

PR #1968 完整报告

radixark/miles

[Bug Fix] Reduce distributed min/max metrics as extrema

合并时间: 2026-08-07 15:04

原文链接: <http://prhub.com.cn/radixark/miles/pull/1968>

执行摘要

- 一句话: 修复分布式日志 min/max 被平均聚合错误, 改为按极值归约
- 推荐动作: 值得精读。这是一个小文件但高信号度的修复: 一、它纠正了分布式训练指标聚合中一个容易被忽视的语义错误 (局部极值不能跨 rank 平均); 二、评论区对 " 后缀推断 vs 显式白名单 " 的取舍是很好的 API 设计案例, 最终选择显式策略以保证默认安全; 三、merge commit 展示了与 #1927 可变 global batch size 引入的 (sum, count) tuple 语义如何共存。阅读时可重点关注 `reduce_gathered_log_dict` 的分支结构和测试对边界 (空 gather、键不一致、未知策略名) 的覆盖。

功能与动机

PR body 明确指出: `gather_log_data` 对每个收集到的标量跨 effective DP/CP rank 做平均, 这对普通均值指标正确, 但对 rank 局部极值错误——平均局部最大值会低估全局最大值, 平均局部最小值会高估全局最小值。多轮训练中 `raw_response_length/response_length_max`、`round_number_max` 等极值指标会被系统性扭曲, 导致监控与训练曲线失真。

实现拆解

实现分四步, 全部集中在 `miles/backends/training_utils/log_utils.py` 和新增测试文件中:

1. 扩展 reducer 的归约策略: `reduce_gathered_log_dict(gathered, dp_size, reduction_by_key=None)` 新增可选的 per-key 归约策略参数。函数先处理空 `gathered` 返回 `{}`; 若调用方显式传入策略, 则校验各 rank 的 metric key 集合完全一致, 不一致时抛出 `ValueError` ("Metric keys differ across ranks"); 随后逐 key 归约——(sum, count) 二元组始终按 $\sum \text{sum} / \sum \text{count}$ 加权平均 (与 #1927 引入的可变 global batch size 语义保持一致), 普通标量按 `reduction_by_key` 选择 mean/min/max, 未指定的键默认 mean, 未知策略名抛出 `ValueError` ("Unsupported metric reduction")。这是根节点上的纯本地计算, 不新增任何分布式 collective, 通信开销不变。
2. 定义多轮指标的极值白名单: 新增模块级常量 `_MULTI_TURN_REDUCTION_BY_KEY`, 把 `log_multi_turn_data` 产出的 6 个已知极值键 (`response_length_max/min`、`wo_obs_response_length_max/min`、`round_number_max/min`) 显式标记为 max/min 归约。采用显式登记而非按 `_min/_max` 后缀推断, 是因为后缀推断覆盖面过宽, 可能误伤普通以 `_max` 结尾的均值指标 (见评论区 maocheng23 的反对意见)。

3. 打通调用链: `gather_log_data` 新增 `reduction_by_key` 参数并透传给 reducer;
`log_multi_turn_data` 在调用 `gather_log_data("multi_turn", ...)` 时传入白名单。未传参数的既有调用方行为完全不变, 保持向后兼容。
4. 测试配套: 新增 `tests/fast/backends/training_utils/test_log_utils_metric_reduction.py`, 8 个用例覆盖: 显式极值归约 (验证全局极值 900/8 而非旧的均值 650/20)、无策略时默认 mean、显式策略优先、`(sum, count)` 元组忽略归约策略、跨 rank key 不一致抛错、未知归约名抛错、空 gather 返回空 dict, 以及用 monkeypatch 验证生产调用点
`log_multi_turn_data` 确实把白名单传给 `gather_log_data`。测试位于 fast 目录, 无 GPU 依赖, 可进 CI 常规回归。
5. merge 冲突处理: 第二个 commit 合入 main 时与 #1927 (可变 global batch size) 对 `reduce_gathered_log_dict` 的改动产生冲突, 解决方式是把 min/max 归约语义折叠进 tuple-aware reducer, 保证 `(sum, count)` 元组仍按 sum/count 归约, 标量才按策略归约。

关键文件:

- `miles/backends/training_utils/log_utils.py` (模块 训练日志; 类别 source; 类型 core-logic; 符号 `reduce_gathered_log_dict`, `gather_log_data`, `log_multi_turn_data`, `_MULTI_TURN_REDUCTION_BY_KEY`): 核心修复所在: reducer 新增 per-key mean/min/max 归约策略与跨 rank key 校验, `gather_log_data` 透传策略, `log_multi_turn_data` 显式传入极值白名单 `_MULTI_TURN_REDUCTION_BY_KEY`
- `tests/fast/backends/training_utils/test_log_utils_metric_reduction.py` (模块 指标归约; 类别 test; 类型 test-coverage; 符号 `test_min_max_keys_reduce_to_global_extrema`, `test_keys_default_to_mean_without_reduction_map`, `test_explicit_reduction_selects_extrema`, `test_sum_count_tuples_ignore_reduction_map`): 8 个回归用例锁定 reducer 语义与生产接线, 覆盖极值归约、默认 mean、tuple 忽略策略、schema 校验等边界, 且位于 fast 目录无 GPU 依赖

关键符号: `reduce_gathered_log_dict`, `gather_log_data`, `log_multi_turn_data`

评论区精华

review 中围绕 " 如何确定极值指标的归约策略 " 发生了两轮实质交锋:

1. 后缀推断方案的提出与回退: Shi-Dong 建议把默认策略下沉到 reducer——"Wouldn't it be better to set a default for reduction key inside `_reduce_gathered_log_dicts`? E.g. if a metric's name ends with `_min`, then the default reduction becomes `min`." 作者随即采纳, 将 `_min/_max` 后缀推断实现进 reducer 并移除了 `log_multi_turn_data` 的接线。但 maocheng23 随即请求变更: "This process would be too broad. We may not only use the key name to determine how to reduce values."——名称推断过于宽泛, 不能仅凭 key 名决定归约语义。
2. 最终收敛为显式白名单: 作者接受该意见, 恢复 mean 作为通用 reducer 的默认值, 改为由 `log_multi_turn_data` 对已知极值键显式传入 `_MULTI_TURN_REDUCTION_BY_KEY`, 并补充相应测试。Shi-Dong 最终 APPROVED ("LGTM with minor comment")。

两轮讨论的核心张力是 " 默认安全性 vs 调用便利性 ": 显式登记更可预测、不意外改变既有指标语义, 代价是新增极值指标时若忘记登记会静默退回均值。

- 是否按 `_min/_max` 后缀自动推断归约策略 (design): 采纳 maocheng23 意见, 恢复 `mean` 作为通用 reducer 默认值, 改由 `log_multi_turn_data` 显式传入 `_MULTI_TURN_REDUCTION_BY_KEY` 白名单, 后缀推断方案回退。
- merge main 时与 `(sum,count)` tuple 归约语义的整合 (correctness): tuple 语义保持不变, 新增测试 `test_sum_count_tuples_ignore_reduction_map` 锁定该行为。

风险与影响

- 风险:
 1. 跨 rank schema 校验是新增的崩溃路径: 当显式传入 `reduction_by_key` 时 (即 `multi_turn` 路径), 若各 rank 的 `log_dict` key 不一致会直接抛 `ValueError`。此前这种不一致会被静默按 `gathered[0]` 的键归约。如果线上存在某些 rank 因数据分支缺指标的罕见情况, 此改动会暴露为日志中断, 需确认 `log_multi_turn_data` 的所有 rank 构造路径一致。
 2. 白名单手工维护风险: `_MULTI_TURN_REDUCTION_BY_KEY` 只覆盖当前 6 个键, 未来在 `log_multi_turn_data` 新增 `_min/_max` 指标时若忘记登记, 又会静默退化为均值, 回到本 PR 修复的失真问题。
 3. 影响面受控: 改动仅在 DP 源根节点的归约阶段, 不涉及前向 / 反向, 不新增 `collective`, 不改变通信量; `mean` 仍为默认, 既有调用方输出不变。
 4. tuple 语义已覆盖: 测试 `test_sum_count_tuples_ignore_reduction_map` 锁定了与 #1927 的 `(sum, count)` 元组共存行为, merge commit 已解决冲突。- 影响: 用户 / 训练侧: 所有依赖 `gather_log_data` 做分布式日志聚合的任务, 多轮指标中的 `response_length_max/min`、`round_number_max/min` 等将首次正确反映全局极值, 监控面板 (wandb/tensorboard) 上的极值曲线不再系统性失真。尤其 `response_length_max` 被低估可能掩盖 rollout 截断问题, 修复后更易暴露真实上限。系统侧: 零新增通信, 只读根节点本地归约逻辑, 性能无影响。团队侧: 为后续新增分布式指标确立了 "显式归约策略 + schema 校验" 的模式, 提升了日志管线的可维护性与可调试性。- 风险标记: 跨 rank 指标 schema 校验新增崩溃路径, 极值白名单需随新指标手工维护, 日志链路变更不影响训练数值路径

关联脉络

- PR #1927 [feat] Support training with variable global batch size: 改动同一函数 `reduce_gathered_log_dict` 引入 `(sum, count)` 元组归约语义, 本 PR 的 merge commit 专门解决与该 PR 的冲突, 并把 `min/max` 策略折进 tuple-aware reducer。
- PR #2134 fix: skip the --dump-details processor dump when it cannot serialise: 同为 rollout 日志 / 观测链路 (`miles/rollout/data_source.py`、`dashboard/observability`) 的正确性修复, 与本 PR 共同构成日志管线健壮性改进方向。