1. Consider a simple neural network with one input layer, one hidden layer, and one output layer. [2 marks]

- **Input:** x = 0.5
- **Weights:**
  - w1 = 0.3 (weight from input to hidden node)
  - w2 = 0.7 (weight from hidden node to output node)
- **Biases:**
  - b1 = 0.2 (bias for the hidden node)
  - b2 = -0.1 (bias for the output node)
- **Activation function:** Sigmoid (for both hidden and output nodes)
- **Target output:** y = 0.8

Task:

1. Perform the backward pass to calculate the gradients of the error with respect to the weight w2.

Hint:

- Sigmoid function: sigmoid(z) = 1 / (1 + exp(-z))
- Error function: E = 0.5 * (y - y_predicted)^2

Options:

A. $\partial E/\partial w2 \approx -0.518$
B. $\partial E/\partial w2 \approx -0.0280$
C. $\partial E/\partial w2 \approx -0.318$
D. $\partial E/\partial w2 \approx -0.0318$

Correct Answer: Option D

Explanation:

1. Forward Pass:

- **Hidden Node:**
  - z1 = (x * w1) + b1 = (0.5 * 0.3) + 0.2 = 0.35
  - a1 = sigmoid(z1) = 1 / (1 + exp(-0.35)) ≈ 0.5868
- **Output Node:**
  - z2 = (a1 * w2) + b2 = (0.5868 * 0.7) + (-0.1) ≈ 0.3108
  - y_predicted = sigmoid(z2) = 1 / (1 + exp(-0.3108)) ≈ 0.5772

2. Error Calculation:

- E = 0.5 * (y - y_predicted)^2 = 0.5 * (0.8 - 0.5772)^2 ≈ 0.0246

3. Backward Pass:

- **Output Layer:**
    - ∂E/∂y_predicted = (y_predicted - y) = (0.5772 - 0.8) ≈ -0.2228
    - ∂y_predicted/∂z2 = y_predicted * (1 - y_predicted) = 0.5772 * (1 - 0.5772) ≈ 0.2435 (derivative of sigmoid)
    - ∂E/∂w2 = (∂E/∂y_predicted) * (∂y_predicted/∂z2) * a1 = (-0.2228) * (0.2435) * 0.5868 ≈ -0.0318
    - ∂E/∂b2 = (∂E/∂y_predicted) * (∂y_predicted/∂z2) = (-0.2228) * (0.2435) ≈ -0.0542
- **Hidden Layer:**
    - ∂E/∂a1 = (∂E/∂y_predicted) * (∂y_predicted/∂z2) * w2 = (-0.2228) * (0.2435) * 0.7 ≈ -0.0380
    - ∂a1/∂z1 = a1 * (1 - a1) = 0.5868 * (1 - 0.5868) ≈ 0.2435 (derivative of sigmoid)
    - ∂E/∂w1 = (∂E/∂a1) * (∂a1/∂z1) * x = (-0.0380) * (0.2435) * 0.5 ≈ -0.0046
    - ∂E/∂b1 = (∂E/∂a1) * (∂a1/∂z1) = (-0.0380) * (0.2435) ≈ -0.0093

Results:

- ∂E/∂w2 ≈ -0.0318

These gradients indicate how much each weight and bias contributes to the error. During training, these values would be used to update the weights and biases to reduce the error.

2. A deep convolutional neural network (CNN) is designed for classifying images with subtle textural variations. The architecture incorporates skip connections, batch normalization, and a residual block structure. Which of the following statements best describe(s) the primary combined benefit of these architectural choices for this classification task? [1 mark]

a) It primarily accelerates the training process by reducing the computational cost of each forward and backward pass, enabling faster convergence.

b) It primarily enhances the network's ability to learn robust feature representations by mitigating the vanishing gradient problem and preserving fine-grained details.

c) It primarily improves the network's resilience to image transformations and noise, increasing its generalization performance on unseen data.

d) It primarily increases the network's capacity to capture global context by expanding the receptive field of the convolutional filters in the later layers.

Correct Answer:

b) It primarily enhances the network's ability to learn robust feature representations by mitigating the vanishing gradient problem and preserving fine-grained details.

3. A convolutional layer in a CNN has an input feature map of size 8x8 with 3 channels. The convolutional layer uses a 3x3 kernel with a stride of 1 and no padding.

Given that the output feature map has a total of 1152 elements, how many output channels does the convolutional layer produce? [2 marks]

a) 8 b) 16 c) 32 d) 64

Correct Answer:

c) 32

Explanation:

1. **Output Feature Map Size:**
    - The output feature map size is calculated using the formula: `output_size = (input_size - kernel_size) / stride + 1`
    - In both height and width: `output_size = (8 - 3) / 1 + 1 = 6`
    - Therefore, the output feature map size is 6x6.
2. **Total Elements in Output:**
    - The total number of elements in the output feature map is given as 1152.
3. **Calculate Number of Output Channels:**
    - The number of output channels is calculated by dividing the total number of elements by the size of the output feature map: `output_channels = total_elements / (output_height * output_width)`
    - `output_channels = 1152 / (6 * 6)=32`

4. **Result:**
    - The convolutional layer produces 32 output channels.

4. Consider a single parameter `w` being updated using AdaGrad. [2 marks]

Given:

- **Initial weight (w0):** 1.0
- **Numerator for learning rate equation (ε):** 0.5
- **Gradient at time step 1 (gt1):** 0.8
- **Gradient at time step 2 (gt2):** -0.4
- **delta (δ):** 1e-8 (to prevent division by zero)

Calculate the updated weight (w2) after two time steps.

a) 0.64 b) 0.82 c) 0.88 d) None of the others

Correct Answer:

d) None of the others

Explanation:

1. **AdaGrad Formula:**
    - `vt = vt-1 + gt^2` (Accumulated squared gradients)
    - `wt+1 = wt - (ε / (δ+sqrt(vt))) * gt` (Weight update)
2. **Time Step 1:**
    - `v1 = 0 + gt1^2 = 0 + (0.8)^2 = 0.64`
    - `w1 = w0 - (ε / (δ+sqrt(v1))) * gt1`
    - `w1 = 1.0 - (0.5 / ((1e-8)+sqrt(0.64))) * 0.8`
    - `w1 = 0.5`
3. **Time Step 2:**
    - `v2 = v1 + gt2^2 = 0.64 + (-0.4)^2 = 0.64 + 0.16 = 0.80`
    - `w2 = 0.5 - (0.5 / (1e-8+sqrt(0.80)) * (-0.4)`
    - `w2 ≈ 0.7236`
    - rounding to the nearest hundredth, w2 = 0.72.

Therefore, the updated weight (w2) is approximately 0.72.

5. Imagine a group of hikers attempting to reach the peak of a mountain. The mountain represents the loss landscape of a deep learning model, and the hikers represent the optimization algorithm. Their goal is to find the lowest point (the minimum loss) as quickly and efficiently as possible. **[1 mark]**

- **Hiker A:** Starts with large, confident steps, quickly covering ground. However, they sometimes overshoot the target, especially in narrow valleys.
- **Hiker B:** Takes small, cautious steps, ensuring they don't miss any subtle dips. They are reliable but slow, especially on flat terrain.
- **Hiker C:** Adjusts their step size based on the steepness of the terrain. They take bigger steps on steep slopes and smaller steps in flatter areas.
- **Hiker D:** Maintains a memory of the terrain they've covered, adjusting their direction based on the accumulated slope. For them, the pattern of the terrain in the recent past matters more in deciding the direction. They adapt their path to avoid getting stuck in local minima.

Question:

Which hiker BEST represents the behavior of the Adam optimizer?

a) Hiker A b) Hiker B c) Hiker C d) Hiker D

Correct Answer:

d) Hiker D

Explanation:

- **A (Large, confident steps):** Represents optimizers with fixed, large learning rates, like basic Gradient Descent with a high learning rate.
- **B (Small, cautious steps):** Represents optimizers with small, fixed learning rates, like basic Gradient Descent with a low learning rate.
- **C (Adjusts step size):** Represents optimizers that adapt the learning rate based on the gradient.
- **D (Maintains memory):** Represents Adam, which combines the adaptive learning rate of RMSprop with momentum, allowing it to maintain a memory of past gradients and adapt its direction.

6. Match the following descriptions with the appropriate concept: [2 marks]

Descriptions:

- A. Model that performs poorly on both the training and test sets.
- B. Model that performs very well on the training set but poorly on the test set.
- C. The tendency of a model to consistently make similar errors across datasets.
- D. The sensitivity of a model's predictions to small fluctuations in the training data.

Concepts:

1. High Bias
2. High Variance
3. Underfitting
4. Overfitting

Which of the following options is correct:

A) A → 3, B → 4, C → 1, D → 2
B) A → 3, B → 2, C → 1, D → 4
C) A → 2, B → 4, C → 2, D → 1
D) A → 1, B → 3, C → 2, D → 4

Correct Answer: Option A

- A. → 3. Underfitting
- B. → 4. Overfitting
- C. → 1. High Bias
- D. → 2. High Variance

7. Match the following descriptions with the appropriate CNN concepts/architectural properties: [1 mark]

Descriptions:

- A. Introduces non-linearity, enabling the network to learn complex relationships in data.
- B. Mitigates the loss of spatial information during downsampling, enabling upsampling and pixel-level predictions.
- C. Reduces the spatial dimensions of feature maps, providing translation equivariance and reducing computational load.
- D. Allows for the training of extremely deep networks by addressing the vanishing gradient problem and facilitating the flow of information.
- E. Introduces parameter sharing in the network, potentially reducing overfitting and computational cost.

- F. Allows the network to perform convolution with reduced number of computations.

Concepts:

1. ReLU Activation
2. Transposed Convolutions
3. Pooling Layers
4. Residual Connections
5. Strided Convolutions
6. Depthwise Separable Convolutions
7. Leaky ReLU Activation

Which of the following options is/ are correct

A) A → 1, B → 2, C → 5, D → 7, E → 3, F → 4
B) A → 7, B → 2, C → 5, D → 4, E → 3, F → 1
C) A → 1, B → 2, C → 3, D → 4, E → 5, F → 6
D) A → 7, B → 2, C → 3, D → 4, E → 5, F → 6

Correct Answers: Option C, D

- A → 1/7, ReLU Activation/ Leaky ReLU Activation
- B → 2, Transposed Convolutions
- C → 3, Pooling Layers
- D → 4, Residual Connections
- E → 5, Strided Convolutions
- F → 6 Depthwise Separable Convolutions

8. A standard convolutional layer with an input feature map of size 64x64x32 and an output feature map of size 64x64x64 uses a 3x3 kernel. If this layer is replaced with a depthwise separable convolution, what is the approximate reduction in the number of parameters, and what is the primary benefit of this reduction? [2 marks]

a) Approximately 53% reduction; primarily reduces computational complexity during inference.

b) Approximately 77% reduction;primarily accelerates the training process by reducing memory requirements.

c) Approximately 88% reduction; primarily reduces the risk of overfitting during training.

d) Approximately 95% reduction; primarily improves the model's ability to generalize to unseen data.

Correct Answer:

c) Approximately 88% reduction; primarily reduces the risk of overfitting during training.

Explanation:

1. **Standard Convolution Parameters:**
    - Number of parameters = kernel_height * kernel_width * input_channels * output_channels
    - Parameters = 3 * 3 * 32 * 64 = 18432
2. **Depthwise Separable Convolution Parameters:**
    - Depthwise parameters = kernel_height * kernel_width * input_channels = 3 * 3 * 32 = 288
    - Pointwise parameters = input_channels * output_channels = 32 * 64 = 2048
    - Total parameters = 288 + 2048 = 2336
3. **Parameter Reduction:**
    - Reduction = (18432 - 2336) / 18432 ≈ 0.873 or 87.3%
    - Therefore the closest value is 88%.
4. **Primary Benefit:**
    - The primary benefit of reducing the number of parameters is to reduce the risk of overfitting, especially when dealing with limited training data. While it also reduces computational complexity and memory requirements, the impact on overfitting is the most significant.

9. Which of the following best describes the primary purpose of dropout in a deep neural network? [1 mark]

a) To accelerate the training process by reducing the number of computations.

b) To reduce overfitting by preventing the network from relying too heavily on specific neurons.

c) To increase the model's capacity to learn complex patterns by adding more neurons.

d) To improve the model's interpretability by simplifying the network architecture.

Correct Answer:

b) To reduce overfitting by preventing the network from relying too heavily on specific neurons.

10. Match the following descriptions with the appropriate optimization algorithm: [2 marks]

Descriptions:

- A. Updates the model parameters using the gradient calculated from the entire training dataset in each iteration.
- B. Updates the model parameters using the gradient calculated from a randomly selected single data point in each iteration.
- C. Updates the model parameters using the gradient calculated from a small, randomly selected subset of the training dataset in each iteration.
- D. Exhibits the most stable convergence behavior but can be computationally expensive for large datasets.
- E. Exhibits the most noisy convergence behavior but can be computationally efficient for large datasets.
- F. Offers a compromise between stability and efficiency, commonly used in practice.

Concepts:

1. Gradient Descent (GD)
2. Stochastic Gradient Descent (SGD)
3. Mini-Batch Gradient Descent (MBGD)

Which of the following options is correct:

A) A → 1, B → 2, C → 3, D → 2, E → 1, F → 3
B) A → 1, B → 2, C → 3, D → 1, E → 2, F → 3
C) A → 1, B → 3, C → 2, D → 1, E → 3, F → 2
D) A → 2, B → 1, C → 3, D → 1, E → 2, F → 3

Correct Answer: Option B

- A. → 1. Gradient Descent (GD)
- B. → 2. Stochastic Gradient Descent (SGD)
- C. → 3. Mini-Batch Gradient Descent (MBGD)
- D. → 1. Gradient Descent (GD)
- E. → 2. Stochastic Gradient Descent (SGD)
- F. → 3. Mini-Batch Gradient Descent (MBGD)

11. What is the primary characteristic that distinguishes Recurrent Neural Networks (RNNs) from traditional feedforward neural networks? [1 mark]

a) RNNs use convolutional layers for feature extraction.

b) RNNs create a time-dependent hidden state which captures the information of past tokens in the sequence.

c) RNNs employ only tanh activation to prevent overfitting.

d) RNNs are useful only for an input sequence to an output sequence.

Correct Answer:

b) RNNs create a time-dependent hidden state which captures the information of past tokens in the sequence.

12. A sequence model designed for complex sequence-to-sequence modeling exhibits the following behaviors during training: [1 mark]

- The training loss initially decreases rapidly, but then plateaus and oscillates slightly.
- The validation loss does not decrease significantly.
- Gradient norms are observed to fluctuate significantly during training, occasionally spiking to very high values.

Which of the following best describes the most likely combination of underlying issues and effective mitigation strategies?

a) Exploding gradients combined with insufficient model capacity; mitigate by switching to GRU cells and increasing the number of hidden units.

b) Exploding gradients combined with overfitting; mitigate by implementing gradient clipping and early stopping.

c) Saddle points in the loss landscape combined with a mismatch between the data distribution and the model's inductive bias; mitigate by using adaptive learning rate optimizers and data augmentation.

d) A poorly chosen initialization scheme; mitigate by using He initialization.

Correct Answer:

b) Exploding gradients combined with overfitting; mitigate by implementing gradient clipping and early stopping.

13. What fundamental challenge does Backpropagation Through Time (BPTT) address in Recurrent Neural Networks (RNNs), and what underlying principle enables it to do so, while also contributing to its primary limitations? [1 mark]

a) BPTT addresses the issue of vanishing gradients by unrolling the RNN and applying gradient clipping, enabling it to learn long-range dependencies but limiting its ability to handle variable-length sequences.

b) BPTT addresses the issue of exploding gradients by applying weight sharing across time steps, enabling it to learn complex temporal patterns but limiting its ability to parallelize computations.

c) BPTT addresses the issue of learning temporal dependencies by treating the unfolded RNN as a deep feedforward network and applying the chain rule, enabling it to propagate gradients through time but leading to potential vanishing or exploding gradients.

d) BPTT addresses the issue of overfitting by introducing dropout in the recurrent connections, enabling it to generalize to unseen sequences but limiting its ability to capture fine-grained temporal details.

Correct Answer:

c) BPTT addresses the issue of learning temporal dependencies by treating the unfolded RNN as a deep feedforward network and applying the chain rule, enabling it to propagate gradients through time but leading to potential vanishing or exploding gradients.

14. Match the following descriptions with the appropriate recurrent neural network architecture:

[1 mark]

Descriptions:

- A. Experiences significant degradation in gradient signal over extended time steps, limiting its ability to propagate information from early inputs.
- B. Employs a dedicated memory pathway that allows for the selective retention and modification of information over arbitrary time intervals.
- C. Utilizes a reduced set of gating mechanisms, effectively merging the cell state and hidden state, leading to a streamlined information flow.
- D. Exhibits a tendency to struggle with tasks requiring precise control over the duration of information retention.
- E. Demonstrates a capacity to adaptively reset or update its memory based on context, facilitating the capture of both short-term and long-term dependencies.

Concepts:

1. Standard RNN
2. Long Short-Term Memory (LSTM)
3. Gated Recurrent Unit (GRU)

Which of the following options is/are correct:

A) A → 1, B → 2, C → 2, D → 3, E → 3
B) A → 1, B → 2, C → 3, D → 1, E → 3
C) A → 1, B → 2, C → 3, D → 1, E → 2
D) A → 1, B → 3, C → 2, D → 1, E → 2

Correct Answers: Option B, C

- A. → 1. Standard RNN
- B. → 2. LSTM
- C. → 3. GRU
- D. → 1. Standard RNN
- E. → 2. LSTM/ 3. GRU

15. In the context of deep learning, which of the following statements about Batch Normalization is *incorrect*? [1 mark]

A) Batch Normalization helps to reduce the internal covariate shift by normalizing the input to each layer.

B) Batch Normalization introduces two additional learnable parameters per feature, known as gamma (γ) and beta (β).

C) Batch Normalization can only be applied to convolutional neural networks (CNNs), not to fully connected networks.

D) During training, Batch Normalization uses the mean and variance calculated over the current mini-batch.

E) During inference, Batch Normalization uses the mean and variance calculated only over the last training mini-batch.

Correct Answers:

- C) Batch Normalization can only be applied to convolutional neural networks (CNNs), not to fully connected networks.

- E) During inference, Batch Normalization uses the mean and variance calculated only over the last training mini-batch.

16. In a Long Short-Term Memory (LSTM) cell, which of the following best describes the primary function of the "forget gate"? [1 mark]

a) To forget a certain amount of current token that is added to the cell state.

b) To determine which parts of the cell state should be discarded.

c) To regulate the amount of information that is passed from the cell state to the output.

d) To introduce non-linearity into the cell state update.

Correct Answer:

b) To determine which parts of the cell state should be discarded.

17. Which of the following best describes the core mechanism that distinguishes the Adam optimizer from standard gradient descent and other adaptive learning rate methods like AdaGrad or RMSprop? [1 mark]

a) Adam primarily focuses on reducing the learning rate over time to ensure convergence, similar to learning rate decay.

b) Adam combines the benefits of both momentum and adaptive learning rates by using estimates of both the first and second moments of the gradients.

c) Adam utilizes a fixed learning rate throughout training, ensuring stable convergence even with noisy gradients.

d) Adam primarily applies L1 and L2 regularization during the optimization process to prevent overfitting.

Correct Answer:

b) Adam combines the benefits of both momentum and adaptive learning rates by using estimates of both the first and second moments of the gradients.

18. Which of the following BEST describes the primary advantage of using a Bidirectional LSTM (BiLSTM) over a traditional Recurrent Neural Network (RNN) or a unidirectional LSTM for sequence modeling tasks? [1 mark]

a) BiLSTMs are less prone to vanishing gradients, allowing them to learn from longer sequences.

b) BiLSTMs can process sequences in parallel, leading to faster training times.

c) BiLSTMs can capture contextual information from both past and future time steps in a sequence.

d) BiLSTMs require significantly less memory to train compared to unidirectional LSTMs.

Correct Answer:

c) BiLSTMs can capture contextual information from both past and future time steps in a sequence.

19. You're training a deep neural network on a dataset of images. You notice that the network performs very well on the training data but struggles to generalize to unseen images in the validation set. You suspect the network is memorizing the training data instead of learning generalizable features. [1 mark]

Question:

Which of the following techniques should you apply during training to address this issue, and why?

a) Increase the learning rate, to help the network find a better minimum.

b) Add more layers to the network, to increase its capacity.

c) Randomly deactivate neurons in fully connected layers during training, to prevent the network from relying on specific features.

d) None of these.

Correct Answer:

c) Randomly deactivate neurons in fully connected layers during training, to prevent the network from relying on specific features.

20. You're designing a Convolutional Neural Network (CNN) to process high-resolution images. You need to reduce the spatial dimensions of the feature maps to decrease computational load and make the network more robust to small variations in the input. [1 mark]

Question:

Which of the following operations should you apply to the feature maps, and why?

a) Increase the number of convolutional filters, to capture more detailed features.

b) Apply a pooling layer, to summarize local regions and reduce spatial dimensions.

c) Add more fully connected layers, to increase the network's capacity.

d) Use a larger kernel size in the convolutional layers, to increase the receptive field.

Correct Answer:

b) Apply a pooling layer, to summarize local regions and reduce spatial dimensions.

21. You're building a neural network to restore damaged or low-resolution images. You need to reconstruct the original image with high fidelity, preserving both coarse structures and fine details. [1 mark]

Question 2:

Which of the following architectural choices would be most beneficial for your network, and why?

a) A purely convolutional network with no upsampling.

b) An encoder-decoder architecture with skip connections, to combine high-level and low-level features.

c) A network with only pooling layers, to simplify the image.

d) A network with many fully connected layers, to capture global features.

Correct Answer:

b) An encoder-decoder architecture with skip connections, to combine high-level and low-level features.

22. Code Snippet: **[2 marks]**

Python

```
Unset
import torch.nn as nn



class MyCNN(nn.Module):

    def __init__(self, num_classes=10):

        super(MyCNN, self).__init__()

           self.conv1 = nn.Conv2d(in_channels=3, out_channels=16, kernel_size=3, stride=1, padding=1)

        self.relu = nn.ReLU()

        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)

          self.conv2 = nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3, stride=1, padding=1)

         self.fc = nn.Linear(32 * 16 * 16, num_classes) # Assuming input image size leads to 16x16 after pooling



    def forward(self, x):

        x = self.pool(self.relu(self.conv1(x)))

        x = self.pool(self.relu(self.conv2(x)))

        x = x.view(x.size(0), -1)

        x = self.fc(x)

        return x
```

Question:

Given the PyTorch code snippet above, which of the following statements is TRUE regarding the `MyCNN` architecture?

a) The input images must be of size 32x32 with 1 channel.

b) The first convolutional layer (`conv1`) will reduce the spatial dimensions of the input.

c) After the second pooling layer (`pool`), the feature maps will have dimensions of 16x16 with 32 channels.

d) The fully connected layer (`fc`) will receive input of size 32x32.

Correct Answer:

c) After the second pooling layer (`pool`), the feature maps will have dimensions of 16x16 with 32 channels.

Explanation:

- **Input Channels:** The first convolutional layer (`conv1`) takes `in_channels=3`, indicating the input images have 3 channels (e.g., RGB).
- **Spatial Dimensions:**
    - The first convolution has `padding=1`, so it preserves the input spatial dimensions.
    - The first `MaxPool2d` reduces the spatial dimensions by a factor of 2.
    - The second convolution also preserves spatial dimensions due to padding.
    - The second `MaxPool2d` reduces the spatial dimensions by a factor of 2 again.
    - The second conv layer has `out_channels=32`.
- **Fully Connected Layer:** The fully connected layer receives a flattened version of the feature maps, not a 3D tensor. The 16x16 is based on the assumption that the input image size and pooling layers result in that size.

Why Other Answers Are Incorrect:

- **a) Input Image Size:** The input channels are 3, but the spatial dimensions are not specified, only that they result in 16x16 after two pooling layers.
- **b) First Convolution:** The padding ensures that the spatial dimensions are not reduced by the first convolution.
- **d) Fully Connected Input:** The fully connected layer receives a flattened tensor, not a 3D tensor of 32x32.

23. Which of the following statements are TRUE regarding Convolutional Neural Networks (CNNs)? Select all that apply. [1 mark]

a) Pooling layers introduce non-linearity to the network.

b) Convolutional layers learn translation-invariant features.

c) Exploding gradient problem never happens in CNN.

d) A CNN model may be required to implement an LSTM.

e) A larger kernel size always leads to better performance.

f) Strided convolutions can be used to perform upsampling.

Correct Answers:

- b) Convolutional layers learn translation-invariant features.
- c) Exploding gradient problem never happens in CNN.

24. Consider a fully connected neural network with the following architecture: [1 mark]

- **Input layer: 10 neurons**
- **Hidden layer: 5 neurons**
- **Output layer: 3 neurons**

Which of the following statements are TRUE regarding the number of parameters (weights and biases) in this network? Select all that apply.

a) The total number of weights between the input and hidden layer is 50.

b) The total number of biases in the hidden layer is 10. c) The total number of weights between the hidden and output layer is 15. d) The total number of biases in the output layer is 3. e) The total number of parameters in the network is 73. f) The total number of weights in the network is 65.

Correct Answers:

- a) The total number of weights between the input and hidden layer is 50.
- c) The total number of weights between the hidden and output layer is 15.
- d) The total number of biases in the output layer is 3.
- e) The total number of parameters in the network is 73.
- f) The total number of weights in the network is 65.

Explanation:

- **Input to Hidden Weights:** 10 neurons (input) * 5 neurons (hidden) = 50 weights
- **Hidden Biases:** 5 neurons (hidden) = 5 biases
- **Hidden to Output Weights:** 5 neurons (hidden) * 3 neurons (output) = 15 weights
- **Output Biases:** 3 neurons (output) = 3 biases
- **Total Weights:** 50 (input-hidden) + 15 (hidden-output) = 65 weights
- **Total Biases:** 5 (hidden) + 3 (output) = 8 biases
- **Total Parameters:** 65 (weights) + 8 (biases) = 73 parameters

Why Other Answers Are Incorrect:

- b) The hidden layer has 5 bias neurons, not 10.

25. Code Snippet: **[2 marks]**

Python

```
Unset
import torch

import torch.nn as nn



class SimpleAutoencoder(nn.Module):

    def __init__(self, input_dim, hidden_dim):

        super(SimpleAutoencoder, self).__init__()

        self.encoder = nn.Linear(input_dim, hidden_dim)

        self.decoder = nn.Linear(hidden_dim, input_dim)



    def forward(self, x):

        encoded = torch.relu(self.encoder(x))

        decoded = torch.sigmoid(self.decoder(encoded))
```

```
        return decoded


input_dim = 64

hidden_dim = 32

autoencoder = SimpleAutoencoder(input_dim, hidden_dim)

input_tensor = torch.randn(10, input_dim) # Batch of 10 samples

output_tensor = autoencoder(input_tensor)
```

Question:

Based on the provided PyTorch code snippet, which of the following statements are TRUE?

a) The autoencoder is designed for dimensionality reduction. b) The encoder uses a sigmoid activation function. c) The decoder uses a ReLU activation function. d) The hidden layer has 32 neurons. e) The input and output tensors have the same dimensions. f) The autoencoder is an overcomplete autoencoder.

Correct Answers:

- a) The autoencoder is designed for dimensionality reduction.
- d) The hidden layer has 32 neurons.
- e) The input and output tensors have the same dimensions.

Explanation:

- **a) Dimensionality Reduction:** The `hidden_dim` (32) is less than the `input_dim` (64), indicating a reduction in dimensionality, which is a core purpose of autoencoders.
- **d) Hidden Layer Neurons:** The `hidden_dim` parameter is set to 32, defining the number of neurons in the hidden layer.
- **e) Input/Output Dimensions:** The `input_dim` is used for both the encoder's input and the decoder's output, ensuring they have the same dimensions

26. DenseNet relies on dense connectivity where each layer receives inputs from all previous layers. What is the key downside of this design in very deep networks? [1 mark]

Options:

a) Increased computational cost due to multiple connections.
b) Gradient vanishing issues become more severe as depth increases.
c) Excessive feature reuse can lead to redundant representations.
d) Skip connections in reduces the number of parameters leading to underfitting

Correct Answers:

- a) Increased memory and computational cost due to multiple connections.
- c) Excessive feature reuse can lead to redundant or less discriminative representations.

27. Which of the following statements about R-CNN family architectures are TRUE?

[1 mark]

A) Faster R-CNN is always less accurate than R-CNN but offers significantly improved speed

B) Faster R-CNN uses a Region Proposal Network (RPN) instead of a selective search

C) Fast R-CNN still relies on external methods like selective search for generating region proposals

D) None of the others

Correct Answers:

- B) Faster R-CNN uses a Region Proposal Network (RPN) instead of a selective search
- C) Fast R-CNN still relies on external methods like selective search for generating region proposals

28. Pooling layers are commonly used in Convolutional Neural Networks (CNNs) to reduce the spatial dimensions of feature maps.

Which of the following are true about pooling layers? Select all that apply. [1 mark]

a) Max pooling selects the maximum value in a local region, helping extract dominant features.
a) Average pooling takes the average of values in a local region, retaining more spatial information.
b) Pooling layers introduce learnable parameters that are updated during training.
c) Max pooling enhances the network's ability to detect small variations in the input.
d) Pooling layers are always necessary in deep learning models.

Correct Answers:

- (a) Max pooling selects the maximum value in a local region, helping extract dominant features.
- (b) Average pooling takes the average of values in a local region, retaining more spatial information.

29. Which of the following statements about autoencoders are TRUE? [1 mark]

Options:

a) Autoencoders can be used for dimensionality reduction by learning compact representations of input data.
b) An undercomplete autoencoder has a latent layer with fewer neurons than the input dimension.
c) Overcomplete autoencoders always perform better than undercomplete autoencoders in feature learning.
e) The decoder in an autoencoder typically has more parameters than the encoder.

Correct Answers:

- (a)Autoencoders can be used for dimensionality reduction by learning compact representations of input data.
- b) An undercomplete autoencoder has a bottleneck layer with fewer neurons than the input dimension.

30. Match the following regularization techniques with the appropriate primary effects [1 mark]

Regularization techniques:

- I) L1 Regularization
- II) L2 Regularization
- III) Dropout

- IV) Early Stopping

Primary Effects:

- 1. Prevents neurons from co-adapting by randomly deactivating them during training.
- 2. Reduces overfitting by forcing some weights to become exactly zero, leading to sparse models.
- 3. Penalizes large weight values to prevent overfitting but does not enforce sparsity.
- 4. Stops training when validation loss stops improving to prevent overfitting.

Which of the following options is correct:

A. I) → 2, II) → 1, III) → 4, IV) → 3
B. I) → 3, II) → 4, III) → 1, IV) → 2
C. I) → 2, II) → 3, III) → 1, IV) → 4
D. I) → 1, II) → 3, III) → 4, IV) → 2

Correct Answer: Option C

- I) → 2. Reduces overfitting by forcing some weights to become exactly zero, leading to sparse models.
- II) → 3. Penalizes large weight values to prevent overfitting but does not enforce sparsity.
- III) → 1. Prevents neurons from co-adapting by randomly deactivating them during training.
- IV) → 4. Stops training when validation loss stops improving to prevent overfitting.

31. Which of the following statements BEST describes the Universal Approximation Theorem in the context of neural networks? [1 mark]

a) Any continuous function can be perfectly approximated by a neural network with a finite number of layers, regardless of the activation functions used.

b) A neural network with a single hidden layer and a sufficient number of neurons can approximate any continuous function to an arbitrary degree of accuracy, provided that a suitable activation function is used.

c) Deep neural networks with multiple hidden layers are strictly necessary to approximate complex functions; a single hidden layer is insufficient for most practical applications.

d) The Universal Approximation Theorem guarantees that a neural network will always find the *best* possible approximation for any given function.

Correct Answer:

b) A neural network with a single hidden layer and a sufficient number of neurons can approximate any continuous function to an arbitrary degree of accuracy, provided that a suitable activation function is used.

32. A team of researchers is developing a system for real-time speech recognition. They are considering different sequence model architectures for the acoustic modeling component, which maps audio features to a sequence of characters. Their primary concerns are: [1 mark]

1. **Accuracy:** The model needs to accurately transcribe speech, even in noisy environments and with varying speaking styles.

2. **Latency:** The system must operate in real-time, meaning the model needs to process audio quickly, introducing minimal delay.

3. **Handling Variable-Length Input:** Spoken utterances have varying lengths, so the model must be able to handle sequences of different durations.

Which of the following architecture(s) can be used for meeting these requirements?

a) A deep, stacked LSTM network with a sequence-to-sequence architecture.

b) A multilayer perceptron

c) A bidirectional LSTM network.

d) A GRU

Correct Answers:

- a) A deep, stacked LSTM network with a sequence-to-sequence architecture.
- d) A GRU