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

**An Autoencoders can be considered as an artificial neural network its main task is basically to convert a high dimensional data to low dimensional data and it helps to reduce the dimensionality and it converts a high dimensional data to low dimensional data by simply capturing the important parts of a data .**

1. **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?**

**Auto encoders help in this situation because,**

**Auto encoders are type of neural networks, which do not need labeled data i.e it involves unsupervised learning.**

**They take raw data(unlabelled) and train from it. In fact auto encoders generate the labels themself which is why they are also known as self supervised neural networks.**

**Now, since there is lot of unlabelled data, the data can be given as an input to a Auto encoder and the auto encoder does the job of generating labels through training.**

**Important: To get even more accuracy, we use one more neural network by initializing weights taken from encoder and also using labeled data. This helps because the auto encoder might have captured the underlying structure/pattern in the raw data**

**If an autoencoder perfectly reconstructs the inputs, is it necessarily a good autoencoder? How can you evaluate the performance of an autoencoder?**

**Yes...it is a good autoencoder. In practice, efficiency of autoencoder depends on how well it reconstructs and also on how robust it is to noise in different scenes.**

**Common practice is to add noise sampled from input distribution to the input space to make sure autoencoder, vanilla or VAE, learns to reconstruct the input more robustly regardless of scenic distortions.**

**However, maybe your goal never was reconstruction and thus it doesn’t matter how good reconstruction is. Maybe you wanted to learn features and leverage it for other use. In that case, you wouldn’t care, mostly, about how well reconstruction happens. It is known that noise in input space doesn’t necessary help in better converage of feature space and thus feature learning is hampered. So, community came up with idea of introducing noise in the feature space instead of input space. It will obviously hurt the reconstruction but definitely learned features would be better and overall your feature vector would be more definitive of the latent space as a whole.**

1. **What are undercomplete and overcompleteautoencoders? What is the main risk of an excessively undercompleteautoencoder? What about the main risk of an overcompleteautoencoder?**

**Variables and Formulas**

**Autoencoder: A neural network that learns to encode and decode data by training on its own inputs. Undercomplete Autoencoder: An autoencoder with a smaller number of hidden units than the input layer. Overcomplete Autoencoder: An autoencoder with a larger number of hidden units than the input layer. Bottleneck: The hidden layer with fewer units in an undercomplete autoencoder.**

**Concepts**

**Undercomplete Autoencoders**

**An undercomplete autoencoder is a neural network that has fewer hidden units in its encoding layer than the input layer. This bottleneck forces the network to learn a more compact and efficient representation of the input data.**

**Overcomplete Autoencoders**

**An overcomplete autoencoder is a neural network that has more hidden units in its encoding layer than the input layer. This allows the network to learn a more complex and detailed representation of the input data.**

**Main Risks**

**The main risk of an excessively undercomplete autoencoder is that it may not be able to learn a sufficiently rich representation of the input data. This can lead to poor performance in tasks such as denoising, compression, or generation.**

**The main risk of an overcomplete autoencoder is that it may overfit the training data, leading to poor generalization performance on new data. This can result in the network learning to memorize the training data rather than learning a more generalizable representation.**

**Main Body**

**Step 1: Undercomplete Autoencoders**

**An undercomplete autoencoder has a smaller number of hidden units than the input layer, creating a bottleneck that forces the network to learn a more compact and efficient representation of the input data. This bottleneck can help the network learn important features and patterns in the data, making it useful for tasks such as denoising, compression, or generation.**

**Step 2: Overcomplete Autoencoders**

**An overcomplete autoencoder has a larger number of hidden units than the input layer, allowing the network to learn a more complex and detailed representation of the input data. This can be useful for tasks such as feature learning or representation learning.**

**Step 3: Main Risks**

**The main risk of an excessively undercomplete autoencoder is that it may not be able to learn a sufficiently rich representation of the input data. This can lead to poor performance in tasks such as denoising, compression, or generation.**

**The main risk of an overcomplete autoencoder is that it may overfit the training data, leading to poor generalization performance on new data. This can result in the network learning to memorize the training data rather than learning a more generalizable representation.**

**Undercomplete autoencoders are neural networks with fewer hidden units than the input layer, forcing them to learn a more compact and efficient representation of the input data. Overcomplete autoencoders have more hidden units than the input layer, allowing them to learn a more complex and detailed representation. The main risk of an excessively undercomplete autoencoder is poor performance due to insufficient representation, while the main risk of an overcomplete autoencoder is overfitting and poor generalization performance.**

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

An autoencoder with tied weights has decoder weights that are the transpose of the encoder weights; this is a form of parameter sharing, which reduces the number of parameters of the model.

Stacked Autoencoder  
  
A single Autoencoder might be unable to reduce the dimensionality of the input features. Therefore for such use cases, we use stacked autoencoders. The stacked autoencoders are, as the name suggests, **multiple encoders stacked on top of one another**. Hence you get "stacked deep autoencoder". Now the advantage is instead of random initialization, all the hidden layers already have a lot of information encoded about the training data. And empirically it has been shown that this method is reliable and usually converges to better local minimum.

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

**Machine learning models can be classified into two types of models – Discriminative and Generative models. In simple words, a discriminative model makes predictions on the unseen data based on conditional probability and can be used either for classification or regression problem statements. On the contrary, a generative model focuses on the distribution of a dataset to return a probability for a given example.**

**Generative models are considered as a class of statistical models that can generate new data instances. These models are used in unsupervised machine learning as a means to perform tasks such as**

* **Probability and Likelihood estimation,**
* **Modeling data points,**
* **To describe the phenomenon in data,**
* **To distinguish between classes based on these probabilities.**

**Since these types of models often rely on the Bayes theorem to find the joint probability, so generative models can tackle a more complex task than analogous discriminative models.**

**So, Generative models focus on the distribution of individual classes in a dataset and the learning algorithms tend to model the underlying patterns or distribution of the data points. These models use the concept of joint probability and create the instances where a given feature (*x*) or input and the desired output or label (*y*) exist at the same time.**

**These models use probability estimates and likelihood to model data points and differentiate between different class labels present in a dataset. Unlike discriminative models, these models are also capable of generating new data points.**

**However, they also have a major drawback – If there is a presence of outliers in the dataset, then it affects these types of models to a significant extent.**

**Variational Autoencoder is a type of generative autoencoder.**

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

**GAN-Generative adversarial networks are a revenyand exciting innovation in machine learning.these are generative models which creates new data that resemble your past data.it is un- supervised and use a cooperative zero sum game framework to learn.**

**These are used for high fidelity natural image synthesis,data augmentation tasks,improving image compressions.**

1. **What are the main difficulties when training GANs?**

**The main difficulties when training GANS(Generative Adversarial Networks ) are:**

**Non-Convergence or slow convergence : AS GANS mostly fail to converge. In the model never converge , Destabilize and the model parameters oscillates**

**Mode Collapse : produces limited varieties of samples by the generator collapses**

**Diminished Gradient : The generator gradient vanishes and learns nothing**

**Overgeneralization : Here the potential data samples should not exist**

**Problems with perspective :Sometimes GANS can't differentiate between front and back view.**

**Setting up failure and bad initialization : when the discriminator gets a loss zero, then you can figure out that there is a wrong with your model. But the worst situation is figuring out what exactly is wrong.**

**Internal covariate shifts : There are Internal Covaraite shifts in GANS**

**Instability : There is no stability**