**1. Vanilla Autoencoders**

**Definition**:  
Vanilla autoencoders are a type of neural network used for unsupervised learning. They consist of an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation (latent space), and the decoder reconstructs the input data from this representation.

**Applications**:

* Dimensionality Reduction
* Feature Extraction
* Data Denoising

**2. Sparse Autoencoders**

**Definition**:  
Sparse autoencoders are a variant of autoencoders that introduce a sparsity constraint on the hidden layer activations. This encourages the model to learn a sparse representation of the input data.

**Applications**:

* Feature Learning
* Data Compression

**Benefits**:

* Promotes the discovery of useful features by enforcing sparsity.

**3. Denoising Autoencoders**

**Definition**:  
Denoising autoencoders are designed to reconstruct the original input from a corrupted version of it. The model learns to remove noise and recover the clean input data.

**Applications**:

* Data Denoising
* Robust Feature Learning

**Benefits**:

* Improves the robustness of the learned features by training on noisy data.

**4. Convolutional Autoencoders**

**Definition**:  
Convolutional autoencoders use convolutional layers in both the encoder and decoder. They are particularly effective for image data, as they can capture spatial hierarchies in the data.

**Applications**:

* Image Compression
* Image Denoising
* Feature Extraction

**Benefits**:

* Leverages the power of convolutional neural networks for image data.

**5. Stacked Autoencoders**

**Definition**:  
Stacked autoencoders consist of multiple layers of autoencoders stacked on top of each other. Each layer is trained sequentially, and the output of one layer serves as the input to the next.

**Applications**:

* Deep Feature Learning
* Pre-training for Deep Neural Networks

**Benefits**:

* Captures hierarchical features by learning multiple levels of representations.

**6. Generating Sentences Using LSTM Autoencoders**

**Process**:

1. **Encoder**: An LSTM network encodes the input sentence into a fixed-size context vector.
2. **Decoder**: Another LSTM network generates the output sentence from the context vector.
3. **Training**: The model is trained to minimize the reconstruction error between the input and output sentences.

**Applications**:

* Text Generation
* Machine Translation
* Text Summarization

**7. Extractive Summarization**

**Definition**:  
Extractive summarization involves selecting the most important sentences or phrases from the original text to create a summary.

**Methods**:

* Sentence Scoring: Assign scores to sentences based on their importance and select the top-scoring ones.
* Graph-Based Methods: Use graph algorithms like TextRank to identify important sentences.

**Applications**:

* News Summarization
* Document Summarization

**8. Abstractive Summarization**

**Definition**:  
Abstractive summarization involves generating a summary by paraphrasing and rephrasing the original text, often producing new sentences that capture the main ideas.

**Methods**:

* Sequence-to-Sequence Models: Use encoder-decoder architectures with attention mechanisms.
* Transformer Models: Utilize models like BERT and GPT for generating summaries.

**Applications**:

* News Summarization
* Document Summarization

**9. Beam Search**

**Definition**:  
Beam search is a heuristic search algorithm used in sequence generation tasks. It explores multiple possible sequences at each step and keeps the top-k candidates based on their scores.

**Process**:

1. **Initialization**: Start with the initial state.
2. **Expansion**: Generate possible next states and keep the top-k candidates.
3. **Selection**: Repeat the process until the end of the sequence is reached.

**Applications**:

* Machine Translation
* Text Generation

**10. Length Normalization**

**Definition**:  
Length normalization is a technique used in sequence generation tasks to adjust the scores of sequences based on their length. It prevents the model from favoring shorter sequences.

**Methods**:

* **Divide by Length**: Divide the sequence score by its length.
* **Exponential Decay**: Apply an exponential decay factor to the sequence score.

**Benefits**:

* Encourages the generation of longer and more informative sequences.

**11. Coverage Normalization**

**Definition**:  
Coverage normalization is a technique used in summarization and translation tasks to ensure that all parts of the input are covered in the output. It penalizes the model for repeatedly focusing on the same parts of the input.

**Methods**:

* **Coverage Vector**: Maintain a coverage vector that tracks the attention given to each part of the input.
* **Penalty Term**: Add a penalty term to the loss function based on the coverage vector.

**Benefits**:

* Improves the coverage and diversity of the generated output.

**12. ROUGE Metric Evaluation**

**Definition**:  
ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used to evaluate the quality of summaries by comparing them to reference summaries.

**Types**:

* **ROUGE-N**: Measures the overlap of n-grams between the generated summary and the reference summary.
* **ROUGE-L**: Measures the longest common subsequence between the generated summary and the reference summary.

**Applications**:

* Summarization Evaluation
* Machine Translation Evaluation

**Benefits**:

* Provides a quantitative measure of summary quality based on overlap with reference summaries.