1. What are Vanilla autoencoders

A1. Vanilla autoencoders are a type of neural network that are used for unsupervised learning. They consist of two main components: an encoder and a decoder. The encoder takes an input and transforms it into a lower-dimensional representation, called a latent representation or code. The decoder takes this code as input and attempts to reconstruct the original input. The objective of the autoencoder is to minimize the difference between the original input and the reconstructed output. This process of encoding and decoding can be used for tasks such as image denoising, dimensionality reduction, and anomaly detection. Vanilla autoencoders are called "vanilla" to differentiate them from more advanced types of autoencoders such as variational autoencoders and generative adversarial networks.

1. What are Sparse autoencoders

A2. Sparse autoencoders are a type of neural network that learns to represent the input data in a compressed form. Similar to vanilla autoencoders, they consist of an encoder and a decoder, which are trained to encode and reconstruct the input data, respectively. However, in sparse autoencoders, the encoder is constrained to learn sparse representations of the input data. This is done by adding a sparsity constraint to the loss function, which penalizes the encoder for activating too many neurons in the hidden layer. The idea behind this is to force the network to learn more informative and less redundant features. Sparse autoencoders have been used in various applications, such as anomaly detection, feature selection, and image denoising.

1. What are Denoising autoencoders

A3. Denoising autoencoders are a type of autoencoder that are trained to reconstruct clean data from corrupted or noisy input data. The idea is to force the autoencoder to learn robust representations of the input data that can handle variations and noise.

1. What are Convolutional autoencoders

A4. Convolutional autoencoders are a type of autoencoder that use convolutional neural network (CNN) layers in both the encoder and decoder parts of the network. They are commonly used for image data and can capture spatial patterns in the input data.

1. What are Stacked autoencoders

A5. Stacked autoencoders are a type of autoencoder that are composed of multiple layers of encoding and decoding units. They are trained in a layer-wise manner and can learn increasingly complex and abstract representations of the input data. The encoded representation from one layer serves as the input to the next layer, allowing the model to capture hierarchical patterns in the data. Stacked autoencoders are commonly used for dimensionality reduction and feature extraction.

1. Explain how to generate sentences using LSTM autoencoders

A6. LSTM autoencoders can be used to generate sentences by training the model to predict the next word in a sequence given the previous words. The input to the model is a sequence of words and the output is the next word in the sequence. During training, the model learns to predict the next word in the sequence based on the previous words. Once the model is trained, it can be used to generate new sentences by providing an initial sequence of words and then sampling the predicted next word at each time step to generate a sequence of words. This process can be repeated to generate multiple sentences. However, the quality of the generated sentences depends on the quality of the training data and the architecture of the LSTM autoencoder model.

1. Explain Extractive summarization

A7. Extractive summarization involves selecting a subset of important sentences or phrases from a given text document to form a summary. In this approach, the summary consists of only those parts of the document that are deemed important based on certain criteria, such as word frequency, sentence relevance, and coherence.

1. Explain Abstractive summarization

A8. Abstractive summarization involves generating a summary that goes beyond simply selecting sentences or phrases from the original document. This approach involves understanding the meaning of the text and generating a summary in natural language that conveys the main idea of the document. Abstractive summarization requires a more advanced natural language generation technique, such as using neural networks.

1. Explain Beam search

A9. Beam search is a search algorithm commonly used in sequence generation tasks such as machine translation and text summarization. It works by generating a set of possible sequences, called hypotheses, and scoring them based on a predetermined metric such as likelihood or fluency. The algorithm then prunes the set of hypotheses and keeps only the top-scoring ones to generate the final sequence.

1. Explain Length normalization

A10. Length normalization is a technique used to prevent the algorithm from favoring shorter or longer sequences. In summarization tasks, this is important because shorter summaries may miss important information while longer summaries may contain irrelevant details. Length normalization involves dividing the score of each hypothesis by its length raised to a certain power.

1. Explain Coverage normalization

A11. Coverage normalization is a technique used to ensure that the generated summary covers all the important aspects of the input text. This technique involves keeping track of the coverage of each input sentence or phrase during the summarization process and adding a penalty to the score of hypotheses that do not cover a certain percentage of the input.

1. Explain ROUGE metric evaluation

A12. Coverage normalization is a technique used to ensure that the generated summary covers all the important aspects of the input text. This technique involves keeping track of the coverage of each input sentence or phrase during the summarization process and adding a penalty to the score of hypotheses that do not cover a certain percentage of the input.