OpenSeek 大模型挑战赛--决赛技术报告

ccabcca06

一、任务描述

决赛阶段以 Openseek-small-v1-SFT 模型为起点，可自由使用除评测集以外的训练数据集，通过强化学习优化模型对数学问题的求解能力。训练完毕后在 GSM8K、AMC23、AIME24、MATH500、MINERVA_MATH、OLYMPIADBENCH 共 6 个数学评测数据集上评估模型性能。

OpenSeek-Small-v1-SFT 文档

原文 译文

概述

我们采用 Octothinker 来构建强大的推理基础。我们的模型训练分为两个阶段：首先在包含2000亿个标记的数学语料库上进行中期稳定训练，然后是200亿个标记的衰减阶段。随后，我们在 Infinity-Instruct 数据集上对模型进行微调，以实现卓越的指令跟随能力。此模型作为开源基准，用于未来的实验，例如通过强化学习增强小型模型的推理能力。模型架构与OpenSeek-Small-v1模型相同。

评估

指标	GSM8K	MATH-500	Minerva Math	OlympiadBench	平均
Pass@1	20.698	13.100	3.470	2.741	10.002
Pass@4	41.768	19.100	8.415	4.997	18.570
Pass@8	51.838	19.599	11.680	5.185	22.075

Openseek-Small-v1-SFT 官方 Benchmark

二、整体思路

数据集方面，我们在决赛中仅使用了 GSM8K 的训练集(非评估的测试集)，主要通过强化学习算法的选择与优化提升模型的能力。

针对 Openseek-small-v1-SFT 模型以及任务的情况，我们在 PPO 算法上进行了一些改进。对比传统 PPO 算法 Actor 和 Cretic 模型使用同一个基准模型的做法，我们将 PPO 中的 Crtic-Model 替换为 Deepseek-R1 0528 Distill-Qwen 8B，利用 PPO 的优势函数计算与策略梯度更新机制，将 8B 模型的能力注入到 1.5B 模型中，有效激活模型的数学推理能力。

三、强化学习算法选择与优化

本次竞赛我们使用 VeRL 强化学习框架，原计划是基于 GSM-8K 训练数据集和 GRPO 强化学习算法快速构建 Baseline，再围绕 Baseline 进行迭代和优化。但是我们进行了多次 GRPO 训练，实验了大量参数配置后，仍然无法正常训练。对训练的过程数据进行分析后，发现是因为激励过于稀疏，导致训练出现模式崩溃的问题。

```
core/openai/gsm8k/reward/mean@1': ''
(TaskRunner pid=132703)  '0.08946171341925702'
(TaskRunner pid=132703) step:0 - val-core/openai/gsm8k/reward/mean@1:0.08946171341925702
Training Progress:   1%|          | 1/140 [00:35<1:21:42, 35.27s/it]
(TaskRunner pid=132703) step:1 - global_seqlen/min:63925 - global_seqlen/max:73442 -
global_seqlen/minmax_diff:9517 - global_seqlen/balanced_min:71126 - global_seqlen/
balanced_max:71127 - global_seqlen/mean:71126.25 - actor/entropy:6.286609649658203 -
actor/reward_kl_penalty:0.0 - actor/reward_kl_penalty_coeff:0.001 - actor/kl_loss:0.
43240806568064727 - actor/kl_coef:0.01 - actor/pg_loss:0.022381234884960577 - actor/
pg_clipfrac:0.05861966409247543 - actor/ppo_kl:-0.37522271115449257 - actor/
pg_clipfrac_lower:0.02601106515066931 - actor/grad_norm:2.094236046075821 - perf/mfu/
actor:0.11034187900452268 - perf/max_memory_allocated_gb:89.35889911651611 - perf/
max_memory_reserved_gb:97.126953125 - perf/cpu_memory_used_gb:112.19536972045898 -
actor/lr:5e-06 - training/global_step:1 - training/epoch:0 - critic/score/mean:0.
05908203125 - critic/score/max:1.0 - critic/score/min:0.0 - critic/rewards/mean:0.
05908203125 - critic/rewards/max:1.0 - critic/rewards/min:0.0 - critic/advantages/
mean:-0.011124623008072376 - critic/advantages/max:1.4999970197677612 - critic/
advantages/min:-1.4999970197677612 - critic/returns/mean:-0.011124623008072376 - critic/
returns/max:1.4999970197677612 - critic/returns/min:-1.4999970197677612 -
response_length/mean:183.9248046875 - response_length/max:512.0 - response_length/min:5.
0 - response_length/clip_ratio:0.02587890625 - response_length_non_aborted/mean:183.
```

GRPO 训练报文：step 0

step:25 - global_seqlen/min:174621 - global_seqlen/max:192014 - global_seqlen/minmax_diff:17393 - global_seqlen/balanced_min:184199 - global_seqlen/balanced_max:184200 - global_seqlen/mean:184199.25 - actor/entropy:8.635790824890137 - actor/reward_kl_penalty:-0.29761677980422974 - actor/reward_kl_penalty_coeff:0.001 - actor/kl_loss:0.16839072154834867 - actor/kl_coef:0.005 - actor/pg_loss:0.054942668648436666 - actor/pg_clipfrac:0.30638663587888004 - actor/ppo_kl:-0.6100024281367951 - actor/pg_clipfrac_lower:0.10227466002106667 - actor/grad_norm:3.3316752910614014 - perf/mfu/actor:0.07929720494641407 - perf/max_memory_allocated_gb:102.30215978622437 - perf/max_memory_reserved_gb:109.599609375 - perf/cpu_memory_used_gb:120.65269088745117 - actor/lr:5e-07 - val-core/openai/gsm8k/reward/mean@1:0.008339651250947688 - training/global_step:25 - training/epoch:3 - critic/score/mean:0.00439453125 - critic/score/max:1.0 - critic/score/min:0.0 - critic/rewards/mean:0.06524507701396942 - critic/rewards/max:1.0841209888458252 - critic/rewards/min:-0.48161572217941284 - critic/advantages/mean:0.22142961621284485 - critic/advantages/max:1.4999034404754639 - critic/advantages/min:-1.4997638463974 - critic/returns/mean:0.22142961621284485 - critic/returns/max:1.4999034404754639 - critic/returns/min:-1.4997638463974 - response_length/mean:267.11767578125 - response_length/max:512.0 - response_length/min:2.0 - response_length/clip_ratio:0.249267578125 - response_length_non_aborted/mean:267.11767578125 - response_length_non_aborted/max:512.0 - response_length_non_aborted/min:2.0 -

GRPO 训练报文：step 25

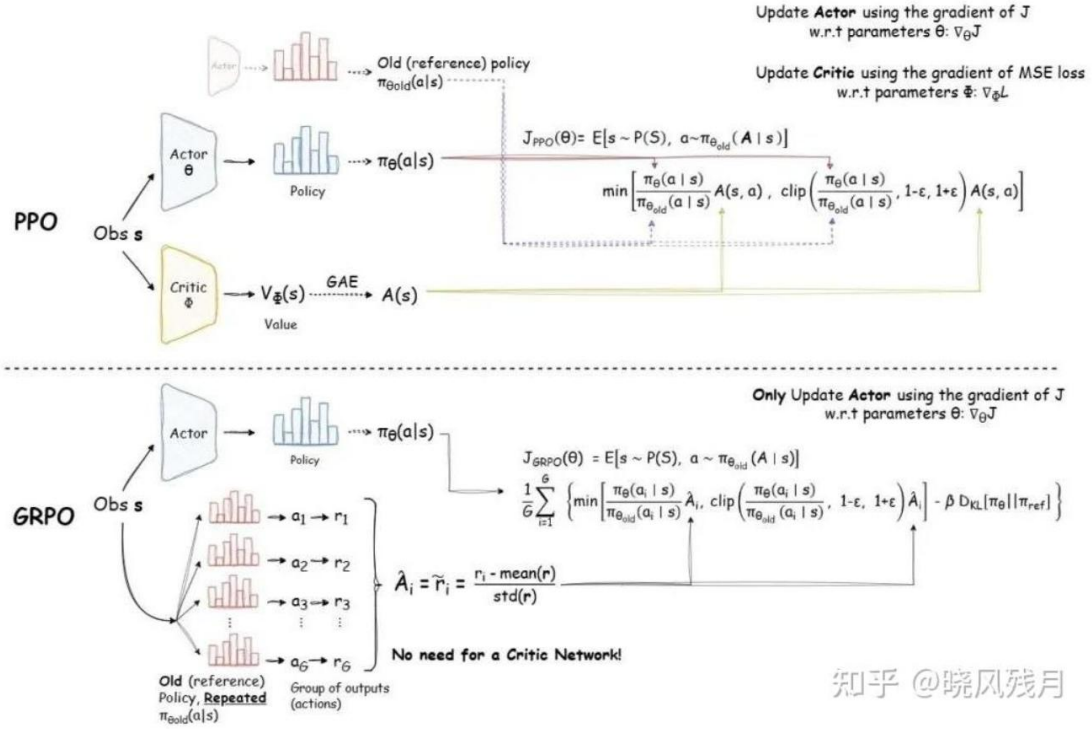

```
{
    "dataset_acc": {
        "olympiadbench": 0.51852,
        "minerva_math": 1.53952,
        "aime24": 0.0,
        "gsm8k": 0.56861,
        "math500": 4.5,
        "amc23": 0.0
    },
    "final_acc": 1.187775
}
```


训练结果(大幅低于 Benchmark)

GRPO 训练中，初赛奖励信号极其微弱并且在训练中持续下降：验证奖励从 0.089（初始）->0.063（第 5 步）->0.025（第 10 步）->0.008（第 25 步），这表明模型在强化学习训练中可能没有学习到正确的行为，甚至在学习错误的行为。

GRPO 是一种 On-Policy 的强化学习算法，在 PPO 的基础上，GRPO 使用了 Reward 的统计量来替代优势函数，省掉了 Cretic 模

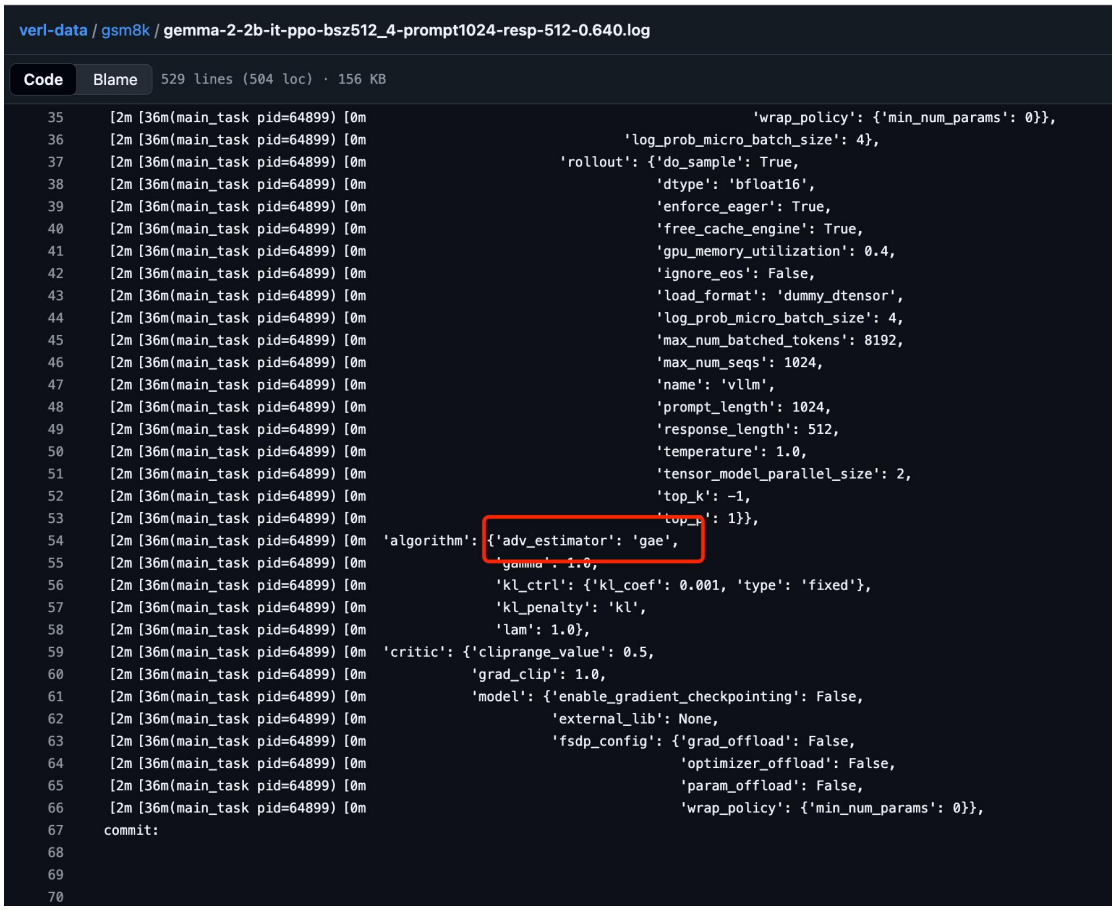
型。但是对于 1.5B 的 Openseek-small-v1-SFT 模型，推测是由于初始模型数学求解能力较弱的原因，生成了大量的错误答案，GRPO 作为一种从自身输出采样的 On-Policy 算法，这种条件下一直无法获取到有效的 Reward，导致奖励稀疏，进一步引发后续训练的模式崩溃。



PPO vs GRPO(图片来源:https://zhuanlan.zhihu.com/p/24456498603)

我们查找了 VeRL 官方文档和实验日志:

https://verl.readthedocs.io/en/latest/algo/baseline.html，以寻求在 1.5B 模型上稳定开展 GRPO 训练的设置。但是发现即使在 VeRL 的 GRPO 目录下，与竞赛模型参数相近的 gemma2-2b、qwen2.5-0.5B 均在训练日志中显示使用的是 PPO 算法，在 7B 以上参数规模的模型上的模型才开始使用 GRPO。因此我们猜测：只有 7B 以上参数规模的模型，具备了一定的推理理解能力，才能够为 GRPO 的冷启动提供足够的 Reward。

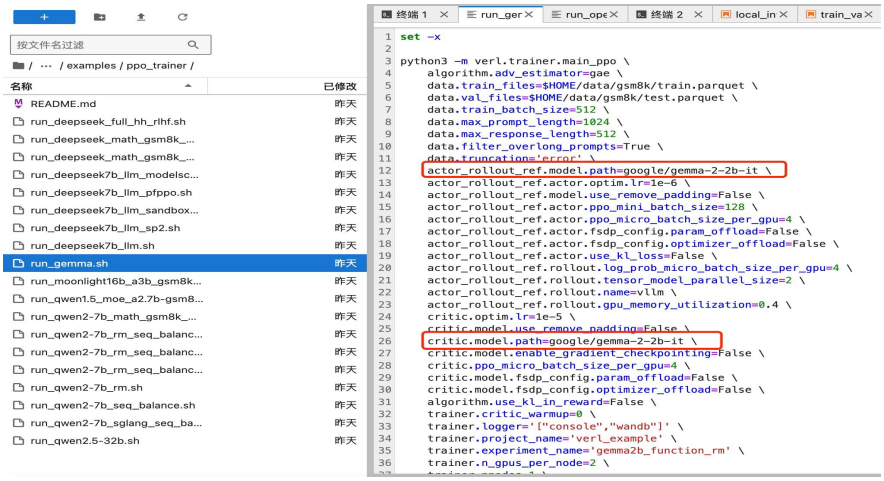


gemma-2b 实验设置(adv_estimator=gae 证明使用的是 ppo 算法)

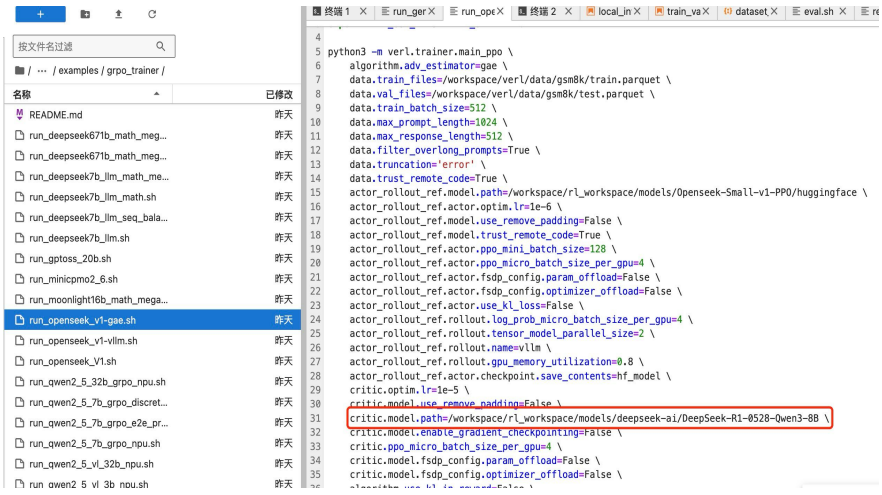
VeRL 的 PPO 算法实现中，通常使用 LLM 作为 Critic 模型。PPO 的 Critic 模型用于近似价值函数，输入为状态 s，估计当前状态的价值 V。在 NLP 任务中，状态价值 V 就是给定输出上文字符串 s 的状态下，求和所有下一个字符出现概率与字符对应价值的乘积。Action 的维度就是词表的长度，Actor 模型是强化学习的 Policy 模型，也是训练的目标模型，Actor 生成的每一个 token 就是一个 action，生成这个 token 的上文对应的强化学习概念是 state，而一个完整的句子则是一条轨迹 (trajectory)。Critic 模型的词表应该与 Actor 一致，对不同轨迹每一个 token 生成的收益进行估计，以指示策略梯度的更新动作。

Critic 模型的预测输出与大语言模型 CasualLM 的输出机制非常匹配，而 Critic 近似优势函数，指导策略梯度更新，对于 LLM 某种程度上来说也是将 Critic 模型中的知识“蒸馏”到了 Actor 中。

在 VeRL 的默认 PPO 实现中, actor 和 critic 通常会使用同一个模型，在本次竞赛的强化学习实验中，我们将 critic 替换为与 Openseek-small-v1-SFT 具有相同词表结构的 Deepseek R1-0528 Distill-Qwen3 8B 模型，利用 PPO 的 Actor-Critic 架构，将更强大的 8B 模型中的知识通过策略梯度的更新蒸馏到 1.5B 模型中。



verl/example/ppo_trainer/run_gemma.sh



我们的训练设置，将 critic.model 替换为 8B 模型

四、实验记录

4.1 Benchmark

使用 openseek-small-v1-sft，直接运行评测脚本，最终评测得分为 7.65，具体如下：

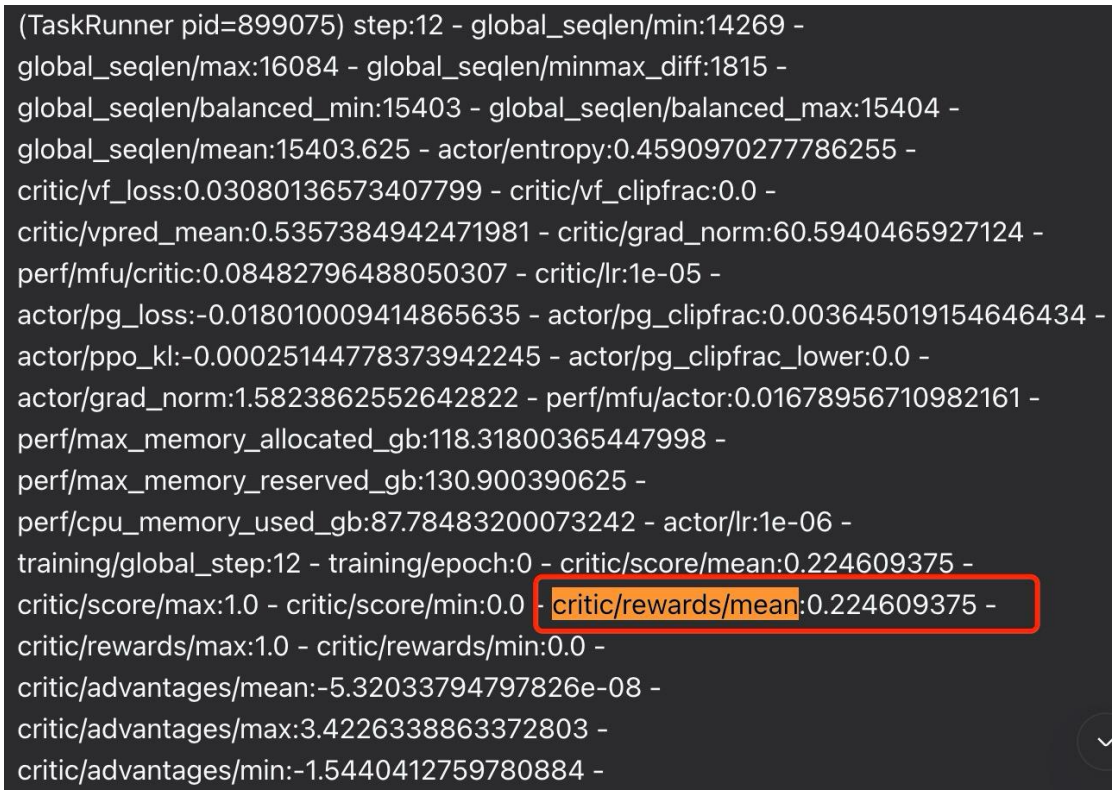
{'num_samples': 40, 'num_scores': 640, 'timeout_samples': 0, 'empty_samples': 0, 'acc': 4.84375, 'pass@k': {'pass@1': 4.84375, 'pass@4': 10.63462, 'pass@8': 12.49456}}
{
  "dataset_acc": {
    "aime24": 0.20833,
    "olympiadbench": 3.18519,
    "math500": 13.3,
    "minerva_math": 3.40074,
    "gsm8k": 21.00076,
    "amc23": 4.84375
  },
  "final_acc": 7.656461666666666
}

4.2 一阶段 PPO 强化学习训练

PPO 使用 VeRL 内置参考脚本，设置训练 epoch=15，train_batch_size=512，max_response_length=512 启动 PPO 训练后检查报文，可以看到初始 Reward 与 GRPO 一致，同样为 0.089：

```
    trainer.total_epochs=15 $@
前3个step的输出如下：
(TaskRunner pid=772061) [ground_truth] 18
(TaskRunner pid=772061) [score] 0.0
(TaskRunner pid=772061) len reward_extra_infos_dict['reward']: 1319
(TaskRunner pid=772061) ("Initial validation metrics: {'val-
core/openai/gsm8k/reward/mean@1': "
(TaskRunner pid=772061)  '0.08946171341925702}')
(TaskRunner pid=772061) step:0 - val-
core/openai/gsm8k/reward/mean@1:0.08946171341925702
Training Progress:   0%|          | 0/210 [00:00<?, ?it/s]
(WorkerDict pid=778294) /usr/local/lib/python3.10/dist-
packages/torch/distributed/fsdp/fully_sharded_data_parallel.py:680: FutureWarning:
FSDP.state_dict_type() and FSDP.set_state_dict_type() are being deprecated. Please
use APIs, get_state_dict() and set_state_dict(), which can support different
parallelisms, FSDP1, FSDP2, DDP. API doc:
https://pytorch.org/docs/stable/distributed.checkpoint.html#torch.distributed.checkpoi
nt.state_dict.get_state_dict .Tutorial:
https://pytorch.org/tutorials/recipes/distributed_checkpoint_recipe.html . [repeated 2x
across cluster]
```

仅训练 12 个 step 后 reward 就增加到了 0.22:



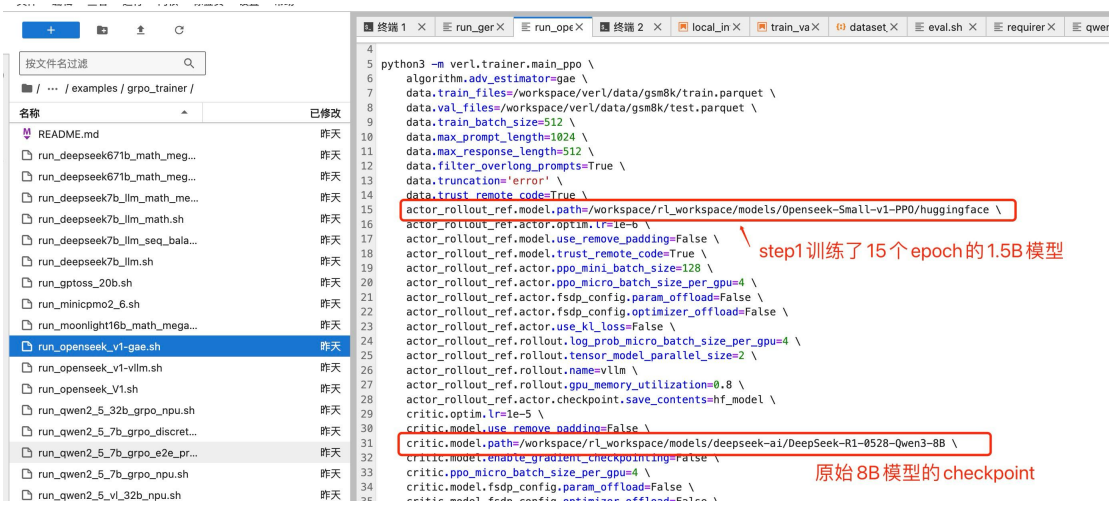
完成 15 个 epoch，最终结果为 9.26：

{'num_samples': 40, 'num_scores': 640, 'timeout_samples': 0, 'empty_samples': 0, 'acc': 5.625, 'pass@k': {'pass@1': 5.625, 'pass@4': 13.90797, 'pass@8': 17.01884}}
{
  "dataset_acc": {
    "aime24": 0.41667,
    "olympiadbench": 3.63889,
    "math500": 16.15,
    "minerva_math": 2.73438,
    "gsm8k": 27.02805,
    "amc23": 5.625
  },
  "final_acc": 9.265498333333333
}

二阶段 PPO 强化学习训练

完成一阶段训练后，我们继续训练到 25 个 epoch，发现最终结果仅轻微提升到 9.7 分，除了 gsm8k 提升到 33.8 分以外，其他数据集

的指标均发生了较大幅度的下降，例如 amc23 从 5.6 分直接下降到了 1.25 分。我们判断在继续训练 10 个 epoch 后，模型对 gsm8k 训练集已经出现了过拟合的情况。我们尝试通过重置 critic 的方式重新激活模型以继续训练：



critic 重置后，我们再再训练了 7 个 epoch，测试得分大幅上涨到了 11.03 分，证明重置 critic 的激活方式是有效的：

{'num_samples': 40, 'num_scores': 640, 'timeout_samples': 8, 'empty_samples': 0, 'acc': 12.5, 'pass@k': {'pass@1': 12.5, 'pass@4': 17.11538, 'pass@8': 17.49922}}
{
  "dataset_acc": {
    "aime24": 0.0,
    "olympiadbench": 3.68519,
    "math500": 17.4,
    "minerva_math": 3.125,
    "gsm8k": 29.49204,
    "amc23": 12.5
  },
  "final_acc": 11.033705
}

Prompt Turning

完成二阶段训练后，我们发现在 gsm8k 和 amc23 两个数据集上，使用不同风格 prompt 设置对最终测试准确率有非常大的影响。gsm8k 在设置 system prompt 以后测试成绩得到大幅提升(29.49->34.02)，而 amc23 则出现较大幅度的下降(12.5->8.7)。为此，

我们针对两组测试数据集进行了 prompt 设置。其中 amc23 数据集单独使用一组 prompt, 其他数据集使用 gsm8k prompt。两组 prompt 具体如下：

gsm8k prompt，主要设置了 system prompt:

```
    'gsm8k':(
        "<|im_start|>system\nYou are an excellent mathematics professor<|im_end|>\n"
        "<|im_start|>user\nPlease response answer, and just put your final answer within
\\boxed{{}}.\nQuestion:\n{input}<|im_end|>\n"
        "<|im_start|>assistant\n",
        "{output}",
        "\n\n"
    ),
```

amc23 设置 system prompt 后得分会大幅下降，故不设置：

```
    'amc':(
        "<|im_start|>system\n<|im_end|>\n"
        "<|im_start|>user\nPlease response answer, and just put your final answer within
\\boxed{{}}.\nQuestion:\n{input}<|im_end|>\n"
        "<|im_start|>assistant\n",
        "{output}",
        "\n\n"
```

经过提示词分组优化后，最终得分为 12.85：


```
{'num_samples': 40, 'num_scores': 640, 'timeout_samples': 4, 'empty_samples': 0, 'acc': 15.9375, 'pass@k': {'pass@1': 15.9375, 'pas
s@4': 23.58242, 'pass@8': 26.59615}}
{
   "dataset_acc": {
      "aime24": 0.0,
      "olympiadbench": 3.99074,
      "math500": 18.3,
      "minerva_math": 4.8943,
      "gsm8k": 34.02199,
      "amc23": 15.9375
   },
   "final_acc": 12.857421666666667
}
```


结论及下一步展望

我们针对本次竞赛的模型特性，利用 PPO 的 actor-critic 机制实现了 8B 模型到 1.5B 模型的知识蒸馏，探索出了在 PPO 算法中通过

重置 critic 将模型从过拟合状态中激活出来的办法，并通过测试数据集的评分效果证明了知识蒸馏和 Critic 重置方法的效果。由于时间和算力的局限性，还有大量的实验未能实施，例如使用更强大的 Critic 模型，扩展数据集等措施的效果等，我们期待后续能有机会针对这些方面开展更深入的研究。