PR #26038 完整报告

sgl-project/sglang

[NPU] fix model ERNIE-4.5-21B-A3B-PT bias need 1D error

合并时间：2026-05-28 16:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26038

执行摘要

- 一句话：修复 ERNIE-4.5 在 NPU 上的 correction_bias 维度错误
- 推荐动作：该 PR 以最小改动修复了 NPU 上的阻塞问题，值得快速合并。但建议后续将 NPU 特定逻辑收敛到公共组件（如 TopK 层或 NPU 后端），避免模型定义中重复硬件判断。RoPE 风格的修复为重要安全措施，已通过 review 确认。

功能与动机

在 NPU 上启动 ERNIE-4.5-21B-A3B-PT 模型时，correction_bias 的维度为 (1, moe_num_experts) 的 2D 张量，但 NPU 的 TopK 实现要求 1D 输入，导致报错。PR body 中附带了报错截图。

实现拆解

1. 修正 correction_bias 维度（ernie4.py）：在 MoEGate.__init__ 中引入 _is_npu 标志，当检测到 NPU 环境时，对 e_score_correction_bias 执行 squeeze(0) 将其从 2D 转为 1D，再传递给 TopK 层。
2. 修复 RoPE 风格传递（llama.py）：在 LlamaAttention.forward_prepare_npu 中，调用 split_qkv_rmsnorm_rope 时增加 is_neox_style=self.rotary_emb.is_neox_style 参数，避免硬编码 False 导致的回归问题（此修复由 review 建议最终采纳）。
3. 导入变更：在 ernie4.py 中新增 is_npu 函数的导入，并添加模块级布尔变量 _is_npu。
4. TopK 层兼容性调整（topk.py）：在 topk.py 的 scoring_func_impl 中增加维度检查，当 correction_bias 维度与 scores 维度不一致时自动 unsqueeze(0)，保证无论传入 1D 还是 2D 都能正确 broadcast。但 review 指出 ERNIE 模型不会执行到该 native 路径，此修改可能多余。

关键文件：

- python/sglang/srt/models/ernie4.py（模块 模型定义；类别 source；类型 data-contract；符号 MoEGate, _is_npu）：核心修复文件：新增 is_npu 导入和模块级 _is_npu 变量，在 MoEGate.__init__ 中根据硬件环境对 correction_bias 做 squeeze 降维，解决 NPU 上的维度不匹配问题。
- python/sglang/srt/models/llama.py（模块 模型定义；类别 source；类型 data-contract；符号 LlamaAttention.forward_prepare_npu）：修复 RoPE 风格传递：在 forward_prepare_npu 中增加 is_neox_style=self.rotary_emb.is_neox_style 参数，避免

硬编码导致 Llama 模型退化。此修改由 review 强制要求。

- python/sglang/srt/layers/moe/topk.py（模块 MoE 层；类别 source；类型 data-contract；符号 scoring_func_impl）：在 topk 的 scoring_func_impl 中增加维度兼容性检查，当 correction_bias 维度与 scores 不一致时自动 unsqueeze。但 review 指出 ERNIE 模型走 NPU 路径不会执行到此 native 代码，该修改存疑。

关键符号：MoEGate.init, LlamaAttention.forward_prepare_npu, scoring_func_impl

关键源码片段

python/sglang/srt/models/ernie4.py

核心修复文件：新增 is_npu 导入和模块级 _is_npu 变量，在 MoEGate.__init__ 中根据硬件环境对 correction_bias 做 squeeze 降维，解决 NPU 上的维度不匹配问题。

```
# python/sglang/srt/models/ernie4.py
from sglang.srt.utils import add_prefix, is_npu, make_layers

# 模块级变量：缓存 NPU 环境检测结果，避免重复调用
_is_npu = is_npu()

class MoEGate(nn.Module):
    def __init__(self, config, prefix: str = ""):
        super().__init__()
        self.weight = nn.Parameter(
            torch.empty((config.moe_num_experts, config.hidden_size))
        )
        # correction_bias 初始化为 2D 形状 (1, moe_num_experts)
        self.e_score_correction_bias = nn.Parameter(
            torch.empty((1, config.moe_num_experts))
        )

class Ernie4Moe(nn.Module):
    def __init__(self, config, layer_id, quant_config=None, prefix=""):
        # ... 其他初始化 ...
        self.gate = MoEGate(config=config, prefix=add_prefix("gate", prefix))

        # 重点：根据硬件环境调整 correction_bias 的维度
        correction_bias = self.gate.e_score_correction_bias
        # NPU 上的 TopK 实现要求 bias 为 1D 张量
        if _is_npu:
            correction_bias = correction_bias.squeeze(0) # (1, E) -> (E,)
        self.topk = TopK(
            top_k=config.moe_k,
            layer_id=layer_id,
            renormalize=True,
            use_grouped_topk=False,
            correction_bias=correction_bias, # 传入调整后的 1D 或 2D 张量
        )
```

python/sglang/srt/models/llama.py

修复 RoPE 风格传递：在 forward_prepare_npu 中增加 is_neox_style=self.rotary_emb.is_neox_style 参数，避免硬编码导致 Llama 模型退化。此修改由 review 强制要求。

```
# python/sglang/srt/models/llama.py
class LlamaAttention(nn.Module):
    def forward_prepare_npu(self, positions, hidden_states, forward_batch):
        qkv, _ = self.qkv_proj(hidden_states)
        if self.attn.layer_id == self.start_layer:
            self.rotary_emb.get_cos_sin_with_position(positions)
        q, k, v = split_qkv_rmsnorm_rope(
            qkv,
            self.rotary_emb.position_sin,
            self.rotary_emb.position_cos,
            self.q_size,
            self.kv_size,
            self.head_dim,
            # 重要：必须动态传递 is_neox_style，避免硬编码导致 Llama 模型 RoPE 错误
            is_neox_style=self.rotary_emb.is_neox_style,
        )
        return q, k, v
```

评论区精华

1. RoPE 风格硬编码回归：gemini-code-assist[bot] 指出在 llama.py 中硬编码 is_neox_style=False 会破坏 Llama 系列模型的 RoPE 计算，建议改为 self.rotary_emb.is_neox_style 动态获取。最终作者采纳了该建议。
2. 硬件特定逻辑的放置位置：gemini-code-assist[bot] 建议将 NPU 维度调整封装在 TopK 层或 NPU 后端实现中，而非模型定义中直接出现硬件判断，以提高可维护性。
3. TopK 层修改的争议：Hexq0210 质疑为何修改 topk.py，因为 ERNIE 模型不会执行到 native 路径，该修改可能引入冗余逻辑。
4. 注释要求：Hexq0210 要求对 NPU 分支添加注释解释原因，作者已补充。
5. PEP 8 风格：gemini-code-assist[bot] 指出赋值运算符两侧应加空格。

- RoPE 风格硬编码导致 Llama 模型回归 (correctness): 作者采纳建议，改为 is_neox_style=self.rotary_emb.is_neox_style。
- 硬件特定逻辑是否应封装在模型定义中 (design): 未明确采纳，但作者已添加注释解释原因。后续重构时可考虑。
- TopK 层修改的有效性 (correctness): 未明确回应，但修改已保留。可能为其他模型提供兼容性保障。
- PEP 8 风格问题 (style): 未明确修复，但为小问题。
- 添加注释解释 NPU 分支逻辑 (documentation): 作者已添加注释 # npu only supports 1D, but current correction_bias is 2D。

风险与影响

- 风险：
  1. 回归风险：若 _is_npu 检测不准确（如环境变量未正确设置），可能导致非 NPU 环境下意外执行 NPU 路径。
  2. TopK 层修改：topk.py 的维度检查虽然无害，但因 ERNIE 不使用该路径，可能掩盖其他模型的维度问题，且增加了维护负担。
  3. RoPE 风格修正：is_neox_style 的动态获取已由 review 确认，风险较低。
  4. 缺少测试：未提供 NPU 上的单元测试，回归风险依赖后续 CI。
- 影响：
  - 用户：ERNIE-4.5-21B-A3B-PT 模型可在 NPU 上正常运行，correction_bias 维度错误消失。
  - 系统：仅影响 NPU 运行路径，对其他硬件无影响。
  - 团队：引入模块级硬件判断变量 _is_npu，后续 NPU 特定逻辑可复用，但需注意维护多个模型中的硬件判断逻辑分散问题。
  - 影响程度：中等，修复了阻塞性错误，但改动范围小，风险可控。
  - 风险标记：NPU 环境检测依赖，TopK 层修改争议，缺少测试覆盖

关联脉络

- 暂无明显关联 PR