ACCELERATED COMPUTING: THE PATH FORWARD

Gunter Roeth, Senior Solution Architect

Agenda

9h30 -10h00 Introduction

10h00 – 10h30 ONTAP

10h30 -12h00 GPU Programming Guide

12h00 12h10 Coffee Break

12h10 – 14h10 Hands-On

14h10-14h40 Lunch

14h40 -15h40 Deep Learning SDK (cuDNN, TensorRT, DL Frameworks)

15h40- 15h50 Coffee Break

15h50-17h50 Image Classification with DIGITS (Hands-On)

17h50-18h00 Wrap up and Q&A

NAVIGATING TO COURSES

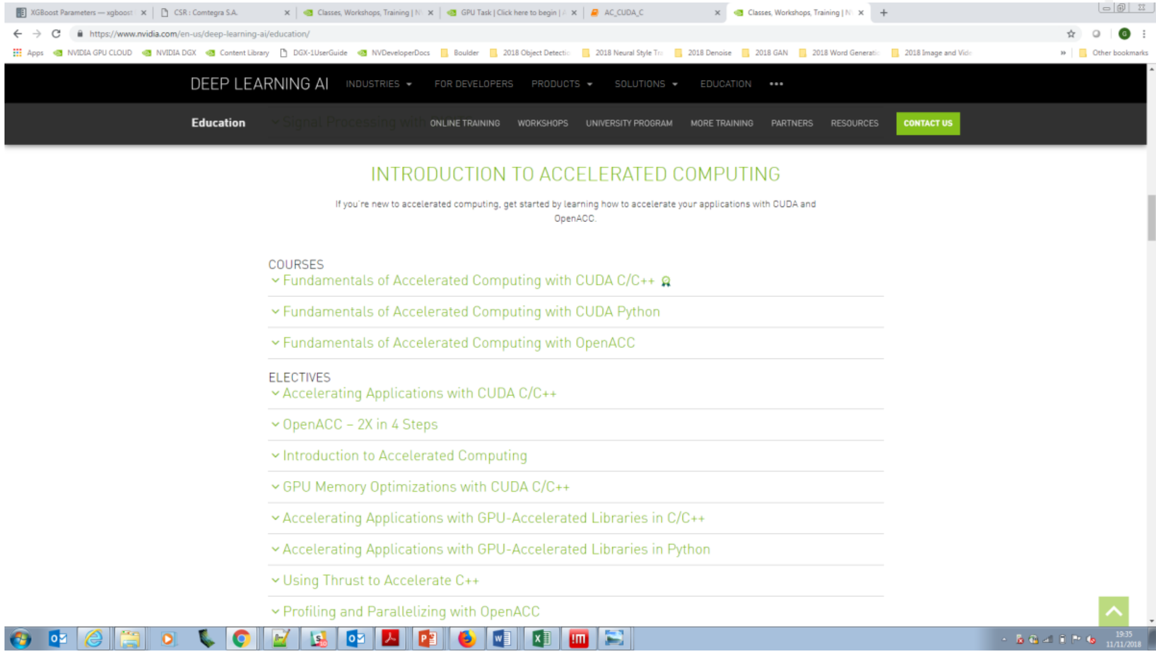
1. Navigate to:
   www.nvidia.co.uk/dlilabs

2. Google search for
   nvidia dli

3. Scroll down
   Training Online ELECTIVES

Use NV Developer login or new account.

Accelerating Applications with CUDA C/C++



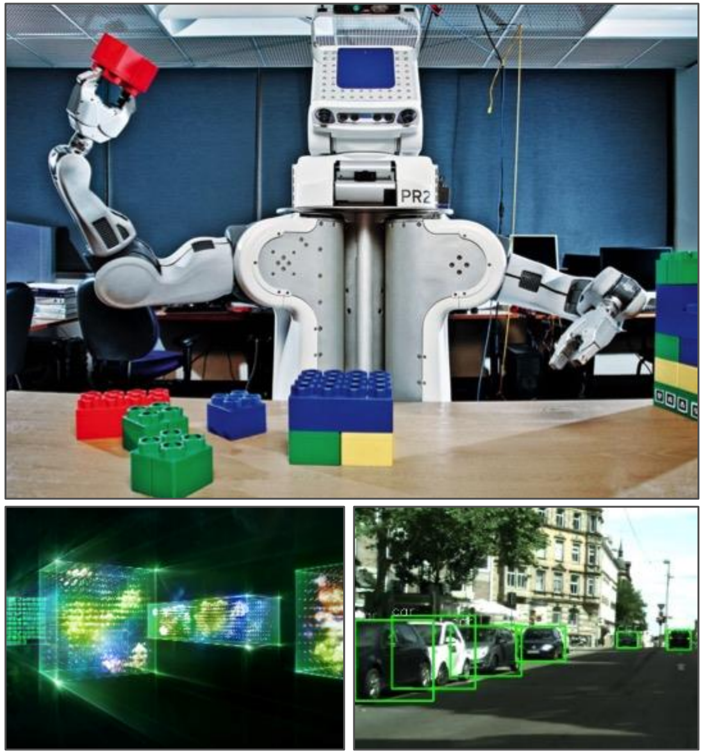
INTRODUCTION

NVIDIA

- Founded in 1993
- HQ in Santa Clara
- Jensen Huang, Founder & CEO
- 11,000 employees
- $9.7B in FY18

Computer Graphics

GPU Computing

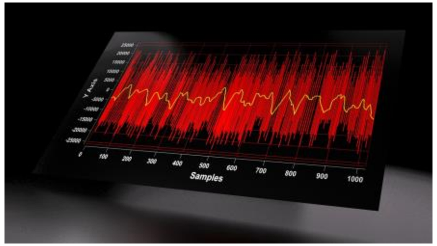


Artificial Intelligence

ACCELERATED COMPUTING

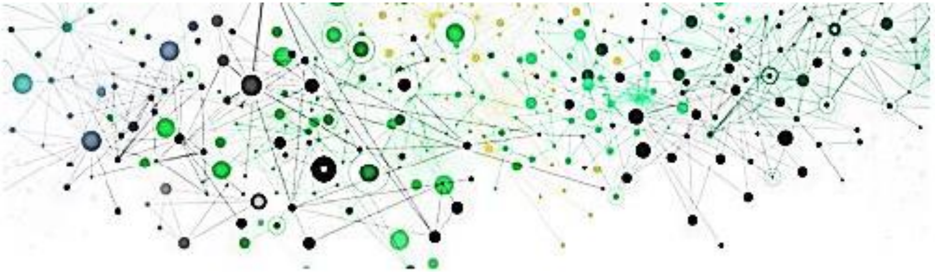
Performance & Energy Efficiency

HIGH PERFORMANCE COMPUTE



AI / DEEP LEARNING

DATA ANALYTICS



ACCELERATED VDI

NVIDIA TESLA PLATFORM

World’s Leading Data Center Platform for Accelerating HPC and AI

CUSTOMER USECASES

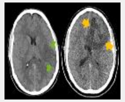
CONSUMER INTERNET


Speech

Translate


Recommender

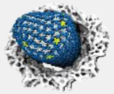
ENTERPRISE APPLICATIONS

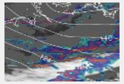

Healthcare

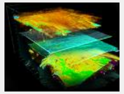

Manufacturing


Engineering

SUPERCOMPUTING


Molecular Simulations


Weather Forecasting


Seismic Mapping

INDUSTRY FRAMEWORKS & APPLICATIONS

Caffe2 · Chainer · Microsoft Cognitive Toolkit · mxnet · PaddlePaddle · PYTORCH · TensorFlow

Amber · ANSYS · CHROMA · GROMACS FAST. FLEXIBLE. FREE. · LAMMPS · NAMD · SIMULIA · VASP

+550 Applications

NVIDIA SDK & LIBRARIES

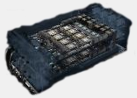
cuBLAS · cuDNN · cuFFT · cuRAND · cuSPARSE · DeepStream · NCCL · TensorRT · PGI · OpenACC Directives for Accelerators

CUDA

TESLA GPUs & SYSTEMS


TESLA GPU


NVIDIA DGX FAMILY


NVIDIA HGX

DELL · Hewlett Packard Enterprise · IBM

SYSTEM OEM

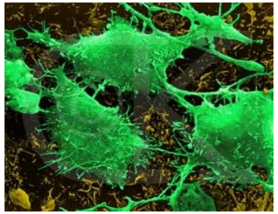
aws · Google Cloud Platform · Microsoft Azure

CLOUD

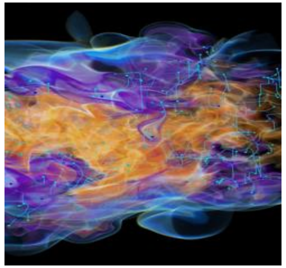
CONTINUED DEMAND FOR COMPUTE POWER

Ever-increasing compute power Demand in HPC

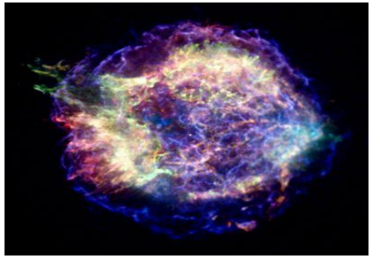
Comprehensive Earth System Model



Coupled simulation of entire cells



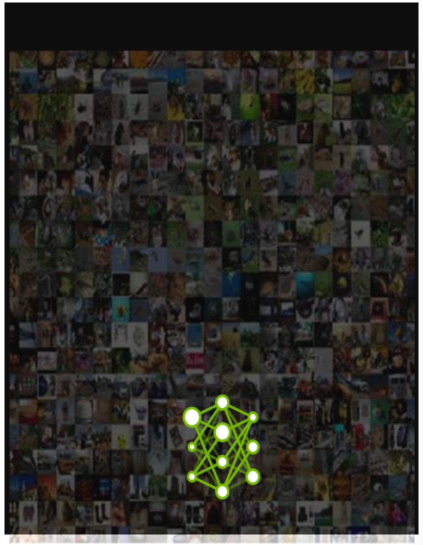
Simulation of combustion for new high-efficiency, low-emision engines.



Predictive calculations for supernovae

Neural Network complexity is Exploding

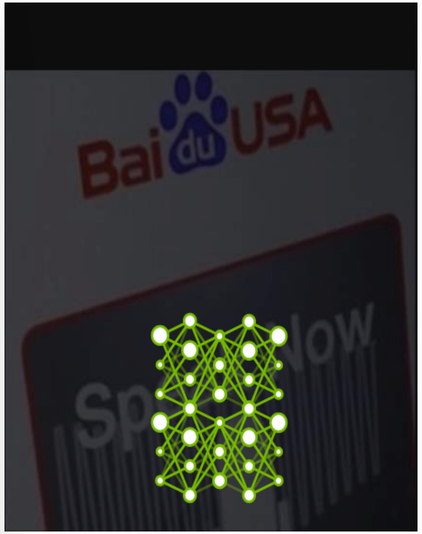
7 ExaFLOPS
60 Million Parameters



2015

Microsoft ResNet
Superhuman Image Recognition

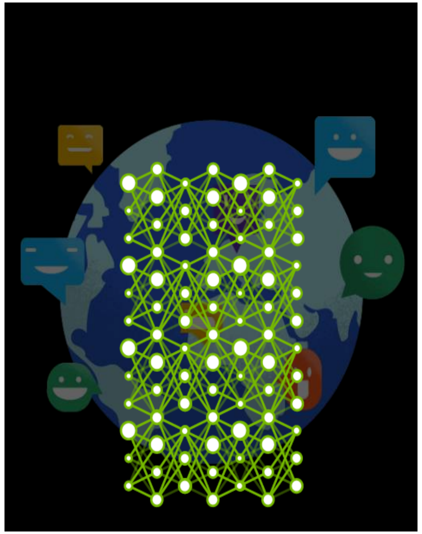
20 ExaFLOPS
300 Million Parameters



2016

Baidu Deep Speech 2
Superhuman Voice Recognition

100 ExaFLOPS
8700 Million Parameters



2017

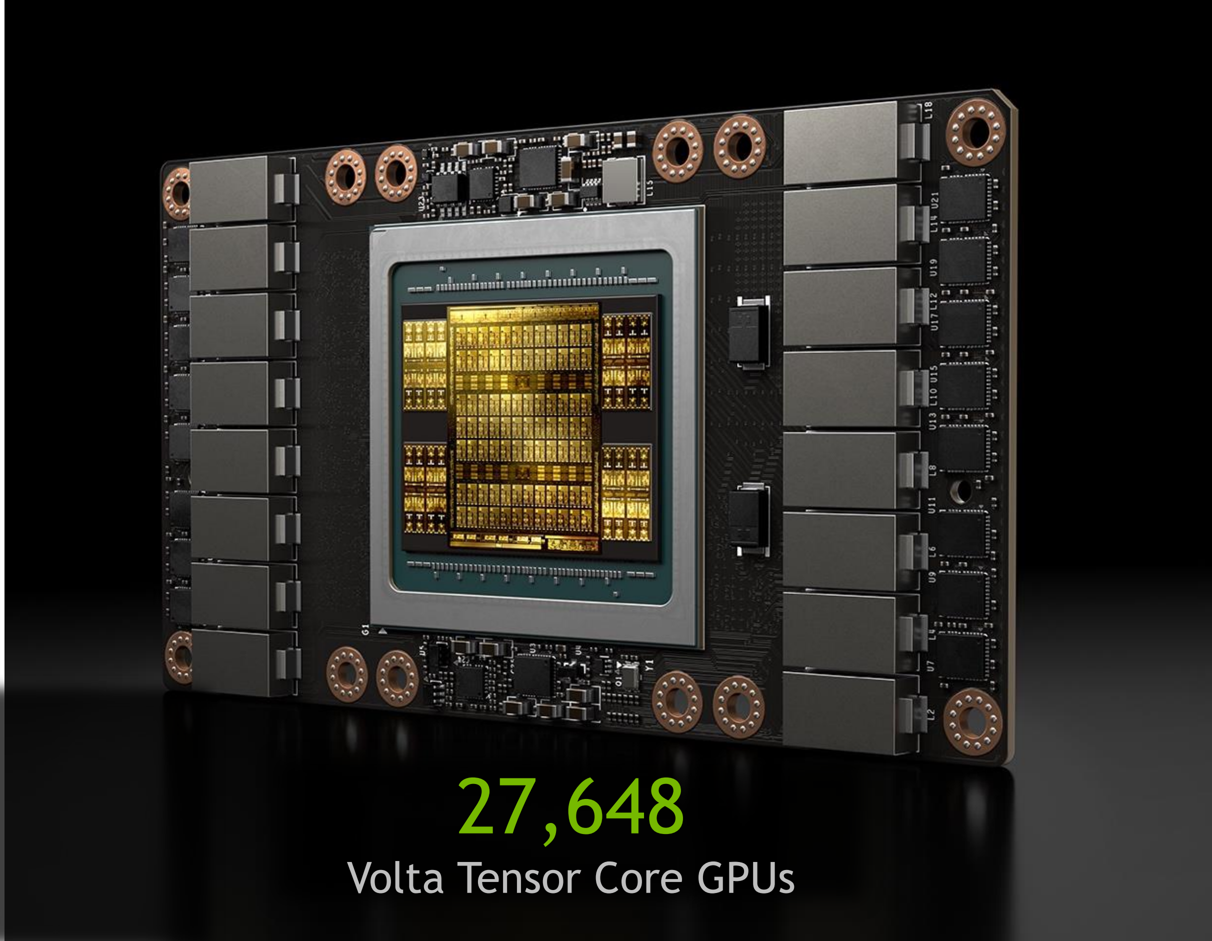
Google Neural Machine Translation
Near Human Language Translation

NVIDIA POWERS WORLD'S FASTEST SUPERCOMPUTER

Summit Becomes First System To Scale The 100 Petaflops Milestone

122 PF
HPC

3 EF
AI

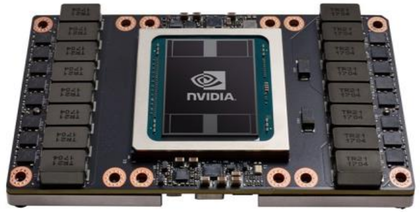


27,648
Volta Tensor Core GPUs

GPUS FOR HPC AND DEEP LEARNING

Huge demand on compute power (FLOPS)

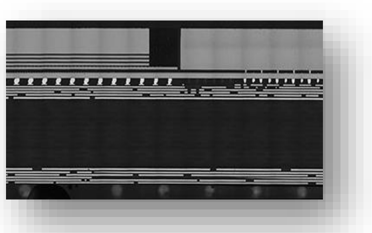
NVIDIA Tesla V100



5120 energy efficient cores + TensorCores
7.8 TF Double Precision (fp64), 15.6 TF Single Precision (fp32) ,
125 Tensor TFLOP/s mixed-precision

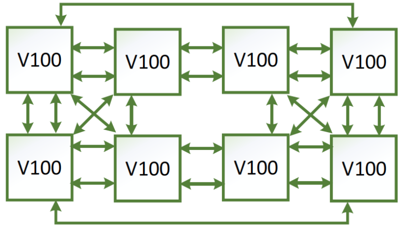
Huge demand on communication and memory bandwidth

CoWoS with HBM2



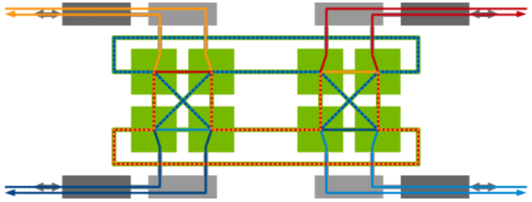
900 GB/s Memory Bandwidth
Unifying Compute & Memory
in Single Package

NVLink



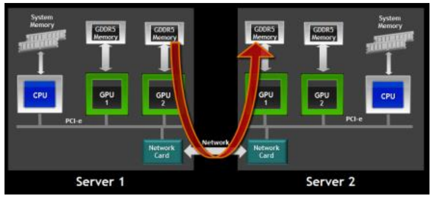
6 links per GPU a 50 GB/s bi-directional for maximum scalability between GPU’s

NCCL



High-performance multi-GPU and multi-node collective communication primitives optimized for NVIDIA GPUs

GPU Direct /
GPU Direct RDMA



Direct communication between GPUs by eliminating the CPU from the critical path

NEW VOLTA SM MICROARCHITECTURE

TESLA V100

21B transistors
815 mm²

80 SM
5120 CUDA Cores
640 Tensor Cores

16/32 GB HBM2
900 GB/s HBM2
300 GB/s NVLink

*full GV100 chip contains 84 SMs

VOLTA GV100 SM

Redesigned for Productivity

Completely new ISA
Twice the schedulers
Simplified Issue Logic
Large, fast L1 cache
Improved SIMT model
Tensor acceleration

	GP100	GV100
FP32 units	64	64
FP64 units	32	32
INT32 units	NA	64
Tensor Cores	NA	8
Register File	256 KB	256 KB
Unified L1/Shared memory	L1: 24KB Shared: 64KB	128 KB
Active Threads	2048	2048

VOLTA TENSOR CORE

TENSOR CORE

Mixed Precision Matrix Math - 4x4 matrices

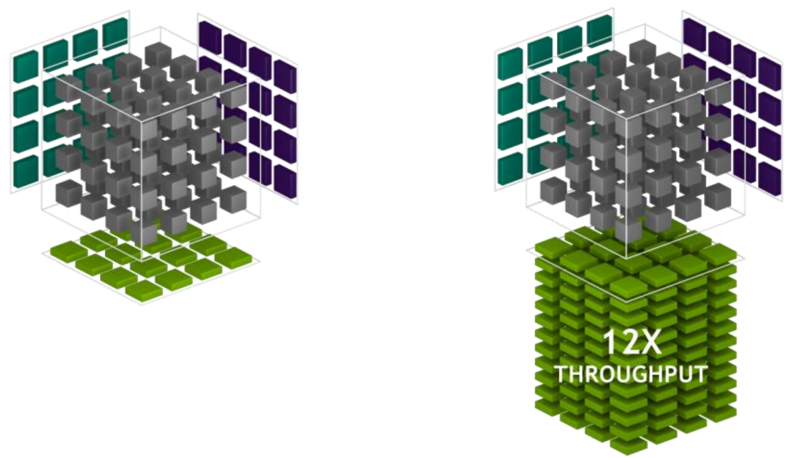
New CUDA TensorOp instructions & data formats

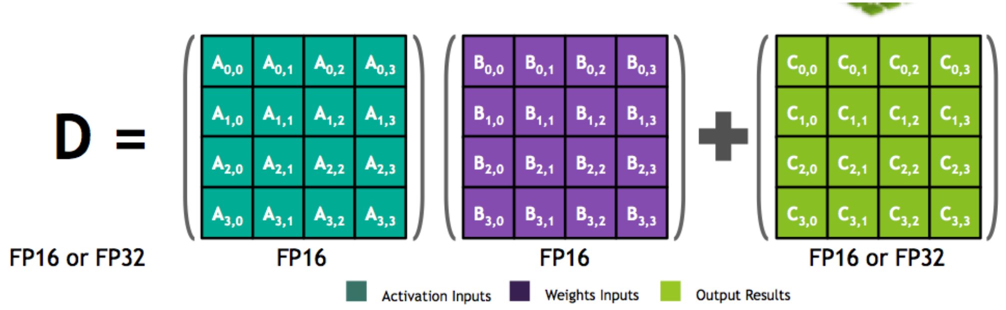
4x4x4 matrix processing array

D[FP32] = A[FP16] * B[FP16] + C[FP32]

Using Tensor cores via

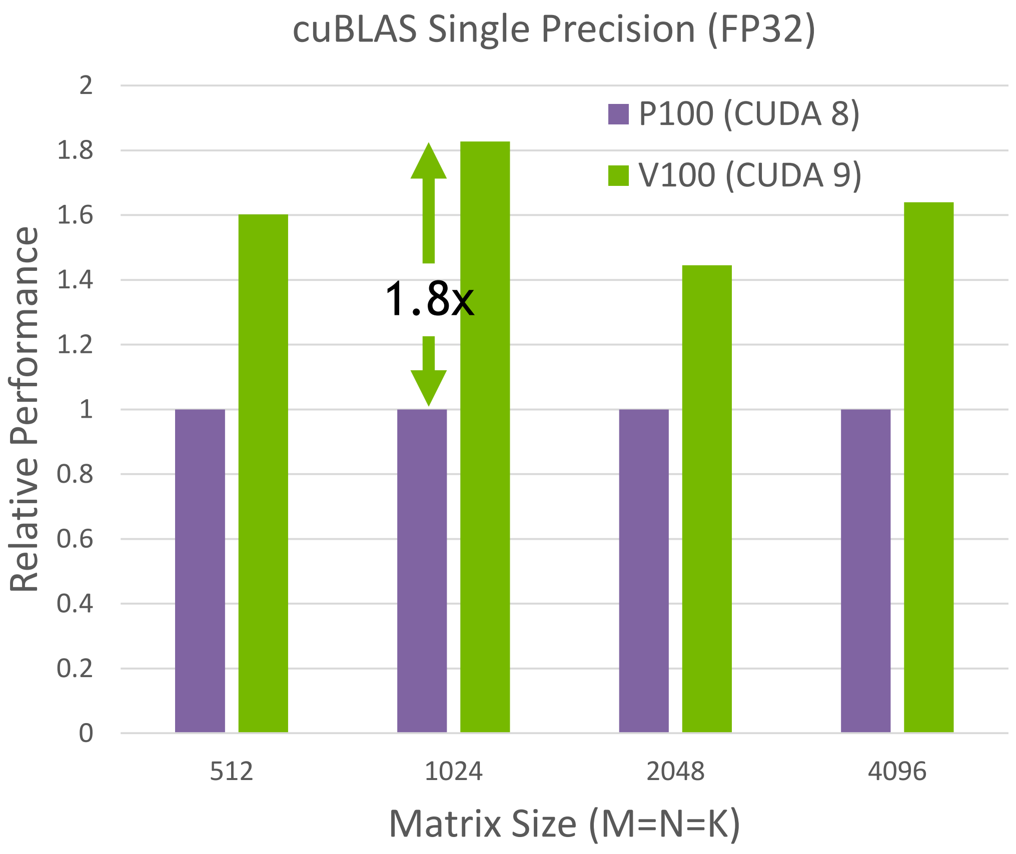
- Volta optimized frameworks and libraries (cuDNN, CuBLAS, TensorRT, ..)
- CUDA C++ Warp Level Matrix Operations

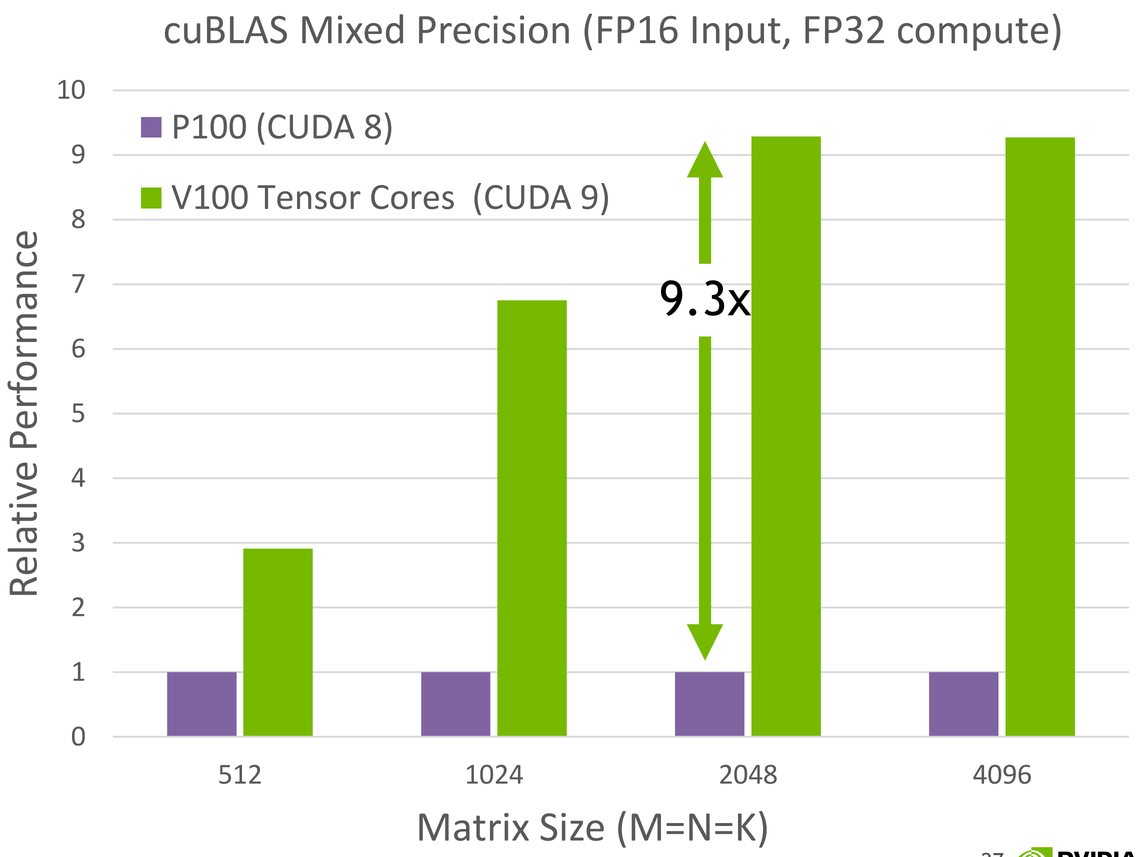




cuBLAS GEMMS FOR DEEP LEARNING

V100 Tensor Cores + CUDA 9: over 9x Faster Matrix-Matrix Multiply





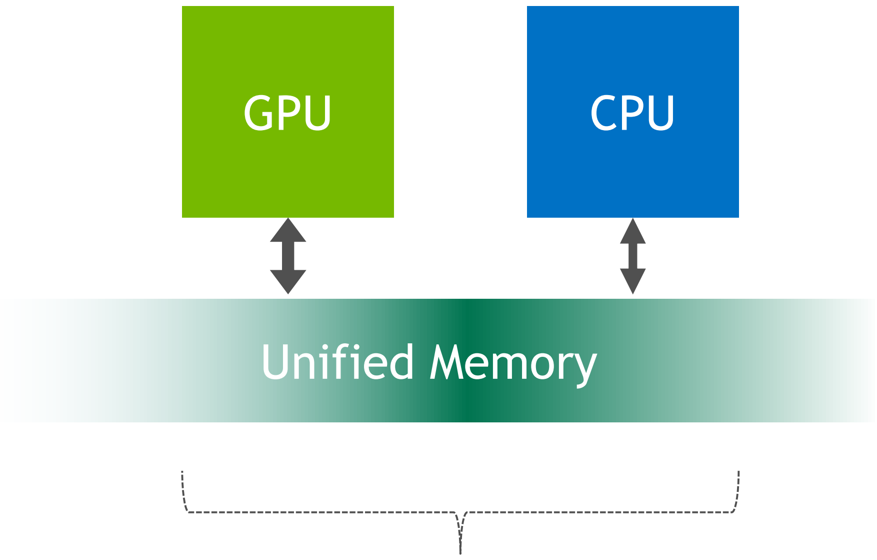
Note: pre-production Tesla V100 and pre-release CUDA 9. CUDA 8 GA release.

MEMORY

UNIFIED MEMORY

Large datasets, simple programming, High Performance

CUDA 8 and beyond



Allocate Beyond
GPU Memory Size

Enable Large Data Models

- Oversubscribe GPU memory
- Allocate up to system memory size

Tune Unified Memory Performance

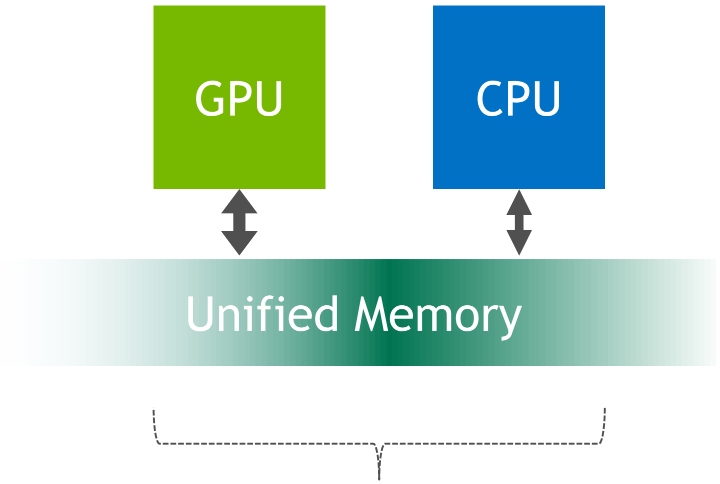
- Usage hints via cudaMemAdvise API
- Explicit prefetching API

Simpler Data Access

- CPU/GPU Data coherence
- Unified memory atomic operations

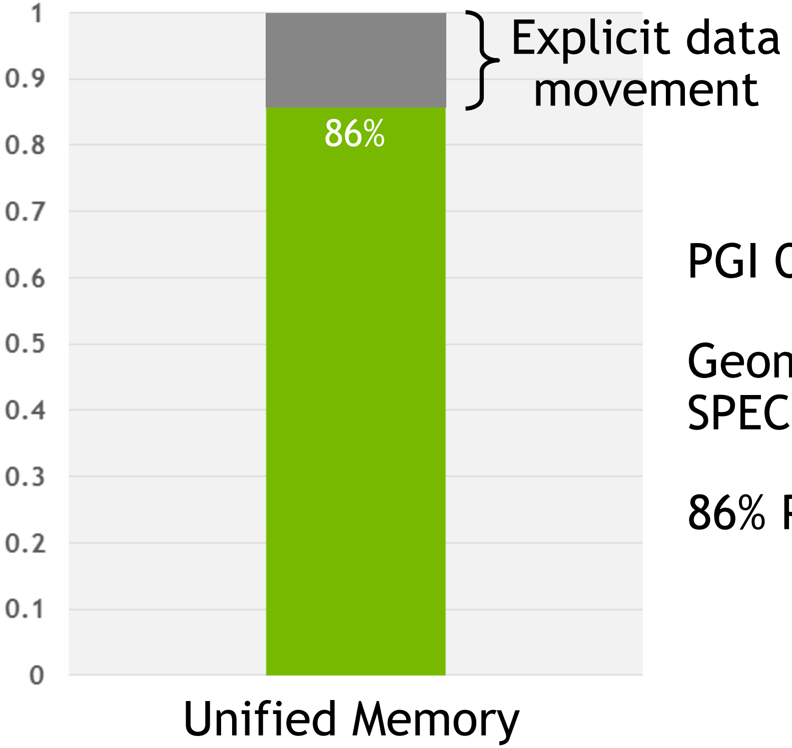
STATE OF UNIFIED MEMORY

High performance, low effort



Allocate Beyond
GPU Memory Size

Performance vs no Unified Memory



Automatic data movement for allocatables

PGI OpenACC on Pascal P100

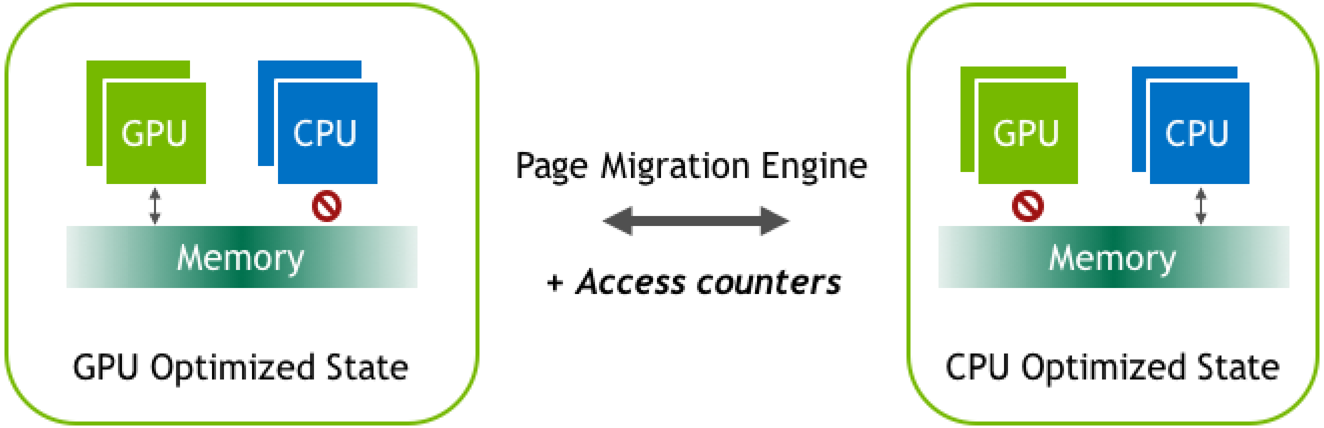
Geometric mean across all 15 SPEC ACCEL™ benchmarks

86% PCI-E, 91% NVLink

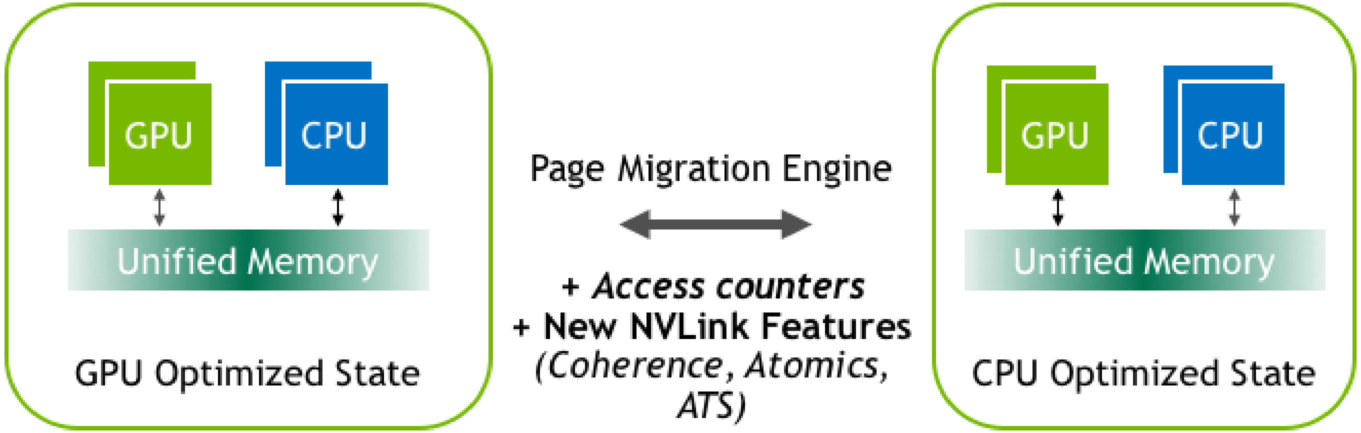
PGI 17.1 Compilers OpenACC SPEC ACCEL™ 1.1 performance measured March, 2017. SPEC® and the benchmark name SPEC ACCEL™ are registered trademarks of the Standard Performance Evaluation Corporation.

VOLTA + UNIFIED MEMORY

VOLTA + PCIE CPU



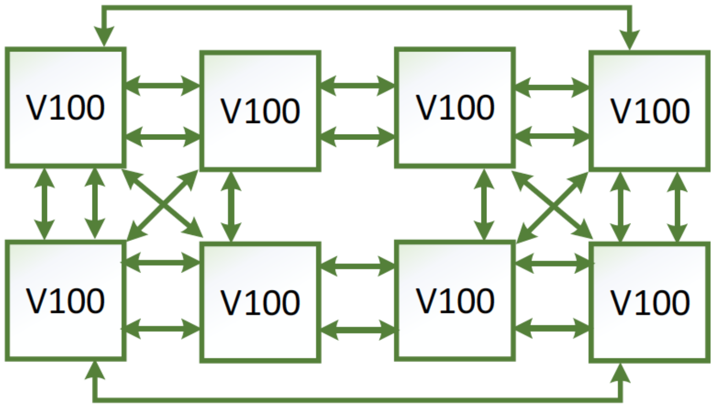
VOLTA + NVLINK CPU



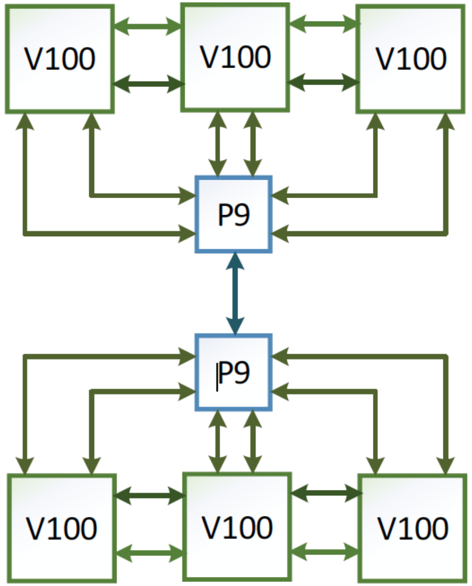
NVLINK

VOLTA NVLINK

- 6 NVLINKS @ 50 GB/s bidirectional
- Reduce number of lanes for lightly loaded link (Power savings)
- Coherence features for NVLINK enabled CPUs



Hybrid cube mesh
(eg. DGX1V)



POWER9 based node

TESLA GPUS

V100 WITH 16 OR 32GB HBM2

Maintain Form Factor Compatibility

Form Factor		
Performance	7.8T F DP, 15.7 TF SP, 125TF TensorCore	7.0 TF DP, 14.0 TF SP, 112 TF TensorCore
Memory Size	16 or 32GB HBM2	16 or 32GB HBM2
Memory Bandwidth	900GB/s	900GB/s
GPU Peer to Peer	NVLink	PCIe Gen3
Power	300W	250W
Available From All Major OEMs	CISCO CRAY DELLEMC Hewlett Packard Enterprise IBM inspur lenovo SUPERMICRO TYAN	

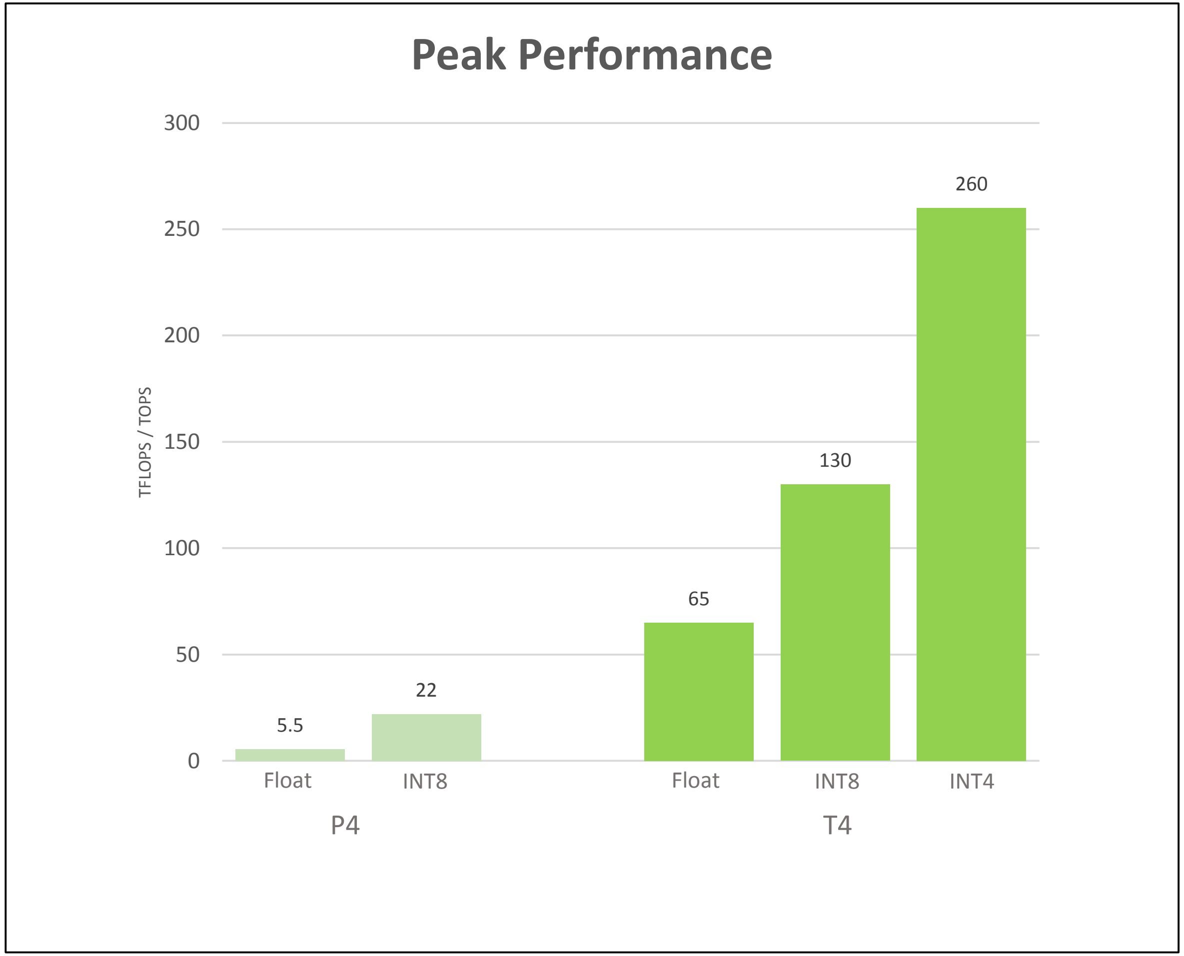
SXM2 32GB P/N = 900-2G503-0010-000, PCIE 32GB P/N = 900-2G500-0010-000

TESLA T4

2,560 CUDA cores + 320 Tensor Cores
8.1 TFLOPS FP32 | 65 FP16 TFLOPS
130 INT8 TOPS | 260 INT4 TOPS

16GB GDDR6 Memory | 320GB/s

75 W Low Profile PCI-e



TESLA PRODUCTS DECODER

	P100 (SXM2)	P100 (PCIE)	P40	P4	T4	V100 (PCIE)	V100 (SXM2)	V100 (FHHL)
GPU CHIP	GP100	GP100	GP102	GP104	TU104	GV100	GV100	GV100
PEAK FP64 (TFLOPs)	5.3	4.7	NA	NA	NA	7	7.8	6.5
PEAK FP32 (TFLOPs)	10.6	9.3	12	5.5	8.1	14	15.7	13
PEAK FP16 (TFLOPs)	21.2	18.7	NA	NA	65	112	125	105
PEAK TOPs	NA	NA	47	22	260	NA	NA	NA
Memory Size	16 GB HBM2	16/12 GB HBM2	24 GB GDDR5	8 GB GDDR5	16 GB HBM2	32 GB HBM2	32 GB HBM2	16GB HBM2
Memory BW	732 GB/s	732/549 GB/s	346 GB/s	192 GB/s	320GB/s	900 GB/s	900 GB/s	900 GB/s
Interconnect	NVLINK + PCIe Gen3	PCIe Gen3	PCIe Gen3	PCIe Gen3	PCIe Gen3	PCIe Gen3	NVLINK + PCIe Gen3	PCIe Gen3
ECC	Internal + HBM2	Internal + HBM2	GDDR5	GDDR5	GDDR6	Internal + HBM2	Internal + HBM2	Internal + HBM2
Form Factor	SXM2	PCIE Dual Slot	PCIE Dual Slot	PCIE LP	PCIE LP	PCIE Dual Slot	SXM2	PCIE Single Slot Full Height Half Length
Power	300 W	250 W	250 W	50-75 W	75 W	250W	300W	150W

DGX-STATION / DGX-1
DGX-2 / HGX-2

NVIDIA DGX-STATION

AI supercomputer for the desk

4x Tesla V100 connected via NVLINK

(60 TFLOPS FP32, 0.5 PFLOPS Tensor performance)

Xeon CPU, 256 GB Memory

Storage:

- 3X 1.92 TB SSD RAID 0 (Data)
- 1X 1.92 TB SSD (OS)

Dual 10GbE

1500W, Water-cooled → Quiet

Optimized Deep Learning Software across the entire stack

- Containerized frameworks
- Always up-to-date via the cloud

NVIDIA DGX-1

AI supercomputer-appliance-in-a-box

8x Tesla V100 connected via NVLINK

(125 TFLOPS FP32, 1 PFLOPS Tensor Core performance)

Dual Xeon CPU, 512 GB Memory

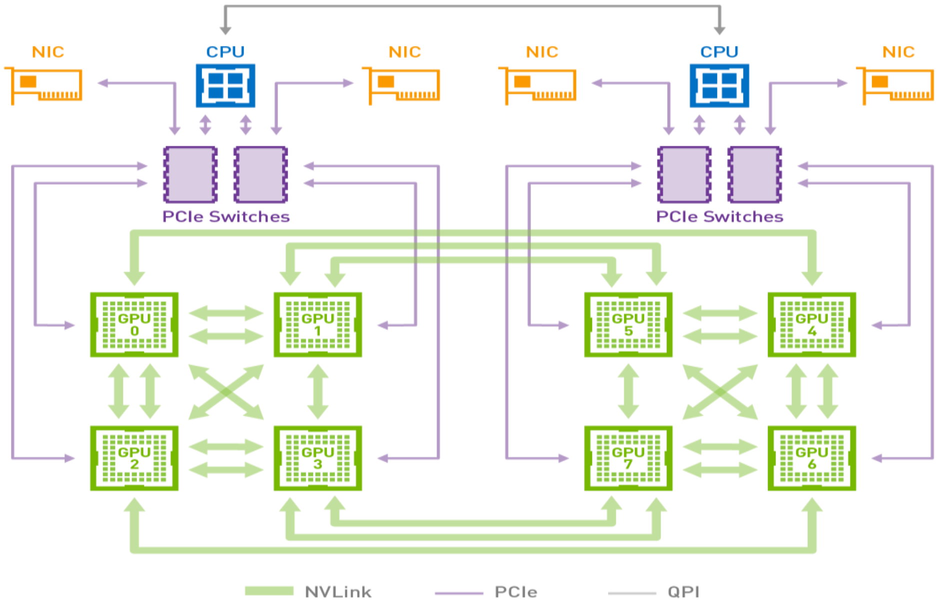
7 TB SSD Deep Learning Cache

Dual 10GbE, Quad IB 100Gb

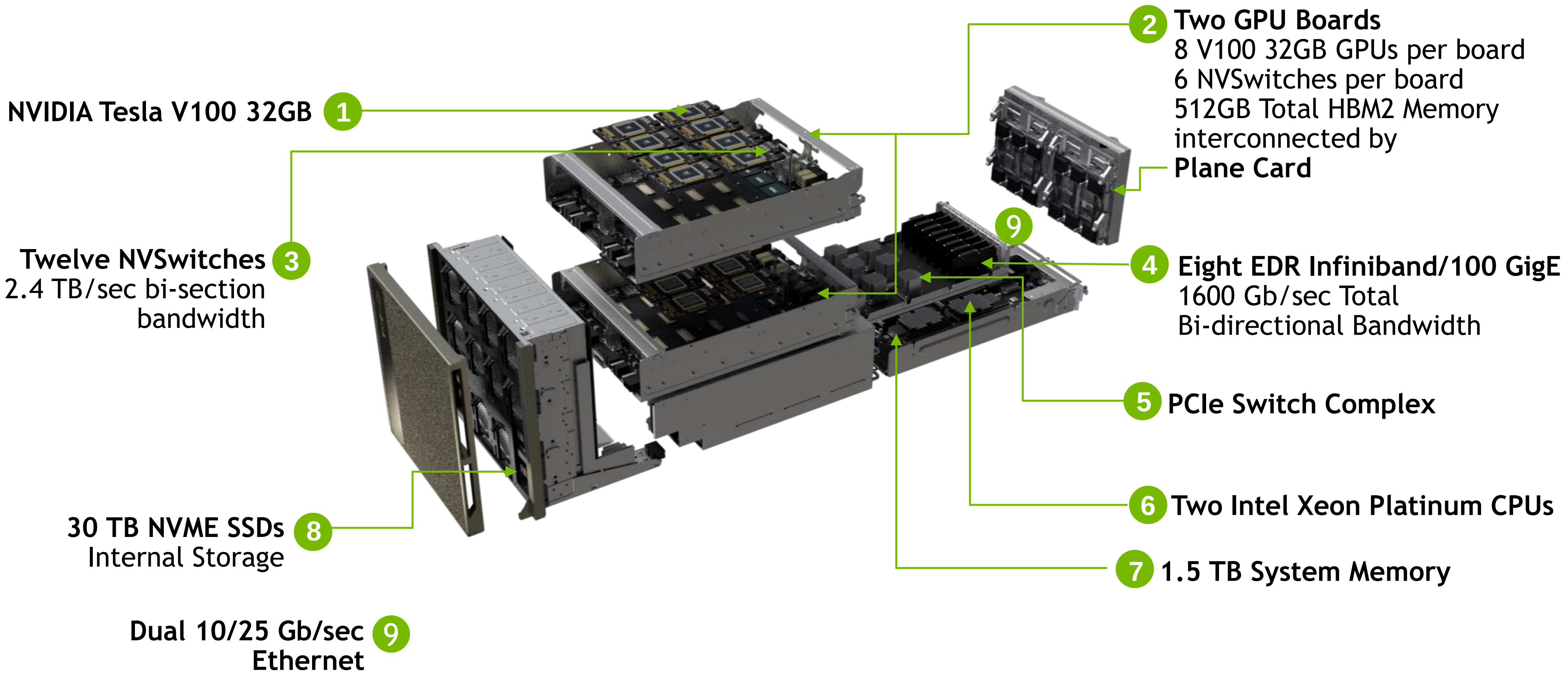
3RU – 3200W

Optimized Deep Learning Software across the entire stack

- Containerized frameworks
- Always up-to-date via the cloud



NVIDIA DGX-2

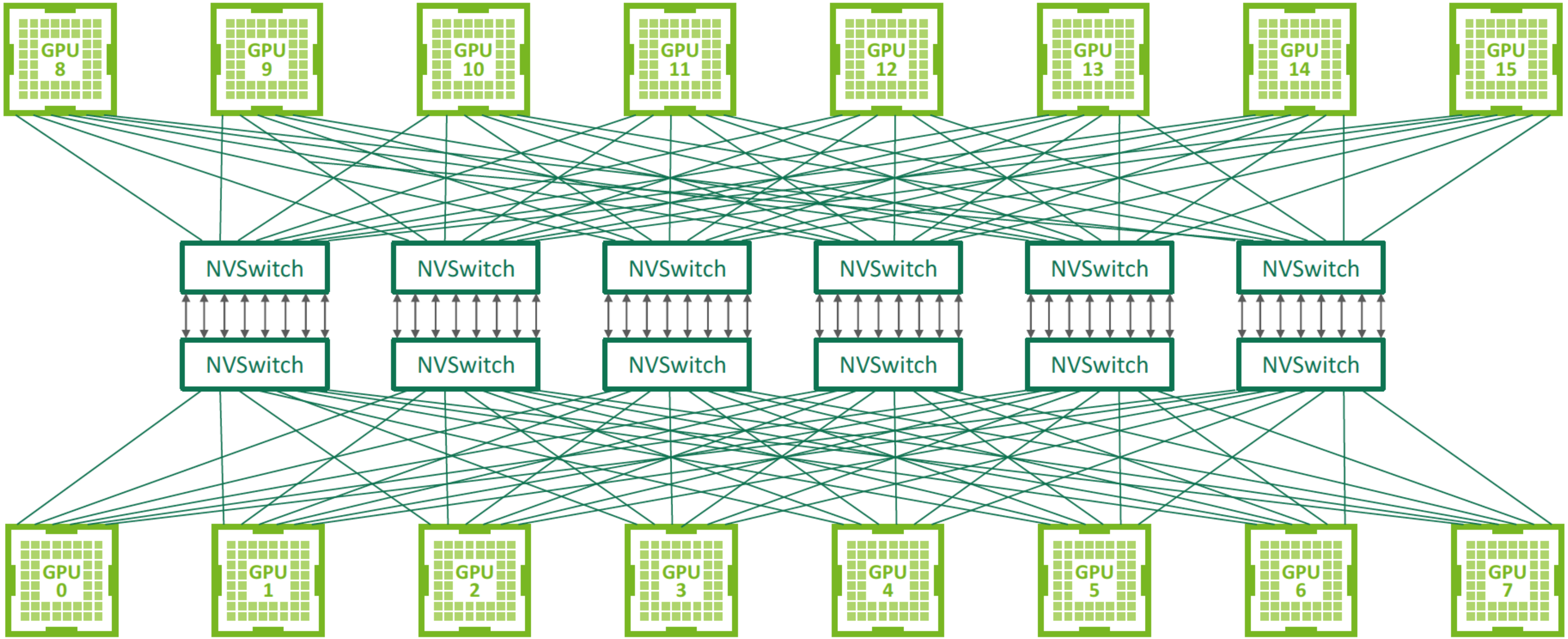


NVIDIA Tesla V100 32GB 1
2 Two GPU Boards
8 V100 32GB GPUs per board
6 NVSwitches per board
512GB Total HBM2 Memory
interconnected by
Plane Card
Twelve NVSwitches 3
2.4 TB/sec bi-section
bandwidth
9
4 Eight EDR Infiniband/100 GigE
1600 Gb/sec Total
Bi-directional Bandwidth
5 PCIe Switch Complex
6 Two Intel Xeon Platinum CPUs
7 1.5 TB System Memory
30 TB NVME SSDs 8
Internal Storage
Dual 10/25 Gb/sec 9
Ethernet

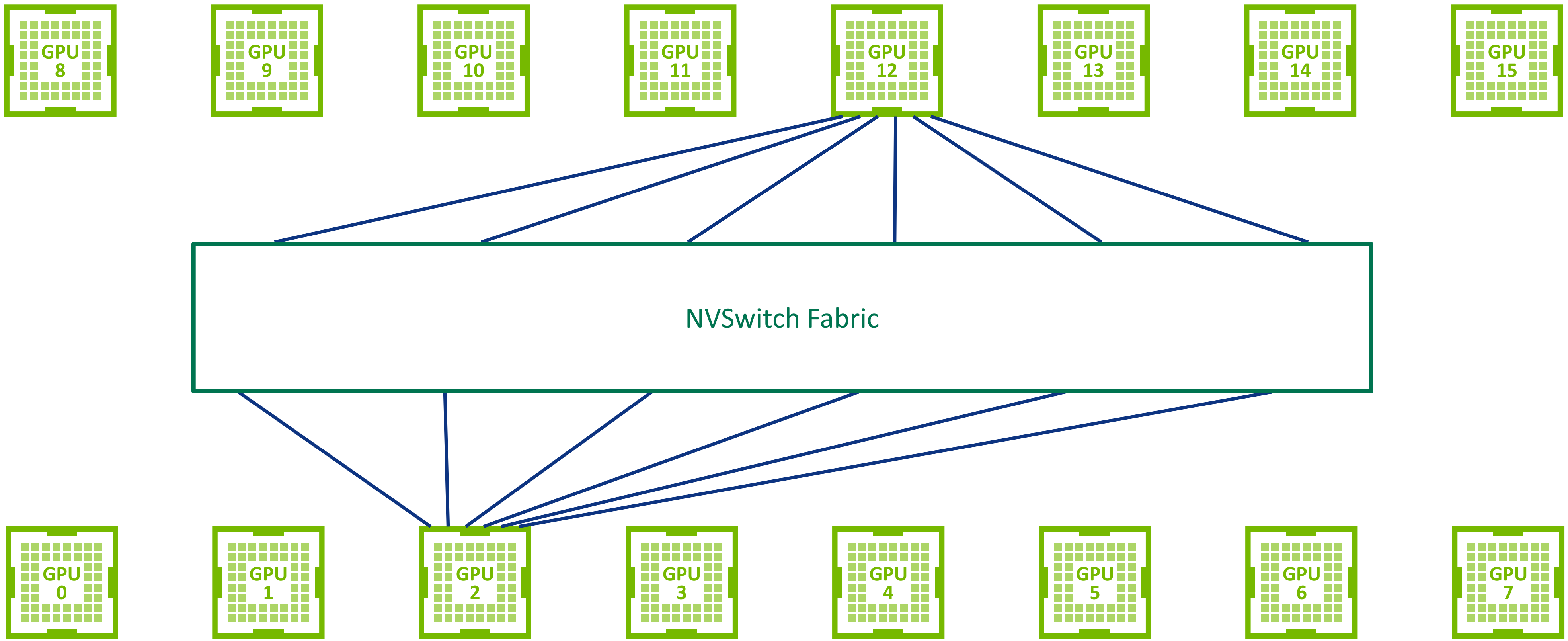
NVSWITCH

- 18 NVLINK ports
  - @50 GB/s per port bi-directional
  - 900 GB/s total bi-directional
- Fully connected crossbar
- X4 PCIe Gen2 Management port
- GPIO
- I2C
- 2 billion transistors

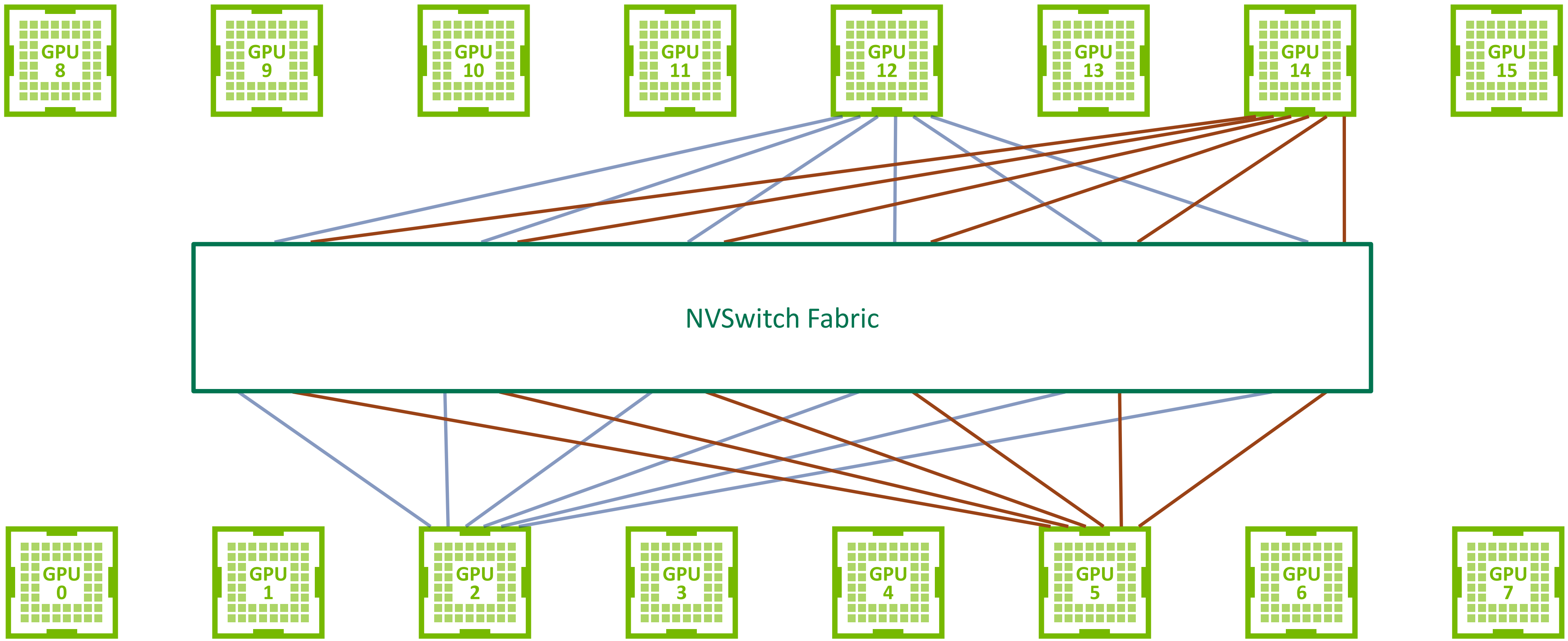
FULL NON-BLOCKING BANDWIDTH



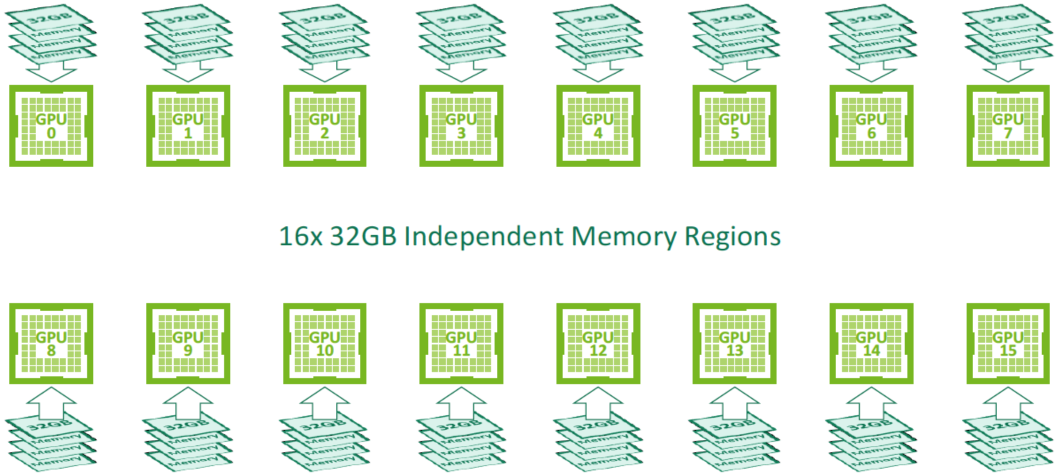
FULL 6-WAY POINT-TO-POINT



INDEPENDENT COMMUNICATION

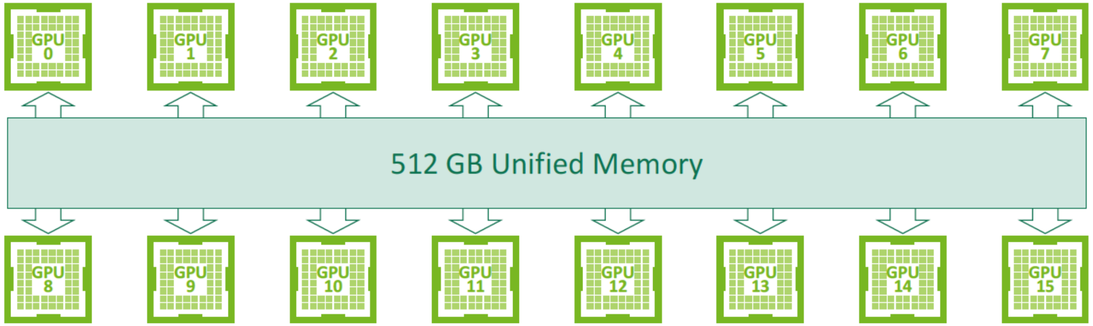


NVSWITCH



NVLINK PROVIDES

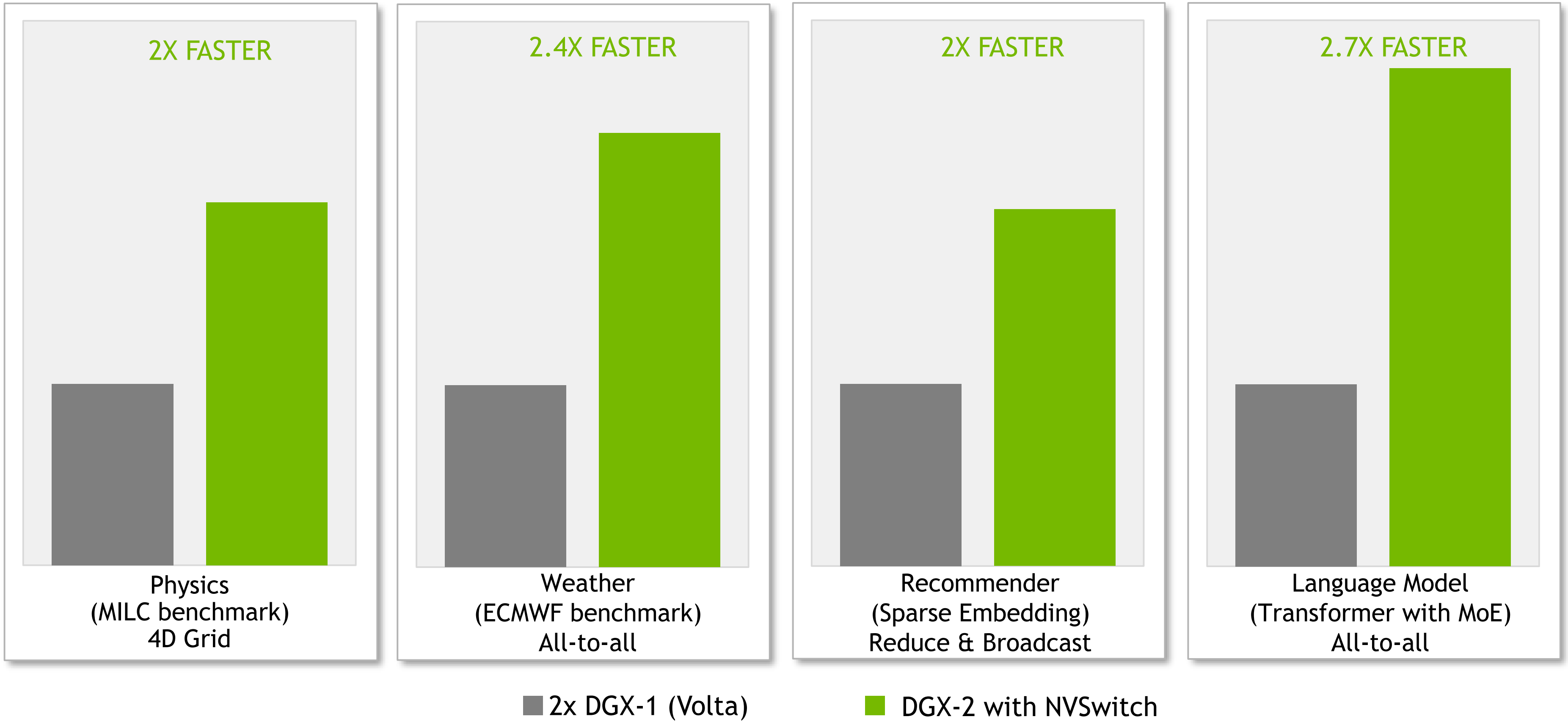
- All-to-all high-bandwidth peer mapping between GPUs
- Full inter-GPU memory interconnect (incl. Atomics)



UNIFIED MEMORY PROVIDES

- Single memory view shared by all GPUs
- Automatic migration of data between GPUs
- User control of data locality

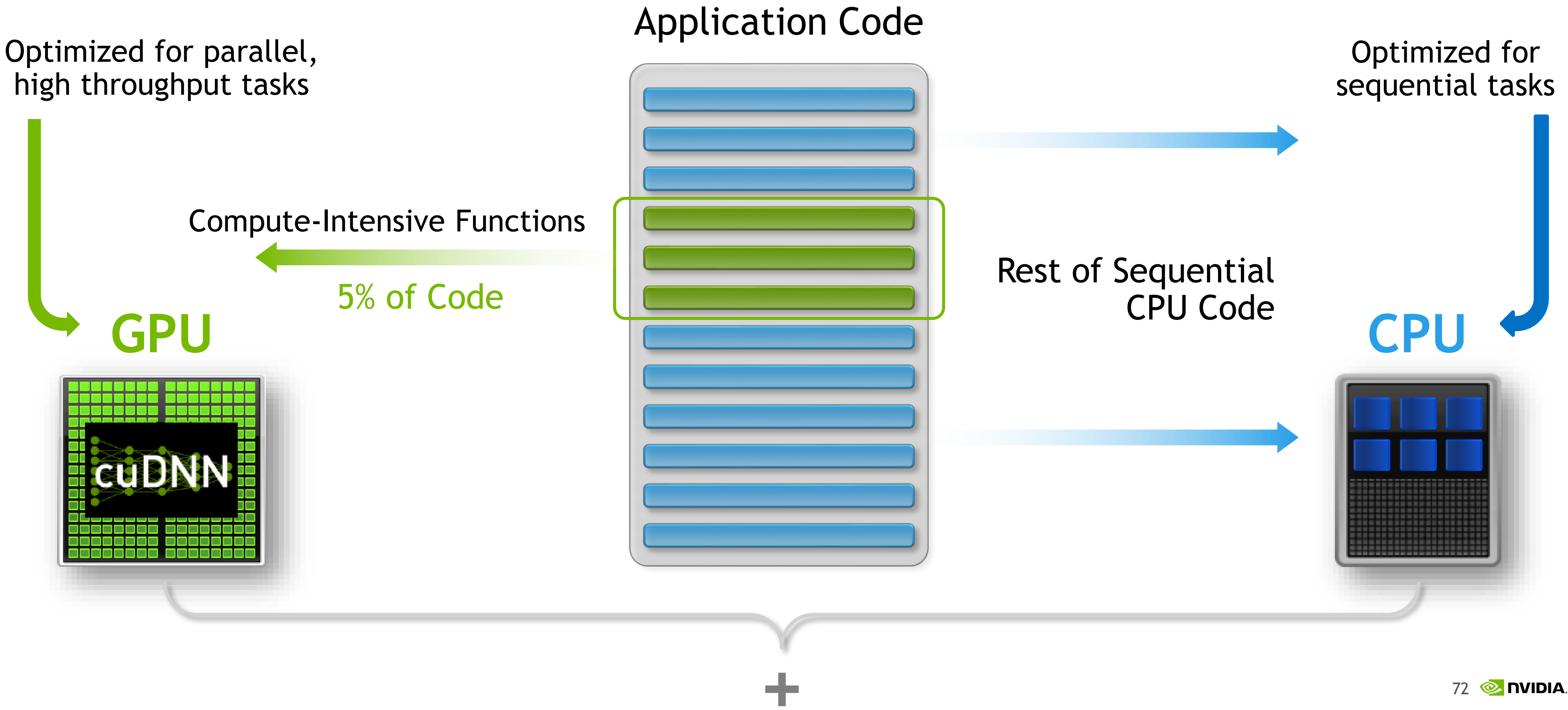
2X HIGHER PERFORMANCE WITH NVSWITCH



2 DGX-1V servers have dual socket Xeon E5 2698v4 Processor. 8 x V100 GPUs. Servers connected via 4X 100Gb IB ports | DGX-2 server has dual-socket Xeon Platinum 8168 Processor. 16 V100 GPUs

GPU PROGRAMMING

HOW GPU ACCELERATION WORKS



HOW TO START WITH GPUS

1 Applications		
2 Libraries	3 Compiler Directives	4 Programming Languages
Easy to use	Easy to Start	Most Performance
Most Performance	Portable Code	Most Flexibility
	OpenACC	CUDA

1. Review available GPU-accelerated applications

2. Check for GPU-Accelerated applications and libraries

3. Add OpenACC Directives for quick acceleration results and portability

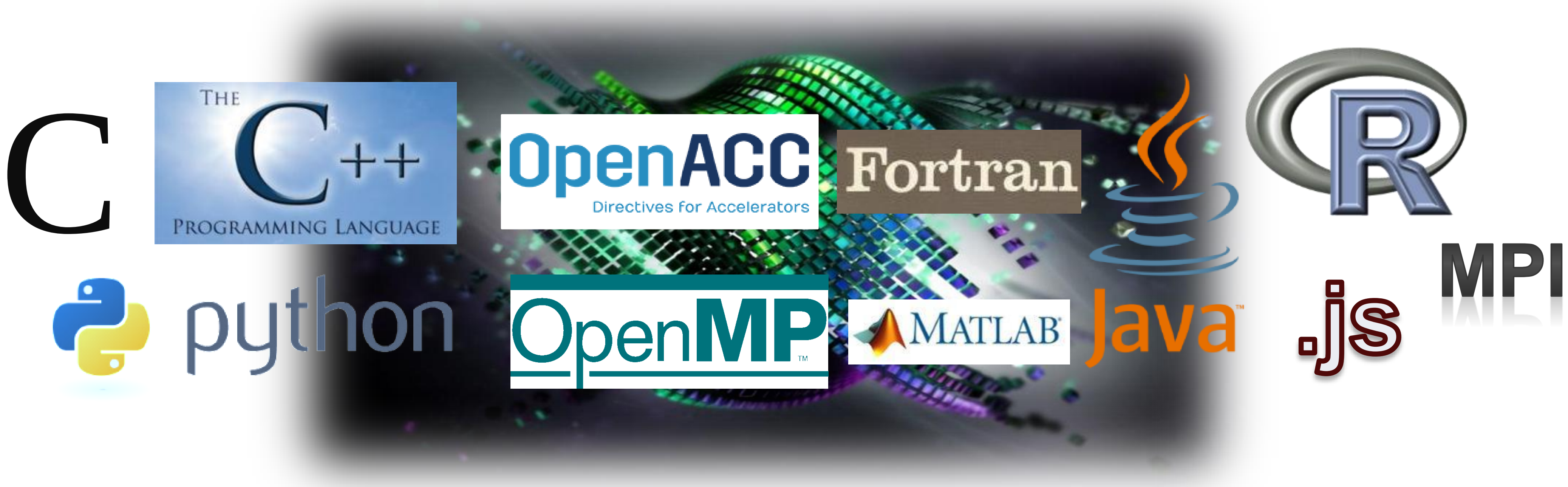
4. Dive into CUDA for highest performance and flexibility

VISION: MAINSTREAM PARALLEL PROGRAMMING

Enable more programmers to write portable parallel software in their language of choice

Embrace and evolve standards in key languages

CUDA continues to evolve as the target low-level platform for GPU acceleration



550+ GPU-Accelerated Applications

www.nvidia.com/appscatalog

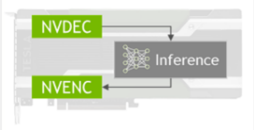
GPU ACCELERATED LIBRARIES

"Drop-in" Acceleration for Your Applications

DEEP LEARNING

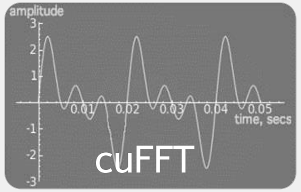
cuDNN

TensorRT

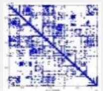


DeepStream SDK

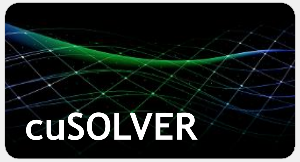
SIGNAL, IMAGE & VIDEO

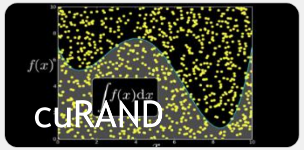


LINEAR ALGEBRA

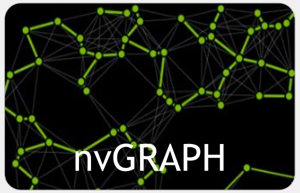


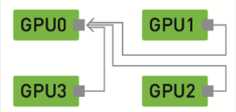
cuSPARSE





PARALLEL ALGORITHMS





NCCL

Thrust

OPENACC

WHAT IS OPENACC

Programming model for an easy onramp to GPUs

Directives-based programming model for **parallel computing**

Add Simple Compiler Directive

```
main()
 {
  <serial code>
  #pragma acc kernels
  {
    <parallel code>
  }
 }
```

Simple

Designed for **performance portability** on CPUs and GPUs

Powerful & Portable

Read more at www.openacc.org/about

OpenACC is an open specification developed by OpenACC.org consortium

OPENACC

Three major concepts

Incremental	Single Source	Low Learning Curve

OPENACC

Incremental

- Start with a working sequential code, and add parallelism
- Make small, incremental changes to the code
- If any errors occur, easily able to revert back to an earlier, working version of the code

Enhance Sequential Code

```
#pragma acc parallel loop
for( i = 0; i < N; i++ )
{
    < loop code >
}

#pragma acc parallel loop
for( i = 0; i < N; i++ )
{
    < loop code >
}
```

Begin with a working sequential code.

Parallelize it with OpenACC.

Rerun the code to verify correct behavior, remove/alter OpenACC code as needed.

OPENACC

Incremental

- Make small, incremental changes to the code
- If any errors occur, easily able to revert back to an earlier, working version of the code
- Start with a working sequential code, and add improvements

Single Source

Low Learning Curve

OPENACC

Supported Platforms

POWER

Sunway

x86 CPU

x86 Xeon Phi

NVIDIA GPU

PEZY-SC

Single Source

- A single OpenACC code can be compiled for, and ran on, many different parallel hardware
- An OpenACC code retains its ability to run sequentially at all times
- No need for multiple versions of your code

The compiler can be told to **ignore** your OpenACC code additions. This allows you to run the code **sequentially**, regardless of the presence of **OpenACC directives.**

```
int main(){

...

    #pragma acc parallel loop
    for(int i = 0; i < N; i++)
        < loop code >

}
```

OPENACC IS FOR MULTICORE, MANYCORE & GPUS

```
 98 !$ACC KERNELS
 99 !$ACC LOOP INDEPENDENT
100      DO k=y_min-depth,y_max+depth
101 !$ACC LOOP INDEPENDENT
102        DO j=1,depth
103          density0(x_min-j,k)=left_density0(left_xmax+1-j,k)
104        ENDDO
105      ENDDO
106 !$ACC END KERNELS
```

CPU

GPU

```
% pgfortran -ta=multicore -fast -Minfo=acc -c \
  update_tile_halo_kernel.f90
  . . .
  100, Loop is parallelizable
       Generating Multicore code
       100, !$acc loop gang
  102, Loop is parallelizable
```

```
% pgfortran -ta=tesla,cc35,cc60 -fast -Minfo=acc -c \
  update_tile_halo_kernel.f90
  . . .
  100, Loop is parallelizable
  102, Loop is parallelizable
       Accelerator kernel generated
       Generating Tesla code
       100, !$acc loop gang, vector(4) ! blockidx%y threadidx%y
       102, !$acc loop gang, vector(32) ! blockidx%x threadidx%x
```

SINGLE CODE FOR MULTIPLE PLATFORMS

OpenACC - Performance Portable Programming Model for HPC

OpenPOWER

Sunway

x86 CPU

x86 Xeon Phi

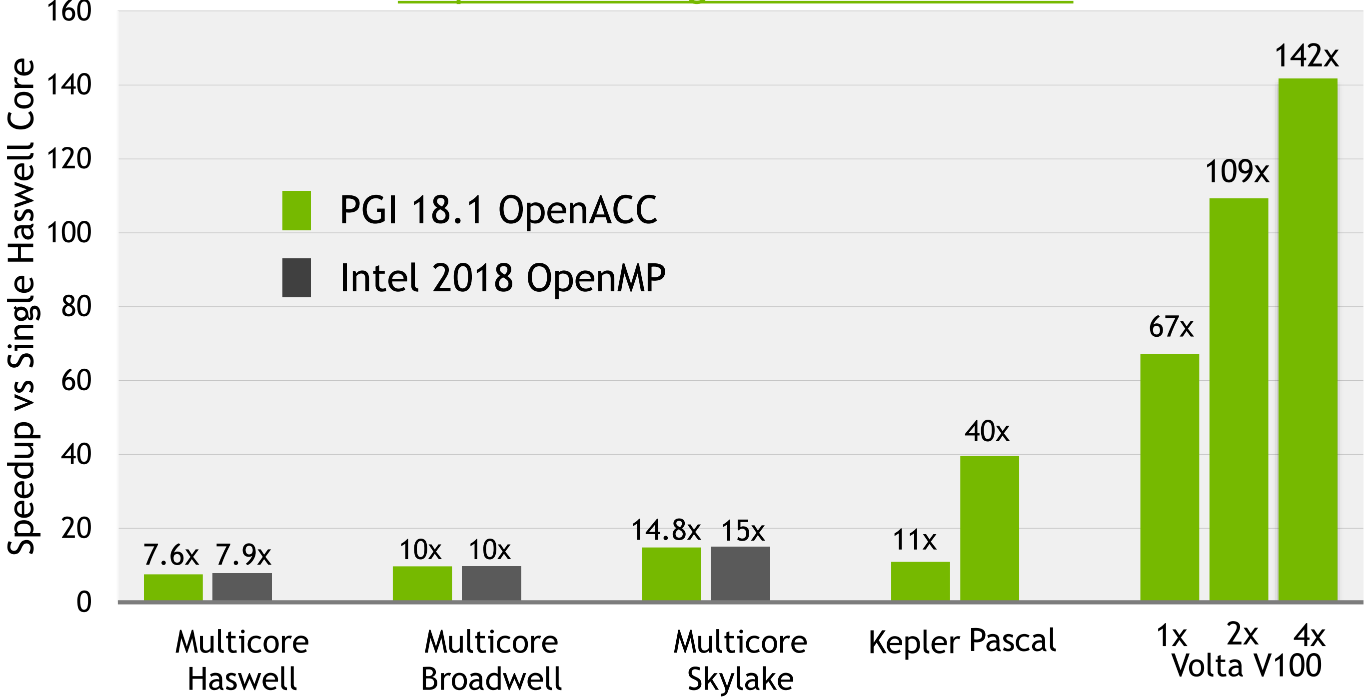
NVIDIA GPU

AMD GPU

PEZY-SC

AWE Hydrodynamics CloverLeaf mini-App, bm32 data set

http://uk-mac.github.io/CloverLeaf



Systems: Haswell: 2x16 core Haswell server, four K80s, CentOS 7.2 (perf-hsw10), Broadwell: 2x20 core Broadwell server, eight P100s (dgx1-prd-01), Broadwell server, eight V100s (dgx07), Skylake 2x20 core Xeon Gold server (sky-4).
Compilers: Intel 2018.0.128, PGI 18.1
Benchmark: CloverLeaf v1.3 downloaded from http://uk-mac.github.io/CloverLeaf the week of November 7 2016; CloverlLeaf_Serial; CloverLeaf_ref (MPI+OpenMP); CloverLeaf_OpenACC (MPI+OpenACC)
Data compiled by PGI February 2018.

OPENACC

Incremental

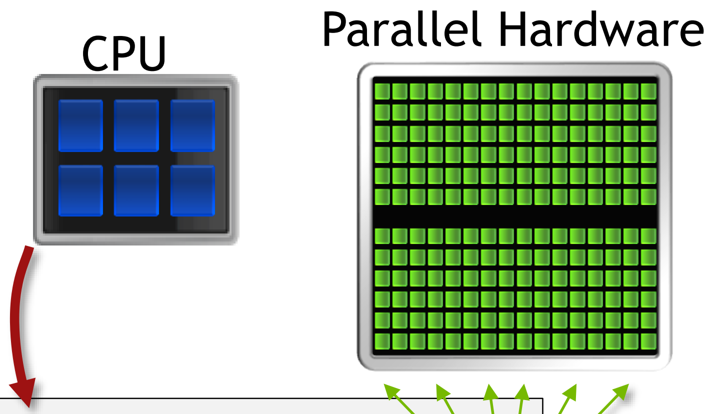
- Make small, incremental changes to the code
- If any errors occur, easily able to revert back to an earlier, working version of the code
- Start with a working sequential code, and add improvements

Single Source

- A single OpenACC code can be compiled for, and ran on, many different parallel hardware
- An OpenACC code retains its ability to run sequentially at all times
- No need for multiple versions of your code

Low Learning Curve

OPENACC




```
int main(){

 <sequential code>

 #pragma acc kernels
 {
  <parallel code>
 }

}
```

Compiler Hint

The programmer will give hints to the compiler about which parts of the code to parallelize.

The compiler will then generate parallelism for the target parallel hardware.

Low Learning Curve

- OpenACC is meant to be easy to use, and easy to learn
- Supports C, C++, and Fortran coding
- Takes a very high-level approach to parallelism, and allows the compiler to do a lot of extra work in parallelizing the code

OPENACC

Incremental

- Make small, incremental changes to the code
- If any errors occur, easily able to revert back to an earlier, working version of the code
- Start with a working sequential code, and add improvements

Single Source

- A single OpenACC code can be compiled for, and ran on, many different parallel hardware
- An OpenACC code retains its ability to run sequentially at all times
- No need for multiple versions of your code

Low Learning Curve

- OpenACC is meant to be easy to use, and easy to learn
- Supports C, C++, and Fortran coding
- Takes a very high-level approach to parallelism, and allows the compiler to do a lot of extra work in parallelizing the code

OPENACC.ORG RESOURCES

Guides ● Talks ● Tutorials ● Videos ● Books ● Spec ● Code Samples ● Teaching Materials ● Events ● Success Stories ● Courses ● Slack ● Stack Overflow

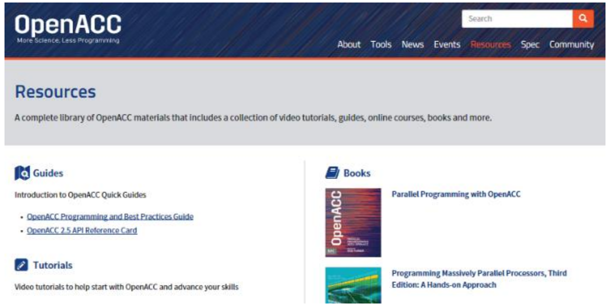
OpenACC Now in GCC

slack

https://www.openacc.org/community#slack

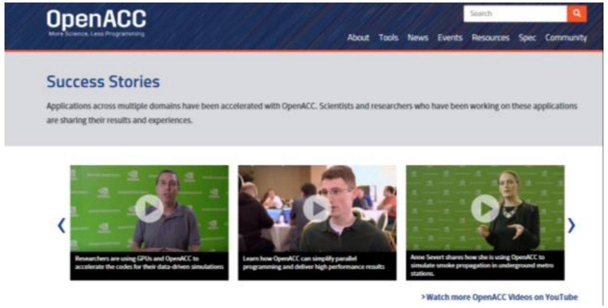
Resources

https://www.openacc.org/resources



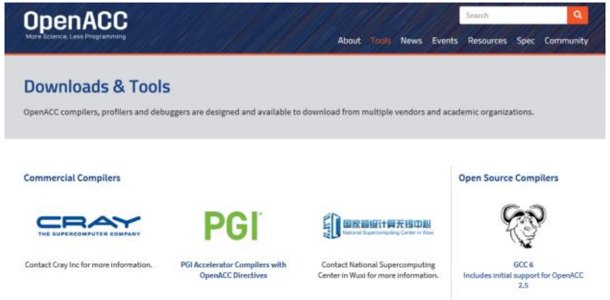
Success Stories

https://www.openacc.org/success-stories



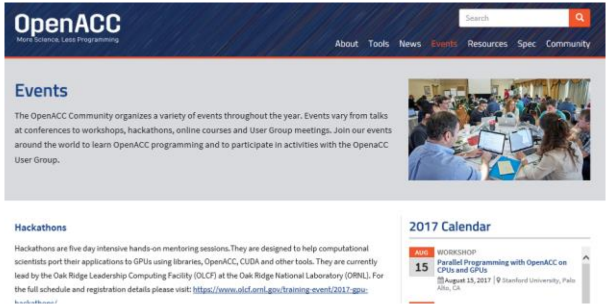
Compilers and Tools

https://www.openacc.org/tools



Events

https://www.openacc.org/events



NVIDIA.

PGI — THE NVIDIA HPC SDK

- Fortran, C & C++ Compilers
  - Optimizing, SIMD Vectorizing, OpenMP
- Accelerated Computing Features
  - OpenACC Directives, CUDA Fortran
- Multi-Platform Solution
  - X86-64 and OpenPOWER Multicore CPUs
  - NVIDIA Tesla GPUs
  - Supported on Linux, macOS, Windows
- MPI/OpenMP/OpenACC Tools
  - Debugger
  - Performance Profiler
  - Interoperable with DDT, TotalView

PGI®

The Compilers & Tools for Supercomputing

PGI COMPILERS FOR EVERYONE

The PGI 18.4 Community Edition

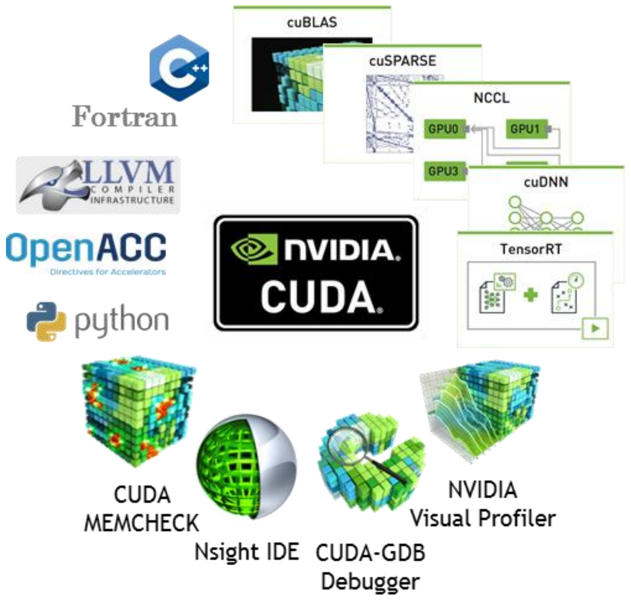
	FREE PGI Community Edition	PGI Professional Edition	PGI Enterprise Edition
PROGRAMMING MODELS OpenACC, CUDA Fortran, OpenMP, C/C++/Fortran Compilers and Tools	✓	✓	✓
PLATFORMS X86, OpenPOWER, NVIDIA GPU	✓	✓	✓
UPDATES	1-2 times a year	6-9 times a year	6-9 times a year
SUPPORT	User Forums	PGI Support	PGI Premier Services
LICENSE	Annual	Perpetual	Volume/Site

pgicompilers.com/community

CUDA

CUDA RELEASES

Accelerating the Pace



Four CUDA releases per year

Faster release cadence for new features and improved stability for existing users

Upcoming limited decoupling of display driver and CUDA release for ease of deployment

Monthly cuDNN & other library updates

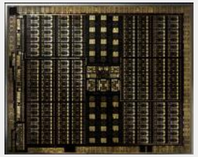
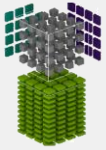
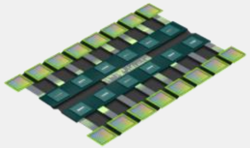
Rapid innovation in library performance and functionality

Library Meta Packages independent of toolkit for easy deployment

INTRODUCING CUDA 10.0

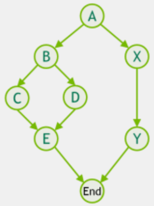
TURING AND NEW SYSTEMS

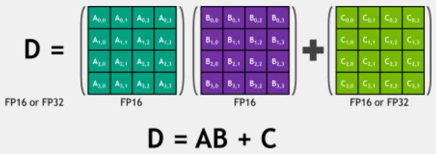
New GPU Architecture, Tensor Cores, NVSwitch Fabric


CUDA PLATFORM

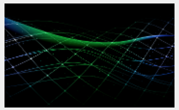
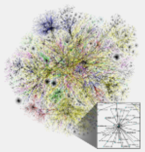
CUDA Graphs, Vulkan & DX12 Interop, Warp Matrix





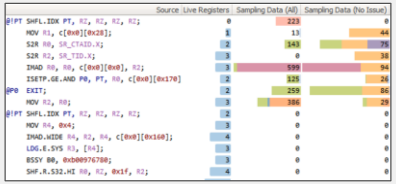
LIBRARIES

GPU-accelerated hybrid JPEG decoding,
Symmetric Eigenvalue Solvers, FFT Scaling


DEVELOPER TOOLS

New Nsight Products – Nsight Systems and Nsight Compute

CUDA LIBRARIES

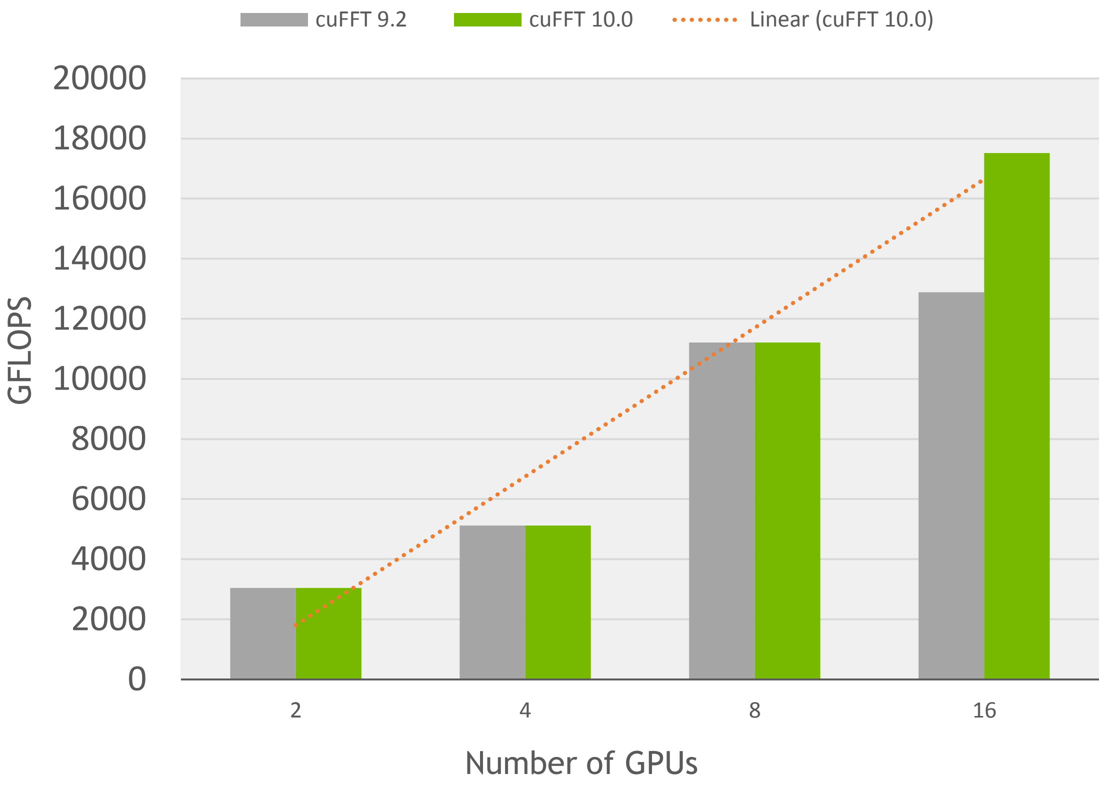
cuFFT 10.0

Multi-GPU Scaling across DGX-2 and HGX-2

- Strong scaling across 16-GPU systems – DGX-2 and HGX-2
- Multi-GPU R2C and C2R support
- Large FFT models across 16-GPUs – effective 512GB vs 32GB capacity

https://developer.nvidia.com/cufft

Up to 17TF performance on 16-GPUs
3D 1K FFT



cuFFT (10.0 and 9.2) using 3D C2C FFT 1024 size on DGX-2 with CUDA 10 (10.0.130)

cuSOLVER 10.0

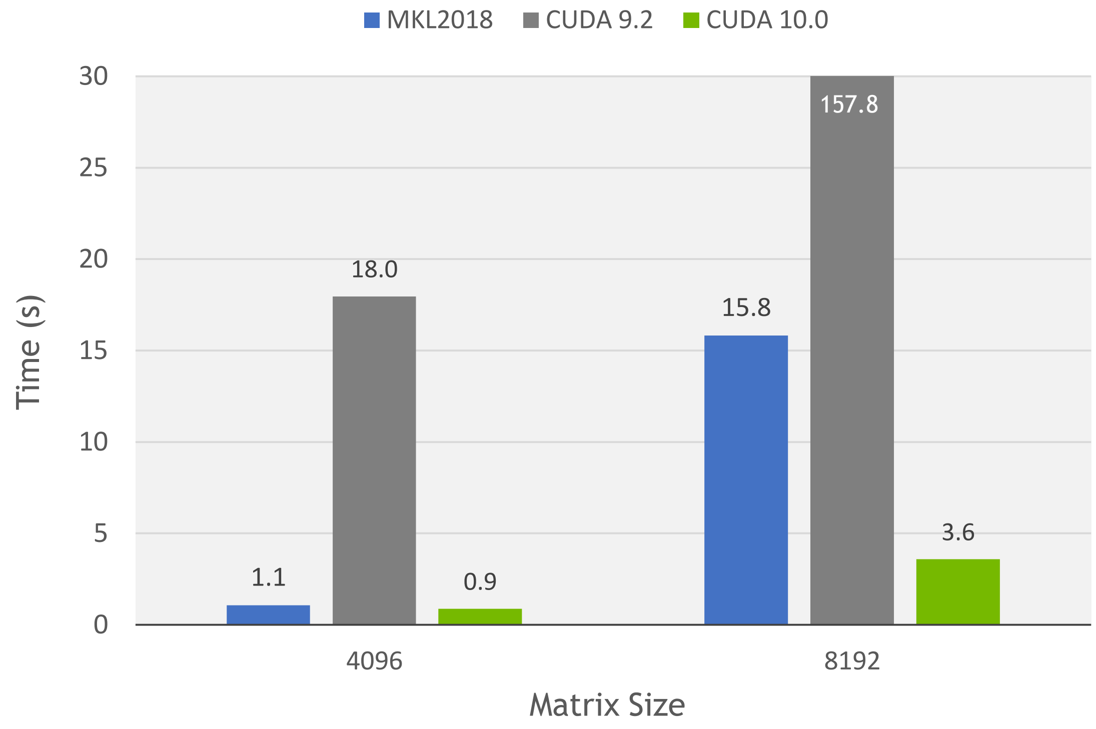
Dense Linear Algebra

Improved performance with new implementations for

- Cholesky factorization
- Symmetric & Generalized Symmetric Eigensolver
- QR factorization

https://developer.nvidia.com/cusparse

Up to 44x Faster on Symmetric Eigensolver (DSYEVD)



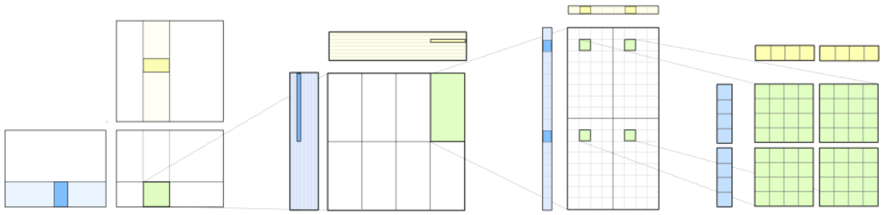
Benchmarks use 2 x Intel Gold 6140 (Skylake) processors with Intel MKL 2018 and NVIDIA Tesla V100 (Volta) GPUs

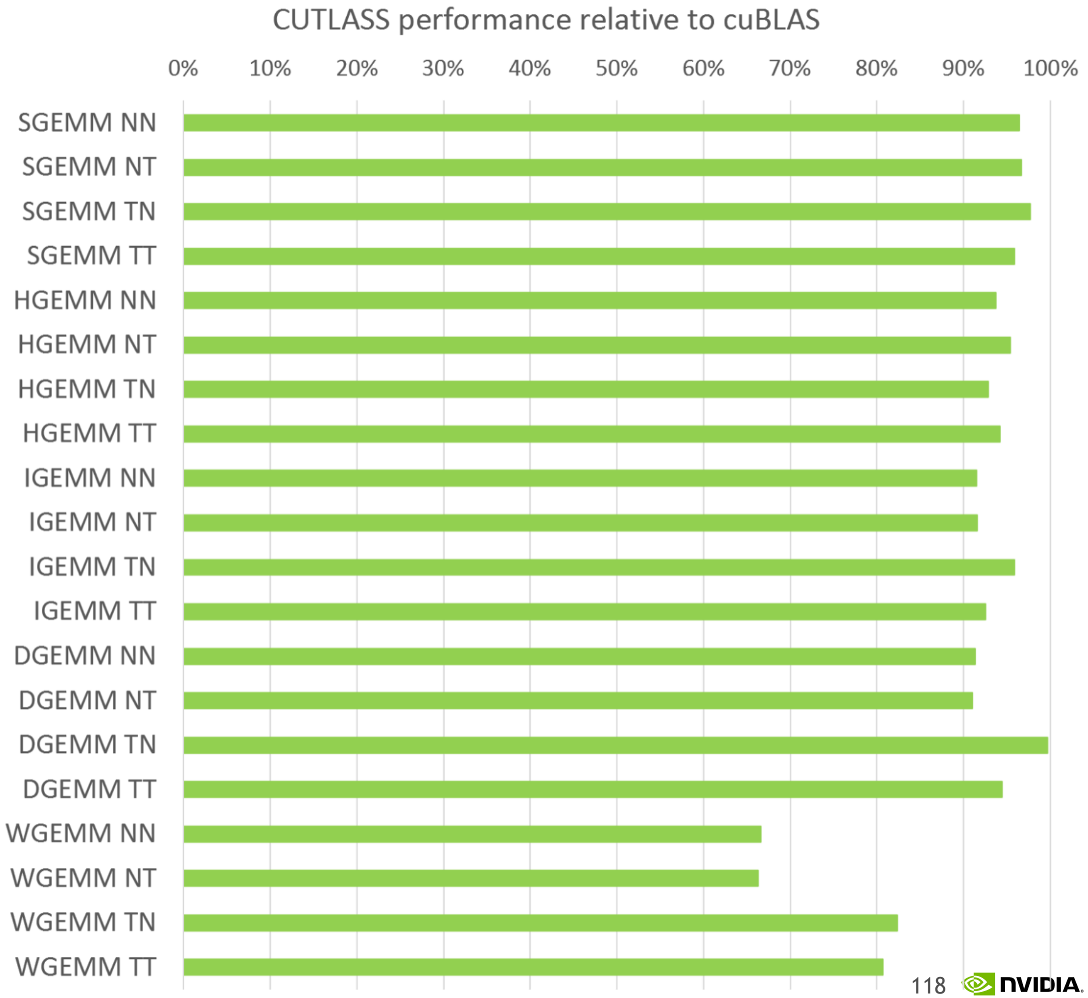
CUTLASS

Template library for linear algebra operations in CUDA C++

>90% CUBLAS performance

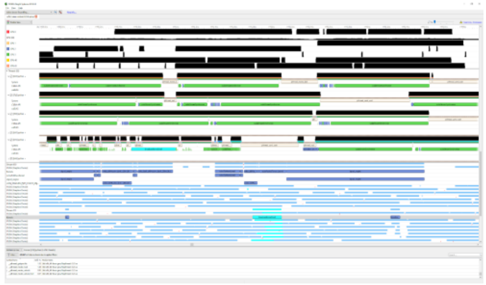
Open Source (3-clause BSD License)
https://github.com/NVIDIA/cutlass





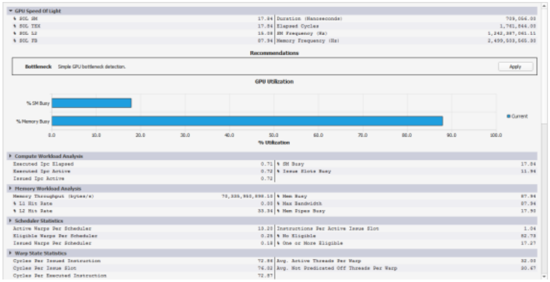
NSIGHT DEVELOPER TOOLS

NSIGHT PRODUCT FAMILY



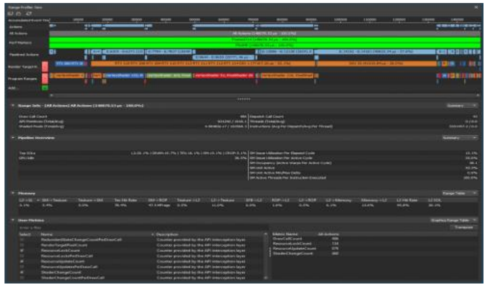
Nsight Systems

System-wide application algorithm tuning



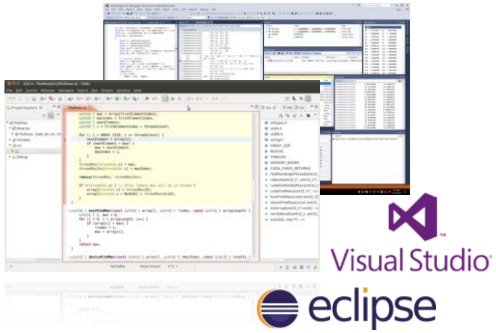
Nsight Compute

CUDA Kernel Profiling and Debugging



Nsight Graphics

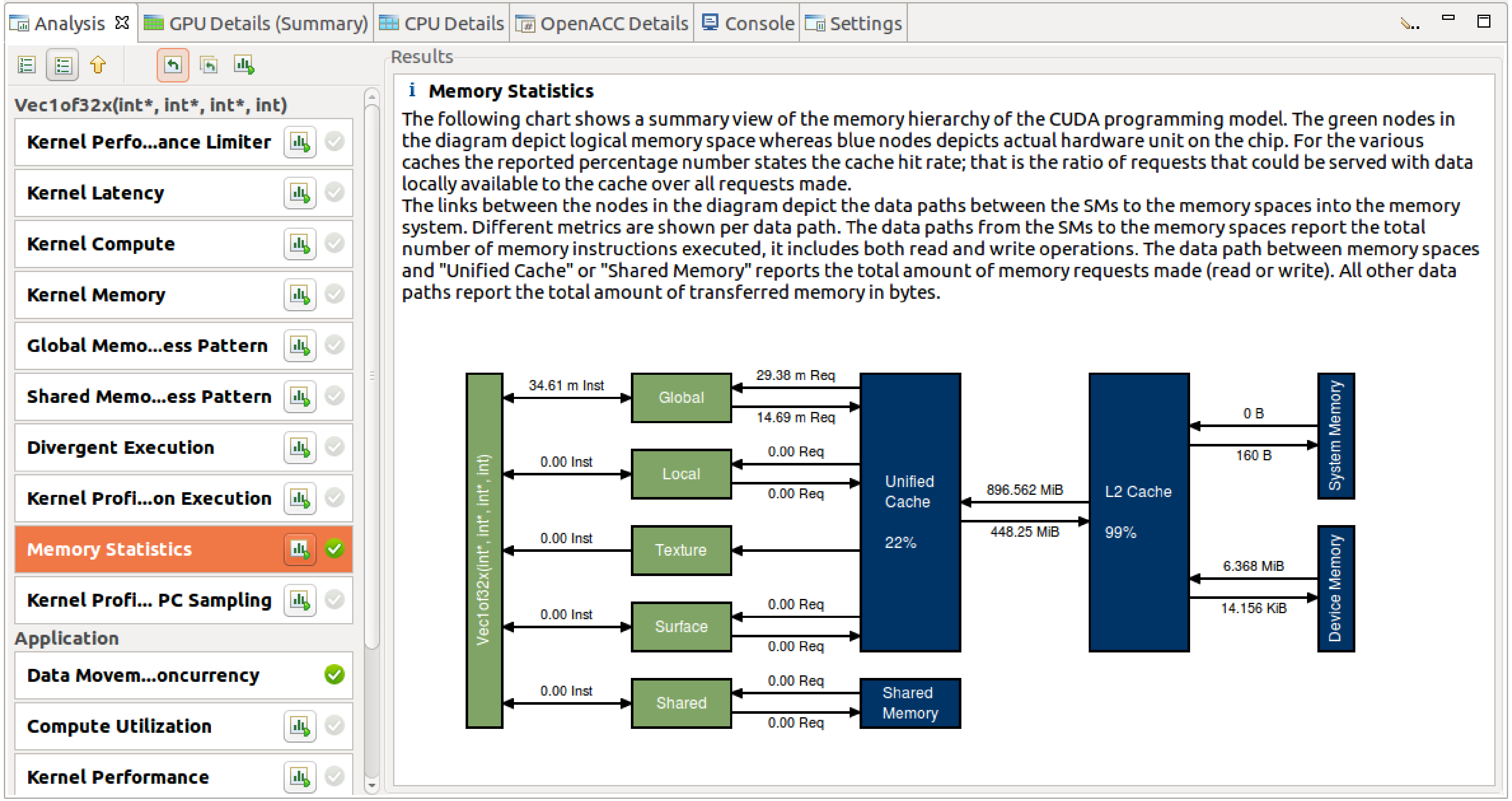
Graphics Shader Profiling and Debugging



IDE Plugins

Nsight Eclipse Edition/Visual Studio (Editor, Debugger)

HIERARCHICAL MEMORY STATISTICS



NAVIGATING TO COURSES

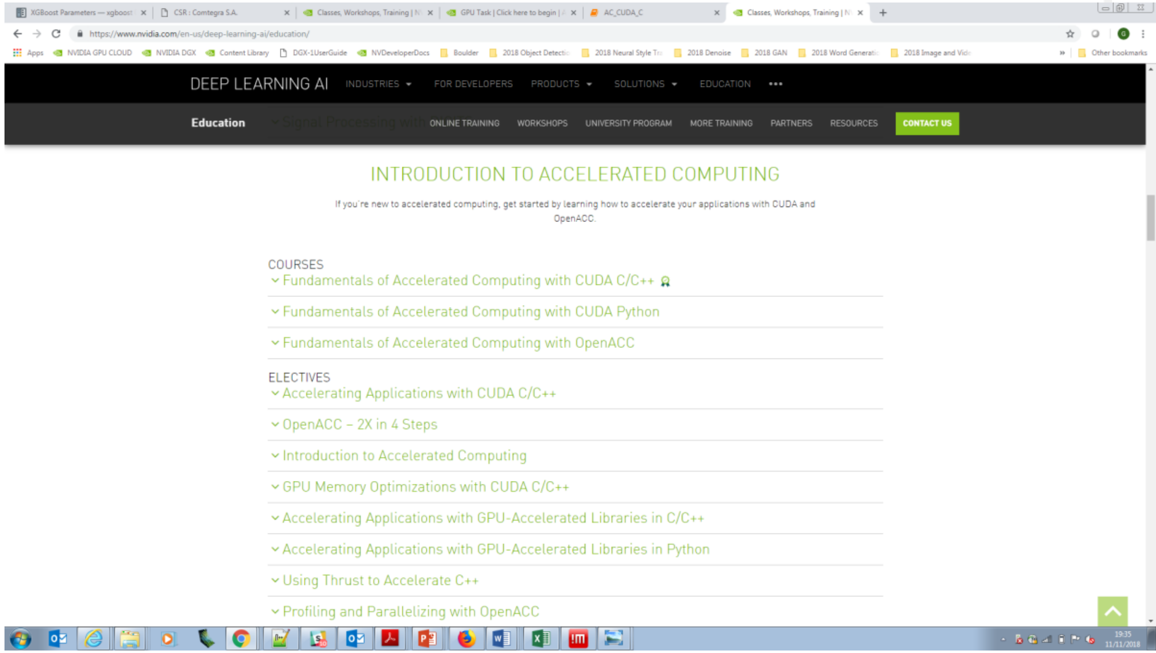
1. Navigate to:
   www.nvidia.co.uk/dlilabs

2. Google search for
   nvidia dli

3. Scroll down
   Training Online ELECTIVES

Use NV Developer login or new account.

Accelerating Applications with CUDA C/C++



DEEP LEARNING SDK

NVIDIA DEEP LEARNING INSTITUTE

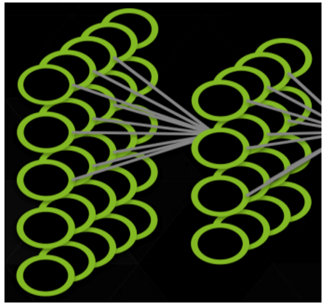
Online self-paced labs and instructor-led workshops on deep learning and accelerated computing

Take self-paced labs at **www.nvidia.co.uk/dlilabs**

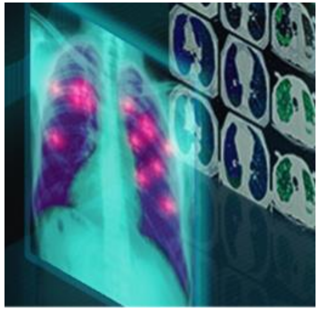
View upcoming workshops and request a workshop onsite at **www.nvidia.co.uk/dli**

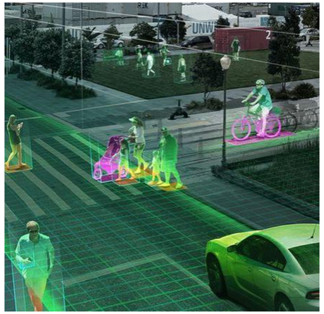
Educators can join the University Ambassador Program to teach DLI courses on campus and access resources. Learn more at **www.nvidia.com/dli**

Caffe2 | Microsoft Cognitive Toolkit | mxnet | TensorFlow | PyTorch


Fundamentals

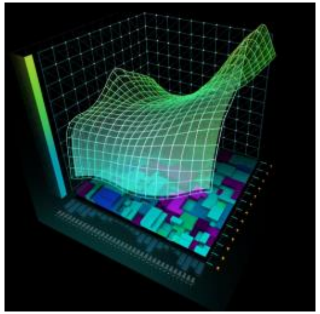

Autonomous Vehicles

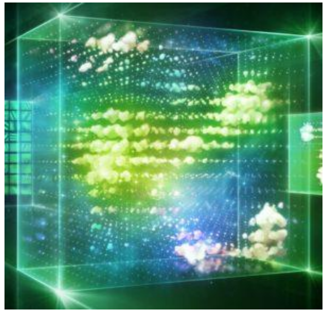

Healthcare


Intelligent Video Analytics


Robotics


Game Development & Digital Content


Finance

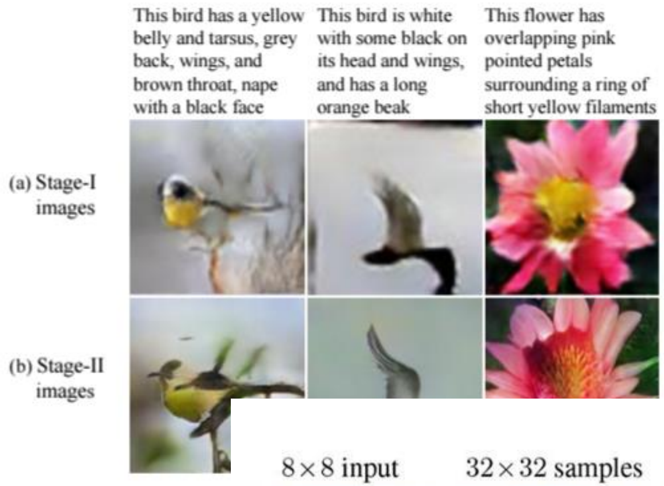

Accelerated Computing

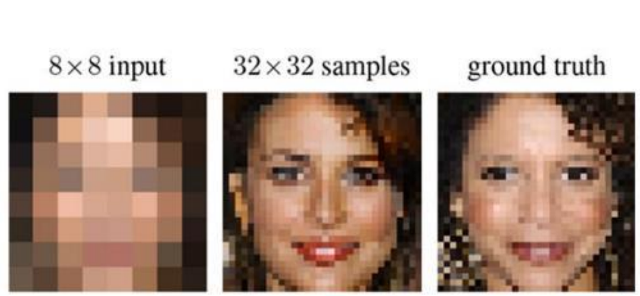

Virtual Reality

NVIDIA DEEP LEARNING INSTITUTE

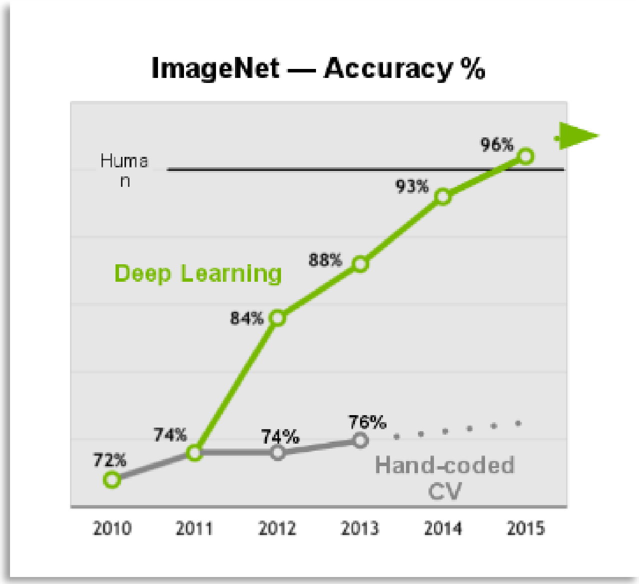
WHY THE EXCITEMENT?

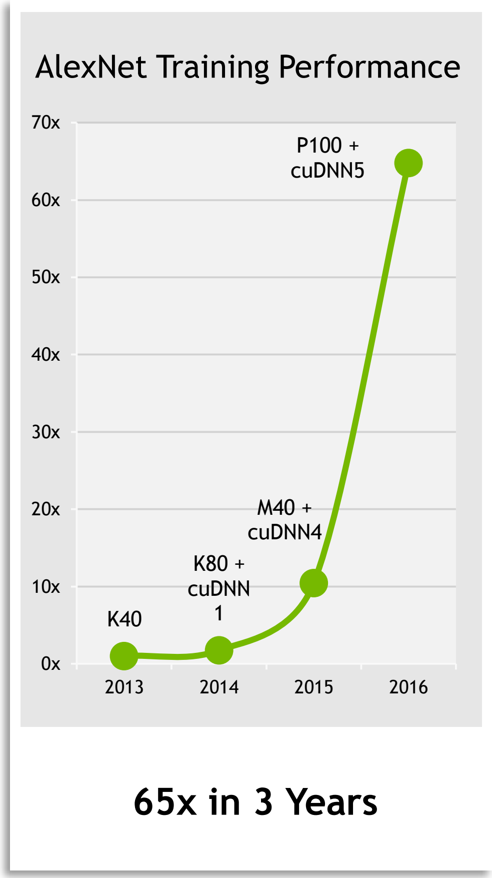
GPUs as Enablers of Breakthrough Results





Dahl et al. 2017





We can generate photorealistic images from textual descriptions and super-enhance blurry photos!

Achieve super-human accuracy in classification

And we are getting faster fast

Paper: H.Zhang et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, arXiv:1612.03242

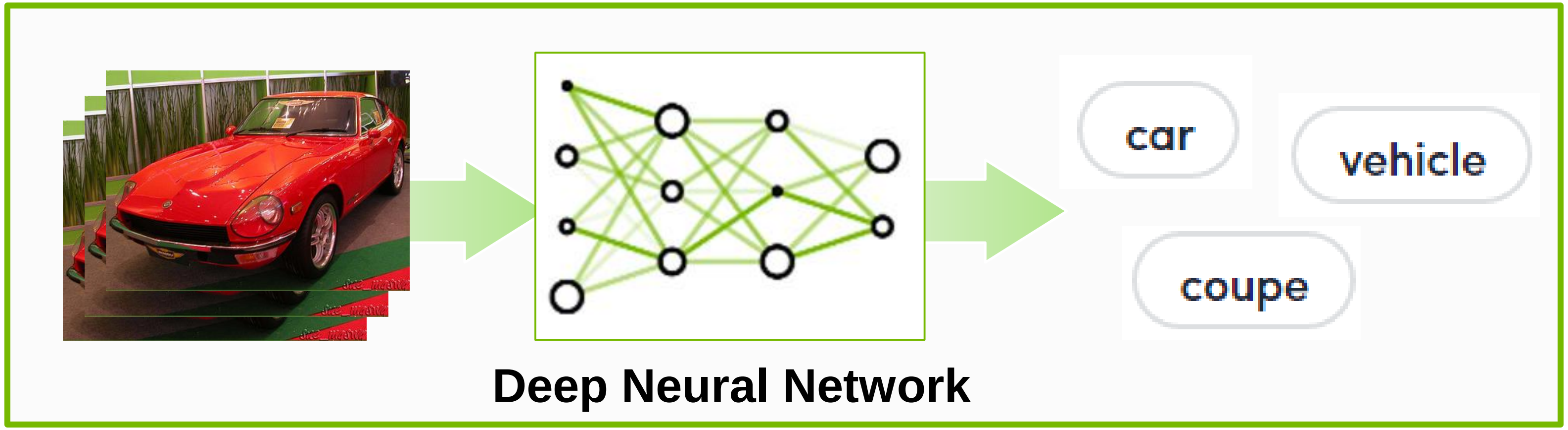
WHAT IS DEEP LEARNING?

A NEW COMPUTING MODEL

Algorithms that Learn from Examples

Traditional CV Approach

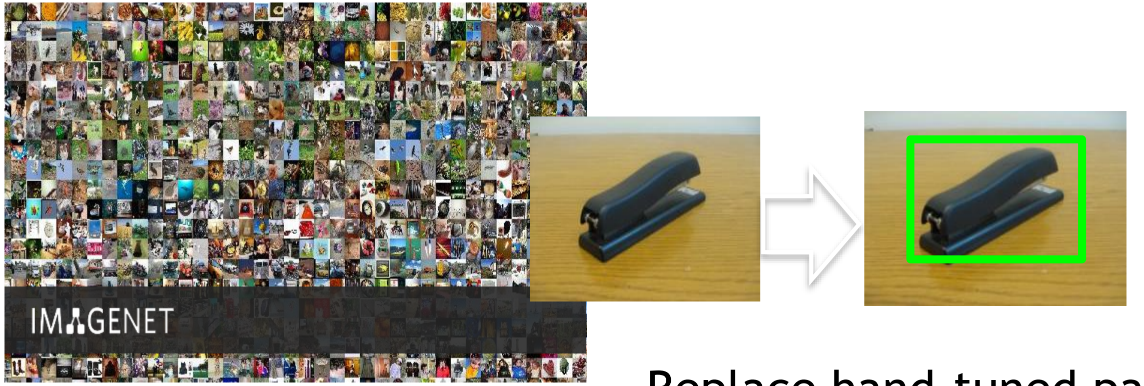
- Domain experts design feature detectors
- Time consuming
- Quality depends on Algorithms
- Error prone
- Not scalable to new problems
- Need CV **experts and time**

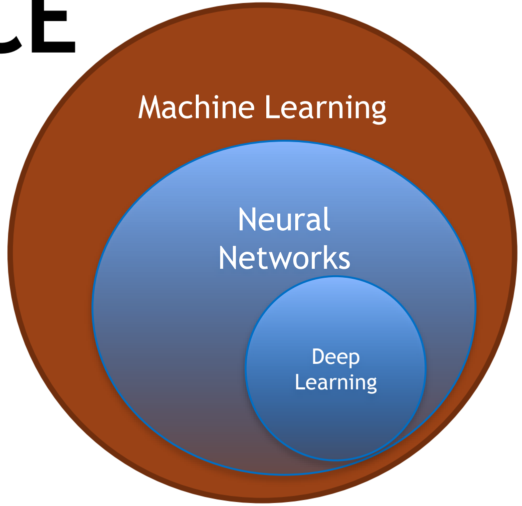


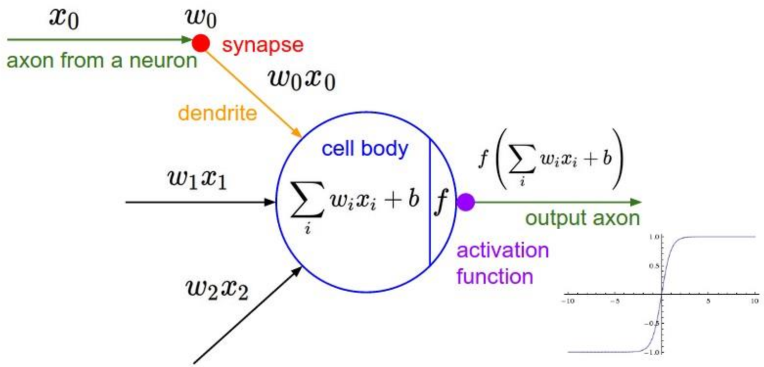
Deep Learning Approach

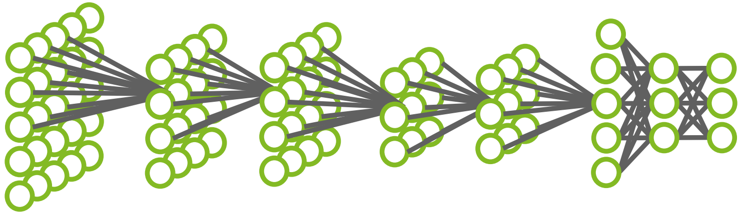
- DNN learn from data
- Quality depends on data & training method
- Easily to extend
- Needs lots of **data and compute**
- Speedup with GPUs

GPUS IN ARTIFICIAL INTELLIGENCE









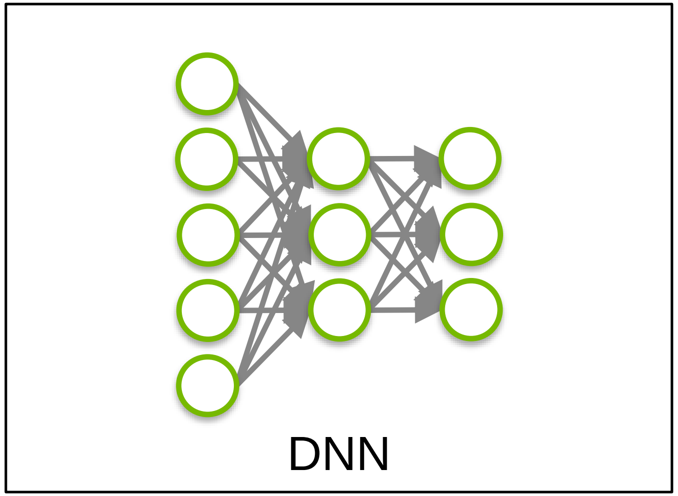
Replace hand-tuned parameters of the feature extraction steps (e.g. in voice and image recognition)

Deep learning is a subset of machine learning that refers to artificial neural networks that are composed of many layers.

Artificial Neural Networks inspired by human brain and need lots of training data (ideal for Big Data).

NVIDIA GPUs and cuDNN software broadly adopted for machine learning.

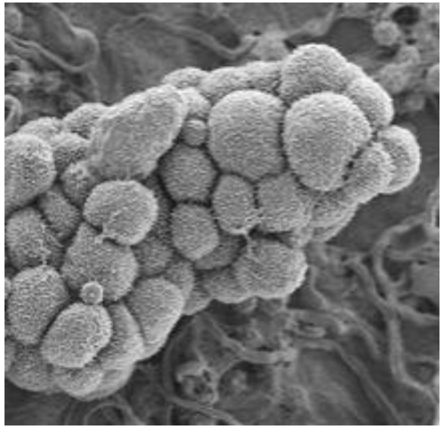
THE BIG BANG IN MACHINE LEARNING

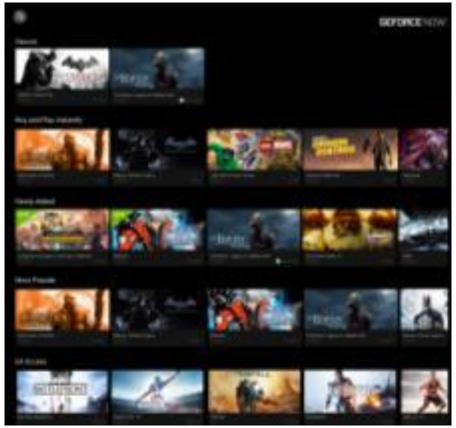


"*Google's AI engine also reflects how the world of computer hardware is changing. (It) depends on machines equipped with GPUs… And it depends on these chips more than the larger tech universe realizes.*"

WIRED

DEEP LEARNING EVERYWHERE





INTERNET & CLOUD

Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation

MEDICINE & BIOLOGY

Cancer Cell Detection
Diabetic Grading
Drug Discovery

MEDIA & ENTERTAINMENT

Video Captioning
Video Search
Real Time Translation

SECURITY & DEFENSE

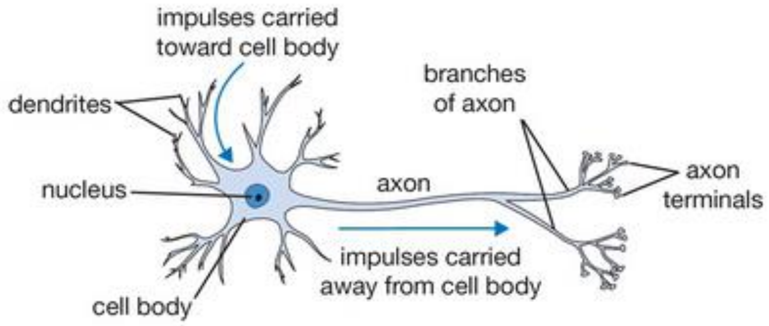
Face Detection
Video Surveillance
Satellite Imagery

AUTONOMOUS MACHINES

Pedestrian Detection
Lane Tracking
Recognize Traffic Sign

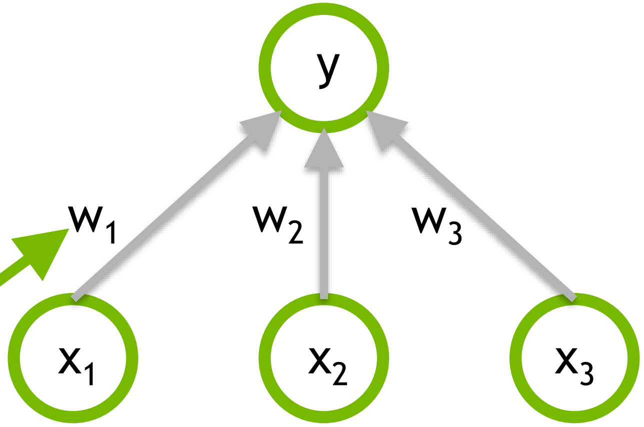
ARTIFICIAL NEURONS

Biological neuron



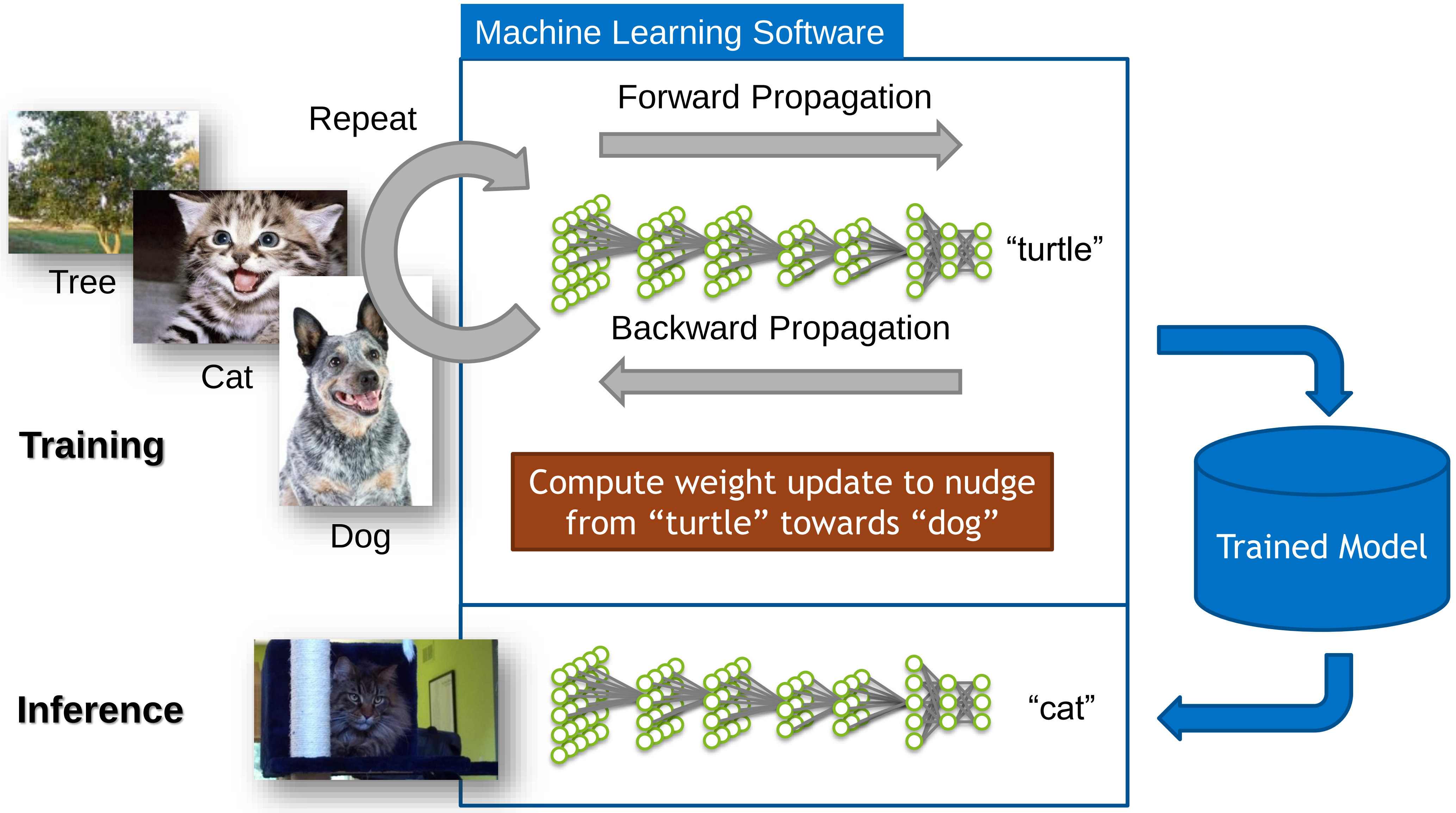
From Stanford cs231n lecture notes

Artificial neuron



Weights (W_n)
= parameters

$$y=F(w_1x_1+w_2x_2+w_3x_3)$$


Machine Learning Software
Repeat
Forward Propagation
Tree
Cat
Dog
"turtle"
Backward Propagation
Training
Compute weight update to nudge from "turtle" towards "dog"
Trained Model
Inference
"cat"

TRAINING NEURAL NETWORKS

Find a set of weights that minimizes the misfit.

Error between the target and computed output

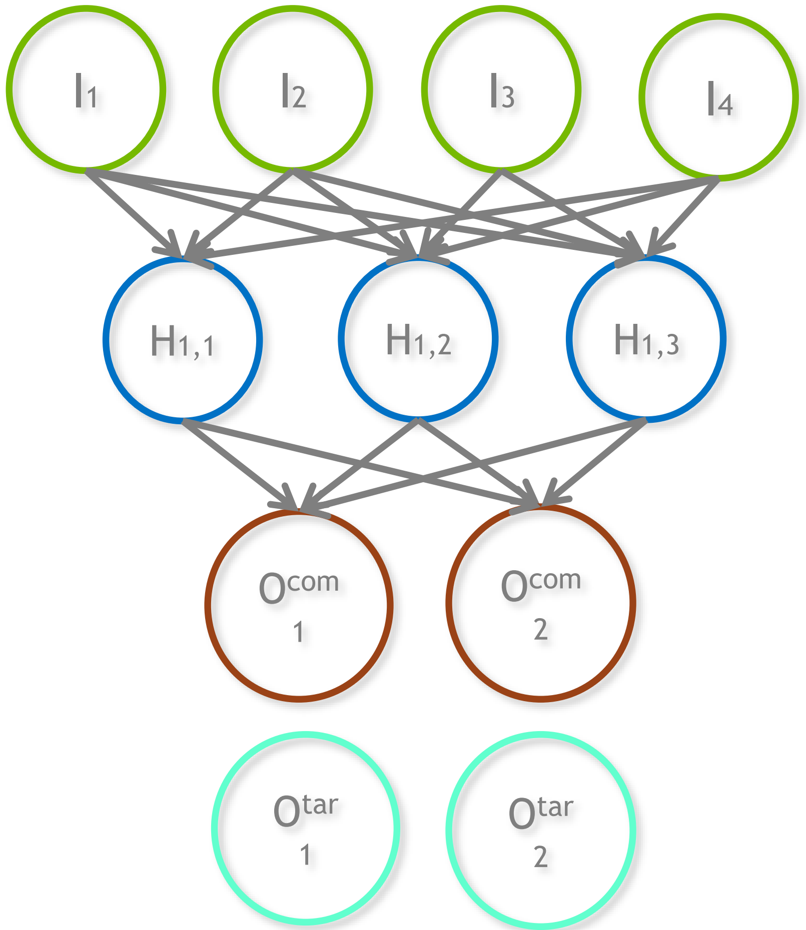
$$\mathrm{M}(\boldsymbol{W}) = \sum_{i=1}^{Examples} \sum_{j=1}^{Output} \left(O_{comp\ i,j} - O_{target\ i,j}\right)^2$$

Least squares optimization problem

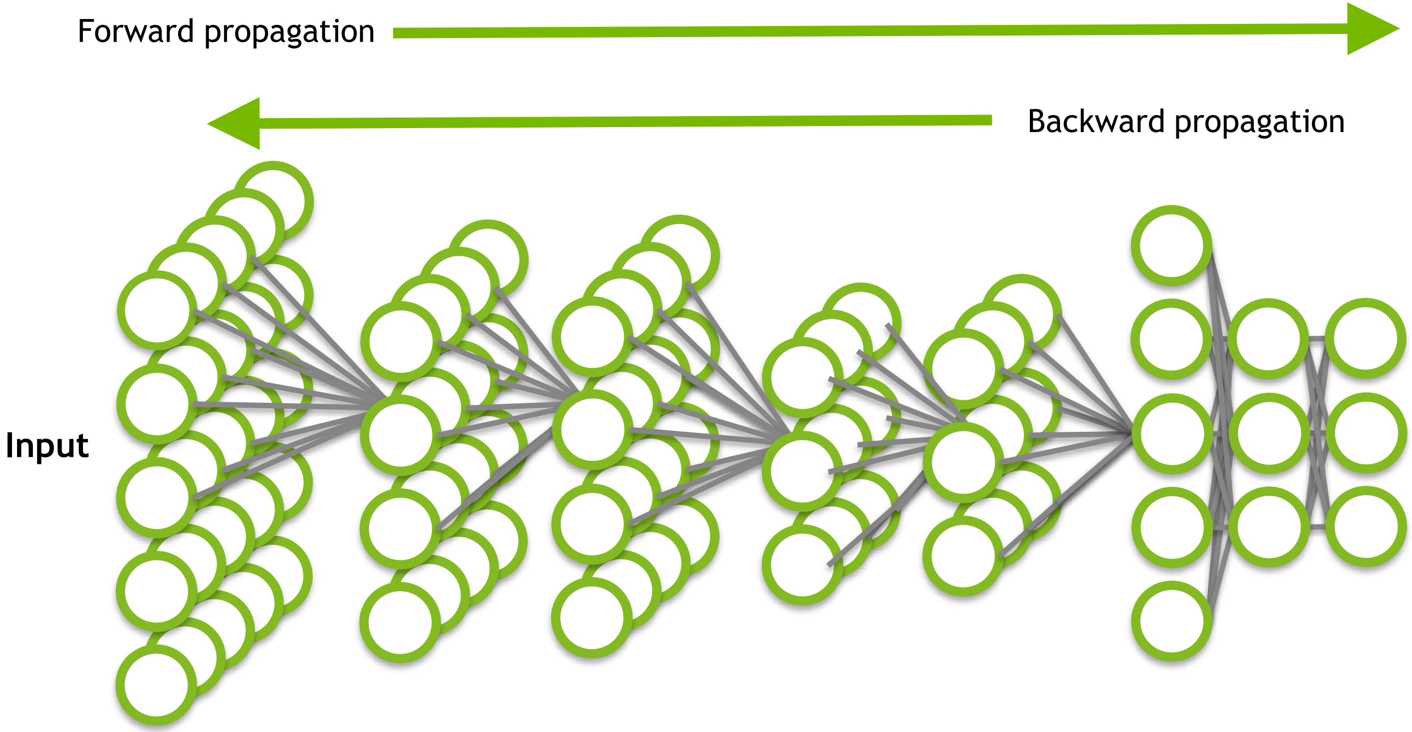
Solution by gradient descent, Monte Carlo, etc.

The gradient can be computed by the backpropagation of the error (delta rule)

$$\delta_{i,j} = g'(I_{i,j})\left(O_{comp\ i,j} - O_{target\ i,j}\right)$$



DEEP LEARNING APPROACH - TRAINING



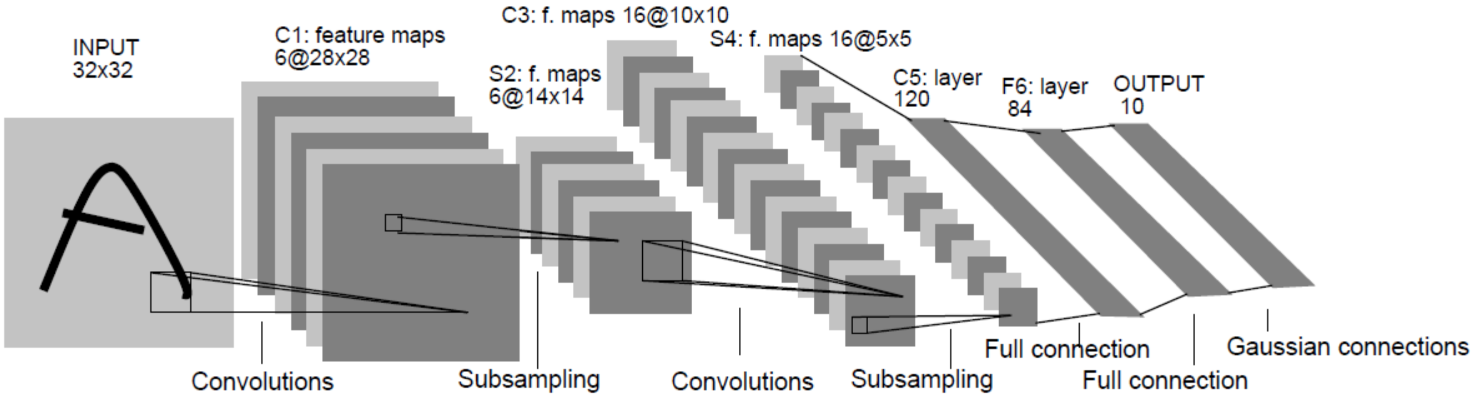
Process

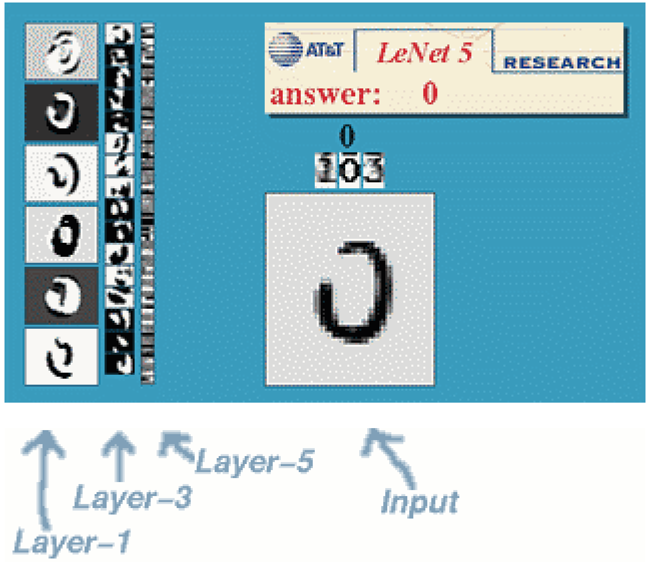
- Forward propagation yields an inferred label for each training image
- Loss function used to calculate difference between known label and predicted label for each image
- Weights are adjusted during backward propagation
- Repeat the process

Convolutional Networks Used Case

Local receptive field + weight sharing

Yann LeCun et al, 1998

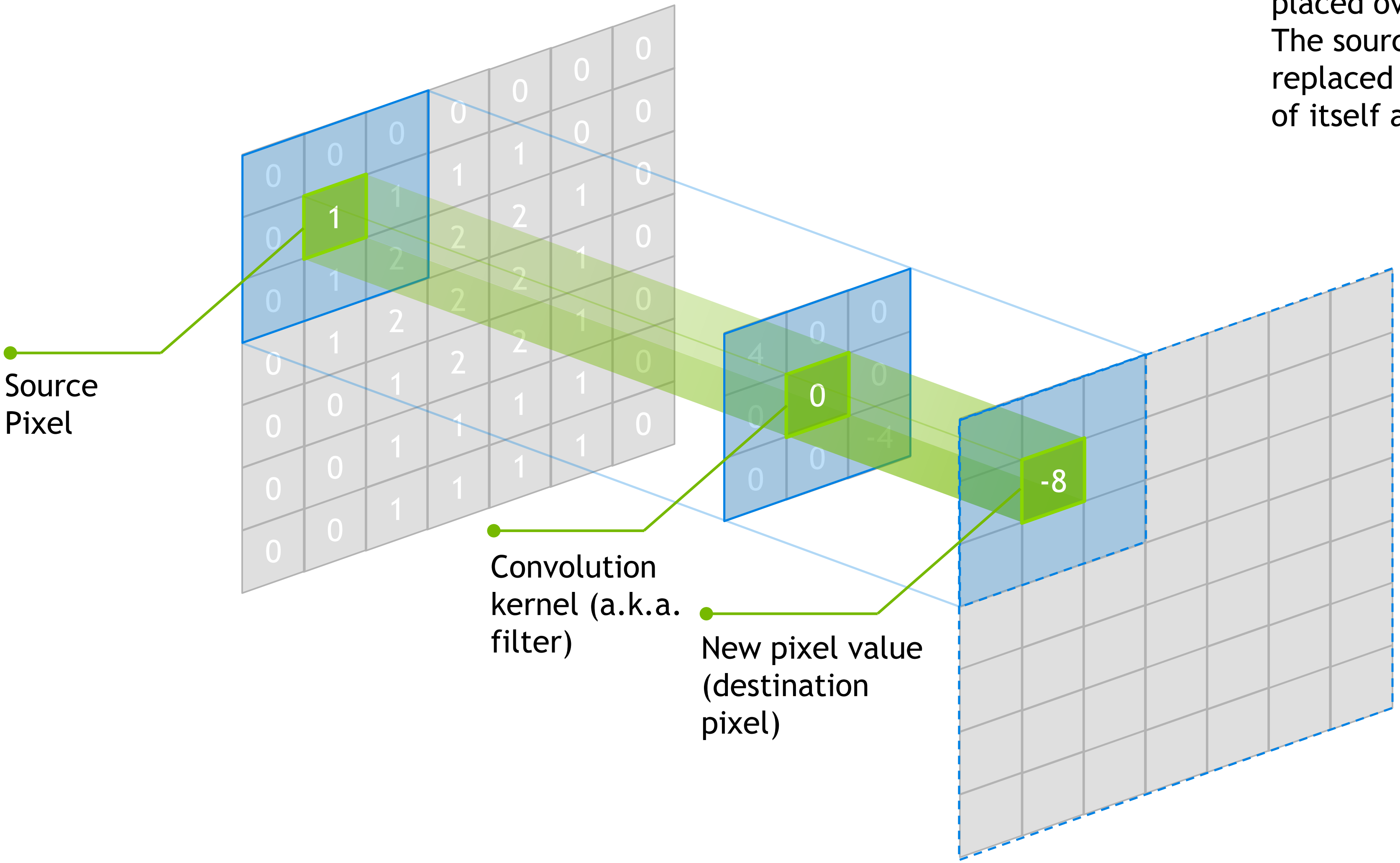




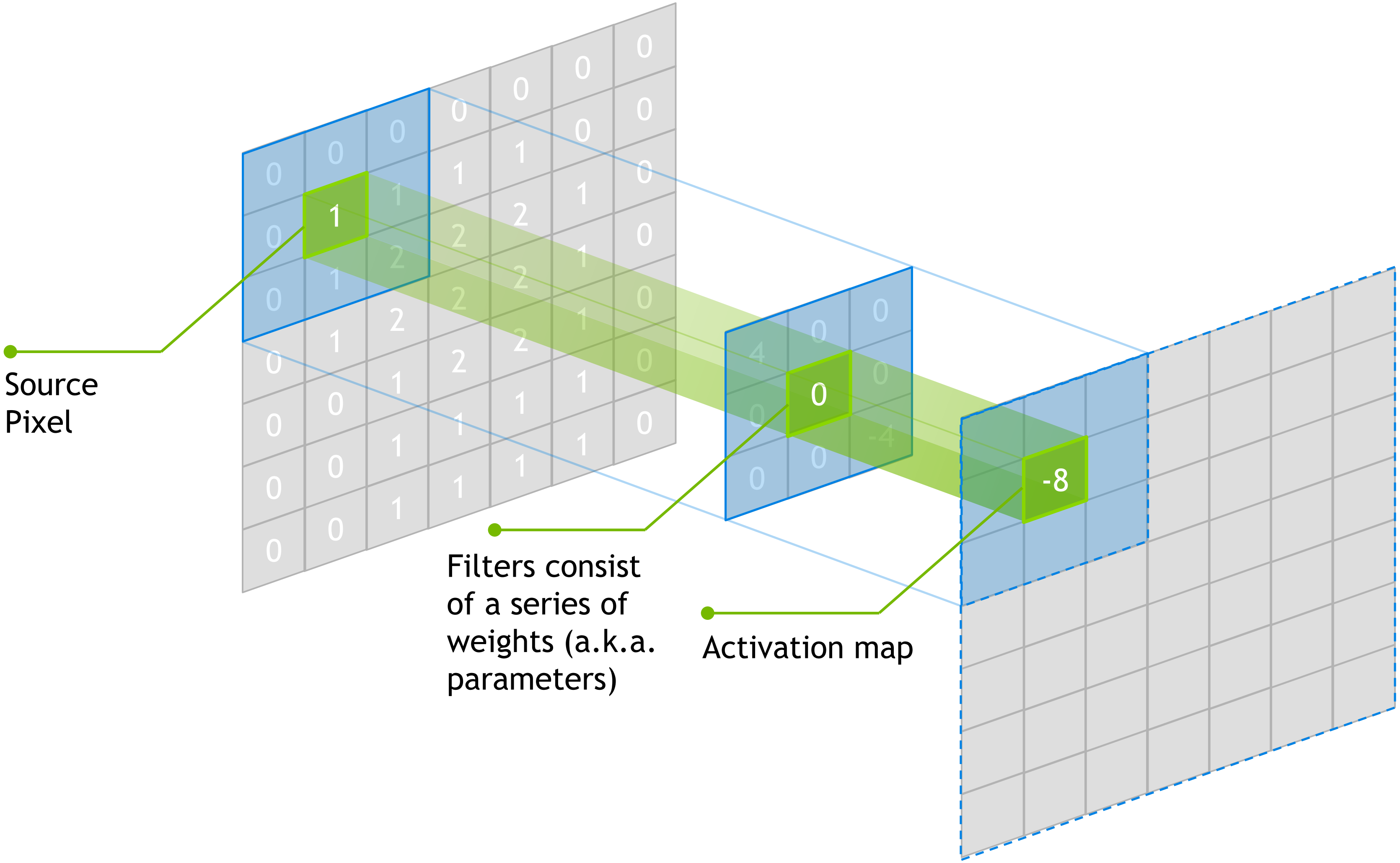
- MNIST: 0.7% error rate

CONVOLUTION

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.



CNN TERMINOLOGY



ADDITIONAL TERMINOLOGY

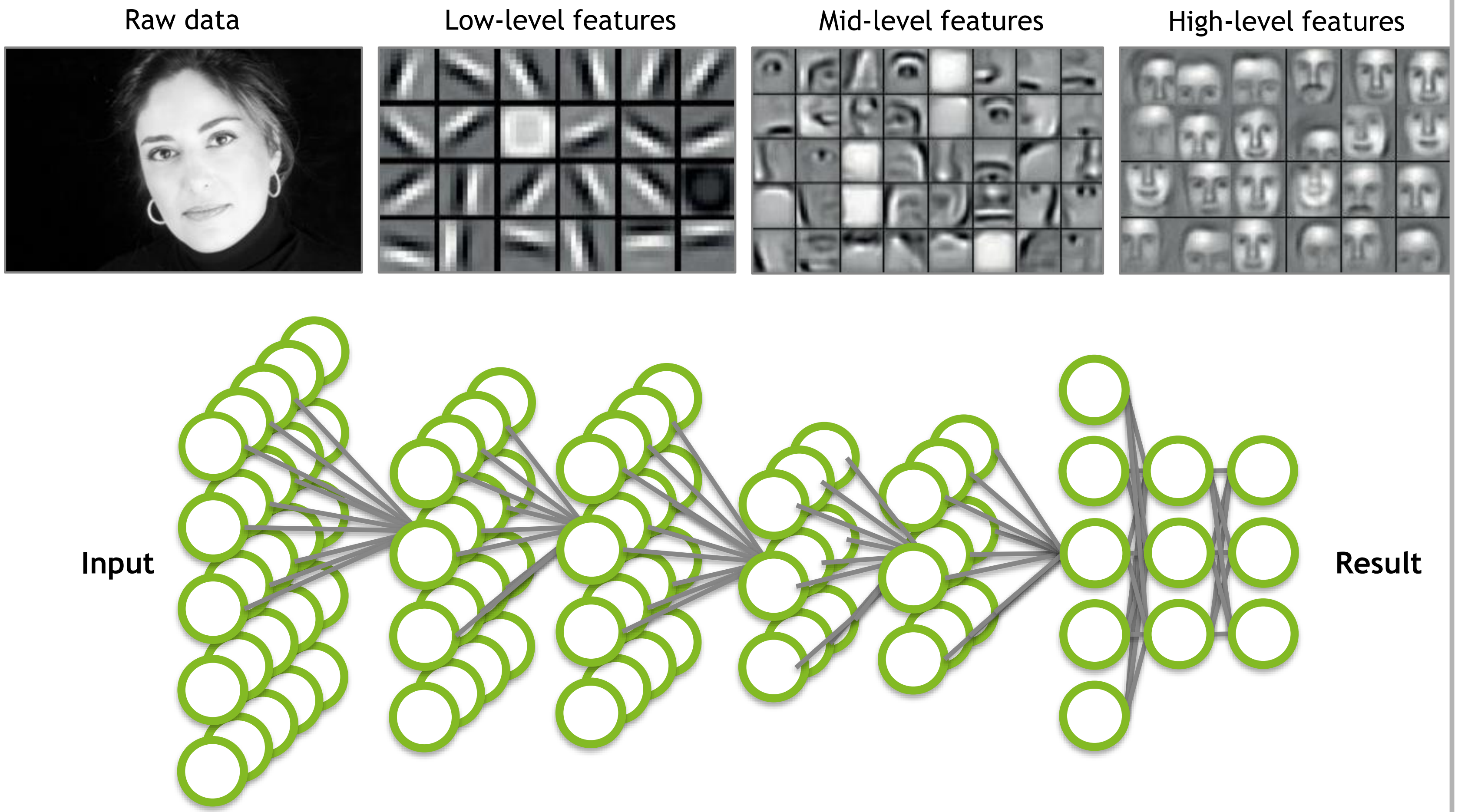
- Hyperparameters – parameters specified before training begins
  - Can influence the speed in which learning takes place
  - Can impact the accuracy of the model
  - Examples: Learning rate, decay rate, batch size

- Epoch – complete pass through the training dataset

- Activation functions – identifies active neurons
  - Examples: Sigmoid, Tanh, ReLU

- Pooling – Down-sampling technique
  - No parameters (weights) in pooling layer

DEEP NEURAL NETWORK (DNN)



Application components:

Task objective
e.g. Identify face

Training data
10-100M images

Network architecture
~10s-100s of layers
1B parameters

Learning algorithm
~30 Exaflops
1-30 GPU days

NVIDIA’S DIGITS

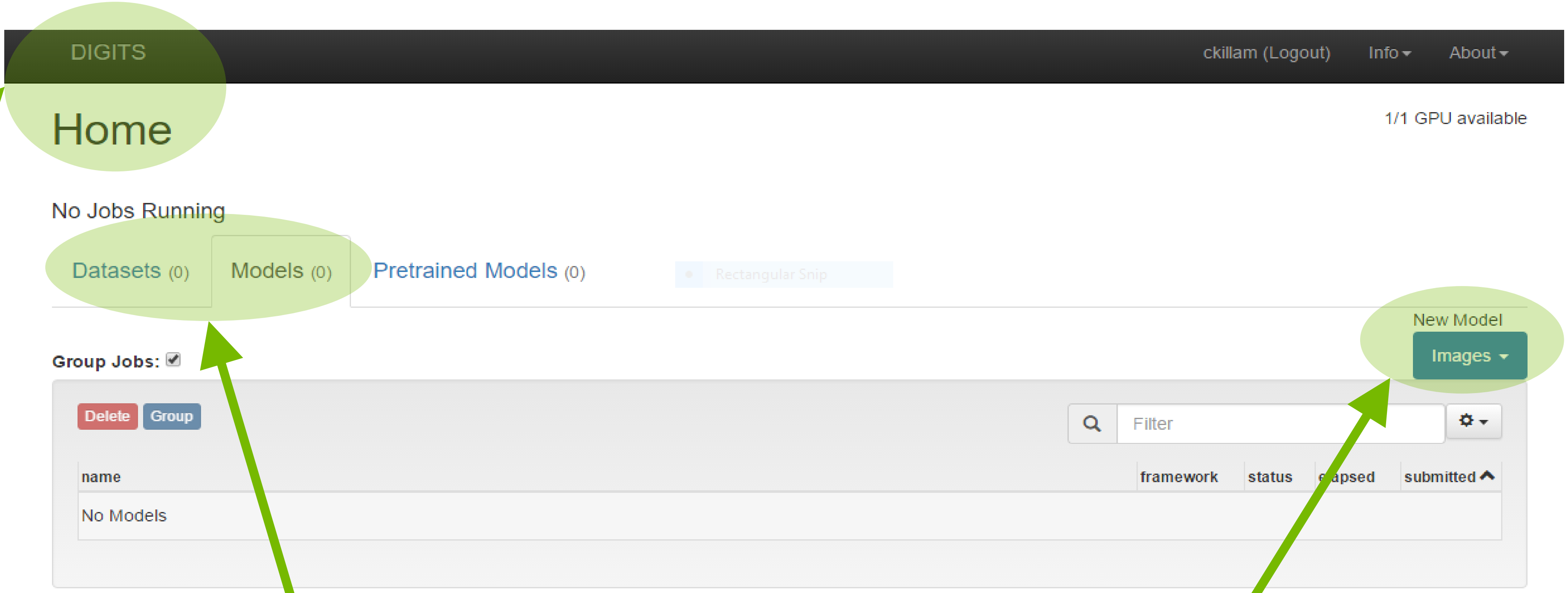
NVIDIA'S DIGITS

Interactive Deep Learning GPU Training System

- Simplifies common deep learning tasks such as:
  - Managing data
  - Designing and training neural networks on multi-GPU systems
  - Monitoring performance in real time with advanced visualizations
- Completely interactive so data scientists can focus on designing and training networks rather than programming and debugging
- Open source

DIGITS - HOME

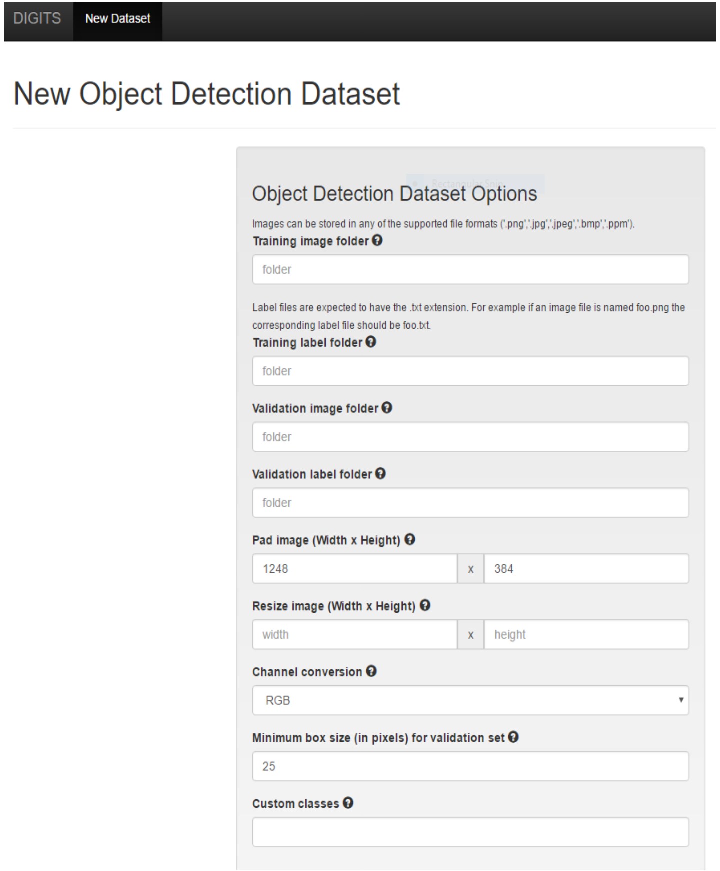
Clicking DIGITS will bring you to this Home screen

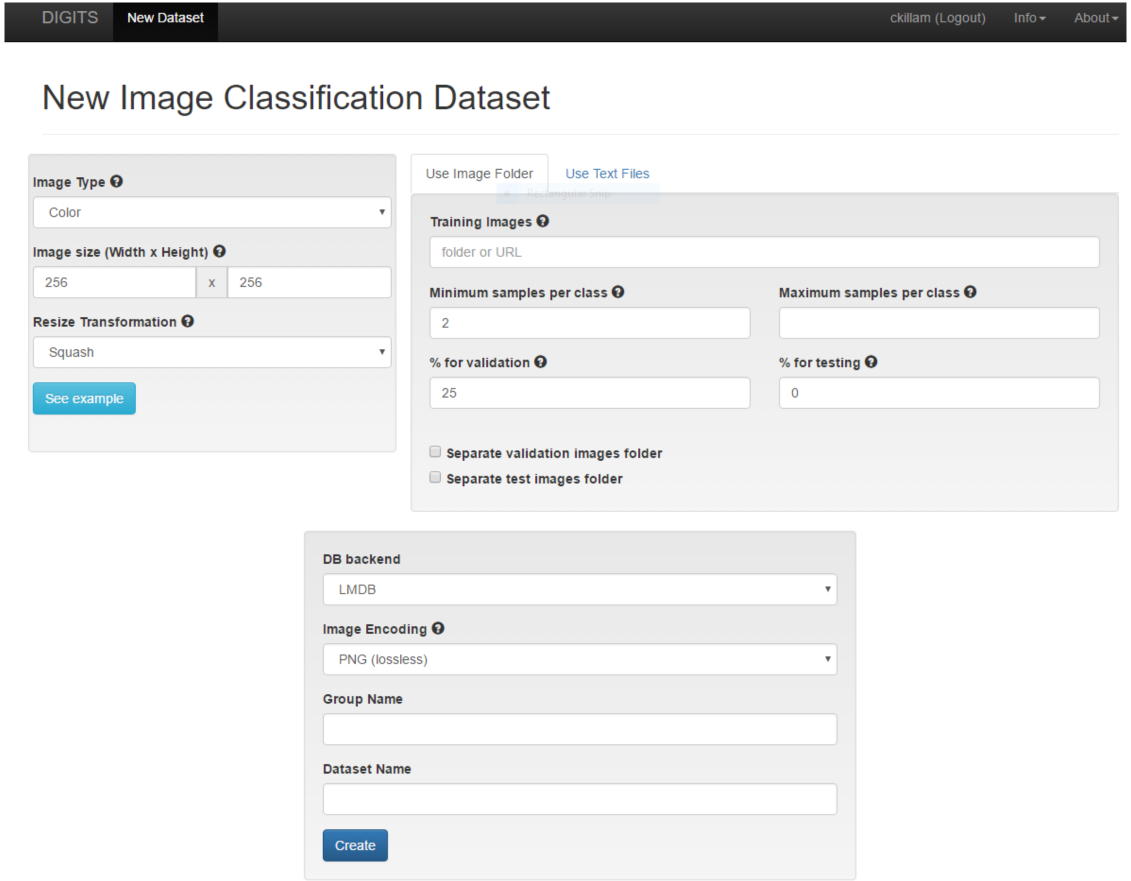


Click here to see a list of existing datasets or models

Clicking here will present different options for model and dataset creation

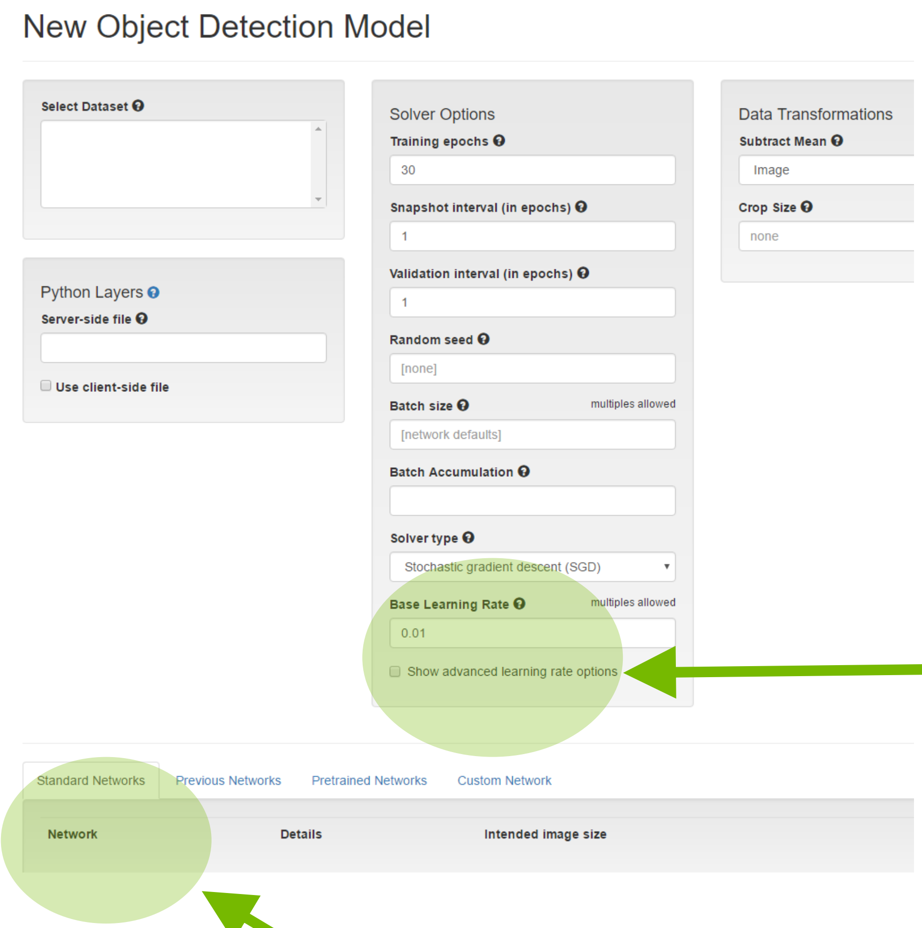
DIGITS - DATASET





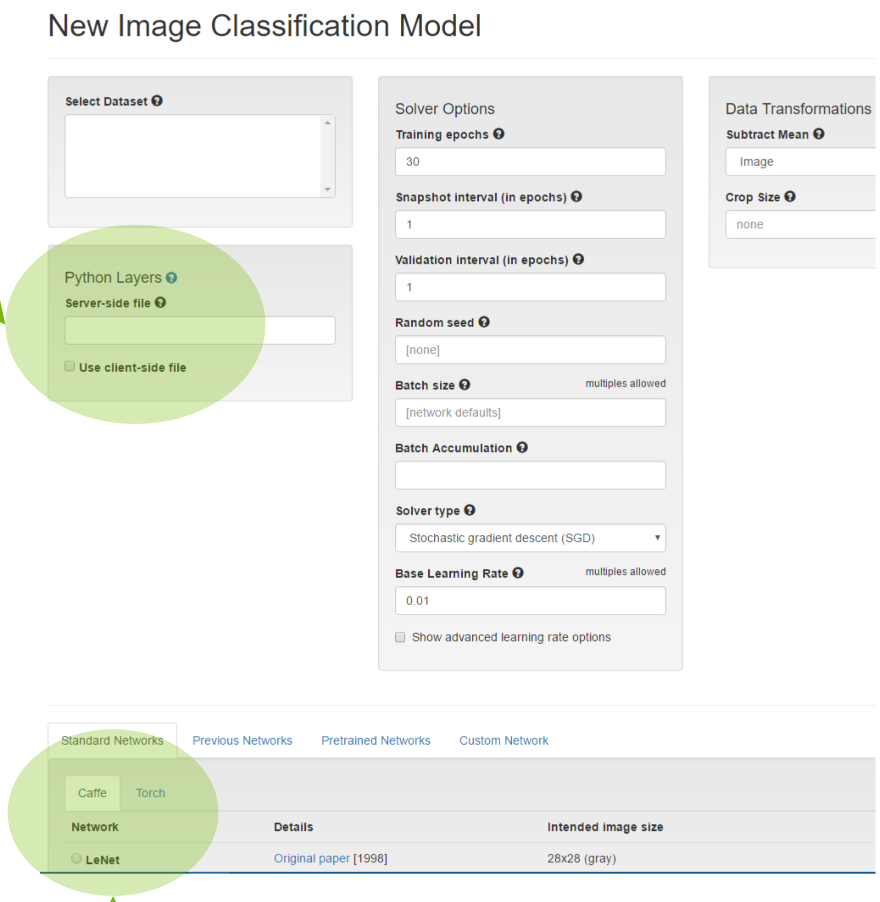
Different options will be presented based upon the task

DIGITS - MODEL



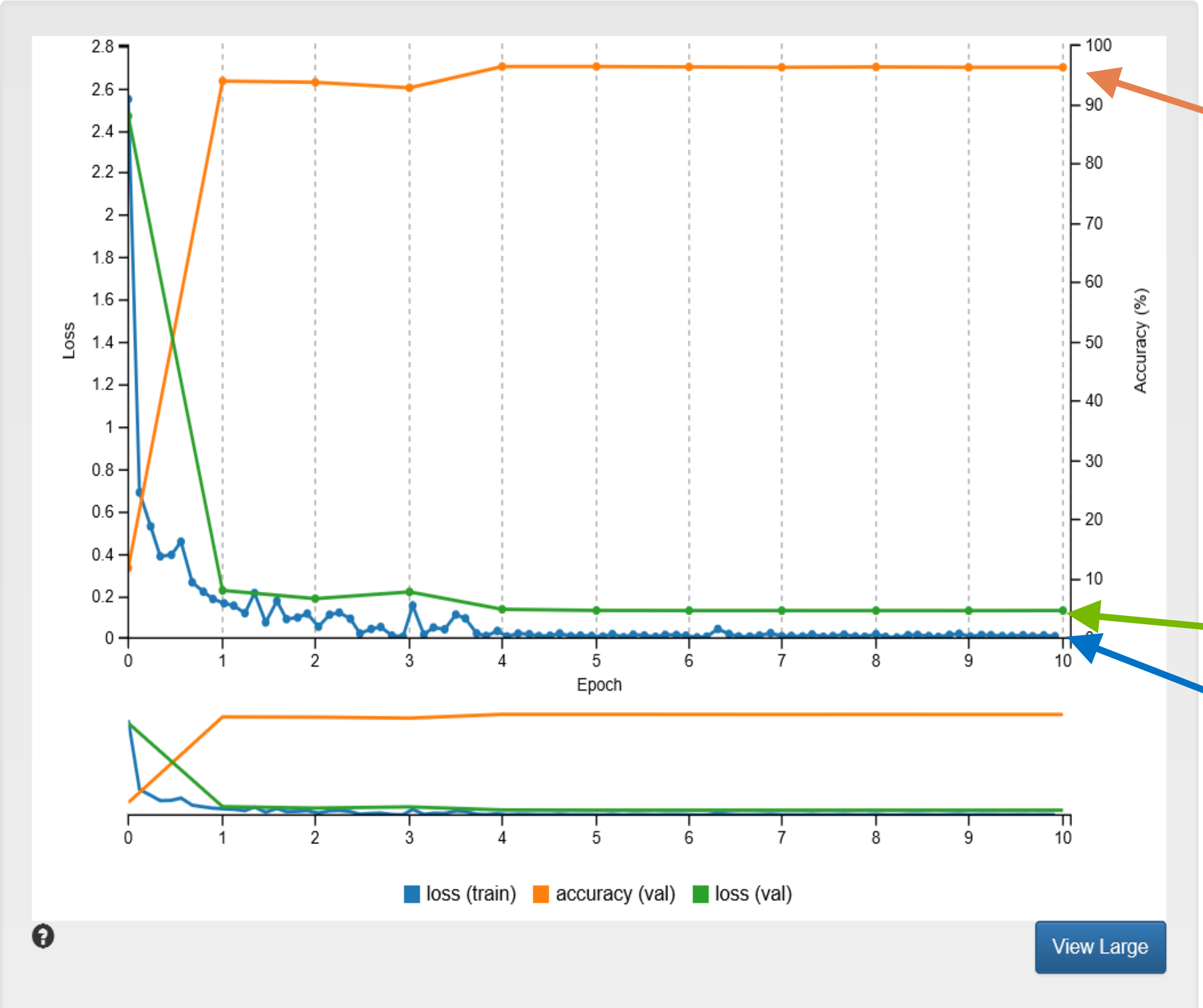
Define custom layers with Python

Can anneal the learning rate



Differences may exist between model tasks

EVALUATE THE MODEL



Accuracy obtained from validation dataset

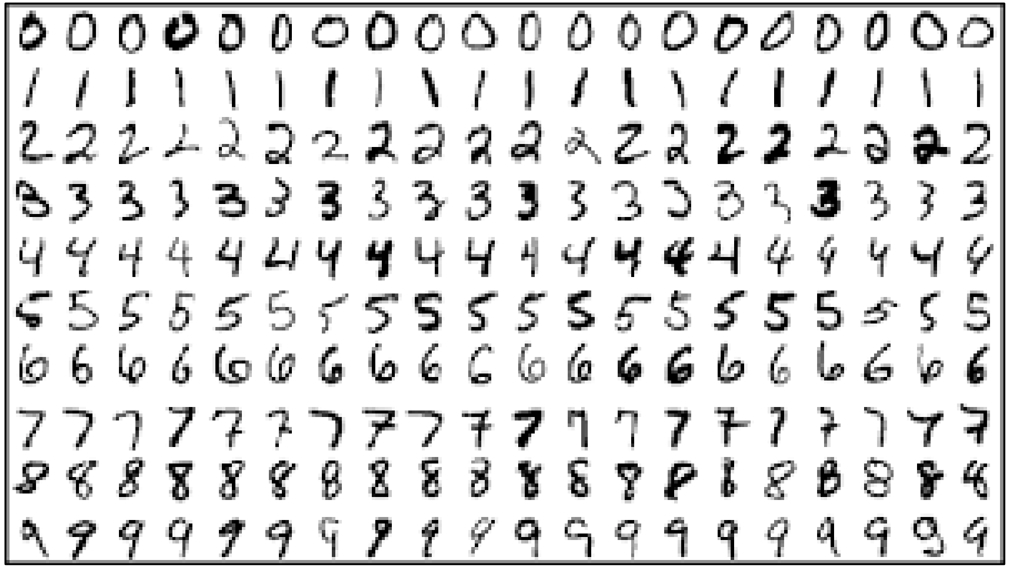
Loss function (Validation)

Loss function (Training)

HANDWRITTEN DIGIT RECOGNITION

HELLO WORLD of machine learning?

- MNIST data set of handwritten digits from Yann Lecun’s website
- All images are 28x28 grayscale
  - Pixel values from 0 to 255
- 60K training examples / 10K test examples
- Input vector of size 784
  - 28 * 28 = 784
- Output value is integer from 0-9



ADDITIONAL TECHNIQUES TO IMPROVE MODEL

- More training data
- Data augmentation
- Modify the network

ADDITIONAL TERMINOLOGY

- Hyperparameters – parameters specified before training begins
  - Can influence the speed in which learning takes place
  - Can impact the accuracy of the model
  - Examples: Learning rate, decay rate, batch size

- Epoch – complete pass through the training dataset

- Activation functions – identifies active neurons
  - Examples: Sigmoid, Tanh, ReLU

- Pooling – Down-sampling technique
  - No parameters (weights) in pooling layer

FIRST RESULTS

Small dataset (10 epochs)

- 96% of accuracy achieved
- Training is done within one minute

	SMALL DATASET
	1 : 99.90 %
	2 : 69.03 %
	8 : 71.37 %
	8 : 85.07 %
	0 : 99.00 %
	8 : 99.69 %
	8 : 54.75 %

SECOND RESULTS

Full dataset (10 epochs)

- 99% of accuracy achieved
- No improvements in recognizing real-world images

	SMALL DATASET	FULL DATASET
	1 : 99.90 %	0 : 93.11 %
	2 : 69.03 %	2 : 87.23 %
	8 : 71.37 %	8 : 71.60 %
	8 : 85.07 %	8 : 79.72 %
	0 : 99.00 %	0 : 95.82 %
	8 : 99.69 %	8 : 100.0 %
	8 : 54.75 %	2 : 70.57 %

DATA AUGMENTATION

Adding Inverted Images

- Pixel(Inverted) = 255 – Pixel(original)
- White letter with black background
  - Black letter with white background
- Training Images: /home/ubuntu/data/train_invert
- Test Image: /home/ubuntu/data/test_invert
- Dataset Name: MNIST invert

DATA AUGMENTATION

Adding inverted images (10 epochs)

	SMALL DATASET	FULL DATASET	+INVERTED
	1 : 99.90 %	0 : 93.11 %	1 : 90.84 %
	2 : 69.03 %	2 : 87.23 %	2 : 89.44 %
	8 : 71.37 %	8 : 71.60 %	3 : 100.0 %
	8 : 85.07 %	8 : 79.72 %	4 : 100.0 %
	0 : 99.00 %	0 : 95.82 %	7 : 82.84 %
	8 : 99.69 %	8 : 100.0 %	8 : 100.0 %
	8 : 54.75 %	2 : 70.57 %	2 : 96.27 %

MODIFY THE NETWORK

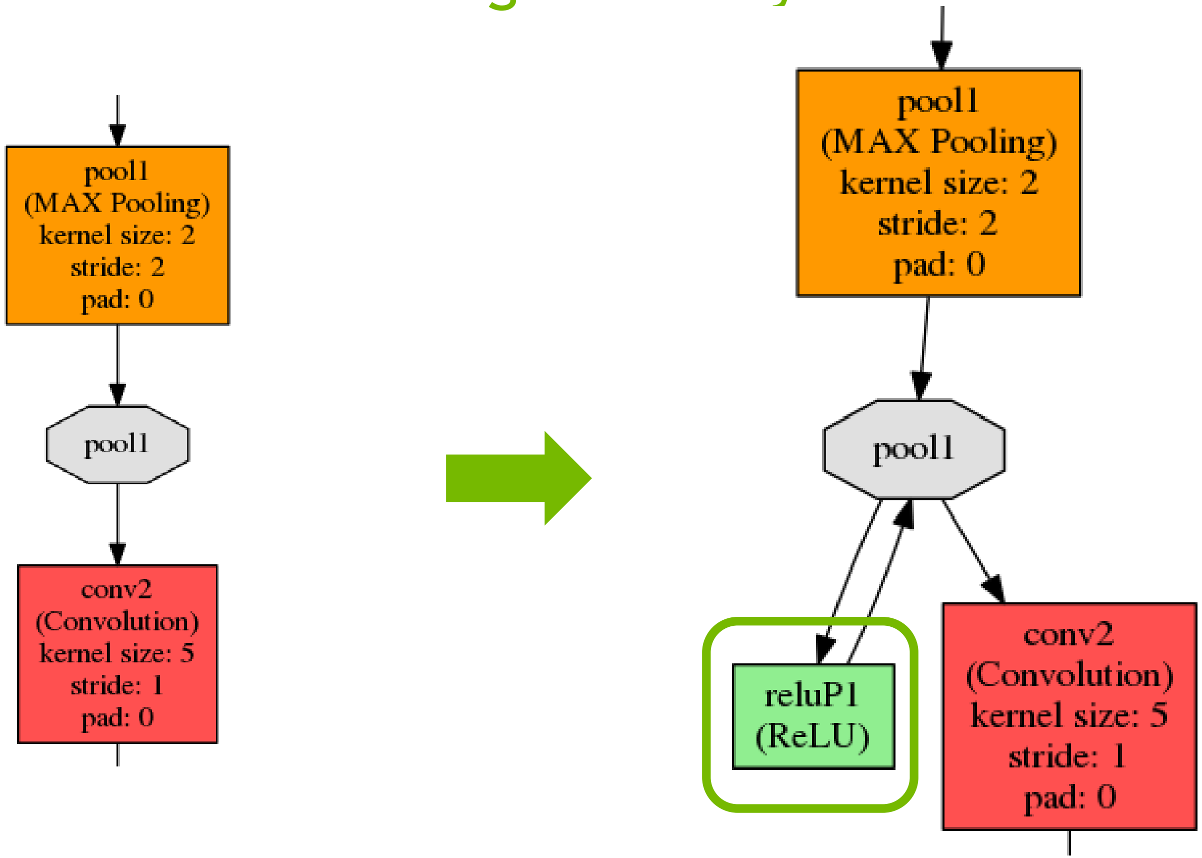
Adding filters and ReLU layer

```
layer {
        name: "pool1“
        type: "Pooling“
        …
}

layer {
        name: "reluP1"
        type: "ReLU"
        bottom: "pool1"
        top: "pool1"
}

layer {
        name: "reluP1“
```

```
layer {
  name: "conv1"
  type: "Convolution"
        ...
        convolution_param {
        num_output: 75
        ...
layer {
        name: "conv2"
        type: "Convolution"
        ...
        convolution_param {
        num_output: 100
        ...
```

MODIFY THE NETWORK

Adding ReLU Layer



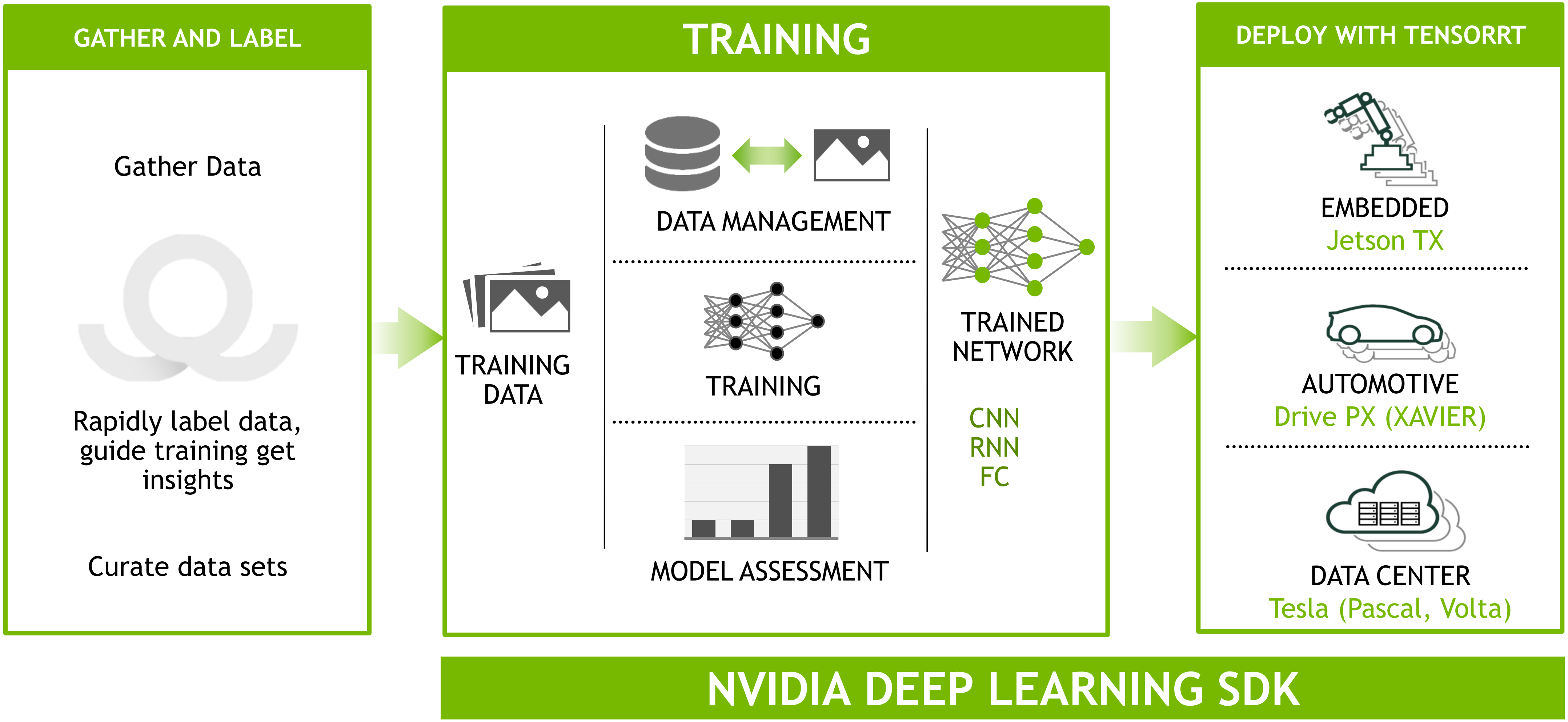
MODIFIED NETWORK

Adding filters and ReLU layer (10 epochs)

	SMALL DATASET	FULL DATASET	+INVERTED	ADDING LAYER
	1 : 99.90 %	0 : 93.11 %	1 : 90.84 %	1 : 59.18 %
	2 : 69.03 %	2 : 87.23 %	2 : 89.44 %	2 : 93.39 %
	8 : 71.37 %	8 : 71.60 %	3 : 100.0 %	3 : 100.0 %
	8 : 85.07 %	8 : 79.72 %	4 : 100.0 %	4 : 100.0 %
	0 : 99.00 %	0 : 95.82 %	7 : 82.84 %	2 : 62.52 %
	8 : 99.69 %	8 : 100.0 %	8 : 100.0 %	8 : 100.0 %
	8 : 54.75 %	2 : 70.57 %	2 : 96.27 %	8 : 70.83 %

DEEP LEARNING SDK

NVIDIA DEEP LEARNING SOFTWARE PLATFORM



AI INFERENCING IS EXPLODING

2 Trillion
Messages Per Day On LinkedIn

PERSONALIZATION

500M
Daily active users of iFlyTek

SPEECH

140 Billion
Words Per Day Translated by Google

TRANSLATION

60 Billion
Video frames/day uploaded on Youtube

VIDEO

NVIDIA TensorRT

Deep Learning Inference Optimizer and Runtime

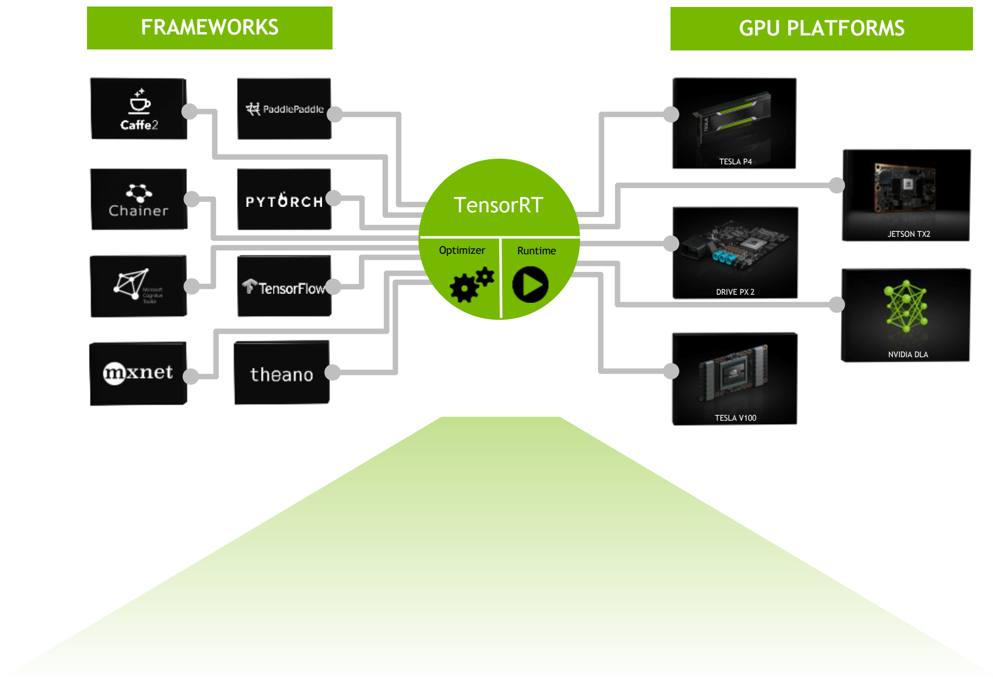
High performance neural network inference optimizer and runtime engine for production deployment

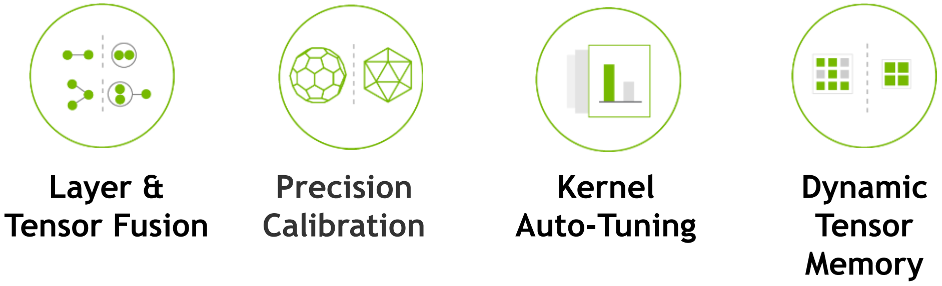
Maximize inference throughput for latency-critical services in hyperscale datacenters, embedded, and automotive production environments

Optimize TensorFlow and ONNX-framework models to generate high-performance runtime engines

Deploy faster, more responsive and memory efficient deep learning applications with INT8 and FP16 optimized precision support

developer.nvidia.com/tensorrt

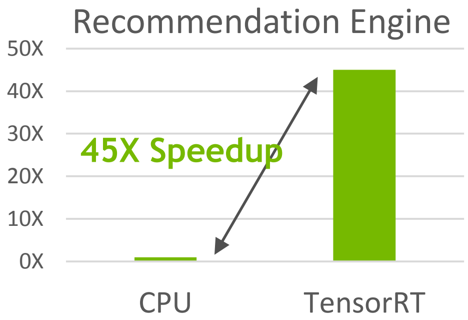




NVIDIA TENSORRT 4

RNN and MLP Layers • ONNX Import • NVIDIA DRIVE Support

Maximize RNN and MLP Throughput



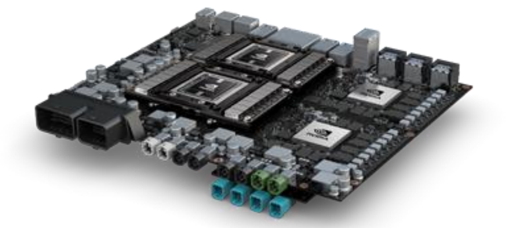
Speed up speech, audio and recommender app inference performance through new layers and optimizations

Optimize and Deploy ONNX Models

ONNX

Easily import and accelerate inference for ONNX frameworks (PyTorch, Caffe 2, CNTK, MxNet and Chainer)

Support for NVIDIA DRIVE Xavier



Deploy optimized deep learning inference models NVIDIA DRIVE Xavier

Free download to members of NVIDIA Developer Program

developer.nvidia.com/tensorrt

DL FRAMEWORKS

NVIDIA Optimized Examples

Over 32 examples with 19 new for Volta Tensor Cores

TensorFlow: FP32 & FP16

ResNet-50, Inception V3, Inception V4, GoogleNet, AlexNet

*Seq2seq (OpenNMT), *BigLSTM, DeepSpech2

MXNet: FP32 & FP16

ResNet-50, Inception V3, Inception V4, AlexNet

*Seq2seq, *word-rnn

Caffe2: FP32 & FP16

ResNet-50, Inception V3, Inception V4, AlexNet

*Seq2seq (OpenNMT), *char-rnn

PyTorch: FP32 & FP16

ResNet-50 and AlexNet

word-level

*Only FP32

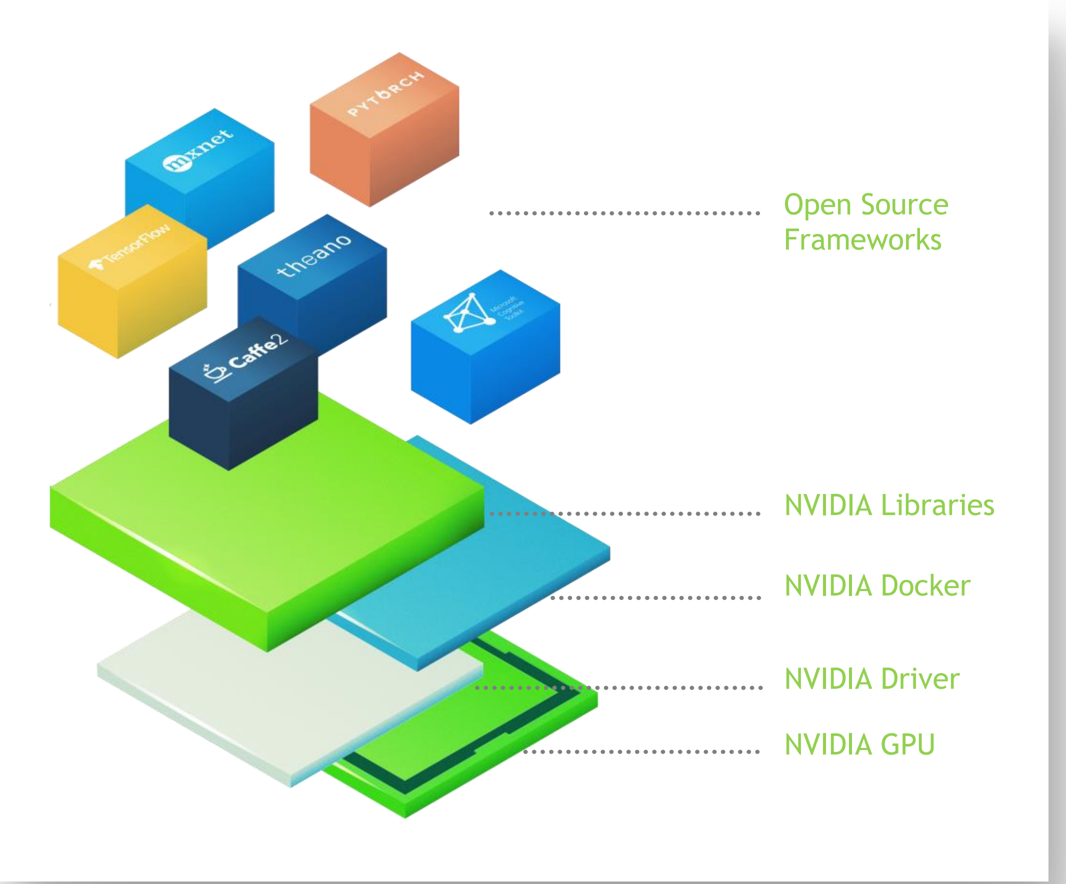
CONTAINER

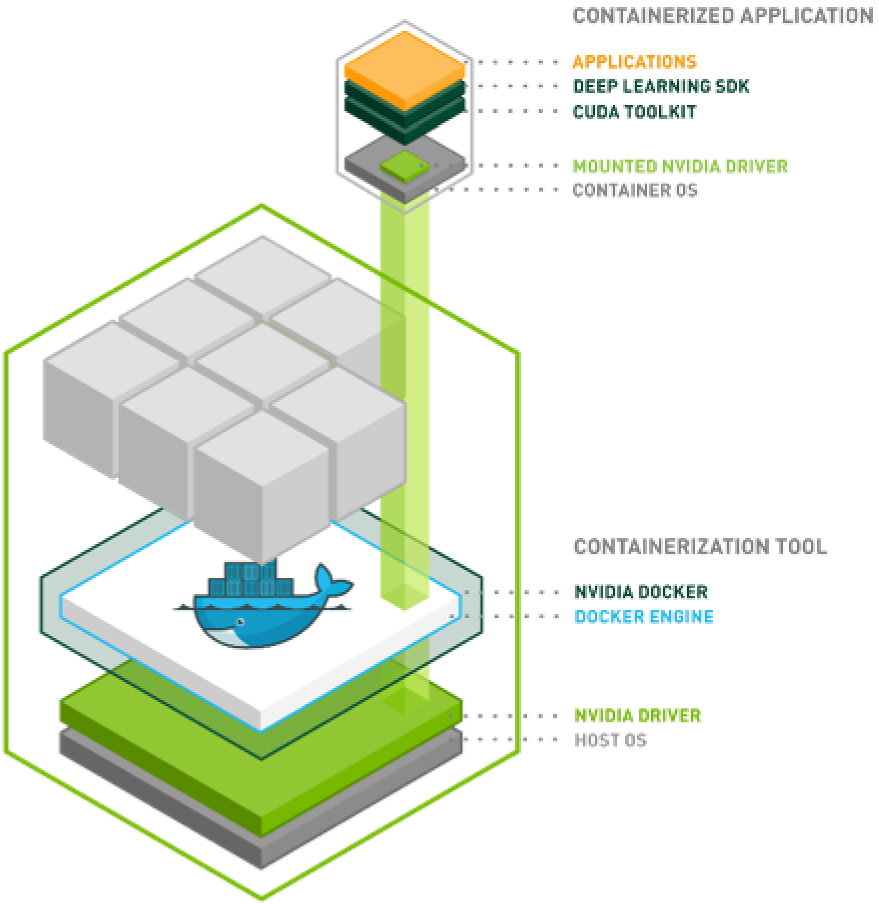
CHALLENGES

Current DIY deep learning environments are complex and time consuming to build, test and maintain

Development of frameworks by the community is moving very quickly

Requires high level of expertise to manage driver, library, framework dependencies





SIMPLIFY PORTABILITY WITH NVIDIA CONTAINERS

Benefits of Containers:

Simplify deployment of GPU-accelerated applications

Isolate individual frameworks or applications

Share, collaborate, and test applications across different environments

NVIDIA GPU CLOUD REGISTRY

Common Software stack across NVIDIA GPUs

Deep Learning

All major frameworks with multi-GPU optimizations Uses NCCL for NVLINK data exchange Multi-threaded I/O to feed the GPUs

Caffe, Caffe2,CNTK, mxnet, PyTorch, Tensorflow, Theano, Torch

HPC

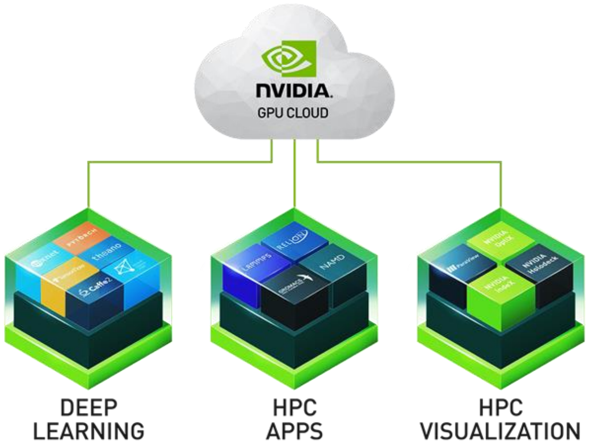
NAMD, Gromacs, LAMMPS, GAMESS, Relion, Chroma, MILC

HPC Visualization

Paraview with Optix, Index and Holodeck with OpenGL visualization base on NVIDIA Docker 2.0, IndeX, VMD

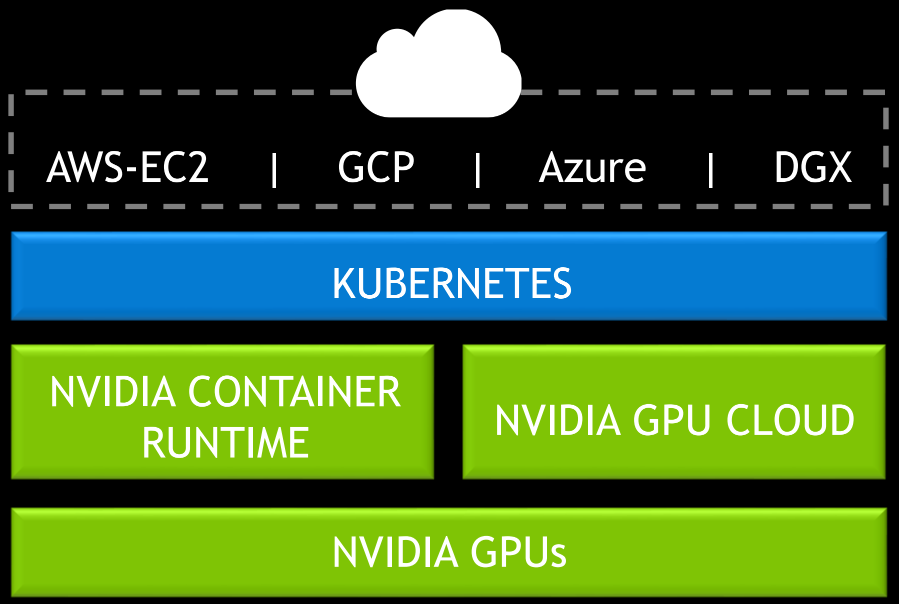
Single NGC Account

For use on GPUs everywhere - https://ngc.nvidia.com



NVIDIA GPU Cloud containerizes GPU-optimized frameworks, applications, runtimes, libraries, and operating system, available at no charge

Container Orchestration for DL Training & Inference



KUBERNETES on NVIDIA GPUs

- Scale-up Thousands of GPUs Instantly
- Self-healing Cluster Orchestration
- GPU Optimized Out-of-the-Box
- Powered by NVIDIA Container Runtime
- Included with Enterprise Support on DGX

CONVERGENCE OF HPC AND AI

INTELLIGENT HPC

DL Driving Future HPC Breakthroughs

- Trained networks as solvers
- Super-resolution of coarse simulations
- Low- and mixed-precision
- Simulation for training, network in production

From calendar time to real time?

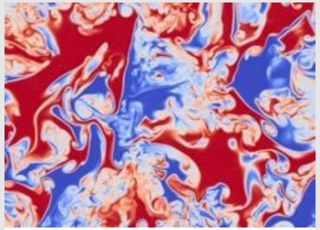
Pre-processing → Simulation → Post-processing

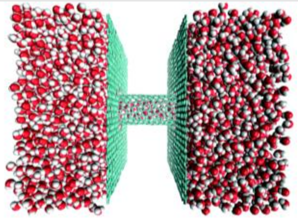
- Select/classify/augment/ distribute input data
- Control job parameters

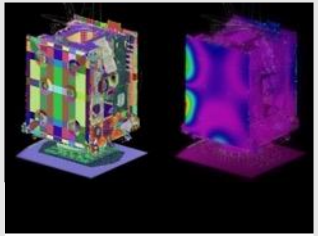
- Analyze/reduce/augment output data
- Act on output data

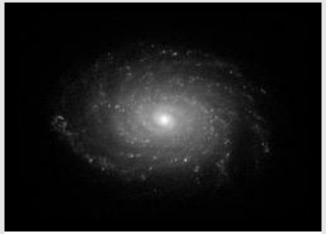
AI Supercomputing is The New computing model

Extending The Reach of HPC By Combining Computational & Data Science


Turbulent Flow

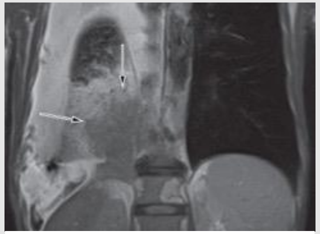

Molecular Dynamics


Structural Analysis


N-body Simulation

COMPUTATIONAL SCIENCE

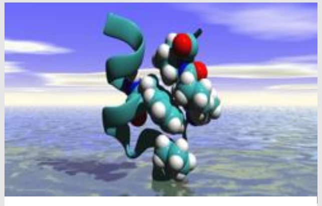

“What’s happening?”


“Is there cancer?”


“Next move?”


“What does she mean?”

DATA SCIENCE


Drug Discovery

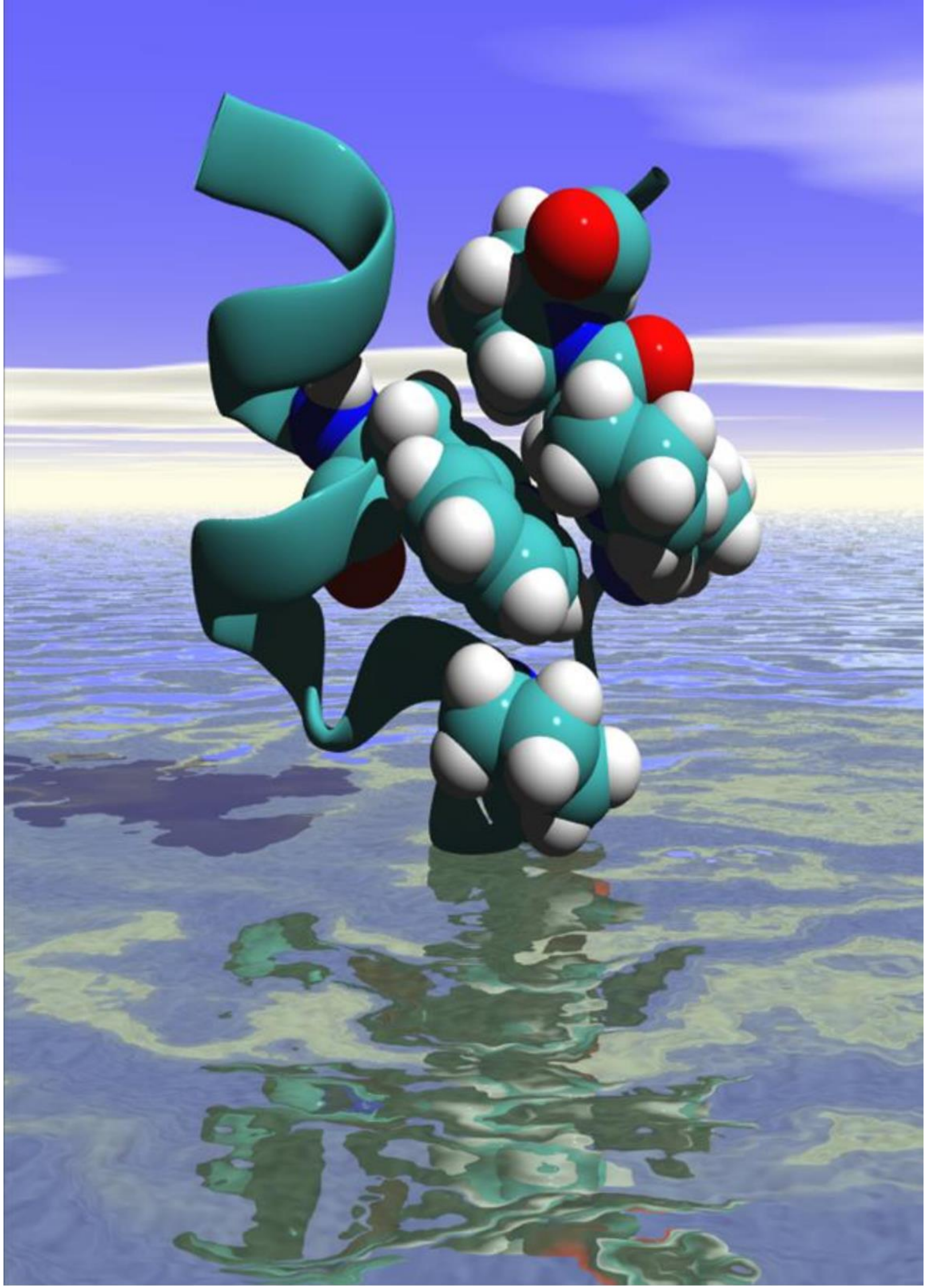

Clean Energy


Understanding Universe


Monitoring Climate Change

COMPUTATIONAL & DATA SCIENCE

S8242 – DL for Computational Science, Jeff Adie & Yang Juntao
Presented ~20 Success Stories of DL in Computational Science
(GTC on-demand: http://on-demand-gtc.gputechconf.com)



AI Quantum Breakthrough

Background

Developing a new drug costs $2.5B and takes 10-15 years. Quantum chemistry (QC) simulations are important to accurately screen millions of potential drugs to a few most promising drug candidates.

Challenge

QC simulation is computationally expensive so researchers use approximations, compromising on accuracy. To screen 10M drug candidates, it takes 5 years to compute on CPUs.

Solution

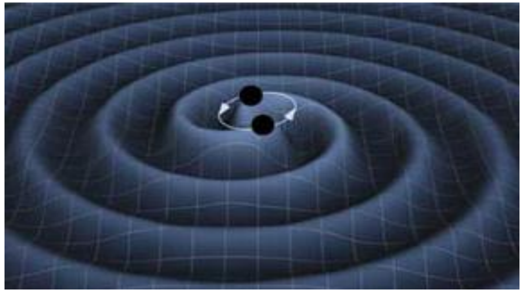
Researchers at the University of Florida and the University of North Carolina leveraged GPU deep learning to develop ANAKIN-ME, to reproduce molecular energy surfaces with super speed (microseconds versus several minutes), extremely high (DFT) accuracy, and at 1-10/millionths of the cost of current computational methods.
Essentially the DL model is trained to learn Hamiltonian of the Schrodinger equation.

Impact

Faster, more accurate screening at far lower cost

DEEP LEARNING FOR GRAVITATIONAL WAVE DETECTION

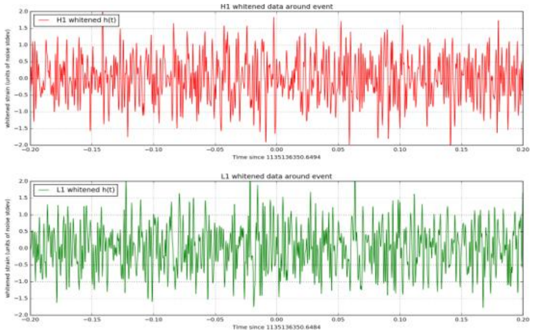
Deep learning method named deep filtering was used in the first detection of gravitational wave. Numerical simulated data was used for training deep filtering, a convolutional neural network to replace matched filtering. It provided 20X speed up on single core and potential to be accelerated further with GPU.



Gravitational wave due to black hole collide and merge

To be observed

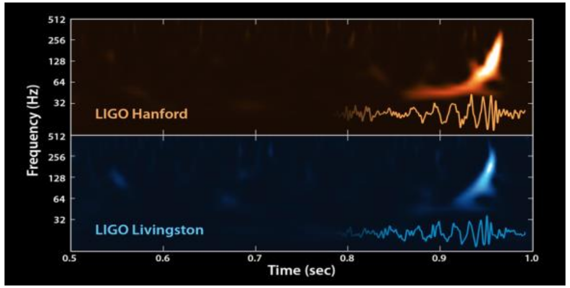
LIGO facility



Actual observed data

How to find
The signal???

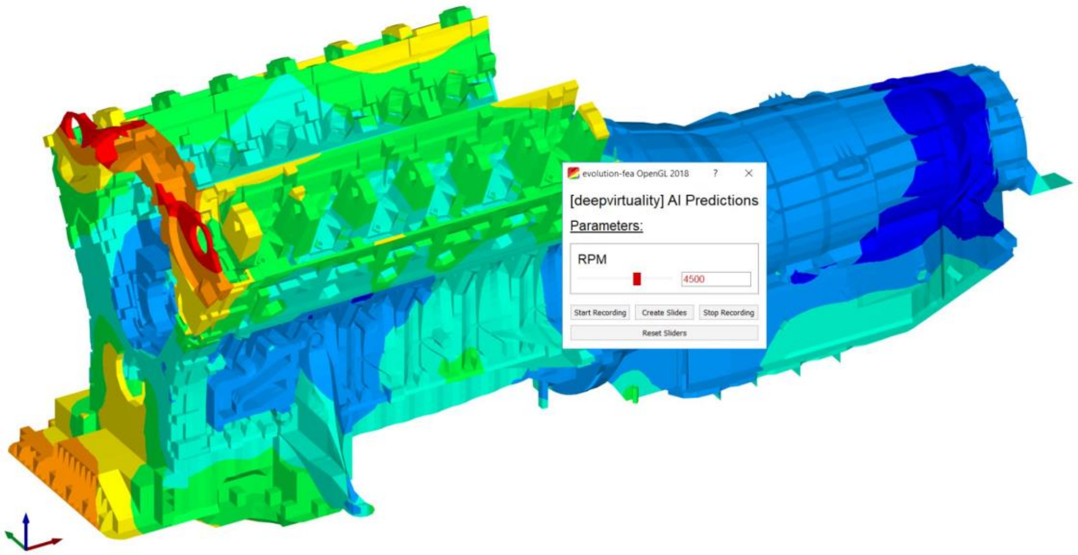
Deep Learning



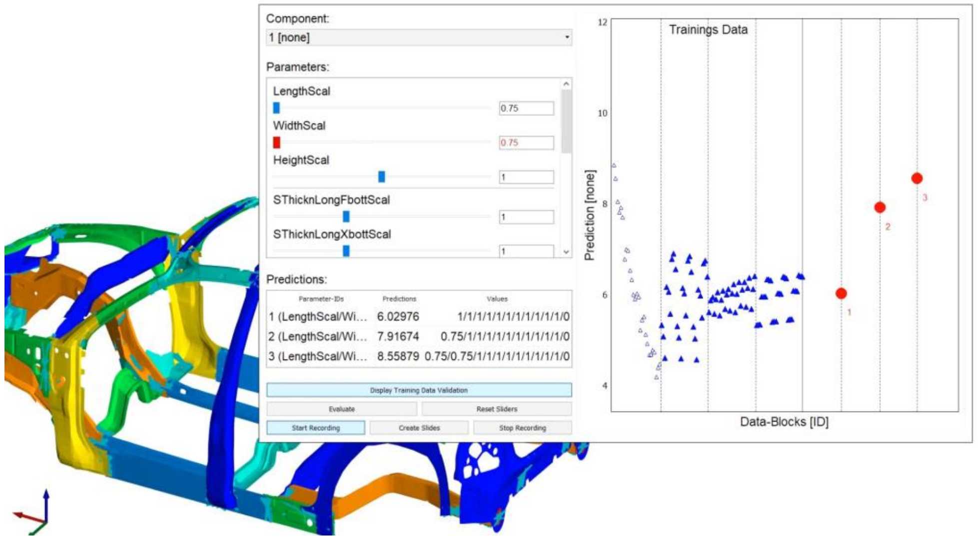
Actual Signal Caused by Gravitational Wave

FEA UPDATED WITH NEURAL NETWORK

FEA trained deep neural network for surrogate modelling of estimated stress distribution. Deepvirtuality, a spinoff from Volkswagen Data:Lab under Nvidia Inception Program has demonstrate with their software aimed for a quicker prediction of structural data.

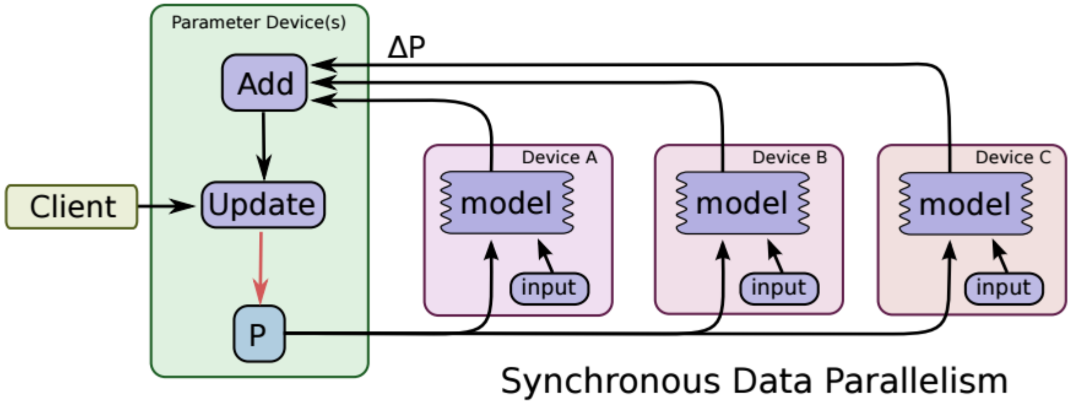


An demonstration of Structure Born Noise of a V12 Engine with Deepvirtuality



Torsional Frequencies of a Car Body by Deepvirtuallity

HOROVOD



https://github.com/uber/horovod, https://eng.uber.com/horovod/

“Horovod is a distributed training framework for TensorFlow. The goal of Horovod is to make distributed Deep Learning fast and easy to use.”

Leverage Tensorflow + MPI + NCCL2 for a simplified and performant API to enable synchronous multigpu + multinode Tensorflow.

Instead of Parameter Server architecture leverage MPI.

Support features such as RDMA, GPUDirectRDMA (GDR), via leveraging MPI and NCCL2.

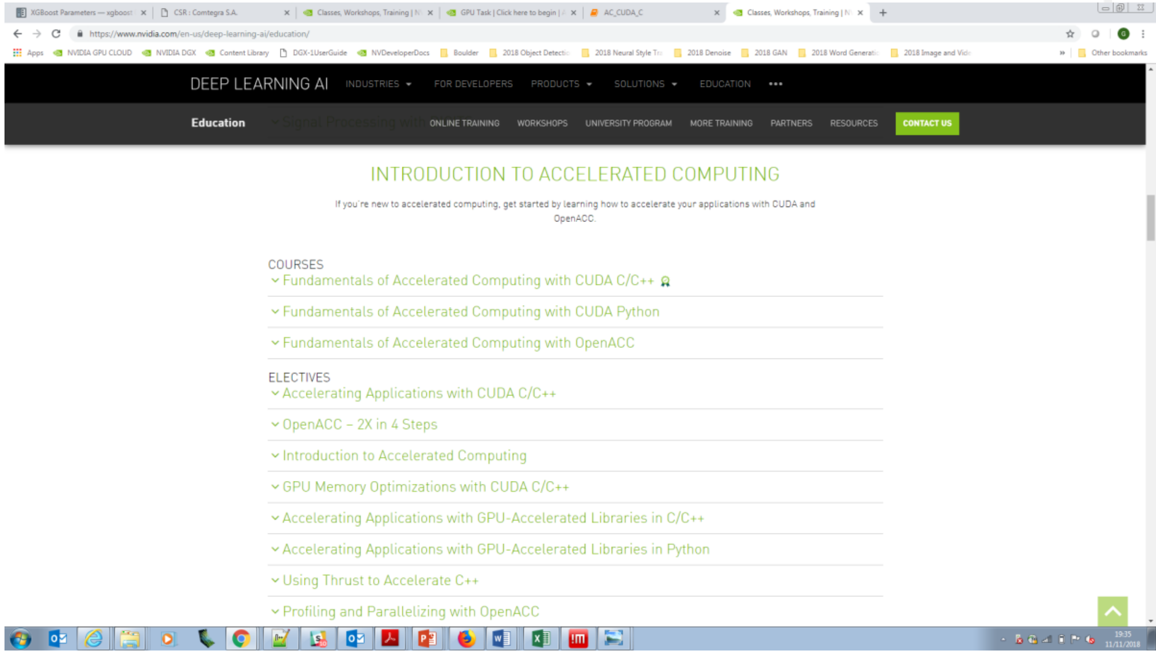
NAVIGATING TO COURSES

1. Navigate to: www.nvidia.co.uk/dlilabs

2. Google search for nvidia dli

3. Scroll down

- Use NV Developer login or new account.
- Image Classification with Digits



NVIDIA DEEP LEARNING INSTITUTE (DLI)

Hands-on training for developers, data scientists, and researchers

Online self-paced labs across beginner and intermediate levels available at
www.nvidia.com/dlilabs

Onsite workshops covering e.g. Deep Learning Fundamentals can be requested through our page www.nvidia.com/requestDLI





Gunter Roeth (gunterr@nvidia.com)

NVIDIA®