Presentation Title

Samy Braik

April 30, 2025

We denote by p the target distribution and q an easy-to-sample distribution, for example a centered Gaussian.

Diffusion

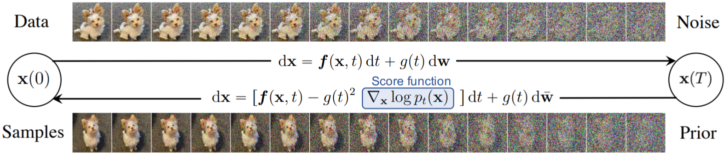
Let $X_0 \sim p$. We want to add noise until we reach pure noise, and denoise it afterward. We choose an horizon of time $T \in \mathbb{N}^*$ and a noise schedule $\beta : [0, T] \to \mathbb{R}^*$, continuous and non decreasing.

Forward process

$$d\overrightarrow{X}_t = \frac{-\beta(t)}{2\sigma^2}\overrightarrow{X}_t dt + \sqrt{\beta(t)}dB_t, \quad \overrightarrow{X}_0 \sim p$$

Backward process

$$d\overleftarrow{X}_t = \left(\frac{\beta(T-t)}{2\sigma^2}\overleftarrow{X}_t + \beta(T-t)\nabla \log p_{T-t}\left(\overleftarrow{X}_t\right)\right)dt + \sqrt{\beta(T-t)}dB_t, \quad \overleftarrow{X}_0 \sim p_T$$



We learn the score by using score-matching techniques

Score matching

$$\mathcal{L}_{\text{score}}(\theta) = \mathbb{E}\left[\left\| s_\theta\left(\tau, \overrightarrow{X}_\tau\right) - \log p_\tau\left(\overrightarrow{X}_\tau | X_0\right)\right\|^2\right]$$

Plug it in the backward process and generate by discretizing the dynamics.

Let $X_0 \sim q$ and $f : \mathbb{R}^d \to \mathbb{R}^d$ an invertible and differentiable function an set $X_1 := f(X_0)$ such that $X_1 \sim p$.
We can write the density of X_1 as

$$p_{X_1} = q(f^{-1}(x_1)) \left| \det \frac{\partial f^{-1}}{\partial x_1}(x_1) \right| \tag{1}$$

$$= p_{X_0}(f^{-1}(x_1)) \left| \det \frac{\partial f}{\partial x_0}(f^{-1}(x_1)) \right|^{-1} \tag{2}$$

We can then write the log-likelihood as

$$\log p_{X_1}(x_1) = \log p_{X_0}(f^{-1}(x_1)) - \log \left| \det \frac{\partial f}{\partial x_0}(f^{-1}(x_1)) \right| \tag{3}$$

Normalizing flow

Let $X_0 \sim q$ and $X_1 \sim p$. We want to learn f_θ such that $X_1 \simeq f_\theta(X_0) = Z \sim p_Z$. To do that, we set a structure on f_θ, with $f_1, \ldots, f_k$ simpler function (all parametrized by θ) such that

$$f_\theta = f_1 \circ f_2 \circ \ldots \circ f_k$$

We determine f_θ by minimizing

$$\mathcal{L}_{\mathrm{NF}}(\theta) = \mathbb{E}\left[-\log p_Z(f_\theta(x)) - \log \left| \det \frac{\partial f_\theta}{\partial x}(x) \right| \right]$$

Flow

A flow is time dependant function $\phi_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. Using that we can define a flow model by applying a flow to (X_0, X_1)

$$X_t = \phi_t(X_0, X_1) \quad t \in [0,1], X_0 \sim p, X_1 \sim q \tag{4}$$

Alternatively, by introducing a velocity field $v_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ the flow can be defined with the following ODE

$$\begin{cases} \partial_t \phi_t(x) & = v_t(\phi_t(x)) \\ \phi_0(x) & = x \end{cases} \tag{5}$$

Flow matching

The framework is as follow :
We define a probability path p_t interpolating from $p_0 = q$ to $p_1 = 1$. Then we learn a velocity field v_t^θ generating the path p_t by minimizing the flow matching loss

$$\mathcal{L}_{\mathrm{FM}}(\theta) := \mathbb{E}[\|v_t^\theta(X_t) - \dot{X}_t\|^2] + c \tag{6}$$

Finally, we can sample from p_1 by solving the ODE (5) with the learned velocity field v_t^θ and the initial condition $X_0 \sim q$.

Comparison

Models	Pros	Cons
Diffusion	1.2	
Normalizing flow	Exact density estimation	Computationaly inte
Flow matching	Exact density estimation Simulation free training	test
Kernel estimator	Flexible Easy to exploit	Slow rate of converg Hard to evaluate at new Hard to choose tuning pa