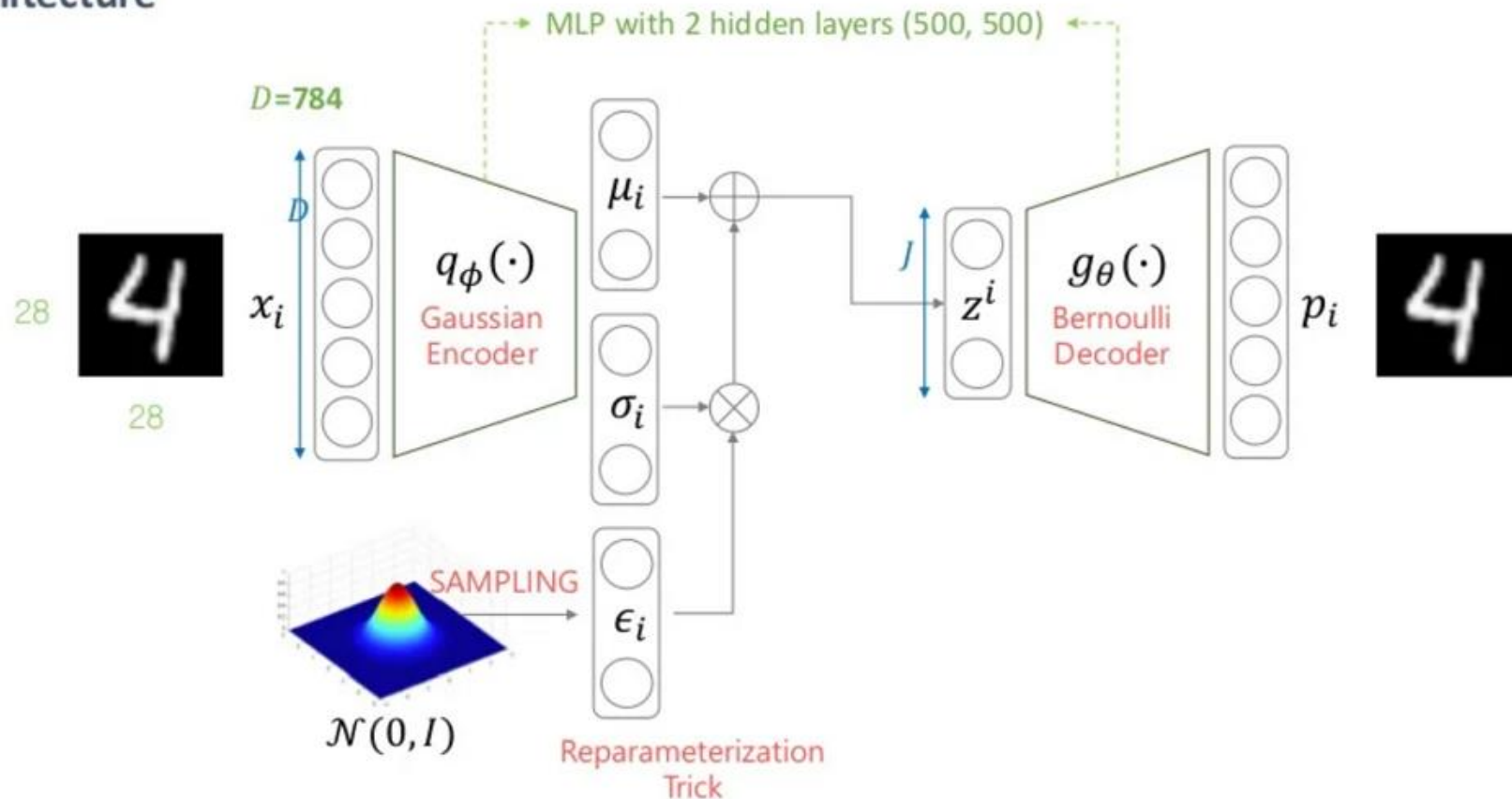


AI 12주차 생성 모델, Stable Diffusion

이미지(데이터) 생성

Variational Auto Encoder (VAE)

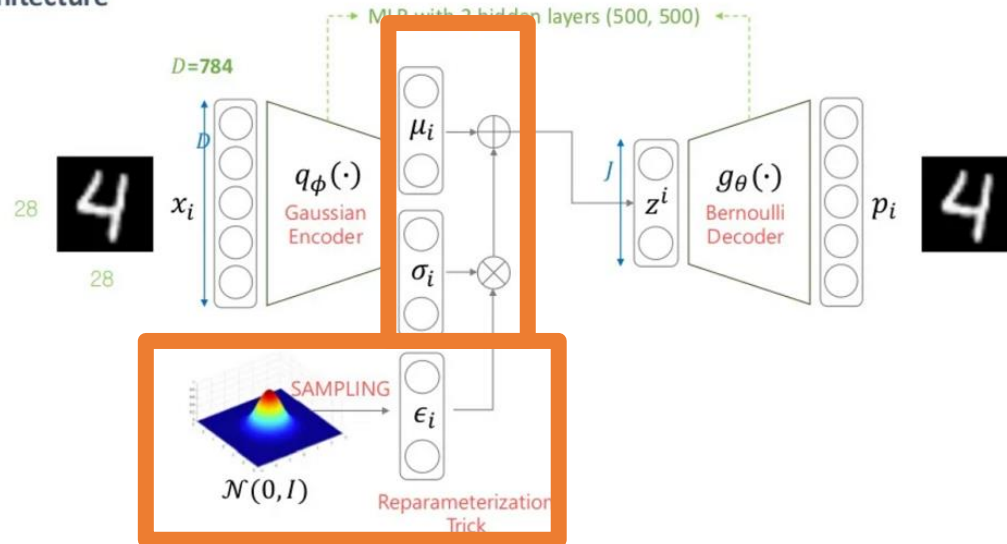
Architecture



*여기에서 $\mathcal{N}(0,1)$ 은 0~1 사이의 실수 Random 값이다.

확률 분포 추가된 Auto Encoder

Architecture



- 확률 분포를 적용
 - 같은 입력값 -> 각자 다른 출력
- 확률 분포의 불확실성
 - 다양한 확률에 의한 잠재 공간이 있기에 정적이 아닌 동적으로 생성 가능.

항목	AutoEncoder	VAE (Variational Autoencoder)
확률적 접근	결정적 (단일 값)	확률적 (확률 분포)
잠재 변수 처리	고정된 값으로 변환	확률 분포로 변환 (평균과 분산)
손실 함수	재구성 오류 (단순히 입력과 출력의 차이)	재구성 오류 + KL-발산 (분포 차이 고려)
생성 능력	새로운 데이터 생성에 제한적	새로운 데이터 생성 가능 (생성 모델)
잠재 공간 해석력	제한적	연속적이고 해석 가능한 잠재 공간

GAN 방식?



판별자 D (Discriminator)

“진짜 화폐랑 G가 만들어내는 가짜 화폐를
잘 구별해야지!!”

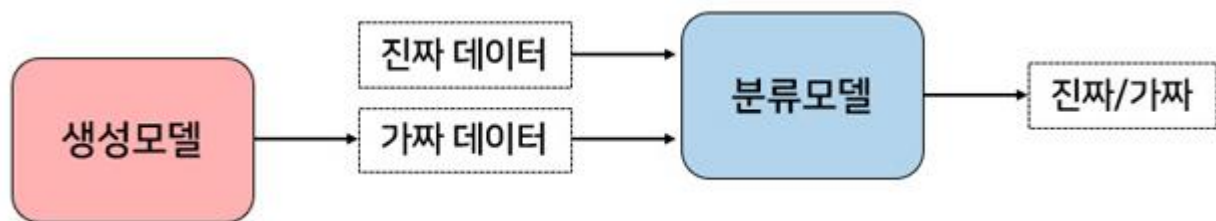
VS



생성자 G (Generator)

“진짜 화폐랑 내가 만들어낸 가짜 화폐를
D가 구분할 수 없도록
완벽하게 진짜같은 가짜 화폐를 만들어낼거야!!”

GAN 구조



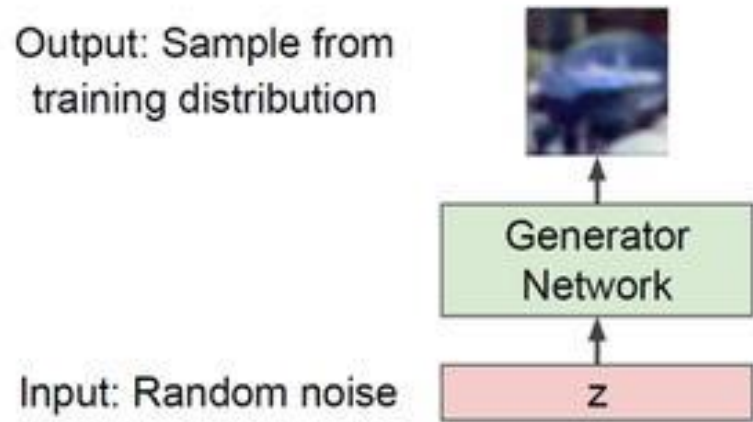
1. 생성 모델 : 진짜와 유사한 데이터를 생성

- 최종목표는 가짜와 진짜 이미지를 구별할 수 없는 수준으로 가짜 이미지 생성하는 것.

2. 분류 모델 : 진짜 이미지와 가짜 이미지를 구별하는 목적

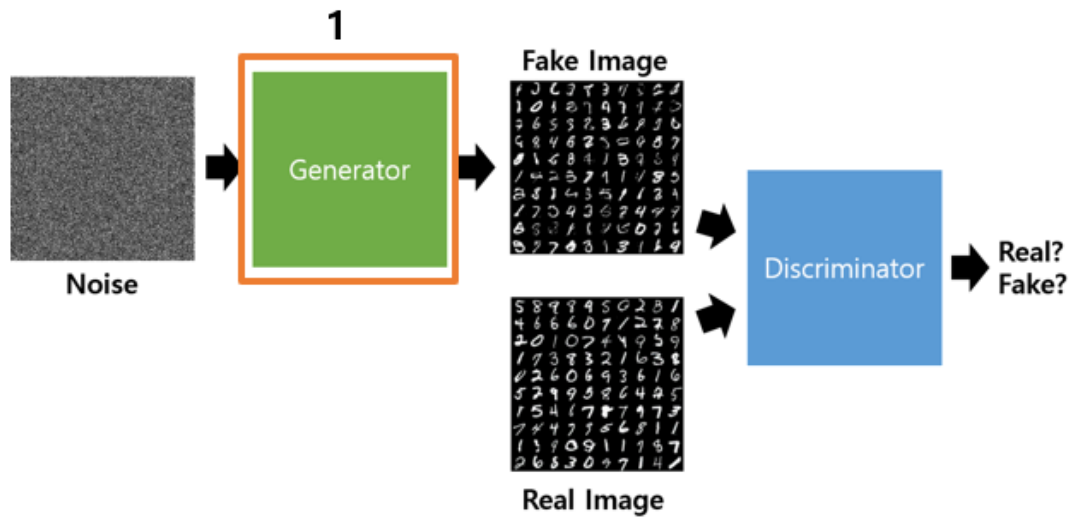
- 최종 목표 : 가짜 이미지를 가짜라고 잘 분류하는 것.

생성 모델



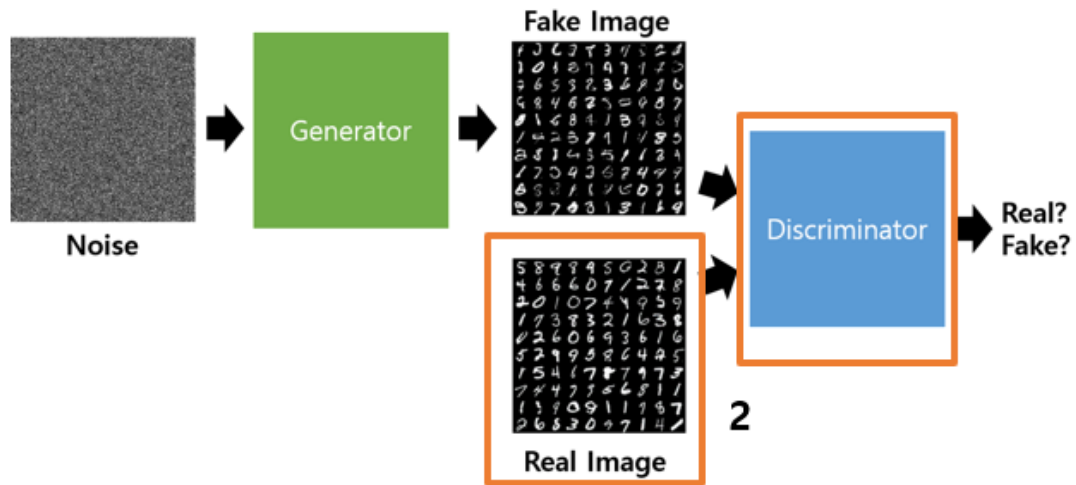
- 특정 모델을 기반으로 특정 데이터로 학습
- Z에서는 알 수 없는 데이터를 기반으로 Input
- 생성하고자 하는 대상과 유사하게 Output
- 초기 생성 모델은 오토 인코딩 방식으로 구조

GAN 처리 순서



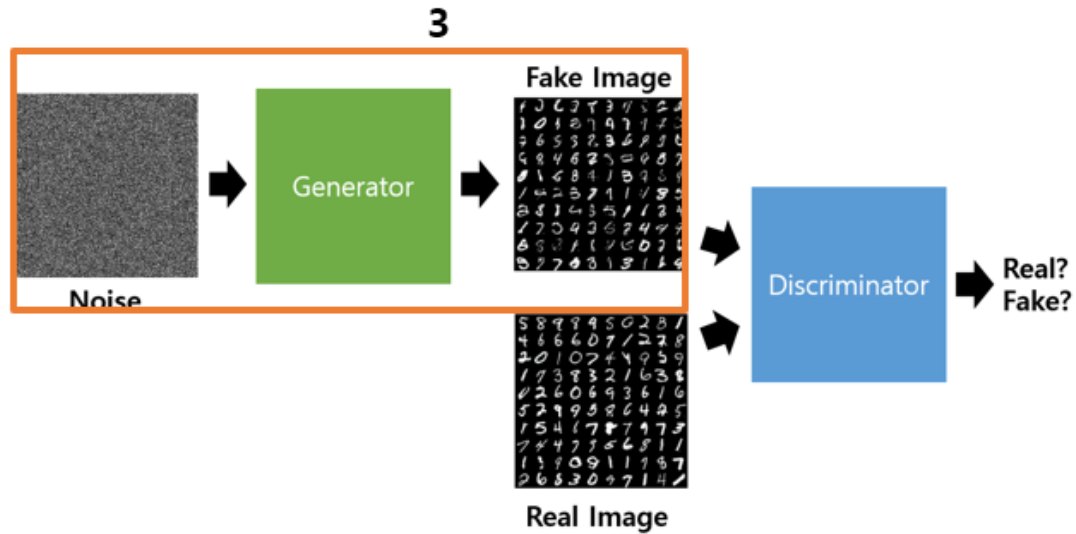
• 생성자 모델 생성

GAN 처리 순서



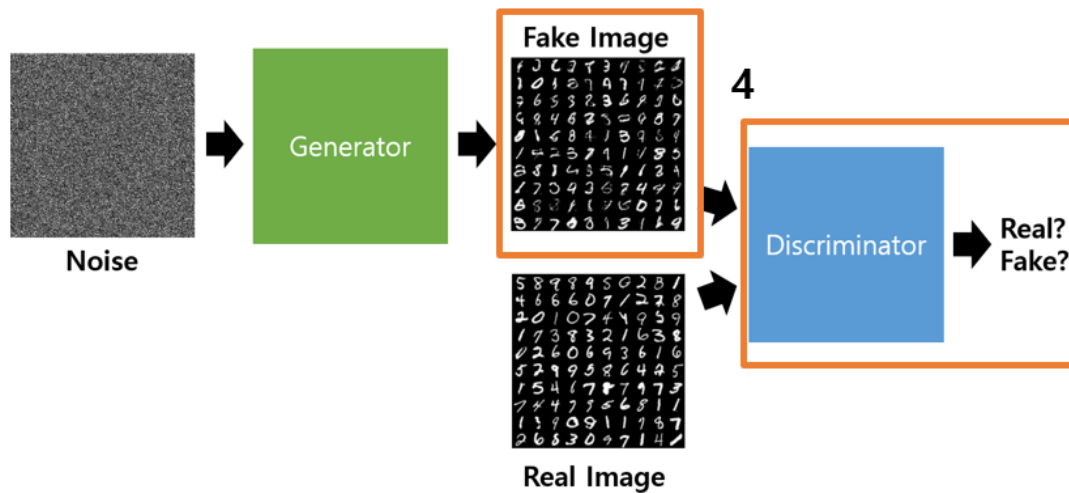
- 판별자 모델 생성
- 판별자 모델을 실제 데이터를 기반으로 학습

GAN 처리 순서



- 생성자 모델 구동할 Input 생성(노이즈 생성)
- 생성자 모델 구동

GAN 처리 순서



- 가짜, 진짜 이미지를 판별자를 통해 판별
- 판별한 결과값을 통해 loss 추출
- 가짜에 대한 loss를 생성자에게 학습
- 가짜 + 진짜에 대한 loss를 판별자에게 학습

Cycle GAN

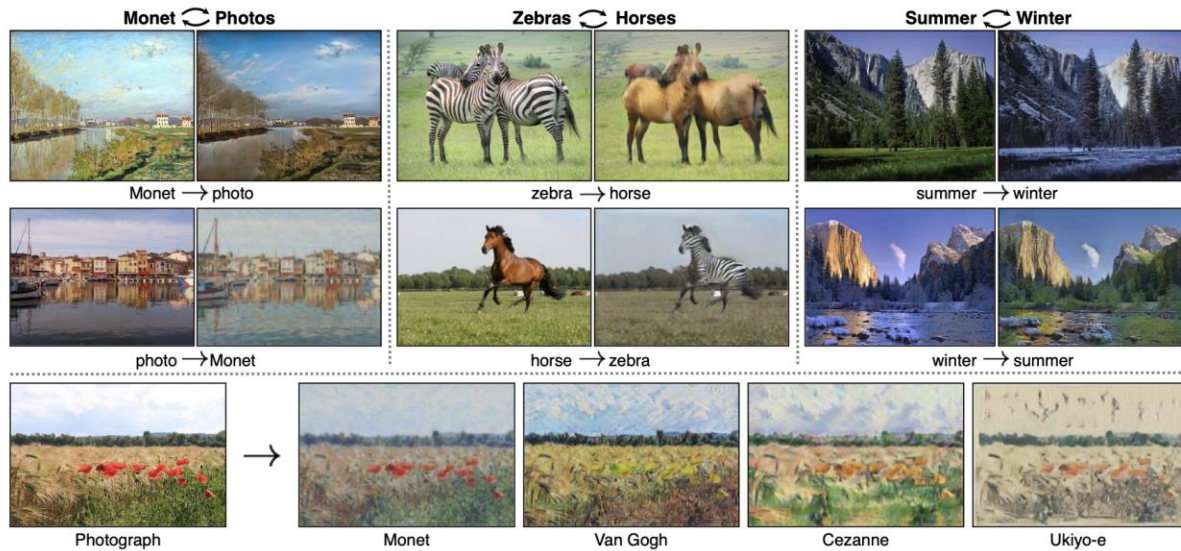


Figure 1: Given any two unordered image collections X and Y , our algorithm learns to automatically “translate” an image from one into the other and vice versa: (*left*) Monet paintings and landscape photos from Flickr; (*center*) zebras and horses from ImageNet; (*right*) summer and winter Yosemite 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.

- 특징이 다른 두 도메인(이미지) 데이터만 가지고 온전히 변환

SRGAN



LR image

SRGAN



4x HR image

- 생성자는 저해상도 이미지를 입력으로 가짜 고해상도 이미지를 생성
- 판별자는 가짜, 진짜 고해상도 이미지 판별

DCGAN?

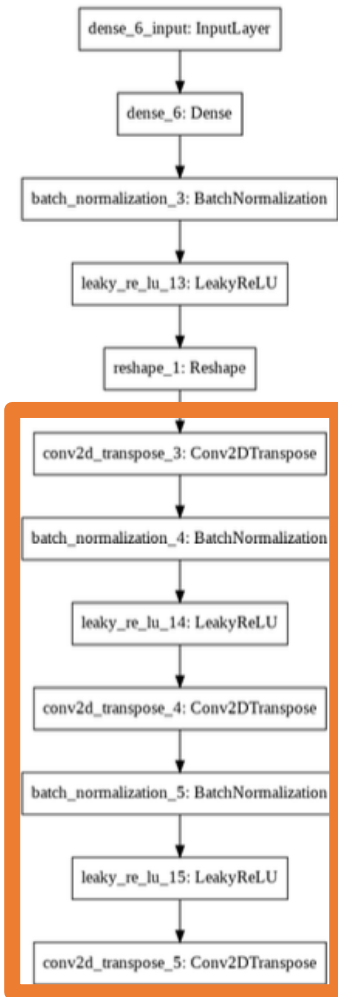


Fig. 5: Generator for DCGAN

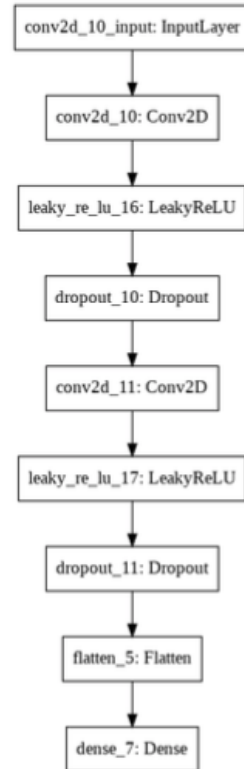
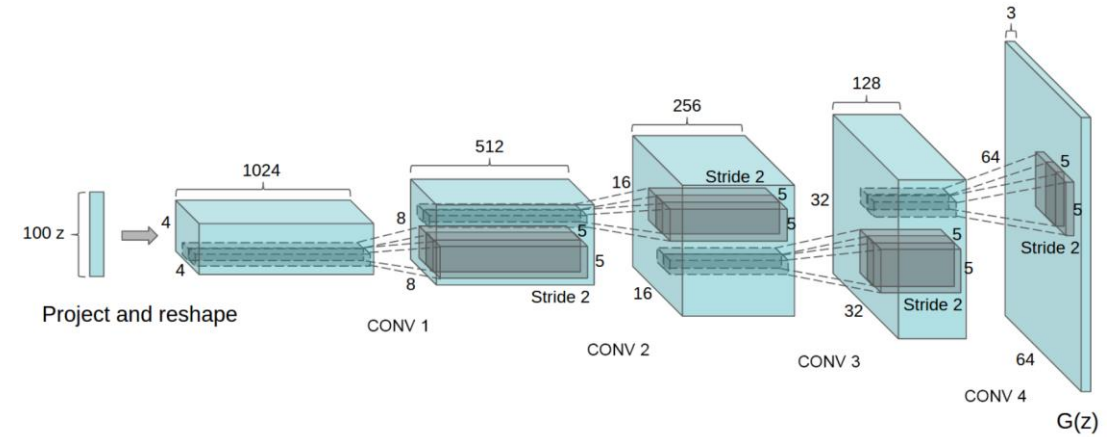
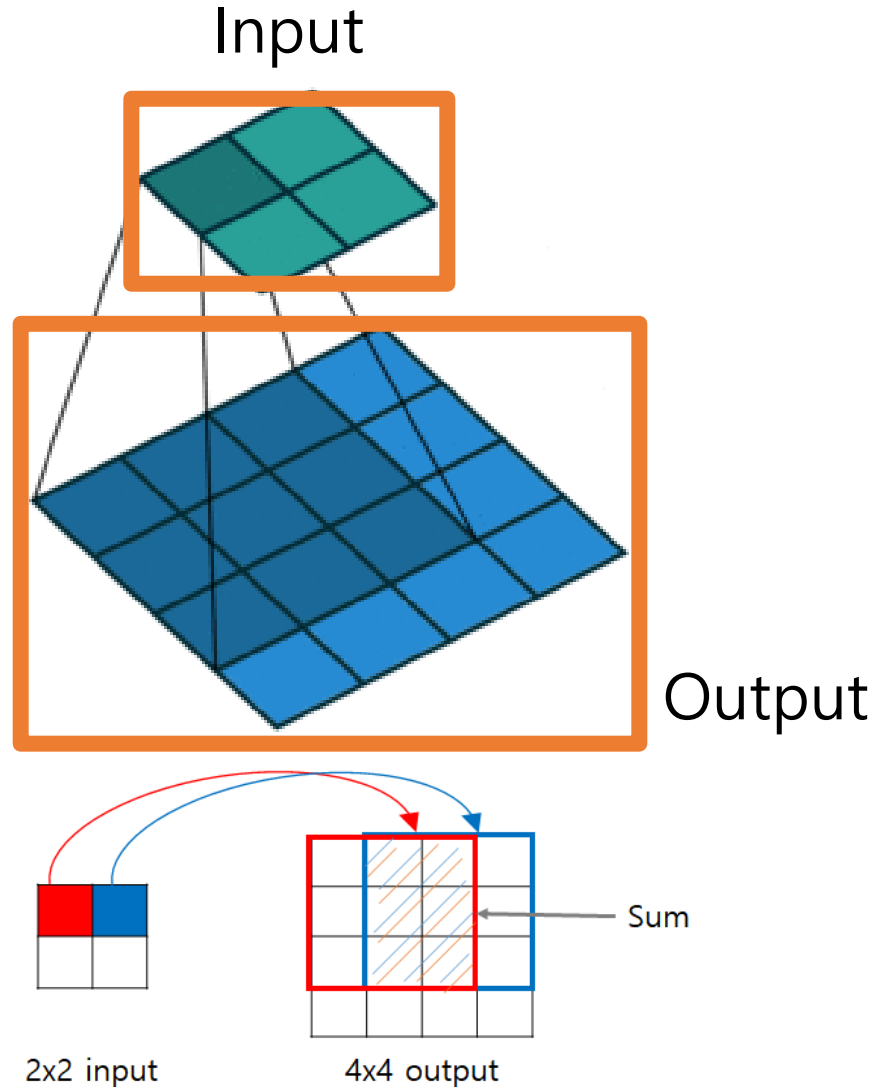


Fig. 6: Discriminator for DCGAN



- 생성 모델을 CNN를 연산을 역으로 함
 - (Transposed Convolution)
- 일반적인 CNN 연산은 downsampling
- Transposed Convolution 는 upsampling

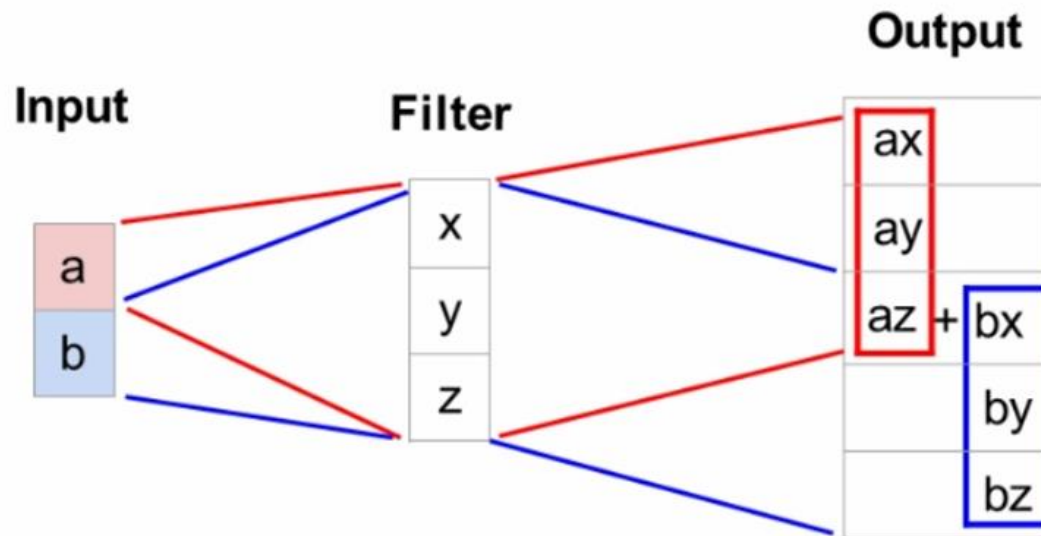
Transposed Convolution (Upsampling)



- Transposed라는 단어를 붙여진 것으로 CNN 연산의 반대 의미
- CNN 연산하면 할수록 OutPut 사이즈가 작아짐
 - 작아짐 = downsampling
- CNN 역으로 계산하는 것은 OutPut 사이즈가 커짐.
 - 커짐 = Upsampling
- Convolution 적분한 것임. 적분하는 과정에서 완벽하게 복구는 못하나, 근사해짐.
 - 학습을 통해 최적의 Kernel 탐색

Transposed Convolution 연산구조

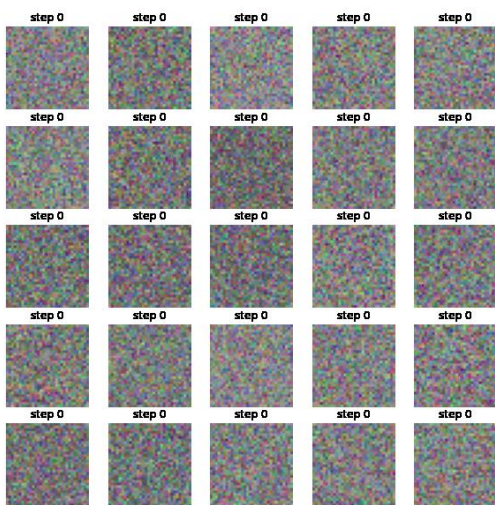
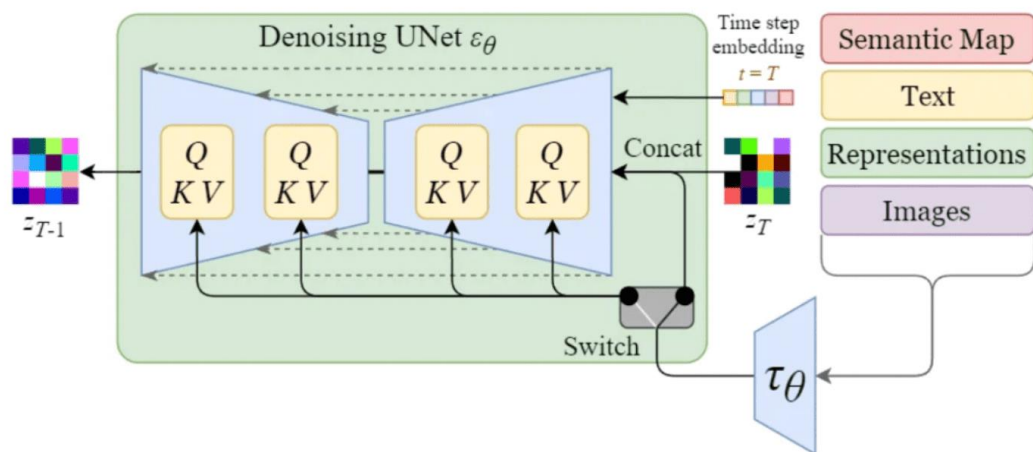
Transpose Convolution: 1D Example



- 말 그대로 CNN 역 연산 작용
- 옆 이미지처럼 Input Filter 를 각 자리에 곱하여 각 자리에 놓아두는 형태
- CNN 옵션 중 padding, stride에 따라 Output 사이즈 영향

(Stable)Diffusion

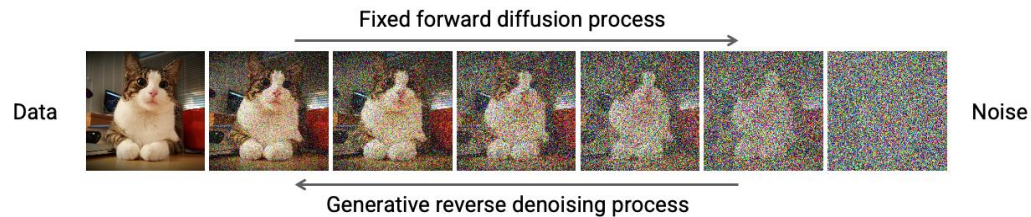
Stable Diffusion?



- 안정적으로 확산(생성)하기 위하여
 - 입력값(Text, Image, Sematic, etc...)
 - 텍스트 인식(**CLIP** 모델)
 - Denoising model(**U-Net**, 인코더-디코더 방식(**VAE**))
 - **Attention** 모델 포함.

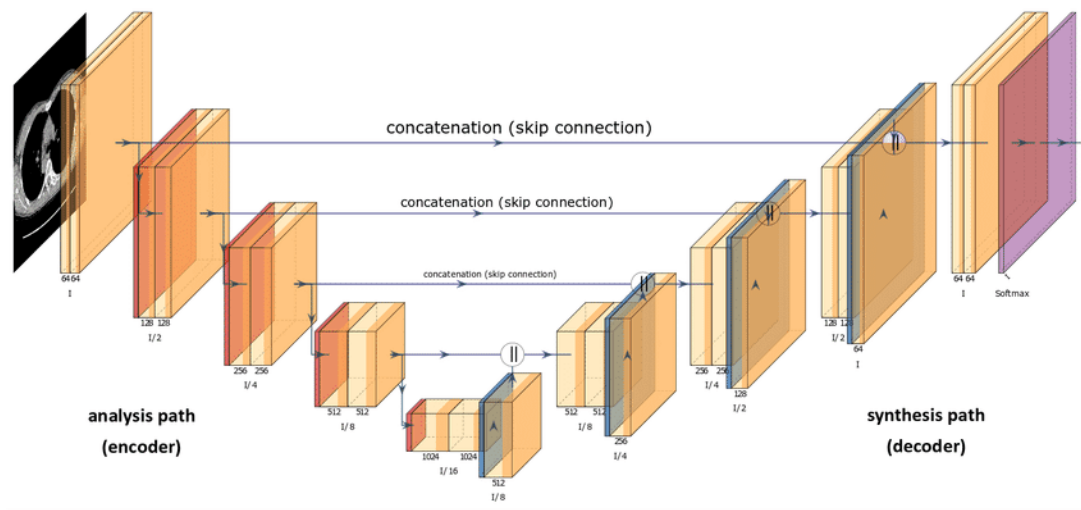
...

Diffusion 모델 목표



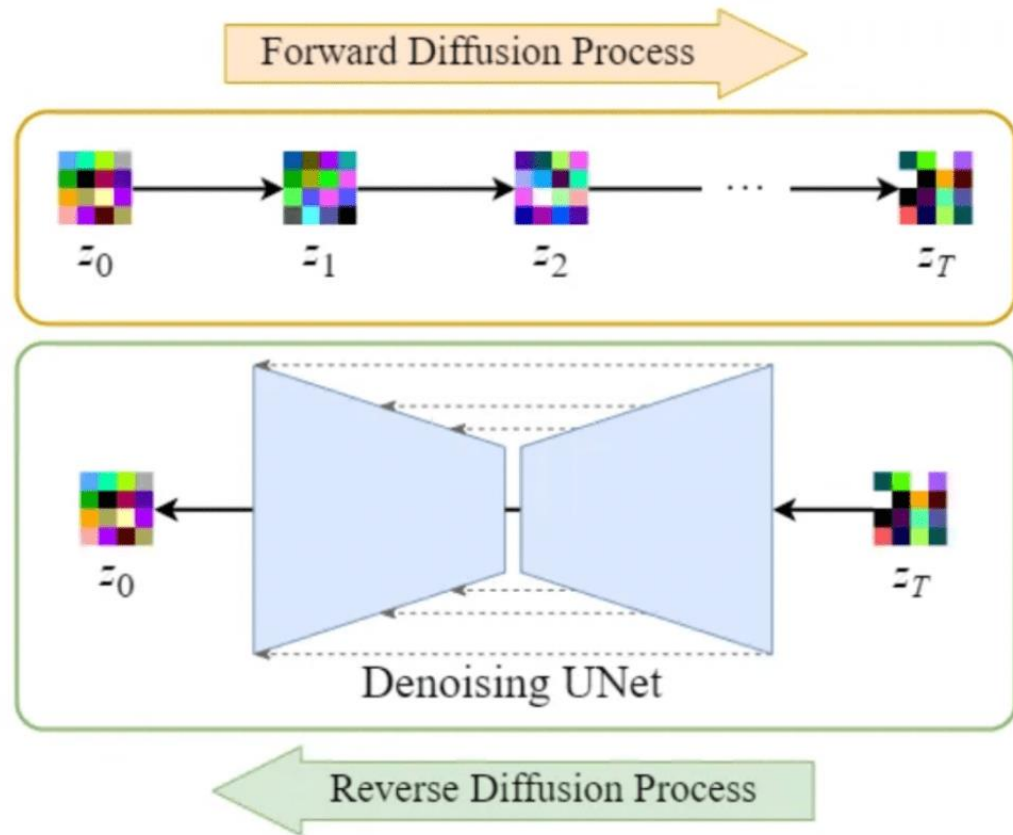
- 노이즈를 줄이면서 우리가 원하는 이미지(데이터)로 하는 것
- 이미지를 줄이는 Denoise 기법을 사용해야함.
 - U-Net 활용

U-Net



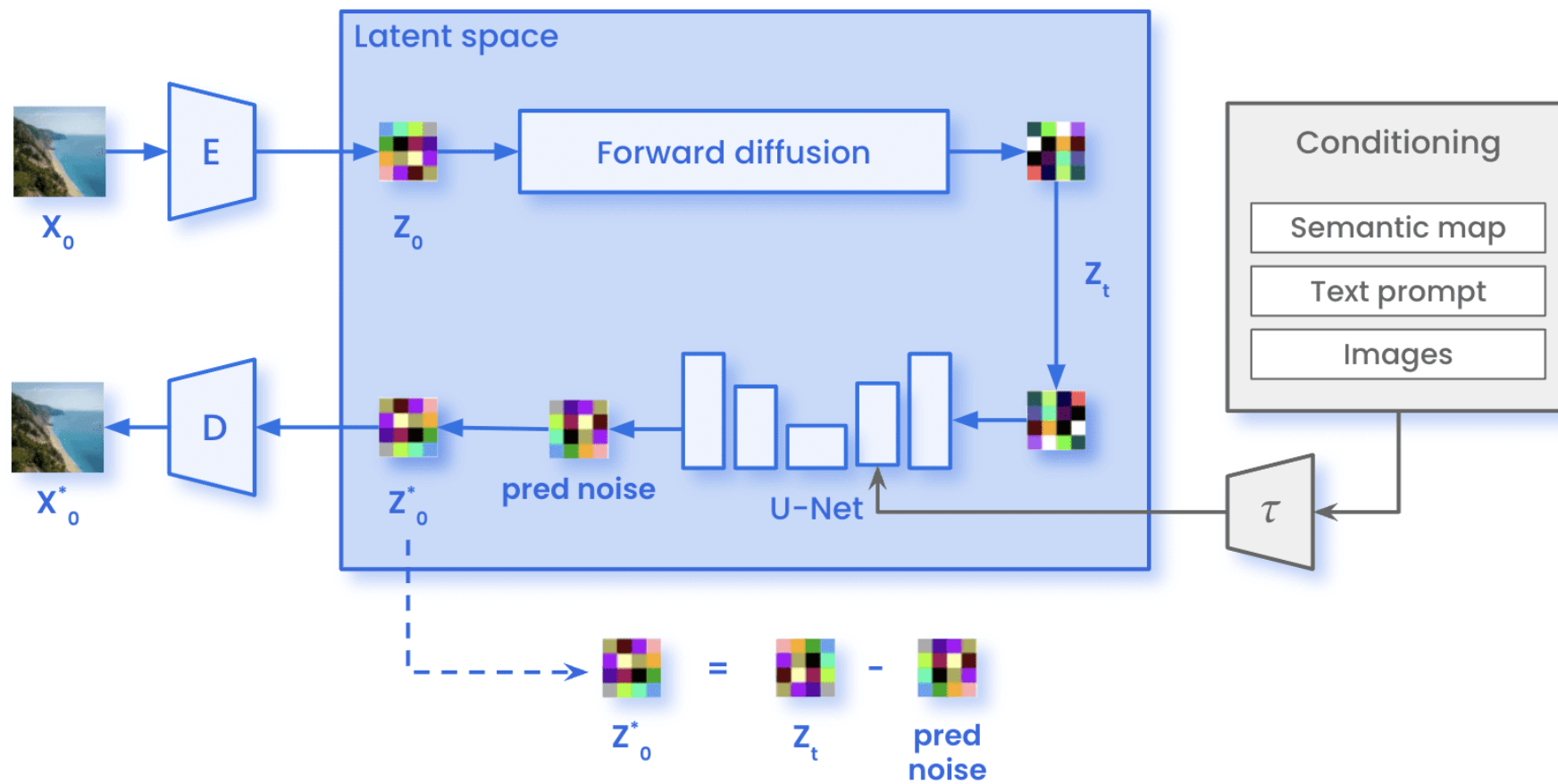
- 우리가 알고 있는 **AutoEncoder** 방식과 유사
 - CNN 모델을 기반으로 이미지 특성 추출 및 CNN 역 계산함으로써 이미지 생성
 - **Skip connection**를 이용하여 각자 레이어에 다른 레이어들간의 결합함.
- 이러한 과정을 통해 **특성추출 및 복원에 쓰임**

Diffusion 작동 방식



- 주어진 노이즈를 복원함(U-Net 구조)으로써(Denoise) 점차 이미지로 확산함.

전체 간략화 구조



Denoise 여러 번 수행방안

