Denoising Diffusion Probabilistic Model

January 01, 2022

CONTACT

Ulsan National Institute of Science and Technology

Address 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Korea
Tel. +82 52 217 0114 **Web.** www.unist.ac.kr

Electronic Engineering

3rd Engineering Building Room 410-1
Web. www.github.io/nakkwan

CONTENTS

Introduction

Energy-based model이 최근 generative 분야에서 큰 성과를 거두었다.

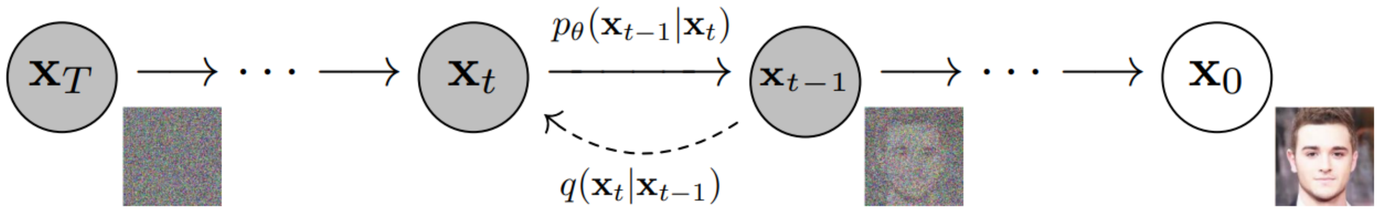
논문에서는 Diffusion probability model을 제시한다.
→ 유한한 시간 후의 data sample과 일치하는 sample을 뽑기 위해,
Variational Inference(VI)를 사용하여 훈련한 Markov chain model

→ VE : $p(x|x) = \frac{p(Z|X)p(z)}{p(x)}$ 에서 posterior를 찾기 힘들 때, 더 쉬운 확률 분포 $q(z)$로 근사하기 위한 방법

Sampling에 반대되는 방향으로 DM(Gaussian noise를 추가하는 것을 data가 파괴(noise)가 될 때까지 하는 Markov chain)의 reverse를 학습한다.

DM에서 gaussian noise가 작다면 Markov chain sampling을 conditional gaussian으로 설정할 수 있고, NN으로 parameterize가 가능하다. 또한 DM이 진행될수록, distinguish한 특성은 사라지고, isotropic gaussian distribution으로 변한다.

DM의 parameterize는 multi-level noise에서 denoising score match와 같다.



Background

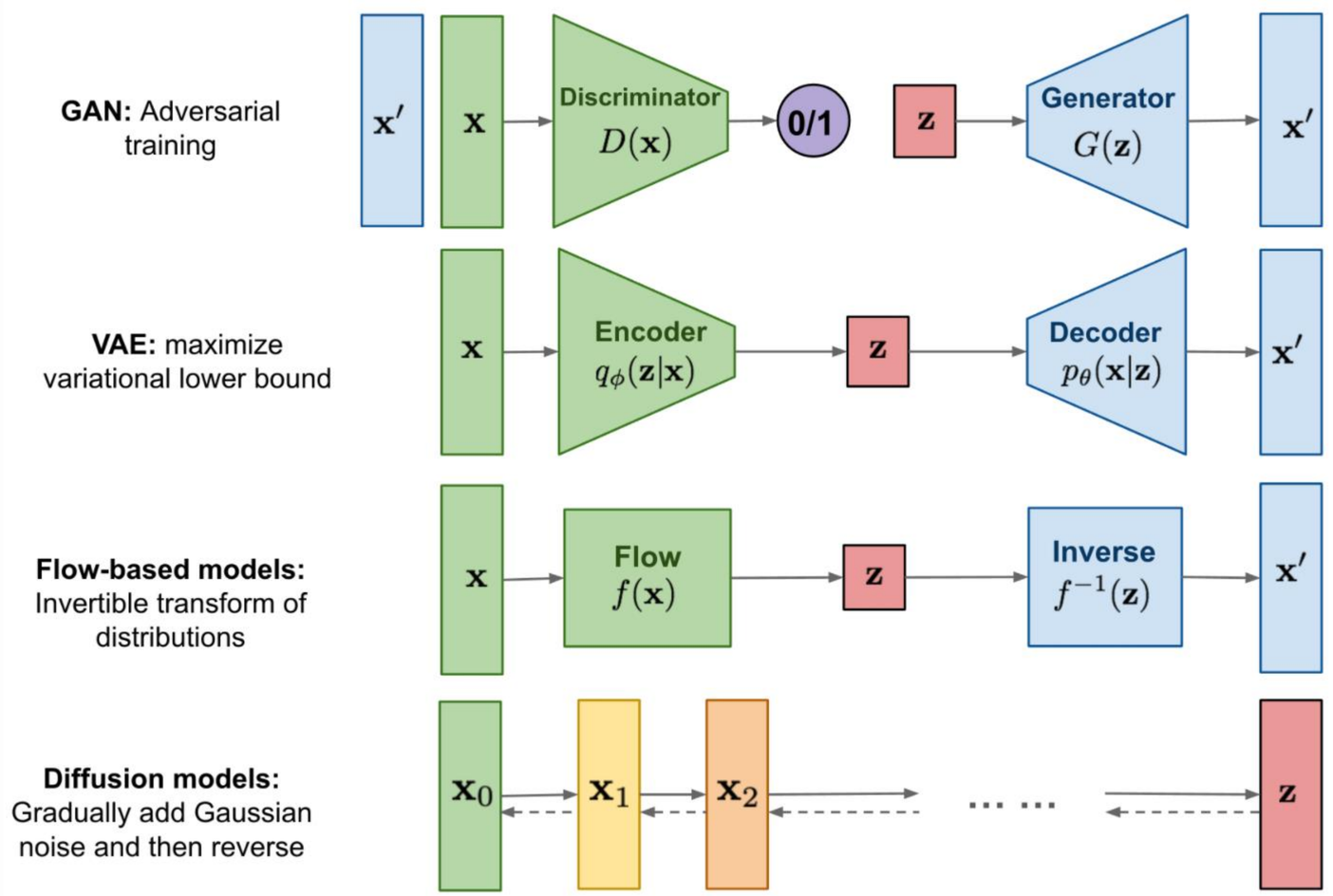
Generative model에는 GAN, VAE, Flow-based 등이 있다.

GAN: 훈련이 불안정함

VAE: surrogate loss에 의존함

Flow-based: Reversible 해야하기 때문에 architecture에 제약이 있다.

Diffusion model은 Markov chain을 이용하여, noise를 조금씩 더하고, reverse process로부터 학습한다.
각 latent가 다 같은 차원을 가진다.



Background

DM은 latent variable model이다.

$$p_\theta(x_0) = \int p_\theta(x_{0:T})dx_{1:T}\,, x_0 \sim q(x_0), D^{x_1} = D^{x_T}$$

Reverse process($p_\theta(x_{0:T})$): $p(x_T) = \mathcal{N}(x_T; 0, I)$로부터 시작해서 Gaussian transition을 훈련하는 Markov chain

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \qquad p_\theta\,(x_{t-1}|x_t) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

훈련은 VLB(variation lower bound)를 이용한 NLL을 optimization하여 수행됨

$$\mathcal{L}\text{: } \mathbb{E}[-\log p_\theta(x_0) \leq \mathbb{E}_q\left[-\log\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] = \mathbb{E}_q\left[-\log p(x_T) - \sum_{t\geq 1}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$$

Variance β_t는 trainable하게 설정할 수도 있고, constant로 설정할 수도 있다.

→ β_t가 작으면 $p = q$로 생각할 수 있다.

Background

Forward process에서 step t에서의 sample은 closed form으로 표현할 수 있다.

$\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ 일 때,

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

$\mathcal{L}$를 SGD함으로써, training이 이뤄지고, $\mathcal{L}$은 다시 표현하면,

$$\mathcal{L}: \mathbb{E}_q \left[D_{KL}\big(q(x_T|x_0) \parallel p(x_T)\big) + \sum_{t\geq 1} D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1) \right]$$

x_0에서 tractable한 $q(x_{t-1}|x_t, x_0)$와 p_θ의 분포를 같도록 학습

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\big(x_t; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I\big), \qquad where\ \ \tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}(\beta_t)}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t, \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

Rao-Blackwell theorem으로 closed form에서 계산한다.

Background

$\mathcal{L} = \mathbb{E}_q\left[D_{KL}\left(q(x_T|x_0) \parallel p(x_T)\right) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)\right]$의 유도과정은 아래와 같다.

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] \\
&= \mathbb{E}_q\left[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] \\
&= \mathbb{E}_q\left[-\log p(x_T) - \sum_{t>1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] \\
&= \mathbb{E}_q\left[-\log p(x_T) - \sum_{t>1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \cdot \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} - \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] \\
&= \mathbb{E}_q\left[-\log \frac{p(x_T)}{q(x_T|x_0)} - \sum_{t>1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} - \log p_\theta(x_0|x_1)\right] \\
&= \mathbb{E}_q\left[D_{KL}\left(q(x_T|x_0) \parallel p(x_T)\right) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)\right]
\end{aligned}
$$

첫번째 term은 x_T에 대한 term으로, variance가 constant로 설정되어 있기 때문에 논문에서는 무시된다.
두번째 term은 $t-1 \sim 1$ step에 관한 term이고, 마지막 term은 x_0에 대한 term으로, discrete output을 위해 나눠주는 것으로 생각된다.

Rao-Blackwell Theorem

모수(parameter) 추정 시, 충분한 data에서 conditional expectation을 취함으로써, estimator의 효율을 높인다.

$\theta = (\theta_1, \cdots, \theta_q)$가 unknown vector이고, $\mathrm{x} = (x_1, \cdots, x_n)$는 확률 분포, $p(x, \theta)$에서의 random sample이라고 할 때,
θ에 대한 estimator $\mathrm{t}(x) = \left(t_1(x), \cdots, t_q(x)\right), C(t) = \left(c_{i,j}\right), \quad i, j = 1 \sim q, \quad c_{i,j} = \mathbb{E}[(t_i(x) - \theta_i)(t_j(x) - \theta_j)]$
→ θ는 모수, x는 랜덤 변수

$E(t|s) = T(x)$가 θ와 독립적이도록, S를 vector valued statistic이라 한다.

이 때, Rao-Blackwell theorem에 따르면 $C(t) - C(T)$는 PD matrix다.

S의 표본이 충분히 크다면, $T(x)$가 θ에 독립적이라는 것은 충족된다. (S는 충분 통계량)
따라서, $\mathbb{E}[\Phi(t_r - \theta_r)] \geq \mathbb{E}[\Phi(T_r - \theta_r)], r = 1 \sim q$가 성립한다.

즉, estimator가 충분 통계량이 아닐 경우, 충분 통계량의 함수인 다른 estimator가 있고, 이 estimator는 MSE의 관점에서 더 효율적이다.

→ 충분 통계량이란 랜덤표본에서 모수를 구할 때, 조건으로 주어지면, 모수에 의존하지 않는 경우를 얘기한다.
→ 예를 들어, 가우시안 분포를 따르는 n개의 표본이 있을 때,
n개의 평균과 분산은 n개의 변수 모두 사용하는 것과 모수를 같게 예측하기 때문에 충분 통계량이 된다.
→ 즉, n개의 변수도 충분 통계량이 되지만, 평균, 분산 2개의 변수로 n개의 변수와 모수 추정에서 같은 효과를 낼 수 있다.

Diffusion model and Denoising autoencoder

Diffusion model(DM)의 구현에는 β_t, reverse architecture, gaussian distribution의 parameterization이 필요하다.
이를 위해 DM과 denoising score matching을 명시적으로 연결한다.

- Score matching

Score matching은 partition function이 다루기 힘든 비정규화 확률 밀도 모델에 적합한 Maximum log-likelihood의 대안이다.
→ Partition function(Z)은 열역학에 나오는 함수로, 에너지 시스템 상태 간의 분배에 관한 함수같다. (자세히 몰라도 될 듯)

Score matching은 training data의 noise에 robust하다.

Energy function(E, model의 output과 같은 것 같음)에 대해서 probability density model($\mathrm{p}(\mathrm{x};\theta)$)은
$\mathrm{p}(\mathrm{x};\theta) = \frac{1}{Z(\theta)}\exp(-E(x;\theta))$의 형태로 나타난다.

$Score$: $\psi(x;\theta) = \frac{\partial \log p(x;\theta)}{\partial x}$, score는 parameter의 gradient를 의미하는 듯 하다.

Score Matching

- Explicit Score Matching(ESM)

$$J_{ESM_q}(\theta) = E_{q(x)}\left[\frac{1}{2} \| \psi(x;\theta) - \frac{\partial \log q(x;\theta)}{\partial x} \|^2\right], \qquad q = true\ pdf$$

와 같이 나타낼 수 있지만 true pdf인 q를 모른다면 쓸 수 없다.

- Implicit Score Matching(ISM)

$$J_{ESM_q}(\theta) = E_{q(x)}\left[\frac{1}{2} \| \psi(x;\theta) - \frac{\partial \log q(x;\theta)}{\partial x} \|^2\right] = E_{q(x)}\left[\frac{1}{2} \| \psi(x;\theta) \|^2 + \sum_{i=1}^{d} \frac{\partial \psi_i(x;\theta)}{\partial x_i}\right] + C_1 = J_{ISM_q}(\theta), \qquad \psi_i(x;\theta) = \psi(x;\theta)_i = \frac{\partial \log p(x;\theta)}{\partial x_i}$$

은 $q(x), \psi(x;\theta)$가 미분 가능할 때 성립하고, ISM은 ESM과 같으면서도 q가 필요하지 않다.

- Denoising Score Matching

DAE와 SM의 관점에서 $\left(clean(x), corrupted(\tilde{x})\right), q_\sigma(\tilde{x}, x) = q_\sigma(\tilde{x}|x)q_0(x)$에서 DSM은 다음과 같이 정의된다.

$$J_{DSM_{q_\sigma}}(\theta) = E_{q_\sigma(x,\tilde{x})}\left[\frac{1}{2} \| \psi(\tilde{x};\theta) - \frac{\partial \log q_\sigma(\tilde{x}|x)}{\partial \tilde{x}} \|^2\right]$$

Gaussian kernel을 고려했을 때, $\frac{\partial \log q_\sigma(\tilde{x}|x)}{\partial \tilde{x}} = \frac{1}{\sigma^2}(x - \tilde{x})$로 나타나기 때문에, $corrupted\ \tilde{x}$는 $clean\ x$의 방향으로 움직인다.
$\log q_\sigma(\tilde{x}|x)$가 $\tilde{x}$에 대해서 미분 가능한 경우, $q_\sigma(\tilde{x}|x), q(x)$는 특정 형식에 의존하지 않고, $J_{ESM_{q_\sigma}} = J_{DSM_{q_\sigma}}$이다.

Forward process and $\mathcal{L}_T$

Forward process의 Variance β_t는 trainable하게 parameter로 설정할 수 있지만 논문에서는 constant한 값으로 설정했다.
T = 1000일 때, $\beta_1 = 10^{-4}, \beta_T = 0.02$로 설정했고, T시점의 variance는 1이 되어, gaussian noise와 같게 input data가 corrupted된다.

Variance가 constant이기 때문에 T시점의 data(noise가 일정한 분포를 따르기 때문)와 loss도 constant라고 볼 수 있고, 이에, 훈련 시 무시된다.

Reverse process and $\mathcal{L}_{1:T-1}$

논문에서는 reverse process를 다음과 같이 설정했다.
$q(x_{t-1}|x_t)$를 이용하여, gaussian noise $\mathcal{N}(0,I)$로부터 true sample을 얻을 수 있다. 하지만 $q(x_{t-1}|x_t)$를 추정하는 것은 target space에 대한 모든 sample이 필요하기 때문에 어렵다. 따라서 $p_\theta(x_{t-1}|x_t)$를 이용하여 근사한다.

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\big(x_{t-1}; \mu_\theta(x_t,t), \Sigma_\theta(x_t,t)\big), \qquad 1 < t \le T$$

$\Sigma_\theta(x_t,t) = \sigma_t^2 I$로, 시간에 독립적인 constant로 설정했다.
경험적으로, $\sigma_t^2 = \beta_t$로 설정하는 것과 $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$의 결과는 유사하다.
첫번째는 $x_0 \sim \mathcal{N}(0,I)$에 대해 optimal이고, 두번째는 x_0의 deterministic한 점에 대해서 optimal이다.

평균을 $\mu_\theta(x_t,t)$로 나타내고, $\mathcal{L}_t$에 의한 parameterization을 제시한다. $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t,t), \sigma_t^2 I)$에서

$$\mathcal{L}_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2} \| \tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t) \|^2\right] + C$$

로 표현할 수 있고, C는 θ에 대해 독립적이다.
직관적으로 μ_θ는 forward process의 posterior $q(x_{t-1}|x_t)$의 평균인 $\tilde{\mu}_t$를 예측하도록 학습된다. (forward는 closed form)

Reverse process and $\mathcal{L}_{1:T-1}$

Slide 6의 식을 $x_t(x_0,\epsilon)=\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon\sim\mathcal{N}(0,I)$로 reparametrizing하고 이용하여, $\mathcal{L}_{t-1}$을 확장하면,

$$\mathcal{L}_{t-1}-C=\mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\parallel\tilde{\mu}_t(x_t(x_0,\epsilon),\frac{1}{\sqrt{\bar{\alpha}_t}}(x_t(x_0,\epsilon)-\sqrt{1-\bar{\alpha}_t}\epsilon))-\mu_\theta(x_t(x_0,\epsilon),t)\parallel^2\right]$$

$$=\mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\parallel\frac{1}{\sqrt{\alpha_t}}(x_t(x_0,\epsilon)-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)-\mu_\theta(x_t(x_0,\epsilon),t)\parallel^2\right]$$

즉, μ_θ는 주어진 $x_t(input)$에 대해서 $\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$를 예측하도록 학습된다.

따라서 위의 식들을 통해,

$$\mu_\theta(x_t,t)=\tilde{\mu}_t\left(x_t,\frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t-\sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t)\right)\right)=\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right), \qquad \epsilon_\theta=\text{predict}\,\epsilon\ from\ x_t$$

의 형식으로 reparameterization할 수 있다.

Reverse process and $\mathcal{L}_{1:T-1}$

$\tilde{\mu}_t$을 예측하는 식을 조금 더 자세히 써보면 x_0에 대해서 reverse condition probability는 tractable한 상황에서 $q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}(x_t, t), \tilde{\beta}_t(x_t, t)\right)$으로 가정하면,

$$q(x_{t-1}|x_t, x_0) = q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{\left(x_t - \sqrt{\alpha_t}x_{t-1}\right)^2}{\beta_t} + \frac{\left(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0\right)^2}{1-\bar{\alpha}_{t-1}} - \frac{\left(x_t - \sqrt{\bar{\alpha}_t}x_0\right)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}x_0\right)x_{t-1} + C(x_t, x_0)\right)\right)$$

$$\tilde{\beta}_t = 1/\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t$$

$$\tilde{\mu}_t(x_t, x_0) = \left(\frac{\sqrt{\alpha_t}}{\beta_t}x_t + \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}x_0\right)\Big/\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0, \qquad x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\alpha_t}\epsilon\right)$$

$$\tilde{\mu}_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\alpha_t}\epsilon\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$$

와 같이 μ_θ이 $\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$을 예측하도록 한다.

Reverse process and $\mathcal{L}_{1:T-1}$

Reverse process에서

$x_{t-1} \sim p_\theta(x_{t-1}|x_t)$를 sampling하기 위해,

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z,\ \ z \sim \mathcal{N}(0, I)$$

을 계산한다.

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Algorithm 2에 나타난 전체 sampling process는 data density의 trained gradient인 ϵ_θ의 Langevin dynamics와 비슷하다.

또한 $\mathcal{L}_{t-1} - C$을 $\mu_\theta(x_t, t)$을 이용하면, 다음과 같이 t level의 multiple noise에 대한 DSM과 닮은 형식으로 간단히 할 수 있다.

$$\mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)} \parallel \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 - \sqrt{1-\bar{\alpha}_t}\epsilon, t) \parallel^2\right]$$

위의 식이 Langevin-like reverse process for variational bound와 같기 때문에, DSM의 optimization은 Langevin dynamics 형식의 sampling chain의 finite-time marginal에 대한 variational inference를 사용하는 것과 같다고 여길 수 있다.

Reverse process and $\mathcal{L}_{1:T-1}$

즉, μ_θ가 $\tilde{\mu}_t$를 근사하도록 reverse process를 학습시킬 수 있고, reparameterization을 이용하여,

ϵ를 근사하도록 학습시킬 수도 있다.

$\epsilon - parameterization$은 효율적고 이에 대한 비교는 experiment 부분에서 $\tilde{\mu}_t$와 비교하여 진행한다.

Langevin Dynamics

Langevin dynamics는 Markov chain update에서 $\nabla_x \log p(x)$만을 이용하여
확률 밀도 $p(x)$에서 sampling할 수 있는 방식을 SGLD(Stochastic Gradient Langevin Dynamics)라고 한다.

ϵ가 step size라고 할 때,

$$x_t = x_{t-1} + \frac{\epsilon}{2}\nabla_x p(x_{t-1}) + \sqrt{\epsilon} Z_t, \qquad Z_t \sim \mathcal{N}(0, I)$$

Local minimize collapse 방지를 위해 gaussian noise를 update parameter에 넣는다.

Data scaling, Reverse process decoder, and $\mathcal{L}_0$

$p(x_T)$로부터 시작한 Neural Network Reverse Process가 지속적으로 동일한 scale의 input에서 수행될 수 있도록 {1 ~ 255}의 range를 가지는 image data를 [-1, 1]의 range를 가지는 data로 scaling시켜, input으로 사용한다.

Discrete한 output을 가지는 image data에 대해 reverse process의 마지막 과정을 Gaussian $\mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I)$로부터 유도된 independent discrete decode로 설정하여, discrete log likelihood term으로 만들면 다음과 같이 표현할 수 있다.
(D는 data의 dimension을 의미)

$$p_\theta(x_0|x_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}\left(x; \mu_\theta^i(x_1, 1), \sigma_1^2\right) dx\,, \qquad \delta_+(x) = \begin{cases} \infty, & if\ x = 1 \\ x + \frac{1}{255}, & if\ x < 1 \end{cases}, \qquad \delta_-(x) = \begin{cases} -\infty, & if\ x = -1 \\ x - \frac{1}{255}, & if\ x > -1 \end{cases}$$

x_1으로부터 x_0을 sampling하여 나온 discrete한 output을 q에 대해서 loss를 구하는 형식으로 생각할 수 있다. (CE와 비슷한 듯)

Simplified training Objective

이전의 reverse process와 decoder와 같이 $\mathcal{L}_{1:T-1}, \mathcal{L}_0$로 구성된 variational bound는 명백히 θ에 대해 미분가능하고, training에 적용할 수 있다.

Variational bound를 간단히 하면, t가 1 ~ T에 대해 uniform할 때 다음과 같이 표현할 수 있다.

$$\mathcal{L}_{simple}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 - \sqrt{1-\bar{\alpha}_t}\epsilon, t) \|^2\right]$$

정확한 동작한 Algorithm 1과 같다.

Slide 13의 식에 비해, coefficient가 없기 때문에 위의 식은 weighted variational bound라고 할 수 있다.

Weighted variational bound는 t가 낮을 때, 가중치가 낮아지는데 t가 높을수록 어려운 task이기 때문에
어려운 task에 훈련을 집중 할 수 있게 하여, 다 좋은 성능을 이끌어 낸다.

Experiments

$\mathrm{T} = 1000, \beta_1 = 10^{-4}, \beta_T = 0.02$으로 linear하게 증가하도록 설정했다.
이 constant는 variance가 작기 때문에, forward와 reverse process가 거의 동일하다.
또한 step T에서는 $\mathcal{N}(0, I)$과의 KL divergence가 최대한 작도록 설정되었다.

Reverse process의 architecture는 unmasked PixelCNN++의 Unet backbone(wide resnet 기반)에 group normalize와 같다.

또한 시간 t에 대해 transformer의 sinusoidal position embedding을 사용하여, parameter를 공유하도록 했다.

32x32 pixel size의 영상에 대해서는 (32x32 ~ 4x4) 4개의 resolution의 architecture를 사용했고,
256x256 pixel size의 영상에는 256부터 6개의 resolution을 사용했다.
각 resolution마다 2개의 residual conv block, 16x16 resolution에는 conv block끼리 self-attention을 사용했다.

EMA의 decay는 0.999를 사용했다.

Sample Quality

Table 1은 Inception, FID score, NLL codelength를 나타낸다.

전체적으로 다른 방식에 비해 뛰어난 성능을 나타냈다고 언급되어 있다.

NLL의 경우 simple loss보다 true loss가 더 좋았지만, sample quality는 simple loss가 더 좋았다.

NLL test는 ln으로 나타나는 loss를 log2 형식으로 바꾸어, 각 픽셀당 필요한(계산되는) bit 수를 나타내는 것 같다. (정확하지 않음)
Data의 range를 0 ~ 256(8bit)으로 바꾸고 color space는 다 다른 pixel로 생각하여, 각 bit를 도출해내는데 까지 필요한 계산량을 나타낸다고 이해했다.

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			$\leq$ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87$\pm$0.12	25.32	
SNGAN [39]	8.22$\pm$0.05	21.7	
SNGAN-DDLS [4]	9.09$\pm$0.10	15.42	
StyleGAN2 + ADA (v1) [29]	**9.74** $\pm$ 0.05	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67$\pm$0.13	13.51	$\leq$ 3.70 (3.69)
Ours (L_{simple})	9.46$\pm$0.11	**3.17**	$\leq$ 3.75 (3.72)

Reverse process parameterization and training objective ablation

Table 2는 $\tilde{\mu}_t$와 ϵ에 대한 reverse process를 비교한 표다.

Baseline으로 사용되는 $\tilde{\mu}_t$의 경우 unweighted variational bound에서만 좋은 성능을 보였고, trainable한 variance에서는 불안정한 모습을 보였다.

제시된 ϵ는 unweighted variational bound에 대해서도 baseline과 비슷한 성능을 보였고, simple loss에 대해서는 더 뛰어난 성능을 보였다.

Trainable한 variance에서는 fixed variance에 비해 모두 불안정한 모습을 보였다.

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
$\tilde{\mu}$ prediction (baseline)		
L, learned diagonal Σ	7.28 ± 0.10	23.69
L, fixed isotropic Σ	8.06 ± 0.09	13.22
$\|\tilde{\mu} - \tilde{\mu}_\theta\|^2$	–	–
ϵ prediction (ours)		
L, learned diagonal Σ	–	–
L, fixed isotropic Σ	7.67 ± 0.13	13.51
$\|\tilde{\epsilon} - \epsilon_\theta\|^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	**3.17**

Progressive generation

Algorithm 2를 사용하여, reverse process의 sampling에서 예측된 x_0의 결과를 얻을 수 있다.

Unconditional한 progressive generate의 결과는 아래의 그림과 같다.

Figure 6: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).

Step t로부터 시작하여 generate한 결과는 아래의 그림과 같다. 낮은 t에서는 결과가 유사한 것을 확인할 수 있다.

Figure 7: When conditioned on the same latent, CelebA-HQ 256×256 samples share high-level attributes. Bottom-right quadrants are $\mathbf{x}_t$, and other quadrants are samples from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

Connection to autoregressive decoding

Slide 6의 variational bound는 다음과 같이 다시 표현할 수 있다.

$$\mathcal{L} = D_{KL}\big(q(x_T) \parallel p(x_T)\big) + \mathbb{E}_q\left[\sum_{t\geq 1} D_{KL}(q(x_{t-1}|x_t) \parallel p_\theta(x_{t-1}|x_t))\right] + H(x_0)$$

T의 길이를 forward process를 정의하는 data dimension으로 설정하고, $q(x_t|x_0)$는 x_0를 t번째 dimension이 masking한 것으로 간주하여, $q(x_T)$는 blank image로 설정한다.

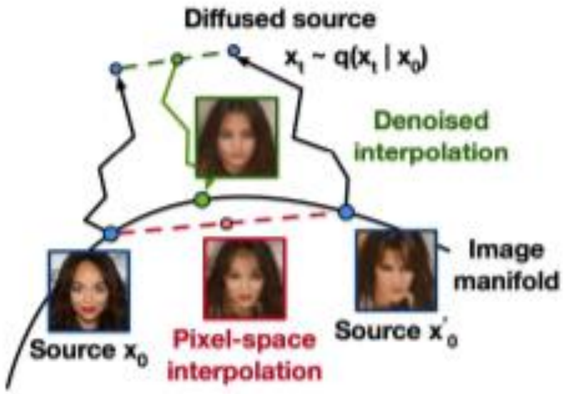
이로 인해, 훈련에서 $D_{KL}\big(q(x_T) \parallel p(x_T)\big) = 0$이고, 선택된 step t에서 $D_{KL}(q(x_{t-1}|x_t) \parallel p_\theta(x_{t-1}|x_t))$은 t+1 ~ T에 대해선 변하지 않고, 주어진 step t를 t+1 ~ T에서 예측한 것으로 간주된다.

따라서 DM은 data의 좌표를 재정렬하는 것으로는 표현할 수 없는 generalize한 bit order를 갖는 autoregression model의 하나로 간주할 수 있다.

Interpolation

두 이미지를 단순히 pixel space에서 interpolation을 하게 된다면, artifact가 발생하게 된다.

이러한 artifact는 forward process q를 stochastic encoder로 사용하여,
x_T의 space에서 interpolation을 수행하게 된다면 artifact를 없앨 수 있다.



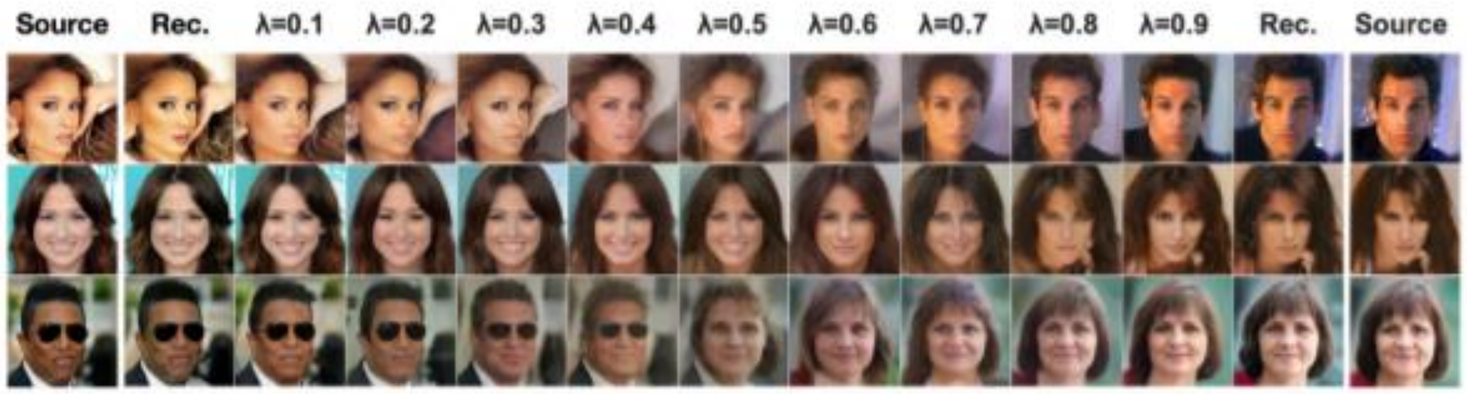


Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

Conclusion

Diffusion model은 고품질 이미지 생성을 제안했고, Markov chain training, Denoising Score Matching, annealed Langevin dynamics, autoregressive를 위한 Diffusion model과 Variational Inference간의 연결점을 찾았다.

또한 diffusion model은 image data에 대해 뛰어난 inductive bias를 가지고 있는 것으로 보여, 다른 분야에도 유용성이 있을 것이라 기대된다.

- Pros
  - Tractability와 flexibility는 원래 상충되는 성능이지만 DDPM에서는 두 성능이 모두 좋다
- Cons
  - Sample generate가 Markov chain에 의존하는데 이 과정에서 많은 시간과 계산이 든다.
  - Improved DDPM도 GAN보다 많은 시간과 비용이 든다.

“ Denoising Diffusion Probabilistic Model. ”


UNIST
ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY
2007

THANK YOU