17기 정규세션

ToBig's 16기 강의자
김권호

Generative Basic

Contents

Contents

Supervised vs Unsupervised

	Supervised Learning	Unsupervised Learning
Train	Data : (X, Y) pair	Data : X only
Prediction	$\hat{y} = f(x)$의 f 를 학습	데이터의 기본 구조(f_X)를 학습

Example of Supervised Learning

Supervised Learning

Data : (x, y) 한 쌍
$\hat{y} = f(x)$의 f를 학습

task → **Classification**

x → 

y → CAT

Example of Supervised Learning

Supervised
Learning

Data : (x, y) 한 쌍
$Y = f(x)$의 f를 학습

task → Instance Segmentation

x → 

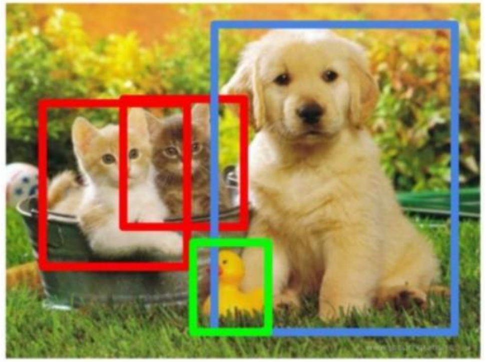
y → CAT, DOG, DUCK

Example of Supervised Learning

Supervised Learning

Data : (x, y) 한 쌍
$Y = f(x)$의 f를 학습

task → Object Detection

x → 

y → CAT, DOG, DUCK

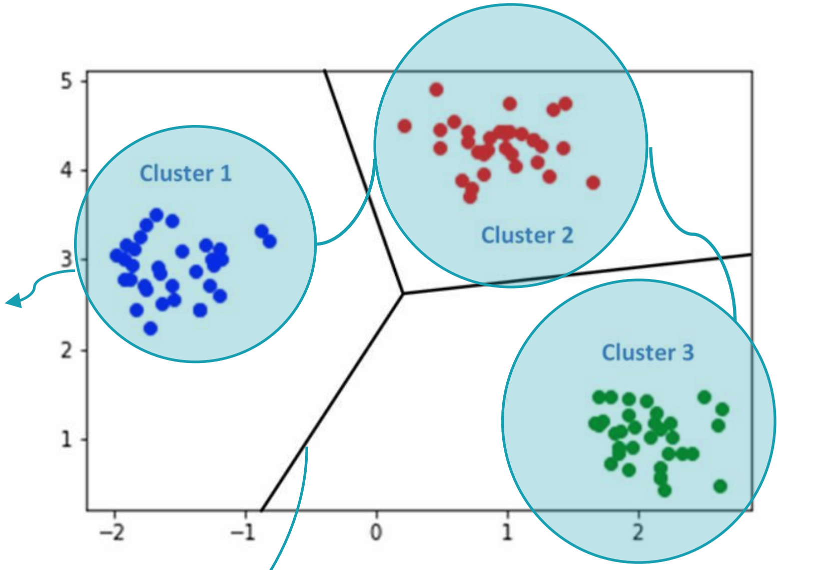
Example of Unsupervised Learning

task **Clustering**

Unsupervised Learning

Data : 오직 x만
데이터의 기본 구조를 학습

X



f_X

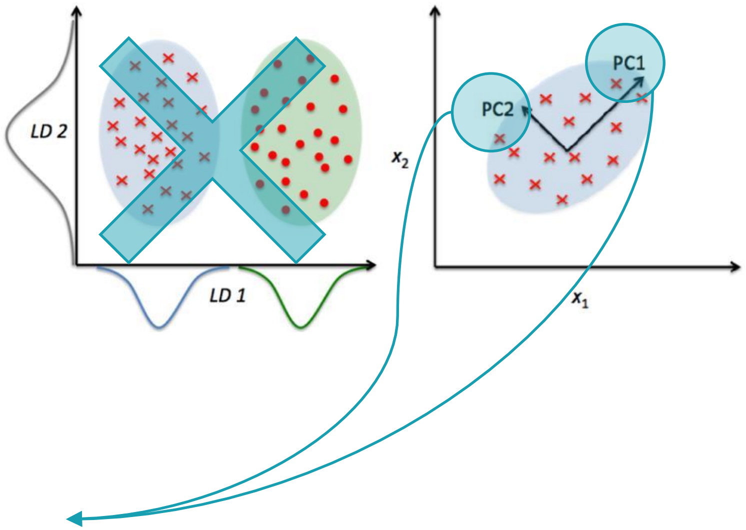
Example of Unsupervised Learning

task

Dimension Reduction

Unsupervised Learning

X



f_X

Data : 오직 x만
데이터의 기본 구조를 학습

Example of Unsupervised Learning

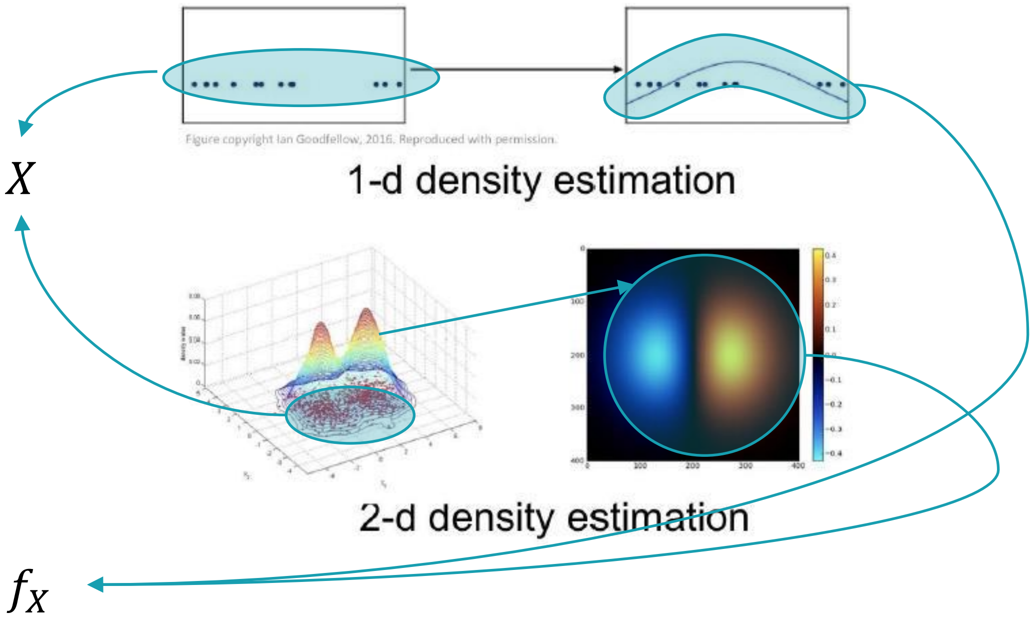
task

Density Estimation

Unsupervised Learning

Data : 오직 x만

데이터의 기본 구조를 학습

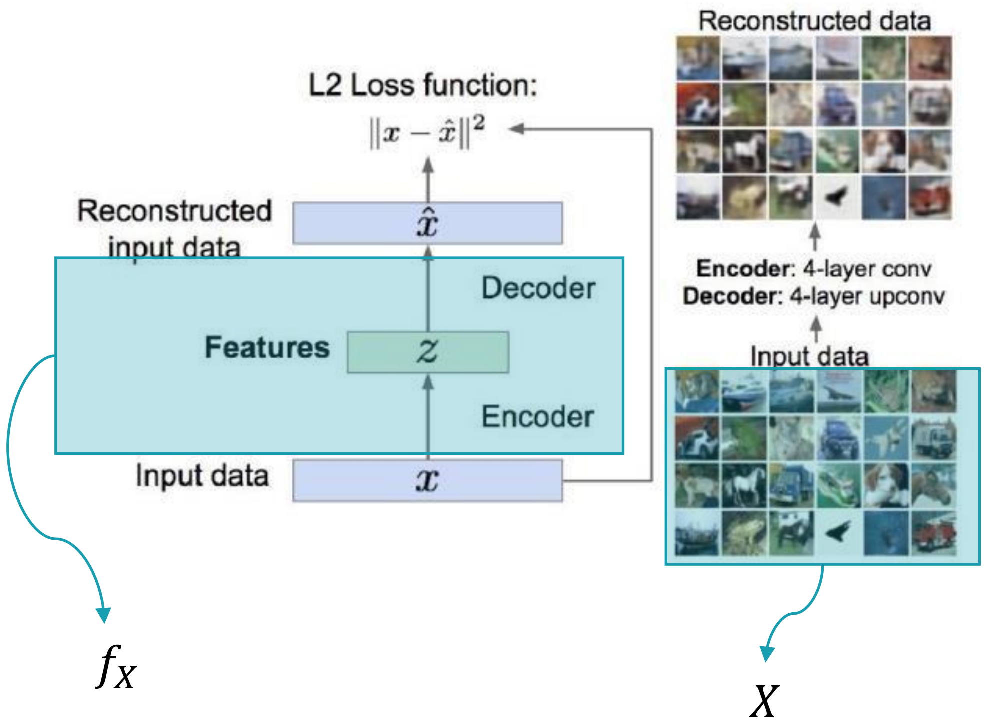


Example of Unsupervised Learning

task **Representation Learning**

Unsupervised Learning

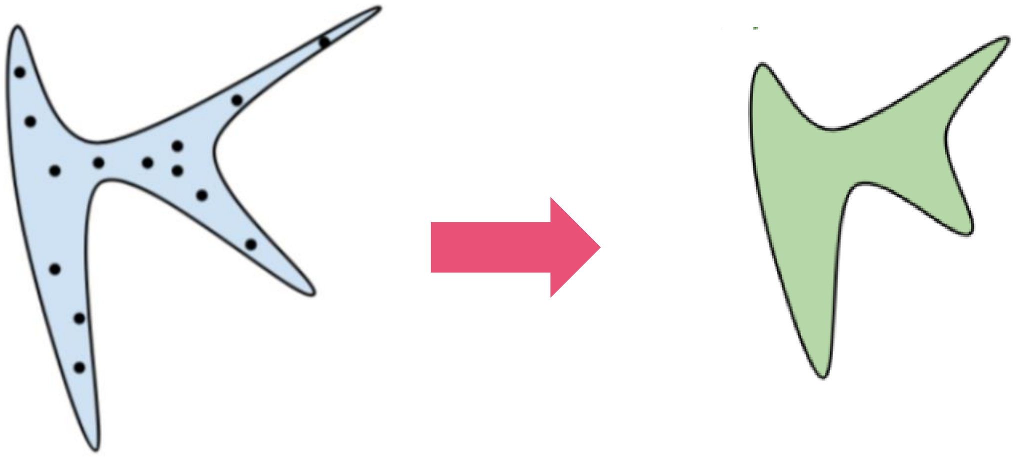
Data : 오직 x만
데이터의 기본 구조를 학습



Generative Modeling

Generative
Modeling

Unsupervised learning
주어진 data의 분포와 비슷한 분포를 모델링
같은 분포로부터 새로운 sample을 생성하는 것이 목적

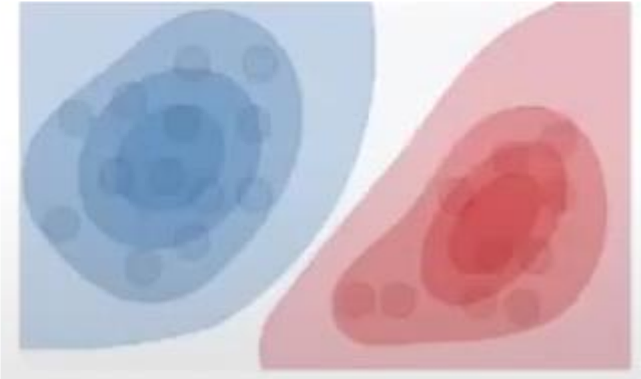


$p_{data}(x)$ $p_{model}(x)$

Unit 01 | Unsupervised Learning

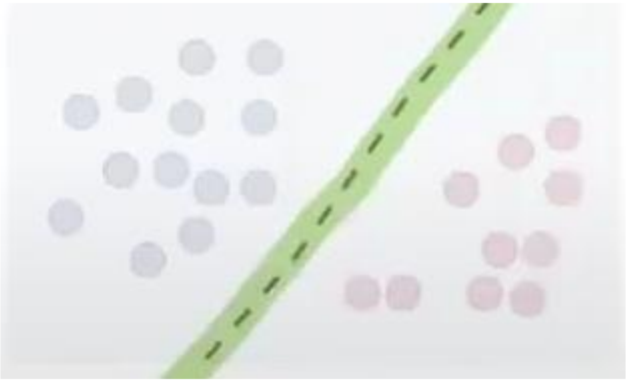
Generative Modeling?

Generative Modeling



Sample x로 $p(x)$를 추정
Label이 있는 경우, $p(x|y)$를 추정
density estimation
probability distribution of data 학습
-> decision 역시 가능하다

Discriminative Modedling



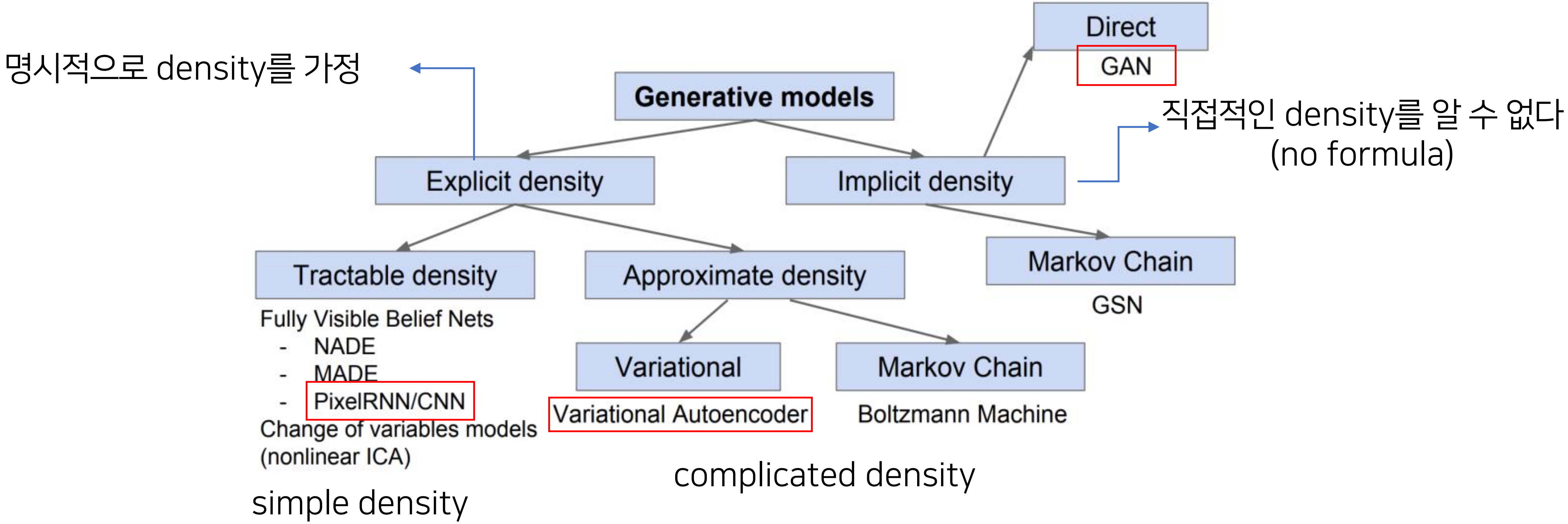
판별 모델링
Sample x가 주어졌을 때
Label y의 확률인 $p(y|x)$ 추정
decision boundary 학습

Generative Modeling

- Sample $dataset\ X$를 가지고 있다.
- Sample이 알려지지 않은 어떤 p_{data}분포로 생성되었다고 가정한다.
- 생성모델에서 p_{model}이 p_{data}를 **모사**하는 것이 목표이다.
  - 이 목표를 달성한다면 p_{model}에서 **샘플링**을 진행해 얻은 샘플은 마치 p_{data}에서 뽑은 것 같은 샘플일 것이다
- p_{model}은 아래의 조건을 만족시키면 좋을 것이다.
  - p_{data}에서 뽑은 것 같은 샘플을 생성할 수 있다. → 유사성 ≈ 성능 ↑
  - $dataset\ X$에 있는 sample과 다른 sample을 생성할 수 있다. → 다양성 ≈ 과적합 ↓

Generative Model의 분류

Taxonomy : 분류 체계



Generative Model의 분류

Taxonomy : 분류 체계

1. explicit density estimation(likelihood-based models)
- explicitly define and solve for p_{model}

2. implicit density estimation
- learn a model that can sample from p_{model} w/o explicitly defining it (just for sampling)

represent / manipulate high-dim, complicated probability distribution
(simulating, imputation…)

Contents

Fully visible belief network(FVBN)

- explilcit density model(tractable density)

1. decompose likelihood of input x into a product of 1D distributio

$$\underbrace{p(\boldsymbol{x})}_{\text{likelihood of input } \boldsymbol{x}} = \underbrace{\prod_{i=1}^{n} p(x_i \mid x_1, \ldots, x_{i-1})}_{\text{probability of } i\text{-th feature given all previous features}}$$

2. maximize likellilhood of training data (data의 분포와 model의 분포 사이의 cross entropy minimize)

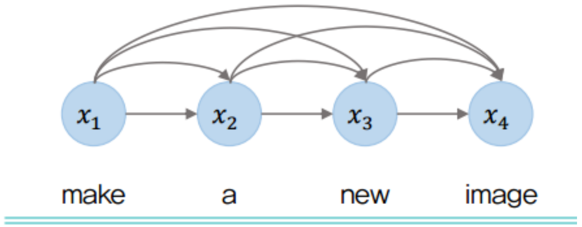
-> 각 feature의 conditional 분포는 매우 복잡하므로 neural net을 이용해 approximate

Need to define ordering of previous feature

Unit 02 | Pixel RNN, Pixel CNN

Pixel RNN

자기 자신을 입력으로 하여 자기 자신을 예측하는 모델. (autoregressive model)
텍스트, 오디오, 이미지 등 다양한 종류의 데이터 특성을 반영할 수 있다.



make	a	new	image
make	a	new	image
make	a	new	image
make	a	new	image

Joint probability를
연속된 conditional probability로
분해

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|x_1,\dots,x_{t-1})$$

$$p(x_1,x_2,x_3,x_4) = p(x_4|x_1,x_2,x_3)\,p(x_1,x_2,x_3)$$

$p(x_1)$

$p(x_2|x_1)$

$p(x_3|x_1,x_2)$

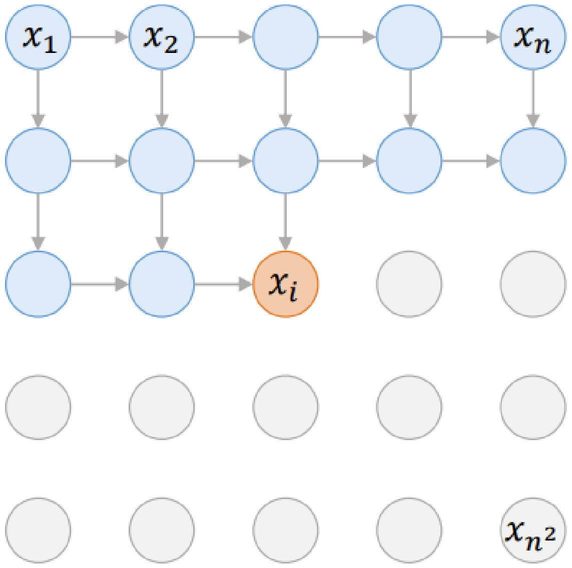
$p(x_4|x_1,x_2,x_3)$

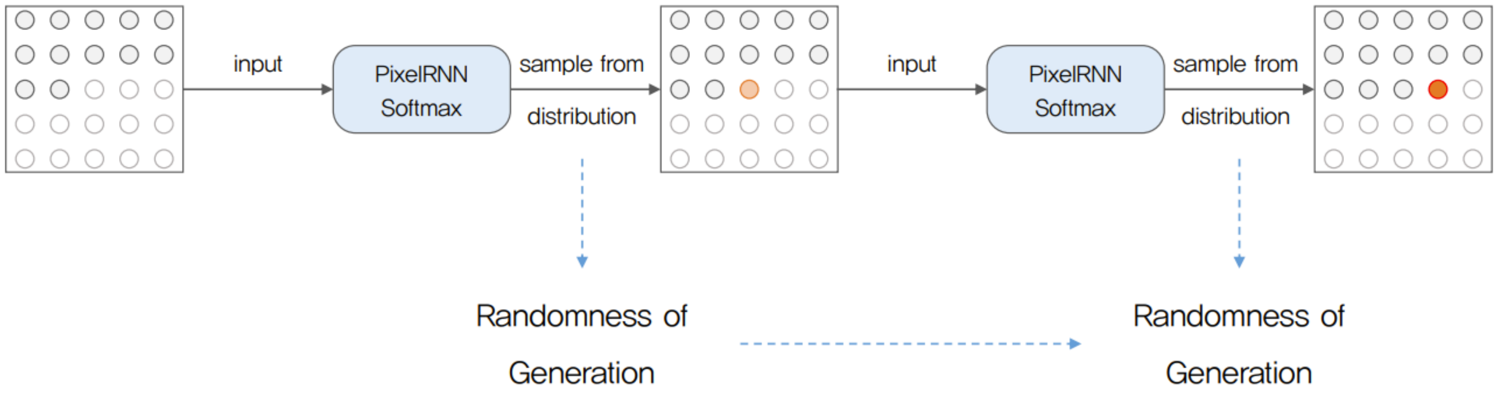
[Seminar - 고려대학교 DMQA 연구실 (korea.ac.kr)]

Unit 02 | Pixel RNN, Pixel CNN

Pixel RNN

- generate image pixels starting from corner
- model dependency on previous pixels using 2D LSTM
- pixel의 값을 0~255까지의 정수 값으로 하여 256개의 class를 갖는 soft max layer를 통해 생성한 conditional discrete distribution에서 sampling하여 inference

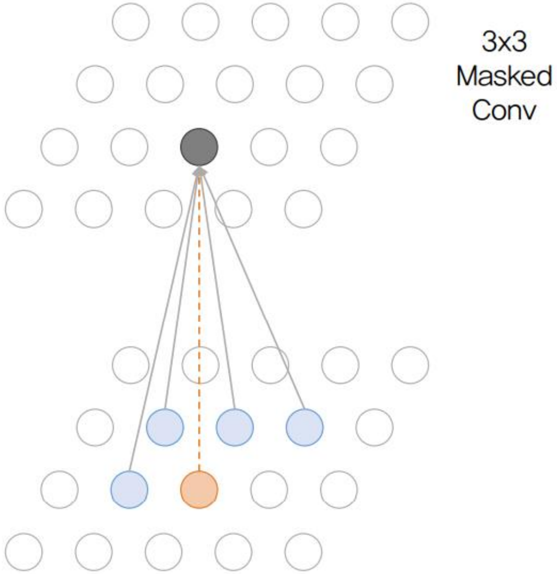


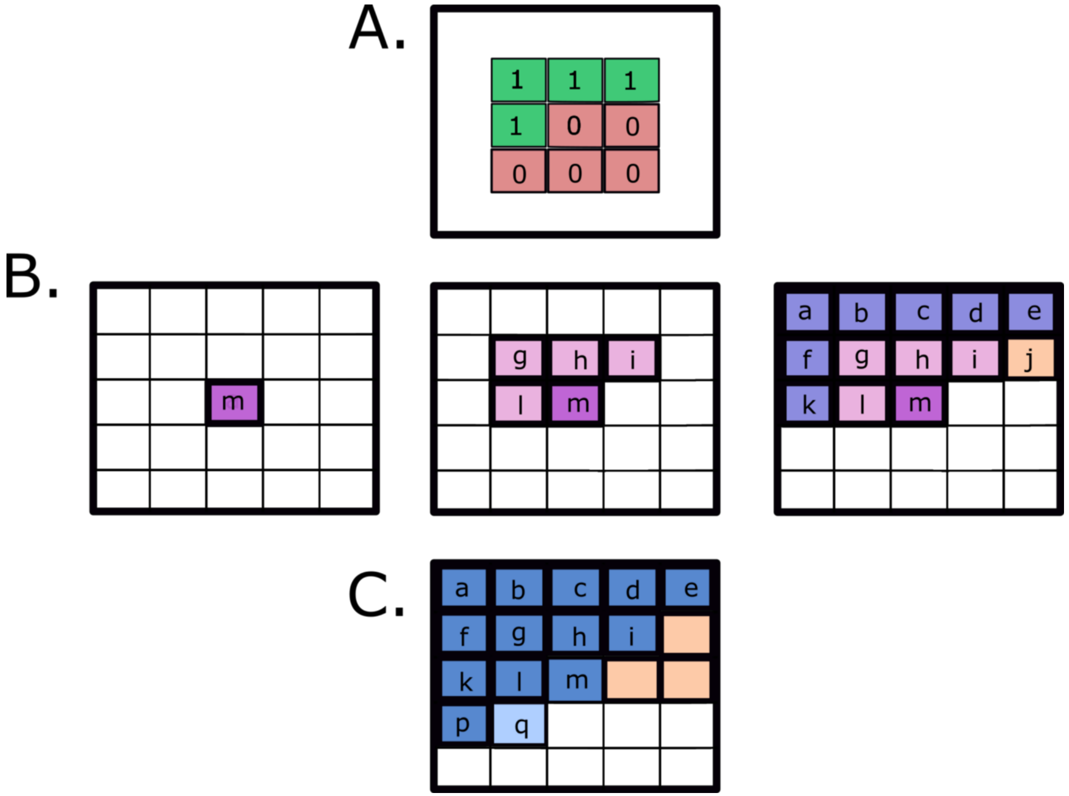


[Seminar - 고려대학교 DMQA 연구실 (korea.ac.kr)]

Pixel CNN

- autoregressive model
- RNN보다 적은 parameter수 + 병렬 처리 가능
- blind spot 문제 발생
- vertical stack+ horizontal stack으로 해결





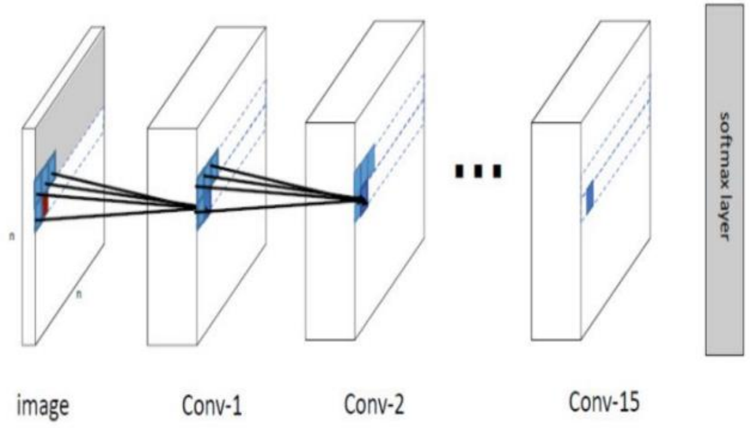
Seminar - 고려대학교 DMQA 연구실 (korea.ac.kr)

Unit 02 | Pixel RNN, Pixel CNN

Autoregressive Model

PixelCNN

- PixelRNN과 마찬가지로 코너에서부터 이전 픽셀에 대한 의존성을 바탕으로 특정 context 영역에 대해서만 CNN을 이용하게 된다.
- PixelRNN보다 속도는 빠르지만, log-likelihood가 나쁘다.

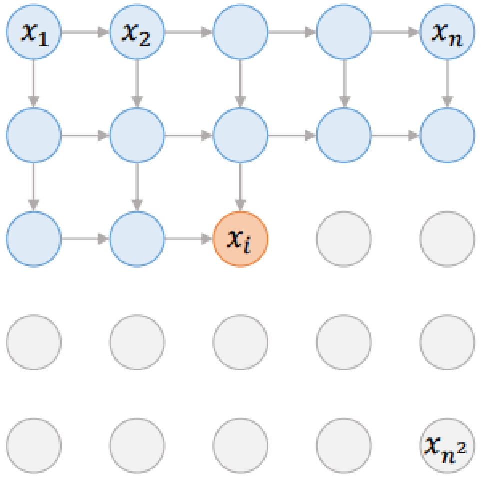


Seminar - 고려대학교 DMQA 연구실 (korea.ac.kr)

Autoregressive Model

PixelRNN

- 이전 픽셀에 대한 의존성을 바탕으로 RNN계열 모델을 이용한다.
- Receptive field를 늘리기 위해 Bi-LSTM을 사용할 수 있다.
- 속도가 굉장히 느리지만, log-likelihood가 좋다.



Seminar - 고려대학교 DMQA 연구실 (korea.ac.kr)

Unit 02 | Pixel RNN, Pixel CNN

Autoregressive Model

장점

1. 정의하기가 쉽다.
2. Likelihood p(x)를 명시적으로 계산할 수 있다.
3. Generation 과정이 쉽다.
4. 이미지, 오디오, 비디오 등에서 log-likelihood 결과가 좋다(좋은 샘플을 만들 수 있다).

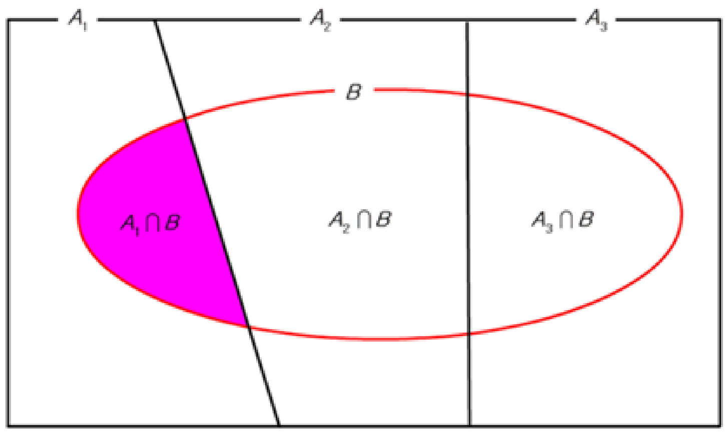
단점

1. 순서를 정의하는 방법에 의존적이다.
2. Generation 과정이 너무 오래 걸린다.
3. 바로 직후의 step만 예측할 수 있다.

Seminar - 고려대학교 DMQA 연구실 (korea.ac.kr)

Contents

Mathematics

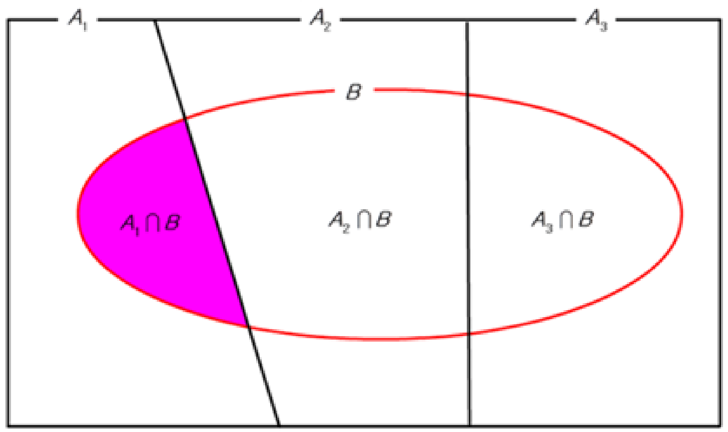


$$\boxed{P(A_i|B) = \frac{P(A_i \cap B)}{P(B)}} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j \cap B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j)\,P(B|A_j)}$$

1. 조건부확률의 정의

[Introduction to Bayesian Statistics (velog.io)](#)

Mathematics

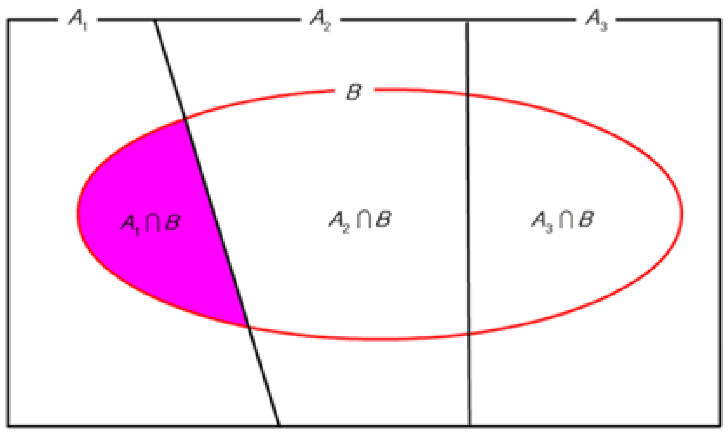


2. 확률의 곱셈법칙

$$P(A_i|B) = \boxed{\frac{P(A_i \cap B)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j \cap B)}} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j)\, P(B|A_j)}$$

[Introduction to Bayesian Statistics (velog.io)]

Mathematics



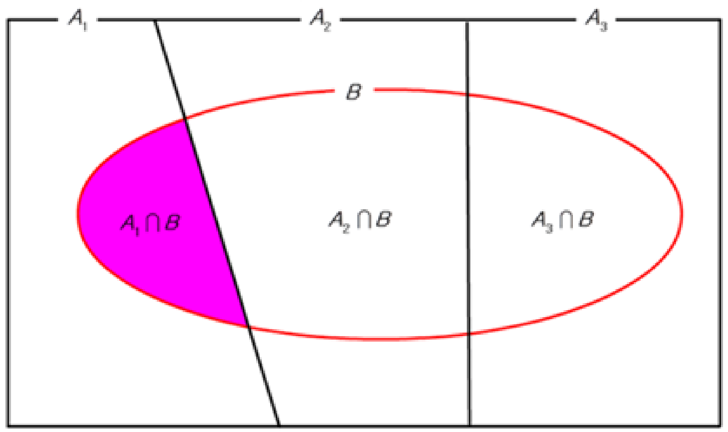
$$P(A_i|B) = \frac{P(A_i \cap B)}{\boxed{P(B)}} = \frac{P(A_i)P(B|A_i)}{\boxed{\sum_{j=1}^{k} P(A_j \cap B)}} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j)\,P(B|A_j)}$$

3. 주변확률분포의 합

Integral form : $P(B) = \int P(A,B)dA$

[Introduction to Bayesian Statistics (velog.io)](#)

Mathematics



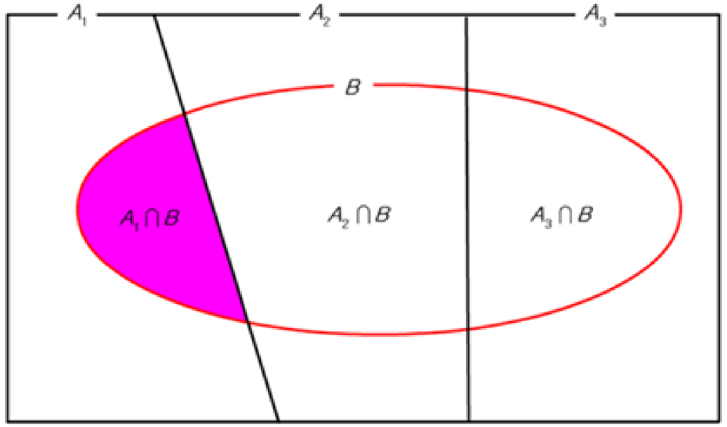
$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)} = \boxed{\frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j \cap B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j)\,P(B|A_j)}}$$

4. Re 확률의 곱셈법칙

Integral form : $P(B) = \int P(A,B)dA = \int P(A)P(B|A)dA$

Introduction to Bayesian Statistics (velog.io)

Mathematics



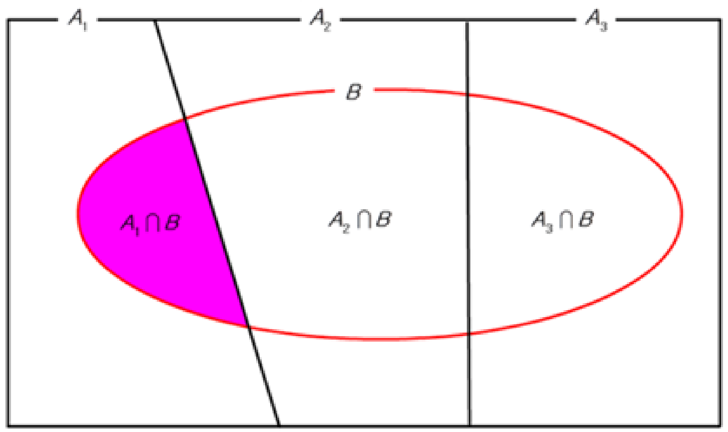
모른다　　　　　　　　　　　　　　　　　　　　　안다　안다

$$\underbrace{P(A_i|B)}_{\text{모른다}} = \frac{P(A_i \cap B)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j \cap B)} = \frac{\overbrace{P(A_i)}^{\text{안다}}\overbrace{P(B|A_i)}^{\text{안다}}}{\sum_{j=1}^{k} \underbrace{P(A_j)}_{\text{안다}}\,\underbrace{P(B|A_j)}_{\text{안다}}}$$

[Introduction to Bayesian Statistics (velog.io)]

Mathematics

$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j \cap B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j)\, P(B|A_j)}$$



posterior : 사후 분포

prior : 사전분포

$$\underbrace{p(\theta|x)}_{\text{posterior}} = \frac{p(\theta)p(x|\theta)}{p(x)} \propto \underbrace{p(\theta)}_{\text{prior}}\underbrace{p(x|\theta)}_{\text{Likelihood}}$$

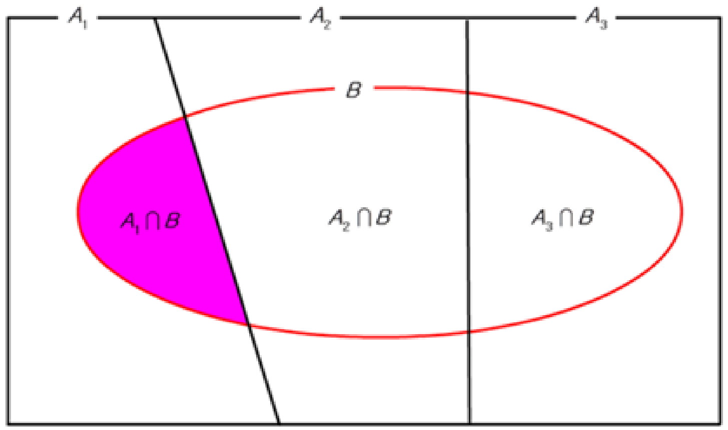
parameter data(sample)

Likelihood : 가능도, 우도
data : 자료(샘플)

Introduction to Bayesian Statistics (velog.io)

Mathematics

$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j \cap B)} = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{k} P(A_j)\, P(B|A_j)}$$



posterior
사후 분포

$$\overbrace{p(\underbrace{\theta}_{\text{parameter}}|\underbrace{x}_{\text{data(sample)}})}^{\text{posterior}} \rightarrow \frac{\overbrace{p(\theta)}^{\text{prior}}\overbrace{p(x|\theta)}^{\text{data}}}{\underbrace{\cancel{p(x)}}_{\text{상수}}}$$

prior
사전분포

data
자료(샘플)

사후 분포(posterior)는 알 수 없으니,
베이즈 정리를 이용해 사전분포와 샘플로 대신 구하자!

[Introduction to Bayesian Statistics (velog.io)]()

Mathematics 2

고전적 추론 – MLE(최대우도추정)

$$\hat{\theta} := \underset{\theta}{\operatorname{argmax}}\, \underline{L(\theta|x)}$$

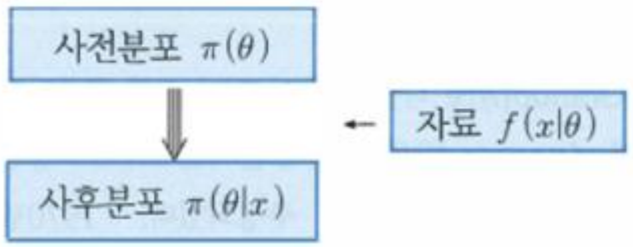
Posterior(사후분포)

$$\Rightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \underline{f(x|\theta)}$$

Likelihood(가능도, 우도)

베이지안 추론

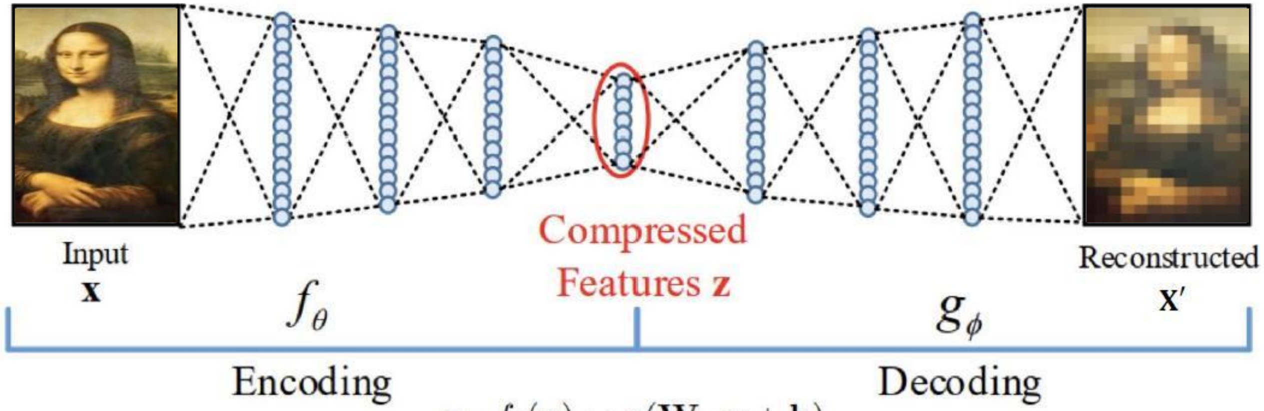
$$\underbrace{p(\theta|x)}_{\text{posterior}} = \frac{p(\theta)p(x|\theta)}{p(x)} \propto \underbrace{p(\theta)p(x|\theta)}_{\text{prior} \quad \text{likelihood}}$$



[Introduction to Bayesian Statistics (velog.io)]

Autoencoder

AutoEncoder

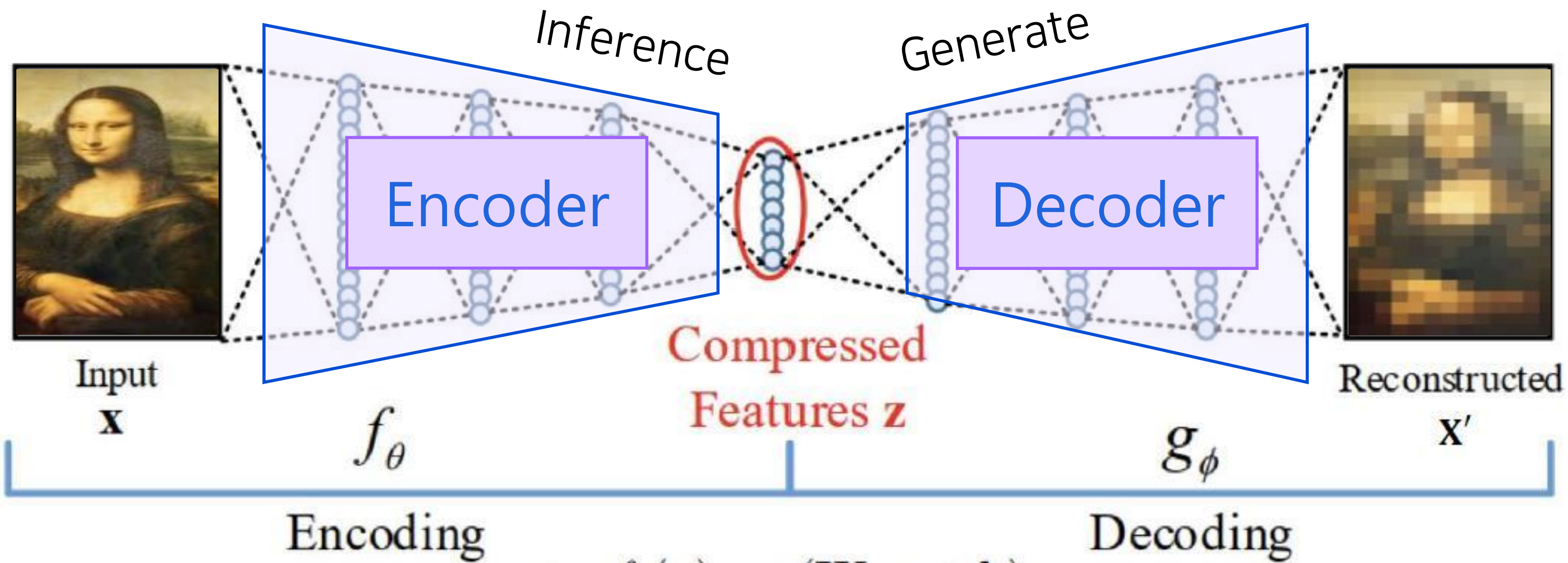


$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W}\cdot\mathbf{x}+\mathbf{b})$$

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

z = latent vector / latent feature / representation / embedding / code / bottle neck

Autoencoder

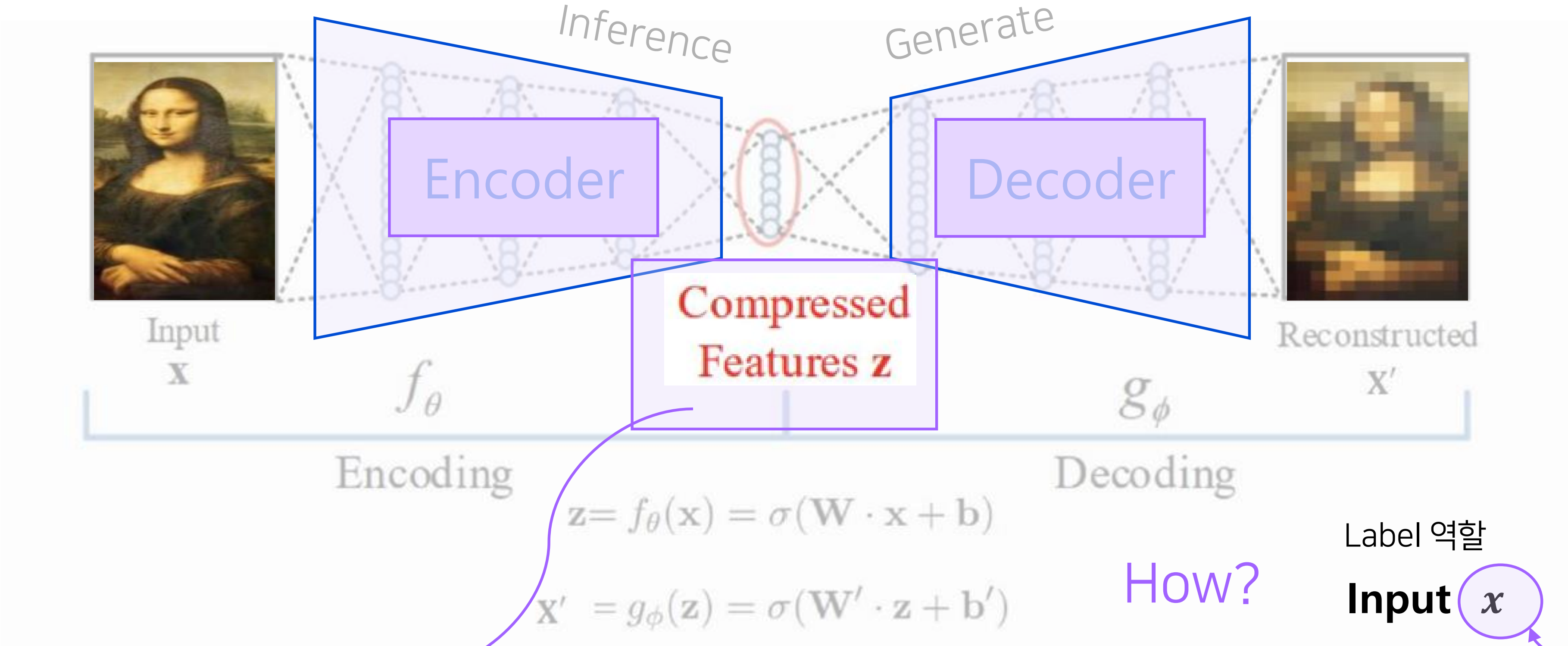


$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$$

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

z = latent vector / latent feature / representation / embedding / code / bottle neck

Autoencoder



Objective

잠재벡터 z 찾기

How?

x와 x'의 차이를 최소화

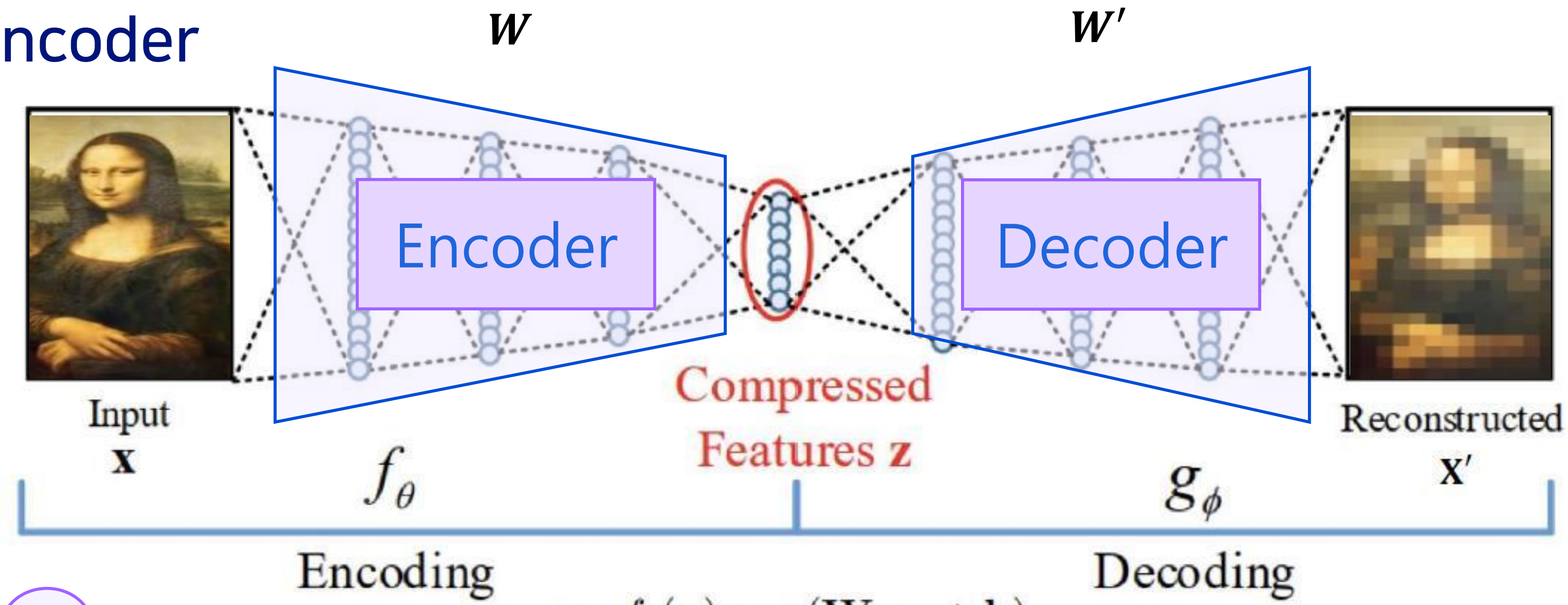
Label 역할

Input x

$\mathcal{L}$: loss

Reconstructed x'

Autoencoder



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

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

How?

X와 x'의 차이를 최소화

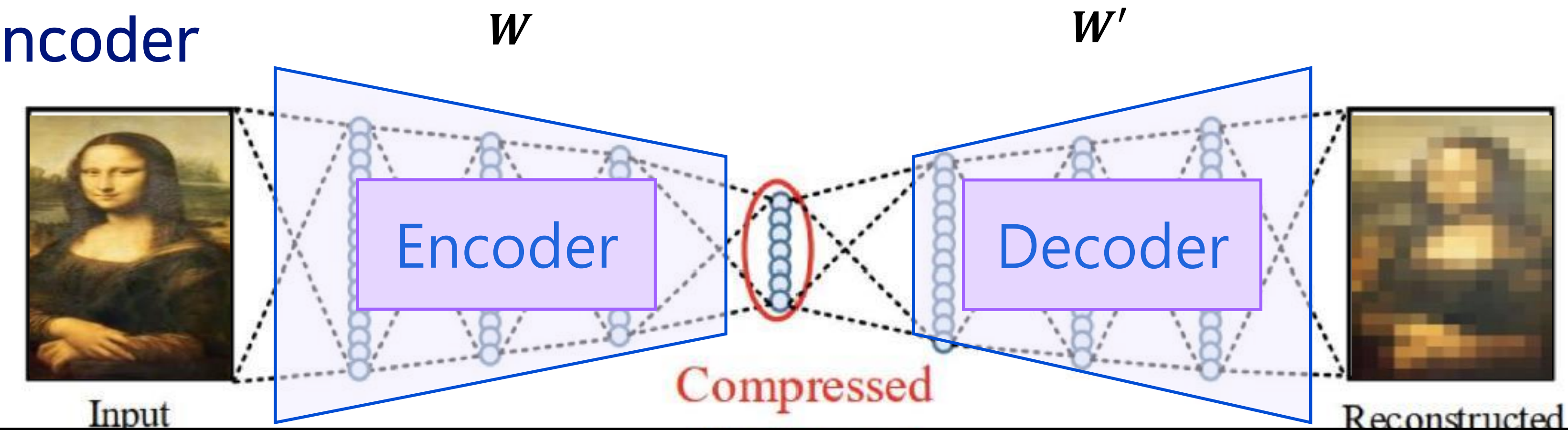
Input x

Reconstructed x'

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

$$\therefore \quad \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{W}', \boldsymbol{b}'}{\operatorname{argmin}} \mathcal{L} \quad \blacksquare$$

Autoencoder



그러면, 잠재벡터 z를 구하기 위한

Loss함수 $\mathcal{L}$는 어떻게 정해야 할까? 각자 생각해보자.

Variational Autoencoder

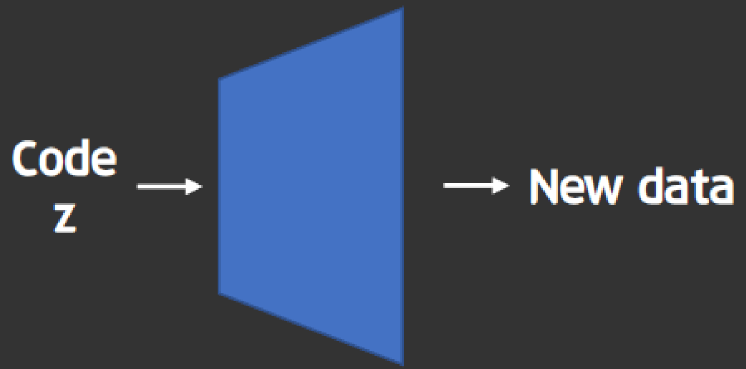
AutoEncoder

Autoencoder => z 찾기

Encoder를 사용하기 위해 Decoder도 학습

Variational Autoencoder

AutoEncoder



Variational Autoencoder => 생성모델

Decoder를 위해 Encoder도 학습

Autoencoder

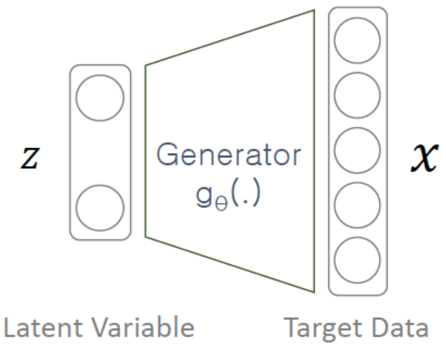
Objective

- 어떤 데이터(x)를 잘 압축하자
  (특징을 잘 뽑아내자)
- 어떻게? 입력 데이터(x)를 잘 복원시켜서

Keywords

- Unsupervised Learning
- Representation Learning
- Dimension Reduction

Variational Autoencoder



$z \sim p(z)$ Random variable

$g_\theta(\cdot)$ Deterministic function parameterized by θ

$x = g_\theta(z)$ Random variable

생성 모델링의 목적 : P_{model}을 P_{data}에 가깝게!

배경 1 - $p(x) = \int p(x, z) dz$

배경 2 - $p(x, z) = p(x|z)p(z)$

use latent variable

$z \sim p(z)$일 때, $p(x)$를 최대화 해야 한다. 이 때,

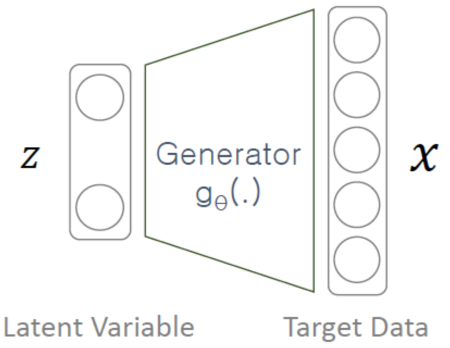
$$p(x)$$
$$= \int p(x, z) dz$$
$$= \int p(x|z) p(z) dz$$

problem 1 : how to define latent variable z
problem 2 : how to define p(x|z)
problem 3 : integral is not tractable

$\mathbb{E}_{z \sim p(z)}[p(x|z)]$ → **최대화!!**

http://arxiv.org/pdf/1606.05908

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

http://arxiv.org/pdf/1606.05908

생성 모델링의 목적 : P_{model}을 P_{data}에 가깝게!

배경 1 - $p(x) = \int p(x,z)dz$

배경 2 - $p(x,z) = p(x|z)p(z)$

use latent variable

$z \sim p(z)$일 때, $p(x)$를 최대화 해야 한다. 이 때,

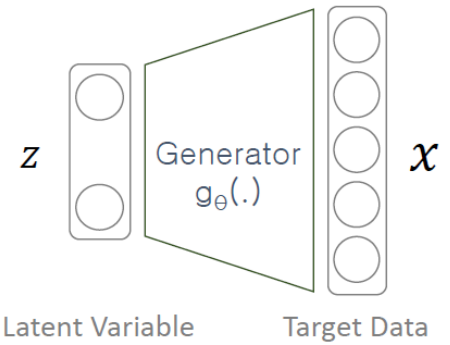
$$p(x) = \int p(x,z)dz = \int p(x|z)p(z)dz$$

problem 1 : how to define latent variable z
problem 2 : how to define p(x|z)
problem 3 : integral is not tractable

problem 1 : how to define latent variable z
-> assume simple prior N(0,I)

Variational Autoencoder

생성 모델링의 목적 : P_{model}을 P_{data}에 가깝게!



$\boxed{z \sim p(z)}$ Random variable

$g_\theta(\cdot)$ Deterministic function parameterized by θ

$x = g_\theta(z)$ Random variable

Gaussian distribution 가정

배경 1 - $p(x) = \int p(x, z)dz$

배경 2 - $p(x, z) = p(x|z)p(z)$

use latent variable

$z \sim p(z)$일 때, $p(x)$를 최대화 해야 한다. 이 때,

$p(x)$

$= \int p(x, z)dz$

$= \int p(x|z)p(z)dz$

problem 1 : how to define latent variable z
problem 2 : how to define p(x|z)
problem 3 : integral is not tractable

problem 2 : how to define p(x|z)
-> using decoder neural net

http://arxiv.org/pdf/1606.05908

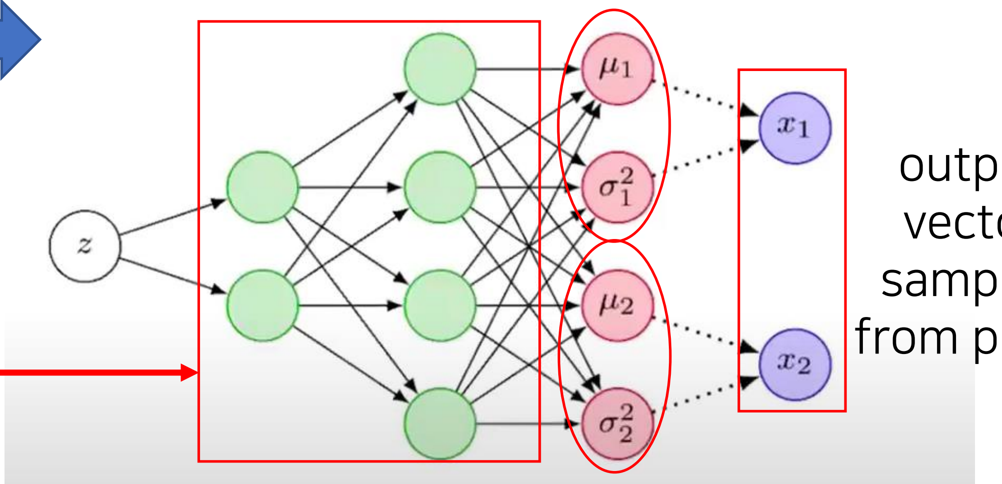
Variational Autoencoder

- at training time
  - likelilhood of each input x can be computed using p(x|z)(decoder net)
  - backprop
- after training
  - a new x is sampled from p(x|z)(decoder net)

suppose p(x|z) = N(x | mu(z), ∑(z))

x ∈ R^2, z ∈ R

mean&var of 1st dim



output vector sampled from p(x|z)

decoder

mean&var of 2nd dim

data(x)가 따르는 분포의 parameter를 주는 것이 decoder의 역할

http://arxiv.org/pdf/1606.05908

Variational Autoencoder

생성 모델링의 목적 : P_{model}을 P_{data}에 가깝게!

배경 1 - $p(x) = \int p(x,z)dz$ 배경 2 - $p(x,z) = p(x|z)p(z)$

use latent variable

$$p_{\theta}(\boldsymbol{x}) = \int \underbrace{p(\boldsymbol{z})}_{\text{simple Gaussian prior}} \underbrace{p_{\theta}(\boldsymbol{x} \mid \boldsymbol{z})}_{\text{learn with neural net "decoder"}} dz$$

$z \sim p(z)$일 때, $p(x)$를 최대화 해야 한다. 이 때,

$p(x)$

$= \int p(x,z)dz$

$= \int p(x|z)p(z)dz$

problem 1 : how to define latent variable z
problem 2 : how to define p(x|z)
problem 3 : integral is not tractable

$$p(\boldsymbol{x}) \approx \frac{1}{n}\sum_{i=1}^{n} p(\boldsymbol{x} \mid \boldsymbol{z}_i)$$

problem 3 : integral is not tractable
-> monte carlo integration

http://arxiv.org/pdf/1606.05908

Variational Autoencoder

problem 3 : integral is not tractable
-> monte carlo integration

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int \underbrace{p(\boldsymbol{z})}_{\substack{\text{simple}\\\text{Gaussian}\\\text{prior}}} \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})}_{\substack{\text{learn with}\\\text{neural net}\\\text{“decoder”}}} d\boldsymbol{z}$$

$$p(\boldsymbol{x}) \approx \frac{1}{n} \sum_{i=1}^{n} p(\boldsymbol{x} \mid \boldsymbol{z}_i)$$

그러나 고차원의 분포일수록
p(x|z) -> 0 for most z
contribute not to integration

따라서 실제로 데이터 x가 분포하고 있을 만한 z에 대해서만 integration을 진행하자

need new function p(z|x)
but p(z|x)은 알 수 없다
q(z|x)를 통해 approximation을 진행하자(variational inference)
-> q(z|x)가 encoder

http://arxiv.org/pdf/1606.05908

Variational Autoencoder

단순히 p(z)~N(0, I)에서 sampling하여 training하는 것은 안되는 것일까?

$p(x) \approx \Sigma p(x|z)p(z)$ <- mc integration

Input Loss ↓↓ Loss ↑↑

(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.

단순 MSE loss로 비교
있음직 하지 않은 sample 생성

[1606.05908.pdf (arxiv.org)](https://arxiv.org/pdf/1606.05908.pdf)

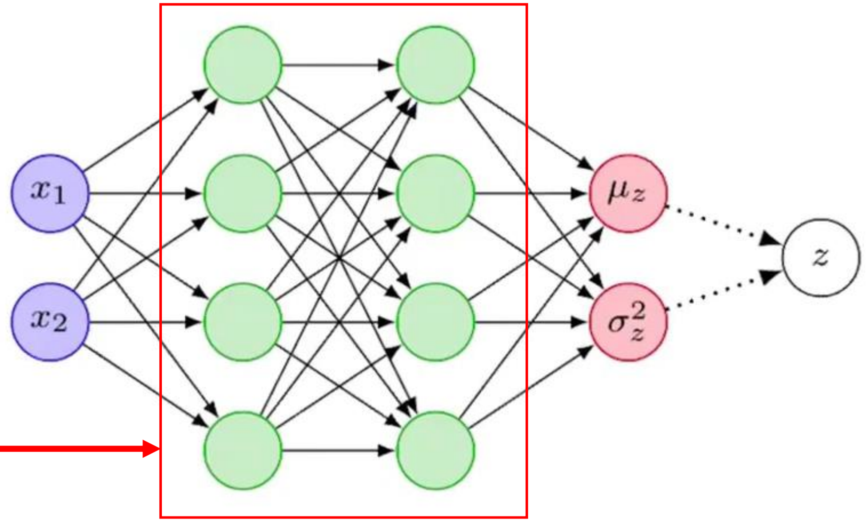
Variational Autoencoder

suppose q(z|x) = N(z | mu(x), ∑(x))

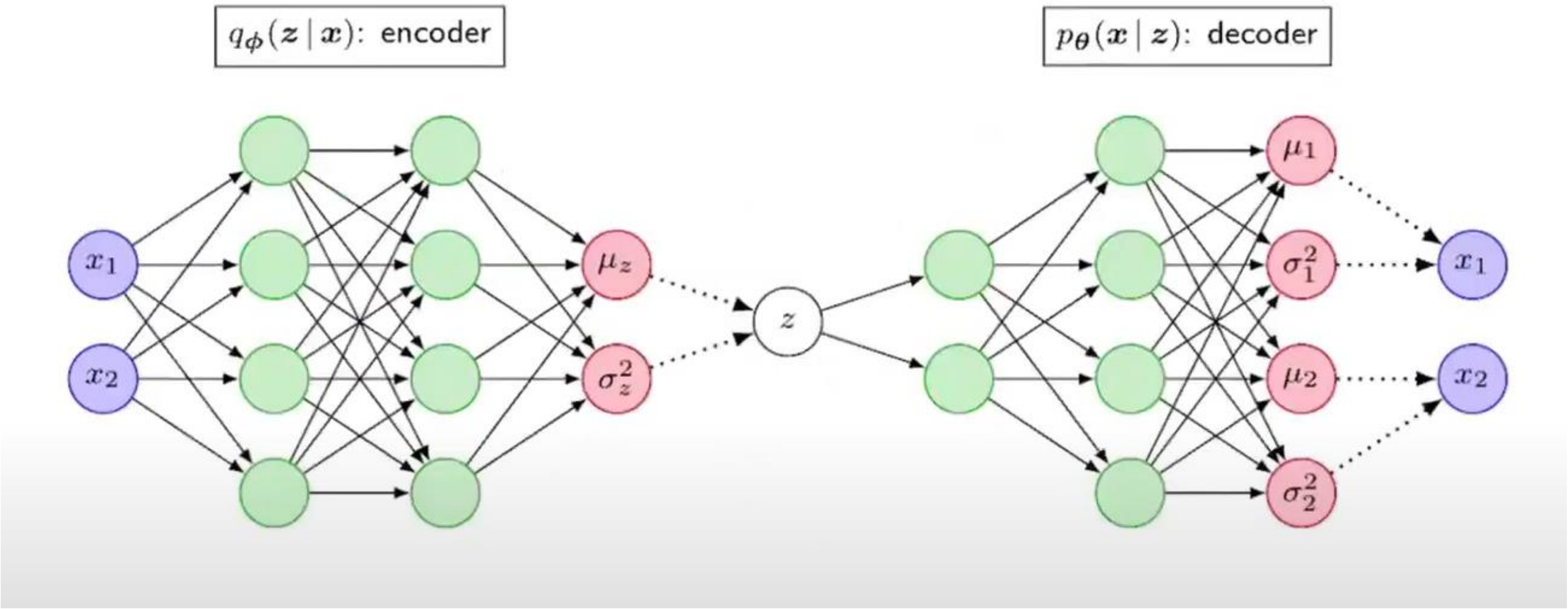
x ∈ R^2, z ∈ R

p(z|x)의 분포를 q(z|x)의 분포로 approximate 이때, q(z|x)를 normal로 assume

encoder



Variational Autoencoder



Variational Autoencoder

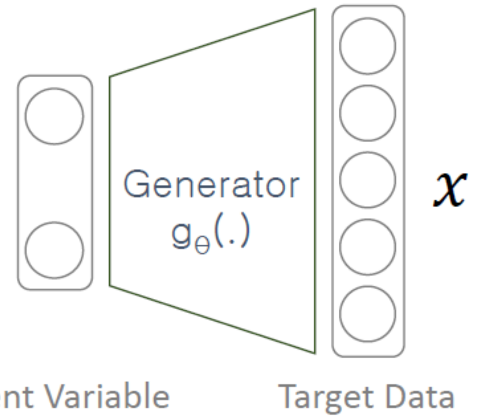
$\int p(x|g_\theta(z))p(z)dz = p(x)$

Prior : $p(z)$ → Gaussian distribution

- z를 **사전 정규분포에서 샘플링하는 것** 대신 input x와 유의미하게 유사한 샘플이 나올 수 있는 **사후확률분포 $p(z|x)$로부터 샘플링**을 진행하자.

true posterior

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



- 단, $p(z|x)$가 무엇인지 알지 못하므로, 우리가 알고 있는 확률분포 중 하나인 $q_\phi(z|x)$를 택해서 그것의 파라미터 값을 조정해 $p(z|x)$와 유사하게 만들어 본다(**variational inference**).
- 따라서, VAE는 **true posterior $p(z|x)$와 유사한 분포 $q_\phi(z|x)$ 를** 만들게 해주는 ϕ와 생성자(generator / decoder)의 θ를 찾는 문제이다.

Gaussian distribution으로 가정
즉, $\phi : mean, std$

Variational Autoencoder

Variational Inference

- True posterior P를 추론할 때, 모델 Q_ϕ를 가지고 추론하되, 파라미터를 잘 조정해서 Q_ϕ를 P에 최대한 가깝게 만드는 것!

- 이 때, **두 분포 간의 "거리"**를 정의하기 위해 KL-divergence를 기본적으로 사용한다.

Kullback-Leibler divergence(KL-D)

- 두 확률분포의 차이를 계산하는 데 사용하는 함수로, 어떤 이상적인 분포(P)에 대해, 그 분포를 근사하는 다른 분포(Q)를 사용해 샘플링을 할 경우 발생하는 **정보 엔트로피 차이**를 계산한다.

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

KL 발산과 Cross entropy?

Kullback-Leibler divergence(KL-D)

- 두 확률분포의 차이를 계산하는 데 사용하는 함수로, 어떤 이상적인 분포(P)에 대해, 그 분포를 근사하는 다른 분포(Q)를 사용해 샘플링을 할 경우 발생하는 **정보 엔트로피 차이**를 계산한다.

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

비대칭적 → 거리 개념에 적합하지 않다.

$$
\begin{aligned}
&D_{KL}(P||Q)\\
&= \int p(x)\log\frac{p(x)}{q(x)}dx\\
&= \int p(x)\log(p(x) - \int p(x)\log q(x)dx\\
&= \underbrace{-H(P)}_{\text{정보 엔트로피}} + \underbrace{H(P,Q)}_{\text{크로스 엔트로피}}
\end{aligned}
$$

정보 이론 2편: KL-Divergence (brunch.co.kr)

Variational Autoencoder

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

Aim :
$maximize\ \ p(x)$

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

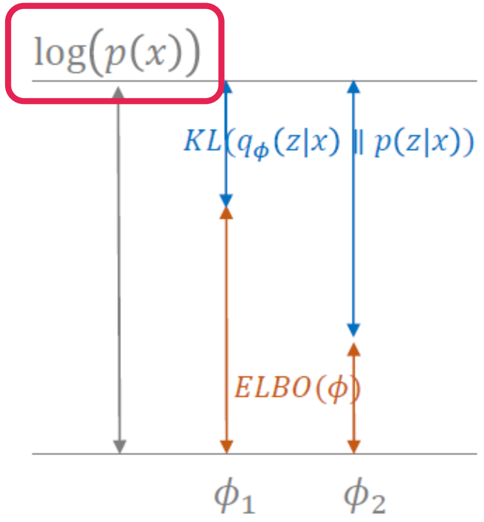
$$= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x) dz \quad \leftarrow p(x) = \frac{p(x,z)}{p(z|x)}$$

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

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

(Evidence LowerBOund)

두 확률분포 간의 거리≥0



want to minimize this term..

추정 불가능

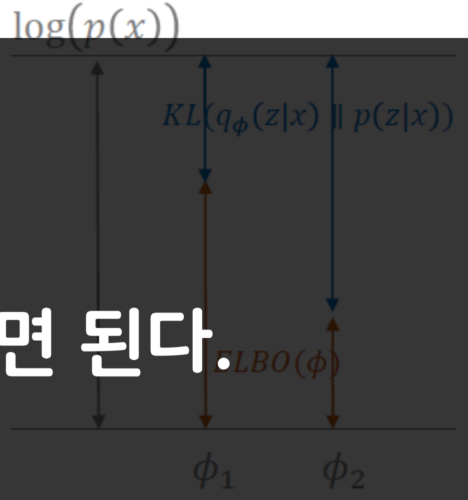
$$\log(p(x)) = ELBO(\phi) + KL(q_\phi(z|x)||p(z|x)) \geq ELBO(\phi)$$

Variational Autoencoder

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

$$\underbrace{\log(p(x))}_{\text{고정}} = \int \log(p(x))q_\phi(z|x)dz \quad \leftarrow \int q_\phi(z|x)dz = 1$$

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



KL 발산을 최소화하는 $q_\phi(z|x)$의 ϕ값을 찾으면 된다.

이 때, 우리는 $p(z|x)$를 모른다.

그래서, KL 발산을 최소화하는 것 대신 $ELBO(\phi)$를 최대화하는 ϕ값을 찾는다.

두 확률분포 간의 거리≥0

Variational Inference

Variational Autoencoder

Parameter(ϕ, θ)를 찾자.

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

최대화 최소화(모른다)

$$\log(p(x)) = \underbrace{ELBO(\phi)}_{\text{최대화}} + \underbrace{KL(q_\phi(z|x)|p(z|x))}_{\text{최소화(모른다)}}$$

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

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

latent z에서 추출

Variational Autoencoder

Parameter(ϕ, θ)를 찾자.

Optimization Problem 1 on ϕ: Variational Inference

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

Optimization Problem 2 on θ: Maximum likelihood

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

Final Optimization Problem

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

Loss funtion

Variational Autoencoder

Parameter(ϕ, θ)를 찾자.

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

원 데이터에 대한 likelihood

Variational inference를 위한
approximation class 중 선택

다루기 쉬운 확률 분포 중 선택

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

Reconstruction Error

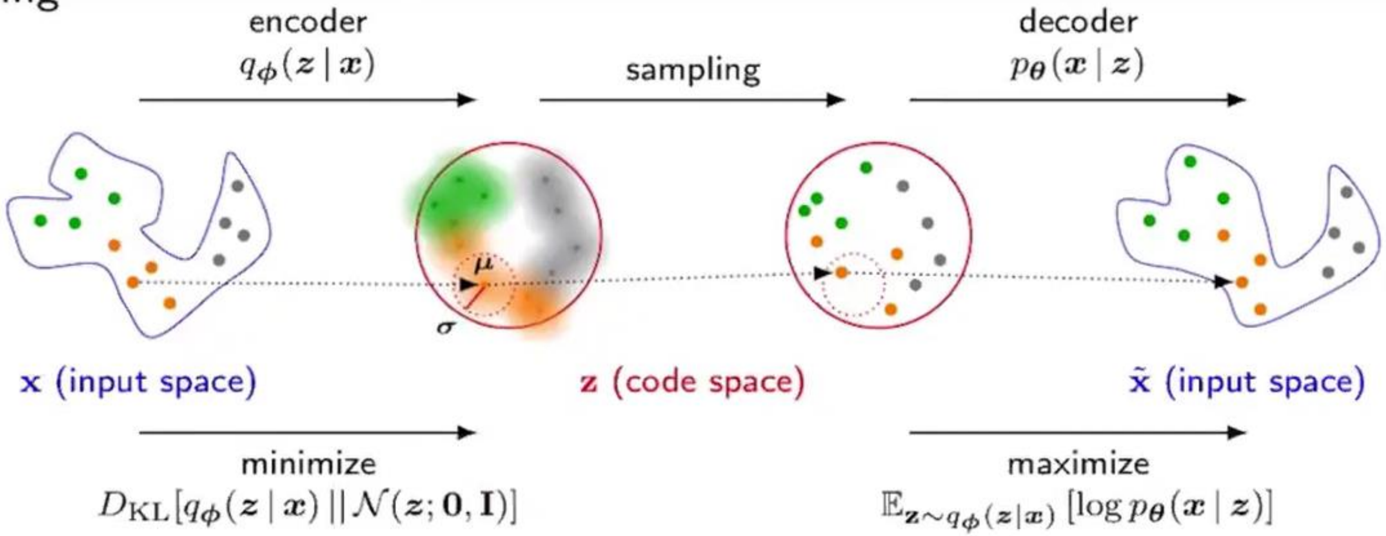
- 현재 샘플링 용 함수에 대한 negative log likelihood
- x_i에 대한 복원 오차 (AutoEncoder 관점)

Regularization

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

Variational Autoencoder

- training



z sampled from

- $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ in training
- $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ after training

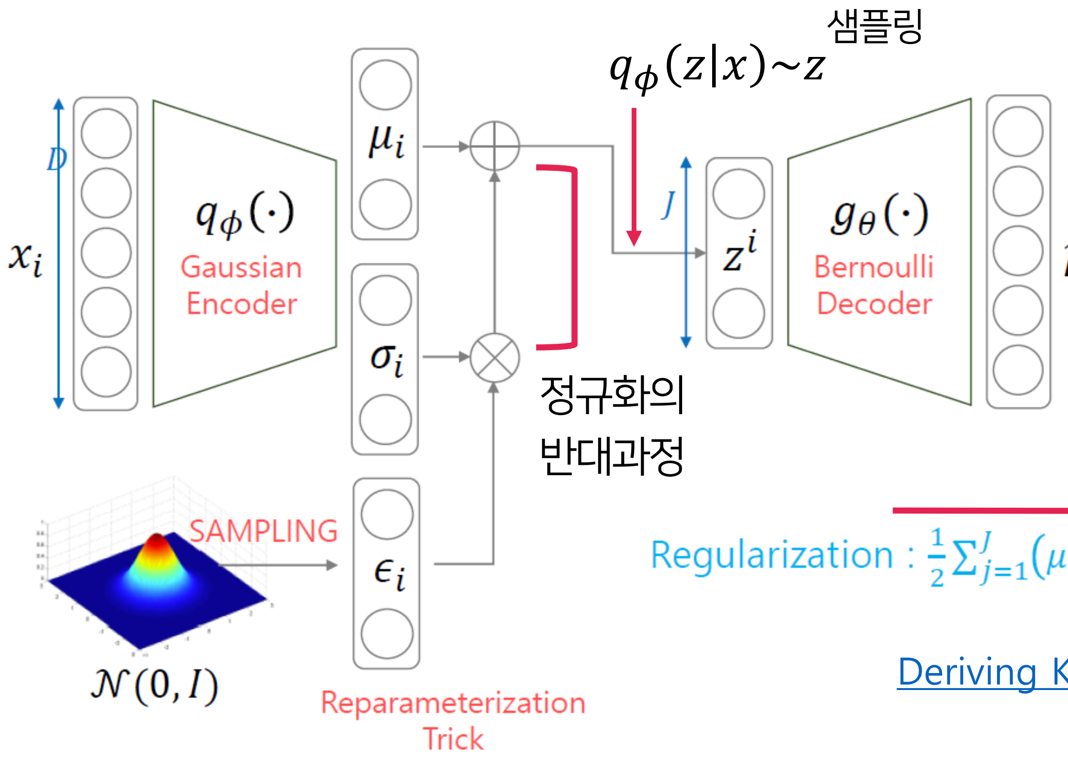
$\Rightarrow$ and then the sampled $\mathbf{z}$ is fed into decoder to generate $\mathbf{x}$

Variational Autoencoder

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

Reconstruction Error: $-\sum_{j=1}^{D} x_{i,j}\log p_{i,j} + (1-x_{i,j})\log\left(1-p_{i,j}\right)$

For backprop



KL-발산 계산식→ Thanks to Gaussian assumption

Regularization : $\frac{1}{2}\sum_{j=1}^{J}\left(\mu_{i,j}^2 + \sigma_{i,j}^2 - \ln\left(\sigma_{i,j}^2\right) - 1\right)$

[[논문] VAE(Auto-Encoding Variational Bayes) 직관적 이해 - Taeu]

[Deriving KL Divergence for Gaussians (leenashekhar.github.io)]

Variational Autoencoder

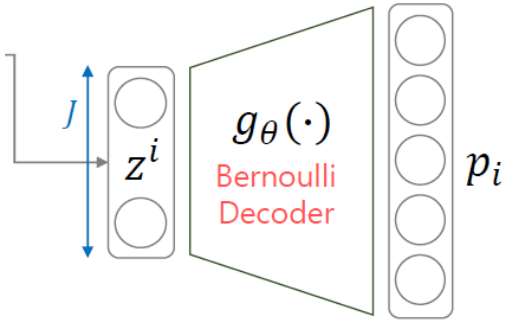
정리

1. MLE 사용 → 유의미한 샘플 생성 x → Posterior를 모름 → Variational Inference로 해결
2. Posterior를 모름 → KLD 계산 불가 → ELBO를 최대화
3. ELBO를 최대화 → Reconstruction 과 regularization을 최적화 → parameter θ, ϕ 탐색(By Training)
4. ϕ를 이용해 샘플링한 latent vector z를 θ를 이용한 generator(decoder) $g_\theta(z)$를 이용해 생성

Variational Autoencoder

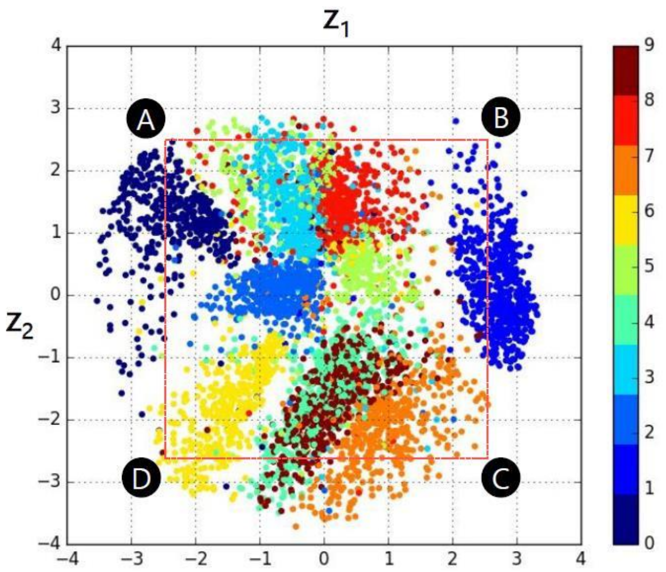
In 2D manifold

$(J = 2)$

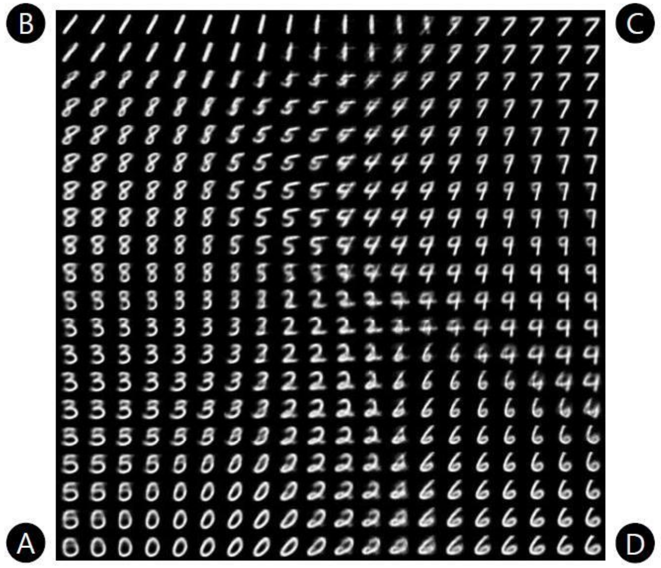


$z_1 \sim N(\mu_1, \sigma_1^2)$
$z_2 \sim N(\mu_2, \sigma_2^2)$

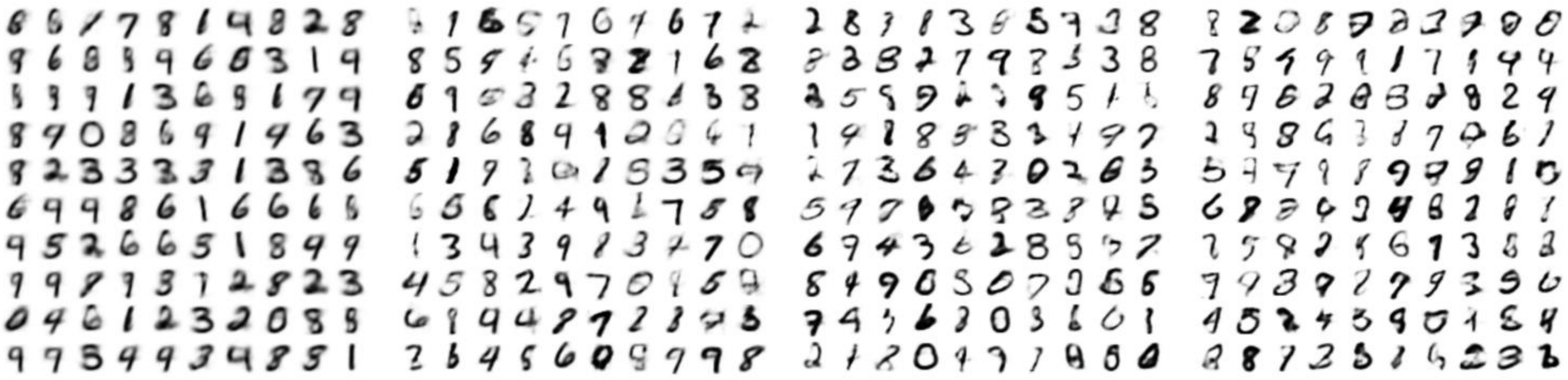
Latent



Uniform



Variational Autoencoder



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

1312.6114.pdf (arxiv.org)

Variational Autoencoder

장점

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

단점

- Likelihood의 lower bound를 최대로 하는 방식이 autoregressive model보다 결과가 안 좋음.
- 생성된 결과가 Blur가 많고 질이 떨어짐

→ GAN

Contents

Generative Adversarial Network?

생성적	적대	신경망
Generative	Adversarial	Network

Generative Adversarial Network?

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 Network?

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…

Generative Adversarial Network?

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 Network?

VS

판별자 D (Discriminator)

"진짜 화폐랑 G가 만들어내는 가짜 화폐를
잘 구별해야징!!"

생성자 G (Generator)

"진짜 화폐랑 내가 만들어낸 가짜 화폐를
D가 구분할 수 없도록
완벽하게 진짜같은 가짜 화폐를 만들어낼거얌!"

Generative Adversarial Network?

Real x

→

판별자 D (Discriminator)

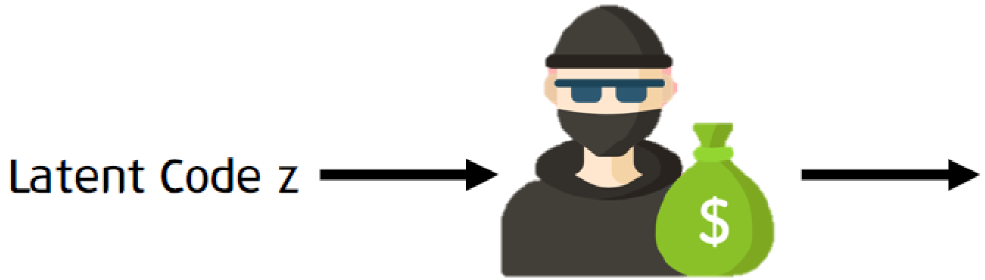
→

D(x)

Sigmoid(내판단) = 0.95정도로
저건 진짜 돈일 것같아

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

Generative Adversarial Network?



생성자 G (Generator)

"최고의 위조지폐범이 될테야!"

Generative Adversarial Network?

Latent Code z →

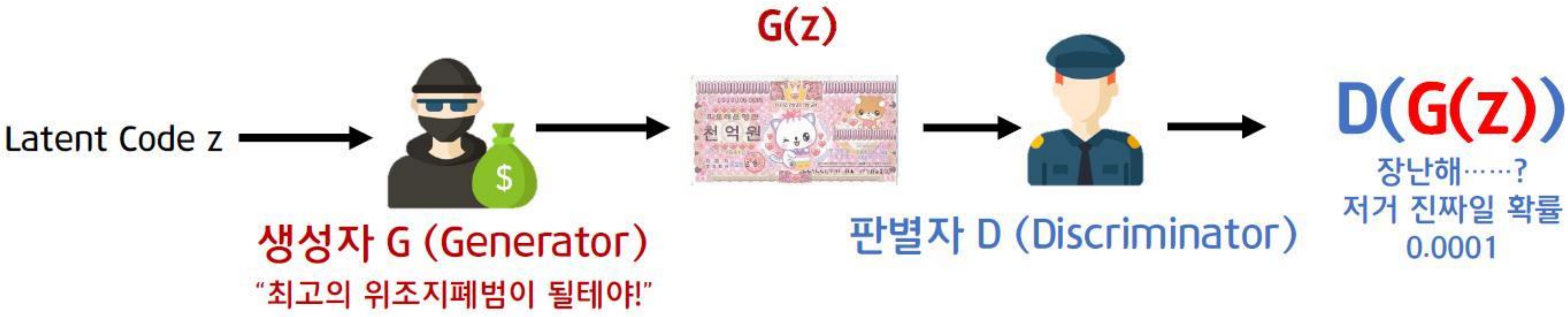
→

생성자 G (Generator)

"최고의 위조지폐범이 될테야!"

G(z)

Generative Adversarial Network?



Generative Adversarial Network?

Latent Code z

생성자 G (Generator)

"최고의 위조지폐범이 될테야!"

Generative Adversarial Network?

Latent Code z →

→

생성자 G (Generator)

"최고의 위조지폐범이 될테야!"

G(z)

Generative Adversarial Network?

Vanilla GAN : MLP
DC GAN : Transposed CNN..

Vanilla GAN : using MLP
DC GAN : using CNN..

Generative Adversarial Network?

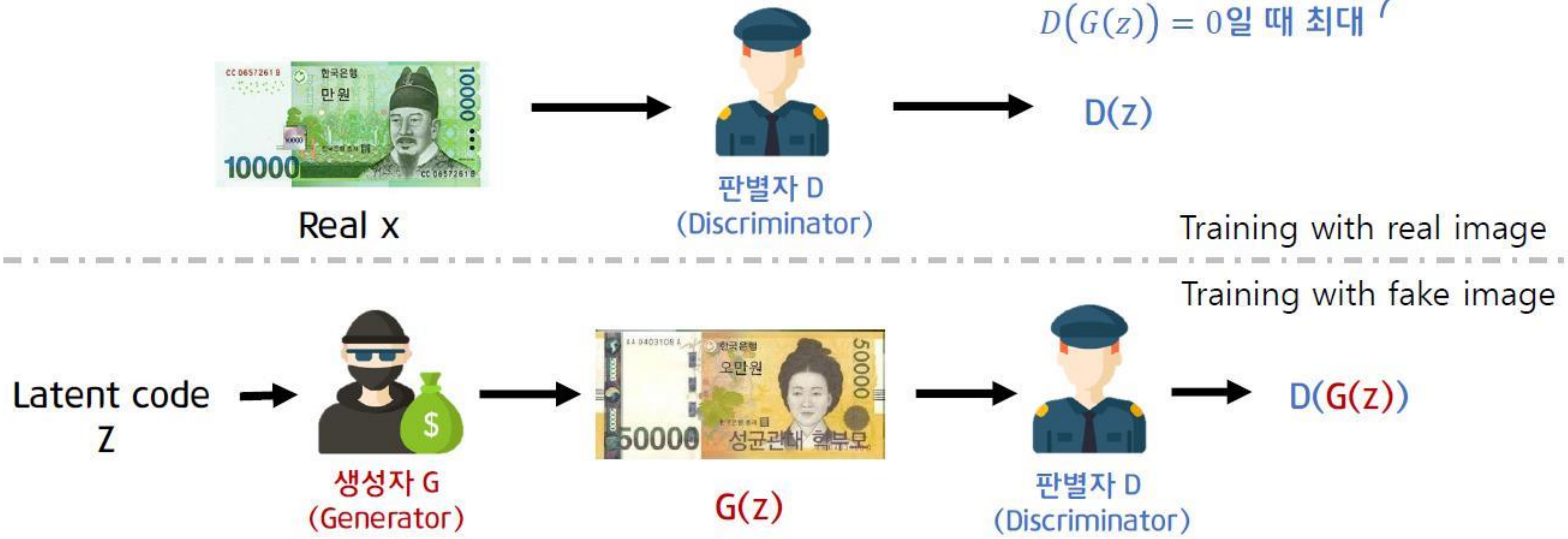
	생성자 G (Generator)	판별자 D (Discriminator)
입력	랜덤한 숫자로 구성된 벡터 z	1. 훈련 데이터셋에 있는 실제 샘플 x 2. 생성자가 만든 가짜 샘플 z
출력	최대한 진짜 같이 보이는 가짜 샘플	입력 샘플이 진짜일 예측 확률
목표	훈련 데이터셋에 있는 샘플 x과 구별이 불가능한 가짜 샘플 $G(z)$ 생성하기	생성자가 만든 가짜 샘플 $G(z)$과 훈련 데이터셋의 진짜 샘플 x 구별하기

Generative Adversarial Network?

Lei Mao's Log Book – Minmax Game for Training Generative Adversarial Networks – min max from cross entropy

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

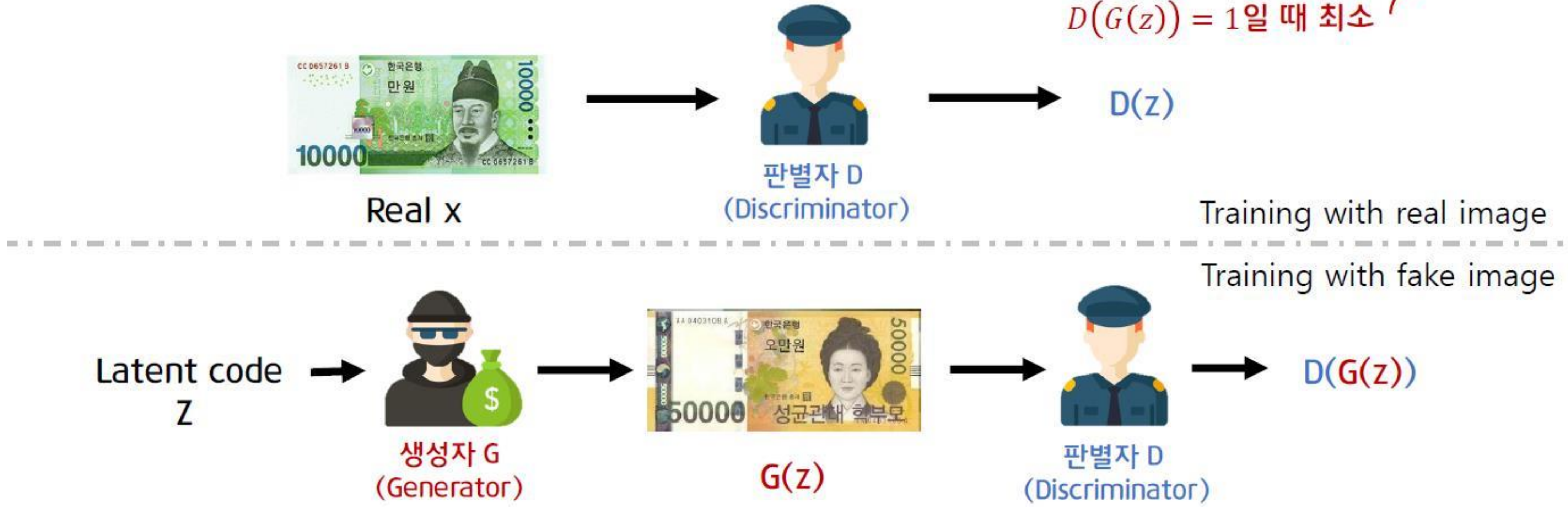
$D(G(z)) = 0$일 때 최대



Generative Adversarial Network?

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

$D(G(z)) = 1$일 때 최소



Generative Adversarial Network?

실제 데이터 분포에서 온 Sample x

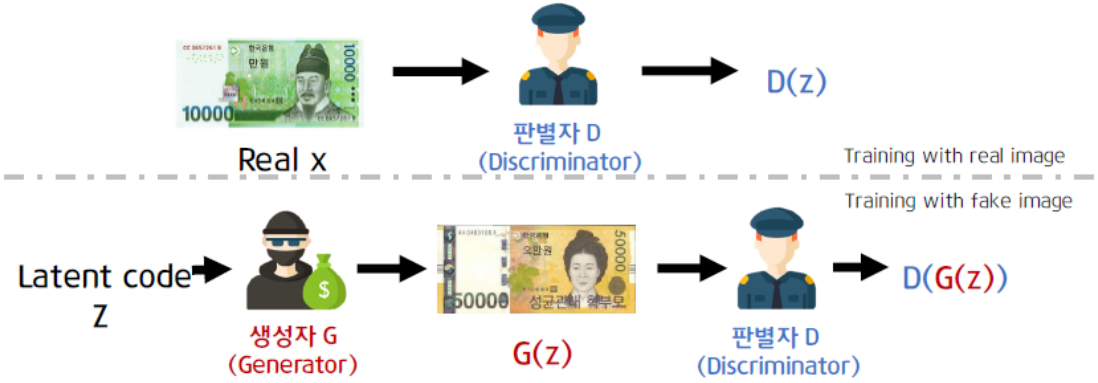
Gaussian 분포(예)에서 온 Sample latent code z

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

$D(x) = 1$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()
```

[Lei Mao's Log Book – Minmax Game for Training Generative Adversarial Networks]

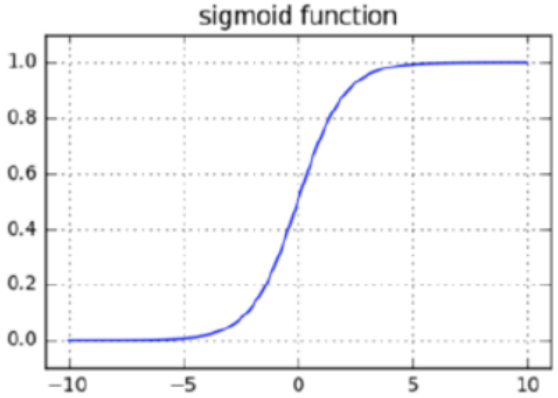
Generative Adversarial Network?

1. 모델이 진동하며, 기울기 소실(Vanishing Gradient)가 자주 일어난다.
2. 하이퍼 파라미터가 많고 민감하다.
3. 모드 붕괴(Mode collapse)를 잘 다루어야 한다(How to identify Failure mode?).
4. 평가지표가 불충분하다(Empirical decision may be needed).

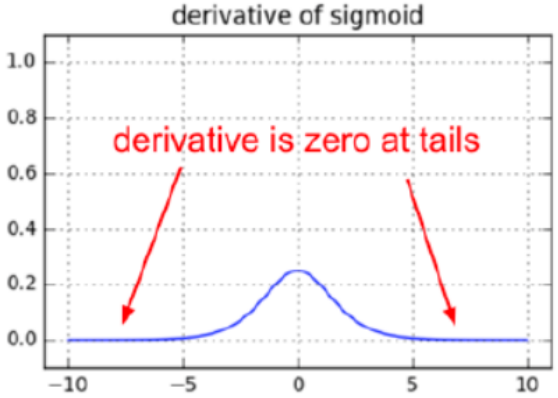
Generative Adversarial Network?

기울기 소실(Vanishing gradient)

→ 판별자의 마지막 활성화 함수인 sigmoid에서는, 값이 양 극단에서 기울기가 거의 존재하지 않는다.



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



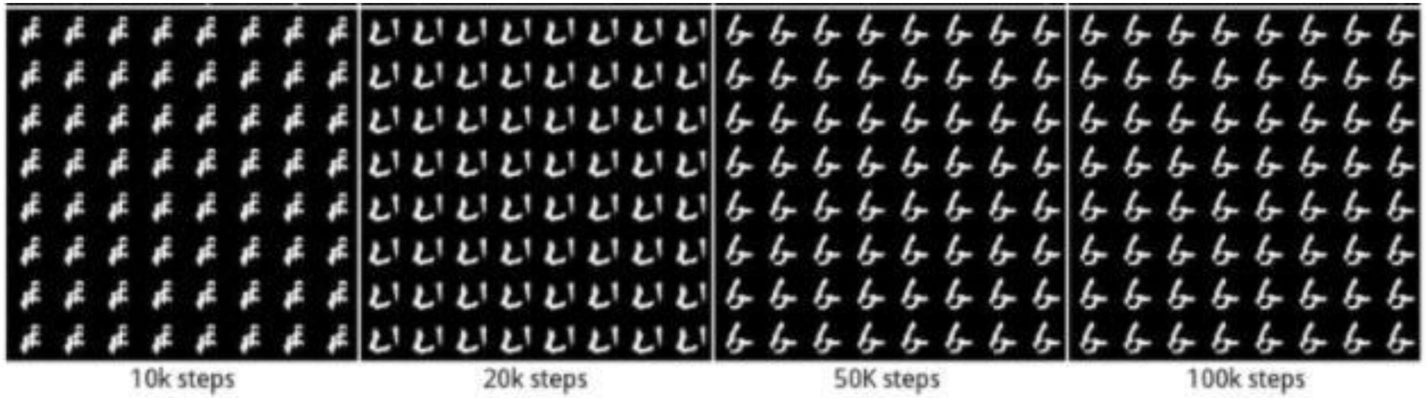
$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

- 즉, 판별자가 생성자에 비해 월등히 성능이 좋을 경우, sigmoid의 output(확률)은 0(fake case)혹은 1(real case)에 가까울 것이고, 기울기가 0에 가까워 gradient feedback이 제대로 전달되지 않는다.

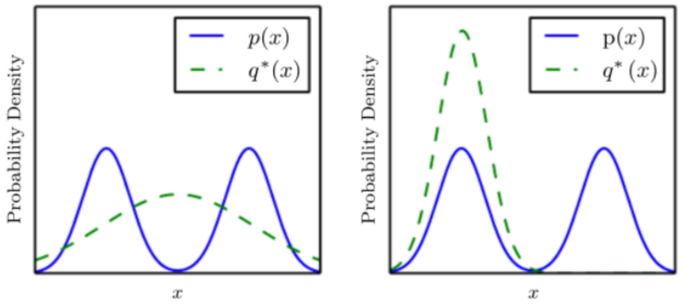
Generative Adversarial Network?

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 몰릴 수 있다.

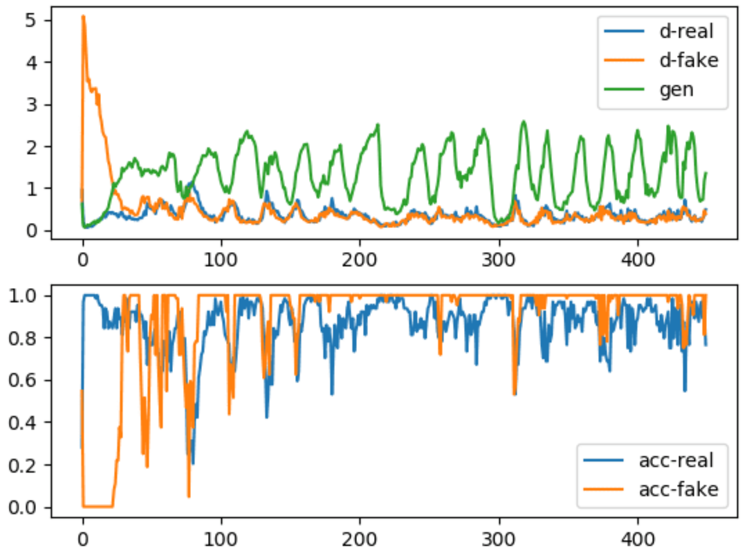


Generative Adversarial Network?

[How to Identify and Diagnose GAN Failure Modes (machinelearningmastery.com)](https://machinelearningmastery.com)

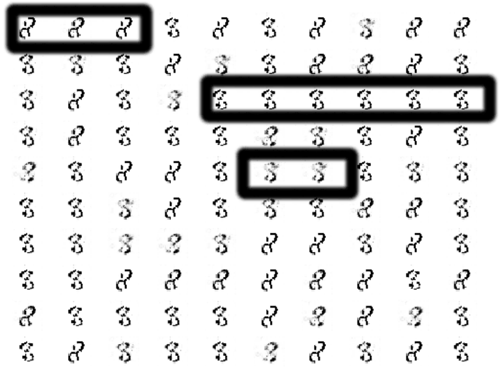
모드 붕괴(mode collapse)

→ 학습시킨 모델의 분포가 실제 데이터 분포의 모든 부분을 커버하지 못하고 다양성을 잃어버리는 현상.



진동하는 모습
(학습이 안 된 것은 아니다)

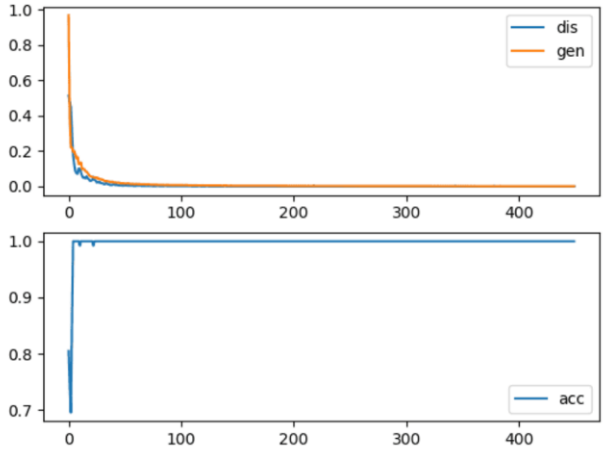
- 생성자의 latent vector z의 차원이 낮은 경우,
  매우 적은 종류의 결과만을 생성할 수 있다. ↓



Generative Adversarial Network?

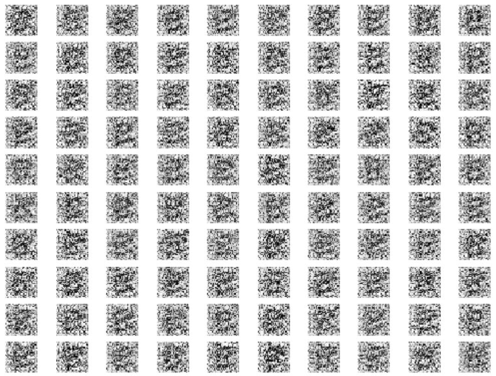
모드 붕괴(mode collapse)

→ 학습시킨 모델의 분포가 실제 데이터 분포의 모든 부분을 커버하지 못하고 다양성을 잃어버리는 현상.



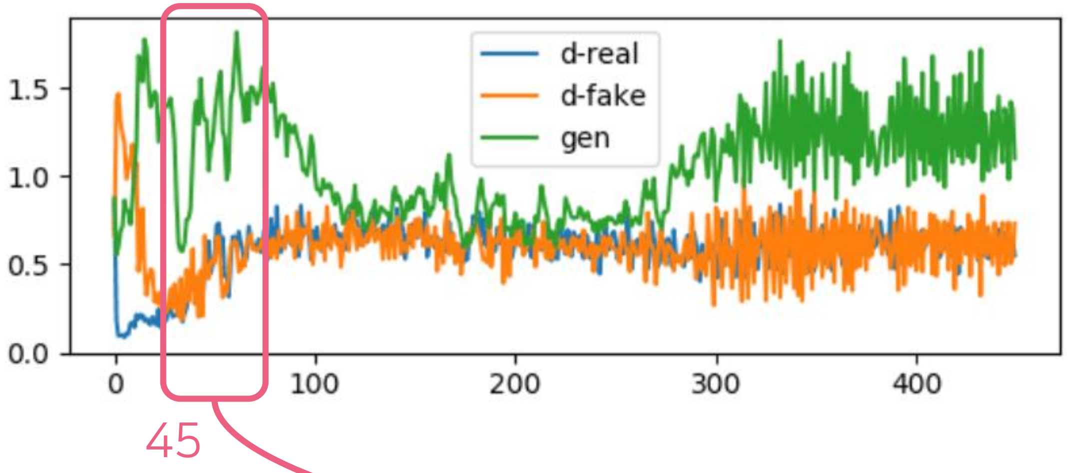
수렴하는 모습(학습 망했다)

- 구별자가 완벽하게 가짜와 진짜를 구별하는 경우



Generative Adversarial Network?

최적의 학습 상태를 어떻게 식별할 수 있을까? GAN은 판별자와 생성자 모두 고려해야 한다.

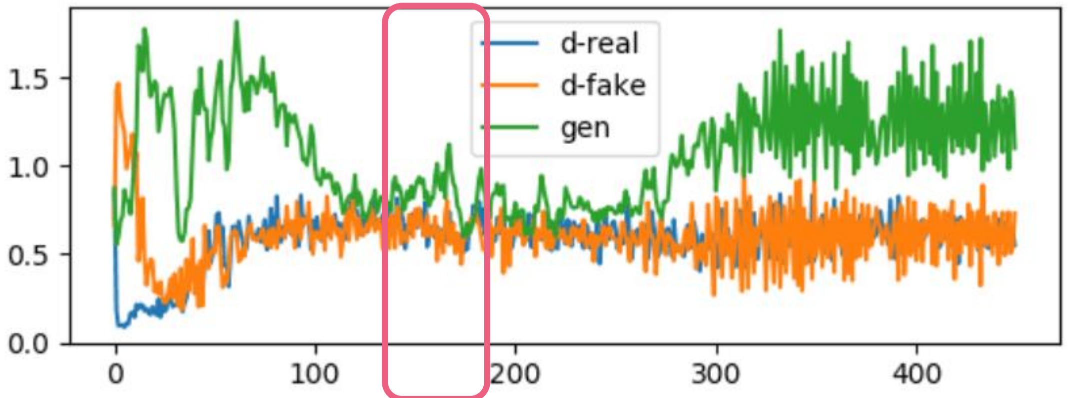


Generated Images of a Handwritten **Number 8**

at Epoch 45

Generative Adversarial Network?

최적의 학습 상태를 어떻게 식별할 수 있을까? GAN은 판별자와 생성자 모두 고려해야 한다.



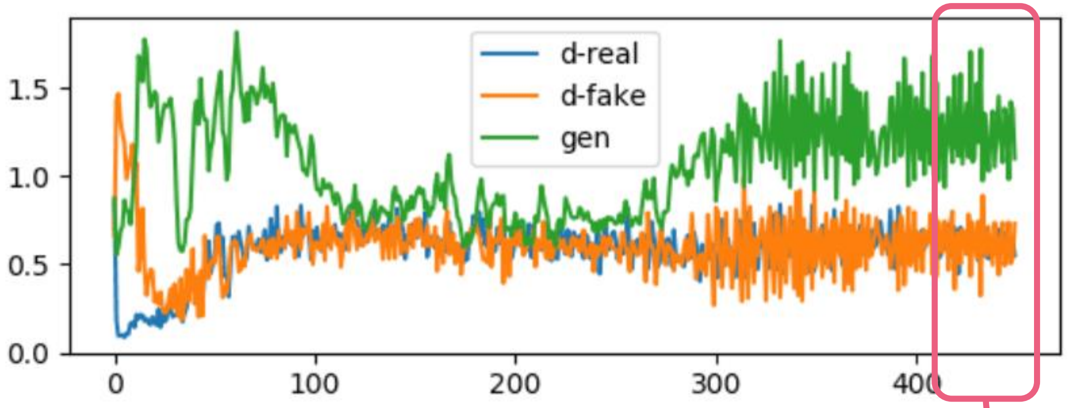
180

Generated Images of a Handwritten **Number 8**

at Epoch 180

Generative Adversarial Network?

최적의 학습 상태를 어떻게 식별할 수 있을까? GAN은 판별자와 생성자 모두 고려해야 한다.

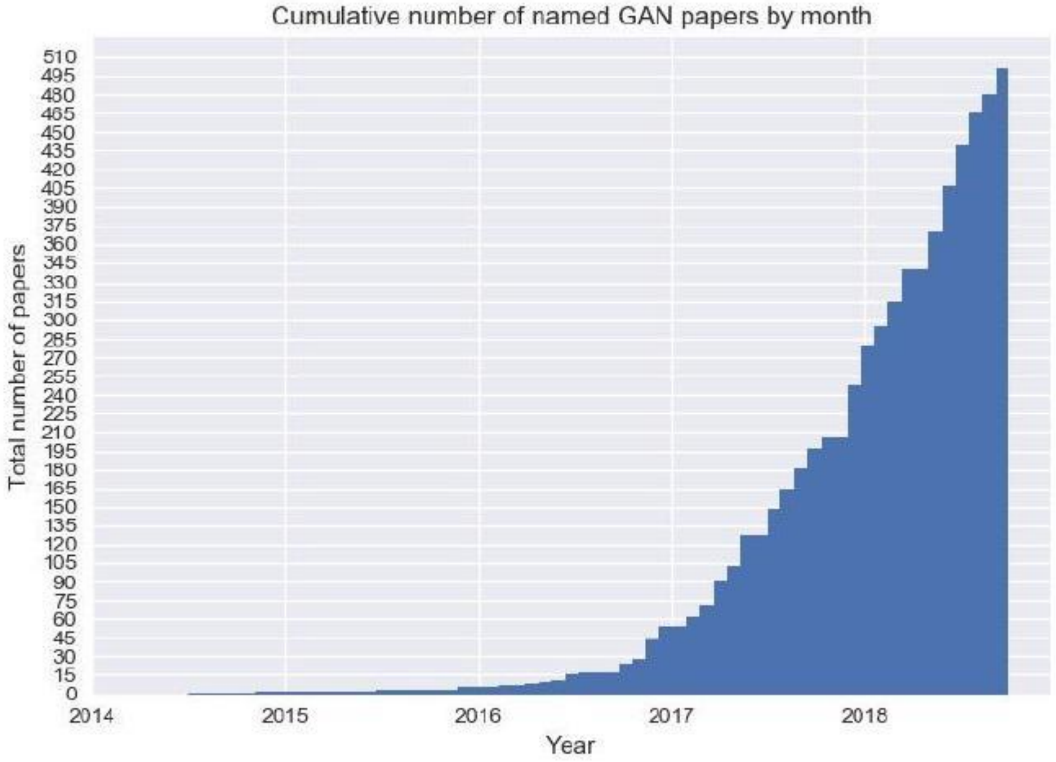


450

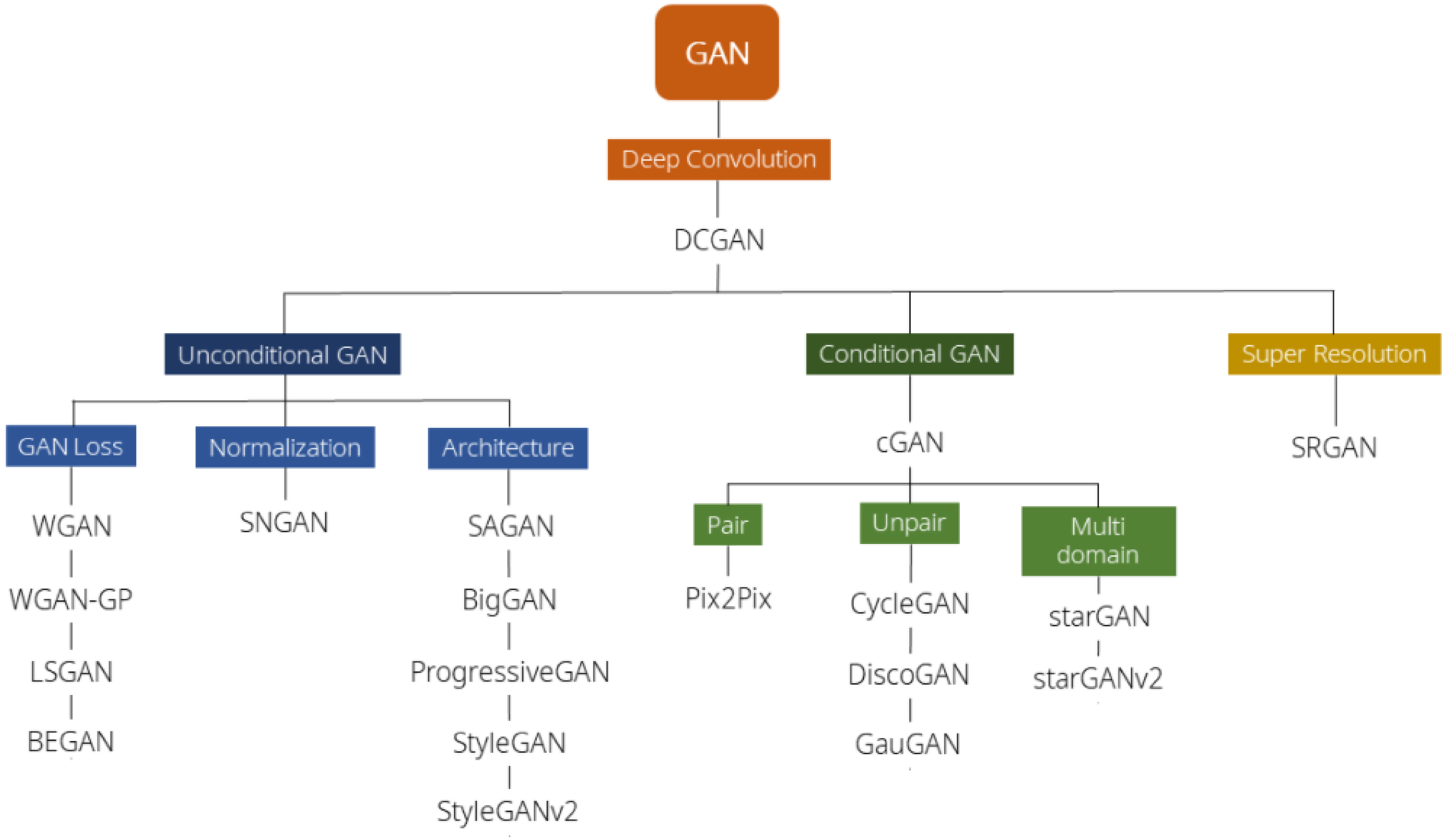
Generated Images of a Handwritten **Number 8**

at Epoch 450

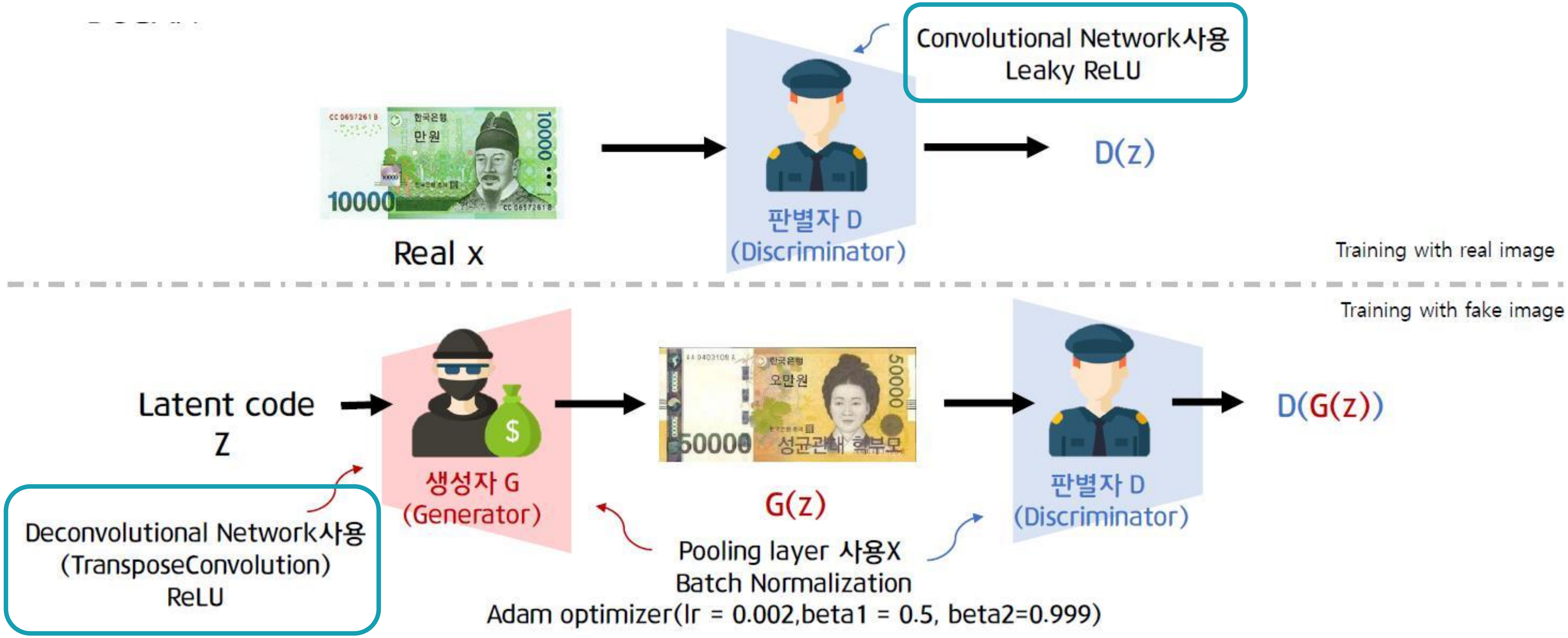
GAN의 논문 동향


Cumulative number of named GAN papers by month
Total number of papers
0
15
30
45
60
75
90
105
120
135
150
165
180
195
210
225
240
255
270
285
300
315
330
345
360
375
390
405
420
435
450
465
480
495
510
2014
2015
2016
2017
2018
Year

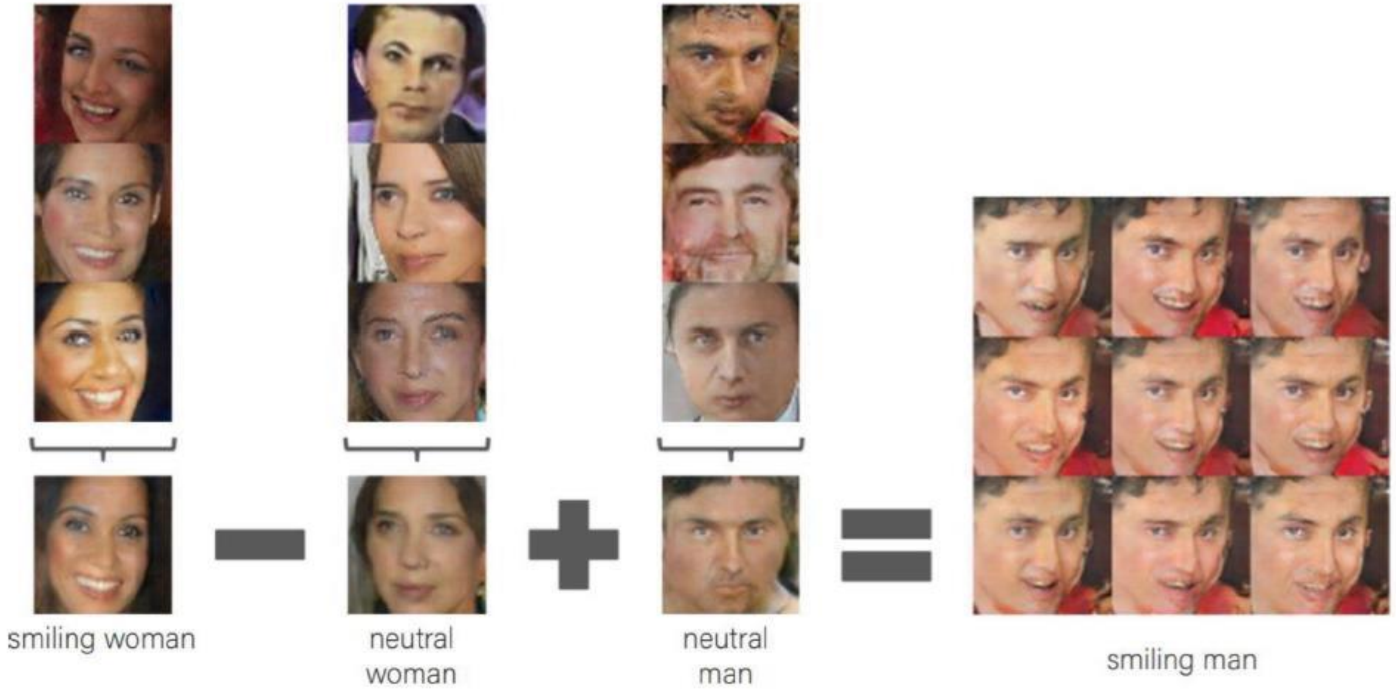
GAN의 종류


GAN
Deep Convolution
DCGAN
Unconditional GAN
Conditional GAN
Super Resolution
GAN Loss
Normalization
Architecture
cGAN
SRGAN
WGAN
WGAN-GP
LSGAN
BEGAN
SNGAN
SAGAN
BigGAN
ProgressiveGAN
StyleGAN
StyleGANv2
Pair
Unpair
Multi domain
Pix2Pix
CycleGAN
DiscoGAN
GauGAN
starGAN
starGANv2

DCGAN


Convolutional Network사용
Leaky ReLU
Real x
판별자 D
(Discriminator)
D(z)
Training with real image
Training with fake image
Latent code
Z
생성자 G
(Generator)
G(z)
판별자 D
(Discriminator)
D(G(z))
Deconvolutional Network사용
(TransposeConvolution)
ReLU
Pooling layer 사용X
Batch Normalization
Adam optimizer(lr = 0.002,beta1 = 0.5, beta2=0.999)

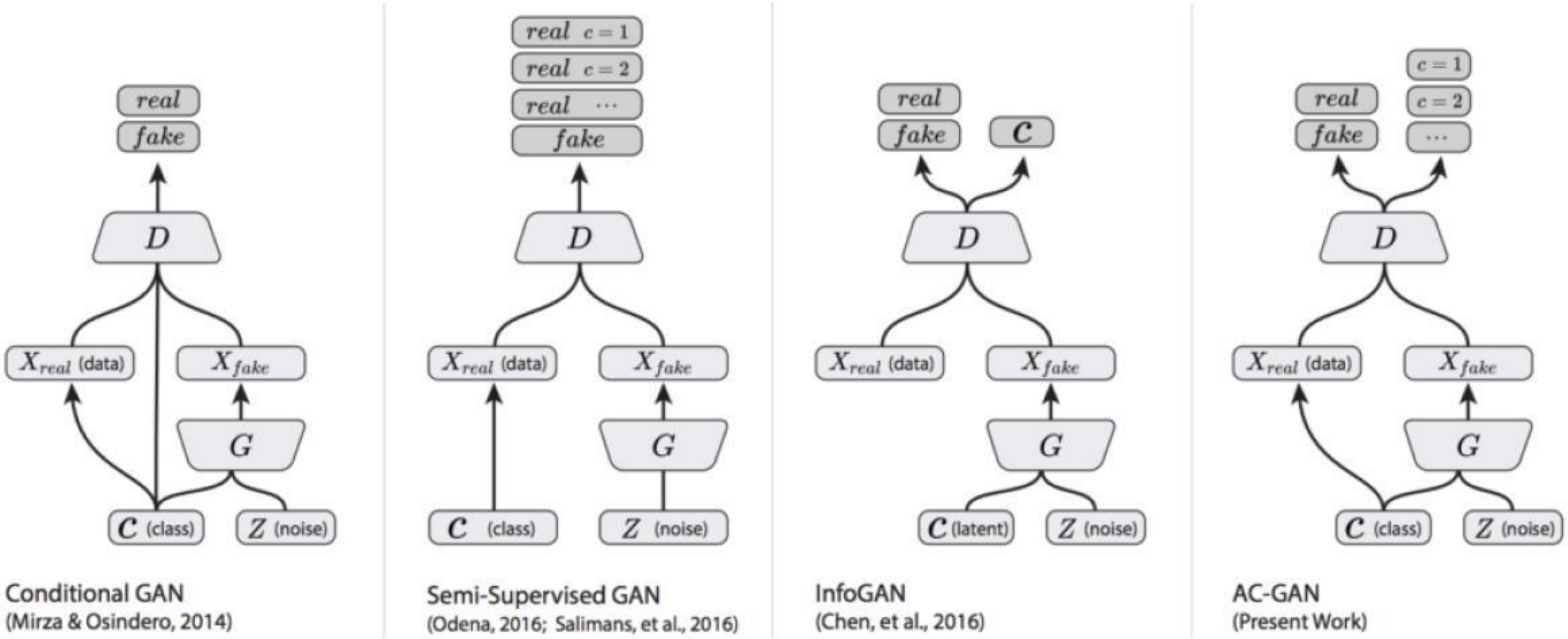
DCGAN – vector space arithmetic



[1511.06434] (arxiv.org)

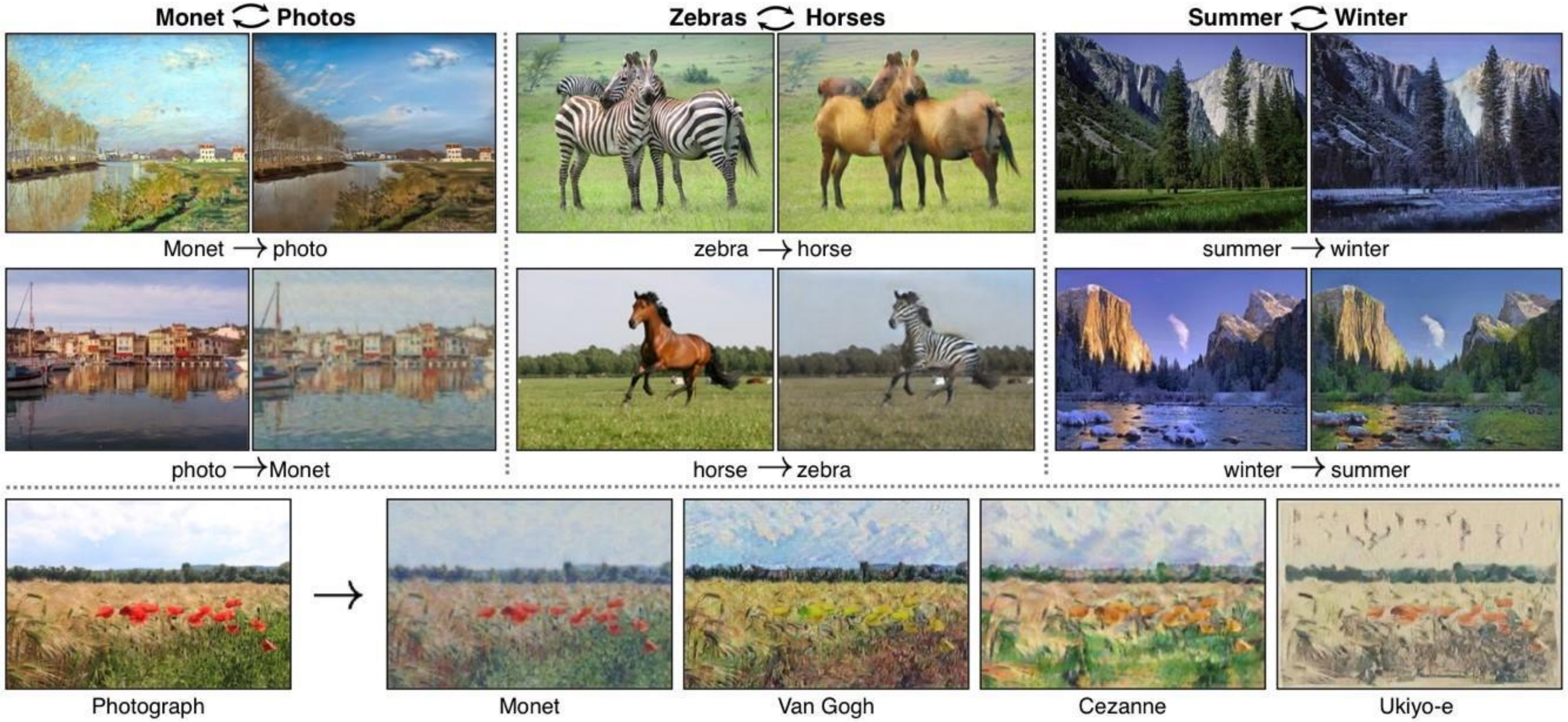
CGANs(conditional GANs)

https://arxiv.org/abs/1411.1784



CycleGAN

https://arxiv.org/abs/1703.10593



Finding connections among images using CycleGAN – YouTube – naver D2(박태성, 저자)

StarGAN

https://arxiv.org/abs/1711.09020

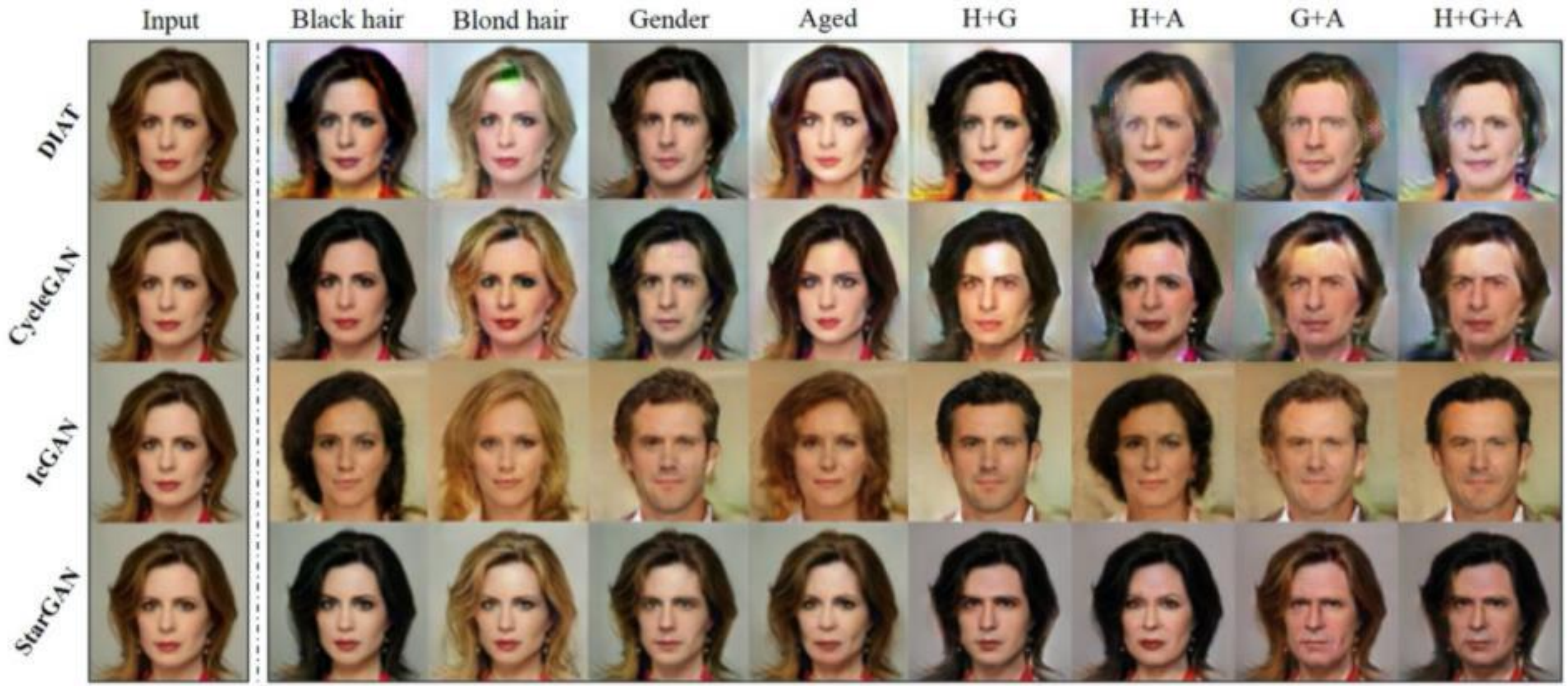


Figure 4. Facial attribute transfer results on the CelebA dataset. The first column shows the input image, next four columns show the single attribute transfer results, and rightmost columns show the multi-attribute transfer results. H: Hair color, G: Gender, A: Aged.

StyleGAN

https://arxiv.org/abs/1812.04948

https://thispersondoesnotexist.com/
By stylegan2

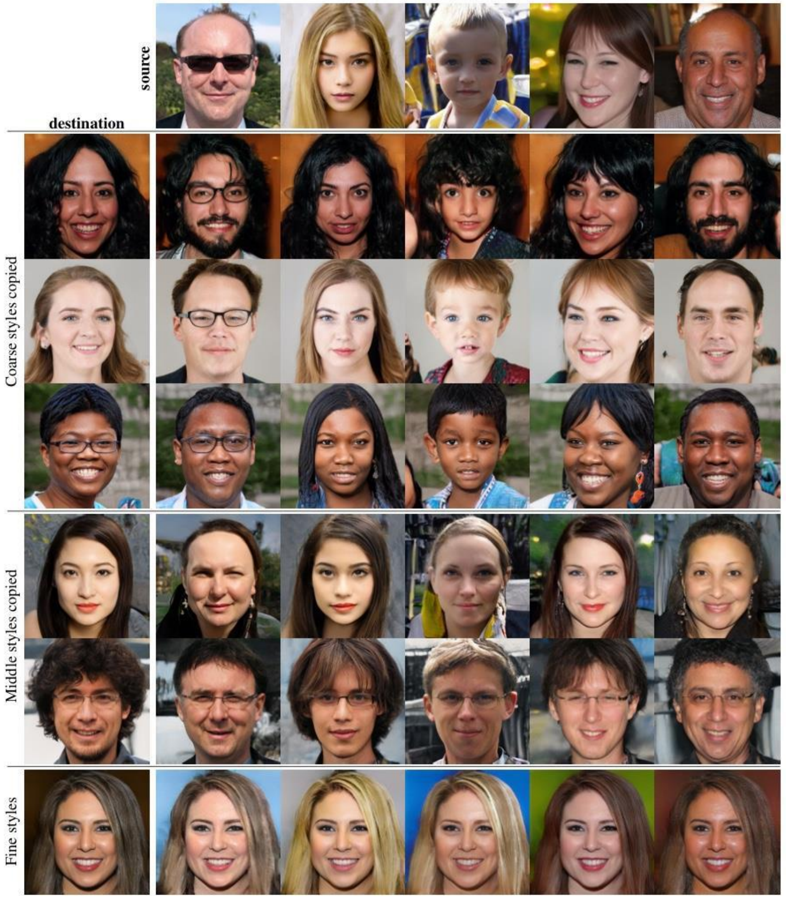


Figure 3. Visualizing the effect of styles in the generator by having the styles produced by one latent code (source) override a subset of the styles of another one (destination). Overriding the styles of layers corresponding to coarse spatial resolutions ($4^2 - 8^2$), high-level aspects such as pose, 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.

과제

- 오늘 강의 내용을 복습해 봅시다

1. VAE 파트 복습 및 내용 정리

2. 언급된 GAN 논문(DCGAN, CGAN, CycleGAN, StarGAN, StyleGAN) 중 하나 논문 리뷰

1, 2 중 택 1

Unit 06 | 출처

출처

- 투빅스 16기 박진수님 강의
- SNU 윤성로 교수님 [Deep Learning]
- 오토인코더의 모든 것 – Naver D2(이활석 님 강의)
- [정리노트] [AutoEncoder의 모든것] Chap4. Variational AutoEncoder란 무엇인가(feat. 자세히 알아보자) (tistory.com)
- discriminative vs generative · ratsgo's blog
- Seminar - 고려대학교 DMQA 연구실 (korea.ac.kr)
- Jaejun Yoo's Playground: [PR12-Video] 24. Pixel Recurrent Neural Network
- How to Identify and Diagnose GAN Failure Modes (machinelearningmastery.com)
- [1606.05908] Tutorial on Variational Autoencoders (arxiv.org)
- Deriving KL Divergence for Gaussians (leenashekhar.github.io)
- Lei Mao's Log Book – Minmax Game for Training Generative Adversarial Networks

이외 출처는 슬라이드 내에 기재하였습니다.

Q & A

들어주셔서 감사합니다.