AI 芯片 – AI 计算体系

AI计算体系总结

ZOMI

Talk Overview

1. AI 计算体系

- 深度学习计算模式
- 计算体系与矩阵运算

2. AI 芯片基础

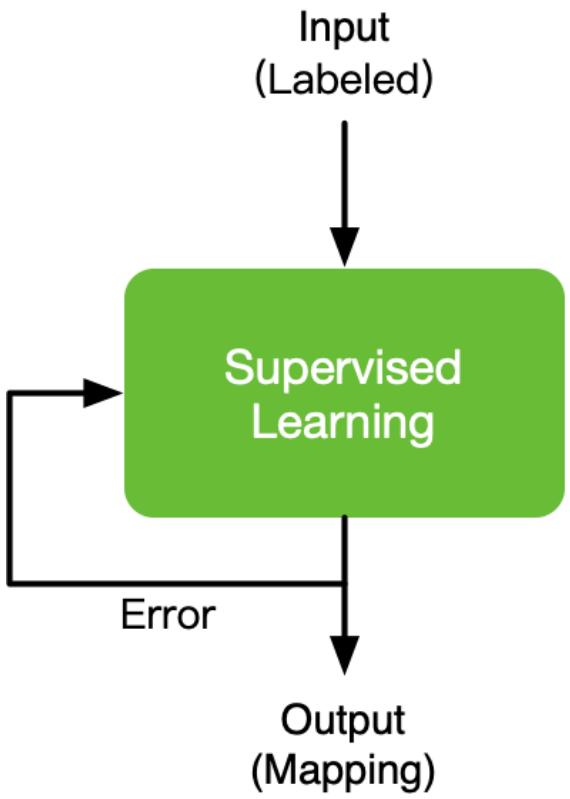
- 通用处理器 CPU
- 从数据看 CPU 计算
- 通用图形处理器 GPU
- AI专用处理器 NPU/TPU
- 计算体系架构的黄金10年

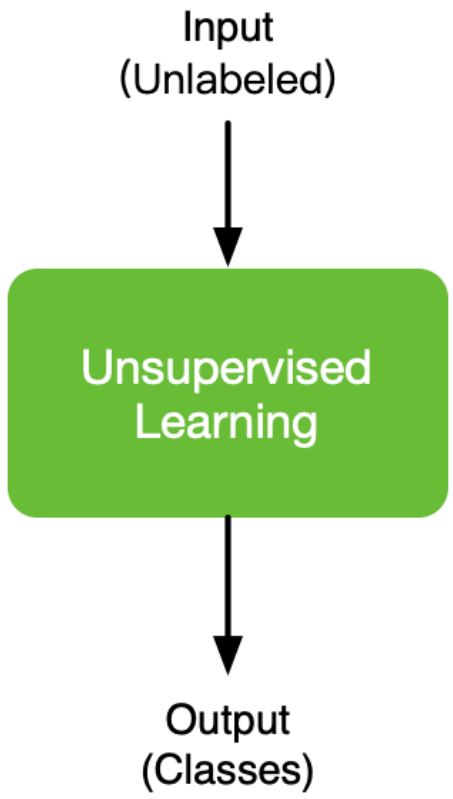
Talk Overview

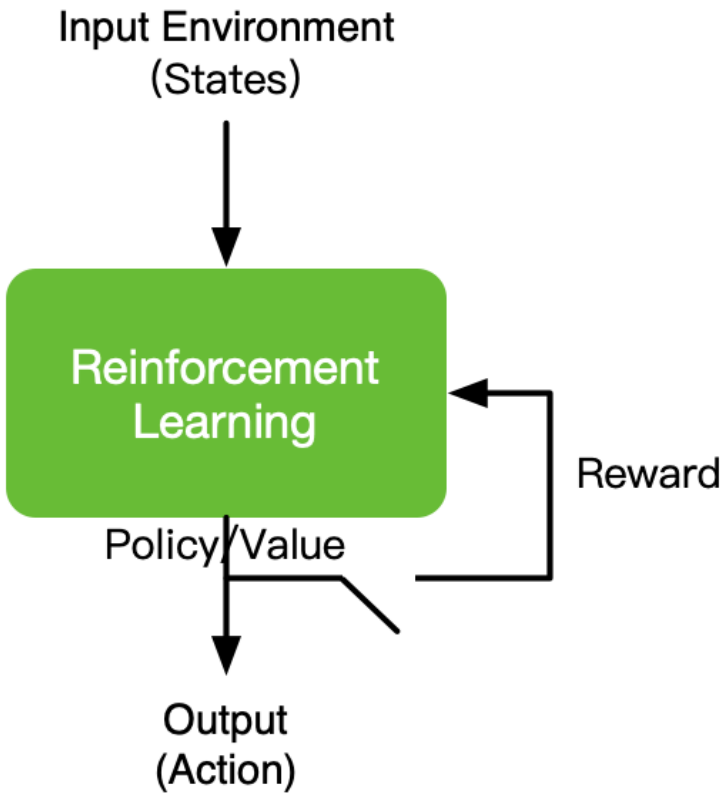
1. 深度学习计算模式

- The History – AI 的发展和范式
- Models Architecture – 经典模型结构
- Quantization and Pruning – 模型量化与剪
- Efficient Models – 轻量化网络模型
- Models Parallel – 大模型分布式并行

AI 三大范式流程



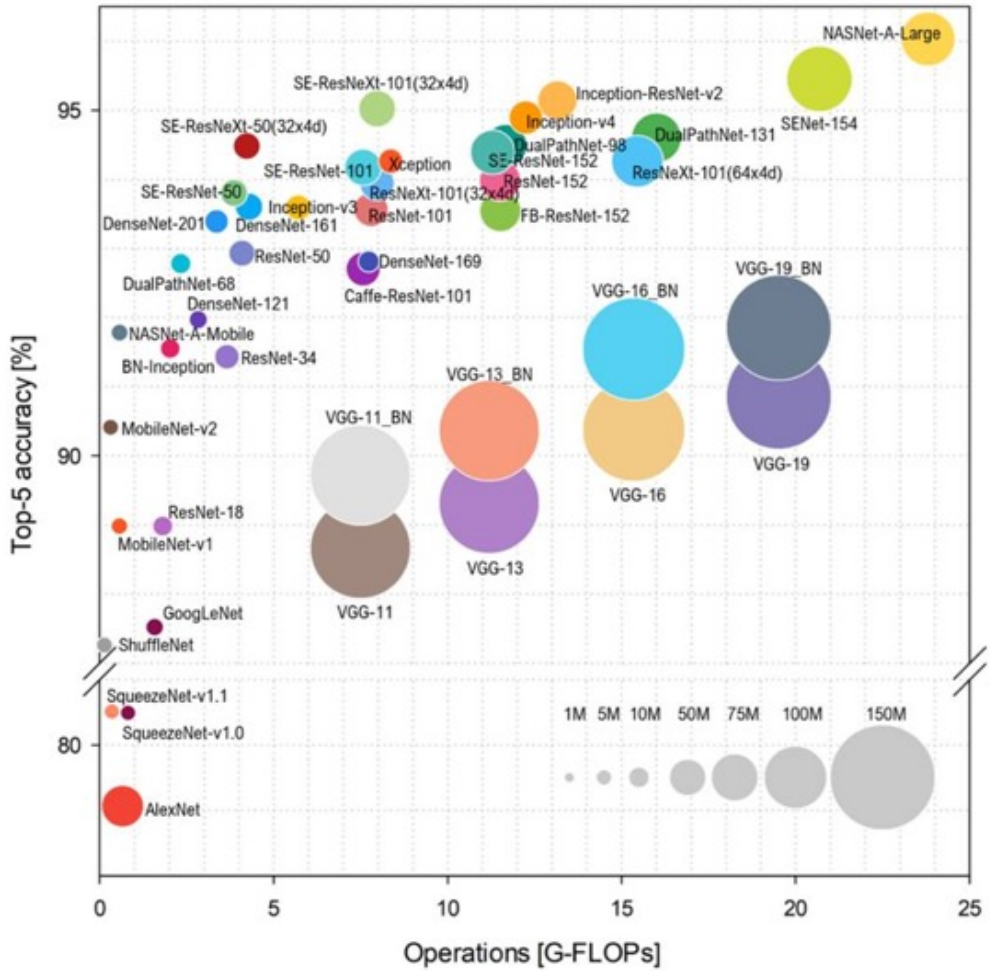




经典网络模型

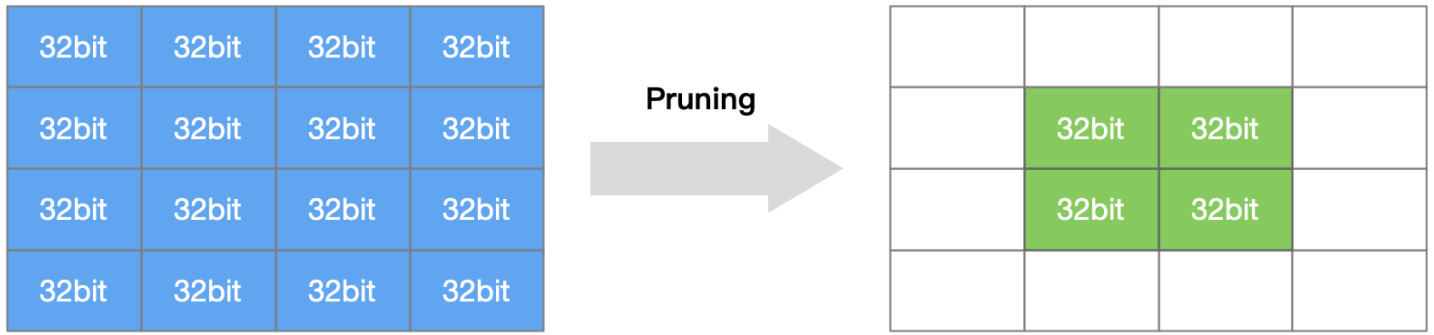
Models getting larger and deeper

Metrics	LeNet-5	AlexNet	VGG16	GoogleNet	ResNet50	EfficientNet-B4
Top-5 error(Image Net)	n/a	16.4	7.4	6.7	5.3	3.7
Input Size	28x28	227x227	224x224	224x224	224x224	380x380
# Conv Layer	**2**	**5**	**16**	**21**	**49**	**96**
# of Weights	2.6k	2.3M	14.7M	6.0M	23.5M	14M
# of MACs	283k	666M	15.3G	1.43G	3.86G	4.4G
# FC Layers	**2**	**3**	**3**	**1**	**1**	**65**
# of Weights	58k	58.6M	124M	1M	2M	4.9M
# of MACs	58k	58.6M	124M	1M	2M	4.8M
Total Weights	**60k**	**61M**	**138M**	**7M**	**25.5M**	**19M**
Total MACs	**341k**	**724M**	**15.5G**	**1.43G**	**3.9G**	**4.4G**
Reference	Lecun,1998	Krizhevsky,2012	Simonyan,2015	Szegedy,2015	He,2016	Tan,2019

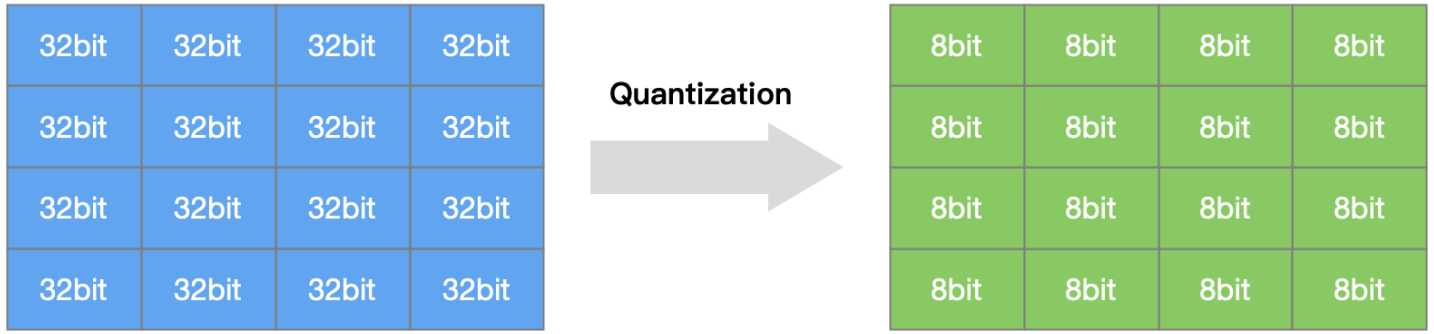


量化压缩 vs 网络剪枝

- 网络剪枝研究模型权重中的冗余， 并尝试删除/修剪冗余和非关键的权重。



- 模型量化是指通过减少权重表示或激活所需的比特数来压缩模型。



AI 计算模式思考

1. **需要支持神经网络模型的计算逻辑**
   - 权重数据共享
   - 需要支持激Vector计算
2. **能够支持高维的张量存储与计算**
   - 内存 Mem 地址随机/自动索引
   - 大 Channel 和 大 Feature Map 高效加载
3. **支持常用神经网络模型结构**
   - Conv、MatMul、Transformer等高效矩阵乘
   - 快速应对新的 AI 算法与结构
4. **提供不同的 bit 位数**
   - 对于低比特量化相关的研究落地提供bits
   - 在 M-bits/E-bits 之间权衡（如 TF32/BF16）
5. **利用硬件提供稀疏计算**
   - 硬件上减少 0 值的重复计算
   - 减少对内存需求，稀疏化网络模型结构

Talk Overview

1. 深度学习计算模式

- The History – AI 的发展和范式
- Models Architecture – 经典模型结构
- Quantization and Pruning – 模型量化与剪
- Efficient Models – 轻量化网络模型
- Models Parallel – 大模型分布式并行

经典的轻量级模型

CNN 系列

1. SqueezeNet 系列（2016）
2. ShuffleNet 系列（2017）
3. MobileNet 系列（2017）
4. ESPnet 系列（2018）
5. FBNet系列（2018）
6. EfficientNet 系列（2019）
7. GhostNet 系列（2019）

Transformer 系列

1. MobileViT （2021）
2. Mobile-Former（2021）
3. EfficientFormer（2022）

Megatron-LM 语言大模型

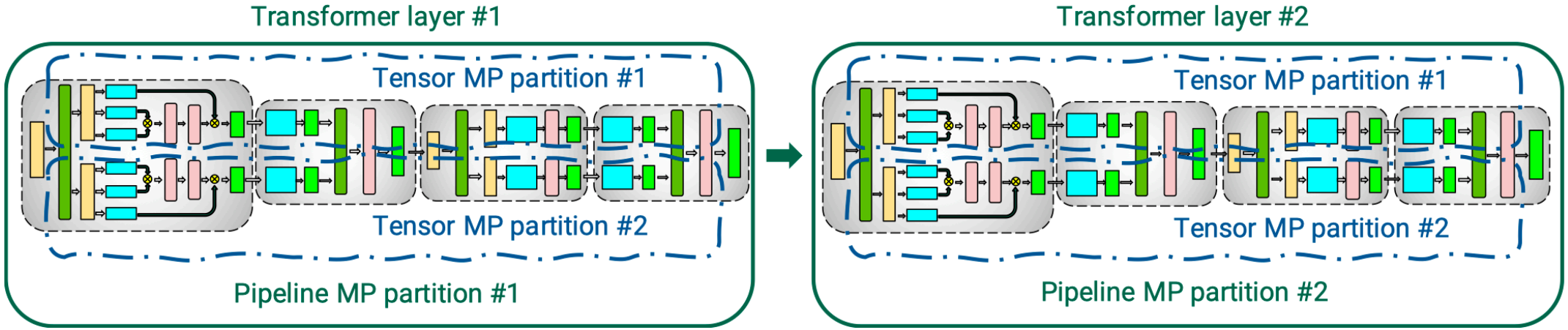
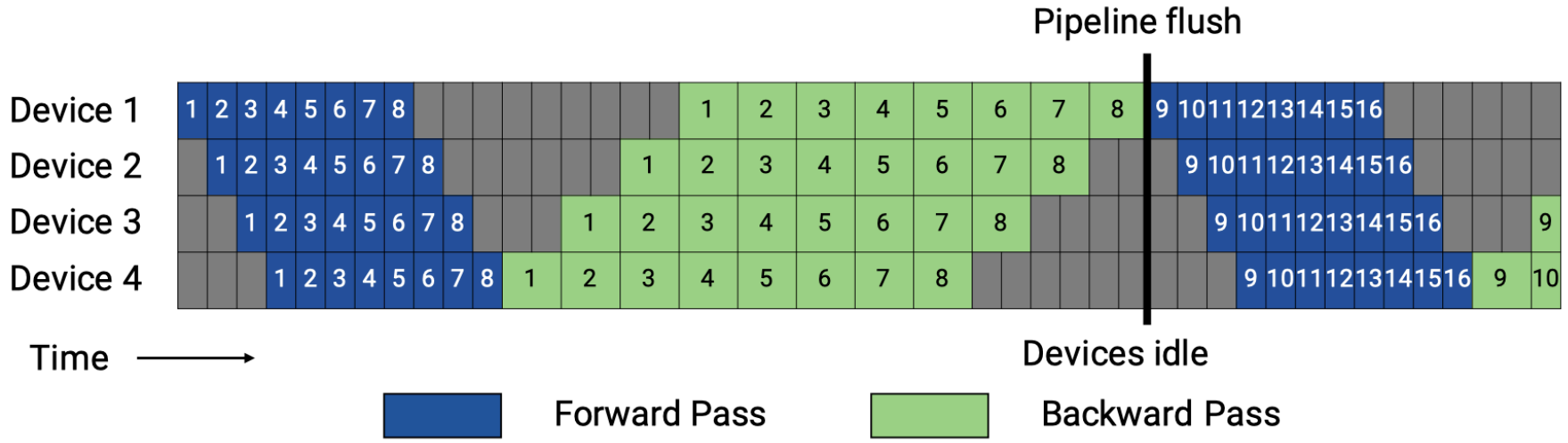


Figure 2: Combination of tensor and pipeline model parallelism (MP) used in this work for transformer-based models.



Narayanan, Deepak, et al. "Efficient large-scale language model training on gpu clusters using megatron-lm." Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2021.

AI 计算模式思考 Summary

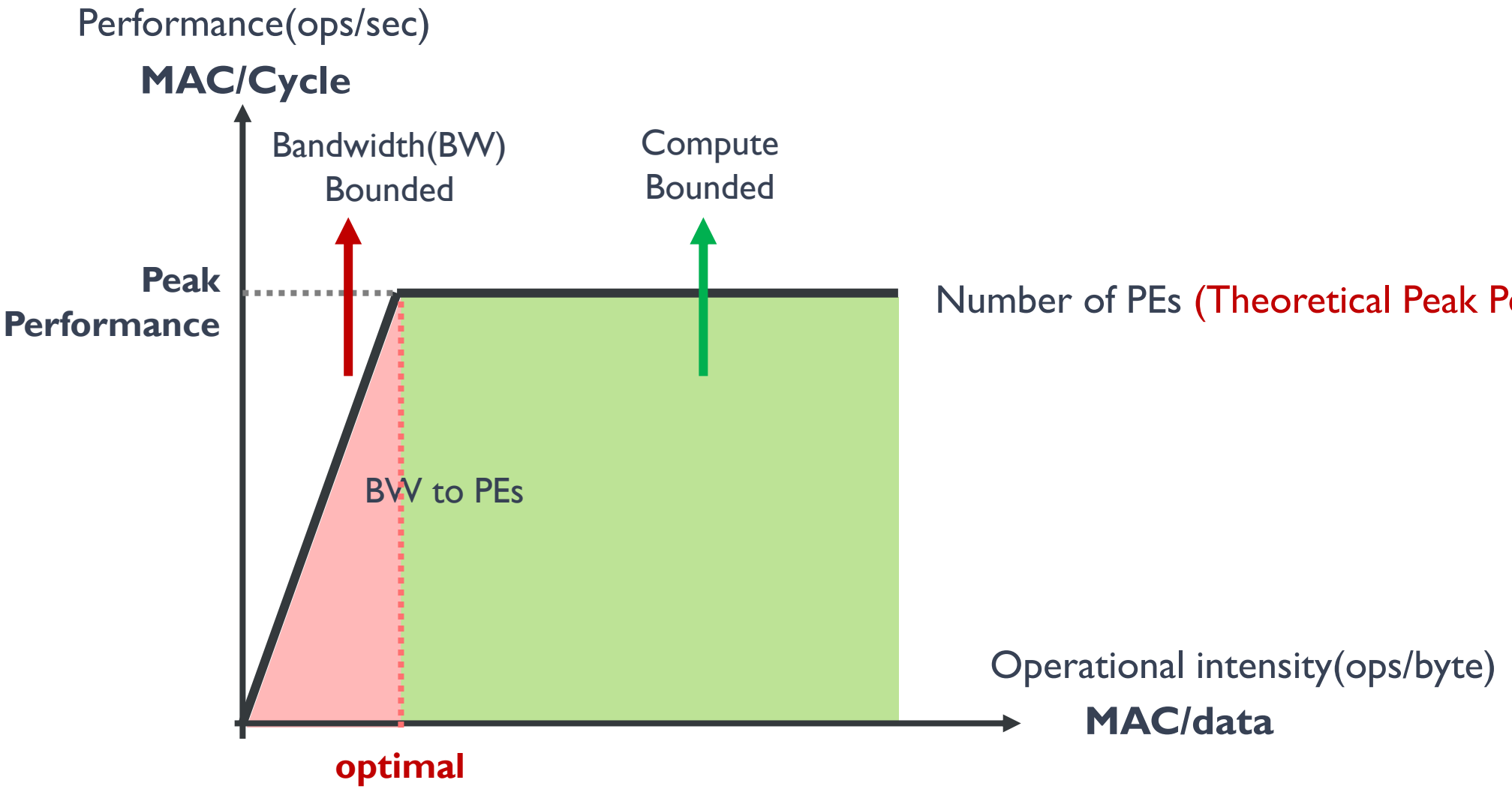
1. **网络模型结构支持 Architecture**
   - 支持高维的张量存储与计算
   - 神经网络模型的计算逻辑
2. **模型压缩(剪枝&量化) Model Compress**
   - 提供不同的 bit 位数
   - 利用硬件提供稀疏计算
3. **轻量化网络模型 Model Slim**
   - 复杂卷积计算（小型卷积核，e.g. 1x1 Conv）
   - 复用卷积核内存信息（Reuse Convolution）
4. **大模型分布式并行 Foundation Model**
   - 大内存容量、高速互联带宽
   - 专用大模型DSA IP模块，提供低比特快速计算

Talk Overview

I. AI 计算体系与矩阵运算

- Key Metrics – AI芯片关键指标
- Bit Width – 比特位数
- Matrix Multiplication – 矩阵运算
- Specialized Hardware – 专用硬件

计算性能仿真



https://arxiv.org/abs/1807.07928

Key Metrics 与计算体系思考 I

1. **精度 Accuracy**
   - 能够处理各类型的无规则数据 >> 异构平台
   - 能够应对复杂网络模型结构 >> 计算冗余性
2. **吞吐量 Throughout**
   - 除了峰值算力，看 PE 的平均利用率 >> 负载均衡
   - SOTA网络模型的运行时间 >> MLPerf
3. **时延 Latency**
   - 通信时延对 MACs 的影响 >> 优化带宽
   - Batch Size 大小与内存大小 >> 多级缓存设计

Key Metrics 与计算体系思考 II

4. **能耗 Energy**
   - 执行SOTA网络模型时候 Ops/W >> 部署场景
   - 内存读写功耗 (e.g., DRAM) >> 降低能耗
5. **系统价格 System Cost**
   - 片内多级缓存 Cache 大小 >> 内存设计
   - PE 数量、芯片大小、纳米制程 >> 电路设计
6. **易用性 Flexibility**
   - 对主流AI框架支持度 (PyTorch) >> 软件栈

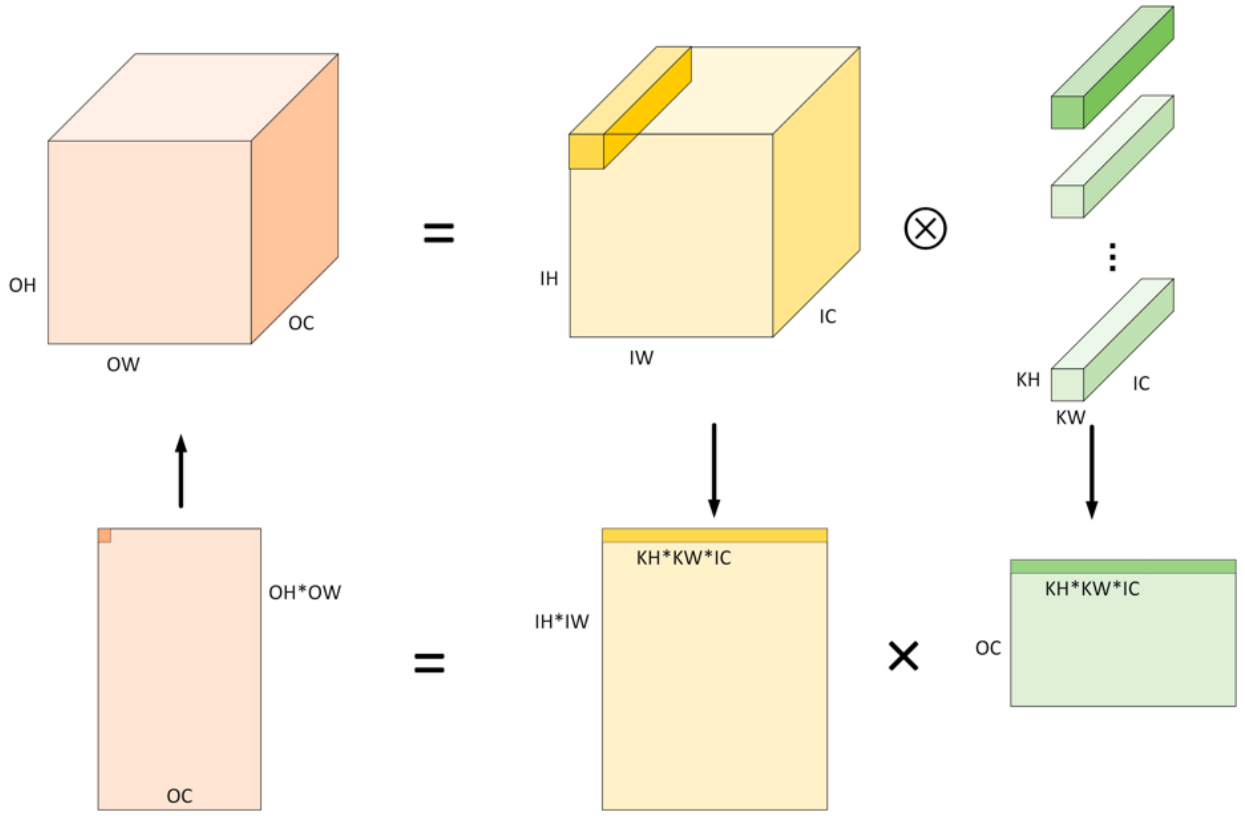
Talk Overview

I. AI 计算体系与矩阵运算

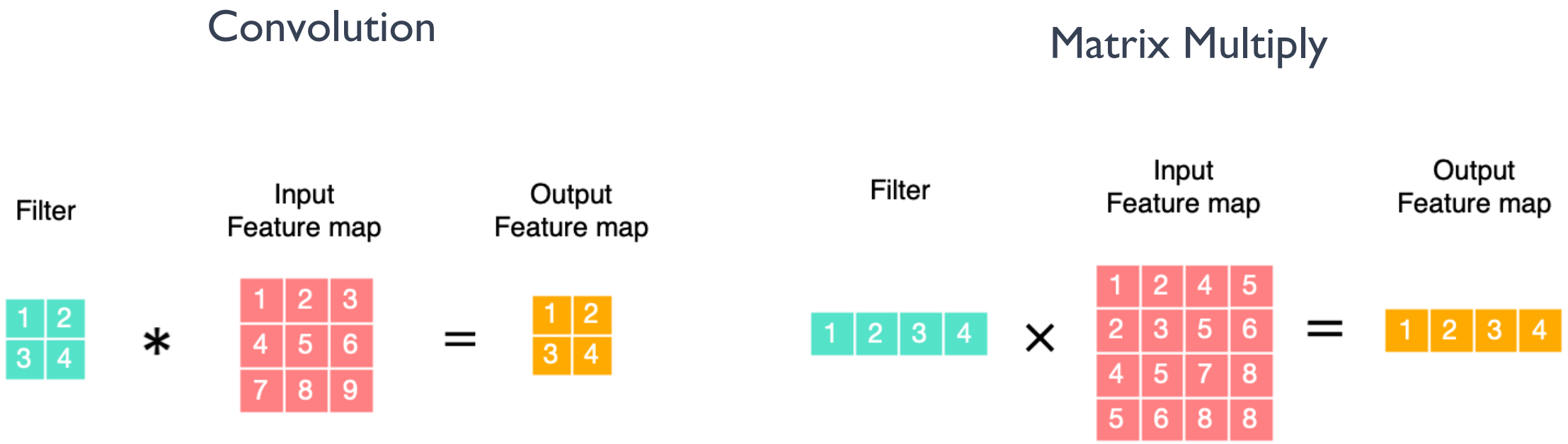
- Key Metrics – AI芯片关键指标
- Bit Width – 比特位数
- Matrix Multiplication – 矩阵运算
- Specialized Hardware – 专用硬件

从卷积 Conv 到矩阵乘 MM

- 通过数据重排，完成 Im2col 的操作之后会得到一个输入矩阵，卷积的 Weights 也可以转换为一个矩阵，卷积的计算就可以转换为两个矩阵相乘的求解，得到最终的卷积计算结果。



从卷积 Conv 到矩阵乘 MM



对于矩阵运算 计算体系思考 I

I. 软件 Software

- **减少没有必要的 MACs**
  - 使用其他代替算法
- **增加 PE 利用率**
  - 对kernel实现进行Loop优化
  - 对kernel实现进行Memory优化

I. 硬件 Hardware

- **减少每次 MAC 计算的时间**
  - 增加 PE 单元计算能力
- **增加 MACs 并行计算能力**
  - 增加片内 PE 数量
  - 支持低bits数PE计算
- **增加 PE 利用率**
  - 增加片内 Cache
  - 额外的内存带宽

Talk Overview

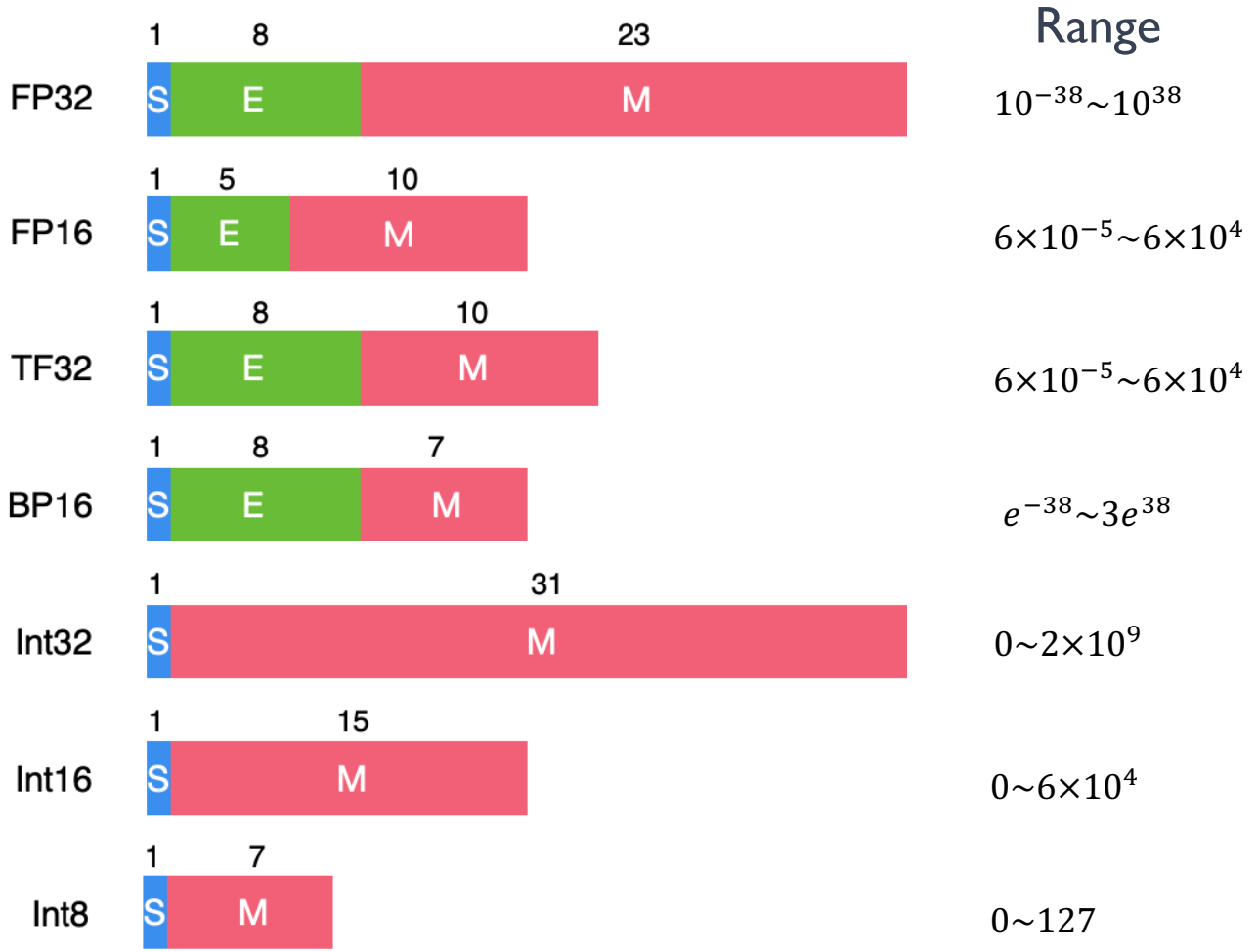
I. AI 计算体系与矩阵运算

- Key Metrics – AI芯片关键指标
- Bit Width – 比特位数
- Matrix Multiplication – 矩阵运算
- Specialized Hardware – 专用硬件

什么决定比特位宽 Bit Width ?

- Number of unique values Precision
  - e.g., M-bits to represent 2^M values
- Dynamic range of values
  - e.g., E-bits to scale value by $2^{(E-127)}$
- Signed or unsigned values
  - e.g., signed requires one extra bit(S)

- **总比特数： S+E+M**



AI 芯片设计的思考

- **对精度的影响 Impact on Accuracy**
  - 需要考虑不同数据集（NLP/CV）、不同任务
  - 不同网络模型之间的差异进行测评（e.g., classification > detection）
- **训练和推理的数据位宽**
  - 32bit float 可以作为弱基线；
  - 对于训练使用 FP16、BF16、TF32；
  - 推理 CV 任务以 int8 为主，NLP 以 FP16为主，大模型 int8/FP16 混合；
- **权衡硬件的成本开销**
  - 支持额外的数据位宽需要引入更多的电路
  - 新增多少额外的数据位宽合适？

Summary

1. 整体看看 AI or 深度学习计算模式：经典模型结构和轻量化模型结构、模型量化和剪枝到大模型分布式并行，从而理解“计算”需要什么。
2. 通过AI芯片关键指标，了解一块AI芯片要更好的支持“计算” ，需要关注那些重点工作；从而引出峰值算力和带宽之间的关系。
3. 最后通过深度学习的计算核心“矩阵乘”来看对“计算”的实际需求和情况，为了提升计算性能、降低功耗和满足训练推理不同场景应用，对“计算”引入 TF32/BF16 等复杂多样的比特位宽。