Deep Generative Models

Lecture 11

Roman Isachenko

Moscow Institute of Physics and Technology
Yandex School of Data Analysis

2025, Autumn

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, this coefficient is typically 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 objectives of DDPM and NCSN are almost identical; however, they differ in their sampling procedures:

- NCSN utilizes annealed Langevin dynamics;
- DDPM employs 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

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

Here, $p(\mathbf{y}|\mathbf{x}_t)$ denotes the classifier operating on noisy samples (this must be trained separately).

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

Recap of previous lecture

Classifier-corrected noise prediction

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

Guidance scale

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

- Train DDPM as usual.
- Separately train an additional classifier $p(\mathbf{y}|\mathbf{x}_t)$ on noisy samples $\mathbf{x}_t$.

Guided sampling

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

Note: The guidance scale γ serves 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

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

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

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

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 a single model $\boldsymbol{\epsilon}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y})$ on **supervised** data, alternating between 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)

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

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

Here, $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{R}^m \times [t_0, t_1] \to \mathbb{R}^m$ is a vector field.

Euler update step (ODESolve)

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

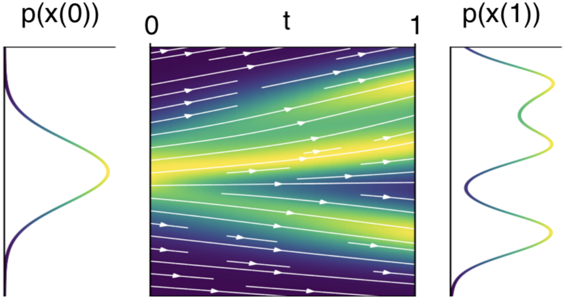
- The Euler method is the simplest version of ODESolve, but it is unstable in practice.
- More advanced numerical methods (e.g., Runge-Kutta methods) can be used instead of Euler.

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

Recap of previous lecture

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

- Suppose $\mathbf{x}(0)$ is a random variable with density $p_0(\mathbf{x})$. Then $\mathbf{x}(t)$ is a random variable with density $p_t(\mathbf{x})$.
- $p_t(\mathbf{x}) = p(\mathbf{x}, t)$ describes 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.

This means we can **uniquely invert** 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-time NF, $\mathbf{f}$ does not need to be invertible (uniqueness assures bijectivity).
How can we compute $p_t(\mathbf{x})$ at any time t?

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

Outline

1. Continuity equation for NF log-likelihood
2. SDE basics
3. Probability flow ODE
4. Reverse SDE

Outline

1. Continuity equation for NF log-likelihood
2. SDE basics
3. Probability flow ODE
4. Reverse SDE

Continuous-in-time NF

Theorem (continuity equation)

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

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

This result states that, given $\mathbf{x}_0 = \mathbf{x}(0)$, the solution to the continuity equation provides the density $p_1(\mathbf{x}(1))$.

Solution of continuity equation

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

- This solution gives the density along the trajectory (not the total probability path).
- However, the latter term is difficult to estimate efficiently.

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

Outline

1. Continuity equation for NF log-likelihood

2. SDE basics

3. Probability flow ODE

4. Reverse SDE

Stochastic differential equation (SDE)

Let us define a 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), characterized by:
  1. $\mathbf{w}(0) = 0$ (almost surely);
  2. $\mathbf{w}(t)$ has independent increments;
  3. $\mathbf{w}(t)$ trajectories are continuous;
  4. $\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(0, (t - s)\mathbf{I})$ for $t > s$;
- $d\mathbf{w} = \mathbf{w}(t + dt) - \mathbf{w}(t) = \mathcal{N}(0, \mathbf{I} \cdot dt) = \epsilon \cdot \sqrt{dt}$, with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.
- If $g(t) = 0$, we recover the standard ODE.

Stochastic differential equation (SDE)

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

- Unlike ODEs, the initial condition $\mathbf{x}(0)$ does not uniquely determine the trajectory of the process.
- There are two sources of randomness: the initial distribution $p_0(\mathbf{x})$ and the Wiener process $\mathbf{w}(t)$.

Discretization of SDE (Euler method) - SDESolve

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

If $dt = 1$, then

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

- At each time t, the process has density $p_t(\mathbf{x}) = p(\mathbf{x}, t)$.
- $p : \mathbb{R}^m \times [0, 1] \to \mathbb{R}_+$ defines a **probability path** from $p_0(\mathbf{x})$ to $p_1(\mathbf{x})$.
- How do we obtain the probability 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)

The evolution of the distribution $p_t(\mathbf{x})$ is governed by:

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

Here

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

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

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

Stochastic differential equation (SDE)

Theorem (Kolmogorov-Fokker-Planck)

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

- The KFP theorem uniquely determines the density $p_t(\mathbf{x})$.
- This generalizes the continuity equation previously used for continuous-time NF:

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

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 us apply the 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

1. Continuity equation for NF log-likelihood
2. SDE basics
3. Probability flow ODE
4. Reverse SDE

Probability flow ODE

ODE and continuity equation

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

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

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

SDE and KFP equation

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

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

We now have two sources of randomness: the initial distribution $p_0(\mathbf{x})$ and the Wiener process $\mathbf{w}(t)$.

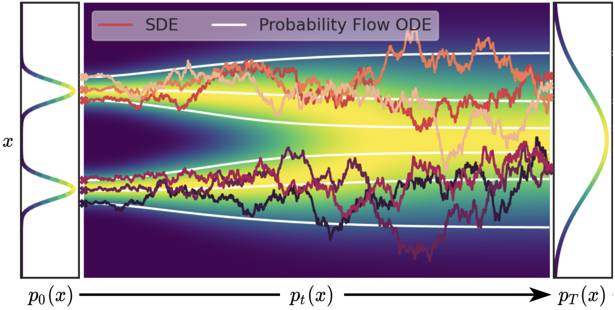
Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Probability flow ODE

Theorem

Assume the 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 an ODE with the identical probability path $p_t(\mathbf{x})$, given by:

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



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

Probability flow ODE

Theorem

Assume the 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 an ODE with the identical probability path $p_t(\mathbf{x})$, given by:

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

Proof

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

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

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

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

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

Probability flow ODE

Proof (continued)

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

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

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

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

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

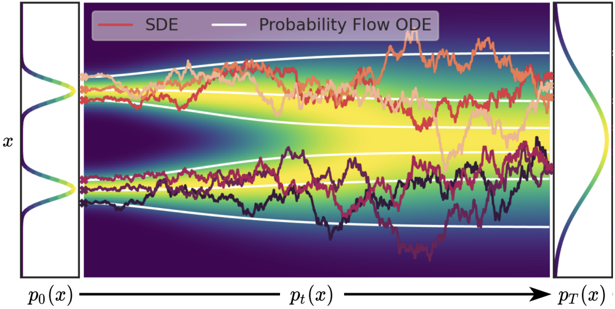
Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

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 the score function in continuous time.
- The ODE has more stable trajectories.



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

Outline

1. Continuity equation for NF log-likelihood
2. SDE basics
3. Probability flow ODE
4. Reverse SDE

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 may 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 do we reverse the SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$?
- The Wiener process introduces randomness that must be reversed.

Theorem

There exists a reverse SDE for $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$, given by:

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

where $dt < 0$.

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

Reverse SDE

Theorem

There exists a reverse SDE for $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$, given by:

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

where $dt < 0$.

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

Sketch of the proof

- Convert the initial SDE to a probability flow ODE.
- Reverse the probability flow ODE.
- Convert the reversed probability flow ODE to a reverse SDE.

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

Reverse SDE

Proof

- Convert the initial SDE to a probability flow ODE:

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

- Reverse the 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 the reversed probability flow ODE to a 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 a reverse SDE for $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$, given by:

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

where $dt < 0$.

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

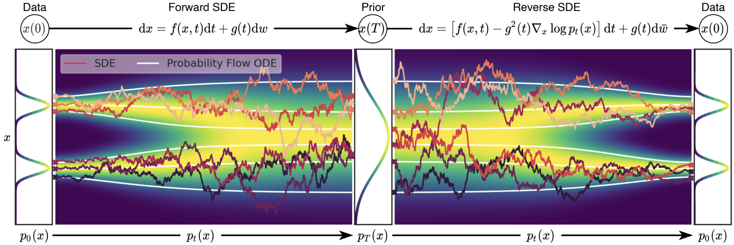
Reverse SDE

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

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

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

- This framework allows us to transform one distribution into another via an SDE with a specified probability path $p_t(\mathbf{x})$.
- We can invert this process using the score function.



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

Summary

- The continuity equation allows the computation of $\log p(\mathbf{x}, t)$ at any time t.
- An SDE defines a stochastic process with drift and diffusion terms; ODEs are a special case of SDEs.
- The KFP equation describes the dynamics of the probability function for the SDE.
- The Langevin SDE preserves a constant probability path.
- For every SDE, there exists a special probability flow ODE that follows the same probability path.
- SDEs can be reversed using the score function.