PR #25794 完整报告

sgl-project/sglang

Fix Gemma3 ModelOpt kv-scale loading

合并时间：2026-06-10 08:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25794

执行摘要

- 一句话：修复 Gemma3 ModelOpt KV-scale 加载顺序 bug
- 推荐动作：该 PR 值得精读，尤其是关注模型加载器中权重名称映射顺序的设计模式。它提供了一个清晰的示例：当存在多种名称变换规则时，需要仔细考虑执行顺序以避免冲突。

功能与动机

修复 Issue #25792：加载 NVIDIA ModelOpt Gemma-3 FP8/NVFP4 检查点（如 nvidia/gemma-3-12b-it-NVFP4）时，因 k_proj.k_scale / v_proj.v_scale 名称被 stacked_params_mapping 错误重写为 qkv_proj.k_scale，而 QKVParallelLinear 未注册该参数，导致 KeyError。

实现拆解

在 python/sglang/srt/models/gemma3_causal.py 的 load_weights 方法中，在 stacked_params_mapping 处理之前，先调用 maybe_remap_kv_scale_name 对当前权重名称进行映射。若映射后名称改变（即命中 KV scale），则直接通过 weight_loader 加载并 continue，跳过后续的融合映射处理；若名称未改变，则继续原有逻辑。该改动确保 KV scale 名称不会被错误地融合到 QKV 路径中。

同时，在 PR 的第二个 commit 中，根据 review 建议移除了最初包含的单元测试文件 test/registered/unit/models/test_gemma3_weight_loading.py，因为该修复可以被端到端加载测试覆盖。

关键文件：

- python/sglang/srt/models/gemma3_causal.py（模块 模型加载；类别 source；类型 data-contract；符号 load_weights）：核心修复文件：调整 load_weights 方法中 KV scale 重命名与 QKV 融合映射的执行顺序，确保 ModelOpt NVFP4/FP8 检查点正确加载。

关键符号：load_weights

关键源码片段

python/sglang/srt/models/gemma3_causal.py

核心修复文件：调整 load_weights 方法中 KV scale 重命名与 QKV 融合映射的执行顺序，确保 ModelOpt NVFP4/FP8 检查点正确加载。

```
def load_weights(self, weights: Iterable[Tuple[str, torch.Tensor]]):
    stacked_params_mapping = [
        # (param_name, shard_name, shard_id)
        ("qkv_proj", "q_proj", "q"),
        ("qkv_proj", "k_proj", "k"),
        ("qkv_proj", "v_proj", "v"),
        ("gate_up_proj", "gate_proj", 0),
        ("gate_up_proj", "up_proj", 1),
    ]
    params_dict = dict(self.named_parameters())
    loaded_params: Set[str] = set()
    for name, loaded_weight in weights:
        # [Fix] 先处理 KV scale 映射，避免被下面的 stacked_params_mapping 错误重写
        remapped_name = maybe_remap_kv_scale_name(name, params_dict)
        if remapped_name is None:
            continue # scale 参数不存在时跳过
        if remapped_name != name:
            # 命中 KV scale: 直接加载并跳过后续的融合映射
            param = params_dict[remapped_name]
            weight_loader = getattr(param, "weight_loader", default_weight_loader)
            weight_loader(param, loaded_weight)
            loaded_params.add(remapped_name)
            continue

        for param_name, shard_name, shard_id in stacked_params_mapping:
            if shard_name not in name:
                continue
            name = name.replace(shard_name, param_name)
            if name.endswith(".bias") and name not in params_dict:
                continue
            param = params_dict[name]
            weight_loader = param.weight_loader
            weight_loader(param, loaded_weight, shard_id)
            break
        else:
            # 未命中融合映射时的 fallback
            if "lm_head.weight" in name:
                continue
            if name.endswith(".bias") and name not in params_dict:
                continue
            # 原代码此处才调用 maybe_remap_kv_scale_name，现已提前
            name = maybe_remap_kv_scale_name(name, params_dict)
            if name is None:
                continue
            param = params_dict[name]
            weight_loader = getattr(param, "weight_loader", default_weight_loader)
            weight_loader(param, loaded_weight)
        loaded_params.add(name)
```

评论区精华

Reviewer kpham-sgl 最初要求添加 GSM8K 等基准测试以验证精度无下降。贡献者提供了 GSM8K、MMLU、GPQA Diamond 的对比数据，显示 FP8/NVFP4 与 BF16 精度差异在统计误差内。此外，kpham-sgl 建议移除单元测试文件，认为对于这个小改动而言，单元测试过于冗余，贡献者随后移除了测试。

- 精度验证 (correctness): 无显著精度下降，验证了修复正确性。
- 单元测试必要性 (testing): 移除了新增的单元测试文件，因为该修复可以被端到端加载测试覆盖。

风险与影响

- 风险：风险极低。改动仅限于 load_weights 方法的控制流顺序，不影响模型前向计算。唯一风险是若其他模型也使用类似的 *maybe_remap_kv_scale_name* 且可能被 stacked_params_mapping 错误接管，但该 PR 为 Gemma3 专用修复，不会影响其他模型。
- 影响：直接影响：使用 NVIDIA ModelOpt 量化 Gemma-3 检查点的用户将能成功加载模型并完成推理。间接影响：无，因为改动仅限于 Gemma3 模型加载路径。
- 风险标记：低风险

关联脉络

- PR #25792 [Bug] Gemma-3 ModelOpt FP8/NVFP4 fails to load with qkv_proj.k_scale KeyError: 此 PR 直接修复该 Issue 报告的问题。