15기 정규세션

ToBig's 14기 김상현

Unsupervised Learning & Generative model

비지도 학습 & 생성 모델

Contents

Unit 01 | Unsupervised Learning

✓ Supervised VS Unsupervised

Supervised Learning

Data: (x,y) 한쌍
y = f(x)의 f를 학습

Unsupervised Learning

Data: 오직 x만
데이터의 기본 구조를 학습

✓ Supervised Learning Example

Supervised Learning

Data: (x,y) 한쌍
y = f(x)의 f를 학습

Classification

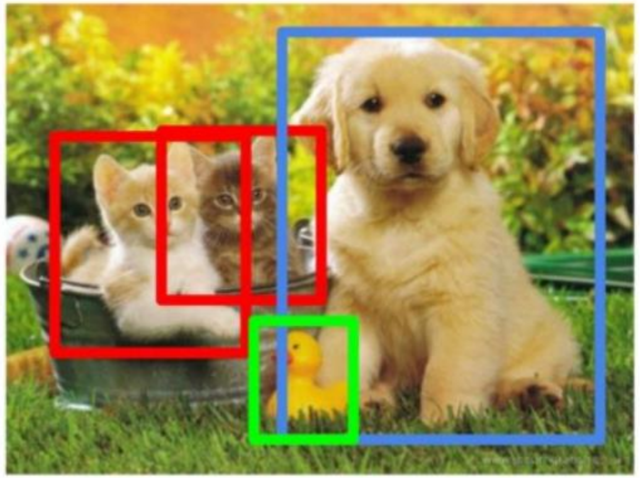
CAT

✓ Supervised Learning Example

Supervised Learning

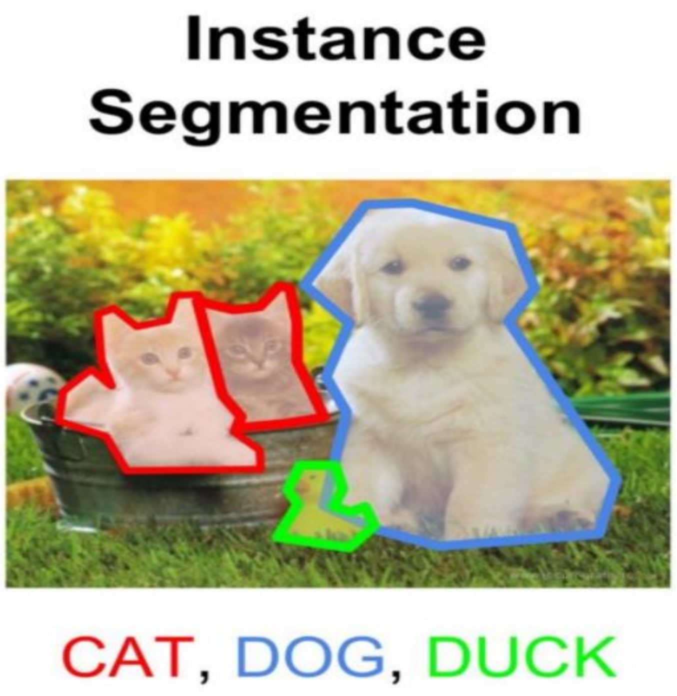
Data: (x,y) 한쌍
y = f(x)의 f를 학습

Object Detection



CAT, DOG, DUCK

✓ Supervised Learning Example



Supervised Learning

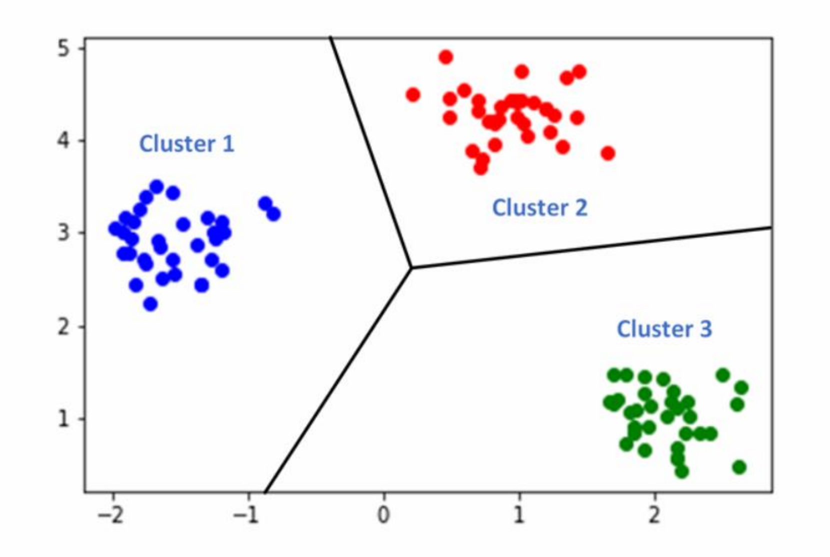
Data: (x,y) 한쌍
y = f(x)의 f를 학습

✓ **Unsupervised Learning Example**

Unsupervised Learning

Data: 오직 x만
데이터의 기본 구조를 학습

Clustering

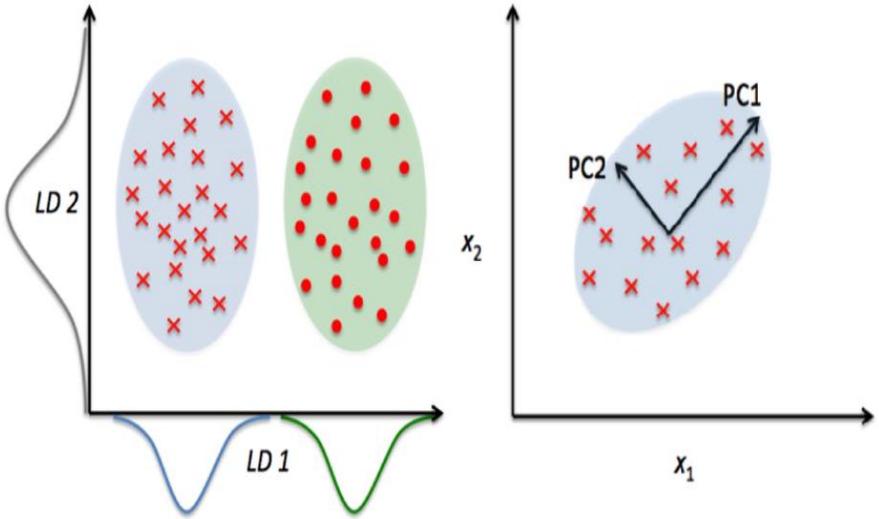


✓ **Unsupervised Learning Example**

Dimension Reduction

Unsupervised Learning

Data: 오직 x만
데이터의 기본 구조를 학습



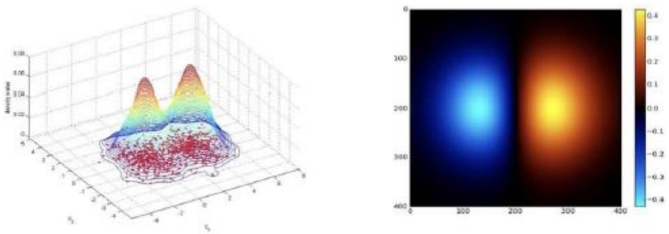
✓ Unsupervised Learning Example

Unsupervised Learning

Data: 오직 x만
데이터의 기본 구조를 학습

Density Estimation

1-d density estimation



2-d density estimation

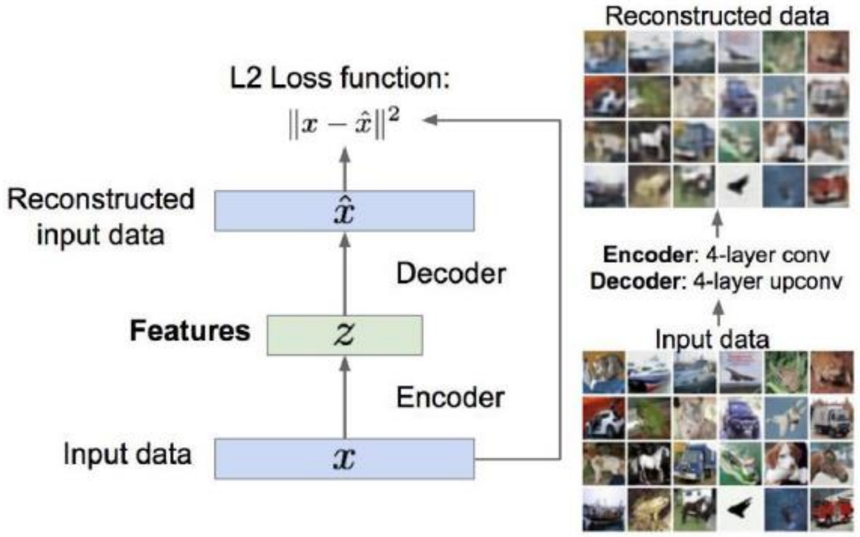
Unit 01 | Unsupervised Learning

✓ Unsupervised Learning Example

Unsupervised Learning

Data: 오직 x만
데이터의 기본 구조를 학습

Feature Learning



✓ **Autoencoder**

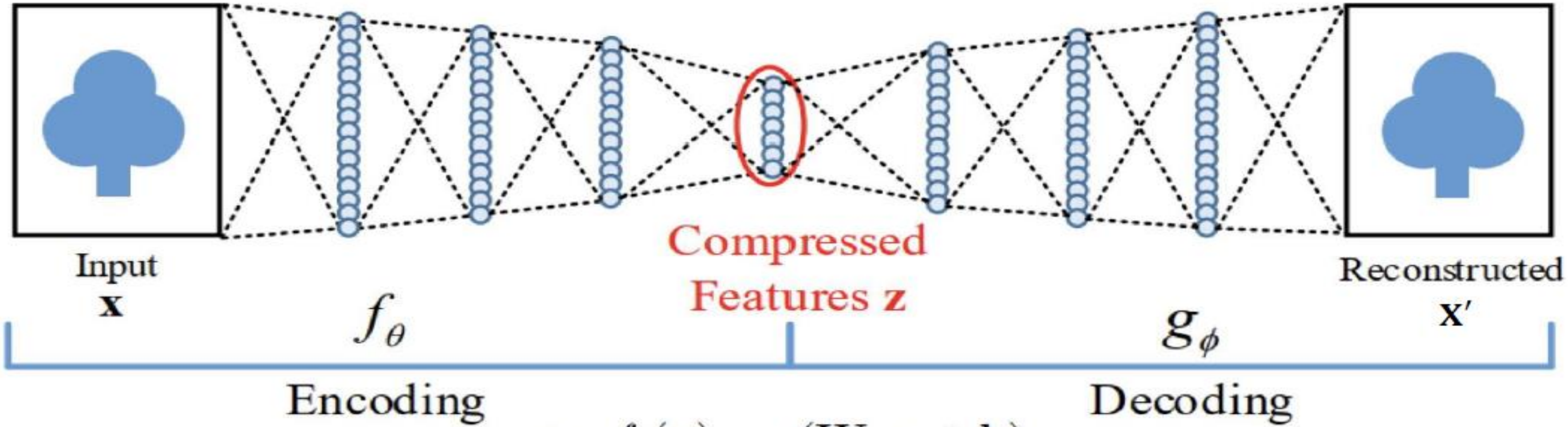
<목적>

입력 데이터를 잘 복원시키자

<KEYWORDS>

- Unsupervised Learning
- Representation Learning
- Dimension Reduction

✓ Autoencoder

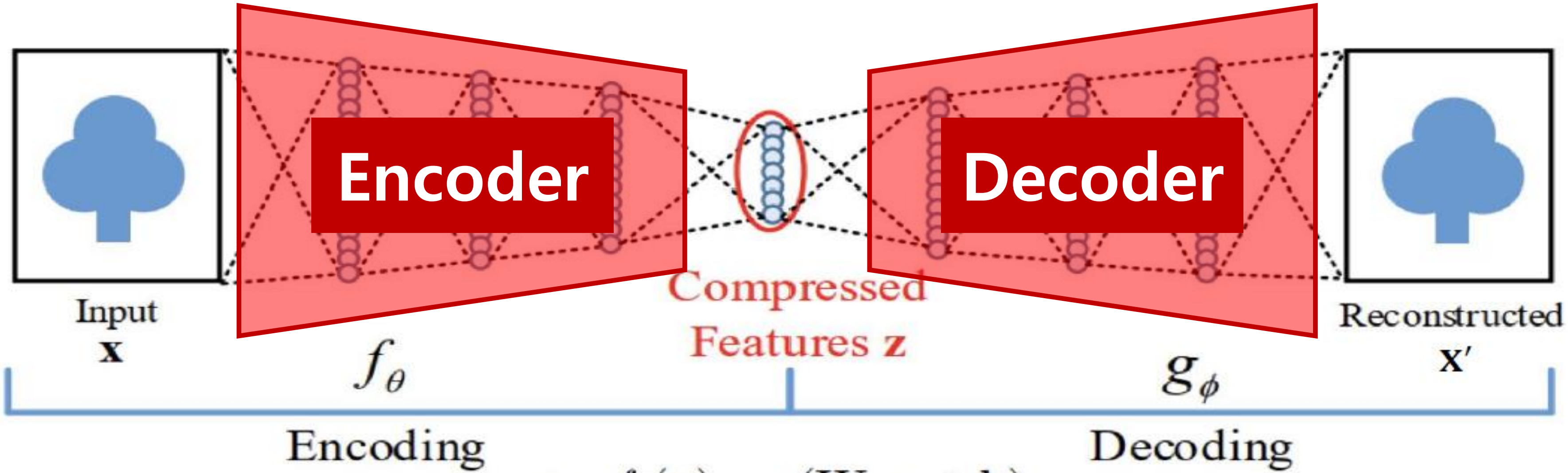


$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$$

$$\mathbf{x}' = g_\phi(\mathbf{z}) = \sigma(\mathbf{W}' \cdot \mathbf{z} + \mathbf{b}')$$

$\mathbf{z} = code/latent\ feature/latent\ vector/representation/embedding$

✓ Autoencoder

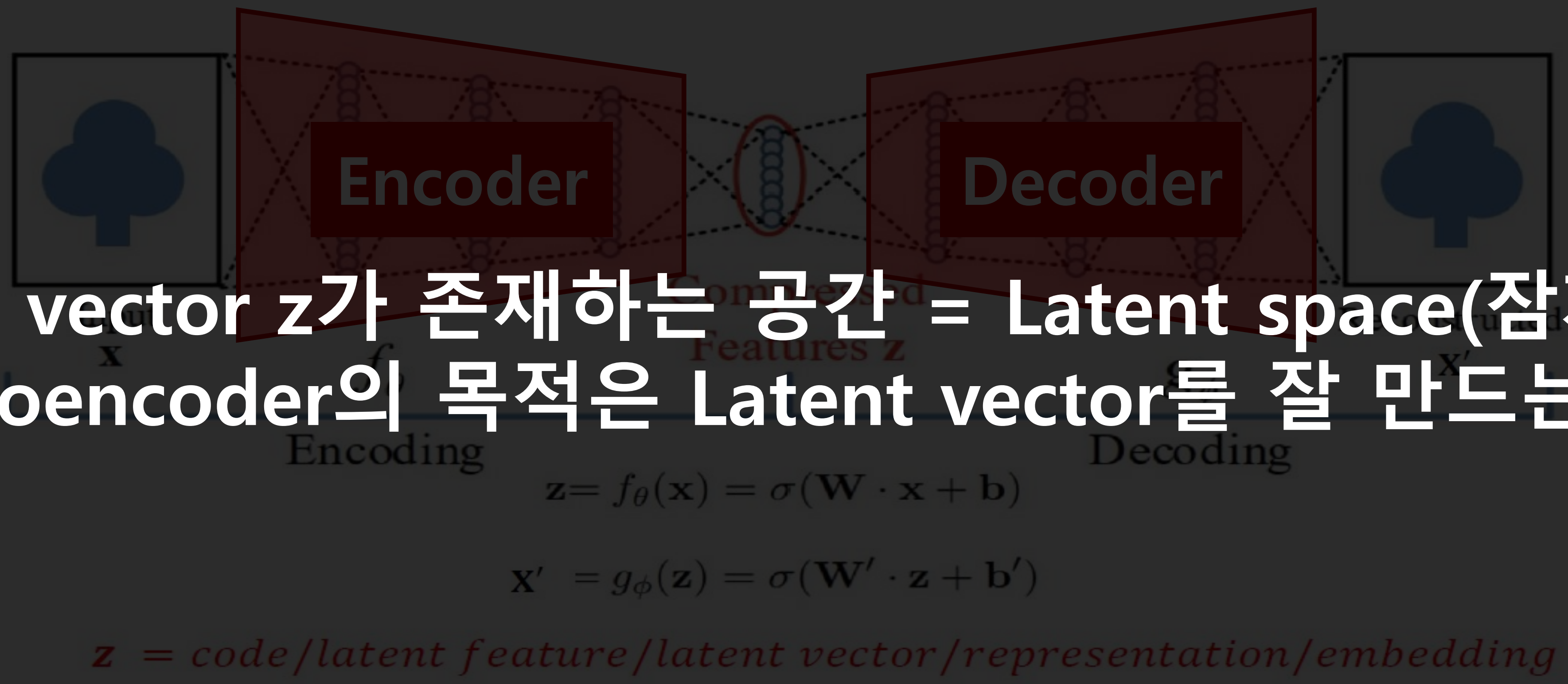


$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$$

$$\mathbf{x}' = g_\phi(\mathbf{z}) = \sigma(\mathbf{W}' \cdot \mathbf{z} + \mathbf{b}')$$

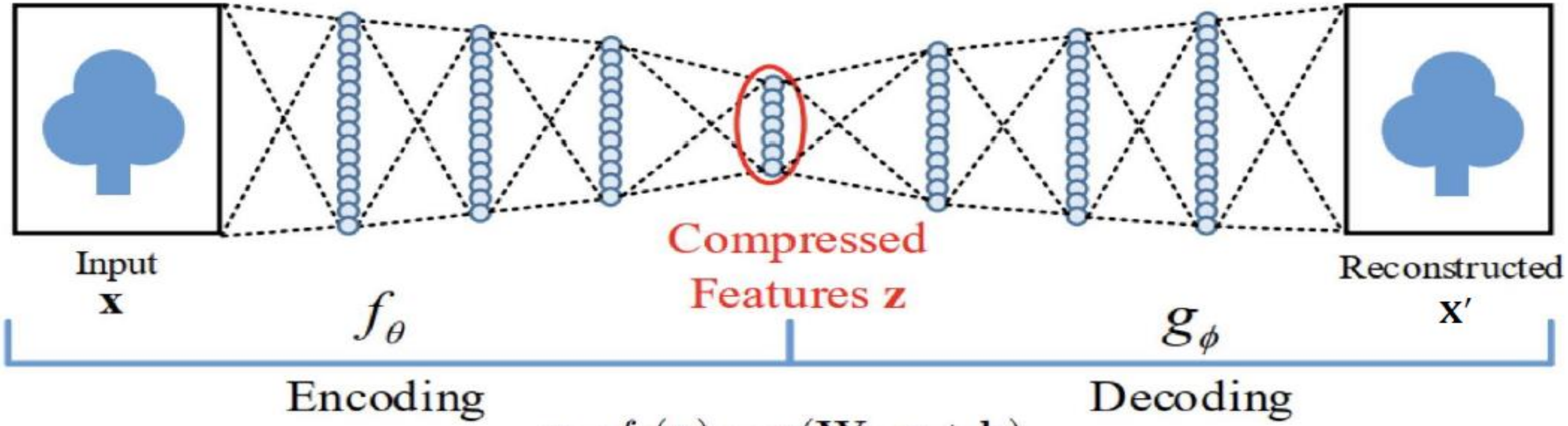
$\mathbf{z} = code/latent\ feature/latent\ vector/representation/embedding$

✓ **Autoencoder**



Latent vector z가 존재하는 공간 = Latent space(잠재공간)
Autoencoder의 목적은 Latent vector를 잘 만드는 것!

✓ Autoencoder



$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$$

$$\mathbf{X}' = g_\phi(\mathbf{z}) = \sigma(\mathbf{W}' \cdot \mathbf{z} + \mathbf{b}')$$

$$\boldsymbol{L(x, x') = L(x - x') = L\big(x - \sigma(W'\sigma(Wx + b) + b')\big)}$$

$$\boldsymbol{W, b, W', b'} = \underset{W,b,W',b'}{\operatorname{argmin}} \boldsymbol{L}$$

✓ **Autoencoder**

Linear Autoencoder

Stacking Autoencoder(SAE)

Denoising Autoencoder(DAE)

Stochastic Contractive Autoencoder(SCAE)

Contractive Autoencoder(CAE)

etc

Contents

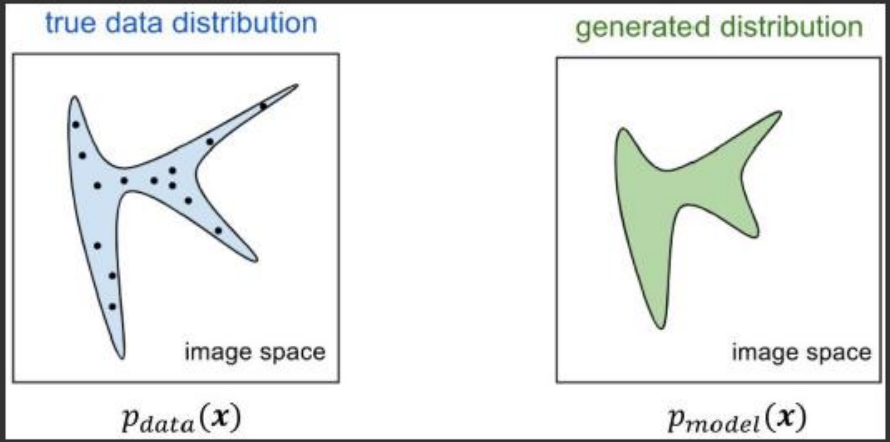
✓ **Generative Modeling?**

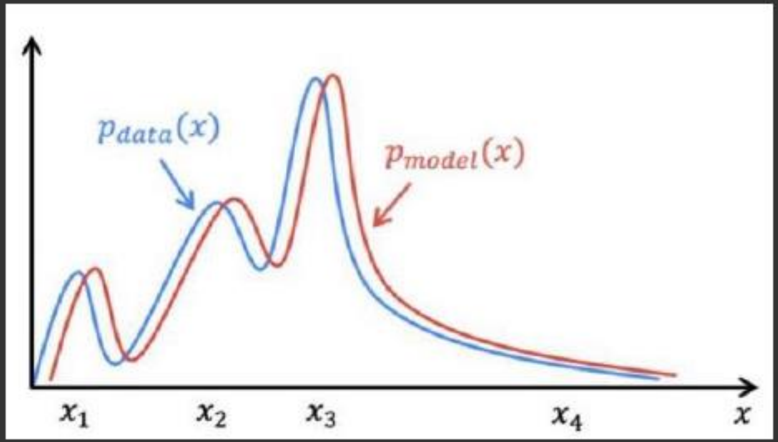
Generative Modeling vs Discriminative Modeling

Generative Modeling	Discriminative Modeling
생성 모델링	판별 모델링
Sample x의 p(x)를 추정	Sample x가 주어졌을 때
Label있는 경우는 p(x|y) 추정	label y의 확률 p(y|x)를 추정

✓ Generative Modeling?

Want to learn $P_{model}(x)$ similar to $P_{data}(x)$





Unit 02 | Generative Model

✓ **Generative Modeling?**

1. Sample dataset X를 가지고 있다.
2. Sample이 알려지지 않은 어떤 p_{data}분포로 생성되었다고 가정한다.
3. 생성모델에서 p_{model} 이 p_{data}를 흉내 내려고 한다. 이 목표를 달성하면 p_{model} 에서 sampling 하여 p_{data}에서 뽑은 것 같은 sample을 생성할 수 있다.
4. p_{model}이 다음과 같다면 good!
   1. p_{data}에서 뽑은 것 같은 sample을 생성할 수 있다.
   2. X에 있는 sample과 다른 sample을 생성할 수 있다.

✓ **Generative Model의 종류**

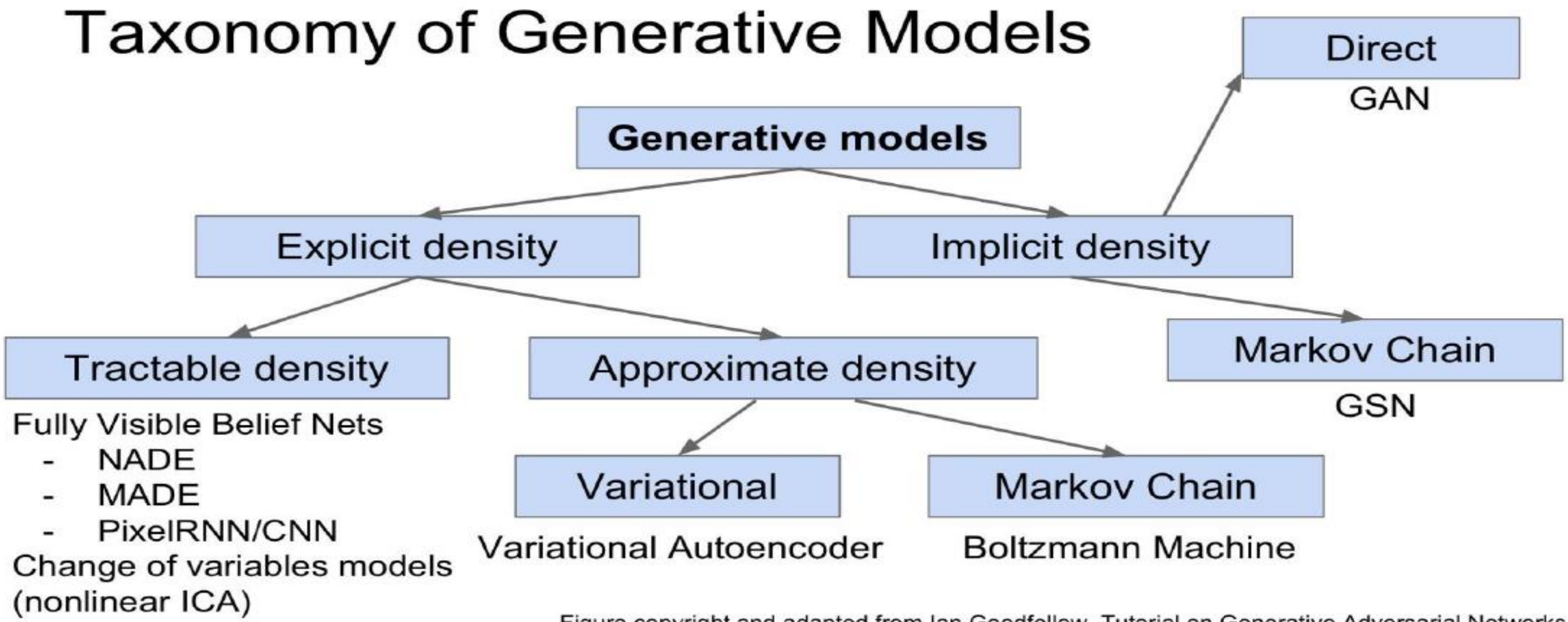


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

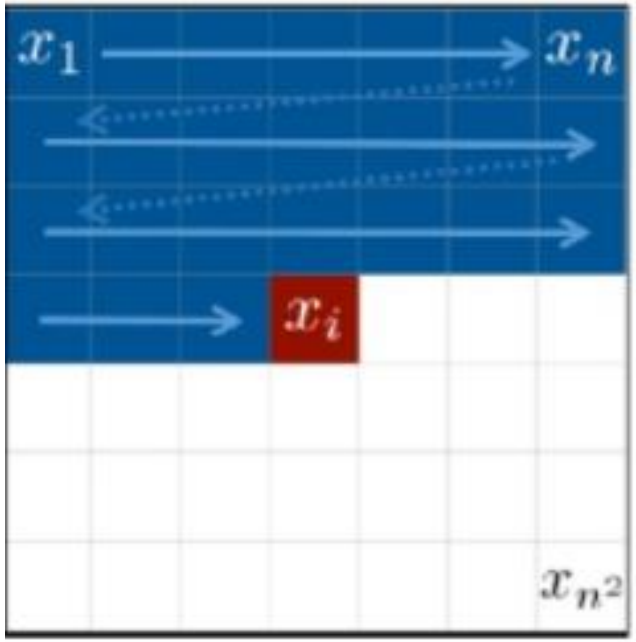
✓ Autoregressive model

Chain rule을 사용해 image x의 likelihood를 1차원 분포의 곱으로 변형
=> 학습데이터로 MLE

$$\underline{p(x)} = \prod_{i=1}^{n^2} \underline{p(x_i | x_1, \dots, x_{i-1})}$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels



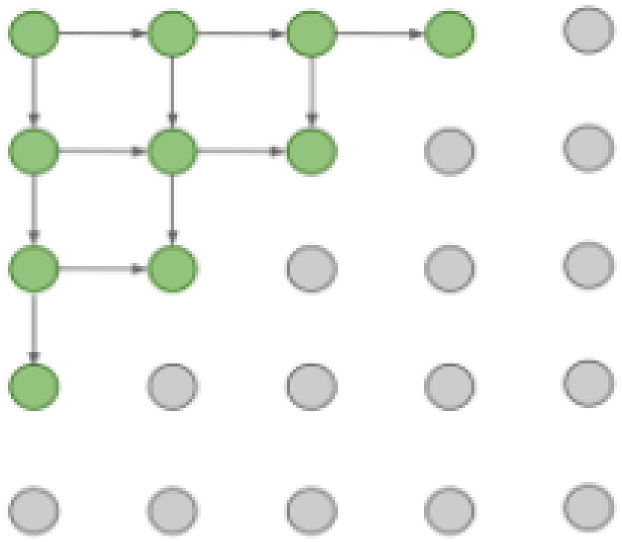
http://jaejunyoo.blogspot.com/2018/02/pr12-video-24-pixel-recurrent-neural.html

✓ Autoregressive model

PixelRNN

이전 픽셀에 대한 의존성을 바탕으로 RNN(LSTM)을 이용하게 된다.
Receptive field를 늘리기 위해 Bi-LSTM을 사용할 수 있다.

속도가 굉장히 느리지만, log-likelihood가 좋다

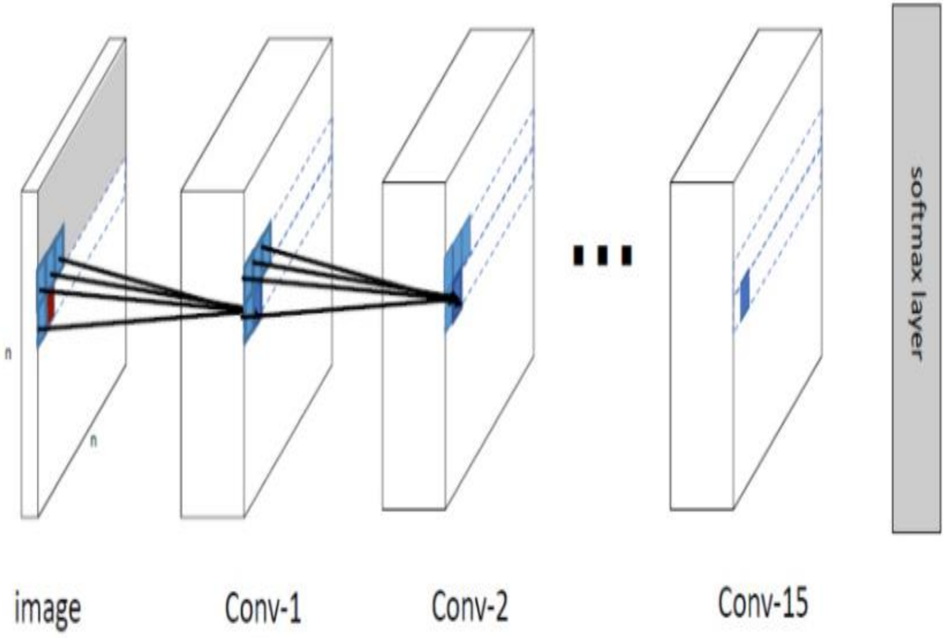


✓ Autoregressive model

PixelCNN

PixelRNN과 마찬가지로 코너에서부터 이전 픽셀에 대한 의존성을 바탕으로 특정 context 영역에 대해서만 CNN을 이용하게 된다.

PixelRNN보다 속도가 빠르지만, log-likelihood가 나쁘다



✓ Autoregressive model

장점

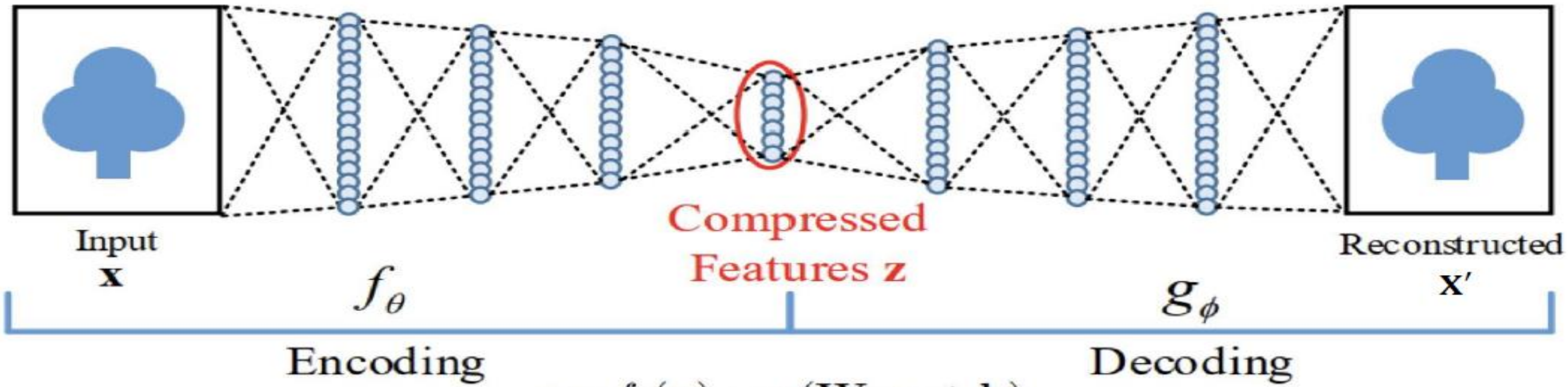
- Likelihood p(x)를 명시적으로 계산할 수 있다.
- 좋은 sample을 만들어 낼 수 있다.

단점

순차적인 생성
=>느리다

✓ **Variational Autoencoder**

Autoencoder Architecture



$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W}\cdot\mathbf{x}+\mathbf{b})$$

$$\mathbf{x}' = g_\phi(\mathbf{z}) = \sigma(\mathbf{W}'\cdot\mathbf{z}+\mathbf{b}')$$

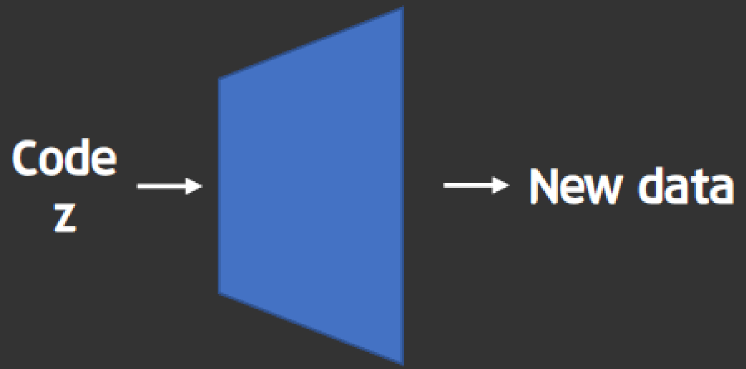
$\mathbf{z}$ $= code/latent\ feature/latent\ vector/representation/embedding$

✓ **Variational Autoencoder**

Autoencoder => z 찾기

Encoder를 사용하기 위해 Decoder도 학습

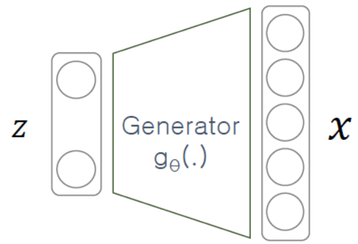
✓ Variational Autoencoder



Variational Autoencoder => 생성모델

Decoder를 위해 Encoder도 학습

✓ Variational Autoencoder



$\boxed{z \sim p(z)}$	Random variable
$g_\theta(\cdot)$	Deterministic function parameterized by θ
$x = g_\theta(z)$	Random variable

→ **Gaussian distribution 가정**

생성모델링 = Want to learn $P_{model}(x)$ similar to $P_{data}(x)$

$$\boxed{\int p(x|g_\theta(z))p(z)dz = p(x)}$$

최대화 하고 싶다

HOW?

MLE할까? **NO!!!**
Variational Inference를 해야한다.

WHY?

(a) (b) (c)

Figure 3: It's hard to measure the likelihood of images under a model using only sampling. Given an image X (a), the middle sample (b) is much closer in Euclidean distance than the one on the right (c). Because pixel distance is so different from perceptual distance, a sample needs to be extremely close in pixel distance to a datapoint X before it can be considered evidence that X is likely under the model.

✓ Variational Autoencoder

VAE의 핵심!!!

Z를 정규분포에서 샘플링하는 것보다 x와 유의미하게 유사한 샘플이 나올 수 있는 확률분포 p(z|x)로 부터 샘플링 하자.

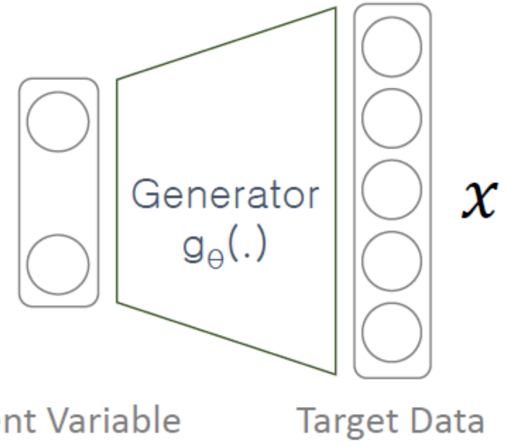
그러나 p(z|x)가 무엇인지 알지 못하므로, 우리가 알고 있는 확률분포 중 하나를 택해서 ($q_\phi(z|x)$) 그것의 파라미터값을 조정하여 p(z|x)와 유사하게 만들어 본다.(variational inference)

따라서 VAE는 true posterior와 유사한 분포를 만들게 해주는 ϕ와 생성자(generator/decoder)의 θ를 찾는 문제이다.

$$p(z|x) \approx q_\phi(z|x) \sim z$$

True posterior

Prior: p(z)



분포를 gaussian으로 가정 <= 계산을 위해서
즉, ϕ는 mean, std

✓ Variational Autoencoder

Variational Inference

true posterior P를 추론할 때, 모델 Q_{ϕ}를 가지고 추론하되 파라미터를 잘 조정해서 P에 최대한 가깝게 만드는 것

이때, 두 분포의 "가깝다 " 를 정의할 때, KL-divergence를 사용한다.

Kullback-Leibler divergence(KLD)

두 확률분포의 차이를 계산하는 데에 사용하는 함수로, 어떤 이상적인 분포에 대해, 그 분포를 근사하는 다른 분포를 사용해 샘플링을 한다면 발생할 수 있는 정보 엔트로피 차이를 계산한다.

$$D_{KL}(P||Q) = \int p(x) log \frac{p(x)}{q(x)} dx$$

Variational inference: http://jaejunyoo.blogspot.com/2017/04/auto-encoding-variational-bayes-vae-2.html
KLD:https://ko.wikipedia.org/wiki/%EC%BF%A8%EB%B0%B1-%EB%9D%BC%EC%9D%B4%EB%B8%94%EB%9F%AC_%EB%B0%9C%EC%82%B0

✓ Variational Autoencoder

parameter(ϕ, θ)찾아보자

$$\log(p(x)) = \int \log(p(x))q_\phi(z|x)dz \qquad \leftarrow \int q_\phi(z|x)dz = 1$$

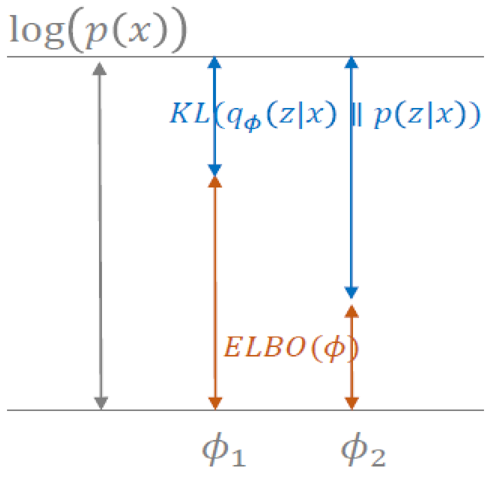
$$= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x)dz \quad \leftarrow p(x) = \frac{p(x,z)}{p(z|x)}$$

$$= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz$$

$$= \underbrace{\int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz}_{ELBO(\phi)} + \underbrace{\int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz}_{KL\left(q_\phi(z|x) \parallel p(z|x)\right)}$$

두 확률분포 간의 거리≥0

Variational inference



✓ Variational Autoencoder

parameter(ϕ, θ)찾아보자

$$\log(p(x)) = \int \log(p(x)) q_\phi(z|x)dz \quad \leftarrow \int q_\phi(z|x)dz = 1$$

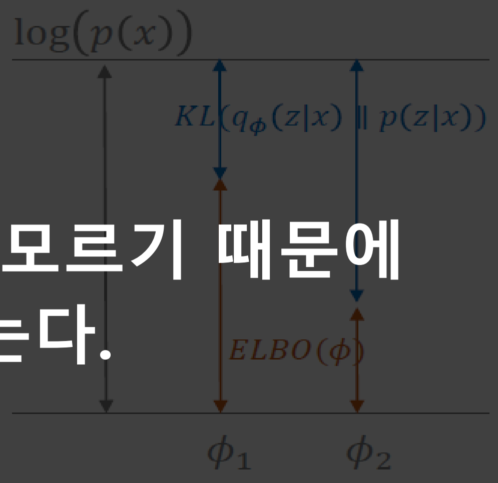
$$= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x)dz \quad \leftarrow p(x) = \frac{p(x,z)}{p(z|x)}$$

$$= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz$$

$$= \underbrace{\int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz}_{ELBO(\phi)} + \underbrace{\int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz}_{KL\left(q_\phi(z|x) \parallel p(z|x)\right)}$$

두 확률분포 간의 거리≥0

Variational inference



KL을 최소화하는 $q_\phi(z|x)$의 ϕ값을 찾으면 되는데 p(z|x)를 모르기 때문에 KL 최소화 대신에 ELBO를 최대화하는 ϕ값을 찾는다.

✓ Variational Autoencoder

parameter(ϕ, θ)찾아보자

$$\log(p(x)) = \underbrace{ELBO(\phi)} + \underbrace{KL(q_\phi(z|x)|p(z|x))}$$

$$q_{\phi^*}(z|x) = \underset{\phi}{\operatorname{argmax}}\, ELBO(\phi)$$

$$\begin{aligned}
\log(p(x)) &= \int \log(p(x))q_\phi(z|x)dz \quad \leftarrow \int q_\phi(z|x)dz = 1 \\
&= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x)dz \quad \leftarrow p(x) = \frac{p(x,z)}{p(z|x)} \\
&= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz \\
&= \underbrace{\int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz}_{ELBO(\emptyset)} + \underbrace{\int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz}_{KL(q_\emptyset(z|x)\,||\,p(z|x))}
\end{aligned}$$

$$\boxed{\begin{aligned}
ELBO(\phi) &= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz \\
&= \int \log\left(\frac{p(x|z)p(z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz \\
&= \int \log(p(x|z))q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right) q_\phi(z|x)dz \\
&= \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)||p(z))
\end{aligned}}$$

✓ Variational Autoencoder

parameter(ϕ, θ)찾아보자

Optimization Problem 1 on ϕ: Variational Inference

$$\log(p(x)) \geq \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)||p(z)) = ELBO(\phi)$$

Optimization Problem 2 on θ: Maximum likelihood

$$-\sum_i \log(p(x_i)) \leq -\sum_i \left\{\mathbb{E}_{q_\phi(z|x_i)}[\log(p(x_i|g_\theta(z)))] - KL(q_\phi(z|x_i)||p(z))\right\}$$

Final Optimization Problem

Loss function

$$\arg\min_{\phi,\theta} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}[\log(p(x_i|g_\theta(z)))] + KL(q_\phi(z|x_i)||p(z))$$

✓ Variational Autoencoder

parameter(ϕ, θ)찾아보자

$$\arg\min_{\phi,\theta} \sum_i \underbrace{-\mathbb{E}_{q_\phi(z|x_i)}[\log(p(x_i|g_\theta(z)))] + KL(q_\phi(z|x_i)||p(z))}_{L_i(\phi,\theta,x_i)}$$

Variational inference를 위한
approximation class 중 선택

원 데이터에 대한 likelihood

다루기 쉬운 확률 분포 중 선택

$$L_i(\phi,\theta,x_i) = \underbrace{-\mathbb{E}_{q_\phi(z|x_i)}[\log(p(x_i|g_\theta(z)))]}_{\text{Reconstruction Error}} + \underbrace{KL(q_\phi(z|x_i)||p(z))}_{\text{Regularization}}$$

Reconstruction Error

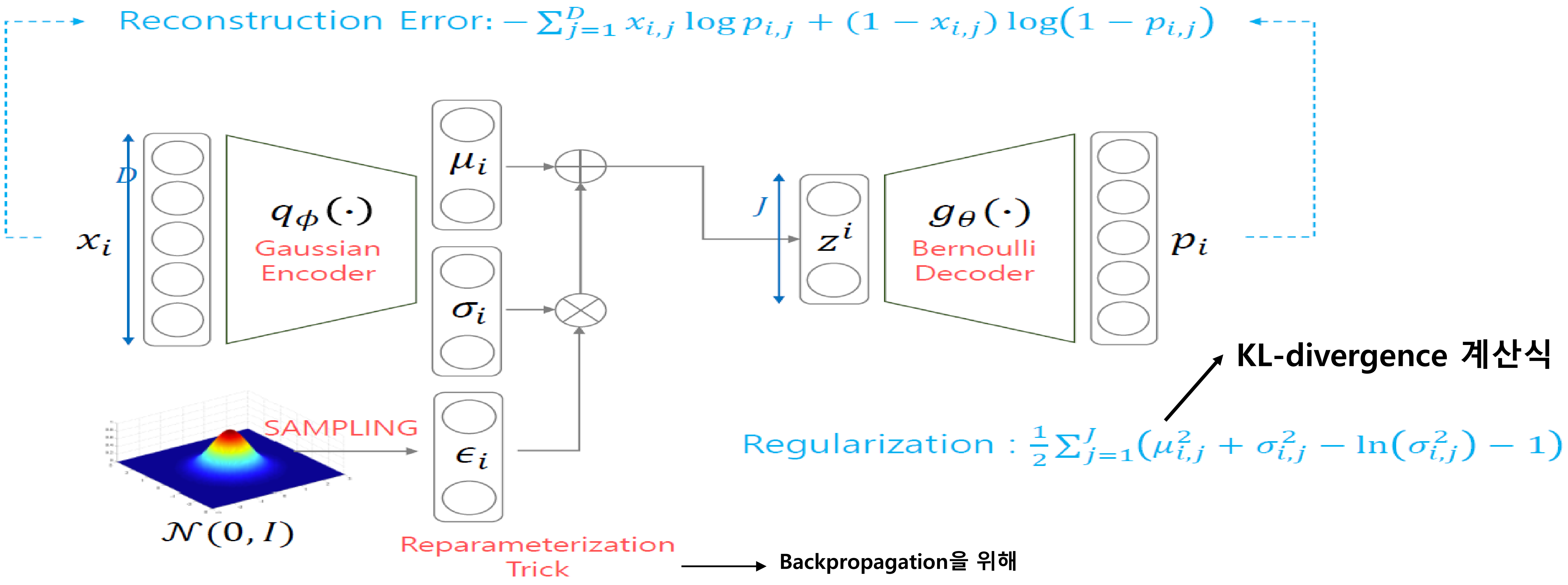
- 현재 샘플링 용 함수에 대한 negative log likelihood
- x_i에 대한 복원 오차 (AutoEncoder 관점)

Regularization

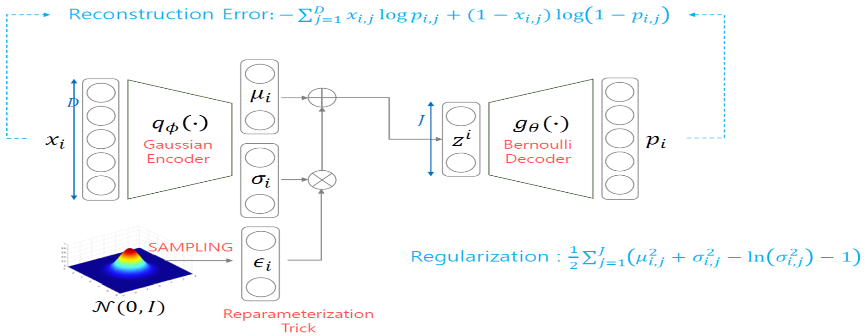
- 현재 샘플링 용 함수에 대한 추가 조건
- 샘플링의 용의성/생성 데이터에 대한 통제성을 위한 조건을 prior에 부여 하고 이와 유사해야 한다는 조건을 부여

코드: https://github.com/hwalsuklee/tensorflow-mnist-VAE

✓ Variational Autoencoder



Unit 02 | Generative Model



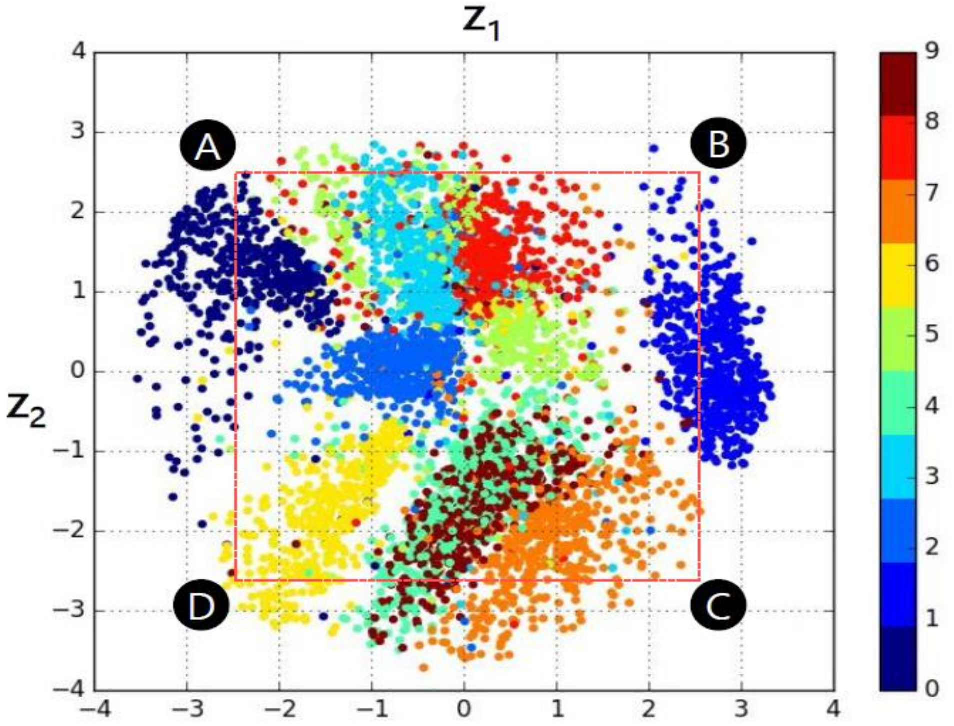
✓ Variational Autoencoder

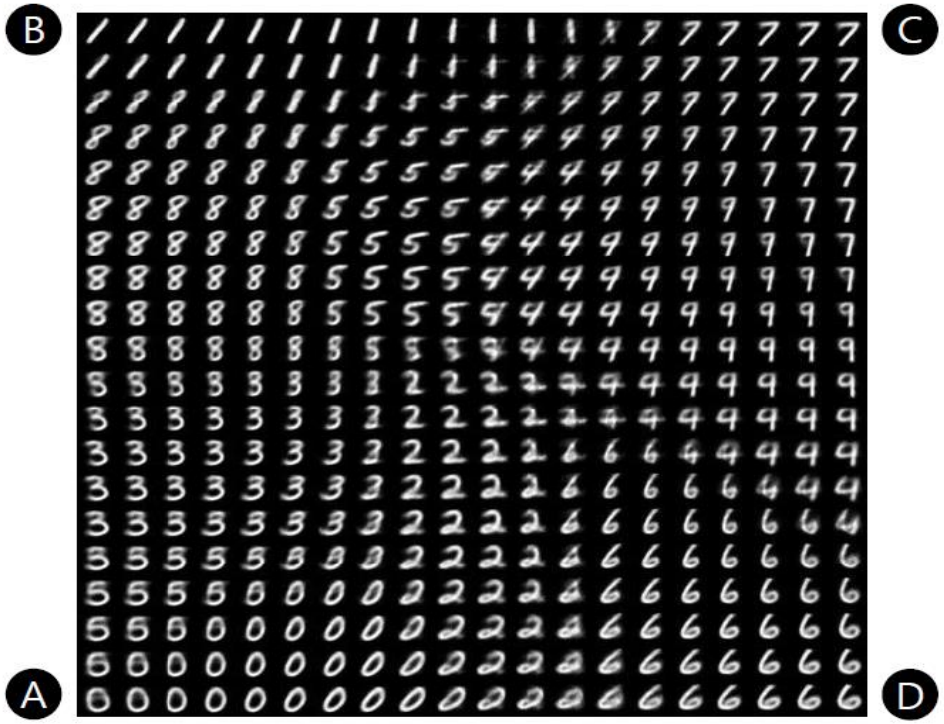
정리!

1. MLE 사용? -> 유의미한 샘플 생성 못함 -> Posterior를 모름 -> Variational Inference 사용
2. Posterior를 모름 -> KLD 계산 불가 -> ELBO를 최대화
3. ELBO최대화 -> reconstruction과 regularization을 최적화 -> 파라미터 θ, ϕ 찾기(training)
4. ϕ를 이용해 샘플링한 latent variable z를 θ를 이용한 generator(decoder)를 통해 생성

✓ **Variational Autoencoder**

Example





✓ Variational Autoencoder

장점

- 생성모델에 대한 수학적 접근
- 인코더가 다른 작업에 대해 유용한 feature representation을 할 수 있음

단점

-가능도의 lower bound를 최대로 하는 방식이 autoregressive model보다 좋은 결과가 있지 않음

- GAN에 비해 생성된 결과가 블러가 많고 질이 떨어짐

Contents

✓ **Generative Adversarial Network란?**

생성적	적대	신경망
Generative	Adversarial	Network

✓ Generative Adversarial Network란?

생성적	적대	신경망
Generative	Adversarial	Network
GAN의 목적 → 새로운 데이터 생성	생성자(G) VS 판별자(D) 두가지 모델이 서로 겨루면서 훈련	생성자(G)와 판별자(D)의 생김새 FNN,CNN,U-Net…

Unit 03 | Generative Adversarial Network

✓ Generative Adversarial Network란?

생성적 Generative	적대 Adversarial	신경망 Network
GAN의 목적 → 새로운 데이터 생성	생성자(G) VS 판별자(D) 두가지 모델이 서로 겨루면서 훈련	생성자(G)와 판별자(D)의 생김새 FNN,CNN,U-Net…

✓ Generative Adversarial Network란?

생성적 Generative	적대 Adversarial	신경망 Network
GAN의 목적 → 새로운 데이터 생성	생성자(G) VS 판별자(D) 두가지 모델이 서로 겨루면서 훈련	생성자(G)와 판별자(D)의 생김새 FNN,CNN,U-Net…

✓ **Generative Adversarial Network란?**

✓ **Generative Adversarial Network란?**

Real x

판별자 D (Discriminator)

D(x)

Sigmoid(내판단) = 0.95정도로
저건 진짜 돈일 것같아

$0 \leq D(x) \leq 1$

✓ Generative Adversarial Network란?

Latent Code z

생성자 G (Generator)

"최고의 위조지폐범이 될테야!"

✓ **Generative Adversarial Network란?**

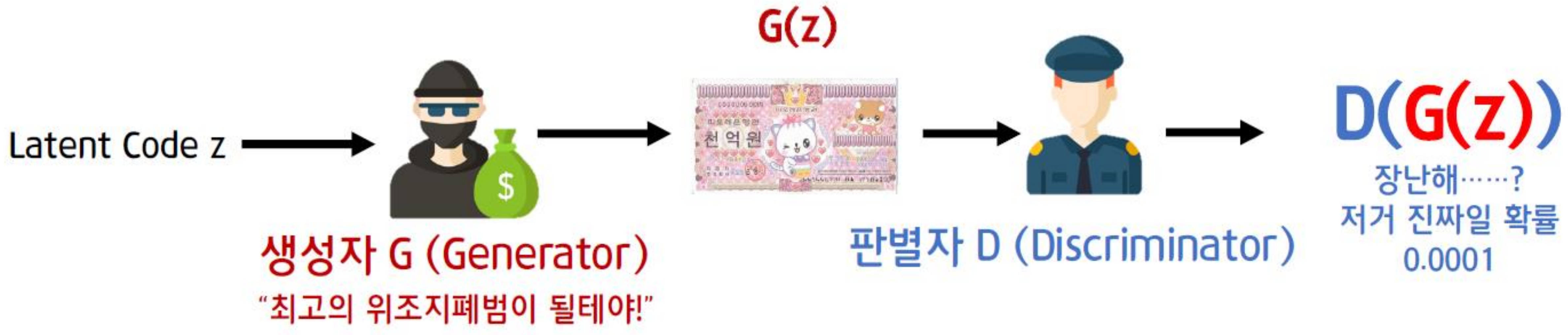
Latent Code z

생성자 G (Generator)

"최고의 위조지폐범이 될테야!"

G(z)

✓ **Generative Adversarial Network란?**



✓ Generative Adversarial Network란?

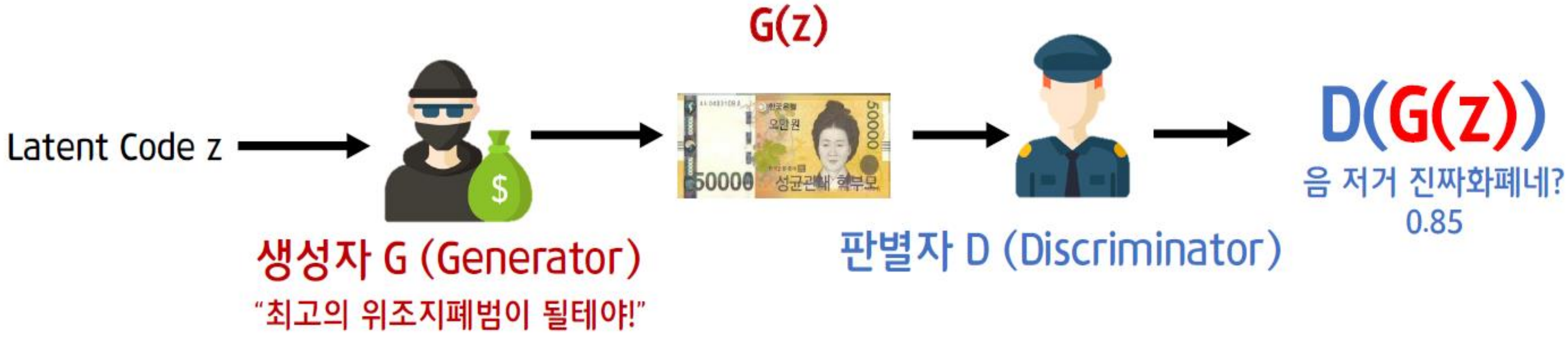
Latent Code z →  →

생성자 G (Generator)

"최고의 위조지폐범이 될테야!"

✓ **Generative Adversarial Network란?**

✓ **Generative Adversarial Network란?**



✓ 생성자와 판별자의 네트워크

	생성자 G (Generator)	판별자 D (Discriminator)
입력	랜덤한 숫자로 구성된 벡터 z	1. 훈련 데이터셋에 있는 실제 샘플 x 2. 생성자가 만든 가짜 샘플 z
출력	최대한 진짜 같이 보이는 가짜 샘플	입력 샘플이 진짜일 예측 확률
목표	훈련 데이터셋에 있는 샘플 x과 구별이 불가능한 가짜 샘플 $G(z)$ 생성하기	생성자가 만든 가짜 샘플 $G(z)$과 훈련 데이터셋의 진짜 샘플 x 구별하기

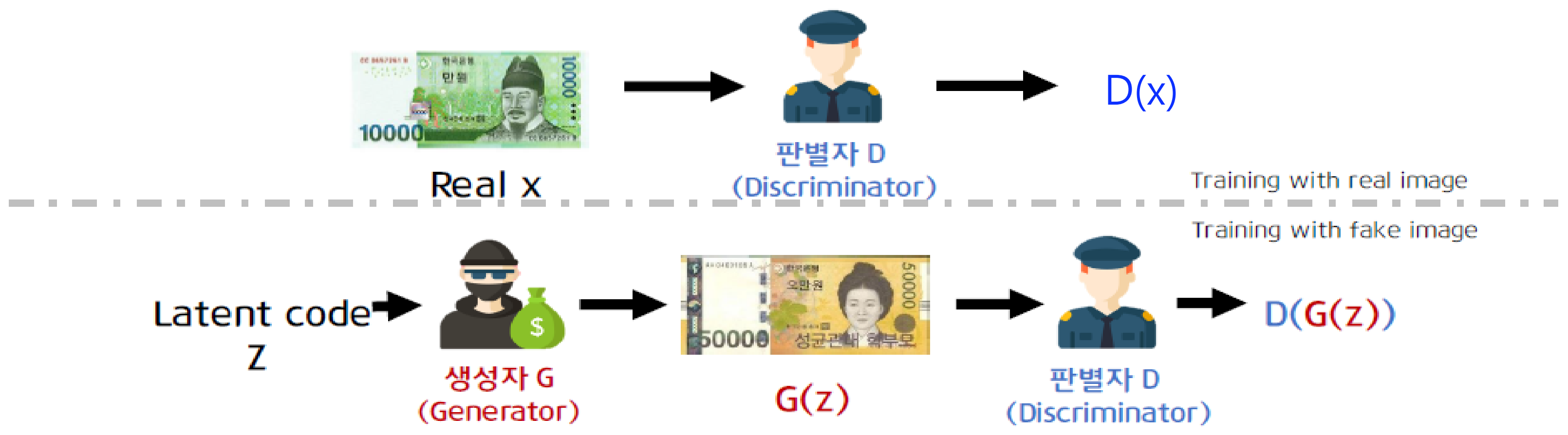
✓ Objective Function

이진크로스엔트로피 BCE = $-\frac{1}{n}\sum_{i=1}^{n}(y_i \log(p_i) + (1-y_i)\log(1-p_i))$

실제 데이터 분포에서 온 Sample x

Gaussian 분포(예)에서 온 Sample latent code z

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



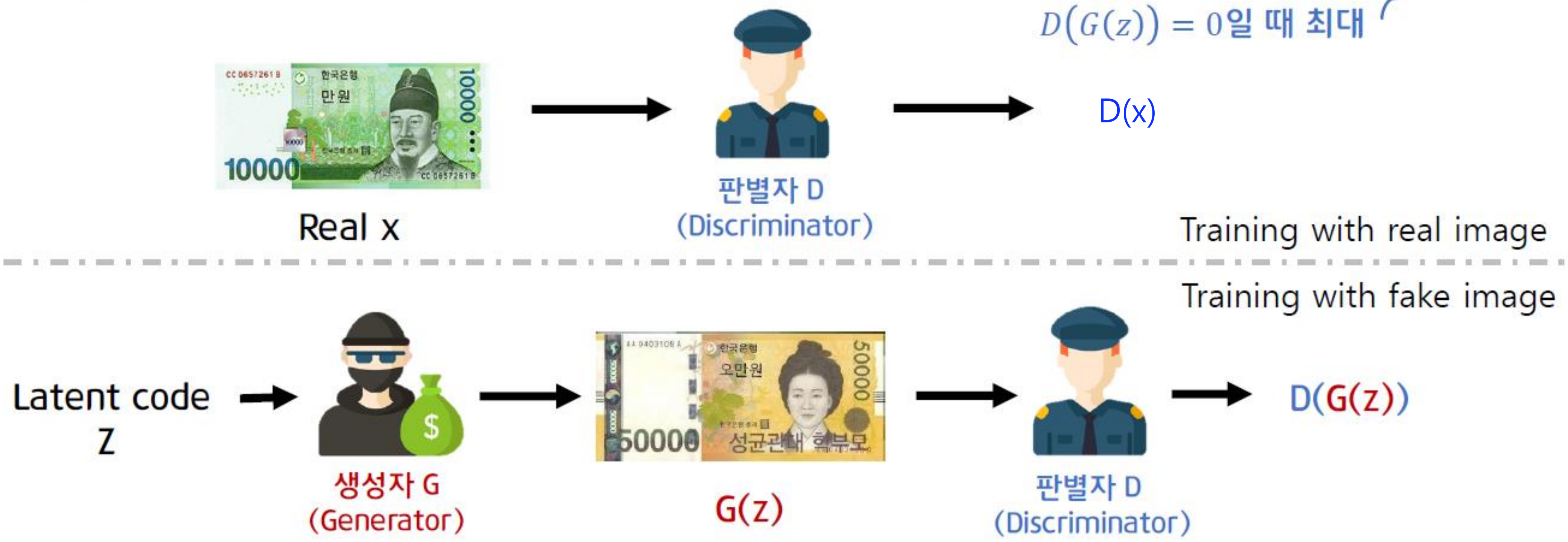
Unit 03 | Generative Adversarial Network

✓ Objective Function

D(x) = 1 일 때 최대

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))]$$

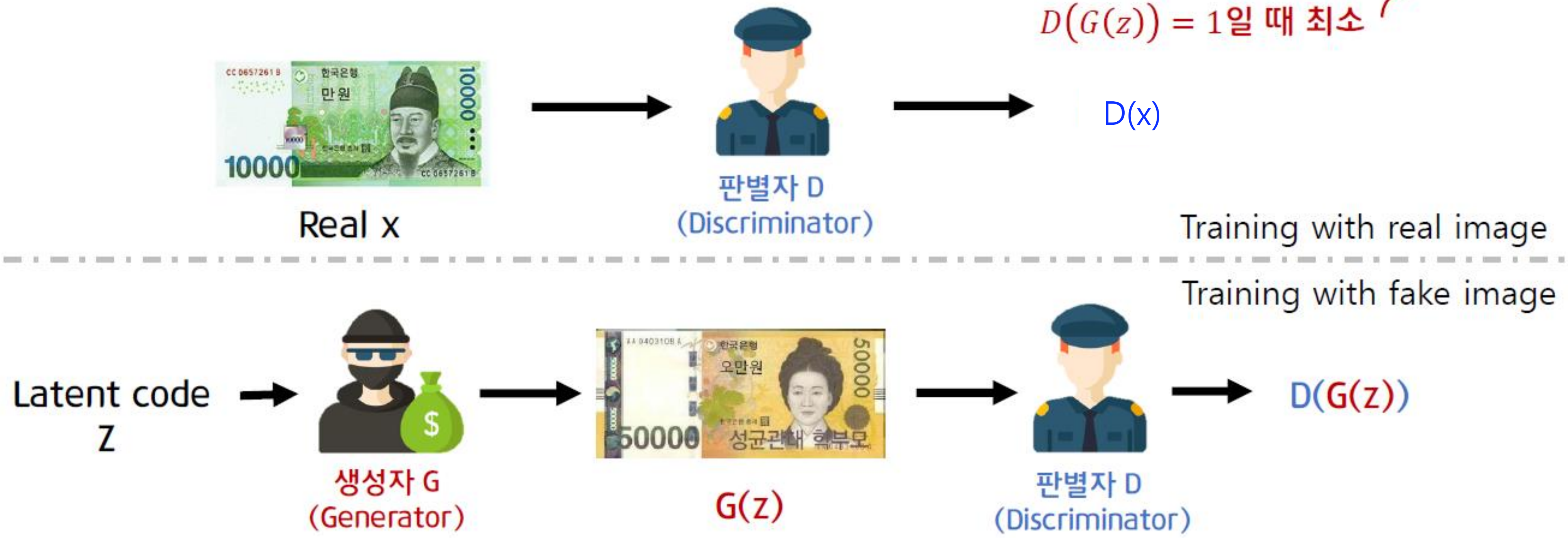
$D(G(z)) = 0$일 때 최대



✓ **Objective Function**

$$\min_{G} \max_{D} V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim P_{z}(z)}[\log(1 - D(G(z)))]$$

$D(G(z)) = 1$일 때 최소



Unit 03 | Generative Adversarial Network

✓ GAN Train

실제 데이터 분포에서 온 Sample x

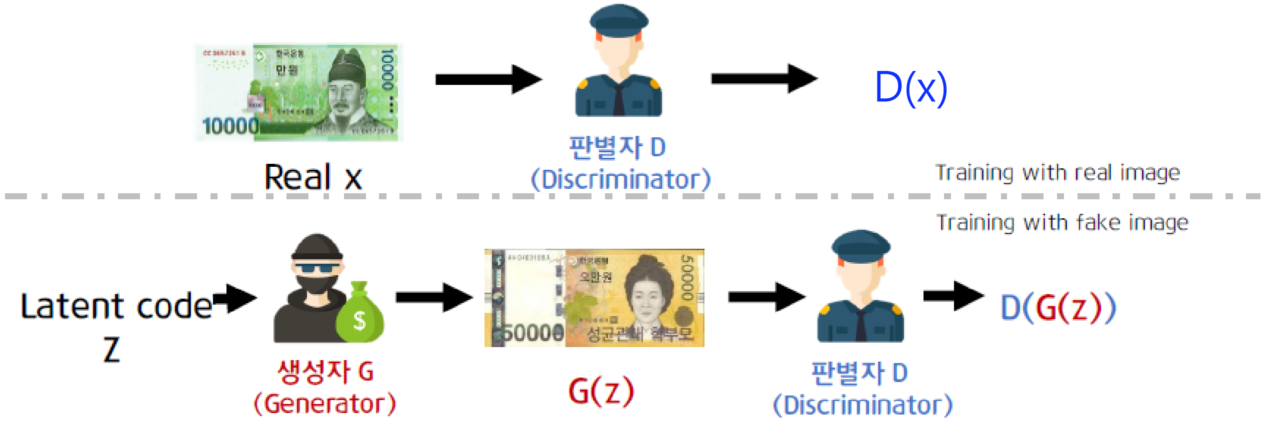
Gaussian 분포(예)에서 온 Sample latent code z

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

$D(x)=1$일 때 최대

$D(G(z))=1$일 때 최소

$D(G(z))=0$일 때 최대




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

latent_size = 64
hidden_size = 256
image_size = 784

# Discriminator
D = nn.Sequential(
    nn.Linear(image_size, hidden_size),
    nn.LeakyReLU(0.2),
    nn.Linear(hidden_size, 1),
    nn.Sigmoid())

# Generator
G = nn.Sequential(
    nn.Linear(latent_size, hidden_size),
    nn.ReLU(),
    nn.Linear(hidden_size, image_size),
    nn.Tanh())


# Binary cross entropy loss and optimizer
criterion = nn.BCELoss()
d_optimizer = torch.optim.Adam(D.parameters(), lr=0.0002)
g_optimizer = torch.optim.Adam(G.parameters(), lr=0.0002)

#Asume x be real image of shape (batch_size,784)
#Asume z be random noise of shape (batch_size,64)

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()
```

✓ GAN의 단점

1. 모델이 진동 및 기울기 소실(vanishing gradient)
2. 하이퍼 파라미터가 많고 민감하다
3. 모드붕괴(Mode collapse)
4. 평가지표 불충분

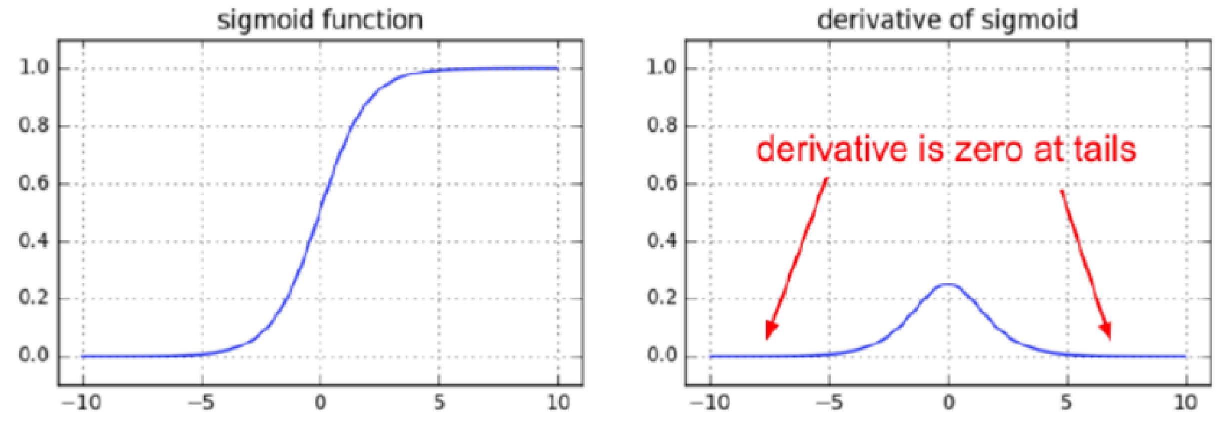
관련 자료: https://machinelearningmastery.com/practical-guide-to-gan-failure-modes/

✓ GAN의 단점

기울기 소실(vanishing gradient): 판별자의 마지막 활성화 함수인 sigmoid로 인해 생기는 문제.
판별자가 생성자에 비해 월등히 성능이 좋은 경우 유의미한 기울기를 전달할 수 없게 된다.

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

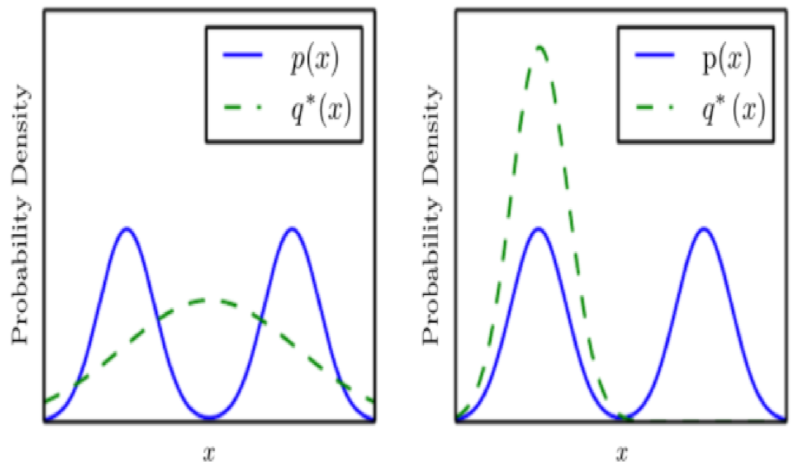
$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

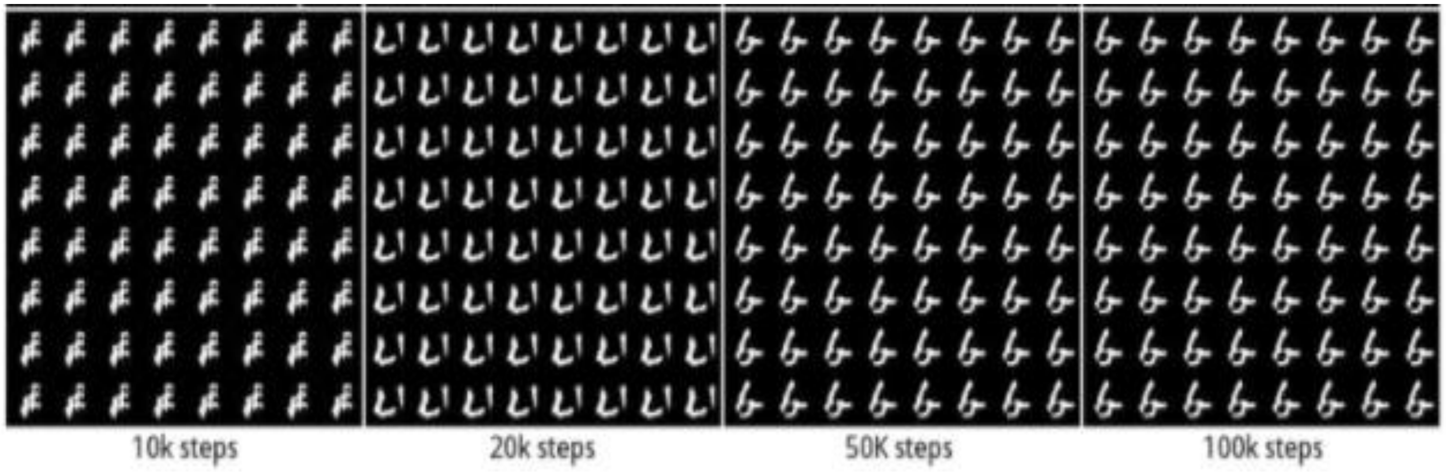


http://jaejunyoo.blogspot.com/2017/02/unrolled-generative-adversarial-network-1.html

✓ GAN의 단점

모드 붕괴(mode collapse): 학습시킨 모델의 분포가 실제 데이터 분포의 모든 부분을 커버하지 못하고 다양성을 잃어버리는 현상.
생성자가 손실(loss)만을 줄이려고 학습을 하기 때문에 하나의 mode에만 강하게 몰리는 경우





Contents

✓ **GAN**

“GANs are the most interesting idea in the 10 years in Machine Learning”

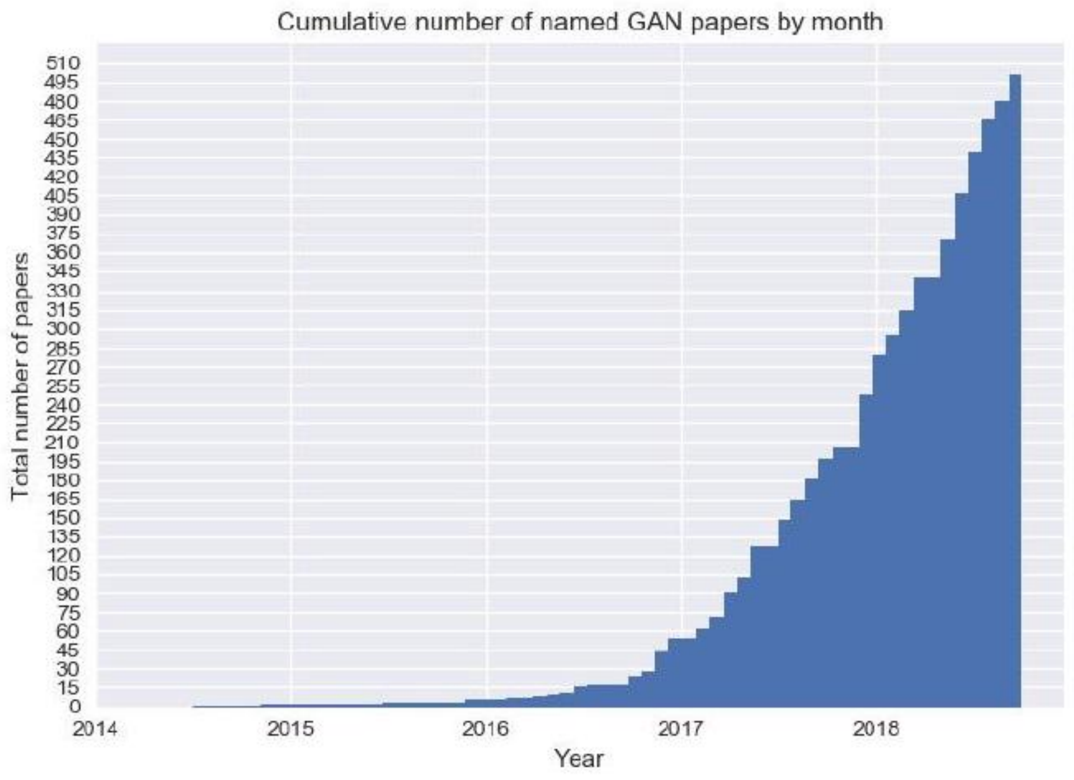
--Yann LeCun, in NIPS(2016)

Yann LeCun
Professor at New York University
Chief AI Scientist at Facebook

https://en.wikipedia.org/wiki/Yann_LeCun

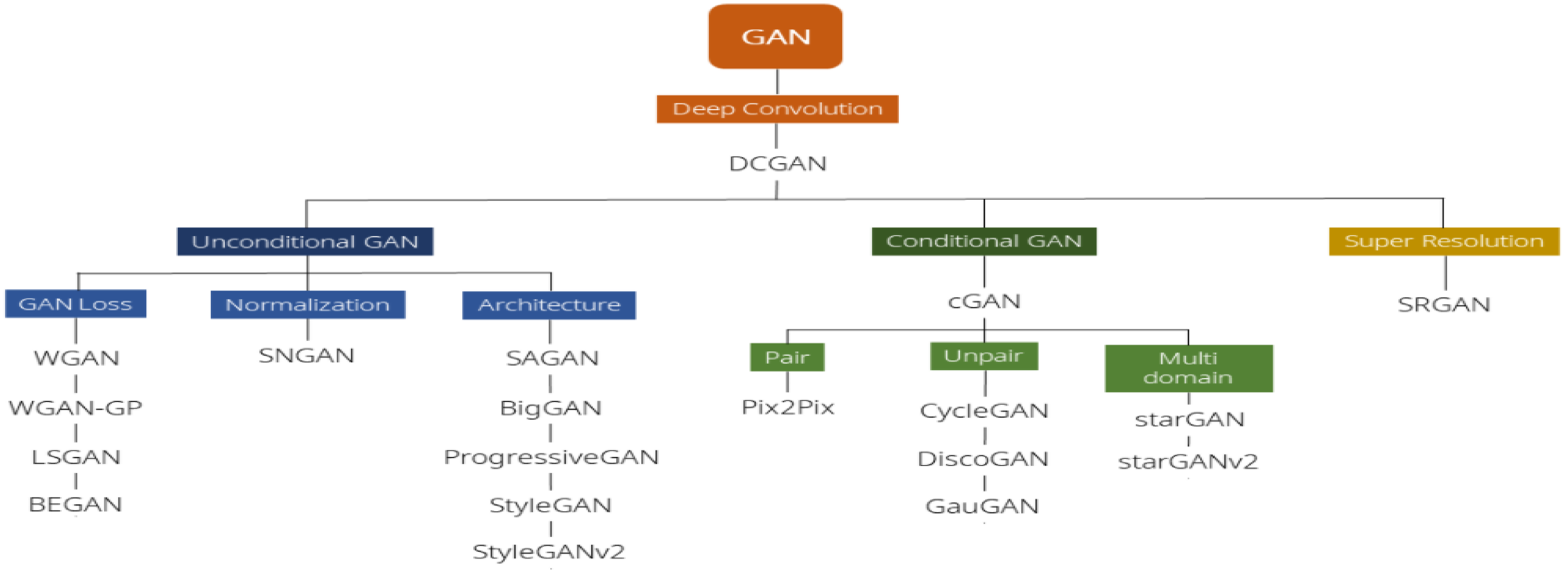
Unit 04 | Beyond Vanilla GAN

✓ GAN의 논문 동향



https://github.com/hindupuravinash/the-gan-zoo

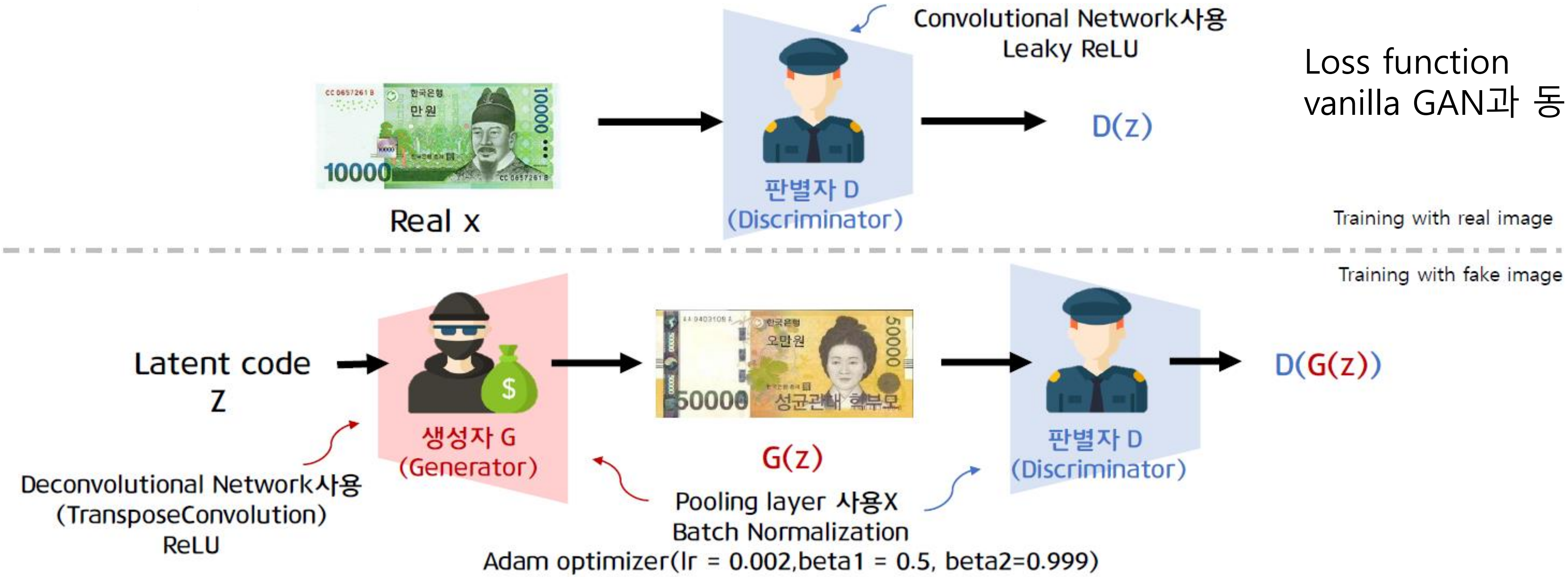
✓ GAN의 종류



https://ysbsb.github.io/gan/2020/06/17/GAN-newbie-guide.html

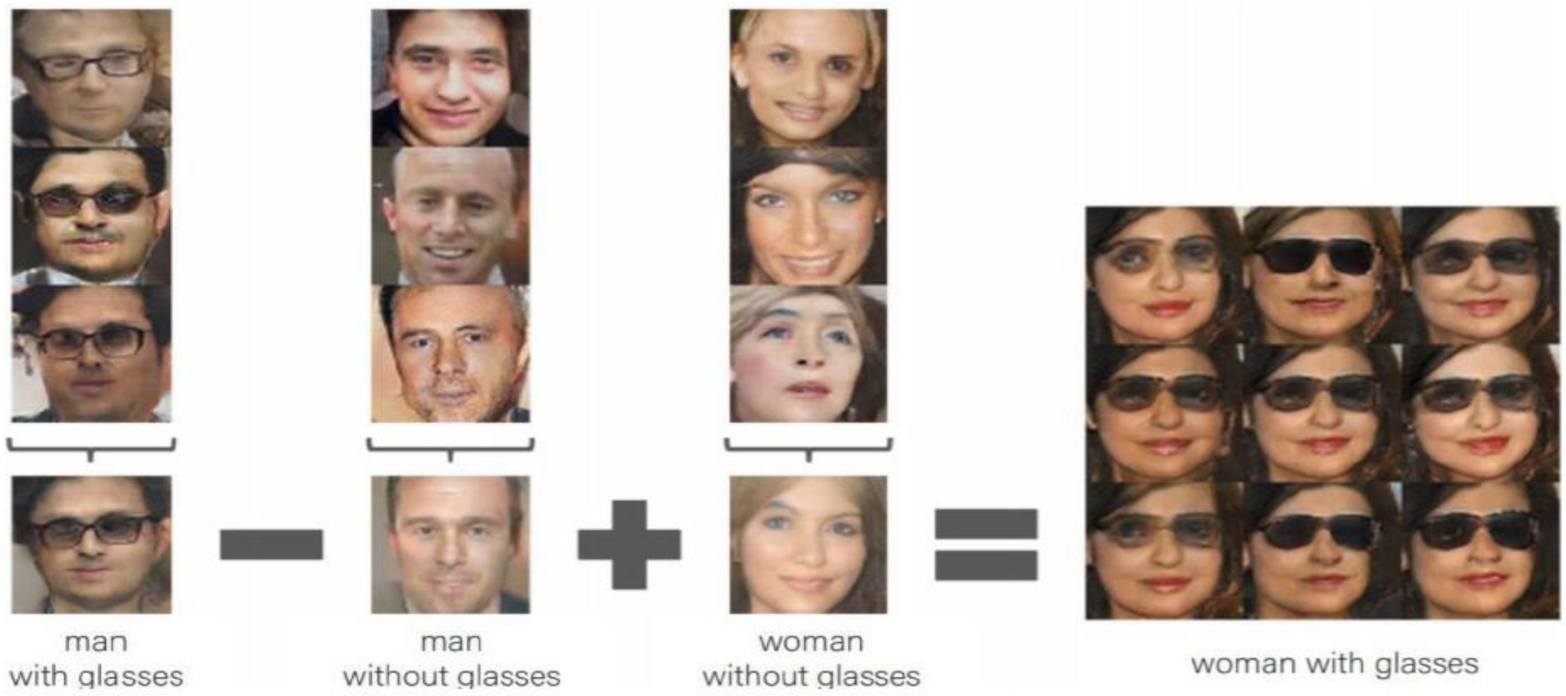
Unit 04 | Beyond Vanilla GAN

✓ DCGAN



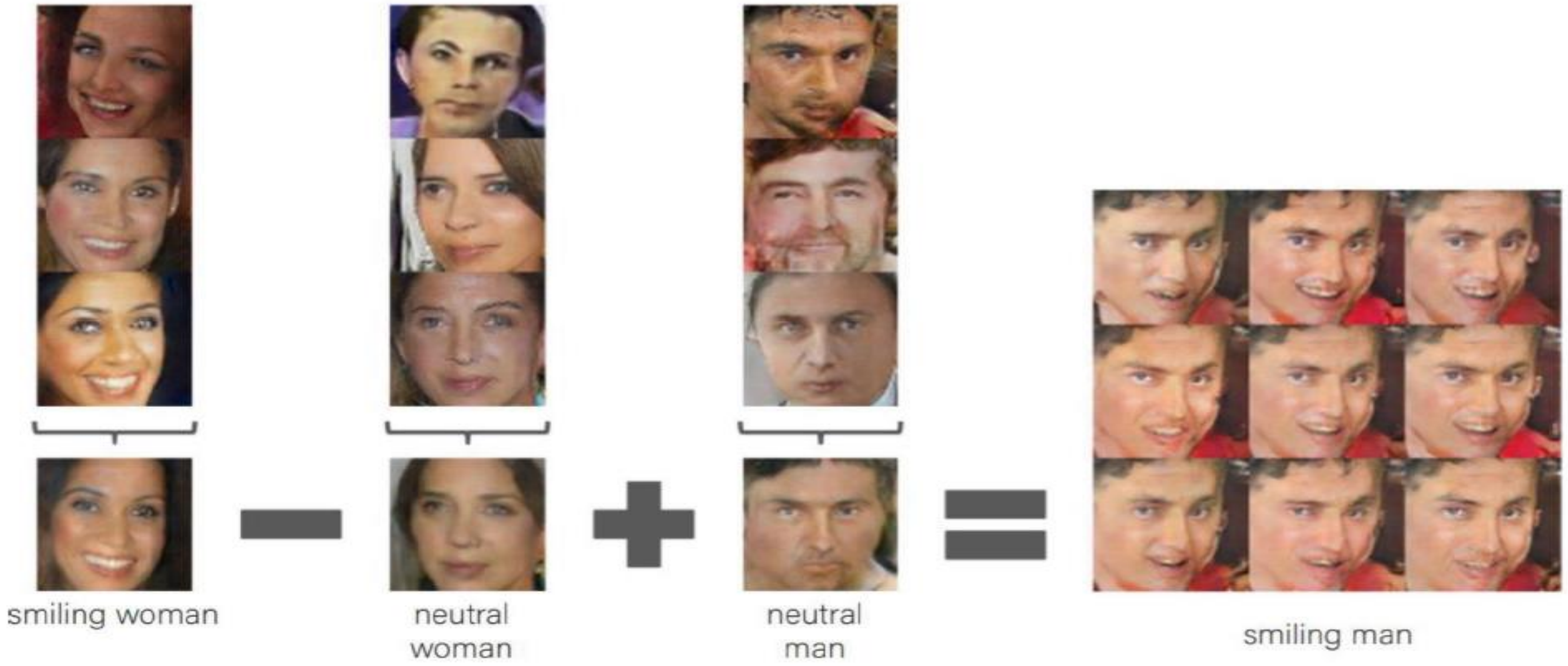
Loss function
vanilla GAN과 동일

✓ DCGAN – vector space arithmetic



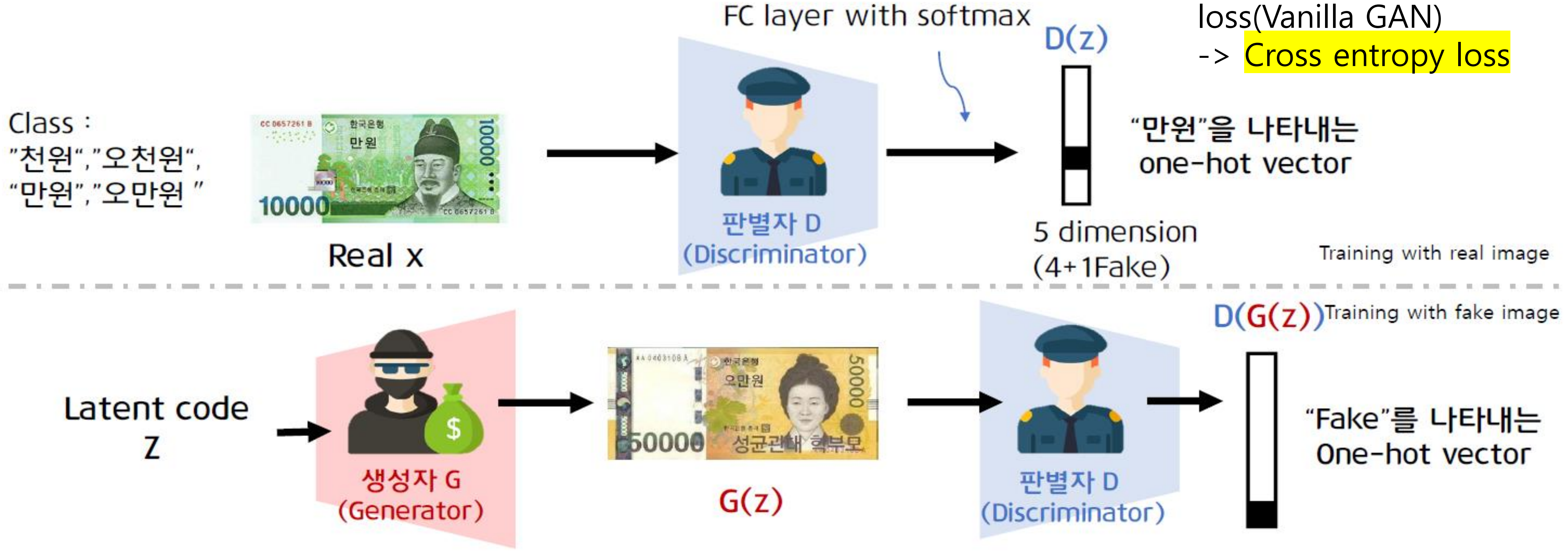
https://arxiv.org/abs/1511.06434

✓ **DCGAN – vector space arithmetic**

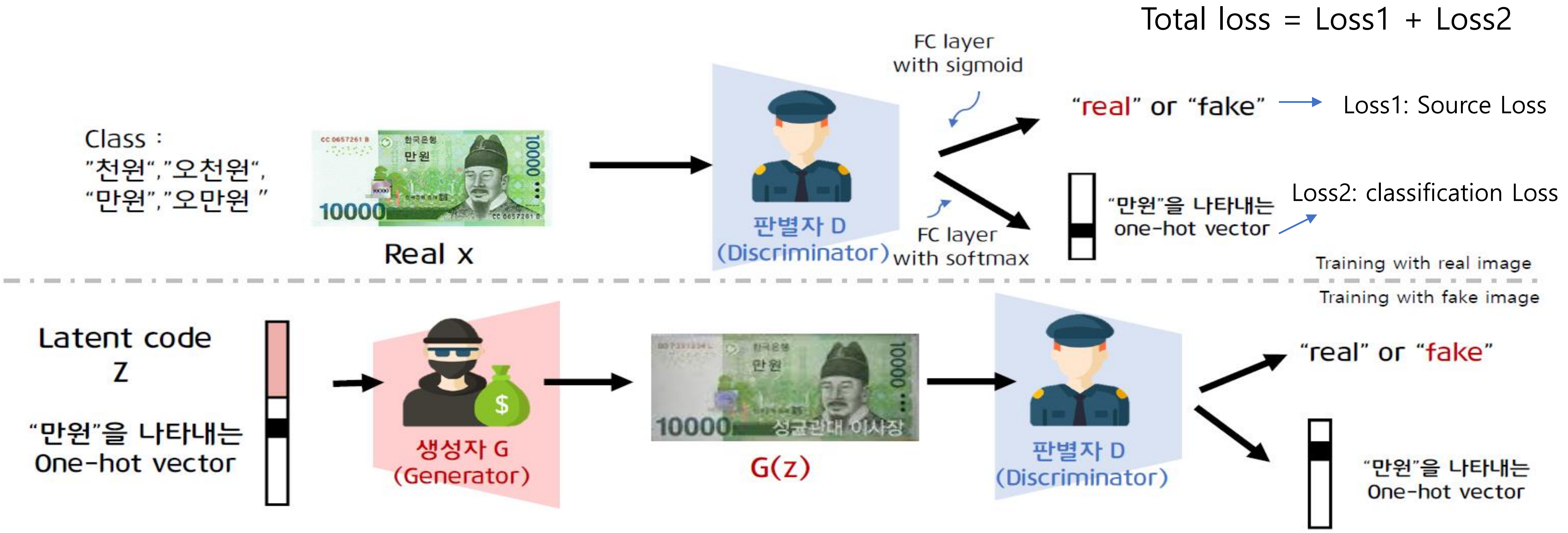


✓ SGAN(Semi-Supervised GAN)

Loss function
Binary cross entropy loss(Vanilla GAN)
-> Cross entropy loss

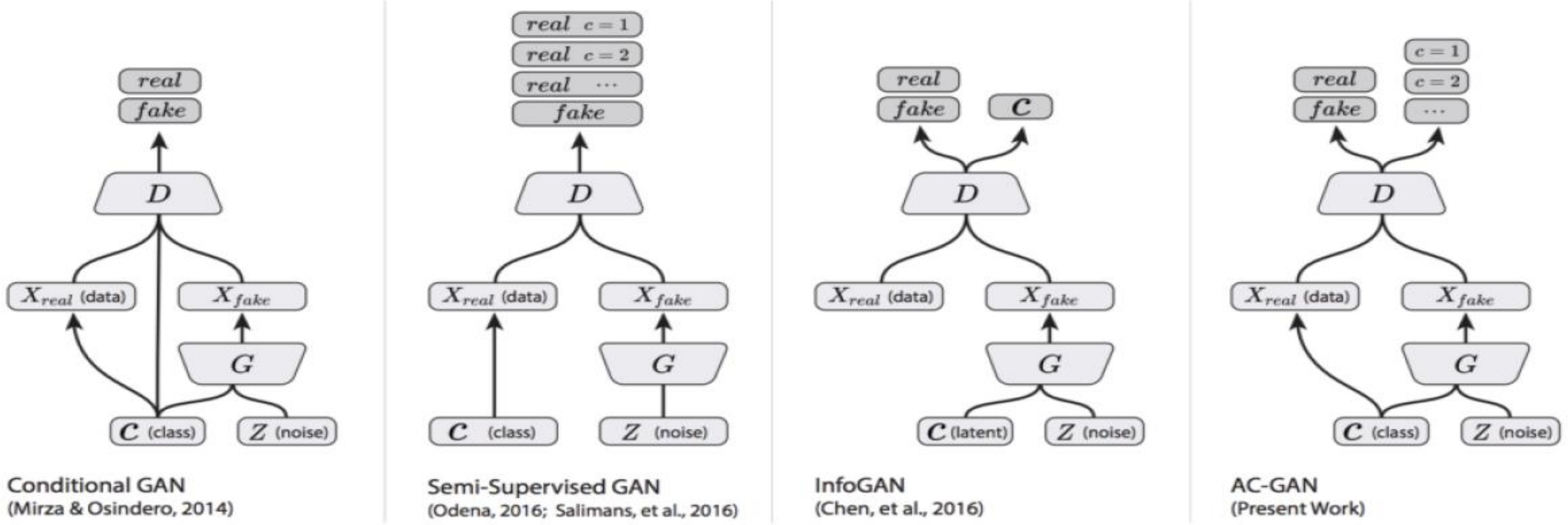


✓ **ACGAN(Auxiliary Classifier GAN)**



Unit 04 | Beyond Vanilla GAN

✓ cGAN(Conditional GANS)



https://arxiv.org/abs/1610.09585v1

✓ CycleGAN

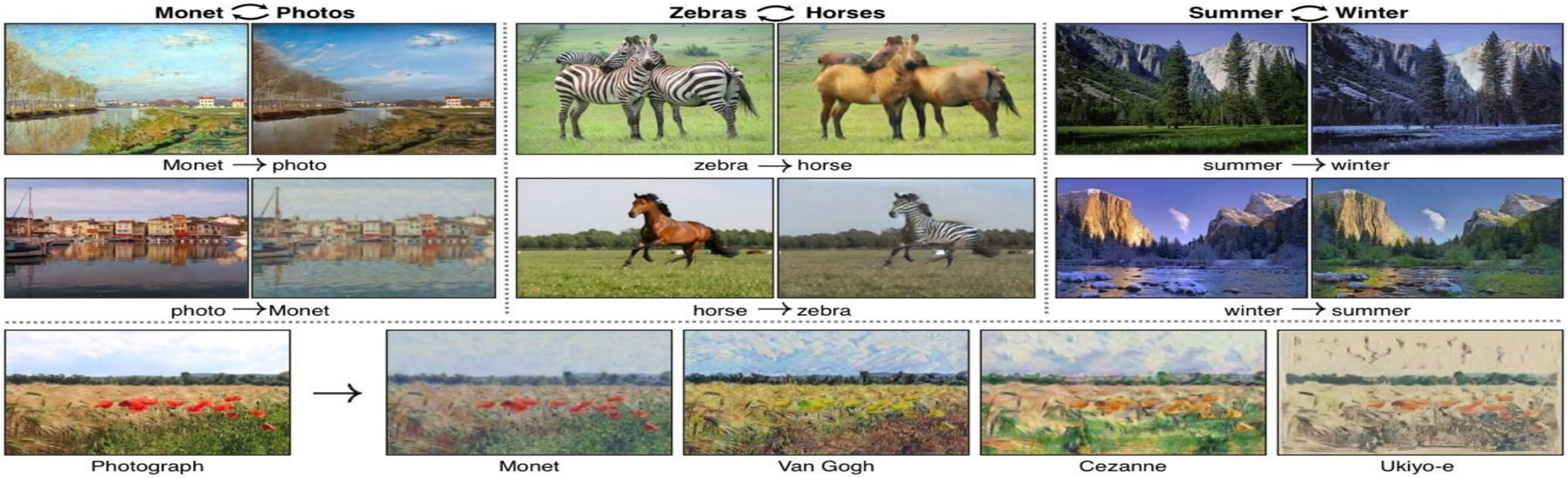


Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically "translate" an image from one into the other and vice versa: *(left)* Monet paintings and landscape photos from Flickr; *(center)* zebras and horses from ImageNet; *(right)* summer and winter Yosemite photos from Flickr. Example application *(bottom)*: using a collection of paintings of famous artists, our method learns to render natural photographs into the respective styles.

https://arxiv.org/abs/1703.10593

✓ StarGAN

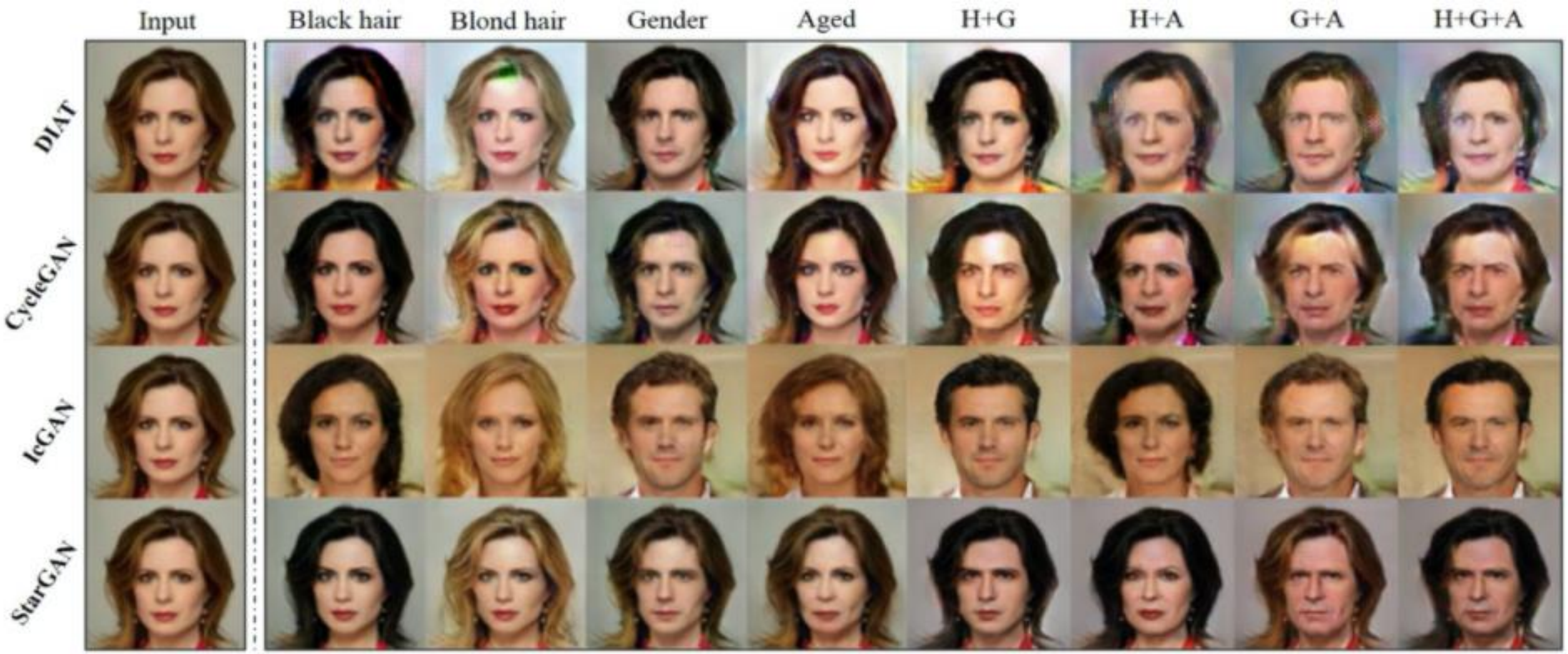


Figure 4. Facial attribute transfer results on the CelebA dataset. The first column shows the input image, next four columns show the single attribute transfer results, and rightmost columns show the multi-attribute transfer results. H: Hair color, G: Gender, A: Aged.

https://arxiv.org/abs/1711.09020

https://arxiv.org/abs/1812.04948

✓ styleGAN

https://thispersondoesnotexist.com/
By stylegan2

Figure 3. Visualizing the effect of styles in the generator by having the styles produced by one latent code (source) override a subset of the styles of another one (destination). Overriding the styles of layers corresponding to coarse spatial resolutions ($4^2 - 8^2$), high-level aspects such as pose, general hair style, face shape, and eyeglasses get copied from the source, while all colors (eyes, hair, lighting) and finer facial features of the destination are retained. If we instead copy the styles of middle layers ($16^2 - 32^2$), we inherit smaller scale facial features, hair style, eyes open/closed from the source, while the pose, general face shape, and eyeglasses from the destination are preserved. Finally, copying the styles corresponding to fine resolutions ($64^2 - 1024^2$) brings mainly the color scheme and microstructure from the source.

Assignment

✓Assignment

DCGAN 논문리뷰 + 코드 구현

Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
https://arxiv.org/abs/1511.06434

제출파일
1. 논문리뷰(형식: word또는 pdf)
2. assignment_wk10.ipynb 빈칸 완성

구현(코드)과제에서 모델을 학습 시킬 때 5epoch 미만으로 모델이 잘 실행되는지 확인해주세요
Cpu학습보단 gpu(colab) 환경에서 학습을 추천합니다.
개인마다 사용할 수 있는 컴퓨팅 자원이 다르므로 학습의 결과(생성 이미지)는 우수과제 선정에서 고려하지 않겠습니다. 따라서, 논문리뷰 + 코드 구현에서 주석 및 마크다운으로 상세한 설명을 기준으로 우수과제 선정하겠습니다
*이해한 만큼 코드주석 및 부가설명(마크다운)!
*과제에 대해 이해가 안 되거나 어렵다면 망설이지 말고 연락주세요~

References

✓ References

✓ Tobig's 13기 신민정님 "비지도&생성모델 " 강의자료

✓ Tobig's 13&14기 생성모델 심화 세미나 https://velog.io/@tobigs-gm1

✓ CS231n 13강 Generative Model

✓ "오토인코더의 모든것" – 이활석님 강의 (NaverD2 Youtube)

✓ "1시간만에 GAN 완전 정복하기" – 최윤제님 강의 (NaverD2 youtube)

✓ 이외의 참고자료들은 해당 슬라이드 내용과 연관된 것으로 해당 슬라이드에 링크를 표시했습니다

Q & A

들어주셔서 감사합니다.