PR #29042 完整报告

sgl-project/sglang

[NPU] Fix the DeepSeek-V2-Coder model accuracy issue

合并时间：2026-06-25 09:14

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29042

执行摘要

- 一句话：修复 NPU 上 DeepSeek-V2-Coder MoE 路由精度
- 推荐动作：建议阅读 fused_topk_npu 函数中 norm_type 的计算方式，以及在 deepseek_v2.py 和 llada2.py 中 scoring_func 的透传方式，以理解 NPU MoE 门控的配置传递链路。对于 NNPU 后端维护者，建议后续为该逻辑添加单元测试。

功能与动机

DeepSeek-V2-Coder 模型（如 DeepSeek-Coder-V2-Lite-Instruct）在 NPU 上运行时，MoE 路由权重的 norm_type 被硬编码为 1（sigmoid），但该模型实际使用 softmax 评分函数，导致路由权重计算错误，模型输出精度下降。PR 描述中附带了精度对比截图（未在材料中展示）表明修复后精度恢复正常。

实现拆解

1. 修改 NPU MoE 门控内核的 norm_type 判断（python/sglang/srt/hardware_backend/npu/moe/topk.py） - 将原来固定为 1（sigmoid）的 norm_type 替换为条件表达式：若 topk_config.scoring_func == "softmax" 则设为 0，否则设为 1（sigmoid）。 - 这确保了在 NPU 专用 npu_moe_gating_top_k 算子中，路由权重的归一化方式与模型配置一致。
2. 在 DeepSeek-V2 MoE 初始化中透传 scoring_func（python/sglang/srt/models/deepseek_v2.py） - 在构造 TopK 对象的 topk_kwargs 字典中新增 scoring_func=config.scoring_func 键值对，使得 fused_topk_npu 能通过 topk_config.scoring_func 访问到实际评分函数。
3. 在 LLaDA2 MoE 初始化中透传 scoring_func（python/sglang/srt/models/llada2.py） - 类似地，在 LLaDA2MoE.__init__ 的 TopK 构造参数中新增 scoring_func=self.score_function，确保 LLaDA2 模型在未来 NPU 运行时也能正确选择归一化方式。
4. 无测试或配置变更：提交历史显示作者在 4 次提交中完成了功能实现与 lint 修复，但未包含单元测试或 CI 配置改动。

关键文件：

- python/sglang/srt/hardware_backend/npu/moe/topk.py（模块 NPU MoE；类别 source；类型 core-logic；符号 fused_topk_npu）：核心修复文件：将 MoE 门控的 norm_type 从固定值改为动态判断，直接解决了精度问题。

- python/sglang/srt/models/deepseek_v2.py（模块 DeepSeek 模型；类别 source；类型 data-contract；符号 DeepseekV2MoE.init）：透传 scoring_func 到 TopK 构造参数，确保 fused_topk_npu 能正确获取评分函数类型。
- python/sglang/srt/models/llada2.py（模块 LLaDA2 模型；类别 source；类型 data-contract；符号 LLaDA2MoE.init）：防御性修改：为 LLaDA2 模型同样透传 scoring_func，以兼容未来 NPU 部署场景。

关键符号：fused_topk_npu, DeepseekV2MoE.init, LLaDA2MoE.init

关键源码片段

python/sglang/srt/hardware_backend/npu/moe/topk.py

核心修复文件：将 MoE 门控的 norm_type 从固定值改为动态判断，直接解决了精度问题。

```
# python/sglang/srt/hardware_backend/npu/moe/topk.py
# 关键修改：根据 scoring_func 动态选择 norm_type
# 原代码：norm_type=1, # 1 for sigmoid, 0 for softmax
# 新代码：
elif (
    correction_bias is not None
    or topk_config.scoring_func == "sigmoid"
    or num_token_non_padded is not None
):
    topk_weights, topk_ids, _ = torch.ops.npu.npu_moe_gating_top_k(
        router_logits.to(torch.float32),
        k=topk_config.top_k,
        bias=(
            correction_bias.to(torch.float32)
            if correction_bias is not None
            else None
        ),
        k_group=topk_config.topk_group if use_grouped_topk else 1,
        group_count=topk_config.num_expert_group if use_grouped_topk else 1,
        group_select_mode=(1 if use_grouped_topk else 0),
        renorm=0,
        # 1 for sigmoid, 0 for softmax
        norm_type=(0 if topk_config.scoring_func == "softmax" else 1),
        routed_scaling_factor=(
            1 if renormalize else topk_config.routed_scaling_factor
        ),
        eps=float(1e-20),
    )
```

评论区精华

机器人 gemini-code-assist[bot] 在 review 中指出了代码中的一个潜在隐患：norm_type 定义时末尾多了一个逗号，可能导致其变成一个单元素元组而不是整数，从而引起 npu_moe_gating_top_k 算子的类型错误。该评论在原始 diff 中显示的是多行版本，但合并后

的最终代码（*head_excerpt*）中已移除该逗号，因此该问题已在合并前解决。无人类 reviewer 参与讨论。

- norm_type 末尾逗号导致的类型错误风险 (correctness): 该问题在合并后的最终代码中已解决（*head_excerpt* 中无逗号），但在提交历史中未见对应修复提交，可能已通过 force push 或合并时修正。

风险与影响

- 风险：
  1. 回归风险低：scoring_func 仅在构造 TopK 时通过参数传递，不影响原有的 correction_bias 或 grouped topk 分支逻辑；且变更仅影响 NPU 后端（topk.py 中的 npu_moe_gating_top_k 调用路径），不影响 GPU 或其他硬件。
  2. LLaDA2 模型暂无 NPU 测试覆盖：虽然 LLaDA2 的修改是防御性的，但由于缺乏 NPU 上的 LLaDA2 测试，该变更在 NPU 上可能被错误激活，需关注后续测试结果。
  3. 无测试配套：PR 未增加单元测试或集成测试，长期维护时可能难以回归验证。
- 影响：
  1. 用户影响：修复了 NPU 上 *DeepSeek-V2-Coder* 系列模型的推理精度问题，用户可直接受益于正确的模型输出。
  2. 系统影响：仅影响 NPU 硬件后端的 MoE 路由逻辑，不影响 GPU 或其他后端。
  3. 团队影响：变更较小且已合并，维护成本低；但缺少测试可能给未来重构带来隐患。 - 风险标记：缺少测试覆盖，防御性修改可能引入未验证路径

关联脉络

- PR #31782 [bugfix][NPU] Fix startup bug in olmoe 1b 7b: 同为 NPU 后端的 MoE 相关 bugfix，修复 topk 为 None 的崩溃，体现了 NPU 上 MoE 逻辑的持续维护。