推理引擎-模型压缩

感知量化训练-QAT

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
   - 模型模型剪枝
   - 模型模型蒸馏
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

Talk Overview

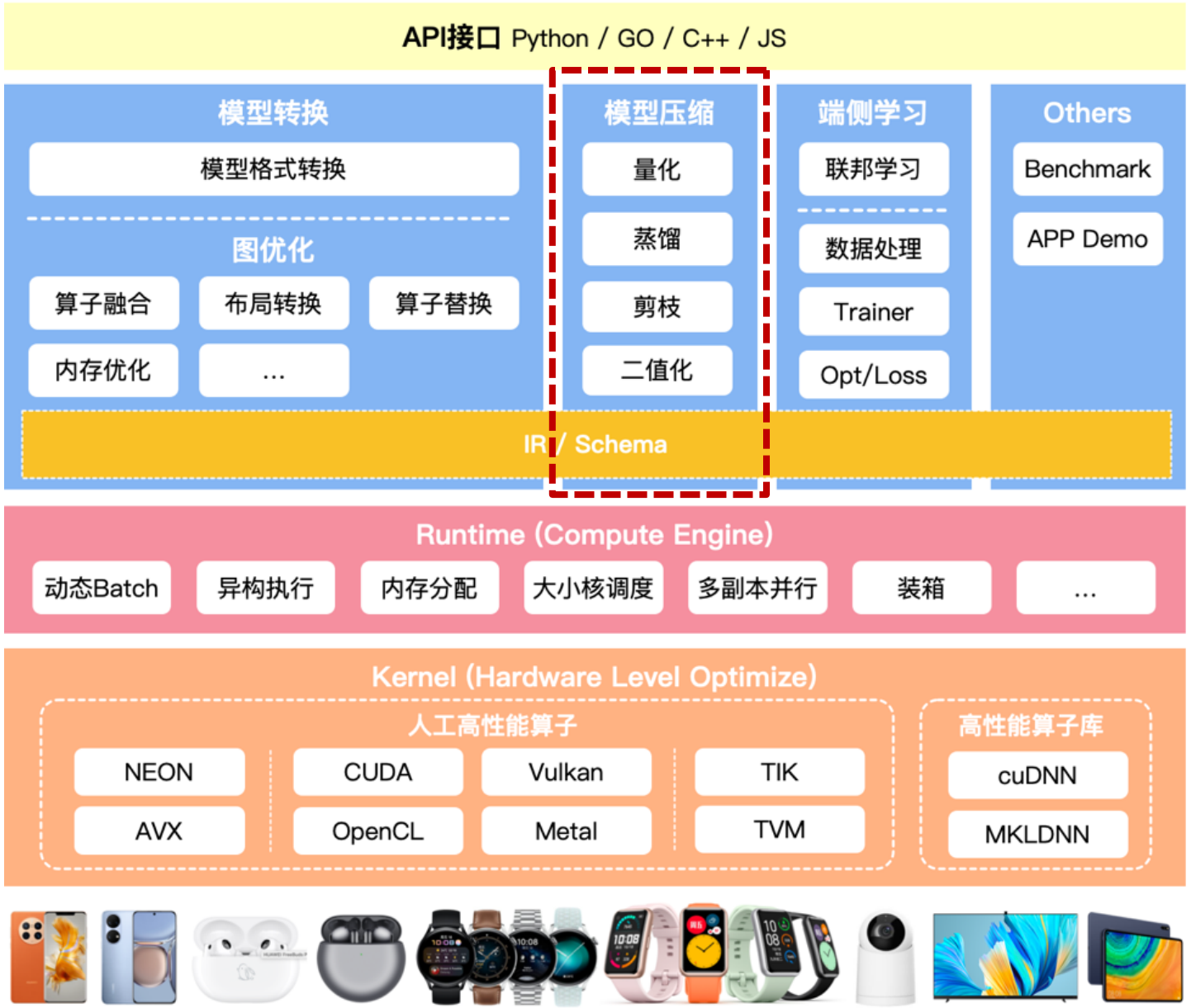
I. 低比特量化

- 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

推理引擎架构

对模型进行压缩

- 减少模型大小
- 加快训练速度
- 保持相同精度

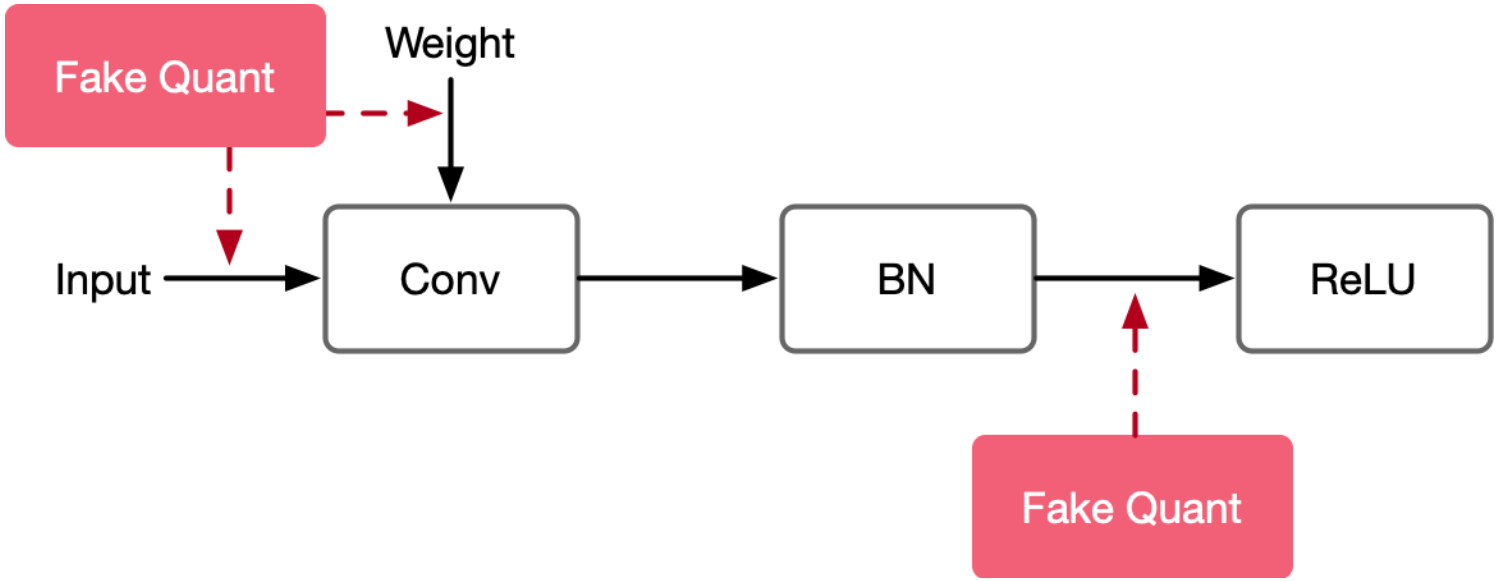


感知量化训练

Quantization Aware Training, QAT

QAT

- 感知量化训练（Aware Quantization Training）模型中插入伪量化节点fake quant来模拟量化引入的误差。端测推理的时候折叠fake quant节点中的属性到tensor中，在端测推理的过程中直接使用tensor中带有的量化属性参数。



伪量化节点

1. 找到输入数据的分布，即找到 min 和 max 值；
2. 模拟量化到低比特操作的时候的精度损失，把该损失作用到网络模型中，传递给损失函数，让优化器去在训练过程中对该损失值进行优化。

伪量化节点：正向传播 Forward

- 为了求得网络模型tensor数据精确的Min和Max值，因此在模型训练的时候插入伪量化节点来模拟引入的误差，得到数据的分布。对于每一个算子，量化参数通过下面的方式得到：

$$\text{clamp}\left(x, x_{\min}, x_{\max}\right) := \min\left(\max\left(x, x_{\min}\right), x_{\max}\right)$$

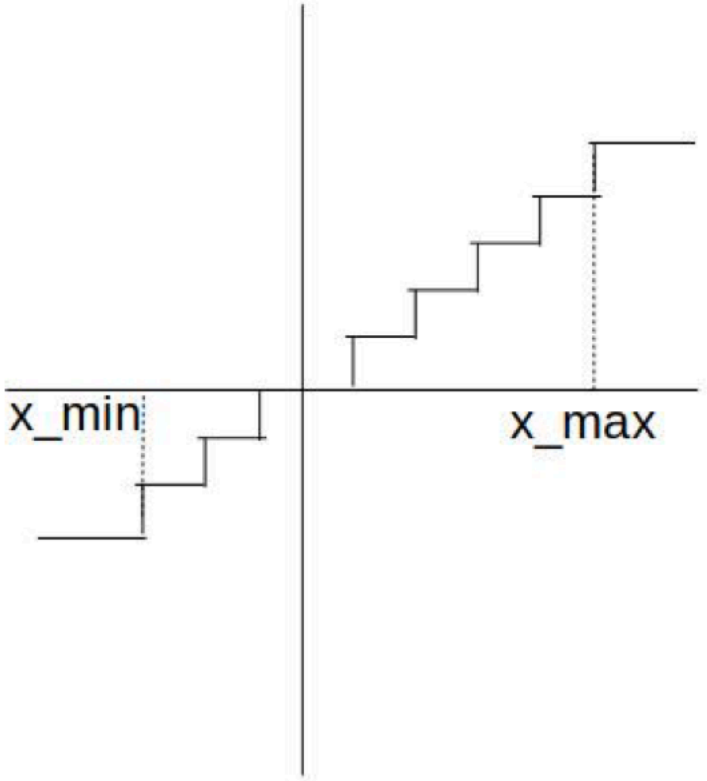
$$Q = \frac{R}{S} + Z$$

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

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

伪量化节点：正向传播 Forward

- 正向传播的时候fake quant节点对数据进行了模拟量化规约的过程，如下图所示：



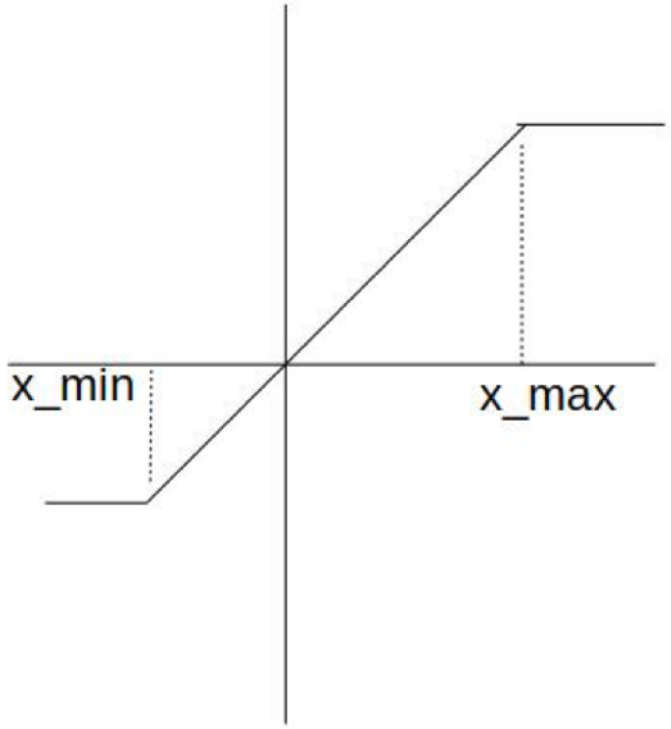
伪量化节点：反向传播 Backward

- 按照正向传播的公式，如果方向传播的时候对其求导数会导致权重为0，因此反向传播的时候相当于一个直接估算器：

$$\delta_{out} = \delta_{in}, I_{(x \in S)} \in S : x : x_{\min} \leq x \leq x_{\max}$$

伪量化节点：反向传播 Backward

- 最终反向传播的时候fake quant节点对数据进行了截断式处理，如下图所示：



伪量化节点：更新Min和Max

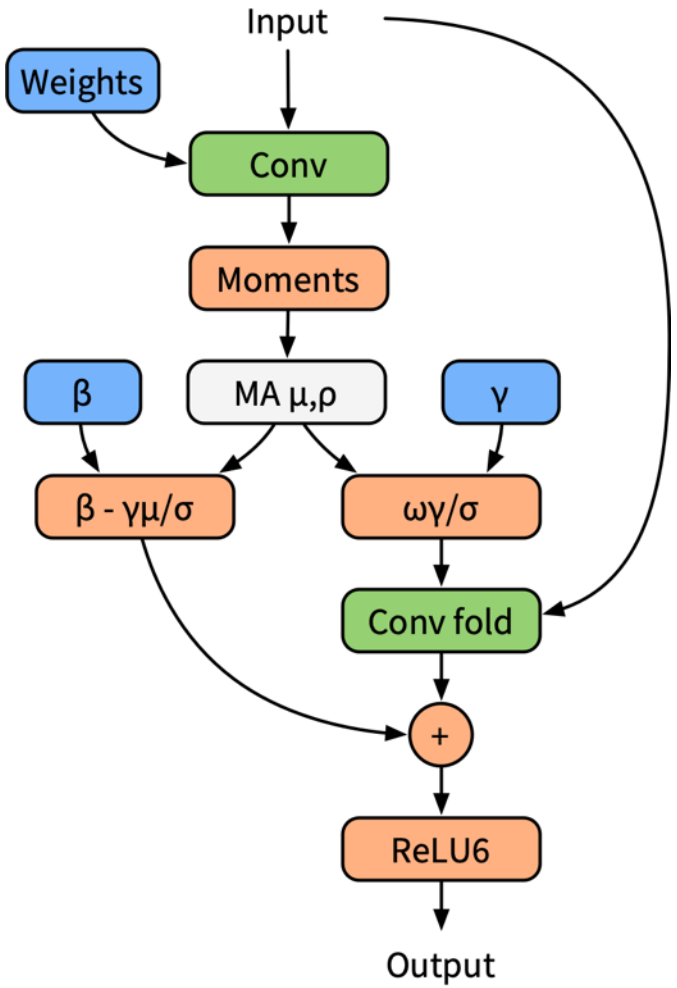
- FakeQuant伪量化节点主要是根据找到的min和max值进行伪量化操作，更新min和max分别为running和moving，跟batch normal中更新 beta 和 gamma 算子相同。

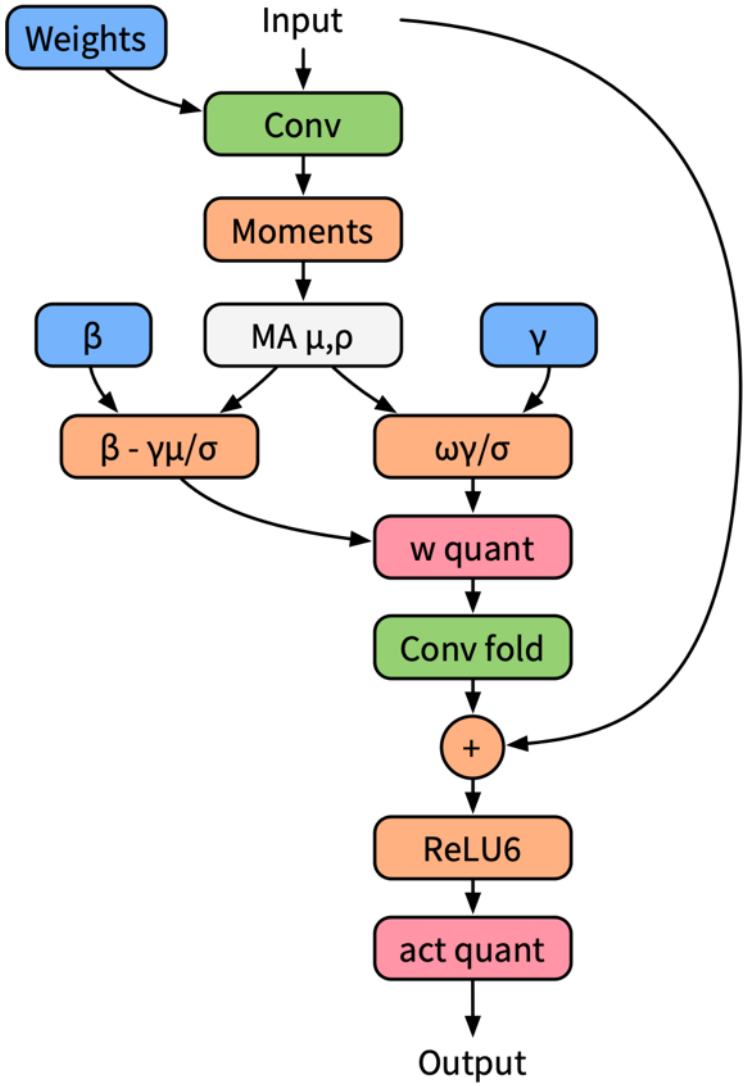
Question?

- 在什么地方插入 Fake Quant 伪量化节点？

- 一般会在密集计算算子、激活算子、网络输入输出等地方插入伪量化节点

伪量化节点：插入方式





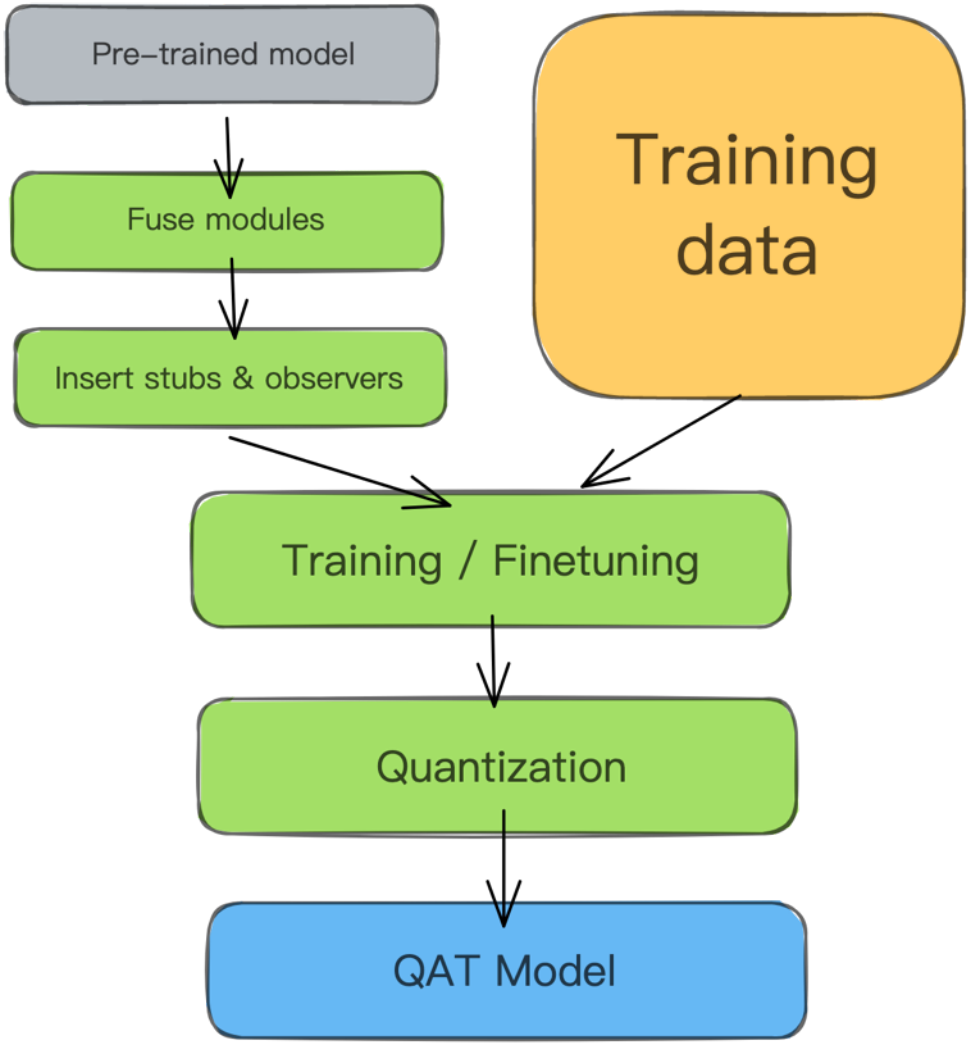
Question?

1. 如何平滑计算伪量化阶段的Min和Max？
2. 我看论文中有 Batch Normal 矫正和 Bessel 校正，具体为什么要进行校正？
3. 如果要对 Batch Normal 进行折叠，那么计算公式或者 kernel 会变成什么样？

AI框架工作流程

Quantization Aware Training, QAT

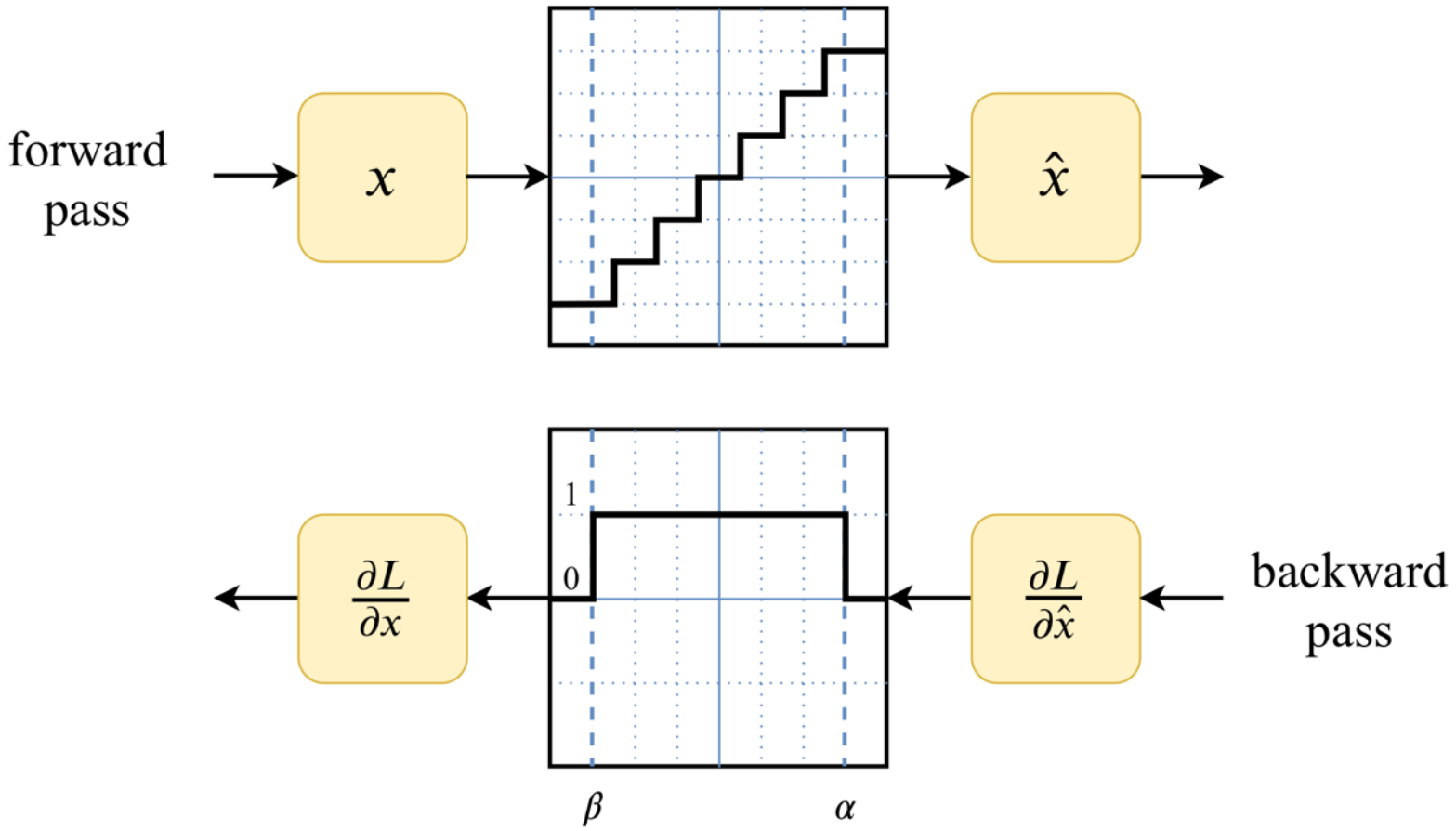
AI framework Workflow



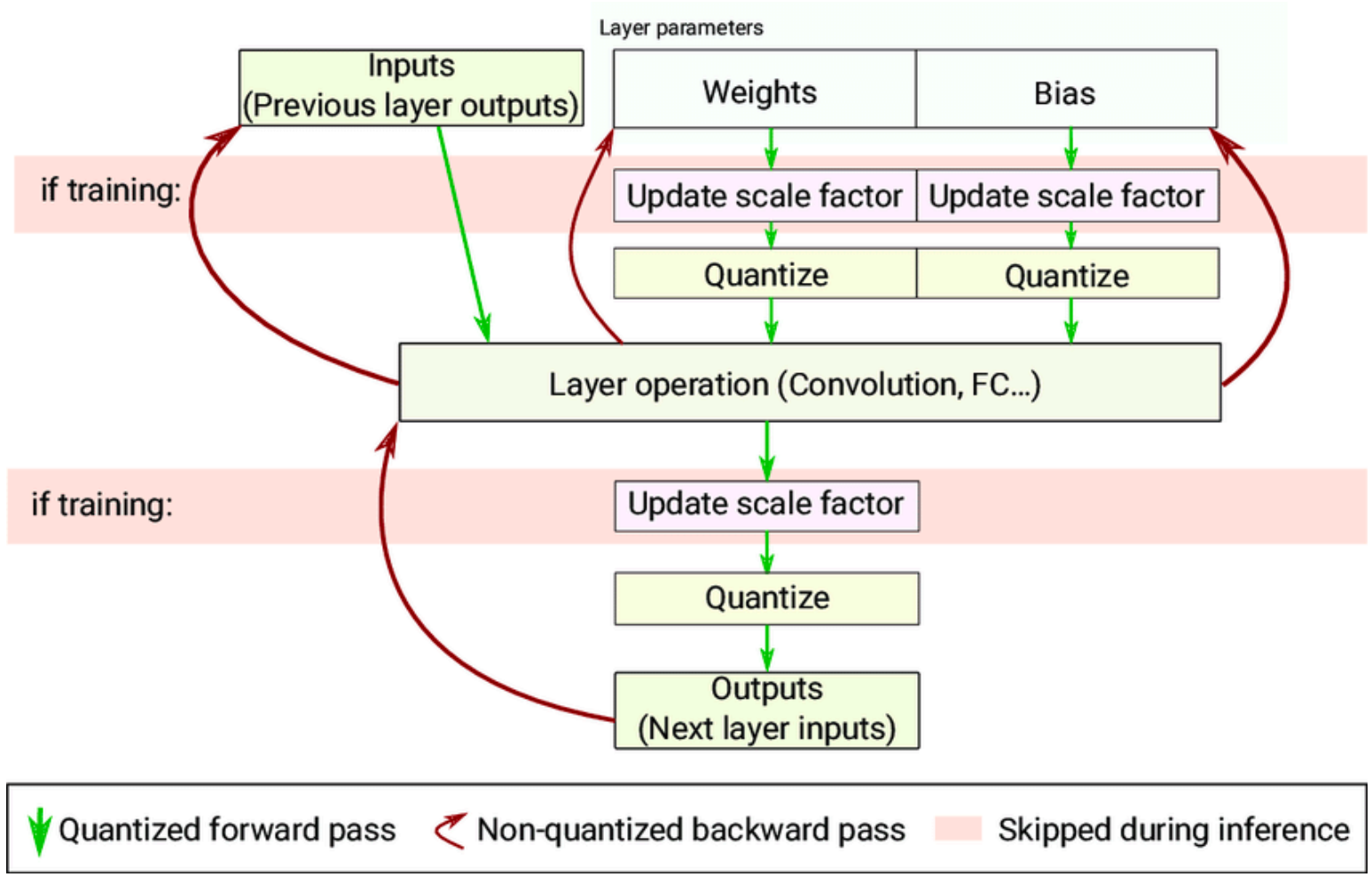
QAT衍生研究

Quantization Aware Training, QAT

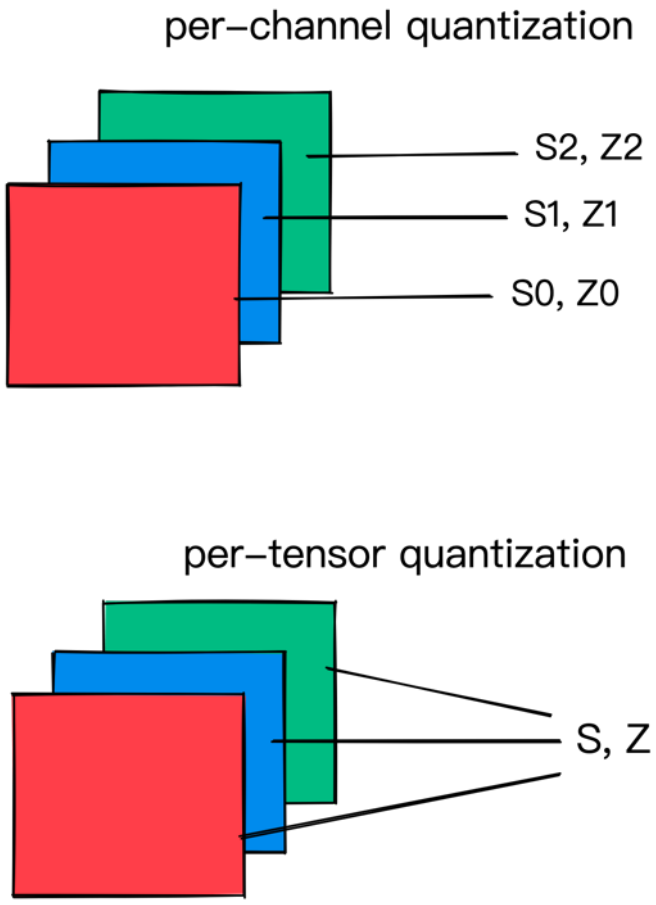
Straight Through Estimation Derivative Approximation

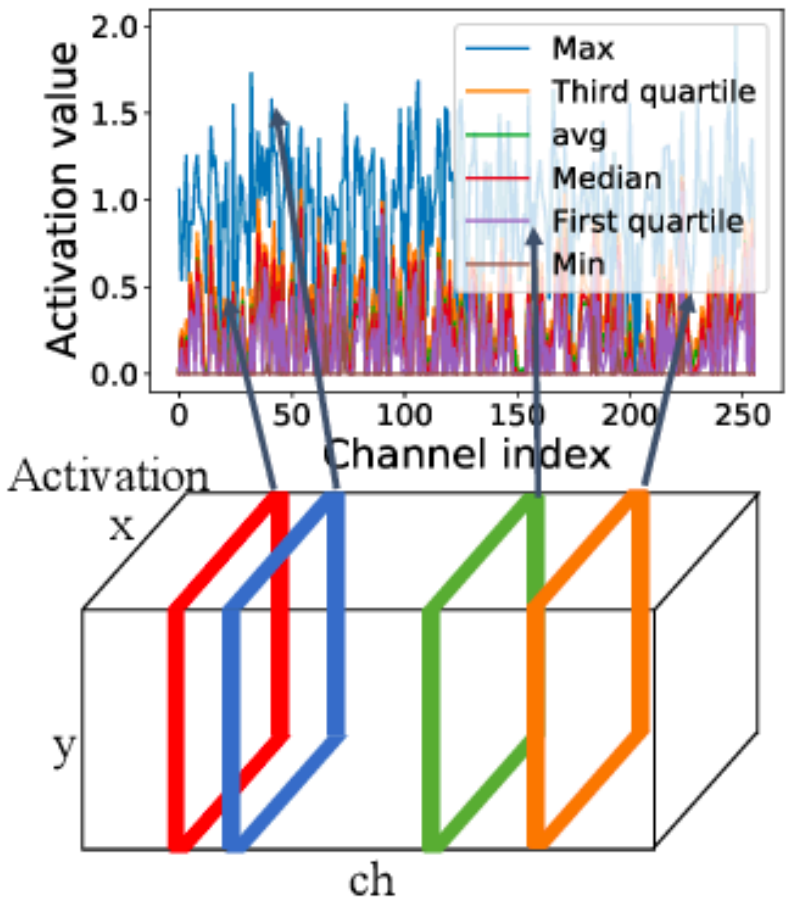


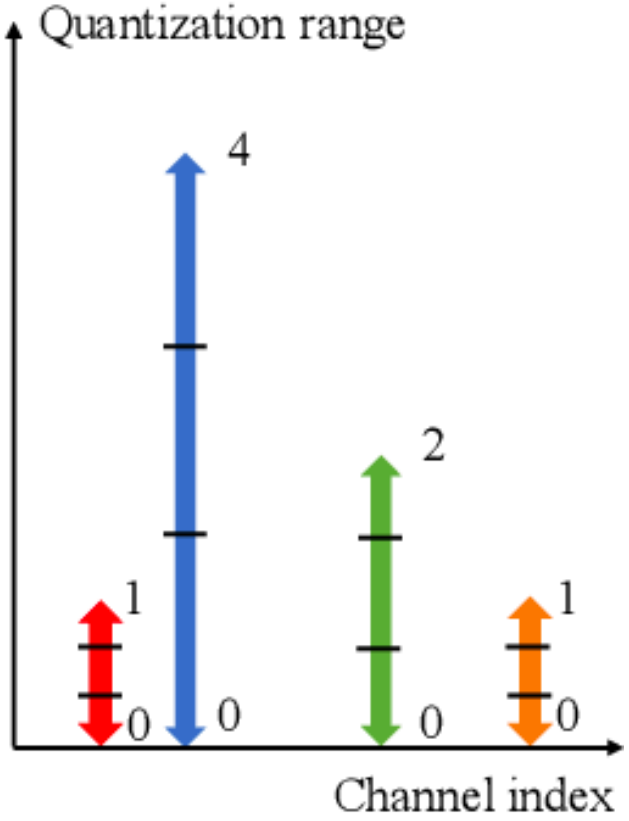
Quantization and Deployment of Deep Neural Networks on Microcontrollers



Per-channel Quantization Level Allocation for Quantizing Convolutional Neural Networks







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