PR #35805 完整报告

sgl-project/sglang

[CPU] Fix weight missing issue in fused_input_proj_cpu for GPTQ INT4 for Qwen 3.5

合并时间：2026-08-31 10:14

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35805

执行摘要

- 一句话：修复 Qwen3.5 CPU GPTQ INT4 权重缺失
- 推荐动作：值得快速阅读：改动虽小，但揭示了 PyTorch 量化模块与 nn.Module.__getattr__的隐蔽交互，是“惰性求值中避免属性递归”的典型范例。关注点：(1) LazyValue 惰性判定与 and 短路的配合；(2) 直接检查 _parameters 以绕过属性兜底的写法及其代价（依赖内部契约）。建议后续为 CPU + GPTQ INT4 场景补充单元测试，防止该路径再次回归。

功能与动机

PR body 明确指出根因：For GPTQ Int4 Qwen3.5, the projection modules do not register a plain weight parameter, so accessing .weight inside LazyValue can recurse through getattr; checking _parameters directly avoids triggering attribute fallback and is equivalent for modules that do register weight. 即 GPTQ INT4 量化后的投影模块未把权重注册进 nn.Module._parameters，LazyValue 惰性求值中直接访问 .weight.dtype 会走属性兜底并递归，最终表现为权重缺失、模型无法在 Intel CPU 上正常加载。

实现拆解

1. 根因定位：qwen3_5.py 的 __init__中 _fused_input_proj_cpu_enabled = LazyValue(lambda: ...) 原写法直接访问 self.in_proj_qkvz.weight.dtype 与 self.in_proj_ba.weight.dtype。对 GPTQ INT4 模块，weight 未注册进 _parameters，访问 .weight 会触发 nn.Module.__getattr__ 兜底并可能递归，导致求值失败。
2. 修复方式：在 and 链最前插入 self.in_proj_qkvz._parameters.get("weight") is not None 与 self.in_proj_ba._parameters.get("weight") is not None 两个前置条件，并把 dtype 判断改为 self.in_proj_qkvz._parameters["weight"].dtype。利用 and 短路求值，后续 use_intel_amx_backend(...) 与 .weight.size(0) 访问时参数必然已存在，不会再落入 __getattr__ 递归；对正常注册 weight 的模块，_parameters["weight"] 与 .weight 完全等价。
3. 配套与验证：本 PR 未新增任何测试文件，也未改文档或配置；CI 中 PR Test Extra 与 AMD ROCm 7.2 任务一度失败，mingfeima 两次 /rerun-failed-ci 后通过，最终以空文本 APPROVED 合并。

关键文件：

- python/sglang/srt/models/qwen3_5.py（模块 模型层；类别 source；类型 bugfix；符号 _fused_input_proj_cpu_enabled）：唯一变更文件：修复 Qwen 3.5 GPTQ INT4 下 fused_input_proj CPU 启用判定因访问 .weight 触发 __getattr__递归而报权重缺失的问题。

关键符号：init（Qwen 3.5，含 _fused_input_proj_cpu_enabled LazyValue 判定），_fused_input_proj_cpu_enabled

关键源码片段

python/sglang/srt/models/qwen3_5.py

唯一变更文件：修复 Qwen 3.5 GPTQ INT4 下 fused_input_proj CPU 启用判定因访问 .weight 触发 __getattr__递归而报权重缺失的问题。

```
# 惰性判定：是否在 Intel CPU 上启用 fused input projection（AMX 后端）。
# 关键点：GPTQ INT4 量化的 Qwen 3.5 中，in_proj_qkvz / in_proj_ba 不会把权重
# 注册进 nn.Module 的 _parameters，直接访问 .weight 会触发 __getattr__ 兜底并可能递归，
# 因此需先查 _parameters 字典；对普通模块而言该写法与 .weight 访问语义等价。
self._fused_input_proj_cpu_enabled = LazyValue(
    lambda: (
        _is_cpu
        # 先确认参数真实存在，避免后续 .weight 访问落入 __getattr__ 递归路径
        and self.in_proj_qkvz._parameters.get("weight") is not None
        and self.in_proj_ba._parameters.get("weight") is not None
        # 直接从 _parameters 读取 dtype，等价于原 self.in_proj_qkvz.weight.dtype
        and self.in_proj_qkvz._parameters["weight"].dtype == torch.bfloat16
        and self.in_proj_ba._parameters["weight"].dtype == torch.bfloat16
        # 融合路径要求两个投影均无独立 bias
        and self.in_proj_qkvz.bias is None
        and self.in_proj_ba.bias is None
        # 仅在 Intel AMX 后端可用时启用
        and use_intel_amx_backend(self.in_proj_qkvz)
        and use_intel_amx_backend(self.in_proj_ba)
        # 权重第一维（输出维度）需满足 32 对齐的 AMX 约束
        and (
            self.in_proj_qkvz.weight.size(0) % 32 == 0
            and self.in_proj_ba.weight.size(0) % 32 == 0
        )
    )
)
```

评论区精华

本 PR 没有任何 review 评论（review_comments_count = 0），唯一的审核记录是 mingfeima 的空文本 APPROVED。技术讨论集中在 PR body 中作者对根因的说明：GPTQ INT4 下投影模块不注册普通 weight 参数，.weight 访问会递归触发 __getattr__，而检查 _parameters 是等价且安全的写法。另外 mingfeima 在 CI 中两次发起 /rerun-failed-ci（PR Test Extra 与 AMD ROCm 7.2 任务曾失败），说明该改动在跨平台 CI 上经历过重跑后才通过。

- GPTQ INT4 下 .weight 访问触发 __getattr__递归 (correctness): 改为 _parameters.get("weight") is not None 前置判断并直接读 _parameters["weight"].dtype，利用 and 短路保证后续 .weight.size(0) 访问安全；讨论无未解决疑虑。
- CI Extra 与 AMD ROCm 任务失败及重跑 (other): 重跑后通过，mingfeima 以空文本 APPROVED 合并；失败与本改动的因果关系未在评论区说明。

风险与影响

- 风险：正确性 / 兼容性风险：直接读写 _parameters 属于 PyTorch 内部约定而非公开 API，若未来 Qwen 3.5 的投影模块改用 buffer 或自定义属性持有权重，_parameters.get("weight") 会返回 None，从而静默禁用 fused 路径，退化为逐模块计算——正确性不受影响，但 CPU 推理性能可能回退。回归风险：PR 未附带任何单元测试，修复依赖 Intel CPU 既有 CI 的端到端覆盖，缺少针对该场景的专项回归保障。平台风险：改动只在 _is_cpu 且 AMX 后端可用的分支中生效，GPU、AMD、XPU 等平台与其它模型完全不受影响；CI 中 AMD ROCm 7.2 与 Extra 任务曾失败，与改动本身的因果关系不明（重跑后通过）。
- 影响：用户侧：解锁 Qwen 3.5 (GPTQ INT4) 在 Intel CPU（AMX bfloat16 后端）上的正常加载与推理，此前会因权重缺失报错。系统侧：改动位于模型构建期的惰性开关，不改变任何 kernel 与 forward 行为，不影响非 CPU 路径。团队侧：为 Intel CPU 量化推理路径补上一个关键缺口，但缺少针对该场景的回归测试，建议后续补充单元测试防止再次回归。
- 风险标记：缺少测试覆盖，依赖 PyTorch 内部 _parameters 契约，跨平台 CI 曾失败

关联脉络

- PR #36975 config: the lazy imports that buy nothing become eager: 同属“惰性机制”治理：本 PR 修复的是 LazyValue 惰性求值内属性访问递归，36975 清理的是惰性导入的无效开销，二者都在审视 SGLang 中延迟求值模式的正确性与代价。
- PR #35244 [Fix] Transformers-fallback (GPT-NeoX) + KV pool config (DeepSeek-VL2): 同属模型加载与量化配置链路的 bugfix，且同样带 intel 标签，反映 Intel CPU 路径上模型加载正确性修复的持续投入。
- PR #36814 xpu: move prefill-only model tests to the nightly-xpu-1-gpu grid: Intel/XPU 平台测试基建调整，本类 CPU 修复的回归验证依赖该平台 CI 网格的稳定运行。