Generative Adversarial Nets

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NeurIPS, 2014.

2025년 7월 24일

이규원

Department of Computer Science and Engineering
Chung-Ang University

Index

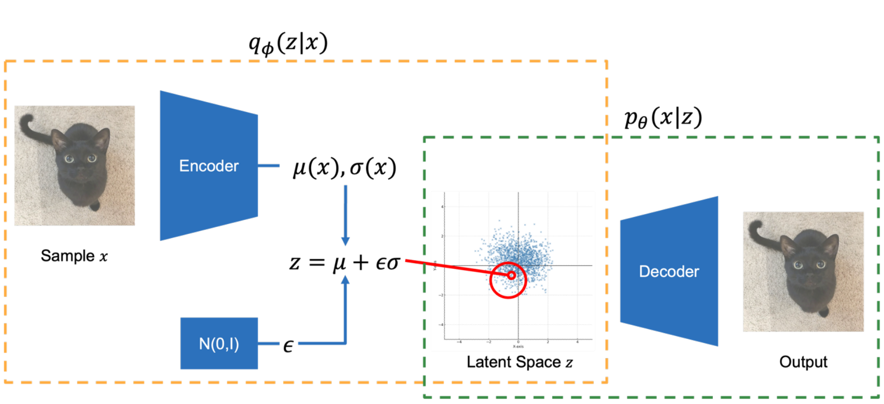
- Backgrounds
- Motivation
- Methods
- Experimental Results
- Conclusions

Backgrounds

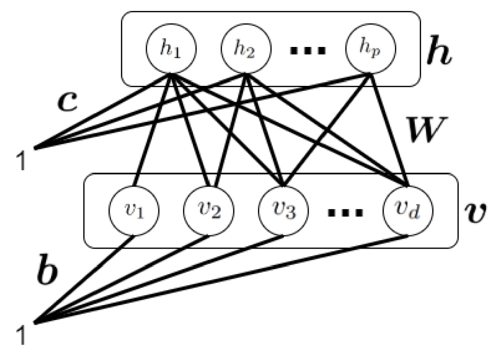
- **Generative models**
  - Models that learn the data generation process and generate new samples
- **Applications of generative models**
  - Creative work (e.g., art, music)
  - Data augmentation – when the dataset is small
  - Privacy protection – synthetic data without revealing sensitive attributes

Motivation

- **Limitations of previous works**
  - VAE
    - Requires an inference network
    - Drawbacks of Assuming Explicit Probability Distributions
  - Undirected models (e.g., RBM, DBM)
    - Require MCMC for sampling and gradient estimation
    - Partition function is intractable → training is slow and unstable



VAE

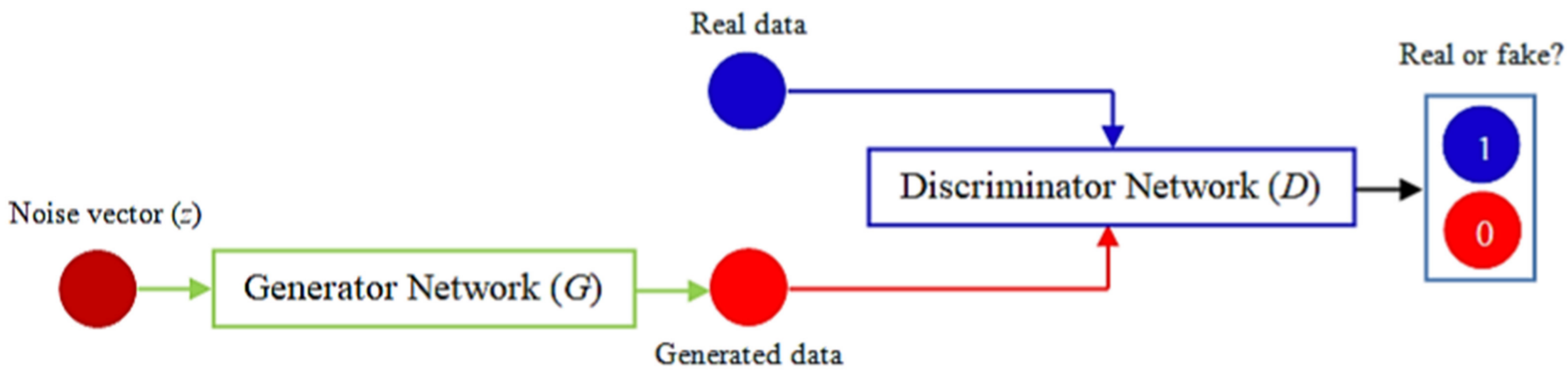


$$\mathbb{P}(\boldsymbol{h}|\boldsymbol{v}) = \frac{\mathbb{P}(\boldsymbol{h},\boldsymbol{v})}{\mathbb{P}(\boldsymbol{v})} = \frac{\mathbb{P}(\boldsymbol{v},\boldsymbol{h})}{\sum_{\boldsymbol{h}\in\mathbb{R}^p}\mathbb{P}(\boldsymbol{v},\boldsymbol{h})}$$

RBM

Motivation

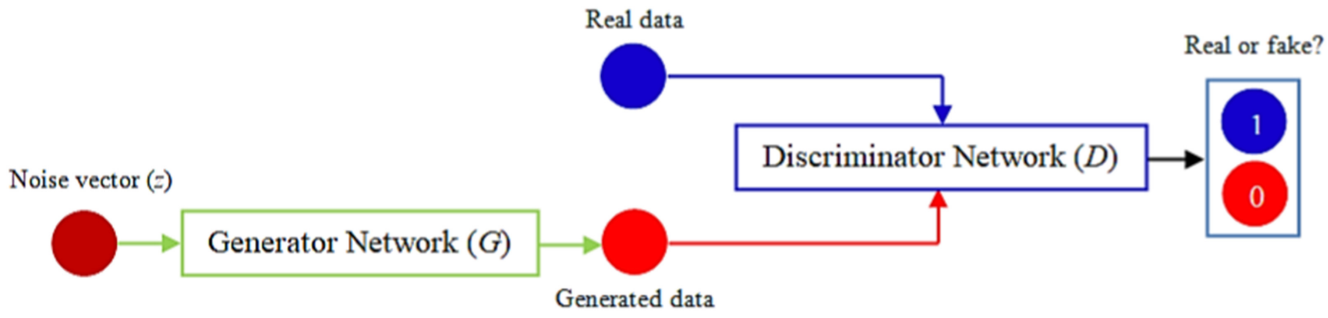
- **Goal**
  - Train generative models without explicit probabilistic modeling
- **Approach**
  - Adversarial training as an optimization framework



Jabbar, Abdul, Xi Li, and Bourahla Omar. "A survey on generative adversarial networks: Variants, applications, and training." *ACM Computing Surveys (CSUR)* 54.8 (2021): 1-49.

Methods – Overview

- **GAN architecture**
  - Generator $G(z)$
    - Input: Random noise $z \sim p(z)$ (e.g., Gaussian)
    - Output: Fake sample $x = G(z)$ shaped like real data
    - Goal: Fool the discriminator / Generate realistic samples indistinguishable from real data
  - Discriminator $D(x)$
    - Input: Real or fake sample x
    - Output: Probability $D(x) \in [0, 1]$, likelihood of being real
    - Goal: Distinguish real data from generated samples



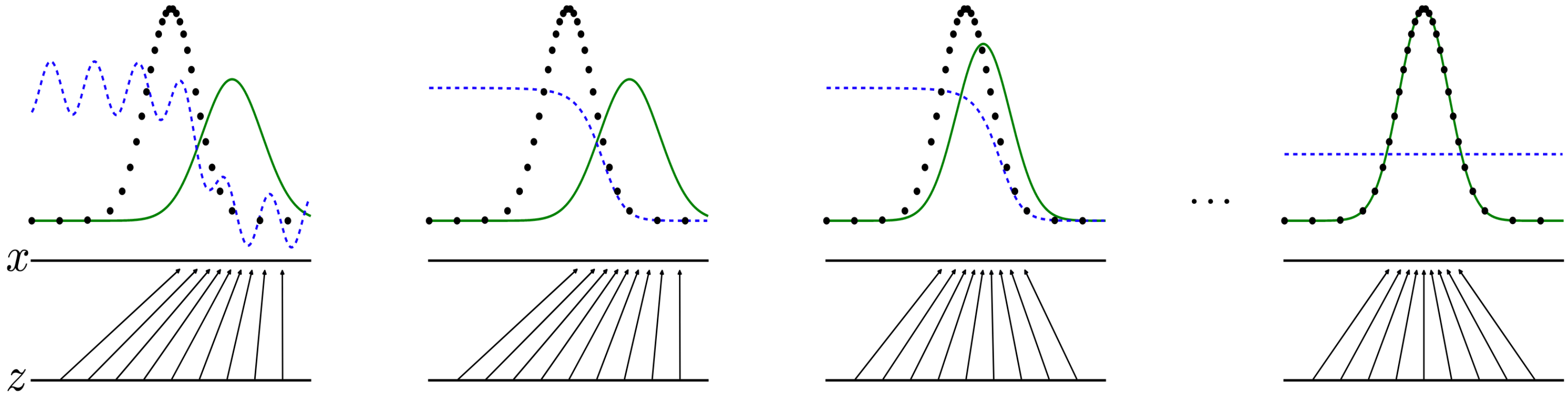
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Jabbar, Abdul, Xi Li, and Bourahla Omar. "A survey on generative adversarial networks: Variants, applications, and training." *ACM Computing Surveys (CSUR)* 54.8 (2021): 1-49.

Methods

- **Adversarial training loop**
  - Temporarily hold G fixed, update D for k steps
  - Then hold D fixed, update G to fool the updated D

Illustrative plot for intuition only
Blue: Discriminative Distribution
Black: Ground Truth Distribution
Green: Generator Distribution



(a) (b) (c) (d)

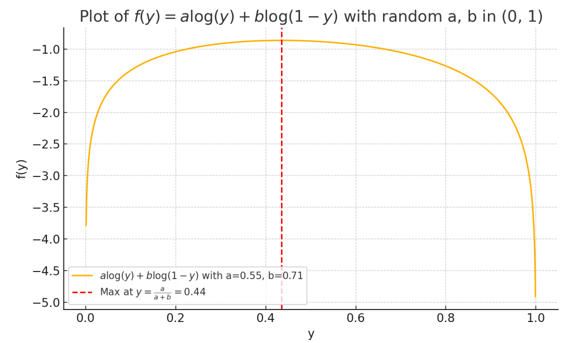
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Methods

- **Global optimality $p_g = p_x$**
  - The optimal discriminator (G fixed)

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



  - The global minimum

$$C(G) = \max_D V(G, D) = \mathbb{E}_{x\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}\right] + \mathbb{E}_{x\sim p_g}\left[\log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)}\right]$$

$$\mathbb{E}_{x\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{2m(x)}\right] + \mathbb{E}_{x\sim p_g}\left[\log \frac{p_g(x)}{2m(x)}\right] \qquad m(x) = \frac{1}{2}(p_{\text{data}}(x) + p_g(x))$$

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right)$$

  - The value function C(G) is minimized $p_g = p_x$
    - The **Jensen-Shannon divergence vanishes** and the GAN reaches its global optimum

Methods

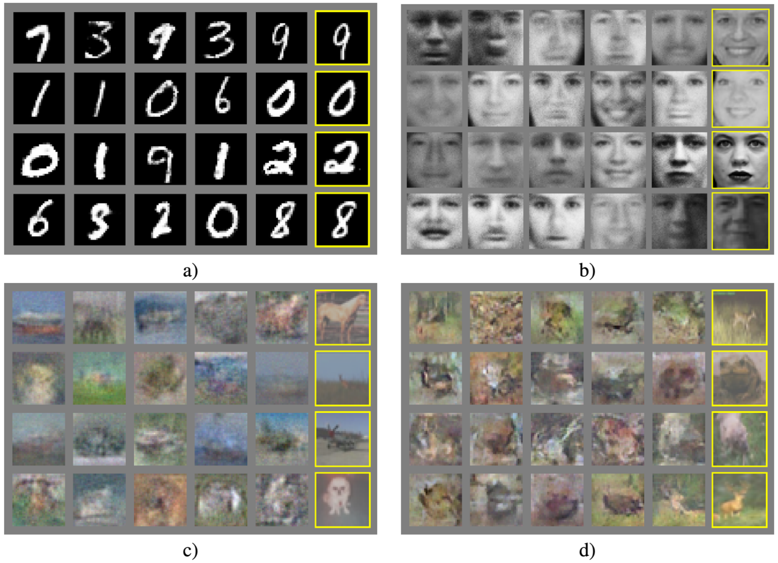
- **Theoretical convergence of GAN**
  - Under ideal conditions (infinite model capacity for G and D, optimal D, direct $\boldsymbol{p_g}$ manipulation)
- **Practical limitations**
  - Limitation model capacity
  - Indirect $\boldsymbol{p_g}$ optimization: Optimize generator parameters θ, leading to a non-convex optimization
  - Empirical Success: Despite theoritical gaps, GANs with MLPs perform well

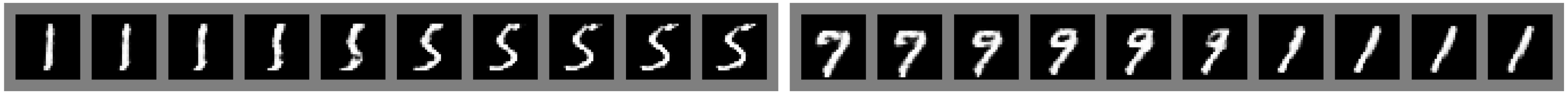
Experimental Results

Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [6]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

- **Parzen window-based log-likelihood**
  - Approximates the data likelihood by placing Gaussian kernels around generated samples
  - Estimates test log-likelihood via a smoothed density over samples
  - Limitation: Inaccurate in high dimensions and sensitive to kernel bandwidth (σ)
- **High GAN performance on MNIST**
  - Simple data distribution: Low-dimensional, structured, and class-separable
  - Sharp sample quality: Adversarial loss leads to clean, high-quality digits
  - Low mode complexity: Limited modes (digits 0–9), so mode collapse is less harmful

Experimental Results





- Rightmost column: Nearest neighbor from the training dataset
- The interpolation results (Fig. 3) imply that the generator has learned a smooth and structured latent space, where linear transitions in input space correspond to semantic transitions in output space

Conclusions

- Validates adversarial training as a viable approach to generative modeling
- Provides theoretical justification for the minimax game formulation, showing convergence to the data distribution under ideal conditions