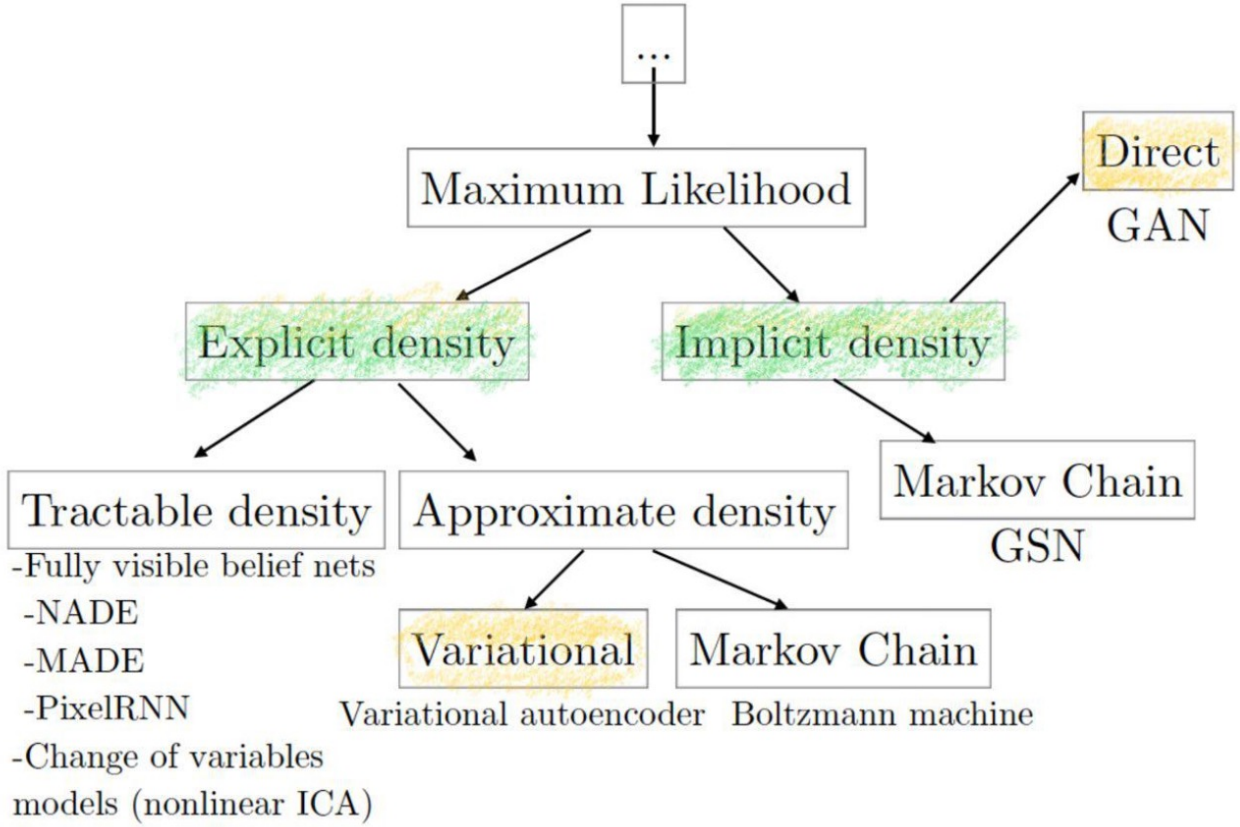
# Generative Adversarial Nets

# Explicit vs Implicit




Explicit Tractable Density

모델의 분포를 가정하고 기존 값으로 부터 분포 추정

Explicit Approximate Density

분포를 근사시켜서 추정

Implicit Density

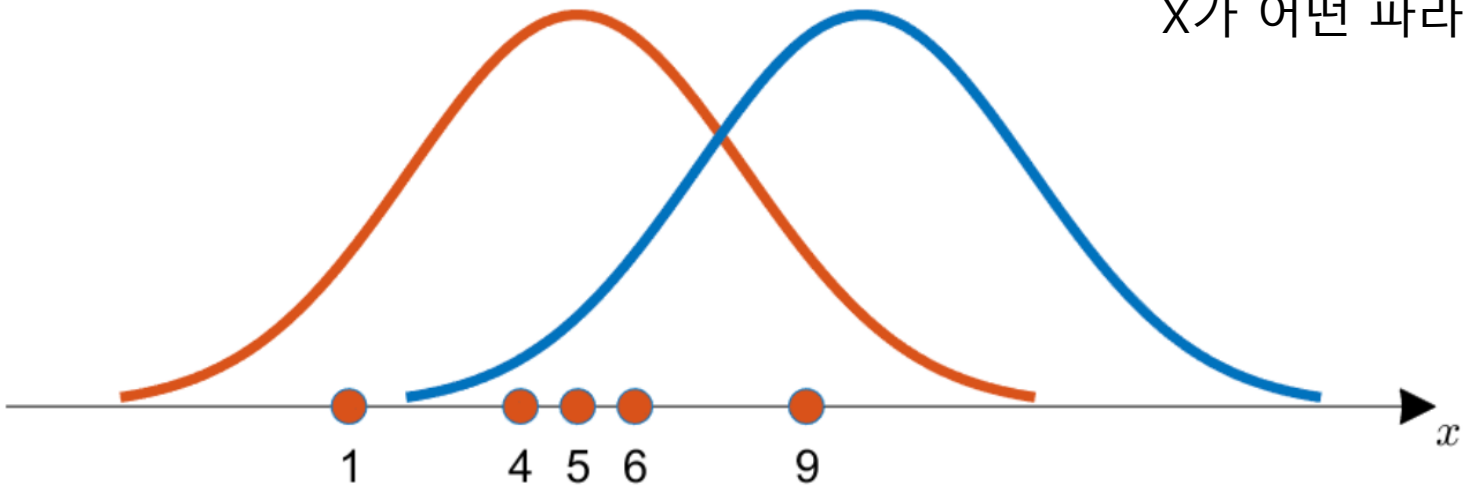
데이터의 분포를 모르는 상태로 샘플링을 통해 특정 분포에 수렴시켜서 분포 추정

# Maximum Likelihood Estimation

- 어떤 파라미터 θ로 이루어진 확률밀도함수 P($\boldsymbol{x}$| θ)에서 표본들의 집합을 $\boldsymbol{x}$라고 할 때 θ를 추정하는 방법

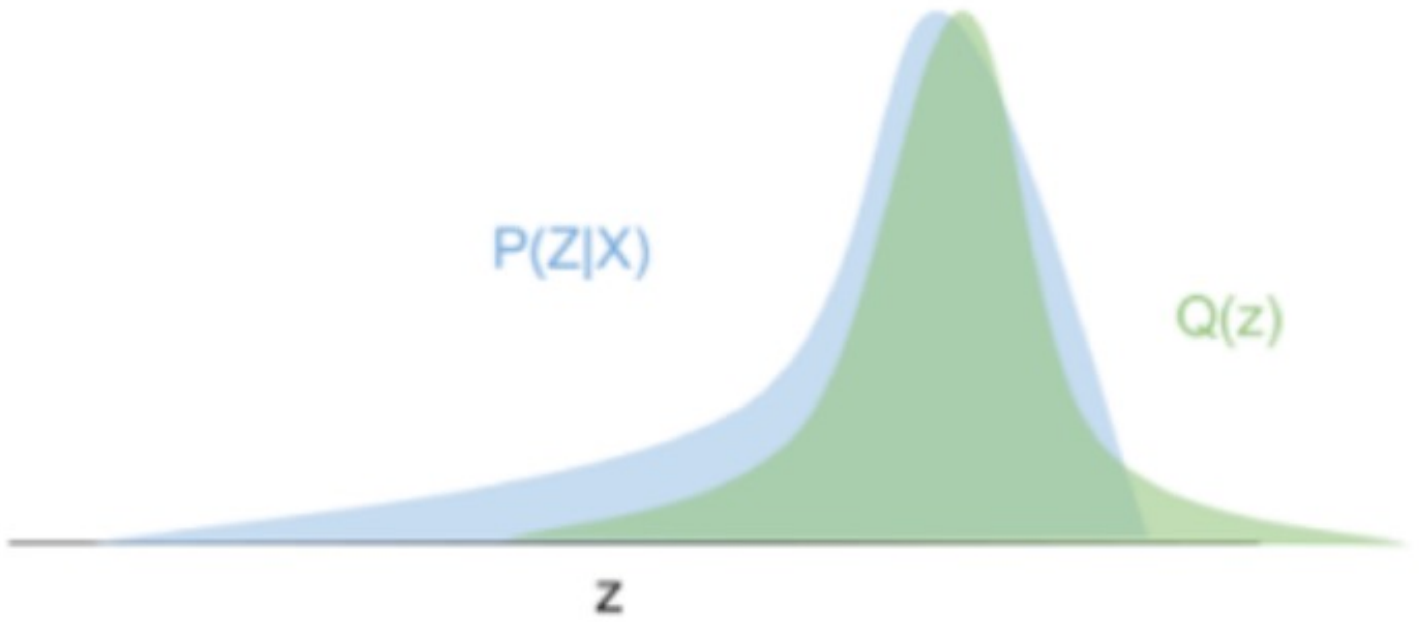
$$\mathbf{X} \sim \mathcal{N}(\theta) = \mathcal{N}(m, \boldsymbol{\sigma}\text{^}2)$$

X가 어떤 파라미터에서 나왔을까?

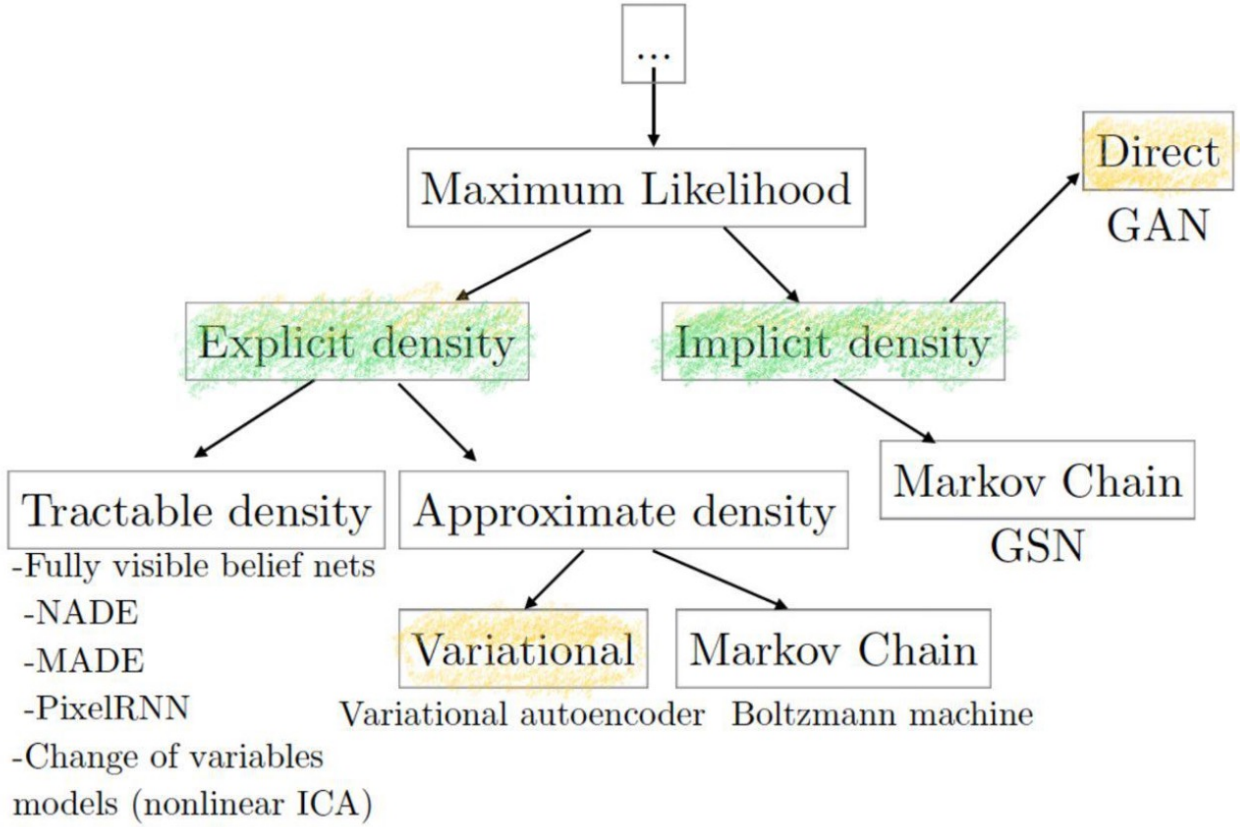



출처: https://angeloyeo.github.io/2020/07/17/MLE.html

# Variational inference



# Explicit vs Implicit

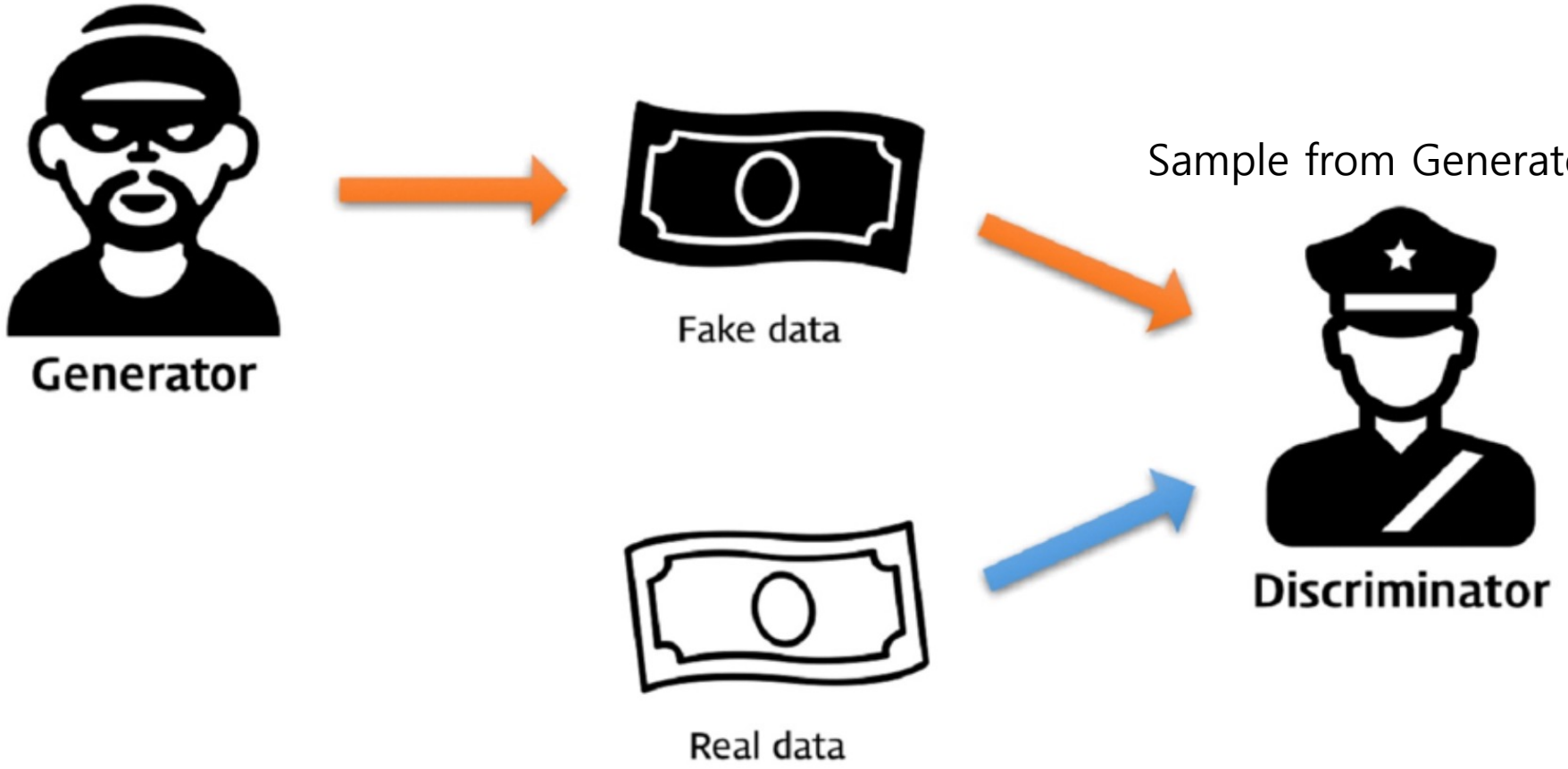



결국 Explicit density에서는 특정 분포와 파라미터를 통해 데이터 설명이 가능하다.

Implicit density에서는 수식적으로 특정 분포로 설명할 수는 없지만 분포에서 수많은 sample를 제시함으로써 분포를 표현할 수 있다.

# Vanilla GAN

Capture Data Distribution

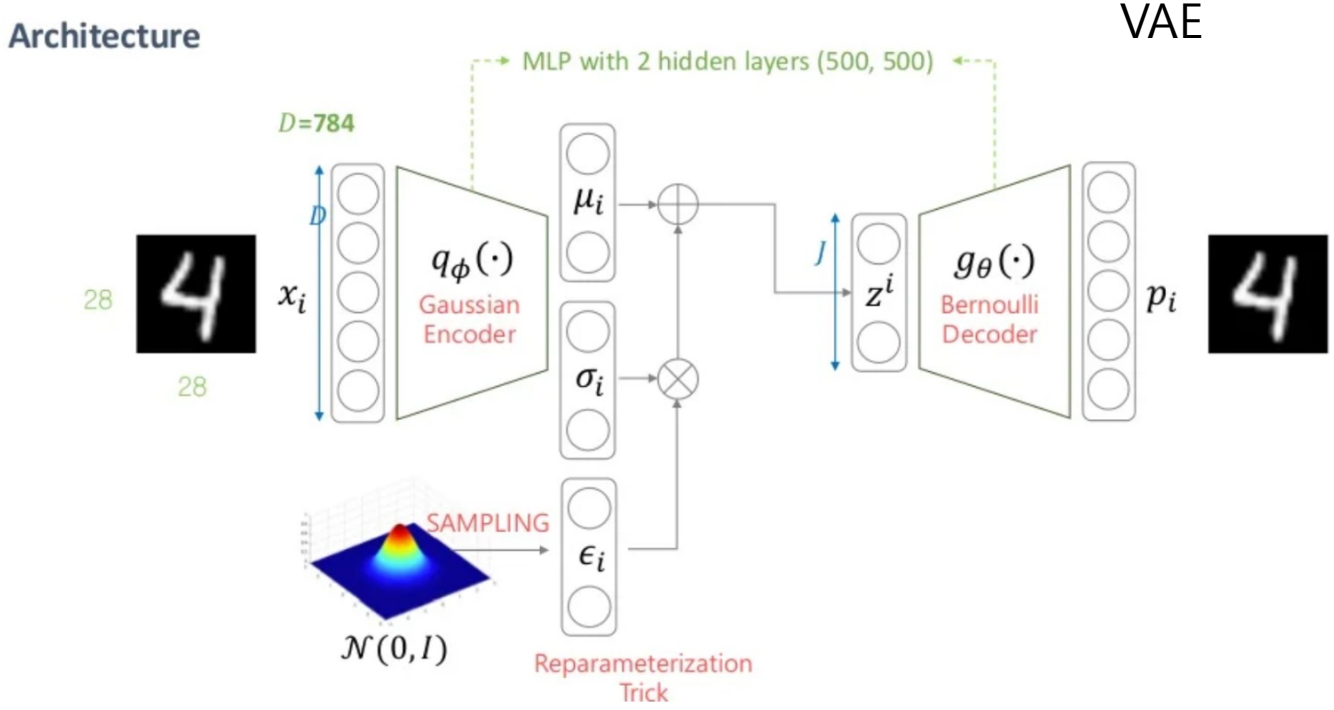



잘 학습이 된 상태라면 Generator는 Training data를 recovering, Discriminator는 0.5의 확률을 갖게 된다.

# Vanilla GAN

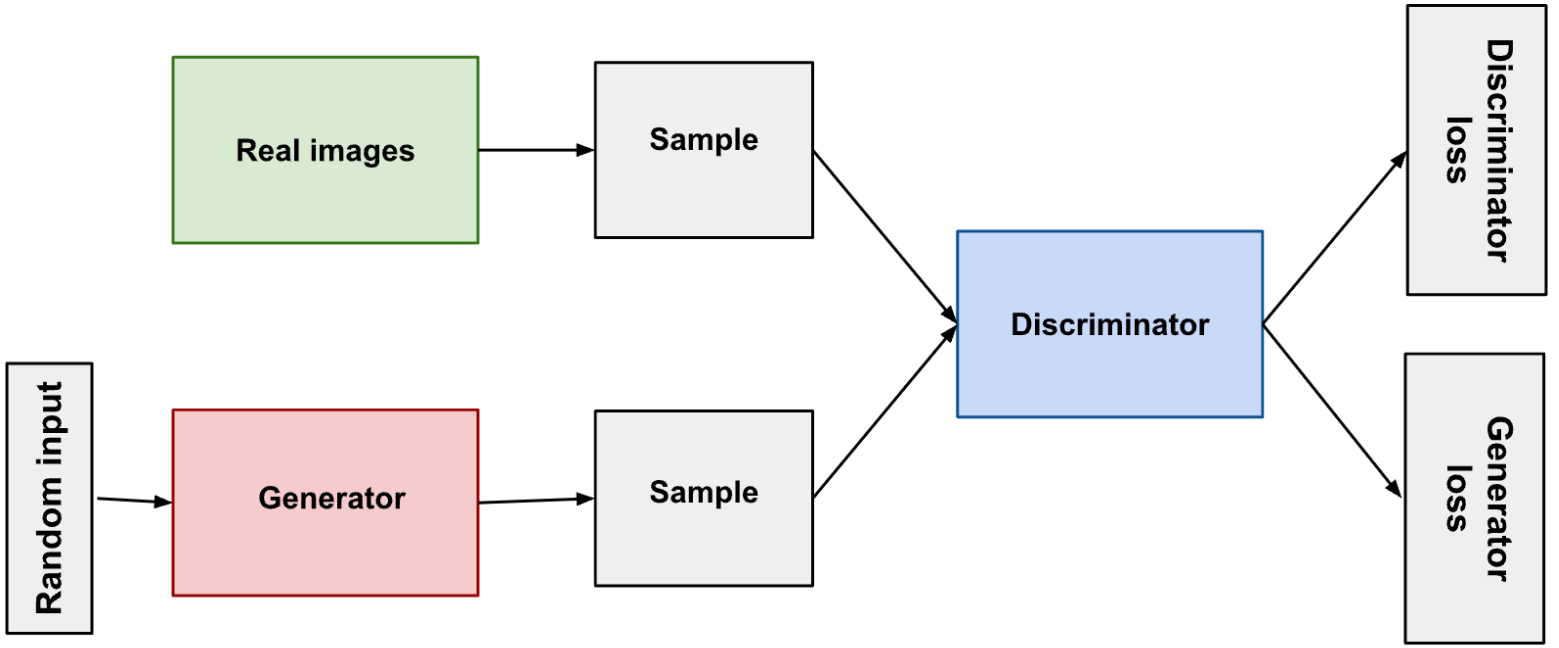
- Deep generative model은 MLE를 계산하기 힘든 intractable problem이 있었고(보통 데이터의 분포를 모르기 때문에) piecewise linear unit의 이점을 가져가기 어려웠다

VAE



# Vanilla GAN

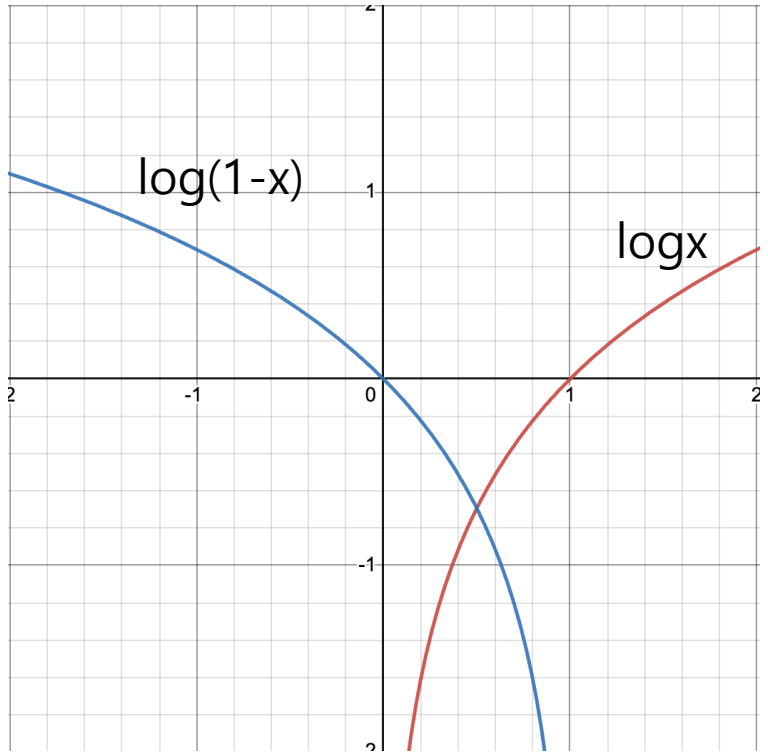
- 하지만 GAN은 사전 분포와 같은 제약이 없기 때문에 intractable problem이 없다.



# GAN notation

x ~ Pdata(x) : real data의 분포 Pdata(x)에서 sampling한 x
z ~ Pz(z) : noise data의 분포 Pz(z)에서 sampling한 z

Discriminator는 real image면 1로, Generator가 만들어낸 Fake Image면 0으로 판별.

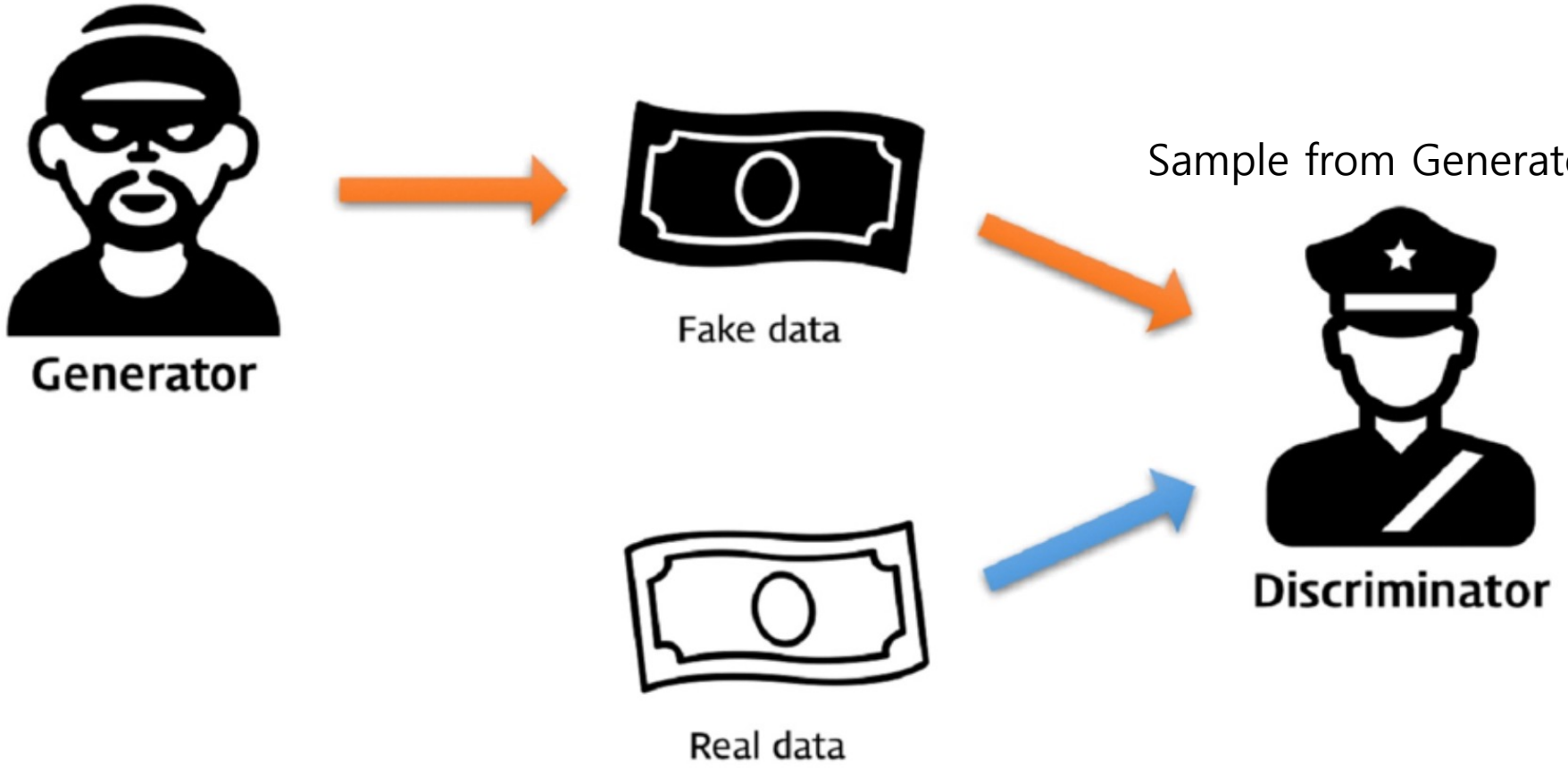



Gan Loss

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

# GAN

Capture Data Distribution



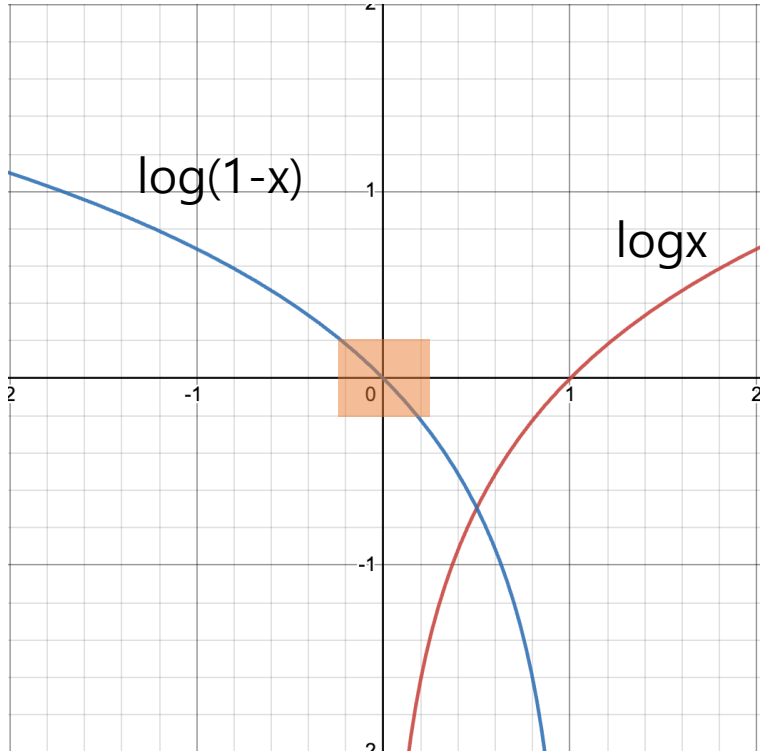
# GAN

But 초기의 **Generator**는 진짜 같은 이미지를 당연히 못 만들 것이고 **Discriminator**는 높은 정확도로 진짜와 가짜를 구분할 것이다.

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

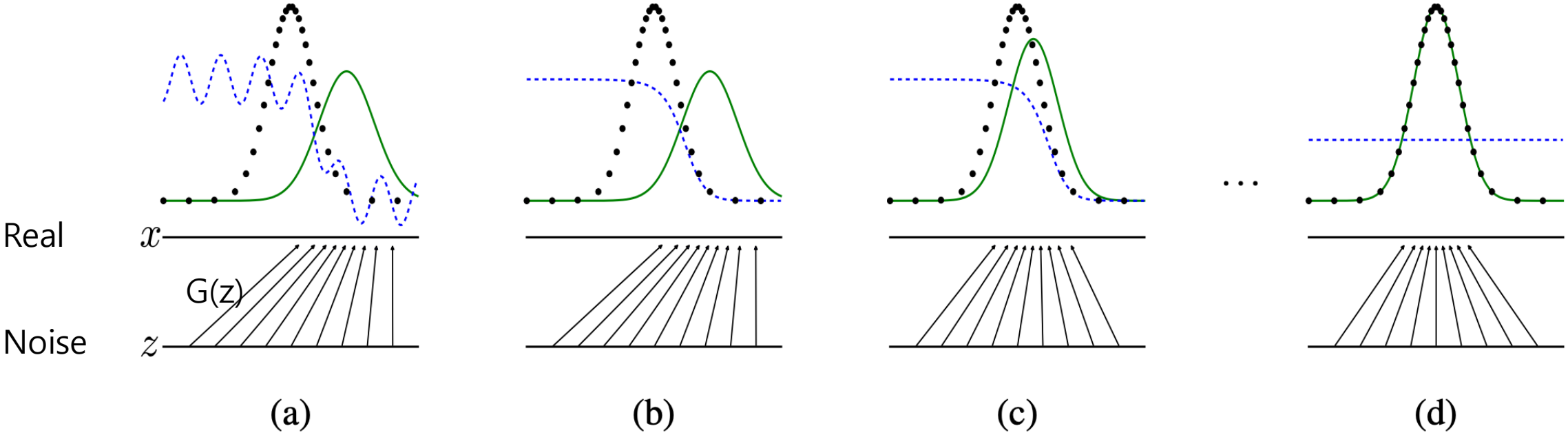
→ Gradient가 너무 작아지게 된다.
→ log(1-x)를 minimize한다 = x가 1을 내뱉으면 된다.
→ D(G(z))가 1을 내뱉도록 학습하면 된다.

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



# GAN

Blue: D
Green: G
Black: real




Real data의 분포는 특정 영역에 몰려있는 non-uniform 분포이다

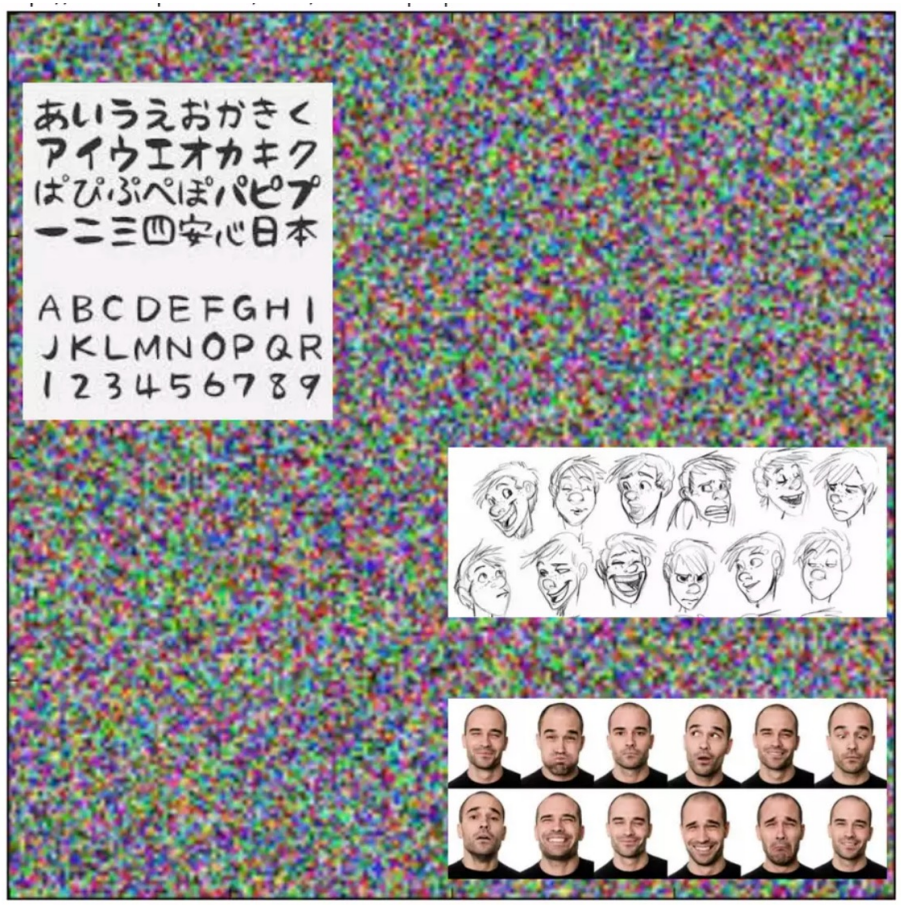
# Why non-uniform?

200x200 RGB 채널에서 random하게 값을 뽑아서 plot하면 오른쪽의 noise이미지만 계속 나온다.

얼굴이나 스케치, 글씨 같은 이미지도 나올 수 있으나 안나온다는 것은?

어떠한 특정한 공간에 데이터가 밀집해있다.

→ 그 공간을 잘 포함하는 저차원의 공간이 있다.



# Algorithm

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

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

  **end for**

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

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

**end for**

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

1. Generator고정, Discriminator 학습

2. 1번 과정을 k번 반복하여 Discriminator가 Generate된 이미지를 잘 구분하도록 함

3. Discriminator고정, Generator 학습

4. 3번과정을 통해 고정된 Discriminator를 속일 수 있는 Generator를 학습

# GAN 모델은 완전한가?

- 모델이 tractable한가?
- 모델이 Global optimum을 갖는가?
- 모델이 Global optimum에 수렴할 수 있는가?
- 모델의 Global optimum을 찾을 수 있는 알고리즘이 존재하는가?

# 모델이 Global optimum을 갖는가?

1. 직관적으로 Real data의 분포와 Generator의 분포가 같을 때 global optimum을 갖는다 $P_{data} = P_g$

2. 그 때 Discriminator는 자신에게 들어온 input이 진짜인지 가짜인지 알 수 없기 때문에 0.5의 확률을 갖는다.

Ex) Discriminator의 sigmoid를 통과한 값이 1이면 real image, 0이면 fake image로 판별하지만 두개를 구분할 수 없어서 0.5의 값을 뱉게된다.

# Algorithm

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

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

  **end for**

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

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

**end for**

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

1. Generator고정, Discriminator 학습

2. 1번 과정을 k번 반복하여 Discriminator가 Generate된 이미지를 잘 구분하도록 함

3. Discriminator고정, Generator 학습

4. 3번과정을 통해 고정된 Discriminator를 속일 수 있는 Generator를 학습

# 모델이 Global optimum을 갖는가?-증명

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

G를 fix 해놓고 위의 식을 바꾸면

$$V(G,D) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_z p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$

$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx$$

← P(z)에서 z를 sampling해서 Generator에 넣으면 real data x와 같은 차원이 된다.

이것은 Pg분포에서 x를 샘플링 한 것과 같다.

**결국 위의 식을 maximize 해야한다!**

# 모델이 Global optimum을 갖는가?-증명

$$V(G,D) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$

$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx$$

$$D^*(x) = \arg\max_{D} V(D) = \int_{x} p_{data}(x) log D(x) dx + p_g(x) log(1 - D(x)) dx$$

적분 식을 최대화 하는건 적분 안의 식을 최대화 하는 것과 같다.

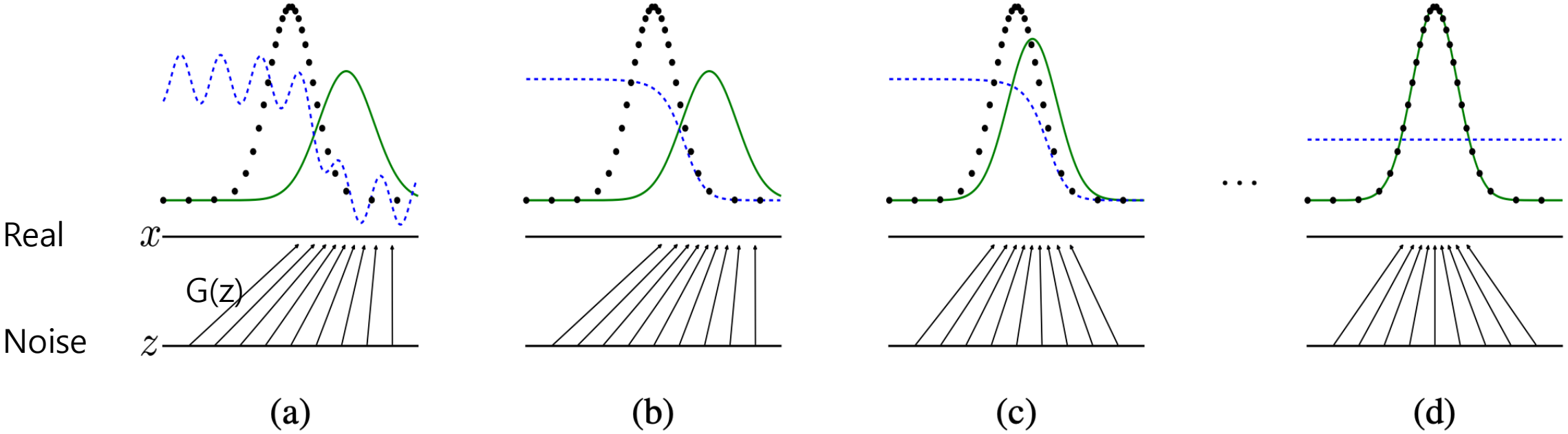
Pdata(x)를 a로, D(x)를 y로, Pg(x)를 b로 치환하여 적분 안의 식을 y에 대해서 미분하면 Y = a/(a+b)

Generator가 고정되어있는 상태에서 최적의 Discriminator는 오른쪽과 같은 확률을 갖는다.

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

# 모델이 Global optimum을 갖는가?-증명

Blue: D
Green: G
Black: real




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

# 모델이 Global optimum을 갖는가?-증명

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

$$\int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx$$

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

KL Divergence : 두 분포 사이의 유사도
대칭이 아니다

JS Divergence: 두 분포 사이의 거리
대칭이며 거리이기 때문에 0보다
같거나 크다

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

$$= \int_x p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} dx + \int_x p_g(x) \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} dx$$

$$= -log4 + log4 + \int_x p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} dx + \int_x p_g(x) \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} dx$$

$$= -log4 + \int_x p_{data}(x) \log \frac{2 \cdot p_{data}(x)}{p_{data}(x) + p_g(x)} dx + \int_x p_g(x) \log \frac{2 \cdot p_g(x)}{p_{data}(x) + p_g(x)} dx$$

$$= -log4 + KL(p_{data} || \frac{p_{data} + p_g}{2}) + KL(p_g || \frac{p_{data} + p_g}{2})$$

$$= -log4 + 2 \cdot JSD(p_{data} || p_g)$$

V(D,G)가 G의 입장에서 minimize되기 위해서는 Pdata = Pg여야 한다.

# 모델이 Global optimum을 갖는가?-증명

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

Discriminator

$$p_g = p_{\text{data}}$$

Generator

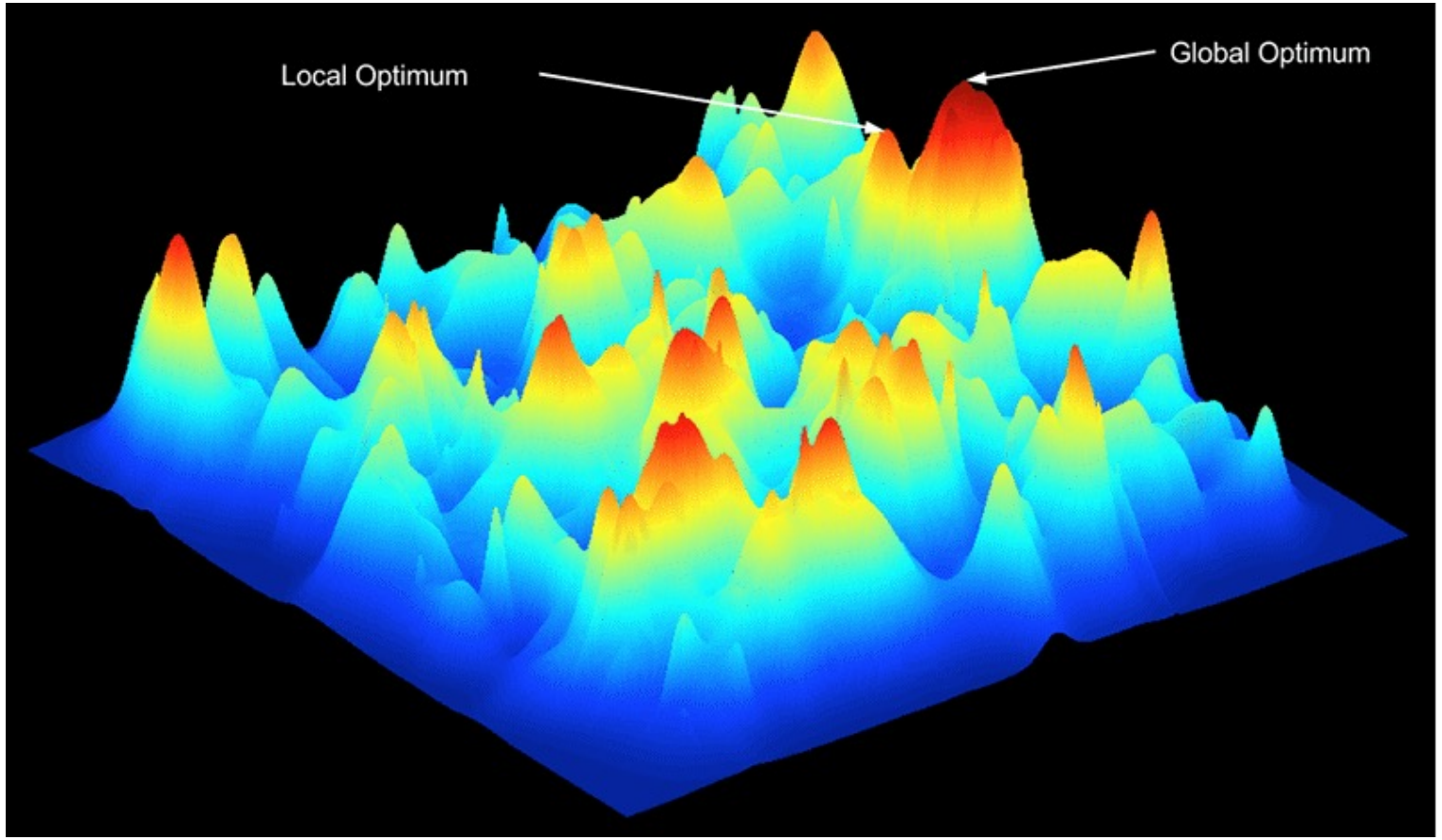
위 두식을 이용하면 최적의 Discriminator의 값은 ½ = 0.5

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

Discriminator의 값이 0.5면 위 식은 log1/2 + log1/2 = -log4

결국 Pg=Pdata일 때 Global optimum을 갖는다!

# Global optimum에 수렴할 수 있나?



# Global optimum에 수렴할 수 있나?

즉, Pg가 Pdata와 같아질 수 있나?

$$V(G, D) = U(p_g, D)$$

Discriminator가 최적의 값을 가지고 있어서 고정을 시켜놓는다.

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

해당 식을 Pg에 대해서 미분하면?
Log(1-D(x)) → D(x)는 고정된 상수이기 때문에
Pg에 대해 미분한 값이 상수이다.
→ Linear한 함수다 ex) 2x-1을 미분하면 2
→ Convex하다.
→ 따라서 Iterative method(GD등등)으로 Global optimum에 수렴할 수 있다.

# GAN의 단점 – 수학적 허점

앞에서 결국 Generator가 Pdata로 수렴하는건 증명이 됐으나,

우리는 실제로 Pg의 분포를 추정하고 최적화 하지 않고 $G(z;\theta g)$

즉, Pg분포를 만드는데 들어가는 파라미터 $\theta g$를 추정하고 최적화 한다.
Ex) Generator의 weight & bias

Generator를 MLP와 같은 method를 사용하게되면 Loss fucntion이
Convex하지 않을 수도 있다.

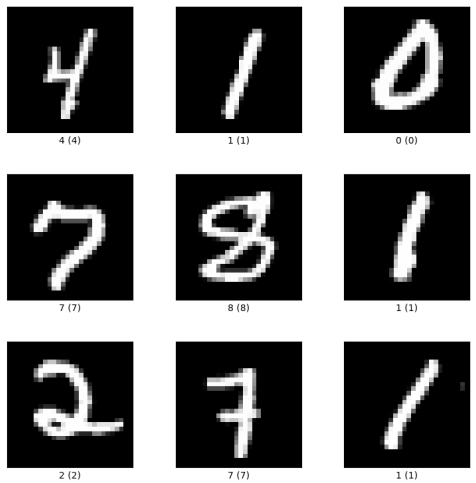
하지만 그래도 수학적 보장 없이 실험적으로 잘 작동하고 있다.

# GAN의 단점 – Mode Collapse

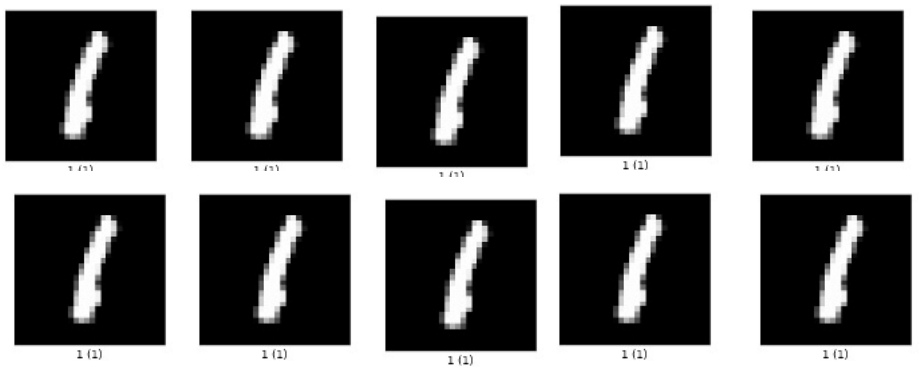
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

위 식을 보게되면 학습시 Generator가 생성하는 이미지의 종류는 고려하지 않는다.

따라서 Generator는 자신이 잘 생성할 수 있는 한가지만 생성을 하고
Discriminator는 단순히 그에 대해 판별을 하더라도 문제 없이 global optimum으로 수렴한다.

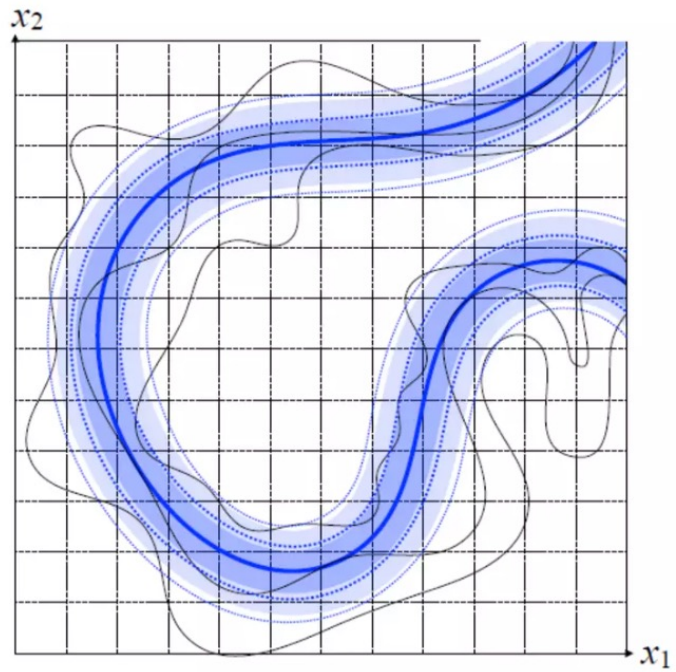


Model

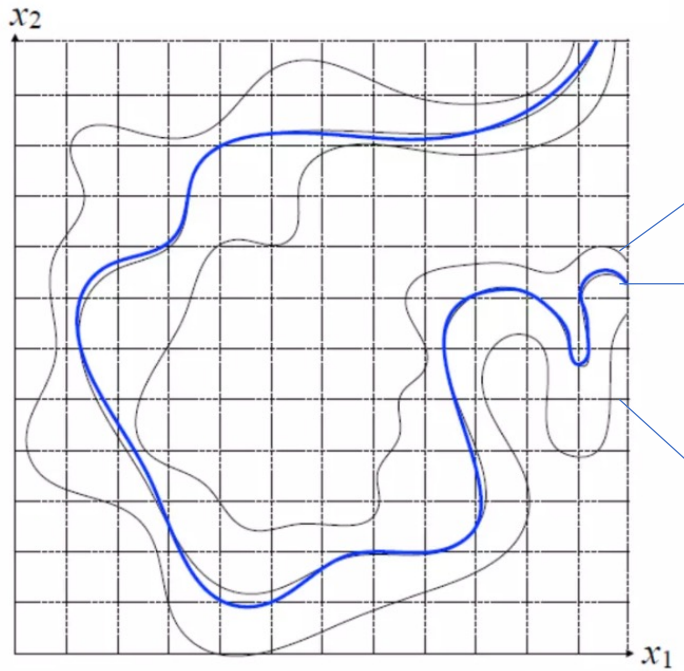


# GAN의 단점 – Mode Collapse

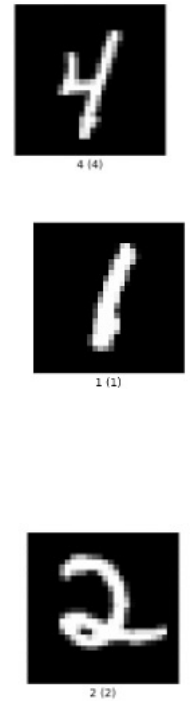
**Comparison between VAE vs GAN**




**VAE : maximum likelihood approach**




**GAN**



# Implementation

```python
import torch
import torch.nn as nn


D = nn.Sequential(
    nn.Linear(784, 128),
    nn.ReLU(),
    nn.Linear(128, 1),
    nn.Sigmoid())

G = nn.Sequential(
    nn.Linear(100, 128),
    nn.ReLU(),
    nn.Linear(128, 784),
    nn.Tanh())

criterion = nn.BCELoss()

d_optimizer = torch.optim.Adam(D.parameters(), lr=0.01)
g_optimizer = torch.optim.Adam(G.parameters(), lr=0.01)

# Assume x be real images of shape (batch_size, 784)
# Assume z be random noise of shape (batch_size, 100)

while True:
    # train D
    loss = criterion(D(x), 1) + criterion(D(G(z)), 0)
    loss.backward()
    d_optimizer.step()

    # train G
    loss = criterion(D(G(z)), 1)
    loss.backward()
    g_optimizer.step()
```

# Q&A

# Reference


- https://www.slideshare.net/NaverEngineering/ss-96581209
- https://89douner.tistory.com/331
- https://www.be-terna.com/nb/innsikt/optimal-planning-production-capacity