

PR #6913 完整报告

verl-project/verl

[fsdp] fix: fused-kernel gradient tracking bug.

合并时间: 2026-07-07 10:56

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6913>

执行摘要

- 一句话: 修复融合核梯度追踪 bug
- 推荐动作: 值得精读, 展示了 autograd 中 `.flatten()` 易被忽略的副作用。维护者可考虑后续采纳 `ctx.needs_input_grad` 的改进方案。

功能与动机

使用 `use_liger` 和 `use_fused_kernel` 标志时, 训练结果被静默破坏: backbone 冻结, 仅 `lm_head` 仍在训练。根本原因是 fused kernel autograd `forward` 内的 `.flatten(0, 1)` 会翻转 `requires_grad` 标志, 导致 `backward` 中 `hidden_states.requires_grad` 为 `False`, 梯度返回 `None`, 影响 reward 提升。

实现拆解

1. 在 `verl/utils/experimental/torch_functional.py` 的 `FusedLinearForPPOFunction.forward` 中, 对 3D `hidden_states` 执行 `.flatten(0, 1)` 前, 先保存原始 `requires_grad` 值。
2. 执行 `flatten` 后, 立即调用 `.requires_grad_()` 恢复该标志。
3. 未修改其他逻辑和 API, 改动仅 2 行新增代码, 零删除。
4. 未包含测试用例变更, 作者称已离线测试。

关键文件:

- `verl/utils/experimental/torch_functional.py` (模块 训练工具; 类别 `source`; 类型 `core-logic`; 符号 `FusedLinearForPPOFunction.forward`): 核心修复文件, 修改了 fused kernel 的 autograd `forward` 逻辑。

关键符号: `FusedLinearForPPOFunction.forward`

关键源码片段

`verl/utils/experimental/torch_functional.py`

核心修复文件, 修改了 fused kernel 的 autograd `forward` 逻辑。

```
class FusedLinearForPPOFunction(torch.autograd.Function):
    @staticmethod
    def forward(
```

```

ctx,
hidden_states: torch.FloatTensor,
vocab_weights: torch.FloatTensor,
input_ids: torch.LongTensor,
temperature: float = 1.0,
chunk_size: int = 512,
) -> tuple[torch.FloatTensor, torch.FloatTensor]:
    ctx.set_materialize_grads(False)

    orig_ndim = hidden_states.ndim
    assert orig_ndim in (2, 3), f"Invalid hidden_states shape, received {hidden_states.shape}"

    orig_batch_size = -1
    if orig_ndim == 3:
        assert input_ids.ndim == 2, f"input_ids shape doesn't match, {hidden_states.shape}
        {input_ids.shape}"
        orig_batch_size = hidden_states.shape[0]
        # 保存原始 requires_grad 标志, 因为 .flatten(0, 1) 会静默重置它
        hidden_states_requires_grad = hidden_states.requires_grad
        hidden_states = hidden_states.flatten(0, 1)
        # 恢复标志, 确保 backward 能正确追踪 backbone 梯度
        hidden_states.requires_grad_(hidden_states_requires_grad)
        input_ids = input_ids.flatten(0, 1)

    # ... 后续前向计算逻辑保持不变 ...

```

评论区精华

审查者 yueyiming2009 指出更稳健的方案应使用 `ctx.needs_input_grad` 在 `backward` 中门控, 而非恢复中间张量的 `requires_grad`。修复作者 kolehma8 原始实现使用了直接属性赋值, 经 gemini-code-assist[bot] 建议改为 `requires_grad_()` 方法并修正变量命名。最终维护者 wuxibin89 认可 yueyiming2009 的方案更优, 但当前修复已合并。

- 使用 `ctx.needs_input_grad` 替代 `requires_grad` 恢复 (design): 维护者 wuxibin89 认为该方案更优, 但当前 PR 的简单修复已被合并。
- 变量命名与 in-place 方法 (style): 作者采纳建议并更新了代码。

风险与影响

- 风险: 改动极小, 仅影响 3D `hidden_states` 路径, 不改变其他功能。恢复 `requires_grad` 可能因 autograd 规则变化而脆弱, 但当前修复是安全的。未引入新测试, 后续重构可能被意外回退。
- 影响: 仅影响使用 fused kernel (`use_liger/use_fused_kernel`) 且 `hidden_states` 为 3D 的场景。修复前训练静默失败 (`backbone` 不更新), 修复后正常。对用户无 API 变更, 不需要配置调整。
- 风险标记: 核心路径变更

关联脉络

- 暂无明显关联 PR