

PR #51498 完整报告

vllm-project/vllm

[Model] Add tower and connector LoRA support for LFM2-VL

合并时间: 2026-08-20 20:33

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51498>

执行摘要

- 一句话: 为 LFM2-VL 添加 tower 与 connector LoRA 支持
- 推荐动作: 该 PR 值得精读, 尤其是对多模态 LoRA 实现感兴趣的开发者。关键设计决策包括将投影层转换为 ReplicatedLinear 以支持 LoRA 包装, 以及通过 token 预算辅助函数桥接视觉与语言 token 数量差异。但需注意测试覆盖的缺失, 建议后续补充 LFM2 专属的单元测试。

功能与动机

issue #31479 旨在为更多多模态模型启用 tower 和 connector 的 LoRA 支持。LFM2-VL 已实现 SupportsLoRA 并暴露了多模态模块映射, 但缺少 tower/connector LoRA 路径所需的 token 预算辅助函数, 且其投影器使用普通 nn.Linear, 无法被 vLLM 的 LoRA 包装器替换。因此本 PR 将投影层转换为 ReplicatedLinear, 并添加 get_num_mm_encoder_tokens 和 get_num_mm_connector_tokens 来计算 token 数量, 以桥接视觉塔输入长度与语言模型 token 数之间的差异。

实现拆解

1. 修改 vllm/model_executor/models/lfm2_vl.py:
 - 导入 ReplicatedLinear。
 - 将 Lfm2VLMultiModalProjector 中的 linear_1 和 linear_2 从 nn.Linear 替换为 ReplicatedLinear, 添加 prefix 参数和 return_bias=False, 保持权重名称和仅张量前向行为。
 - 新增 get_num_mm_encoder_tokens 和 get_num_mm_connector_tokens 方法, 根据 downsample_factor 计算 token 数量: $\text{encoder tokens} = \text{image tokens} \times \text{factor}^2$, $\text{connector tokens} = \text{encoder tokens} \div \text{factor}^2$ 。
2. 修改 vllm/model_executor/models/lfm2_siglip2.py:
 - 将 Siglip2VisionEmbeddings 中的 patch_embedding 从 nn.Linear 转换为 ReplicatedLinear, 使用 input_size 和 output_size 参数, return_bias=False。
 - 导入 ReplicatedLinear。
3. 测试配套: 根据 review 反馈删除了原本新增的 LFM2-VL 专属测试文件, 转而依赖现有的 GPU model runner 测试 (如 test_set_active_mm_loras_builds_tower_and_connector_mappings) 来覆盖通用 tower/connector 映射构建, 但 LFM2 专属细节并未被覆盖。

4. 配置与依赖：无额外配置变更，仅修改了模型实现。

关键文件：

- `vllm/model_executor/models/lfm2_vl.py`（模块 模型实现；类别 `source`；类型 `data-contract`；符号 `get_num_mm_encoder_tokens`, `get_num_mm_connector_tokens`）：核心变更文件，将投影器线性层转换为 `ReplicatedLinear` 并新增 `token` 预算辅助函数，是启用 LoRA 的关键。
- `vllm/model_executor/models/lfm2_siglip2.py`（模块 模型实现；类别 `source`；类型 `data-contract`）：将 `Siglip2 patch` 投影层转换为 `ReplicatedLinear`，使 `tower LoRA` 目标可被发现和包装。

关键符号：`get_num_mm_encoder_tokens`, `get_num_mm_connector_tokens`

关键源码片段

`vllm/model_executor/models/lfm2_vl.py`

核心变更文件，将投影器线性层转换为 `ReplicatedLinear` 并新增 `token` 预算辅助函数，是启用 LoRA 的关键。

```
# vllm/model_executor/models/lfm2_vl.py
```

```
class Lfm2VLMultiModalProjector(nn.Module):
    def __init__(self, config, prefix: str = ""):
        super().__init__()
        self.use_data_parallel = is_vit_use_data_parallel()

        in_channels = config.vision_config.hidden_size * (config.downsample_factor**2)
        self.factor = config.downsample_factor
        self.projector_use_layernorm = config.projector_use_layernorm
        if self.projector_use_layernorm:
            self.layer_norm = nn.LayerNorm(in_channels)

        # 将 nn.Linear 替换为 ReplicatedLinear，使投影层可被 LoRA 包装器替换
        self.linear_1 = ReplicatedLinear(
            in_channels,
            config.projector_hidden_size,
            bias=config.projector_bias,
            prefix=maybe_prefix(prefix, "linear_1"),
            return_bias=False, # 保持仅张量输出，便于后续包装
        )
        self.act = ACT2FN[config.projector_hidden_act]
        self.linear_2 = ReplicatedLinear(
            config.projector_hidden_size,
            config.text_config.hidden_size,
            bias=config.projector_bias,
            prefix=maybe_prefix(prefix, "linear_2"),
            return_bias=False,
        )
```

```
def get_num_mm_encoder_tokens(self, num_image_tokens: int) -> int:
    # 视觉塔 token 数 = 图像 token 数 × downsample_factor²
    downsample_factor = self.config.downsample_factor
    return num_image_tokens * downsample_factor ** 2

def get_num_mm_connector_tokens(self, num_vision_tokens: int) -> int:
    # 连接器 token 数 = 视觉塔 token 数 ÷ downsample_factor²
    downsample_factor = self.config.downsample_factor
    return num_vision_tokens // downsample_factor ** 2
```

vllm/model_executor/models/lfm2_siglip2.py

将 Siglip2 patch 投影层转换为 ReplicatedLinear，使 tower LoRA 目标可被发现和包装。

```
# vllm/model_executor/models/lfm2_siglip2.py
```

```
class Siglip2VisionEmbeddings(nn.Module):
    def __init__(self, config: Siglip2VisionConfig):
        super().__init__()
        self.config = config
        self.embed_dim = config.hidden_size
        self.patch_size = config.patch_size

        # 转换为 ReplicatedLinear，使该层可被 LoRA 包装器发现和替换
        self.patch_embedding = ReplicatedLinear(
            input_size=config.num_channels * self.patch_size * self.patch_size,
            output_size=self.embed_dim,
            return_bias=False,
        )
        self.num_patches = config.num_patches
        self.position_embedding_size = int(self.num_patches ** 0.5)
        self.position_embedding = nn.Embedding(self.num_patches, self.embed_dim)
```

评论区精华

review 中主要讨论了以下两点：

- 删除新增的 LFM2 专属测试文件：linitra24 建议移除 tests/models/multimodal/processing/test_lfm2_vl.py，作者回应已删除，并指出这会丢失 LFM2 专属的 token 计数往返、零 token 和投影器 LoRA 包装断言，但现有测试仍覆盖通用映射构建。
- 将 Siglip2 patch 投影层也改为 ReplicatedLinear：linitra24 询问 lfm2_siglip2.py 中的 patch_embedding 是否也需要更改。作者确认已更改，因为该层作为 nn.Linear 无法被 get_supported_lora_modules() 发现，转换为 ReplicatedLinear 后即可被 LoRA 包装，同时保留偏置、状态字典名称、仅张量前向和 TP 语义。
- 删除 LFM2 专属测试文件 (testing)：测试文件被删除，作者承认丢失了 LFM2 专属断言，但现有测试仍覆盖通用映射。
- Siglip2 patch 投影层也应转换为 ReplicatedLinear (design)：作者将该层转换为 ReplicatedLinear，保留了偏置、名称和 TP 语义。

风险与影响

- 风险:

1. 行为改变: 将 `nn.Linear` 替换为 `ReplicatedLinear` 可能改变模型的数值行为, 尤其是当 `return_bias=False` 时, 虽然作者声称已通过数值一致性测试, 但未运行完整生成测试, 存在潜在回归风险。
2. 测试覆盖缺口: 删除了 LFM2 专属测试文件后, LFM2 的 token 计算和 LoRA 包装细节缺少直接测试覆盖, 未来变更可能不易察觉回归。
3. 兼容性: `ReplicatedLinear` 与 `nn.Linear` 在权重加载和状态字典名称上保持一致, 但若模型配置变化或 TP 维度不同, 可能影响加载。
4. 性能: `ReplicatedLinear` 在 TP=1 时可能略有性能开销, 但通常可忽略。- 影响: 对用户: LFM2-VL 模型现在可以使用 tower 和 connector LoRA 进行微调和推理, 扩展了模型的定制能力。对系统: 改动仅影响 LFM2-VL 相关路径, 不涉及核心路径。对团队: 为后续其他多模态模型启用 LoRA 提供了可参考的实现模式, 促进了 issue #31479 的整体进展。

- 风险标记: 测试覆盖缺口, 行为改变风险

关联脉络

- PR #31479 [Feature]: Enable LoRA support for tower and connector in more MM models: 本 PR 是该 issue 的一部分, 实现了 LFM2-VL 的 tower/connector LoRA 支持。