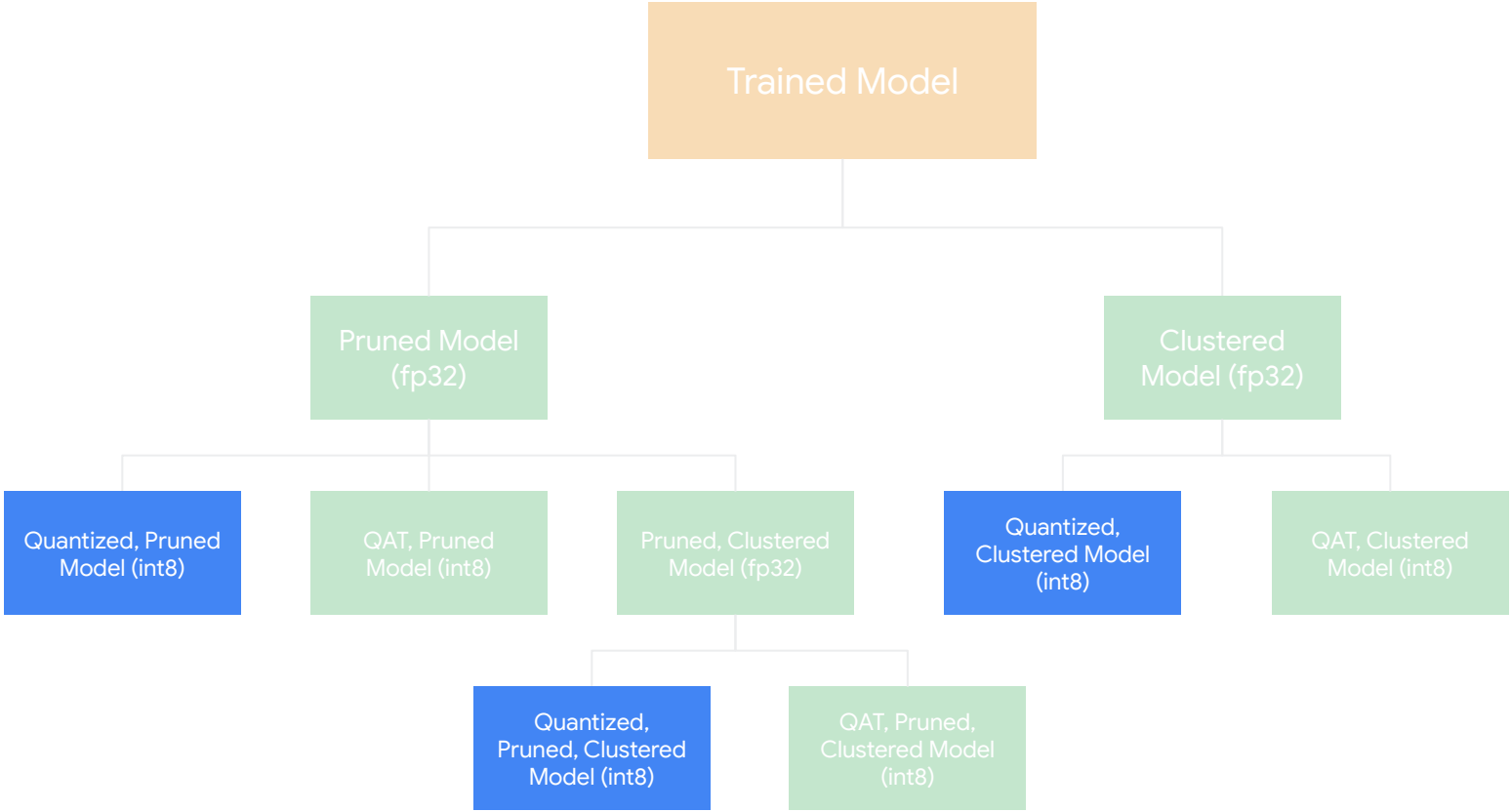
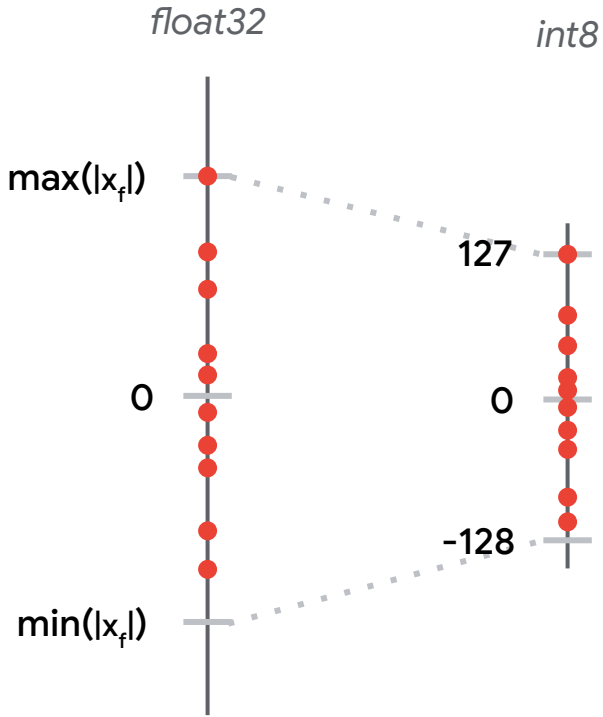
Model Optimizations: Quantization

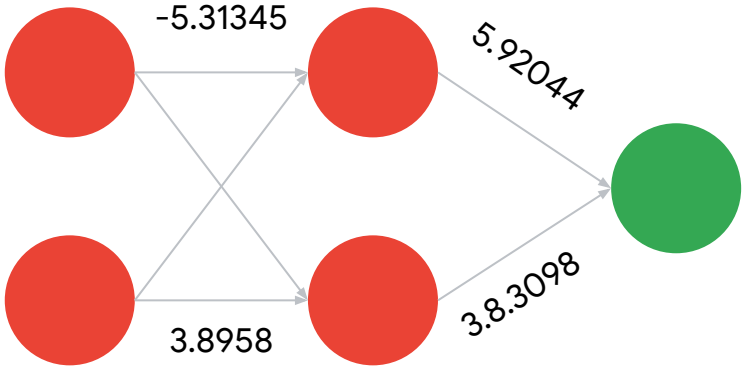


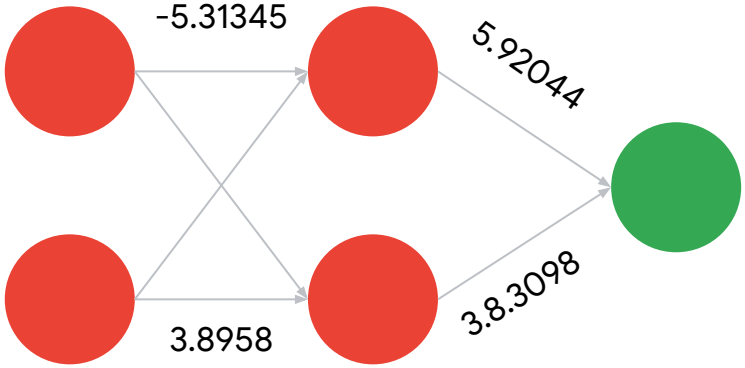
Quantization

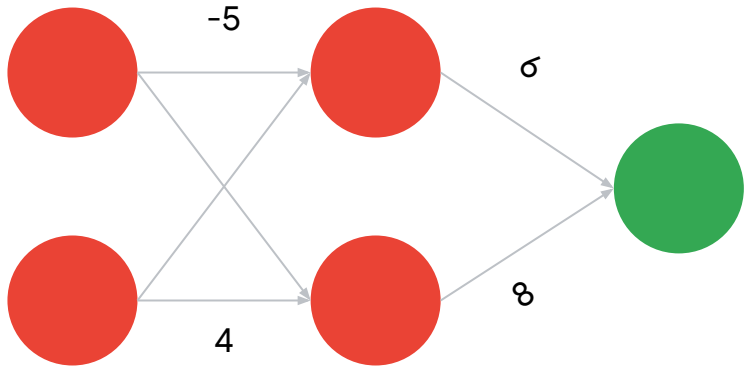
Quantization is the process of transforming an ML program into an approximated representation with available lower precision operations.

Quantization









Lower Precision Efficiencies

Reduce Memory

8-bit integer parameters means 4x smaller model

Faster Compute

Integer operations are faster

Reduce Power

Integer operations consume less power

Reduce Bandwidth

Smaller models and dynamic values reduce bandwidth pressure

Hardware Compatibility

Integer operations supported across CPU/DSP/NPUs

Lower Precision Efficiencies

Reduce Memory

8-bit integer parameters means 4x smaller model

Lower Precision Efficiencies

Reduce Memory

8-bit integer parameters means 4x smaller model

Faster Compute

Integer operations are faster

Lower Precision Efficiencies

Reduce Memory

8-bit integer parameters means 4x smaller model

Faster Compute

Integer operations are faster

Reduce Power

Integer operations consume less power

Lower Precision Efficiencies

Reduce Memory

8-bit integer parameters means 4x smaller model

Faster Compute

Integer operations are faster

Reduce Power

Integer operations consume less power

Reduce Bandwidth

Smaller models and dynamic values reduce bandwidth pressure

Lower Precision Efficiencies

Reduce Memory

8-bit integer parameters means 4x smaller model

Faster Compute

Integer operations are faster

Reduce Power

Integer operations consume less power

Reduce Bandwidth

Smaller models and dynamic values reduce bandwidth pressure

Hardware Compatibility

Integer operations supported across CPU/DSP/NPUs

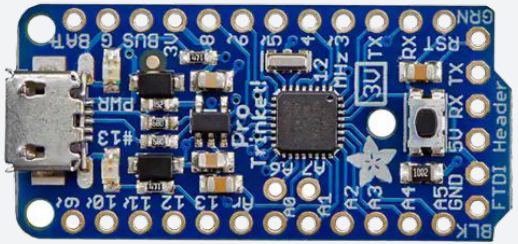
Why Quantization is Necessary

AVR microcontroller

- Manufacturer: Atmel
- ATmega328P-PU MCU

Features

- Core size: 8-bit
- Speed: up to 20MHz
- Flash memory size: 32Kb (16K x 16)
- RAM size: 2K x 8



Quantization Types

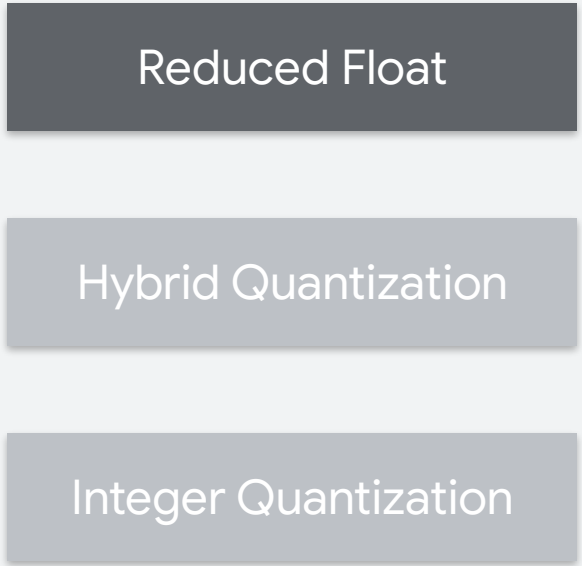
Reduced Float

Hybrid Quantization

Integer Quantization

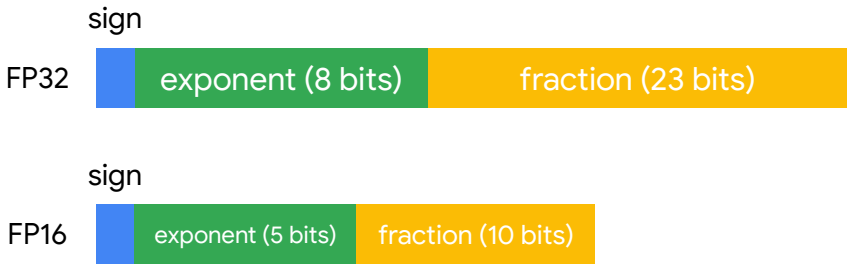
Reduced float

- float16 parameters
- float16 computations



Reduced float

- float16 parameters
- float16 computations



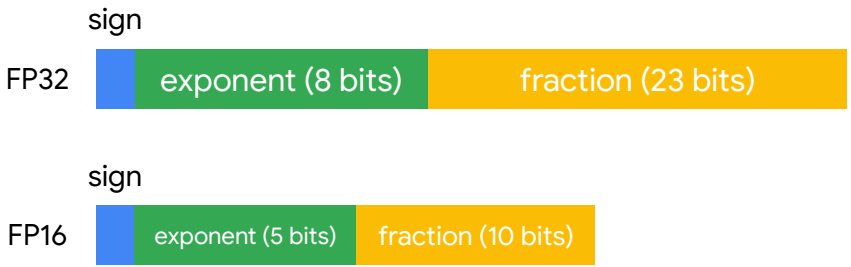
Reduced Float

Hybrid Quantization

Integer Quantization

Reduced float

- float16 parameters
- float16 computations



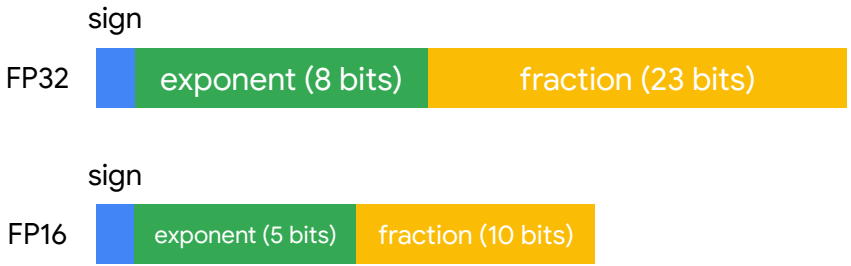
Benefits

2x reduction in model parameters

(32 bit ⟶ 16 bit)

Reduced float

- float16 parameters
- float16 computations



Benefits

2x reduction in model parameters (32 bit ⟶ 16 bit)

Potential future speed-ups faster as hardware enables optimized operations

Negligible accuracy loss

Quantization Types

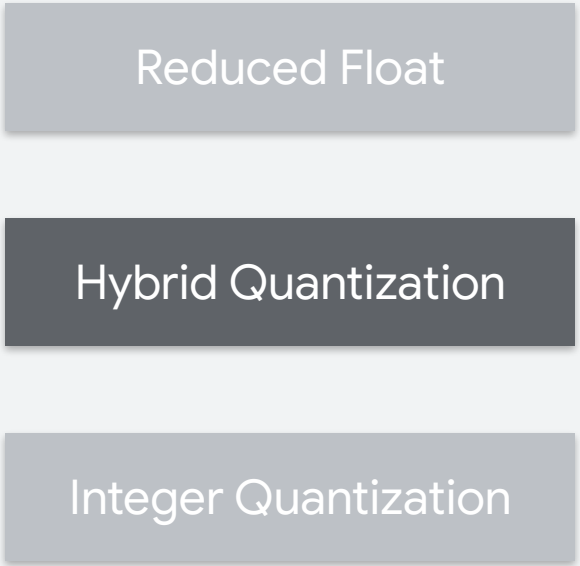
Reduced Float

Hybrid Quantization

Integer Quantization

Hybrid quantization

- 8-bit integer weights
- 32-bit float biases & activations
- Integer and floating point computations



Hybrid quantization

- 8-bit integer weights
- 32-bit float biases & activations
- Integer and floating point computations

Benefits

4x reduction in model parameters (32 bit ⟶ 8 bit)

Hybrid quantization

- 8-bit integer weights
- 32-bit float biases & activations
- Integer and floating point computations

Benefits

4x reduction in model parameters (32 bit ⟶ 8 bit)

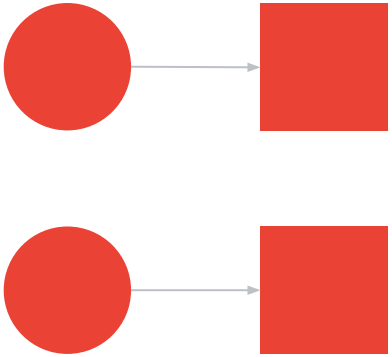
10-50% faster execution for Convolution Models (CPU hybrid v. CPU float)

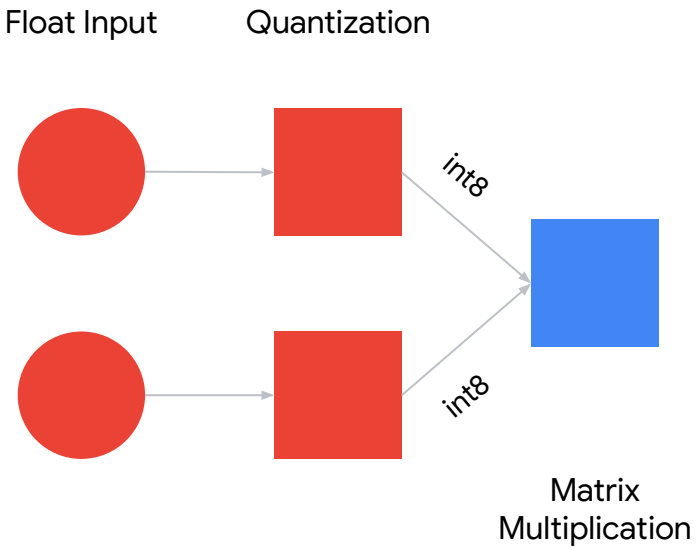
2-3x faster on fully-connected & RNN-based models (CPU quant v. CPU float)

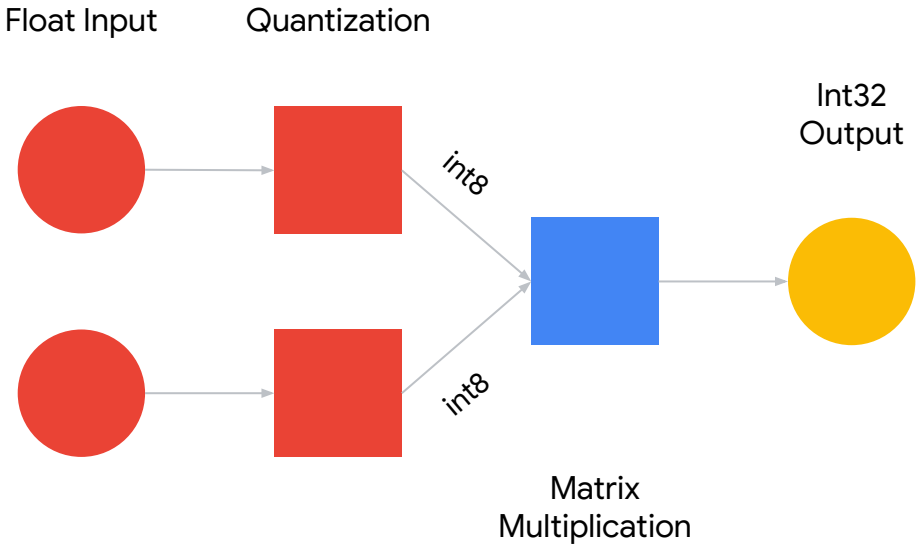
Float Input

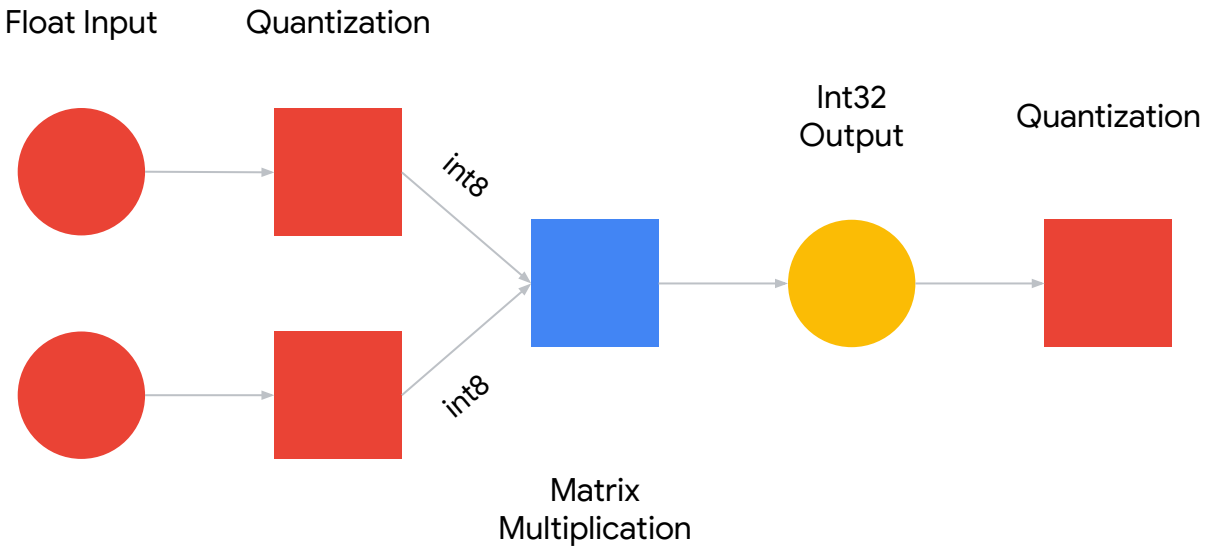
Float Input

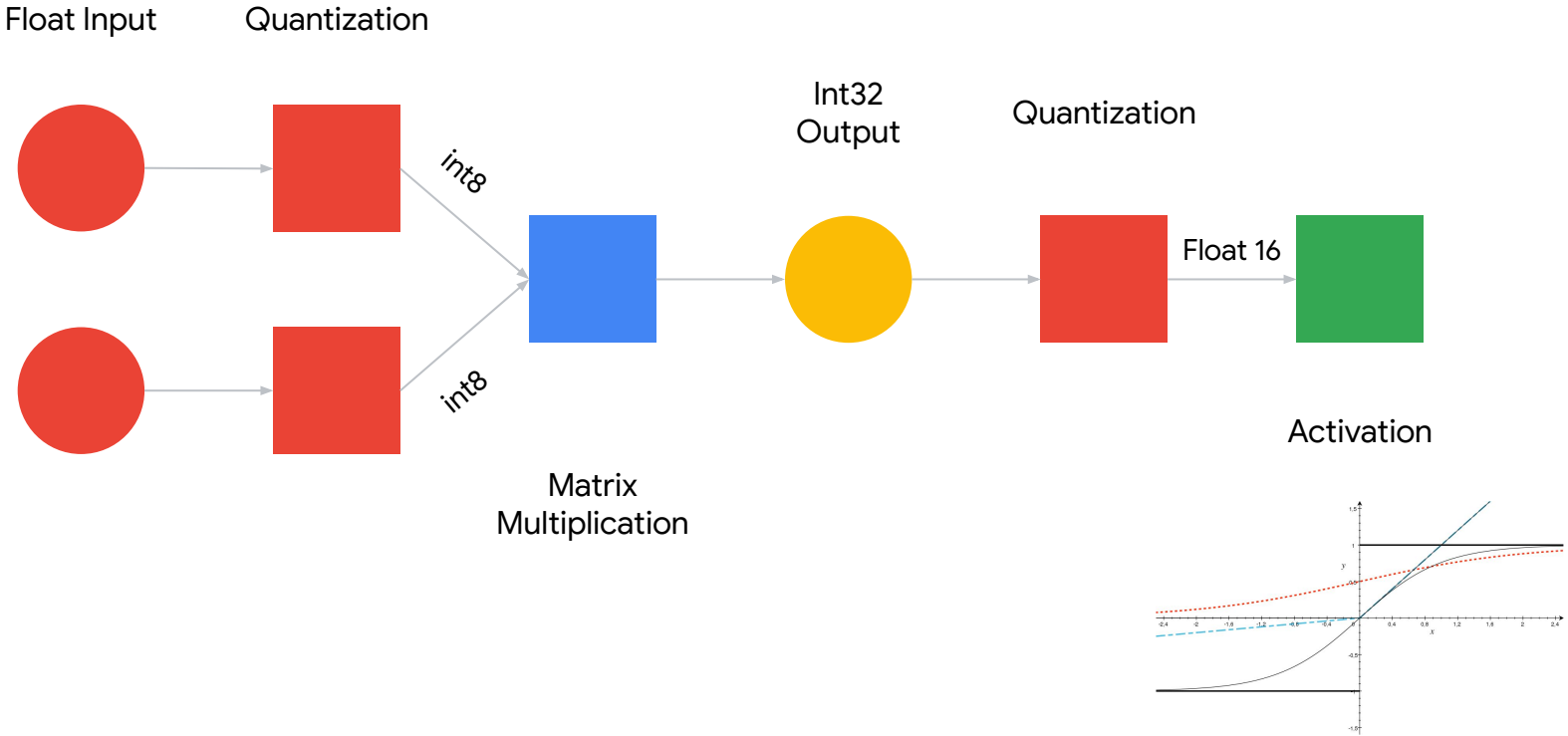
Quantization

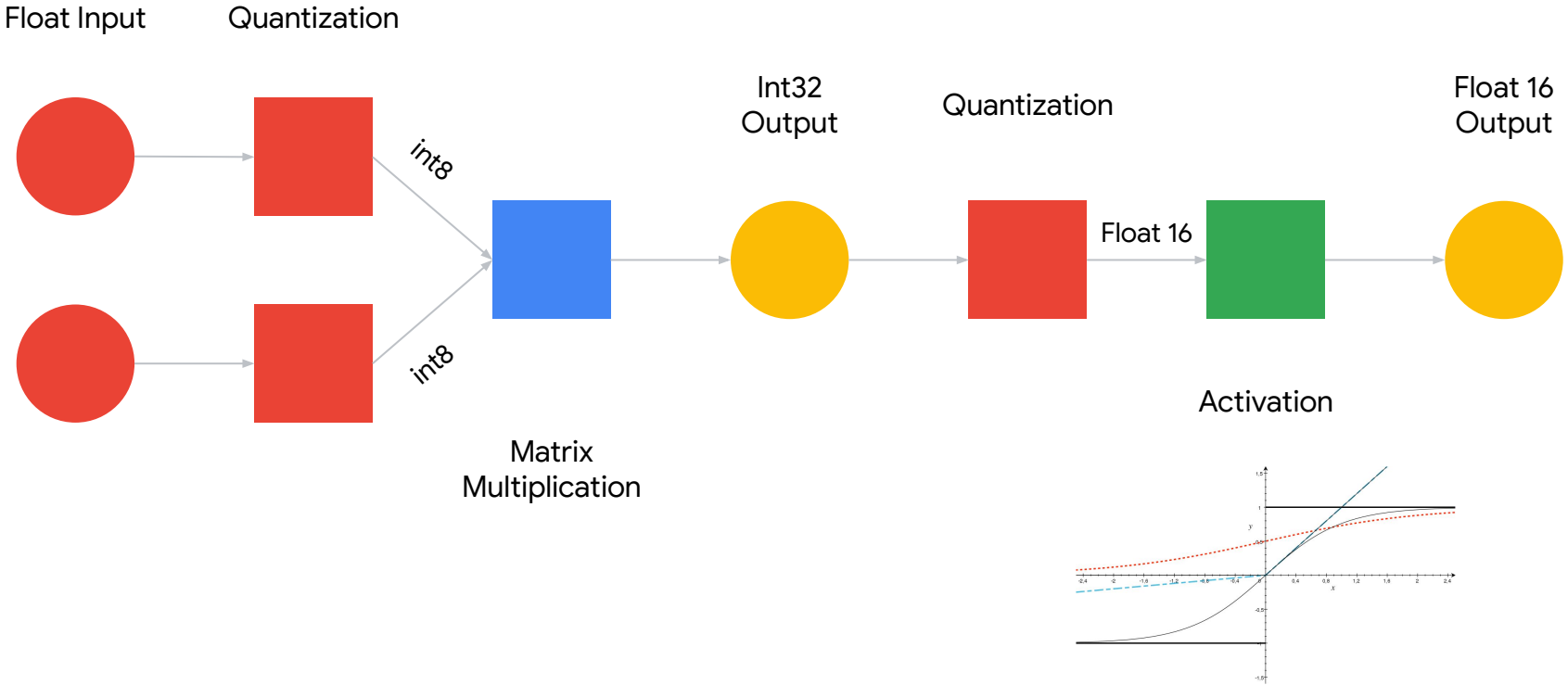












Hybrid quantization

- 8-bit integer weights
- 32-bit float biases & activations
- Integer and floating point computations

Benefits

4x reduction in model parameters (32 bit ⟶ 8 bit)

10-50% faster execution for Convolution Models (CPU hybrid v. CPU float)

2-3x faster on fully-connected & RNN-based models (CPU quant v. CPU float)

Quantization Types

Reduced Float

Hybrid Quantization

Integer Quantization

Integer quantization

- 8-bit integer weights
- 8 and 16-bit biases & activations
- Only integer computations

Reduced Float

Hybrid Quantization

Integer Quantization

Integer quantization

- 8-bit integer weights
- 8 and 16-bit biases & activations
- Only integer computations

Benefits

4x reduction in model parameters (32 bit → 8 bit)

1.5x faster execution for Convolution Models (CPU integer v. CPU float)

2-4x faster on fully-connected & RNN-based models (CPU quant v. CPU float)

Enables execution on ML accelerators (e.g., Edge-TPU)

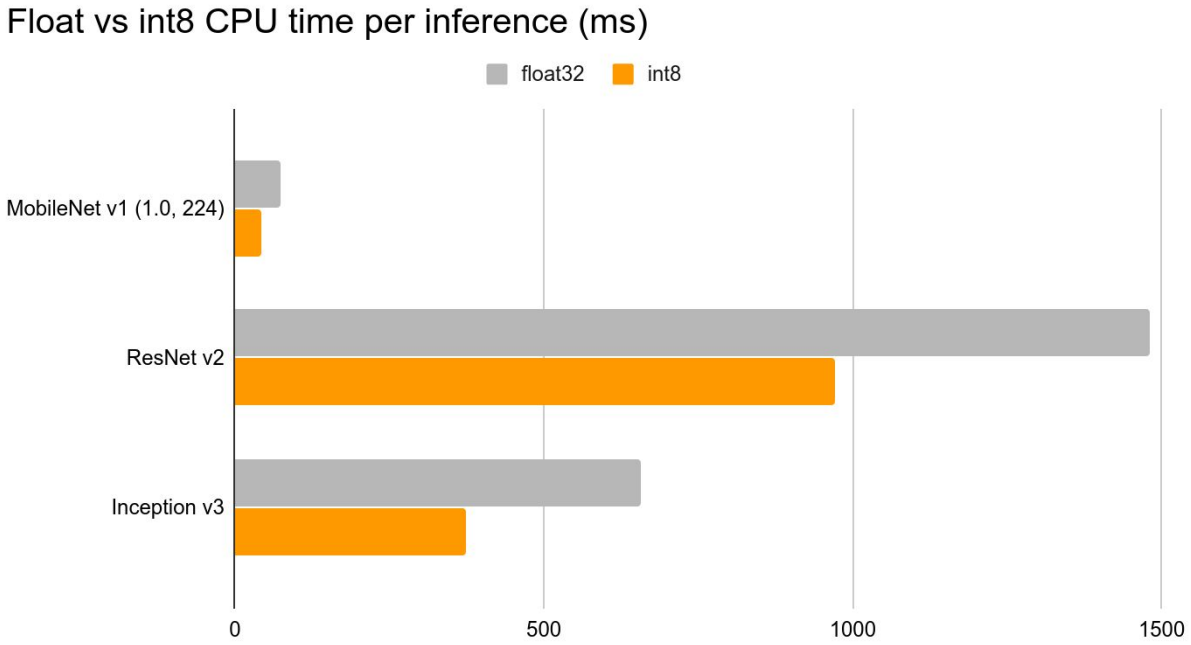
Technique	Ease of use	Accuracy	Latency	Compatibility
Reduced float (post-training)	No data required	Negligible loss	Same or faster than float32	Float16 support or fallback to float32
“Hybrid” quantization (post-training)	No data required	Small loss (≤ float16)	Faster than float	Needs float and integer support
Integer quantization (post-training0	Unlabeled data	Accuracy ≤ hybrid	Fastest	Integer only
Integer quantization (during training)	Labeled training data	Accuracy ≥ integer	Fastest	Integer only

Technique	Ease of use	Accuracy	Latency	Compatibility
Reduced float (post-training)	No data required	Negligible loss	Same or faster than float32	Float16 support or fallback to float32
"Hybrid" quantization (post-training)	No data required	Small loss (≤ float16)	Faster than float	Needs float and integer support
Integer quantization (post-training0	Unlabeled data	Accuracy ≤ hybrid	Fastest	Integer only
Integer quantization (during training)	Labeled training data	Accuracy ≥ integer	Fastest	Integer only

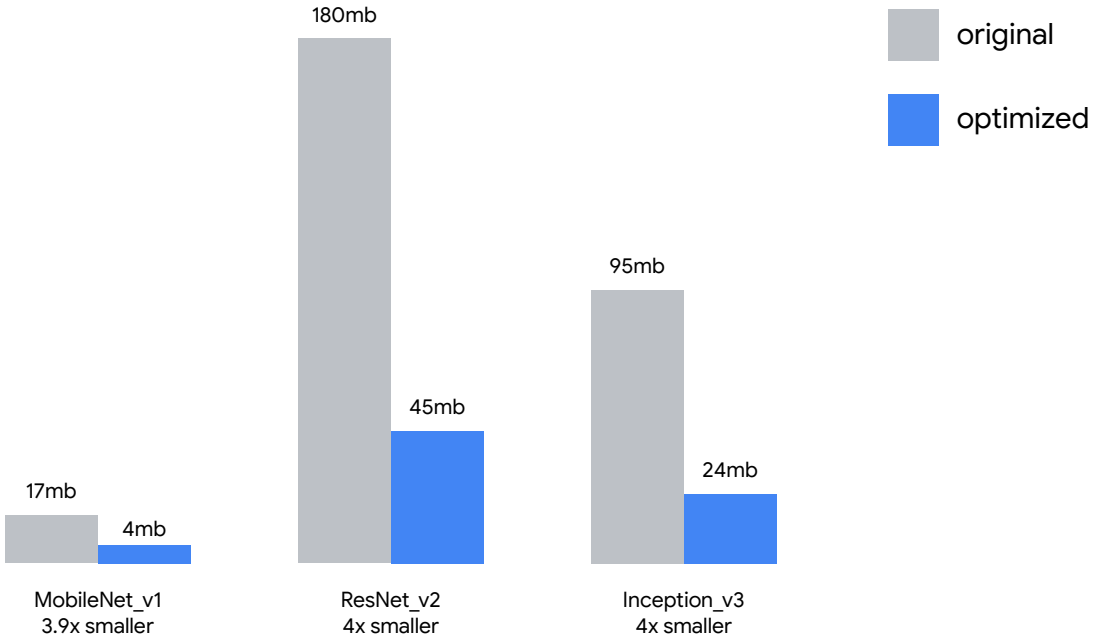
Technique	Ease of use	Accuracy	Latency	Compatibility
Reduced float (post-training)	No data required	Negligible loss	Same or faster than float32	Float16 support or fallback to float32
"Hybrid" quantization (post-training)	No data required	Small loss (≤ float16)	Faster than float	Needs float and integer support
Integer quantization (post-training0	Unlabeled data	Accuracy ≤ hybrid	Fastest	Integer only
Integer quantization (during training)	Labeled training data	Accuracy ≥ integer	Fastest	Integer only

Technique	Ease of use	Accuracy	Latency	Compatibility
Reduced float (post-training)	No data required	Negligible loss	Same or faster than float32	Float16 support or fallback to float32
"Hybrid" quantization (post-training)	No data required	Small loss (≤ float16)	Faster than float	Needs float and integer support
Integer quantization (post-training0	Unlabeled data	Accuracy ≤ hybrid	Fastest	Integer only
Integer quantization (during training)	Labeled training data	Accuracy ≥ integer	Fastest	Integer only

Runtime Performance



Model Size



Accuracy

Model	Floating point Baseline	Post-training Quantization	QAT
MobileNet v1	71.03%	69.57%	71.06%
MobileNet v2	70.77%	70.2%	70.01%

Quantization in Deep Learning