Generative Adversarial Nets

SKT Fellowship

LEE JINKYU

Department of Civil, Environmental and Architectural Engineering, Korea University

February 5, 2023

SKT AI Fellowship

Introduction

About GAN

1. Before GAN

2. What is GAN

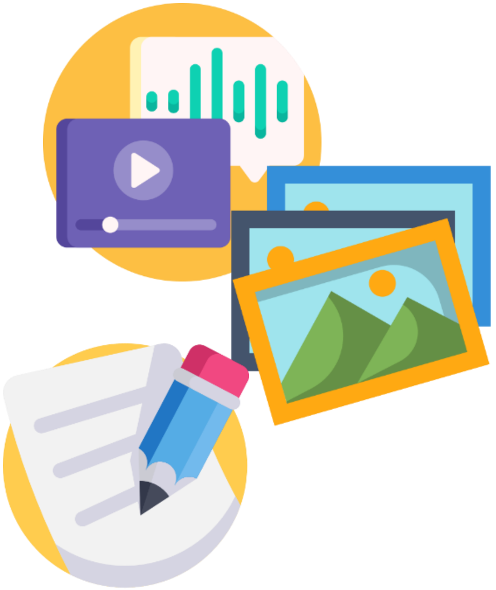
3. Structure of GAN

- Value function V(G,D)
- Update Algorithm
- Derivation of Algorithm

4. Advantages of GAN

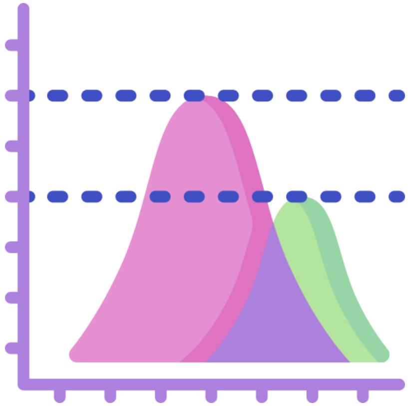
Before GAN

Deep learning before GAN

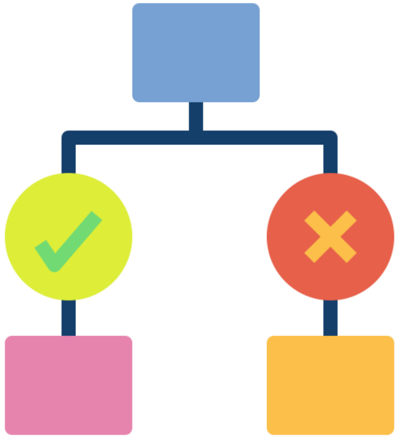


다양한 데이터
(이미지, 오디오, 자연어)

순전파, 역전파



데이터 확률 분포 해석 및 예측



데이터 해석 및 분류

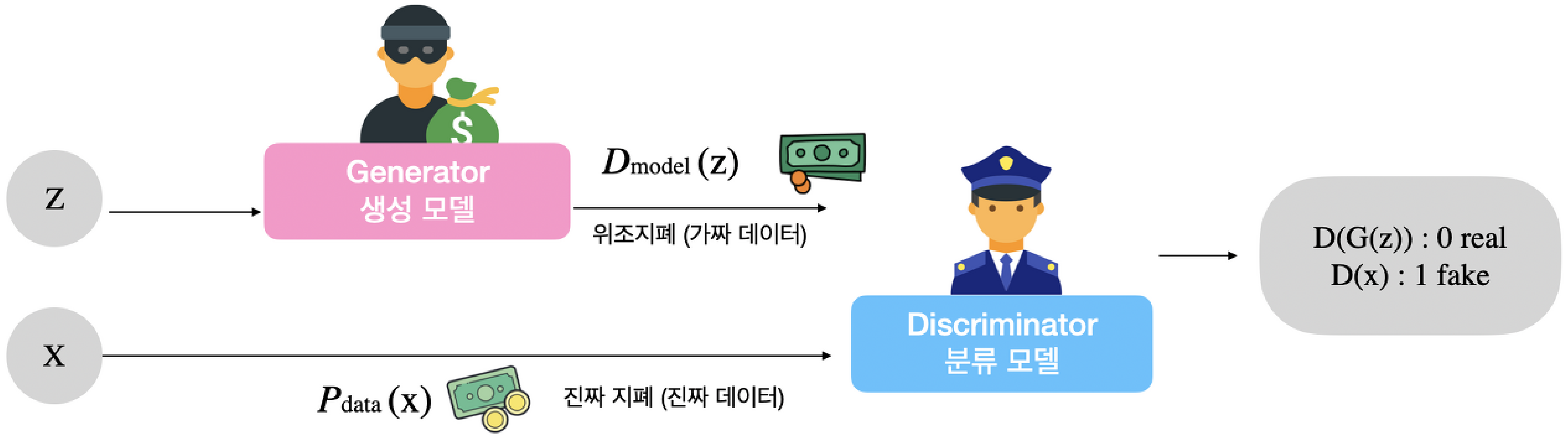
Deep Generative model은 확률적 계산의 어려움 + gradient를 이용한 역전파를 이용하기 어렵다는 문제로 인해 발전이 더뎠음

→ 이러한 문제를 해결할 수 있는 새로운 개념의 적대적 신경망을 제안

What is GAN

Mechanism of GAN

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



Generator : noise Z를 통해 이미지를 만들어 냄. 원본 데이터 X와 최대한 유사하게 만들어 Discriminator를 속이는 것을 목표로 함

Discriminator : Classifier, 이미지가 원본인지 Generator를 통해 만들어진 것인지를 판별

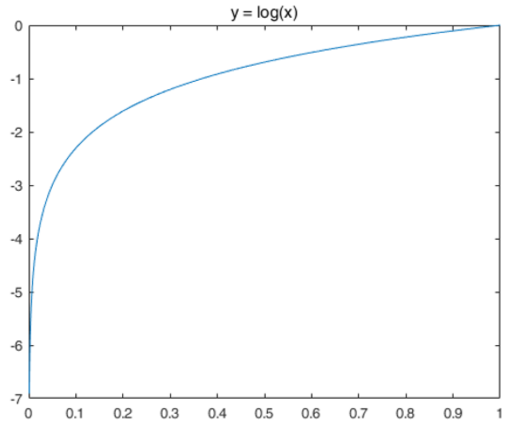
→ Min Max two player game이라고 불리는 이유 (loss function이 각 모델의 목표에 맞게 설정 되어있으며 계속 경쟁한다)
→ 최종적으로 Discriminator가 이미지의 진위 여부를 알 수 없을 때까지 학습한다. (그만큼 Generator의 성능이 좋아졌다는 것을 의미)

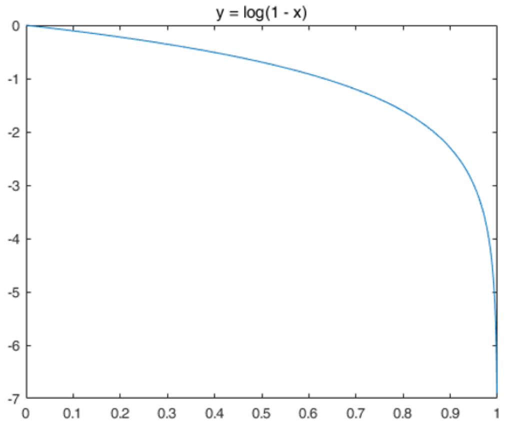
Structure of GAN – Value function V(G,D)

Min Max value function

$$\min_{G}\max_{D} V(G,D) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

기호	의미
x	Real Data
z	noise, 확률분포로부터 추출한 샘플 z, 가짜 이미지를 생성할 '재료'
p_g	x에 대한 G의 분포
$G(z)$	G가 Noise z를 받아서 생성한 Fake Data. (Real data와 사이즈가 같아야함)
θ_g	multilayer perceptrions의 parameters
$G(z;\theta_g)$	data space에 대한 mapping
$D(x)$	x가 p_g가 아니라 원본 데이터에서 나왔을 확률





G : log(1-D(G(z)))에만 관여한다. D(G(z)) = 1을 만들어 V(G,D)를 최소화하는 방향으로 학습
→ 이 때, log(1-D(G(z)))를 minimize 시키는 것이 아닌 log(D(G(z))) Maximize 시킨다. (gradient saturate 문제, 그래프 참고)

D : Function 전체에 관여한다. 진위 여부를 판단해 V(G,D)를 최대화하는 방향으로 학습

Structure of GAN – Update Algorithm

Update Generate & Discriminate model

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
  - Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right].$$

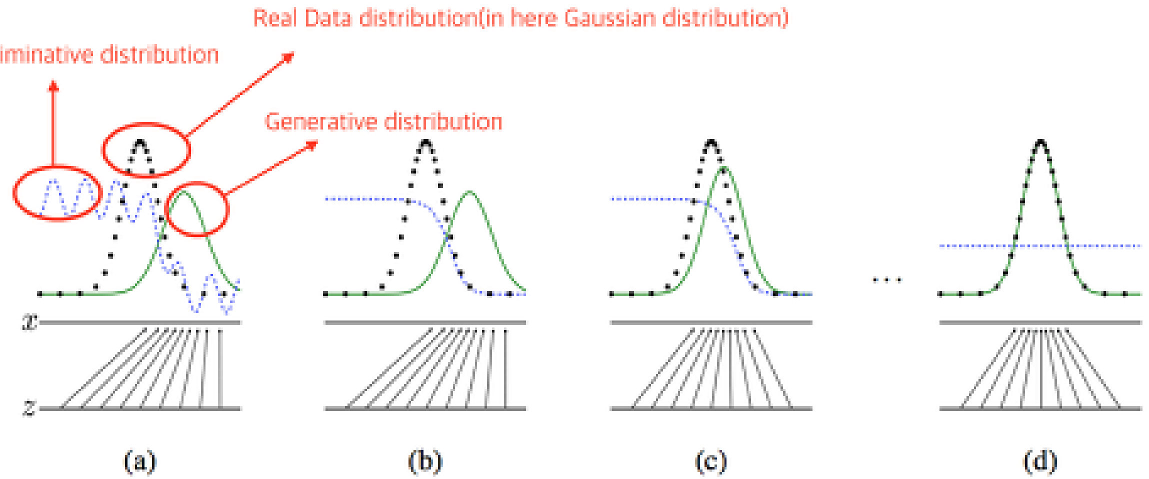
  end for

  - Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$

end for

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



1 epoch을 돌릴 때, D를 k번 학습 한 후 G를 한번 학습한다 → Computation Cost의 부담 때문

Structure of GAN – Derivation of Algorithm

Optimize value function

Value Function이 P_g = P_data로 수렴하게 만들어줄 수 있는가

① Global optimality of $P_g = P_{data}$

↳ 생성된 분포와 실제분포가 일치하는 global optima

- G는 고정, 최적의 Discriminator D ?

loss: $E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))]$

<Proof 1>

: G가 고정일 때, D는 V(G,D)를 최대화하고 싶어함

$$\min_G \max_D V(G,D) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))]$$

= (if G is fixed)

$$D^*(x) = \arg\max_D V(D) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))]$$

P_z에서 샘플링한 z를 G(z)에 넣으면 G(z)는 P_g를 따르게 된다. ⇒ X를 P_g에서 샘플링한 꼴.

$$= E_{x\sim P_{data}(x)}[\log D(x)] + E_{x\sim P_g(x)}[\log(1-D(x))]$$

↳ 실제 분포에서 x 샘플링 ↳ 생성 이미지에서 X 샘플링

☆ Expectation 정의

$E_{x\sim p(x)}[f(x)] = \int_x p(x)f(x)dx$

$$= \int_x P_{data}(x)\log D(x)dx + \int_x P_g(x)\log(1-D(x))dx$$

$$= \int_x P_{data}\log D(x)dx + P_g(x)\log(1-D(x))dx$$

$= (P_{data} = a,\ D(x) = y,\ P_g(x) = b)$

$$\int_x a\log y + b\log(1-y)\,dx$$

미분

$$\frac{a}{y} - \frac{b}{1-y} = \frac{a-(a+b)y}{y(1-y)}$$

→ $y = \frac{a}{a+b}$ 에서 최대

∴ 고정 G에서 optimal $D = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$

<proof 2>

optimal D를 이용해 V(G,D) 재정의

$$\to \min_G \max_D V(G,D) = \min_G V(G^*,D)$$

$$V(D^*,G) = E_{x\sim P_{data}(x)}[\log D^*(x)] + E_{x\sim P_g}[\log(1-D^*(x))]$$

$$= \int_x P_{data}(x)\log\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}dx + \int_x P_g(x)\log\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}dx$$

$$= -\log 4 + \log 4 + \int_x P_{data}(x)\log\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}dx + \int_x P_g(x)\log\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}dx$$

$$= -\log 4 + \int_x P_{data}(x)\log\frac{2P_{data}(x)}{P_{data}(x)+P_g(x)}dx + \int_x P_g(x)\log\frac{2P_{data}(x)}{P_{data}(x)+P_g(x)}dx$$

☆ KL Divergence

: 두 함수의 확률분포가 얼마나 다른지 측정.

$D_{KL}(B\|A)$

$= E_{x\sim B}\left[\log\frac{B(x)}{A(x)}\right]$

$= \sum_x B(x)\log\frac{B(x)}{A(x)}$

$$= -\log 4 + KL\left(P_{data}\,\middle\|\,\frac{P_{data}+P_g}{2}\right) + KL\left(P_g\,\middle\|\,\frac{P_{data}+P_g}{2}\right)$$

Jensen-Shannon divergence (JSD)

$$= -\log 4 + 2\cdot JSD(P_{data}\|P_g)$$

↳ JSD는 $P_{data} = P_g$ 일때만 0 나머지는 항상 양수값이기에!

∴ V(G,D)의 global minimum은

$\boxed{P_{data} = P_g}$

Value Function의 global optima가 p_g = p_data이다

Advantages of GAN

Pros & cons

	Deep directed graphical models	Deep undirected graphical models	Generative autoencoders	Adversarial models
Training	Inference needed during training.	Inference needed during training. MCMC needed to approximate partition function gradient.	Enforced tradeoff between mixing and power of reconstruction generation	Synchronizing the discriminator with the generator. Helvetica.
Inference	Learned approximate inference	Variational inference	MCMC-based inference	Learned approximate inference
Sampling	No difficulties	Requires Markov chain	Requires Markov chain	No difficulties
Evaluating $p(x)$	Intractable, may be approximated with AIS	Intractable, may be approximated with AIS	Not explicitly represented, may be approximated with Parzen density estimation	Not explicitly represented, may be approximated with Parzen density estimation
Model design	Nearly all models incur extreme difficulty	Careful design needed to ensure multiple properties	Any differentiable function is theoretically permitted	Any differentiable function is theoretically permitted

Table 2: Challenges in generative modeling: a summary of the difficulties encountered by different approaches to deep generative modeling for each of the major operations involving a model.

다른 Deep Generate model들에 비해 computation 적으로 효율적이며 (markov cahin 등을 사용하지 않음)
Deep learning의 최대 장점 중 하나인 gradient를 이용한 학습이 가능하다