Deep Generative Models

Lecture 11

Roman Isachenko

AI Masters

2025, Spring

Recap of previous lecture

DDPM objective

$$\mathbb{E}_{\pi(\mathbf{x}_0)}\mathbb{E}_{t\sim U\{1,T\}}\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t\alpha_t}\left\|\mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t)-\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t|\mathbf{x}_0)\right\|_2^2\right]$$

In practice the coefficient is omitted.

NCSN objective

$$\mathbb{E}_{\pi(\mathbf{x}_0)}\mathbb{E}_{t\sim U\{1,T\}}\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma_t}(\mathbf{x}_t)-\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t|\mathbf{x}_0)\right\|_2^2$$

Note: The objective of DDPM and NCSN is almost identical. But the difference in sampling scheme:

- ▶ NCSN uses annealed Langevin dynamics;
- ▶ DDPM uses ancestral sampling.

Ho J. Denoising Diffusion Probabilistic Models, 2020

Recap of previous lecture

Unconditional generation

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \cdot \mathbf{x}_t + \frac{\beta_t}{\sqrt{1-\beta_t}} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) + \sigma_t \cdot \boldsymbol{\epsilon}$$

Conditional generation

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \cdot \mathbf{x}_t + \frac{\beta_t}{\sqrt{1-\beta_t}} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y},\boldsymbol{\theta}) + \sigma_t \cdot \boldsymbol{\epsilon}$$

Conditional distribution

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y},\boldsymbol{\theta}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta})$$

$$= \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) - \frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta},t}(\mathbf{x}_t)}{\sqrt{1-\bar{\alpha}_t}}$$

Here $p(\mathbf{y}|\mathbf{x}_t)$ – classifier on noisy samples (we have to learn it separately).

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Recap of previous lecture

Classifier-corrected noise prediction

$$\epsilon_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y}) = \epsilon_{\boldsymbol{\theta},t}(\mathbf{x}_t) - \sqrt{1 - \bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

Guidance scale

$$\epsilon_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y}) = \epsilon_{\boldsymbol{\theta},t}(\mathbf{x}_t) - \gamma \cdot \sqrt{1 - \bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

▶ Train DDPM as usual.

▶ Train the additional classifier $p(\mathbf{y}|\mathbf{x}_t)$ on the noisy samples $\mathbf{x}_t$.

Guided sampling

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

Note: Guidance scale γ tries to sharpen the distribution $p(\mathbf{y}|\mathbf{x}_t)$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Recap of previous lecture

Previous method requires training the additional classifier model $p(\mathbf{y}|\mathbf{x}_t)$ on the noisy data. Let try to avoid this requirement.

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta})$$

$$\begin{aligned}\nabla_{\mathbf{x}_t}^{\gamma} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \\ &= (1-\gamma) \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta})\end{aligned}$$

Classifier-free-corrected noise prediction

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y}) = \gamma \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y}) + (1-\gamma) \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta},t}(\mathbf{x}_t)$$

▶ Train the single model $\boldsymbol{\epsilon}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y})$ on **supervised** data alternating with real conditioning $\mathbf{y}$ and empty conditioning $\mathbf{y} = \emptyset$.

▶ Apply the model twice during inference.

Ho J., Salimans T. Classifier-Free Diffusion Guidance, 2022

Recap of previous lecture

Continuous-in-time dynamics (Neural ODE)

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

$$\mathbf{x}(t_1) = \int_{t_0}^{t_1} \mathbf{f}_\theta(\mathbf{x}(t), t)dt + \mathbf{x}_0 \approx \text{ODESolve}_f(\mathbf{x}_0, \theta, t_0, t_1).$$

Here $\mathbf{f}_\theta : \mathbb{R}^m \times [t_0, t_1] \to \mathbb{R}^m$ is a vector field.
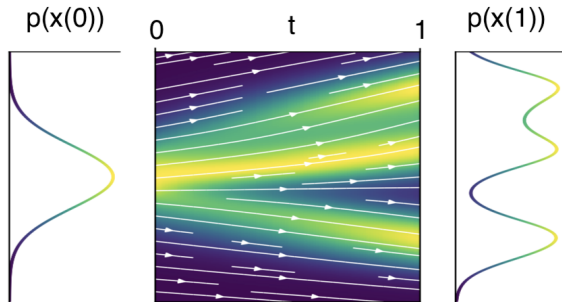
Euler update step (ODESolve)

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{f}_\theta(\mathbf{x}(t), t)$$

▶ Euler method is the simplest version of the ODESolve that is unstable in practice.

▶ It is possible to use more sophisticated numerical methods instead if Euler (e.x. Runge-Kutta methods).

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

Recap of previous lecture

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

▶ Assume that $\mathbf{x}(0)$ is a random variable with the density function $p_0(\mathbf{x})$. Then $\mathbf{x}(t)$ is a random variable with the density function $p_t(\mathbf{x})$.

▶ $p_t(\mathbf{x}) = p(\mathbf{x}, t)$ is the **probability path** between $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$.

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

Recap of previous lecture

Theorem (Picard)

If $\mathbf{f}$ is uniformly Lipschitz continuous in $\mathbf{x}$ and continuous in t, then the ODE has a **unique** solution.

It means that we are able **uniquely revert** our ODE.

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

$$\mathbf{x}(0) = \mathbf{x}(1) + \int_1^0 \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) dt$$

Note: Unlike discrete-in-time NF, $\mathbf{f}$ does not need to be invertible (uniqueness guarantees bijectivity).

How to compute $p_t(\mathbf{x})$ at any moment t?

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

Outline

Outline

Continuous-in-time NF

Theorem (continuity equation)

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

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

It means that if we have the value $\mathbf{x}_0 = \mathbf{x}(0)$ then the solution of the continuity equation will give us the density $p_1(\mathbf{x}(1))$.

Solution of continuity equation

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

▶ This solution will give us the density along the trajectory (not the total probability path).

▶ But it is difficult to estimate the last term efficiently.

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

Outline

Stochastic differential equation (SDE)

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

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

▶ $\mathbf{f}(\mathbf{x}, t) : \mathbb{R}^m \times [0, 1] \to \mathbb{R}^m$ is the **drift** function of $\mathbf{x}(t)$.

▶ $g(t) : \mathbb{R} \to \mathbb{R}$ is the **diffusion** function of $\mathbf{x}(t)$.

▶ $\mathbf{w}(t)$ is the standard Wiener process (Brownian motion):
   1. $\mathbf{w}(0) = 0$ (almost surely);
   2. $\mathbf{w}(t)$ has independent increments;
   3. $\mathbf{w}(t)$ has continuous trajectories;
   4. $\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(0, (t - s)\mathbf{I})$, for $t > s$;

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

▶ If $g(t) = 0$ we get standard ODE.

Stochastic differential equation (SDE)

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

- In contrast to ODE, initial condition $\mathbf{x}(0)$ does not uniquely determine the process trajectory.
- We have two sources of randomness: initial distribution $p_0(\mathbf{x})$ and Wiener process $\mathbf{w}(t)$.

Discretization of SDE (Euler method) - SDESolve

$$\mathbf{x}(t + dt) = \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), t) \cdot dt + g(t) \cdot \boldsymbol{\epsilon} \cdot \sqrt{dt}$$

If $dt = 1$, then

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{f}(\mathbf{x}_t, t) + g(t) \cdot \boldsymbol{\epsilon}$$

- At each moment t we have the density $p_t(\mathbf{x}) = p(\mathbf{x}, t)$.
- $p : \mathbb{R}^m \times [0, 1] \to \mathbb{R}_+$ is a **probability path** between $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$.
- How to get the distribution path $p_t(\mathbf{x})$ for $\mathbf{x}(t)$?

Stochastic differential equation (SDE)

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

Theorem (Kolmogorov-Fokker-Planck)

Evolution of the distribution $p_t(\mathbf{x})$ is given by the following equation:

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\text{div}\left(\mathbf{f}(\mathbf{x}, t)p_t(\mathbf{x})\right) + \frac{1}{2}g^2(t)\Delta_{\mathbf{x}}p_t(\mathbf{x})$$

Here

$$\text{div}(\mathbf{v}) = \sum_{i=1}^{m} \frac{\partial v_i(\mathbf{x})}{\partial x_i} = \text{tr}\left(\frac{\partial \mathbf{v}(\mathbf{x})}{\partial \mathbf{x}}\right)$$

$$\Delta_{\mathbf{x}}p_t(\mathbf{x}) = \sum_{i=1}^{m} \frac{\partial^2 p_t(\mathbf{x})}{\partial x_i^2} = \text{tr}\left(\frac{\partial^2 p_t(\mathbf{x})}{\partial \mathbf{x}^2}\right)$$

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

Stochastic differential equation (SDE)

Theorem (Kolmogorov-Fokker-Planck)

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

▶ KFP theorem uniquely defines the density $p_t(\mathbf{x})$.
▶ This is the generalization of continuity equation that we used in continuous-in-time NF:

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

Langevin SDE (special case)

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

Let apply KFP theorem to this SDE.

Langevin SDE (special case)

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

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

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

The density $p_t(\mathbf{x}) = \mathrm{const}(t)$!
If $\mathbf{x}(0) \sim p_0(\mathbf{x})$, then $\mathbf{x}(t) \sim p_0(\mathbf{x})$.

Discretized Langevin SDE

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

Langevin dynamic

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

Outline

Probability flow ODE

ODE and continuity equation

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

$$\frac{d \log p_t(\mathbf{x}(t))}{dt} = -\text{tr}\left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, t)}{\partial \mathbf{x}}\right) \quad \Leftrightarrow \quad \frac{\partial p_t(\mathbf{x})}{\partial t} = -\text{div}\left(\mathbf{f}(\mathbf{x}, t)p_t(\mathbf{x})\right)$$

The only source of stochasticity is the distibution $p_0(\mathbf{x})$.

SDE and KFP equation

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

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\text{div}\left(\mathbf{f}(\mathbf{x}, t)p_t(\mathbf{x})\right) + \frac{1}{2}g^2(t)\Delta_{\mathbf{x}}p_t(\mathbf{x})$$
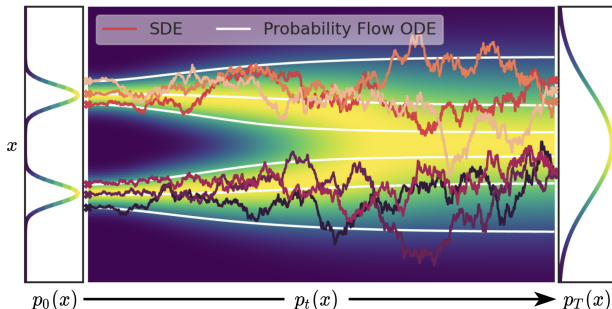
We have two sources of randomness: initial distribution $p_0(\mathbf{x})$ and Wiener process $\mathbf{w}(t)$.

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

Probability flow ODE

Theorem

Assume SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$ induces the probability path $p_t(\mathbf{x})$. Then there exists ODE with identical probability path $p_t(\mathbf{x})$ of the form

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

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

Probability flow ODE

Theorem

Assume SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$ induces the probability path $p_t(\mathbf{x})$. Then there exists ODE with identical probability path $p_t(\mathbf{x})$ of the form

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

Proof

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

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

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

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

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

Probability flow ODE

Proof (continued)

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

$$= \text{tr}\left(-\frac{\partial}{\partial \mathbf{x}}\left[\tilde{\mathbf{f}}(\mathbf{x}, t)p_t(\mathbf{x})\right]\right) = -\text{div}\left(\tilde{\mathbf{f}}(\mathbf{x}, t)p_t(\mathbf{x})\right)$$

$$\tilde{\mathbf{f}}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\frac{\partial}{\partial \mathbf{x}}\log p_t(\mathbf{x}); \quad \tilde{g}(t) = 0$$

$$d\mathbf{x} = \tilde{\mathbf{f}}(\mathbf{x}, t)dt + 0 \cdot d\mathbf{w} = \left(\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\frac{\partial}{\partial \mathbf{x}}\log p_t(\mathbf{x})\right)dt$$
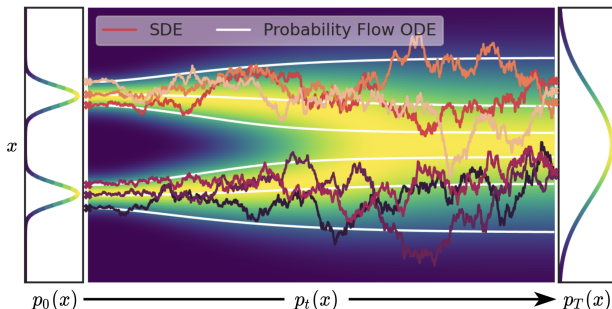
$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\text{div}\left(\tilde{\mathbf{f}}(\mathbf{x}, t)p_t(\mathbf{x})\right) + \frac{1}{2}\tilde{g}^2(t)\Delta_{\mathbf{x}}p_t(\mathbf{x})$$

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

Probability flow ODE

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

$$d\mathbf{x} = \left(\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt - \text{probability flow ODE}$$

▶ The term $\mathbf{s}(\mathbf{x}, t) = \frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x})$ is a score function for continuous time.

▶ ODE has more stable trajectories.

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

Outline

Reverse SDE

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

Here dt could be > 0 or < 0.

Reverse ODE
Let $\tau = 1 - t$ $(d\tau = -dt)$.

$$d\mathbf{x} = -\mathbf{f}(\mathbf{x}, 1 - \tau)d\tau$$

▶ How to revert SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$?

▶ Wiener process gives the randomness that we have to revert.

Theorem
There exists the reverse SDE for the SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$ that has the following form

$$d\mathbf{x} = \left(\mathbf{f}(\mathbf{x}, t) - g^2(t)\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt + g(t)d\mathbf{w}$$

with $dt < 0$.

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

Reverse SDE

Theorem

There exists the reverse SDE for the SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$ that has the following form

$$d\mathbf{x} = \left(\mathbf{f}(\mathbf{x}, t) - g^2(t)\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt + g(t)d\mathbf{w}$$

with $dt < 0$.

Note: Here we also see the score function $\mathbf{s}(\mathbf{x}, t) = \frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x})$.

Sketch of the proof

▶ Convert initial SDE to probability flow ODE.

▶ Revert probability flow ODE.

▶ Convert reverse probability flow ODE to reverse SDE.

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

Reverse SDE

Proof

► Convert initial SDE to probability flow ODE

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

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

► Revert probability flow ODE

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

$$d\mathbf{x} = \left(-\mathbf{f}(\mathbf{x}, 1-\tau) + \frac{1}{2}g^2(1-\tau)\frac{\partial}{\partial \mathbf{x}} \log p_{1-\tau}(\mathbf{x}) \right) d\tau$$

► Convert reverse probability flow ODE to reverse SDE

$$d\mathbf{x} = \left(-\mathbf{f}(\mathbf{x}, 1-\tau) + \frac{1}{2}g^2(1-\tau)\frac{\partial}{\partial \mathbf{x}} \log p_{1-\tau}(\mathbf{x}) \right) d\tau$$

$$d\mathbf{x} = \left(-\mathbf{f}(\mathbf{x}, 1-\tau) + g^2(1-\tau)\frac{\partial}{\partial \mathbf{x}} \log p_{1-\tau}(\mathbf{x}) \right) d\tau + g(1-\tau)d\mathbf{w}$$

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

Reverse SDE

Theorem

There exists the reverse SDE for the SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$ that has the following form

$$d\mathbf{x} = \left(\mathbf{f}(\mathbf{x}, t) - g^2(t)\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt + g(t)d\mathbf{w}$$

with $dt < 0$.

Proof (continued)

$$d\mathbf{x} = \left(-\mathbf{f}(\mathbf{x}, 1 - \tau) + g^2(1 - \tau)\frac{\partial}{\partial \mathbf{x}} \log p_{1-\tau}(\mathbf{x}) \right) d\tau + g(1 - \tau)d\mathbf{w}$$

$$d\mathbf{x} = \left(\mathbf{f}(\mathbf{x}, t) - g^2(t)\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt + g(t)d\mathbf{w}$$

Here $d\tau > 0$ and $dt < 0$.

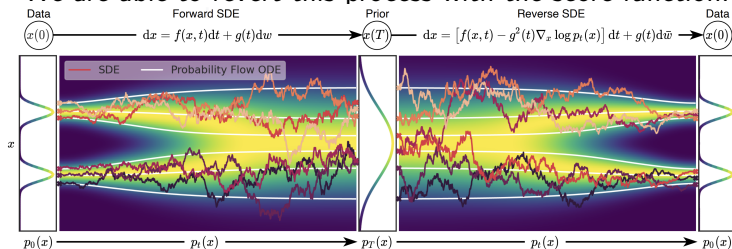Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Reverse SDE

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

$$d\mathbf{x} = \left(\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\frac{\partial}{\partial \mathbf{x}}\log p_t(\mathbf{x})\right)dt - \text{probability flow ODE}$$

$$d\mathbf{x} = \left(\mathbf{f}(\mathbf{x}, t) - g^2(t)\frac{\partial}{\partial \mathbf{x}}\log p_t(\mathbf{x})\right)dt + g(t)d\mathbf{w} - \text{reverse SDE}$$

▶ We got the way to transform one distribution to another via SDE with some probability path $p_t(\mathbf{x})$.

▶ We are able to revert this process with the score function.

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

Outline

Score matching SDE

Denoising score matching

$$\mathbf{x}_t = \mathbf{x} + \sigma_t \cdot \boldsymbol{\epsilon}_t, \qquad q(\mathbf{x}_t|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma_t^2 \cdot \mathbf{I})$$

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

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2} \cdot \boldsymbol{\epsilon}, \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_{t-1}, (\sigma_t^2 - \sigma_{t-1}^2) \cdot \mathbf{I})$$

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

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

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

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

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

Score matching SDE

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

Variance Exploding SDE

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

$\sigma(t)$ is a monotonically increasing function.

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \quad \mathbf{f}(\mathbf{x}, t) = 0, \quad g(t) = \sqrt{\frac{d[\sigma^2(t)]}{dt}}$$

$$d\mathbf{x} = \left(-\frac{1}{2}\frac{d[\sigma^2(t)]}{dt}\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt - \text{probability flow ODE}$$

$$d\mathbf{x} = \left(-\frac{d[\sigma^2(t)]}{dt}\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt + \sqrt{\frac{d[\sigma^2(t)]}{dt}}d\mathbf{w} - \text{reverse SDE}$$

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 $\beta_t = \beta(\frac{t}{T}) \cdot \frac{1}{T}$ (with $dt = \frac{1}{T}$)

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

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 Preserving SDE

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

$$\mathbf{f}(\mathbf{x}, t) = -\frac{1}{2}\beta(t)\mathbf{x}(t), \quad g(t) = \sqrt{\beta(t)}$$

Variance is preserved if $\mathbf{x}(0)$ has a unit variance.

$$d\mathbf{x} = \left(-\frac{1}{2}\beta(t)\mathbf{x}(t) - \frac{1}{2}\beta(t)\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt - \text{probability flow ODE}$$

$$d\mathbf{x} = \left(-\frac{1}{2}\beta(t)\mathbf{x}(t) - \beta(t)\frac{\partial}{\partial \mathbf{x}} \log p_t(\mathbf{x}) \right) dt + \sqrt{\beta(t)}d\mathbf{w} - \text{reverse SDE}$$

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

Diffusion SDE

$$dx = f(x, t)dt + g(t)dw$$

Variance Exploding SDE (NCSN)

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} \cdot dw, \quad f(x, t) = 0, \quad g(t) = \sqrt{\frac{d[\sigma^2(t)]}{dt}}$$

Variance Preserving SDE (DDPM)

$$dx = -\frac{1}{2}\beta(t)x(t)dt + \sqrt{\beta(t)} \cdot dw$$

$$f(x, t) = -\frac{1}{2}\beta(t)x(t), \quad g(t) = \sqrt{\beta(t)}$$

Is it possible to train score-based generative model (DDPM or NCSN) in continuous time?

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

Summary

▶ Continuity equation allows to calculate $\log p(\mathbf{x}, t)$ at arbitrary moment t.

▶ SDE defines a stochastic process with drift and diffusion terms. ODEs are the special case of SDEs.

▶ 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).