NVIDIA
NCA-GENM
NVIDIA Generative AI Multimodal

For More Information – Visit link below:
https://www.examsempire.com/

Product Version

1. **Up to Date products, reliable and verified.**
2. **Questions and Answers in PDF Format.**

Latest Version: 6.0

Question: 1

You are building a multimodal Generative AI model that takes text and images as input to generate a story. The text encoder uses a pre-trained BERT model, and the image encoder uses a pre-trained ResNet50 model. What is the BEST strategy to align the feature spaces of these two encoders during training to ensure effective multimodal fusion?

A. Fine-tune only the BERT model while keeping the ResNet50 model frozen.
B. Fine-tune only the ResNet50 model while keeping the BERT model frozen.
C. Use a contrastive loss function that encourages similar representations for semantically related text and images, and dissimilar representations otherwise. Fine-tune BERT and ResNet50.
D. Train a separate linear projection layer for each encoder and minimize the LI distance between the projected features. Freeze BERT and ResNet50.
E. Concatenate the outputs of BERT and ResNet50 directly without any alignment strategy.

Answer: C

Explanation:
Contrastive learning is a powerful technique for aligning feature spaces in multimodal learning. By encouraging similar representations for semantically related inputs and dissimilar representations for unrelated inputs, it allows the model to learn a shared representation space that facilitates effective fusion. Fine- tuning both encoders allows for adaptation to the specific task. Other methods are less effective for aligning high-dimensional feature spaces from different modalities.

Question: 2

You are training a multimodal model that combines audio and video dat
a. You observe that the model performs well on the training data but generalizes poorly to unseen data. Which of the following regularization techniques is MOST likely to improve the generalization performance in this scenario?
A. L1 Regularization (Lasso)
B. Dropout
C. Early Stopping
D. Weight Decay (L2 Regularization)
E. Data Augmentation

Answer: E

Explanation:
Data augmentation is the most effective regularization technique in this scenario because it increases the diversity of the training data, making the model more robust to variations in unseen data L1 and L2 regularization can help prevent over fitting, but data augmentation directly addresses the issue of

limited training data. Dropout also helps, but data augmentation is generally more impactful for multimodal data where variations are significant. Early stopping can also help, but it is not as effective as data augmentation.

Question: 3

Which of the following is NOT a common challenge in training multimodal Generative AI models?

A. Handling different data modalities with varying statistical properties.
B. Aligning feature spaces of different modalities.
C. Dealing with missing modality data during inference.
D. Optimizing for a single modality at the expense of others.
E. The computational complexity associated with training large unimodal models.

Answer: E

Explanation:
The computational complexity of training large unimodal models is a challenge for unimodal models, but not a distinct challenge inherent to multimodal models. Multimodal models have unique challenges related to data heterogeneity, feature alignment, handling missing modalities, and balancing performance across modalities.

Question: 4

Consider the following code snippet used for creating a multimodal dataset with PyTorch. The dataset contains images and corresponding text descriptions. However, during training, you observe a significant imbalance in the data distribution of text lengths. Which of the following techniques would BEST address this issue?

A. Applying standard image augmentation techniques to the image data.
B. Padding or truncating text sequences to a fixed length.
C. Using a learning rate scheduler to adjust the learning rate during training-
D. Applying Batch Normalization to the image features.
E. Using the exact same length of text and same images.

Answer: B

Explanation:
Padding or truncating text sequences to a fixed length is a standard technique for handling variable-length sequences in NLP tasks. This ensures that all text inputs have the same dimensionality, which is required for efficient batch processing in neural networks- While image augmentation can improve the model's robustness to variations in image data, it does not directly address the issue of text length imbalance. Learning rate scheduling and batch normalization are general training techniques that can improve convergence, but they do not specifically address the text length imbalance.

Question: 5

You are training a multimodal model with text and audio inputs. You notice that the audio modality dominates the training process, and the text modality is not contributing significantly to the final performance. Which of the following strategies can you use to address this modality imbalance? (Select TWO)

A. Increase the learning rate for the text encoder.
B. Decrease the batch size for the audio data
C. Apply a modality-specific weighting scheme to the loss function, giving more weight to the text loss
D. Remove the audio modality altogether to force the model to rely on text
E. Increase the size of the audio dataset.

Answer: A,C

Explanation:
Increasing the learning rate for the text encoder can help the text modality learn more effectively Applying a modality-specific weighting scheme to the loss function allows you to explicitly control the contribution of each modality to the overall loss, giving more weight to the underperforming text modality. Decreasing the batch size for audio data might have a small impact, but it's not a primary strategy for addressing modality imbalance. Removing the audio modality is not a desirable solution, as it eliminates valuable information. Increasing the size of audio dataset will even more dominate_ So, the most effective strategies are increasing the learning rate for text and weighting the loss function.

Question: 6

You are building a generative model that takes both image and text input to generate novel images. You are using a Variational Autoencoder (VAE) architecture with separate encoders for images and text. After training, you observe that the generated images are heavily influenced by the image input and barely incorporate the text information. Which of the following techniques would MOST likely improve the incorporation of text information into the generated images?

A. Increasing the capacity of the image encoder and decoder.
B. Decreasing the capacity of the text encoder.
C. Using a cross-attention mechanism in the decoder to allow the image features to attend to the text features during image generatiom
D. Removing the text encoder and only using the image encoder.
E. Train two separate VAE models. One for Text and another for images.

Answer: C

Explanation:
A cross-attention mechanism allows the image features to selectively attend to the relevant parts of the text features during the image generation process. This enables the model to effectively incorporate the text information into the generated images- Increasing the capacity of the image encoder/decoder

might further bias the model towards the image input Decreasing the capacity of the text encoder would further reduce the influence of text. Removing the text encoder is obviously not a solution- Training two separate VAE models won't generate correlated Image and Text.

Question: 7

You are fine-tuning a pre-trained multimodal model for a new task. You have limited computational resources. Which of the following fine-tuning strategies would be the MOST computationally efficient while still achieving good performance?

A. Fine-tune all the layers of the model.
B. Freeze all layers except the classification head and fine-tune only the classification head.
C. Freeze the lower layers of the model and fine-tune the upper layers and the classification head.
D. Train a new random model from scratch for the task, which will avoid the need to load the pre-trained model.
E. Randomize the model to train, if it improves the training rate.

Answer: C

Explanation:
Freezing the lower layers and fine-tuning the upper layers and classification head strikes a balance between computational efficiency and performance. The lower layers typically capture more general features that are less specific to the task, while the upper layers capture more task-specific features. Freezing the lower layers reduces the number of trainable parameters, making the fine-tuning process more computationally efficient. Fine-tuning all layers is computationally expensive, freezing all layers except the classification head might not be sufficient for adapting to the new task, and training from scratch does not leverage the knowledge learned during pre-training. Randomizing model is not a general practice.

Question: 8

When training a multimodal generative model for image captioning, you notice the model generates grammatically correct but generic and uninformative captions. Which technique is MOST likely to improve the in formativeness and specificity of the generated captions?

A. Increase the size of the image encoder.
B. Use beam search during inference with a large beam size.
C. Employ a diverse beam search or sampling strategy during inference to encourage exploration of different caption possibilities.
D. Decrease the learning rate during training.
E. Decrese the size of the vocabulary.

Answer: C

Explanation:

Diverse beam search or sampling strategies encourage the model to explore different caption possibilities during inference, leading to more diverse and informative captions. Standard beam search often converges to the most likely caption, which tends to be generic. Increasing the image encoder Size might improve image feature extraction but doesn't directly address the caption informativeness problem. Decreasing the learning rate is a general training technique that might improve convergence but doesn't specifically target caption informativeness.

Question: 9

You're tasked with building a system that can generate realistic images from text descriptions and, conversely, generate accurate text descriptions from images. You decide to use a GAN (Generative Adversarial Network) architecture, but need to handle both modalities effectively. What GAN variant would be MOST suitable for this bi-directional multimodal task?

A. Vanilla GAN
B. Conditional GAN (cGAN)
C. CycleGAN
D. Deep Convolutional GAN (DCGAN)
E. Super-Resolution GAN (SRGAN)

Answer: C

Explanation:
CycleGAN is designed for unpaired image-to-image translation. In this scenario, it can be adapted to translate between the image and text modalities without requiring paired data. One generator learns to generate images from text, while another learns to generate text from images. Cycle consistency ensures that translating an image to text and then back to an image results in an image similar to the original. Vanilla GANI cGAN, and DCGAN are not inherently designed for bi- directional translation between modalities without paired data. SRGAN is for image super-resolution.

Question: 10

Consider the following Python code snippet using PyTorch. What does this code do in the context of data preprocessing for a Generative AI model?

A.

```
import torch
import torchvision.transforms as transforms

transform = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])
```

It applies data augmentation techniques to increase the training dataset size.

B.

```
import torch
import torchvision.transforms as transforms

transform = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])
```

It resizes images to 256x256, converts them to tensors, and normalizes the pixel values to the range [-1, 1].

C.

```
import torch
import torchvision.transforms as transforms

transform = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])
```

It converts images to grayscale and applies a Gaussian blur.

D.

```
import torch
import torchvision.transforms as transforms

transform = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])
```

It encodes images into a latent space representation using a pre-trained autoencoder.

E.

```
import torch
import torchvision.transforms as transforms

transform = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])
```

It performs image segmentation to identify objects within the images.

Answer: B

Explanation:
The code snippet first resizes the images to a fixed size (256x256). Then, it converts the images into PyTorch tensors, which are the standard data format for PyTorch models. Finally, it normalizes the pixel values to a range of approximately [-1, 1]. This normalization helps to improve the training stability and performance of the generative A1 model by scaling the input values.

Thank You for Trying Our Product

Special 16 USD Discount Coupon: NSZUBG3X

Email: support@examsempire.com

Check our Customer Testimonials and ratings available on every product page.

Visit our website.

https://examsempire.com/