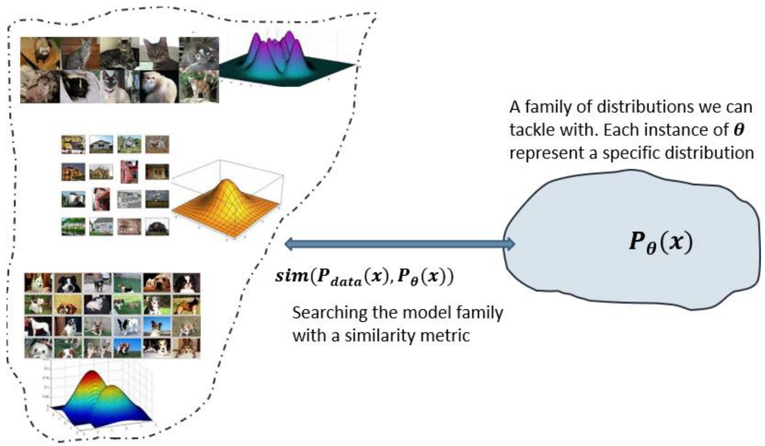
Flow Matching

22-808: Generative models
Sharif University of Technology
Fall 2025

Fatemeh Seyyedsalehi

Recap



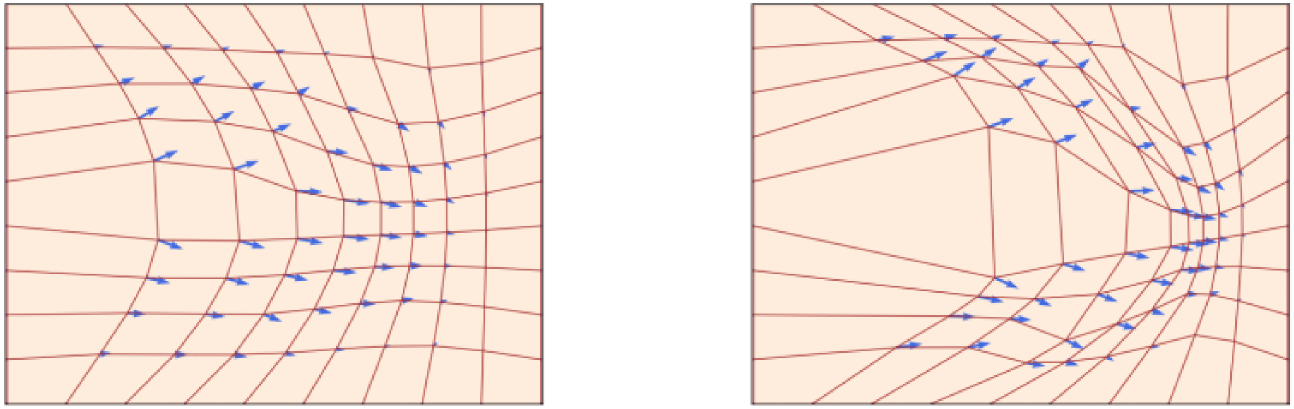
- We need a framework to interact with distributions for statistical generative models.
  - Probabilistic generative models
  - Deep generative models
    - Autoregressive models
    - Variational Autoencoders
    - Generative adversarial networks
    - Energy based models
    - Score based models
    - **Flow Matching**

Ordinary differential equations (ODEs)

- Every ODE is defined by a vector field u, i.e. a function of the form:

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

- i.e. for every time t and location x we get a vector $u_t(x) \in R^d$ specifying a velocity in space



Ordinary differential equations (ODEs)

- An ODE imposes a condition on a trajectory: we want a trajectory X that “follows along the lines” of the vector field u_t, starting at the point x_0. We may formalize such a trajectory as being the solution to the equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}X_t = u_t(X_t) \qquad \blacktriangleright \text{ ODE}$$

$$X_0 = x_0 \qquad \blacktriangleright \text{ initial conditions}$$

Ordinary differential equations (ODEs)

- We may now ask: if we start at $X_0 = x_0$ at $t = 0$, where are we at time t (what is X_t)? This question is answered by a function called the **flow**, which is a solution to the ODE:

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

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x_0) = u_t(\psi_t(x_0)) \qquad \blacktriangleright \text{ flow ODE}$$

$$\psi_0(x_0) = x_0 \qquad \blacktriangleright \text{ flow initial conditions}$$

- **vector fields define ODEs whose solutions are flows.**

Flow models

- We can now construct a generative model via an ODE. Remember that our goal was to convert a simple distribution p_{init} into a complex distribution p_{data}. The simulation of an ODE is thus a natural choice for this transformation. A **flow model** is described by the ODE:

$$X_0 \sim p_{\text{init}} \qquad \blacktriangleright \text{ random initialization}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} X_t = u_t^\theta(X_t) \qquad \blacktriangleright \text{ ODE}$$

- where the vector field u_t^θ is a neural network u_t^θ with parameters θ

$$\text{a continuous function } u_t^\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$$

Flow models

- Our goal is to make the endpoint X_1 of the trajectory have distribution p_{data}, i.e.

$$X_1 \sim p_{\text{data}} \quad \Leftrightarrow \quad \psi_1^\theta(X_0) \sim p_{\text{data}}$$

- where ψ_t^θ describes the flow induced by u_t^θ.
- Note however: although it is called flow model, the neural network parameterizes the vector field, not the flow.

Flow models

- In order to compute the flow, we need to simulate the ODE.
- Following, we summarize the procedure how to sample from 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

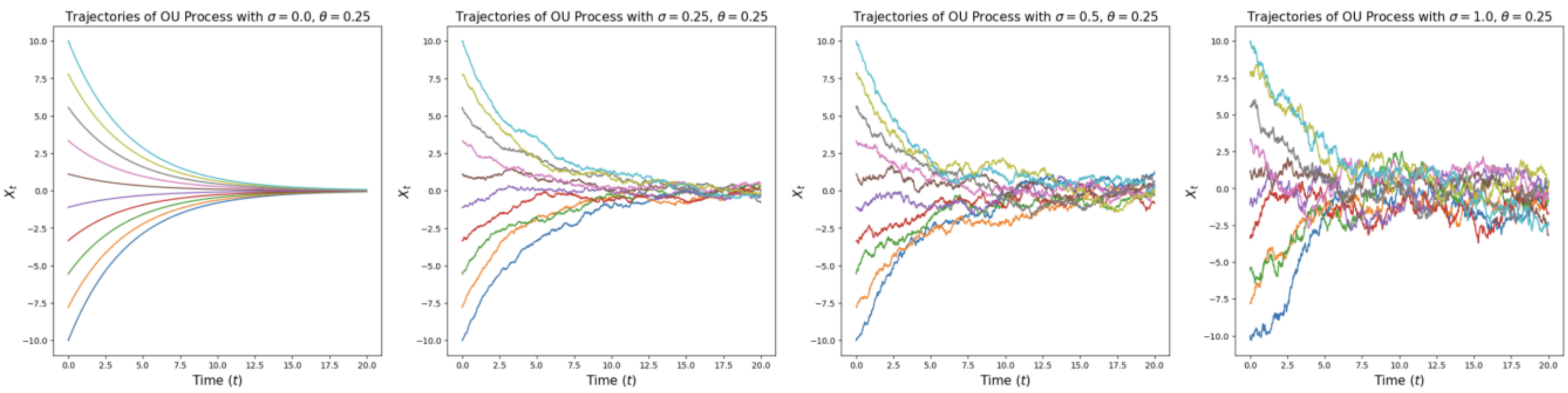
Set $t = 0$
Set step size $h = \frac{1}{n}$
Draw a sample $X_0 \sim p_{\text{init}}$
for $i = 1, \ldots, n$ **do**
  $X_{t+h} = X_t + h u_t^\theta(X_t)$
  Update $t \leftarrow t + h$
end for
return X_1

From ODEs to SDEs

- The idea of an SDE is to extend the deterministic dynamics of an ODE by adding stochastic dynamics driven by a Brownian motion

$$\mathrm{d}X_t = u_t(X_t)\mathrm{d}t + \sigma_t \mathrm{d}W_t$$ ► SDE

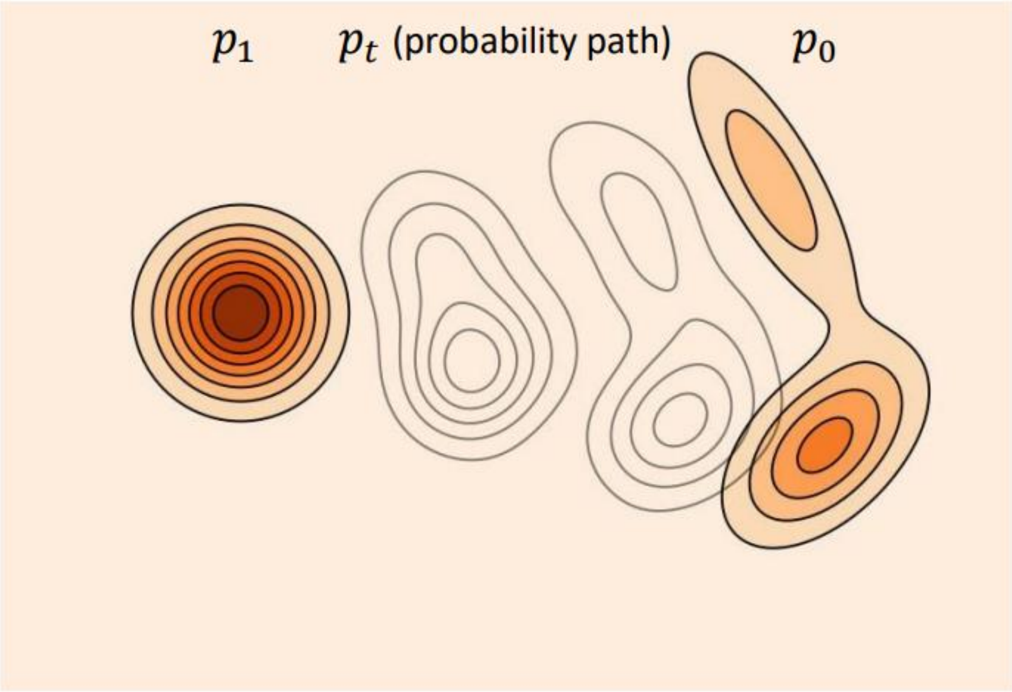
$$X_0 = x_0$$ ► initial condition



Flow matching

- Given a vector field u_t^θ that transports the density p_{init} to p_{data}, we train a NN to regress the vector field:
- However, we do not know the target ☹

$$\mathbb{E}_{t\sim\mathrm{Unif},x\sim p_t}[\|u_t^\theta(x) - u_t^{\mathrm{target}}(x)\|^2]$$



(Conditional) Flow matching

$$\mathbb{E}_{t\sim\text{Unif},z\sim p_{\text{data}},x\sim p_t(\cdot|z)}[\|u_t^\theta(x) - u_t^{\text{target}}(x|z)\|^2]$$

Denoising score matching

$$\mathbb{E}_{\tilde{x}\sim p(\tilde{x}|x),\, x\sim p(x)}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p(\tilde{x})\|^2$$

$$= \mathbb{E}_{\tilde{x}\sim p(\tilde{x}|x),\, x\sim p(x)}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2 + C$$

(Conditional) Flow matching

- Flow matching for Gaussian conditional probability paths

$$\epsilon \sim \mathcal{N}(0, I_d) \quad \Rightarrow \quad x_t = \alpha_t z + \beta_t \epsilon \sim \mathcal{N}(\alpha_t z, \beta_t^2 I_d) = p_t(\cdot|z).$$

- The conditional vector field:

$$u_t^{\text{target}}(x|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right) z + \frac{\dot{\beta}_t}{\beta_t} x,$$

- The conditional flow matching loss:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t\sim\text{Unif}, z\sim p_{\text{data}}, x\sim\mathcal{N}(\alpha_t z, \beta_t^2 I_d)}[\|u_t^\theta(x) - \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right) z - \frac{\dot{\beta}_t}{\beta_t} x\|^2]$$

$$\stackrel{(i)}{=} \mathbb{E}_{t\sim\text{Unif}, z\sim p_{\text{data}}, \epsilon\sim\mathcal{N}(0, I_d)}[\|u_t^\theta(\alpha_t z + \beta_t \epsilon) - (\dot{\alpha}_t z + \dot{\beta}_t \epsilon)\|^2]$$