AI-504

Generative Adversarial Network (GAN)

Week 03

인하대학교 통계학과 김현수

스터디 목표 : AI504 + Project

1강 : Introduction
2강 : Numpy
3~4강 : Sklearn + Practice => 1주차

5~6강 : Pytorch - Logreg, NN + Practice
7~8강 : AutoEncoder+Practice => 2주차

9~10강 : Variational AutoEncoder + Practice
11~12강 : GAN + Practice => 3주차

13~14강 : CNN + Practice
15~16강 : Word Embedding + Practice => 4주차

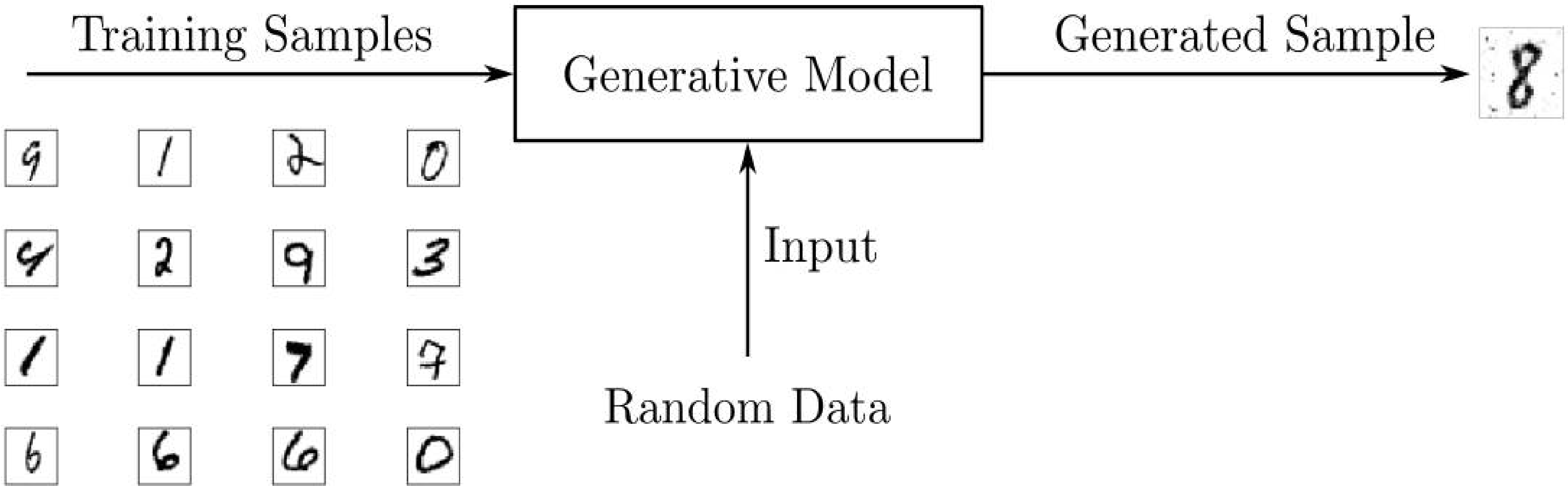
17~18강 : RNN + Practice
19~20강 : Img2Txt + Practice => 5주차

21~22강 : Transformer + Practice
23~24강 : BERT & GPT + Practice => 6주차

25~26강 : Graph NN + Practice
27~28강 : Neural ODE + Practice => 7주차

이후 AI Hub에 있는 데이터로 자율 프로젝트 진행 => 8~9주차

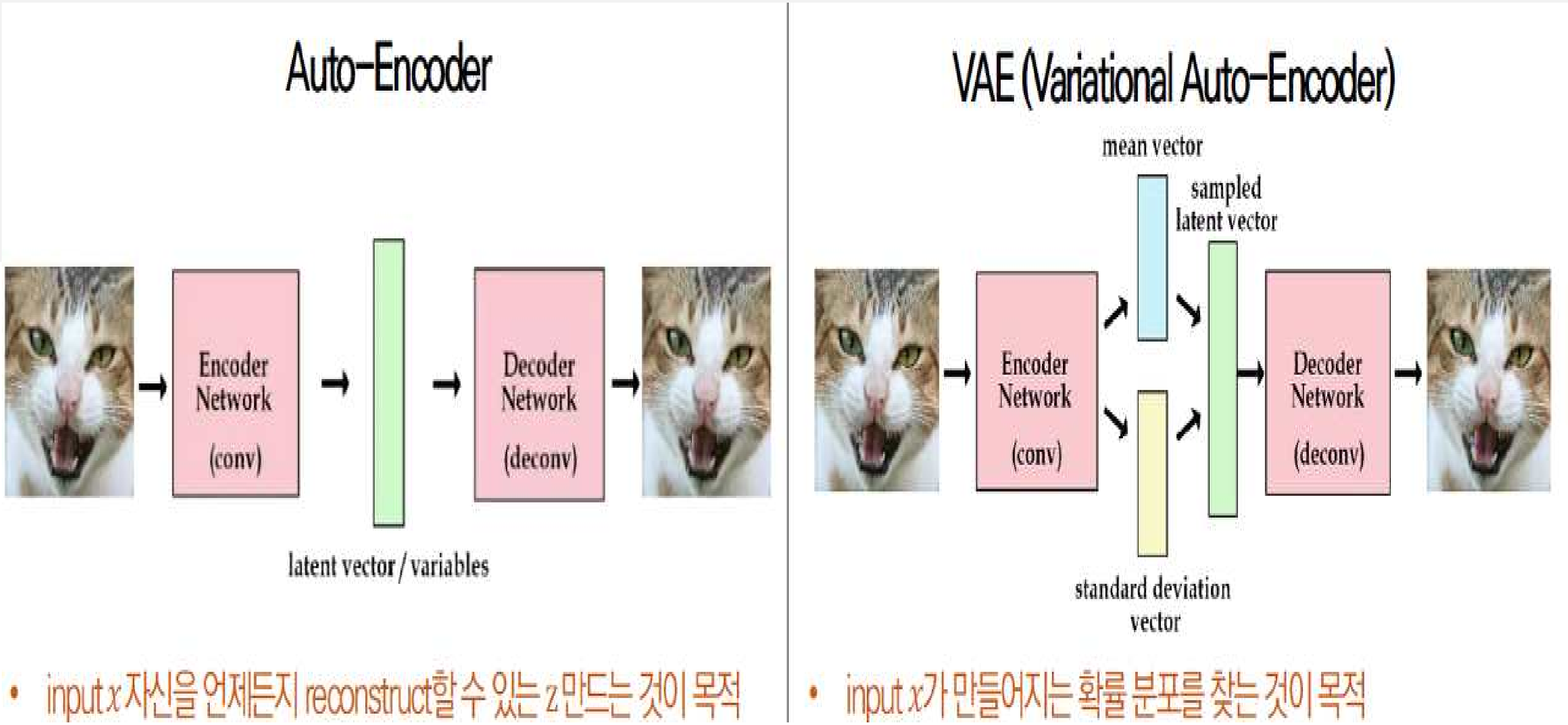
Contents



Generative Model : training data가 주어졌을 때 이 training data가 가지는 real 분포와 같은 분포에서 sampling된 값으로 new data를 생성하는 model

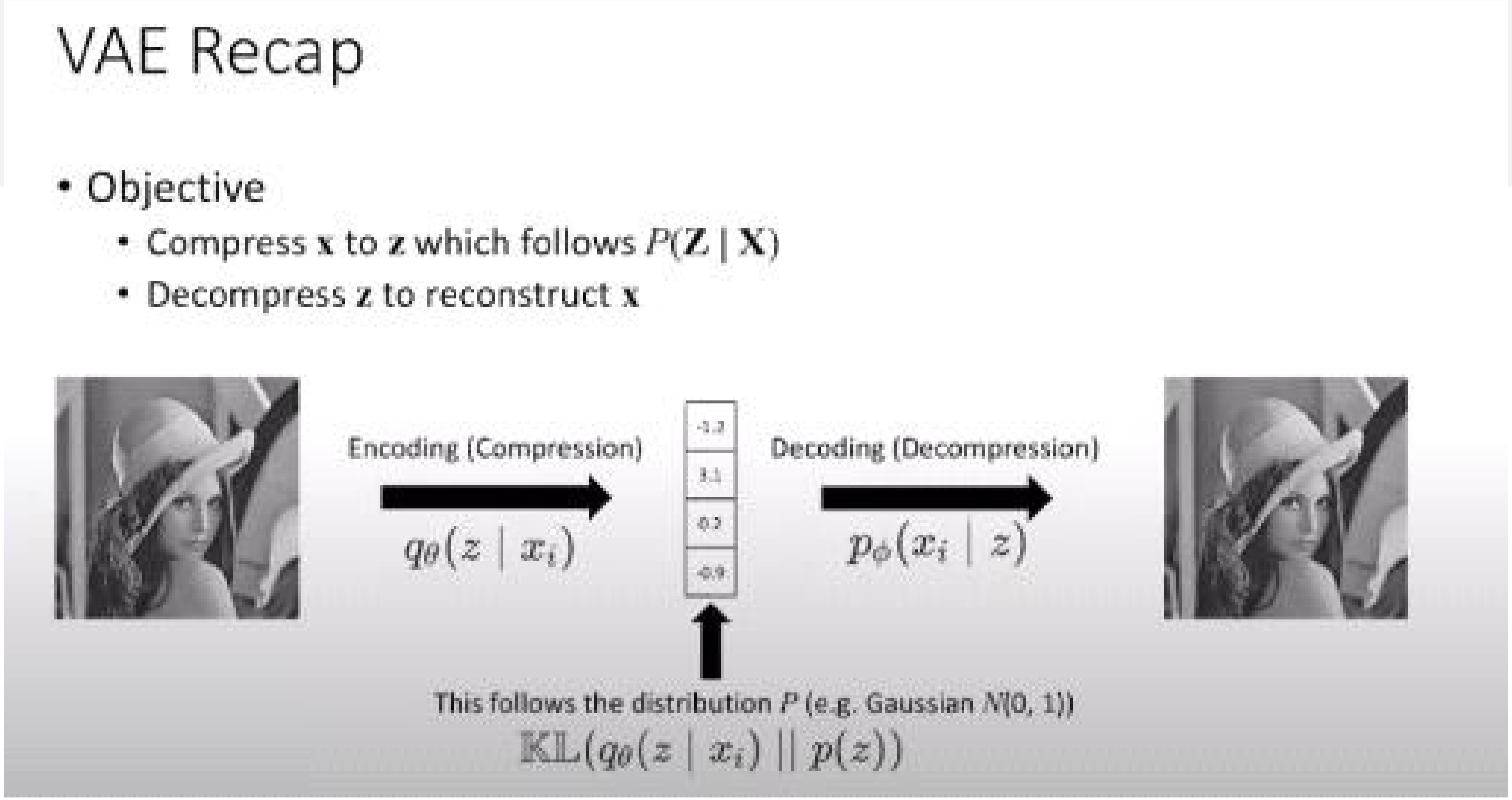
AE vs VAE

AE?
VAE?



Variation Auto Encoder

VAE



KLD (쿨백-라이블러 발산)

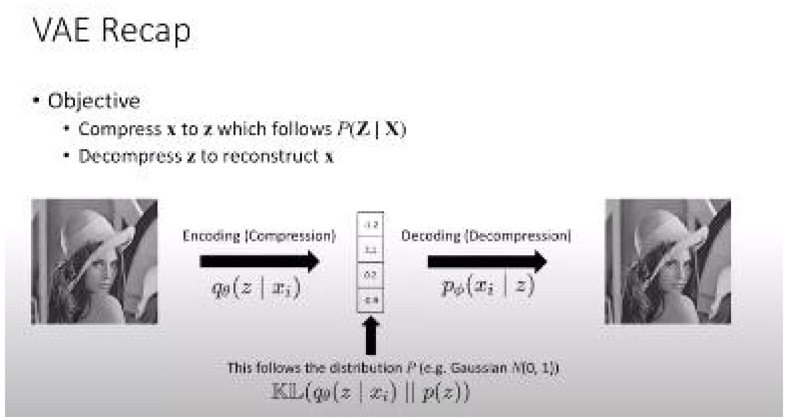
(kullback-leibler divergence)

=> P분포와 Q분포가 얼마나 다른지를 측정

=> 값이 낮을수록 두 분포가 유사하다고 해석

=> 항상 0 이상

* 계산 순서가 바뀌면 값이 변함 (거리 개념x)



ELBO?
Entropy?
trade-off?

$D_{KL}(q(z)||p(z|x)) = E_q[\log q(z)] - E_q[\log p(z|x)]$ (기본적인 KLD)

$= E_q[\log q(z)] - E_q[\log \frac{p(x|z)p(z)}{p(x)}]$

$= E_q[\log q(z)] - E_q[\log \frac{p(x,z)}{p(x)}]$

$= E_q[\log q(z)] - E_q[\log p(x,z) - \log p(x)]$

$= E_q[\log q(z)] - E_q[\log p(x,z)] + \log p(x)$ ($\int q(x) = 1$ 이므로)

$$\log p(x) = D_{KL}(q(z)||p(z|x)) + E_q[\log p(x,z)] - E_q[\log q(z)]$$

결국 p(x)는 constant로 변하지 않을 때, ELBO이 증가하면 D_{KL}은 감소하므로 EBLO를 최대화 함으로써 D_{KL}을 최소화 하여 q(z)와 p(z|x)의 차이를 최소화 할 수 있다.

-> ELBO
(Evidence of Lower BOund)

Generative Adversarial Network

GAN

- Two person game
  - Game between Generator (G) and Discriminator (D)
  - Think of game theory
- Generator (G)
  - Tries to fool D with fake samples x’
- Discriminator (D)
  - Tries to discriminate between real samples x and fake samples x’

Maximum Likelihood Estimate

- Need to estimate θ.
  - Learn from data ➔ Training pairs $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_N, y_N)$
- Log Likelihood Function

Probability Function

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}} = \Pr(Y = 1 \mid X; \theta)$$

Likelihood Function

$$\begin{aligned} L(\theta \mid x) &= \Pr(Y \mid X; \theta) \\ &= \prod_i \Pr(y_i \mid x_i; \theta) \\ &= \prod_i \boxed{h_\theta(x_i)^{y_i} (1 - h_\theta(x_i))^{(1-y_i)}} \end{aligned}$$

Y is a binary variable following the Bernoulli Distribution

Log Likelihood Function

$$N^{-1} \log L(\theta \mid x) = N^{-1} \sum_{i=1}^{N} \log \Pr(y_i \mid x_i; \theta)$$

Negative Log Likelihood Function

$$-\frac{1}{N} \sum_{n=1}^{N} \left[y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right]$$

Minimize the negative log likelihood (NLL) by Gradient Descent

Generative Adversarial Network

GAN

- Remember Cross Entropy?

$$\frac{1}{N}\sum_{n=1}^{N}\Big[\underbrace{y_n \log \hat{y}_n}_{\text{Positive Samples}} + \underbrace{(1-y_n)\log(1-\hat{y}_n)}_{\text{Negative Samples}}\Big]$$

$$= \frac{1}{M}\sum_{m=1}^{M}[\log \hat{y}_m] + \frac{1}{N}\sum_{n=1}^{N}[\log(1-\hat{y}_n)]$$

Separate positive samples and negative samples.

$$= \mathbb{E}_{x\sim p_{positive}} \log D(x) + \mathbb{E}_{x'\sim p_{negative}} \log\big(1-D(x')\big)$$

Sample mean is an unbiased estimator of population mean.

$$\mathbb{E}_{x\sim p_{positive}} \log D(x) + \mathbb{E}_{x'\sim p_{negative}} \log\big(1-D(x')\big)$$

$$= \mathbb{E}_{x\sim p_{real}} \log D(x) + \mathbb{E}_{x'\sim p_{fake}} \log\big(1-D(x')\big)$$

Positive VS Negative → Real VS Fake

$$= \mathbb{E}_{x\sim p_{real}} \log D(x) + \mathbb{E}_{z\sim p_z} \log\Big(1-D\big(G(z)\big)\Big)$$

Fake samples come from random noise p(z) and Generator G.

$$= \mathbb{E}_{x\sim p_{real}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p_z} \log\Big(1-D_{\theta_d}\big(G_{\theta_g}(z)\big)\Big)$$

Use parameterized functions for D and G.

Logistic Function

GAN

- 생성자는 minimize (D(x)를 1로)
- 구분자는 maximize (D(G(z)를 0으로)

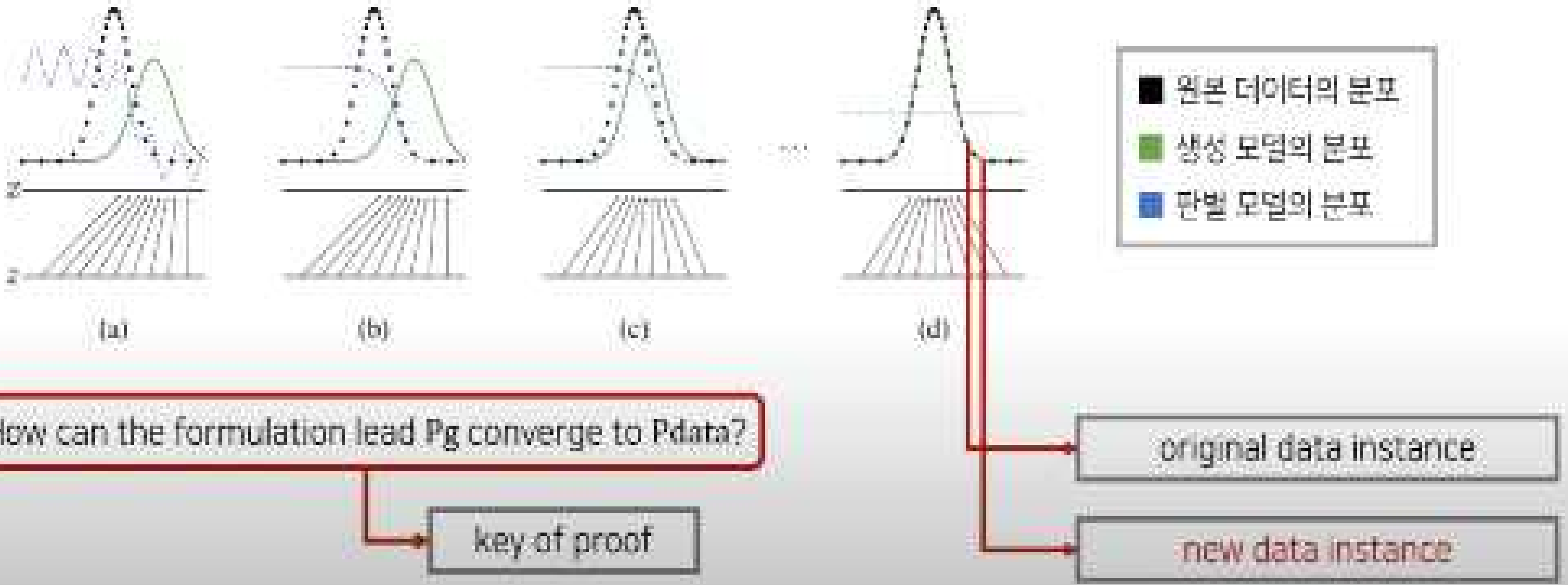
- GAN is a two-person MinMax game between D & G

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{real}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p_z} \log \left(1 - D_{\theta_d} \left(G_{\theta_g}(z) \right) \right) \right]$$

- Discriminator (θ_d) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)

GAN의 수렴 과정

- 공식의 목표(Goal of Formulation)
  - $Pg \rightarrow Pdata, D(G(z)) \rightarrow 1/2$ $(G(z)$ *is not distinguishable by* $D)$



- How can the formulation lead Pg converge to Pdata?

key of proof

감사합니다

Thank You