PR #22427 完整报告

sgl-project/sglang

[Score API] Add return_pooled_hidden_states to Scoring API for SequenceClassification / RewardModel

合并时间：2026-04-16 05:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22427

执行摘要

- 一句话：为评分 API 添加返回 pooled 隐藏状态功能，支持序列分类和奖励模型。
- 推荐动作：此 PR 值得精读，特别关注池化层和调度器中的设计决策：如何平衡功能需求与性能（如 CUDA 图处理）、如何处理混合 batch 中的标志聚合，以及模型文件中的条件返回模式。这些决策对类似 API 扩展有参考价值。

功能与动机

根据 PR body，此功能旨在为下游消费者提供模型内部表示，用于蒸馏、可解释性或二次评分管道。引用 PR body 原话：'This is useful for downstream consumers that need the model's internal representation alongside the final scores — e.g., for distillation, interpretability, or secondary scoring pipelines.'

实现拆解

1. 核心数据结构扩展：在 EmbeddingPoolerOutput（pooler.py）和 ScoreResult（tokenizer_manager_score_mixin.py）中添加 pooled_hidden_states 字段，用于存储任务头前的隐藏状态。
2. 池化逻辑更新：在 pooler.py 中新增 pool_hidden_states() 函数实现 LAST/CLS 池化，并修改 score_and_pool() 在单项目和多项目评分（MIS）路径下条件性捕获隐藏状态。
3. 请求流水线贯通：在 ForwardBatch、Req、ScheduleBatch 等数据结构中添加 return_pooled_hidden_states 标志，并通过调度器（scheduler.py）和输出处理器（scheduler_output_processor_mixin.py）传递，确保从请求到响应的完整链路。
4. 模型适配：更新多个序列分类和奖励模型文件（如 llama_reward.py、qwen2_rm.py），在 forward 方法中根据标志返回 pooled hidden states。
5. 测试与配置配套：新增测试文件 test_pooled_hidden_states.py 覆盖 Engine API、HTTP 集成和错误场景；在 server_args.py 中添加 MIS 模式下的 CUDA 图禁用逻辑，确保兼容性。

关键文件：

- python/sglang/srt/layers/pooler.py（模块 池化层；类别 source；类型 core-logic；符号 pool_hidden_states, score_and_pool, EmbeddingPoolerOutput）：核心池化层，新增 pool_hidden_states 函数并扩展 score_and_pool 以支持 pooled hidden states 捕获，是

功能实现的基础。

- python/sglang/srt/managers/tokenizer_manager_score_mixin.py（模块 评分管理器；类别 source；类型 dependency-wiring；符号 ScoreResult, _process_multi_item_scoring_results, _process_single_item_scoring_results）：评分结果处理的核心文件，扩展 ScoreResult 以包含 pooled_hidden_states，并更新处理逻辑以支持新标志。
- python/sglang/srt/models/llama_reward.py（模块 模型适配；类别 source；类型 data-contract；符号 LlamaForSequenceClassification.forward, LlamaForSequenceClassificationWithNormal_Weights.forward）：关键模型文件，展示如何在新功能中适配序列分类模型的 forward 方法，条件性返回 pooled hidden states。
- test/registered/prefill_only/test_pooled_hidden_states.py（模块 测试套件；类别 test；类型 test-coverage；符号 TestPooledHiddenStatesEngine, TestPooledHiddenStatesMISEngine, TestPooledHiddenStatesHTTP, TestPooledHiddenStatesCausalLMRejection）：新增的全面测试文件，覆盖 Engine API、HTTP 集成、MIS 模式和错误处理，确保功能正确性和稳定性。

关键符号：pool_hidden_states, score_and_pool, LlamaForSequenceClassification.forward, TokenizerManagerScoreMixin.score_request, Scheduler.run_batch

关键源码片段

python/sglang/srt/layers/pooler.py

核心池化层，新增 pool_hidden_states 函数并扩展 score_and_pool 以支持 pooled hidden states 捕获，是功能实现的基础。

```
def pool_hidden_states(
    pooling_type: PoolingType,
    hidden_states: torch.Tensor,
    forward_batch: ForwardBatch,
) -> torch.Tensor:
    """Pool hidden_states by PoolingType (LAST/CLS).
    Raw pooling only — no normalize, no dim truncation.
    Returns shape (batch_size, hidden_size).
    """
    if pooling_type == PoolingType.LAST:
        # 提取每个序列的最后一个token的隐藏状态
        last_token_indices = torch.cumsum(forward_batch.extend_seq_lens, dim=0) - 1
        return hidden_states[last_token_indices]
    elif pooling_type == PoolingType.CLS:
        # 提取每个序列的第一个token（CLS）的隐藏状态
        prompt_lens = forward_batch.extend_seq_lens
        first_token_flat_indices = torch.zeros_like(prompt_lens)
        first_token_flat_indices[1:] += torch.cumsum(prompt_lens, dim=0)[:-1]
        return hidden_states[first_token_flat_indices]
    else:
```

```
        raise ValueError(f"Unsupported pooling type: {pooling_type}")
```

python/sglang/srt/managers/tokenizer_manager_score_mixin.py

评分结果处理的核心文件，扩展 ScoreResult 以包含 pooled_hidden_states，并更新处理逻辑以支持新标志。

```
@dataclass(frozen=True, slots=True)
class ScoreResult:
    scores: List[List[float]]
    prompt_tokens: int = 0
    # Per-item pooled hidden states (pre-head transformer output).
    # CPU tensors when return_pooled_hidden_states=True; kept as tensors so
    # in-process consumers (gRPC, engine API) avoid a .tolist() round-trip.
    # The HTTP path converts to lists in serving_score.py before JSON serialization.
    # Same layout as scores: one tensor per item (not a single packed 2D tensor).
    pooled_hidden_states: Optional[List[Optional[torch.Tensor]]] = None
```

评论区精华

- CUDA 图支持：reviewer kpham-sgl 询问是否支持 CUDA 图，作者 sundar24295s 回应单项目评分已在 piecewise_cuda_graph_runner 中处理，但 MIS 模式暂时禁用，留待后续 PR 优化。
- 数据结构澄清：kpham-sgl 质疑 pooled_hidden_states 在 MIS 模式下是 List 还是 packed tensor，作者澄清为“每个项目一个 tensor”，并在代码注释中明确。
- 错误处理增强：kpham-sgl 建议对不支持 pooled hidden states 的模型（如 BERT/XLM-R）添加拒绝检查，作者采纳并在 configs/model_config.py 中新增 is_cross_encoding_pooler_model() 函数实现。
- 性能权衡：讨论中确认 return_pooled_hidden_states 在 batch 级别以 any() 聚合，意味着即使部分请求不需要，相关计算和内存复制仍会发生，这是为简化实现而接受的折衷。
  - CUDA 图支持与性能优化 (performance): 作者回应单项目评分已在 piecewise_cuda_graph_runner 中处理（capture 时设置 return_pooled_hidden_states=True），但 MIS 模式在 server_args.py 中禁用 CUDA 图，留待后续 PR 优化。
  - 数据结构格式澄清 (correctness): 作者澄清为“每个项目一个 tensor”，并在 ScoreResult 注释中明确布局，避免混淆。
  - 错误处理与模型兼容性 (correctness): 作者采纳建议，在 configs/model_config.py 中添加 is_cross_encoding_pooler_model() 函数，并在 tokenizer_manager_score_mixin.py 中集成检查。

风险与影响

- 风险：1. 性能开销：当 batch 中包含混合请求（部分需要 pooled hidden states，部分不需要）时，所有请求都会触发隐藏状态的计算和 CPU 复制，可能增加延迟和内存使用。 2. CUDA 图兼容性：MIS 模式下强制禁用 CUDA 图和 piecewise CUDA 图（通过 server_args.py 的 _handle_multi_item_scoring），可能影响高吞吐场景的性能。 3. 回归

风险：变更涉及调度器、池化层和多个模型文件，若新标志处理不当，可能破坏现有评分功能的正确性，尤其是 MIS 路径下的边界情况。 4. 数据一致性：在 scheduler_output_processor_mixin.py 中，初始实现存在 IndexError 风险（混合 batch 中 pooled hidden states 长度不匹配），虽已修复，但类似逻辑需持续关注。

- 影响：对用户：新增 API 参数 return_pooled_hidden_states，为序列分类和奖励模型用户提供模型内部表示，扩展了蒸馏、可解释性分析等下游应用场景，默认关闭不影响现有行为。对系统：在评分流水线中引入额外数据流，增加了复杂性和轻微性能开销，但通过优化（如条件性捕获）和全面测试最小化影响。对团队：需维护新字段和相关逻辑，测试覆盖确保了功能稳定性，但后续需关注 CUDA 图在 MIS 模式下的优化。
- 风险标记：性能开销，CUDA 图兼容性，回归风险

关联脉络

- PR #21887 [Ray] Add data parallel (DP) and DP attention support to RayEngine: 同样扩展评分 API 功能，涉及 RayEngine 的评分支持，可参考其跨模块变更模式。
- PR #22897 streaming session: trim spec v2 overshoot in cache_finished_req: 涉及评分相关的缓存和会话管理，但焦点不同；此 PR 补充了评分 API 的新特性。