AI: UNLEASHING THE NEXT WAVE

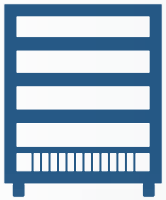
intel.com/ai
#intelAI

intel Artificial Intelligence

Intel strategy on al: improving cern Deep learning workflows and models using intel optimized solutions

Vilen Jumutcs
Solutions Architect, ML/DL Expert
Intel Corp (EMEA)

The next big wave of computing



mainframes

Standards-based servers

Cloud computing

- Data deluge
- COMPUTE
- breakthrough
- Innovation surge

Artificial intelligence

AI Compute Cycles will grow 12X by 2020

Intel ai approach

"One note does not make a symphony; one artist does not make an orchestra..."

– Matshona Dhliwayo, Philosopher

connectionist

models inspired by neural networks

Symbolist

rules and models using logical reasoning

evolutionaries

models inspired by Darwinian evolution

bayesians

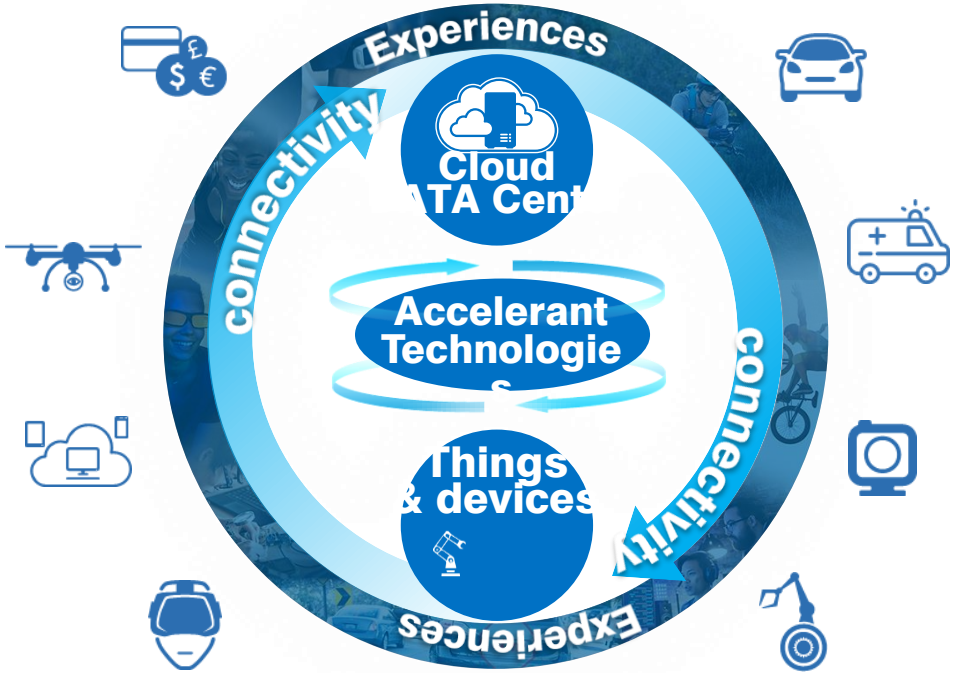
models using probabilistic inference

analogizers

reason from similar cases

*5 Tribes of Learners as described in "The Master Algorithm" by researcher Pedro Domingas

Artificial intelligence @ Intel



- MACHINE/DEEP LEARNING
- REASONING SYSTEMS
- Programmable solutions
- COMPUTER VISION
- TOOLS & STANDARDS
- Memory/Storage
- Networking communications

Unleash Your Potential with Intel's Complete AI Portfolio

intel AI portfolio

experiences

tools

Intel® Deep Learning SDK

Intel® Computer Vision SDK

Movidius Neural Compute Stick

saffron TECHNOLOGY an intel company

Frameworks

Apache Spark MLlib BigDL

TensorFlow

mxnet

theano

neon

Caffe

torch

E2E Tool

libraries

python

Intel® Dist DAAL

Intel® Nervana™ Graph*

Intel® MLSL

Intel® MKL

MKL-DNN

Movidius MvTensor Library

Associative Memory Base

hardware

Compute: intel ATOM inside | intel QUARK inside | intel CORE i7 inside | intel XEON inside | intel XEON PHI inside | intel ARRIA 10 inside | Lake Crest*

Memory & Storage

Networking

intel REALSENSE TECHNOLOGY

Movidius an intel company

Visual Intelligence

Unleash Full potential

*Coming 2017

Intel® Nervana™ Portfolio

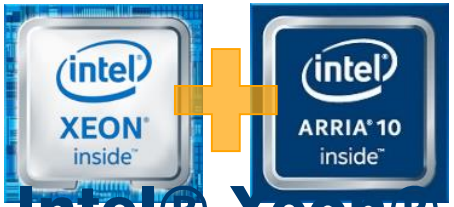
Common Architecture for Machine & Deep Learning

Intel® Xeon® Processors

Most Widely Deployed Machine Learning Platform (>97%*)

Intel® Xeon Phi™ Processors

Higher Performance Machine Learning, General Purpose



Intel® Xeon® Processor +FPGA

Breakthrough Deep Learning Inference & Workload Flexibility

Intel® Xeon® Processor + LakE CREST

Best-in-Class Neural Network Training Performance

Targeted acceleration

*Intel® Xeon® processors are used in 97% of servers that are running machine learning workloads today (Source: Intel)

ntel® Nervana™ porftolio (Detail

Trainin g

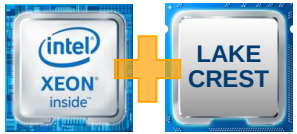
Batch

Train machine learning models across a diverse set of dense and sparse data

OR


Many batch models

Train large deep neural networks
Train large models as fast as possible

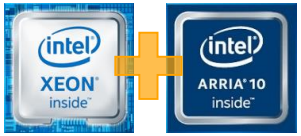
OR


Future

infere nce

Batch

Infer billions of data samples at a time and feed applications within ~1 day



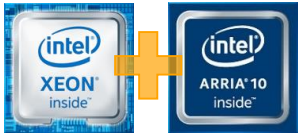
OR


Option for higher throughput/watt

Stream


…

Infer deep data streams with low latency in order to take action within milliseconds



Required for low latency

Edge

Power-constrained environments

Movidius

or other Intel® edge processor

oadmap: Intel® nervana™ platfor

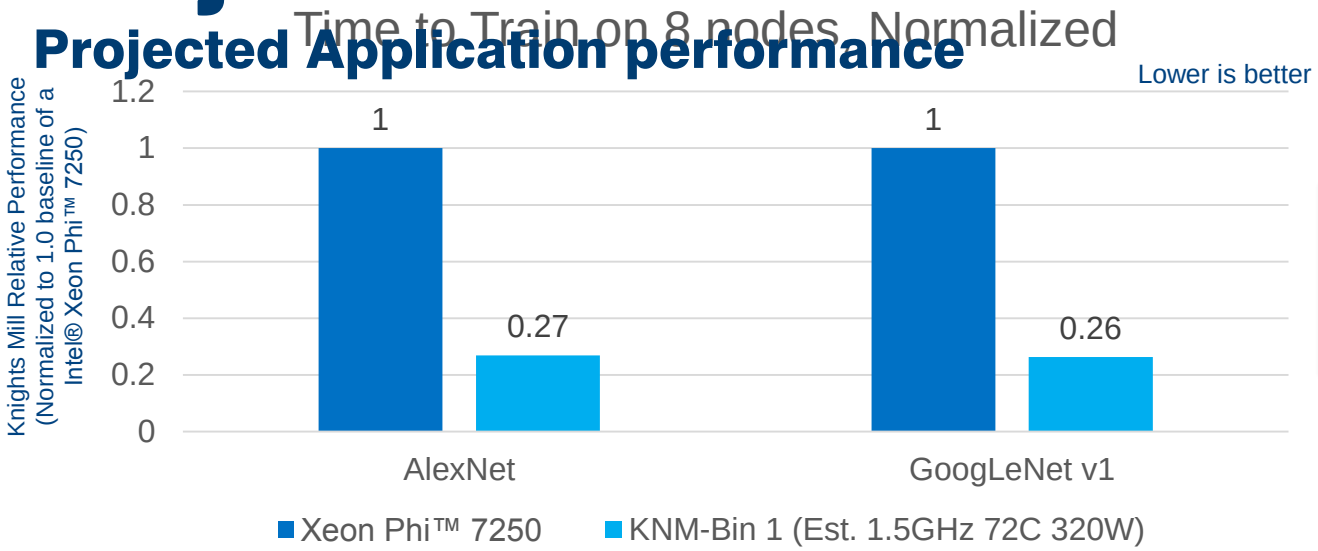
Shipping
Coming Soon

Targeted acceleration

	Today	2017	Future
Crest family (nervana)		Lake Crest Lake Crest	*TBA*
altera FPGA	Arria 10 FPGA	Canyon Vista	*TBA*
Intel® Xeon Phi™	Knights Landing	Knights Mill	*TBA*
Intel® Xeon® pro	Broadwell	Skylake, +FPGA	*TBA*

Knights Mill Performance Projections

Projected Application performance



KNM expected to deliver better deep learning time to train: Up to 74% faster than Xeon Phi™ 7250 estimated on 8 nodes

AI Software

Intel® Nervana™ portfolio

experiences

platforms

Intel® Nervana™ Cloud & Appliance

Intel® Nervana™ DL Studio

Intel® Computer Vision SDK

Movidius Fathom

intel Saffron™

Frameworks

Apache Spark MLlib BigDL

neon

TensorFlow

theano

mxnet

torch

Caffe

Caffe2

Chainer

libraries

python Intel Python Distribution

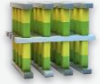
Intel® Data Analytics Acceleration Library (DAAL)

Intel® Nervana™ Graph*

Intel® Math Kernel Library (MKL, MKL-DNN)

hardware

Compute



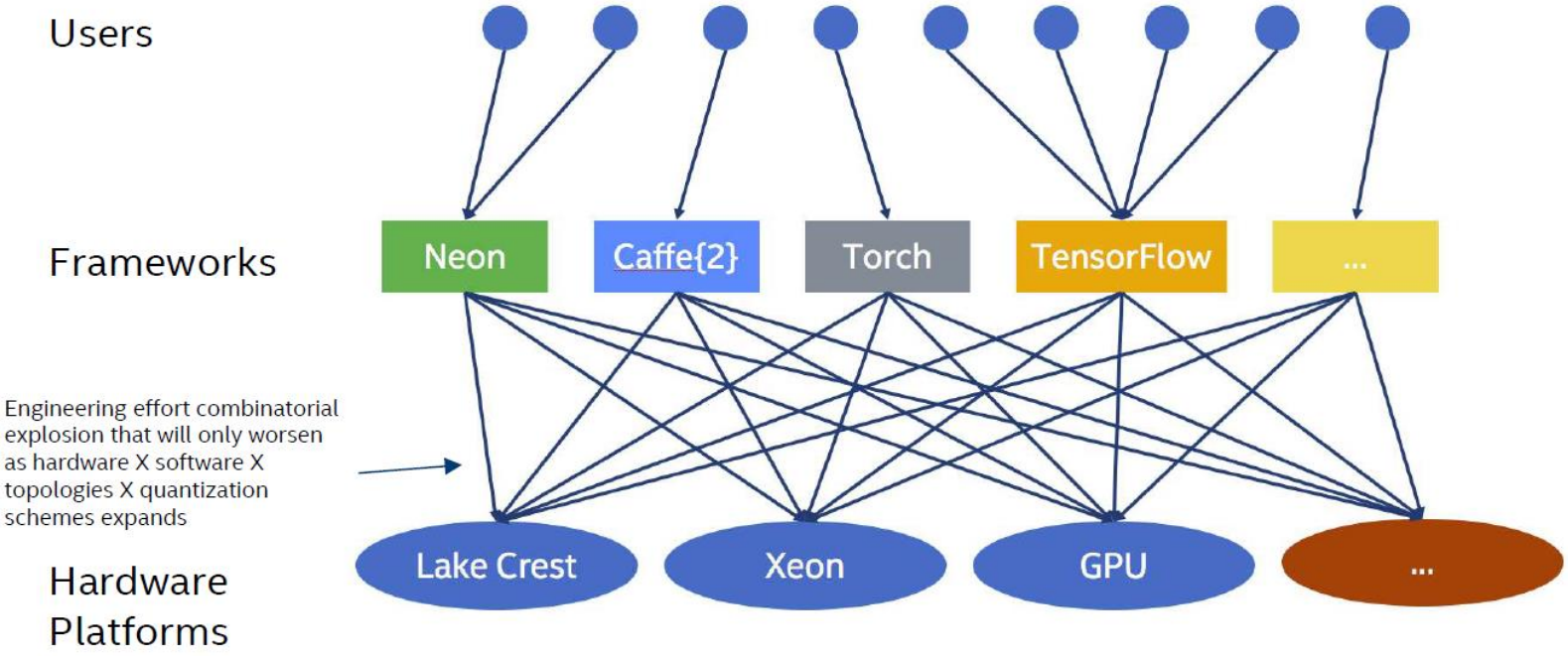
Memory & Storage

Networking

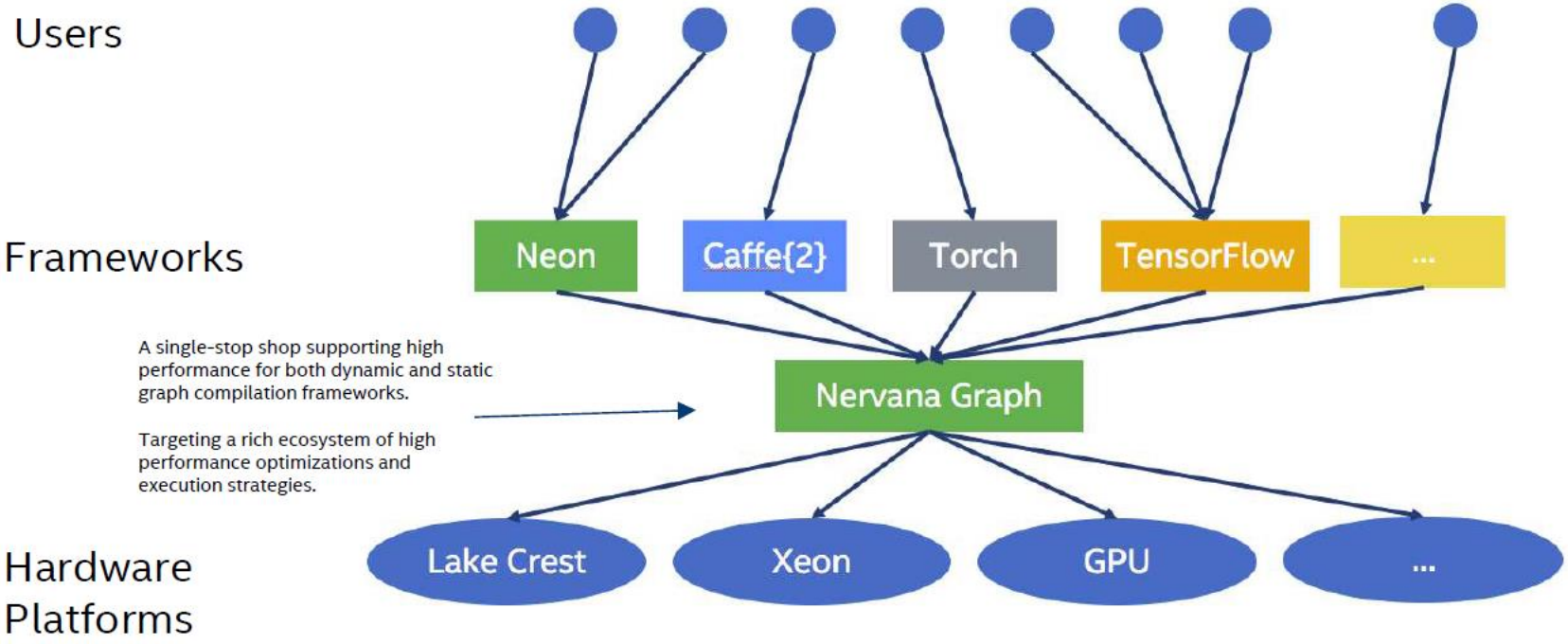
*Future

inside AI

Deep learning software – a many to many problem



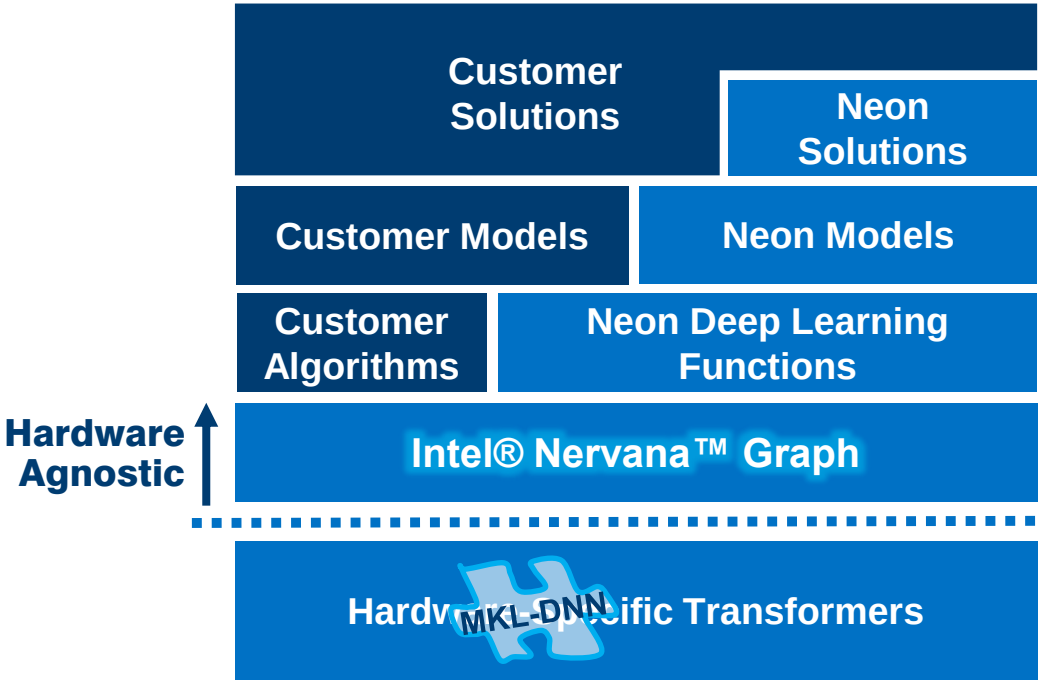
Nervana graph



intel® Nervana™ graph

COMING SOON

High-Performance Execution Graph for Neural Networks



Intel® Nervana™ Graph
enables optimizations that are applicable across multiple HW targets.

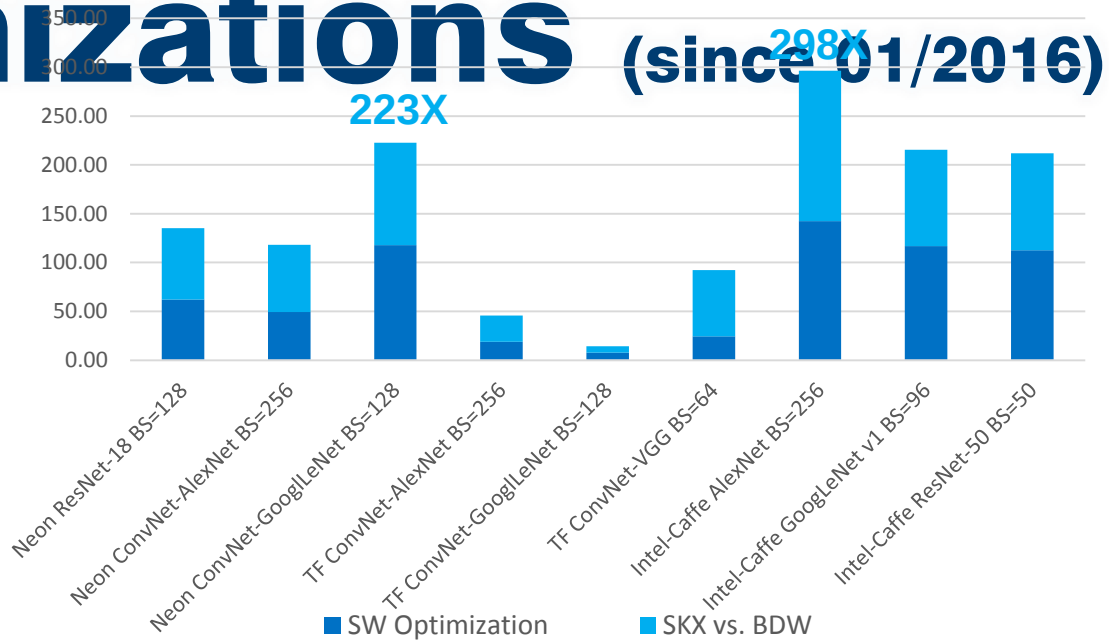
- Efficient buffer allocation
- Training vs inference optimizations
- Efficient scaling across multiple nodes
- Efficient partitioning of subgraphs
- Compounding of ops

The Intel® Nervana™ Graph will scale performance across hundreds of machine and deep learning frameworks

Benchmarks

Deep learning optimizations (since 01/2016)

Gain for SKX Platinum 8180 28 cores 2.5Ghz with optimized SW vs HSW E5-2699v3 18 cores 2.3Ghz un-optimized SW

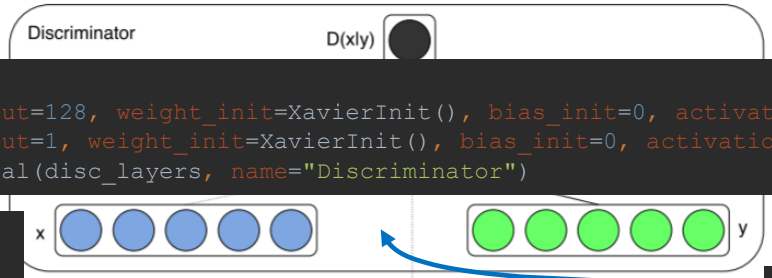


Configuration details on slide: 30
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any 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 Source: Intel measured as of November 2016
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

NEOn + ngraph solution example

Conditional GAN: c
chunks

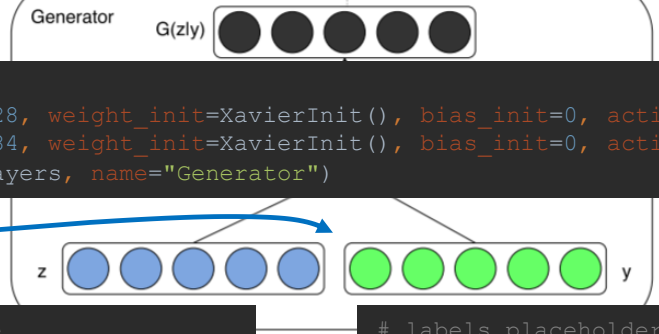
```
# batch size
args.batch_size = 128

# batch axis
N = ng.make_axis(name='N', length=args.batch_size)
```




```
# discriminator network
disc_layers = [Affine(nout=128, weight_init=XavierInit(), bias_init=0, activation=relu, batch_norm=True),
               Affine(nout=1, weight_init=XavierInit(), bias_init=0, activation=Logistic())]
discriminator = Sequential(disc_layers, name="Discriminator")
```

```
# image placeholder
L = ng.make_axis(name='L', length=784)
image_axes = ng.make_axes([L, N])
image = ng.placeholder(axes=image_axes)
```

ng.concat_along_axis




```
# generator network
gen_layers = [ Affine(nout=128, weight_init=XavierInit(), bias_init=0, activation=relu, batch_norm=True),
               Affine(nout=784, weight_init=XavierInit(), bias_init=0, activation=Tanh())]
generator = Sequential(gen_layers, name="Generator")
```

ng.concat_along_axis

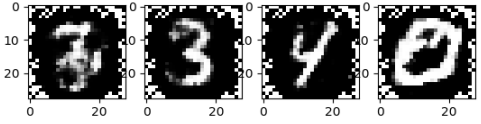
```
# noise placeholder for noise source
noisel = 100
noise_axis = ng.make_axis(name='M', length=noisel)
z_ax = ng.make_axes([noise_axis, N])
z = ng.placeholder(axes=z_ax)
```

```
# labels placeholder for conditioning
Y = ng.make_axis(name='Y', length=10)
# label_axes = ng.make_axes([Y,N])
labels = ng.one_hot(inputs['label'], axis=Y)
```

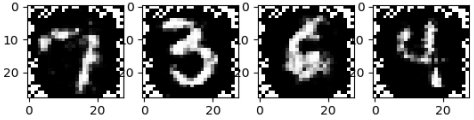
```
# input data place holders
inputs = train_set.make_placeholders()
```

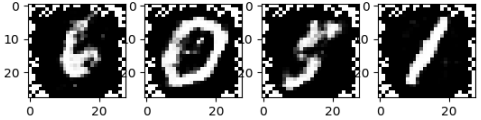
Mnist Conditional gan with ngraph – output examples

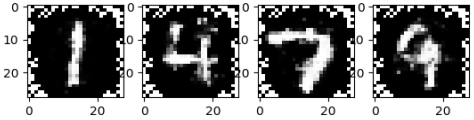
iteration 180 0 [3, 3, 4, 0, 6, 0, 5, 1]



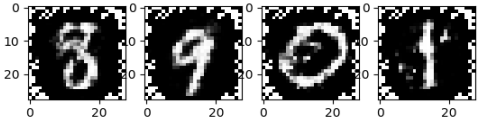
iteration 190 0 [7, 3, 6, 4, 1, 4, 7, 9]



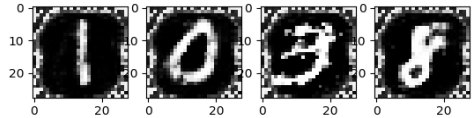


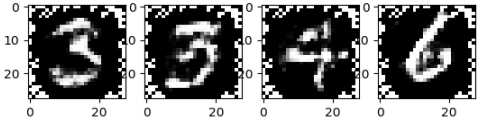


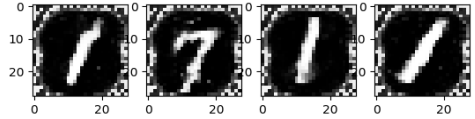
iteration 240 0 [8, 9, 0, 1, 3, 3, 4, 6]



iteration 260 0 [1, 0, 3, 8, 1, 7, 1, 1]







File Edit View Navigate Code Refactor Run Tools VCS Window Help

ngraph > private-ngraph > examples > gan > MNIST-conditional-gan.py

Project

- doc
- examples
  - benchmarks
  - cifar10
  - convnet-benchmarks
  - deepspeech
  - dqn
  - gan
  - imdb
  - mnist
  - ptb
  - ptr-net
  - walk_through
  - __init__.py
  - dist_hetr.py
- flex_examples
- integration_tests
- ngraph

optimizer.py | __builtin__.py | MNIST-conditional-gan.py | exceptions.py | mnist_dcgan.py | mnist_mlp.py | toy_mlp_gan.py

```
184     # noise generator
185     noise_generator = Noise(train_set.ndata, shape=(noisel,) + (args.batch_size,), seed=args.seed)
186
187     # dictionaries for input and outputs
188     generator_train_inputs = {'noise': z, 'labels' : inputs['label']}
189     discriminator_train_inputs = {'image': image, 'noise': z, 'labels' : inputs['label']}
190     generator_train_outputs = {'batch_cost': mean_cost_g, 'updates': updates_g,
191                                'generated': generated }  # for plots and debug
192     discriminator_train_outputs = {'batch_cost': mean_cost_d,
193                                    'updates': updates_d, 'generated': generated} # for plots and debug
194
195     with closing(ngt.make_transformer()) as transformer:
196         print('preparing the required computations')
197         train_computation_g = make_bound_computation(transformer,
198                                                      generator_train_outputs,
199                                                      generator_train_inputs)
200         train_computation_d = make_bound_computation(transformer,
201                                                      discriminator_train_outputs,
202                                                      discriminator_train_inputs)
203
```

Run: MNIST-conditional-gan | mnist_dcgan

```
Iteration 474 300 complete. Discriminator avg loss: 0.0288748890162 Generator avg loss: 5.24118232727
Iteration 474 400 complete. Discriminator avg loss: 0.493980497122 Generator avg loss: 5.53833007812
Iteration 475 0 complete. Discriminator avg loss: 0.019677888602 Generator avg loss: 5.34118938446
Iteration 475 100 complete. Discriminator avg loss: 0.0271359626204 Generator avg loss: 5.05509376526
Iteration 475 200 complete. Discriminator avg loss: 0.0117205902934 Generator avg loss: 5.6538028717
Iteration 475 300 complete. Discriminator avg loss: 0.0274948757142 Generator avg loss: 5.16712856293
Iteration 475 400 complete. Discriminator avg loss: 0.0318383127451 Generator avg loss: 5.14598083496
Iteration 476 0 complete. Discriminator avg loss: 0.061867415905 Generator avg loss: 5.59818458557
Iteration 476 100 complete. Discriminator avg loss: 0.0205344911665 Generator avg loss: 5.73826026917
Iteration 476 200 complete. Discriminator avg loss: 0.0174968317151 Generator avg loss: 5.18991279602
Iteration 476 300 complete. Discriminator avg loss: 0.0251116007566 Generator avg loss: 4.84896230698
Iteration 476 400 complete. Discriminator avg loss: 0.0139540676028 Generator avg loss: 5.63727378845
```

3: Find | 4: Run | 5: Debug | 6: TODO | 9: Version Control | Python Console | Terminal | File Transfer | Event Log

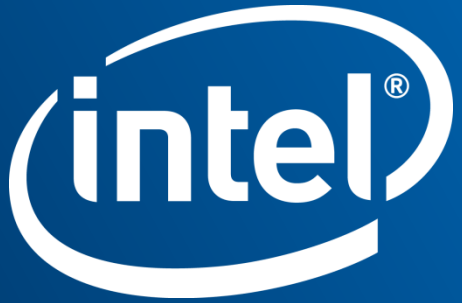
Push successful: private-ngraph: pushed 1 commit to origin/andreazanetti/MNIST-GAN-and-condGAN (yesterday 4:28 PM) 2382:63 LF UTF-8 Git: andreazanetti/MNIST-GAN-and-condGAN

summary

- Artificial intelligence (AI), the next big wave in computing, is an increasingly important source for competitive advantage that is already transforming industries
- Today is the ideal time to begin integrating AI into your products, services and business processes
- Intel has the complete AI portfolio, world-class silicon, and experience from successfully driving previous major computing transformations

Get started with #IntelAI today!

Use Intel's performance-optimized libraries & frameworks	Contact your Intel representative for help and POC opportunities	Find out more at www.intel.com/ai & software.intel.com/ai


intel®

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, 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.

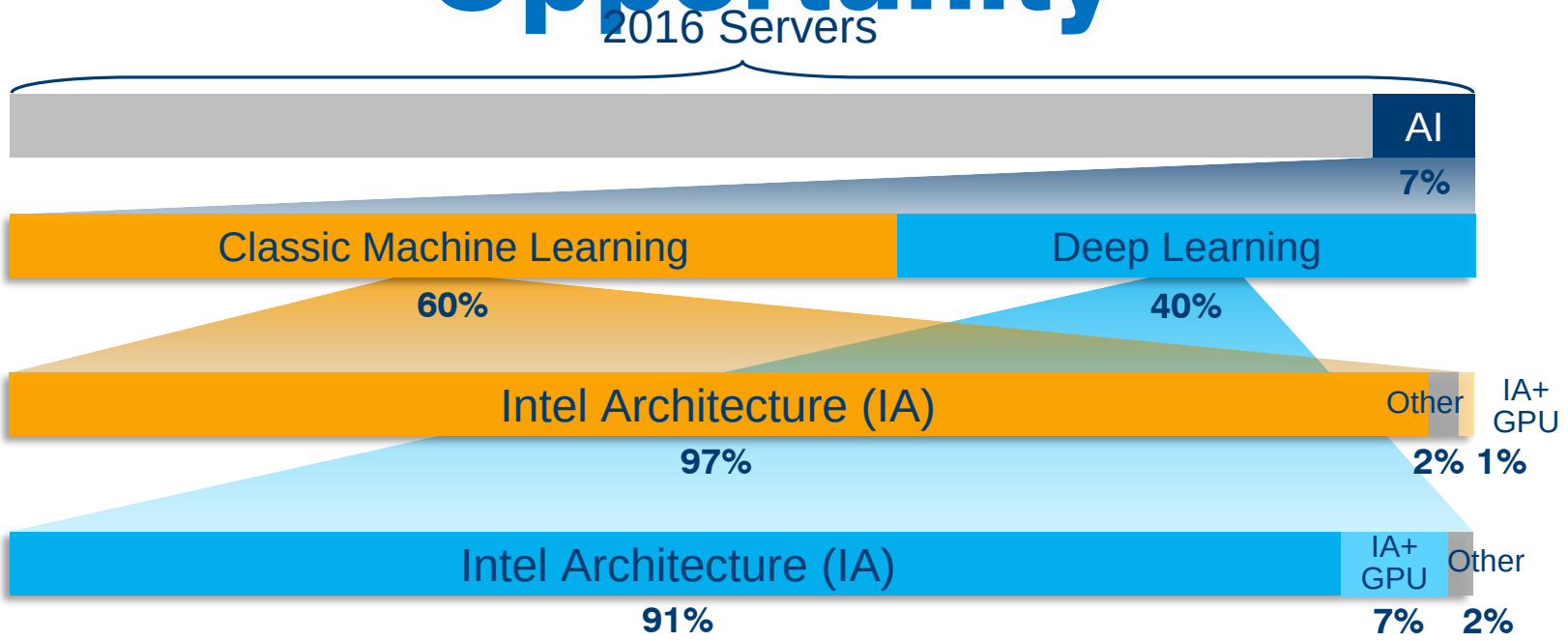
Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Pentium, Celeron, Atom, Core, 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.

additional info

Artificial Intelligence Opportunity

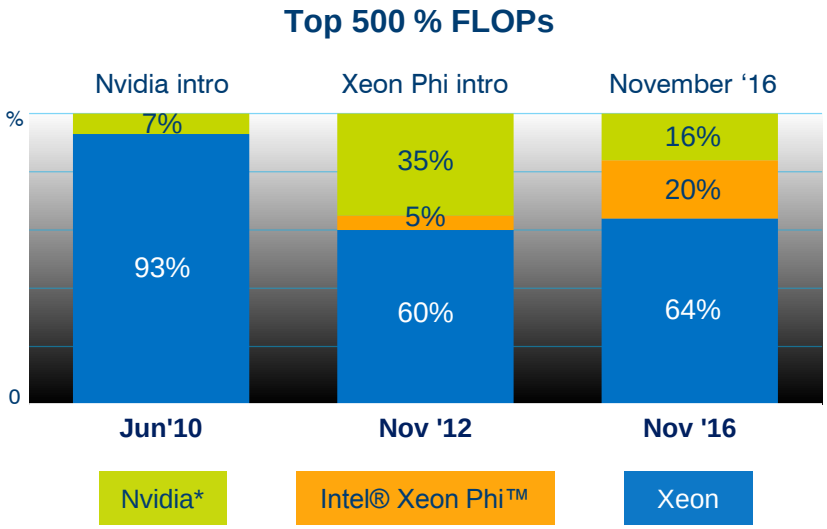
AI is the fastest growing data center workload



Source: Amalgamation of Intel data, analyst data and Intel analysis.

Artificial Intelligence Plan

Bringing the HPC Strategy to AI



Intel® Nervana™ Portfolio

Most widely deployed machine learning solution

COMING 2017
SKYLAKE

High performance, classic machine learning

COMING 2017
KNIGHTS MILL

Programmable, low-latency inference

SDVs SHIPPING TODAY
BROADWELL + ARRIA 10

Best in class neural network performance

COMING 2017
LAKE CREST

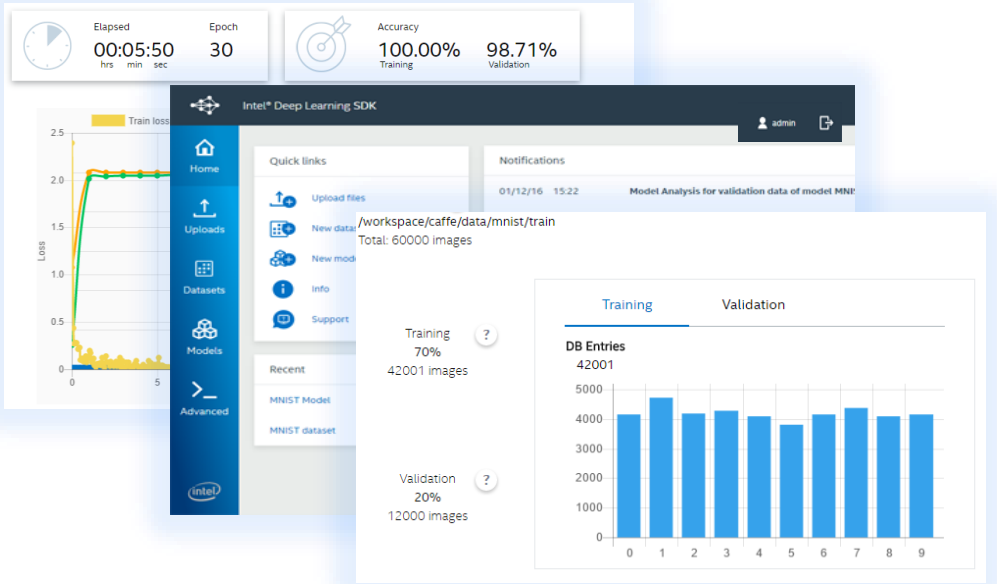
Intel® Deep Learning SDK

BETA Available Now

Accelerate Deep Learning Development

For developers looking to accelerate deep learning model design, training & deployment

- **FREE** for data scientists and software developers to develop, train & deploy deep learning
- **Simplify installation** of Intel optimized frameworks and libraries
- **Increase productivity** through simple and highly-visual interface
- **Enhance deployment** through model compression and normalization
- **Facilitate integration** with full software stack via inference engine



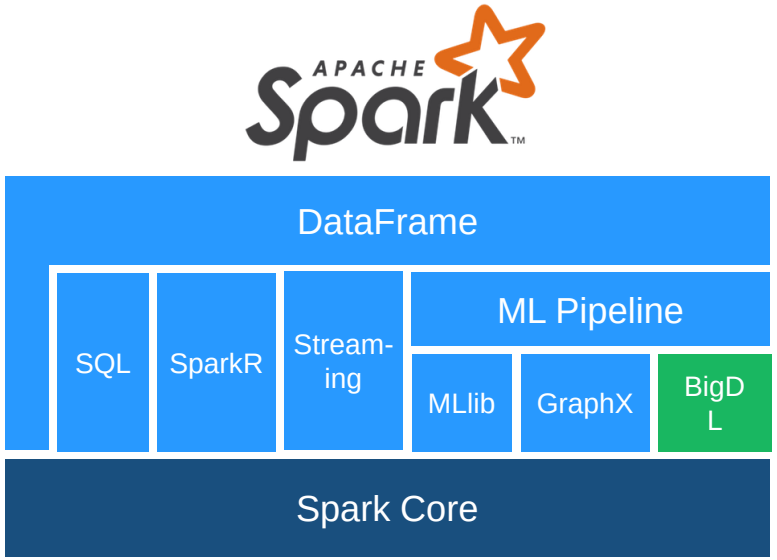
software.intel.com/deep-learning-sdk

BIGDL

Bringing Deep Learning to Big Data

For developers looking to run deep learning on Hadoop/Spark due to familiarity or analytics use

- **Open Sourced** Deep Learning Library for Apache Spark*
- **Make Deep learning more Accessible** to Big data users and data scientists.
- **Feature Parity** with popular DL frameworks like Caffe, Torch, Tensorflow etc.
- **Easy Customer and Developer Experience**
  - Run Deep learning Applications as Standard Spark programs;
  - Run on top of existing Spark/Hadoop clusters (No Cluster change)
- **High Performance** powered by Intel MKL and Multi-threaded programming.
- **Efficient Scale out** leveraging Spark architecture.



github.com/intel-analytics/BigDL

Intel distribution for python

Advancing Python Performance Closer to Native Speeds

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

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

- Prebuilt, optimized for numerical computing, data analytics, HPC
- Drop in replacement for your existing Python (no code changes required)

Drive Performance with Multiple Optimization Techniques

- Accelerated NumPy/SciPy/Scikit-Learn with Intel® MKL
- Data analytics with pyDAAL, enhanced thread scheduling with TBB, 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

software.intel.com/intel-distribution-for-python

Intel® MKL-DNN

Math Kernel Library for Deep Neural Networks

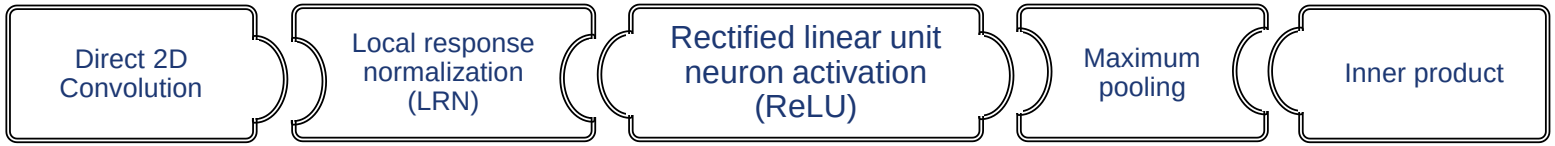
For developers of deep learning frameworks featuring optimized performance on Intel hardware

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® MKL library.

BETA Now Available!

github.com/01org/mkl-dnn



Machine learning scaling library

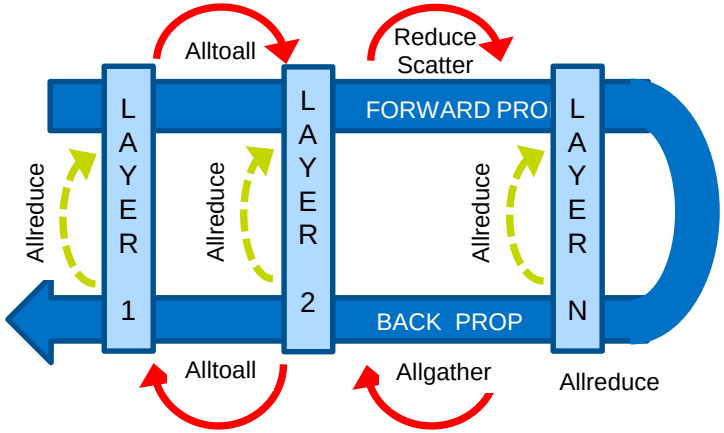
Scaling Deep Learning to 32 Nodes and Beyond

For maximum deep learning scale-out performance on Intel® architecture

BETA Now Available!

Deep learning abstraction of message-passing implementation

- Built on top of MPI; allows other communication libraries to be used as well
- Optimized to drive scalability of communication patterns
- Works across various interconnects: Intel® Omni-Path Architecture, InfiniBand, and Ethernet
- Common API to support Deep Learning frameworks (Caffe, Theano, Torch etc.)



github.com/01org/MLSL/releases