Generative Adversarial Nets

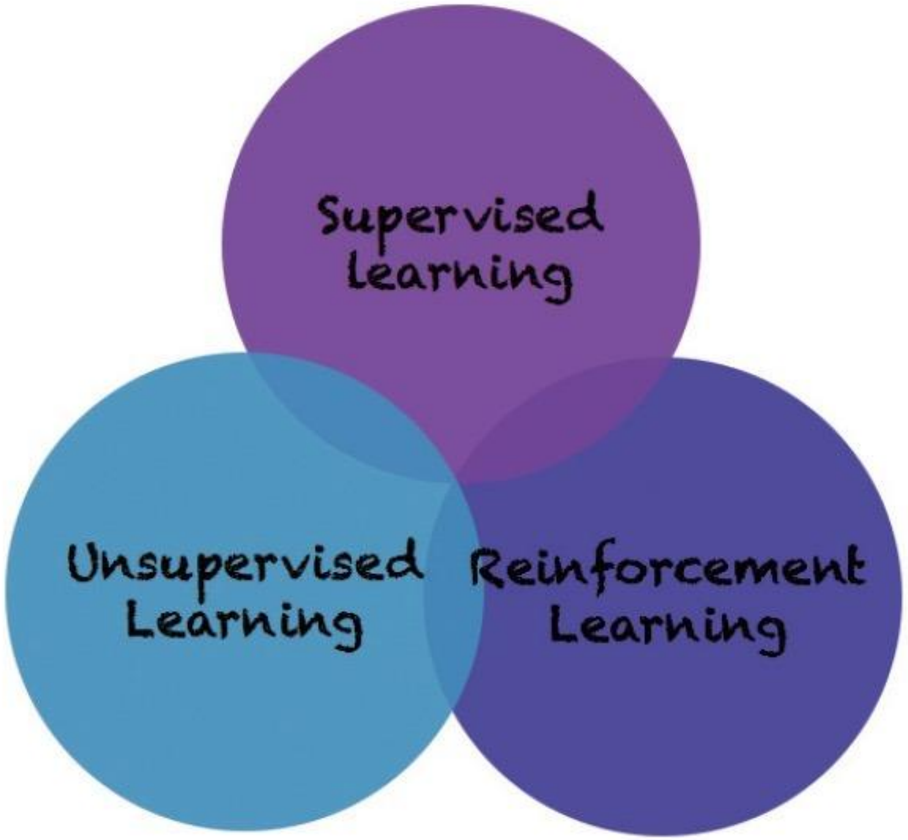
Samsung Software Developer Community
Korea Vision & Robotics
HoChan Jeong
2023.06.17

Contents

01. Background

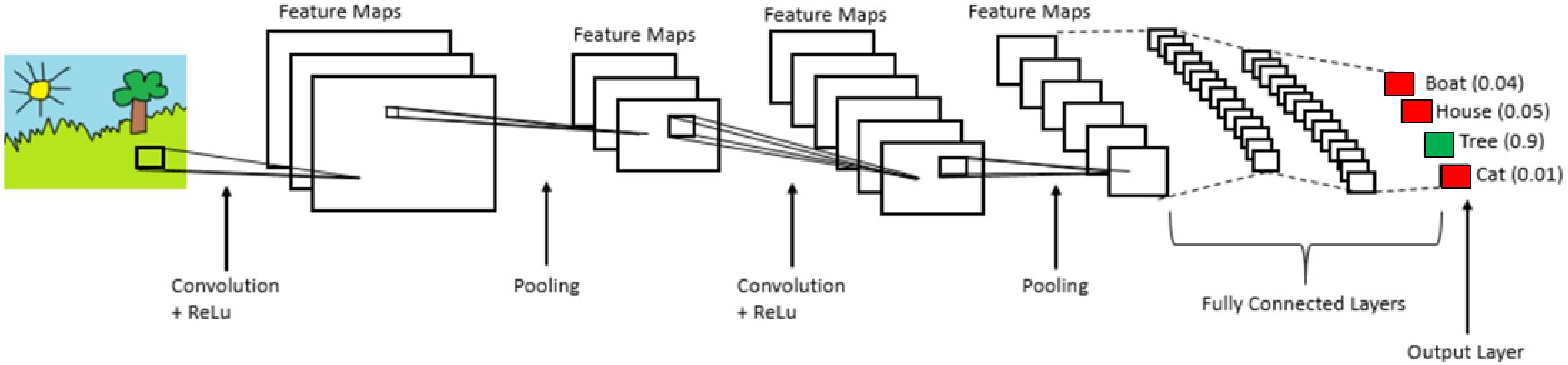
Background

In Machine Learning…



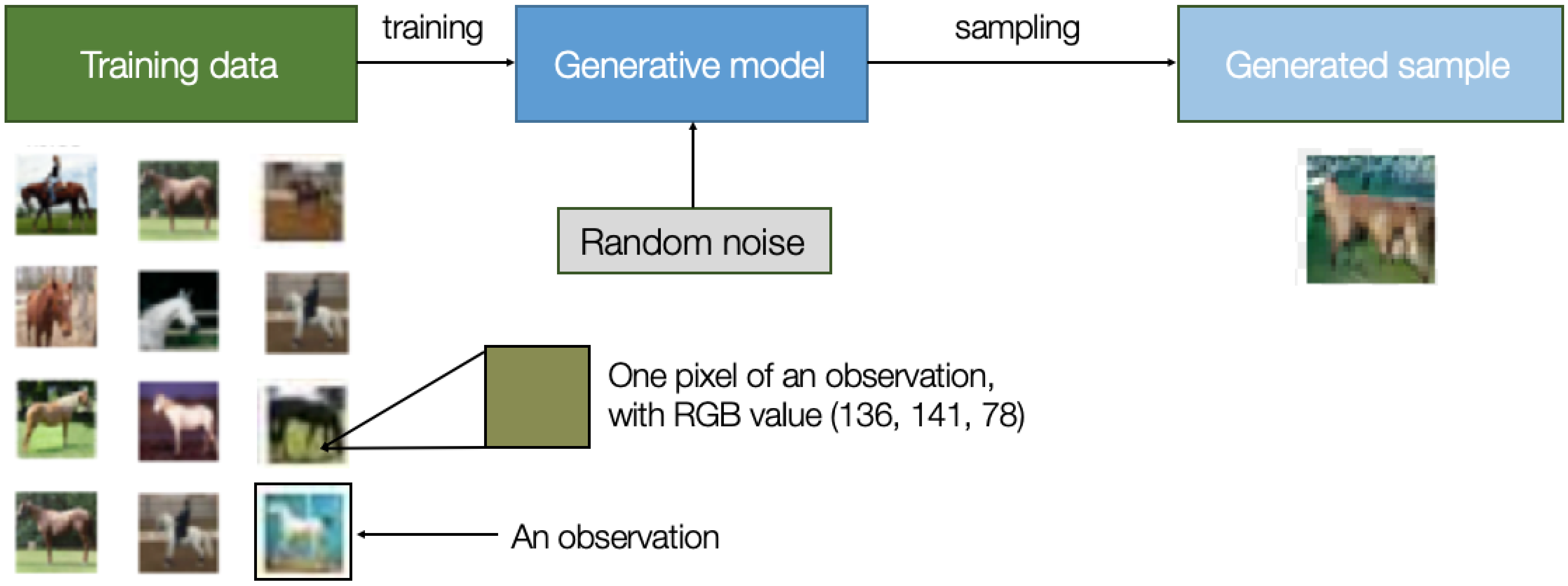
Background

대표적인 Image Classification, Object Detection …

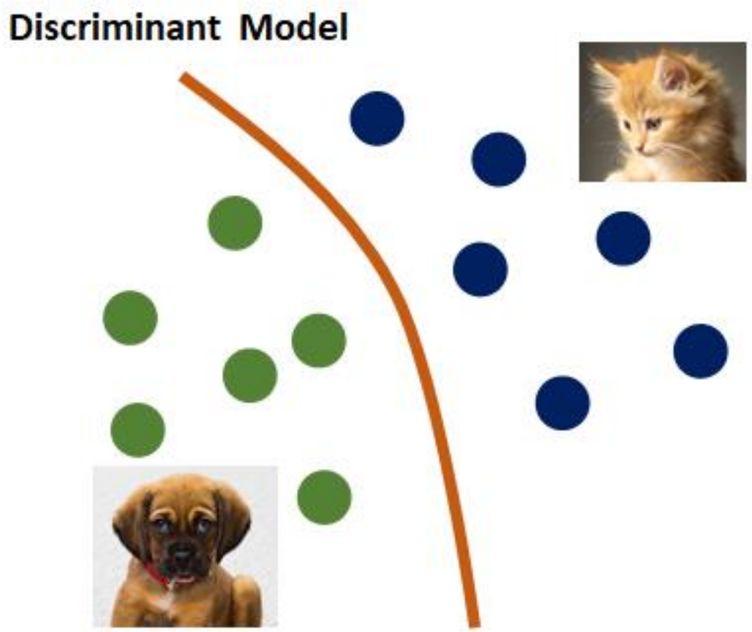


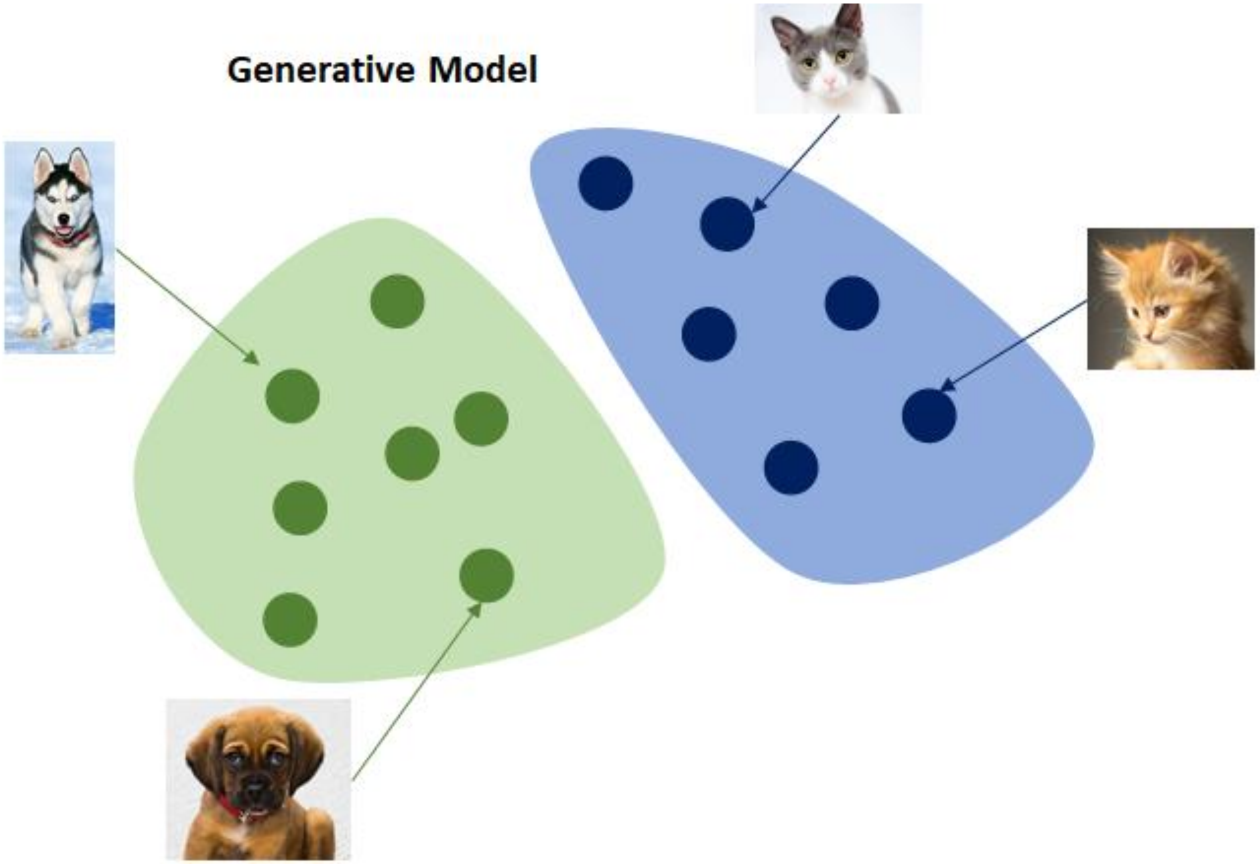
Background

우리가 배울 Generative Model



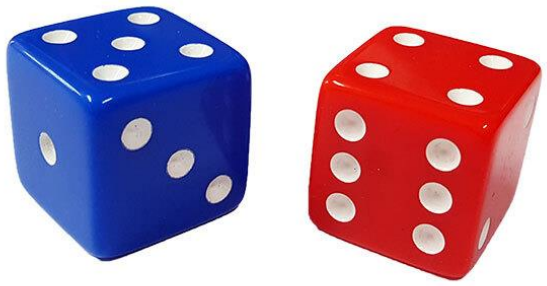
Background

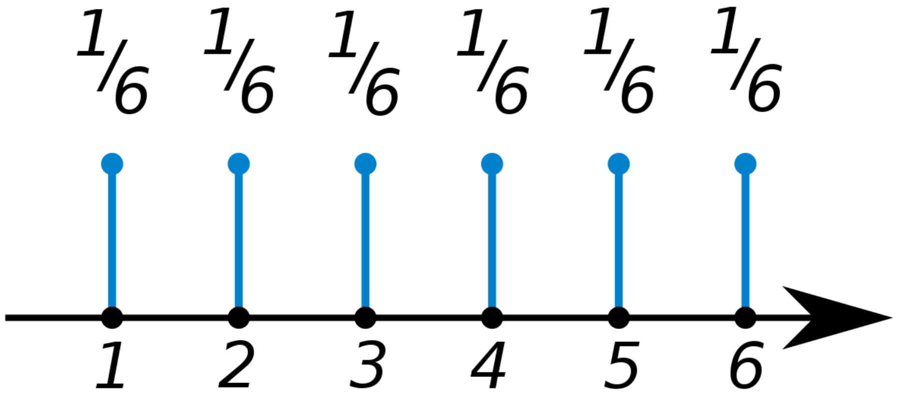


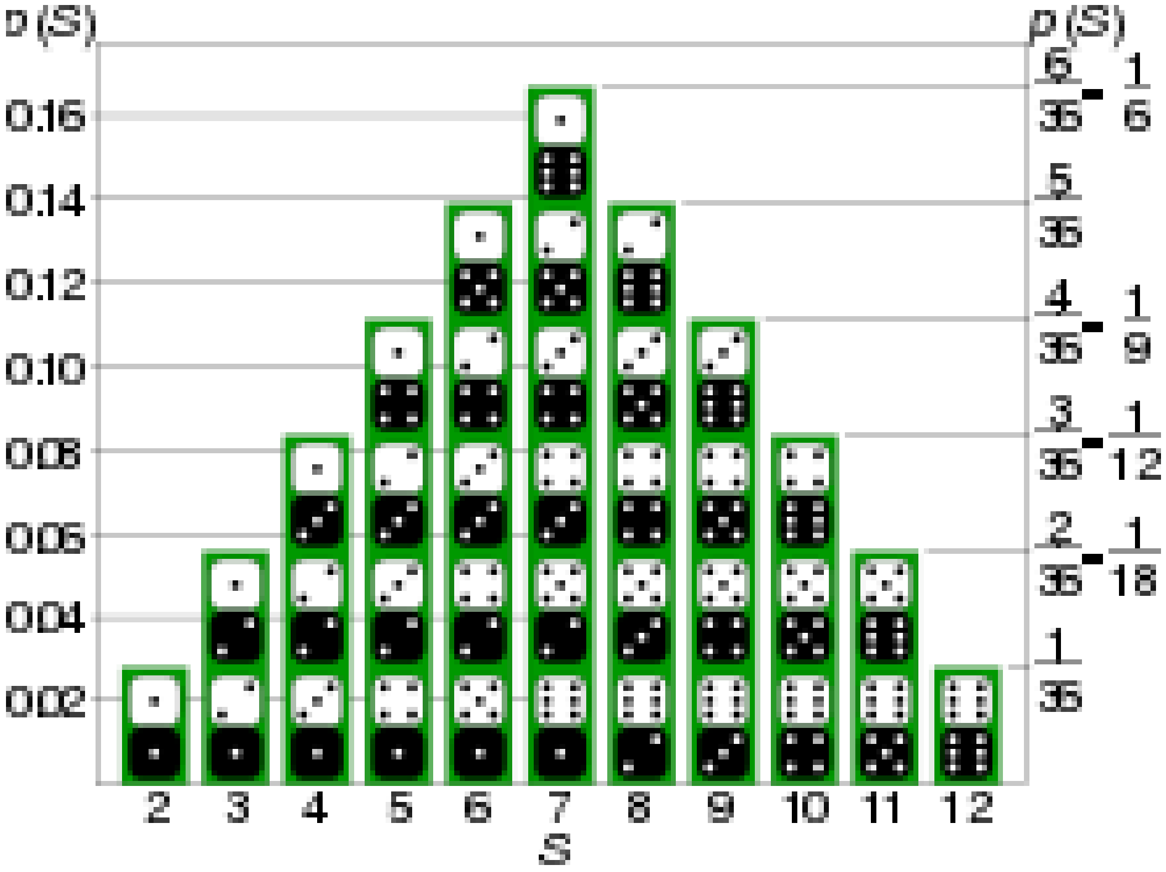


Background

확률분포?

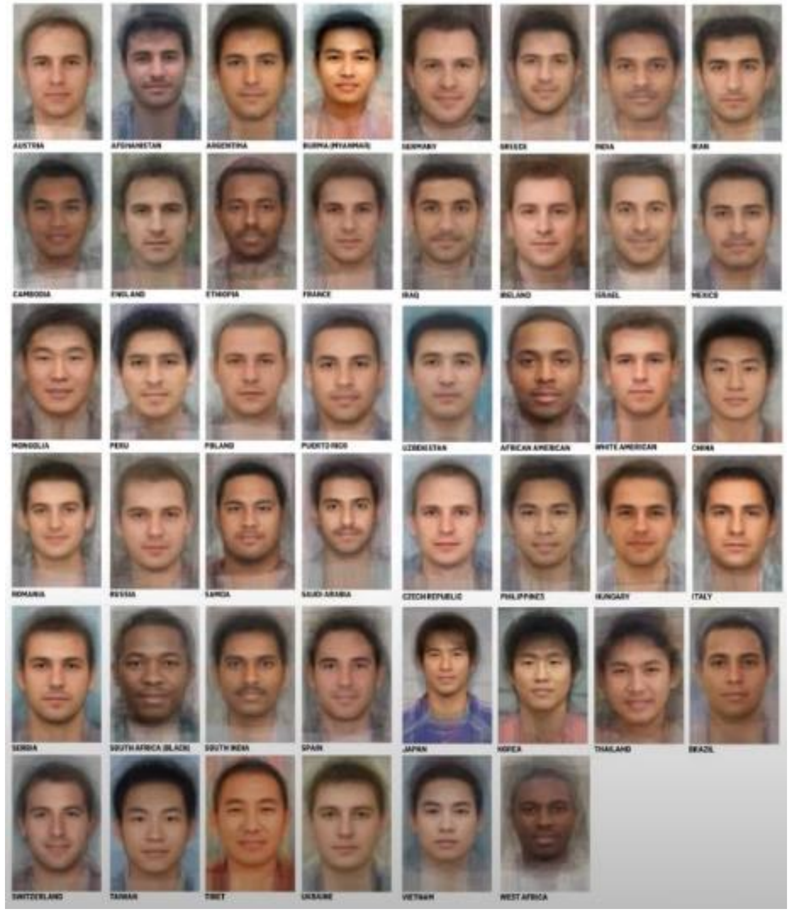


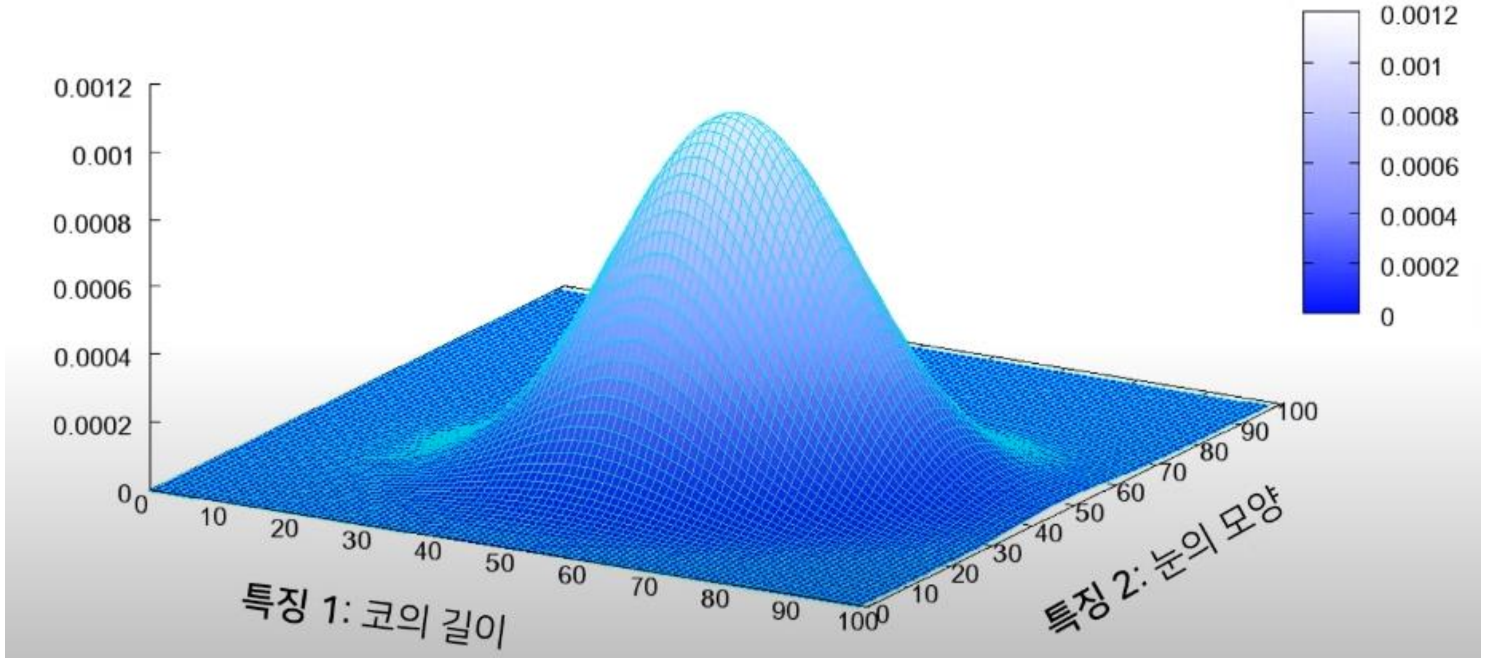




Background

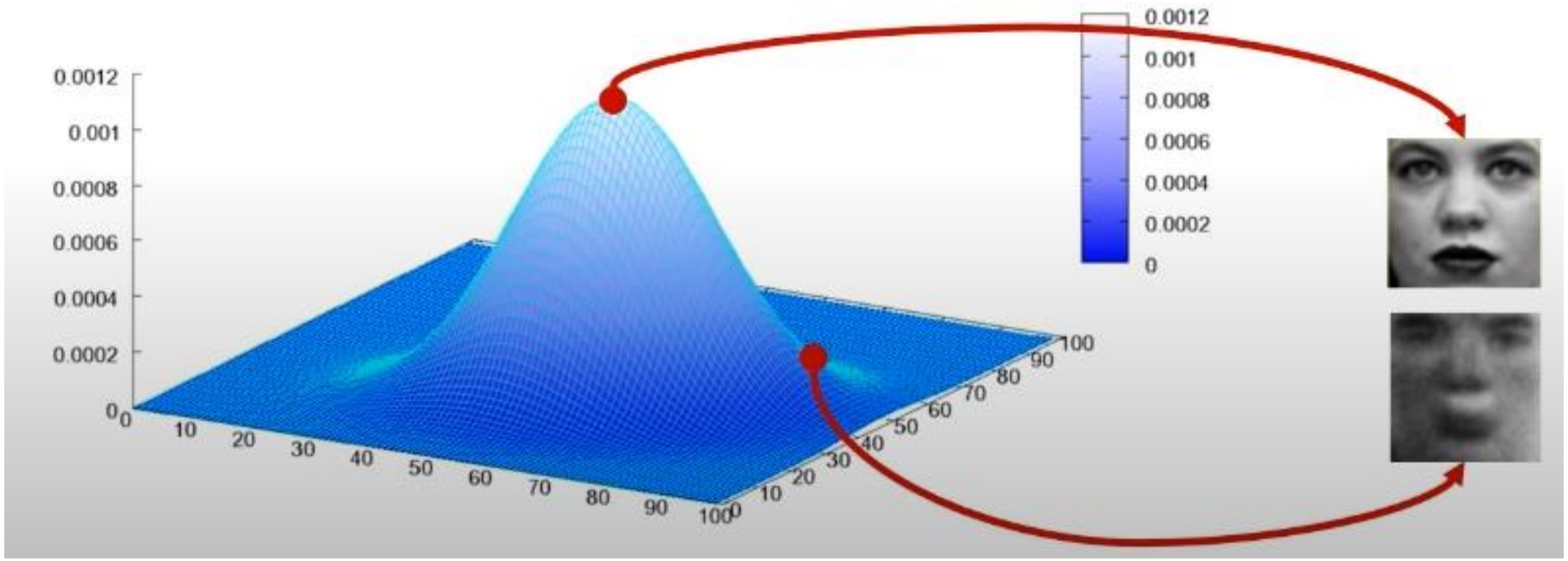
In images…





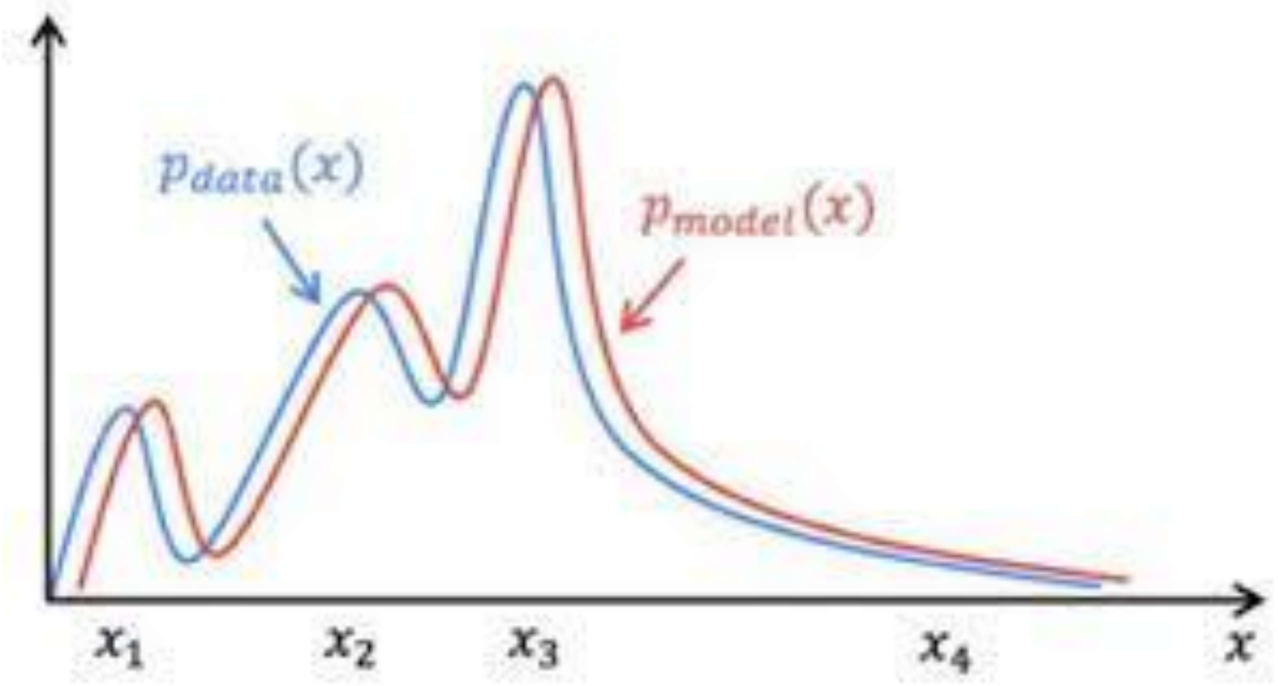
Background

In images…



Background

Generative model 목표 = 데이터의 분포에 근사시키는 것



Background

이를 학습시키는 방법 – GAN 이전에는..

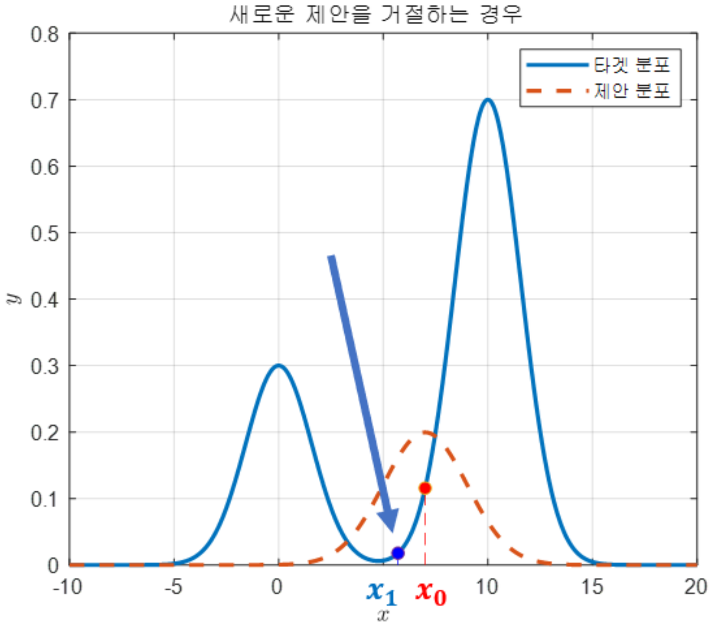
Markov Chains Monte Carlo (MCMC)

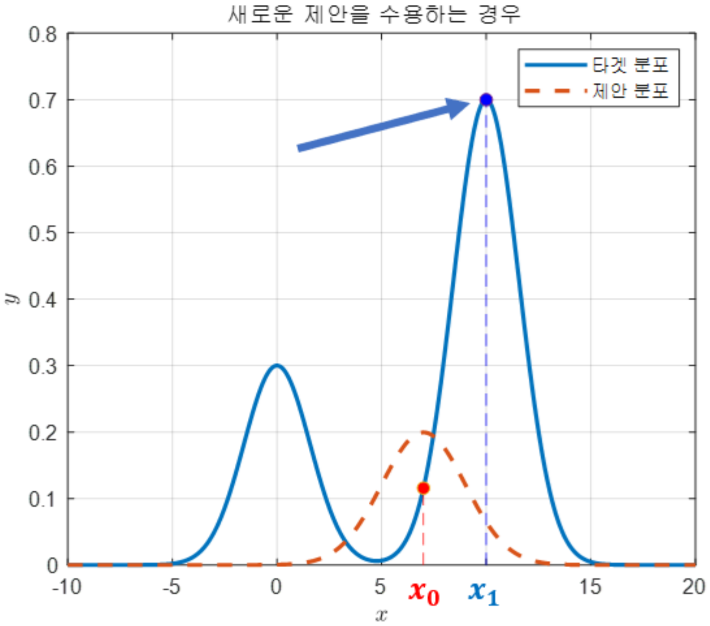
https://angeloyeo.github.io/2020/09/17/MCMC.html

Background

Monte Carlo + Markov Chain을 합친 개념 = "통계적인 특성을 이용해 무수히 뭔가를 많이 시도해본다

"가장 마지막에 뽑힌 샘플이 다음 번 샘플을 추천해준다"





Background

- Metropolis : Metropolis는 symmetric 한 확률분포를 사용하는 경우에 대한 알고리즘을 제안
  - 수식
    - $f(x)$: 타겟분포
    - $g(x)$: 제안분포
    - $\frac{f(x_1)}{f(x_0)} > 1$인 경우 accept
    - 위를 만족하지 못할 경우, $\frac{f(x_1)}{f(x_0)} > u$ 인 경우 accept
    - u 값은 uniform 분포 $U_{(0,1)}$ 에서 임의로 추출한 값
    - 제안 분포 $g(x)$ 의 역할은? : 제안 분포 내에서 다음 포인트(x_1)를 추천 받음

https://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=banana

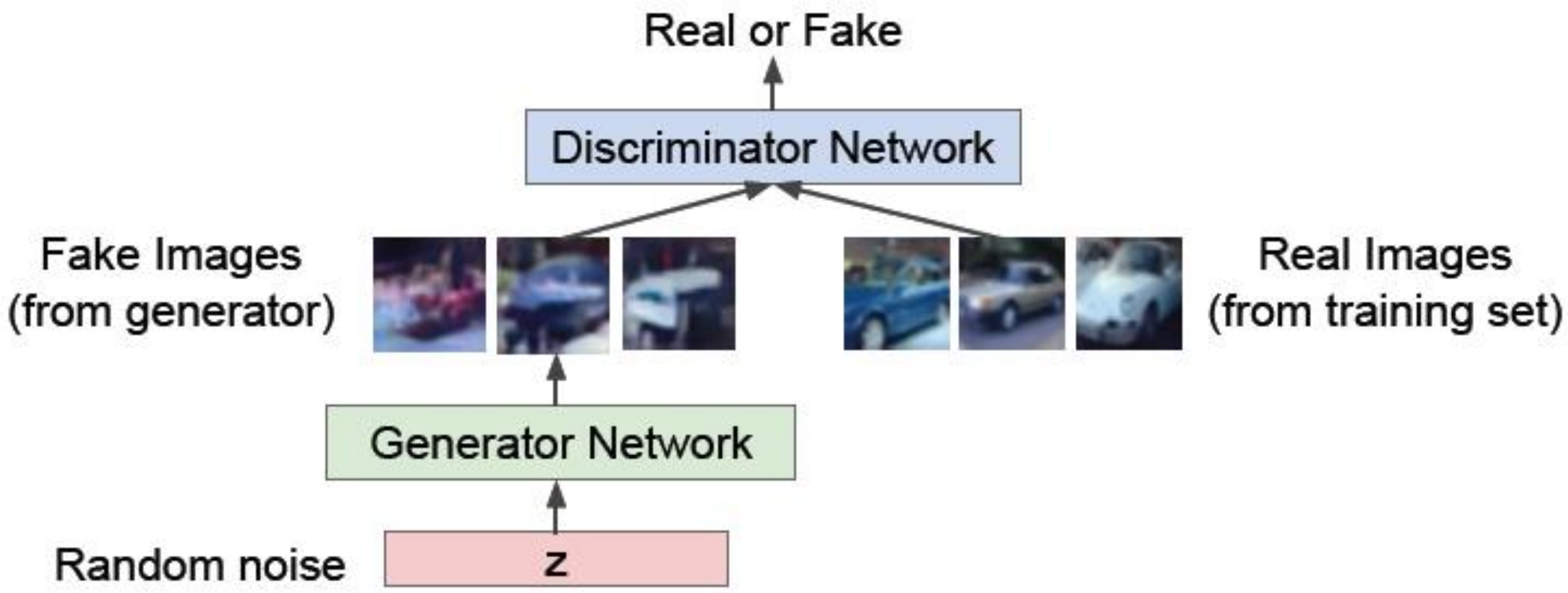
Background

MCMC 단점

- inference를 위한 별도의 가정 등이 필요 (구현이 복잡)
- 무수히 많은 시도가 있어야 함 (시간 소요가 많음)
- High dimensional vector space에서 잘 되기 어렵다 (앞에서 시각화로 확인)
- 그리고 Backpropagation, Batch normalization 등의 방법을 활용하기 힘들다

Background

GAN 등장



Background

GAN 장점

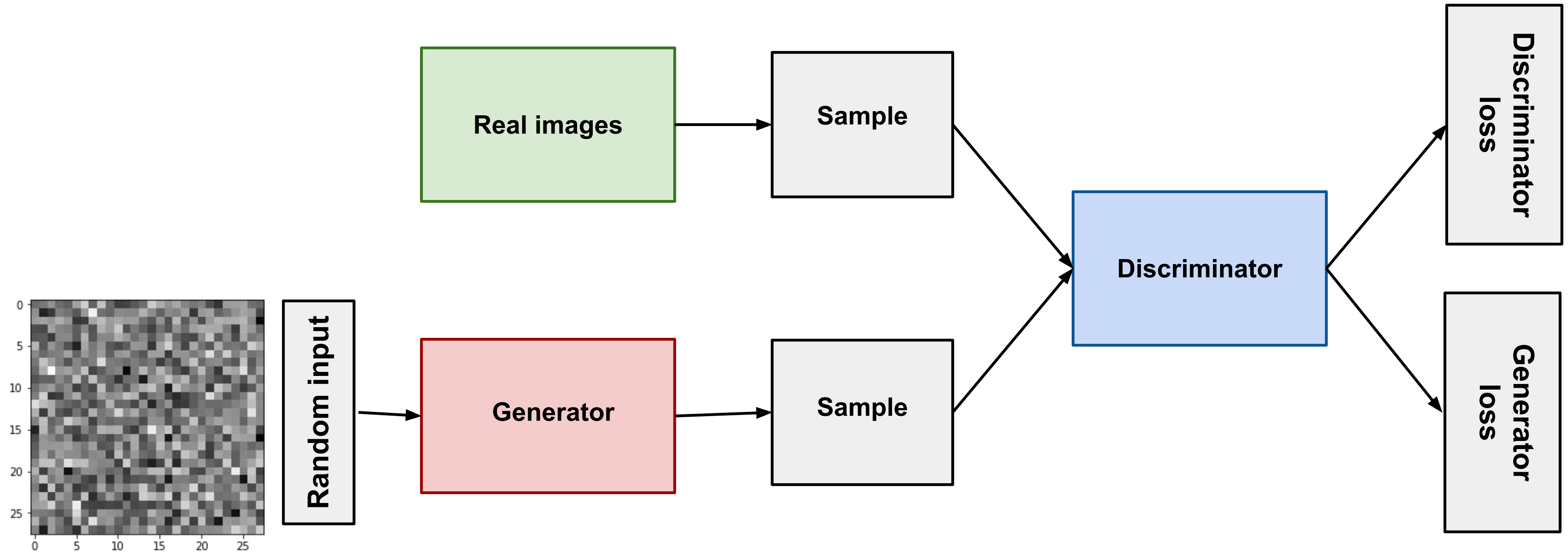
- inference를 위한 별도의 가정 등이 필요 x
- 무수히 많은 시도 x
- High dimensional vector space 작동가능
- Backpropagation, Batch normalization 등 활용가능

Finally, we would like to thank Les Trois Brasseurs for stimulating our creativity.

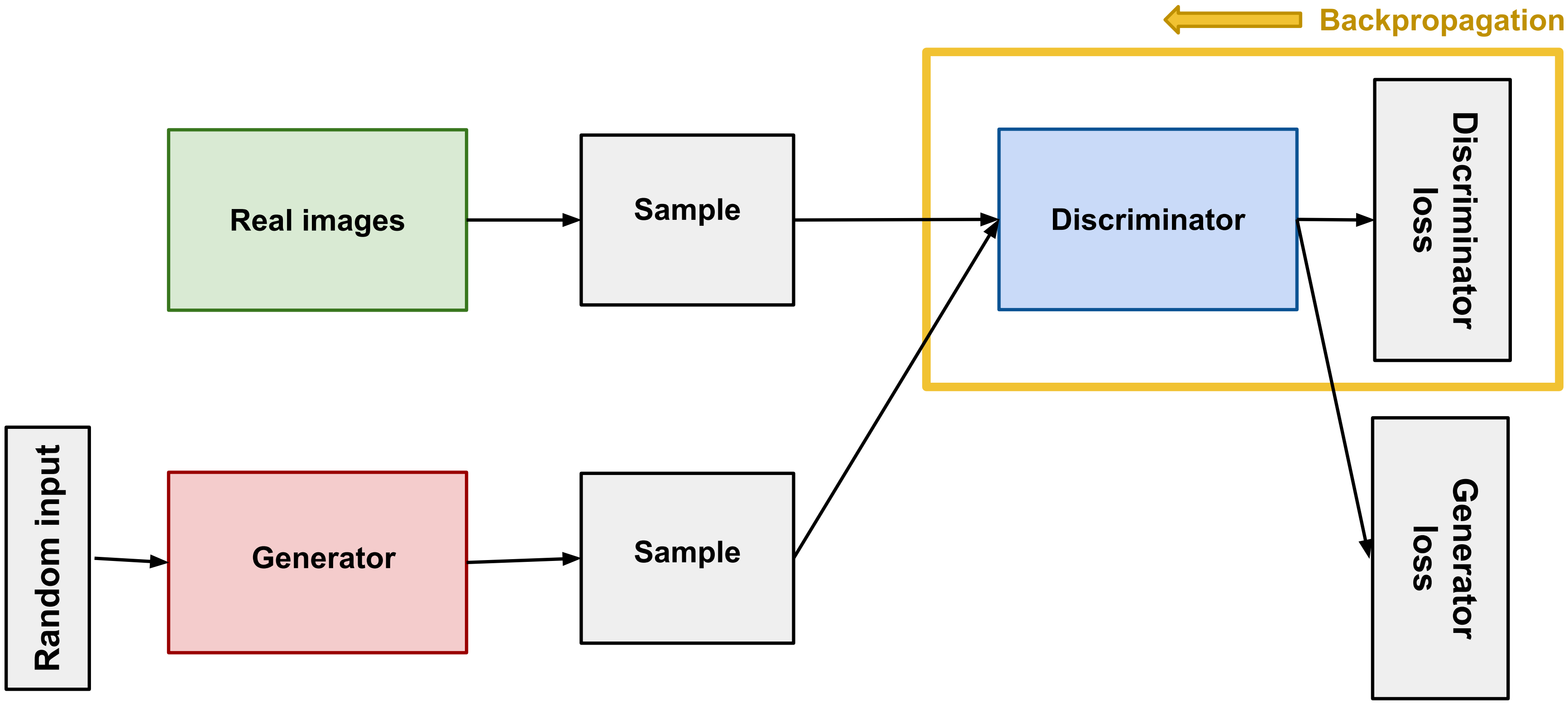
02.
GAN : Generative Adversarial Nets

GAN

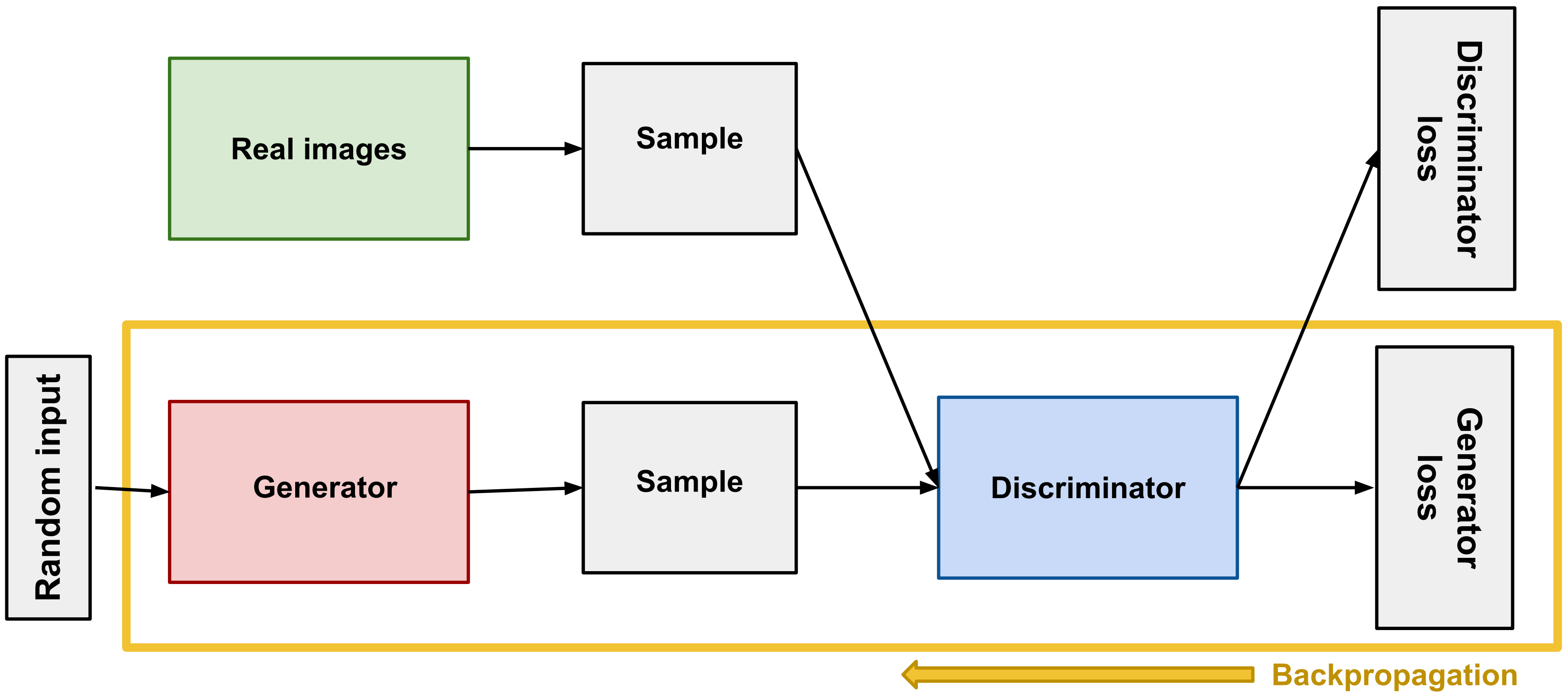
직관적 이해


Real images
Sample
Random input
Generator
Sample
Discriminator
Discriminator loss
Generator loss
0
5
10
15
20
25
0
5
10
15
20
25

GAN


Backpropagation
Real images
Sample
Discriminator
Discriminator loss
Random input
Generator
Sample
Generator loss

GAN


Real images
Sample
Discriminator loss
Random input
Generator
Sample
Discriminator
Generator loss
Backpropagation

GAN – Object Function

Sample x from real data

Sample latent code z from Gaussian distribution

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

Maximum when D(x) = 1

Maximum when D(G(z)) = 0

GAN – Object Function

D의 입장에서 보면…

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

**이 친구는 0이 될 때
D입장에서는 최적**

D(G(z)) = 0, 따라서 log(1) = 0

즉, D의 입장에서는 V(D, G)가 = 0이 되는 것이 `최댓값` = 이상적 결과

GAN – Object Function

G의 입장에서 보면

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

이 친구는 상관없는 상수

D(G(z)) = 1, 즉, log(0) = - ∞

즉, G입장에서의 이상적인 '최솟값' = - ∞

GAN – Object Function

$$\max_D V(G,D) = E_{x\sim p_{data}}[\log D(x)] + E_{x\sim P_G}[\log(1 - D(x))]$$

실제 이미지를 가지고 판별한 결과,

$$D(x) = \begin{cases} 1 \rightarrow \log(1) = 0 \\ 0 \rightarrow \log(0) = -\infty \end{cases}$$

fake 이미지를 가지고 판별한 결과,

$$D(x) = \begin{cases} 1 \rightarrow \log(1-1) = -\infty \\ 0 \rightarrow \log(1-0) = 0 \end{cases}$$

즉, GAN 모델에서
D는 Object Function을 최댓값으로
G는 Object Function을 최솟값으로

하기위해서 two-play min-max game 하는 것과 같다

GAN – Non-Saturating Game

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

log(D(G(z)))

그런데 여기서 문제점 = GAN 모델을 학습시키기 어려운 점

학습 초기에 G모델이 만들어 내는 이미지 = 품질이 매우 낮음
따라서 D모델이 G가 만든 모델이 무조건 형편없다고 생각

학습 초기에 G loss를 Log(D(G(z))를 최대화 하는 함수로
수정하여 학습

GAN – Training 방법

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.

adversarial nets의 data의 분포 학습하는 과정

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:

(kstep)

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

최대화

  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:

(1 step) G

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

최소화

end for

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

GAN – Theoretical Results

- Why does GANs work?

Because it actually minimizes the distance between the real data distribution and the model distribution.

$$\min_G \max_D V(D,G) \xrightarrow{\text{same}} \min_{G,D} JSD(p_{data}||p_g)$$

Objective function of GANs

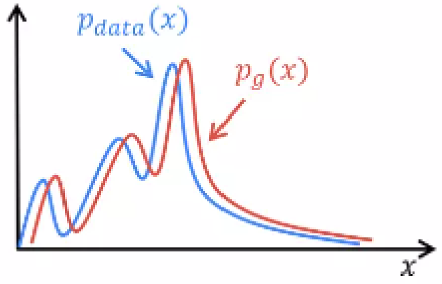
$$E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

Jenson-Shannon divergence

$$JSD(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M)$$

$$where\ M = \frac{1}{2}(P+Q)$$

KL Divergence



Please see Appendix for details

GAN – Theoretical Results

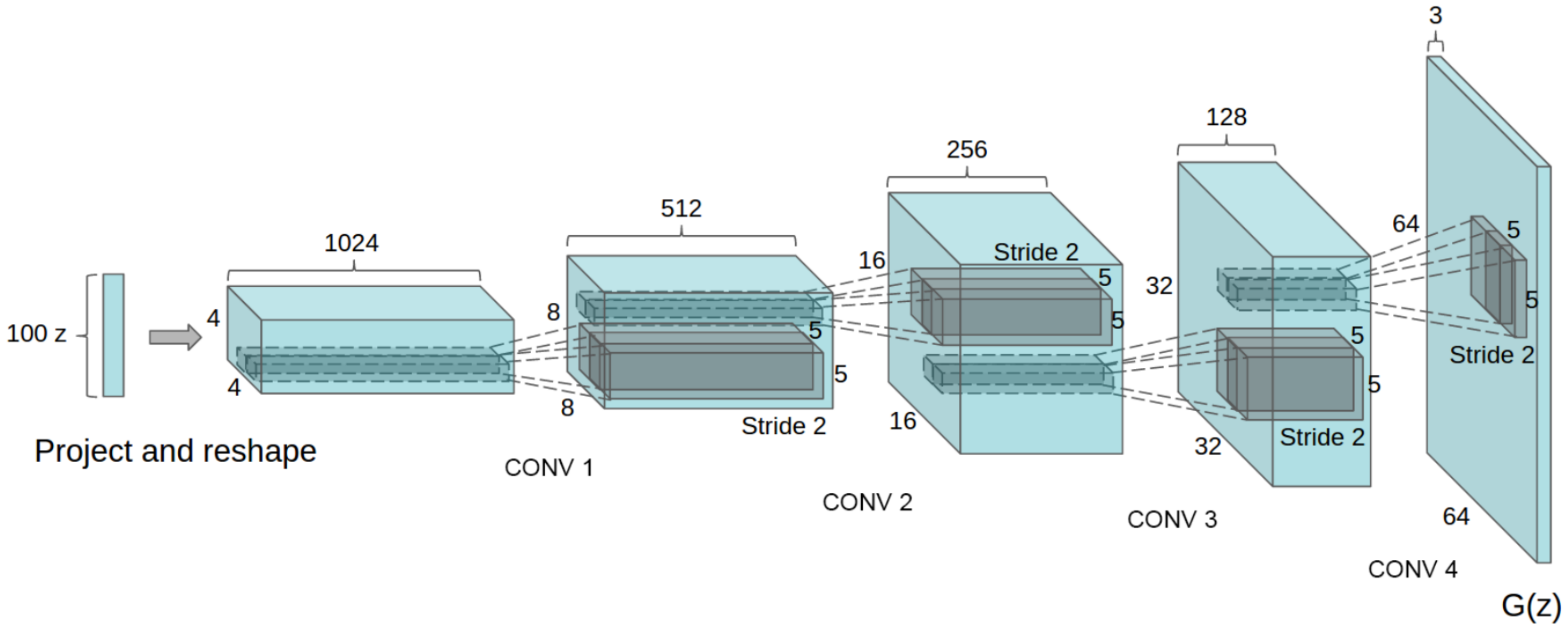
https://memesoo99.tistory.com/27

https://tobigs.gitbook.io/tobigs/deep-learning/computer-vision/gan-generative-adversarial-network

03.
Variants of GAN

1. DCGAN
2. CGAN
3. Cycle GAN

Variants of GAN - DCGAN



Variants of GAN - DCGAN

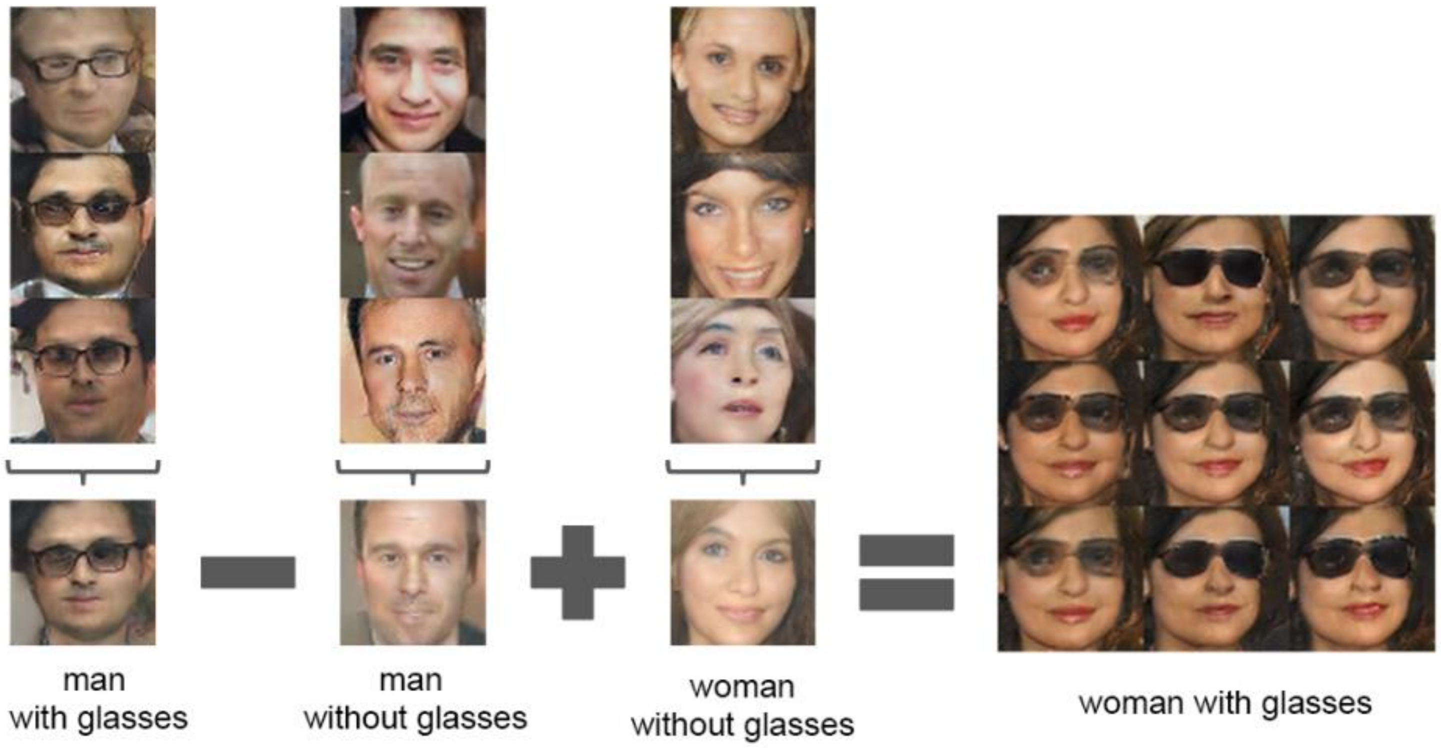
GAN은 고해상도 이미지 생성 불가,
학습 불안정 등 문제

따라서 DCGAN은 Convolution Layer를 MLP 대신활용

특징

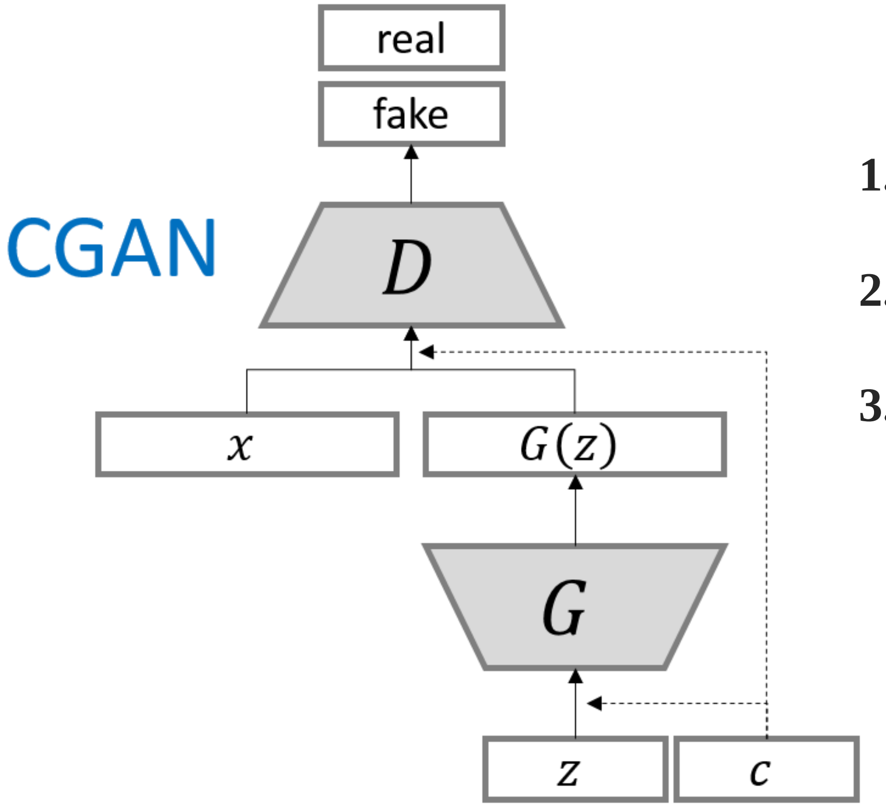
1. D모델에서는 Pooling Layer 제거한 Conv
2. G모델에서는 Pooling Layer 제거한 Deconv (transfer conv)

Variants of GAN - DCGAN



DCGAN을 통해서 얻은 Z를 통해서
다양한 응용 가능

Variants of GAN - CGAN

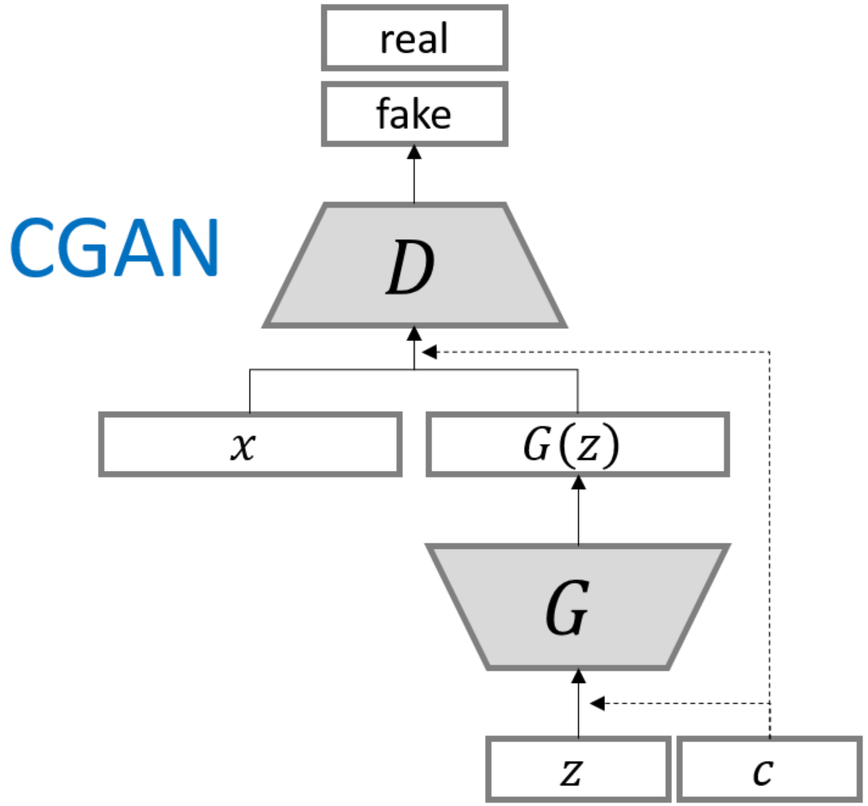


1. GAN의 출력을 제어하기 위해서 C를 추가

2. 원래 GAN의 Output으로 나오는 결과를 제어할 수 없었음

3. 원래 GAN은 Noise만 넣었다면 여기에 C라는 조건을 추가

Variants of GAN - CGAN

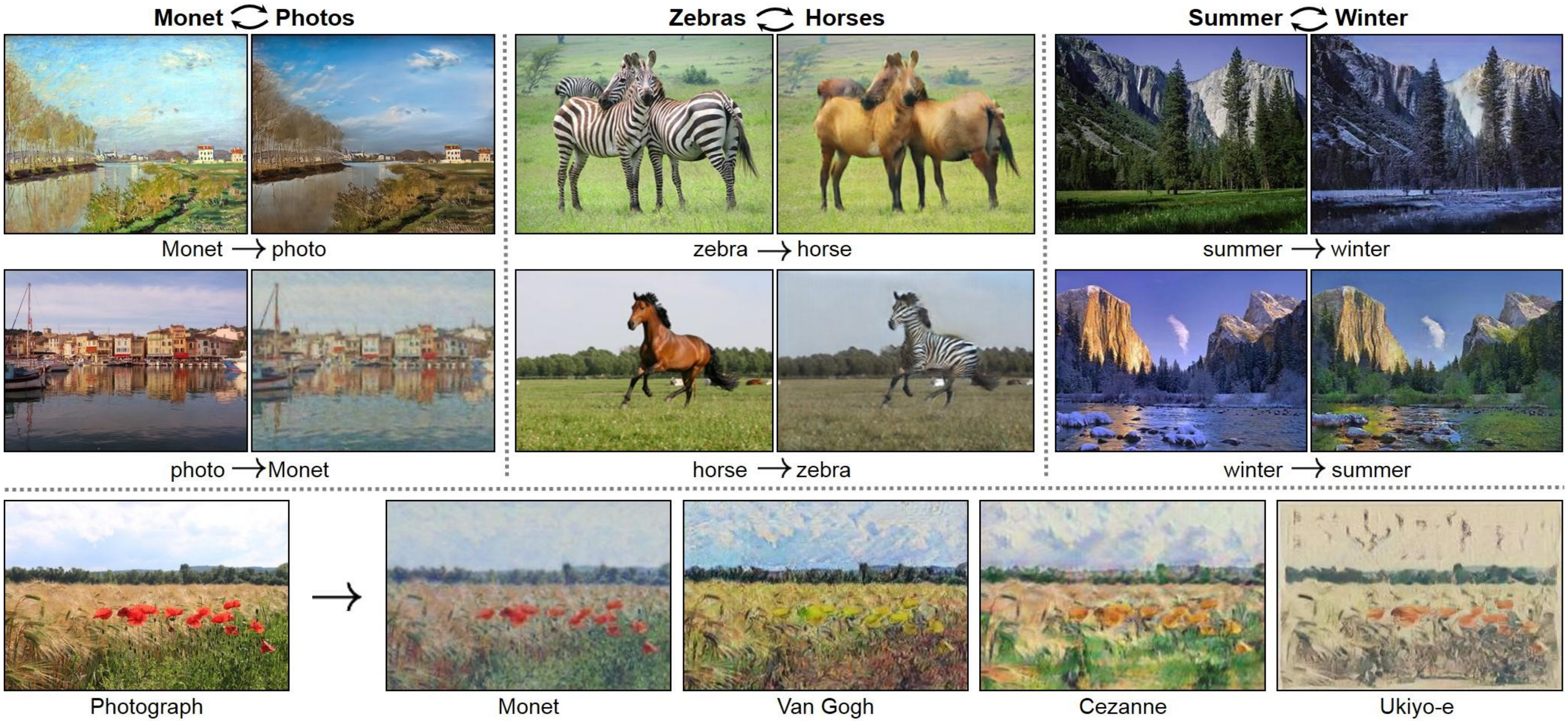


숫자 이미지를 예시로 들면

만약 0숫자 이미지를 만들고 싶다

1. 0을 입력, 0을 one-hot encoding
2. One hot encoding 된 C와 Z를 concate
3. 이를 G의 input으로 투입

Variants of GAN - Cycle GAN



Variants of GAN - Cycle GAN