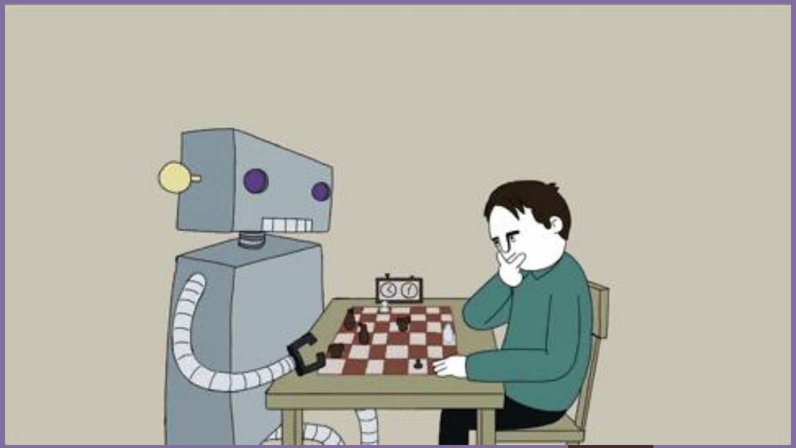
CSE4006 DEEP LEARNING



Dr K G Suma
Associate Professor
School of Computer Science and Engineering

Module No. 6
VAEs and GANS
9 Hours

- Variational Autoencoders
- Generative Adversarial Networks
- Multi-task Deep Learning
- Multi-view Deep Learning

- Various Applications - speech, text, image and video

Generative Adversarial Networks

- Generative Adversarial Networks (GANs) were developed in 2014 by Ian Goodfellow and his teammates.
- GAN is basically an approach to generative modeling that generates a new set of data based on training data that look like training data.
- ANs have two main blocks(two neural networks) which compete with each other and are able to capture, copy, and analyze the variations in a **dataset**.

Generative Adversarial Networks

GAN let’s break it into separate three parts

- **Generative** – Learn a generative model that describes how data is generated using a probabilistic approach. In simple terms, it visually explains data generation.
- **Adversarial** – The training of the model is done in an adversarial (Oppositional) setting.
- **Networks** – use deep neural networks for training purposes.

Generative Adversarial Networks

GAN Techniques have shown impressive results in various domains like image synthesis, text generation, and video generation, enhancing the field of generative modeling and enabling new creative applications in artificial intelligence.

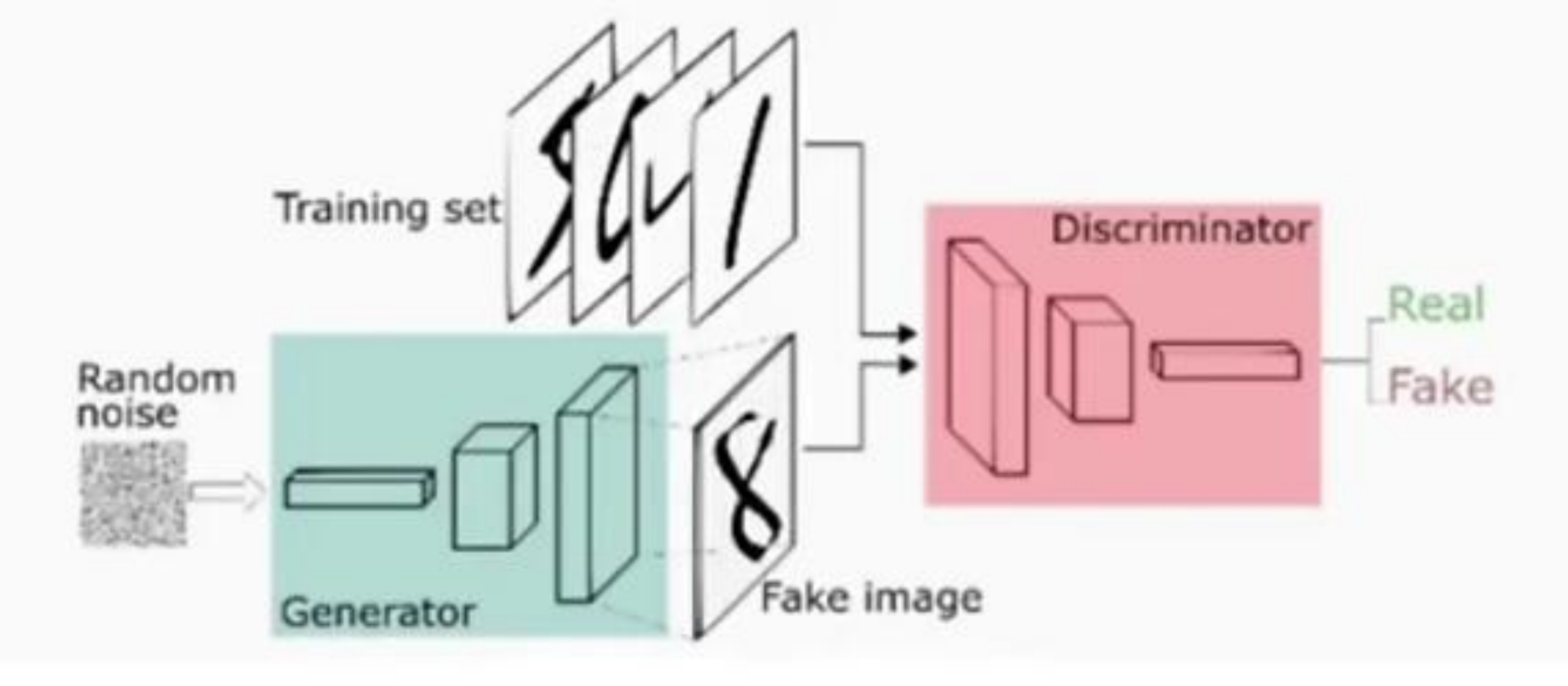
- **Generator Network**: Takes random input to generate samples resembling training data.
- **Discriminator Network**: Distinguishes between real and generated samples.
- **Adversarial Training**: Generator tries to fool the discriminator, and the discriminator improves its distinguishing skills.
- **Progression**: Generator produces more realistic samples; discriminator becomes better at identifying them.
- **Applications**: Image synthesis, text generation, video generation, creating deepfakes, enhancing low-resolution images.

Over time, the generator becomes better at creating realistic samples, and the discriminator becomes more skilled at identifying them. This process ideally leads to the generator producing high-quality samples that are hard to distinguish from real data.

Why GANs was Developed?

Machine learning algorithms and Deep neural networks can easily be fooled to misclassify things by adding some amount of noise to data. After adding some amount of noise, the chances of misclassifying the images increase.

Hence the small rise that, is it possible to implement something that neural networks can start visualizing new patterns like sample train data. Thus GANs were built that generate new fake results similar to the original.


Training set
Discriminator
Real
Fake
Random noise
Generator
Fake image

Components of GAN

Generator Network

The generator network's purpose is to generate new data samples that resemble the training data. It takes random noise as input and produces various types of data samples, such as images, text, or audio. The primary objective of the generator is to fool the discriminator by creating data samples that are realistic enough to be indistinguishable from real data.

The generator is a deep neural network that takes random noise as input to generate realistic data samples (e.g., images or text). It learns the underlying data distribution by adjusting its parameters through backpropagation.

- The generator's objective is to produce samples that the discriminator classifies as real. The loss function is:

Components of GAN

Generator Network

$$J_G = -\frac{1}{m}\Sigma_{i=1}^{m} logD(G(z_i))$$

Where,

- J_G measure how well the generator is fooling the discriminator.
- log $D(G(z_i))$represents log probability of the discriminator being correct for generated samples.
- The generator aims to minimize this loss, encouraging the production of samples that the discriminator classifies as real $(logD(G(z_i))$, close to 1.

Components of GAN

Discriminator Network

The discriminator network's purpose is to distinguish between real data and data generated by the generator. It receives both real data from the training set and fake data from the generator as input. The output is a probability indicating whether the input data is real or fake. The primary objective of the discriminator is to correctly identify real versus generated data.

The **discriminator** acts as a **binary classifier**, distinguishing between real and generated data. It learns to improve its classification ability through training, refining its parameters to **detect fake samples more accurately**.

When dealing with image data, the discriminator often employs convolutional layers or other relevant architectures suited to the data type. These layers help extract features and enhance the model's ability to differentiate between real and generated samples.

The discriminator reduces the negative log likelihood of correctly classifying both produced and real samples. This loss incentivizes the discriminator to accurately categorize generated samples as fake and real samples with the following equation:

Components of GAN

Dicriminator Network

$$J_D = -\frac{1}{m}\Sigma_{i=1}^{m} log\ D(x_i) - \frac{1}{m}\Sigma_{i=1}^{m} log(1 - D(G(z_i))$$

- J_D assesses the discriminator's ability to discern between produced and actual samples.
- The log likelihood that the discriminator will accurately categorize real data is represented by $logD(x_i)$.
- The log chance that the discriminator would correctly categorize generated samples as fake is represented by $log(1 - D(G(z_i)))$.

By **minimizing this loss**, the discriminator becomes more effective at distinguishing between real and generated samples.

Components of GAN

Adversarial Training Process

The adversarial training process aims to improve both the generator and the discriminator through a competitive dynamic. The generator’s goal is to enhance its ability to create realistic data that can fool the discriminator. Conversely, the discriminator’s goal is to improve its capability to distinguish between real and fake data. Through iterative improvement, both networks continuously advance as they learn from each other’s feedback.

Components of GAN

Loss Functions

Generator loss measures how well the generator fools the discriminator, typically aiming to minimize the discriminator's ability to detect fake data. Discriminator loss measures how well the discriminator differentiates between real and fake data, aiming to maximize correct classifications of real and fake samples.

MinMax Loss

GANs follow a minimax optimization where the generator and discriminator are adversaries:

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

Where,

- G is generator network and is D is the discriminator network
- Actual data samples obtained from the true data distribution $p_{data}(x)$ are represented by x.
- Random noise sampled from a previous distribution $p_z(z)$(usually a normal or uniform distribution) is represented by z.
- D(x) represents the discriminator's likelihood of correctly identifying actual data as real.
- D(G(z)) is the likelihood that the discriminator will identify generated data coming from the generator as authentic.

The generator aims to **minimize** the loss, while the discriminator tries to **maximize** its classification accuracy.

Components of GAN

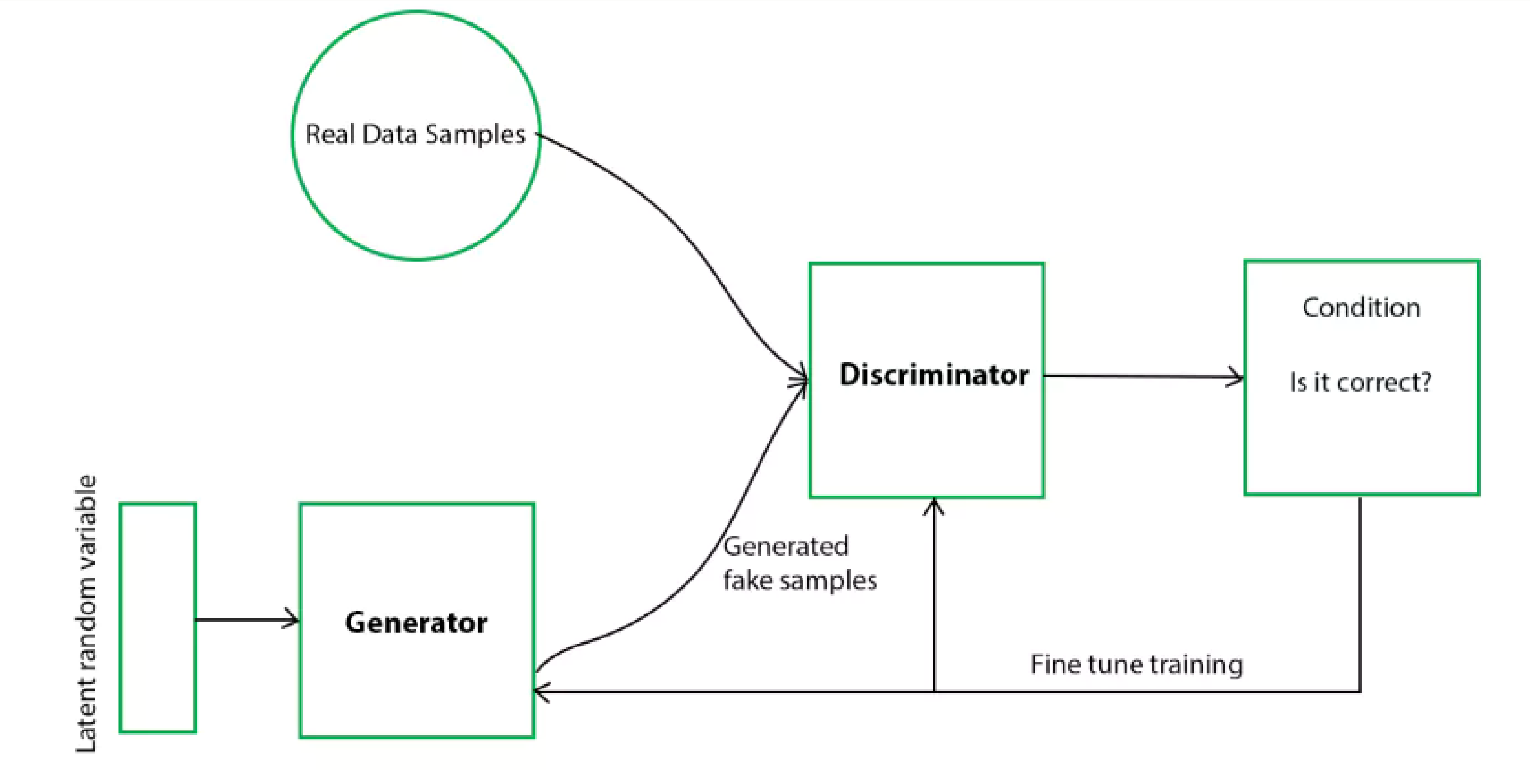
Random Noise Input

Random noise serves as the input to the generator network. It is usually a vector of random values sampled from a uniform or **normal distribution**. This noise provides the generator with a diverse set of inputs, enabling it to produce varied data samples.

Training Data

Real data consists of genuine data samples from the domain of interest. Fake data refers to the samples generated by the generator during training.

Geometric Intuition behind the working of GANs

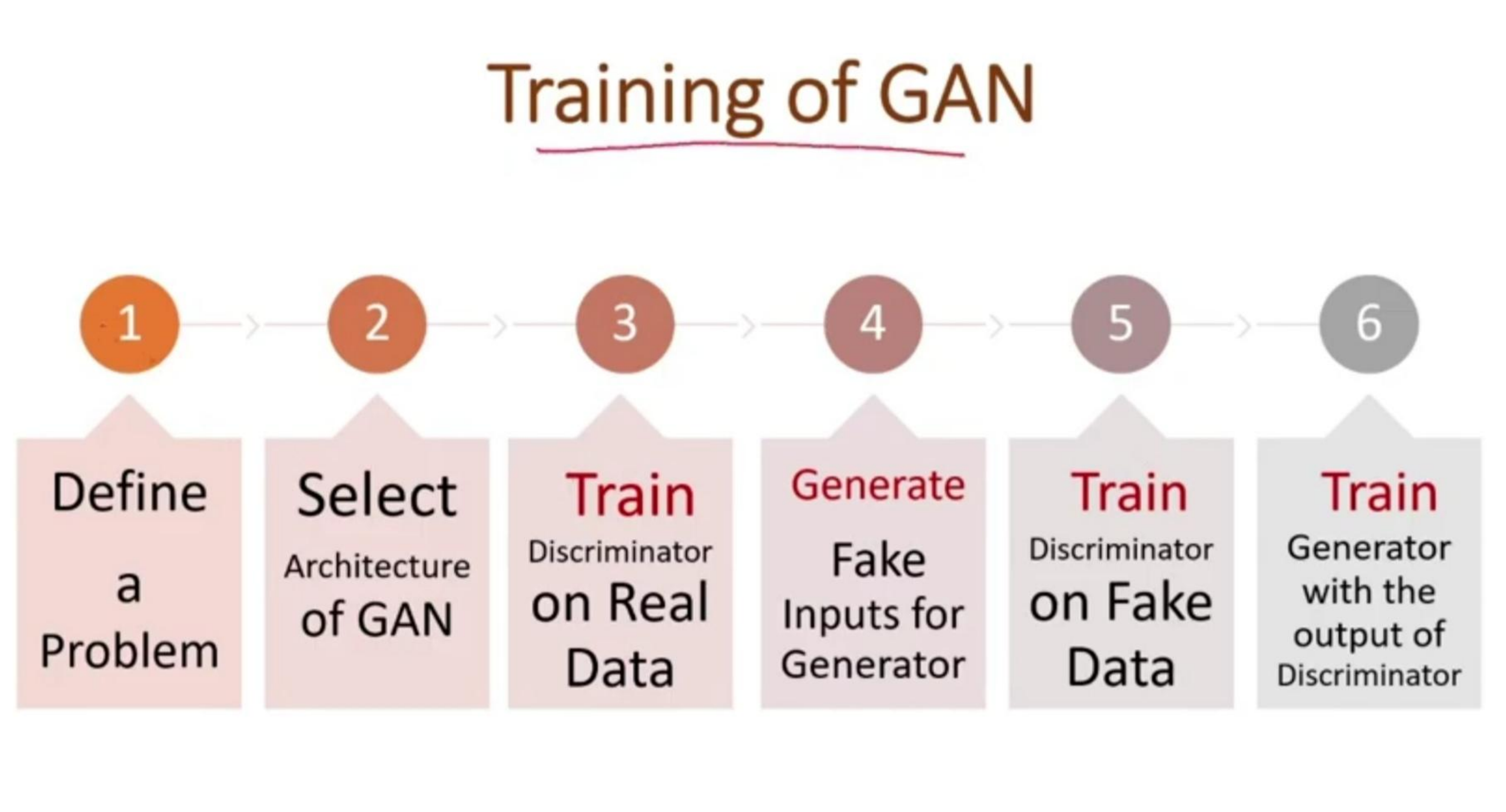


Geometric Intuition behind the working of GANs

Here the generative model captures the distribution of data and is trained in such a manner to generate the new sample that tries to maximize the probability of the discriminator to make a mistake (maximize discriminator loss). The discriminator on other hand is based on a model that estimates the probability that the sample it receives is from training data not from the generator and tries to classify it accurately and minimize the GAN accuracy. Hence, The GAN architecture formulates as a minimax game where the Discriminator tries to minimize its reward V(D,G) while the Generator aims to maximize the Discriminator's loss.

- We know that both components are neural networks. we can see that generator output is directly connected to the input of the discriminator. And discriminator predicts it and through backpropagation, the generator receives a feedback signal to update weights and improve performance. The discriminator is a feed-forward neural network.

Training & Prediction of Generative Adversarial Networks (GANs)



Challenges Faced GANs

- The problem of stability between generator and discriminator. We do not want that discriminator should be too strict, we want to be lenient
- Problem to determine the positioning of objects. suppose in a picture we have 3 horse and generator have created 6 eyes and 1 horse.
- The problem in understanding the global objects – GANs do not understand the global structure or holistic structure which is similar to the problem of perspective. It means sometimes GAN Architecture generates an image that is unrealistic and cannot be possible.
- A problem in understanding the perspective – It cannot understand the 3-d images and if we train it on such types of images then it will fail to create 3-d images because today GANs are capable to work on 1-d images.

Types of GANs

Vanilla GAN: Vanilla GAN is the simplest type of GAN. It consists of:

- A generator and a discriminator, both are built using multi-layer perceptrons (MLPs).
- The model optimizes its mathematical formulation using stochastic gradient descent (SGD).

While Vanilla GANs serve as the foundation for more advanced GAN models, they often struggle with issues like mode collapse and unstable training.

DC GAN (Deep Convolutional GAN) : It is a Deep convolutional GAN Techniques. It is one of the most used, powerful, and successful types of GAN architecture. It is implemented with help of ConvNets in place of a Multi-layered perceptron. The ConvNets use a convolutional stride and do not include max pooling. Additionally, the layers in this network are not fully connected

- Uses Convolutional Neural Networks (CNNs) instead of simple multi-layer perceptrons (MLPs).
- Max pooling layers are replaced with convolutional stride, making the model more efficient.
- Fully connected layers are removed, allowing for better spatial understanding of images.

Types of GANs

Conditional GAN and Unconditional GAN (CGAN): A Conditional GAN is a deep learning neural network that uses additional parameters. The inputs to the Discriminator also include labels that help it classify the input correctly and avoid easily being fooled by the generator.

- A conditional variable (y) is fed into both the generator and the discriminator.
- This ensures that the generator creates data corresponding to the given condition (e.g., generating images of specific objects).
- The discriminator also receives the labels to help distinguish between real and fake data.

Least Square GAN (LSGAN): It is a type of GAN Techniques that adopts the least-square loss function for the discriminator. Minimizing the objective function of LSGAN results in minimizing the Pearson divergence.

Types of GANs

Auxilary Classifier GAN(ACGAN): It is the same as CGAN and an advanced version of it. It says that the Discriminator should not only classify the image as real or fake but should also provide the source or class label of the input image.

Dual Video Discriminator GAN: DVD-GAN is a generative adversarial network model for video generation built upon the BigGAN architecture. DVD-GAN uses two discriminators: a Spatial Discriminator and a Temporal Discriminator.

Info GAN: Advance version of GAN which is capable to learn to disentangle representation in an unsupervised learning approach.

Types of GANs

SRGAN Super Resolution GAN: Its main function is to transform low resolution to high resolution known as Domain Transformation.

- Uses a deep neural network combined with an adversarial loss function.
- Enhances low-resolution images by adding finer details, making them appear sharper and more realistic.
- Helps reduce common image upscaling errors, such as blurriness and pixelation.

Cycle GAN: It is released in 2017 which performs the task of Image Translation. Suppose we have trained it on a horse image dataset and we can translate it into zebra images.

Types of GANs

Laplacian Pyramid GAN (LAPGAN): Laplacian Pyramid GAN (LAPGAN) is designed to generate ultra-high-quality images by leveraging a multi-resolution approach.

Working of LAPGAN:

- Uses multiple generator-discriminator pairs at different levels of the Laplacian pyramid.
- Images are first downsampled at each layer of the pyramid and upscaled again using Conditional GANs (CGANs).
- This process allows the image to gradually refine details, reducing noise and improving clarity.

Due to its ability to generate highly detailed images, LAPGAN is considered a superior approach for photorealistic image generation.

Applications of GANs

1. Generate new data from available data – It means generating new samples from an available sample that is not similar to a real one.
2. Generate realistic pictures of people that have never existed.
3. Gans is not limited to Images, It can generate text, articles, songs, poems, etc.
4. Generate Music by using some clone Voice – If you provide some voice then GANs can generate a similar clone feature of it. Researchers from NIT in Tokyo proposed a system that is able to generate melodies from lyrics with help of learned relationships between notes and subjects.
5. Text to Image Generation (Object GAN and Object Driven GAN)
6. Creation of anime characters in Game Development and animation production.

Applications of GANs

7. Image to Image Translation – We can translate one Image to another without changing the background of the source image. For example, Gans can replace a dog with a cat.
8. Low resolution to High resolution – If you pass a low-resolution Image or video, GAN can produce a high-resolution Image version of the same.
9. Prediction of Next Frame in the video – By training a neural network on small frames of video, GANs are capable to generate or predict a small next frame of video. For example, you can have a look at below GIF
10. Interactive Image Generation – This means that GANs can generate images and video footage in an artistic form if they train on the right real dataset.
11. Speech – Researchers from the College of London recently published a system called GAN-TTS that learns to generate raw audio through training on 567 corpora of speech data.

Applications of GANs

1. **Image Synthesis & Generation:** GANs generate realistic images, avatars, and high-resolution visuals by learning patterns from training data. They are widely used in art, gaming, and AI-driven design.

2. **Image-to-Image Translation:** GANs can transform images between domains while preserving key features. Examples include converting day images to night, sketches to realistic images, or changing artistic styles.

3. **Text-to-Image Synthesis:** GANs create visuals from textual descriptions, enabling applications in AI-generated art, automated design, and content creation.

4. **Data Augmentation:** GANs generate synthetic data to improve machine learning models, making them more robust and generalizable, especially in fields with limited labeled data.

5. **High-Resolution Image Enhancement:** GANs upscale low-resolution images, improving clarity for applications like medical imaging, satellite imagery, and video enhancement.

Advantages of GAN

1. **Synthetic data generation**: GANs can generate new, synthetic data that resembles some known data distribution, which can be useful for data augmentation, anomaly detection, or creative applications.

2. **High-quality results**: GANs can produce high-quality, photorealistic results in image synthesis, video synthesis, music synthesis, and other tasks.

3. **Unsupervised learning**: GANs can be trained without labeled data, making them suitable for unsupervised learning tasks, where labeled data is scarce or difficult to obtain.

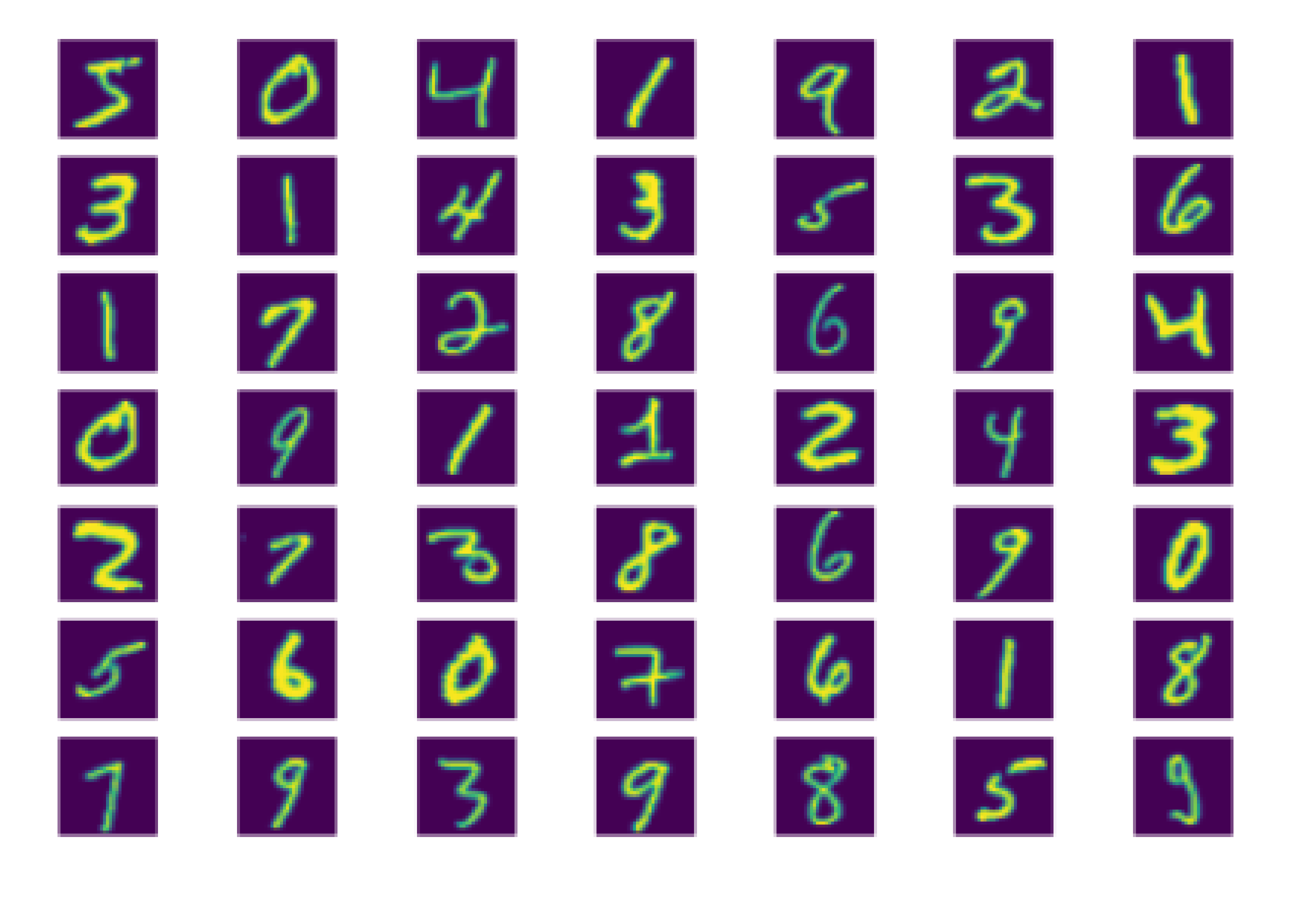
4. **Versatility**: GANs can be applied to a wide range of tasks, including image synthesis, text-to-image synthesis, image-to-image translation, anomaly detection, data augmentation, and others.

Steps to Implement a Basic GAN

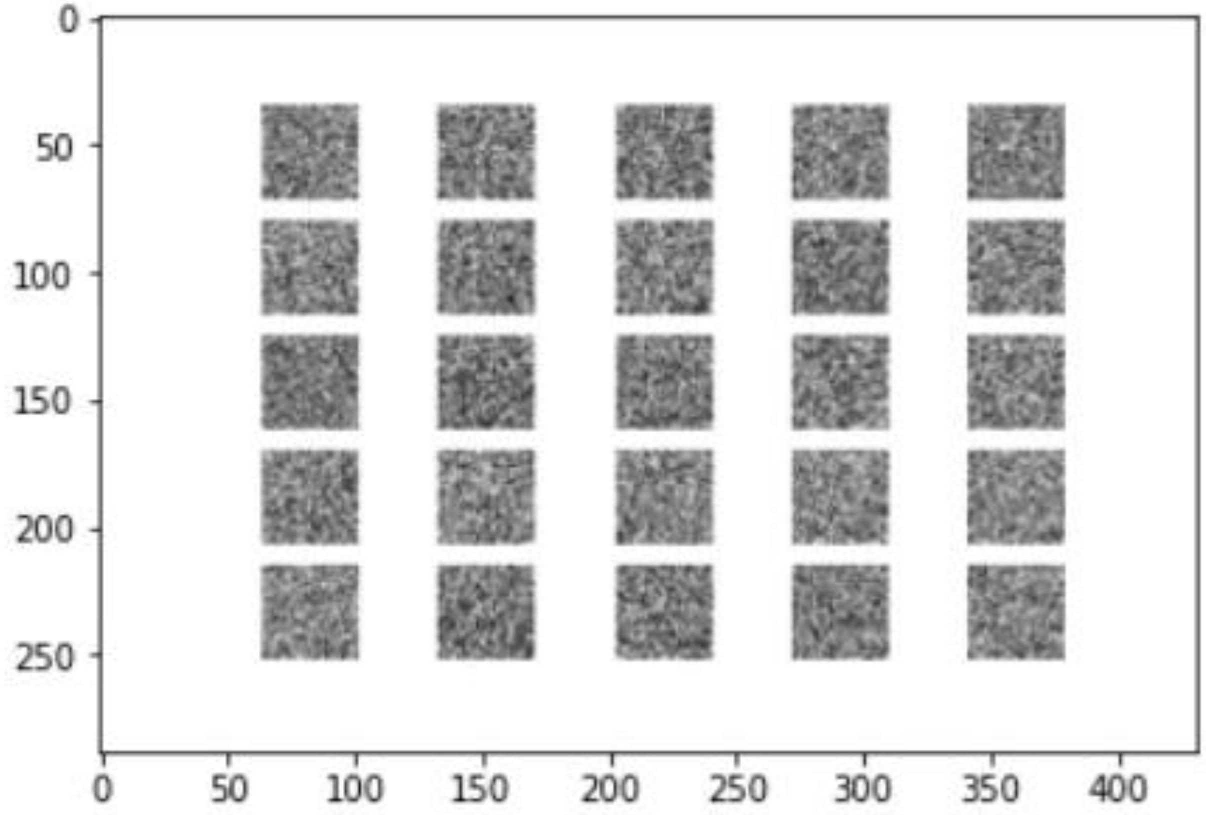
1. **Import Libraries**: Import all necessary libraries for the GAN implementation.
2. **Get the Dataset**: Acquire the dataset to be used for training the GAN.
3. **Data Preparation**: Preprocess the data, including steps such as scaling, flattening, and reshaping.
4. **Define Networks**: Create the Generator and Discriminator functions.
5. **Generate Initial Images**: Create images from random noise using the generator.
6. **Set Parameters**: Define parameters like epoch count, batch size, and sample size.
7. **Sample Image Generation**: Define the function to generate and display sample images during training.
8. **Training Process**: Train the discriminator, then the generator iteratively to improve generated images.
9. **Evaluate Results**: Assess the quality of images produced by the generator.

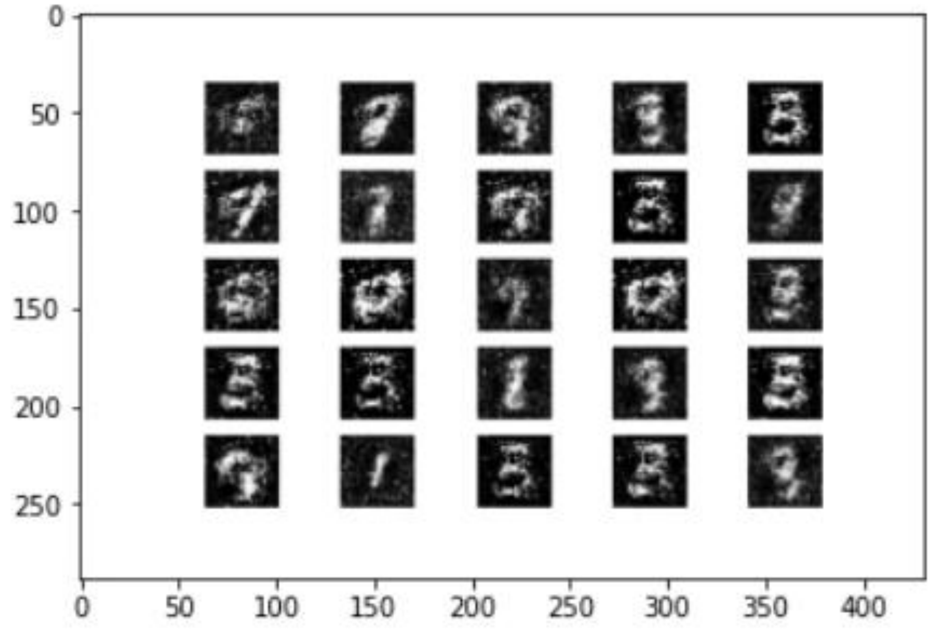
Practical Implementation of GANs on MNIST Dataset

Modified National Institute of Standards and Technology dataset



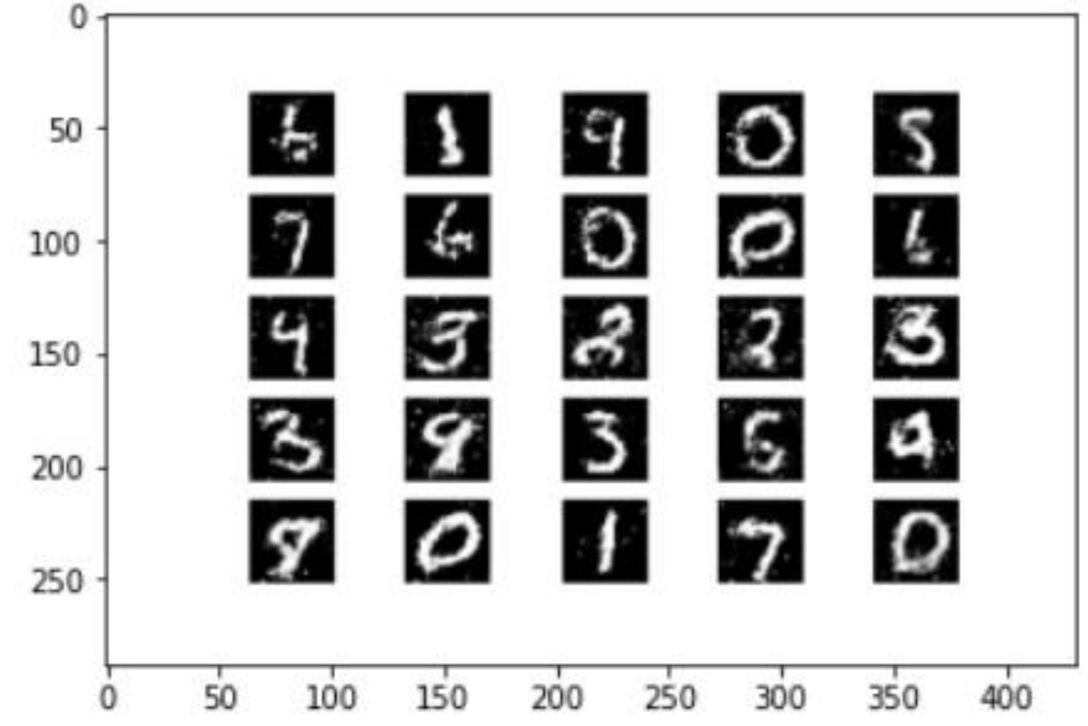
initial epoch what results in it being generated.





Now Generator is slowly being capable to extract some information that can be observed.

Plot Image Generated after training on 10000 Epochs



- Now Generator is capable to build as it is an image as of MNIST dataset and there are high chances of the Discriminator being Fool.

Conclusion

- Generative Adversarial Networks (GANs) represent a powerful paradigm in the field of machine learning, offering diverse applications and functionalities. This analysis of the table of contents highlights the comprehensive nature of GANs, covering their definition, applications, components, training methodologies, loss functions, challenges, variations, implementation steps, and practical demonstrations.
- GANs have demonstrated remarkable capabilities in generating realistic data, enhancing image processing, and facilitating creative applications. Despite their effectiveness, challenges such as mode collapse and training instability persist, necessitating ongoing research efforts. Nevertheless, with proper understanding and implementation, GANs hold immense potential to revolutionize various domains, as exemplified by their practical utilization on datasets like MNIST.