MODEL OPTIMIZER

INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

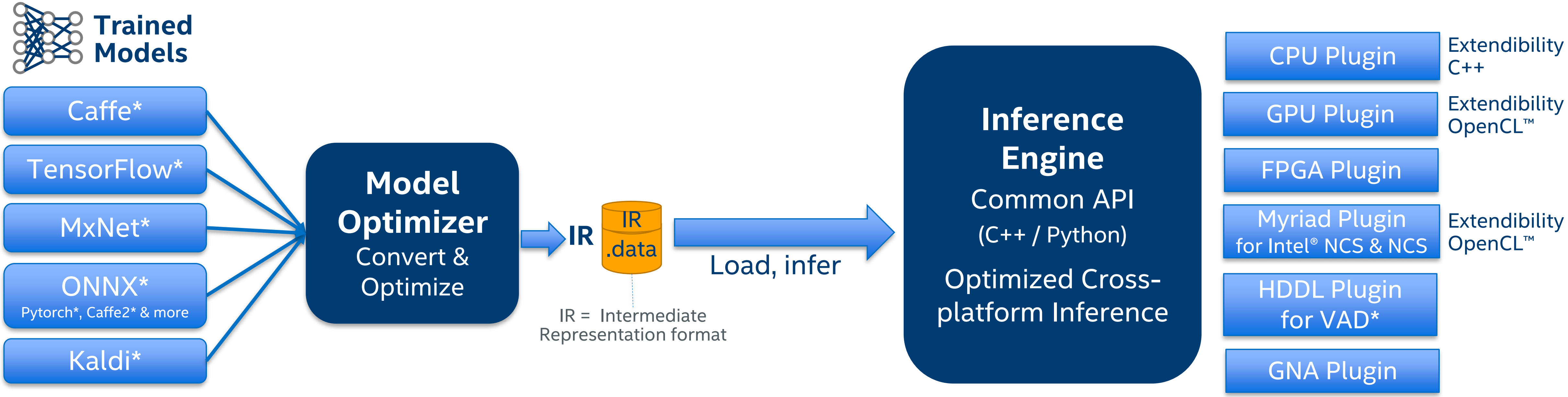
FOR DEEP LEARNING INFERENCE

Model Optimizer

- A Python* based tool to **import** trained models and **convert** them to Intermediate Representation
- **Optimizes for performance** or space with conservative topology transformations
- **Hardware-agnostic** optimizations

Inference Engine

- High-level, C/C++ and Python, inference **runtime API**
- Interface is implemented as **dynamically loaded plugins** for each hardware type
- Delivers advanced performance for each type **without requiring users to implement and maintain multiple code pathways**



GPU = Intel® CPU with integrated GPU/Intel® Processor Graphics, Intel® NCS = Intel® Neural Compute Stick (VPU)
*VAD = Intel® Vision Accelerator Design Products (HDDL-R)

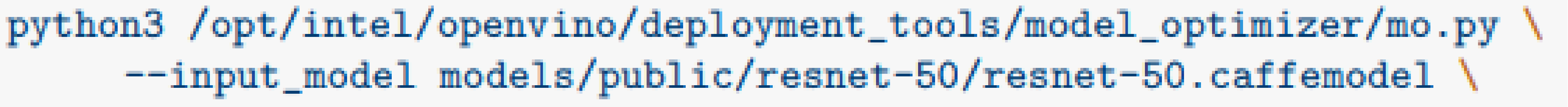
MODEL OPTIMIZER: GENERIC OPTIMIZATION

Model optimizer performs generic optimization

- Node merging
- Horizontal fusion
- Batch normalization to scale shift
- Fold scale shift with convolution
- Drop unused layers (dropout)

The simplest way to convert a model is to run mo.py with a path to the input model file

- By default, generic optimization will be automatically applied, unless manually set disable

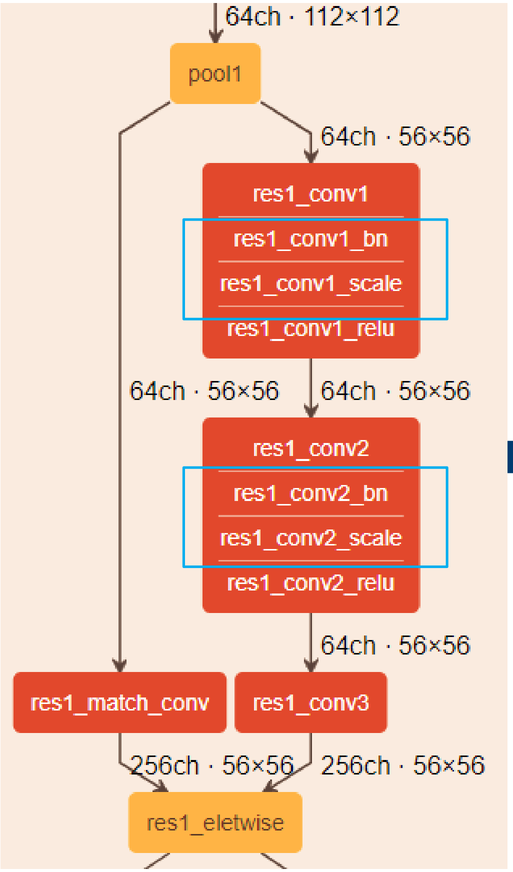


```
python3 /opt/intel/openvino/deployment_tools/model_optimizer/mo.py \
     --input_model models/public/resnet-50/resnet-50.caffemodel \
```

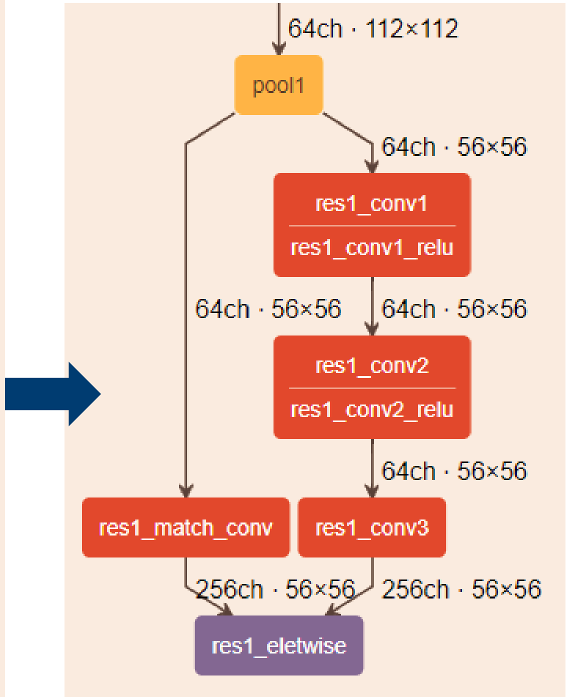
MODEL OPTIMIZATION TECHNIQUES

Linear Operation Fusing: 3 stages

1. **BatchNorm and ScaleShift decomposition:** *BN* layers decomposes to *Mul->Add->Mul->Add* sequence; ScaleShift layers decomposes to *Mul->Add* sequence.
2. **Linear operations merge:** Merges sequences of Mul and Add operations to the **single** Mul->Add instance.
3. **Linear operations fusion:** Fuses Mul and Add operations to Convolution or FullybConnected layers.



Caffe* Resnet269 block (from Netscope)

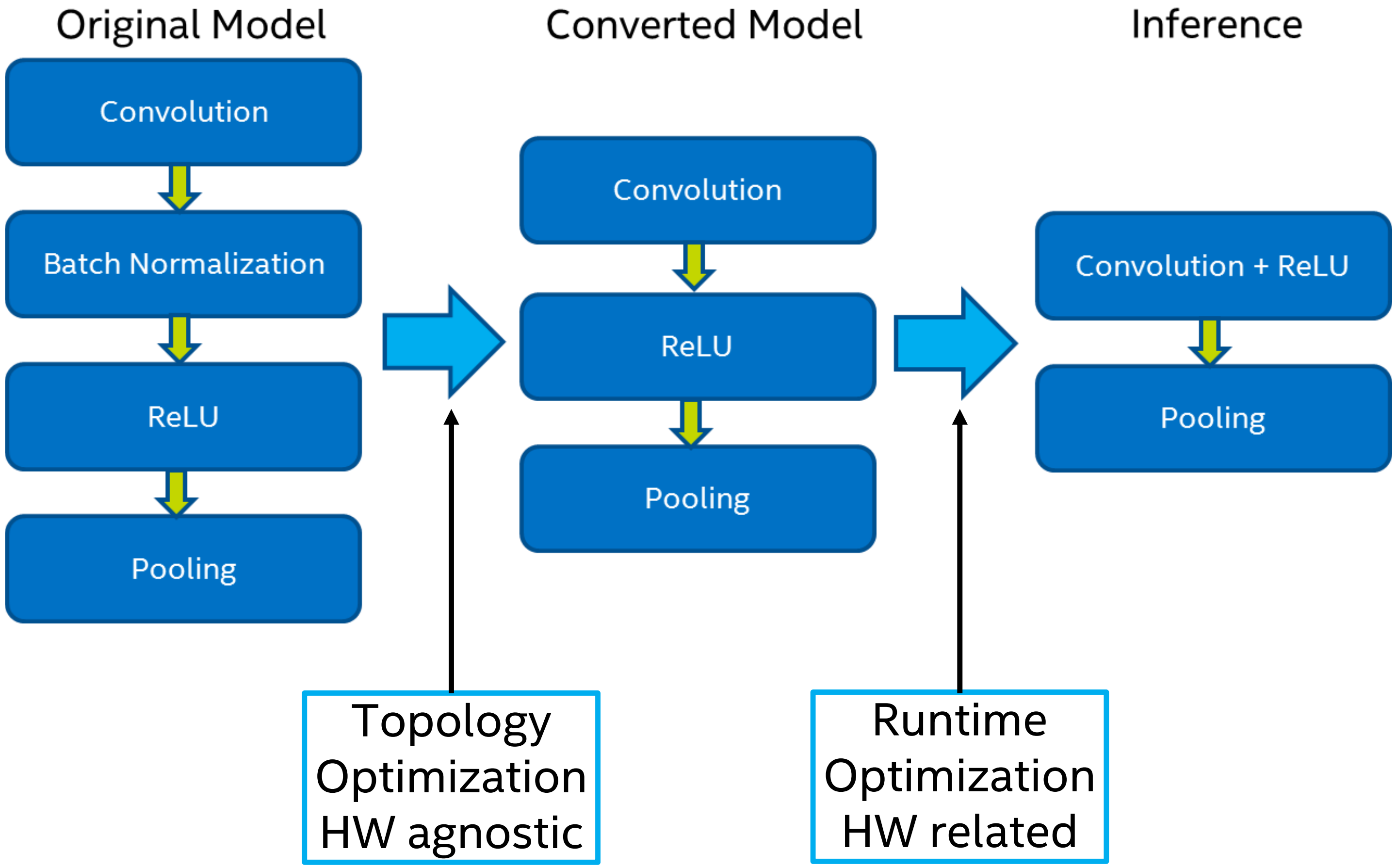


Merged Caffe* Resnet269 block (from Netscope*)

MODEL OPTIMIZER: LINEAR OPERATION FUSING

Example

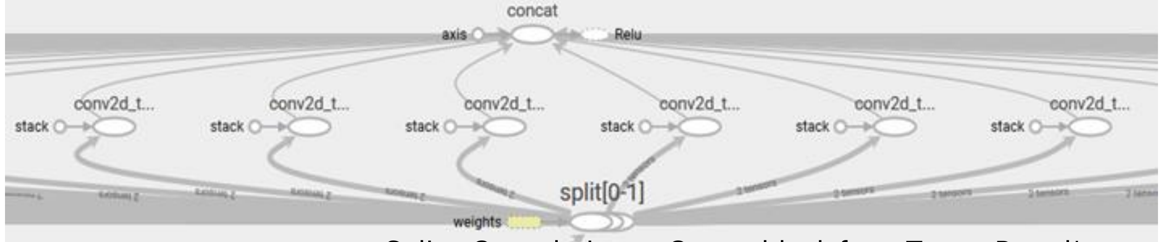
1. Remove Batch normalization stage.
2. Recalculate the weights to ‘include’ the operation.
3. Merge Convolution and ReLU into one optimized kernel.



MODEL OPTIMIZER: FRAMEWORK OR TOPOLOGY SPECIFIC OPTIMIZATION

Grouped Convolutions Fusing

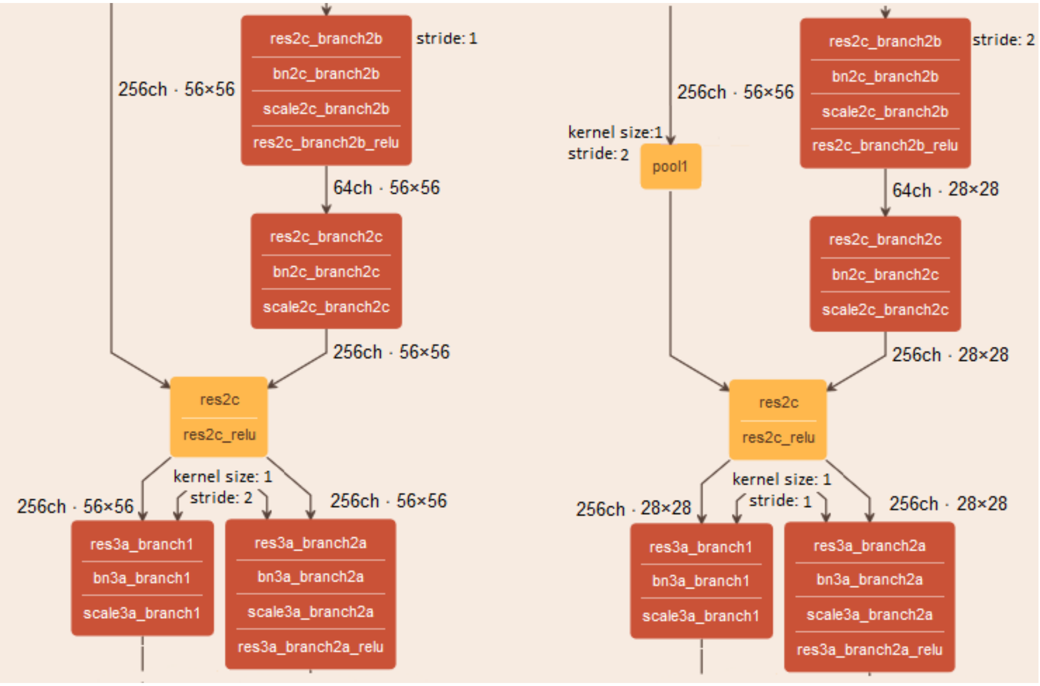
- Grouped convolution fusing is a specific optimization that applies for TensorFlow* topologies. The main idea of this optimization is to combine convolutions results for the Split outputs and then recombine them using **Concat** operation in the same order as they were out from **Split**.



Split->Convolutions->Concat block from TensorBoard*

ResNet* optimization (stride optimization)

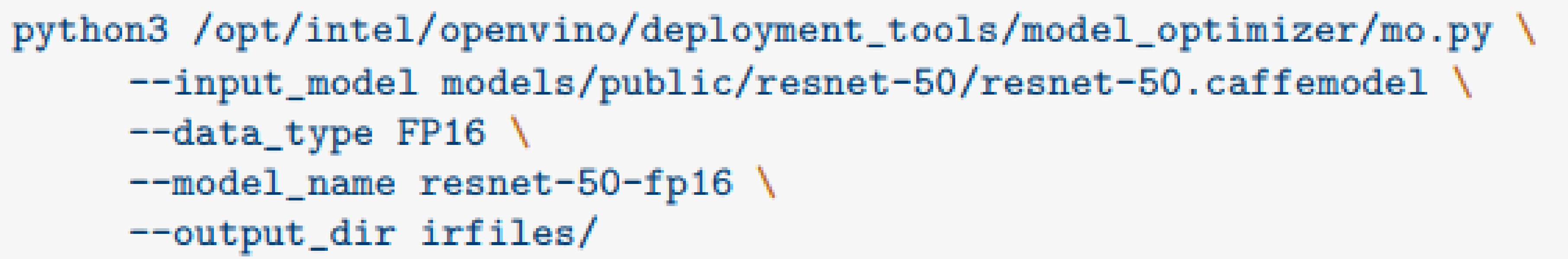
- This optimization is to move the stride that is greater than 1 from Convolution layers with the kernel size = 1 to upper Convolution layers. In addition, the Model Optimizer adds a Pooling layer to align the input shape for a Eltwise layer, if it was changed during the optimization.



MODEL OPTIMIZER: QUANTIZATION

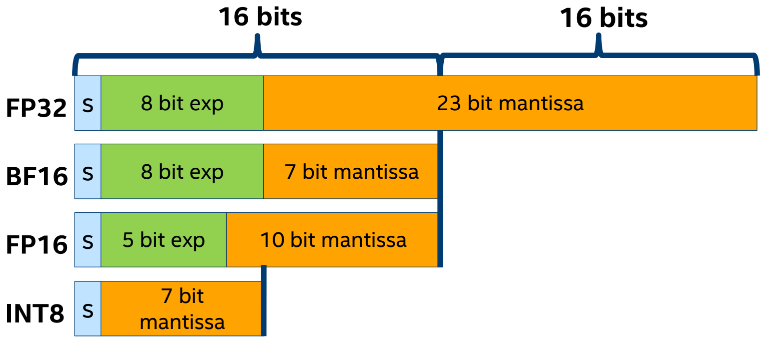
--data_type {FP16,FP32,half,float}

- Data type for all intermediate tensors and weights.
- If original model is in FP32 and --data_type=FP16 is specified, all model weights and biases are quantized to FP16.



```
python3 /opt/intel/openvino/deployment_tools/model_optimizer/mo.py \
     --input_model models/public/resnet-50/resnet-50.caffemodel \
     --data_type FP16 \
     --model_name resnet-50-fp16 \
     --output_dir irfiles/
```

PLUGIN	FP32	FP16	INT8
CPU plugin	Supported and preferred	Supported	Supported
GPU plugin	Supported	Supported and preferred	Supported*
VPU plugins	Not supported	Supported	Not supported
GNA plugin	Supported	Supported	Not supported
FPGA plugin	Supported	Supported	Not supported

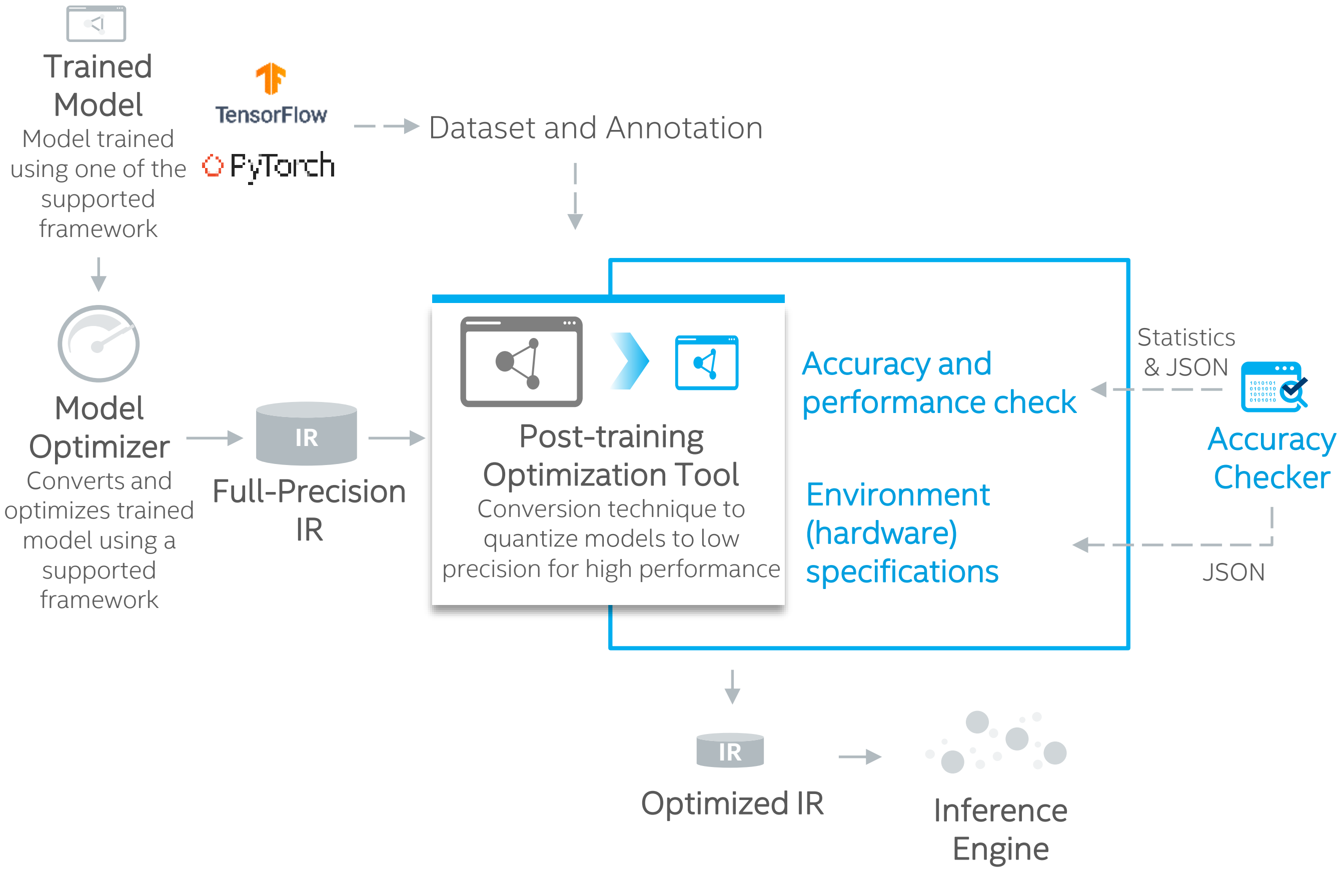


Note:
1. To create INT8 models, you will need DL Workbench or Post Training Optimization Tool
2. FPGA also support FP11, convert happens on FPGA

Post-Training Optimization Tool

- Using the Python API, the Post-training Optimization Tool integrates with the Model Optimizer, DL Workbench and accuracy checker tools to streamline the development process
- Enables a conversion technique of deep learning model that **reduces model size into low precision data types**, such as INT8, without re-training
- Reduces model size **while also improving latency, with little degradation** in model accuracy and without model re-training.
- Different optimization approaches are supported: quantization algorithms, sparsity, etc.

Performance Benchmarks ▸
https://docs.openvinotoolkit.org/latest/_docs_performance_int8_vs_fp32.html


Trained Model
Model trained using one of the supported framework
TensorFlow
PyTorch
Dataset and Annotation
Model Optimizer
Converts and optimizes trained model using a supported framework
IR
Full-Precision IR
Post-training Optimization Tool
Conversion technique to quantize models to low precision for high performance
Accuracy and performance check
Environment (hardware) specifications
Statistics & JSON
Accuracy Checker
JSON
IR
Optimized IR
Inference Engine

SPEED UP DEVELOPMENT WITH OPEN SOURCE RESOURCES

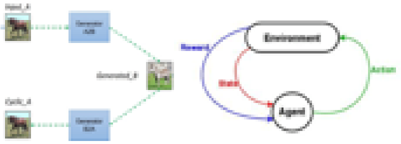
Open source resources with pre-trained models, samples and demos

Computer Vision

Object detection
Human pose estimation
Object recognition
Image processing
Reidentification
Action recognition
Volumetric segmentation
Image super resolution
Semantic segmentation
Instance segmentation
3D reconstruction

Audio, Speech, Language

Language processing
Speech to text
Text detection
Text recognition
Natural Language Processing



Other

(Data Generation, Reinforcement Learning)

Compressed models
Image retrieval

And more..

Model Downloader

- Provides an easy way of accessing a number of public models as well as a set of pre-trained Intel models

Accuracy Checker

- Check for accuracy of the model (original and after conversion) to IR file using a known data set

PRE-TRAINED MODELS

https://github.com/opencv/open_model_zoo