10주차 Gen Adv 과제

대표적인 이미지 생성 모델들 - GAN, VAE, Flow-base, Diffusion 각각의 개념, 구조, 차이점 정리

GAN - Generative Adversarial Network (생성적 적대 신경망)

> 덥러닝 모델 중 이미지 생성에 많이 쓰이는 모델
>
> 새로운 데이터를 생성하기 위해 NN으로 이루어진 생성자(Generator)와 판별자(Discriminator)가 서로 겨루며 훈련한다는 것을 의미한다.

	생성자(G)	판별자(D)
입력	랜덤한 숫자로 구성된 벡터 z	1. 훈련 데이터셋에 있는 실제 샘플 x 2. 생성자가 만든 가짜 샘플 z
출력	최대한 진짜 같이 보이는 가짜 샘플 G (z)	입력 샘플이 진짜일 예측 확률
목표	훈련 데이터셋에 있는 샘플 x와 구별이 불가능한 가짜 샘플 G(z) 생성하기	생성자가 만든 가짜 샘플 G(z)와 훈련 데이터셋의 진짜 샘플 x 구별하기

과정

- 판별자 D 먼저 학습
  - 진짜 이미지 x가 들어가면 True(1)로 구분한다.
  - 가짜 이미지(생성자가 random code를 받아 만들어낸 G(z))를 입력으로 받으면 False(0)으로 구분해야 한다.
- output layer에는 시그모이드 함수를 활성화 함수로 사용 (0.5를 기준으로 real or fake 이진 분류)
- 생성자 G는 random code z를 입력으로 받아 진짜 이미지 x와 유사한 이미지 생성 (G가 잘 학습된다면, D(G(z))가 1에 가까운 값이 나온다. - 가짜를 진짜라고 구분하게 된다.)

이진크로스엔트로피 BCE = $-\frac{1}{n}\sum_{i=1}^{n}(y_i \log(p_i) + (1-y_i)\log(1-p_i))$

LOSS

- Binary Cross Entropy (BCE) 사용
- real → y =1, fake → y = 0으로 이진 분류
- $x - P_{data}(x)$: 실제 데이터 분포에서 온 sample x
- $z - P_x(x)$: 가우시안 분포에서 온 Sample latent code z

GAN, 묵시적(implicit)으로 확률 분포를 모델링 하기 때문에 모델 구성에 제한이 없고 생성된 결과물의 품질이 뛰어난 편이다. 하지만 생성자와 판별자가 함께 학습이 되어야 한다는 점 그리고 모드 붕괴 등 학습에 어려움이 있다.

- 모드 붕괴 (mode - collapse)

생성자(G)와 판별자(D) 중 하나가 너무 학습이 지나치게 잘 돼서 다른 하나의 학습이 진행되지 않는 것을 의미한다. ex) 생성자(G)가 하나의 정말 진짜 같은 가짜 데이터를 생성할 경우, 판별자(D)는 이를 항상 구별할 수 없을 것이고, 판별자(D)는 더 이상 학습을 진행할 수 없다.

VAE - Variational AutoEncoder

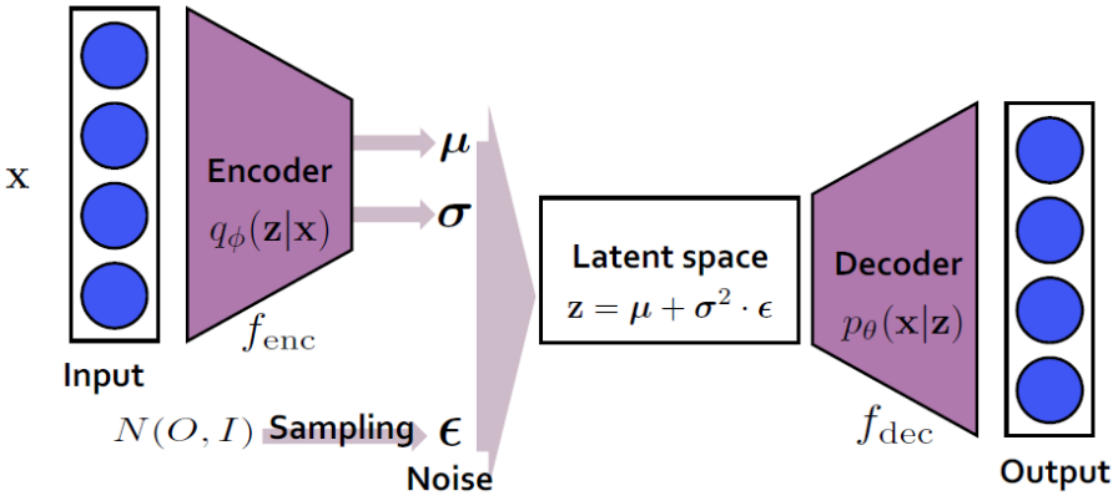
> 오토 인코더 개념을 활용한, Generative Model 중 하나이다. 인코더를 통해 입력 값을 특정 확률 분포 상의 한 점으로 만들고, 디코더를 통해 해당 점으로부터 입력 값을 생성한다. (즉, encoder와 decoder를 활용해 latent space를 도출하고, 도출된 latent space로부터 우리가 원하는 output을 decoding함으로써 data generation 진행한다.)

구조 및 학습

- 많은 생성 모델들이 그렇듯, VAE 또한 현실과 비슷한 데이터들을 만들어내는 것을 목적으로 한다. 이를 달리 말하면, input data의 true distribution인 p(x)를

approximate하는 것을 목적으로 한다.

- 이를 위해 VAE는 아래 그림처럼 인코더와 디코더 그리고 latent space로 구성된 구조를 가진다. 그리고 VAE는 학습 단계에서 이 인코더와 디코더의 parameter를 학습하는 것을 목적으로 한다.



Encoder

- input을 latent space(잠재 공간)으로 변환하는 역할을 한다.
- input x가 주어졌을 때 latent vector z의 분포, 즉 위 그림에서 나타나듯이 $q(z|x)$를 approximate하는 것을 목적으로 한다.

Decoder

- Encoder와 반대로 latent space(잠재 공간)를 input으로 변환하는 역할을 한다.
- latent vector z가 주어졌을 때, x의 분포, 즉 위 그림에서 나타나듯이 $p(z|x)$를 approximate하는 것을 목적으로 한다.
- 어떤 z라는 vector가 주어짐에 따라 다시 데이터 x를 생성하는 역할을 하기 때문에, decoder가 생성 모델의 역할을 한다.

Latent space

- 숨겨져 있는 vector들을 의미한다. 이 latent space가 주어져야 디코더는 이를 활용해 data를 생성할 수 있다.
- VAE가 오토인코더처럼 input과 output을 똑같이 만드는 것을 목적으로 한다고 가정했을 때, 이 목적으로 만들어진 latent space는 항상 input과 같은 모양의 데이터

를 만들어낼 수밖에 없다.

- 이를 방지하기 위해 noise를 샘플링해서 이로부터 latent space를 만든다.
- 예를 들어 어떤 표준 정규분포 (평균 0, 표준편차 1)로부터 하나의 noise epsilon을 샘플링해서 얻고, 이에 encoder로부터 얻은 분산을 곱하고 평균을 더해서 최종적으로 latent vector z를 얻는다. 이를 reparametrization trick이라고 불린다.

ELBO(Evidence Lower Bound) - 목적함수

- $p_{theta}(x)$를 maximize하는$theta$를 찾는 것

$$\log p_\theta(\mathbf{x})$$

log-likelihood of VAE

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})} \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z} - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))
\end{aligned}$$

VAE의 maximum likelihood 증명

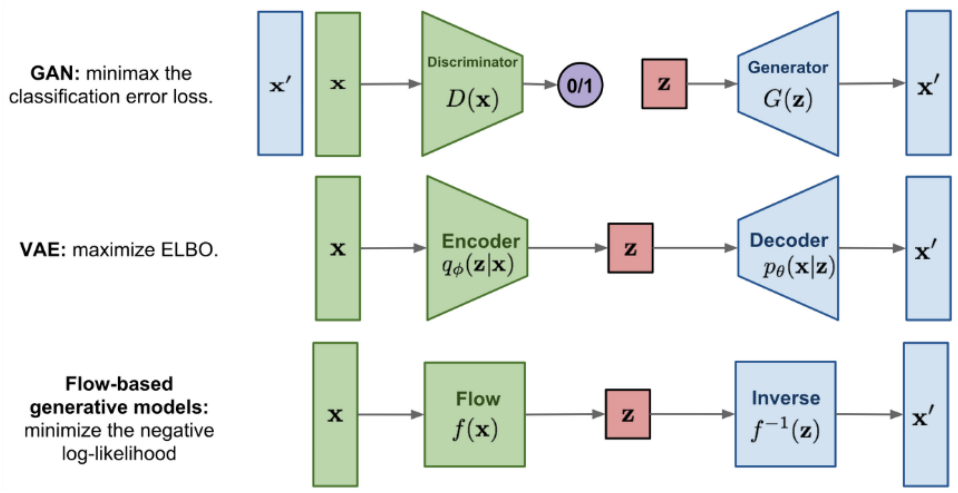
참고할 것....

VAE는 명시적인 확률 분포를 모델링 한다는 장점이 있으나, 가능도(likelihood)가 아닌 정확히는 ELBO를 통한 학습이라는 점에서 아쉬움이 존재한다. 또한 다른 모델들에 비해 비교적 생성된 결과물의 퀄리티가 떨어진다는 단점이 존재한다.

Flow-base

> 앞서 다룬 GAN과 VAE는 Encoder-Decoder 구조, Generator, Discriminator 구조를 가지고 있어서 묵시적(implicit)으로 실제 데이터의 분포를 학습한다. 하지만 Flow based Model은 앞에서 다

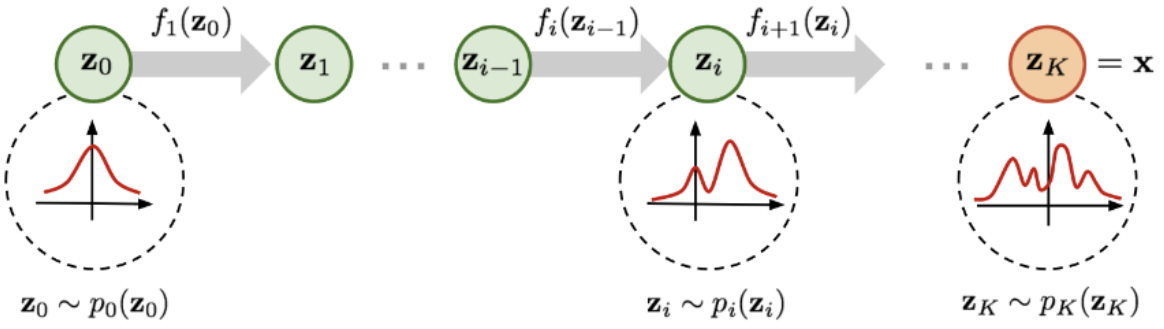
> 룬 것과는 다르다. 잠재 벡터 z의 확률 분포에 대한 일련의 역변환 (a sequence of invertible transformations)을 통해 데이터 x의 분포를 명시적으로 학습하여 이를 간단하게 negative log-likelihood로 해결한다.



세 모델 차이점

Normalizing Flow

- NF는 실제 데이터의 복잡한 확률 분포를 예측하는 데 있어서 효과적인 방식 중 하나
- 어떠한 확률 분포에 역변환 함수를 적용해서 새로운 확률 분포로 변환할 수 있다.
- NF는 단순한 확률 분포에서부터 일련의 역변환 함수를 적용하여 점차 복잡한 확률 분포로 변환해 나가는 과정이다.
- 이런 일련의 변환과 변수 변환 이론을 통해 우리는 단순한 분포로부터 새로운 변수들을 반복해서 대체하고 결과적으로 목표하는 최종 변수의 확률 분포를 얻을 수 있게 된다.



$$\mathbf{z}_{i-1} \sim p_{i-1}(\mathbf{z}_{i-1})$$

$$\mathbf{z}_i = f_i(\mathbf{z}_{i-1}), \text{thus } \mathbf{z}_{i-1} = f_i^{-1}(\mathbf{z}_i)$$

$$p_i(\mathbf{z}_i) = p_{i-1}(f_i^{-1}(\mathbf{z}_i)) \left| \det \frac{df_i^{-1}}{d\mathbf{z}_i} \right|$$

연속된 역변환을 나타내기 위해 i번째의 변수와 확률분포 그리고 역변환을 표시했다.

$$\begin{aligned} p_i(\mathbf{z}_i) &= p_{i-1}(f_i^{-1}(\mathbf{z}_i)) \left| \det \frac{df_i^{-1}}{d\mathbf{z}_i} \right| \\ &= p_{i-1}(\mathbf{z}_{i-1}) \left| \det \left(\frac{df_i}{d\mathbf{z}_{i-1}} \right)^{-1} \right| \\ &= p_{i-1}(\mathbf{z}_{i-1}) \left| \det \frac{df_i}{d\mathbf{z}_{i-1}} \right|^{-1} \\ \log p_i(\mathbf{z}_i) &= \log p_{i-1}(\mathbf{z}_{i-1}) - \log \left| \det \frac{df_i}{d\mathbf{z}_{i-1}} \right| \end{aligned}$$

역함수 이론 (Inverse function theorem)과 가역함수의 자코비안 (Jacobians of invertible function) 특성에 따라 전개

- 역함수 이론 (만약 $y = f(x)$ 와 $x = f^{-1}(y)$가 있다면,)

$$\frac{df^{-1}(y)}{dy} = \frac{dx}{dy} = (\frac{dy}{dx})^{-1} = (\frac{df(x)}{dx})^{-1}$$

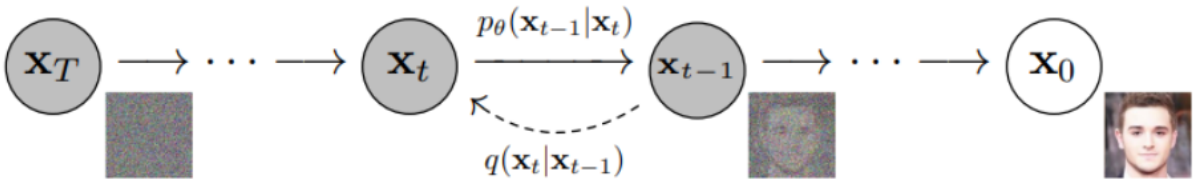
즉 역함수의 미분과 함수의 미분은 inverse 관계

- 가역함수의 자코비안 - 가역행렬인 경우 행렬식 특성 가진다.

$$\det(M^{-1}) = (\det(M))^{-1}$$
$$\det(M)\det(M^{-1}) = \det(M \cdot M^{-1}) = \det(I) = 1$$

Diffusion

> data로부터 noise를 조금씩 더해가면서 data를 완전히 noise로 만드는 forward process와 이와 반대로 noise로부터 조금씩 복원해 가면서 data를 만들어내는 reverse process를 활용한다.



- 위 그림에서 실제 데이터 X_0, 최종 noise X_T, 그리고 그 사이에 X_t는 데이터에 noise가 더해진 상태의 latent variable을 의미한다.
- 오른쪽에서 왼쪽 방향으로 noise를 점점 더해가는 forward process q를 진행
- 그리고, 이 forward process를 반대로 추정하는 reverse process p를 학습함으로써 noise(X_T)로부터 data(X_0)를 복원하는 과정을 학습한다.
- 이 reverse process를 활용해 random noise로부터 우리가 원하는 image, text, graph 등을 generate할 수 있는 모델을 만들어내는 것이다.

Reverse process

- Reverse process p는 noise(X_T)로부터 data(X_0)를 복원하는 과정이다.
- 이는 최종적으로 random noise로부터 data를 생성하는 generative model로 사용되기 때문에 diffusion model을 사용하기 위해서는 모델링하는 것이 필수적이지만, 이를 실제로 알아내는 것은 어렵다.
- 따라서 우리는 P_{theta}를 활용해 이를 approximate한다. 이 때 이 approximation은 가우시안 transition을 활용해 마르코프 체인의 형태를 가진다.

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

위 식에서 평균과 표준편차 (시그마세타)는 학습되어야 하는 파라미터들이다. 그리고 위 식의 시작 지점인 noise의 분포는 다음과 같이 가장 간단한 형태의 표준정규분포로 정의한다.(Reverse process 수식 표현)

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

noise의 분포

Forward process

- Forward process q는 data(X_0)로부터 noise를 더해가면서 최종 noise(X_T)형태로 가는 과정
- 이 과정의 분포는 reverse process의 학습을 forward process의 정보를 활용해서 하기 때문이다.
- Forward process는 reverse process와 비슷하지만, 조금 다르게 data에 Gaussian noise를 조금씩 더하는 마르코프 연쇄를 가진다.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Forward process 수식 표현

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

x_0가 주어졌을 때 x_t의 분포

$$\alpha_t = 1 - \beta_t$$

α

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

ᾱ

Training

앞서 Reverse process와 forward process에 대한 과정을 살펴보았고 마지막으로 P_{theta}의 파라미터 추정을 위해 diffusion model 어떻게 학습되는지 살펴보고자 한다.

- 실제 data의 분포인 P(x_0)를 찾아내는 것을 목적으로 하기 때문에 결국 이의 likelihood를 최대화하는 것이 우리가 원하는 목적이다.

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

diffusion 모델의 training loss

Diffusion model에서도 DDPM, D3PM 등 다양한 변형 형태들이 많이 나와 있다.

▼ 참고자료

▼ GAN

https://process-mining.tistory.com/169

https://futures-studies.tistory.com/entry/GAI-생성AI-기초4-생성모델-소개

투빅스 20기 Gen basic 강의 자료

https://blog.est.ai/2022/02/생성-모델의-새로운-흐름-확산-모델diffusion-model에-관하여/

▼ VAE

https://velog.io/@idj7183/AE-VAE

https://process-mining.tistory.com/161

https://seongukzz.tistory.com/3

▼ Flow-base

https://judy-son.tistory.com/12

https://devkihyun.github.io/study/Flow-based-Generative-Models-1-Normalizing-Flow/

▼ Diffusion

https://process-mining.tistory.com/182

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

https://devkihyun.github.io/study/Flow-based-Generative-Models-1-Normalizing-Flow/

→ 무조건 참고하면 좋은 자료들 많음.