LLMs Post-training: Dreams, Reality, and Fallacies

vs.

Supervised Fine-Tuning (SFT)

在SFT阶段，最直观的目标是，对于一个给定的问题（prompt, x），我们希望模型能够生成我们期望的答案（response, y）的概率尽可能大，y的概率，等于生成答案中每一个词 y_j 的概率的连乘积

$$\max\ p_\theta(y|x)=\prod_{j=1}^{m}p_\theta(y_j|x,y_{<j})$$

转化为损失函数：最小化负对数似然。我们通常不是最大化一个目标，而是最小化一个“损失函数”（Loss Function）。一个标准的操作就是对概率取“负对数”（negative log-likelihood）

$$KL(P||Q)=\sum P(x)\log\frac{P(x)}{Q(x)}=\sum P(x)\log P(x)-\sum P(x)\log Q(x)$$

$$\begin{aligned}L_{SFT}\quad(\theta)\ &=-E_{x\sim q(\cdot),y\sim p_d(\cdot|x)}\left[\log p_\theta(y|x)\right]\\&=-\sum_{x,y}p_d(y|x)\left[\log p_\theta(y|x)\right]+\sum_{x,y}p_d(y|x)\left[\log p_d(y|x)\right]\\&=-KL(p_d(y|x)||p_\theta(y|x))\end{aligned}$$

SFT训练的目标，本质上是**让模型的输出概率分布去尽可能地模仿和接近一个理想的、高质量的答案的概率分布**

Supervised Fine-Tuning (SFT)

在SFT中，我们理论上希望模型学习一个“最优”的答案分布，但我们无法直接从这个理想分布中获取数据。我们实际能获取数据的是从某个现有的、高质量的**“参考模型”（reference model）pr** 中采样。

$$
\begin{aligned}
L_{SFT}(\theta) &= -E_{x\sim q(\cdot),y\sim p_d(\cdot\,|x)}\left[\log p_\theta(y|x)\right] \\
&= -\sum_{x,y} p_d(y\,|x)\left[\log p_\theta(y|x)\right] \\
&= -\sum_{x,y} \frac{p_d(y\,|x)}{p_r(y\,|x)} p_r(y\,|x)\left[\log p_\theta(y|x)\right] \\
&= -E_{x\sim q(\cdot),y\sim p_r(\cdot\,|x)}\left[\frac{p_d(y\,|x)}{p_r(y\,|x)} \log p_\theta(y|x)\right]
\end{aligned}
$$

$$
w(y|x) = \frac{p_d(y|x)}{p_r(y|x)}
$$

$$
\boxed{L_{\mathrm{SFT}}(\theta) = -\,\mathbb{E}_{x\sim q(x),\, y\sim p_r(\cdot|x)}\Big[w(y|x)\ \log p_\theta(y|x)\Big].}
$$

所以我们用权重w矫正采样偏差，
才能做到**无偏估计**

目前大部分SFT工作都忽略了这个权重（相当于默认它为1），这可能会导致模型学习**出现偏差**

- **权重从哪来？** 真实的 p_d 不可见，所以 $w=\frac{p_d}{p_r}$ 需要近似：
  - 若你的数据是“用 p_r 生成 + 人工筛选/打分”的，p_d 就是“被接受”的条件分布；可以用**倾向得分/接受概率**或**奖励模型**给出 $w \propto \exp(R_\phi(x,y))$，再做自归一化。
  - 若 p_r 已很接近 p_d，可简单取 $w\approx 1$（退化回普通 SFT）。
  - 也可以训练一个二分类器区分 p_d 与 p_r，用**密度比估计**得到 w。

很多SFT优化的方法，例如SNIS，DFT，iw-sft都是在上面玩花活

Reinforcement Learning (RL)

policy gradient loss 函数： $L_{pq}(\theta) = -\mathbb{E}_{x\sim q(\cdot),\, y\sim p_\theta(\cdot|x)}\left[R \cdot \log p_\theta(y|x)\right]$

让我们推着玩一遍： $J(\theta) = \mathbb{E}_{(x,y)\sim p_\theta(\cdot)}[R(x,y)] = \sum_x p(x) \sum_y p_\theta(y|x) R(x,y)$ $\theta^* = \underset{\theta}{\mathrm{argmax}} J(\theta)$

$$\nabla_\theta J(\theta) = \nabla_\theta \left[\sum_x p(x) \sum_y p_\theta(y|x) R(x,y)\right] = \sum_x p(x) \sum_y \nabla_\theta p_\theta(y|x) R(x,y)$$

$$\nabla_\theta p_\theta(y|x) = p_\theta(y|x) \frac{\nabla_\theta p_\theta(y|x)}{p_\theta(y|x)} = p_\theta(y|x) \nabla_\theta \log p_\theta(y|x)$$

$$\nabla_\theta J(\theta) = \sum_x p(x) \sum_y p_\theta(y|x) \nabla_\theta \log p_\theta(y|x) R(x,y) \quad = \mathbb{E}_{x\sim p(\cdot),\, y\sim p_\theta(\cdot|x)}\left[\nabla_\theta \log p_\theta(y|x) \cdot R(x,y)\right]$$

如果我们也把reference LLM做成online的setup，\theta = r

$$L_{pq}(\theta) = -\mathbb{E}_{x\sim q(\cdot),\, y\sim p_\theta(\cdot|x)}\left[R \cdot \log p_\theta(y|x)\right]$$

$$L_{SFT}(\theta) = -E_{x\sim q(\cdot), y\sim p_r(\cdot|x)}\left[\frac{p_d(y|x)}{p_r(y|x)} \log p_\theta(y|x)\right]$$

$$R_{pg} \propto \frac{p_d(y|x)}{p_\theta(y|x)}$$

Reinforcement Learning (RL)

PPO的梯度：

$$\nabla L_{ppo}(\theta) = -E_{x\sim q(\cdot), y\sim p_r(\cdot|x)}\left[\frac{p_\theta(y|x)}{p_r(y|x)} R \nabla \log p_\theta(y|x)\right] + \nabla KL(p_r(y|x)||p_\theta(y|x))$$

$$\nabla J(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)\right]$$

$$\nabla J(\theta) = \mathbb{E}_{a\sim\pi_{\text{old}}(\cdot|s)}\left[\frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)} \nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)\right]$$

由于 $\nabla_\theta \log \pi_\theta(a|s) = \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}$，上式可重写为：

$$\nabla J(\theta) = \mathbb{E}_{a\sim\pi_{\text{old}}(\cdot|s)}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_{\text{old}}(a|s)} A^\pi(s,a)\right]$$

$$\nabla J(\theta) = \mathbb{E}_{x\sim q(\cdot), y\sim p_r(\cdot|x)}\left[\frac{p_\theta(y|x)}{p_r(y|x)} \nabla_\theta \log p_\theta(y|x) R\right]$$

$$\text{KL}(p_r||p_\theta) = \mathbb{E}_{y\sim p_r(\cdot|x)}\left[\log p_r(y|x) - \log p_\theta(y|x)\right]$$

Reinforcement Learning (RL)

PPO的梯度:

$$\nabla\ L_{ppo}(\theta) = -E_{x\sim q(\cdot),y\sim\ p_r(\cdot\ |x)}\ [\frac{p_\theta(y\ |x)}{p_r(y\ |x)} R\ \nabla \log p_\theta(y|x)]\ +\ \nabla\ KL(p_r(y|x)||p_\theta(y|x))$$

$$\nabla_\theta \mathrm{KL}(p_r \| p_\theta) = \nabla_\theta \mathbb{E}_{y\sim p_r(\cdot|x)} \left[\log p_r(y|x) - \log p_\theta(y|x)\right] = -\mathbb{E}_{y\sim p_r(\cdot|x)} \left[\nabla_\theta \log p_\theta(y|x)\right]$$

$$J(\theta) = \mathbb{E}_{x\sim q(\cdot),y\sim p_r(\cdot|x)} \left[\frac{p_\theta(y|x)}{p_r(y|x)} R\right] - \beta \mathrm{KL}(p_r \| p_\theta)$$

$$J(\theta) = \mathbb{E}\left[\frac{p_\theta(y|x)}{p_r(y|x)} R\right] - \mathrm{KL}(p_r \| p_\theta)$$

$$\nabla J(\theta) = \nabla \mathbb{E}\left[\frac{p_\theta(y|x)}{p_r(y|x)} R\right] - \nabla \mathrm{KL}(p_r \| p_\theta)$$

$$\nabla \mathbb{E}\left[\frac{p_\theta(y|x)}{p_r(y|x)} R\right] = \mathbb{E}\left[\nabla\left(\frac{p_\theta(y|x)}{p_r(y|x)} R\right)\right] = \mathbb{E}\left[R \frac{\nabla_\theta p_\theta(y|x)}{p_r(y|x)}\right] = \mathbb{E}\left[R \frac{p_\theta(y|x)}{p_r(y|x)} \nabla_\theta \log p_\theta(y|x)\right]$$

Reinforcement Learning (RL)

PPO的梯度:

$$\nabla L_{ppo}(\theta) = -E_{x\sim q(\cdot),y\sim p_r(\cdot|x)}[\frac{p_\theta(y|x)}{p_r(y|x)} R \nabla \log p_\theta(y|x)] + \nabla KL(p_r(y|x)||p_\theta(y|x))$$

$$-\nabla \mathrm{KL}(p_r \| p_\theta) = -(-\mathbb{E}[\nabla_\theta \log p_\theta(y|x)]) = \mathbb{E}[\nabla_\theta \log p_\theta(y|x)]$$

$$\nabla J(\theta) = \mathbb{E}\left[R\frac{p_\theta(y|x)}{p_r(y|x)} \nabla_\theta \log p_\theta(y|x)\right] + \mathbb{E}[\nabla_\theta \log p_\theta(y|x)]$$

$$\nabla L_{ppo}(\theta) = -\nabla J(\theta) = -\mathbb{E}\left[R\frac{p_\theta(y|x)}{p_r(y|x)} \nabla_\theta \log p_\theta(y|x)\right] - \mathbb{E}[\nabla_\theta \log p_\theta(y|x)]$$

Reinforcement Learning (RL)

PPO的梯度：

$$\nabla L_{ppo}(\theta) = -E_{x\sim q(\cdot), y\sim p_r(\cdot\,|x)}\left[\frac{p_\theta(y\,|x)}{p_r(y\,|x)} R\,\nabla \log p_\theta(y|x)\right] + \nabla KL(p_r(y|x)||p_\theta(y|x))$$

具体推导见上学期我组会分享的，我就不推了，优势函数 A 可能被奖励 R 替代（但严格来说，应使用优势函数以减少方差）

$$\frac{p_\theta(y\,|x)}{p_r(y\,|x)} R_{pg} \propto \frac{p_d(y\,|x)}{p_r(y\,|x)}$$

$$\begin{aligned} R_{ppo} &\propto \frac{p_d(y\,|x)}{p_r(y\,|x)} \frac{p_r(y\,|x)}{p_\theta(y\,|x)} \\ &= \frac{p_d(y\,|x)}{p_\theta(y\,|x)} \end{aligned}$$

我们可以发现，对于policy gradient和PPO，我们的最优reward function 其实**是一致的**

Direct Preference Optimization (DPO)

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}[r(x,y)] - \beta\mathbb{D}_{KL}[\pi_\theta(y|x)||p_{ref}(y|x)]$$

对于一个固定的奖励函数 $r(x,y)$，上述优化问题存在一个**解析最优解**

$$\pi^*(y|x) = \frac{1}{Z(x)} p_{ref}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

$$\log\pi^*(y|x) = \log p_{ref}(y|x) + \frac{1}{\beta}r(x,y) - \log Z(x)$$

$$\frac{1}{\beta}r(x,y) = \log\pi^*(y|x) - \log p_{ref}(y|x) + \log Z(x)$$

$$r(x,y) = \beta\log\frac{\pi^*(y|x)}{p_{ref}(y|x)} + \beta\log Z(x)$$

$$r(x,y_w) - r(x,y_l) = \beta\left(\log\frac{\pi^*(y_w|x)}{p_{ref}(y_w|x)} - \log\frac{\pi^*(y_l|x)}{p_{ref}(y_l|x)}\right)$$

告诉我们，最优策略 π^* 正比于参考策略 p_{ref} 乘以奖励的指数缩放。

DPO的核心洞察在于：我们可以**反过来用最优策略 π^* 和参考策略 p_{ref} 来表示奖励函数 $r(x,y)$**。

Direct Preference Optimization (DPO)

Bradley-Terry 模型是一个经典的**概率模型**，用于处理和预测**一对物品（或实体）之间比较的结果**。它的核心思想是：为每个物品赋予一个潜在的、无法直接观测的“实力”或“偏好度”分数，然后一对物品之间比较的胜负概率，可以由它们各自的实力分数决定。

1. 核心公式：

对于两个物品 i 和 j，模型规定 i 战胜（或优于）j 的概率为：

$$P(i \succ j) = \frac{\exp(\sigma_i)}{\exp(\sigma_i) + \exp(\sigma_j)} = \frac{1}{1 + \exp(-(\sigma_i - \sigma_j))} = \sigma(\sigma_i - \sigma_j)$$

请牢记这个表达式，后面我用另一个角度推导出来

$$P(y_w \succ y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))} = \sigma(r(x, y_w) - r(x, y_l))$$

$$\mathcal{L}(\theta) = \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log P(y_w \succ y_l | x)\right]$$

$$= \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \left(\log \frac{p_\theta(y_w|x)}{p_{ref}(y_w|x)} - \log \frac{p_\theta(y_l|x)}{p_{ref}(y_l|x)}\right)\right)\right]$$

如何推导的解析最优解？

$$\pi^*(y|x) = \frac{1}{Z(x)} p_{ref}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

$$\sum_y \pi_\theta(y|x) = 1 \quad \text{且} \quad \pi_\theta(y|x) \geq 0 \quad \text{对于所有y}$$

$$\mathcal{L}(\pi_\theta, \lambda) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x,y)] - \beta \mathbb{D}_{KL}[\pi_\theta(y|x)||p_{ref}(y|x)] + \lambda(x)(1 - \sum_y \pi_\theta(y|x))$$

$$\frac{\delta \mathcal{L}}{\delta \pi_\theta(y|x)} = 0$$

$$r(x,y) - \beta\left[\log \frac{\pi_\theta(y|x)}{p_{ref}(y|x)} + 1\right] - \lambda(x) = 0$$

$$\beta \log \frac{\pi_\theta(y|x)}{p_{ref}(y|x)} = r(x,y) - \beta - \lambda(x)$$

$$\log \frac{\pi_\theta(y|x)}{p_{ref}(y|x)} = \frac{r(x,y)}{\beta} - 1 - \frac{\lambda(x)}{\beta}$$

$$\frac{\pi_\theta(y|x)}{p_{ref}(y|x)} = \exp\left(\frac{r(x,y)}{\beta} - 1 - \frac{\lambda(x)}{\beta}\right)$$

$$\pi_\theta(y|x) = p_{ref}(y|x) \cdot \exp\left(\frac{r(x,y)}{\beta}\right) \cdot \exp\left(-1 - \frac{\lambda(x)}{\beta}\right)$$

$$\pi^*(y|x) = \frac{1}{Z(x)} p_{ref}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

$$\sum_y \pi^*(y|x) = 1 \quad \Rightarrow \quad \sum_y \frac{1}{Z(x)} p_{ref}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right) = 1$$

$$\Rightarrow \quad \frac{1}{Z(x)} \sum_y p_{ref}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right) = 1$$

$$\Rightarrow \quad Z(x) = \sum_y p_{ref}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

Direct Preference Optimization (DPO)

当我们用参考模型 p_r 产生 pairwise 样本 (y_1, y_2)时，如果这些样本对是按“近似最优的 reward”挑出来的，那么对每个 pair 都有一个**固定为正的间隔**

$$R \propto \frac{p_d(y \mid x)}{p_r(y \mid x)}$$

设定：同一输入 x 下有一对响应 (y_1, y_2)，其中 y_1 质量高于 y_2。若样本对来自参考策略 p_r，并且是按近似最优的 reward 选的（文中由前面的分析给出最优 reward 与 $\frac{p_r}{p_d}$ 成正比），则有

$$\frac{p_d(y_1 \mid x)}{p_r(y_1 \mid x)} > \frac{p_d(y_2 \mid x)}{p_r(y_2 \mid x)}$$

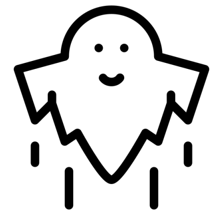
$$\begin{aligned} 0 < M &= \log \frac{p_d(y_1 \mid x)}{p_r(y_1 \mid x)} - \log \frac{p_d(y_2 \mid x)}{p_r(y_2 \mid x)} \\ &= [\log \frac{p_d(y_1 \mid x)}{p_\theta(y_1 \mid x)} - \log \frac{p_d(y_2 \mid x)}{p_\theta(y_2 \mid x)}] + [\log \frac{p_\theta(y_1 \mid x)}{p_r(y_1 \mid x)} - \log \frac{p_\theta(y_2 \mid x)}{p_r(y_2 \mid x)}] \\ &= M_1 + M_2 \end{aligned}$$

由于对已选定的样本对来说 M 是个**固定正数**（由前面的“按最优 reward 选样本”保证），因此

$$\text{maximize } M_2 \iff \text{minimize } M_1.$$

而 M_1 的下界是 0（当且仅当 $p_\theta(y_i|x) = p_d(y_i|x)$ 时取到），所以在理想条件下，最大化 M_2 会把 p_θ 推到 p_d。这解释了为什么只看得到 M_2 也能朝着正确方向优化。

Direct Preference Optimization (DPO)的隐患



$$L_{DPO}(\theta) = -E_{x\sim q(\cdot),y_1\sim p_r(\cdot|x),y_2\sim p_r(\cdot|x,y_1)}[\log\sigma(\log\frac{p_\theta(y_1|x)}{p_r(y_1|x)} - \log\frac{p_\theta(y_2|x)}{p_r(y_2|x)})]$$

现实里 DPO 的隐患：它没强制 $M_1 \geq 0$, $M_2 \geq 0$：解释了为什么你的DPO会训崩？

于是会出现：当 y_1, y_2质量非常接近（hard pair）时，很多 pair 其实不满足 $M_2 \geq 0$。在这种情况下，训练会“放弃”这些难以区分的 pair，转而主要优化那些差距本来就大的 easy pairs；结果是对 easy pairs 的分离被过度强化，出现 **过拟合**——例如把 $p_\theta(y_1 | x)$拉得过大，或把 $p_\theta(y_2 | x)$压得过小，使得对这些 pair 的 M_1被推到**远小于 0**（偏离真实目标分布）。这些现象对 SFT 不是问题，但对 DPO 属于“条件优化（需要满足间隔条件）”的困难。

DPO-high quality in other source?

SPIN(self-play)

这种做法类似：固定 positive + 动态 negative

你在一个分布上算梯度，却指望在另一个分布上收敛?

1. 假设过强：默认参考模型生成的质量一定比 source 差；现实不必然成立，也**限制了上限**（迭代后参考模型变强，收益变小）。
2. 分布偏差：source 的分布与参考模型不同，采样有偏。
3. 缺乏探索：正样 y_s固定，既不探索新的 y_1也不探索新的 x，样本上限受限。
因此，SPIN 可看成“省略了 reward model 或样本比较”的 **DPO 的近似版**，但有上述代价

我们想要最小化的理想目标其实是在参考分布 p_r 或目标分布 p_d 下的期望损失：

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim p_r}[\ell_\theta(x,y)].$$

但是 **SPIN 正样**是从 source 数据 p_s 采的，于是训练中我们优化的实际上是：

$$\hat{\mathcal{L}}(\theta) = \mathbb{E}_{(x,y)\sim p_s}[\ell_\theta(x,y)].$$

如果直接用 $\hat{\mathcal{L}}$ 更新参数，就等于用 p_s **下的梯度** 去优化 θ。而最终要泛化到的是 p_r（甚至 p_d）下的性能。

除非 $p_s = p_r$，否则 $\nabla_\theta \hat{\mathcal{L}}$ 和 $\nabla_\theta \mathcal{L}$ 不一样，更新方向偏了。这就是「你在一个分布上算梯度，却指望在另一个分布上收敛」的含义。