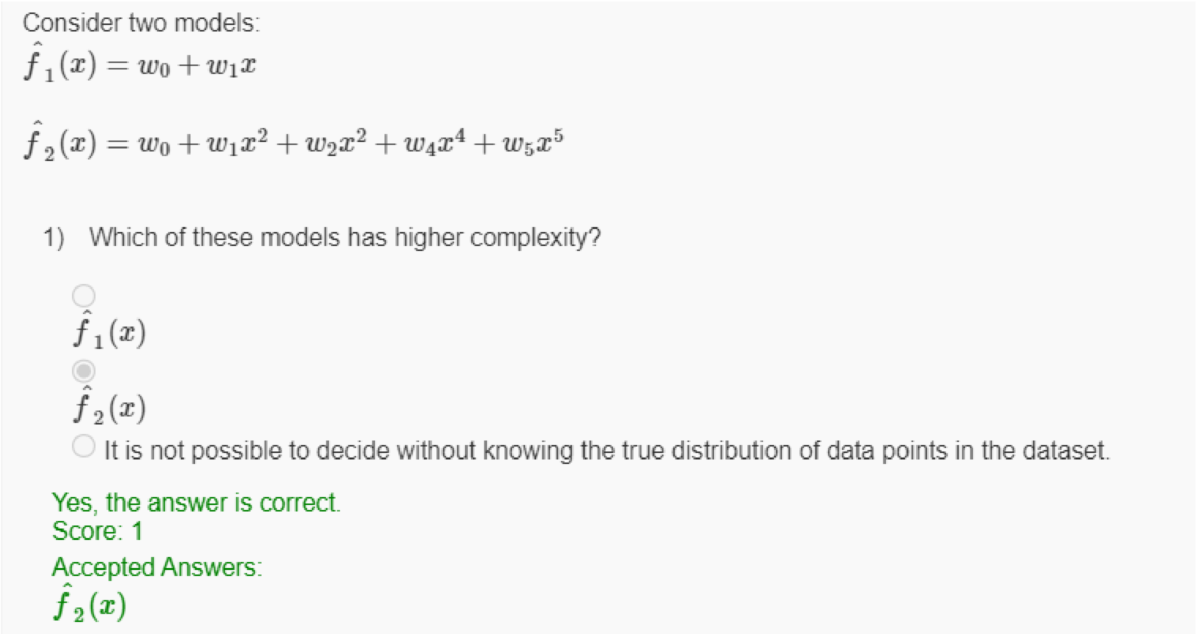


Consider two models:

$\hat{f}_1(x) = w_0 + w_1 x$

$\hat{f}_2(x) = w_0 + w_1 x^2 + w_2 x^2 + w_4 x^4 + w_5 x^5$

1) Which of these models has higher complexity?

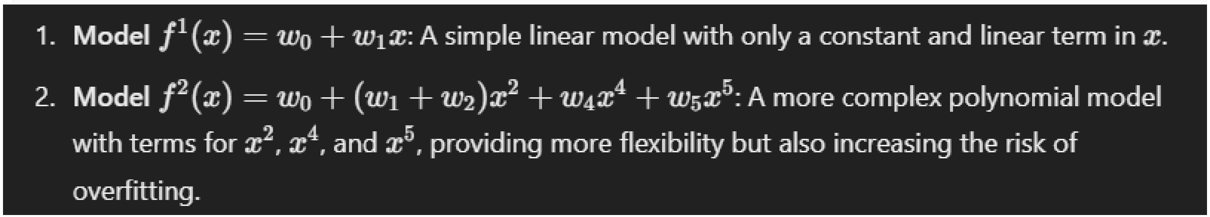
○ $\hat{f}_1(x)$

● $\hat{f}_2(x)$

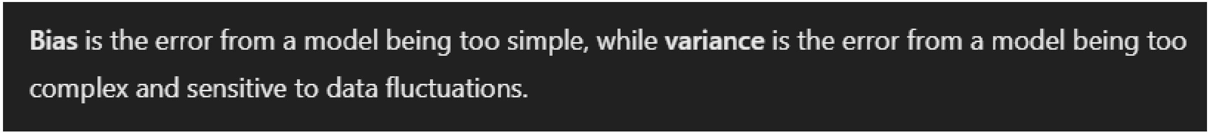
○ It is not possible to decide without knowing the true distribution of data points in the dataset.

Yes, the answer is correct.
Score: 1

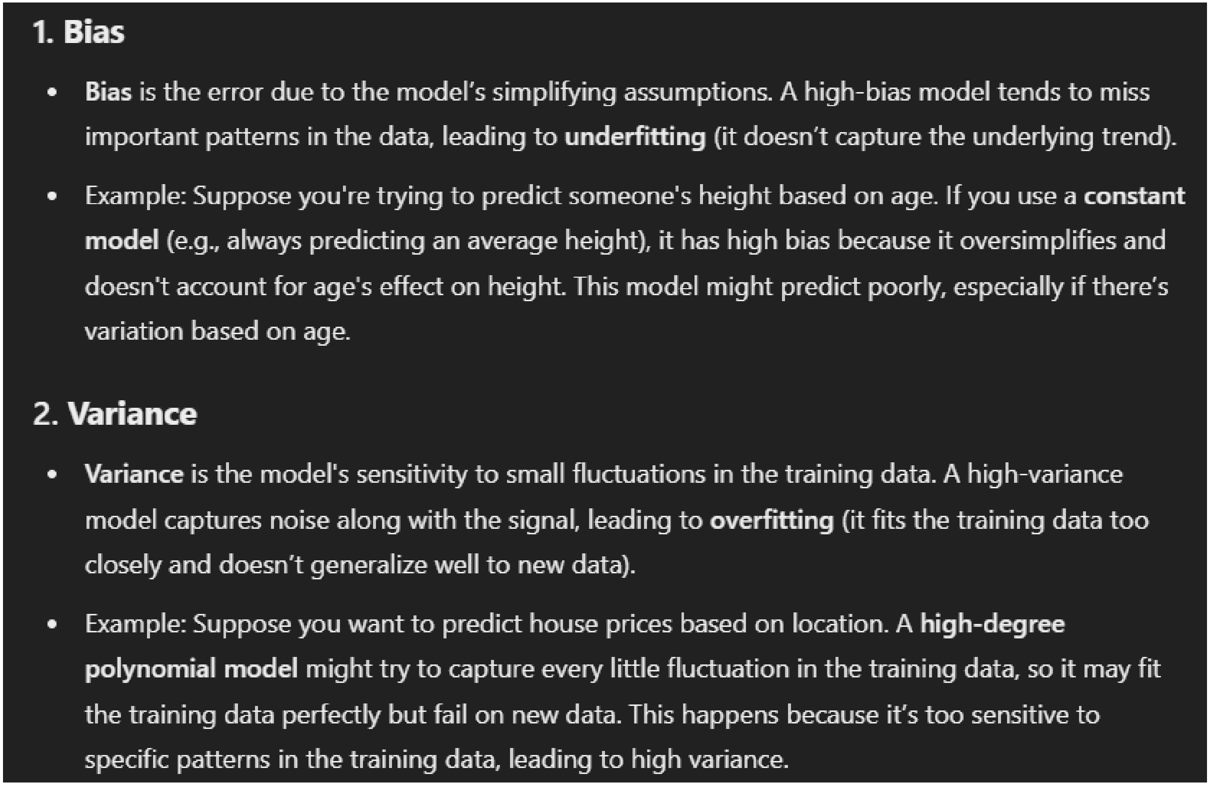
Accepted Answers:
$\hat{f}_2(x)$



1. **Model $f^1(x) = w_0 + w_1 x$**: A simple linear model with only a constant and linear term in x.
2. **Model $f^2(x) = w_0 + (w_1 + w_2)x^2 + w_4 x^4 + w_5 x^5$**: A more complex polynomial model with terms for x^2, x^4, and x^5, providing more flexibility but also increasing the risk of overfitting.



Bias is the error from a model being too simple, while **variance** is the error from a model being too complex and sensitive to data fluctuations.

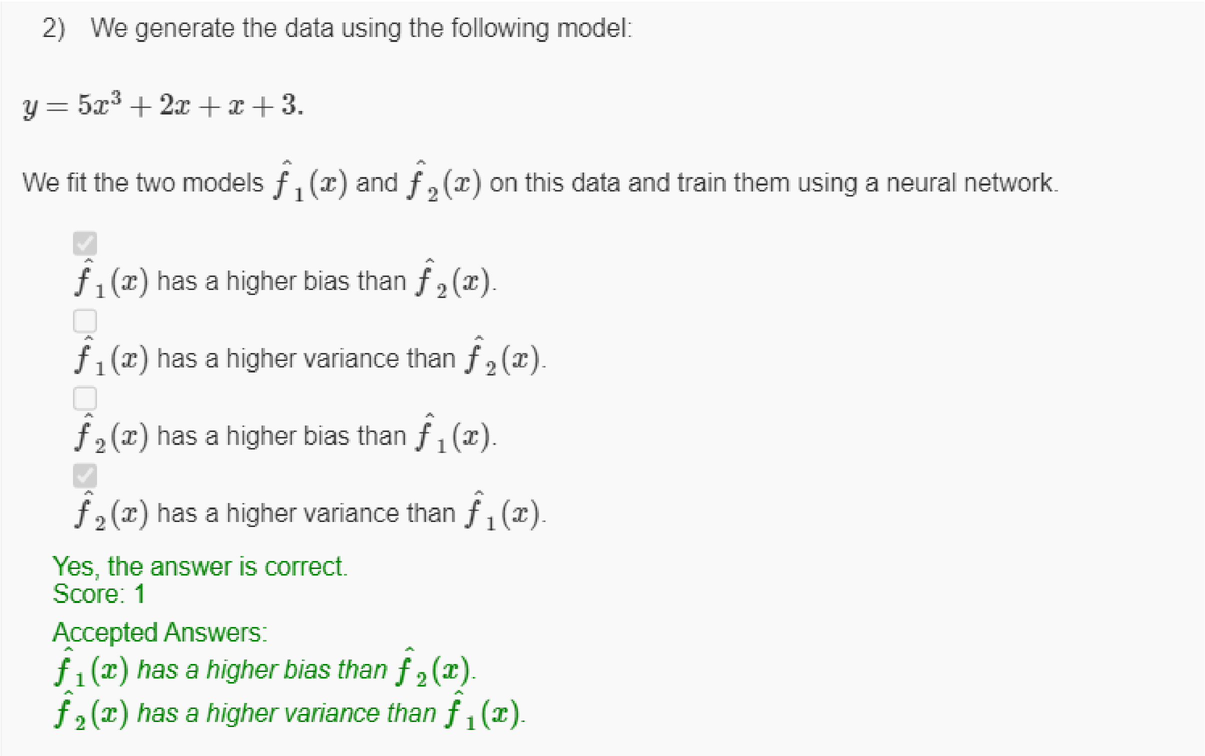


1. Bias

- **Bias** is the error due to the model’s simplifying assumptions. A high-bias model tends to miss important patterns in the data, leading to **underfitting** (it doesn’t capture the underlying trend).
- Example: Suppose you're trying to predict someone's height based on age. If you use a **constant model** (e.g., always predicting an average height), it has high bias because it oversimplifies and doesn't account for age's effect on height. This model might predict poorly, especially if there’s variation based on age.

2. Variance

- **Variance** is the model's sensitivity to small fluctuations in the training data. A high-variance model captures noise along with the signal, leading to **overfitting** (it fits the training data too closely and doesn’t generalize well to new data).
- Example: Suppose you want to predict house prices based on location. A **high-degree polynomial model** might try to capture every little fluctuation in the training data, so it may fit the training data perfectly but fail on new data. This happens because it’s too sensitive to specific patterns in the training data, leading to high variance.



2) We generate the data using the following model:

$y = 5x^3 + 2x + x + 3.$

We fit the two models $\hat{f}_1(x)$ and $\hat{f}_2(x)$ on this data and train them using a neural network.

☑ $\hat{f}_1(x)$ has a higher bias than $\hat{f}_2(x)$.

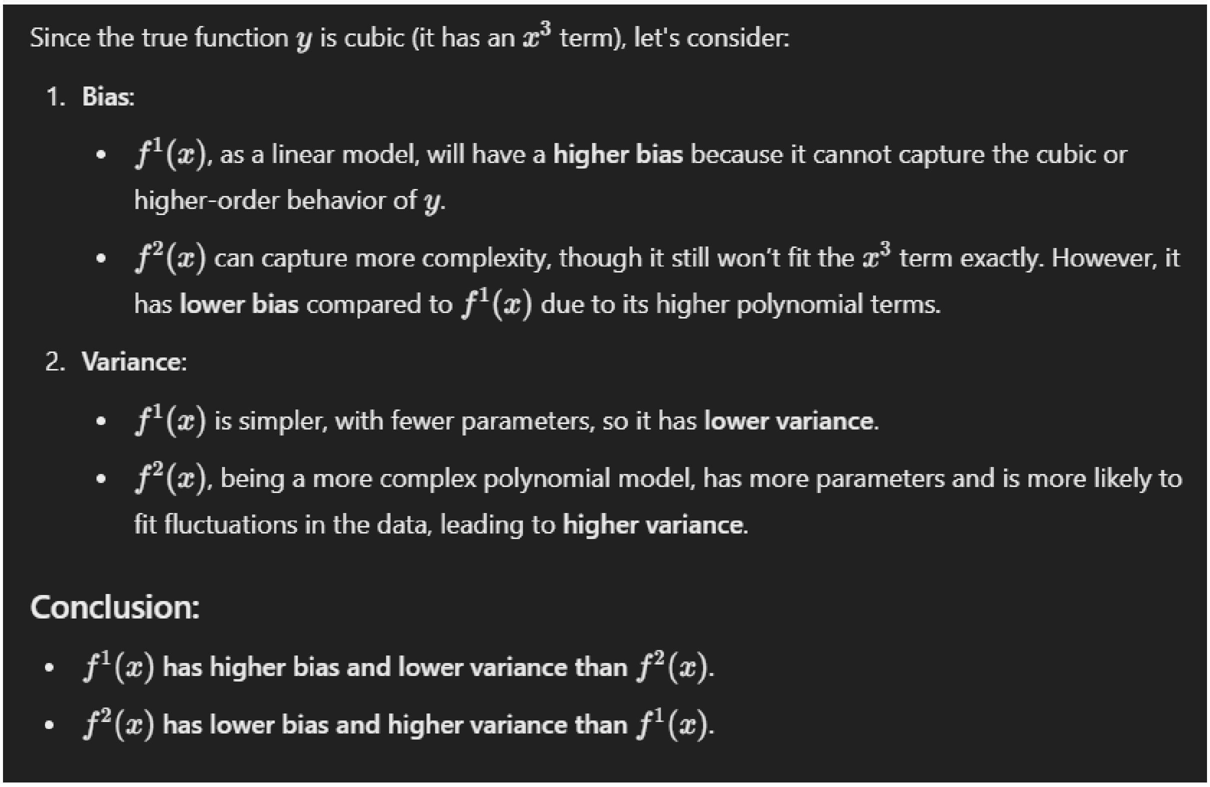
☐ $\hat{f}_1(x)$ has a higher variance than $\hat{f}_2(x)$.

☐ $\hat{f}_2(x)$ has a higher bias than $\hat{f}_1(x)$.

☑ $\hat{f}_2(x)$ has a higher variance than $\hat{f}_1(x)$.

Yes, the answer is correct.
Score: 1

Accepted Answers:
$\hat{f}_1(x)$ has a higher bias than $\hat{f}_2(x)$.
$\hat{f}_2(x)$ has a higher variance than $\hat{f}_1(x)$.



Since the true function y is cubic (it has an x^3 term), let's consider:

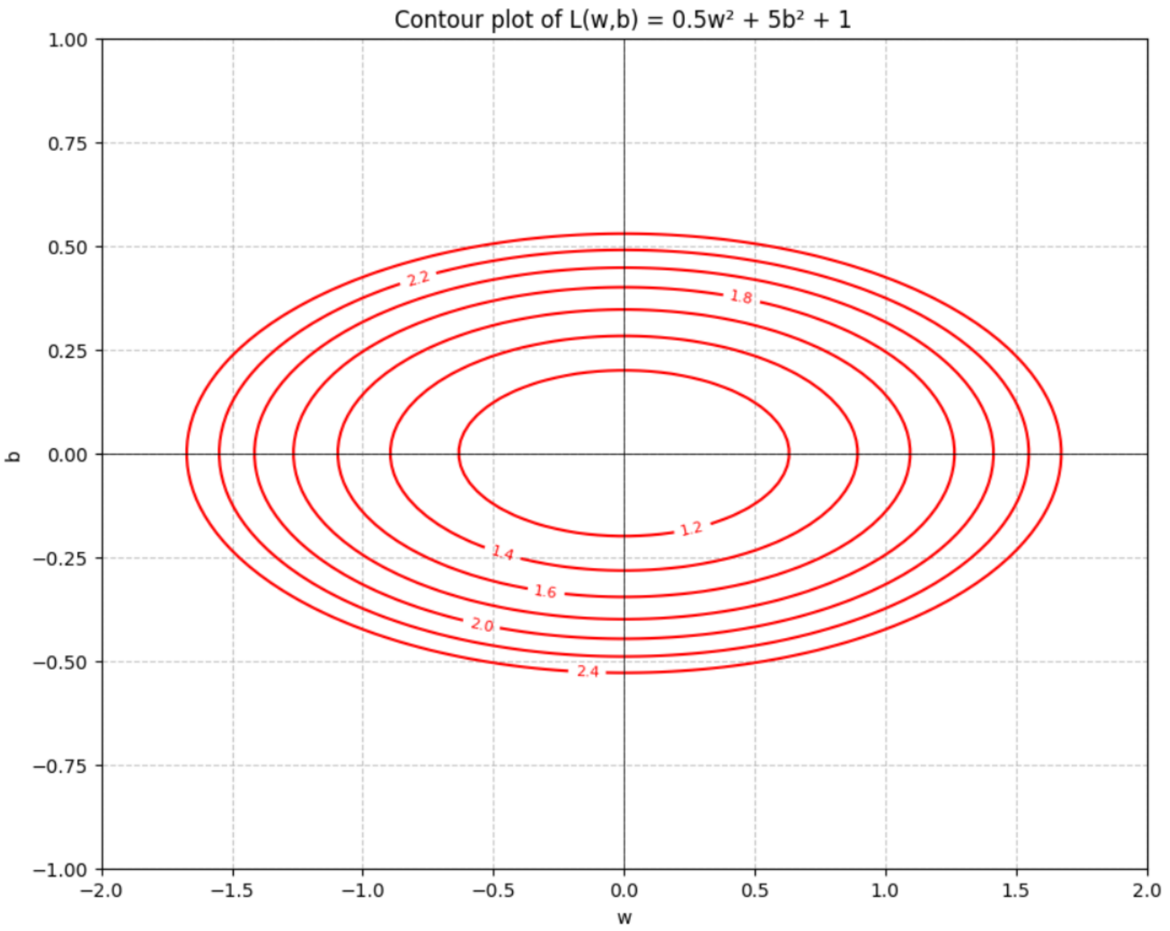
1. **Bias:**
   - $f^1(x)$, as a linear model, will have a **higher bias** because it cannot capture the cubic or higher-order behavior of y.
   - $f^2(x)$ can capture more complexity, though it still won't fit the x^3 term exactly. However, it has **lower bias** compared to $f^1(x)$ due to its higher polynomial terms.
2. **Variance:**
   - $f^1(x)$ is simpler, with fewer parameters, so it has **lower variance.**
   - $f^2(x)$, being a more complex polynomial model, has more parameters and is more likely to fit fluctuations in the data, leading to **higher variance.**

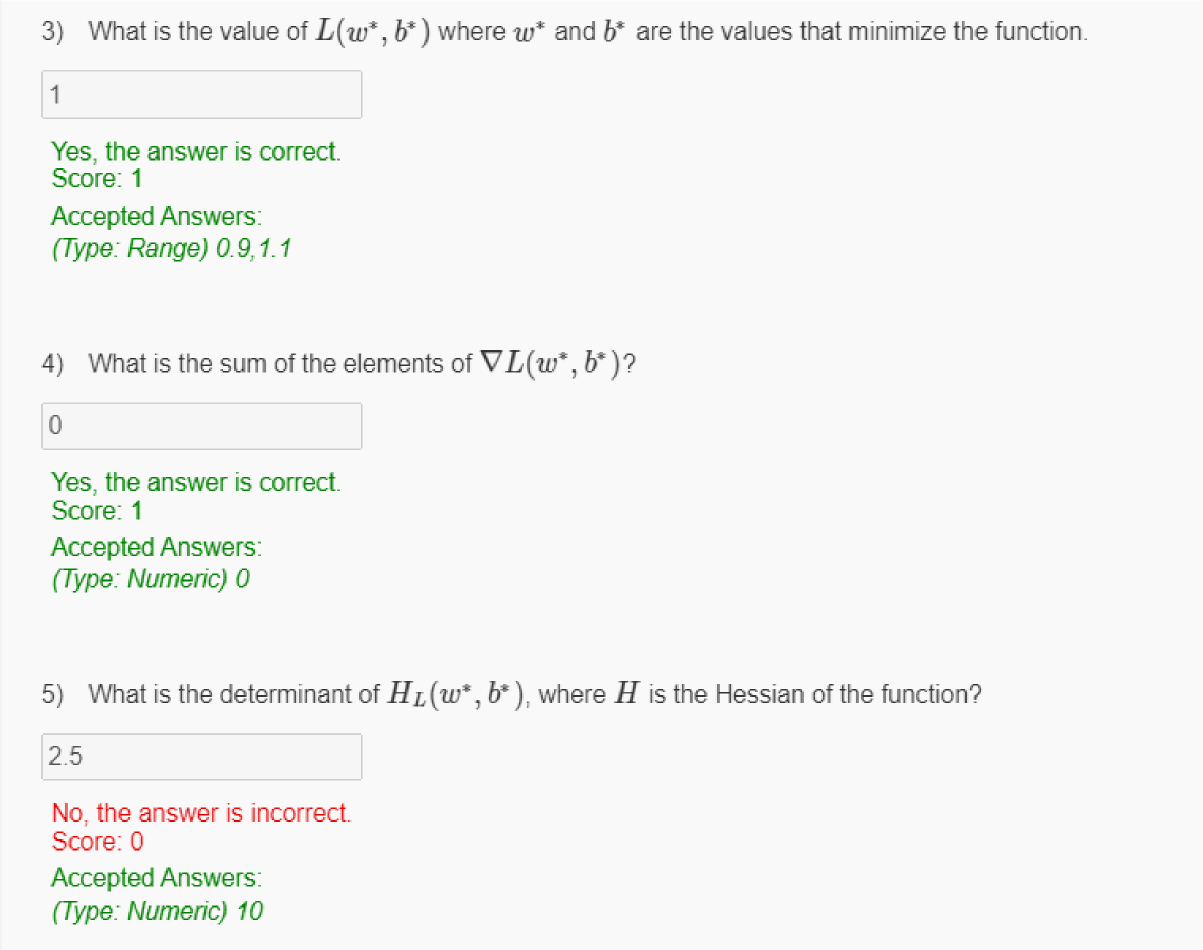
Conclusion:

- **$f^1(x)$ has higher bias and lower variance than $f^2(x)$.**
- **$f^2(x)$ has lower bias and higher variance than $f^1(x)$.**

Common Data Q3-Q6

Consider a function $L(w, b) = 0.5w^2 + 5b^2 + 1$ and its contour plot given below:





3) What is the value of $L(w^*, b^*)$ where w^* and b^* are the values that minimize the function.

1

Yes, the answer is correct.
Score: 1

Accepted Answers:
(Type: Range) 0.9,1.1

4) What is the sum of the elements of $\nabla L(w^*, b^*)$?

0

Yes, the answer is correct.
Score: 1

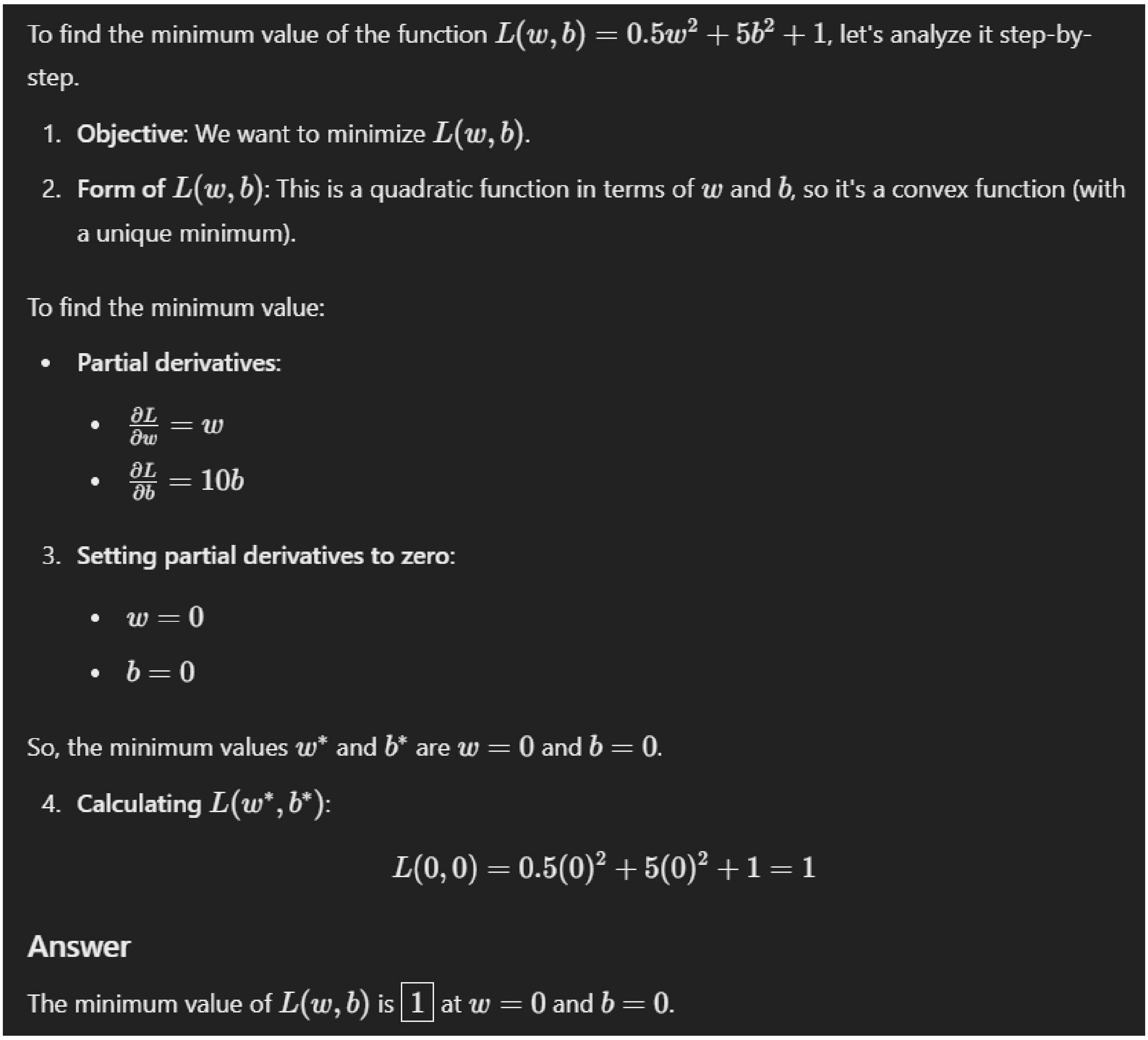
Accepted Answers:
(Type: Numeric) 0

5) What is the determinant of $H_L(w^*, b^*)$, where H is the Hessian of the function?

2.5

No, the answer is incorrect.
Score: 0

Accepted Answers:
(Type: Numeric) 10



To find the minimum value of the function $L(w, b) = 0.5w^2 + 5b^2 + 1$, let's analyze it step-by-step.

1. **Objective**: We want to minimize $L(w, b)$.
2. **Form of $L(w, b)$**: This is a quadratic function in terms of w and b, so it's a convex function (with a unique minimum).

To find the minimum value:

- **Partial derivatives:**
  - $\frac{\partial L}{\partial w} = w$
  - $\frac{\partial L}{\partial b} = 10b$

3. **Setting partial derivatives to zero**:
   - $w = 0$
   - $b = 0$

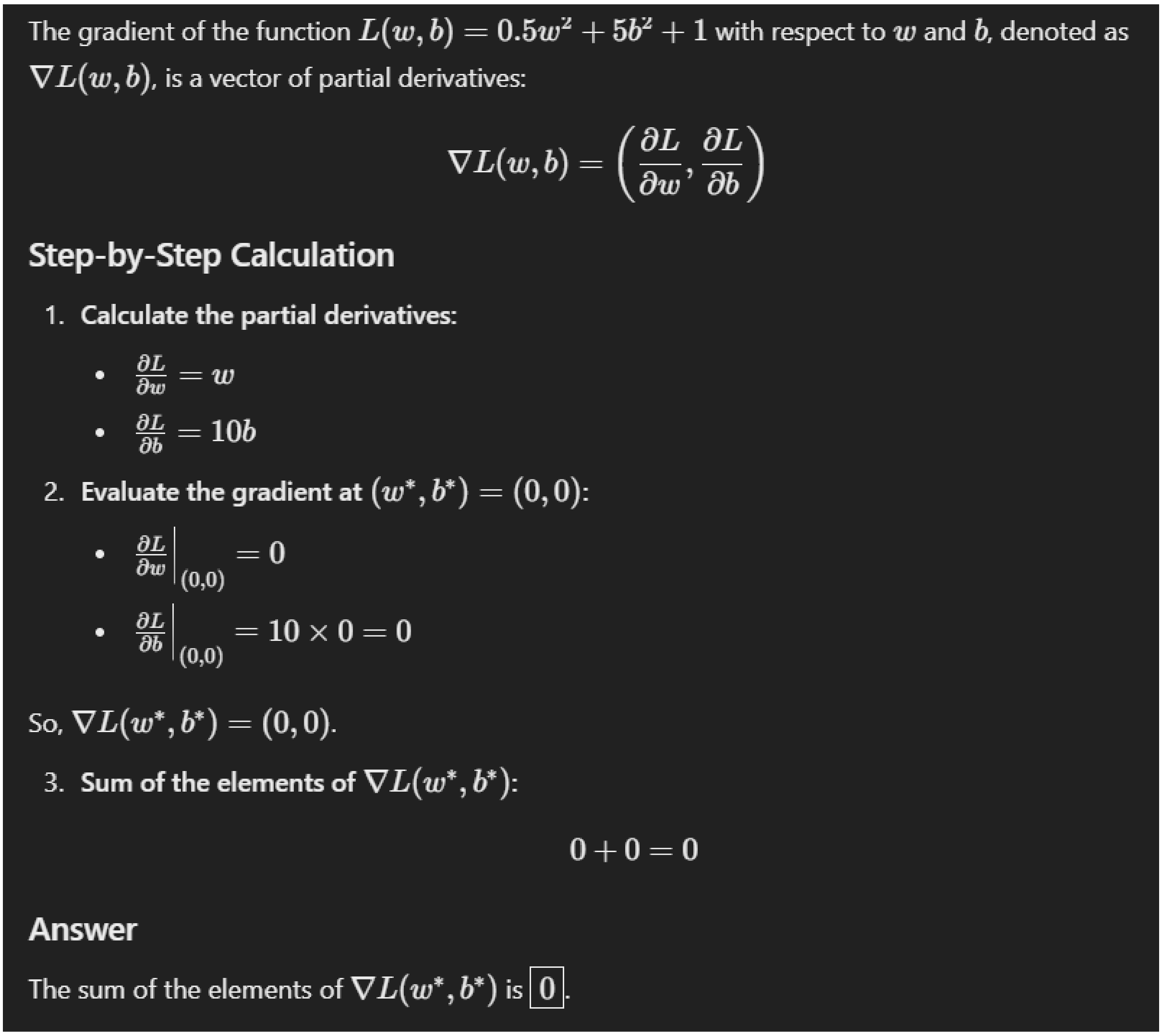
So, the minimum values w^* and b^* are $w = 0$ and $b = 0$.

4. **Calculating $L(w^*, b^*)$**:

$$L(0, 0) = 0.5(0)^2 + 5(0)^2 + 1 = 1$$

Answer

The minimum value of $L(w, b)$ is $\boxed{1}$ at $w = 0$ and $b = 0$.



The gradient of the function $L(w, b) = 0.5w^2 + 5b^2 + 1$ with respect to w and b, denoted as $\nabla L(w, b)$, is a vector of partial derivatives:

$$\nabla L(w, b) = \left(\frac{\partial L}{\partial w}, \frac{\partial L}{\partial b} \right)$$

Step-by-Step Calculation

1. **Calculate the partial derivatives:**
    - $\frac{\partial L}{\partial w} = w$
    - $\frac{\partial L}{\partial b} = 10b$
2. **Evaluate the gradient at $(w^*, b^*) = (0, 0)$:**
    - $\frac{\partial L}{\partial w}\Big|_{(0,0)} = 0$
    - $\frac{\partial L}{\partial b}\Big|_{(0,0)} = 10 \times 0 = 0$

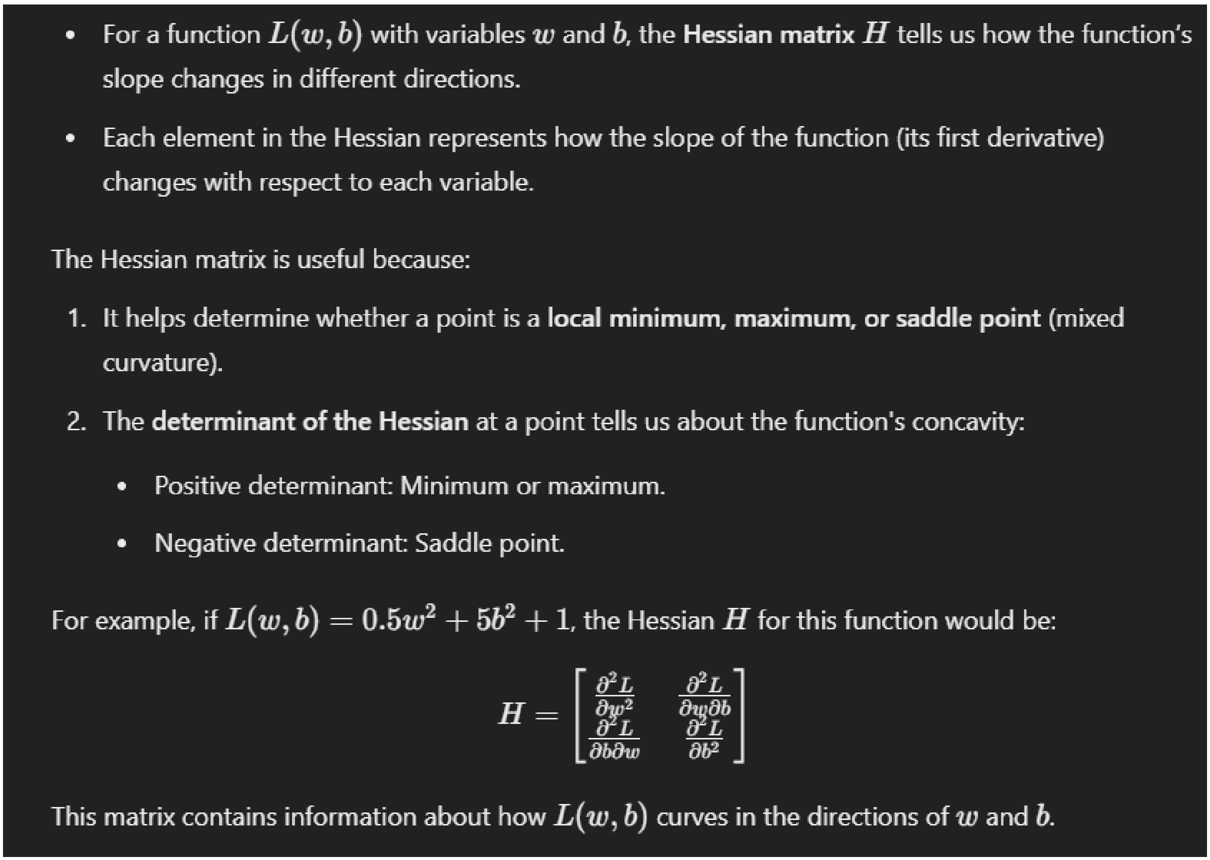
So, $\nabla L(w^*, b^*) = (0, 0)$.

3. **Sum of the elements of $\nabla L(w^*, b^*)$:**

$$0 + 0 = 0$$

Answer

The sum of the elements of $\nabla L(w^*, b^*)$ is $\boxed{0}$.



- For a function $L(w, b)$ with variables w and b, the **Hessian matrix H** tells us how the function's slope changes in different directions.
- Each element in the Hessian represents how the slope of the function (its first derivative) changes with respect to each variable.

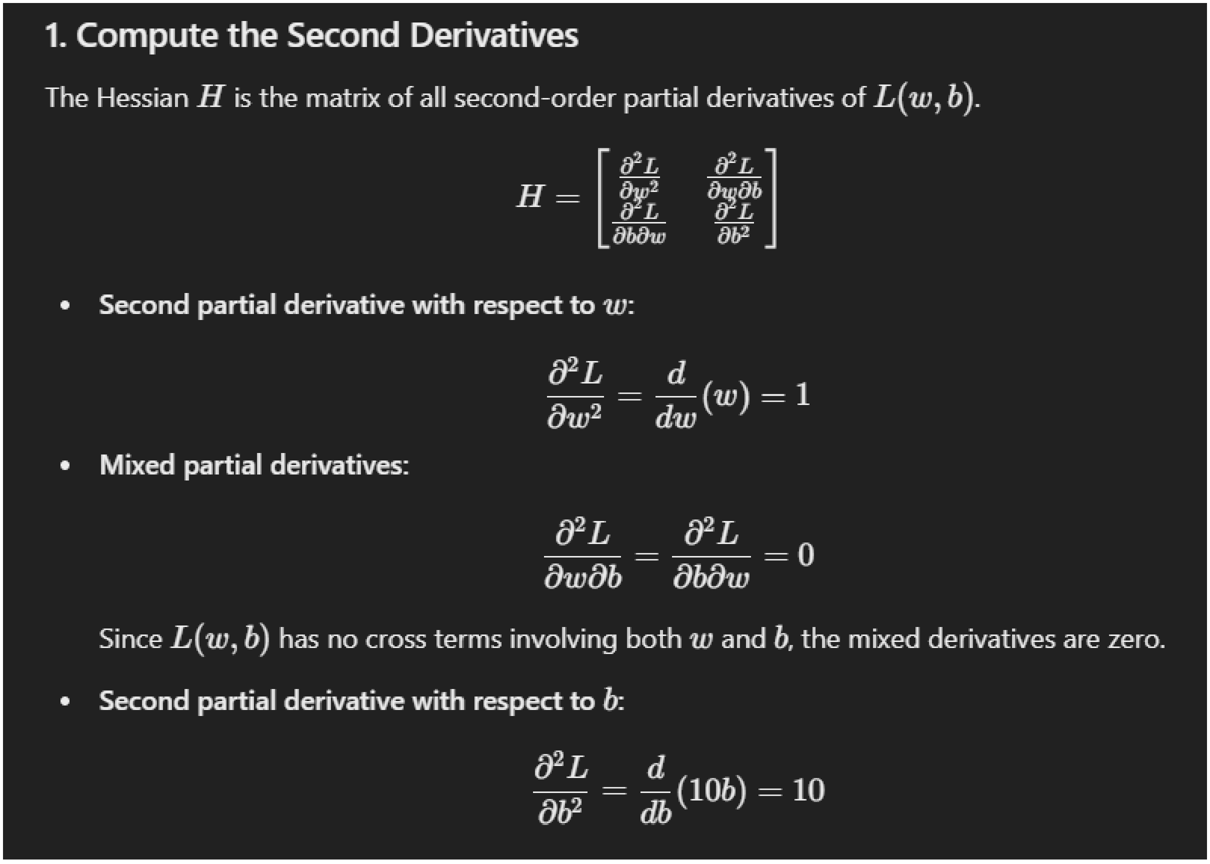
The Hessian matrix is useful because:

1. It helps determine whether a point is a **local minimum, maximum, or saddle point** (mixed curvature).
2. The **determinant of the Hessian** at a point tells us about the function's concavity:
   - Positive determinant: Minimum or maximum.
   - Negative determinant: Saddle point.

For example, if $L(w, b) = 0.5w^2 + 5b^2 + 1$, the Hessian H for this function would be:

$$H = \begin{bmatrix} \frac{\partial^2 L}{\partial w^2} & \frac{\partial^2 L}{\partial w \partial b} \\ \frac{\partial^2 L}{\partial b \partial w} & \frac{\partial^2 L}{\partial b^2} \end{bmatrix}$$

This matrix contains information about how $L(w, b)$ curves in the directions of w and b.



1. Compute the Second Derivatives

The Hessian H is the matrix of all second-order partial derivatives of $L(w, b)$.

$$H = \begin{bmatrix} \frac{\partial^2 L}{\partial w^2} & \frac{\partial^2 L}{\partial w \partial b} \\ \frac{\partial^2 L}{\partial b \partial w} & \frac{\partial^2 L}{\partial b^2} \end{bmatrix}$$

- **Second partial derivative with respect to w:**

$$\frac{\partial^2 L}{\partial w^2} = \frac{d}{dw}(w) = 1$$

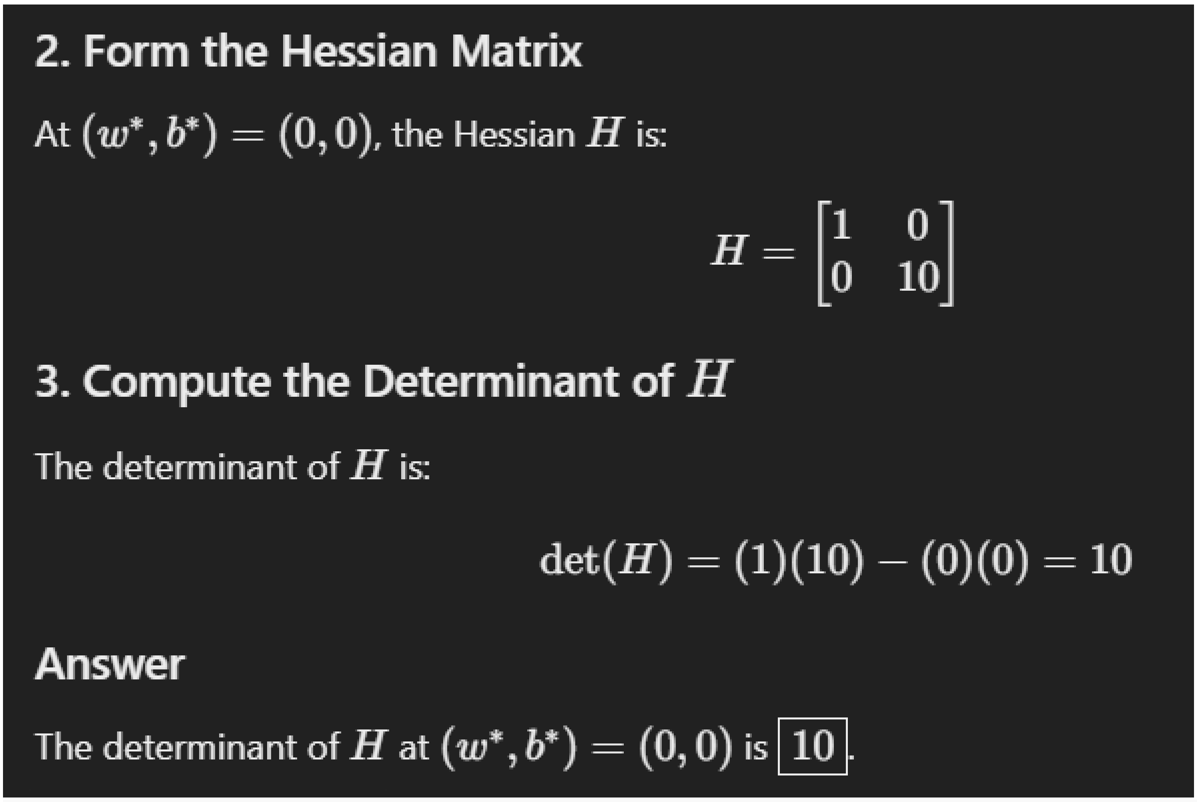
- **Mixed partial derivatives:**

$$\frac{\partial^2 L}{\partial w \partial b} = \frac{\partial^2 L}{\partial b \partial w} = 0$$

  Since $L(w, b)$ has no cross terms involving both w and b, the mixed derivatives are zero.

- **Second partial derivative with respect to b:**

$$\frac{\partial^2 L}{\partial b^2} = \frac{d}{db}(10b) = 10$$



2. Form the Hessian Matrix

At $(w^*, b^*) = (0, 0)$, the Hessian H is:

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$$

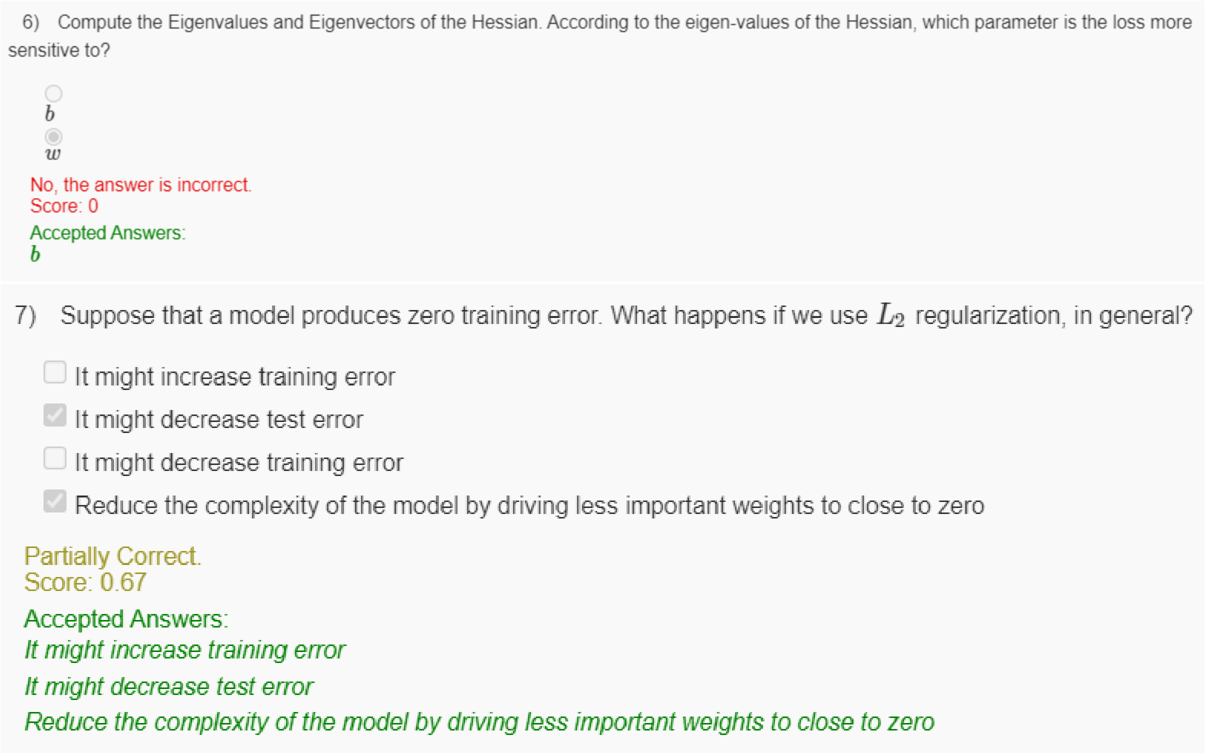
3. Compute the Determinant of H

The determinant of H is:

$$\det(H) = (1)(10) - (0)(0) = 10$$

Answer

The determinant of H at $(w^*, b^*) = (0, 0)$ is $\boxed{10}$.



6) Compute the Eigenvalues and Eigenvectors of the Hessian. According to the eigen-values of the Hessian, which parameter is the loss more sensitive to?

○ b

● w

No, the answer is incorrect.
Score: 0

Accepted Answers:
b

7) Suppose that a model produces zero training error. What happens if we use L_2 regularization, in general?

☐ It might increase training error

☑ It might decrease test error

☐ It might decrease training error

☑ Reduce the complexity of the model by driving less important weights to close to zero

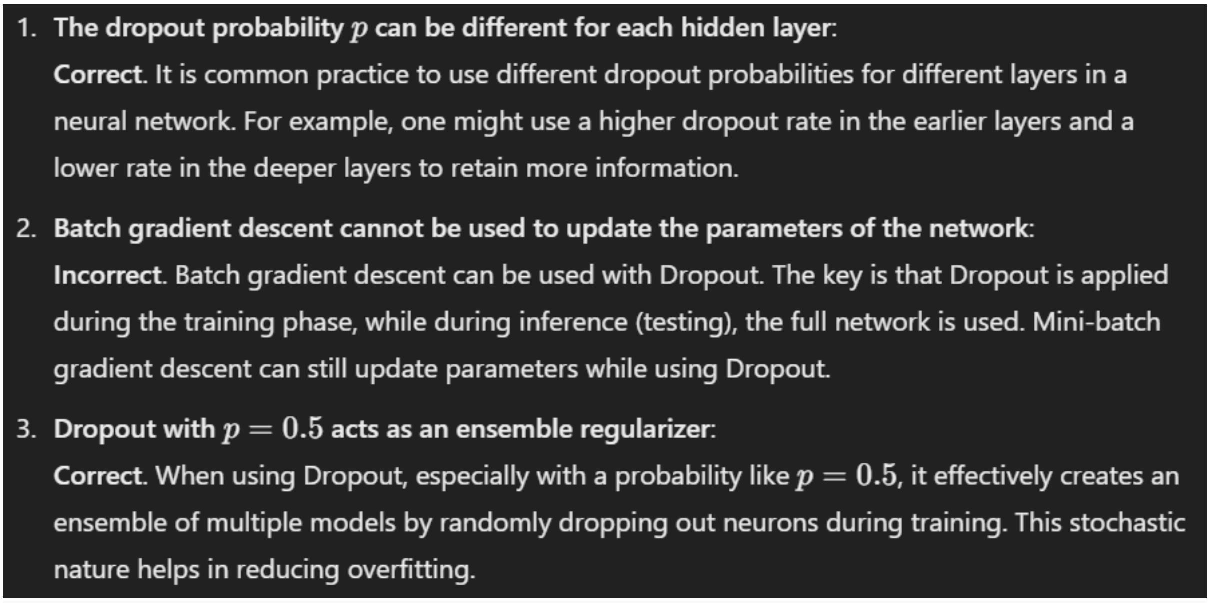
Partially Correct.
Score: 0.67

Accepted Answers:

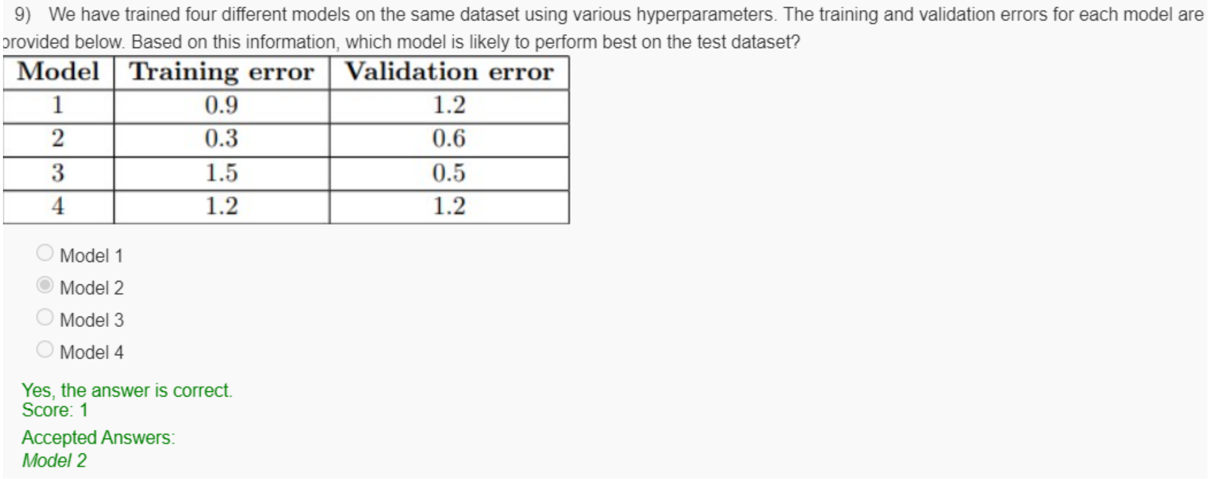
It might increase training error

It might decrease test error

Reduce the complexity of the model by driving less important weights to close to zero



1. **The dropout probability p can be different for each hidden layer**:
   Correct. It is common practice to use different dropout probabilities for different layers in a neural network. For example, one might use a higher dropout rate in the earlier layers and a lower rate in the deeper layers to retain more information.
2. **Batch gradient descent cannot be used to update the parameters of the network**:
   Incorrect. Batch gradient descent can be used with Dropout. The key is that Dropout is applied during the training phase, while during inference (testing), the full network is used. Mini-batch gradient descent can still update parameters while using Dropout.
3. **Dropout with $p = 0.5$ acts as an ensemble regularizer**:
   Correct. When using Dropout, especially with a probability like $p = 0.5$, it effectively creates an ensemble of multiple models by randomly dropping out neurons during training. This stochastic nature helps in reducing overfitting.



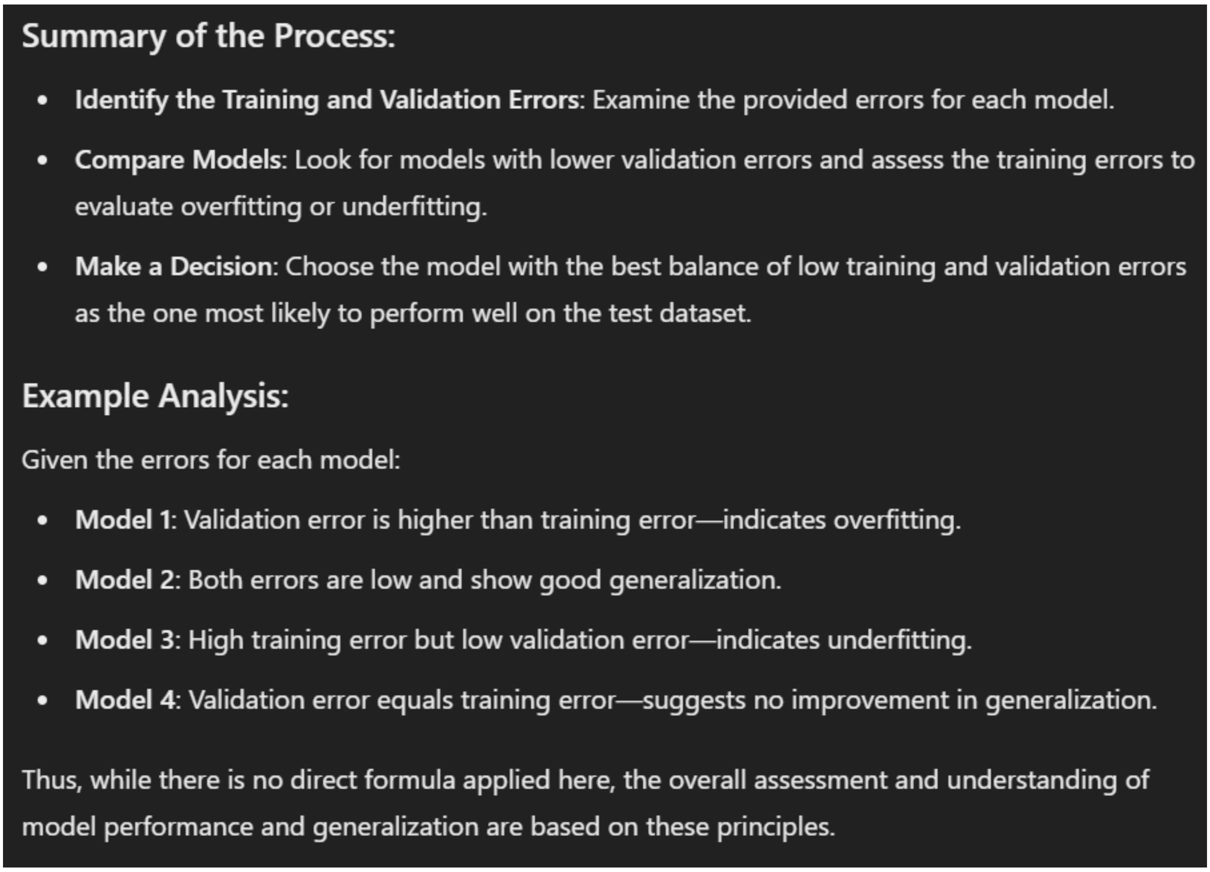
9) We have trained four different models on the same dataset using various hyperparameters. The training and validation errors for each model are provided below. Based on this information, which model is likely to perform best on the test dataset?

Model	Training error	Validation error
1	0.9	1.2
2	0.3	0.6
3	1.5	0.5
4	1.2	1.2

○ Model 1
● Model 2
○ Model 3
○ Model 4

Yes, the answer is correct.
Score: 1

Accepted Answers:
Model 2



Summary of the Process:

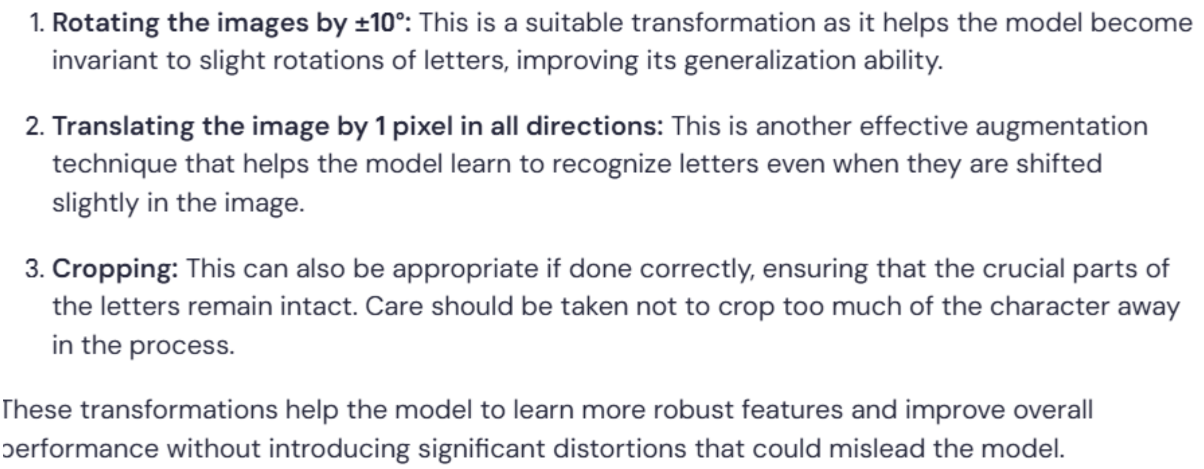
- **Identify the Training and Validation Errors**: Examine the provided errors for each model.
- **Compare Models**: Look for models with lower validation errors and assess the training errors to evaluate overfitting or underfitting.
- **Make a Decision**: Choose the model with the best balance of low training and validation errors as the one most likely to perform well on the test dataset.

Example Analysis:

Given the errors for each model:

- **Model 1**: Validation error is higher than training error—indicates overfitting.
- **Model 2**: Both errors are low and show good generalization.
- **Model 3**: High training error but low validation error—indicates underfitting.
- **Model 4**: Validation error equals training error—suggests no improvement in generalization.

Thus, while there is no direct formula applied here, the overall assessment and understanding of model performance and generalization are based on these principles.



1. **Rotating the images by ±10°:** This is a suitable transformation as it helps the model become invariant to slight rotations of letters, improving its generalization ability.
2. **Translating the image by 1 pixel in all directions:** This is another effective augmentation technique that helps the model learn to recognize letters even when they are shifted slightly in the image.
3. **Cropping:** This can also be appropriate if done correctly, ensuring that the crucial parts of the letters remain intact. Care should be taken not to crop too much of the character away in the process.

These transformations help the model to learn more robust features and improve overall performance without introducing significant distortions that could mislead the model.

Which of the following statements is true about the bias-variance tradeoff in deep learning?

Options:

1. Increasing the learning rate reduces bias
2. Increasing the learning rate reduces variance
3. Decreasing the learning rate reduces bias
4. None of These

Answer:

- **None of These.**

Explanation:

1. **Increasing the learning rate reduces bias:** While a higher learning rate can make the training process faster, it does not inherently reduce bias. Bias is related to the model's ability to capture the underlying patterns in the data; a model can still be biased regardless of the learning rate.

2. **Increasing the learning rate reduces variance:** Increasing the learning rate can lead to overshooting during training, potentially resulting in a higher variance rather than a reduction in it. A high learning rate can cause a model to be less consistent across different training runs and more prone to fluctuations in the learned patterns.

3. **Decreasing the learning rate reduces bias:** A lower learning rate might allow a model to train more thoroughly and potentially learn more complex patterns. However, decreasing the learning rate itself does not directly reduce bias; it may help improve model training but does not guarantee a decrease in bias.

In summary, none of the provided options correctly capture the relationship between learning rate adjustments and the bias-variance tradeoff in deep learning. Thus, the correct answer is **None of These.**

Which of the following statements is true about the bias-variance tradeoff in deep learning?

Options:

1. Increasing the size of the training dataset reduces bias
2. Increasing the size of the training dataset reduces variance
3. Decreasing the size of the training dataset reduces bias
4. Decreasing the size of the training dataset reduces variance

Answer:

- **Increasing the size of the training dataset reduces variance.**

Explanation:

1. **Increasing the size of the training dataset reduces bias:** While a larger dataset can help in better estimating the underlying true function, it does

not guarantee a reduction in bias by itself. Bias is primarily influenced by model complexity and assumptions, not just the size of the dataset.

2. **Increasing the size of the training dataset reduces variance:** This statement is true. By providing more data, the model can learn more generalized features and reduce its sensitivity to noise present in the training dataset, which helps to lower variance. A larger dataset helps to average out individual anomalies or fluctuations.

3. **Decreasing the size of the training dataset reduces bias:** This statement is false. Reducing the training dataset size generally increases bias because the model may not be able to capture the underlying patterns accurately due to insufficient data.

4. **Decreasing the size of the training dataset reduces variance:** This statement is also false. Decreasing the dataset size can lead to higher variability in model predictions, as the model might become overfitted to the smaller set of examples, capturing noise instead of the true underlying distribution.

In summary, the most accurate statement regarding the bias-variance tradeoff is that increasing the size of the training dataset reduces variance, which improves the overall generalization of the model. Thus, the correct answer is **Increasing the size of the training dataset reduces variance.**

Question:

What is the effect of high bias on a model's performance?

Options:

1. The model will overfit the training data.
2. The model will underfit the training data.
3. The model will be unable to learn anything from the training data.
4. The model's performance will be unaffected by bias.

Answer:

- **The model will underfit the training data.**

Explanation:

High bias occurs when a model is too simplistic to capture the underlying patterns of the data, leading to oversimplification. This causes the model to perform poorly on both the training set and the test set, resulting in underfitting.

It fails to learn from the training data adequately and cannot generalize well to new instances, leading to high errors in both cases. Therefore, high bias primarily leads to underfitting.

Second Question:

Question:

What is the usual relationship between train error and test error?

Options:

1. Train error is usually higher than test error
2. Train error is usually lower than test error
3. Train error and test error are usually the same
4. Train error and test error are unrelated

Answer:

- **Train error is usually lower than test error.**

Explanation:

In machine learning, the training error represents how well the model performs on the training dataset, while the test error indicates how well the model generalizes to unseen data. Typically, the training error is lower than the test error because the model is optimized to perform well on the data it was trained on, potentially capturing noise and specifics of that dataset. Consequently, when evaluated on a separate test set, the model often shows a higher error rate due to its inability to generalize effectively outside the training data. This observation highlights the common issue of overfitting

Question:

What is overfitting in deep learning?

Options:

1. When the model performs well on the training data but poorly on new, unseen data

2. When the model performs poorly on the training data and on new, unseen data
3. When the model has a high test error and a low train error
4. When the model has a low test error and a high train error

Answer:

- **When the model performs well on the training data but poorly on new, unseen data.**

Explanation:

Overfitting occurs when a model learns the training data too well, capturing noise and details that do not generalize to unseen data. As a result, while it may have a very low training error, its ability to predict accurately on new data is compromised, resulting in high test error. This is a key characteristic of overfitting.

Second Question:

Question:

How can overfitting be prevented in deep learning?

Options:

1. By increasing the complexity of the model
2. By decreasing the size of the training data
3. By adding more layers to the model
4. By using regularization techniques such as dropout

Answer:

- **By using regularization techniques such as dropout.**

Explanation:

Regularization techniques, such as dropout, L1 regularization, and L2 regularization, help to penalize complexity in models, encouraging them to generalize better to unseen data rather than merely memorizing the training data. Dropout, in particular, randomly sets a portion of the neurons to zero during training, which helps in reducing reliance on any single path within the network and thus combats overfitting. Increasing model complexity or adding more layers typically exacerbates overfitting.

Third Question:

Question:

Which of the following statements is true about L2 regularization?

Options:

1. It adds a penalty term to the loss function that is proportional to the absolute value of the weights.
2. It adds a penalty term to the loss function that is proportional to the square of the weights.
3. It gives us sparse solutions for w.
4. It is equivalent to adding Gaussian noise to the weights.

Answer:

- **It adds a penalty term to the loss function that is proportional to the square of the weights.**

Explanation:

L2 regularization (also known as weight decay) incorporates a penalty term to the loss function that is proportional to the square of the weights ($\|\mathrm{w}\|2\|w\|_2$). This effectively discourages large weights, promoting smaller weight values across the model's parameters. It does not inherently produce sparse solutions; that property is more associated with L1 regularization. Hence, the correct statement is that L2 regularization adds a penalty that is proportional to the square of the weights.