扩散模型 DDPM.

一、《一个视频看懂DDPM原理推导》—— Nik-Li

·降噪过程每一步学的是一个分布（而非确切的 X_{t-1}）⇒ 更好随机性

噪音

但 uNet 的输出最后是个噪音 ⇐ 重参数化

零：理论基础.

1. 正态分布（高斯分布），$x \sim N(\mu, \sigma^2)$

(1) 扩散过程的终点 X_T 是一个标准正态分布。

(2) 多维高斯分布

$$\boxed{x \sim N(\mu, \Sigma)}$$

· x 是图片展平后的向量

· μ：均值向量

· Σ：协方差矩阵 ⇒ $\sigma^2 I$（每个像素噪声独立）

(3) DDPM 用到的"三大高斯性质".

① 重参数化（Reparameterization）

· 前向加噪核心：让随机采样可导.

· 从 $N(\mu, \sigma^2)$ 中采样一个 x：

1° 从 $N(0,1)$ 中采样一个随机噪声 ϵ.

2° 线性变换得 x：$x = \mu + \sigma\varepsilon$

· DDPM 中：

$$X_t \sim N(\sqrt{1-\beta_t}\, X_{t-1}, \beta_t I)$$

↓

$$X_t = \sqrt{1-\beta_t}\, X_{t-1} + \sqrt{\beta_t} \cdot \varepsilon$$

noise ⇒ 转换成预测噪声 ε.

② 高斯分布的可加性.

（VAE、DDPM）

在生成模型论文中的符号：

$\boxed{q}$：人为设定的、已知的真实分布

- 破坏者 · 完全已知 · $q(X_t|X_{t-1})$：在前一时刻的基础上加噪.
- Encoder

$\boxed{p}$：神经网络试图学习的、模拟的分布.

- 重建者 · 可学习的，p_θ（θ为权重）
- $p_\theta(X_{t-1}|X_t)$：神经网络猜上一时刻的分布.
- Decoder.

· 两个独立高斯分布的随机变量相加，结果依然高斯.

$X \sim N(\mu_1, \sigma_1^2)$
$Y \sim N(\mu_2, \sigma_2^2)$ $\Rightarrow X+Y \sim N(\mu_1+\mu_2, \sigma_1^2+\sigma_2^2)$

· 在DDPM中：

让扩散中的加噪一步到位 $\Rightarrow q(x_t|x_0)$ 的分布

③ 贝叶斯公式与高斯乘积

$$N(x;\mu_1,\sigma_1^2)\cdot N(x;\mu_2,\sigma_2^2) \propto N(x;\mu_{new},\sigma_{new}^2)$$

· 在DDPM中：$q(x_t|x_{t-1})\cdot q(x_{t-1}|x_0)$，具体见下节.

2、贝叶斯公式

$$\boxed{P(A|B)=\frac{P(B|A)\cdot P(A)}{P(B)}}$$

(1) 先验概率 (Prior, $P(A)$)

· 根据以往的经验 $\longrightarrow q(x_{t-1}|x_0)$

(2) 似然 (Likelihood, $P(B|A)$)

· 假设A已发生的情况下，出现B的概率 $\longrightarrow q(x_t|x_{t-1})$

(3) 后验 (Posterior, $P(A|B)$) (从果推因).

· 最想求的值. 看到证据B后，需要修正原来的判断，得到事件A发生的概率.

⇓

逆向去噪过程：已知：全是噪声的 x_t (证据B)

想求：这图上一时刻 x_{t-1} (A) 的概率

$q(x_{t-1}|x_t)$

(4) 贝叶斯在DDPM.

$$\boxed{q(x_{t-1}|x_t,x_0)=\frac{q(x_t|x_{t-1},x_0)\,q(x_{t-1}|x_0)}{q(x_t|x_0)}}$$

后验概率公式

· 假设训练已知原图 x_0：

$q(x_{t-1}|x_t) \rightarrow q(x_{t-1}|x_t,x_0)$

· 右侧三项全部已知且高斯

⇓

$q(x_{t-1}|x_t,x_0)$ 的均值与方差可以写出解析解

4. 联合分布 (Joint Distribution)

(1) 描述了多个随机变量同时取特定值的概率。

(2) 离散 (Discrete)

$$\boxed{P_{XY}(x,y) \triangleq P(X=x, Y=y)}$$

·性质：(1)归一性

$$\sum_{x\in X}\sum_{y\in Y} P(X=x, Y=y) = 1$$

- X, Y：随机变量（不确定的函数/状态）
- x, y：该变量取到的具体数值.

(3) 连续 (Continuous)

·使用联合概率密度函数 (PDF)：

$f_{XY}(x,y) \to p(x,y)$

·对连续变量，单点概率 $P(X=x, Y=y)=$
因此我们讨论的是落在一个小区域内的概率.

$$P(x\le X\le x+dx,\ y\le Y\le y+dy) \approx f_{XY}(x,y)\,dx\,dy$$

·归一性：

$$\boxed{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\,dx\,dy = 1}$$

5. 边缘分布 (Marginal Distribution)

边缘化

(1) 对联合分布中"无关变量"进行求和(离散)/求积分(连续).

(2) 离散

$$P_X(x) = \sum_{y\in Y} P_{XY}(x,y)$$

(3) 连续

$$f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x,y)\,dy$$

$\downarrow$ DDPM

$$\boxed{p(x) = \int p(x,y)\,dy.}$$

6. 重要法则

(1) 加法法则 (边缘化)

$\to$ 消变量

$$\boxed{p(x) = \int p(x,y)\,dy.}$$

(2) 乘法

$$\boxed{p(x,y) = p(y|x)\cdot p(x) = p(x|y)\cdot p(y)}$$

(3) 全概率公式 $\to$ 描述隐变量模型的生成过程.

$$\boxed{p(x) = \int \underbrace{p(x|z)}_{\text{Likelihood}}\cdot\underbrace{p(z)}_{\text{Prior}}\,dz}$$

7. 期望 (expectation view)

·把积分写成期望

$$\boxed{p(x) = E_{z\sim p(z)}\left[p(x|z)\right]}$$

·含义：边缘概率 $p(x)$ 可以看作似然函数 $p(x|z)$ 在先验分布 $p(z)$ 下的期望值.

·$p(z)$：z 服从的分布

·DDPM：图像 x 的分布，本质上是无数条可能噪声路径的生成概率的平均值.

边缘概率

权重：z 出现的先验概率

·物理意义：x发生的概率 = 在所有可能的隐状态 z 下生成 x 的概率的加权平均.

→将神经网络的任务→去学 $q(x_{t-1}|x_t, x_0)$ 的均值→预测噪声 ε.

3、最大化似然(Log likelihood)+变分推断(Variational Inference)

(1) 定义：真实数据 $X=\{x_1, x_2, \dots, x_n\}$，模型是一个概率分布 $p_\theta(x)$。似然函数就是模型生成这些真实数据的联合概率：

$$L(\theta)=p_\theta(x_1, x_2, \dots, x_n) \overset{\text{独立}}{=} \prod_{i=1}^{n} p_\theta(x_i)$$

目标是找一个 θ，让 $L(\theta)$ 最大。

(2) Why Log?

- 连乘→连加：防止 underflow
- 方便求导.

$$\Rightarrow \boxed{\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p_\theta(x_i)}$$

(3) 最小化 Loss ⟺ 最小化负的 Likelihood.

$$Loss = -\sum_{i=1}^{n} \log p_\theta(x_i)$$

(4) DDPM 中的最大似然.

1° 由于预测数据～高斯分布 → $p(x)=\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

$$\Rightarrow -\log p(x) = \underbrace{\frac{1}{2\sigma^2}(x-\mu)^2}_{MSE} + \underbrace{\log(\sqrt{2\pi\sigma^2})}_{\text{常数项}}$$

∴在高斯分布前提下：Max Log-Likelihood ⟺ MSE.

2° 由于解析的 $p_\theta(x_0)$ 无法计算 → $\int \overbrace{p_\theta(x_0, x_1, \dots, x_T)}^{\text{无数条}x_0\to x_T\text{的路径}} dx_{1:T} = p(x_0)$ (边缘概率)

∴要转去优化其下界(Evidence Lower Bound, ELBO)

∵ $ELBO \approx \underbrace{\sum D_{KL}(\text{后验}\,\|\,\text{模型})}_{\text{高斯分布下, } D_{KL} \propto \|\mu_1-\mu_2\|^2 \to \text{均值的均方误差}} \approx \sum MSE$

☆ $\boxed{\text{最小化分布差异} \Leftrightarrow \text{最小化 } \varepsilon \text{ 和 } \varepsilon_\theta \text{ 差异}}$ ⟸ 由噪声决定

4. KL 散度 (KL Divergence)

- 衡量2个概率分布之间的差异.
- $D_{KL}(A\|B)=0$：A和B完全一致.
- 值越大，表示差距越大.

推导:

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \varepsilon_{t-1}$$

$$= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\, x_{t-2} + \sqrt{1-\alpha_{t-1}}\, \varepsilon_{t-2}\right) + \sqrt{1-\alpha_t}\, \varepsilon_{t-1}$$

$$= \sqrt{\alpha_t \alpha_{t-1}}\, x_{t-2} + \underbrace{\overset{①}{\sqrt{\alpha_t(1-\alpha_{t-1})}\, \varepsilon_{t-2}} + \overset{②}{\sqrt{1-\alpha_t}\, \varepsilon_{t-1}}}_{\text{noise}}$$

$$\sqrt{①^2+②^2} = \sqrt{1-\alpha_t\alpha_{t-1}} \rightarrow \varepsilon$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\, x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\, \varepsilon$$

$$\Rightarrow \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t} \cdot \varepsilon$$

一、前向过程

1、单步加噪： $\boxed{x_t=\sqrt{1-\beta_t}\,x_{t-1}+\sqrt{\beta_t}\,\varepsilon_{t-1}}$ （重参数化）

· β_t：方差调度参数（Variance Schedule），很小接近于0.

· ε_{t-1}：$N(0,I)$采样的随机噪声.

2、任意步加噪（$x_0\to x_t$）

· 记：$\alpha_t=1-\beta_t$　　$\boxed{x_t=\sqrt{\bar\alpha_t}\,x_0+\sqrt{1-\bar\alpha_t}\,\varepsilon}$

$\bar\alpha_t=\prod_{i=1}^{t}\alpha_i$　　i.e. $q(x_t|x_0)\sim N(x_t;\sqrt{\bar\alpha_t}x_0,(1-\bar\alpha_t)I)$

· 物理含义：

∘ $\sqrt{\bar\alpha_t}\,x_0$：信号项、随着t↑，原图信息↓

∘ $\sqrt{1-\bar\alpha_t}\,\varepsilon$：噪声项。t↑，权重→1，图像变为纯噪声.

· 当x_T中T会很大时，$x_T\approx\varepsilon$意味着什么？

Ans. 扩散过程最终将任意数据分布收敛到标准正态分布，保证了我们可以通过采样标准正态分布的噪声来开始生成过程.

二、逆向过程

$= \frac{p(x_t|x_{t-1})\cdot p(x_{t-1})}{p(x_t)}$ 不好求.

· 思路：直接学$p(x_{t-1}|x_t)$很难，但可以学$q(x_{t-1}|x_t,x_0)$的均值

i.e. $q(x_{t-1}|x_t,x_0)\sim N(x_{t-1};\boxed{\tilde\mu_t},\tilde\beta_t I)$

$$\tilde\mu_t(x_t,x_0)=\frac{\sqrt{\alpha_t}(1-\bar\alpha_{t-1})}{1-\bar\alpha_t}x_t+\frac{\sqrt{\bar\alpha_{t-1}}\,\beta}{1-\bar\alpha_t}x_0$$ ←无证推理

⇓ 任意步加噪反解 $x_0=\frac{x_t-\sqrt{1-\bar\alpha_t}\,\varepsilon}{\sqrt{\bar\alpha_t}}$

$$\boxed{\tilde\mu_t=\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{\beta_t}{\sqrt{1-\bar\alpha_t}}\right)\varepsilon}$$

$\tilde\beta_t=\frac{1-\bar\alpha_{t-1}}{1-\bar\alpha_t}\beta_t$

没有未知数！

· x_t：输入给网络的当前图像（已知）

· $\alpha_t,\bar\alpha_t,\beta_t$：人为设定超参　（已知）

· ε：导致x_0变成x_t的那个噪声.（未知）

不需要预测 x_{t-1}，也不需要预测原图 x_0.

$\Rightarrow$ 训练网络 $\varepsilon_\theta(x_t, t) \rightarrow$ 预测噪声 → 该噪声对于第t个逆向步来说是分别预测出来的(每步单独预测)，但会同时从噪t的噪声。

$\varepsilon_\theta \rightarrow \tilde{\mu}_t \xrightarrow{\text{采样}} x_{t-1} \rightarrow \cdots x_t \rightarrow x_0$

$= \tilde{\mu}_t(x_0, x_t) + \sqrt{\tilde{\beta}_t}\,\underset{\sim N(0,1)}{\varepsilon}$

2、关于方差 $\tilde{\beta}_t$

- 神经网络在设定方差时直接固定.
- 物理含义：σ：这步去噪有多大随机性/模糊度.
  μ：图像长什么样.
- IDDPM (OpenAI) 发现：预测/学习方差，生成的图片效果不明显，但 log-likelihood (对数似然) 会显著提升，且允许用更少的步数生成图片.

二、常见问题.

1、~~一下取到多大的时候，x_T 满足 $N(0, I)$？ Ans. 取大些~~.

2、Why $\beta_t = 1-\alpha_t$？ i.e. $\mu^2+\sigma^2=1$

Ans、保证均值→0时，方差→1.

3、能否跳步？

Ans、$P(x_{t-1}|x_t, x_0)$ 是基于马尔可夫性质建立的. 所以要在逆向过程跳步的话，需要去马尔可夫化 → DDIM.
(本质是一种采样方式).

4、为什么不用 UNet 去做 $x_T \xrightarrow{\text{预测}} x_{T-1}$？

Ans. 直接去预测 x_{T-1} 类似退化为风格任务. 此时每一个 x_T 都对应一个 x_0. 即每一步都 deterministic. 取 2个一样的 x_T 会得到 2个一样的 x_0.

但去做 $P(x_{t-1}|x_t)$ 多样性更高. $x_T \Rightarrow x_0$, $x_T \Rightarrow x_0'$
(因为涉及采样)

四、Loss Function.

$$\boxed{L(\theta) = E_{t,x_0,\varepsilon}\left[\|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t)\|^2\right]}$$

$$Loss = \|\varepsilon - \varepsilon_\theta(x_t, t)\|^2$$

- 上述是简化后的Loss：将所有t的权重都设为1，但简化后生成的图片质量反而更好.
- 由变分下界推导，我们要优化的是两个高斯分布（后验q和模p）之间的KL散度.

$$L_{vlb} = E\left[\sum_{t=1}^{T}\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\right]$$

- Why 简化更好？

Ans. · 加权Loss会更多关注t很小的步骤（对应于不可感知 imperceptible 的细节），通常是图像几乎完全去噪的微小噪声.

· 不加权：相当于增加了t较大时的权重，迫使模型更好地学习图像的整体结构和内容.

五、DDPM中的UNet结构.

1. DDPM中特有改动.

- 同一个U-Net要处理 t=1~1000 所有情况 → 需知道现在第几步
- 方法：正弦位置编码（Sinusoidal Positional Embeddings）.
  - 输入：标量t
  - 编码：正/余弦 ⟶ 高维向量
  - 注入：送入U-Net的每个残差块中；

Scale & shift : $Feature_{new} = Feature_{old} \times (1 + Scale_t) + Shift_t$