



THE RACE FOR FASTER MACHINE LEARNING - INTEL ARTIFICIAL INTELLIGENCE TECHNICAL UPDATE

Paweł Gepner

Robert Adamski

Pascal Lassaigue

Legal Notices and Disclaimers

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life-saving, life-sustaining, critical control or safety systems, or in nuclear facility applications.

Intel products 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.

Intel may make changes to dates, specifications, product descriptions, and plans referenced in this document at any time, without notice.

This document may contain information on products in the design phase of development. The information herein is subject to change without notice. Do not finalize a design with this information.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel Corporation or its subsidiaries in the United States and other countries may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Wireless connectivity and some features may require you to purchase additional software, services or external hardware.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

Intel, the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2016 Intel Corporation. All rights reserved.

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.

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.

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>.

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.

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

Intel, the Intel logo 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.

© 2016 Intel Corporation.

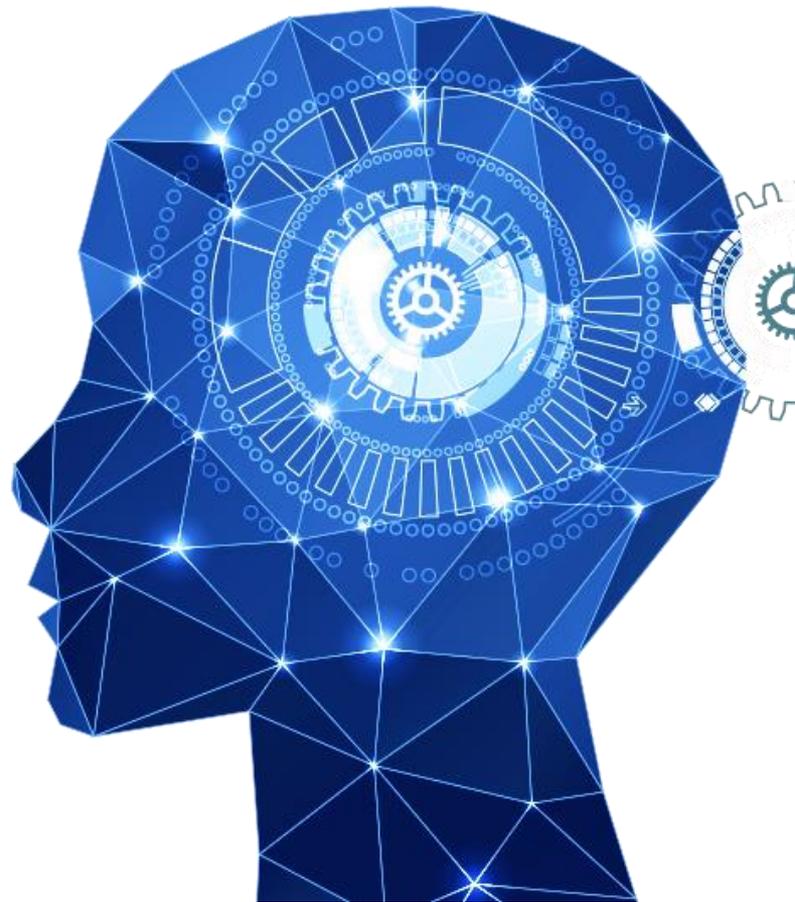
Optimization Notice

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.

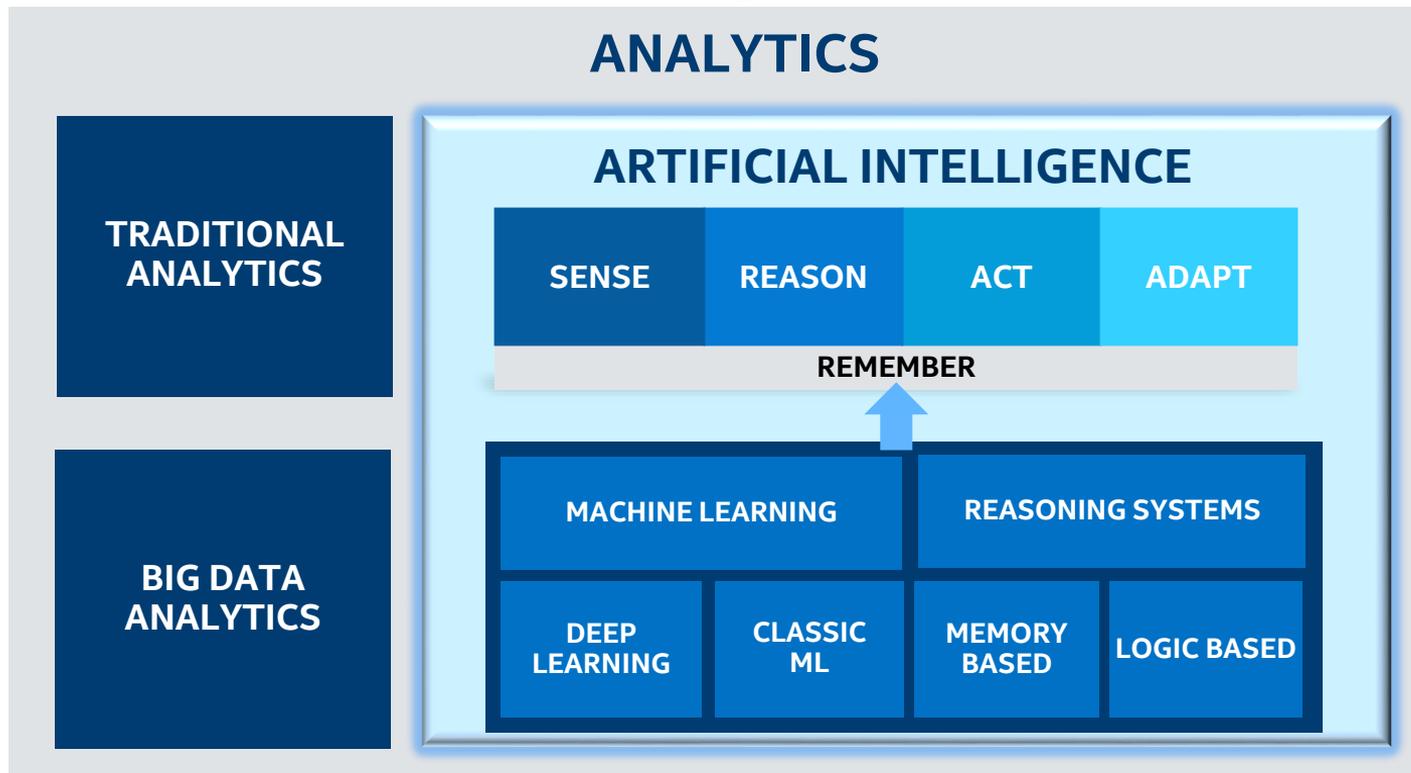
Notice revision #20110804

Agenda

- What is AI?
- Intel AI portfolio
- MKL-DNN
- Reinforcement Learning on IA
 - Atari Games experiment on Xeon/Xeon Phi
 - Environment - Open AI Gym, A3C, PLGRID
 - First results



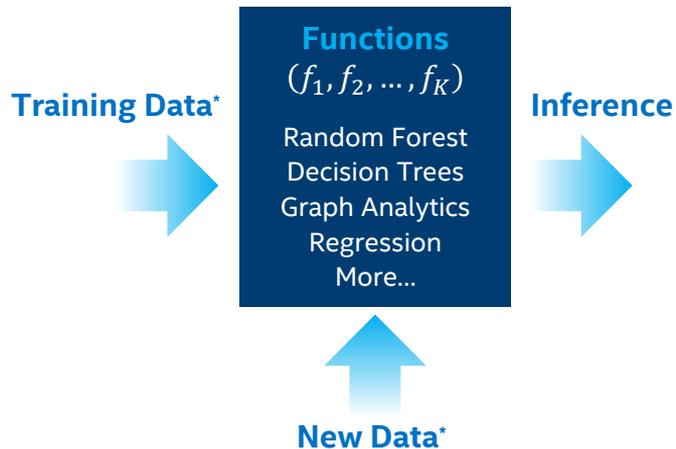
Focus on AI – Part of Analytics



What is Machine Learning?

CLASSIC ML

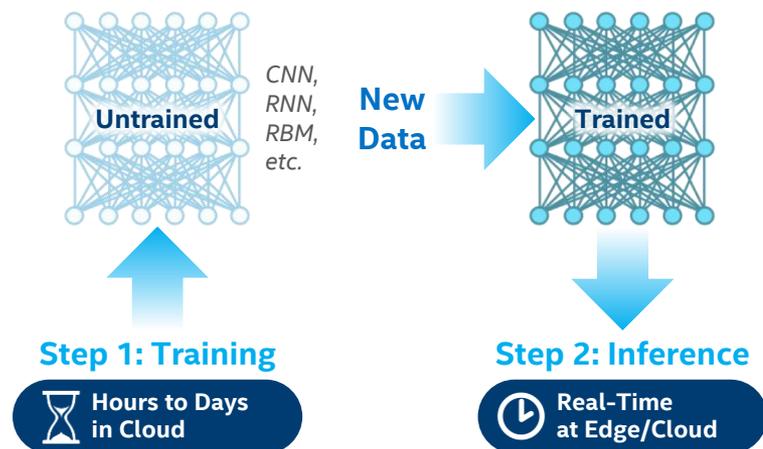
Using functions or algorithms to extract insights from new data



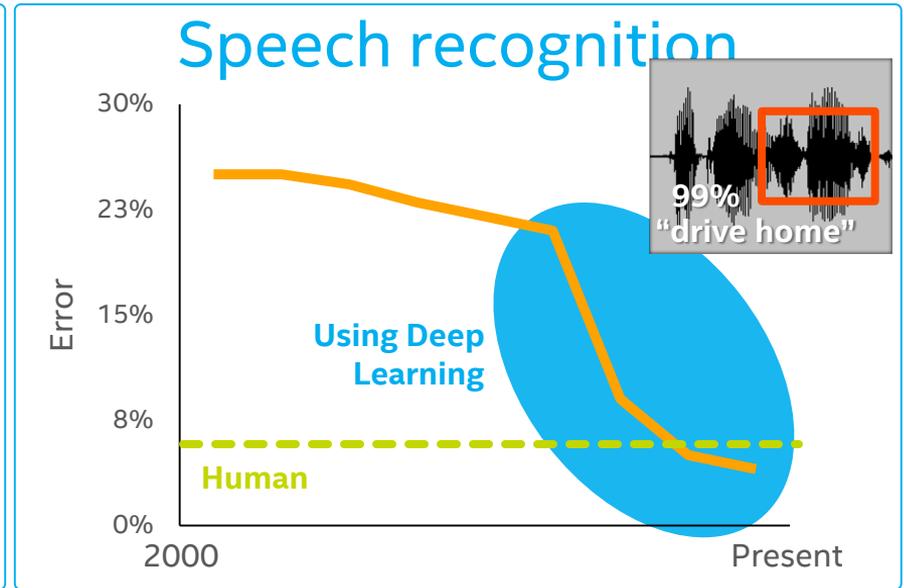
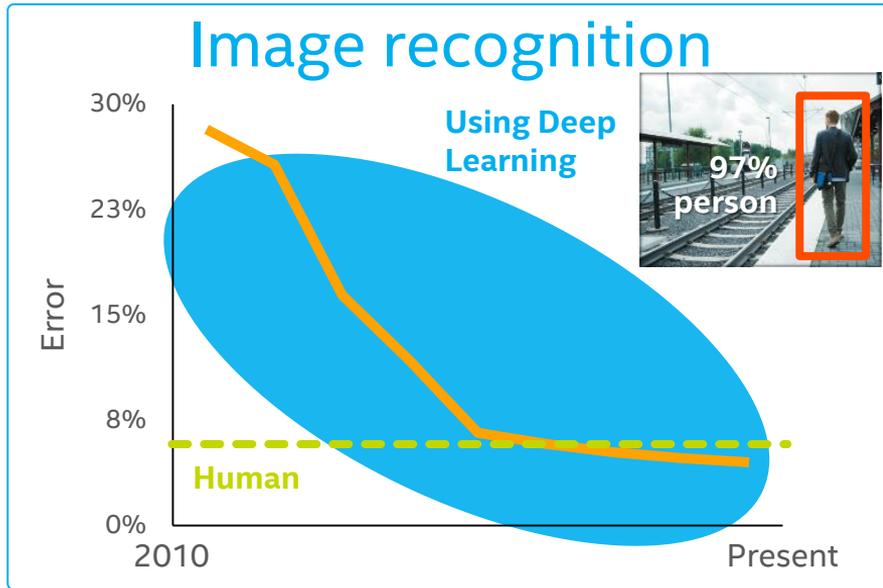
*Not all classical machine learning algorithms require separate training and new data sets

DEEP LEARNING

Using massive data sets to train deep (neural) graphs that can extract insights from new data



Deep Learning Breakthroughs



enabling improved and all new applications !

AI Will Usher in a Better World

on the scale of the agricultural, industrial and digital revolutions

ACCELERATE Large-Scale Solutions



Cure Diseases
Eliminate Fraud
Unlock Dark Data

UNLEASH Scientific Discovery



Explore Deep Sea/Space
Solve Particle Physics
Decode the Brain

Augment Human Capability



Personalized Guidance
Enhance Decisions
Prevent Crime

Automate Risky/Tedious Tasks



Automate Driving
Search & Rescue
No More Chores

Source: Intel

Intel Strategy: Intel® Nervana™ Portfolio

Machine Learning
Framework
Optimizations

Spark MLlib

Intel® Distribution for
Python

Deep Learning
Framework
Optimizations



Caffe theano

Low Level
Software
Primitives

Intel® DAAL

Intel® MKL

Nervana Graph

Intel® MKL-DNN

Intel® Silicon



+ Storage, Network

Xeon Phi: Scalable, Larger Memory Footprint & Great Performance

UP TO
400GB DIRECT
MEMORY
ACCESS

vs 16GB with a GPU¹

NEAR LINEAR SCALING
31X REDUCTION IN
TIME TO TRAIN

when scaling to 32 nodes²



AVAILABLE
2017

Knights Mill

Next-Gen Intel Xeon Phi

4X Deep learning
Performance
vs current gen³

Intel Xeon Phi Results

Nov'16 Top500 List

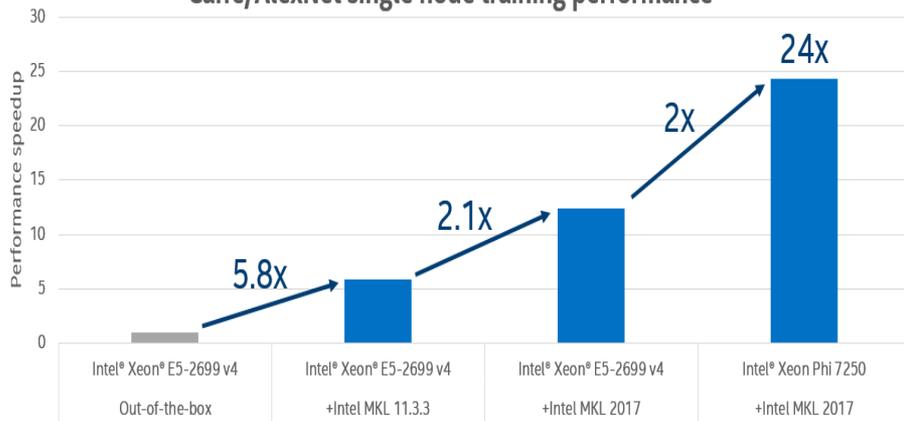
+45 → 80% NEW
PFLOPS System Accelerator Flops

Caffe + Intel® MKL 2017

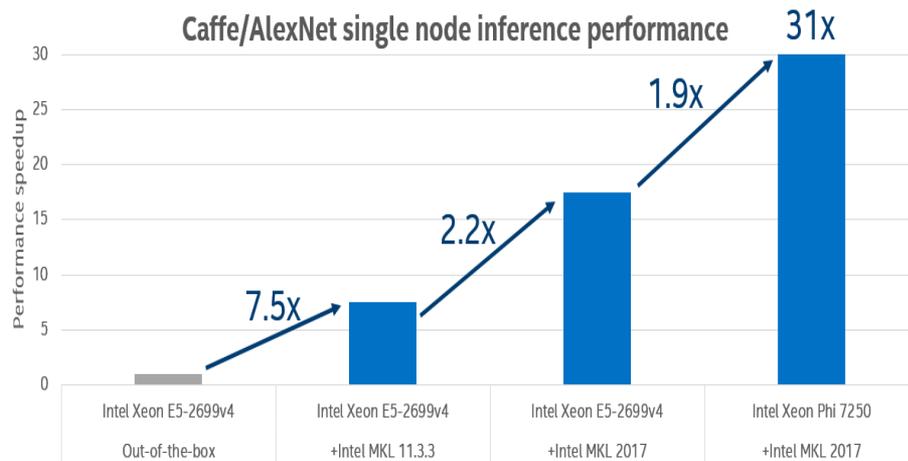
Intel Caffe <https://github.com/intelcaffe/caffe>

- The fork is aimed at improving Caffe performance on Intel® Xeon® CPUs.

Caffe/AlexNet single node training performance



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

- 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.

Reinforcement Learning on IA

Experiments on **Xeon/Xeon Phi**
Team: **deepsense.io, Intel**
Platform: **PLGRID Prometheus**

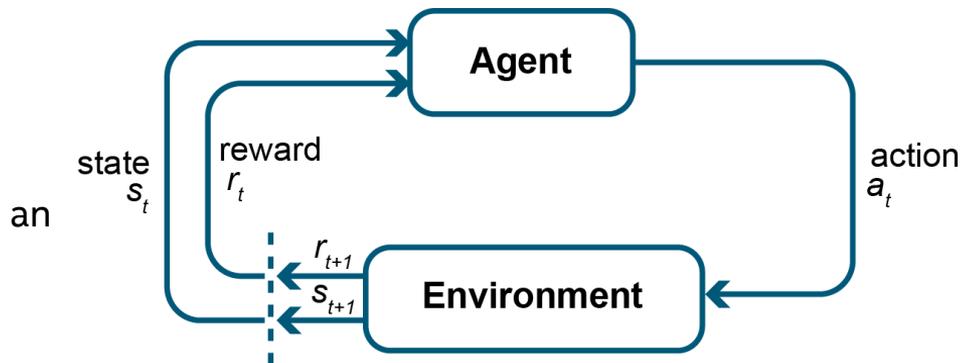
Reinforcement Learning

Agent learns from interaction with an **Environment**.

Very general problem

Examples:

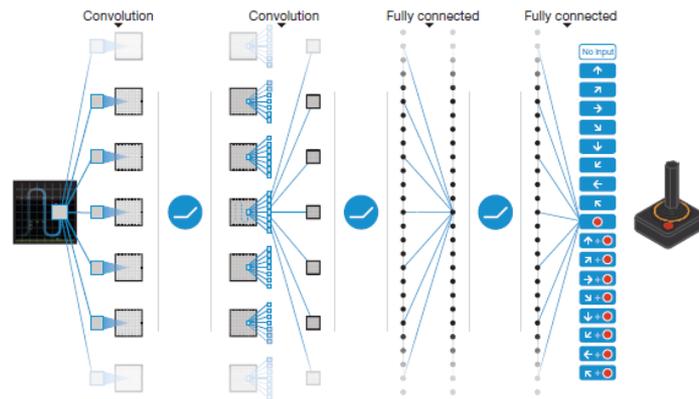
- Robot learning to move items in real environment - a reward is given when item is moved from A to B.
- Robot learning the same task in a simulator.
- An agent playing a board game like Chess - reward for winning a game.
- An agent playing a video game – rewards like in the actual game.



Reinforcement Learning

The task: Atari games on CPU

- Train agents for playing **Atari games** from pixel information
- The agent should maximize their score in the game
- Async A2C as an RL algorithm
- 4-layer ConvNet for processing input images



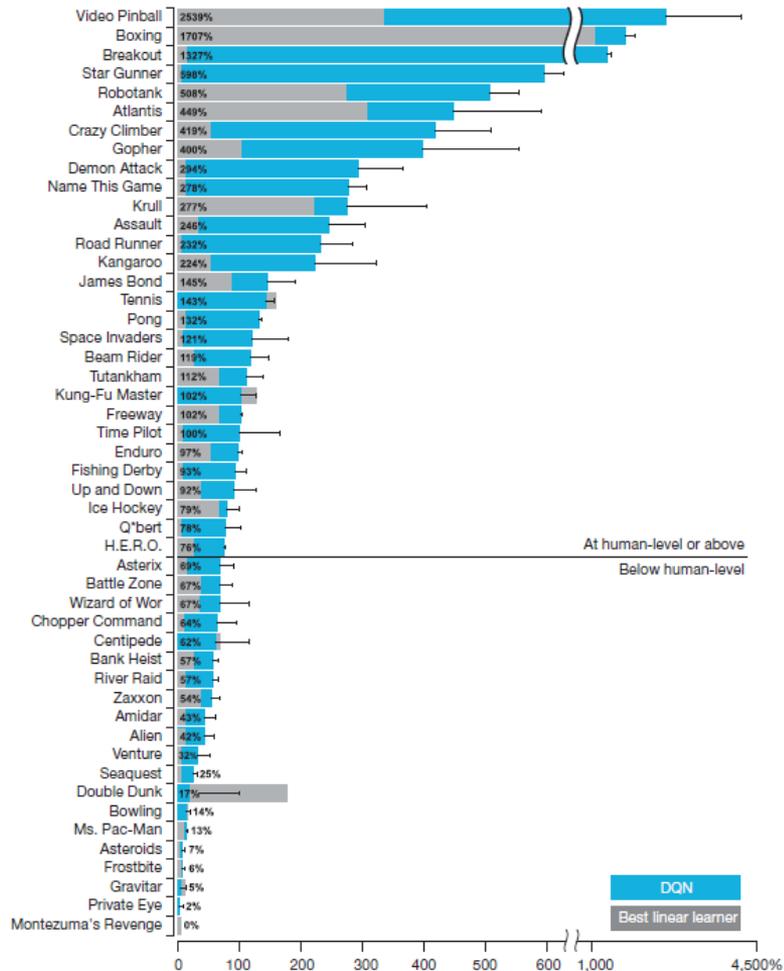
Reinforcement Learning

Benchmark games: Atari 2600 classics

Environment: Open AI Gym

Input: game screens, 210 x 160 pixels, 3 color channels

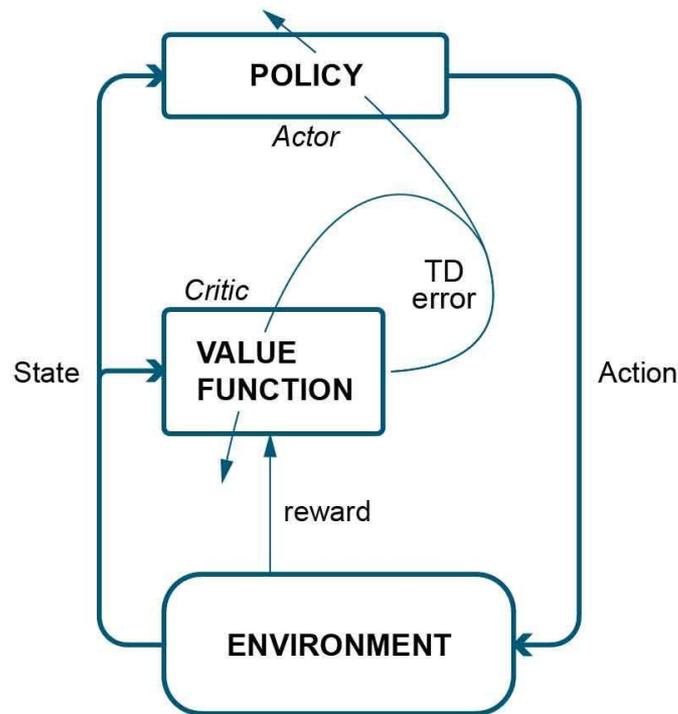
Output: one of 18 controller actions



Reinforcement Learning

Features of the Batch Asynchronous Advantage Actor-Critic Algorithm (BA3C):

- Hundreds of game simulators are running **in parallel** on a single machine
- The simulators use a shared model to evaluate actions
- The model can batch predictions from multiple simulators to increase efficiency
- The games played by the simulators are also batched and used for training the model

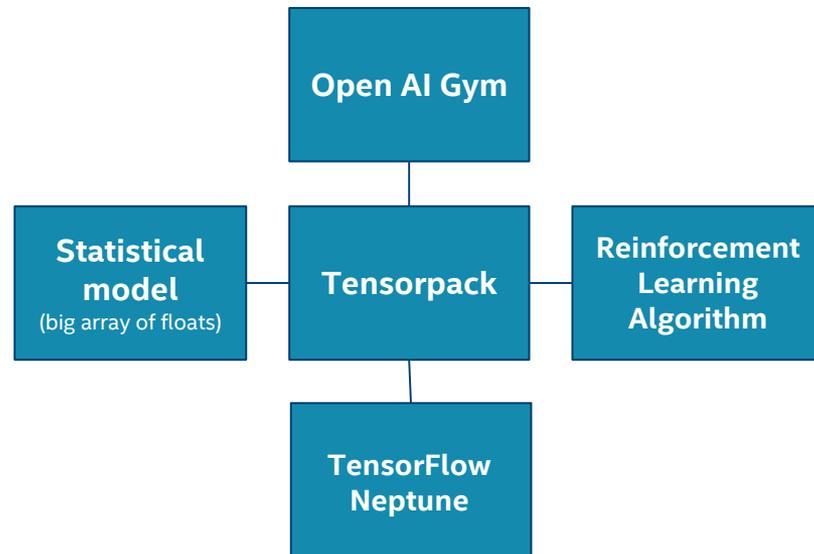


Source: Ben Lau, Using Keras...

Reinforcement Learning

Software and hardware stack:

- **Tensorpack** Framework implementing selected learning algorithms in TensorFlow ([Yuxin Wu](#)). Provides an efficient implementation of Async A2C algorithm
- **TensorFlow** General framework for machine learning
- **OpenAI Gym** Framework providing standard environments for reinforcement learning
- **Neptune** Tool for monitoring and managing experiments ([deepsense.io](#))
- **Prometheus** and Xeon Phi server (KNL)



DNN functions from Math Kernel Library

- We discovered that some TF convolutions were significantly slowing down the training.
- We used [MKL](#) (version 2017.0.098) for better performance
- We forked TensorFlow and provided alternative implementation of convolution using MKL primitives

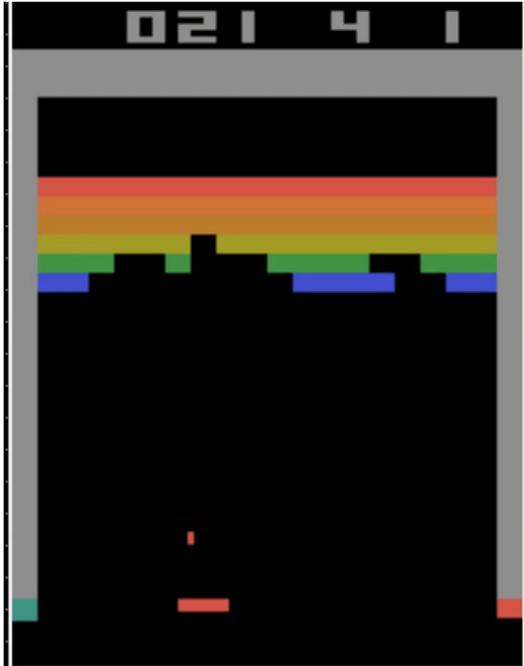
MKL convolution - backpropagation

Input shape	Kernel shape	Default TF time	MKL TF time	Default TF time	MKL TF time
		(Xeon) [ms]	(Xeon) [ms]	(Xeon Phi) [ms]	(Xeon Phi) [ms]
128x84x84x16,	5x5x16x32	368.18	29.63	1,236.98	8.97
128x40x40x32,	5x5x32x32	114.72	19.55	343.73	6.33
128x18x18x32	5x5x32x64	28.82	6.07	36.74	2.52
128x7x7x64	3x3x64x64	5.57	3.18	7.38	2.31

Results

Reinforcement Learning

Top performance in Breakout



Top performance in River Raid



Reinforcement Learning

Monitoring the learning process using the Neptune tool (Breakout on Xeon)



Breakout

Dashboard

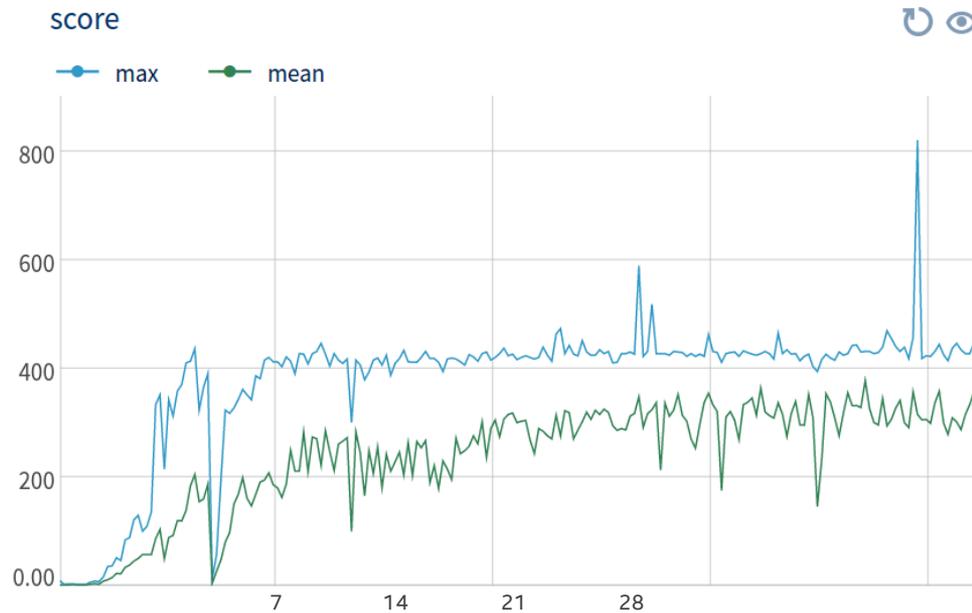
Channels

Charts

Parameters

Actions

TensorFlow



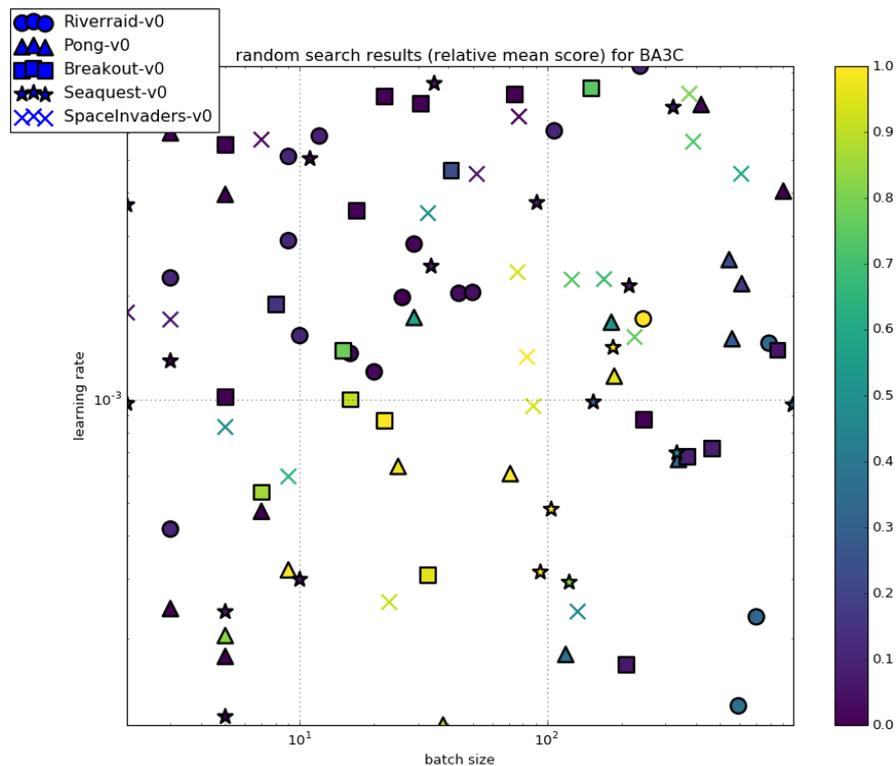
Reinforcement Learning

Monitoring the learning process using the Neptune tool (RiverRaid on Xeon)

Riverraid
Running for 4 days abort



Experiments on PLGRID Prometheus - Results



Reinforcement Learning

Summary

- RL agents trained on **CPU** in just a few hours
- 10x performance gain with MKL DNN implementation, 2.5x for convolutions only
- The performance vary drastically depending on the batch size and learning rate

Challenges and future work:

- Multinode impementation of code on optimized TensorFlow

