Deep Generative Models

Lecture 13

Roman Isachenko

AI Masters

2024, Spring

Recap of previous lecture

Training of DDPM

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

Sampling of DDPM

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Compute mean of $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t), \sigma_t^2 \cdot \mathbf{I})$:

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

3. Get denoised image $\mathbf{x}_{t-1} = \boldsymbol{\mu}_{\boldsymbol{\theta},t}(\mathbf{x}_t) + \sigma_t \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.

Ho J. Denoising Diffusion Probabilistic Models, 2020

Recap of previous lecture

NCSN objective

$$\mathbb{E}_{p(\mathbf{x}'|\mathbf{x},\sigma_t)}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}', \sigma_t) - \nabla_{\mathbf{x}'} \log p(\mathbf{x}'|\mathbf{x}, \sigma_t)\right\|_2^2 \rightarrow \min_{\boldsymbol{\theta}}$$

DDPM objective

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

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

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

Let reparametrize our model:

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

1. SDE basics
2. Diffusion and Score matching SDEs
3. Probability flow ODE

Outline

1. SDE basics

2. Diffusion and Score matching SDEs

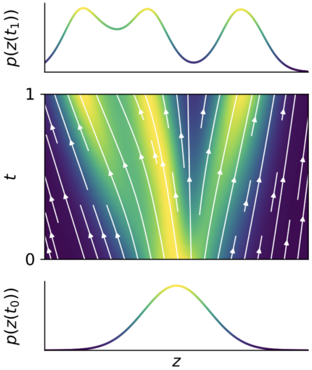
3. Probability flow ODE

Ordinary differential equantion (ODE)

Continuous-in-time Normalizing Flows

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t); \quad \text{with initial condition } \mathbf{z}(t_0) = \mathbf{z}_0$$

- Let $\mathbf{z}(t_0)$ will be a random variable with some density function $p(\mathbf{z}(t_0))$.
- Then $\mathbf{z}(t_1)$ will be also a random variable with some other density function $p(\mathbf{z}(t_1))$.
- We could say that we have the joint density function $p(\mathbf{z}(t), t)$.
- What is the difference between $p(\mathbf{z}(t), t)$ and $p(\mathbf{z}, t)$?



Grathwohl W. et al. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, 2018

Continuous-in-time Normalizing Flows

What do we need?

- We need the way to compute $p(\mathbf{z}, t)$ at any moment t.
- We need the way to find the optimal parameters $\boldsymbol{\theta}$ of the dynamic $\mathbf{f}_{\boldsymbol{\theta}}$.

Theorem (Kolmogorov-Fokker-Planck: special case)

If $\mathbf{f}$ is uniformly Lipschitz continuous in $\mathbf{z}$ and continuous in t, then

$$\frac{d \log p(\mathbf{z}(t), t)}{dt} = -\mathrm{tr}\left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)}\right).$$

$$\log p(\mathbf{z}(t_1), t_1) = \log p(\mathbf{z}(t_0), t_0) - \int_{t_0}^{t_1} \mathrm{tr}\left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)}\right) dt.$$

It means that if we have the value $\mathbf{z}_0 = \mathbf{z}(t_0)$ then the solution of the ODE will give us the density at the moment t_1.

Chen R. T. Q. et al. Neural Ordinary Differential Equations, 2018

Stochastic differential equation (SDE)

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

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

- $\mathbf{f}(\mathbf{x}, t) : \mathbb{R}^m \to \mathbb{R}^m$ is the **drift** function of $\mathbf{x}(t)$.
- $g(t) : \mathbb{R} \to \mathbb{R}$ is the **diffusion** function of $\mathbf{x}(t)$.
- If $g(t) = 0$ we get standard ODE.
- $\mathbf{w}(t)$ is the standard Wiener process (Brownian motion):
    1. $\mathbf{w}(0) = 0$ (almost surely);
    2. $\mathbf{w}(t)$ has independent increments;
    3. $\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(0, (t - s)\mathbf{I})$.
- $d\mathbf{w} = \mathbf{w}(t + dt) - \mathbf{w}(t) = \mathcal{N}(0, \mathbf{I} \cdot dt) = \boldsymbol{\epsilon} \cdot \sqrt{dt}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.

Note: In contrast to ODE, initial condition $\mathbf{x}(0)$ does not uniquely determine the process trajectory.

Stochastic differential equation (SDE)

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

- At each moment t we have the density $p(\mathbf{x}(t), t)$.
- 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:

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

Note: This is the generalization of KFP theorem that we used in continuous-in-time NF.

Langevin SDE (special case)

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

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

Langevin SDE (special case)

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

Let apply KFP theorem.

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

The density $p(\mathbf{x}, t) = \mathrm{const}(t)$!

Discretized Langevin SDE

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

Langevin dynamic

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

ODE and SDE discretization

Outline

Score matching SDE

Denoising score matching

$$\mathbf{x}_l = \mathbf{x} + \sigma_l \cdot \boldsymbol{\epsilon}_l, \quad p(\mathbf{x}_l | \mathbf{x}, \sigma_l) = \mathcal{N}(\mathbf{x}, \sigma_l^2 \mathbf{I})$$

$$\mathbf{x}_{l-1} = \mathbf{x} + \sigma_{l-1} \cdot \boldsymbol{\epsilon}_{l-1}, \quad p(\mathbf{x}_{l-1} | \mathbf{x}, \sigma_{l-1}) = \mathcal{N}(\mathbf{x}, \sigma_{l-1}^2 \mathbf{I})$$

$$\mathbf{x}_l = \mathbf{x}_{l-1} + \sqrt{\sigma_l^2 - \sigma_{l-1}^2} \cdot \boldsymbol{\epsilon}, \quad p(\mathbf{x}_l | \mathbf{x}_{l-1}, \sigma_l) = \mathcal{N}(\mathbf{x}_{l-1}, (\sigma_l^2 - \sigma_{l-1}^2) \cdot \mathbf{I})$$

Let turn this Markov chain to the continuous stochastic process $\mathbf{x}(t)$ taking $L \to \infty$:

$$\mathbf{x}(t + dt) = \mathbf{x}(t) + \sqrt{\frac{\sigma^2(t + dt) - \sigma^2(t)}{dt}} dt \cdot \boldsymbol{\epsilon} = \mathbf{x}(t) + \sqrt{\frac{d[\sigma^2(t)]}{dt}} \cdot d\mathbf{w}$$

Variance Exploding SDE

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}} \cdot d\mathbf{w}$$

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Diffusion SDE

Denoising Diffusion

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

Let turn this Markov chain to the continuous stochastic process taking $T \to \infty$ and taking $\beta(\frac{t}{T}) = \beta_t \cdot T$

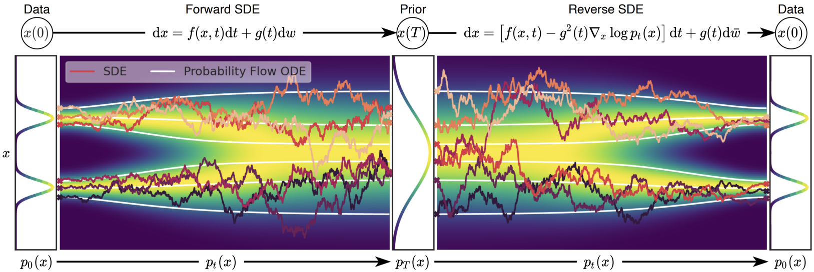
$$\begin{aligned}
\mathbf{x}(t) = \sqrt{1-\beta(t)dt} \cdot \mathbf{x}(t-dt) + \sqrt{\beta(t)dt} \cdot \boldsymbol{\epsilon} \approx \\
\approx (1 - \frac{1}{2}\beta(t)dt) \cdot \mathbf{x}(t-dt) + \sqrt{\beta(t)dt} \cdot \boldsymbol{\epsilon} = \\
= \mathbf{x}(t-dt) - \frac{1}{2}\beta(t)\mathbf{x}(t-dt)dt + \sqrt{\beta(t)} \cdot d\mathbf{w}
\end{aligned}$$

Variance Preserving SDE

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}(t)dt + \sqrt{\beta(t)} \cdot d\mathbf{w}$$

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Diffusion SDE



Variance Exploding SDE (NCSN)

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}} \cdot d\mathbf{w}$$

Variance Preserving SDE (DDPM)

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}(t)dt + \sqrt{\beta(t)} \cdot d\mathbf{w}$$

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Outline

1. SDE basics

2. Diffusion and Score matching SDEs

3. Probability flow ODE

Probability flow ODE

Stochastic differential equation

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

Theorem (Kolmogorov-Fokker-Planck)

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

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\frac{\partial}{\partial \mathbf{x}}\log p(\mathbf{x}, t)\right]dt = \tilde{\mathbf{f}}(\mathbf{x}, t)dt$$

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Probability flow ODE

Kolmogorov-Fokker-Planck equation

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

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Probability flow ODE

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

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

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Summary

- Score matching (NCSN) and diffusion models (DDPM) are the discretizations of the SDEs (variance exploding and variance preserving).