

PR #1930 完整报告

THUDM/slime

[1/N] Support training with variable global batch size

合并时间: 2026-05-21 16:45

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1930>

执行摘要

- 一句话: 支持训练过程中动态变化的全局 batch size
- 推荐动作: 值得精读, 尤其关注 loss scaling 和 LR increment 的传递链路; 对于参与训练核心开发的同学, 理解 global_batch_sizes 的引入意图对后续合入 uneven-DP 特性很有帮助。

功能与动机

PR 标题标明 [1/N], 说明这是支持 variable global batch size 的系列变更的第一步。从代码注释可知, 此变更旨在打破 '每个 DP rank 持有相同数量的样本' 的假设, 为 uneven-DP partition 做准备 (见 data.py 中 gather_log_data 的新注释)。

实现拆解

1. dp_schedule.py: build_dp_schedule 返回值新增第四项 global_batch_sizes (list[int]), 每个 step 记录实际样本总数, 不再是固定的 global_batch_size。
2. rollout.py: _split_train_data_by_dp 中将原来 rollout_data['dynamic_global_batch_size'] 替换为 rollout_data['global_batch_sizes'], 内容来自 schedule 的新返回值。
3. model.py: train_one_step 新增参数 step_global_batch_size, 并传至 loss_function; train 函数新增参数 global_batch_sizes (与 num_microbatches 等长), 并在循环中逐 step 传递。
4. loss.py: loss_function 新增 step_global_batch_size 参数, 替代原来从 batch 中读取 dynamic_global_batch_size 或回退到 args.global_batch_size 的逻辑, 用于 loss 缩放。LR 调度器 increment 也改为直接使用该参数。
5. data.py: gather_log_data 支持 (sum, count) 三元组格式, 实现加权平均, 便于后续每个 DP rank 持有不同样本数时正确聚合; log_rollout_data 中跳过的键由 dynamic_global_batch_size 改为 global_batch_sizes。
6. actor.py: 适配 train 函数签名变更, 传递 global_batch_sizes。
7. test_dp_schedule.py: assert_invariants 新增对 global_batch_sizes 的校验; 所有测试用例解包新返回值, 并在断言中传递 global_batch_sizes 和 global_batch_size。

关键文件:

- slime/backends/megatron_utils/data.py (模块 数据加载; 类别 source; 类型 core-logic) : 核心变更: gather_log_data 支持加权平均, log_rollout_data 跳过键名变更, 移除 dynamic_global_batch_size 注入。
- tests/test_dp_schedule.py (模块 调度器; 类别 test; 类型 test-coverage; 符号 assert_invariants) : 测试覆盖新增的 global_batch_sizes 返回值, assert_invariants 增加对应校验。
- slime/backends/megatron_utils/model.py (模块 训练引擎; 类别 source; 类型 data-contract) : train_one_step 新增 step_global_batch_size 参数, 传递至 loss_function 和 LR scheduler; train 新增 global_batch_sizes 参数。
- slime/utils/dp_schedule.py (模块 调度器; 类别 source; 类型 core-logic) : build_dp_schedule 返回新增 global_batch_sizes 列表, 为训练侧提供每步真实样本数。
- slime/backends/megatron_utils/loss.py (模块 损失函数; 类别 source; 类型 core-logic) : loss_function 新增 step_global_batch_size 参数, 替代 dynamic_global_batch_size, 用于 loss 缩放。
- slime/ray/rollout.py (模块 Rollout; 类别 source; 类型 core-logic) : 替换 dynamic_global_batch_size 为 global_batch_sizes, 解包 build_dp_schedule 新返回值。
- slime/backends/megatron_utils/actor.py (模块 Actor; 类别 source; 类型 core-logic) : 适配 train 函数新增的 global_batch_sizes 参数。

关键符号: gather_log_data, log_rollout_data, build_dp_schedule, train_one_step, train, loss_function, _split_train_data_by_dp

关键源码片段

slime/backends/megatron_utils/data.py

核心变更: gather_log_data 支持加权平均, log_rollout_data 跳过键名变更, 移除 dynamic_global_batch_size 注入。

slime/backends/megatron_utils/data.py — gather_log_data 新实现

```
def gather_log_data(
    metric_name: str,
    args: Namespace,
    rollout_id: int,
    log_dict: dict[str, "float | tuple[float, float]"],
) -> dict[str, float] | None:
    """
    跨 DP rank 归约指标。

    log_dict 中的每个值可以是:
    * 标量 —— 简单取平均 (旧路径, 仅在每 rank 样本数相同时正确) ;
    * (sum, count) 元组 —— 作加权平均  $\Sigma \text{sum} / \Sigma \text{count}$ ,
      为 uneven-DP partition 做准备。
    """
    if mpu.get_data_parallel_rank(with_context_parallel=True) == 0:
        dp_size = mpu.get_data_parallel_world_size(with_context_parallel=True)
```

```

gathered_log_dict = [None] * dp_size
dist.gather_object(
    log_dict,
    gathered_log_dict,
    dst=mpu.get_data_parallel_src_rank(with_context_parallel=True),
    group=mpu.get_data_parallel_group_gloo(with_context_parallel=True),
)
reduced_log_dict: dict[str, float] = {}
for key in log_dict:
    values = [d[key] for d in gathered_log_dict]
    first = values[0]
    if isinstance(first, tuple) and len(first) == 2:
        # 加权平均路径
        total_sum = sum(v[0] for v in values)
        total_count = sum(v[1] for v in values)
        reduced = total_sum / total_count if total_count != 0 else 0.0
    else:
        # 旧标量路径: 直接平均
        reduced = sum(values) / dp_size
    reduced_log_dict[f"{metric_name}/{key}"] = reduced
logger.info(f"{metric_name} {rollout_id}: {reduced_log_dict}")
# ... 后续计算 step 并 log
return reduced_log_dict
return None

```

tests/test_dp_schedule.py

测试覆盖新增的 global_batch_sizes 返回值, assert_invariants 增加对应校验。

tests/test_dp_schedule.py — assert_invariants 新逻辑

```

def assert_invariants(
    partitions,
    micro_batch_indices,
    num_microbatches,
    *,
    dp_size,
    total_lengths,
    max_per_bin=None,
    global_batch_sizes=None,
    global_batch_size=None,
):
    """检查 dp_schedule.py 文档中列出的不变性约定。"""
    # ... 原有 partition/mbs 不变性检查 ...

    if global_batch_sizes is not None:
        # 每个 step 的 global_batch_sizes 必须与 num_microbatches 等长
        assert len(global_batch_sizes) == len(num_microbatches), (
            f"global_batch_sizes/num_microbatches length mismatch: "
            f"{len(global_batch_sizes)} vs {len(num_microbatches)}"

```

```
)
if global_batch_size is not None:
    # 在等样本 case 中，每个 step 的 gbs 必须等于传入的 global_batch_size
    for s, gbs in enumerate(global_batch_sizes):
        assert gbs == global_batch_size, f"step {s} gbs {gbs} != {global_batch_size}"

# ... 原有 max_per_bin 检查 ...
```

评论区精华

本 PR 无 review 评论；唯一 commit message 简单描述功能，无设计讨论。

- 暂无高价值评论线程

风险与影响

- 风险：训练损失缩放风险：若 rollout 侧未能正确提供 global_batch_sizes（如未适配新版本 schedule），会导致 loss 缩放错误，影响训练收敛。LR 调度风险：LR increment 源从 dynamic_global_batch_size 改为显式参数，若接力代码未更新，可能导致 LR 更新步长与实际样本数不匹配。向后兼容：此 PR 移除了对 dynamic_global_batch_size 的全部依赖，若存在外部 hook 或自定义迭代器依赖该键，将失效。
- 影响：影响范围：修改了训练核心路径（loss 计算、LR 调度）、DP 调度、Rollout 数据传输及日志聚合，涉及 7 个文件。对使用者而言，启用 variable batch size 时需确保 rollout 侧调用新 API，原有 static batch size 路径在测试中仍通过（等样本 case global_batch_sizes 与 global_batch_size 一致）。对团队开发，此 PR 为后续 uneven-DP partition 铺平了数据通路，是架构演进的关键步骤。
- 风险标记：核心训练路径变更，损失缩放依赖新参数，LR 调度器增量源改变，移除 dynamic_global_batch_size 可能影响外部 hook

关联脉络

- PR #1926 Move micro-batch scheduling from training side to rollout side: 本 PR 使用 #1926 的调度产出，将 global_batch_sizes 传递到训练侧，是同一技术路线的延续。