Deepseek

llm基于Transformer → BERT (encoder) 编码
→ GPT (decoder) openAI 生成式注用
→ T5 (encdecoder) 谷歌清华 GLM…

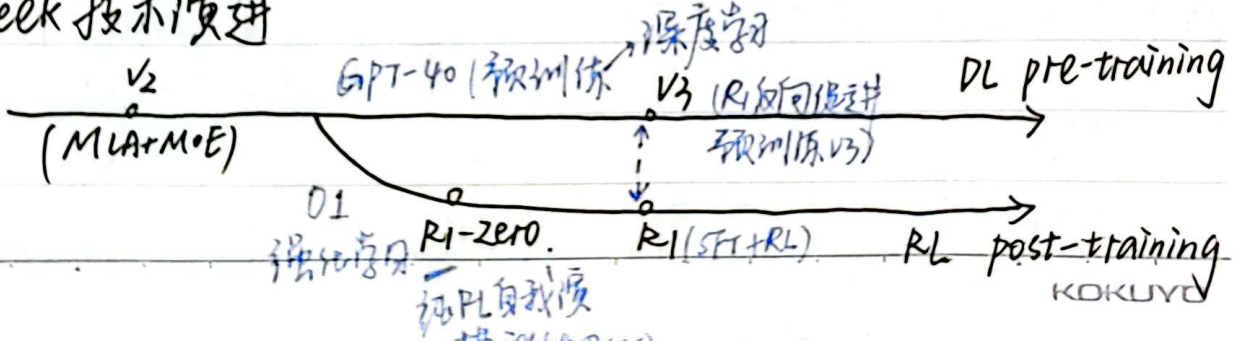
早期围于"预训练+微调"范式 (pretraining + finetune)
Pretraining 缺点(1高) 在非常大的数据集上从头训练模型.需要大量计算资源和时间.
1. 数据集要求高
2. 模型参数规模要求高
3. 算力要求高

Deepseek 预训练 V3模型.
- MLA (Multi-Head Latent Attention) 降维.
- DeepseekMoE 混合专家模型.
- R1 提升推理性能 → post-training 的一种.

后处理机制
- SFT supervised fine-tune "小段"对下游任务加训
- RLHF (GPT) 引入强化学习
- R1 还是强化学习重强.
- KD (knowledge Distillation) 知识蒸馏

区分fine-tune / KD
微调的来源是额外被标注的数据集
而KD是用大模型提供的知识,让小模型拟合大模型.
训练的是小模型 (A New model!)

Deepseek 技术演进



No.

Date

R1-Zero. 损失函数

损失函数：GPRO(Group Relative Policy Optimization)+更改损失函数

$$J_{GPRO}(\theta)=\mathbb{E}[q\sim P(Q),\{o_i\}_{i=1}^{G}\sim \pi_{\theta old}(O|q)]$$

$$=\mathbb{E}_{[q\sim P(Q),\{o_i\}_{i=1}^{G}\sim \pi_{\theta old}(O|q)]}\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta old}(o_i|q)}A_i,\ clip\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta old}(o_i|q)},1-\varepsilon,1+\varepsilon\right)A_i\right)-\beta\mathbb{D}_{KL}(\pi_\theta\|\pi_{ref})\right)$$

策略模型

两个随机变量.

q = query 输入的问题查询

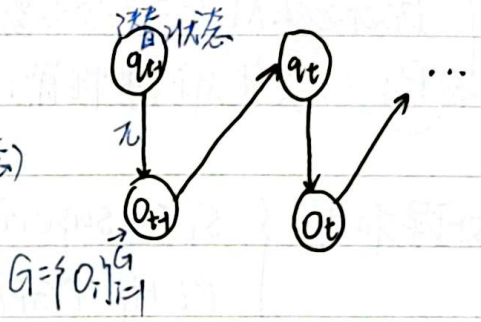
$$\mathbb{D}_{KL}(\pi_\theta\|\pi_{ref})=\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)}-\log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)}-1.$$

G: 所有回答构成的样本空间

o_i: 每个生成的回答

$\pi_{\theta old}(o|q)$ 策略模型 输出 ↑ 输入 (token状态)

old: 原/之前/老策略.



$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta old}(o_i|q)}A_i,\ clip\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta old}(o_i|q)},1-\varepsilon,1+\varepsilon\right)A_i\right)-\beta\mathbb{D}_{KL}(\pi_\theta\|\pi_{ref})\right)$$

重要性采样

权重 w "刚刚" | 价值函数 Q-V 优势 Advantage "采小车" | "考驾驶" 理想值

w: $\pi_\theta(o_i|q)$ 以 q 生成 o_i 的概率 (在现在策略 π_θ 下)

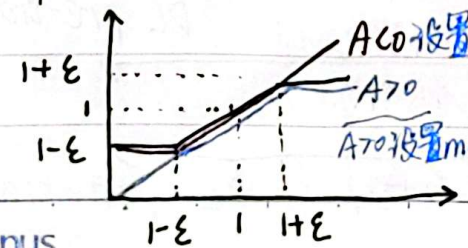
$\pi_{\theta old}(o_i|q)$ 以 q 生成 o_i 的概率 (在原策略 $\pi_{\theta old}$ 下)

w>1 作放大此 A_i

A_i: 相当于给 $o_1, o_2, \cdots, o_{t-1}, o_t, \cdots$ 生成的序列打分函数

函数中 min、clip、KL 都是均源于 GPRO 近端优化策略算法 (PPO 改进) Proximal

用 KL 项进行正则化约束



clip 划分区间→充当"安全带"/"限速 80/120" 高速公路

防止在优化过程中变化过大，导致模型崩溃/偏到边缘

clip: 限制 w 最小值最大值 $[1-\varepsilon,1+\varepsilon]$

No.

Date

损失函数：{未体现方向感 / 未体现各种情况下场景

GPRO 损失函数目的：防止策略更新过大不稳定导致崩溃.
稳定策略优化过程.

KL散度：

标准KL散度计算公式：$\mathbb{D}_{KL}(\pi_\theta|\pi_{ref}) = \sum_i \pi_\theta(o_i|q) \log \frac{\pi_\theta(o_i|q)}{\pi_{ref}(o_i|q)}$

变体公式：$\mathbb{D}_{KL}(\pi_\theta|\pi_{ref}) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1$

此处可以进

本质：实现非线性制约.
"方向阻尼运动"

奖励函数 A (reward)

价值函数

$$A_i = \frac{r_i - \text{mean}(r_1, r_2, \cdots, r_G)}{\text{std}(r_1, r_2, \cdots, r_G)}$$

r_i：奖励值

完全取决于 r_i（方向）

商业秘密.未披露.但仍是基于规则的设计

基于推理链条 CoT 设计奖励函数

e.g. 回答数学题
按步骤（每步的结果准确性、格式）得到相应的奖励）步步给分
奖励汇总
得到最终得分.

解题思路模型化 "白箱"

区别 RLHF 黑箱深度学习

数据来源不同

不同后处理方式：加训方式不同

- SFT 受制于数据质量
- RLHF 受制于打分数据
- R1-Zero 纯RL规则奖励 ×不标数据

Deepseek 在坚定走RL路线的同时
et 较好的解决了降低搜索难度，提升
升级效率的问题.

"基于思维链CoT的奖励过程解决了RL搜索路径优

KOKUYO

放空

R1-Zero存在的问题：

可读性和语言混合方面存在困难 → 贴近人类数据(SFT)

① 生成结果符合人类预期但人看不懂

② 中英混杂语无伦次

Deepseek-R1：SFT+RL

流程：

一次SFT 1. 冷启动：用SFT优化强化学习的起始点(可控)

人工"过滤+后处理"收集一批数据，定义了输出数据格式样例

用SFT训练得到能生成清晰思维链的初始策略

一次RL 2. 面向推理的RL：准确性奖励+语言一致性奖励

针对性给出同类题思维链 → 奖励函数

二次SFT 3. 拒绝采样与监督微调

用RL训练出的策略生成数据、人工筛选、过滤、优化、SFT.

二次RL 4. 全场景RL