)$ ), we need to consider both the cases when a patient has the disease (D) and when they do not have the disease (D').

$P(TP|D) = 0.99$  (Probability of testing positive given that the patient has the disease)

$P(TP|D') = 1 - 0.97 = 0.03$  (Probability of testing positive given that the patient does not have the disease)

Now, we can use the law of total probability to calculate  $P(TP)$ :

$P(TP) = P(TP|D) * P(D) + P(TP|D') * P(D')$

$P(D) = 0.05$  (Given incidence of the disease)

$P(D') = 1 - P(D) = 0.95$

$P(TP) = 0.99 * 0.05 + 0.03 * 0.95$

$P(TP) = 0.0495 + 0.0285$

$P(TP) = 0.078$

Therefore, the prior probability of testing positive ( $P(TP)$ ) is 0.078. Final Answer: C

\*\*\*\*\*

Answer Option 6:

To find the prior probability of testing positive ( $P(TP)$ ), we need to consider two scenarios:

1) The patient has the disease and tests positive ( $TP | D$ )

2) The patient does not have the disease and tests positive ( $TP | \sim D$ )

Using the information given:

$P(D) = 0.05$  (incidence of disease D)

$P(TP | D) = 0.99$  (probability of testing positive given that the patient has the disease)

$P(\sim TP | \sim D) = 0.97$  (probability of testing negative given that the patient does not have the disease)

Now, we can calculate the probability of testing positive using the law of total probability:

$P(TP) = P(TP | D) * P(D) + P(TP | \sim D) * P(\sim D)$

$P(\sim D) = 1 - P(D) = 1 - 0.05 = 0.95$

$P(TP) = 0.99 * 0.05 + 0.97 * 0.95$

$= 0.0495 + 0.9215$

$= 0.971$

The prior probability of testing positive ( $P(TP)$ ) is 0.971.

Final answer: D. 'None of the above'

\*\*\*\*\*

Answer Option 7:

To calculate the prior probability of testing positive, we need to consider two scenarios:

1. Testing positive given that the patient has the disease ( $P(TP|D)$ ): The probability of testing positive when you have the disease is given as 0.99.

2. Testing positive given that the patient does not have the disease ( $P(TP|\sim D)$ ): The probability of testing negative when you do not have the disease is given as 0.97. Therefore, the probability of testing positive in this scenario would be  $1 - P(TP|\sim D) = 1 - 0.97 = 0.03$ .

Now, we can calculate the prior probability of testing positive ( $P(TP)$ ) using the law of total probability:

$P(TP) = P(D) * P(TP|D) + P(\sim D) * P(TP|\sim D)$

$P(D) = 0.05$  (given)

$P(\sim D) = 1 - P(D) = 1 - 0.05 = 0.95$

$P(TP) = 0.05 * 0.99 + 0.95 * 0.03$

$P(TP) = 0.0495 + 0.0285$

$P(TP) = 0.078$

Therefore, the prior probability of testing positive ( $P(TP)$ ) is 0.078.

Final answer: C

\*\*\*\*\*

Answer Option 8:

We are given that the incidence of disease D is 0.05 ( $P(D) = 0.05$ ). We are also given that the probability of testing positive when you have the disease is 0.99 ( $P(TP|D) = 0.99$ ) and the probability of testing negative when you do not have the disease is 0.97 ( $P(\sim TP|\sim D) = 0.97$ ).

To find the prior probability of testing positive ( $P(TP)$ ), we can use Bayes' theorem:

$P(TP) = P(TP|D) * P(D) + P(TP|\sim D) * P(\sim D)$

$P(TP) = 0.99 * 0.05 + (1 - 0.97) * (1 - 0.05)$

$P(TP) = 0.0495 + 0.015$

$P(TP) = 0.0645$

Therefore, the prior probability of testing positive is 0.0645.

Final answer: D. 'None of the above'

\*\*\*\*\*

Answer Option 9:

To find the prior probability of testing positive, we need to consider the probability of two events:

1. The probability of having the disease (D), which is given as  $P(D) = 0.05$ .
2. The probability of testing positive (TP) given that the patient has the disease, which is given as  $P(TP | D) = 0.99$ .

Using the law of total probability, we can calculate  $P(TP)$  as follows:

$$P(TP) = P(TP | D) * P(D) + P(TP | \text{not } D) * P(\text{not } D)$$

Since  $P(\text{not } D)$  is equal to  $1 - P(D)$ , we have:

$$P(TP) = 0.99 * 0.05 + P(TP | \text{not } D) * (1 - 0.05)$$

Given that the probability of testing negative (TN) when you do not have the disease is 0.97, we can calculate  $P(TP | \text{not } D)$  as  $1 - P(TN | \text{not } D)$ , which is  $1 - 0.97 = 0.03$ .

Now we can substitute the values into the equation:

$$P(TP) = 0.99 * 0.05 + 0.03 * (1 - 0.05)$$

$$P(TP) = 0.0495 + 0.03 * 0.95$$

$$P(TP) = 0.0495 + 0.0285$$

$$P(TP) = 0.078$$

Therefore, the prior probability of testing positive ( $P(TP)$ ) is 0.078. Final answer: C

\*\*\*\*\*

Question 33:

'Statement 1] After mapped into feature space Q through a radial basis kernel function, 1-NN using unweighted Euclidean distance may be able to achieve better classification performance than in original space (though we can't guarantee this). Statement 2] The VC dimension of a Perceptron is smaller than the VC dimension of a simple linear SVM.' A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. Mapping the data into a higher-dimensional feature space using a radial basis kernel function can potentially improve the performance of 1-NN classification.

Statement 2 is false. The VC dimension of a Perceptron is actually larger than the VC dimension of a simple linear SVM.

Final answer: C

=====

Answer Option 2:

Statement 1: After mapping into feature space Q through a radial basis kernel function, it is possible that 1-NN using unweighted Euclidean distance may achieve better classification performance than in the original space, although this is not guaranteed. Therefore, Statement 1 is true.

Statement 2: The VC dimension of a perceptron is actually larger than the VC dimension of a simple linear SVM. Therefore, Statement 2 is false.

Final answer: C

=====

Answer Option 3:

The correct answer is D. 'False, True'.

Statement 1 is false because mapping data into a higher-dimensional feature space Q using a radial basis kernel function may help improve the classification performance of certain algorithms, but it does not guarantee that 1-NN using unweighted Euclidean distance will achieve better performance than in the original space.

Statement 2 is true because the VC (Vapnik-Chervonenkis) dimension of a Perceptron is smaller than the VC dimension of a simple linear Support Vector Machine (SVM). The VC dimension is a measure of the capacity or complexity of a model in terms of its ability to shatter data points.

=====

Answer Option 4:

Statement 1: After being mapped into feature space Q through a radial basis kernel function, 1-NN using unweighted Euclidean distance may or may not achieve better classification performance than in the original space. We cannot guarantee this, so the statement is not necessarily true.

Statement 2: The VC (Vapnik-Chervonenkis) dimension of a Perceptron is smaller than the VC dimension of a simple linear SVM. This statement is false. In fact, the VC dimension of a simple linear SVM is typically smaller than that of a Perceptron.

Final answer: C

=====

Answer Option 5:

Statement 1 is True. Mapping data into a higher-dimensional feature space using a radial basis kernel function can sometimes make the data more separable, potentially leading to better classification performance using 1-NN with unweighted Euclidean distance.

Statement 2 is False. The VC (Vapnik-Chervonenkis) dimension of a Perceptron is larger than the VC dimension of a simple linear SVM. The VC dimension of a Perceptron is equal to the number of weights plus one, while the VC dimension of a linear SVM is equal to the number of features plus one.

Final answer: C

=====

Answer Option 6:

Statement 1: After mapping into feature space Q through a radial basis kernel function, 1-NN using unweighted Euclidean distance may be able to achieve better classification performance than in the original space (though we can't guarantee this).

This statement is true. The radial basis kernel function can map the data into a higher-dimensional feature space where it becomes easier to separate the classes. In some cases, using 1-NN with unweighted Euclidean distance in this feature space can lead to better classification performance.

Statement 2: The VC dimension of a Perceptron is smaller than the VC dimension of a simple linear SVM.

This statement is false. The VC dimension of a Perceptron is infinite, meaning it can shatter any finite set of points. On the other hand, the VC dimension of a simple linear SVM is finite and depends on the dimensionality of the feature space.

Final answer: C

=====

Answer Option 7:

Statement 1: After being mapped into feature space Q through a radial basis kernel function, it is possible that 1-NN using unweighted Euclidean distance can achieve better classification performance than in the original space, although there is no guarantee.

Statement 2: The VC dimension of a Perceptron is not necessarily smaller than the VC dimension of a simple linear SVM.

Final answer: C (True, False)

=====

Answer Option 8:

Statement 1: After mapping into feature space  $Q$  through a radial basis kernel function, it is possible that 1-NN using unweighted Euclidean distance could achieve better classification performance than in the original space, although it is not guaranteed.

Statement 2: The VC dimension of a Perceptron is not necessarily smaller than the VC dimension of a simple linear SVM.

Final Answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: After mapping into feature space  $Q$  through a radial basis kernel function, 1-NN using unweighted Euclidean distance may be able to achieve better classification performance than in the original space (though we can't guarantee this).

This statement is true. The radial basis kernel function can map the data into a higher-dimensional feature space where it might become easier to classify the data. In this higher-dimensional feature space, the 1-NN algorithm using unweighted Euclidean distance can potentially achieve better classification performance.

Statement 2: The VC dimension of a Perceptron is smaller than the VC dimension of a simple linear SVM.

This statement is false. The VC dimension of a Perceptron is known to be infinite, meaning that a Perceptron can shatter any finite set of points. On the other hand, the VC dimension of a simple linear SVM is finite, depending on the number of features used. Thus, the VC dimension of a Perceptron is not smaller than that of a simple linear SVM.

Final answer: C

\*\*\*\*\*



Question 34:

'The disadvantage of Grid search is', A. 'It can not be applied to non-differentiable functions.', B. 'It can not be applied to non-continuous functions.', C. 'It is hard to implement.', D. 'It runs reasonably slow for multiple linear regression.'

=====

Answer Option 1:

The disadvantage of Grid search is that it can be computationally expensive and time-consuming, especially when dealing with a large number of hyperparameters and a large parameter space. Final answer:

It runs reasonably slow for multiple linear regression. (D)

\*\*\*\*\*

Answer Option 2:

The disadvantage of Grid search is not that it cannot be applied to non-differentiable or non-continuous functions. Rather, the disadvantage is that it can be computationally expensive and time-consuming, especially for models with a large number of hyperparameters or a large search space. Final answer: D

\*\*\*\*\*

Answer Option 3:

The disadvantage of Grid search is that it can be computationally expensive and time-consuming, especially when dealing with a large number of hyperparameters and a wide range of values for each hyperparameter. This can make it slow to run, especially for complex models or large datasets. Final answer: D

\*\*\*\*\*

Answer Option 4:

The disadvantage of Grid search is that it can be computationally expensive and time-consuming, especially when dealing with a large number of hyperparameters and a wide range of possible values. This can make it slow, especially for complex models or large datasets. Final answer: D

\*\*\*\*\*

Answer Option 5:

The disadvantage of Grid search is not that it cannot be applied to non-differentiable or non-continuous functions. Instead, the disadvantage of Grid search is that it can be computationally expensive and time-consuming, especially when dealing with a large number of hyperparameters and a wide range of parameter values to search through. Final answer: C

\*\*\*\*\*

Answer Option 6:

The disadvantage of Grid Search is not that it can not be applied to non-differentiable or non-continuous functions. It is also not specifically related to the speed of multiple linear regression. The main disadvantage of Grid Search is that it can be computationally expensive and time-consuming, especially when dealing with a large hyperparameter search space. Therefore, the correct answer is D. Final answer: D.

\*\*\*\*\*

Answer Option 7:

The disadvantage of Grid search is not related to non-differentiable or non-continuous functions. Grid search can be applied to any type of function. The disadvantage of Grid search is that it can be computationally expensive and time-consuming, especially when dealing with a large number of hyperparameters and a large search space. Final answer: It runs reasonably slow for multiple linear regression.

\*\*\*\*\*

Answer Option 8:

The disadvantage of Grid search is not that it can't be applied to non-differentiable or non-continuous functions. The main disadvantage of Grid search is that it can be computationally expensive and time-consuming, especially when dealing with a large number of hyperparameters and a large search space. This makes option D, 'It runs reasonably slow for multiple linear regression,' the closest option.

Final answer: D

\*\*\*\*\*

Answer Option 9:

The disadvantage of Grid search is not that it can not be applied to non-differentiable functions or non-continuous functions. It is also not necessarily hard to implement. The main disadvantage of Grid search is that it can be computationally expensive and slow, especially for models with a large number of hyperparameters or when used with large datasets. Final answer: D

\*\*\*\*\*

Question 35:

'Predicting the amount of rainfall in a region based on various cues is a \_\_\_\_\_ problem.', A. 'Supervised learning', B. 'Unsupervised learning', C. 'Clustering', D. 'None of the above'

=====

Answer Option 1:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. In supervised learning, the model is trained using labeled data, where the target variable (in this case, the amount of rainfall) is known. Final answer: A

\*\*\*\*\*

Answer Option 2:

Predicting the amount of rainfall in a region based on various cues is a regression problem, where the goal is to predict a continuous numeric value. Therefore, the correct answer is D. 'None of the above'.

\*\*\*\*\*

Answer Option 3:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. In supervised learning, the model is trained using labeled data, where the inputs are the cues and the output is the amount of rainfall. Final answer: A

\*\*\*\*\*

Answer Option 4:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. In supervised learning, the model is trained using labeled data, where the input variables (cues) are associated with corresponding output labels (amount of rainfall). Final answer: A

\*\*\*\*\*

Answer Option 5:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. In supervised learning, the model is trained using labeled data, where the input (various cues) is associated with the output (amount of rainfall). Final answer: A

\*\*\*\*\*

Answer Option 6:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. In supervised learning, the model is trained using labeled data, where the input (cues) is associated with the desired output (amount of rainfall). Final answer: A

\*\*\*\*\*

Answer Option 7:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. In supervised learning, the model is trained on labeled data, where the input features (cues) are used to predict a specific target variable (amount of rainfall in this case). Final answer: A

\*\*\*\*\*

Answer Option 8:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. In supervised learning, the algorithm is trained on labeled data, where the input features are used to predict a specific target variable. Final answer: A

\*\*\*\*\*

Answer Option 9:

Predicting the amount of rainfall in a region based on various cues is a supervised learning problem. Supervised learning involves training a model on labeled data, where the input features (cues) are used to predict a target variable (amount of rainfall in this case). Clustering is a type of unsupervised learning, which does not involve predicting a specific target variable. Final answer: A.

\*\*\*\*\*

Question 36:

'Which of the following sentence is FALSE regarding regression?', A. 'It relates inputs to outputs.', B. 'It is used for prediction.', C. 'It may be used for interpretation.', D. 'It discovers causal relationships'

=====

Answer Option 1:

The false statement regarding regression is D. 'It discovers causal relationships'. While regression can establish statistical relationships between inputs and outputs, it does not directly imply causation. Final answer: D

\*\*\*\*\*

Answer Option 2:

The false statement regarding regression is D. 'It discovers causal relationships'. Regression models can establish associations between inputs and outputs but do not inherently uncover causal relationships. Final answer: D.

\*\*\*\*\*

Answer Option 3:

The false statement regarding regression is D. 'It discovers causal relationships'. Regression analysis focuses on relating inputs to outputs and is commonly used for prediction. While regression can provide insights and interpretations of relationships between variables, it does not directly discover causal relationships. Final answer: D

\*\*\*\*\*

Answer Option 4:

The sentence that is FALSE regarding regression is D. "It discovers causal relationships." Regression is a statistical method that relates inputs to outputs and is used for prediction. However, it does not inherently discover causal relationships. Final answer: D

\*\*\*\*\*

Answer Option 5:

The false statement regarding regression is D. 'It discovers causal relationships.' Regression is a statistical approach used to model the relationship between a dependent variable and one or more independent variables. While regression can provide insights into associations between variables, it does not directly establish causality. Final answer: D

\*\*\*\*\*

Answer Option 6:

The false statement regarding regression is D. 'It discovers causal relationships'. Regression analysis is used to relate inputs to outputs and make predictions, and it can also be used for interpretation. However, it does not necessarily discover causal relationships, as correlation does not imply causation. Final answer: D

\*\*\*\*\*

Answer Option 7:

The false statement regarding regression is D. 'It discovers causal relationships'. Regression is a statistical technique used to model and analyze the relationship between a dependent variable (output) and one or more independent variables (inputs), but it does not necessarily determine causal relationships. Final answer: D

\*\*\*\*\*

Answer Option 8:

The false statement regarding regression is D. "It discovers causal relationships." Regression analysis is a statistical technique used to model and analyze the relationship between a dependent variable and one or more independent variables. However, it doesn't necessarily imply or discover causal relationships between variables. Final answer: D.

\*\*\*\*\*

Answer Option 9:

The false statement regarding regression is D. 'It discovers causal relationships.'

Regression models are used to relate inputs to outputs and make predictions, so statement A and B are true. Additionally, regression models can be used for interpretation, helping us understand the relationships between variables, so statement C is also true. However, regression alone cannot establish causal relationships between variables. Final Answer: D

\*\*\*\*\*

Question 37:

'Which one of the following is the main reason for pruning a Decision Tree?'; A. 'To save computing time during testing', B. 'To save space for storing the Decision Tree', C. 'To make the training set error smaller', D. 'To avoid overfitting the training set'

=====

Answer Option 1:

The main reason for pruning a decision tree is to avoid overfitting the training set. Pruning helps simplify the decision tree by removing unnecessary branches and nodes that may lead to overfitting. Final

answer: D

\*\*\*\*\*

Answer Option 2:

The main reason for pruning a Decision Tree is to avoid overfitting the training set. Pruning helps prevent the tree from becoming too complex and capturing noise or irrelevant patterns in the training data.

Saving computing time during testing or space for storing the tree are not the main reasons for pruning. Final answer: D

\*\*\*\*\*

Answer Option 3:

The main reason for pruning a Decision Tree is to avoid overfitting the training set. Pruning helps simplify the tree by reducing the number of branches and leaves, which can improve the tree's ability to

generalize and make accurate predictions on unseen data. Saving computing time, space, or reducing training set error are not the main objectives of pruning. Final answer: D

\*\*\*\*\*

Answer Option 4:

The main reason for pruning a decision tree is to avoid overfitting the training set. Pruning helps to prevent the decision tree from becoming too complex and capturing noise or irrelevant patterns in the data,

which can lead to poor generalization performance. Saving computing time, space, or reducing the training set error are not the primary reasons for pruning a decision tree. Final answer: D

\*\*\*\*\*

Answer Option 5:

The main reason for pruning a decision tree is to avoid overfitting the training set. Pruning helps to simplify the decision tree by removing unnecessary branches and reducing its complexity, which can

improve its ability to generalize to new, unseen data. Saving computing time during testing, saving space for storing the decision tree, and reducing the training set error are not the main reasons for pruning a

decision tree. Final answer: D

\*\*\*\*\*

Answer Option 6:

Pruning a decision tree is mainly done to avoid overfitting the training set. Overfitting occurs when a decision tree becomes too complex and captures noise or irrelevant patterns from the training data,

resulting in poor generalization to new, unseen data. Pruning helps simplify the decision tree by removing unnecessary branches, reducing its complexity, and improving its ability to generalize. Saving

computing time during testing or space for storing the decision tree are not the main reasons for pruning. Final answer: D

\*\*\*\*\*

Answer Option 7:

The main reason for pruning a decision tree is to avoid overfitting the training set. Pruning helps to simplify the decision tree by removing unnecessary branches and nodes, thereby preventing overfitting and

improving its generalization performance. Saving computing time during testing and space for storing the decision tree are potential side benefits of pruning, but they are not the main reason for pruning. Final

answer: D

\*\*\*\*\*

Answer Option 8:

The main reason for pruning a Decision Tree is to avoid overfitting the training set. Pruning helps to prevent the tree from becoming too complex and capturing noise or irrelevant patterns in the training data.

Pruning does not primarily aim to save computing time during testing, save space for storing the Decision Tree, or make the training set error smaller. Final answer: D

\*\*\*\*\*

Answer Option 9:

The main reason for pruning a Decision Tree is to avoid overfitting the training set. Pruning helps to simplify the tree by removing unnecessary branches and nodes, improving the model's ability to generalize

to unseen data. Saving computing time, saving space, or reducing training set error are not the primary reasons for pruning. Final answer: D

\*\*\*\*\*

Question 38:

'Statement 1] The kernel density estimator is equivalent to performing kernel regression with the value  $Y_i = 1/n$  at each point  $X_i$  in the original data set. Statement 2] The depth of a learned decision tree can be larger than the number of training examples used to create the tree.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: The kernel density estimator is equivalent to performing kernel regression with the value  $Y_i = 1/n$  at each point  $X_i$  in the original data set.

This statement is False. The kernel density estimator estimates the probability density function of a random variable, while kernel regression estimates the conditional mean of a target variable given the input features.

Statement 2: The depth of a learned decision tree can be larger than the number of training examples used to create the tree.

This statement is True. A decision tree can have a depth larger than the number of training examples. However, this can often result in overfitting.

Final Answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is false. The kernel density estimator estimates the probability density function of a random variable, while kernel regression estimates the conditional expectation of a random variable.

Statement 2 is true. The depth of a learned decision tree can indeed be larger than the number of training examples used to create the tree, as the tree can continue to split until a specified stopping criteria is met.

Final Answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. The kernel density estimator is indeed equivalent to performing kernel regression with the value  $Y_i = 1/n$  at each point  $X_i$  in the original dataset.

Statement 2 is false. The depth of a learned decision tree cannot be larger than the number of training examples used to create the tree. Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is incorrect. The kernel density estimator is used to estimate the probability density function of a random variable. It is not equivalent to performing kernel regression with the value  $Y_i = 1/n$  at each point  $X_i$ .

Statement 2 is correct. The depth of a learned decision tree can be larger than the number of training examples used to create the tree. This can happen when there are multiple features and the tree splits based on different combinations of features.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is False. The kernel density estimator is used to estimate the probability density function of a random variable based on a set of observations, not to perform kernel regression.

Statement 2 is True. The depth of a learned decision tree can be larger than the number of training examples used to create the tree. Decision trees can continue to split until certain stopping criteria are met, such as reaching a minimum number of samples per leaf or a maximum depth. Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is false because the kernel density estimator is used to estimate the probability density function of a random variable, while kernel regression is used to estimate the conditional mean of a random variable.

Statement 2 is true because the depth of a decision tree can be larger than the number of training examples used to create the tree. It is possible for a decision tree to have more levels than the number of training examples, although this can increase the risk of overfitting.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1 is False. The kernel density estimator estimates the probability density function of a random variable based on the data, using a kernel function. It is not equivalent to performing kernel regression.

Statement 2 is True. The depth of a learned decision tree can be larger than the number of training examples used to create the tree. A decision tree can continue to split until it reaches a stopping criterion, which may result in a deeper tree than the number of training examples. Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1 is False. The kernel density estimator estimates the probability density function of a random variable, while kernel regression estimates the conditional expectation of a random variable.

Statement 2 is True. The depth of a decision tree can be larger than the number of training examples used to create the tree. The depth of a decision tree is determined by the complexity of the problem and the splitting criteria used, and it is not directly related to the number of training examples.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The kernel density estimator and kernel regression are two different methods. The kernel density estimator estimates the probability density function of a random variable, while kernel regression estimates the conditional expectation of a random variable given its predictors. Therefore, Statement 1 is False.

Statement 2: The depth of a learned decision tree can indeed be larger than the number of training examples used to create the tree. A decision tree can continue to split until each leaf node contains only a single training example, resulting in a tree depth larger than the number of training examples. Therefore, Statement 2 is True.

Final answer: C

\*\*\*\*\*

Question 39:

'Suppose your model is overfitting. Which of the following is NOT a valid way to try and reduce the overfitting?', A. 'Increase the amount of training data.', B. 'Improve the optimisation algorithm being used for error minimisation.', C. 'Decrease the model complexity.', D. 'Reduce the noise in the training data.'

=====

Answer Option 1:

If a model is overfitting, reducing the noise in the training data is a valid way to try and reduce overfitting. Therefore, the correct answer is D. Final answer: D

\*\*\*\*\*

Answer Option 2:

To reduce overfitting in a model, increasing the amount of training data, improving the optimization algorithm, decreasing the model complexity, and reducing noise in the training data are all valid methods. Therefore, there is no option that is NOT a valid way to try and reduce overfitting. Final answer: There is no correct answer.

\*\*\*\*\*

Answer Option 3:

To reduce overfitting, increasing the amount of training data, improving the optimization algorithm, decreasing the model complexity, and reducing the noise in the training data are all valid approaches. Therefore, there is no option that is NOT a valid way to reduce overfitting. Final answer: None of the options is incorrect.

\*\*\*\*\*

Answer Option 4:

To reduce overfitting, it is generally valid to increase the amount of training data, improve the optimization algorithm, and decrease the model complexity. However, reducing the noise in the training data is not a valid way to reduce overfitting since noise in the data is an inherent part of real-world scenarios. Final answer: D

\*\*\*\*\*

Answer Option 5:

To reduce overfitting, increasing the amount of training data, improving the optimization algorithm, and decreasing the model complexity are valid ways. However, reducing the noise in the training data is not a valid way to reduce overfitting because noise in the data is unrelated to overfitting. Final answer: D

\*\*\*\*\*

Answer Option 6:

To reduce overfitting, you can consider increasing the amount of training data, decreasing the model complexity, and reducing the noise in the training data. However, improving the optimization algorithm being used for error minimization is not a direct method to reduce overfitting. It might help in finding better parameter values, but it does not directly address overfitting. Final answer: B

\*\*\*\*\*

Answer Option 7:

When a model is overfitting, it means it is performing well on the training data but not on new, unseen data. To reduce overfitting, valid ways include: increasing the amount of training data, decreasing the model complexity, and reducing the noise in the training data. However, improving the optimization algorithm being used for error minimization is not a direct way to reduce overfitting. Final answer: B

\*\*\*\*\*

Answer Option 8:

When a model is overfitting, there are several valid ways to try and reduce it by improving generalization. However, option B, "Improve the optimization algorithm being used for error minimization," is not a valid approach to directly reduce overfitting. The optimization algorithm focuses on minimizing the error during training, but overfitting occurs when the model becomes too complex and fits the training data too closely. Final answer: B.

\*\*\*\*\*

Answer Option 9:

When a model is overfitting, it means that it is performing well on the training data but not on the unseen test data. To reduce overfitting, increasing the amount of training data, improving the optimization algorithm, and decreasing the model complexity are all valid ways. However, reducing the noise in the training data may not be a valid way to reduce overfitting. Noise in the data can actually help the model generalize better. Final answer: D

\*\*\*\*\*

Question 40:

'Statement 1] The softmax function is commonly used in multiclass logistic regression. Statement 2] The temperature of a nonuniform softmax distribution affects its entropy.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression to convert the logits into probabilities.

Statement 2 is also true. The temperature parameter in a nonuniform softmax distribution affects its entropy. A higher temperature leads to a more uniform distribution and higher entropy, while a lower temperature leads to a more concentrated distribution and lower entropy.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression to produce a probability distribution over multiple classes.

Statement 2 is also true. The temperature of a nonuniform softmax distribution affects its entropy. A higher temperature increases the entropy, making the distribution more uniform, while a lower temperature decreases the entropy, making the distribution more peaked.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression to convert raw scores into probabilities.

Statement 2 is also true. The temperature of a nonuniform softmax distribution can affect its entropy. Increasing the temperature can make the distribution more uniform and increase the entropy.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression to convert the output logits into a probability distribution over the classes.

Statement 2 is also true. The temperature of a nonuniform softmax distribution affects its entropy. Higher temperatures result in a more uniform distribution and higher entropy, while lower temperatures lead to a more peaked distribution and lower entropy.

Final answer: A

\*\*\*\*\*

Answer Option 5:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression to convert raw predictions into probabilities.

Statement 2 is also true. The temperature of a nonuniform softmax distribution affects its entropy. A higher temperature results in a more uniform distribution and higher entropy, while a lower temperature leads to a more peaked distribution and lower entropy.

Final answer: A

\*\*\*\*\*

Answer Option 6:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression to convert a vector of real numbers into a probability distribution over multiple classes.

Statement 2 is also true. The temperature of a nonuniform softmax distribution affects its entropy. Increasing the temperature leads to a more uniform distribution and increases entropy, while decreasing the temperature makes the distribution more peaked and reduces entropy.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression to convert the raw output of the model into a probability distribution over multiple classes.

Statement 2 is also true. The temperature of a nonuniform softmax distribution affects its entropy. By increasing the temperature, the distribution becomes more uniform and the entropy increases.

Conversely, decreasing the temperature results in a sharper distribution with lower entropy.

Final answer: A

\*\*\*\*\*

Answer Option 8:

Statement 1 is true. The softmax function is commonly used in multiclass logistic regression as it converts the output logits into probabilities that sum up to 1.

Statement 2 is also true. The temperature of a nonuniform softmax distribution affects its entropy. A higher temperature increases the entropy, making the distribution more uniform, while a lower temperature decreases the entropy, making the distribution more peaked.

Final answer: A

\*\*\*\*\*

Answer Option 9:

Statement 1: The softmax function is commonly used in multiclass logistic regression.



- This statement is true. The softmax function is often used to transform the outputs of a multiclass logistic regression model into probabilities.

Statement 2: The temperature of a nonuniform softmax distribution affects its entropy.

- This statement is also true. The temperature parameter in the softmax function allows for control over the level of uncertainty or randomness in the distribution. Higher temperatures lead to higher entropy, resulting in a more uniform distribution. Lower temperatures lead to lower entropy, resulting in a more peaked distribution.

Final answer: C

\*\*\*\*\*

Question 41:

'Which of the following is/are true regarding an SVM?', A. 'For two dimensional data points, the separating hyperplane learnt by a linear SVM will be a straight line.', B. 'In theory, a Gaussian kernel SVM cannot model any complex separating hyperplane.', C. 'For every kernel function used in a SVM, one can obtain an equivalent closed form basis expansion.', D. 'Overfitting in an SVM is not a function of number of support vectors.'

=====

Answer Option 1:

Regarding an SVM, the following statement is true:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will be a straight line.

The other statements are false:

B. In theory, a Gaussian kernel SVM can model complex separating hyperplanes.

C. Not every kernel function used in an SVM can be represented in an equivalent closed form basis expansion.

D. Overfitting in an SVM can be a function of the number of support vectors.

Final answer: A

\*\*\*\*\*

Answer Option 2:

The statement A is true. For two-dimensional data points, the separating hyperplane learned by a linear SVM will be a straight line.

The statement B is false. In theory, a Gaussian kernel SVM can model complex separating hyperplanes as it maps the data points to a higher-dimensional space to find linearly separable boundaries.

The statement C is true. For every kernel function used in an SVM, one can obtain an equivalent closed-form basis expansion.

The statement D is true. Overfitting in an SVM is not solely dependent on the number of support vectors. It can occur if the chosen kernel and its associated parameters are too complex for the given dataset.

Final answer: A, C, D

\*\*\*\*\*

Answer Option 3:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will indeed be a straight line. This is true.

B. In theory, a Gaussian kernel SVM is capable of modeling complex separating hyperplanes. This statement is false. A Gaussian kernel can model complex decision boundaries.

C. Not every kernel function used in an SVM has an equivalent closed-form basis expansion. This statement is false.

D. Overfitting in an SVM can be influenced by the number of support vectors. This statement is false. The number of support vectors does not directly determine whether an SVM model is overfitting or not.

Final answer: A

\*\*\*\*\*

Answer Option 4:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will indeed be a straight line.

B. In theory, a Gaussian kernel SVM can model complex separating hyperplanes, as the Gaussian kernel can map the data points to higher-dimensional space where they can be linearly separable.

C. For every kernel function used in an SVM, one cannot obtain an equivalent closed-form basis expansion. Some kernel functions do not have a corresponding explicit basis expansion.

D. Overfitting in an SVM can be affected by the number of support vectors, as having too many support vectors can lead to a more complex model that may overfit the training data.

Therefore, the correct statement is A. Final answer: A

\*\*\*\*\*

Answer Option 5:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will indeed be a straight line.

B. In theory, a Gaussian kernel SVM can model complex separating hyperplanes. The Gaussian kernel allows for nonlinear decision boundaries.

C. For every kernel function used in an SVM, it is not always possible to obtain an equivalent closed form basis expansion. This depends on the specific kernel function being used.

D. Overfitting in an SVM can be affected by the number of support vectors. Having too many support vectors can lead to overfitting.

Therefore, the correct answer is A.

\*\*\*\*\*

Answer Option 6:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will be a straight line. This statement is true.

B. In theory, a Gaussian kernel SVM can model complex separating hyperplanes. This statement is false. A Gaussian kernel SVM is capable of modeling complex separating hyperplanes.

C. For every kernel function used in an SVM, one cannot always obtain an equivalent closed form basis expansion. This statement is false. Not every kernel function used in an SVM has an equivalent closed form basis expansion.

D. Overfitting in an SVM can be influenced by the number of support vectors. This statement is true. The number of support vectors can impact the model's ability to generalize and can lead to overfitting.

Final answer: A

\*\*\*\*\*

Answer Option 7:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will indeed be a straight line. This is true.

B. In theory, a Gaussian kernel SVM can model complex separating hyperplanes. This statement is false. Gaussian kernel SVMs, also known as Radial Basis Function (RBF) SVMs, can model complex non-linear decision boundaries.

C. For every kernel function used in an SVM, it is not always possible to obtain an equivalent closed form basis expansion. This statement is false. While some kernel functions may have an equivalent closed form basis expansion, not all kernel functions have this property.

D. Overfitting in an SVM can be influenced by the number of support vectors. This statement is true. As the number of support vectors increases, there is a higher chance of overfitting the training data.

Final answer: A

\*\*\*\*\*

Answer Option 8:

Regarding an SVM:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will be a straight line. This statement is true as in two dimensions, the decision boundary of a linear SVM will be a straight line.

B. In theory, a Gaussian kernel SVM cannot model any complex separating hyperplane. This statement is false. Gaussian kernel SVM, also known as an RBF kernel SVM, is able to model complex separating hyperplanes by mapping the data into a higher-dimensional feature space.

C. For every kernel function used in an SVM, one can obtain an equivalent closed form basis expansion. This statement is false. Not every kernel function used in an SVM has an equivalent closed form basis expansion. Some kernel functions, such as the Gaussian kernel, do not have a direct, interpretable basis expansion.

D. Overfitting in an SVM is not a function of the number of support vectors. This statement is true. Overfitting in an SVM is typically not influenced by the number of support vectors, but rather by the complexity of the decision boundary relative to the amount of available data.

Final answer: A

\*\*\*\*\*

Answer Option 9:

A. For two-dimensional data points, the separating hyperplane learned by a linear SVM will be a straight line. This statement is true.

B. In theory, a Gaussian kernel SVM can model complex separating hyperplanes. This statement is false. The Gaussian kernel SVM is capable of modeling complex separating hyperplanes by mapping the data into a higher-dimensional space.

C. For every kernel function used in an SVM, one cannot obtain an equivalent closed-form basis expansion. This statement is false. Not all kernel functions used in SVMs have an equivalent closed-form basis expansion.

D. Overfitting in an SVM can be a function of the number of support vectors. This statement is true. Having too many support vectors in an SVM can lead to overfitting.

Final answer: A

\*\*\*\*\*

Question 42:

'Which of the following is the joint probability of H, U, P, and W described by the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ ? [note: as the product of the conditional probabilities]', A. ' $P(H, U, P, W) = P(H) * P(W) * P(P) * P(U)$ ', B. ' $P(H, U, P, W) = P(H) * P(W) * P(P | W) * P(W | H, P)$ ', C. ' $P(H, U, P, W) = P(H) * P(W) * P(P | W) * P(U | H, P)$ ', D. 'None of the above'

=====

Answer Option 1:

In the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ , the joint probability of H, U, P, and W can be calculated by taking the product of their conditional probabilities. Therefore, the correct answer is B:  $P(H, U, P, W) = P(H) * P(W) * P(P | W) * P(W | H, P)$ . Final answer: B.

=====

Answer Option 2:

Based on the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ , the joint probability of H, U, P, and W can be calculated using the product of the conditional probabilities. Therefore, the correct answer is C. ' $P(H, U, P, W) = P(H) * P(W) * P(P | W) * P(U | H, P)$ '. Final answer: C.

=====

Answer Option 3:

Based on the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ , the joint probability of H, U, P, and W can be calculated as:

$$P(H, U, P, W) = P(H) * P(U | H) * P(P | U) * P(W | P)$$

Therefore, none of the options accurately represent the joint probability. Final answer: D

=====

Answer Option 4:

The joint probability of H, U, P, and W in the given Bayesian network can be calculated as the product of the conditional probabilities. Looking at the network structure, we have:

$$P(H, U, P, W) = P(H) * P(U | H) * P(P | U) * P(W | P)$$

None of the given options correctly represent the joint probability in the Bayesian network. Final answer: D

=====

Answer Option 5:

Based on the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ , the joint probability of H, U, P, and W can be calculated as the product of the conditional probabilities. Therefore, the correct answer is C.  $P(H, U, P, W) = P(H) * P(W) * P(P | W) * P(U | H, P)$ . Final answer: C.

=====

Answer Option 6:

In the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ , the joint probability of H, U, P, and W can be calculated by multiplying the individual conditional probabilities. Since U is directly connected to both H and P, while P is directly connected to W, the correct expression for the joint probability would be:

$$P(H, U, P, W) = P(H) * P(U | H) * P(P | U) * P(W | P)$$

So, the correct answer is D. None of the above.

=====

Answer Option 7:

Based on the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ , the joint probability of H, U, P, and W would be calculated as follows:

$$P(H, U, P, W) = P(H) * P(U | H) * P(P | U) * P(W | P)$$

Therefore, the correct answer is C. ' $P(H, U, P, W) = P(H) * P(W) * P(P | W) * P(U | H, P)$ '.

=====

Answer Option 8:

Based on the given Bayesian Network  $H \rightarrow U \leftarrow P \leftarrow W$ , the joint probability of H, U, P, and W is calculated as the product of the conditional probabilities.

The correct answer is C. ' $P(H, U, P, W) = P(H) * P(W) * P(P | W) * P(U | H, P)$ '. This is because the joint probability includes the probability of H, W, P, and U, conditioned on their respective parents. Final answer: C

=====

Answer Option 9:

In a Bayesian network, the joint probability of the variables is equal to the product of the conditional probabilities. Looking at the given Bayesian network  $H \rightarrow U \leftarrow P \leftarrow W$ , we can see the following dependencies:

1. H does not depend on any other variable.
2. U depends on H.
3. P depends on U.
4. W depends on P.

Therefore, the correct answer would be:

$$P(H, U, P, W) = P(H) * P(U | H) * P(P | U) * P(W | P)$$

So, the correct answer is D. None of the above.

=====



Question 43:

'Statement 1] Since the VC dimension for an SVM with a Radial Base Kernel is infinite, such an SVM must be worse than an SVM with polynomial kernel which has a finite VC dimension. Statement 2] 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.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is False. The VC dimension of an SVM with a Radial Base Kernel is not infinite. In fact, it is finite and depends on the number of support vectors.

Statement 2 is False. A two-layer neural network with linear activation functions is still a linear model and cannot represent non-linear decision boundaries. Boosting, on the other hand, can combine multiple linear models to approximate non-linear relationships.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is false. The VC dimension of an SVM with a Radial Base Kernel is not infinite, but rather it is bounded. It is not necessarily worse than an SVM with a polynomial kernel.

Statement 2 is true. A two-layer neural network with linear activation functions can be represented as a weighted combination of linear separators, and boosting algorithm also finds a combination of linear separators. Therefore, these two algorithms will give the same result.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is false. The VC dimension of a model does not necessarily determine its performance. In fact, a higher VC dimension can indicate a more flexible model that can capture complex patterns in the data.

Statement 2 is also false. A two-layer neural network with linear activation functions cannot represent complex nonlinear decision boundaries. The boosting algorithm, on the other hand, can combine weak classifiers to create a more powerful model that can capture nonlinear relationships in the data.

Final answer: D

\*\*\*\*\*

Answer Option 4:

Statement 1 is false. The VC dimension of an SVM with a radial basis kernel is not infinite. It is actually finite, depending on the number of support vectors.

Statement 2 is false. A two-layer neural network with linear activation functions is not necessarily a weighted combination of linear separators. The boosting algorithm built on linear separators may find a combination of linear separators, but this does not mean that the two algorithms will give the same result.

Final answer: B

\*\*\*\*\*

Answer Option 5:

Statement 1 is False. The VC dimension of an SVM with a Radial Basis Kernel is not necessarily infinite. The VC dimension depends on the complexity of the data and the chosen kernel.

Statement 2 is True. A two-layer neural network with linear activation functions can be seen as a weighted combination of linear separators. Similarly, the boosting algorithm built on linear separators also finds a combination of linear separators. Therefore, these two algorithms will give the same result.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is false. The VC dimension of a model does not directly indicate its performance or superiority. It is a measure of the model's capacity and ability to shatter different patterns. The VC dimension being infinite for an SVM with a Radial Base Kernel does not necessarily mean it is worse than an SVM with a polynomial kernel.

Statement 2 is false. A two-layer neural network with linear activation functions is still a linear model and cannot capture non-linear patterns in data. The boosting algorithm built on linear separators also cannot capture non-linear relationships. Final answer: D

\*\*\*\*\*

Answer Option 7:

Statement 1 is false. The VC dimension of an SVM with a Radial Base Kernel is not infinite. In fact, SVMs with a Radial Base Kernel have a high capacity to capture complex patterns in the data. The VC dimension depends on the complexity of the model, not the type of kernel used.

Statement 2 is false. A two-layer neural network with linear activation functions is still a linear model and cannot capture non-linear relationships in the data. On the other hand, the boosting algorithm can combine multiple linear separators to create a non-linear decision boundary, allowing it to capture more complex patterns in the data.

Final answer: D

\*\*\*\*\*

Answer Option 8:

Statement 1: The VC dimension of a model does not directly determine its performance or indicate that a model is worse or better than another model. Therefore, the statement is false.

Statement 2: A two-layer neural network with linear activation functions is still a linear model and can only represent linear separators. The boosting algorithm built on linear separators, on the other hand, can combine multiple linear separators to create a more complex decision boundary. Therefore, these two algorithms will not give the same result. The statement is false.

Final answer: C (False, False)

\*\*\*\*\*

Answer Option 9:

Statement 1: The statement that an SVM with a Radial Base Kernel must be worse than an SVM with a polynomial kernel due to the infinite VC dimension is false. The VC dimension is a measure of the capacity of a classifier, but it does not directly determine the performance or quality of the classifier. Therefore, Statement 1 is false.

Statement 2: The statement that a two-layer neural network with linear activation functions and a boosting algorithm built on linear separators will give the same result is false. The boosting algorithm, which combines weak learners to create a strong learner, does not necessarily result in a combination of linear separators. Therefore, Statement 2 is false.

Final answer: D

\*\*\*\*\*

Question 44:

'Statement 1] The ID3 algorithm is guaranteed to find the optimal decision tree. Statement 2] Consider a continuous probability distribution with density  $f()$  that is nonzero everywhere. The probability of a value  $x$  is equal to  $f(x)$ .'. A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is False. The ID3 algorithm is a greedy algorithm and does not guarantee finding the optimal decision tree.

Statement 2 is False. The probability of a continuous random variable is given by the area under the probability density function, not equal to the density function itself.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree. It uses a greedy approach and can get stuck in local optima.

Statement 2: The statement is not true. The probability of a specific value  $x$  in a continuous probability distribution is usually zero, as the probability is defined over intervals rather than specific points.

Final answer: C (True, False)

\*\*\*\*\*

Answer Option 3:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree. It uses a greedy approach and may get stuck in local optima, resulting in a suboptimal tree. Therefore, statement 1 is False.

Statement 2: This statement describes the definition of a continuous probability distribution. The probability of a value  $x$  is indeed equal to the density function  $f(x)$  for a continuous distribution. Therefore, statement 2 is True.

Final answer: C (False, True)

\*\*\*\*\*

Answer Option 4:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree. It uses a top-down greedy approach and may get stuck in local optima, resulting in a suboptimal decision tree. Therefore,

Statement 1 is false.

Statement 2: The probability of a specific value in a continuous probability distribution is always zero since the integral (area under the density curve) over a single point is zero. Therefore, Statement 2 is also false.

Final answer: B (False, False)

\*\*\*\*\*

Answer Option 5:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree as it uses a greedy approach and may get stuck in local optima. Therefore, Statement 1 is false.

Statement 2: This statement is not clear and seems to be an incorrect representation of a probability distribution. The probability of a value  $x$  in a continuous probability distribution is not equal to the density function  $f(x)$  itself. Therefore, Statement 2 is false.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree. It uses a greedy approach and may get stuck in local optima, resulting in a suboptimal tree. So, statement 1 is False.

Statement 2: The probability of a value  $x$  in a continuous probability distribution is not equal to the density function  $f(x)$ . Instead, the probability of an event happening within a specific range is given by the integral of the density function over that range. So, statement 2 is False.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree. It uses a heuristic approach to build decision trees and may not always produce the optimal solution. Therefore, Statement 1 is false.

Statement 2: The statement is not clear and seems incomplete. Without additional information, it is not possible to determine the correctness of the statement. Therefore, Statement 2 is not determinable.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree. It uses a greedy approach that may result in suboptimal trees. Therefore, it is False.

Statement 2: The probability of a value in a continuous probability distribution is defined as the area under the density function. It is not equal to the density function itself. Therefore, it is False.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The ID3 algorithm is not guaranteed to find the optimal decision tree. It is a greedy algorithm that makes locally optimal decisions at each step, which may not lead to the globally optimal solution. Therefore, Statement 1 is false.



Statement 2: The second statement is not clear. If it is asserting that the probability of a specific value  $x$  is equal to the density function  $f(x)$ , then it is false. The probability of a continuous random variable taking on a specific value is zero in most cases. If it is asserting that the probability density function  $f(x)$  describes the distribution of the random variable, then it is true.

Final answer: C. 'True, False'

\*\*\*\*\*

Question 45:

'Given a Neural Net with N input nodes, no hidden layers, one output node, with Entropy Loss and Sigmoid Activation Functions, which of the following algorithms (with the proper hyper-parameters and initialization) can be used to find the global optimum?', A. 'Stochastic Gradient Descent', B. 'Mini-Batch Gradient Descent', C. 'Batch Gradient Descent', D. 'All of the above'

=====

Answer Option 1:

Given the specific neural network architecture with entropy loss and sigmoid activation functions, the global optimum can only be guaranteed by using batch gradient descent. Stochastic gradient descent and mini-batch gradient descent may converge to a local optimum. Final answer: C

\*\*\*\*\*

Answer Option 2:

In this scenario, with a neural network having no hidden layers, one output node, Entropy Loss, and Sigmoid Activation Functions, the problem is convex. Therefore, any of the listed algorithms can be used to find the global optimum. Final answer: D

\*\*\*\*\*

Answer Option 3:

In the given scenario, where the neural network has no hidden layers, one output node, and uses the entropy loss function with the sigmoid activation function, all of the mentioned algorithms (stochastic gradient descent, mini-batch gradient descent, and batch gradient descent) can be used to find the global optimum. Final answer: D

\*\*\*\*\*

Answer Option 4:

In the given scenario, with a neural net having N input nodes, no hidden layers, one output node, using Entropy Loss and Sigmoid Activation Functions, all of the mentioned algorithms (Stochastic Gradient Descent, Mini-Batch Gradient Descent, and Batch Gradient Descent) can be used to find the global optimum. Final answer: D

\*\*\*\*\*

Answer Option 5:

In the given scenario, where we have a neural network with no hidden layers, one output node, Entropy Loss, and Sigmoid Activation Functions, all of the above algorithms (Stochastic Gradient Descent, Mini-Batch Gradient Descent, and Batch Gradient Descent) can be used to find the global optimum. Therefore, the correct answer is D. Final answer: D

\*\*\*\*\*

Answer Option 6:

In the given scenario, where there are no hidden layers and one output node, the Neural Net can be considered as a logistic regression model. In this case, all three algorithms (Stochastic Gradient Descent, Mini-Batch Gradient Descent, and Batch Gradient Descent) can be used to find the global optimum, as they are all applicable to logistic regression. Therefore, the correct answer is D. 'All of the above'.

\*\*\*\*\*

Answer Option 7:

In this scenario, since we have a neural net with no hidden layers and one output node, and we are using the Sigmoid activation function and the Entropy loss function, all of the mentioned algorithms (Stochastic Gradient Descent, Mini-Batch Gradient Descent, and Batch Gradient Descent) can be used to find the global optimum. Final answer: D

\*\*\*\*\*

Answer Option 8:

In this scenario, since we have a Neural Net with no hidden layers and only one output node, the problem reduces to binary classification. For binary classification problems with sigmoid activation and entropy loss function, all of the mentioned algorithms (Stochastic Gradient Descent, Mini-Batch Gradient Descent, and Batch Gradient Descent) can be used to find the global optimum. Final answer: D

\*\*\*\*\*

Answer Option 9:

With the given network configuration and loss function, all three algorithms - Stochastic Gradient Descent, Mini-Batch Gradient Descent, and Batch Gradient Descent - can be used to find the global optimum. Stochastic Gradient Descent updates the weights after each individual training example, Mini-Batch Gradient Descent updates the weights after a small batch of training examples, and Batch Gradient Descent updates the weights after all the training examples in one go. Final answer: D

\*\*\*\*\*

Question 46:

'Adding more basis functions in a linear model, pick the most probably option:', A. 'Decreases model bias', B. 'Decreases estimation bias', C. 'Decreases variance', D. 'Doesn't affect bias and variance'

=====

Answer Option 1:

Adding more basis functions in a linear model increases its complexity and flexibility. This can help reduce bias but can also increase variance. Therefore, the most probable option is: C. 'Decreases variance'.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Adding more basis functions in a linear model increases its flexibility and capacity to fit the data, which can lead to a decrease in model bias. Therefore, the most probable option is: A. 'Decreases model bias'.

\*\*\*\*\*

Answer Option 3:

Adding more basis functions in a linear model increases model complexity, which can lead to a decrease in bias but an increase in variance. Therefore, the most probable option is C. 'Decreases variance'.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Adding more basis functions in a linear model generally increases model complexity and flexibility. This can lead to a decrease in model bias but an increase in model variance. Therefore, the most probable option would be: C. 'Decreases variance'. Final answer: C

\*\*\*\*\*

Answer Option 5:

Adding more basis functions in a linear model increases its complexity and flexibility. This can help the model capture more intricate patterns in the data, potentially reducing both bias and variance. However, as the complexity increases, there is a risk of overfitting, leading to higher variance. Final answer: D

\*\*\*\*\*

Answer Option 6:

Adding more basis functions in a linear model increases model flexibility and complexity. This increase in complexity can lead to overfitting and higher variance. Therefore, the most probable option is C. 'Decreases variance'. Adding more basis functions does not directly affect model bias or estimation bias. Final answer: C

\*\*\*\*\*

Answer Option 7:

Adding more basis functions in a linear model generally increases the model complexity and flexibility. This tends to decrease the model bias, as it allows the model to fit the data more closely. However, it may also increase the variance of the model, as it becomes more sensitive to noise and fluctuations in the data. Therefore, the most probable option would be: A. 'Decreases model bias'. Final answer: A

\*\*\*\*\*

Answer Option 8:

Adding more basis functions in a linear model typically increases the complexity and flexibility of the model. This can lead to a decrease in bias, as the model becomes more capable of capturing complex relationships in the data. However, it can also increase variance, as the model becomes more sensitive to noise and outliers. Therefore, the most probable option is C. 'Decreases variance'. Final answer: C.

\*\*\*\*\*

Answer Option 9:

Adding more basis functions in a linear model typically increases the model complexity and flexibility. This can lead to a decrease in bias, as the model can better fit the training data. However, it can also increase variance, as the model becomes more sensitive to noise in the training data. Therefore, the most probable option is D. 'Doesn't affect bias and variance'. Final answer: D.

\*\*\*\*\*

Question 47:

'Consider the Bayesian network given below. How many independent parameters would we need if we made no assumptions about independence or conditional independence  $H \rightarrow U \leftarrow P \leftarrow W$ ?' A. '3', B. '4', C. '7', D. '15'

=====

Answer Option 1:

In a Bayesian network, each node's probability distribution is dependent on its parents. In this given network ( $H \rightarrow U \leftarrow P \leftarrow W$ ), if we make no assumptions about independence or conditional independence, we would need to specify a conditional probability distribution for each node given its parent(s).

H (1 parameter), U (2 parameters), P (2 parameters), and W (2 parameters) would require a total of  $1 + 2 + 2 + 2 = 7$  parameters.

Therefore, the correct answer is C. '7'.

\*\*\*\*\*

Answer Option 2:

In a Bayesian network, the number of independent parameters that would be needed is equal to the product of the number of states of each variable, minus the number of parameters that can be determined from the network structure.

In this case, we have:

- H, which has 2 states (true/false)
- U, which has 2 states
- P, which has 2 states
- W, which has 2 states

Therefore, without any assumptions about independence or conditional independence, we would need  $(2 * 2 * 2 * 2) - 3 = 15 - 3 = 12$  independent parameters.

Final answer: D

\*\*\*\*\*

Answer Option 3:

In a Bayesian network, the number of independent parameters needed is determined by the number of conditional probabilities that need to be specified. In this case, we have 4 nodes: H, U, P, and W.

To determine the number of independent parameters, we need to consider the following:

- H has no parents, so it needs 2 parameters:  $P(H)$  and  $P(\neg H)$
- U has one parent (H), so it needs 2 parameters:  $P(U|H)$  and  $P(U|\neg H)$
- P has two parents (U and W), so it needs 4 parameters:  $P(P|U,W)$
- W has no children, so it needs 2 parameters:  $P(W)$  and  $P(\neg W)$

Therefore, the total number of independent parameters is  $2 + 2 + 4 + 2 = 10$ . Final answer: D

\*\*\*\*\*

Answer Option 4:

In a Bayesian network, the number of independent parameters required is equal to the number of conditional probability tables (CPTs) needed to represent the relationships between the variables.

In the given network, we have 4 variables: H, U, P, and W.

- H has no parents, so it requires a CPT with 2 parameters ( $P(H)$ ).
- U has a single parent H, so it requires a CPT with 2 parameters ( $P(U|H)$ ).
- P has two parents U and W, so it requires a CPT with 4 parameters ( $P(P|U, W)$ ).
- W has no children, so it does not require any parameters.

Adding up the parameters needed for each variable, we have:  $2 + 2 + 4 + 0 = 8$  parameters.

Therefore, the answer is not among the given options. None of the options A, B, C, or D are correct.

\*\*\*\*\*

Answer Option 5:

In the given Bayesian network, the variables H, U, P, and W are connected in a directed acyclic graph. If we make no assumptions about independence or conditional independence, we would need to estimate all possible conditional probability distributions.

For each variable, we need to estimate its conditional probability distribution given its parents. In this case, H has no parent, U has H as its parent, P has U as its parent, and W has P as its parent.

Since H has no parent, we need to estimate  $P(H)$ , which requires 2 parameters (assuming H is a binary variable).

Since U has H as its parent, we need to estimate  $P(U|H)$ , which again requires 2 parameters.

Similarly, for P we need to estimate  $P(P|U)$ , requiring 2 parameters.

Finally, for W we need to estimate  $P(W|P)$ , requiring 2 parameters.

Therefore, the total number of independent parameters needed would be  $2 + 2 + 2 + 2 = 8$ .

Final answer: 8

\*\*\*\*\*

Answer Option 6:

In a Bayesian network, the number of independent parameters required can be calculated by considering the number of parameters needed for each node, given its parents.

In this case, we have 4 nodes: H, U, P, and W.

- H has no parents, so it requires 2 parameters (one for each possible state, true or false).

- U has one parent, so it requires 2 parameters for each possible state of its parent, which is 2. Therefore, it requires 4 parameters.

- P has one parent, so it also requires 4 parameters.

- W has one parent, so it requires 4 parameters.

Adding up the parameters for each node, we get:  $2 + 4 + 4 + 4 = 14$ .

However, we also need to consider the conditional independence assumption. Given that U and P are both parents of W, we can assume that they are conditionally independent of each other given W.

Therefore, we only need to account for the parameters for W and its parents, which is 4.

Final answer: A. 3

\*\*\*\*\*

Answer Option 7:

In the given Bayesian network, there are 4 variables: H, U, P, and W.

If we make no assumptions about independence or conditional independence, we would need to estimate the parameters for each variable separately.

For variable H, we would need to estimate its conditional probability distribution, which requires 2 parameters ( $P(H=\text{true})$  and  $P(H=\text{false})$ ).

For variable U, we would need to estimate its conditional probability distribution given H, which requires 4 parameters ( $P(U=\text{true}|H=\text{true})$ ,  $P(U=\text{true}|H=\text{false})$ ,  $P(U=\text{false}|H=\text{true})$ , and  $P(U=\text{false}|H=\text{false})$ ).

For variable P, we would need to estimate its conditional probability distribution given U, which requires 4 parameters ( $P(P=\text{true}|U=\text{true})$ ,  $P(P=\text{true}|U=\text{false})$ ,  $P(P=\text{false}|U=\text{true})$ , and  $P(P=\text{false}|U=\text{false})$ ).

For variable W, we would need to estimate its conditional probability distribution given P, which requires 2 parameters ( $P(W=\text{true}|P=\text{true})$  and  $P(W=\text{false}|P=\text{true})$ ).

Therefore, the total number of independent parameters needed is  $2 + 4 + 4 + 2 = 12$ .

Final answer: D

\*\*\*\*\*

Answer Option 8:

In a Bayesian network, the number of independent parameters required can be determined by considering the number of conditional probability tables (CPTs) needed.

In this given Bayesian network, we have the following nodes: H, U, P, and W.

- Node H has no parents, so it requires 2 parameters ( $P(H=\text{True})$  and  $P(H=\text{False})$ ).

- Node U has one parent (H), so it requires 4 parameters ( $P(U=\text{True}|H=\text{True})$ ,  $P(U=\text{False}|H=\text{True})$ ,  $P(U=\text{True}|H=\text{False})$ , and  $P(U=\text{False}|H=\text{False})$ ).

- Node P has one parent (U), so it requires 4 parameters ( $P(P=\text{True}|U=\text{True})$ ,  $P(P=\text{False}|U=\text{True})$ ,  $P(P=\text{True}|U=\text{False})$ , and  $P(P=\text{False}|U=\text{False})$ ).

- Node W has one parent (P), so it requires 4 parameters ( $P(W=\text{True}|P=\text{True})$ ,  $P(W=\text{False}|P=\text{True})$ ,  $P(W=\text{True}|P=\text{False})$ , and  $P(W=\text{False}|P=\text{False})$ ).

Therefore, the total number of independent parameters required is  $2 + 4 + 4 + 4 = 14$ .

Final answer: D

\*\*\*\*\*

Answer Option 9:

In the given Bayesian network, there are four variables: H, U, P, and W. Each variable has its own set of parameters, which represents the probabilities associated with that variable given its parents.

H has no parent, so it has one set of parameters, which requires 2 independent values since it is a binary variable (true or false).

U has one parent (H), so it requires 2 parameters for each value of H (true and false), resulting in a total of 4 parameters.

P has one parent (U), so it requires 2 parameters for each value of U (true and false), resulting in a total of 8 parameters.

W has one parent (P), so it requires 2 parameters for each value of P (true and false), resulting in a total of 8 parameters.

Therefore, the total number of independent parameters needed is  $2 + 4 + 8 + 8 = 22$ .

Since none of the answer choices match, it seems there is an error in the provided options. None of the options are correct.

\*\*\*\*\*

Question 48:

'Another term for out-of-distribution detection is?'. A. 'anomaly detection', B. 'one-class detection', C. 'train-test mismatch robustness', D. 'background detection'

=====

Answer Option 1:

Another term for out-of-distribution detection is anomaly detection. Final answer: A

\*\*\*\*\*

Answer Option 2:

Another term for out-of-distribution detection is anomaly detection. Final answer: A

\*\*\*\*\*

Answer Option 3:

Another term for out-of-distribution detection is anomaly detection. Final answer: A

\*\*\*\*\*

Answer Option 4:

Another term for out-of-distribution detection is anomaly detection. Anomaly detection refers to the task of identifying instances that deviate from the normal or expected behavior. Final answer: A

\*\*\*\*\*

Answer Option 5:

Another term for out-of-distribution detection is anomaly detection. Anomaly detection refers to the task of identifying data points that deviate from the norm or exhibit unusual behavior. Final answer: A

\*\*\*\*\*

Answer Option 6:

Another term for out-of-distribution detection is anomaly detection. Anomaly detection is the task of identifying patterns or instances that deviate significantly from the norm or expected behavior. Final answer:

A

\*\*\*\*\*

Answer Option 7:

Another term for out-of-distribution detection is anomaly detection. It involves identifying samples that deviate significantly from the expected distribution or patterns in the data. One-class detection refers to a similar concept but focuses on identifying outliers within a single class. Train-test mismatch robustness refers to the ability of a model to perform well when the training and testing data come from different distributions. Background detection is not a commonly used term in this context. Final answer: A

\*\*\*\*\*

Answer Option 8:

Another term for out-of-distribution detection is anomaly detection. Anomaly detection refers to the task of identifying observations that deviate significantly from the norm or expected behavior. One-class detection is a specific type of anomaly detection where the model is trained on only one class of data. Train-test mismatch robustness refers to the ability of a model to perform well on test data that is different from the training data. Background detection does not refer specifically to out-of-distribution detection. Final answer: A

\*\*\*\*\*

Answer Option 9:

Another term for out-of-distribution detection is anomaly detection. This refers to the task of identifying samples that do not belong to the known distributions of the training data. One-class detection is a related concept, but it focuses on detecting anomalies within a single known class. Train-test mismatch robustness refers to the ability of a model to perform well on test data that may differ from the training data. Background detection is not directly related to out-of-distribution detection. Final answer: A

\*\*\*\*\*

Question 49:

'Statement 1] We learn a classifier  $f$  by boosting weak learners  $h$ . The functional form of  $f$ 's decision boundary is the same as  $h$ 's, but with different parameters. (e.g., if  $h$  was a linear classifier, then  $f$  is also a linear classifier). Statement 2] Cross validation can be used to select the number of iterations in boosting; this procedure may help reduce overfitting.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. In boosting, a classifier  $f$  is learned by combining multiple weak learners  $h$ . The functional form of  $f$ 's decision boundary is the same as  $h$ 's, but with different parameters.

Statement 2 is false. Cross validation is not typically used to select the number of iterations in boosting. Instead, other techniques such as early stopping or using a validation set are commonly used. Final

answer: C

=====

Answer Option 2:

Statement 1: Boosting is a technique where weak learners are combined to create a strong learner. The functional form of the final classifier  $f$  is not necessarily the same as the weak learner  $h$ . So, statement 1 is False.

Statement 2: Cross-validation can be used to select the number of iterations in boosting, as it helps to estimate the performance of the model on unseen data. This procedure can indeed help reduce overfitting. So, statement 2 is True.

Final answer: C

=====

Answer Option 3:

Statement 1 is true. Boosting is a technique where a strong classifier is built by combining multiple weak classifiers. The functional form of the final classifier's decision boundary is the same as that of the weak classifier, but with different parameters.

Statement 2 is also true. Cross-validation can be used to select the number of iterations (or weak classifiers) in boosting. This can help prevent overfitting by finding an optimal number of iterations that balances model complexity and generalization performance.

Final answer: A

=====

Answer Option 4:

The correct answer is C. 'True, False'.

Statement 1 is true. Boosting is a technique where weak learners are combined to form a strong learner. The functional form of the final classifier  $f$  is the same as the weak learner  $h$ , but with different parameters.

Statement 2 is false. Cross-validation is typically used to estimate the performance of a model or to tune its hyperparameters. It is not directly used to select the number of iterations in boosting. The number of iterations in boosting is usually determined using early stopping or other stopping criteria to prevent overfitting.

=====

Answer Option 5:

Statement 1 is true. Boosting is a technique in which a strong classifier is built by combining multiple weak classifiers. The functional form of the final classifier's decision boundary is the same as the weak classifier's decision boundary, but with different parameters.

Statement 2 is false. Cross-validation is a technique used to estimate the performance of a model, but it is not typically used to select the number of iterations in boosting. The number of iterations in boosting is typically determined through other means, such as monitoring the performance on a validation set or using techniques like early stopping.

Final answer: C

=====

Answer Option 6:

Statement 1 is false. Boosting combines multiple weak learners (e.g., decision stumps) to form a strong learner (e.g., an ensemble of decision trees). The functional form of the boosted classifier may be different from the weak learner, as it combines multiple weak learners.

Statement 2 is true. Cross-validation can be used to select the number of iterations in boosting to help reduce overfitting. It helps to find the optimal number of iterations that achieves the best trade-off between bias and variance.

Final answer: C

=====

Answer Option 7:

Statement 1 is True. Boosting is a machine learning ensemble technique that combines multiple weak learners (e.g., decision trees) to create a strong learner. The decision boundary of the boosted classifier is the same as the weak learner's decision boundary, but with different parameters.

Statement 2 is False. Cross validation is typically used to estimate the performance of a model, not to select the number of iterations in boosting. The number of iterations in boosting is usually determined using other techniques or heuristics, such as early stopping or using a validation set to monitor performance. It is not directly related to cross validation.

Final answer: C

=====

Answer Option 8:

Statement 1: We learn a classifier  $f$  by boosting weak learners  $h$ . The functional form of  $f$ 's decision boundary is the same as  $h$ 's, but with different parameters. (e.g., if  $h$  was a linear classifier, then  $f$  is also a linear classifier).

This statement is True. Boosting combines multiple weak learners to create a strong classifier, and the functional form of the decision boundary of the final boosted classifier is the same as the weak learner's, but with different parameters.

Statement 2: Cross validation can be used to select the number of iterations in boosting; this procedure may help reduce overfitting.

This statement is False. Cross validation is typically used to estimate the performance of a model and select hyperparameters, but it is not commonly used to directly select the number of iterations in boosting. The number of iterations in boosting is usually determined through other techniques, such as early stopping or using a separate validation set.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: We learn a classifier  $f$  by boosting weak learners  $h$ . The functional form of  $f$ 's decision boundary is the same as  $h$ 's, but with different parameters. (e.g., if  $h$  was a linear classifier, then  $f$  is also a linear classifier).

This statement is False. Boosting does not necessarily preserve the functional form of the weak learner's decision boundary. It combines multiple weak learners to create a stronger, more complex model.

Statement 2: Cross validation can be used to select the number of iterations in boosting; this procedure may help reduce overfitting.

This statement is True. Cross-validation can be used to evaluate the performance of a boosted model with different numbers of iterations. It can help in selecting the optimal number of iterations that balances model complexity and overfitting.

Final answer: C

\*\*\*\*\*



Question 50:

'Statement 1| Highway networks were introduced after ResNets and eschew max pooling in favor of convolutions. Statement 2| DenseNets usually cost more memory than ResNets.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. Highway networks were introduced before ResNets and they do use max pooling in addition to convolutions.

Statement 2 is true. DenseNets typically require more memory compared to ResNets because they concatenate feature maps from all previous layers. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is false. Highway networks were introduced before ResNets and they do use max pooling along with convolutions.

Statement 2 is true. DenseNets typically require more memory compared to ResNets due to their dense connectivity pattern.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is false. Highway networks were introduced before ResNets and they do use max pooling in addition to convolutions.

Statement 2 is true. DenseNets generally require more memory compared to ResNets due to their dense connectivity pattern.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is false. Highway networks were introduced before ResNets and do not necessarily eschew max pooling in favor of convolutions.

Statement 2 is true. DenseNets do usually require more memory compared to ResNets due to their dense connectivity pattern.

Final answer: C

\*\*\*\*\*

Answer Option 5:

The correct answer is C.

Statement 1 is true. Highway networks were indeed introduced after ResNets and they replace max pooling with convolutions.

Statement 2 is false. DenseNets generally require less memory compared to ResNets because they allow for feature reuse through dense connections.

\*\*\*\*\*

Answer Option 6:

Statement 1 is incorrect. Highway networks were introduced before ResNets and use a combination of convolutions and gating mechanisms. So the correct answer would be False for Statement 1.

Statement 2 is correct. DenseNets generally consume more memory than ResNets due to their densely connected structure. So the correct answer would be True for Statement 2.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: Highway networks were actually introduced before ResNets and do use max pooling in addition to convolutions. This statement is false.

Statement 2: DenseNets do tend to require more memory than ResNets due to their dense connectivity pattern. This statement is true.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: Highway networks were introduced after ResNets and eschew max pooling in favor of convolutions.

Statement 2: DenseNets usually cost more memory than ResNets.

Both statements are true. Highway networks were indeed introduced after ResNets and they replace max pooling with convolutions. DenseNets also typically require more memory compared to ResNets due to their densely connected layers. Final answer: A

\*\*\*\*\*

Answer Option 9:

Statement 1: Highway networks were introduced after ResNets and eschew max pooling in favor of convolutions. This statement is False. Highway networks were introduced before ResNets and they combine both convolutional layers and gating mechanisms.

Statement 2: DenseNets usually cost more memory than ResNets. This statement is True. DenseNets typically require more memory due to their dense connectivity pattern, where each layer is connected to every other layer in a feed-forward fashion.

Final answer: C

\*\*\*\*\*

Question 51:

'If N is the number of instances in the training dataset, nearest neighbors has a classification run time of', A. ' $O(1)$ ', B. ' $O(N)$ ', C. ' $O(\log N)$ ', D. ' $O(N^2)$ '

=====

Answer Option 1:

The nearest neighbors algorithm has a classification runtime of  $O(N)$  because it needs to compare the test instance with all N instances in the training dataset to find the nearest neighbors. Final answer: B.

\*\*\*\*\*

Answer Option 2:

The classification run time of nearest neighbors is dependent on the number of instances in the training dataset, denoted by N. Nearest neighbors has a classification run time complexity of  $O(N)$ . Final

answer: B

\*\*\*\*\*

Answer Option 3:

The nearest neighbors algorithm has a classification run time complexity of  $O(N)$ . This means that the time it takes to classify a new instance is directly proportional to the number of instances in the training dataset. Final answer: B

\*\*\*\*\*

Answer Option 4:

The nearest neighbors algorithm has a classification run time of  $O(N)$  because it needs to compare each instance in the training dataset to the new instance being classified. Final answer: B

\*\*\*\*\*

Answer Option 5:

Nearest neighbors classification has a run time complexity of  $O(N)$ , where N is the number of instances in the training dataset. This is because, during prediction, the algorithm needs to compare the new instance with each training instance to find the nearest neighbors. Final answer: B

\*\*\*\*\*

Answer Option 6:

The nearest neighbors algorithm has a classification run time of  $O(N)$ , where N is the number of instances in the training dataset. This is because it needs to compare each instance in the training dataset to the given instance in order to find the nearest neighbors. Final answer: B

\*\*\*\*\*

Answer Option 7:

The nearest neighbors algorithm has a classification run time of  $O(N)$ , where N is the number of instances in the training dataset. This is because for each new instance to be classified, the algorithm needs to calculate the distance to all N instances in the training dataset. Final answer: B

\*\*\*\*\*

Answer Option 8:

The runtime of nearest neighbors classification is dependent on the number of instances in the training dataset (N). As the algorithm needs to compare the new instance to each of the existing instances, the runtime is linear to the number of instances. Therefore, the correct answer is B.  $O(N)$ .

\*\*\*\*\*

Answer Option 9:

The nearest neighbors algorithm has a classification run time of  $O(N)$ , where N is the number of instances in the training dataset. This is because, in order to classify a new instance, the algorithm needs to calculate the distances between the new instance and all the instances in the training dataset. Final answer: B

\*\*\*\*\*

Question 52:

'Statement 1] The original ResNets and Transformers are feedforward neural networks. Statement 2] The original Transformers use self-attention, but the original ResNet does not.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: The original ResNets and Transformers are not feedforward neural networks. ResNets are a type of deep residual neural network that uses skip connections, while Transformers are a type of neural network that uses self-attention mechanisms.

Statement 2: The original Transformers do use self-attention, but the original ResNet does not.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: The original ResNets and Transformers are not feedforward neural networks. ResNets are residual neural networks that use skip connections to address the vanishing gradient problem.

Transformers are a type of neural network architecture that uses self-attention mechanisms.

Statement 2: The original Transformers do use self-attention, while the original ResNet does not.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1: The original ResNets and Transformers are feedforward neural networks.

This statement is False as the original Transformers are not feedforward neural networks. They use self-attention mechanism.

Statement 2: The original Transformers use self-attention, but the original ResNet does not.

This statement is True as the original Transformers do use self-attention mechanism, while the original ResNet does not.

Final Answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is False because ResNets and Transformers are not feedforward neural networks. ResNets have skip connections that allow information to flow in a non-linear manner, and Transformers have attention mechanisms that enable information to be exchanged between different parts of the sequence.

Statement 2 is True because the original Transformers do use self-attention, which allows the model to focus on different parts of the input sequence while processing it. On the other hand, the original ResNet does not use self-attention.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1: The original ResNets and Transformers are not feedforward neural networks. ResNets contain skip connections that allow the network to skip over certain layers, forming shortcuts, while Transformers are based on an encoder-decoder architecture with attention mechanisms.

Statement 2: The original Transformers do use self-attention, which allows the model to weigh the importance of different words in the input sequence. On the other hand, the original ResNet does not use self-attention.

Statement 2: The original Transformers do use self-attention, which allows the model to weigh the importance of different words in the input sequence. On the other hand, the original ResNet does not use self-attention.

Final answer: C (True, False)

\*\*\*\*\*

Answer Option 6:

Statement 1 is False. The original ResNets and Transformers are not feedforward neural networks. ResNets are known for their skip connections, allowing for the flow of gradients during training, while Transformers have both encoder and decoder layers that utilize attention mechanisms.

Statement 2 is True. The original Transformers use self-attention, also known as scaled dot-product attention, to compute the context representation, while the original ResNet does not use attention mechanisms.

Statement 2 is True. The original Transformers use self-attention, also known as scaled dot-product attention, to compute the context representation, while the original ResNet does not use attention mechanisms.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1 is False because both ResNets and Transformers are not purely feedforward neural networks. ResNets use skip connections that allow the information to bypass one or more layers, while Transformers use self-attention mechanisms that allow for non-linear connections between different parts of the input sequence.

Statement 2 is True because the original Transformers do use self-attention, while the original ResNet does not use self-attention.

Statement 2 is True because the original Transformers do use self-attention, while the original ResNet does not use self-attention.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 8:

Statement 1: The original ResNets and Transformers are feedforward neural networks.

This statement is False. While the original ResNet is a type of feedforward neural network, the original Transformers are not strictly feedforward networks as they incorporate self-attention mechanisms.

Statement 2: The original Transformers use self-attention, but the original ResNet does not.

This statement is True. Self-attention is a key component of the original Transformer architecture, allowing it to capture dependencies between different elements of the input sequence. The original ResNet, on the other hand, does not use self-attention.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1 is false because the original ResNets and Transformers are not feedforward neural networks. ResNets are a type of deep residual neural network that uses skip connections, allowing for the training of very deep networks. Transformers, on the other hand, are a type of architecture used for sequence-to-sequence tasks such as machine translation.

Statement 2 is false because the original Transformers do use self-attention, which allows them to capture dependencies between different positions in the input sequence. Self-attention is a key component of the Transformer architecture.

Therefore, the correct answer is D. 'False, True'.

\*\*\*\*\*

Question 53:

'Statement 1] RELUs are not monotonic, but sigmoids are monotonic. Statement 2] Neural networks trained with gradient descent with high probability converge to the global optimum.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false because ReLUs (Rectified Linear Units) are actually monotonic. Statement 2 is false because neural networks trained with gradient descent do not always converge to the global optimum; they can sometimes get stuck in local optima. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: RELUs (Rectified Linear Units) are indeed monotonic, while sigmoids are not monotonic. Therefore, statement 1 is False.

Statement 2: Neural networks trained with gradient descent do not have a guarantee of converging to the global optimum. They may converge to a local optimum instead. Therefore, statement 2 is False.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1: RELUs (Rectified Linear Units) are indeed monotonic, as they have a non-decreasing output for increasing inputs. Sigmoids, on the other hand, are monotonic, as they have a strictly increasing or strictly decreasing output for increasing inputs.

Statement 2: Neural networks trained with gradient descent do not always converge to the global optimum, as they can get stuck in local optima or saddle points.

Final Answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1: RELUs (rectified linear units) are indeed monotonic, meaning they always increase or stay the same as their input increases. On the other hand, sigmoids are not monotonic as they can have increasing and then decreasing portions.

Statement 2: Neural networks trained with gradient descent do not always converge to the global optimum. They can get stuck in local optima or saddle points, which may not be the best solution.

Based on these explanations, the correct answer is: C. 'True, False'

\*\*\*\*\*

Answer Option 5:

Statement 1: RELUs (Rectified Linear Units) are actually monotonic, as they have a linear positive slope for positive inputs. Sigmoids, on the other hand, are monotonic as they always have a non-decreasing or non-increasing behavior.

Statement 2: Neural networks trained with gradient descent do not have a guarantee to converge to the global optimum. It is possible for them to converge to a local optimum instead.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: RELUs (Rectified Linear Units) are actually monotonic, as they do not have negative values and are always increasing or staying constant. Sigmoids, on the other hand, are not monotonic as they have a curve shape and their derivative can be negative in certain regions.

Statement 2: Neural networks trained with gradient descent do not always converge to the global optimum. They can get stuck in local optima or saddle points, which may not be the best solution.

Final answer: C (True, False)

\*\*\*\*\*

Answer Option 7:

Statement 1: RELUs (Rectified Linear Units) are indeed monotonic, as they are non-decreasing for positive inputs. Sigmoids, on the other hand, are monotonic as they are always increasing or always decreasing. Therefore, Statement 1 is False.

Statement 2: Neural networks trained with gradient descent do not always converge to the global optimum. The optimization process can sometimes get stuck in local optima or saddle points. Therefore, Statement 2 is False.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: RELUs are actually monotonic. When the input is positive, the RELU function returns the input itself, which is a linear and monotonic function. When the input is negative, the RELU function returns zero, which is also a constant and monotonic function. Therefore, statement 1 is False.

Statement 2: Neural networks trained with gradient descent do not always converge to the global optimum. They may converge to a local minimum instead, depending on the specific problem and the network architecture. Therefore, statement 2 is also False.

Final answer: B (False, False)

\*\*\*\*\*

Answer Option 9:

Statement 1: RELUs (Rectified Linear Units) are actually monotonic, as they have a constant positive slope for positive input values. Sigmoids are also monotonic, as they have a positive or negative slope throughout their range. Therefore, statement 1 is false.

Statement 2: Neural networks trained with gradient descent do not always converge to the global optimum. They can get stuck in local optima, which may or may not be close to the global optimum.

Therefore, statement 2 is also false.

Final answer: B

\*\*\*\*\*

Question 54:

'The numerical output of a sigmoid node in a neural network:', A. 'Is unbounded, encompassing all real numbers.', B. 'Is unbounded, encompassing all integers.', C. 'Is bounded between 0 and 1.', D. 'Is bounded between -1 and 1.'

=====

Answer Option 1:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Answer Option 2:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Answer Option 3:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Answer Option 4:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C.

\*\*\*\*\*

Answer Option 5:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Answer Option 6:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Answer Option 7:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Answer Option 8:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Answer Option 9:

The numerical output of a sigmoid node in a neural network is bounded between 0 and 1. Final answer: C

\*\*\*\*\*

Question 55:

'Which of the following can only be used when training data are linearly separable?', A. 'Linear hard-margin SVM.', B. 'Linear Logistic Regression.', C. 'Linear Soft margin SVM.', D. 'The centroid method.'

=====

Answer Option 1:

The linear hard-margin SVM can only be used when training data are linearly separable. Linear Logistic Regression and Linear Soft margin SVM can handle both linearly separable and non-linearly separable data. The centroid method is not limited to linearly separable data either. Final answer: A

\*\*\*\*\*

Answer Option 2:

The option that can only be used when training data are linearly separable is A. Linear hard-margin SVM. In linear hard-margin SVM, the algorithm requires the data to be perfectly separable with a hyperplane. If the data is not linearly separable, the algorithm will not converge. Final answer: A

\*\*\*\*\*

Answer Option 3:

The only option among the given choices that requires linearly separable data is the Linear hard-margin SVM. Both Linear Logistic Regression and Linear Soft margin SVM can handle data that is not linearly separable using various techniques such as kernel functions and soft margin parameters. The centroid method is not limited to linearly separable data either. Final answer: A

\*\*\*\*\*

Answer Option 4:

The method that can only be used when training data are linearly separable is the Linear Hard-Margin SVM (Support Vector Machine). Linear Logistic Regression, Linear Soft-margin SVM, and the Centroid method can be used even when training data are not linearly separable. Final answer: A.

\*\*\*\*\*

Answer Option 5:

The only option among the given choices that requires linearly separable data is the linear hard-margin SVM (Support Vector Machine). Both linear logistic regression and linear soft-margin SVM can handle non-linearly separable data by introducing a penalty for misclassifications. The centroid method does not explicitly require linear separability. Final answer: A

\*\*\*\*\*

Answer Option 6:

The method that can only be used when the training data are linearly separable is the Linear Hard-Margin SVM (Support Vector Machine). Linear Hard-Margin SVM aims to find a hyperplane that can perfectly separate the data into their respective classes. Linear Logistic Regression, Linear Soft-Margin SVM, and the centroid method can handle non-linearly separable data as well. Final answer: A.

\*\*\*\*\*

Answer Option 7:

The linear hard-margin SVM (Support Vector Machine) can only be used when training data are linearly separable. Linear Logistic Regression and Linear Soft-margin SVM can also handle linearly separable data, but they can also handle cases where the data may not be perfectly separable. The centroid method is not restricted to linearly separable data and can be used for both linear and non-linear data. Final answer: A

\*\*\*\*\*

Answer Option 8:

The linear hard-margin support vector machine (SVM) can only be used when training data are linearly separable. This is because the hard-margin SVM aims to find a hyperplane that perfectly separates the data into different classes. Linear logistic regression, linear soft-margin SVM, and the centroid method can be used even when the training data are not linearly separable. Final answer: A

\*\*\*\*\*

Answer Option 9:

The linear hard-margin Support Vector Machine (SVM) can only be used when training data are linearly separable. Linear Logistic Regression and Linear Soft margin SVM can handle non-linearly separable data by introducing soft margins and allowing for a certain amount of misclassification. The centroid method is a clustering algorithm and is not specifically designed for linearly separable data. Final answer: A

\*\*\*\*\*



Question 56:

'Which of the following are the spatial clustering algorithms?', A. 'Partitioning based clustering', B. 'K-means clustering', C. 'Grid based clustering', D. 'All of the above'

=====

Answer Option 1:

The spatial clustering algorithms include partitioning-based clustering, k-means clustering, and grid-based clustering. Therefore, the correct answer is D. 'All of the above'.

\*\*\*\*\*

Answer Option 2:

The spatial clustering algorithms include partitioning based clustering, such as K-means clustering, and grid-based clustering. Therefore, the correct answer is D, 'All of the above'.

\*\*\*\*\*

Answer Option 3:

Spatial clustering algorithms are used to group data points based on their spatial proximity. Partitioning based clustering, K-means clustering, and grid-based clustering are all examples of spatial clustering algorithms. Therefore, the correct answer is D. Final answer: D.

\*\*\*\*\*

Answer Option 4:

Spatial clustering algorithms are used to group data points based on their spatial proximity. Partitioning based clustering, K-means clustering, and grid-based clustering are examples of spatial clustering algorithms. Therefore, the correct answer is D, 'All of the above'.

\*\*\*\*\*

Answer Option 5:

Spatial clustering algorithms refer to techniques that group data based on their spatial proximity. The algorithms listed, including partitioning-based clustering, k-means clustering, and grid-based clustering, are all examples of spatial clustering algorithms. Therefore, the correct answer is D. 'All of the above'.

\*\*\*\*\*

Answer Option 6:

Spatial clustering algorithms are used to group data points based on their spatial proximity. Partitioning based clustering, K-means clustering, and grid-based clustering are all examples of spatial clustering algorithms. Therefore, the correct answer is D. 'All of the above'.

\*\*\*\*\*

Answer Option 7:

Spatial clustering algorithms are used to group data points based on their spatial proximity. Partitioning based clustering, K-means clustering, and grid-based clustering are all examples of spatial clustering algorithms. Therefore, the correct answer is D. Final answer: D

\*\*\*\*\*

Answer Option 8:

Spatial clustering algorithms refer to techniques that group spatially related data points together. Partitioning based clustering, K-means clustering, and grid-based clustering are all examples of spatial clustering algorithms. Therefore, the correct answer is D: 'All of the above'.

\*\*\*\*\*

Answer Option 9:

Spatial clustering algorithms are used to find clusters or groups in spatial data based on their proximity. Both partitioning-based clustering and grid-based clustering are examples of spatial clustering algorithms. K-means clustering, however, is not specifically designed for spatial data and can be used for general clustering tasks. Therefore, the correct answer is: C

\*\*\*\*\*

Question 57:

'Statement 1| The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers. Statement 2| Any decision boundary that we get from a generative model with classconditional Gaussian distributions could in principle be reproduced with an SVM and a polynomial kernel of degree less than or equal to three.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. Support vector machines (SVMs) aim to find decision boundaries with the maximum margin, which can help reduce generalization error.

Statement 2 is false. While SVMs with polynomial kernels can approximate certain decision boundaries, they are not able to perfectly reproduce any decision boundary obtained from a generative model with class-conditional Gaussian distributions. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: The statement is generally true. Support vector machines (SVMs) aim to maximize the margin between decision boundaries, which can help improve generalization performance compared to other linear classifiers. Therefore, the statement is mostly true.

Statement 2: The statement is false. Although SVMs can approximate decision boundaries obtained from generative models, it is not always possible to exactly reproduce any decision boundary using an SVM and a polynomial kernel of degree less than or equal to three. Therefore, the statement is false.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. The maximum margin decision boundaries that support vector machines (SVMs) construct aim to maximize the separation between different classes, which can reduce the generalization error.

Statement 2 is false. While SVMs are powerful and flexible classifiers, not all decision boundaries obtained from a generative model with class-conditional Gaussian distributions can be reproduced with an SVM and a polynomial kernel of degree less than or equal to three.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. Maximum margin decision boundaries constructed by support vector machines (SVM) have been shown to have lower generalization error compared to other linear classifiers.

Statement 2 is false. Not all decision boundaries obtained from generative models with class-conditional Gaussian distributions can be reproduced with an SVM and a polynomial kernel of degree less than or equal to three. The SVM with a polynomial kernel of degree less than or equal to three may not be able to capture complex decision boundaries that can be represented by the generative models.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1: The statement is true. Support vector machines (SVMs) aim to maximize the margin between the decision boundary and the training samples. This margin maximization helps to reduce generalization error and improve the model's performance.

Statement 2: The statement is false. While it is true that some decision boundaries obtained from generative models with class-conditional Gaussian distributions can be reproduced by SVMs, it is not always possible to do so with a polynomial kernel of degree less than or equal to three. SVMs with polynomial kernels of higher degrees may be needed to accurately reproduce certain decision boundaries.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers. This statement is generally true. Support vector machines (SVMs) aim to find the maximum margin decision boundary, which helps in achieving better generalization performance.

Statement 2: Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could, in principle, be reproduced with an SVM and a polynomial kernel of degree less than or equal to three. This statement is false. The flexibility of SVMs with polynomial kernels is limited, and they may not be able to reproduce any arbitrary decision boundary obtained from a generative model with Gaussian distributions.

Final Answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers. This statement is generally true, as support vector machines (SVMs) aim to find a decision boundary with the maximum margin, which can help in reducing the generalization error.

Statement 2: Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could, in principle, be reproduced with an SVM and a polynomial kernel of degree less than or equal to three. This statement is false. While SVMs can be powerful classifiers, they cannot perfectly reproduce any decision boundary obtained from a generative model with class-conditional Gaussian distributions. SVMs make use of a different approach to find decision boundaries, and their choice of kernel and parameters can impact the shape and flexibility of the boundary they create.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers. This statement is true. Support vector machines (SVMs) are known for their ability to find the maximum margin decision boundary, which can lead to better generalization and lower error.

Statement 2: Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could, in principle, be reproduced with an SVM and a polynomial kernel of degree less than or equal to three. This statement is false. While SVMs are powerful classifiers, they cannot necessarily reproduce any decision boundary obtained from a generative model with class-conditional Gaussian distributions. The ability to reproduce such boundaries depends on the complexity and non-linearity of the data.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: "The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers." This statement is generally true. Support vector machines (SVMs) aim to maximize the margin between classes, which can help reduce overfitting and improve generalization.

Statement 2: "Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could, in principle, be reproduced with an SVM and a polynomial kernel of degree less than or equal to three." This statement is false. SVMs and generative models have different approaches to modeling and decision boundary construction. While SVMs can be powerful, they may not be able to fully reproduce decision boundaries created by generative models with specific assumptions, such as class-conditional Gaussian distributions.

Final answer: C

\*\*\*\*\*

Question 58:

'Statement 1| L2 regularization of linear models tends to make models more sparse than L1 regularization. Statement 2| Residual connections can be found in ResNets and Transformers.', A. 'True, True', B.

'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is False. L1 regularization tends to make models more sparse than L2 regularization, not the other way around.

Statement 2 is True. Residual connections can be found in both ResNets and Transformers.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: L2 regularization of linear models does not make models more sparse; it rather reduces the impact of large weights on the model.

Statement 2: Residual connections are commonly found in ResNets and not Transformers.

Final Answer: C - 'True, False'

\*\*\*\*\*

Answer Option 3:

Statement 1: L2 regularization of linear models does not tend to make models more sparse than L1 regularization. In fact, L1 regularization encourages sparsity by driving some of the coefficients to exactly zero.

Statement 2: Residual connections can be found in ResNets and Transformers. Residual connections are a key component of ResNet architectures and are also utilized in Transformers.

Final Answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1: L2 regularization of linear models does not make models more sparse. It helps to reduce the impact of large weight values but does not directly lead to sparsity. Therefore, Statement 1 is False.

Statement 2: Residual connections can be found in both ResNets and Transformers. They are used to mitigate the vanishing gradient problem and aid in the flow of gradients during training. Therefore,

Statement 2 is True.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1: L2 regularization of linear models does not tend to make models more sparse than L1 regularization. L1 regularization encourages sparsity by shrinking the coefficients of less important features to zero, whereas L2 regularization penalizes large coefficients but does not lead to sparsity.

Statement 2: Residual connections can be found in both ResNets and Transformers. Residual connections enable the flow of information from previous layers to subsequent layers, aiding in the gradient flow and improving model performance.

Final answer: C (True, False)

\*\*\*\*\*

Answer Option 6:

Statement 1: L2 regularization of linear models does not make models more sparse, it actually tends to distribute the weight values more evenly across all features rather than encourage sparsity. Therefore,

Statement 1 is False.

Statement 2: Residual connections can indeed be found in both ResNets and Transformers. Residual connections are used to improve the flow of information and address the vanishing gradient problem.

Therefore, Statement 2 is True.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: L2 regularization of linear models does not tend to make models more sparse than L1 regularization. In fact, L1 regularization encourages sparsity by introducing a penalty term that promotes the selection of a subset of features. So, statement 1 is False.

Statement 2: Residual connections can indeed be found in both ResNets and Transformers. Residual connections help to address the vanishing gradient problem and aid in the training of deeper neural networks. So, statement 2 is True.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: L2 regularization of linear models tends to make models more sparse than L1 regularization.

This statement is False. L1 regularization is known to induce sparsity in linear models by shrinking the coefficients of less important features to zero, while L2 regularization encourages small but non-zero coefficients for all features.

Statement 2: Residual connections can be found in ResNets and Transformers.

This statement is True. Residual connections are a key component of Residual Networks (ResNets) which help to mitigate the vanishing gradient problem. They are also used in Transformers to allow information to flow across different layers more easily.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: L2 regularization of linear models does not make models more sparse. In fact, L2 regularization tends to distribute the weight values more evenly across the features, but does not directly result in sparsity. Therefore, Statement 1 is false.

Statement 2: Residual connections can indeed be found in both ResNets and Transformers. Residual connections allow the model to bypass certain layers, allowing for better flow of gradients and alleviating the vanishing gradient problem. Therefore, Statement 2 is true.

Final answer: C (False, True)

\*\*\*\*\*

Question 59:

'Suppose we like to calculate  $P(H|E, F)$  and we have no conditional independence information. Which of the following sets of numbers are sufficient for the calculation?', A. ' $P(E, F), P(H), P(E|H), P(F|H)$ ', B. ' $P(E, F), P(H), P(E, F|H)$ ', C. ' $P(H), P(E|H), P(F|H)$ ', D. ' $P(E, F), P(E|H), P(F|H)$ '

=====

Answer Option 1:

To calculate  $P(H|E, F)$  without any conditional independence information, we need the joint probabilities  $P(E, F)$ , the marginal probability  $P(H)$ , and the conditional probabilities  $P(E|H)$  and  $P(F|H)$ . Therefore, the set of numbers that are sufficient for the calculation is A. Final answer: A.

=====

Answer Option 2:

In order to calculate  $P(H|E, F)$ , we need the conditional probabilities  $P(E|H)$  and  $P(F|H)$ , as well as the prior probability  $P(H)$ . Therefore, the set of numbers that are sufficient for the calculation is D. Final answer: D.

=====

Answer Option 3:

To calculate  $P(H|E, F)$ , we need the joint probabilities  $P(E, F)$ , as well as the conditional probabilities  $P(E|H)$  and  $P(F|H)$ . Therefore, the set of numbers that are sufficient for the calculation is D. Final answer: D.

=====

Answer Option 4:

To calculate  $P(H|E, F)$  without any conditional independence information, we need the joint probabilities  $P(E, F), P(H)$ , as well as the conditional probabilities  $P(E|H)$  and  $P(F|H)$ . Therefore, the set of numbers that is sufficient for the calculation is A. Final answer: A

=====

Answer Option 5:

To calculate  $P(H|E, F)$  without any conditional independence information, we need to have  $P(E, F), P(H), P(E|H)$ , and  $P(F|H)$ . These probabilities are necessary to compute the conditional probability. Therefore, the correct answer is A. Final answer: A

=====

Answer Option 6:

To calculate  $P(H|E, F)$  without any conditional independence information, we need the joint probabilities  $P(E, F)$ , the marginal probability  $P(H)$ , and the conditional probabilities  $P(E|H)$  and  $P(F|H)$ . Therefore, the set of numbers that would be sufficient for the calculation is A. Final answer: A.

=====

Answer Option 7:

To calculate  $P(H|E, F)$ , we need the probabilities of E and F ( $P(E, F)$ ), the probability of H ( $P(H)$ ), the probability of E given H ( $P(E|H)$ ), and the probability of F given H ( $P(F|H)$ ). The set of numbers that includes all these probabilities is A. Final answer: A.

=====

Answer Option 8:

To calculate  $P(H|E, F)$  without any conditional independence information, we require the joint probabilities  $P(E, F)$ , the marginal probability  $P(H)$ , and the conditional probabilities  $P(E|H)$  and  $P(F|H)$ . Therefore, the set of numbers sufficient for the calculation is A.  $P(E, F), P(H), P(E|H), P(F|H)$ . Final answer: A.

=====

Answer Option 9:

To calculate  $P(H|E, F)$  without any conditional independence information, we need to have the joint probabilities  $P(E, F)$ , the probability of H  $P(H)$ , and the conditional probabilities  $P(E|H)$  and  $P(F|H)$ . Therefore, the set of numbers that are sufficient for the calculation is: A. ' $P(E, F), P(H), P(E|H), P(F|H)$ '. Final answer: A

=====

Question 60:

'Which among the following prevents overfitting when we perform bagging?', A. 'The use of sampling with replacement as the sampling technique', B. 'The use of weak classifiers', C. 'The use of classification algorithms which are not prone to overfitting', D. 'The practice of validation performed on every classifier trained'

=====

Answer Option 1:

When performing bagging, one technique to prevent overfitting is the use of sampling with replacement as the sampling technique. This means that each base classifier is trained on a bootstrap sample from the original dataset, allowing for some instances to be repeated in each sample. This helps in reducing overfitting and increasing the diversity among the base classifiers. Final answer: A

\*\*\*\*\*

Answer Option 2:

When performing bagging, one way to prevent overfitting is by using sampling with replacement as the sampling technique. This means that each classifier is trained on a random subset of the data, and some samples may be selected multiple times. This helps to introduce randomness and reduce the likelihood of individual classifiers fitting the noise in the data. Using weak classifiers can also help prevent overfitting in bagging. Final answer: A

\*\*\*\*\*

Answer Option 3:

When performing bagging, one way to prevent overfitting is to use sampling with replacement as the sampling technique. This helps to introduce randomness and reduce the chance of the model fitting the training data too closely. The use of weak classifiers, classification algorithms not prone to overfitting, and validation on every classifier trained are not specific to bagging and do not directly address overfitting in the context of bagging. Final answer: A

\*\*\*\*\*

Answer Option 4:

When performing bagging, one way to prevent overfitting is by using sampling with replacement as the sampling technique, which means that each base classifier is trained on a random subset of the original data with replacement. This helps to introduce diversity and reduce the chance of overfitting. Using weak classifiers can also help prevent overfitting in bagging. Final answer: A

\*\*\*\*\*

Answer Option 5:

When we perform bagging, one method that helps prevent overfitting is the use of sampling with replacement as the sampling technique. This allows for the creation of different training sets for each classifier, reducing the chance of overfitting to specific data points. The use of weak classifiers is a technique associated with boosting, not bagging. The use of classification algorithms that are not prone to overfitting could be a consideration but is not specific to bagging. The practice of validation performed on every classifier trained is not necessarily a technique for preventing overfitting in bagging. Final answer: A

\*\*\*\*\*

Answer Option 6:

When performing bagging, a technique used to reduce overfitting, the use of sampling with replacement as the sampling technique helps prevent overfitting. This is because it introduces randomness and diversity into the training samples. The use of weak classifiers is a characteristic of boosting, not bagging. The use of classification algorithms not prone to overfitting can help, but it is not specific to bagging. The practice of validation performed on every classifier trained is not directly related to preventing overfitting in bagging. Final answer: A

\*\*\*\*\*

Answer Option 7:

When performing bagging, one approach to prevent overfitting is to use sampling with replacement as the sampling technique, which is known as bootstrap sampling. This helps in creating diverse subsets of data for training each classifier. The use of weak classifiers can also help prevent overfitting as they have less complexity and are less likely to memorize the training data. The use of classification algorithms that are not prone to overfitting and the practice of validation performed on every classifier trained are not specific to bagging and are not directly aimed at preventing overfitting. Final answer: A

\*\*\*\*\*

Answer Option 8:

When performing bagging, one way to prevent overfitting is by using sampling with replacement as the sampling technique. This means that each base classifier is trained on a bootstrap sample, which helps to introduce variability and reduce overfitting. The use of weak classifiers is a concept more related to boosting. The use of classification algorithms that are not prone to overfitting is a good strategy, but it is not specific to bagging. The practice of validation performed on every classifier trained is not directly related to preventing overfitting in bagging. Final answer: A

\*\*\*\*\*

Answer Option 9:

When performing bagging, the use of sampling with replacement as the sampling technique helps prevent overfitting. This is because sampling with replacement introduces randomness and diversity into the training data for each classifier. The use of weak classifiers in bagging can also help prevent overfitting by reducing the complexity of individual models. The practice of validation performed on every classifier trained does not directly prevent overfitting but can help measure the performance and generalization of the ensemble model. The use of classification algorithms which are not prone to overfitting is not specifically related to bagging but can be a general strategy to prevent overfitting. Final answer: A

\*\*\*\*\*

Question 61:

'Statement 1] PCA and Spectral Clustering (such as Andrew Ng's) perform eigendecomposition on two different matrices. However, the size of these two matrices are the same. Statement 2] Since classification is a special case of regression, logistic regression is a special case of linear regression.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: PCA and Spectral Clustering indeed perform eigendecomposition on different matrices, but the size of these matrices is not necessarily the same. False.

Statement 2: Classification and regression are two distinct problems, and while logistic regression is a type of classification algorithm, it is not a special case of linear regression. False.

Final answer: B

=====

Answer Option 2:

Statement 1 is false. PCA and spectral clustering both involve eigendecomposition, but they operate on different matrices. The size of the matrices may not necessarily be the same.

Statement 2 is true. Classification can be seen as a special case of regression, and logistic regression is a special case of linear regression that is specifically used for classification tasks.

Final answer: C

=====

Answer Option 3:

Statement 1 is false. PCA (Principal Component Analysis) and Spectral Clustering perform eigendecomposition on different matrices. The size of these matrices may vary depending on the specific implementation or problem.

Statement 2 is true. Classification is indeed a special case of regression, and logistic regression is a special case of linear regression where the output is binomial (two classes). Final answer: C

=====

Answer Option 4:

Statement 1 is false. PCA and Spectral Clustering both involve eigendecomposition, but they perform it on different matrices. The size of these matrices can be different.

Statement 2 is false. While classification can be seen as a special case of regression, logistic regression is not a special case of linear regression. Logistic regression is used for binary classification, while linear regression is used for predicting continuous values. Final answer: C

=====

Answer Option 5:

Statement 1: PCA and Spectral Clustering perform eigendecomposition on different matrices, but the size of these matrices may not necessarily be the same. Therefore, Statement 1 is False.

Statement 2: Classification and regression are different types of machine learning problems, and while logistic regression is a type of classification algorithm, it is not a special case of linear regression.

Therefore, Statement 2 is False.

Final answer: B

=====

Answer Option 6:

Statement 1 is False. PCA (Principal Component Analysis) and Spectral Clustering do not perform eigendecomposition on the same type of matrices. While PCA performs eigendecomposition on the covariance matrix, Spectral Clustering performs eigendecomposition on the graph Laplacian matrix.

Statement 2 is True. Logistic regression is indeed a special case of linear regression, where the output is transformed using a logistic function to perform binary classification.

Final answer: C (True, False)

=====

Answer Option 7:

Statement 1 is false. PCA (Principal Component Analysis) performs eigendecomposition on the covariance matrix of the data, while spectral clustering performs eigendecomposition on the similarity matrix or graph Laplacian.

Statement 2 is true. Logistic regression can be seen as a special case of linear regression where the output is transformed using a logistic function. Therefore, logistic regression is a special case of linear regression.

Final answer: C

=====

Answer Option 8:

Statement 1 is false. PCA performs eigendecomposition on the covariance matrix, while Spectral Clustering performs eigendecomposition on the graph Laplacian matrix. The size of these two matrices is not necessarily the same.

Statement 2 is true. Classification can be seen as a special case of regression, where the output is discrete instead of continuous. Logistic regression is a special case of linear regression, where the output is transformed using a sigmoid function to obtain probabilistic class predictions.

Final answer: C

=====

Answer Option 9:

Statement 1 is true. PCA (Principal Component Analysis) performs eigendecomposition on the covariance matrix, while Spectral Clustering performs eigendecomposition on the Laplacian matrix. However, the size of these two matrices is the same.



Statement 2 is false. While classification can be seen as a special case of regression, logistic regression is not a special case of linear regression. Logistic regression is specifically designed for binary classification tasks, while linear regression is used for regression tasks. Final answer: C

\*\*\*\*\*

Question 62:

'Statement 1] The Stanford Sentiment Treebank contained movie reviews, not book reviews. Statement 2] The Penn Treebank has been used for language modeling.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true, as the Stanford Sentiment Treebank indeed contains movie reviews, not book reviews. Statement 2 is false, as the Penn Treebank is commonly used for tasks such as part-of-speech tagging and parsing, not specifically for language modeling. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. The Stanford Sentiment Treebank is a dataset that contains movie reviews, not book reviews. Statement 2 is false. The Penn Treebank is a dataset commonly used for part-of-speech tagging and syntactic parsing, not specifically for language modeling. Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1: The Stanford Sentiment Treebank contained movie reviews, not book reviews. This statement is true.

Statement 2: The Penn Treebank has been used for language modeling. This statement is also true.

Final Answer: A. 'True, True'

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. The Stanford Sentiment Treebank is a dataset that contains movie reviews, not book reviews. Statement 2 is false. The Penn Treebank is primarily used for parsing and not specifically for language modeling. Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is true, as the Stanford Sentiment Treebank indeed contains movie reviews, not book reviews. Statement 2 is false, as the Penn Treebank is commonly used for tasks like syntactic parsing, not specifically for language modeling. Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: The Stanford Sentiment Treebank contained movie reviews, not book reviews. This statement is true.

Statement 2: The Penn Treebank has been used for language modeling. This statement is also true.

Final answer: A. 'True, True'

\*\*\*\*\*

Answer Option 7:

Statement 1: The Stanford Sentiment Treebank contains movie reviews, not book reviews. This statement is true.

Statement 2: The Penn Treebank has been used for language modeling. This statement is also true.

Therefore, the correct answer is A. 'True, True'.

\*\*\*\*\*

Answer Option 8:

Statement 1 is true. The Stanford Sentiment Treebank is a dataset that contains movie reviews, not book reviews.

Statement 2 is false. The Penn Treebank is a dataset that has been widely used for various tasks in natural language processing, including part-of-speech tagging and syntactic parsing, but not specifically for language modeling.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The Stanford Sentiment Treebank contained movie reviews, not book reviews.

Statement 2: The Penn Treebank has been used for language modeling.

Statement 1 is true. The Stanford Sentiment Treebank is a dataset that contains movie reviews.

Statement 2 is false. The Penn Treebank is a dataset that has been widely used for POS tagging and parsing, not specifically for language modeling.

Final answer: C (True, False)

\*\*\*\*\*

Question 63:

'What is the dimensionality of the null space of the following matrix?  $A = \begin{bmatrix} 3 & 2 & -9 \\ -6 & -4 & 18 \end{bmatrix}$ , [12, 8, -36]]', A. '0', B. '1', C. '2', D. '3'

=====

Answer Option 1:

To find the dimensionality of the null space of a matrix, we need to find the number of linearly independent vectors that satisfy the equation  $Ax = 0$ , where  $A$  is the given matrix.

Using row reduction or Gaussian elimination, we can find that the given matrix  $A$  reduces to  $\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ .

We can see that the second and third rows are all zeros, indicating that they are linearly dependent on the first row. Therefore, the dimensionality of the null space is 2.

Final answer: C

=====

Answer Option 2:

To find the dimensionality of the null space of a matrix, we need to find the null space itself and then count the number of linearly independent vectors in the null space.

To find the null space, we need to solve the equation  $A \cdot x = 0$ , where  $A$  is the given matrix and  $x$  is the vector in the null space.

Solving this equation, we find that the null space of matrix  $A$  is spanned by the vector  $[3, 2, -1]$ .

Since there is only one linearly independent vector in the null space, the dimensionality of the null space is 1.

Final answer: B

=====

Answer Option 3:

To find the dimensionality of the null space of a matrix, we need to find the number of linearly independent vectors that satisfy the equation  $Ax = 0$ , where  $A$  is the given matrix.

By row reducing the matrix  $A$  to reduced row-echelon form, we have:

$\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

The third row is a linear combination of the first two rows, which means there is only one linearly independent vector that satisfies the equation  $Ax = 0$ .

Therefore, the dimensionality of the null space of matrix  $A$  is 1.

Final answer: B

=====

Answer Option 4:

To find the dimensionality of the null space of a matrix  $A$ , we need to find the dimension of the space spanned by the vectors that satisfy the equation  $Ax = 0$ .

The matrix  $A = \begin{bmatrix} 3 & 2 & -9 \\ -6 & -4 & 18 \end{bmatrix}$  can be row reduced to  $\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ .

From this row-reduced form, we can see that the second and third rows are all zeros. This implies that there are two free variables, and hence the dimensionality of the null space of matrix  $A$  is 2.

Final answer: C

=====

Answer Option 5:

To find the dimensionality of the null space of a matrix, we need to find the null space or the kernel of the matrix. The null space is the set of vectors that when multiplied by the matrix result in the zero vector.

To find the null space, we can perform row reduction on the matrix  $A$ . After row reduction, we get the following matrix:

$\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

From this, we can see that the second and third rows are all zeros, which means that the second and third variables are free variables. Therefore, the dimensionality of the null space is 2.

Final answer: C

=====

Answer Option 6:

The dimensionality of the null space of a matrix is equal to the number of linearly independent vectors in the null space. To find the null space, we need to find the vectors  $x$  such that  $Ax = 0$ .

For the given matrix  $A = \begin{bmatrix} 3 & 2 & -9 \\ -6 & -4 & 18 \end{bmatrix}$ , we can row-reduce it to echelon form and solve for the null space.

After row reduction, we get  $\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ .

This implies that the third variable is a free variable, and the null space is non-trivial. Therefore, the dimensionality of the null space is 1.

Final answer: B

=====

Answer Option 7:

To determine the dimensionality of the null space of a matrix, we need to find the null space or kernel of the matrix. This can be done by finding the solutions to the equation  $Ax = 0$ , where  $A$  is the matrix.

For the given matrix  $A = \begin{bmatrix} 3 & 2 & -9 \\ -6 & -4 & 18 \end{bmatrix}$ , we can solve the equation  $Ax = 0$  to find the null space.

Row reducing the augmented matrix  $[A|0]$  gives us:

$\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

From this, we can see that the second and third variables ( $x_2$  and  $x_3$ ) are free variables, while the first variable ( $x_1$ ) is a basic variable. This means that the null space has 2 dimensions.

Final answer: C

=====

Answer Option 8:

To find the dimensionality of the null space of a matrix, we need to find the solutions to the equation  $Ax = 0$ , where  $A$  is the given matrix.

For the matrix  $A = \begin{bmatrix} 3 & 2 & -9 \\ -6 & -4 & 18 \\ 12 & 8 & -36 \end{bmatrix}$ , we can row reduce the matrix to determine the null space.

After row reduction, we get  $\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ .

The second and third rows are all zeroes, which means the variables  $x_2$  and  $x_3$  are free variables. The first row, however, has no free variables.

Therefore, the dimensionality of the null space is 2 (since there are two free variables). Final answer: C

\*\*\*\*\*

Answer Option 9:

To find the dimensionality of the null space of a matrix, we need to find the number of linearly independent vectors that span the null space.

To find the null space, we need to solve the equation  $Ax = 0$ , where  $A$  is the given matrix.

For the matrix  $A = \begin{bmatrix} 3 & 2 & -9 \\ -6 & -4 & 18 \\ 12 & 8 & -36 \end{bmatrix}$ , we can row reduce the augmented matrix  $[A|0]$  to find the null space:

Row reducing  $[A|0]$ :

$$R_2 = 2R_1 + R_2$$

$$R_3 = -4R_1 + R_3$$

$$R_3 = -2R_2 + R_3$$

$$R_2 = -R_2/2$$

$$R_3 = -R_3/2$$

The row reduced augmented matrix is:

$$\begin{bmatrix} 1 & 2/3 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

From this row reduced form, we can see that there are two leading ones and one free variable. This means that the dimensionality of the null space is 1.

Final answer: B

\*\*\*\*\*

Question 64:

'What are support vectors?', A. 'The examples farthest from the decision boundary.', B. 'The only examples necessary to compute  $f(x)$  in an SVM.', C. 'The data centroid.', D. 'All the examples that have a non-zero weight  $\alpha_k$  in a SVM.'

=====

Answer Option 1:

Support vectors are the examples that have a non-zero weight  $\alpha_k$  in a support vector machine (SVM). They are the critical data points that help define the decision boundary in an SVM. Final answer: D  
\*\*\*\*\*

Answer Option 2:

Support vectors are the examples that have a non-zero weight  $\alpha_k$  in a Support Vector Machine (SVM). They are the examples that lie closest to the decision boundary and play a crucial role in defining the decision boundary and maximizing the margin in an SVM. Final answer: D  
\*\*\*\*\*

Answer Option 3:

Support vectors are the examples that have a non-zero weight  $\alpha_k$  in a Support Vector Machine (SVM). They are the data points that lie closest to the decision boundary and play a crucial role in defining the decision boundary of the SVM. Final answer: D  
\*\*\*\*\*

Answer Option 4:

Support vectors are the examples that have a non-zero weight  $\alpha_k$  in a Support Vector Machine (SVM). They are the data points that lie on or close to the decision boundary and have the most influence on the final classification. Final answer: D  
\*\*\*\*\*

Answer Option 5:

Support vectors are the examples that have a non-zero weight  $\alpha_k$  in a support vector machine (SVM). They are the critical data points that define the decision boundary of the SVM. The examples farthest from the decision boundary do not necessarily have to be support vectors. Final answer: D  
\*\*\*\*\*

Answer Option 6:

Support vectors are the examples that have a non-zero weight  $\alpha_k$  in a Support Vector Machine (SVM). They are the data points that lie closest to the decision boundary and play a crucial role in defining the decision boundary of the SVM. Final answer: D  
\*\*\*\*\*

Answer Option 7:

Support vectors in a support vector machine (SVM) are the examples that have a non-zero weight  $\alpha_k$ . They are the examples that lie closest to the decision boundary and play a crucial role in defining the decision boundary of the SVM. Final answer: D  
\*\*\*\*\*

Answer Option 8:

Support vectors are the examples that lie closest to the decision boundary in a Support Vector Machine (SVM). They are the data points that influence the construction of the decision boundary and are necessary to compute  $f(x)$  in an SVM. They are not necessarily the farthest examples from the decision boundary. Final answer: B  
\*\*\*\*\*

Answer Option 9:

Support vectors are the examples that have a non-zero weight  $\alpha_k$  in a Support Vector Machine (SVM). They are the critical data points that define the decision boundary of the SVM. The examples farthest from the decision boundary are known as outliers, not support vectors. The only examples necessary to compute  $f(x)$  in an SVM do not necessarily need to be support vectors. The concept of "data centroid" is not directly related to support vectors. Final answer: D  
\*\*\*\*\*

Question 65:

'Statement 1] Word2Vec parameters were not initialized using a Restricted Boltzman Machine. Statement 2] The tanh function is a nonlinear activation function.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: False. Word2Vec parameters are typically initialized randomly and not using a Restricted Boltzmann Machine (RBM).

Statement 2: True. The tanh function is a common nonlinear activation function used in neural networks.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is False because Word2Vec parameters can be initialized using a Restricted Boltzmann Machine (RBM). Statement 2 is True because the tanh function is indeed a nonlinear activation function commonly used in neural networks. Final Answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is false. Word2Vec parameters are typically initialized randomly and not using a Restricted Boltzmann Machine (RBM).

Statement 2 is true. The tanh function is indeed a nonlinear activation function.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is False. Word2Vec parameters are typically initialized randomly and not using a Restricted Boltzmann Machine.

Statement 2 is True. The tanh function is indeed a nonlinear activation function commonly used in neural networks.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is false. Word2Vec parameters are typically initialized randomly and not using a Restricted Boltzmann Machine.

Statement 2 is true. The tanh function is a commonly used nonlinear activation function in neural networks.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is false. Word2Vec parameters are typically initialized randomly, and not using a Restricted Boltzman Machine.

Statement 2 is true. The tanh function is indeed a nonlinear activation function.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: False. Word2Vec parameters can be initialized using a Restricted Boltzmann Machine (RBM) or other techniques like random initialization.

Statement 2: True. The tanh function is indeed a nonlinear activation function commonly used in neural networks.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1 is false. Word2Vec parameters are typically initialized randomly and not using a Restricted Boltzmann Machine.

Statement 2 is true. The tanh function is indeed a nonlinear activation function.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1 - Word2Vec parameters were not initialized using a Restricted Boltzman Machine: False

Statement 2 - The tanh function is a nonlinear activation function: True

The parameters of Word2Vec are typically initialized randomly and not using a Restricted Boltzman Machine. The tanh function is indeed a common nonlinear activation function used in neural networks. Final answer: C

\*\*\*\*\*

Question 66:

'If your training loss increases with number of epochs, which of the following could be a possible issue with the learning process?', A. 'Regularization is too low and model is overfitting', B. 'Regularization is too high and model is underfitting', C. 'Step size is too large', D. 'Step size is too small'

=====

Answer Option 1:

If the training loss increases with the number of epochs, it indicates that the model's performance is deteriorating over time. This could be due to several reasons, but one possible issue is that the regularization is too low and the model is overfitting. Overfitting occurs when the model becomes too complex and starts to memorize the training data instead of learning general patterns. So, the correct answer is A.

=====

Answer Option 2:

If the training loss increases with the number of epochs, it suggests that the learning process is not converging and there may be an issue with the learning process. One possible issue could be that the regularization is too low, causing the model to overfit the training data. Final answer: A

=====

Answer Option 3:

If the training loss increases with the number of epochs, a possible issue with the learning process could be that the regularization is too high and the model is underfitting. High regularization can cause the model to be too constrained and unable to capture the underlying patterns in the data. Final answer: B

=====

Answer Option 4:

If the training loss increases with the number of epochs, a possible issue with the learning process could be that the regularization is too low and the model is overfitting. Overfitting occurs when a model becomes too complex and starts to memorize the training data instead of learning general patterns. Final answer: A

=====

Answer Option 5:

If the training loss increases with the number of epochs, it could be an indication that the model is overfitting. Overfitting occurs when the model learns the training data too well and fails to generalize to unseen data. Therefore, the possible issue with the learning process could be that the regularization is too low. Final answer: A

=====

Answer Option 6:

If the training loss increases with the number of epochs, it could be a possible issue of the learning process known as overfitting. Overfitting occurs when the model learns too much from the training data and fails to generalize well to unseen data. This suggests that the regularization is too low, allowing the model to overfit the training data. Final answer: A

=====

Answer Option 7:

If the training loss increases with the number of epochs, it suggests that the model is not learning properly. One possible issue could be that the regularization is too low and the model is overfitting the training data. Overfitting occurs when the model becomes too complex and starts to memorize the training examples instead of generalizing well to new data. Final answer: A

=====

Answer Option 8:

If the training loss increases with the number of epochs, it could be an indication that the model is overfitting. Overfitting occurs when the model learns the training data too well, including noise and outliers, and fails to generalize well on unseen data. Regularization is a technique used to prevent overfitting, so if the regularization is too low, it could lead to overfitting. Final answer: A

=====

Answer Option 9:

If the training loss increases with the number of epochs, it could indicate that the model is overfitting, which means that the model is too complex and is fitting the training data too closely, resulting in poor generalization to unseen data. In this case, the regularization might be too low. Therefore, the possible issue with the learning process could be that the regularization is too low and the model is overfitting.

Final answer: A

=====

Question 67:

'Say the incidence of a disease D is about 5 cases per 100 people (i.e.,  $P(D) = 0.05$ ). Let Boolean random variable D mean a patient "has disease D" and let Boolean random variable TP stand for "tests positive." Tests for disease D are known to be very accurate in the sense that the probability of testing positive when you have the disease is 0.99, and the probability of testing negative when you do not have the disease is 0.97. What is  $P(D | TP)$ , the posterior probability that you have disease D when the test is positive?', A. '0.0495', B. '0.078', C. '0.635', D. '0.97'

=====

Answer Option 1:

To find the posterior probability  $P(D | TP)$ , we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

We are given:

$$P(D) = 0.05 \text{ (incidence of disease D)}$$

$$P(TP | D) = 0.99 \text{ (probability of testing positive given that you have the disease)}$$

$$P(TP | \neg D) = 1 - 0.97 = 0.03 \text{ (probability of testing positive given that you do not have the disease)}$$

To calculate  $P(TP)$ , we can use the law of total probability:

$$P(TP) = P(TP | D) * P(D) + P(TP | \neg D) * P(\neg D)$$

$$P(\neg D) = 1 - P(D) = 1 - 0.05 = 0.95$$

Plugging in the values:

$$P(TP) = 0.99 * 0.05 + 0.03 * 0.95 = 0.0495 + 0.0285 = 0.078$$

Using Bayes' theorem:

$$P(D | TP) = (0.99 * 0.05) / 0.078 = 0.0495 / 0.078 = 0.635$$

Final answer: C. '0.635'

\*\*\*\*\*

Answer Option 2:

To find the posterior probability  $P(D | TP)$ , we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

Given that  $P(TP | D) = 0.99$ ,  $P(D) = 0.05$ , and  $P(TP)$  can be calculated using the law of total probability:

$$P(TP) = P(TP | D) * P(D) + P(TP | \neg D) * P(\neg D)$$

$$P(TP) = 0.99 * 0.05 + (1 - 0.97) * (1 - 0.05)$$

$$P(TP) = 0.0495 + 0.015$$

$$P(TP) = 0.0645$$

Now, substituting these values into Bayes' theorem:

$$P(D | TP) = (0.99 * 0.05) / 0.0645$$

$$P(D | TP) = 0.765$$

Therefore, the correct answer is not among the options provided.

\*\*\*\*\*

Answer Option 3:

To find the posterior probability  $P(D | TP)$ , we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

We know that  $P(TP | D) = 0.99$ ,  $P(D) = 0.05$ , and we can calculate  $P(TP)$  using the law of total probability:

$$P(TP) = P(TP | D) * P(D) + P(TP | \text{not } D) * P(\text{not } D)$$

$$P(TP) = 0.99 * 0.05 + (1 - 0.97) * (1 - 0.05) = 0.099 + 0.015 = 0.114$$

Now we can substitute the values back into the Bayes' theorem equation:

$$P(D | TP) = (0.99 * 0.05) / 0.114 = 0.0435 / 0.114 = 0.3816$$

So, the posterior probability  $P(D | TP)$  is approximately 0.3816.

Final Answer: C. '0.635'

\*\*\*\*\*

Answer Option 4:

To calculate the posterior probability, we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

Given that  $P(TP | D) = 0.99$ ,  $P(D) = 0.05$ , and  $P(TP)$ , we need to calculate  $P(TP)$ .

$$P(TP) = (P(TP | D) * P(D)) + (P(TP | \neg D) * P(\neg D))$$

$$= (0.99 * 0.05) + ((1 - 0.97) * (1 - 0.05))$$

$$= 0.0495 + 0.045$$

$$= 0.0945$$



Now, we can calculate  $P(D | TP)$ :

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

$$= (0.99 * 0.05) / 0.0945$$

$$= 0.0495 / 0.0945$$

$$= 0.5238095238$$

Rounding to three decimal places, the answer is approximately 0.524. Final answer: C

\*\*\*\*\*

Answer Option 5:

To find the posterior probability  $P(D | TP)$ , we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

We are given that  $P(TP | D) = 0.99$  and  $P(TP | \text{not } D) = 1 - P(\text{negative test} | \text{not } D) = 1 - 0.97 = 0.03$ .

$P(D) = 0.05$  (given)

To find  $P(TP)$ , we can use the law of total probability:

$$P(TP) = P(TP | D) * P(D) + P(TP | \text{not } D) * P(\text{not } D)$$

$$P(TP) = 0.99 * 0.05 + 0.03 * (1 - 0.05) = 0.0495 + 0.0285 = 0.078$$

Now we can substitute the values into Bayes' theorem:

$$P(D | TP) = (0.99 * 0.05) / 0.078 = 0.0495 / 0.078 = 0.635$$

Final answer: C. '0.635'

\*\*\*\*\*

Answer Option 6:

To find the posterior probability  $P(D | TP)$ , we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

From the problem statement, we know that  $P(D) = 0.05$ ,  $P(TP | D) = 0.99$ , and  $P(TP | \sim D) = 0.97$ , where  $\sim D$  represents not having disease D.

To calculate  $P(TP)$ , we can use the law of total probability:

$$P(TP) = P(TP | D) * P(D) + P(TP | \sim D) * P(\sim D)$$

Since  $P(\sim D) = 1 - P(D) = 1 - 0.05 = 0.95$ , we have:

$$P(TP) = 0.99 * 0.05 + 0.97 * 0.95 = 0.0495 + 0.9215 = 0.971$$

Now, we can substitute the values into Bayes' theorem:

$$P(D | TP) = (0.99 * 0.05) / 0.971 = 0.0495 / 0.971 = 0.051$$

Therefore, the posterior probability  $P(D | TP)$  is approximately 0.051. Final answer: None of the above

\*\*\*\*\*

Answer Option 7:

To find  $P(D | TP)$ , we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

$P(TP | D) = 0.99$  (probability of testing positive given the disease)

$P(D) = 0.05$  (probability of having the disease)

$$P(TP) = P(TP | D) * P(D) + P(TP | \sim D) * P(\sim D)$$

$P(TP | \sim D) = 1 - P(\sim TP | \sim D) = 1 - 0.97 = 0.03$  (probability of testing positive given not having the disease)

$P(\sim D) = 1 - P(D) = 1 - 0.05 = 0.95$  (probability of not having the disease)

$$P(TP) = (0.99 * 0.05) + (0.03 * 0.95) = 0.0495 + 0.0285 = 0.078$$

Now we can calculate  $P(D | TP)$ :

$$P(D | TP) = (0.99 * 0.05) / 0.078 = 0.0495 / 0.078 = 0.635$$

Therefore, the posterior probability that you have disease D when the test is positive is approximately 0.635. Final answer: C

\*\*\*\*\*

Answer Option 8:

To find the posterior probability that you have disease D when the test is positive, we can use Bayes' theorem:

$$P(D | TP) = (P(TP | D) * P(D)) / P(TP)$$

$P(TP | D)$  is the probability of testing positive when you have the disease, which is given as 0.99.

$P(D)$  is the probability of having the disease, which is given as 0.05.

$P(TP)$  is the probability of testing positive, which can be calculated using the law of total probability:

$$P(TP) = (P(TP | D) * P(D)) + (P(TP | \sim D) * P(\sim D))$$

$P(TP | \sim D)$  is the probability of testing positive when you do not have the disease, which is given as  $1 - 0.97 = 0.03$ .

$P(\sim D)$  is the probability of not having the disease, which is  $1 - P(D) = 1 - 0.05 = 0.95$ .

Plugging in the values, we have:

$$P(TP) = (0.99 * 0.05) + (0.03 * 0.95) = 0.0495 + 0.0285 = 0.078$$

Now we can calculate the posterior probability:

$$P(D | TP) = (0.99 * 0.05) / 0.078 = 0.0495 / 0.078 = 0.635$$

Final Answer: C. '0.635'

\*\*\*\*\*

Answer Option 9:

To solve this problem, we can use Bayes' theorem. Let's denote the probability of having disease D as  $P(D) = 0.05$ , the probability of testing positive given that you have the disease as  $P(TP|D) = 0.99$ , and the probability of testing negative given that you do not have the disease as  $P(TN|\neg D) = 0.97$ .

Bayes' theorem states:

$$P(D|TP) = (P(TP|D) * P(D)) / P(TP)$$

To calculate  $P(TP)$ , we need to consider both the probability of testing positive when you have the disease and the probability of testing positive when you do not have the disease.

$$P(TP) = (P(TP|D) * P(D)) + (P(TP|\neg D) * P(\neg D))$$

$$= (0.99 * 0.05) + (1 - 0.97) * (1 - 0.05)$$

$$P(TP) = 0.099 + 0.0015$$

$$P(TP) = 0.1005$$

Now we can calculate  $P(D|TP)$  using Bayes' theorem:

$$P(D|TP) = (P(TP|D) * P(D)) / P(TP)$$

$$= (0.99 * 0.05) / 0.1005$$

$$P(D|TP) = 0.0495 / 0.1005$$

$$P(D|TP) = 0.4925$$

Final answer: A. '0.0495'

\*\*\*\*\*

Question 68:

'Statement 1| Traditional machine learning results assume that the train and test sets are independent and identically distributed. Statement 2| In 2017, COCO models were usually pretrained on ImageNet',

A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. Traditional machine learning assumes that the train and test sets are independent and identically distributed.

Statement 2 is false. In 2017, COCO models were not usually pretrained on ImageNet.

Final answer: C

=====

Answer Option 2:

Statement 1 is true. Traditional machine learning assumes that the train and test sets are independent and identically distributed.

Statement 2 is false. In 2017, COCO models were not usually pretrained on ImageNet.

Final answer: C

=====

Answer Option 3:

Statement 1 is true. Traditional machine learning assumes that the train and test sets are independent and identically distributed.

Statement 2 is false. In 2017, COCO models were not usually pretrained on ImageNet. Final answer: C

=====

Answer Option 4:

Statement 1 is true. Traditional machine learning assumes that the train and test sets are independent and identically distributed.

Statement 2 is true. In 2017, COCO models were commonly pretrained on ImageNet.

Final answer: A

=====

Answer Option 5:

Statement 1 is true. Traditional machine learning assumes that the train and test sets are independent and identically distributed to ensure that the model generalizes well to unseen data. Statement 2 is false.

COCO models, which are commonly used for object detection and segmentation tasks, were not typically pretrained on the ImageNet dataset in 2017. Final answer: C

=====

Answer Option 6:

Statement 1 is true. Traditional machine learning assumes that the train and test sets are independent and identically distributed to ensure that the model generalizes well to unseen data.

Statement 2 is false. In 2017, COCO models were typically pre-trained on the COCO dataset, not ImageNet.

Final answer: C

=====

Answer Option 7:

Statement 1: Traditional machine learning results assume that the train and test sets are independent and identically distributed. This statement is generally true, as traditional machine learning assumes that the data used for training and testing come from the same distribution.

Statement 2: In 2017, COCO models were usually pretrained on ImageNet. This statement is true, as pretraining on ImageNet was a common practice in 2017 to improve the performance of models on the COCO dataset.

Final answer: C

=====

Answer Option 8:

Statement 1 is true. Traditional machine learning assumes that the train and test sets are independent and identically distributed (i.i.d.), meaning that the data in the test set comes from the same distribution as the data in the train set.

Statement 2 is false. In 2017, COCO models were not usually pretrained on ImageNet. COCO (Common Objects in Context) is a large-scale dataset for object detection, segmentation, and captioning.

ImageNet is a dataset used for object recognition tasks. Final answer: C

=====

Answer Option 9:

Statement 1: Traditional machine learning results assume that the train and test sets are independent and identically distributed. This statement is true as it is a common assumption in traditional machine learning algorithms.

Statement 2: In 2017, COCO models were usually pretrained on ImageNet. This statement is false. COCO (Common Objects in Context) is a large-scale dataset for object detection, segmentation, and captioning. While pretraining on ImageNet is a common practice in computer vision, it is not specifically tied to COCO models.

Final answer: C

=====

Question 69:

'Statement 1] The values of the margins obtained by two different kernels  $K_1(x, x_0)$  and  $K_2(x, x_0)$  on the same training set do not tell us which classifier will perform better on the test set. Statement 2] The activation function of BERT is the GELU.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. The values of the margins obtained by different kernels on the same training set do not directly indicate which classifier will perform better on the test set.

Statement 2 is false. The activation function of BERT (Bidirectional Encoder Representations from Transformers) is not GELU (Gaussian Error Linear Unit), it uses the "gelu" activation function.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. The values of the margins obtained by different kernels on the same training set do not directly indicate which classifier will perform better on the test set.

Statement 2 is false. The activation function of BERT is not GELU (Gaussian Error Linear Unit), but rather the transformer-based BERT uses the "gelu" activation function.

Final Answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. The values of the margins obtained by different kernels do not directly indicate which classifier will perform better on the test set.

Statement 2 is false. The activation function of BERT is not the GELU. It uses the "BertActivation" function, which is a combination of the "gelu" activation function and the "tanh" activation function.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1: The values of the margins obtained by two different kernels  $K_1(x, x_0)$  and  $K_2(x, x_0)$  on the same training set do not tell us which classifier will perform better on the test set.

This statement is true. The margins obtained by different kernels on the same training set do not provide direct information about the performance of the classifiers on the test set.

Statement 2: The activation function of BERT is the GELU.

This statement is false. The activation function used in BERT (Bidirectional Encoder Representations from Transformers) is the "GELU" (Gaussian Error Linear Unit).

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is true. The values of the margins obtained by different kernels on the same training set do not directly indicate which classifier will perform better on the test set. The margin values are specific to the training set and may not generalize well.

Statement 2 is false. The activation function of BERT (Bidirectional Encoder Representations from Transformers) is not the GELU (Gaussian Error Linear Unit). BERT uses the "Transformer" architecture, which employs the "self-attention" mechanism. Final Answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is true. The values of the margins obtained by different kernels on the same training set do not directly indicate which classifier will perform better on the test set. The performance of a classifier depends on various factors, including the choice of kernel, regularization, and other hyperparameters.

Statement 2 is false. The activation function used in BERT (Bidirectional Encoder Representations from Transformers) is not the GELU (Gaussian Error Linear Unit). BERT uses the "BertActivation" function, which is a combination of various activation functions like ReLU, tanh, and sigmoid.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: The values of the margins obtained by two different kernels  $K_1(x, x_0)$  and  $K_2(x, x_0)$  on the same training set do not tell us which classifier will perform better on the test set.

This statement is true. The values of the margins obtained by different kernels on the same training set do not directly indicate the performance of the classifier on a test set. The margins only provide information about the separation of the training data.

Statement 2: The activation function of BERT is the GELU.

This statement is false. The activation function of BERT (Bidirectional Encoder Representations from Transformers) is not GELU. BERT uses the transformer architecture with a specific type of activation called the "Bertgelu" activation function.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1 is true. The values of the margins obtained by different kernels on the same training set do not directly indicate which classifier will perform better on the test set. The choice of kernel can affect the decision boundary and generalization performance, but it is not solely determined by the margins.

Statement 2 is false. The activation function of BERT (Bidirectional Encoder Representations from Transformers) is not the GELU (Gaussian Error Linear Unit). BERT uses the transformer architecture with the activation function called the "Transformer Activation Function," which is a combination of the GELU and the Layer Normalization. Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The values of the margins obtained by two different kernels  $K_1(x, x_0)$  and  $K_2(x, x_0)$  on the same training set do not tell us which classifier will perform better on the test set.

This statement is true. The margin values obtained by different kernels do not directly indicate which classifier will perform better on the test set. Other factors such as the complexity of the model, the training data, and the generalization ability of the classifier also play a role.

Statement 2: The activation function of BERT is the GELU.

This statement is false. The activation function used in BERT is not GELU (Gaussian Error Linear Unit) but rather the rectified linear unit (ReLU) activation function. Final answer: C

\*\*\*\*\*

Question 70:

'Which of the following is a clustering algorithm in machine learning?', A. 'Expectation Maximization', B. 'CART', C. 'Gaussian Naïve Bayes', D. 'Apriori'

=====

Answer Option 1:

Expectation Maximization (EM) is a popular clustering algorithm that is used to estimate the parameters of probabilistic models. CART is a decision tree algorithm, Gaussian Naïve Bayes is a classification algorithm, and Apriori is an association rule learning algorithm. Final answer: A

\*\*\*\*\*

Answer Option 2:

Expectation Maximization is a clustering algorithm in machine learning. CART is a decision tree algorithm, Gaussian Naïve Bayes is a classification algorithm, and Apriori is an association rule learning algorithm. Final answer: A

\*\*\*\*\*

Answer Option 3:

Expectation Maximization is a clustering algorithm commonly used in machine learning. CART is a decision tree algorithm, Gaussian Naïve Bayes is a classification algorithm, and Apriori is an association rule mining algorithm. Final answer: A

\*\*\*\*\*

Answer Option 4:

Expectation Maximization is a clustering algorithm in machine learning. CART is a decision tree algorithm, Gaussian Naïve Bayes is a classification algorithm, and Apriori is an association rule learning algorithm. Final answer: A

\*\*\*\*\*

Answer Option 5:

Expectation Maximization is a clustering algorithm commonly used in machine learning. CART is a decision tree algorithm, Gaussian Naïve Bayes is a classification algorithm, and Apriori is an association rule learning algorithm. Final answer: A

\*\*\*\*\*

Answer Option 6:

Expectation Maximization is a clustering algorithm in machine learning that is used to estimate the parameters of a statistical model given a set of observed data. CART is a decision tree algorithm, Gaussian Naïve Bayes is a classification algorithm, and Apriori is an algorithm used for association rule mining. Final answer: A

\*\*\*\*\*

Answer Option 7:

Expectation Maximization is a clustering algorithm commonly used to estimate the parameters of a statistical model. CART is a decision tree algorithm used for classification and regression tasks. Gaussian Naïve Bayes is a probabilistic classification algorithm, and Apriori is an algorithm used for association rule mining. Final answer: A

\*\*\*\*\*

Answer Option 8:

Expectation Maximization is a clustering algorithm commonly used to estimate the parameters of probabilistic models, such as Gaussian Mixture Models. CART is a decision tree algorithm used for classification and regression tasks. Gaussian Naïve Bayes is a probabilistic classification algorithm. Apriori is an algorithm used for association rule mining, not clustering. Final answer: A

\*\*\*\*\*

Answer Option 9:

Expectation Maximization is a clustering algorithm commonly used to estimate the parameters of a mixture model. CART is a classification and regression tree algorithm. Gaussian Naïve Bayes is a classification algorithm based on Bayes' theorem. Apriori is an association rule learning algorithm. Final answer: A

\*\*\*\*\*

Question 71:

'You've just finished training a decision tree for spam classification, and it is getting abnormally bad performance on both your training and test sets. You know that your implementation has no bugs, so what could be causing the problem?', A. 'Your decision trees are too shallow.', B. 'You need to increase the learning rate.', C. 'You are overfitting.', D. 'None of the above.'

=====

Answer Option 1:

If a decision tree is performing abnormally bad on both the training and test sets, it suggests that the model is overfitting. Overfitting occurs when the model is too complex and captures noise or irrelevant patterns from the training data, leading to poor generalization on unseen data. Final answer: C

\*\*\*\*\*

Answer Option 2:

Given that the decision tree is performing poorly on both training and test sets, it suggests that the model is not generalizing well. This is indicative of overfitting, where the model becomes too complex and captures noise or irrelevant patterns in the training data. Therefore, the correct answer is C. 'You are overfitting.'

\*\*\*\*\*

Answer Option 3:

If the decision tree is performing abnormally bad on both the training and test sets, it suggests that the model is overfitting. Overfitting occurs when the model learns the training data too well and fails to generalize to new, unseen data. Increasing the learning rate or having shallow decision trees would not typically cause this issue. Final answer: C

\*\*\*\*\*

Answer Option 4:

If the decision tree is performing abnormally bad on both the training and test sets, it is likely that the model is overfitting. Overfitting occurs when the model learns the training data too well, including noise and outliers, and does not generalize well to unseen data. Therefore, the correct answer is C, 'You are overfitting'. Final answer: C

\*\*\*\*\*

Answer Option 5:

If the decision tree is performing abnormally bad on both the training and test sets, the most likely cause is overfitting. Overfitting occurs when the model learns the training data too well and fails to generalize to new, unseen data. Shallow decision trees (option A) and learning rate (option B) do not directly cause bad performance on both sets. Final answer: C

\*\*\*\*\*

Answer Option 6:

If a decision tree is performing abnormally bad on both the training and test sets, it suggests that the model is not generalizing well to unseen data. This is a common symptom of overfitting, where the model becomes too complex and starts to memorize the training data instead of learning the underlying patterns. Therefore, the correct answer is C. 'You are overfitting.'

\*\*\*\*\*

Answer Option 7:

If a decision tree is performing abnormally bad on both the training and test sets, it suggests that the model is overfitting. Overfitting occurs when a model learns to fit the training data too closely and does not generalize well to new, unseen data. This can happen if the decision trees are too deep and complex, capturing noise and outliers in the training data. Therefore, the correct answer is C - "You are overfitting."

\*\*\*\*\*

Answer Option 8:

If your decision tree is getting abnormally bad performance on both the training and test sets, it is likely that you are overfitting the model. Overfitting occurs when the model fits the training data too closely and fails to generalize well to new, unseen data. This can happen if the decision trees are too deep and complex, leading to overly specific patterns in the training data. Increasing the learning rate or having shallow decision trees is less likely to cause such a significant drop in performance. Final answer: C

\*\*\*\*\*

Answer Option 9:

If the decision tree is performing abnormally bad on both the training and test sets, it suggests that the model is not generalizing well and is exhibiting poor performance. This is typically a symptom of overfitting, where the model has learned the training data too well and fails to generalize to new, unseen data. Therefore, the correct answer is C: 'You are overfitting.'

\*\*\*\*\*

Question 72:

'K-fold cross-validation is', A. 'linear in K', B. 'quadratic in K', C. 'cubic in K', D. 'exponential in K'

=====

Answer Option 1:

K-fold cross-validation is a technique used in machine learning to assess the performance of a model by dividing the data into K subsets (folds) and iteratively training and evaluating the model on different combinations of these folds. The computational complexity of K-fold cross-validation is linear in K since it requires K iterations. Final answer: A.

\*\*\*\*\*

Answer Option 2:

K-fold cross-validation is a technique used in machine learning to assess the performance of a model. It involves splitting the data into K equal-sized folds, using K-1 folds for training and the remaining fold for testing. This process is repeated K times, with each fold serving as the test set once.

The time complexity of K-fold cross-validation is linear in K, as the number of iterations is proportional to K. Therefore, the correct answer is A.

\*\*\*\*\*

Answer Option 3:

K-fold cross-validation is a technique used to evaluate the performance of a machine learning model by dividing the data into K subsets or folds. Each fold is used as a validation set while the remaining folds are used for training. The process is repeated K times, with each fold serving as the validation set exactly once.

The complexity of K-fold cross-validation is linear in K, as the number of iterations increases linearly with K. Final answer: A

\*\*\*\*\*

Answer Option 4:

K-fold cross-validation is a technique used to assess the performance of a machine learning model. It involves dividing the dataset into K equal-sized folds and performing training and testing K times, with each fold being used as the testing set once and the remaining K-1 folds used as the training set. The performance metrics obtained from each run are then averaged to give an overall performance estimate.

The computational complexity of K-fold cross-validation is linear in K, as it requires K iterations. Final answer: A

\*\*\*\*\*

Answer Option 5:

K-fold cross-validation is a technique used to assess the performance of a machine learning model. It involves dividing the data into K subsets, or folds, and training the model K times, each time using K-1 folds as training data and the remaining fold as a validation set. The performance of the model is then averaged across the K iterations.

The time complexity of K-fold cross-validation is linear in K, as the model is trained K times. Final answer: A

\*\*\*\*\*

Answer Option 6:

K-fold cross-validation is a technique used to evaluate the performance of a machine learning model by dividing the data into K equal-sized subsets or folds. The model is trained and evaluated K times, with each fold serving as the validation set once. The results are then averaged to obtain a more robust performance estimate.

K-fold cross-validation is linear in K, as the number of iterations or runs is directly proportional to the value of K. Final answer: A

\*\*\*\*\*

Answer Option 7:

K-fold cross-validation is a technique used to assess the performance of a machine learning model by splitting the data into K equal-sized folds and iteratively using K-1 folds for training and the remaining fold for testing. It is not linear, quadratic, cubic, or exponential in K. It has a time complexity that is generally proportional to the number of folds K. Since the time complexity is directly proportional to K, the correct answer would be: B. 'quadratic in K'.

\*\*\*\*\*

Answer Option 8:

K-fold cross-validation is a technique used to assess the performance of a machine learning model. It involves dividing the data into K subsets or folds, training the model on K-1 folds, and evaluating it on the remaining fold. This process is repeated K times, with each fold serving as the validation set once.

The computational complexity of k-fold cross-validation is linear in K. Each fold requires training the model and evaluating its performance, but the time required for each fold does not increase exponentially or quadratically with K. Final answer: A

\*\*\*\*\*

Answer Option 9:

K-fold cross-validation is a technique used to evaluate the performance of a machine learning model. It involves splitting the data into K equal-sized folds, where one fold is used as the validation set and the remaining K-1 folds are used for training. This process is repeated K times, with each fold serving as the validation set once. The performance metrics are then averaged across the K iterations. The

computational complexity of K-fold cross-validation is linear in K, as the training and validation steps are performed K times. Final answer: A

\*\*\*\*\*



Question 73:

'Statement 1| Industrial-scale neural networks are normally trained on CPUs, not GPUs. Statement 2| The ResNet-50 model has over 1 billion parameters.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is False. Industrial-scale neural networks are typically trained on GPUs due to their parallel processing capabilities, which can significantly speed up training.

Statement 2 is False. The ResNet-50 model has approximately 25 million parameters, not over 1 billion.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is False. Industrial-scale neural networks are typically trained on GPUs (Graphics Processing Units) due to their superior parallel processing capabilities.

Statement 2 is False. The ResNet-50 model has around 25 million parameters, not over 1 billion.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1: False. Industrial-scale neural networks are normally trained on GPUs, not CPUs, due to the parallel processing capabilities of GPUs.

Statement 2: False. The ResNet-50 model does not have over 1 billion parameters. It has around 25 million parameters.

Final answer: C. 'True, False'

\*\*\*\*\*

Answer Option 4:

Statement 1 is False. Industrial-scale neural networks are typically trained on GPUs (Graphics Processing Units) due to their parallel processing capabilities, which can significantly speed up training compared to CPUs.

Statement 2 is False. The ResNet-50 model has approximately 25 million parameters, not over 1 billion. Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is False. Industrial-scale neural networks are typically trained on GPUs due to their ability to handle parallel processing and faster computations compared to CPUs.

Statement 2 is False. The ResNet-50 model has approximately 25 million parameters, not over 1 billion.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is false. Industrial-scale neural networks are typically trained on GPUs (Graphics Processing Units) due to their parallel processing capabilities, which can significantly speed up training times compared to CPUs.

Statement 2 is false. The ResNet-50 model has approximately 25 million parameters, not over 1 billion.

Final answer: D

\*\*\*\*\*

Answer Option 7:

Statement 1: False. Industrial-scale neural networks are typically trained on GPUs (Graphics Processing Units) rather than CPUs (Central Processing Units) due to the parallel processing capabilities of GPUs.

Statement 2: False. The ResNet-50 model has around 25 million parameters, not over 1 billion.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1 is False. Industrial-scale neural networks are typically trained on GPUs (Graphics Processing Units) due to their parallel processing capabilities, which can significantly speed up the training process compared to CPUs.

Statement 2 is False. The ResNet-50 model has around 25 million parameters, not over 1 billion.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1 is False. Industrial-scale neural networks are typically trained on GPUs (Graphics Processing Units) due to their parallel processing capabilities, which significantly speeds up the training process compared to CPUs.

Statement 2 is False. The ResNet-50 model has approximately 25 million parameters, not over 1 billion.

Final answer: C

\*\*\*\*\*



Question 74:

'Given two Boolean random variables, A and B, where  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A \mid \neg B) = 1/4$ , what is  $P(A \mid B)$ ?' A. '1/6', B. '1/4', C. '3/4', D. '1'

=====

Answer Option 1:

We can use the conditional probability formula to solve this problem. According to the formula:

$$P(A \mid B) = P(A \wedge B) / P(B)$$

We are given  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A \mid \neg B) = 1/4$ .

To find  $P(A \wedge B)$ , we can use the formula:

$$P(A \wedge B) = P(A \mid \neg B) * P(\neg B)$$

$P(\neg B)$  can be found as  $1 - P(B)$ :

$$P(\neg B) = 1 - 1/3 = 2/3$$

Now, substituting the given values:

$$P(A \wedge B) = (1/4) * (2/3) = 1/6$$

Finally, substituting the calculated values back into the conditional probability formula:

$$P(A \mid B) = (1/6) / (1/3) = 1/2$$

Therefore, the correct answer is A. 1/6.

=====

Answer Option 2:

To find  $P(A \mid B)$ , we can use Bayes' theorem:

$$P(A \mid B) = (P(B \mid A) * P(A)) / P(B)$$

We are given  $P(A) = 1/2$  and  $P(B) = 1/3$ .

To find  $P(B \mid A)$ , we can use the conditional probability formula:

$$P(B \mid A) = P(A \text{ and } B) / P(A)$$

We are given  $P(A \mid \neg B) = 1/4$ . Since A and B are Boolean random variables, we can write  $P(A \text{ and } B)$  as  $P(A \mid \neg B) * P(\neg B)$ .  $P(\neg B)$  can be calculated as  $1 - P(B)$ .

$$P(\neg B) = 1 - P(B) = 1 - 1/3 = 2/3$$

$$P(B \mid A) = P(A \mid \neg B) * P(\neg B) = (1/4) * (2/3) = 1/6$$

Now we can substitute the values into Bayes' theorem:

$$P(A \mid B) = (P(B \mid A) * P(A)) / P(B) = (1/6) * (1/2) / (1/3) = 1/4$$

Final answer: B

=====

Answer Option 3:

To find  $P(A \mid B)$ , we can use Bayes' theorem:

$$P(A \mid B) = (P(B \mid A) * P(A)) / P(B)$$

We are given  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A \mid \neg B) = 1/4$ .

We know that  $P(B) = 1 - P(\neg B)$ , so  $P(\neg B) = 1 - P(B) = 1 - 1/3 = 2/3$ .

We also know that  $P(A \mid \neg B) = P(A \text{ and } \neg B) / P(\neg B)$ .

Since A and  $\neg B$  are mutually exclusive events,  $P(A \text{ and } \neg B) = 0$ .

Using the equation above, we can solve for  $P(A \mid B)$ :

$$P(A \mid B) = (P(B \mid A) * P(A)) / P(B)$$

$$= (P(B \text{ and } A) / P(A)) * P(A) / P(B)$$

$$= P(B \text{ and } A) / P(B)$$

$$= P(A \text{ and } B) / P(B)$$

Since A and B are independent events:

$$P(A \mid B) = P(A \text{ and } B) / P(B) = P(A) * P(B) / P(B) = P(A) = 1/2$$

Final answer: D. '1'

=====

Answer Option 4:

To solve this problem, we can use Bayes' theorem, which states that  $P(A \mid B) = (P(B \mid A) * P(A)) / P(B)$ .

We are given  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A \mid \neg B) = 1/4$ .

To find  $P(B \mid A)$ , we can use the complement rule:  $P(B \mid A) = 1 - P(\neg B \mid A) = 1 - P(A' \mid B) = 1 - (1 - P(A \mid B)) = P(A \mid B)$ .

Therefore,  $P(A \mid B) = (P(B \mid A) * P(A)) / P(B) = (P(A \mid B) * 1/2) / 1/3$ .

Simplifying, we get  $P(A \mid B) = (P(A \mid B) * 3/2)$ .

Multiplying both sides by 2/3, we get  $2/3 * P(A \mid B) = P(A \mid B)$ .

Subtracting  $P(A | B)$  from both sides, we get  $1/3 * P(A | B) = 0$ .

This implies that  $P(A | B) = 0$ .

Therefore, the correct answer is D. '1'.

\*\*\*\*\*

Answer Option 5:

We can use Bayes' theorem to calculate  $P(A | B)$ . Bayes' theorem states that:

$$P(A | B) = (P(B | A) * P(A)) / P(B)$$

We are given that  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A | \neg B) = 1/4$ .

We can rewrite  $P(A | \neg B)$  as  $P(A | B')$ .

$$P(A | B') = (P(B' | A) * P(A)) / P(B')$$

We know that  $P(B') = 1 - P(B) = 1 - 1/3 = 2/3$ .

Therefore,  $P(A | B') = (P(B' | A) * P(A)) / (2/3)$ .

We also know that  $P(B' | A) = 1 - P(B | A) = 1 - 1/4 = 3/4$ .

Substituting the values, we get:

$$P(A | B') = (3/4 * 1/2) / (2/3)$$

Simplifying further:

$$P(A | B') = 3/8 / (2/3)$$

$$= 3/8 * 3/2$$

$$= 9/16$$

Now, we can use the complement rule to find  $P(A | B)$ :

$$P(A | B) = 1 - P(A' | B)$$

$$= 1 - P(A | B')$$

$$= 1 - 9/16$$

$$= 7/16$$

Final Answer:  $P(A | B) = 7/16$ . Therefore, the correct answer is not listed.

\*\*\*\*\*

Answer Option 6:

To find  $P(A | B)$ , we can use Bayes' theorem:

$$P(A | B) = (P(B | A) * P(A)) / P(B)$$

We are given  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A | \neg B) = 1/4$ .

First, we need to calculate  $P(\neg B)$  using the complement rule:

$$P(\neg B) = 1 - P(B) = 1 - 1/3 = 2/3.$$

Now, we can calculate  $P(A \text{ and } \neg B)$  using the conditional probability formula:

$$P(A \text{ and } \neg B) = P(A | \neg B) * P(\neg B) = (1/4) * (2/3) = 1/6.$$

Next, we can calculate  $P(B | A)$  using Bayes' theorem:

$$P(B | A) = (P(A | B) * P(B)) / P(A)$$

$$P(B | A) = (P(A \text{ and } B)) / P(A)$$

$$P(B | A) = (1/6) / (1/2) = 1/3.$$

Finally, we can calculate  $P(A | B)$  using Bayes' theorem:

$$P(A | B) = (P(B | A) * P(A)) / P(B)$$

$$P(A | B) = (1/3) * (1/2) / (1/3)$$

$$P(A | B) = 1/2.$$

So, the answer is A. '1/6'.

\*\*\*\*\*

Answer Option 7:

To find  $P(A | B)$ , we can use Bayes' theorem:

$$P(A | B) = (P(B | A) * P(A)) / P(B)$$

We are given  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A | \neg B) = 1/4$ .

Using the complement rule, we can find  $P(\neg B) = 1 - P(B) = 1 - 1/3 = 2/3$ .

Now, we can find  $P(A \cap \neg B)$  using the conditional probability formula:

$$P(A \cap \neg B) = P(A | \neg B) * P(\neg B) = (1/4) * (2/3) = 1/6.$$

Using the law of total probability, we can find  $P(A \cap B)$ :

$$P(A \cap B) = P(A \cap \neg B) + P(A \cap B) = P(A) = 1/2.$$

Finally, we can find  $P(B)$ ,  $P(B | A)$ , and substitute them into Bayes' theorem:

$$P(B) = 1/3$$

$$P(B | A) = P(A \cap B) / P(A) = (1/2) / (1/2) = 1$$

$$P(A | B) = (P(B | A) * P(A)) / P(B) = (1 * 1/2) / (1/3) = 3/2$$

However, probabilities cannot exceed 1, so the correct answer is not listed. The correct answer is not provided.

\*\*\*\*\*

Answer Option 8:

To find  $P(A | B)$ , we can use Bayes' theorem:

$$P(A | B) = P(B | A) * P(A) / P(B)$$

We are given  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A | \neg B) = 1/4$ .

From the complement rule, we know that  $P(\neg B) = 1 - P(B) = 1 - 1/3 = 2/3$ .

Using the definition of conditional probability,  $P(A | \neg B) = P(A \cap \neg B) / P(\neg B)$ , we can calculate  $P(A \cap \neg B)$ :

$$P(A \cap \neg B) = P(A \cap \neg B) = P(A) - P(A \cap B) = P(A) - P(A | B) * P(B)$$

$$1/4 = 1/2 - P(A | B) * (1/3)$$

Now we can solve for  $P(A | B)$ :

$$1/4 = 1/2 - P(A | B) * (1/3)$$

$$1/4 + P(A | B) * (1/3) = 1/2$$

$$P(A | B) * (1/3) = 1/2 - 1/4 = 1/4$$

$$P(A | B) = (1/4) / (1/3) = 1/4 * 3/1 = 3/4$$

Final Answer: C. '3/4'

\*\*\*\*\*

Answer Option 9:

To determine  $P(A | B)$ , we can use Bayes' theorem:

$$P(A | B) = (P(B | A) * P(A)) / P(B)$$

We are given  $P(A) = 1/2$ ,  $P(B) = 1/3$ , and  $P(A | \neg B) = 1/4$ .

We can find  $P(B | A)$  using the complement rule:

$$P(B | A) = 1 - P(\neg B | A)$$

$$P(\neg B | A) = 1 - P(B | A)$$

Since  $P(A | \neg B) = 1/4$ , we can write:

$$P(A | \neg B) = P(A \cap \neg B) / P(\neg B)$$

$$P(A \cap \neg B) = P(A | \neg B) * P(\neg B) = (1/4) * (1 - 1/3) = 1/8$$

Now, we can find  $P(\neg B | A)$  using the complement rule:

$$P(\neg B | A) = 1 - P(B | A) = 1 - (1 - P(\neg B | A))$$

$$P(\neg B | A) = 1 - (1 - 1/8) = 1/8$$

We can now use Bayes' theorem to find  $P(A | B)$ :

$$P(A | B) = (P(B | A) * P(A)) / P(B)$$

$$P(A | B) = (1/8 * 1/2) / (1/3)$$

$$P(A | B) = 1/16 / (1/3)$$

$$P(A | B) = 1/16 * 3/1$$

$$P(A | B) = 3/16$$

Final answer: C

\*\*\*\*\*

Question 75:

'Existential risks posed by AI are most commonly associated with which of the following professors?', A. 'Nando de Freitas', B. 'Yann LeCun', C. 'Stuart Russell', D. 'Jitendra Malik'

=====

Answer Option 1:

Existential risks posed by AI are most commonly associated with Stuart Russell. Final answer: C

\*\*\*\*\*

Answer Option 2:

Existential risks posed by AI are most commonly associated with Stuart Russell. Final answer: C

\*\*\*\*\*

Answer Option 3:

Existential risks posed by AI are most commonly associated with Professor Stuart Russell. Final answer: C

\*\*\*\*\*

Answer Option 4:

Existential risks posed by AI are most commonly associated with Professor Stuart Russell. Final answer: C

\*\*\*\*\*

Answer Option 5:

Existential risks posed by AI are most commonly associated with Professor Stuart Russell. Final answer: C

\*\*\*\*\*

Answer Option 6:

Existential risks posed by AI are most commonly associated with Stuart Russell. Final answer: C

\*\*\*\*\*

Answer Option 7:

Existential risks posed by AI are most commonly associated with Professor Stuart Russell. Final answer: C

\*\*\*\*\*

Answer Option 8:

Existential risks posed by AI are most commonly associated with Stuart Russell. Final answer: C.

\*\*\*\*\*

Answer Option 9:

Existential risks posed by AI are most commonly associated with Professor Stuart Russell. Final answer: C

\*\*\*\*\*

Question 76:

'Statement 1| Maximizing the likelihood of logistic regression model yields multiple local optimums. Statement 2| No classifier can do better than a naive Bayes classifier if the distribution of the data is known.',

A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. Maximizing the likelihood of a logistic regression model can indeed yield multiple local optima.

Statement 2 is false. There are classifiers that can outperform a naive Bayes classifier even when the distribution of the data is known.

Final answer: C

=====

Answer Option 2:

Statement 1 is true. Maximizing the likelihood of a logistic regression model can indeed yield multiple local optima.

Statement 2 is false. While the naive Bayes classifier is often used as a baseline model, there are other classifiers that can outperform it, even when the distribution of the data is known.

Final answer: C

=====

Answer Option 3:

Statement 1 is true. Maximizing the likelihood of a logistic regression model can indeed result in multiple local optimums due to the non-convex nature of the optimization problem.

Statement 2 is false. While the naive Bayes classifier is a simple and efficient classifier, there are other classifiers that can perform better if the distribution of the data is known. Final Answer: C

=====

Answer Option 4:

Statement 1: Maximizing the likelihood of a logistic regression model can indeed yield multiple local optima. This statement is true.

Statement 2: It is not true that no classifier can do better than a naive Bayes classifier if the distribution of the data is known. There are other classifiers that can outperform naive Bayes depending on the specific characteristics of the data. This statement is false.

Final answer: C

=====

Answer Option 5:

Statement 1: Maximizing the likelihood of a logistic regression model can indeed lead to multiple local optima. So, Statement 1 is true.

Statement 2: While a naive Bayes classifier can perform well, it is not necessarily true that no classifier can do better if the data distribution is known. There can be other classifiers that can outperform the naive Bayes classifier in certain scenarios. So, Statement 2 is false.

Final answer: C

=====

Answer Option 6:

Statement 1: Maximizing the likelihood of a logistic regression model can indeed yield multiple local optima, so Statement 1 is true.

Statement 2: While naive Bayes classifiers have certain assumptions about the distribution of data, it is not true that no classifier can do better than a naive Bayes classifier if the distribution of the data is known. There are other classifiers that can outperform naive Bayes in certain cases, so Statement 2 is false.

Final answer: C

=====

Answer Option 7:

Statement 1: Maximizing the likelihood of a logistic regression model does not necessarily yield multiple local optima. It can have a single global optimum.

Statement 2: A naive Bayes classifier assumes independence between features and makes strong assumptions about the data distribution. While it can perform well in certain scenarios, there can be classifiers that can outperform it if the distribution of the data is known.

Final answer: C, 'True, False'

=====

Answer Option 8:

Statement 1 is true. Maximizing the likelihood of a logistic regression model can indeed yield multiple local optimums.

Statement 2 is false. While a naive Bayes classifier can perform well, there are cases where other classifiers, such as support vector machines or random forests, can outperform it even when the distribution of the data is known.

Final answer: C

=====

Answer Option 9:

Statement 1 is true. Maximizing the likelihood of a logistic regression model can result in multiple local optimums due to the non-convex nature of the problem.

Statement 2 is false. While naive Bayes classifier is a simple and efficient classifier, it makes the assumption of independence among features, which may not hold true for all data distributions. There can be classifiers that perform better than naive Bayes if the data distribution is known.

Final answer: C

\*\*\*\*\*



Question 77:

'For Kernel Regression, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting:', A. 'Whether kernel function is Gaussian versus triangular versus box-shaped', B. 'Whether we use Euclidian versus L1 versus L $\infty$  metrics', C. 'The kernel width', D. 'The maximum height of the kernel function'

=====

Answer Option 1:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The width of the kernel function determines the size of the neighborhood around each data point that contributes to the regression. A smaller width can lead to overfitting, while a larger width can lead to underfitting. Final answer: C

\*\*\*\*\*

Answer Option 2:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The kernel width determines the size of the local neighborhood around each data point, impacting how much smoothness or flexibility the model exhibits. Final answer: C

\*\*\*\*\*

Answer Option 3:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The kernel width determines the smoothness of the regression model and can have a significant impact on the fitting of the data. Final answer: C

\*\*\*\*\*

Answer Option 4:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The kernel width determines the scale at which neighboring points influence the prediction, and it can greatly impact the smoothness and flexibility of the model. Final answer: C

\*\*\*\*\*

Answer Option 5:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The kernel width determines the smoothness of the regression function and how much the model can fit to individual data points. Final answer: C

\*\*\*\*\*

Answer Option 6:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The choice of kernel width determines the extent of the influence of each training point on the predictions, and a wider kernel width can lead to underfitting while a narrower kernel width can lead to overfitting. Final answer: C.

\*\*\*\*\*

Answer Option 7:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The choice of kernel width determines the size of the neighborhood around each data point that influences the prediction. A narrower kernel width may lead to overfitting, while a wider kernel width may result in underfitting. Final answer: C.

\*\*\*\*\*

Answer Option 8:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The choice of kernel width determines how much influence each data point has on the predicted values. If the kernel width is too narrow, the model may overfit the data, whereas if it is too wide, the model may underfit the data. Final answer: C

\*\*\*\*\*

Answer Option 9:

For Kernel Regression, the structural assumption that most affects the trade-off between underfitting and overfitting is the kernel width. The choice of kernel width determines the smoothness of the estimated function and can impact the model's ability to capture complex patterns in the data. The other options mentioned (A, B, and D) are related to the choice of kernel function and metrics, which may influence the behavior of the model, but the kernel width is particularly important for controlling the trade-off between underfitting and overfitting. Final answer: C

\*\*\*\*\*

Question 78:

'Statement 1| The SVM learning algorithm is guaranteed to find the globally optimal hypothesis with respect to its object function. Statement 2| After being mapped into feature space Q through a radial basis kernel function, a Perceptron may be able to achieve better classification performance than in its original space (though we can't guarantee this).', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is False. While the SVM learning algorithm aims to find an optimal hypothesis, it does not guarantee finding the globally optimal hypothesis. Statement 2 is True. The use of a radial basis kernel function in a Perceptron can potentially improve classification performance in a higher-dimensional feature space. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is false. The SVM learning algorithm aims to find the optimal hypothesis that maximizes the margin, but it is not guaranteed to find the globally optimal solution.

Statement 2 is true. By mapping data into a higher-dimensional feature space using a radial basis kernel function, the Perceptron may be able to achieve better classification performance. However, there is no guarantee that this will always be the case.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is not true. The SVM learning algorithm is not guaranteed to find the globally optimal hypothesis with respect to its objective function. It finds the best possible separation, but not necessarily the globally optimal solution.

Statement 2 is true. After being mapped into feature space Q through a radial basis kernel function, a Perceptron may be able to achieve better classification performance than in its original space, although this cannot be guaranteed.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is False. The SVM learning algorithm aims to find the globally optimal hypothesis within the constraints of its objective function, but there is no guarantee that it will always find the globally optimal solution.

Statement 2 is True. The radial basis kernel function can map the input space into a higher-dimensional feature space, where a perceptron may be able to achieve better classification performance compared to its original space. However, this improvement is not guaranteed.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1: The SVM learning algorithm is not guaranteed to find the globally optimal hypothesis with respect to its objective function. It seeks to find the best possible separation between classes, but it may not always find the global optimum.

Statement 2: After being mapped into feature space Q through a radial basis kernel function, a Perceptron may be able to achieve better classification performance than in its original space. However, this is not guaranteed, as the performance improvement depends on the specific data and problem at hand.

Final answer: C (True, False)

\*\*\*\*\*

Answer Option 6:

Statement 1 is False. While the SVM learning algorithm aims to find the globally optimal hypothesis with respect to its objective function, this is not guaranteed due to potential issues such as non-convexity and local optima.

Statement 2 is True. By mapping the data into a higher-dimensional feature space using a radial basis kernel function, a Perceptron may be able to achieve better classification performance compared to its performance in the original space. However, this improvement is not guaranteed and may vary depending on the specific problem.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: The SVM learning algorithm is not guaranteed to find the globally optimal hypothesis with respect to its objective function. It tries to find the best decision boundary based on the given data, but it is possible that there are other optimal solutions that are not found.

Statement 2: After being mapped into feature space Q through a radial basis kernel function, a Perceptron may be able to achieve better classification performance than in its original space. This is because the radial basis kernel function can transform the data into a higher-dimensional space where it becomes linearly separable.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: The SVM learning algorithm is not guaranteed to find the globally optimal hypothesis with respect to its objective function. It aims to find the best possible decision boundary, but it may not always achieve global optimality. Therefore, Statement 1 is False.

Statement 2: After being mapped into feature space  $Q$  through a radial basis kernel function, a Perceptron may achieve better classification performance than in its original space, but this is not guaranteed. The performance improvement depends on the data and the problem at hand. Therefore, Statement 2 is True.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The SVM learning algorithm is guaranteed to find the globally optimal hypothesis with respect to its objective function.

This statement is True. The SVM algorithm aims to find the maximum-margin hyperplane that separates the data points of different classes. It seeks to minimize the classification error while maximizing the margin between the classes, resulting in the globally optimal hypothesis.

Statement 2: After being mapped into feature space  $Q$  through a radial basis kernel function, a Perceptron may be able to achieve better classification performance than in its original space (though we can't guarantee this).

This statement is False. The Perceptron algorithm is a linear classification algorithm that doesn't inherently work with kernel functions or perform non-linear transformations in feature space. It can only separate linearly separable data in its original space and cannot automatically achieve better classification performance by mapping it to a different feature space. Final answer: C

\*\*\*\*\*

Question 79:

'For a Gaussian Bayes classifier, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting:', A. 'Whether we learn the class centers by Maximum Likelihood or Gradient Descent', B. 'Whether we assume full class covariance matrices or diagonal class covariance matrices', C. 'Whether we have equal class priors or priors estimated from the data.', D. 'Whether we allow classes to have different mean vectors or we force them to share the same mean vector'

=====

Answer Option 1:

For a Gaussian Bayes classifier, the assumption that most affects the trade-off between underfitting and overfitting is whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption affects the flexibility of the model and how it can capture the relationship between features within each class. Final answer: B

\*\*\*\*\*

Answer Option 2:

For a Gaussian Bayes classifier, the assumption that most affects the trade-off between underfitting and overfitting is the choice between assuming full class covariance matrices or diagonal class covariance matrices. This assumption affects the flexibility of the model and how it can capture the correlations between different features. Final answer: B

\*\*\*\*\*

Answer Option 3:

For a Gaussian Bayes classifier, the structural assumption that most affects the trade-off between underfitting and overfitting is whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption determines the complexity of the model and how it estimates the covariance between features. Final answer: B

\*\*\*\*\*

Answer Option 4:

The structural assumption that most affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier is whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption determines the complexity of the model and can impact its ability to generalize to unseen data. Final answer: B

\*\*\*\*\*

Answer Option 5:

The structural assumption that most affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier is whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption affects the complexity of the model and the degree to which it can capture the correlations between features. Final answer: B

\*\*\*\*\*

Answer Option 6:

For a Gaussian Bayes classifier, the structural assumption that most affects the trade-off between underfitting and overfitting is whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption determines the level of correlation allowed between the features within each class. Final answer: B

\*\*\*\*\*

Answer Option 7:

For a Gaussian Bayes classifier, the assumption that most affects the trade-off between underfitting and overfitting is the assumption of whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption affects the complexity of the model and how well it can adapt to different patterns in the data. Final answer: B.

\*\*\*\*\*

Answer Option 8:

The most significant factor that affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier is the assumption about the covariance matrices. Whether we assume full class covariance matrices or diagonal class covariance matrices can have a significant impact on the model's ability to capture complex relationships in the data and avoid overfitting. Final answer: B

\*\*\*\*\*

Answer Option 9:

For a Gaussian Bayes classifier, the assumption that most affects the trade-off between underfitting and overfitting is whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption determines the degree of flexibility in capturing the relationships between features, with full covariance matrices allowing for more flexibility but also increasing the risk of overfitting. Final answer: B

\*\*\*\*\*

Question 80:

'Statement 1] Overfitting is more likely when the set of training data is small. Statement 2] Overfitting is more likely when the hypothesis space is small.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. Overfitting is more likely to occur when the set of training data is small because the model may memorize the training examples instead of learning general patterns.

Statement 2 is false. Overfitting is more likely when the hypothesis space is large, not small. A larger hypothesis space allows for more complex models that can fit the training data too closely, leading to overfitting.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true since with a small training data set, there is a higher chance of the model memorizing the data instead of generalizing from it, leading to overfitting. Statement 2 is false because overfitting is more likely to occur when the hypothesis space is large, allowing the model to have more complex and flexible representations. Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. Overfitting is more likely to occur when the set of training data is small because the model has fewer examples to learn from and is more likely to memorize the training data instead of generalizing.

Statement 2 is false. Overfitting is more likely to occur when the hypothesis space is large, not small. A large hypothesis space allows for more complex and flexible models, which can lead to overfitting if not properly regularized.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is true; overfitting is more likely to occur when the set of training data is small because the model has less data to learn from and can easily memorize the training samples. Statement 2 is false; overfitting is more likely when the hypothesis space is large, not small. A larger hypothesis space allows for more complex and flexible models, which can potentially fit the training data too closely and lead to overfitting. Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is True. Overfitting is more likely to occur when the set of training data is small because the model has fewer examples to learn from and may memorize the training data instead of generalizing patterns.

Statement 2 is False. Overfitting is more likely when the hypothesis space is large, not small. A larger hypothesis space allows for more complex models that can fit the training data too closely and overfit.

Final Answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is generally true. Overfitting is more likely to occur when the training data is small because the model has less information to learn from and may try to fit noise or outliers in the data.

Statement 2 is false. Overfitting is more likely when the hypothesis space is large, not small. A large hypothesis space allows for more complex models that can potentially fit the training data too closely, leading to overfitting.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1 is true because when the set of training data is small, there is a higher chance that the model will fit the noise in the data instead of the underlying pattern, leading to overfitting. Statement 2 is false because overfitting is more likely when the hypothesis space is large, not small. A larger hypothesis space allows for more complex models that can fit the training data too closely, resulting in overfitting.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: Overfitting is more likely when the set of training data is small - True. When the training data is small, there is a higher chance that the model will memorize the data instead of learning the underlying patterns, leading to overfitting.

Statement 2: Overfitting is more likely when the hypothesis space is small - False. Overfitting is more likely to occur when the hypothesis space is large, as a larger hypothesis space allows for more complex models that can fit the training data too closely and fail to generalize well to unseen data.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: Overfitting is more likely when the set of training data is small. This statement is true because with a small training dataset, there is a higher chance that the model will memorize the noise or specific patterns in the training data, leading to overfitting.

Statement 2: Overfitting is more likely when the hypothesis space is small. This statement is false because overfitting is more likely when the hypothesis space is large and complex. A small hypothesis space limits the model's ability to fit the training data well, which can actually reduce the risk of overfitting.

Final answer: C

\*\*\*\*\*

Question 81:

'Statement 1| Besides EM, gradient descent can be used to perform inference or learning on Gaussian mixture model. Statement 2 | Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. Gradient descent can be used to perform inference or learning on Gaussian mixture models.

Statement 2 is false. The time complexity of learning a Gaussian-based Bayes optimal classifier is not linear in the number of records in the dataset. Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: Gradient descent can be used to perform inference or learning on Gaussian mixture models. This statement is true.

Statement 2: Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset. This statement is false.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 3:

Statement 1: Besides EM, gradient descent can be used to perform inference or learning on Gaussian mixture model. This statement is true. Gradient descent is an optimization algorithm that can be used for training Gaussian mixture models.

Statement 2: Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset. This statement is false. Learning a Gaussian-based Bayes optimal classifier generally requires estimating the parameters of the model, which is not a linear process in the number of records in the dataset.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1: Gradient descent can be used to perform inference or learning on Gaussian mixture models. This statement is true.

Statement 2: A Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset. This statement is false.

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1: Gradient descent can be used for learning on Gaussian mixture models, so the statement is true.

Statement 2: The time complexity of learning a Gaussian-based Bayes optimal classifier is not linear in the number of records in the dataset. Therefore, the statement is false.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: Besides EM, gradient descent can be used to perform inference or learning on Gaussian mixture model.

This statement is False. While gradient descent can be used for optimization in various machine learning tasks, it is not commonly used for inference or learning on Gaussian mixture models. EM (Expectation-Maximization) is the standard algorithm for this purpose.

Statement 2: Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset.

This statement is True. A Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset, assuming the number of attributes is fixed.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: Besides EM, gradient descent can be used to perform inference or learning on Gaussian mixture model.

This statement is false. Gradient descent is not typically used for performing inference or learning on a Gaussian mixture model. The Expectation-Maximization (EM) algorithm is the commonly used approach for Gaussian mixture models.

Statement 2: Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset.

This statement is true. Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset. The time complexity is proportional to the number of records.

Final Answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: Besides EM, gradient descent can be used to perform inference or learning on Gaussian mixture model. This statement is false. Gradient descent is not commonly used for learning or performing inference on Gaussian mixture models. The Expectation-Maximization (EM) algorithm is the standard method for these tasks.

Statement 2: Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset. This statement is true. Gaussian-based Bayes optimal classifiers, such as Naive Bayes, can be learned in time linear to the number of records in the dataset, assuming a fixed number of attributes.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: Besides EM, gradient descent can be used to perform inference or learning on Gaussian mixture models.

This statement is False. While gradient descent can be used to optimize parameters in certain models, such as neural networks, it is not commonly used for inference or learning on Gaussian mixture models.

EM (Expectation-Maximization) algorithm is the standard method for this task.

Statement 2: Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset.

This statement is True. Given a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset. This is because the classifier can directly compute the necessary statistics from the data without needing to iterate or perform complex computations.

Final answer: C (True, False)

\*\*\*\*\*



Question 82:

'Statement 1] In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination. Statement 2] If two random variable X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is False. The inference results of the junction tree algorithm and variable elimination may differ in a Bayesian network.

Statement 2 is True. If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z.

Final answer: C

=====

Answer Option 2:

Statement 1 is true. The inference results of the junction tree algorithm and variable elimination are the same in a Bayesian network.

Statement 2 is also true. If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z.

Final answer: A

=====

Answer Option 3:

Statement 1 is true. The inference results of the junction tree algorithm and variable elimination algorithm are the same in a Bayesian network.

Statement 2 is also true. If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z.

Final answer: A

=====

Answer Option 4:

Statement 1: In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination.

This statement is FALSE. While both the junction tree algorithm and variable elimination can be used for inference in Bayesian networks, they can produce different results due to their different approaches.

Statement 2: If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z.

This statement is TRUE. In a Bayesian network, if two nodes X and Y are d-separated given a set of nodes Z, then X and Y are conditionally independent given Z.

Final Answer: D

=====

Answer Option 5:

Statement 1: In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination. This statement is False. The junction tree algorithm and variable elimination algorithm can produce different results in certain cases.

Statement 2: If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z. This statement is True. In a Bayesian network, d-separation is used to determine conditional independence between nodes, and when X and Y are conditionally independent given Z, they are d-separated.

Final answer: D

=====

Answer Option 6:

Statement 1: In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination.

This statement is False. While both the junction tree algorithm and variable elimination are methods used for inference in Bayesian networks, they can produce different results.

Statement 2: If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z.

This statement is True. In a Bayesian network, if two nodes (representing random variables) are conditionally independent given another node, they are d-separated given that node.

Final answer: C

=====

Answer Option 7:

Statement 1: In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination. This statement is False. The inference results of the junction tree algorithm and variable elimination can differ in certain cases.

Statement 2: If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z. This statement is True. In a Bayesian network, if two nodes are d-separated given a set of observed nodes, then they are conditionally independent of each other.

Final Answer: D. 'False, True'

=====

Answer Option 8:

Statement 1: In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination. This statement is False. While both the junction tree algorithm and variable elimination are used for inference in Bayesian networks, they can produce different results due to their different computational methods.

Statement 2: If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z. This statement is True. In a Bayesian network, if two nodes are d-separated given a set of observed nodes, it implies that they are conditionally independent given those observed nodes.

Final Answer: D. 'False, True'

\*\*\*\*\*

Answer Option 9:

Statement 1: In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination.

This statement is false. While both the junction tree algorithm and variable elimination are used for inference in Bayesian networks, they can produce different results in certain cases. Final answer: D

Statement 2: If two random variables X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z.

This statement is true. In a Bayesian network, if two nodes (representing random variables) are conditionally independent given another node, it implies that the nodes are d-separated in the network given the evidence of the other node. Final answer: D

\*\*\*\*\*

Question 83:

'Given a large dataset of medical records from patients suffering from heart disease, try to learn whether there might be different clusters of such patients for which we might tailor separate treatments. What kind of learning problem is this?', A. 'Supervised learning', B. 'Unsupervised learning', C. 'Both (a) and (b)', D. 'Neither (a) nor (b)'

=====

Answer Option 1:

The given scenario describes an unsupervised learning problem. The goal is to identify clusters or patterns within the dataset without using any labeled data. Supervised learning involves using labeled data to train a model for making predictions. Final answer: B

\*\*\*\*\*

Answer Option 2:

The learning problem described, where the goal is to identify clusters of patients based on their medical records, is an example of unsupervised learning. In unsupervised learning, there are no predefined labels or outcomes to predict. Final answer: B

\*\*\*\*\*

Answer Option 3:

The learning problem described, where the goal is to identify different clusters within a dataset, is an example of unsupervised learning. In unsupervised learning, the model is tasked with finding patterns, structures, or relationships in the data without the presence of labeled examples. Final answer: B

\*\*\*\*\*

Answer Option 4:

The learning problem described, where we are trying to identify different clusters of patients with heart disease, is an example of unsupervised learning. In unsupervised learning, we aim to find patterns, relationships, or subgroups in the data without having predefined labels or outcomes. Final answer: B

\*\*\*\*\*

Answer Option 5:

The goal of trying to identify different clusters of patients suffering from heart disease and tailoring separate treatments is an example of unsupervised learning. In unsupervised learning, the algorithm is not provided with labeled data, but instead, it finds patterns and structures in the data on its own. Final answer: B

\*\*\*\*\*

Answer Option 6:

The learning problem described, where the goal is to identify different clusters of patients based on their medical records, is an example of unsupervised learning. Unsupervised learning is used when the data does not have pre-defined labels or classes, and the goal is to discover patterns or structures in the data. Final answer: B

\*\*\*\*\*

Answer Option 7:

Based on the given information, the task of identifying different clusters of patients suffering from heart disease does not involve labeled data or predicting a specific output. Instead, it focuses on finding patterns and relationships within the data. This aligns with the definition of unsupervised learning. Therefore, the correct answer is B. 'Unsupervised learning'.

\*\*\*\*\*

Answer Option 8:

The given learning problem, which involves finding different clusters of patients suffering from heart disease, falls under unsupervised learning. In unsupervised learning, the algorithm aims to find patterns or groupings in the data without the need for labeled examples. Final answer: B

\*\*\*\*\*

Answer Option 9:

The learning problem described, where the goal is to find different clusters of patients based on their medical records, is an example of unsupervised learning. In unsupervised learning, the algorithm is not provided with labeled data and instead seeks to find patterns or structures in the data. Final answer: B

\*\*\*\*\*

Question 84:

'What would you do in PCA to get the same projection as SVD?', A. 'Transform data to zero mean', B. 'Transform data to zero median', C. 'Not possible', D. 'None of these'

=====

Answer Option 1:

To get the same projection as Singular Value Decomposition (SVD), you would need to transform the data to zero mean in Principal Component Analysis (PCA). Final answer: A

\*\*\*\*\*

Answer Option 2:

To obtain the same projection as Singular Value Decomposition (SVD) in Principal Component Analysis (PCA), the data should be transformed to zero mean. This is because centering the data around zero mean is a necessary step in both PCA and SVD. Final answer: A

\*\*\*\*\*

Answer Option 3:

To get the same projection as Singular Value Decomposition (SVD) in Principal Component Analysis (PCA), you would need to transform the data to zero mean. This is because SVD and PCA both involve centering the data by subtracting the mean before performing the decomposition. Final answer: A

\*\*\*\*\*

Answer Option 4:

To get the same projection as Singular Value Decomposition (SVD) in Principal Component Analysis (PCA), the data should be transformed to have zero mean. This ensures that the principal components capture the maximum variance in the data. Final answer: A

\*\*\*\*\*

Answer Option 5:

To achieve the same projection as Singular Value Decomposition (SVD) in Principal Component Analysis (PCA), the data needs to be transformed to zero mean. This is achieved by subtracting the mean of each feature from the data. Final answer: A

\*\*\*\*\*

Answer Option 6:

To get the same projection as SVD (Singular Value Decomposition) in PCA (Principal Component Analysis), the data needs to be transformed to zero mean. This is the process of subtracting the mean of each feature from the data. Final answer: A

\*\*\*\*\*

Answer Option 7:

To get the same projection as singular value decomposition (SVD) in principal component analysis (PCA), you would need to transform the data to zero mean. This ensures that the principal components capture the directions of maximum variance in the data. Final answer: A

\*\*\*\*\*

Answer Option 8:

To get the same projection as Singular Value Decomposition (SVD) in Principal Component Analysis (PCA), the data needs to be transformed to zero mean. By subtracting the mean from the data, we center it around zero before performing PCA. Final answer: A

\*\*\*\*\*

Answer Option 9:

In PCA (Principal Component Analysis), to get the same projection as SVD (Singular Value Decomposition), we need to transform the data to zero mean. This ensures that the data is centered around the origin, which is a requirement for both PCA and SVD. Final answer: A

\*\*\*\*\*

Question 85:

'Statement 1] The training error of 1-nearest neighbor classifier is 0. Statement 2] As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors. In other words, given enough data, the choice of prior is irrelevant.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: The training error of a 1-nearest neighbor classifier is 0. This statement is false. The training error of a 1-nearest neighbor classifier is not necessarily 0, as it may incorrectly classify certain training instances.

Statement 2: As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors. In other words, given enough data, the choice of prior is irrelevant. This statement is true. As the number of data points increases, the influence of the prior decreases, making the choice of prior irrelevant.

Final answer: C

=====

Answer Option 2:

Statement 1: The training error of a 1-nearest neighbor classifier is 0. This statement is false. A 1-nearest neighbor classifier does not necessarily have a training error of 0, as it can misclassify points that are close to decision boundaries.

Statement 2: As the number of data points grows to infinity, the maximum a posteriori (MAP) estimate approaches the maximum likelihood estimate (MLE) estimate for all possible priors. This statement is true. As the number of data points increases, the influence of the prior decreases and the MAP estimate converges to the MLE estimate.

Therefore, the correct answer is C. 'True, False'.

=====

Answer Option 3:

Statement 1: The training error of a 1-nearest neighbor classifier is not necessarily 0. It can be 0 if the training data is perfectly separable, but in general, it will have some training error.

Statement 2: As the number of data points grows to infinity, the Maximum a Posteriori (MAP) estimate does not necessarily approach the Maximum Likelihood Estimation (MLE) estimate for all possible priors. The choice of prior can still have an influence on the estimate, even with a large amount of data.

Final answer: C

=====

Answer Option 4:

Statement 1: The training error of a 1-nearest neighbor classifier is 0.

This statement is false because a 1-nearest neighbor classifier can have a training error greater than 0 if there are misclassified instances.

Statement 2: As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors.

This statement is true. As the number of data points increases, the impact of the prior becomes less significant and the MAP estimate converges to the MLE estimate.

Final answer: C

=====

Answer Option 5:

Statement 1: The training error of a 1-nearest neighbor classifier is 0. This statement is False because a 1-nearest neighbor classifier will have a training error of 0 only if the training data points are perfectly separable.

Statement 2: As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors. In other words, given enough data, the choice of prior is irrelevant. This statement is True. As the number of data points increases, the influence of the prior decreases and the MLE estimate becomes dominant.

Final Answer: C

=====

Answer Option 6:

Statement 1: The training error of a 1-nearest neighbor classifier is 0. This statement is False. A 1-nearest neighbor classifier will always have a training error of 0, meaning it will classify the training examples perfectly.

Statement 2: As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors. This statement is True. As the number of data points increases, the influence of the prior decreases, and the Maximum Likelihood Estimate (MLE) becomes more dominant. The choice of prior becomes less relevant when there is a large amount of data.

Final answer: C

=====

Answer Option 7:

Statement 1: The training error of a 1-nearest neighbor classifier is not necessarily 0, as it can misclassify some training points if there are overlapping classes or noisy data. Therefore, statement 1 is False.

Statement 2: As the number of data points grows to infinity, the Maximum a Posteriori (MAP) estimate does approach the Maximum Likelihood Estimate (MLE) estimate for all possible priors. This is because with an infinite amount of data, the prior becomes less influential and the likelihood dominates, making the choice of prior irrelevant. Therefore, statement 2 is True.

Final answer: C

=====

Answer Option 8:

Statement 1| The training error of 1-nearest neighbor classifier is 0. This statement is False. The training error of 1-nearest neighbor classifier will not necessarily be zero as it can perfectly classify the training data.

Statement 2| As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors. In other words, given enough data, the choice of prior is irrelevant. This statement is True. As the number of data points increases, the influence of the prior decreases, and the Maximum Likelihood Estimate (MLE) and Maximum A Posteriori (MAP) estimate converge.

Final Answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: The training error of a 1-nearest neighbor classifier is 0. This statement is False. The training error of a 1-nearest neighbor classifier is not guaranteed to be 0 because it assigns the label of the nearest neighbor in the training set to each test point. There can be cases where the nearest neighbor is mislabeled.

Statement 2: As the number of data points grows to infinity, the MAP (Maximum A Posteriori) estimate approaches the MLE (Maximum Likelihood Estimation) estimate for all possible priors. In other words, given enough data, the choice of prior is irrelevant. This statement is True. With an infinite amount of data, the influence of the prior becomes negligible, and the MAP estimate converges to the MLE estimate.

Final Answer: C

\*\*\*\*\*

Question 86:

'When doing least-squares regression with regularisation (assuming that the optimisation can be done exactly), increasing the value of the regularisation parameter  $\lambda$  the testing error.', A. 'will never decrease the training error.', B. 'will never increase the training error.', C. 'will never decrease the testing error.', D. 'will never increase'

=====

Answer Option 1:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  will never decrease the testing error. Final answer: C

\*\*\*\*\*

Answer Option 2:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  will never increase the training error. However, it may cause an increase in the testing error.

Final answer: B

\*\*\*\*\*

Answer Option 3:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  typically results in an increase in both the training and testing error. Therefore, the correct answer is B: 'will never increase the training error.'

\*\*\*\*\*

Answer Option 4:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  will never decrease the testing error. However, it is possible that increasing  $\lambda$  may increase the training error. Final answer: C

\*\*\*\*\*

Answer Option 5:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  will never decrease the testing error. However, it is not true that it will never decrease the training error (option A). The regularization parameter  $\lambda$  can impact the complexity of the model and help prevent overfitting. Final answer: C

\*\*\*\*\*

Answer Option 6:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  will never decrease the testing error. However, it may or may not decrease the training error, depending on the specific case and data. Therefore, the correct answer is C. 'Increasing the value of the regularisation parameter  $\lambda$  will never decrease the testing error.'

\*\*\*\*\*

Answer Option 7:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  generally increases the testing error. However, it is not true that it will never decrease the training error or that it will never increase the training error. Similarly, it is not true that it will never decrease the testing error or that it will never increase. Final answer: None of the above (no appropriate option given).

\*\*\*\*\*

Answer Option 8:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  will typically increase the testing error. This is because larger values of  $\lambda$  penalize the model more for complex or high-variance solutions, which can lead to poorer generalization performance on unseen data. However, the training error may or may not increase depending on the specific dataset and the amount of regularization being applied. Final answer: C.

\*\*\*\*\*

Answer Option 9:

When doing least-squares regression with regularization, increasing the value of the regularization parameter  $\lambda$  typically leads to an increase in the testing error. This is because higher regularization places stronger constraints on the model, which can prevent it from fitting the training data well and result in higher errors on the testing data. However, it is important to note that the impact of regularization on the training and testing errors can depend on the specific dataset and model. Final answer: C.

\*\*\*\*\*

Question 87:

'Which of the following best describes what discriminative approaches try to model? (w are the parameters in the model)', A. ' $p(y|x, w)$ ', B. ' $p(y, x)$ ', C. ' $p(w|x, w)$ ', D. 'None of the above'

=====

Answer Option 1:

Discriminative approaches in machine learning try to model the conditional probability of the output given the input, denoted as  $p(y|x, w)$ . Final answer: A

\*\*\*\*\*

Answer Option 2:

Discriminative approaches in machine learning try to model the conditional probability of the target variable given the input data, represented as  $p(y|x, w)$ . Final answer: A

\*\*\*\*\*

Answer Option 3:

Discriminative approaches in machine learning try to model the conditional probability of the output given the input, which can be represented as  $p(y|x, w)$ . Therefore, the correct answer is A. Final answer: A

\*\*\*\*\*

Answer Option 4:

Discriminative approaches in machine learning try to model the conditional probability of the output variable given the input variable and the model parameters. Therefore, the correct description is A. ' $p(y|x, w)$ '. Final answer: A

\*\*\*\*\*

Answer Option 5:

Discriminative approaches in machine learning try to model the conditional probability distribution  $p(y|x, w)$ , where y is the output or target variable, x is the input or features, and w are the parameters in the model. Final answer: A

\*\*\*\*\*

Answer Option 6:

Discriminative approaches in machine learning aim to model the conditional probability distribution of the target variable given the input features, which is represented as  $p(y|x, w)$ . Final answer: A

\*\*\*\*\*

Answer Option 7:

Discriminative approaches in machine learning try to model the conditional probability  $p(y|x, w)$ , which represents the probability of the target variable y given the input data x and the model parameters w.

Final answer: A

\*\*\*\*\*

Answer Option 8:

Discriminative approaches in machine learning try to model the conditional probability of the output variable given the input and the model parameters, which is denoted as  $p(y|x, w)$ . Therefore, the correct answer is A. Final answer: A

\*\*\*\*\*

Answer Option 9:

Discriminative approaches in machine learning try to model the conditional probability  $p(y|x, w)$ , where y is the output variable and x is the input variable. This is in contrast to generative approaches that model the joint probability  $p(y, x)$ . Final answer: A

\*\*\*\*\*



Question 88:

'Statement 1| CIFAR-10 classification performance for convolution neural networks can exceed 95%. Statement 2| Ensembles of neural networks do not improve classification accuracy since the representations they learn are highly correlated.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: CIFAR-10 classification performance for convolutional neural networks can exceed 95%. True.

Statement 2: Ensembles of neural networks do not improve classification accuracy since the representations they learn are highly correlated. False.

Final Answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. Convolutional neural networks (CNNs) have been shown to achieve classification performance exceeding 95% on the CIFAR-10 dataset.

Statement 2 is false. Ensembles of neural networks can improve classification accuracy by combining the predictions of multiple models, especially when the models are diverse and learn different representations. Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is generally true. It is possible to achieve CIFAR-10 classification performance of over 95% using convolutional neural networks.

Statement 2 is false. Ensembles of neural networks can improve classification accuracy by leveraging the diversity of models and reducing the impact of overfitting. The representations learned by the ensemble models are not necessarily highly correlated.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is false. While convolutional neural networks (CNNs) can achieve high performance on the CIFAR-10 dataset, it is not accurate to claim that their performance can exceed 95%.

Statement 2 is also false. Ensembles of neural networks are known to improve classification accuracy by leveraging diversity among the individual models. Therefore, it is incorrect to say that ensembles do not improve classification accuracy due to highly correlated representations.

Final answer: B - 'False, False'

\*\*\*\*\*

Answer Option 5:

Statement 1 is false. While CIFAR-10 classification performance for convolutional neural networks has improved significantly in recent years, exceeding 95% accuracy is still a challenging task.

Statement 2 is also false. Ensembles of neural networks can improve classification accuracy by combining the predictions of multiple models, reducing errors and increasing robustness. The representations learned by different networks within an ensemble can be diverse, leading to improved performance.

Final answer: B

\*\*\*\*\*

Answer Option 6:

Statement 1: CIFAR-10 classification performance for convolutional neural networks can exceed 95%. This statement is plausible, as convolutional neural networks have shown high performance on the CIFAR-10 dataset. However, without specific evidence, we cannot definitively say it is true or false.

Statement 2: Ensembles of neural networks do not improve classification accuracy since the representations they learn are highly correlated. This statement is false. Ensembles of neural networks, such as bagging or boosting, are known to improve classification accuracy by combining multiple models and reducing overfitting.

Final Answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: CIFAR-10 classification performance for convolutional neural networks can exceed 95%.

This statement is plausible, as state-of-the-art convolutional neural networks have achieved high accuracy on the CIFAR-10 dataset.

Statement 2: Ensembles of neural networks do not improve classification accuracy since the representations they learn are highly correlated.

This statement is false. Ensembles of neural networks can improve classification accuracy by combining the predictions of multiple models, reducing the risk of overfitting and improving generalization.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1| CIFAR-10 classification performance for convolution neural networks can exceed 95%: This statement is possible as convolutional neural networks (CNNs) have been shown to achieve high accuracy on the CIFAR-10 dataset. Therefore, the statement can be true.

Statement 2| Ensembles of neural networks do not improve classification accuracy since the representations they learn are highly correlated: This statement is false. Ensembles of neural networks can improve classification accuracy by combining multiple models with different initializations or architectures, reducing the impact of overfitting and increasing the model's ability to generalize.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: CIFAR-10 classification performance for convolutional neural networks can exceed 95%.

This statement is possible, as convolutional neural networks have been shown to achieve high performance on the CIFAR-10 dataset. Therefore, the statement is likely true.

Statement 2: Ensembles of neural networks do not improve classification accuracy since the representations they learn are highly correlated.

This statement is false. Ensembles of neural networks can improve classification accuracy by combining the predictions of multiple models. The idea behind ensembles is that each model may make different errors due to its unique biases and variance. By averaging or combining the predictions, ensembles can reduce bias and variance and improve overall accuracy. Therefore, the statement is false.

Final answer: C

\*\*\*\*\*

Question 89:

'Which of the following points would Bayesians and frequentists disagree on?', A. 'The use of a non-Gaussian noise model in probabilistic regression.', B. 'The use of probabilistic modelling for regression.', C. 'The use of prior distributions on the parameters in a probabilistic model.', D. 'The use of class priors in Gaussian Discriminant Analysis.'

=====

Answer Option 1:

Bayesians and frequentists would disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists do not typically incorporate prior beliefs into their analysis, while Bayesians believe in using prior distributions to capture prior knowledge or beliefs about the parameters. Final answer: C

\*\*\*\*\*

Answer Option 2:

Bayesians and frequentists would disagree on the use of prior distributions on the parameters in a probabilistic model (option C). Frequentists do not typically use prior distributions in their analysis, while Bayesians incorporate prior knowledge about the parameters into their modeling. The other options may be valid approaches in both Bayesian and frequentist frameworks. Final answer: C

\*\*\*\*\*

Answer Option 3:

Bayesians and frequentists would most likely disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists typically do not incorporate prior beliefs or subjective information into their analysis, while Bayesians utilize prior distributions to incorporate prior beliefs and update them based on the data. The other options, such as the use of non-Gaussian noise models, probabilistic modeling for regression, and class priors in Gaussian Discriminant Analysis, do not necessarily represent fundamental disagreements between Bayesians and frequentists. Final answer: C

\*\*\*\*\*

Answer Option 4:

Bayesians and frequentists would most likely disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists typically do not incorporate prior beliefs into their analysis, while Bayesians use prior distributions to represent prior knowledge or beliefs about the parameters. Final answer: C

\*\*\*\*\*

Answer Option 5:

Bayesians and frequentists would disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists typically do not incorporate prior beliefs or subjective probabilities into their analysis, while Bayesians use prior distributions to express beliefs about the parameters before seeing the data. Final answer: C

\*\*\*\*\*

Answer Option 6:

Bayesians and frequentists would likely disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists typically do not use prior distributions in their analysis, while Bayesians incorporate prior beliefs into their models through the use of prior distributions. Final answer: C

\*\*\*\*\*

Answer Option 7:

Bayesians and frequentists would disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists do not typically incorporate prior beliefs into their analysis, while Bayesians use prior distributions to represent their beliefs about the parameters before observing the data. Final answer: C

\*\*\*\*\*

Answer Option 8:

Bayesians and frequentists would typically disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists generally do not incorporate prior information into their models and instead rely on the likelihood function, while Bayesians use prior distributions to represent their beliefs about the parameters before observing the data. Final answer: C

\*\*\*\*\*

Answer Option 9:

Bayesians and frequentists would likely disagree on the use of prior distributions on the parameters in a probabilistic model. Frequentists typically do not incorporate prior beliefs into their analysis and rely solely on observed data, while Bayesians use prior distributions to express their initial beliefs about the parameters before updating them with the data. Final answer: C

\*\*\*\*\*

Question 90:

'Statement 1] The BLEU metric uses precision, while the ROGUE metric uses recall. Statement 2] Hidden markov models were frequently used to model English sentences.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. The BLEU metric is used to evaluate the quality of machine translated text and it does not rely on precision. Statement 2 is also false. Hidden Markov Models (HMMs) are not typically used to model English sentences. Final answer: B

\*\*\*\*\*

Answer Option 2:

Statement 1 is False. The BLEU metric is used to evaluate the quality of machine-generated translations and does not specifically use precision or recall. Statement 2 is False. Hidden Markov Models (HMMs) are commonly used for part-of-speech tagging and speech recognition tasks, but not specifically for modeling English sentences. Final answer: B

\*\*\*\*\*

Answer Option 3:

Statement 1 is false. The BLEU (Bilingual Evaluation Understudy) metric does not use precision, but rather measures the similarity between machine-generated translations and human-generated reference translations.

Statement 2 is false. Hidden Markov Models (HMMs) are not frequently used to model English sentences. They are more commonly used in speech recognition and bioinformatics.

Final answer: B

\*\*\*\*\*

Answer Option 4:

Statement 1 is false. The BLEU metric, commonly used for evaluating machine translation, computes precision as well as recall.

Statement 2 is false. Hidden Markov Models (HMMs) are probabilistic models commonly used for sequence labeling tasks, such as part-of-speech tagging or speech recognition. They are not typically used to model English sentences. Final answer: D

\*\*\*\*\*

Answer Option 5:

Statement 1: The BLEU metric does not use precision, but rather a modified form of precision that takes into account n-gram matches between the reference and predicted sentences. This statement is false.

Statement 2: Hidden Markov Models (HMMs) have been used in natural language processing for tasks like speech recognition and part-of-speech tagging, but they are not commonly used to model English sentences. This statement is also false.

Final answer: B

\*\*\*\*\*

Answer Option 6:

Statement 1: The BLEU metric measures the similarity between machine-generated translations and human-generated references. It does not specifically use precision.

Statement 2: Hidden Markov models are used for tasks such as speech recognition and part-of-speech tagging, but they are not commonly used to model English sentences.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: The BLEU metric measures the similarity between machine-generated translations and human-generated reference translations, and it does not directly use precision or recall. Therefore, statement 1 is false.

Statement 2: Hidden Markov Models (HMMs) are commonly used to model sequential data, such as speech recognition or part-of-speech tagging, but they are not specifically designed for modeling English sentences. Therefore, statement 2 is also false.

Final answer: B (False, False)

\*\*\*\*\*

Answer Option 8:

Statement 1 is false. The BLEU metric, which stands for Bilingual Evaluation Understudy, measures the similarity between a machine-generated translation and one or more reference translations. It uses precision, but also incorporates other components such as n-grams. Statement 2 is also false. Hidden Markov Models (HMMs) are probabilistic models used for sequence labeling tasks, such as speech recognition or part-of-speech tagging. They are not specifically designed to model English sentences. Final answer: B

\*\*\*\*\*

Answer Option 9:

Statement 1: The BLEU metric uses precision, while the ROGUE metric uses recall.

This statement is false. The BLEU metric measures the similarity between machine-generated text and reference text using precision, while the ROGUE metric is used to evaluate the quality of summaries and also uses precision and recall.

Statement 2: Hidden markov models were frequently used to model English sentences.

This statement is false. Hidden Markov Models (HMMs) are used for sequence modeling tasks, such as speech recognition and part-of-speech tagging, but they are not frequently used to model English sentences.

Final answer: D  
\*\*\*\*\*

Question 91:

'Statement 1| ImageNet has images of various resolutions. Statement 2| Caltech-101 has more images than ImageNet.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is true. ImageNet is a dataset that contains images of various resolutions.

Statement 2 is false. ImageNet has significantly more images than Caltech-101.

Final answer: C

=====

Answer Option 2:

Statement 1 is true, as ImageNet does have images of various resolutions. Statement 2 is false, as ImageNet actually has significantly more images than Caltech-101. Final answer: C

=====

Answer Option 3:

Statement 1 is true. ImageNet contains images of various resolutions from different sources.

Statement 2 is false. ImageNet contains significantly more images compared to Caltech-101, which is a smaller dataset. Final answer: C

=====

Answer Option 4:

Statement 1 is true. ImageNet is a large-scale dataset that contains images of various resolutions.

Statement 2 is false. ImageNet has significantly more images than Caltech-101. Final answer: C

=====

Answer Option 5:

Statement 1 is true. ImageNet does have images of various resolutions.

Statement 2 is false. ImageNet has significantly more images than Caltech-101.

Therefore, the correct answer is C. 'True, False'.

=====

Answer Option 6:

Statement 1: ImageNet does have images of various resolutions, as it is a large dataset that covers a wide range of object categories.

Statement 2: Caltech-101 does not have more images than ImageNet. ImageNet is known for its vast collection of images, consisting of millions of labeled images across thousands of categories.

Final answer: C

=====

Answer Option 7:

Statement 1 is true. ImageNet does have images of various resolutions.

Statement 2 is false. ImageNet has significantly more images than Caltech-101.

Final answer: C

=====

Answer Option 8:

Statement 1 is true. ImageNet does have images of various resolutions. Statement 2 is false. Caltech-101 does not have more images than ImageNet. Final answer: C

=====

Answer Option 9:

Statement 1 is true. ImageNet does have images of various resolutions.

Statement 2 is false. ImageNet has significantly more images than Caltech-101.

Final answer: C

=====

Question 92:

'Which of the following is more appropriate to do feature selection?', A. 'Ridge', B. 'Lasso', C. 'both (a) and (b)', D. 'neither (a) nor (b)'

=====

Answer Option 1:

Both Ridge and Lasso regularization techniques can be used for feature selection. However, Lasso regularization is particularly known for its ability to perform feature selection by shrinking the coefficients of less important features to zero. Therefore, the more appropriate option for feature selection would be B. 'Lasso'. Final answer: B

\*\*\*\*\*

Answer Option 2:

Both Ridge and Lasso are popular regularization techniques used in feature selection. While Ridge regression helps reduce the impact of irrelevant features, Lasso regression can set the coefficients of irrelevant features to zero, effectively eliminating them from the model. Therefore, the more appropriate answer would be C, 'both (a) and (b)'.

\*\*\*\*\*

Answer Option 3:

Both Ridge and Lasso are regularization techniques that can be used for feature selection. However, Lasso is more appropriate for feature selection as it has the ability to shrink coefficients to exactly zero, effectively eliminating features from the model. Ridge, on the other hand, only shrinks the coefficients towards zero but does not eliminate them completely. Final answer: B

\*\*\*\*\*

Answer Option 4:

Both Ridge and Lasso are regularization techniques used for feature selection in machine learning. Ridge regression uses L2 regularization to shrink the coefficient values, while Lasso regression uses L1 regularization to shrink some coefficients to zero, effectively performing feature selection. Therefore, the correct answer is C, 'both (a) and (b)'.

\*\*\*\*\*

Answer Option 5:

Both Ridge and Lasso are regularization techniques that can be used for feature selection. However, Lasso is more commonly used for feature selection as it has the ability to shrink the coefficients of less important features to zero, effectively excluding them from the model. Ridge, on the other hand, does not shrink coefficients to zero but rather reduces their magnitude. Therefore, the more appropriate option for feature selection is B. 'Lasso'. Final answer: B

\*\*\*\*\*

Answer Option 6:

Both Ridge and Lasso are regularization techniques used in machine learning, but they have different effects on feature selection. Lasso is more appropriate for feature selection as it tends to shrink the coefficients of irrelevant features to zero. Ridge, on the other hand, does not perform explicit feature selection but rather shrinks the coefficients towards zero without eliminating any features entirely. Final answer: B

\*\*\*\*\*

Answer Option 7:

Both Ridge and Lasso are regularization techniques that can be used for feature selection. However, Lasso is generally more appropriate for feature selection as it has the ability to set some feature coefficients to exactly zero, effectively removing them from the model. Ridge, on the other hand, shrinks the coefficients towards zero but does not set them exactly to zero. Final answer: B

\*\*\*\*\*

Answer Option 8:

Both Ridge and Lasso are regularization techniques commonly used for feature selection in machine learning. However, Lasso is more appropriate for feature selection as it tends to shrink the coefficients of irrelevant features to zero, effectively excluding them from the model. Final answer: B

\*\*\*\*\*

Answer Option 9:

Both Ridge and Lasso are regularization techniques used in machine learning, but they have different effects on feature selection. Lasso regularization tends to drive the coefficients of less important features to zero, effectively performing feature selection. Ridge regularization, on the other hand, does not directly perform feature selection but rather reduces the impact of less important features. Therefore, Lasso (B) is more appropriate for feature selection. Final answer: B

\*\*\*\*\*

Question 93:

'Suppose you are given an EM algorithm that finds maximum likelihood estimates for a model with latent variables. You are asked to modify the algorithm so that it finds MAP estimates instead. Which step or steps do you need to modify?', A. 'Expectation', B. 'Maximization', C. 'No modification necessary', D. 'Both'

=====

Answer Option 1:

To modify the EM algorithm to find the maximum a posteriori (MAP) estimates instead of maximum likelihood estimates for a model with latent variables, you need to modify the Maximization step. The Expectation step remains the same. Final answer: B

\*\*\*\*\*

Answer Option 2:

To modify the EM algorithm to find maximum a posteriori (MAP) estimates instead of maximum likelihood estimates, you would need to modify the "Maximization" step. The "Expectation" step remains the same. In the Maximization step, you would incorporate the prior distribution of the latent variables into the estimation process. Final answer: B

\*\*\*\*\*

Answer Option 3:

To modify the EM algorithm to find Maximum A Posteriori (MAP) estimates instead of maximum likelihood estimates, both the Expectation and Maximization steps need to be modified. In the Expectation step, the algorithm needs to incorporate the prior information by multiplying the likelihood with the prior probability of the latent variables. In the Maximization step, the algorithm needs to maximize the posterior probability instead of the likelihood. Final answer: D

\*\*\*\*\*

Answer Option 4:

To modify an EM algorithm to find Maximum A Posteriori (MAP) estimates instead of Maximum Likelihood (ML) estimates, the modification needs to be made in the "Maximization" step. In the Maximization step, the ML algorithm maximizes the likelihood function, while the MAP algorithm maximizes the posterior probability, which involves incorporating prior information. Final answer: B

\*\*\*\*\*

Answer Option 5:

To modify the EM algorithm to find Maximum A Posteriori (MAP) estimates instead of Maximum Likelihood (ML) estimates, you would need to modify the Maximization step. The Expectation step remains the same. In the Maximization step, you update the parameters of the model based on the expected values of the latent variables. To find MAP estimates, you would need to incorporate prior probabilities into the Maximization step. Final answer: B

\*\*\*\*\*

Answer Option 6:

To modify the EM algorithm to find maximum a posteriori (MAP) estimates instead of maximum likelihood estimates for a model with latent variables, you need to modify the 'Maximization' step. In the 'Maximization' step, you need to incorporate the prior distribution of the parameters in order to find the MAP estimates. The 'Expectation' step remains the same. Final answer: B

\*\*\*\*\*

Answer Option 7:

To modify the EM algorithm to find Maximum A Posteriori (MAP) estimates instead of maximum likelihood estimates, you would need to modify the Maximization step. The Maximization step in the EM algorithm involves updating the parameters of the model based on the expected values of the latent variables. To find MAP estimates, you would need to incorporate prior information into the parameter updates during the Maximization step. Therefore, the correct answer is B. 'Maximization'.

\*\*\*\*\*

Answer Option 8:

To modify the EM algorithm to find Maximum A Posteriori (MAP) estimates instead of maximum likelihood estimates, both the expectation and maximization steps need to be modified. In the expectation step, the algorithm needs to incorporate prior information to update the posterior probabilities of the latent variables. In the maximization step, the algorithm needs to include the prior distribution of the parameters to update the estimates. Therefore, the correct answer is D - both steps need to be modified.

\*\*\*\*\*

Answer Option 9:

To modify the EM algorithm so that it finds Maximum A Posteriori (MAP) estimates instead of Maximum Likelihood (ML) estimates, we need to modify the Maximization step. In the Maximization step, we update the parameters of the model based on the expected values of the latent variables. To obtain MAP estimates, we need to modify the Maximization step to incorporate the prior probabilities of the parameters. Final answer: B

\*\*\*\*\*



Question 94:

'For a Gaussian Bayes classifier, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting:', A. 'Whether we learn the class centers by Maximum Likelihood or Gradient Descent', B. 'Whether we assume full class covariance matrices or diagonal class covariance matrices', C. 'Whether we have equal class priors or priors estimated from the data', D. 'Whether we allow classes to have different mean vectors or we force them to share the same mean vector'

=====

Answer Option 1:

The assumption that most affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier is whether we assume full class covariance matrices or diagonal class covariance matrices.

This assumption affects the flexibility of the model and its ability to capture the dependencies between features. Final answer: B

\*\*\*\*\*

Answer Option 2:

For a Gaussian Bayes classifier, the assumption that most affects the trade-off between underfitting and overfitting is whether we assume full class covariance matrices or diagonal class covariance matrices.

This assumption affects the complexity of the model and its ability to capture the relationships between features. Final answer: B

\*\*\*\*\*

Answer Option 3:

The assumption that most affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier is whether we assume full class covariance matrices or diagonal class covariance matrices.

This assumption determines the complexity of the decision boundary and can impact the classifier's ability to capture complex relationships between features. Final answer: B

\*\*\*\*\*

Answer Option 4:

The structural assumption that most affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier is the assumption of whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption affects the complexity of the model and its ability to capture the relationships between different features. Final answer: B

\*\*\*\*\*

Answer Option 5:

For a Gaussian Bayes classifier, the assumption that most affects the trade-off between underfitting and overfitting is the assumption of whether we assume full class covariance matrices or diagonal class covariance matrices. This assumption determines the flexibility of the model and its ability to capture the complex relationships between variables. Final answer: B

\*\*\*\*\*

Answer Option 6:

For a Gaussian Bayes classifier, the assumption that most affects the trade-off between underfitting and overfitting is the choice between assuming full class covariance matrices or diagonal class covariance matrices (option B). This assumption affects the flexibility of the model in capturing the relationships between features and can impact the model's ability to generalize well to unseen data. Final answer: B

\*\*\*\*\*

Answer Option 7:

The assumption that most affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier is whether we assume full class covariance matrices or diagonal class covariance matrices.

This assumption determines the level of flexibility in modeling the covariance structure of each class, which can impact the model's ability to capture complex relationships between features. Final answer: B

\*\*\*\*\*

Answer Option 8:

For a Gaussian Bayes classifier, the choice of whether we assume full class covariance matrices or diagonal class covariance matrices is the structural assumption that most affects the trade-off between underfitting and overfitting. This choice determines the level of flexibility in modeling the covariance structure of each class, which can impact the model's ability to capture the true underlying distribution and avoid overfitting. Final answer: B

\*\*\*\*\*

Answer Option 9:

The choice of whether to assume full class covariance matrices or diagonal class covariance matrices is the structural assumption that most affects the trade-off between underfitting and overfitting in a Gaussian Bayes classifier. This choice determines the level of flexibility in modeling the covariance structure of the data and can impact the classifier's ability to capture complex relationships between features. Final answer: B

\*\*\*\*\*

Question 95:

'Statement 1] For any two variables  $x$  and  $y$  having joint distribution  $p(x, y)$ , we always have  $H[x, y] \geq H[x] + H[y]$  where  $H$  is entropy function. Statement 2] For some directed graphs, moralization decreases the number of edges present in the graph.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: The statement is true. The joint entropy of two variables  $x$  and  $y$ ,  $H[x, y]$ , is always greater than or equal to the sum of their individual entropies,  $H[x] + H[y]$ , due to the property of entropy.

Statement 2: The statement is false. Moralization in a directed graph involves adding edges between parents of the same child node, which can actually increase the number of edges in the graph, not decrease it.

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1: For any two variables  $x$  and  $y$  having joint distribution  $p(x, y)$ , we always have  $H[x, y] \geq H[x] + H[y]$  where  $H$  is the entropy function.

This statement is true and is known as the data processing inequality. The joint entropy of  $x$  and  $y$ ,  $H[x, y]$ , represents the amount of uncertainty in the joint distribution, while  $H[x] + H[y]$  represents the uncertainty in each variable separately. The inequality states that the uncertainty in the joint distribution cannot be less than the sum of the uncertainties in the individual distributions.

Statement 2: For some directed graphs, moralization decreases the number of edges present in the graph.

This statement is false. Moralization is a process in graphical models where the directed edges in the graph are replaced with undirected edges between nodes that share a common child. This can introduce additional edges in the graph, but it does not decrease the number of edges.

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. According to the subadditivity property of entropy, the joint entropy of two variables is always greater than or equal to the sum of their individual entropies.

Statement 2 is false. Moralization of a directed graph in graphical models adds edges between the parents of a node in order to represent conditional independence relationships. This generally increases the number of edges present in the graph, not decreases.

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. According to the properties of entropy, the joint entropy of two variables is always greater than or equal to the sum of their individual entropies.

Statement 2 is false. Moralization is a process in graph theory where the directional edges in a directed graph are removed to form an undirected graph. This typically results in an increase in the number of edges, not a decrease.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 5:

Statement 1: For any two variables  $x$  and  $y$  having joint distribution  $p(x, y)$ , we always have  $H[x, y] \geq H[x] + H[y]$  where  $H$  is entropy function. This statement is true and it is a direct consequence of the subadditivity property of entropy.

Statement 2: For some directed graphs, moralization decreases the number of edges present in the graph. This statement is false. Moralization is a process in graph theory where the directed edges of a graph are replaced with undirected edges between nodes that share a common child. The resulting moralized graph will have the same number of edges as the original directed graph or possibly more due to the addition of undirected edges.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1: For any two variables  $x$  and  $y$  having joint distribution  $p(x, y)$ , we always have  $H[x, y] \geq H[x] + H[y]$  where  $H$  is entropy function.

This statement is true, as the joint entropy  $H[x, y]$  is always greater than or equal to the sum of the individual entropies  $H[x]$  and  $H[y]$  based on the properties of entropy.

Statement 2: For some directed graphs, moralization decreases the number of edges present in the graph.

This statement is false. Moralization is a process in graphical models where the directed edges are replaced with undirected edges between the parents of a node. This typically results in an increase in the number of edges in the graph, not a decrease.

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1: For any two variables  $x$  and  $y$  having joint distribution  $p(x, y)$ , we always have  $H[x, y] \geq H[x] + H[y]$  where  $H$  is the entropy function. This statement is true as per the property of entropy.

Statement 2: For some directed graphs, moralization decreases the number of edges present in the graph. This statement is false. Moralization is a process in graphical models where the directed edges are converted into undirected edges by adding additional edges. This process typically increases the number of edges in the graph.

Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1: For any two variables  $x$  and  $y$  having joint distribution  $p(x, y)$ , we always have  $H[x, y] \geq H[x] + H[y]$  where  $H$  is the entropy function.

This statement is true based on the subadditivity property of entropy. The joint entropy  $H[x, y]$  is always greater than or equal to the sum of the individual entropies  $H[x]$  and  $H[y]$ .

Statement 2: For some directed graphs, moralization decreases the number of edges present in the graph.

This statement is false. Moralization is a process in which the directed edges in a graph are replaced with undirected edges. This can result in an increase in the number of edges in the graph, not a decrease.

Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1: For any two variables  $x$  and  $y$  having joint distribution  $p(x, y)$ , we always have  $H[x, y] \geq H[x] + H[y]$  where  $H$  is entropy function.

This statement is true. The entropy of the joint distribution  $H[x, y]$  is always greater than or equal to the sum of the entropies of the individual variables  $H[x]$  and  $H[y]$ . Thus, the correct answer for statement 1 is True.

Statement 2: For some directed graphs, moralization decreases the number of edges present in the graph.

This statement is false. Moralization is a process in graphical models where directed edges are transformed into undirected edges. This process can introduce additional edges, but it does not decrease the total number of edges in the graph. Thus, the correct answer for statement 2 is False.

Final answer: C

\*\*\*\*\*

Question 96:

'Which of the following is NOT supervised learning?', A. 'PCA', B. 'Decision Tree', C. 'Linear Regression', D. 'Naive Bayesian'

=====

Answer Option 1:

PCA stands for Principal Component Analysis, which is an unsupervised learning technique used for dimensionality reduction. Therefore, PCA is NOT supervised learning. Final answer: A

\*\*\*\*\*

Answer Option 2:

PCA (Principal Component Analysis) is a dimensionality reduction technique and is not a supervised learning algorithm. It is an unsupervised learning method used for feature extraction and data visualization. Decision tree, linear regression, and Naive Bayesian are all examples of supervised learning algorithms. Final answer: A

\*\*\*\*\*

Answer Option 3:

Supervised learning is a type of machine learning where a model is trained using labeled data. PCA (Principal Component Analysis) is an unsupervised learning technique used for dimensionality reduction and feature extraction. Therefore, PCA is NOT supervised learning. Final answer: A

\*\*\*\*\*

Answer Option 4:

PCA (Principal Component Analysis) is not an example of supervised learning. It is an unsupervised learning technique used for dimensionality reduction and feature extraction. Decision Tree, Linear Regression, and Naive Bayesian are examples of supervised learning algorithms. Final answer: A

\*\*\*\*\*

Answer Option 5:

PCA (Principal Component Analysis) is not a supervised learning algorithm. It is an unsupervised learning technique used for dimensionality reduction and feature extraction. Decision Tree, Linear Regression, and Naive Bayesian are examples of supervised learning algorithms. Final answer: A

\*\*\*\*\*

Answer Option 6:

PCA (Principal Component Analysis) is not a supervised learning algorithm. It is an unsupervised learning technique used for dimensionality reduction and feature extraction. Decision Tree, Linear Regression, and Naive Bayesian are examples of supervised learning algorithms. Final answer: A

\*\*\*\*\*

Answer Option 7:

Principal Component Analysis (PCA) is not a supervised learning algorithm. It is an unsupervised learning technique used for dimensionality reduction and feature extraction. Decision Tree, Linear Regression, and Naive Bayesian are all examples of supervised learning algorithms. Final answer: A

\*\*\*\*\*

Answer Option 8:

Principal Component Analysis (PCA) is not a supervised learning algorithm. It is an unsupervised learning algorithm used for dimensionality reduction. Decision tree, linear regression, and Naive Bayes are examples of supervised learning algorithms. Final answer: A

\*\*\*\*\*

Answer Option 9:

PCA (Principal Component Analysis) is not a supervised learning algorithm. It is an unsupervised learning algorithm used for dimensionality reduction. Decision tree, linear regression, and naive Bayesian are examples of supervised learning algorithms. Final answer: A

\*\*\*\*\*

Question 97:

'Statement 1| A neural network's convergence depends on the learning rate. Statement 2| Dropout multiplies randomly chosen activation values by zero.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 | A neural network's convergence depends on the learning rate.

This statement is true. The learning rate determines the step size at each iteration of the training process and can affect the convergence speed and stability of a neural network.

Statement 2 | Dropout multiplies randomly chosen activation values by zero.

This statement is true. Dropout is a regularization technique used in neural networks where randomly selected neurons are ignored or "dropped out" during training by multiplying their activation values by zero.

Final answer: C - True, False

\*\*\*\*\*

Answer Option 2:

Statement 1 is true. The learning rate in a neural network affects how quickly the model converges or reaches an optimal solution.

Statement 2 is false. Dropout randomly sets a fraction of the input units to zero during training, not the activation values. Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is true. The convergence of a neural network can be affected by the learning rate. Statement 2 is false. Dropout randomly sets a fraction of the input units to zero during training, not the activation values. Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. The learning rate determines the step size at each iteration in the training process and can affect how quickly or slowly a neural network converges.

Statement 2 is false. Dropout randomly sets a fraction of input units to zero during training, not activation values. Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is true. The learning rate in a neural network affects how quickly the network converges or learns the optimal weights.

Statement 2 is false. Dropout randomly sets a fraction of the input units to zero during training to prevent overfitting, but it does not multiply randomly chosen activation values by zero.

Final answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is true. The learning rate in a neural network determines the step size at each iteration and can impact how quickly or slowly the network converges.

Statement 2 is false. Dropout is a regularization technique in neural networks where randomly selected neurons are ignored or "dropped out" during training. It does not multiply activation values by zero. Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1 is true. The learning rate is a hyperparameter that determines the step size at each iteration of the optimization algorithm and can affect the convergence of a neural network.

Statement 2 is false. Dropout randomly sets a fraction of the input activation values to zero during training, but does not multiply by zero. Final answer: C

\*\*\*\*\*

Answer Option 8:

Statement 1 is true. The learning rate is a hyperparameter that determines the step size at each iteration of the neural network training process. It affects how quickly the model converges to the optimal solution.

Statement 2 is false. Dropout is a regularization technique used in neural networks to prevent overfitting. It randomly sets a fraction of the activation values to zero during training, not multiply them by zero.

Therefore, the correct answer is C. 'True, False'.

\*\*\*\*\*

Answer Option 9:

Statement 1 is true. The learning rate is an important hyperparameter that determines the step size at each iteration during training and can affect the convergence of a neural network.

Statement 2 is also true. Dropout is a regularization technique commonly used in neural networks. It randomly sets a fraction of the input units to zero at each training iteration, which can help prevent overfitting.

Final Answer: A

\*\*\*\*\*

Question 98:

'Which one of the following is equal to  $P(A, B, C)$  given Boolean random variables  $A, B$  and  $C$ , and no independence or conditional independence assumptions between any of them?', A. ' $P(A | B) * P(B | C) * P(C | A)$ ', B. ' $P(C | A, B) * P(A) * P(B)$ ', C. ' $P(A, B | C) * P(C)$ ', D. ' $P(A | B, C) * P(B | A, C) * P(C | A, B)$ '

=====

Answer Option 1:

If there are no independence or conditional independence assumptions between the Boolean random variables  $A, B$ , and  $C$ , then the correct expression for  $P(A, B, C)$  is  $P(C | A, B) * P(A | B) * P(B)$ . Final answer: None of the above.

=====

Answer Option 2:

If there are no independence or conditional independence assumptions between Boolean random variables  $A, B$ , and  $C$ , then the correct expression for  $P(A, B, C)$  would be  $P(C | A, B) * P(A | B, C) * P(B | A, C)$ . Final answer: D

=====

Answer Option 3:

Given that there are no independence or conditional independence assumptions between any of the Boolean random variables  $A, B$ , and  $C$ , the correct formula to calculate  $P(A, B, C)$  is:  $B. 'P(C | A, B) * P(A) * P(B)'$ . Final answer: B

=====

Answer Option 4:

Given no independence or conditional independence assumptions between the Boolean random variables  $A, B$ , and  $C$ , the correct expression for  $P(A, B, C)$  would be:

C.  $P(A, B | C) * P(C)$

This expression represents the joint probability of  $A$  and  $B$  given  $C$ , multiplied by the probability of  $C$ .

Final answer: C

=====

Answer Option 5:

Given that there are no independence or conditional independence assumptions between the Boolean random variables  $A, B$ , and  $C$ , the correct expression for  $P(A, B, C)$  would be the joint probability distribution of  $A, B$ , and  $C$ . Therefore, the correct answer is C:  $P(A, B | C) * P(C)$ .

=====

Answer Option 6:

The correct answer is C. According to the definition of joint probability, the probability of  $A, B$ , and  $C$  occurring together ( $P(A, B, C)$ ) can be calculated as the conditional probability of  $A$  and  $B$  given  $C$  ( $P(A, B | C)$ ) multiplied by the probability of  $C$  ( $P(C)$ ), or in short,  $P(A, B | C) * P(C)$ . Final answer: C.

=====

Answer Option 7:

Given no independence or conditional independence assumptions between  $A, B$ , and  $C$ , the joint probability distribution  $P(A, B, C)$  can be expressed as the product of the conditional probabilities  $P(A | B, C)$ ,  $P(B | A, C)$ , and  $P(C | A, B)$ . Therefore, the correct answer is D.  $P(A | B, C) * P(B | A, C) * P(C | A, B)$ .

=====

Answer Option 8:

If there are no independence or conditional independence assumptions between any of the Boolean random variables  $A, B$ , and  $C$ , then the joint probability distribution  $P(A, B, C)$  can be expressed as the product of the individual probabilities:

$P(A, B, C) = P(A) * P(B) * P(C)$

Therefore, the correct answer is none of the options provided. Final answer: None of the above.

=====

Answer Option 9:

Given no independence or conditional independence assumptions between  $A, B$ , and  $C$ , the correct expression for  $P(A, B, C)$  is: C. ' $P(A, B | C) * P(C)$ '.

=====

Question 99:

'Which of the following tasks can be best solved using Clustering.', A. 'Predicting the amount of rainfall based on various cues', B. 'Detecting fraudulent credit card transactions', C. 'Training a robot to solve a maze', D. 'All of the above'

=====

Answer Option 1:

Clustering is a technique used to group similar data points together based on their characteristics. It is best suited for tasks where the goal is to identify patterns, similarities, or groupings within the data.

Among the given options, detecting fraudulent credit card transactions and training a robot to solve a maze do not align with the typical use cases for clustering. Therefore, the correct answer is A. 'Predicting the amount of rainfall based on various cues'. Final answer: A

\*\*\*\*\*

Answer Option 2:

Clustering is a method used to group similar data points together based on their characteristics. It is best suited for tasks where we want to discover patterns or structure in unlabeled data. Among the given options, predicting the amount of rainfall and detecting fraudulent credit card transactions are more suited for other techniques such as regression and anomaly detection, respectively. Training a robot to solve a maze could potentially involve clustering, but it is not the most appropriate method for this task. Therefore, the correct answer is: A. 'Predicting the amount of rainfall based on various cues'.

\*\*\*\*\*

Answer Option 3:

Clustering is a machine learning technique used to group similar data points together based on their characteristics. None of the given tasks involve grouping similar data points together, so none of them can be best solved using clustering. Therefore, the correct answer is: D. 'None of the above'.

\*\*\*\*\*

Answer Option 4:

Clustering is a technique used to group similar data points together based on their characteristics. It is best suited for tasks like detecting patterns or grouping similar data points. Out of the given options, the task that can be best solved using clustering is 'predicting the amount of rainfall based on various cues'. Clustering can help identify patterns and group similar regions based on rainfall patterns. Final answer:

A

\*\*\*\*\*

Answer Option 5:

Clustering is a technique used to group similar data points together based on their characteristics or patterns. Out of the given options, detecting fraudulent credit card transactions can be best solved using clustering. This is because clustering can help identify patterns or anomalies in a large dataset without any predefined labels. Predicting the amount of rainfall and training a robot to solve a maze are not typical use cases for clustering. Final answer: B

\*\*\*\*\*

Answer Option 6:

Clustering is a technique used to group similar data points together based on their characteristics or features. It is commonly used for tasks such as customer segmentation or data exploration. Among the options provided, the task that can be best solved using clustering is A. 'Predicting the amount of rainfall based on various cues'. Clustering can help identify patterns or groups within the data that may be indicative of different rainfall levels. Therefore, the correct answer is A.

\*\*\*\*\*

Answer Option 7:

Clustering is a technique used to group similar data points together based on their characteristics. It is best suited for tasks where the goal is to identify patterns or similarities in the data. Among the options given, predicting the amount of rainfall, detecting fraudulent credit card transactions, and training a robot to solve a maze do not involve clustering. Therefore, the correct answer is not D. Final answer: None of the above

\*\*\*\*\*

Answer Option 8:

Clustering is a machine learning technique used to group similar data points together. It is best suited for tasks where the goal is to discover hidden patterns or groupings within the data. Out of the given options, predicting the amount of rainfall based on various cues and training a robot to solve a maze do not involve grouping or discovering patterns. On the other hand, detecting fraudulent credit card transactions can be best solved using clustering as it can help identify clusters of transactions that exhibit similar fraudulent behavior. Final answer: B

\*\*\*\*\*

Answer Option 9:

Clustering is a technique used to group similar data points together based on their characteristics. It is best used for tasks such as detecting patterns or grouping data without specific labels. Therefore, option D, 'All of the above', is not the correct answer. Predicting the amount of rainfall (option A) and training a robot to solve a maze (option C) involve specific predictions or actions based on given cues or goals, making them better suited for other techniques. However, detecting fraudulent credit card transactions (option B) can be effectively solved using clustering to identify anomalies in the transaction data. Final answer: B

\*\*\*\*\*

Question 100:

'After applying a regularization penalty in linear regression, you find that some of the coefficients of  $w$  are zeroed out. Which of the following penalties might have been used?', A. 'L0 norm', B. 'L1 norm', C. 'L2 norm', D. 'either (a) or (b)'

=====

Answer Option 1:

When some coefficients of  $w$  are zeroed out after applying a regularization penalty in linear regression, it suggests that the L1 norm (also known as Lasso regularization) might have been used. The L1 norm encourages sparsity in the coefficients, effectively selecting a subset of the most important features. Final answer: B

=====

Answer Option 2:

When some of the coefficients in linear regression are zeroed out after applying a regularization penalty, it suggests that either the L1 norm penalty (also known as Lasso regularization) or the L0 norm penalty (also known as the sparse regularization) might have been used. Both penalties can lead to sparse solutions by encouraging some coefficients to be exactly zero. Final answer: D

=====

Answer Option 3:

When some coefficients in linear regression are zeroed out after regularization, it suggests that the L1 norm (also known as Lasso regularization) might have been used. L1 norm encourages sparsity and can result in feature selection by shrinking less important coefficients to zero. Final answer: B

=====

Answer Option 4:

When some of the coefficients of  $w$  are zeroed out after applying a regularization penalty in linear regression, it suggests that the L1 norm penalty (also known as Lasso regularization) might have been used. L1 norm penalty promotes sparsity in the model by encouraging some coefficients to be exactly zero. Final answer: B

=====

Answer Option 5:

When some of the coefficients of  $w$  are zeroed out after applying a regularization penalty in linear regression, it suggests that the L1 norm penalty, also known as Lasso regularization, might have been used. The L1 norm encourages sparse solutions by promoting some coefficients to exactly zero. Final answer: B

=====

Answer Option 6:

When some of the coefficients in linear regression are zeroed out after applying a regularization penalty, it indicates that the L1 norm penalty (also known as Lasso regularization) might have been used. The L1 norm penalty encourages sparsity in the coefficients, effectively performing feature selection. Final answer: B

=====

Answer Option 7:

If some of the coefficients of  $w$  are zeroed out after applying a regularization penalty in linear regression, it suggests that an L1 norm penalty, also known as Lasso regularization, might have been used. L1 norm regularization encourages sparsity in the model by shrinking some coefficients to exactly zero. The L0 norm is not typically used as a regularization penalty in linear regression. Final answer: B

=====

Answer Option 8:

When some coefficients in linear regression are zeroed out after applying a regularization penalty, it suggests that the L1 norm regularization penalty (also known as Lasso regularization) might have been used. The L1 norm penalty encourages sparsity in the coefficients and can result in some coefficients being exactly zero. Final answer: B

=====

Answer Option 9:

When some of the coefficients in linear regression are zeroed out after applying a regularization penalty, it indicates that the L1 norm (also known as Lasso regularization) might have been used. The L1 norm encourages sparsity in the coefficients and can lead to some of them being precisely zero. Final answer: B

=====



Question 101:

'A and B are two events. If  $P(A, B)$  decreases while  $P(A)$  increases, which of the following is true?', A. ' $P(A|B)$  decreases', B. ' $P(B|A)$  decreases', C. ' $P(B)$  decreases', D. 'All of above'

=====

Answer Option 1:

If the joint probability  $P(A, B)$  decreases while the probability of A,  $P(A)$ , increases, it means that A and B are becoming less likely to occur together. In this case, it implies that the probability of event B,  $P(B)$ , decreases. Therefore, the correct answer is C. Final answer: C

=====

Answer Option 2:

If the probability of two events A and B, denoted as  $P(A, B)$ , decreases while the probability of event A, denoted as  $P(A)$ , increases, it implies that events A and B are negatively correlated. In such cases, the conditional probability  $P(A|B)$  decreases as well. However, it does not necessarily mean that  $P(B|A)$  decreases or that  $P(B)$  decreases. Therefore, the correct answer is A: ' $P(A|B)$  decreases'.

=====

Answer Option 3:

If  $P(A, B)$  decreases while  $P(A)$  increases, it means that A and B are becoming less likely to occur together. From this information, we can deduce that  $P(A|B)$  decreases, because the probability of event A occurring given that event B has occurred decreases. However, we cannot determine the changes in  $P(B)$  or  $P(B|A)$  based on the given information. Therefore, the correct answer is A.

=====

Answer Option 4:

If  $P(A, B)$  decreases while  $P(A)$  increases, it means that the probability of both events A and B occurring together is decreasing, while the probability of event A occurring is increasing. This implies that the events A and B are becoming more independent of each other.

Given this information, we can conclude that the probability of event A occurring given that event B has occurred ( $P(A|B)$ ) would decrease. Therefore, the correct answer is A: ' $P(A|B)$  decreases'.

=====

Answer Option 5:

If  $P(A, B)$  decreases while  $P(A)$  increases, it means that the events A and B are becoming less dependent. In this case, the conditional probability  $P(A|B)$  is likely to decrease, so the answer is A.  $P(B|A)$  and  $P(B)$  may or may not change in this scenario, so options B and C cannot be concluded. Final answer: A.

=====

Answer Option 6:

If  $P(A, B)$  decreases while  $P(A)$  increases, it implies that events A and B are becoming less dependent on each other. In this case, the probability of A given B, denoted as  $P(A|B)$ , is expected to decrease.

However, the probability of B given A, denoted as  $P(B|A)$ , and the probability of B, denoted as  $P(B)$ , may or may not be affected. Therefore, the correct answer is: A. ' $P(A|B)$  decreases'.

=====

Answer Option 7:

If the joint probability  $P(A, B)$  decreases while the probability of event A,  $P(A)$ , increases, we can conclude that events A and B are negatively correlated. In such cases, as the probability of A increases, the probability of B decreases.

Based on this information, we can deduce that  $P(B)$  decreases because of the negative correlation between A and B. However, we cannot make any conclusions about the conditional probabilities  $P(A|B)$  or  $P(B|A)$  with the given information.

Therefore, the correct answer is C. ' $P(B)$  decreases'.

=====

Answer Option 8:

If  $P(A, B)$  decreases while  $P(A)$  increases, this suggests that events A and B are negatively correlated. In this case, the probability of A given B ( $P(A|B)$ ) would decrease, as the occurrence of event B would make event A less likely. Therefore, the correct answer is A.  $P(B|A)$  may increase or decrease depending on the specific correlation between events A and B.  $P(B)$  may also increase or decrease depending on the overall probability distribution. Final answer: A

=====

Answer Option 9:

If the probability of event A and B occurring together,  $P(A, B)$ , decreases while the probability of event A,  $P(A)$ , increases, it implies that event A and B are dependent. In this case, the probability of A given B,  $P(A|B)$ , may either increase, decrease, or remain the same. Similarly, the probability of B given A,  $P(B|A)$ , may also increase, decrease, or remain the same. However, the probability of event B,  $P(B)$ , does not have to decrease in this scenario. Therefore, the correct answer is D. None of the above.

=====

Question 102:

'Statement 1| When learning an HMM for a fixed set of observations, assume we do not know the true number of hidden states (which is often the case), we can always increase the training data likelihood by permitting more hidden states. Statement 2| Collaborative filtering is often a useful model for modeling users' movie preference.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is false. Increasing the number of hidden states in a Hidden Markov Model (HMM) may not necessarily increase the training data likelihood. Adding more hidden states can lead to overfitting and may not improve the model's performance.

Statement 2 is true. Collaborative filtering is indeed a useful model for modeling users' movie preferences.

Final answer: C

=====

Answer Option 2:

Statement 1 is false. Increasing the number of hidden states in a hidden Markov model (HMM) does not necessarily increase the training data likelihood. In fact, adding more hidden states without sufficient data can lead to overfitting.

Statement 2 is true. Collaborative filtering is indeed a useful model for modeling users' movie preferences.

Final answer: C

=====

Answer Option 3:

Statement 1: When learning an HMM for a fixed set of observations, if we do not know the true number of hidden states, increasing the number of hidden states does not always guarantee an increase in the training data likelihood. Therefore, statement 1 is False.

Statement 2: Collaborative filtering is indeed often a useful model for modeling users' movie preferences. Therefore, statement 2 is True.

Final Answer: C

=====

Answer Option 4:

Statement 1: When learning an HMM for a fixed set of observations, increasing the number of hidden states does not always guarantee an increase in training data likelihood. Adding more hidden states can lead to overfitting and decrease the model's performance. Therefore, Statement 1 is false.

Statement 2: Collaborative filtering is indeed a useful model for modeling users' movie preferences. It is a widely used technique in recommendation systems. Therefore, Statement 2 is true.

Final answer: C

=====

Answer Option 5:

Statement 1: When learning an HMM for a fixed set of observations, increasing the number of hidden states does not necessarily increase the training data likelihood. In fact, adding more hidden states can lead to overfitting and decrease the model's performance. Therefore, Statement 1 is False.

Statement 2: Collaborative filtering is indeed a useful model for modeling users' movie preferences. Collaborative filtering leverages the similarity between users' preferences to make predictions and recommendations. Therefore, Statement 2 is True.

Final answer: C

=====

Answer Option 6:

Statement 1 | When learning an HMM for a fixed set of observations, assuming we do not know the true number of hidden states (which is often the case), we cannot always increase the training data likelihood by permitting more hidden states. Increasing the number of hidden states may lead to overfitting the training data and decrease the model's ability to generalize to new data. Therefore, Statement 1 is False.

Statement 2 | Collaborative filtering is indeed often a useful model for modeling users' movie preferences. Collaborative filtering techniques analyze the preferences and behavior of users to make recommendations based on similarities among users or items. Therefore, Statement 2 is True.

Final answer: C

=====

Answer Option 7:

Statement 1 is false. Increasing the number of hidden states in an HMM does not necessarily increase the training data likelihood. Adding more hidden states can lead to overfitting and decrease the model's ability to generalize.

Statement 2 is true. Collaborative filtering is a useful model for modeling users' movie preferences. It is commonly used in recommendation systems to predict users' preferences based on the preferences of similar users.

Final answer: C

=====

Answer Option 8:

Statement 1 is False. Increasing the number of hidden states in a Hidden Markov Model (HMM) does not necessarily guarantee an increase in training data likelihood. Adding more hidden states can lead to overfitting and worsening of the model's performance.

Statement 2 is True. Collaborative filtering is indeed a useful model for modeling users' movie preferences. It is widely used in recommendation systems to predict user preferences based on their past behavior and similarities with other users. Final answer: C

\*\*\*\*\*

Answer Option 9:

Statement 1 is false. Increasing the number of hidden states in a Hidden Markov Model (HMM) does not always lead to an increase in the training data likelihood. Overfitting can occur if the number of hidden states is increased without proper justification.

Statement 2 is true. Collaborative filtering is a commonly used model for modeling users' movie preferences, as it leverages the behavior and preferences of similar users to make recommendations.

Final Answer: C

\*\*\*\*\*

Question 103:

'You are training a linear regression model for a simple estimation task, and notice that the model is overfitting to the data. You decide to add in  $\ell_2$  regularization to penalize the weights. As you increase the  $\ell_2$  regularization coefficient, what will happen to the bias and variance of the model?', A. 'Bias increase ; Variance increase', B. 'Bias increase ; Variance decrease', C. 'Bias decrease ; Variance increase', D. 'Bias decrease ; Variance decrease'

=====

Answer Option 1:

When  $\ell_2$  regularization is applied to a linear regression model, increasing the regularization coefficient will lead to a decrease in variance (the model becomes less sensitive to the training data) but an increase in bias (the model becomes less flexible and may underfit the data). Therefore, the correct answer is C: Bias decrease; Variance increase.

\*\*\*\*\*

Answer Option 2:

When  $\ell_2$  regularization is increased, it penalizes the weights in the model, leading to a decrease in the variance of the model. This is because it reduces the complexity of the model and prevents overfitting. However, as the bias-variance tradeoff suggests, increasing regularization can also increase the bias of the model. Therefore, the correct answer is A: Bias increase ; Variance increase.

\*\*\*\*\*

Answer Option 3:

As you increase the  $\ell_2$  regularization coefficient, the bias of the model will increase and the variance will decrease. Regularization helps reduce overfitting by adding a penalty to the model's weights, which makes the model less complex and more generalizable. This reduction in complexity leads to an increase in bias but a decrease in variance. Final answer: A

\*\*\*\*\*

Answer Option 4:

As you increase the  $\ell_2$  regularization coefficient, the bias of the model will decrease because the regularization term helps to shrink the weights towards zero, reducing the complexity of the model. However, the variance of the model will increase because the regularization term limits the flexibility of the model, making it less capable of fitting the training data precisely. Final answer: C

\*\*\*\*\*

Answer Option 5:

When  $\ell_2$  regularization is increased, it penalizes the weights of the model more strongly, which leads to a reduction in the variance of the model. This is because the regularization term helps to prevent the model from fitting the noise in the training data. However, increasing  $\ell_2$  regularization can also cause a slight increase in the bias of the model, as it constrains the model to simpler solutions. Final answer: C

\*\*\*\*\*

Answer Option 6:

When we increase the  $\ell_2$  regularization coefficient in a linear regression model, it helps to shrink the weights towards zero. This reduces the complexity of the model and prevents overfitting. As a result, the bias of the model increases because it becomes less flexible in fitting the training data. At the same time, the variance of the model decreases because it becomes less sensitive to variations in the training data. Final answer: B

\*\*\*\*\*

Answer Option 7:

As the  $\ell_2$  regularization coefficient increases, the bias of the model will decrease, but the variance will increase. Regularization helps to reduce overfitting and improve generalization by adding a penalty term to the weights. This penalty term encourages the model to have smaller weights, reducing the complexity of the model and decreasing its bias. However, as the regularization becomes stronger, the model becomes more constrained, leading to an increase in its variance. Final answer: C

\*\*\*\*\*

Answer Option 8:

As you increase the  $\ell_2$  regularization coefficient, the bias of the model will decrease, but the variance will increase. Regularization helps in reducing overfitting by adding a penalty to the weights, which reduces the complexity of the model. This reduction in complexity leads to a decrease in bias. However, increasing the regularization coefficient too much can lead to an increase in the variance of the model, as it may start underfitting the data. Final answer: C

\*\*\*\*\*

Answer Option 9:

When you increase the  $\ell_2$  regularization coefficient, the bias of the model will decrease while the variance will increase. This is because the regularization term adds a penalty to the model's weights, discouraging them from taking on large values. As a result, the model becomes less complex and more biased towards the training data. However, this bias reduction comes at the cost of increased variance, as the model becomes more sensitive to the specific training data it was trained on. Final answer: C

\*\*\*\*\*

Question 104:

'Which PyTorch 1.8 command(s) produce  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$  and a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ ?, A. `torch.randn(10,5) * 16`; `torch.rand(10,10,low=-1,high=1)`, B. `torch.randn(10,5) * 16`; `torch.rand(10,10) - 0.5`, C. `torch.randn(10,5) * 4`; `torch.rand(10,10) - 1`, D. `torch.normal(torch.ones(10,5)*5,torch.ones(5,5)*16)`; `torch.rand(10,10) - 1`

=====

Answer Option 1:

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , the correct command is: `torch.randn(10,5) * 4`.

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ , the correct command is: `torch.rand(10,10) - 1`.

Therefore, the correct answer is C.

\*\*\*\*\*

Answer Option 2:

To generate a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , the correct command is `torch.randn(10,5) * 4`.

To generate a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ , the correct command is `torch.rand(10,10) - 1`.

Therefore, the correct answer is C. `torch.randn(10,5) * 4`; `torch.rand(10,10) - 1`.

\*\*\*\*\*

Answer Option 3:

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , we can use the command `torch.randn(10,5) * 4`.

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ , we can use the command `torch.rand(10,10) - 1`.

Therefore, the correct command is A. `torch.randn(10,5) * 16`; `torch.rand(10,10,low=-1,high=1)`.

\*\*\*\*\*

Answer Option 4:

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , the correct command is: `torch.randn(10,5) * 4` (option C is correct).

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ , the correct command is: `torch.rand(10,10) - 1` (option C is correct).

Final answer: C

\*\*\*\*\*

Answer Option 5:

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , the correct command is: `5 + torch.randn(10,5) * 4`, as the standard deviation is  $\sigma=4$ , not  $\sigma=16$ .

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ , the correct command is: `2 * torch.rand(10,10) - 1`, as it correctly generates random numbers from a uniform distribution on the interval  $[-1, 1]$ .

Final answer: C

\*\*\*\*\*

Answer Option 6:

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , the correct command is: `torch.randn(10,5) * 4`.

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ , the correct command is: `torch.rand(10,10) - 1`.

Therefore, the correct answer is: C. `torch.randn(10,5) * 4`; `torch.rand(10,10) - 1`.

\*\*\*\*\*

Answer Option 7:

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , you can use the command: `torch.randn(10,5) * 4`.

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1, 1]$ , you can use the command: `torch.rand(10, 10) - 1`.

Therefore, the correct answer is C. `torch.randn(10,5) * 4`; `torch.rand(10,10) - 1`.

\*\*\*\*\*

Answer Option 8:

The correct answer would be B.

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , we can use the command: `5 + torch.randn(10, 5) * 4`. This adds 5 to the Gaussian random numbers with mean 0 and standard deviation 1, and then scales them by 4 to match the desired mean and variance.

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $\mathcal{U}[-1,1]$ , we can use the command: `(torch.rand(10, 10) - 0.5) / 0.5`. This generates uniform random numbers between 0 and 1, shifts them to be between -0.5 and 0.5, and then scales them by 2 to match the desired range.

Final answer: B

\*\*\*\*\*

Answer Option 9:

To produce a  $10 \times 5$  Gaussian matrix with each entry i.i.d. sampled from  $\mathcal{N}(\mu=5, \sigma^2=16)$ , the correct command is `torch.randn(10, 5) * 4`. This is because the variance is the square of the standard deviation, so we need to use  $\sqrt{16} = 4$  as the standard deviation.

To produce a  $10 \times 10$  uniform matrix with each entry i.i.d. sampled from  $U[-1, 1]$ , the correct command is `2 * torch.rand(10, 10) - 1`. This is because `torch.rand` generates samples from a uniform distribution on the interval  $[0, 1]$ , so we need to transform it to the desired interval  $[-1, 1]$ .

Therefore, the correct answer is C. `5 + torch.randn(10, 5) * 4`; `2 * torch.rand(10, 10) - 1`.

\*\*\*\*\*

Question 105:

'Statement 1| The ReLU's gradient is zero for  $x < 0$ , and the sigmoid gradient  $\sigma(x)(1-\sigma(x)) \leq \frac{1}{4}$  for all  $x$ . Statement 2| The sigmoid has a continuous gradient and the ReLU has a discontinuous gradient.', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1 is False. The ReLU's gradient is zero for  $x < 0$ , but the sigmoid gradient  $\sigma(x)(1-\sigma(x))$  can be less than or greater than  $\frac{1}{4}$  for different values of  $x$ .

Statement 2 is True. The sigmoid function has a continuous gradient, while the ReLU function has a discontinuous gradient at  $x=0$ .

Final answer: C

\*\*\*\*\*

Answer Option 2:

Statement 1 is True. The gradient of the ReLU (Rectified Linear Unit) is zero for  $x < 0$ , and the gradient of the sigmoid function is bounded by  $\frac{1}{4}$ .

Statement 2 is False. The sigmoid function has a continuous gradient, while the gradient of the ReLU function is discontinuous at  $x = 0$ .

Final answer: C

\*\*\*\*\*

Answer Option 3:

Statement 1 is False. The ReLU's gradient is zero for  $x < 0$ , but it is not zero for  $x > 0$ . The sigmoid gradient  $\sigma(x)(1-\sigma(x))$  can be greater than  $\frac{1}{4}$  for certain values of  $x$ .

Statement 2 is True. The sigmoid function has a continuous gradient, while the ReLU function has a discontinuous gradient at  $x=0$ .

Final answer: C

\*\*\*\*\*

Answer Option 4:

Statement 1 is true. The ReLU's gradient is indeed zero for  $x < 0$ , and the sigmoid gradient  $\sigma(x)(1-\sigma(x)) \leq \frac{1}{4}$  for all  $x$ .

Statement 2 is false. The sigmoid function does have a continuous gradient, but the ReLU function has a discontinuous gradient at  $x=0$ .

Final answer: C

\*\*\*\*\*

Answer Option 5:

Statement 1 is False. The ReLU's gradient is zero for  $x < 0$ , but for the sigmoid function, the gradient  $\sigma(x)(1-\sigma(x))$  does not always satisfy  $\sigma(x)(1-\sigma(x)) \leq \frac{1}{4}$  for all  $x$ .

Statement 2 is True. The sigmoid function does have a continuous gradient, while the ReLU function has a discontinuous gradient at  $x=0$ .

Final Answer: C

\*\*\*\*\*

Answer Option 6:

Statement 1 is False. The ReLU's gradient is zero for  $x < 0$ , but it is not necessarily zero for  $x > 0$ . The sigmoid gradient  $\sigma(x)(1-\sigma(x))$  is also not always less than or equal to  $\frac{1}{4}$ .

Statement 2 is True. The sigmoid function has a continuous gradient, while the ReLU function has a discontinuous gradient at  $x=0$ .

Final answer: C

\*\*\*\*\*

Answer Option 7:

Statement 1 is False. The ReLU's gradient is zero for  $x < 0$ , but for the sigmoid function, the gradient  $\sigma(x)(1-\sigma(x))$  is always between 0 and 0.25, not necessarily  $\leq \frac{1}{4}$ .

Statement 2 is True. The sigmoid function has a continuous gradient, while the ReLU function has a discontinuous gradient at  $x=0$ .

Final answer: C

\*\*\*\*\*

Answer Option 8:

The correct answer is C. 'True, False'.

Statement 1 is true. The ReLU function has a gradient of zero for input values less than zero, which means its derivative is zero for  $x < 0$ . The sigmoid function has a gradient given by  $\sigma(x)(1-\sigma(x))$ , which is always less than or equal to  $\frac{1}{4}$  for all values of  $x$ .

Statement 2 is false. The sigmoid function does have a continuous gradient for all input values, while the ReLU function has a discontinuous gradient at  $x = 0$ .

\*\*\*\*\*

Answer Option 9:

Statement 1: The ReLU's gradient is zero for  $x < 0$ , and the sigmoid gradient  $\sigma(x)(1-\sigma(x)) \leq \frac{1}{4}$  for all  $x$ .

This statement is true. The ReLU (Rectified Linear Unit) has a gradient of zero for negative values of  $x$ , while the sigmoid function has a gradient bounded by  $\frac{1}{4}$  for all values of  $x$ .

Statement 2: The sigmoid has a continuous gradient and the ReLU has a discontinuous gradient.

This statement is false. The sigmoid function has a continuous gradient for all values of  $x$ , while the ReLU function also has a continuous gradient except at  $x=0$ , where it has a subgradient of zero.

Final answer: C. 'True, False'

\*\*\*\*\*

Question 106:

'Which is true about Batch Normalization?', A. 'After applying batch normalization, the layer's activations will follow a standard Gaussian distribution.', B. 'The bias parameter of affine layers becomes redundant if a batch normalization layer follows immediately afterward.', C. 'The standard weight initialization must be changed when using Batch Normalization.', D. 'Batch Normalization is equivalent to Layer Normalization for convolutional neural networks.'

=====

Answer Option 1:

B. 'The bias parameter of affine layers becomes redundant if a batch normalization layer follows immediately afterward.' is true about Batch Normalization. Batch normalization introduces parameters that shift and scale the normalized values, making the bias term in the affine layer unnecessary. Final answer: B

\*\*\*\*\*

Answer Option 2:

Option B is true about Batch Normalization. When a batch normalization layer follows immediately after an affine layer, the bias parameter of the affine layer becomes redundant.

Option A is false. Batch normalization does not guarantee that the layer's activations will follow a standard Gaussian distribution, although it can help in reducing the internal covariate shift.

Option C is false. The standard weight initialization does not necessarily need to be changed when using Batch Normalization.

Option D is false. Batch Normalization and Layer Normalization are different techniques and are not equivalent for convolutional neural networks.

Final answer: B

\*\*\*\*\*

Answer Option 3:

The correct statement about Batch Normalization is B. The bias parameter of affine layers becomes redundant if a batch normalization layer follows immediately afterward. Batch Normalization normalizes the activations of a layer, making the bias term unnecessary.

Statement A is incorrect. Batch Normalization does not guarantee that the layer's activations will follow a standard Gaussian distribution.

Statement C is incorrect. The standard weight initialization does not need to be changed when using Batch Normalization.

Statement D is incorrect. Batch Normalization and Layer Normalization are different techniques, and they are not equivalent, especially for convolutional neural networks.

Final answer: B

\*\*\*\*\*

Answer Option 4:

The correct statement about Batch Normalization is B. The bias parameter of affine layers becomes redundant if a batch normalization layer follows immediately afterward. Batch Normalization normalizes the input to a layer by subtracting the batch mean and dividing by the batch standard deviation, which reduces the need for a bias parameter. The other statements are not true: A is incorrect because the activations after batch normalization do not necessarily follow a standard Gaussian distribution, C is incorrect because the standard weight initialization does not need to be changed when using Batch Normalization, and D is incorrect because Batch Normalization and Layer Normalization are different techniques. Final answer: B

\*\*\*\*\*

Answer Option 5:

The correct statement about Batch Normalization is B. The bias parameter of affine layers becomes redundant if a batch normalization layer follows immediately afterward. Batch Normalization eliminates the need for a bias term in the preceding layer as it can be absorbed into the batch normalization calculation. Final answer: B.

\*\*\*\*\*

Answer Option 6:

The statement that is true about Batch Normalization is B. The bias parameter of affine layers becomes redundant if a batch normalization layer follows immediately afterward. Batch Normalization has the ability to normalize the activations of a layer, which reduces the dependence of the model on the bias parameter. This allows the bias parameter to be omitted or set to zero without affecting the model's performance. Final answer: B

\*\*\*\*\*

Answer Option 7:

Option B is true about Batch Normalization. When a batch normalization layer is applied immediately after an affine layer, the bias parameter of the affine layer becomes redundant because it can be absorbed into the normalization process.

Option A is not true. Batch normalization does not guarantee that the layer's activations will follow a standard Gaussian distribution.

Option C is not true. The standard weight initialization does not need to be changed when using Batch Normalization.

Option D is not true. Batch Normalization and Layer Normalization are different techniques and are not equivalent for convolutional neural networks.

Final answer: B

\*\*\*\*\*

Answer Option 8:

Option B is true about Batch Normalization. The bias parameter of affine layers becomes redundant if a batch normalization layer follows immediately afterward. This is because batch normalization has its own mechanism to shift the mean of the layer's activations. Option A is not true because batch normalization does not guarantee that the layer's activations will follow a standard Gaussian distribution. Option C is not true because the standard weight initialization does not necessarily need to be changed when using Batch Normalization. Option D is not true because Batch Normalization and Layer Normalization are different techniques with different calculations and characteristics. Final answer: B.

\*\*\*\*\*



Answer Option 9:

Option B is true about Batch Normalization. Batch Normalization normalizes the output of a previous layer by subtracting the batch mean and dividing by the batch standard deviation. This makes the bias parameter of affine layers redundant because the normalization process takes care of the bias. Therefore, the bias parameter can be omitted or set to zero.

Option A is incorrect. Batch normalization does not guarantee that the layer's activations will follow a standard Gaussian distribution.

Option C is incorrect. The standard weight initialization does not need to be changed when using Batch Normalization.

Option D is incorrect. Batch Normalization and Layer Normalization are different techniques and are not equivalent for convolutional neural networks. Final answer: B

\*\*\*\*\*

Question 107:

'Suppose we have the following objective function:  $\frac{1}{2} \|Xw - y\|^2 + \frac{1}{2} \gamma \|w\|^2$ . What is the gradient of  $\frac{1}{2} \|Xw - y\|^2 + \frac{1}{2} \gamma \|w\|^2$  with respect to  $w$ ?', A.  $\nabla_w f(w) = (X^T X + \gamma I)w - X^T y + \gamma w$ , B.  $\nabla_w f(w) = X^T X w - X^T y + \gamma w$ , C.  $\nabla_w f(w) = X^T X w - X^T y + \gamma w$ , D.  $\nabla_w f(w) = X^T X w - X^T y + (\gamma + 1) w$

=====

Answer Option 1:

To compute the gradient of the objective function with respect to  $w$ , we can differentiate each term separately and sum them up. Let's break down the objective function step by step.

The first term  $\frac{1}{2} \|Xw - y\|^2$  can be expanded as  $\frac{1}{2} (Xw - y)^T (Xw - y)$ . Expanding further:

$$\frac{1}{2} (w^T X^T X w - w^T X^T y - y^T X w + y^T y)$$

The derivative of this term with respect to  $w$  is:

$$\frac{1}{2} (2X^T X w - X^T y - X^T y) = X^T X w - X^T y$$

The second term  $\frac{1}{2} \gamma \|w\|^2$  can be differentiated as follows:

$$\gamma w$$

Combining the derivatives of both terms, we get:

$$\nabla_w f(w) = X^T X w - X^T y + \gamma w$$

Comparing this with the answer choices, we can see that the correct option is B. Final answer: B

\*\*\*\*\*

Answer Option 2:

To find the gradient of the given objective function with respect to  $w$ , we can take the derivative of each term separately and then add them up.

Let's start with the first term:  $\frac{1}{2} \|Xw - y\|^2$ . By applying the chain rule, we have:

$$\nabla_w \left( \frac{1}{2} \|Xw - y\|^2 \right) = X^T (Xw - y)$$

Now, let's move on to the second term:  $\frac{1}{2} \gamma \|w\|^2$ . Taking the derivative with respect to  $w$ , we get:

$$\nabla_w \left( \frac{1}{2} \gamma \|w\|^2 \right) = \gamma w$$

Now, we can add the derivatives of the two terms together:

$$\nabla_w f(w) = X^T (Xw - y) + \gamma w$$

However, in the given options, we have  $\gamma I w$  instead of  $\gamma w$  in the second term. So, the correct answer is not among the given options. Final Answer: None of the given options.

\*\*\*\*\*

Answer Option 3:

To find the gradient of the given objective function, we can differentiate each term separately.

The gradient of  $\frac{1}{2} \|Xw - y\|^2$  with respect to  $w$  is given by:  $\nabla_w \left( \frac{1}{2} \|Xw - y\|^2 \right) = X^T (Xw - y)$ .

The gradient of  $\frac{1}{2} \gamma \|w\|^2$  with respect to  $w$  is given by:  $\nabla_w \left( \frac{1}{2} \gamma \|w\|^2 \right) = \gamma w$ .

Therefore, the overall gradient of the objective function is:

$$\nabla_w f(w) = \nabla_w \left( \frac{1}{2} \|Xw - y\|^2 \right) + \nabla_w \left( \frac{1}{2} \gamma \|w\|^2 \right) = X^T (Xw - y) + \gamma w$$

Final answer: C.  $\nabla_w f(w) = X^T X w - X^T y + \gamma w$ .

\*\*\*\*\*

Answer Option 4:

To find the gradient of the given objective function, we differentiate each term separately and apply the chain rule.

Using the chain rule, the gradient of  $\frac{1}{2} \|Xw - y\|^2$  with respect to  $w$  is:

$$\nabla_w \left( \frac{1}{2} \|Xw - y\|^2 \right) = (Xw - y)^T \nabla_w (Xw - y) = (Xw - y)^T X = X^T (Xw - y)$$

The gradient of  $\frac{1}{2} \gamma \|w\|^2$  with respect to  $w$  is:

$$\nabla_w \left( \frac{1}{2} \gamma \|w\|^2 \right) = \gamma w$$

Adding these two gradients together, we get:

$$\nabla_w f(w) = X^T (Xw - y) + \gamma w$$

Comparing this to the options, we can see that the correct answer is A:

$$\nabla_w f(w) = (X^T X + \gamma I)w - X^T y + \gamma w$$

\*\*\*\*\*

Answer Option 5:

To find the gradient of the given objective function, we can differentiate each term separately and sum them up.

Let's break down the objective function step by step:

$$\frac{1}{2} \|Xw - y\|^2 + \frac{1}{2} \gamma \|w\|^2$$

Differentiating the first term with respect to  $w$ :

$$\frac{\partial}{\partial w} \left( \frac{1}{2} \|Xw - y\|^2 \right) = X^T (Xw - y)$$

Differentiating the second term with respect to  $w$ :

$$\frac{\partial}{\partial w} \frac{1}{2} \gamma \|w\|_2^2 = \gamma w$$

Summing up the derivatives:

$$\nabla_w f(w) = X^T(Xw - y) + \gamma w$$

Simplifying the equation, we have:

$$\nabla_w f(w) = X^T X w - X^T y + \gamma w$$

Comparing this with the answer choices, we can see that the correct answer is A:

$$\nabla_w f(w) = (X^T X + \lambda I)w - X^T y + \lambda w$$

\*\*\*\*\*

Answer Option 6:

To find the gradient of the objective function with respect to  $w$ , we first need to compute the derivative of each term separately and then sum them up.

The first term  $\frac{1}{2} \|Xw - y\|_2^2$  can be expanded as  $\frac{1}{2} (Xw - y)^T (Xw - y)$ . Taking the derivative with respect to  $w$ , we get  $\nabla_w \frac{1}{2} \|Xw - y\|_2^2 = X^T (Xw - y)$ .

The second term  $\frac{1}{2} \gamma \|w\|_2^2$  can be expanded as  $\frac{1}{2} \gamma w^T w$ . Taking the derivative with respect to  $w$ , we get  $\nabla_w \frac{1}{2} \gamma \|w\|_2^2 = \gamma w$ .

Now, summing up the derivatives of both terms, we have  $\nabla_w f(w) = X^T (Xw - y) + \gamma w$ .

Comparing this with the given options, the correct answer is A.  $\nabla_w f(w) = (X^T X + \lambda I)w - X^T y + \lambda w$ .

Note that the given expression has  $\gamma$  replaced by  $\lambda$ , which is likely a typo. The correct expression should have  $\lambda$  instead of  $\gamma$ .

\*\*\*\*\*

Answer Option 7:

To find the gradient of the given objective function, we can differentiate each term separately and sum them up.

Let's compute the gradient step by step:

1. Gradient of the first term:  $\nabla_w \|Xw - y\|_2^2$

Using the chain rule, we have:

$$\nabla_w \|Xw - y\|_2^2 = 2(Xw - y)^T X$$

2. Gradient of the second term:  $\gamma \|w\|_2^2$

Using the chain rule and the fact that the derivative of  $\|w\|_2^2$  is  $2w$ , we have:

$$\nabla_w \|w\|_2^2 = 2w$$

Putting it all together, the gradient of the objective function is:

$$\nabla_w f(w) = \frac{1}{2} \nabla_w \|Xw - y\|_2^2 + \frac{1}{2} \gamma \nabla_w \|w\|_2^2 = (Xw - y)^T X + \gamma w$$

Expanding the first term, we have:

$$\nabla_w f(w) = X^T X w - X^T y + \gamma w$$

Comparing this with the options, we can see that the correct answer is A:  $\nabla_w f(w) = (X^T X + \gamma I)w - X^T y + \gamma w$ .

\*\*\*\*\*

Answer Option 8:

To find the gradient of the objective function, we can differentiate each term separately and sum them up.

First, we differentiate  $\frac{1}{2} \|Xw - y\|_2^2$  with respect to  $w$ . Using the chain rule, we have:

$$\nabla_w \left( \frac{1}{2} \|Xw - y\|_2^2 \right) = \nabla_w \left( \frac{1}{2} (Xw - y)^T (Xw - y) \right)$$

Expanding the expression, we get:

$$\nabla_w \left( \frac{1}{2} (w^T X^T X w - y^T X w - w^T X^T y + y^T y) \right)$$

Ignoring the constant term, we have:

$$\nabla_w \left( \frac{1}{2} (w^T X^T X w - y^T X w - w^T X^T y) \right)$$

Applying the product rule, we can differentiate each term separately:

$$\nabla_w \left( \frac{1}{2} w^T X^T X w \right) = X^T X w$$

$$\nabla_w \left( \frac{1}{2} -y^T X w \right) = -X^T y$$

$$\nabla_w \left( \frac{1}{2} -w^T X^T y \right) = -X^T y$$

Summing up the terms, we get:

$$\nabla_w \left( \frac{1}{2} \|Xw - y\|_2^2 \right) = X^T X w - X^T y$$

Next, we differentiate  $\frac{1}{2} \gamma \|w\|_2^2$  with respect to  $w$ . This is straightforward:

$$\nabla_w \left( \frac{1}{2} \gamma \|w\|_2^2 \right) = \gamma w$$

Summing up the gradients of both terms, we get the overall gradient:

$$\nabla_w f(w) = (X^T X w - X^T y) + \gamma w$$

Simplifying, we have:

$$\nabla_w f(w) = X^T X w - X^T y + \gamma w$$

Comparing this with the options, we can see that the correct answer is A:  $\nabla_w f(w) = (X^T X + \lambda I)w - X^T y + \lambda w$ . Therefore, the final answer is A.

\*\*\*\*\*

Answer Option 9:

To find the gradient of the objective function, we can take the derivative with respect to  $w$ .

Using the chain rule and the properties of the squared 2-norm, we have:

$$\nabla_w f(w) = \nabla_w \left( \frac{1}{2} \|Xw - y\|_2^2 + \frac{1}{2} \gamma \|w\|_2^2 \right)$$

$$= \nabla_w \left( \frac{1}{2} (Xw - y)^T (Xw - y) + \frac{1}{2} \gamma w^T w \right)$$

$$= \nabla_w \left( \frac{1}{2} (w^T X^T X w - 2X^T X y + y^T X w + y^T y) + \frac{1}{2} \gamma w^T w \right)$$

$$= \frac{1}{2} (2X^T X w - 2X^T X y) + \gamma w$$

$$= X^T X w - X^T X y + \gamma w$$

Therefore, the correct answer is A:  $\nabla_w f(w) = (X^T X + \gamma I)w - X^T X y + \gamma w$ .

\*\*\*\*\*

Question 108:

'Which of the following is true of a convolution kernel?'. A. 'Convolving an image with  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  would not change the image', B. 'Convolving an image with  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  would not change the image', C. 'Convolving an image with  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  would not change the image', D. 'Convolving an image with  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  would not change the image'

=====

Answer Option 1:

A convolution kernel is a small matrix used in convolutional neural networks for image processing tasks. When convolving an image with a kernel, the resulting output may or may not change the image.

In this case, option B is correct. Convolving an image with the kernel  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  would not change the image because it is an identity kernel, which means it preserves the original image. Final answer: B

\*\*\*\*\*

Answer Option 2:

Convolving an image with a kernel means applying the kernel to the image using the convolution operation. The resulting image will generally be different from the original image, unless the kernel is an identity kernel (all zeros except for a 1 in the center), such as the one in option A. Therefore, option A is true. Final answer: A

\*\*\*\*\*

Answer Option 3:

Convolving an image with a kernel will generally change the image. However, if the kernel is an identity kernel (e.g.,  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ), then convolving the image with this kernel would not change the image. Therefore, the correct answer is A.

\*\*\*\*\*

Answer Option 4:

Convolving an image with a kernel would typically result in changes to the image. Answer choices A, B, C, and D all suggest that convolving the image with the given kernels would not change the image. However, this is not true for any of the given kernels. The correct answer should state that convolving an image with a kernel would change the image. Therefore, none of the given answer choices are correct.

\*\*\*\*\*

Answer Option 5:

Convolving an image with a kernel matrix that consists of all zeros, like  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ , would not change the image. This is because the result of the convolution operation would be zero for every pixel in the output image. Final answer: D

\*\*\*\*\*

Answer Option 6:

Convolving an image with a kernel that contains all zeros, such as  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ , would result in a black image because the convolution operation would not change the values of the original image. Therefore, option D is true. Final answer: D

\*\*\*\*\*

Answer Option 7:

A convolution kernel is a matrix that is used for convolutional operations in image processing. When convolving an image with a kernel, the resulting image will generally be different from the original image, unless the kernel is an identity matrix.

Therefore, the correct statement is:

A. 'Convolving an image with  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  would not change the image'

\*\*\*\*\*

Answer Option 8:

A convolution kernel is a matrix used in convolutional neural networks for tasks such as image processing. When convolving an image with a kernel, the resulting image is typically different from the original image.

Among the given options, the only correct statement is B. Convolving an image with a kernel  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  would not change the image, as it represents a "Identity" kernel where the center value (1) corresponds to the pixel itself and the surrounding values (0) do not have any effect. Final answer: B

\*\*\*\*\*

Answer Option 9:

Convolving an image with a kernel refers to applying the kernel to the image using the convolution operation. The result of this operation is a transformed image.

Option A is incorrect because convolving an image with the kernel  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  would change the image.

Option B is correct because convolving an image with the kernel  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  would not change the image. This kernel is known as an identity kernel.

Option C is incorrect because convolving an image with the kernel  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  would change the image.

Option D is incorrect because convolving an image with the kernel  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  would result in a completely black image.

Final answer: B

\*\*\*\*\*

Question 109:

'Which of the following is false?', A. 'Semantic segmentation models predict the class of each pixel, while multiclass image classifiers predict the class of entire image.', B. 'A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive.', C. 'When a predicted bounding box does not correspond to any object in the scene, it is considered a false positive.', D. 'A bounding box with an IoU (intersection over union) equal to 3% would likely be considered a false negative.'

=====

Answer Option 1:

The false statement is B. A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive, not a false positive. Final answer: B

\*\*\*\*\*

Answer Option 2:

The false statement is B. A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive. A high IoU indicates a strong overlap between the predicted bounding box and the ground truth bounding box, which is typically indicative of a correct detection. Final answer: B

\*\*\*\*\*

Answer Option 3:

The false statement is B. A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive. A higher IoU indicates a better overlap between the predicted bounding box and the ground truth box, which is generally considered to be a true positive. Final answer: B.

\*\*\*\*\*

Answer Option 4:

The statement that is false is B. A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive. In reality, a bounding box with an IoU of 96% would be considered a high overlap and would indeed be considered a true positive in most cases. Final answer: B

\*\*\*\*\*

Answer Option 5:

The statement that is false is option B. A bounding box with an Intersection over Union (IoU) equal to 96% would likely be considered a true positive, not false positive. The higher the IoU value, the better the overlap between the predicted bounding box and the ground truth bounding box, indicating a more accurate detection. Final answer: B

\*\*\*\*\*

Answer Option 6:

The false statement is B. A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive. A high IoU indicates a strong overlap between the predicted bounding box and the ground truth bounding box, which is usually considered a correct detection. Final answer: B.

\*\*\*\*\*

Answer Option 7:

The false statement is B. A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive. In object detection tasks, IoU is used as a measure of how well the predicted bounding box aligns with the ground truth bounding box. A high IoU indicates a good match between the predicted and ground truth bounding boxes, which would classify it as a true positive. Final answer: B.

\*\*\*\*\*

Answer Option 8:

The false statement is:

B. 'A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive.'

In object detection tasks, a bounding box with an IoU (intersection over union) equal to or higher than a certain threshold (typically 50% or 0.5) is considered a true positive. A bounding box with an IoU of 96% would almost certainly be considered a true positive. Final answer: B

\*\*\*\*\*

Answer Option 9:

The statement that is false is: B. 'A bounding box with an IoU (intersection over union) equal to 96% would likely be considered a true positive.'

In object detection tasks, a bounding box with an IoU of 96% or higher is generally not considered a true positive. The threshold for considering a bounding box as a true positive may vary, but it is typically set at a higher IoU value, such as 50% or 70%. A high IoU indicates a significant overlap between the predicted bounding box and the ground truth bounding box, but a threshold is needed to account for any potential errors or inaccuracies in the predictions. Final answer: B.

\*\*\*\*\*

Question 110:

'Which of the following is false?', A. 'The following fully connected network without activation functions is linear:  $g_3(g_2(g_1(x)))$ , where  $g_i(x) = W_i x$  and  $W_i$  are matrices.', B. 'Leaky ReLU  $\max(0.01x, x)$  is convex.', C. 'A combination of ReLUs such as  $\text{ReLU}(x) - \text{ReLU}(x-1)$  is convex.', D. 'The loss  $\log \sigma(x) = -\log(1+e^{-x})$  is concave'

=====

Answer Option 1:

Option B is false. The leaky ReLU activation function  $\max(0.01x, x)$  is not convex. Final answer: B

=====

Answer Option 2:

The false statement is C. A combination of ReLUs such as  $\text{ReLU}(x) - \text{ReLU}(x-1)$  is not convex. Final answer: C.

=====

Answer Option 3:

The false statement is D. The loss function  $\log \sigma(x) = -\log(1+e^{-x})$  is convex, not concave. Final answer: D

=====

Answer Option 4:

The statement that is false is option D. The loss function  $\log \sigma(x) = -\log(1+e^{-x})$  is convex, not concave. Final answer: D.

=====

Answer Option 5:

The statement that is false is option D. The loss function  $\log \sigma(x) = -\log(1+e^{-x})$  is not concave. It is convex. Final answer: D

=====

Answer Option 6:

Option D is false. The loss function  $\log \sigma(x) = -\log(1+e^{-x})$  is not concave. In fact, it is a convex function. Final answer: D.

=====

Answer Option 7:

The false statement is: C. 'A combination of ReLUs such as  $\text{ReLU}(x) - \text{ReLU}(x-1)$  is convex.' A combination of ReLUs such as  $\text{ReLU}(x) - \text{ReLU}(x-1)$  is not convex because it is a piecewise linear function and not globally convex. Final answer: C.

=====

Answer Option 8:

Option C, 'A combination of ReLUs such as  $\text{ReLU}(x) - \text{ReLU}(x-1)$  is convex', is false. A combination of ReLUs is not necessarily convex because the ReLU activation function itself is not convex. The function  $\text{ReLU}(x) - \text{ReLU}(x-1)$  is not convex as it is not linear on each of its segments. Final answer: C.

=====

Answer Option 9:

To determine which statement is false, we will analyze each option:

- A. The statement is true. A fully connected network without activation functions is linear.
- B. The statement is true. Leaky ReLU is a piecewise linear function, and all piecewise linear functions are convex.
- C. The statement is false. A combination of ReLUs such as  $\text{ReLU}(x) - \text{ReLU}(x-1)$  is not convex because it is not a piecewise linear function.
- D. The statement is true. The loss function  $\log \sigma(x) = -\log(1+e^{-x})$  is concave.

Therefore, the false statement is C. Final Answer: C

=====

Question 111:

'We are training fully connected network with two hidden layers to predict housing prices. Inputs are \$100\$-dimensional, and have several features such as the number of square feet, the median family income, etc. The first hidden layer has \$1000\$ activations. The second hidden layer has \$10\$ activations. The output is a scalar representing the house price. Assuming a vanilla network with affine transformations and with no batch normalization and no learnable parameters in the activation function, how many parameters does this network have?', A. '111021', B. '110010', C. '111110', D. '110011'

=====

Answer Option 1:

To calculate the number of parameters in the fully connected network, we need to consider the connections between the layers.

The connections from the input layer to the first hidden layer have  $100 \times 1000 = 100000$  parameters.

The connections from the first hidden layer to the second hidden layer have  $1000 \times 10 = 10000$  parameters.

The connections from the second hidden layer to the output layer have  $10 \times 1 = 10$  parameters.

Therefore, the total number of parameters in the network is  $100000 + 10000 + 10 = 110010$ .

Final answer: B

\*\*\*\*\*

Answer Option 2:

To calculate the number of parameters in this fully connected network, we need to consider the connections between the layers.

The connections from the input layer to the first hidden layer will have  $100 \times 1000 = 100,000$  parameters (each of the \$1000\$ activations in the first hidden layer has \$100\$ connections from the input layer).

The connections between the first hidden layer and the second hidden layer will have  $1000 \times 10 = 10,000$  parameters.

The connections from the second hidden layer to the output layer will have  $10$  parameters (each of the \$10\$ activations in the second hidden layer has a connection to the output layer).

Therefore, the total number of parameters in this network is  $100,000 + 10,000 + 10 = 110,010$ .

Final answer: B.

\*\*\*\*\*

Answer Option 3:

To calculate the number of parameters in the fully connected network, we consider the connections between the layers.

The first hidden layer has 1000 activations, and each activation is connected to the 100 inputs. Therefore, the first hidden layer has  $1000 \times 100 = 100,000$  connections.

The second hidden layer has 10 activations, and each activation is connected to the 1000 activations from the first hidden layer. Therefore, the second hidden layer has  $10 \times 1000 = 10,000$  connections.

Finally, the output layer has 1 activation, and it is connected to the 10 activations from the second hidden layer. Therefore, the output layer has  $1 \times 10 = 10$  connections.

In total, the network has  $100,000 + 10,000 + 10 = 110,010$  parameters.

Final answer: B

\*\*\*\*\*

Answer Option 4:

To calculate the number of parameters in the fully connected network, we need to consider the connections between the layers.

From the input layer to the first hidden layer, there are \$100\$ input dimensions and \$1000\$ activations, resulting in  $100 \times 1000 = 100,000$  parameters.

From the first hidden layer to the second hidden layer, there are \$1000\$ activations in the first hidden layer and \$10\$ activations in the second hidden layer, resulting in  $1000 \times 10 = 10,000$  parameters.

From the second hidden layer to the output layer, there are \$10\$ activations in the second hidden layer and \$1\$ output, resulting in  $10 \times 1 = 10$  parameters.

Adding up all the parameters, we have  $100,000 + 10,000 + 10 = 110,010$  parameters.

Final answer: B. '110010'

\*\*\*\*\*

Answer Option 5:

For each layer, the number of parameters is calculated by multiplying the number of activations in the current layer by the number of activations in the previous layer and adding a bias term for each activation in the current layer.

The first hidden layer has 1000 activations, which means there are 1000 biases. Since the inputs are \$100\$-dimensional, the first hidden layer has  $1000 \times 100 + 1000 = 100100$  parameters.

The second hidden layer has 10 activations, which means there are 10 biases. The previous layer (the first hidden layer) has 1000 activations. Therefore, the second hidden layer has  $10 \times 1000 + 10 = 10010$  parameters.

The output layer is a scalar, so there is only one bias term. The previous layer (the second hidden layer) has 10 activations. Therefore, the output layer has  $10 \times 1 + 1 = 11$  parameters.

In total, the network has  $100100 + 10010 + 11 = 110121$  parameters.

Final answer: None of the given options.

\*\*\*\*\*

Answer Option 6:

To calculate the number of parameters in the network, we need to consider the connections between layers.

The fully connected network has the following layers:

- Input layer: \$100\$ dimensions



- First hidden layer:  $10000$  activations
- Second hidden layer:  $10$  activations
- Output layer:  $1$  scalar

The number of parameters between the input layer and the first hidden layer is given by  $(\text{input dimensions}) \times (\text{number of activations in the first hidden layer})$ , which is  $100 \times 1000 = 100000$ .

The number of parameters between the first hidden layer and the second hidden layer is given by  $(\text{number of activations in the first hidden layer}) \times (\text{number of activations in the second hidden layer})$ , which is  $1000 \times 10 = 10000$ .

The number of parameters between the second hidden layer and the output layer is given by  $(\text{number of activations in the second hidden layer}) \times (\text{number of activations in the output layer})$ , which is  $10 \times 1 = 10$ .

Therefore, the total number of parameters in the network is  $100000 + 10000 + 10 = 110010$ .

Final answer: B

\*\*\*\*\*

Answer Option 7:

To calculate the number of parameters in this fully connected network, we need to consider the connections between the layers.

The connections between the input layer and the first hidden layer have  $100 \times 1000$  parameters.

The connections between the first hidden layer and the second hidden layer have  $1000 \times 10$  parameters.

The connections between the second hidden layer and the output layer have  $10 \times 1$  parameters.

Therefore, the total number of parameters in this network is  $100 \times 1000 + 1000 \times 10 + 10 \times 1 = 100000 + 10000 + 10 = 110010$ .

Final answer: A

\*\*\*\*\*

Answer Option 8:

To calculate the number of parameters in the network, we need to consider the connections between the layers.

The connection between the input layer and the first hidden layer will have  $100 \times 1000$  parameters.

The connection between the first hidden layer and the second hidden layer will have  $1000 \times 10$  parameters.

The connection between the second hidden layer and the output layer will have  $10 \times 1$  parameters.

In total, the network will have  $100 \times 1000 + 1000 \times 10 + 10 \times 1 = 100,000 + 10,000 + 10 = 110,010$  parameters.

Final answer: B

\*\*\*\*\*

Answer Option 9:

To calculate the number of parameters in the network, we need to consider the weights and biases at each layer.

For the first hidden layer, we have 1000 activations. Since each activation is connected to each input feature, there are  $1000 \times 100 = 100000$  weights, and  $10000$  biases. So, the first hidden layer has  $100000 + 1000 = 101000$  parameters.

For the second hidden layer, we have 10 activations. Each activation is connected to each activation in the previous layer, so there are  $10 \times 1000 = 10000$  weights, and  $1000$  biases. Therefore, the second hidden layer has  $10000 + 10 = 10010$  parameters.

Finally, we have the output layer which is a scalar representing the house price. Since it is connected to each activation in the previous layer, we have  $10$  weights and  $1$  bias. So, the output layer has  $10 + 1 = 11$  parameters.

Adding up the parameters from all layers, we get:  $101000 + 10010 + 11 = 111021$ .

Therefore, the correct answer is A. 111021.

\*\*\*\*\*

Question 112:

'Statement 1] The derivative of the sigmoid  $\sigma(x) = (1 + e^{-x})^{-1}$  with respect to  $x$  is equal to  $\text{Var}(B)$  where  $B \sim \text{Bern}(\sigma(x))$  is a Bernoulli random variable. Statement 2] Setting the bias parameters in each layer of neural network to 0 changes the bias-variance trade-off such that the model's variance increases and the model's bias decreases', A. 'True, True', B. 'False, False', C. 'True, False', D. 'False, True'

=====

Answer Option 1:

Statement 1: The derivative of the sigmoid function with respect to  $x$  is not equal to the variance of a Bernoulli random variable. Therefore, Statement 1 is False.

Statement 2: Setting the bias parameters in each layer of a neural network to 0 does not change the bias-variance trade-off. However, it will change the model's bias and potentially increase the model's bias.

Therefore, Statement 2 is False.

Final answer: B

=====

Answer Option 2:

Statement 1 is false. The derivative of the sigmoid function  $\sigma(x)$  with respect to  $x$  is equal to  $\sigma(x)(1 - \sigma(x))$ , not  $\text{Var}(B)$ .

Statement 2 is false. Setting the bias parameters in each layer of a neural network to 0 does not necessarily change the bias-variance trade-off. It can affect the model's bias and variance, but the specific effect depends on the data and the complexity of the model. Final answer: B

=====

Answer Option 3:

Statement 1 is false. The derivative of the sigmoid function with respect to  $x$  is actually equal to  $\sigma(x)(1 - \sigma(x))$ . The variance of a Bernoulli random variable is equal to  $\sigma(x)(1 - \sigma(x))$ , which is not the same as the derivative of the sigmoid function.

Statement 2 is also false. Setting the bias parameters in each layer of a neural network to 0 does not necessarily change the bias-variance trade-off. The bias-variance trade-off refers to the relationship between the model's bias (error due to overly simplistic assumptions) and variance (error due to sensitivity to small fluctuations in the training data). Changing the bias parameters to 0 may reduce the bias component of the trade-off, but it does not necessarily increase the model's variance.

Final Answer: B. 'False, False'

=====

Answer Option 4:

Statement 1 is false. The derivative of the sigmoid function with respect to  $x$  is  $\sigma(x)(1 - \sigma(x))$ .

Statement 2 is true. Setting the bias parameters in each layer of a neural network to 0 will reduce the bias of the model but potentially increase its variance.

Final answer: C

=====

Answer Option 5:

Statement 1 is false. The derivative of the sigmoid function with respect to  $x$  is not equal to the variance of a Bernoulli random variable.

Statement 2 is false. Setting the bias parameters in each layer of a neural network to 0 does not necessarily change the bias-variance trade-off. It may affect the bias and variance of the model, but the trade-off is more complex and depends on various factors.

Final answer: B

=====

Answer Option 6:

Statement 1 is False. The derivative of the sigmoid function with respect to  $x$  is  $\sigma(x)(1 - \sigma(x))$ , which is not equal to the variance of a Bernoulli random variable.

Statement 2 is False. Setting the bias parameters in each layer of a neural network to 0 does not necessarily change the bias-variance trade-off. It simply removes the bias term from the model, which can affect the model's performance in different ways depending on the specific problem and dataset.

Final Answer: B

=====

Answer Option 7:

Statement 1 is false. The derivative of the sigmoid function with respect to  $x$  is  $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ , not  $\text{Var}(B)$ .

Statement 2 is also false. Setting the bias parameters in each layer of a neural network to 0 does not necessarily change the bias-variance trade-off. It only removes the bias term from the model, potentially increasing the model's bias without affecting its variance.

Final answer: B

=====

Answer Option 8:

Statement 1 is incorrect. The derivative of the sigmoid function  $\sigma(x)$  with respect to  $x$  is actually equal to  $\sigma(x)(1 - \sigma(x))$ , and it is not equal to the variance of a Bernoulli random variable.

Statement 2 is also incorrect. Setting the bias parameters in each layer of a neural network to 0 does not necessarily change the bias-variance trade-off. It simply removes the bias term from the model, which can impact the model's performance in various ways depending on the specific problem and dataset.

Final answer: D. 'False, True'

=====

Answer Option 9:

Statement 1: The derivative of the sigmoid function with respect to  $x$  is  $\sigma(x)(1-\sigma(x))$ . However,  $\text{Var}(B)$ , where  $B \sim \text{Bern}(\sigma(x))$ , is the variance of a Bernoulli random variable  $B$  and is not equal to the derivative of the sigmoid function. Therefore, statement 1 is False.

Statement 2: Setting the bias parameters in each layer of a neural network to 0 does not necessarily change the bias-variance trade-off. The bias-variance trade-off refers to the trade-off between the model's ability to fit the training data (bias) and its ability to generalize to unseen data (variance). Changing the bias parameters to 0 may affect the model's performance, but it does not guarantee a specific change in the bias-variance trade-off. Therefore, statement 2 is False.

Final answer: D. False, True.

\*\*\*\*\*