3 Autoencoder: Working Principle and Architecture

An **Autoencoder (AE)** is a type of neural network used for *unsupervised learning*, primarily for dimensionality reduction, feature extraction, and data denoising. It consists of two main components:

- **Encoder**: Compresses the input into a lower-dimensional latent representation.
- **Decoder**: Reconstructs the original input from the compressed representation.

The network is trained to minimize the reconstruction error, forcing it to capture essential features of the input data.

4 Working Principle of Autoencoders

4.1 Step 1: Encoding

The encoder transforms the input X into a latent representation Z:

$$Z = f(W_e X + b_e) \tag{39}$$

where:

- W_e and b_e are the encoder weights and biases.
- $f(\cdot)$ is the activation function (e.g., ReLU, Sigmoid).

4.2 Step 2: Bottleneck (Latent Space)

The latent space contains the compressed version of the input, enforcing dimensionality reduction.

4.3 Step 3: Decoding

The decoder reconstructs the original input from the latent representation:

$$\hat{X} = g(W_d Z + b_d) \tag{40}$$

where:

- W_d and b_d are the decoder weights and biases.
- $g(\cdot)$ is the activation function.

4.4 Step 4: Loss Function

The goal of training is to minimize the reconstruction error, often measured using Mean Squared Error (MSE):

$$L = ||X - \hat{X}||^2 \tag{41}$$

where:

- X is the original input.
- $\hat{X}$ is the reconstructed output.

5 Autoencoder Architecture

A basic autoencoder consists of the following layers:

Layer	Function
Input Layer	Takes the original data (e.g., images, text, audio)
Encoder	Compresses input into a lower-dimensional representation
Bottleneck (Latent Space)	Stores the compressed feature representation
Decoder	Reconstructs the input from the latent space
Output Layer	Outputs a reconstruction of the original input

6 Mathematical Representation

The overall transformation can be represented as:

$$\hat{X} = g(f(X)) \tag{42}$$

where:

- $f(X)$ represents the **Encoder** transformation.
- $g(Z)$ represents the **Decoder** transformation.
- $\hat{X}$ is the reconstructed input.

7 Types of Autoencoders

Autoencoders can be modified for different applications:

- **Vanilla Autoencoder**: Basic encoder-decoder structure with MSE loss.
- **Denoising Autoencoder (DAE)**: Learns to reconstruct input from a corrupted version.

- **Sparse Autoencoder**: Uses sparsity constraints on latent representation.
- **Variational Autoencoder (VAE)**: Learns a probabilistic latent space for generating new data.
- **Convolutional Autoencoder (CAE)**: Uses CNNs for image-based feature extraction.

8 Applications of Autoencoders

Autoencoders are widely used for:

- **Dimensionality Reduction** (alternative to PCA)
- **Data Denoising** (removing noise from images, signals)
- **Feature Extraction** (unsupervised representation learning)
- **Anomaly Detection** (fraud detection, cybersecurity)
- **Image Compression** (reducing storage size)
- **Generative Modeling** (Variational Autoencoders for image synthesis)