Diffusion Model and Stochastic Differential Equation

Chuwei Wang

School of Mathematical Sciences
Peking University

2022.10.21

Outline

(i) Background of generative model

(ii) Discrete diffusion model

(iii) Continuous diffusion model

(iv) Diffusion model with Bridge

(v) Summary

Generative Model

Generative Model plays an important role in numerous tasks, such as computer vision and natural language processing.

Figure: 1

Figure: 2

Formulation of generative tasks

- INPUT: Samplings $\{x_i\}$ from an unknown distribution μ.
- OUTPUT: Generated $\{\hat{x}_i\}$ whose distribution $\hat{\mu} \approx \mu$.

Methodology

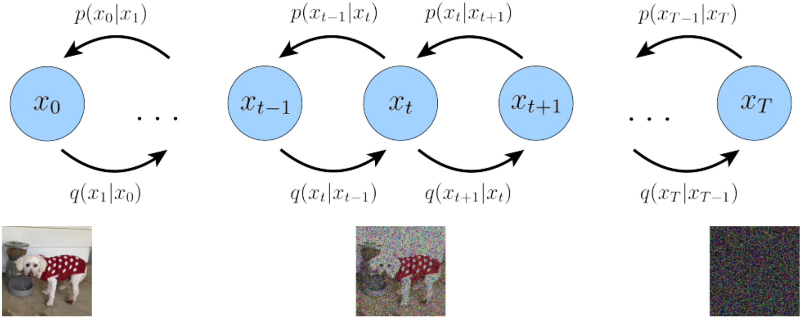
(i) Based on the samplings: to deal with the empirical distribution $\frac{1}{n}\sum_i \delta_{x_i}$

(ii) Based on an known distribution ν(usually Gaussian): try to transform ν into μ.

How to transform ν into μ?

- (straightforward) Look for a mapping F_θ such that the $\mu_{F_\theta(X)} \approx \mu,\ X \sim \nu$.
- Find a sequence of mapping $\{F_\theta^i\}_{i=1}^n$, such that $\mu_{F_\theta^n \circ \ldots \circ F_\theta^1(X)} \approx \mu,\ X \sim \nu$.
- Find a continuous path $F(t)$ in the space of probability measure, such that $F(0) = \mu, F(1) = \nu$.

Discrete diffusion model



x_0 : sampling from the unknown distribution μ (: $p(x)$).

Towards the left: adding noise. $q(x_{t+1}|x_t) = N(\sqrt{\alpha_t}x_t, (1-\alpha_t)I)$

Towards the right: denoising. $p(x_{t-1}|x_t)$ is parameterized and to be learnt.

Training diffusion model

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}_{\text{denoising matching term}}$$

$x_t \sim N(\sqrt{\bar{\alpha}_t}, (1-\bar{\alpha}_t)I)$, $\bar{\alpha}_t := \Pi_{i=1}^{t}\alpha_i$

Parameterized $p_\theta(x_{t-1}|x_t)$ as $N(\mu_\theta(x_t), \sigma(t)^2 I)$, and further write $\mu_\theta(x_t)$ as $C_1(\alpha)x_t + C_2(\alpha)x_\theta(\hat{x}_t, t)$. We have

$$\arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\right\|_2^2\right]$$

Training diffusion model

(i) Sample x_0 from the latent distribution.

(ii) Generate the sequence $\{x_t\}$ by adding gaussian noise.

(iii) Parameterize $x_\theta(x_t, t)$, the estimation of x_0 at the state (x_t, t).

(iv) $argmin_\theta \sum_t \mathbb{E}_{q(x_t|x_0)} \lambda(t) \|x_\theta - x_0\|_2^2$.

Generating

Assumption: $p_{x_T} \approx N(0, I)$.

(i) Generate $z \sim N(0, I)$,

(ii) Generate $\hat{x}$ through the denoising sequence:

$p(z)p_\theta(x_{T-1}|x_T)...p_\theta(x_1|x_2)p_\theta(x_0|x_1)$.

Training diffusion model

Remark

(1) There are other parametrizations, eg. the scope ($\nabla logp(x_t)$) and the source noise, while the loss function remains similar (weighted L^2).

(2) The choice for the hyperparameter $\{\alpha_t\}$:

- Too large: $\{x_t\}$ converges to $N(0, I)$ slowly.
- Too small: Approximating $p(x_{t-1}|x_t)$ with normal distribution would bring much error.

Continuous diffusion model

Published as a conference paper at ICLR 2021

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

Basic Knowledge

- [Def] **Ito Integral**: $\int_0^T f(t)dW_t := \lim_{\Delta\to 0}\sum f(\boldsymbol{t_n})(W_{t_{n+1}} - W_{t_n})$, W_t is standard Brownian Motion in R^d.
- [Def]SDE(Diffusion Process) $dX_t = b(X_t,t)dt + \sigma(X_t,t)dW_t$:
$$X_T - X_0 = \int_{t=0}^{T} b(X_t,t)dt + \int_{t=0}^{T} \sigma(X_t,t)dW_t$$
- $\{X_t\}$ is a stochastic process, b: R^d-valued, σ: $R^{d\times m}$-valued, W_t: standard Brownian Motion in R^m.

Continuous diffusion model

Original diffusion model is the discretization of a continuous version.

Example

$$x_{t+1} = \sqrt{\alpha_t} x_t + \sqrt{1-\alpha_t} w_t, \quad w_t \sim N(0, I) \tag{1}$$

$$\Longleftrightarrow x_{t+1} - x_t = (\sqrt{\alpha_t} - 1) x_t + \sqrt{1-\alpha_t} w_t \tag{2}$$

$$\Longleftrightarrow \Delta x_t = (\sqrt{\alpha_t} - 1) x_t \Delta t + \sqrt{1-\alpha_t} \Delta w_t, \quad \Delta t = 1 \tag{3}$$

$$\to dX_t = (\sqrt{\alpha(t)} - 1) X_t dt + \sqrt{1-\alpha(t)} dW_t. \tag{4}$$

Original diffusion model corresponds to Ornstein – Uhlenbeck process, which converges to normal distribution as $t \to \infty$.

Continuous diffusion model

Theorem

The reverse of the diffusion process characterized by $dx = f(x,t)dt + g(t)dw,\ x(0) \sim p_0$ is the solution to $dx = [f(x,t) - g(t)^2 \nabla log p_t(x)]dt + g(t)d\tilde{w}$, where p_t denotes the marginal distribution of x at time t.

This inspires us to approximate $\nabla log p_t(x)$ with an NN function $s_\theta(x(t),t)$ and reconstruct x_0 from the noise basing on the reverse equation. We could learn s_θ through the following optimization.

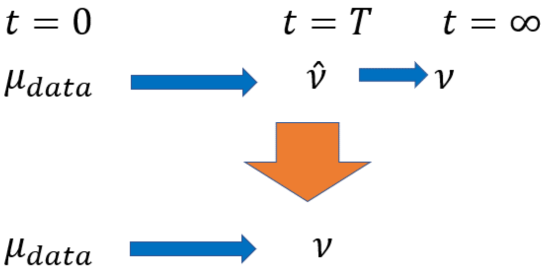
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t \Big\{ \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \Big[\big\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t),t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) \big\|_2^2 \Big] \Big\}$$

Recent Advancement

(1) Apply computational techniques to improve efficiency and accuracy, e.g. high order scheme and adaptive sampling for solving SDE, predictor-corrector scheme for higher accuracy.

(2) Some researcher observe that the marginal probability density $\{p_t(x)\}_{t=0}^{T}$ shares the same trajectory with a deterministic ODE defined as $dx = [f(x,t) - \frac{1}{2}g(t)^2 \nabla log p_t(x)]dt$. Thus, one can avoid simulating SDEs for generating data.

Diffusion model with bridge



In previous SDE, the distribution converges to $N(0, I)$ when $t \to \infty$. Thus, we have to solve the SDE for a large time scale, which is very time-consuming.

Deep Generative Learning via Schrödinger Bridge

Gefei Wang [1] Yuling Jiao [2] Qian Xu [3] Yang Wang [1 4] Can Yang [1 4]

Schrodinger Bridge Problem

Let $\Omega = C([0,1],\mathbb{R}^d)$. $\mathscr{P}(\Omega)$ is the space of probability measure on the path space Ω. P_τ is a Brownian motion.Given $\mu, \nu \in \mathscr{P}(\mathbb{R}^d)$, find $Q^* \in argmin_{\mathscr{P}(\Omega)} \mathbb{D}_{KL}(Q||P_\tau)\ s.t. Q_0 = \mu, Q_1 = \nu$.

Diffusion model with bridge

Fortunately, Schrodinger Bridge Problem has been well-studied in math. Applying the result about the properties of the solution, this work derives a finite-time generative method.

Theorem 3 *Define the density ratio* $f(\mathbf{x}) = \frac{q_\sigma(\mathbf{x})}{\Phi_{\sqrt{\tau}}(\mathbf{x})}$. *Then for the SDE*

$$\mathrm{d}\mathbf{x}_t = \tau \nabla \log \mathbb{E}_{\mathbf{z}\sim\Phi_{\sqrt{\tau}}}[f(\mathbf{x}_t + \sqrt{1-t}\mathbf{z})]\mathrm{d}t + \sqrt{\tau}\mathrm{d}\mathbf{w}_t \quad (4)$$

with initial condition $\mathbf{x}_0 = \mathbf{0}$, *we have* $\mathbf{x}_1 \sim q_\sigma(\mathbf{x})$.

And, for the SDE

$$\mathrm{d}\mathbf{x}_t = \sigma^2 \nabla \log q_{\sqrt{1-t}\sigma}(\mathbf{x}_t)\mathrm{d}t + \sigma \mathrm{d}\mathbf{w}_t \quad (5)$$

with initial condition $\mathbf{x}_0 \sim q_\sigma(\mathbf{x})$, *we have* $\mathbf{x}_1 \sim p_{\mathrm{data}}(\mathbf{x})$.

$\Phi_\sigma(\cdot) : N(0, \sigma^2 I), \; q_\sigma = p_{data} * \Phi_\sigma.$

Diffusion-based Molecule Generation with Informative Prior Bridges

Lemeng Wu*
University of Texas at Austin
lmwu@cs.utexas.edu

Chengyue Gong*
University of Texas at Austin
cygong@cs.utexas.edu

Xingchao Liu
University of Texas at Austin
xcliu@cs.utexas.edu

Mao Ye
University of Texas at Austin
my21@cs.utexas.edu

Qiang Liu
University of Texas at Austin
lqiang@cs.utexas.edu

How to design SDE?

Target

Transform ν into μ. (Create an SDE $dx = f(x,t)dt + g(x,t)dw$ such that $x_{\sim}\nu,\ x_1 \sim \mu.$)

(i) Try to transform ν in to $\delta_{x'}$.

(ii) Every $x \in \mathbb{R}^d$ is related to a transformation (a mapping/trajectory from ν towards δ_x). Take expectation of these transformations over $x \sim \mu$ and derive a transformation from ν to μ.

For (i), get inspiration from the gradient field of Lyapunov function and consider $f(x,t) = -\alpha_t \nabla U(x) + v(x,t)$, where U is a Lyapunov function at x', α_t controls the step size of gradient flow, $v(x,t)$ is a perturbation term.

Summary

- VAE $\rightarrow$ HVAE and diffusion model
- discrete $\rightarrow$ continuous
- infinite time scale $\rightarrow$ finite time convergence

Inspiration for FermiNet

For Diffusion Monte Carlo, a large proportion of iterations play the role of "tending to infinity", which is very time-consuming. It might be helpful if we could design an SDE dynamic such that ψ_I evolves into ψ^* within finite time.

Thanks!