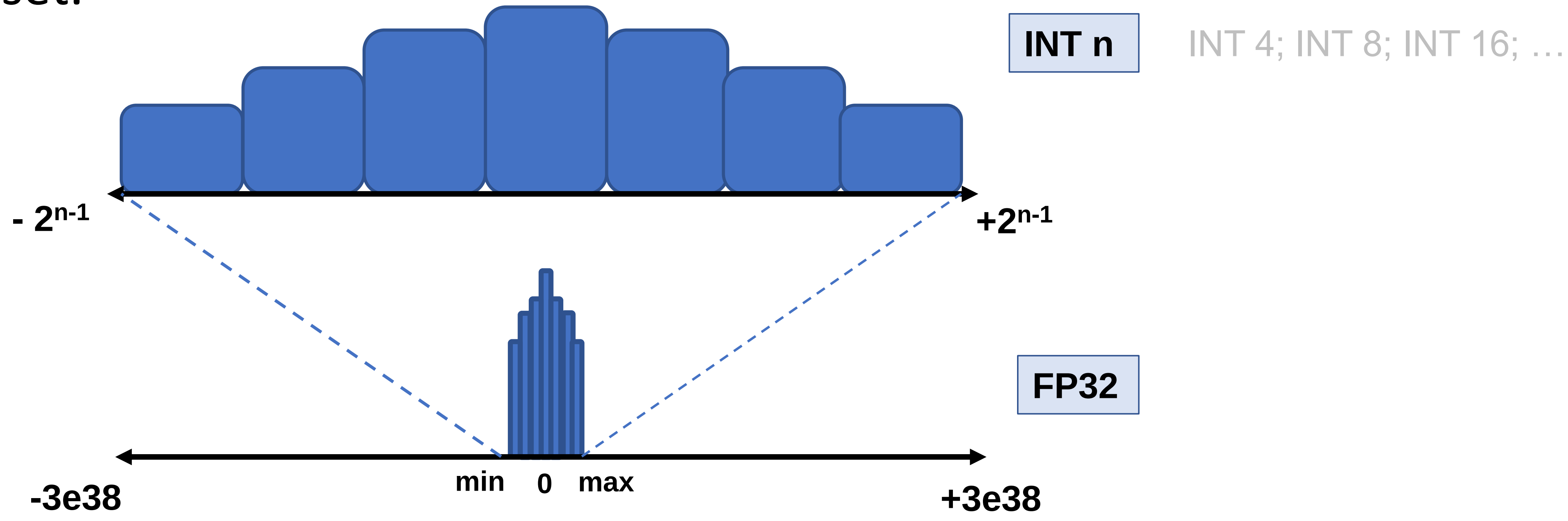
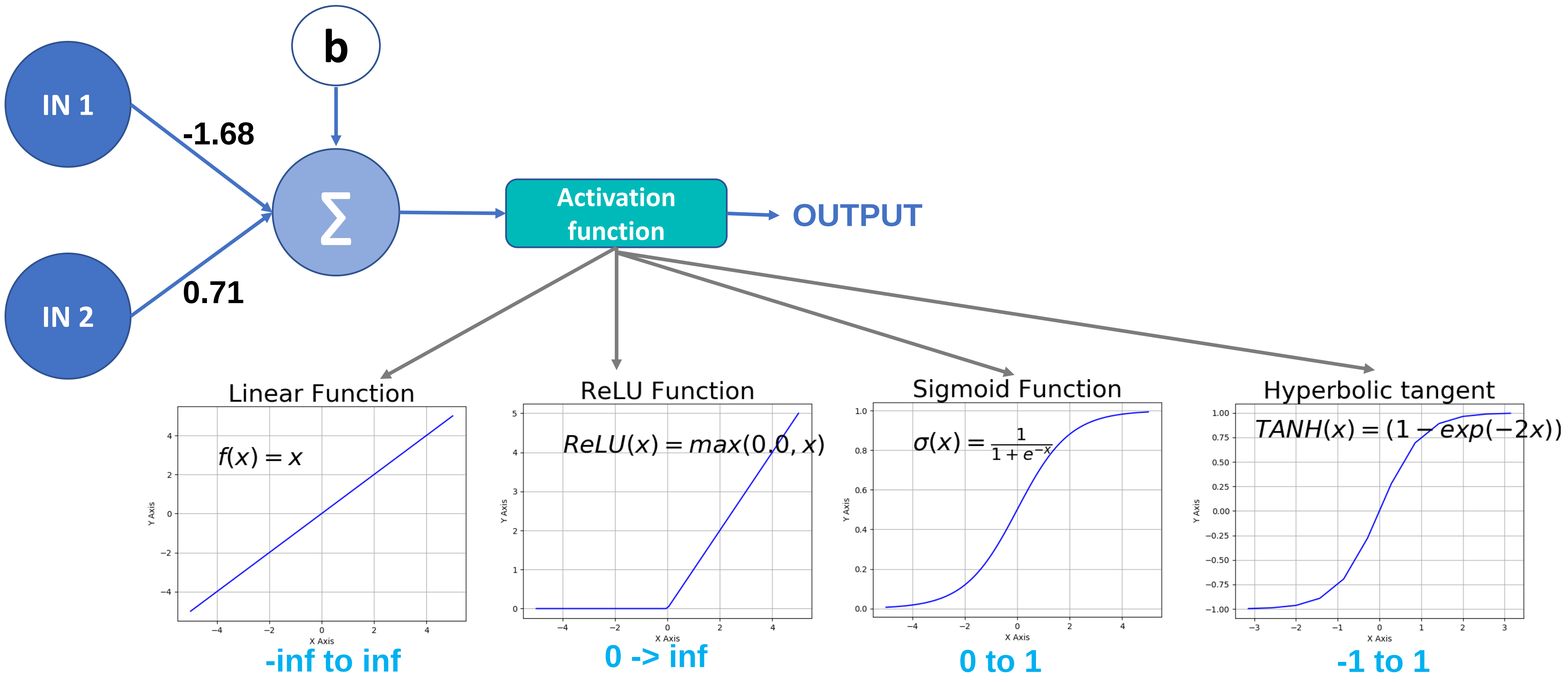
Quantization

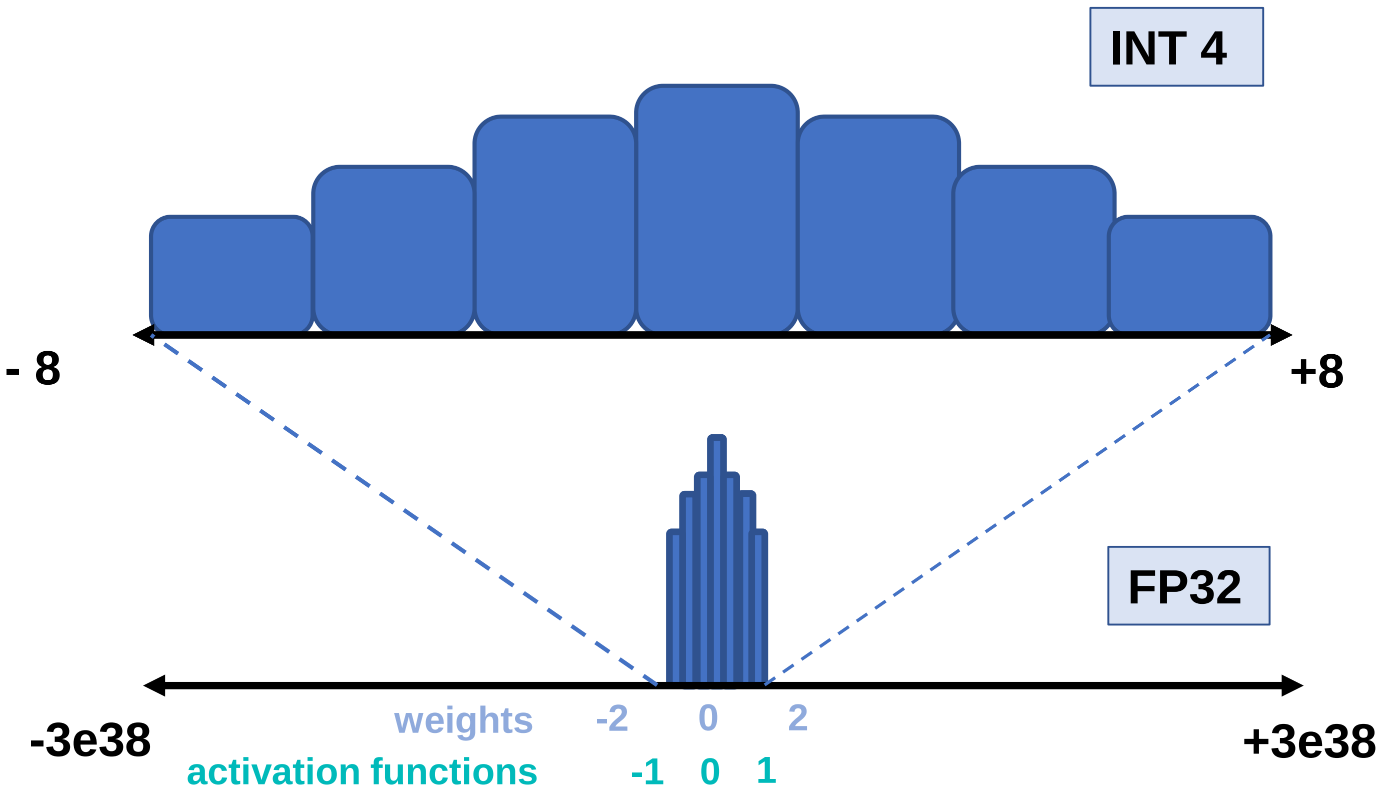
- The process of constraining an input from a continuous set of values to a discrete set.



Weights and Activation Functions



EXAMPLE – INT4 QUANTIZE



QUANT WEIGHTS

- scale by **4**
- round to nearest int

-2 to 2 → -8 to 8

round (-1.68 * 4) = round (- 6.72) = -7

QUANT ACTIVATIONS

- scale by **8**
- round to nearest int

-1 to 1 → -8 to 8

round (-0.3 * 8) = round (- 2.4) = -2

RESULTS

scale by 4*8 = 32

INITIAL RESULT: 0.73
AFTER INT4 QUANT: round (0.73 * 32) = round(23.36) = 23
DEQUANTIZE: 23 / 32 = 0.718

Quantization

Why does it work?

- DNNs are robust to noise and small perturbations.
- Weights and activations tend to lie in a small range which can be estimated
- Small losses in accuracy can be recovered by retraining the quantized models

Advantages

- 4x memory reduction (FP32 -> INT8)
- Arithmetic with lower bit-depth is faster
- Less RAM accesses -> less power and time
- FP arithmetic is not always available on embedded devices.

QUANTIZATION TYPES

- Weights
- Weights + activations

- INT4
- INT 8
- FP16

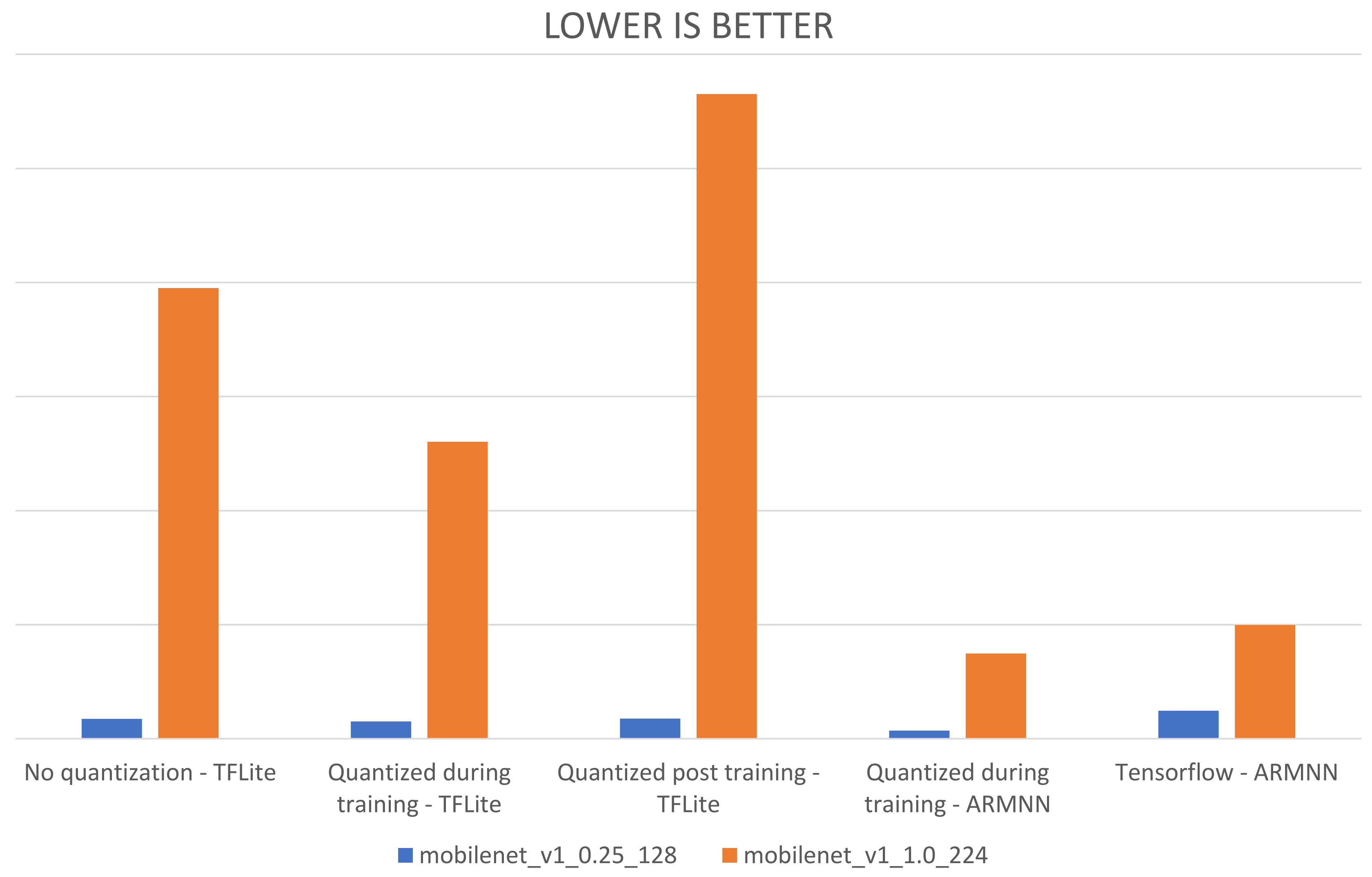
- FULL
- HYBRID

- Post training quantization
- Quantization aware training
  - Train the model in a way that considers quantization -> simulate quantization
  - Match precision for both training and inference

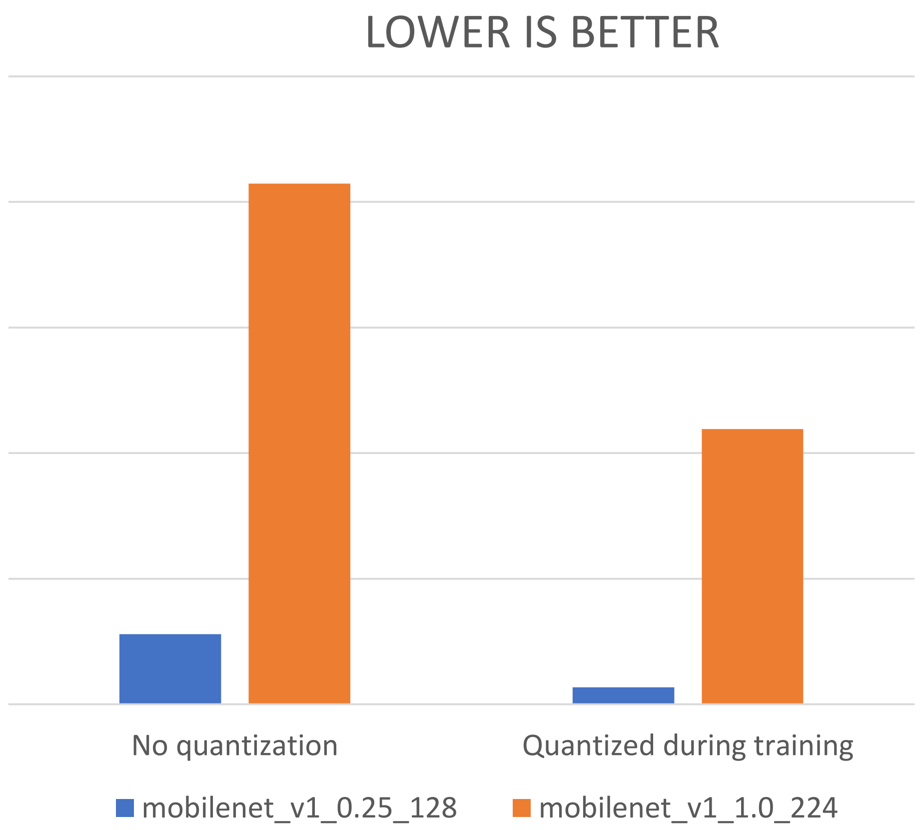
TFLite Conversion and Quantization Analysis

HW Accelerator	INFERENCE	MODEL TYPE	SIZE (KB)	
			mobilenet_v1_0.25_128	mobilenet_v1_1.0_224
CPU	TF Lite	TFLite no quantization	1840	16506
	TF Lite	TFLite quantized during training	486	4177
	TF Lite	TFLite quantized post training	489	4178
	ARMNN	TFLite quantized during training	486	4177
	ARMNN	Tensorflow	1923	16685
GPU	TF Lite	TFLite no quantization	1840	16506
	TF Lite	TFLite quantized during training	486	4177

CPU | TFLite, ARMNN



GPU | TFLite



CPU | ARMNN