

〈데이터분석과기계학습 13주차〉
인공신경망 (4) - GAN 과 생성모델

인공지능융합공학부 데이터사이언스전공
곽찬희

무언가 만들어 내려면

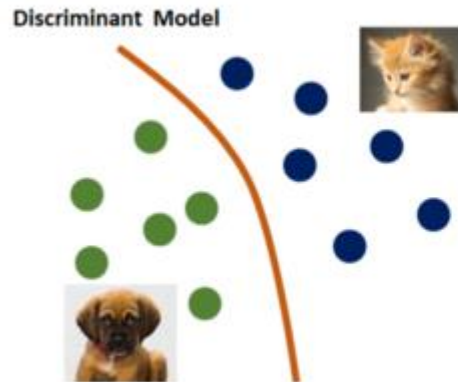
- 생성된 데이터의 범위라는 것
 - ✓ 키의 분포를 생각해봅시다.
- 범위와 그 퍼진 정도를 알면 데이터를 만들 수 있을까?
 - ✓ 분포!
- 그럼, 분포는 어떻게 구하지?



생성모델과 판별모델

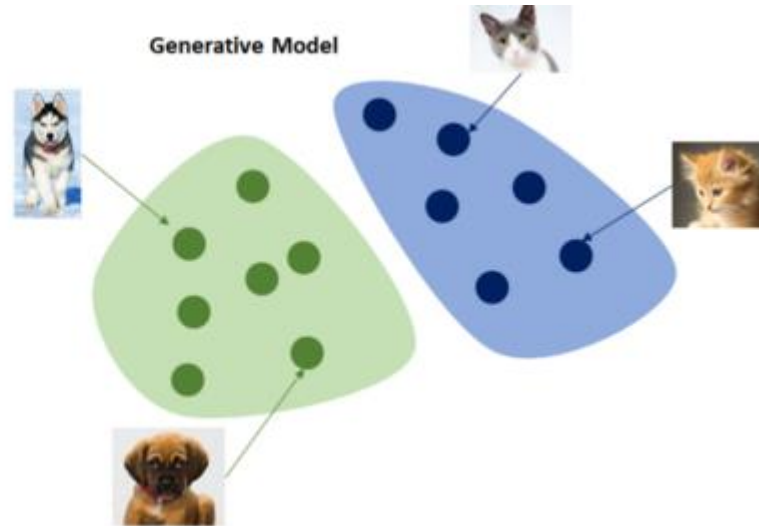
- 판별 모델(Discriminative Model)

- ✓ 데이터 X가 주어졌을 때 Y가 나타날 확률 (즉, $P(Y|X)$)을 반환하는 모델
- ✓ Supervised Learning
- ✓ 결정 경계 (Decision Boundary)를 잘 찾는 것이 목표



- 생성 모델(Generative Model)

- ✓ 지도적 생성 모델(supervised)은 $P(X|Y)$ 를 추정한 뒤 $P(Y|X)$ 를 계산
- ✓ 비지도적 생성 모델(unsupervised)는 X 자체의 분포를 추정하여 데이터 생성에 활용
- ✓ 좋은 분포 추정이 목표



생성 모델의 종류

- Tractable Density: X에서 확률 분포를 계산
- Approximate Density: X에서 확률 분포를 추정
- Implicit Density: X의 분포는 모르겠고 최대한 비슷하게!

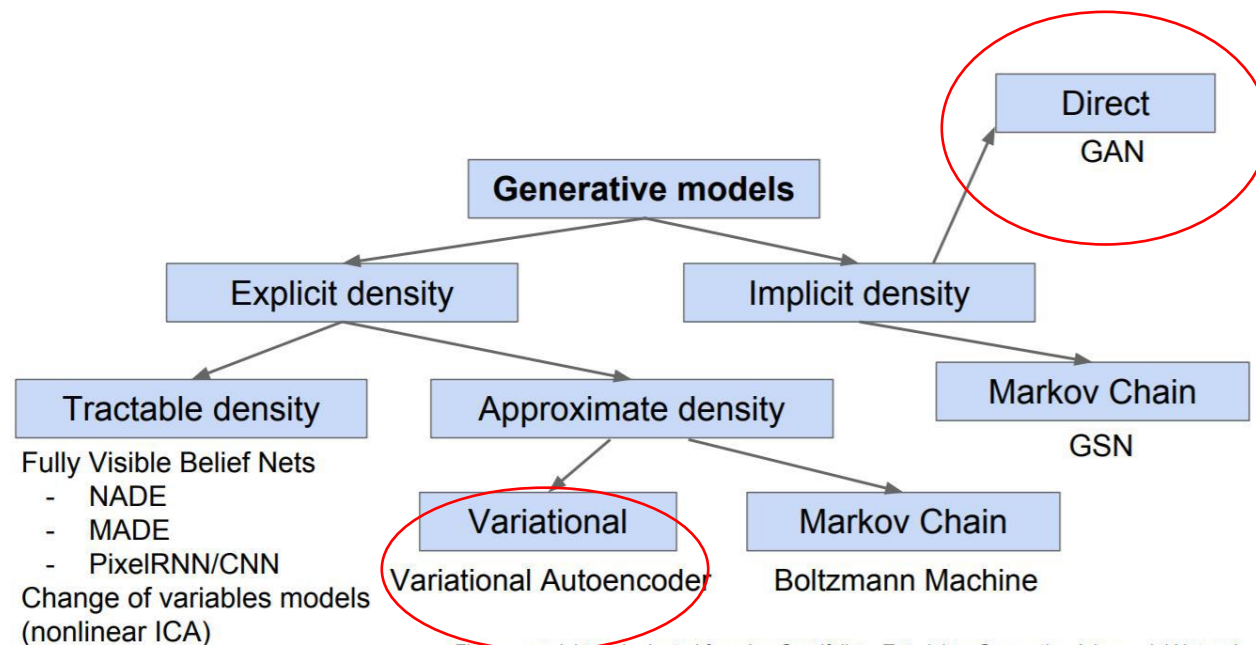


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.



오토인코더 (Autoencoder)

- 인코더(Encoder)

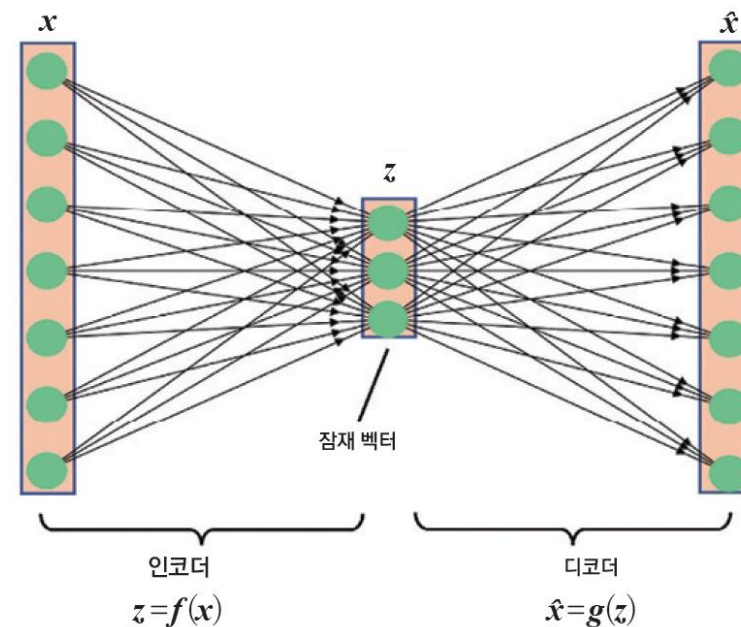
- ✓ d 차원 x 를 p 차원 z 로 변환하는 학습된 신경망 ($z = f(x)$)
- ✓ z 를 잠재 벡터 (latent vector) 라 부름
- ✓ 보통 $d > p$

- 디코더(Decoder)

- ✓ p 차원의 z 를 다시 x 로 추정(\hat{x})
- ✓ $G(z) = \hat{x}$

- 여러 은닉층을 추가해 복잡도를 증가시킬 수 있음(+비선형성 고려)

- ✓ CNN 활용 가능



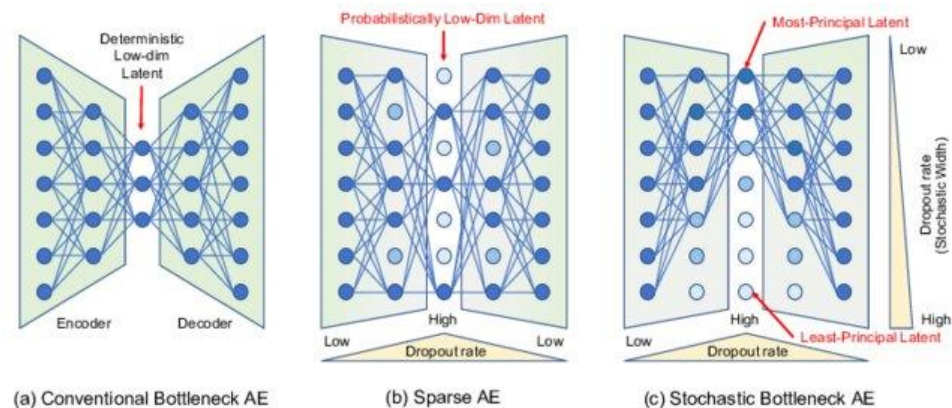
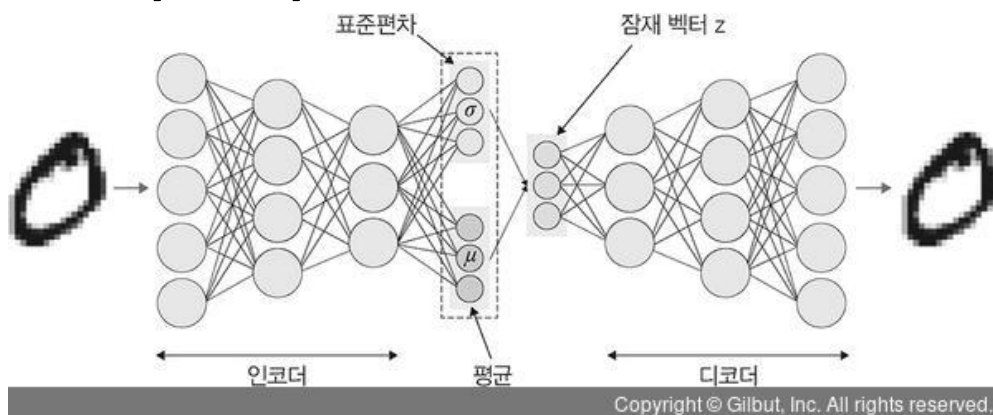
오토인코더 (Autoencoder)

- 왜 중요해요?

- ✓ 데이터 압축 - 적은 차원으로 원래 데이터 구성을 복원 가능
- ✓ 차원의 저주 - 특성 개수를 줄여 차원 감소 유도
- ✓ 특성 추출 - 중요한 특성을 찾아줌 (latent vector / latent space)

- 역시나! 다양한 변형이 존재

- ✓ SAE, DAE, CAE...



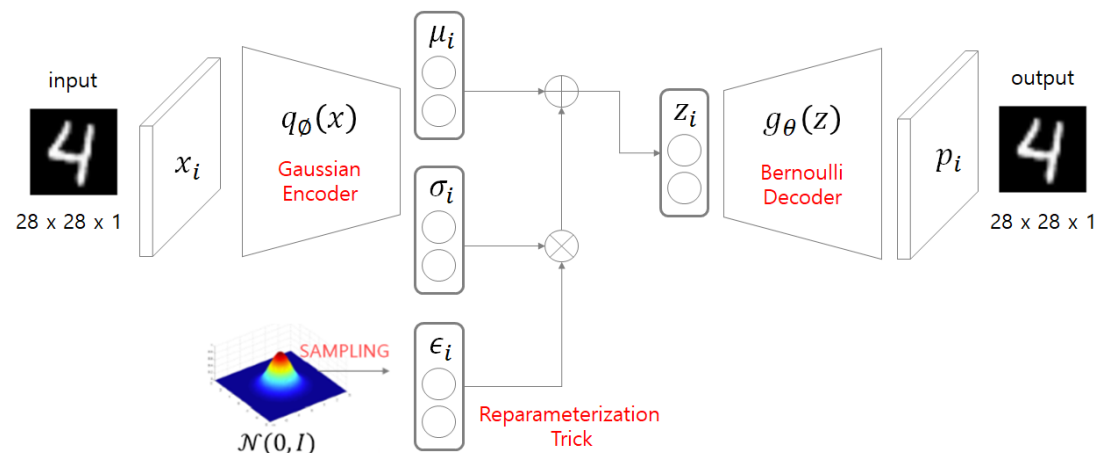
비슷한 듯 다른 VAE

- Variational AutoEncoder (VAE)

- ✓ z (latent vector)의 분포를 추정하고, 여기서 샘플링한 정보와 오차를 고려해 입력 데이터와 유사한 데이터를 생성함

- 핵심 아이디어

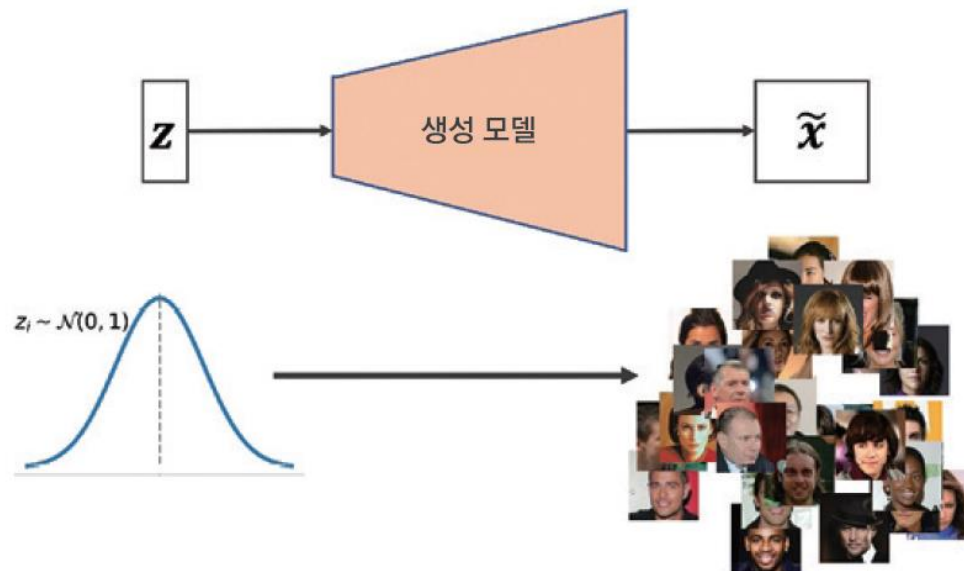
- ✓ Encoder 함수의 output은 latent variable의 분포의 μ 와 σ 를 내고, 이 output값을 표현하는 확률밀도함수를 생각해볼 수 있다.
- ✓ latent variable에서 sample된 z 라는 value (= decoder input)이 만들어진다.

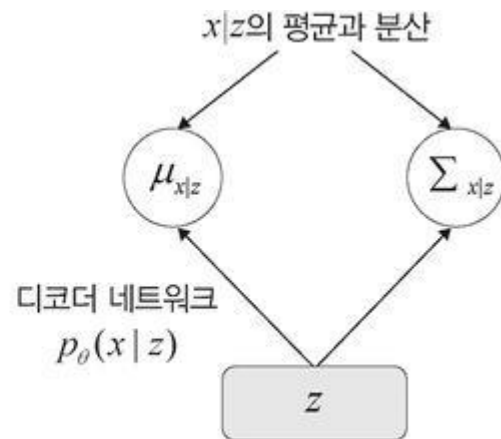
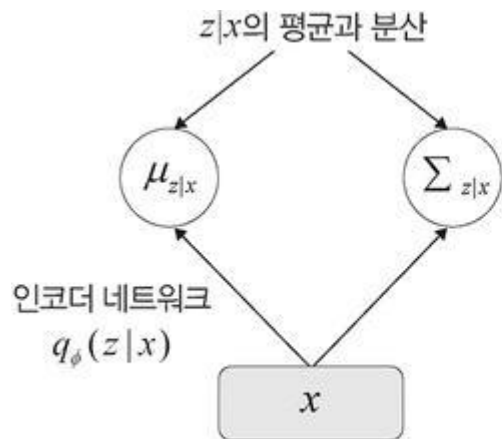


비슷한 듯 다른 VAE

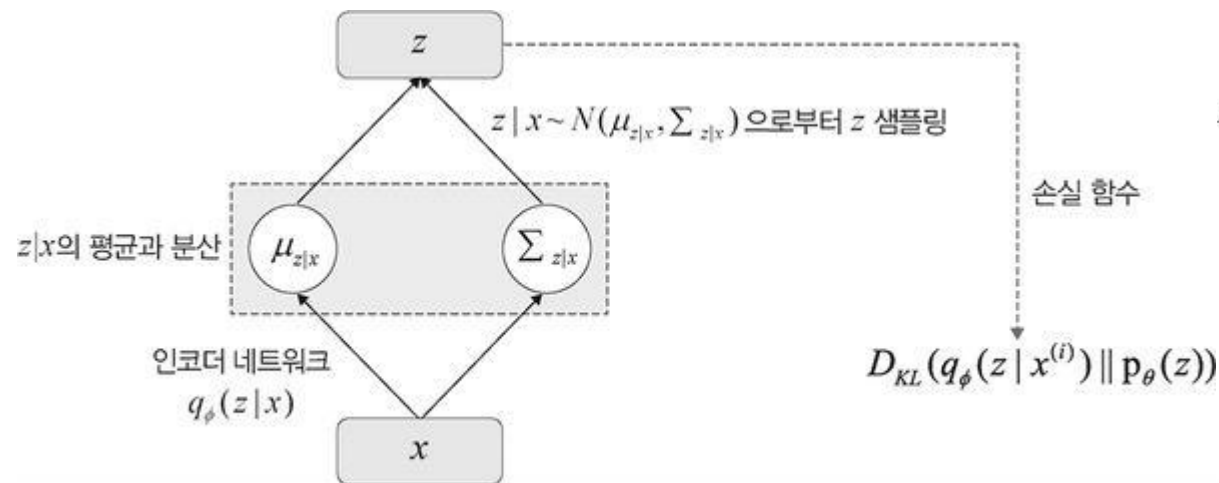
- AE와의 차이점?

- ✓ AE의 목표는 encoder(차원의 축소) \rightarrow x 를 나타낼 수 있는 z 의 도출이 목적. Decoder 는 encoder 학습에서 나온 산물이지 목적이 아님
- ✓ VAE 의 목표는 decoder (생성) \rightarrow x 와 유사한 샘플 생성이 목적. Generative model!
- ✓ AE는 분포 가정 없음(z 의 분포 모름) vs. VAE는 분포를 가정 ($z \sim N(0,1)$)
- ✓ AE를 잘 일반화시키면 생성모델로 바꿀 수 있음 \rightarrow VAE

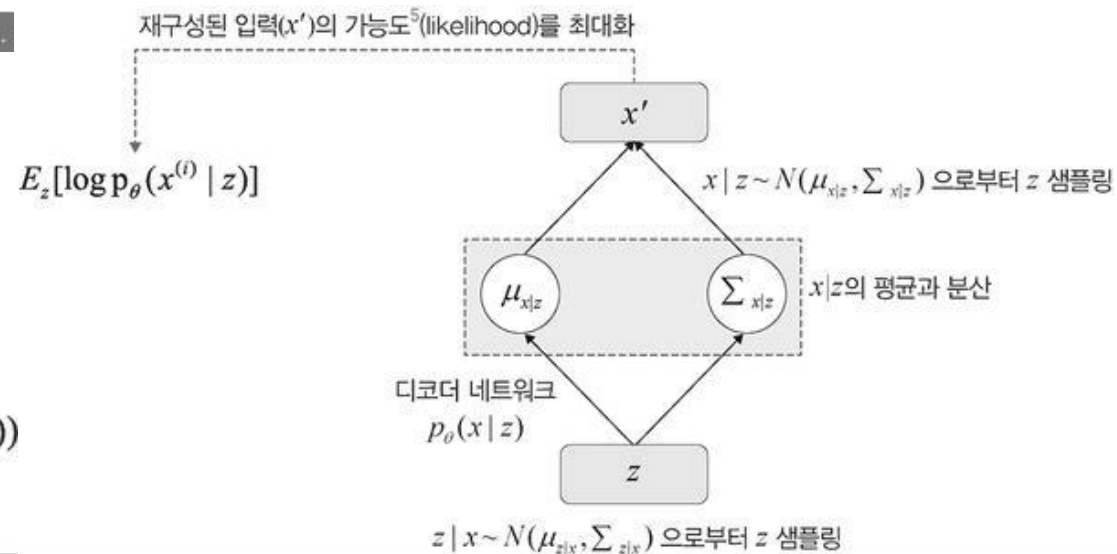




Copyright © Gilbut, Inc. All rights reserved.

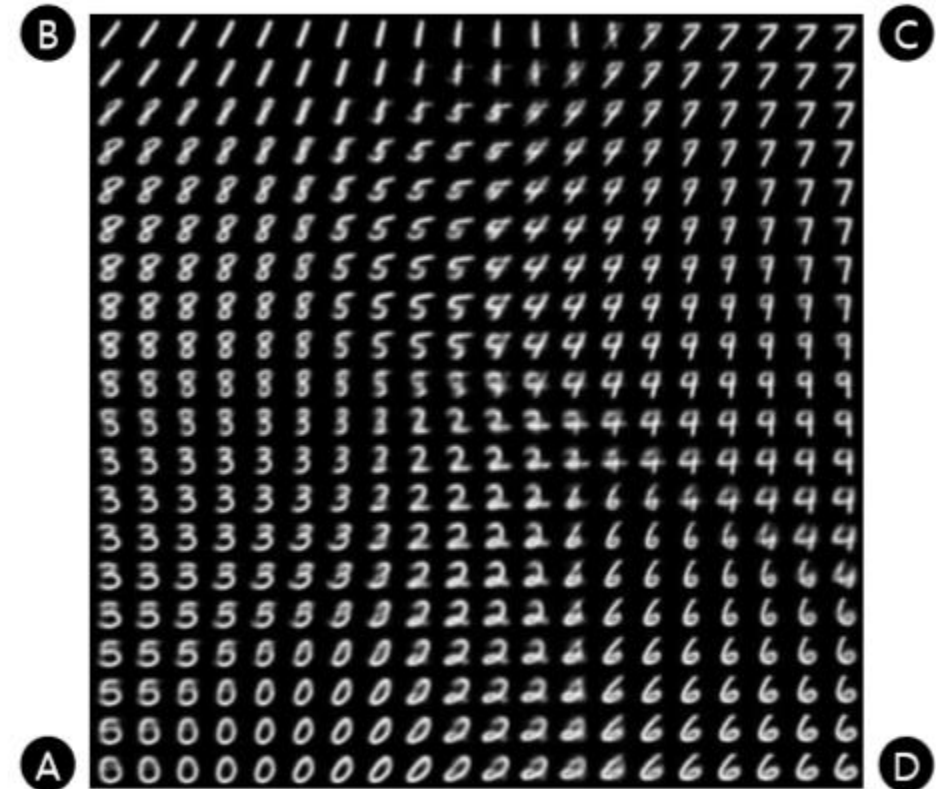
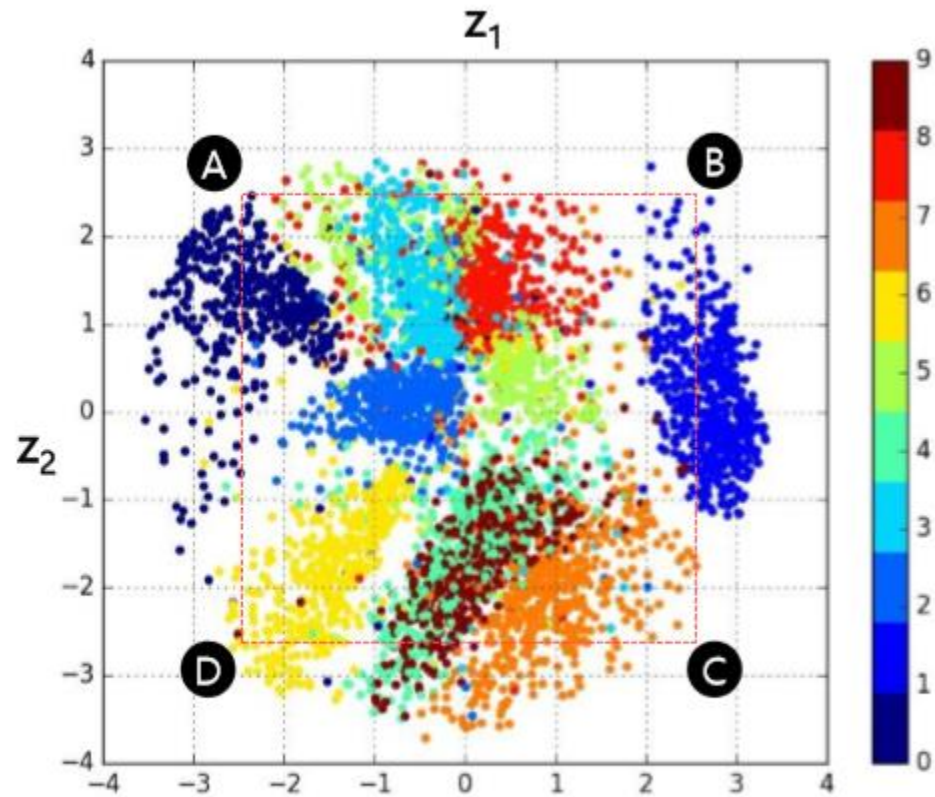


Copyright © Gilbut, Inc. All rights reserved.



Copyright © Gilbut, Inc. All rights reserved.

Learned Manifold

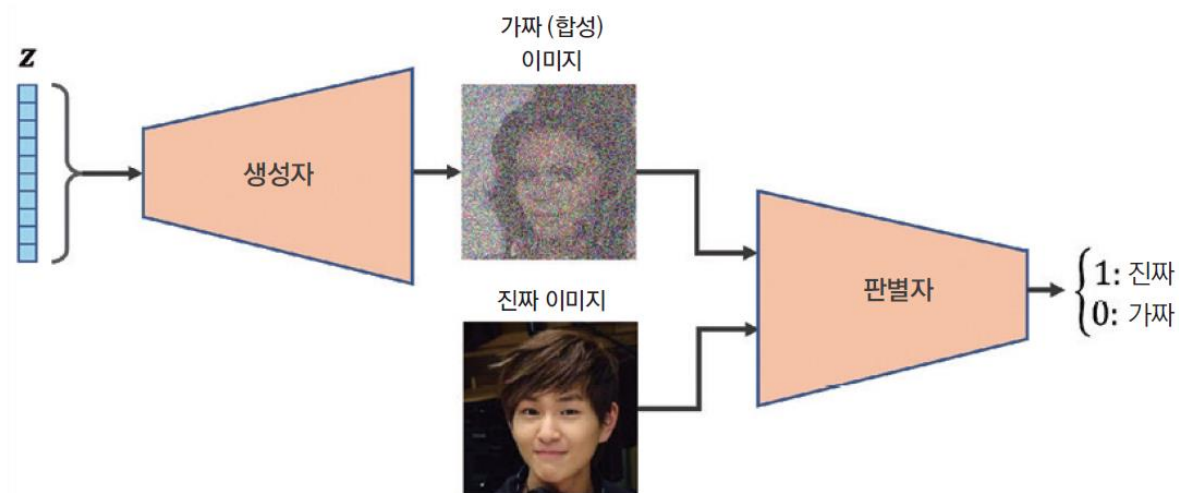


학습이 잘 되었을 수록 2D공간에서 같은 숫자들을 생성하는 z 들은 뭉쳐있고,
다른 숫자들은 생성하는 z 들은 떨어져 있어야 한다.

<https://github.com/hwalsuklee/tensorflow-mnist-VAE>

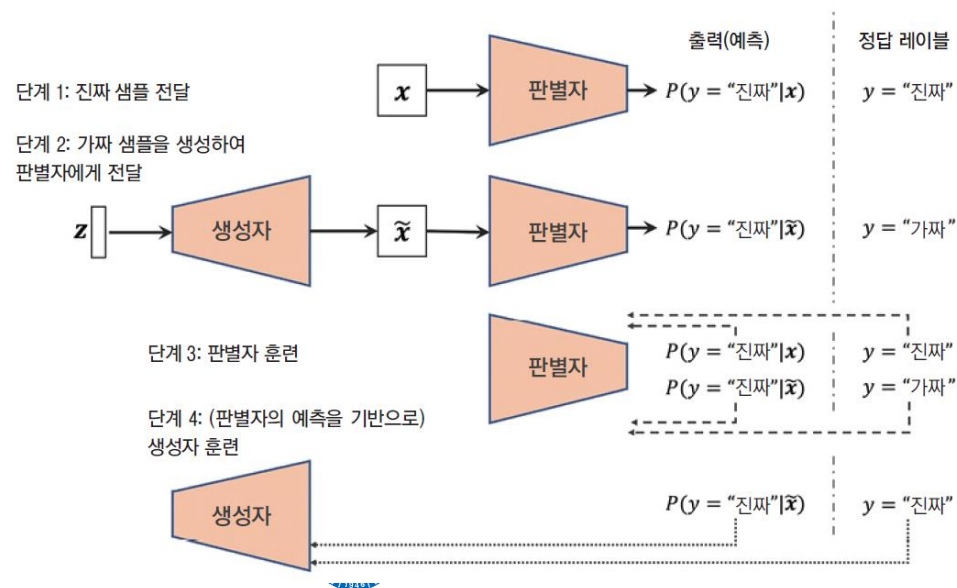
GAN의 구조

- 생성자 (Generator, G)
 - ✓ 알려진 분포에서 샘플링한 랜덤 벡터 z 를 입력받아 출력 이미지 x 를 생성하는 신경망
- 판별자 (Discriminator, D)
 - ✓ 진짜 이미지 x 에서 합성 이미지 \hat{x} 를 감지하는 법을 학습하는 분류기



GAN의 훈련

- GAN에서 생성자와 판별자 두 신경망이 함께 훈련
 - ✓ 먼저 모델 가중치를 초기화한 후 생성자가 진짜처럼 보이지 않은 이미지를 만듦
 - ✓ 비슷하게 판별자는 진짜 이미지와 생성자가 합성한 가짜 이미지를 구분하는 능력이 형편없음
 - ✓ 시간이 지남에 따라 (즉, 훈련을 통해) 두 신경망이 서로 상호 작용을 하면서 향상됨
 - ✓ 두 신경망은 적대적인 게임(adversarial game)을 수행
 - ✓ 생성자는 판별자를 속이기 위해 출력을 향상시키도록 학습
 - ✓ 동시에 판별자는 합성 이미지를 더 잘 감지하도록 훈련



GAN의 훈련

- 그럼, 진짜 사진과 만든 사진의 오류는 어떻게?
- 가치함수 (Value Function)

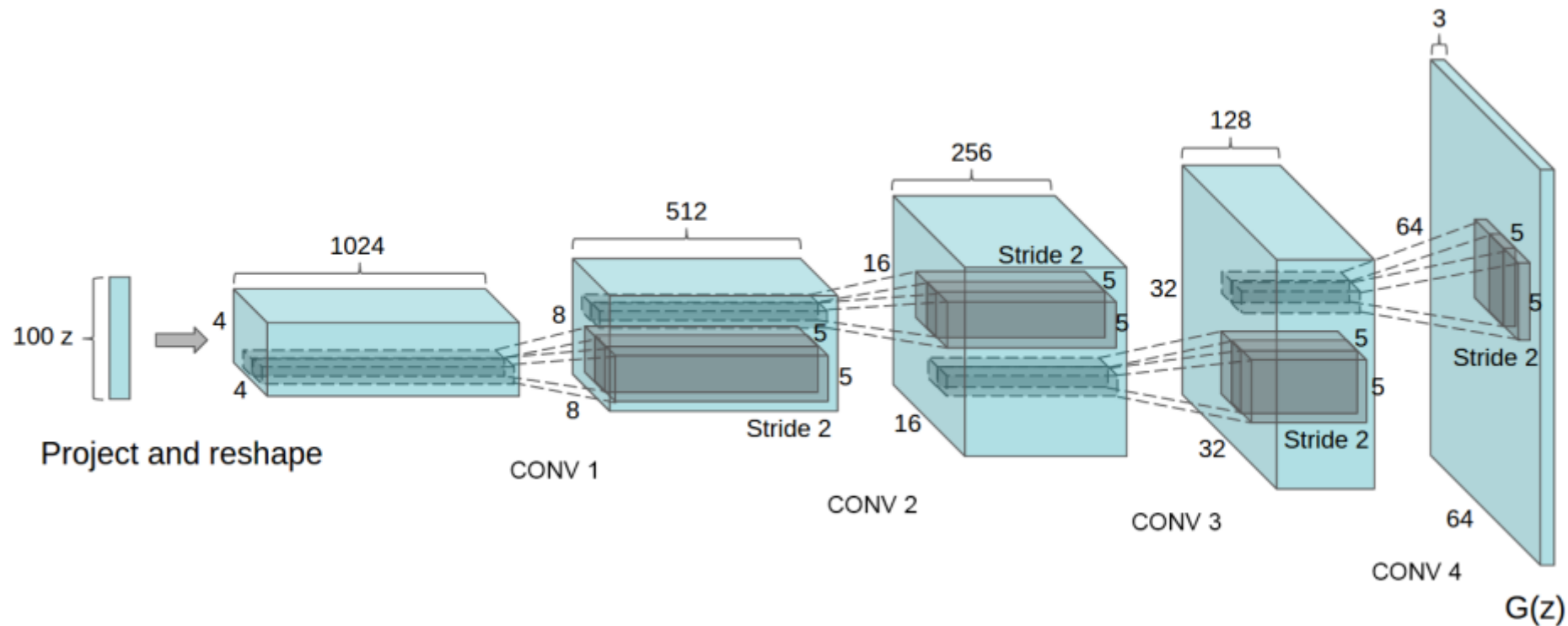
$$V(\theta^{(D)}, \theta^{(G)}) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

- ✓ 첫 번째 항: 진짜 샘플에 연관된 손실
 - ✓ 두 번째 항: 가짜 샘플에 대한 손실
 - ✓ G를 고정하면 $V(\theta(D), \theta(G))$ 를 최대화하는 것이 목적 함수가 됨
 - ✓ 즉, 판별자가 진짜와 가짜 이미지를 더 잘 구분하도록 만드는 것
- 가치함수의 최적화 - 두 최적화 단계를 교대로 수행하는 것
 - ✓ (1) 한 신경망의 가중치를 고정(동결)하고 다른 신경망의 가중치를 최적화
 - ✓ (2) 두 번째 신경망을 고정하고 첫 번째 신경망을 최적화

DCGAN

- DCGAN

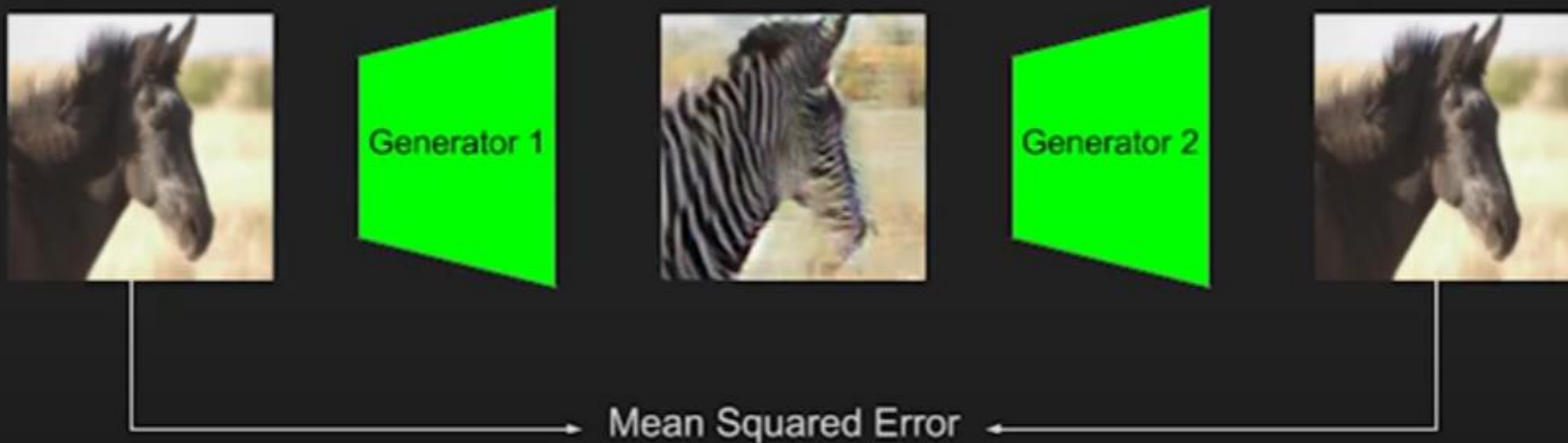
- ✓ 같은 접근을 CNN을 붙여서 해봅시다



CycleGAN

- <https://www.youtube.com/watch?v=-8hfnlxEPn4>

CycleGAN Architecture - Cycle-Consistency Loss



더 찾아보면 좋은 것들

- (수학을 이해하며) AutoEncoder 와 친구들
- (수학을 이해하며) GAN 과 친구들
- Diffusion model
- 실제로 해보면 더 재미있어요(with 두통)

