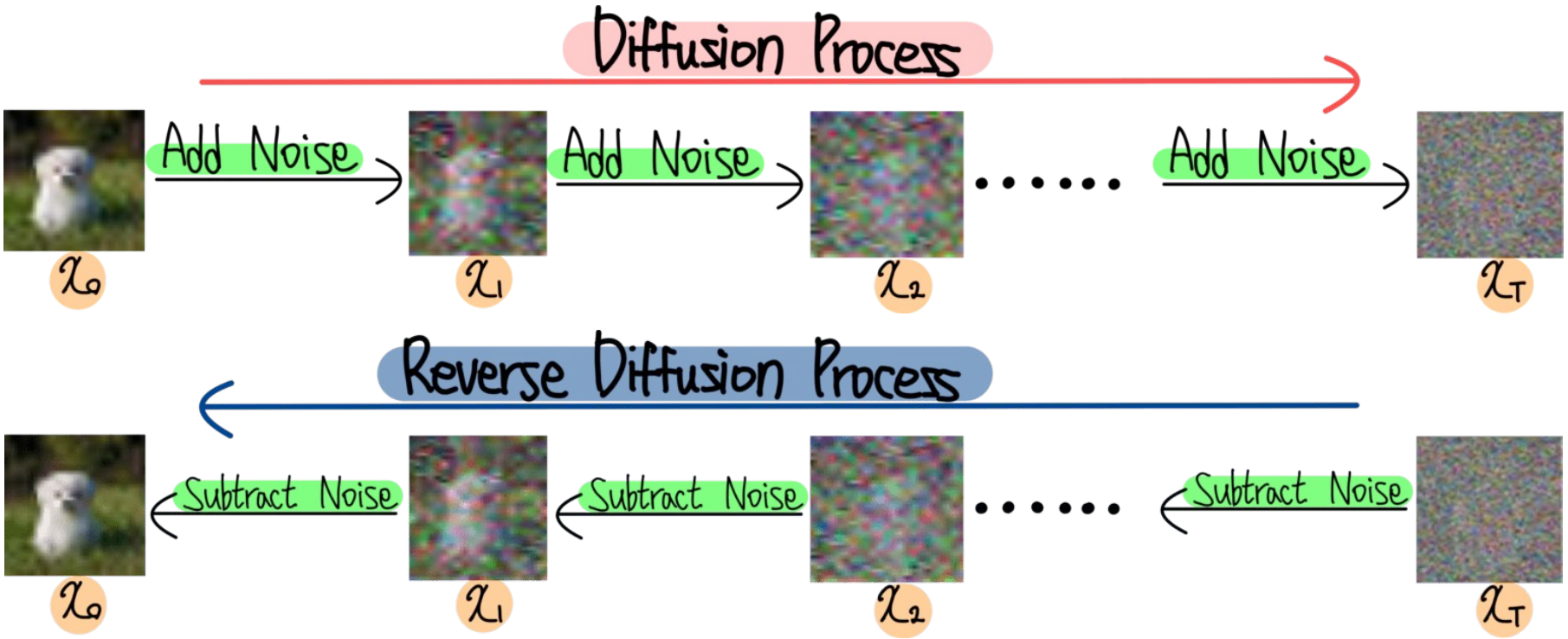
Structured Denoising Diffusion Models in Discrete State-Spaces

윤세환

목차

- diffusion 간단 복습
- 사전지식
  - Markov Transition Matrix
- introduction
- Diffusion models for discrete state spaces
- 손실 함수
- 실험 결과

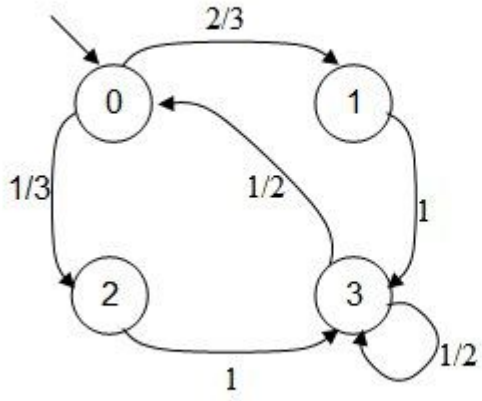
diffusion 간단 복습



Markov Transition Matrix

- 현재 상태에서 다른 상태로 전이할 확률을 행렬 형태로 표현한 것

State diagram



Matrix

$$\begin{pmatrix} 0 & 2/3 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 1/2 \end{pmatrix}$$

Probability distribution

Start.	(1,	0,	0,	0)
1.	(0,	2/3,	1/3,	0)
2.	(0,	0,	0,	1)
3.	(1/2,	0,	0,	1/2)
4.	(1/4,	1/3,	1/6,	1/4)

각 i번째 행은 i 상태에서 다른 상태로 전이할 확률을,
각 j번째 열은 이전 상태에서 j상태로 전이할 확률을 의미

1행 2열 -> 1번째 상태에서 2번째 상태로 전이할 확률

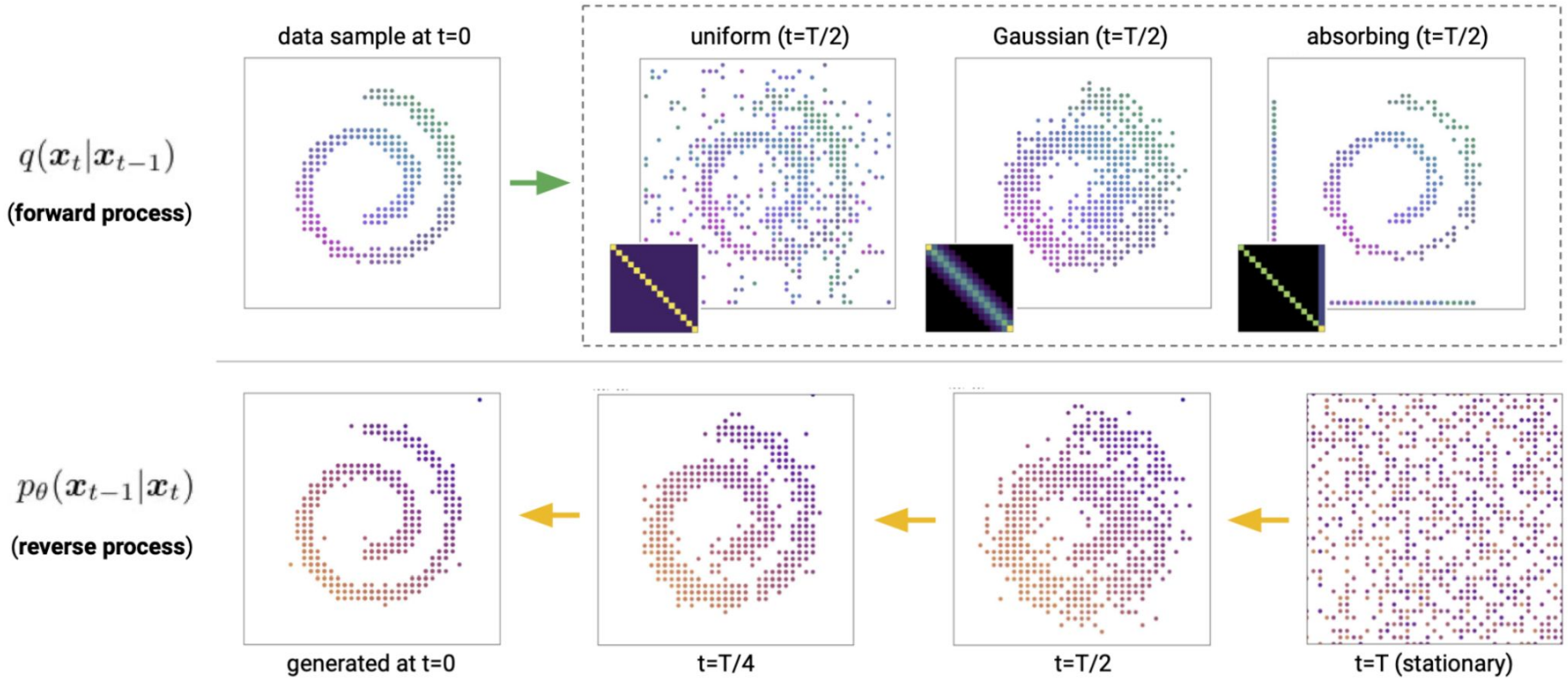
Introduction

- diffusion 모델은 이산형과 연속형 데이터 처리를 수행하기 위해 제안되었지만, 당시 연구들은 모두 연속형 데이터에 초점이 맞추어져 있었음
  (real-value 이미지 및 신호 데이터같은 waveform 데이터)
- 또한 이산형 데이터 분야에 있어서는, diffusion model이 텍스트 및 이미지 segmentation에 적용되었으나, 아직 대규모 텍스트나 이미지 생성 분야에서는 아직 입증되지 않음

Introduction

- 본 논문에서는 markov transition matrix를 사용한 보다 구조화된 diffusion process를 통해 diffusion 모델을 이산형 데이터 영역까지 발전시키고, 확장시키고자 함
- 기존 각 데이터에 random noise를 더해가는 과정 대신, transition matrix를 정의하고 이를 각 diffusion step마다 데이터를 나타내는 벡터에 곱해주는 방식

Introduction



Diffusion models for discrete state spaces

- K개의 카테고리가 있는 스칼라 이산 확률 변수의 경우, forward transition 확률은 markov transition matrix 의 행렬로 표현할 수 있다.

$$[\boldsymbol{Q}_t]_{ij} = q(x_t = j | x_{t-1} = i)$$

- 또한, K개의 카테고리를 가질 수 있는 x를 one-hot 행 백터로 표현할 경우, forward process 수식은 아래와 같이 표현할 수 있다.

$$q(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}) = \mathrm{Cat}(\boldsymbol{x}_t; \boldsymbol{p} = \boldsymbol{x}_{t-1}\boldsymbol{Q}_t)$$

Diffusion models for discrete state spaces

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \text{Cat}\left(\boldsymbol{x}_t; \boldsymbol{p} = \boldsymbol{x}_0\overline{\boldsymbol{Q}}_t\right), \quad \text{with} \quad \overline{\boldsymbol{Q}}_t = \boldsymbol{Q}_1\boldsymbol{Q}_2\ldots\boldsymbol{Q}_t$$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} = \text{Cat}\left(\boldsymbol{x}_{t-1}; \boldsymbol{p} = \frac{\boldsymbol{x}_t\boldsymbol{Q}_t^\top \odot \boldsymbol{x}_0\overline{\boldsymbol{Q}}_{t-1}}{\boldsymbol{x}_0\overline{\boldsymbol{Q}}_t\boldsymbol{x}_t^\top}\right)$$

Diffusion models for discrete state spaces

- Markov transition matrix 선택
  - 본 방식의 장점은, Qt를 선택함으로써 forward/reverse process를 제어할 수 있다는 점
  - Qt는 각 행의 합이 1이어야 하고, t가 점점 커짐에 따라 station하게 수렴해야 한다는 조건이 존재
- 본 논문에서 이미지 및 텍스트 데이터셋 실험을 위해 탐색할 transition 행렬들은 아래과 같음
  - Uniform
  - Absorbing state
  - Discretized gaussian
  - Token embedding distance

Diffusion models for discrete state spaces

uniform

$$[\boldsymbol{Q}_t]_{ij} = \begin{cases} 1 - \frac{K-1}{K}\beta_t & \text{if} \quad i = j \\ \frac{1}{K}\beta_t & \text{if} \quad i \neq j \end{cases},$$

Absorbing state

$$[\boldsymbol{Q}_t]_{ij} = \begin{cases} 1 & \text{if} \quad i = j = m \\ 1 - \beta_t & \text{if} \quad i = j \neq m \\ \beta_t & \text{if} \quad j = m, i \neq m \end{cases}$$

손실 함수

- Nichol, Dhariwal의 hybrid loss 연구에 영감을 받아
  reverse process에서의 x0-parameterization를 위한 보조 denosining 목적 함수를 도입하여 이를 negative variational lower bound Lvb와 결합

$$L_\lambda = L_{\mathrm{vb}} + \lambda\, \mathbb{E}_{q(\boldsymbol{x}_0)} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} [-\log \widetilde{p}_\theta(\boldsymbol{x}_0|\boldsymbol{x}_t)].$$

기존 텍스트 확률 모델과의 연관성

- Bert 모델의 경우, one-step diffusion model이다
  - 10%의 확률로 [MASK] 토큰으로 전환되고 5%의 확률로 랜덤하게 변화된다.
  - 따라서, absorbing state와 uniform markov transition matrix를 결합한 형태가 된다.
- Masked Language Models (MLMs)는 diffusion model이다.
  - [MASK] 토큰들로부터 단어를 생성하는 모델
  - x0문장을 샘플링한 후, 스케쥴 기법에 따라 k 개의 토큰을 마스킹한다. 이후 해당 마스킹된 토큰을 예측한다.
  - 이는 absorbing state를 사용한 d3pm 방식이다.

실험 결과

- text8에서 문자 레벨 텍스트 생성

Model	Model steps	NLL (bits/char) (↓)	Sample time (s) (↓)
Discrete Flow [49] (8×3 layers)	-	1.23	0.16
Argmax Coupling Flow [20]	-	1.80	0.40 ± 0.03
IAF / SCF [57]‡	-	1.88	0.04 ± 0.0004
Multinomial Diffusion (D3PM uniform) [20]	1000	≤ 1.72	26.6 ± 2.2
D3PM uniform [20] (ours)	1000	$\leq 1.61 \pm 0.02$	3.6 ± 0.4
D3PM NN (L_{vb}) (ours)	1000	$\leq 1.59 \pm 0.03$	3.1474 ± 0.0002
D3PM mask ($L_{\lambda=0.01}$) (ours)	1000	$\leq 1.45 \pm 0.02$	3.4 ± 0.3
D3PM uniform [20] (ours)	256	$\leq 1.68 \pm 0.01$	0.5801 ± 0.0001
D3PM NN (L_{vb}) (ours)	256	$\leq 1.64 \pm 0.02$	0.813 ± 0.002
D3PM absorbing ($L_{\lambda=0.01}$) (ours)	256	$\leq 1.47 \pm 0.03$	0.598 ± 0.002
Transformer decoder (ours)	256	1.23	0.3570 ± 0.0002
Transformer decoder [1]	256	1.18	-
Transformer XL [10]†	256	1.08	-
D3PM uniform [20] (ours)	20	$\leq 1.79 \pm 0.03$	0.0771 ± 0.0005
D3PM NN (L_{vb}) (ours)	20	$\leq 1.75 \pm 0.02$	0.1110 ± 0.0001
D3PM absorbing ($L_{\lambda=0.01}$) (ours)	20	$\leq 1.56 \pm 0.04$	0.0785 ± 0.0003

실험 결과

- CIFAR-10 (inception score, Frechet Inception Distance, Negative Log Likelihood)

Model	IS (↑)	FID (↓)	NLL (↓)
Sparse Transformer [9]			2.80
NCSN [45]	8.87 ± 0.12	25.32	
NCSNv2 [46]	8.40 ± 0.07	10.87	
StyleGAN2 + ADA [22]	9.74 ± 0.05	3.26	
Diffusion (original), L_{vb} [43]			≤ 5.40
DDPM L_{vb} [19]	7.67 ± 0.13	13.51	≤ 3.70
DDPM L_{simple} [19]	9.46 ± 0.11	3.17	≤ 3.75
Improved DDPM L_{vb} [30]		11.47	≤ 2.94
Improved DDPM L_{simple} [30]		2.90	≤ 3.37
DDPM++ cont [47]		2.92	2.99
NCSN++ cont. [47]	9.89	2.20	
D3PM uniform L_{vb}	5.99 ± 0.14	51.27 ± 2.15	$\leq 5.08 \pm 0.02$
D3PM absorbing L_{vb}	6.26 ± 0.10	41.28 ± 0.65	$\leq 4.83 \pm 0.02$
D3PM absorbing $L_{\lambda=0.001}$	6.78 ± 0.08	30.97 ± 0.64	$\leq 4.40 \pm 0.02$
D3PM Gauss L_{vb}	7.75 ± 0.13	15.30 ± 0.55	$\leq 3.966 \pm 0.005$
D3PM Gauss $L_{\lambda=0.001}$	8.54 ± 0.12	8.34 ± 0.10	$\leq 3.975 \pm 0.006$
D3PM Gauss + logistic $L_{\lambda=0.001}$	8.56 ± 0.10	7.34 ± 0.19	$\leq 3.435 \pm 0.007$