



FPGAs FOR DEEP LEARNING

Intel Network & Custom Logic Group

Greg Nash, System Architect – Government, Aerospace & Military

Jim Moawad, Technical Solution Specialist – FPGA Acceleration

July 29, 2019 - Argonne Training Program on Extreme Scale Computing

Public

NOTICES & DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No product or component can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit <http://www.intel.com/benchmarks>.

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 <http://www.intel.com/benchmarks>.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at <http://www.intel.com/go/turbo>.

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.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

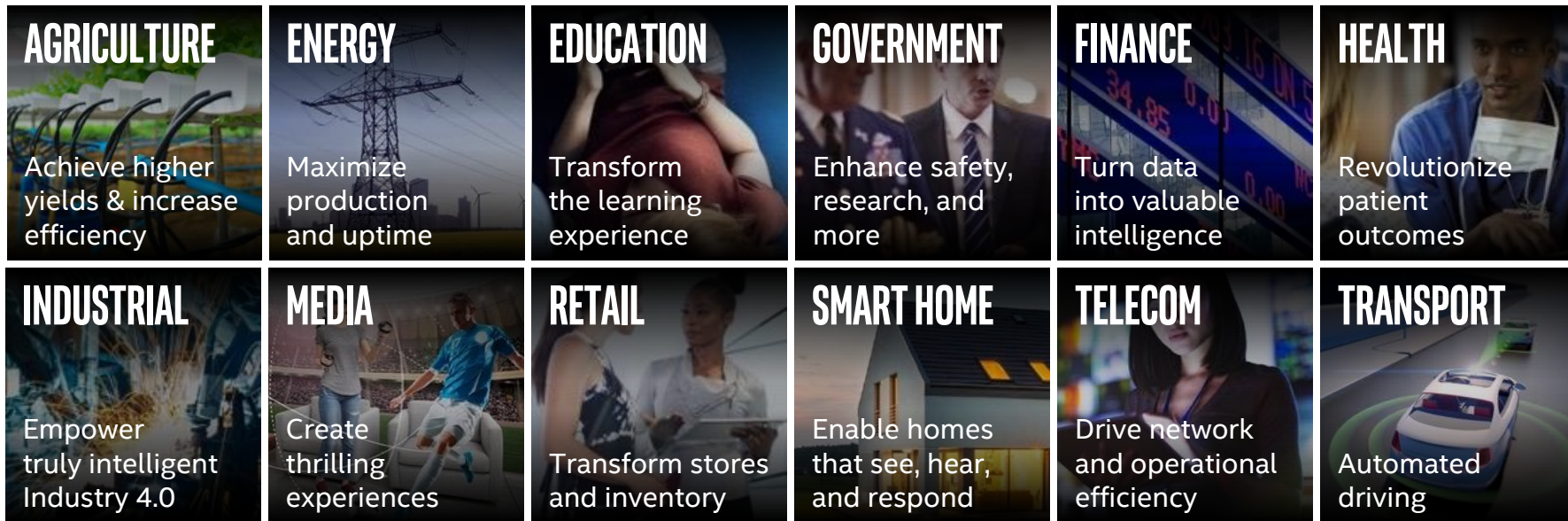
Intel, the Intel logo, Intel Atom, Intel Core, Intel Nervana, Intel Optane, Intel Xeon, Iris, Movidius, OpenVINO, Pentium, Celeron, Stratix, Arria and the Stratix logo and are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as property of others.

AGENDA

- **Intel® and AI / Machine Learning**
- Accelerate Deep Learning Using OpenVINO Toolkit
- Deep Learning Acceleration with FPGA
 - FPGAs and Machine Learning
 - Intel® FPGA Deep Learning Acceleration Suite
 - Execution on the FPGA (Model Optimizer & Inference Engine)
- Intel® Agilex® FPGA
- OneAPI

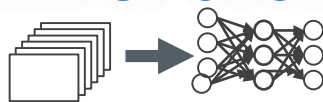
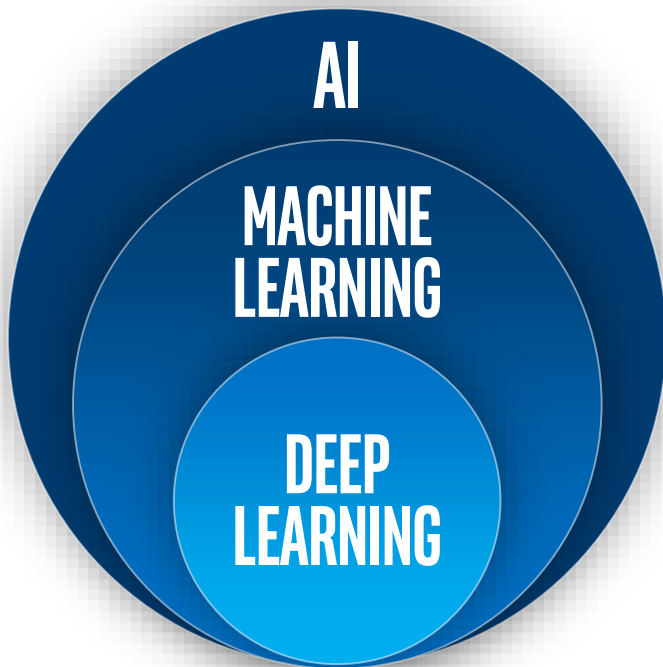
MACHINE LEARNING APPLIES TO NEARLY EVERY MARKET



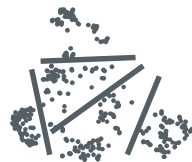
Wide variety of market segments incorporating AI and Deep Learning which is increasing compute demands across the board.

AI & MACHINE LEARNING IS A VAST FIELD OF STUDY

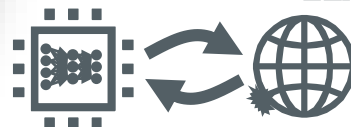
- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation
- Image Processing
- Speech Processing
- Natural Language Processing
- Recommender Systems
- Adversarial Networks
- Reinforcement Learning



**SUPERVISED
LEARNING**



**UNSUPERVISED
LEARNING**

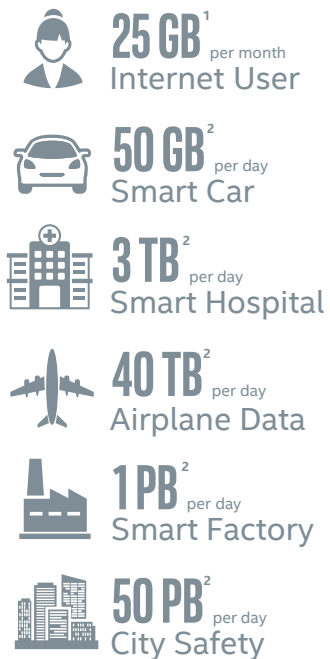


**REINFORCEMENT
LEARNING**

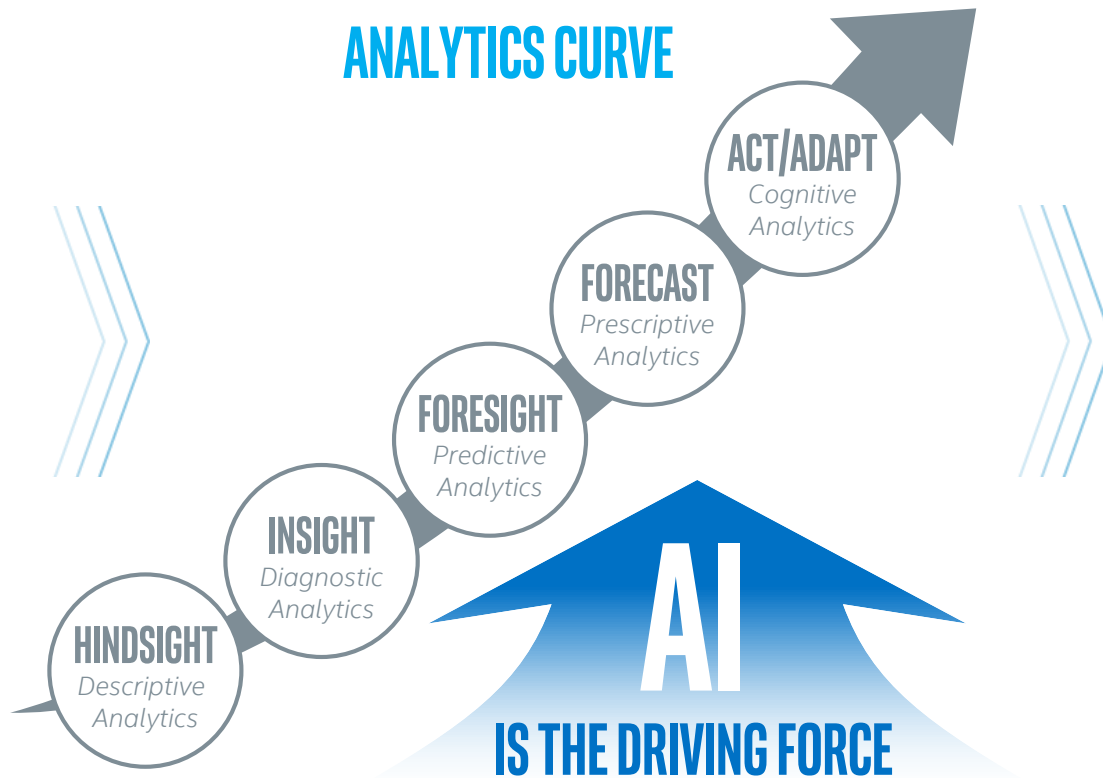
No one size fits all approach to AI

DATA IS DRIVING THE DESIRE FOR AI

DATA DELUGE (2019)



ANALYTICS CURVE



INSIGHTS



BUSINESS



OPERATIONAL



SECURITY

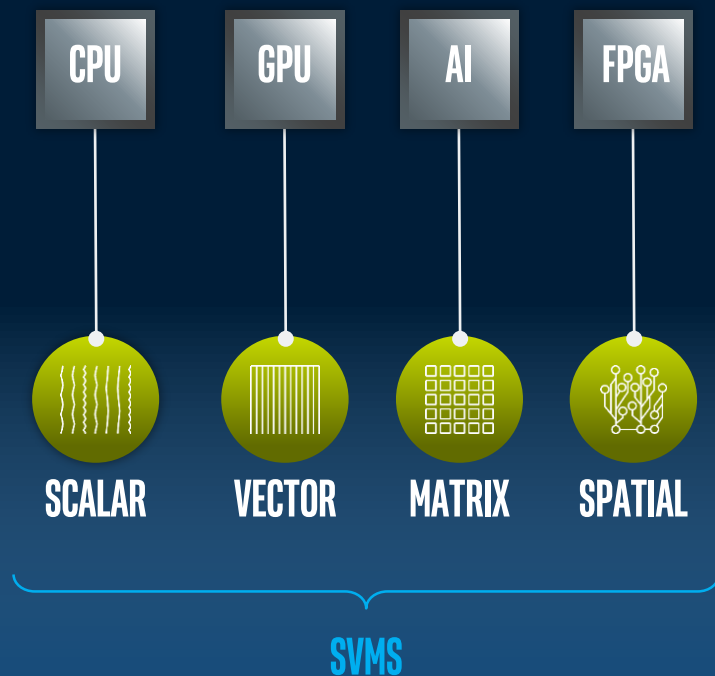
AI
IS THE DRIVING FORCE

1. Source: <http://www.cisco.com/c/en/us/solutions/service-provider/vni-network-traffic-forecast/infographic.html>

2. Source: https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/547_11_10-15-DocumentsCisco_GCI_Deck_2014-2019_for_CKN_10NOV2015_.pdf

DIVERSE WORKLOADS REQUIRE DIVERSE ARCHITECTURES

The future is a **diverse** mix of scalar, vector, matrix, and spatial **architectures** deployed in CPU, GPU, AI, FPGA and other accelerators



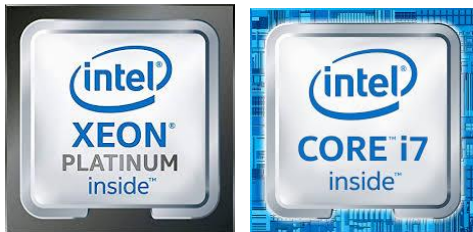
AI COMPUTE GUIDE

CPU

ACCELERATORS

FPGA, VPU, GPU

INFERENCE



Foundation for all workloads including AI
Intel® DL Boost for inference & training acceleration

TRAINING



NNP-I

Highly efficient multi-model inferecing for cloud, data center and appliances at scale



NNP-T

Fastest time-to-train at scale with high bandwidth AI server connections for the most persistent, intense usage



Delivering Real time workflows with AI for optimized system performance



Energy efficient Edge AI for Video/Vision



GPU discrete & Integrated, optimized for highly parallel workload acceleration

Multi models – high performance across broad set of existing and emerging DL models for applications like Natural Language Processing, Speech & Text, Recommendation Engines, and Video & Image search & filtering

A+ - Combined workflows where use cases span AI plus additional workloads (Eg - Immersive Media - Media + Graphics in GPU or L3 forwarding in FPGA)



TAME YOUR DATA

with a robust data layer

011010110110
110101101011
001011010100



SOURCE(S)?
STRUCTURED?
VOLUME?
DURABILITY?
STREAMING?
LOCALITY?
GOVERNANCE?
OTHER?

INGEST

Tool for **live streaming data ingestion** from Internet of Things (IOT) sensors in endpoint devices

STORE

File, block or object-based storage solution given cost, access, volume and perf requirements

PROCESS

Integration, cleaning, normalization and more transformations on batch and/or streaming data

ANALYZE

Applications in **HPC, Big Data, HPDA, AI** & more that have access to a common compute and data pool

Visit: intel.com/content/www/us/en/analytics/tame-the-data-deluge.html

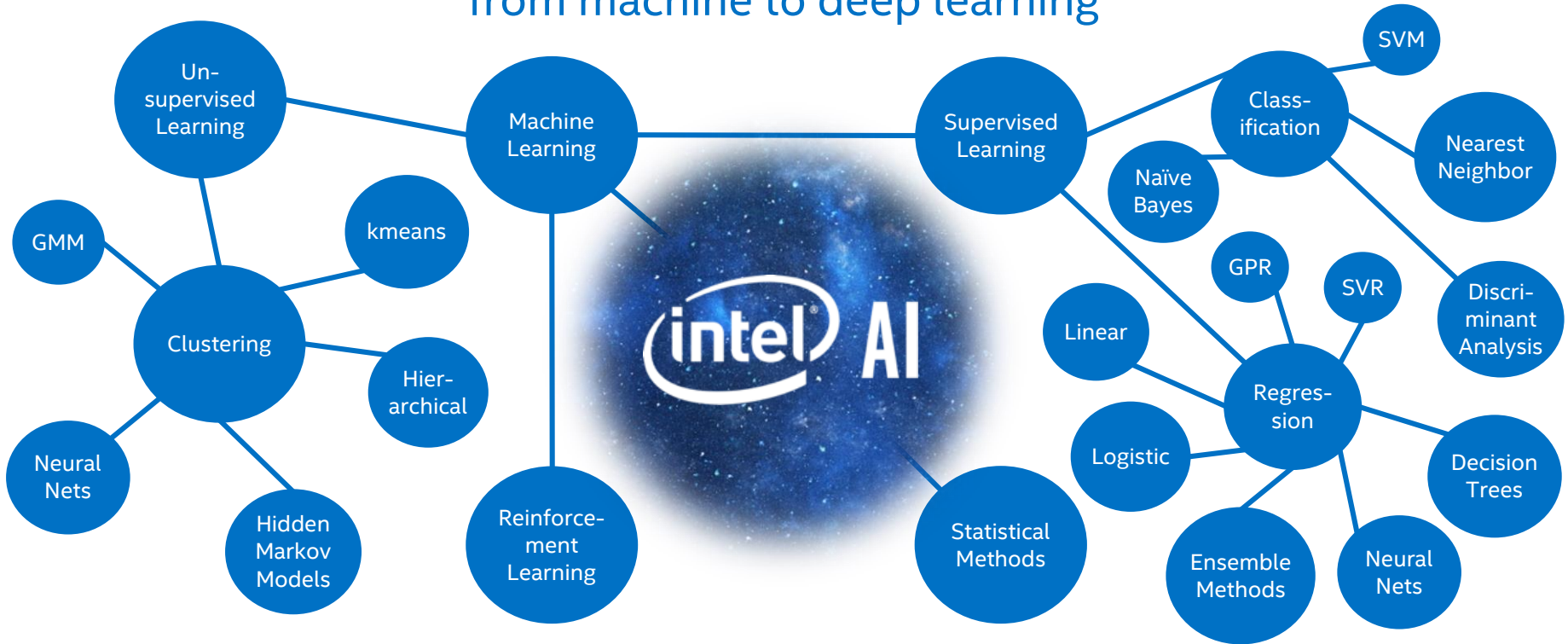
*Other names and brands may be claimed as the property of others

Visit:

software.intel.com/ai/courses

CHOOSE ANY APPROACH

from machine to deep learning



Visit:

www.intel.ai/technology



SPEED UP DEVELOPMENT

using open AI software



TOOLKITS
App developers



Open source platform for building E2E Analytics & AI applications on Apache Spark* with distributed TensorFlow*, Keras*, BigDL



Deep learning inference deployment on CPU/GPU/FPGA/VPU for Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*



Open source, scalable, and extensible distributed deep learning platform built on Kubernetes (BETA)



LIBRARIES
Data scientists

Python

- Scikit-learn
- Pandas
- NumPy

R

- Cart
- Random Forest
- e1071

Distributed

- MLlib (on Spark)
- Mahout



Intel-optimized Frameworks



And more framework optimizations underway including PaddlePaddle*, Chainer*, CNTK* & others



KERNELS
Library developers

Intel® Distribution for Python*

Intel distribution optimized for machine learning

Intel® Data Analytics Acceleration Library (DAAL)

High performance machine learning & data analytics library

Intel® Math Kernel Library for Deep Neural Networks (MKL-DNN)

Open source DNN functions for CPU / integrated graphics



Open source compiler for deep learning model computations optimized for multiple devices (CPU, GPU, NNP) from multiple frameworks (TF, MXNet, ONNX)

*An open source version is available at: 01.org/openvintoolkit

*Other names and brands may be claimed as the property of others.

Developer personas show above represent the primary user base for each row, but are not mutually-exclusive

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

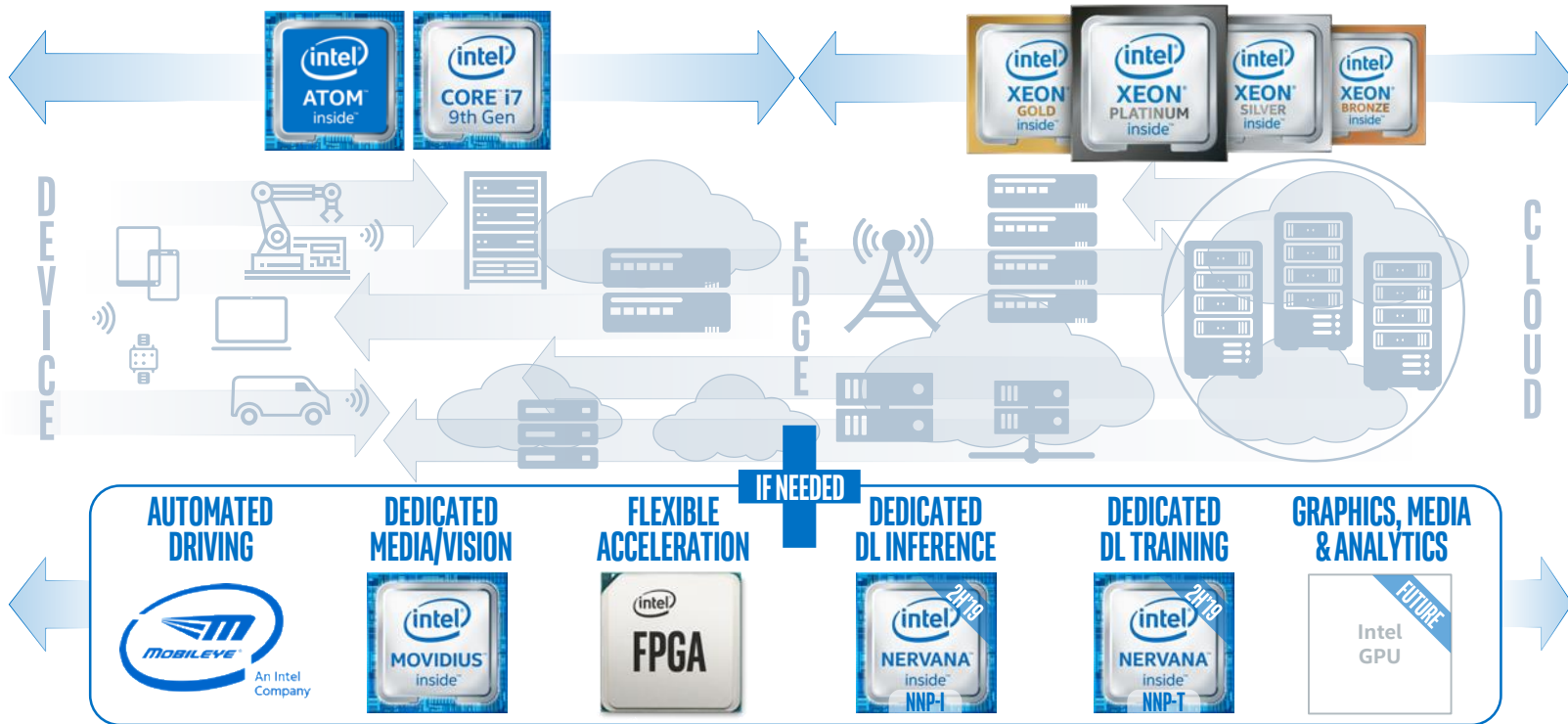


Visit:

www.intel.ai/technology

DEPLOY AI ANYWHERE

with unprecedented hardware choice



All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

DEEP LEARNING DEPLOYED

DATA

TOOLS

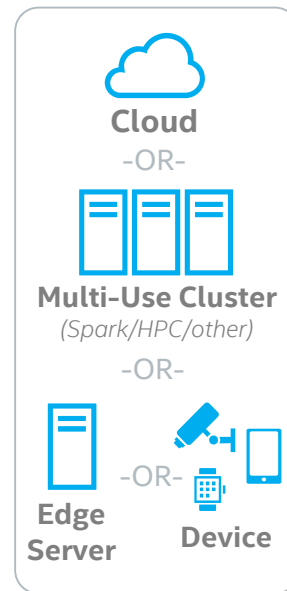
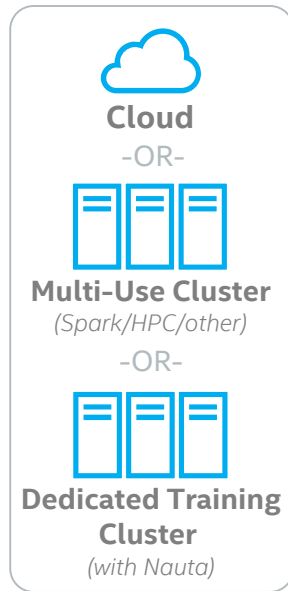
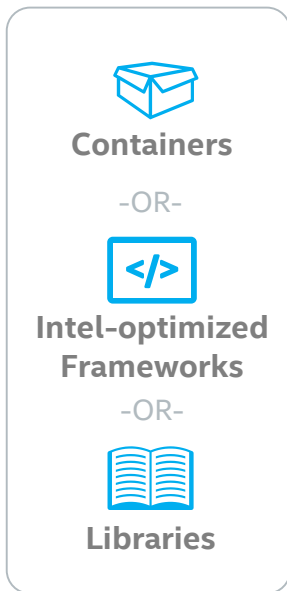
TRAINING

MODEL

OPTIMIZATION

INFERENCE

011010110110
110101101011
001011010100
011010110110
110101101011
001011010100
011010110110
110101101011
001011010100
011010110110
110101101011
001011010100
011010110110
110101101011
001011010100
011010110110
110101101011
001011010100
011010110110
110101101011
001011010100
011010110110
110101101011
001011010100



End-to-end deep learning on Intel



DEEP LEARNING PRODUCT BRIEFS

2ND GENERATION INTEL® XEON® SCALABLE PROCESSOR

formerly known as Cascade Lake



Drop-in compatible CPU on Intel® Xeon® Scalable platform

TCO/FLEXIBILITY

Begin your AI journey efficiently, now with even more agility...

- ✓ IMT – Intel® Infrastructure Management Technologies
- ✓ ADQ – Application Device Queues
- ✓ SST – Intel® Speed Select Technology

PERFORMANCE

Built-in Acceleration with Intel® Deep Learning Boost...



deep learning throughput!¹

Throughput (img/s)

SECURITY

Hardware-Enhanced Security...

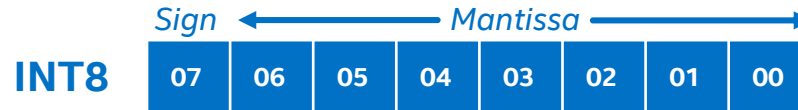
- ✓ Intel® Security Essentials
- ✓ Intel® Secl: Intel® Security Libraries for Data Center
- ✓ TDT – Intel® Threat Detection Technology

¹ Based on Intel internal testing: 1X, 5.7x, 14x and 30x performance improvement based on Intel® Optimization for Café ResNet-50 inference throughput performance on Intel® Xeon® Scalable Processor. See Configuration Details 3 Performance results are based on testing as of 7/11/2017(1x), 11/8/2018 (5.7x), 2/20/2019 (14x) and 2/26/2019 (30x) and may not reflect all publicly available security updates. No product can be absolutely secure. See configuration disclosure for details.

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. 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: <http://www.intel.com/performance>

INTEL[®] DEEP LEARNING BOOST (DL BOOST)

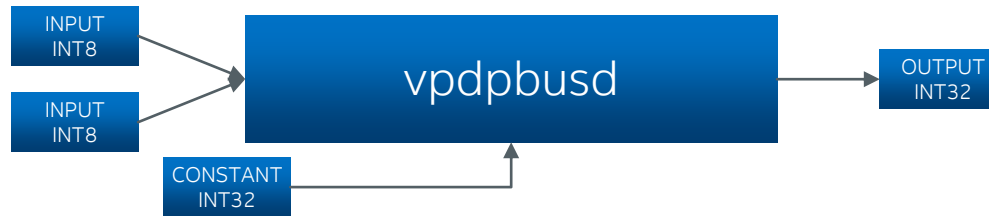
featuring Vector Neural Network Instructions (VNNI)



Current AVX-512 instructions to perform INT8 convolutions: `vpaddubsw`, `vpaddwd`, `vpadd`

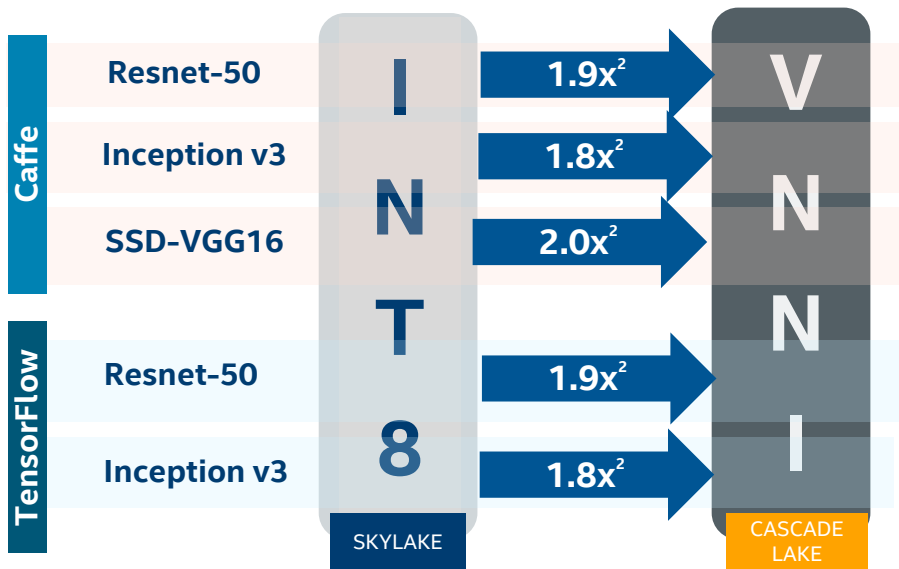
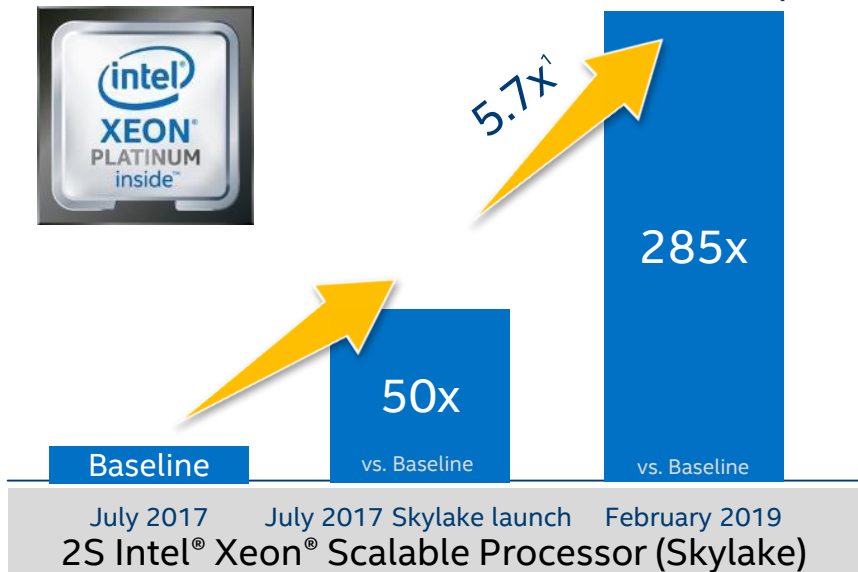


NEW AVX-512 (VNNI) instruction to accelerate INT8 convolutions: `vpdpbusd`



DEEP LEARNING PERFORMANCE ON CPU

Hardware + Software Improvements for Intel® Xeon® Processors



¹ 5.7x inference throughput improvement with Intel® Optimizations for Caffe ResNet-50 on Intel® Xeon® Platinum 8180 Processor in Feb 2019 compared to performance at launch in July 2017. See configuration details on Config 1 Performance results are based on testing as of dates shown in configuration and may not reflect all publicly available security updates.
²8/24/2018) Results have been estimated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance. 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. No product can be absolutely secure. See configuration disclosure for details. 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. 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: <http://www.intel.com/performance>

INTEL® NERVANA™ NEURAL NETWORK PROCESSORS (NNP)‡



NNP-T
DEDICATED
DL TRAINING



Fastest time-to-**train** with high bandwidth AI server connections for the most persistent, intense usage

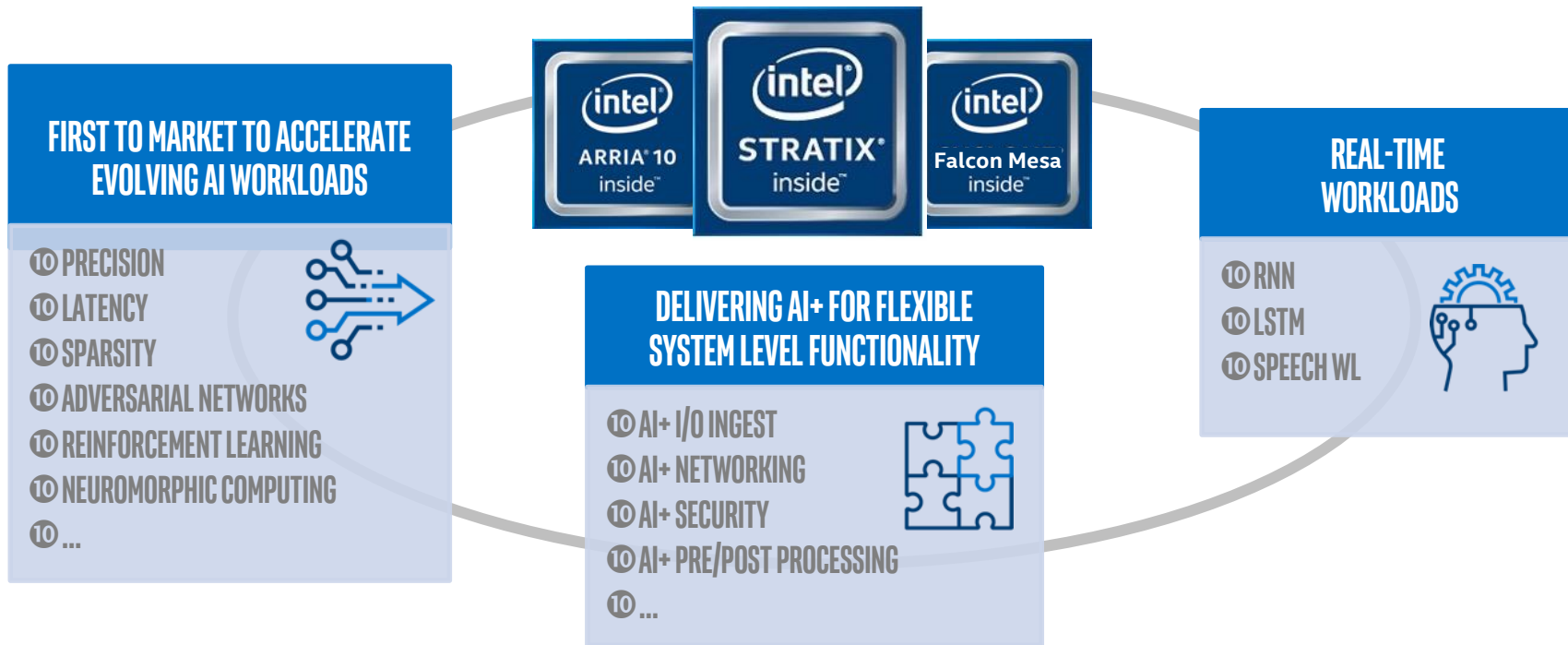


NNP-I
DEDICATED
DL INFERENCE



Highly-efficient multi-model **inference** for cloud, data center and intense appliances

INTEL® FPGA FOR AI



Enabling real-time AI in a wide range of embedded, edge and cloud apps

INTEL® MOVIDIUS™ VISION PROCESSING UNIT (VPU)

SERVICE ROBOTS

- Navigation
- 3D Vol. mapping
- Multimodal sensing



SURVEILLANCE

- Detection/classification
- Identification
- Multi-nodal systems
- Multimodal sensing
- Video, image capture



WEARABLES

- Detection, tracking
- Recognition
- Video, image, session capture



DRONES

- Sense and avoid
- GPS denied hovering
- Pixel labeling
- Video, image capture



AR-VR HMD

- 6DOF pose, position, mapping
- Gaze, eye tracking
- Gesture tracking, recognition
- See-through camera



SMART HOME

- Detection, tracking
- Perimeter, presence monitoring
- Recognition, classification
- Multi-nodal systems
- Multimodal sensing
- Video, image capture

Power-efficient image processing, computer vision & deep learning for devices

INTEL® GAUSSIAN NEURAL ACCELERATOR (GNA)

AMPLE THROUGHPUT

For speech, language, and other sensing inference

LOW POWER

<100 mW power consumption for always-on applications

FLEXIBILITY

Gaussian mixture model (GMM) and neural network inference support



TRY IT TODAY!



Intel® Speech Enabling Developer Kit

<https://software.intel.com/en-us/iot/speech-enabling-dev-kit>

Learn more: <https://sigport.org/sites/default/files/docs/PosterFinal.pdf>

Streaming Co-Processor for Low-Power Audio Inference & More

INTEL INTEGRATED PROCESSOR GRAPHICS

UBIQUITY/SCALABILITY

- Shipped in 1 billion+ Intel® SoCs
- Broad choice of performance/power across Intel Atom®, Intel® Core™, and Intel® Xeon® processors

MEDIA LEADERSHIP

- Intel® Quick Sync Video – fixed-function media blocks to improve power and performance
- Intel® Media SDK – API that provides access to hardware-accelerated codecs

HARDWARE INTEGRATION



POWERFUL, FLEXIBLE ARCHITECTURE

- Rich data type support for 32bitFP, 16bitFP, 32bitInteger, 16bitInteger with SIMD multiply-accumulate instructions

MEMORY ARCHITECTURE

- Shared memory architecture on die between CPU and GPU to enable lower latency and power

SOFTWARE SUPPORT

MacOS (CoreML and MPS¹)
Windows O/S (WinML)
OpenVINO™ Toolkit (Win, Linux)
cLDNN

Built-in deep learning inference acceleration

MOBILEYE – AN INTEL COMPANY



Visit www.mobileye.com

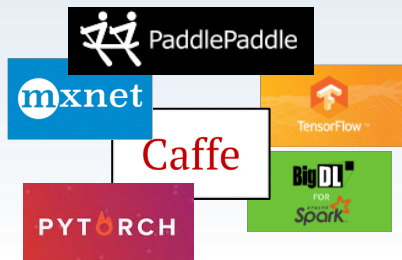
Automated Driving Platform

DEEP LEARNING SOFTWARE

FRAMEWORK

Provides a standard way to customize and deploy Deep Neural Networks

Supplies generic functionality which can be selectively extended with user defined additions to enhance specific tasks



TOPOLOGY

A set of algorithms modeled loosely after the human brain that forms a “network” designed to recognize complex patterns

Also referred to as a network, NN, or model.



LIBRARY

Highly optimized functions intended to be used in Neural Network implementations to maximize performance

Libraries are framework and model agnostic, hardware specific

MACHINE LEARNING LIBRARIES

Python R Distributed

- [Scikit-learn](#)
- [Pandas](#)
- [NumPy](#)
- [Cart](#)
- [Random Forest](#)
- [e1071](#)
- [MLlib \(on Spark\)](#)
- [Mahout](#)

DAAL

Intel® Data Analytics Acceleration Library (includes machine learning)

MKL-DNN

Open-source deep neural network functions for CPU / integrated graphics

c1DNN

FRAMEWORKS

Google



Amazon

MXNet

Microsoft



Facebook



ONNX promises to ease interoperability between frameworks.



ONNX

Open Neural Network Exchange (ONNX)

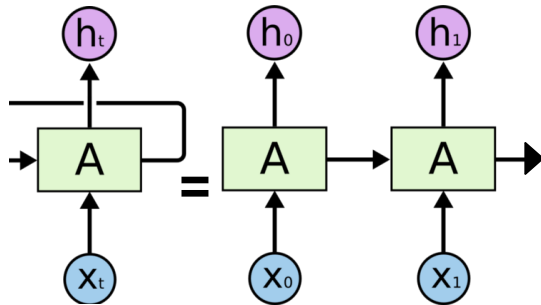
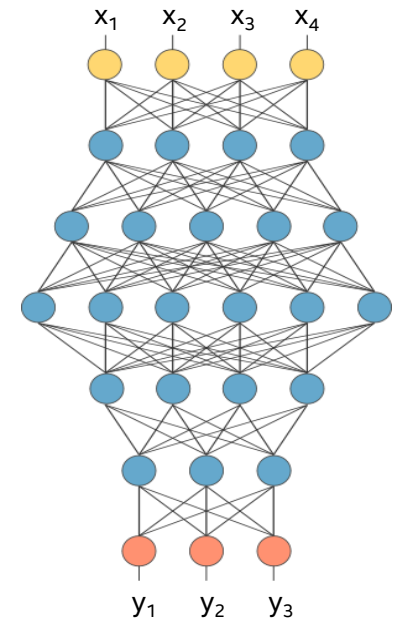
- Enabling interoperability between different frameworks
- Streamlining the path from research to production

Caffe2, PyTorch, Microsoft Cognitive Toolkit, Apache MXNet

<https://github.com/onnx/onnx>

CONVOLUTIONAL NEURAL NETWORK (CNN)

- Layers of neurons connected by learned weights
- Three layer classes:
 - Convolution Layer
 - Pooling Layer
 - Fully connected Layer
- Connections between the layers point in one direction (Feed Forward Network)
- Used to extract features from images for recognition or detection



RECURRENT NEURAL NETWORK (RNN)

- Allows long-term dependencies to affect the output by relying on internal memory to process inputs
- Types of RNN:
 - LSTM (Long Short Term Memory)
 - GRU (Gated Recurrent Unit)
- Designed to analyze sequential data (i.e., NLP)

IMAGE RECOGNITION

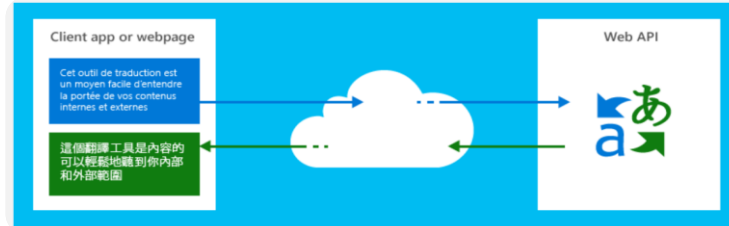
Deep Learning to Detect and Classify Colon Cancer



<http://ieeexplore.ieee.org/document/7124883/>

Workload: ResNet50, InceptionV3, SSD-VGG16

TEXT PROCESSING



<https://www.microsoft.com/en-us/translator/mt.aspx#nnt>

Workload: Neural Machine Translation

CNN

RNN









SPEECH RECOGNITION



<https://www.news-medical.net/whitepaper/20170821/Speech-Recognition-in-Healthcare-a-Significant-Improvement-or-Severe-Healthcare.aspx>

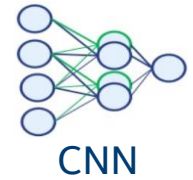
Workload: DeepSpeech2

TOPOLOGIES FOR DEEP LEARNING

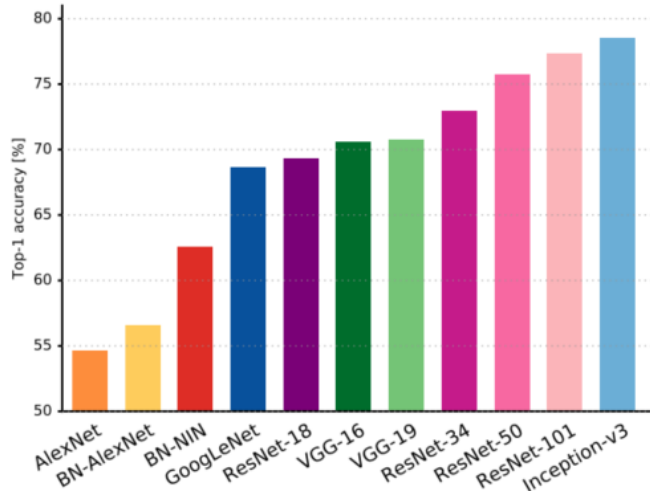
	Purpose	Neural Network Type	Neural Network Example
 Image Recognition	Classify image(s)	CNN - Convolutional Neural Network	ResNet50, Inception V3, Inception_ResV2, MobileNet, SqueezeNet, DenseNet
 Object Detection	Locate AND classify object(s) in image	CNN	SSD- Single Shot Detector, R-FCN, Yolo_V2, VGG_16, DDRN/D-DBPN
 Natural Language Processing (NLP)	Extract context & meaning from text	LTSM (RNN)- Long Short Term Memory	Google NMT
 Speech Recognition	Convert speech to text	RNN - Recursive Neural Network	Baidu DeepSpeech, Transformer
 Text-to-Speech	Convert text to speech	GAN - Generative Adversarial Network with CNN/RNN	WaveNet
 Recommendation Systems	Recommend ads, search, apps, etc.	MLP - Multilayer Perceptron	Wide & Deep MLP, NCF
 Data Generation	Create images like training data	GAN with CNN	DCGAN, DRAW
 Reinforcement Learning	Learning from feedback on action	CNN or CNN+RNN	A3C

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

COMPARISON OF DIFFERENT NETWORKS

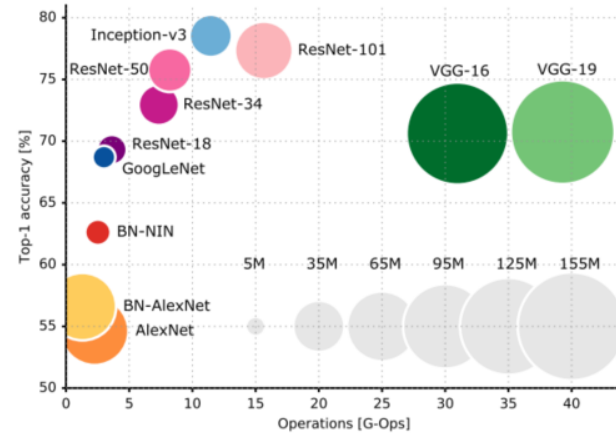


The number of network parameters captured and depth of the networks is not necessarily linear with Top-1 accuracy that can be achieved



Single-Crop top-1 validation accuracies

AlexNet depth: 8 layers, GoogleNet depth: 22 layers, VGG depth: 16 layers or 19 layers, ResNet depth: 50 layers or 101 layers



Single-Crop top-1 validation accuracies versus amount of operation for a single forward pass

Legend in gray circles in the image shows number of network parameters

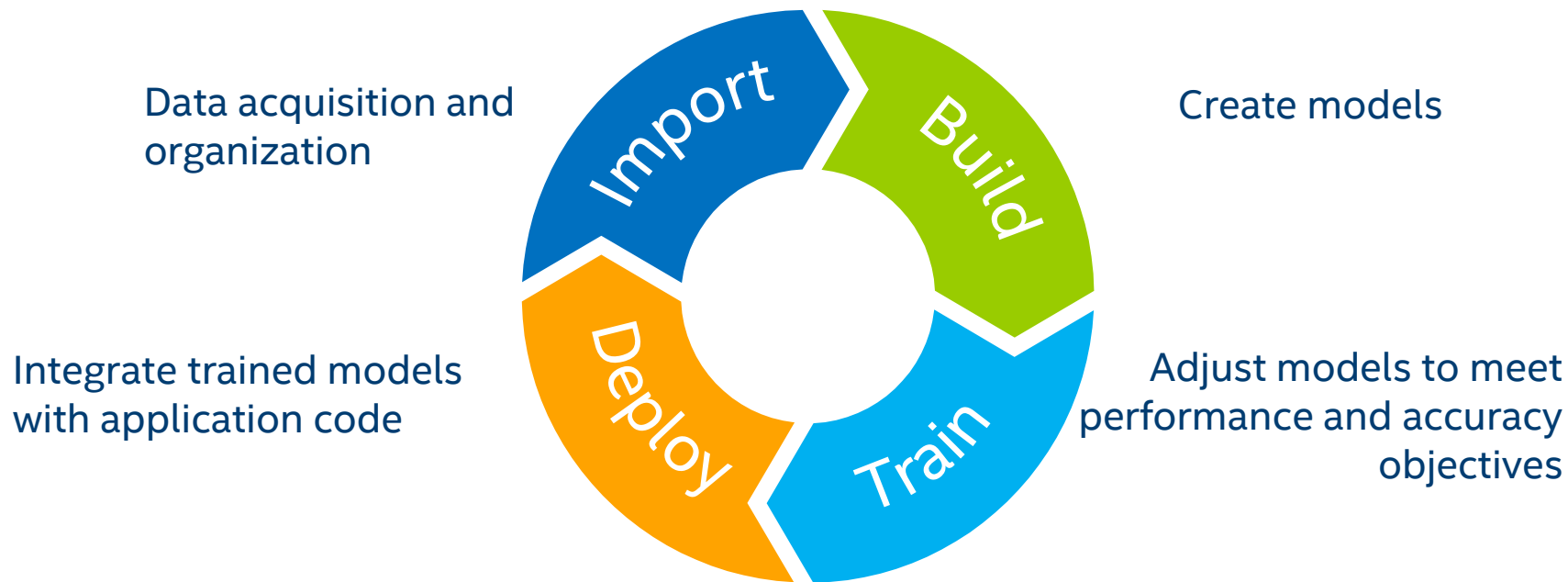
AGENDA

- Intel® and AI / Machine Learning
- **Accelerate Deep Learning Using OpenVINO Toolkit**
- Deep Learning Acceleration with FPGA
 - FPGAs and Machine Learning
 - Intel® FPGA Deep Learning Acceleration Suite
 - Execution on the FPGA (Model Optimizer & Inference Engine)
- Intel® Agilex® FPGA
- OneAPI



ACCELERATE DEEP LEARNING INFERENCE USING INTEL OPENVINO TOOLKIT

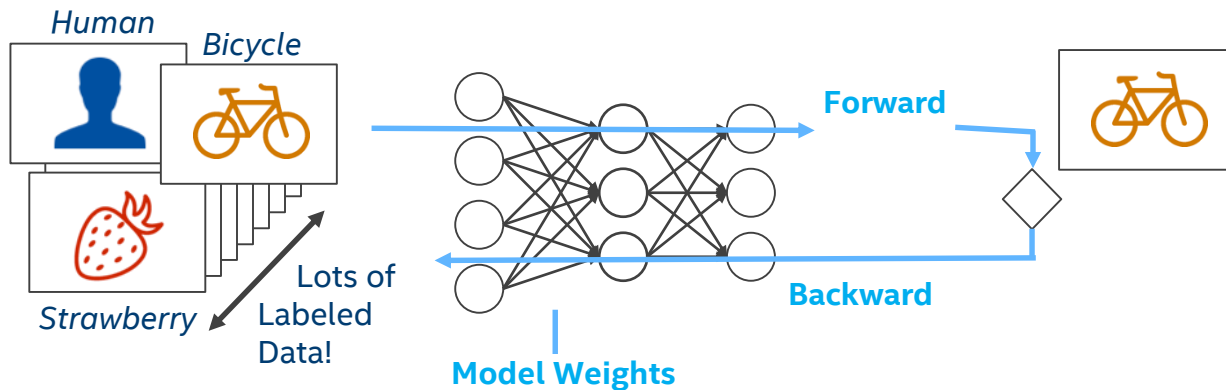
Deep Learning Development Cycle



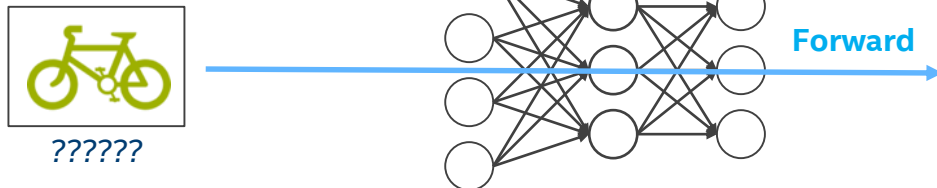
Intel® Distribution OpenVINO™ Toolkit Provides Deployment™ from Intel® Edge to Cloud

Deep Learning: Training vs. Inference

Training

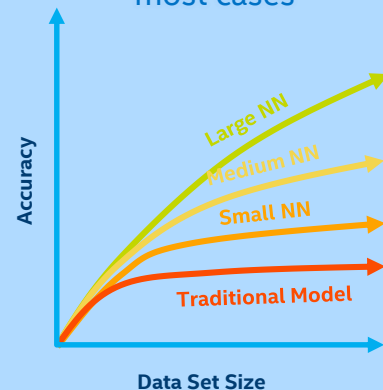


Inference



Did You Know?

Training requires a very large data set and deep neural network (many layers) to achieve the highest accuracy in most cases



INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

Take your computer vision solutions to a new level
with deep learning inference intelligence.

What it is

A toolkit to accelerate development of **high performance computer vision & deep learning into vision applications** from device to cloud. It enables deep learning on hardware accelerators and easy deployment across multiple types of Intel® platforms.

Who needs this product?

- Computer vision/deep learning software developers
- Data scientists
- OEMs, ISVs, System Integrators

Usages

Security surveillance, robotics, retail, healthcare, AI, office automation, transportation, non-vision use cases (speech, text) & more.



HIGH PERFORMANCE, PERFORM AI AT THE EDGE



STREAMLINED & OPTIMIZED DEEP LEARNING INFERENCE



HETEROGENEOUS, CROSS-PLATFORM FLEXIBILITY

Free Download ▶ software.intel.com/openvino-toolkit

Open Source version ▶ 01.org/openvintoolkit

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT



DEEP LEARNING

Caffe TensorFlow ONNX mxnet KALDI

Model
Optimizer

Inference
Engine

Supports 100+ public
models, incl. 30+
pretrained models

COMPUTER VISION



Computer vision library
(kernel & graphic APIs)



Optimized media
encode/decode functions

SUPPORTS MAJOR AI FRAMEWORKS



Rapid adoption by developers

CROSS-PLATFORM FLEXIBILITY



Multiple products launched
based on this toolkit

HIGH PERFORMANCE, HIGH EFFICIENCY



Breadth of product
portfolio

Strong Adoption + Rapidly Expanding Capability

software.intel.com/openvino-toolkit

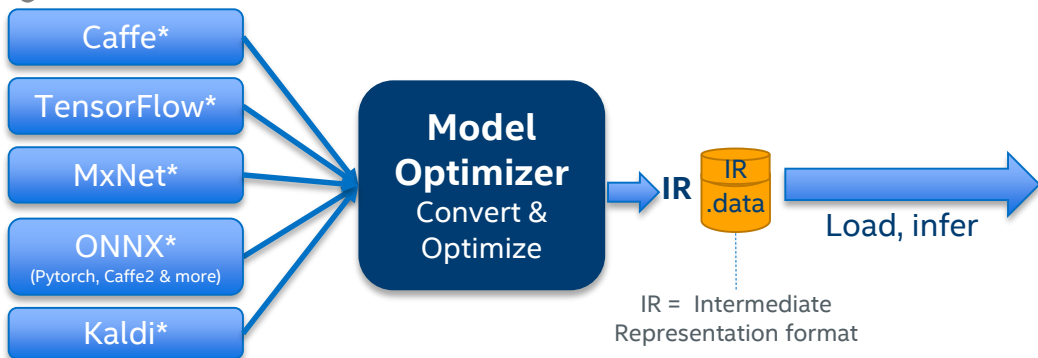
Obtain open source version at 01.org/openvino/toolkit

INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

For Deep Learning Inference – Part of Intel® Distribution of OpenVINO toolkit

Model Optimizer

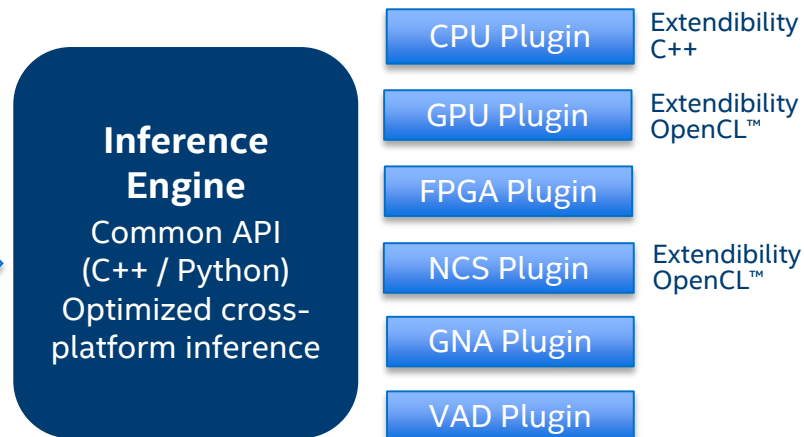
- **What it is:** A Python*-based tool to import trained models and convert them to Intermediate representation.
- **Why important:** Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.



GPU = Intel CPU with integrated graphics processing unit/Intel® Processor Graphics

Inference Engine

- **What it is:** High-level inference API
- **Why important:** Interface is implemented as dynamically loaded plugins for each hardware type. Delivers best performance for each type without requiring users to implement and maintain multiple code pathways.



VAD = Vision Accelerator Design Products; includes FPGA and 8 MyriadX versions

Also available in the [open source version \(01.org/openvino-toolkit\)](https://open-source-version.01.org/openvino-toolkit)

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos



AI SOFTWARE IN THE BIG PICTURE



Architects



Data Engineers



Data Scientists



App Developers



Unified Workflows *And more...*

Collect, Integrate, ETL & ELT

Open *Open (Managed)* *Proprietary*

Manage Metadata

And more...

Deep Learning

And more...

Deploy Inference

And more...

Store & Manage Big Data

Open *Open (Managed)* *Proprietary*

Pre-Process Data

And more...

Machine Learning & Analytics

And more...

Visualize

And more...

IT Systems Management

API's

Enterprise Applications

Library Developers



*Other names and brands may be claimed as the property of others.
Note: displayed logos are not a complete list of solutions/providers in each category, personas are not mutually-exclusive by workflow step, and categorization is generalized*

Quick Guide: What's Inside the Intel Distribution vs Open Source version of OpenVINO™ toolkit

Tool/Component	Intel® Distribution of OpenVINO™ toolkit	OpenVINO™ toolkit (open source)	Open Source Directory https://github.com
Installer (including necessary drivers)	✓		
Intel® Deep Learning Deployment toolkit			
Model Optimizer	✓	✓	/opencv/dldt/tree/2018/model-optimizer
Inference Engine	✓	✓	/opencv/dldt/tree/2018/inference-engine
Intel CPU plug-in	✓ Intel® Math Kernel Library (Intel® MKL) only ¹	✓ BLAS, Intel® MKL ¹ , jit (Intel MKL)	/opencv/dldt/tree/2018/inference-engine
Intel GPU (Intel® Processor Graphics) plug-in	✓	✓	/opencv/dldt/tree/2018/inference-engine
Heterogeneous plug-in	✓	✓	/opencv/dldt/tree/2018/inference-engine
Intel GNA plug-in	✓		
Intel® FPGA plug-in	✓		
Intel® Neural Compute Stick (1 & 2) VPU plug-in	✓		
Intel® Vision Accelerator based on Movidius plug-in	✓		
30+ Pretrained Models - incl. Model Zoo (IR models that run in IE + open sources models)	✓	✓	/opencv/open_model_zoo
Samples (APIs)	✓	✓	/opencv/dldt/tree/2018/inference-engine
Demos	✓	✓	/opencv/open_model_zoo
Traditional Computer Vision			
OpenCV*	✓	✓	/opencv/opencv
OpenVX (with samples)	✓		
Intel® Media SDK	✓	✓ ²	/Intel-Media-SDK/MediaSDK
OpenCL™ Drivers & Runtimes	✓	✓ ²	/intel/compute-runtime
FPGA RunTime Environment, Deep Learning Acceleration & Bitstreams (Linux* only)	✓		

¹Intel MKL is not open source but does provide the best performance

²Refer to readme file for validated versions

Speed Deployment with Pre-trained Models & Samples

Expedite development, accelerate deep learning inference performance, and speed production deployment.

Pretrained Models in Intel® Distribution of OpenVINO™ toolkit

- Age & Gender
- Face Detection—standard & enhanced
- Head Position
- Human Detection—eye-level & high-angle detection
- Detect People, Vehicles & Bikes
- License Plate Detection: small & front facing
- Vehicle Metadata
- Human Pose Estimation
- Action recognition – encoder & decoder
- Text Detection & Recognition
- Vehicle Detection
- Retail Environment
- Pedestrian Detection
- Pedestrian & Vehicle Detection
- Person Attributes Recognition Crossroad
- Emotion Recognition
- Identify Someone from Different Videos—standard & enhanced
- Facial Landmarks
- Gaze estimation
- Identify Roadside objects
- Advanced Roadside Identification
- Person Detection & Action Recognition
- Person Re-identification—ultra small/ultra fast
- Face Re-identification
- Landmarks Regression
- Smart Classroom Use Cases
- Single image Super Resolution (3 models)
- Instance segmentation
- and more...

Binary Models

- Face Detection Binary
- Pedestrian Detection Binary
- Vehicle Detection Binary
- ResNet50 Binary

Save Time with Deep Learning Samples

Use Model Optimizer & Inference Engine for public models & Intel pretrained models

- Object Detection
- Standard & Pipelined Image Classification
- Security Barrier
- Object Detection SSD
- Neural Style Transfer
- Object Detection for Single Shot Multibox Detector using Asynch API+
- Hello Infer Classification
- Interactive Face Detection
- Image Segmentation
- Validation Application
- Multi-channel Face Detection

AGENDA

- Intel® and AI / Machine Learning
- Accelerate Deep Learning Using OpenVINO Toolkit
- **Deep Learning Acceleration with FPGA**
 - **FPGAs and Machine Learning**
 - Intel® FPGA Deep Learning Acceleration Suite
 - Execution on the FPGA (Model Optimizer & Inference Engine)
- Intel® Agilex® FPGA
- OneAPI

DEEP LEARNING ACCELERATION WITH FPGA

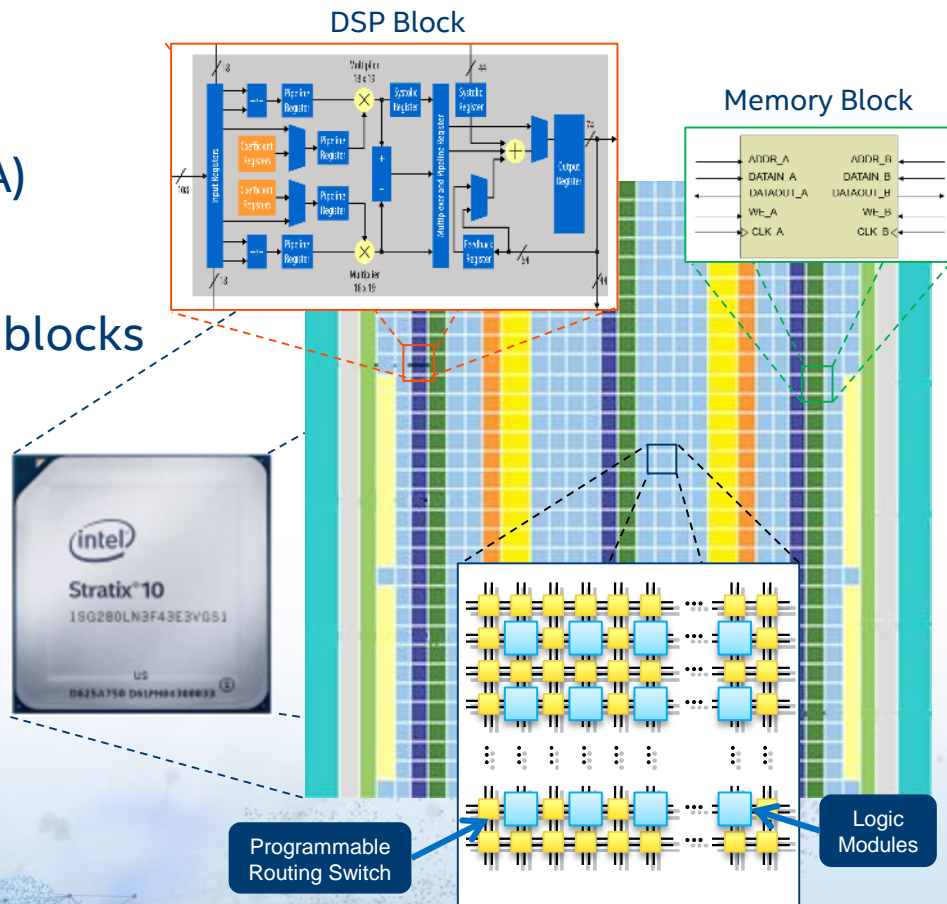
VISION INFERENCE REQUIREMENTS VARY

Design Variable	Considerations
Performance	Frames/sec, latency (real-time constraints)
Power	Power envelope, power per inference
Cost	Hardware cost, development cost, cost per inference
Batch size	1, 8, 32, etc.
Precision	FP32, FP16, FP11
Image size	720p, 1080p, 4K
Camera input	Raw video, encoded
Network support	CNN, RNN, LSTM, custom
Operating environment	Temperature controlled environment, outdoor use, 24/7 operation
Product life	Operational lifespan, product availability
Regulation	Security, privacy

*There is no "one-size-fits-all" solution . . .
. . . and sometimes requirements change*

FPGA OVERVIEW

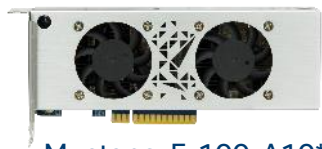
- Field Programmable Gate Array (FPGA)
 - Millions of logic elements
 - Thousands of embedded memory blocks
 - Thousands of DSP blocks
 - Programmable routing
 - High speed transceivers
 - Various built-in hardened IP
- Used to create **Custom Hardware**
- Well-suited for **Matrix Multiplication**



INTEL® FPGA – APPLICATION ACCELERATION FROM EDGE TO CLOUD

Combining Intel FPGA hardware and software to efficiently accelerate workloads for processing-intensive tasks

Programmable Acceleration Card (PAC) offering



Mustang-F-100-A10*



Intel® FPGA PAC
N3000 for networking



Intel® FPGA PAC
with Arria® 10 GX



Intel® FPGA PAC
D5005 for Datacentre

DEVICES / EDGE



NETWORK/NFV



CLOUD/ENTERPRISE



WHY FPGA FOR DEEP LEARNING INFERENCE

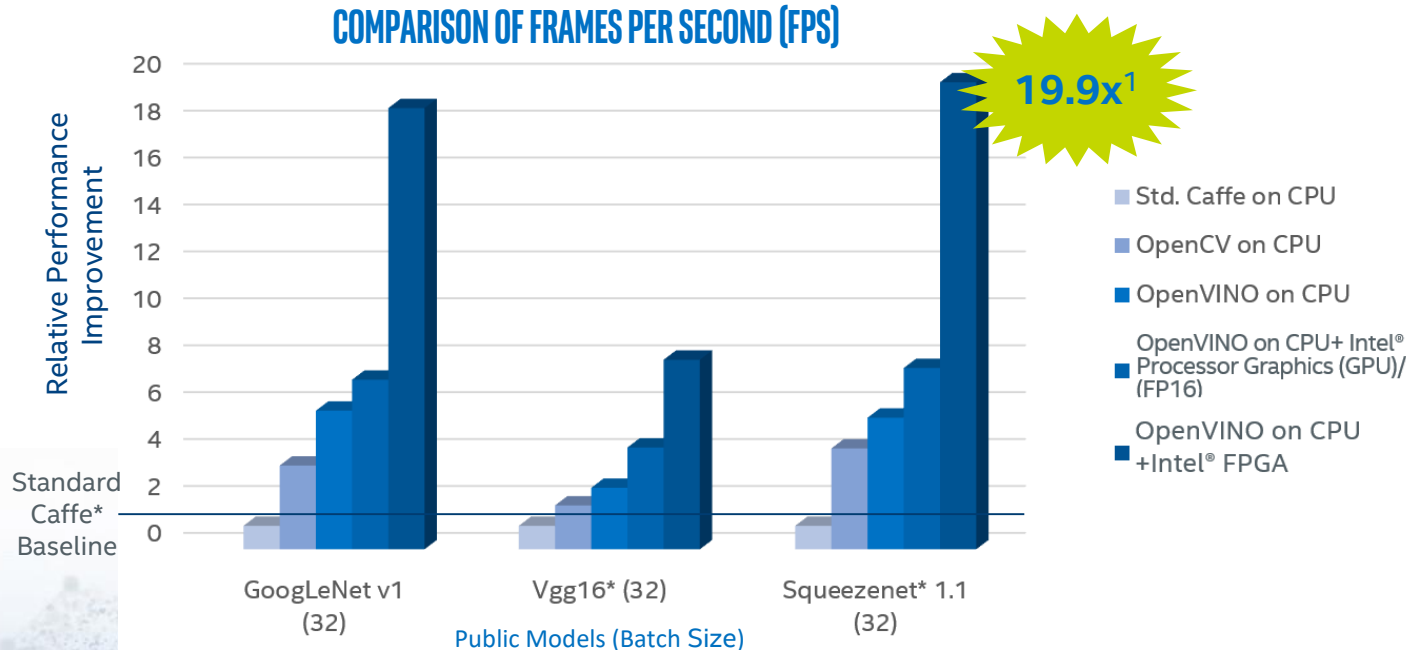
KEY DIFFERENTIATORS

- Low latency
- Low power
- High throughput (FPS)
- Flexibility to adapt to new, evolving, and custom networks
- Supports large image sizes (e.g., 4K)
- Large networks (up to 4 billion parameters)
- Security / Encryption
- Wide ambient temperature range (-40° C to 105° C)*
- 24/7/365 operation
- Long lifespan (8–10 years)

*-40°C to 105°C temp range is for chip down design. Board temp range is 0° to 65°C.



INCREASE DEEP LEARNING WORKLOAD PERFORMANCE ON PUBLIC MODELS USING INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT & INTEL® ARCHITECTURE



Optimize with OpenVINO on CPU, Get an even Bigger Performance Boost with Intel® FPGA

¹Depending on workload, quality/resolution for FP16 may be marginally impacted. A performance/quality tradeoff from FP32 to FP16 can affect accuracy; customers are encouraged to experiment to find what works best for their situation. Performance results are based on testing as of June 13, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. **Configuration:** Testing by Intel as of June 13, 2018. Intel® Core™ i7-6700K CPU @ 2.90GHz fixed, GPU GT2 @ 1.00GHz fixed Internal ONLY testing, Test v3.15.21 – Ubuntu* 16.04, OpenVINO 2018 RC4, Intel® Arria® 10 FPGA 1150GX. Tests were based on various parameters such as model used (these are public), batch size, and other factors. Different models can be accelerated with different Intel hardware solutions, yet use the same Intel software tools.

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



Solving Machine Learning Challenges with FPGA



EASE-OF-USE

SOFTWARE ABSTRACTION,
PLATFORMS & LIBRARIES

Intel FPGA solutions enable software-defined programming of customized machine learning accelerator libraries.



REAL-TIME

DETERMINISTIC
LOW LATENCY

Intel FPGA hardware implements a deterministic low latency data path unlike any other competing compute device.

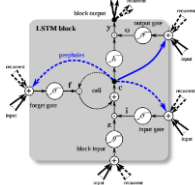


FLEXIBILITY

CUSTOMIZABLE HARDWARE
FOR NEXT GEN DNN ARCHITECTURES

Intel FPGAs can be customized to enable advances in machine learning algorithms.

Public Intel FPGA Machine Learning Success



Microsoft: “Microsoft has revealed that Altera FPGAs have been installed across every Azure cloud server, creating what the company is calling “the world’s first AI supercomputer.” -Sept 2016

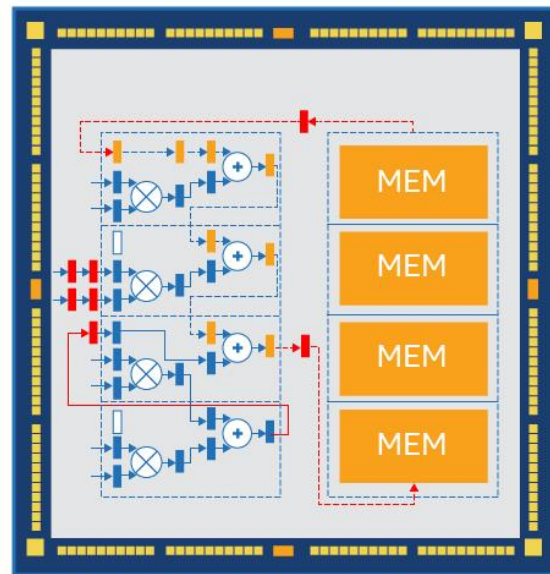
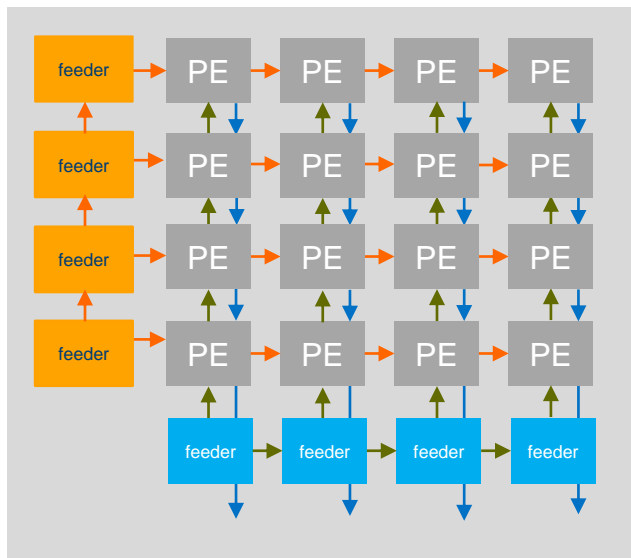
NEC: “To create the NeoFace Accelerator, the engine software IP was integrated into an Intel Arria 10 FPGA, which operate in Xeon processor–based servers.” -June 2017

JD.COM: “Arria® 10 FPGA can achieve 5x improvement in the performance of LSTM accelerator card compared to GPU.” –Apr 2017

Inspur/iFlytech: “Leading server vendor Inspur Group and Altera today launched a speech recognition acceleration solution based on Altera's Arria® 10 FPGAs and DNN algorithm from iFLYTEK.” -Dec 2016

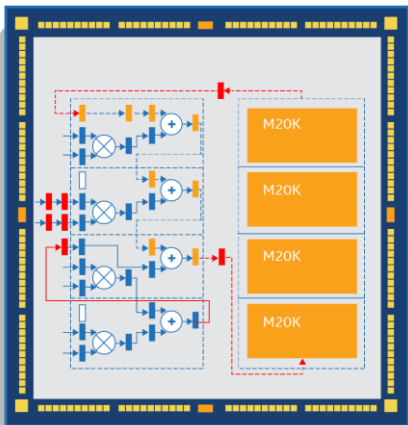
ZTE: “Using Intel's Arria 10 FPGA, ZTE engineers were able to achieve more than 1000 images per second in facial recognition.” –Jan 2017

Why FPGAs for Deep Learning?

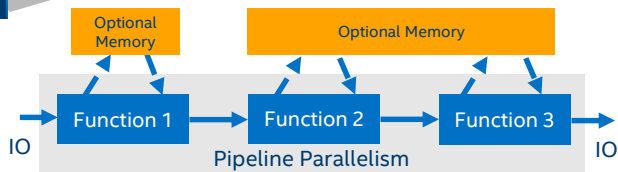


Customized Performance

Convolutional Neural Networks are Compute Intensive



Fine-grained & low latency
between compute and memory

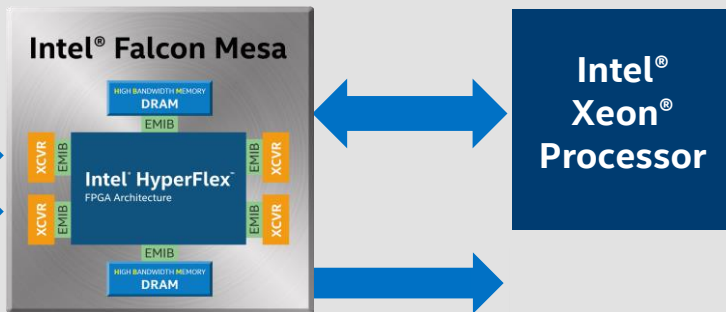


Feature	Benefit
Highly parallel architecture	Facilitates efficient low-batch video stream processing and reduces latency
Configurable Distributed Floating Point DSP Blocks	FP32 9Tflops, FP16, FP11 Accelerates computation by tuning compute performance
Tightly coupled high-bandwidth memory	>50TB/s on chip SRAM bandwidth, random access, reduces latency, minimizes external memory access
Programmable Data Path	Reduces unnecessary data movement, improving latency and efficiency
Configurability	Support for variable precision (trade-off throughput and accuracy). Future proof designs, and system connectivity

FPGAs System Flexibility to Control the Data path

Compute Acceleration/Offload

- Workload agnostic compute
- FPGAaaS
- Virtualization



Inline Data Flow Processing

- Machine learning
- Object detection and recognition
- Advanced driver assistance system (ADAS)
- Gesture recognition
- Face detection

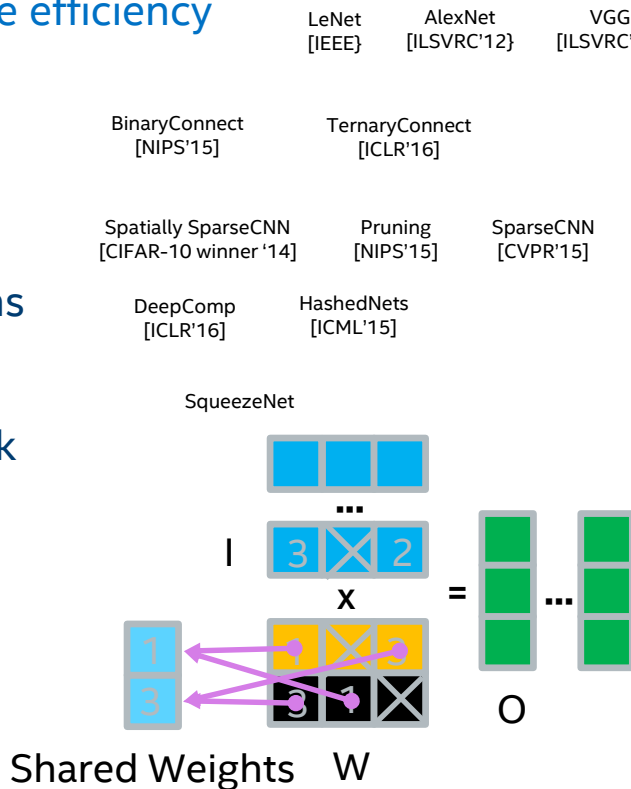
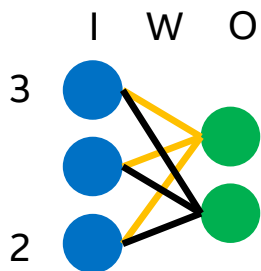
Storage Acceleration

- Machine learning
- Cryptography
- Compression
- Indexing

FPGA Flexibility Supports Arbitrary Architectures

Many efforts to improve efficiency

- Batching
- Reduce bit width
- Sparse weights
- Sparse activations
- Weight sharing
- Compact network



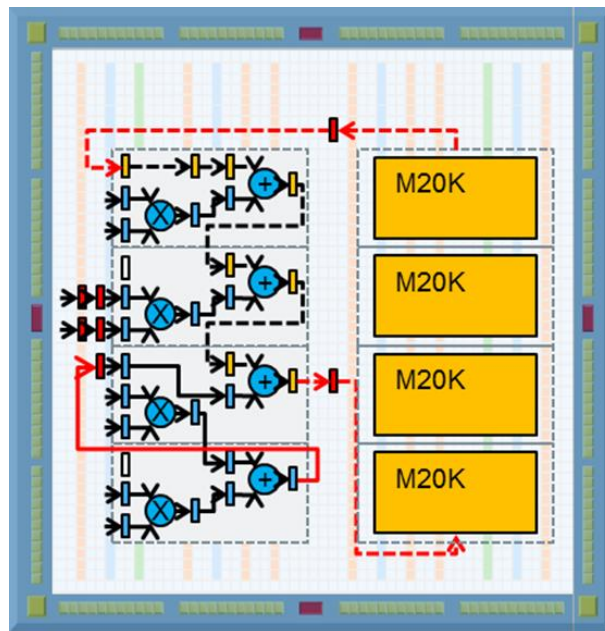
LeNet [IEEE] AlexNet [ILSVRC'12] VGG [ILSVRC'14] GoogleNet [ILSVRC'14] ResNet [ILSVRC'15] XNORNet

BinaryConnect [NIPS'15] TernaryConnect [ICLR'16]

Spatially SparseCNN [CIFAR-10 winner '14] Pruning [NIPS'15] SparseCNN [CVPR'15]

DeepComp [ICLR'16] HashedNets [ICML'15]

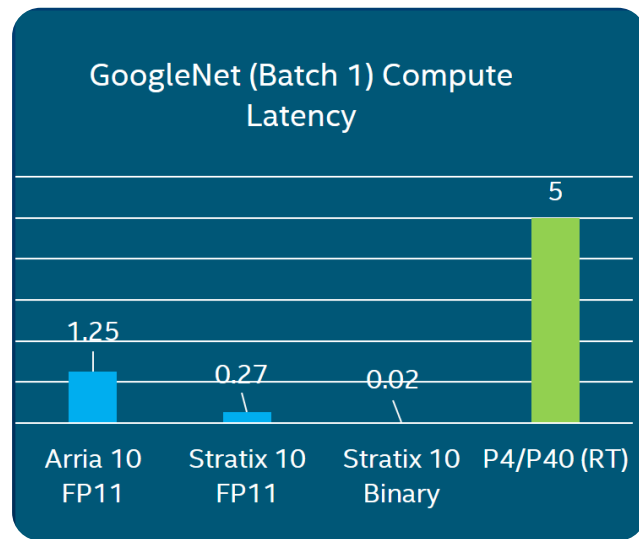
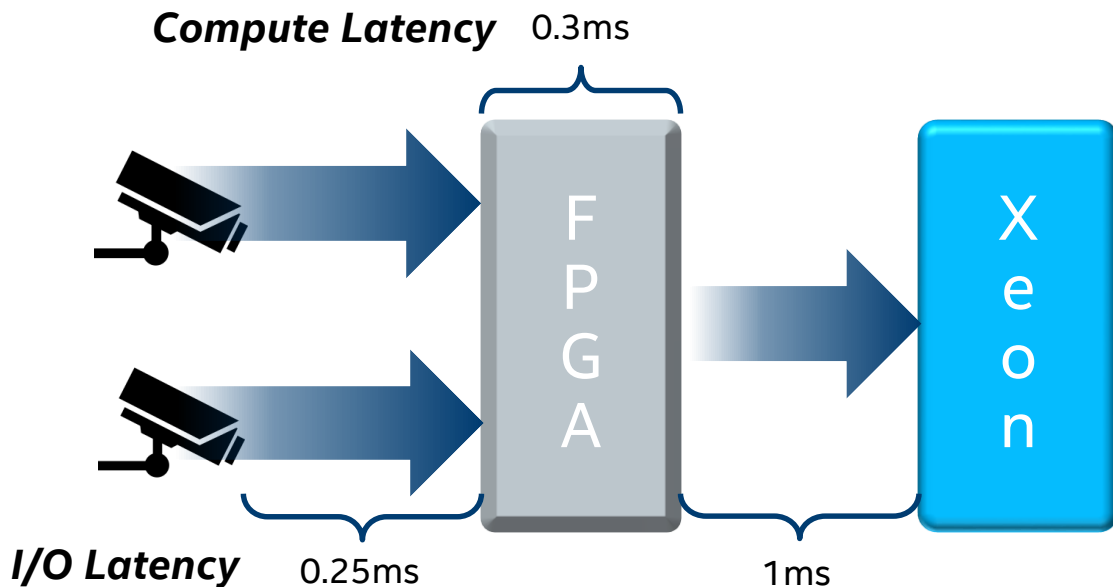
SqueezeNet



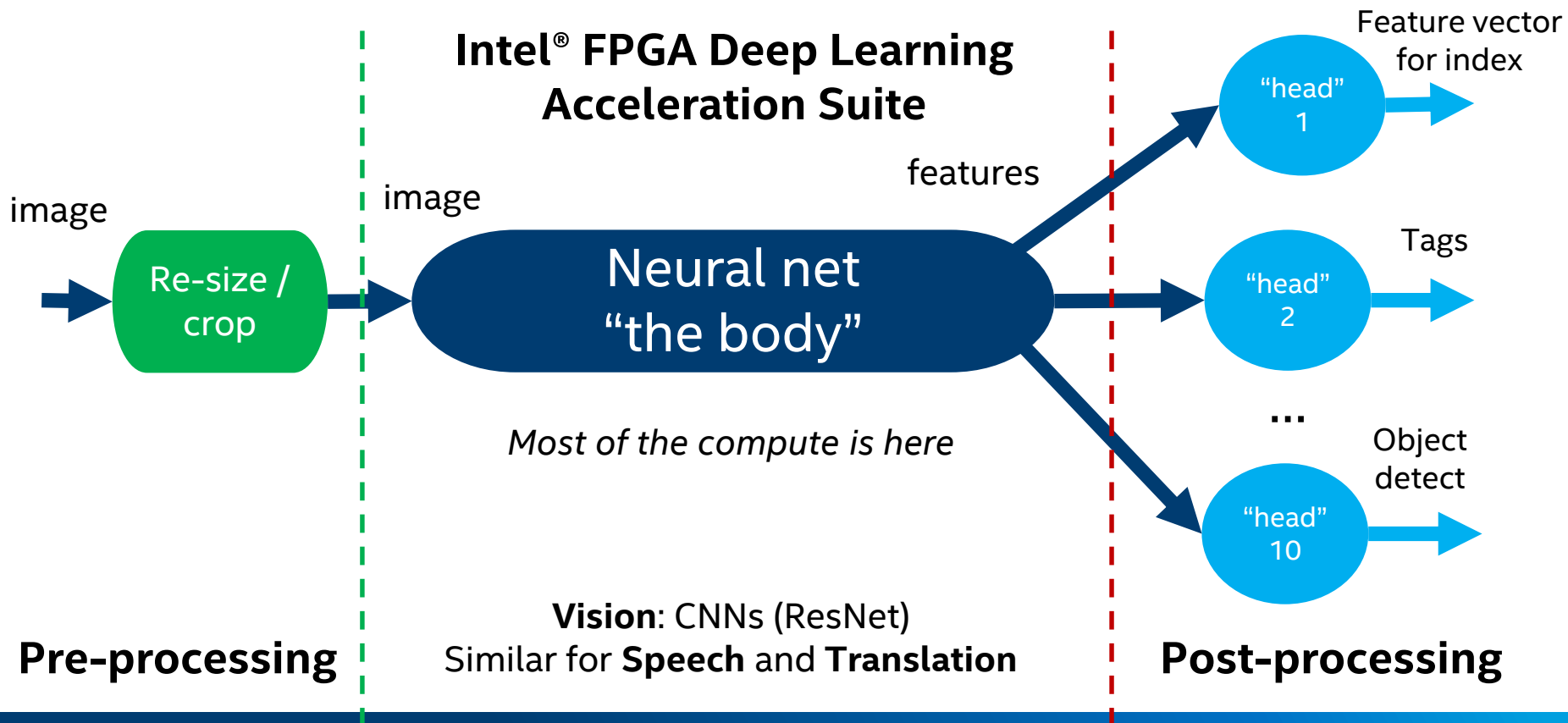
FPGAs Provide Deterministic System Latency

FPGAs can leveraging parallelism across the entire chip reducing the compute time to a fraction

$$\text{System Latency} = \text{I/O Latency} + \text{Compute Latency}$$



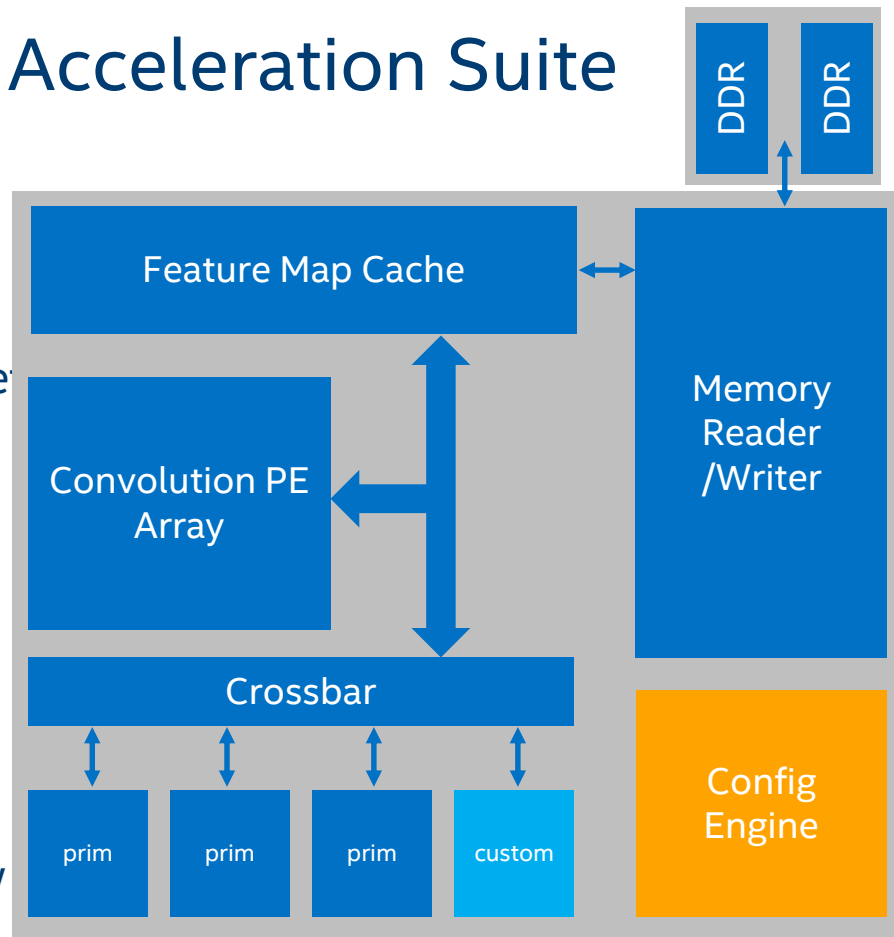
Deep Learning Topology Inference Processing



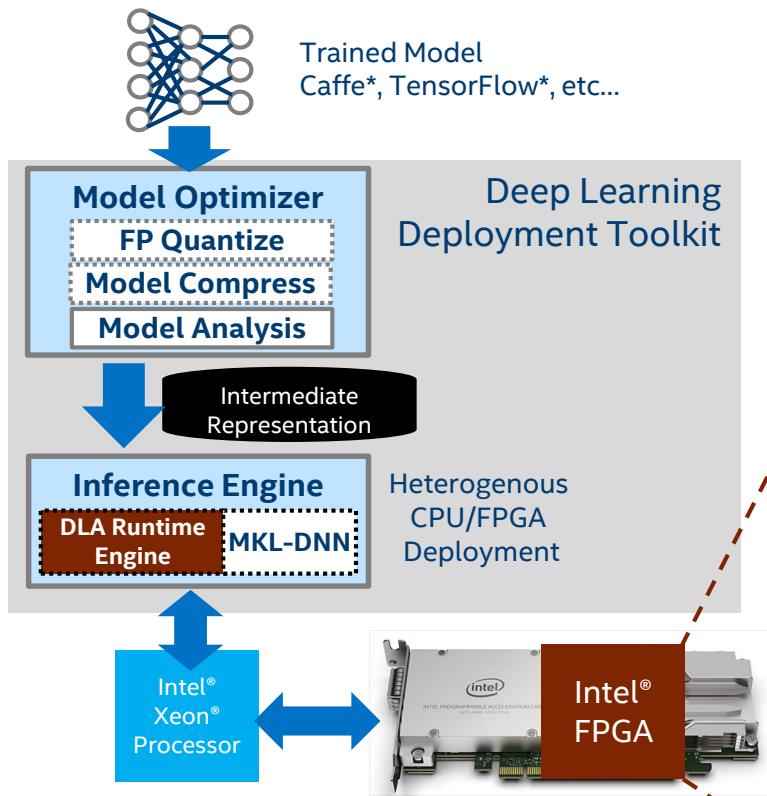
Intel® FPGA Deep Learning Acceleration Suite

Features

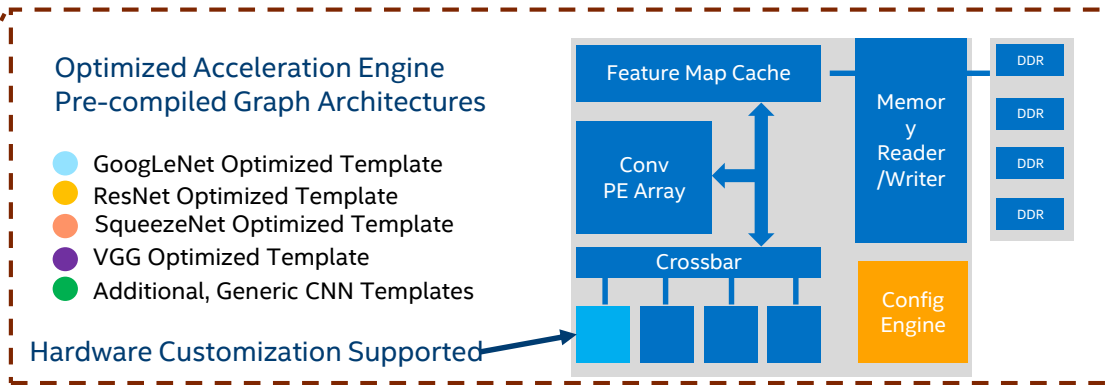
- CNN acceleration engine for common topologies executed in a graph loop architecture
 - AlexNet, GoogleNet, LeNet, SqueezeNet, VGG16, ResNet, Yolo, SSD...
- Software Deployment
 - No FPGA compile required
 - Run-time reconfigurable
- Customized Hardware Development
 - Custom architecture creation w/ parameters
 - Custom primitives using OpenCL™ flow



FPGA Usage with Intel® Distribution of OpenVINO™ toolkit

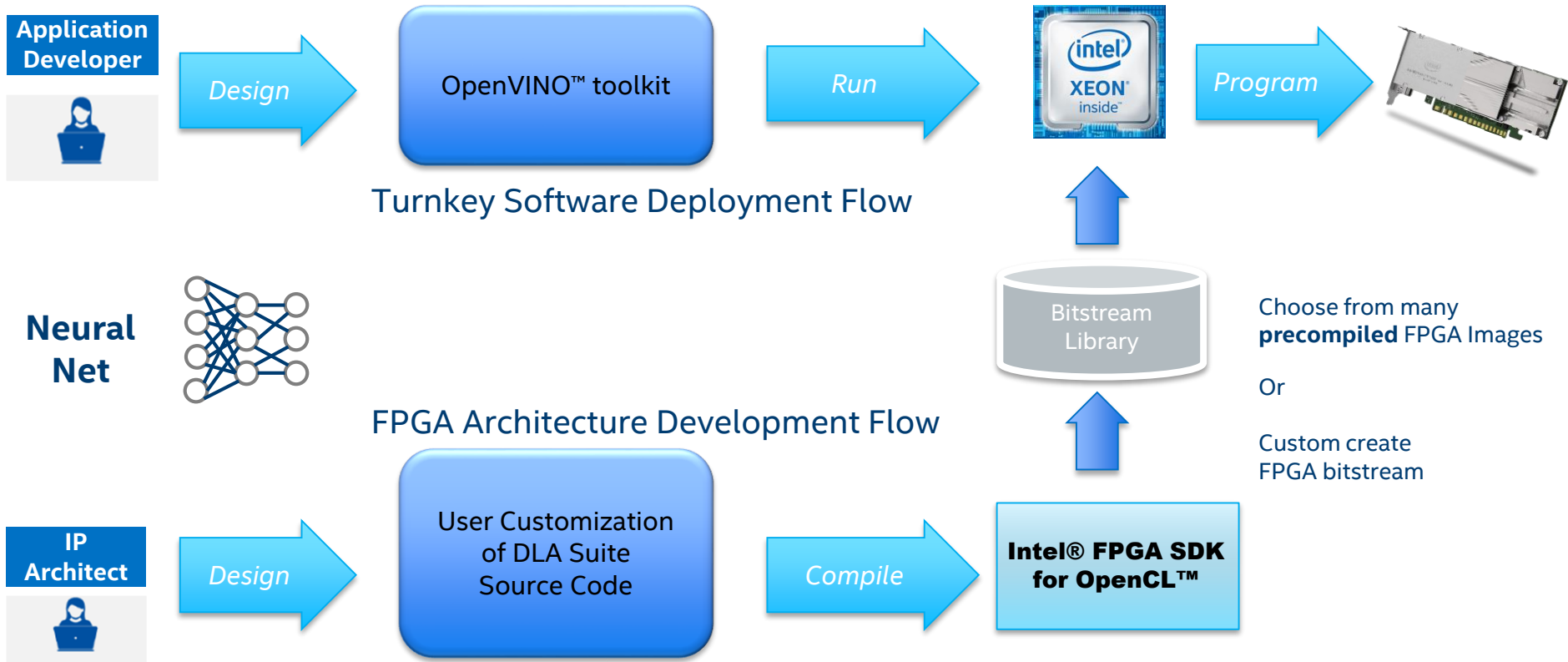


- Supports common software frameworks (Caffe*, TensorFlow*)
- Model Optimizer enhances model for improved execution, storage, and transmission
- Inference Engine optimizes inference execution across Intel® hardware solutions using unified deployment API
- Intel® FPGA DLA Suite provides turn-key or customized CNN acceleration for common topologies



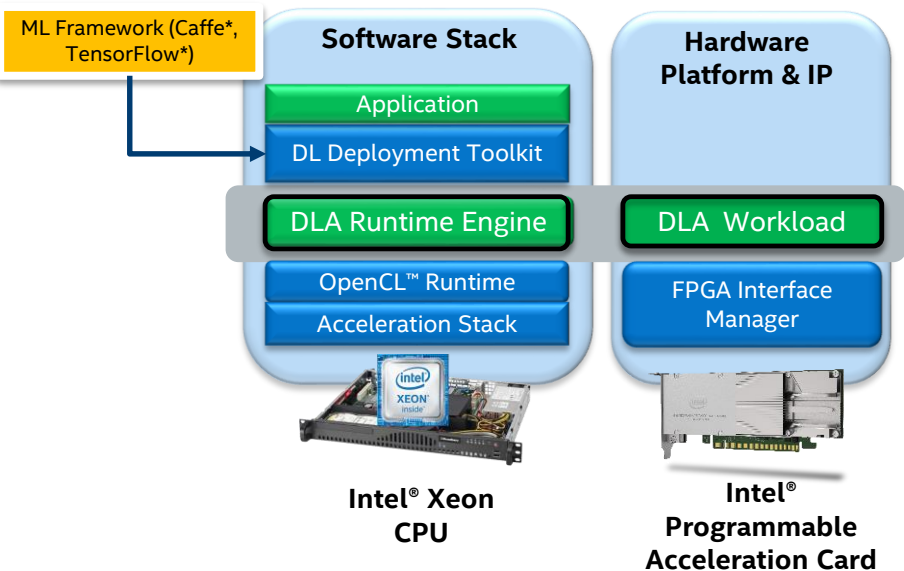
*Other names and brands may be claimed as the property of others

OpenVINO™ with DLA User Flows



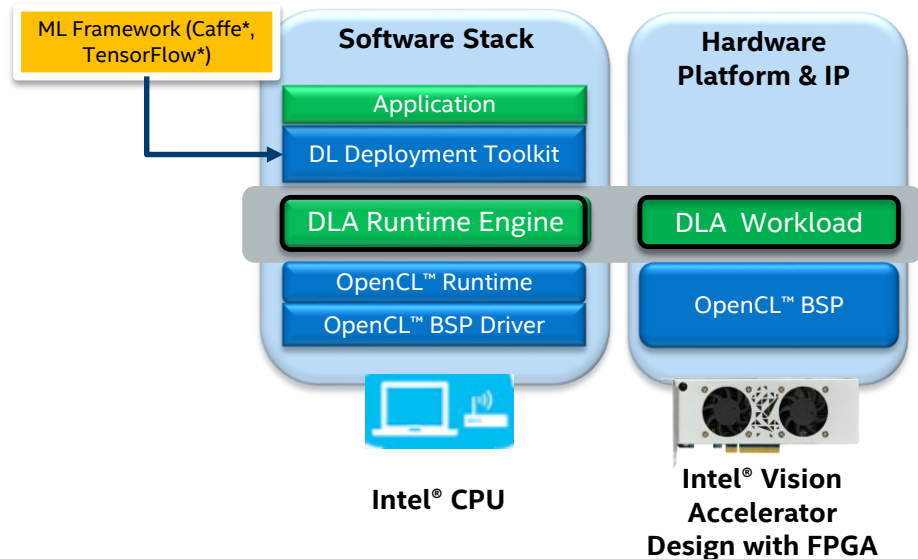
Machine Learning on Intel® FPGA Platform

Acceleration Stack Platform Solution



[Intel® FPGA Acceleration Hub](#)

Edge Computing Solution

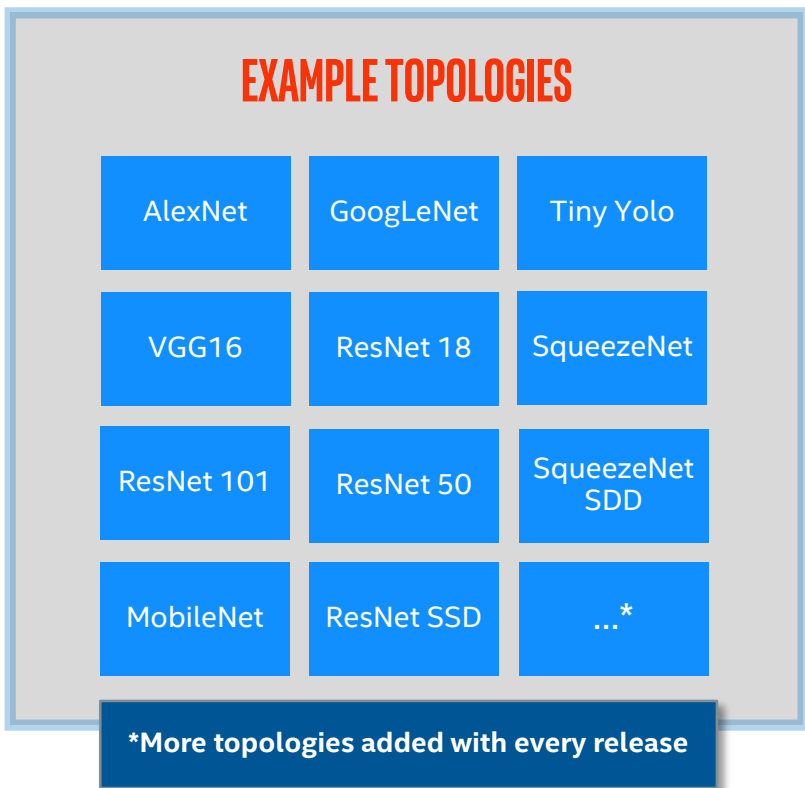
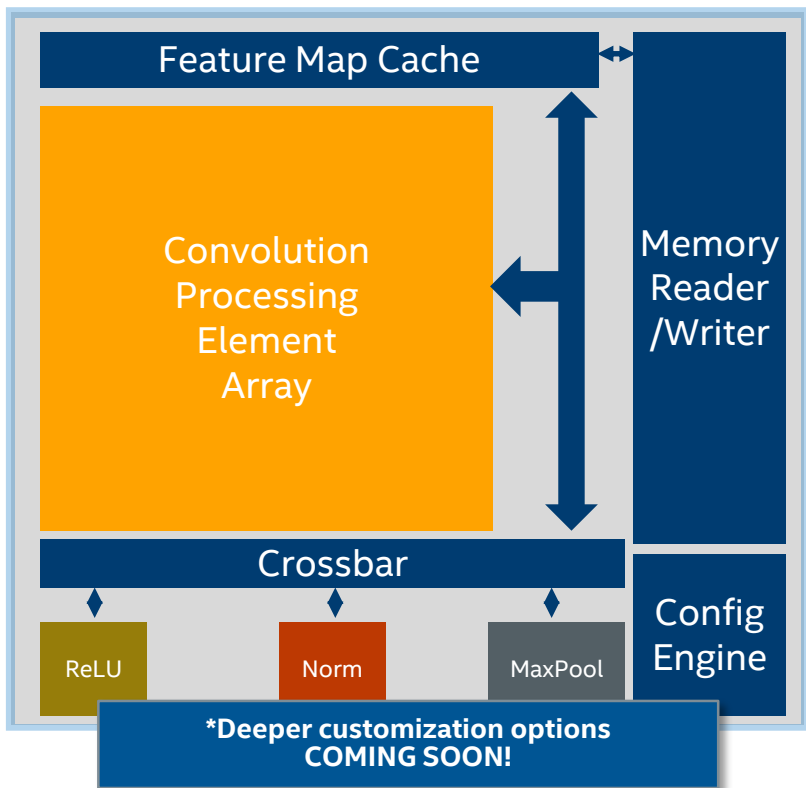


[Intel® Vision Accelerator Design Products](#)

AGENDA

- Intel® and AI / Machine Learning
- Accelerate Deep Learning Using OpenVINO Toolkit
- **Deep Learning Acceleration with FPGA**
 - FPGAs and Machine Learning
 - **Intel® FPGA Deep Learning Acceleration Suite**
 - Execution on the FPGA (Model Optimizer & Inference Engine)
- Intel® Agilex® FPGA
- OneAPI

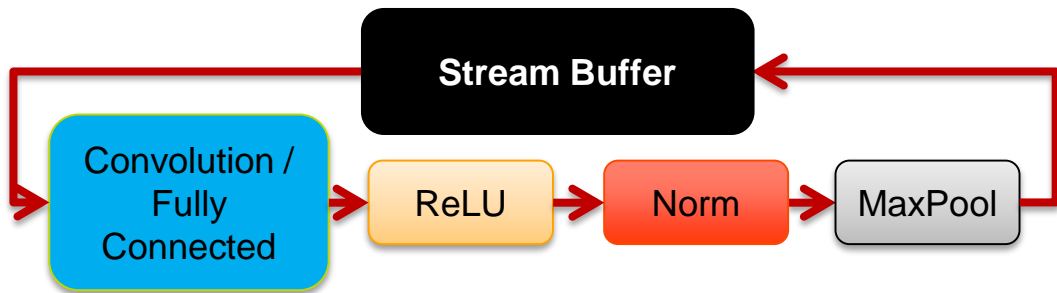
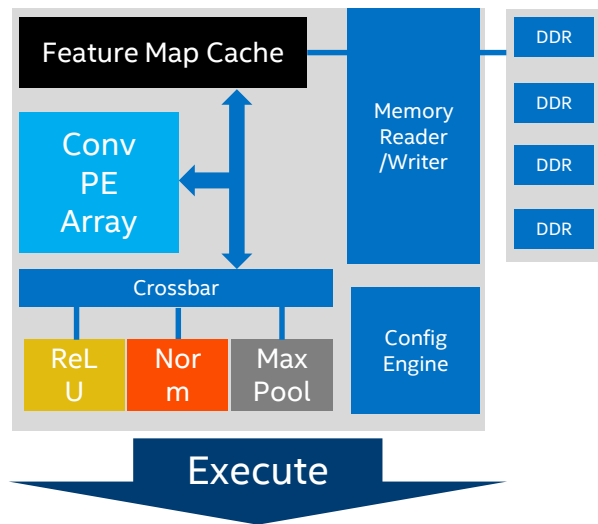
INTEL® FPGA DEEP LEARNING ACCELERATION (DLA FOR OPENVINO)



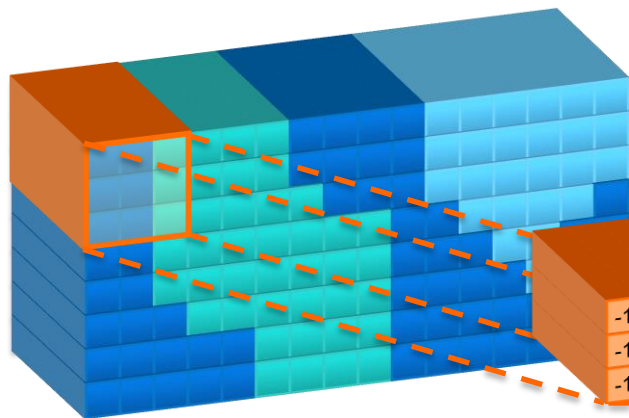
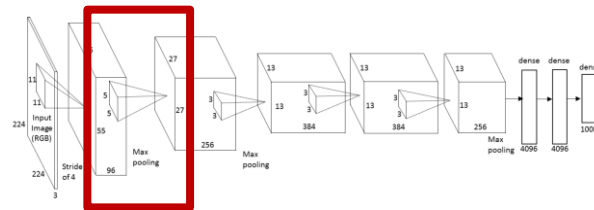
* Other names and brands may be claimed as the property of others.

DLA Architecture: Built for Performance

- Maximize Parallelism on the FPGA
 - Filter Parallelism (Processing Elements)
 - Input-Depth Parallelism
 - Winograd Transformation
 - Batching
 - Feature Stream Buffer
 - Filter Cache
- Choosing FPGA Bitstream
 - Data Type / Design Exploration
 - Primitive Support



CNN Computation in One Slide



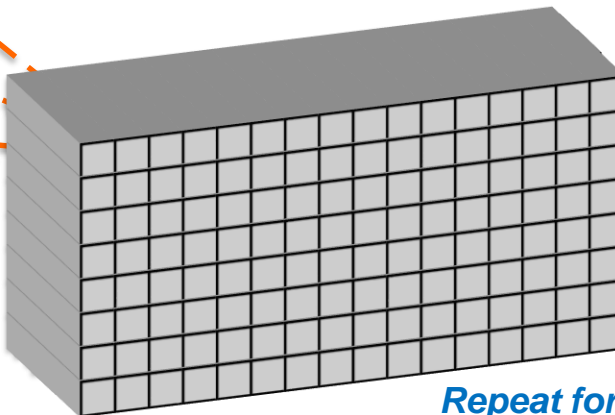
Input Feature Map
(Set of 2D Images)

Filter
(3D Space)

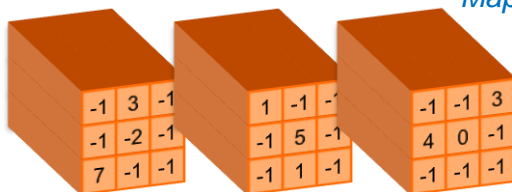
-1	-1	-1
-1	9	-1
-1	-1	-1

$$I_{\text{new}}[x][y] = \sum_{x'=-1}^1 \sum_{y'=-1}^1 I_{\text{old}}[x+x'][y+y'] \times F[x'][y']$$

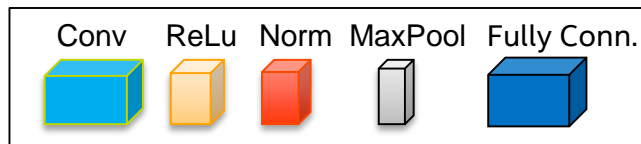
Output Feature Map



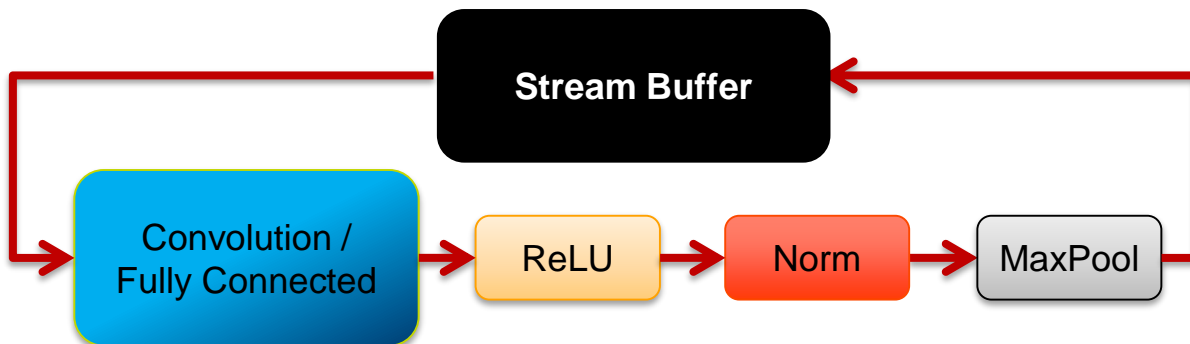
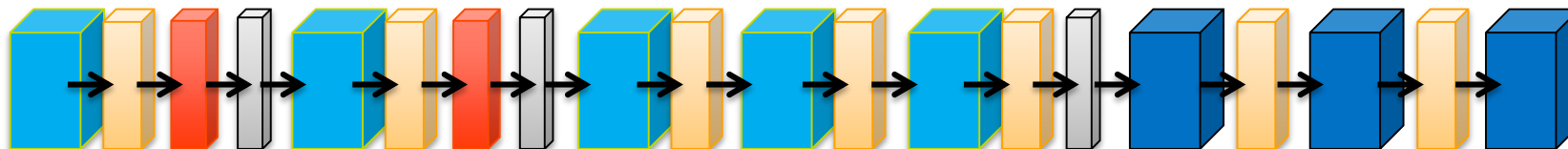
**Repeat for Multiple Filters
to Create Multiple "Layers"
of Output Feature Map**



Mapping Graphs in DLA

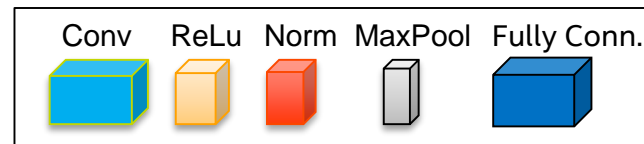


AlexNet Graph

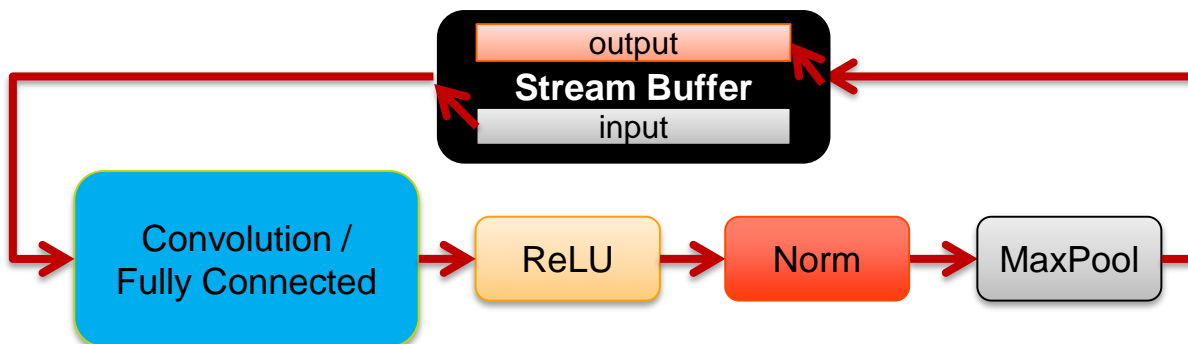
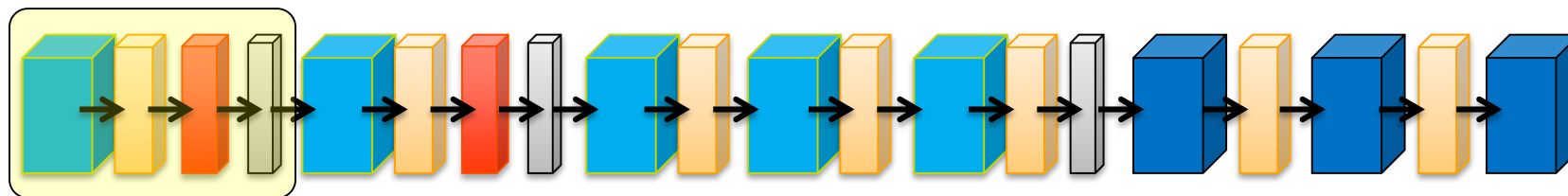


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

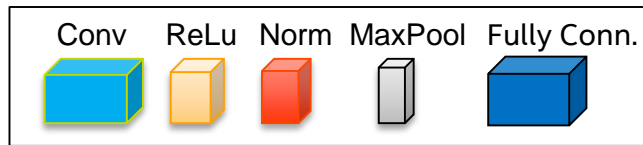


AlexNet Graph

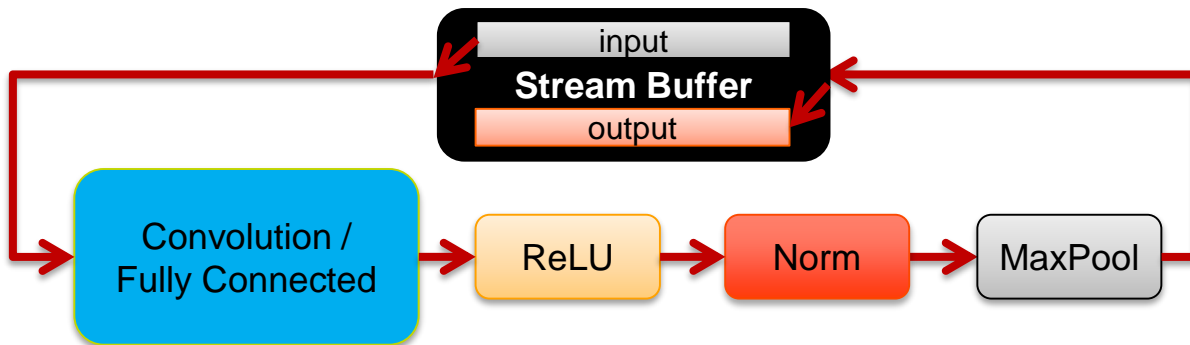
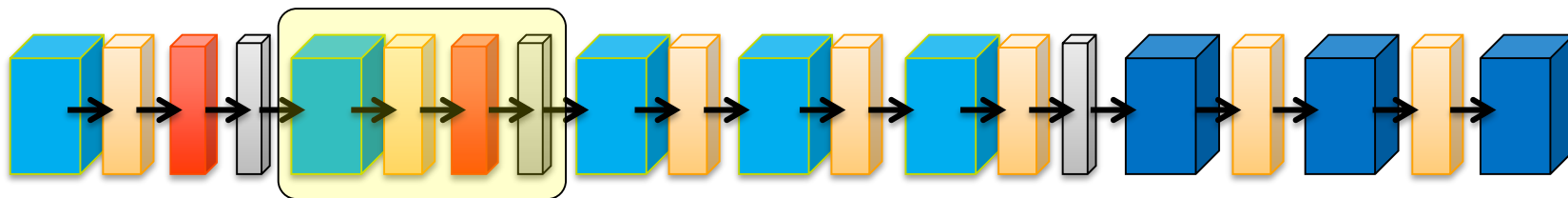


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

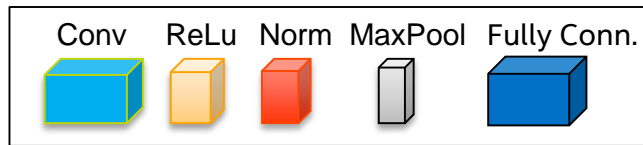


AlexNet Graph

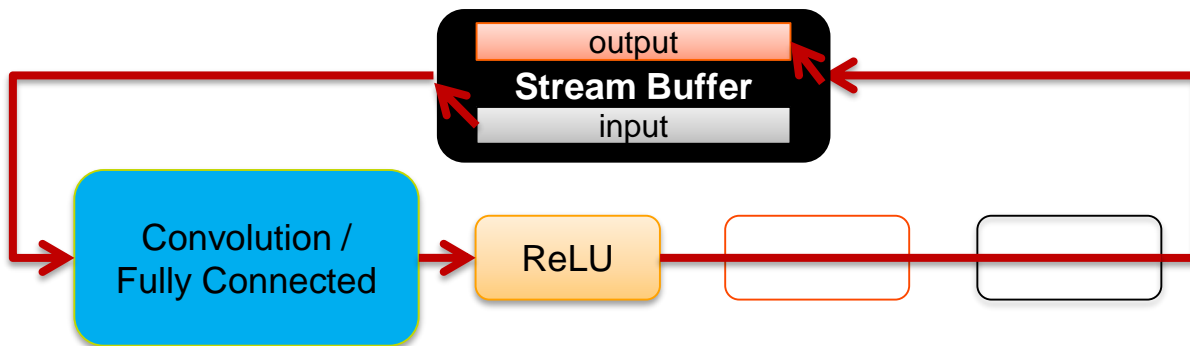
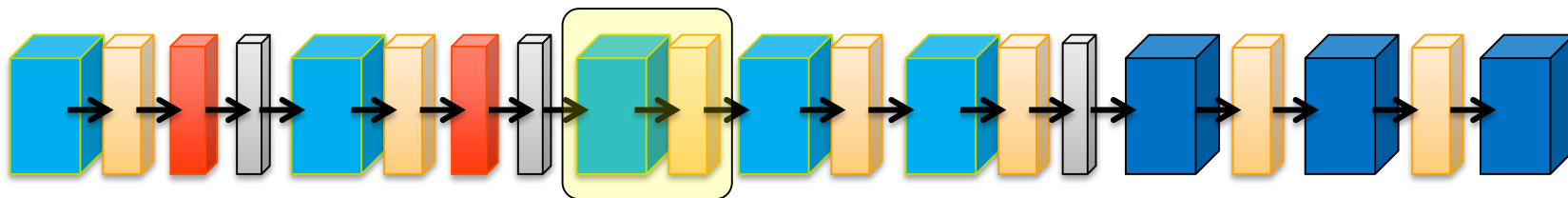


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

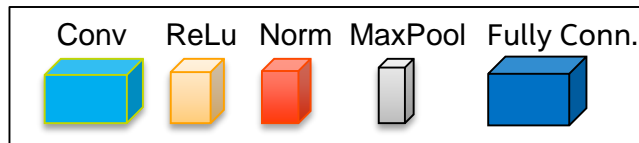


AlexNet Graph

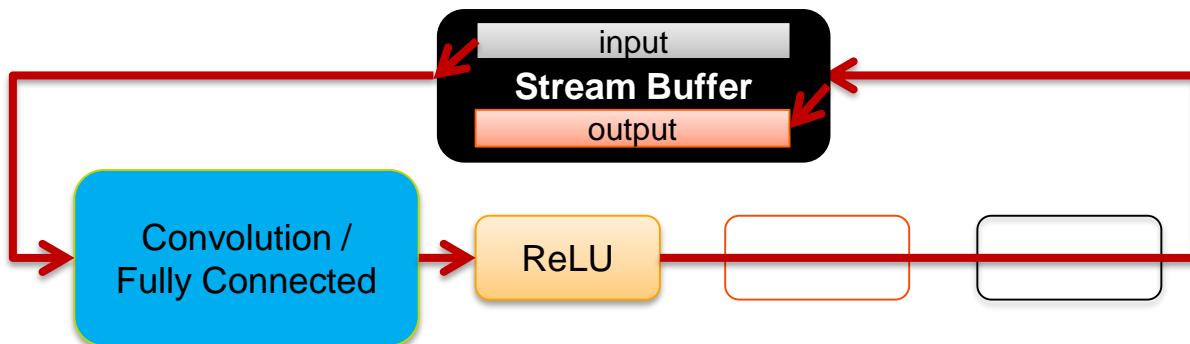
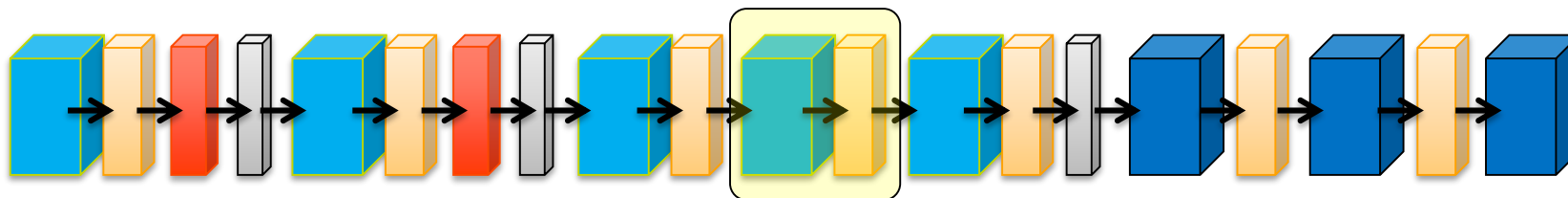


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

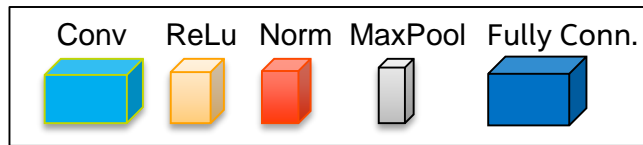


AlexNet Graph

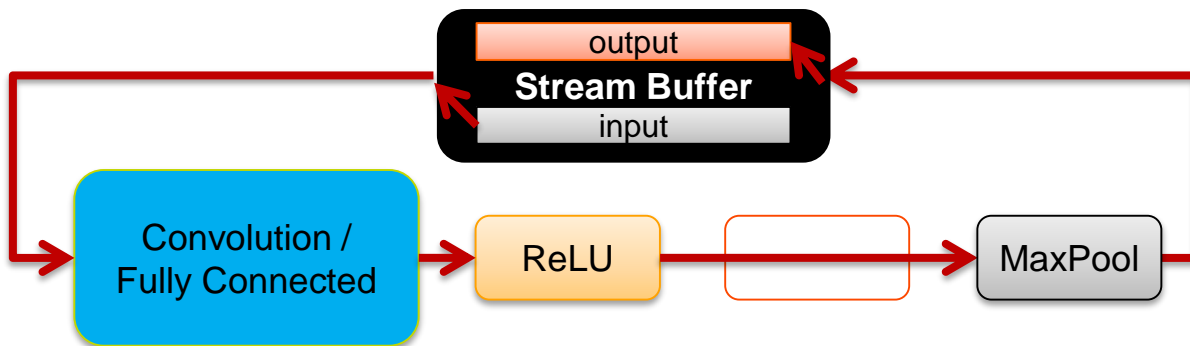
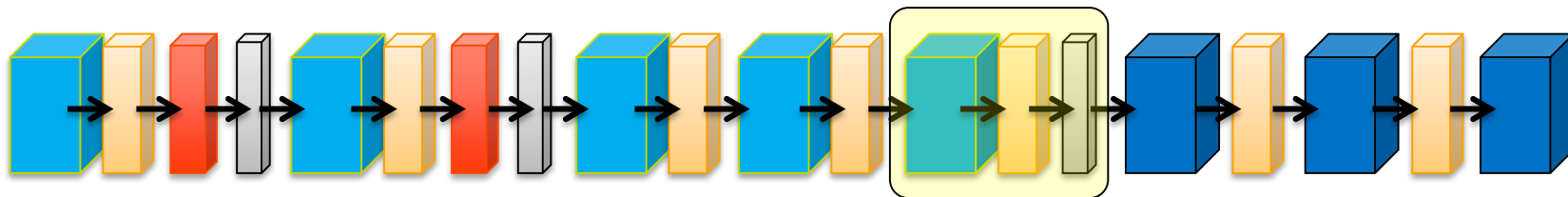


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

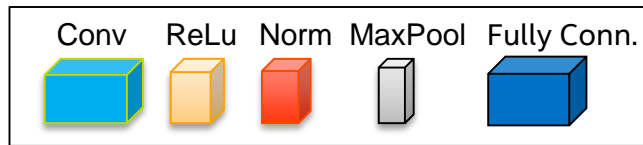


AlexNet Graph

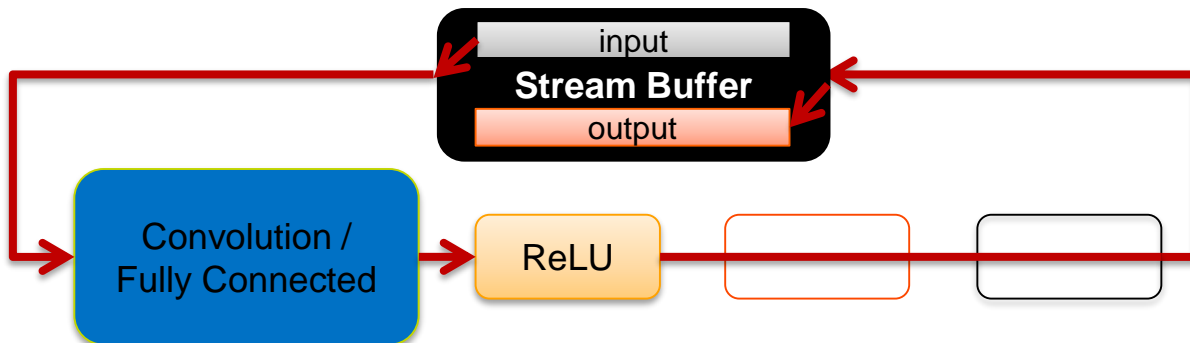
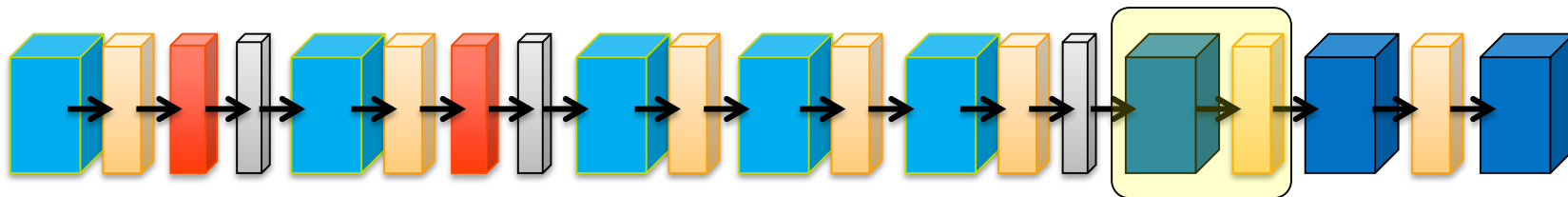


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

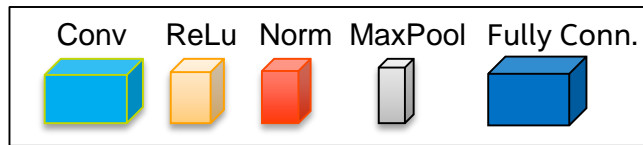


AlexNet Graph

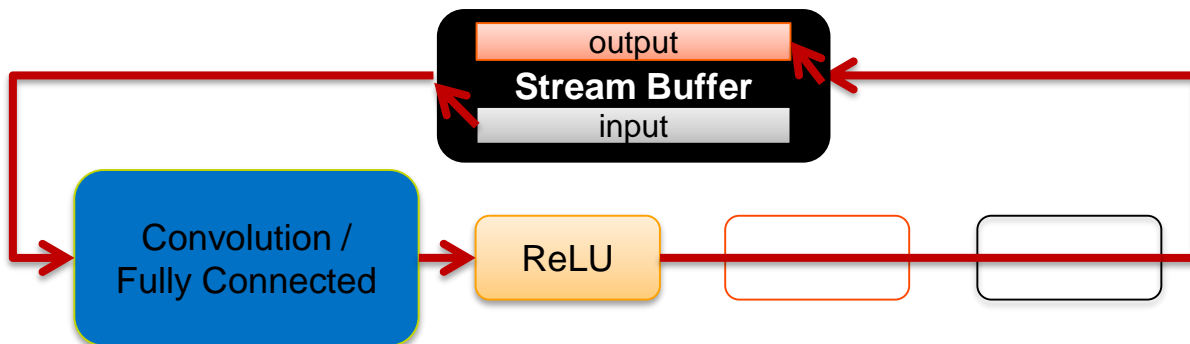
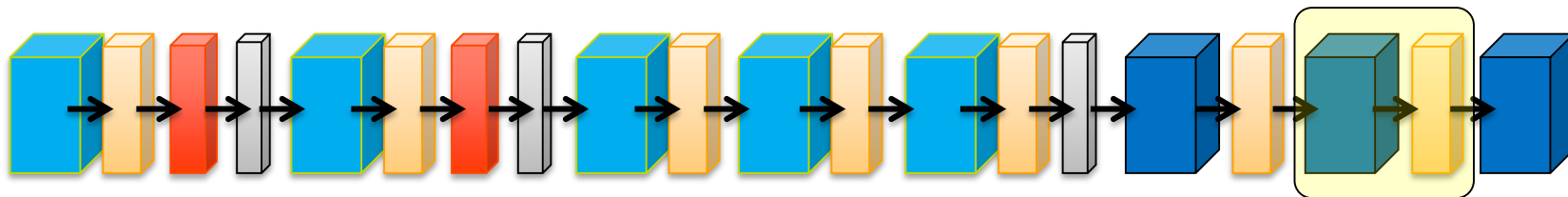


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

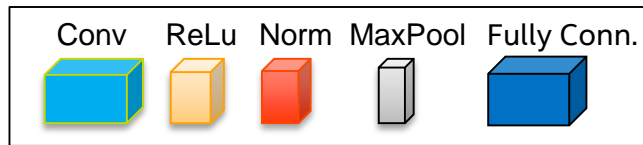


AlexNet Graph

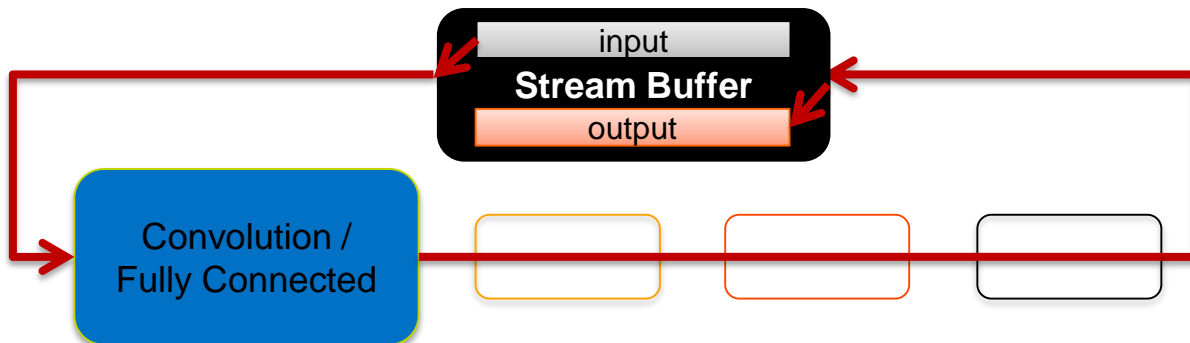
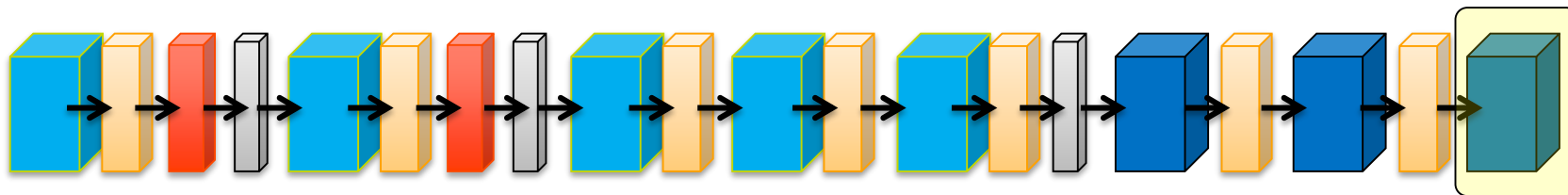


Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA

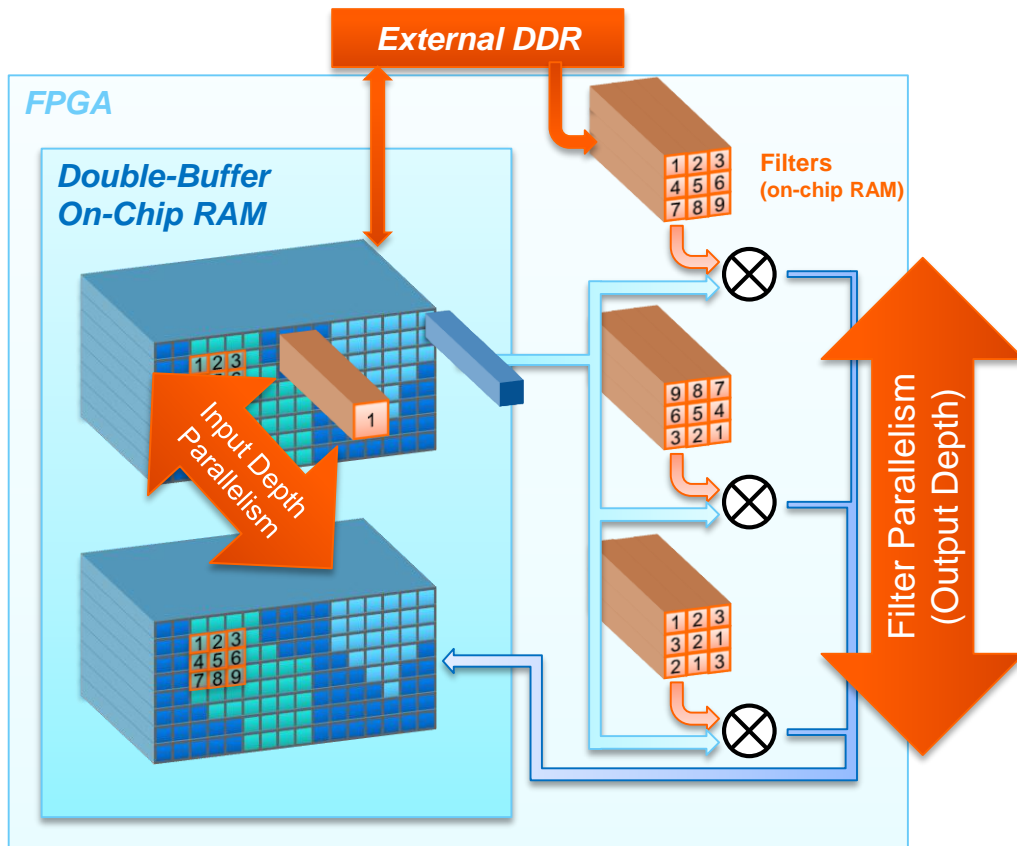


AlexNet Graph



Blocks are run-time reconfigurable and bypassable

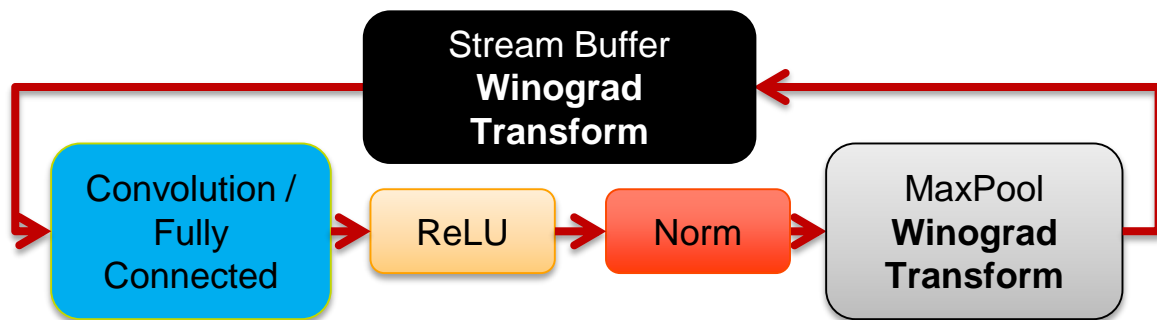
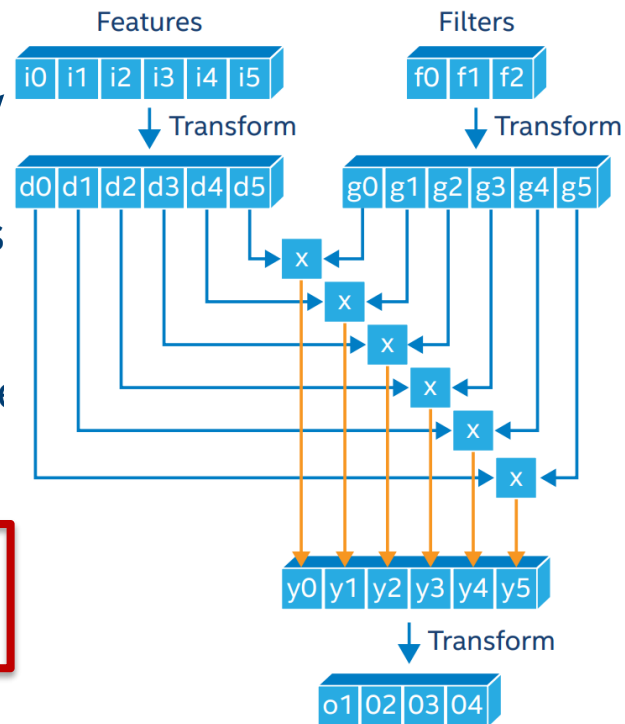
Efficient Parallel Execution of Convolutions



- Parallel Convolutions
 - Different filters of the same convolution layer processed in parallel in different processing elements (PEs)
- Vectored Operations
 - Across the depth of feature map
- PE Array geometry can be customized to hyperparameters of given topology

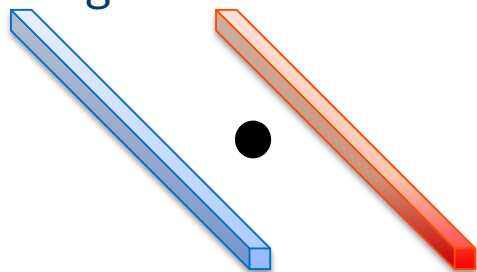
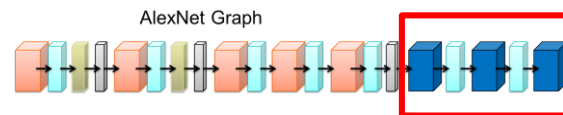
Winograd Transformation

- Perform convolutions with fewer multiplication
 - Allows more convolutions to be done on FPGA
- Take 6 input features elements and 3 filter elements
 - Standard convolution requires 12 multiplies
 - Transformed convolution requires just 6 multiplies

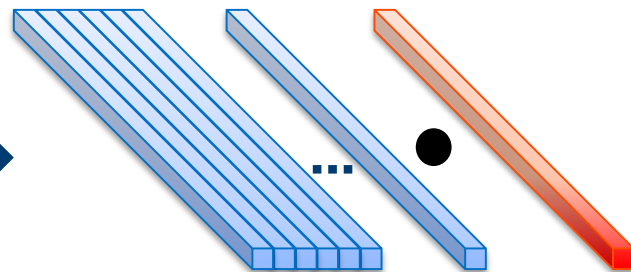


Fully Connected Computation and Batching

- Fully Connected Layer computation does not allow for data reuse of weights
 - Different from convolutions
 - Very memory bandwidth intensive
- Solution: Batch up images
 - Weights reused across multiple images



$$O = I_{vec} * W_{vec}$$

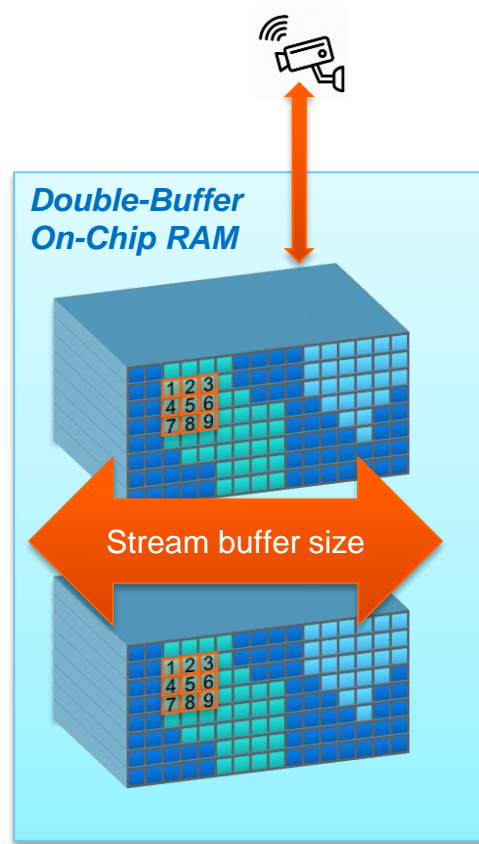


$$O_{vec} = I_{mat} * W_{vec}$$

Feature Cache

Feature data cached on-chip

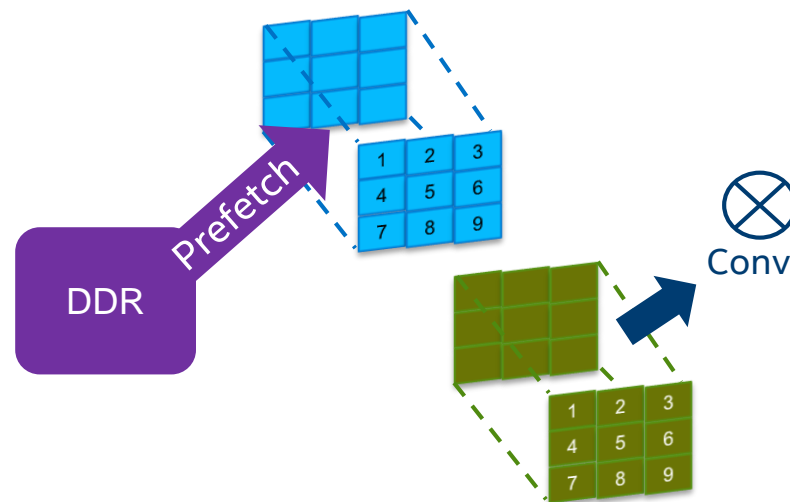
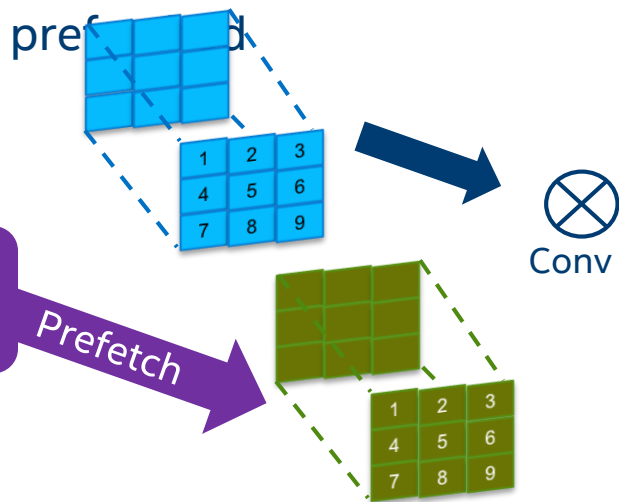
- Streamed to a daisy chain of parallel processing elements
- Double buffered
 - Overlap convolution with cache updates
 - Output of one subgraph becomes input of another
 - Eliminates unnecessary external memory accesses



Filter Cache

Filter weights cached in each processing element

- Double buffered in order to support prefetching
 - While one set is used to calculate output feature maps, another set is



Design Exploration with Reduced Precision

Tradeoff between performance and accuracy

- Reduced precision allows more processing to be done in parallel
- Using smaller Floating Point format does not require retraining of network
- FP11 benefit over using INT8/9
 - No need to retrain, better performance, less accuracy loss

FP16 

Sign, 5-bit exponent, 10-bit mantissa

FP11 

Sign, 5-bit exponent, 5-bit mantissa

FP10 

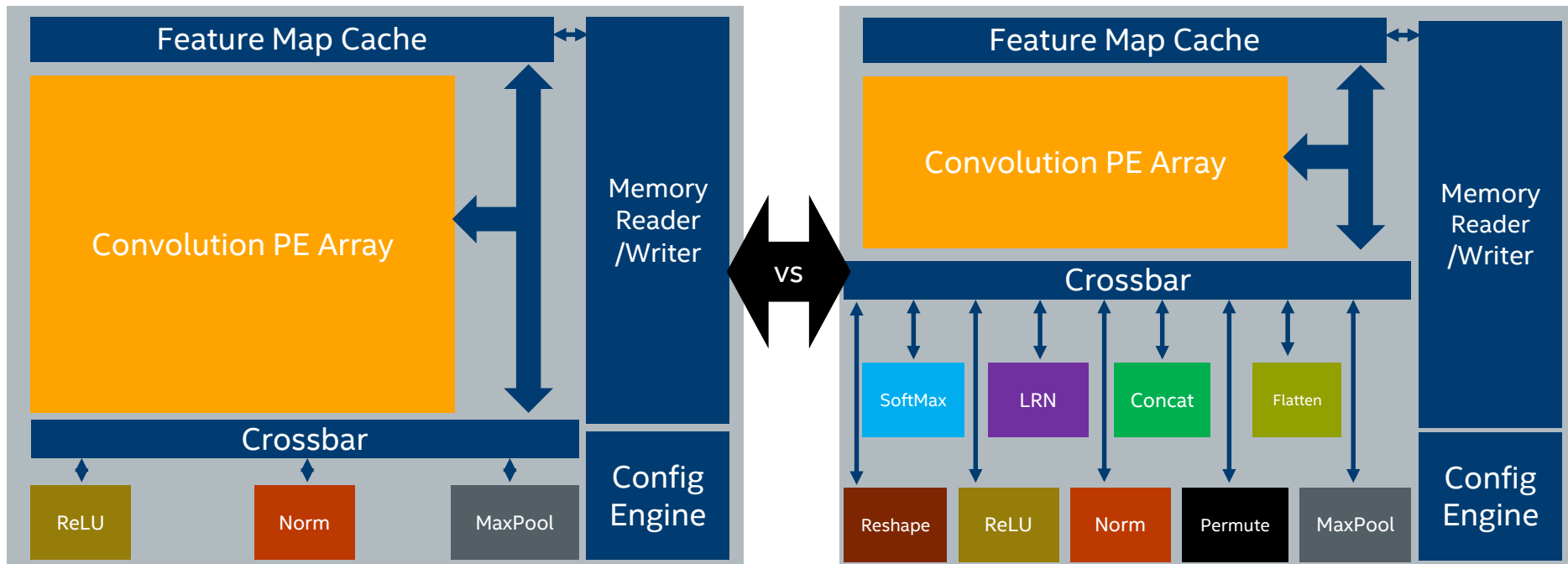
Sign, 5-bit exponent, 4-bit mantissa

FP9 

Sign, 5-bit exponent, 3-bit mantissa

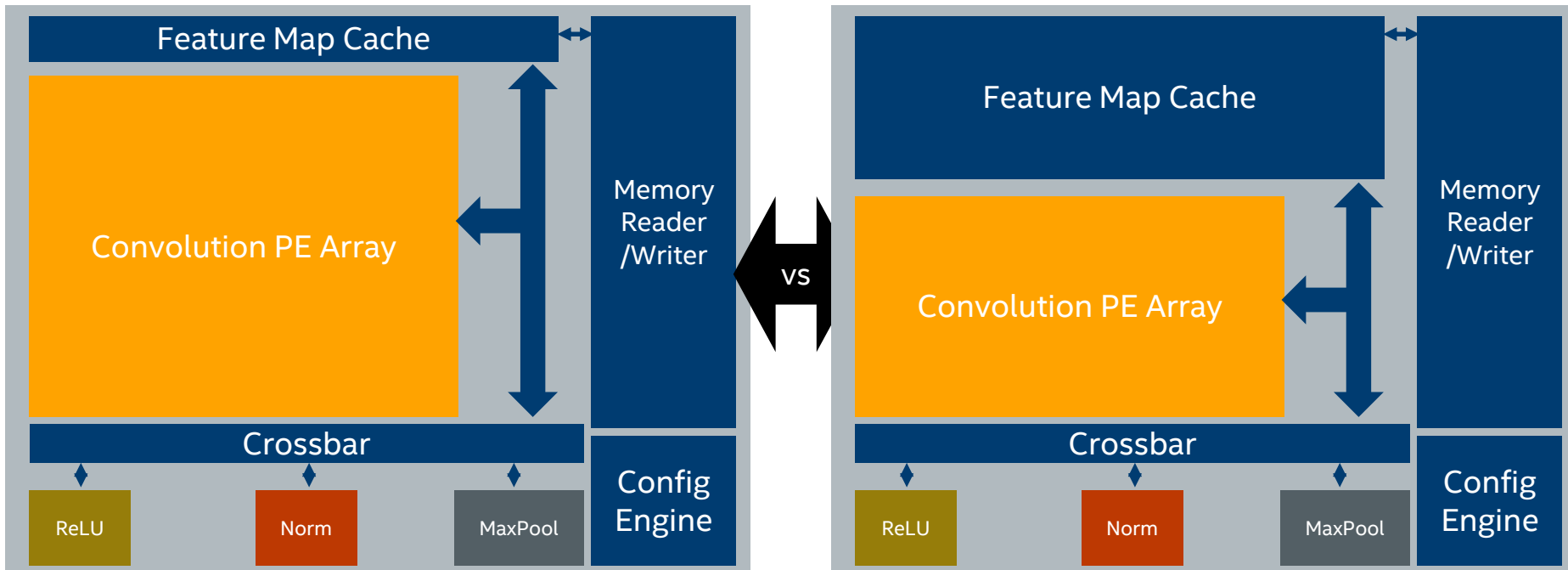
Support for Different Topologies

Tradeoff between features and performance

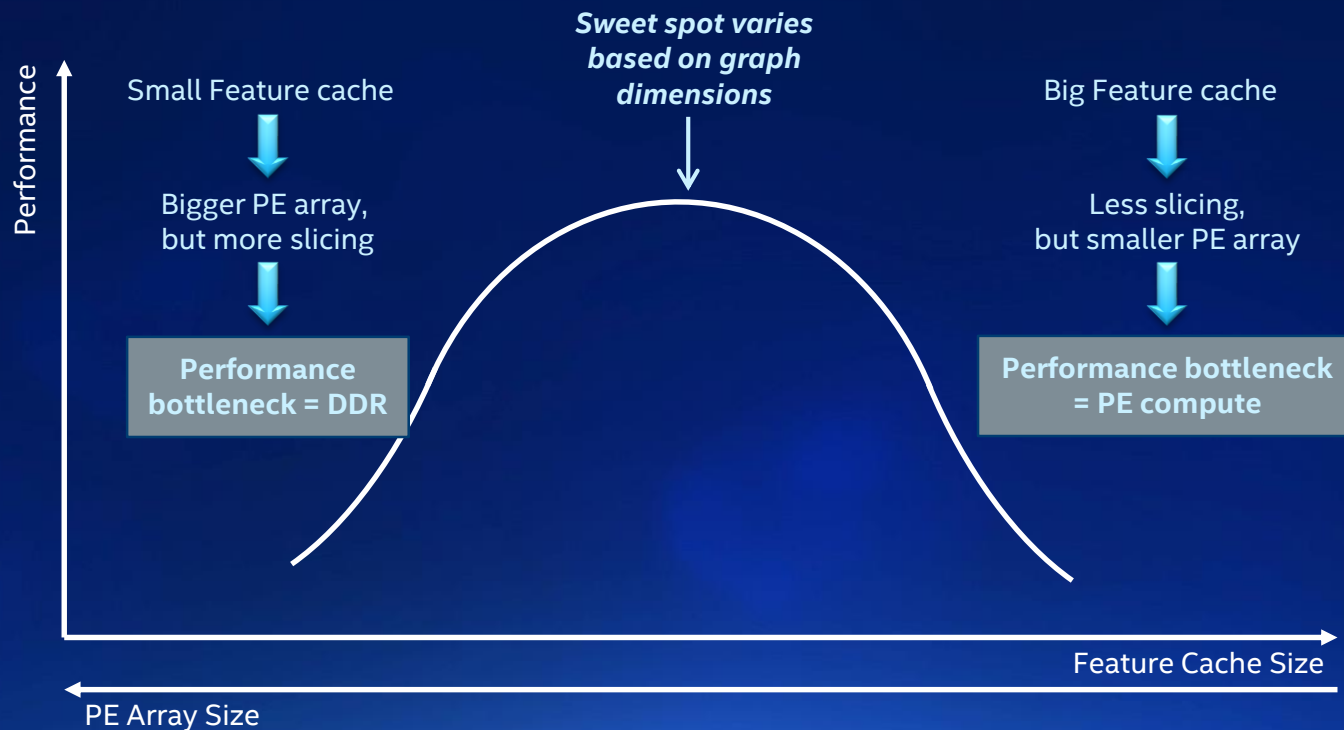


Optimize for Best Performance

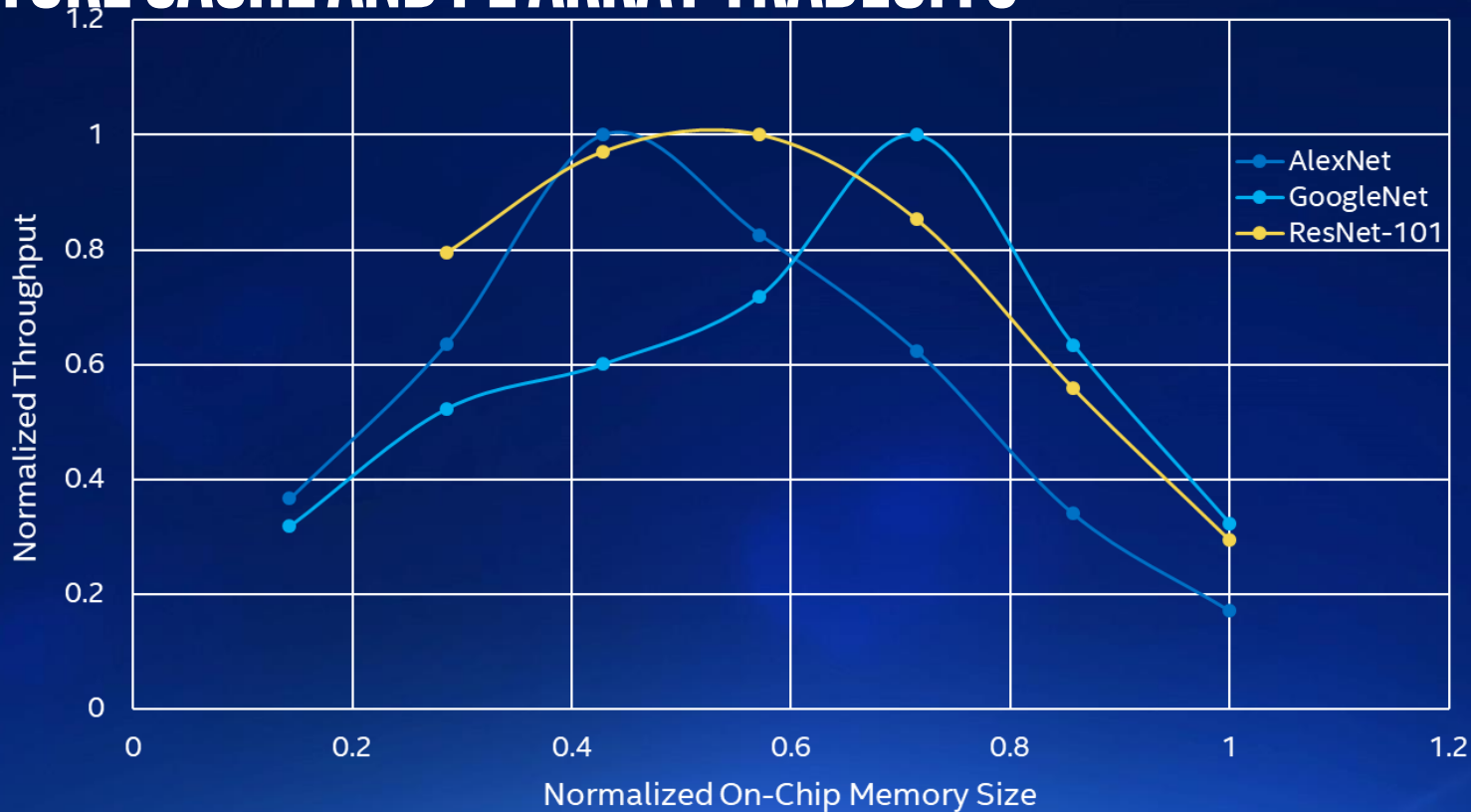
Tradeoff between size of Feature Map cache and convolutional PE array



FEATURE CACHE AND PE ARRAY TRADEOFFS

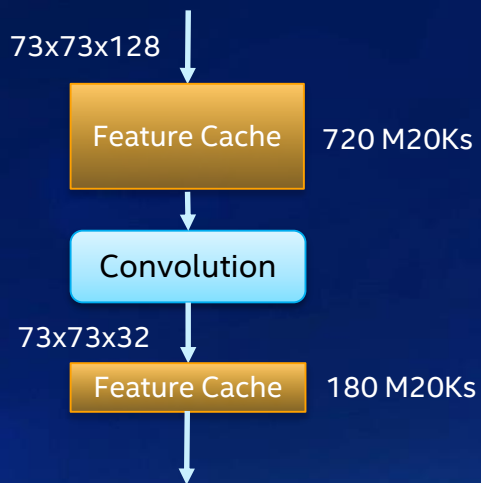


FEATURE CACHE AND PE ARRAY TRADEOFFS



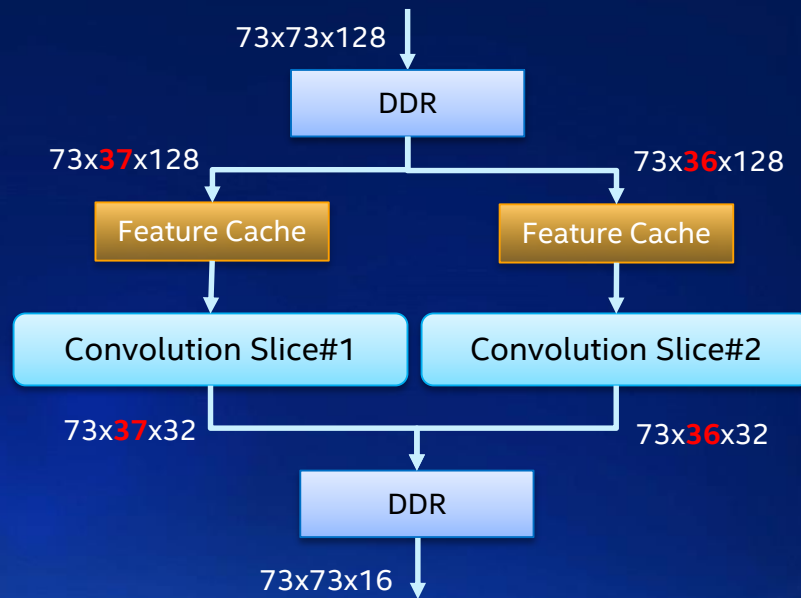
FEATURE CACHE AND SLICING FOR BIG GRAPHS

Un-Sliced Graph



Feature Cache Required: **900** M20Ks

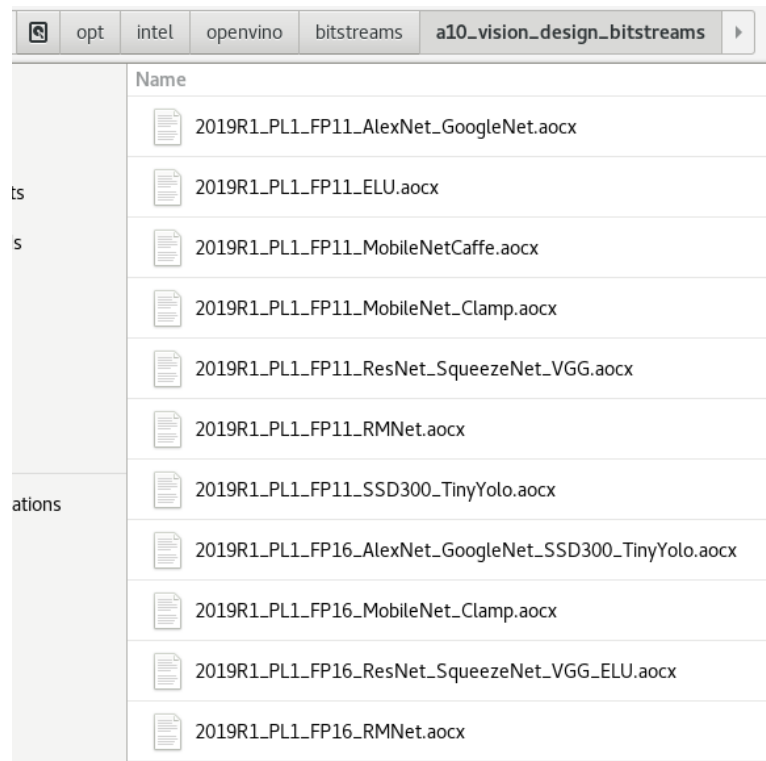
Sliced Graph



Feature Cache Required: **384** M20Ks

Precompiled Architectures (aocx) included with OpenVINO™ toolkit

- Precompiled and included architectures are optimized for standard topologies
- Better performance maybe achieved with custom architectures for custom deep learning networks
 - Size requirement maybe different from standard topology
 - Primitive requirement may be different from standard networks



Supported Primitives and Topologies

Primitives

- ✓ Conv
- ✓ Concat
- ✓ Pooling
- ✓ ScaleShift
- ✓ Fully Connected
- ✓ Custom
- ✓ ReLu, Leaky ReLU
- ✓ Eltwise
- ✓ Power
- ✓ Batch Norm
- ✓ LRM Normalization

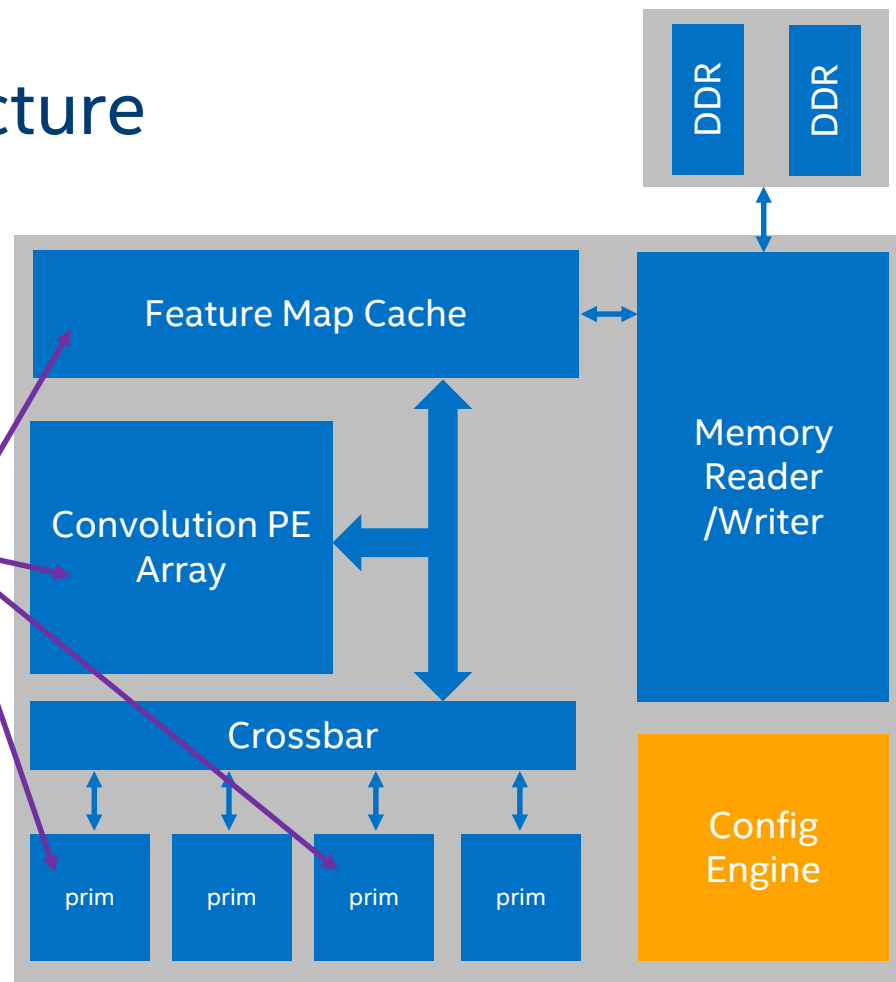
Topologies

- ✓ AlexNet
- ✓ GoogLeNet
- ✓ ResNet-18/50/101/152
- ✓ SqueezeNet
- ✓ VGG-16/19
- ✓ Tiny Yolo
- ✓ LeNet
- ✓ MobileNet v1/v2
- ✓ SSD
- ✓ SSD
- ✓ SSD
- ✓ SSD

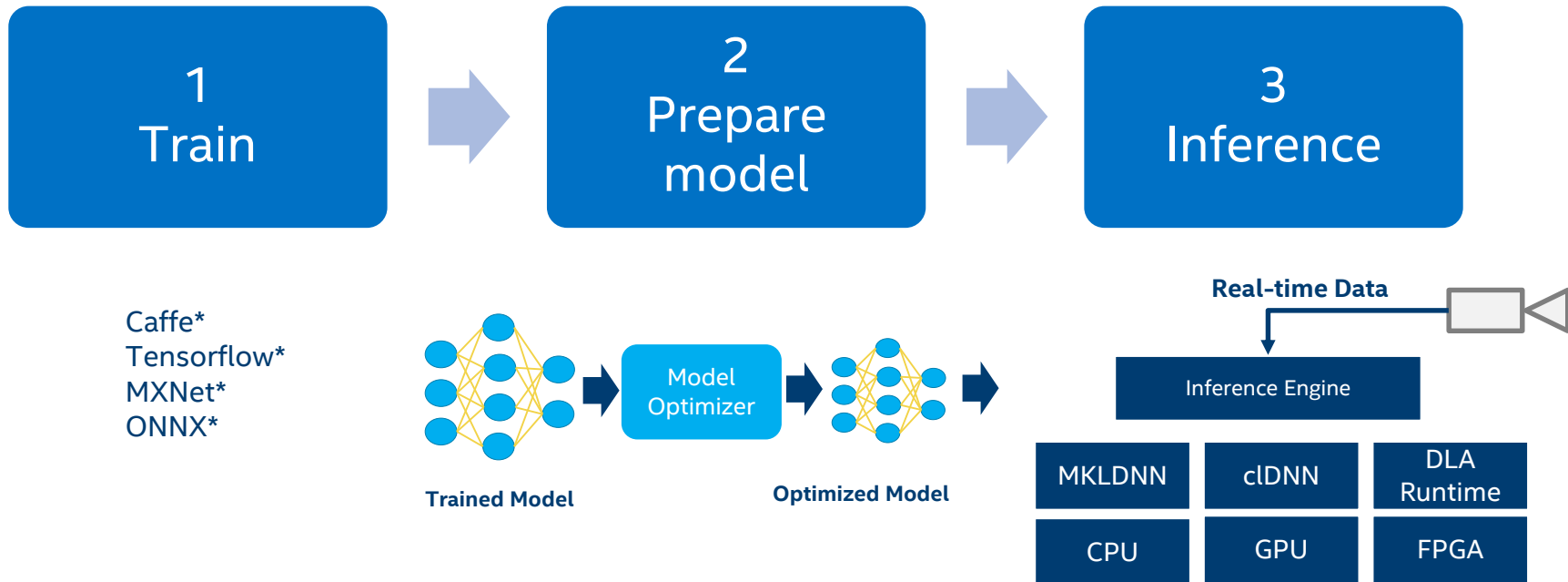
Customizable DLA Architecture

- Many aspects of a DLA architecture can be customized
 - Convolution Processing Element parallelization
 - Datatype
 - Primitive Support
 - Stream Buffer Size
 - Custom Primitives
 - etc.

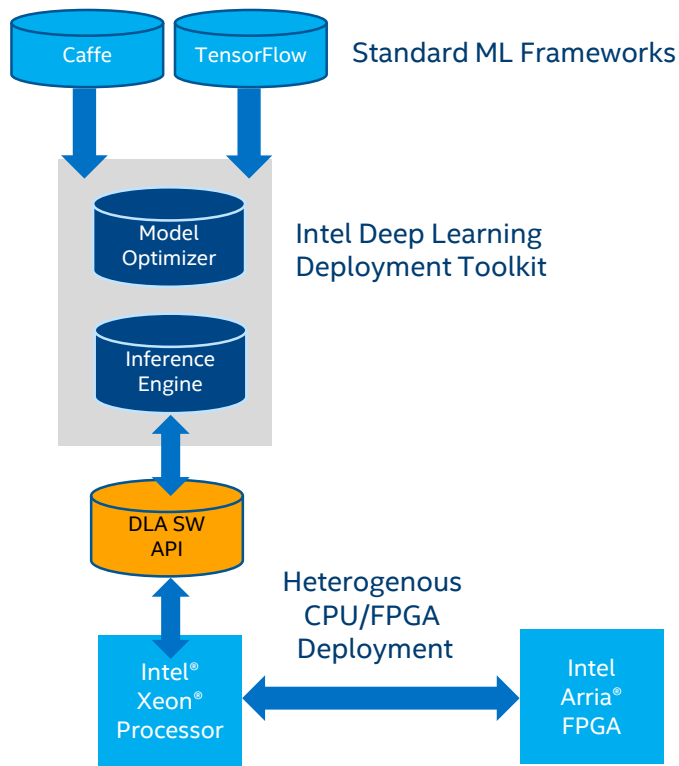
Customizable



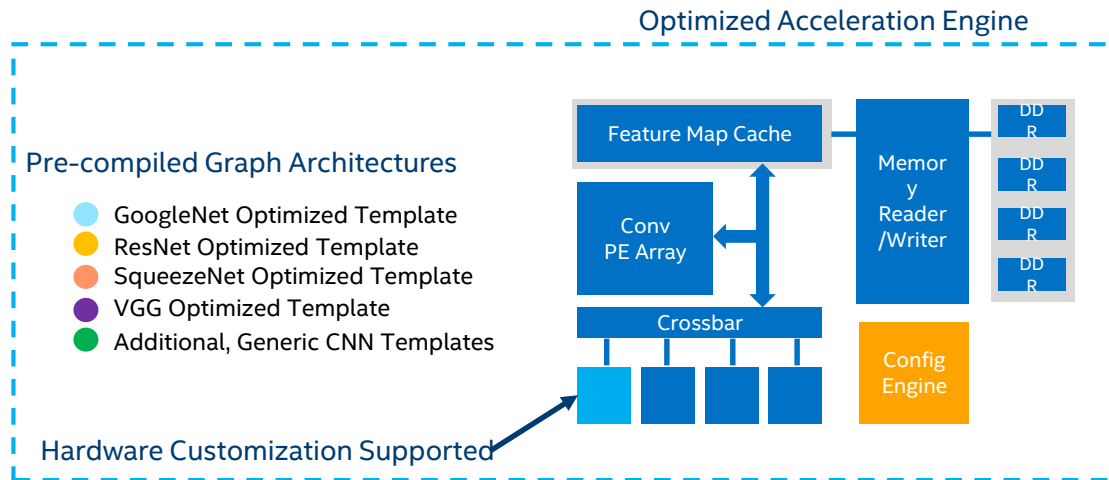
End-to-End Machine Learning



Intel FPGA Deep Learning Acceleration Suite



- Supports common software frameworks (Caffe, Tensorflow)
- Intel DL software stack provides graph optimizations
- Intel FPGA Deep Learning Acceleration Suite provides turn-key or customized CNN acceleration for common topologies



Intel® FPGA Deep Learning Acceleration Suite

- Contents
 - Pre-compiled FPGA Bitstreams
 - To support various layers required by different networks with different precisions
 - Primitives supported:
 - Convolution, Fully Connected, Inner Product, ReLU, pReLU, Pooling, Norm, Concat, Elsewise Add, etc...
 - Software runtime library
 - Board Support Packages to support various boards
 - Tools
 - Diagnose
 - Bitstream customization and generation

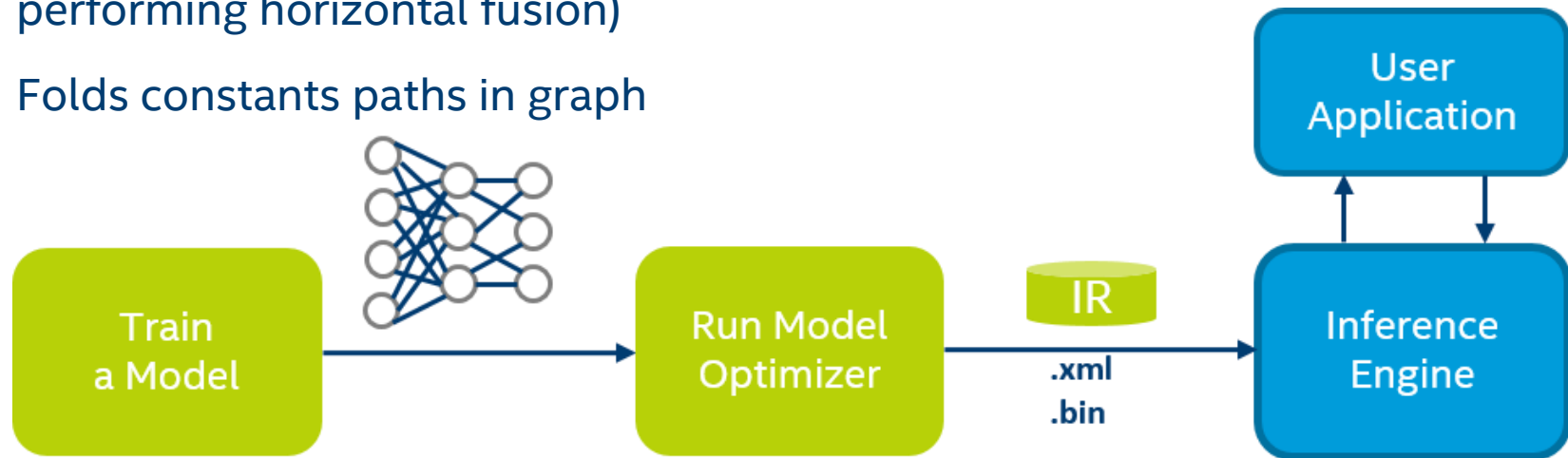
AGENDA

- Intel® and AI / Machine Learning
- Accelerate Deep Learning Using OpenVINO Toolkit
- **Deep Learning Acceleration with FPGA**
 - FPGAs and Machine Learning
 - Intel® FPGA Deep Learning Acceleration Suite
 - **Execution on the FPGA (Model Optimizer & Inference Engine)**
- Intel® Agilex® FPGA
- OneAPI

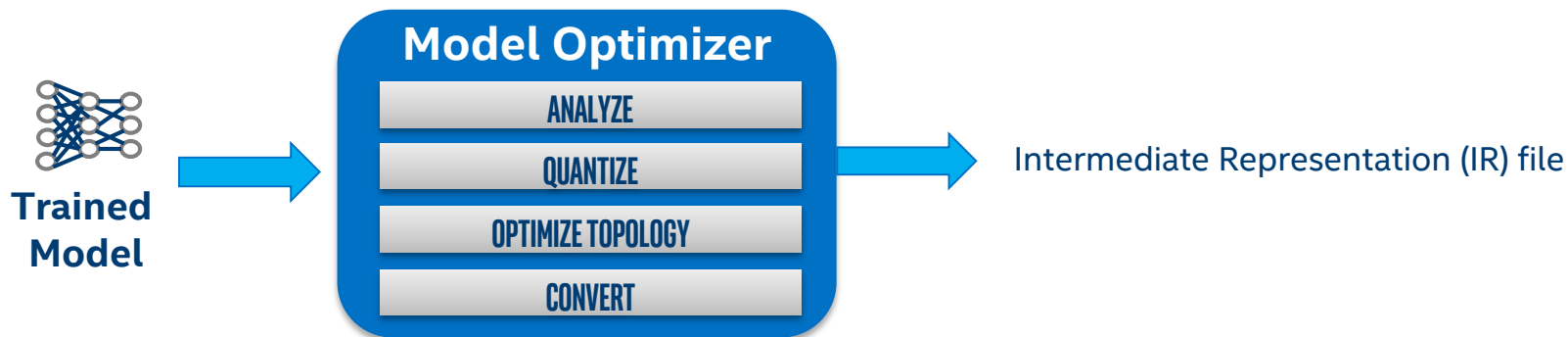
MODEL OPTIMIZER

Model Optimizer

- Convert models from various frameworks (Caffe, TensorFlow, MXNet, ONNX)
- Converts to a unified Model (IR, later n-graph)
- Optimizes topologies (Node merging, batch normalization elimination, performing horizontal fusion)
- Folds constants paths in graph



Improve Performance with Model Optimizer



- Easy to use, Python*-based workflow does not require rebuilding frameworks.
- Import Models from various supported frameworks - Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*.
- 100+ models for Caffe, MXNet and TensorFlow validated. All public models on ONNX* model zoo supported.
- With Kaldi support, the model optimizer extends inferencing for non-vision networks.
- IR files for models using standard layers or user-provided custom layers do not require Caffe.
- Fallback to original framework is possible in cases of unsupported layers, but requires original framework.

Model Optimizer

Model optimizer performs generic optimization:

- Node merging
- Horizontal fusion
- Batch normalization to scale shift
- Fold scale shift with convolution
- Drop unused layers (dropout)
- FP16/Int8 quantization
- Model optimizer can add normalization and mean operations, so some preprocessing is 'added' to the IR

	FP32	FP16	FP11	INT8
CPU	yes	no	no	yes
GPU	yes	recommended	no	no
MYRIAD	no	yes	no	no
FPGA/DLA	no	yes	yes	no

`--mean_values (104.006, 116.66, 122.67)`

`--scale_values (0.07, 0.075, 0.084)`

Model Optimization Techniques

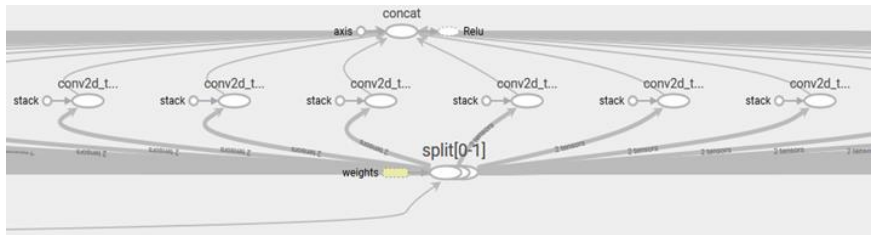
Linear Operation Fusing & Grouped Convolutions Fusing

Linear Operation Fusing: 3 stages

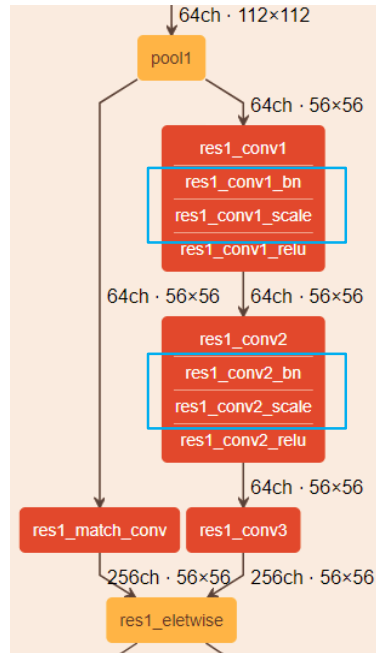
1. **BatchNorm and ScaleShift decomposition:** BN layers decomposes to *Mul->Add->Mul->Add* sequence; ScaleShift layers decomposes to *Mul->Add* sequence.
2. **Linear operations merge:** Merges sequences of Mul and Add operations to the **single** Mul->Add instance.
3. **Linear operations fusion:** Fuses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

Grouped Convolutions Fusing

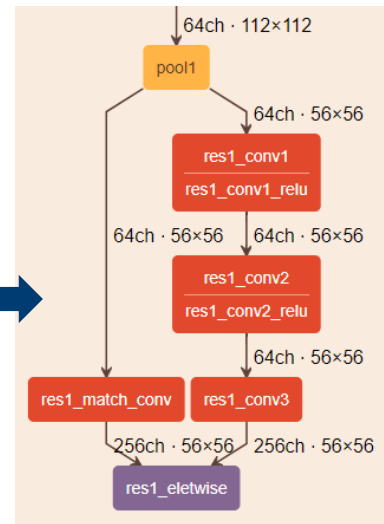
Specific optimization that applies for TensorFlow* topologies. (Xception*)



Split->Convolutions->Concat block from TensorBoard*



Caffe Resnet269 block (from Netscope)

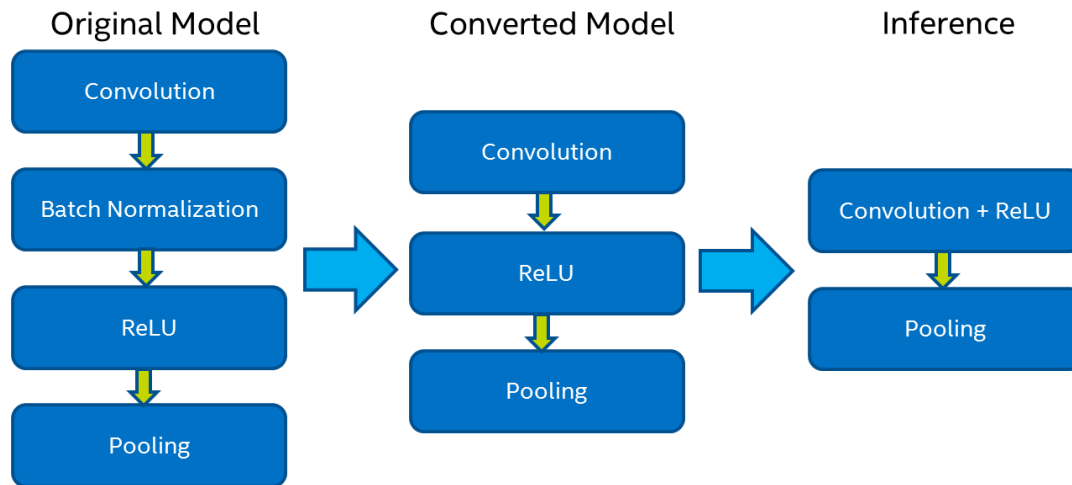


Merged Caffe* Resnet269 block (from Netscope*)

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: Cutting Off Parts of a Model

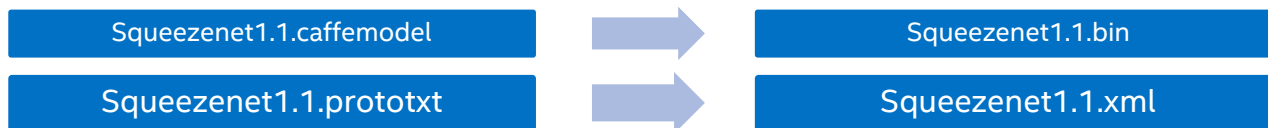
Model optimizer can cut out a portion of the network:

- Model has pre/post-processing parts that cannot be mapped to existing layers.
- Model has a training part that is not used during inference.
- Model is too complex and cannot be converted in one shot.

Command line options

- Model Optimizer provides command line options `--input` and `--output` to specify new entry and exit nodes ignoring the rest of the model:
- `--input` option accepts a comma-separated list of layer names of the input model that should be treated as new entry points to the model;
- `--output` option accepts a comma-separated list of layer names of the input model that should be treated as new exit points from the model.

Intermediate Representation (IR)



```
layer {
  name: "data"
  type: "Input"
  top: "data"
  input_param { shape: { dim: 1 dim: 3 dim: 227 dim: 227 } }
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  convolution_param {
    num_output: 64
    kernel_size: 3
    stride: 2
  }
}
```

```
<net batch="1" name="model" version="2">
  <layers>
    <layer id="100" name="data" precision="FP32" type="Input">
      <output>
        <port id="0">
          <dim>1</dim>
          <dim>3</dim>
          <dim>227</dim>
          <dim>227</dim>
        </port>
      </output>
    </layer>
    <layer id="129" name="conv1" precision="FP32" type="Convolution">
      <data dilation-x="1" dilation-y="1" group="1" kernel-x="3" kernel-y="3" output="64" pa>
        <input>
          <port id="0">
            <dim>1</dim>
            <dim>3</dim>
            <dim>227</dim>
            <dim>227</dim>
          </port>
        </input>
        <output>
          <port id="3">
            <dim>1</dim>
            <dim>64</dim>
            <dim>113</dim>
            <dim>113</dim>
          </port>
        </output>
      </data>
      <blobs>
        <weights offset="2275104" size="6912"/>
        <biases offset="4805920" size="256"/>
      </blobs>
    </layer>
  </layers>
</net>
```

Model Optimizer Options

Python script: `$MO_DIR/mo.py`

Option for Deployment	Description
<code>--input_model</code>	Network binary weights file TensorFlow* .pb Caffe* .caffemodel MXNet* .params
<code>--input_proto</code>	Caffe .prototxt file
<code>--data_type</code>	IP Precision (i.e. FP16)
<code>--scale</code>	Network normalization factor (Optional)
<code>--output_dir</code>	Output directory path (Optional)

Full Model Optimizer options covered in OpenVINO™ documentations

Run Model Optimizer

- To generate IR .xml and .bin files for Inference Engine

```
$ source $MO_DIR/venv/bin/activate
$ cd $MO_DIR/

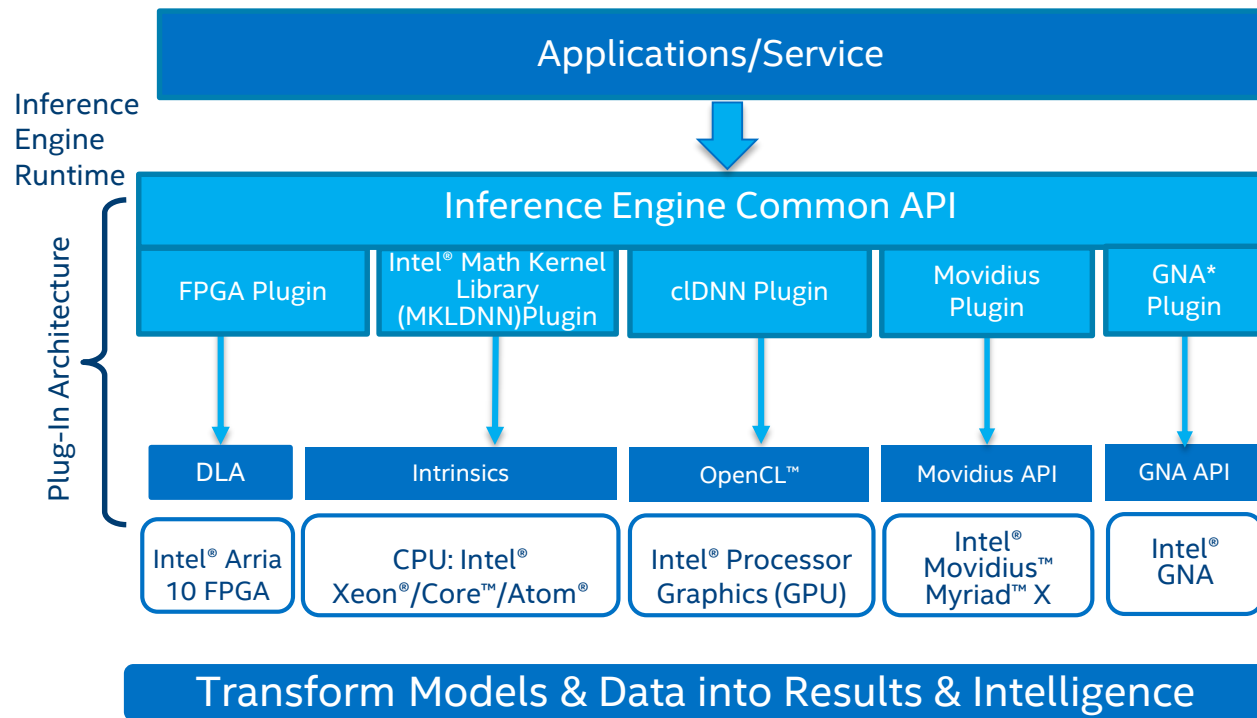
$ python mo.py \
  --input_model <model dir>/<weights>.caffemodel \
  --scale 1 \
  --data_type FP16 \
  --output_dir <output dir>
```

```
Model Optimizer arguments
Batch: 1
Precision of IR: FP16
Enable fusing: True
Enable gfusing: True
...
[ SUCCESS ] Generated IR model.
[ SUCCESS ] XML file: ...
[ SUCCESS ] BIN file: ...
```

INFERENCE ENGINE

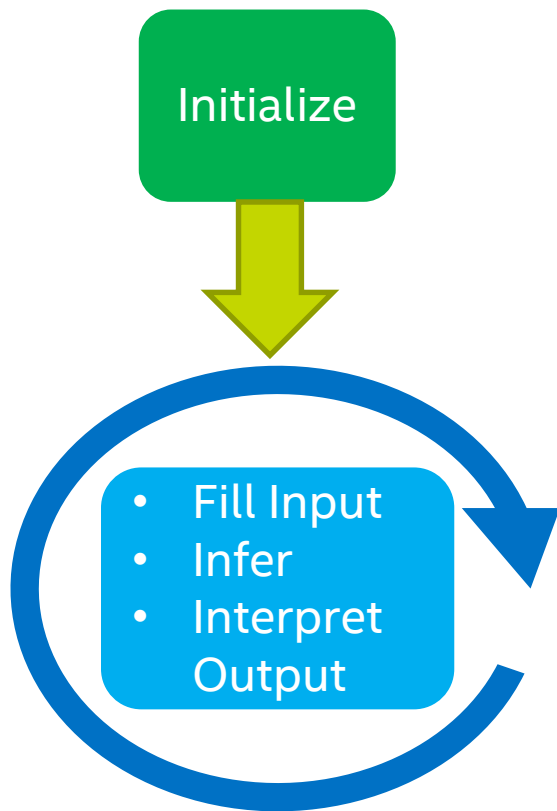
Optimal Model Performance Using the Inference Engine

- Simple & Unified API for Inference across all Intel® architecture
- Optimized inference on large IA hardware targets (CPU/GEN/FPGA)
- Heterogeneity support allows execution of layers across hardware types
- Asynchronous execution improves performance
- Futureproof/scale your development for future Intel® processors



GPU = Intel CPU with integrated graphics/Intel® Processor Graphics/GEN
GNA = Gaussian mixture model and Neural Network Accelerator

Inference Engine Workflow



Initialization

Create IE Instance
Load IE Plugin (MKL-DNN/cIDNN/DLA)
Read an IR Model
Set Target Device (CPU/GPU/Movidius/FPGA)
Load network to plugin
Allocate input, output buffers

Main loop

Fill input buffer with data
Run inference
Interpret output results

Inference Engine Details

A runtime with a unified API for integrating inference with application logic

- Delivers optimized inference solution with reduced footprint on inference execution for target hardware
- `libinference_engine.so` library implements core functions
 - Loading and parsing of model IR
 - Preparation of input and output blobs
 - Triggers inference using specified plug-in
- **Include file:** `inference_engine.hpp`

Inference Engine Plugins

- CPU MKLDNN Plugin (Intel® Math Kernel Library for Deep Neural Networks)
 - Supports Intel® Xeon®/Core®/Atom® platform
 - Widest set of network classes supported, easiest way to enable topology
- GPU clDNN Plugin (Compute Library for Deep Neural Networks)
 - Supports 9th generation and above Intel® HD and Iris graphics processors
 - Extensibility mechanism to develop custom layers through OpenCL™
- FPGA DLA Plugin
 - Supports Intel® FPGAs
 - Basic set of layers are supported on FPGA, non-supported layers inferred through other plugins

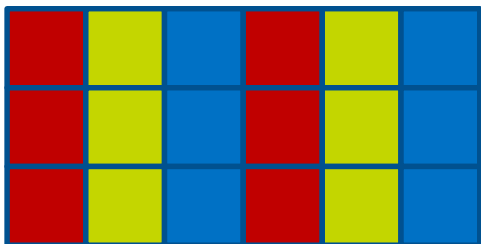
Heterogeneous Plugin

- Enables inference of one network on several devices.
 - Calculate heaviest parts of network on accelerator, unsupported fallback to CPU
 - To utilize all available hardware more efficiently during one inference
- Heterogeneous execution can be triggered in two independent steps:
 - Loading a network to the Heterogeneous plugin, splitting the network to parts, and executing them through the plugin
 - Setting of affinity to layers (binding them to devices in `InferenceEngine::ICNNNetwork`)
 - Can be done automatically or in manual mode

Pre-processing

- Most image formats are interleaved (RGB, BGR, BGRA, etc.)
- Models usually expect RGB planar format:
 - R-plane, G-plane, B-plane

Interleaved



Planar



Post-processing

Developers are responsible for parsing inference output.

Many output formats are available. Some examples include:

- **Simple Classification (alexnet*)**: an array of float confidence scores, # of elements=# of classes in the model
- **SSD**: Many “boxes” with a confidence score, label #, xmin,ymin, xmax,ymax

Unless a model is well documented, the output pattern may not be immediately obvious.

Inference Engine Classes

```
using namespace InferenceEngine;
```

Class	Details
InferencePlugin	Main plugin interface
PluginDispatcher	Finds suitable plug-in for specified device
CNNNetReader	Build and parse a network from given IR
CNNNetwork	Neural Network and binary information
Blob, BlobMaps	Container object representing a tensor (e.g. input/output)
InputsDataMap	Information about input of the network
ExecutableNetwork	Loaded Network
InferRequest	Interact with inputs/outputs and performs inference

Inference Engine API Usage (1)

1. Load Plugin

- FPGA Plugin: libdliaPlugin.so
 - Others: libclDNNPlugin.so (GPU), libMKLDNNPlugin.so (CPU)
- Plugin Dir: <OpenVINO install

```
dir>/inference_engine/lib/<OS>/intel64  
InferenceEngine::PluginDispatcher dispatcher(<pluginDir>);  
InferencePlugin plugin = dispatcher.getPluginByDevice("HETERO:FPGA, CPU");
```

- ## 2. Load Model
- ```
InferenceEngine::CNNNetReader netBuilder
netBuilder.ReadNetwork("<Model>.xml");
netBuilder.ReadWeights("<Model>.bin");
```

# Inference Engine API Usage (2)

## 3. Prepare Input and Output Blobs

- For Inputs
  - Allocate based on the size of the input, number of channels, batch size, etc.
  - Set input precision
  - Set layout
  - Fill in data (i.e. from rgb value of image)
- For Output
  - Set output precision
  - Allocate based on output format



# Inference Engine API Usage (3)

## 4. Load the model to the plugin

```
ExecutableNetwork exeNet = plugin.LoadNetwork(netBuilder.getNetwork(), {});
```

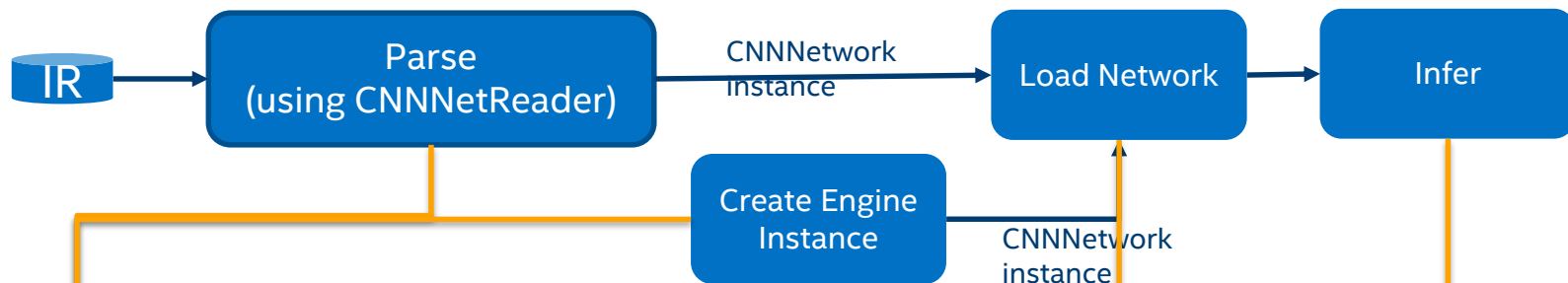
## 5. Perform inference

```
InferRequest infer_request = exeNet.CreateInferRequest();
infer_request.Infer();
```

## 6. Process output blobs

```
const Blob::Ptr output_blob = infer_request.GetBlob(firstOutputName);
```

# Using the Inference Engine API



```
CNNNetReader netBuilder;
netBuilder->ReadNetwork("Model.xml");
netBuilder->ReadWeights("Model.bin")
```

```
InferencePlugin plugin PluginDispatcher({}).getPluginbyDevice("HETERO:FPGA, CPU");
```

```
ExecutableNetwork exeNet = plugin.LoadNetwork(netBuilder->getNetwork(), {});
```

```
InferRequest infer_request = exeNet.CreateInferRequest();
//Create Input Blob, iterate through and fill in data
```

```
infer_request.Infer();
```

```
//Process Output
```

# Batch Execution

- For better performance, using a larger batch size may help
- Allocate input and output Blob according to batch size
- Set the Batch size on the network

```
netBuilder.getNetwork().setBatchSize(<size>);
```

# Inference Engine Example Applications

- Execute samples and demos with FPGA support
  - Shipped with the OpenVINO™ toolkit
  - classification\_sample
  - object\_detection\_sample\_ssd
  - many others...

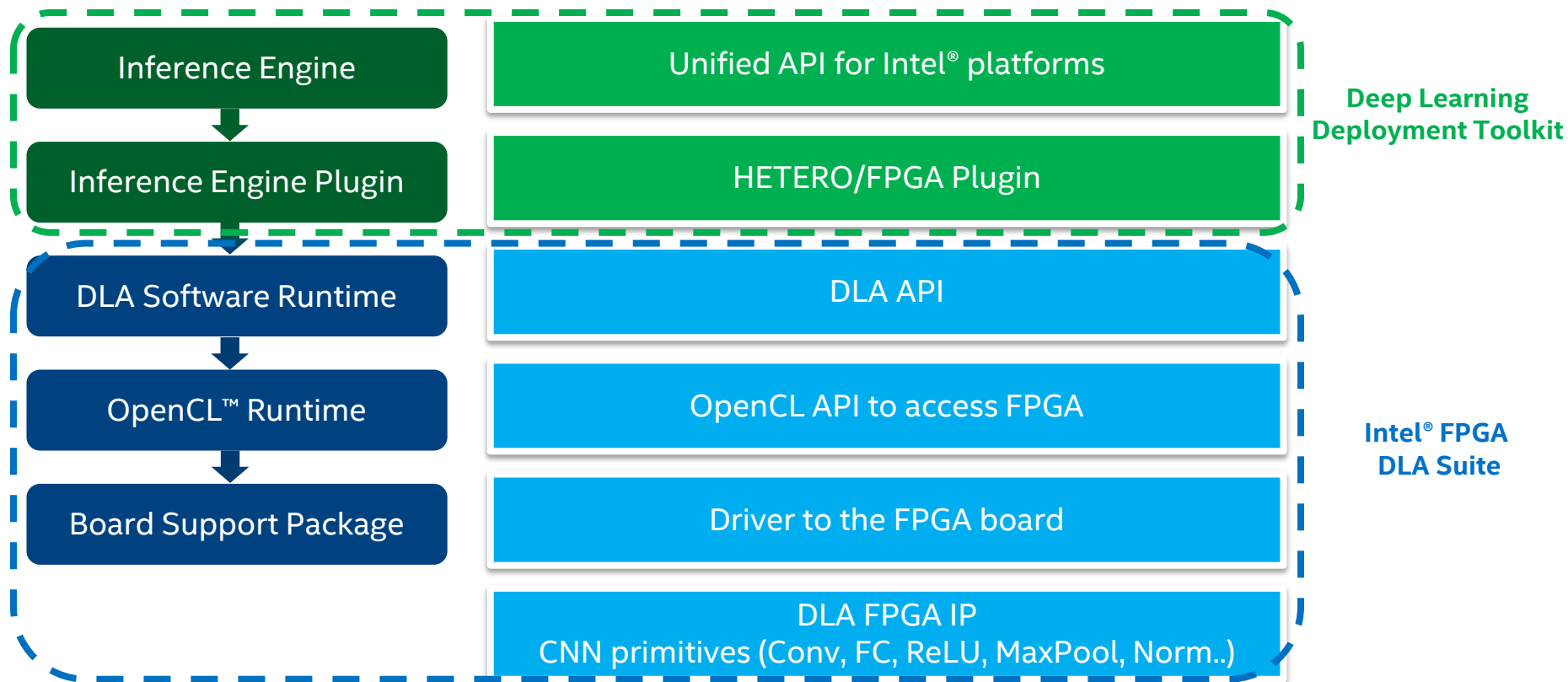
# Example Application using FPGAs

```
$./object_detection_sample_ssd -m <path_to_model>/Model.xml -i
<path_to_pictures>/picture.jpg -d HETERO:FPGA,CPU
```

The “**priorities**” defines a greedy behavior:

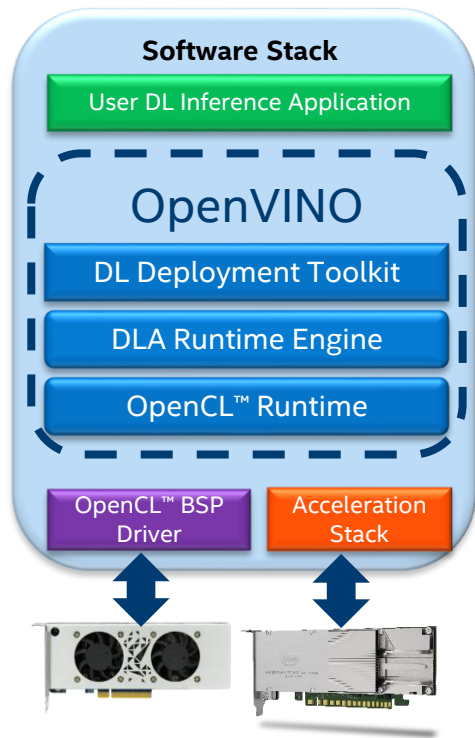
- Keeps all layers that can be executed on the device (FPGA)
- Carefully respects topological and other limitations
- Follows priorities when searching (for example, CPU)

# IE Software Architecture with FPGAs



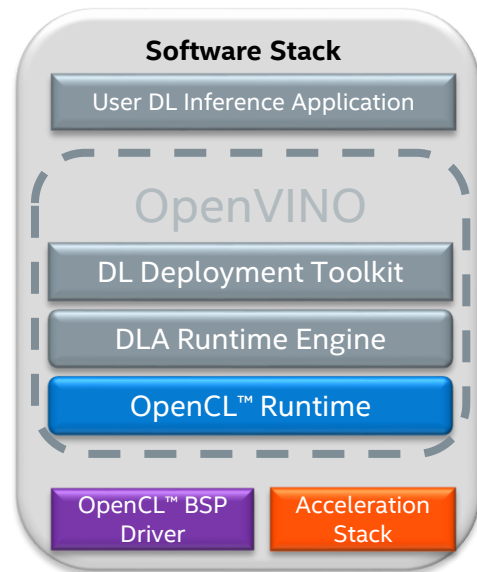
# Software Requirements

- **Intel® Distribution of OpenVINO™ toolkit (Linux for FPGA)**
  - Includes Intel® FPGA Runtime (RTE) for OpenCL™
- For Vision Accelerator Design with FPGA board
  - **OpenCL™ Board Support Package for Intel® Vision Accelerator Design with an Intel® Arria® 10 FPGA**
  - Intel® Quartus® Prime Pro Edition Programmer
- For Programmable Acceleration Card
  - **Intel® Acceleration Stack for Runtime**
    - Includes Programmer, RTE for OpenCL, and OpenCL BSP



# FPGA Software Components

- Intel® FPGA OpenCL™ Runtime
  - Included as part of OpenVINO™
  - Inference acceleration done through OpenCL calls
- Board Support Package
  - Translates OpenCL calls to low-level device driver calls
  - Allows programming of FPGA with different bitstreams
  - Included as part of Acceleration Stack for PAC boards
- Intel® Quartus® Prime Programmer
  - Possibly needed to flash/program the board with FPGA image to communicate with host





# Prepare FPGA Environment for Inference

- Once software stack is installed and environment set
- Ensure PCIe\* connection is operating

- `lspci | grep Altera`

```
01:00.0 Processing accelerators: Altera Corporation Device 2494 (rev 01).
```

- Ensure FPGA OpenCL is working correctly

- `aocl diagnose`

- Ensure `DIAGNOSTIC_PASSED`

```
bash-4.2$ aocl diagnose

Device Name:
acl0

BSP Install Location:
/opt/altera/aocl-pro-rte/aclrte-linux64/board/hddlf_1150_sg1

Vendor: Intel(R) Corporation

Phys Dev Name Status Information

acla10_1150_sg10Passed Intel Vision Accelerator Design with Intel Arria 10 FPGA (acla10_1150_sg10)
 PCIe dev_id = 2494, bus:slot.func = 05:00.00, Gen3 x8
 FPGA temperature = 42.9727 degrees C.

DIAGNOSTIC_PASSED

Call "aocl diagnose <device-names>" to run diagnose for specified devices
Call "aocl diagnose all" to run diagnose for all devices
```

# Load FPGA Image and Execute IE Application

- FPGAs needs to be preconfigured with primitives prior to application execution
- Choose FPGA bitstream from the DLA suite
  - Based on topology needs and data type requirements
  - Option to create custom FPGA bitstream based on requirements

```
aocl program acl0
```

```
/opt/intel/opencvino/bitstreams/<board>/2019R1_PL1_FP11_MobileNet_SqueeeNet_VGG.aocx
```

- Execute User or Example Application

# FPGA Image Selection

- Precompiled FPGA images included
- Choose image based on
  - Primitives needed (architecture)
  - Data type support for accuracy/performance trade-off
  - K (filter) and C (channel depth) vectorization for performance
  - Data path width
  - On-chip stream buffer depth
- May also generate customized FPGA image

| Name                                                   |
|--------------------------------------------------------|
| 2019R1_PL1_FP11_AlexNet_GoogleNet.aocx                 |
| 2019R1_PL1_FP11_ELU.aocx                               |
| 2019R1_PL1_FP11_MobileNetCaffe.aocx                    |
| 2019R1_PL1_FP11_MobileNet_Clamp.aocx                   |
| 2019R1_PL1_FP11_ResNet_SqueezeNet_VGG.aocx             |
| 2019R1_PL1_FP11_RMNet.aocx                             |
| 2019R1_PL1_FP11_SSD300_TinyYolo.aocx                   |
| 2019R1_PL1_FP16_AlexNet_GoogleNet_SSD300_TinyYolo.aocx |
| 2019R1_PL1_FP16_MobileNet_Clamp.aocx                   |
| 2019R1_PL1_FP16_ResNet_SqueezeNet_VGG_ELU.aocx         |
| 2019R1_PL1_FP16_RMNet.aocx                             |

# Use OpenVINO™

## Running SqueezeNet example manually or create your own application

1. Create a new test directory and cd into it, ensure environment script is ran
2. Copy original Caffe\* files from the demo directory into the new directory

```
cp ~/openvino_models/models/FP32/classification/squeezenet/1.1/caffe/squeezenet1.1.* .
```

3. Copy the imagenet labels from the demo directory

```
cp ~/openvino_models/ir/FP32/classification/squeezenet/1.1/caffe/squeezenet1.1.labels .
```

4. Execute model optimizer to convert to intermediate representation

```
mo -input_model squeezenet1.1.caffemodel
```

5. Execute classification sample

```
classification_sample -m squeezenet1.1.xml -i $IE_INSTALL/demo/car.png
classification_sample -m squeezenet1.1.xml -i $IE_INSTALL/demo/car.png -d HETERO:FPGA,CPU
classification_sample -m squeezenet1.1.xml -i $IE_INSTALL/demo/car.png -d HETERO:FPGA,CPU -ni 100
```

# Summary

- The Intel® Deep Learning Acceleration Suite part of Intel® Distribution of OpenVINO™ toolkit provides out-of-box deployment of deep learning inference on FPGAs
- Deep Learning Deployment Toolkit can be used as a frontend for the DLA
  - Consists of the Model Optimizer and Inference Engine
  - Use Model Optimizer to optimize and convert framework model to unified internal representation
  - Use Inference Engine unified API to deploy network on various devices including the FPGA

# References

- Follow-on Training
  - [Deploying FPGAs for Deep Learning Inference with Intel® Distribution of OpenVINO™ toolkit](#)
- Intel® AI
  - [www.intel.com/ai](http://www.intel.com/ai)
- Intel® Distribution of OpenVINO™ toolkit
  - <https://software.intel.com/en-us/openvino-toolkit>
- [Intel® FPGAs for AI](#)
- [Intel® FPGA Acceleration Stack for Intel® Xeon® CPU with FPGAs](#)

# AGENDA

- Intel® and AI / Machine Learning
- Accelerate Deep Learning Using OpenVINO Toolkit
- Deep Learning Acceleration with FPGA
  - FPGAs and Machine Learning
  - Intel® FPGA Deep Learning Acceleration Suite
  - Execution on the FPGA (Model Optimizer & Inference Engine)
- **Intel® Agilex® FPGA**
- OneAPI



# INTRODUCING THE FPGA FOR THE DATA-CENTRIC WORLD

Intel® Agilex® FPGAs





# Introducing Intel® Agilex™ FPGAs

**Intel innovation for ultimate  
agility and flexibility**



## **HIGH-PERFORMANCE COMPUTE**

10nm Process

Memory-Coherent Intel® Xeon®  
Processor Acceleration

Massive Bandwidth

## **ANY-TO-ANY INTEGRATION**

Memory, Analog, Logic

Any Node, Supplier, IP

Rapid Intel® eASIC™  
Devices Optimization

## **ANY DEVELOPER**

Intel® Quartus® Prime  
Design Tool for  
Hardware Developers

One API for Software  
Developers

# The FPGA for the Data-Centric World

## PROCESS DATA

2<sup>ND</sup>  
GENERATION  
INTEL<sup>®</sup>  
HYPERFLEX<sup>™</sup>  
ARCHITECTURE

UP TO  
40%  
HIGHER  
PERFORMANCE<sup>1,3</sup>

UP TO  
40%  
LOWER  
POWER<sup>1,3</sup>

UP TO  
40 TFLOPS  
DSP PERFORMANCE<sup>2,3</sup>

## STORE DATA

DDR5 &  
HBM

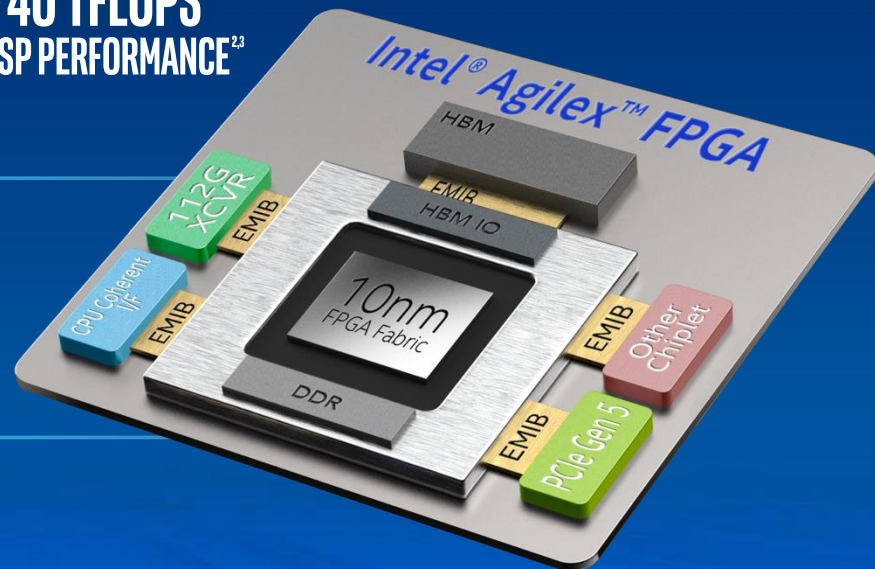
INTEL<sup>®</sup> OPTANE<sup>™</sup> DC  
PERSISTENT MEMORY SUPPORT

## MOVE DATA



INTEL<sup>®</sup> XEON<sup>®</sup> PROCESSOR COHERENT  
CONNECTIVITY & PCIe GEN5

112G  
TRANSCIVER  
DATA RATES



<sup>1</sup> Compared to Intel<sup>®</sup> Stratix<sup>®</sup> 10 FPGAs

<sup>2</sup> With FP16 configuration

<sup>3</sup> Based on current estimates, see slide 180 for details

# Any-to-Any Heterogenous 3D Integration

## EMBEDDED MULTI-DIE INTERCONNECT BRIDGE

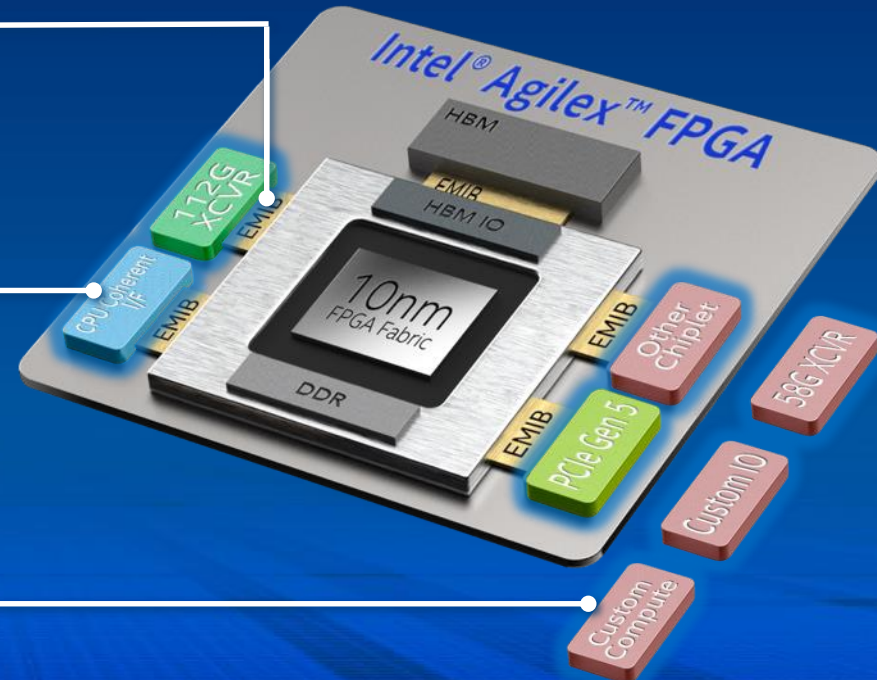
No-compromise die-to-die  
3D packaging interconnect

## CHIPLET-BASED ARCHITECTURE

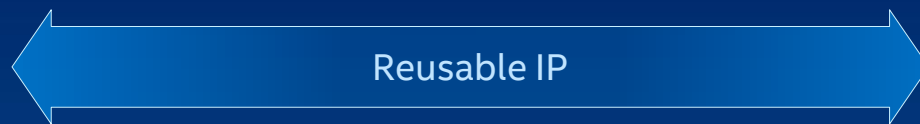
Library of chiplets:  
transceivers, custom I/O &  
custom compute tiles

## INTEL® eASIC™ TILE CUSTOMIZATION

Ultimate customization with  
integrated Intel® eASIC™ tile



# Custom Logic Continuum



Fastest Time to Market  
Highest Flexibility



Optimized TTM with  
Performance / Cost

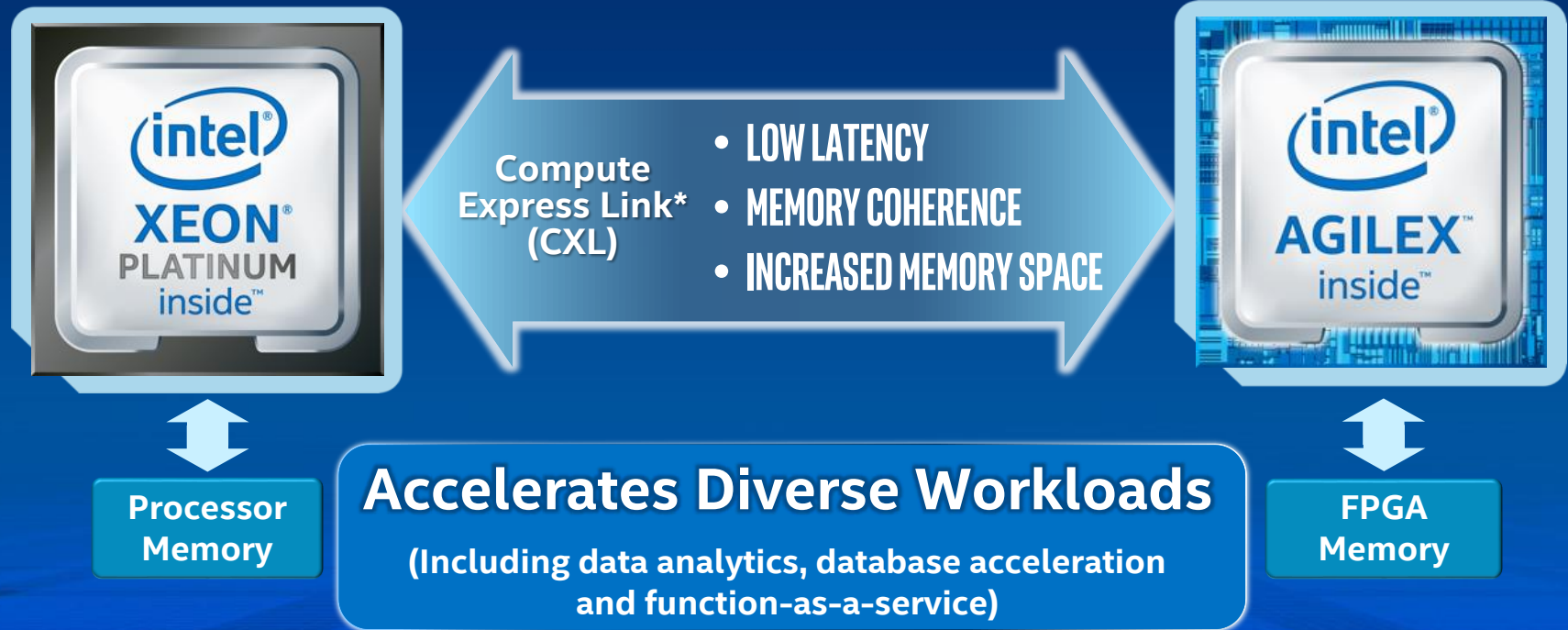


Highest Performance  
Lowest Power & Cost\*

**Flexible and Agile Optimization Across Product Lifecycle**

<sup>1</sup>Refers to Intel® eASIC™ Devices  
\* Compared to other options shown

# First Memory-Coherent Accelerator for Intel® Xeon® Scalable Processors



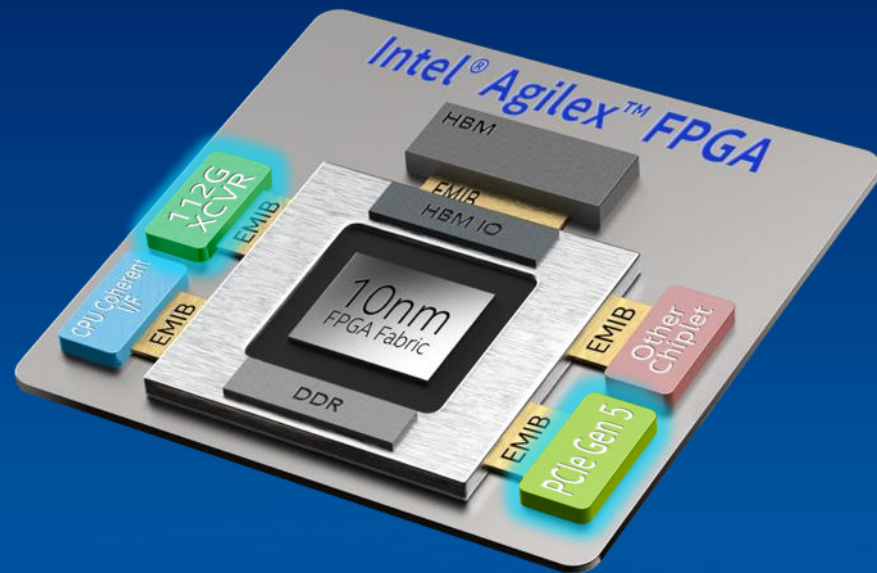


# Transceiver Leadership

# 112G



# PCIe GEN5\*



**High Bandwidth for Applications Including  
400G Networks, Edge Analytics, Data Center Workloads**

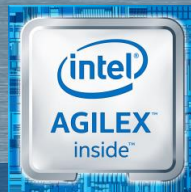
# Well Suited for Artificial Intelligence Applications

UP TO  
**40 TFLOPS**<sup>3</sup>  
FP16 PERFORMANCE  
UP TO

**92 TOPS**<sup>4</sup>  
INT8 PERFORMANCE

Configurable DSP

- FP32
- BFLOAT16
- FP16
- INT8 through INT2

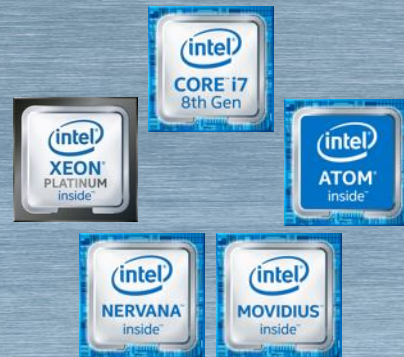


Flexibility for evolving workloads and multi-function integration

AI+ usage model

Only FPGA supporting hardened  
BFLOAT16 & FP16

Complementary to:



**OpenVINO™ toolkit**

**One API**

<sup>3,4</sup> based on current estimates, see slide 180 for details

# Intel® Agilex™ FPGA Tools for Developers



## ONE API

## Hardware Developers

- Higher productivity:
  - 30% improvement in compile times<sup>5</sup>
  - New productivity flows and usability features for faster design convergence
- Higher efficiency: 15% improvement in memory utilization<sup>5</sup>

## Software Developers

- Single source, heterogenous programming environment
- Support for common performance library APIs
- FPGA support with Intel software development tools including Intel® VTune™ Amplifier & Intel® Advisor



<sup>5</sup>See slide 180 for details



# Intel® Agilex™ FPGA Family

## F - SERIES

For wide range of applications

Up to 58G transceivers

PCIe\* Gen4

DDR4

Quad-Core ARM\* Cortex\*-A53  
SoC Option

## I - SERIES

For high-performance  
processor interface and  
bandwidth-intensive applications

Up to 112G transceivers

PCIe\* Gen5

DDR4

Quad-Core ARM\* Cortex\*-A53

Coherent attach to Intel® Xeon®  
Scalable Processor option (CXL)

## M - SERIES

For compute-intensive applications

Up to 112G transceivers

PCIe\* Gen5

DDR4, DDR5 and Intel® Optane™  
DC Persistent Memory support

Quad-Core ARM\* Cortex\*-A53

Coherent attach to Intel® Xeon®  
Scalable Processor option (CXL)

HBM Option

**Intel® Quartus® Prime Design Software Support April 2019;  
First Device Availability Q3 2019**

# Efficient Network Transformation

## DATAPATH ACCELERATION

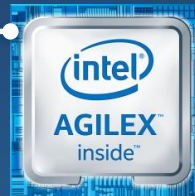
### VNF PERFORMANCE OPTIMIZATION

Load Balancing | Data Integrity  
Network Translation

### SIGNIFICANT IMPROVEMENTS<sup>1</sup>

Throughput | Jitter | Latency

28G-112G  
per channel



PCIe Gen 4/5



## INFRASTRUCTURE OFFLOAD

### OPTIMIZED ARCHITECTURE

vSwitch | vRouter | Security

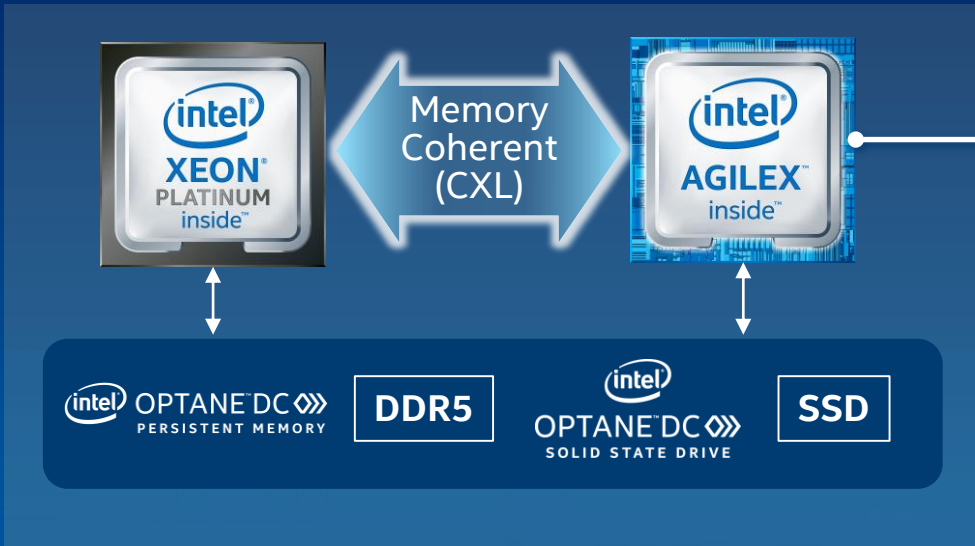
### SMALL FORM FACTOR & LOW POWER

Wide Range of Servers

**First FPGA Providing Flexibility and Agility From 100Gbps to 1Tbps**

<sup>1</sup> Compared to Intel® Stratix® 10 FPGAs

# Converged Workload Acceleration for the Datacenter



## INFRASTRUCTURE ACCELERATION

Network | Security | Remote Memory Access

## APPLICATION ACCELERATION

AI | Search | Video Transcode | Database  
40 TFLOPs of DSP Performance<sup>1</sup>

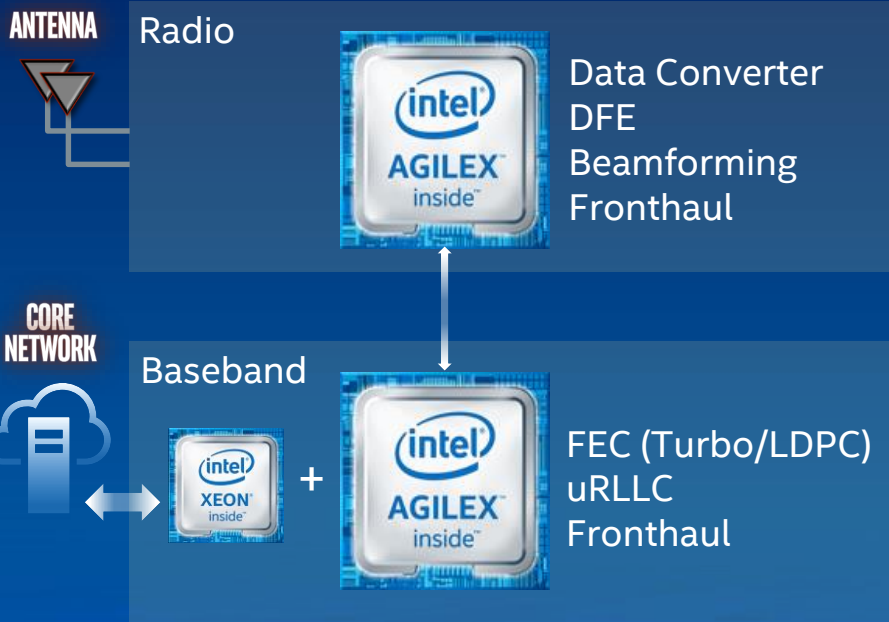
## STORAGE ACCELERATION

Compression | Decompression | Encryption  
Memory Hierarchy Management

**First FPGA with Comprehensive Memory Support and Coherent Attach to Intel® Xeon® Scalable Processors**

<sup>1</sup> with FP16 configuration, based on current estimates, see slide 180 for details

# Agility & Flexibility for All Stages of 5G Implementation



## CUSTOM LOGIC CONTINUUM

### FPGA FLEXIBILITY

High Flexibility | Short Time-to-Market

### RAPID INTEL® eASIC™ DEVICE OPTIMIZATION

Power & Cost Optimization

### FULL CUSTOM ASIC OPTIMIZATION

Best Power<sup>1</sup> | Best Performance<sup>1</sup> | Best Cost<sup>1</sup>

### APPLICATION-SPECIFIC TILE OPTIONS

Data Converter | Vector Engine | Custom Compute

**Only Intel® Provides All of These Customization Options**

<sup>1</sup> Compared to the other customization options shown

# Intel® Agilex™ FPGAs for the Data-Centric World

Intel® Agilex™ FPGAs for transformative applications in edge computing, embedded, networking (5G/NFV) and data centers



Any-to-Any integration enables Intel® FPGAs with application-specific optimization and customization, delivering new levels of flexibility & agility

First FPGAs leveraging Intel's unmatched innovation:  
10nm process, 3D integration, Intel® Xeon® Scalable Processor memory coherency (CXL), 112G XCVRs, PCIe\* Gen 5, Intel® eASIC™ devices, One API, Intel® Optane™ DC Persistent Memory support

# AGENDA

- Intel® and AI / Machine Learning
- Accelerate Deep Learning Using OpenVINO Toolkit
- Deep Learning Acceleration with FPGA
  - FPGAs and Machine Learning
  - Intel® FPGA Deep Learning Acceleration Suite
  - Execution on the FPGA (Model Optimizer & Inference Engine)
- Intel® Agilex® FPGA
- **OneAPI**



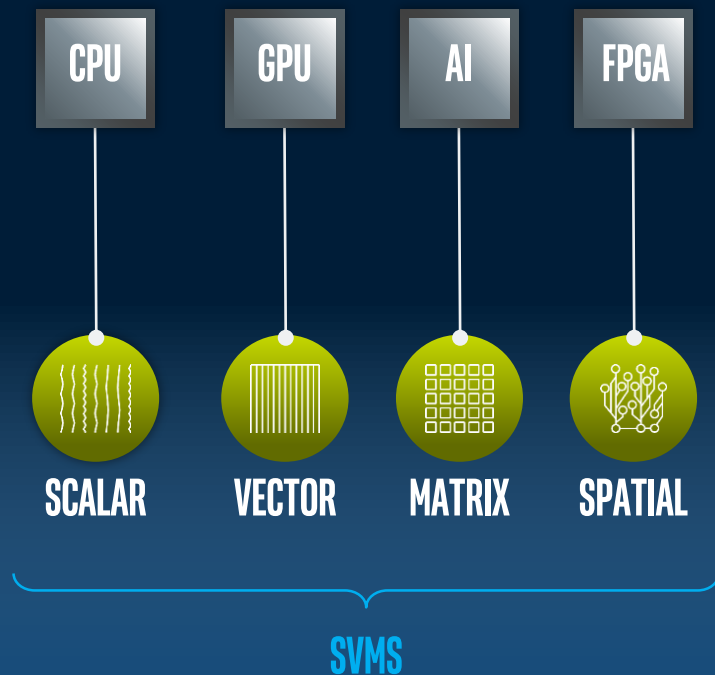


# ONEAPI

Single Programming Model  
to Deliver Cross-Architecture Performance

# DIVERSE WORKLOADS REQUIRE DIVERSE ARCHITECTURES

The future is a **diverse** mix of scalar, vector, matrix, and spatial **architectures** deployed in CPU, GPU, AI, FPGA and other accelerators





# INTEL'S ONEAPI CORE CONCEPT

**Project oneAPI** delivers a unified programming model to simplify development across diverse architectures

Common developer experience across Scalar, Vector, Matrix and Spatial (SVMS) architecture

Unified and simplified language and libraries for expressing parallelism

Uncompromised native high-level language performance

Support for CPU, GPU, AI and FPGA

Based on industry standards and open specifications

oneAPI  
Tools

Optimized Applications

Optimized  
Middleware / Frameworks

oneAPI Language & Libraries

CPU

SCALAR

GPU

VECTOR

AI

MATRIX

FPGA

SPATIAL

# ONEAPI FOR CROSS-ARCHITECTURE PERFORMANCE

Optimized Applications

Optimized Middleware & Frameworks

oneAPI Product

Direct Programming

Data  
Parallel  
C++

API-Based Programming

Math

Threading

DPC++ Library

Analytics/ML

DNN

ML Comm

Video Processing

Analysis &  
Debug Tools  
VTune™  
Advisor  
Debugger

Compatibility  
Tool

CPU

SCALAR

GPU

VECTOR

AI

MATRIX

FPGA

SPATIAL

Some capabilities may differ per architecture.



**THANK YOU**



# APPENDIX A – INTEL MACHINE LEARNING TOOLS

## UNIFIED BIG DATA ANALYTICS + AI OPEN SOURCE PLATFORM



|                               |                                                                                                                                                                                                                                                              |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reference Use Cases           | <ul style="list-style-type: none"><li>Anomaly detection, sentiment analysis, fraud detection, image generation, chatbot, sequence prediction, etc.</li></ul>                                                                                                 |
| Built-In Deep Learning Models | <ul style="list-style-type: none"><li>Image classification, object detection, text classification, recommendations, GANs, Sequence to Sequence, etc.</li></ul>                                                                                               |
| Feature Engineering           | Feature transformations for <ul style="list-style-type: none"><li>Image, text, 3D imaging, time series, speech, etc.</li></ul>                                                                                                                               |
| High-Level Pipeline APIs      | <ul style="list-style-type: none"><li>Distributed Tensorflow* &amp; Keras* on Apache Spark/BigDL</li><li>Support for Autograd*, transfer learning, Spark DataFrame and ML Pipeline</li><li>Model serving API for model serving/inference pipelines</li></ul> |
| Backends                      | Apache Spark, TensorFlow*, BigDL, Python, etc.                                                                                                                                                                                                               |

## Build E2E Analytics & AI Applications for Big Data at Scale

# INTEL DISTRIBUTION FOR PYTHON\*



[software.intel.com/intel-distribution-for-python](https://software.intel.com/intel-distribution-for-python)

For developers using the most popular and fastest growing programming language for AI

## Easy, Out-of-the-box Access to High Performance Python\*

- Prebuilt, delivers faster, close-to-native code application performance for machine learning and scientific computing from device to cloud
- Drop in replacement for your existing Python (no code changes required)

## Drive Performance with Multiple Optimization Techniques

- Accelerated NumPy/SciPy/Scikit-Learn with Intel® Math Kernel Library
- Data analytics with pyDAAL, enhanced thread scheduling with Threading Building Blocks, Jupyter\* Notebook interface, Numba, Cython
- Scale easily with optimized MPI4Py and Jupyter notebooks

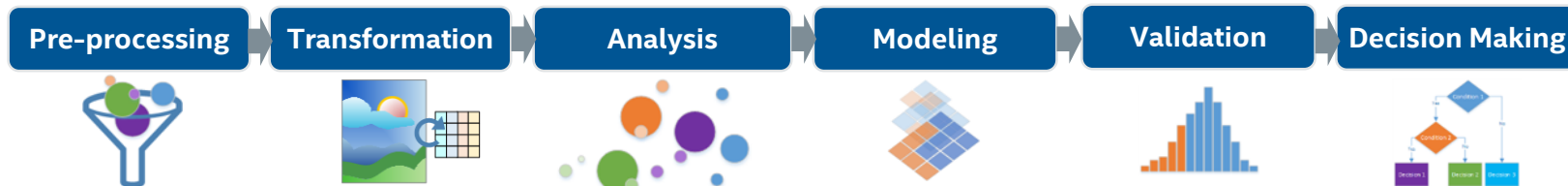
## Faster Access to Latest Optimizations for Intel® Architecture

- Distribution and individual optimized packages available through conda\* and Anaconda Cloud\*
- Optimizations upstreamed back to main Python trunk

Advancing Python Performance Closer to Native Speeds

# INTEL® DATA ANALYTICS ACCELERATION LIBRARY (INTEL® DAAL)

Building blocks for all data analytics stages, including data preparation, data mining & machine learning



Common Python\*, Java\*, C++ APIs across all Intel hardware

Optimizes data ingestion & algorithmic compute together for highest performance

Supports offline, streaming & distributed usage models for a range of application needs

Flexible interfaces to leading big data platforms including Spark\*

Split analytics workloads between edge devices & cloud to optimize overall application throughput

## High Performance Machine Learning & Data Analytics Library

Open Source, Apache\* 2.0 License

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

© 2019 Intel Corporation [Optimization Notice](#)

# INTEL<sup>®</sup> MATH KERNEL FOR DEEP NEURAL NETWORKS (INTEL<sup>®</sup> MKL-DNN)

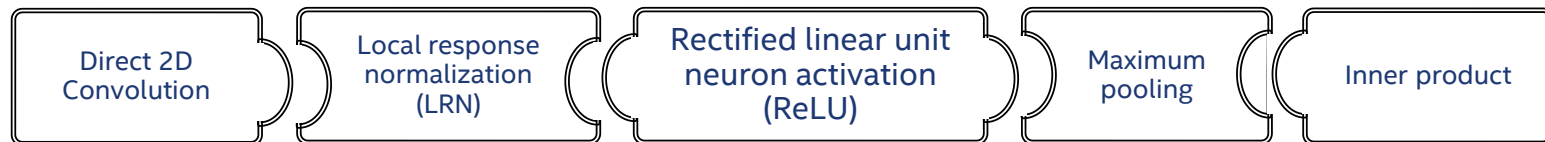
For developers of deep learning frameworks featuring optimized performance on Intel hardware

## Distribution Details

- Open Source
- Apache\* 2.0 License
- Common DNN APIs across all Intel hardware.
- Rapid release cycles, iterated with the DL community, to best support industry framework integration.
- Highly vectorized & threaded for maximal performance, based on the popular Intel<sup>®</sup> Math Kernel Library.

[github.com/01org/mkl-dnn](https://github.com/01org/mkl-dnn)

Examples:



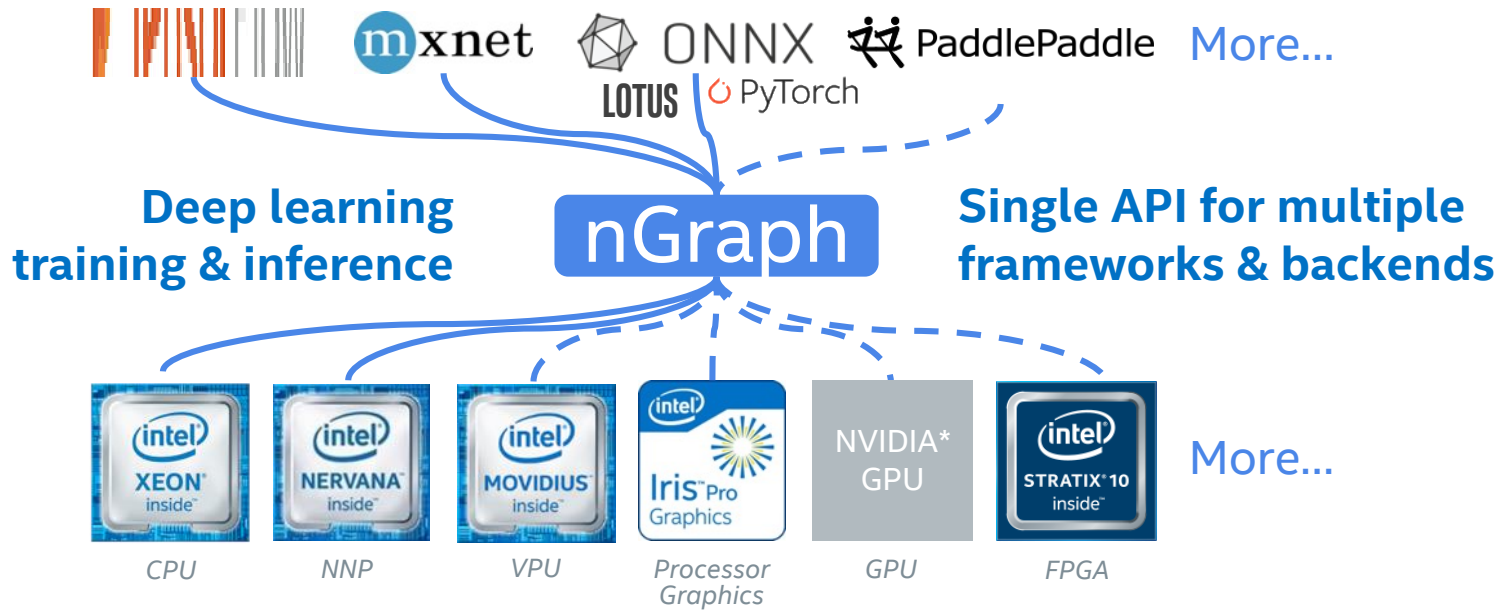
## Accelerate Performance of Deep Learning Models



NOW IN BETA!

--- Work in progress

# INTEL<sup>®</sup> NGRAPH<sup>™</sup> COMPILER



Open-source C++ library, compiler & runtime for deep learning enabling flexibility to run models across a variety of frameworks & hardware

\*Other names and brands may be claimed as the property of others. All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

[github.com/NervanaSystems/ngraph](https://github.com/NervanaSystems/ngraph)

NOW IN BETA!

# NAUTA



**BUILD**

Multi-user collaboration

Interactive sessions

Template functionality

**TRAIN**

Fast training

Batch training

Experiment tracking

Multi-node distribution

Analytics & visualization using TensorBoard\*

**EVALUATE**

Batch inference

Inference deployment

Export to edge devices



Open-Source Distributed Deep Learning Platform for Kubernetes\*

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

\* Other names and brands may be claimed as the property of others.

# APPENDIX B – BONUS MACHINE LEARNING SLIDES

# UNSUPERVISED LEARNING EXAMPLE

MACHINE LEARNING

Regression

Classification

Clustering

Decision Trees

Data Generation

Image Processing

DEEP LEARNING

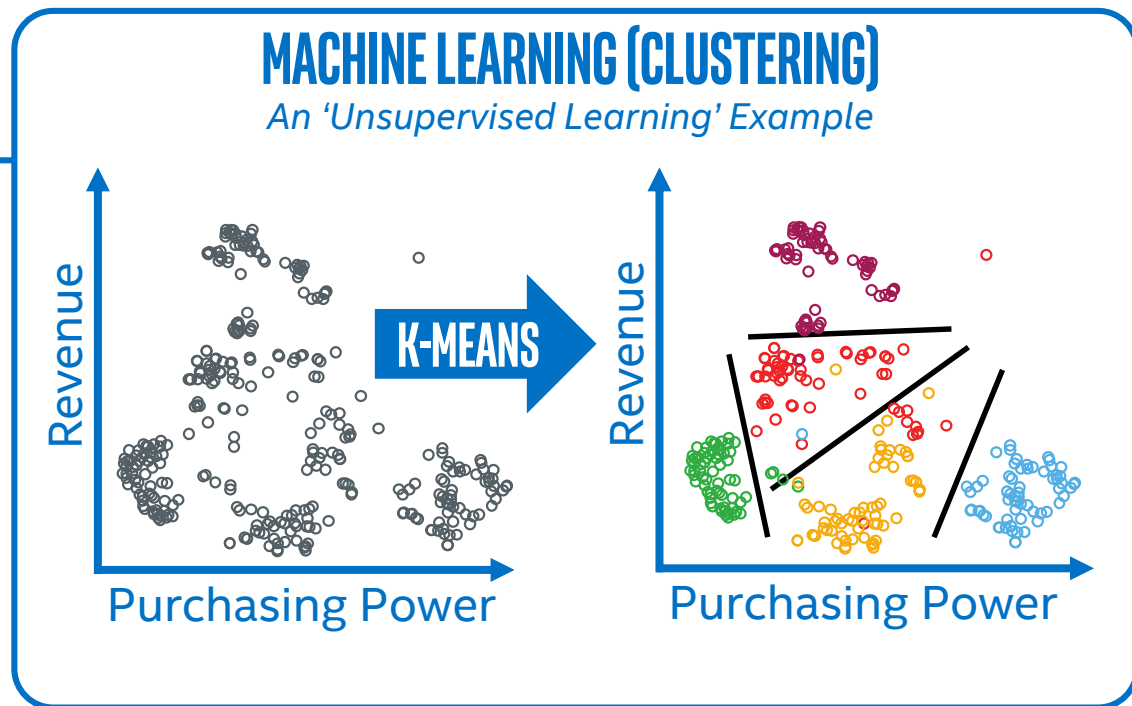
Speech Processing

Natural Language Processing

Recommender Systems

Adversarial Networks

Reinforcement Learning



Choose the right AI approach for your challenge

# SUPERVISED LEARNING EXAMPLE

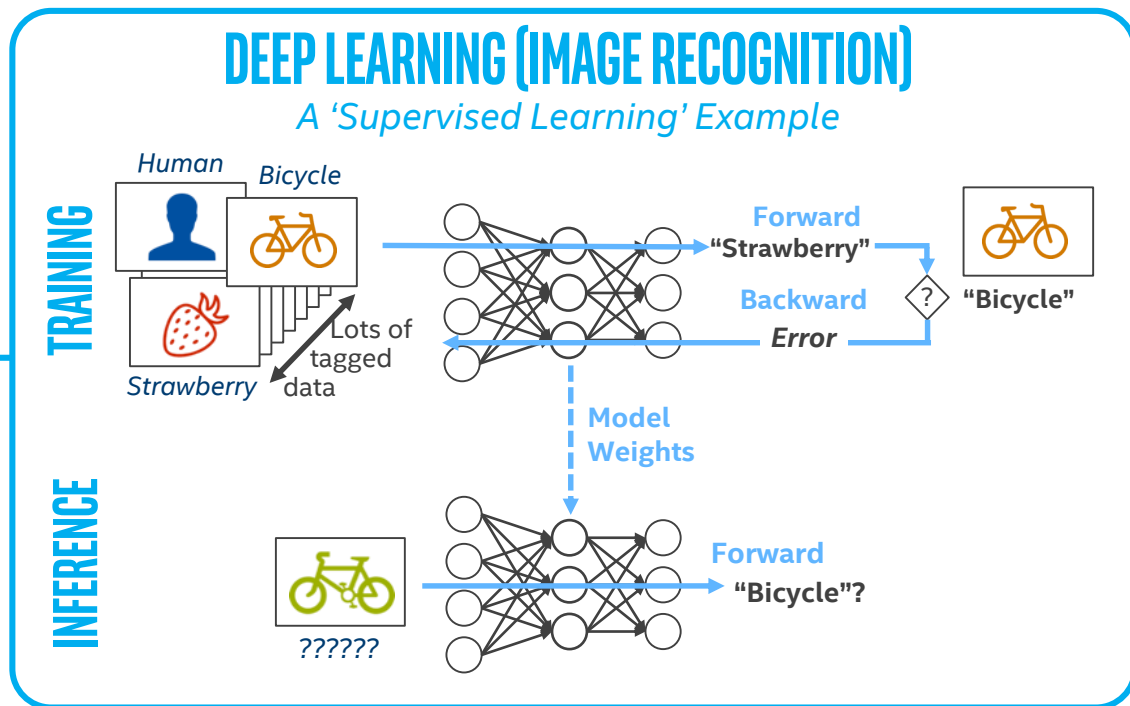
MACHINE LEARNING

- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation

Image Processing

DEEP LEARNING

- Speech Processing
- Natural Language Processing
- Recommender Systems
- Adversarial Networks
- Reinforcement Learning



Choose the right AI approach for your challenge

# REINFORCEMENT LEARNING EXAMPLE

MACHINE LEARNING

- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation

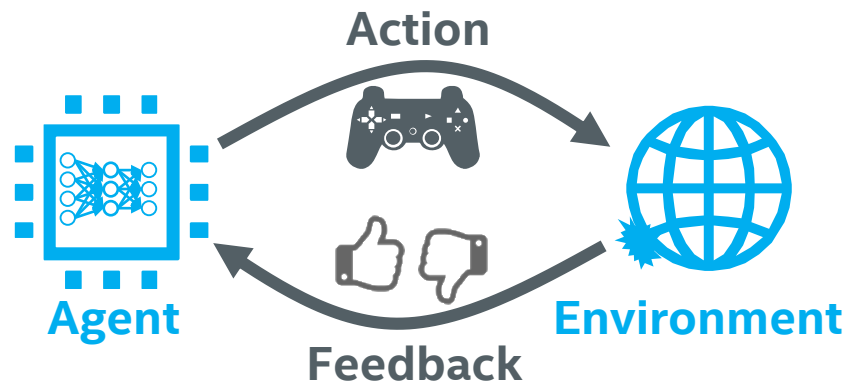
DEEP LEARNING

- Speech Processing
- Natural Language Processing
- Recommender Systems
- Adversarial Networks

Reinforcement Learning

## DEEP LEARNING (REINFORCEMENT LEARNING)

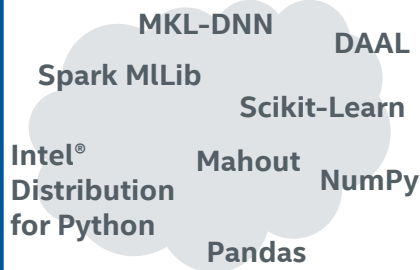
*A 'Deep Reinforcement Learning' Example*



Choose the right AI approach for your challenge

# DEEP LEARNING GLOSSARY

## LIBRARY



Optimized primitive functions for AI

## FRAMEWORK



Open-source development environments

## TOPOLOGY



Specific neural network implementations

## CONTAINER



Pre-configured AI environments ready to deploy

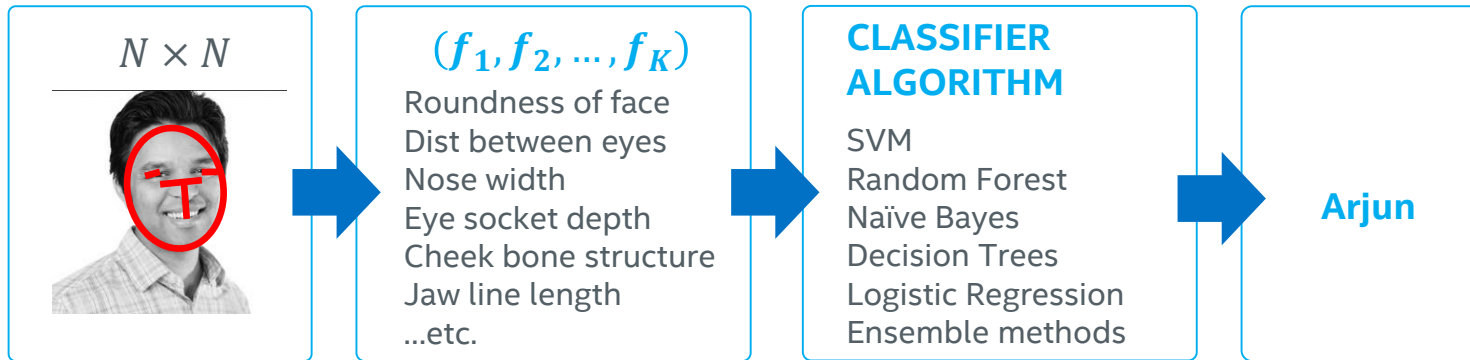
Learn more about “what is AI” at [software.intel.com/ai/course](https://software.intel.com/ai/course)



# MACHINE VS. DEEP LEARNING

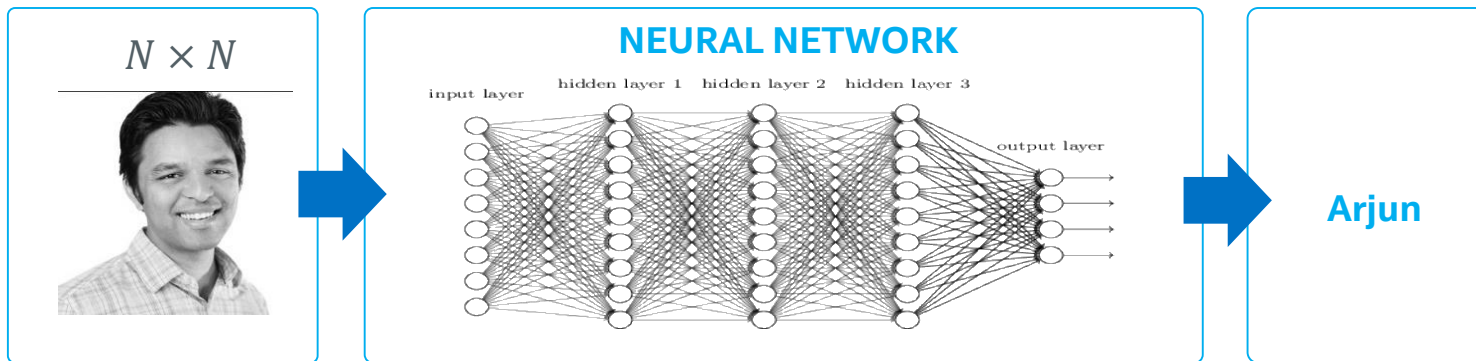
## MACHINE LEARNING

How do you engineer the best features?

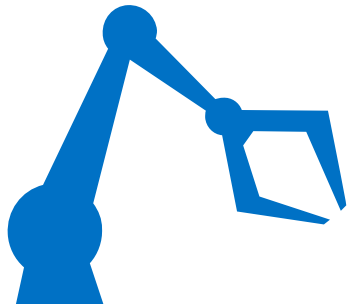


## DEEP LEARNING

How do you guide the model to find the best features?







# WHICH APPROACH IS RIGHT?

A large **manufacturer** uses data to improve their operations, with each challenge using a different approach to deliver maximum business value at the lowest possible cost

| CHALLENGE                               | BEST APPROACH                                       | APPROACH                            | ANSWER                                                             |
|-----------------------------------------|-----------------------------------------------------|-------------------------------------|--------------------------------------------------------------------|
| How many widgets should we manufacture? | Analyze historical supply/demand                    | Analytics/<br>Business Intelligence | 10,000                                                             |
| What will our yield be?                 | Algorithm that correlates many variables to yield   | Statistical/<br>Machine Learning    | At current conditions, yield will be at 90% with 10% loss expected |
| Which widgets have visual defects?      | Algorithm that learns to identify defects in images | Deep Learning                       | Widget 1003, Widget 1094 ...                                       |

LEARN  
MORE IN  
THE NEXT  
SLIDES

# AI CLOSER LOOK



## DEEP LEARNING

*Neural networks used to infer meaning from large dense datasets*



## REASONING

*Hybrid of analytics & AI techniques designed to find meaning in diverse datasets*



## MACHINE LEARNING

*Algorithms designed to deliver better insight with more data*

**Regression** (Linear/Logistic)

**Classification** (Support Vector Machines/SVM, Naïve Bayes)

**Clustering** (Hierarchical, Bayesian, K-Means, DBSCAN)

**Decision Trees** (RandomForest)

**Extrapolation** (Hidden Markov Models/HMM)

**More...**

**Image Recognition** (Convolutional Neural Networks/CNN, Single-Shot Detector/SSD)

**Speech Recognition** (Recurrent Neural Network/RNN)

**Natural Language Processing** (Long-Short Term Memory/LSTM)

**Data Generation** (Generative Adversarial Networks/GAN)

**Recommender System** (Multi-Layer Perceptron/MLP)

**Time-Series Analysis** (LSTM, RNN)

**Reinforcement Learning** (CNN, RNN)

**More...**

**Associative Memory** (Intel® Saffron AI memory base)

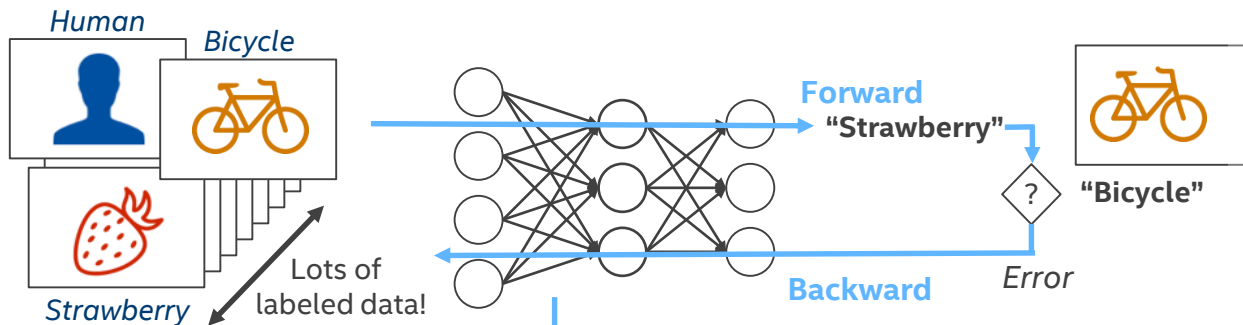
← **See also:** machine & deep learning techniques

**More...**

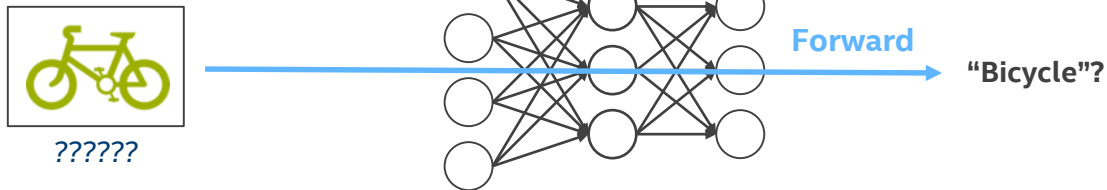
# DEEP LEARNING BASICS



## TRAINING

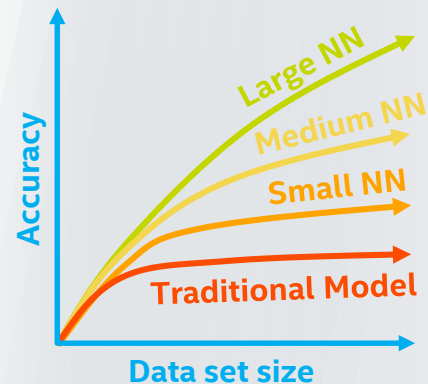


## INFERENCE



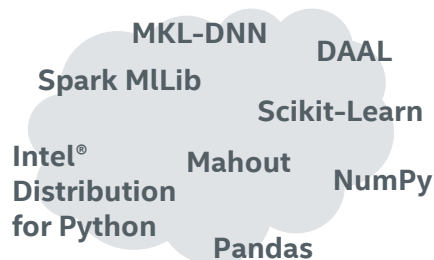
## DID YOU KNOW?

Training with a large data set AND deep (many layered) neural network often leads to the highest accuracy inference



# DEEP LEARNING GLOSSARY

## LIBRARY



Hardware-optimized mathematical and other primitive functions that are commonly used in machine & deep learning algorithms, topologies & frameworks

## FRAMEWORK



Open-source software environments that facilitate deep learning model development & deployment through built-in components and the ability to customize code

## TOPOLOGY



Wide variety of algorithms modeled loosely after the human brain that use neural networks to recognize complex patterns in data that are otherwise difficult to reverse engineer

Translating common deep learning terminology

# DEEP LEARNING USAGES & KEY TOPOLOGIES

## Image Recognition

Resnet-50  
Inception V3  
MobileNet  
SqueezeNet



## Object Detection

R-FCN  
Faster-RCNN  
Yolo V2  
SSD-VGG16, SSD-MobileNet



## Image Segmentation

Mask R-CNN



## Language Translation

GNMT

## Text to Speech

Wavenet

## Recommendation System

Wide & Deep, NCF



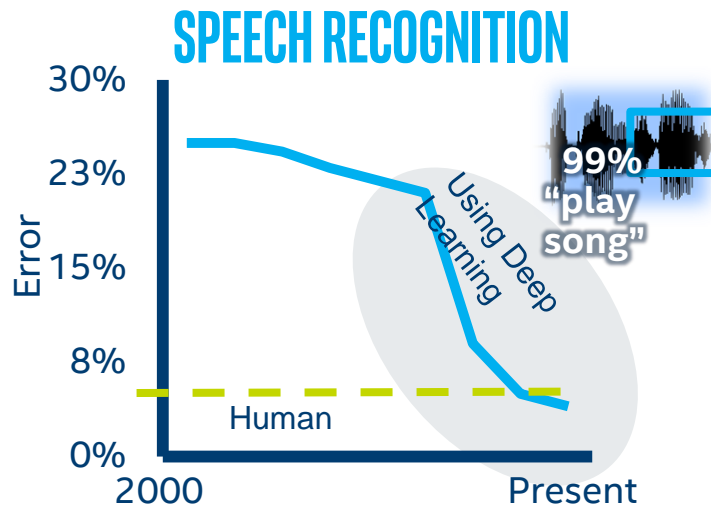
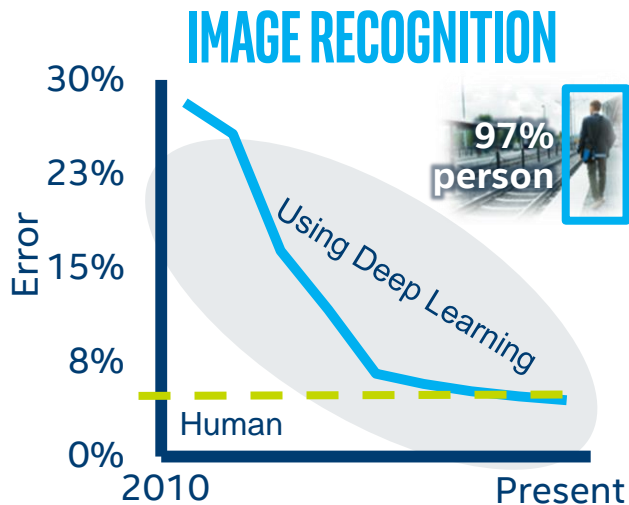
Understand Legalese



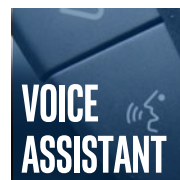
There are many deep learning usages and topologies for each

# DEEP LEARNING BREAKTHROUGHS

Machines able to meet or exceed human image & speech recognition



e.g.



Source: ILSVRC ImageNet winning entry classification error rate each year 2010-2016 (Left), <https://www.microsoft.com/en-us/research/blog/microsoft-researchers-achieve-new-conversational-speech-recognition-milestone/> (Right)

# HARDWARE

Multi-purpose to purpose-built  
AI compute from cloud to device



## MAINSTREAM

## INTENSIVE

DEEP  
LEARNING

TRAINING

INFERENCE



MOST  
OTHER AI



All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

# ONE SIZE DOES NOT FIT ALL

# HARDWARE

Multi-purpose to purpose-built  
AI compute from device to cloud



## END POINT



User-touch end point devices with lower power requirements such as laptops, tablets, smart home devices, drones

Varies to <1ms

## EDGE



Small scale data centers, small business IT infrastructure, to few on-premise server racks and workstations

<5ms

<10-40ms

## DATA CENTER



Large scale data centers such as public cloud or comms service providers, gov't and academia, large enterprise IT

~100ms

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

# AI IS EXPANDING





# HARDWARE

Multi-purpose to purpose-built  
AI compute from device to cloud



## END POINT



**IOT SENSORS**  
(Security, home, retail, industrial...)



Vision & Inference

Speech



### SELF-DRIVING VEHICLES



Autonomous Driving



### DESKTOP & MOBILITY



SOC  
Display, Video, AR/VR, Gestures, Speech

M.2 Card

## EDGE

**SERVERS, APPLIANCES & GATEWAYS**



Most Use Cases

+ Special Purpose



Dedicated Media & Vision Inference



Latency-Bound Inference



Basic Inference, Media & Vision

## DATA CENTER

**SERVERS & APPLIANCES**



Most Use Cases



NNP-T  
Most Intensive Use Cases

+ Special Purpose



Flexible & Memory Bandwidth-Bound Use Cases

Varies to <1ms      <5ms      <10-40ms      ~100ms

<sup>1</sup>GNA=Gaussian Neural Accelerator  
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice. Images are examples of intended © 2019 Intel Corporation. list.

# ONE SIZE DOES NOT FIT ALL





## Bring Your AI Vision to Life Using Our Complete Portfolio

### DATA

Intel analytics ecosystem to get your data ready

### SOLUTIONS

Partner ecosystem to facilitate AI in finance, health, retail, industrial & more

### TOOLS

Software to accelerate development and deployment of real solutions

### HARDWARE

Multi-purpose to purpose-built AI compute from cloud to device

### FUTURE

Driving AI forward through R&D, investments and policy



# APPENDIX C – CONFIGURATION DETAILS

# Configurations for Continued Innovation Driving DL Gains on Xeon® (March 2019)

**1x inference throughput improvement on Intel® Xeon® Platinum 8180 processor (July 2017) baseline :** Tested by Intel as of July 11<sup>th</sup> 2017: Platform: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86\_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC). **Performance measured with:** Environment variables: KMP\_AFFINITY='granularity=fine, compact', OMP\_NUM\_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Caffe: (<http://github.com/intel/caffe/>), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward\_only" command, training measured with "caffe time" command. For "ConvNet" topologies, synthetic dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from [https://github.com/intel/caffe/tree/master/models/intel\\_optimized\\_models](https://github.com/intel/caffe/tree/master/models/intel_optimized_models) (ResNet-50), and [https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet\\_winners](https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners) (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

**5.7x inference throughput improvement on Intel® Xeon® Platinum 8180 processor (December 2018) with continued optimizations :** Tested by Intel as of November 11<sup>th</sup> 2018 :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.46GB (12slots / 32 GB / 2666 MHz). CentOS Linux-7.3.1611-Core, kernel: 3.10.0-862.3.3.el7.x86\_64, SSD sda RS3WC080 HDD 744.1GB, sdb RS3WC080 HDD 1.5TB, sdc RS3WC080 HDD 5.5TB , Deep Learning Framework Intel® Optimization for caffe version: 551a53d63a6183c233abaa1a19458a25b672ad41 Topology::ResNet\_50\_v1 BIOS:SE5C620.86B.00.01.0014.070920180847 MKLDNN: 4e333787e0d66a1dca1218e99a891d493dbc8ef1 instances: 2 instances socket:2 (Results on Intel® Xeon® Scalable Processor were measured running multiple instances of the framework. Methodology described here: <https://software.intel.com/en-us/articles/boosting-deep-learning-training-inference-performance-on-xeon-and-xeon-phi>) Synthetic data. Datatype: INT8 Batchsize=64 vs Tested by Intel as of July 11<sup>th</sup> 2017:2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86\_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC). **Performance measured with:** Environment variables: KMP\_AFFINITY='granularity=fine, compact', OMP\_NUM\_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Caffe: (<http://github.com/intel/caffe/>), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward\_only" command, training measured with "caffe time" command. For "ConvNet" topologies, synthetic dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from [https://github.com/intel/caffe/tree/master/models/intel\\_optimized\\_models](https://github.com/intel/caffe/tree/master/models/intel_optimized_models) (ResNet-50). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

**14x inference throughput improvement on Intel® Xeon® Platinum 8280 processor with Intel® DL Boost:** Tested by Intel as of 2/20/2019. 2 socket Intel® Xeon® Platinum 8280 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode: 0x200004d), Ubuntu 18.04.1 LTS, kernel 4.15.0-45-generic, SSD 1x sda INTEL SSDSC2BA80 SSD 745.2GB, nvme1n1 INTEL SSDPE2KX040T7 SSD 3.7TB, Deep Learning Framework: Intel® Optimization for Caffe version: 1.1.3 (commit hash: 7010334f159da247db3fe3a9d96a3116ca06b09a) , ICC version 18.0.1, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a, model: [https://github.com/intel/caffe/blob/master/models/intel\\_optimized\\_models/int8/resnet50\\_int8\\_full\\_conv.prototxt](https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt), BS=64, syntheticData, 4 instance/2 socket, Datatype: INT8 vs Tested by Intel as of July 11<sup>th</sup> 2017: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86\_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC). **Performance measured with:** Environment variables: KMP\_AFFINITY='granularity=fine, compact', OMP\_NUM\_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Caffe: (<http://github.com/intel/caffe/>), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward\_only" command, training measured with "caffe time" command. For "ConvNet" topologies, synthetic dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from [https://github.com/intel/caffe/tree/master/models/intel\\_optimized\\_models](https://github.com/intel/caffe/tree/master/models/intel_optimized_models) (ResNet-50),. Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

**30x inference throughput improvement on Intel® Xeon® Platinum 9282 processor with Intel® DL Boost :** Tested by Intel as of 2/26/2019. Platform: Dragon rock 2 socket Intel® Xeon® Platinum 9282(56 cores per socket), HT ON, turbo ON, Total Memory 768 GB (24 slots/ 32 GB/ 2933 MHz), BIOS:SE5C620.86B.0D.01.0241.112020180249, Centos 7 Kernel 3.10.0-957.5.1.el7.x86\_64, Deep Learning Framework: Intel® Optimization for Caffe version: <https://github.com/intel/caffe/d554cbf1>, ICC 2019.2.187, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a), model: [https://github.com/intel/caffe/blob/master/models/intel\\_optimized\\_models/int8/resnet50\\_int8\\_full\\_conv.prototxt](https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt), BS=64, No datalayer syntheticData:3x224x224, 56 instance/2 socket, Datatype: INT8 vs Tested by Intel as of July 11<sup>th</sup> 2017: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86\_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC). **Performance measured with:** Environment variables: KMP\_AFFINITY='granularity=fine, compact', OMP\_NUM\_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Caffe: (<http://github.com/intel/caffe/>), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward\_only" command, training measured with "caffe time" command. For "ConvNet" topologies, synthetic dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from [https://github.com/intel/caffe/tree/master/models/intel\\_optimized\\_models](https://github.com/intel/caffe/tree/master/models/intel_optimized_models) (ResNet-50),. Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

# CONFIGURATION DETAILS

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at [intel.com](http://intel.com), or from the OEM or retailer. No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit <http://www.intel.com/performance>.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Intel inside, the Intel inside logo, Xeon, the Xeon logo, Xeon Phi, the Xeon Phi logo, Core, the Core logo, Atom, the Atom logo, Movidius, the Movidius logo, Stratix, the Stratix logo, Arria, the Arria logo, Myriad, Nervana and others are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

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.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

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

© 2019 Intel Corporation.

# CONFIGURATION DETAILS (CONT'D)

## Configuration: AI Performance – Software + Hardware

INFERENCE using FP32 Batch Size Caffe GoogleNet v1 128 AlexNet 256.

The benchmark results 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. 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 <http://www.intel.com/performance> Source: Intel measured as of June 2017 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.

### Configurations for Inference throughput

Platform :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.28GB (12slots / 32 GB / 2666 MHz),4 instances of the framework, CentOS Linux-7.3.1611-Core , SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework caffe version: a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology:GoogleNet v1 BIOS:SE5C620.86B.00.01.0004.071220170215 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 1449.9 imgs/sec vs Platform: 2S Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz (18 cores), HT enabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 64GB DDR4-2133 ECC RAM. BIOS: SE5C610.86B.01.01.0024.021320191901, CentOS Linux-7.5.1804(Core) kernel 3.10.0-862.3.2.el7.x86\_64, SSD sdb INTEL SSDSC2BW24 SSD 223.6GB. Framework BVLC-Caffe: <https://github.com/BVLC/caffe>, Inference & Training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. BVLC Caffe (<http://github.com/BVLC/caffe>), revision 2a1c552b66f026c7508d390b526f2495ed3be594

### Configuration for training throughput:

Platform :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.28GB (12slots / 32 GB / 2666 MHz),4 instances of the framework, CentOS Linux-7.3.1611-Core , SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework caffe version: a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology:alexnet BIOS:SE5C620.86B.00.01.0004.071220170215 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 1257 imgs/sec vs Platform: 2S Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz (18 cores), HT enabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 64GB DDR4-2133 ECC RAM. BIOS: SE5C610.86B.01.01.0024.021320191901, CentOS Linux-7.5.1804(Core) kernel 3.10.0-862.3.2.el7.x86\_64, SSD sdb INTEL SSDSC2BW24 SSD 223.6GB. Framework BVLC-Caffe: <https://github.com/BVLC/caffe>, Inference & Training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. BVLC Caffe (<http://github.com/BVLC/caffe>), revision 2a1c552b66f026c7508d390b526f2495ed3be594

# CONFIGURATION DETAILS (CONT'D)

INFERENCE using FP32 Batch Size Caffe GoogleNet v1 128 AlexNet 256.

The benchmark results 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. 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 <http://www.intel.com/performance> Source: Intel measured as of June 2017 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.

## Configurations for Inference throughput

Platform :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.28GB (12slots / 32 GB / 2666 MHz),4 instances of the framework, CentOS Linux-7.3.1611-Core , SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework caffe version: a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology:GoogleNet v1 BIOS:SE5C620.86B.00.01.0004.071220170215 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 1449.9 imgs/sec vs Platform: 2S Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz (18 cores), HT enabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 64GB DDR4-2133 ECC RAM. BIOS: SE5C610.86B.01.01.0024.021320181901, CentOS Linux-7.5.1804(Core) kernel 3.10.0-862.3.2.el7.x86\_64, SSD sdb INTEL SSDSC2BW24 SSD 223.6GB. Framework BVLC-Caffe: <https://github.com/BVLC/caffe>, Inference & Training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. BVLC Caffe (<http://github.com/BVLC/caffe>), revision 2a1c552b66f026c7508d390b526f2495ed3be594

## Configuration for training throughput:

Platform :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.28GB (12slots / 32 GB / 2666 MHz),4 instances of the framework, CentOS Linux-7.3.1611-Core , SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework caffe version: a3d5b022fe026e9092fc7abc765b1162ab9940d Topology:alexnet BIOS:SE5C620.86B.00.01.0004.071220170215 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 1257 imgs/sec vs Platform: 2S Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz (18 cores), HT enabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 64GB DDR4-2133 ECC RAM. BIOS: SE5C610.86B.01.01.0024.021320181901, CentOS Linux-7.5.1804(Core) kernel 3.10.0-862.3.2.el7.x86\_64, SSD sdb INTEL SSDSC2BW24 SSD 223.6GB. Framework BVLC-Caffe: <https://github.com/BVLC/caffe>, Inference & Training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. BVLC Caffe (<http://github.com/BVLC/caffe>), revision 2a1c552b66f026c7508d390b526f2495ed3be594



# CONFIGURATION DETAILS (CONT'D)

## 1.4x training throughput improvement in August 2018:

Tested by Intel as of measured August 2<sup>nd</sup> 2018. Processor: 2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.46GB (12slots / 32 GB / 2666 MHz). CentOS Linux-7.3.1611-Core kernel 3.10.0-693.11.6.el7.x86\_64, SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework Intel® Optimizations for caffe version:a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology::resnet\_50 BIOS:SE5C620.86B.00.01.0013.030920180427 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 123 imgs/sec vs Intel tested July 11th 2017 Platform: Platform: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86\_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC).Performance measured with: Environment variables: KMP\_AFFINITY='granularity=fine, compact', OMP\_NUM\_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Caffe: (<http://github.com/intel/caffe/>), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward\_only" command, training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from [https://github.com/intel/caffe/tree/master/models/intel\\_optimized\\_models](https://github.com/intel/caffe/tree/master/models/intel_optimized_models) (GoogLeNet, AlexNet, and ResNet-50), [https://github.com/intel/caffe/tree/master/models/default\\_vgg\\_19](https://github.com/intel/caffe/tree/master/models/default_vgg_19) (VGG-19), and [https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet\\_winners](https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners) (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

## 5.4x inference throughput improvement in August 2018:

Tested by Intel as of measured July 26<sup>th</sup> 2018 :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.46GB (12slots / 32 GB / 2666 MHz). CentOS Linux-7.3.1611-Core, kernel: 3.10.0-862.3.3.el7.x86\_64, SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB, Deep Learning Framework Intel® Optimized caffe version:a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology::resnet\_50\_v1 BIOS:SE5C620.86B.00.01.0013.030920180427 MKLDNN: version:464c268e544bae26f9b85a2acb9122c766a4c396 instances: 2 instances socket:2 (Results on Intel® Xeon® Scalable Processor were measured running multiple instances of the framework. Methodology described here: <https://software.intel.com/en-us/articles/boosting-deep-learning-training-inference-performance-on-xeon-and-xeon-phi>) NoDataLayer. Datatype: INT8 Batchsize=64 Measured: 1233.39 imgs/sec vs Tested by Intel as of July 11<sup>th</sup> 2017:2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86\_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC).Performance measured with: Environment variables: KMP\_AFFINITY='granularity=fine, compact', OMP\_NUM\_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Caffe: (<http://github.com/intel/caffe/>), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward\_only" command, training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from [https://github.com/intel/caffe/tree/master/models/intel\\_optimized\\_models](https://github.com/intel/caffe/tree/master/models/intel_optimized_models) (ResNet-50). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

## 11X inference throughput improvement with CascadeLake:

Future Intel Xeon Scalable processor (codename Cascade Lake) results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance vs Tested by Intel as of July 11<sup>th</sup> 2017: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel\_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86\_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC).Performance measured with: Environment variables: KMP\_AFFINITY='granularity=fine, compact', OMP\_NUM\_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Caffe: (<http://github.com/intel/caffe/>), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward\_only" command, training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from [https://github.com/intel/caffe/tree/master/models/intel\\_optimized\\_models](https://github.com/intel/caffe/tree/master/models/intel_optimized_models) (ResNet-50). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".



# CONFIGURATION DETAILS (CONT'D)

## Intel® Arria® 10 – 1150 FPGA energy efficiency on Caffe/AlexNet up to 25 img/s/w with FP16 at 297MHz

Vanilla AlexNet Classification Implementation as specified by <http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf>, Training Parameters taken from Caffe open-source Framework are 224x224x3 Input, 1000x1 Output, FP16 with Shared Block-Exponents, All compute layers (incl. Fully Connected) done on the FPGA except for Softmax, Arria 10-1150 FPGA, -1 Speed Grade on Altera PCIe DevKit with x72 DDR4 @ 1333 MHz, Power measured through on-board power monitor (FPGA POWER ONLY), ACDS 16.1 Internal Builds + OpenCL SDK 16.1 Internal Build, Compute machine is an HP Z620 Workstation, Xeon E5-1660 at 3.3 GHz with 32GB RAM. The Xeon is not used for compute.

# DETAILS ON INTEL® AGILEX™ FPGA PERFORMANCE, POWER AND SOFTWARE SUPPORT NUMBERS

## (1) Up to 40% Higher Performance Compared to Intel Stratix 10 FPGAs

Derived from benchmarking an example design suite comparing maximum clock speed (Fmax) achieved in Intel Stratix 10 devices with the Fmax achieved in Intel Agilex devices, using Intel Quartus Prime Software. On average, designs running in the fastest speed grade of Intel Agilex FPGAs achieve a 40% improvement in Fmax compared to the same designs running in the most popular speed grade of Stratix 10 devices (-2 speed grade), tested February 2019.

## (2) Up to 40% Lower Total Power Compared to Intel Stratix 10 FPGAs

Derived from benchmarking an example design suite comparing total power estimates of each design running in Intel Stratix 10 FPGAs compared to the total power consumed by the same design running in Intel Agilex FPGAs. Power estimates of Intel Stratix 10 FPGA designs are obtained from Intel Stratix 10 Early Power Estimator; power estimates for Intel Agilex FPGA designs are obtained using internal Intel analysis and architecture simulation and modeling, tested February 2019.

## (3) Up to 40 TFLOPs of DSP Performance (FP16 Configuration)

Each Intel Agilex DSP block can perform two FP16 floating-point operations (FLOPs) per clock cycle. Total FLOPs for FP16 configuration is derived by multiplying 2x the maximum number of DSP blocks to be offered in a single Intel Agilex FPGA by the maximum clock frequency that will be specified for that block.

## (4) Up to 92 TOPs of DSP Performance (INT8 Configuration)

Each Intel Agilex DSP block can perform eight INT8 operations per clock cycle. Total TOPs for INT8 configuration is derived by multiplying 8x the maximum number of DSP blocks to be offered in a single Intel Agilex FPGA by the maximum clock frequency that will be specified for that block.

## (5) 30% Improvement in Compile Times / 15% Improvement in Memory Utilization

Comparison is made between Intel Quartus Prime Software 18.1 and Intel Quartus Prime 19.1. Derived from benchmarking an example design suite comparing compile times and memory utilization for designs in Intel Quartus Prime Software 18.1 with compile times and memory utilization for same designs in Intel Quartus Prime Software 19.1, tested February 2019.

Results have been estimated or simulated using internal Intel analysis, architecture simulation, and modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

# LEGAL NOTICES & DISCLAIMERS

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at [intel.com](http://intel.com), or from the OEM or retailer. No computer system can be absolutely secure.

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](http://www.intel.com/benchmarks).

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Atom, Xeon and others are trademarks of Intel Corporation in the U.S. and/or other countries. \*Other names and brands may be claimed as the property of others.

© 2019 Intel Corporation.

# LEGAL NOTICES & DISCLAIMERS

## FTC Optimization Notice

If you make any claim about the performance benefits that can be achieved by using Intel software developer tools (compilers or libraries), you must use the entire text on the same viewing plane (slide) as the claim.

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