PR #25054 完整报告

sgl-project/sglang

Support Gemma4 MoE NVFP4

合并时间：2026-05-21 13:45

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25054

执行摘要

- 一句话：支持 Gemma4 MoE NVFP4 模型推理
- 推荐动作：该 PR 值得精读，特别是 FusedMoE.make_expert_params_mapping 的复用模式以及 get_activation_type 的 gated/non-gated 抽象设计。权重加载部分的 per-expert 映射逻辑是值得关注的设计决策。建议在合并后补充单元测试覆盖新的映射路径。

功能与动机

来自 PR 描述：'Support NVFP4 Gemma4 MoE nvidia/Gemma-4-26B-A4B-NVFP4'。需要使 SGLang 能够加载并高效推理 Gemma-4 系列的 NVFP4 量化模型。

实现拆解

1. 扩展权重加载器：在 gemma4_causal.py 和 gemma4_mm.py 的 load_weights 方法中引入 per_expert_params_mapping，利用 FusedMoE.make_expert_params_mapping 生成每个专家的参数映射，覆盖 per-expert 检查点格式（包括 weight, weight_scale, weight_scale_2, input_scale），替代了原有的正则匹配逻辑，同时保留 fused 格式的向后兼容。
2. 重构激活类型映射：在 flashinfer_trtllm.py 中将 get_activation_type 函数签名扩展为 (activation: str, is_gated: bool = True) -> int，根据 is_gated 区分 gated（Swiglu/Geglu）与非 gated（Silu/Gelu/Relu2）激活类型，并新增 GELU/GEGLU 支持。
3. 更新量化配置校验：在 modelopt_quant.py 中扩展回退 ActivationType 枚举（新增 Geglu, Identity），统一使用 _SUPPORTED_ACT_STRS 校验激活字符串，并在 apply 方法中将 activation 映射与 is_gated 结合传递给 FlashInfer 内核。
4. 自动选择 MoE 后端：在 server_args.py 的 _handle_model_specific_adjustments 中，当模型架构为 Gemma4ForCausalLM 且 GPU 为 SM100 时，自动将 moe_runner_backend 设为 flashinfer_trtllm。
5. 兼容 TopK 属性名：在 scheduler.py 的 init_moe_gemm_config 中扩展属性检测列表，加入 top_k_experts，使 Gemma4 的 MoE 配置能正确初始化。

关键文件：

- python/sglang/srt/models/gemma4_causal.py（模块 模型层；类别 source；类型 data-contract；符号 load_weights）：核心模型权重加载改造，新增 per-expert NVFP4 检查点格式支持

- python/sglang/srt/models/gemma4_mm.py（模块 多模态模型；类别 source；类型 data-contract）：多模态模型的权重加载同样适配 per-expert 格式，与 gemma4_causal 同步修改
- python/sglang/srt/layers/quantization/modelopt_quant.py（模块 量化层；类别 source；类型 data-contract；符号 Geglu, get_activation_type）：扩展 ActivationType 枚举以支持 GEGLU/Identity，更新激活校验逻辑以支持新激活函数
- python/sglang/srt/layers/moe/moe_runner/flashinfer_trtllm.py（模块 MoE 执行器；类别 source；类型 core-logic；符号 get_activation_type）：重构 get_activation_type 函数以区分 gated/non-gated 激活，新增 GELU/GEGLU 映射
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：扩展 MoE top-k 属性名检测以兼容 Gemma4 的 top_k_experts 字段
- python/sglang/srt/server_args.py（模块 服务参数；类别 source；类型 configuration）：为 Gemma-4 模型在 SM100 上自动选择 flashinfer_trtllm MoE 后端

关键符号：get_activation_type, Gemma4TextModel.load_weights, Gemma4ForCausalLM.load_weights, init_moe_gemm_config, _handle_model_specific_adjustments

关键源码片段

python/sglang/srt/models/gemma4_causal.py

核心模型权重加载改造，新增 per-expert NVFP4 检查点格式支持

```
# 在 Gemma4TextModel.load_weights 中：

# Per-expert checkpoint format used by compressed-tensors / FP8
# (e.g. RedHatAI/*-FP8-Dynamic) and by ModelOpt NVFP4
# (e.g. nvidia/Gemma-4-*-NVFP4). Each expert is stored as a
# separate key with shape (out, in):
# experts.<id>.{gate,up,down}_proj.{weight,weight_scale,
# weight_scale_2,input_scale}
# `make_expert_params_mapping` emits tuples whose `weight_name` ends
# in a trailing dot, so the standard `name.replace(weight_name,
# param_name)` collapses every suffix uniformly to the fused
# FusedMoE params (experts.w13_*, experts.w2_*).
per_expert_params_mapping = FusedMoE.make_expert_params_mapping(
    ckpt_gate_proj_name="gate_proj",
    ckpt_down_proj_name="down_proj",
    ckpt_up_proj_name="up_proj",
    num_experts=num_experts,
)

# 在权重加载循环中，优先匹配 per-expert 格式：
for param_name, weight_name, expert_id, shard_id in per_expert_params_mapping:
    if weight_name not in orig_name:
        continue
    name = orig_name.replace(weight_name, param_name)
```

```
            if name not in params_dict:
                continue
            param = params_dict[name]
            weight_loader = param.weight_loader
            # 直接传递 expert_id 和 shard_id，无需手动分块
            weight_loader(param, loaded_weight, name, shard_id, expert_id)
```

python/sglang/srt/layers/moe/moe_runner/flashinfer_trtllm.py

重构 get_activation_type 函数以区分 gated/non-gated 激活，新增 GELU/GEGLU 映射

```
def get_activation_type(activation: str, is_gated: bool = True) -> int:
    """Map SGLang activation string to FlashInfer ActivationType int value."""
    from flashinfer.fused_moe.core import ActivationType

    if is_gated:
        # Gated variants: SwiGLU, GEGLU (used by Gemma-4 MoE)
        _ACTIVATION_STR_TO_TYPE = {
            "silu": ActivationType.Swiglu,
            "gelu": ActivationType.Geglu,
        }
    else:
        # Non-gated variants: traditional activations
        _ACTIVATION_STR_TO_TYPE = {
            "silu": ActivationType.Silu,
            "gelu": ActivationType.Gelu,
            "relu2": ActivationType.Relu2,
        }
    act = _ACTIVATION_STR_TO_TYPE.get(activation)
    if act is None:
        raise ValueError(
            f"Unsupported activation '{activation}' for TRTLLM MoE "
            f"(is_gated={is_gated}). "
            f"Expected one of {list(_ACTIVATION_STR_TO_TYPE.keys())}."
        )
    return act.value
```

评论区精华

审查中主要讨论了以下话题：

- torch.compile 性能问题：wenscarl 指出 trtllm_mha 在 --enable-torch-compile 下性能下降，kpham-sgl 建议将相关修复延迟到另一个 PR，避免影响范围过大。
- 权重加载鲁棒性：gemini-code-assist 建议在 per-expert 匹配循环中仅当参数成功处理时才 continue，并添加调试日志；kpham-sgl 询问是否有更优雅的映射方式。
- 激活类型向后兼容：b8zhong 询问 get_activation_type 是否需要向后兼容，pyc96 回复已移除不常用的非 gated relu2，仅保留必要的分支。
- MoE 后端选择：b8zhong 询问为何不统一使用 trtllm-gen，pyc96 说明 trtllm-gen 性能更好且 cutlass 存在 padding 问题，因此保留 flashinfer_trtllm 作为默认后端。

- torch.compile 性能问题修复推迟 (performance): 决定不在本 PR 包含 torch.compile 相关改动，推迟至后续 PR。
- per-expert 权重加载映射的鲁棒性 (correctness): 未明确答复，但代码最终保留了 per_expert_params_mapping 的新实现，替换了之前正则匹配，可能已隐含改进。
- get_activation_type 向后兼容性 (design): 接受简化，移除不常用的非 gated relu2。
- cutlass_moe 与 trtllm-gen 后端选择 (performance): 决定暂时保留 cutlass 改动不影响主要路径，默认后端为 flashinfer_trtllm。

风险与影响

- 风险:
  1. 权重加载路径测试不足：per-expert 映射的新逻辑无对应新增单元测试，可能遗漏边界情况（如 scale 字段缺失）。
  2. 激活映射扩展影响其他模型：get_activation_type 的 is_gated 参数修改了现有调用接口，虽已更新所有调用点，但若外部代码直接导入该函数可能需适配。
  3. 自动后端选择条件：server_args.py 中自动切换 flashinfer_trtllm 的条件仅基于 is_sm100_supported()，未来若其他 GPU 也支持需扩展。
  4. torch.compile 兼容性：当前 NVFP4 路径在 --enable-torch-compile 下存在性能退化，本 PR 未修复，需后续跟进。 - 影响：用户可直接加载 nvidia/Gemma-4-*-NVFP4 模型并运行推理，输出 token 吞吐量在 trtllm_mha 后端下可达 225.78 tok/s（TP=1，输入 1000，输出 250）。对其他使用 FP4 MoE 的模型（如 RedHatAI FP8 系列）无影响，因为 per-expert 映射兼容 fused 格式。团队需关注后续 PR 对 torch.compile 性能的修复及更广泛的 activation 类型支持。 - 风险标记：缺少测试覆盖，activation 映射扩展影响其他模型，torch.compile 性能待跟进

关联脉络

- 暂无明显关联 PR