[week6] DDPM

내용이 너무 어려워서 참고할만한 기술블로그를 찾고 해당 블로그의 내용을 정리하듯이 필사하며 공부하였습니다.
참고한 블로그는 맨 아래에 reference로 달아두었습니다.

Abstract

- 본 논문에서는diffusion probabilistic models를 이용한 고품질 이미지 생성 결과를 제시한다.
- 이는 비평형 열역학(nonequilibrium thermodynamics)의 개념에서 영감을 받은 latent variable models의 일종이다.
- denoising score matching과 Langevin dynamics 사이의 새로운 연결 고리를 기반으로 설계된 weighted variational bound를 통해 모델을 학습하며, 이로부터 최고의 성능을 얻었다.
- 또한, 제안된 모델은 progressive lossy decompression 방식으로 해석될 수 있는 자연스러운 생성 구조를 가지며, 이는autoregressive decoding의 일반화된 형태로 해석될 수 있다.

Introduction

- DDPM은 주어진 이미지에 time에 따른 상수의 파라미터를 갖는 작은 가우시안 노이즈를 time에 대해 더해나가는데, image가 destroy하게 되면 결국 noise의 형태로 남을 것이다. (normal distribution을 따른다.)
- 이런 상황에서 normal distribution 에 대한 noise가 주어졌을때 어떻게 복원할 것인가에 대한 문제

  ⇒ 주어진 Noise를 통해서 완전히 이미지 복구 = image generation 성공!
- 논문에서는 diffusion probabilistic models의 과정을 보여줌
- diffusion model은 유한한 시간 뒤에 이미지를 생성하는 variational inference을 통해 훈련된 Markov chain을 parameterized한 형태

Background

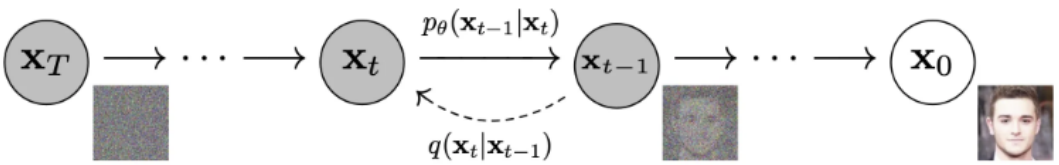

Figure 2: The directed graphical model considered in this work.

Denoising Diffusion Probabilistic Models 2020

- Forward Diffusion Process (q), Reverse Diffusion Process (p)
- 둘의 resolution은 서로 같음
- Objective Function (L)
  : 주어진 noise에 대해 어떻게 noise를 점진적으로 걷어낼 것이냐의 문제
  : *xt가 들어왔을 때 xt−1을 예측할 수 있게 된다면, 우리는 x0 또한 예측할 수 있다.*
- 생성된 이미지의 log likelihood를 최대화 = negative log likelihood로 최소화

Loss Function

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

Diffusion Model의 Loss Function

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

Diffusion Model의 최종적인 Loss Function

Diffusion models and denoising autoencoders

Objective Function (L)

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T))}_{L_T}+\sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

Diffusion Model의 최종적인 Loss Function

- 여기서 가장 중요한 term은 *Lt*−1이다. 우리는 *x*0부터 시작하여 conditional하게 식을 전개하다보면, tractable한 forward process posterior q(xt−1 ∣ xt,x0)의 정규분포를 알 수 있는데, 이를 바탕으로 KL divergence를 계산하면 우리가 결과적으로 학습하고자 하는 pθ(xt−1 ∣ xt)를 학습시킬 수 있다.

1. **Forward Diffusion Process (LT)**
   - 논문에서는 forward process variances인 *β*를 learnable한 파라미터로 두는게 아니라 상수로서 Fix하기 때문에 *LT*는 고려를 하지 않아도 된다.
   - 이 loss term은 항상 0에 가까운 상수이며, 학습과정에서 무시된다.
2. **Reverse Diffusion Process (L1:T−1)**
   - 이를 계산하기 위해서 첫 번째로 q(xt−1 ∣ xt,x0)의 분포를 알아내고, 두 번째로 pθ(xt−1 ∣ xt)를 알아내기 위해 ∑θ와 μθ를 알아내야 한다.
     1. q(xt−1 ∣ xt,x0)

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

     2. pθ(xt−1 ∣ xt)

$$p_\theta(x_{t-1}|x_t)=\mathcal{N}(x_{t-1};\ \mu_\theta(x_t,t),\ \sum_\theta(x_t,t))$$

     3. ∑θ(xt,t) : 학습이 필요없는 term

$$\sum_\theta(x_t,t)=\sigma_t^2 I$$

     4. μθ(xt,t)

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t))\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) \quad (11)$$

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \quad (12)$$

5. Sampling
    - 위 과정을 통해서 μθ를 얻어내면, 이를 활용해서 우리는 pθ(xt−1 ㅣ xt)로부터 xt−1을 샘플링할 수 있다.

Algorithm 2 Sampling

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

샘플링 알고리즘

6. Lt-1
    - 위의 재료들을 잘 조합하여 D_KL을 계산하면 Lt-1을 얻어낼 수 있다.

$$\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \text{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2\right] + C \quad (8)$$

$$L_{t-1} - C = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon})\right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t)\right\|^2\right] \quad (9)$$

$$= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t)\right\|^2\right] \quad (10)$$

7. Lsimple(θ)
    - loss function을 epsilon에 대한 식의 형태로 다시 한 번 표현할 수 있다. (= simplified objective function)
    - 이러한 loss function을 통해 training을 하면 학습이 좀 더 잘 된다.

- 큰 t에 대해서도 network 학습이 가능하기 때문에 매우 효과적이다.

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2 \right] + C \tag{8}$$

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

8. Training (DDPM의 training 과정 요약)

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

학습 알고리즘

Experiments

- DDPM image generation score

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]			≤ 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 ± 0.12	25.32	
SNGAN [39]	8.22 ± 0.05	21.7	
SNGAN-DDLS [4]	9.09 ± 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	$\mathbf{9.74} \pm 0.05$	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	**3.17**	≤ 3.75 (3.72)

Figure 3: LSUN Church samples. FID=7.89

Figure 4: LSUN Bedroom samples. FID=4.90

References

- https://jang-inspiration.com/ddpm-2

$$\mathbb{E}_q \Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Bigg]$$

Diffusion Model의 최종적인 Loss Function