**Department of Computer Science and Engineering**
**Indian Institute of Technology Jodhpur**
**CSL 4020: Deep Learning**
**Major Examination**

**Total marks: 60** **Solution Set** **Time: 3 hours**

---

**Instructions:**
No queries will be answered during the exam. If you find anything unclear/incorrect in any question, make a reasonable assumption and proceed. The total marks for each question are indicated beside or below the question inside the brackets.

Mark all the answers in the OMR sheet only.

You may do the rough work on additional white pages. Do not write anything other than your name and roll number on the OMR sheet.

Every question may have multiple correct options. If so, you have to mark all the correct options.

---

**Answer all the questions.**

1. In Adam optimization, bias correction is applied to: **[1 mark]**
    a. Accumulated learning rate
    **b. Accumulated squared gradient**
    **c. Accumulated gradient**
    d. None of the others
2. Which technique introduces stochasticity during training? **[1 mark]**
    a. BatchNorm
    **b. Dropout**
    c. Early Stopping
    d. None of the others
3. A 1x1 convolution in CNNs is primarily used to: **[1 mark]**
    a. Increase spatial dimensions
    **b. Reduce channel depth**
    c. Normalize input data
    d. Replace pooling
    e. None of the others
4. In a Variational Autoencoder (VAE), what is the purpose of the reparameterization trick?

**[1 mark]**

a. Improving the generator's ability to create sharp images.
**b. Facilitating gradient flow through a stochastic layer during training**
c. None of the others
d. Enforcing sparsity in the learned features

5. What is the main idea behind transfer learning? **[1 mark]**
   a. Train a model from scratch for each task
   **b. Use knowledge from one task to improve the performance of the model on another related task**
   c. Increase the model size for improving performance
   d. Reduce model size for improving performance
   e. None of the others
6. Which type of domain adaptation assumes a gradual change in the distribution of data? **[1 mark]**
   **a. Concept drift**
   b. Zero-shot learning
   c. Few-shot learning
   d. None of the others
7. In a Transformer architecture, self-attention helps the model to: **[1 mark]**
   a. Cluster the inputs before passing them to the decoder
   **b. Utilize positional information to correlate tokens**
   c. Correlate a token only with itself
   d. None of the others
8. Which of the following statements about RNNs is/are correct? **[1 mark]**
   **a. They process sequences one token at a time**
   b. They require each input sequence to have the same length
   **c. They share parameters across time steps**
   d. They suffer from the vanishing gradient problem but do not suffer from the exploding gradient problem
   e. None of the others
9. Which of the following mechanisms helps LSTMs handle long-term dependencies better than standard RNNs? **[1 mark]**
   **a. Cell state propagation**
   **b. Gating mechanisms**

**c.** Directly using the softmax activation in the hidden state

d. Bi-directional connection

e. None of the others

10. Which of the following are generally true about Conditional GANs (CGANs)? **[1 mark]**

**a. CGANs take both noise and conditional labels as input.**

b. CGANs eliminate the need for a discriminator.

**c. CGANs generate outputs that are conditioned on specific classes.**

**d. CGANs always require labeled datasets for training.**

e. None of the others

**Note: Marks will be awarded for selecting either options a and c, or a, c, and d.**

11. For a transposed convolution with stride=2, padding=1, and input size 5x5, output size is: **[1 mark]**

a. 9x9

b. 10x10

c. 11x11

d. None of the others

**Note: As the kernel size is missing, none of the options are correct. Bonus Marks will be awarded for attempting this question.**

12. Regarding L1 and L2 regularization in neural networks, which of the following statements are correct? **[1 mark]**

**a. L1 regularization encourages sparsity of the parameters**

**b. L2 regularization shrinks the values of the parameters**

c. Both L1 and L2 produce identical gradient update rules

d. None of the others

13. In training deep CNNs, which of the following practices generally improves convergence and stability? **[1 mark]**

**a. Using batch normalization**

**b. Employing learning rate scheduling**

c. Using a larger kernel size

d. Using very wide, fully connected layers

e. None of the others

14. Which of the following factors do not generally contribute to mode collapse in GANs?

**[1 mark]**

a. Choice of the loss function of the discriminator
**b. Choice of the loss function of the generator**
**c. The choice of the optimizer**
d. The distribution of the real data
e. None of the others

15. Which of the following improvements does WGAN-GP introduce over WGAN? **[1 mark]**

**a. Gradient penalty instead of weight clipping.**
b. A separate encoder to stabilize training.
**c. More stable discriminator training.**
**d. The removal of batch normalization in the generator.**
e. None of the others

**Note: Marks will be awarded for selecting either options a, c, and d or options a and c.**

16. Which of the following are key characteristics of CycleGAN? **[1 mark]**

**a. It can perform unpaired image-to-image translation**
b. It requires a fixed latent space dimension for both domains
**c. It introduces a cycle consistency loss to enforce the reversibility of the mapping**
d. It uses one generator and two discriminators
e. None of the others

17. Which of the following statements is/are correct about VAEs? **[1 mark]**

**a. VAEs explicitly learn a probabilistic latent space**
b. VAEs always generate sharper images than GANs
**c. VAEs use KL divergence as a loss term**
d. Sampling from the latent space is performed only during inference and not during training in VAEs
e. None of the others

18. Which of the following statements about Transformer models is/are correct? **[1 mark]**

**a. Transformers allow parallel processing of input tokens**
b. Transformers use masked self-attention in the encoder and decoder

c. Padding tokens are never required in Transformers

d. Transformers always use mean-squared error as a loss function

e. None of the others

19. Find out the correct statements about Transformers **[1 mark]**

a. Transformers never require fully connected layers

b. Vanishing gradient problems never occur for transformers

**c. Positional encoding is performed using the same equation for all tokens**

**d. Apart from self-attention, transformers may also require cross-attention**

e. None of the others

20. Code Snippet **[1 mark]**

```python
import torch

import torch.nn as nn

x = torch.rand(1, 10, 512) # batch size = 1, seq_len = 10, embedding dim = 512

attn = nn.MultiheadAttention(embed_dim=512, num_heads=8)

out, _ = attn(x, x, x)

print(out.shape)
```

What is the shape of the “out” variable?

a. (1, 512, 10)

b. (10, 1, 512)

**c. (1, 10, 512)**

d. None of the others

21. Code Snippet **[1 mark]**

```python
class TransferModel(nn.Module):

	def __init__(self, base_model):

		super().__init__()
```

```
        self.base = base_model
        self.classifier = nn.Linear(512, 10)
    def forward(self, x):
        with torch.no_grad():
            x = self.base(x)
        return self.classifier(x)
```

What is the effect of torch.no_grad() in the above model?

a. Enables weight updates for the base model
**b. Prevents backpropagation through the base model**
c. None of the others
d. Freezes the classifier

22. Match the following types of Autoencoders with the appropriate characteristics (find the best match) **[1 mark]**

| | | | |
|---|---|---|---|
| A. | Overcomplete | I. | Can be used for dimensionality reduction |
| B. | Denoising | II. | Reduces noise from input data |
| C. | Variational | III. | May not be able to extract salient features |
| D. | Undercomplete | IV. | Can be used as a generative model |

**Options:**

a. A -> (II) , B -> (III), C -> (IV), D -> (I)
**b. A -> (III) , B -> (II), C -> (IV), D -> (I)**
c. A -> (III) , B -> (IV), C -> (II), D -> (I)
d. A -> (III) , B -> (II), C -> (I), D -> (IV)
e. None of the others

23. Find the best match among the following **[1 mark]**

| | | | |
|---|---|---|---|
| A. | DenseNet | I. | Leverages pre–trained models to adapt efficiently to new tasks with limited data. |
| B. | Depthwise Separable | II. | Decomposes standard convolutions into |

| | | | |
|---|---|---|---|
| | Convolution | | separate spatial and channel–wise operations, reducing computation. |
| C. | CNN–based Transfer Learning | III. | Utilizes dense connectivity to promote feature reuse and improved gradient flow. |
| D. | Residual Connections | IV. | Adds shortcut pathways that bypass one or more layers, reducing the vanishing gradient problem. |

**Options:**

a. A -> (IV) , B -> (V), C -> (I), D -> (III)

b. A -> (IV) , B -> (III), C -> (I), D -> (II)

c. A -> (III) , B -> (V), C -> (I), D -> (IV)

**d. A -> (III) , B -> (II), C -> (I), D -> (IV)**

e. None of the others

24. Match the following concepts with their corresponding descriptions: **[1 mark]**

| | | | |
|---|---|---|---|
| A. | Fine-tuning | I. | Prevents weight updates for specific parts of a model |
| B. | Freezing layers | II. | Computes attention between all positions in a sequence |
| C. | Self-attention | III. | Difference in data distribution between source and target domains |
| D. | Domain shift | IV. | Adapting a pre-trained model to a specific task with full training |

**Options:**

a. A -> (I) , B -> (III), C -> (IV), D -> (II)

b. A -> (II) , B -> (IV), C -> (III), D -> (I)

c. A -> (II) , B -> (III), C -> (IV), D -> (I)

d. A -> (I) , B -> (IV), C -> (III), D -> (II)

**e. None of the others**

25. In domain adaptation, a model is trained for 20 epochs with an initial learning rate of 0.01. The learning rate decays by a factor of 0.1 every 10 epochs. What is the learning rate at epoch 22 (Epoch count starts with 0)? **[1 mark]**

**Options:**

a. 0.001

b. **0.0001**
c. 0.002
d. 0.0002
e. None of the others

**Note: Marks will be awarded for attempting this question.**

**Solution:**

- Initial learning rate = 0.01
- Decay factor = 0.1
- Decay period = 10 epochs
- Number of epochs = 20
- Epochs 0 to 9: The learning rate remains at the initial value of 0.01.
- Epochs 10 to 19: The learning rate decays by a factor of 0.1 after epoch 10. So, after epoch 10, the learning rate will be: (0.01 \times 0.1) = 0.001
- Since the learning rate decays again after another 10 epochs (i.e., at epoch 20), the learning rate will be (0.001 \times 0.1) = **0.0001**. Will remain the same till Epoch 29.

26. The self-attention mechanism in Transformers operates using three different vectors: **(I)**, which is responsible for selecting relevant information from other elements in the sequence **(II)**, which holds stored information that the model can access and **(III)**, which determines how much information is retrieved from each element based on a learned weighting mechanism. The attention score is computed as a scaled dot-product between **(I)** and **(II)**, followed by a softmax operation to normalize the weights before applying them to **(III)**. Unlike traditional RNN-based models, the Transformer leverages parallel computation and long-range dependency capture through this self-attention mechanism. **[1 mark]**
    a. (I) Key (II) Value (III) Query
    b. **(I) Query** **(II) Key** **(III) Value**
    c. (I) Value (II) Query (III) Key
    d. (I) Query (II) Value (III) Key
    e. None of the others
27. Gradient descent optimizers play a crucial role in training deep neural networks. **(I)** is one of the simplest optimization methods, updating weights by computing gradients

concerning a fixed learning rate. However, it struggles with adapting learning rates dynamically for different parameters. To address this, **(II)** introduces an adaptive learning rate by accumulating past squared gradients but suffers from a continuously decreasing step size, which can slow down convergence in later stages of training. Unlike **(II)** , **(III)** leverages adaptive moment estimation, combining both first-order momentum (exponentially decaying average of past gradients) and second-order momentum (exponentially decaying average of squared gradients) to achieve a balance between fast convergence and stability. Because of these properties, **(III)** is one of the most widely used optimizers in modern deep learning applications. **[1 mark]**

a. **(I) SGD (II) Adagrad (III) Adam**
b. (I) Adam (II) SGD (III) Adagrad
c. (I) Adagrad (II) Adam (III) SGD
d. (I) SGD (II) Adam (III) Adagrad
e. None of the others

28. Both **(I)** and **(II)** are convolution-based architectures used for feature extraction in deep learning, but they differ in computational complexity and design. While **(I)** applies a standard convolutional filter that operates across all input channels simultaneously, **(II)** decomposes this operation into two separate steps: first, a depthwise convolution that processes each channel independently, followed by a pointwise convolution to mix information across channels. This decomposition drastically reduces the number of parameters and FLOPs compared to **(I)** while maintaining comparable performance in some cases. However, **(II)** can sometimes suffer from **(III)** due to its reduced ability to fully capture cross-channel dependencies in early layers unless additional processing is applied. **[1 mark]**

a. (I) CNN (II) Depthwise Separable Convolution (III) vanishing gradient
b. (I) Depthwise Separable Convolution (II) CNN (III) vanishing gradient
c. **(I) CNN (II) Depthwise Separable Convolution (III) reduced representational capacity**
d. (I) Depthwise Separable Convolution (II) CNN (III) increased parameter count
e. None of the others

29. A generative model is evaluated using the Inception Score (IS). Two sets of images are generated: **[1 mark]**

**Set 1:** 1000 images with high quality and diversity.

**Set 2:** 1000 images with high quality but low diversity (i.e., many images are very similar).

Which of the following Inception Score (IS) values is most plausible for each set?

a. **Set 1: IS = 12, Set 2: IS = 3**
b. Set 1: IS = 5, Set 2: IS = 7
c. Set 1: IS = 2, Set 2: IS = 10
d. Set 1: IS = 8, Set 2: IS = 8
e. None of the others

30. Match the following terms with their key characteristics. **[1 mark]**

| Term | Description |
|---|---|
| 1. RNN | A. Relies on parallel processing of input sequences using self-attention, enabling effective capture of long-range dependencies but lacking inherent mechanisms to track positional information. |
| 2. LSTM | B. Addresses the vanishing gradient problem with gating mechanisms, allowing for the processing of longer sequences by controlling the flow of information through memory cells. |
| 3. GRU | C. Offers a simplified alternative to LSTMs, with reduced complexity due to fewer gating mechanisms while achieving comparable performance in many sequence modeling tasks. |
| 4. Transformer | D. Processes sequential data iteratively, updating a hidden state for each element, which can lead to difficulties in capturing long-range dependencies and susceptibility to the vanishing gradient problem. |
| 5. ViT | E. Adapts a non-sequential architecture to image analysis by treating image patches as a sequence, leveraging self-attention to achieve strong performance in computer vision tasks. |

Select the correct matching of Term to Description:

**Options:**

a. **1-D, 2-B, 3-C, 4-A, 5-E**
b. 1-A, 2-B, 3-C, 4-D, 5-E
c. 1-D, 2-C, 3-B, 4-A, 5-E

d. 1-E, 2-B, 3-C, 4-A, 5-D

e. None of the others

31. Consider a convolutional layer in a CNN with the following specifications: **[2 marks]**

- Input feature map size: (32 × 32 × 3) (Height × Width × Channels)
- Number of filters: 16
- Filter size: (5 × 5)
- Stride: 1
- Padding: 'same'

I. What will be the size of the output feature map after applying this convolutional layer?

II. Compute the total number of trainable parameters (weights + biases) in this convolutional layer.

**Options:**

**a. Size of the output feature map: (32 × 32 × 16), Parameters = 1216**

b. Size of the output feature map: (28 × 28 × 16), Parameters = 1216

c. Size of the output feature map: (32 × 32 × 16), Parameters = 12176

d. Size of the output feature map: (28 × 28 × 16), Parameters = 12176

e. None of the others

**Solution:**

- The output feature map size is:
  - Height × Width = 32 × 32 (since padding = 'same')
  - Channels = 16 (equal to the number of filters)
  - Output size = **(32 × 32 × 16)**
- Weights per filter = 5 × 5 × 3 = 75
- Plus 1 bias term per filter. So, total parameters per filter = 76
- For 16 filters, total parameters = 16 × 76 = **1216**

32. Consider a standard convolution layer with an input of 128 channels, an output of 256 channels, and a 3×3 kernel. **[2 marks]**

I. Compute the total number of parameters in a standard convolution.

II. For a depthwise separable convolution that performs a depthwise convolution (with a 3×3 kernel applied per input channel) followed by a pointwise convolution

(1×1 kernel mapping 128 channels to 256 channels), compute the total parameters.

III. What is the approximate percentage reduction in parameter count when using the depthwise separable convolution compared to the standard convolution?

**Options:**

a. **Total number of parameters in the standard convolution = 295,168, total number of parameters in the depthwise separable convolution = 34,304, reduction = 88.4%**
b. Total number of parameters in the standard convolution = 294,912, total number of parameters in the depthwise separable convolution = 33,024, reduction = 88.8%
c. Total number of parameters in the standard convolution = 294,912, total number of parameters in the depthwise separable convolution = 34,304, reduction = 88.3%
d. Total number of parameters in the standard convolution = 295,168, total number of parameters in the depthwise separable convolution = 33,024, reduction = 88.8%
e. None of the others

**Note: Marks will be awarded if option a or e is selected.**

**Solution:**

- Input channels = 128
- Output channels = 256
- Kernel size = 3×3
- Total parameters in standard convolution: (3×3×128)×256+256(bias) = **295,168**
- Depthwise: 3×3×128+128(bias) = 1,280
- Pointwise: 1×1×128×256 + 256(bias) = 33,024
- Total parameters in depthwise separable convolution: 1,280 + 33,024 = **34,304**
- Reduction = (295,168 - 34,304)×100 / 295,168 = **88.4%**

33. For a simple RNN, you have:

- $W_h$ = 0.8 (recurrent weight)
- $h_{t-1}$ = 0.5 (previous hidden state)

- $W_x$ = 1.2 (input weight)
- $x_t$ = 0.7 (current input)
- b = -0.3 (bias)
- Use tanh(x) as the activation function

Compute $h_t$ rounded to 3 decimal places. **[2 marks]**

**Options:**

a. None of the others
b. 0.845
**c. 0.735**
d. 0.629
e. 0.696

**Solution:**

- $h_t = \tanh(W_h h_{t-1} + W_x x_t + b) = \tanh(0.8*0.5+1.2*0.7-0.3) = \tanh(0.94) =$ **0.735**

34. In a Standard GAN, suppose the discriminator assigns probabilities D(G(z))=[0.1,0.4,0.6,0.9] to four generated samples. Compute $L_G$ (Generator Loss) rounded to 3 decimal places. (Use base e for log term) **[2 marks]**

**Options:**

**a. None of the others**
b. 0.332
c. 6.835
d. 1.211
e. 2.824

**Solution:**

- $L_G = -\frac{1}{n}\sum_{i=1}^{n} log_e(D(G(z_i)))$
- $L_G$ = -0.25* (ln(0.1)+ln(0.4)+ln(0.6)+ln(0.9)) = **0.959**

35. In a Variational Autoencoder (VAE), the approximate posterior $q(z|x)$ is parameterized by a mean μ and variance $\sigma^2$ for each latent dimension, while the prior p(z) is assumed to follow a standard normal distribution (mean = 0, variance = 1). **[2 marks]**

For a 2D latent space, given the following posterior parameters:

Dimension 1: $\mu_1=0$, $\sigma_1^2=0.3$

Dimension 2: $\mu_2=-0.2$, $\sigma_2^2=0.7$

Compute the KL divergence between $q(z|x)$ and $p(z)$, rounded to 3 decimal places.

**Options:**

a. 0.141
b. 0.242
**c. None of the others**
d. -0.141
e. -0.242

**Solution:**

- KL divergence between a multivariate Normal distribution and a standard normal distribution:
  - $= \frac{1}{2}\sum_{i=1}^{k}\left(\sigma_i^2 + \mu_i^2 - 1 - ln(\sigma_i^2)\right)$. Here, $k = 2$.
  - $= \frac{1}{2}\left(\left(0.3 + 0^2 - 1 - ln(0.3)\right) + \left(0.7 + (-0.2)^2 - 1 - ln(0.7)\right)\right)$
  - $= \frac{1}{2}\left(\left(0.3 + 0^2 - 1 + 1.204\right) + \left(0.7 + (-0.2)^2 - 1 + 0.357\right)\right)$
  - $= 0.3$

36. Both **(I)** and **(II)** address the instability of standard GANs by replacing the Jensen-Shannon divergence with the Wasserstein distance, improving gradient flow. However, **(I)** enforces the Lipschitz constraint by clipping the critic's weights to a fixed range, which can lead to optimization issues and poor convergence. To mitigate this, **(II)** removes weight clipping and instead applies a gradient penalty term, ensuring the Lipschitz condition is approximately satisfied without overly restricting the critic's capacity. Unlike **(I)**, where weight clipping can cause the critic to become too constrained, **(II)** allows the critic to learn a more expressive function while still maintaining stability. Furthermore, (II) computes the gradient penalty on **(III)**, while **(I)** does not include such a step. Theoretically, **(II)** is more stable and achieves better empirical results in high-dimensional spaces, whereas **(I)** may suffer from an overly simplistic function approximation if the weight clipping is too aggressive.**[2 marks]**

a. **(I) WGAN (II) WGAN-GP (III) the interpolated points between real and synthetic samples**
b. (I) VAE (II) WGAN (III) the interpolated points between real and synthetic samples
c. (I) WGAN-GP (II) WGAN (III) the synthetic samples
d. (I) WGAN (II) VAE (III) the real samples
e. None of the others

37. Suppose a WGAN critic is trained with a batch size of **64**, and during training:
    - The gradient penalty term is computed using a weighting factor ( \lambda = 10 ).
    - The interpolated sample gradient norm is measured as **0.05** for a batch sample.
    - The critic loss for that sample (without penalty) is **-0.8**.

    What is the total loss applied to update the critic on that sample? **[2 marks]**

    **Options:**

    a. -0.8
    b. -0.3
    c. 4.5
    d. 9.2
    e. **None of the others**

    **Solution:**

    - From the question:
        - ( L_{\text{critic}} = -0.8 )
        - ( \lambda = 10 )
        - Gradient norm for the sample is 0.05.
    - Now, compute the gradient penalty term:
        - [ (0.05 - 1)^2 = (-0.95)^2 = 0.9025 ]
    - Multiply by ( \lambda ):
        - [ 10 \times 0.9025 = 9.025 ]
    - Compute Total Loss
        - [ L = -0.8 + 9.025 = 8.225 ]

38. Both **(I)** and **(II)** are widely used for generative modeling but fundamentally differ in their training objectives and the nature of the distributions they learn. While **(I)**

optimizes a variational lower bound on the data likelihood using an encoder-decoder framework, **__(II)__** employs an adversarial approach where a generator competes against a discriminator. Unlike **__(II)__** , which directly maps a noise vector to the data space, **__(I)__** explicitly models the latent space using a learned posterior distribution. A key issue in **__(I)__** is **__(III)__** , which occurs when the learned latent space collapses, leading to poor generation diversity. On the other hand, **__(II)__** often suffers from **__(IV)__** , where the generator produces limited variations of samples rather than fully capturing the data distribution. **[2 marks]**

a. (I) GAN (II) VAE (III) mode collapse (IV) discrete latent space
**b. (I) VAE (II) GAN (III) latent space with overlapping class boundaries (IV) mode collapse**
c. (I) GAN (II) VAE (III) continuous latent space (IV) mode collapse
d. (I) VAE (II) GAN (III) latent space with codebook (IV) posterior collapse
e. None of the others

39. Consider the following PyTorch code snippet, which is part of a VAE implementation:

**[2 marks]**

Python

```
class VAE(nn.Module):
    def __init__(self, encoder, decoder):
        super(VAE, self).__init__()
        self.encoder = encoder
        self.decoder = decoder
    def reparameterize(self, mu, logvar):
        std = torch.exp(0.5 * logvar)
        eps = torch.randn_like(std)
        z = mu + eps * std
```

```
        return z
    def forward(self, x):
        mu, logvar = self.encoder(x)
        z = self.reparameterize(mu, logvar)
        x_recon = self.decoder(z)
        return x_recon, mu, logvar
def loss_function(recon_x, x, mu, logvar):
    # Reconstruction loss
    recon_loss = F.mse_loss(recon_x, x, reduction='sum')
    # KL divergence loss
    kl_div = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())
    return recon_loss + kl_div
```

Which of the following statements is/are correct, based on this code?

a. **The reparameterize function returns a sample z from a Gaussian distribution with mean mu and variance logvar.**
b. The encoder in the VAE class is expected to return the mean (mu) and the standard deviation (logvar) of the latent distribution.
c. **The kl_div variable in the loss_function calculates the Kullback-Leibler divergence between the learned latent distribution and a standard normal distribution.**
d. **The forward function takes an input x, encodes it into a latent space, samples from that latent space, decodes the sample, and returns the reconstructed input along with mu and logvar.**
e. The reconstruction loss in loss_function is calculated using binary cross-entropy.

**Note: Marks will be awarded for selecting either options a, c, and d or options c and d.**

40. Consider two generative models, Model A and Model B, evaluated using Inception Score (IS) and Fréchet Inception Distance (FID). Assume the following:
    - Model A generates images with high quality and diversity.
    - Model B generates images with high quality but low diversity.

Based on this information, which of the following statements is/are most likely to be correct? **[2 marks]**

a. **Model A has a higher Inception Score than Model B, and Model A has a lower FID than Model B.**
b. Model A has a lower Inception Score than Model B, and Model A has a higher FID than Model B.
c. **If Model A has an Inception Score of 10 and Model B has an Inception Score of 8, this indicates that Model A's generated images are of higher quality and diversity than Model B's.**
d. **If Model A has an FID of 50 and Model B has an FID of 100, this indicates that the distribution of images generated by Model A is closer to the distribution of real images than Model B.**
e. **A very low Inception Score (e.g., below 2) combined with a very high FID (e.g., above 200) would suggest that a model generates images that are both of poor quality and significantly different from real images.**

41. A machine learning model is trained to classify emails as either "spam" or "not spam." The training data consists of emails from a corporate email server. Which of the following scenarios best describes a domain adaptation problem? **[2 marks]**
    a. **The model is used to classify spam emails from the same corporate email server six months later, and the characteristics of spam emails have changed slightly.**
    b. **The model is used to classify spam emails from a personal email account (e.g., Gmail) with different email formatting and spam characteristics.**
    c. **The model is used to classify spam emails in a different language (e.g., training on English emails, classifying Spanish emails).**
    d. None of the others

    **Note: Marks will be awarded for selecting any of the options a, b, or c, individually or in combination.**

42. Which of the following statements are true about zero-shot learning? **[2 marks]**
    a. **The model must generalize to unseen classes without having observed any labeled examples of those classes during training.**
    b. **Effective zero-shot learning requires a strong reliance on auxiliary information, such as semantic embeddings or attributes, which may not always be readily available or perfectly aligned with the data.**
    c. Zero-shot learning models are inherently limited to recognizing only those unseen classes that are semantically dissimilar to the seen classes in the training data.
    d. None of the others
    e. Zero-shot learning models usually do not overfit
43. In the Adam optimization algorithm, bias correction is applied to the first and second moment estimates ((m_t) and (v_t)). Consider an Adam optimizer with hyperparameters (\beta_1 = 0.9), (\beta_2 = 0.999), and initial moment estimates (m_0 = 0), (v_0 = 0). At

timestep (t = 3), the uncorrected first moment estimate (m_3) is calculated to be 0.5, and the uncorrected second moment estimate (v_3) is 0.2.

What are the values of the bias-corrected first moment estimate (\hat{m}_3) and the bias-corrected second moment estimate (\hat{v}_3)? **[2 marks]**

**Options:**

a. (\hat{m}_3 = 0.7407), (\hat{v}_3 = 0.2012)
b. (\hat{m}_3 = 0.5555), (\hat{v}_3 = 0.2002)
c. (\hat{m}_3 = 0.5), (\hat{v}_3 = 0.2)
d. (\hat{m}_3 = 0.6555), (\hat{v}_3 = 0.2022)
**e. None of the others**

**Solution:**

- The bias correction formulas for Adam are:
  - [ \hat{m}_t = \frac{m_t}{1 - \beta_1^t} ]
  - [ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} ]
- Given values: ( \beta_1=0.9 ) ( \beta_2=0.999 ) ( m_3=0.5 ) ( v_3=0.2 ) ( t=3 )
- Substitute into the formulas:
  - [ \hat{m}_3 = \frac{0.5}{1 - (0.9)^3} = \frac{0.5}{1 - 0.729} = \frac{0.5}{0.271} \approx **1.845** ]
  - [ \hat{v}_3 = \frac{0.2}{1 - (0.999)^3} = \frac{0.2}{1 - 0.997002999} = \frac{0.2}{0.002997001} \approx **66.733** ]

44. Which of the following statements about Vision Transformers (ViT) are accurate and reflect key challenges or advanced concepts in their design and application? **[2 marks]**
    a. ViT models are inherently translation-invariant due to the patchification of the input image, which allows them to recognize objects regardless of their location in the image.
    b. **Positional embeddings are crucial in ViT to compensate for the lack of inherent sequential processing in the Transformer architecture, and different types of positional embeddings can significantly impact model performance.**
    c. The computational complexity of the self-attention mechanism in ViT scales linearly with the number of image patches, making it highly efficient for high-resolution images.
    d. None of the others
    e. **The basic ViT does not require masked self-attention**
45. A convolutional layer outputs a feature map with dimensions 8x8. This feature map is then passed through a max pooling layer with a filter size of 2x2 and a stride of 2.

    What are the dimensions of the output feature map after the max pooling operation?

    **Options:** **[2 marks]**

a. 2x2
b. **4x4**
c. 6x6
d. 16x16
e. None of the others

**Solution:**

- Input dimensions: 8x8
- Filter size: 2x2
- Stride: 2
- The formula to calculate the output dimensions (width and height) of a max pooling layer is:
    - Output Dimension = (Input Dimension - Filter Size) / Stride + 1
- In this case:
    - Output Width = (8 - 2) / 2 + 1 = 6 / 2 + 1 = 3 + 1 = 4
    - Output Height = (8 - 2) / 2 + 1 = 6 / 2 + 1 = 3 + 1 = 4
- Therefore, the output feature map dimensions are **4x4**.