北京大学
PEKING UNIVERSITY

智能硬件体系结构

第十三讲：软硬件协同设计

主讲：陶耀宇、李萌

2024年秋季

注意事项

北京大学 PEKING UNIVERSITY

- **课程作业情况**

  - 未来**1次作业（12.15-12.30）**
    - AI芯片架构、软硬加协同优化
  - **Paper Review（12月25号/27号1-3点，每个同学10分钟）**
    - 时间有疑义请尽快联系老师或助教提出更换
    - 请自行阅读近5年内的体系结构相关论文1-2篇
    - 做8页左右的PPT，**每个同学10分钟演讲 + 1~2分钟答疑**
  - 第二次实验已经上传到课程网站，截止日期为**12月31号**

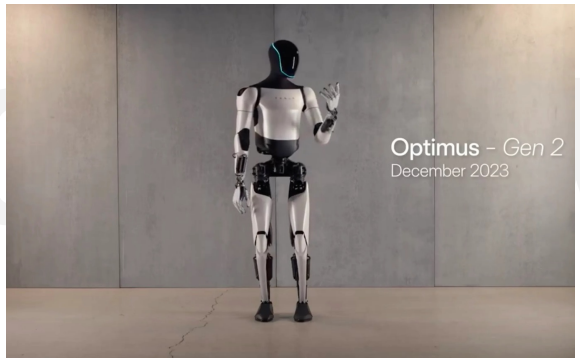
人工智能处理器芯片

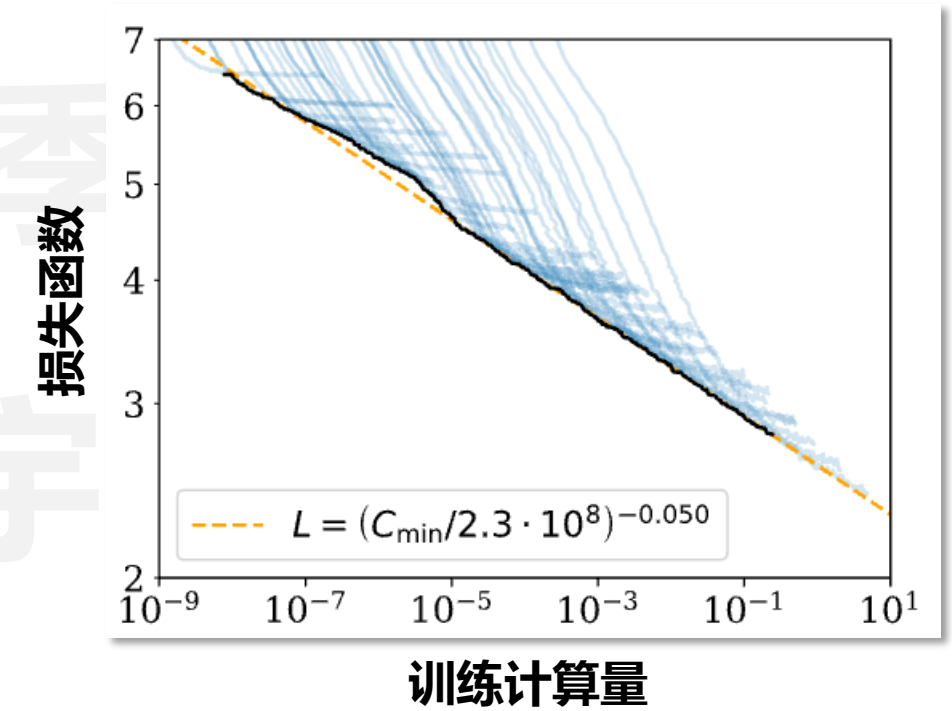
北京大学 PEKING UNIVERSITY

- 前面两次课，我们介绍了人工智能处理器芯片的基本组成和代表性架构
  - 基本组成：计算单元、访存、互连通信
  - 代表性架构：DianNao系列、Eyeriss系列、Google TPU系列
- 本节课我们将介绍如何通过**神经网络/处理器架构**的协同设计和优化，进一步提升处理器芯片效率
  - 神经网络模型**量化**与混合精度人工智能处理器
  - 神经网络模型**稀疏化**与人工智能处理器架构

人工智能模型的发展与挑战

- 过去10年，以深度学习为代表的人工智能飞速发展，广泛应用于视觉、语言、语音等领域
- 算力是人工智能快速发展的基础，也面临着更加严峻的挑战
- 特别是对于大模型时代，依据scaling law，算力需求呈现指数级增长



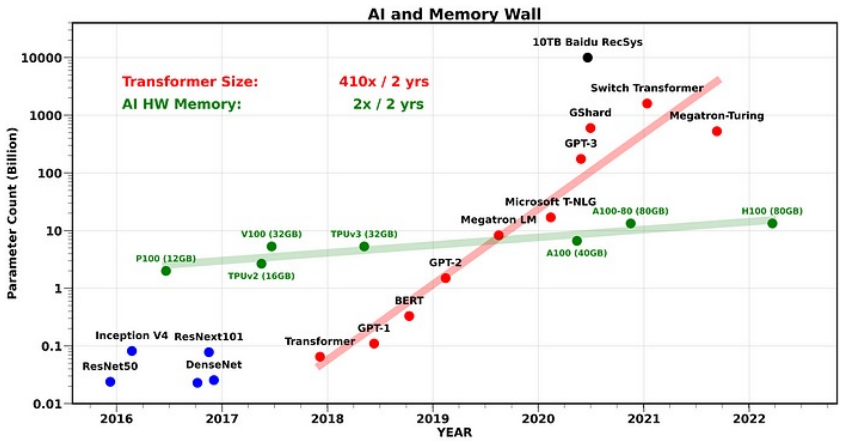


人工智能模型的发展与挑战

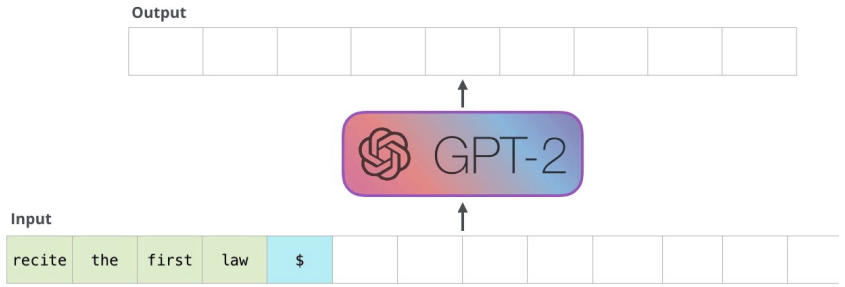
北京大学 PEKING UNIVERSITY

- 高效人工智能模型计算面临高存储、高带宽、动态计算图、部署平台复杂多样的挑战

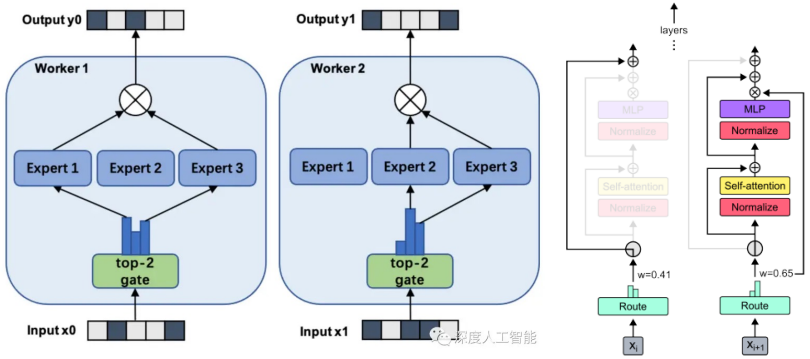
模型参数指数级增长，造成单模型**存储需求>1 TB**



大模型自回归解码，导致**平均访存带宽需求>1 TB/s**



动态计算图导致传统静态数据流优化失效

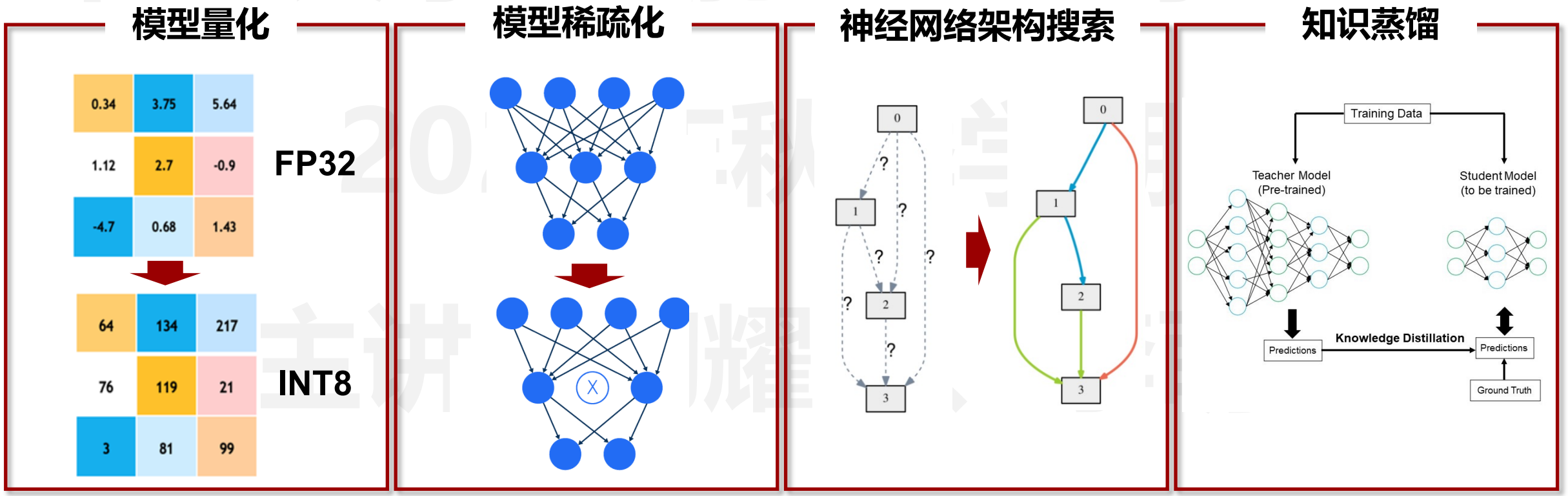


部署平台的**计算、存储**能力存在显著差距

	Cloud AI	Mobile AI	Tiny AI
Memory	24 GB	4 GB	500 KB
Storage	~ TB/PB	~100s GB	~1s MB
# params	>70 B	3 – 13 B	~1s M

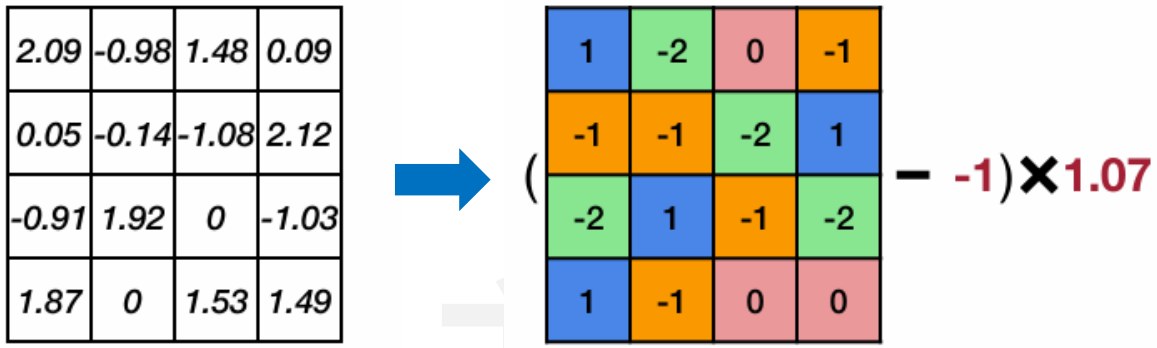
人工智能模型的发展与挑战

- 为了进一步提升人工智能模型的推理效率，采用算法和AI处理器协同设计和优化方法
  - 神经网络模型**量化**、**稀疏化**、神经网络架构搜索、知识蒸馏等
- 本次课我们将以神经网络模型量化、稀疏化为例，介绍算法和加速器的协同设计



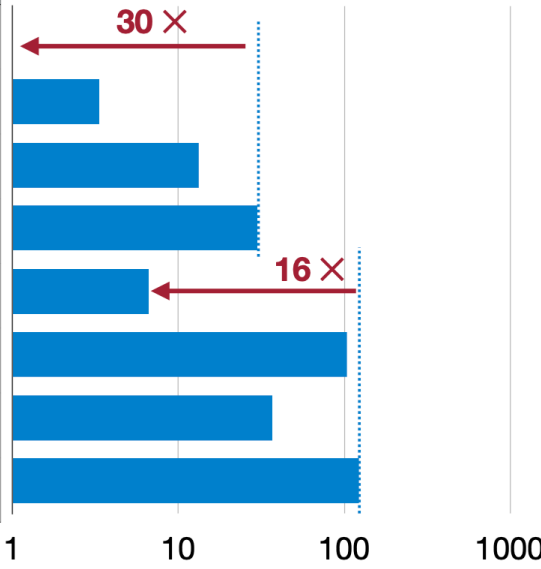
基于模型量化的软硬件协同设计

- 量化是将数据从**连续值集**映射为**离散值集**的过程
- 模型量化能够利用神经网络对于量化噪声的容错能力，显著提升模型推理计算效率
  - 无论是输入图像，还是神经网络本身都对于量化噪声具有较高的容忍能力



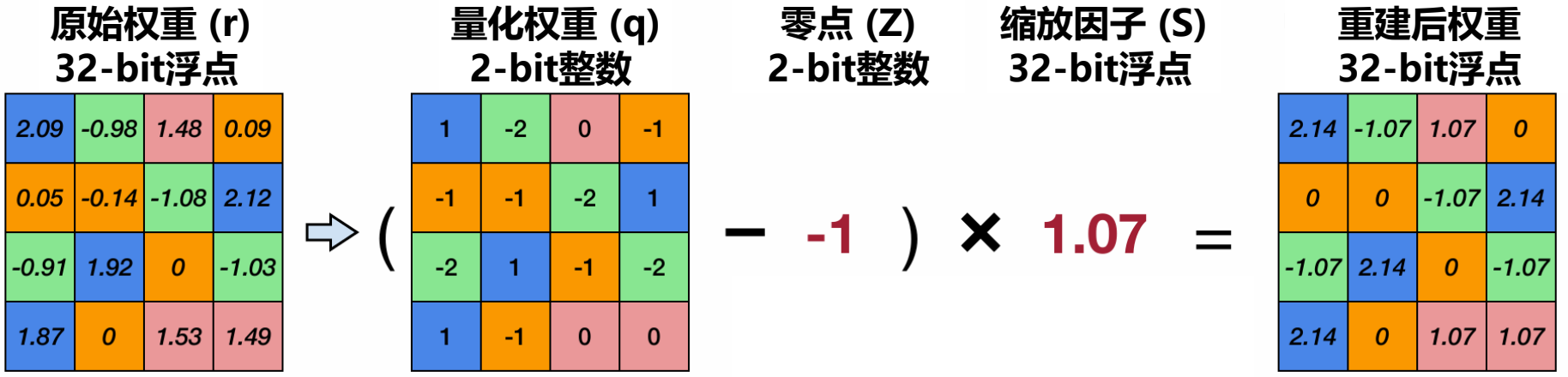
Operation	Energy [pJ]
8 bit int ADD	0.03
32 bit int ADD	0.1
16 bit float ADD	0.4
32 bit float ADD	0.9
8 bit int MULT	0.2
32 bit int MULT	3.1
16 bit float MULT	1.1
32 bit float MULT	3.7

Rough Energy Cost For Various Operations in 45nm 0.9V



量化神经网络基础

- 量化过程中的主要参数：缩放因子、零点、量化后数值
  - **静态**量化：量化参数在推理前确定，推理时保持不变
  - **动态**量化：量化参数在推理时计算得到



2-bit整数表示方法

Binary	Decimal
01	1
00	0
11	-1
10	-2

量化误差

-0.05	0.09	0.41	0.09
0.05	-0.14	-0.01	-0.02
0.16	-0.22	0	0.04
-0.27	0	0.46	0.42

- 权重的量化与激活值的量化对于计算的影响不同

$$Y = WX$$

$$S_Y(q_Y - Z_Y) = S_W(q_W - Z_W) \cdot S_X(q_X - Z_X)$$

$$q_Y = \frac{S_W S_X}{S_Y}(q_W - Z_W)(q_X - Z_X) + Z_Y$$

$$q_Y = \frac{S_W S_X}{S_Y}(q_W q_X - Z_W q_X - Z_X q_W + Z_W Z_X) + Z_Y$$

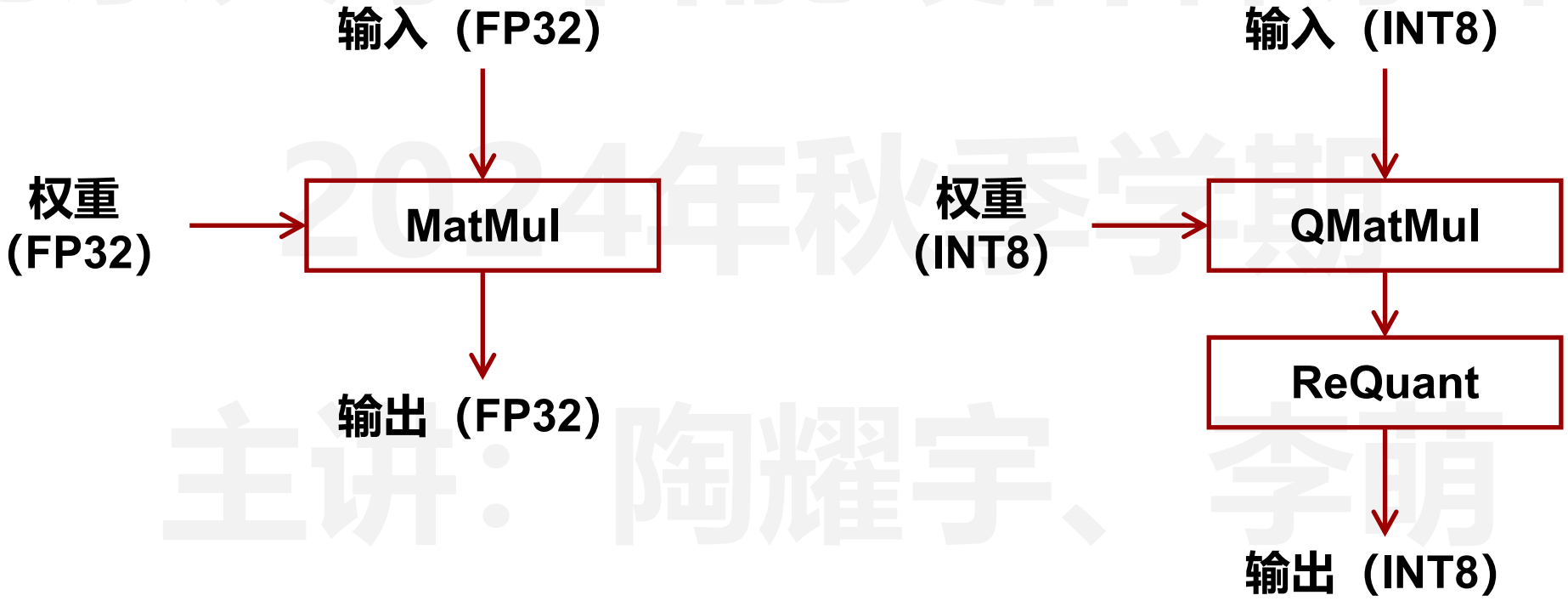
QMatMul

Z_w通常为0

提前计算

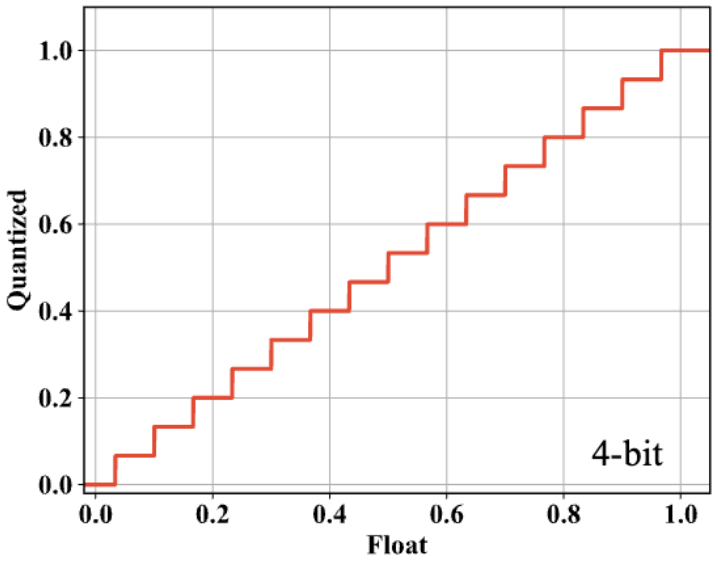
量化神经网络基础

- 权重的量化与激活值的量化对于计算的影响不同
- 量化神经网络推理：引入新的重量化计算，将高比特精度计算结果重量化为低比特精度输出
- 思考：QMatMul和MatMul的计算有什么区别?

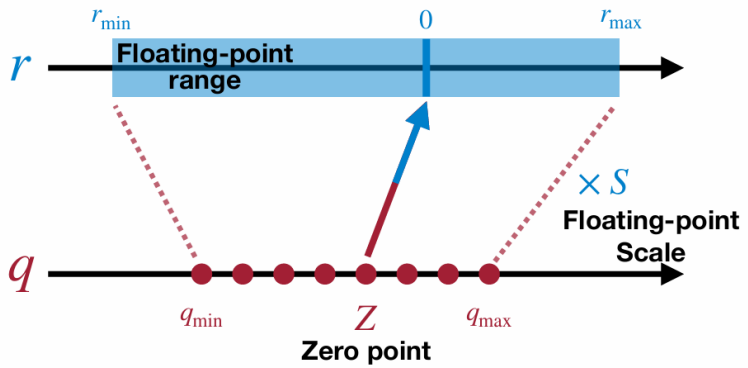


均匀量化与非均匀量化

- **均匀量化：输入到输出的映射为线性函数，因此均匀间隔的输入产生均匀间隔的输出**
  - **针对均匀量化，QMatMul和MatMul的计算，除比特精度外，基本相同**



(a) Uniform Quantization



$$\mathrm{r}_{max} = S\,(q_{max} - Z)$$

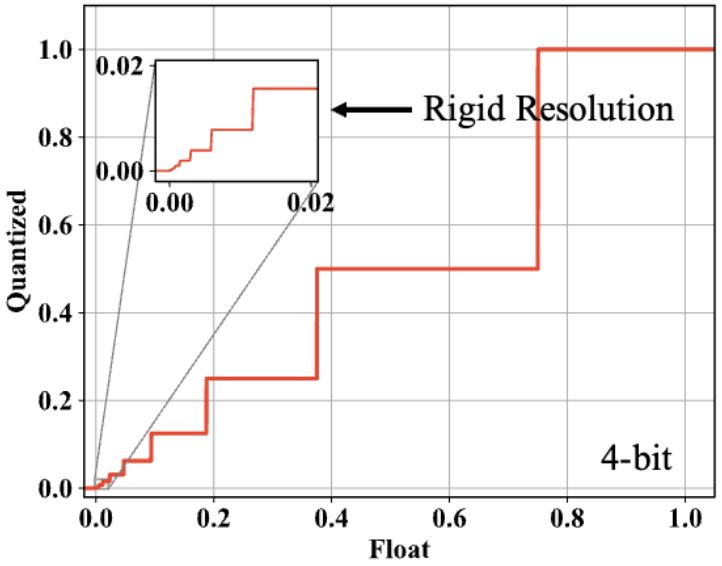
$$\mathrm{r}_{min} = S\,(q_{min} - Z)$$

$$\mathrm{r}_{max} - \mathrm{r}_{min} = S\,(q_{max} - q_{min})$$

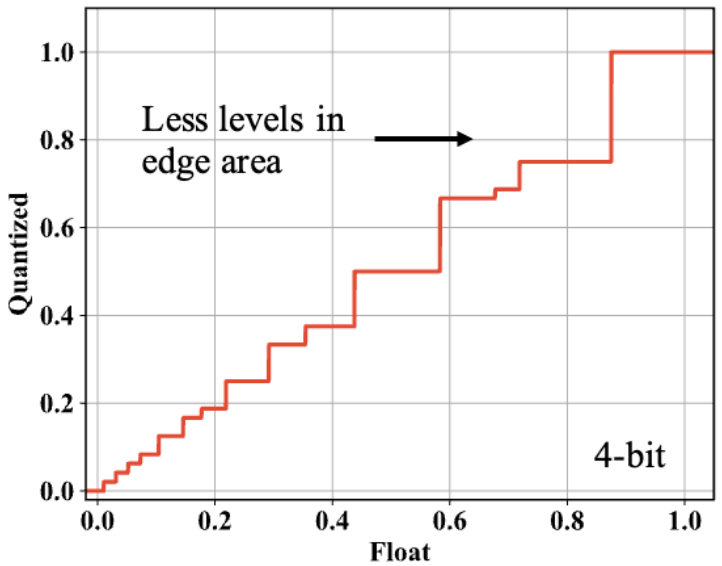
$$S = \frac{\mathrm{r}_{max} - \mathrm{r}_{min}}{q_{max} - q_{min}}$$

均匀量化与非均匀量化

- 均匀量化：输入到输出的映射为线性函数，因此均匀间隔的输入产生均匀间隔的输出
- 非均匀量化：输入到输出的映射为非线性函数，常见算法包括Power of Two（PoT）、APoT等
- 思考：均匀和非均匀量化还有哪些类别？



(b) Power-of-Two Quantization



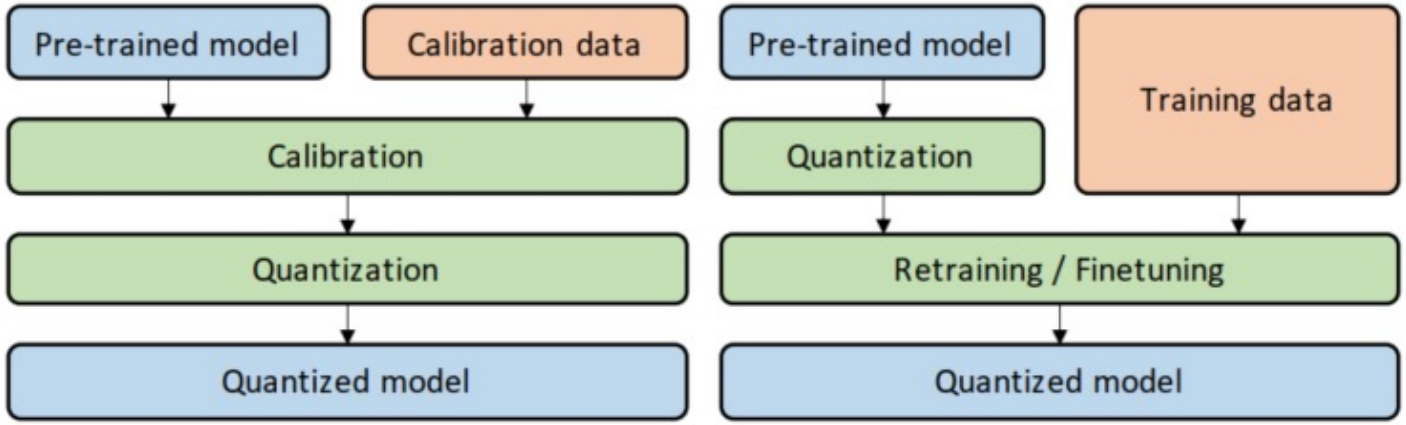
(c) Additive Power-of-Two Quantization

PoT量化：乘法可以用移位进行计算，显著降低计算复杂度，但是，对于较高比特数，PoT量化提升有限

APoT量化：乘法通过移位和加法实现，计算效率受APoT的项数决定

量化方法比较

- **训练后量化（post training quantization）：模型训练完成后对权重、激活值进行量化，其中，激活值的量化，需要借助校准数据**
- **量化感知训练（quantization-aware training）：将训练过的模型量化后又再进行重训练，或者直接训练量化模型**



PTQ运行时间短，数据需求小，但是模型推理准确率下降较大

QAT准确率高，但是训练时间长，数据要求高，在特定任务上可能过拟合

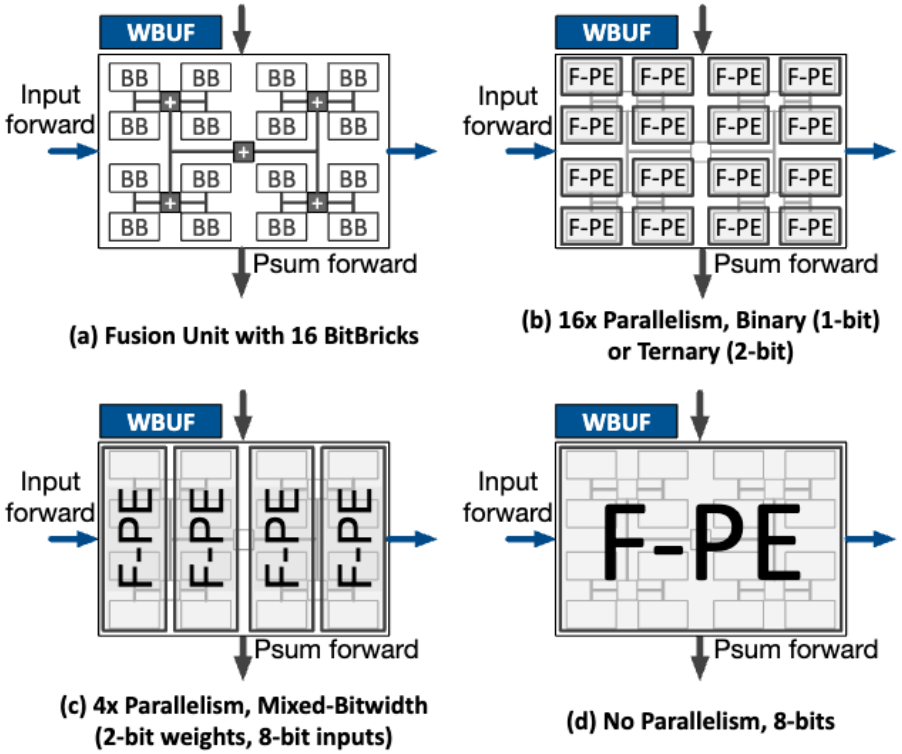
神经网络量化与AI处理器

北京大学 PEKING UNIVERSITY

- 量化神经网络想要真正实现计算能效、吞吐等的提升，离不开AI处理器的支持
  - 例如支持低比特计算的计算单元（乘法器、累加器等）、支持低比特数据的存储方式等等
- 因此，神经网络量化往往需要配合专用硬件（或核函数）设计，形成软硬件协同设计和优化
- 本节课，我们重点介绍3个例子
  - BitFusion：支持混合精度计算的AI处理器
  - OliVe：面向低比特大模型的AI处理器
  - ANT：基于定制化数据类型的AI处理器

神经网络量化与AI处理器 —— BitFusion

- Bit Fusion文章发表于ISCA 2018，核心在于高效支持混合精度神经网络推理
- 核心观察：不同神经网络模型以及同一神经网络模型中的不同层，对于量化噪声的容错能力不同
  - 采用混合精度量化可以在保持模型精度的同时，最大化压缩模型
- 文章提出Bit Fusion架构，重点通过可重构的计算单元，实现不同计算精度的灵活支持

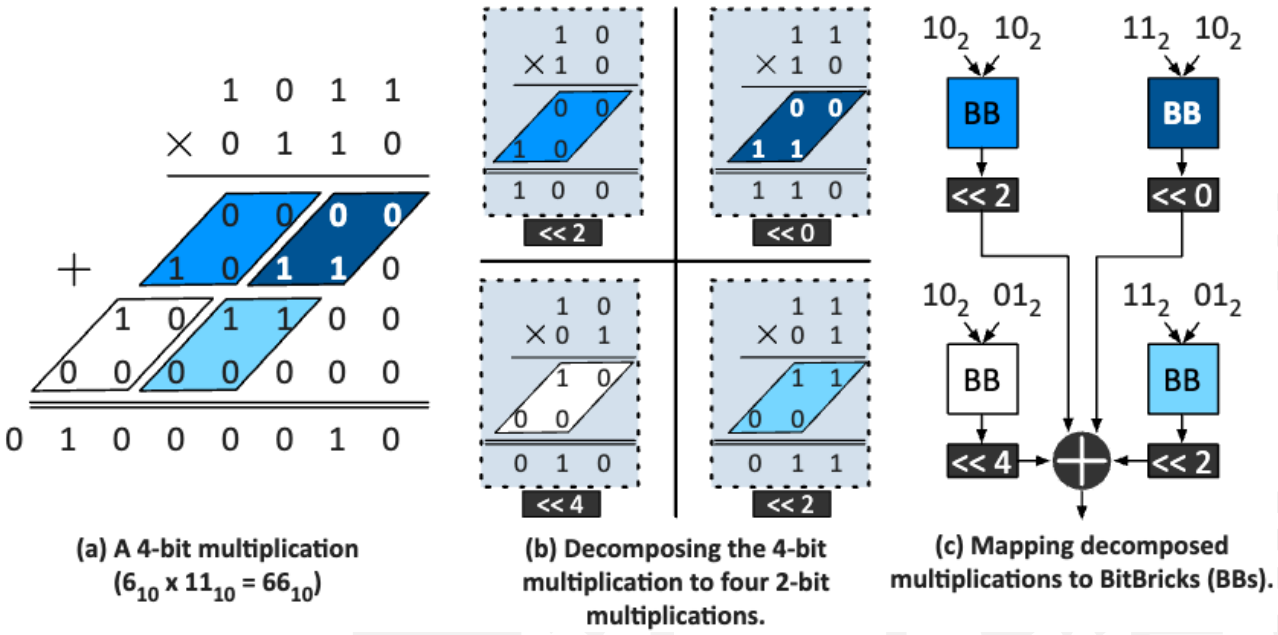


- Fusion单元由16个BitBricks组成，每个BitBrick每周期可以支持2比特计算
- 多个BitBricks组合，能够实现W4A4、W2A8、W8A2以及W8A8的计算

Hardik Sharma et al., *Bit Fusion: Bit-Level Dynamically Composable Architecture for Accelerating Deep Neural Network*, ISCA 2018

神经网络量化与AI处理器 —— BitFusion

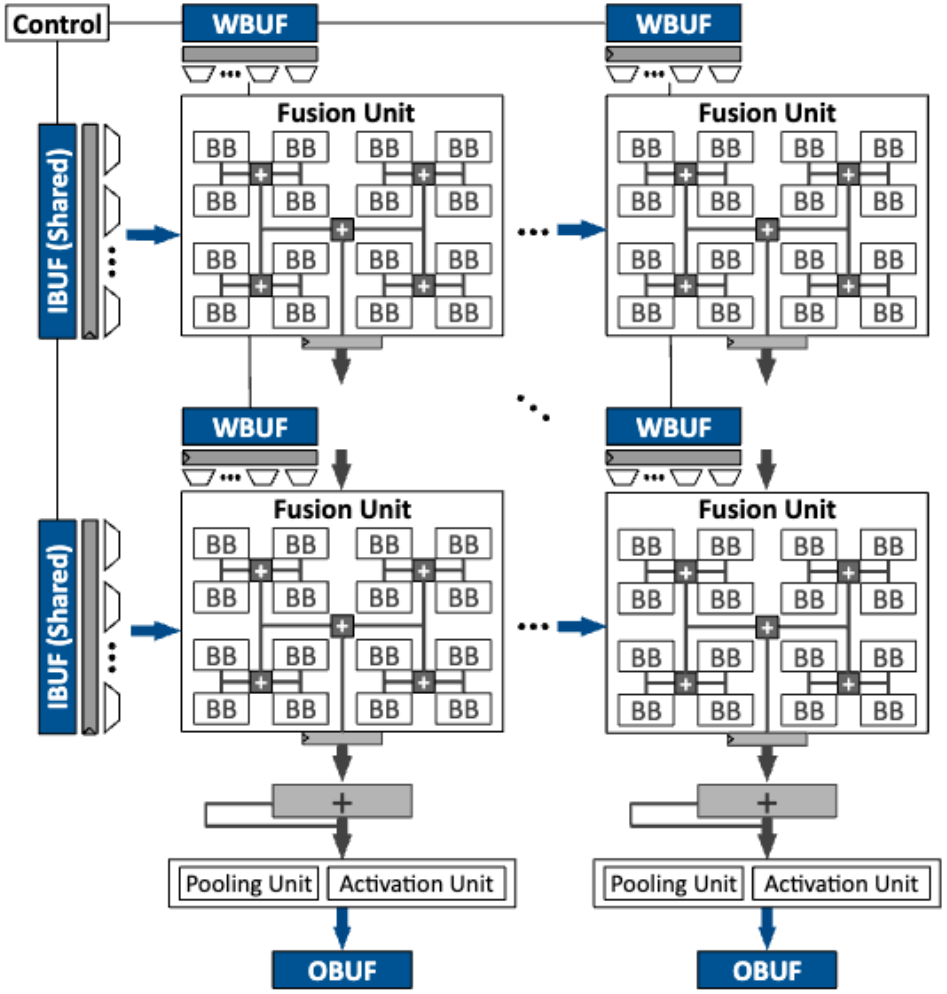
- 每个BitBrick里面需要**灵活的累加器设计**，才能有效实现不同比特精度的累加
- BitFusion采用脉动阵列的设计，进一步降低访存开销



(a) A 4-bit multiplication ($6_{10} \times 11_{10} = 66_{10}$)

(b) Decomposing the 4-bit multiplication to four 2-bit multiplications.

(c) Mapping decomposed multiplications to BitBricks (BBs).



神经网络量化与AI处理器——ANT

- ANT发表于MICRO 2022，提出了新的数据类型，适应神经网络推理需求
- 核心观察：1. 神经网络中不同的tensor分布各不相同；2.同一个tensor内部对于接近0的值或特别大的值不需要使用高精度表示；使用现有的INT或Float格式量化难以适应这两种变化
- 提出了一种新的数据类型flint，适合表示高斯分布的数据

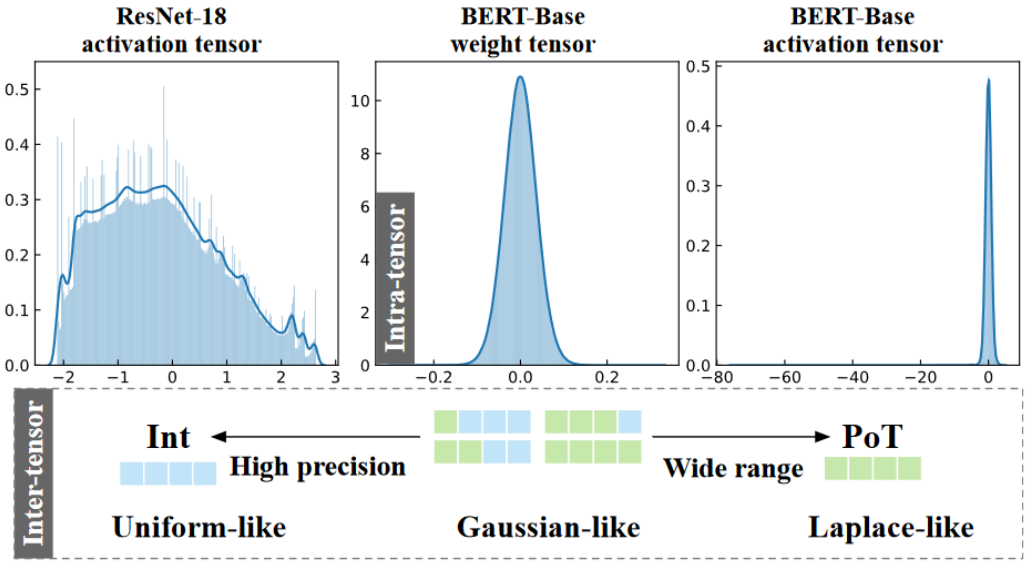


Figure 1: Intra-tensor and inter-tensor adaptivity.

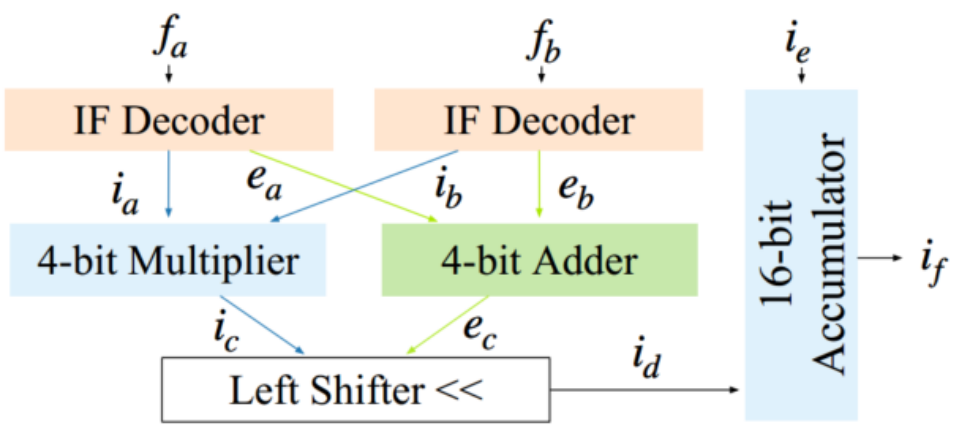
Bits	Exponent Value	Fraction Value	Value in Decimal
0000	-	0	0
0001	$1-1=0$	1	$2^0 \times 1 = 1$
001x	$2-1=1$	1, 1.5	2, 3
01xx	$3-1=2$	1, 1.25, 1.5, 1.75	4, 5, 6, 7
11xx	$4-1=3$	1, 1.25, 1.5, 1.75	8, 10, 12, 14
101x	$5-1=4$	1, 1.5	16, 24
1001	$6-1=5$	1	32
1000	$7-1=6$	1	$2^6 \times 1 = 64$

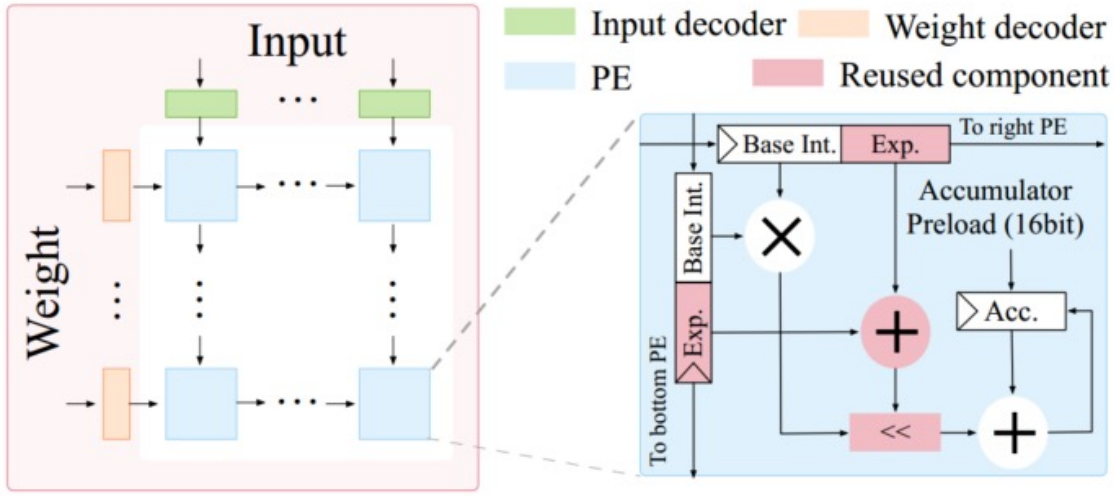
Table II: The value table of 4-bit unsigned `flint` with the exponent bias of -1. The blue numbers are the first-one-encoded exponent and "x" is mantissa with value of 0 or 1.

Cong Guo et al., *ANT: Exploiting Adaptive Numerical Data Type for Low-bit Deep Neural Network Quantization*, MICRO 2022

神经网络量化与AI处理器——ANT

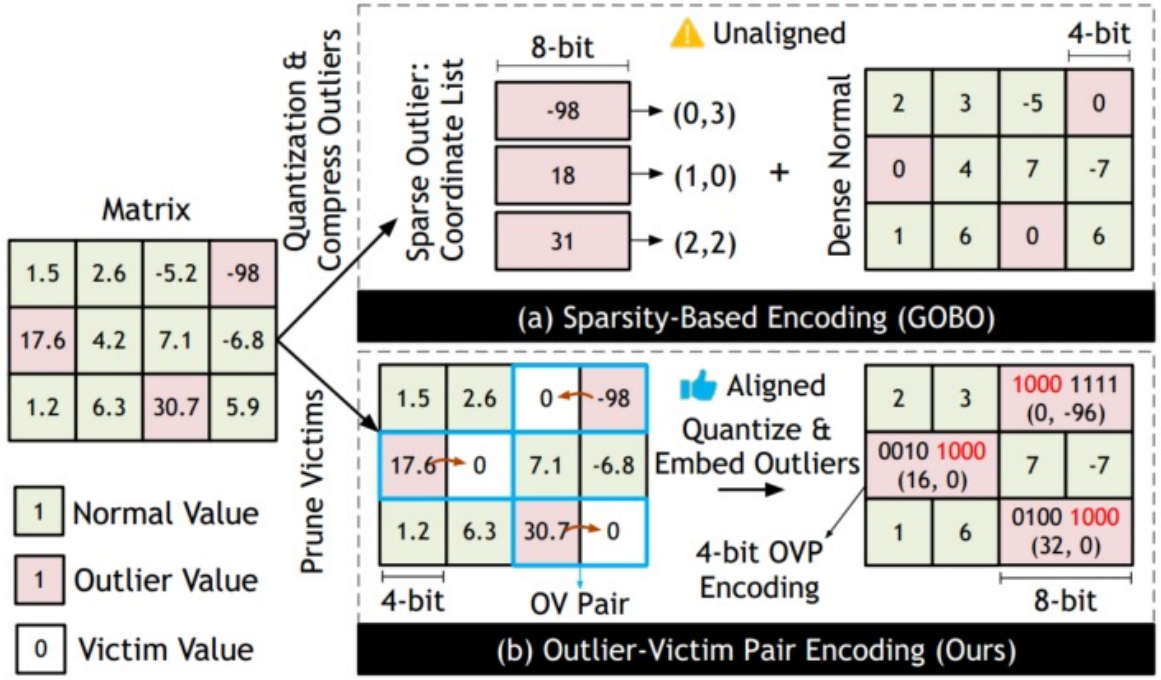
- 为了解决tensor之间数据分布不同的问题，通过校准数据集选择量化后均方误差最小的数据类型
  - 直觉上int类型适合均匀分布，Float/PoT类型适合拉普拉斯分布，Flint类型适合高斯分布
- 设计了可以支持以上**四种格式的处理单元（PE）**，主要包括解码器，乘法器，加法器和累加器
- 将设计的PE集成到了**脉动阵列**形成新的架构



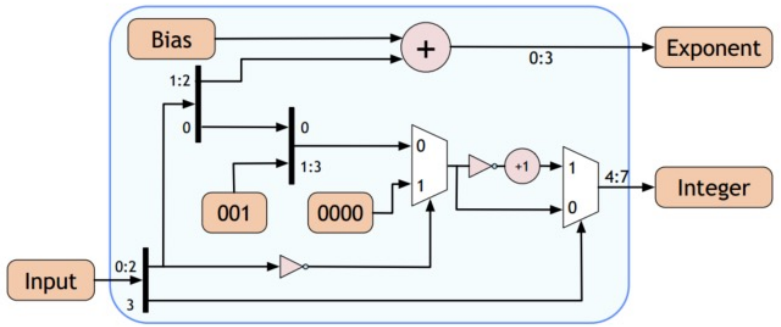


神经网络量化与AI处理器——OliVe

- OliVe发表于ISCA 2023，针对大模型中的离群值（outliers）进行设计
- 核心观察：大语言模型中存在一些ouliers, 这些outliers很重要，但旁边的正常值（称为victims）不重要，因此可以牺牲victims进而用更多比特表示outliers
- 对于outliers, 提出了新的数据格式Abfloat, 通过自适应的bias使编码的值表示比正常值更大的数



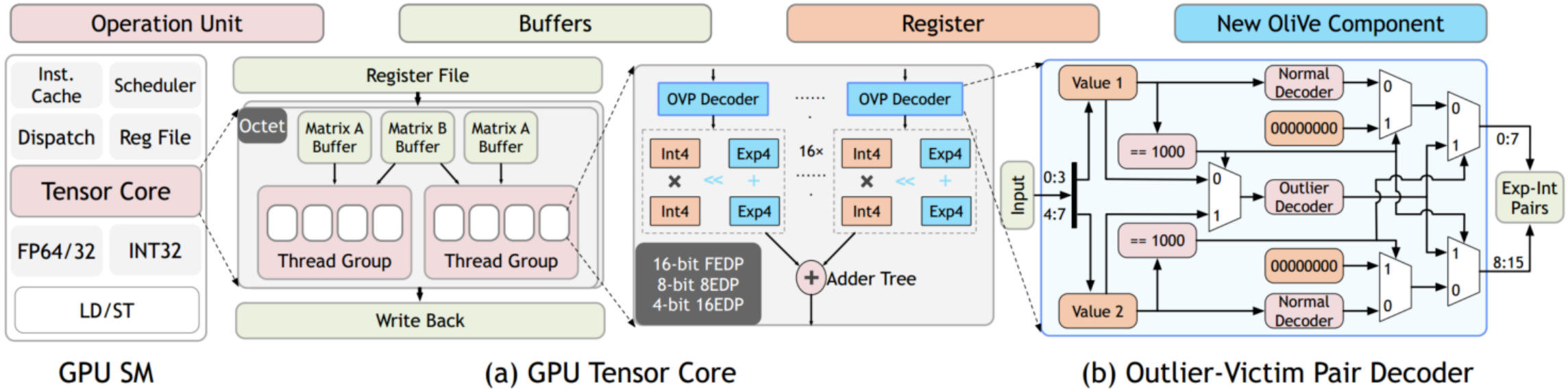
Binary	Exponent	Integer	Real Value
000	0	0	0
001	0	3	$3 \times 2^0 = 3$
01x	1	2, 3	$2 \times 2^1 = 4, 3 \times 2^1 = 6$
10x	2	2, 3	$2 \times 2^2 = 8, 3 \times 2^2 = 12$
11x	3	2, 3	$2 \times 2^3 = 16, 3 \times 2^3 = 24$



Cong Guo et al., *OliVe: Accelerating Large Language Models via Hardware-friendly Outlier-Victim Pair Quantization*, ISCA 2023

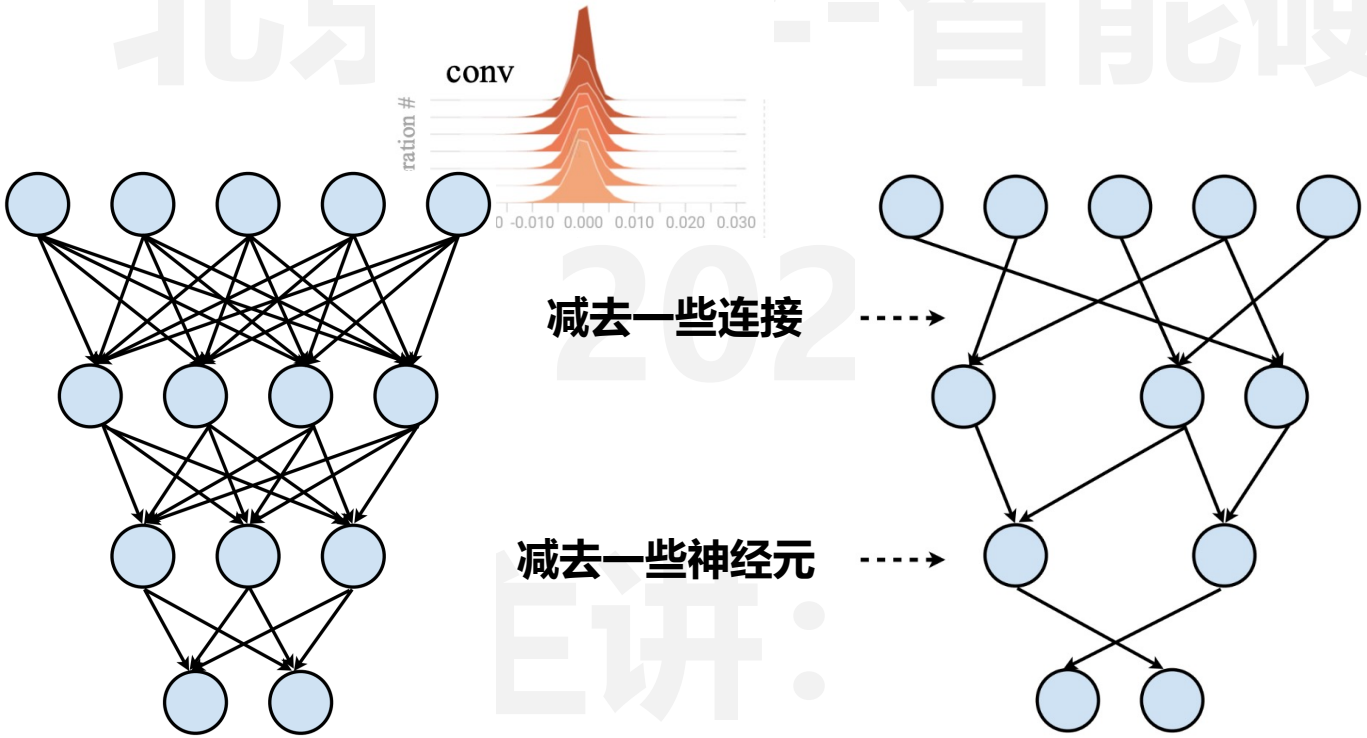
神经网络量化与AI处理器——OliVe

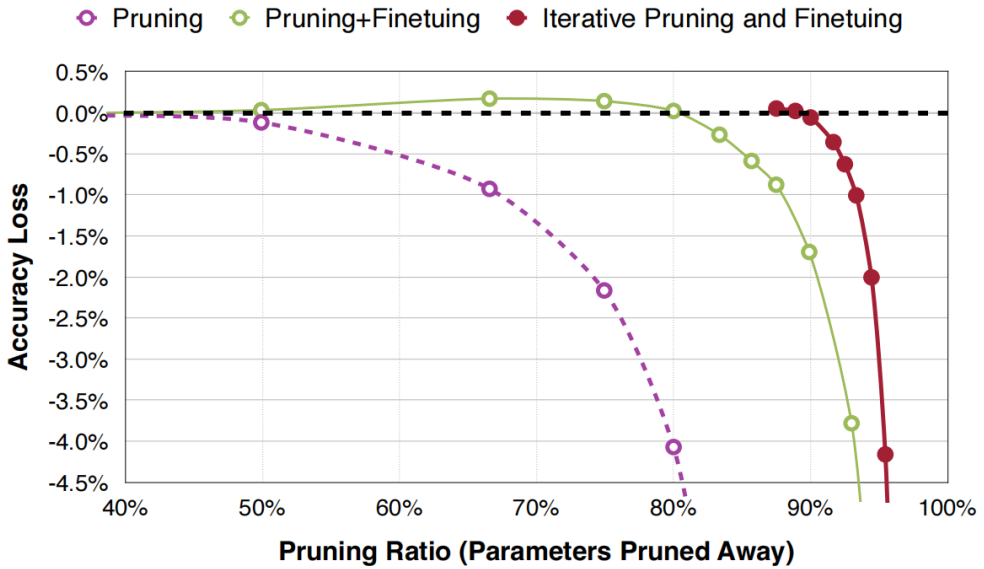
- 解码完成后，文章设计了乘累加单元电路（MAC Unit），并将他们集成到了GPU tensor core以及脉动阵列中



基于模型稀疏化的软硬件协同设计

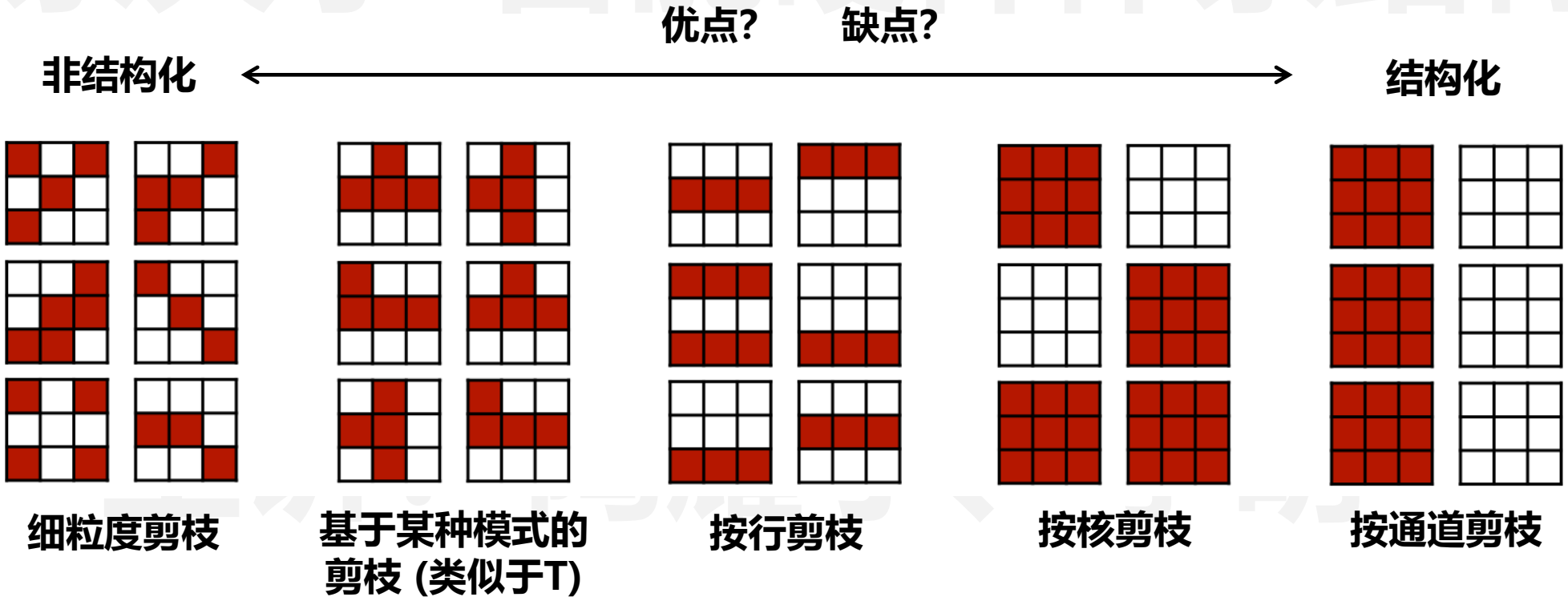
- 除了对于量化噪声的容错性，神经网络往往呈现显著的**稀疏性**，即对特定神经元或者模型权重进行剪枝，不会影响其推理准确率





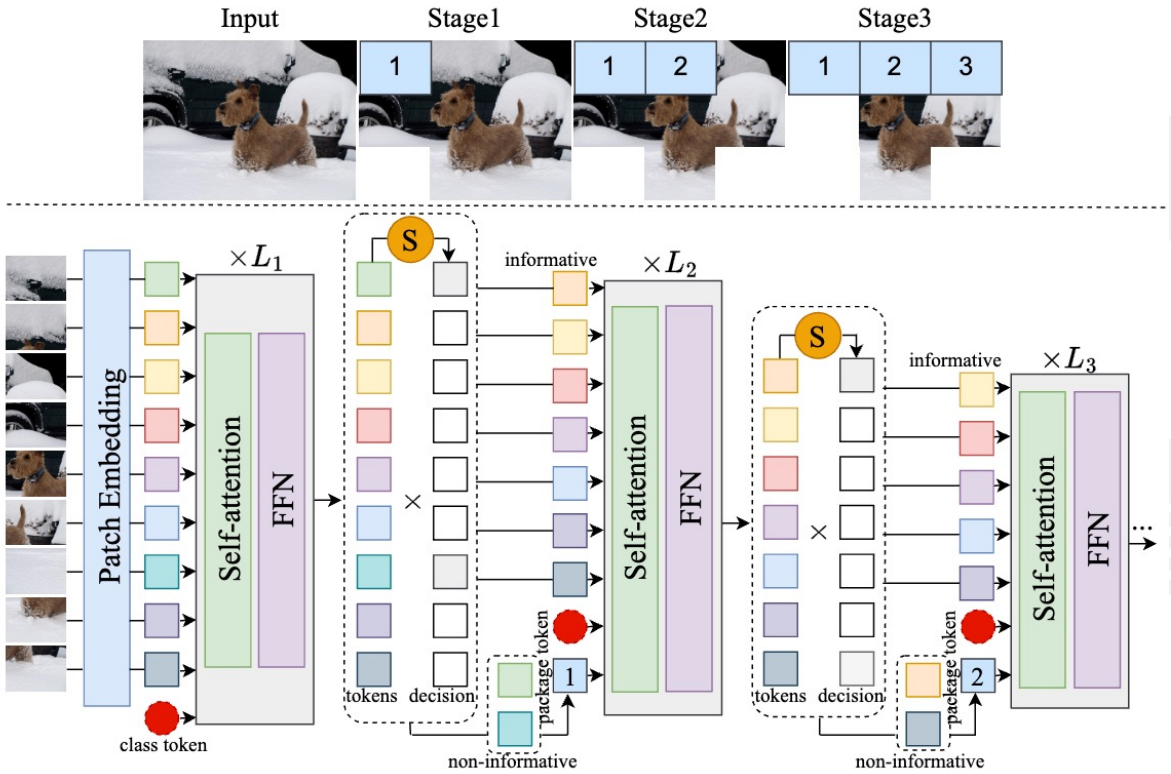
模型稀疏化的Taxonomy

- 依据剪枝粒度不同，神经网络剪枝可以分为**结构化**和**非结构化剪枝**
- 细粒度非结构化剪枝更加灵活，往往准确率更高（或稀疏度更高），但是硬件加速更加困难
- 粗粒度结构化剪枝则更加硬件友好，但是灵活性受限



模型稀疏化的Taxonomy

- 依据剪枝对象不同，神经网络剪枝可以分为**权重剪枝**与**激活值剪枝**
- 权重剪枝适用于CNN、RNN、Transformer等，通常为**静态**剪枝
- 激活值剪枝则更常见于Transformer模型或基于ReLU的CNN模型中，通常为**动态**剪枝



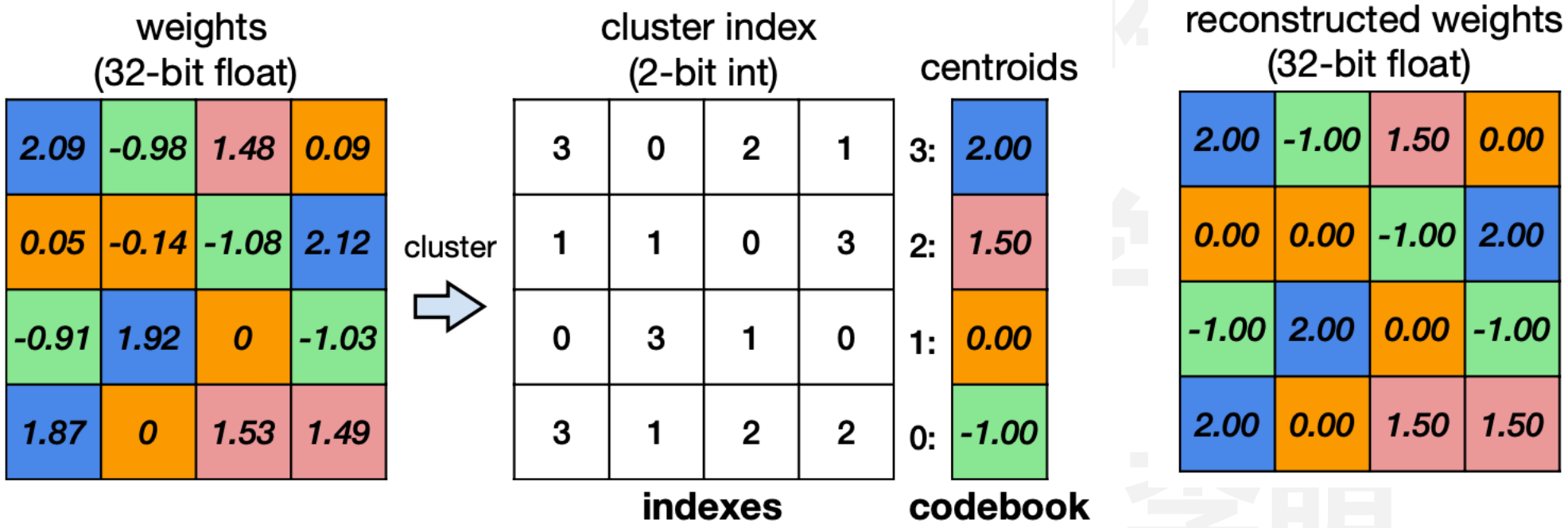
Peiyan Dong et al., *HeatViT: Hardware-Efficient Adaptive Token Pruning for Vision Transformers*, HPCA 2023

神经网络稀疏化与AI处理器

- 稀疏神经网络想要真正实现计算能效、吞吐等的提升，同样需要底层AI处理器的支持
  - 特别是对于稀疏激活值，导致计算呈现显著动态性
  - 例如动态剪枝单元、动态计算单元等
- 因此，神经网络量化往往需要配合专用硬件（或核函数）设计，形成软硬件协同设计和优化
- 本节课，我们重点介绍3个例子
  - EIE、SNAP：同时考虑权重和激活稀疏的AI处理器架构
  - Nvidia sparse tensor core：商用稀疏AI处理器架构

神经网络量化与AI处理器 —— EIE

- 文章发表于ISCA 2016，针对基于量化和剪枝的压缩模型，提出了高效推理引擎EIE



Song Han et al., *EIE: efficient inference engine on compressed deep neural network*, ISCA 2016

神经网络量化与AI处理器 —— EIE

- 文章发表于ISCA 2016，针对基于量化和剪枝的压缩模型，提出了高效推理引擎EIE
- 核心观察：缺少有效处理不规则存储的稀疏数据的硬件单元
  - 静态的权重稀疏+动态的激活稀疏
- 文章提出高效推理引擎EIE，实现了高效的稀疏矩阵向量乘法

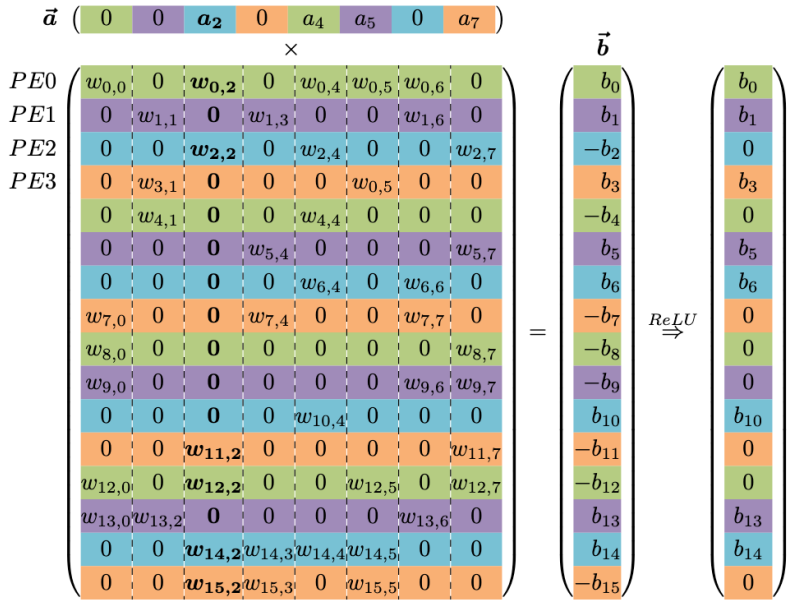


Figure 2. Matrix W and vectors a and b are interleaved over 4 PEs. Elements of the same color are stored in the same PE.

- 权重稀疏通过 Index + value 存储为 Compressed sparse column (CSC) 格式
- 激活稀疏通过只向PE广播非零激活值实现
- 权重共享通过将权重固定为16个值并用4位索引查找实现

神经网络量化与AI处理器 —— EIE

- **Act Queue平衡负载**：非零激活值对应的一列权重中的非零数量不同导致不同PE间负载不平衡
- Leading Non-zero Detection Node：检测非零激活值并广播到所有PE

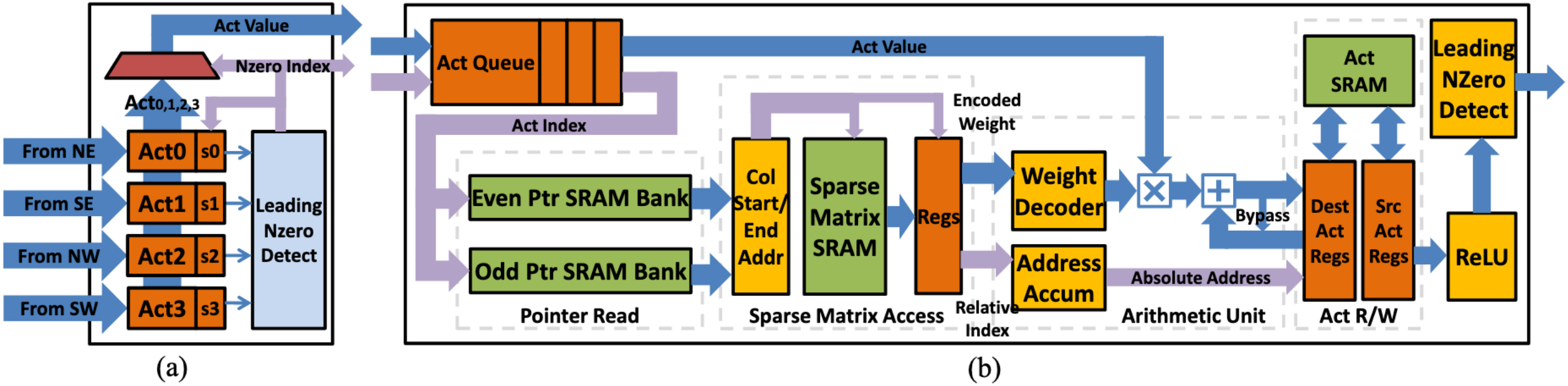
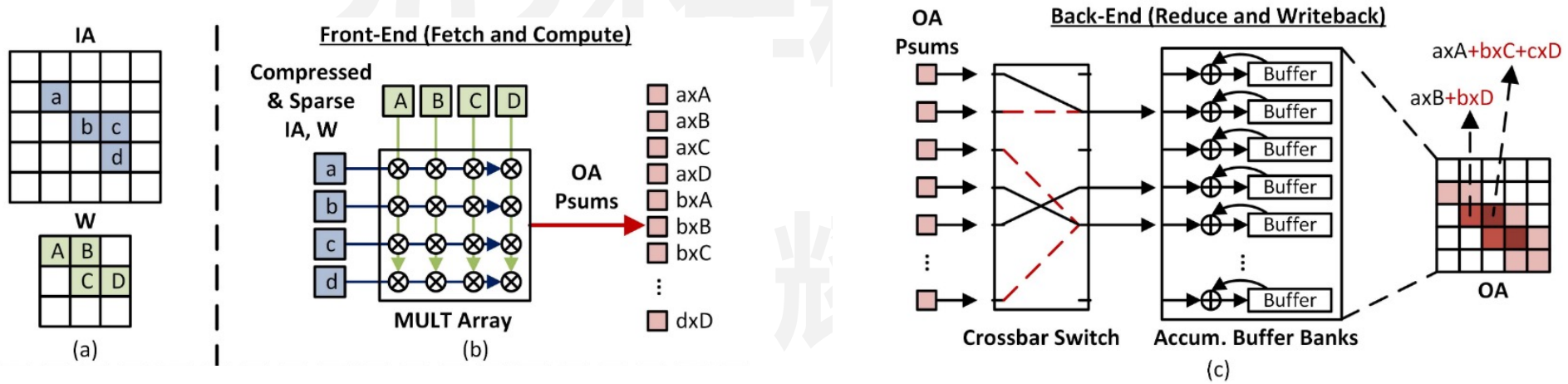


Figure 4. (a) The architecture of Leading Non-zero Detection Node. (b) The architecture of Processing Element.

神经网络量化与AI处理器 —— SNAP

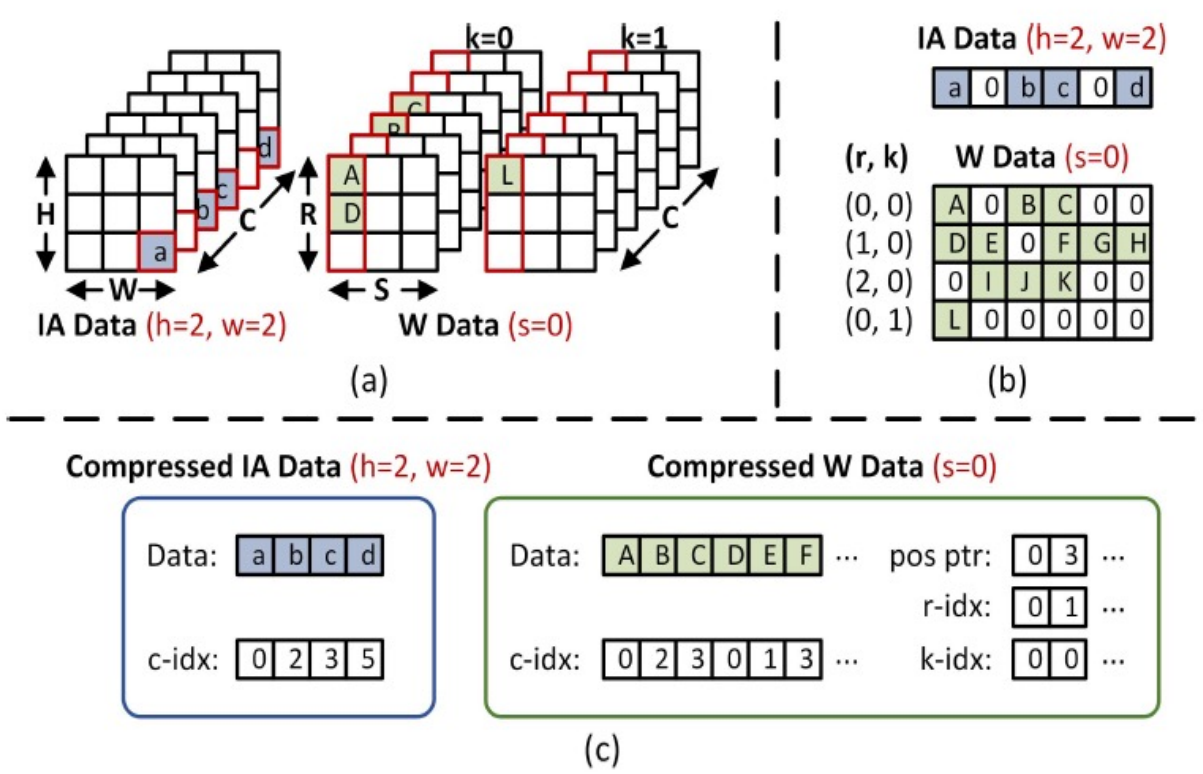
- 文章发表于JSSCC 2021，通过channel-first数据流对稀疏网络模型进行了硬件加速
- 核心观察：数据稀疏使得网络推理更加高效，但现有稀疏计算的数据流仍面临以下挑战
  - Front-end：读取的W-IA pairs数量不足导致计算单元利用率低
  - Back-end：计算结果写回的地址冲突
  - 灵活性：对于不同kernel大小和层类型的支持性不足

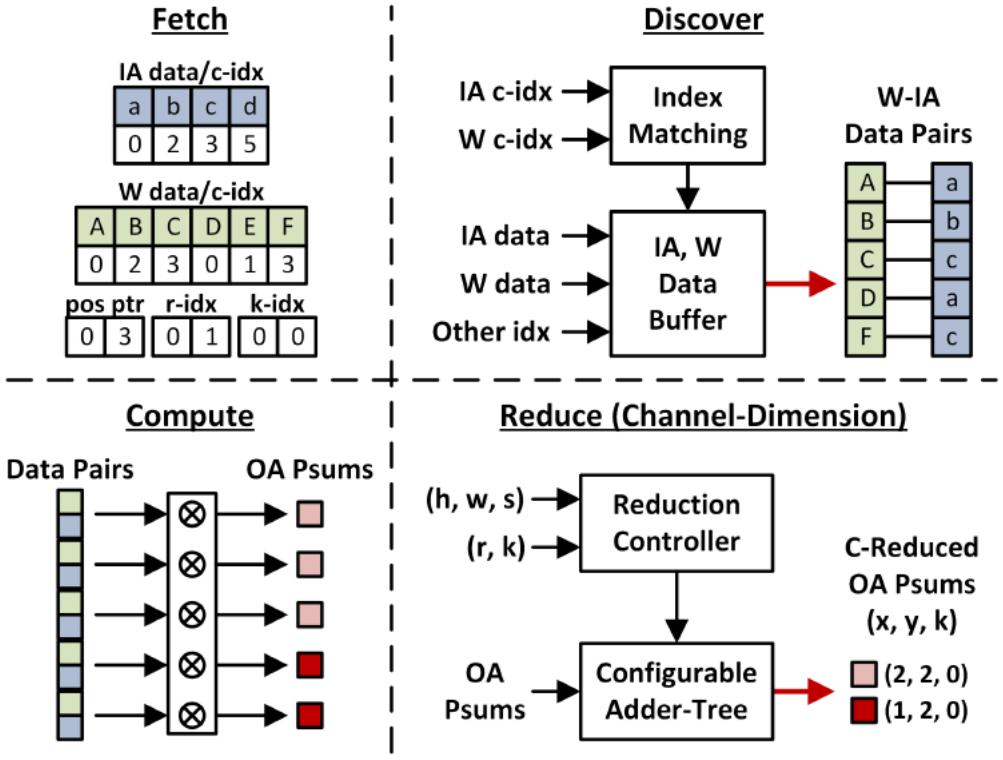


Jie-Fang Zhang et al., *SNAP: An Efficient Sparse Neural Acceleration Processor for Unstructured Sparse Deep Neural Network Inference*, JSSC 21

神经网络量化与AI处理器 —— SNAP

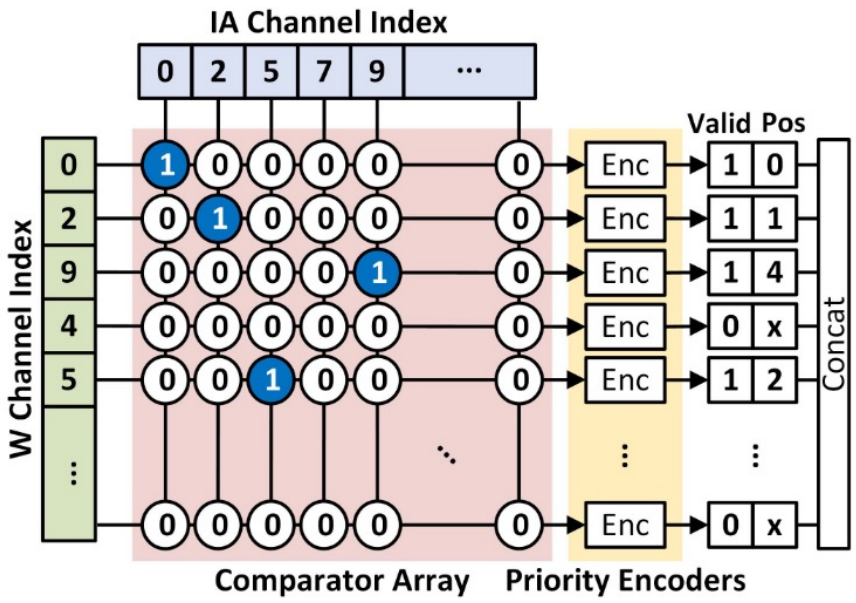
- Channel-first数据流：同一个channel的可以任意配对，产生的乘法结果都是有用的
- 通过可配置加法树实现先归约加和，然后更新Psum，从而减少Psum更新时的地址冲突

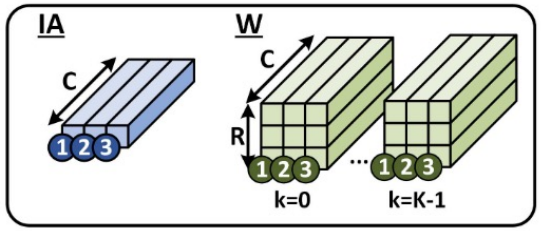


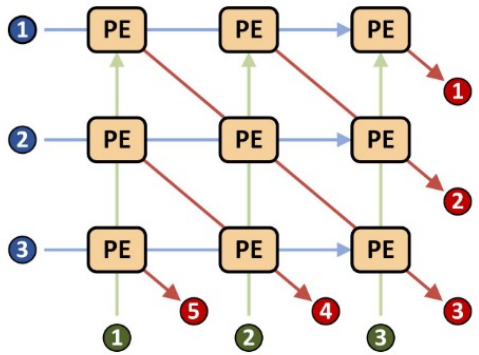


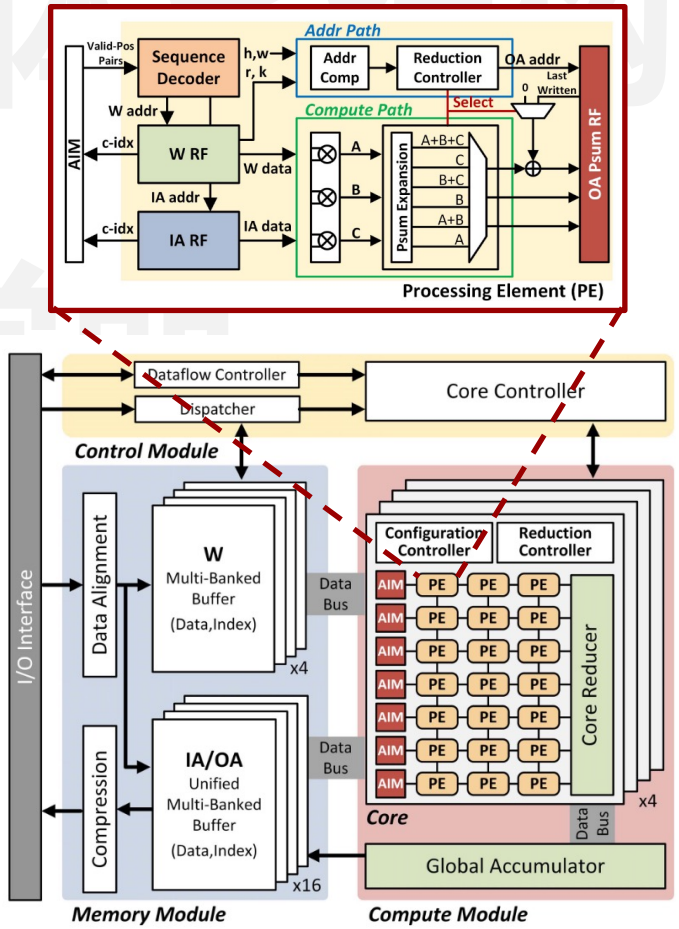
神经网络量化与AI处理器 —— SNAP

- Front-end：Index Matching单元提取足够数量的 W-IA pairs，提高乘法阵列利用率
- Back-end：两级 partial sum 压缩，PE-level 和 core-level，减少内存访问冲突
- 灵活性：core-level的压缩可以支持不同层的计算



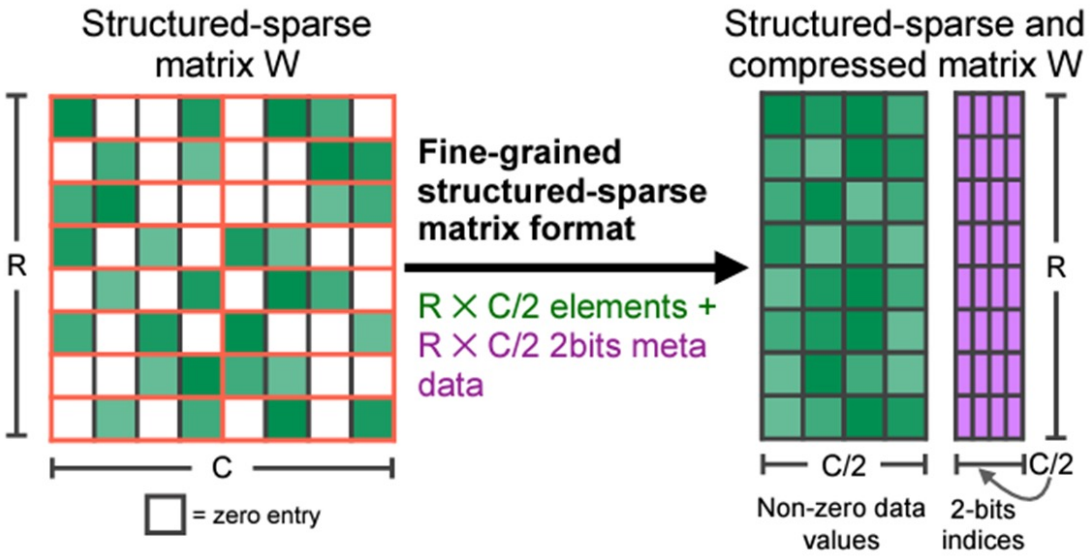


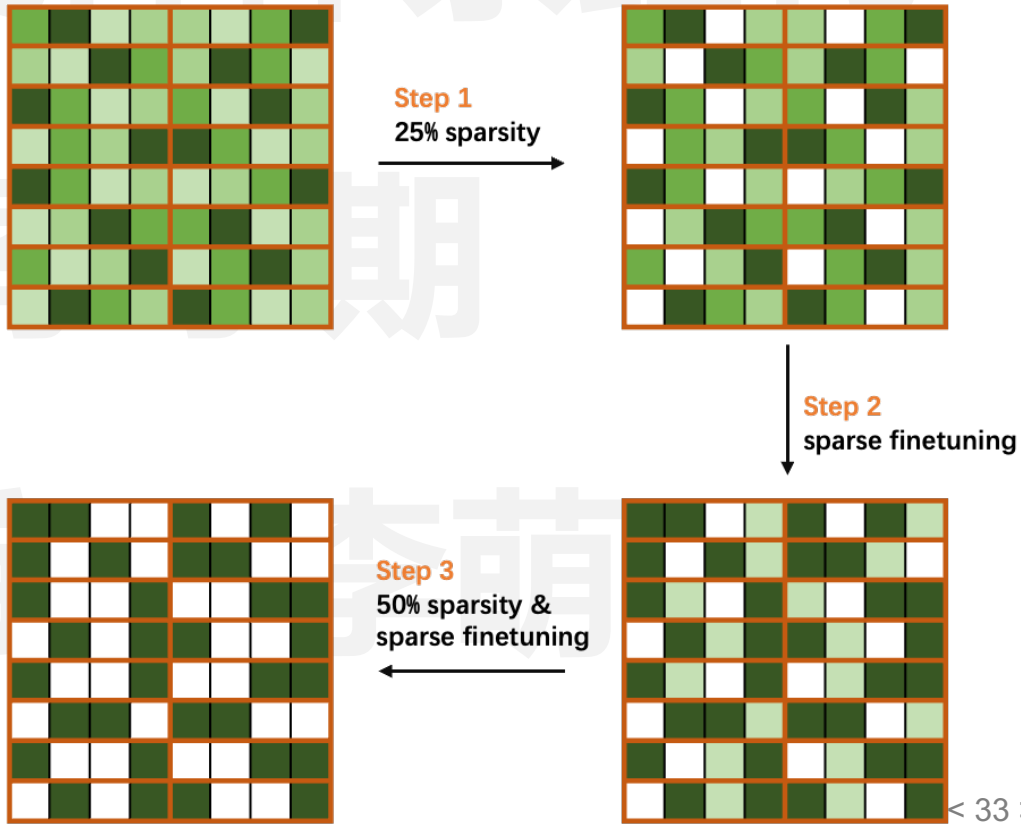




神经网络量化与AI处理器——NV Sparse TensorCore

- 2020年，英伟达推出Ampere架构，支持稀疏张量计算
- Ampere架构支持固定的2:4稀疏模式，即每4个元素中，2个为0，并且能够实现2倍的加速比
- 该稀疏模式可以拓展为更加普适的N:M稀疏，即每M个元素中，N个为0





Reference: NV Technical Blog