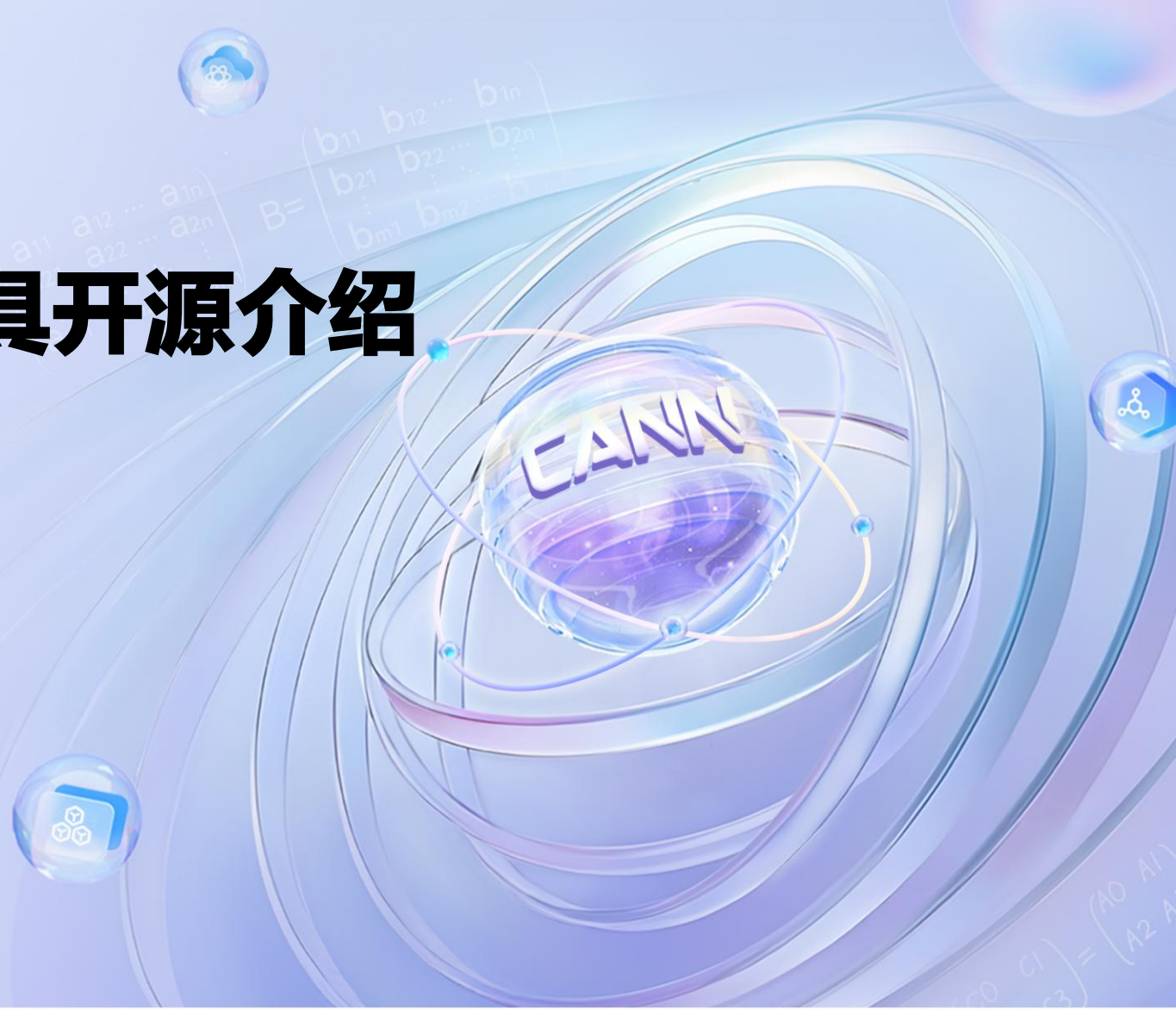
AMCT模型压缩工具开源介绍

作者：姚广修

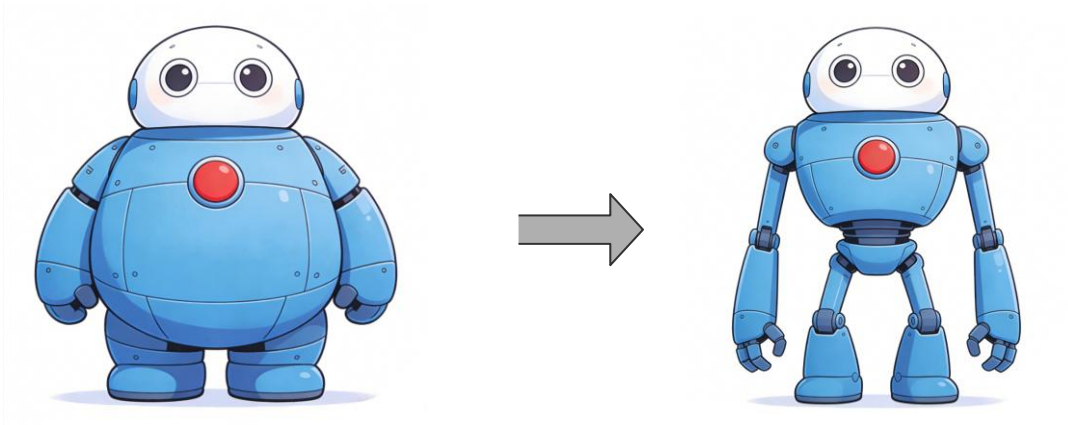
时间：2025/12/24



CANN

目录

AMCT工具简介



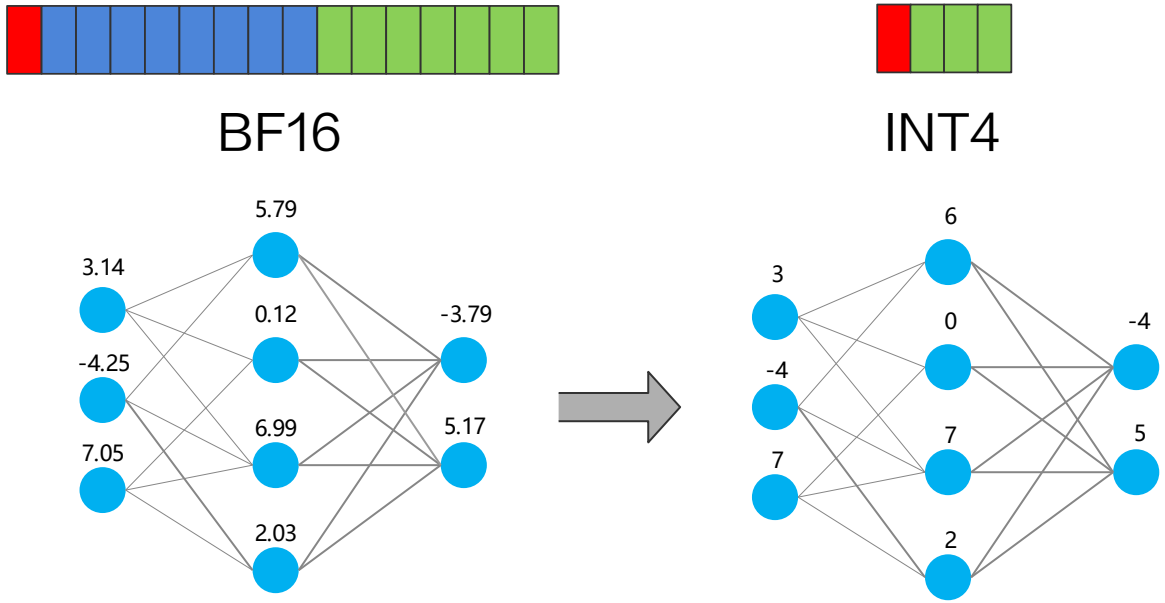
为了让模型变得越来越聪明，我们使用更多的数据训练出更大的模型。DeepSeekV3、Qwen3等业界标杆模型，包含数千亿个参数。这一方面使的它们的功能强大，但同时也让它们运行缓慢，成本高昂，难以在普通设备上使用。

AMCT全称Ascend Model Compression Toolkit，是亲和昇腾处理器的模型压缩工具包。

致力于通过量化、稀疏等有效的手段，在保留模型能力的同时，让臃肿的模型“瘦下来”，让迟缓的模型“跑的快”。

AMCT工具简介

量化Quantization是模型压缩中一种常用的技术，用于将高精度的权重和激活数据转换为低精度数据。



减小模型尺寸
INT4数据位宽是BF16的1/4，显存占用节省75%。使用更少的硬件资源即可完成模型部署。

降低功耗
对AI加速硬件来说，整数计算的功耗更低。对功耗敏感的设备来说，量化提升的能效至关重要。

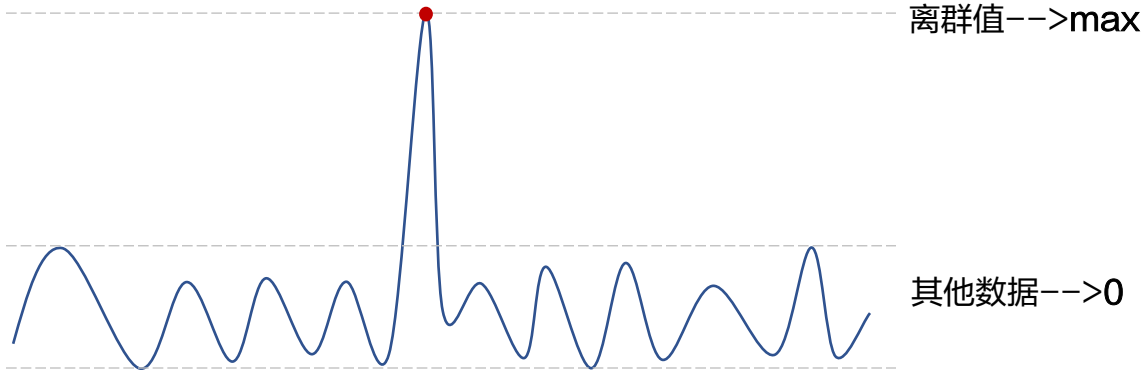
推理速度更快
通过量化将浮点数据转换成整数，可以更快完成计算。减少模型推理的整体时延。

兼容性
功耗敏感的端侧设备只支持整数计算，量化模型对这些平台的兼容性更好。

AMCT工具简介

量化公式 $x_{\mathrm{q}} = \mathrm{clip}(\mathrm{round}\left(\frac{x}{s} + z\right), \alpha_q, \beta_q)$

其中，
x是高精度数据；x_{q}是量化后的数据；s是量化缩放因子；
z是量化偏移因子；α_q, β_q是量化上下限范围



量化优化&权衡

量化效果与数据分布紧密相关。

如果原始数据中存在outlier离群数据，直接量化因为outlier和其他数据共享缩放系数，导致离群值被量化到最大值，其他数据量化值为0，丢失了绝大部分数据信息，往往会造成较大的精度损失。

量化算法算法通过调整数据分布或调整量化方式，使数据分布尽可能均匀，减少量化带来的信息损失，提升最终的压缩效果。

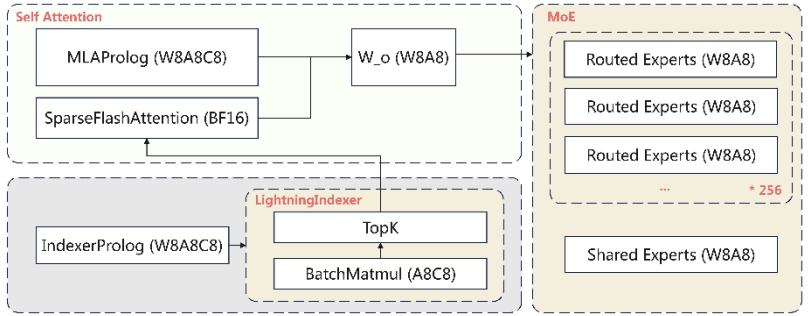
实际开发过程中需要在模型精度和模型性能之间权衡。

1. 根据先验知识识别到模型中的量化敏感层（e.g. 首尾层），不做量化压缩。
2. 识别模型中量化性能收益小的层（e.g. shape小的层），不做量化压缩。

✓ AMCT工具将min-max/gptq/awq/smooth_quant等优化算法集成在工具中，并提供典型量化配置，开发者可以拿来即用，满足不同场景的使用述求。

CANN

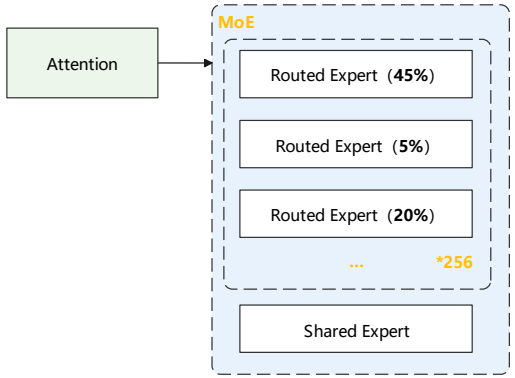
AMCT工具简介



DeepSeek-V3.2-Exp量化部署场景。

mla：只量化部分层，q_b_proj、w_o_proj层
indexer：只量化部分层，wq_b_proj层
moe：W8A8量化；
lm_head：不量化。

量化的选择和融合算子的设计方式相关，IndexerProlog融合算子设计成weight_proj出fp16，所以MLA输入关联的Linear统一不做量化。



量化需要统计数据的分布情况。

对MoE多专家模型来说，每个专家在模型中的热度是不一样的，通过整网校准推理来完整覆盖全部专家模块，需要大量校准数据，极大降低了量化校准的效率。

逐block量化算法，将模型拆分成小的量化模块单元，每个量化模块都可以单独执行量化校准过程。

这些量化模块可以分配在不同的NPU卡上并行执行，大大提升了开发效率。

即使开发者只有一张NPU卡，通过分块执行的方式也可以完成deepseekv3.2 这种尺寸规模的模型量化。

✓ AMCT工具提供逐block量化算法，单卡即可完成deepseekV3.2大尺寸模型量化校准

CANN

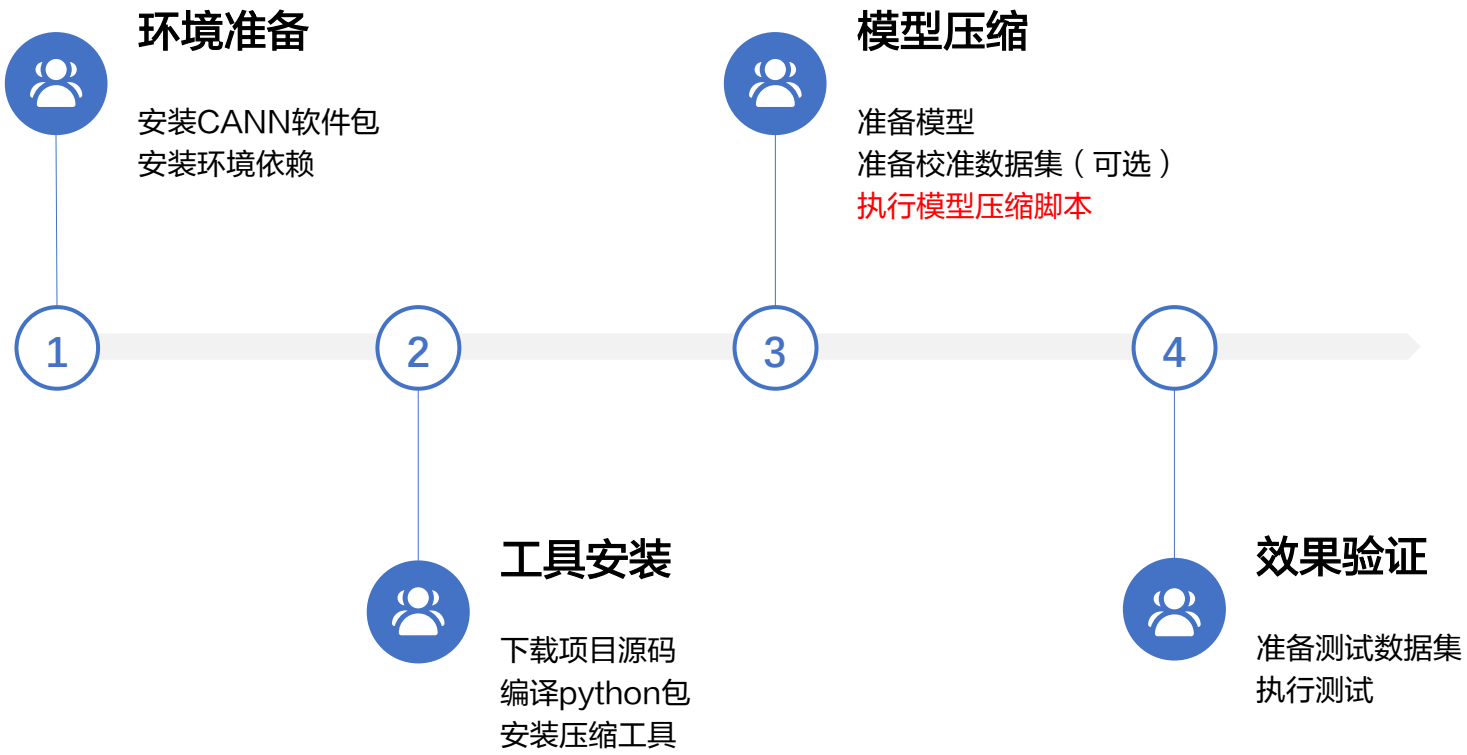
AMCT特性规划

- ✓ **低比特量化**
  - 可学习转换矩阵优化
  - 低比特A4W4的支持
  - 丰富benchmark
  - ……
- ✓ **易用性提升**
  - 一键完成校准流程
  - 对接Hugging Face模型
  - 对接常用推理框架
  - ……

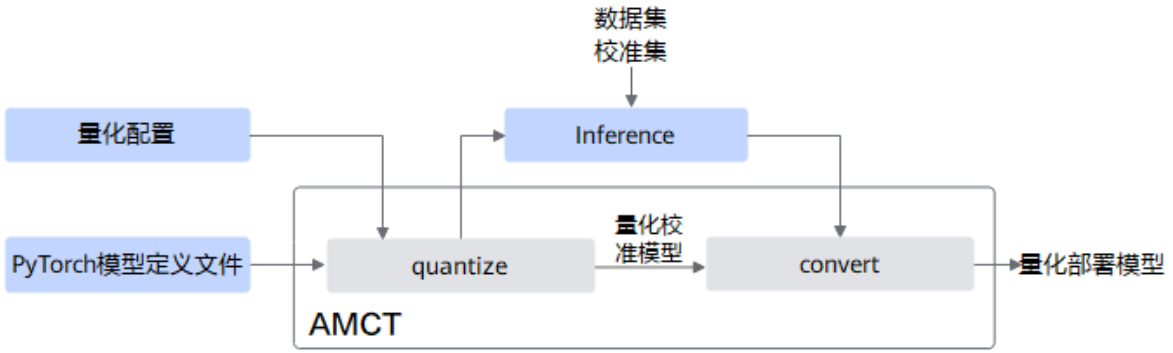
目录

AMCT工具使用

✓ 工具使用流程



AMCT工具使用



✓ 工具接口

quantize(model, config)　　1、**模型量化**：根据量化配置，将待量化模型转换为量化校准模型。校准模型经过推理，计算得到量化参数。

convert(model)　　2、**模型转换**：将量化校准之后的模型，转换为量化部署模型。

algorithm_register(name, src_op, quant_op, deploy_op)　　3、**算法注册**：用户将自定义算法注册到AMCT工具中。

CANN

AMCT工具使用

✓ 使用样例

- 模型： Qwen3-8B
- 校准数据： mit-han-lab/pile-val-backup
- 测试数据： mindchain/wikitext2
- 目标：W4量化

以Qwen网络为例，用户只需要在脚本中添加几行代码，即可完成量化操作。

1. 加载模型和数据集；
2. 量化配置：选择合适的算法并生成量化校准模型；
3. 校准数据：使用选定的数据集做校准，得到量化结果；
4. 模型转换：校准模型转换为部署模型，获取量化收益；
5. 对量化模型进行测试验证

```
torch_npu.npu.set_device(0)

# Phase1: prepare test model & data
model = get_test_model('Qwen/Qwen3-8B')                                          1
calibration_data = get_test_data('mit-han-lab/pile-val-backup').npu()

# Phase2: quantize model
cfg = amct.INT4_AWQ_WEIGHT_QUANT_CFG                                             2
amct.quantize(model, cfg)

# Phase3: do weights calibration and generate calibration model
with torch.no_grad():                                                            3
    model(calibration_data[:, :model.seqlen].to(next(model.parameters()).device))

# Phase4: convert to deploy model
amct.convert(model)                                                              4

# Phase5: model test
test_data = get_test_data('mindchain/wikitext2').npu()                           5
result = ppl_test(model, test_data)
print(f'quantize ppl: {result}')
```

量化压缩

CANN

AMCT工具使用

✓ 使用样例

量化配置

```
INT4_AWQ_WEIGHT_QUANT_CFG = {
    'batch_num': 1,
    'quant_cfg': {
        'weights': {
            'type': 'int4',
            'symmetric': True,
            'strategy': 'channel'
        },
    },
    'algorithm': {'awq': {'grids_num': 20}},
    'skip_layers': {'lm_head'}
}
```

校准数据
1. 校准数据是在量化过程中为获取激活数据的分布准备的小规模数据集

量化参数
1. 量化数据类型，更低比特的量化能够获取更大的性能收益，但量化精度损失通常越大
2. 量化粒度选择，支持per-tensor/per-channel/per-group量化，粒度更细的量化通常会有更高的精度收益，但是会引入更多的量化参数，性能往往会下降

量化算法
1. 不同是使用场景需要不同的量化配置，比如仅权重量化可以选择awq，激活权重都量化可以选择smoothquant。用户还可以根据量化网络的实际情况选择不同的算法参数。

跳过量化敏感层
1. 网络中存在量化敏感层，这些层经过量化会导致严重的精度劣化，需要跳过这些层的量化来保持整网的量化精度损失在可接受范围内。

AMCT工具使用

✓ 开源贡献

```
├── experimental                          # 实验特性
│   ├── sample1                           # 特性名称（如custom_defined_quantize_alg）
│   │   ├── doc                           # doc目录：存放优化文档、配图等
│   │   ├── src                           # src目录：存放代码
│   │   ├── README.md                     # 上述提到的README文档
│   │   └── ...                           # 其他必要文件（如环境配置文件等）
│   ├── sample2
│   └── ...
```

- 欢迎到仓库提交issue/PR，复现、捉虫、贡献**压缩算法**，和广大开发者交流讨论。
  我们在开源代码中规划了experimental的目录，在我们达成方案共识后，提交PR将特性合入到amct代码仓。
- 欢迎参加sig双周例会，反馈您的述求和建议。

CANN

欢迎到仓库提交issue/PR

欢迎通过SIG联系我们

https://gitcode.com/cann

CANN

Thank you.

社区愿景：打造开放易用、技术领先的AI算力新生态

社区使命：使能开发者基于CANN社区自主研究创新，构筑根深叶茂、跨产业协同共享共赢的CANN生态

Vision: Building an Open, Easy-to-Use, and Technology-leading AI Computing Ecosystem

Mission: Enable developers to independently research and innovate based on the CANN community and build a win-win CANN ecosystem with deep roots and cross-industry collaboration and sharing.

上CANN社区获取干货

关注CANN公众号获取资讯

https://gitcode.com/cann

CANN