MACHINE LEARNING AND DEEP LEARNING

MACHINE LEARNING
How do you engineer the best features?

\( N \times N \)

\((f_1, f_2, \ldots, f_K)\)
- Roundness of face
- Distance between eyes
- Nose width
- Eye socket depth
- Cheek bone structure
- Jaw line length
- Etc.

CLASSIFIER ALGORITHM
- SVM
- Random Forest
- Naïve Bayes
- Decision Trees
- Logistic Regression
- Ensemble methods

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

\( N' \times N' \)

NEURAL NETWORK

Séverine

Séverine
Intel Hardware for AI
Flexible AI Acceleration

**CPU only**
Built-in AI acceleration for mainstream AI use cases

**CPU + GPU**
When compute is dominated by AI, HPC, graphics, and/or real-time media

**CPU + custom**
When compute is dominated by deep learning (DL)

Cloud / Data Center: Intel Xeon

Edge: Intel Core i7

Device: Intel Atom

DL Training/Inference: Habana

DL Custom: Intel AGILEX

DL Inference: Intel MOVIDIUS
Intel® Movidius™ VPU and Accelerator Cards

**Built for Edge AI**
- Deep learning inference + computer vision + media
- Faster memory bandwidth
- Groundbreaking high-efficiency architecture
- OpenVINO toolkit enabled

**Flexible Form Factors**

**Edge Experiences**
- Cars
- Robotics
- Elon Musk
- Trucks
- Cloud
- Storage
- Security
Intel® Xeon® Scalable Processors

The Only Data Center CPU with Built-in AI Acceleration
- Intel Advanced Vector Extensions 512
- Intel Deep Learning Boost (Intel DL Boost)
- Intel Optane Persistent Memory

Shipping
Cascade Lake
- New Intel DL Boost (VNNI)
- New memory storage hierarchy

Cooper Lake
- Intel DL Boost (BFLOAT16)

April 2021
Ice Lake
- Intel DL Boost (VNNI) and new Intel Software Guard Extensions (Intel® SGX) that enable new AI use cases like federated learning

2022
Sapphire Rapids
- Intel Advanced Matrix Extensions (AMX) extends built-in AI acceleration capabilities on Xeon Scalable

Leadership performance

The Only Data Center CPU with Built-in AI Acceleration
Intel Deep Learning Boost
A Vector Neural Network Instruction (VNNI)
Extends Intel AVX-512 to Accelerate AI/DL Inference

Combining three instructions into one maximizes the use of compute resources, improves cache utilization and avoids potential bandwidth bottlenecks.

1.74x faster inference performance (BERT) with enhanced Intel DL Boost vs. prior gen\(^1\)

Xe HPC (Ponte Vecchio)
Leadership Performance for Data-level Parallel AI Workloads

>40 active tiles, over 100 billion transistors integrated into a single package

Powering New Phase of SuperMUC-NG at Leibniz Supercomputing Centre (LRZ)

https://www.youtube.com/watch?v=JzbNI0AcwY
Deep Learning ASIC for Training and Inference

Gaudi accelerators coming in AWS EC2 instances in 2021 - will leverage up to 8 Gaudi accelerators and deliver up to 40% better price performance than current GPU-based EC2 instances for training.

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.
AI Software Stack for Intel XPU
oneAPI
One Programming Model for Multiple Architectures & Vendors

Freedom to Make Your Best Choice
- Choose the best accelerated technology the software doesn’t decide for you

Realize all the Hardware Value
- Performance across CPU, GPUs, FPGAs, and other accelerators

Develop & Deploy Software with Peace of Mind
- Open industry standards provide a safe, clear path to the future
- Compatible with existing languages and programming models including C++, Python, SYCL, OpenMP, Fortran, and MPI
Intel’s oneAPI Ecosystem

Built on Intel’s Rich Heritage of CPU Tools Expanded to XPU

**oneAPI**
A cross-architecture language based on C++ and SYCL standards

Powerful libraries designed for acceleration of domain-specific functions

**Powered by oneAPI**
Frameworks and middleware that are built using one or more of the oneAPI industry specification elements, the DPC++ language, and libraries listed on oneapi.com.

Visit [software.intel.com/oneapi](http://software.intel.com/oneapi) for more details.

Some capabilities may differ per architecture and custom-tuning will still be required. Other accelerators to be supported in the future.
Intel oneAPI Software Tools for AI & Analytics

**Intel® oneAPI Toolkits**

**Intel® oneAPI AI Analytics Toolkit**
Accelerate machine learning & data science pipelines with optimized deep learning frameworks & high-performing Python libraries
Data Scientists, AI Researchers, DL/ML Developers

**Intel® oneAPI Base Toolkit**
Incl. Intel® oneAPI Deep Neural Network Library (oneDNN), Intel® oneAPI Collective Communications Library (oneCCL), & Intel® oneAPI Data Analytics Library (oneDAL)
Optimize primitives for algorithms and framework development
DL Framework Developers - Optimize algorithms for Machine Learning & Analytics

**Toolkit Powered by oneAPI**

**Intel® Distribution of OpenVINO™ Toolkit**
Deploy high performance inference & applications from edge to cloud
AI Application, Media, & Vision Developers
The industry needs a programming model where developers can take advantage of an array of innovative hardware architectures. The goal of oneAPI is to provide increased choice of hardware vendors, processor architectures, and faster support of next-generation accelerators. Microsoft has been using oneAPI elements across Intel hardware offerings as part of its initiatives and supports the open standards-based specification. We are excited to support our customers with choice and accelerate the growth of AI and machine learning.

- Tim Harris, Principal Architect, Azure AI, Microsoft

With the growth of AI, machine learning, and data-centric applications, the industry needs a programming model that allows developers to take advantage of rapid innovation in processor architectures. TensorFlow supports the oneAPI industry initiative and its standards-based open specification. oneAPI complements TensorFlow's modular design and provides increased choice of hardware vendor and processor architecture, and faster support of next-generation accelerators. TensorFlow uses oneAPI today on Xeon processors and we look forward to using oneAPI to run on future Intel architectures.
AI Software Stack for Intel XPU

Intel offers a Robust Software Stack to Maximize Performance of Diverse Workloads

**Intel® oneAPI AI Analytics Toolkit**

- Develop DL models in Frameworks, ML & Analytics in Python
- E2E Workloads (Census, NYTaxi, Mortgage...)

**Intel® Low Precision Optimization Toolkit**

- Intel® Low Precision Optimization Toolkit

**Model Zoo for Intel® Architecture**

- Model Zoo for Intel® Architecture

**Intel® oneAPI Base Toolkit**

- Kernel Selection, Write, Customize Kernels

**Intel® oneAPI AI Analytics Toolkit**

- Deploy DL models

**Open Model Zoo**

- Deploy DL models

**Intel® OpenVINO™ Toolkit**

- Model Optimizer & Inference Engine

**DL/ML Tools**

- Intel® OpenVINO™

**DL/ML Middleware & Frameworks**

- Libraries & Compiler

- numba
- pandas
- Modin
- scipy
- daal4Py
- xgboost
- TensorFlow
- PyTorch

**Full Set of Intel oneAPI cross-architecture AI ML & DL Software Solutions**
## Intel® AI Analytics Toolkit

**Powered by oneAPI**

Accelerate end-to-end AI and data analytics pipelines with libraries optimized for Intel® architectures

### Who Uses It?
Data scientists, AI researchers, ML and DL developers, AI application developers

### Top Features/Benefits
- Deep learning performance for training and inference with Intel optimized DL frameworks and tools
- Drop-in acceleration for data analytics and machine learning workflows with compute-intensive Python packages

Learn More: [software.intel.com/oneapi/ai-kit](http://software.intel.com/oneapi/ai-kit)

### Deep Learning
- Intel® Optimization for TensorFlow
- Intel® Optimization for PyTorch
- Intel® Low Precision Optimization Tool
- Model Zoo for Intel® Architecture

### Data Analytics & Machine Learning
- **Accelerated Data Frames**
  - Intel® Distribution of Modin
  - OmniSci Backend
- **Intel® Distribution for Python**
  - XGBoost
  - Scikit-learn
  - NumPy
  - SciPy
  - XGBoost
  - Daal-4Py
  - Pandas

### Samples and End2End Workloads
- Supported Hardware Architectures

### Get the Toolkit
- **HERE** or via these locations
  - Intel Installer
  - Docker
  - Apt, Yum
  - Conda
  - Intel® DevCloud

---

*Hardware support varies by individual tool. Architecture support will be expanded over time. Other names and brands may be claimed as the property of others.*

[Back to Domain-specific Toolkits for Specialized Workloads](#)
Key Features & Benefits - a little teaser

Intel® oneAPI AI Analytics Toolkit

▪ Accelerate end-to-end AI and Data Science pipelines, achieve drop-in acceleration with optimized Python tools built using oneAPI libraries (i.e. oneMKL, oneDNN, oneCCL, oneDAL, and more)

▪ Achieve high-performance deep learning training and inference with Intel-optimized TensorFlow and PyTorch versions, and low-precision optimization with support for fp16, int8 and bfloat16

▪ Expedite development using open source Intel-optimized pre-trained deep learning models for best performance via Model Zoo for Intel® Architecture (IA)

▪ Supports cross-architecture development (Intel® CPUs/GPUs) and compute
# Getting Started with Intel® oneAPI AI Analytics Toolkit

## Overview
- Visit Intel® oneAPI AI Analytics Toolkit (AI Kit) for more details and up-to-date product information
- Release Notes

## Installation
- Download the AI Kit from Intel, Anaconda or any of your favorite package managers
- Get started quickly with the AI Kit Docker Container
- Utilize the Getting Started Guide

## Hands on
- Code Samples
- Build, test and remotely run workloads on the Intel® DevCloud for free. No software downloads. No configuration steps. No installations.

## Learning
- Machine Learning & Analytics Blogs
- Intel AI Blog site
- Webinars & articles

## Support
- Ask questions and share information with others through the Community Forum
- Discuss with experts at AI Frameworks Forum

Download Now
A development sandbox to develop, test and run workloads across a range of Intel CPUs, GPUs, and FPGAs using Intel’s oneAPI software.

Get Up & Running In Seconds!

Sign up at: software.intel.com/devcloud/oneapi
High-Performance Deep Learning Using Intel® Distribution of OpenVINO™ toolkit - Powered by oneAPI

A toolkit for fast, more accurate real-world results using high-performance AI and computer vision inference deployed into production on Intel XPU architectures (CPU, GPU, FPGA, VPU) from edge to cloud

Who needs this product?
AI application developers, OEMs, ISVs, System Integrators, Vision and Media developers

Top Features/Benefits
High-performance, deep learning inference deployment
Streamlined development; ease of use
Write once, deploy anywhere

software.intel.com/openvino-toolkit
# Getting Started with Intel® Distribution of OpenVINO™ Toolkit

## Overview
- Visit Intel® Distribution of OpenVINO Toolkit for more details and up-to-date product information
- [Release Notes](#)

## Installation
- Download the Intel® Distribution of OpenVINO™ toolkit, or get via YUM or APT repositories
- Utilize the [Getting Started Guide](#)

## Hands on
- Understand sample demos and tools included
- Build, test and remotely run workloads on the Intel® DevCloud for the Edge before buying hardware

## Learning
- [Intel AI Blog site](#)
- [Webinars & articles](#)
- Choose hardware option with [Performance Benchmarks](#)

## Support
- Ask questions and share information with others through the [Community Forum](#)

## Download Now
Accelerate Time to Production with Intel® DevCloud for the Edge

See immediate AI Model performance across Intel’s vast array of Edge Solutions

- **Instant, Global Access**
  Run AI applications from anywhere in the world

- **Prototype on the Latest Hardware and Software**
  Develop knowing you’re using the latest Intel technology

- **Benchmark your Customized AI Application**
  Immediate feedback - frames per second, performance

- **Reduce Development Time and Cost**
  Quickly find the right compute for your edge solution

Sign up now for access
AI Containers for Flexibility

- Optimized, validated, deployable AI containers
- Available via Docker containers. Will expand to include Kubernetes orchestrations, Helm charts

**Access from oneContainer Portal**
- Include containers with ready-to-use AI software stacks
- And containers with full AI workloads (including models)

<table>
<thead>
<tr>
<th>Topology</th>
<th>Frameworks</th>
<th>Topology</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLRM</td>
<td>PYT</td>
<td>Mask R-CNN</td>
<td>PYT, TF, OV</td>
</tr>
<tr>
<td>ResNet50</td>
<td>PYT, TF, OV</td>
<td>RNN-T</td>
<td>PYT, TF, OV</td>
</tr>
<tr>
<td>BERT-large</td>
<td>PYT, TF, OV</td>
<td>3D-UNet</td>
<td>TF, OV</td>
</tr>
<tr>
<td>Transformer-LT</td>
<td>PYT, TF</td>
<td>DIEN</td>
<td>TF</td>
</tr>
<tr>
<td>MobileNet-v1</td>
<td>PYT, TF, OV</td>
<td>Wide &amp; Deep</td>
<td>PYT, TF</td>
</tr>
<tr>
<td>SSD-Mobilenet-v1</td>
<td>PYT, TF, OV</td>
<td>RNX101</td>
<td></td>
</tr>
<tr>
<td>SSD-Resnet34</td>
<td>PYT, TF, OV</td>
<td>Yolo-V3</td>
<td>PYT, TF, OV</td>
</tr>
<tr>
<td>WaveNet*</td>
<td>TF</td>
<td>NCF*</td>
<td>TF</td>
</tr>
</tbody>
</table>
Which Toolkit Should I Use
Use Both!

Intel® oneAPI Analytics Toolkit & Intel® Distribution of OpenVINO™ toolkit

Toolkits are complementary to each other and recommendation is to use them both based on your current phase of AI Journey

Data Scientist/ML Developer
Intel® oneAPI AI Analytics Toolkit

- I am exploring and analyzing data; I am developing models
- I want performance and compatibility with frameworks and libraries I use
- I would like to have drop-in acceleration with little to no additional code changes
- I prefer not to learn any new tools or languages

App Developer
Intel® Distribution of OpenVINO™ toolkit

- I am deploying models
- I want leading performance and efficiency across multiple target HW
- I’m concerned about having lower memory footprint, which is critical for deployment
- I am comfortable with learning and adopting a new tool or API to do so

If you prefer working on primitives and to optimize kernels and algorithms directly using oneAPI libraries (oneDNN, oneCCL & oneDAL), then use Intel® oneAPI Base Toolkit
Accrad AI-based Solution Helps Accelerate COVID-19 Diagnosis
Optimized by Intel® oneAPI Analytics Toolkit & Intel® Distribution of OpenVINO™ toolkit

CheXRad helps radiologists and physicians identify COVID-19, viral pneumonia and other diseases on chest X-ray images, and predict the need for ventilators.

- **CheXRad** comes pre-configured with a COVID-19 and viral pneumonia classification neural network.

- To architect, train and validate the neural network, Accrad used **Intel Tensorflow from AI Analytics Toolkit** and the **Intel oneAPI DevCloud** to develop the model.

- To optimize its model for deployment, Accrad used **OpenVINO™ toolkit** and **Intel® DevCloud for Edge**.

- **CheXRad** could classify pathologies in 140 chest x-rays in just **90 seconds** —up to **160x faster** than radiologists, at comparable levels of accuracy, sensitivity and specificity.

Learn more in this [solution brief](#).
Key Takeaways & Call to Action

- Intel toolkits are **FREE**, complementary & work seamlessly together
- They help achieve performance & efficiency across different stages of AI Journey
- Recommend the toolkits based on current phase of customer pipeline

Download the toolkits

- [Intel® oneAPI AI Analytics Toolkit](https://intel.com/products/intel-oneapi-ai-analytics-toolkit)
- [Intel® Distribution of OpenVINO™ toolkit](https://intel.com/products/intel-oneapi-distribution-of-openvinotm)
- [Intel® oneAPI Base Toolkit](https://intel.com/products/intel-oneapi-base-toolkit)

Learn more about Intel® oneAPI Toolkits
[intel.com/oneAPI-AllToolkits](https://intel.com/oneAPI-AllToolkits)
### Intel® oneAPI AI Analytics Toolkit
- Provides performance and easy integration across end-to-end data science pipeline for efficient AI model development
- Maximum compatibility with open source FWKs and Libs with drop-in acceleration that requires minimal to no code changes
- Provides high performance and efficiency for DL inference solutions to deploy across Intel XPU architectures (cloud to edge)
- Users: Data Scientists, AI Researchers, DL/ML Developers, AI Application Developers, Media and Vision Developers
- Use Cases:
  - Data Ingestion, data pre-processing, ETL operations
  - Model training and inference
  - Scaling to multi-core / multi-nodes / clusters
  - CPU – Intel Xeon, Core and Atom processors
  - GPU – Intel® Processor Graphics (integrated), Intel® Iris® Xe Max Graphics, Future Intel Xe architecture Artic Sound/Ponte Vecchio
- Hardware Support:
  - CPUs – Data center, server, workstation segments – Intel® Xeon® and Core™ processors
  - Future Intel Xe GPUs – Artic Sound/Ponte Vecchio
  - FPGA – Intel® Arria® 10 FPGA
  - GNA – Intel® Gaussian & Neural Accelerator
- Low Precision Support: Use Intel® Low Precision Optimization Tool when using the Intel oneAPI AI Analytics Toolkit
  - Supports BF16 for training and FP16, Int8 and BF16 for inference
  - Seamlessly integrates with Intel optimized frameworks
  - Available in the AI toolkit and independently

### Intel® Distribution of OpenVINO™ toolkit
- Provides high performance and efficiency for DL inference solutions to deploy across Intel XPU architectures (cloud to edge)
- Optimized package size for deployment based on memory requirements
- Users: AI Application Developers, Media and Vision Developers
- Use Cases:
  - Inference applications for vision, speech, text, NLP
  - Media streaming / encode, decode
  - Scale across hardware architectures – edge, cloud, datacenter, device
- Hardware Support:
  - CPU – Intel Xeon, Core and Atom processors
  - GPU – Intel® Processor Graphics (integrated), Intel® Iris® Xe Max Graphics, Future Intel Xe architecture Artic Sound/Ponte Vecchio
  - VPU - NCS & Intel® Vision Accelerator Design Products
  - FPGA - Intel® Arria® 10 FPGA
  - GNA - Intel® Gaussian & Neural Accelerator
- Low Precision Support: Use Post Training Optimization Tool when using OpenVINO
  - Supports FP16, Int8 and BF16 for inference
  - Directly works with Intermediate Representation Format
  - Available in the Intel Distribution of OpenVINO toolkit
  - Provides Training extension via NNCF for PyTorch with FP16, Int8

**Exception**: If a model is not supported by OpenVINO™ toolkit for inference deployment, build custom layers; or fall back to use the Intel oneAPI AI Analytics Toolkit and use optimized DL frameworks for inference.

### Which Toolkit to Use When?

<table>
<thead>
<tr>
<th>Intel® oneAPI AI Analytics Toolkit</th>
<th>Intel® Distribution of OpenVINO™ toolkit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key Value/ Benefits</strong></td>
<td><strong>Provides high performance and efficiency for DL inference solutions to deploy across Intel XPU architectures (cloud to edge)</strong></td>
</tr>
<tr>
<td>Provides performance and easy integration across end-to-end data science pipeline for efficient AI model development</td>
<td><strong>Optimized package size for deployment based on memory requirements</strong></td>
</tr>
<tr>
<td>Maximum compatibility with open source FWKs and Libs with drop-in acceleration that requires minimal to no code changes</td>
<td></td>
</tr>
<tr>
<td><strong>Users</strong></td>
<td><strong>AI Application Developers, Media and Vision Developers</strong></td>
</tr>
<tr>
<td>Data Scientists, AI Researchers, DL/ML Developers</td>
<td></td>
</tr>
<tr>
<td><strong>Use Cases</strong></td>
<td><strong>Inference applications for vision, speech, text, NLP</strong></td>
</tr>
<tr>
<td>Data Ingestion, data pre-processing, ETL operations</td>
<td><strong>Media streaming / encode, decode</strong></td>
</tr>
<tr>
<td>Model training and inference</td>
<td><strong>Scale across hardware architectures – edge, cloud, datacenter, device</strong></td>
</tr>
<tr>
<td>Scaling to multi-core / multi-nodes / clusters</td>
<td></td>
</tr>
<tr>
<td><strong>Hardware Support</strong></td>
<td><strong>CPU – Intel Xeon, Core and Atom processors</strong></td>
</tr>
<tr>
<td>CPUs – Data center, server, workstation segments – Intel® Xeon® and Core™ processors</td>
<td><strong>GPU – Intel® Processor Graphics (integrated), Intel® Iris® Xe Max Graphics, Future Intel Xe architecture Artic Sound/Ponte Vecchio</strong></td>
</tr>
<tr>
<td>Future Intel Xe GPUs – Artic Sound/Ponte Vecchio</td>
<td><strong>VPU - NCS &amp; Intel® Vision Accelerator Design Products</strong></td>
</tr>
<tr>
<td><strong>Low Precision Support</strong></td>
<td><strong>FPGA - Intel® Arria® 10 FPGA</strong></td>
</tr>
<tr>
<td>Use Intel® Low Precision Optimization Tool when using the Intel oneAPI AI Analytics Toolkit</td>
<td><strong>GNA - Intel® Gaussian &amp; Neural Accelerator</strong></td>
</tr>
<tr>
<td>Supports BF16 for training and FP16, Int8 and BF16 for inference</td>
<td><strong>Use Post Training Optimization Tool when using OpenVINO</strong></td>
</tr>
<tr>
<td>Seamlessly integrates with Intel optimized frameworks</td>
<td><strong>Supports FP16, Int8 and BF16 for inference</strong></td>
</tr>
<tr>
<td>Available in the AI toolkit and independently</td>
<td><strong>Directly works with Intermediate Representation Format</strong></td>
</tr>
</tbody>
</table>

**Exception**: If a model is not supported by OpenVINO™ toolkit for inference deployment, build custom layers; or fall back to use the Intel oneAPI AI Analytics Toolkit and use optimized DL frameworks for inference.
BackUp
Intel® oneAPI Base Toolkit

Accelerate Data-centric Workloads

A set of core tools and libraries for developing high-performance applications on Intel® CPUs, GPUs, and FPGAs

Who Uses It?
- A broad range of developers across industries
- Native Code Developers/Framework Developers

Top Features/Benefits
- Data Parallel C++ (DPC++) compiler, library and analysis tools; DPC++ Compatibility tool helps migrate existing code written in CUDA
- Optimized performance libraries for threading, math, data analytics, deep learning, and video/image/signal processing

Learn More: [intel.com/oneAPI-BaseKit](https://intel.com/oneAPI-BaseKit)
AI Development Workflow

**Data Analytics**
- **Data Ingestion & Pre-processing**
- **Classical ML Training & Prediction**

**Machine Learning**
- **Optimize Primitives for DL FWKs**
- **Train DL Model on Intel CPU/dGPU**
- **Re-train a model on custom data**
- **Pick a Pre-trained Intel-optimized Model**

**Deep Learning**
- **Use AI Kit (Modin, Omnisci, Pandas, Numpy, Scipy)**
- **Use AI Kit (Scikit-learn+Daal4py, XGBoost)**
- **Use Base Kit (oneDNN, oneCCL)**
- **Use AI Kit (Intel-optimized TensorFlow, Pytorch)**
- **Use AI Kit (Intel-optimized TensorFlow, Pytorch)**
- **Use AI Kit (Model Zoo for Intel® architecture)**

- **Train DL Model on Intel CPU/dGPU**
- **Run DL Inference on trained model**
- **Convert to Low Precision & run inference**
- **Use AI Kit (Low precision Opt Tool + Intel-optimized TensorFlow, Pytorch)**
- **Use AI Kit (Intel-optimized TensorFlow, Pytorch)**

- **Further optimize**

**Deploy DL Models on Intel® platforms**

**Trained Model**

**Pick a pre-trained model in IR format (Open Model Zoo)**

**Use OpenVINO™ Toolkit**

- **CPU**
- **GPU**
- **VPU**
- **FPGA**
- **GNA**

**Native Code Developers, Framework Developers**
- **Data Scientists, AI Researchers, ML/DL Developers**
- **AI, Media & Computer Vision Application Developers**

While there are a few distribution options to directly download Intel-optimized FWKs, machine learning libraries and tools individually. **Our recommendation is to get them via the Intel® oneAPI AI Analytics Toolkit** for seamless interoperability and good out-of-box experience.

**AI Kit = Intel® oneAPI AI Analytics Toolkit**
**Base Kit = Intel® oneAPI Base Toolkit**
AI Model Deployment Workflow

1. BUILD
   - Find a model
   - Build a model (ie train for accuracy)

1A. TRAIN
   - Bring a DL model

Is the model accurate enough?

No

Train/Re-train model

Yes

Use AI Kit

0. PLAN
   - Determine model type & framework needed for your usage

1. BUILD
   - Find a model
   - Build a model (ie train for accuracy)

1A. TRAIN
   - Bring a DL model

Is the model accurate enough?

No

Train/Re-train model

Yes

Use AI Kit

3. OPTIMIZE
   - Run Model Optimizer
   - Did the model convert?
     - No
     - Yes
       - Use Custom Layers

3A. MODIFY THE MODEL POST-TRAINING
   - Use POT to change attributes of IR
   - Use multi-device plugin
   - Use Intel® DevCloud to find the right hardware

4. TEST
   - Test the model with IE & Benchmark
     - Do you prefer GUI?
       - Yes
       - No
         - Use DL Workbench

4A. ADVANCE MODEL TUNING
   - Use POT to change attributes of IR
     - Did you quantize the model?
       - Yes/NA
       - No
         - Use multi-device plugin

5. PACKAGE
   - Use Deployment Manager
     - Did you use multi-device?
       - Yes
       - No
         - Can you use different hardware?

6. INTEGRATE
   - Integrate Model to Application

7. DEPLOY
   - Get Intel Hardware for Deployment
   - Deploy App & Model

A comprehensive workflow to optimize your DL model for the Intel Hardware that will be used for running inference.

- Model Optimizer (MO)
- Open Model Zoo (OMZ)
- Deep Learning (DL)
- Intermediate Representation (IR)
- Inference Engine (IE)
- Hardware (HW)
- Post Training Optimization Tool (POT)
Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.
Configurations

Deep Learning Training and Inference Performance using Intel® Optimization for PyTorch with 3rd Gen Intel® Xeon® Scalable Processors

DLRM batch size (FP32/BF16): 2K/instance, 1 instance
DLRM dataset (FP32/BF16): Criteo Terabyte Dataset
DLRM batch size (INT8): 16/instance, 28 instances, dummy data.
Tested by Intel as of 6/2/2020.

Intel® Xeon® Platinum 8380H Processor, 4 socket, 28 cores HT On Turbo ON Total Memory 768 GB (24 slots/ 32GB/ 3200 MHz), BIOS: WLYDCRB1.SYS.0015.P96.2005070242 (ucode: 0x700001b), Ubuntu 20.04 LTS, kernel 5.4.0-29-generic
PyTorch: https://github.com/intel/intel-extension-for-pytorch.git
Intel® Xeon® Platinum 8380H Processor, 4 socket, 28 cores HT On Turbo ON Total Memory 768 GB (24 slots/ 32GB/ 3200 MHz), BIOS: WLYDCRB1.SYS.0015.P96.2005070242 (ucode: 0x700001b), Ubuntu 20.04 LTS, kernel 5.4.0-29-generic
PyTorch: https://github.com/intel/intel-extension-for-pytorch.git
Inferepec Throughput FP32 vs Int8 optimized by Intel® Optimization for Tensorflow and Intel® Low Precision Optimization Tool (part of the Intel® oneAPI AI Analytics Toolkit)
Tested by Intel as of: 10/26/2020: TensorFlow v2.2 (https://github.com/Intel-tensorflow/tensorflow/tree/v2.2.0); Compiler: GCC 7.2.1; DNNL (https://github.com/oneapi-src/oneDNN) v1.2.0 75d0b1a7f3586c212e37accebb8ac0d221ce7216; Dataset: ImageNet/Coco/Dummy, refer to each model README; Precision: FP32 and Int8
Platform: Intel® Xeon® Platinum 8280 CPU; #Nodes: 1; #Sockets: 2; Cores/socket: 28; Threads/socket: 56; HT: On; Turbo: On; BIOS version: SE5C620.86B.02.01.0010.010620200716; System DDR Mem Config: 12 slots / 16GB / 2933; OS: CentOS Linux 7.8; Kernel: 4.4.240-1.el7.elrepo.x86_64

Stock scikit-learn vs Intel-optimized scikit-learn
Testing by Intel as of 10/23/2020. Intel® oneAPI Data Analytics Library 2021.1 (oneDAL), scikit-learn 0.23.1, Intel® Distribution for Python 3.8; Intel® Xeon® Platinum 8280LCPU @ 2.70GHz, 2Sockets, 28 cores per socket, 10M samples, 10 features, 100 clusters, 100 iterations, float32

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.
XGBoost CPU vs GPU
Test configs: Tested by Intel as of 10/13/2020; 
CPU: c5.18xlarge AWS Instance [2 x Intel® Xeon Platinum 8124M @ 18 cores, OS: Ubuntu 20.04.2 LTS, 193 GB RAM. GPU: p3.2xlarge AWS Instance (GPU: NVIDIA Tesla V100 16GB, 8 vCPUs), OS: Ubuntu 18.04.2 LTS, 61 GB RAM. SW: XGBoost 1.1:build from sources. compiler – G++ 7.4, nvcc 9.1. Intel® Data Analytics Acceleration Library (Intel® DAAL): 2019.4 version; Python env: Python 3.6, Numpy 1.16.4, Pandas 0.25, Scikit-learn 0.21.2.

XGBoost fit CPU acceleration
Test configs: Tested by Intel as of 10/13/2020; c5.24xlarge AWS Instance, CLX 8275 @ 3.0GHz, 2 sockets, 24 cores per socket, HT:on, DRAM (12 slots / 32 GB / 2933 MHz); SW: XGBoost 0.81, 0.9, 1.0 and 1.1:build from sources. compiler – G++ 7.4, nvcc 9.1. Intel® DAAL: 2019.4 version; Python env: Python 3.6, Numpy 1.16.4, Pandas 0.25, Scikit-learn 0.21.2.

End-to-End Census Workload Performance
Tested by Intel as of 10/15/2020. 2x Intel® Xeon® Platinum 8280 @ 28 cores, OS: Ubuntu 19.10.5.3.0-64-generic Mitigated, 384GB RAM. SW: Modin 0.8.1, scikit-learn 0.22.2, Pandas 1.0.1, Python 3.8.5, Daal4Py 2020.2 Census Data, (21721922, 45). Dataset is from IPUMS USA, University of Minnesota, www.ipums.org. Version 10.0.

Tiger Lake + Intel® Distribution of OpenVINO™ toolkit vs Coffee Lake CPU

<table>
<thead>
<tr>
<th>System Board</th>
<th>Intel prototype, TGL U DDR4 SODIMM RVP</th>
<th>ASUSTeK COMPUTER INC. / PRIME Z370-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>11th Gen Intel® Core™-S-1145G7E @ 2.6 GHz.</td>
<td>8th Gen Intel® Core™ i5-8500T @ 3.0 GHz.</td>
</tr>
<tr>
<td>Sockets / Physical cores</td>
<td>1 / 4</td>
<td>1 / 6</td>
</tr>
<tr>
<td>HyperThreading / Turbo Setting</td>
<td>Enabled / On</td>
<td>Na / On</td>
</tr>
<tr>
<td>Memory</td>
<td>2 x 198 MB 3200 MT/s DDR4</td>
<td>2 x 16384 MB 2667 MT/s DDR4</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu* 18.04 LT</td>
<td>Ubuntu* 18.04 LT</td>
</tr>
<tr>
<td>Kernel</td>
<td>5.8.0-050800-generic</td>
<td>5.3.0-24-generic</td>
</tr>
<tr>
<td>Software</td>
<td>Intel® Distribution of OpenVINO™ toolkit 2021.1.075</td>
<td>Intel® Distribution of OpenVINO™ toolkit 2021.1.075</td>
</tr>
<tr>
<td>BIOS</td>
<td>Intel® TGLIFUI1.R00.3243.A04.2006302148</td>
<td>AMI, version 2401</td>
</tr>
<tr>
<td>BIOS release date</td>
<td>Release Date: 06/30/2020</td>
<td>7/12/2019</td>
</tr>
<tr>
<td>BIOS Setting</td>
<td>Load default settings</td>
<td>Load default settings, set XMP to 2667</td>
</tr>
<tr>
<td>Test Date</td>
<td>9/9/2020</td>
<td>9/9/2020</td>
</tr>
<tr>
<td>Precision and Batch Size</td>
<td>CPU: INT8, GPU: FP16-INT8, batch size: 1</td>
<td>CPU: INT8, GPU: FP16-INT8, batch size: 1</td>
</tr>
<tr>
<td>Number of Inference Requests</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Number of Execution Streams</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Power (TDP Link)</td>
<td>28 W</td>
<td>35W</td>
</tr>
</tbody>
</table>