Lec 04 How to Train Flow and Diffusion Model

3.1 Flow Matching

首先，考虑以下常规流模型：

$$X_0 \sim p_{\text{init}}, \quad dX_t = u_t^\theta(X_t)dt$$

我们希望神经网络u_t^θ与边际向量场u_t^{target}尽可能相等，也就是说，我们需要找到参数θ，使得$u_t^\theta \approx u_t^{\text{target}}$。直觉上，我们可以通过均方误差MSE来衡量两者之间的差别，因此我们可以定义流匹配损失(Flow Matching Loss)：

$$\begin{aligned}\mathcal{L}_{FM}(\theta) &= \boldsymbol{E}_{t\sim\text{Unif},x\sim p_t}\left[\left\|u_t^\theta(x) - u_t^{\text{target}}(x)\right\|^2\right] \\ &= \boldsymbol{E}_{t\sim\text{Unif},z\sim p_{\text{data}},x\sim p_t(\cdot|z)}\left[\left\|u_t^\theta(x) - u_t^{\text{target}}(x)\right\|^2\right]\end{aligned}$$

- 其中$\text{Unif} = \text{Unif}_{[0,1]}$表示在$[0,1]$区间内的均匀分布。
- $\boldsymbol{E}$表示某随机变量的期望。

这个损失函数可以通过以下过程理解：

1. 选择一个随机时间步$t \in [0,1]$
2. 从数据集中选出一个随机的样本点z，从条件概率路径$p_t(\cdot|z)$中采样，并计算神经网络表示的向量场$u_t^\theta(x)$
3. 计算神经网络的输出结果和边际向量场$u_t^{\text{target}}(x)$之间的均方误差

但是我们无法通过积分直接计算边际向量场$u_t^{\text{target}}(x)$，即无法通过下式计算：

$$u_t^{\text{target}}(x) = \int u_t^{\text{target}}(x|z)\frac{p_t(x|z)p_{\text{data}}(z)}{p_t(x)}dz$$

因为这个积分是一个高维度的积分(一个数据集中可能有几千万张图片)，存在“维度的诅咒”，但是我们可以计算条件流匹配损失(Conditional Flow Matching Loss)，即：

$$\mathcal{L}_{CFM}(\theta) = \boldsymbol{E}_{t\sim\text{Unif},z\sim p_{\text{data}},x\sim p_t(\cdot|z)}\left[\left\|u_t^\theta(x) - u_t^{\text{target}}(x|z)\right\|^2\right]$$

该式子和原来的流匹配损失的公式的唯一区别就是把边际向量场$u_t^{\text{target}}(x)$换成了条件向量场$u_t^{\text{target}}(x|z)$，对于条件向量场我们是有解析解的，所以我们就可以把上述损失函数最小化。

流匹配与条件流匹配的关系

流匹配损失函数与条件流匹配损失函数之间的差是一个定值，与参数θ无关，即：

$$\mathcal{L}_{FM}(\theta) = \mathcal{L}_{CFM}(\theta) + C$$

因此我们可以很容易地发现他们的梯度是完全相同的，即：

$$\nabla_\theta\mathcal{L}_{FM}(\theta) = \nabla_\theta\mathcal{L}_{CFM}(\theta)$$

因此最小化$\mathcal{L}_{CFM}(\theta)$等价于最小化$\mathcal{L}_{FM}(\theta)$，也就是说，对于能够使得$\mathcal{L}_{CFM}(\theta)$最小的参数$\theta^*$，$u_t^{\theta^*} = u_t^{\text{target}}$恒成立。

证明：

$$
\begin{aligned}
\mathcal{L}_{FM}(\theta) &= \boldsymbol{E}_{t\sim \mathrm{Unif}, x\sim p_t}\left[\left\|u_t^\theta(x) - u_t^{\text{target}}(x)\right\|^2\right] \\
&= \boldsymbol{E}_{t\sim \mathrm{Unif}, x\sim p_t}\left[\left\|u_t^\theta(x)\right\|^2 - 2u_t^\theta(x)^T u_t^{\text{target}}(x) + \left\|u_t^{\text{target}}(x)\right\|^2\right] \\
&= \boldsymbol{E}_{t\sim \mathrm{Unif}, x\sim p_t}\left[\left\|u_t^\theta(x)\right\|^2\right] - 2\boldsymbol{E}_{t\sim \mathrm{Unif}, x\sim p_t}\left[u_t^\theta(x)^T u_t^{\text{target}}(x)\right] + \underbrace{\boldsymbol{E}_{t\sim \mathrm{Unif}, x\sim p_t}\left[\left\|u_t^{\text{target}}(x)\right\|^2\right]}_{=:C_1} \\
&= \boldsymbol{E}_{t\sim \mathrm{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)}\left[\left\|u_t^\theta(x)\right\|^2\right] - 2\boldsymbol{E}_{t\sim \mathrm{Unif}, x\sim p_t}\left[u_t^\theta(x)^T u_t^{\text{target}}(x)\right] + C_1
\end{aligned}
$$

对于第二项，我们可以继续展开：

$$
\begin{aligned}
\boldsymbol{E}_{t\sim \mathrm{Unif}, x\sim p_t}\left[u_t^\theta(x)^T u_t^{\text{target}}(x)\right] &= \int_0^1 \int p_t(x) \cdot u_t^\theta(x)^T u_t^{\text{target}}(x) dx dt \\
&= \int_0^1 \int p_t(x) \cdot u_t^\theta(x)^T \left[\int u_t^{\text{target}}(x|z) \frac{p_t(x|z) p_{\text{data}}(z)}{p_t(x)} dz\right]) dx dt \\
&= \int_0^1 \int \int u_t^\theta(x)^T u_t^{\text{target}}(x|z) p_t(x|z) p_{\text{data}}(z) dz dx dt \\
&= \boldsymbol{E}_{t\sim \mathrm{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)}\left[u_t^\theta(x)^T u_t^{\text{target}}(x|z)\right]
\end{aligned}
$$

所以我们可以把第二项代入回原式：

$$
\begin{aligned}
\mathcal{L}_{FM}(\theta) &= \boldsymbol{E}_{t\sim \mathrm{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)}\left[\left\|u_t^\theta(x)\right\|^2\right] - 2\boldsymbol{E}_{t\sim \mathrm{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)}\left[u_t^\theta(x)^T u_t^{\text{target}}(x|z)\right] + C_1 \\
&= \boldsymbol{E}_{t\sim \mathrm{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)}\left[\left\|u_t^\theta(x)\right\|^2 - 2u_t^\theta(x)^T u_t^{\text{target}}(x|z) + \left\|u_t^{\text{target}}(x|z)\right\|^2 - \left\|u_t^{\text{target}}(x|z)\right\|^2\right] + C_1 \\
&= \boldsymbol{E}_{t\sim \mathrm{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)}\left[\left\|u_t^\theta(x) - u_t^{\text{target}}(x|z)\right\|^2\right] - \underbrace{\boldsymbol{E}_{t\sim \mathrm{Unif}, z\sim p_{\text{data}}, x\sim p_t(\cdot|z)}\left[\left\|u_t^{\text{target}}(x|z)\right\|^2\right]}_{=:C_2} + C_1 \\
&= \mathcal{L}_{CFM}(\theta) + \underbrace{C_1 - C_2}_{=:C}
\end{aligned}
$$

训练过程

Algorithm 3 Flow Matching Training Procedure (General)

Require: A dataset of samples $z \sim p_{\text{data}}$, neural network u_t^θ

for each mini-batch of data **do**
    Sample a data example z from the dataset.
    Sample a random time $t \sim \mathrm{Unif}_{[0,1]}$.
    Sample $x \sim p_t(\cdot|z)$
    Compute loss

$$\mathcal{L}(\theta) = \|u_t^\theta(x) - u_t^{\text{target}}(x|z)\|^2$$

    Update the model parameters θ via gradient descent on $\mathcal{L}(\theta)$
end for

Score Matching

通过上节课的内容，我们知道边际分数函数的定义，以及如何将分数函数从ODEs扩展至SDEs，即：

$$\nabla \log p_t(x) = \int \nabla \log p_t(x|z) \frac{p_t(x|z) p_{data}(z)}{p_t(x)} dz \qquad \text{(边际分数函数)}$$

$$X_0 \sim p_{init},\ dX_t = \left[u_t^{target}(X_t) + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t) \right] dt + \sigma_t dW_t$$

$$\Rightarrow X_t \sim p_t \ (0 \leq t \leq 1)$$

为了逼近边际分数$\nabla \log p_t$，我们可以使用一个神经网络s_t^θ，名为分数网络(Score Network)，它接收X的“位置”与时间步，并将其映射到一个R^d空间内的向量。因此我们可以设计一个分数匹配损失函数(Score Matching Loss)以及条件分数匹配损失函数(Conditional Score Matching Loss)：

$$\mathcal{L}_{SM}(\theta) = \boldsymbol{E}_{t \sim \text{Unif}, z \sim p_{\text{data}}, x \sim p_t(\cdot|z)} \left[\left\| s_t^\theta(x) - \nabla \log p_t(x) \right\|^2 \right]$$

$$\mathcal{L}_{CSM}(\theta) = \boldsymbol{E}_{t \sim \text{Unif}, z \sim p_{\text{data}}, x \sim p_t(\cdot|z)} \left[\left\| s_t^\theta(x) - \nabla \log p_t(x|z) \right\|^2 \right]$$

它们还有另外一个名字，叫去噪分数匹配损失(Denoising Score Match Loss)

分数匹配与条件分数匹配的关系

分数匹配损失与条件分数匹配损失之间的差是一个定值，与参数θ无关，即：

$$\mathcal{L}_{SM}(\theta) = \mathcal{L}_{CSM}(\theta) + C$$

因此我们可以很容易地发现他们的梯度是完全相同的，即：

$$\nabla_\theta \mathcal{L}_{SM}(\theta) = \nabla_\theta \mathcal{L}_{CSM}(\theta)$$

其证明步骤与上文几乎完全相同，只是把$u_t^{\text{target}}(x)$改成$\nabla \log p_t$而已。

训练过程

Algorithm 6 Score Matching Training Procedure (General)

Require: A dataset of samples $z \sim p_{\text{data}}$, score network s_t^θ

for each mini-batch of data **do**
  Sample a data example z from the dataset.
  Sample a random time $t \sim \text{Unif}_{[0,1]}$.
  Sample $x \sim p_t(\cdot|z)$
  Compute loss

$$\mathcal{L}(\theta) = \| s_t^\theta(x) - \nabla \log p_t(x|z) \|^2$$

  Update the model parameters θ via gradient descent on $\mathcal{L}(\theta)$
end for

训练完之后，我们就可以通过下式生成样本：

$$X_0 \sim p_{init},\ dX_t = \left[u_t^\theta(X_t) + \frac{\sigma_t^2}{2} s_t^\theta(X_t)\right] dt + \sigma_t dW_t$$