推理引擎-模型压缩

模型量化

ZOMI

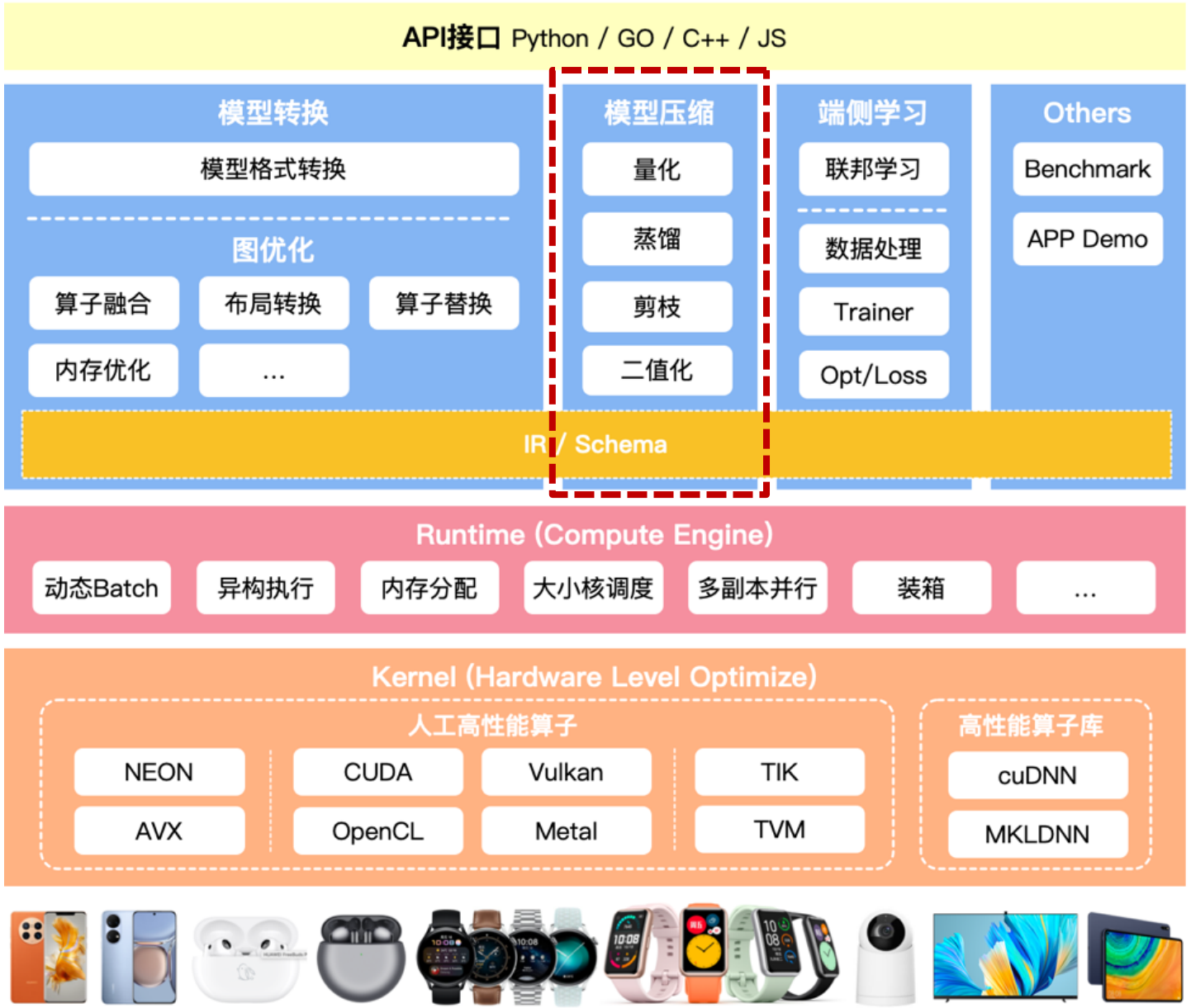
Talk Overview

1. **推理系统介绍**
   - 推理系统与推理引擎区别
   - 推理工作流程
   - 推理系统介绍
   - 推理引擎介绍
2. **模型小型化**
   - 基础参数概念
   - CNN小型化结构
   - Transform小型化结构
3. **离线优化压缩**
   - 低比特量化
   - 二值化网络
   - 模型剪枝
   - 知识蒸馏
4. **部署和运行优化**
   - 图转换优化（算子融合/重排/替换）
   - 并发执行与内存分配
   - 动态batch与bin Packing

推理引擎架构

对模型进行压缩

- 减少模型大小
- 加快推理速度
- 保持相同精度



Talk Overview

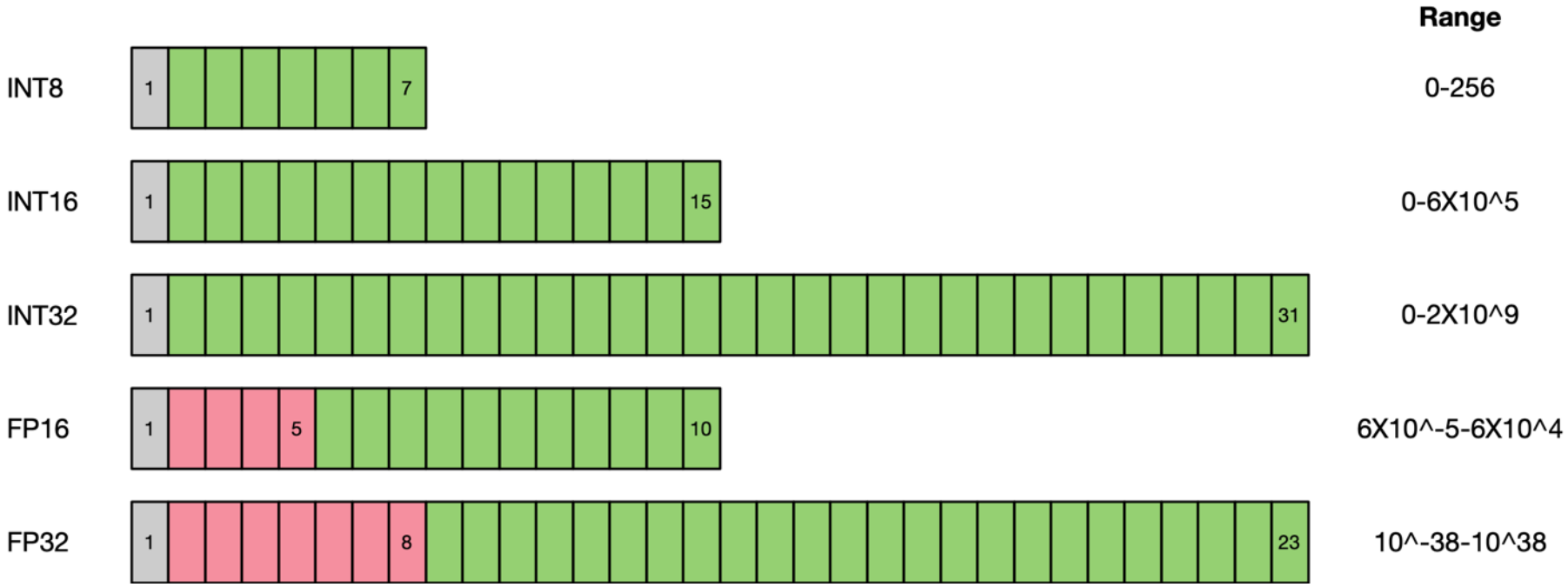
I. 低比特量化

- Base Concept of Quantization - 量化基础
- Quantization principle - 量化原理
- Quantization Aware Training - 感知量化（QAT）
- Post-Training Quantization - 训练后量化（PTQ）
- Deployment of Quantization - 量化部署

量化基础

模型量化

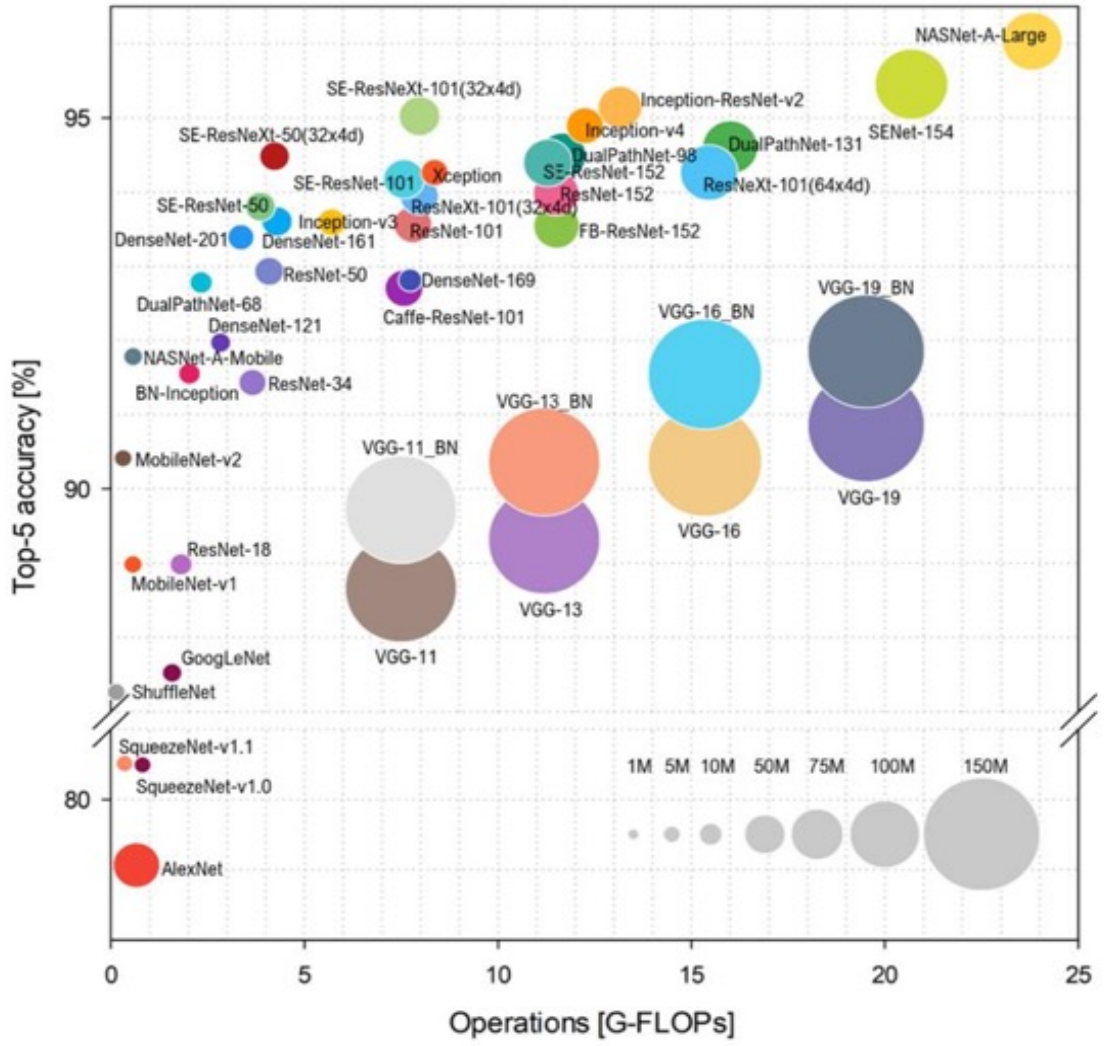
- 模型量化是一种将浮点计算转成低比特定点计算的技术，可以有效的降低模型计算强度、参数大小和内存消耗，但往往带来巨大的精度损失。尤其是在极低比特(<4bit)、二值网络(1bit)、甚至将梯度进行量化时，带来的精度挑战更大。



模型量化

神经网络模型特点：

1. 数据参数量大 ；
2. 计算量大；
3. 内存占用大；
4. 模型精度高；



模型量化优点

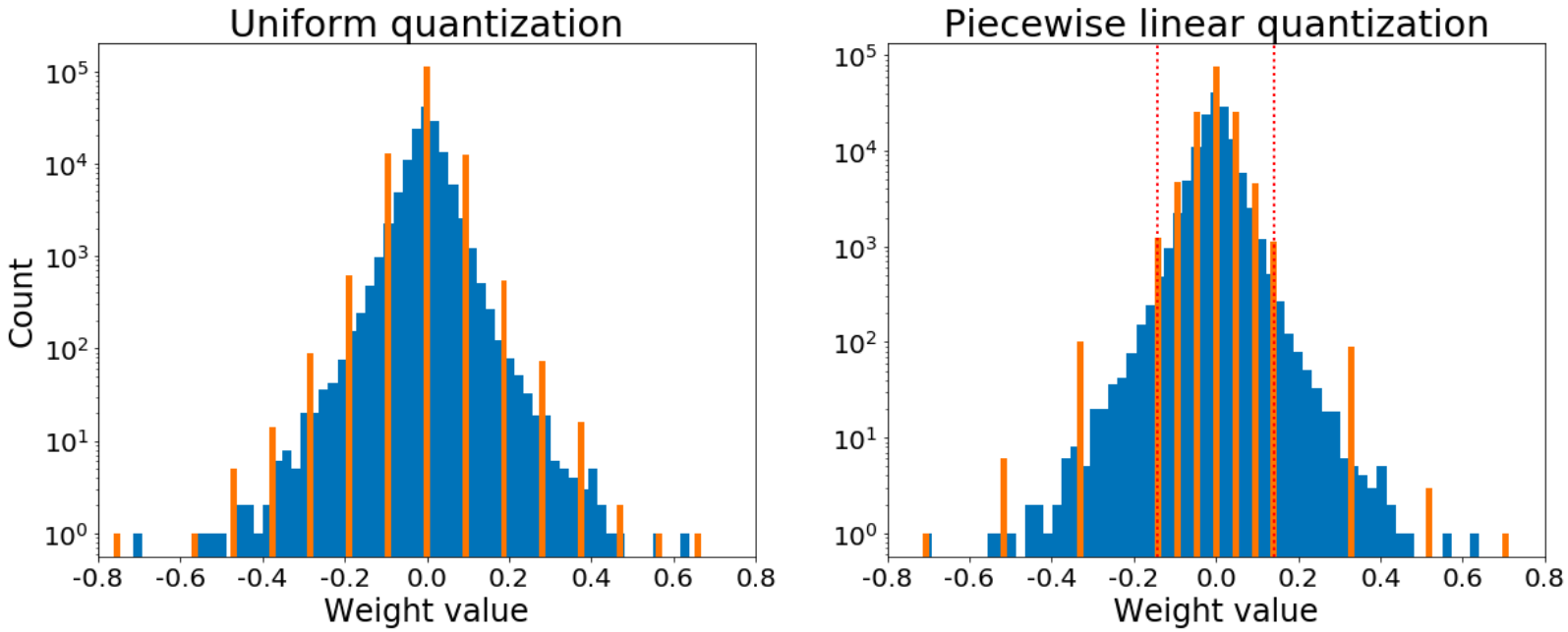
- **保持精度：**量化会损失精度，这相当于给网络引入了噪声，但是神经网络一般对噪声是不太敏感的，只要控制好量化的程度，对高级任务精度影响可以做到很小。
- **加速计算：**传统的卷积操作都是使用FP32浮点，低比特的位数减少少计算性能也更高，INT8 相对比 FP32 的加速比可达到3倍甚至更高

- **节省内存：**与 FP32 类型相比，FP16、INT8、INT4 低精度类型所占用空间更小，对应存储空间和传输时间都可以大幅下降。
- **节能和减少芯片面积：**每个数使用了更少的位数，做运算时需要搬运的数据量少了，减少了访存开销（节能），同时所需的乘法器数目也减少（减少芯片面积）

模型量化

模型量化特点：

1. 参数压缩；
2. 提升速度；
3. 降低内存；
4. 功耗降低；
5. 提升芯片面积；



量化技术落地的三大挑战

精度挑战

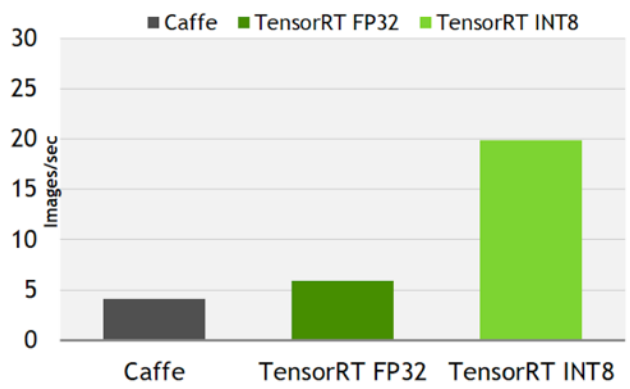
- **量化方法**：线性量化对数据分布的描述不精确
- **低比特**：从 16bits -> 4bits 比特数越低，精度损失越大
- **任务**：分类、检测、识别中任务越复杂，精度损失越大
- **大小**：模型越小，精度损失越大

量化技术落地的三大挑战

硬件支持程度

- 不同硬件支持的低比特指令不相同
- 不同硬件提供不同的低比特指令计算方式不同（PF16、HF32）
- 不同硬件体系结构Kernel优化方式不同

	CAFFE	TENSORRT FP32	TENSORRT INT8
Runtime (ms)	242	170	50
Images/sec	4	6	20
Class IoU	48.4	48.4	48.1
Category IoU	76.9	76.9	76.8

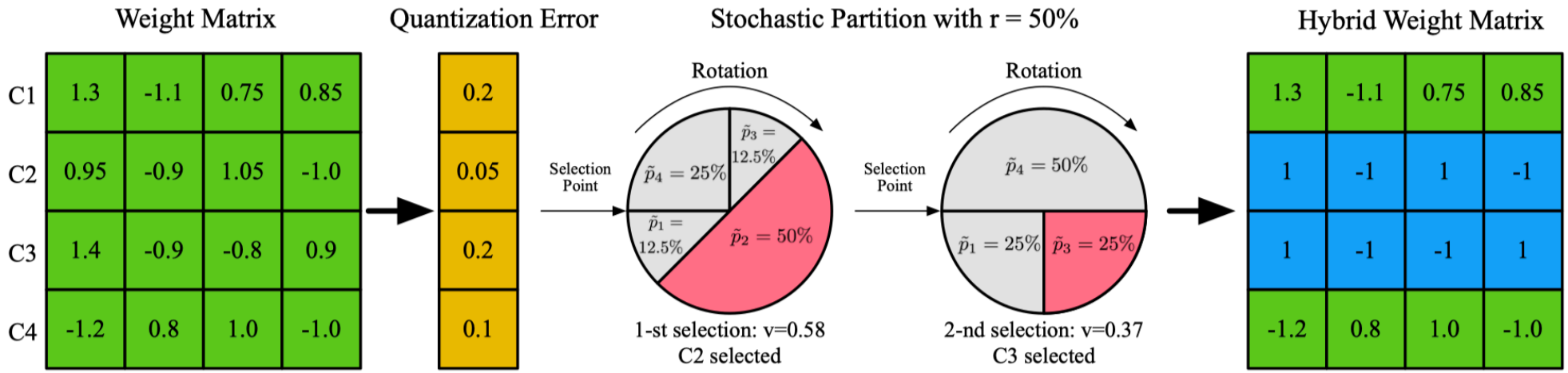


Batch Size = 1, Input/Output Resolution = 512 x 1024

量化技术落地的三大挑战

软件算法是否能加速

- 混合比特量化需要进行量化和反向量，插入 Cast 算子影响 kernel 执行性能
- 降低运行时内存占用，与降低模型参数量的差异
- 模型参数量小，压缩比高，不代表执行内存占用少



Question?

1. 为什么模型量化技术能够对实际部署起到加速作用？
2. 为什么需要对网络模型进行量化压缩？
3. 为什么不直接训练低精度的模型？（大模型呢？）
4. 什么情况下不应该/应该使用模型量化？

量化原理

量化方法

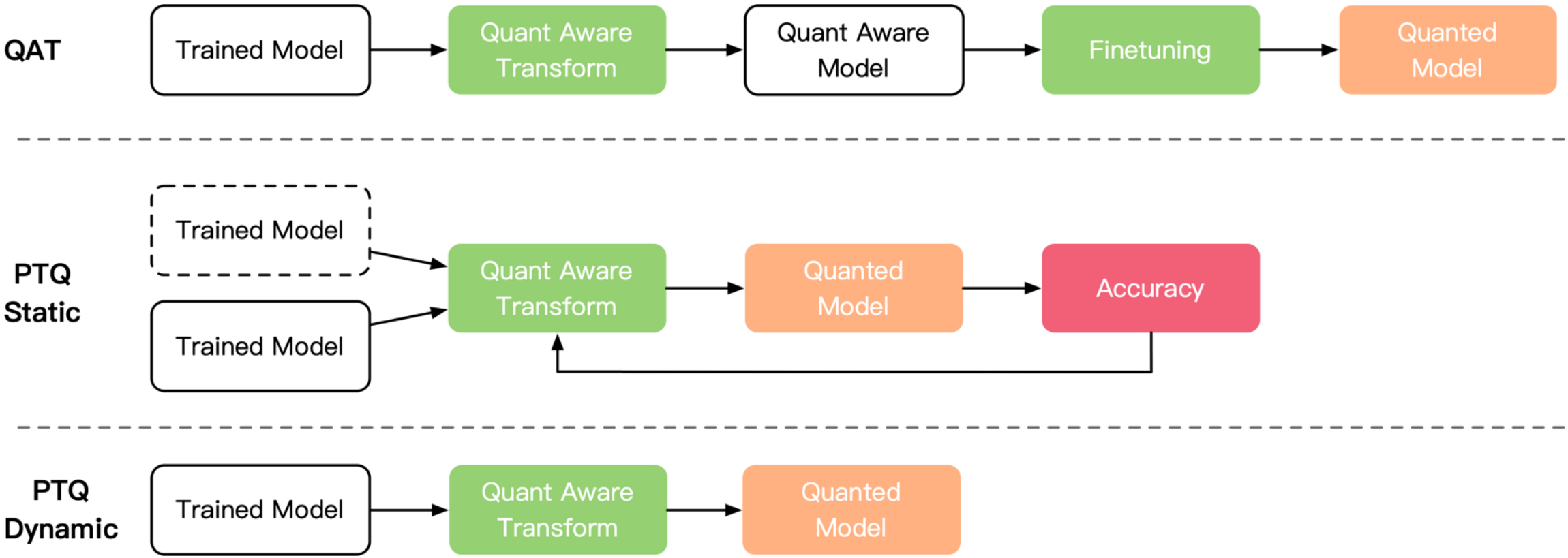
- **量化训练 (Quant Aware Training, QAT)**

  量化训练让模型感知量化运算对模型精度带来的影响，通过 finetune 训练降低量化误差。
- **动态离线量化 (Post Training Quantization Dynamic, PTQ Dynamic)**

  动态离线量化仅将模型中特定算子的权重从FP32类型映射成 INT8/16 类型。
- **静态离线量化 (Post Training Quantization Static, PTQ Static)**

  静态离线量化使用少量无标签校准数据，采用 KL 散度等方法计算量化比例因子。

量化方法

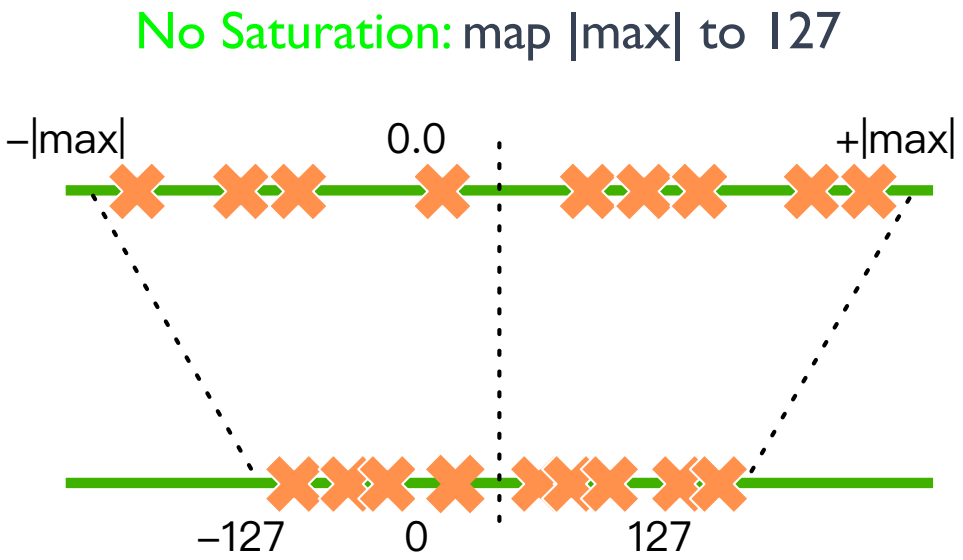


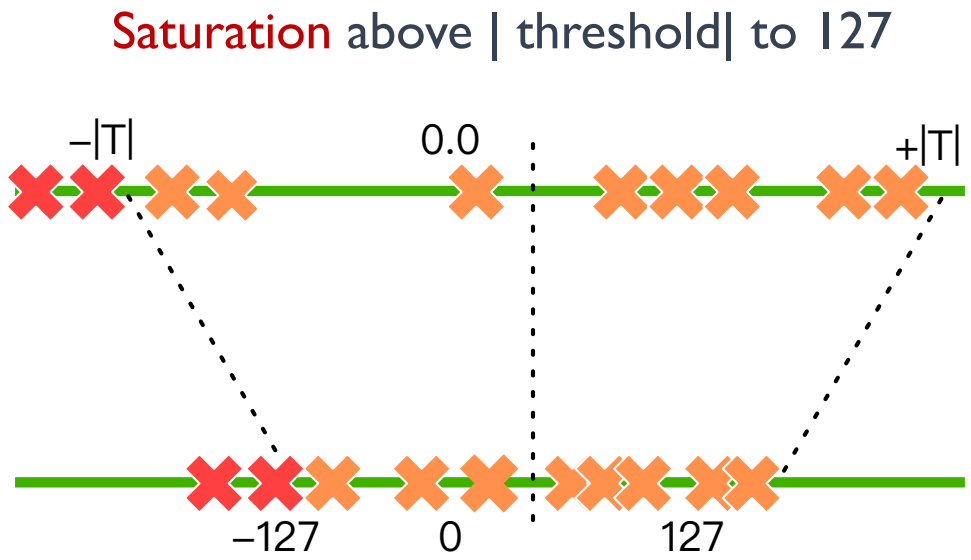
量化方法比较

量化方法	功能	经典适用场景	使用条件	易用性	精度损失	预期收益
量化训练 (QAT)	通过 Finetune 训练将模型量化误差降到最小	对量化敏感的场景、模型，例如目标检测、分割、OCR 等	有大量带标签数据	好	极小	减少存续空间4X，降低计算内存
静态离线量化 (PTQ Static)	通过少量校准数据得到量化模型	对量化不敏感的场景，例如图像分类任务	有少量无标签数据	较好	较少	减少存续空间4X，降低计算内存
动态离线量化 (PTQ Dynamic)	仅量化模型的可学习权重	模型体积大、访存开销大的模型，例如 BERT 模型	无	一般	一般	减少存续空间2/4X，降低计算内存

量化原理

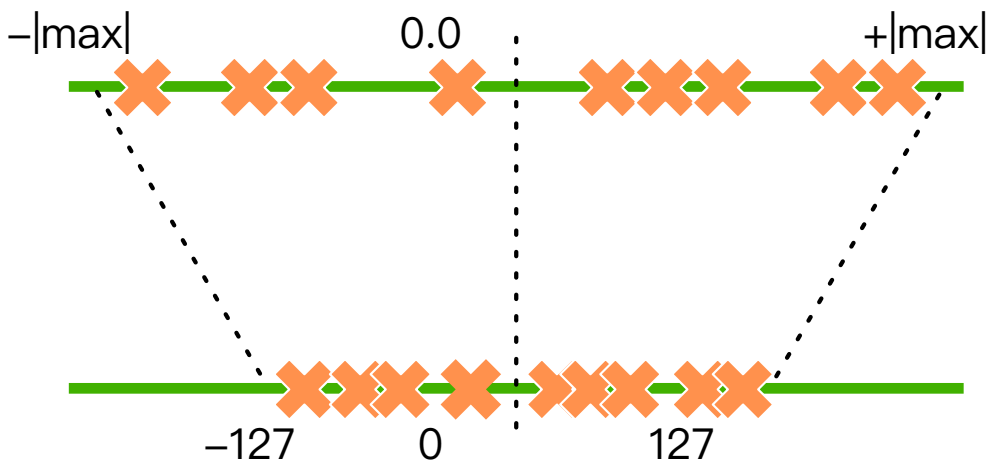
- 模型量化桥接了定点与浮点，建立了一种有效的数据映射关系，使得以较小的精度损失代价获得了较好的收益

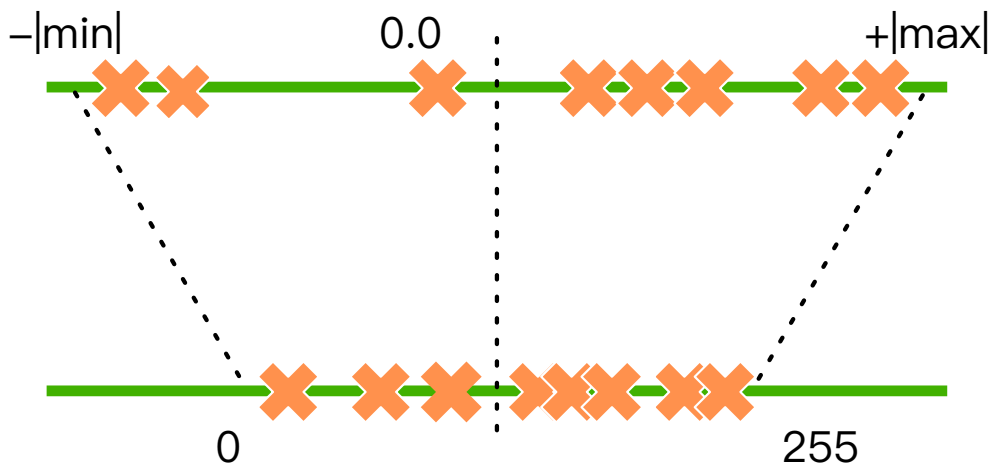




量化类型

- 线性量化可分为对称量化和非对称量化





量化原理

- 要弄懂模型量化的原理就是要弄懂这种数据映射关系，浮点与定点数据的转换公式如下：

$$Q = \frac{R}{S} + Z$$

$$R = (Q - Z) * S$$

- R 表示输入的浮点数据
- Q 表示量化之后的定点数据
- Z 表示零点（Zero Point）的数值
- S 表示缩放因子（Scale）的数值

量化原理

- 求解 S 和 Z 有很多种方法，这里列举中其中一种线性量化的求解方式（MinMax）如下：

$$S = \frac{R_{\max} - R_{\min}}{Q_{\max} - Q_{\min}}$$

$$Z = Q_{\max} - R_{\max} \div S$$

- Rmax 表示输入浮点数据中的最大值
- Rmin 表示输入浮点数据中的最小值
- Qmax 表示最大的定点值（127 / 255）
- Qmin 表示最小的定点值（-128 / 0）

量化原理

- 量化算法原始浮点精度数据与量化后 INT8 数据的转换如下：

$$float = scale \times (uint + offset)$$

- 确定后通过原始float32高精度数据计算得到uint8数据的转换即为如下公式所示：

$$uint8 = round(float/scale) - offset$$

- 若待量化数据的取值范围为 $[Xmin, Xmax]$，则 $scale$ 的计算公式如下：

$$scale = \left(x_{\max} - x_{\min}\right) / \left(Q_{\max} - Q_{\min}\right)$$

- offset的计算方式如下：

$$offset = Q_{\min} - round\left(x_{\min}/scale\right)$$

参考文献

1. Learning Accurate Low-Bit Deep Neural Networks with Stochastic Quantization
- Differentiable Soft Quantization: Bridging Full-Precision and Low-Bit Neural Networks（ICCV 2019）
- IR-Net: Forward and Backward Information Retention for Highly Accurate Binary Neural Networks（CVPR 2020）
- Towards Unified INT8 Training for Convolutional Neural Network（CVPR 2020）
- Rotation Consistent Margin Loss for Efficient Low-bit Face Recognition（CVPR 2020）
- DMS: Differentiable diMension Search for Binary Neural Networks（ICLR 2020 Workshop）
- Nagel, Markus, et al. "A white paper on neural network quantization." *arXiv preprint arXiv:2106.08295* (2021).
- Krishnamoorthi, Raghuraman. "Quantizing deep convolutional networks for efficient inference: A whitepaper." arXiv preprint arXiv:1806.08342 (2018)
- 全网最全-网络模型低比特量化 https://zhuanlan.zhihu.com/p/453992336