Deep Generative Models

Lecture 8

Roman Isachenko

AI Masters

2025, Spring

Recap of previous lecture

Frechet Inception Distance (FID)

In case of Normal distributions $\pi(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\pi, \boldsymbol{\Sigma}_\pi)$, $p(\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$:

$$\begin{aligned} \mathrm{FID}(\pi, p) = W_2^2(\pi, p) &= \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\|^2 \\ &= \|\boldsymbol{\mu}_\pi - \boldsymbol{\mu}_p\|^2 + \mathrm{tr}\left[\boldsymbol{\Sigma}_\pi + \boldsymbol{\Sigma}_p - 2\left(\boldsymbol{\Sigma}_\pi^{1/2}\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_\pi^{1/2}\right)^{1/2}\right] \end{aligned}$$

- Needs a large sample size for evaluation.
- Calculation of FID is slow.
- High dependence on the pretrained classification model.
- Uses the normality assumption!

Heusel M. et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, 2017

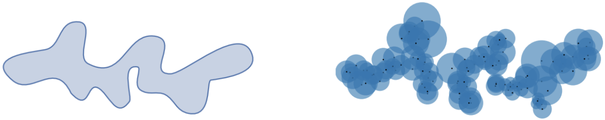
Recap of previous lecture

- $\mathcal{S}_\pi = \{\mathbf{x}_i\}_{i=1}^n \sim \pi(\mathbf{x})$ – real samples;
- $\mathcal{S}_p = \{\mathbf{x}_i\}_{i=1}^n \sim p(\mathbf{x}|\boldsymbol{\theta})$ – generated samples.

Define binary function:

$$\mathbb{I}(\mathbf{x}, \mathcal{S}) = \begin{cases} 1, & \text{if exists } \mathbf{x}' \in \mathcal{S} : \|\mathbf{x} - \mathbf{x}'\|_2 \leq \|\mathbf{x}' - \text{NN}_k(\mathbf{x}', \mathcal{S})\|_2; \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{Precision}(\mathcal{S}_\pi, \mathcal{S}_p) = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{S}_p} \mathbb{I}(\mathbf{x}, \mathcal{S}_\pi); \;\; \text{Recall}(\mathcal{S}_\pi, \mathcal{S}_p) = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{S}_\pi} \mathbb{I}(\mathbf{x}, \mathcal{S}_p).$$



(a) True manifold (b) Approx. manifold

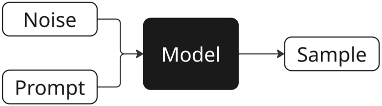
Embed the samples using the pretrained network (as for FID).

Kynkäänniemi T. et al. Improved precision and recall metric for assessing generative models, 2019

Recap of previous lecture

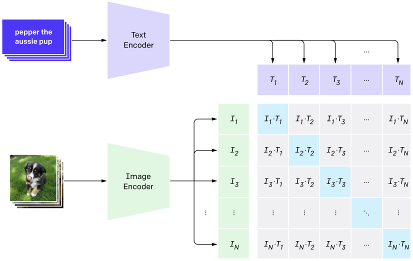
Unconditional model

Conditional model



We need the way to measure not only generated image quality, but also its relevance to the prompt.

CLIP score



Radford A. et al. Learning transferable visual models from natural language supervision, 2021

Recap of previous lecture

- There is no perfect automated metric.
- The best way to evaluate the generative model is to make human evaluation.
- It is essential to evaluate different aspects.

Human Evaluation

Аспект	Yandex ART 2.0	Mj 6.1	Mj 6	Ideogram	Recraft	Google Imagen3	Dall-E 3	FLUX	SBER Kandi3.1
Релевантность	0,59	0,58	0,63	**0,45**	**0,51**	**0,50**	**0,50**	**0,54**	0,75
Эстетика	**0,49**	0,55	0,55	**0,51**	**0,51**	0,61	0,61	**0,54**	0,59
Комплексность	**0,44**	0,73	0,70	0,68	0,76	0,75	0,75	0,71	0,74
Дефектность	0,69	0,57	0,68	0,55	0,59	0,63	0,63	**0,50**	0,75
Предпочтение	0,66	0,60	0,69	**0,49**	**0,54**	0,63	0,63	**0,51**	0,84

YandexART 2.5, 2025

Recap of previous lecture

Langevin dynamic

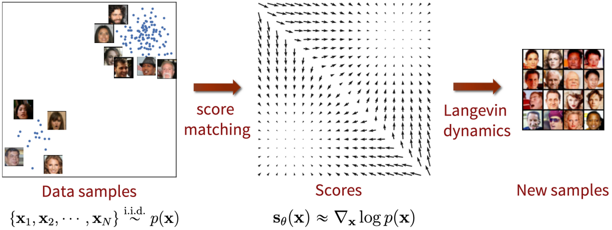
$$\mathbf{x}_{l+1} = \mathbf{x}_l + \frac{\eta}{2} \cdot \nabla_{\mathbf{x}_l} \log p(\mathbf{x}_l|\boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_l, \quad \boldsymbol{\epsilon}_l \sim \mathcal{N}(0, \mathbf{I}).$$

Fisher divergence

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

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

Outline

1. Score matching
   - Denoising score matching
   - Noise Conditioned Score Network (NCSN)
2. Forward gaussian diffusion process
3. Denoising score matching for diffusion
4. Reverse Gaussian diffusion process

Outline

1. Score matching
   - Denoising score matching
   - Noise Conditioned Score Network (NCSN)
2. Forward gaussian diffusion process
3. Denoising score matching for diffusion
4. Reverse Gaussian diffusion process

Outline

Denoising score matching

Let perturb original data $\mathbf{x} \sim \pi(\mathbf{x})$ by random normal noise

$$\mathbf{x}_\sigma = \mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad q(\mathbf{x}_\sigma | \mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \cdot \mathbf{I})$$

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

Assumption

The solution of

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

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.

- The score function of the noised data is almost the same as the score function of the original data.
- Score function $\mathbf{s}_{\boldsymbol{\theta}, \sigma}(\mathbf{x}_\sigma)$ parametrized by σ.
- **Note:** We don't know $q(\mathbf{x}_\sigma)$, just like $\pi(\mathbf{x})$.

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

Denoising score matching

Theorem

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

Gradient of the noise kernel

$$\mathbf{x}_\sigma = \mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}, \quad q(\mathbf{x}_\sigma|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \cdot \mathbf{I})$$

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

- The RHS does not need to compute $\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)$ and even $\nabla_{\mathbf{x}_\sigma} \log \pi(\mathbf{x}_\sigma)$.
- $\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)$ tries to **denoise** the noised samples $\mathbf{x}_\sigma$.

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

Denoising score matching

Theorem

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} \underbrace{\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\right\|_2^2}_{h(\mathbf{x}_\sigma)} =$$

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

Proof

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} h(\mathbf{x}_\sigma) = \int q(\mathbf{x}_\sigma) h(\mathbf{x}_\sigma) d\mathbf{x}_\sigma =$$

$$= \int \left(\int q(\mathbf{x}_\sigma|\mathbf{x})\pi(\mathbf{x})d\mathbf{x}\right) h(\mathbf{x}_\sigma) d\mathbf{x}_\sigma = \mathbb{E}_{\pi(\mathbf{x})}\mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} h(\mathbf{x}_\sigma)$$

$$\mathbb{E}_{q(\mathbf{x}_\sigma)}\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\right\|_2^2 =$$

$$= \mathbb{E}_{q(\mathbf{x}_\sigma)}\left[\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)\|^2 + \underbrace{\|\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\|_2^2}_{\text{const}(\boldsymbol{\theta})} - 2\mathbf{s}_{\boldsymbol{\theta},\sigma}^T(\mathbf{x}_\sigma)\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\right]$$

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

Denoising score matching

Theorem

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

Proof (continued)

$$\mathbb{E}_{q(\mathbf{x}_\sigma)}\left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\right] = \int q(\mathbf{x}_\sigma)\left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)\frac{\nabla_{\mathbf{x}_\sigma} q(\mathbf{x}_\sigma)}{q(\mathbf{x}_\sigma)}\right] d\mathbf{x}_\sigma =$$
$$= \int \left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)\nabla_{\mathbf{x}_\sigma}\left(\int q(\mathbf{x}_\sigma|\mathbf{x})\pi(\mathbf{x}) d\mathbf{x}\right)\right] d\mathbf{x}_\sigma =$$
$$= \int\int \pi(\mathbf{x})\left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)\nabla_{\mathbf{x}_\sigma} q(\mathbf{x}_\sigma|\mathbf{x})\right] d\mathbf{x}_\sigma d\mathbf{x} =$$
$$= \int\int \pi(\mathbf{x})q(\mathbf{x}_\sigma|\mathbf{x})\left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x})\right] d\mathbf{x}_\sigma d\mathbf{x} =$$
$$= \mathbb{E}_{\pi(\mathbf{x})}\mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})}\left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x})\right]$$

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

Denoising score matching

Theorem

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} \underbrace{\left\| \mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right\|_2^2}_{h(\mathbf{x}_\sigma)} =$$

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

Proof (continued)

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} h(\mathbf{x}_\sigma) = \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} h(\mathbf{x}_\sigma) d\mathbf{x}_\sigma$$

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} \left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right] = \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} \left[\mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x}) \right]$$

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} \left\| \mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right\|_2^2 =$$

$$= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} \left[\| \mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) \|^2 - 2 \mathbf{s}^T_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x}) \right] + \text{const}(\boldsymbol{\theta})$$

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

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

Denoising score matching

Initial objective:

$$\mathbb{E}_{\pi(\mathbf{x})}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x})\right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

Noised objective:

$$\mathbb{E}_{q(\mathbf{x}_\sigma)}\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}} \log q(\mathbf{x}_\sigma)\right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

This is equivalent to denoising task

$$\mathbb{E}_{\pi(\mathbf{x})}\mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})}\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x})\right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

$$\mathbb{E}_{\pi(\mathbf{x})}\mathbb{E}_{\mathcal{N}(0,\mathbf{I})}\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}) + \frac{\boldsymbol{\epsilon}}{\sigma}\right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

Langevin dynamic

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

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

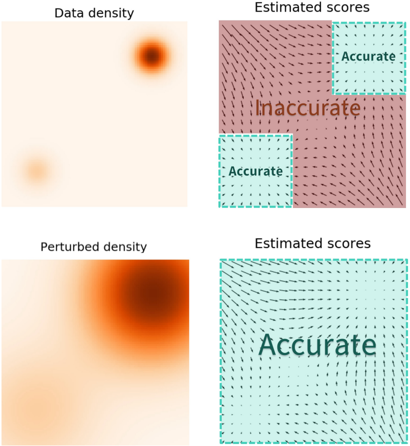
Outline

1. Score matching
   - Denoising score matching
   - Noise Conditioned Score Network (NCSN)
2. Forward gaussian diffusion process
3. Denoising score matching for diffusion
4. Reverse Gaussian diffusion process

Denoising score matching

$$\mathbb{E}_{\pi(\mathbf{x})}\mathbb{E}_{\mathcal{N}(0,\mathbf{I})}\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}+\sigma\cdot\boldsymbol{\epsilon})+\frac{\boldsymbol{\epsilon}}{\sigma}\right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

- If σ is **small**, the score function is not accurate and Langevin dynamics will probably fail to jump between modes.
- If σ is **large**, it is good for low-density regions and multimodal distributions, but we will learn too corrupted distribution.



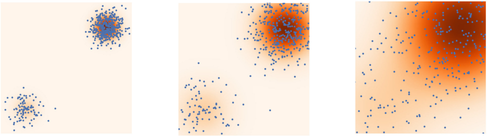
Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021

Noise Conditioned Score Network (NCSN)

- Define the sequence of the noise levels: $\sigma_1 < \sigma_2 < \cdots < \sigma_T$.
- Perturb the original data with the different noise levels to obtain $\mathbf{x}_t = \mathbf{x} + \sigma_t \cdot \boldsymbol{\epsilon}$, $\mathbf{x}_t \sim q(\mathbf{x}_t)$.
- Choose σ_1, σ_T such that:

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

$\sigma_1 < \sigma_2 < \sigma_3$



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

Noise Conditioned Score Network (NCSN)

Train the denoising score function $\mathbf{s}_{\boldsymbol{\theta},\sigma_t}(\mathbf{x}_t)$ for each noise level σ_t using unified weighted objective:

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

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

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 $\mathcal{L} = \sigma_t^2 \cdot \|\mathbf{s}_{\boldsymbol{\theta},\sigma_t}(\mathbf{x}_t) + \frac{\boldsymbol{\epsilon}}{\sigma_t}\|^2$.

How to sample from this model?

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

Noise Conditioned Score Network (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

Outline

Forward gaussian diffusion process

Let $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$, $\beta_t \ll 1$. Define the Markov chain

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

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

Langevin dynamics

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

$$\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t \approx \left(1 - \frac{\beta_t}{2}\right) \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t =$$

$$= \mathbf{x}_{t-1} + \frac{\beta_t}{2} \cdot (-\mathbf{x}_{t-1}) + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t$$

- $\beta_t = \eta$
- $\nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{x}_{t-1} | \boldsymbol{\theta}) = -\mathbf{x}_{t-1} = \nabla_{\mathbf{x}_{t-1}} \log \mathcal{N}(0, \mathbf{I})$

Sohl-Dickstein J. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, 2015

Forward gaussian diffusion process

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

Statement 1

Let denote $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s = \prod_{s=1}^{t}(1-\beta_s)$. Then

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

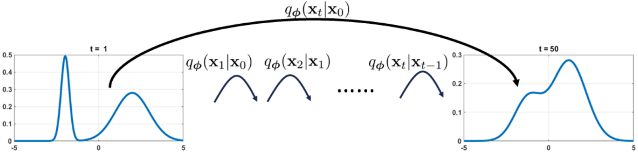
We are able to sample from any timestamp using only $\mathbf{x}_0$!

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t} \cdot \mathbf{x}_{t-1} + \sqrt{1-\alpha_t} \cdot \boldsymbol{\epsilon}_t = \\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}} \cdot \mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}} \cdot \boldsymbol{\epsilon}_{t-1}) + \sqrt{1-\alpha_t} \cdot \boldsymbol{\epsilon}_t = \\
&= \sqrt{\alpha_t \alpha_{t-1}} \cdot \mathbf{x}_{t-2} + (\sqrt{\alpha_t(1-\alpha_{t-1})} \cdot \boldsymbol{\epsilon}_{t-1} + \sqrt{1-\alpha_t} \cdot \boldsymbol{\epsilon}_t) = \\
&= \sqrt{\alpha_t \alpha_{t-1}} \cdot \mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}\alpha_t} \cdot \boldsymbol{\epsilon}'_t = \\
&= \dots = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).
\end{aligned}$$

Sohl-Dickstein J. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, 2015

Forward gaussian diffusion process

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



Statement 2

Applying the Markov chain to samples from any $\pi(\mathbf{x})$ we will get $\mathbf{x}_\infty \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$. Here $p_\infty(\mathbf{x})$ is a **stationary** and **limiting** distribution:

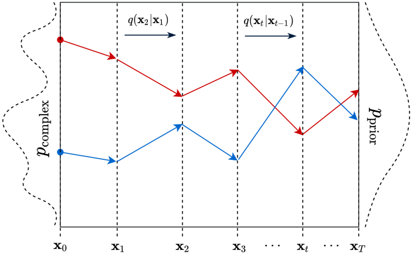
$$p_\infty(\mathbf{x}) = \int q(\mathbf{x}|\mathbf{x}')p_\infty(\mathbf{x}')d\mathbf{x}'.$$

$$p_\infty(\mathbf{x}) = \int q(\mathbf{x}_\infty|\mathbf{x}_0)\pi(\mathbf{x}_0)d\mathbf{x}_0 \approx \mathcal{N}(0, \mathbf{I}) \int \pi(\mathbf{x}_0)d\mathbf{x}_0 = \mathcal{N}(0, \mathbf{I})$$

Chan S. Tutorial on Diffusion Models for Imaging and Vision, 2024
Sohl-Dickstein J. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, 2015

Forward gaussian diffusion process

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

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

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

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

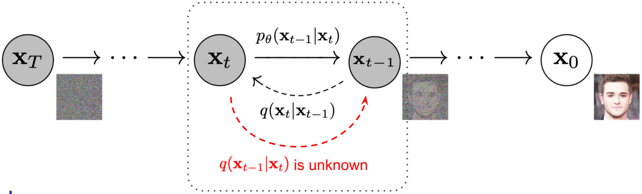
Theorem (denoising score matching)

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

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:

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

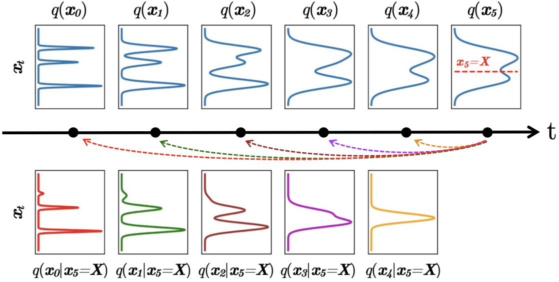
Weng L. What are Diffusion Models?, blog post, 2021

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)

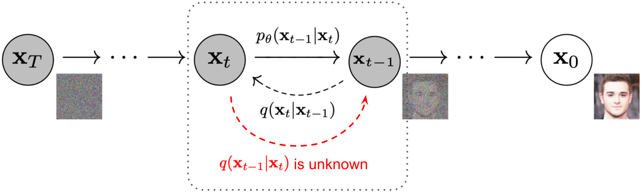
If β_t is small enough, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ will be Gaussian (that is why diffusion needs $T \approx 1000$ steps to converge).



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

Conditioned reverse distribution

Reverse kernel (**intractable**)

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

Conditioned reverse kernel (**tractable**)

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

Here

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

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Distribution summary

Forward process goes from any distribution $\pi(\mathbf{x})$ to $\mathcal{N}(0, \mathbf{I})$ via noise injection.

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

Reverse process is Intractable distribution that is able to be approximated by Normal (with unknown parameters) for small β_t.

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

Conditioned reverse process is Normal with the known parameters, which defines how to denoise a noisy image $\mathbf{x}_t$ with access to what the final, completely denoised image $\mathbf{x}_0$ should be.

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Summary

- Denoising score matching minimizes the Fisher divergence on noisy samples. It allows to estimate the Fisher divergence using samples.
- Noise conditioned score network uses multiple noise levels and annealed Langevin dynamics to fit the score function and sample from the model.
- Gaussian diffusion process is a Markov chain that injects special form of Gaussian noise to the samples.
- Denoising score matching with Langevin dynamics is applicable to Gaussian diffusion process.
- Reverse process allows to recover the real samples from the noise, but it is intractable.