

PR #6195 完整报告

verl-project/verl

[worker] fix: grad_norm as non-tensor for metrics

合并时间: 2026-04-29 11:13

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6195>

执行摘要

- 一句话: 转换 grad_norm 为标量避免 Ray 序列化崩溃
- 推荐动作: 建议合并, 改动小而精确。可考虑在类似度量 (如 lr) 上也做防御性转换以增加鲁棒性。

功能与动机

训练时因 Ray 尝试在 CPU-only 节点上反序列化 CUDA Tensor 而崩溃 (错误: `torch.cuda.is_available() is False`), 根因是 `grad_norm` 保留为 Tensor 类型未转换。

实现拆解

1. 修改文件: verl/workers/engine_workers.py 中的 `_postprocess_output` 方法。
2. 逻辑变更: 在 `metrics.pop("grad_norm", None)` 后, 增加 `isinstance(grad_norm, torch.Tensor)` 判断, 若为 Tensor 则通过 `.detach().item()` 转为 Python float。
3. 无其他改动: 初版中涉及 LoRA 权重同步的无关删除 (`sleep_level`、`base_sync_done`) 经 reviewer 指出后由作者回退, 最终仅保留该 2 行防御性转换。

关键文件:

- verl/workers/engine_workers.py (模块 工作进程; 类别 source; 类型 core-logic; 符号 `_postprocess_output`): 唯一变更文件, 在 `_postprocess_output` 中添加 Tensor 转标量逻辑, 是修复的核心。

关键符号: `_postprocess_output`

关键源码片段

`verl/workers/engine_workers.py`

唯一变更文件, 在 `_postprocess_output` 中添加 Tensor 转标量逻辑, 是修复的核心。

文件: verl/workers/engine_workers.py, 位于 `_postprocess_output` 方法内

```
# For grad_norm, we do not perform all reduce because it is already been done when
clipping grad
grad_norm = metrics.pop("grad_norm", None)
if isinstance(grad_norm, torch.Tensor):
```

```
grad_norm = grad_norm.detach().item() # 转为 Python 标量避免 Ray 在 CPU
节点上反序列化出错
lr = metrics.pop("lr", None)
# ... 后续逻辑保持不变 ...
if grad_norm is not None:
    final_metrics["grad_norm"] = grad_norm
```

评论区精华

- **lr 也应转换?** : gemini-code-assist[bot] 建议对 lr 做相同处理, 作者未采纳 (未遇到的问题无需过度防御)。
- 无关变更争议: copilot-pull-request-reviewer[bot] 指出初始 diff 中还删除了 `self.rollout.sleep_level = 1` 和 `self.base_sync_done = True`, 经作者澄清为误引入的无关修改, 在最终合并时已回退。
- 版本问题: wuxibin89 询问是否为新的 mcore 版本所致, 作者确认使用了 mcore main 分支, 但未进一步定位上游变更。
 - 对 lr 也做相同转换 (correctness): 作者未采纳, 认为当前仅 grad_norm 出现此问题。
 - 无关变更 (sleep_level 和 base_sync_done 删除) (other): 作者澄清为无关修改, 并在最终版本中还原, 仅保留 grad_norm 转换。
 - 版本关联性确认 (question): 作者确认使用 mcore main 分支, 但未追踪具体上游变更。

风险与影响

- 风险: 风险极低: 变更仅为 2 行防御性代码, 仅在 grad_norm 为 Tensor 时生效, 对原有逻辑无副作用。但若 grad_norm 为其他不可序列化类型 (如 numpy array) 则不会处理, 不过当前路径中仅可能出现 Tensor 或 None。
- 影响: 影响范围限于使用 Megatron-Core 最新版且 grad_norm 返回 Tensor 的训练任务。对其他用户无感知, 修复后训练可正常进行。
- 风险标记: 序列化错误风险, 缺乏测试覆盖

关联脉络

- 暂无明显关联 PR