Carnegie Mellon University

14-763/18-763

Lecture 22: TinyML – Cont’d

Agenda

- EON Compiler and Tuning
- Model Compression Topics:
    1. Pruning
    2. Knowledge Distillation
    3. Quantization
- Model Compression in Edge Impulse
- Demo

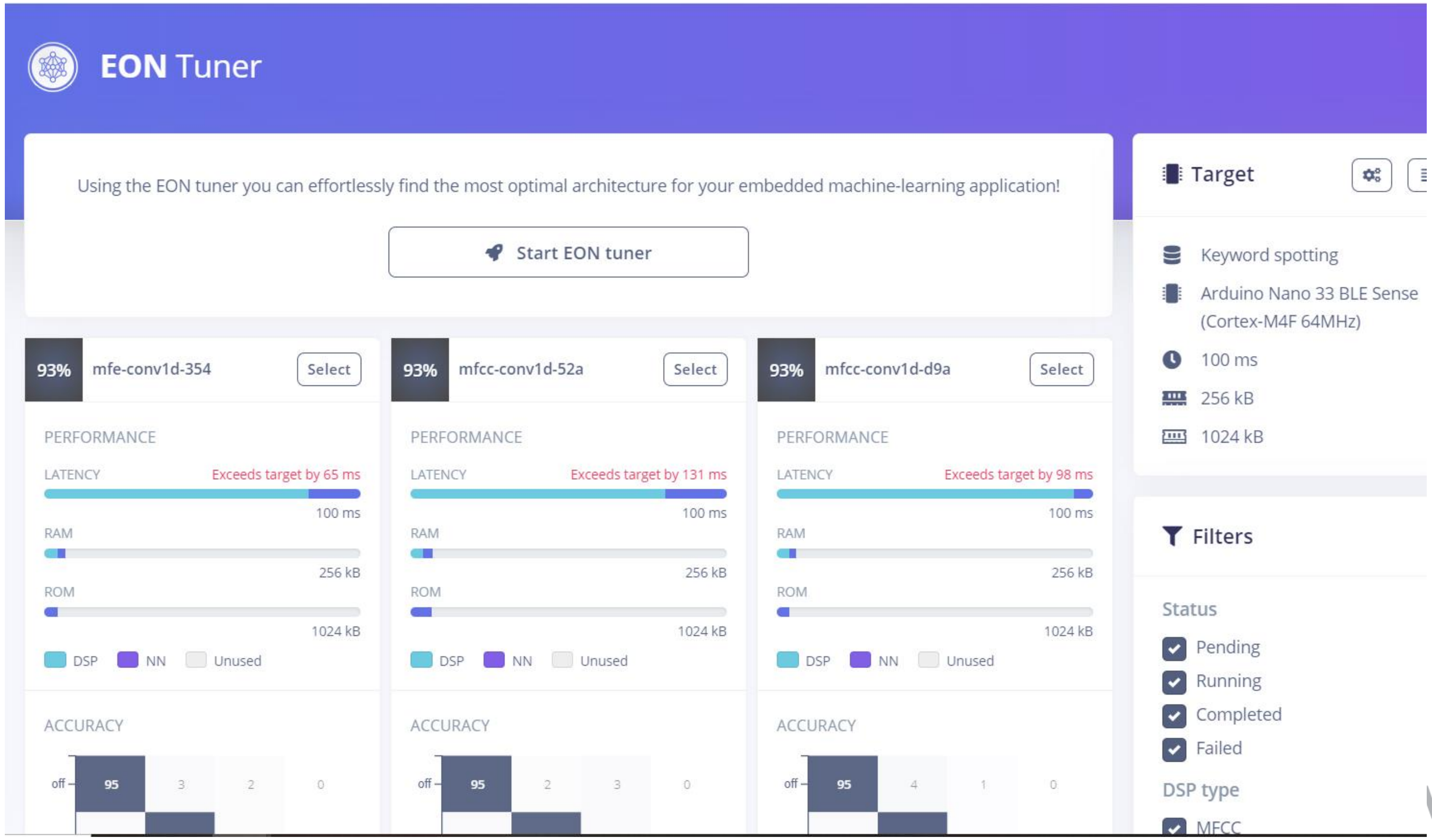
EON Compiler

- ❑ EON Compiler is a tool developed by Edge Impulse that allows users to compile and deploy machine learning models for edge devices.

- ❑ EON Compiler allows Edge Impulse to optimize machine learning models for a specific target device.

    - EON analyzes the models and the target device's capabilities, and then generating code that is optimized for that specific device.
    - This allows developers to build TinyML applications that are both accurate and efficient, while also being able to run on devices with limited resources.

EON Tuner in Edge Impulse



EON Tuner

Using the EON tuner you can effortlessly find the most optimal architecture for your embedded machine-learning application!

	mfe-conv1d-354	mfcc-conv1d-52a	mfcc-conv1d-d9a
Accuracy	93%	93%	93%
Latency	Exceeds target by 65 ms	Exceeds target by 131 ms	Exceeds target by 98 ms
Latency target	100 ms	100 ms	100 ms
RAM	256 kB	256 kB	256 kB
ROM	1024 kB	1024 kB	1024 kB
Accuracy (off row)	95, 3, 2, 0	95, 2, 3, 0	95, 4, 1, 0

Target

- Keyword spotting
- Arduino Nano 33 BLE Sense (Cortex-M4F 64MHz)
- 100 ms
- 256 kB
- 1024 kB

Filters

Status
- ✓ Pending
- ✓ Running
- ✓ Completed
- ✓ Failed

DSP type
- ✓ MFCC

Mellon University

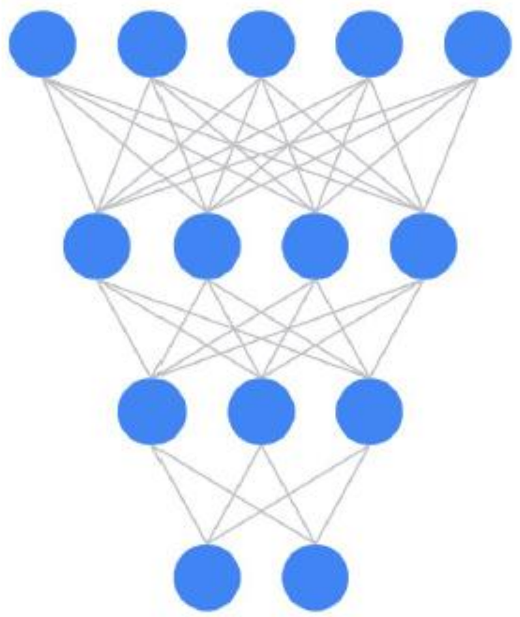
Pruning

- Pruning is used to compress models in TinyML.
- Dense Neural Networks (DNNs) take more time to train. Fewer network synapses will make the network faster.
- Pruning aims to delete weights and connections from a neural network model that are deemed to be not required.
- It is predicated on the concept that a neural network model typically has many redundant or unnecessary connections and weights, which do not significantly contribute to the performance of the model.

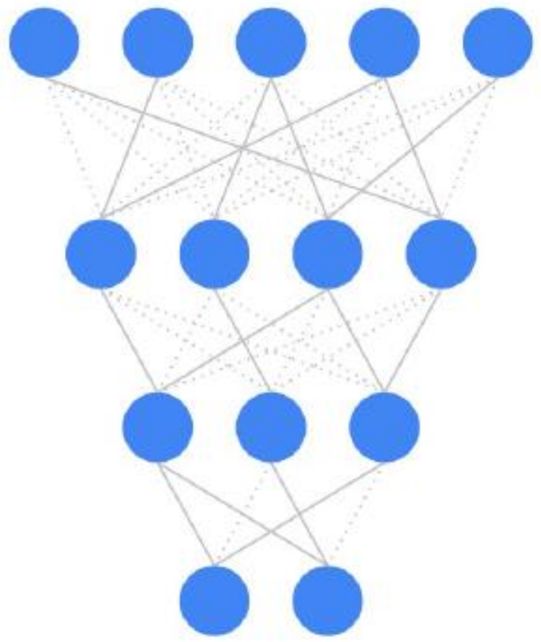
Pruning Methods

- Zeroing out small weights
- Lasso regularization (L1 regularization)
- Tensorflow Model optimization toolkit

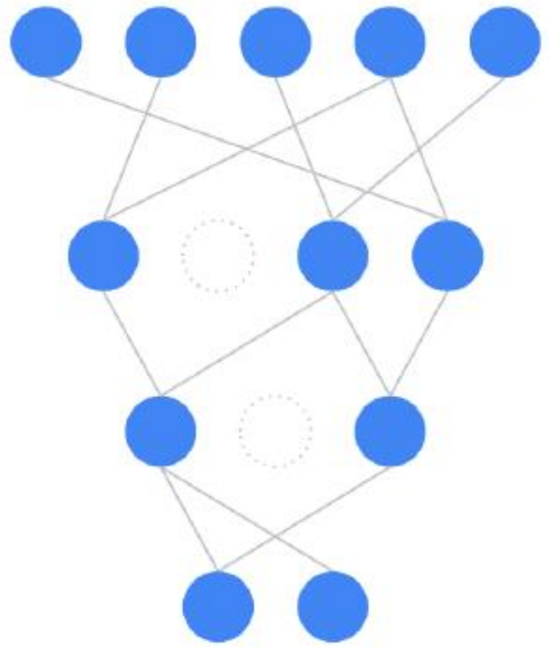
Pruning Example



DNN



NN when eliminating synapses with small weights

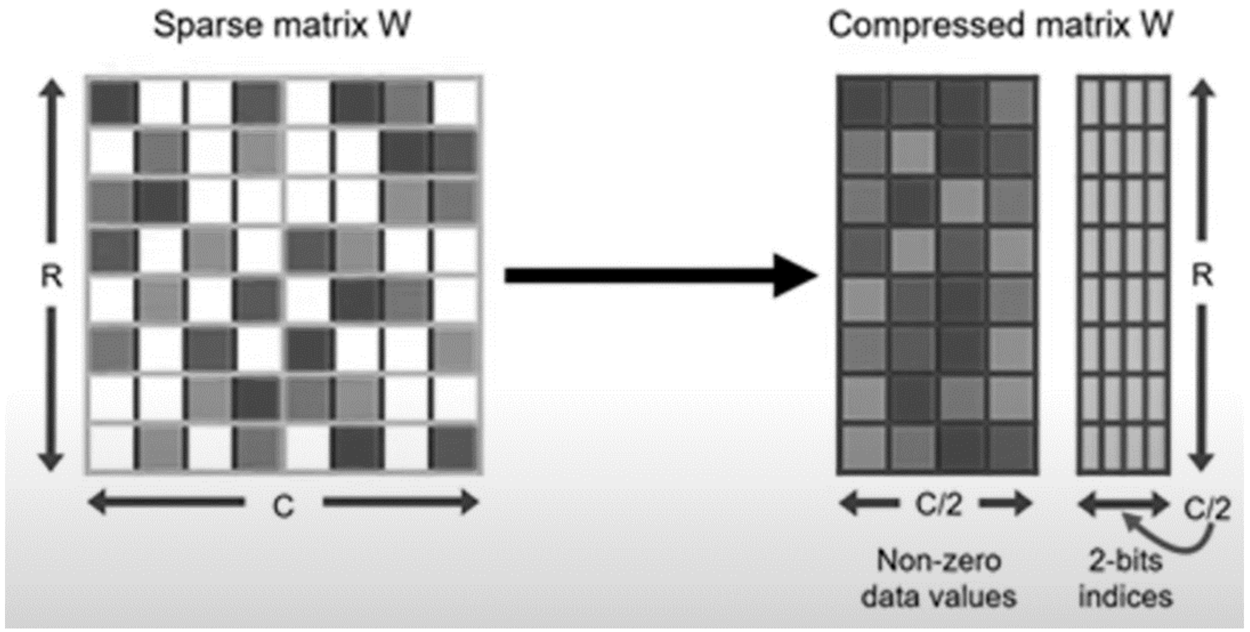


NN with eliminating neurons with no synapses

Pruning Types

- **Unstructured Pruning:** like what we saw in the previous example. This approach is hard to model and represent for hardware devices.

- **Structured Pruning:** Specifies a pattern in applying pruning. E.g., 2:4 Structured Sparsity



Knowledge Distillation

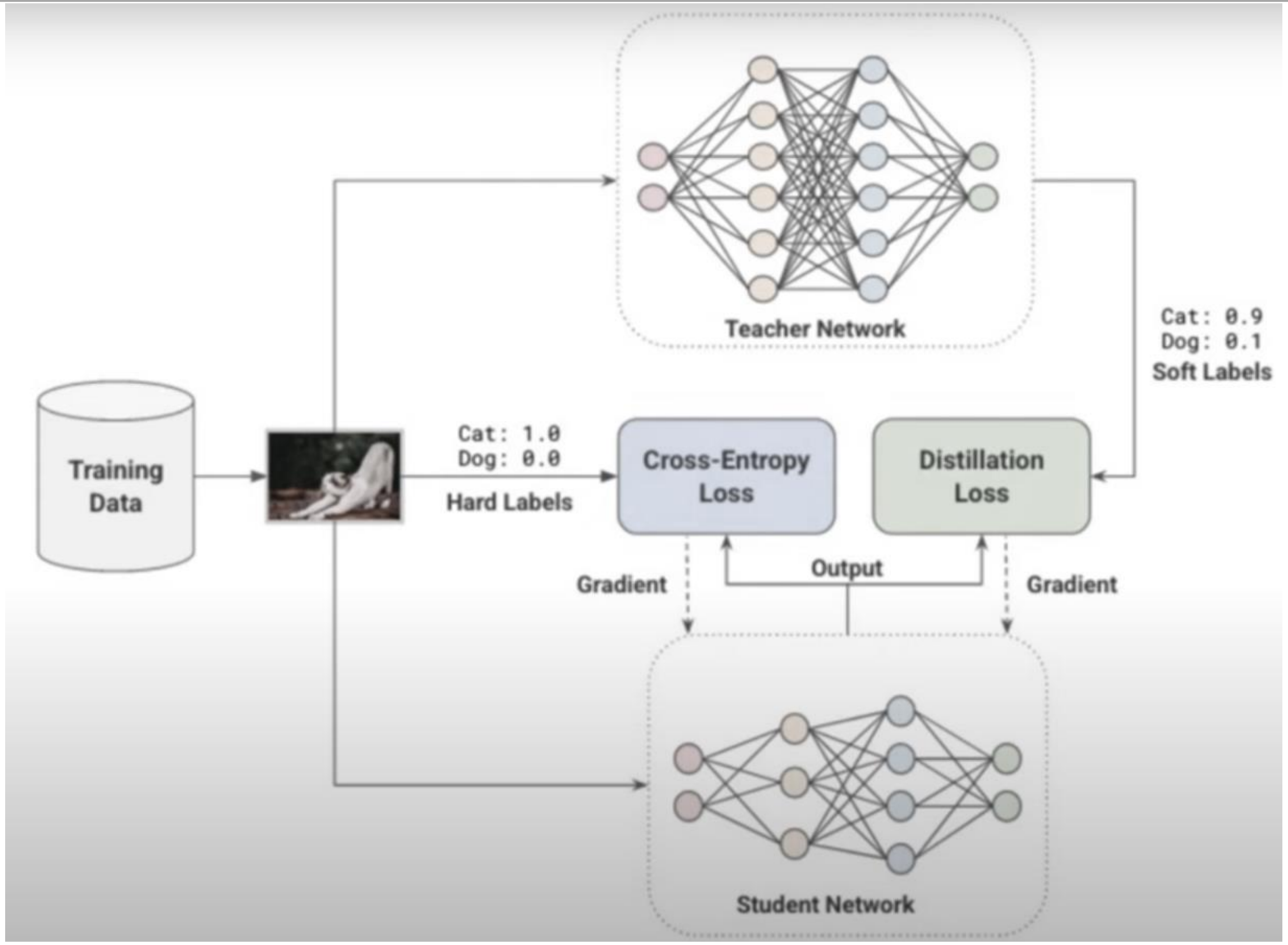
- In Knowledge Distillation, you teach a more compact model sometimes known as the "student model" to behave in the same way as a more extensive model, that has already been pre-trained (called the "teacher" model).
- The purpose of knowledge distillation is to transfer the knowledge from the teacher model to the student model, so that the student model can perform similarly to the teacher model, while having a smaller model size and faster inference speed.

Knowledge Distillation Steps

1. Begin with a teacher model that has already been pre-trained, and a dataset.
2. Train the student model on the dataset to predict the output of the teacher network

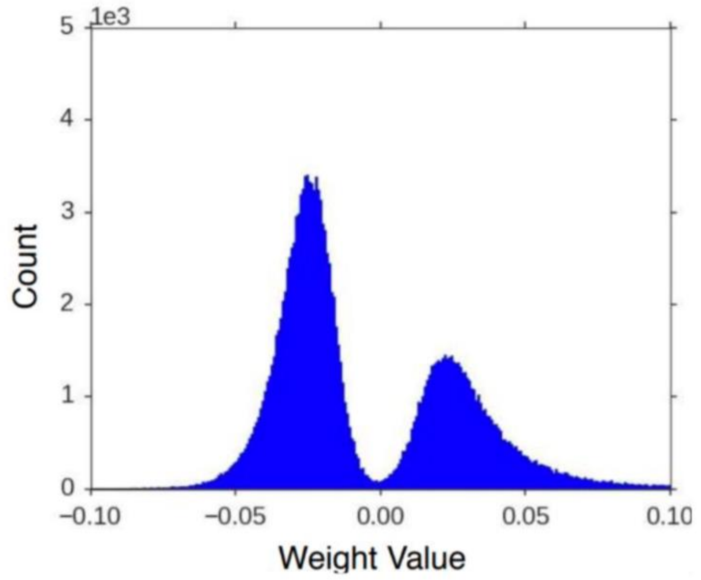
- In your opinion, why would we train the student model to predict the output of the teacher network instead of training it to predict the output from the labels?

Knowledge Distillation Example



Quantization

- Most DNN models have weights concentrated in a small range.
- The concentration of weights makes quantization into 8-bit integer feasible for inferencing.
- Quantization is the process of decreasing the precision of a neural network model's weights and activations, to decrease the model's memory footprint and computational complexity.



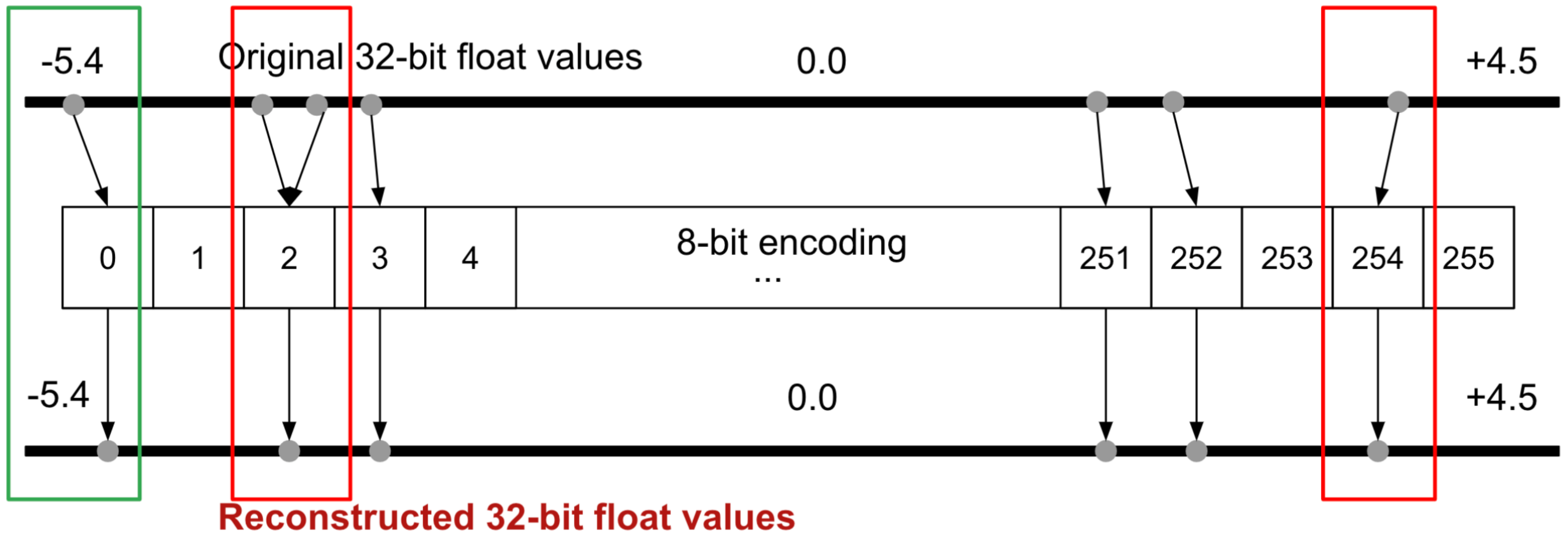
Quantization Formula

- Can you find the quantization formula to convert a 32-bit number $float$ to an 8-bit signed number $int8$ given a floating -point range $R = max f - \min f$?
- Suppose R is centered around zero.
- Hint: First find the scale $s = \frac{R}{int8\ range}$

Carnegie Mellon University

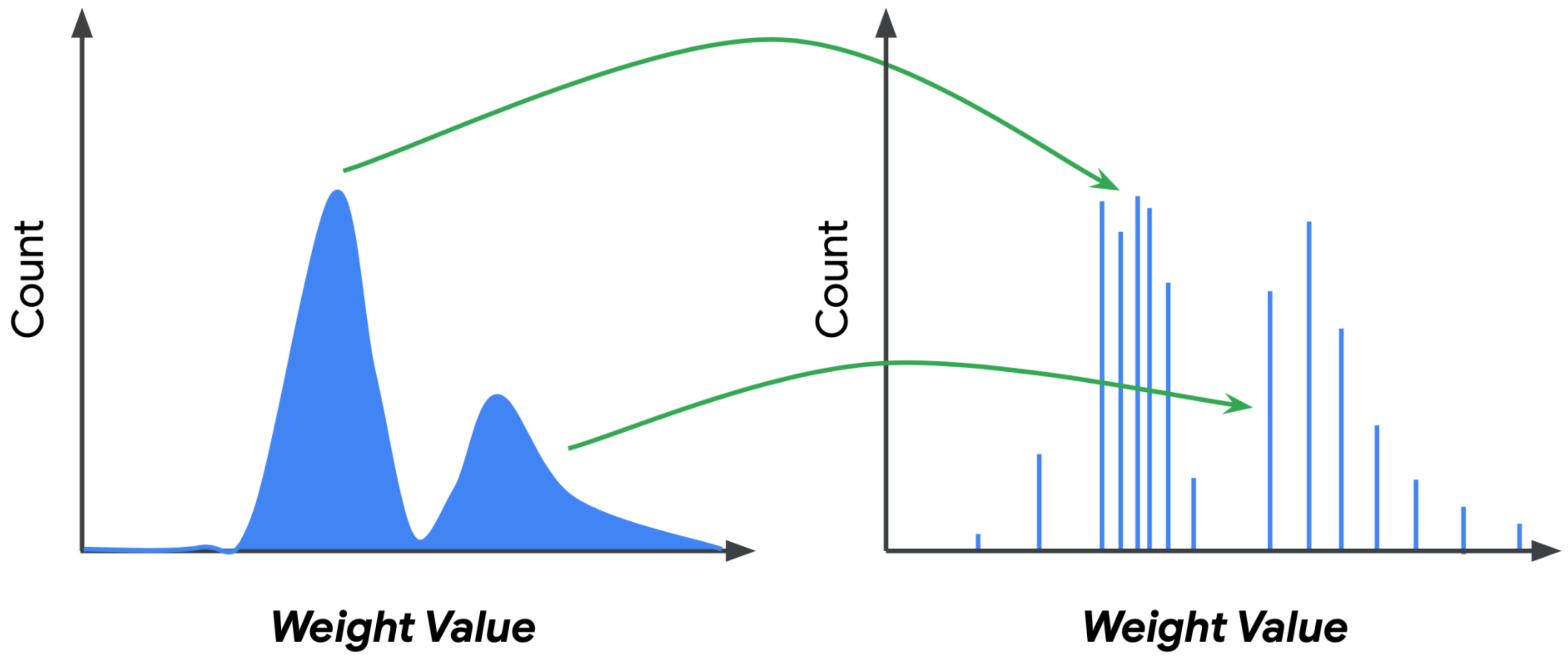
Quantization Binning

- Reduction of precision could result from binning floats into the same integer


-5.4
Original 32-bit float values
0.0
+4.5
0
1
2
3
4
8-bit encoding
...
251
252
253
254
255
-5.4
0.0
+4.5
Reconstructed 32-bit float values

Carnegie Mellon University

Quantization – Loss of Binning Due to Binning



Quantization Types

- **Weight quantization**: It involves reducing the precision of the weights of the model, typically from 32-bit floating point numbers to 8-bit integers or even lower-precision fixed-point numbers.
- **Activation quantization**: It involves reducing the precision of the activations of the model, typically from 32-bit floating point numbers to 8-bit integers or lower-precision fixed-point numbers.
- In your opinion, which type leads to better performance?

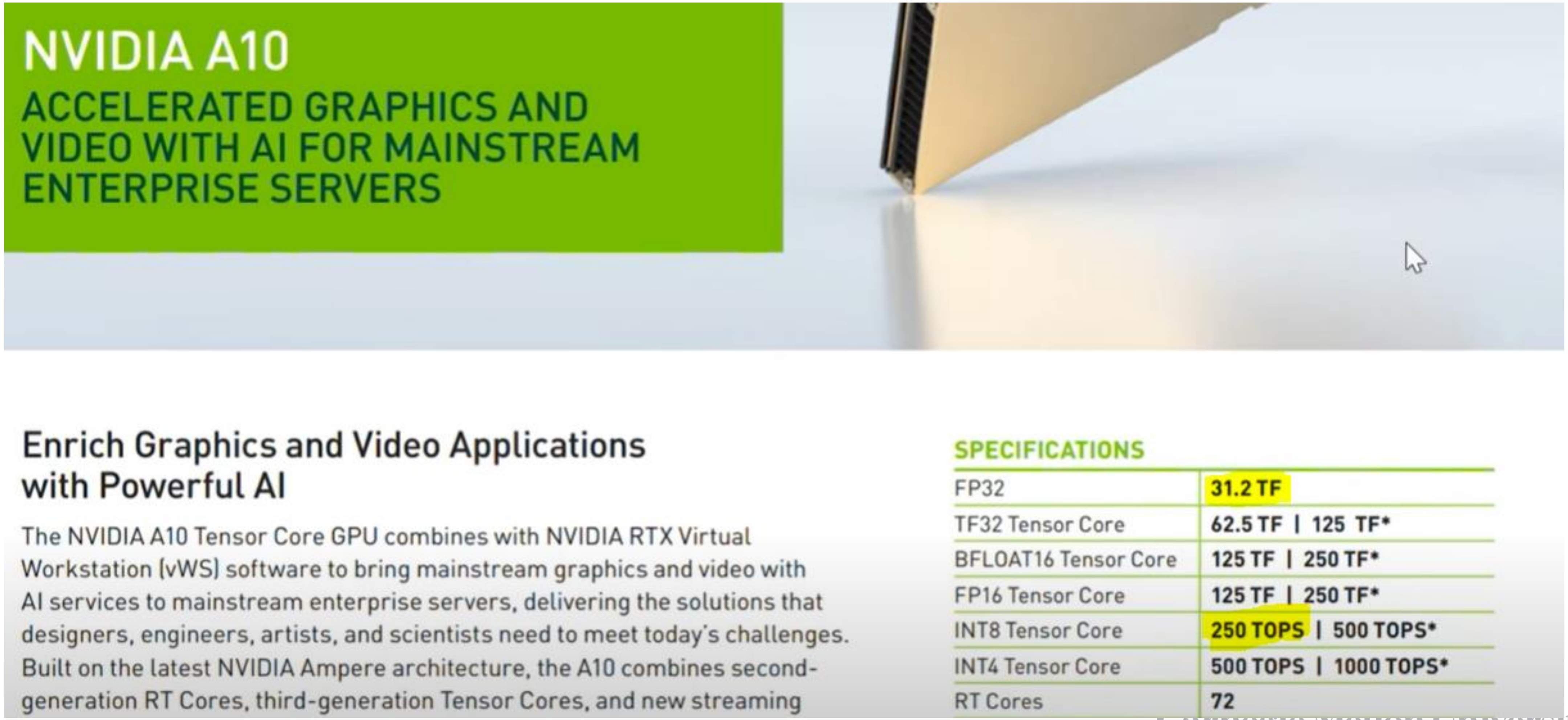
Quantization Implementation Strategies

- **Dynamic fixed-point quantization** involves converting the weights and activations to fixed-point numbers at runtime.
- **Static fixed-point quantization** involves converting the weights and activations to fixed-point numbers ahead of time and storing them in the model.
- **Quantization-aware training** involves training the model with quantization in mind, using techniques such as fake quantization to mimic the effects of quantization during training

Loss of Accuracy Due to Quantization

	Floating-point Baseline	Post-training Quantization (PTQ)	Quantization-Aware Training (QAT)
MobileNet v1 1.0 224	71.03%	69.57%	71.06%
MobileNet v2 1.0 224	70.77%	70.20%	70.01%
Resnet v1 50	76.30%	75.95%	76.10%

Quantization Performance Example



NVIDIA A10

ACCELERATED GRAPHICS AND VIDEO WITH AI FOR MAINSTREAM ENTERPRISE SERVERS

Enrich Graphics and Video Applications with Powerful AI

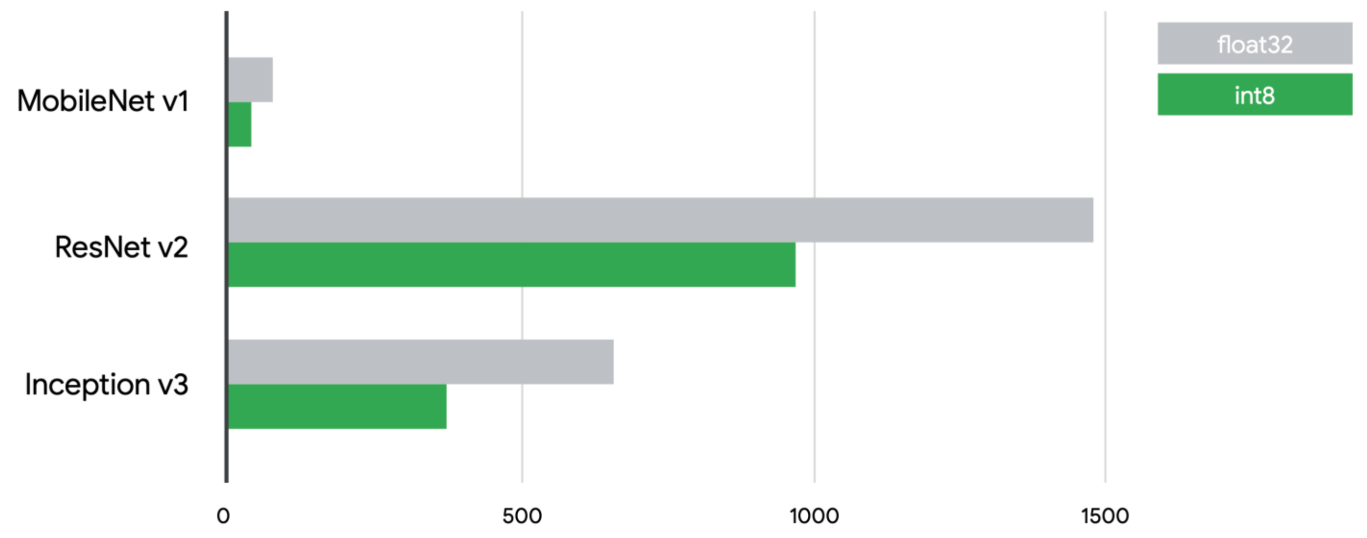
The NVIDIA A10 Tensor Core GPU combines with NVIDIA RTX Virtual Workstation (vWS) software to bring mainstream graphics and video with AI services to mainstream enterprise servers, delivering the solutions that designers, engineers, artists, and scientists need to meet today's challenges. Built on the latest NVIDIA Ampere architecture, the A10 combines second-generation RT Cores, third-generation Tensor Cores, and new streaming

SPECIFICATIONS

FP32	31.2 TF
TF32 Tensor Core	62.5 TF | 125 TF*
BFLOAT16 Tensor Core	125 TF | 250 TF*
FP16 Tensor Core	125 TF | 250 TF*
INT8 Tensor Core	250 TOPS | 500 TOPS*
INT4 Tensor Core	500 TOPS | 1000 TOPS*
RT Cores	72

Speed of Quantized Models

Int8 v. Float (CPU time per inference)



Quantized models are up to 2–4x faster on CPU and 4x smaller.

Benefits of Quantization

- Smaller model in term of memory size
- Faster inference
- More portable model

Model Compression in Edge Impulse

- **Automated model pruning:** Edge Impulse can automatically identify and remove unnecessary weights and connections from a model, resulting in a smaller and more efficient model.
- **Quantization:** Edge Impulse can convert a model's floating-point weights and activations to fixed-point values, which requires fewer bits to represent and can result in a more efficient model.

Model Compression in Edge Impulse – Cont’d

- **Weight sharing:** Edge Impulse can identify shared weights within a model and consolidate them into a single weight, reducing the number of parameters in the model.
- **Low-precision inference**: Edge Impulse can reduce the precision of the weights and activations in a model during inference, while maintaining accuracy.

Demo

Carnegie Mellon University