PR #20562 完整报告

sgl-project/sglang

Use torch.addmm instead of separate mm and add_ calls for LoRA torch.native

合并时间：2026-03-27 05:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/20562

执行摘要

此 PR 通过使用 torch.addmm 融合 LoRA torch-native 后端中的矩阵乘法和加法操作，提升了计算性能约 4.4%。变更涉及两个核心文件，优化了 GPU 操作以减少开销，风险较低，适用于 LoRA 数量少且输入大的场景，直接改善服务吞吐量。

功能与动机

动机是提升 torch-native 后端在特定 LoRA 场景下的性能。当 LoRA 数量少（4-8）且输入为大型提示（如长令牌字符串）时，torch-native 后端表现优于 csgmv 后端。PR body 中指出："Torch Native performs better than csgmv when number of LoRAs is small (4-8) and inputs are larger prompts"，因此通过融合操作来加速计算。

实现拆解

实现主要分为两个模块：

1. lora backend 模块：在 python/sglang/srt/lora/backend/torch_backend.py 中，修改 TorchNativeLoRABatchInfo 类，添加 scalings_cpu 字段以存储缩放因子，并更新 run_lora_a_sgemm、run_qkv_lora、run_gate_up_lora 等方法，将 scaling_tensor 引用从 GPU tensor 改为 CPU tensor，避免 GPU 到 CPU 的同步开销。
2. lora ops 模块：在 python/sglang/srt/lora/torch_ops/lora_ops.py 中，修改核心函数：
   - sgemm_lora_a_fwd：将 torch.mm 和标量乘法替换为 torch.addmm(out_slice, x_seq, w_seq.T, beta=0, alpha=scaling_tensor[lora_idx].item(), out=out_slice)。
   - sgemm_lora_b_fwd：将 torch.mm 和 add_ 替换为 torch.addmm(out_slice, x_slice, w_slice.T, beta=1, alpha=1, out=out_slice)。 这些变更减少了 GPU 内核调用次数，提升了计算效率。

评论区精华

Review 过程中没有具体技术讨论，三位 reviewer（zminglei, jasperjiaguo, Fridge003）均直接批准。在 Issue 评论中，claude-pr-review-bot 提供了补充分析：

> "Clean optimization replacing separate torch.mm + scalar multiply / add_ with fused torch.addmm calls in the torch-native LoRA backend, plus adding scalings_cpu to avoid implicit GPU-to-CPU sync when indexing a GPU tensor in a CPU-side loop. Both changes are semantically equivalent to the original code." 这确认了优化的正确性

和低风险性，但缺乏深度交锋。

风险与影响

风险：

- 数值精度：操作融合可能导致浮点运算顺序变化，但测试显示 cosine 相似度接近 1（平均 1.000124），远高于阈值 0.9999，影响可忽略。
- 兼容性：变更仅限 torch-native 后端，不影响其他后端；添加 scalings_cpu 需确保在 prepare_lora_batch 中正确初始化。
- 性能回归：通过基准测试验证，RPS 从 40.12 提升到 41.89（+4.4%），表明无回归。

影响：

- 用户：吞吐量提升，改善请求处理速度。
- 系统：优化 GPU 资源使用，减少内核启动开销。
- 团队：代码更简洁，便于维护，但需确保跨环境一致性。

关联脉络

与此 PR 相关的历史 PR 包括 #20606，它同样涉及 LoRA 性能优化，但针对不同后端（NSA），显示团队在多个后端持续进行性能改进的趋势。本 PR 没有直接关联的 Issue，但基于近期 PR 分析，性能优化是仓库的常见主题，例如 #21421 也涉及操作融合。这揭示了团队对 GPU 计算效率的关注和演进方向。