Normalizing Flow

- 앞 장의 식

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

- 앞 장에서 계산 편의를 위해 취한 log 덕분에 역변환 과정을 확률분포의 합으로 나타낼 수 있다.

드디어 우리는 $\mathbf{z}_0$ 의 확률분포에서부터 시작해 K 번의 역변환을 통해 $\mathbf{x}$ 의 확률분포를 구하는 수식을 앞에서 배웠던 것들을 활용하여 정리할 수 있게 되었습니다.

$$\mathbf{x} = \mathbf{z}_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{z}_0)$$

$$\begin{aligned}
\log p(\mathbf{x}) = \log \pi_K(\mathbf{z}_K) &= \log \pi_{K-1}(\mathbf{z}_{K-1}) - \log\left|\det \frac{df_K}{d\mathbf{z}_{K-1}}\right| \\
&= \log \pi_{K-2}(\mathbf{z}_{K-2}) - \log\left|\det \frac{df_{K-1}}{d\mathbf{z}_{K-2}}\right| - \log\left|\det \frac{df_K}{d\mathbf{z}_{K-1}}\right| \\
&= \dots \\
&= \log \pi_0(\mathbf{z}_0) - \sum_{i=1}^{K} \log\left|\det \frac{df_i}{d\mathbf{z}_{i-1}}\right|
\end{aligned}$$

Normalizing Flow

딥러닝 모델이 학습할 수 있게 해주려면 우리는 항상 적절한 학습 criterion을 제공해야됩니다. 분류 학습에는 주로 Cross-Entropy 가 사용되는 것처럼요. 다행인 것은 우리가 NF 를 이용해 비교적 쉽게 $\log p(\mathbf{x})$ 를 정의할 수 있었습니다. 그럼 남은 것은 우리가 가지고 있는 학습 데이터셋 $\mathcal{D}$ 에 대해 간단하게 Negative Log-Likelihood (NLL) 을 만들어서 생성 모델에게 학습 criterion 으로 넘겨주면 끝납니다.

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

이것이 바로 Normalizing Flow 방식입니다. 간단하죠? 그 대신, 해당 방정식의 계산이 가능하게 하려면 두 가지 조건을 충족해야됩니다.

- 함수 f 는 가역적이여야 한다.
- f 에 대한 자코비안 행렬식은 계산하기 쉬워야 된다.

이런 조건들이 만족하지 않으면 아무리 NF 가 강력하더라도 딥러닝 모델이 학습하기가 쉽지 않습니다. 그런데 문제는, 위의 조건을 만족하도록 하는 것부터가 쉽지 않다는 것이죠. 그래서 다음 시간

- 부족한 점
  - 고=>저 하는 방정식f은 서술x
  - 예시 / 한계점 서술x
- 앞으로…
  - 예시 / 한계점 조사
  - ddpm 조사
  - 세인 군 python