

PR #2108 完整报告

THUDM/slime

Extract append_response_tokens to Sample

合并时间: 2026-06-21 09:07

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2108>

执行摘要

- 一句话: 将 response token 追加逻辑提取至 Sample 类
- 推荐动作: 该 PR 值得精读, 展示了如何将引擎相关数据处理逻辑抽象到数据模型层 (Sample 类), 使上层 rollout 解耦。特别关注其 Tensor 化设计和 decode_int32_meta_array 的鲁棒性。对于开发新 rollout 的团队是参考范例。

功能与动机

PR #2108 旨在将 rollout 中重复的 token 追加和 top-p 数据处理逻辑集中到 Sample 数据模型中, 以简化各 rollout 实现并减少 bug 风险。原来的 `sglang_rollout.py` 和 `sglang_streaming_rollout.py` 各自维护了 `_append_rollout_top_p_token_data`、`_merge_rollout_top_p_token_data` 等函数, 每次追加 token 都需要手动处理; retool 等自定义 rollout 甚至需要直接操作 Sample 内部字段。通过提取 `append_response_tokens` 方法到 Sample 类, 各 rollout 只需调用 `sample.append_response_tokens(...)` 即可自动更新 tokens、response、log_probs、top-p 数据以及 loss_mask, 使代码更一致、可维护。

实现拆解

1. 新增 `decode_int32_meta_array` 至 `slime/utils/misc.py`: 统一解析 meta_info 中 base64 编码的 int32 数组, 返回 torch.Tensor, 处理字符串、字节、张量等多种输入格式。
2. 在 `slime/utils/types.py` 中新增模块级辅助函数: `_extract_rollout_top_p_token_data`、`_merge_rollout_top_p_token_data`、`_pad_rollout_top_p_offsets`, 这些函数操作 torch.Tensor, 由 `decode_int32_meta_array` 提供基础解码。
3. 为 Sample 类新增 `append_response_tokens` 方法: 接收 tokens、log_probs、trainable 标志、meta_info (含 top-p 和 finish_reason 等) 和 text, 内部调用上述辅助函数和 `_apply_meta_info` 更新状态, 同时支持流式场景的 `update_terminal_info` 参数。
4. 修改 `sglang_rollout.py` 和 `sglang_streaming_rollout.py`: 删除原有的 `_decode_int32_meta_array`、`_extract_rollout_top_p_token_data`、`_merge_rollout_top_p_token_data`、`_append_rollout_top_p_token_data`, 改为在新位置导入并使用 `Sample.append_response_tokens`。
5. 更新示例 rollout 代码: `examples/retool`、`examples/geo3k_vlm_multi_turn`、`examples/multi_agent`、`examples/search-r1` 中的 rollout 逻辑统一使用 `sample.append_response_tokens` 替代手动字段操作。

6. 调整 `slime/ray/rollout.py`: `_compute_top_p_kept_vocab_metrics` 兼容新的张量字段。
7. 新增测试文件 `tests/test_rollout_metrics.py`, 覆盖 `decode_int32_meta_array`、`append_response_tokens` 的合并、流式跳过、`routed_experts` 解码等场景; 修改 `tests/test_sample.py` 适应变更。

关键文件:

- `slime/utils/types.py` (模块 数据模型; 类别 source; 类型 dependency-wiring; 符号 `append_response_tokens`, `_extract_rollout_top_p_token_data`, `_merge_rollout_top_p_token_data`, `_pad_rollout_top_p_offsets`): 核心文件, 新增 `Sample.append_response_tokens` 方法及模块级辅助函数, 统一 rollout 中的 token 追加与 top-p 数据处理。
- `slime/utils/misc.py` (模块 通用工具; 类别 source; 类型 core-logic; 符号 `decode_int32_meta_array`): 新增通用工具 `decode_int32_meta_array`, 统一处理 base64 编码的 int32 数组解码, 支持多格式输入。
- `tests/test_rollout_metrics.py` (模块 测试; 类别 test; 类型 test-coverage; 符号 `_make_args`, `test_top_p_kept_vocab_metric_uses_loss_mask`, `test_top_p_kept_vocab_metric_skips_removed_samples`, `_b64_int32`): 新增测试文件, 覆盖 `decode_int32_meta_array` 和 `append_response_tokens` 的多种场景, 保证重构正确性。

关键符号: `append_response_tokens`, `decode_int32_meta_array`, `_extract_rollout_top_p_token_data`, `_merge_rollout_top_p_token_data`, `_pad_rollout_top_p_offsets`, `_to_int_list`, `_to_float_list`, `update_from_meta_info`, `_apply_meta_info`

关键源码片段

`slime/utils/types.py`

核心文件, 新增 `Sample.append_response_tokens` 方法及模块级辅助函数, 统一 rollout 中的 token 追加与 top-p 数据处理。

```
from slime.utils.misc import decode_int32_meta_array
```

```
# 元数据中可能的 top-p 键名 (兼容新旧名称)
```

```
_TOP_P_TOKEN_ID_META_KEYS = ("top_p_token_ids", "top_p_kept_token_ids")
```

```
_TOP_P_TOKEN_OFFSET_META_KEYS = ("top_p_token_offsets", "top_p_kept_token_offsets")
```

```
def _extract_rollout_top_p_token_data(
```

```
    meta_info: dict[str, Any],
```

```
    *,
```

```
    expected_num_tokens: int | None = None,
```

```
) -> tuple[torch.Tensor, torch.Tensor] | None:
```

```
    """从 meta_info 中提取 top-p token ids 和 offsets, 并进行合法性校验。
```

参数:

```

    meta_info: rollout 引擎返回的元数据字典。
    expected_num_tokens: 期望生成的 token 数，用于校验 offsets 长度。
返回:
    如果 meta_info 中不包含 top-p 键则返回 None;
    否则返回 (token_ids, offsets) 均为 int32 一维 Tensor。
异常:
    ValueError: 当键值不完整、offsets 不合法或不匹配时抛出。
"""
# 尝试多个可能的键名，兼容新旧 sglang 版本
token_ids = decode_int32_meta_array(meta_info, _TOP_P_TOKEN_ID_META_KEYS)
offsets = decode_int32_meta_array(meta_info, _TOP_P_TOKEN_OFFSET_META_KEYS)

# 两者都缺失则跳过
if token_ids is None and offsets is None:
    return None
# 只有一个缺失则报错
if token_ids is None or offsets is None:
    raise ValueError("SGLang top-p token replay must include both token ids and offsets.")

# 校验 offsets 首元素必须为 0
if offsets.numel() == 0 or int(offsets[0]) != 0:
    raise ValueError(f"SGLang top-p token offsets must start with 0, got {offsets[:1].tolist()}.")

# 校验 offsets[-1] 必须等于 token_ids 数量
if int(offsets[-1]) != token_ids.numel():
    raise ValueError(
        "SGLang top-p token ids/offsets mismatch: "
        f"offsets[-1]={int(offsets[-1])}, len(token_ids)={token_ids.numel()}."
    )

# 如果提供了预期 token 数，校验 offsets 长度是否为 expected_num_tokens + 1
if expected_num_tokens is not None and offsets.numel() != expected_num_tokens + 1:
    raise ValueError(
        "SGLang top-p token offsets length must equal generated token count + 1: "
        f"len(offsets)={offsets.numel()}, generated={expected_num_tokens}."
    )

return token_ids, offsets

```

评论区精华

该 PR 未产生 review 评论（0 review comments），因此无记录。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险： 1) 多个 rollout 实现（尤其是自定义 rollout，如 retool）需要同步修改，任何遗漏可能导致运行时错误。PR 已对所有已知使用点更新。 2) 从 list 操作切换为

torch.Tensor 操作，需要注意类型兼容性，但辅助函数 `_to_int_list/_to_float_list` 保证了向下兼容。3) 新增测试覆盖了核心路径，但仍需确保与现有 CI 中的 megatron 测试集成。

- 影响：对使用方：自定义 rollout 代码需要改用 `sample.append_response_tokens`，但简化了字段赋值。对系统：减少代码重复，统一 top-p 数据处理路径，有利于后续扩展。影响范围包括 `sglang_rollout`、`streaming_rollout` 及所有示例 rollout 文件。
- 风险标记：多入口改造需测试覆盖，字段类型变更兼容性

关联脉络

- PR #2102 Support top_p mask: 引入 top-p token 处理的核心功能，本 PR 将该功能提取到 `Sample` 类中统一管理。
- PR #2107 Fix wrongly removed RouterArgs registry: 同一分支上之前修复的 bug，与本 PR 共享分支 `zilin/bugfix`。
- PR #2005 [coding-agent-rl] Refactor coding-agent RL: turn-node TrajectoryManager + pluggable harness layer: 类似的重构模式，将分散逻辑提取到类中，可对比学习。