Lec 1. From Generation to Sampling.

Formalize "generating"

objects: vectors $z \in \mathbb{R}^d$. -- we generate.

How good a image is $\simeq$ How likely it is under data distribution.

$p_{data} : \mathbb{R}^d \to \mathbb{R}$.

$z \mapsto p_{data}(z)$

Note: we do NOT know the prob density!

- Generation means sampling from the distribution.
- A dataset consists of finite samples of the distribution.

  $z_1, \dots, z_n \sim p_{data}$.

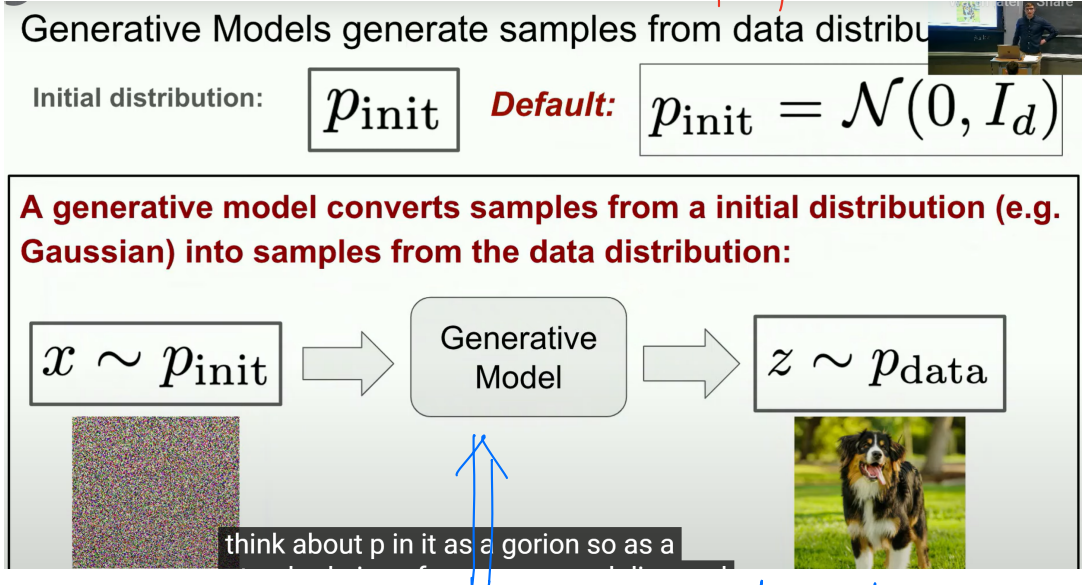
- Conditional / Non-conditional generation.

  ↓ Fixed prompt.

  ↓ sampling the conditional data dist.

  $z \sim p_{data}(\cdot \mid y)$

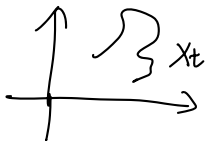
  Flexible prompt.



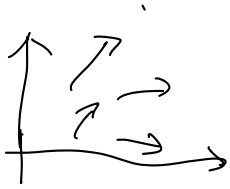
How this achieved. This is what we cares about.

Flow Models

Def. Trajectory. $X: [0,1] \to \mathbb{R}^d \quad t \to X_t$.



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



ODE: $X_0 = x_0$ (initial condition)

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

Flow: A collection of solutions to ODE. (i.e, a collection of trajectory)

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

↑ Init ↑ time.

$$\begin{cases} \psi_0(x_0) = x_0 \\ \frac{d}{dt}\psi_t(x_0) = u_t(\psi_t(x_0)). \end{cases}$$

理解: ODE由向量场确定.

若我们知道 flow 后, ODE 的解 就可

↓ 给出了 $\forall x_0$ (init) $\forall t$ 下的位置

↑ Fix init. find trajectory.

Vector fields define ODEs whose solutions are flows.

Existence and Uniqueness Theorem ODEs

Theorem (Picard–Lindelöf theorem): If the vector field $u_t(x)$ is continuously differentiable with bounded derivatives, then a *unique* solution to the ODE

↘ Lipschitz.

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

exists. In other words, a flow map exists. More generally, this is true if the vector field is **Lipschitz**.

Key takeaway: In the cases of practical interest for machine learning, **unique solutions to ODE/flows exist**.

Eg. Linear ODE.

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

Flow given by: $\varphi_t(x_0) = \exp(-\theta t)\, x_0$.

Check: init, vector field.

Pf: (Verification)

$\varphi_0(x_0) = \exp(0) x_0 = x_0$ ✓

$\frac{d}{dt} \varphi_t(x_0) = -\theta x_0 \exp(-\theta t) = -\theta \varphi_t(x_0) = u_t(\varphi_t(x_0))$ ✓.

Numerical ODE simulation - Euler method

Algorithm 1 Simulating an ODE with the Euler method

Require: Vector field u_t, initial condition x_0, number of steps n

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Set $X_0 = x_0$
4: **for** $i = 1, \ldots, n-1$ **do**
5: $\quad X_{t+h} = X_t + h u_t(X_t)$ *Small step into direction of vector field*
6: $\quad$ Update $t \leftarrow t + h$
7: **end for**
8: **return** $X_0, X_h, X_{2h}, \ldots, X_1$ *Return trajectory*

Flow model: $p_{init} \xrightarrow{ODE} p_{data}$.

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

θ: parameter.

$\theta \in \mathbb{R}^k$ ... to specify vector field nn.

Random init: $X_0 \sim p_{init}$.

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

Goal: $X_1 \sim p_{data}$

(end)

How to generate objects with a Flow Model

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 $X_0 \sim p_{\text{init}}$ *Random initialization!*
4: **for** $i = 1, \ldots, n-1$ **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 *Return final point*

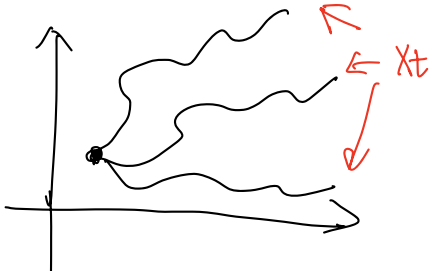
2.2. Diffusion.

SDE.

The solution is random trajectory. (Can be seen as stochastic process).

X_t: random var. (即使 X_0, t 确定, X_t 仍然 uncertain).

$X: [0,1] \to \mathbb{R}^d$. $t \mapsto X_t$.



Vector field. $u \in \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$. + Diffusion coefficient

$\sigma : [0,1] \to \mathbb{R}_{\geq 0}$

$t \to \sigma_t$.

↙ put randomness into ODE.

SDE:

$X_0 = x_0$ init cond. (We fix x_0).

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

Brownian motion. (A specific Gaussian process)

$W = (W_t)_{t \geq 0} \in \mathbb{R}^d$.

① $W_0 = 0$.

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

↑ linear increased variance.

③ Independent increments.

$W_{t_1} - W_{t_0}, \dots, W_{t_n} - W_{t_{n-1}}$ are independent. $0 \leq t_0 < t_1 < \dots < t_n$.

Problems: Brownian motion has no derivatives.

"dX_t" notation:

1. $\frac{d}{dt} X_t = u_t(X_t) \iff X_{t+h} = X_t + h u_t(X_t) + h R_t(h)$

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

$O(h)$

Think this as a taylor approximation.

pf: $\lim_{h \to 0} \frac{X_{t+h} - X_t}{h} = u_t(X_t) \Rightarrow \frac{X_{t+h} - X_t}{h} = u_t(X_t) + R_t(h)$. □

2. $dX_t = u_t(X_t)\,dt + \sigma_t\,dW_t \iff X_{t+h} = X_t + h\,u_t(X_t) + \sigma_t(W_{t+h} - W_t) + h\,R_t(h)$

$W_{t+h} - W_t \sim \mathcal{N}(0, hI_d)$

Sampling.

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

$(\mathrm{var}(R_t(h)) = 0)$

Existence and Uniqueness Theorem SDEs

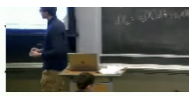
Theorem: If the vector field $u_t(x)$ is continuously differentiable with bounded derivatives and the diffusion coeff. is continuous, then a *unique* solution to the SDE

$$X_0 = x_0, \quad \mathrm{d}X_t = u_t(X_t)\mathrm{d}t + \sigma_t \mathrm{d}W_t$$

exists. More generally, this is true if the vector field is **Lipschitz**.

Key takeaway: In the cases of practical interest for machine learning, **unique solutions to SDE**.

Numerical SDE simulation (Euler-Maruyama method



Algorithm 2 Sampling from a SDE (Euler-Maruyama method)

Require: Vector field u_t, number of steps n, diffusion coefficient σ_t

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Set $X_0 = x_0$
4: **for** $i = 1, \ldots, n-1$ **do**
5: Draw a sample $\epsilon \sim \mathcal{N}(0, I_d)$
6: $X_{t+h} = X_t + h u_t(X_t) + \sigma_t \sqrt{h}\epsilon$
7: Update $t \leftarrow t + h$
8: **end for**
9: **return** $X_0, X_h, X_{2h}, X_{3h}, \ldots, X_1$

↑ var. As we proved.

Add additional noise with var=h scaled by diffusion coefficient σ_t

This is just 1 possible process.

σ_t: Can be dependent on X? (state-dependent)

Eg. OU-process

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

Diffusion model:

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

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

Diffusion coefficient: σ_t (Fixed)

Random init: $X_0 \sim$ input. Goal: $X_1 \sim p_{data}$.

SDE: $u_t^\theta(X_t)dt + \sigma_t d(W_t)$

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** $i = 1, \ldots, n-1$ **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