

PR #30655 完整报告

sgl-project/sglang

Fix rel_diff being nan for bitwise-identical tensors

合并时间: 2026-07-09 20:16

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/30655>

执行摘要

- 一句话: 修复位相同张量 rel_diff 为 NaN 问题
- 推荐动作: 值得合并。这是一个明确的正确性修复, 逻辑清晰且测试覆盖充分。

功能与动机

当两个张量位相同时, calc_rel_diff 计算 0/0 得到 NaN, 而 NaN 与任何值的比较结果均为 False, 导致确定性模式下的精确匹配检查 (如 'rel <= 0') 在完美匹配的张量上误报失败, 包括合法的全零张量 (如饥饿 MoE 专家的梯度)。

实现拆解

1. 修改 compute_diff 函数: 在 python/sglang/srt/debug_utils/comparator/tensor_comparator/comparator.py 中, 将计算 rel_diff 的行从无条件调用 calc_rel_diff 改为先检查 max_abs_diff 是否为零, 若为零则直接返回 0.0, 否则调用原函数。
2. 添加回归测试: 在 test/registered/debug_utils/comparator/tensor_comparator/test_comparator.py 中新增 test_bitwise_predicate 方法, 验证 'rel <= 0' 对克隆张量返回 passed=True, 对符号翻转的近零对返回 passed=False。

关键文件:

- python/sglang/srt/debug_utils/comparator/tensor_comparator/comparator.py (模块 比较器; 类别 source; 类型 core-logic): 核心修复位置, 修改了 compute_diff 函数中 rel_diff 的计算逻辑。
- test/registered/debug_utils/comparator/tensor_comparator/test_comparator.py (模块 测试; 类别 test; 类型 test-coverage; 符号 test_bitwise_predicate): 新增回归测试, 验证修复的正确性。

关键符号: compute_diff

关键源码片段

python/sglang/srt/debug_utils/comparator/tensor_comparator/comparator.py

核心修复位置, 修改了 compute_diff 函数中 rel_diff 的计算逻辑。

```
# compute_diff 函数中关键变更 (comparator.py)
```

```

raw_abs_diff = (x_target - x_baseline).abs()
max_diff_coord = argmax_coord(raw_abs_diff)

max_abs_diff = raw_abs_diff.max().item()
# 当 max_abs_diff == 0.0 时，两个张量位相同，rel_diff 应为 0.0
# 直接计算 calc_rel_diff 会得到 0/0 = NaN，导致谓词比较失败
rel_diff = (
    0.0 if max_abs_diff == 0.0 else calc_rel_diff(x_target, x_baseline).item()
)
mean_abs_diff = raw_abs_diff.mean().item()

```

test/registered/debug_utils/comparator/tensor_comparator/test_comparator.py

新增回归测试，验证修复的正确性。

```

# test_bitwise_predicate 方法 (test_comparator.py)
def test_bitwise_predicate(self) -> None:
    """'rel <= 0' passes only for bitwise-identical tensors."""
    ident = torch.randn(5, 5)
    # 克隆张量位相同，应返回 passed=True
    assert (
        compute_diff(
            x_baseline=ident, x_target=ident.clone(), predicate="rel <= 0"
        ).passed
        is True
    )
    x, y = self._near_zero_pair()
    # 符号翻转的近零对，位不同，应返回 passed=False
    assert (
        compute_diff(x_baseline=x, x_target=y, predicate="rel <= 0").passed is False
    )

```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅影响 `max_abs_diff == 0.0` 的分支，该分支原本返回 NaN，现在返回 0.0，行为更符合预期。现有测试（如空张量短路）不受影响。
- 影响：影响范围限于 comparator 工具的用户，特别是使用 `'rel <= 0'` 谓词进行精确匹配检查的场景。修复后，位相同张量（包括全零张量）将正确通过断言，避免误报。
- 风险标记：暂无

关联脉络

- PR #30656 Cap diagnostic detail computation for failing tensors: 修改了同一 comparator 模块, 涉及 compute_diff 的调用链和性能优化。
- PR #30657 Support grad injection and step override in the dumper's model dump: 涉及同一 debug_utils 模块, 可能依赖 comparator 的正确行为。