Denoising Diffusion Models

Gabriel Peyré
CNRS & DMA
École Normale Supérieure
gabriel.peyre@ens.fr
https://mathematical-tours.github.io
www.numerical-tours.com

July 31, 2023

Abstract

This document provides a concise overview of denoising diffusion models. Most of it is adapted from the excellent slides of Valentin de Bortoli[1]. The discussion begins with a review of classical Langevin sampling, a method that, while not ideally suited for generative modeling, establishes a clear connection between sampling methods, stochastic differential equations (SDEs), and linear partial differential equations (PDEs). We then explore the fundamental concept underlying diffusion models, which is the ability to invert linear SDEs through SDEs of the same class, but with a problem-dependent drift. This inversion involves the so-called "score", defined as the gradient of the log of the evolving density. The final section introduces score matching, a technique to estimate this score from samples of the trajectories. Denoising score matching reframes the problem, recasting it as the computation of an optimal denoiser.

1 Langevin sampling

Langevin diffusion is a method to accelerate sampling from a distribution with density $\rho_0(x) \triangleq e^{-f(x)}$, leveraging the smoothness of f. In its discrete (and approximate) form, it corresponds to a noisy gradient descent, where the noise is Gaussian

$$Z_{k+1} = Z_k - \tau \nabla f(Z_k) + \sqrt{2\tau} W_k,$$

where $W_k \sim \mathcal{N}(0, \mathrm{Id}_d)$ are i.i.d.

Setting $t = \tau k$, as $\tau \to 0$, this leads to considering the following Langevin stochastic differential equation (SDE)

$$\mathrm{d}Z_t = -\nabla f(Z_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t, \tag{1}$$

where $t \mapsto W_t$ is a Wiener process. One can show that, regardless of the distribution of Z_0, the law of Z_t converges in law towards a density $e^{-f(x)}$ with respect to Lebesgue measure.

Convergence issues. Note that if one replaces f by f/ε, this converges to $e^{-f(x)/\varepsilon}$, so that as $\varepsilon \to 0$ one recovers (noiseless) gradient descent, which converges to stationary points of f. The caveat is that the convergence of Langevin will become slower and slower as ε becomes smaller.

The power of Langevin lies in its independence with respect to initialization. Its weaknesses are its slow convergence for non-convex f and the necessity to have direct access to $\nabla \log(\rho_0) = \nabla f$. For applications

[1] https://vdeborto.github.io/project/generative_modeling/

to generative models, this is not acceptable because one can only assume to have access to ρ_0 from samples. These two drawbacks can somehow be alleviated by the diffusion model framework, which loosely speaking, corresponds to replacing the drift $\nabla \log(\rho_0)$ by $\nabla \log(\rho_t)$ where ρ_t is a suitable smoothing of ρ_0.

PDE interpretation. For a vector field v (for instance, $v = -\nabla f$ in (1)), the law ρ_t of a process Z_t satisfying

$$\mathrm{d}Z_t = v(Z_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t,$$

can be shown to satisfy (in the weak sense) the following heat diffusion equation with drift (also called the Fokker-Planck equation)

$$\partial_t \rho_t = -\mathrm{div}(\rho_t v) + \Delta \rho_t. \tag{2}$$

2 Diffusion model

To obtain an exact synthesis process leveraging a time-dependent smoothing, one can invert a forward diffusion process. The disadvantage of this approach is that it only works for a specific initial condition (as opposed to Langevin, which works for any initialization), which is the limit density of the forward diffusion (here a Gaussian).

Forward flow. To converge towards a Gaussian, one can consider a Langevin flow with a linear drift, $-x = -\nabla f$ where $f(x) = \frac{\|x\|^2}{2}$, which defines an Ornstein-Uhlenbeck process. The continuous forward flow (noising process) is thus

$$\mathrm{d}X_t = -X_t\mathrm{d}t + \sqrt{2}\mathrm{d}W_t.$$

It converges in law exponentially fast toward $\mathcal{N}(0, \mathrm{Id})$, and more precisely, one has equality in law

$$X_t \sim e^{-t}X_0 + \sqrt{1 - e^{-2t}}Z, \tag{3}$$

where $Z \sim \mathcal{N}(0, \mathrm{Id})$. This means that ρ_t is a Gaussian smoothing (with an increasing bandwidth) of a rescaled version of ρ_0

$$\rho_t = \rho_0(\cdot/e^t) \star \mathcal{N}(0, 1 - e^{-2t}), \qquad f \star g(x) = \int f(y)g(x-y)\mathrm{d}y.$$

Backward flow. The actual sampling (the generative process) is now done by reverting in time this process, i.e., for a large enough $T \gg 0$, one seeks to approximate $Y_t \triangleq X_{T-t}$. Denoting ρ_t the law of X_t and $\xi_t = \rho_{T-t}$ the law of Y_t, the first idea is to reverse in time the Fokker-Planck PDE (2), since $\partial_t \xi_t = -\partial_t \rho_{T-t}$,

$$\partial_t \rho_t = -\mathrm{div}(-\rho_t x) + \Delta \rho_t \quad \Rightarrow \quad \partial_t \xi_t = -\mathrm{div}(\xi_t x) - \Delta \xi_t.$$

This corresponds to a backward heat equation, which is unstable and cannot be computed (and also, it cannot be represented using an SDE).

An alternative approach is to re-write $-\Delta \xi_t$ as a Laplacian plus a drift which is equal to the score $\nabla \log(\xi_t)$, since one has

$$-\Delta \xi_t = \Delta \xi_t - 2\mathrm{div}(\xi_t \nabla \log(\xi_t)) = \Delta \xi_t - 2\mathrm{div}(\xi_t \nabla \log(\xi_t)).$$

One thus has that ξ_t is also a solution of a Fokker-Planck equation

$$\partial_t \xi_t = -\mathrm{div}(\xi_t x + 2\xi_t \nabla \log(\xi_t)) + \Delta \xi_t.$$

This shows that ξ_t is the law of a process Y_t, satisfying the following Langevin SDE, initialized with $Y_0 = X_T$

$$\mathrm{d}Y_t = [Y_t + 2\nabla \log(\rho_{T-t})(Y_t)]\mathrm{d}t + \sqrt{2}\mathrm{d}W_t.$$

This can be discretized using an Euler-Maruyama scheme, starting from $Y_0 = X_{T/\tau}$

$$Y_{k+1} = Y_k + \tau Y_t + 2\tau \nabla \log(\rho_{T-t}) + \sqrt{2\tau} W_k. \tag{4}$$

The idea of using Langevin with a score function to perform generative modeling was introduced in [2]. The rigorous backward SDE presented here is due to [3].

Initialization. One issue in this approach is that the exact initialization $Y_0 = X_T$ is not possible since in practice ρ_0 is only approximately known. This is circumvented by replacing ρ_T by $\rho_\infty = \mathcal{N}(0, \mathrm{Id})$.

3 Denoising Score Matching

In order to be able to implement (4), one needs to compute the score $\nabla \log(\rho_t)$, where ρ_t is the density of the distribution of X_t, defined as in (3). The idea is to approximate this score using a function computed from samples of X_t.

In the following, we denote X_0 as X and X_t as Y. The following derivation is valid for any pair of random vectors (X, Y) such that one can sample from the pair (X, Y) and has a closed-form expression for the conditional density $\mathbb{P}_{Y|X}(\cdot|x)$ of Y given $X = x$. According to (3), in the special case of a diffusion model, Y is a Gaussian random variable with mean $e^{-t}X_0$ and variance $(1 - e^{-2t})\mathrm{Id}$, so that

$$\log \mathbb{P}_{Y|X}(y|x) = -\frac{\|y - x\|^2}{2(1 - e^{-2t})} \quad \Rightarrow \quad \nabla_y \log \mathbb{P}_{Y|X}(y|x) = -\frac{y - e^{-t}x}{1 - e^{-2t}}. \tag{5}$$

The following derivation is informal, and we denote $\mathbb{P}_{(Y,X)}(x, y)$ as the law of (X, Y) (with respect to some fixed reference measure, such as Lebesgue), and for the sake of readability, denote it as $\mathbb{P}(y, x)$. The same goes for the conditionals $\mathbb{P}(y|x)$ and $\mathbb{P}(x|y)$. One has $\mathbb{P}(y|x) = \frac{\mathbb{P}(y,x)}{\mathbb{P}(x)}$, so that

$$\frac{\nabla_y \mathbb{P}(y|x)}{\mathbb{P}(y|x)} = \nabla_y \log \mathbb{P}(y|x) = \nabla_y \log \frac{\mathbb{P}(y, x)}{\mathbb{P}(x)} = \frac{\nabla_y \mathbb{P}(y, x)}{\mathbb{P}(y, x)}.$$

One also has $\mathbb{P}(y) = \int \mathbb{P}(y, x)\,\mathrm{d}x$, so that using the previous equation

$$\nabla \log \mathbb{P}(y) = \frac{\nabla \mathbb{P}(y)}{\mathbb{P}(y)} = \frac{1}{\mathbb{P}(y)} \int \nabla_y \mathbb{P}(y, x)\,\mathrm{d}x = \frac{1}{\mathbb{P}(y)} \int \mathbb{P}(y, x) \nabla_y \log \mathbb{P}(y|x)\,\mathrm{d}x = \int_x \nabla_y \log \mathbb{P}(y|x)\,\mathrm{d}\mathbb{P}(x|y).$$

The last expression writes $\nabla \log \mathbb{P}(y)$ as an average of $\nabla_y \log \mathbb{P}(y|x)$ according to the probability distribution $\mathbb{P}(x|y)$. This can equivalently be re-written as the minimization of a mean square

$$\nabla \log \mathbb{P}(y) = \operatorname*{arg\,min}_{\varphi(y) \in \mathbb{R}^d} \int_x \|\nabla_y \log \mathbb{P}(y|x) - \varphi(y)\|^2\,\mathrm{d}\mathbb{P}(x|y).$$

Note that the function $\nabla_y \log \mathbb{P}(y|x)$ is assumed to be computable in closed form.

In practice, this non-parametric estimation of φ is replaced by a parametric estimation, as one has to perform it by sampling $\mathbb{P}(\cdot|y)$. This can be done by integrating over y with random sampling from $\mathbb{P}(y)$, resulting in an integration over $\mathbb{P}(y, x)$ (from which one can sample by first sampling x and then y according to $\mathbb{P}(y|x)$)

$$\nabla \log \mathbb{P}(\cdot) \approx \operatorname*{arg\,min}_{\theta} \int_y \int_x \|\nabla_y \log \mathbb{P}(y|x) - \varphi_\theta(y)\|^2\,\mathrm{d}\mathbb{P}(y, x). \tag{6}$$

The function $\varphi_\theta : \mathbb{R}^d \to \mathbb{R}^d$ is usually a neural network but of a specific type because it maps $\mathbb{R}^d$ to itself. For images, a model of choice is U-Nets, which operate similarly to wavelet analysis and synthesis. The minimization of (6) is performed by stochastic gradient descent.

The initial idea of using score matching to perform density estimation was introduced in [1]. The variational formulation as an optimal denoiser is due to [4].

References

[1] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[2] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[4] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.