

PR #2682 完整报告

radixark/miles

fix: resume from the checkpoint step in bridge mode

合并时间: 2026-08-21 07:14

原文链接: <http://prhub.com.cn/radixark/miles/pull/2682>

执行摘要

- 一句话: 桥接模式恢复训练从检查点步骤继续
- 推荐动作: 此 PR 值得精读, 特别是理解桥接模式的恢复逻辑。设计上通过在参数验证阶段调整 `start_rollout_id` 的默认值, 避免修改下游 `create_training_models` 的复杂逻辑, 是简洁且低风险的修复。重点关注 Codex 提出的 `release tracker` 边界情况, 未来可考虑补充处理。

功能与动机

PR 描述明确说明了症状和复现步骤: 在桥接模式下使用 `--load` 加载保存的检查点, 期望从 rollout N+1 继续, 但实际从 rollout 0 重新开始。根本原因是 `miles_validate_args` 无条件设置 `start_rollout_id = 0`, 而 `create_training_models` 仅在 `start_rollout_id` 为 `None` 时才采用加载的 rollout id。

实现拆解

1. 定位问题: 在 `miles/utils/arguments.py` 的 `miles_validate_args` 函数中, 桥接模式分支无条件设置 `args.start_rollout_id = 0`, 导致即使有有效检查点也强制从 0 开始。
2. 修改条件赋值: 将 `args.start_rollout_id = 0` 移入检查点缺失的条件块内, 与已有 `args.load` 回退逻辑并列。当 `--load` 指向有效检查点时, `start_rollout_id` 保持 `None`, 从而让下游 `create_training_models` 采用加载的迭代号。
3. 保持新鲜运行行为: 对于无检查点的运行, 仍显式设置为 0, 因为 `_load_checkpoint_hf` 报告迭代 0, 防止 actor 返回 1。

关键文件:

- `miles/utils/arguments.py` (模块 参数校验; 类别 `source`; 类型 `core-logic`; 符号 `miles_validate_args`): 此文件包含 `miles_validate_args` 函数, 其中将 `start_rollout_id` 的赋值移入条件块, 是修复核心。

关键符号: `miles_validate_args`

关键源码片段

`miles/utils/arguments.py`

此文件包含 `miles_validate_args` 函数，其中将 `start_rollout_id` 的赋值移入条件块，是修复核心。

```
# miles/utils/arguments.py ( 在 miles_validate_args 函数中 )
# 桥接模式分支：处理 --load 参数，并决定 start_rollout_id
if args.megatron_to_hf_mode == "bridge":
    # 新鲜运行时，--load 目录尚未创建或无效，回退到参考权重（通过 HF 桥加载）
    if (
        args.load is None
        or not os.path.exists(args.load)
        or not os.path.exists(os.path.join(args.load, "latest_checkpointed_iteration.txt"))
    ):
        args.load = args.ref_load or args.hf_checkpoint
        # 仅在无有效检查点时才强制从 rollout 0 开始
        # 有检查点时保持 None，以便 create_training_models 采用加载的迭代号
        args.start_rollout_id = 0
        # 注意：有有效检查点时不再设置 start_rollout_id，保持 None
# 非桥接模式分支（供参考）
else:
    if (
        args.load is None
        or not os.path.exists(args.load)
        or not os.path.exists(os.path.join(args.load, "latest_checkpointed_iteration.txt"))
    ):
        args.no_load_optim = True
        args.no_load_rng = True
        args.finetune = True
        args.load = args.ref_load
        if args.ref_ckpt_step is not None:
            args.ckpt_step = args.ref_ckpt_step
        args.start_rollout_id = 0
```

评论区精华

唯一实质性 review 评论来自 ChatGPT Codex 机器人，提出 P2 级别建议：当 `--load` 指向转换后的 Megatron 发布检查点时，`tracker` 文件存在但内容是 'release' 而非可恢复的 rollout id，此时应视为新启动，需要将 `start_rollout_id` 设置为 0。但该建议未被修改，PR 已合并。

- release tracker 应视为新鲜开始 (correctness): 未被修改，PR 已合并。该边界情况未处理，可能仍有潜在问题。

风险与影响

- 风险：该改动会导致在有效检查点存在时，`start_rollout_id` 不再显式设置为 0。如果检查点内容无法被 `create_training_models` 正确解析（例如 optimizer 状态缺失），可能仍然从错误迭代开始。Codex 指出的 release tracker 问题是一个未解决的边界情况，但当前改动未处理。

- 影响：影响桥接模式（--megatron-to-hf-mode bridge）下使用 --load 恢复训练的用户。修复后，恢复训练从检查点步骤继续而不是重新开始，节省计算资源，并保证训练连续性。对新鲜运行无影响。
- 风险标记：边界情况未覆盖，缺少测试覆盖，核心路径变更

关联脉络

- PR #2576 fix: skip rollout construction for debug replay: 涉及 rollout 构造逻辑，与本 PR 的恢复逻辑相关。