

PR #6409 完整报告

verl-project/verl

[ckpt, model] fix: save LoRA train metadata for PPO actor checkpoint

合并时间: 2026-05-22 22:03

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6409>

执行摘要

- 一句话: 修复 PPO actor LoRA 检查点丢失训练元数据
- 推荐动作: 值得阅读的 PR, 展示了如何通过扩展 CheckpointManager 而非修改 trainer 来支持新功能, 以及基于分布式后端设计 (FSDP vs Megatron) 决定是否复用已有机制的权衡决策。review 中关于 Megatron Bridge 的讨论尤其值得关注。

功能与动机

Issue #6380 报告了 PPO 使用 LoRA 训练时, 合并检查点得到的适配器 lora_alpha 变为 0, 导致合并后模型无有效 adapter 效果。问题根源在于 PPO actor 检查点未像 SFT 检查点 (PR #5326) 一样保存 lora_train_meta.json。本 PR 修复此遗漏。

实现拆解

1. 在 FSDPCheckpointManager 中添加私有方法 `_get_lora_train_meta`, 从 `unwrap_model` 的 `peft_config` 中提取 `r`、`lora_alpha`、`task_type`, 并返回字典 (如果 `peft_config` 不存在或 `r <= 0` 则返回 `None`)。
2. 添加 `_save_lora_train_meta` 方法, 调用 `_get_lora_train_meta`, 若非 `None` 则在 `local_path` 下写入 `lora_train_meta.json`。
3. 在 `save_checkpoint` 方法中, 写入 `fsdp_config` 后立即调用 `_save_lora_train_meta(local_path, unwrap_model)`, 确保 LoRA 元数据随检查点一起持久化。
4. 调整默认逻辑: 从 `peft_config` 中获取 `lora_alpha` 时若未设置则默认为 `lora_rank`, 遵循 PEFT 惯例; 且仅当 `lora_rank > 0` 时才写入。 (原实现还包含 `MegatronCheckpointManager` 的修改, 但在 review 中被删除, 因为 Megatron Bridge 已通过 `save_hf_adapter` 处理 PEFT 导出, 无需 `model_merger`。)

关键文件:

- `verl/utils/checkpoint/fsdp_checkpoint_manager.py` (模块 检查点管理; 类别 source; 类型 core-logic; 符号 `_get_lora_train_meta`, `_save_lora_train_meta`): 所有变更集中于此文件, 新增 LoRA 训练元数据持久化逻辑。

关键符号: `_get_lora_train_meta`, `_save_lora_train_meta`

关键源码片段

verl/utils/checkpoint/fsdp_checkpoint_manager.py

所有变更集中于此文件，新增 LoRA 训练元数据持久化逻辑。

```
# pylint: skip-file
# 从 unwrapped model 的 peft_config 中提取 LoRA 训练超参数
def _get_lora_train_meta(self, unwrap_model):
    peft_config = getattr(unwrap_model, "peft_config", None)
    if not peft_config:
        return None
    # peft_config 可能是 dict，取默认适配器
    if isinstance(peft_config, dict):
        peft_config = peft_config.get("default") or next(iter(peft_config.values()), None)
    if peft_config is None:
        return None

    lora_rank = int(getattr(peft_config, "r", 0) or 0)
    if lora_rank <= 0:
        return None

    # 遵循 PEFT 约定：lora_alpha 未设置时默认等于 lora_rank
    lora_alpha = int(getattr(peft_config, "lora_alpha", lora_rank) or 0)
    task_type = getattr(peft_config, "task_type", None) or "CAUSAL_LM"
    if hasattr(task_type, "value"):
        task_type = task_type.value

    return {"r": lora_rank, "lora_alpha": lora_alpha, "task_type": str(task_type)}

def _save_lora_train_meta(self, local_path: str, unwrap_model):
    lora_train_meta = self._get_lora_train_meta(unwrap_model)
    if lora_train_meta is None:
        return None

    lora_meta_path = os.path.join(local_path, "lora_train_meta.json")
    with open(lora_meta_path, "w", encoding="utf-8") as f:
        json.dump(lora_train_meta, f, ensure_ascii=False, indent=4)
    # 仅在 rank 0 打印日志，但实际写入在所有 rank 上都执行
    log_with_rank(
        f"Saved LoRA rank/alpha metadata to {os.path.abspath(lora_meta_path)}",
        rank=self.rank,
        logger=logger,
        log_only_rank_0=True,
    )
    return lora_meta_path

# 在 save_checkpoint 方法中，保存 fsdp_config 后立即调用
# （位于 torch.distributed.barrier() 之前，与 fsdp_config 保存对齐）
...
with open(fsdp_config_path, "w") as f:
    json.dump(asdict(fsdp_config), f, indent=4)
```

```
self._save_lora_train_meta(local_path, unwrap_model) # 新增：持久化 LoRA 元数据
# wait for everyone to dump to local
torch.distributed.barrier()
...
```

评论区精华

wuxibin89建议将 LoRA 元数据保存逻辑从 trainer 移入 CheckpointManager: "Better move into CheckpointManager(FSDP/Megatron) instead of exposing to trainer"。Yatogaii 接受建议，将代码迁移至 `FSDPCheckpointManager` 和 `MegatronCheckpointManager`（后移除 Megatron）。HollowMan6指出 Megatron Bridge 已通过 `save_hf_adapter` 正确导出 PEFT 适配器，质疑 `model_merger + lora_train_meta` 路径的必要性。Yatogaii 认同并移除 Megatron 部分，仅保留 FSDP 路径。gemini-code-assist[bot]提醒原逻辑可能导致 `lora_alpha` 为 0，应默认使用 `lora_rank`: "Following standard PEFT conventions, `lora_alpha` should default to `lora_rank` if not explicitly specified." 该建议被采纳到最终代码中。

- 将 LoRA 元数据保存逻辑从 trainer 移至 CheckpointManager (design): Yatogaii 接受建议，将代码从 trainer 移至 `FSDPCheckpointManager` 和 `MegatronCheckpointManager`（随后移除了 Megatron 部分）
- 移除 Megatron 端的 `lora_train_meta` 保存 (design): Yatogaii 认同，移除 `MegatronCheckpointManager` 的相关修改，仅保留 FSDP 路径。
- 确保 `lora_alpha` 默认不为 0 (correctness): 代码中已采用 `lora_alpha = int(getattr(peft_config, 'lora_alpha', lora_rank) or 0)` 实现默认值 `lora_rank`。
- 缺少自动化测试 (testing): 仅通过手动合成检查点验证合并流程。

风险与影响

- 风险：
 1. 多 rank 写入冲突风险: `_save_lora_train_meta` 在所有 rank 上执行（无 rank 0 限制），若 `local_path` 共享，多个进程同时写同一文件可能导致内容损坏。不过 PPO actor 检查点通常只在 rank 0 实际执行写入，实际风险较低。
 2. 缺少测试覆盖: 未添加自动化测试，仅通过手动合成检查点验证。合并流程变更依赖人工确认。
 3. 仅 FSDP 路径覆盖: Megatron 后端已移除，如果未来 Megatron PPO LoRA 用户需要使用 `model_merger` 合并，该路径仍会缺失元数据。但 Megatron Bridge 已有自己的 PEFT 导出路径，故风险可控。
 - 影响：用户：修复了 PPO LoRA 用户检查点合并后 `lora_alpha` 为 0 的 bug，使合并的适配器能正确生效。系统：每次 actor 检查点保存时增加一次小文件写入（约 100 字节），性能影响可忽略。团队：统一了 SFT 和 PPO 的 LoRA 元数据保存方式，降低后续维护复杂度。
 - 风险标记：多 rank 写入竞争风险，缺少测试覆盖

关联脉络

- PR #5326 [ckpt, model] fix: preserve lora_alpha in model_merger via training meta: 前序修复, 为 SFT 检查点保存 lora_train_meta.json, 本 PR 将其扩展到 PPO actor 检查点