# 24-789 Intermediate Deep Learning (Spring 2025)
# Midterm Exam

Ryan Wu (Andrew ID: weihuanw)

Thursday April 15, 2025

## 1 Short Answer Questions

(a) How do the underlying mechanisms of diffusion models differ from those of GANs and VAEs? What are the respective advantages and limitations of each approach in terms of stability and efficiency of training, as well as efficiency and output quality of data generation?

**Diffusion Models:**
Diffusion models consist of forward and reverse diffusion processes. The forward step uses a Markov chain process that gradually adds Gaussian noise to the input data. The reverse step trains a neural network to iteratively remove the added noise, effectively reversing the forward process to reconstruct or generate new data samples that resemble the original data. Diffusion models are stable in training and produce high-quality data outputs but suffer from slow, iterative sample generation.

**GANs:**
Generative Adversarial Networks (GANs) consist of two neural networks: the generator and the discriminator, which are trained simultaneously through an adversarial game. The generator learns to transform random noise into synthetic data samples, while the discriminator learns to distinguish between real data and the generator's outputs. This adversarial setup will result in the generator producing realistic output. GANs can generate samples quickly in a single forward pass, but adversarial training can be unstable and prone to issues such as mode collapse.

**VAEs:**
Variational Autoencoders (VAEs) consist of two neural networks: the encoder and the decoder. The encoder maps high-dimensional input data to a low-dimensional probabilistic latent space by predicting the parameters (mean and standard deviation) of a Gaussian distribution. The decoder reconstructs the original data by sampling from this latent distribution using the reparameterization trick. VAEs typically train stably and efficiently, though their reconstructions tend to be blurrier compared to those produced by diffusion models and GANs.

(b) What is the difference between an autoencoder and a variational autoencoder (VAE)? How does the latent space differ between the two types of autoencoders?

The difference between an autoencoder and a variational autoencoder lies in their latent space representation. Conventional autoencoders use a deterministic approach to encode input data into fixed latent vectors, resulting in an unstructured latent space. In contrast, VAEs encode input probabilistically by predicting a distribution (Gaussian) for the latent variables, which creates a smooth, structured latent space.

(c) What are the main limitations of RNNs and LSTMs that the Transformer architecture mitigates?

Transformer architectures mitigate the following limitations of RNNS and LSTMs:

- Enable parallel processing instead of sequential to speed up training.
- Capture long-range dependencies more effectively using the self-attention mechanism.
- Avoid issues like vanishing or exploding gradients inherent in recurrent architectures.

(d) What's the main difference between Transformer's encoder and decoder?

The main difference between Transformer's encoder and decoder is their functional purpose and how they handle input sequences. The encoder uses bidirectional self-attention to process the entire input in parallel and build rich contextual representations. On the other hand, the decoder uses masked (unidirectional) self-attention to generate the output sequence sequentially, ensuring each token is predicted based only on past tokens and the encoder's contextual output.

(e) Explain the role of Generator and Discriminator in Generative Adversarial Networks (GANs). How do GANs generate data at inference time? How does this differ from the approach used by Variational Autoencoders (VAE) to generate new data?

In GANs, the generator transforms random noise into synthetic data samples that mimic the real data distribution. Its goal is to generate data that is as close as possible to real data to fool the discriminator. The discriminator takes real data and the data generated by the generator to learn to distinguish the two and outputs the probability that the given data is real.

At inference time, only the generator is used. It takes a noise vector from a predefined distribution (Gaussian) and maps it directly to the data space to produce synthetic samples in a single pass. On the other hand, VAEs generate data by sampling a learned probabilistic latent space and then reconstructing the data via a decoder.

(f) How does a Bidirectional GAN differ from a vanilla GAN? How does the structure map to the latent space?

The main difference is that a Bidirectional GAN (BiGAN) adds an encoder that maps real data back into the latent space compared to a standard GAN architecture. While a vanilla GAN only has a generator and a discriminator, a BiGAN jointly trains both the generator and the encoder. The BiGAN's discriminator evaluates both the data pair and their latent codes to ensure that the latent space is structured to capture meaningful representations of the real input data.

(g) What is the point of including the KL divergence loss in the objective function while training a VAE? What happens if we assign too much or too little weight on this loss term?

The KL divergence loss regularizes the latent space by encouraging the encoder's output distribution to match a smooth, continuous Gaussian prior. If too much weight is given to KL, the latent space can be over-constrained, leading to posterior collapse where the encoder output becomes identical and uninformative. However, if too little weight is given to KL, the latent space may become unstructured and overfit the training data, resulting in generated samples that fail to generalize.

(h) Explain what happens if you over-train the discriminator in a GAN? What is a way to prevent over-training of the discriminator?

Over-training the discriminator drives its loss toward zero, causing the generator's gradients to vanish and halting its learning (the vanishing gradient problem). Some techniques we can use to prevent over-training the discriminator are: varying training steps, applying regularization, label smoothing, noise injection, early stoppage, etc.

(i) Mode collapse is a possible problem when generating data using GANs. Explain what that is, when it might happen, and how one can fix it.

Mode collapse occurs when the generator produces limited, repetitive outputs regardless of noise inputs. It often happens when the generator finds a few samples that consistently fool the discriminator and thus never explores other modes of the data distribution. Some techniques we can use to address the mode collapse problem are: alternative loss functions (WGAN), apply regularization, minibatch discrimination, feature matching, etc.

(j) What is the role of transposed convolutional layers in a Deep Convolutional Generative Adversarial Network (DCGAN)? How do they enable the generator network to produce high-resolution images?

The transposed convolutional layer (deconvolutional layer) increases the spatial dimensions (height and width) of its input feature map while reducing the number of channels. By stacking multiple transposed convolution layers, the generator performs learned upsampling—gradually expanding a low-resolution latent tensor into a full-size image and synthesizing finer details at each step—enabling generator network to produce high-resolution images.

## 2 Calculation

(a) In this question you are asked to calculate a causal self-attention by hand. The input $x_i$ is the encoding of each word in the sentence, which contains three words:

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \\ 1 & 0.5 \end{bmatrix}.$$

The following weight matrices are applied to the input features to project them into query, key, and value:

$$Q = XW_q, \quad K = XW_k, \quad V = XW_v$$

where:

$$W_q = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix}, \quad W_k = \begin{bmatrix} 1 & 0 \\ -0.5 & 0.5 \end{bmatrix}, \quad W_v = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The attention used here will be standard scaled-dot product attention (with scaling factor $d_k$ set to 1), and is causal attention (as in the Transformer decoder), which means that a token cannot attend to a token that appears after it, e.g. $x_0$ cannot attend to $x_2$ but $x_2$ can attend to $x_0$. The attention here will contain only one head.

Please calculate the final output features of this attention layer, $O = A(Q, K)V$, where $A$ is the causal self-attention ( $O$ should also be a $3 \times 2$ matrix).

**Given:**

$$X = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \\ 1.0 & 0.5 \end{bmatrix}, \quad W_q = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix}, \quad W_k = \begin{bmatrix} 1 & 0 \\ -0.5 & 0.5 \end{bmatrix}, \quad W_v = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

**Want:**
Compute the final causal self-attention output

$$O = A(Q, K)\, V$$

where

$$Q = XW_q, \quad K = XW_k, \quad V = XW_v,$$

and $A(Q, K)$ is the masked scaled-dot product attention (with scale $d_k = 1$).

**Equation:**

$$(1)\ Q = XW_q, \quad K = XW_k, \quad V = XW_v$$

$$(2)\ s_{ij} = Q_i \cdot K_j \quad \big(\text{for } j \le i, \text{ else masked}\big)$$

$$(3)\ \alpha_{ij} = \frac{\exp(s_{ij})}{\sum_{k=0}^{i} \exp(s_{ik})}, \quad \sum_{j=0}^{i} \alpha_{ij} = 1$$

$$(4)\ O_i = \sum_{j=0}^{i} \alpha_{ij}\, V_j \quad (i = 0, 1, 2)$$

**Solution:**

**Compute $Q$, $K$, and $V$:**

$$Q = XW_q = \begin{bmatrix} 0.1 & 0.9 \\ 0.25 & 0.75 \\ 0.5 & 1.0 \end{bmatrix}, \quad K = XW_k = \begin{bmatrix} -0.2 & 0.4 \\ 0.25 & 0.25 \\ 0.75 & 0.25 \end{bmatrix}, \quad V = XW_v = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \\ 1.0 & 0.5 \end{bmatrix}.$$

**Token 0 ($i = 0$):**

$$s_{00} = Q_0 \cdot K_0 = (0.1)(-0.2) + (0.9)(0.4) = 0.34,$$
$$\alpha_{00} = 1, \quad O_0 = \alpha_{00} V_0 = \begin{bmatrix} 0.2 & 0.8 \end{bmatrix}.$$

**Token 1 ($i = 1$):**

$$s_{10} = Q_1 \cdot K_0 = (0.25)(-0.2) + (0.75)(0.4) = 0.25, \quad s_{11} = Q_1 \cdot K_1 = (0.25)(0.25) + (0.75)(0.25) = 0.25,$$
$$\alpha_{10} = \alpha_{11} = \tfrac{1}{2}, \quad O_1 = \tfrac{1}{2} V_0 + \tfrac{1}{2} V_1 = \begin{bmatrix} 0.35 & 0.65 \end{bmatrix}.$$

**Token 2 ($i = 2$):**

$$s_{20} = Q_2 \cdot K_0 = (0.5)(-0.2) + (1.0)(0.4) = 0.30,$$
$$s_{21} = Q_2 \cdot K_1 = (0.5)(0.25) + (1.0)(0.25) = 0.375, \quad s_{22} = Q_2 \cdot K_2 = (0.5)(0.75) + (1.0)(0.25) = 0.625,$$
$$\alpha_{2\bullet} = \text{softmax}([0.30,\, 0.375,\, 0.625]) \approx [0.289,\, 0.311,\, 0.400],$$
$$O_2 = 0.289 V_0 + 0.311 V_1 + 0.400 V_2 \approx \begin{bmatrix} 0.6132 & 0.5867 \end{bmatrix}.$$

**Final Output $O$:**

$$O = \begin{bmatrix} 0.2000 & 0.8000 \\ 0.3500 & 0.6500 \\ 0.6132 & 0.5867 \end{bmatrix}.$$

(b) In this question you are asked to calculate the FID scores for two sets of fake images with respect to a set of real images on dummy values with 4 features to find out which set of generated images would be considered better.

The FID score is a measure of the similarity between the feature distributions of the real and generated image sets and can be used as a way to evaluate the performance of generative models like GANs. In reality, to calculate the FID score, you need to use an Inception network to extract 1000s of features from both the real images in your dataset and the generated images produced by your GAN.

For this dummy example, the features have already been extracted for you and the specifications are as follows:

Real Image set:

$$\mu_{\text{real}} = \begin{bmatrix} 0.5 & 0.2 & 0.7 & 0.4 \end{bmatrix}$$

$$\Sigma_{\text{real}} = \begin{bmatrix} 0.2 & 0.1 & 0.05 & 0.02 \\ 0.1 & 0.3 & 0.1 & 0.05 \\ 0.05 & 0.1 & 0.4 & 0.1 \\ 0.02 & 0.05 & 0.1 & 0.2 \end{bmatrix}$$

Fake Images Set I:

$$\mu_1 = \begin{bmatrix} 0.4 & 0.3 & 0.5 & 0.6 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 0.7 & 0.1 & 0.05 & 0.02 \\ 0.1 & 0.2 & 0.1 & 0.05 \\ 0.05 & 0.1 & 0.3 & 0.1 \\ 0.02 & 0.05 & 0.1 & 0.15 \end{bmatrix}$$

Fake Images Set II:

$$\mu_2 = \begin{bmatrix} 0.7 & 0.1 & 0.3 & 0.5 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 0.4 & 0.05 & 0.05 & 0.01 \\ 0.05 & 0.3 & 0.1 & 0.05 \\ 0.05 & 0.1 & 0.2 & 0.05 \\ 0.01 & 0.05 & 0.05 & 0.2 \end{bmatrix}$$

Please report your $\text{FID}_1$ and $\text{FID}_2$ scores along with calculations. Also mention which set of generated images is better in your opinion based on FID score.
**Note:** $\mu$ here refers to the mean vector and $\Sigma$ here refers to the covariance matrix of the feature representations of a set of images. $\mu$ contains feature-wise means, where each element represents the mean value of a particular feature across all images in the set. The covariance matrix $\Sigma$ is a square matrix that describes the relationships between different features in the set.

**Given:**

$$\mu_{\text{real}} = [0.5,\ 0.2,\ 0.7,\ 0.4], \quad \Sigma_{\text{real}} = \begin{bmatrix} 0.20 & 0.10 & 0.05 & 0.02 \\ 0.10 & 0.30 & 0.10 & 0.05 \\ 0.05 & 0.10 & 0.40 & 0.10 \\ 0.02 & 0.05 & 0.10 & 0.20 \end{bmatrix},$$

$$\mu_1 = [0.4,\ 0.3,\ 0.5,\ 0.6], \quad \Sigma_1 = \begin{bmatrix} 0.70 & 0.10 & 0.05 & 0.02 \\ 0.10 & 0.20 & 0.10 & 0.05 \\ 0.05 & 0.10 & 0.30 & 0.10 \\ 0.02 & 0.05 & 0.10 & 0.15 \end{bmatrix},$$

$$\mu_2 = [0.7,\ 0.1,\ 0.3,\ 0.5], \quad \Sigma_2 = \begin{bmatrix} 0.40 & 0.05 & 0.05 & 0.01 \\ 0.05 & 0.30 & 0.10 & 0.05 \\ 0.05 & 0.10 & 0.20 & 0.05 \\ 0.01 & 0.05 & 0.05 & 0.20 \end{bmatrix}.$$

**Want:**
$\mathrm{FID}_1$ and $\mathrm{FID}_2$ scores and mention which set of generated images is better in your opinion based on FID score.

**Equation:**

$$\Delta\mu = \mu_{\text{real}} - \mu_g, \quad \|\Delta\mu\|^2 = \sum_{i=1}^{4} (\Delta\mu_i)^2,$$

$$C = \left(\Sigma_{\text{real}}\,\Sigma_g\right)^{\frac{1}{2}}, \quad T = \mathrm{Tr}\big(\Sigma_{\text{real}} + \Sigma_g - 2\,C\big),$$

$$\mathrm{FID}_g = \|\Delta\mu\|^2 + T.$$

**Solution:**

**$\mathrm{FID}_1$:**

$$\Delta\mu = [0.5 - 0.4,\ 0.2 - 0.3,\ 0.7 - 0.5,\ 0.4 - 0.6] = [0.1,\ -0.1,\ 0.2,\ -0.2],$$

$$\|\Delta\mu\|^2 = 0.10, \quad \mathrm{Tr}(\Sigma_{\text{real}} + \Sigma_1) = 2.45,$$

$$\mathrm{Tr}\big(2\,(\Sigma_{\text{real}}\Sigma_1)^{1/2}\big) \approx 2.2689, \quad T = 2.45 - 2.2689 = 0.1811,$$

$$\mathrm{FID}_1 = 0.10 + 0.1811 = 0.2811.$$

**$\mathrm{FID}_2$:**

$$\Delta\mu = [0.5 - 0.7,\ 0.2 - 0.1,\ 0.7 - 0.3,\ 0.4 - 0.5] = [-0.2,\ 0.1,\ 0.4,\ -0.1],$$

$$\|\Delta\mu\|^2 = 0.22, \quad \mathrm{Tr}(\Sigma_{\text{real}} + \Sigma_2) = 2.20,$$

$$\mathrm{Tr}\big(2\,(\Sigma_{\text{real}}\Sigma_2)^{1/2}\big) \approx 2.1156, \quad T = 2.20 - 2.1156 = 0.0844,$$

$$\mathrm{FID}_2 = 0.22 + 0.0844 = 0.3044.$$

Since a lower FID indicates a closer match to the real distribution, Fake Set I ($\mathrm{FID}_1 = 0.2811$) is better than Fake Set II ($\mathrm{FID}_2 = 0.3044$).

(c) In this question, you are given the image $x_0$ and schedule $\beta$. Perform two iterations of the forward diffusion process and report the noisy image output $x_1$ and $x_2$. The forward diffusion equation is given below:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t I\big)$$

Image:

$$x_0 = \begin{bmatrix} 0.8 & 0.5 & 0.9 \\ 0.1 & 1 & 0.6 \\ 0.4 & 0.6 & 0.3 \end{bmatrix}$$

Standard deviation $\beta_0$: 0.0001 and $\beta_{1000}$: 0.02. Use a linear schedule to estimate the values of $\beta_1$ and $\beta_2$. The noise at $t = 1$ and $t = 2$ are given as:

$$\epsilon_1 = \begin{bmatrix} 1.0 & -1.0 & 1.0 \\ -1.0 & 1.0 & 1.0 \\ 1.0 & -1.0 & 1.0 \end{bmatrix} \quad \epsilon_2 = \begin{bmatrix} 2.0 & -2.0 & 2.0 \\ -2.0 & 2.0 & 2.0 \\ 2.0 & -2.0 & 2.0 \end{bmatrix}$$

**Note:** Use the reparameterization trick.

**Given:**

$$x_0 = \begin{bmatrix} 0.8 & 0.5 & 0.9 \\ 0.1 & 1.0 & 0.6 \\ 0.4 & 0.6 & 0.3 \end{bmatrix}, \quad \epsilon_1 = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}, \quad \epsilon_2 = \begin{bmatrix} 2 & -2 & 2 \\ -2 & 2 & 2 \\ 2 & -2 & 2 \end{bmatrix},$$

$$\beta_0 = 0.0001, \quad \beta_{1000} = 0.02.$$

**Want:** Compute the noisy images $x_1$ and $x_2$ after two iterations of the forward diffusion process.

**Equation:**

$$\beta_t = \beta_0 + t\,\frac{\beta_{1000} - \beta_0}{1000}, \tag{1}$$

$$x_t = \sqrt{1-\beta_t}\; x_{t-1} \;+\; \sqrt{\beta_t}\; \epsilon_t. \tag{2}$$

**Solution:**

**Compute $\beta_1$ and $\beta_2$:**

$$\beta_1 = 0.0001 + 1 \cdot \frac{0.0199}{1000} = 0.0001199, \quad \beta_2 = 0.0001 + 2 \cdot \frac{0.0199}{1000} = 0.0001398.$$

**Iteration $t = 1$:**

$$\sqrt{1-\beta_1} \approx 0.99994, \quad \sqrt{\beta_1} \approx 0.01095,$$

$$x_1 = 0.99994\, x_0 + 0.01095\, \epsilon_1 \approx \begin{bmatrix} 0.81090 & 0.48902 & 0.91090 \\ 0.08904 & 1.01089 & 0.61092 \\ 0.41093 & 0.58899 & 0.31093 \end{bmatrix}.$$

**Iteration $t = 2$:**

$$\sqrt{1-\beta_2} \approx 0.99993, \quad \sqrt{\beta_2} \approx 0.01182,$$

$$x_2 = 0.99993\, x_1 + 0.01182\, \epsilon_2 \approx \begin{bmatrix} 0.83448 & 0.46534 & 0.93448 \\ 0.06539 & 1.03446 & 0.63451 \\ 0.43454 & 0.56530 & 0.33455 \end{bmatrix}.$$

# 3 Multiple Choice Questions

**(1)** Which of the following is true about Diffusion Models.

A) It is theoretically impossible to construct a diffusion model that does not use conjugate prior and posterior distributions.

B) Conjugate prior and posterior distributions allow us to evaluate the loss function rather than rely on sampling techniques.

C) Latent Diffusion Models must learn the encoding and diffusion process simultaneously due to numerical instability during reverse diffusion sampling.

D) In Cascading Diffusion Models, conditioning augmentation refers to augmenting the initial step at each level of the cascade.

**Answer: B**

**(2)** Suppose you are training a VAE model. Even though the loss function converges after a good amount of iterations, you observe that the reconstructed samples are very similar even for different input samples. What is a possible way of addressing this issue?

A) Decrease the weight of the KL divergence term in the loss function and retrain the model from scratch.

B) Reduce the dimensionality of the input data and retrain the model from scratch.

C) Increase the learning rate and continue training the current model.

D) None of the Above.

**Answer: A**

**(3)** Which of the following is **NOT** a benefit of using attention mechanisms in Transformer models?

A) Improved performance on tasks that require modeling long-range dependencies.

B) To have a scalable model in which performance improved when having more parameters.

C) Ability to process inputs of varying lengths.

D) To reduce number of model parameters.

**Answer: D**

**(4)** Which of the following statements about self-attention in the Transformer architecture is **TRUE**?

A) Self-attention is only used to process the input sequence and is not used during the generation of new outputs.

B) Self-attention computes a weighted sum of the values using the dot product (normalized with softmax) of the queries and keys as the weights.

C) Self-attention is only used in the encoder layers of the Transformer, not in the decoder layers.

D) Self-attention cannot be used with variable-length input sequences.

**Answer: B**

**(5)** Which of the following statements about the attention mechanism in the Transformer architecture is **FALSE**?

A) The attention mechanism is a form of dynamic weight learning that allows the model to learn to selectively attend to different parts of the input sequence when computing each output.

B) The dot product attention used in the Transformer has a quadratic complexity with respect to the number of input tokens.

C) The multi-head attention mechanism in the Transformer allows the model to attend to different sequences within the same batch.

D) The attention mechanism can be modified to allow the model to attend to different modalities, such as image features.

**Answer: C**

**(6)** Which of the following is **NOT** a commonly used Loss Function in Generative Adversarial Networks (GANs)?

A) Binary cross-entropy.

B) Huber Loss.

C) Kullback-Leibler Divergence.

D) Wasserstein Distance.

**Answer: B**

**(7)** What is the role of noise in generative networks?

A) To decrease the loss in the minimax game.

B) To ensure KL divergence is large enough.

C) To ensure that our training is random.

D) Enable the creation of new and unique outputs.

**Answer: D**

**(8)** What is the role of a kernel in the discriminator of a GAN?

A) Down sample the images from a feature map.

B) Up sample the images from a feature map.

C) Generate new information from the latent space.

D) Reduce the size of an image without any learned parameters.

**Answer: A**

**(9)** Why do we use a variance schedule in the forward process of diffusion models?

A) To control the amount of noise added at each diffusion step.

B) To adjust the learning rate of the optimization process.

C) To increase the diversity of the generated image.

D) To regulate the color balance and saturation of the image.

**Answer: A**

**(10)** In the context of attention mechanisms in neural networks, what is the purpose of the query, key, and value vectors?

A) Query vectors encode the information from the input sequence, key vectors determine the relevance of each element, and value vectors store the learned context.

B) TQuery vectors determine the relevance of each element, key vectors encode the information from the input sequence, and value vectors store the learned context.

C) Query vectors store the learned context, key vectors determine the relevance of each element, and value vectors encode the information from the input sequence.

D) Adding attention to the zt and ft activations of a GRU to add attention information alongside long and short term memory.

**Answer: B**