

가천대 회화·조소과 AI 특강

2021-06-26

조형래

GAN

Generative Adversarial Nets

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley,
Sherjil Ozair[†], Aaron Courville, Yoshua Bengio[†]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

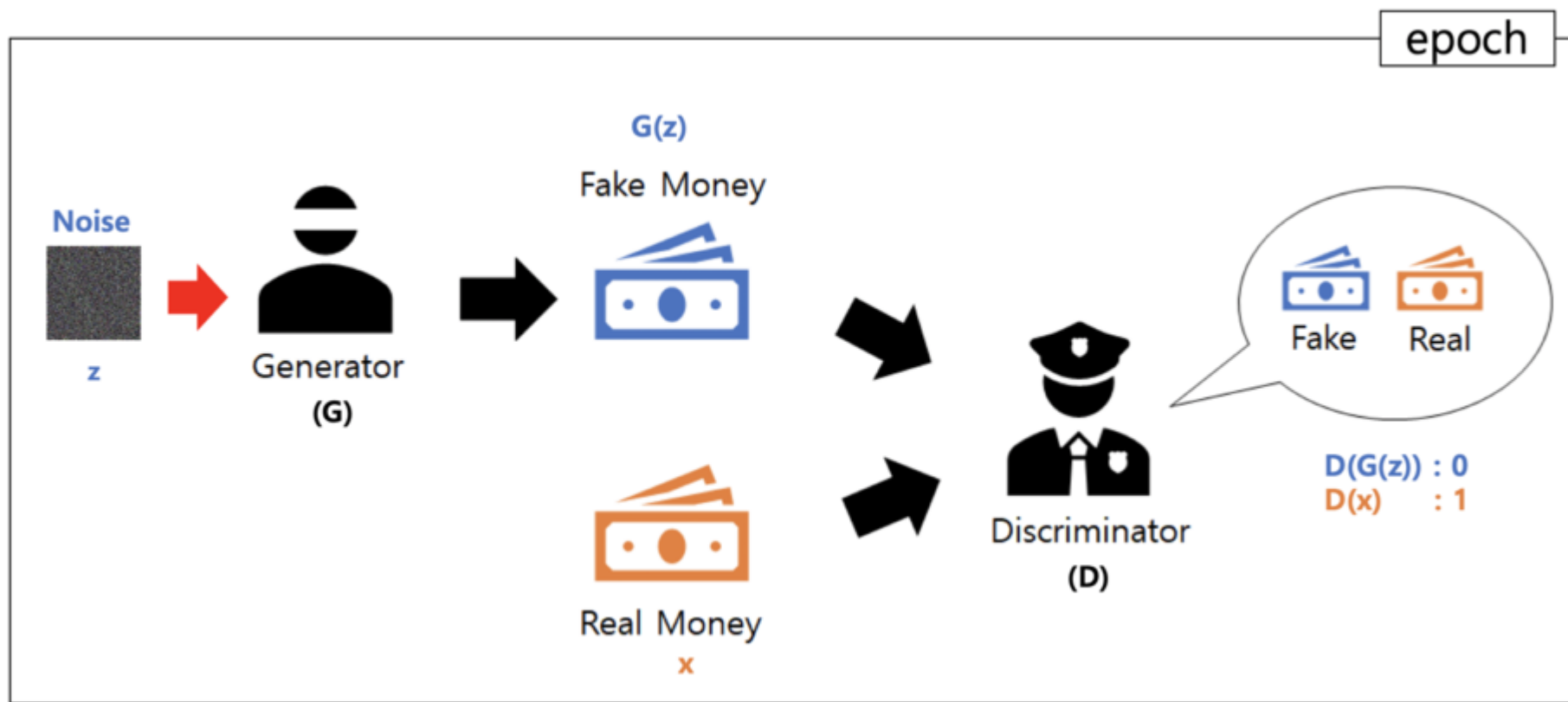
We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G . The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D , a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

<https://arxiv.org/pdf/1406.2661.pdf>

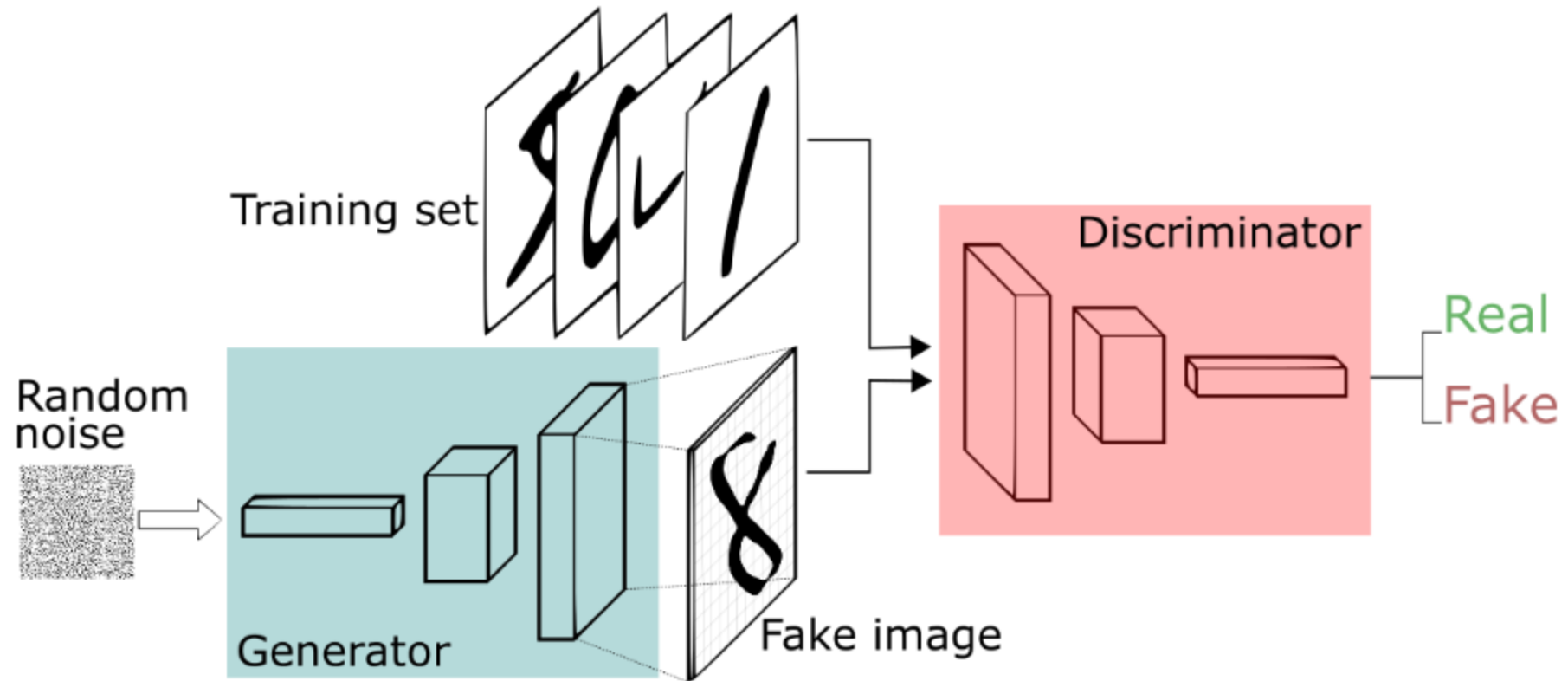
GAN (Generative Adversarial Network)

GAN은 Generator (생성자)와 Discriminator (판별자) 두 개의 모델이 동시에 적대적인 과정으로 학습
생성자 G는 실제 데이터 분포를 학습하고, 판별자 D는 원래의 데이터인지 생성자로부터 생성이 된 것인지 구분한다.

생성자 G의 학습 과정은 이미지를 잘 생성해서 속일 확률을 높이고 판별자 D는 제대로 구분하는 확률을 높이는 두 플레이어의 minmax game의 과정



GAN (Generative Adversarial Network)



GAN (Generative Adversarial Network)

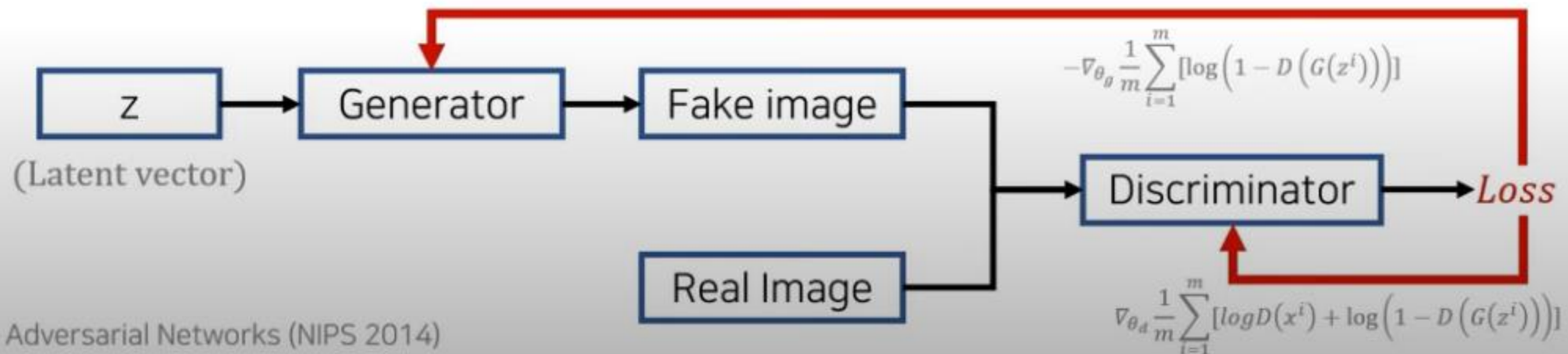
$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Generator

$G(z)$: new data instance

Discriminator

$D(x)$ = Probability: a sample came from the real distribution (Real: 1 ~ Fake: 0)



Generative Adversarial Networks (NIPS 2014)

$\log(1)=0$, $\log(10)=1$, $\log(0)=-\text{infinity}$ (무한대)

Discriminator 입장

Sample x from real data distribution

Sample latent code z from Gaussian distribution

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

D should maximize $V(D, G)$

Maximum when $D(x) = 1$

Maximum when $D(G(z)) = 0$

판별자의 목표=>

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)} [\log \overset{D(x)=1}{D(x)}] + \mathbb{E}_{z \sim P_Z(z)} [\log(1 - \overset{D(G(z))=0}{D(G(z))})]$$

$V(D, G)$ 은 GAN의 Loss 함수, Objective 함수
 D 의 목적은 $V(D, G)$ 가 최대가 되도록 하는 것을 의미한다.
 $\log D(x)$ 가 최대가 되도록 하는 것

Generator 입장

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

\swarrow
G should minimize $V(D, G)$

\swarrow
G doesn't care

\uparrow
Minimum when $D(G(z)) = 1$

생성자의
목표=>

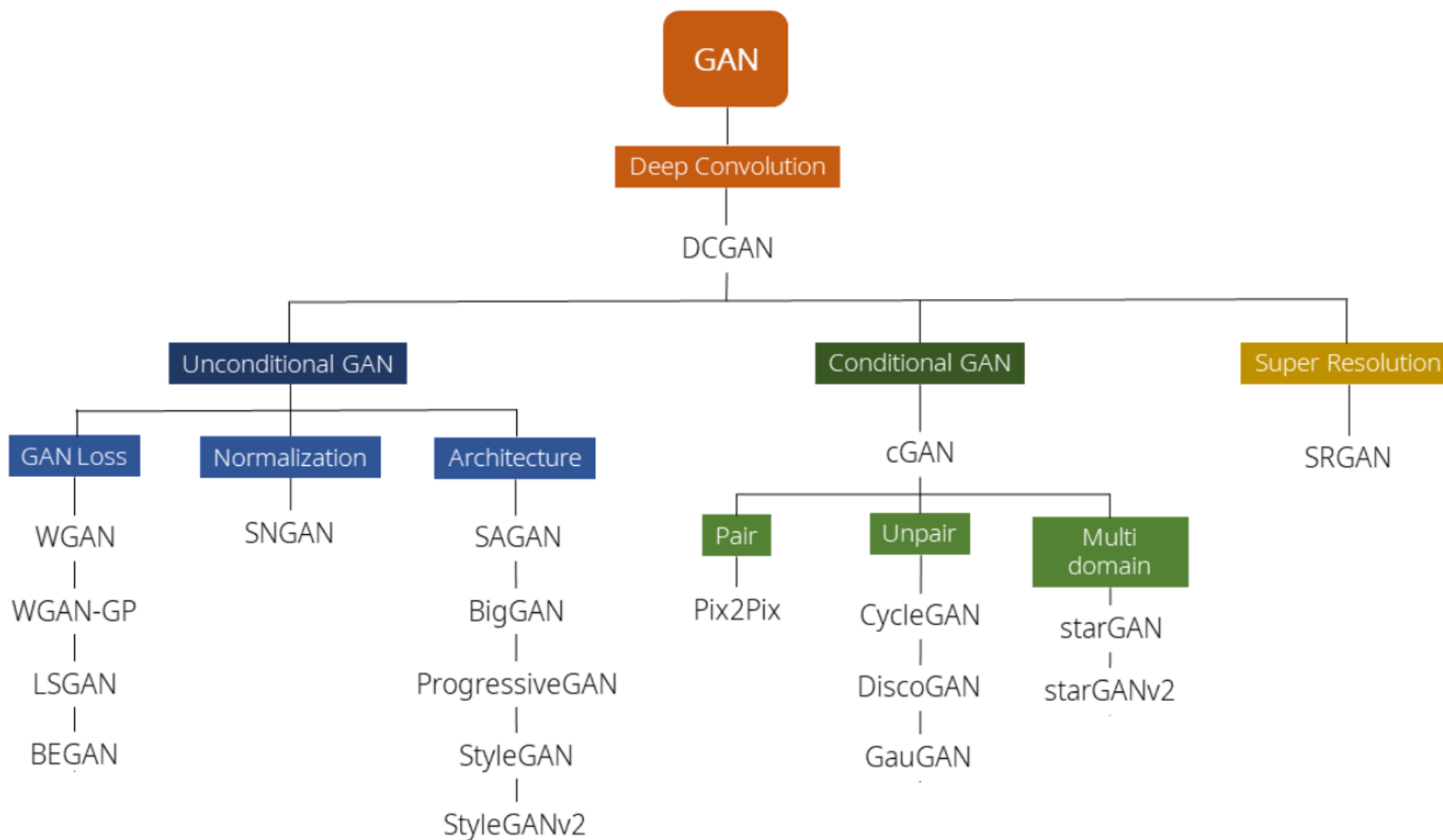
$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim P_Z(z)} [\log(1 - \overset{D(G(z)) = 1}{D(G(z))})]$$

$\log(1-1)$ 으로 인해 $\log 0$ 이 된다면 해당 값은 $-\infty$ 를 향하게 된다.

이는 0보다도 매우 작은 무한대의 음수로, 결국 G의 목표는 0 보다 매우 작은 음의 무한대를 향하는 것이 목적

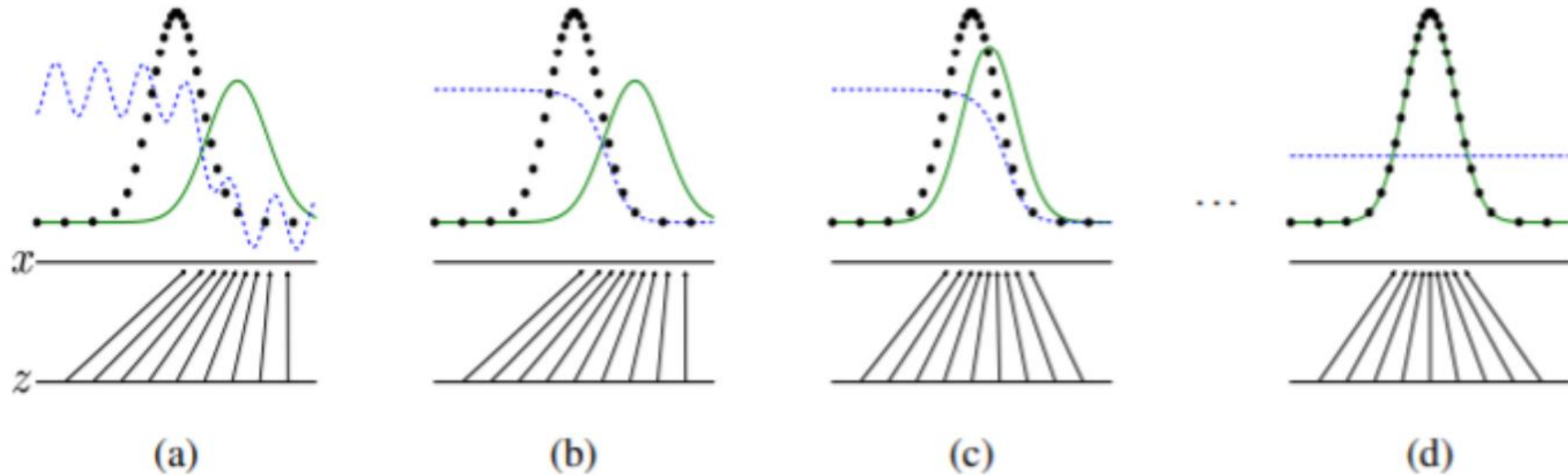
따라서 **G가 원하는 최적의 상황은 매우 작은 음의 무한대가 되는 방향이 되는 것이다.**

주요 GAN의 종류



GAN (Generative Adversarial Network)

- Ian Goodfellow가 최초로 GAN (Generative Adversarial Nets)를 제안한 논문.



검정실선; P_{data} 의 분포

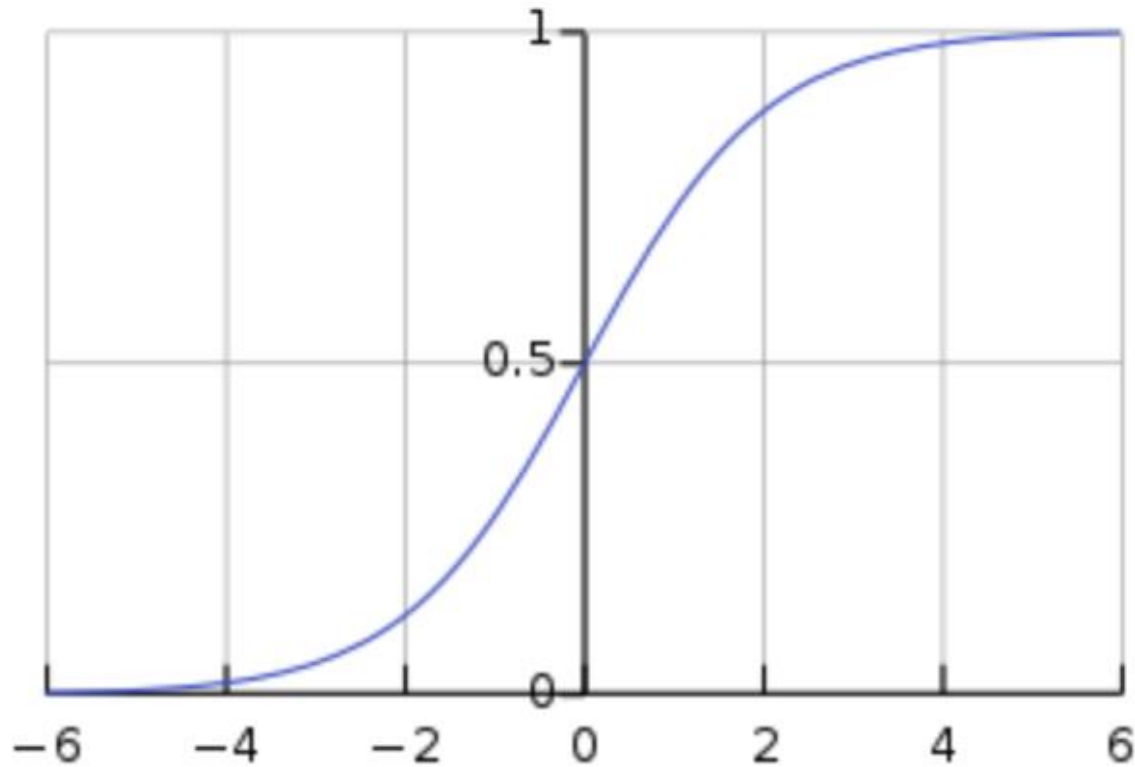
초록실선; 생성자의 분포 P_g

파랑점선; 판별자모델 $D(x)$

=> 저차원 벡터인 임의의 z 를 이미지 생성을 위한 고차원 이미지인 x 에 매핑 시키는 작업

GAN의 판결

최종적으로 sigmoid 함수를 써서 0.5를 기준으로 0과 1을 구분하게 된다.
0이면 가짜 이미지, 1이면 진짜 이미지로 판정



Sigmoid 함수

GAN Loss= Loss function

1. Mean Square Error(MSE) : 평균 제곱 오차

$$\frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2$$

손실 함수 코딩 실습)

[https://github.com/artjow/-AI-/blob/main/%EC%88%98%EC%97%85/%EC%88%98%EC%97%852/%EC%86%90%EC%8B%A4%ED%95%A8%EC%88%98\(Loss_Function\).ipynb](https://github.com/artjow/-AI-/blob/main/%EC%88%98%EC%97%85/%EC%88%98%EC%97%852/%EC%86%90%EC%8B%A4%ED%95%A8%EC%88%98(Loss_Function).ipynb)

GAN Loss= Loss function

2. Cross Entropy Error(CEE) : 교차 엔트로피 오차

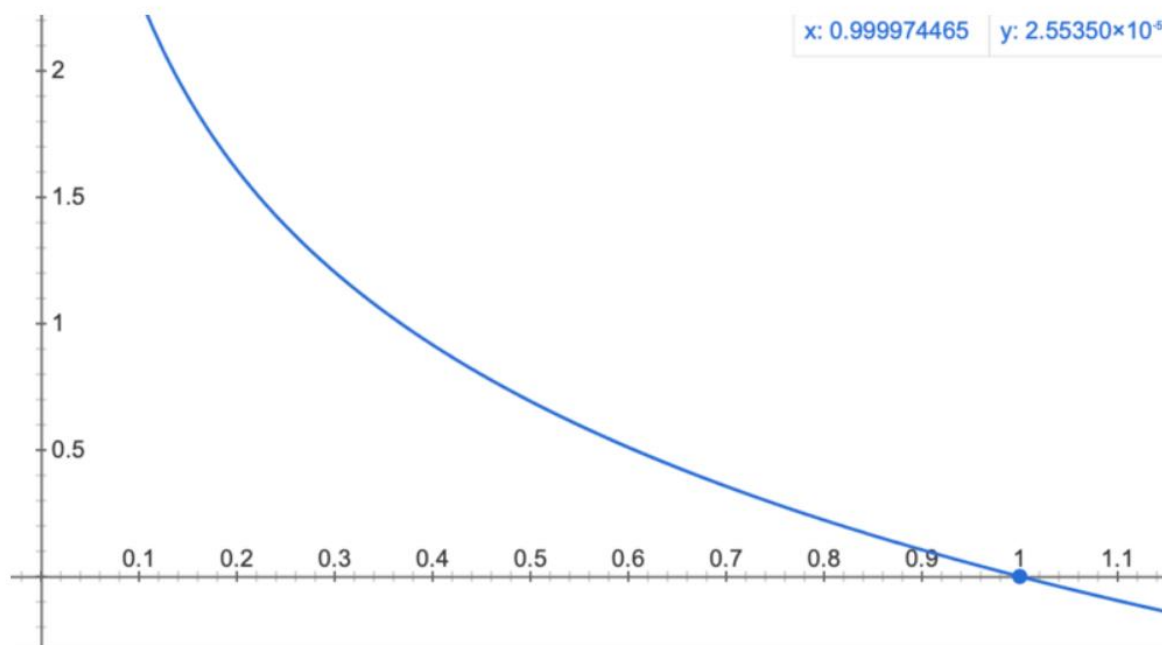
$$-\sum_{i=1}^n \ln(y_i) \times t_i$$

** 교차 엔트로피는 정답 값이 원한 인코딩인 경우에만 사용할 수 있다.

$-\ln(x)$ 그래프

x가 1일 때 y는 0이 되는 것을 확인할 수 있다.
x가 1이라는 것은 정답이라는 것을 의미하므로
오차를 의미하는 cost 값은 0이 되는 것을
의미한다.

반대로 x가 0에 점차 가까워질수록 y 값이 커지게
되는데,
이는 정답과 점차 멀어지므로 오차를 의미하는
cost가 증가하게 된다는 것을 의미한다.



DCGAN.ipynb 실습

[https://github.com/artjow/-AI-
/blob/main/%EC%9D%B4%EB%A1%A0/dcgan.ipynb](https://github.com/artjow/-AI-/blob/main/%EC%9D%B4%EB%A1%A0/dcgan.ipynb)

[https://github.com/artjow/-AI-
/blob/main/%EC%9D%B4%EB%A1%A0/Vanilla_GAN_.ipynb](https://github.com/artjow/-AI-/blob/main/%EC%9D%B4%EB%A1%A0/Vanilla_GAN_.ipynb)