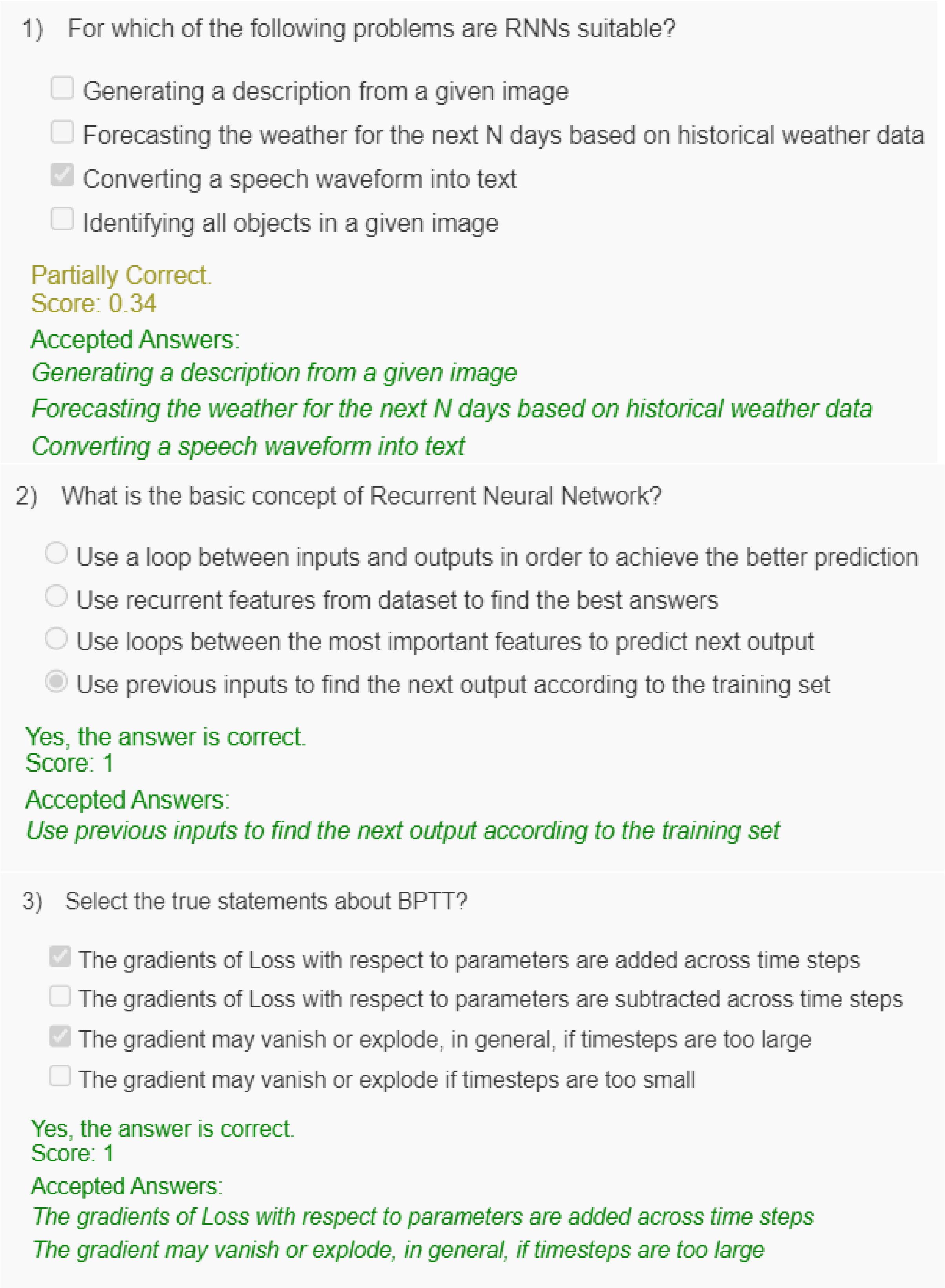


1) For which of the following problems are RNNs suitable?

☐ Generating a description from a given image

☐ Forecasting the weather for the next N days based on historical weather data

☑ Converting a speech waveform into text

☐ Identifying all objects in a given image

Partially Correct.
Score: 0.34

Accepted Answers:

Generating a description from a given image

Forecasting the weather for the next N days based on historical weather data

Converting a speech waveform into text

2) What is the basic concept of Recurrent Neural Network?

○ Use a loop between inputs and outputs in order to achieve the better prediction

○ Use recurrent features from dataset to find the best answers

○ Use loops between the most important features to predict next output

● Use previous inputs to find the next output according to the training set

Yes, the answer is correct.
Score: 1

Accepted Answers:

Use previous inputs to find the next output according to the training set

3) Select the true statements about BPTT?

☑ The gradients of Loss with respect to parameters are added across time steps

☐ The gradients of Loss with respect to parameters are subtracted across time steps

☑ The gradient may vanish or explode, in general, if timesteps are too large

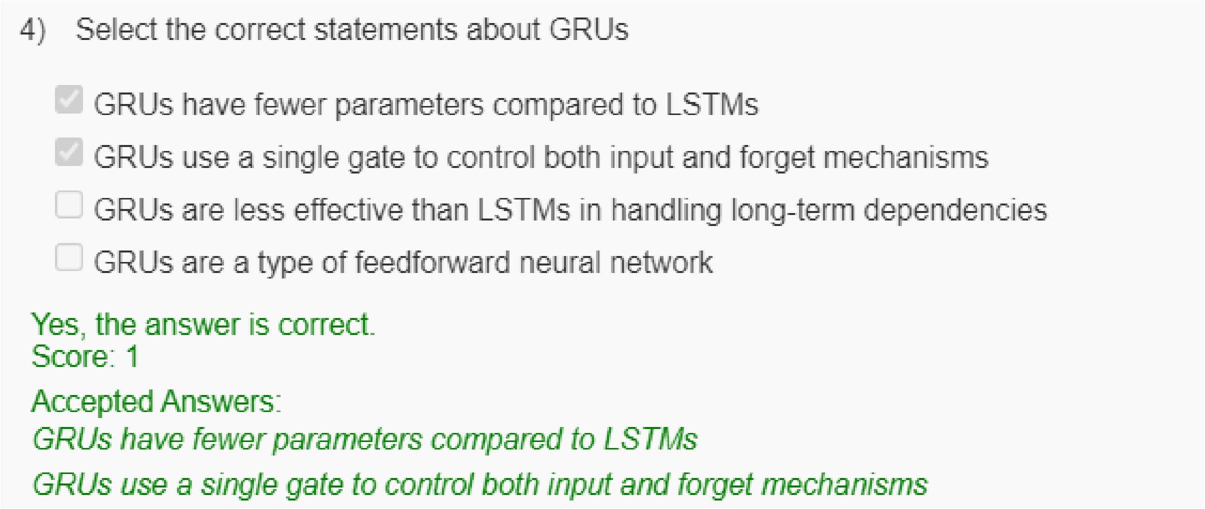
☐ The gradient may vanish or explode if timesteps are too small

Yes, the answer is correct.
Score: 1

Accepted Answers:

The gradients of Loss with respect to parameters are added across time steps

The gradient may vanish or explode, in general, if timesteps are too large



4) Select the correct statements about GRUs

☑ GRUs have fewer parameters compared to LSTMs

☑ GRUs use a single gate to control both input and forget mechanisms

□ GRUs are less effective than LSTMs in handling long-term dependencies

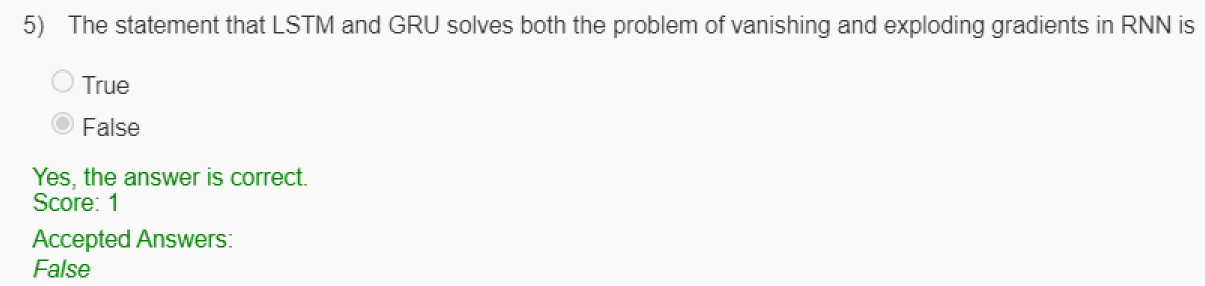
□ GRUs are a type of feedforward neural network

Yes, the answer is correct.
Score: 1

Accepted Answers:

GRUs have fewer parameters compared to LSTMs

GRUs use a single gate to control both input and forget mechanisms



5) The statement that LSTM and GRU solves both the problem of vanishing and exploding gradients in RNN is

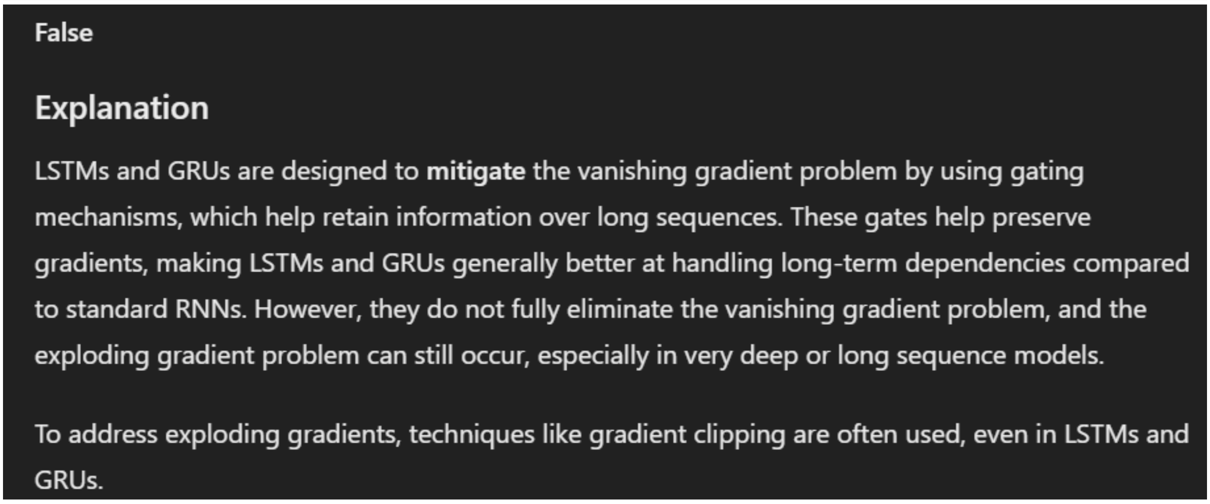
○ True

● False

Yes, the answer is correct.
Score: 1

Accepted Answers:

False

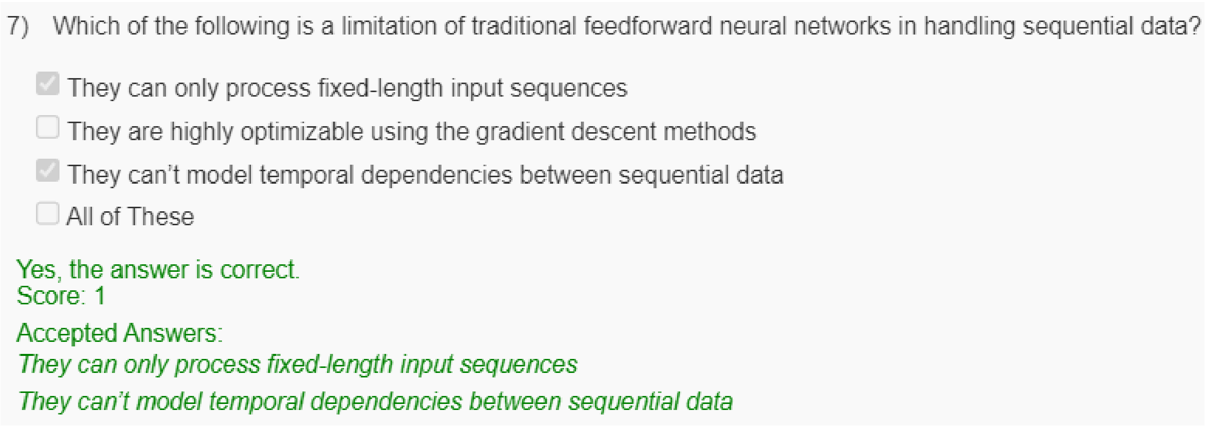


False

Explanation

LSTMs and GRUs are designed to **mitigate** the vanishing gradient problem by using gating mechanisms, which help retain information over long sequences. These gates help preserve gradients, making LSTMs and GRUs generally better at handling long-term dependencies compared to standard RNNs. However, they do not fully eliminate the vanishing gradient problem, and the exploding gradient problem can still occur, especially in very deep or long sequence models.

To address exploding gradients, techniques like gradient clipping are often used, even in LSTMs and GRUs.



7) Which of the following is a limitation of traditional feedforward neural networks in handling sequential data?

☑ They can only process fixed-length input sequences

□ They are highly optimizable using the gradient descent methods

☑ They can’t model temporal dependencies between sequential data

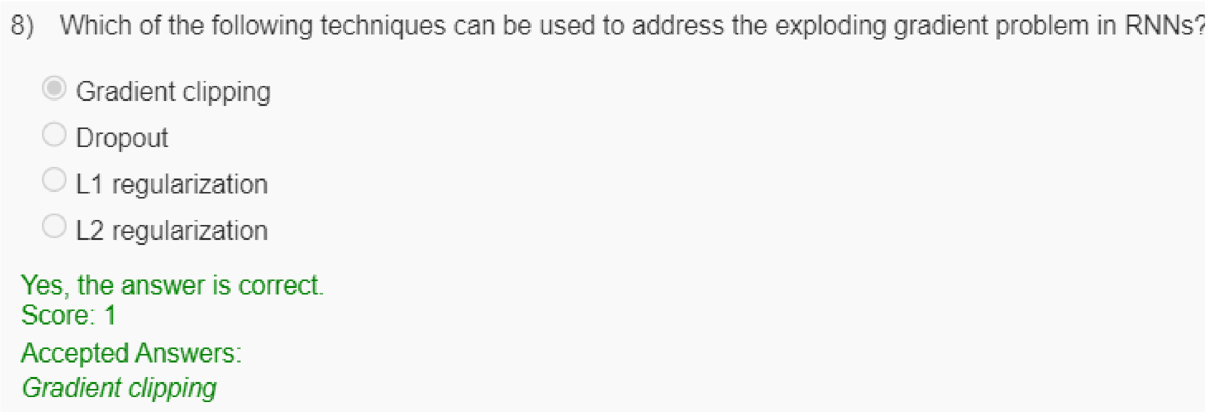
□ All of These

Yes, the answer is correct.
Score: 1

Accepted Answers:

They can only process fixed-length input sequences

They can’t model temporal dependencies between sequential data



8) Which of the following techniques can be used to address the exploding gradient problem in RNNs?

● Gradient clipping

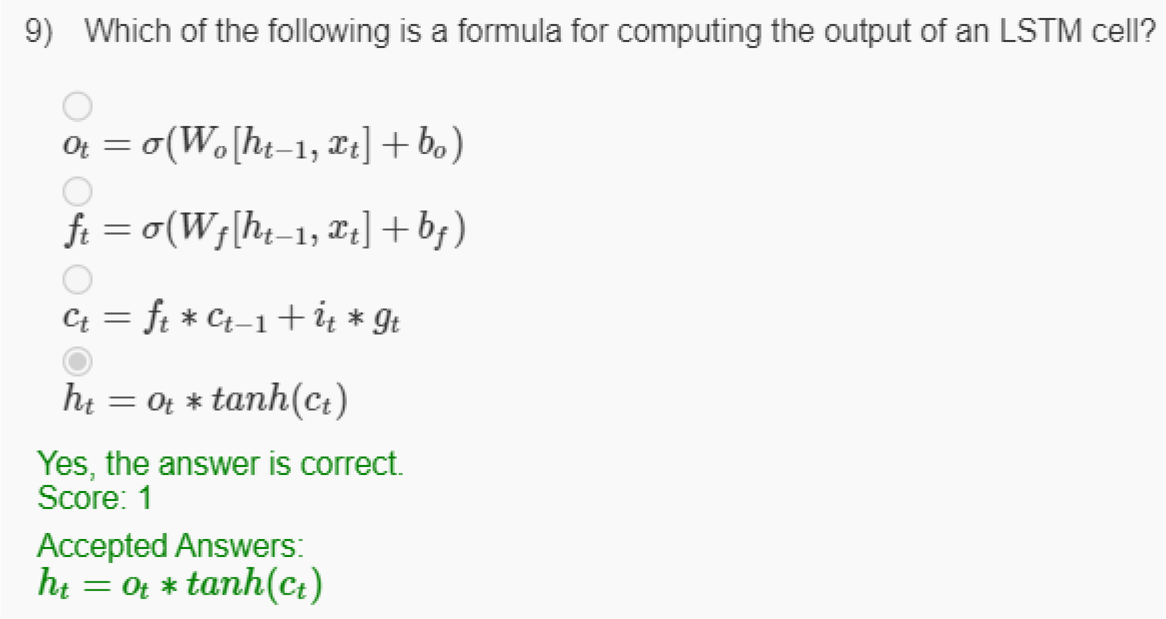
○ Dropout

○ L1 regularization

○ L2 regularization

Yes, the answer is correct.
Score: 1

Accepted Answers:
Gradient clipping



9) Which of the following is a formula for computing the output of an LSTM cell?

○ $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

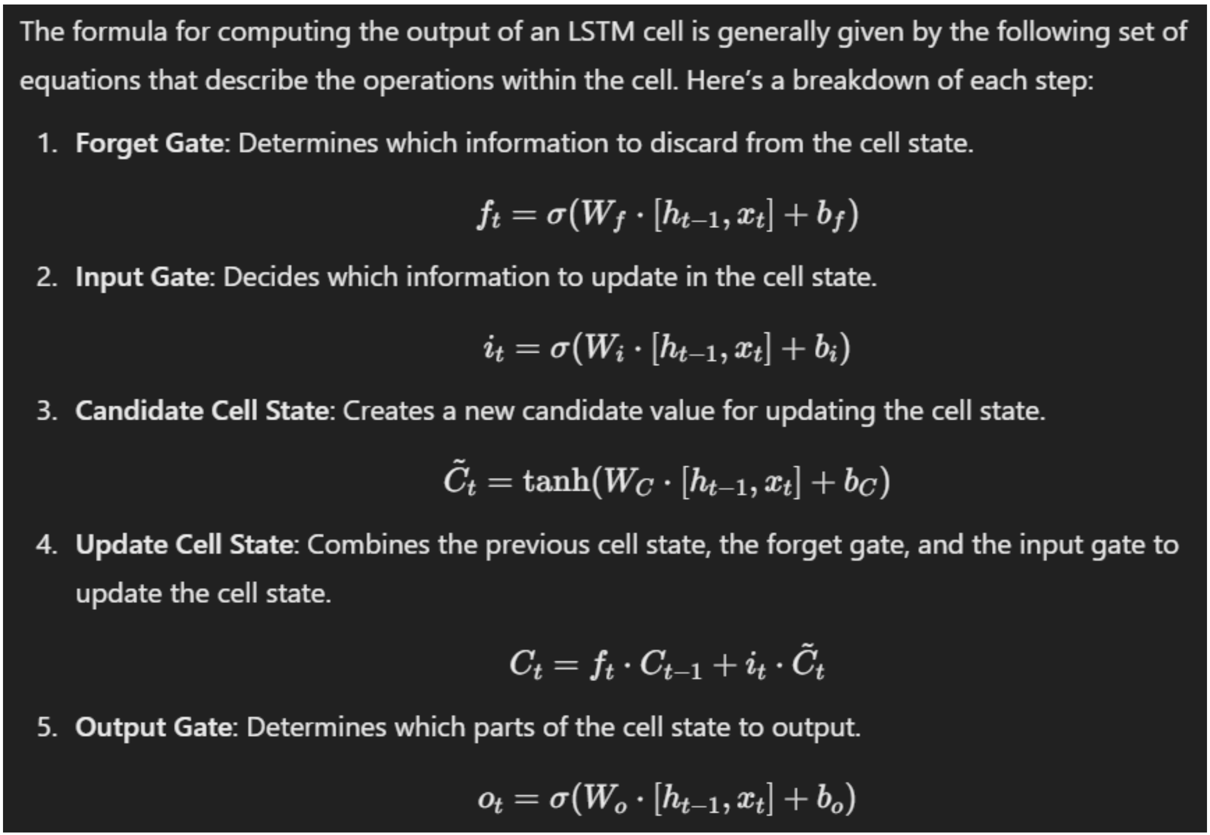
○ $f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$

○ $c_t = f_t * c_{t-1} + i_t * g_t$

● $h_t = o_t * tanh(c_t)$

Yes, the answer is correct.
Score: 1

Accepted Answers:
$h_t = o_t * tanh(c_t)$



The formula for computing the output of an LSTM cell is generally given by the following set of equations that describe the operations within the cell. Here's a breakdown of each step:

1. **Forget Gate**: Determines which information to discard from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. **Input Gate**: Decides which information to update in the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

3. **Candidate Cell State**: Creates a new candidate value for updating the cell state.

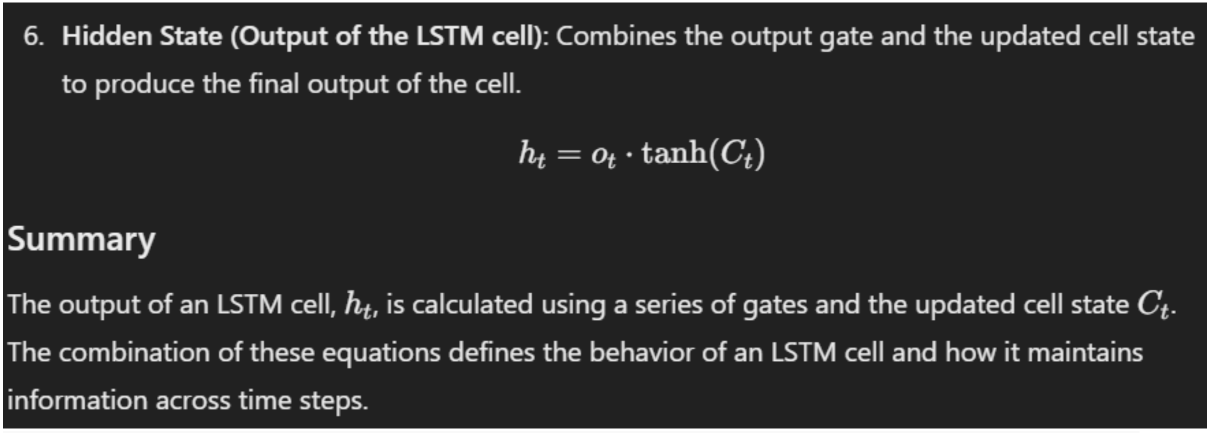
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

4. **Update Cell State**: Combines the previous cell state, the forget gate, and the input gate to update the cell state.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

5. **Output Gate**: Determines which parts of the cell state to output.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

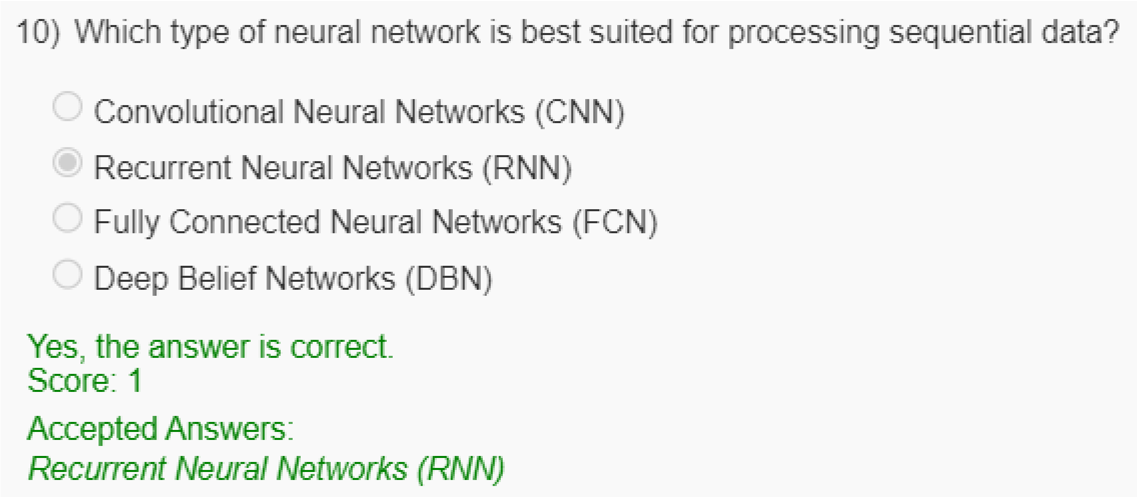


6. **Hidden State (Output of the LSTM cell)**: Combines the output gate and the updated cell state to produce the final output of the cell.

$$h_t = o_t \cdot \tanh(C_t)$$

Summary

The output of an LSTM cell, h_t, is calculated using a series of gates and the updated cell state C_t. The combination of these equations defines the behavior of an LSTM cell and how it maintains information across time steps.



10) Which type of neural network is best suited for processing sequential data?

○ Convolutional Neural Networks (CNN)

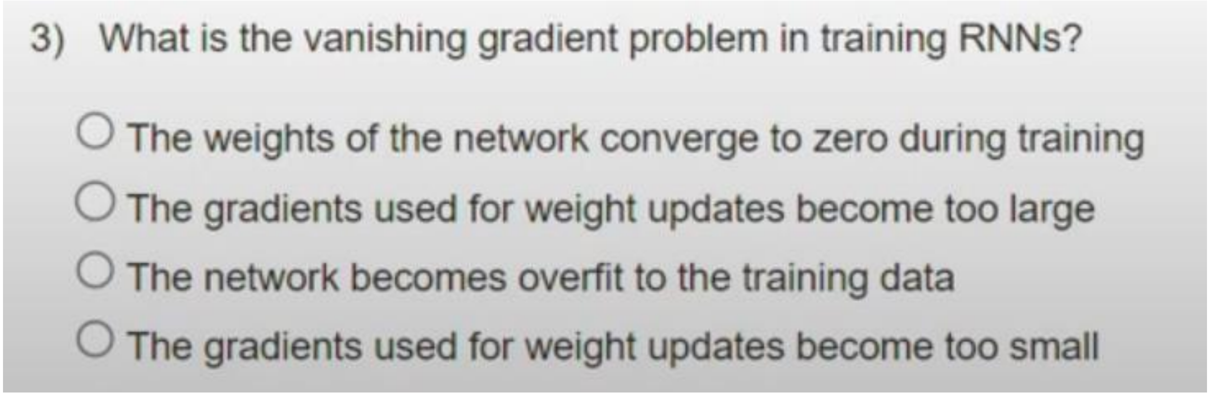
● Recurrent Neural Networks (RNN)

○ Fully Connected Neural Networks (FCN)

○ Deep Belief Networks (DBN)

Yes, the answer is correct.
Score: 1

Accepted Answers:
Recurrent Neural Networks (RNN)



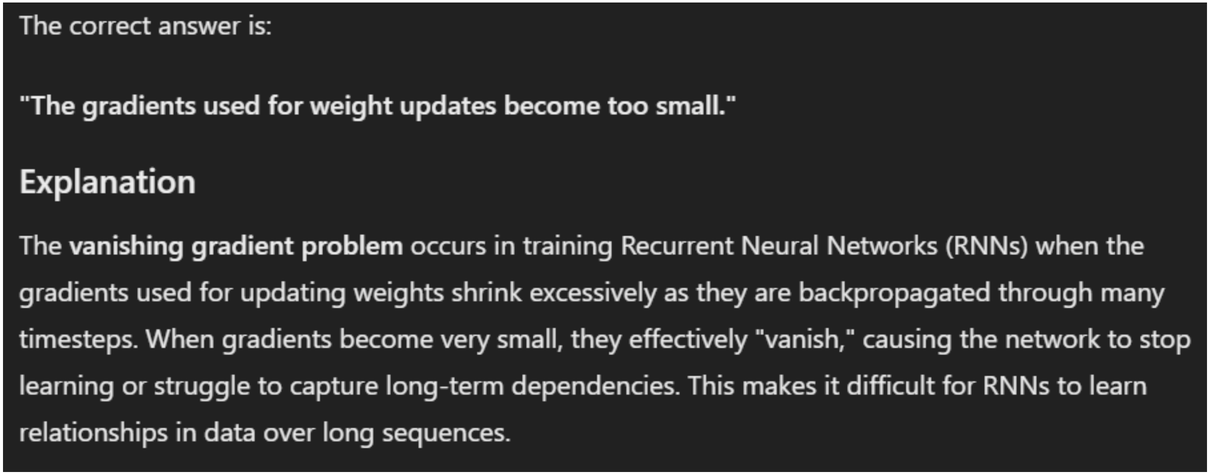
3) What is the vanishing gradient problem in training RNNs?

○ The weights of the network converge to zero during training

○ The gradients used for weight updates become too large

○ The network becomes overfit to the training data

○ The gradients used for weight updates become too small

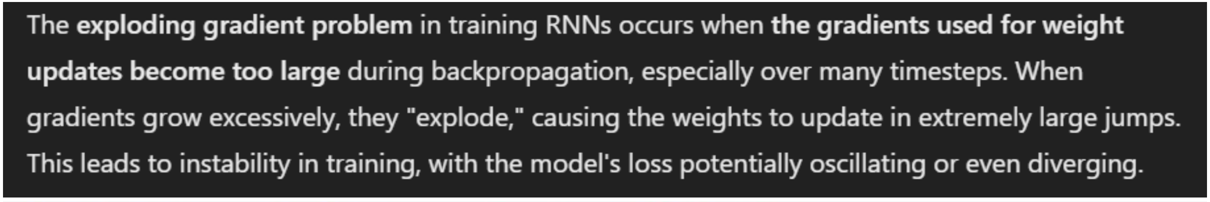


The correct answer is:

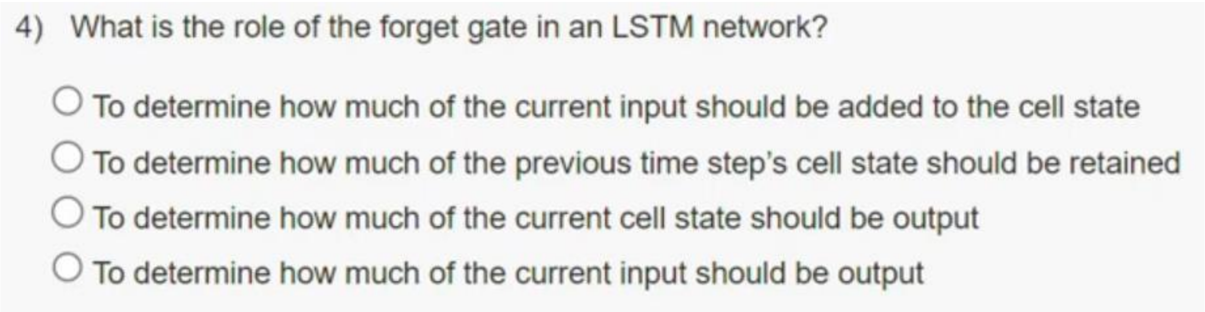
"The gradients used for weight updates become too small."

Explanation

The **vanishing gradient problem** occurs in training Recurrent Neural Networks (RNNs) when the gradients used for updating weights shrink excessively as they are backpropagated through many timesteps. When gradients become very small, they effectively "vanish," causing the network to stop learning or struggle to capture long-term dependencies. This makes it difficult for RNNs to learn relationships in data over long sequences.



The **exploding gradient problem** in training RNNs occurs when **the gradients used for weight updates become too large** during backpropagation, especially over many timesteps. When gradients grow excessively, they "explode," causing the weights to update in extremely large jumps. This leads to instability in training, with the model's loss potentially oscillating or even diverging.



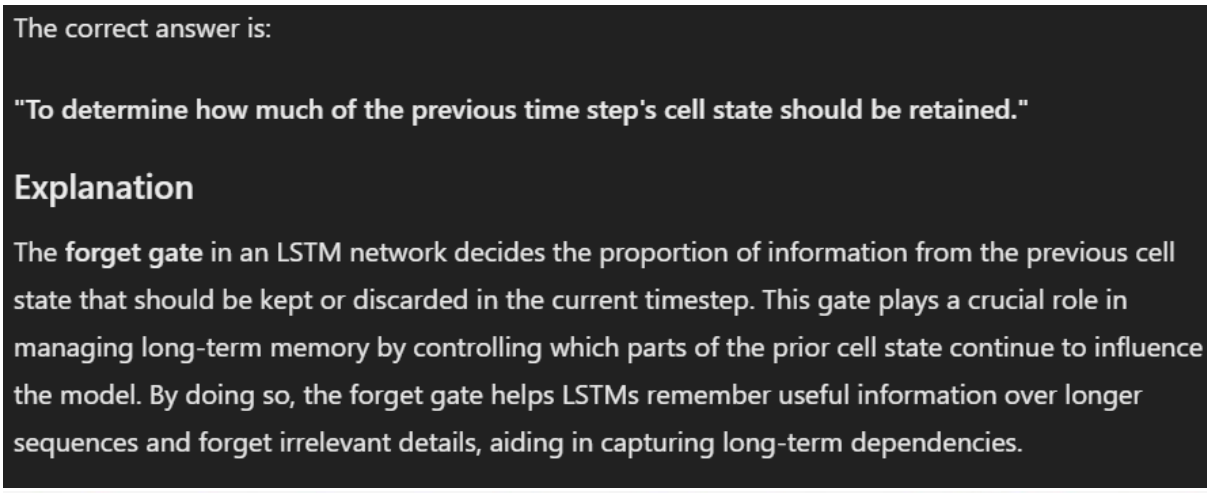
4) What is the role of the forget gate in an LSTM network?

○ To determine how much of the current input should be added to the cell state

○ To determine how much of the previous time step’s cell state should be retained

○ To determine how much of the current cell state should be output

○ To determine how much of the current input should be output

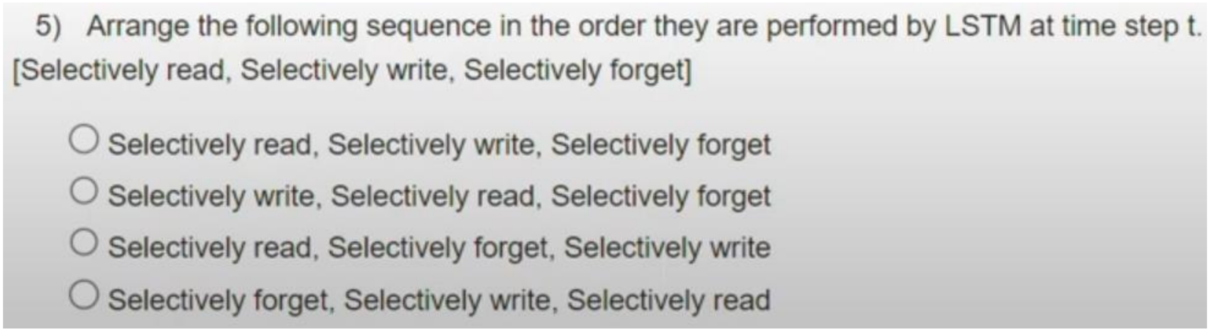


The correct answer is:

"To determine how much of the previous time step's cell state should be retained."

Explanation

The **forget gate** in an LSTM network decides the proportion of information from the previous cell state that should be kept or discarded in the current timestep. This gate plays a crucial role in managing long-term memory by controlling which parts of the prior cell state continue to influence the model. By doing so, the forget gate helps LSTMs remember useful information over longer sequences and forget irrelevant details, aiding in capturing long-term dependencies.



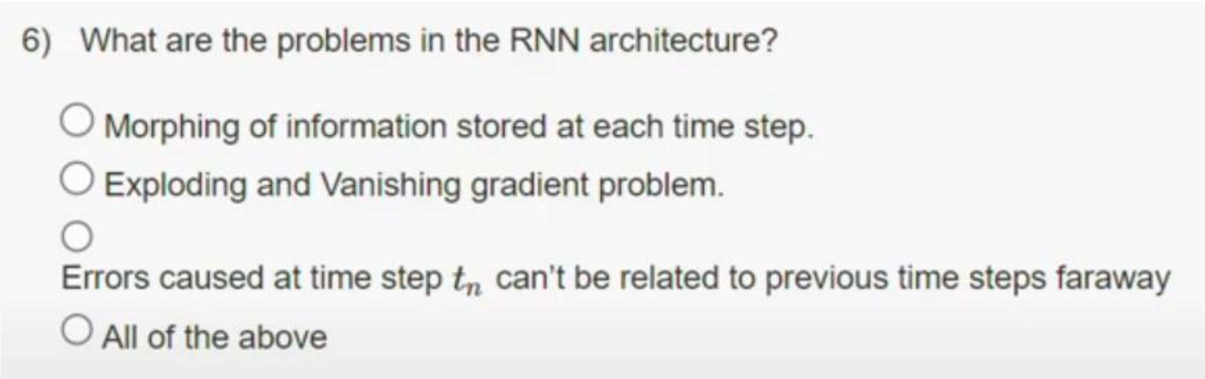
5) Arrange the following sequence in the order they are performed by LSTM at time step t. [Selectively read, Selectively write, Selectively forget]

○ Selectively read, Selectively write, Selectively forget

○ Selectively write, Selectively read, Selectively forget

○ Selectively read, Selectively forget, Selectively write

○ Selectively forget, Selectively write, Selectively read



6) What are the problems in the RNN architecture?

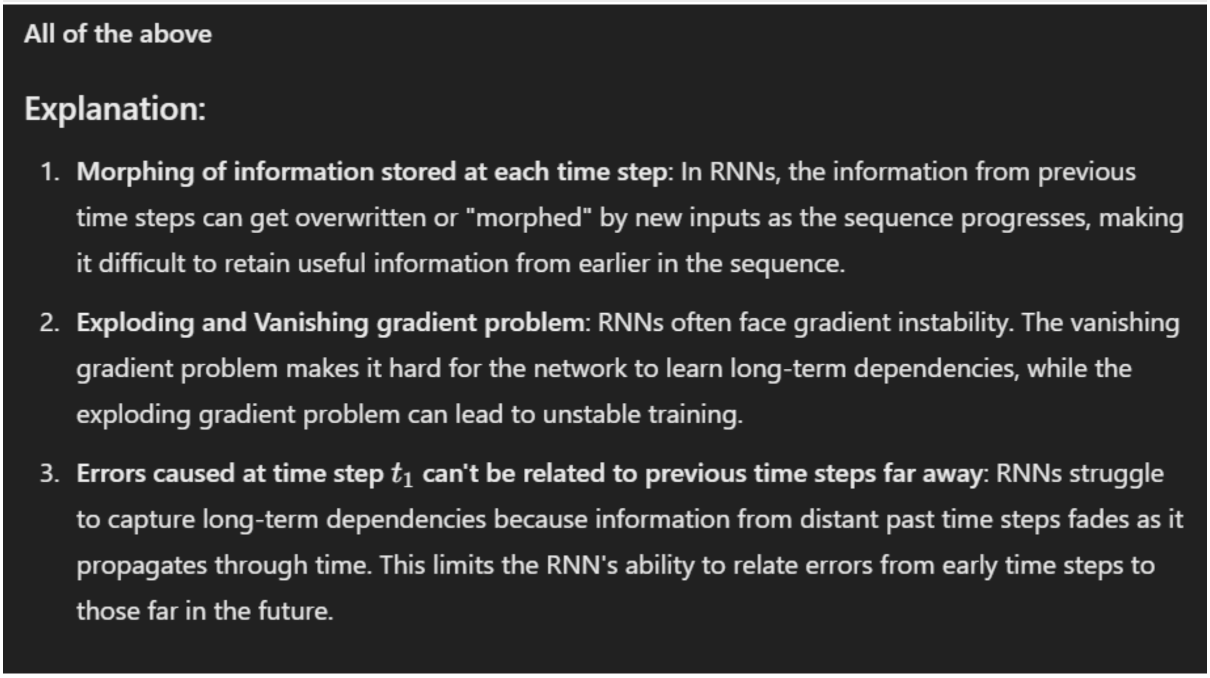
○ Morphing of information stored at each time step.

○ Exploding and Vanishing gradient problem.

○

Errors caused at time step t_n can't be related to previous time steps faraway

○ All of the above

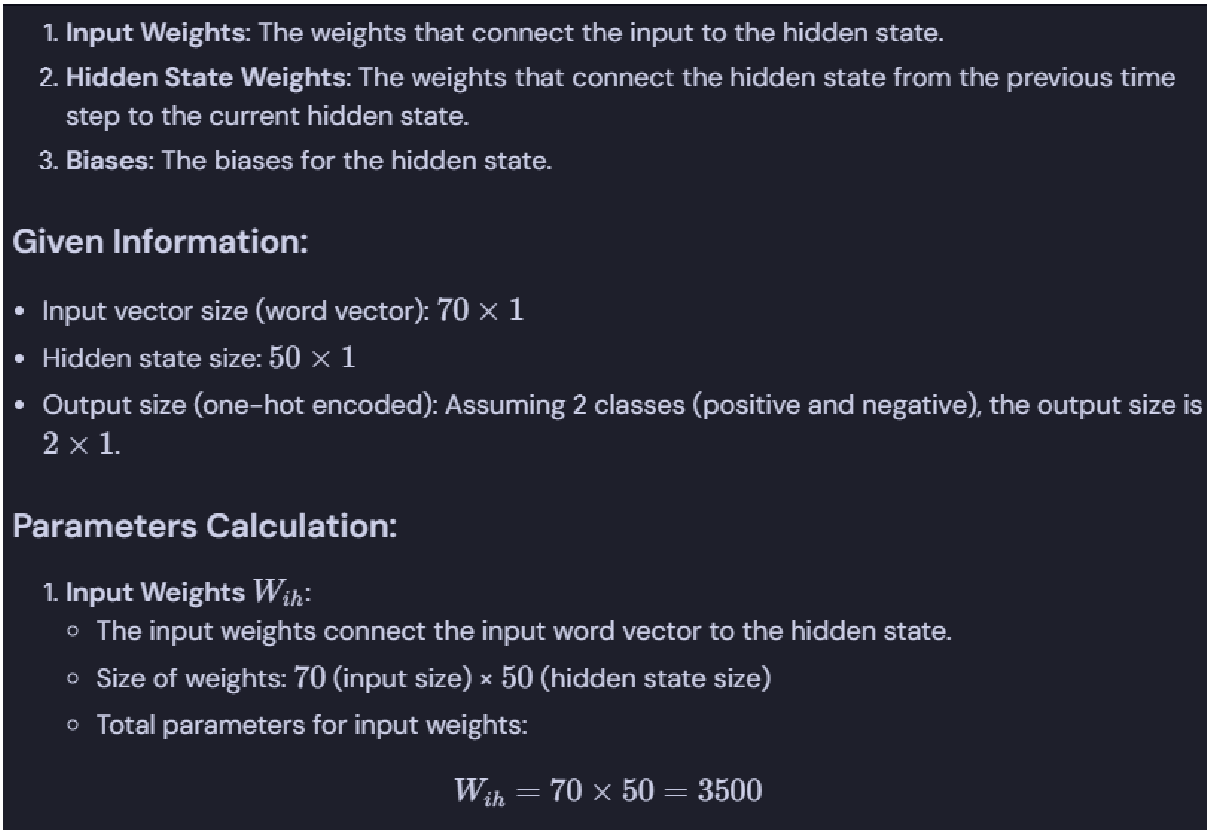


All of the above

Explanation:

1. **Morphing of information stored at each time step**: In RNNs, the information from previous time steps can get overwritten or "morphed" by new inputs as the sequence progresses, making it difficult to retain useful information from earlier in the sequence.
2. **Exploding and Vanishing gradient problem**: RNNs often face gradient instability. The vanishing gradient problem makes it hard for the network to learn long-term dependencies, while the exploding gradient problem can lead to unstable training.
3. **Errors caused at time step t_1 can't be related to previous time steps far away**: RNNs struggle to capture long-term dependencies because information from distant past time steps fades as it propagates through time. This limits the RNN's ability to relate errors from early time steps to those far in the future.

Suppose that we need to develop an RNN model for sentiment classification. The input to the model is a sentence composed of five words, and the output is the sentiment (positive or negative). Assume that each word is represented as a vector of length 70 x 1 and the output labels are one-hot encoded. Further, the state vector s is initialized with all zeros of size 50 x 1. How many parameters (including bias) are there in the network?



1. **Input Weights**: The weights that connect the input to the hidden state.
2. **Hidden State Weights**: The weights that connect the hidden state from the previous time step to the current hidden state.
3. **Biases**: The biases for the hidden state.

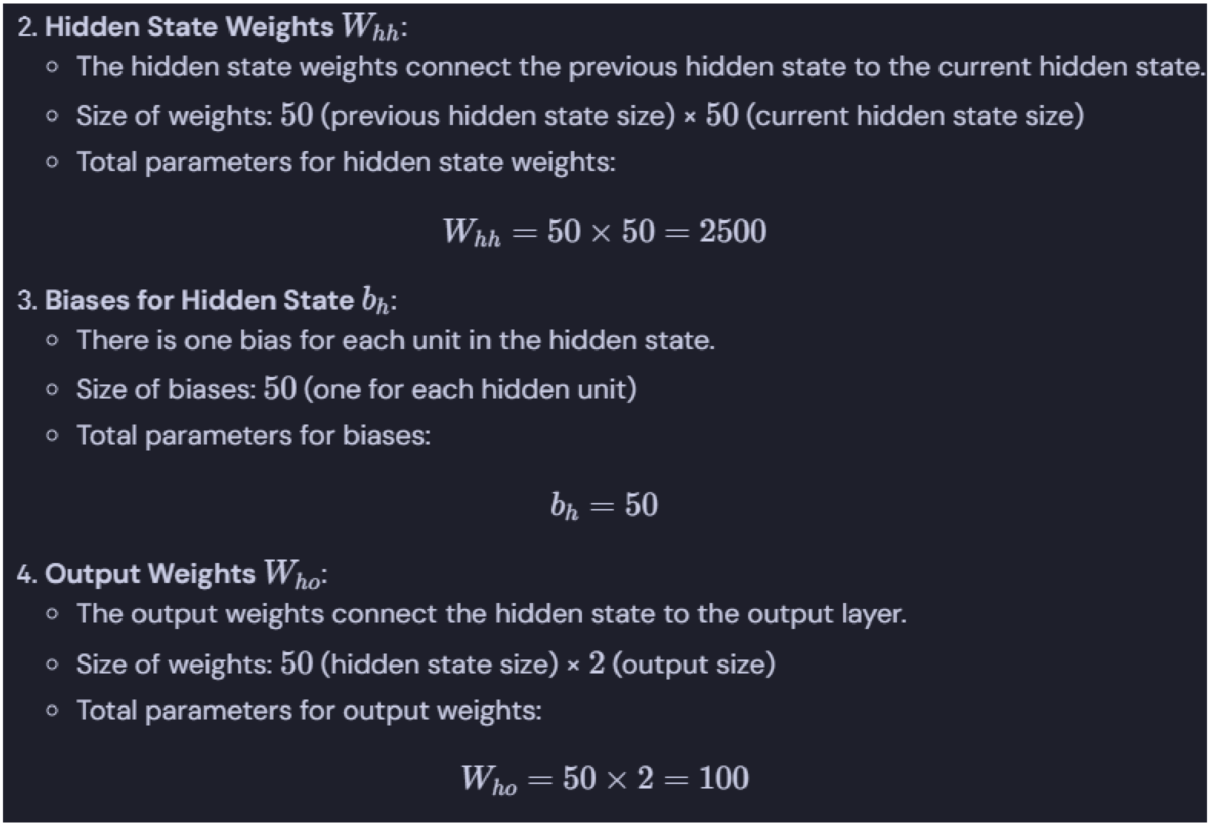
Given Information:

- Input vector size (word vector): 70×1
- Hidden state size: 50×1
- Output size (one-hot encoded): Assuming 2 classes (positive and negative), the output size is 2×1.

Parameters Calculation:

1. **Input Weights** W_{ih}:
   - The input weights connect the input word vector to the hidden state.
   - Size of weights: 70 (input size) × 50 (hidden state size)
   - Total parameters for input weights:

$$W_{ih} = 70 \times 50 = 3500$$



2. **Hidden State Weights** W_{hh}:
   - The hidden state weights connect the previous hidden state to the current hidden state.
   - Size of weights: 50 (previous hidden state size) × 50 (current hidden state size)
   - Total parameters for hidden state weights:

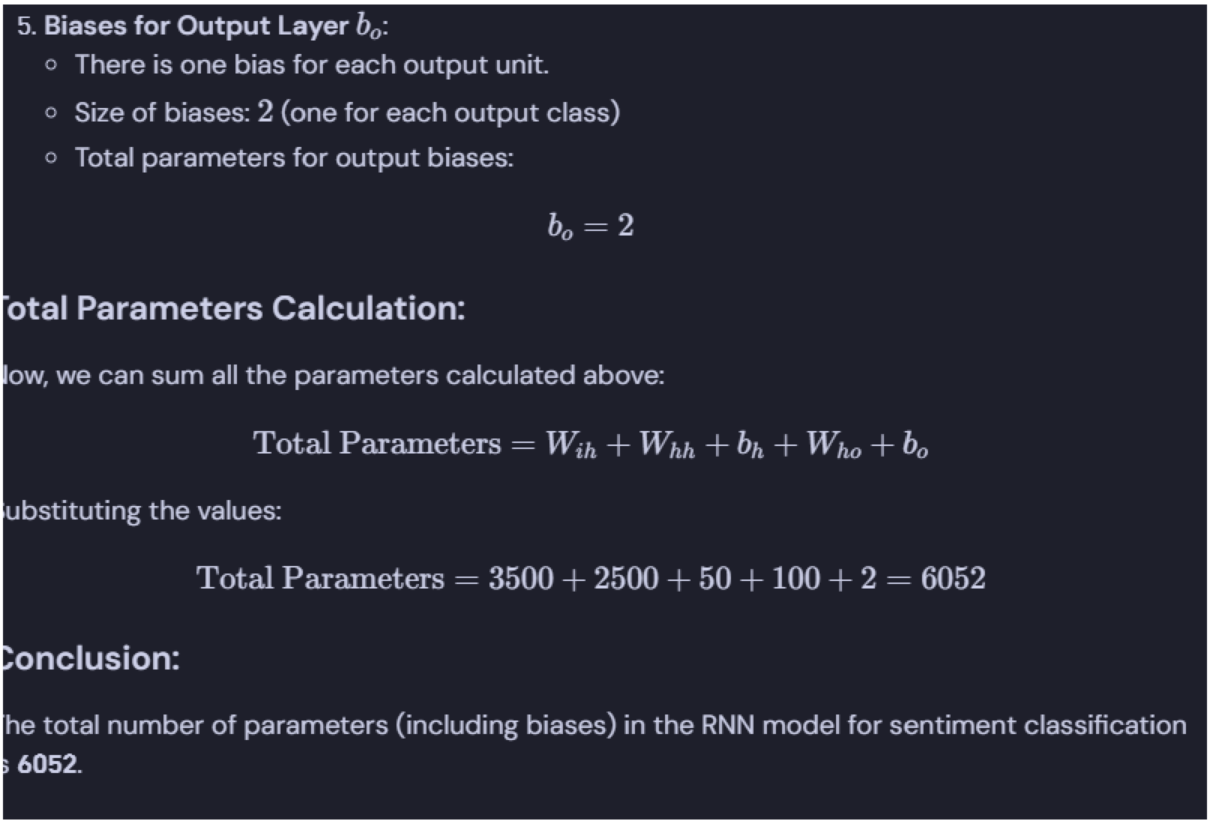
$$W_{hh} = 50 \times 50 = 2500$$

3. **Biases for Hidden State** b_h:
   - There is one bias for each unit in the hidden state.
   - Size of biases: 50 (one for each hidden unit)
   - Total parameters for biases:

$$b_h = 50$$

4. **Output Weights** W_{ho}:
   - The output weights connect the hidden state to the output layer.
   - Size of weights: 50 (hidden state size) × 2 (output size)
   - Total parameters for output weights:

$$W_{ho} = 50 \times 2 = 100$$



5. **Biases for Output Layer b_o:**
    - There is one bias for each output unit.
    - Size of biases: 2 (one for each output class)
    - Total parameters for output biases:

$$b_o = 2$$

otal Parameters Calculation:

ow, we can sum all the parameters calculated above:

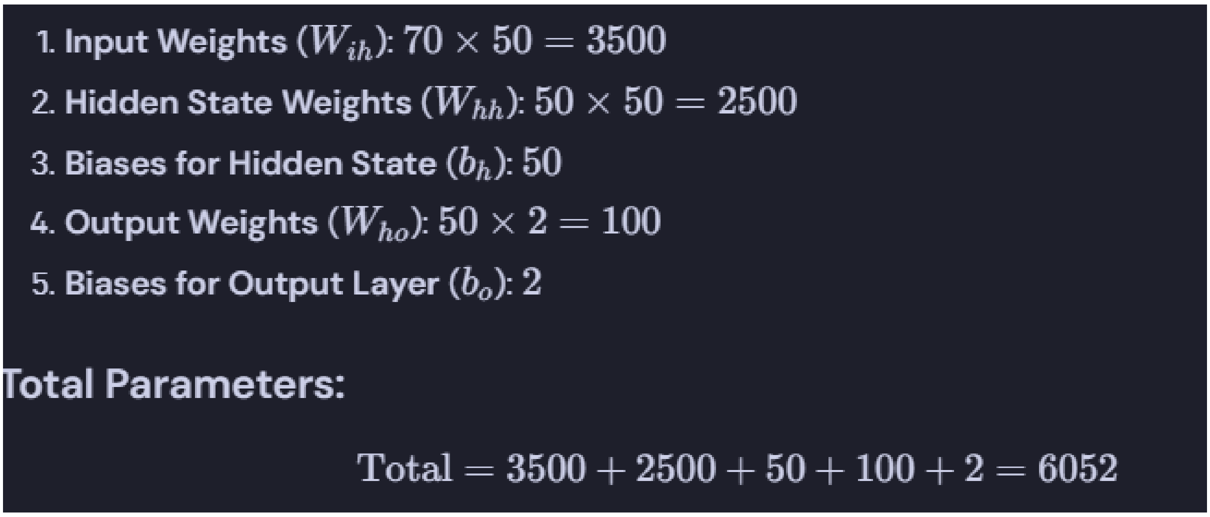
$$\text{Total Parameters} = W_{ih} + W_{hh} + b_h + W_{ho} + b_o$$

ubstituting the values:

$$\text{Total Parameters} = 3500 + 2500 + 50 + 100 + 2 = 6052$$

onclusion:

he total number of parameters (including biases) in the RNN model for sentiment classification **6052**.



1. **Input Weights (W_{ih}):** $70 \times 50 = 3500$
2. **Hidden State Weights (W_{hh}):** $50 \times 50 = 2500$
3. **Biases for Hidden State (b_h):** 50
4. **Output Weights (W_{ho}):** $50 \times 2 = 100$
5. **Biases for Output Layer (b_o):** 2

Total Parameters:

$$\text{Total} = 3500 + 2500 + 50 + 100 + 2 = 6052$$

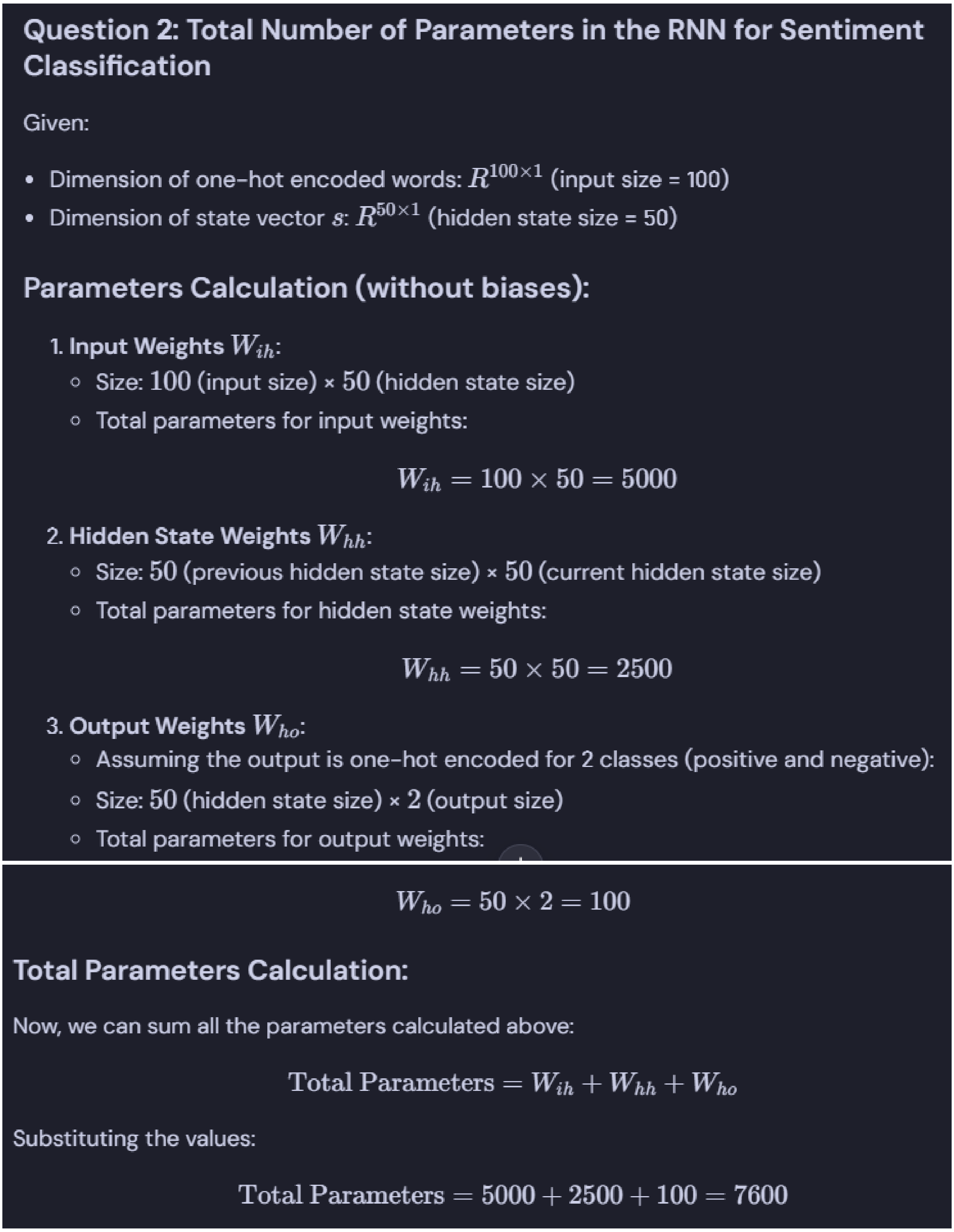
How does LSTM prevent the problem of vanishing gradients?

Options:

- Different activation functions, such as ReLU, are used instead of sigmoid in LSTM.
- Gradients are normalized during backpropagation.
- The learning rate is increased in LSTM.
- Forget gates regulate the flow of gradients during backpropagation.

Question 6:

We construct an RNN for the sentiment classification of text where a text can have positive sentiment or negative sentiment. Suppose the dimension of one-hot encoded words is R100x1, dimension of state vector s is R50x1. What is the total number of parameters in the network? (Don't include biases also in the network)



Question 2: Total Number of Parameters in the RNN for Sentiment Classification

Given:

- Dimension of one-hot encoded words: $R^{100\times1}$ (input size = 100)
- Dimension of state vector s: $R^{50\times1}$ (hidden state size = 50)

Parameters Calculation (without biases):

1. **Input Weights W_{ih}:**
   - Size: 100 (input size) × 50 (hidden state size)
   - Total parameters for input weights:

$$W_{ih} = 100 \times 50 = 5000$$

2. **Hidden State Weights W_{hh}:**
   - Size: 50 (previous hidden state size) × 50 (current hidden state size)
   - Total parameters for hidden state weights:

$$W_{hh} = 50 \times 50 = 2500$$

3. **Output Weights W_{ho}:**
   - Assuming the output is one-hot encoded for 2 classes (positive and negative):
   - Size: 50 (hidden state size) × 2 (output size)
   - Total parameters for output weights:

$$W_{ho} = 50 \times 2 = 100$$

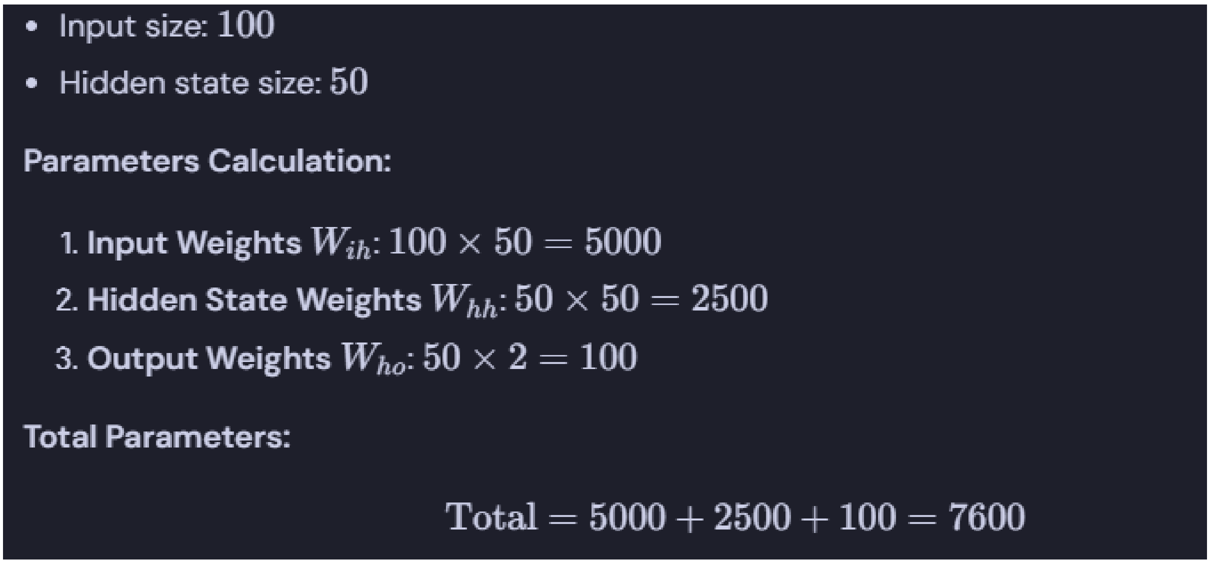
Total Parameters Calculation:

Now, we can sum all the parameters calculated above:

$$\text{Total Parameters} = W_{ih} + W_{hh} + W_{ho}$$

Substituting the values:

$$\text{Total Parameters} = 5000 + 2500 + 100 = 7600$$



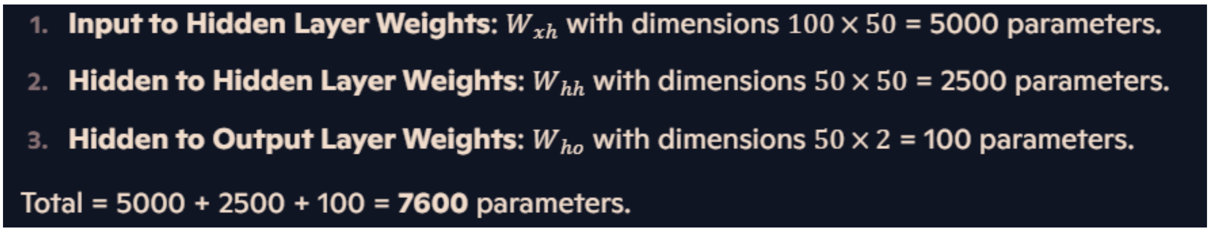
- Input size: 100
- Hidden state size: 50

Parameters Calculation:

1. **Input Weights** W_{ih}: $100 \times 50 = 5000$
2. **Hidden State Weights** W_{hh}: $50 \times 50 = 2500$
3. **Output Weights** W_{ho}: $50 \times 2 = 100$

Total Parameters:

$$\text{Total} = 5000 + 2500 + 100 = 7600$$



1. **Input to Hidden Layer Weights:** W_{xh} with dimensions 100×50 = 5000 parameters.
2. **Hidden to Hidden Layer Weights:** W_{hh} with dimensions 50×50 = 2500 parameters.
3. **Hidden to Output Layer Weights:** W_{ho} with dimensions 50×2 = 100 parameters.

Total = 5000 + 2500 + 100 = **7600** parameters.

What is the objective (loss) function in the RNN?

Options:

- Cross Entropy
- Sum of cross-entropy
- Squared error
- Accuracy

Answer:

Sum of cross-entropy

Explanation:

In RNNs, the sum of cross-entropy loss is a commonly used objective function. It measures the difference between the predicted probability distribution and the true distribution of the target labels at each time step. By minimizing this loss, the RNN learns to make accurate predictions.

The activation function used in the RNN is logistic/sigmoid. What can we say about the value of V = ∂θn/∂θ1?

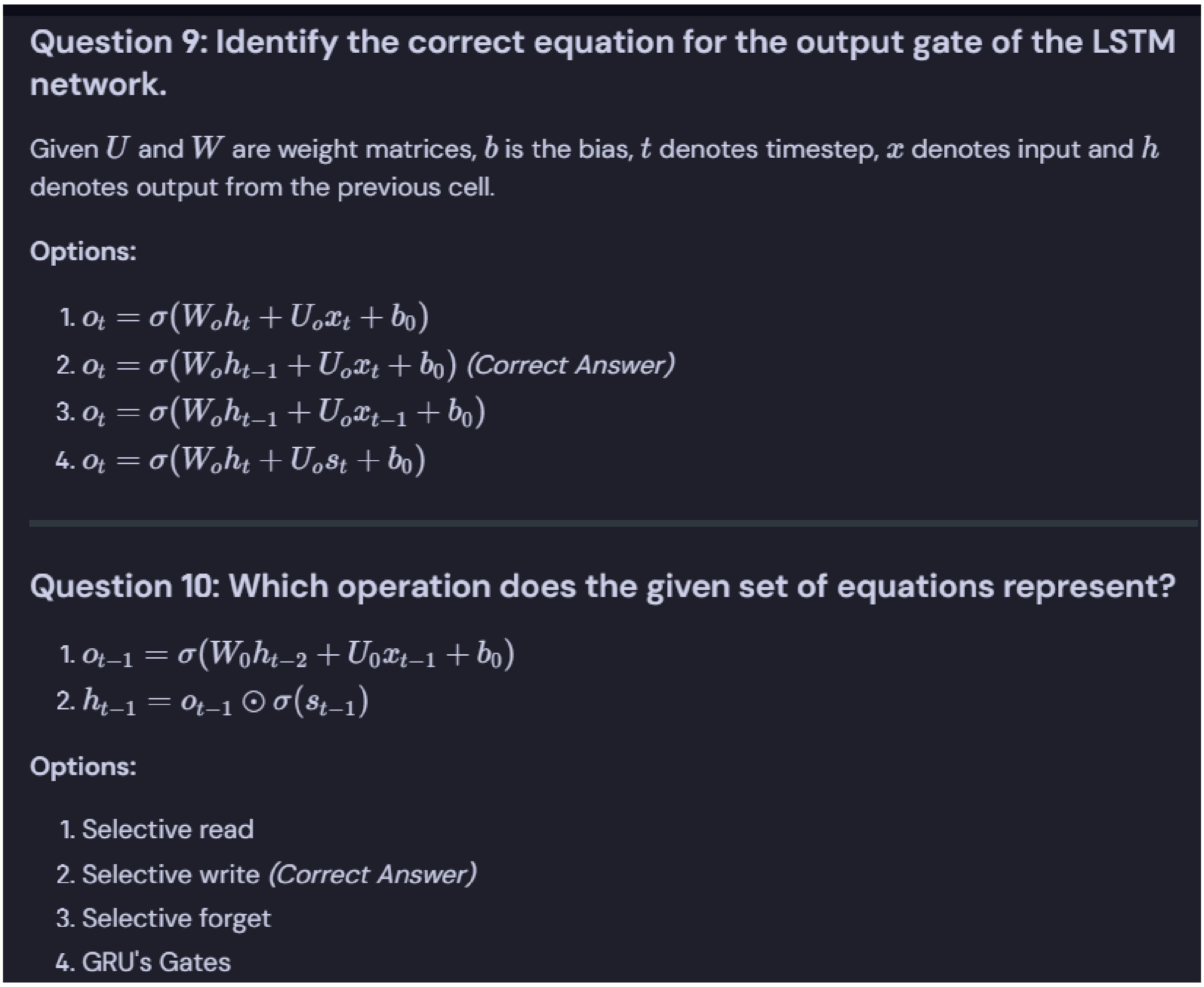
Options:

- Value of V is close to 0.
- Value of V is very high.
- Value of V is 3.5.
- Insufficient information to say anything.

Answer:

Value of V is close to 0.

Explanation:



Question 9: Identify the correct equation for the output gate of the LSTM network.

Given U and W are weight matrices, b is the bias, t denotes timestep, x denotes input and h denotes output from the previous cell.

Options:

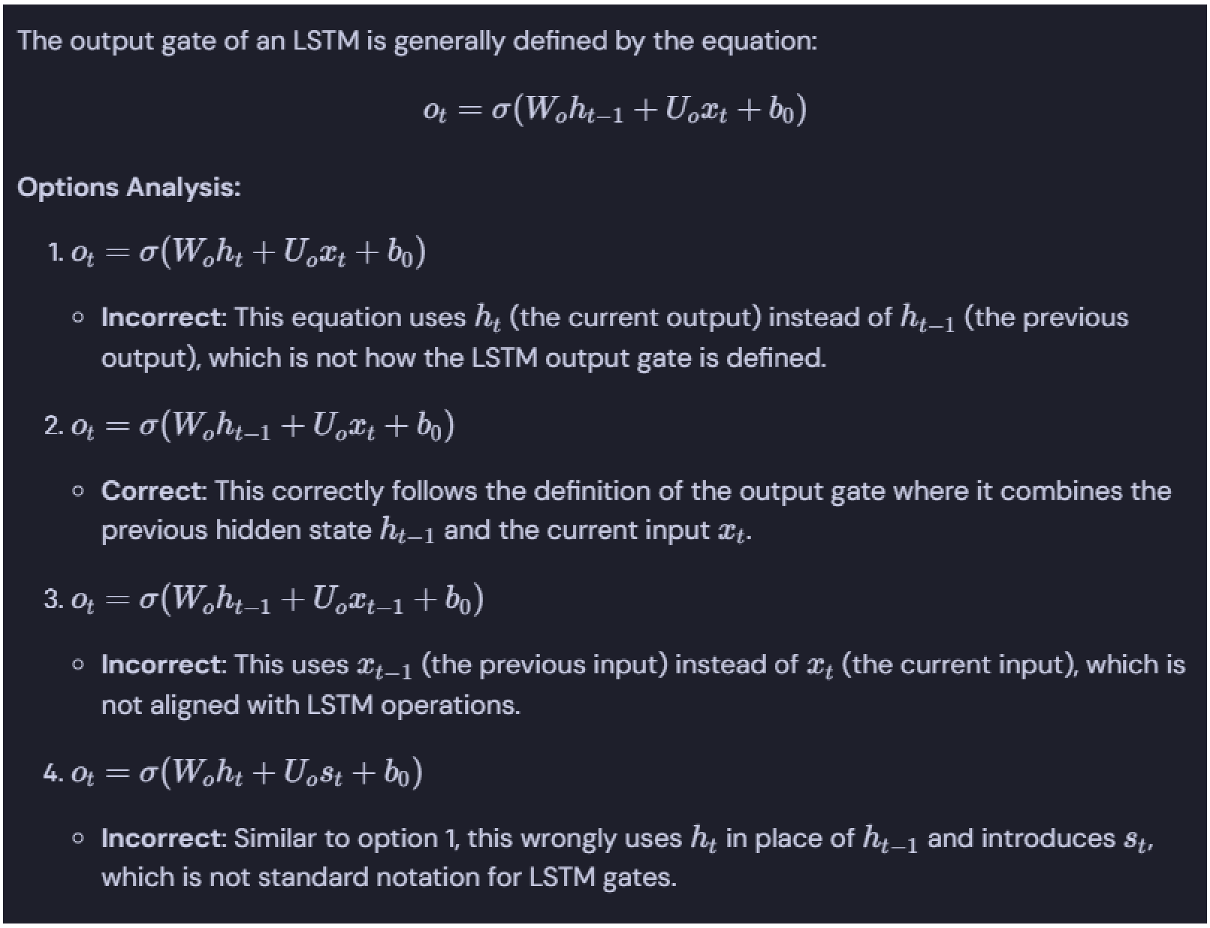
1. $o_t = \sigma(W_o h_t + U_o x_t + b_0)$
2. $o_t = \sigma(W_o h_{t-1} + U_o x_t + b_0)$ *(Correct Answer)*
3. $o_t = \sigma(W_o h_{t-1} + U_o x_{t-1} + b_0)$
4. $o_t = \sigma(W_o h_t + U_o s_t + b_0)$

Question 10: Which operation does the given set of equations represent?

1. $o_{t-1} = \sigma(W_0 h_{t-2} + U_0 x_{t-1} + b_0)$
2. $h_{t-1} = o_{t-1} \odot \sigma(s_{t-1})$

Options:

1. Selective read
2. Selective write *(Correct Answer)*
3. Selective forget
4. GRU's Gates

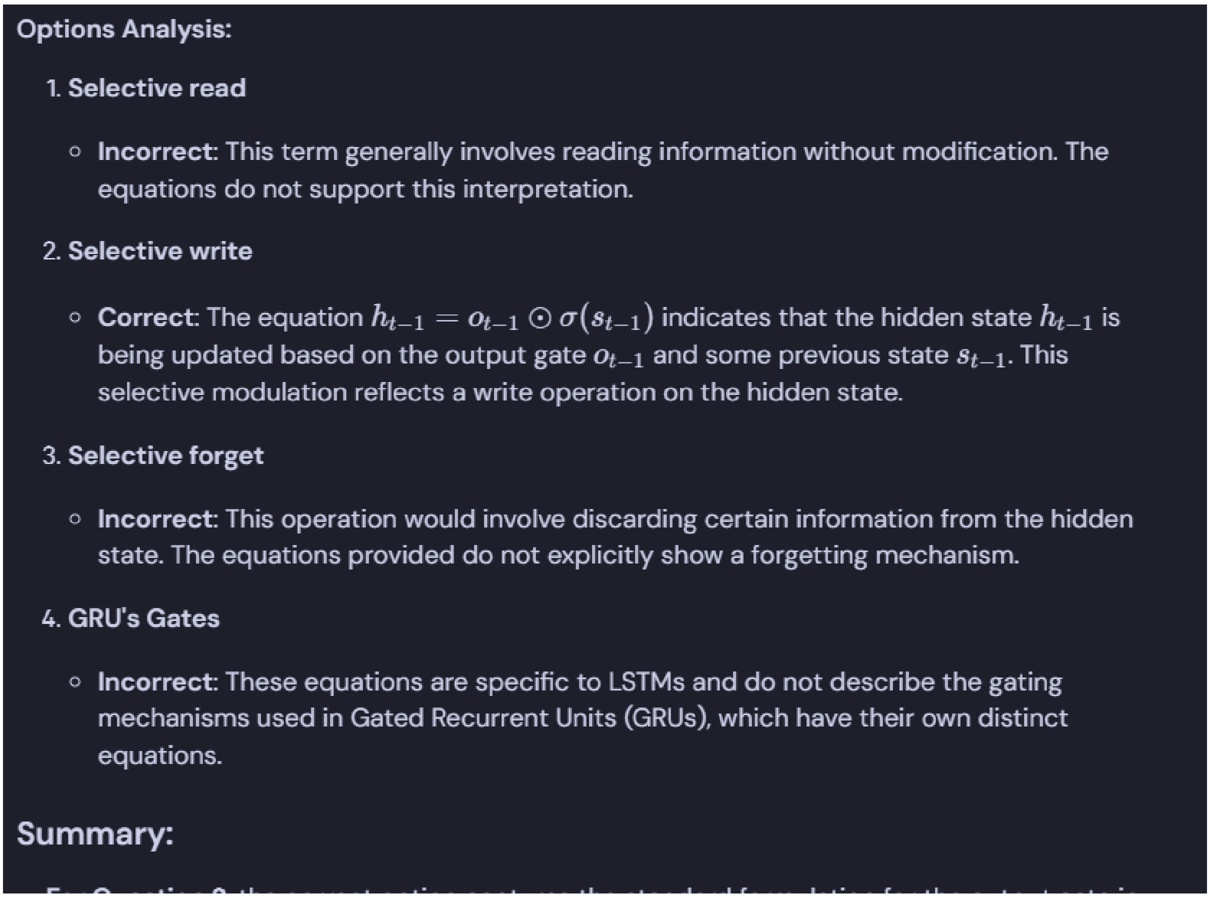


The output gate of an LSTM is generally defined by the equation:

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_0)$$

Options Analysis:

1. $o_t = \sigma(W_o h_t + U_o x_t + b_0)$
   - **Incorrect**: This equation uses h_t (the current output) instead of h_{t-1} (the previous output), which is not how the LSTM output gate is defined.
2. $o_t = \sigma(W_o h_{t-1} + U_o x_t + b_0)$
   - **Correct**: This correctly follows the definition of the output gate where it combines the previous hidden state h_{t-1} and the current input x_t.
3. $o_t = \sigma(W_o h_{t-1} + U_o x_{t-1} + b_0)$
   - **Incorrect**: This uses x_{t-1} (the previous input) instead of x_t (the current input), which is not aligned with LSTM operations.
4. $o_t = \sigma(W_o h_t + U_o s_t + b_0)$
   - **Incorrect**: Similar to option 1, this wrongly uses h_t in place of h_{t-1} and introduces s_t, which is not standard notation for LSTM gates.



Options Analysis:

1. **Selective read**
   - **Incorrect**: This term generally involves reading information without modification. The equations do not support this interpretation.
2. **Selective write**
   - **Correct**: The equation $h_{t-1} = o_{t-1} \odot \sigma(s_{t-1})$ indicates that the hidden state h_{t-1} is being updated based on the output gate o_{t-1} and some previous state s_{t-1}. This selective modulation reflects a write operation on the hidden state.
3. **Selective forget**
   - **Incorrect**: This operation would involve discarding certain information from the hidden state. The equations provided do not explicitly show a forgetting mechanism.
4. **GRU's Gates**
   - **Incorrect**: These equations are specific to LSTMs and do not describe the gating mechanisms used in Gated Recurrent Units (GRUs), which have their own distinct equations.

Summary:

- For Question 2, the correct option captures the standard formulation for the output gate is