

과제1

Generative Models

GAN

[구조적 특징]

GAN은 생성자(Generator)와 판별자(Discriminator) 두 개의 신경망으로 구성된다. 생성자는 random noise에서 시작하여 실제 데이터와 유사한 데이터를 생성하려 하고, 판별자는 제공된 데이터가 실제 데이터인지 생성자가 만든 것인지를 판별한다.

생성자(Generator):

생성자는 random noise로부터 데이터를 생성한다. 이 노이즈는 latent space에서 샘플링되며, 생성자는 이를 실제 데이터와 유사한 데이터로 변환하는 함수를 학습한다. 생성자의 목표는 판별자를 속이고 판별자가 생성된 데이터를 실제 데이터로 분류하게 만드는 것이다.

판별자(Discriminator):

판별자의 역할은 입력된 데이터가 실제 데이터인지, 생성자가 생성한 가짜 데이터인지를 구별하는 것이다. 판별자는 이진 분류 문제를 해결하는 모델로, 생성된 데이터를 가능한 한 정확하게 식별해내고자 한다. 판별자의 목표는 생성자가 생성한 가짜 데이터를 가짜로, 실제 데이터를 실제로 정확히 분류하는 것이다.

GANs의 핵심 아이디어는 생성된 분포와 실제 분포 간의 직접적인 비교를 하는 대신에, 이 두 분포에 대한 downstream task의 형태로 간접적인 비교를 도입하는 것이다. 이 downstream task는 실제 샘플과 생성된 샘플을 구별하는 차별화 작업으로, 생성된 분포가 실제 분포에 점점 더 가까워지도록 생성 네트워크의 학습을 유도한다.

[학습 과정]

두 네트워크가 정의되면, 서로 반대되는 목표를 가지고 동시에 학습이 진행된다:

- 생성자의 목표는 판별자를 속이는 것이므로, 생성 신경망은 최종 classification error (between true and generated data)를 최소화하도록 학습된다.

- 반면에, 판별자의 목표는 진짜 생성 데이터를 탐지하는 것이므로, 판별 신경망은 최종 classification error를 최소화하도록 학습된다.

따라서 학습 과정의 각 반복에서, 생성 네트워크의 가중치는 분류 오류를 증가시키기 위해 업데이트된다(생성자 파라미터에 대한 오류 기울기 상승). 반면 판별 네트워크의 가중치는 이 오류를 감소시키기 위해 업데이트된다(판별자 파라미터에 대한 오류 기울기 하강).

$$\begin{aligned} E(G, D) &= \frac{1}{2} \mathbb{E}_{x \sim p_t} [1 - D(x)] + \frac{1}{2} \mathbb{E}_{z \sim p_z} [D(G(z))] \\ &= \frac{1}{2} (\mathbb{E}_{x \sim p_t} [1 - D(x)] + \mathbb{E}_{x \sim p_g} [D(x)]) \end{aligned}$$

$$\max_G \left(\min_D E(G, D) \right)$$

VAE

[구조]

VAE(Variational Autoencoder)는 훈련 과정에서 인코딩 분포가 규제되어 새로운 데이터를 생성할 수 있는 좋은 특성을 가진 latent space를 확보하도록 하는 autoencoder의 한 형태다. 여기서 autoencoder는 인코더와 디코더를 신경망으로 설정하고 반복 최적화 과정을 사용하여 최적의 인코딩-디코딩 방식을 학습하는 것으로 구성된다. 따라서 각 반복마다 autoencoder 구조(인코더 다음에 디코더)에 일부 데이터를 공급하고, 인코딩된 데이터를 초기 데이터와 비교한 후 오류를 역전파하여 네트워크의 가중치를 업데이트한다.

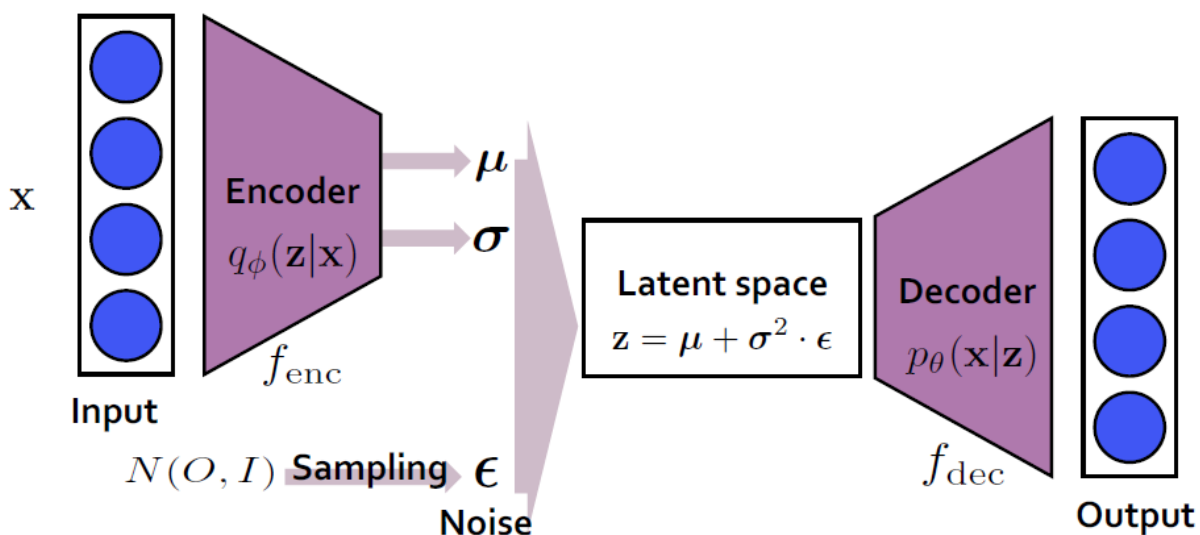
오토인코더는 단지 가능한 한 적은 손실로 인코딩 및 디코딩을 수행하도록 훈련되며, 잠재 공간의 구성 방식은 중요하지 않다. 따라서, 아키텍처의 정의에 주의를 기울이지 않으면, 네트워크 훈련 중에 과적합이 발생할 수도 있다.

⇒ autoencoder의 디코더를 생성 목적으로 사용하기 위해서는 잠재 공간이 충분히 규칙적이어야 한다. 이러한 규칙성을 얻기 위한 하나의 가능한 해결책은 훈련 과정 중에 *명시적인 규제* 를 도입하는 것이다. 따라서, VAE는 과적합을 피하고 생성 과정을 가능하게 하는 좋은 잠재 공간의 특성을 보장하기 위해 훈련이 규제된 오토인코더로 정의될 수 있다.

[학습 과정]

표준 오토인코더와 마찬가지로, VAE도 인코더와 디코더로 구성된 아키텍처이며, 인코딩된 데이터와 초기 데이터 사이의 재구성 오류를 최소화하기 위해 훈련된다. 그러나 latent space 의 규제를 도입하기 위해, 조금 다른 인코딩-디코딩을 가진다 - 입력을 단일 점으로 인코딩하는 대신, 잠재 공간 위의 분포로 인코딩한다. 그런 다음 모델은 다음과 같이 훈련된다:

1. 입력이 잠재 공간 위의 분포로 인코딩된다.
2. 그 분포로부터 잠재 공간의 점이 샘플링된다.
3. 샘플링된 점이 디코딩되고, 재구성 오류를 계산할 수 있다.
4. 재구성 오류가 네트워크를 통해 역전파된다.



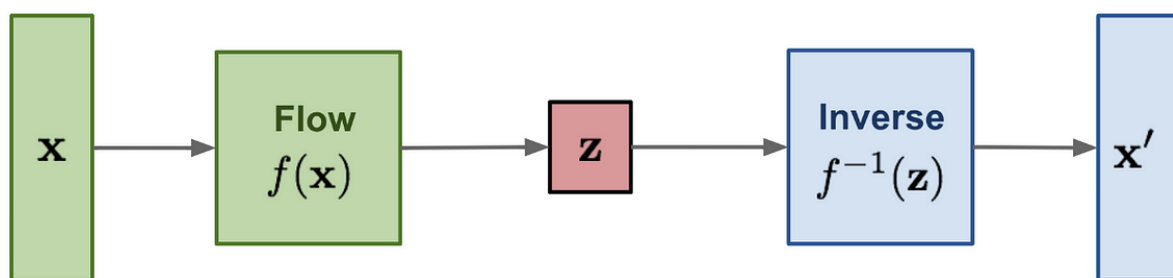
실제로, 인코딩된 분포로 정규분포를 선택하는데, 이는 인코더가 이러한 가우시안을 설명하는 평균과 공분산 행렬을 반환하도록 훈련하기 위함이다. input이 일정한 분산을 가진 분포로 인코딩되는 이유는 latent space 규제를 매우 자연스럽게 표현할 수 있기 때문이다: 인코더에 의해 반환된 분포는 표준 정규 분포에 가깝도록 만든다.

따라서, VAE를 훈련할 때 최소화되는 loss function은 "reconstruction term" (on the final layer)과 "regularization term" (on the latent layer)으로 구성된다. reconstruction 항은 인코딩-디코딩 방식을 가능한 한 효과적으로 만들려고 하고, regularization 항은 인코더에 의해 반환된 분포를 표준 정규 분포에 가깝게 만들어 latent space를 규제하려고 한다.

Flow-based Model

Flow-based generative models는 복잡한 데이터 분포를 학습하는 데 있어서 GAN(Generative Adversarial Networks)과 VAE(Variational Autoencoders)의 단점을 극복하기 위해 제안된 모델 유형이다. GAN과 VAE는 복잡한 데이터 분포를 모델링하는 데 뛰어난 결과를 보여줬지만, 정확한 확률 분포 평가와 추론이 어렵다는 한계가 있다. 이는 VAE에서의 낮은 품질 결과와 GAN 훈련 과정에서 모델 collapse, gradient loss 같은 문제로 이어졌다.

Flow-based generative models는 이러한 문제를 해결하기 위해 역변환 함수를 사용한다. 이 모델들은 이미지를 생성하기 위해 별도의 노이즈를 추가할 필요가 없으며, GAN에 비해 훈련 과정이 훨씬 더 안정적이다. 가장 중요한 특징 중 하나는 flow-based 모델이 실제 데이터 분포를 모델링하고, 데이터의 정확한 가능도(likelihood)를 제공한다는 점이다. (flow-based 모델은 negative log-likelihood를 손실 함수로 사용한다.)

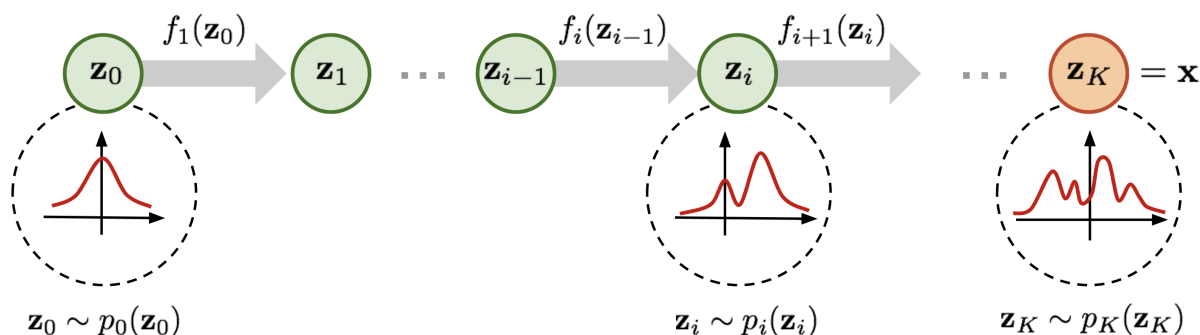


Flow-based generative models는 가역적인(invertible) 변환 함수를 사용하여 데이터 x 를 잠재 표현 z 로 매핑한다. 여기서 중요한 점은 z 가 x 와 동일한 형태를 가져야 한다는 것이다. 이러한 가역성은 모델이 복잡한 데이터 분포를 학습하면서도, 정확한 역변환을 통해 원래 데이터를 재구성할 수 있게 한다. 결과적으로, flow-based 모델은 높은 품질의 데이터 생성, 정밀한 분포 모델링, 그리고 안정적인 학습 과정을 장점으로 갖는다.

Flow-based generative models는 Normalizing flows를 활용해서 데이터의 복잡한 분포를 정확하게 학습하고, 이를 통해 높은 품질의 새로운 데이터 샘플을 생성할 수 있는 강력한 도

구인데, normalizing flows에 대해서도 간단히 짚고 넘어가보자.

Normalizing Flows 역할



Normalizing flows는 바로 이러한 가역적인 변환의 연속을 사용하여 복잡한 데이터 분포를 모델링한다. 초기에는 간단한 분포에서 시작해서, 일련의 변환 과정을 거치면서 점차 복잡한 데이터의 실제 분포를 학습한다. 중요한 점은, 이 변환 과정이 언제든지 역으로 수행될 수 있다는 것이다. 이를 통해, 복잡한 데이터 분포에서도 새로운 샘플을 효과적으로 생성할 수 있게 된다.

Flow-based generative models는 바로 이러한 Normalizing flows의 원리를 기반으로 한다. 모델은 여러 개의 가역적인 변환 단계를 거쳐 복잡한 데이터 분포를 학습한다. 각 변환 단계는 데이터를 조금씩 변형시키며, 이 변형 과정 전체를 거쳐서 최종적으로 원하는 복잡한 분포를 얻게 되는 것이다.

이러한 특성 때문에, Flow-based generative models는 데이터의 정확한 확률 분포를 학습할 수 있으며, 고품질의 데이터 샘플 생성이 가능해진다. 가역적인 변환의 연속을 사용하는 이 방법은, 모델이 복잡한 데이터 분포를 정밀하게 학습하면서도, 새로운 샘플을 효과적으로 생성할 수 있게 해 준다.

Flow-based model의 범주에 포함되는 모델들 중 대표적인 것으로는 Wavenet, Masked Autoregressive Flow, Inverse Autoregressive Flow 등이 있다고 한다.

Diffusion

[구조]

Diffusion 모델은 점진적으로 노이즈를 추가하여 데이터를 가우시안 분포로 변환하는 과정과, 그 반대 과정인 점진적인 노이즈 제거를 통해 원본 데이터를 복구하는 과정을 반복적으로 수행

함으로써 작동한다. 초기에는 완전한 노이즈 상태에서 시작했다가, 각 단계를 거침에 따라 노이즈를 점차 줄여나가면서 실제 데이터와 유사한 샘플을 생성한다.

[학습과정]

forward diffusion process & reverse diffusion process

forward diffusion process에서는 먼저 데이터에 노이즈를 추가하는 단계들을 정의하고, 이후 reverse diffusion process에서는 이 노이즈를 단계적으로 제거하는 방식으로 모델을 학습한다. 학습 과정은 노이즈가 추가된 데이터로부터 원본 데이터를 복구하는 방법을 모델이 학습하게 만든다. 이 과정에서 중요한 것은 각 단계에서 얼마나 많은 노이즈를 제거해야 하는지를 결정하는 것이며, 이는 모델이 데이터의 복잡한 분포를 효과적으로 학습할 수 있게 한다.

확산 모델은 생성된 샘플의 다양성과 품질 면에서 뛰어난 성능을 보이며, 특히 natural image와 같은 복잡한 데이터에 대해 매우 사실적인 결과를 생성할 수 있다. 이러한 모델은 높은 계산 비용과 긴 학습 시간을 필요로 하지만, 그 결과는 매우 인상적인 이미지나 다른 형태의 데이터를 생성할 수 있음을 보여준다.

Diffusion 모델의 핵심은 데이터 생성 과정을 노이즈의 점진적 제거로 보는 새로운 관점을 제공하는 점에 있으며, 기존의 생성 모델과 크게 차별화된다.

