Sampling Methods: From MCMC to Generative Modeling
Generative Modeling - 2 (Diffusion models)

Anna Korba

CREST, ENSAE, Institut Polytechnique de Paris

Outline

Reverse-Time SDE

Training via score matching

Discrete-Time Diffusion Models

Ordinary differential equation

Consider the ordinary differential equation (ODE)

$$\frac{dX_t}{dt} = f(X_t, t)$$

which we also express as

$$dX_t = f(X_t, t)dt$$

where $X_t, f(X_t, t) \in \mathbb{R}^d$. Then, $(X_t)_{t \geq 0}$ is a deterministic curve.

We can think of the ODE as the limit

$$X_{t+\Delta t} \approx X_t + f(X_t, t)\Delta t$$

under $\Delta t \to 0$, where $t = k\Delta t$.

Solution for ODE

$(X_t)_{t=0\dots T}$ solves ODE if it satisfies the

- differential form of the ODE
- or the integral form of the ODE:

$$X_t = X_0 + \int_0^t f(X_s, s)ds$$

Example:

$$\frac{dX_t}{dt} = -X_t, \quad X_0 = 1 \Rightarrow X_t = e^{-t}$$

Stochastic differential equation

Consider the stochastic differential equation (SDE)

$$dX_t = f(X_t, t)dt + g(t)dW_t$$

where $X_t, f(X_t, t) \in \mathbb{R}^d$, $g(t) \in \mathbb{R}^{d \times d}$, and W_t is a d-dimensional Brownian motion or Wiener process.

X_t is a random process. (We can allow g to also depend on X_t, but this makes the equations more complicated.)

We can think of the SDE as the limit

$$X_{t+\Delta t} \approx X_t + f(X_t, t)\Delta t + g(t)Z_k\sqrt{\Delta t}$$

under $\Delta t \to 0$, where $t = k\Delta t$ and $Z_0, Z_1, \cdots \sim \mathcal{N}(0, I)$.

Solution for SDE

$(X_t)_{t=0,\dots T}$ is a solution path for SDE if $(X_t)_{t=0,\dots T}$ is nice[1] with probability distribution defined by

$$X_t = X_0 + \int_0^t f(X_s, s)ds + \int_0^t g(s)dW_s$$

where the Itô stochastic integral is defined as

$$\int_0^t g(s)dW_s = \lim_{\Delta t \to 0} \sum_{k=0}^{K-1} g(k\Delta t)\sqrt{\Delta t} Z_k, \quad Z_1, Z_2, \cdots \sim \mathcal{N}(0, I) \text{ are IID.}$$

[1] right-continuous with left limits (càdlàg)

Solution for SDE

For a given fixed path $(X_t)_{t=0,\dots T}$, we cannot determine whether it was generated as an instance of the SDE. (Given a fixed sequence 00110011, can you determine whether it was generated as 8 independent Bernoulli random variables?)

Rather, we can talk about whether a distribution of paths solve the SDE. A "solution" of an SDE is a probability distribution of $(X_t)_{t=0,\dots T}$ (the joint distribution over all X_t for $t \in [0, T]$).

For diffusion probabilistic models, we will consider a weaker notion: the marginal probability distributions $(p_t)_{t=0,\dots T}$ such that $X_t \sim p_t$ for all $t \in [0, T]$.

A first question of interest is: how does p_t evolve as a function of time t?

Fokker–Planck equation 1D

The time evolution of p_t under the SDE

$$dX_t = f(X_t, t)dt + g(t)dW_t$$

is governed by the Fokker–Planck (FP) equation.

For $d = 1$, the FP equation is

$$\partial_t p_t = -\partial_x(f p_t) + \frac{g^2}{2}\partial_x^2(p_t)$$

More precisely, this means

$$\partial_t p_t(x) = -\partial_x(f(x, t)p_t(x)) + \frac{g^2(t)}{2}\partial_x^2(p_t(x))$$

for all $t > 0$ and $x \in \mathbb{R}$. This is a partial differential equation (PDE).

Derivation of FP equation

Let $d = 1$. Let $\{p_t\}_{t=0}^T$ be a family of pdfs such that $X_t \sim p_t$ for $0 \le t \le T$. For any $\varphi \in \mathcal{C}_c^\infty(\mathbb{R})$ (set of smooth compactly supported functions on $\mathbb{R}$), we have

$$\begin{aligned}
\partial_t \mathbb{E}_{X\sim p_t}[\varphi(X)] &\approx \frac{1}{\varepsilon}\left(\mathbb{E}_{X\sim p_{t+\varepsilon}}[\varphi(X)] - \mathbb{E}_{X\sim p_t}[\varphi(X)]\right) \\
&\approx \frac{1}{\varepsilon}\mathbb{E}_{\substack{X\sim p_t \\ Z\sim\mathcal{N}(0,I)}}\left[\varphi(X+\varepsilon f + \sqrt{\varepsilon} g Z) - \varphi(X)\right] \\
&\approx \frac{1}{\varepsilon}\mathbb{E}_{\substack{X\sim p_t \\ Z\sim\mathcal{N}(0,I)}}\Big[\varphi(X) + \varepsilon\varphi'(X)f(X,t) + \sqrt{\varepsilon}\varphi'(X)g(t)Z \\
&\quad + \frac{1}{2}\varphi''(X)g^2(t)\varepsilon Z^2 + \mathcal{O}(\varepsilon^{3/2}) - \varphi(X)\Big] \\
&\approx \mathbb{E}_{X\sim p_t}\left[\varphi'(X)f(X,t) + \frac{1}{2}\varphi''(X)g^2(t)\right]
\end{aligned}$$

Therefore,

$$\partial_t \int \varphi(x)p_t(x)dx = \int \varphi'(x)f(x,t)p_t(x)dx + \frac{1}{2}\int \varphi''(x)g^2(t)p_t(x)dx$$

$$\int \varphi(x)\partial_t p_t(x)dx = \int \varphi(x)\left(-\partial_x(fp_t) + \frac{1}{2}\partial_x^2(g^2 p_t)\right)dx$$

using integration by parts.

$$\Rightarrow \partial_t p_t = -\partial_x(fp_t) + \frac{g^2}{2}\partial_x^2(p_t)$$

Fokker–Planck equation (multi-dim)

The multi-dimensional Fokker–Planck equation is

$$\begin{aligned}
\partial_t p_t(x) &= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}(f_i(x,t)p_t(x)) + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\left(p_t(x)\sum_{k=1}^{d} g_{ik}(t)g_{jk}(t)\right) \\
&= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}(f_i(x,t)p_t(x)) + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\left(p_t(x) g_{i,:}(t) g_{j,:}^{\top}(t)\right) \\
&= -\nabla_x \cdot (f p_t) + \frac{1}{2}\mathrm{Tr}(g g^{\top} \nabla_x^2 p_t) \\
&= -\nabla_x \cdot (f p_t) + \frac{1}{2}\mathrm{Tr}(g^{\top} \nabla_x^2 p_t g)
\end{aligned}$$

Example SDE: Ornstein–Uhlenbeck process

Example:

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

$$X_t \mid X_0 \sim \mathcal{N}\left(e^{-\beta t}X_0,\ \frac{\sigma^2}{2\beta}(1 - e^{-2\beta t})\right)$$

If $X_0 \sim \mathcal{N}(0, \sigma^2/2\beta)$

$$X_t \sim \mathcal{N}\left(0,\ \frac{\sigma^2}{2\beta}\right)$$

$$p_t(X_t) = \frac{1}{\sqrt{\pi\sigma^2/\beta}} \exp\left[-\frac{\beta}{\sigma^2}(X_t)^2\right]$$

With direct calculations, we can verify that p_t satisfies the FP equation.

$$0 = \partial_t p_t(x) = -\partial_x(f p_t) + \frac{g^2}{2}\partial_x^2(p_t)$$

$$= \partial_x(\beta x p_t(x)) + \frac{\sigma^2}{2}\partial_x^2(p_t(x))$$

Corruption via Ornstein–Uhlenbeck

The Ornstein–Uhlenbeck process

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

with $\beta \geq 0$ and $\sigma > 0$ adds noise to a datapoint X_0. As $T \to \infty$, all information is lost.

Since

$$X_t \mid X_0 \sim \mathcal{N}\left(e^{-\beta t} X_0,\ \frac{\sigma^2}{2\beta}(1 - e^{-2\beta t})I\right),$$

we have X_T is approximately distributed as

$$\mathcal{N}\left(0,\ \frac{\sigma^2}{2\beta} I\right) \text{ if } \beta > 0 \text{ and } T \approx \infty.$$

Question: Sampling $X_T \sim \mathcal{N}\left(0,\ \frac{\sigma^2}{2\beta} I\right)$ is easy. Can we reverse the SDE to sample X_0?

Forward-time ODE: To simulate

$$dX_t = f(X_t, t)dt$$

for $0 < t$, set $X_0 = X(0)$ and compute

$$X_{(k+1)\Delta t} = X_{k\Delta t} + f(X_{k\Delta t}, k\Delta t)\Delta t$$

for sufficiently small Δt and set $t = k\Delta t$.

Reverse-time ODE: To simulate

$$dX_t = f(X_t, t)dt$$

for $0 < t < T$, set $K = \lfloor T/\Delta t \rfloor$ and $X_K = X(T)$ and compute

$$X_{(k-1)\Delta t} = X_{k\Delta t} - f(X_{k\Delta t}, k\Delta t)\Delta t$$

for sufficiently small Δt and set $t = k\Delta t$.
Reversing time for ODEs is easy.
(Mapping from $X(0)$ to $X(T)$ is, after all, a one-to-one map.)

Forward-time SDE: To simulate

$$dX_t = f(X_t, t)dt + g(t)dW_t$$

for $0 < t$, sample $X_0 \sim p_0$ and compute

$$X_{(k+1)\Delta t} = X_{k\Delta t} + f(X_{k\Delta t}, k\Delta t)\Delta t + g(k\Delta t)Z_k\sqrt{\Delta t}$$

for sufficiently small Δt and set $t = k\Delta t$, where $Z_1, Z_2, \cdots \sim \mathcal{N}(0, I)$.

Reverse-time SDE: To simulate

$$dX_t = f(X_t, t)dt + g(t)dW_t$$

for $0 < t < T$, set $X_{\lfloor T/\Delta t \rfloor} = X_T$, and compute

$$X_{(k-1)\Delta t} = X_{k\Delta t} - f(X_{k\Delta t}, k\Delta t)\Delta t - g(k\Delta t)Z_k\sqrt{\Delta t}$$

This does not work!
Rewinding time in SDEs takes more care.

Anderson's reverse-time SDE theorem

Instead, given the forward-time SDE

$$dX_t = f(X_t, t)dt + g(t)dW_t, \qquad X_0 \sim p_0$$

the corresponding *reverse-time* SDE is

$$d\overline{X}_t = \left(f(\overline{X}_t, t) - g^2(t)\nabla_x \log p_t(\overline{X}_t)\right) dt + g(t)d\overline{W}_t, \qquad \overline{X}_T \sim p_T$$

where $\overline{W}_t$ is the reverse-time Brownian motion and p_t is the pdf of X_t defined by the forward-time SDE.

Alternatively (**most common definition**), setting $\overline{X}_{T-t} = Y_t$, we can define $\{Y_t\}_{t=0}^T$ via

$$dY_t = -\left(f(Y_t, T-t) + g^2(T-t)\nabla_x \log p_{T-t}(Y_t)\right) dt + g(T-t)dW_t, \qquad Y_0 \sim p_T$$

(Note that $dW_t \overset{D}{=} -dW_t$.) Then $X_t \overset{D}{=} \overline{X}_t = Y_{T-t}$.

B. D. O. Anderson, *Reverse-time diffusion equation models*, *Stochastic Processes and their Applications*, 1982.

Marginal vs. joint distributions

Note that Anderson's theorem is claiming

$$[X_t \overset{D}{=} \overline{X}_t \text{ for all } 0 \leq t \leq T],$$

which is a weaker statement than

$$\{X_t\}_{t=0}^T \overset{D}{=} \{\overline{X}_t\}_{t=0}^T.$$

The latter

$$\{X_t\}_{t=0}^T \overset{D}{=} \{\overline{X}_t\}_{t=0}^T$$

asserts that the two processes have equal (joint) distributions, while the former

$$[X_t \overset{D}{=} \overline{X}_t \text{ for all } 0 \leq t \leq T]$$

asserts that the marginal distributions are equal for all t.

Diffusion probabilistic models are concerned with the marginal distributions.

Sample generation via SDE

Let $X_0 \sim p_0$, where p_0 corresponds to the data distribution (eg of images of MNIST or ImageNet).

$$dX_t = f\,dt + g\,dW_t, \qquad X_0 \sim p_0$$

Then the forward-time SDE produces $X_T \sim p_T$.

If we can sample $\overline{X}_T \sim p_T$ and run the reverse-time SDE

$$d\overline{X}_t = (f - g^2 \nabla \log p_t(\overline{X}_t))dt + g\,d\overline{W}_t, \qquad \overline{X}_T \sim p_T$$

this would be a generative model producing images $X_0 \sim p_0$.

Sample generation via SDE

Consider the Ornstein–Uhlenbeck forward-time SDE

$$dX_t = -\beta X_t dt + \sigma dW_t, \qquad X_0 \sim p_0$$

Remember that

$$X_t \mid X_0 \sim \mathcal{N}(e^{-\beta t} X_0, \sigma_t^2 I), \qquad \sigma_t^2 = \frac{\sigma^2}{2\beta}(1 - e^{-2\beta t})$$

If T is sufficiently large, $p_T \approx \mathcal{N}(0, \sigma_T^2 I)$.

Consider the reverse-time counterpart

$$d\overline{X}_t = (-\beta \overline{X}_t - \sigma^2 \nabla \log p_t(\overline{X}_t))dt + \sigma d\overline{W}_t, \qquad \overline{X}_T \sim \mathcal{N}(0, \sigma_T^2 I)$$

(It would be better to sample $\overline{X}_T \sim p_T$ exactly, but we do not know p_T because we do not know $p_0 = p_{data}$.)

Sample generation via SDE

Set $K = \lfloor T/\Delta t \rfloor$ and sample $\overline{X}_K \sim \mathcal{N}(0, \sigma_T^2 I)$. Using a standard discretization (Euler–Maruyama), we get:

$$
\begin{aligned}
&\overline{X}_K \sim \mathcal{N}(0, \sigma_T^2 I) \\
&\text{for } k = K, K-1, \ldots, 2, 1 \\
&\quad Z_k \sim \mathcal{N}(0, I) \\
&\quad \overline{X}_{k-1} = \overline{X}_k - \Delta t \left(-\beta \overline{X}_k - \sigma^2 \nabla \log p_{k\Delta t}(\overline{X}_k) \right) + \sigma \sqrt{\Delta t} Z_k \\
&\text{end}
\end{aligned}
$$

The output $\overline{X}_0$ is approximately distributed as p_0.

Interestingly, there is randomness in the generation process.
This is not yet implementable since we do not have access to $\nabla \log p_t$.

Alternative process: Reverse-time ODE

last remark: we could have considered an alternative (deterministic) process called the reverse-time ODE.

Let $\{p_t\}_{t=0}^T$ be the marginal density functions of the forward-time SDE

$$dX_t = f\,dt + g\,dW_t, \qquad X_0 \sim p_0$$

and reverse-time SDE

$$d\overline{X}_t = \Big(f(\overline{X}_t, t) - g^2(t)\nabla \log p_t(\overline{X}_t)\Big)\,dt + g(t)\,d\overline{W}_t, \qquad \overline{X}_T \sim p_T$$

Then, $\{p_t\}_{t=0}^T$ is also the marginal density function of the following reverse-time ODE

$$d\overline{X}_t = \left(f(\overline{X}_t, t) - \frac{g^2(t)}{2}\nabla \log p_t(\overline{X}_t)\right)dt, \qquad \overline{X}_T \sim p_T$$

This ODE defines a flow model, a one-to-one mapping between $\overline{X}_T$ and $\overline{X}_0$.

we can follow a similar strategy for sampling with the deterministic process ("probability flow ode").

Outline

Reverse-Time SDE

Training via score matching

Discrete-Time Diffusion Models

Practical reverse-time SDE

Simulating the reverse-time SDE

$$d\overline{X}_t = \left(f - g^2 \nabla \log p_t\right) dt + g\, d\overline{W}_t, \qquad \overline{X}_T \sim p_T$$

requires (i) sampling from p_T and (ii) evaluating of the *score function* $\nabla_x \log p_t$.

Solution:

(i) Design forward-time SDE, i.e., choose f, g, T, so that $p_T \approx \mathcal{N}(0, \sigma_T^2 I)$ and σ_T^2 is known (ex: OU process).

(ii) Learn $\nabla_x \log p_t(x) \approx s_\theta(x, t)$ via a neural network $s_\theta(x, t)$.
We call $s_\theta(x, t)$ the **score network**.

VE and VP forward SDEs

Two types of processes are primarily considered for the forward SDE.

First, variance-exploding (VE)

$$dX_t = \sigma dW_t \qquad \gamma_t = 1 \qquad \sigma_t^2 = t\sigma^2$$

$$X_t \mid X_0 \sim \mathcal{N}(\gamma_t X_0, \sigma_t^2 I)$$

Although the mean is preserved, the variance explodes (if σ_t explodes).

Relative to the noise, the original signal X_0 is corrupted as $t \to \infty$.

Second, variance-preserving (VP)

$$dX_t = -\beta X_t dt + \sigma dW_t \quad \gamma_t = e^{-\beta t} \quad \sigma_t^2 = \frac{\sigma^2}{2\beta}(1-e^{-2\beta t}), \quad X_t \mid X_0 \sim \mathcal{N}(\gamma_t X_0, \sigma_t^2 I)$$

In particular,

$$\mathrm{Var}(X_t) = I + e^{-\beta t}(\mathrm{Var}(X_0) - I)$$

and if $\mathrm{Var}(X_0) = I$, then

$$\mathrm{Var}(X_t) = I$$

So variance is "preserved".

In both cases,

$$X_t \overset{D}{=} \gamma_t X_0 + \sigma_t \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I)$$

Score matching

To learn the score function, consider

$$\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_t}\left[\|s_\theta(X_t, t) - \nabla_x \log p_t(X_t)\|^2\right] dt$$

where $\lambda(t) > 0$ is a weighing factor. However, we cannot use this as is, since p_t is inaccessible.

Alternatively, use the equivalent losses:

1.

$$\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_0}\left[\mathbb{E}_{X_t|X_0}\left[\left\|s_\theta(X_t, t) - \nabla_x \log p_{t|0}(X_t \mid X_0)\right\|^2 \middle| X_0\right]\right] dt + C$$

2.

$$\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_t}\left[\|s_\theta(X_t, t)\|^2 + 2\mathbb{E}_\nu\left[\frac{d}{dh}\nu^\top s_\theta(X_t + h\nu, t)\Big|_{h=0}\right]\right] dt + C$$

where C are constants independent of θ.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, ICLR 2021.

proof (1) $\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_0}\left[\mathbb{E}_{X_t|X_0}\left[\|s_\theta(X_t,t) - \nabla_{X_t}\log p_{t|0}(X_t|X_0)\|^2 \middle| X_0\right]\right] dt + C$

The replacement of $\nabla_{X_t}\log p_t(X_t)$ with $\nabla_{X_t}\log p_{t|0}(X_t|X_0)$ requires justification.

$$
\begin{aligned}
\nabla_{X_t}\log p_t(X_t) &= \frac{\nabla_{X_t} p_t(X_t)}{p_t(X_t)} \\
&= \frac{1}{p_t(X_t)}\nabla_{X_t}\int_{\mathbb{R}^d} p_{t|0}(X_t|X_0)p_0(X_0)\,dX_0 \\
&= \int_{\mathbb{R}^d}\left(\nabla_{X_t} p_{t|0}(X_t|X_0)\right)\frac{p_0(X_0)}{p_t(X_t)}\,dX_0 \\
&= \int_{\mathbb{R}^d}\left(\nabla_{X_t}\log p_{t|0}(X_t|X_0)\right)\frac{p_{t|0}(X_t|X_0)p_0(X_0)}{p_t(X_t)}\,dX_0 \\
&= \int_{\mathbb{R}^d}\left(\nabla_{X_t}\log p_{t|0}(X_t|X_0)\right)p_{0|t}(X_0|X_t)\,dX_0 \\
&= \mathbb{E}_{X_0|X_t}\left[\nabla_{X_t}\log p_{t|0}(X_t|X_0)\middle|X_t\right]
\end{aligned}
$$

P. Vincent, *A connection between score matching and denoising autoencoders*, Neural Computation, 2011.

proof (1) $\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_0}\left[\mathbb{E}_{X_t|X_0}\left[\|s_\theta(X_t,t) - \nabla_{X_t}\log p_{t|0}(X_t|X_0)\|^2 \middle| X_0\right]\right] dt + C$

The replacement of $\nabla_{X_t}\log p_t(X_t)$ with $\nabla_{X_t}\log p_{t|0}(X_t|X_0)$ requires justification.

$$
\begin{aligned}
\mathcal{L}(\theta) &= \int_0^T \lambda(t)\mathbb{E}_{X_t}\left[\|s_\theta(X_t,t) - \nabla_{X_t}\log p_t(X_t)\|^2\right] dt \\
&= \int_0^T \lambda(t)\mathbb{E}_{X_t}\left[\|s_\theta(X_t,t)\|^2 - 2\langle s_\theta, \nabla_{X_t}\log p_t\rangle\right] dt + C \\
&= \int_0^T \lambda(t)\mathbb{E}_{X_t}\left[\|s_\theta\|^2 - 2\langle s_\theta, \mathbb{E}_{X_0|X_t}[\nabla \log p_{t|0}]\rangle\right] dt + C \\
&= \int_0^T \lambda(t)\mathbb{E}_{X_t,X_0}\left[\|s_\theta - \nabla \log p_{t|0}\|^2\right] dt + C
\end{aligned}
$$

Called denoising score matching (DSM).

P. Vincent, *A connection between score matching and denoising autoencoders*, Neural Computation, 2011.

proof (1) $\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_0}\left[\mathbb{E}_{X_t|X_0}\left[\|s_\theta(X_t,t) - \nabla_{X_t}\log p_{t|0}(X_t|X_0)\|^2 \middle| X_0\right]\right] dt$

Conditional score function $\nabla_{X_t}\log p_{t|0}(X_t|X_0)$ is implementable if f and g are nice.

Ornstein–Uhlenbeck process is one such example.

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

$$p_{t|0}(X_t|X_0) \sim \mathcal{N}(e^{-\beta t}X_0, \sigma_t^2 I), \qquad \sigma_t^2 = \frac{\sigma^2}{2\beta}(1 - e^{-2\beta t})$$

$$\begin{aligned}\nabla_{X_t}\log p_{t|0}(X_t|X_0) &= \frac{1}{\sigma_t^2}(X_t - e^{-\beta t}X_0) \\ &= \frac{2\beta}{\sigma^2(1-e^{-2\beta t})}(X_t - e^{-\beta t}X_0)\end{aligned}$$

Hutchinson's trace estimator

Let $\nu \in \mathbb{R}^n$ be a random vector such that

$$\mathbb{E}_\nu[\nu_i \nu_j] = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

i.e., $\mathbb{E}_\nu[\nu\nu^\top] = I \in \mathbb{R}^{n \times n}$.

One example is $\nu_1, \ldots, \nu_n \sim \mathcal{N}(0, 1)$ IID Gaussian.

Another example is $\nu_1, \ldots, \nu_n$ drawn as IID Rademacher (± 1 realization with probability $1/2$) random variables.

M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, *Communications in Statistics - Simulation and Computation*, 1990.

Hutchinson's trace estimator

Let $A \in \mathbb{R}^{n \times n}$. Then

$$\begin{aligned}
\mathbb{E}_\nu[\nu^\top A \nu] &= \mathbb{E}_\nu[\mathrm{Tr}(\nu^\top A \nu)] \\
&= \mathbb{E}_\nu[\mathrm{Tr}(A \nu \nu^\top)] \\
&= \mathrm{Tr}(\mathbb{E}_\nu[A \nu \nu^\top]) \\
&= \mathrm{Tr}(A \mathbb{E}_\nu[\nu \nu^\top]) \\
&= \mathrm{Tr}(AI) \\
&= \mathrm{Tr}(A)
\end{aligned}$$

So $\nu^\top A \nu$ serves as an unbiased estimator of $\mathrm{Tr}(A)$.

M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, *Communications in Statistics - Simulation and Computation*, 1990.

proof

(2) $\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_t}\left[\|s_\theta(X_t,t)\|^2 + 2\mathbb{E}_\nu\left[\frac{d}{dh}\nu^\top s_\theta(X_t+h\nu,t)\Big|_{h=0}\right]\right]dt + C$

$$
\begin{aligned}
&-\mathbb{E}_{X_t}\left[\langle s_\theta(X_t,t), \nabla_{X_t}\log p_t(X_t)\rangle\right] = -\int\left\langle s_\theta(x,t), \frac{\nabla_x p_t(x)}{p_t(x)}\right\rangle p_t(x)dx \\
&= -\int \langle s_\theta(x,t), \nabla_x p_t(x)\rangle\, dx \\
&= \int (\nabla\cdot s_\theta(x,t))\, p_t(x)dx \\
&= \mathbb{E}_{X_t\sim p_t}\left[\nabla_{X_t}\cdot s_\theta(X_t,t)\right] \\
&= \mathbb{E}_{X_t}\left[\mathrm{Tr}(D_{X_t}s_\theta(X_t,t))\right] \\
&= \mathbb{E}_{X_t}\mathbb{E}_\nu\left[\nu^\top D_{X_t}s_\theta(X_t,t)\nu\right] \\
&= \mathbb{E}_{X_t}\mathbb{E}_\nu\left[\frac{d}{dh}\nu^\top s_\theta(X_t+h\nu,t)\Big|_{h=0}\right]
\end{aligned}
$$

where we use integration by parts and the Hutchinson estimator.
Called *sliced score matching (SSM)*.

Training with O-U and DSM

Using $X_t \overset{D}{=} \gamma_t X_0 + \sigma_t \varepsilon$, the score function simplifies to

$$\nabla_{X_t} \log p_t(X_t \mid X_0) = \frac{\gamma_t X_0 - X_t}{\sigma_t^2} \overset{D}{=} -\frac{\varepsilon}{\sigma_t}$$

Define the **scaled score network**

$$\varepsilon_\theta(X_t, t) = -\sigma_t s_\theta(X_t, t)$$

Then the denoising score matching loss becomes

$$\begin{aligned}
\mathcal{L}(\theta) &= \int_0^T \lambda(t) \mathbb{E}_{X_0} \left[\mathbb{E}_{X_t \mid X_0} \left[\left\| s_\theta(X_t, t) - \nabla_{X_t} \log p_{t|0}(X_t \mid X_0) \right\|^2 \right] \right] dt \\
&= \int_0^T \frac{\lambda(t)}{\sigma_t^2} \mathbb{E}_{X_0} \left[\mathbb{E}_{\varepsilon \sim \mathcal{N}(0,I)} \left[\left\| \varepsilon_\theta(\gamma_t X_0 + \sigma_t \varepsilon, t) - \varepsilon \right\|^2 \right] \right] dt \\
&= T \, \mathbb{E}_{\substack{X_0 \sim p_0 \\ t \sim \mathrm{Uniform}([0,T]) \\ \varepsilon \sim \mathcal{N}(0,I)}} \left[\frac{\lambda(t)}{\sigma_t^2} \left\| \varepsilon_\theta(\gamma_t X_0 + \sigma_t \varepsilon, t) - \varepsilon \right\|^2 \right]
\end{aligned}$$

Interpretation: $\varepsilon_\theta(X_t, t)$ predicts noise ε from noised data $X_t \overset{D}{=} \gamma_t X_0 + \sigma_t \varepsilon$.

Training with O-U and DSM

Using the Ornstein–Uhlenbeck forward SDE and the denoising score matching loss (DSM), we get the training algorithm:

while (not converged)

$$X_0 \sim p_0 = p_{\text{data}}$$

$$t \sim \text{Uniform}([0, T])$$

$$\varepsilon \sim \mathcal{N}(0, I)$$

$$X_t = \gamma_t X_0 + \sigma_t \varepsilon$$

Call optimizer with $\frac{\lambda(t)}{\sigma_t^2} \nabla_\theta \left\| \varepsilon_\theta(X_t, t) - \varepsilon \right\|^2$

end

Blow-up at $t = 0$

For both VP and VE SDEs, $\sigma_0 = 0$ and the loss blows up. Several ways to deal with this.

Option 1: Start the integral from a small $\delta > 0$

$$\mathcal{L}(\theta) = \int_{\delta}^{T} \frac{\lambda(t)}{\sigma_t^2} \mathbb{E}_{\substack{x_0 \sim p_0 \\ \varepsilon \sim \mathcal{N}(0,I)}} \left\| \varepsilon_\theta(\gamma_t x_0 - \sigma_t \varepsilon, t) - \varepsilon \right\|^2 dt$$

Option 2: Choose $\lambda(t) \to 0$ as $t \to 0$ so that $\lambda(t)/\sigma_t^2$ does not blow up. This makes the mean well-behaved, but the variance of the stochastic gradients may still blow up as $t \to 0$.

Training with SSM

Using the sliced score matching loss (SSM)

$$\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{X_t}\left[\|s_\theta(X_t,t)\|^2 + 2\mathbb{E}_\nu\left[\frac{d}{dh}\nu^\top s_\theta(X_t+h\nu,t)\Big|_{h=0}\right]\right]dt$$

We get the training routine:

```
while (not converged)
  t ~ Uniform([0, T])
  X_t ~ p_t                          # forward-simulate SDE from X_0 ~ p_data
  ν ~ p_ν                            # E_{ν~p_ν}[νν^⊤] = I
  Backprop on h with (d/dh) ν^⊤ s_θ(X_t + hν, t)|_{h=0}
  Call optimizer with λ(t)∇_θ ( ||s_θ(X_t, t)||^2 + 2 (d/dh) ν^⊤ s_θ(X_t + hν, t)|_{h=0} )
end
```

DSM vs SSM

- SSM is more broadly applicable than DSM.
  - SSM requires efficient sampling of X_t given X_0.
  - DSM additionally requires evaluation of conditional density $p_{t|0}(X_t \mid X_0)$. (More precisely, the conditional score $\nabla_{X_T} \log p_{t|0}(X_t \mid X_0)$ is required.)
- SSM allows a broader range of forward-diffusions to be used. Useful in, say, DSB.[1]
- When applicable, DSM performs better than SSM.
- SSM requires mixed (2nd-order) derivatives, while DSM requires 1st-order derivatives.
  (Most modern DL libraries are capable of efficiently computing higher-order derivatives.)

[1] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, *Diffusion Schrödinger bridge with applications to score-based generative modeling*, NeurIPS, 2021.

SDE Sampling with trained score

Once s_θ has been trained, we can generate new samples with the approximate reverse-time SDE

$$d\overline{X}_t = \Big(f(\overline{X}_t, t) - g^2(t) s_\theta(\overline{X}_t, t)\Big)\, dt + g(t) d\overline{W}_t, \qquad \overline{X}_T \sim \mathcal{N}(0, \sigma_T^2 I)$$

Usually, one uses the reverse-time Ornstein–Uhlenbeck process

$$\begin{aligned} d\overline{X}_t &= \Big(-\beta \overline{X}_t - \sigma^2 s_\theta(\overline{X}_t, t)\Big)\, dt + \sigma d\overline{W}_t, \qquad \overline{X}_T \sim \mathcal{N}(0, \sigma_T^2 I) \\ &= \left(\frac{\sigma^2}{\sigma_t} \varepsilon_\theta(\overline{X}_t, t) - \beta \overline{X}_t\right) dt + \sigma d\overline{W}_t \end{aligned}$$

Using a standard discretization (Euler–Maruyama), we get

$\overline{X}_K \sim \mathcal{N}(0, \sigma_T^2 I)$

for $k = K, K-1, \ldots, 2, 1$

$$\overline{X}_{k-1} = \overline{X}_k - \Delta t \left(\frac{\sigma^2}{\sigma_t} \varepsilon_\theta(\overline{X}_k, k\Delta t) - \beta \overline{X}_k\right) + \sigma \sqrt{\Delta t} Z_k, \qquad Z_k \sim \mathcal{N}(0, I)$$

`end`

The output $\overline{X}_0$ is approximately distributed as p_0.
Called **DDPM sampling** for reasons to be explained later.

Samples via SDE

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, ICLR, 2021.

ODE Sampling with trained score

Once s_θ has been trained, we can also use approximate reverse-time ODE

$$d\overline{X}_t = \left(f(\overline{X}_t, t) - \frac{g^2(t)}{2} s_\theta(\overline{X}_t, t) \right) dt, \qquad \overline{X}_T \sim \mathcal{N}(0, \sigma_T^2 I)$$

Usually, one uses the reverse-time ODE of Ornstein–Uhlenbeck process

$$\begin{aligned} d\overline{X}_t &= \left(-\beta \overline{X}_t - \frac{\sigma^2}{2} s_\theta(\overline{X}_t, t) \right) dt, \qquad \overline{X}_T \sim \mathcal{N}(0, \sigma_T^2 I) \\ &= \left(\frac{\sigma^2}{2\sigma_t} \varepsilon_\theta(\overline{X}_t, t) - \beta \overline{X}_t \right) dt \end{aligned}$$

Using an Euler discretization we get

$\overline{X}_K \sim \mathcal{N}(0, \sigma_T^2 I)$

for $k = K, K-1, \ldots, 2, 1$

$$\overline{X}_{k-1} = \overline{X}_k - \Delta t \left(\frac{\sigma^2}{2\sigma_t} \varepsilon_\theta(\overline{X}_k, k\Delta t) - \beta \overline{X}_k \right)$$

`end`

The output $\overline{X}_0$ is approximately distributed as p_0.
This is called **DDIM sampling** for reasons to be explained later.

SDE vs ODE sampling

SDE sampling produces higher fidelity (based on visual inspection) images. Why?
Theoretically, not understood well. Intuitively, noise steps of SDE sampling correct for any errors from inaccurate terminal distribution p_T, inaccurate score function, and discretization.

However, ODE sampling is useful for applications such as image interpolation, which can be used for image editing (more on this later), and for likelihood computation (based on the observation that the ODE sampling defines a flow model).

Outline

Reverse-Time SDE

Training via score matching

Discrete-Time Diffusion Models

Discrete- to continuous-time diffusion

Publication dates:

- DDPM (NeurIPS 20)
- DDIM (ICLR 21)
- SDE Diffusion (ICLR 21)

After the dust settled, people now understand that

- DDPM is a discretization of SDE sampling of VP SDE.
- DDIM is a discretization of ODE sampling of VP SDE. (One specific instance of DDIM.)

Tweedie's formula: 1st order

Consider the random variable

$$Y = X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

(We don't assume p_X is Gaussian.) Then,

$$\mathbb{E}[X \mid Y] = Y + \sigma^2 \nabla_Y \log p_Y(Y)$$

Tweedie's formula: 1st order

Consider the random variable

$$Y = X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

(We don't assume p_X is Gaussian.) Then,

$$\mathbb{E}[X \mid Y] = Y + \sigma^2 \nabla_Y \log p_Y(Y)$$

Proof: Y has a density given by:

$$p_Y(dy) = \int p_X(x)\, \rho_\sigma(y - x)\, dx$$

where $\rho_\sigma(z) \propto \exp\left(-\frac{\|z\|^2}{2\sigma^2}\right)$ is a centered Gaussian with variance σ^2. It follows

$$\frac{\mathbb{E}[X \mid Y = y] - y}{\sigma^2} = \frac{\int \left(\frac{x-y}{\sigma^2}\right) p_X(x) \rho_\sigma(y - x)\, dx}{\int p_X(x) \rho_\sigma(y - x)\, dx} = \nabla_y \log \left\{ \int p_X(x) \rho_\sigma(y - x)\, dx \right\}.$$

Tweedie's formula: 1st order

Consider the random variable

$$Y = X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

(We don't assume p_X is Gaussian.) Then,

$$\mathbb{E}[X \mid Y] = Y + \sigma^2 \nabla_Y \log p_Y(Y)$$

Proof: Y has a density given by:

$$p_Y(dy) = \int p_X(x)\, \rho_\sigma(y - x)\, dx$$

where $\rho_\sigma(z) \propto \exp\left(-\frac{\|z\|^2}{2\sigma^2}\right)$ is a centered Gaussian with variance σ^2. It follows

$$\frac{\mathbb{E}[X \mid Y = y] - y}{\sigma^2} = \frac{\int \left(\frac{x-y}{\sigma^2}\right) p_X(x)\rho_\sigma(y-x)\, dx}{\int p_X(x)\rho_\sigma(y-x)\, dx} = \nabla_y \log \left\{ \int p_X(x)\rho_\sigma(y-x)\, dx \right\}.$$

If

$$Y = \gamma X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

with $\gamma \neq 0$, then

$$\mathbb{E}[X \mid Y] = \frac{1}{\gamma}\mathbb{E}[\gamma X \mid Y] = \frac{1}{\gamma}\left(Y + \sigma^2 \nabla_Y \log p_Y(Y)\right)$$

B. Efron, Tweedie's formula and selection bias, *Journal of the American Statistical Association*, 2012.

Tweedie's formula: 2nd order

Consider the random variable

$$Y = X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

(We don't assume p_X is Gaussian.) Then,

$$\mathrm{Var}[X \mid Y] = \sigma^2 I + \sigma^4 \nabla_Y^2 \log p_Y(Y)$$

Tweedie's formula: 2nd order

Consider the random variable

$$Y = X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

(We don't assume p_X is Gaussian.) Then,

$$\mathrm{Var}[X \mid Y] = \sigma^2 I + \sigma^4 \nabla_Y^2 \log p_Y(Y)$$

If

$$Y = \gamma X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

with $\gamma \neq 0$, then

$$\mathrm{Var}[X \mid Y] = \frac{\sigma^2}{\gamma^2} \left(I + \sigma^2 \nabla_Y^2 \log p_Y(Y) \right)$$

B. Efron, Tweedie's formula and selection bias, *Journal of the American Statistical Association*, 2012.

Reverse cond. distribution ≈ Gaussian

Consider the random variable

$$Y = X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

By definition, $p_{Y|X} = \mathcal{N}(X, \sigma^2 I)$ is Gaussian. (We don't assume p_X is Gaussian.) In general, $p_{X|Y}$ is not a Gaussian, but $p_{X|Y}$ is approximately Gaussian in the limit of $\sigma \to 0$.

$$p_{X|Y}(x \mid y) \approx \mathcal{N}\left(y + \sigma^2 \nabla \log p_Y(y),\ \sigma^2 I\right)$$

If

$$Y = \gamma X + \sigma Z, \quad X \sim p_X, \quad Z \sim \mathcal{N}(0, I)$$

with $\gamma \neq 0$, then, in the limit of $\sigma \to 0$,

$$p_{X|Y}(x \mid y) \approx \mathcal{N}\left(\frac{1}{\gamma}(y + \sigma^2 \nabla \log p_Y(y)),\ \frac{\sigma^2}{\gamma^2} I\right)$$

Reverse cond. distribution $\approx$ Gaussian

$$
\begin{aligned}
p_{X|Y}(x \mid y) &= \frac{p_{Y|X}(y \mid x)\, p_X(x)}{p_Y(y)} \\
&= \frac{\frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) p_X(x)}{\int_{\mathbb{R}^d} \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) p_X(x)\, dx} \\
&= \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) \left(p_X(y) + \langle \nabla p_X(y), x-y\rangle + O(\|x-y\|^2)\right) \\
&= \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) \left(\mathbb{E}_{x\sim\mathcal{N}(y,\sigma I)}\left[p_X(y) + \langle \nabla p_X(y), x-y\rangle + O(\|x-y\|^2)\right]\right. \\
&= \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) \left(p_X(y) + \langle \nabla p_X(y), x-y\rangle + O(\|x-y\|^2)\right) \\
&= \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) \left(\frac{p_X(y) + \langle \nabla p_X(y), x-y\rangle + O(\|x-y\|^2)}{p_X(y) + 0 + O(\sigma^2)}\right) \\
&= \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) (1 + \langle \nabla \log p_X(y), x-y\rangle + \text{h.o.t.}) \\
&= \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-x\|^2\right) \exp\left(\langle \nabla \log p_X(y), x-y\rangle\right) + \text{h.o.t.} \\
&= \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|x-y-\sigma^2\nabla \log p_X(y)\|^2 + \text{h.o.t.}\right) \\
&\approx \mathcal{N}\left(y + \sigma^2 \nabla \log p_Y(y),\ \sigma^2 I\right)
\end{aligned}
$$

DDPM

Forward model: $X_0 \sim p_0 = p_{\text{data}}$

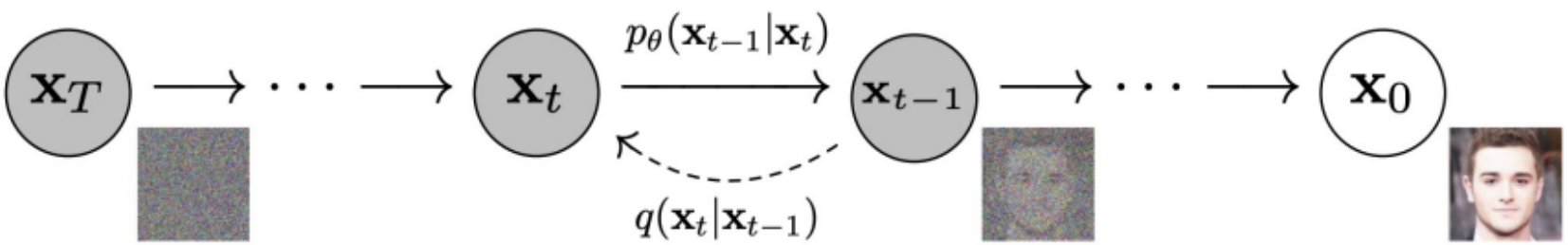
$$X_t \mid X_{t-1} \sim \mathcal{N}\left(\sqrt{1-\beta_t}X_{t-1},\, \beta_t I\right) \quad \text{for} \quad t = 1, \ldots, T \quad (0 < \beta_t < 1)$$

So,

$$X_t \stackrel{\mathcal{D}}{=} \sqrt{1-\beta_t}X_{t-1} + \sqrt{\beta_t}Z_t, \quad Z_t \sim \mathcal{N}(0, I), \quad \text{for} \quad t = 1, \ldots, T$$

and, after some calculations, this implies

$$X_t \mid X_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}X_0,\, (1-\bar{\alpha}_t)I\right), \qquad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$$



J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, *NeurIPS*, 2020.

DDPM (Denoising Diffusion Probabilistic Models)

Reverse model:

$$\text{True probabilitiy:} \quad p(X_{t-1} \mid X_t) \approx \mathcal{N}(\mu(X_t, t),\, \beta_t I) (\text{for small } \beta_t)$$

$$\text{Learned:} \quad p_\theta(X_{t-1} \mid X_t) = \mathcal{N}(\mu_\theta(X_t, t),\, \tilde{\beta}_t I)$$

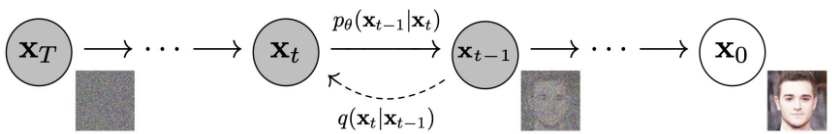
where:

$$\mu(X_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(X_t + \beta_t \nabla \log p_t(X_t)\right)$$

$$\mu_\theta(X_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(X_t + \beta_t s_\theta(X_t, t)\right)$$

$$\tilde{\beta}_t = \begin{cases} \beta_t \text{ or} \\ \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \end{cases}$$

Note, for small β_t $\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t = \beta_t + \text{h.o.t.}$



J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, *NeurIPS*, 2020.

DDPM loss

$$
\begin{aligned}
\mathcal{L}(\theta) &= \sum_{t=1}^{T} \lambda_t \mathbb{E}_{X_t}\left[\|\mu(X_t,t) - \mu_\theta(X_t,t)\|^2\right] \\
&= \sum_{t=1}^{T} \frac{\lambda_t \beta_t^2}{1-\beta_t} \mathbb{E}_{X_t}\left[\|\nabla_{X_t} \log p_t(X_t) - s_\theta(X_t,t)\|^2\right] \\
&= \sum_{t=1}^{T} \frac{\lambda_t \beta_t^2}{1-\beta_t} \mathbb{E}_{X_0,X_t}\left[\left\|\nabla_{X_t} \log p_{t|0}(X_t \mid X_0) - s_\theta(X_t,t)\right\|^2\right] + C \\
&= \sum_{t=1}^{T} \frac{\lambda_t \beta_t^2}{1-\beta_t} \mathbb{E}_{X_0,X_t}\left[\left\|\frac{1}{1-\bar{\alpha}_t}(X_t - \sqrt{\bar{\alpha}_t}X_0) - s_\theta(X_t,t)\right\|^2\right] + C \\
&= \sum_{t=1}^{T} \frac{\lambda_t \beta_t^2}{(1-\beta_t)(1-\bar{\alpha}_t)} \mathbb{E}_{\substack{X_0\sim p_{\text{data}} \\ \varepsilon\sim\mathcal{N}(0,I)}}\left[\left\|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t)\right\|^2\right] + C \\
&= \sum_{t=1}^{T} \tilde{\lambda}_t \mathbb{E}_{\substack{X_0\sim p_{\text{data}} \\ \varepsilon\sim\mathcal{N}(0,I)}}\left[\left\|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t)\right\|^2\right] + C
\end{aligned}
$$

$$
X_t \overset{\mathcal{D}}{=} \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0,I), \quad \varepsilon_\theta \triangleq -\sqrt{1-\bar{\alpha}_t}\, s_\theta, \quad \tilde{\lambda}_t = \frac{\lambda_t \beta_t^2}{(1-\beta_t)(1-\bar{\alpha}_t)}
$$

DDPM training

The training of DDPM is hence analogous to the continuous-time (SDE) setup we have already seen.

while(not converged)

$X_0 \sim p_0 = p_{\text{data}}$

$t \sim \text{Uniform}(\{1, \dots, T\})$

$\varepsilon \sim \mathcal{N}(0, I)$

$X_t = \sqrt{\bar{\alpha}_t} X_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon$

Call optimizer with $\tilde{\lambda}_t \nabla_\theta \left\| \varepsilon_\theta(X_t, t) - \varepsilon \right\|^2$

end

DDPM sampling

The true distribution of X_T is

$$X_T \mid X_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_T} X_0, (1-\bar{\alpha}_T) I\right) \qquad \bar{\alpha}_T = \prod_{s=1}^{T} (1-\beta_s)$$

If T and $\beta_1, \ldots, \beta_T$ are chosen such that $\bar{\alpha}_T \approx 0$, then $p_T \approx \mathcal{N}(0, I)$.

Sampling from the learned distribution can be done as follows:

$$p_\theta(X_{t-1} \mid X_t) = \mathcal{N}(\mu_\theta(X_t, t),\, \tilde{\beta}_t^2 I) \qquad \mu_\theta(X_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(X_t + \beta_t s_\theta(X_t, t)\right)$$

$\overline{X}_T \sim \mathcal{N}(0, I)$

for $t = T, T-1, \ldots, 2, 1$

$$\overline{X}_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \left(\overline{X}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta(\overline{X}_t, t)\right) + \tilde{\beta}_t Z_t, \quad Z_t \sim \mathcal{N}(0, I)$$

end

Idea: Sample X_t via the approximation of $p(X_t \mid X_{t-1})$. It is an approximation because $p(X_t \mid X_{t-1})$ is not exactly Gaussian and because the scaled score network ε_θ is not exact.

Reinterpreting DDPM sampling

Consider the case

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t.$$

We can equivalently express DDPM sampling as:

$\overline{X}_T \sim \mathcal{N}(0, I)$

for $t = T, T-1, \ldots, 2, 1$

$$\hat{X}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\overline{X}_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\,\varepsilon_\theta(\overline{X}_t, t)$$

$$Z_t \sim \mathcal{N}(0, I)$$

$$\overline{X}_{t-1} = \frac{\sqrt{\bar{\alpha}_t}\beta_t}{1-\bar{\alpha}_t}\hat{X}_0 + \frac{\sqrt{1-\beta_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t}\overline{X}_t + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\,Z_t$$

end

Reinterpreting DDPM sampling

Since $X_t \mid X_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t} X_0, (1-\bar{\alpha}_t) I\right)$, Tweedie's formula tells us

$$\begin{aligned}
\mathbb{E}[X_0 \mid X_t] &= \frac{1}{\sqrt{\bar{\alpha}_t}} X_t + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} \nabla_{X_t} \log p_{X_t}(X_t) \\
&\approx \frac{1}{\sqrt{\bar{\alpha}_t}} X_t + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} s_\theta(X_t, t) \\
&= \frac{1}{\sqrt{\bar{\alpha}_t}} X_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \varepsilon_\theta(X_t, t)
\end{aligned}$$

Also, using

$$p(x_{t-1} \mid x_t, x_0) = \frac{p(x_t \mid x_{t-1}, x_0)\, p(x_{t-1} \mid x_0)}{p(x_t \mid x_0)} = \frac{p(x_t \mid x_{t-1})\, p(x_{t-1} \mid x_0)}{p(x_t \mid x_0)},$$

we can compute

$$p(X_{t-1} \mid X_t, X_0) = \mathcal{N}\left(\frac{\sqrt{\bar{\alpha}_t}\beta_t}{1-\bar{\alpha}_t} X_0 + \frac{\sqrt{1-\beta_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} X_t,\ \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I\right)$$

Reinterpreting DDPM sampling

Using these identities, we can reinterpret DDPM sampling as

$\overline{X}_T \sim \mathcal{N}(0, I)$

for $t = T, T-1, \ldots, 2, 1$

$$\hat{X}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \overline{X}_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \varepsilon_\theta(\overline{X}_t, t) \qquad \overline{X}_{t-1} \sim p\left(\overline{X}_{t-1} \mid \overline{X}_t, \overline{X}_0 = \hat{X}_0\right)$$

end

At each step, (i) estimate X_0 and (ii) sample $\overline{X}_{t-1}$ conditioned on $\overline{X}_t$ and $\overline{X}_0 = \hat{X}_0$.

DDPM = discretization of VP SDE

DDPM forward process in the limit $\beta_t \to 0$

$$X_{t+1} = \sqrt{1-\beta_t} X_t + \sqrt{\beta_t} Z_t \approx \left(1 - \frac{\beta_t}{2}\right) X_t + \sqrt{\beta_t} Z_t$$

Consider the general VP forward-time SDE

$$dX_t = -\frac{\beta(t)}{2} X_t dt + \sqrt{\beta(t)} dW_t$$

With $\Delta t = 1$, the Euler–Maruyama discretization is

$$X_{t+1} = \left(1 - \frac{\beta(t)}{2}\right) X_t + \sqrt{\beta(t)} Z_t$$

and the two agree.

DDPM = discretization of VP SDE

DDPM sampling when $\beta_t \to 0$ (and slowly varying)

$$\begin{aligned}\overline{X}_{t-1} &= \frac{1}{\sqrt{1-\beta_t}}\left(\overline{X}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\overline{X}_t, t)\right) + \sigma_t Z_t \\ &\approx \left(1 + \frac{\beta_t}{2}\right)\overline{X}_t + \frac{\beta_t}{\sqrt{1 - \exp\left(-\int_0^t \beta(s)\,ds\right)}}\varepsilon_\theta(\overline{X}_t, t) + \sigma_t Z_t\end{aligned}$$

Here, we identify $\beta(t) = \beta_t$ and argue that

$$\bar{\alpha}_t = \prod_{s=0}^{t}(1-\beta_s) \approx \prod_{s=0}^{t}\exp(-\beta_s) = \exp\left(-\sum_{s=0}^{t}\beta_s\right) \approx \exp\left(-\int_0^t \beta(s)\,ds\right)$$

DDPM = discretization of VP SDE

Reverse-time VP SDE

$$d\overline{X}_t = \left(\frac{\beta(t)}{\sigma_t} \varepsilon_\theta(\overline{X}_t, t) - \frac{\beta(t)}{2} \overline{X}_t \right) dt + \sqrt{\beta(t)} d\overline{W}_t$$

With $\Delta t = -1$, the Euler–Maruyama discretization is

$$\overline{X}_{t-1} = \overline{X}_t - \left(\frac{\beta(t)}{\sqrt{1 - \exp\left(-\int_0^t \beta(s)\, ds\right)}} \varepsilon_\theta(\overline{X}_t, t) + \frac{\beta(t)}{2} \overline{X}_t \right) - \sqrt{\beta(t)} Z_t$$

and the two agree.