1. What are the main tasks that autoencoders are used for?
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?
3. If an autoencoder perfectly reconstructs the inputs, is it necessarily a good autoencoder? How can you evaluate the performance of an autoencoder?
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?
5. How do you tie weights in a stacked autoencoder? What is the point of doing so?
6. What is a generative model? Can you name a type of generative autoencoder?
7. What is a GAN? Can you name a few tasks where GANs can shine?
8. What are the main difficulties when training GANs?

Answer:

Autoencoders are neural network models that can learn to reconstruct their input data with a compressed representation. They can be used for a variety of tasks, including image and video compression, data denoising, dimensionality reduction, anomaly detection, and feature extraction.

Autoencoders can be used for unsupervised pre-training of deep neural networks. By training an autoencoder on the unlabeled data, the encoder network can learn to extract useful features from the data, which can then be used to initialize the weights of a classifier. The classifier can then be fine-tuned on the labeled data to improve its performance.

No, a perfect reconstruction does not necessarily mean that the autoencoder is good. A good autoencoder should be able to capture the most important features of the input data in the compressed representation, allowing it to generalize well to new data. The performance of an autoencoder can be evaluated using metrics such as reconstruction loss, mean squared error, or accuracy.

An undercomplete autoencoder has a compressed representation that is smaller than the input, while an overcomplete autoencoder has a compressed representation that is larger than the input. An excessively undercomplete autoencoder can result in a loss of information and poor reconstruction quality, while an overcomplete autoencoder can learn to simply memorize the input data and fail to generalize to new data.

In a stacked autoencoder, the encoder of one layer and the decoder of the subsequent layer share the same weights, tying the weights. This can reduce the number of parameters in the model, leading to better generalization and faster training.

A generative model is a type of model that can learn to generate new data that is similar to the training data. A generative autoencoder is an autoencoder that is trained to generate new data that is similar to the input data, by sampling from the compressed representation.

A GAN, or Generative Adversarial Network, is a type of generative model that consists of two neural networks: a generator network and a discriminator network. The generator network generates new data that is similar to the training data, while the discriminator network learns to distinguish between the generated data and the real data. GANs can be used for tasks such as image and video synthesis, text generation, and data augmentation.

The main difficulties when training GANs include mode collapse, where the generator produces a limited set of outputs, and instability in the training process, where the generator and discriminator networks can become unstable and fail to converge. Other challenges include finding the right hyperparameters, selecting appropriate loss functions, and dealing with high-dimensional data. To handle these difficulties, techniques such as regularization, alternative loss functions, and network architectures have been developed.