

深入解析Flow Matching技术



笑书神侠 ✓
北京航空航天大学 工学硕士

关注他

收录于 · AIGC理论与应用 >

1457 人赞同了该文章 >

【随着Stable Diffusion 3的发布，Flow Matching技术逐渐进入公众视野，成为了学术界和工业界关注的焦点。本文将梳理并阐述Flow Matching的核心概念，详细解读其在生成模型中的应用，并探讨其背后的数学原理。】

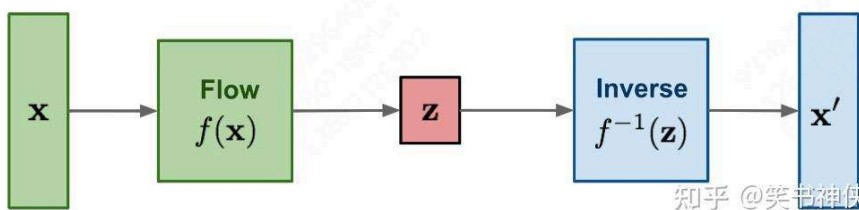


Fig.1 Illustration of a Flow-based model

Flow-based Model是一种基于**Normalizing Flows(NFs)**的生成模型，它通过一系列概率密度函数的变量变换，将复杂的概率分布转换为简单的概率分布，并通过逆变换生成新的数据样本。而**Continuous Normalizing Flows(CNFs)**是**Normalizing Flows**的扩展，它使用常微分方程(ODE)来表示连续的变换过程，用于建模概率分布。

Flow Matching(FM)是一种训练**Continuous Normalizing Flows**的方法，它通过学习与概率路径相关的向量场(**Vector Field**)来训练模型，并使用**ODE**求解器来生成新样本。

扩散模型是**Flow Matching**的一个应用特例，使用FM可以提高其训练的稳定性。此外,使用最优传输(**Optimal Transport⁺**)技术构建概率路径可以进一步加快训练速度，并提高模型的泛化能力。

一、Change of Variable for Probability Density Function

给定一个随机变量 z 及其概率密度函数 $z \sim \pi(z)$ ，通过一个一对一的映射函数 f 构造一个新的随机变量 $x = f(z)$ 。如果存在逆函数 f^{-1} ，那么新变量 x 的概率密度函数 $p(x)$ 计算如下：

(1) 当 z 为随机变量：

$$p(x) = \pi(z) \left| \frac{dz}{dx} \right| = \pi(f^{-1}(x)) \left| \frac{df^{-1}}{dx} \right| = \pi(f^{-1}(x)) |(f^{-1})'(x)|$$

(2) 当 z 为随机向量：

$$p(\mathbf{x}) = \pi(\mathbf{z}) \left| \det \frac{d\mathbf{z}}{d\mathbf{x}} \right| = \pi(f^{-1}(\mathbf{x})) \left| \det \frac{d\mathbf{f}^{-1}}{d\mathbf{x}} \right|$$

其中， \det 是行列式， $\frac{d\mathbf{f}}{d\mathbf{x}}$ 是雅可比矩阵。

特例：如果 $x \sim N(\mu, \sigma^2)$ ，当 a, b 为实数时，则有
 $z = f(x) = ax + b \sim N(a\mu + b, (a\sigma)^2)$

二、Normalizing Flows

Normalizing Flows(NFs)是一种可逆的概率密度变换方法，它的核心思想是通过一系列可逆的变换函数来逐步将一个简单分布（通常是高斯分布）转换成一个复杂的目标分布。**这个过程可以被看作是一连串的变量替换的迭代过程，每次替换都遵循概率密度函数的变量变换原则。**通过这种方式，**Normalizing Flows**能够精确地计算出变换后的分布的概率密度，从而实现从简单分布到复杂分布的精确映射。

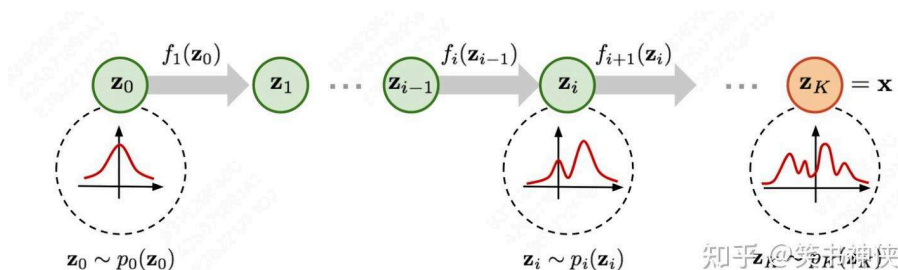


Fig.2 Illustration of a normalizing flow model

设 $p_0(z_0)$ 是原始的简单分布（例如标准高斯分布），Normalizing Flows 希望通过一系列可逆变换 $\{f_i\}$ 将其转换成目标分布 $p(x)$ 。这些变换定义了从 z_0 到 x 的映射，并且每一步变换 f_i 都有其逆变换 f_i^{-1} 。那么，变换过程可以表示为：

$$\mathbf{x} = \mathbf{z}_K = f_K \circ f_{K-1} \circ \dots \circ f_1(\mathbf{z}_0)$$

对于其中第 i 步，有：

$$\mathbf{z}_{i-1} \sim p_{i-1}(\mathbf{z}_{i-1})$$

$$\mathbf{z}_i = f_i(\mathbf{z}_{i-1}), \text{ thus } \mathbf{z}_{i-1} = f_i^{-1}(\mathbf{z}_i)$$

根据率密度函数的变量变换关系可得：

$$\begin{aligned}
 &= p_{i-1}(\mathbf{z}_{i-1}) \left| \det \left(\frac{d\mathbf{f}_i}{d\mathbf{z}_{i-1}} \right)^{-1} \right| \\
 &= p_{i-1}(\mathbf{z}_{i-1}) \left| \det \frac{d\mathbf{f}_i}{d\mathbf{z}_{i-1}} \right|^{-1}
 \end{aligned}$$

其对数似然为：

$$\log p_i(\mathbf{z}_i) = \log p_{i-1}(\mathbf{z}_{i-1}) - \log \left| \det \frac{d\mathbf{f}_i}{d\mathbf{z}_{i-1}} \right|$$

给定这样一连串的概率密度函数和变换关系，可以逐步展开直至追溯到初始分布，可得：

$$\begin{aligned}
 \log p(\mathbf{x}) &= \log \pi_K(\mathbf{z}_K) = \log \pi_{K-1}(\mathbf{z}_{K-1}) - \log \left| \det \frac{d\mathbf{f}_K}{d\mathbf{z}_{K-1}} \right| \\
 &= \log \pi_{K-2}(\mathbf{z}_{K-2}) - \log \left| \det \frac{d\mathbf{f}_{K-1}}{d\mathbf{z}_{K-2}} \right| - \log \left| \det \frac{d\mathbf{f}_K}{d\mathbf{z}_{K-1}} \right| \\
 &= \dots \\
 &= \log \pi_0(\mathbf{z}_0) - \sum_{i=1}^K \log \left| \det \frac{d\mathbf{f}_i}{d\mathbf{z}_{i-1}} \right|
 \end{aligned}$$

当这一系列变换函数 \mathbf{f}_i 可逆，且雅可比矩阵易于计算，模型训练时，优化目标为负对数似然：

$$\mathcal{L}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log p(\mathbf{x})$$

三、Continuous Normalizing Flows

Continuous Normalizing Flows (CNFs) 是 **Normalizing Flows** 的一种扩展，它可以更好地建模复杂的概率分布。在传统的 **Normalizing Flows** 中，变换通常是通过一系列**可逆的离散函数**来定义的，而在 **CNFs** 中，这种变换是**连续的**，这使得模型能够更加平滑地适应数据的分布，提高了模型的表达能力。**CNFs**过程通过常微分方程（ODE）来表示：

$$\frac{dz_t}{dt} = v(z_t, t)$$

其中， $t \in [0, 1]$ ， z_t 是 **Flow Map**，或者 **Transport Map**，可简单理解为时间 t 下的数据点， $v(z_t, t)$ 是一个**向量场**，它定义了每一个数据点在状态空间中随时间的变化方向和大小，**通常为神经网络预测**。

如果知道了这个向量场 $v(z_t, t)$ ，那么通过求解这个 **ODE** 就可以找到从初始概率分布到目标概率分布的连续路径，从而将简单分布转换成复杂分布。这个 **ODE** 可以采用**欧拉方法**来求解，从初始值 z_0 开始，使用下面的**迭代公式**来计算 z 在后续时间点的近似值：

$$z_{t+\Delta t} = z_t + \Delta t \cdot v(z_t, t)$$

其中， $\Delta t = 1/N$ 是步长， $t = i/N$ 是采样时间点， N 是最大采样步数， z_t 是在时间 t 的近似解。

这意味着，给定一个初始概率分布（通常是标准高斯分布），向量场 $v(z_t, t)$ 可以描述这个分布随时间的演变，最终达到目标分布。这是 **CNFs** 建模复杂概率分布的基础，即可以通过学习向量场来学习数据的变换过程。

四、Continuity Equation

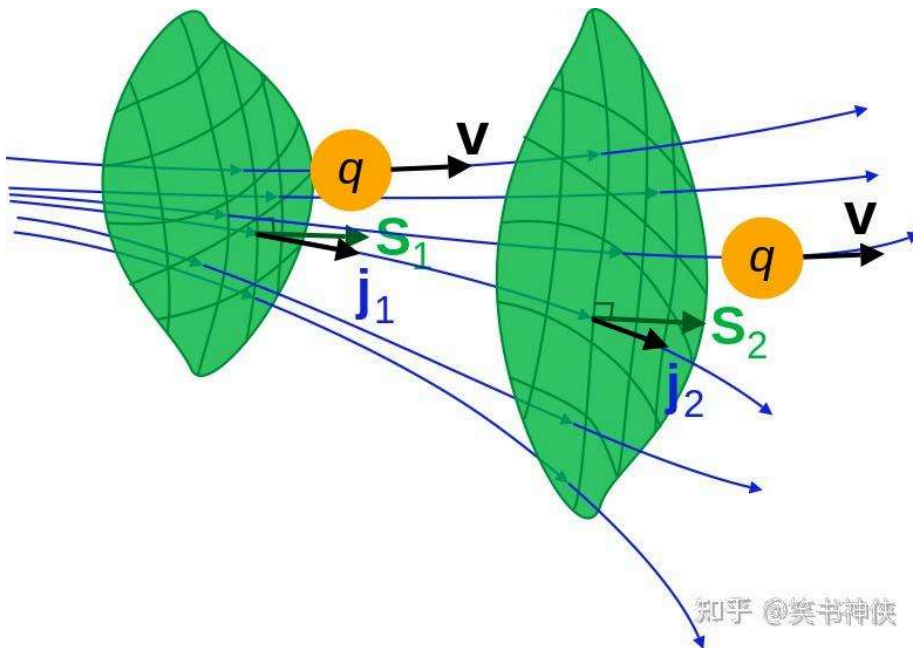


Fig.3 Illustration of vector field

连续性方程的一般形式是通过对流体的质量守恒定律进行数学推导得到的：

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{v}) = 0$$

- ρ 是流体的密度。
- \mathbf{v} 是流体的速度矢量。
- $\frac{\partial \rho}{\partial t}$ 是密度随时间的变化率。
- $\text{div}(\rho \mathbf{v})$ 是质量通量密度的散度，表示单位时间内通过单位面积的净质量流量。

这个方程表明，流体密度的时间变化率加上质量通量密度的散度等于零。换句话说，流体中任何封闭体积内的质量变化率等于流入和流出该体积的质量流量的差。

由于概率密度函数的性质确保了在全体分布上的积分为1，这反映了概率的总和是固定的，即概率是守恒的。在CNFs中，可以将这个性质与流体力学中的连续性方程类比，从而得到概率密度的连续性方程：

$$\frac{\partial p_t(\mathbf{x})}{\partial t} + \text{div}(p_t(\mathbf{x}) \mathbf{v}_t(\mathbf{x})) = 0 \quad \star$$

其中， $p_t(\mathbf{x})$ 是 t 时刻的概率密度函数， $\mathbf{v}_t(\mathbf{x})$ 是与 $p_t(\mathbf{x})$ 相关联的向量场，它描述了概率密度随位置和时间变化的变化， $\text{div}(p_t(\mathbf{x}) \mathbf{v}_t(\mathbf{x}))$ 是向量场与概率密度的乘积的散度，表示概率流通过某个区域的净变化率。

性质：Continuity Equation 是判断向量场 $\mathbf{v}_t(\mathbf{x})$ 产生对应的概率密度路径 $p_t(\mathbf{x})$ 的充分必要条件。如果向量场 $\mathbf{v}_t(\mathbf{x})$ 和概率密度路径 $p_t(\mathbf{x})$ 满足 Continuity Equation，则在CNFs中该向量场 $\mathbf{v}_t(\mathbf{x})$ 就能产生对应的概率密度路径 $p_t(\mathbf{x})$ 。

五、Conditional and Marginal Probability Paths and Vector Fields

对于目标分布为 $q(\mathbf{x}_1)$ 的每个数据样本 \mathbf{x}_1 ，可以定义一个随时间变化的条件概率路径 $p_t(\mathbf{x}|\mathbf{x}_1)$ 。对条件概率路径进行边缘化积分，就得到了一个边缘概率路径：

$$p_t(\mathbf{x}) = \int p_t(\mathbf{x} | \mathbf{x}_1) q(\mathbf{x}_1) d\mathbf{x}_1$$

这个边缘概率路径考虑了所有样本及其对应的条件概率路径。

$$u_t(x) = \int u_t(x | x_1) \frac{p_t(x | x_1)}{p_t(x)} dx_1 \quad \star$$

在数据点 x 处的向量场 $u_t(x)$ 是通过对所有可能的初始条件 x_1 的条件向量场 $u_t(x | x_1)$ 加权积分得到的，权重是由条件概率密度 $p_t(x | x_1)$ 和边缘概率密度 $p_t(x)$ 的比值决定的。

这个公式十分重要，是连接条件向量场和边缘向量场的桥梁。论文通过 **Theorem 1** 进行了总结，只要条件向量场 $u_t(x|x_1)$ 能生成对应的条件概率路径 $p_t(x|x_1)$ （满足连续性条件），那么上述边缘向量场 $u_t(x)$ 能够生成对应的边缘概率路径 $p_t(x)$ 。

Theorem 1: 给定向量场 $u_t(x|x_1)$ ，它生成条件概率路径 $p_t(x|x_1)$ ，对于任何分布 $q(x_1)$ ，上述定义的边缘向量场 $u_t(x)$ 就能生成对应的边缘概率路径 $p_t(x)$ 。

六、Flow Matching

训练 **Continuous Normalizing Flows** 的直观方法是，在给定初始条件 x_0 的情况下，通过 **ODE** 求解得到的 x_1 的分布，然后通过一种最小化差异度量（如 KL 散度）来约束 x_1 与真实数据的分布保持一致。然而，由于中间轨迹多而且未知，推断 x_1 （通过采样或者计算似然概率）需要反复模拟 **ODE**，计算量非常巨大。为此，论文提出了新的方法 **Flow Matching (FM)**。

Flow Matching 是一种适用于训练 **Continuous Normalizing Flows** 的技术，它是 **Simulation-Free** 的，即无需通过 ODE 推理目标数据分布。它的核心思想在于，通过确保模型预测的向量场与描述数据点实际运动的向量场之间的动态特性保持一致性，从而确保通过 CNFs 变换得到的最终概率分布与期望的目标分布相匹配。

具体来说，给定一个目标概率密度路径 $p_t(x)$ 及其对应的向量场 $u_t(x)$ ，这里的概率密度路径 $p_t(x)$ 是由这个向量场 $u_t(x)$ 生成的， $v_t(x)$ 是待学习的向量场，那么 **Flow Matching** 的优化目标可以定义为：

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t, p_t(x)} \|v_t(x) - u_t(x)\|^2 \quad \star$$

其中， $t \sim U[0, 1]$ ， $x \sim p_t(x)$ ，保证在整个时间范围和数据范围内对模型进行训练，确保模型能够处理从初始分布到目标分布的整个转换过程。**Flow Matching** 目标的核心是最小化这个损失函数，使得它预测的向量场 $v_t(x)$ 尽可能接近于实际的向量场 $u_t(x)$ ，从而能够准确地生成目标概率密度路径 $p_t(x)$ 。

Flow Matching 看起来很直观，但由于缺乏先验知识来确定合适的 $p_t(x)$ 和 $u_t(x)$ ，因此无法直接使用。因为尽管存在许多概率路径使得变换后的概率分布与目标分布相同或接近，但通常无法得到一个的封闭形式的向量场 $u_t(x)$ 来生成期望的概率路径 $p_t(x)$ 。

为了解决这个问题，论文提出 **Conditional Flow Matching**，可以采用一种基于样本的方法，为每个样本独立地定义概率路径和向量场，而不是依赖于一个全局的、封闭形式的向量场，然后通过适当的聚合方法，将这些个体样本的概率路径和向量场聚合起来以生成所需的 $p_t(x)$ 和 $u_t(x)$ 。

七、Conditional Flow Matching

由于 $u_t(x)$ 难以直接获得，也难以通过 $u_t(x|x_1)$ 进行边缘积分计算得到，因此直接优化原始 **Flow Matching** 目标函数不可行。

论文提出了 **Conditional Flow Matching** 方法，只要 $u_t(x|x_1)$ 和 $u_t(x)$ 满足上述加权边缘积分条件，那么 **Conditional Flow Matching** 优化目标与原始 **Flow Matching** 目标函数具有相同的最优解。**Conditional Flow Matching** 优化目标为：

关于作者



笑书神侠

知乎 深入解析Flow Matching技术

首发于
AIGC理论与应用

论文通过 Theorem 2 进行了证明 Conditional Flow Matching 优化目标与原始 Flow Matching 目标函数具有相同梯度，也即具有相同的最优解。

回答 21 文章 54 关注者 3,132

关注他

发私信

Theorem 2: 假设对于所有的 $x \in \mathbb{R}^d$ 和 $t \in [0, 1]$ ，都有 $p_t(x) > 0$ ，那么 \mathcal{L}_{CFM} 和 \mathcal{L}_{FM} 相差一个与 θ 无关的常数，即有 $\nabla_{\theta} \mathcal{L}_{FM}(\theta) = \nabla_{\theta} \mathcal{L}_{CFM}(\theta)$ 。

性质: 优化 Conditional Flow Matching 目标等同于优化 Flow Matching 目标，从而无需计算边缘概率路径或边缘向量场，只需要设计一个合适的条件概率路径和向量场即可。

八、Calculate Conditional Probability Paths and Conditional Vector Fields

Conditional Flow Matching 可以选择任意的条件概率路径，只要满足边界条件即可，这里针对一般高斯条件概率路径，分析如何构建 $p_t(x | x_1)$ 和 $u_t(x | x_1)$ 。

假设条件概率路径为高斯概率路径：

$$p_t(x | x_1) = \mathcal{N}(x | \mu_t(x_1), \sigma_t(x_1)^2 I)$$

只要满足如下两个边界条件，就能作为条件概率路径：

- 在时间开始时 ($t = 0$)，满足 $\mu_0(x_1) = 0, \sigma_0(x_1) = 1$ ，确保所有的条件概率路径都会收敛到相同的标准高斯噪声分布，即 $p(x) = \mathcal{N}(x | 0, I)$ 。
- 在时间进行到 ($t = 1$)，满足 $\mu_1(x_1) = x_1, \sigma_1(x_1) = \sigma_{min}$ ，这个 σ_{min} 值要设定得足够小，确保 $p_1(x | x_1)$ 是一个均值为 x_1 ，方差很小的高斯分布，这样条件概率路径会收敛到 x_1 附近。

这种设定方式定义了一个确定的变换过程，从 $t = 0$ 时的标准高斯分布开始，逐渐转变为 $t = 1$ 时的目标分布。

对于一个概率路径，存在无限多个向量场可以生成它，比如通过向 Continuity Equation 添加一个无散度分量，但这会导致不必要的计算量，这里可以使用最简单的向量场，它对应于高斯分布的标准变换，对应的数据点 Flow Map 为：

$$\psi_t(x) = \sigma_t(x_1)x + \mu_t(x_1)$$

可以根据概率密度变量变换可以直接得出，这个 Flow Map 满足上述高斯概率路径。

根据第三节中 CNFs 关于向量场的定义，求解以下 ODE，可以得到对应的条件向量场 $u_t(x | x_1)$ ：

$$\frac{d}{dt} \psi_t(x) = u_t(\psi_t(x) | x_1)$$

注意，该 ODE 中 u_t 是关于 $\psi_t(x)$ 的函数。

论文通过 Theorem 3 进行推导得到了条件向量场的封闭解。

Theorem 3: 设 $p_t(x | x_1)$ 为上述高斯概率路径， $\psi_t(x)$ 为其对应的 Flow Map，那么 $\psi_t(x)$ 具有的唯一向量场 $u_t(x | x_1)$ ，其形式为：

$$u_t(x | x_1) = \frac{\sigma'_t(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu'_t(x_1)$$

该向量场 $u_t(x | x_1)$ 可以生成对应的高斯概率路径 $p_t(x | x_1)$ 。

性质: 得到了 $u_t(x | x_1)$ 就可以根据 Conditional Flow Matching 的优化目标 \mathcal{L}_{CFM} 进行优化，从而得到待学习的全局向量场 $v_t(x)$ ，优化过程的关键就在于找到合适的条件向量

值得注意的是，上述这些不同随机噪声分布及路径在生成过程中，实际上可以根据需要设计任何合理的路径，SD3也是Conditional Flow Match的一种应用。

九、Diffusion conditional Vector Fields and Optimal Transport conditional Vector Fields

Flow Map公式设计时，允许使用各种可微分函数来定义 $\mu_t(x_1)$ 和 $\sigma_t(x_1)$ ，所以可以根据不同的应用场景和边界条件选择合适的函数。下面介绍两种特例，Diffusion conditional Vector Fields 和 Optimal Transport conditional Vector Fields。

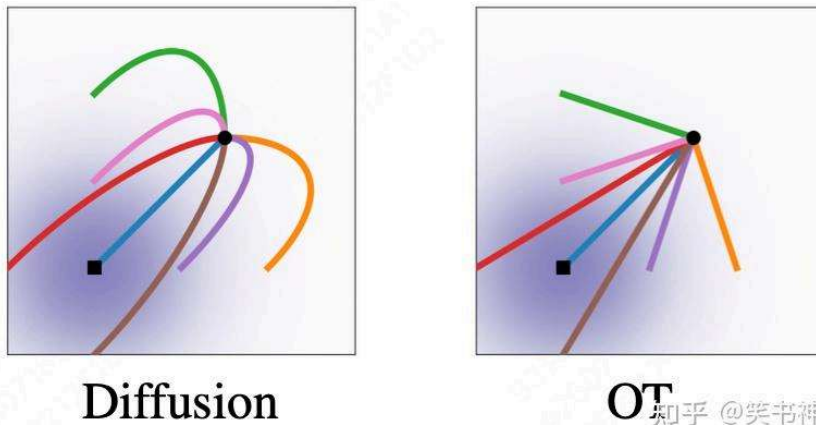


Fig.4 Diffusion and OT trajectories

9.1 Diffusion conditional Vector Fields

扩散模型中的 Variance Exploding (VE)和 Variance Preserving (VP)是两种不同类型的扩散过程，它们在生成模型中用于模拟两种不同的数据分布变化过程。

(1) **Variance Exploding (VE)**: VE 扩散模型是一种在生成过程中增加数据方差的扩散过程。在这种模型中，随着时间的推移，数据样本会逐渐变得更加嘈杂，即方差会不断增大，直到达到一个稳定的状态。VE 过程的一个特点是，它允许模型在生成数据时探索更广泛的潜在空间，这有助于生成多样化的样本。

VE的条件概率路径为：

$$p_t(x | x_1) = \mathcal{N}(x | x_1, \sigma_{1-t}^2 I)$$

其中， σ_t 是递增函数， $\sigma_0 = 0, \sigma_1 \gg 1$ ，对应均值和标准差为

$$\mu_t(x_1) = x_1, \sigma_t(x_1) = \sigma_{1-t}.$$

根据 Theorem 3 可以计算条件向量场为：

$$u_t(x | x_1) = -\frac{\sigma'_{1-t}}{\sigma_{1-t}}(x - x_1)$$

(2) **Variance Preserving (VP)**: VP 扩散模型是一种在生成过程中保持数据方差不变的扩散过程。在这种模型中，数据样本在生成过程中的方差保持恒定，这意味着模型在引入噪声的同时，也会以某种方式减少噪声，以保持数据的整体方差不变。VP 模型通常用于那些需要保持数据分布稳定性的应用场景，例如在图像生成中保持图像的清晰度和结构特征。

VP的条件概率路径为：

$$p_t(x | x_1) = \mathcal{N}(x | \alpha_{1-t}x_1, (1 - \alpha_{1-t}^2)I), \text{ where } \alpha_t = e^{-\frac{1}{2}T(t)}, T(t) = \int_0^t \beta(s)ds$$

根据 Theorem 3 可以计算条件向量场为：

$$u_t(x | x_1) = \frac{\alpha'_{1-t}}{1-\alpha_{1-t}^2} (\alpha_{1-t}x - x_1) = -\frac{T'(1-t)}{2} \left[\frac{e^{-T(1-t)}x - e^{-\frac{1}{2}T(1-t)}x_1}{1-e^{-T(1-t)}} \right]$$

结论：实验发现，将扩散模型条件向量场与 Flow Matching 目标结合起来优化，相比于现有的 Score Matching 方法，训练更加稳定。

9.2 Optimal Transport conditional Vector Fields

最优传输(Optimal Transport, 简称 OT)选择定义条件概率路径的均值和标准差为简单的时间线性函数，当时间 $t : 0 \rightarrow 1$ ，对应概率密度路径从 $p(x) = \mathcal{N}(x | 0, I)$ 到 $p_1(x | x_1)$ ，均值和标准差定义为：

$$\mu_t(x) = tx_1, \text{ and } \sigma_t(x) = 1 - (1 - \sigma_{\min})t$$

那么可得其对应 Flow Map 为：

$$\psi_t(x) = (1 - (1 - \sigma_{\min})t)x + tx_1$$

根据 Theorem 3 可计算条件向量场的封闭解为：

$$u_t(x | x_1) = \frac{x_1 - (1 - \sigma_{\min})x}{1 - (1 - \sigma_{\min})t}$$

结论：最优传输路径轨迹为直线，而扩散路径轨迹为曲线，因而可以得到更快的训练速度和生成速度，以及更好的性能表现。

十、References

1. [Flow Matching for Generative Modeling](#)
2. [Continuity equation](#)
3. [Flow-based Deep Generative Models](#)

【END】

送礼物

还没有人送礼物，鼓励一下作者吧

所属专栏 · 2024-11-20 16:55 更新



AIGC理论与应用

笑书神侠 北京航空航天大学 工学硕士

18 篇内容 · 3874 赞同

订阅

最热内容 · 通俗理解 Classifier Guidance 和 Classifier-Free Guidance 的扩散模型

编辑于 2025-02-10 14:35 · 浙江



理性发言，友善互动

56 条评论

默认 最新



知乎用户 nVsGX3

2024 年的扩散模型怎么成这样了😭一点都看不懂

2024-07-15 · 广东

回复 47



知乎用户 nVsGX3 · 小牛

让物理专业的人别学计算机了🤡🤡🤡顶不住了，什么薛定谔的桥都来了🤡

2024-07-28 · 广东

回复 61



小牛

我学习的过程经历了从ddpm到score based generative model，在后者里面先后经历了从SDE到ODE，最近刚到Flow Matching😂。真的是感叹于扩散模型发展之快。每到一阶段都发现好难，再到下一阶段又发现下一个阶段更难😂。

2024-07-26 · 上海

回复 15

查看全部 10 条回复 >



炼丹门下炼丹猫

物理学专业的不准转计算机!!!🤡

01-22 · 上海

回复 32



ycz

这个形式好优美

2024-03-28 · 日本

回复 5



COZeYu

Rectified Flow 等一系列的工作确实形式简洁

2024-03-28 · 广东

回复 6



CW 不要無聊的風格

我第一次看到 RF 的时候也惊了，感觉就是在用我初中数学在玩😂

2024-08-07 · 广东

回复 5



小白

符号逻辑混乱了。

2024-12-07 · 新加坡

回复 4



得意

最新文生图技术的基础理论😂

2024-03-27 · 北京

回复 4



欲壑难填

《五》里面，条件向量场到边缘向量场的加权积分怎么理解？为什么要有公式中红色部分的权重，这个权重是什么含义？为什么与上面条件概率路径到边缘概率路径的积分方式不同？

求教🙏

2024-06-20 · 北京

回复 1



却东

直接看这个形式可能不能理解，作者应该是用continuty eq解出来的这个形式

09-05 · 浙江

回复 1



祝君不记忧

[展开其他 1 条回复](#)**MountainChicken**

请教下flow-matching与reflow的区别是什么？

2024-10-21 · 新加坡

[回复](#) [❤️ 1](#)**zzzzhkzzzz**

retified flow 是flow-matching的一种特殊形式，其实从字面意思就能理解了，retified flow 直译就是整流，在flow-matching的基础作了路径直线化的约束。

07-15 · 北京

[回复](#) [❤️ 3](#)**Simin Xiu**

我理解是：flow-matching 是一种更加general的形式，而reflow 是其中的一种实例化

2024-11-07 · 北京

[回复](#) [❤️ 喜欢](#)[展开其他 1 条回复](#)**闲人**想问一下第八节为什么是 $u(x|x_1)$ 而不是 $u(x)$

2024-12-31 · 浙江

[回复](#) [❤️ 1](#)**轩尘熙**因为前面证明了原始的 $u(x)$ 和 $u(x|x_1)$ 优化上是相同的，然后求条件形式的会更好做，原始的 $u(x)$ 可能是求不出来（或者很难求？）🤔

02-23 · 广东

[回复](#) [❤️ 喜欢](#)**asdfgh**有大佬能解释下蓝框的这几步为什么能这么变换吗？为什么第二行直接变成 z_{i-1} 了，没想明白求指点

$$\begin{aligned} p_i(z_i) &= p_{i-1}(f_i^{-1}(z_i)) \left| \det \frac{df_i^{-1}}{dz_i} \right| \\ &= p_{i-1}(z_{i-1}) \left| \det \left(\frac{df_i}{dz_{i-1}} \right)^{-1} \right| \\ &= p_{i-1}(z_{i-1}) \left| \det \frac{df_i}{dz_{i-1}} \right|^{-1} \end{aligned}$$

2024-10-11 · 中国香港

[回复](#) [❤️ 1](#)**阿笠博士IIIIIIII**

我理解的是，互为反函数的方程组雅各比行列式互为倒数。

 $z_i = f_i(z_{i-1})$ $z_{i-1} = f_i^{-1}(z_i)$ 所以 $\det(df_i / dz_{i-1}) * \det(df_i^{-1} / dz_i) = 1$

2024-10-15 · 上海

[回复](#) [❤️ 5](#)**Echo** $z_i = f(z_{i-1})$ 你把这个带到图里的Inverse function theorem就好了

$$\frac{dx}{dy} = \left(\frac{dy}{dx} \right)^{-1} =$$

03-25 · 安徽

[回复](#) [❤️ 2](#)[点击查看全部评论](#)

理性发言，友善互动

推荐阅读

知乎 深入解析Flow Matching技术

首发于
AIGC理论与应用



Flow Matching 相关原理

Ferry

单的理解 flow matching。 1、几个概念概念粗糙着看看就行，不懂不要紧 flow：flow 就是一个流，将一个分布，通过一系列的可逆变换变成另一个分布，它建模了...

哆啦的梦 发表于深度乱学

么算法 Flow Matching 是一种用于机器学习生成模型训练的技术，主要出现在扩散模型（Diffusion Models）和流模型（Flow-based Models）的背景中。通俗易懂地...

天宇

Flow matching Background 在深入流匹配模型之前，我们需要先定义两个概念：流（flow）和速度场（vector field）。速度场 \mathbf{v}_t 定义为 $\mathbf{u} : [0,1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d, (t, \mathbf{u}) \mapsto \mathbf{v}_t(\mathbf{u})$ 。Escap... 发表于 Gener...