1)what are the main tasks that autoencoders used for ?

Ans : The main task of an autoencoder is to learn a lower-dimensional representation (encoding) for a higher-dimensional data, typically for dimensionality reduction, by training the network to capture the most important parts of the input image.

2)suppose you want to train a classifier,and you have plenty of unlabeled training data but only a few thousand labeled instance,how can autoencoders help ?how would you proceed ?

Ans : Autoencoders are learned automatically from data examples. It means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input and that it does not require any new engineering, only the appropriate training data.

However, autoencoders will do a poor job for image compression. As the autoencoder is trained on a given set of data, it will achieve reasonable compression results on data similar to the training set used but will be poor general-purpose image compressors. Compression techniques like JPEG will do vastly better.

Autoencoders are trained to preserve as much information as possible when an input is run through the encoder and then the decoder, but are also trained to make the new representation have various nice properties. Different kinds of autoencoders aim to achieve different kinds of properties.

Types of autoencoder :

Vanilla autoencoder

Multilayer autoencoder

Convolutional autoencoder

Regularized autoencoder

3)if an autoencoders Perfectly reconstruct the input,is it Necessarily a good autoencoders? How can you evaluate the performance of an autoencoder ?

Ans : There are many kinds of auto-encoders but they typically aim to reconstruct a higher dimensional space from either a (1) lower dimensional or (2) heavily regularized and redundant representation. The key ingredient of a bottleneck is what allows for generalization if an auto-encoder reconstructs its inputs perfectly but lacks that bottleneck, then it’s essentially an identity function.

4) what are undercomplete and overcomplete autoencoders ?what is the main risk of an excessively under complete autoencoder ?what about the main risk of an overcomplete autoencoders ?

Ans : undercomplete :This is when our encoding output's dimension is smaller than our input's dimension.

Essentially we reduced the dimension of our data with an under complete AE.

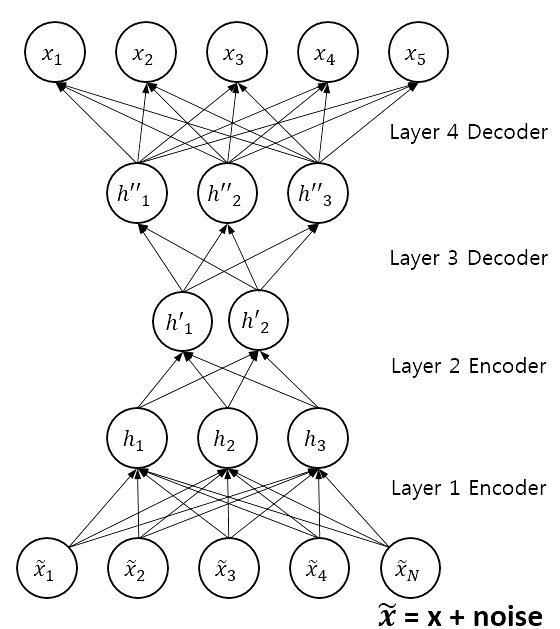
Overcomplete : This is when our encoding output's dimension is larger than our input's dimension.

Essentially we increased the dimension of our data with an overcomplete AE

5) how do you the weights in a stacked autoencoders ? What is the point of doing so ?

Ans :

Stacked Autoencoder is a multi-layer neural network which consists of Autoencoders in each layer. Each layer's input is from previous layer's output. The greedy layer wise pre-training is an unsupervised approach that trains only one layer each time.



Weight Tying improves the performance of language models by tying (sharing) the weights of the embedding and softmax layers. This method also massively reduces the total number of parameters in the language models that it is applied to

6)what is a generative model ?can you nama a type of generative autoencoder ?

Ans : A Generative Model is a way of learning any kind of data distribution. It is used in unsupervised machine learning as a means to describe phenomena in data, enabling the computers to understand the real world.

In unsupervised machine learning, generative modeling algorithms process the training data and make reductions in the data. These models generally are run on neural networks and can come to naturally recognize the distinctive features of the data .The main aim of all types of generative models is to learn the true data distribution of the training set so that the new data points are generated with some variations

The two types of generative models are:

1.Variational Autoencoder (VAE)

2.Generative Adversarial Networks (GAN)

7)what is GAN ?can you name a few tasks where GANs can shine ?

Ans : Generative adversarial networks (GANs) are deep neural net architectures comprised of two nets, pitting one against the other GAN is having a huge scope or potential because they can learn to mimic any distribution of data. That is, GANs can be taught to learn anything in any domain: images, music, speech, prose. They are robot artists in a sense, and their output is impressive.

A generative adversarial network (GAN) is a machine learning (ML) model in which two neural networks compete with each other to become more accurate in their predictions. GANs typically run unsupervised and use a cooperative zero-sum game framework to learn.

Generative: To learn a generative model, which describes how data is generated in terms of a probabilistic model.

Adversarial: The training of a model is done in an adversarial setting.

Networks: Use deep neural networks as the artificial intelligence (AI) algorithms for training purpose.

8 )what are the main difficulties when training GANs ?

Ans : Non-convergence: the model parameters oscillate, destabilize and never converge,

Mode collapse: the generator collapses which produces limited varieties of samples,

Diminished gradient: the discriminator gets too successful that the generator gradient vanishes and learns nothing,

Unbalance between the generator and discriminator causing overfitting, &

Highly sensitive to the hyperparameter selections.