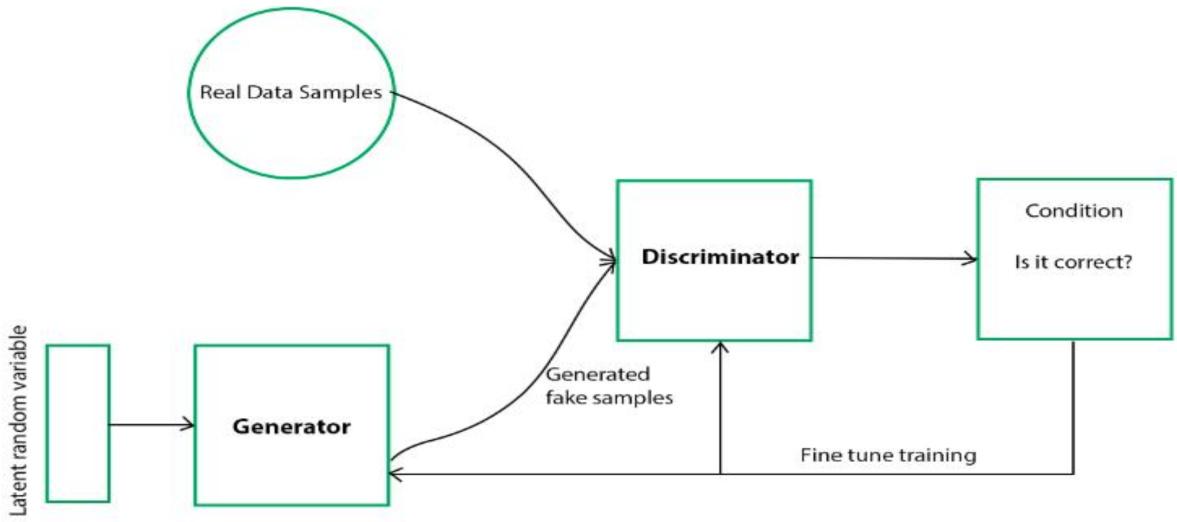
Semester: VIII Subject: Advanced AI Academic Year:2024-2025

Module 2

Generative Adversarial Networks (GAN):

-A generative adversarial network (GAN) is a machine learning (ML) model in which two neural networks compete with each other by using deep learning methods to become more accurate in their predictions.

- The two neural networks that make up a GAN are referred to as the generator and the discriminator.



How GANs work?

GANs are typically divided into the following three categories:

-Generative - This describes how data is generated in terms of a probabilistic model.

-Adversarial - A model is trained in an adversarial setting.

-Networks - Deep neural networks can be used as artificial intelligence (AI) algorithms for training purposes.

-The generator network takes random input (typically noise) and generates samples, such as images, text, or audio, that resemble the training data it was trained on. The goal of the generator is to produce samples that are indistinguishable from real data.

- The discriminator network, on the other hand, tries to distinguish between real and generated samples. It is trained with real samples from the training data and generated samples from the generator. The discriminator's objective is to correctly classify real data as real and generated data as fake.

- The training process involves an adversarial game between the generator and the discriminator. The generator aims to produce samples that fool the discriminator, while the discriminator tries to improve its ability to distinguish between real and generated data. This adversarial training pushes both networks to improve over time.

Training & Prediction of Generative Adversarial Networks (GANs):

Step-1) Define a Problem

GANs work with a different set of problems you are aiming so you need to define What you are creating like audio, poem, text, Image is a type of problem.

Step-2) Select Architecture of GAN

we have to define which type of GAN architecture we are using.

Step-3) Train Discriminator on Real Dataset

And the Data you are providing is without Noise and only contains real images, and for fake images.

-It classifies both real and fake data.

-The discriminator loss helps improve its performance and penalize it when it misclassifies real as fake or vice-versa.

-weights of the discriminator are updated through discriminator loss.

Step-4) Train Generator

-get random noise and produce a generator output on noise sample

-predict generator output from discriminator as original or fake.

-we calculate discriminator loss.

-perform backpropagation through discriminator, and generator both to calculate gradients.

-Use gradients to update generator weights.

Step-5) Train Discriminator on Fake Data

The samples which are generated by Generator will pass to Discriminator and It will predict the data passed to it is Fake or real and provide feedback to Generator again.

Step-6) Train Generator with the output of Discriminator

Again Generator will be trained on the feedback given by Discriminator and try to improve performance.

This is an iterative process and continues running until the Generator is not successful in making the discriminator fool.

Generative Adversarial Networks (GANs) Loss Function:

The generator tries to minimize the following loss function while the discriminator tries to maximize it. It is the same as a minimax game if you have ever played.

$$\min_{G} \max_{D} V(D, G)$$

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

- D(x) is the discriminator's estimate of the probability that real data instance x is real.
- E_x is the expected value over all real data instances.
- G(z) is the generator's output when given noise z.
- D(G(z)) is the discriminator's estimate of the probability that a fake instance is real.
- E_z is the expected value over all random inputs to the generator (in effect, the expected value over all generated fake instances G(z)).
- The formula is derived from cross-entropy between