PR #23856 完整报告

sgl-project/sglang

Use Torch `torch.mm` for Deepseek V3.2 Indexer GEMM

合并时间：2026-05-11 15:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23856

执行摘要

- 一句话：DeepSeek V3.2 Indexer GEMM 精度与性能优化
- 推荐动作：值得精读，展示了如何通过 Kernel 选择和数据类型布局优化来提升深度学习模型的精度和性能，同时保持了代码简洁性。

功能与动机

原始代码中 weights_proj(x) 输出 BF16，随后调用 .float() 转 FP32，导致 BF16 精度丢失和冗余内存拷贝。PR 作者通过 CUPTI 分析发现原路径使用 TST（BF16 输出），而 torch.mm 使用 TSS（FP32 输出），并提供了详细的精度对比表，证明 torch.mm 方案精度远优于原方案和 DeepGEMM，速度也最快。

实现拆解

1. 移除 DeepGEMM 分支：删除 _weights_proj_bf16_in_fp32_out 中 deep_gemm_wrapper.ENABLE_JIT_DEEPGEMM 条件下的整段逻辑（分配输出张量、调用 deep_gemm_wrapper.gemm_nt_bf16bf16f32），因为 DeepGEMM 性能更差且精度不如 torch.mm。
2. 添加 CUDA 专用路径：在 _is_cuda 条件下直接调用 torch.mm(x, self.weights_proj.weight.t(), out_dtype=torch.float32)，利用 Torch 对 out_dtype 参数的支持，实现 BF16 输入、FP32 累加和 FP32 输出，避免中间 BF16 转换。
3. 保留其他后端逻辑：对于 ROCm（_is_hip）和 CPU 等其他平台，仍保留原有的 self.weights_proj(x) 加类型转换路径，确保兼容性。

关键文件：

- python/sglang/srt/layers/attention/nsa/nsa_indexer.py（模块 注意力层；类别 source；类型 core-logic；符号 _weights_proj_bf16_in_fp32_out）：核心变更文件，修改了 Indexer 的 GEMM 计算路径，通过替换算子实现精度和性能双提升。

关键符号：_weights_proj_bf16_in_fp32_out

关键源码片段

python/sglang/srt/layers/attention/nsa/nsa_indexer.py

核心变更文件，修改了 Indexer 的 GEMM 计算路径，通过替换算子实现精度和性能双提升。

```
def _weights_proj_bf16_in_fp32_out(
    self, x: Union[torch.Tensor, Tuple[torch.Tensor, ...]]
) -> torch.Tensor:
    # aiter (ROCm gfx95): extract the passthrough bf16 tensor from the
    # 3-tuple (fp8, scale, bf16) produced by fused_rms_fp8_group_quant,
    # avoiding an expensive FP8-to-bf16 dequantization.
    if _use_aiter and _is_gfx95_supported and isinstance(x, tuple) and len(x) == 3:
        x = x[2]
    # CUDA path: use torch.mm with out_dtype=fp32 to keep fp32 accumulator,
    # avoiding bf16 -> fp16 -> fp32 round trips. This provides better accuracy
    # and performance than both DeepGEMM and F.linear+float().
    if _is_cuda:
        return torch.mm(x, self.weights_proj.weight.t(), out_dtype=torch.float32)
    # Fallback for non-CUDA backends (e.g., AMD ROCm): use F.linear and
    # optionally cast to fp32.
    weights, _ = self.weights_proj(x)
    if _is_hip:
        # Return bf16; multiplying with q_scale promotes back to fp32.
        return weights
    return weights.float()
```

评论区精华

Reviewer @Fridge003 建议移除自定义 op 包装器（custom_op），因为所需 Torch 版本已升级到 2.11，torch.mm 的 out_dtype 参数不再需要兼容性包装。作者 @b8zhong 回应"Just dropped it"，表明该建议已被采纳。

- 移除自定义 op 包装器 (design): 作者已移除该包装器，代码中不再包含自定义 op。

风险与影响

- 风险：该变更仅限于 NVIDIA CUDA 后端，对 AMD ROCm 和其他后端无影响。torch.mm 的 out_dtype 参数要求 Torch >= 2.10，当前仓库已满足。由于 torch.mm 是标准操作，回归风险较低，但建议验证 torch.compile 仍能正常工作。
- 影响：直接影响 DeepSeek V3.2 模型的推理精度和性能。精度提升可减少 token 选择中的平局（tie）频率；性能提升约 2 倍（根据 PR 提供的 CUPTI 数据，torch.mm 耗时约 5-6 us，原方案约 12-14 us）。对用户透明，无需配置更改。
- 风险标记：CUDA 专有优化，依赖 Torch 版本 >= 2.10

关联脉络

- PR #24799 [AMD] Fix DeepSeek import cascade by supporting both pre- and post-#2958 aiter fused_qk_rmsnorm APIs: 同为 DeepSeek 模型优化相关，但针对 AMD 后端的 import 兼容性修复。
- PR #24850 [MoE] Fix NaN in flashinfer TRT-LLM A2A dispatch by sanitizing padding slots: 同为 DeepSeek 模型相关的 MoE 调度 bug 修复，涉及精度问题。