Quick start for the theory of diffusion model

Wu Haoyang

July 2025

1 Introduction

The original derivation of Denoising Diffusion Probabilistic Model (DDPM) is a little bit too tedious. The author finds a way to comprehend the diffusion model more easily. Here is an effort to share it with others

2 Symbols we use in this text

$\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_n \in \mathbb{R}^{n\times 1}$: vector, representing the high dimensional data like image, video;

$q(\mathbf{X}), p(\mathbf{X};\theta)$: scalar, the probabilistic distribution of the high dimensional data;

t : scalar, representing a nominal time in DDPM;

$\mathbf{s}_\theta(\mathbf{x}, t)$: denoise function, which is approximated by a neural network;

3 A brief introduction of the DDPM algorithm

3.1 What we do while generation?

Before we tackle the full derivation in the next section, we first develop an intuitive feel for DDPM: how the algorithm turns a purely noisy input—resembling television static in Fig.1—into a coherent image, such as the kitten also shown in Fig.1.

As an old saying goes, 'If you can't open it, you don't own it.' What DDPM does Fig. 1 illustrates two main processes involved: the forward diffusion process and the reverse denoising process. The forward diffusion process indicates a trivial fact: any image could be degenerated by continuously adding noise. Following the reverse denoising process, it could be seen that a cute kitty could be gradually generated from an image full of random noise. As we could see, for a given image, we could get a series of images with varying degrees of noise.

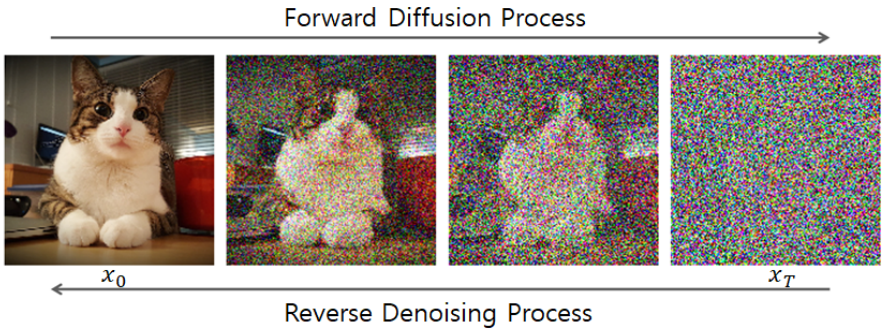

Figure 1: This figure illustrates two main processes of DDPM: the forward diffusion process (the top arrow, from left to right) and the reverse denoising process (the bottom arrow, from right to left).

Algorithm 1 Sampling

$x_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots, 1$ **do**
  if $t > 1$ **then**
    $z \sim \mathcal{N}(0, I)$
  else
    $z \leftarrow 0$
  end if
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(x_t, t)\right) + \sigma_t z$
end for
return $x_0 = 0 = 0$

3.2 Our strategy to understand the derivation

In Sec. 3.1, we explained how the model is trained by incrementally corrupting images with noise, and how the trained model, in turn, produces meaningful images by successively denoising an input that begins as TV snow. And based Empirically, it would be more easy to understand if we know the key idea before reading a mathematical derivation.

4 The spectacular original derivation

Many newcomers to DDPM would complain about its long, tedious and intimidating standard derivation. And most technical posts just literally reproduce the original derivation without conveying the intuition and the motivation. However, in my view, this derivation showcases several elegant and skillful tricks that are worth mastering and appreciating. So in this section, we would walk

through the elegant derivation step by step, helping readers grasp and retain its finer details.

Here, assume that we have a large dataset $\mathcal{D} = \{\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, ...\mathbf{X}_n\}$, and we have a neural network with parameters θ to be optimized on $\mathcal{D}$.

$$\log\big(p(\mathbf{X};\theta)\big) = \log(p(\mathbf{X}_0;\theta)) \tag{1}$$

$$= \log \int p(\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T;\theta)\, d\mathbf{X}_1\, d\mathbf{X}_2\, d\mathbf{X}_3 \ldots d\mathbf{X}_T \tag{2}$$

$$= \log \int p(\mathbf{X}_{0:T};\theta)\, d\mathbf{X}_{1:T} \tag{3}$$

$$= \log \int p(\mathbf{X}_{0:T};\theta) \frac{q(\mathbf{X}_{1:T} \mid \mathbf{X}_0)}{q(\mathbf{X}_{1:T} \mid \mathbf{X}_0)}\, d\mathbf{X}_{1:T} \tag{4}$$

$$= \log \int p(\mathbf{X}_{0:T};\theta) \frac{q(\mathbf{X}_{1:T} \mid \mathbf{X}_0)}{q(\mathbf{X}_{1:T} \mid \mathbf{X}_0)}\, d\mathbf{X}_{1:T} \tag{5}$$

$$= \log \mathbb{E}_{\mathbf{X}_{1:T}\sim q(\mathbf{X}_{1:T}|\mathbf{X}_0)}\Big[\frac{p(\mathbf{X}_{0:T};\theta)}{q(\mathbf{X}_{1:T} \mid \mathbf{X}_0)}\Big] \tag{6}$$

$$\geq \mathbb{E}_{\mathbf{X}_{1:T}\sim q(\mathbf{X}_{1:T}|\mathbf{X}_0)}\Big[\log \frac{p(\mathbf{X}_{0:T};\theta)}{q(\mathbf{X}_{1:T} \mid \mathbf{X}_0)}\Big] \tag{7}$$

$$= \mathbb{E}_{\mathbf{X}_{1:T}\sim q(\mathbf{X}_{1:T}|\mathbf{X}_0;\theta)}\Big[\log \frac{\prod_{t=1}^{T} p(\mathbf{X}_{t-1} \mid \mathbf{X}_t;\theta)}{\prod_{t=1}^{T} q(\mathbf{X}_t \mid \mathbf{X}_{t-1})}\Big] \tag{8}$$

Let's pause here for a moment and review the assumptions and conditions we have used so far in the derivation. From Eq. 1 to Eq. 2, we introduce more variables to characterize $\mathbf{X}_0$.

5 From the perspective of score based model

6 Associate the score based model and the diffusion model in implementation

7 A very simple demo