intel Software

OpenVINO™

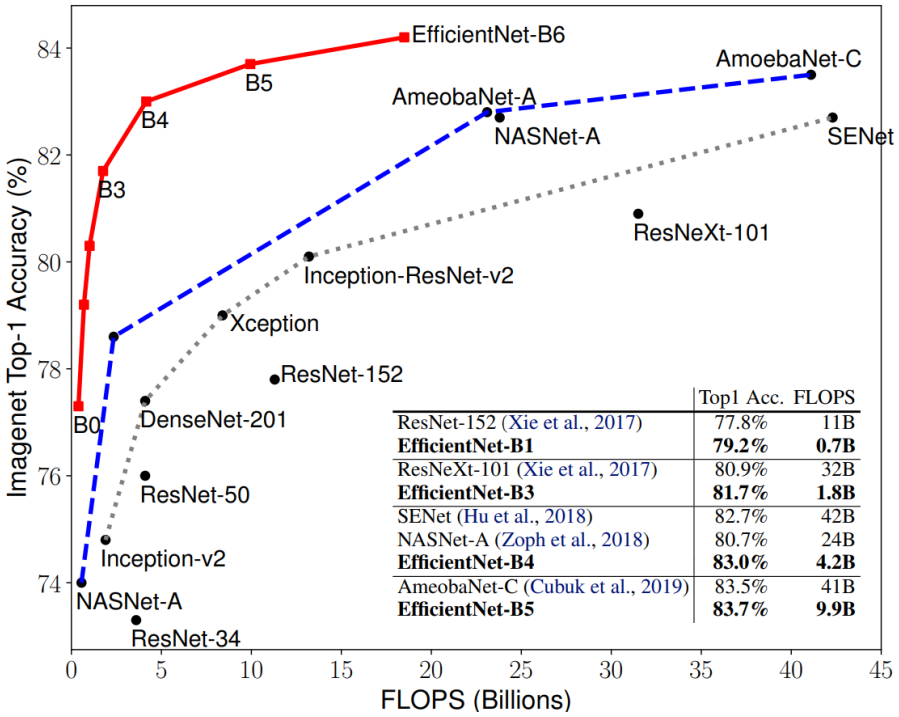
IMPROVED QUANTIZATION TECHNIQUES TO BOOST PERFORMANCE OF INFERENCE WORKLOADS

Alexander Kozlov

Low-Precision Architect and Deep Learning R&D Engineer, Intel Corporation

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**

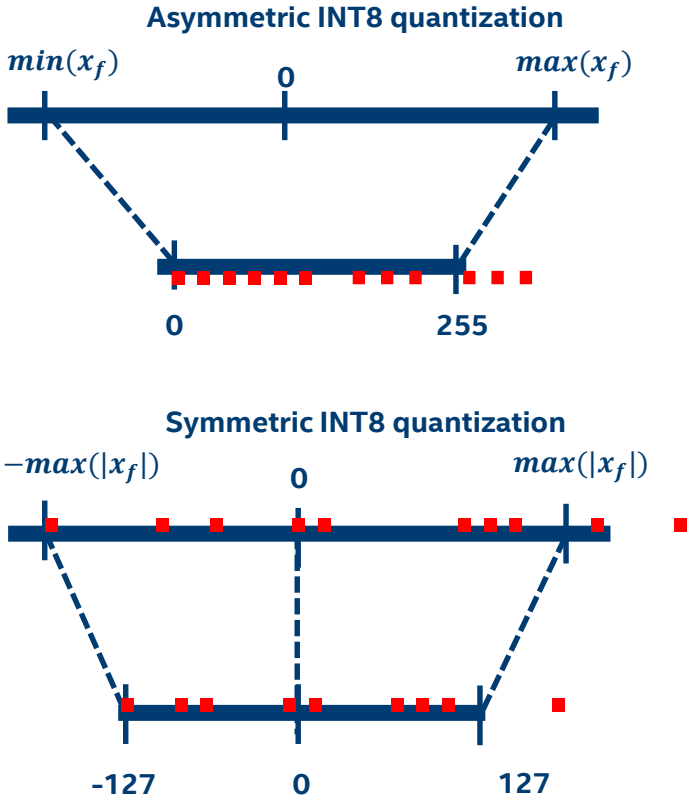
https://arxiv.org/pdf/1905.11946v3.pdf

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:

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

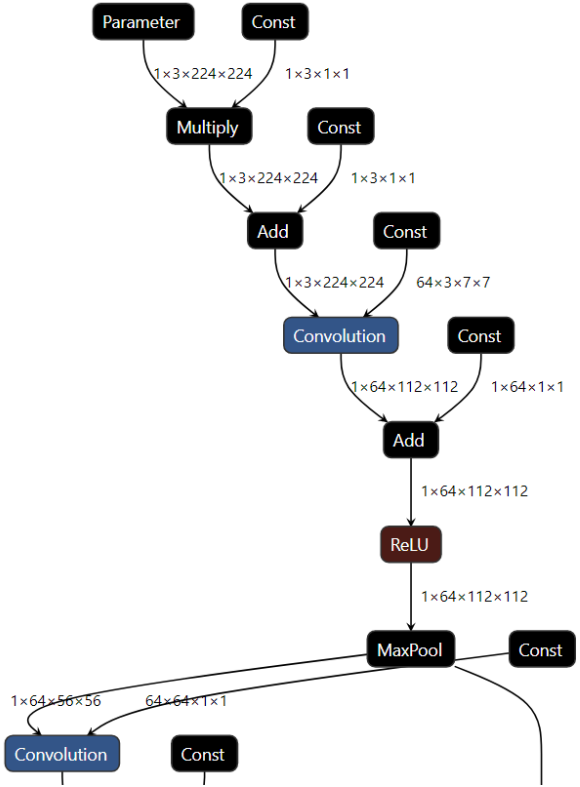
- INT8 quantization is a mainstream method
- INT4:
  - Accurate model is accessible via QAT
  - Currently storming in post-training



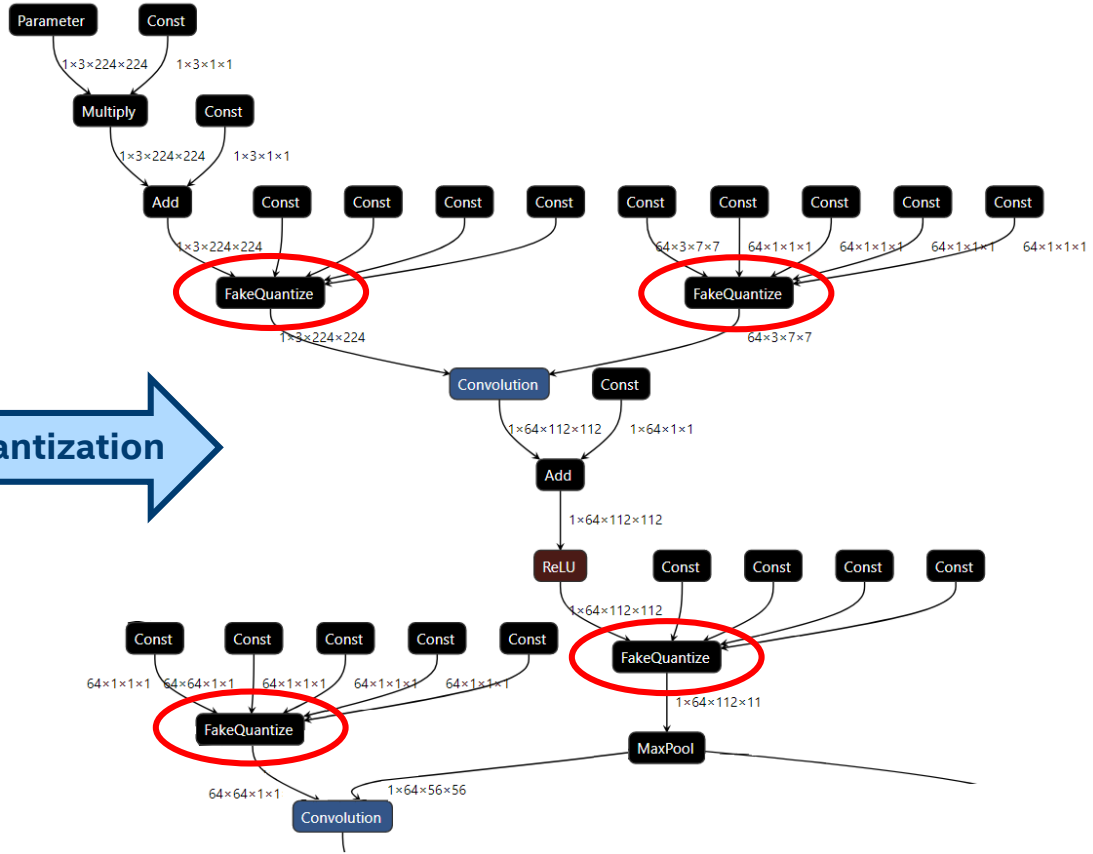
Intel® Distribution of OpenVINO™ toolkit and Quantization

- Unified representation for quantized models via **FakeQuantization** primitive
- Support of quantized models from multiple sources:
  - Quantization-aware training:
    - TensorFlow* QAT
    - PyTorch* NNCF
  - Post-training quantization:
    - Post-training Optimization Tool (set of tools part of the Intel® Distribution of OpenVINO™ toolkit, as introduced on release 2020.1)

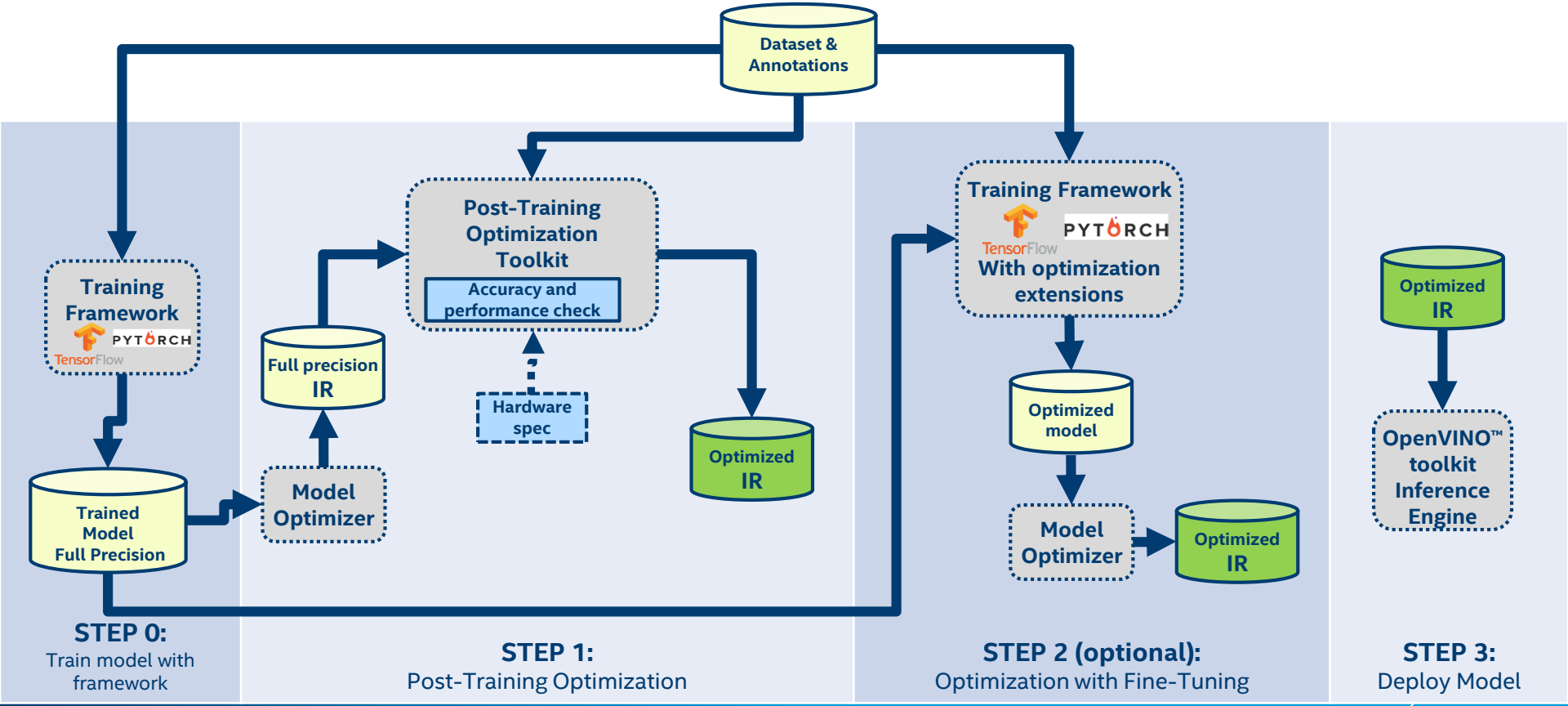
IR transformation



Quantization



OpenVINO™ toolkit model optimization flow

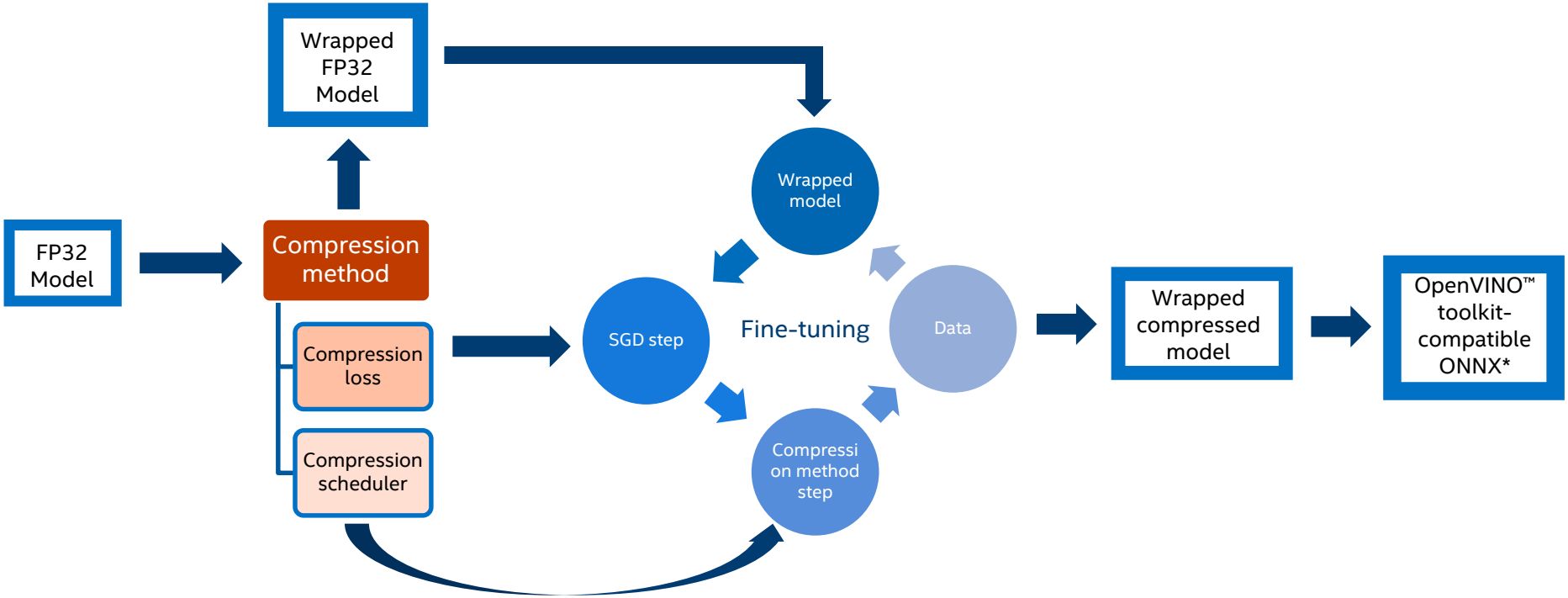


Neural networks compression framework: PyTorch*

OpenVINO™

- PyTorch*-based solution to get compressed DL models with one stage of fine-tuning
- **Features:**
  - INT8 quantization (per-channel/per-tensor quantization, symmetric/asymmetric, etc.)
  - Two sparsity algorithms
  - Image Classification, Object Detection, Semantic Segmentation samples
  - Support of standalone usage within the existing training pipeline with minimum adaptations of the training code (mmdetection, transformers)
  - Binary networks

NNCF compression pipeline



Introducing.. Post-Training Optimization Tool

OpenVINO™

Main goal of the tool:

- Transform Deep Learning model into a representation specific for selected type of optimization (e.g. quantization or sparsity) without model retraining
- The tool is extendable to support multiple quantization algorithms

Distributed as a component within the Intel® Distribution of OpenVINO™ toolkit

- Written in Python
- Uses Intel® Distribution of OpenVINO™ toolkit's Python* API to run inference on Intel® architecture (IA)
- Integrated with other Intel® Distribution of OpenVINO™ toolkit tools:
  - Model Optimizer
  - Accuracy Checker
  - Used by Deep Learning Workbench (visual profiling extension of OpenVINO™ toolkit)

Post-Training Optimization Tool – features

Supports quantization of OpenVINO™ toolkit's IR models for various types of Intel® hardware

Learn more: https://docs.openvinotoolkit.org/latest/_compression_algorithms_quantization_README.html

- Two algorithms supported and exposed through Deep Learning Workbench:
  - Default algorithm: essentially a pipeline running three base algorithms:
    i. Activation Channel Alignment (applied to align activation ranges)
    ii. MinMax
    iii. Bias Correction (runs atop naive algorithm; based on minimization of per-channel quantization error)
  - Accuracy-Aware algorithm: preserves accuracy of the resulting model, keeping accuracy drop below threshold
- Provides hardware-specific configurations
- Features per-channel/per-tensor quantization granularity
- Supports symmetric/asymmetric quantization through presets mechanism

Post-Training Optimization Toolkit – features *(continued)*

OpenVINO™

Two default presets are introduced for MinMax and Default algorithms to simplify user experience

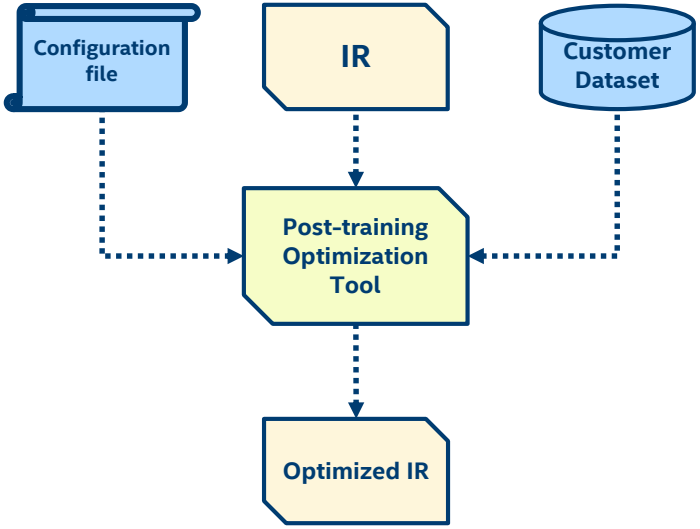
- *Performance preset*—stands for symmetric channel-wise (both for weights and partially for activations) quantization
- *Accuracy preset*—stands for symmetric weights + asymmetric activations quantization

Per-layer quantization tuning is also possible; some layers can be included into ‘ignored scope’ to skip quantization for those

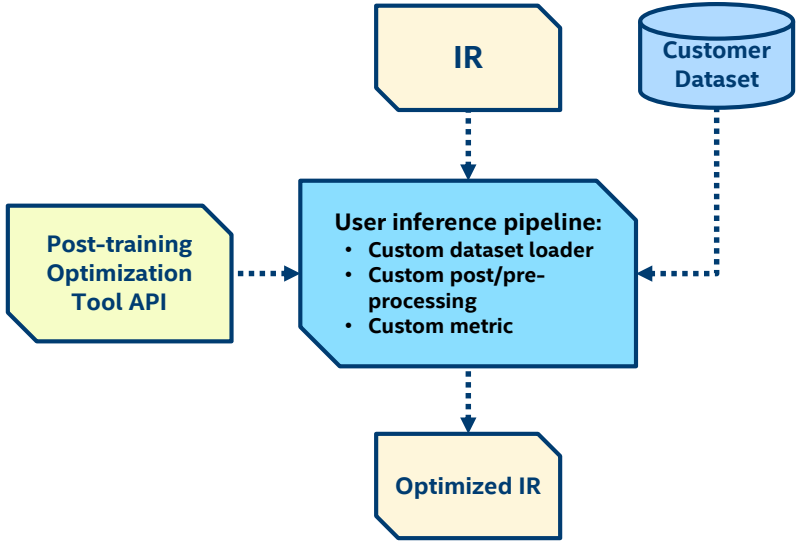
Usage scenarios

OpenVINO™

1 Used as-is. Command line/Workbench scenarios.



2 Integration in user pipeline.



Accuracy and Performance Results

OpenVINO™

- Benchmarks to be posted soon on: https://docs.openvinotoolkit.org/latest/_docs_performance_benchmarks.html
- Quantization-aware training results (NNCF): https://github.com/opencv/openvino_training_extensions/tree/develop/pytorch_toolkit/nncf
- Results from NVIDIA TensorRT* and TensorFlow Lite*:
  - https://developer.download.nvidia.com/video/gputechconf/gtc/2019/presentation/s9659-inference-at-reduced-precision-on-gpus.pdf
  - https://www.tensorflow.org/lite/performance/model_optimization

Key learnings

OpenVINO™

- 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 more 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 the future through

Next steps

OpenVINO™

- Continue developing quantization methods with & without fine-tuning
  - Hardware-aware QAT
  - Annotation-free post-training methods
- Break INT4 barrier based on the community success and own research
- Work on other optimization methods (pruning, block sparsity, NAS)

Useful materials

OpenVINO™

- **Get Started**: https://software.intel.com/openvino-toolkit
- **2020.1 Release Notes**: https://software.intel.com/en-us/articles/OpenVINO-RelNotes
- Community Forum: https://software.intel.com/en-us/forums/intel-distribution-of-openvino-toolkit
- Documentation: https://docs.openvinotoolkit.org/latest/_docs_IE_DG_Tools_Overview.html
- Blog: https://www.intel.ai/open-vino-low-precision-pipeline/
- QAT by Google*: https://arxiv.org/pdf/1712.05877v1.pdf

Notices & Disclaimers

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

The benchmark results reported herein may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
Notice revision #20110804

QUESTIONS & ANSWERS

TECH.
DECODED

intel
Software