How Diffusion Model Works

Zhe Li

In this note, we will explain three components to understand how diffusion model works. My objective is to provide a general logic chain to understand this topic, rather than diving into derivation of mathematical equations. Nevertheless some equations will be included to make the context clear. For detail mathematical derivation, follow the link provided in the footnote for corresponding section.

1 Forward Process

Starting from original data point $\mathbf{x}_0$, add the controlled noise at every step t, then we have the following sequence of noise data $\mathbf{x}_i, i \in [1, T]$

$$\mathbf{x}_0 \to \mathbf{x}_1 \to \mathbf{x}_2 \to \cdots \to \mathbf{x}_T \tag{1}$$

Since every time $\mathbf{x}_t$ is from $\mathbf{x}_{t-1}$ and some controlled noise, we denote this step as $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. Naively, we can add noise by using this:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + N(0, \beta_t), \tag{2}$$

here β_t could be constant or value depending on the step t. However, no matter β_t is const or value depending on the step t, $\mathbf{x}_t$ will have the larger and larger variance as $t \to T$. This is undesired, because of the range of magnitude of the vectors you get at the end of will depend on the number of T of steps[1]. So we have to scale down the mean of $\mathbf{x}_{t-1}$ with some parameter γ. In order to compute what is proper value of γ. Let's assume $\mathbf{x}_0 \sim N(\mu, 1)$ and we would like to keep the variance at every step t. To obtain $\mathbf{x}_1$, we add noise $\epsilon_1 \sim N(0, \beta_1)$, then

$$\mathbf{x}_1 = \gamma \mathbf{x}_0 + \sqrt{\beta_1}\epsilon_1 \tag{3}$$

The variance of $\mathbf{x}_1$ is

$$Var(\mathbf{x}_1) = \gamma^2 + \beta_1 \tag{4}$$

In order to keep variance of $\mathbf{x}_t$ to 1, then we have $\gamma^2 + \beta_1 = 1$, which gives

$$\gamma = \sqrt{1 - \beta_1} \tag{5}$$

[1] Directly from here https://stats.stackexchange.com/questions/600127/purpose-of-scaling-mean-by-sqrt1-beta-t-in-forward-diffusion-process

The above analysis can be extended to all $\mathbf{x}_t$. Thus we can have the following sampling formula for $\mathbf{x}_t$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t I) \tag{6}$$

or equivalently,

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \beta_t \epsilon_{t-1} \tag{7}$$

If we define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, then we have

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{8}$$

The benefit of this re-parameter on α_t and β_t is that we can directly compute $\mathbf{x}_t$ based on $\mathbf{x}_0$ without going through entire sequential $\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \cdots \mathbf{x}_1$. The goal is as $t \to \infty$, $\mathbf{x}_t \to N(0, I)$ random gaussian noise, which needs $\bar{\alpha}_t$ is decreasing sequence and $\bar{\alpha}_t \to 0$

2 Backward Process

Starting from original pure Gaussian noise $\mathbf{x}_T$, reduce the controlled amount of noise sequentially, then we have the following sequence of noise data $\mathbf{x}_i, i \in [1, T]$.

$$\mathbf{x}_T \to \mathbf{x}_{T-1} \to \mathbf{x}_{T-2} \to \cdots \to \mathbf{x}_0 \tag{9}$$

In order to compute $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$, we have to have $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, but unfortunately it is not easily to estimate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$

- not easily to estimate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$
- tractable to estimate $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$, which is conditional probability of $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ on $\mathbf{x}_0$ training data point[2].

Let's assume that $p(\mathbf{x}_{t-1}|, \mathbf{x}_t, \mathbf{x}_0)$ has the following form

$$p(\mathbf{x}_{t-1}|, \mathbf{x}_t, \mathbf{x}_0) = N(\mathbf{x}_{t-1}|\mu(\mathbf{x}_t, \mathbf{x}_0), \beta_t I) \tag{10}$$

To justify the second point, based on bayes' theorem, we have the following:

$$\begin{aligned} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\ &\propto \exp(\text{mean from } q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) + \text{mean from } q(\mathbf{x}_{t-1}|\mathbf{x}_0) - \text{mean from } q(\mathbf{x}_t|\mathbf{x}_0)) \end{aligned} \tag{11}$$

[2] Following the mathematical proof https://lilianweng.github.io/posts/2021-07-11-diffusion-models/#reverse-diffusion-process

Note all $q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0), q(\mathbf{x}_{t-1}|\mathbf{x}_0, q(\mathbf{x}_t|\mathbf{x}_0$ are all forward process, gaussian distribution. We can plug into analytic formula of $q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0), q(\mathbf{x}_{t-1}|\mathbf{x}_0, q(\mathbf{x}_t|\mathbf{x}_0)$ and mathematic exercise, we can have mean and variance of $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$, thus we have

$$\mu_t(\mathbf{x}_t,\mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t) \tag{12}$$

3 Loss function and Maximum Likelihood

So far we discussed the forward process from $\mathbf{x}_0 \to \mathbf{x}_1 \to \mathbf{x}_2 \to \cdots \to \mathbf{x}_T$ and backward process from $\mathbf{x}_T \to \mathbf{x}_{T-1} \to \mathbf{x}_{T-2} \to \cdots \to \mathbf{x}_0$. We have not touched the loss function yet. What is the loss function that we are going to optimize?

The fundamentals of generative model (including diffusion model) is to approximate the real world (image, video, text) distribution p_r(real world) or more accurate $p_{\text{data}}(x)$. How to achieve this?

- sample an vector $\mathbf{z}$ from a simpler distribution (normal distribution) $N(0, I)$;
- feed that vector $\mathbf{z}$ to an network $\mathbf{G}$ to generate image or video or text from generative distribution $p_G(\mathbf{x})$ (or using $P_\theta(\mathbf{x})$, θ is the parameter is the network), which should best approximate $p_{\text{data}}(x)$.

How to measure how close of two distribution $p_G(\mathbf{x})$ or $p_\theta(\mathbf{x})$ and $p_{\text{data}}(\mathbf{x})$? we can use Maximum likelihood on data, which is equivalent to minimize KL divergence. Back to diffusion model, the goal is to maximize

$$p_\theta(\mathbf{x}_0) = \int_{\mathbf{x}_1:\mathbf{x}_T} p(\mathbf{x}_T)p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T)p_\theta(\mathbf{x}_{T-2}|\mathbf{x}_{T-1})\cdots p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\cdots p_\theta(\mathbf{x}_0|\mathbf{x}_1)d\mathbf{x}_1 : \mathbf{x}_T \tag{13}$$

Let's compare VAE and diffusion model [3]:

$$\begin{aligned}&\text{VAE Maximize} \log(p_\theta(\mathbf{x})) \to \text{maximize} E_{q(\mathbf{z}|\mathbf{x})}[\log(\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}|\mathbf{x})})]\\&\text{Diffusion Maximize} \log(p_\theta(\mathbf{x}_0)) \to \text{maximize} E_{q(\mathbf{x}_1:\mathbf{x}_T|\mathbf{x}_0)}[\log(\frac{p(\mathbf{x}_0:\mathbf{x}_T)}{q(\mathbf{x}_1:\mathbf{x}_T|\mathbf{x}_0)})]\end{aligned} \tag{14}$$

In VAE, $q(\mathbf{z}|\mathbf{x})$ is encode while in diffusion model $q(\mathbf{x}_1 : \mathbf{x}_T|\mathbf{x}_0)$ is diffusion process or forward process.

[3] we largely follow this lecture Diffusion Model

If we massage the diffusion $\log(p_\theta(\mathbf{x}_0))$, we eventually have

$$\begin{aligned} E_{q(\mathbf{x}_1:\mathbf{x}_T|\mathbf{x}_0)}[\log(\frac{p(\mathbf{x}_0:\mathbf{x}_T)}{q(\mathbf{x}_1:\mathbf{x}_T|\mathbf{x}_0)})] =& E_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log(p(\mathbf{x}_0|\mathbf{x}_1))] - KL(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)) \\ & - \sum_{t=2}^{T} E_{q(\mathbf{x}_t|\mathbf{x}_0)}[KL(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t))] \end{aligned} \tag{15}$$

The second term in the right hand side does not depend on network θ. Let's focus on the third term. $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ is the back forward process, in the second section, we have already computed its mean in Eq. 12 based on bayes's theorem. For convenience, we can copy that here:

$$\mu_t(\mathbf{x}_t,\mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t) \tag{16}$$

In the third term, $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ depends on network G or explicitly $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ depending on network parameter. To maximize $\log(p_\theta(\mathbf{x}_0))$, is to minimize the third term KL divergence between $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

Let's thinking the process, for $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$, we start with $\mathbf{x}_0$ and can compute $\mathbf{x}_t$. Based on $\mathbf{x}_0$ and $\mathbf{x}_t$ and Eq. 12, we can compute the mean of $\mathbf{x}_{t-1}$. In order to minimize the KL divergence between $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, we have tune network θ so that $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ have same or close mean of $\mathbf{x}_{t-1}$. Since for the mean in the Eq. 12, $\mathbf{x}_t$ and α_t are fixed, the only tunable output is θ_t, thus we optimize network to tune θ_t.

The DL divergence of two gaussian distribution of loss becomes

$$\begin{aligned} L_t &= E_q[||\mu_t(\mathbf{x}_t,\mathbf{x}_0) - \mu_\theta(\mathbf{x}_t,t)||^2] \\ &\equiv E_q[||\epsilon_t - \epsilon_\theta(\mathbf{x}_t,t)||^2] \\ &\equiv E_q[||\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t,t)||^2] \end{aligned} \tag{17}$$

We plug Eq 8 $\mathbf{x}_t$ into equation in second line to get the last equation.