Deep Generative Models

Lecture 12

Roman Isachenko

Moscow Institute of Physics and Technology

2023, Autumn

Recap of previous lecture

Forward gaussian diffusion process

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

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

- $p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$
- $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1-\bar{\alpha}_t) \cdot \mathbf{I})$

Reverse gaussian diffusion process

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

Forward process

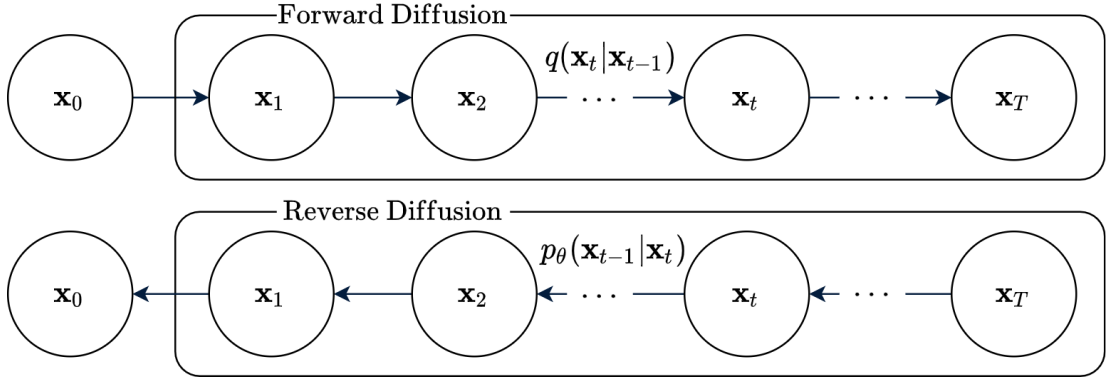
1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$.

Reverse process

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

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

Recap of previous lecture



- Let treat $\mathbf{z} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ as a latent variable (**note:** each $\mathbf{x}_t$ has the same size).
- Variational posterior distribution (**note:** there is no learnable parameters)

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

- Probabilistic model

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

- Generative distribution and prior

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}); \quad p(\mathbf{z}|\boldsymbol{\theta}) = \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot p(\mathbf{x}_T)$$

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

Outline

1. Denoising Diffusion Probabilistic Model (DDPM)
   Reparametrization of gaussian diffusion model
   Overview of DDPM

2. Langevin dynamic and SDE basics

3. Score matching

Outline

Outline

Reparametrization of DDPM

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

$\mathcal{L}_t$ is a KL between two normal distributions:

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

Here

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

Let assume

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Reparametrization of DDPM

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

Use KL formula between two normal distributions:

$$\mathcal{L}_t = KL\Big(\mathcal{N}\big(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}\big) || \mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I}\big)\Big)$$
$$= \frac{1}{2\tilde{\beta}_t} \left\| \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right\|^2$$

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

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Reparametrization of DDPM

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

Reparametrization

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

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

$$\mathcal{L}_t = \frac{(1 - \alpha_t)^2}{2\tilde{\beta}_t \alpha_t (1 - \bar{\alpha}_t)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right\|^2$$

$$= \frac{(1 - \alpha_t)^2}{2\tilde{\beta}_t \alpha_t (1 - \bar{\alpha}_t)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t\right) \right\|^2$$

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

Denoising diffusion probabilistic model (DDPM)

DDPM is a VAE model

- Encoder is a fixed Gaussian Markov chain.
- Latent variable is a hierarchical (in each step the dim. of the latent equals to the dim of the input).
- Decoder is a simple Gaussian model.
- Prior distribution is given by parametric Gaussian Makov chain.

Forward process

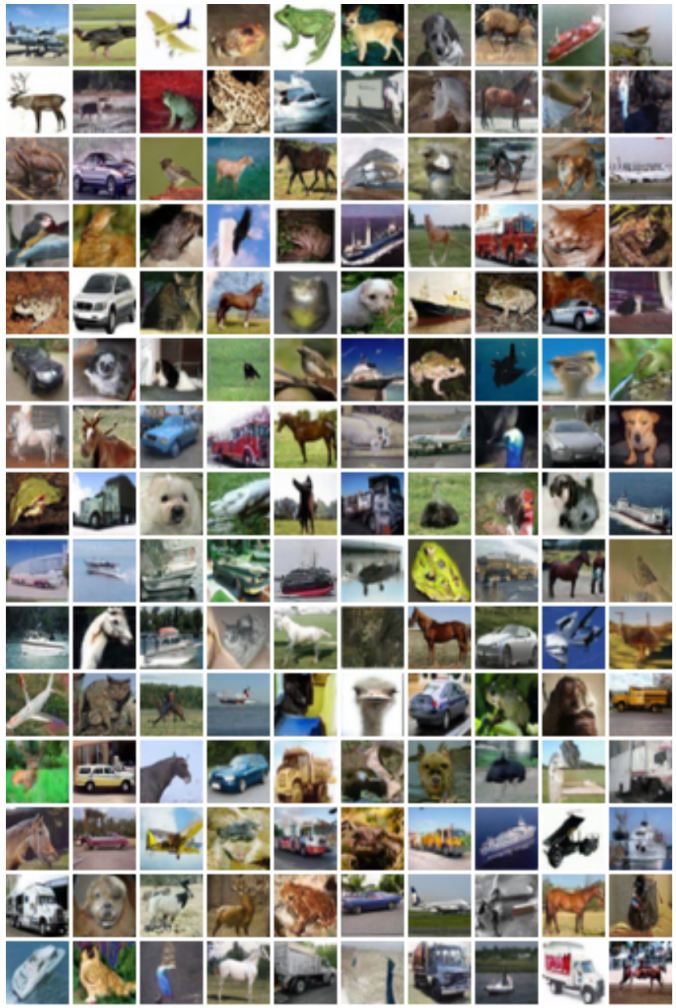
1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$.

Reverse process

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

Denoising diffusion probabilistic model (DDPM)

Samples



Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

Langevin dynamic

Imagine that we have some generative model $p(\mathbf{x}|\boldsymbol{\theta})$.

Statement

Let $\mathbf{x}_0$ be a random vector. Then under mild regularity conditions for small enough η samples from the following dynamics

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

will comes from $p(\mathbf{x}|\boldsymbol{\theta})$.

What do we get if $\boldsymbol{\epsilon} = \mathbf{0}$?

Energy-based model

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{\hat{p}(\mathbf{x}|\boldsymbol{\theta})}{Z_{\boldsymbol{\theta}}}, \quad \text{where } Z_{\boldsymbol{\theta}} = \int \hat{p}(\mathbf{x}|\boldsymbol{\theta})d\mathbf{x}$$

$$\nabla_{\mathbf{x}}\log p(\mathbf{x}|\boldsymbol{\theta}) = \nabla_{\mathbf{x}}\log\hat{p}(\mathbf{x}|\boldsymbol{\theta}) - \nabla_{\mathbf{x}}\log Z_{\boldsymbol{\theta}} = \nabla_{\mathbf{x}}\log\hat{p}(\mathbf{x}|\boldsymbol{\theta})$$

Gradient of normalized density equals to gradient of unnormalized density.

Welling M. Bayesian Learning via Stochastic Gradient Langevin Dynamics, 2011

Stochastic differential equation (SDE)

Let define stochastic process $\mathbf{x}(t)$ with initial condition $\mathbf{x}(0) \sim p_0(\mathbf{x})$:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$$

- $\mathbf{f}(\mathbf{x}, t)$ is the **drift** function of $\mathbf{x}(t)$.
- $g(t)$ is the **diffusion** coefficient of $\mathbf{x}(t)$.
- If $g(t) = 0$ we get standard ODE.
- $\mathbf{w}(t)$ is the standard Wiener process (Brownian motion)

$$\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(0, t-s), \quad d\mathbf{w} = \boldsymbol{\epsilon} \cdot \sqrt{dt}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

How to get distribution $p(\mathbf{x}, t)$ for $\mathbf{x}(t)$?

Theorem (Kolmogorov-Fokker-Planck)

Evolution of the distribution $p(\mathbf{x}, t)$ is given by the following ODE:

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = \operatorname{tr}\left(-\frac{\partial}{\partial \mathbf{x}} \left[\mathbf{f}(\mathbf{x}, t) p(\mathbf{x}, t) \right] + \frac{1}{2} g^2(t) \frac{\partial^2 p(\mathbf{x}, t)}{\partial \mathbf{x}^2} \right)$$

Langevin SDE (special case)

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \quad d\mathbf{w} = \boldsymbol{\epsilon} \cdot \sqrt{dt}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

$$d\mathbf{x} = \frac{1}{2}\frac{\partial}{\partial \mathbf{x}} \log p(\mathbf{x}, t)dt + 1d\mathbf{w}$$

$$\mathbf{x}_{t+1} - \mathbf{x}_t = \eta\frac{1}{2}\frac{\partial}{\partial \mathbf{x}} \log p(\mathbf{x}, t) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}, \quad \eta \approx dt.$$

Let apply KFP theorem.

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = \text{tr}\left(-\frac{\partial}{\partial \mathbf{x}}\left[p(\mathbf{x}, t)\frac{1}{2}\frac{\partial}{\partial \mathbf{x}} \log p(\mathbf{x}, t)\right] + \frac{1}{2}\frac{\partial^2 p(\mathbf{x}, t)}{\partial \mathbf{x}^2}\right) =$$

$$= \text{tr}\left(-\frac{\partial}{\partial \mathbf{x}}\left[\frac{1}{2}\frac{\partial}{\partial \mathbf{x}} p(\mathbf{x}, t)\right] + \frac{1}{2}\frac{\partial^2 p(\mathbf{x}, t)}{\partial \mathbf{x}^2}\right) = 0$$

The density $p(\mathbf{x}, t) = \text{const}$.

Langevin dynamic

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta\frac{1}{2}\nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}, \quad \eta \approx dt.$$

Stochastic differential equation (SDE)

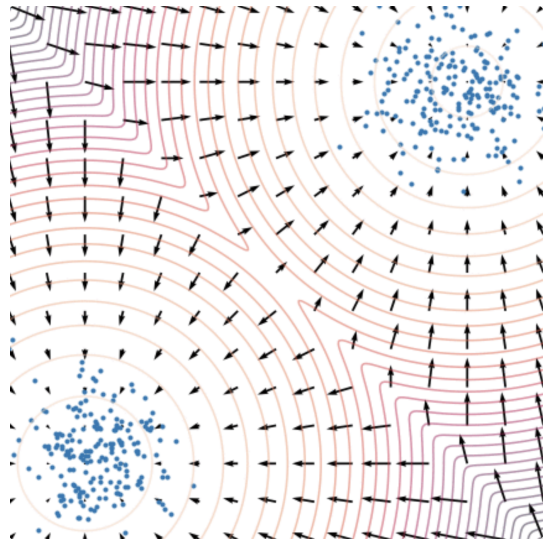
Statement

Let $\mathbf{x}_0$ be a random vector. Then samples from the following dynamics

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

will come from $p(\mathbf{x}|\boldsymbol{\theta})$ under mild regularity conditions for small enough η and large enough t.

The density $p(\mathbf{x}|\boldsymbol{\theta})$ is a **stationary** distribution for this SDE.



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

Outline

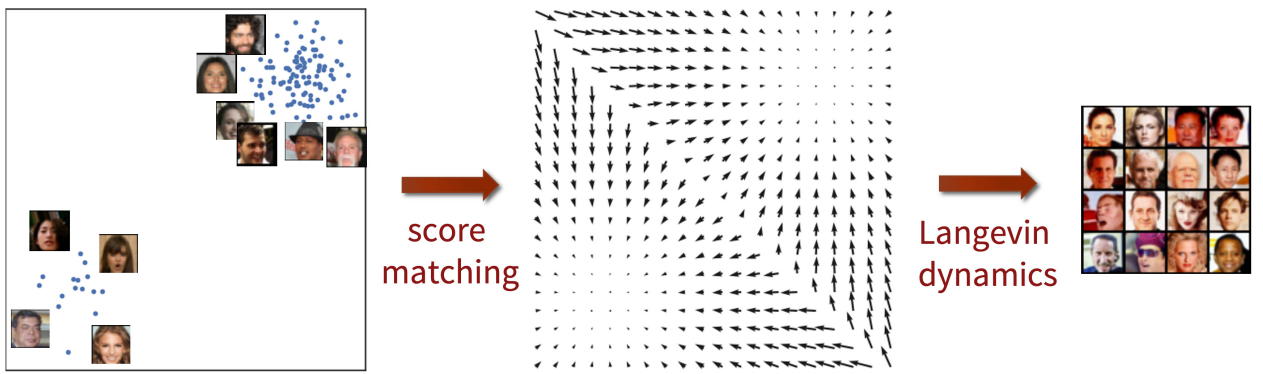
Score matching

We could sample from the model using Langevin dynamics if we have $\nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$.

Fisher divergence

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

Let introduce **score function** $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$.



Problem: we do not know $\nabla_{\mathbf{x}} \log \pi(\mathbf{x})$.

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

Score matching

Theorem (implicit score matching)

Under some regularity conditions, it holds

$$\frac{1}{2}\mathbb{E}_\pi \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 = \mathbb{E}_\pi \left[\frac{1}{2} \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2 + \operatorname{tr}\left(\nabla_{\mathbf{x}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) \right) \right] + \text{const}$$

Proof (only for 1D)

$$\mathbb{E}_\pi \left\| s(x) - \nabla_x \log \pi(x) \right\|_2^2 = \mathbb{E}_\pi \left[s(x)^2 + (\nabla_x \log \pi(x))^2 - 2[s(x) \nabla_x \log \pi(x)] \right]$$

$$\begin{aligned}
\mathbb{E}_\pi [s(x) \nabla_x \log \pi(x)] &= \int \pi(x) \nabla_x \log p(x) \nabla_x \log \pi(x) dx \\
&= \int \nabla_x \log p(x) \nabla_x \pi(x) dx = \pi(x) \nabla_x \log p(x) \Big|_{-\infty}^{+\infty} \\
&- \int \nabla_x^2 \log p(x) \pi(x) dx = -\mathbb{E}_\pi \nabla_x^2 \log p(x) = -\mathbb{E}_\pi \nabla_x s(x)
\end{aligned}$$

$$\frac{1}{2}\mathbb{E}_\pi \left\| s(x) - \nabla_x \log \pi(x) \right\|_2^2 = \mathbb{E}_\pi \left[\frac{1}{2} s(x)^2 + \nabla_x s(x) \right] + \text{const.}$$

Hyvarinen A. Estimation of non-normalized statistical models by score matching, 2005

Score matching

Theorem (implicit score matching)

$$\frac{1}{2}\mathbb{E}_\pi \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 = \mathbb{E}_\pi \left[\frac{1}{2} \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2 + \operatorname{tr}\left(\nabla_{\mathbf{x}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\right) \right] + \text{const}$$

Here $\nabla_{\mathbf{x}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\mathbf{x}}^2 \log p(\mathbf{x}|\boldsymbol{\theta})$ is a Hessian matrix.

1. The left hand side is intractable due to unknown $\pi(\mathbf{x})$ – **denoising score matching**.
2. The right hand side is complex due to Hessian matrix – **sliced score matching**.

Sliced score matching (Hutchinson's trace estimation)

$$\operatorname{tr}\left(\nabla_{\mathbf{x}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\right) = \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\boldsymbol{\epsilon}^T \nabla_{\mathbf{x}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) \boldsymbol{\epsilon} \right]$$

Song Y. Sliced Score Matching: A Scalable Approach to Density and Score Estimation, 2019
Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021

Denoising score matching

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

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

$$\pi(\mathbf{x}'|\sigma) = \int \pi(\mathbf{x}) p(\mathbf{x}'|\mathbf{x}, \sigma) d\mathbf{x}.$$

Then the solution of

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

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

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

Denoising score matching

Theorem

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

Gradient of the noise kernel

$$\nabla_{\mathbf{x}'} \log p(\mathbf{x}'|\mathbf{x}, \sigma) = \nabla_{\mathbf{x}'} \log \mathcal{N}(\mathbf{x}'|\mathbf{x}, \sigma^2\mathbf{I}) = -\frac{\mathbf{x}' - \mathbf{x}}{\sigma^2}$$

- The RHS does not need to compute $\nabla_{\mathbf{x}'} \log \pi(\mathbf{x}'|\sigma)$ and even $\nabla_{\mathbf{x}'} \log \pi(\mathbf{x}')$.
- $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}', \sigma)$ tries to **denoise** a corrupted sample $\mathbf{x}'$.
- Score function $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}', \sigma)$ parametrized by σ. How to make it?

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

Summary

- At each step DDPM predicts the noise that was used in forward diffusion process.
- Langevin dynamics allows to sample from the model using the score function (due to the existence of stationary distribution for SDE).
- Score matching proposes to minimize Fisher divergence to get score function.
- Sliced score matching and denoising score matching are two techniques to get scalable algorithm for fitting Fisher divergence.
- Noise conditioned score network uses multiple noise levels and annealed Langevin dynamics to fit score function.
- Objective of DDPM is closely related to the noise conditioned score network and score matching.