Generative Adversarial Networks (GANs)

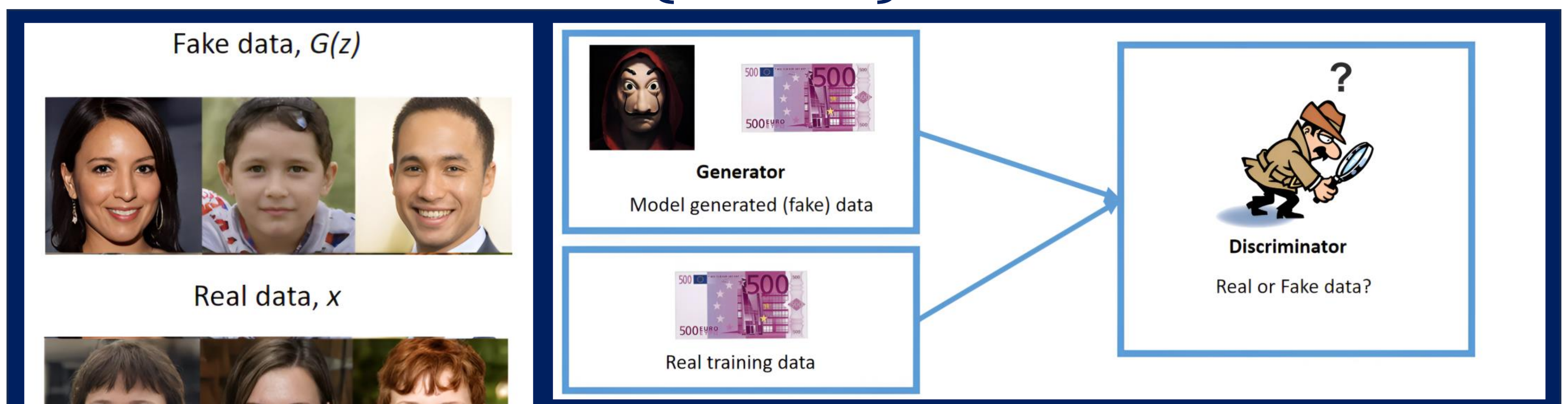
Rahul Madhav M
School of Computer Sciences, NISER Bhubaneswar

Introduction to GANs:

In Machine Learning, two primary model types exist:

- Discriminative models focus on learning the boundary between different classes in the data, estimating the conditional probability of the target variable given input features.
- Generative models aim to model the joint probability distribution of input features and the target variable, enabling them to generate new samples resembling the training data.

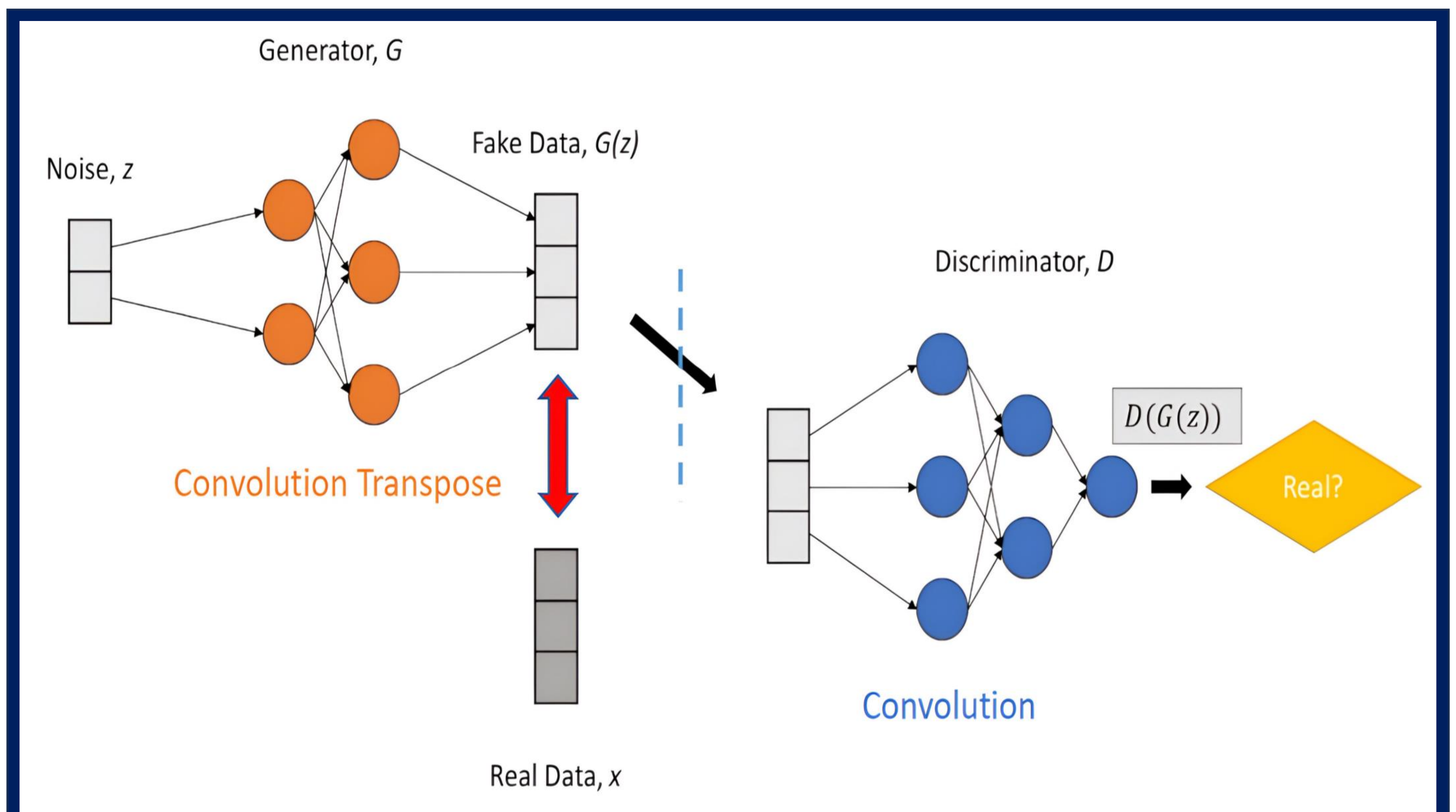
(GANs)



GANs: A revolutionary approach to generating lifelike synthetic data.

- GANs are a class of machine learning frameworks introduced by Ian Goodfellow and colleagues in June 2014.
- GANs utilize a competition between two neural networks: the generator and the discriminator, in a zero-sum game setting.
- The generator tries to produces data indistinguishable from real data, while the discriminator differentiates between real and fake data.
- Through adversarial training, both networks iteratively improve their performance, leading to the generation of high-quality synthetic data.

GAN Architecture



Noise (z) is an input into the Generator network (G) that outputs Fake data G(z) as input into the Discriminator network (D), then D outputs D(G(z)) that can be classified as Real = 1 or Fake = 0. Fine tune G and D weights during non-supervised training

Cost Function:

$$V(D,G) = \mathrm{E}_{x\sim p\mathrm{data}(x)}[\log D(x)] + \mathrm{E}_{z\sim pz(z)}[\log(1 - D(G(z)))]$$

Discriminator wants to maximize the cost function while Generator wants to minimize the cost function.

Training Algorithm

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{z_1, \ldots, z_m\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x_1, \ldots, x_m\}$ from data generating distribution $p_{data}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x_i) + \log(1 - D(z_i))]$$

end **for**

Sample minibatch of m noise samples $\{z_1, \ldots, z_m\}$ from noise prior $p_g(z)$.

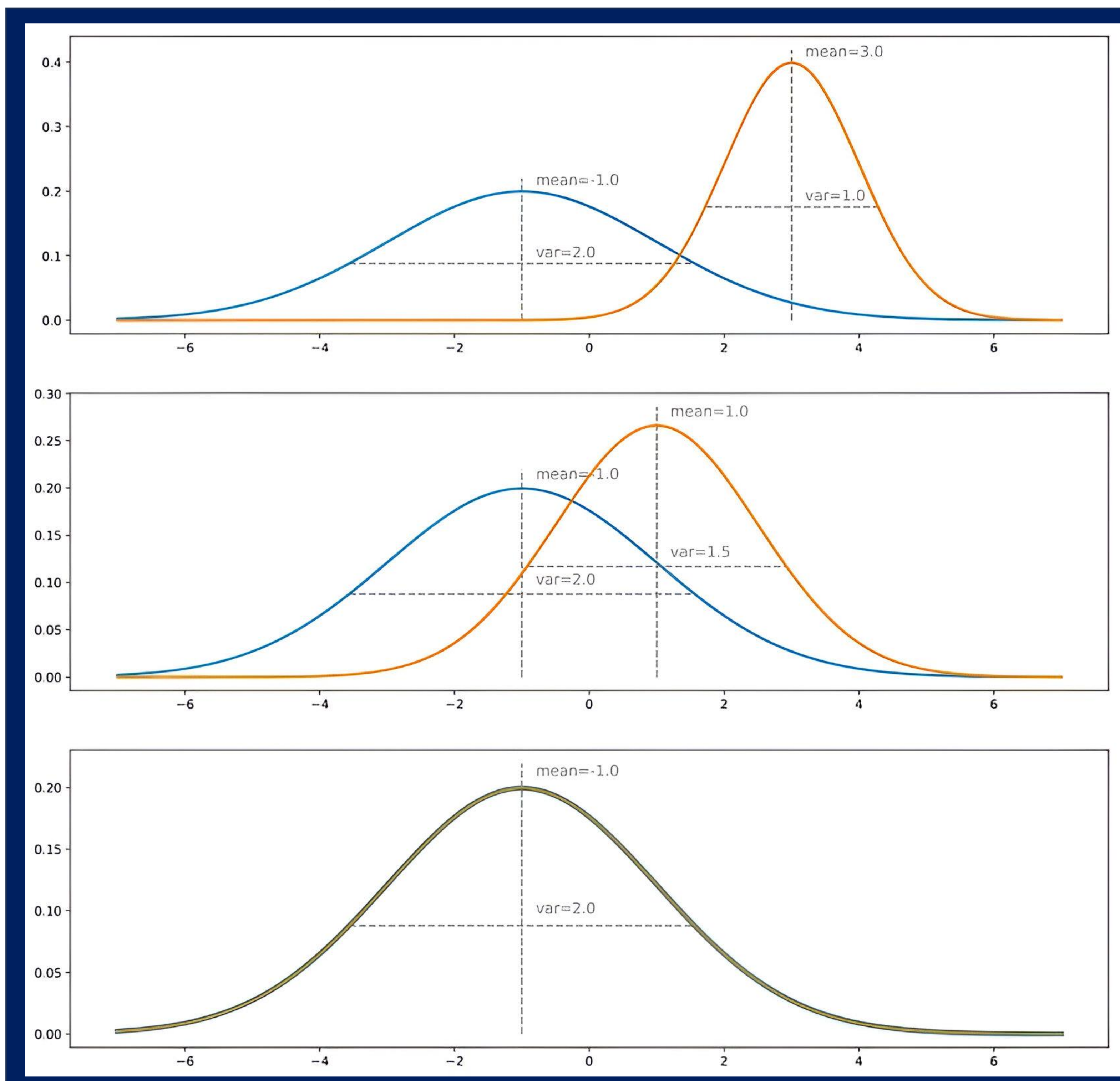
Update the generator by descending its stochastic gradient:

$$\nabla_{\theta g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(z_i))$$

end **for**

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Perfectly matched distributions



Ian Godfellow and his colleagues have proved that the minimum of the cost function is achieved if and only if the probability distribution of the fake samples matches the probability distribution of the real samples. This means that the fake samples become identical to the real samples, therefore we are no longer capable of distinguishing if a sample is real or fake

$$\min_G \max_D V(D,G) = -\log 4$$

Advantages and Disadvantages

Advantages	Disadvantages
• High-quality data generation. • Unsupervised learning capabilities. • Versatile applications in various domains	• Training instability. • Mode collapse. • Sensitivity to hyperparameters. • Evaluation challenges.