1. What are the main tasks that autoencoders are used for?

**Data Compression: Autoencoders are used for data compression, where they learn a compact representation of input data while retaining its essential features.**

**Feature Learning: They can learn meaningful features from unlabeled data, which can be useful for downstream supervised tasks.**

**Anomaly Detection: Autoencoders can detect anomalies or outliers by reconstructing data, and deviations from the reconstruction indicate anomalies.**

**Noise Reduction: They can denoise data by learning to reconstruct clean samples from noisy input.**

2. Suppose you want to train a classifier, and you have plenty of unlabeled training data but

only a few thousand labeled instances. How can autoencoders help? How would you

proceed?

**Autoencoders can help in semi-supervised learning scenarios with limited labeled data:**

**Pretrain an autoencoder on the large amount of unlabeled data to learn useful representations.**

**Fine-tune the autoencoder as a feature extractor with the limited labeled data using a classifier head.**

**The pretrained features often provide a good starting point for the classifier, improving its performance with limited labeled data.**

3. If an autoencoder perfectly reconstructs the inputs, is it necessarily a good autoencoder?

How can you evaluate the performance of an autoencoder?

**While perfect reconstruction is desirable, it may not be sufficient to judge the quality of an autoencoder.**

**Performance evaluation should consider additional factors like the utility of the learned representations for downstream tasks, the ability to handle noisy data, and the efficiency of encoding and decoding.**

4. What are undercomplete and overcomplete autoencoders? What is the main risk of an

excessively undercomplete autoencoder? What about the main risk of an overcomplete

autoencoder?

**Undercomplete Autoencoders: Have an encoder with fewer neurons in the hidden layer than the input layer. The main risk is losing important information due to severe dimensionality reduction.**

**Overcomplete Autoencoders: Have more neurons in the hidden layer than the input layer. The main risk is that the model might memorize the training data instead of learning meaningful features.**

5. How do you tie weights in a stacked autoencoder? What is the point of doing so?

**In stacked autoencoders, the weights of the decoder's layers are tied to the transpose of the corresponding encoder layers' weights.**

**Tying weights reduces the number of parameters in the model and enforces symmetry between the encoder and decoder, improving training efficiency and generalization.**

6. What is a generative model? Can you name a type of generative autoencoder?

**Generative models are capable of generating new data samples that resemble the training data.**

**A type of generative autoencoder is the Variational Autoencoder (VAE), which learns to generate data samples and has applications in generating new images, text, and more.**

7. What is a GAN? Can you name a few tasks where GANs can shine?

**GANs are a type of generative model that consists of a generator and a discriminator, trained adversarially.**

**GANs can shine in tasks like image generation, style transfer, data augmentation, super-resolution, and anomaly detection.**

8. What are the main difficulties when training GANs?

**Mode Collapse: GANs can generate limited diversity in the data they produce, resulting in mode collapse.**

**Training Instability: GAN training can be unstable, with the generator and discriminator oscillating during training.**

**Hyperparameter Sensitivity: GANs are sensitive to hyperparameters like learning rates and architectural choices.**

**Evaluation: Evaluating GANs' performance and comparing different models is challenging.**

**Convergence: Ensuring that GANs converge to a stable equilibrium can be difficult.**