

TensorCIM: Digital Computing-in-Memory Tensor Processor With Multichip-Module-Based Architecture for Beyond-NN Acceleration 改进意见

支持动态及混合精度的稀疏张量乘法改进建议

当前的TensorCIM已经可以支持FP32/INT16/INT8等精度的计算，不过目前的精度是静态配置，无法实现不同精度的动态切换，同时目前的TensorCIM不支持混合精度计算。

关于动态精度的改进方式

可以在ILA-CIM的Booth控制器中加入精度模式寄存器，在运行时进行动态配置。精度切换的触发条件可以设定为增加检测电路来动态判断当前计算的稀疏度是否超过某个阈值，在被判定为较高稀疏度的情况下自动切换为FP32精度，在较低稀疏度的情况下切换为INT16或INT8。

这样可以实现在关键信息处保留FP32精度的同时，节省能耗，加快计算速度。

关于混合精度的改进方式

可以引入 *An 88.36TOPS/W Bit-Level-Weight-Compressed LLM Accelerator Cluster-Aligned INT-FP-GEMM and Bi-Dimensional Workflow Reformulation* 中的CAMP-PE机制，根据输入数据的指数分布进行精度划分分为多组，组内指数相近，将组内的指数对齐操作需要的移位操作作用于weight，同时设计分离权重和激活值的双向数据流，实现对混合精度计算的加速。

实际实验步骤和验证

1. 在ILA-CIM中加入精度控制寄存器，Booth多精度乘法路径，位宽选择器（用于精度切换），扩展位缓冲器（用于CAMP-PE的混合精度计算）。
2. 根据运行时的稀疏度等条件动态进行精度调整或者选用混合精度计算；

验证：对不同任务进行不同精度选择策略，对实际测试结果进行精度损失，能耗变化，计算速度等指标进行评估。

引入任务自适应的稀疏调度机制

当前的TensorCIM对于GCN和DLRM这种有结构的稀疏访问模型效果已经得到了验证，但是对于结构不规则或者结构相对动态变化的相关稀疏任务（如Transformer等），其性能如何尚不明确。

由于REGM依赖于固定的频繁访问特征阈值，无法适应动态稀疏模式；EOCI在初始化进行负载分配，无法考虑到运行时由于稀疏性变化导致的可能出现的负载不均的情况；ILA-CIM也未考虑块稀疏或者是极端不

规则访问的情况，所以对于不规则稀疏或者动态变化结构的稀疏任务，TensorCIM也许无法适应。

为了能够让TensorCIM适应各种不同的输入情况，需要设计一个根据任务和输入数据进行自适应的稀疏调度策略选择机制。

关于任务自适应的稀疏调度的改进建议

实验步骤

1. 可以加入稀疏度感知器，探知当前输入的稀疏情况。
2. 在ILA-CIM的预取单元里加入一个判断逻辑，当当前层的稀疏度低于某个阈值时，直接预取整个Tile，用来应对极端不规则访问的情况，减少频繁无效调度带来的能耗。
3. 在EOCI中增加可以控制Tile大小的逻辑，根据Tile的大小动态调整有效计算的ID的聚合粒度用来适应变化的稀疏度/不规则稀疏情况。

验证：在不规则模型比如transformer等条件下进行测试，分析与原来的TensorCIM在不规则模型/动态稀疏结构条件下相比，现在新电路的能耗，运行时间等指标的变化，分析能耗和执行速度的tradeoff是否可以接受，同时寻求最佳平衡点。