量化是一种以参数或计算中间结果精度下降换空间节省（以及同时带来的性能提升）的策略。对模型进行量化。主要包括 KV Cache 量化和模型参数量化。

- KV Cache 量化是指将逐 Token（Decoding）生成过程中的上下文 K 和 V 中间结果进行 INT8 量化（计算时再反量化），以降低生成过程中的显存占用。
  - 计算 minmax。主要思路是通过计算给定输入样本在每一层不同位置处计算结果的统计情况。
  - 对于 Attention 的 K 和 V：取每个 Head 各自维度在所有Token的最大、最小和绝对值最大值。对每一层来说，上面三组值都是 `(num_heads, head_dim)` 的矩阵。这里的统计结果将用于本小节的 KV Cache。
  - 对于模型每层的输入：取对应维度的最大、最小、均值、绝对值最大和绝对值均值。每一层每个位置的输入都有对应的统计值，它们大多是 `(hidden_dim, )` 的一维向量，当然在 FFN 层由于结构是先变宽后恢复，因此恢复的位置维度并不相同。这里的统计结果用于下个小节的模型参数量化，主要用在缩放环节
- 4bit Weight 量化，将 FP16 的模型权重量化为 INT4，Kernel 计算时，访存量直接降为 FP16 模型的 1/4，大幅降低了访存成本。
- Weight Only 是指仅量化权重，数值计算依然采用 FP16（需要将 INT4 权重反量化）。

1 创建环境

```
# 创建环境
conda create -n lmdeploy --clone /share/conda_envs/internlm-base

# 激活环境
conda activate lmdeploy

# 安装packaging
pip install packaging

# 安装flash_attn
pip install /root/share/wheels/flash_attn-2.4.2+cu118torch2.0cxx11abiTRUE-cp310-cp310-linux_x86_64.whl

# 安装lmdeploy
pip install 'lmdeploy[all]==v0.1.0'
```

- 创建conda环境

```
(base) root@intern-studio:/opt/jupyterlab# conda create -n lmdeploy --clone /share/conda_envs/internlm-base
Retrieving notices: ...working... done
Source:      /share/conda_envs/internlm-base
Destination: /root/.conda/envs/lmdeploy
Packages: 96
Files: 0

Downloading and Extracting Packages:


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate lmdeploy
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) root@intern-studio:/opt/jupyterlab# conda activate lmdeploy
```

【问题1】缺少packaging包

```
pip install packaging
```

```
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/4e/2f/a8d5495cd020e70474095107f0519a11fdff81a2a6cc15755feb1e19b027/
flash_attn-2.4.2.tar.gz (2.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 46.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... error
  error: subprocess-exited-with-error

  ×python setup.py egg_info did not run successfully.
  │ exit code: 1
  ╰─> [6 lines of output]
      Traceback (most recent call last):
        File "<string>", line 2, in <module>
        File "<pip-setuptools-caller>", line 34, in <module>
        File "/tmp/pip-install-xi8op62o/flash-attn_2adc49d6bdb54d3abeec8381738f3bff/setup.py", line 9, in <module>
          from packaging.version import parse, Version
      ModuleNotFoundError: No module named 'packaging'
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
error: metadata-generation-failed

×Encountered error while generating package metadata.
╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
```

【问题2】安装速度过慢

```
pip install /root/share/wheels/flash_attn-2.4.2+cu118torch2.0cxx11abiTRUE-cp310-cp310-linux_x86_64.whl
```

```
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/d9/66/48866fc6b158c81cc2bfecc04c480f105c6040e8b077bc54c634b4a67926
/zipp-3.17.0-py3-none-any.whl (7.4 kB)
Collecting jsonschema-specifications>=2023.03.6 (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio<4.0.0->lmdeploy[all]==v0.1
.0)
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/ee/07/44bd408781594c4d0a027666ef27fab1e441b109dc3b76b4f836f8fd04fe
/jsonschema_specifications-2023.12.1-py3-none-any.whl (18 kB)
Collecting referencing>=0.28.4 (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio<4.0.0->lmdeploy[all]==v0.1.0)
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/14/2a/0a9f649354cd2d40f6c4f16eadabd9727377e3b9bc2ccec6cb630d9a6765
/referencing-0.32.1-py3-none-any.whl (26 kB)
Collecting rpds-py>=0.7.1 (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio<4.0.0->lmdeploy[all]==v0.1.0)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/8c/f1/09bee4d70305e79ecad4f3ccee583f0185c06a5f58befdc3544cf8b18536/
rpds_py-0.17.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 25.9 MB/s eta 0:00:00
Collecting mdurl~=0.1 (from markdown-it-py>=2.2.0->rich->mmengine-lite->lmdeploy==v0.1.0->lmdeploy[all]==v0.1.0)
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/b3/38/89ba8ad64ae25be8de66a6d463314cf1eb366222074cfda9ee839c56a4b4
/mdurl-0.1.2-py3-none-any.whl (10.0 kB)
Building wheels for collected packages: fire, flash-attn
  Building wheel for fire (setup.py) ... done
  Created wheel for fire: filename=fire-0.5.0-py2.py3-none-any.whl size=116934 sha256=cb3582bf654082929e77b49453bf15fbf4cefb
b0debaea9ef3163b9d0075c1ae
  Stored in directory: /root/.cache/pip/wheels/1f/b3/61/733f76a36386b7131a22a3eab4b92741e3ee75a9ed2a8f8460
  Building wheel for flash-attn (setup.py) ... ^C^Z
[1]+  Stopped                 pip install 'lmdeploy[all]==v0.1.0'
```

一直卡在这里，安装速度过慢

2 模型转换

使用 TurboMind 推理模型需要先将模型转化为 TurboMind 的格式，目前支持在线转换和离线转换两种形式。

在线转换可以直接加载 Huggingface 模型，离线转换需需要先保存模型再加载。以下以离线转换为例。

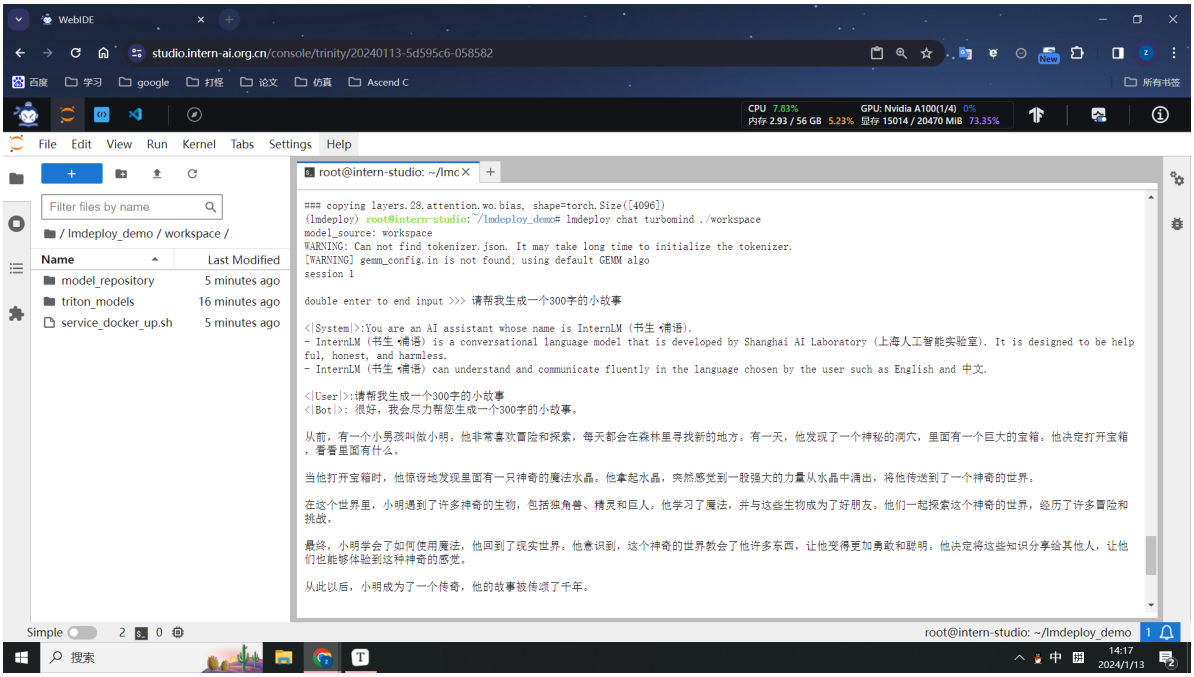
```
cd /root/
mkdir lmdeploy_demo && cd lmdeploy_demo
lmdeploy convert internlm-chat-7b  /root/share/temp/model_repos/internlm-chat-7b/
```

```
*** splitting layers.28.feed_forward.w1.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.28.feed_forward.w3.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.28.feed_forward.w2.weight, shape=torch.Size([11008, 4096]), split_dim=0, tp=1
*** splitting layers.29.attention.w_qkv.weight, shape=torch.Size([4096, 12288]), split_dim=-1, tp=1
*** splitting layers.29.attention.wo.weight, shape=torch.Size([4096, 4096]), split_dim=0, tp=1
*** splitting layers.29.attention.w_qkv.bias, shape=torch.Size([1, 12288]), split_dim=-1, tp=1
### copying layers.29.attention.wo.bias, shape=torch.Size([4096])
*** splitting layers.29.feed_forward.w1.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.29.feed_forward.w3.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.29.feed_forward.w2.weight, shape=torch.Size([11008, 4096]), split_dim=0, tp=1
*** splitting layers.30.attention.w_qkv.weight, shape=torch.Size([4096, 12288]), split_dim=-1, tp=1
*** splitting layers.30.attention.wo.weight, shape=torch.Size([4096, 4096]), split_dim=0, tp=1
*** splitting layers.30.attention.w_qkv.bias, shape=torch.Size([1, 12288]), split_dim=-1, tp=1
### copying layers.30.attention.wo.bias, shape=torch.Size([4096])
*** splitting layers.30.feed_forward.w1.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.30.feed_forward.w3.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.30.feed_forward.w2.weight, shape=torch.Size([11008, 4096]), split_dim=0, tp=1
*** splitting layers.31.attention.w_qkv.weight, shape=torch.Size([4096, 12288]), split_dim=-1, tp=1
*** splitting layers.31.attention.wo.weight, shape=torch.Size([4096, 4096]), split_dim=0, tp=1
*** splitting layers.31.attention.w_qkv.bias, shape=torch.Size([1, 12288]), split_dim=-1, tp=1
### copying layers.31.attention.wo.bias, shape=torch.Size([4096])
*** splitting layers.31.feed_forward.w1.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.31.feed_forward.w3.weight, shape=torch.Size([4096, 11008]), split_dim=-1, tp=1
*** splitting layers.31.feed_forward.w2.weight, shape=torch.Size([11008, 4096]), split_dim=0, tp=1
Convert to turbomind format: 100%|[redacted]| 32/32 [00:15<00:00,  2.09it/s]
```

执行完成后将会在当前目录生成一个 `workspace` 的文件夹。

3 命令行本地对话

```
lmdeploy chat turbomind ./workspace
```



4 API服务

```
lmdeploy serve api_server ./workspace \
    --server_name 0.0.0.0 \
    --server_port 23333 \
    --instance_num 64 \
    --tp 2
```

`server_name` 和 `server_port` 分别表示服务地址和端口；tp表示 Tensor 并行；`instance_num` 参数，表示实例数。

```
lmdeploy serve api_client http://localhost:23333

ssh -CNg -L 23333:127.0.0.1:23333 root@ssh.intern-ai.org.cn -p [端口号]
```

5 网页Gradio

```
lmdeploy serve gradio http://0.0.0.0:23333 \
    --server_name 0.0.0.0 \
    --server_port 6006 \
    --restful_api True
```