

PR #32457 完整报告

sgl-project/sglang

[Model] Serve bare Qwen3Model backbone natively as an embedding model

合并时间: 2026-07-27 15:49

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32457>

执行摘要

- 一句话: bare Qwen3Model 原生嵌入适配, 吞吐提升 32%
- 推荐动作: PR 质量高, 包含详尽的性能与准确性测试。实现模式 (包装骨干 + Pooler + 手动 prefix remap) 可供其他 embedding 模型注册参考。建议合并。

功能与动机

Checkpoints 导出为 architectures=['Qwen3Model'] 时, 由于没有原生实现, ModelRegistry 会回退到 Transformers 后端, 导致吞吐受限 (profile 显示 GPU 时间被 copy 操作主导, GEMM 仅占 14%)。此 PR 使其运行在 SGLang 原生融合 Qwen3 内核上, 显著提升性能。

实现拆解

1. 新增模型文件 `python/sglang/srt/models/qwen3_embedding.py`: 定义 Qwen3Model 类继承 `nn.Module`, 包装 Qwen3TransformerModel。构造函数中初始化 backbone 和 Pooler(pooling_type=PoolingType.LAST, normalize=True)。
2. 权重加载 `load_weights`: 处理裸 backbone 检查点缺少 model. 前缀的情况, 将 layers.、embed_tokens.、norm. 重映射到 model.*; 跳过 rotary_emb、lm_head 和无关张量; 处理 stacked params mapping (qkv_proj, gate_up_proj)。
3. 白名单 `python/sglang/srt/configs/model_config.py`: 在 `is_generation_model()` 函数中添加 'Qwen3Model' 到非生成模型列表, 一行改动。
4. 单元测试 `test/registered/unit/models/test_qwen3_embedding_registration.py`: 四个 CPU 测试覆盖 EntryClass 身份、注册解析、非生成分类、现有架构无影响。
5. 文档 `docs_new/docs/supported-models/embedding_models.mdx`: 在支持嵌入模型列表中添加 'Qwen3 (bare backbone)' 条目。

关键文件:

- `python/sglang/srt/models/qwen3_embedding.py` (模块 嵌入层; 类别 source; 类型 data-contract; 符号 Qwen3Model, init, forward, load_weights): 核心新增文件, 实现 Qwen3Model 类, 包含权重加载逻辑和 Pooler 配置。
- `test/registered/unit/models/test_qwen3_embedding_registration.py` (模块 测试; 类别 test; 类型 test-coverage; 符号 TestQwen3ModelEmbeddingRegistration, test_entry_class_is_native_qwen3model, test_registry_resolves_native_not_transforme

rs_fallback, test_bare_qwen3model_classified_as_embedding)：单元测试覆盖注册和分类验证，确保正确性和无回归。

- python/sglang/srt/configs/model_config.py（模块 模型配置；类别 source；类型 data-contract）：一行白名单使 Qwen3Model 架构归类为 embedding 模型。
- docs_new/docs/supported-models/embedding_models.mdx（模块 文档；类别 docs；类型 data-contract）：文档更新，列出 bare Qwen3Model 支持。

关键符号：Qwen3Model.init, Qwen3Model.forward, Qwen3Model.load_weights, Qwen3Model.get_input_embeddings

关键源码片段

test/registered/unit/models/test_qwen3_embedding_registration.py

单元测试覆盖注册和分类验证，确保正确性和无回归。

```
"""Unit tests for native registration of the bare ``Qwen3Model`` embedding arch.

Checkpoints such as ``microsoft/harrier-oss-v1-0.6b`` declare
``architectures=["Qwen3Model"]`` (a bare Qwen3 backbone). These must resolve to
the native SGLang implementation (``sglang.srt.models.qwen3_embedding.Qwen3Model``)
and be served as an embedding model, NOT fall back to the Transformers backend.
"""

from sglang.test.ci.ci_register import register_cpu_ci

register_cpu_ci(est_time=10, suite="base-a-test-cpu")

import unittest

from sglang.srt.configs.model_config import is_generation_model
from sglang.test.test_utils import CustomTestCase

class TestQwen3ModelEmbeddingRegistration(CustomTestCase):
    def test_entry_class_is_native_qwen3model(self):
        """验证 EntryClass 的 __name__ 和 __module__ 均指向原生实现"""
        from sglang.srt.models import qwen3_embedding
        entry = qwen3_embedding.EntryClass
        self.assertEqual(entry.__name__, "Qwen3Model")
        self.assertEqual(
            entry.__module__, "sglang.srt.models.qwen3_embedding"
        )
        # 确保具有 forward 和 load_weights 方法
        self.assertTrue(hasattr(entry, "forward"))
        self.assertTrue(hasattr(entry, "load_weights"))

    def test_registry_resolves_native_not_transformers_fallback(self):
        """ModelRegistry.resolve_model_cls 应返回原生类，而非回退"""
```

```

from sglang.srt.models.registry import ModelRegistry
model_cls, resolved_arch = ModelRegistry.resolve_model_cls("Qwen3Model")
self.assertEqual(resolved_arch, "Qwen3Model")
self.assertEqual(model_cls.__name__, "Qwen3Model")
self.assertEqual(
    model_cls.__module__, "sglang.srt.models.qwen3_embedding"
)
self.assertNotIn("Transformers", model_cls.__name__)

def test_bare_qwen3model_classified_as_embedding(self):
    """无论 --is-embedding 标志如何，该架构均被视为非生成模型"""
    self.assertFalse(is_generation_model(["Qwen3Model"]))
    self.assertFalse(is_generation_model(["Qwen3Model"], is_embedding=False))
    self.assertFalse(is_generation_model(["Qwen3Model"], is_embedding=True))

def test_existing_qwen3_archs_unaffected(self):
    """确保已有的 Qwen3 架构（如 Qwen3ForCausalLM）行为不变"""
    # Qwen3ForCausalLM 默认为生成模型，仅在 --is-embedding 下变为嵌入
    self.assertTrue(is_generation_model(["Qwen3ForCausalLM"]))
    self.assertTrue(
        is_generation_model(["Qwen3ForCausalLM"], is_embedding=False)
    )
    self.assertFalse(
        is_generation_model(["Qwen3ForCausalLM"], is_embedding=True)
    )
    # Sequence classification / reward 类始终为非生成
    self.assertFalse(
        is_generation_model(["Qwen3ForSequenceClassification"])
    )
    self.assertFalse(is_generation_model(["Qwen3ForRewardModel"]))

```

评论区精华

PR 没有公开 review 讨论；maintainer mickqian 直接批准。作者在 body 的 notes 中提到了两个值得关注的设计决策：pooler 硬编码为 LAST+L2（与 harrier/Qwen3-Embedding 一致）以及缺少 PP 支持。作者建议将自动 pooling 选择作为后续跨架构改进。

- Pooling 策略是否应自动适配配置 (design)：当前保持硬编码，留作后续改进。

风险与影响

- 风险：兼容性风险：新增文件不影响现有 Qwen3ForCausalLM 路径。白名单仅作用于架构字符串 'Qwen3Model'，不会误伤其他架构。但如果未来有生成模型也使用此架构名，则会被错误分类为 embedding，不过这不符合 Hugging Face 惯例（backbone 用于 embedding）。性能风险：原生路径更快，无退化。测试覆盖：单元测试验证注册逻辑，但缺少端到端集成测试（例如启动服务器并调用 /v1/embeddings）。weight loading 针对标准 bare backbone 检查点，但某些变体（如使用 'model.' 前缀的）也能处理（因为 load_weights 会检查前缀）。

- 影响：直接影响使用 microsoft/harrier-oss-v1-0.6b 等 arch=Qwen3Model 的用户：吞吐提升 32%，精度保持不变。对现有 Qwen3 用户无影响。文档更新有助于用户发现该支持。
- 风险标记：新架构注册，权重前缀映射假设，缺少集成测试

关联脉络

- PR #32375 model: support EmbeddingGemma: 同为 embedding 模型注册，涉及 model_config 白名单和 Pooler 使用，可对比做法。