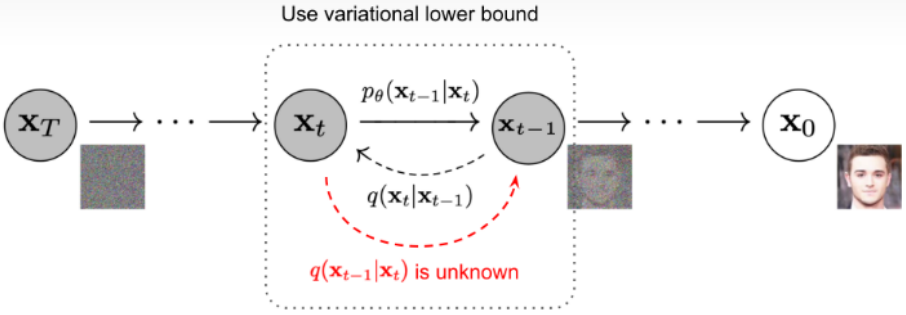


DDPM解读（一）| 数学基础，扩散与逆扩散过程和训练推理方法

卡卡猡特
AI摸鱼选手 ZJU博士在读

今天为大家介绍一下最近在生成模型+中逐渐受到重视的扩散模型DDPM（全称Denoising Diffusion Probabilistic Models)。由于扩散模型+相较于普通的深度学习模型，数学难度大很多，因此该模型理解起来非常吃力。我主要根据博客What are Diffusion Models?中的讲解，同时加上一些数学基础+知识和自己的理解，写成本系列博文。如果我的理解是有错误的或者公式有误，欢迎指正。

目前，已经有几种diffusion-based generative models被提出，包括diffusion probabilistic models (Sohl-Dickstein et al., 2015), noise-conditioned score network (**NCSN**; Yang & Ermon, 2019), and denoising diffusion probabilistic models (**DDPM**; Ho et al. 2020)。现在，最主流的DDPM模型都是基于2020年的文章《Denoising Diffusion Probabilistic Models+》，作者Jonathan Ho等人来自UC Berkeley.

一、数学基础

- 部分内容来自：bilibili.com/video/BV1b...

1. 先验概率和后验概率

- 先验概率+：根据以往经验和分析得到的概率,如全概率公式+ 中的,它往往作为“由因求果”问题中的“因”出现，如 $q(\mathbf{x}_t|\mathbf{x}_{t-1})$.
- 后验概率+：指在得到“结果”的信息后重新修正的概率,是“执果寻因”问题中的“因", 如 $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

2. 条件概率+的一般形式

$$P(A,B,C) = P(C|B,A)P(B,A) = P(C|B,A)P(B|A)P(A)$$
$$P(B,C|A) = P(B|A)P(C|A,B)$$

3. 马尔可夫链+条件概率形式

马尔科夫链指当前状态的概率只与上一时刻有关，例如如满足马尔可夫关系 $A \to B \to C$ ，那么有：

$$P(A,B,C) = P(C|B,A)P(B,A) = P(C|B)P(B|A)P(A)$$
$$P(B,C|A) = P(B|A)P(C|B)$$

4. 高斯分布的KL散度公式+

知乎 首发于 人工智能技术学习

$$KL(p,q) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

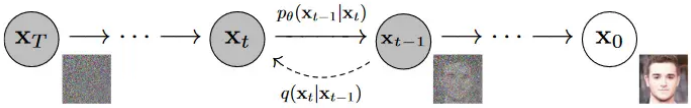
5. 参数重整化（重参数技巧）

若希望从高斯分布 $N(\mu, \sigma^2)$ 中采样，可以先从标准分布 $N(0,1)$ 采样出 z，再得到 $\sigma * z + \mu$，这就是我们想要采样的结果。这样做的好处是将随机性转移到了 z 这个常量上，而 σ 和 μ 则当作仿射变换网络的一部分。

二、DDPM模型

1. 模型总览

如下图所示。DDPM模型主要分为两个过程：forward加噪过程（从右往左）和reverse去噪过程（从左往右）。加噪过程意思是指向数据集的真实图片中逐步加入高斯噪声，而去噪过程是指对加了噪声的图片逐步去噪，从而还原出真实图片。加噪过程满足一定的数学规律，而去噪过程则采用神经网络来学习。这么一来，神经网络就可以从一堆杂乱无章的噪声图片中生成真实图片了。



2. Diffusion扩散过程（Forward加噪过程）

- **逐步加噪过程**

给定初始数据分布 $x_0 \sim q(x)$，我们定义一个前向扩散过程（forward diffusion process）：我们向数据分布中逐步添加高斯噪声，加噪过程持续 T 次，产生一系列带噪声图片 $x_1, \ldots, x_T$。在由 x_{t-1} 加噪至 x_t 的过程中，噪声的标准差/方差是以一个在区间 $(0,1)$ 内的固定值 β_T 来确定的，均值是以固定值 β_T 和当前时刻的图片数据 x_{t-1} 来确定的。

> Given a data point sampled from a real data distribution $x_0 \sim q(x)$, let us define a *forward diffusion process* in which we add small amount of Gaussian noise to the sample in T steps, producing a sequence of noisy samples $x_1, \ldots, x_T$. The step sizes are controlled by a variance schedule $\{\beta_T \in (0,1)\}_{t=1}^T$.

以上描述的加噪过程可以写成公式：

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

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

上式意思是：由 $\mathbf{x}_{t-1}$ 得到 $\mathbf{x}_t$ 的过程 $q(\mathbf{x}_t|\mathbf{x}_{t-1})$，满足分布 $\mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$。（这个分布是指以 $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$ 为均值，β_t 为方差的高斯分布）。因此我们看到这个噪声只由 β_T 和 x_{t-1} 来确定，是一个固定值而不是一个可学习过程。因此，只要我们有了 x_0，并且提前确定每一步的固定值 $\beta_1, \ldots, \beta_T$，我们就可以推出任意一步的加噪数据 $x_1, \ldots, x_T$。这里Forward加噪过程是一个马尔科夫链过程。

- **加噪结果**

随着 t 的不断增大，最终原始数据 x_0 会逐步失去它的特征。最终当 $T \to \infty$ 时，x_T 趋近于一个各向独立的高斯分布。从视觉上来看，就是将原本一张完好的照片加噪很多步后，图片几乎变成了一张完全是噪声的图片。

> The data sample x_0 gradually loses its distinguishable features as the step t becomes larger. Eventually when $T \to \infty$, x_T is equivalent to an isotropic Gaussian

知乎 首发于 人工智能技术学习

- **任意时刻数据$\mathbf{x}_t$的计算**

在逐步加噪的过程中，我们其实并不需要一步一步地从 $\mathbf{x}_0, \mathbf{x}_1, \ldots$ 去迭代得到 $\mathbf{x}_t$ 。事实上，我们可以直接从 x_0 和固定值序列 $\{\beta_T \in (0,1)\}_{t=1}^T$ 直接计算得到。

我们定义 $\alpha_t = 1 - \beta_t$ ， $\bar{\alpha}_t = \prod_{i=1}^T \alpha_i$ ，那么根据参数重整化技巧，我们得到：

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} && ; \text{where } \mathbf{z}_{t-1}, \mathbf{z}_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\mathbf{z}}_{t-2} && ; \text{where } \bar{\mathbf{z}}_{t-2} \text{ merges two Gaussians (*).} \\
&= \cdots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z} \\
q(\mathbf{x}_t \mid \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})
\end{aligned}
$$

(*) 这里注意：当我们合并两个均值都为0，方差分别为 σ_1^2 和 σ_2^2 的高斯分布$^+$ $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ 和 $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$ 时，我们得到的新的高斯分布为 $\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1}$ 。因此可以通过参数重整化，变成只含一个随机变量$^+$ z 构成的 $\sqrt{1-\alpha_t\alpha_{t-1}}\bar{\mathbf{z}}_{t-2}$ 形式。

> (*) Recall that when we merge two Gaussians with different variance, $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ and $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$, the new distribution is $\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1}$. Here the merged standard deviation is $\sqrt{1-\alpha_t\alpha_{t-1}}\bar{\mathbf{z}}_{t-2}$.

因此由公式可以看出，只要我们有了 x_0 和固定值序列 $\{\beta_T \in (0,1)\}_{t=1}^T$ ，再从标准分布 $N(0,1)$ 采样出一个 z ，我们就可以直接计算出 x_t 了。

- **β_t 的取值设置**

一般地，我们随着 t 的增大，数据越接近随机的高斯分布， β_t 的取值的取值就越大。所以 $\beta_1 < \beta_2 < \ldots < \beta_T$.

同时，上面我们已经提到 $\{\beta_T \in (0,1)\}_{t=1}^T$,即 β_1， $\beta_2, \ldots, \beta_T$ 的取值在区间 $(0,1)$ 内，因此 α_1， $\alpha_2, \ldots, \alpha_T$ 的取值也在区间 $(0,1)$ 内。所以 $\bar{\alpha}_1 > \bar{\alpha}_2 > \ldots > \bar{\alpha}_T$.

> Usually, we can afford a larger update step when the sample gets noisier$^+$, so $\beta_1 < \beta_2 < \ldots < \beta_T$ and therefore $\bar{\alpha}_1 > \ldots > \bar{\alpha}_T$.

3. Reverse Diffusion逆扩散过程$^+$（Reverse去噪过程）

- **逆扩散过程近似模型p_θ**

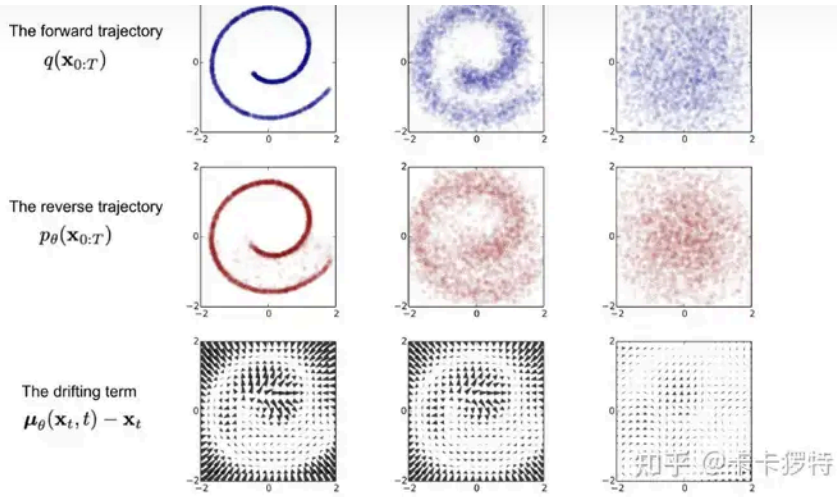
如果我们能将上述过程转换方向，即从 $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 中采样，那么我们就可以从一个随机的高斯分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 中重建出一个真实的原始样本，也就是从一个完全杂乱无章的噪声图片中得到一张真实图片。但是，由于需要从完整数据集中找到数据分布$^+$，我们没办法很简单地预测 $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ ，因此我们需要学习一个模型 p_θ 来近似模拟这个条件概率，从而运行逆扩散过程。

$$
p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)
$$

$$
p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sum_\theta(\mathbf{x}_t, t)\right)
$$

下图为扩散过程$^+$和逆扩散过程的示例：

知乎 首发于 人工智能技术学习



- **后验扩散条件概率**$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$

在逆扩散过程中，如果我们给定了 $\mathbf{x}_t$ 和 $\mathbf{x}_0$,那我们是可以计算出 $\mathbf{x}_{t-1}$ 的，即后验扩散条件概率 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ 是可以计算的。（注意：这里不是 $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 。如果不需要 $\mathbf{x}_0$ ，那我们可以直接用公式表达逆扩散过程，直接由 $\mathbf{x}_t$ 推出 $\mathbf{x}_{t-1}$ 了，那这就有点扯了）

后验扩散条件概率可以写成：

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

使用贝叶斯公式（Bayes' rule），可以得到：

$$\begin{aligned} q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) &= q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)}{q(\mathbf{x}_t \mid \mathbf{x}_0)} \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t -}{1-}\right.\right. \\ &= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right.\right.\right. \end{aligned}$$

这里 $C(\boldsymbol{x}_t, \boldsymbol{x}_0)$ 是一个关于 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_0$ ，而不包含 $\boldsymbol{x}_{t-1}$ 的函数，这里省去了具体细节。根据高斯分布的概率密度，这里的方差和均值可以分别写成：

$$\tilde{\beta}_t = 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$$

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathbf{x}_0)/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0$$

由前面Forward过程我们推导得到的 $\mathbf{x}_0$ 和 $\mathbf{x}_t$ 的关系，我们有 $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t)$ ，代入上式后得到：

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t) \\ &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right) \end{aligned}$$

- **目标数据分布的似然函数**

这里与VAE很类似，我们可以使用variational lower bound来优化负对数似然函数。

$$
\begin{aligned}
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}\Big] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)\Big] \\
&= \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \\
\text{Let } L_{\text{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)
\end{aligned}
$$

> As demonstrated in Fig. 2., such a setup is very similar to VAE and thus we can use the variational lower bound to optimize the negative log-likelihood.

使用 Jensen 不等式[+]也很容易得到相同的结果。 假设我们想最小化交叉熵[+]作为学习目标，那么：

$$
\begin{aligned}
L_{\text{CE}} &= -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \Big(\int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \Big) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \Big(\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} \Big) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \Big(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \Big) \\
&\leq -\mathbb{E}_{q(\mathbf{x}_{0:T})} \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] = L_{\text{VLB}}
\end{aligned}
$$

为了将方程中的每一项都转换使其可分析计算，可以将目标进一步重写为几个 KL 散度和熵项的组合：

> To convert each term in the equation to be analytically computable, the objective can be further rewritten to be a combination of several KL-divergence and entropy terms[+] (See the detailed step-by-step process in Appendix B in Sohl-Dickstein et al., 2015):

$$
\begin{aligned}
L_{\text{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_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{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}
$$

我们分别标记变分下界损失中的每个分量：

$$
\begin{aligned}
L_{\text{VLB}} &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where } L_T &= D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T)) \\
L_t &= D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{ for } 1 \leq t \leq T-1 \\
L_0 &= -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)
\end{aligned}
$$

知乎 首发于 人工智能技术学习

al. 2020 使用一个从 $\mathcal{N}(\mathbf{x}_0; \mu_\theta(\mathbf{x}_1, 1), \sum_\theta(\mathbf{x}_1, 1))$ 推导出的独立的离散解码器⁺来模拟 L_0 .

> Every KL term in L_{VLB} (except for L_0) compares two Gaussian distributions and therefore they can be computed in closed form. L_T is constant and can be ignored during training because has q no learnable parameters and $\mathbf{x}_T$ is a Gaussian noise. Ho et al. 2020 models using a separate discrete decoder derived from $\mathcal{N}(\mathbf{x}_0; \mu_\theta(\mathbf{x}_1, 1), \sum_\theta(\mathbf{x}_1, 1))$.

三、DDPM的训练

1. 训练损失 L_t 的参数化⁺

回忆一下，我们之前说到需要学习一个神经网络来近似模拟在逆扩散过程中的条件概率分布⁺ $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sum_\theta(\mathbf{x}_t, t))$ 。我们希望 $\boldsymbol{\mu}_\theta$ 来预测 $\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t)$.因为我们已经有了 $\boldsymbol{x_t}$ 作为训练时的输入，我们可以将高斯噪声项进行参数化，从而在时刻 $\boldsymbol{t}$ 从 $\mathbf{x_t}$ 来预测 $\mathbf{z_t}$ 。

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right)$$
$$\text{Thus } \mathbf{x}_{t-1} = \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

损失项 $\boldsymbol{L_t}$ 被参数化，从而最小化它与 $\tilde{\boldsymbol{\mu}}_t$ 的差距。

> The loss term L_t is parameterized to minimize the difference from $\tilde{\boldsymbol{\mu}}_t$:

$$\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t,t)\right)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\mathbf{z}_t - \mathbf{z}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\mathbf{z}_t - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t,t)\|^2\right]
\end{aligned}$$

（*）简化：根据经验，Ho et al. (2020) 发现，在忽略加权项的简化目标函数⁺下，扩散模型可以训练得效果更好：

$$L_t^{\text{simple}} = \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\|\mathbf{z}_t - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t, t)\|^2\right]$$

因此最终简化后得目标函数为：

$$L_{\text{simple}} = L_t^{\text{simple}} + C$$

其中， $\boldsymbol{C}$ 是一个与 $\boldsymbol{\theta}$ 无关的常量。

2. 训练与推理过程伪代码⁺

Algorithm 1 Training

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

Algorithm 2 Sampling

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

- 《Deep Unsupervised Learning using Nonequilibrium Thermodynamics》
- 《Denoising Diffusion Probabilistic Models》
- lilianweng.github.io/po...
- bilibili.com/video/BV1b...

编辑于 2022-07-02 10:03

深度学习（Deep Learning） 计算机视觉 生成模型

理性发言，友善互动

46 条评论 默认 最新

评论内容由作者筛选后展示

伊文
好屌啊。。。
2022-09-04 · 山东 回复 8

rabbbbit
谁研究的捏😳
2022-10-12 · 河南 回复 1

拓海不爱学习 ▸ **rabbbbit**
扩散模型这玩意儿扩两头啊
2023-09-22 · 广东 回复 喜欢

凤年
同学你这张图片（pic1.zhimg.com/80/v2-4f...）应该是原作者当时写错了。明显x(t-1)的一次项系数第二项应该是2\alpha_{t-1} / (1-\alpha_{t-1})。但是这里错误写成了2\alpha_{t} / (1-\alpha_{t})。同学有空了可以check一下。
2022-07-01 · 中国香港 回复 6

卡卡猙特 作者
感谢指正！
2022-07-02 · 浙江 回复 喜欢

小葱
有点像苏联的很多教材，十分工整，但有点缺少引导串联。适合有数学基础和对DDPM有一番了解的人。
2023-01-09 · 山东 回复 5

愤怒的小鸟
你好，我想问一下，那个L_{VLB}最后一行的KL散度是怎么得到的
2022-10-17 · 北京 回复 4

徐瀚辰
看了您的讲解，很有帮助，不过我有一个问题想请教下，既然神经网络模型预测了出了噪声\epsilon_t，为什么不能通过图中的式子，直接由x_t推出x_0呢？（我知道这个问题很愚蠢😄

$$\left(\mathbf{x}_t - \sqrt{}\right.$$

2023-02-11 · 江西 回复 2

Gm83ZC3BdAkwLoj7
它是预测了噪声的概率分布，不是预测噪声。所以无法一步到位，就得一步一采样的往前推。你和我以前一样，没理解这一点。随时切记，预测的是噪声的概率分布，要一步一步采样才能到x0
2023-04-04 · 浙江 回复 9

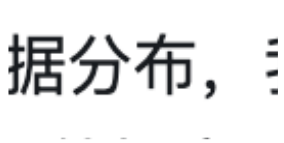
并不准确。通过神经网络求噪声，得到分布均值，再采样得到下一个时刻的x_{t-1}，这样的迭代过程可以有效减小误差。

2023-04-21 · 浙江 回复 1

查看全部 6 条回复 ›

倾城和诗

你好看了你的文章我收获挺大，但是还是有一些不懂。数据集里面求分布应该很难求吧，然后，应该是很难确定几步去去躁，去躁的过程中也很难求出每一步要去掉多少



05-17 · 北京 回复 喜欢

狗毛

谢谢博主，博主讲的很清楚🥹🥹

2023-12-17 · 美国 回复 喜欢

烟泯

我想问个问题，随机噪声中的I是什么？

2023-11-24 · 上海 回复 喜欢

目眺远方

老哥，高斯分布的KL散度公式错误，有个sigma少个下标1

2023-11-20 · 广东 回复 喜欢

却道

老哥，为啥我们在前向过程中已经知道x_{t-1}和x_{t}的关系了，采样时却不能用这个反推呢？但是在建立反向过程的期望值公式时，确实用到了这个关系（用x_t代替x_0）？

2023-10-01 · 中国香港 回复 喜欢

点击查看全部评论 ›

理性发言，友善互动

文章被以下专栏收录

人工智能技术学习

sci写作
sci写作常用句式

推荐阅读

扩散模型框架——随机微分方程（SDE）

阅读本文前最好对DDPM（扩散模型经典之作——DDPM）和SMLD（Score-based生成扩散模型（一）——SMLD）有一定了解。宋飚在《Score-Based Generative Modeling through Stochastic...

双重隐喻

数值分析浅谈——以离散最小二乘有限元为例

Numerical analysis is the study of algorithms for the problems of continuous mathematics. --- Lloyd N. Trefethen什么是数值分析（numerical analysis）？作为一个计算数学专业的高年...

Jackie Lee

如何理解概率密

大学的时候，我的统计》这门课一共
记得最后一次考过
格，只有60分。你
《概率论》这门课
了。后来，我工作

那可真蠢