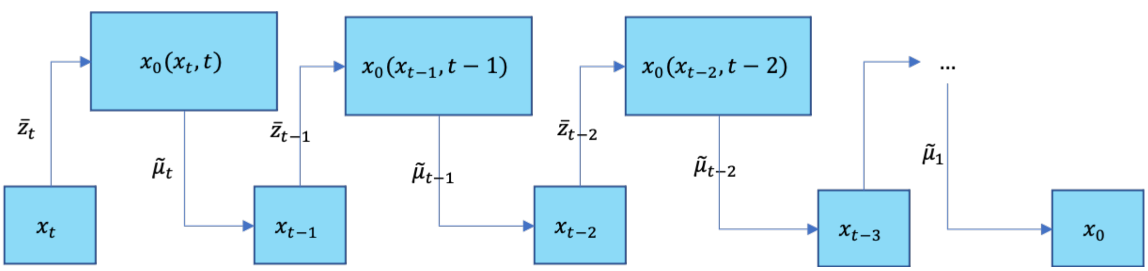
由浅入深了解Diffusion Model

目录



前言

其实早在去年就看过大佬Lil关于diffusion model精彩的介绍What are Diffusion Models?[1]但是后面一直没深入研究，很快就忘细节了。最近Diffusion Model火到爆炸（GLIDE[2],DALLE2[3],Imagen[4],和一系列Image Editing方法等等），所以又重新建起来学习了下。恐怕diffusion拥有成为下一代图像生成模型的代表的潜力（或者已经是了？）本文主要是对Lil博客进行翻译整理，会添加一些细节的理解和对照代码的思考，主要是为了方便自己学习记录，如果有理解错误的地方还请指出。

什么是Diffusion Model（扩散模型）？

首先我们来看一下最近火爆各个公众号的text-to-image结果：

图1. DALLE2生成结果

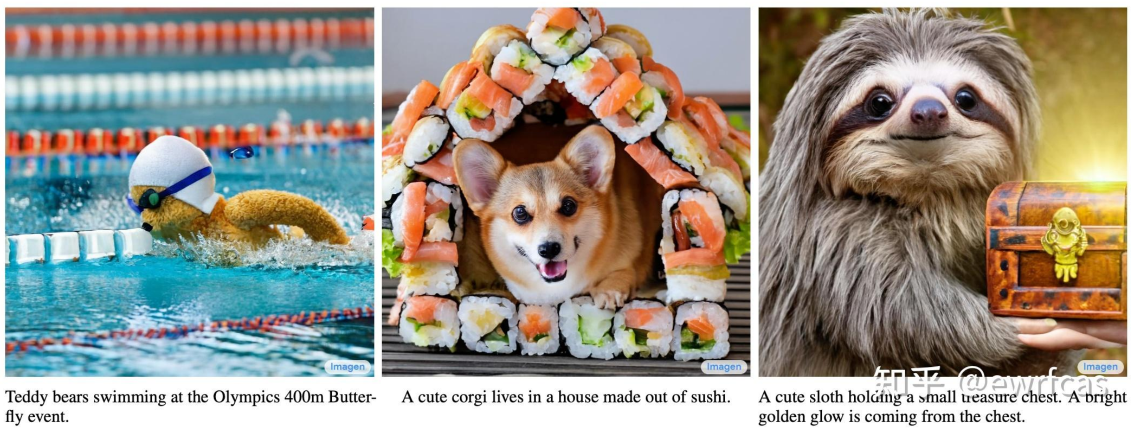


图2. Imagen生成结果

上述图片的结果都非常惊人，无论从真实度还是还原度都几乎无可挑剔。这里我们从由浅入深来了解一下Diffusion Model。首先还是放一个各类生成模型对比图:

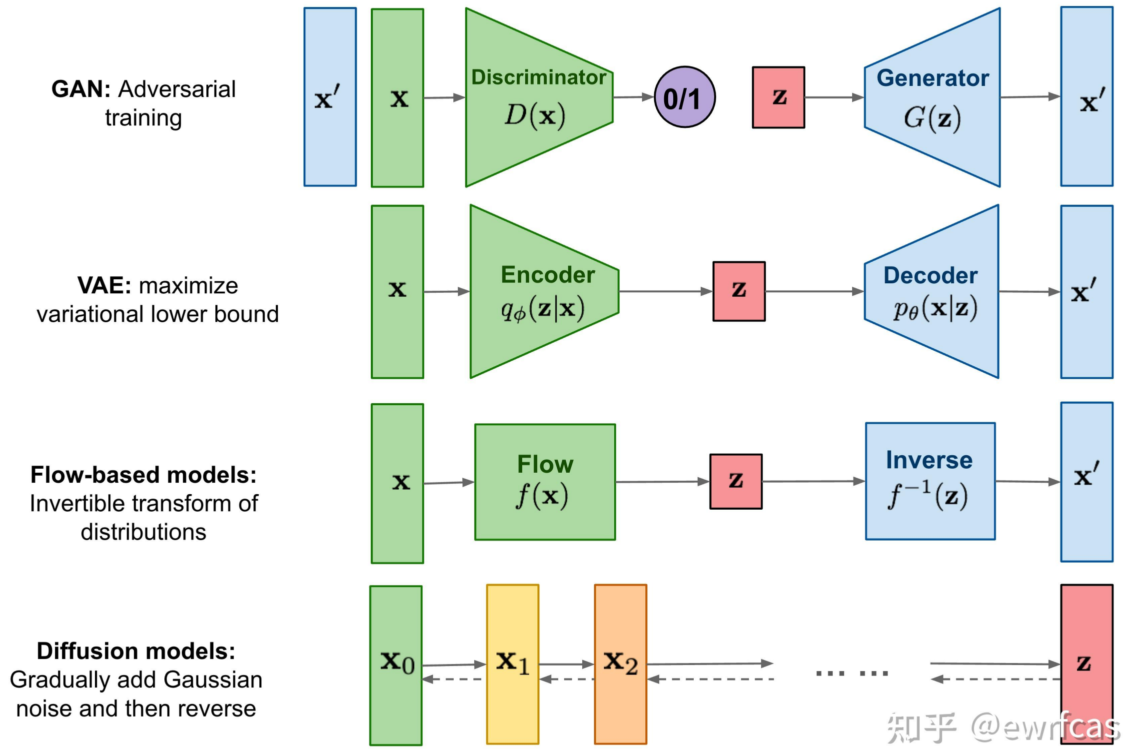


图3. 不同生成模型对比图（来源：Lil博客）

diffusion model和其他模型最大的区别是它的latent code(z)和原图是同尺寸大小的，当然最近也有基于压缩的latent diffusion model[5]，不过是后话了。一句话概括diffusion model，即存在一系列高斯噪声（T轮），将输入图片 x_0变为纯高斯噪声 x_T。而我们的模型则负责将 x_T复原回图片 x_0。这样一来其实diffusion model和GAN很像，都是给定噪声 x_T生成图片 x_0，但是要强调的是，这里噪声 x_T与图片x_0是**同维度** 的。

diffusion model有很多种理解，这里介绍是基于denoising diffusion probabilistic models (DDPM)[6]的。

Diffusion前向过程

所谓前向过程，即往图片上加噪声的过程。虽然这个步骤无法做到图片生成，但是这是理解diffusion model以及**构建训练样本GT** 至关重要的一步。

给定真实图片 $x_0 \sim q(x)$,diffusion前向过程通过 T次累计对其添加高斯噪声，得到 $x_1, x_2, \ldots, x_T$，如下图的q过程。这里需要给定一系列的高斯分布方差的超参数 $\{\beta_t \in (0,1)\}_{t=1}^T$.前向过程由于每个时刻 t只与 $t-1$时刻有关，所以也可以看做马尔科夫过程：

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}).$$

这个过程中，随着 t的增大， x_t越来越接近纯噪声。当 $T \rightarrow \infty$， x_T是完全的高斯噪声（下面会证明，且与均值系数$\sqrt{1-\beta_t}$的选择有关）。且实际中 β_t随着t增大是递增的，即 $\beta_1 < \beta_2 < \ldots < \beta_T$。在GLIDE的code中， β_t是由0.0001到0.02线性插值（以 $T=1000$为基准， T增加， β_t对应降低）。

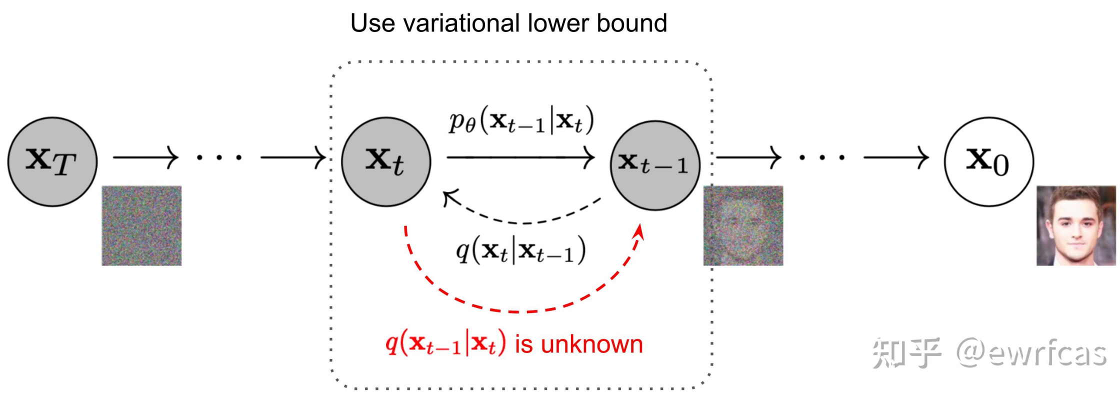


图4. diffusion的前向(q)和逆向(p)过程，来源：DDPM

前向过程介绍结束前，需要讲述一下diffusion在实现和推导过程中要用到的两个重要特性。

特性1：重参数（reparameterization trick）

重参数技巧在很多工作（gumbel softmax, VAE）中有所引用。如果我们要从某个分布中随机采样(高斯分布)一个样本，这个过程是无法反传梯度的。而这个通过高斯噪声采样得到 x_t的过程在diffusion中到处都是，因此我们需要通过重参数技巧来使得他可微。最通常的做法是吧随机性通过一个独立的随机变量(ϵ)引导过去。举个例子，如果要从高斯分布 $z \sim \mathcal{N}(z; \mu_\theta, \sigma_\theta^2\mathbf{I})$采样一个z，我们可以写成:

$$z = \mu_\theta + \sigma_\theta \odot \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}).$$

上式的z依旧是有随机性的， 且满足均值为 μ_θ方差为 σ_θ^2的高斯分布。这里的μ_θ， σ_θ^2可以是由参数 θ的神经网络推断得到的。整个“采样”过程依旧梯度可导，随机性被转嫁到了ϵ上。

特性2：任意时刻的 xtx_tx_t 可以由 x0x_0x_0 和 β\beta\beta 表示

能够通过 x_0和 β快速得到 x_t对后续diffusion model的推断和推导有巨大作用。首先我们假设 $\alpha_t = 1-\beta_t$，并且 $\overline{\alpha}_t = \prod_{i=1}^T \alpha_i$，展开 x_t可以得到:

$$
\begin{aligned}
x_t &= \sqrt{a_t}x_{t-1} + \sqrt{1-\alpha_t}z_1 \quad \text{where} \quad z_1, z_2, \ldots \sim \mathcal{N}(0, \mathbf{I}); & (1)\\
&= \sqrt{a_t}(\sqrt{a_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}z_2) + \sqrt{1-\alpha_t}z_1 & (2)\\
&= \sqrt{a_t a_{t-1}}x_{t-2} + (\sqrt{a_t(1-\alpha_{t-1})}z_2 + \sqrt{1-\alpha_t}z_1) & (3)\\
&= \sqrt{a_t a_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{z}_2 \quad \text{where} \quad \bar{z}_2 \sim \mathcal{N}(0, \mathbf{I}); & (4)\\
&= \ldots & (5)\\
&= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\bar{z}_t. & (3)
\end{aligned}
$$

由于独立高斯分布可加性，即$\mathcal{N}(0, \sigma_1^2\mathbf{I}) + \mathcal{N}(0, \sigma_2^2\mathbf{I}) \sim \mathcal{N}(0, (\sigma_1^2+\sigma_2^2)\mathbf{I})$，所以

$$
\begin{aligned}
&\sqrt{a_t(1-\alpha_{t-1})}z_2 \sim \mathcal{N}(0, a_t(1-\alpha_{t-1})\mathbf{I}) & (6)\\
&\sqrt{1-\alpha_t}z_1 \sim \mathcal{N}(0, (1-\alpha_t)\mathbf{I}) & (7)\\
&\sqrt{a_t(1-\alpha_{t-1})}z_2 + \sqrt{1-\alpha_t}z_1 \sim \mathcal{N}(0, [\alpha_t(1-\alpha_{t-1}) + (1-\alpha_t)]\mathbf{I}) & (8)\\
&= \mathcal{N}(0, (1-\alpha_t\alpha_{t-1})\mathbf{I}). & (4)
\end{aligned}
$$

因此可以混合两个高斯分布得到标准差为 $\sqrt{1-\alpha_t\alpha_{t-1}}$的混合高斯分布，然而Eq(3)中的 $\bar{z}_2$仍然是标准高斯分布。而任意时刻的 x_t满足 $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{a}_t}x_0, (1-\bar{a}_t)\mathbf{I})$.

一开始笔者一直不清楚为什么Eq(1)中diffusion的均值每次要乘上 $\sqrt{1-\beta_t}$.明明β_t只是方差系数，怎么会影响均值呢？替换为任何一个新的超参数，保证它<1，也能够保证值域并且使得最后均值收敛到0（但是方差并不为1）. 然而通过Eq(3)(4)，可以发现当 $T \to \infty, x_T \sim \mathcal{N}(0, \mathbf{I})$.所以$\sqrt{1-\beta_t}$的均值系数能够稳定保证 x_T最后收敛到方差为1的标准高斯分布，且在Eq(4)的推导中也更为简洁优雅。（注:很遗憾，笔者并没有系统地学习过随机过程，也许 $\sqrt{1-\beta_t}$就是diffusion model前向过程收敛到标准高斯分布的唯一解，读者有了解也欢迎评论）

Diffusion逆向（推断）过程

如果说前向过程(forward)是加噪的过程，那么逆向过程(reverse)就是diffusion的去噪推断过程。如果我们能够逐步得到逆转后的分布 $q(x_{t-1}|x_t)$，就可以从完全的标准高斯分布 $x_T \sim \mathcal{N}(0, \mathbf{I})$还原出原图分布 x_0.在文献[7]中证明了如果$q(x_t|x_{t-1})$满足高斯分布且 β_t足够小，$q(x_{t-1}|x_t)$仍然是一个高斯分布。然而我们无法简单推断$q(x_{t-1}|x_t)$，因此我们使用深度学习模型（参数为 θ，目前主流是U-Net+attention的结构）去预测这样的一个逆向的分布 p_θ（类似VAE）：

$$p_\theta(X_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t); \tag{5-1}$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{5-2}$$

虽然我们无法得到逆转后的分布 $q(x_{t-1}|x_t)$，但是如果知道 x_0，是可以通过贝叶斯公式得到 $q(x_{t-1}|x_t, x_0)$为：

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I})$$

过程如下：

$$\begin{aligned} q(x_{t-1}|x_t, x_0) &= q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \\ &\propto \exp\Bigg(-\frac{1}{2}\Big(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\overline{\alpha}_{t-1}}x_0)^2}{1-\overline{a}_{t-1}} - \\ &= \exp\Bigg(-\frac{1}{2}\Big(\underbrace{(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\overline{\alpha}_{t-1}})}_{x_{t-1}\text{方差}}x_{t-1}^2 - \underbrace{(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\overline{a}}}{1-\overline{\alpha}}}_{x_{t-1}\text{均值}} \end{aligned}$$

上式(7-1)巧妙地将**逆向** 过程全部变回了**前向**，即
$(x_{t-1}, x_0) \rightarrow x_t; \quad x_0 \rightarrow x_t; \quad x_0 \rightarrow x_{t-1}$，而(7-2)分别写出其对应的高斯概率密度函数，(7-3)则整理成了 x_{t-1}的高斯分布概率密度函数形式。一般的高斯概率密度函数的指数部分应该写为
$\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \exp\left(-\frac{1}{2}(\frac{1}{\sigma^2}x^2 - \frac{2\mu}{\sigma^2}x + \frac{\mu^2}{\sigma^2})\right)$，因此稍加整理我们可以得到(6)中的方差和均值为：

$$\frac{1}{\sigma^2} = \frac{1}{\tilde{\beta}_t} = (\frac{\alpha_t}{\beta_t} + \frac{1}{1-\overline{\alpha}_{t-1}}); \quad \tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\cdot\beta_t \tag{8-1}$$

$$\frac{2\mu}{\sigma^2} = \frac{2\tilde{\mu}_t(x_t, x_0)}{\tilde{\beta}_t} = (\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\overline{a}_{t-1}}}{1-\overline{\alpha}_{t-1}}x_0); \tag{9}$$

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{a}_t(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}x_t + \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}x_0. \tag{8-2}$$

根据**特性2**，我们得知 $x_0 = \frac{1}{\sqrt{\overline{a}_t}}(x_t - \sqrt{1-\overline{a}_t}\overline{z}_t)$，因此带入(8-2)可以得到

$\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{a}_t}}\overline{z}_t)$，

其中高斯分布 $\bar{z}_t$为深度模型所预测的噪声（用于去噪），可看做为 $z_\theta(x_t, t)$，即得到：

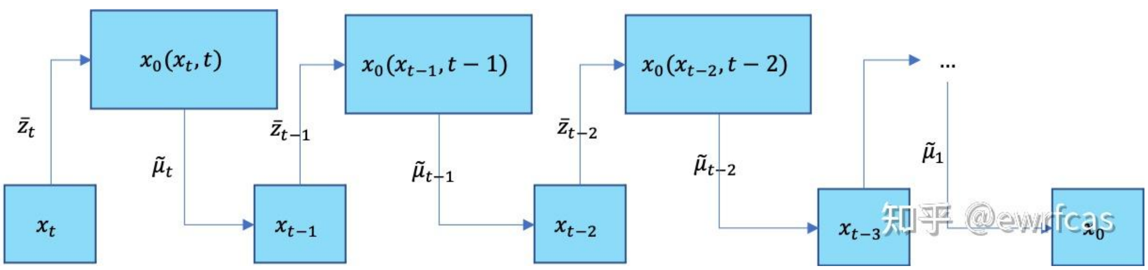
$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{a_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}} z_\theta(x_t, t)).$$

这样一来,DDPM的每一步的推断可以总结为：

1) 每个时间步通过 x_t和t来预测高斯噪声$z_\theta(x_t, t)$，随后根据(9)得到均值 $\mu_\theta(x_t, t)$.

2) 得到方差 $\Sigma_\theta(x_t, t)$，DDPM中使用untrained $\Sigma_\theta(x_t, t) = \tilde{\beta}_t$，且认为 $\tilde{\beta}_t = \beta_t$和 $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$结果近似，在GLIDE中则是根据网络预测trainable方差 $\Sigma_\theta(x_t, t)$.

3) 根据(5-2)得到 $q(x_{t-1}|x_t)$，利用重参数得到 x_{t-1}.



在x0和xt反复横跳的diffusion逆向过程

Diffusion训练

搞清楚diffusion的逆向过程之后，我们算是搞清楚diffusion的推断过程了。但是如何训练diffusion model以得到靠谱的$\mu_\theta(x_t, t)$和$\Sigma_\theta(x_t, t)$呢？通过对真实数据分布下，最大化模型预测分布的对数似然，即优化在 $x_0 \sim q(x_0)$下的 $p_\theta(x_0)$交叉熵：

$$\mathcal{L} = \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)].$$

从图4可以得知这个过程很像VAE，即可以使用变分下限(VLB)来优化负对数似然。由于KL散度非负，可得到：

$$
\begin{aligned}
-\log p_\theta(x_0) &\le -\log p_\theta(x_0) + D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0)) \\
&= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\right]; \quad \text{where} \quad p_\theta(x_{1:T}|x_0) = \\
&= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \underbrace{\log p_\theta(x_0)}_{\text{与}q\text{无关}}\right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right].
\end{aligned}
$$

对(11)左右取期望 $\mathbb{E}_{q(x_0)}$，利用到重积分中的Fubini定理：

$$
\mathcal{L}_{VLB} = \underbrace{\mathbb{E}_{q(x_0)}\left(\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]\right) = \mathbb{E}_{q(x_{0:T})}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]}_{Fubini\text{定理}} \ge \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)].
$$

能够最小化 $\mathcal{L}_{VLB}$即可最小化我们的目标损失(10)。

另一方面，通过Jensen不等式也可以得到一样的目标：

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)] && (13)\\
&= -\mathbb{E}_{q(x_0)} \log\left(p_\theta(x_0)\cdot\int p_\theta(x_{1:T})dx_{1:T}\right) && (14)\\
&= -\mathbb{E}_{q(x_0)} \log\left(\int p_\theta(x_{0:T})dx_{1:T}\right) && (15)\\
&= -\mathbb{E}_{q(x_0)} \log\left(\int q(x_{1:T}|x_0)\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}dx_{1:T}\right) && (16)\\
&= -\mathbb{E}_{q(x_0)} \log\left(\mathbb{E}_{q(x_{1:T}|x_0)}\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right) && (17)\\
&\le -\mathbb{E}_{q(x_{0:T})} \log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}; \qquad Jensen\text{不等式} && (18)\\
&= \mathbb{E}_{q(x_{0:T})} \log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} = \mathcal{L}_{VLB}. && (13)
\end{aligned}
$$

进一步对$\mathcal{L}_{VLB}$推导，可以得到熵与多个KL散度的累加，具体可见文献[8].这里我就复制一波Lil的博客中的推导过程：

$$
\begin{aligned}
L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \Big(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\cdot\frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\Big) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Big] \\
&= \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]
\end{aligned}
$$

进一步推导VLB，得到组合的KL散度和熵

也可写为：

$$
\begin{aligned}
\mathcal{L}_{VLB} &= L_T + L_{T-1} + \ldots + L_0 && (14\text{-}1) \\
L_T &= D_{KL}(q(x_T|x_0)||p_\theta(x_T)) && (14\text{-}2) \\
L_t &= D_{KL}(q(x_t|x_{t+1},x_0)||p_\theta(x_t|x_{t+1})); \qquad 1 \le t \le T-1 && (14\text{-}3) \\
L_0 &= -\log p_\theta(x_0|x_1). && (14\text{-}4)
\end{aligned}
$$

由于前向 q没有可学习参数，而 x_T则是纯高斯噪声， L_T可以当做常量忽略。而 L_t则可以看做拉近2个高斯分布 $q(x_{t-1}|x_t,x_0) = \mathcal{N}(x_{t-1};\tilde{\mu}(x_t,x_0),\tilde{\beta}_t\mathbf{I})$和 $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1};\mu_\theta(x_t,t),\Sigma_\theta)$，根据[多元高斯分布的KL散度求解]：

$$L_t = \mathbb{E}_q\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)||^2\right] + C,$$

其中C是与模型参数 θ无关的常量。吧(8-3)的 $\tilde{\mu}_t(x_t,x_0)$(9)的 $\mu_\theta(x_t,t)$和(3)的 x_t带入(15)可以得到：

$$
\begin{aligned}
L_t &= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\tilde{\mu}_t(x_t,x_0)-\mu_\theta(x_t,t)||^2\right] \\
&= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\frac{1}{\sqrt{\bar{a}_t}}(x_t-\frac{\beta_t}{\sqrt{1-\bar{a}_t}}\bar{z}_t)-\frac{1}{\sqrt{\bar{a}_t}}(x_t-\frac{\beta_t}{\sqrt{1-}}\right. \\
&= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t||\Sigma_\theta||_2^2)}||\bar{z}_t-z_\theta(x_t,t)||^2\right] \\
&= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t||\Sigma_\theta||_2^2)}||\bar{z}_t-z_\theta(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\bar{z}_t,t)||^2\right].
\end{aligned}
$$

从(16)可以看出，diffusion训练的核心就是取学习高斯噪声 $\bar{z}_t$, z_θ之间的MSE。

$L_0=-\log p_\theta(x_0|x_1)$相当于最后一步的熵，DDPM论文指出，从 x_1到 x_0应该是一个离散化过程，因为图像RGB值都是离散化的。DDPM针对 $p_\theta(x_0|x_1)$构建了一个离散化的分段积分累乘，有点类似基于分类目标的自回归(auto-regressive)学习。有兴趣的同学可以去参考原文。

DDPM将loss进一步简化为:

$$L_t^{simple}=\mathbb{E}_{x_0,\bar{z}_t}\left[||\bar{z}_t-z_\theta(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\bar{z}_t,t)||^2\right].$$

正如之前提过的，DDPM并没有将模型预测的方差 $\Sigma_\theta(x_t,t)$考虑到训练和推断中，而是通过untrained β_t或者(8-1) $\tilde{\beta}_t$代替。他们发现 Σ_θ可能导致训练的不稳定。

训练过程可以看做：

1）获取输入 x_0，从 $1\ldots T$随机采样一个 t.

2) 从标准高斯分布采样一个噪声 $\bar{z}_t\sim\mathcal{N}(0,\mathbf{I})$.

3) 最小化 $||\bar{z}_t-z_\theta(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\bar{z}_t,t)||$.

最后再附上DDPM提供的训练/测试（采样）流程图

Algorithm 1 Training

1: **repeat**
2: $\quad\mathbf{x}_0\sim q(\mathbf{x}_0)$
3: $\quad t\sim\mathrm{Uniform}(\{1,\ldots,T\})$
4: $\quad\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
5: $\quad$Take gradient descent step on
$\qquad\nabla_\theta\left\|\boldsymbol{\epsilon}-\mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t)\right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
2: **for** $t=T,\ldots,1$ **do**
3: $\quad\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ if $t>1$, else $\mathbf{z}=\mathbf{0}$
4: $\quad\mathbf{x}_{t-1}=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t,t)\right)+\sigma_t\mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

DDPM训练测试算法流程图（来源：DDPM）

加速Diffusion采样和方差的选择(DDIM)

DDPM的高质量生成依赖于较大的 T（一般为1000或以上），这就导致diffusion的前向过程非常缓慢。在denoising diffusion implicit model (DDIM)[9]中提出了一种牺牲多样性来换取更快推断的手段。

根据**特性2** 和独立高斯分布可加性，我们可以得到 x_{t-1}为：

$$\begin{aligned} x_{t-1} &= \sqrt{\overline{\alpha}_{t-1}}x_0 + \sqrt{1-\overline{a}_{t-1}}\overline{z}_{t-1} \\ &= \sqrt{\overline{\alpha}_{t-1}}x_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\overline{z}_t + \sigma_t z_t \\ &= \sqrt{\overline{\alpha}_{t-1}}x_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}(\frac{x_t-\sqrt{\overline{a}_t}x_0}{\sqrt{1-\overline{\alpha}_t}}) + \sigma_t z_t \end{aligned}$$

$$q_\sigma(x_{t-1}|x_t,x_0) = \mathcal{N}(x_{t-1}; \sqrt{\overline{a}_{t-1}}x_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}(\frac{x_t-\sqrt{\overline{a}_t}x_0}{\sqrt{1-\overline{\alpha}_t}}), \sigma_t^2$$

不同于(6)和(9)，(18)**将方差 σ_t^2引入到了均值中**，当 $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$时，(18)等价于(6)。

在DDIM中吧由(18)经过贝叶斯得到的 $q_\sigma(x_t|x_{t-1},x_0)$称为非马尔科夫过程，因为 x_t的概率同时依赖于 x_{t-1}和 x_0。（笔者并不了解刻意强调这个非马尔科夫是原因，也许是为了使得(18)中方差出现在均值合理化？）DDIM进一步定义了 $\sigma_t(\eta)^2 = \eta\cdot\tilde{\beta}_t$.当 $\eta=0$时，diffusion的sample过程会丧失所有随机性从而得到一个deterministic的结果（但是可以改变 x_T）。而 $\eta=1$则DDIM等价于DDPM(使用 $\tilde{\beta}_t$作为方差的版本).用随机性换取生成性能的类似操作在GAN中也可以通过对latent code操作实现。

对于方差 σ_t^2的选择，我们在这里重新整理一下

DDPM：

1） $\sigma_{t,\theta}^2 = \Sigma_\theta(x_t,t)$相当于模型学习的方差，DDPM称为learned，实际没有使用（但是GLIDE使用的是这种方差）。

2） $\sigma_{t,s}^2 = \tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$，由(8-1)得到，DDPM称为fixedsmall，用于celebahq和lsun。

3） $\sigma_{t,l}^2 = \beta_t$，DDPM称为fixedlarge，用于cifar10，注意 $\sigma_{t,l} > \sigma_{t,s}$，**fixedlarge的方差大于fixedsmall的** 。

DDIM：

$\sigma_t(\eta)^2 = \eta \cdot \tilde{\beta}_t$，DDIM所选择的是基于fixedsmall版本上再乘以一个 η.

假设总的采样步 $T = 1000$，间隔是 Q，DDIM采样的步数为 $S = T/Q$，S 和 η 的实验结果如下：

	S	CIFAR10 (32×32)					CelebA (64×64)				
		10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
$\hat{\sigma}$		367.43	133.37	32.72	9.99	**3.17**	299.71	183.83	71.71	45.20	**3.26**

来自DDIM的FID结果（不同步数S和方差设置）

可以发现在 S很小的时候 $\eta = 0$取得了最好的结果。值得一提的是， $\eta = 1$是等价于DDPM的fixedsmall版本。而 $\hat{\sigma} = \sqrt{\beta_t}$表示的是DDPM的fixedlarge版本。因此当 T足够大的时候使用更大的方差 σ_t^2能取得更好的结果。

参考

1. ^https://lilianweng.github.io/posts/2021-07-11-diffusion-models
2. ^Nichol, Alex, et al. "Glide: Towards photorealistic image generation and editing with text-guided diffusion models." arXiv preprint arXiv:2112.10741 (2021).
3. ^Ramesh, Aditya, et al. "Hierarchical text-conditional image generation with clip latents." arXiv preprint arXiv:2204.06125 (2022).
4. ^Saharia, Chitwan, et al. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding." arXiv preprint arXiv:2205.11487 (2022).
5. ^Rombach, Robin, et al. "High-Resolution Image Synthesis with Latent Diffusion Models." arXiv preprint arXiv:2112.10752 (2021).
6. ^Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

7. ^Feller, William. "On the theory of stochastic processes, with particular reference to applications." Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, 1949.
8. ^Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." International Conference on Machine Learning. PMLR, 2015.
9. ^Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).