

---

# GAN Basic

## GAN Paper

---

## •배경

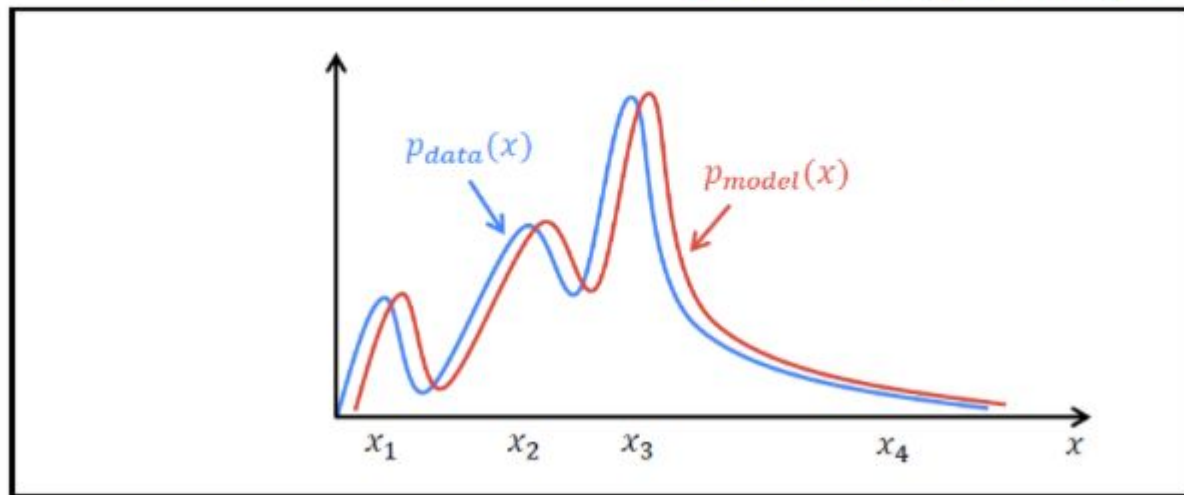
- 지금까지는, 고차원의 방대한 센싱 데이터를 클래스 레이블에 mapping해서 구분하는 모델 사용  
-> 많은 비용

## •목표

- Training data  $P_{data}(x)$ 과 비교하였을 때 구분할 수 없을 정도로 유사한 가짜 데이터를 생성해낼 수 있도록 가짜 데이터의 분포  $P_{model}(x)$ 를 찾는 것이다.

The goal of the generative model is to find a  $p_{model}(x)$  that approximates  $p_{data}(x)$  well.

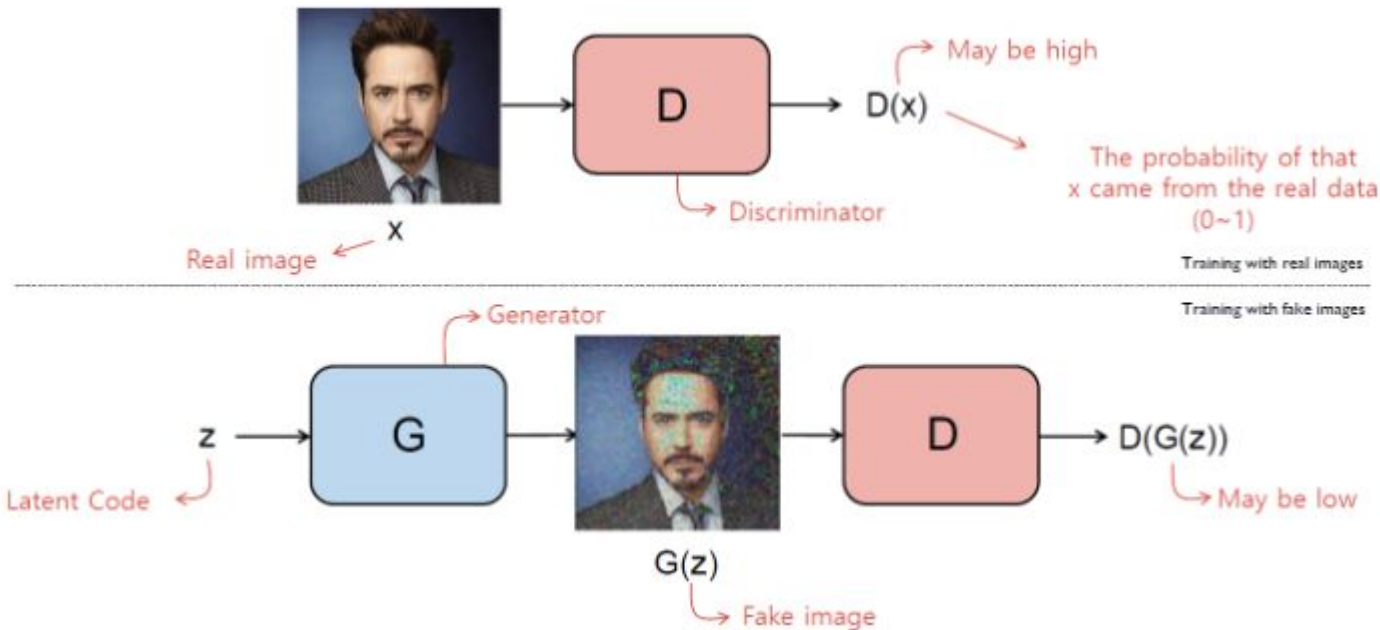
↪ Distribution of actual images



GAN의 핵심 컨셉은 각각의 역할을 가진 두 모델을 통해 적대적 학습을 하면서 '진짜같은 가짜'를 생성해내는 능력을 키워주는 것

## Discriminator

- D의 역할은 주어진 input이 real data인지 구별하는 것이다. Data  $x$ 가 input으로 주어졌을 때, D의 output  $D(x)$ 는  $x$ 가 real data일 확률을 return한다.



## Generator

- G(Generator)의 역할은 D(Discriminator)가 진짜인지 구별할 수 없을 만큼 진짜같은 Fake data를 만들어내는 것이다

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

GAN 손실함수를 maximize하는 방향

## • Discriminator

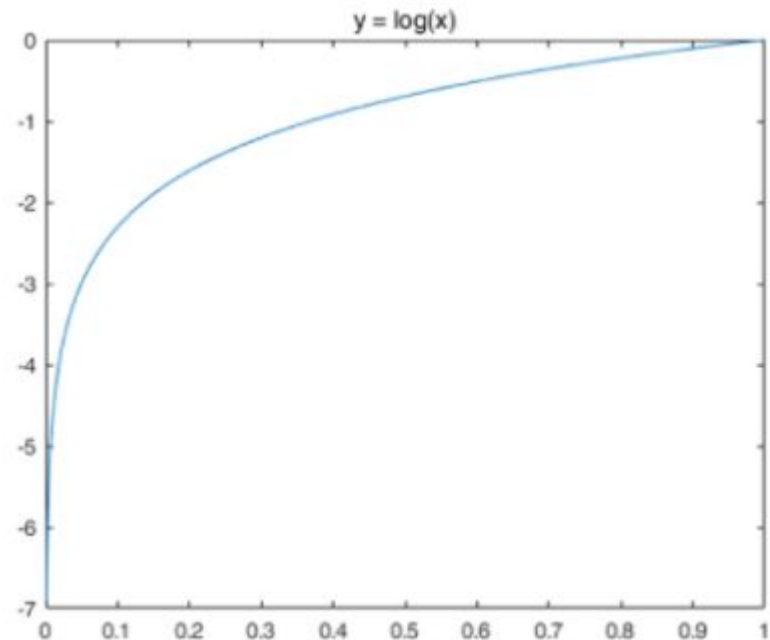
D의 경우,  $V(D, G)$ 를 D에 대하여 Maximize하는 방향으로 weight를 update하게 된다

$$\begin{aligned}\max_D V(D, G) &= E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \\ &= \frac{1}{m} \sum_{i=1}^m \log D(x_i) + \frac{1}{m} \sum_{i=1}^m \log \{1 - D(G(z_i))\}\end{aligned}$$

이상적인 Case:  $D(x) = 1$ , 즉  $\log(D(x)) = 0$

$\log(D(x))$ 의 범위는  $[-\infty, 0]$  이므로, 0으로 만든다는 것은 maximize!

$$\begin{aligned}x \sim p_{data}(x) &\iff D(x) \approx 1 \\ &\iff \log[D(x)] \approx 0 \\ &\iff \text{Maximizing } \log[D(x)] \quad (\because D(x) \in [0, 1])\end{aligned}$$



GAN 손실함수를 maximize하는 방향

## • Discriminator

D의 경우,  $V(D, G)$ 를 D에 대하여 Maximize하는 방향으로 weight를 update하게 된다

$$\begin{aligned}\max_D V(D, G) &= E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \\ &= \frac{1}{m} \sum_{i=1}^m \log D(x_i) + \frac{1}{m} \sum_{i=1}^m \log \{1 - D(G(z_i))\}\end{aligned}$$

latent  $z$  sample :  $D(G(z))$ 는 0에 가까워야 하므로  $1 - D(G(z)) = 1$

$\log(1 - D(G(z)))$ 의 범위는  $[-\infty, 0]$  이므로, 0으로 만든다는 것은 maximize!

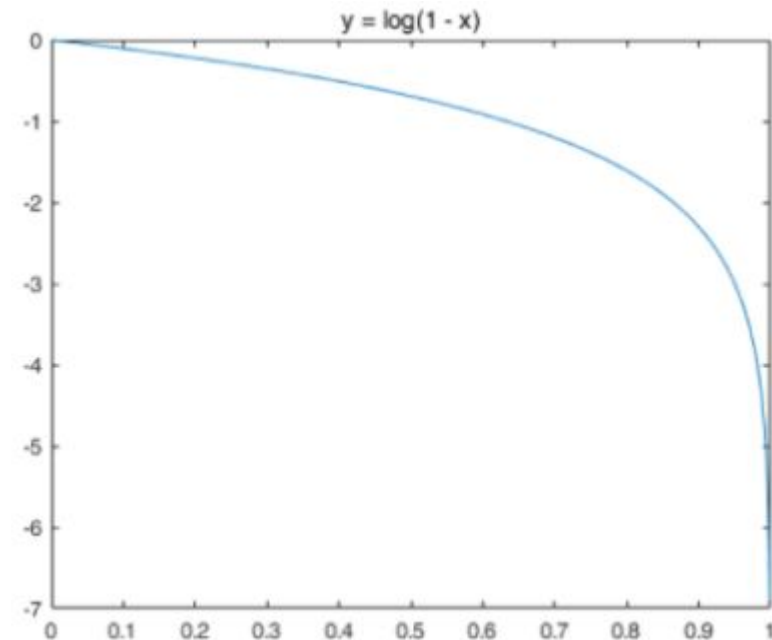
$z \sim p_z(z) \rightarrow \text{generate } G(z)$

$$\iff D(G(z)) \approx 0$$

$$\iff 1 - D(G(z)) \approx 1$$

$$\iff \log[1 - D(G(z))] \approx 0$$

$$\iff \text{Maximizing } \log[1 - D(G(z))] \quad (\because D(G(z)) \in [0, 1])$$



GAN 손실함수를 minimize하는 방향

## • Generator

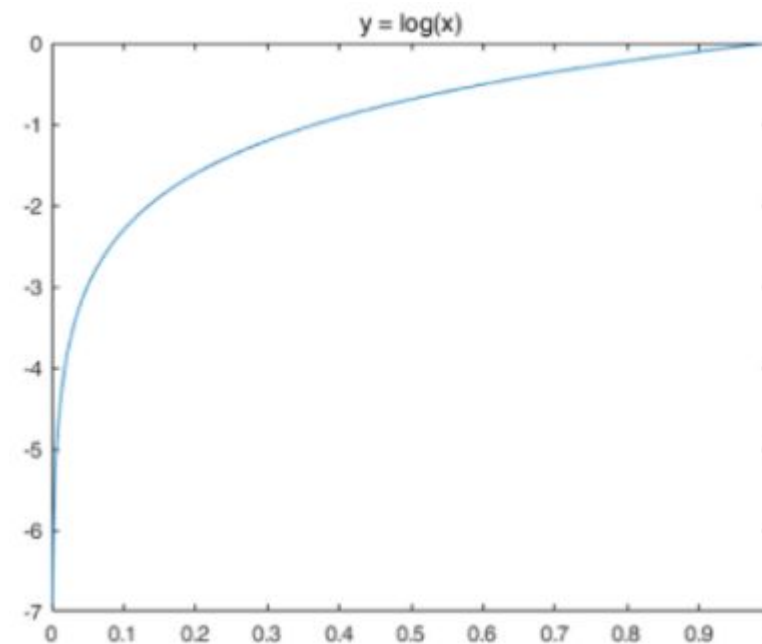
G의 경우,  $V(D, G)$ 를 G에 대하여 Minimize하는 방향으로 weight를 update하게 된다  
G에 대한 목적 함수에서는 진짜 데이터가 들어오는 경우를 고려할 필요가 없으므로,  
앞의 항을 삭제한다

$$\begin{aligned}\min_G V(G) &= E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \\ &= \frac{1}{m} \sum_{j=1}^m \log\{1 - D(G(z_j))\}\end{aligned}$$

G 입장에서는 D가 진짜로 판별할 수 있도록  $D(G(z))$ 값이 1이 되어야 함  
 $1 - D(G(z))$ 의 값이 0이 되도록 한다는 것은  $\log$ 값을 최소화 시킨다는 의미!

$$\begin{aligned}z \sim p_z(z) &\rightarrow \text{generate } G(z) \\ &\iff D(G(z)) \approx 1 \\ &\iff 1 - D(G(z)) \approx 0 \\ &\iff \text{Minimize } \log[1 - D(G(z))] \quad (\because D(G(z)) \in [0, 1])\end{aligned}$$

출처: [https://kjhov195.github.io/2020-03-09-generative\\_adversarial\\_network/](https://kjhov195.github.io/2020-03-09-generative_adversarial_network/)



```
Discriminator.trainable = False
```

Discriminator가 너무 세지면 Generator가 더이상 학습되지 않음

```
discriminator.trainable = True
```

```
gan.discriminator.trainable = False
```

- discriminator 학습 시킬 때는 가중치를 업데이트
- 이 discriminator가 gan 안에서 generator와 같이 학습할 때는 가중치를 고정
- discriminator network는 discriminator 모델과 gan 모델에 둘 다 사용되고, 가중치도 공유
- gan 모델에서의 discriminator 네트워크는 단순 가중치를 가진 네트워크로만 받아들인다.