大模型零基础入门

从预训练到 GRPO 强化学习

非凡爱捯饬

河南大学

February 19, 2025

目录

1. 预训练——填词游戏
2. SFT 监督微调
3. RLHF 人类反馈强化学习
4. GRPO 优化策略
5. DeepSeek R1 解读

预训练的基本原理

预训练的本质

通过大量文本数据进行自监督学习，类似人类的阅读学习过程

- 基本原理：遮住句子后半部分，预测下一个词
- 示例：
  - 输入：" 今天天气真..."
  - 预测：" 好/差/热/冷"

主要挑战

统计概率 vs 期望回答:

- Q: " 你是谁？"
- A(统计概率): " 我是小明..."
- A(期望): " 我是一个 AI 助手..."

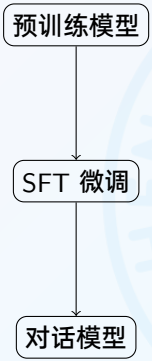
SFT 监督微调——形成对话格式

SFT 训练数据示例

- Q: " 你是谁？ "
- A: " 我是一个 AI 助手..."
- Q: "2+2 等于几？ "
- A: "2+2 等于 4"

SFT 的局限性

- 需要大量人工标注数据
- 容易过拟合到特定任务
- 无法处理复杂偏好（如多个合理答案）



RLHF 人类反馈强化学习

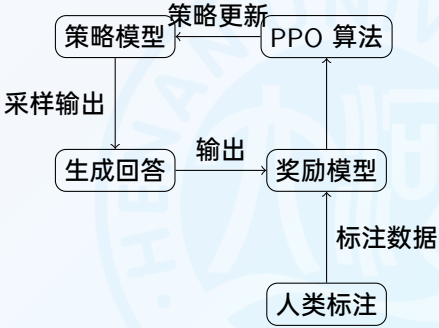
示例问题及人类偏好

问题： " 珠穆朗玛峰多高？"

- A 选项：" 本尼维斯山..." ——得分低
- B 选项：" 我不太清楚" ——得分中等
- C 选项："8848 米" ——得分高

RLHF 核心组件

- 奖励模型（Reward Model）
- PPO 强化学习算法
- 人类偏好数据
- 策略模型迭代更新



三阶段对比：预训练 / SFT / RLHF

预训练阶段

- 无监督学习
- 海量文本数据
- 关注语言建模

SFT 阶段

- 监督式微调
- 需要标注数据
- 关注任务完成

RLHF 阶段

- 强化学习框架
- 偏好数据驱动
- 关注人类偏好

奖励模型提供反馈

RLHF 的局限性

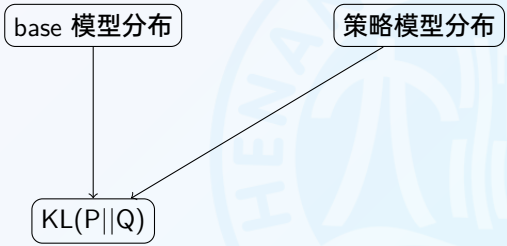
主要挑战

- 双网络架构负担：需维护策略网络和价值网络
- 内存消耗大：价值网络与策略网络规模相当
- 训练不稳定：优势函数估计方差较大
- 奖励黑客：奖励模型并不能代表真实的人类反馈

KL 散度简介

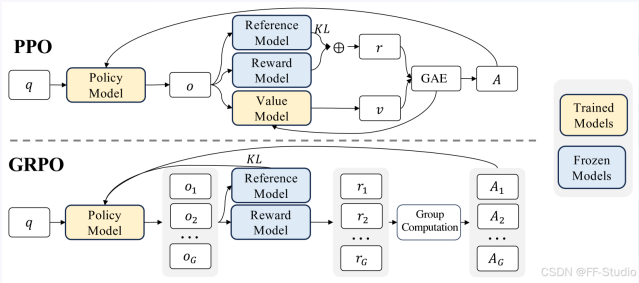
什么是 KL 散度

- 衡量大模型训练过程中的分布差异
- 通俗理解：测量正在训练的模型与原始模型的" 偏离程度"
- 控制 KL 散度可以避免灾难性遗忘



分布偏移度量

GRPO 的优化思路



主要步骤

1. 分组采样
2. 奖励归一化
3. 策略更新

GRPO 三大优势

- 单网络架构：仅需策略网络
- 分组相对奖励：同一问题下进行多候选对比
- 内存效率提升：节省 40% 显存

GRPO Trainer 参数设置

Python 代码示例

```
from trl import GRPOConfig, GRPOTrainer

# 训练参数配置
training_args = GRPOConfig(
    output_dir="my_model",
    num_generations=4,
    max_completion_length=256,
    temperature=0.9,
    per_device_train_batch_size=4,
    logging_steps=10
)
```

核心参数解析

- **生成样本数**: 每个问题生成 4-8 个回答对比（num_generations）
- **最大补全长度**: 控制生成文本长度（max_completion_length）
- **温度参数**: 0.9 平衡多样性与质量（temperature）
- **KL 系数**: 0.04 防止模型跑偏
- **批次大小**: 根据显存调整（per_device_train_batch_size）

推理模板

system：用户与助手的对话。用户提出问题，助手解决问题。
助手首先在脑海中思考推理过程，然后向用户提供答案。
推理过程和答案分别用 <reasoning></reasoning>和<answer></answer>标签包裹，
即：<reasoning> 此处为推理过程 </reasoning> <answer> 此处为答案 </answer>。

user：（提示词，用户的问题）

assistant：（为空，等待模型预测）

模板设计原则

为引导基础模型遵循指定指令，设计了简洁的推理模板，仅约束结构格式，避免内容偏见。

- **模板结构**: 用户与助手的对话形式
- **推理过程**: 使用 <reasoning></reasoning> 标签包裹
- **最终答案**: 使用 <answer></answer> 标签包裹
- **设计特点**: 避免强制特定推理策略，观察模型自然进化

奖励函数

实现方式概述

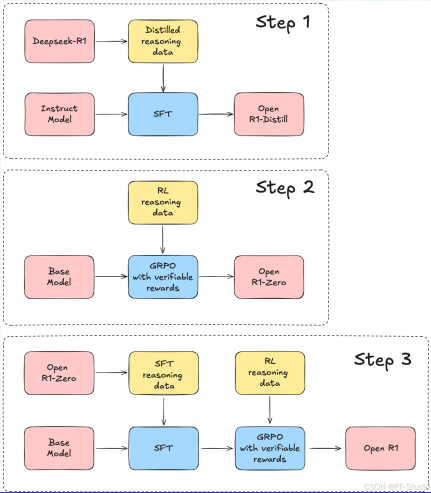
每个奖励函数都有其特定的实现逻辑，通常通过对生成结果进行解析和比较来计算得分。

- **正确性奖励函数**：通过提取生成答案中的内容，与真实答案进行比较，返回相应的分数。
- **格式奖励函数**：使用正则表达式检查生成答案是否符合预定格式，返回相应的分数。

DeepSeek R1 解读

论文解读

- **简化版 Pipeline 如图**：实际 Pipeline 比较复杂（Checkpoint 生成数据），R0 虽然输出胡言乱语，但 Reasoning 数据质量足够（拒绝采样挑出来好的，也可以整理成人话，作为高质量数据集）。
- **RL 与 SFT 交错**：RL：提升智力；SFT：增加知识量，规范格式。（现阶段）RL 会让模型输出胡言乱语，SFT 会让模型变笨。
- **小模型蒸馏有效**：对小模型 RL，不如用高质量 Reasoning 数据 SFT。



参考文献

- DeepSeek R1 论文《DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning》
- DeepSeek-R1 译文及论文笔记《DeepSeek-R1：通过强化学习激发大语言模型的推理能力》
- GRPO 论文《DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models》
- GRPO 译文《DeepSeekMath: 推动开放语言模型在数学推理能力上的极限》
- 【DeepSeek】一文详解 GRPO 算法——为什么能减少大模型训练资源？
- 【DeepSeek】LLM 强化学习 GRPO Trainer 详解
- Open R1 项目《A fully open reproduction of DeepSeek-R1》
- 【DeepSeek】复现 DeepSeek R1？快来看这个 Open R1 项目实践指南
- 【解惑】Steps、Epochs、Batchsize？梯度累计步数、样本数？他们有什么关系？