<Target> : 확산모델 (디노이징 하는!)

DDPM

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

cs.LG] 16 Dec 2020

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at `https://github.com/hojonathanho/diffusion`.

<Reference>

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

https://cvpr2022-tutorial-diffusion-models.github.io/

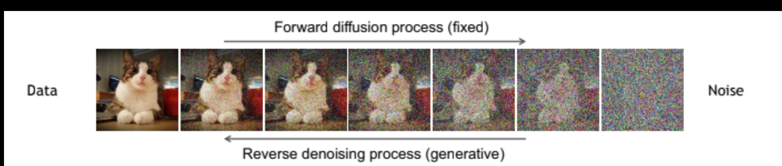
Denoising Diffusion Models
Learning to generate by denoising

<history ⇒ past ~ to SOTA>

1. **Denoising Diffusion Probabilistic Models (DDPM, 2020):**
   - 제안자: Ho et al.
   - 개요: 확산 과정과 역확산 과정을 통해 이미지 데이터를 점진적으로 노이즈화하고, 이를 역으로 복원하여 이미지를 생성하는 모델입니다.
   - 주요 특징: Markovian 확산 과정과 비슷한 역확산 과정을 사용합니다.
2. **Denoising Diffusion Implicit Models (DDIM, 2020):**
   - 제안자: Song et al.
   - 개요: DDPM의 효율성을 개선한 모델로, 직접적인 역확산을 통해 빠른 샘플링을 가능하게 합니다.
   - 주요 특징: 비-Markovian 샘플링 방법을 사용하여 더 빠르고 효율적인 이미지 생성을 구현합니다.
3. **Improved Denoising Diffusion Probabilistic Models (Improved DDPM, 2021):**
   - 제안자: Nichol and Dhariwal
   - 개요: DDPM의 성능을 개선하기 위해 하이퍼파라미터 튜닝과 아키텍처 수정 등을 도입한 모델입니다.
   - 주요 특징: 성능 향상과 더 높은 품질의 이미지 생성을 목표로 합니다.
4. **Latent Diffusion Models (LDM, 2021):**
   - 제안자: Rombach et al.
   - 개요: 고해상도 이미지를 생성하기 위해 잠재 공간(latent space)에서 확산 과정을 수행하는 모델입니다.
   - 주요 특징: 효율성을 높이기 위해 잠재 공간에서의 연산을 활용합니다.
5. **Stable Diffusion (2022):**
   - 제안자: Stability AI
   - 개요: LDM의 발전된 형태로, 높은 품질의 이미지를 빠르게 생성할 수 있는 모델입니다.
   - 주요 특징: 일반 사용자도 쉽게 사용할 수 있도록 접근성을 높인 모델입니다.

DDPM (2020)
↓
DDIM (2020)
↓
Improved DDPM (2021)
(Nichol et al.)
↓
Latent Diffusion model
(2022)
↓
stable Diffusion (2022)

오늘 정리할 것은 SD(2022)의
조상격 모델인 DDPM(2020)



Learning to Generate by Denoising

< What Diffusion Model Does? >

· 확산 모델은 노이즈를 생성해서 점진적으로 데이터에 추가하고 이를 역으로 복원하면서 새로운 데이터를 생성함 (출처 GPT4o)

학습 과정 (DDPM)

① Forward Diffusion Process (점진적 노이즈화 과정)

입력데이터를 점진적으로 노이즈화 시켜 x_T에 도달함

$(t = 1, \cdots 2 \cdots T)$

$$q(x_t | x_{t-1}) = N(x_t ; \sqrt{\alpha}\, x_{t-1}, (1-\alpha_t) I)$$

여기서 α_t는 노이즈의 크기 조절 파라미터

$$q(x_{1:T} | x_0) = \prod_{t=1}^{T} q(x_t | x_{t-1}) \quad \cdots\cdots (1)$$

↳ 전체 노이즈화 과정

② Reverse Diffusion process (역 확산 과정)

노이즈화된 데이터 x_T에서 원본 데이터 x_0로 복원 하는 과정

여기서 역확산 과정의 확률 분포를 모델링 함

(2) $\cdots\cdots$ ↳ $p_\theta(x_{t-1} | x_t) = N(x_{t-1} ; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$

(p_θ μ_θ Σ_θ 는) 신경망을 통해 학습되는 파라미터들.

③ 파라미터 (p_θ μ_θ Σ_θ)를 Gradient Descent로 최적화.

역확산 과정의 파라미터를 변분 바운드 (VB. Variational Bound)를 최소화 하는 방향임

Very similarly to VAE and thus we can use the variational lower bound to optimize the negative log-likelihood.

$$\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\le -\log p_\theta(\mathbf{x}_0) + \boxed{D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \| p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0))} \ge 0 \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}\Big] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)\Big] \\
&= \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big]
\end{aligned}$$

1

Point !!!

05_03 lecture note 의 16 페이지

여기에서 q는 $X_{1:T} \sim q(X_{1:T}|X_0)$ $= q(X_1, X_2, \cdots, X_T)$

<loss function>

$$\text{Let } L_{\mathrm{VLB}} = \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \ge -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)$$

2

$L_{VLB} = E_{q(x_{0:T})}[-\log p_\theta(x_{0:T}) + \log q(x_{1:T}|x_0)] \cdots (3)$

↳ (1)식 Forward Diffusion process

↳ (2) 식 (Reverse Diffusion process)

why? L_{VLB}?

↳ x_0가 주어졌을때 $D_{KL}(q \| p_\theta)$를 최소화 하는게 목표임

(p_θ를 q분포와 최대한 가깝게 하는것임)

↳ $E_q\Big[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\Big]$ 최소화 $\cdots (4)$

$= D_{KL}(q(x_{1:T}|x_0) \| p_\theta(x_{1:T}|x_0))$를 최소화

↳ $q(x_{1:T}|x_0) \approx p_\theta(x_{1:T}|x_0)$을 갖는것

x_0가 Given 되었음을 주의!

< L_{VLB} 최적화. 식(4)를 풀어서 쓰면 >

q는 $q(X_{0:T}) = q(X_0, X_1, \cdots X_T)$

$$
\begin{aligned}
L_{\text{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \Big(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\Big) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[\underbrace{\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)}}_{\text{term}} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{\text{term}}\Big]
\end{aligned}
$$

<불필요한 항 1>

X_T는 이제 거의 Noise이다.
$q(X_T|X_0)$나 $p_\theta(X_T)$나 거의 $N(0, I)$로 의미가 없다.

실제로 최소화 하는 항이다.

<불필요한 항 2>

β_1을 매우 작게 주면 X_1는 거의 X_0이므로 관리가 가능한 항

$$L_{VLB} = \mathbb{E}_q\Big[\text{상수} + \boxed{\sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)}} - \text{상수}\Big]$$

우리에게 x_0, x_t 가 있어서 ϵ_θ를 구할수 있음

$$
\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= \mathcal{N}\Big(\frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\Big), \tilde{\beta}_t\mathbf{I}\Big) \\
p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &= \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t,t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)) \\
&= \mathcal{N}\Big(\frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\Big), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)\Big)
\end{aligned}
$$

대입

$$\mathbb{E}_q\Big[\boxed{\sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)}} \longrightarrow L_{t-1} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2\Big]$$

∴ 우리는 L_{VLB}를 계산가능해짐.

< Simplified L_{VB} : Ignoring the weighting term >

training. However, we found it beneficial to sample quality (and simpler to implement) to train on the following variant of the variational bound:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{14}$$

↳ 그냥 이렇게 쓰는 좋음을 Empirically 알아냄.

Ho et al. NeurIPS 2020 observe that simply setting $\lambda_t = 1$ improves sample quality. So, they propose to use:

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I}), t \sim \mathcal{U}(1,T)}\left[\|\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon}_{\mathbf{x}_t}, t)\|^2\right]$$

For more advanced weighting see Choi et al., Perception Prioritized Training of Diffusion Models, CVPR 2022.

weighting term은 다르게 줄수있음

< Summary > → ① Training

② Sample Generation (Inference)

Algorithm 1 Training

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

Algorithm 2 Sampling

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