Deep Generative Models

Lecture 9

Roman Isachenko

Moscow Institute of Physics and Technology
Yandex School of Data Analysis

2024, Autumn

Recap of previous lecture

Langevin dynamic

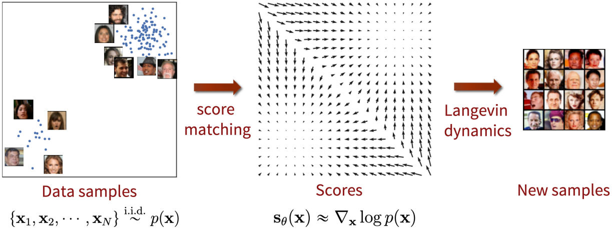
$$\mathbf{x}_{l+1} = \mathbf{x}_l + \frac{\eta}{2} \cdot \nabla_{\mathbf{x}_l} \log p(\mathbf{x}_l|\boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_l, \quad \boldsymbol{\epsilon}_l \sim \mathcal{N}(0, \mathbf{I}).$$

Fisher divergence

$$D_F(\pi, p) = \frac{1}{2}\mathbb{E}_\pi \left\| \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

Score function

$$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$$



Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021

Recap of previous lecture

Let perturb original data by normal noise $q(\mathbf{x}_\sigma | \mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \cdot \mathbf{I})$

$$q(\mathbf{x}_\sigma) = \int q(\mathbf{x}_\sigma | \mathbf{x}) \pi(\mathbf{x}) d\mathbf{x}.$$

Then the solution of

$$\frac{1}{2} \mathbb{E}_{q(\mathbf{x}_\sigma)} \bigl\| \mathbf{s}_{\boldsymbol{\theta}, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \bigr\|_2^2 \rightarrow \min_{\boldsymbol{\theta}}$$

satisfies $\mathbf{s}_{\boldsymbol{\theta}, \sigma}(\mathbf{x}_\sigma) \approx \mathbf{s}_{\boldsymbol{\theta}, 0}(\mathbf{x}_0) = \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$ if σ is small enough.

Theorem (denoising score matching)

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} \bigl\| \mathbf{s}_{\boldsymbol{\theta}, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \bigr\|_2^2 =$$
$$= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma | \mathbf{x})} \bigl\| \mathbf{s}_{\boldsymbol{\theta}, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma | \mathbf{x}) \bigr\|_2^2 + \text{const}(\boldsymbol{\theta})$$

Here $\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma | \mathbf{x}) = -\frac{\mathbf{x}_\sigma - \mathbf{x}}{\sigma^2} = -\frac{\boldsymbol{\epsilon}}{\sigma}$.

- We do not need to compute $\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)$ at the RHS.
- $\mathbf{s}_{\boldsymbol{\theta}, \sigma}(\mathbf{x}_\sigma)$ tries to **denoise** a corrupted sample.

Vincent P. A Connection Between Score Matching and Denoising Autoencoders, 2010

Recap of previous lecture

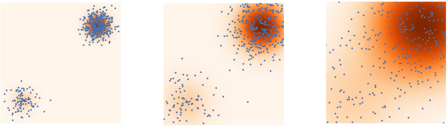
Noise conditioned score network

- Define the sequence of noise levels: $\sigma_1 < \sigma_2 < \dots < \sigma_T$.
- Train denoised score function $\mathbf{s}_{\boldsymbol{\theta}, \sigma_t}(\mathbf{x}_t)$ for each noise level:

$$\sum_{t=1}^{T} \sigma_t^2 \cdot \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \big\| \mathbf{s}_{\boldsymbol{\theta}, \sigma_t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}) \big\|_2^2 \to \min_{\boldsymbol{\theta}}$$

- Sample from **annealed** Langevin dynamics (for $t = 1, \dots, T$).

$$\sigma_1 \quad < \quad \sigma_2 \quad < \quad \sigma_3$$



Song Y. et al. Generative Modeling by Estimating Gradients of the Data Distribution, 2019

Recap of previous lecture

NCSN training

1. Get the sample $\mathbf{x}_0 \sim \pi(\mathbf{x})$.
2. Sample noise level $t \sim U\{1, T\}$ and the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.
3. Get noisy image $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \cdot \boldsymbol{\epsilon}$.
4. Compute loss $\mathcal{L} = \sigma_t^2 \cdot \|\mathbf{s}_{\boldsymbol{\theta}, \sigma_t}(\mathbf{x}_t) + \frac{\boldsymbol{\epsilon}}{\sigma_t}\|^2$.

NCSN sampling (annealed Langevin dynamics)

- Sample $\mathbf{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \cdot \mathbf{I}) \approx q(\mathbf{x}_T)$.
- Apply L steps of Langevin dynamic

$$\mathbf{x}_l = \mathbf{x}_{l-1} + \frac{\eta_t}{2} \cdot \mathbf{s}_{\boldsymbol{\theta}, \sigma_t}(\mathbf{x}_{l-1}) + \sqrt{\eta_t} \cdot \boldsymbol{\epsilon}_l.$$

- Update $\mathbf{x}_0 := \mathbf{x}_L$ and choose the next σ_t.

Song Y. et al. Improved Techniques for Training Score-Based Generative Models, 2020

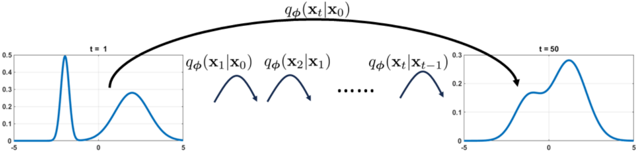
Recap of previous lecture

Forward gaussian diffusion process

Let $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$, $\beta_t \ll 1$, $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

$$\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I});$$
$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

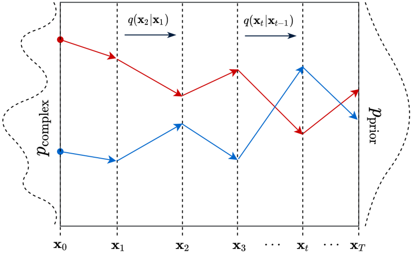
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1}, \beta_t \cdot \mathbf{I});$$
$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1-\bar{\alpha}_t) \cdot \mathbf{I}).$$



Chan S. Tutorial on Diffusion Models for Imaging and Vision, 2024

Recap of previous lecture

Diffusion refers to the flow of particles from high-density regions towards low-density regions.



1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$, where $T \gg 1$.

If we are able to invert this process, we will get the way to sample $\mathbf{x} \sim \pi(\mathbf{x})$ using noise samples $p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$.
Now our goal is to revert this process.

Das A. An introduction to Diffusion Probabilistic Models, blog post, 2021

Outline

Outline

Denoising score matching

NCSN

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2 \cdot \mathbf{I}), \quad q(\mathbf{x}_1) \approx \pi(\mathbf{x}), \quad q(\mathbf{x}_T) \approx \mathcal{N}(0, \sigma_T^2 \cdot \mathbf{I}).$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}) = -\frac{\mathbf{x}_t - \mathbf{x}}{\sigma_t^2}$$

Gaussian diffusion

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1 - \bar{\alpha}_t) \cdot \mathbf{I}), \quad q(\mathbf{x}_1) \approx \pi(\mathbf{x}), \quad q(\mathbf{x}_T) \approx \mathcal{N}(0, \mathbf{I})$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0}{1 - \bar{\alpha}_t}$$

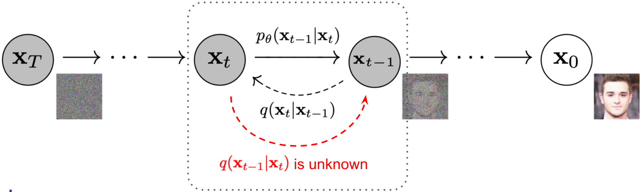
Theorem (denoising score matching)

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_t)} \big\| \mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) \big\|_2^2 = \\
&\qquad = \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \big\| \mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}) \big\|_2^2 + \text{const}(\boldsymbol{\theta})
\end{aligned}$$

Note: We are able to apply NCSN approach with annealed Langevin dynamics to get diffusion denoising model.

Outline

Reverse gaussian diffusion process



Forward process

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}\cdot\mathbf{x}_{t-1}, \beta_t\cdot\mathbf{I}\right).$$

Reverse process

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)} \approx p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})$$

$q(\mathbf{x}_{t-1})$, $q(\mathbf{x}_t)$ are intractable:

$$q(\mathbf{x}_t) = \int q(\mathbf{x}_t|\mathbf{x}_0)\pi(\mathbf{x}_0)d\mathbf{x}_0$$

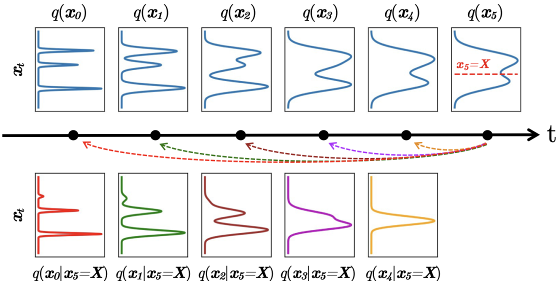
Weng L. What are Diffusion Models?, blog post, 2021

Reverse gaussian diffusion process

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)}$$

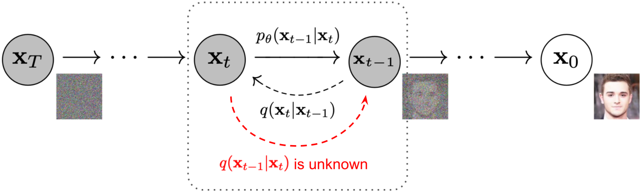
Theorem (Feller, 1949)

If β_t is small enough, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ will be Gaussian (that is why diffusion needs $T \approx 1000$ steps to converge).



Feller W. On the theory of stochastic processes, with particular reference to applications, 1949
Xiao Z., Kreis K., Vahdat A. Tackling the generative learning trilemma with denoising diffusion GANs, 2021

Reverse gaussian diffusion process



Let define the reverse process:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t), \boldsymbol{\sigma}^2_{\boldsymbol{\theta},t}(\mathbf{x}_t)\right)$$

Feller theorem shows that it is a reasonable assumption.

Forward process

1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$.

Reverse process

1. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$;
2. $\mathbf{x}_{t-1} = \boldsymbol{\sigma}_{\boldsymbol{\theta},t}(\mathbf{x}_t) \cdot \boldsymbol{\epsilon} + \boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t)$;
3. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$.

Note: The forward process does not have any learnable parameters!

Weng L. What are Diffusion Models?, blog post, 2021

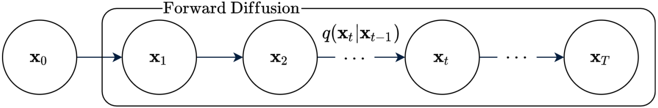
Outline

Gaussian diffusion model as VAE

Let treat $\mathbf{z} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ as a latent variable (**note:** each $\mathbf{x}_t$ has the same size) and $\mathbf{x} = \mathbf{x}_0$ as observed samples.

Latent Variable Model

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})$$



Forward diffusion

- Variational posterior distribution (encoder)

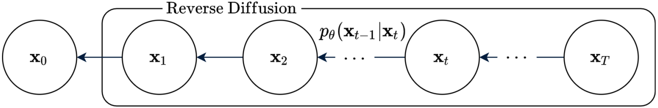
$$q(\mathbf{z}|\mathbf{x}) = q(\mathbf{x}_1, \dots, \mathbf{x}_T|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

- **Note:** there is no learnable parameters.

Das A. An introduction to Diffusion Probabilistic Models, blog post, 2021

Gaussian diffusion model as VAE

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})$$



Reverse diffusion

- Generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}).$$

- Prior distribution

$$p(\mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}_1, \dots, \mathbf{x}_T|\boldsymbol{\theta}) = \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot p(\mathbf{x}_T).$$

Note: this differs from the vanilla VAE with the complex decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ and the standard normal prior $p(\mathbf{z})$.

Das A. An introduction to Diffusion Probabilistic Models, blog post, 2021

Conditioned reverse distribution

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)}$$

$$\begin{aligned} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\ &= \frac{\mathcal{N}(\sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \cdot \mathcal{N}(\sqrt{\bar{\alpha}_{t-1}} \cdot \mathbf{x}_0, (1-\bar{\alpha}_{t-1}) \cdot \mathbf{I})}{\mathcal{N}(\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1-\bar{\alpha}_t) \cdot \mathbf{I})} \\ &= \mathcal{N}(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \cdot \mathbf{I}) \end{aligned}$$

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \cdot \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t} \cdot \mathbf{x}_0;$$

$$\tilde{\beta}_t = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} = \text{const}.$$

$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ defines how to denoise a noisy image $\mathbf{x}_t$ with access to what the final, completely denoised image $\mathbf{x}_0$ should be.

Ho J. Denoising Diffusion Probabilistic Models, 2020

ELBO for gaussian diffusion model

Standard ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x})} = \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) \rightarrow \max_{q,\boldsymbol{\theta}}$$

Derivation

$$\begin{aligned}
\mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_0, \mathbf{x}_{1:T}|\boldsymbol{\theta})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) \prod_{t=1}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) \prod_{t=1}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}
\end{aligned}$$

We add conditioning on $\mathbf{x}_0$ to make reverse distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ tractable and to get KL divergences.

Ho J. Denoising Diffusion Probabilistic Models, 2020

ELBO for gaussian diffusion model

Derivation (continued)

$$\mathcal{L}_{\phi,\theta}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \theta) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta)}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} =$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \theta) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta)}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} =$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \theta) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta)}{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} =$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log p(\mathbf{x}_0|\mathbf{x}_1, \theta) + \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \sum_{t=2}^{T} \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right] =$$

$$= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \theta) + \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} +$$

$$+ \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{x}_0)} \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right)$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

ELBO for gaussian diffusion model

$$\mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) + \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} +$$

$$+ \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{x}_0)} \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) =$$

$$= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) - KL\big(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)\big) -$$

$$- \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} KL\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})\big)}_{\mathcal{L}_t}$$

- First term is a decoder distribution

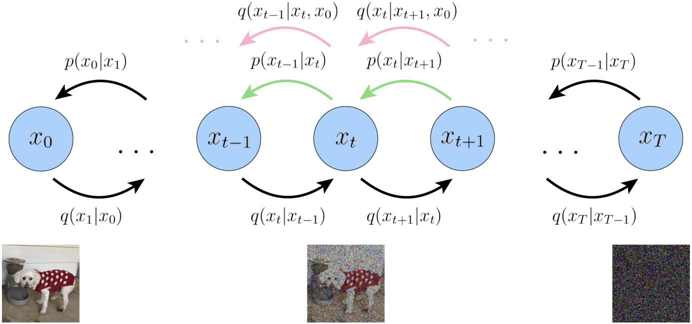
$$\log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) = \log \mathcal{N}\big(\mathbf{x}_0|\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_1), \boldsymbol{\sigma}^2_{\boldsymbol{\theta},t}(\mathbf{x}_1)\big).$$

- Second term is constant ($p(\mathbf{x}_T)$ is a standard Normal, $q(\mathbf{x}_T|\mathbf{x}_0)$ is a non-parametrical Normal).

Ho J. Denoising Diffusion Probabilistic Models, 2020

ELBO for gaussian diffusion model

$$\mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) - KL\big(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)\big) - \\ - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} KL\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})\big)}_{\mathcal{L}_t}$$



Luo C. Understanding Diffusion Models: A Unified Perspective, 2022

Outline

1. Denoising score matching for diffusion
2. Reverse gaussian diffusion process
3. Gaussian diffusion model as VAE
4. Reparametrization of gaussian diffusion model

Reparametrization of DDPM

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} KL\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})\big)$$

$\mathcal{L}_t$ is the mean of KL between two normal distributions:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}),$$
$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{x}_{t-1}|\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t), \boldsymbol{\sigma}^2_{\boldsymbol{\theta},t}(\mathbf{x}_t)\big)$$

Here $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)$, $\tilde{\beta}_t = \frac{\beta_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$ have analytical expressions. Let assume

$$\boldsymbol{\sigma}^2_{\boldsymbol{\theta},t}(\mathbf{x}_t) = \tilde{\beta}_t \mathbf{I} \quad \Rightarrow \quad p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{x}_{t-1}|\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t), \tilde{\beta}_t \mathbf{I}\big).$$

Theoretically optimal $\boldsymbol{\sigma}^2_{\boldsymbol{\theta},t}(\mathbf{x}_t)$ lies in the range $[\tilde{\beta}_t, \beta_t]$:

- β_t is optimal for $\mathbf{x}_0 \sim \mathcal{N}(0, \mathbf{I})$;
- $\tilde{\beta}_t$ is optimal for $\mathbf{x}_0 \sim \delta(\mathbf{x}_0 - \mathbf{x}^*)$.

Ho J. Denoising Diffusion Probabilistic Models, 2020

Reparametrization of DDPM

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I});$$
$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_{t-1}|\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t), \tilde{\beta}_t \mathbf{I}).$$

Use the formula for KL between two normal distributions:

$$\begin{aligned}\mathcal{L}_t &= \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} KL\Big(\mathcal{N}\big(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}\big)||\mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t), \tilde{\beta}_t \mathbf{I}\big)\Big) \\ &= \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\frac{1}{2\tilde{\beta}_t}\left\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t)\right\|^2\right]\end{aligned}$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon} \quad \Rightarrow \quad \mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}}$$

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \cdot \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t} \cdot \mathbf{x}_0 \\ &= \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \cdot \boldsymbol{\epsilon}\end{aligned}$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

Reparametrization of DDPM

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[\frac{1}{2\tilde{\beta}_t}\left\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t)\right\|^2\right]$$

Reparametrization

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \cdot \boldsymbol{\epsilon}$$

$$\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta},t}(\mathbf{x}_t)$$

$$\begin{aligned}\mathcal{L}_t &= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta},t}(\mathbf{x}_t)\right\|^2\right] \\ &= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta},t}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right)\right\|^2\right]\end{aligned}$$

At each step of reverse diffusion process we try to predict the noise $\boldsymbol{\epsilon}$ that we used in the forward diffusion process!

Ho J. Denoising Diffusion Probabilistic Models, 2020

Reparametrization of DDPM

$$\mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) - KL\big(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)\big) - \\ - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} KL\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})\big)}_{\mathcal{L}_t}$$

$$\mathcal{L}_t = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})} \left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t \alpha_t (1-\bar{\alpha}_t)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta},t}\big(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\big) \right\|^2 \right]$$

Simplified objective

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t\sim U\{2,T\}} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta},t}\big(\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon}\big) \right\|^2$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

Summary

- Denoising score matching is applicable to gaussian diffusion process.
- Reverse process allows to sample from the real distribution $\pi(\mathbf{x})$ using samples from noise, but it is intractable.
- We will use approximation to get the reverse process.
- Diffusion model is a VAE model which reverts gaussian diffusion process using variational inference.
- ELBO of DDPM could be represented as a sum of KL terms.
- DDPM is a VAE model with hierarchical latent variables.
- At each step DDPM predicts the noise that was used in the forward diffusion process.