→ The no. of O/p units in the O/p layer is equal to the no. of I/p " " " I/p ".

Note Auto encoder

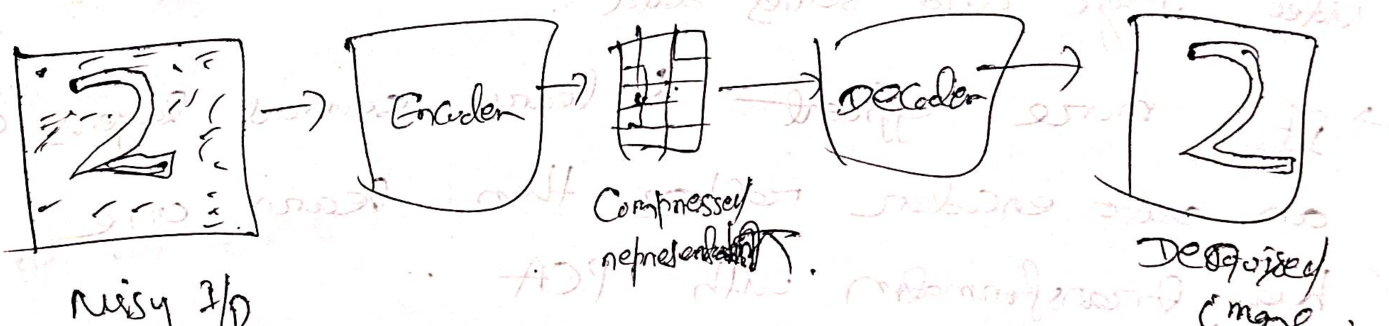
An auto encoder is a type of unsupervised ANN designed to learn efficient (compressed) representations of I/p data and reconstruct the original I/p from those representations.

Characteristics

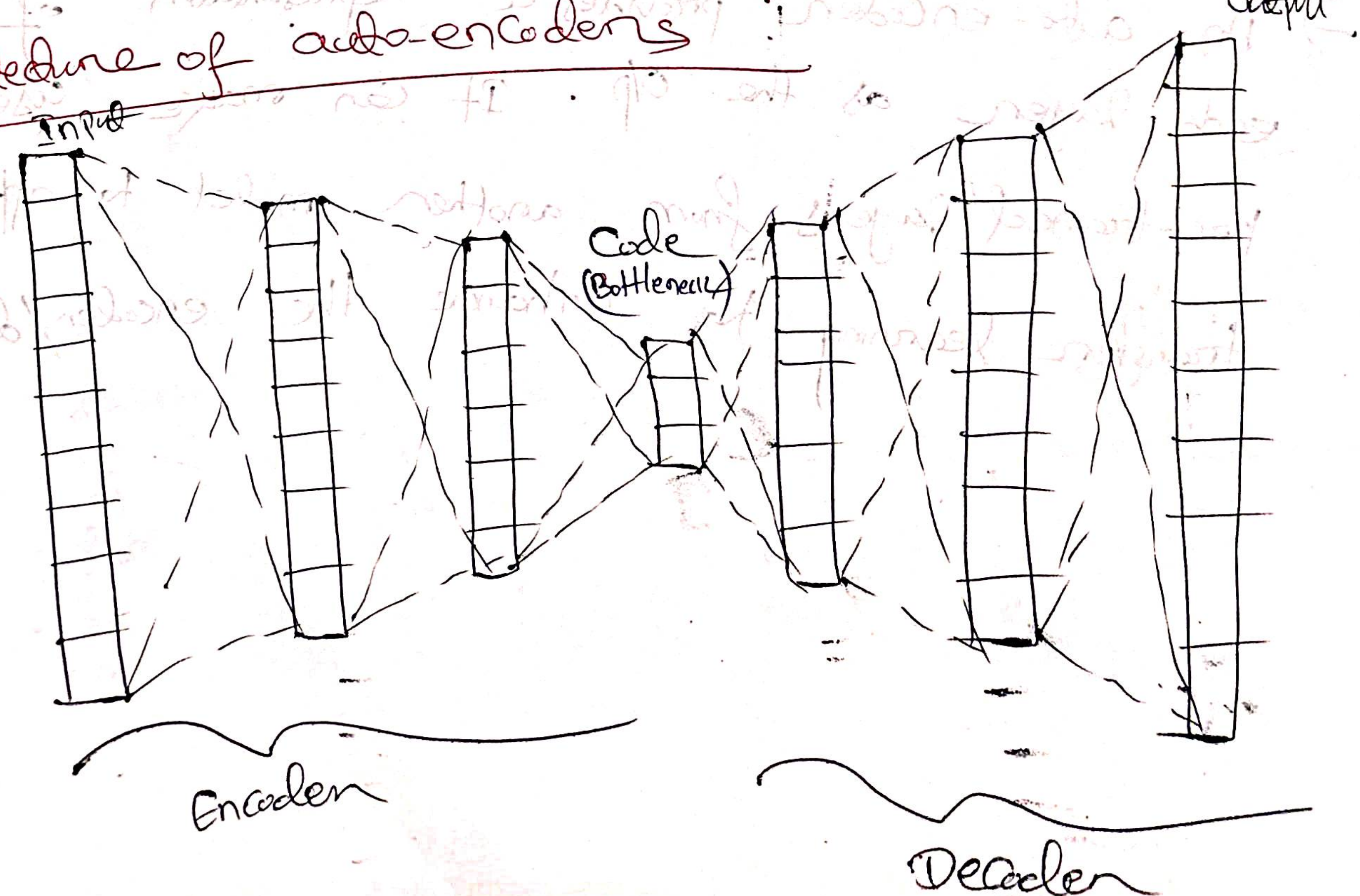
→ It learns without labeled data.
→ It compresses data into a lower-dimensional representation (latent space) and then reconstruct it
→ It learns to preserve the most essential features of the I/p.

Applications of Autoencoders :

- → Image Coloring :- Converting any black & white picture into a colored image
- → Feature Variation :- extracts only the required features and generates the o/p by removing noise or unnecessary interruptions.
- → Dimensionality reduction :- * reconstructed image is the same as our I/p but with reduced dimension
  * similar image with reduced pixel value.
- → Image denoising :- reconstruct the original I/p from the noisy version.
- → Watermark removal :- removing watermarks from images
  - remove any object while filming a video.
- → Anomaly detection (outlier detection)
- → Sequence to sequence prediction
- → Recommendation systems.



Architecture of auto-encoders :



→ An auto-encoder consists of 3 layers.

(i) Encoder
(ii) code
(iii) decoder

(i) Encoder :

→ The N/w compresses the I/p into a latent space representation.

→ The encoder layer encodes the I/p image as a compressed representation in a reduced dimension.

→ The compressed image is the distorted version of the original image.

(ii) Code :

This part of the N/w represents the compressed I/p which is fed to the decoder.

(iii) Decoder :

→ This layer decodes the encoded image back to the original dimension

→ The decoded image is a lossy reconstruction of the original image and it is reconstructed from the latent space representation.

Hyper-parameters of auto-encoders :

⇒ Code size :

→ It represents the no. of nodes in the middle layer.

→ smaller size results in more compression.

⇒ No. of layers :

The auto-encoder can consist of as many layers as we want.

⇒ No. of nodes per layer :

→ The no. of nodes per layer decreases with each subsequent layer of the encoder, and increases back in the decoder.

→ The decoder is symmetric to the encoder in terms of the layer structure.

⇒ Loss function :

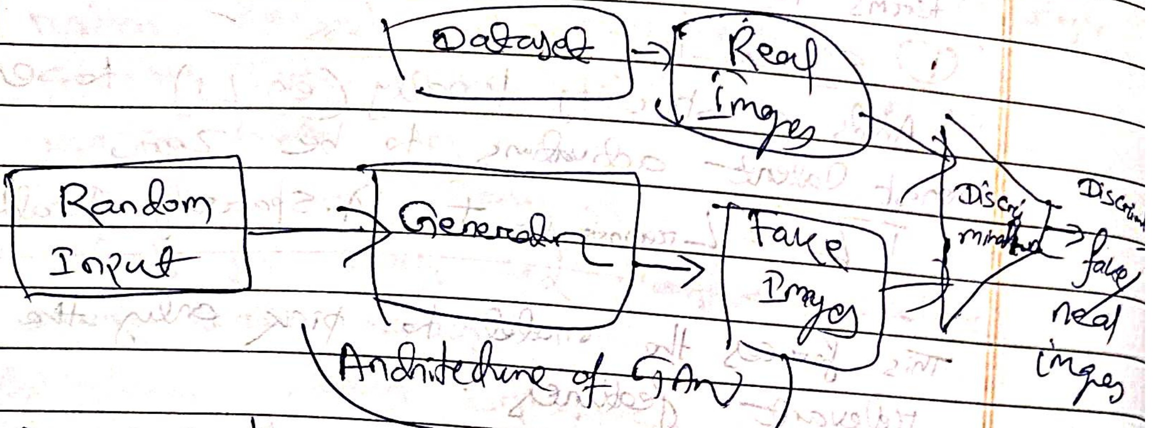
→ We either use MSE or binary-cross-entropy.

→ If the I/p values are in the range [0, 1], then we typically use cross-entropy, otherwise, we use MSE.

Chapter-10

GAN (Generative Adversarial Network)

It is a form of generative model designed to produce realistic samples of entities such as images, from noise.



Architecture of GAN

→ A GAN has 2 main Components.

(i) Generator
(ii) Discriminator

(i) Generator :

→ It creates fake data (e.g. images) from random noise.
→ Its goal is to make the data look as real as possible to fool the discriminator.

(ii) Discriminator :

→ It evaluates data and decides whether it is real (from the dataset) or fake (from the generator)?

Adversarial Training :

→ It refers to a training process in which 2 models are trained simultaneously in competition with each other, with opposing objectives.

* Generator tries to produce realistic fake data to fool the discriminator.

* The discriminator tries to correctly distinguish between real and fake data.

→ This creates a minimax game.

* The Generator minimizes the discriminator's ability to detect fake samples.
* The Discriminator maximizes its accuracy in identifying real vs fake samples.

Generator Loss & Discriminator Loss

(i) Generator Loss

→ It measures how well the generator fools the discriminator.

Mathematically, $\boxed{L_G = -E_{z \sim P_z}\left[\log D(G(z))\right]}$

→ It penalizes the generator when the discriminator correctly identifies generated samples as fake.

(ii) Discriminator Loss:

→ It measures how well the discriminator distinguishes real data from fake data.

Mathematically, $$\boxed{L_D = -E_{x \sim P_{data}}\left[\log D(x)\right] - E_{z \sim P_z}\left[\log\left(1 - D(G(z))\right)\right]}$$

→ It penalizes the discriminator when

* Real samples are classified as fake.
* Fake " " " as real.

Date
Page No.

Key challenges of GAN

→ Training instability
→ Mode collapse (Generator produces limited varieties of o/p)
→ sensitive to hyper parameters.

Applications of GAN

→ Image Generation
→ Data Augmentation
→ Medical imaging synthesis
→ Super resolution and image enhancement
→ Game content generation.