TOWARDS NEW HORIZONS WITH MACHINE LEARNING SOLUTIONS ON INTEL® IA

Ivan Kuzmin

Intel® Data Analytics Acceleration Library manager
Fast Evolution of Technology

**Bigger Data**
- Image: 1000 KB / picture
- Audio: 5000 KB / song
- Video: 5,000,000 KB / movie

**Better Hardware**
- Transistor density doubles every 18 months
- Cost / GB in 1995: $1000.00
- Cost / GB in 2015: $0.03

**Smarter Algorithms**
- Advances in neural networks leading to better accuracy in training models
Artificial Intelligence (AI)
Machines that can sense, reason, act without explicit programming

Machine Learning (ML), a key tool for AI, is the development, and application of algorithms that improve their performance at some task based on experience (previous iterations)

Deep Learning (DL)
Algorithms where multiple layers of neurons learn successively complex representations
- CNN
- RNN
- RBM
- ...

Statistical/Other Machine Learning
Algorithms based on statistical or other techniques for estimating functions from examples
- Naïve Bayes
- Support Vector Machines
- GA
- Linear Regression

Training: Build a mathematical model based on a data set

Inference: Use trained model to make predictions about new data
Machine Learning: Example Use Cases

- **Image classification and detection**
- **Image recognition/tagging**
- **Natural language recognition**
- **Big data pattern detection**
- **Targeted ads**
- **Fraud / face detection**
- **Gaming, check processing**
- **Computer server monitoring**
- **Financial forecasting and prediction**
- **Network intrusion detection**
Machine Learning Software Challenges

- Machine learning open source frameworks and libraries are often not well-optimized for newer Intel-based systems
- Frameworks can be difficult to configure and use
- Need to target heterogeneous hardware from model training in the datacenter to deployment on endpoint systems

![Chart showing Intel Xeon processor families with Core(s), Threads, and SIMD Width comparisons]

- More cores → More Threads → Wider vectors

*Product specification for launched and shipped products available on intel.com.
1. Not launched or in planning.
Machine Learning: Your Path to Deeper Insight
Driving increasing innovation and competitive advantage across industries

- **Solutions** for reference across industries
- **Tools/Platforms** to accelerate deployment
- **Optimized Frameworks** to simplify development
- **Libraries/Languages** featuring optimized building blocks
- **Hardware Technology** portfolio that is broad and cross-compatible

---

Intel® Math Kernel Library (Intel® MKL & MKL-DNN)
Intel® Data Analytics Acceleration Library (Intel® DAAL)
Intel® Deep Learning SDK for Training & Deployment
Intel® Distribution for Python*

Datacenter → Endpoint
+ Network
+ Memory
+ Storage

*Other names and brands may be claimed as the property of others.
## Comprehensive Toolbox for Range of Needs

<table>
<thead>
<tr>
<th><strong>High Level Overview</strong></th>
<th><strong>Intel® Math Kernel Library (Intel® MKL and MKL-DNN)</strong></th>
<th><strong>Intel® Data Analytics Acceleration Library (Intel® DAAL)</strong></th>
<th><strong>Intel® Distribution for Python</strong></th>
<th><strong>Intel-Optimized Open Source Frameworks</strong></th>
<th><strong>Intel® Deep Learning SDK</strong></th>
<th><strong>Trusted Analytics Platform</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High performance math primitives granting low level of control</td>
<td>Broad data analytics acceleration object oriented library supporting distributed ML at the algorithm level</td>
<td>Most popular and fastest growing language for machine learning</td>
<td>Toolkits driven by academia and industry for training machine learning algorithms</td>
<td>Accelerate deep learning model design, training and deployment</td>
<td>Single platform from data science to application development</td>
</tr>
</tbody>
</table>

| **Example Usage** | Framework developers call matrix multiplication, convolution functions | Call distributed alternating least squares algorithm for a recommendation system | Call scikit-learn k-means function for credit card fraud detection | Script and train a convolution neural network for image recognition | Train and deploy an optimized model on a constrained digital surveillance device | Application creation from the Big Data infrastructure, data science tools up to app development |

---

**Optimization Notice**

Copyright © 2016, Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.*
Intel® Math Kernel Library (Intel® MKL) Introduction

Highly optimized threaded math routines
- Performance, Performance, Performance!

Industry's leading math library
- Widely used in science, engineering, data processing

Tuned for Intel® processors – current and next generation

More math library users depend on MKL than any other library

Be multiprocessor aware
- Cross-Platform Support
- Be vectorised, threaded, and distributed multiprocessor aware
Intel® Math Kernel Library and Intel® MKL-DNN for Deep Learning Framework Optimization

Deep Learning Frameworks
- Caffe
- Theano
- Nn
- Google Tensorflow
- Torch

Comparison Table:

<table>
<thead>
<tr>
<th>Intel® MKL</th>
<th>Intel® MKL-DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN primitives + wide variety of other math functions</td>
<td>DNN primitives</td>
</tr>
<tr>
<td>C DNN APIs</td>
<td>C/C++ DNN APIs</td>
</tr>
<tr>
<td>Binary distribution</td>
<td>Open source DNN code*</td>
</tr>
<tr>
<td>Free community license. Premium support available as part of Parallel Studio XE</td>
<td>Apache 2.0 license</td>
</tr>
<tr>
<td>Broad usage DNN primitives; not specific to individual frameworks</td>
<td>Multiple variants of DNN primitives as required for framework integrations</td>
</tr>
<tr>
<td>Quarterly update releases</td>
<td>Rapid development ahead of Intel MKL releases</td>
</tr>
</tbody>
</table>

* GEMM matrix multiply building blocks are binary

Optimization Notice
Copyright © 2016, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
Improved Deep Neural Network training performance using Intel® Math Kernel Library (Intel® MKL)

Caffe/AlexNet single node training performance

Optimization Notice: Intel’s compiler 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.
Improved Deep Neural Network inference performance using Intel® Math Kernel Library (Intel® MKL)

Caffe/AlexNet single node inference 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 information go to https://www.intel.com/performance. *Other names and brands may be 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.
Case Study I: Deep Learning

LeCloud* Illegal Video Detection

- LeCloud: leading video cloud provider in China who provides illegal video detection service
- Originally: Adopted open source BVLC Caffe w/OpenBlas as CNN framework
- Now: Using Intel Optimized Caffe plus Intel® Math Kernel Library, achieved 30x performance improvement for training in production

* The test data is based on Intel® Xeon® E5 2680 V3 processor
Intel Optimized Machine Learning Frameworks
Simplify Development on Intel® Architecture

Supports**
• Topologies: AlexNet, GoogLeNet, VGG-19...
• Intel® Xeon Processors and Intel® Xeon Phi™ Processors
• Single-node and Multi-node
• Intel branch of Caffe and Theano available on GitHub

Links and Resources available from Intel® Developer Zone
software.intel.com/machine-learning/deep-learning

** Support varies by framework. Additional topologies in process.

*Other names and brands may be claimed as property of others.
Intel® Data Analytics Acceleration Library (Intel® DAAL)

An Intel-optimized library that provides building blocks for all data analytics stages, from data preparation to data mining & machine learning

- Python, Java & C++ APIs
- Can be used with many platforms (Hadoop*, Spark*, R*, ...) but not tied to any of them
- Flexible interface to connect to different data sources (CSV, SQL, HDFS, ...)
- Windows*, Linux*, and OS X*
- Developed by same team as the industry-leading Intel® Math Kernel Library
- Open source, Free community-supported and commercial premium-supported options
- Also included in Parallel Studio XE suites
Intel DAAL Overview

Industry leading performance, C++/Java/Python library for machine learning and deep learning optimized for Intel® Architectures.

- Pre-processing
  - (De-)Compression

- Transformation
  - PCA
  - Statistical moments
  - Variance matrix
  - QR, SVD, Cholesky
  - Apriori

- Analysis

- Modeling
  - Linear regression
  - Naïve Bayes
  - SVM
  - Classifier boosting
  - Kmeans
  - EM GMM
  - Collaborative filtering
  - Neural Networks

- Validation

- Decision Making
Problem Statement

Big data analytics
*Current common practice*
- Run on state-of-art hardware
- Built with a patchwork of math libs
- Under-exploiting hardware performance features

- Limited performance
- Many layers of dependencies
- Low ROI on HW investment

Big data frameworks: Hadoop, Spark, Cassandra, etc.
Big data frameworks: Hadoop, Spark, Cassandra, etc.

Problem Statement

- Limited performance
- Many layers of dependencies
- Low ROI on HW investment

Data sources
- SQL stores
- NoSQL stores
- In-memory stores

Connectors
- Data mining
- Recommendation engines
- Customer behavior modeling
- BI analytics
- Real time analytics

Spark* MLLib → Breeze
Netlib-Java → JVM
JNI → Netlib BLAS

Data mining
Recommendation engines
Customer behavior modeling
BI analytics
Real time analytics

Current common practice
- Limited performance
- Many layers of dependencies
- Low ROI on HW investment

*Other names and brands may be claimed as the property of others.
Big data analytics
*Desired practice*

- Run on state-of-art hardware
- Single library to cover all stages of data analytics
- Fully optimized for underlying hardware

<table>
<thead>
<tr>
<th>Data sources</th>
<th>SQL stores</th>
<th>NoSQL stores</th>
<th>In-memory stores</th>
<th>Connectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social media</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IoT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mfg</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Big data frameworks: Hadoop, Spark, Cassandra, etc.

- Optimized performance
- Simpler development & deployment
- High ROI on HW investment

Desired Solution
Intel® DAAL vs. Spark* Mlib

K-means Performance Comparison on Eight-node Cluster

<table>
<thead>
<tr>
<th>Speedup</th>
<th>300K rows, 4K columns, 120 clusters</th>
<th>3000K rows, 4K columns, 120 clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7x</td>
<td>4x</td>
</tr>
</tbody>
</table>

Configuration Info - Versions: Intel® Data Analytics Acceleration Library 2017, Spark 1.2; Hardware: Intel® Xeon® Processor E5-2699 v3, 2 Eighteen-core CPUs (40MB LLC, 2.3GHz), 128GB of RAM per node; Operating System: CentOS 6.6 x86_64.

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. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

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.
INTEL® DISTRIBUTION FOR PYTHON
OUR APPROACH

1. Enable hooks to Intel® MKL, Intel® DAAL, Intel® IPP functions in the most popular numerical packages
   - NumPy, SciPy, Scikit-Learn, PyTables, Scikit-Image, ...

2. Available through Intel® Distribution for Python* and as Conda packages
   - Most optimizations eventually upstreamed to home open source projects

3. Provide Python interfaces for Intel® DAAL (a.k.a PyDAAL)

---

**Optimization Notice**

Copyright © 2016, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
Numpy & Scipy optimizations with Intel® MKL

### Linear Algebra
- BLAS
- LAPACK
- ScaLAPACK
- Sparse BLAS
- Sparse Solvers
  - Iterative
  - PARDISO SMP & Cluster

### Fast Fourier Transforms
- Multidimensional
- FFTW interfaces
- Cluster FFT

### Vector Math
- Trigonometric
- Hyperbolic
- Exponential
- Log
- Power
- Root

### Vector RNGs
- Multiple BRNG
- Support methods for independent streams creation
- Support all key probability distributions

### Summary Statistics
- Kurtosis
- Variation coefficient
- Order statistics
- Min/max
- Variance-covariance

### And More
- Splines
- Interpolation
- Trust Region
- Fast Poisson Solver

---

**Configuration info:**
- Versions: Intel® Distribution for Python 2017 Beta, icc 15.0
- Hardware: Intel® Xeon® CPU E5-2698 v3 @ 2.30GHz (2 sockets, 16 cores each, HT=OFF), 64 GB of RAM, 8 DIMMS of 8GB@2133MHz; Operating System: Ubuntu 14.04 LTS.

---

**Optimization Notice**
Copyright © 2016, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
Near native performance on Intel® Xeon™ and Intel® Xeon Phi™

- Runs out-of-the-box with any Python
- Intel Distribution for Python delivers much greater efficiency than "system" Python
- Potential for future multi-threaded performance tuneings in numpy and scipy
INTEL® DEEP LEARNING SDK
Intel® Deep Learning SDK
Accelerate Your Deep Learning Solution

A free set of tools for data scientists and software developers to develop, train, and deploy deep learning solutions

“Plug & Train/Deploy”
Simplify installation & preparation of deep learning models using popular deep learning frameworks on Intel hardware

Maximum Performance
Optimized performance for training and inference on Intel® Architecture

Increased Productivity
Faster Time-to-market for training and inference, Improve model accuracy, Reduce total cost of ownership
Deep Learning Training Tool
Intel® Deep Learning SDK

- Simplify installation of Intel optimized Deep Learning Frameworks
- Easy and Visual way to Set-up, Tune and Run Deep Learning Algorithms:
  - Create training dataset
  - Design model with automatically optimized hyper-parameters
  - Launch and monitor training of multiple candidate models
  - Visualize training performance and accuracy
Deep Learning Deployment Tool
Intel® Deep Learning SDK

Unleash fast scoring performance on Intel products while abstracting the HW from developers

- Imports trained models from all popular DL framework regardless of training HW
- Compresses model for improved execution, storage & transmission (pruning, quantization)
- Generate Inference HW-Specific Code (C/C++, OpenVX, OpenCL, etc.)
- Enables seamless integration with full system / application software stack
Deep Learning Tools for End-to-End Workflow

Intel® Deep Learning SDK

**Intel DL Training Tool**
- INSTALL / SELECT IA-Optimized Frameworks
- PREPARE / CREATE Dataset with Ground-truth
- DESIGN / TRAIN Model(s) with IA-Opt. Hyper-Parameters
- MONITOR Training Progress across Candidate Models
- EVALUATE Results and ITERATE

**Intel DL Deployment Tool**
- IMPORT Trained Model (trained on Intel or 3rd Party HW)
- COMPRESS Model for Inference on Target Intel HW
- GENERATE Inference HW-Specific Code (OpenVX, C/C++)
- INTEGRATE with System SW / Application Stack & TUNE
- EVALUATE Results and ITERATE

**MKL-DNN Optimized Machine Learning Frameworks**
- theano
- Caffe
- Microsoft
- Google
- NVIDIA
- torch

Xeon (local or cloud)

**Optimized libraries & run-times (MKL-DNN, OpenVX, OpenCL)**

*Data acquisition (sensors) and acceleration HW (FPGA, etc)*

Target Inference Hardware Platform (physical or simulated)

Copyright © 2016, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.*
Intel® Deep Learning SDK Availability

Training Tool Technical Preview

• Easy install includes Caffe* optimized for Intel® Architecture and libraries

• Provides a user-friendly, intuitive interface (through Chrome browser) while running on a Linux* server

• Supports Image Classification using popular topologies: LeNet, AlexNet and GoogLeNet

Now Available!
software.intel.com/deep-learning-sdk
Conclusions

• Intel® is committed to get best performance and user experience for AI workloads on IA

• Intel® DAAL optimizes the whole analytical flow from data acquisition till ML/DL model training and inference

• Intel® MKL supports primitives for optimization of ML applications

• Intel® MKL-DNN enables faster optimization of DL frameworks and serves as the platform for innovation on Intel® CPU

• Intel® Deep Learning SDK free set of tools for data scientists and software developers to easily develop, train, and deploy deep learning solutions
Legal Disclaimer & Optimization Notice

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

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.

Copyright © 2016, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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