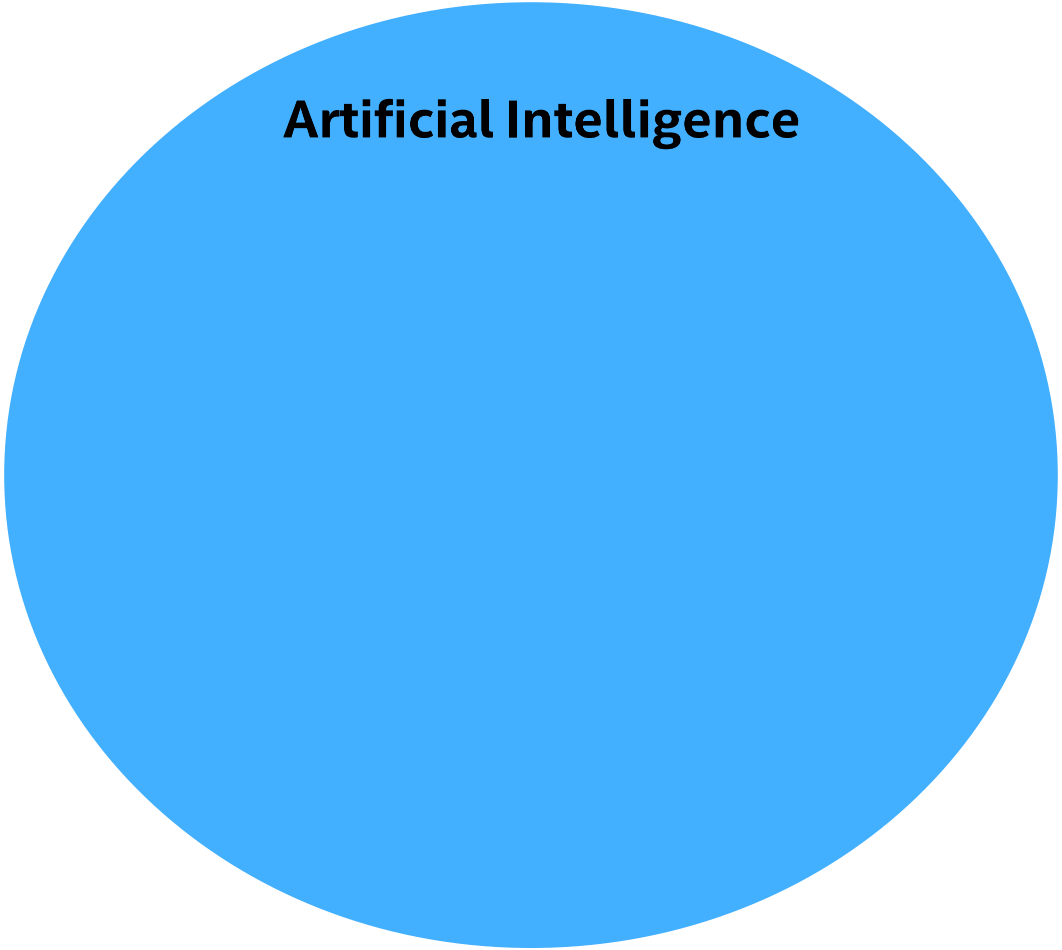
IDF16
INTEL DEVELOPER FORUM

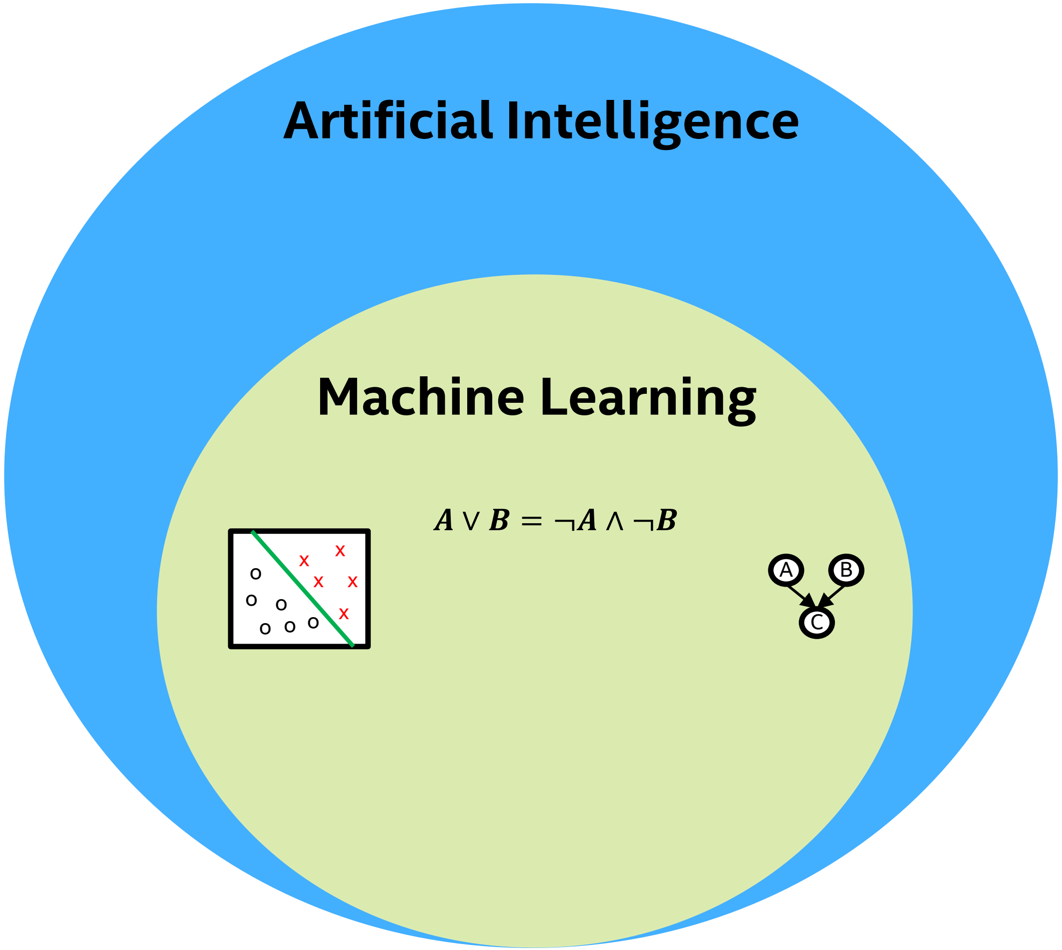
Scaling to Meet the Growing Needs of Artificial Intelligence (AI)

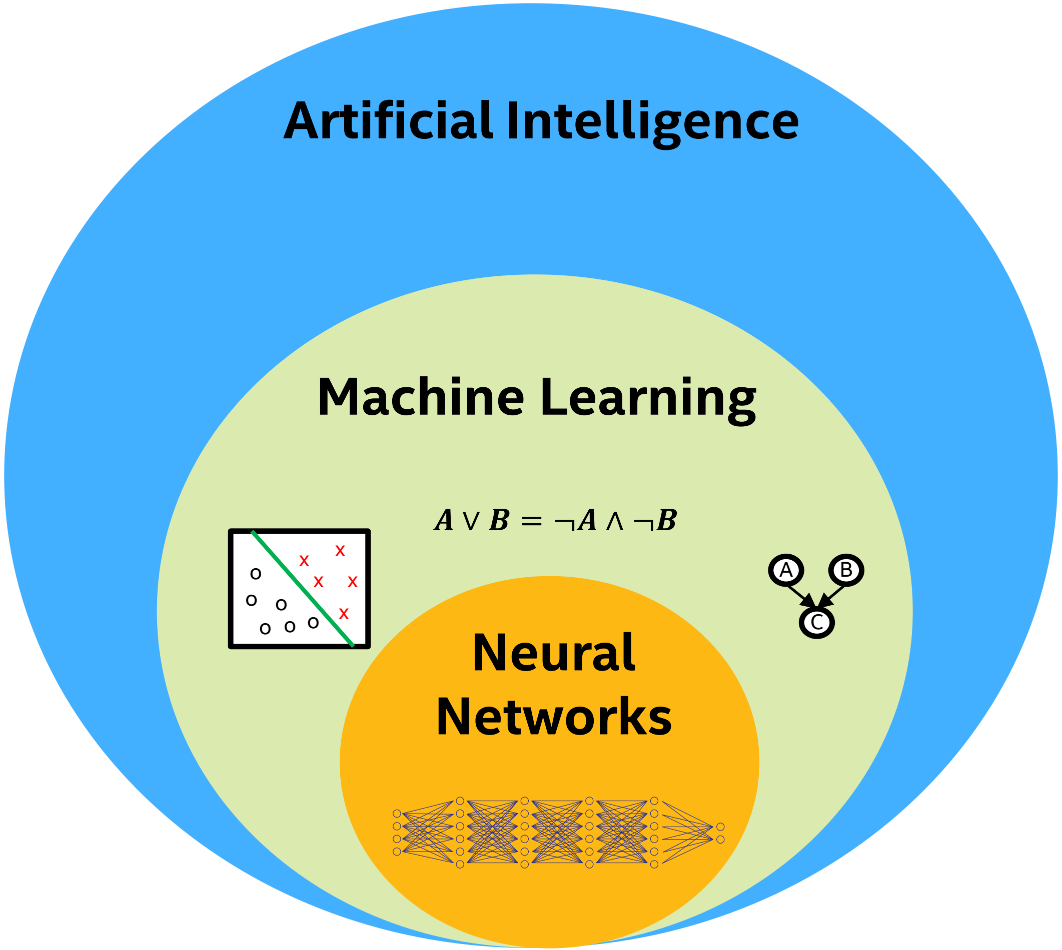
Pradeep K. Dubey

Intel Fellow, Intel Labs, Intel Corporation

experience what's inside™


Artificial Intelligence


Artificial Intelligence
Machine Learning
$A \vee B = \neg A \wedge \neg B$
A
B
C



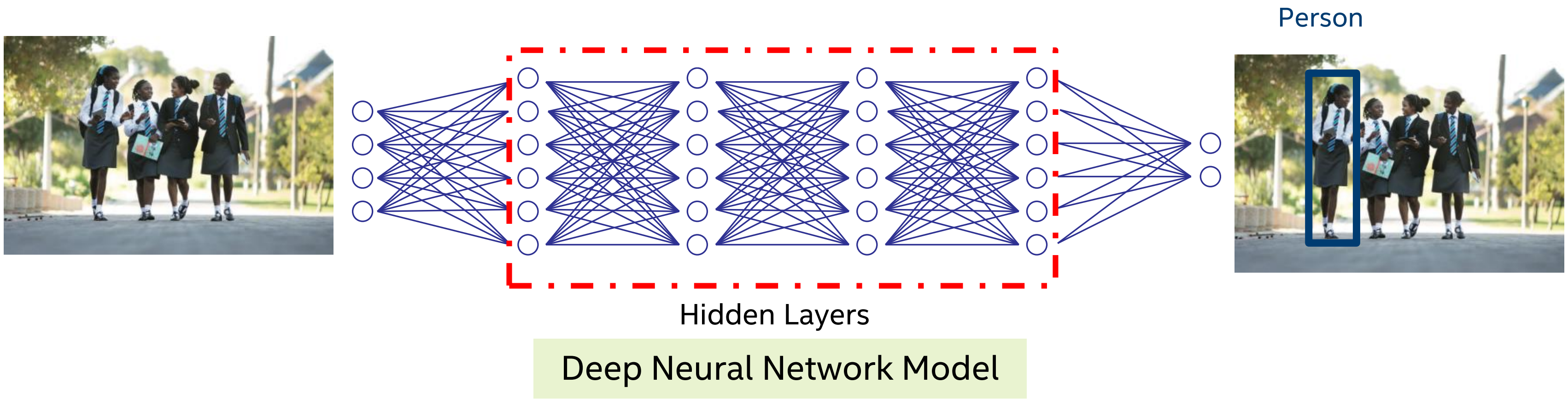
IDF16
INTEL DEVELOPER FORUM

Agenda

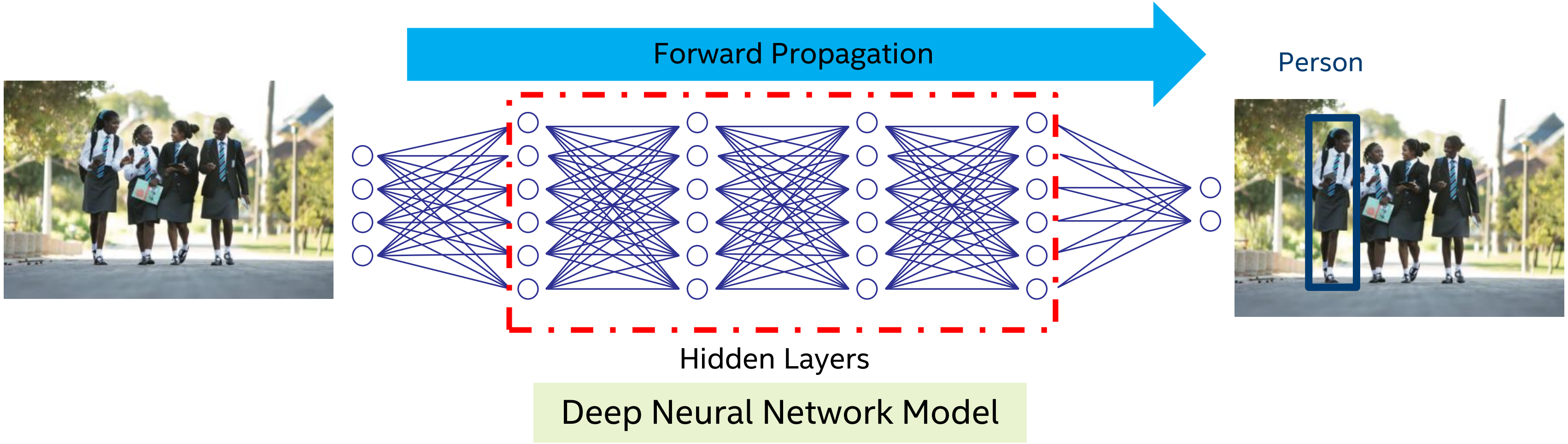
- Why do we need to scale machine learning?
- What makes it hard to scale and how we are addressing it
- Real-world applications
- Hardware roadmap, software tools and frameworks update

IDF16
INTEL DEVELOPER FORUM

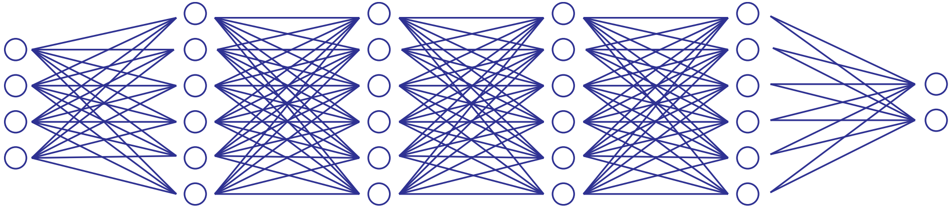
Deep Learning: Scoring or Inferencing



Deep Learning: Scoring or Inferencing

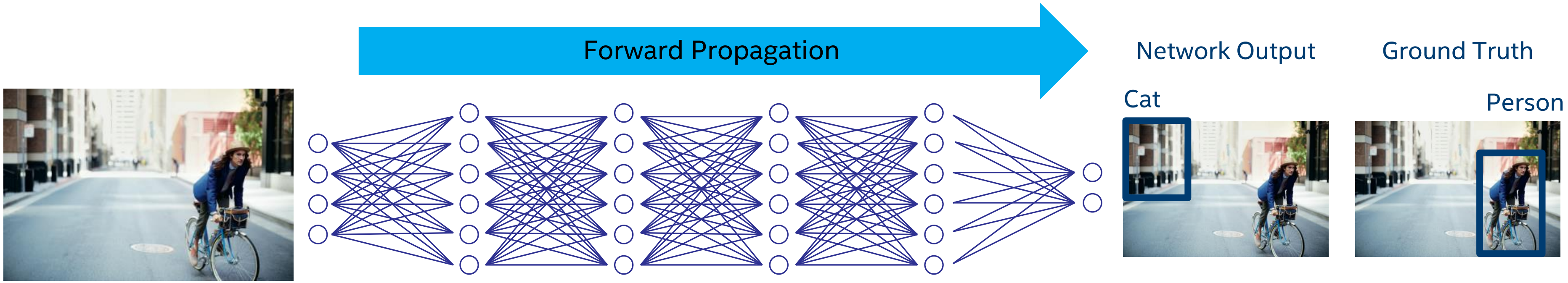


Deep Learning: Training



* Shihao Ji, S. V. N. Viswanathan, Nadathur Satish, Michael Anderson, and Pradeep Dubey. Blackout: Speeding up Recurrent Neural Network Language Models with very large vocabularies. http://arxiv.org/pdf/1511.06909v5.pdf. ICLR 2016

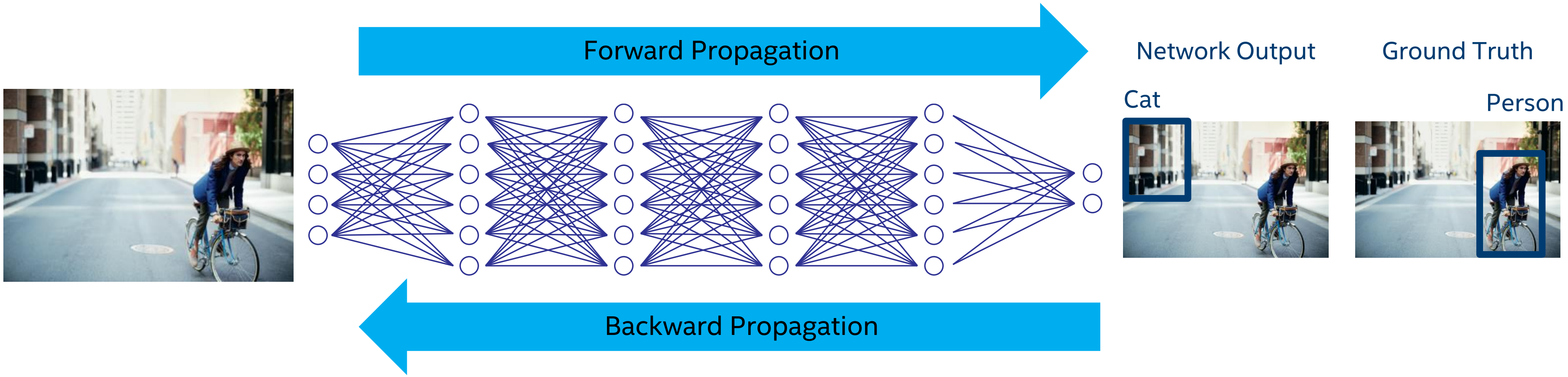
Deep Learning: Training



* Shihao Ji, S. V. N. Viswanathan, Nadathur Satish, Michael Anderson, and Pradeep Dubey. Blackout: Speeding up Recurrent Neural Network Language Models with very large vocabularies. http://arxiv.org/pdf/1511.06909v5.pdf. ICLR 2016

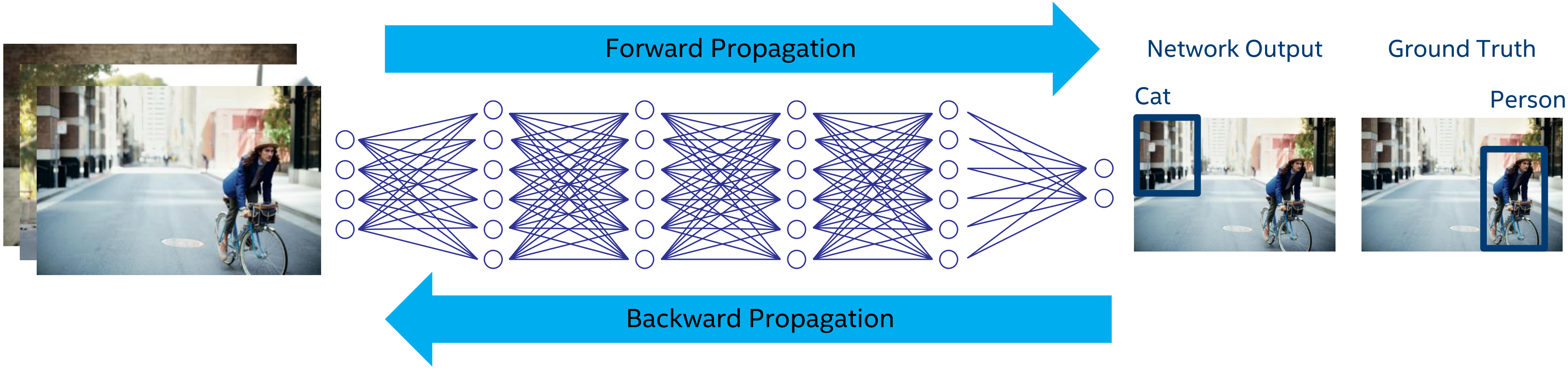
IDF16 INTEL DEVELOPER FORUM

Deep Learning: Training



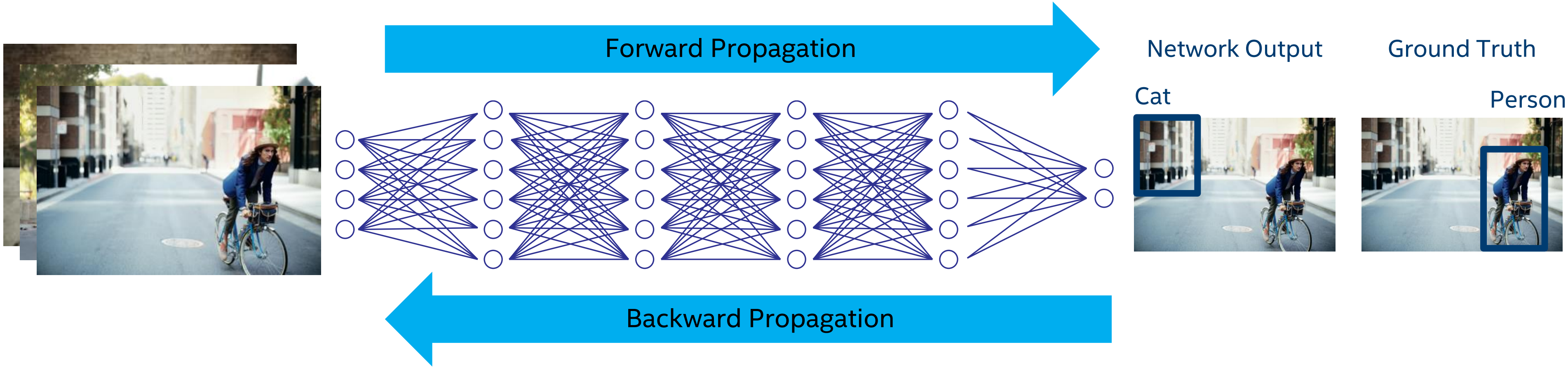
* Shihao Ji, S. V. N. Viswanathan, Nadathur Satish, Michael Anderson, and Pradeep Dubey. Blackout: Speeding up Recurrent Neural Network Language Models with very large vocabularies. http://arxiv.org/pdf/1511.06909v5.pdf. ICLR 2016

Deep Learning: Training



* Shihao Ji, S. V. N. Viswanathan, Nadathur Satish, Michael Anderson, and Pradeep Dubey. Blackout: Speeding up Recurrent Neural Network Language Models with very large vocabularies. http://arxiv.org/pdf/1511.06909v5.pdf. ICLR 2016

Deep Learning: Training

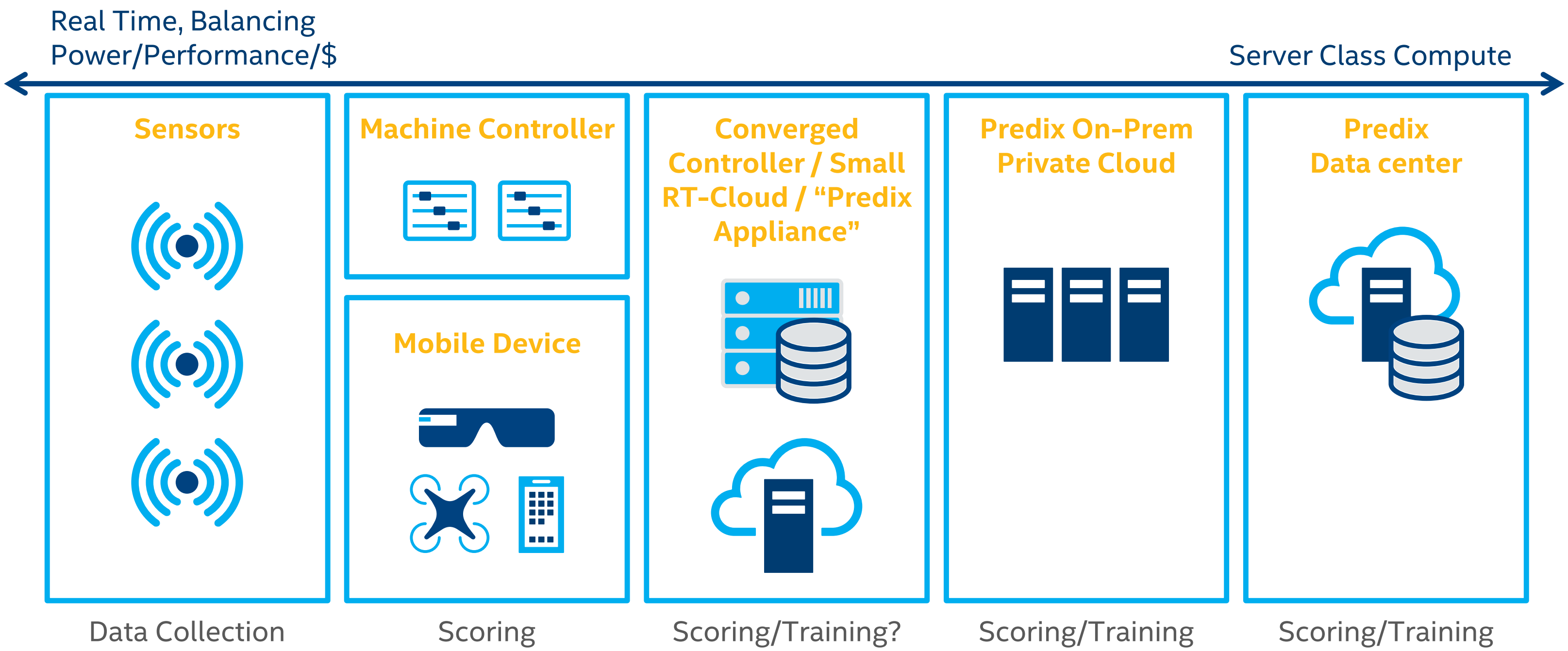


Complex Networks with billions of parameters can take days to train on a modern processor*

Hence, the need to reduce time-to-train using a cluster of processing nodes

* Shihao Ji, S. V. N. Viswanathan, Nadathur Satish, Michael Anderson, and Pradeep Dubey. Blackout: Speeding up Recurrent Neural Network Language Models with very large vocabularies. http://arxiv.org/pdf/1511.06909v5.pdf. ICLR 2016

Machine Learning Continuum: Connected Factory

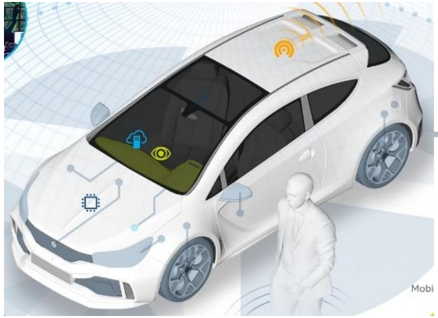


IDF16 INTEL DEVELOPER FORUM

Machine Learning Continuum: Self-Driving Cars

In-Vehicle

- Autonomous Driving Functions
- Sensor Processing
- Sensor Capture



Data Center

- Vehicle Endpoint Management
- Vehicle Simulation & Validation
- Captured Sensor Data Analytics

Data

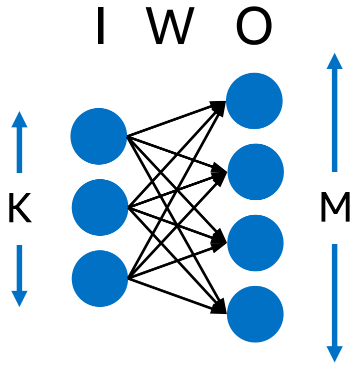
IDF16
INTEL DEVELOPER FORUM

Agenda

- Why do we need to scale machine learning
- What makes it hard to scale and how we are addressing it
- Real-world experience of an industry leader
- Hardware roadmap, software tools and frameworks update
- Summary

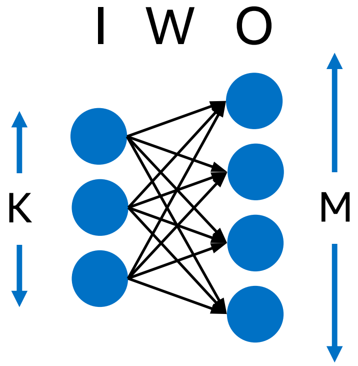
Compute Kernels

Compute Kernels

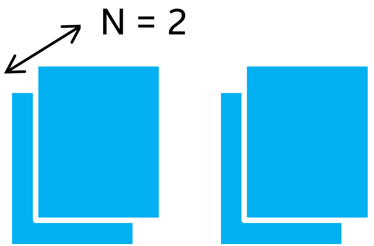


Fully connected layer

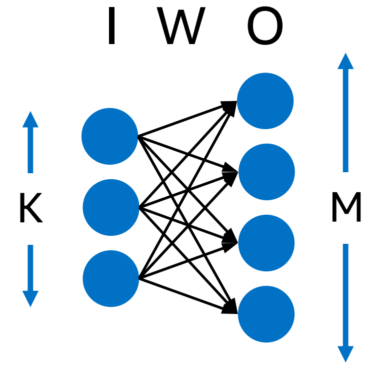
Compute Kernels



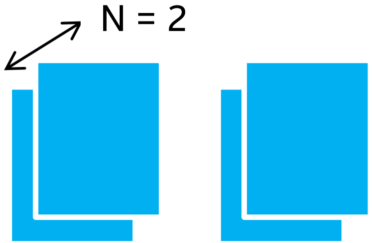
Fully connected layer



Compute Kernels



Fully connected layer



$\boldsymbol{I} \in \boldsymbol{R}^{KxN}$

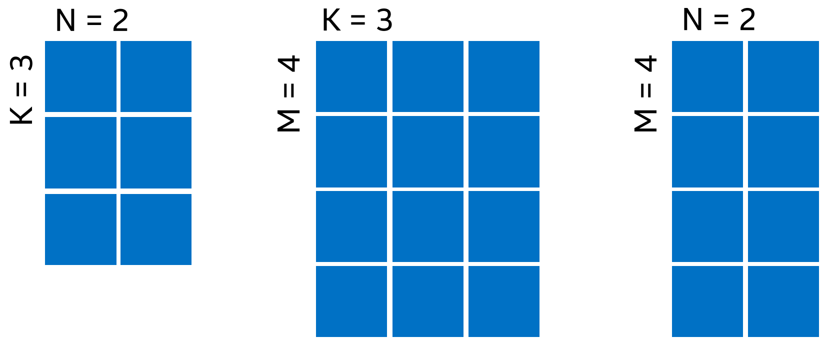
Input

$\boldsymbol{W} \in \boldsymbol{R}^{MxK}$

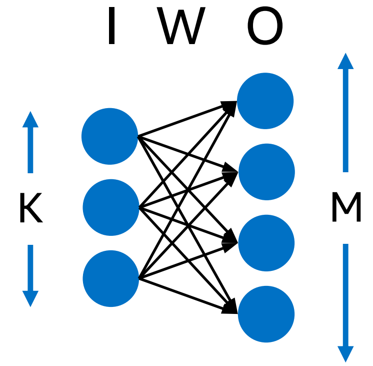
Weights or model

$\boldsymbol{O} \in \boldsymbol{R}^{MxN}$

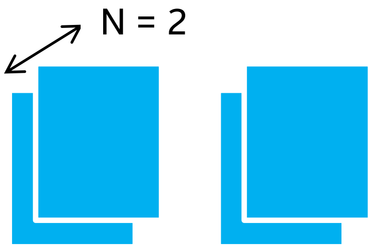
Output or activations



Compute Kernels



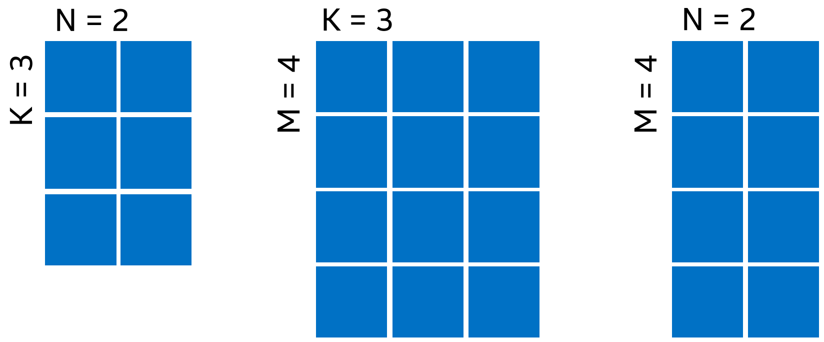
Fully connected layer



$\boldsymbol{I} \in \boldsymbol{R}^{KxN}$
Input

$\boldsymbol{W} \in \boldsymbol{R}^{MxK}$
Weights or model

$\boldsymbol{O} \in \boldsymbol{R}^{MxN}$
Output or activations



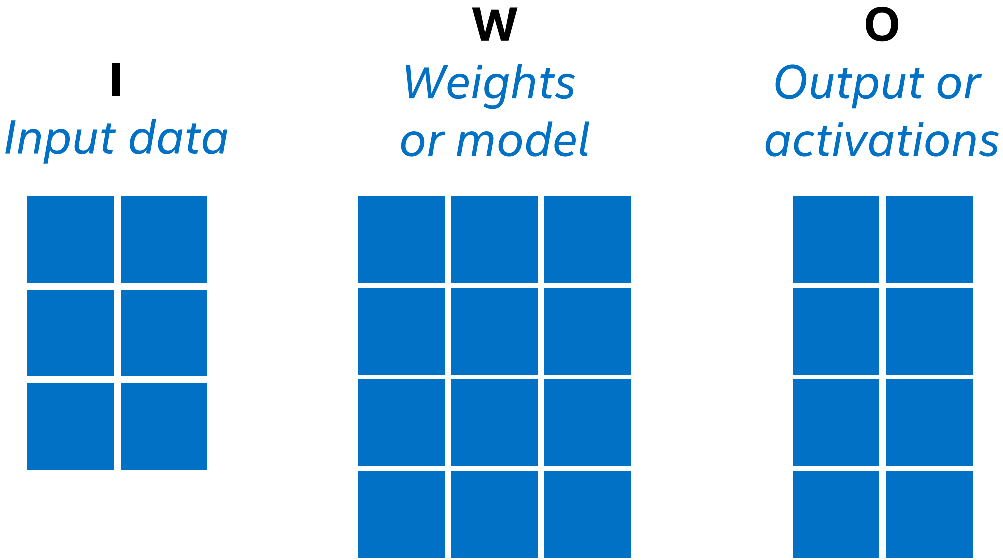
Forward propagation: (M x K) * (K x N)

Backward propagation: $(M \times K)^T$ * (M x N)

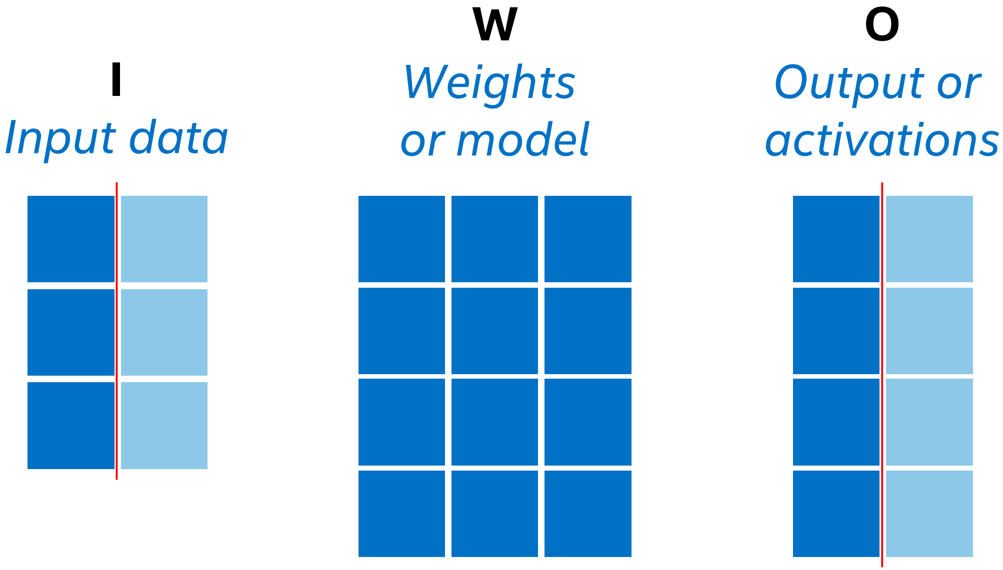
Weight update: (M x N) * $(K \times N)^T$

IDF16 INTEL DEVELOPER FORUM

Parallelism Options

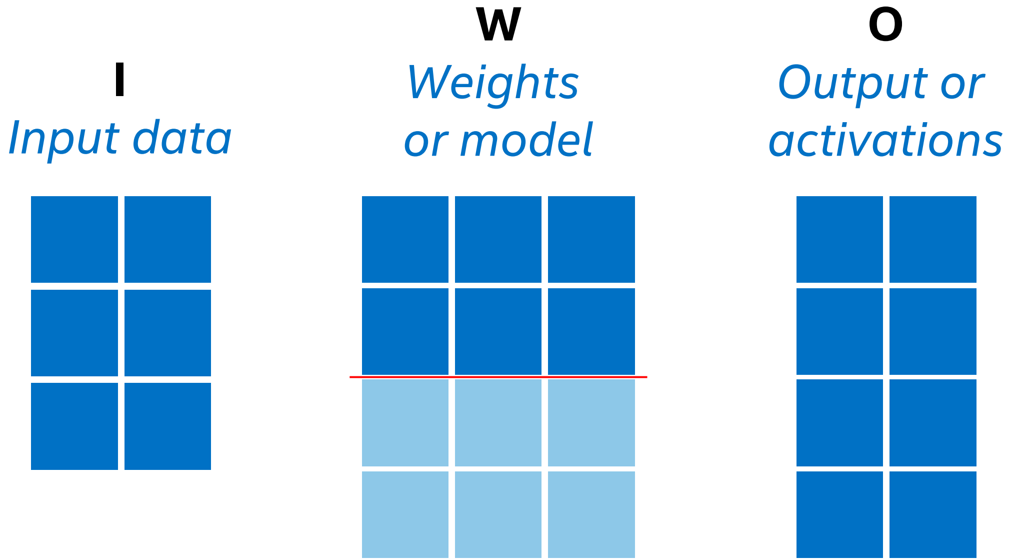


Parallelism Options



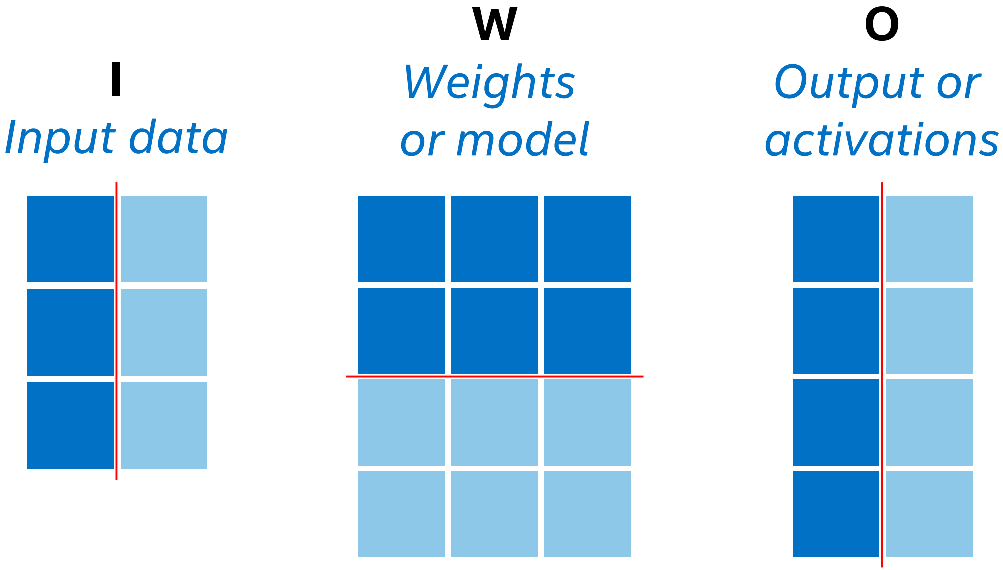
Data

Parallelism Options



Model

Parallelism Options



Hybrid

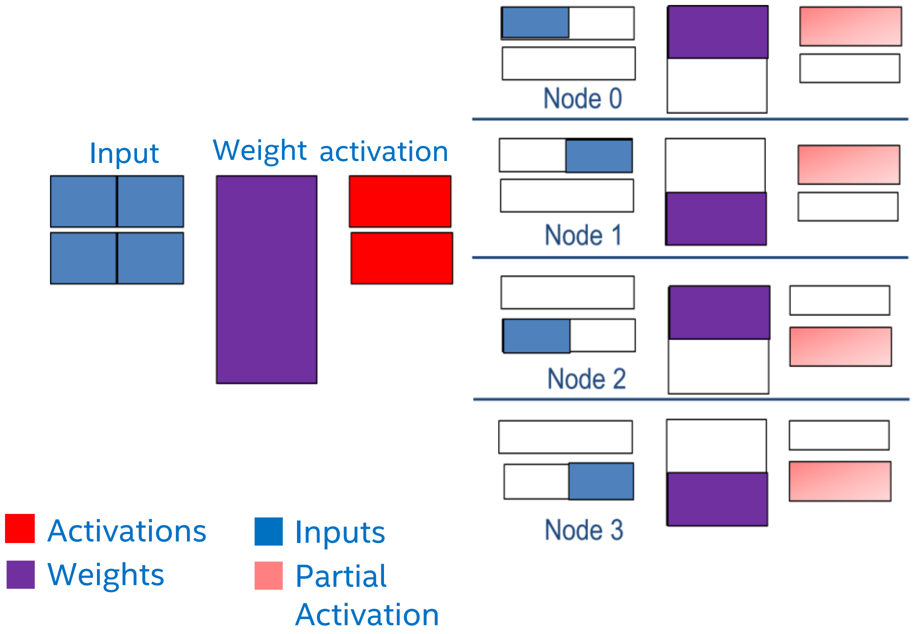
Data/Model Parallelism at Scale

- General rule of thumb
  - Use data parallelism when activations > weights
  - Use model parallelism when weights > activations

- Implications of data and model parallelism
  - Data parallelism at scale makes activations << weights
  - Model parallelism at scale makes weights << activations
  - Compute efficiency goes down due to skewed matrices
  - Communication time dominates at scale

Addressing the scaling challenge

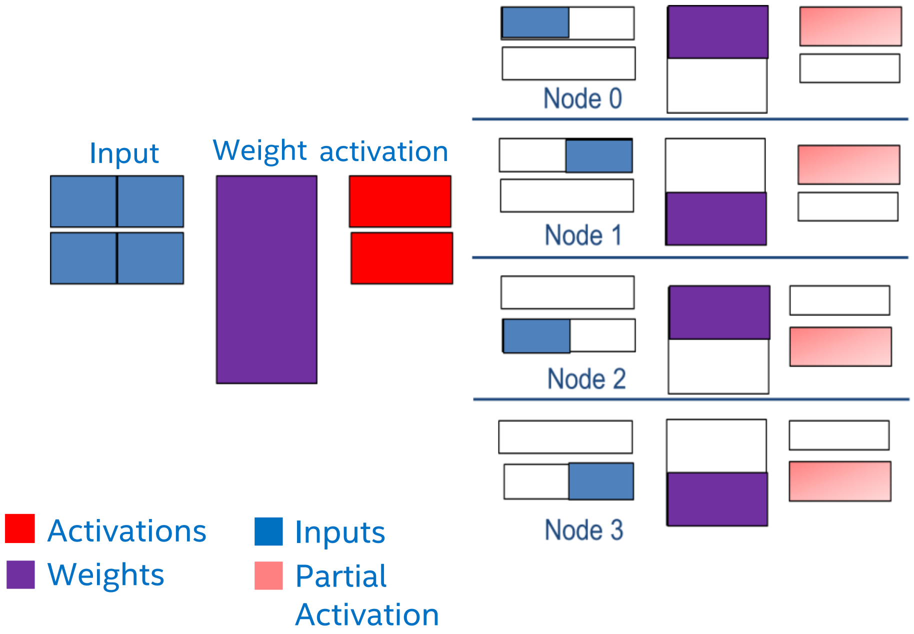
- Hybrid parallelism to improve compute efficiency
  - Partition across activations and weights to minimize skewed matrices



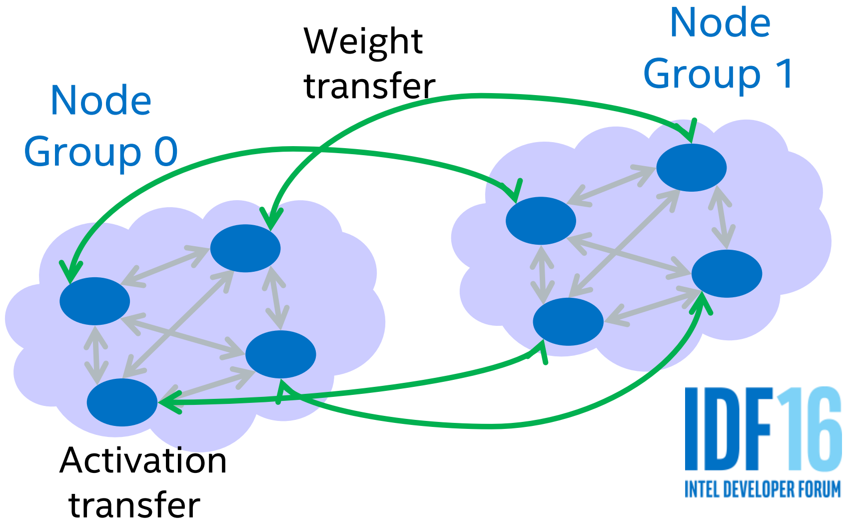
IDF16
INTEL DEVELOPER FORUM

Addressing the scaling challenge

- Hybrid parallelism to improve compute efficiency
  - Partition across activations and weights to minimize skewed matrices

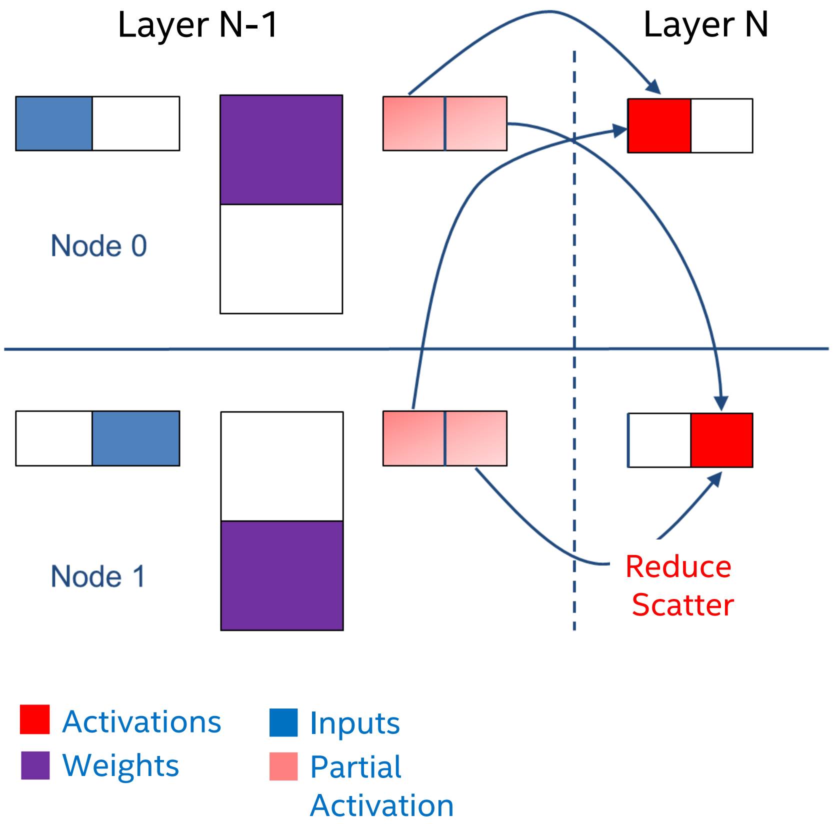


- Node groups to improve communication efficiency
  - Avoid global transfer of activations and weights via node groups
    - Activations transfer within a group
    - Weight transfer across groups



IDF16 INTEL DEVELOPER FORUM

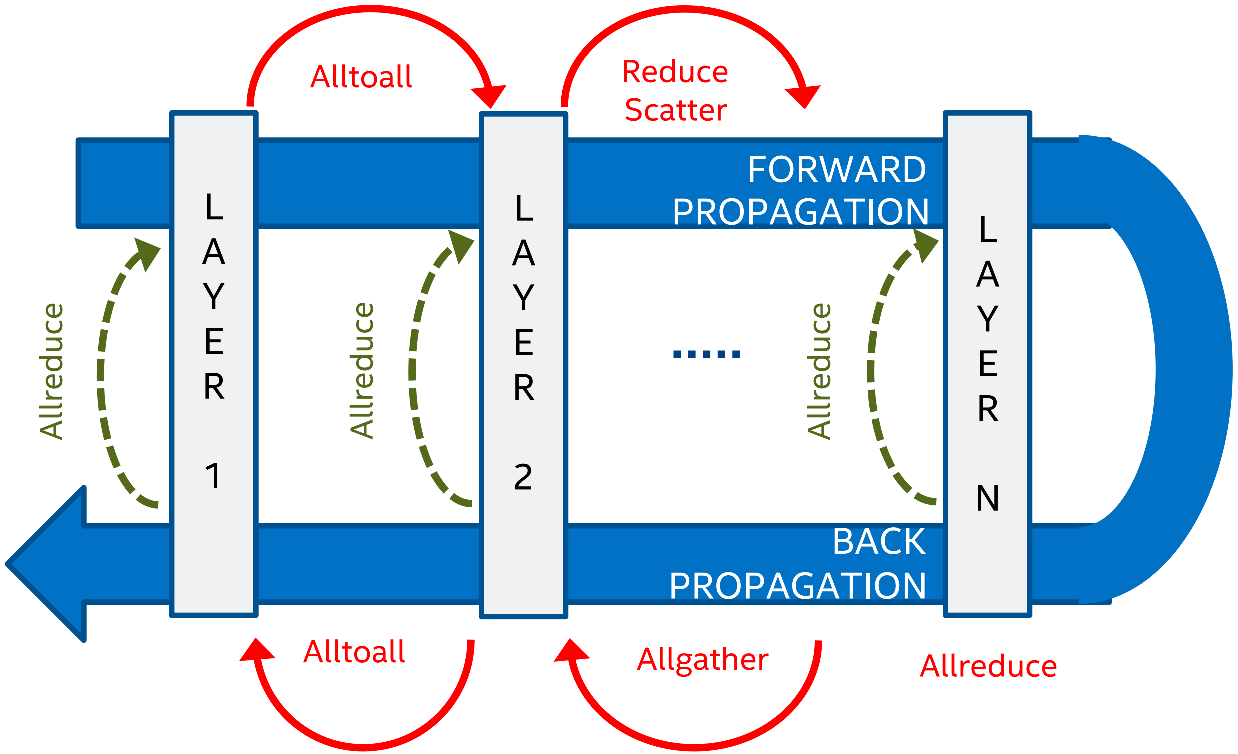
Communication Patterns in Deep Learning



Reduce the activations from layer N-1 and scatter at layer N

Common MPI collectives in DL
- Reduce Scatter
- AllGather
- AllReduce
- AlltoAll

Communication Patterns in Deep Learning … contd.

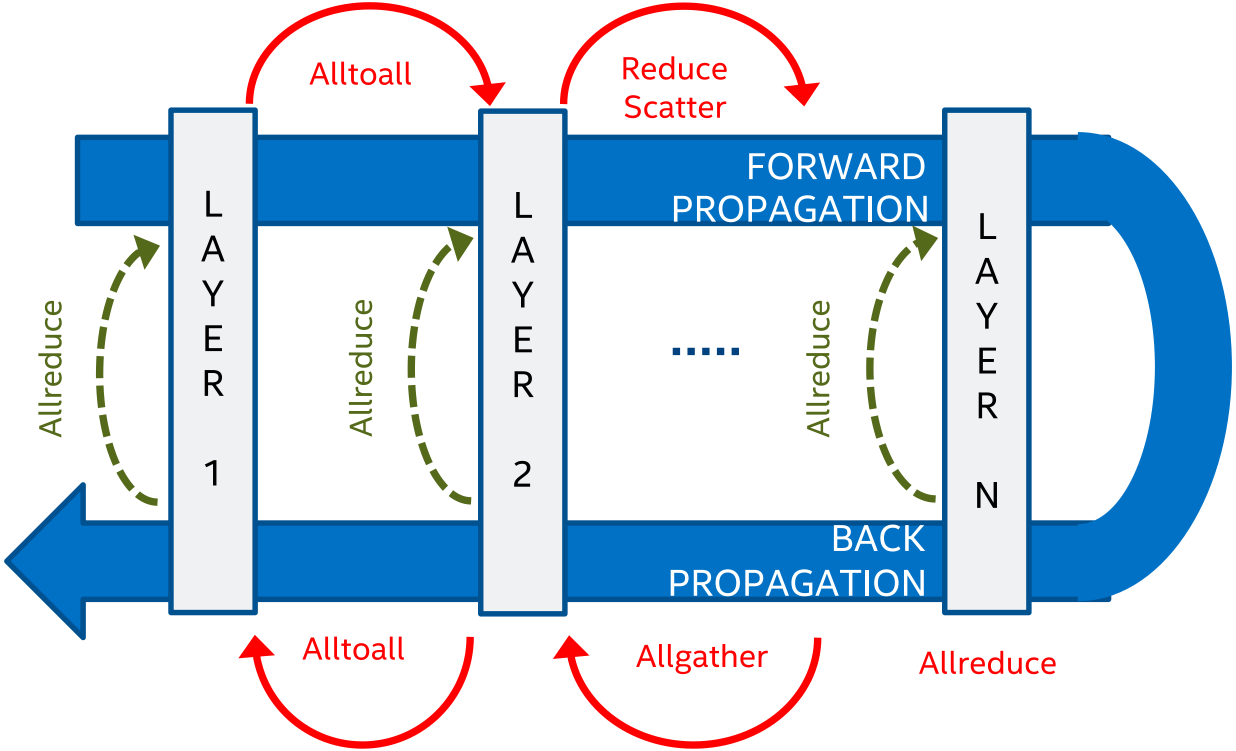


Activations (required immediately in next layer)

Updated weights (required during forward propagation of the corresponding layer)

IDF16 INTEL DEVELOPER FORUM

Communication Patterns in Deep Learning … contd.



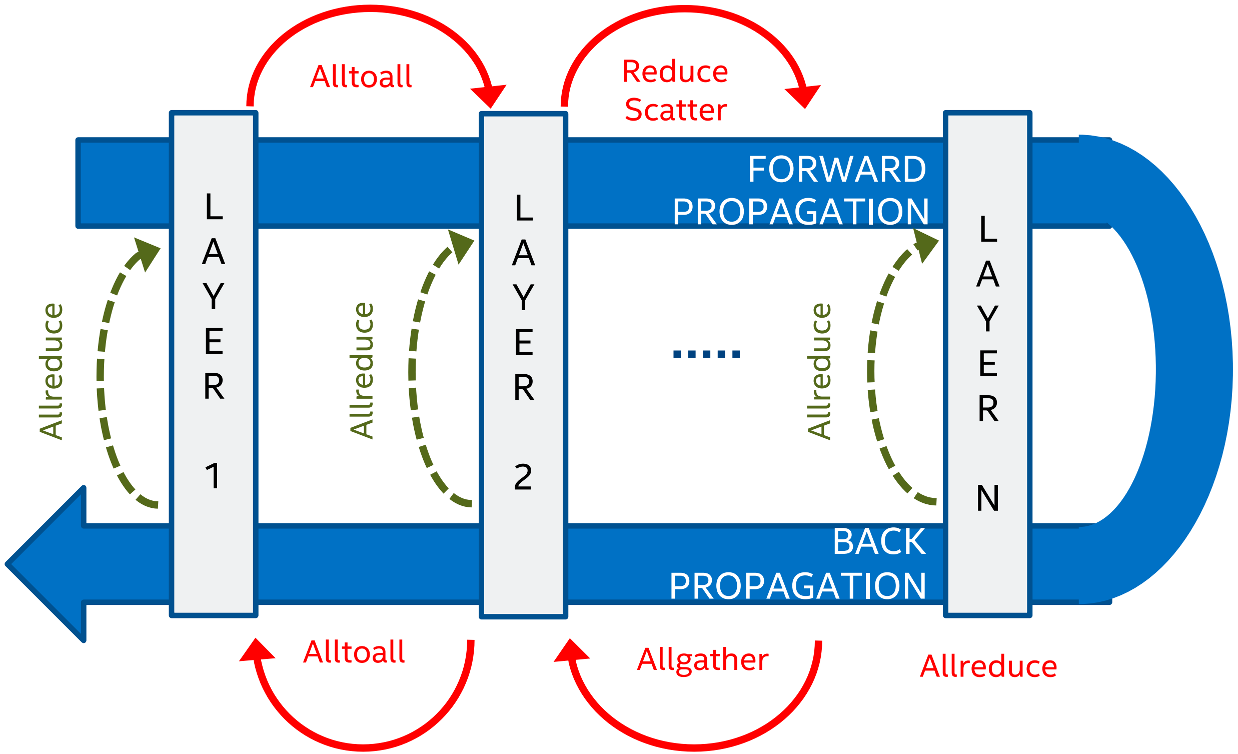
1. Optimized Collectives

Activations (required immediately in next layer)

Updated weights (required during forward propagation of the corresponding layer)

IDF16
INTEL DEVELOPER FORUM

Communication Patterns in Deep Learning … contd.

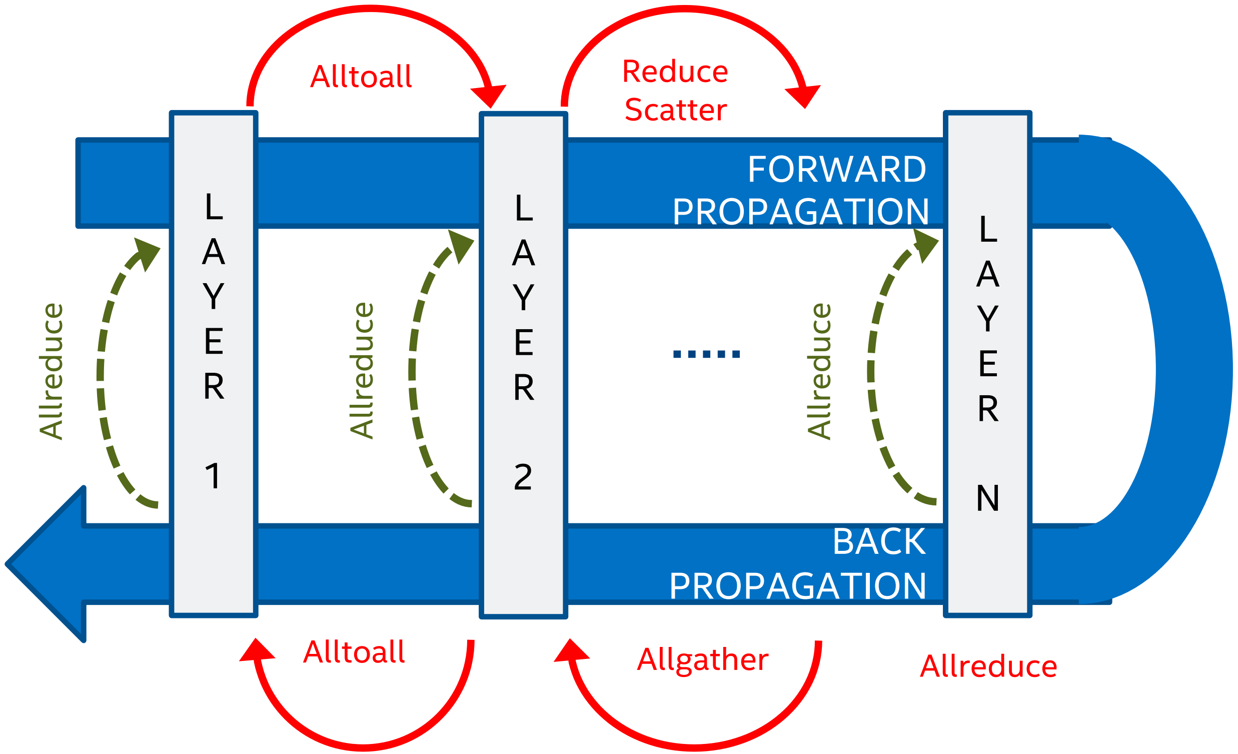


1. Optimized Collectives
2. Compute Communication Overlap

Activations (required immediately in next layer)

Updated weights (required during forward propagation of the corresponding layer)

Communication Patterns in Deep Learning … contd.



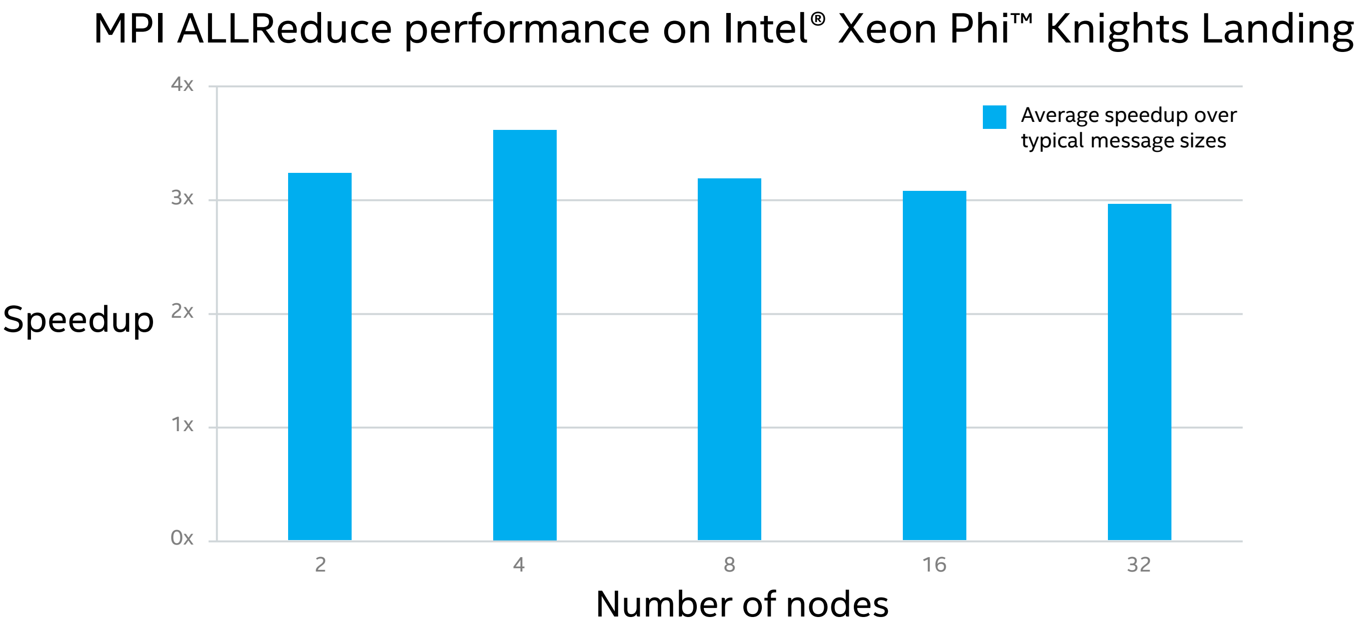
1. Optimized Collectives
2. Compute Communication Overlap
3. Smart Message and Task Scheduling

Activations (required immediately in next layer)

Updated weights (required during forward propagation of the corresponding layer)

IDF16 INTEL DEVELOPER FORUM

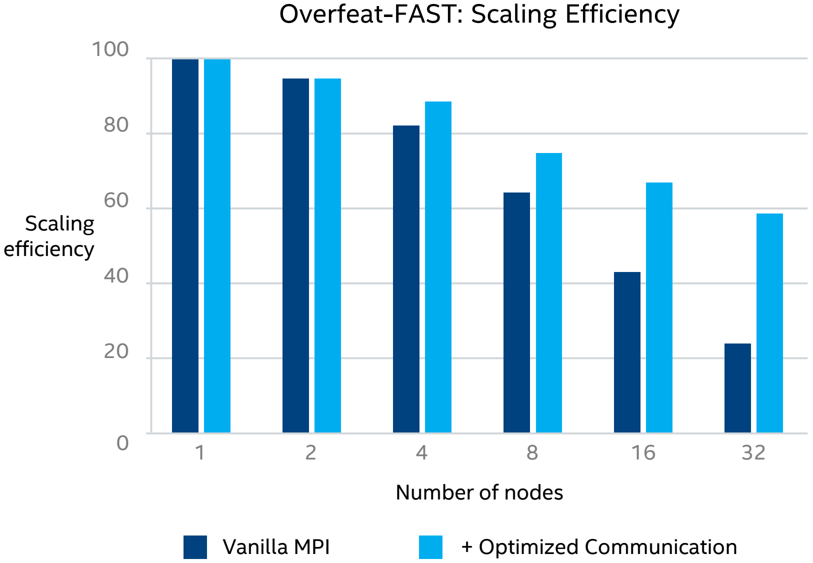
Scaling Deep Learning Communication Primitives

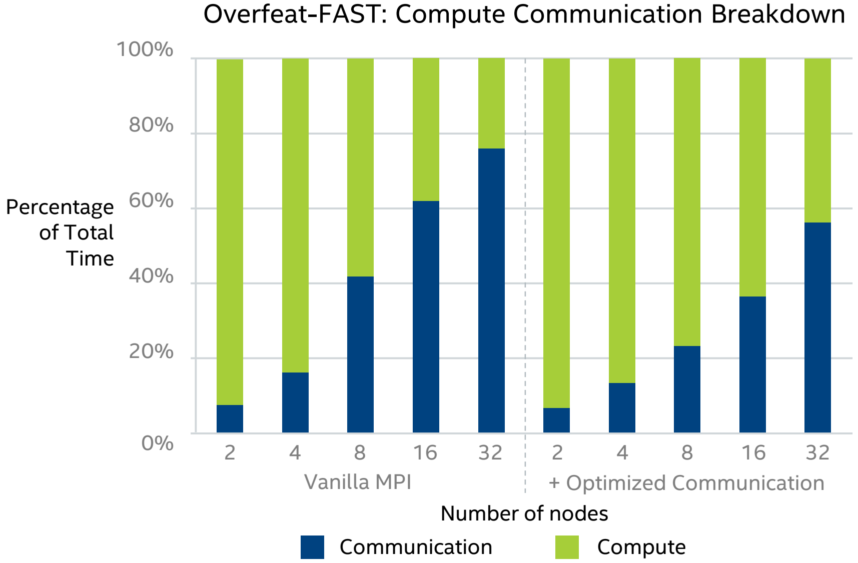


Deep learning specific optimizations result in3X speedup for the Allreduce collective
(average for the message profile of16KB – 16MB floats)

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 http://www.intel.com/performance/datacenter
Configuration: Intel® Xeon Phi™ Processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM), 96 GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Intel® Omni-Path Host Fabric Interface Adapter 100 Series 1 Port, Red Hat* Enterprise Linux 6.7, Intel® ICC version 16.0.2, Intel® MPI Library 5.1.3 for Linux.

Benefits of Multinode Optimizations





Higher is better

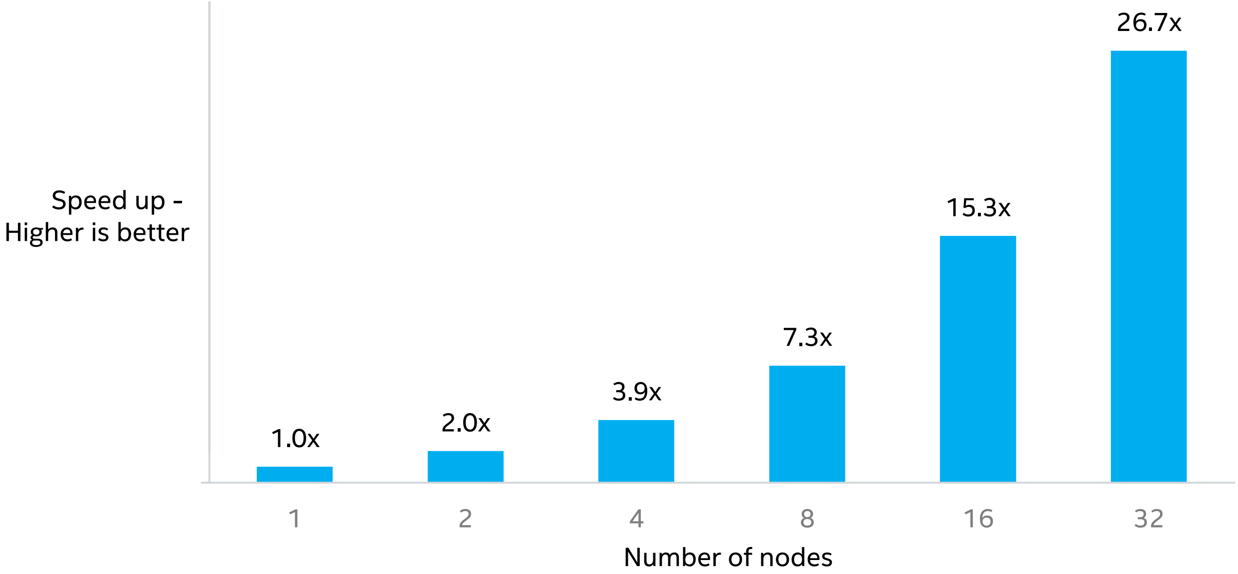
Higher % of compute (green) is better

Scaling efficiency without multi-node optimizations drops 1.3-2.1X for large node counts

Configuration: Intel® Xeon Phi™ Processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM), 96 GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Intel® Omni-Path Host Fabric Interface Adapter 100 Series 1 Port, Red Hat* Enterprise Linux 6.7, Intel® ICC version 16.0.2, Intel® MPI Library 5.1.3 for Linux, Intel® Optimized DNN Framework.

IDF16 INTEL DEVELOPER FORUM

Scaling Training Time of a common Neural Network



Topology: **AlexNet*** Dataset: **Large image database**

With convergence and I/O overhead included

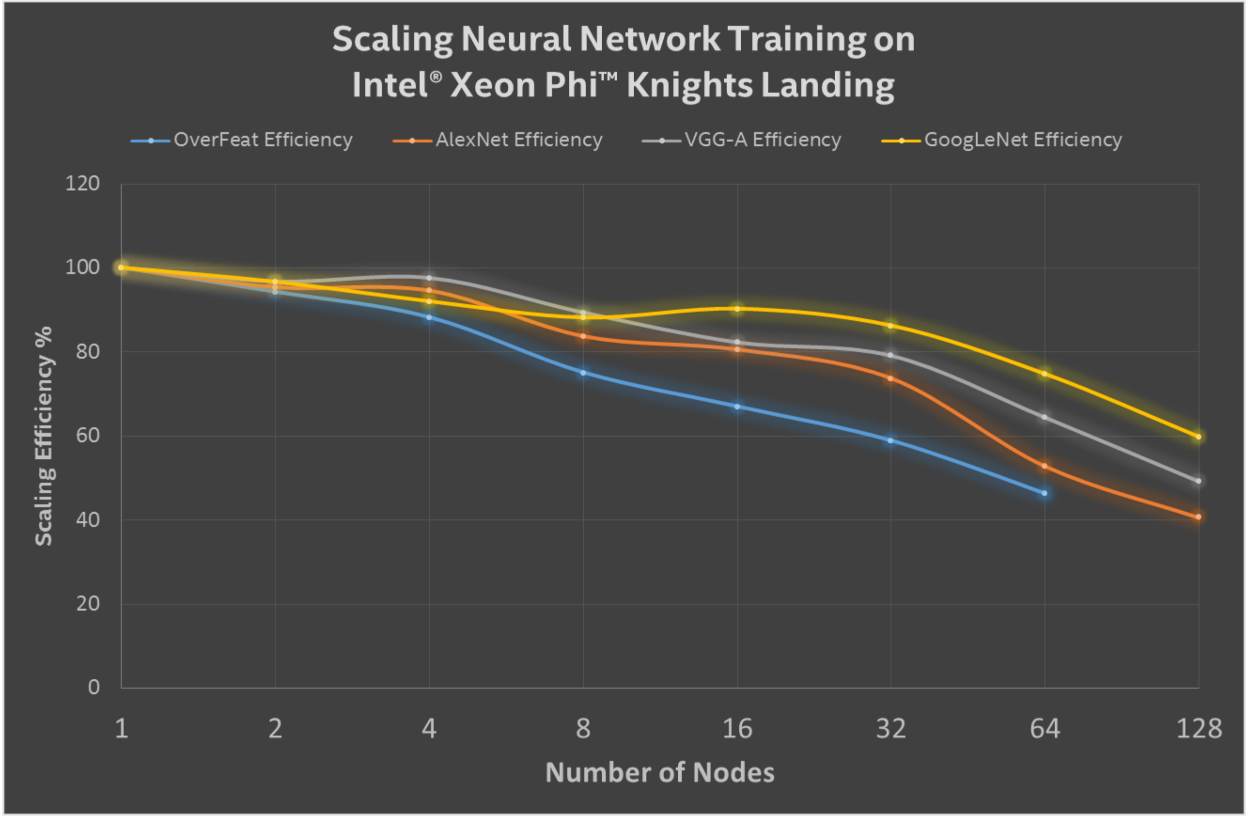
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors 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 http://www.intel.com/performance/datacenter.

Configurations: Up to 27x faster training on 32-node as compared to single-node based on AlexNet* topology workload (batch size = 1024) running one node Intel Xeon Phi processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM) in Intel® Server System LADMP2312KXXX41, 96 GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat Enterprise Linux* 6.7 (Santiago), Intel® ICC version 16.0.2, Intel® MPI Library 5.1.3 for Linux, running Intel® Optimized DNN Framework compared to identically configured 32-node system with Intel® Omni-Path Host Fabric Interface Adapter 100 Series 1 Port PCIe x16 connectors. Image database in 25TB Luster file system accessed using same Intel® Omni-Path Host Fabric Interface Adapter 100 Series 1 Port PCIe x16 connectors.. Contact your Intel representative for more information on how to obtain the binary. For information on workload, see https://papers.nips.cc/paper/4824-Large image database-classification-with-deep-convolutional-neural-networks.pdf.

IDF16 INTEL DEVELOPER FORUM

* https://github.com/soumith/convnet-benchmarks/blob/master/caffe/imagenet_winners/alexnet.prototxt

Scaling efficiently popular neural network topologies

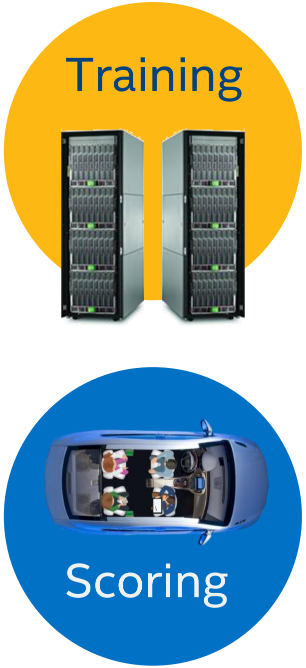


Dataset: **Large image database**

Without convergence and I/O overhead

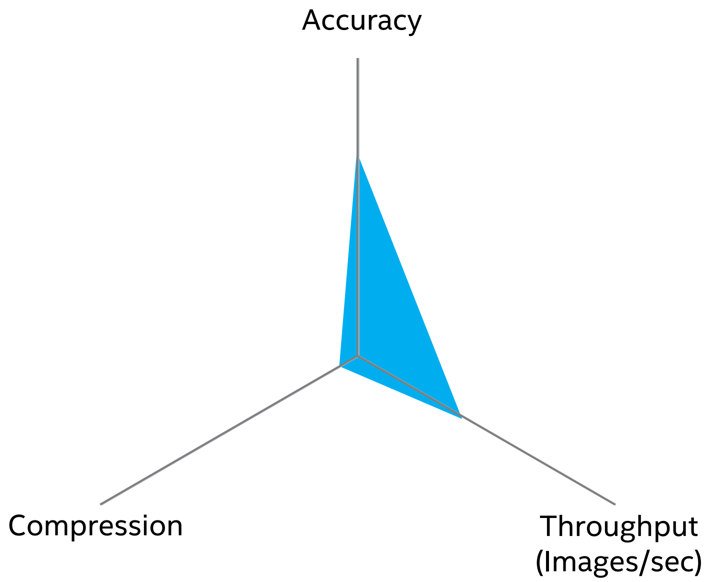
Configuration: Intel® Xeon Phi™ Processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM), 96 GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Intel® Omni-Path Host Fabric Interface Adapter 100 Series 1 Port, Red Hat* Enterprise Linux 6.7, Intel® ICC version 16.0.2, Intel® MPI Library 5.1.3 for Linux, Intel® Optimized DNN Framework.

The Virtuous Cycle of Compute

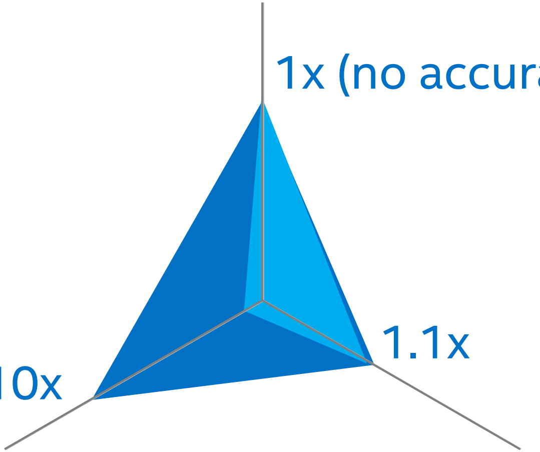


The Virtuous Cycle of Compute

Delivering Trained Model to Edge Device



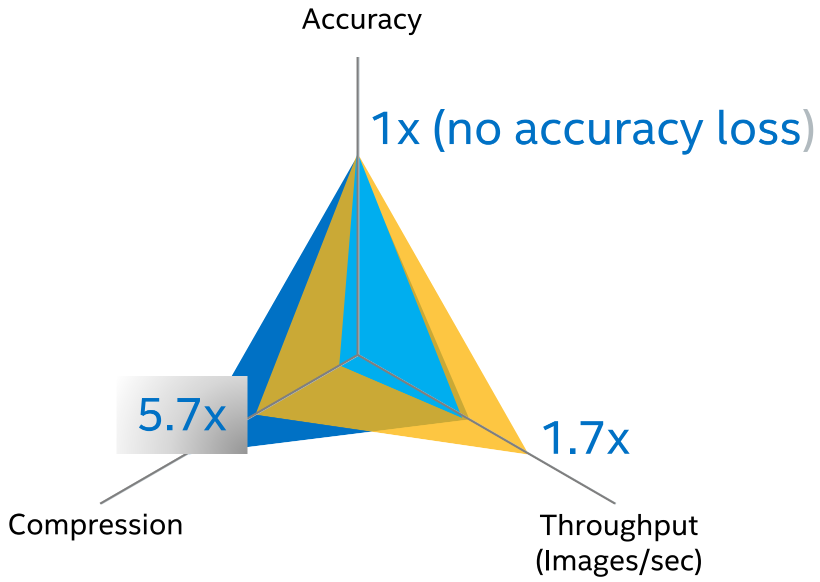
Delivering Trained Model to Target Device†



Sparsifying FC layers (e.g., Deep Compression*) -> mobile

† http://arxiv.org/abs/1608.01409
* http://arxiv.org/abs/1510.00149
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 http://www.intel.com/performance/datacenter. Configurations: Intel® Xeon® E5-2697 v4, with 2 socket, 18 cores per socket, 45MB LLC, running at 2.3GHz to train Calffe/AlexNet. For information on workload, see https://papers.nips.cc/paper/4824-Large image database-classification-with-deep-convolutional-neural-networks.pdf.

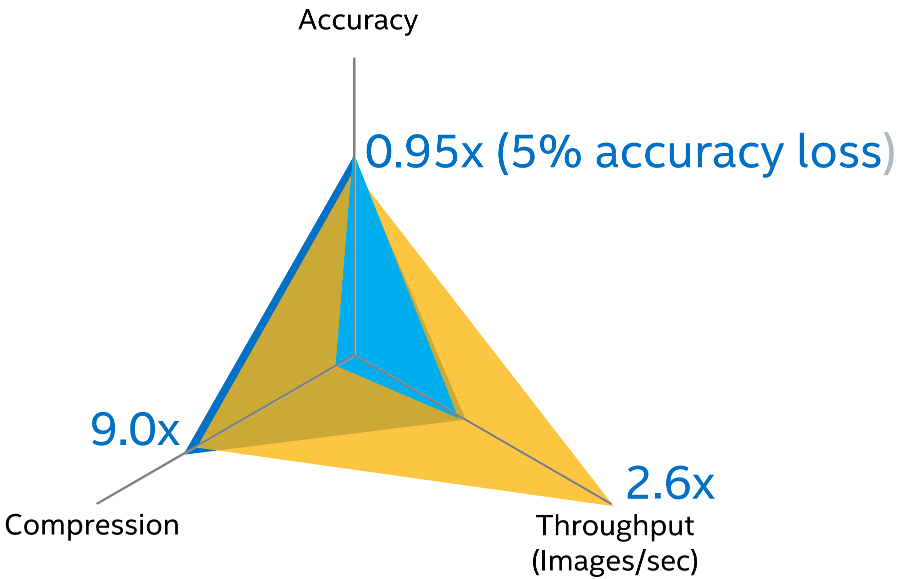
Delivering Trained Model to Target Device †



Balanced sparsifying of Conv and FC layers → automotives

† http://arxiv.org/abs/1608.01409
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 http://www.intel.com/performance/datacenter. Configurations: Intel® Xeon® E5-2697 v4, with 2 socket, 18 cores per socket, 45MB LLC, running at 2.3GHz to train Cafffe/AlexNet. For information on workload, see https://papers.nips.cc/paper/4824-Large image database-classification-with-deep-convolutional-neural-networks.pdf.

Delivering Trained Model to Target Device †



† http://arxiv.org/abs/1608.01409

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 http://www.intel.com/performance/datacenter. Configurations: Intel® Xeon® E5-2697 v4, with 2 socket, 18 cores per socket, 45MB LLC, running at 2.3GHz to train Cafffe/AlexNet. For information on workload, see https://papers.nips.cc/paper/4824-Large image database-classification-with-deep-convolutional-neural-networks.pdf.

Agenda

- Why do we need to scale machine learning
- What makes it hard to scale and how we are addressing it
- Real-world applications
  - Introducing Dr. Amir Khosrowshahi
- Hardware roadmap, software tools and frameworks update
- Summary

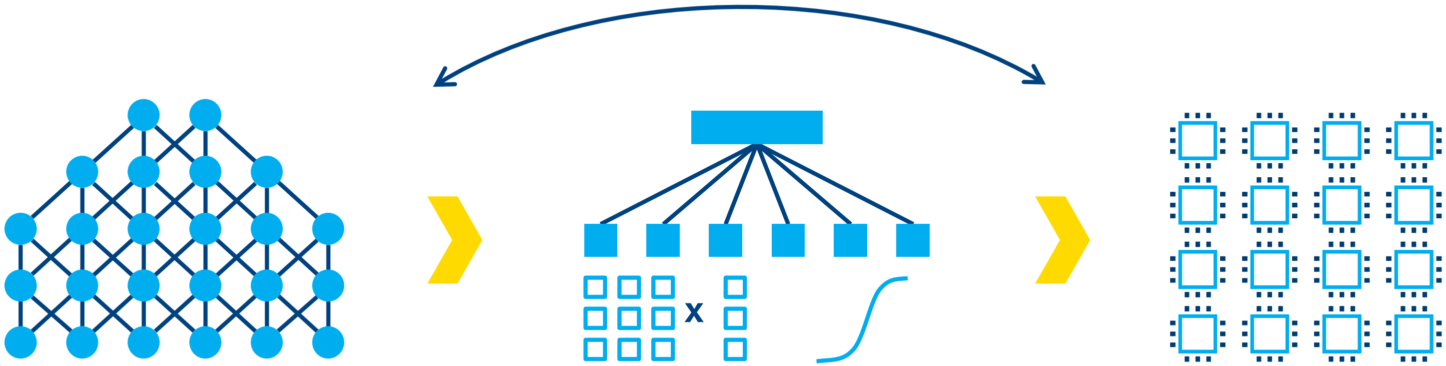
Introducing Dr. Amir Khosrowshahi

CTO, Nervana Systems

About Nervana

A platform for machine intelligence

- Enable deep learning at scale
- Optimized from algorithms to silicon



IDF16 INTEL DEVELOPER FORUM

Application Areas

Healthcare

Agriculture

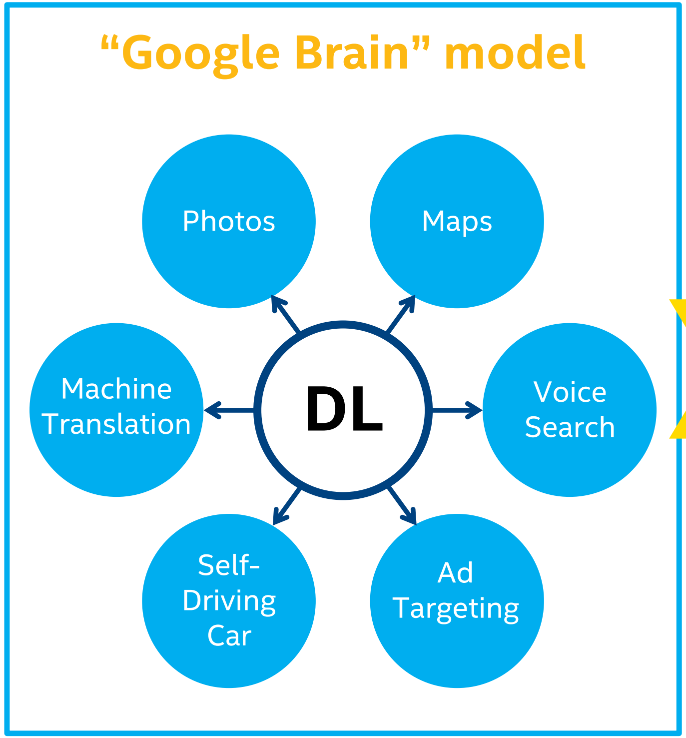
Finance

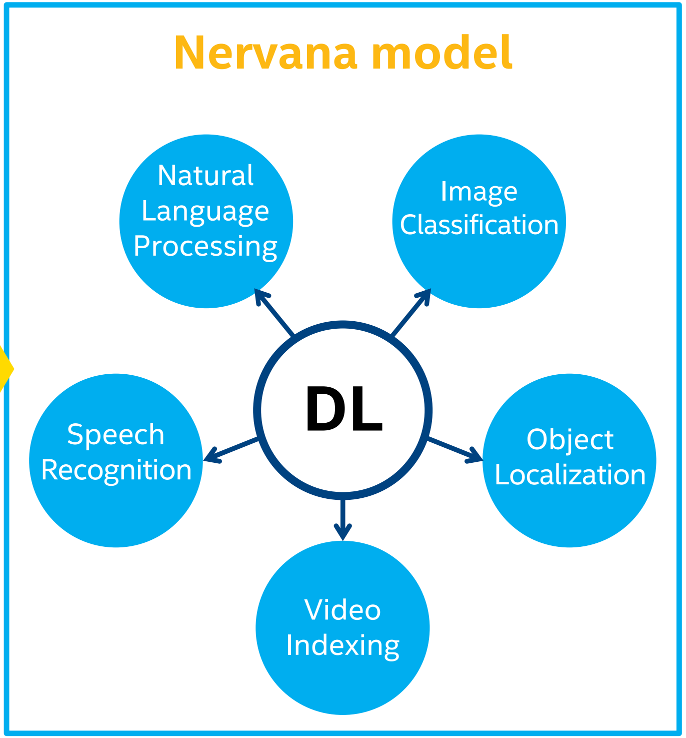
Online Services

Automotive

Energy

Deep Learning as a Core Technology





Nervana Cloud

Data

Images

Video

Text

Speech

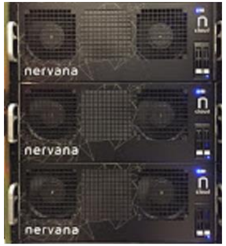
Tabular

Time Series

Cloud

Import

Build

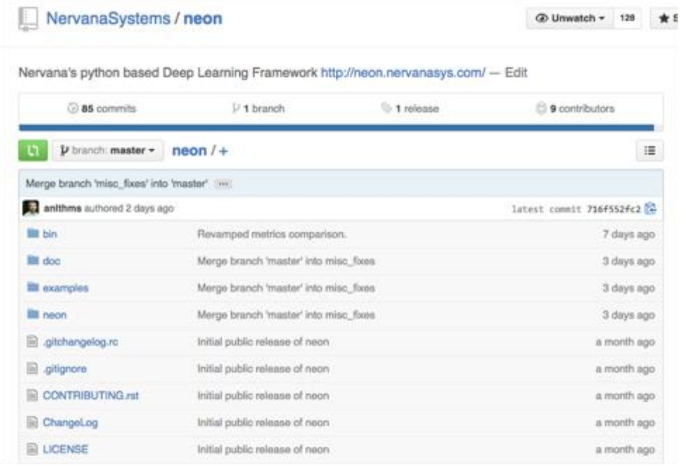


Train

Deploy

Neon: Nervana Python* Deep Learning Library

- User-friendly, extensible, abstracts parallelism
- Support for many deep learning models
- Interface to **Nervana** cloud
- Supports multiple backends
- Integrates with large cloud datastores
- Core routines written in assembler



See github for details

Agenda

- Why do we need to scale machine learning
- What makes it hard to scale and how we are addressing it
- Real-world experience of an industry leader
- Hardware roadmap, software tools and frameworks update
- Summary

Intel® Xeon Phi™ Processor Family for Performance

Enables shorter time to train

Breakthrough Highly-Parallel Performance

- Up to ~6 SGEMM TFLOPs[1] per socket
- 1.38x[2] better scaling efficiency resulting in lower time to train for multi-node
- Eliminates add-in card PCIe* offload bottleneck and utilization constraints

Removes Barriers through Integration

- Integrated Intel® Omni-Path fabric (dual-port; 50 GB/s) increases price-performance and reduces communication latency for deep learning networks

Better Programmability

- Binary-compatible with Intel® Xeon® processors
- Open standards, libraries and frameworks

1.. *Up to 6 SP TFLOPS based on the Intel Xeon Phi processor peak theoretical single-precision performance (FLOPS = cores x clock frequency x single-precision floating-point operations per second per cycle).*
2. Up to 38% better scaling efficiency at 32-nodes claim based on GoogLeNet deep learning image classification training topology using a large image database comparing one node Intel Xeon Phi processor 7250 (16 GB, 1.4 GHz, 68 Cores) in Intel® Server System LADMP2312KXXX41, DDR4 96GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat* Enterprise Linux 6.7, Intel® Optimized DNN Framework with 87% efficiency to unknown hosts running 32 each NVIDIA Tesla* K20 GPUs with a 62% efficiency (Source: http://arxiv.org/pdf/1511.00175v2.pdf showing FireCaffe* with 32 each NVIDIA Tesla* K20s (Titan Supercomputer*) running GoogLeNet* at 20x speedup over Caffe* with 1 each K20).

Intel® Xeon Phi™ Processor Family for Performance

Enables shorter time to train

Breakthrough Highly-Parallel Performance

- Up to ~6 SGEMM TFLOPs[1] per socket
- 1.38x[2] better scaling efficiency resulting in lower time to train for multi-node
- Eliminates add-in card PCIe* offload bottleneck and utilization constraints

Removes Barriers through Integration

- Integrated Intel® Omni-Path fabric (dual-port; 50 GB/s) increases price-performance and reduces communication latency for deep learning networks

Better Programmability

- Binary-compatible with Intel® Xeon® processors
- Open standards, libraries and frameworks

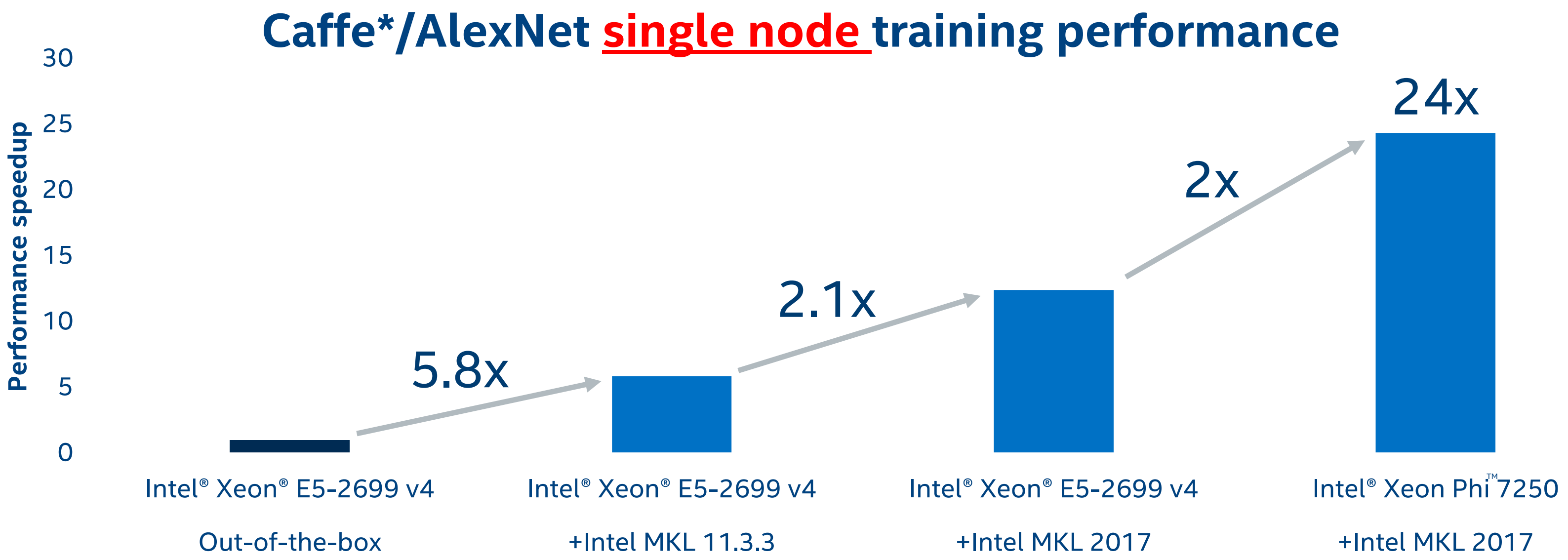
For an exciting new Intel® Xeon Phi™ roadmap update for machine learning/AI: Please attend Intel EVP Diane Bryant's Keynote tomorrow, Aug 17, 9am

1.. *Up to 6 SP TFLOPS based on the Intel Xeon Phi processor peak theoretical single-precision performance (FLOPS = cores x clock frequency x single-precision floating-point operations per second per cycle).*
2. Up to 38% better scaling efficiency at 32-nodes claim based on GoogLeNet deep learning image classification training topology using a large image database comparing one node Intel Xeon Phi processor 7250 (16 GB, 1.4 GHz, 68 Cores) in Intel® Server System LADMP2312KXXX41, DDR4 96GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat* Enterprise Linux 6.7, Intel® Optimized DNN Framework with 87% efficiency to unknown hosts running 32 each NVIDIA Tesla* K20 GPUs with a 62% efficiency (Source: http://arxiv.org/pdf/1511.00175v2.pdf showing FireCaffe* with 32 each NVIDIA Tesla* K20s (Titan Supercomputer*) running GoogLeNet* at 20x speedup over Caffe* with 1 each K20).

IDF16 INTEL DEVELOPER FORUM

Better performance in Deep Neural Network workloads with Intel® Math Kernel Library (**Intel® MKL**)

Better performance in Deep Neural Network workloads with Intel® Math Kernel Library (**Intel® MKL**)



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 http://www.intel.com/performance . *Other names and brands may be property of others

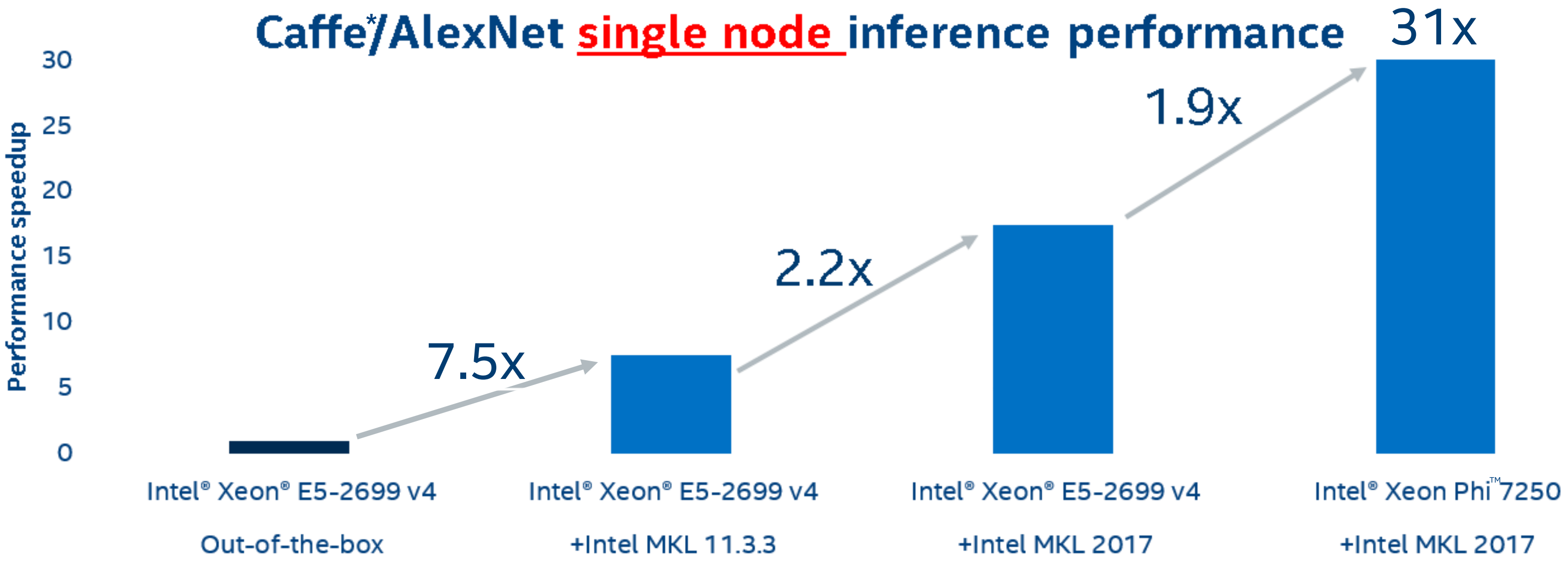
Configurations:

- 2 socket system with Intel® Xeon Processor E5-2699 v4 (22 Cores, 2.2 GHz,), 128 GB memory, Red Hat* Enterprise Linux 6.7, BVLC Caffe, Intel Optimized Caffe framework, Intel® MKL 11.3.3, Intel® MKL 2017
- Intel® Xeon Phi™ Processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM), 128 GB memory, Red Hat* Enterprise Linux 6.7, Intel® Optimized Caffe framework, Intel® MKL 2017

All numbers measured without taking data manipulation into account.

IDF16 INTEL DEVELOPER FORUM

Better performance in Deep Neural Network workloads with **Intel® Math Kernel Library (Intel® MKL)**

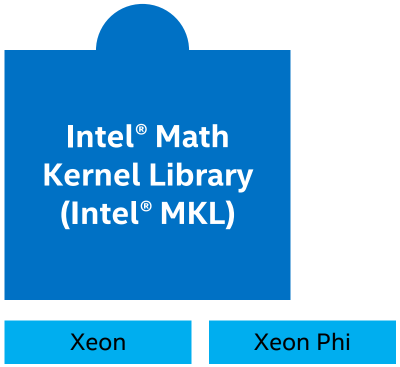


Configurations:

- 2 socket system with Intel® Xeon® Processor E5-2699 v4 (22 Cores, 2.2 GHz,), 128 GB memory, Red Hat* Enterprise Linux 6.7, BVLC Caffe, Intel Optimized Caffe framework, Intel® MKL 11.3.3, Intel® MKL 2017
- Intel® Xeon Phi™ Processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM), 128 GB memory, Red Hat* Enterprise Linux 6.7, Intel® Optimized Caffe framework, Intel® MKL 2017

All numbers measured without taking data manipulation into account.

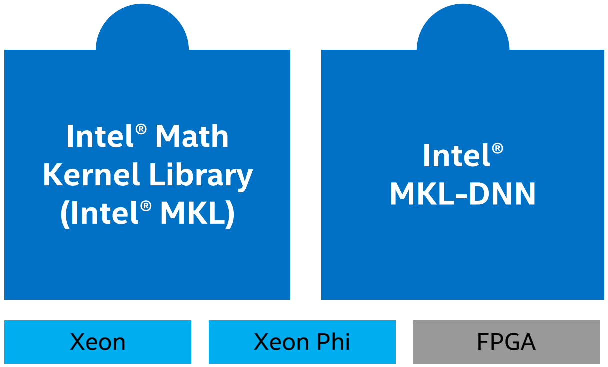
Intel Deep Learning Software Stack and Timeline



Intel MKL is SW building block to extract max Intel HW performance and provide common interface to all Intel accelerators.

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

Intel Deep Learning Software Stack and Timeline



Intel MKL-DNN is an open source IA optimized DNN APIs, combined with Intel® MKL and build tools designed for scalable, high-velocity integration with ML/DL frameworks.

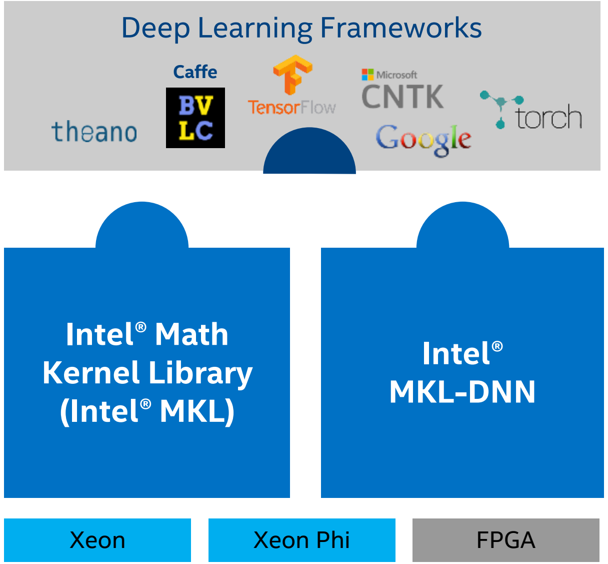
Targeted release: Q3 2016

Includes:

- Open Source implementations of new DNN functionality included in MKL 2017 Beta, new algorithms ahead of MKL releases
- IA optimizations contributed by community

Intel MKL is SW building block to extract max Intel HW performance and provide common interface to all Intel accelerators.

Intel Deep Learning Software Stack and Timeline



Popular Deep Learning frameworks

Intel MKL-DNN is an open source IA optimized DNN APIs, combined with Intel® MKL and build tools designed for scalable, high-velocity integration with ML/DL frameworks.

Targeted release: Q3 2016

Includes:

- Open Source implementations of new DNN functionality included in MKL 2017 Beta, new algorithms ahead of MKL releases
- IA optimizations contributed by community

Intel MKL is SW building block to extract max Intel HW performance and provide common interface to all Intel accelerators.

- **Multi-Node scaling for Knights Landing: Caffe* by EoY and 1H'17 in other frameworks**

Intel Deep Learning Software Stack and Timeline

Intel Deep Learning Tools

Deep Learning Frameworks

theano Caffe BVLC TensorFlow Microsoft CNTK Google torch

Intel® Math Kernel Library (Intel® MKL)

Intel® MKL-DNN

Xeon | Xeon Phi | FPGA

Tools to accelerate design, training and deployment of deep learning solutions

Targeted release: Q3'2016

Popular Deep Learning frameworks

Intel MKL-DNN is an open source IA optimized DNN APIs, combined with Intel® MKL and build tools designed for scalable, high-velocity integration with ML/DL frameworks.

Targeted release: Q3 2016

Includes:

- Open Source implementations of new DNN functionality included in MKL 2017 Beta, new algorithms ahead of MKL releases
- IA optimizations contributed by community

Intel MKL is SW building block to extract max Intel HW performance and provide common interface to all Intel accelerators.

- **Multi-Node scaling for Knights Landing: Caffe* by EoY and 1H'17 in other frameworks**
- **Intel Deep Learning Tools with support for model compression by end of 2016**

IDF16 INTEL DEVELOPER FORUM

Call to action

- Machine learning is the key enabler for a new virtuous cycle of compute triggered by explosion of digital data and ubiquitous connectivity
- It can vastly expand the reach of computing for applications like self-driving, agriculture, health and manufacturing
- Help machine learning unlock the true potential of AI
- Consider the full, end-to-end pipeline when you think about your AI needs.
- Try Intel MKL, Intel optimized frameworks & Intel Xeon Phi

Summary

- Machine learning must scale out to bring down the training time of weeks/days to days/hours
- Machine learning compute infrastructure must be both performant & productive for developers, and leverage the efficiency of cloud
- Scaling distributed machine learning is challenging as it pushes the limits of available data/model parallelism and internode communication
- Intel's new deep learning tools -- with the upcoming integration of Nervana cloud stack -- are designed to hide/reduce the complexity of strong scaling time-to-train and model deployment tradeoffs on resource-constrained edge devices without compromising the performance need

Related Tech Sessions

ANATS01: Deep Learning Frameworks and Optimization Paths on Intel® Architecture
By Andres Rodriguez, Panchumarthy, Ravi, and Tom "Elvis" Jones, Amazon

ANATS03: Enabling an End to End Architecture for Autonomous Vehicles
By Jack Weast

ANATS05: How to Parallelize Neural Networks (xNNs) for Intel® Xeon Phi™
By Nadathur R. Satish

For more information on machine learning at Intel: intel.com/machinelearning

A PDF of this presentation is available is available from our Technical Session Catalog: www.intel.com/idfsessionsSF. This URL is also printed on the top of Session Agenda Pages in the Pocket Guide.

Legal Notices and Disclaimers

IDF16
INTEL DEVELOPER FORUM

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 SSE3 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

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the second quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as "anticipates," "expects," "intends," "plans," "believes," "seeks," "estimates," "may," "will," "should" and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be important factors that could cause actual results to differ materially from the company's expectations. Demand for Intel's products is highly variable and could differ from expectations due to factors including changes in business and economic conditions; consumer confidence or income levels; the introduction, availability and market acceptance of Intel's products, products used together with Intel products and competitors' products; competitive and pricing pressures, including actions taken by competitors; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel's gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; and product manufacturing quality/yields. Variations in gross margin may also be caused by the timing of Intel product introductions and related expenses, including marketing expenses, and Intel's ability to respond quickly to technological developments and to introduce new products or incorporate new features into existing products, which may result in restructuring and asset impairment charges. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Results may also be affected by the formal or informal imposition by countries of new or revised export and/or import and doing-business regulations, which could be changed without prior notice. Intel operates in highly competitive industries and its operations have high costs that are either fixed or difficult to reduce in the short term. The amount, timing and execution of Intel's stock repurchase program could be affected by changes in Intel's priorities for the use of cash, such as operational spending, capital spending, acquisitions, and as a result of changes to Intel's cash flows or changes in tax laws. Product defects or errata (deviations from published specifications) may adversely impact our expenses, revenues and reputation. Intel's results could be affected by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel's ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. Intel's results may be affected by the timing of closing of acquisitions, divestitures and other significant transactions. A detailed discussion of these and other factors that could affect Intel's results is included in Intel's SEC filings, including the company's most recent reports on Form 10-Q, Form 10-K and earnings release.

Rev. 4/14/15