

PR #6882 完整报告

verl-project/verl

[fully_async] fix: correct the use of partial_rollout

合并时间: 2026-06-29 15:34

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6882>

执行摘要

- 一句话: 修复 partial_rollout 禁用时丢弃中止样本的逻辑
- 推荐动作: 可以合并, 修复了废弃样本重试的逻辑。后续若 OmegaConf 配置访问方式统一, 可考虑使用 OmegaConf.select 替代 hasattr。

功能与动机

PR body 说明当 partial_rollout 禁用时, aborted (如 staleness samples) 样本应被丢弃, 但原有实现没有正确处理, 且 V1 trainer 缺乏 async_training 配置会导致逻辑错误。

实现拆解

1. 在 verl/workers/rollout/llm_server.py 的 generate 方法中, 修改停止原因检查处的条件判断。
2. 引入 should_retry 变量, 默认设为 True。
3. 通过 hasattr(self.config, "async_training") 判断是否存在 async_training 配置, 仅在存在且 partial_rollout 为 False 时将 should_retry 设为 False。
4. 将停止条件改为: 当 stop_reason 不是 "aborted"/"abort" 或 should_retry 为 False 时 break。
5. 这样 V1 trainer (无 async_training 属性) 默认 should_retry=True, 但条件中 not should_retry 为 False, 结合 stop_reason 判断, 实际上 V1 对 abort 依然会 break (因不是 abort)。但核心意图是当 partial_rollout 禁用时, aborted 样本不再重试。

关键文件:

- verl/workers/rollout/llm_server.py (模块 rollout; 类别 source; 类型 core-logic; 符号 generate): 修改了停止条件判断逻辑, 修复 partial_rollout 禁用时 abort 样本不丢弃的问题

关键符号: generate

关键源码片段

[verl/workers/rollout/llm_server.py](#)

修改了停止条件判断逻辑, 修复 partial_rollout 禁用时 abort 样本不丢弃的问题

```
# 关键位置: generate 方法中停止原因检查
# 原代码 (head 版本前三行是已有逻辑, 后七行为本次修改)
# ... 已完成 token 拼接和 max_new_tokens 更新 ...

# 4. check stop reason
# 如果 partial_rollout 未启用, aborted 样本应被丢弃。
# 对于 V1 trainer, should_retry 始终为 True, 因为 self.config.async_training 不存在。
should_retry = True
if hasattr(self.config, "async_training") and not self.config.async_training.partial_rollout:
    should_retry = False
# 当 stop_reason 不是 aborted 或不应重试时, 跳出循环
if output.stop_reason not in ("aborted", "abort") or not should_retry:
    break

await asyncio.sleep(1)
```

评论区精华

1. gemini-code-assist[bot] 指出使用 `hasattr` 和直接属性访问 `OmegaConf DictConfig` 不安全, 建议使用 `OmegaConf.select` 安全访问嵌套配置。但 reviewer wuxibin89 指出 V1 trainer 没有 `async_training`, 直接 `hasattr` 会导致 V1 无法正确 `retry`, 因此最终实现采用 `hasattr` 加条件判断。
2. 该 PR 没有其他审查评论。
 - 使用 `OmegaConf.select` 替代 `hasattr` 的安全性 (design): 未采纳, 因为 V1 trainer 无 `async_training` 配置, 使用 `hasattr` 是预期行为; `OmegaConf` 当前未启用 `struct` 模式
 - V1 trainer 缺少 `async_training` 配置导致不重试 `abort` (correctness): 已修复, 通过 `should_retry` 变量确保 V1 默认不重试 `abort`

风险与影响

- 风险: 风险低。仅涉及一个条件判断逻辑的修改, 影响范围局限于 rollout 的 `abort` 重试行为。但需注意: 如果未来 `OmegaConf` 启用 `struct` 模式, `hasattr` 可能失效, 但当前仓库未启用。
- 影响: 影响范围小, 仅影响使用 `partial_rollout` 功能的异步训练流程。对于 V1 trainer (无 `async_training` 配置), 行为与之前一致 (始终不重试 `abort`)。对于 V2 `fully_async` 训练, 当 `partial_rollout` 禁用时, `aborted` 样本将不再重试, 直接丢弃, 减少无效生成。
- 风险标记: 条件判断隐式依赖配置结构, 缺少测试覆盖

关联脉络

- 暂无明显关联 PR