论文解读：*Denoising Diffusion Probabilistic Models*

DDPM是扩散模型的奠基之作，它并非首先提出扩散模型的文章，但是是首先将扩散模型用于图像生成领域的关键论文。本文将尝试对其进行剖析

注意：因为我阅读本篇论文前并没有提前查阅专有名词的翻译，所以导致本文中的一些用词与常见的翻译存在偏差。下面给出一张对照表：

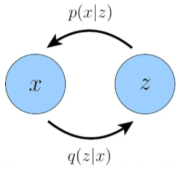
- **正向过程**：Forward process，通常译作前向过程
- **反向过程**：Reverse process，也译作后向过程
- **高斯分布**：就是正态分布。注意文中的高斯分布是多维的，因此采用向量的标识形式；比如多维标准正态分布记作$\mathcal{N} \sim (z; \mathbf{0}, \mathbf{I})$，其中$\mathbf{I}$表示单位矩阵

建议阅读顺序：如果是第一次接触DDPM，可以先阅读Introduction部分，对DDPM的原理有一个大概的了解，然后再阅读前置知识部分，补充数学基础。

前置知识：从VAE到DDPM

VAE的基本原理

x可以看作是一张图片，而z是一个（一些）变量（latent variable，隐变量），满足某种自定义的分布，一般选择标准高斯分布$\mathcal{N} \sim (z; \mathbf{0}, \mathbf{I})$。模型的目标是通过$z$得到$x$。而为了实现这一点，我们建立两个步骤：



- **Encoder**：$q(z|x)$，给定一张图片x，生成一些维度较低的embedding z
- **Decoder**：$p(x|z)$，根据高斯分布的一个样本z，生成图片x

这两个部分将维度较高的图片和维度较低且满足常见分布的数据联系起来。VAE的目标就是拟合出一个$p_\phi(x)$，尽可能地去逼近x的真实分布$p(x)$

其实VAE可以看作是DDPM的一个链，DDPM就相当于是把T个VAE串在了一起

$p(x)$下界

$$
\begin{aligned}
\log p(x) &= \log \int p(x,z) dz \\
&= \log \int \frac{p(x,z) q_\phi(z|x)}{q_\phi(z|x)} dz \\
&= \log \mathbb{E}_{q_\phi(z|x)} \left[\frac{p(x|z)}{q_\phi(z|x)} \right] && \left[\frac{p(A)}{p(B)} \right] \text{对} B \text{的期望} E_{P(B)} \left[\frac{p(A)}{p(B)} \right] = \int \frac{p(A)}{p(B)} \cdot p(B) dB \\
&\geq \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p(x,z)}{q_\phi(z|x)} \right] && \text{琴生不等式，可画图直观理解}
\end{aligned} \tag{1}
$$

这个下界还有另外一种展开方法，同样可以证明上面的结论：

$$
\begin{aligned}
\log p(x) &= \log p(x) \int q_\phi(z|x) dz && \int p(A) dA = 1 \\
&= \int q_\phi(z|x) \log p(x) dz \\
&= \mathbb{E}_{q_\phi(z|x)}[\log p(x)] && \text{期望的定义} E_{p(B)}[p(A)] = \int p(A)p(B)dB \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p(x,z)}{p(z|x)} \right] \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p(x,z) q_\phi(z|x)}{p(z|x) q_\phi(z|x)} \right] \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p(x,z)}{q_\phi(z|x)} \right] + \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{q_\phi(z|x)}{p(z|x)} \right] \\
&= \underbrace{\mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p(x,z)}{q_\phi(z|x)} \right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_\phi(z|x) \| p(z|x))}_{\text{prior matching term}} && \text{KL散度定义}
\end{aligned} \tag{2}
$$

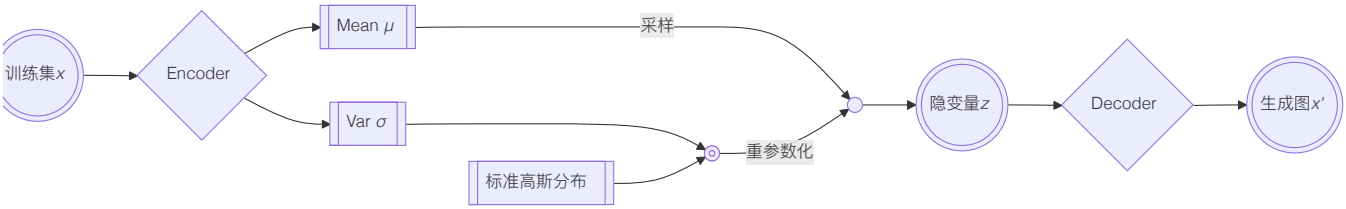
上面的展开结果称为ELBO，而我们的目标是最大化ELBO。它可以拆分为如下两个子目标：

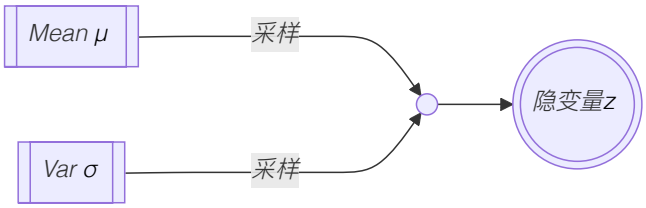
- **最小化 prior matching term**：$\log p(x)$与其下界相差1个KL散度，它代表的是encoder拟合出的$p_\phi(z|x)$和真实$p(z|x)$间的差距；当encoder完美拟合时，两者相等
- **最大化 reconstruction term**：想要让decoder以最大的可能性从隐变量z生成真实的原始数据x，就要最大化reconstruction term

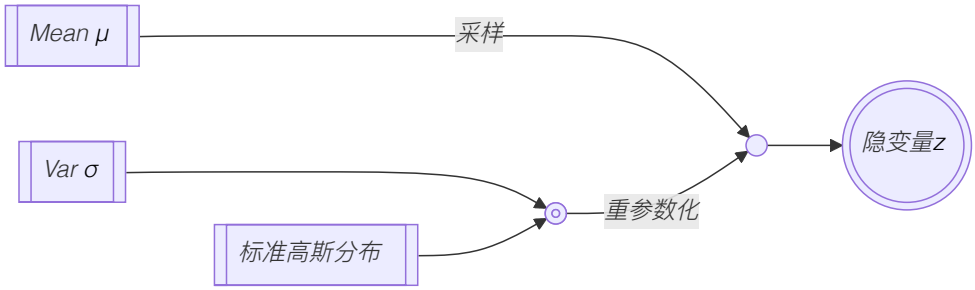
因为真实分布$p(z|x)$是未知的，所以我们最小化prior matching term的方式是：在encoder将真实数据x映射到隐变量z上时，让z尽可能满足某种指定的分布，通常为$\mathcal{N}(z; \mathbf{0}, \mathbf{I})$

VAE模型结构

下面的流程图展示了VAE的训练和生成过程。给定一张训练图x，首先经过编码器将x映射到z上，拟合标准正态分布（尽可能让均值接近0向量，方差接近单位向量），这样就得到了隐变量z。而解码器则可以根据隐变量中随机抽取的一个点生成相应的图片x'



> ***重参数化***：*如果我们不采用重参数化技巧（如下图），因为$z \sim \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x)\mathbf{I})$的采样是一个随机过程，包含需要优化的参数$\phi$，但是随机过程对参数$\phi$是不可导的*
>
> 
>
>
> *因此，我们引入一个标准正态分布$\epsilon \sim \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I})$，在这个正态分布中随机取值，将它与方差相乘后再与均值相加：*
>
> $$z^l = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon \tag{3}$$
>
> *得到的z^l不改变原先分布z的均值与方差，并将ϕ从随机过程中剥离出来，这样就保证了参数ϕ可导*



马尔可夫VAE

马尔可夫VAE其实就是将多个VAE过程连在一起，数学性质的证明和VAE是同理的

VDM

DDPM是VDM的一种，只在一些细节上有差别。而VDM实际上是MHVAE（马尔可夫VAE）增加了一些限制条件

假设我们的训练图像为x，中间状态为z_t（$1 \leq t < T$），最终状态为z_T。限制条件如下：

- x和所有隐变量z_t的维度相同
- 所有encoder $q(z_t|z_{t-1})$都是预先定义好的高斯分布模型，不需要学习（上一个约束，即，维度相同，保证了这一条约束的可行性）
- 最终状态z_T是标准高斯分布

在DDPM中，我们人为定义$q(\mathbf{x}_t|\mathbf{x}_{t-1})$满足高斯分布$\mathcal{N}(\mathbf{x}_t; \sqrt{a_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$，其中$\alpha$是超参，文中是人为定义的，所以需要学习的部分只有$p$。当然也可以通过模型学习。

> *我们来进一步理解一下$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{a_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$，$\alpha$是一个小于1的数，这意味着，每经过一次$q$，均值的期望都会缩小一点，更加接近0，也就离标准高斯分布更接近了一点*

有了上面这些前置知识，你就可以较为直观地理解DDPM的基本思路了。DDPM中的公式几乎都可以在上面找到相应的原型，因此虽然证明可能依然存在一定困难，但我们暂时就先不细究了，有一个直观的理解就可以

Introduction

本文提出了扩散概率模型，这是一个马尔可夫链（Markov chain）。包含正向过程和反向过程，正向过程（扩散过程）$q(\mathbf{x}_t|\mathbf{x}_{t-1})$是在一张图片上不断添加高斯噪声，有具体的表达式可以计算；而我们的目标就是估计反向过程$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$的概率密度函数（PDF），这样我们就可以最终推导出我们想要的结果$p_\theta(\mathbf{x}_0)$

> ***马尔可夫链**：简单来说，马尔可夫链一个序列，这个序列的每一个节点的状态都只与上一个状态有关，那么它就是一个马尔克夫链。可以理解为滞后期为1的相关序列，数学表达为：*
>
> $$P(X_{t+1}|X_0, \ldots, X_{t-2}, X_{t-1}, X_t) = P(X_{t+1}|X_t) \tag{4}$$

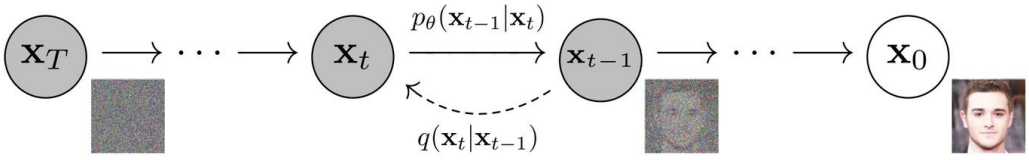


Figure 2: The directed graphical model considered in this work.

Background

扩散模型

文章中将联合概率密度函数$p_\theta(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T)$写作$p_\theta(\mathbf{x}_{0:T})$。由此，我们可以得到$\mathbf{x}_0$的边缘概率密度函数：

$$\begin{aligned} p_\theta(\mathbf{x}_0) &= \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_T \\ &= \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \end{aligned} \tag{5}$$

反向过程

上面的$p_\theta(\mathbf{x}_{0:T})$就是反向过程的概率分布。利用乘法公式和马尔可夫链的定义可以给出它的概率：

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T) p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T) p_\theta(\mathbf{x}_{T-2}|\mathbf{x}_{T-1}\mathbf{x}_T) \dots p_\theta(\mathbf{x}_0|\mathbf{x}_{1:T-1}) \\ &= p(\mathbf{x}_T) p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T) p_\theta(\mathbf{x}_{T-2}|\mathbf{x}_{T-1}) \dots p_\theta(\mathbf{x}_0|\mathbf{x}_1) \\ &= p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \end{aligned} \tag{6}$$

其中，$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$的分布为：

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{7}$$

在反向过程中，转移概率分布函数为关于$\mathbf{x}_{t-1}$的高斯分布，其均值$\mu_\theta(\mathbf{x}_t, t)$和协方差矩阵$\mathbf{\Sigma}_\theta(\mathbf{x}_t, t)$是$\mathbf{x}_t, t$的函数，它们就是我们需要学习的参数。

> ***乘法公式***
>
> *对于联合概率$P(A_1 A_2 \dots A_n)$，我们有：*
>
> $$\begin{aligned} P(A_1 A_2 \dots A_n) &= P(A_1 A_2 \dots A_{n-1}) P(A_n|A_1 A_2 \dots A_{n-1}) \\ &= \dots \\ &= P(A_1) P(A_2|A_1) P(A_3|A_1 A_2) \dots P(A_n|A_1 A_2 \dots A_{n-1}) \end{aligned} \tag{8}$$

正向过程

正向过程（Forward process）也叫做扩散过程（diffusion process），是不断添加高斯噪声的过程。同样是一个马尔可夫链，起点和反向过程相反，为$q(\mathbf{x}_0)$。我们不难给出正向过程的条件概率：

$$\begin{aligned} q(\mathbf{x}_{0:T}) &= q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ &= q(\mathbf{x}_0) q(\mathbf{x}_{1:T}|\mathbf{x}_0) \\ \Rightarrow q(\mathbf{x}_{1:T}|\mathbf{x}_0) &= \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \end{aligned} \tag{9}$$

满足如下的概率分布：

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \tag{10}$$

其中$\beta_{1:T}$是添加高斯噪声的方差表（variance schedule），$\mathbf{I}$是单位矩阵

注意到，论文在反向过程给出的是$p_\theta(\mathbf{x}_{0:T})$，而正向过程给出的则是$q(\mathbf{x}_{1:T}|\mathbf{x}_0)$。是因为在正向过程中，初始图像$\mathbf{x}_0$是已知的，给出单步转移概率更能体现噪声的添加过程

最大似然

训练模型的过程是对$p_\theta(\mathbf{x}_0)$最大似然估计进行优化的过程。具体来说，优化的是usual variational bound on negative log likelihood

优化的目标是让$p_\theta(\mathbf{x}_0)$的期望最大，即$-\log p_\theta(\mathbf{x}_0)$的期望最小（采用对数是因为对数可以将乘法降维为加法，简化求导且不会影响结果）：

$$
\begin{aligned}
\mathbb{E}_q[-\log p_\theta(\mathbf{x}_0)] &\leq \mathbb{E}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \\
&= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] \\
&=: L
\end{aligned} \tag{11}
$$

L重写

前向过程有一个重要的性质，就是在任意时刻t，都有：

$$
q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\overline{\alpha}_t}\mathbf{x}_0, (1-\overline{\alpha}_t)\mathbf{I}) \tag{12}
$$

其中$\alpha_t := 1-\beta_t$，$\overline{\alpha}_t := \prod_{s=1}^{t}\alpha_s$

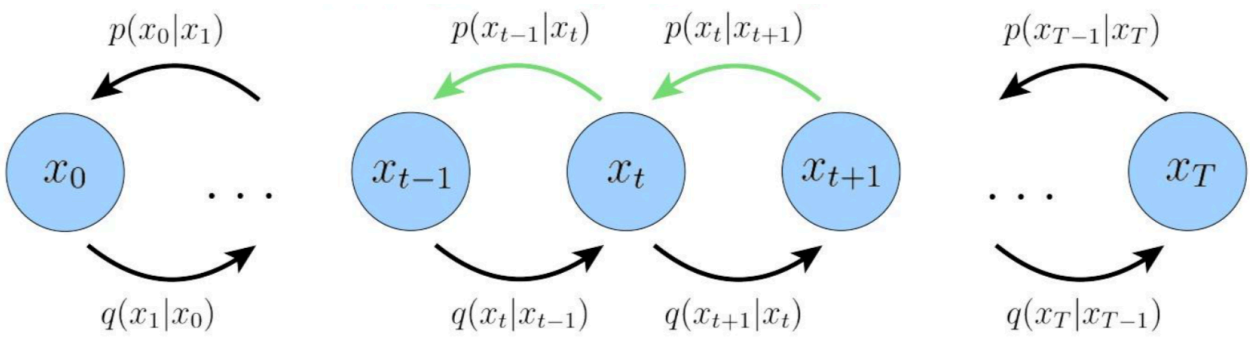
然后，我们将L改写为（推导不难，暂时先不细究）：

$$
L = \mathbb{E}\left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\right] \tag{13}
$$

> *其中，D_{KL}是Kullback-Leibler散度，用于衡量两个概率分布之间的差异，定义为：*
>
> $$
> D_{\mathrm{KL}}(P||Q) = \int_{-\infty}^{\infty} p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x \tag{14}
> $$
>
> *也就是说，L中的D_{KL}比较的是反向过程的$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$和正向过程的后验（forward process posteriors），当给定$\mathbf{x}_0$时，不难给出：*
>
> $$
> q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \tag{15}
> $$
>
> *其中*
>
> $$
> \begin{aligned}
> \tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) &:= \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\overline{\alpha}_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\mathbf{x}_t \\
> \tilde{\beta}_t &:= \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t
> \end{aligned} \tag{16}
> $$
>
> *也是高斯分布，因此，所有KL散度都是高斯间的比较，可以用闭表达式的Rao-Blackwellized计算（暂时先不细究）*

我们对照下图来直观理解一下这个公式：

- 左上方的$p(x_0|x_1)$对应的是上面公式中的L_0（由x_T到x_0的最后一步）；
- 右下角的$q(x_t|x_{T-1})$对应的是上面公式中的L_T（虽然写法不一样，但本质上就是比较$\mathbf{x}_T$和$\mathbf{x}_{T-1}$）；
- 剩下的所有箭头都是L_{t-1}里的项，这也是我们优化的重心



注意到，当$T=1$时，$\mathbf{x}_1$就是隐变量$\mathbf{z}$。L_{t-1}项消失，L_0就是VAE中的prior matching term，而L_T就是VAE中的reconstruction term。VAE实际上就是DDPM的一种特殊情况

Diffusion Models and Denoising Autoencoders 扩散模型和自动降噪编码器

非常好，我们终于把背景看完了。虽然从数学上来看，扩散模型似乎已经非常明确且唯一，但在工程实践中，还有非常大的自由度。比如，我们需要选择正向过程的方差β_t和反向过程的模型结构等

L_0和L_T

- L_T**为定值**：因为我们将正向过程的方差β_t定为常量，因此q是确定的，没有可学习的参数。所以L_T是定值，没有优化空间
- L_0**优化空间较小**：而L_0虽然有优化空间，但因为只有一项，所以实际上优化效果并不明显，因此也不是我们优化的重点

反向过程和$L_{1:T-1}$

想要优化$L_{1:T-1}$，有两个需要考虑的分布——$q((\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$和$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$。前者是人为确定的，所以我们的任务就是优化$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$，让它尽可能接近$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$。其中$p_\theta$满足如下分布：

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)=\mathcal{N}(\mathbf{x}_{t-1};\mu_\theta(\mathbf{x}_t,t),\mathbf{\Sigma}_\theta(\mathbf{x}_t,t)) \tag{17}$$

方差

我们将$\mathbf{\Sigma}_\theta(\mathbf{x}_t,t)=\sigma_t^2\mathbf{I}$定为未训练的时间相关的常数，实验表明，取$\sigma_t^2=\beta_t$或$\sigma_t^2=\tilde{\beta}_t=\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$的结果是类似的。选择前者在$\mathbf{x}_0\sim\mathcal{N}(0,\mathbf{I})$情况下最优，后者在$\mathbf{x}_0$为某确定的点时最优

均值

我们先对L_{t-1}进行改写：

$$L_{t-1}=\mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0)-\mu_\theta(\mathbf{x}_t,t)\|^2\right]+C \tag{18}$$

其中C是一个常数，不可优化。因此我们的优化目标是让μ_θ尽可能地接近$\tilde{\mu}_t$

上面我们提到$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)=\mathcal{N}(\mathbf{x}_{t-1};\mu_\theta(\mathbf{x}_t,t),\sigma_t^2\mathbf{I})$，我们对$L_{t-1}$进行重参数化，让$\mathbf{x}_t(\mathbf{x}_0,\epsilon)=\sqrt{\overline{\alpha}_t}\mathbf{x}_0+\sqrt{1-\overline{\alpha}_t}\epsilon$（$\epsilon$为标准正态分布$\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$）：

$$\begin{aligned}L_{t-1}-C&=\mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\left\|\tilde{\mu}_t\left(\mathbf{x}_t(\mathbf{x}_0,\epsilon),\frac{1}{\sqrt{\overline{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0,\epsilon)-\sqrt{1-\overline{\alpha}_t}\epsilon)\right)-\mu_\theta(\mathbf{x}_t(\mathbf{x}_0,\epsilon),t)\right\|^2\right]\\&=\mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t(\mathbf{x}_0,\epsilon)-\frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon\right)-\mu_\theta(\mathbf{x}_t(\mathbf{x}_0,\epsilon),t)\right\|^2\right]\end{aligned} \tag{19}$$

训练算法

为了让L尽可能小，我们需要让μ_θ尽可能拟合$\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t(\mathbf{x}_0,\epsilon)-\frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon\right)$。因为$\mathbf{x}_0$是给定的，因此它可以作为模型的输入。我们选择如下重参数化：

$$\mu_\theta(\mathbf{x}_t,t)=\tilde{\mu}_t\left(\mathbf{x}_t,\frac{1}{\sqrt{\overline{\alpha}_t}}(\mathbf{x}_t-\sqrt{1-\overline{\alpha}_t}\epsilon_\theta(\mathbf{x}_t))\right)=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t,t)\right) \tag{20}$$

所以实际上我们需要学习的参数就是ϵ_θ，也就是下面Algorithm 2的学习过程。因为$\mathbf{x}_{t-1}=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t,t)\right)+\sigma_t\mathbf{z}$（$\mathbf{z}\sim(\mathbf{0},\mathbf{I})$），loss可以进一步化简：

$$\mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{\beta^2}{2\sigma_t^2\alpha_t(1-\overline{\alpha}_t)}\|\epsilon-\epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0+\sqrt{1-\overline{\alpha}_t}t\epsilon,t)\|\right] \tag{21}$$

这个式子就代表了图像生成的质量。因为前面的常数系数没什么用，所以可以进一步化简为

$$L_{\text{simple}}(\theta):=\mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon-\epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0+\sqrt{1-\overline{\alpha}_t}\epsilon,t)\|\right] \tag{22}$$

Algorithm 1 Training

1. **循环**
2. 采样原始图像$\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3. 采样时间步$t \sim \text{Uniform}(\{1, \dots, T\})$
4. 增加噪声$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5. 计算损失，更新模型$\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)\|^2$
6. **直到**收敛

Algorithm 2 Sampling

1. 得到一个高斯噪声$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2. **for** $t = T, \dots, 1$ **do**
    - 当$t > 1$时，$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，否则$\mathbf{z} = \mathbf{0}$
    - $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
3. **end for**
4. **return** $\mathbf{x}_0$
