Day 6,7 : Conditional Diffusion Model

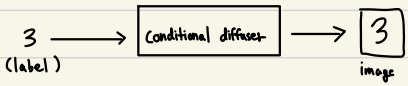
- Since now, we have modeled generative model that models population of x, $p(x)$. What if we model $P(x|y)$ where y is a condition.

⇒ 조건 y(텍스트, 레이블, 이미지 등)를 바꿔며 생성하고자 하는 x를 제어한다!

✳ Set condition y as label. (Conditional Diffusion Model)

- Train with MNIST images and give condition (label) when generating!

fig)



• Conditional Diffusion Model where the neural network predicts $u_\theta(x_t, t)$ directly.

1. Modeling $P_\theta(x)$

$$P_\theta(x_0) = \int P_\theta(x_0, x_1, x_2 \cdots x_T)\, dx_{1:T} \quad \text{(marginalization)}$$

($x_1, x_2 \cdots x_T$: latent variables)

이것도 학습 수식으로 $P_\theta(x_T)$이나, $P(x_T) = N(x_T; 0, I)$로 가정. (prior를 가정한다! 학습 x_T가 어떤 분포를 따를지 가정)

$$= \int P_\theta(x_0|x_{1:T}) P_\theta(x_1|x_{2:T}) \cdots P_\theta(x_{T-1}|x_T) P(x_T)\, dx_{1:T} \quad \text{(chain rule)}$$

$$= \int P_\theta(x_0|x_1) P_\theta(x_1|x_2) \cdots P_\theta(x_{T-1}|x_T) P(x_T)\, dx_{1:T} \quad \text{(markov property)} \quad \text{where } P(x_T) = N(x_T; 0, I)$$

$$P_\theta(x_{t-1}|x_t) = N(x_{t-1}; u_\theta(x_t, t), \delta_f^2(t) I)$$

2. Modeling $P_\theta(x_0|y)$

$$P_\theta(x_0|y) = \int P_\theta(x_0, x_1, \cdots x_T|y)\, dx_{1:T} \quad \text{(marginalize)}$$

(y: added condition; $x_1, \cdots x_T$: latent variables)

$$= \int P_\theta(x_0|x_{1:T}, y) P_\theta(x_1|x_{2:T}, y) \cdots P_\theta(x_{T-1}|x_T, y) P_\theta(x_T|y)\, dx_{1:T} \quad \text{(chain rule)}$$

↓ Markov (t-1 step은 t와, 그 y에 대해서만 결정됨): $P_\theta(x_{t-1}|x_{t:T}, y) \approx P_\theta(x_{t-1}|x_t, y)$

$$= \int P_\theta(x_0|x_1, y) P_\theta(x_1|x_2, y) \cdots P_\theta(x_{T-1}|x_T, y) P_\theta(x_T|y)\, dx_{1:T} \quad \text{(markov property)}$$

여기도 수식적으로 이렇게 전개.. But, diffusion model assumes $P_\theta(x_T) \to P(x_T) = N(x_T; 0, I)$

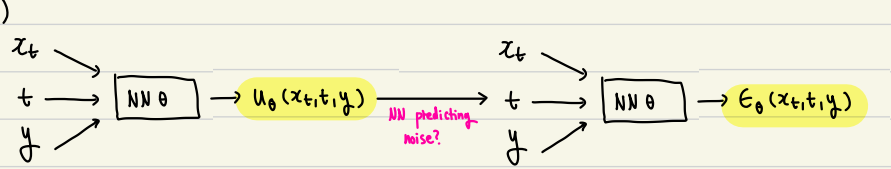
And also in conditional diffusion model $P_\theta(x_T|y) \to P(x_T|y) = P(x_T) = N(x_T; 0, I)$

↓

모든 조건에 대해 동일한 Gaussian 노이즈로 가정. (prior를 가정한다)

$$P_\theta(x_{t-1}|x_t, y) = N(x_{t-1}; u_\theta(x_t, t, y), \delta_f^2(t) I) \quad \text{(added condition } y \text{ for Neural Network's input)}$$

fig)



=> 훈련할 때 이렇게 condition은 작은 행렬로 노이즈 예측 시 내분점을 직접 예측하는 방식으로 신경망이 훈련됨.

=> forward diffusion process에는 label 적용 X

✳ Score function

- At previous step, just provided condition y...

→ issue : The condition 'y' can be ignored.

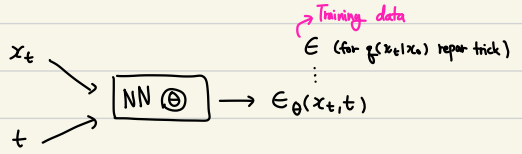
Is there any way we can streghthen the condition?

↓

"Guidance Method"

In diffusion model, the neural network predicted noise at each step ⇒ $\epsilon_\theta(x_t, t)$

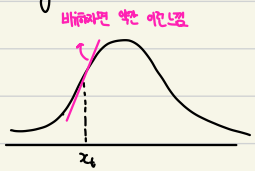
fig)



where $\epsilon \approx -\sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t} \log P(x_t)$ ... ① (학습하려는 ϵ과 score function $\nabla_{x_t}\log P(x_t)$과 관련 있음.)

Ⓐ

Ⓐ : A gradient of x_t related to $\log P(x_t)$ ⇒ Score function / score



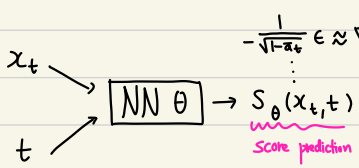
① 에서 ϵ는 $\nabla_{x_t} \log P(x_t)$ (score)로 근사적으로 나타낼 수 있음.

$$\epsilon \approx -\sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t} \log P(x_t)$$

ϵ과 score는 어떤 상수배만큼 차이 난다.

⇒ 그럼 ϵ만 알면, score는 $-\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}} = \nabla_{x_t}\log P(x_t)$로 학습 데이터로 쓰자! (신경망이 score를 예측하도록 한다)

fig



⇒ "Diffusion model where neural network predicts score $S_\theta(x_t, t) \longrightarrow \nabla_{x_t} \log P(x_t) \approx -\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}}$"

✱ Why ① in page 3?

$$\epsilon \approx -\sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log P(x_t) \cdots ①$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I)$$
$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon \quad (\epsilon \sim N(0, I)) \cdots ②$$

- Tweedie's Formula

For $x \sim N(x; u, \Sigma)$, ($N(x; u, \Sigma)$의 값)

$E[u|x] = x + \Sigma \nabla_x \log P(x)$ ($N(x; u, \Sigma)$에서 x를 관측했을 때, u에 대한 기댓값)

- Adopt Tweedie's formula to Diffusion

$x_t \sim N(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I)$ ②

$E[\sqrt{\bar{\alpha}_t} x_0 \mid x_t] = x_t + (1-\bar{\alpha}_t) I \nabla_{x_t} \log P(x_t)$ ⑥

Since ②, (x_t가 관측되었음 → constant)

$E[\sqrt{\bar{\alpha}_t} x_0 \mid x_t] = x_t + (1-\bar{\alpha}_t) I \nabla_{x_t} \log P(x_t)$

$\Leftrightarrow E[x_t - \sqrt{1-\bar{\alpha}_t}\epsilon \mid x_t] = x_t + (1-\bar{\alpha}_t) I \nabla_{x_t} \log P(x_t)$

$\Leftrightarrow E[x_t|x_t] - E[\sqrt{1-\bar{\alpha}_t}\epsilon \mid x_t] = x_t + (1-\bar{\alpha}_t) I \nabla_{x_t} \log P(x_t)$

$\Leftrightarrow x_t - \sqrt{1-\bar{\alpha}_t}\, E[\epsilon|x_t] = x_t + (1-\bar{\alpha}_t) I \nabla_{x_t} \log P(x_t)$

$\Leftrightarrow E[\epsilon|x_t] = -\sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log P(x_t)$

$\Leftrightarrow$ Monte carlo for $E[\epsilon|x_t]$ (one sample x_t) : $\epsilon \approx -\sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log P(x_t)$

✳ Neural Network predicting score function 부가설명

- tweedie 공식으로 구해 알고있는 ϵ을 $-\sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t}\log P(x_t)$ 에 근사하여 $-\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}}$ (score function)을 학습데이터로 사용 → 신경망이 score를 예측할 수 있게됨.
- 이 모든 과정은 $\nabla_{x_t}\log P(x_t)$를 해석적으로 구할 수 없기에 어떤식으로 학습시킬 score를 근사할 지에 대한 방법임. → tweedie를 통해 score를 근사할 방법을 찾음.

의미) x_t는 한 데이터 x_0의 시점 t에 해당하여 noise가 섞인 데이터 (x_0로부터 x_t를 sampling함) ⓐ
tweedie 공식으로 추출된 x_t와 x_t에 대한 score function을 기반으로 (ⓑ $\bar{\alpha}$ hyperparam)
스케일된 원본 $\sqrt{\bar{\alpha}_t}x_0$에 대한 기댓값을 구할 수 있다 (사후 평균)

↓

트위디 공식의 식을 변형하여 $\sqrt{\bar{\alpha}_t}x_0$를 노이즈 ϵ에 대한 식으로 바꿀 수 있고, ϵ에 ($E[\sqrt{\bar{\alpha}_t}x_0|x_t] \to E[\epsilon|x_t]$)
대한 기댓값으로 바꿀수 있다 : $E[\epsilon|x_t]$

↓

$$E[\epsilon|x_t] = \underbrace{-\sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t}\log P(x_t)}_{\text{Ⓐ}} \text{ 라는 식이 유도된다}$$

monte carlo를 통해 $E[\epsilon|x_t]$를 근사하는 것은 Ⓐ를 근사하는 것과 같음.

=) 몬테카를로 방식을 통해 주어진 x_t에 대한 ϵ의 전체 분포에서 ϵ을 여러개 또는 1개 구하고 평균내어 Ⓐ라는 tweedie formula로부터 이론적으로 도출된 식을 근사.
$P(\epsilon|x_t)$에서 ϵ을 근사적 sampling하면, $\epsilon \approx -\sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t}\log P(x_t)$가 나온다.
이를 통해 $-\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}}$를 모델이 예측할 수 있게하여 결과적으로 score function을 예측할 수 있게됨. (이를 통해 결과적 forward diffusion process 에 샘플링된 ϵ을 score function으로 변환함!)

✳ 조건부 기댓값

조건부 기댓값(conditional expectation)의 정의는 다음과 같습니다.

- 조건부 기댓값은 어떤 확률 변수 X와 Y에 대해, $X = x$라는 정보가 주어졌을 때 Y의 평균(기댓값)을 의미합니다. 즉, "$X = x$라는 상황 아래에서 Y가 가질 수 있는 값들의 평균"입니다 [1] [2] [3] [4].
- 조건부 확률 분포를 사용하기 때문에, 조건이 주어진 상태에서의 기댓값이라고도 하고, 조건에 따라 값이 바뀌는 함수입니다.

이산형(Discrete) 확률변수의 경우

$$E[Y|X = x] = \sum_y y \cdot P(Y = y \mid X = x)$$

연속형(Continuous) 확률변수의 경우

$$E[Y|X = x] = \int y \cdot f_{Y|X}(y|x)\,dy$$

여기서 $f_{Y|X}(y|x)$는 조건부 확률밀도함수입니다 [1] [3].

즉, 조건부 기댓값은 어떤 사건이나 랜덤변수의 값이 주어진 상황에서, 다른 변수의 평균적인 값을 의미합니다.
이는 데이터 예측, 통계 모델링, 머신러닝 등 거의 모든 확률적 모델에서 중요한 개념입니다.

* Guidance (with aspect of neural net predicting score in diffusion)

- More details of Classifier Guidance & Classifier Free guidance will be dealt in Github README.md files!

① Classifier Guidance : Noisy한 이미지로 부터도 분류를 이용해 GUIDE 해줄 수 있다!

- Score function predicting diffusion model : $\nabla_{x_t} \log P(x_t)$
- Conditional Score function predicting diffusion model : $\nabla_{x_t} \log P(x_t|y)$

· Induction of conditional score function

$$P(x_t|y) = \frac{P(x_t)P(y|x_t)}{P(y)}$$

$$\nabla_{x_t} \log P(x_t|y) = \nabla_{x_t} \log \frac{P(x_t)P(y|x_t)}{P(y)}$$

$$= \nabla_{x_t} \log P(x_t) + \nabla_{x_t} \log P(y|x_t) - \nabla_{x_t} \log P(y) \quad (\nabla_{x_t} \log P(y) \to 0)$$

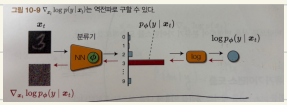
$$= \underbrace{\nabla_{x_t} \log P(x_t)}_{\text{score prediction neural network}} + \underbrace{\nabla_{x_t} \log P(y|x_t)}_{\text{classifier gradient}}$$

① ② 로 각각 구해줄 수 있고 이를 통해 조건절의 추정 가능!

① : Diffusion neural network를 통해 score 구하기

② : Classifier로 y에 대한 log 확률 계산 후 입력 x_t에 대한 gradient 구하기

fig of ②



γ : 이를 통해 조건절 추정시 분류 기준도를 조정함. (생성시 class y별로 안내하는 정도)

$$\Rightarrow \nabla_{x_t} \log P(x_t|y) = \nabla_{x_t} \log P(x_t) + \gamma \nabla_{x_t} \log P(y|x_t)$$

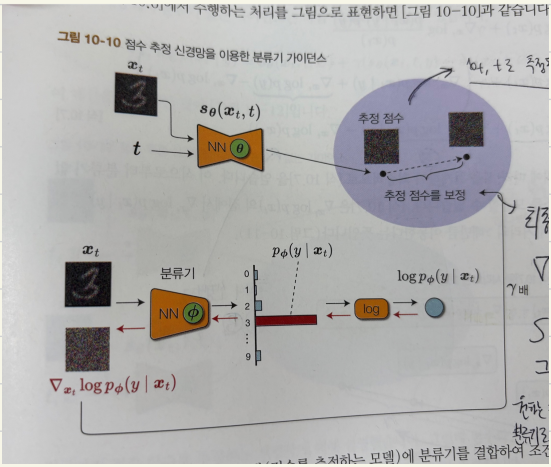
✱ Summary of classifier guidance of diffusion model

- 일반적인 점수 추정 확산 모델에 분류기를 결합시켜서 조건부 확산 모델을 만들고 데이터를 생성!
  ① ②

- 잡음 있는 score 예측을 학습된 classifier에서 생성한 gradient를 통해 보정

- 이는 생성 시에 확률적 경로로 y label 방향으로 보정을 지속 여러 경로를 나타냄.

- 생성 방식 fig



② Classifier free Guidance

- ①은 분류기를 따로 훈련해야 해서 효율성 ↓
- Give guidance without classifier

$$\nabla_{x_t} \log P(x_t|y) = \nabla_{x_t} \log \frac{P(x_t)\,P(y|x_t)}{P(y)}$$

$$= \nabla_{x_t} \log P(x_t) + \nabla_{x_t} \log P(y|x_t) - \nabla_{x_t} \log P(y) \quad (\nabla_{x_t} \log P(y) \to 0)$$

$$= \nabla_{x_t} \log P(x_t) + \nabla_{x_t} \log P(y|x_t)$$

↳ C.G induction

bayes 정리 한번 더 적용!

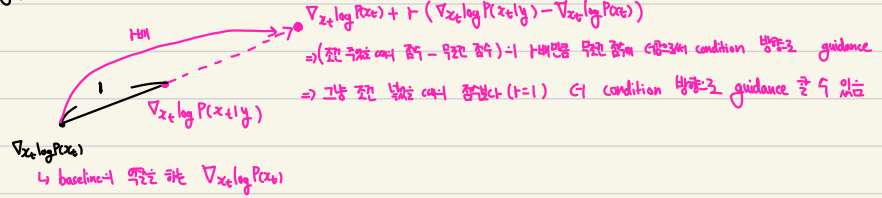
$$= \nabla_{x_t} \log P(x_t) + \nabla_{x_t} \log \frac{P(y)\,P(x_t|y)}{P(x_t)}$$

$$= \nabla_{x_t} \log P(x_t) + \nabla_{x_t} \left(\log P(y) + \log P(x_t|y) - \log P(x_t) \right) \quad (\nabla_{x_t} \log P(y) \to 0)$$

$$= \nabla_{x_t} \log P(x_t) + \nabla_{x_t} \left(\log P(x_t|y) - \log P(x_t) \right)$$

$$\therefore \nabla_{x_t} \log P(x_t|y) = \underset{①}{\nabla_{x_t} \log P(x_t)} + \nabla_{x_t} \left(\underset{②}{\log P(x_t|y)} - \log P(x_t) \right)$$

fig.)



=> "결과적" Diffusion의 neural network로 조건점수와 무조건 점수를 모두 추정하도록 훈련시키고, generation시에 무조건점수 + ∇(조건점수 − 무조건점수)를 통해 condition guiding을 큼 (더욱 강조해 condition 방향으로 전진)

• How to predict $\nabla_{x_t} \log P(x_t)$, $\nabla_{x_t} \log P(x_t|y)$

Predict with Single Neural Network! (S_θ)

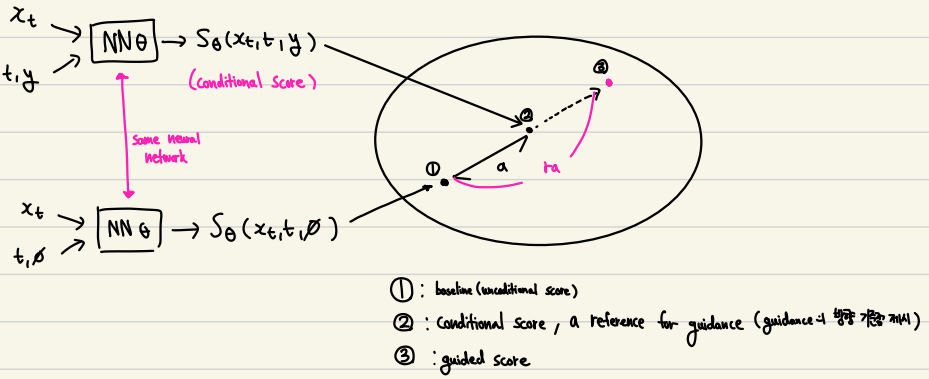
① $\nabla_{x_t} \log P(x_t) = S_\theta(x_t, t, \varnothing)$ ← zero embedded

② $\nabla_{x_t} \log P(x_t|y) = S_\theta(x_t, t, y)$ ← condition embedded

⇒ 학습 시에 ①, ②를 다 예측할 수 있도록 학습시켜야 함

$\therefore \nabla_{x_t} \log P(x_t|y) = S_\theta(x_t, t, \varnothing) + \gamma (S_\theta(x_t, t, y) - S_\theta(x_t, t, \varnothing))$

fig) Generation with Classifier Free Guidance



①: baseline (unconditional score)

②: conditional score, a reference for guidance (guidance의 방향 가늠해 줌)

③: guided score

• What if Neural Network predicts noise ϵ?

Since $\epsilon \approx -\sqrt{1-\bar{\alpha}_t} \nabla_{x_t} \log P(x_t)$, Score와 상수배밖에 차이가 없음.

⇒ 똑같은 구조 하에, $\epsilon_\theta(x_t, t, \varnothing)$ / $\epsilon_\theta(x_t, t, y)$로 Unconditional / Conditional 점들을 예측하도록 학습함!

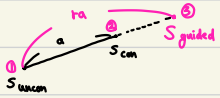
★ Summary) 훈련 시에 동일한 network가 unconditional / Conditional score를 모두 계산하도록 train, generation 시에 unconditional score와 전체 guidance를 추가한

baseline ↙ ↘ direction reference

conditional score를 모두 계산하여 $uncon + r(con - uncon)$ 식으로 baseline uncon score를 보정하여 전체 condition 방향으로 보정

· Appedix

$$S_{guided} = S_{uncon} + r(S_{con} - S_{uncon})$$



$r = 0$: unconditional generation, ①

$r = 1$: 전체 score만 반영된 conditional generation

$r > 1$: conditional signal을 강화, 더 "확실한" 전체 샘플 생성 ⇒ r이 너무 크면 artifact가 생길 가능 ↗ 인위적, 부자연스러움

"Please also refer to README.md"