Flow matching and Diffusion Models

Lecture 1: Generative AI with SDEs

Section 1: From Generation to Sampling

Data distribution: how "likely" are we to find this picture in the Internet.

translate subjective statement (how good an image is) into a statement of probability theory (how likely it is under the data distribution)

- Data distribution: distribution of objects that we want to generate: p_{data}
- Probability density: $p_{data}: \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, $z \to p_{data}(z)$

look a little different

Generation means sampling the data distribution

unconditional generation: a model with fixed prompt: eg. prompt: "dog", you always get dog from this model

conditional generation involves sampling from $z \sim p_{data}(\cdot|y)$, where y is a conditioning variable.

($p_{data}(\cdot|y)$: conditional data distribution)

our goal is to train a generative model to trasform samples from a simple distribution p_{init} (e.g. Gaussian) into samples from p_{data}

Section 2: Flow and Diffusion Models

Flow Models

fundamental object of flow: ① trajectory → a function of time $X: [0,1] \to \mathbb{R}^d$, $t \to$ vector X_t

② vector field → $u: \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, $(x,t) \to u_t(x)$

it gives you directions at every point

ODE: $X_0 = x_0$ (initial condition)

$\frac{d}{dt} X_t = u_t(X_t)$ ⇒ the direction of X_t is given by the location of X_t, and then specified by u_t

Flow: a collection of trajectories that follow the ODE, collection of solutions to an ODE for a lot of initial condition.

we gather a lot of solutions for different initial conditions, and then we gather them all in one function and call that a flow

$$\psi: \mathbb{R}^d \times [0,1] \to \mathbb{R}^d, \quad (x_0, t) \to \psi_t(x_0)$$

$$\frac{d}{dt}\psi_t(x_0) = u(\psi_t(x_0))$$

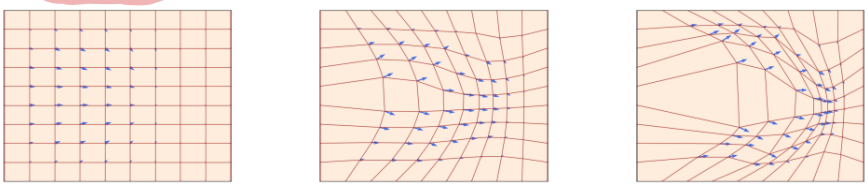
$$\psi_0(x_0) = x_0$$

⇒ flow ODE

⇒ flow initial condition

flow ↑

trajectory —solution→ ODE ←define— vector field



Existence and Uniqueness Theorem ODEs

if the vector field $u_t(x)$ is continuously differentiable with bounded derivatives, then a unique solution to the ODE

$$X_0 = x_0, \quad \frac{d}{dt} X_t = u_t(X_t)$$

exist. In other words, a flow map exists.

Example: Linear ODE

$$u_t(x) = -\theta x \quad (\theta > 0)$$

$$\psi_t(x_0) = \exp(-\theta t)\, x_0$$

Proof:

initial condition: $\psi_0(x_0) = \exp(0)\, x_0 = x_0$

ODE: $\frac{d}{dt}\psi_t(x_0) = \frac{d}{dt}[\exp(-\theta t)\, x_0] = -\theta \exp(-\theta t)\, x_0$

$= -\theta \psi_t(x_0) = u_t(\psi_t(x_0))$

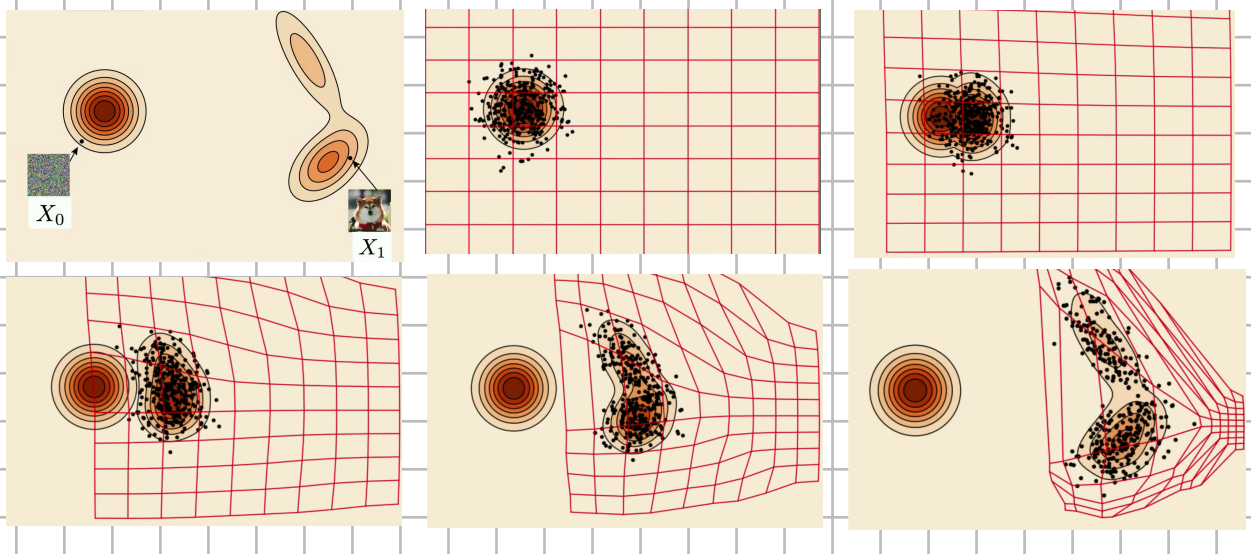
$p_{init} \xrightarrow{ODE} p_{data}$

Neural network: make the vector field a neural network

$u_t^\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ θ is parameters

Random init: $X_0 \sim p_{init}$ goal $X_1 \sim p_{data}$

ODE $\frac{d}{dt} X_t = u_t^\theta(X_t)$



Algorithm 1 Sampling from a Flow Model with Euler method

Require: Neural network vector field u_t^θ, number of steps n

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Draw a sample $X_0 \sim p_{\text{init}}$
4: **for** $i = 1, \ldots, n$ **do**
5: $\quad X_{t+h} = X_t + h u_t^\theta(X_t)$
6: $\quad$ Update $t \leftarrow t + h$
7: **end for**
8: **return** X_1

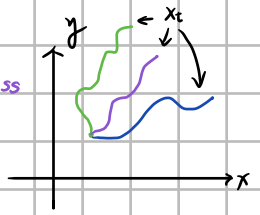
Diffusion Models

SDEs (Stochastic differential equations)

ODE: deterministic trajectory ⇒ SDE: stochastic trajectory ⇒ stochastic process

X_t: random variable $(0 \le t \le 1)$

$X: [0,1] \to \mathbb{R}^d$, $t \to X_t$ is a random trajectory for every draw of X



fundamental object: ① vector field: $u: \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$

② Diffusion coefficient: σ $[0,1] \to \mathbb{R}$ σ is a function of t: σ_t

injecting randomness or stochasticity into ODE

$X_0 = x_0$ (initial condition)

stochastic in SDE and fixed in ODE

$dX_t = u_t(X_t)\, dt + \sigma_t\, dW_t$ W: Brownian motion

direction of a vector field; inject noise

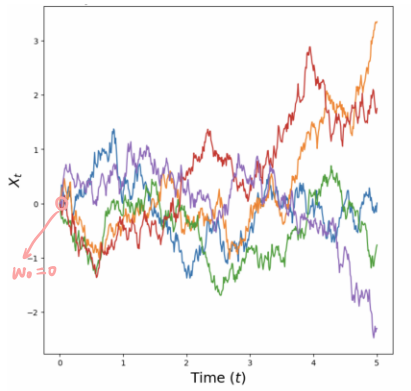
Brownian motion : $W = (W_t)_{t \geq 0}$

① $W_0 = 0$ $\quad W_t \in \mathbb{R}^d$

② Gaussian increments : $W_t - W_s \sim \mathcal{N}(0, (t-s) I_d)$ $\quad (0 \leq s < t)$

③ Independent increments : $W_{t_1} - W_{t_0}, \cdots, W_{t_n} - W_{t_{n-1}}$ are independent

$0 \leq t_0 \leq t_1 \leq \cdots \leq t_n$



"dX_t" notation :

ODE : $\frac{d}{dt} X_t = u_t(X_t) \Leftrightarrow X_{t+h} = X_t + h u_t(X_t) + h R_t(h)$ $\quad \rightarrow o(h)$, error term

$\lim_{h \to 0} R_t(h) = 0$

SDE : $dX_t = u_t(X_t)\,dt + \sigma_t dW_t \Leftrightarrow X_{t+h} = X_t + h u_t(X_t) + \sigma_t (W_{t+h} - W_t) + h R_t(h)$

$\sim \mathcal{N}(0, hI_d)$

$\lim_{h \to 0} \sqrt{\mathbb{E}[\| R_t(h) \|^2]} = 0$

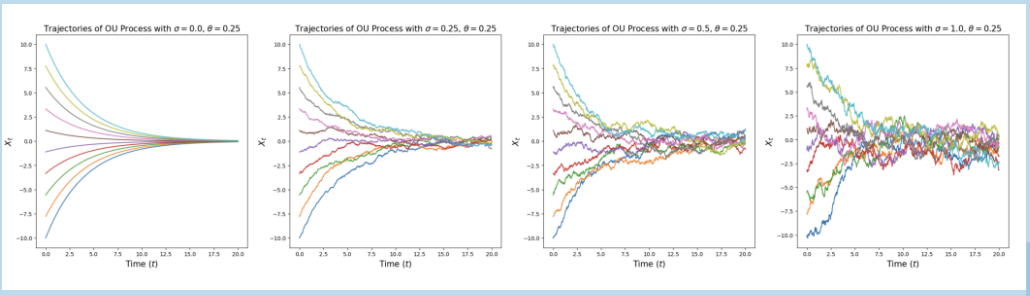
Existence and Uniqueness Theorem SDEs.

if the vector field $u_t(x)$ is continuously differentiable with bounded derivatives and the diffusion coeff. is continuous, then a unique solution to the SDE $X_0 = x_0$, $dX = u_t(X_t)dt + \sigma_t dW_t$ exists.

Example : Ornstein - Uhlenbeck Process :

a constant diffusion coefficient $\sigma_t = \sigma > 0$, a constant linear drift $u_t(x) = -\theta x$ $(\theta > 0)$

$dX_t = -\theta X_t dt + \sigma dW_t$



$p_{init} \xrightarrow{\text{SDE}} p_{data}$

Neural network $u_t^\theta : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$

Diffusion coefficient : σ_t (Fixed)

random init : $X_0 \sim p_{init}$

SDE : $dX_t = u_t^\theta(X_t)\,dt + \sigma_t dW_t$

goal : $X_1 \sim p_{data}$

Algorithm 2 Sampling from a Diffusion Model (Euler-Maruyama method)

Require: Neural network u_t^θ, number of steps n, diffusion coefficient σ_t

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Draw a sample $X_0 \sim p_{\text{init}}$
4: **for** $i = 1, \ldots, n$ **do**
5: Draw a sample $\epsilon \sim \mathcal{N}(0, I_d)$
6: $X_{t+h} = X_t + h u_t^\theta(X_t) + \sigma_t \sqrt{h} \epsilon$
7: Update $t \leftarrow t + h$
8: **end for**
9: **return** X_1