

PR #28583 完整报告

sgl-project/sglang

Revert "revert the head_dim assignment from PR 23862"

合并时间: 2026-06-18 08:11

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28583>

执行摘要

- 一句话: 重新应用 PR#23862 的 head_dim 赋值逻辑
- 推荐动作: 该 PR 是团队在 head_dim 赋值策略上的反复, 目前恢复到 PR#23862 的方案, 并附带单元测试。建议阅读 model_config.py 中的 _derive_model_shapes 方法与新增测试文件, 理解 shape 派生与回写的设计决策。

功能与动机

PR#28571 (revert of PR#23862) 导致了手动测试失败, 因此需要 revert 该 revert, 恢复 PR#23862 的改动。

实现拆解

1. 在 model_config.py 的 _derive_model_shapes 方法中, head_dim 等属性的 getattr 默认值从直接的计算表达式改为 None, 然后在缺失时计算并调用 setattr 回写。这样做使得计算出的 shape 在 hf_text_config 上可供后续代码使用。
2. 新增 test_model_config_shapes.py 测试文件, 包含两个测试用例:
test_optional_head_dims_default_when_none 验证所有 head dim 为 None 时都能正确计算默认值 (128), 并回写到 text_config; test_explicit_head_dims_are_preserved 验证显式指定的不同 head dim 值 (128/96/64/48) 能被保留。
3. 测试使用 _make_text_config 辅助构造 Mixtral 模型的配置, 通过直接调用 ModelConfig._derive_model_shapes 来验证 shape 派生逻辑。

关键文件:

- python/sglang/srt/configs/model_config.py (模块 模型配置; 类别 source; 类型 data-contract; 符号 _derive_model_shapes): 核心变更文件, 修改了 head_dim/v_head_dim/swa_head_dim/swa_v_head_dim 的赋值逻辑
- test/registered/unit/configs/test_model_config_shapes.py (模块 测试; 类别 test; 类型 test-coverage; 符号 _make_text_config, TestModelConfigShapes, _derive_shapes, test_optional_head_dims_default_when_none): 新增测试文件, 覆盖 head_dim 默认值与显式赋值场景

关键符号: _derive_model_shapes, _make_text_config, _derive_shapes, test_optional_head_dims_default_when_none, test_explicit_head_dims_are_preserved

关键源码片段

python/sglang/srt/configs/model_config.py

核心变更文件，修改了 head_dim/v_head_dim/swa_head_dim/swa_v_head_dim 的赋值逻辑

```
def _derive_model_shapes(self):
    # head_dim: 尝试从 hf_text_config 获取，若 None 则计算并回写
    self.head_dim = getattr(self.hf_text_config, 'head_dim', None)
    if self.head_dim is None:
        self.head_dim = (
            self.hf_text_config.hidden_size
            // self.hf_text_config.num_attention_heads
        )
    # 回写 head_dim 供后续模块使用
    setattr(self.hf_text_config, 'head_dim', self.head_dim)

    # v_head_dim: 同样逻辑，默认与 head_dim 一致
    self.v_head_dim = getattr(self.hf_text_config, 'v_head_dim', None)
    if self.v_head_dim is None:
        self.v_head_dim = self.head_dim
        setattr(self.hf_text_config, 'v_head_dim', self.v_head_dim)

    # swa_head_dim: 默认使用 head_dim
    self.swa_head_dim = getattr(self.hf_text_config, 'swa_head_dim', None)
    if self.swa_head_dim is None:
        self.swa_head_dim = self.head_dim
        setattr(self.hf_text_config, 'swa_head_dim', self.swa_head_dim)

    # swa_v_head_dim: 默认使用 swa_head_dim
    self.swa_v_head_dim = getattr(self.hf_text_config, 'swa_v_head_dim', None)
    if self.swa_v_head_dim is None:
        self.swa_v_head_dim = self.swa_head_dim
        setattr(self.hf_text_config, 'swa_v_head_dim', self.swa_v_head_dim)

    # FIXME: temporary special judge for MLA architecture
    if (
        "DeepseekV2ForCausalLM" in self.hf_config.architectures
        or "DeepseekV32ForCausalLM" in self.hf_config.architectures
        or "DeepseekV3ForCausalLM" in self.hf_config.architectures
        or "DeepseekV3ForCausalLMNextN" in self.hf_config.architectures
        or "Glm4MoeLiteForCausalLM" in self.hf_config.architectures
        or "Glm4MoeLiteForCausalLMNextN" in self.hf_config.architectures
        or "GlmMoeDsaForCausalLM" in self.hf_config.architectures
        or "LongcatFlashForCausalLM" in self.hf_config.architectures
        or "LongcatFlashForCausalLMNextN" in self.hf_config.architectures
        or "DotsVLMForCausalLM" in self.hf_config.architectures
        or "MistralLarge3ForCausalLM" in self.hf_config.architectures
        or (
            "PixtralForConditionalGeneration" in self.hf_config.architectures
```

```

        and getattr(self.hf_text_config, "kv_lora_rank", None) is not None
    )
    or "MistralLarge3ForCausalLMEagle" in self.hf_config.architectures
    or "KimiK25ForConditionalGeneration" in self.hf_config.architectures
    or "Eagle3DeepseekV2ForCausalLM" in self.hf_config.architectures
):
    self.head_dim = 256
    self.attention_arch = AttentionArch.MLA
    self.kv_lora_rank = self.hf_text_config.kv_lora_rank
    self.qk_nope_head_dim = self.hf_text_config.qk_nope_head_dim
    self.qk_rope_head_dim = self.hf_text_config.qk_rope_head_dim
    self.v_head_dim = self.hf_text_config.v_head_dim
    self.index_head_dim = (
        get_dsa_index_head_dim(self.hf_text_config)
        if is_deepseek_dsa(self.hf_text_config)
        else None
    )

```

test/registered/unit/configs/test_model_config_shapes.py

新增测试文件，覆盖 head_dim 默认值与显式赋值场景

```

"""Unit tests for ModelConfig shape normalization."""

import unittest
from types import SimpleNamespace

from sglang.srt.configs.model_config import ModelConfig
from sglang.test.ci.ci_register import register_cpu_ci
from sglang.test.test_utils import CustomTestCase

# 注册为 CPU CI 测试，预计运行时间 1 秒
register_cpu_ci(est_time=1, suite="base-a-test-cpu")

def _make_text_config(**overrides):
    # 构造一个简易的 Mixtral 模型配置，允许覆盖字段
    defaults = dict(
        architectures=["MixtralForCausalLM"],
        model_type="mixtral",
        hidden_size=4096,
        num_attention_heads=32,
        num_hidden_layers=2,
        vocab_size=32000,
        num_key_value_heads=8,
    )
    defaults.update(overrides)
    return SimpleNamespace(**defaults)

class TestModelConfigShapes(CustomTestCase):
    def _derive_shapes(self, text_config):

```

```

# 直接创建 ModelConfig 实例（绕过 __init__）并调用 _derive_model_shapes
model_config = ModelConfig.__new__(ModelConfig)
model_config.hf_config = text_config
model_config.hf_text_config = text_config
model_config._derive_model_shapes()
return model_config

def test_optional_head_dims_default_when_none(self):
    # 验证当 head_dim 等都为 None 时，能被正确计算为默认值 128
    text_config = _make_text_config(
        head_dim=None,
        v_head_dim=None,
        swa_head_dim=None,
        swa_v_head_dim=None,
    )

    model_config = self._derive_shapes(text_config)

    # 断言 model_config 上的值为 128
    self.assertEqual(model_config.head_dim, 128)
    self.assertEqual(model_config.v_head_dim, 128)
    self.assertEqual(model_config.swa_head_dim, 128)
    self.assertEqual(model_config.swa_v_head_dim, 128)
    # 断言回写到 text_config 上的值也为 128
    self.assertEqual(text_config.head_dim, 128)
    self.assertEqual(text_config.v_head_dim, 128)
    self.assertEqual(text_config.swa_head_dim, 128)
    self.assertEqual(text_config.swa_v_head_dim, 128)

def test_explicit_head_dims_are_preserved(self):
    # 验证显式指定不同 head dim 值时能被保留，不应用默认值
    text_config = _make_text_config(
        head_dim=128,
        v_head_dim=96,
        swa_head_dim=64,
        swa_v_head_dim=48,
    )

    model_config = self._derive_shapes(text_config)

    self.assertEqual(model_config.head_dim, 128)
    self.assertEqual(model_config.v_head_dim, 96)
    self.assertEqual(model_config.swa_head_dim, 64)
    self.assertEqual(model_config.swa_v_head_dim, 48)

if __name__ == "__main__":
    unittest.main()

```

评论区精华

该 PR 没有 review 评论。从原始 PR#23862 的讨论看 ([discussion r3416313980](#))，原有 revert 的理由可能是对 setattr 回写的安全性有所担忧，但本次 PR 认为该逻辑是必要的，且新增测试可保证正确性。

- 暂无高价值评论线程

风险与影响

- 风险：核心风险在于 head_dim 回写 hf_text_config 可能导致后续模块重复获取到意外的属性值，但新增的显式赋值测试覆盖了最常见路径。对于 MLA 架构的模型（如 DeepSeek v2/v3），后续仍然会通过 FIXME 分支覆盖 head_dim 和其他参数，所以影响有限。
- 影响：直接影响所有模型在 _derive_model_shapes 阶段的 head_dim 派生逻辑，间接影响依赖 hf_text_config.head_dim 等属性的下游模块（如 attention 层、量化层）。影响范围属于 SRT 代码的核心配置路径，但回写操作本身是幂等的。
- 风险标记：重新应用先前已合并变更，核心路径变更，新增测试覆盖

关联脉络

- PR #28571 revert the head_dim assignment from PR 23862: 本 PR revert 了 #28571，即还原了该 revert，恢复 PR#23862 的变更。
- PR #23862 head_dim assignment in model_config: 原始 PR，本 PR 重新应用其 head_dim 赋值逻辑。