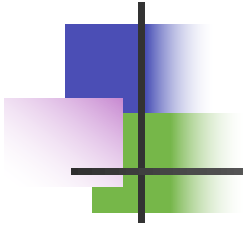




# VAE & GAN

---

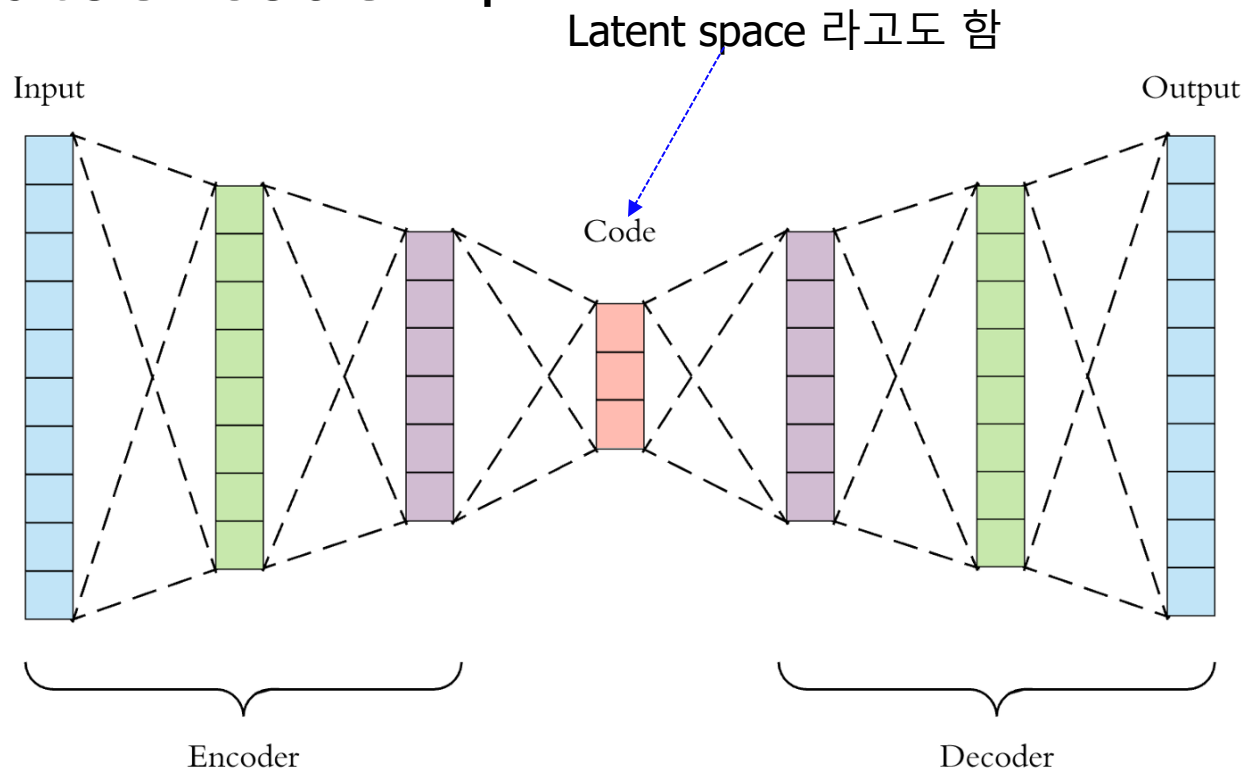
Sang Yup Lee



# Autoencoder

# Autoencoder

## ■ Autoencoder 구조





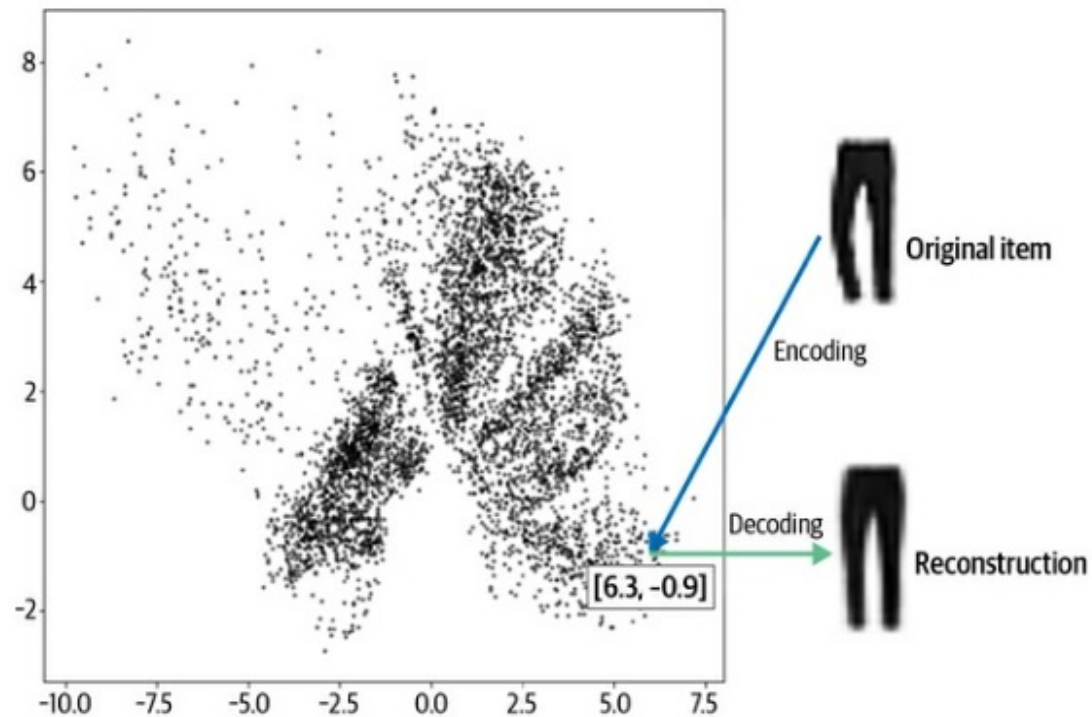
# Autoencoder

---

- Autoencoder
  - Encoder 부분
    - 차원축소의 역할
  - Decoder 부분
    - Generative model 역할
      - Latent space의 어떠한 점을 이용해서 기존에 없던 새로운 이미지를 생성할 수 있기 때문에 생성 모형으로 사용
    - 학습 데이터에 존재하는 이미지와 유사한 이미지 생성

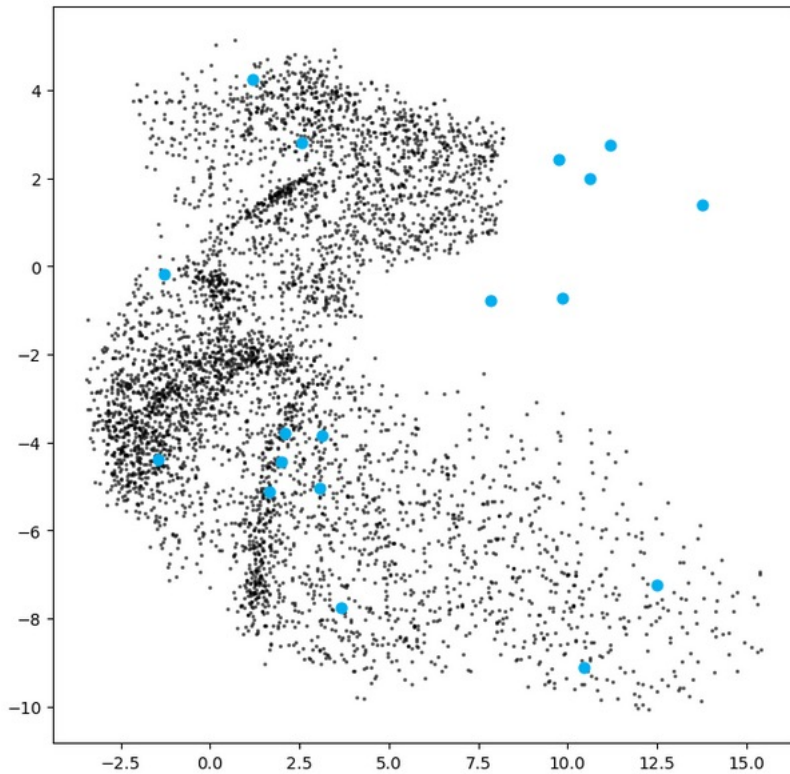
# Autoencoder

- 작동 예시: 잠재 공간이 2차원인 경우

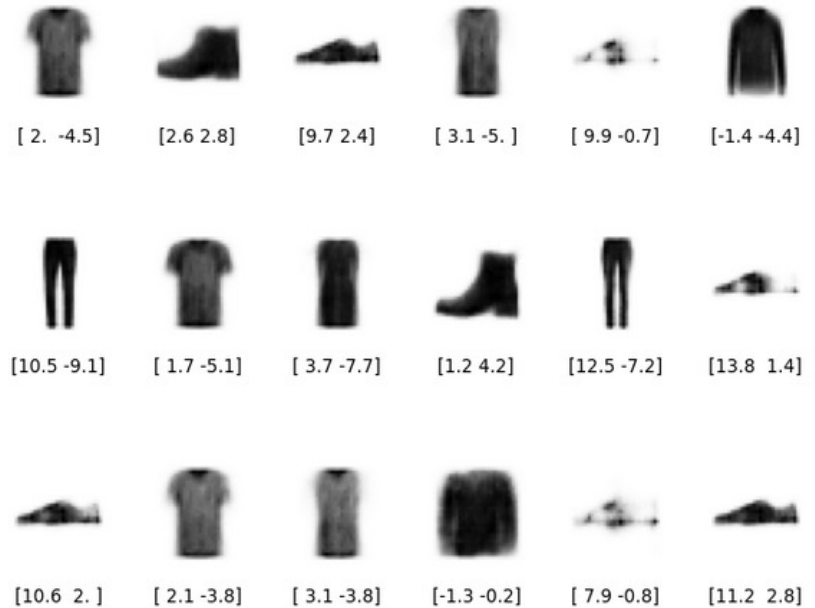


# Autoencoder

## ■ 생성 모형으로서의 사용 예



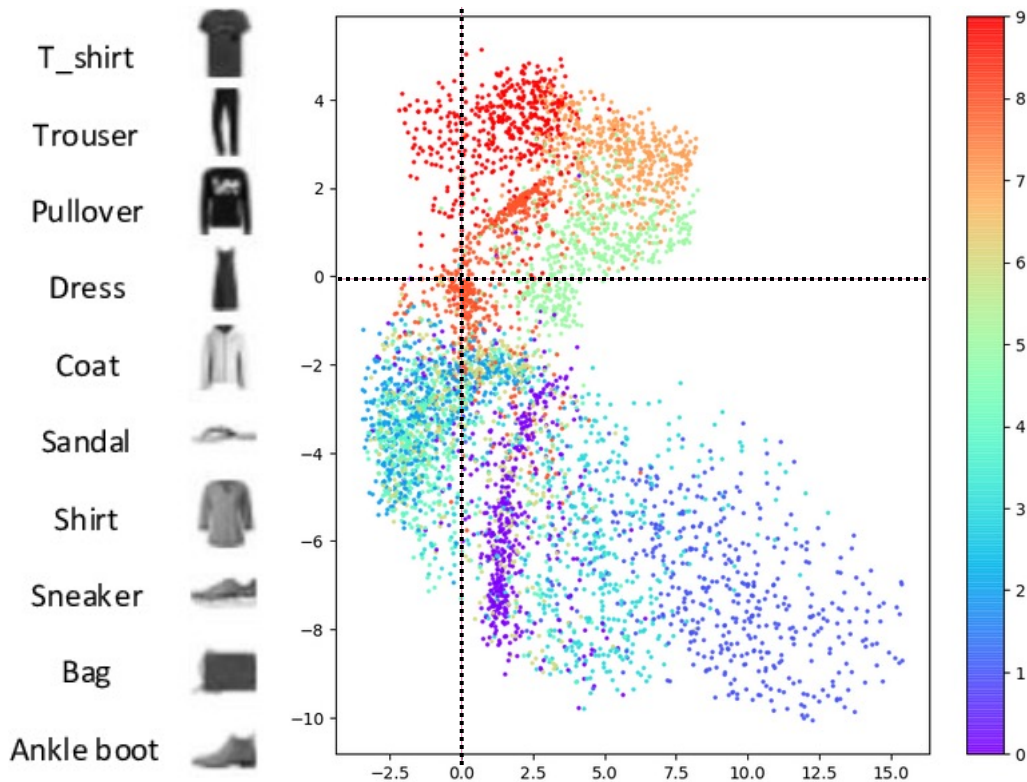
12/11/23



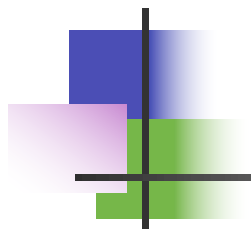
Generative AI

# Autoencoder

## ■ 생성 모형으로서의 한계



- 일부 아이템은 좁게 일부는 넓게 분포
- 원점을 기준으로 대칭 분포가 아님
  - 4사분면에 제일 많이 분포
  - 규칙성이 떨어짐
- 빈공간이 너무 많음
- 공간이 비연속적



# Variational AE





# Variational Autoencoder

---

- 주로 이미지 생성의 목적으로 사용
- AE의 한계
  - 일반 AE의 경우, latent space 정보를 이용해서 (decoder에서) 이미지를 생성할 경우, 규칙성과 연속성이 부족해서 제대로 된 이미지가 생성되지 않음
- VAE의 주요 특징
  - 입력 데이터를 하나의 저차원 벡터로 encoding 하는 것이 아니라, 하나의 확률 분포로 encoding
  - 해당 확률 분포를 이용해서 하나의 벡터를 생성 (이 벡터를 잠재 변수라고 간주 가능)
  - 해당 벡터를 이용해서 디코더에서 이미지 생성  $\Rightarrow$  이는 디코더를 확률적 생성 모형으로 사용하겠다는 것을 의미



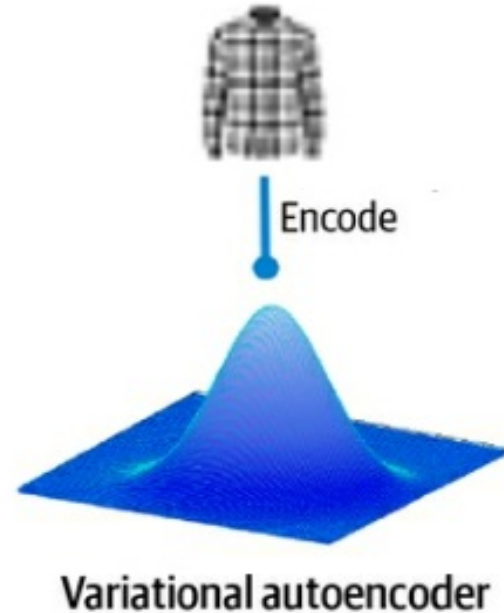
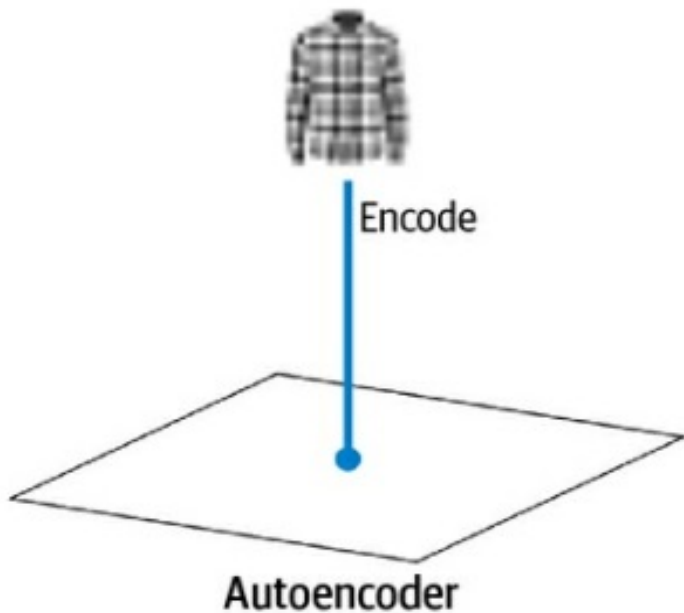
# Variational Autoencoder

---

- 정규 분포에 대한 학습
  - 정규성을 갖게 하기 위해서 인코더를 통해 생성되는 정규 분포가 표준 정규 분포가 되도록 학습
  - 즉, 오차항으로 인코더가 출력하는 정규 분포와 표준 정규 분포의 차이 반영
    - 이러한 차이는 Kulback-Leibler divergence를 사용해서 표현
    - 이러한 방법이 베이지안 추론에서 사용되는 variational inference (변분 추론)와 유사
- 전체 비용함수
  - Reconstruction error + KL divergence

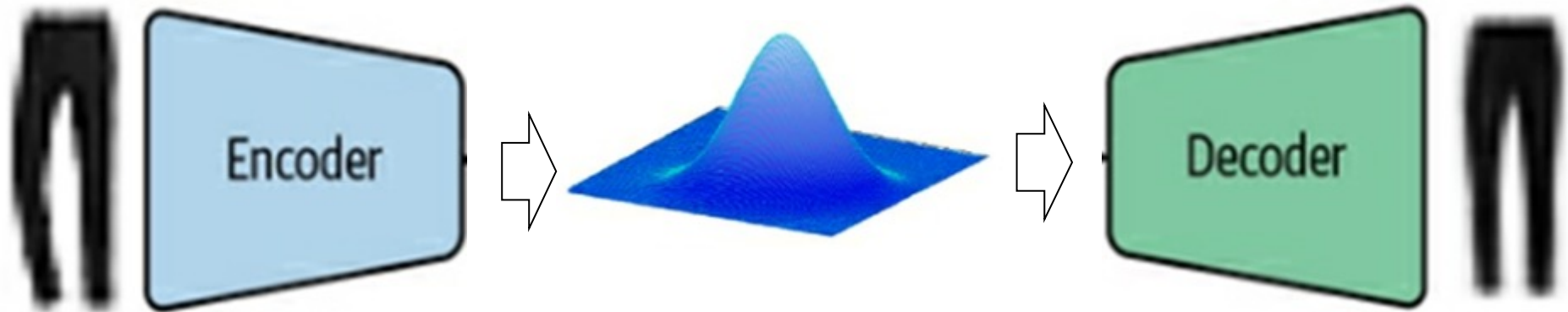
# Variational Autoencoder

- Encoder 부분



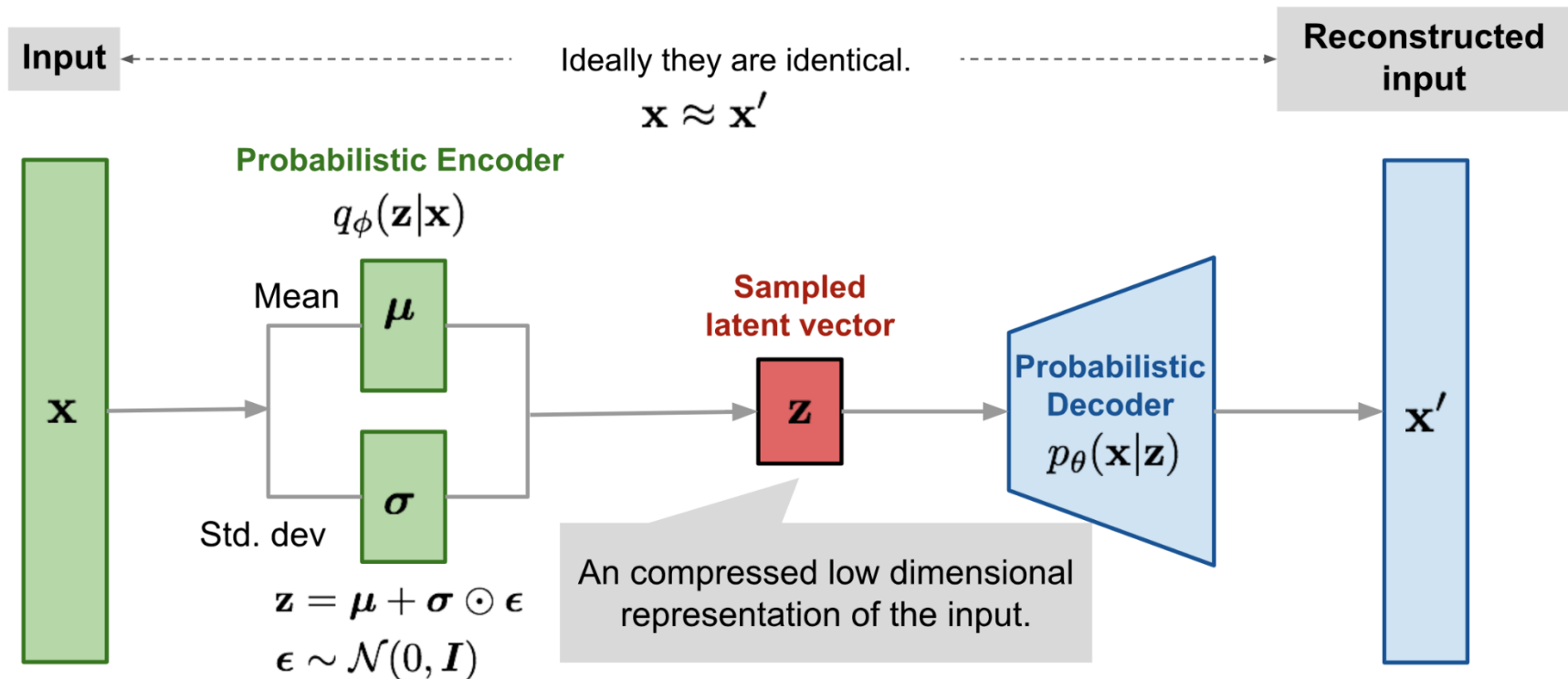
# Variational Autoencoder

- 전체 구조



# Variational Autoencoder

## ■ 구조





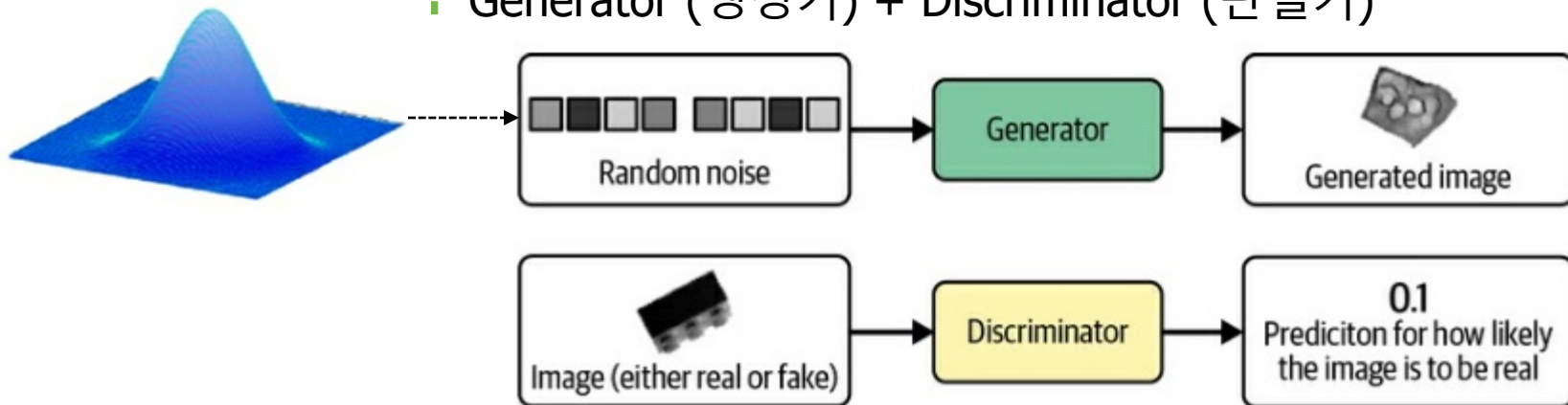
---

# **GAN (Generative adversarial network)**

# GAN

- GAN 의 구조

- 서로 다른 두 개의 신경망 모형으로 구성
  - Generator (생성기) + Discriminator (판별기)



- Generator와 discriminator 간의 battle
  - Generator는 discriminator를 속이기 위해 노력
  - Discriminator는 진짜와 가짜를 정확하게 구분하기 위해 노력



# GAN

---

## ■ Generator

- 표준정규분포를 통해 추출되는 random noise 를 (학습 데이터에 존재하는) 실제 이미지와 유사한 이미지를 생성
- Generator는 VAE의 decoder와 유사

## ■ Discriminator

- 입력된 이미지가 학습 데이터에 존재하는 실제 이미지인지 아니면 generator가 생성한 가짜 이미지인지를 맞히는 역할
- 되도록 정확하게 구분할 수 있도록 학습
- 일반적인 이미지 분류 작업을 수행하기 때문에 CNN 기반의 모형과 유사





# GAN

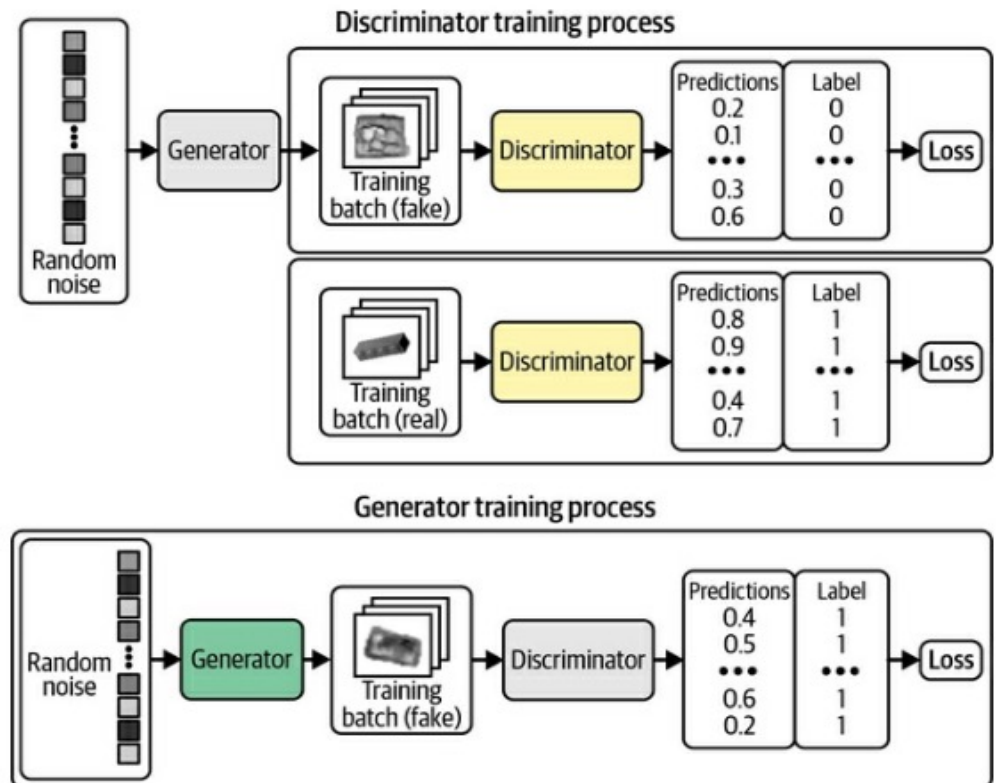
---

## ■ 학습

- GAN의 경우는 학습 과정을 이해하는 것이 중요
- Discriminator 학습
  - 실제 이미지와 generator를 이용해 생성된 이미지로 구축된 학습 데이터 사용
- Generator 학습
  - Generator가 생성한 이미지에 대한 discriminator의 예측값, 즉, 해당 이미지가 진짜일 확률이 1에 가깝도록 학습

# GAN

- 학습 과정
  - Discriminator의 학습과 generator의 학습을 번갈아 가면서 수행
    - 한번에 한 네트워크 (즉, discriminator 또는 generator)의 weights 만 업데이트 필요 (다른 네트워크의 weights는 고정)





# GAN

---

- 비용함수

- Discriminator 비용함수

- 교차엔트로피

- $-\frac{1}{N} \sum_{i=1}^N \{y_i \ln p_i + (1 - y_i) \ln(1 - p_i)\}$

- 입력되는 이미지가 진짜 이미지인 경우

- $y_i = 1, p_i = D(x_i)$

- 입력되는 이미지가 가짜인 경우

- $y_i = 0, p_i = D(G(z_i))$

- 비용함수 최소화 문제

- $\min_D - \left[ E_{x \sim p_X} \{ \ln D(x) \} + E_{z \sim p_Z} \{ \ln (1 - D(G(z))) \} \right]$



# GAN

---

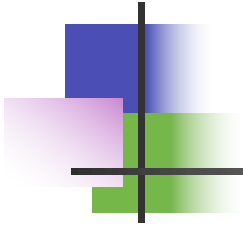
- 비용함수
  - Generator 비용함수
    - $p_i = D(G(z_i))$  의 값을 1에 가깝게 할수록 모형의 성능이 좋은 것을 의미
    - 따라서 종속변수의 값을 1로 지정하고 아래의 비용함수를 사용
      - $-\frac{1}{N} \sum_{i=1}^N (y_i \ln p_i)$ , where  $y_i = 1, p_i = D(G(z_i))$
  - 비용함수 최소화 문제
    - $\min_G -[E_{x \sim p_Z} \{\ln D(G(z))\}]$



# GAN

---

- Python code
  - GAN\_example.ipynb



# Q & A