

PR #6167 完整报告

verl-project/verl

[trainer] fix: convert numpy arrays to native types before dumping rollout JSONL

合并时间: 2026-04-27 19:02

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6167>

执行摘要

- 一句话: 修复 numpy 数组 JSON 序列化崩溃
- 推荐动作: 值得精读, 尤其是理解 numpy 序列化问题的通用解决模式: 先尝试早期类型转换, 再用 `default=str` 作为最后兜底。同时应关注 review 中指出的 `request_id` 逻辑问题, 建议在后续 PR 中修复。

功能与动机

自定义奖励函数在 `reward_extra_info` 中返回 numpy 类型的值 (如 `np.int64`) 时, JSON 序列化抛出 `TypeError: Object of type int64 is not JSON serializable`。PR body 明确说明这是对 #6062 的补充: `default=str` 会将数值转为字符串 (如 `"3"`), 而本 PR 通过早期转换让整数保持为原生类型。

实现拆解

1. 定位问题: 在 `verl/trainer/ppo/ray_trainer.py` 的 `_log_rollout_data` 方法中, `reward_extra_infos_dict` 直接传递给 `_dump_generations`, 但该字典的值可能包含 numpy 数组 (源自 `batch.non_tensor_batch`), 而 `json.dumps` 无法序列化 numpy 类型。
2. 添加边界转换: 将原 `reward_extra_infos_to_dump = reward_extra_infos_dict.copy()` 替换为字典推导式, 对每个值检查是否为 `np.ndarray` 实例, 若是则调用 `.tolist()` 转换为原生 Python 列表, 否则保持原值。
3. 保持兼容性: 新构建的 `reward_extra_infos_to_dump` 仍传递给 `_dump_generations`, 该函数内部已包含 `default=str` 后备, 两者协作确保 numpy 数据正确转换为原生类型, 其他异常对象仍能被 `default=str` 捕获。
4. 测试验证: 通过下游训练复现崩溃, 确认补丁后 JSONL 输出中整数保持为原生 `int` 而非字符串。

关键文件:

- `verl/trainer/ppo/ray_trainer.py` (模块 训练器; 类别 source; 类型 core-logic): 唯一变更文件, 核心修复位置。修改了 `_log_rollout_data` 方法中 `reward_extra_infos_to_dump` 的构建逻辑。

关键符号: `_log_rollout_data`

关键源码片段

verl/trainer/ppo/ray_trainer.py

唯一变更文件，核心修复位置。修改了 `_log_rollout_data` 方法中 `reward_extra_infos_to_dump` 的构建逻辑。

```
def _log_rollout_data(
    self, batch: DataProto, reward_extra_infos_dict: dict, timing_raw: dict, rollout_data_dir: str
):
    """Log rollout data to disk."""
    with marked_timer("dump_rollout_generations", timing_raw, color="green"):
        inputs = self.tokenizer.batch_decode(batch.batch["prompts"], skip_special_tokens=True)
        outputs = self.tokenizer.batch_decode(batch.batch["responses"], skip_special_tokens=True)
        scores = batch.batch["token_level_scores"].sum(-1).cpu().tolist()
        sample_gts = [item.non_tensor_batch.get("reward_model", {}).get("ground_truth", None)
                       for item in batch]

        # 核心修复：遍历 reward_extra_infos_dict 中的每个值，
        # 如果值是 numpy 数组 (np.ndarray)，则调用 .tolist() 转换为原生 Python 列表，
        # 这样 json.dumps 就不会因遇到 numpy 类型而崩溃。
        # 这里使用 dict comprehension 替换了原来的 .copy()。
        reward_extra_infos_to_dump = {
            k: (v.tolist() if isinstance(v, np.ndarray) else v)
            for k, v in reward_extra_infos_dict.items()
        }
        if "request_id" in batch.non_tensor_batch:
            reward_extra_infos_dict.setdefault(
                "request_id",
                batch.non_tensor_batch["request_id"].tolist(),
            )

        self._dump_generations(
            inputs=inputs,
            outputs=outputs,
            gts=sample_gts,
            scores=scores,
            reward_extra_infos_dict=reward_extra_infos_to_dump,
            dump_path=rollout_data_dir,
        )
```

评论区精华

Review 中 `gemini-code-assist[bot]` 指出了一个逻辑错误：`request_id` 在 `reward_extra_infos_to_dump` 创建之后才被添加到 `reward_extra_infos_dict`，导致 `request_id` 会丢失。但该评论未被提交者回复或修正，而仓库维护者 `wuxibin89` 已批准 PR，可能认为该问题不在此次修复范围内或已在线下处理。

- `request_id` 在转换后丢失 (correctness): 未在本次 PR 中修复，已合并可能不影响统计日志的核心功能，但建议后续修复。

风险与影响

- 风险：风险较低。变更仅涉及 `_log_rollout_data` 中一行代码的改写，添加了 `numpy` 类型判断与转换。主要风险在于：若 `reward_extra_infos_dict` 中存在非 `numpy` 数组但具有 `.tolist()` 方法的其他类型（如自定义类），`isinstance(v, np.ndarray)` 检查不会触发转换，保持原样，异常类型仍由 `default=str` 兜底。另外，review 指出的 `request_id` 丢失问题可能影响日志完整性，但未在当前 PR 中修复。
- 影响：影响范围有限，仅作用于训练时生成的 JSONL 日志文件。修复后，使用自定义奖励函数且返回 `numpy` 值的用户不会再遇到序列化崩溃，且日志中数值类型更加准确。该修复对系统整体功能无副作用。
- 风险标记：review 中发现逻辑错误未修复

关联脉络

- PR #6062 [trainer] fix: add `default=str` to `json.dumps` in `_dump_generations`: 此前置 PR 在 `json.dumps` 中添加了 `default=str` 后备，本 PR 是其补充，两者共同确保序列化正确性。