

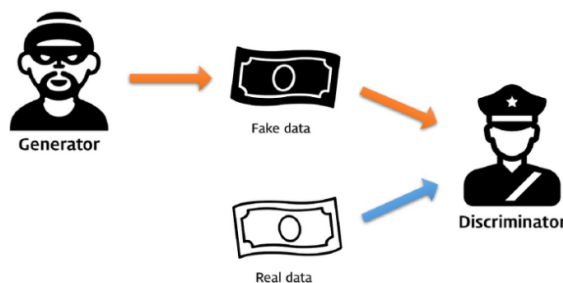
Generative Adv

GAN, VAE, Flow-based model, Diffusion 개념, 구조, 차이점

GAN

GAN은 Generative Adversarial Networks의 줄임말로, 생산적 적대 신경망이다. 적이 있다는 말은 상대가 있다는 말로 Real Image와 Generate된 Image를 적대적으로 학습시키며, 실제 데이터와 비슷한 데이터를 생성하도록 학습된다.

GAN은 실제 데이터와 비슷한 이미지를 생성하고자 하는 생성자 **generator**와 데이터를 구별하는 판별자 **discriminator**로 구성된다. 생성자는 끊임없이 거짓 예제를 실제 데이터와 비슷하게 만들고, 판별자는 실제 데이터와 만들어진 가짜 데이터를 구별하는 것으로 학습한다. 이것이 GAN의 목적이라고 할 수 있다.



위조지폐범인 Generator는 끊임없이 Real data와 비슷한 Fake data를 만들고, 경찰인 Discriminator는 둘을 구별하도록 학습한다.

<https://velog.io/@wo7864/GAN-개념-이해>

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

위의 식은 GAN의 목적함수를 나타낸 것이다. 함수 D는 얼마나 진짜 같은지를 나타내는 확률로, D는 maximize되는 방향으로 학습한다. 원본 데이터 x에 대해서는 1로 분류할 수 있도록, 즉 real로 분류할 수 있도록 학습되는 것이다.

가짜 이미지 G(z)가 real일수록 D(G(z))는 1에 가까워지고, 1에서 빠지므로 $\log(1 - D(G(z)))$ 는 작은 값을 가진다. 즉 생성자는 자신이 만든 이미지가 실제와 비슷하도록 학습을 진행한다.

VAE

Variational autoencoder, 줄여서 VAE는 generative model의 한 종류로, input image x을 잘 설명하는 **feature**를 추출하여 **Latent Vector z**에 담고, 이 Latent Vector z를 통해 x와 유사하지만 완전히 새로운 데이터를 생성하는 것이 목표다.

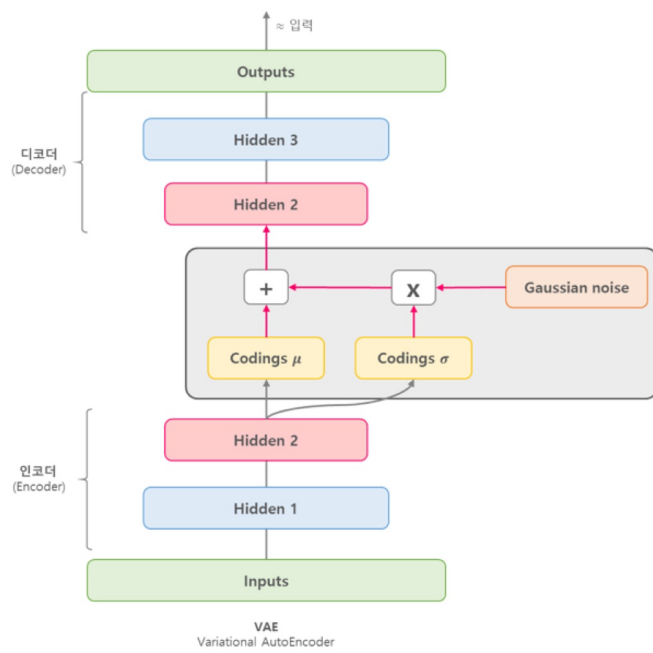
이때 각 feature는 가우시안 분포를 따른다고 가정하고, latent z는 각 feature의 평균과 분산값을 나타낸다. latent vector로부터 이미지를 생성한다고 보면 된다.

$$z \rightarrow x \\ p_{\theta^*}(z) \rightarrow p_{\theta^*}(x|z^{(i)})$$

$p(z)$: latent vector z의 확률밀도 함수로 가우시안 분포를 따른다고 가정한다. 즉, latent vector z를 sampling할 수 있는 확률밀도함수이다.

$p(x|z)$: 주어진 z에서 특정 x가 나올 확률에 대한 확률밀도함수이다.

θ : 모델의 파라미터

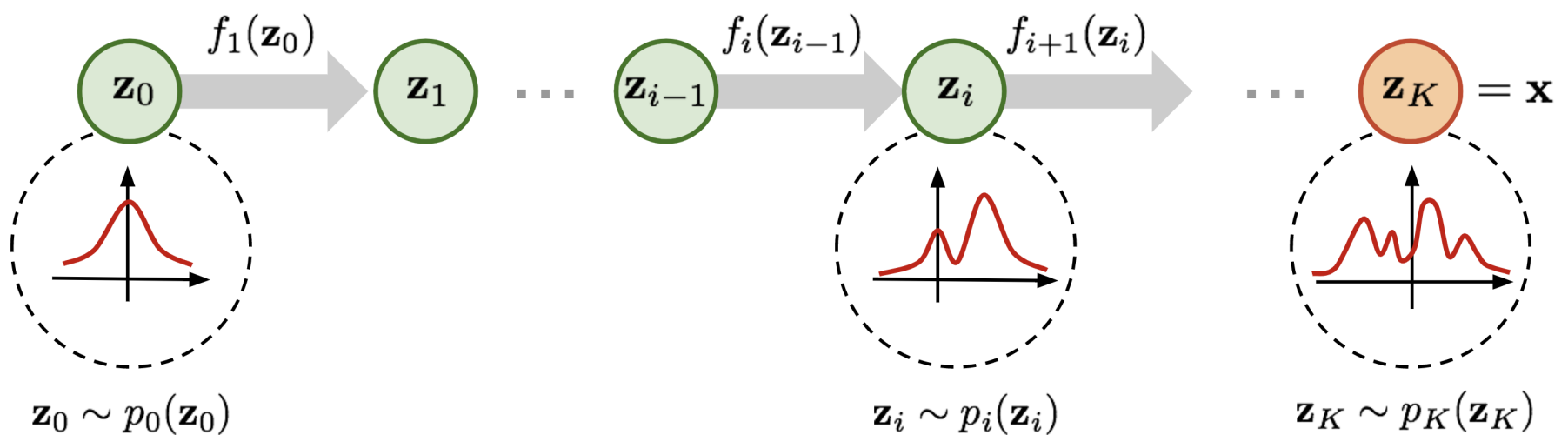


VAE는 Input image x를 Encoder에 통과시켜 Latent vector z를 구하고, Latent vector z를 다시 Decoder에 통과시켜 기존 Input image와 비슷하지만 새로운 이미지 X를 찾아내는 구조를 가지고 있다.

VAE는 Input Image가 들어오면, 그 이미지에서의 다양한 특징들이 각각의 확률 변수가 되는 어떤 확률 분포를 만들게 된다. 이런 확률 분포를 잘 찾아내고, 확률값이 높은 부분을 이용하면 실제에 있을 법한 이미지를 새롭게 만들 수 있다.

Flow-based model

Flow-based generative model은 데이터의 분포에서 학습하여, 연속적인 역변환을 통해 새로운 데이터를 생성하는 모델이다. 잠재 벡터 z의 확률 분포에 대한 일련의 역변환(a sequence of invertible transformations)을 통해 데이터 x의 분포를 명시적으로 학습하여 이를 negative log-likelihood로 해결한다.



Normalizing Flow(NL)는 어떠한 확률 분포(z)에 연속적인 **역변환 함수**를 적용해서 실제의 확률 분포(x)를 예측하는 방식이다.

z_{i-1} 에서 z_i 로 역변환을 하는 수식은 다음과 같다.

$$\begin{aligned}
 p_i(z_i) &= p_{i-1}(f_i^{(-1)}(z_i)) \left| \det \frac{df_i^{-1}}{dz_i} \right| \\
 &= p_{i-1}(f_i^{(-1)}(z_{i-1})) \left| \det \left(\frac{df_i}{dz_{i-1}} \right)^{-1} \right| \\
 &= p_{i-1}(f_i^{(-1)}(z_{i-1})) \left| \det \frac{df_i}{dz_{i-1}} \right|^{-1} \\
 \log p_i(z_i) &= \log p_{i-1}(z_{i-1}) - \log \left| \det \frac{df_i}{dz_{i-1}} \right|
 \end{aligned}$$

이러한 역변환을 일련의 과정을 거치면 x의 확률분포를 구할 수 있다. z_0 에서 K번의 역변환을 통해 x의 확률분포를 구해보자.

$$\begin{aligned}
x &= z_K = f_K \circ f_{K-1} \circ \dots \circ f_1(z_0) \\
\log p(x) &= \log \pi_K(z_K) = \log \pi_{K-1}(z_{K-1}) - \log \left| \det \frac{df_K}{dz_{K-1}} \right| \\
&= \log \pi_{K-2}(z_{K-2}) - \log \left| \det \frac{df_{K-1}}{dz_{K-2}} \right| - \log \left| \det \frac{df_K}{dz_{K-1}} \right| \\
&= \dots \\
&= \log \pi_0(z_0) - \sum_{i=1}^K \log \left| \det \frac{df_i}{dz_{i-1}} \right|
\end{aligned}$$

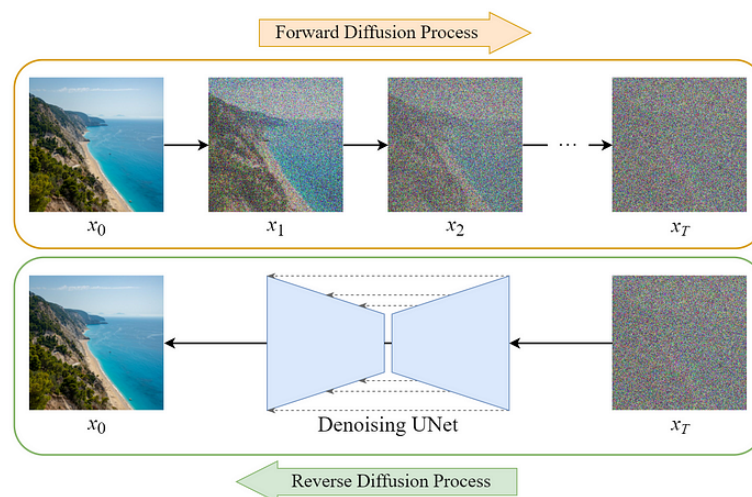
$\log p(x)$ 를 정의했다면 학습 데이터셋 D 에 negative log-likelihood(NLL)을 만들어 생성 모델에게 넘겨주면 된다.

$$L(D) = -\frac{1}{|D|} \sum_{x \in D} \log p(x)$$

그러나 위의 식이 가능하게 하려면 함수 f 는 가역적(invertible)이어야 하고, f 에 대한 자코비안 행렬식은 계산하기 쉬워야 된다는 조건이 필요하다.

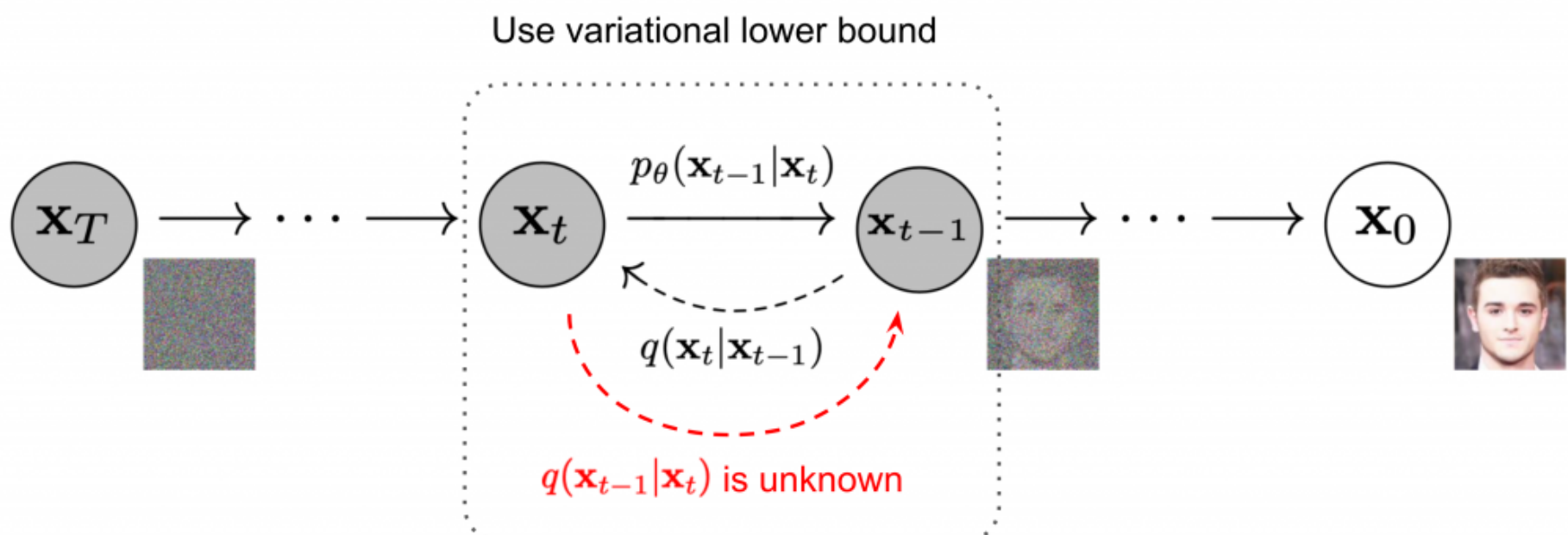
Diffusion

Diffusion model은 입력 이미지에 정규 분포를 따르는 noise를 여러 단계에 걸쳐 추가하고, 여러 단계에 걸쳐 noise를 제거함으로써, 입력 이미지와 유사한 확률 분포를 가진 결과 이미지를 생성하는 모델이다. 입력 이미지가 서서히 확산하기 때문에 Diffusion 이라는 이름이 붙었다.



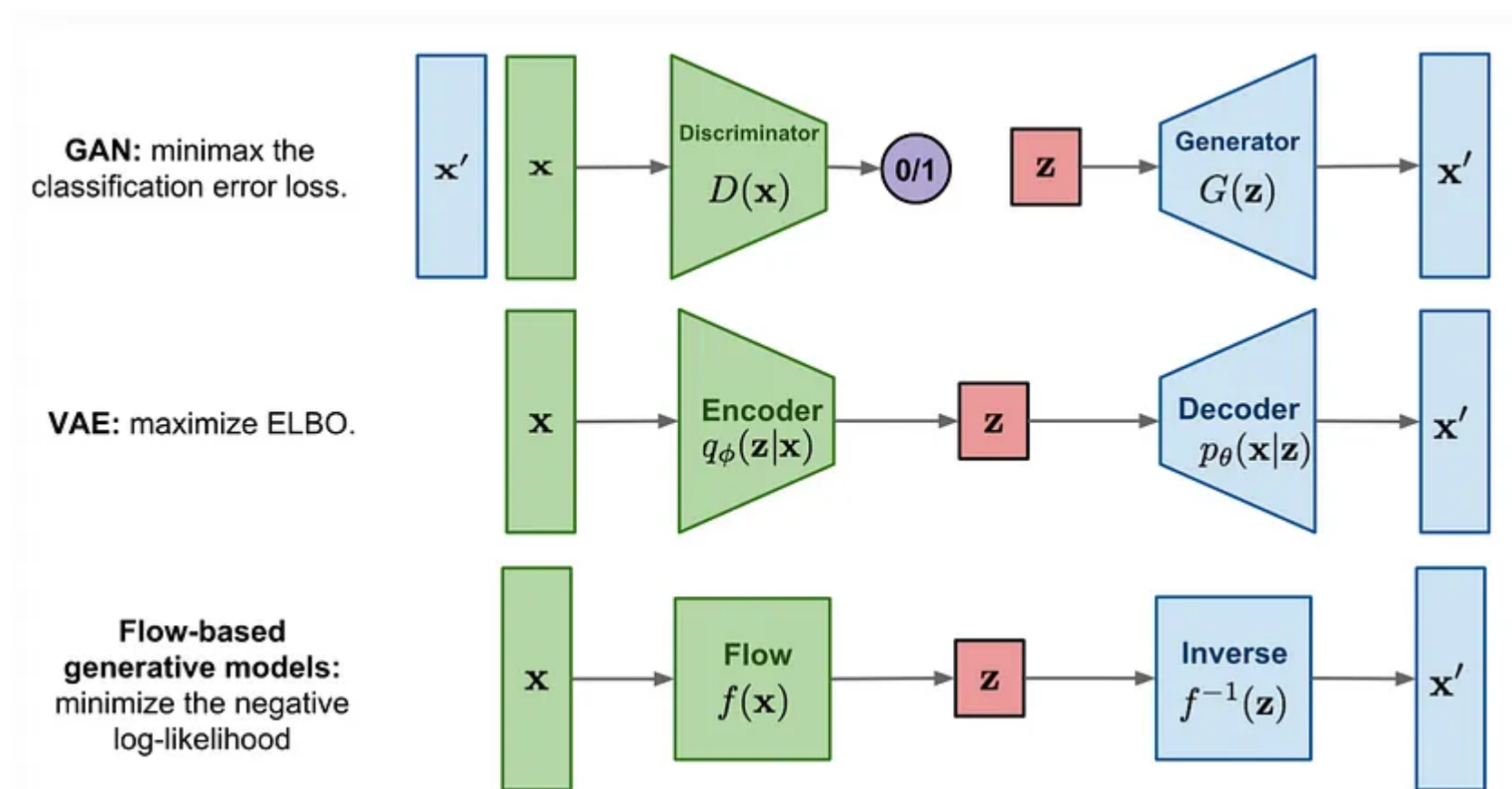
Forward Diffusion Process에서는 이미지에 고정된(fixed) 정규 분포(Gaussian 분포)를 따르는 noise가 더해지고, Reverse Diffusion Process에서는 이미지를 학습된 정규 분포를 따르는 noise이미지로 빼준다.

Diffusion 모델은 Forward → Reverse 단계를 거친 결과 이미지를 입력 이미지의 확률 분포와 유사하게 만든다. 이를 위해 reverse 단계에서 noise 생성 확률 분포 parameter인 평균과 표준편차를 업데이트하며 학습이 진행된다.



이전 상태($t-1$)이 주어질 때 현재 상태(t)가 될 확률분포 q 는 평균(μ)과 분산(σ^2)으로 구성된 가우시안 분포를 따른다. 이 식을 noise 크기 parameter인 β 를 포함하여 다르게 표현하면, 이전 상태($t-1$)의 이미지가 β 만큼 다른 pixel을 선택하게 되고, $\text{root}(1-\beta)$ 만큼 이전 pixel값을 선택하게 하는 식으로 정의할 수 있다. t 가 점점 증가함에 따라 β 가 커짐으로써 noise가 강해진다.

차이점



GAN은 discriminator에서, VAE는 encoder에서 원본 데이터를 압축한다. 그림에서도 볼 수 있듯이 사다리꼴 모양이다. 원본 데이터를 압축 및 확장하는 과정에서 데이터의 손실이 발생할 수 있다. 반면에 flow-base 모델은 역함수 변환을 통해 데이터의 손실을 줄일 수 있다.

flow 모델은 층 사이의 변환을 결정론적(deterministic) 함수로 학습하고, 이 함수를 통해 노이즈 분포에서 뽑힌 무작위 값을 데이터 분포가 정의된 공간으로 보내 데이터를 생성한다. 결정론적 함수의 경우 입력과 출력 사이의 역변환(일대일대응)이 잘 정의되어야 하므로 함수는 상대적으로 제한적이다. 반면에 diffusion model에서 사용하는 확률과정은 초기 시점의 데이터 분포를 복원할 수 있으면 가역적이라고 말한다. forward process의 전이 확률분포를 가지고 시간 축을 거꾸로 돌린 reverse process의 전이 확률 분포를 추정할 수 있다면, 역확률과정을 통해 데이터 분포에서 샘플링하는 것이 가능해지므로 가역적 변환이 된다.

Diffusion model은 Flow model처럼 생성이 결정론적이지 않고 확률적 변동성을 가지면서 샘플링 과정이 이루어진다.