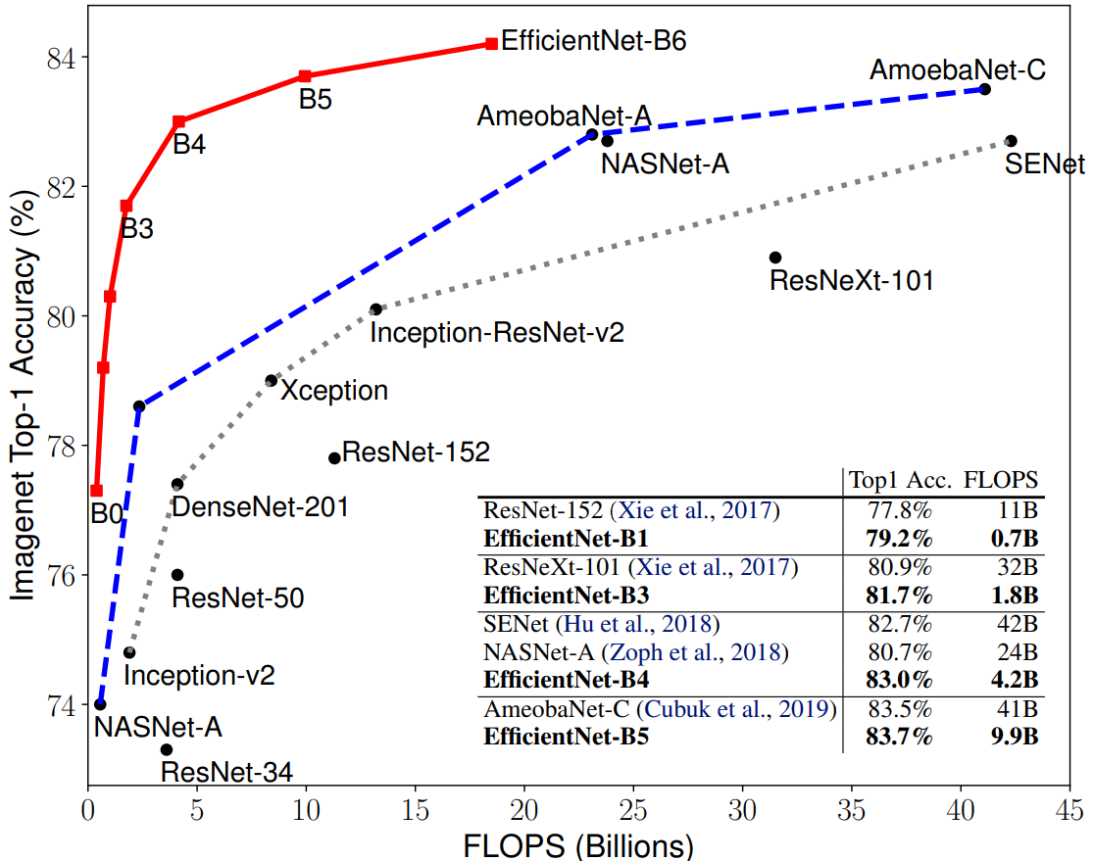
Recent Advances in Post-training Quantization

Alexander Kozlov
Intel Corporation
September 2020

Introduction

Deep Learning (DL) models optimization trends

- **Optimization by design**
  - Lightweight building blocks
  - Neural Architecture Search (NAS)
- **Optimization of existing SOTA models**
  - Optimization with fine-tuning (e.g. QAT)
  - Post-training methods (e.g. post-training quantization)



	Top1 Acc.	FLOPS
ResNet-152 (Xie et al., 2017)	77.8%	11B
EfficientNet-B1	**79.2%**	**0.7B**
ResNeXt-101 (Xie et al., 2017)	80.9%	32B
EfficientNet-B3	**81.7%**	**1.8B**
SENet (Hu et al., 2018)	82.7%	42B
NASNet-A (Zoph et al., 2018)	80.7%	24B
EfficientNet-B4	**83.0%**	**4.2B**
AmeobaNet-C (Cubuk et al., 2019)	83.5%	41B
EfficientNet-B5	**83.7%**	**9.9B**

https://arxiv.org/pdf/1905.11946v3.pdf

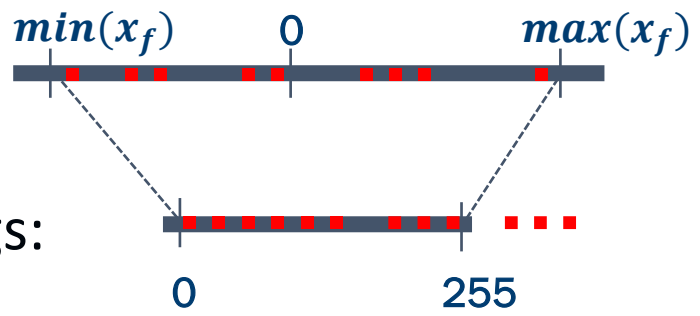
Quantization in DL

- Simplest and fastest way to speed up the model inference
- Can be used both with fine-tuning and post-training
- The idea is to approximate FP32 operations with integer analogs:

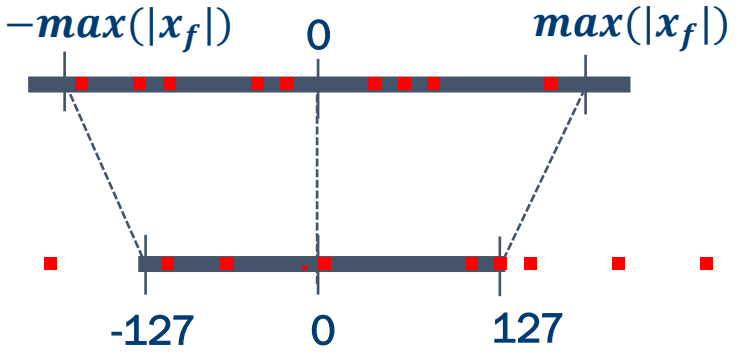
$$Y = W_{f32}X_{f32} \approx s_w W_{i8} s_x X_{i8} = s_w s_x (W_{i8} X_{i8})$$

- INT8 quantization is a mainstream method
- INT4:
  - Accurate model is accessible via QAT
  - Currently passing frontier in post-training

Asymmetric INT8 quantization



Symmetric INT8 quantization



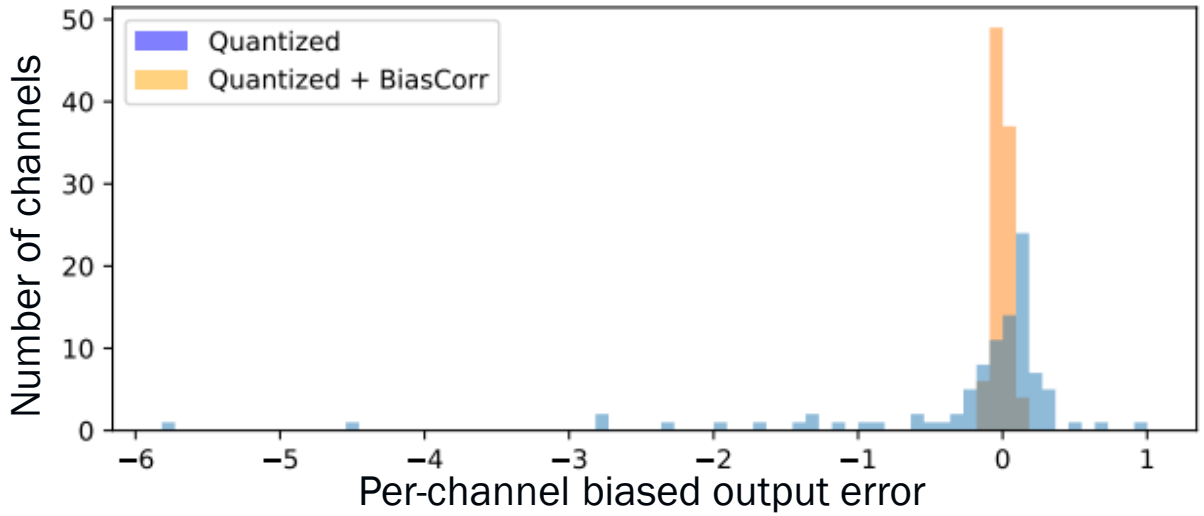
Post-training quantization implementation

INT8 quantization implemented in almost any framework:

- Training:
  - TensorFlow Quantization-Aware Training
  - PyTorch
  - MXNet
- Inference:
  - Microsoft ONNX Runtime
  - Nvidia TensorRT
  - Intel OpenVINO

Recent advances in post-training quantization

- Channel alignment: https://arxiv.org/abs/1902.01917
- Bias correction and variance correction: https://arxiv.org/pdf/1810.05723
- Outliers channel splitting: https://arxiv.org/abs/1901.09504
- Global optimization of quantization parameters: https://arxiv.org/pdf/1911.07190
- “Data-free” quantization: https://arxiv.org/abs/1906.04721



What we tried for INT8

- Channel alignment: for Conv->Conv pattern
- Bias correction and variance correction
  - Honest and fast bias correction
  - Variance correction: does not help in general for INT8
- Outliers channel splitting: tiny gain in case of INT8
- Global optimization (TPE, genetic algorithms): make sense for some topologies
- “Data-free” quantization: performs worse than statistics based methods

Proposed quantization flow

Default method for INT8 post-training quantization*:

- Symmetric per-channel weights
- Per-channel activations whenever it is possible, can be symmetric or asymmetric
- Activation Channel alignment (conv->conv pattern)
- Statistics based initialization:
  - 300 samples
  - Scales for weights are initialized by $\max_i(abs(w_i))$
  - For activations – $\underset{j}{\text{mean}} \max_i(abs(a_i))$, where j – is sample number
- Bias correction

*Validated on 100+ DL models

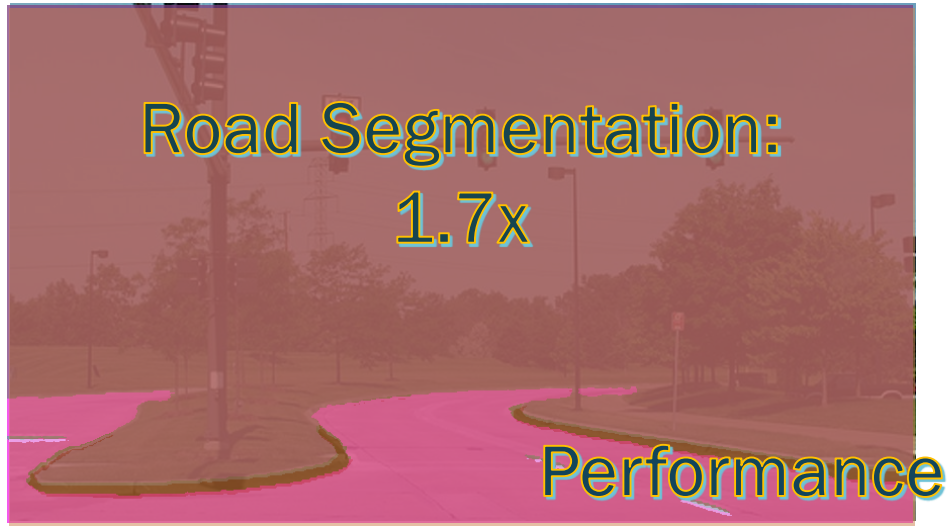
Accuracy-aware quantization

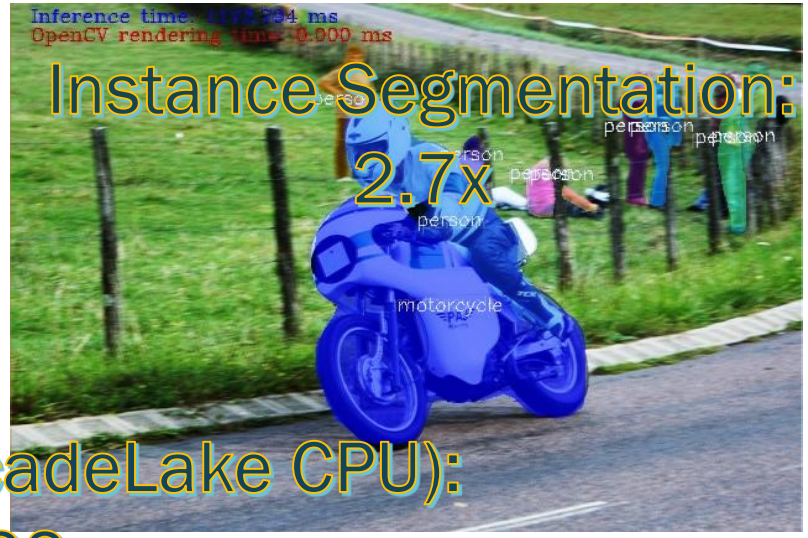
- Revert layers back to FP32 after default quantization if low accuracy
- Layers ranking criterion:
  - KL-divergence
  - SQNR
  - Other metrics
- The best way is to get per-layer contribution to the target metric drop
- Problem: how to get maximal subgraph when reverting to FP32 to avoid performance drop due to precision conversion (INT8->FP32 and back)?

Hardware specific

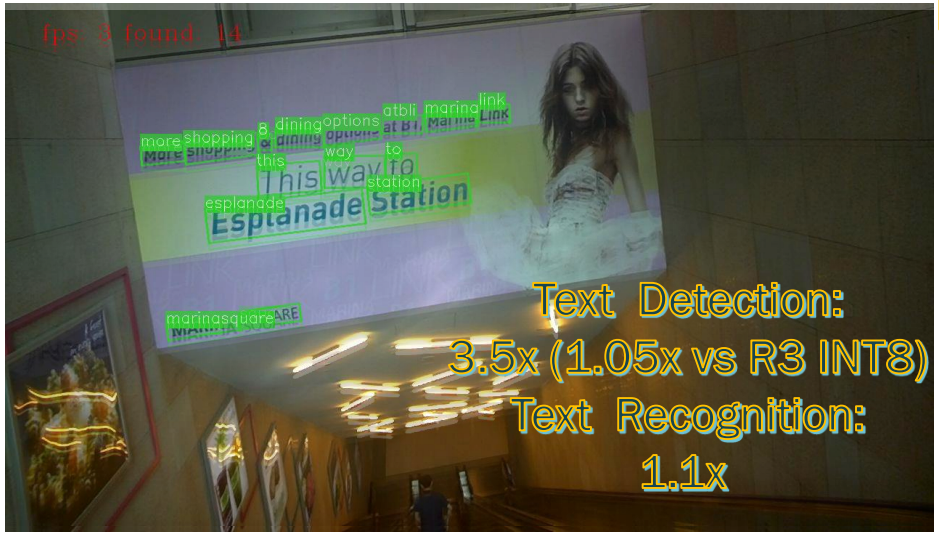
- Mixed precision
  - FP32/INT8, FP16/INT8, etc.
  - Efficiency (general-purpose HW can use it more efficiently)
- Quantization schemes
  - Per-tensor/per-channel parameters
  - Symmetric/asymmetric quantization
- Instruction set specifics (e.g. saturation)

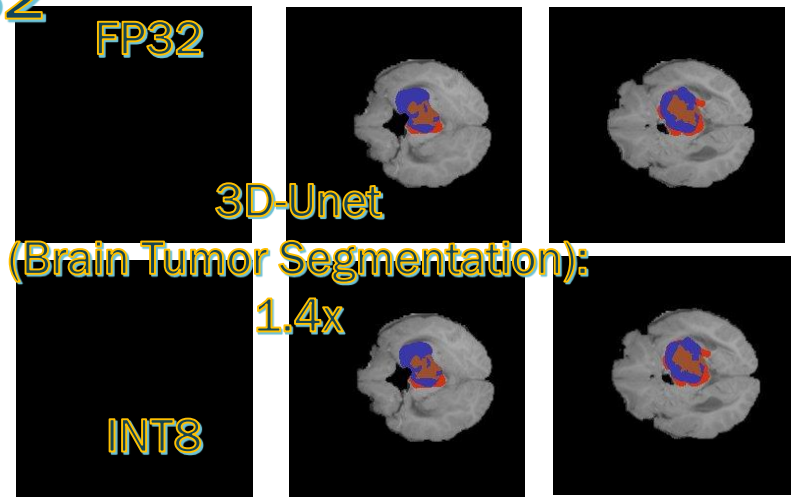
Results





Performance gain (CascadeLake CPU):
INT8 vs FP32





Results

Model name	Framework	Dataset	FP32 accuracy	INT8 accuracy	Accuracy Drop	Speed up
MobileNet-v2	Caffe	ImageNet	0.7181	0.71518	0.00292	3.46x
MobileNet v1	TF	ImageNet	0.71848	0.71284	0.00564	3.49x
SE-ResNet-50	Caffe	ImageNet	0.77598	0.77398	0.002	4.76x
SqueezeNet v1.1	Caffe	ImageNet	0.58384	0.57668	0.00716	3.16x
SSD-MobileNet	Caffe	VOC2012	0.231664	0.229522	-0.003528	3.42x
SSDLite-MobileNet-v2	TF	COCO	0.241497	0.240297	0.0012	2.5x
SSD-ResNet-34 1200	MXNet	COCO	0.198346	0.197137	0.001209	3.34x
Faster R-CNN ResNet-50	TF	COCO	0.298975	0.296356	0.002619	3.24x
Facenet Inception ResNet v1	TF	CASIA-WebFace	0.988695	0.988547	0.000148	3.16x
Deeplab v3	TF	VOC 2012	0.630643	0.625723	0.00492	2.52x
Brain-tumor-segmentation	MXNet	BraTS	0.923999	0.922889	0.00111	1.38x

*All models were obtained with INT8 post-training quantization algorithm within this toolkit

**Accuracy and performance numbers were measured on Intel® Xeon® Gold 8270 Processor

Key learnings

- Use more granular quantization scales as much as possible; e.g., per-channel (output) scales for conv filters, per-channel activations in Depth-wise convolution, etc.
- Make the model output unbiased (bias correction)
- Handle zero filters
- Mixed mode is preferable for CPU (symmetric weights/asymmetric activations)
- INT4 requires asymmetric quantization to get accurate models (even with fine-tuning)
- Mixed precision with automatic bit-width selection is the future

Resources

OpenVINO

OpenVINO Toolkit

https://docs.openvinotoolkit.org/

Post-training Optimization Toolkit

https://docs.openvinotoolkit.org/latest/_README.html

Blog post on low-precision

https://www.intel.ai/open-vino-low-precision-pipeline

Quantization-aware training

NNCF paper with code for PyTorch

https://arxiv.org/abs/2002.08679v2

2020 Embedded Vision Summit

“Advanced Workshop on Intel® Vision Technology and OpenVINO™ Toolkit”