Denoising Diffusion Probabilistic Models

Intuition

잉크나 연기는 시간이 지나면 고르게 퍼져나갈 것이다.(diffusion)

결국 고르게 분포되어 밀도가 uniform해질 것이다.

출처 : https://gifer.com/en/GWtc

Intuition

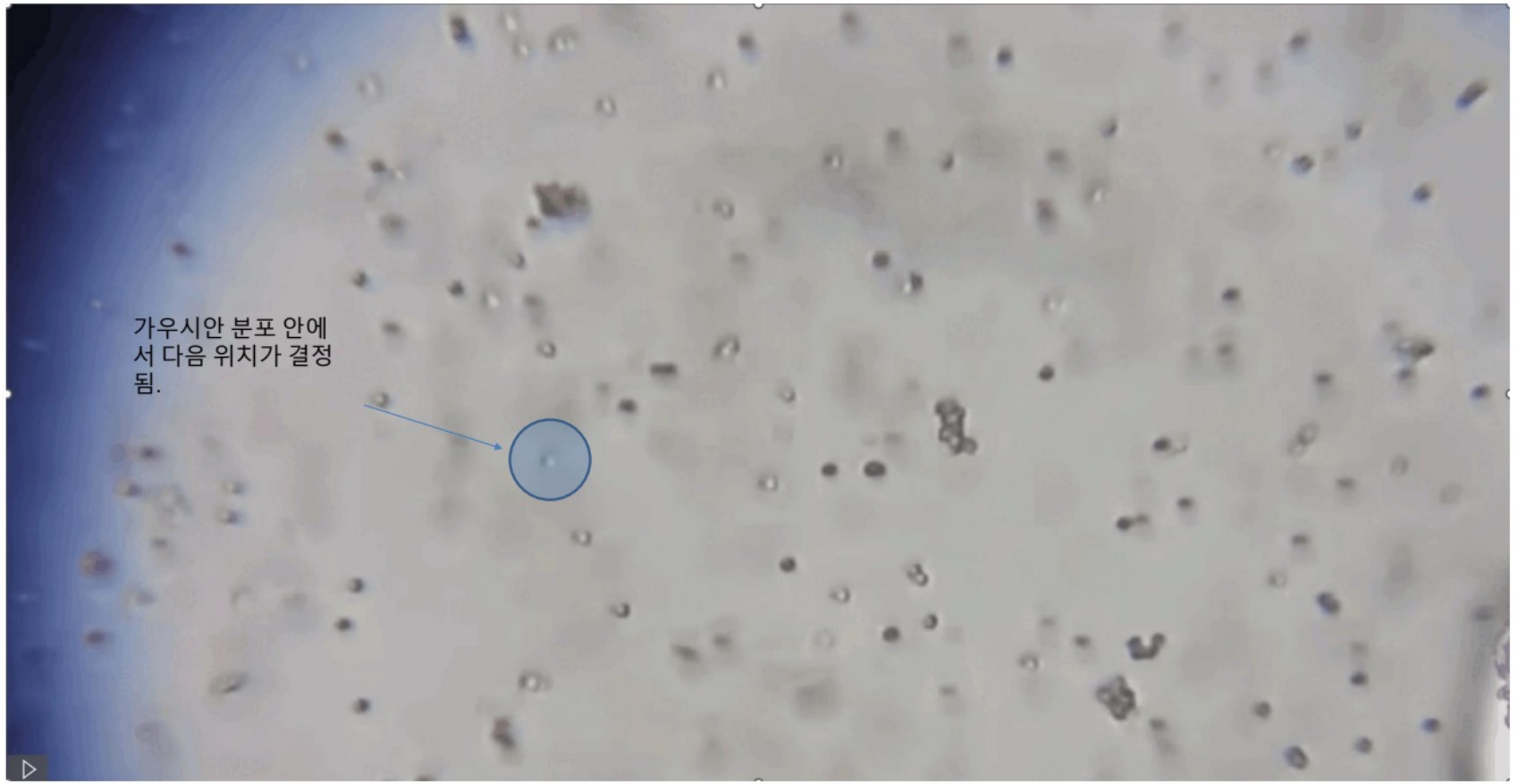
Uniform에서 시작해서 다시 처음상태로 되돌릴 수 있다면?

→ 딥러닝을 가지고 해보자!

출처 : https://gifer.com/en/GWtc

Intuition

작은 sequence에서의 확산은 forward와 reverse 모두 가우시안일 수 있다. (물리적으로)



출처 : https://www.youtube.com/watch?v=uFoGaIVHfoE

Prerequisite

- Prob & Stats
  - $X = N(\mu, \Sigma)$ mean, variance
    - $\varepsilon \sim N(0, I)$
  - $E(aX + b) = aE(x) + b$
  - $V(aX + b) = a^2V(X)$
  - $\sigma(aX + b) = |a|\sigma(X)$ standard deviation = var^2
  - $P(B \mid A) = P(A \mid B)\frac{P(B)}{P(A)}$
- Distribution
  - $q(x)$: *real distribution*
  - $p_\theta(x)$: *network distribution*

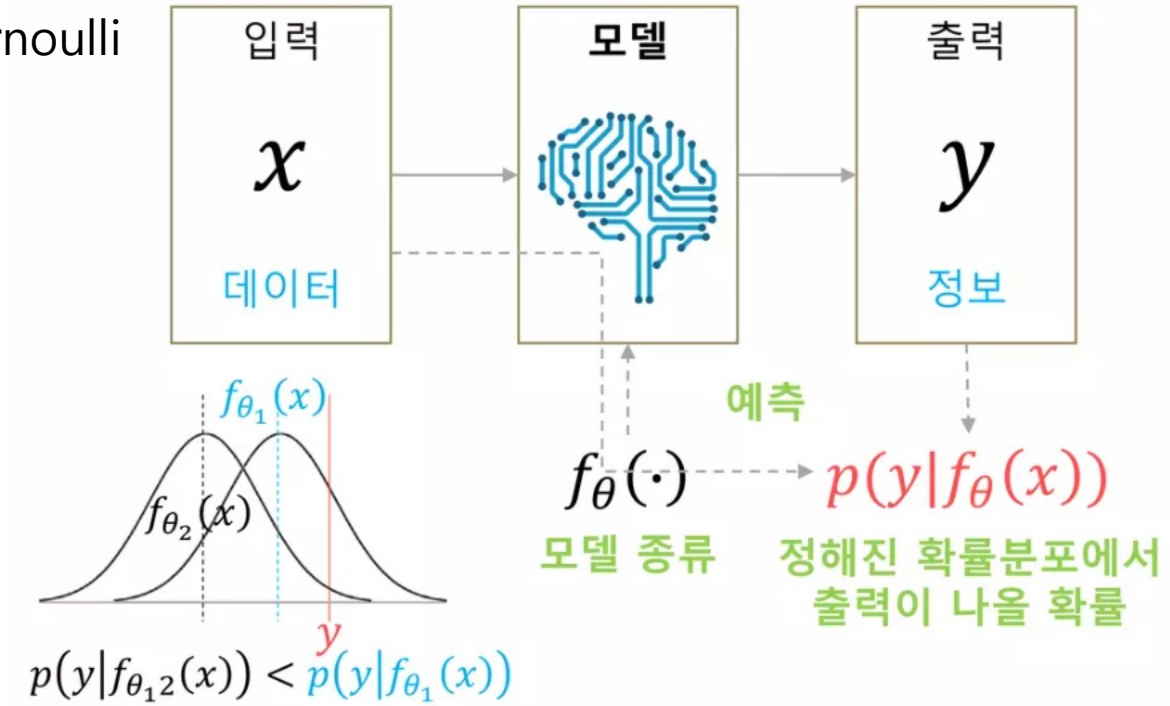
출처 : https://cvpr2022-tutorial-diffusion-models.github.io/

DeepLearning의 Maximum Likelihood 관점 해석

P(Y|f(x))에 대한 분포를 가정 ex)Gaussian, Bernoulli

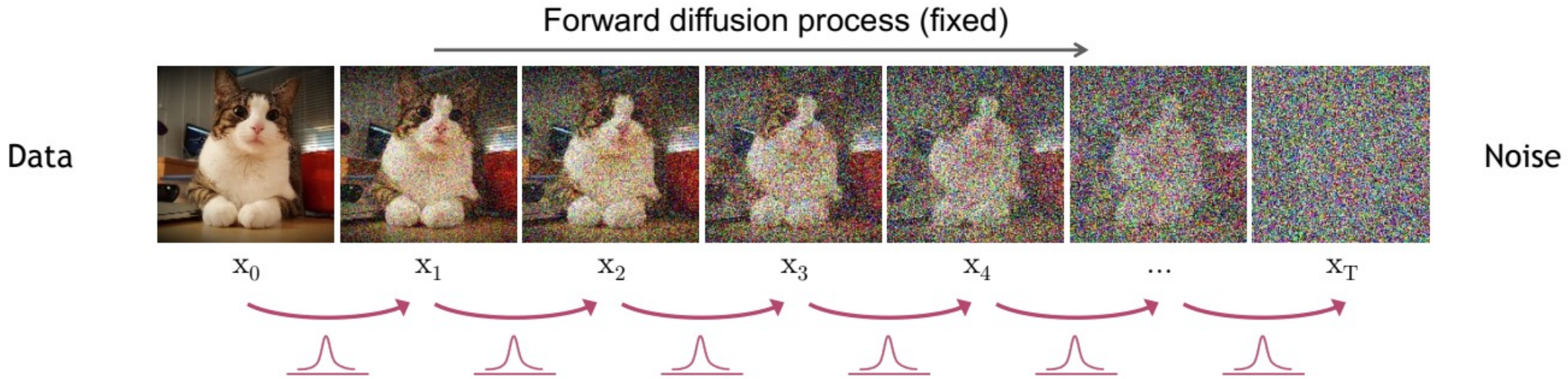
딥러닝 모델을 통한 output은 분포에 대한 Likelihood를 maximize하는 파라미터 θ를 추정하는 것 (가우시안 일때는 평균과 분산)

그렇다면
Loss = $-\log(p(y|f_\theta(x)))$



출처: 이활석 - 오토인코더의 모든것

Diffusion Model Overview

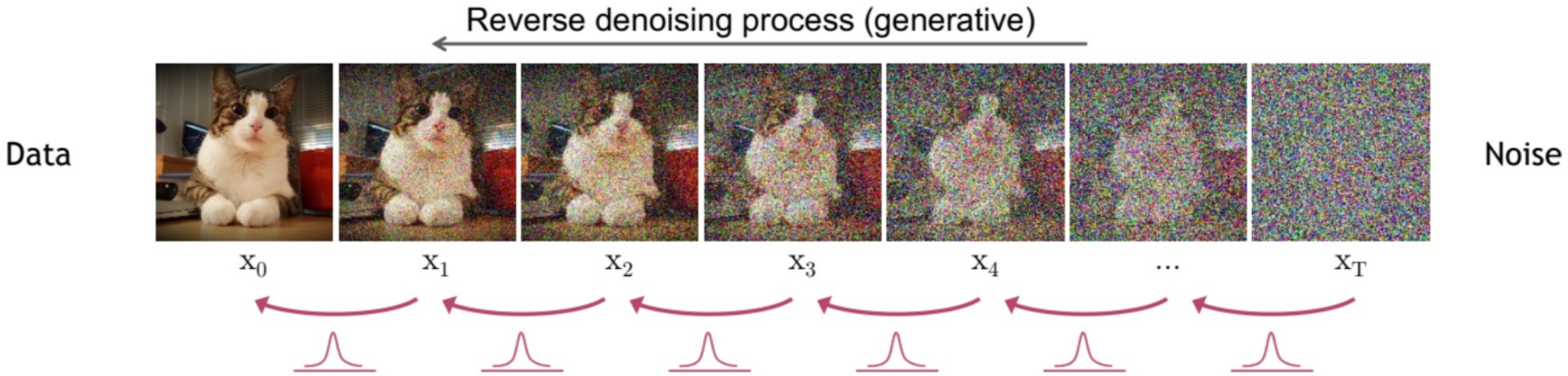


1. Forward Process

Noise를 점진적으로 더한다

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

Diffusion Model Overview



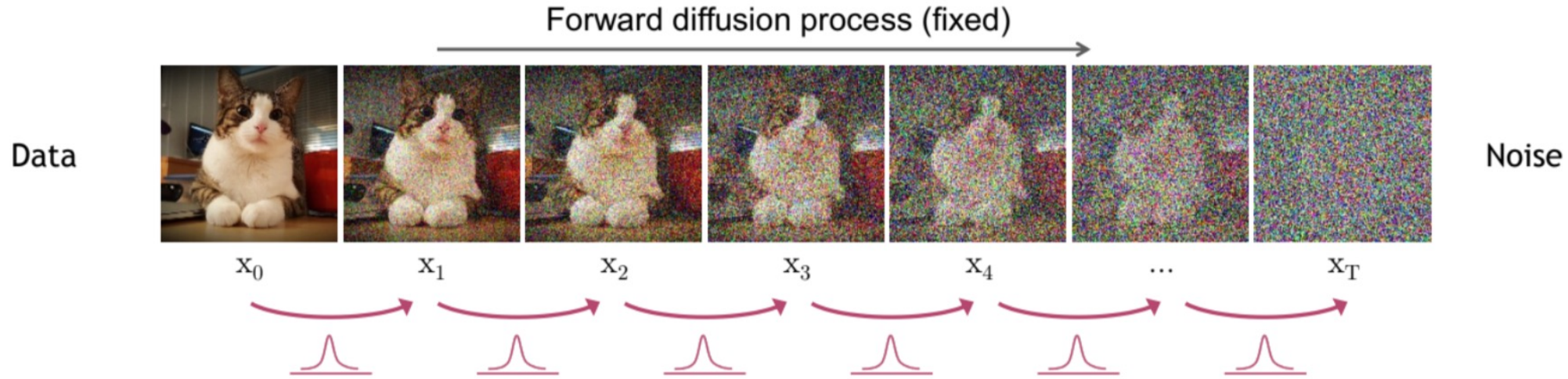
2. Reverse Process

Noise를 점진적으로 제거한다.
→ 이를 통해 Image를 Generate 할 수 있다.

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

Forward Process

The formal definition of the forward process in T steps:



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

β_t는 0.0001부터 0.02까지 scheduling Ex) 총 1000Step이면 0.0001부터 점점커져서 0.02까지

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

Forward Process

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

reparameterization trick

μ : $\sqrt{1-\beta_t}\,x_{t-1}$

Σ : β_t

Noise 이전 pixel값

$$\rightarrow \sqrt{\beta_t}\,\varepsilon + \sqrt{1-\beta_t}\,x_{t-1} \qquad \varepsilon \sim N(0,1)$$

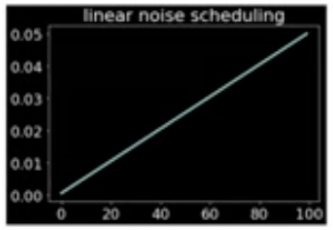
모든 Step의 Variance를 1로 맞춰주기 위함!

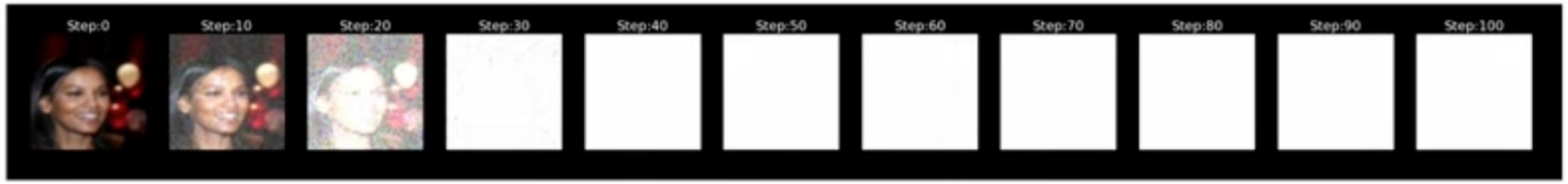
$$Var\left(\sqrt{\beta_t}\,\varepsilon + \sqrt{1-\beta_t}\,x_{t-1}\right) = \beta_t\,Var(\varepsilon) + (1-\beta_t)\,Var(x_{t-1})$$

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

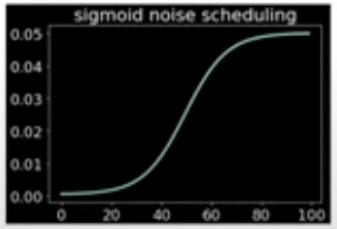
Forward Process

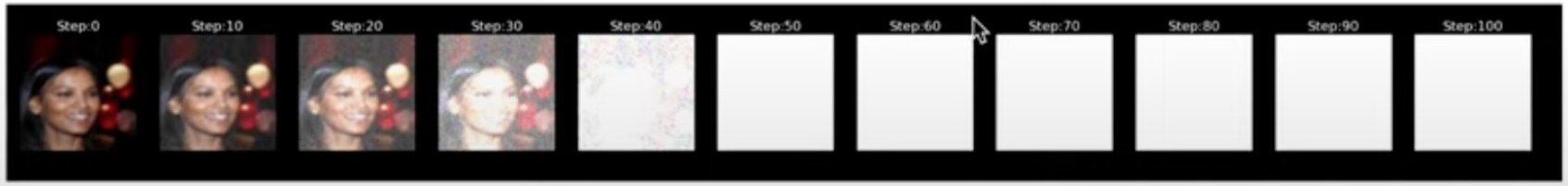
- ✓ Linear scheduling



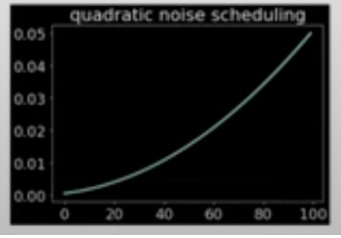


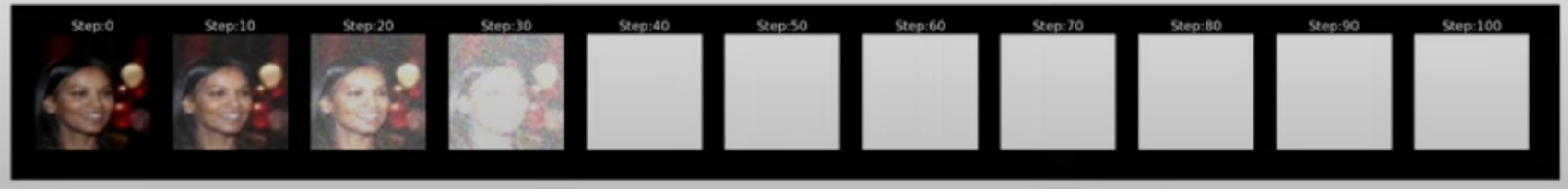
- ✓ Sigmoid scheduling





- ✓ Quadratic scheduling





Forward Process

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

$q(x_{1:T}|x_0) = \frac{q(x_0,x_1,x_2,x_3,\ldots,x_T)}{q(x_0)}$ =?? $\prod_{t=1}^{T} q(x_t|x_{t-1}) = q(x_1|x_0)q(x_2|x_1)q(x_3|x_2)\ldots q(x_T|x_{T-1})$

Marcov chain!!

$q(x_T|x_{T-1}) = q(x_T|x_{T-1}, x_{T-2}, \ldots, x_1, x_0)$

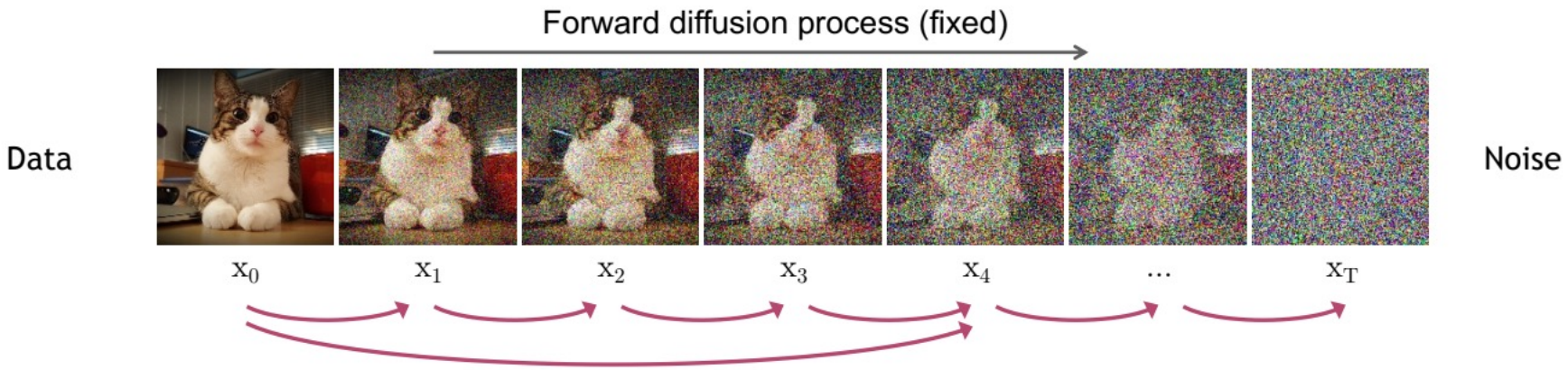
$$q(x_1|x_0)q(x_2|x_1)q(x_3|x_2)\ldots q(x_T|x_{T-1}) = \frac{q(x_1,x_0)}{x_0}\frac{q(x_2,x_1)}{x_1}\ldots\frac{q(x_T,x_{T-1})}{x_{T-1}}$$

$$= \frac{q(x_1,x_0)}{x_0}\frac{q(x_2,x_1,x_0)}{x_1,x_0}\ldots\frac{q(x_T,x_{T-1},\ldots,x_1,x_0)}{x_{T-1},\ldots,x_0}$$

$$= \frac{q(x_0,x_1,x_2,x_3,\ldots,x_T)}{q(x_0)}$$

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

Forward Process



Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$ (Diffusion Kernel)

For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

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

Forward Process

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

Define $\bar{\alpha}_t = \prod_{s=1}(1-\beta_s)$ $\Rightarrow$ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

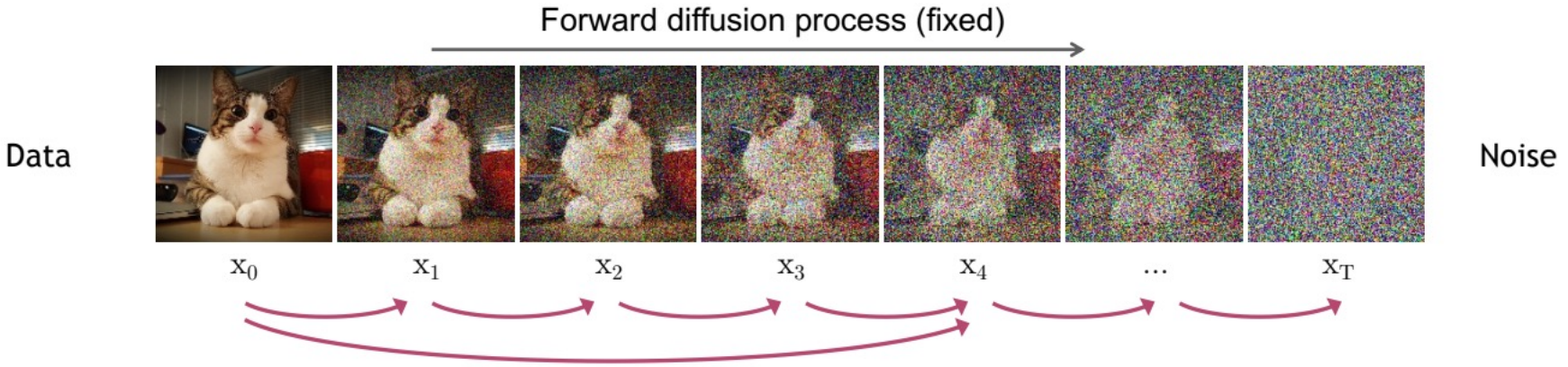
$$q(x_1|x_0) = \sqrt{1-\beta_1}\,x_0$$

$$q(x_2|x_1) = \sqrt{1-\beta_2}\cdot\sqrt{1-\beta_1}\,x_0$$

$$\vdots$$

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

Forward Process



Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

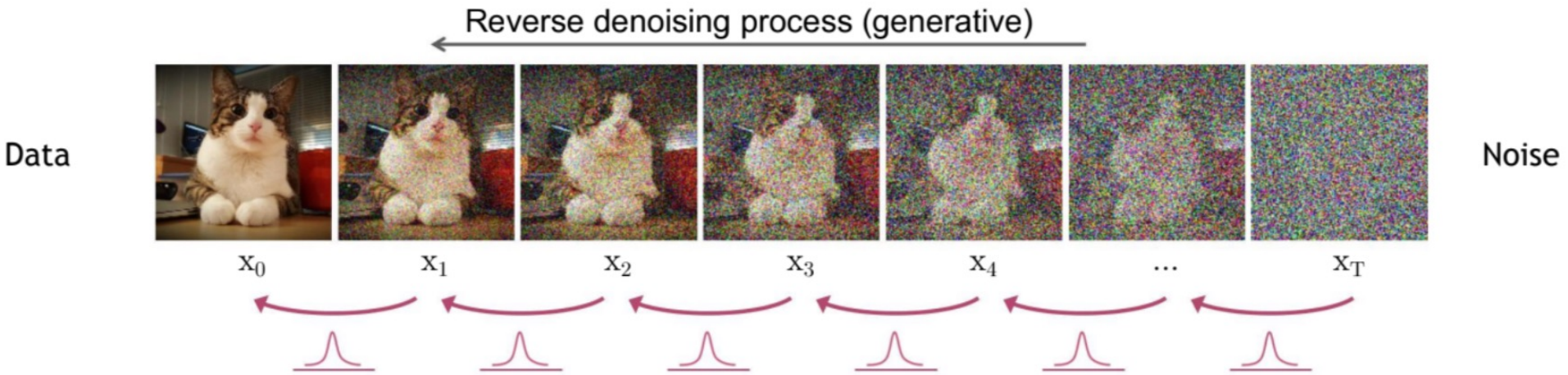
For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

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

Reverse Process

Formal definition of forward and reverse processes in T steps:



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}, \sigma_t^2\mathbf{I})$$

Trainable network
(U-net, Denoising Autoencoder)

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

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

Reverse Process

왜 Network를 사용해서 예측할까?

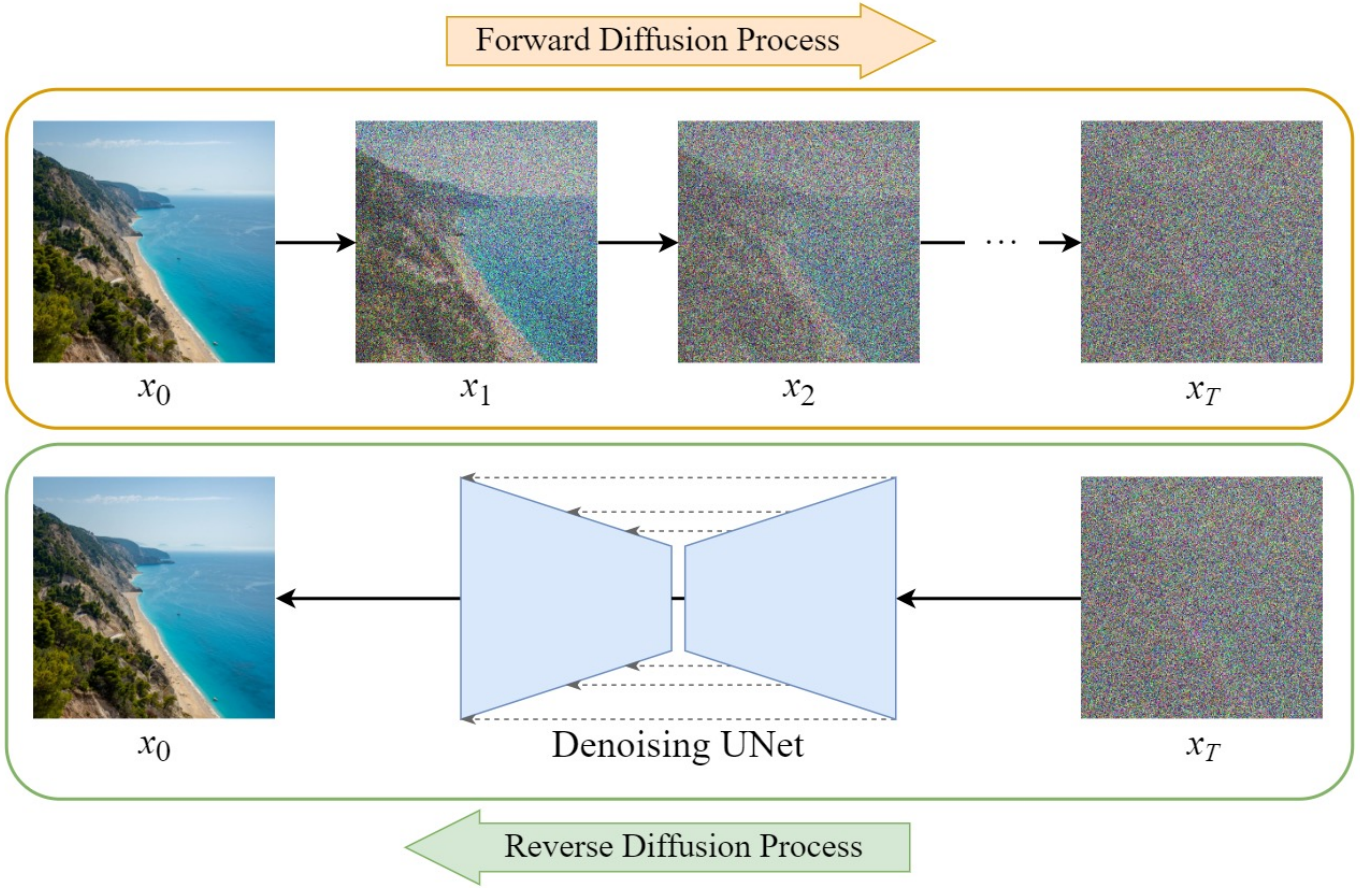
$q(x_{t-1}|x_t)$를 Bayes 법칙을 이용해서 사용할 수 있지 않을까?

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

$q(x_{t-1})$과 $q(x_t)$는 실제 정답의 분포이다.

실제 정답의 분포를 알려면? → 각 step의 Noise낀 Dataset을 모두 봐야한다.
→ Intractable
→ 따라서 $q(x_{t-1}|x_t)$를 $p_\theta(x_{t-1}|x_t)$로 근사하자!

Reverse Process



출처 : https://medium.com/@jaskaranbhatia/summarizing-the-evolution-of-diffusion-models-insights-from-three-research-papers-6889339eba4

Reverse Process

그러면 어떤 Loss를 가지고 Model을 Training 해야할까?

Generative Model의 목표는 Dataset의 분포를 학습하는것

$p_\theta(x_0)$의 Likelihood가 최대화 되야 한다 라는 점에서 시작하자!

Dataset에서 Sampling한 데이터의 Likelihood 값이 최대여야 모델이 잘 학습된 것이다.

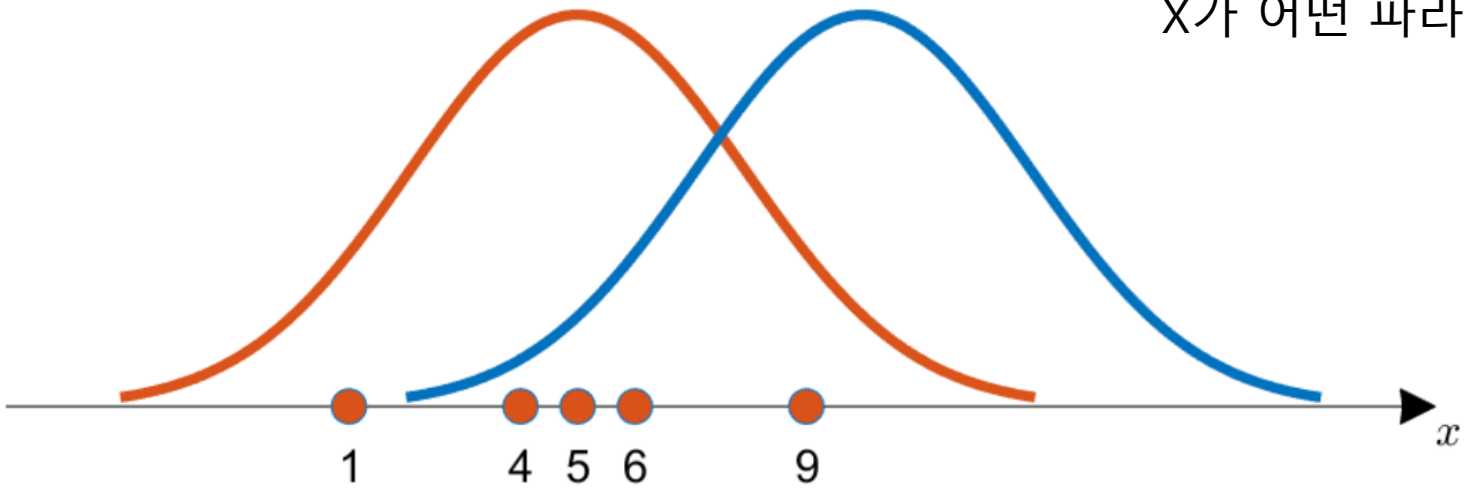
즉, Likelihood가 최대일 때 모델이 Dataset의 분포를 잘 학습했다고 할 수 있다.

Maximum Likelihood Estimation

- 어떤 파라미터 θ로 이루어진 확률밀도함수 P($\boldsymbol{x}$| θ)에서 표본들의 집합을 $\boldsymbol{x}$라고 할 때 θ를 추정하는 방법

$$\mathbf{X} \sim \mathcal{N}(\theta) = \mathcal{N}(m, \boldsymbol{\sigma}\text{^}2)$$

X가 어떤 파라미터에서 나왔을까?



출처: https://angeloyeo.github.io/2020/07/17/MLE.html

Reverse Process

VAE to DDPM

$$\mathbb{E}_{x_T \sim q(x_T|x_0)}[-\log p_\theta(x_0)]$$

① $$= \mathbb{E}_{x_T \sim q(x_T|x_0)}\left[-\log \frac{p_\theta(x_0, x_1, x_2, \dots, x_T)}{p_\theta(x_1, x_2, x_3, \dots, x_T|x_0)}\right] \quad \because \text{bayes rule}, p_\theta(x_T|x_0) = \frac{p_\theta(x_T, x_0)}{p_\theta(x_0)}$$

② $$= \mathbb{E}_{x_T \sim q(x_T|x_0)}\left[-\log \frac{p_\theta(x_0, x_1, x_2, \dots, x_T)}{p_\theta(x_1, x_2, x_3, \dots, x_T|x_0)} \cdot \frac{q(x_{1:T}|x_0)}{q(x_{1:T}|x_0)}\right]$$

③ $$\leq \mathbb{E}_{x_T \sim q(x_T|x_0)}\left[-\log \frac{p_\theta(x_0, x_1, x_2, \dots, x_T)}{q(x_{1:T}|x_0)}\right] \quad \because KL\ divergence > 0, \text{"ELBO"}$$

④ $$= \mathbb{E}_{x_T \sim q(x_T|x_0)}\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] \quad \because Notation$$

⑤ $$= \mathbb{E}_{x_T \sim q(x_T|x_0)}\left[-\log \frac{p_\theta(\boldsymbol{x_T}) \prod_{t=1}^{T} p_\theta(\boldsymbol{x_{t-1}}|\boldsymbol{x_t})}{\prod_{t=1}^{T} q(\boldsymbol{x_t}|\boldsymbol{x_{t-1}})}\right] \quad \because Below\ Markov\ chain\ property$$

⑥ $$= \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)}\left[-\log p_\theta(\boldsymbol{x_T}) - \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] \quad \because separating\ to\ summation\ in\ logarithm$$

$$p_\theta(\boldsymbol{x_{0:T}}) \coloneqq p_\theta(\boldsymbol{x_T}) \prod_{t=1}^{T} p_\theta(\boldsymbol{x_{t-1}}|\boldsymbol{x_t}) \qquad q(\boldsymbol{x_{1:T}}|\boldsymbol{x_0}) \coloneqq \prod_{t=1}^{T} q(\boldsymbol{x_t}|\boldsymbol{x_{t-1}})$$

출처 : https://drive.google.com/file/d/1u8EWfDvaJQGKKC4akQDy50kP-qF_MT09/view

수학 주의...

Diffusion을 사용만 하실 분들은 Pass..

Reverse Process

DDPM loss

$$\mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}[-\log p_\theta(x_0)]$$

$$⑦ \le \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log p_\theta(\boldsymbol{x_T}) - \sum_{t=1}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$$

$$⑧ = \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log p_\theta(\boldsymbol{x_T}) - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - \log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$⑨ = \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log p_\theta(\boldsymbol{x_T}) - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\cdot\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} - \log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] \quad \because *$$

$$⑩ = \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log p_\theta(\boldsymbol{x_T}) - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - \sum_{t=2}^{T}\log\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} - \log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$⑪ = \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log p_\theta(\boldsymbol{x_T}) - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - \log\frac{q(x_1|x_0)}{q(x_T|x_0)} - \log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$⑫ = \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[-\log\frac{p_\theta(\boldsymbol{x_T})}{q(x_T|x_0)} - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - \log p_\theta(x_0|x_1)\right]$$

* $q(x_t|x_{t-1})$
$= q(x_t|x_{t-1},x_0)$ ∵ Markov chain property
$= \frac{q(x_t,x_{t-1},x_0)}{q(x_{t-1},x_0)}$ ∵ bayes rule
$= \frac{q(x_{t-1},x_t,x_0)}{q(x_{t-1},x_0)}\cdot\frac{q(x_t,x_0)}{q(x_t,x_0)}$
$= q(x_{t-1}|x_t,x_0)\cdot\frac{q(x_t,x_0)}{q(x_{t-1},x_0)}$

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

수학 주의...

Diffusion을 사용만 하실 분들은 Pass..

출처 : https://drive.google.com/file/d/1u8EWfDvaJQGKKC4akQDy50kP-qF_MT09/view

Reverse Process

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

Lt와 L0는 실제로 Loss를 구할 때는 사용되지 않는다!

Why?

Reverse Process

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

직관적으로,

P(xt)는 Xt의 분포. Xt는 N(0,1)의 가우시안 분포이다.
q(xt|x0)는 x0로부터 Noise를 더하는 과정을 거쳐 생성된 xt의 분포이다.

결국 두 수식 모두 N(0,1)의 가우시안 분포일 것이다. 따라서 둘의 차이는 같거나 굉장히 작기 때문에 굳이 Loss에 포함 할 필요가 없다.

Reverse Process

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

수식적으로는..

실제로 논문의 내용을 확인하면 $beta_1 = 10^{-4}$에서 $beta_T = 0.02$까지 $T = 1000$으로 linear하게 증가한다고 하며, 이 정보에 따라 $q(x_T|x_0)$를 계산해보면

$$q(x_T|x_0) = \mathcal{N}(x_T; 0.00635x_0, 0.99995I)$$

로 거의

$$\mathcal{N}(0, I)$$

에 가깝습니다. 즉 noise latent x_T를 얻을때 VAE와 같이 따로 어떤 분포를 따라갈 필요가 없이 **x_T의 sampling은 Gaussian distribution에서 sampling**하게 됩니다.

출처 : https://developers-shack.tistory.com/9

Reverse Process

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

직관적으로,

$logp_\theta(x_0|x_1)$은 약간의 Noise가 더해진 Image x1이 주어졌을 때 x0을 reconstruction하는 loss term

그런데 x1은 x0에 비해 엄청나게 미세한 Noise가 들어갔기 때문에 굉장히 쉬운 task이다.

따라서 해당 Loss는 엄청나게 작을 것이고 굳이 Loss에 포함시킬 필요 없다.

Reverse Process

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

수식적으로는..

$$p_\theta(x_0|x_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(x_1, 1), \sigma_1^2) dx$$

$$\delta_+(x) = \begin{cases} \infty & if\ x = 1 \\ x + \frac{1}{255} & if\ x < 1 \end{cases}, \qquad \delta_-(x) = \begin{cases} -\infty & if\ x = -1 \\ x - \frac{1}{255} & if\ x > -1 \end{cases}$$

출처 : https://developers-shack.tistory.com/9

Reverse Process

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

따라서 최종 Loss는 Lt-1만 쓰이게된다!

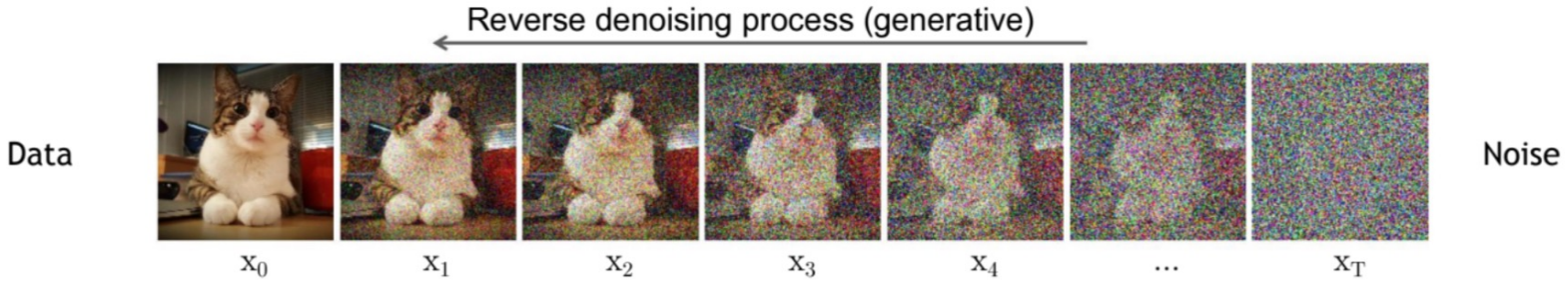
Reverse Process

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

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

직관적으로,

어떠한 step x_t에서 x_{t-1}을 만들어 낼 때, 사실 어떠한 Image x_0는 이미 주어졌다고 볼 수 있다.



Reverse Process

수식주의

$q(x_{t-1}|x_t, x_0)$을 그렇다면 전개해보자

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

$$q(x_t|x_{t-1}) = \frac{1}{\sqrt{2\pi\beta_t}} \exp\left(-\frac{(x_t - \sqrt{1-\beta_t}x_{t-1})^2}{2\beta_t}\right)$$

$$q(x_t|x_0) = \frac{1}{\sqrt{2\pi(1-\bar{\alpha}_t)}} \exp\left(-\frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{2(1-\bar{\alpha}_t)}\right)$$

$$q(x_{t-1}|x_0) = \frac{1}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})}} \exp\left(-\frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{2(1-\bar{\alpha}_{t-1})}\right)$$

Note

$$P(B \mid A) = P(A \mid B) \frac{P(B)}{P(A)}$$

$$N(x;\ \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- $q(x_t \mid x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t * I)$
- $q(x_t \mid x_0) = N(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t) * I)$
  - $\alpha_t = 1 - \beta_t$
  - $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

출처 : https://www.youtube.com/watch?v=jaPPALsUZo8

Reverse Process

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

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

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

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

b a

- $q(X_{t-1} \mid X_t) \rightarrow q(X_{t-1} \mid X_t, X_0)$
  - $N\left(X_{t-1};\ \tilde{\mu}(X_t, X_0), \tilde{\Sigma}(X_t, X_0)\right)$
  - $N\left(X_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}}x_t\ ,\ \beta_t\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)$

출처 : https://www.youtube.com/watch?v=jaPPALsUZo8

Reverse Process

- $q(X_{t-1} \mid X_t) \rightarrow q(X_{t-1} \mid X_t, X_0)$
  - $N\left(X_{t-1};\ \tilde{\mu}(X_t, X_0), \tilde{\Sigma}(X_t, X_0)\right)$
  - $N\left(X_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}} x_t\ , \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)$

위 식을 알 수 있으니, 이제는 Model없이 위의 공식을 이용해서 Reverse Process를 진행하면 되지 않을까?

→ Gaussian Noise x_T에서 출발 하여 x_0를 Generate 해야하는데, 우리는 Noise만 있을 때 부터 x_0가 무엇인지 알지 못한다. 따라서 식을 변형시켜야 한다.

출처 : https://www.youtube.com/watch?v=jaPPALsUZo8

Reverse Process

- $q(X_{t-1} \mid X_t) \rightarrow q(X_{t-1} \mid X_t, X_0)$
  - $N\left(X_{t-1};\ \tilde{\mu}(X_t, X_0), \tilde{\Sigma}(X_t, X_0)\right)$
  - $N\left(X_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}} x_t\ ,\ \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)$

위 식을 알 수 있으니, 이제는 Model없이 위의 공식을 이용해서 Reverse Process를 진행하면 되지 않을까?

→ Gaussian Noise x_T에서 출발 하여 x_0를 Generate 해야하는데, 우리는 Noise만 있을 때 부터 x_0가 무엇인지 알지 못한다. 따라서 식을 변형시켜야 한다.

Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

출처 : https://www.youtube.com/watch?v=jaPPALsUZo8

Reverse Process

- $q(X_{t-1} \mid X_t) \rightarrow q(X_{t-1} \mid X_t, X_0)$
  - $N\left(X_{t-1};\ \tilde{\mu}(X_t, X_0), \tilde{\Sigma}(X_t, X_0)\right)$
  - $N\left(X_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}} x_t\ , \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)$
  - $N\left(X_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right), \tilde{\beta}_t\right)$

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

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

$$= \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$$

ϵ는 N(0,1)에서 Sampling한 Noise이다. 그러면 Model이 해야 할 일은?

출처 : https://www.youtube.com/watch?v=jaPPALsUZo8

Reverse Process

ϵ는 N(0,1)에서 Sampling한 Noise이다. 그러면 Model이 해야 할 일은?

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

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

$$= \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$$

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right)$$

Mu의 나머지 항들은 모두 상수. 따라서 N(0,1)에서 실제로 어떤 Noise가 Sampling됐는지 맞추면 된다!

출처 : https://www.youtube.com/watch?v=jaPPALsUZo8

Reverse Process

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}$$

이론적으로, 두 Gaussian의 KL Divergence는 $KL(p,q) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$

$$L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} ||\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)||^2 \right] + C$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right)$$

They propose to represent the mean of the denoising model using a *noise-prediction* network:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right)$$

With this parameterization

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

출처 :
https://drive.google.com/file/d/1u8EWfDvaJQGKKC4akQDy50kP-qF_MT09/view

Reverse Process

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$$

They propose to represent the mean of the denoising model using a *noise-prediction* network:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{x}_t, t)\right)$$

With this parameterization

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

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

논문에서는 앞의 계수 λ_t를 1로 놓는다. Why? 학습을 위해서

Reverse Process

최종 Loss

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

Algorithm 1 Training

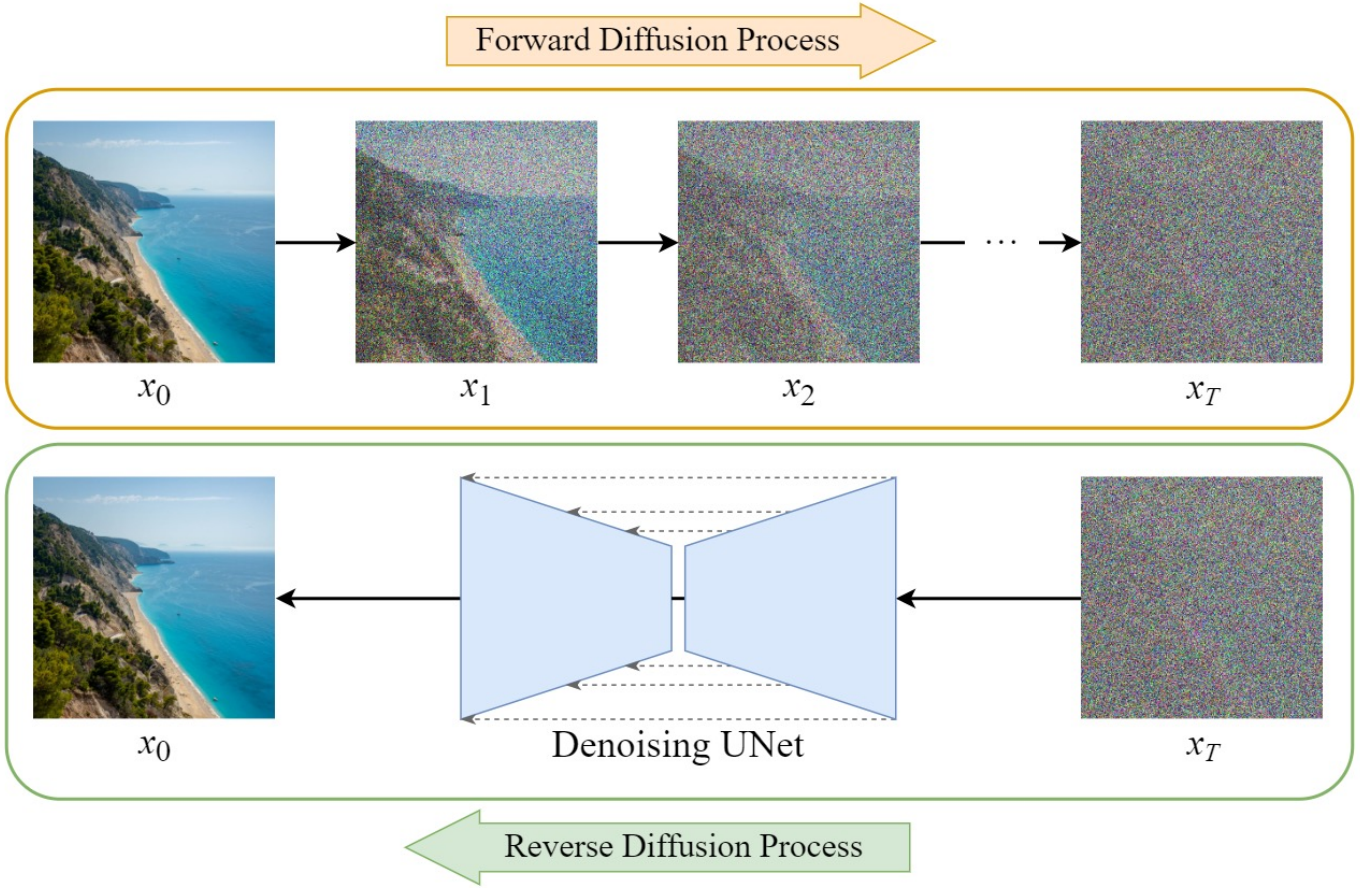
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0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

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$$

Reverse Process



출처 : https://medium.com/@jaskaranbhatia/summarizing-the-evolution-of-diffusion-models-insights-from-three-research-papers-6889339eba4

Summary

Summary

Denoising Diffusion Probabilistic Models

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I})$$

Forward diffusion process (fixed)

Data

Noise

Reverse denoising process (generative)

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right)$$

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