1. Is an autoencoder for supervised learning or for unsupervised learning?

🔹 **Autoencoders** are primarily used for **unsupervised learning**

✅ **Why?**  
They don’t require labeled data. The model learns to **reconstruct the input from itself** by encoding and then decoding it. The training goal is to minimize reconstruction loss.

1. Explain briefly. 2. List the difference between Boltzmann Machine and Deep Belief Network.

| **Feature** | **Boltzmann Machine (BM)** | **Deep Belief Network (DBN)** |
| --- | --- | --- |
| **Structure** | Fully connected, stochastic neural network | Stack of **Restricted Boltzmann Machines (RBMs)** |
| **Training** | Difficult to train due to complex connections | Layer-wise pre-training using greedy unsupervised learning |
| **Efficiency** | Slow and computationally expensive | More efficient and scalable |
| **Use Case** | Theoretical, limited practical use | Feature extraction, pre-training deep networks |

1. How does the variational auto-encoder architecture allow it to generate new data points, compared to auto-encoder, which cannot generate new data points?

🔹 **Standard Autoencoder**: Learns a deterministic mapping (input → latent → output). The latent space isn’t continuous or well-structured → **hard to sample** new data points.

🔹 **VAE**: Learns a **probability distribution** in latent space.  
✅ Instead of encoding an input to a point, it encodes to a **distribution (mean and variance)**. Sampling from this distribution allows generating **new data**.

1. Why do autoencoders fail to generate realistic new data, and how do VAEs overcome this limitation?

🔻 **Autoencoders fail** because:

* They don’t enforce structure in the latent space.
* The latent space may be sparse and discontinuous → **poor sampling** → unrealistic outputs.

✅ **VAEs overcome this by**:

* Forcing the latent space to follow a known distribution (usually Gaussian).
* Ensuring similar inputs map to nearby points in latent space → allows **smooth and realistic generation**.

1. Generative Adversarial Networks(GANs) include a generator and a discriminator. Sketch a basic GAN using those elements, a source of real images, and a source of randomness.

[Random Noise z]

│

┌─────▼──────┐

│ Generator │

└─────┬──────┘

▼

[Generated (Fake) Image]

▼

┌─────────┴─────────┐

│ Discriminator │

└─────────┬─────────┘

▲

│

[Real Image Dataset]

1. The word “adversarial” in the acronym for GANs suggests a two-player game. What are the two players, and what are their respective goals?

GANs are called **“adversarial”** because they simulate a **two-player game**:

| **Player** | **Goal** |
| --- | --- |
| **Generator** | Create realistic fake data to **fool the discriminator** |
| **Discriminator** | Accurately **detect fake vs real** data |

🎯 It’s a **minimax game**:

* The **generator minimizes** the probability of being caught.
* The **discriminator maximizes** the probability of catching fakes.

The competition drives both networks to improve, leading to highly realistic data generation.

1. Explain auto encoder with an example.

🔹 **Autoencoder** is a type of neural network that learns to **compress** data (encoding) and then **reconstruct** it (decoding).

**🧠 Architecture:**

* **Input Layer → Encoder → Bottleneck → Decoder → Output Layer**
* Objective: Minimize reconstruction loss between input and output.

**✅ Example:**

* Input: Image of digit '5' (28×28 pixels)
* Encoder compresses the image to a 32-dimensional vector.
* Decoder reconstructs the image from this compressed representation.

🛠️ Use Case: Dimensionality reduction, anomaly detection, image compression.

1. Explain Generative Adversarial Networks using suitable diagram.How can GANs help in improving model performance on imbalanced datasets?

**Generative Adversarial Networks (GANs)**

It consists of:

* **Generator (G)**: Learns to create realistic fake data from random noise.
* **Discriminator (D)**: Learns to differentiate between real and fake data.

**📊 Diagram:**

css

Copy code

[Random Noise z] [Real Data x]

│ │

┌──────▼──────┐ ┌──────▼──────┐

│ Generator G │ │Discriminator│

└──────┬──────┘ └──────┬──────┘

▼ ▼

[Generated Fake Data] [Real vs Fake Decision]

**🎯 Use in Imbalanced Datasets:**

* GANs can **generate synthetic samples** for minority classes → balances the dataset → improves classifier performance.

1. Compare Denoising Autoencoders and Regularized Autoencoders.
2. **9. Compare Denoising vs Regularized Autoencoders**

| **Feature** | **Denoising Autoencoder** | **Regularized Autoencoder** |
| --- | --- | --- |
| Purpose | Learn robust representations by removing noise | Prevent overfitting via constraints |
| Technique | Input is corrupted (noise added) and model learns to reconstruct original | Adds regularization (like sparsity, weight penalties) to the loss |
| Use Case | Image denoising, anomaly detection | Feature learning, sparse encoding |

1. Explain the concept of Variational Autoencoders.

🔹 **VAE** is a probabilistic autoencoder that **learns a distribution** over the latent space, not just a deterministic mapping.

**🧠 Key Features:**

* Encodes input into mean (μ) and standard deviation (σ).
* Samples from latent space: z = μ + σ \* ε, where ε ~ N(0,1)
* Ensures smooth and continuous latent space.

✅ **Why important?**  
Allows generation of new data by sampling latent varia

1. Explain the Deep belief Networks and their significance in the field of deep learning.

🔹 A **DBN** is a stack of **Restricted Boltzmann Machines (RBMs)**, where each RBM layer is trained greedily in an unsupervised manner.

**🧠 Structure:**

cpp

Copy code

Input → RBM1 → RBM2 → ... → Classifier (optional)

**✅ Significance:**

* Captures complex patterns layer-by-layer.
* Used for **feature learning**, **pretraining** deep neural networks before fine-tuning.
* Especially useful when labeled data is limited.

1. Discuss the applications of GAN.

c📌 **Real-world Uses**:

* **Image Generation**: Creating realistic faces, artworks (e.g., ThisPersonDoesNotExist.com)
* **Data Augmentation**: For imbalanced datasets.
* **Super-Resolution**: Enhancing image quality (e.g., ESRGAN)
* **Image-to-Image Translation**: Convert sketches to photos (e.g., pix2pix, CycleGAN)
* **Medical Imaging**: Generate synthetic scans for training.
* **Text-to-Image Synthesis**: Generate images from textual descriptions.

1. Describe Boltzmann Machines. How do they learn and generate samples in a probabilistic manner?

**Boltzmann Machine** is a **stochastic recurrent neural network** that learns probability distributions over binary variables.

**🧠 Structure:**

* Fully connected neurons (visible + hidden units)
* Units are binary and probabilistic.

**🧠 Learning:**

* Uses **energy-based models**.
* Goal: Minimize energy for probable states.
* Trained using **Stochastic Gradient Descent** and algorithms like Contrastive Divergence.

**🔄 Sampling:**

* After training, you can **sample new data** from the learned distribution using **Gibbs Sampling**.

✅ **Use Cases**: Dimensionality reduction, collaborative filtering, pretraining for DBNs.

1. Explain Denoising Autoencoders.

🔹 **Denoising Autoencoders** (DAEs) are a variant of autoencoders that learn to reconstruct the original input from a **noisy version** of it.

**🧠 How It Works:**

1. Add **random noise** to the input (e.g., Gaussian noise).
2. The autoencoder is trained to **recover the original clean input** from the noisy version.
3. Helps in learning robust features that generalize well.

**✅ Example:**

* Original input: "image of digit 7"
* Noisy input: blurred or pixel-corrupted image
* Output: reconstructed clean "digit 7"

**📌 Applications:**

* Image denoising
* Robust feature extraction
* Anomaly detection

1. List out the applications of GAN. With the help of a diagram, explain the training process of Generative Adversarial Networks (GANs) and the adversarial relationship between Generator and Discriminator.

**✅ Applications of GANs:**

* **Image Generation**: Face generation (e.g., StyleGAN)
* **Data Augmentation**: Creating synthetic data for training
* **Image-to-Image Translation**: Convert sketches → photos (e.g., pix2pix)
* **Super-resolution**: Increase image resolution (e.g., SRGAN)
* **Medical Imaging**: Synthetic scans for rare conditions
* **Video Prediction**: Forecasting future video frames
* **Art and Creativity**: AI-generated music, art, and design

**🧠 GAN Training Process & Adversarial Relationship**

**Components:**

* **Generator (G)**: Tries to produce realistic fake data from noise.
* **Discriminator (D)**: Tries to distinguish between real and fake data.

**Diagram:**

mathematica

Copy code

[Random Noise z] ─▶ Generator G ─▶ Fake Image ─┐

▼

Discriminator D ◀─ Real Images

│

Output: Real or Fake?

**🎯 Adversarial Game:**

* **Generator** tries to **fool** the Discriminator.
* **Discriminator** tries to correctly identify fake vs. real.
* Training is a **minimax game**:
  + G tries to **minimize** log(1 − D(G(z)))
  + D tries to **maximize** log(D(x)) + log(1 − D(G(z)))

1. Compare Boltzmann machine and traditional neural network?

| **Feature** | **Boltzmann Machine** | **Traditional Neural Network** |
| --- | --- | --- |
| Structure | Fully connected undirected graph | Feedforward (directed) layers |
| Activation | Stochastic (probabilistic) binary states | Deterministic (ReLU, sigmoid, etc.) |
| Learning | Unsupervised (energy minimization) | Mostly supervised (error backpropagation) |
| Training Algorithm | Contrastive Divergence, Gibbs Sampling | Backpropagation |
| Usage | Dimensionality reduction, sampling | Classification, regression |
| Interpretability | Models joint probability distribution | Models mapping from input to output |

1. Explain the basic idea behind generative models and how they differ from discriminative models.

**🔹 Generative Models:**

* Learn **joint probability**: P(x, y)
* Can generate new data instances
* Examples: GANs, VAEs, Naive Bayes

**🔹 Discriminative Models:**

* Learn **conditional probability**: P(y | x)
* Focus on classification or prediction
* Examples: Logistic Regression, SVM, Neural Networks

| **Aspect** | **Generative Models** | **Discriminative Models** |
| --- | --- | --- |
| Learns | P(x, y) | P(y |
| Generates | Yes | No |
| Goal | Model data distribution | Predict output class |

1. Compare undercomplete autoencoders and regularized autoencoders.

| **Feature** | **Undercomplete Autoencoder** | **Regularized Autoencoder** |
| --- | --- | --- |
| Latent Space | Small (bottleneck) | May not restrict size |
| How It Prevents Overfitting | By limiting encoding capacity | By applying regularization (sparsity, dropout) |
| Loss Function | Just reconstruction loss | Reconstruction + regularization term |
| Goal | Dimensionality reduction | Robust feature learning |
| Example Regularizations | – | L1, L2, sparsity penalty, noise injection |