A Survey of Quantization Methods for Efficient Neural Network Inference

Low-power computer vision. Chapman and Hall/CRC, 2022

University of California, Berkeley

01. Basic Concepts of Quantization

02. Advanced Concepts: Quantization Below 8-bits

01. Simulated and Integer-only Quantization

02. Mixed-Precision Quantization

03. Hardware Aware Quantization

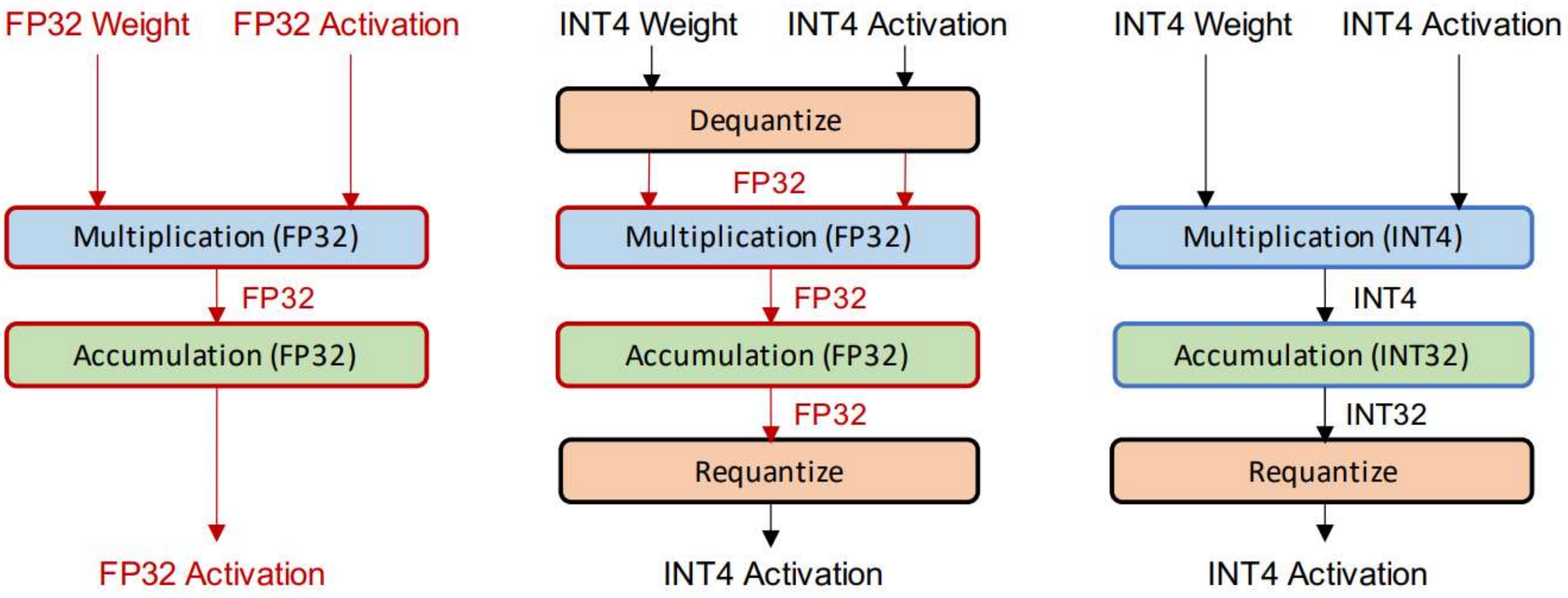
04. Distillation-Assisted Quantization

05. Extreme Quantization

06. Vector Quantization

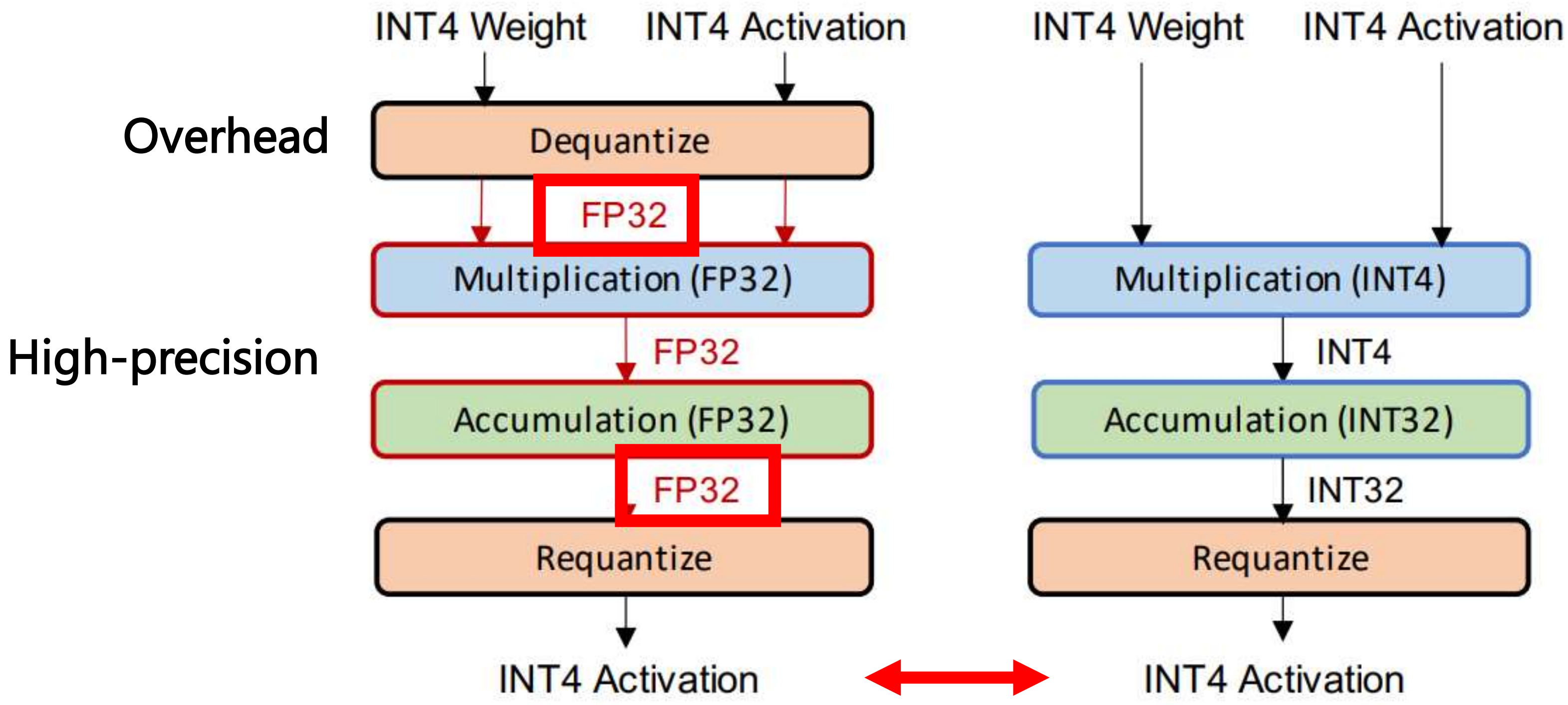
Simulated and Integer-Only Quantization

- Simulated quantization
  - Save model parameters to low-precision
  - Operation is performed by floating point
- Integer-only quantization: all operations are performed with low-precision



Simulated and Integer-Only Quantization

- Note that the objective of quantization is to make de-quantized output similar with original FP output after calibration
- Integer-only quantization: it's mathematically the same as simulated quantization without de-quantization



Simulated and Integer-Only Quantization

- Note that the objective of quantization is to make de-quantized output similar with original FP output after calibration
- Integer-only quantization: it's mathematically the same as simulated quantization without de-quantization

$$\mathrm{Y} = XW$$
$$= S_x X_q S_w W_q$$
$$= S_x S_w X_q W_q$$

$$Y = S_y Y_q$$

$$\mathrm{Y_q} = (S_x S_w / S_y) X_q W_q$$

$(S_x S_w / S_y)$: implemented by bit shifting

$S_x, S_w, S_y : scale\ factor$

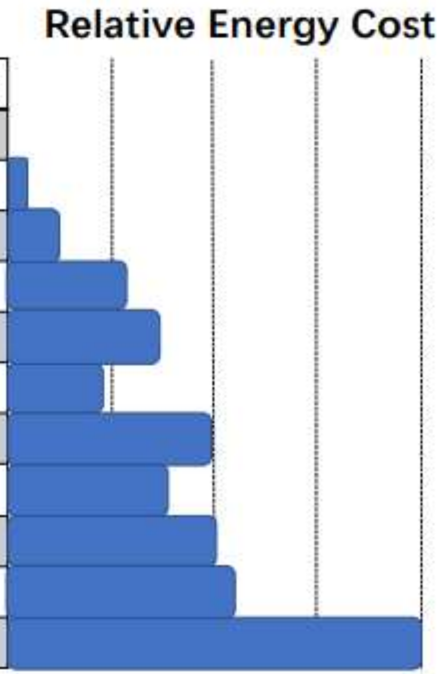
$X_q, W_q, Y_q : quantized\ values$

Tensor-wise & Symmetric quantization

Simulated and Integer-Only Quantization

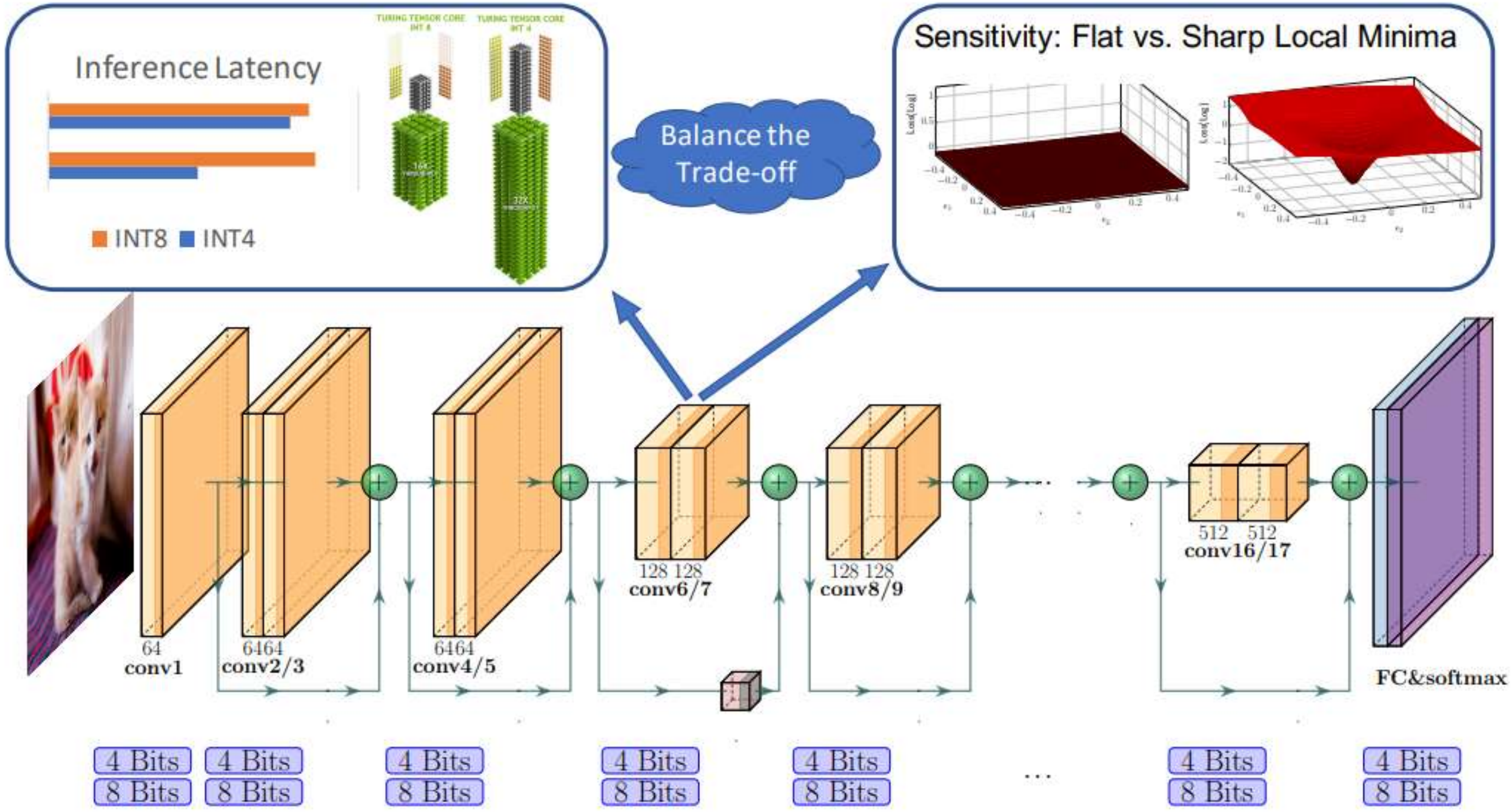
- CMSIS-NN is a library from ARM that helps quantizing and deploying NN models onto the ARM Cortex-M cores (fixed-point quantization with power of two scaling factors)
- Low-precision provides exponentially better energy efficiency
  - E.g., 8b Add vs 32b Add

Operation:	Energy(pJ):
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FP Add	0.4
32b FP Add	0.9
8b Mult	0.2
32b Mult	3.1
16b FP Mult	1.1
32b FP Mult	3.7
32b SRAM Read (8kb)	5.0
32b DRAM Read	640



Mixed-Precision Quantization

- Quantize each layer to a different precision
  - Important and sensitive layer: higher precision
  - Inefficient and robust layer: lower precision



Mixed-Precision Quantization

- Selecting mixed-precision of each layer is searching problem
  - RL
  - NAS (Neural Architecture Search)
  - Etc.

Hardware Aware Quantization

- One of the objectives of quantization is to improve the inference latency
- However, quantizing certain layer/operation doesn't result in the same speedup on all hardware
- It is important to perform quantization considering hardware to obtain optimal performance

Distillation-Assisted Quantization

- Improve accuracy of quantization utilizing model distillation

Extreme Quantization

- Quantized values are constrained to a 1-bit representation (memory requirements by 32 x)
- **BinaryConnect**: the method limiting the weight +1 or -1
- Ternery-Binary Network (TBN): +1, 0, -1

Vector Quantization

- Apply classical quantization method to NN
- E.g., cluster weights into multiple groups and apply them in inference using the centroid of each group as quantized values

$$\min_{c_1,\ldots,c_k} \sum_i \|w_i - c_j\|^2$$

Future Directions

1. Quantization software: INT8 vs below INT8
   - It is important to deploy API assisting lower precision
2. Hardware and NN Architecture Co-Design
   - By changing the width of NN, can improve generalization performance of quantization
   - Tune architecture parameters, such as depth and individual kernels, in the quantization process
3. Coupled Compression Methods: Quantization + Pruning/Knowledge distillation
4. Quantized Training
   - Accelerating NN learning with FP16 is an example of successful quantized training
   - It is still difficult to expand to INT8 level