Deep Generative Models

Lecture 12

Roman Isachenko

AI Masters

2025, Spring

Recap of previous lecture

Theorem (continuity equation)

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

$$\frac{d \log p_t(\mathbf{x}(t))}{dt} = -\text{tr}\left(\frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}\right)$$

$$\log p_1(\mathbf{x}(1)) = \log p_0(\mathbf{x}(0)) - \int_0^1 \text{tr}\left(\frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}\right) dt.$$

- **Discrete-in-time NF**: evaluation of determinant of the Jacobian costs $O(m^3)$ (we need invertible $\mathbf{f}$).
- **Continuous-in-time NF**: getting the trace of the Jacobian costs $O(m^2)$ (we need smooth $\mathbf{f}$).

Hutchinson's trace estimator

$$\log p_1(\mathbf{x}(1)) = \log p_0(\mathbf{x}(0)) - \mathbb{E}_{p(\boldsymbol{\epsilon})} \int_0^1 \left[\boldsymbol{\epsilon}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \boldsymbol{\epsilon}\right] dt.$$

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

Recap of previous lecture

Forward pass (Loss function)

$$L(\mathbf{x}) = -\log p_1(\mathbf{x}(1)|\boldsymbol{\theta}) = -\log p_0(\mathbf{x}(0)) + \int_0^1 \operatorname{tr}\left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}\right) dt$$

Adjoint functions

$$\mathbf{a}_{\mathbf{x}}(t) = \frac{\partial L}{\partial \mathbf{x}(t)}; \quad \mathbf{a}_{\boldsymbol{\theta}}(t) = \frac{\partial L}{\partial \boldsymbol{\theta}(t)}.$$

Theorem (Pontryagin)

$$\frac{d\mathbf{a}_{\mathbf{x}}(t)}{dt} = -\mathbf{a}_{\mathbf{x}}(t)^T \cdot \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)}{\partial \mathbf{x}}; \quad \frac{d\mathbf{a}_{\boldsymbol{\theta}}(t)}{dt} = -\mathbf{a}_{\mathbf{x}}(t)^T \cdot \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)}{\partial \boldsymbol{\theta}}.$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}(0)} = \mathbf{a}_{\boldsymbol{\theta}}(0) = -\int_1^0 \mathbf{a}_{\mathbf{x}}(t)^T \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)}{\partial \boldsymbol{\theta}(t)} dt + 0$$

$$\frac{\partial L}{\partial \mathbf{x}(0)} = \mathbf{a}_{\mathbf{x}}(0) = -\int_1^0 \mathbf{a}_{\mathbf{x}}(t)^T \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} dt + \frac{\partial L}{\partial \mathbf{x}(1)}$$

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

Recap of previous lecture

Forward pass

$$\mathbf{x}(1) = \mathbf{x}(0) + \int_0^1 \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)dt \quad \Rightarrow \quad \text{ODE Solver}$$

Backward pass

$$\left.\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{\theta}(0)} &= \mathbf{a}_{\boldsymbol{\theta}}(0) = -\int_1^0 \mathbf{a}_{\mathbf{x}}(t)^T \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)}{\partial \boldsymbol{\theta}(t)} dt + 0 \\
\frac{\partial L}{\partial \mathbf{x}(0)} &= \mathbf{a}_{\mathbf{x}}(0) = -\int_1^0 \mathbf{a}_{\mathbf{x}}(t)^T \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} dt + \frac{\partial L}{\partial \mathbf{x}(1)} \\
\mathbf{x}(0) &= -\int_0^1 \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)dt + \mathbf{x}(1).
\end{aligned}\right\} \Rightarrow \text{ODE Solver}$$

Note: These scary formulas are the standard backprop in the discrete case.

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

Outline

Summary

- KFP equation defines the dynamic of the probability function for the SDE.
- Langevin SDE has constant probability path.
- There exists special probability flow ODE for each SDE that gives the same probability path.
- It is possible to revert SDE using the score function.
- Score matching (NCSN) and diffusion models (DDPM) are the discretizations of the SDEs (variance exploding and variance preserving).