



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)

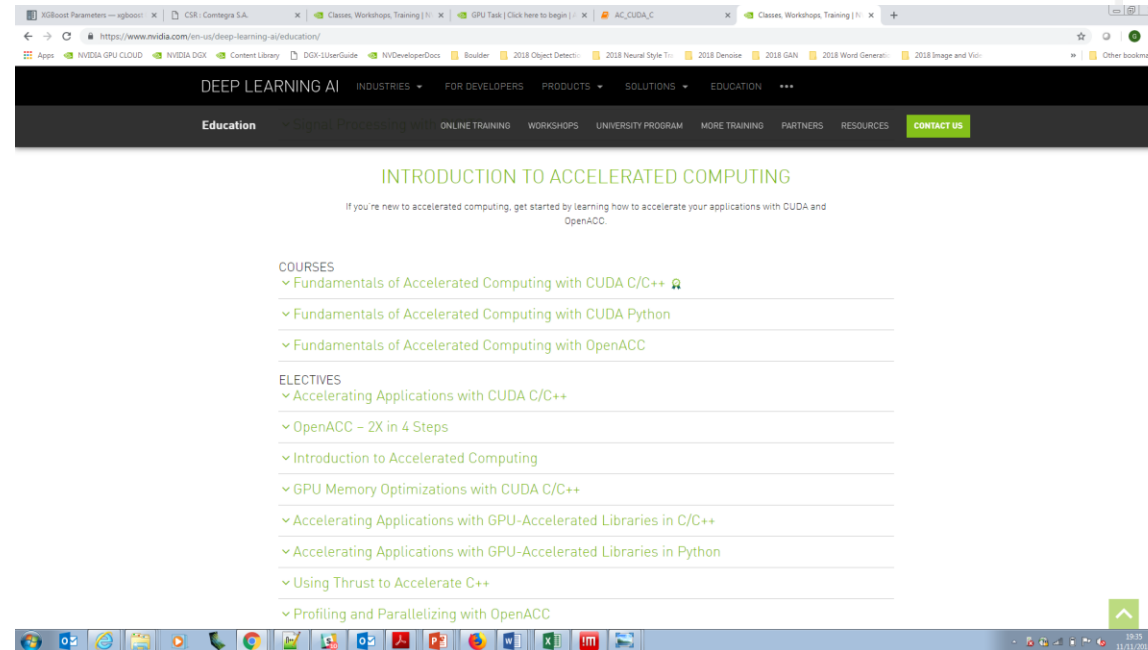
17h50-18h00 Wrap up and Q&A

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++



The background features a complex network of thin, light green lines connecting various nodes. The nodes are represented by small, glowing green circles of varying sizes and brightness. The overall aesthetic is futuristic and digital, set against a dark blue gradient background with some subtle bokeh effects.

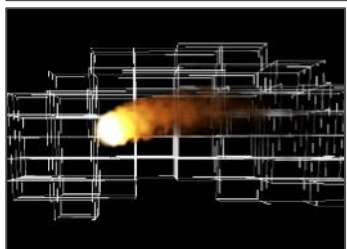
INTRODUCTION



NVIDIA

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

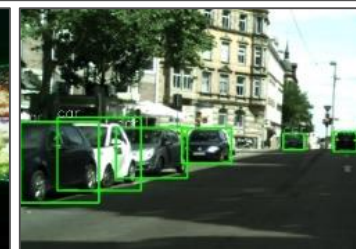
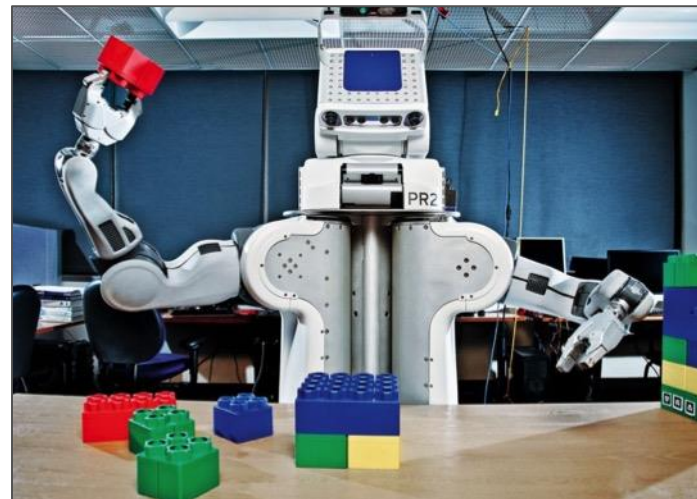




Computer Graphics



GPU Computing



Artificial Intelligence

ACCELERATED COMPUTING

Performance & Energy Efficiency

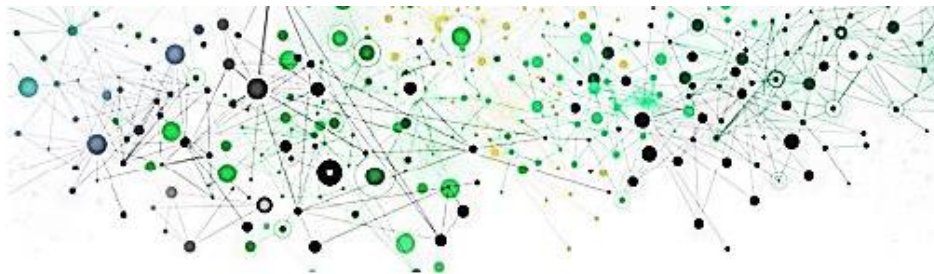
HIGH PERFORMANCE COMPUTE



AI / DEEP LEARNING



DATA ANALYTICS



ACCELERATED VDI



NVIDIA TESLA PLATFORM

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

CUSTOMER USECASES



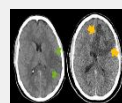
Speech



Translate



Recommender



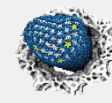
Healthcare



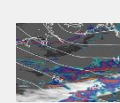
Manufacturing



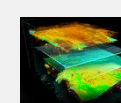
Engineering



Molecular Simulations



Weather Forecasting



Seismic Mapping

CONSUMER INTERNET

ENTERPRISE APPLICATIONS

SUPERCOMPUTING

INDUSTRY FRAMEWORKS & APPLICATIONS



Caffe2



Chainer



Microsoft Cognitive Toolkit



mxnet

Amber ANSYS

CHROMA



GROMACS
FAST. FLEXIBLE. FREE.



PaddlePaddle



PYTORCH



TensorFlow

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



SYSTEM OEM



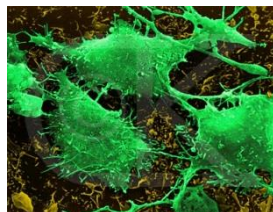
CLOUD

CONTINUED DEMAND FOR COMPUTE POWER

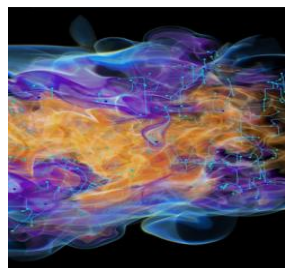
Ever-increasing compute power
Demand in HPC



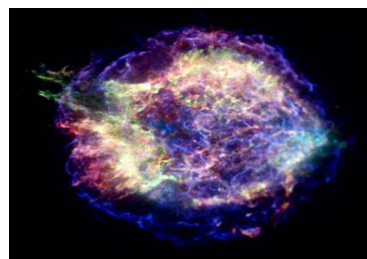
Comprehensive
Earth System
Model



Coupled simulation
of entire cells



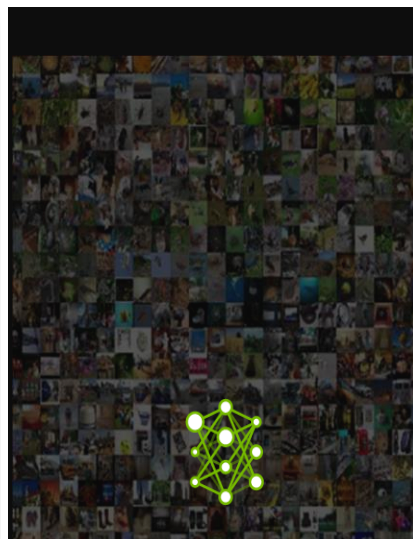
Simulation of
combustion for new
high-efficiency, low-
emission engines.



Predictive
calculations for
supernovae

Neural Network complexity is Exploding

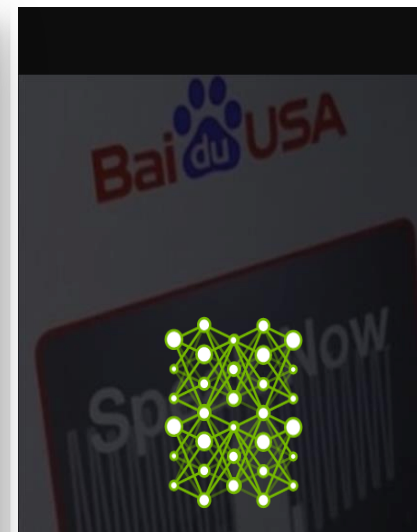
7 ExaFLOPS
60 Million Parameters



2015

Microsoft ResNet
Superhuman Image
Recognition

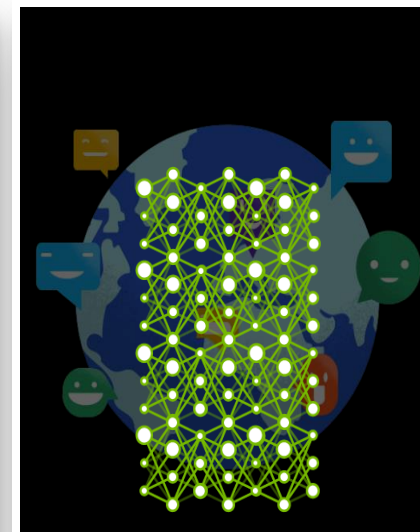
20 ExaFLOPS
300 Million Parameters



2016

Baidu Deep Speech 2
Superhuman Voice
Recognition

100 ExaFLOPS
8700 Million Parameters

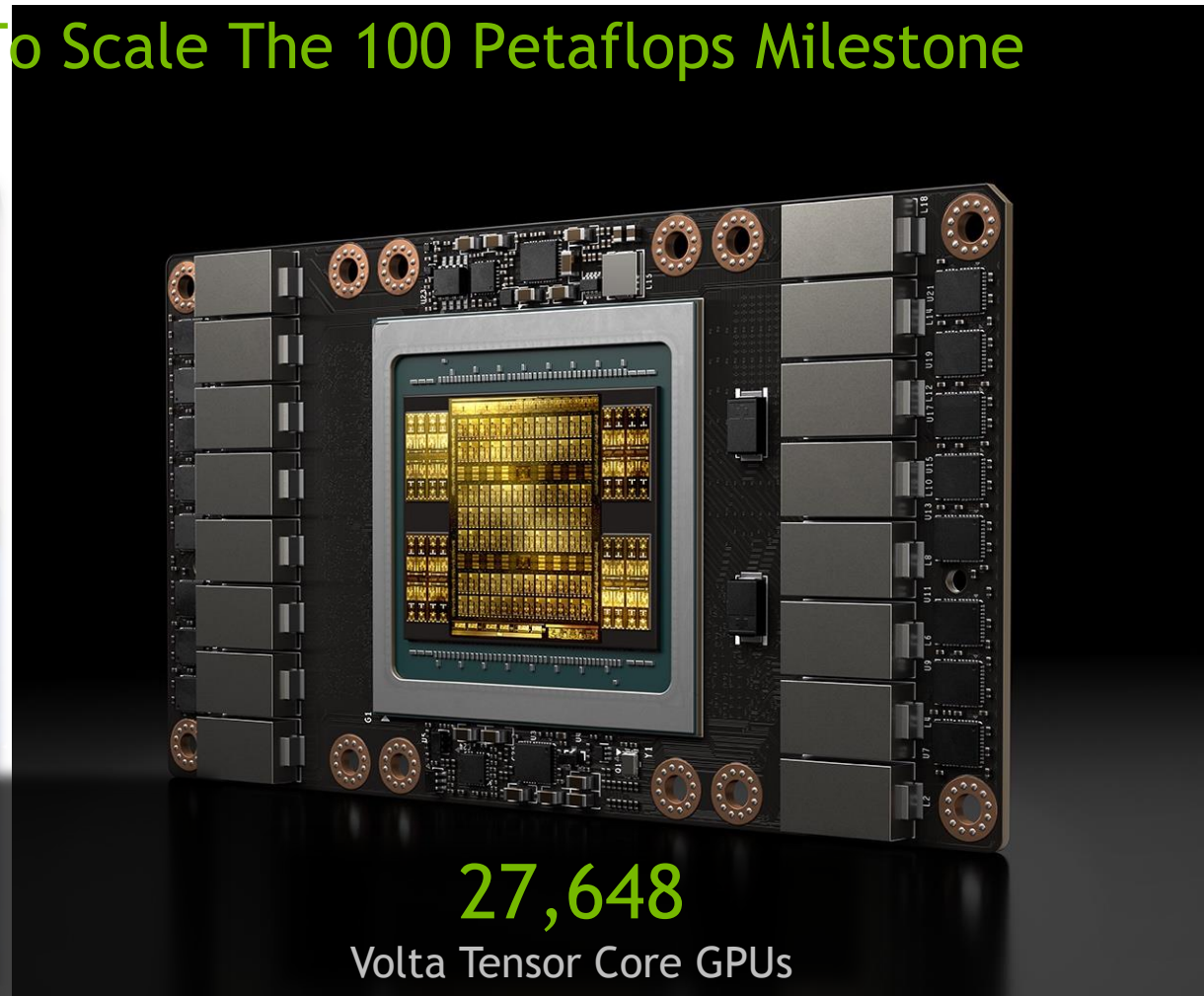


2017

Google Neural
Machine Translation
Near Human
Language Translation

NVIDIA POWERS WORLD'S FASTEST SUPERCOMPUTER

Summit Becomes First System To Scale The 100 Petaflops Milestone



GPUS FOR HPC AND DEEP LEARNING

Huge demand on compute power (FLOPS)

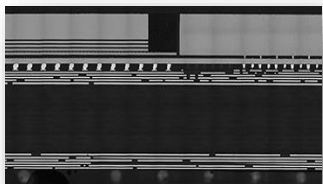
NVIDIA Tesla V100



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

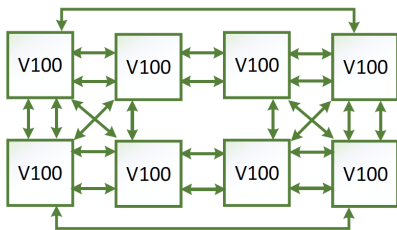
Huge demand on communication and memory bandwidth

CoWoS with HBM2



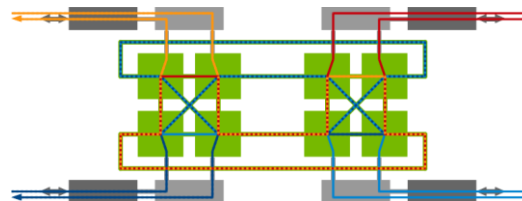
900 GB/s Memory Bandwidth
Unifying Compute & Memory
in Single Package

NVLink



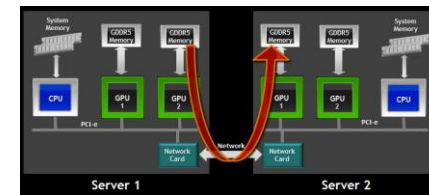
6 links per GPU a 50 GB/s bi-directional for maximum scalability between GPU's

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

The background features a complex network of thin, light green lines connecting various nodes. The nodes are represented by small, glowing green circles of varying sizes and brightness. The overall aesthetic is futuristic and technical, suggesting a digital or networked environment.

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

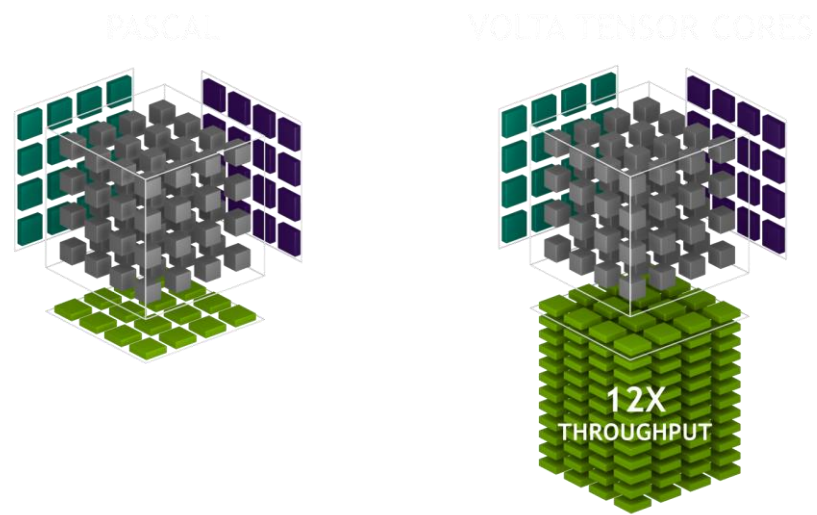
New CUDA TensorOp instructions & data formats

4x4x4 matrix processing array

$$D[\text{FP32}] = A[\text{FP16}] * B[\text{FP16}] + C[\text{FP32}]$$

Using Tensor cores via

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



$$D = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

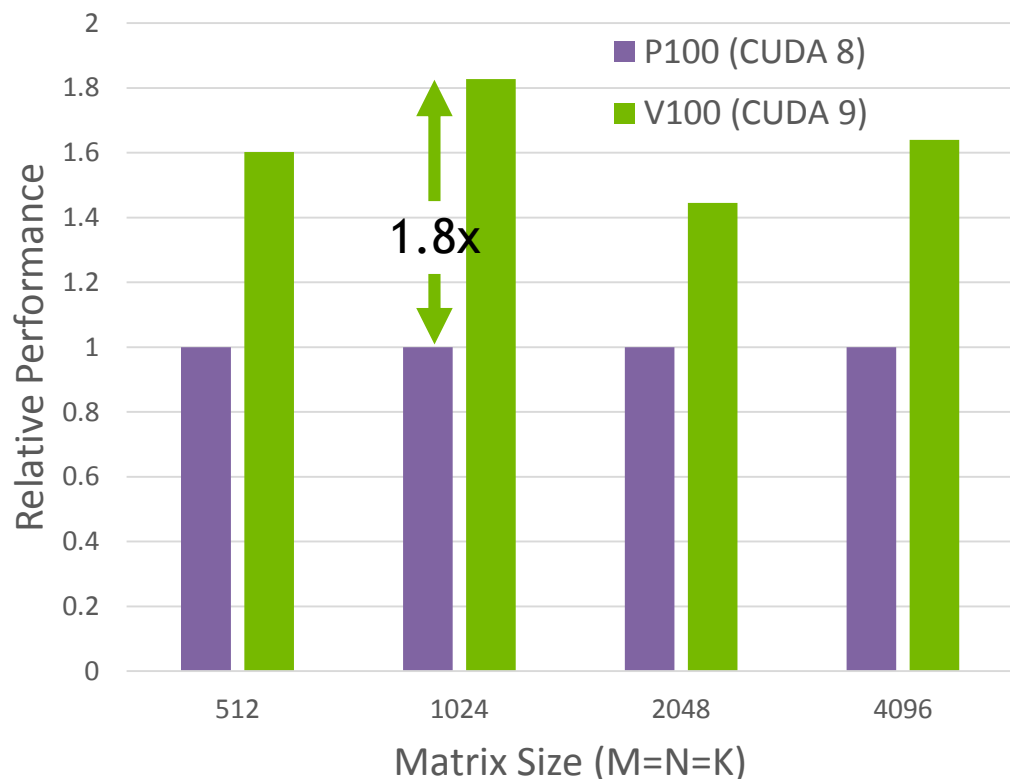
FP16 or FP32 FP16 FP16 FP16 or FP32

■ Activation Inputs ■ Weights Inputs ■ Output Results

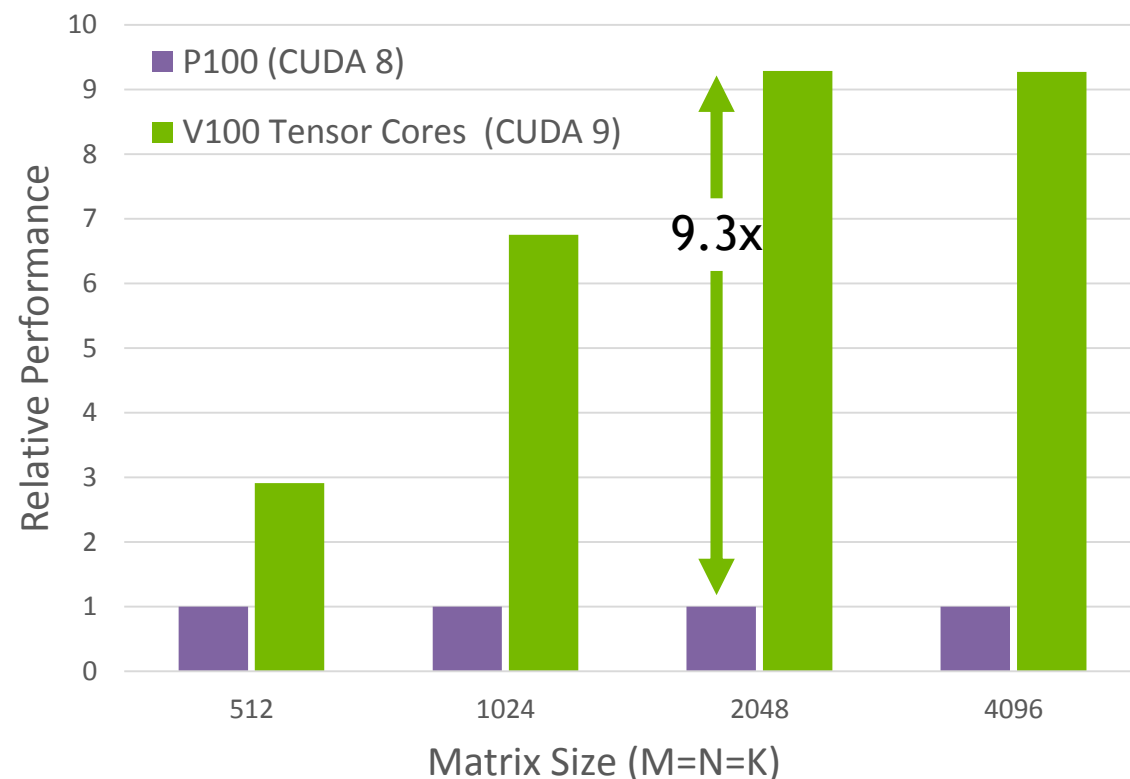
cuBLAS GEMMS FOR DEEP LEARNING

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

cuBLAS Single Precision (FP32)



cuBLAS Mixed Precision (FP16 Input, FP32 compute)



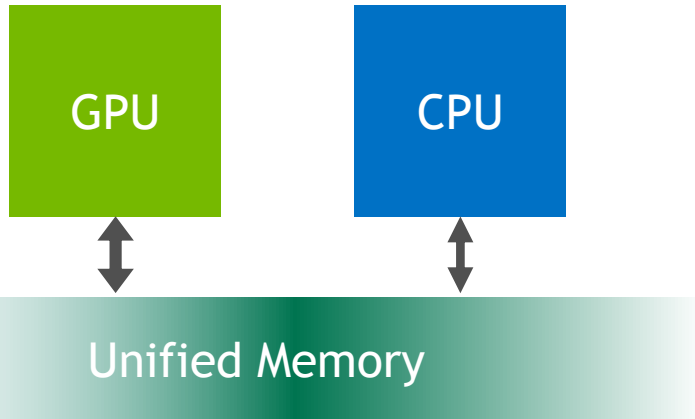


MEMORY

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

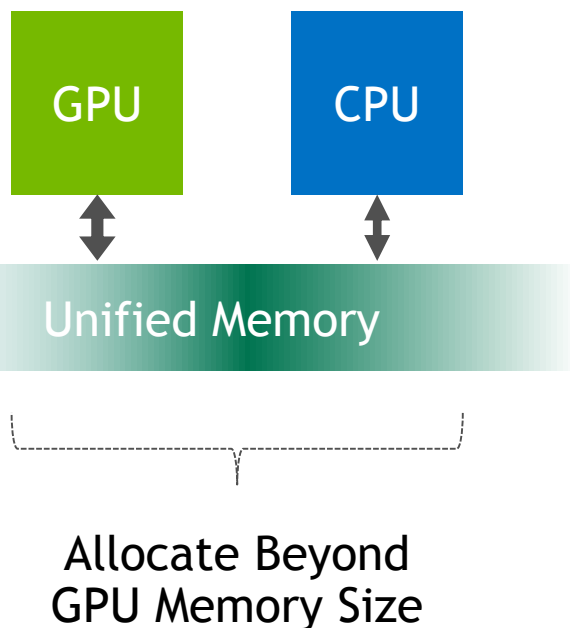
Usage hints via `cudaMemAdvise` API
Explicit prefetching API

Simpler
Data Access

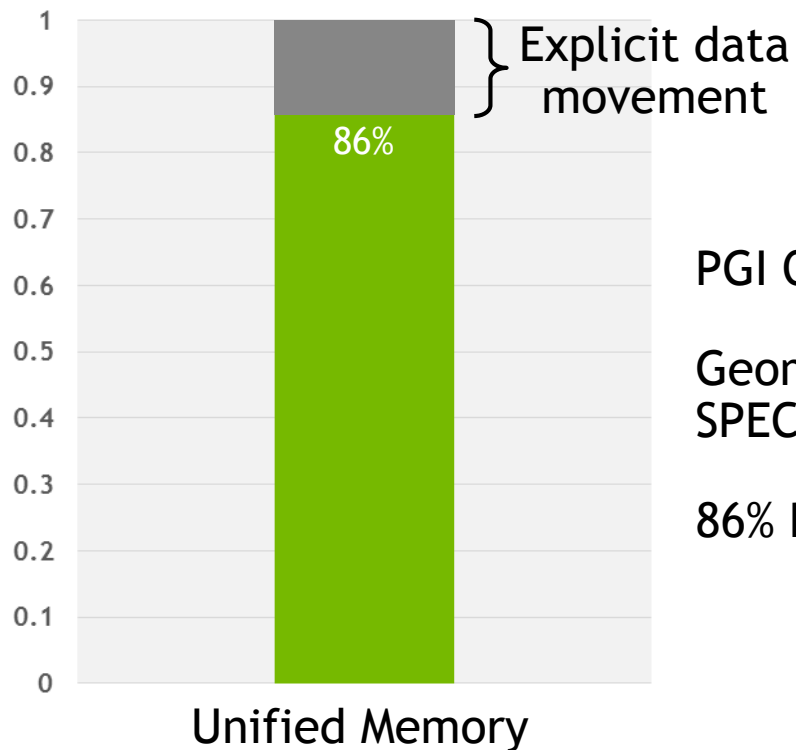
CPU/GPU Data coherence
Unified memory atomic operations

STATE OF UNIFIED MEMORY

High performance, low effort



Performance vs no Unified Memory



PGI OpenACC on Pascal P100

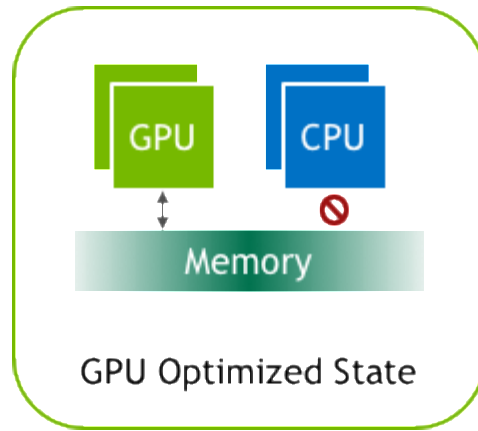
Geometric mean across all 15 SPEC ACCEL™ benchmarks

86% PCI-E, 91% NVLink

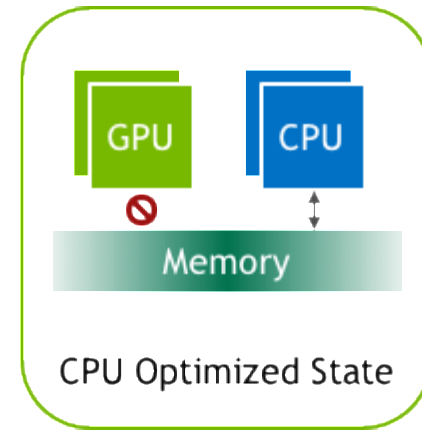
Automatic data movement for allocatables

VOLTA + UNIFIED MEMORY

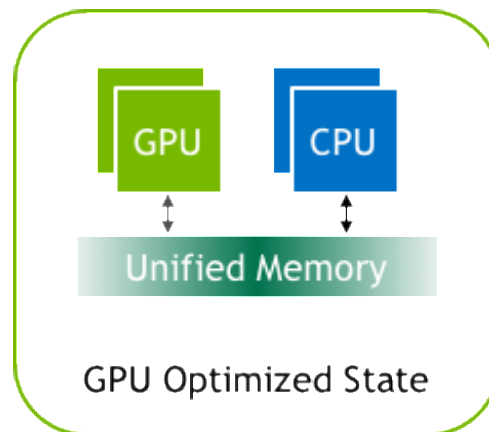
VOLTA + PCIE CPU



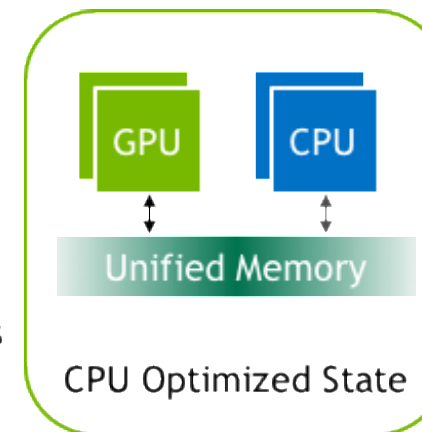
Page Migration Engine
↔
+ *Access counters*



VOLTA + NVLINK CPU



Page Migration Engine
↔
+ *Access counters*
+ *New NVLink Features*
(*Coherence, Atomics, ATS*)

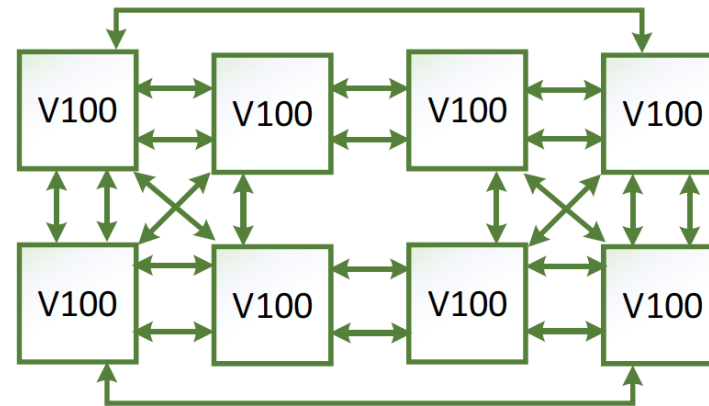




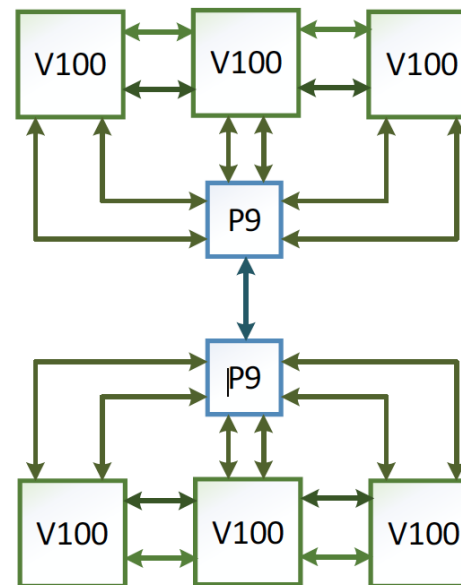
NVLINK

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		

TESLA T4

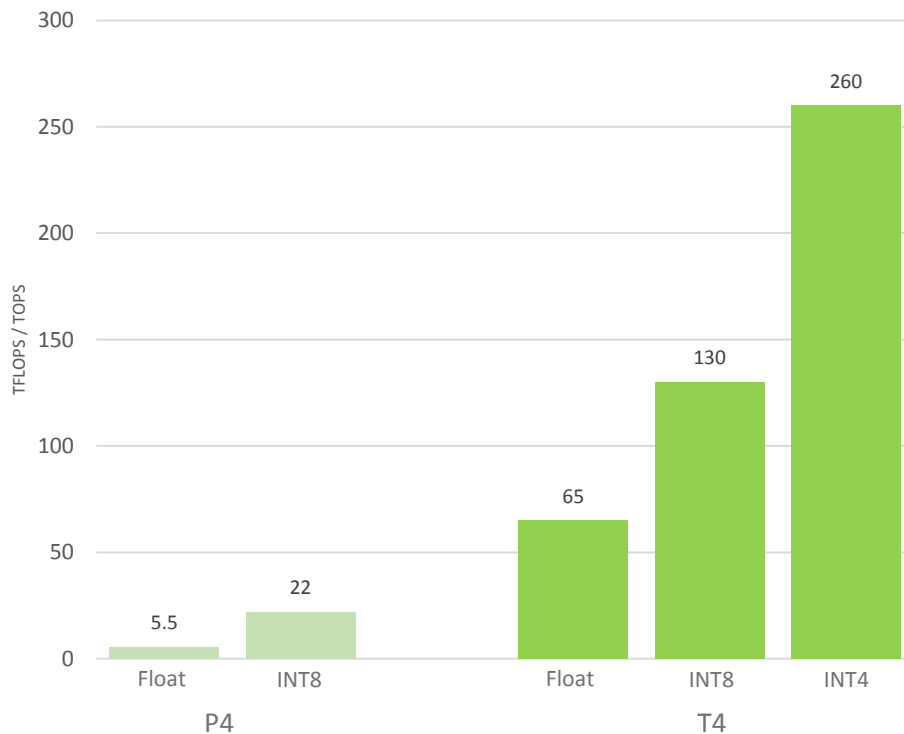


2,560 CUDA cores + 320 Tensor Cores
8.1 TFLOPS FP32 | 65 FP16 TFLOPS
130 INT8 TOPS | 260 INT4 TOPS

16GB GDDR6 Memory | 320GB/s

75 W Low Profile PCI-e

Peak Performance



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

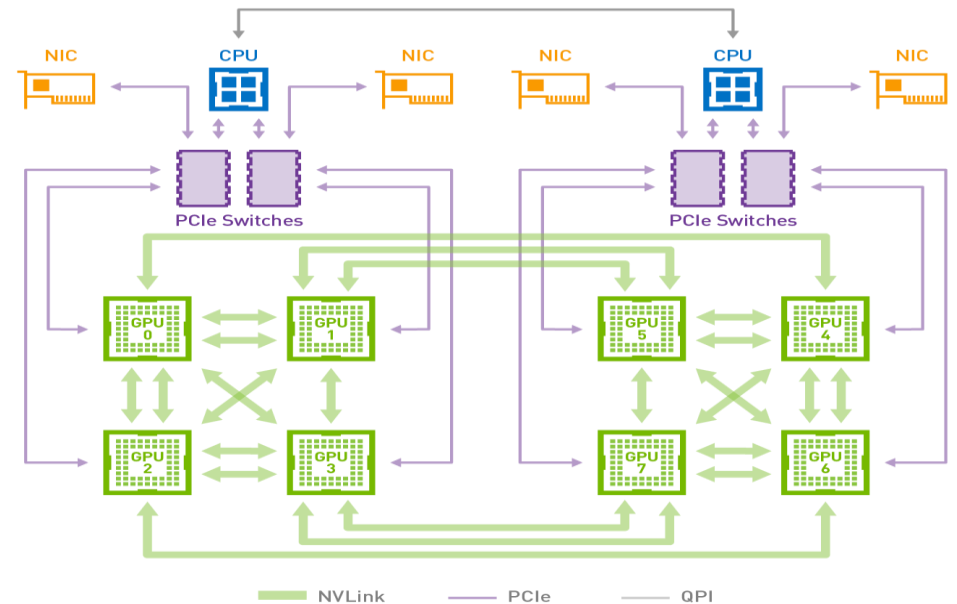
Dual 10GbE, Quad IB 100Gb

3RU - 3200W

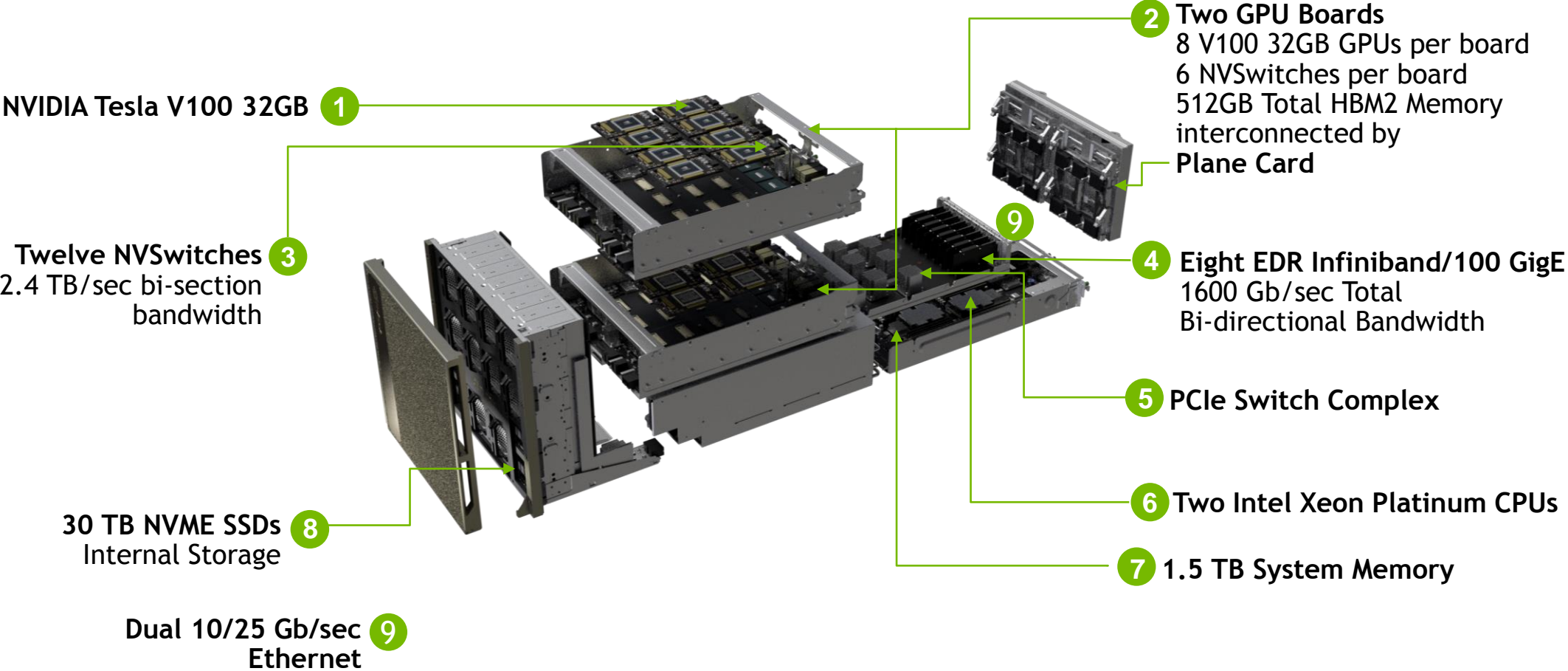
Optimized Deep Learning Software
across the entire stack

Containerized frameworks

Always up-to-date via the cloud



NVIDIA DGX-2

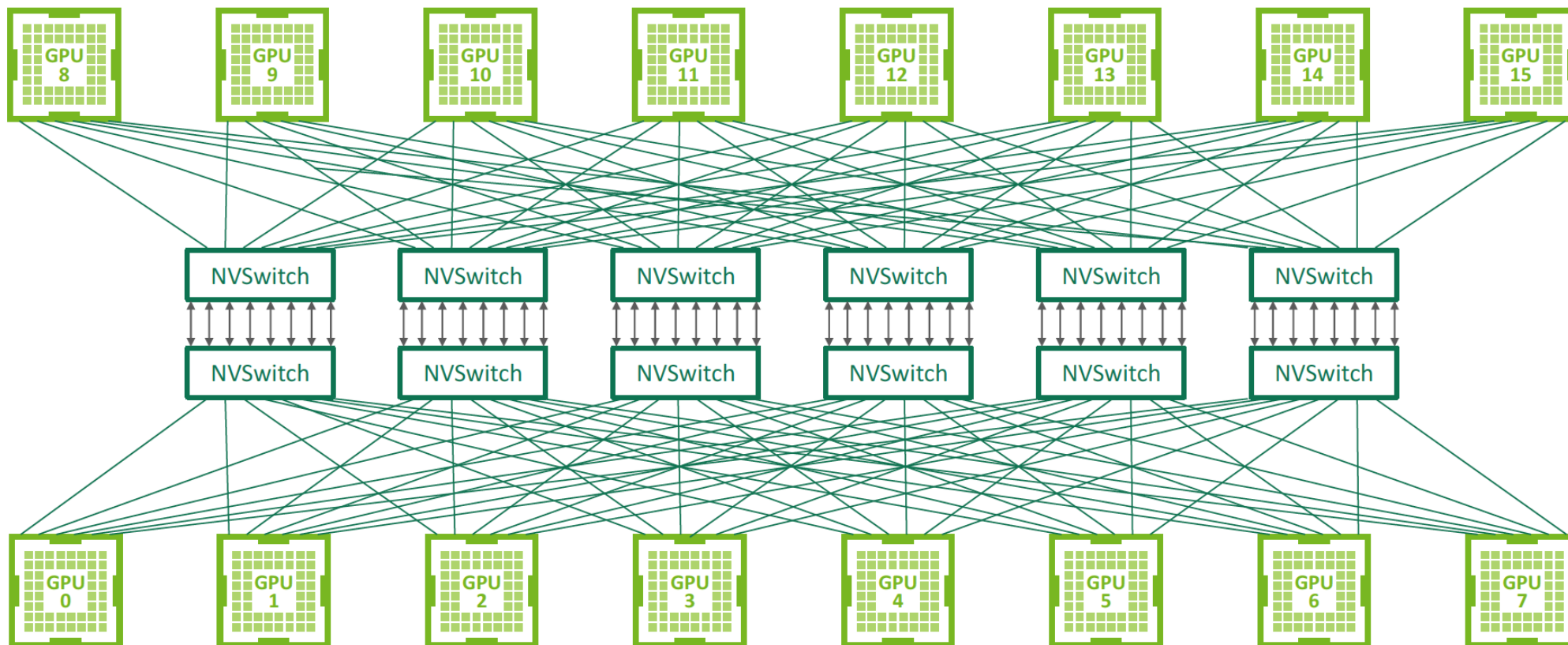


NVSWITCH

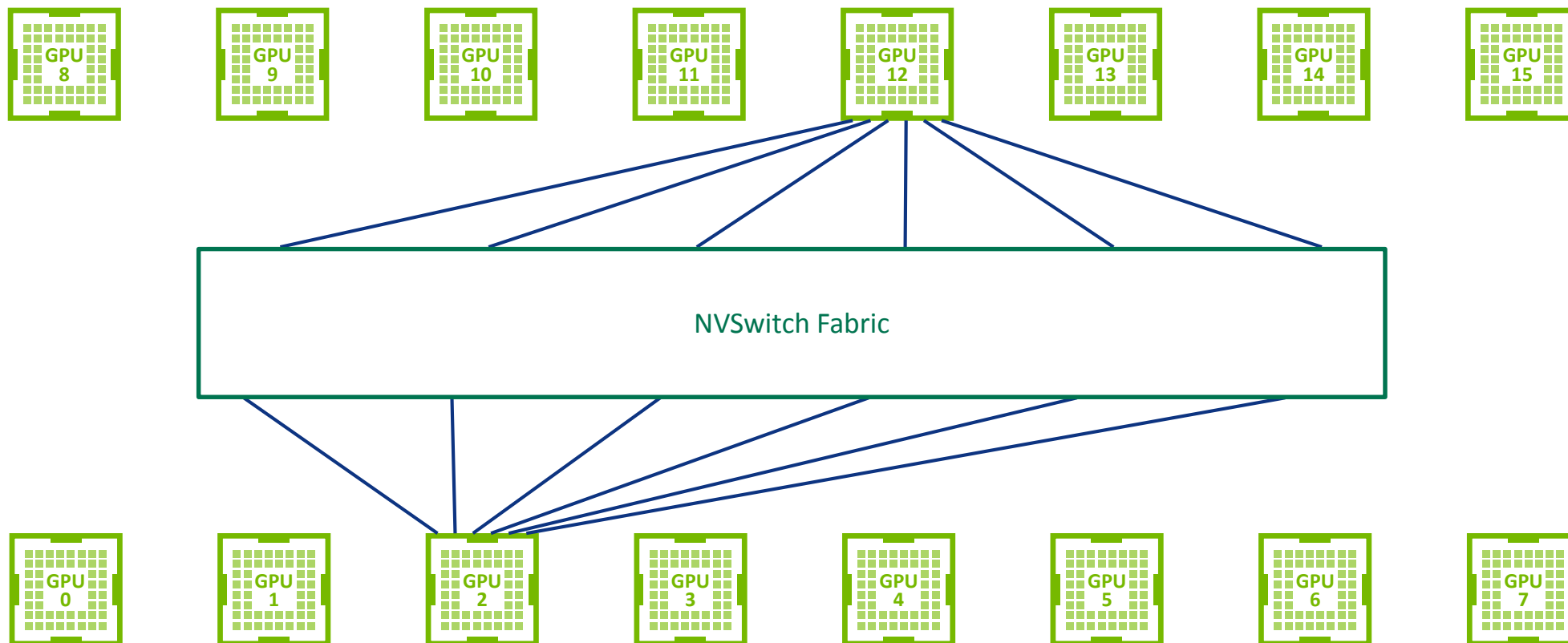


- 18 NVLINK ports
 - @50 GB/s per port bi-directional
 - 900 GB/s total bi-directional
- Fully connected crossbar
- X4 PCIe Gen2 Management port
- GPIO
- I2C
- 2 billion transistors

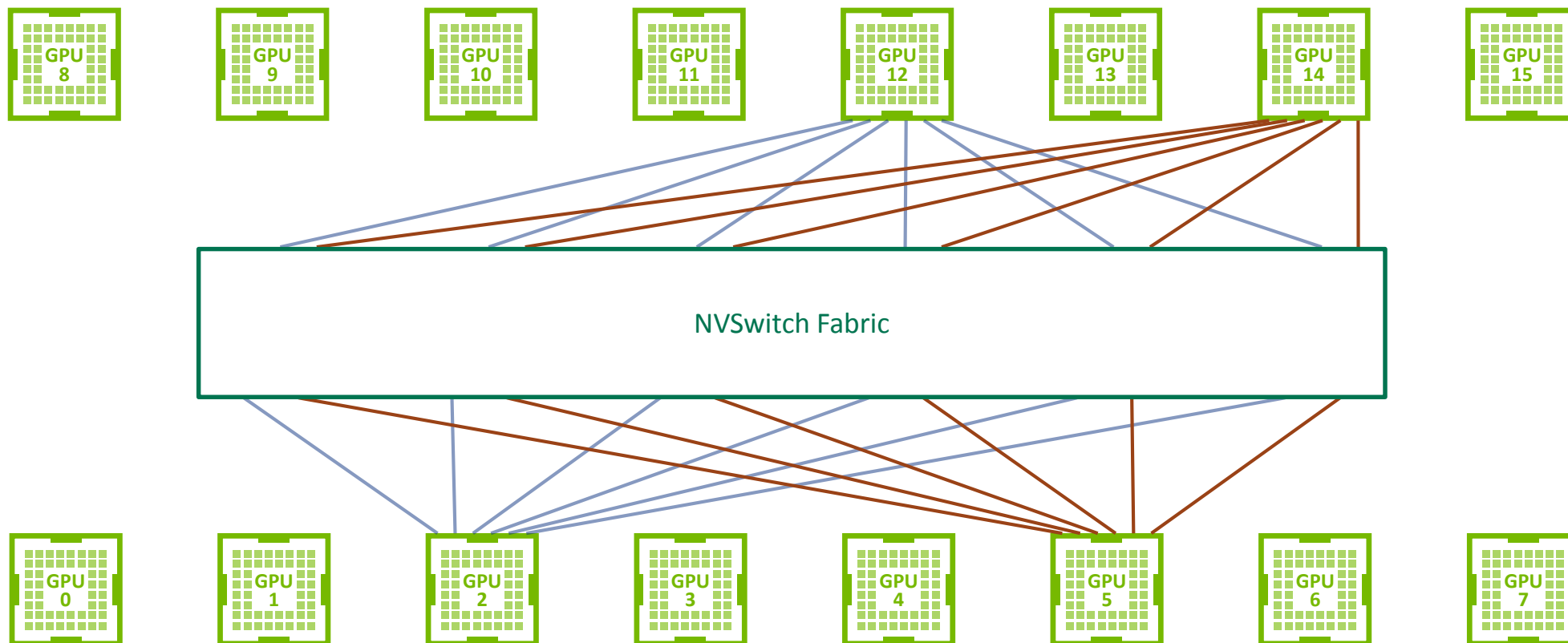
FULL NON-BLOCKING BANDWIDTH



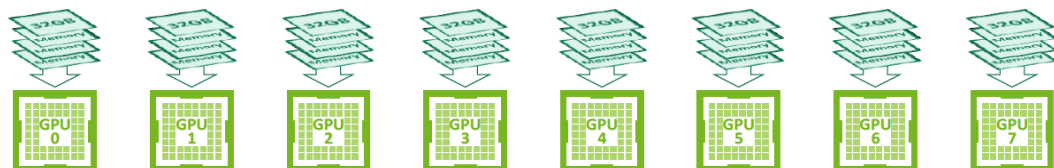
FULL 6-WAY POINT-TO-POINT



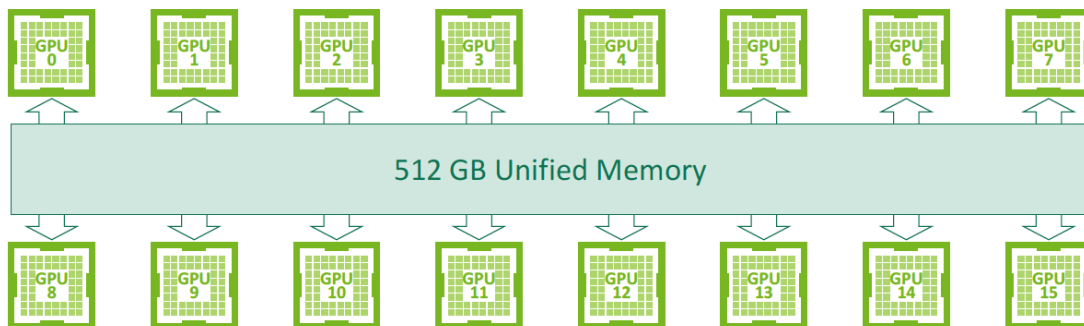
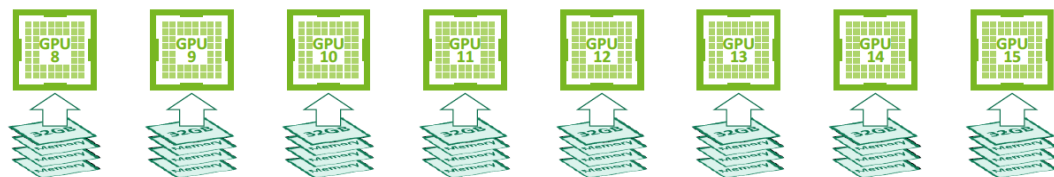
INDEPENDENT COMMUNICATION



NVSWITCH



16x 32GB Independent Memory Regions



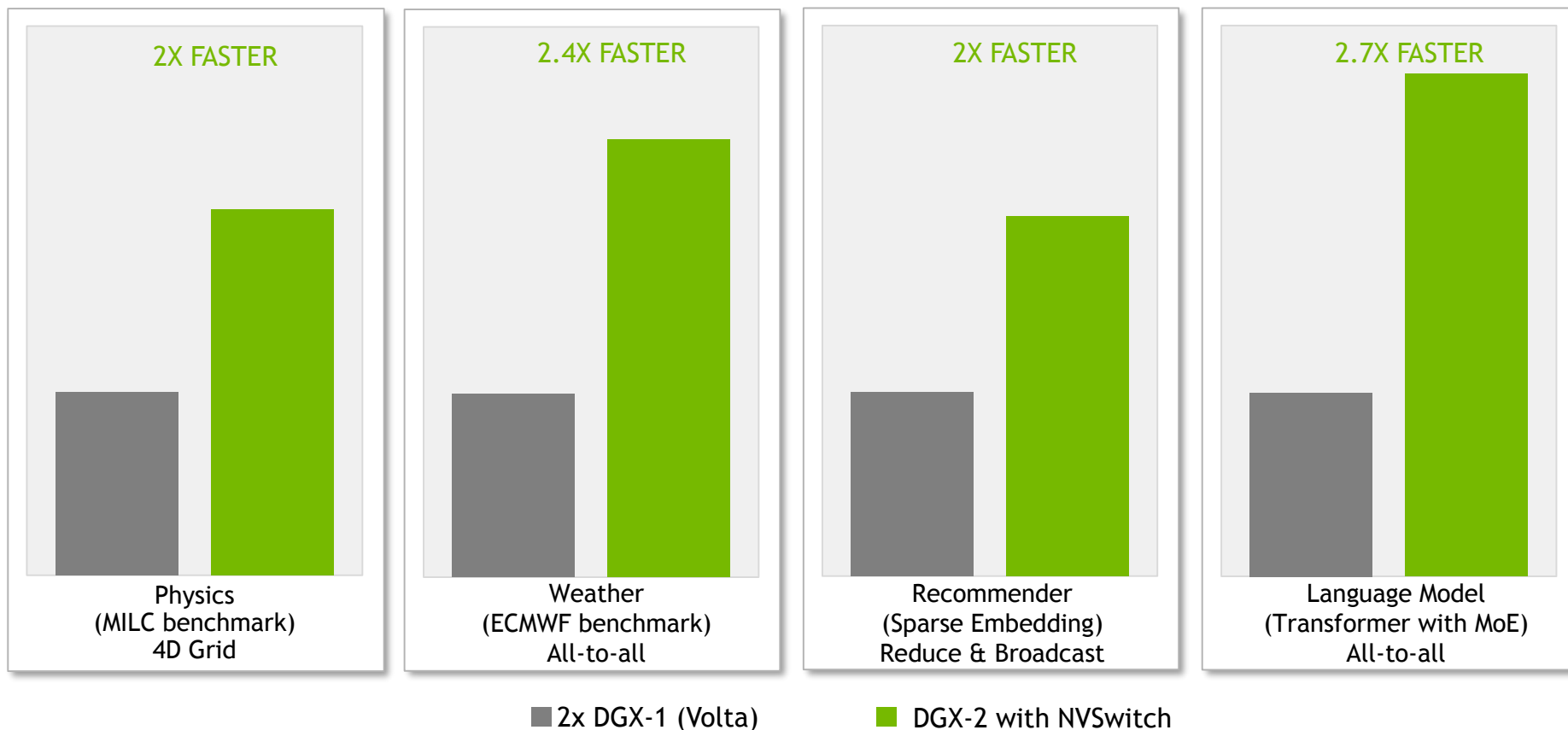
NVLINK PROVIDES

- 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

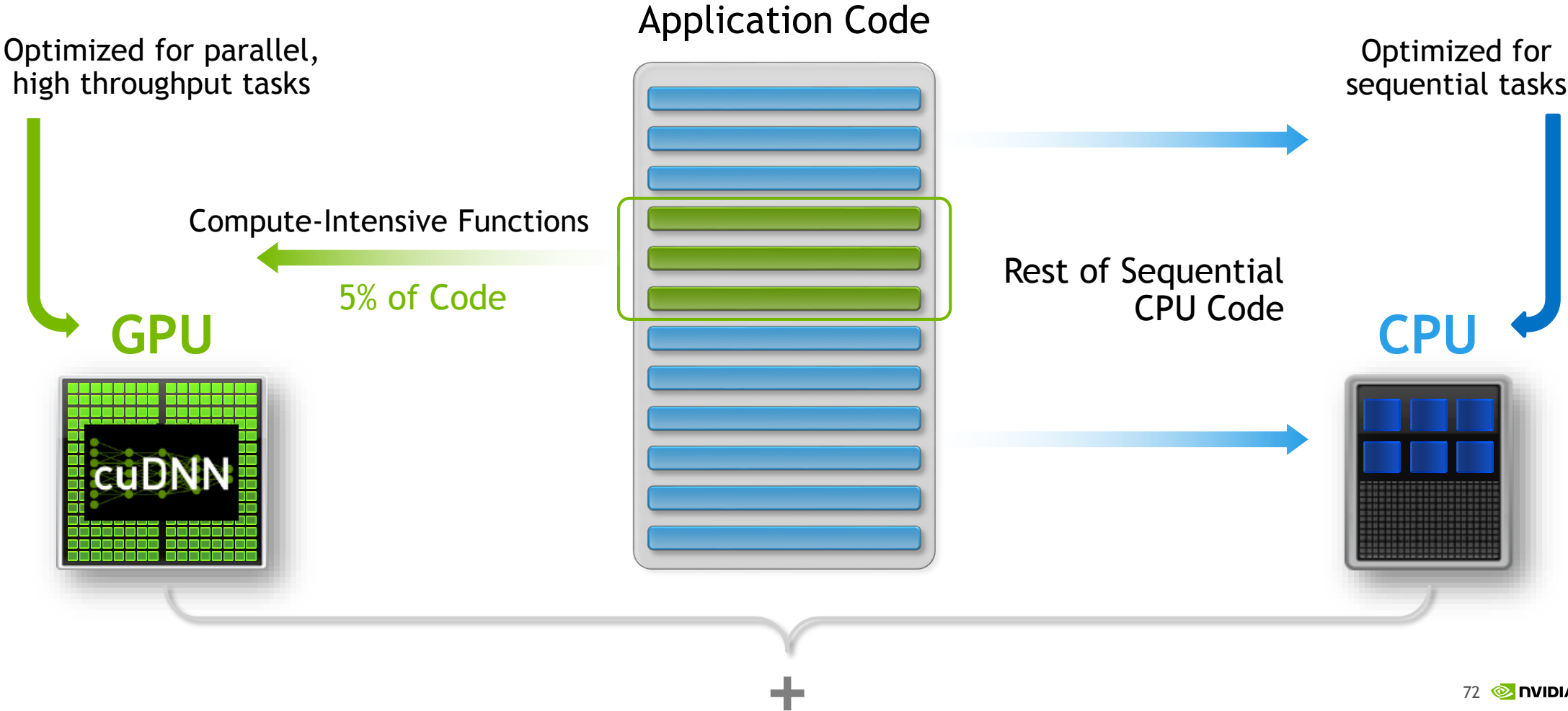
2X HIGHER PERFORMANCE WITH NVSWITCH



The background features a complex network of thin, light green lines connecting various glowing green nodes of different sizes. The nodes are scattered across the dark blue and black background, creating a sense of depth and connectivity. The overall aesthetic is futuristic and technical.

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





POPULAR GPU-ACCELERATED APPLICATIONS

- 02 Research: Higher Education and Supercomputing
 - COMPUTATIONAL CHEMISTRY AND BIOLOGY
 - BIOLOGICAL ANALYTICS
 - PHYSICS
 - HEALTHCARE AND CLIMATE PREDICTION
- 06 Defense and Intelligence
- 07 Computational Finance
- 08 Manufacturing: CAD and CAE
 - COMPUTER AIDED DESIGN
 - COMPUTATIONAL FLUID DYNAMICS
 - COMPUTATIONAL STRUCTURAL MECHANICS
 - ELECTRONIC DESIGN AUTOMATION
- 10 Media and Entertainment
 - ANIMATION, MODELING AND RENDERING
 - COLLISION DETECTION AND DRAW MANAGEMENT
 - COMPUTATIONAL FINISHING AND EFFECTS
 - EDITING
 - EMERGING AND DIGITAL DISTRIBUTION
 - ON-SET, REVIEW AND STEREO TOOLS
 - SIMULATION
 - WEATHER GRAPHS
- 14 Oil and Gas

Research: Higher Education and Supercomputing

COMPUTATIONAL CHEMISTRY AND BIOLOGY

Bioinformatics

Application	Description	Parallelization	Performance	GPU Support	Availability
Barrage	Sequence mapping software	Alignment of short sequencing reads	3-18s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 8.0.2
CUBAM++	Open source software for Smith-Waterman protein database searches on GPUs	Parallel search of Smith-Waterman Database	10-50s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 2.0.8
CUSHAM	Parallelized short read aligner	Parallel, accurate long read aligner - gapped alignments to large genomes	10s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 1.0.40
GPU-BLAST	Local search with fast k-tuple heuristic	Pruned alignment according to k-tuple, multi-cpu threads	3-4s	T.2075, 2090, 410, 420, K20K	Single only Available now Version 2.2.28
GPU-HMMER	Parallelized local and global search with profile Hidden Markov models	Parallel local and global search of Hidden Markov Models	60-100s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 2.2.2
mDUSA-MEME	Ultra-fast scalable motif discovery algorithm based on MEME	Scalable motif discovery algorithm based on MEME	4-15s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 3.0.12
SeqFlux	A GPU Accelerated Sequence Analysis Toolkit	Reference assembly, blast, ortho-sequences, term, de novo assembly	400s	T.2075, 2090, 410, 420, K20K	Yes Available now
USEN	Open-source Smith-aligner for SRA/EU/DA. Suffix array based repeats filter and output	Fast short read alignment	0-5s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 1.11
WishLM	Fits numerous linear models to a fixed design and response	Parallel linear regression on multiple similarly-shaped models	100s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 8.1-1

Molecular Dynamics

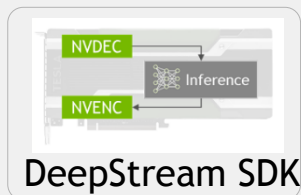
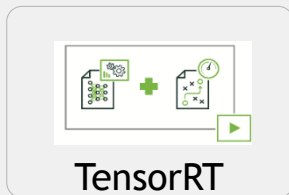
Application	Description	Parallelization	Performance	GPU Support	Availability
Abalone	Models molecular dynamics of biopolymers for simulations of proteins, DNA and ligands	Simulations on 1000 GPUs	4-27s	T.2075, 2090, 410, 420, K20K	Single Only Available now Version 1.0.40
ACEMD	GPU simulation of molecular mechanics force fields, implicit and explicit solvent	Written for use on GPUs	140 ns/day GPU version only	T.2075, 2090, 410, 420, K20K	Yes Available now
AMBER	Suite of programs to simulate molecular dynamics on biomolecules	PMEMD: explicit and implicit solvent	80-140 ns/day JAC NVE	T.2075, 2090, 410, 420, K20K	Yes Available now Version 12 + SUGRA
DL-POLY	Simulate macromolecules, polymers, nano systems, etc on a distributed memory parallel computer	Two-body forces, Link-cell pairs, Ewald SPME forces, SHAKE W	4s	T.2075, 2090, 410, 420, K20K	Yes Available now Version 0. Source only
CHARMM	MD package to simulate molecular dynamics on biomolecules	Implicit Sol, Explicit Sol Solvent via OpenMM	700s	T.2075, 2090, 410, 420, K20K	Yes in Development 06/12
DRIMACS	Simulation of biomolecular molecules with complicated bead interactions	Implicit Sol, ExplicitSol solvent	140 ns/day SHAR	T.2075, 2090, 410, 420, K20K	Single only Available now Version 6.6 in SH/12
HOOMD-blue	Particle dynamics package written primarily for GPUs	Written for GPUs	3s	T.2075, 2090, 410, 420, K20K	Yes Available now
LAMMPS	Classical molecular dynamics package	Lemmer-Jones, Morse, Buckingham, CHARMM, Tabular, Course granular, Acceleration, etc.	3-18s	T.2075, 2090, 410, 420, K20K	Yes Available now

550+ GPU-Accelerated Applications
www.nvidia.com/appscatalog

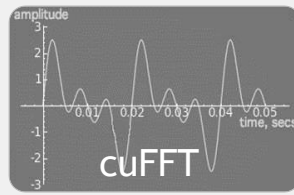
GPU ACCELERATED LIBRARIES

“Drop-in” Acceleration for Your Applications

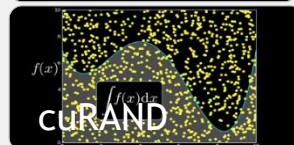
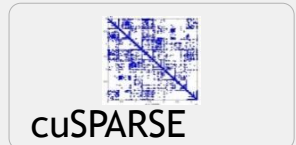
DEEP LEARNING



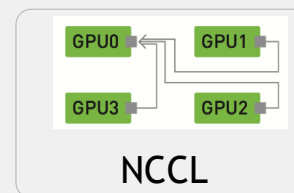
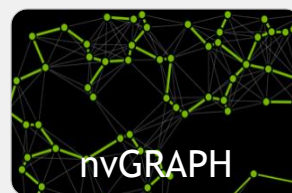
SIGNAL, IMAGE & VIDEO



LINEAR ALGEBRA



PARALLEL ALGORITHMS





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>
  }
}
```

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

Simple

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

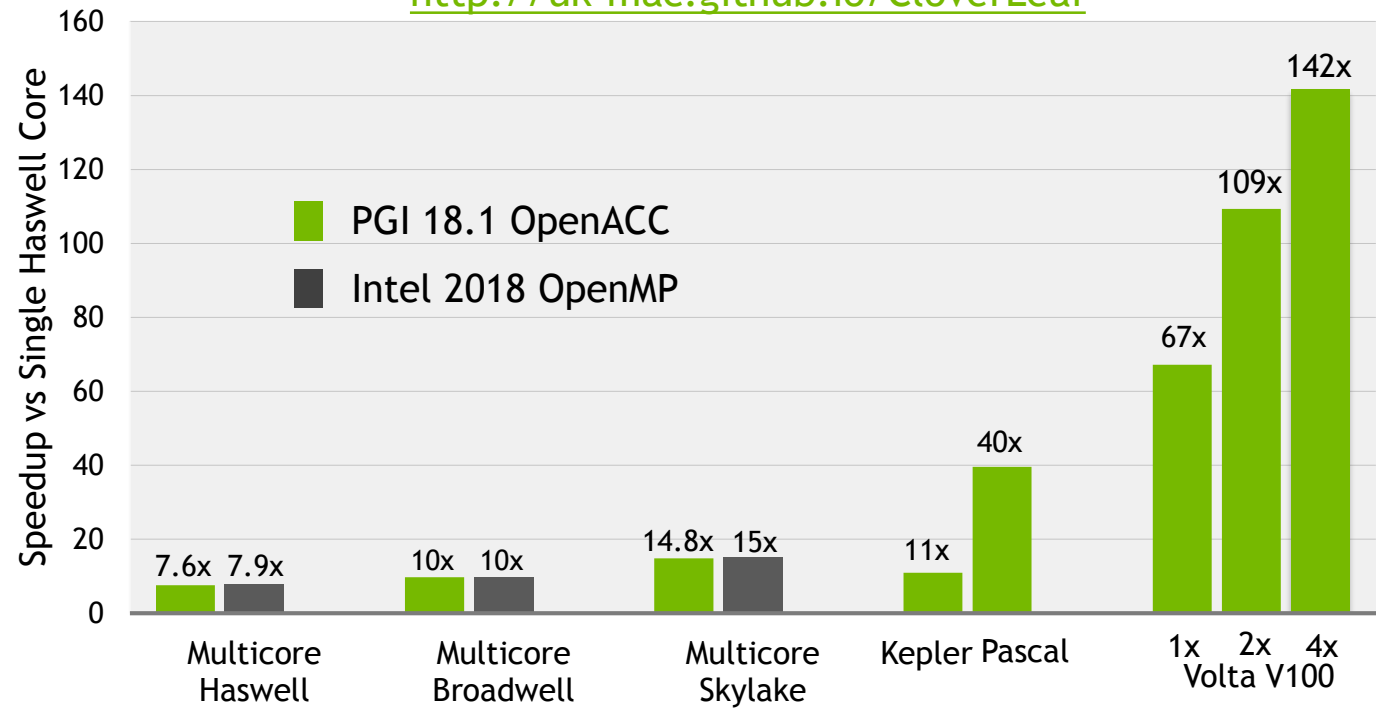
NVIDIA GPU

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; CloverLeaf_Serial; CloverLeaf_ref (MPI+OpenMP); CloverLeaf_OpenACC (MPI+OpenACC)

Data compiled by PGI February 2018.

OPENACC

Incremental

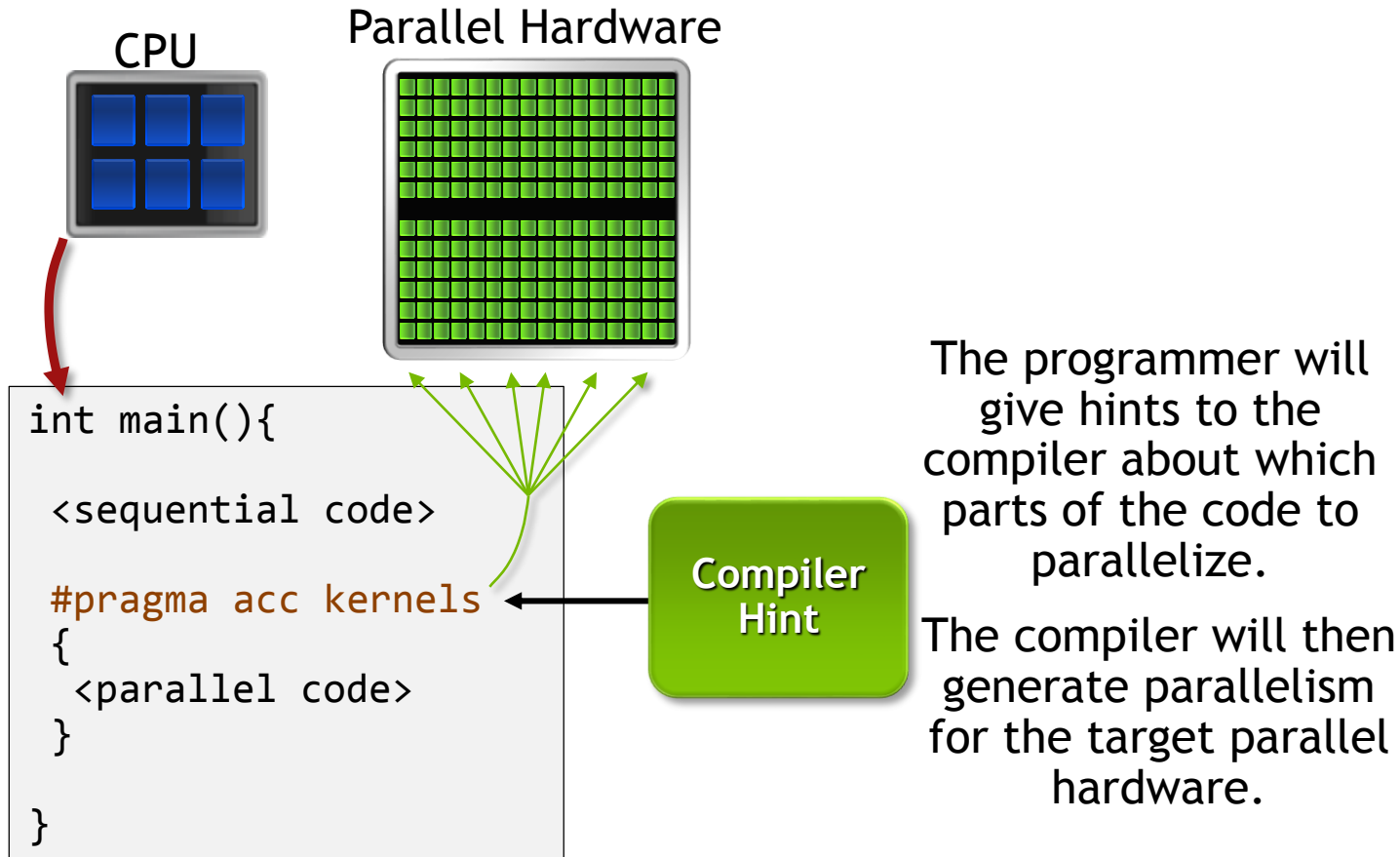
- Make small, incremental changes to the code
- If any errors occur, easily able to revert back to an earlier, working version of the code
- Start with a working sequential code, and add improvements

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



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



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

Resources

<https://www.openacc.org/resources>

The screenshot shows the OpenACC website's 'Resources' page. It features a navigation bar with 'About', 'Tools', 'News', 'Events', 'Resources', 'Spec', and 'Community'. Below the navigation, there's a search bar and a 'Resources' section with a description: 'A complete library of OpenACC materials that includes a collection of video tutorials, guides, online courses, books and more.' The page is divided into three columns: 'Guides' (listing 'Introduction to OpenACC Quick Guides', 'OpenACC Programming and Best Practices Guide', and 'OpenACC 2.3 API Reference Card'), 'Books' (listing 'Parallel Programming with OpenACC' and 'Programming Massively Parallel Processors, Third Edition: A Hands-on Approach'), and 'Tutorials' (described as 'Video tutorials to help start with OpenACC and advance your skills').

Compilers and Tools

<https://www.openacc.org/tools>

The screenshot shows the OpenACC website's 'Downloads & Tools' page. It features a navigation bar with 'About', 'Tools', 'News', 'Events', 'Resources', 'Spec', and 'Community'. Below the navigation, there's a search bar and a 'Downloads & Tools' section with a description: 'OpenACC compilers, profilers and debuggers are designed and available to download from multiple vendors and academic organizations.' The page is divided into two columns: 'Commercial Compilers' (listing 'CRAY THE SUPERCOMPUTER COMPANY' and 'PGI Accelerator Compilers with OpenACC Directives') and 'Open Source Compilers' (listing 'Contact National Supercomputing Center in Wuji for more information.' and 'GCC 6 Includes initial support for OpenACC 2.3').

Success Stories

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

The screenshot shows the OpenACC website's 'Success Stories' page. It features a navigation bar with 'About', 'Tools', 'News', 'Events', 'Resources', 'Spec', and 'Community'. Below the navigation, there's a search bar and a 'Success Stories' section with a description: 'Applications across multiple domains have been accelerated with OpenACC. Scientists and researchers who have been working on these applications are sharing their results and experiences.' The page displays three video thumbnails with play buttons and a 'Watch more OpenACC Videos on YouTube' link.

Events

<https://www.openacc.org/events>

The screenshot shows the OpenACC website's 'Events' page. It features a navigation bar with 'About', 'Tools', 'News', 'Events', 'Resources', 'Spec', and 'Community'. Below the navigation, there's a search bar and an 'Events' section with a description: 'The OpenACC Community organizes a variety of events throughout the year. Events vary from talks at conferences to workshops, hackathons, online courses and User Group meetings. Join our events around the world to learn OpenACC programming and to participate in activities with the OpenACC User Group.' The page displays a photo of a workshop and a '2017 Calendar' section with a '15 AUG WORKSHOP Parallel Programming with OpenACC on CPUs and GPUs' entry for August 15, 2017, at the Oak Ridge National Laboratory (ORNL).

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

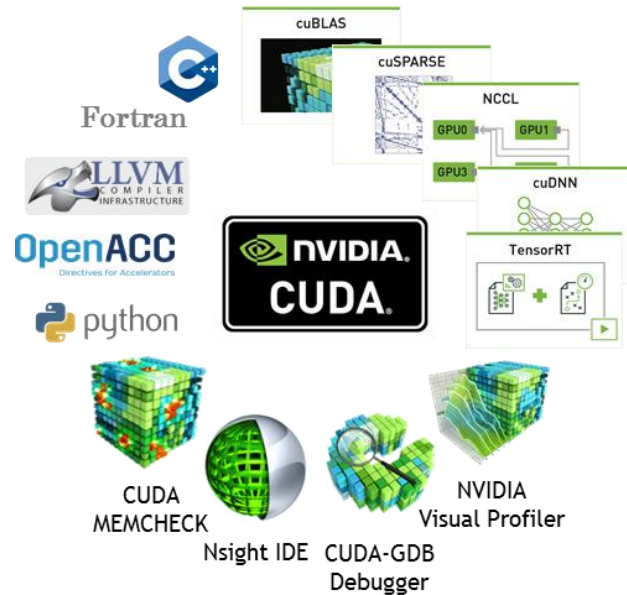
pgicompilers.com/community



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

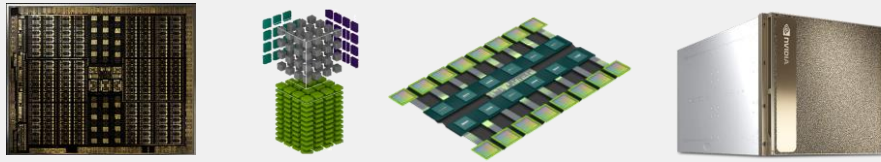
Rapid innovation in library performance and functionality

Library Meta Packages independent of toolkit for easy deployment

INTRODUCING CUDA 10.0

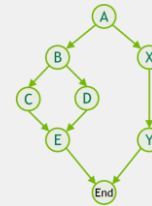
TURING AND NEW SYSTEMS

New GPU Architecture, Tensor Cores, NVSwitch Fabric



CUDA PLATFORM

CUDA Graphs, Vulkan & DX12 Interop, Warp Matrix

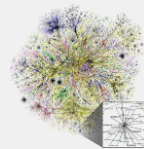
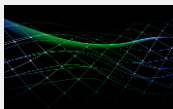


$$D = \begin{matrix} \text{FP16 or FP32} \\ \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix} \\ \text{FP16} & \text{FP16} & \text{FP16 or FP32} \end{matrix}$$

$D = AB + C$

LIBRARIES

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



DEVELOPER TOOLS

New Nsight Products - Nsight Systems and Nsight Compute



	Source Line Registers	Sampling Data (All)	Sampling Data (No Issue)
0	0	223	0
1	13	44	0
2	130	296	0
3	30	30	0
4	590	54	0
5	325	26	0
6	200	0	0
7	386	25	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0

An abstract network visualization featuring a complex web of thin, light green lines connecting various nodes. The nodes are represented by small, glowing green circles of varying sizes and brightness. The background is a dark, almost black, gradient with some faint, larger, out-of-focus green and blue circular shapes, suggesting a vast, interconnected system. The overall aesthetic is technical and futuristic.

CUDA LIBRARIES

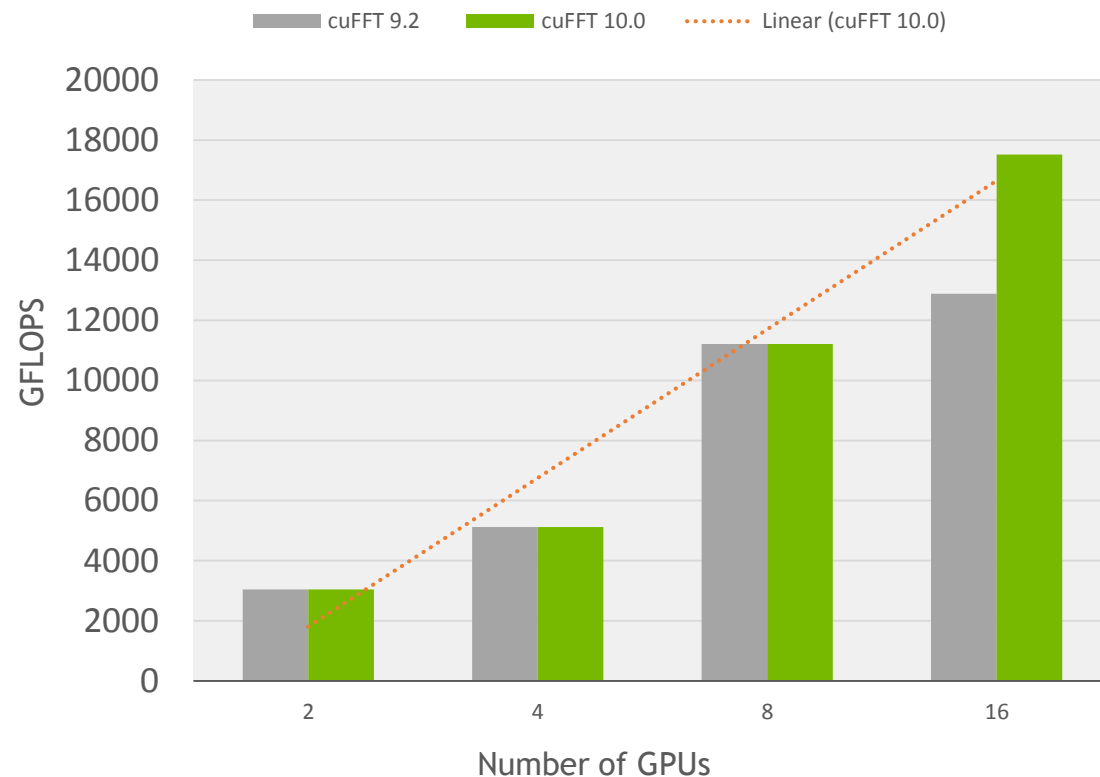
cuFFT 10.0

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)

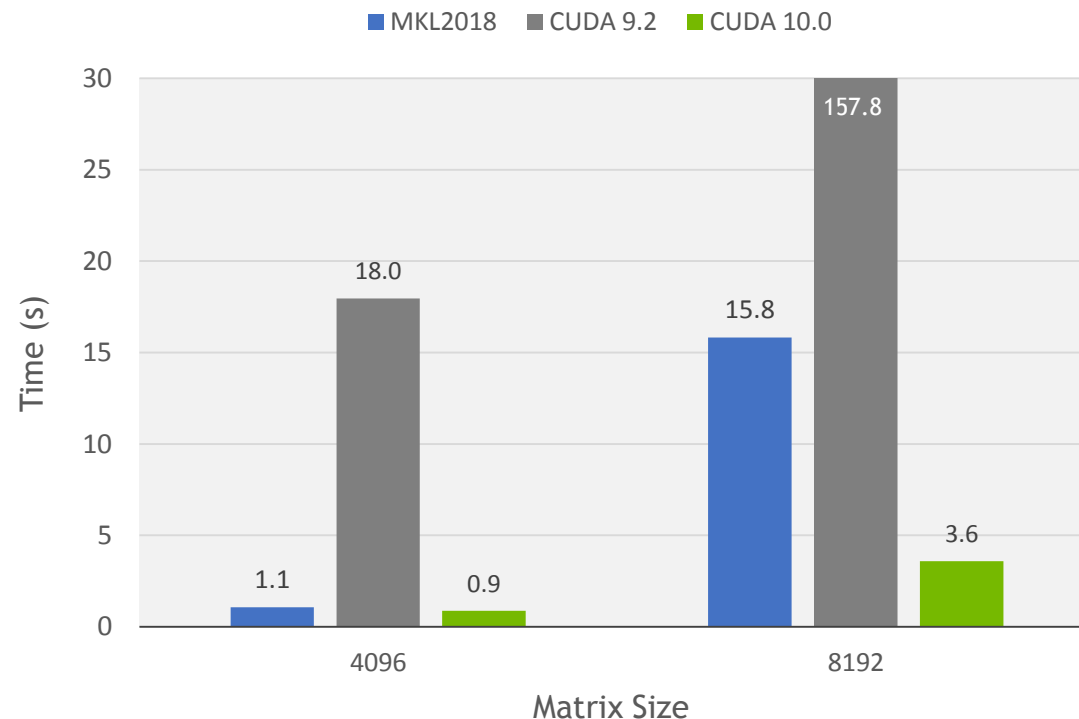
cuSOLVER 10.0

Dense Linear Algebra

Improved performance with new implementations for

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

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

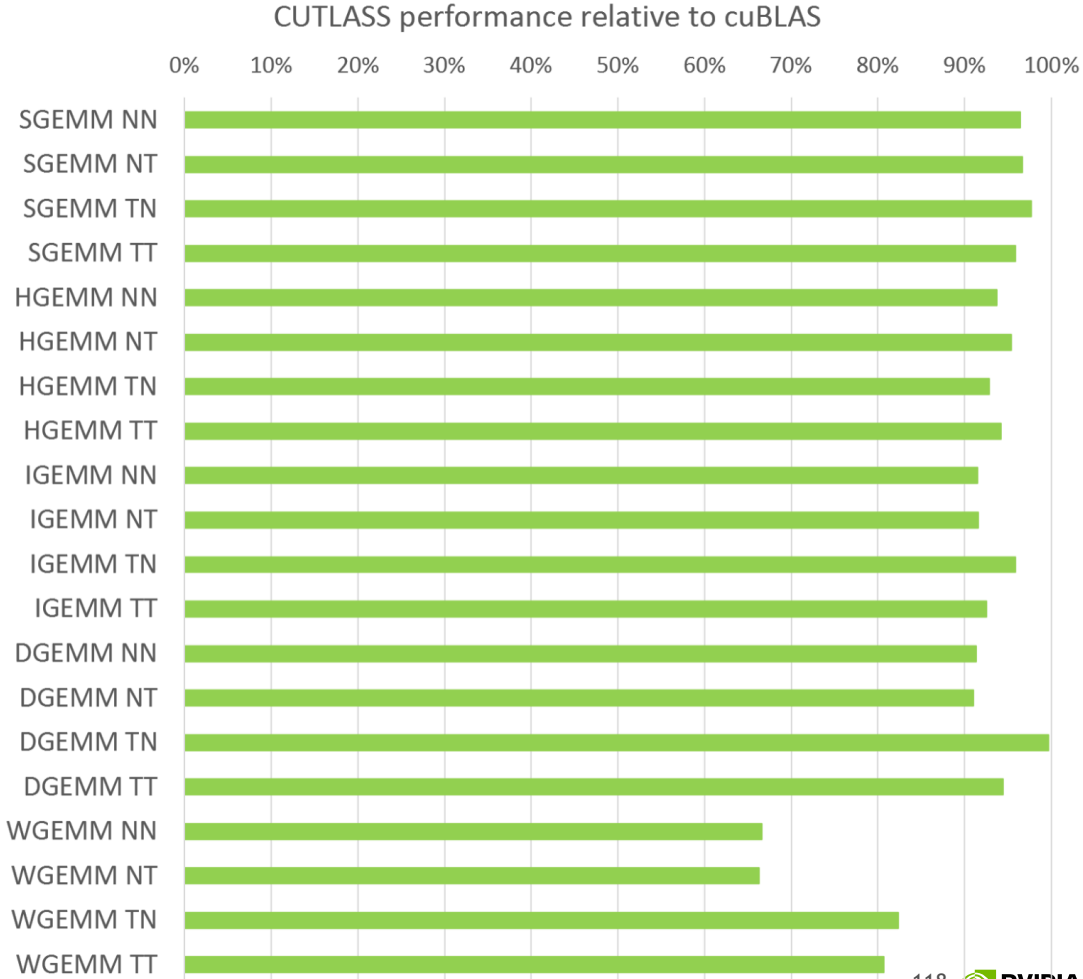
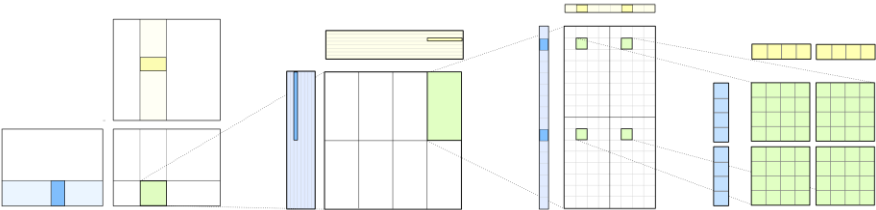
CUTLASS

Template library for linear algebra operations in CUDA C++

>90% CUBLAS performance

Open Source (3-clause BSD License)

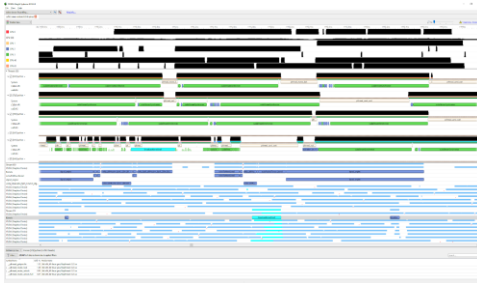
<https://github.com/NVIDIA/cutlass>





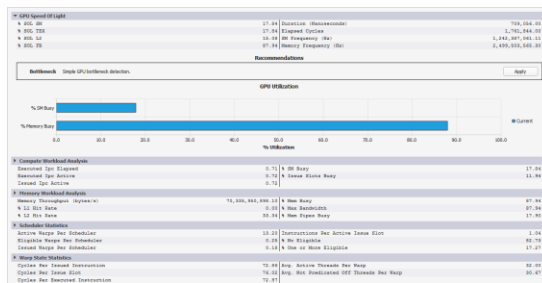
**NSIGHT
DEVELOPER TOOLS**

NSIGHT PRODUCT FAMILY



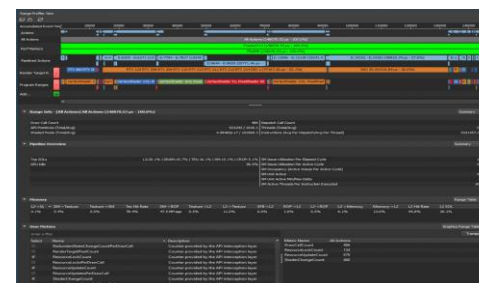
Nsight Systems

System-wide application algorithm tuning



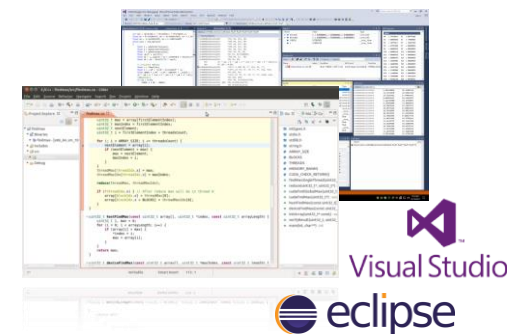
Nsight Compute

CUDA Kernel Profiling and Debugging



Nsight Graphics

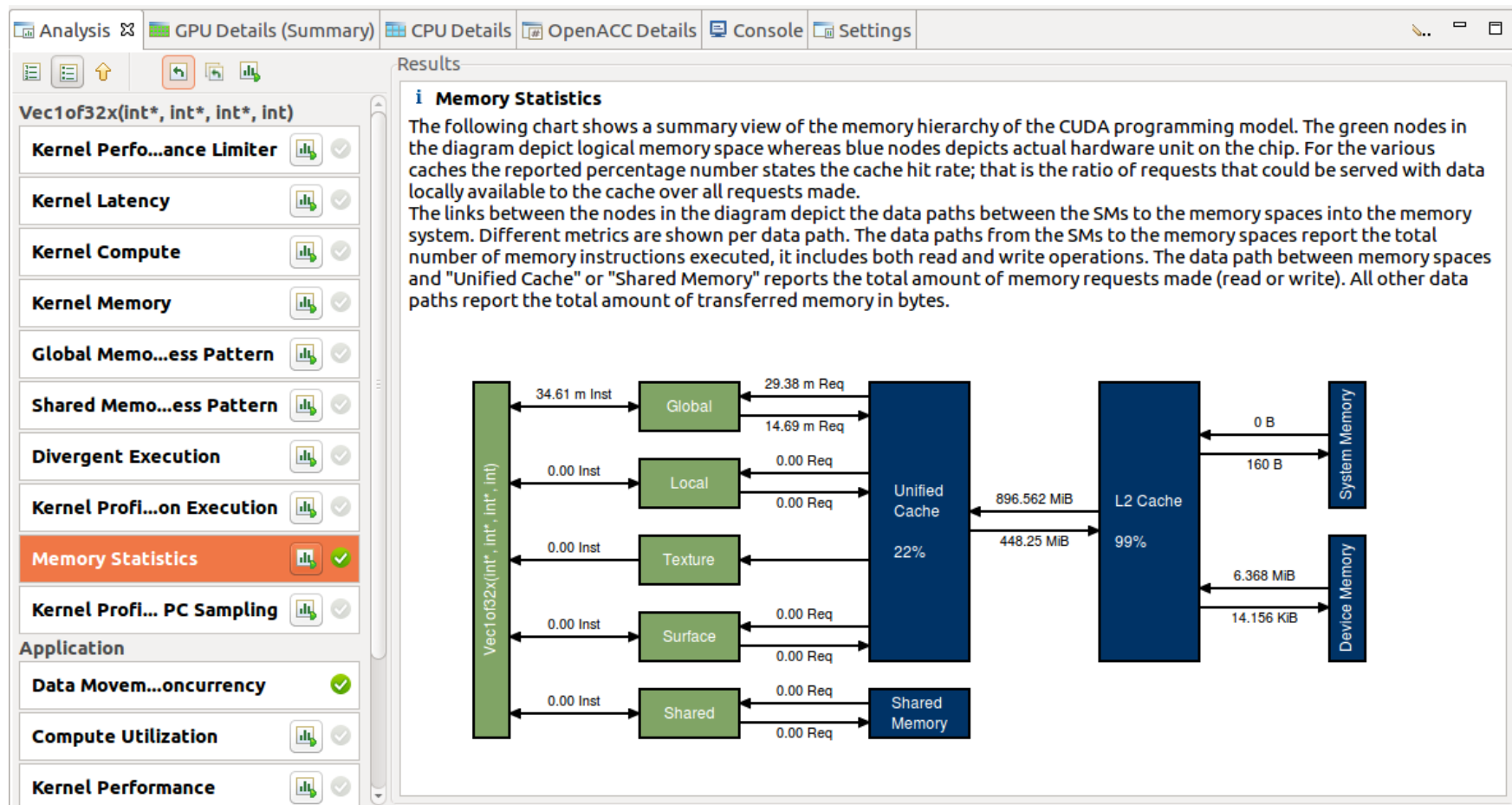
Graphics Shader Profiling and Debugging



IDE Plugins

Nsight Eclipse Edition/Visual Studio (Editor, Debugger)

HIERARCHICAL MEMORY STATISTICS

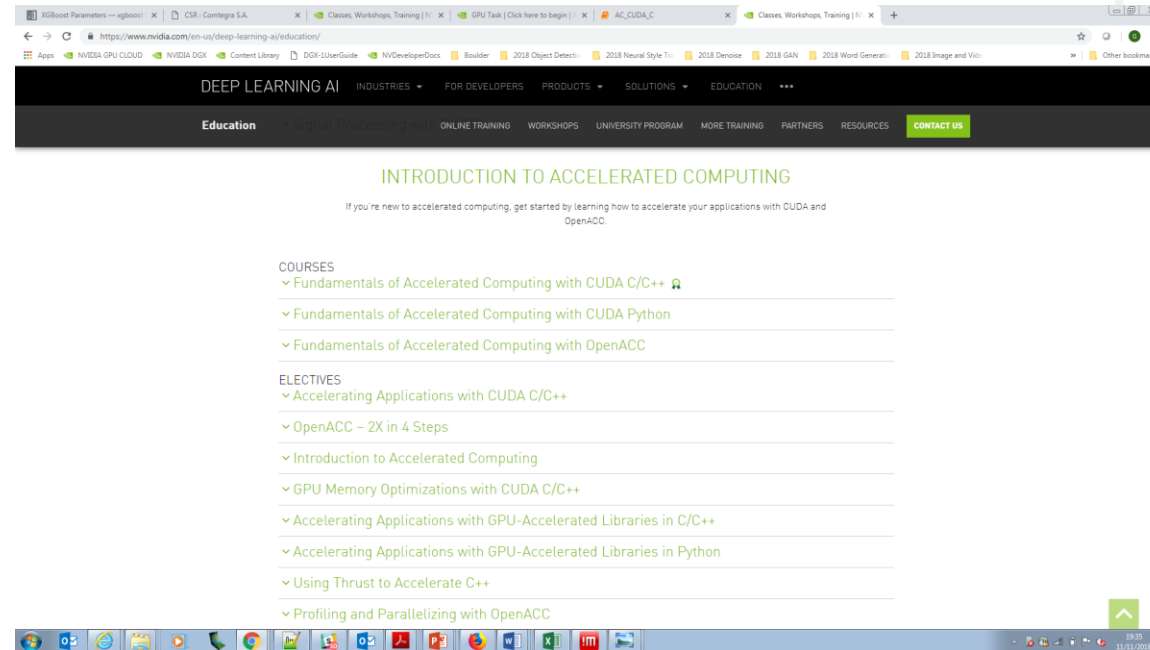
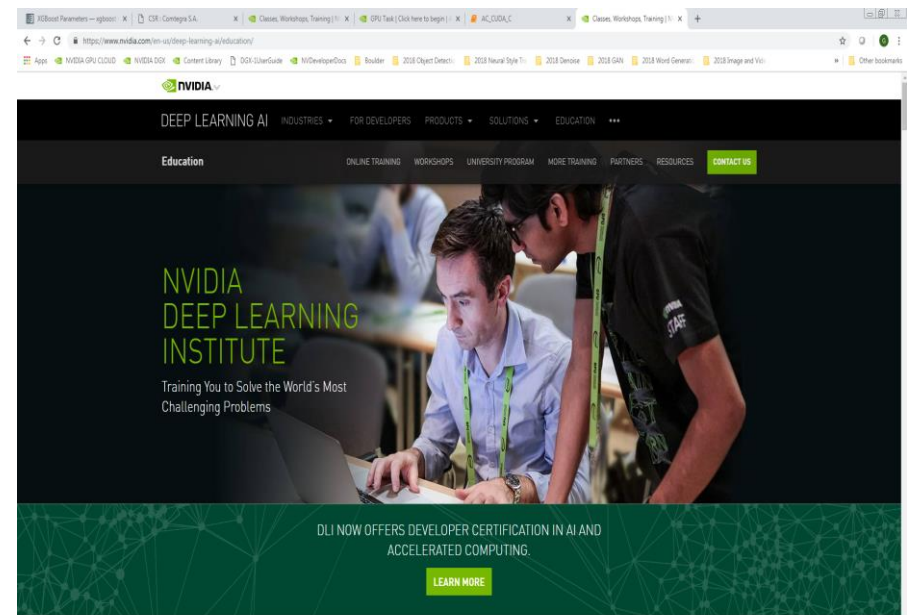


NAVIGATING TO COURSES

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++



The background features a complex network of glowing green lines and nodes. The nodes are small, bright green circles of varying sizes, some appearing as larger, more prominent hubs. The lines are thin and connect these nodes in a dense, web-like pattern. The overall color palette is dark, with the green elements standing out against a black or very dark blue background. The text 'DEEP LEARNING SDK' is positioned in the lower right quadrant of the image.

DEEP LEARNING SDK

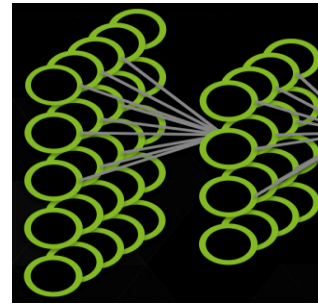
NVIDIA DEEP LEARNING INSTITUTE

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

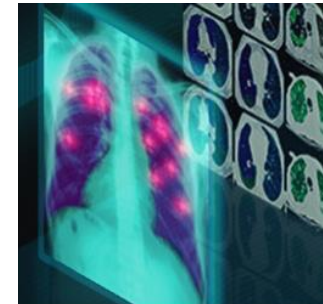
Educators can join the University Ambassador Program to teach DLI courses on campus and access resources. Learn more at www.nvidia.com/dli



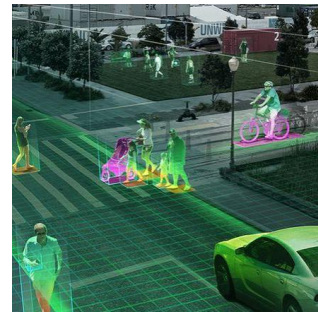
Fundamentals



Autonomous Vehicles



Healthcare



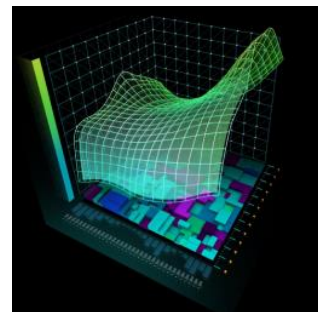
Intelligent Video Analytics



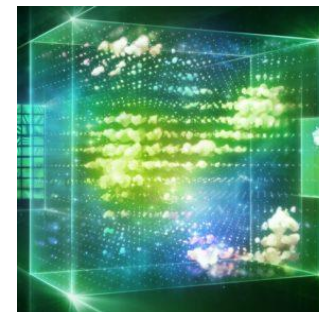
Robotics



Game Development & Digital Content



Finance



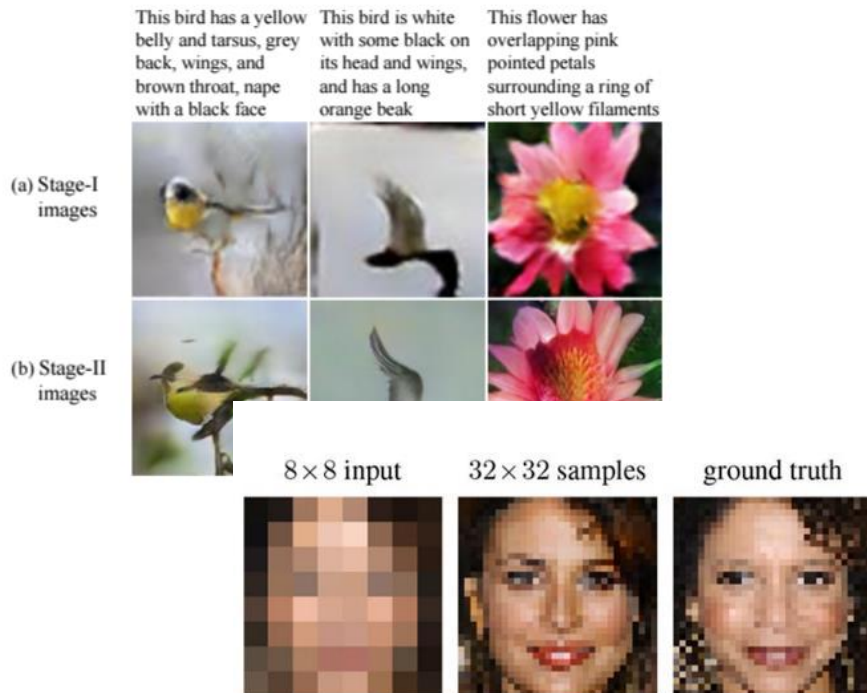
Accelerated Computing



Virtual Reality

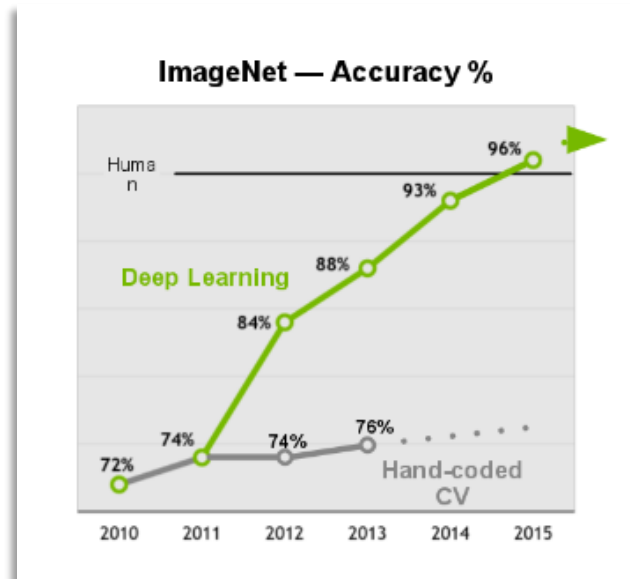
WHY THE EXCITEMENT?

GPUs as Enablers of Breakthrough Results

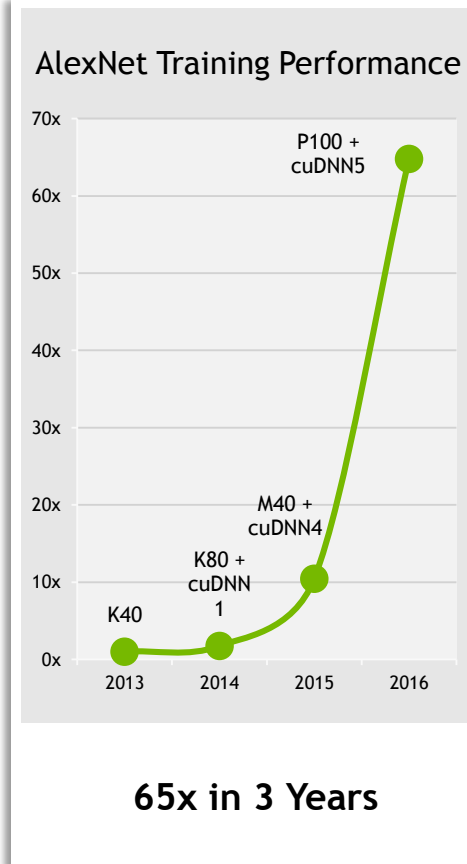


Dahl et al. 2017

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



Achieve super-human accuracy in classification

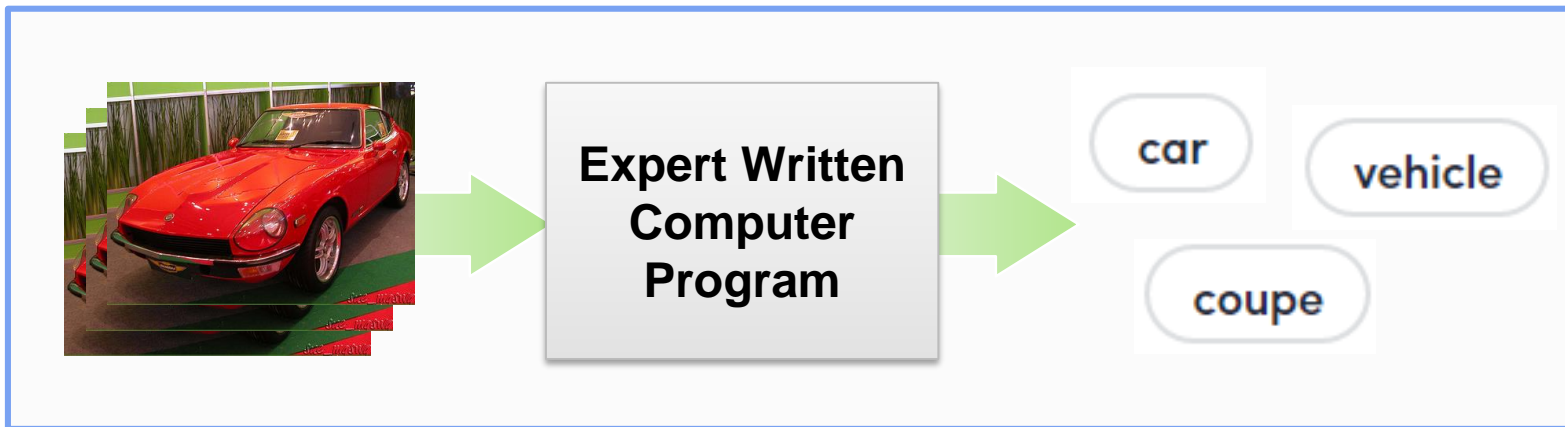


And we are getting faster fast

WHAT IS DEEP LEARNING?

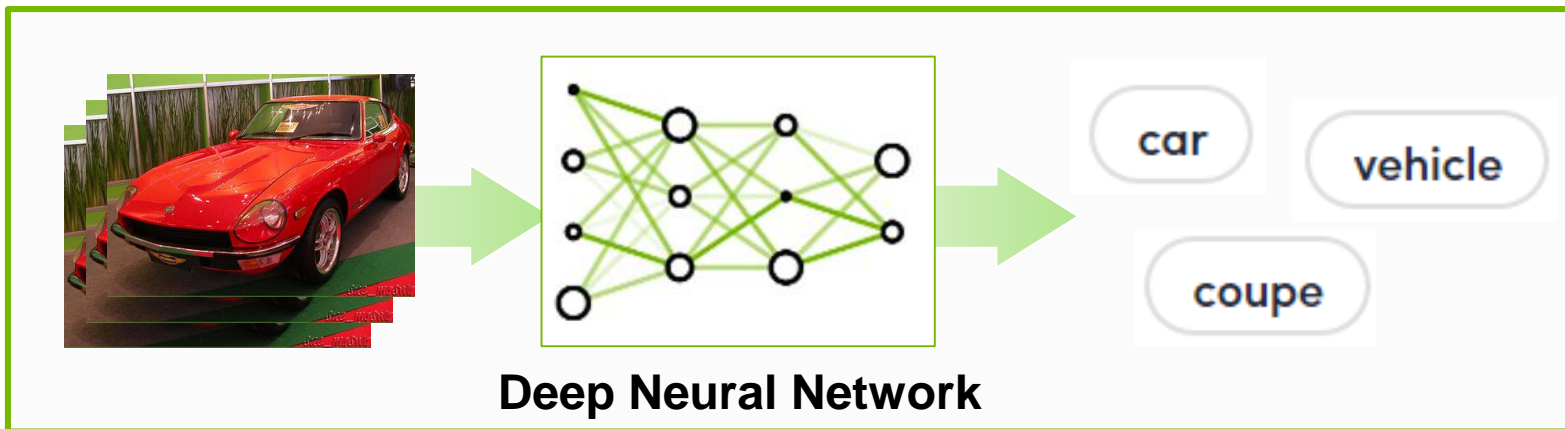
A NEW COMPUTING MODEL

Algorithms that Learn from Examples



Traditional CV Approach

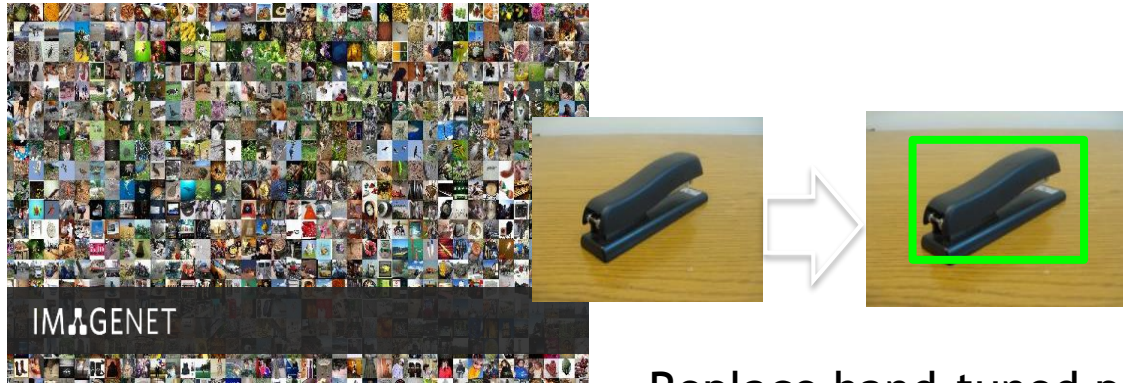
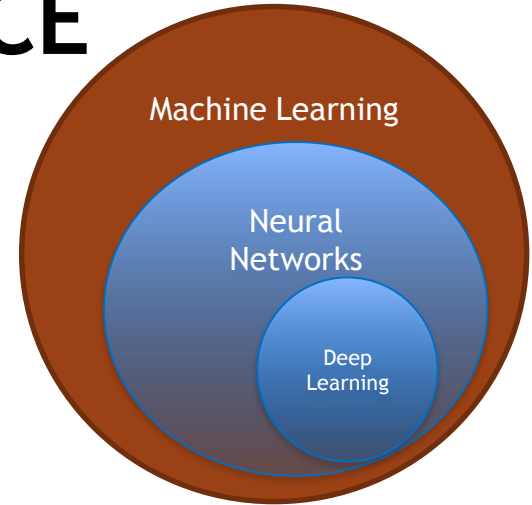
- Domain experts design feature detectors
- Time consuming
- Quality depends on Algorithms
- Error prone
- Not scalable to new problems
- Need CV experts and time



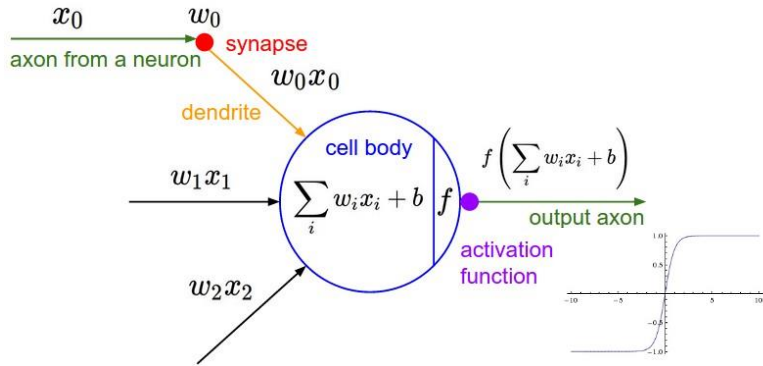
Deep Learning Approach

- 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



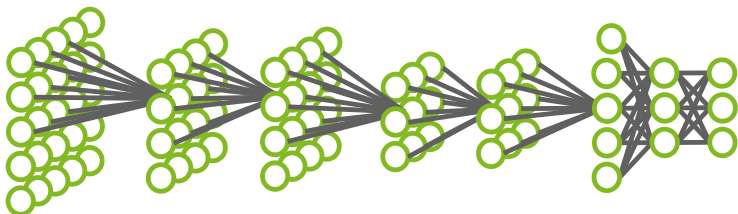
Replace hand-tuned parameters of the feature extraction steps (e.g. in voice and image recognition)



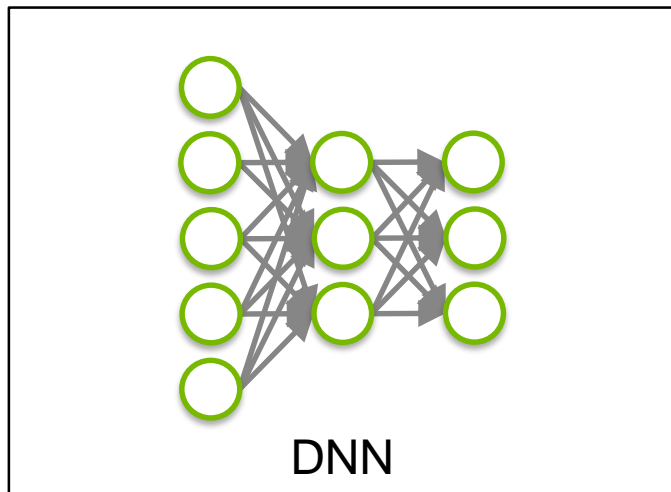
Deep learning is a subset of machine learning that refers to artificial neural networks that are composed of many layers.

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.



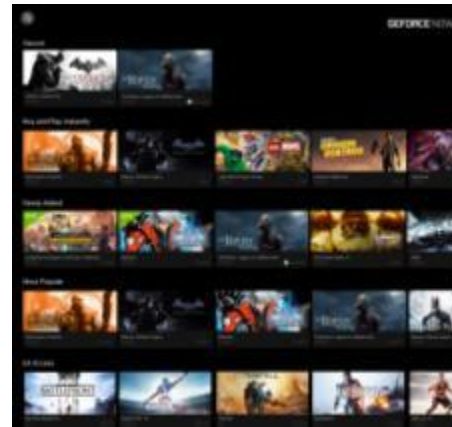
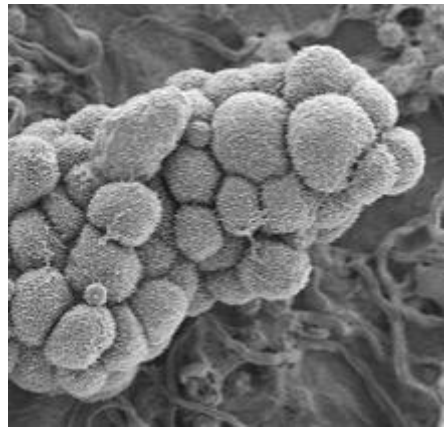
THE BIG BANG IN MACHINE LEARNING



“ 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

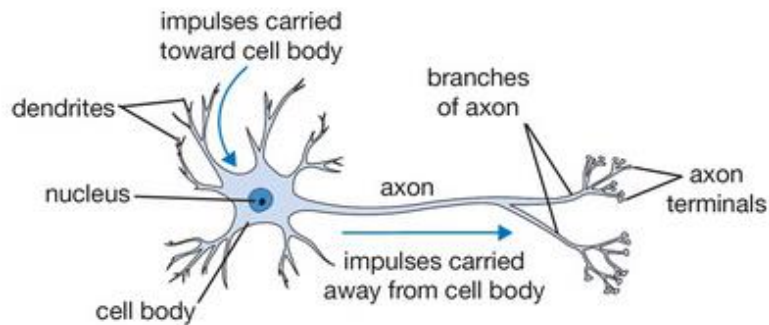
AUTONOMOUS MACHINES

Pedestrian Detection
Lane Tracking
Recognize Traffic Sign



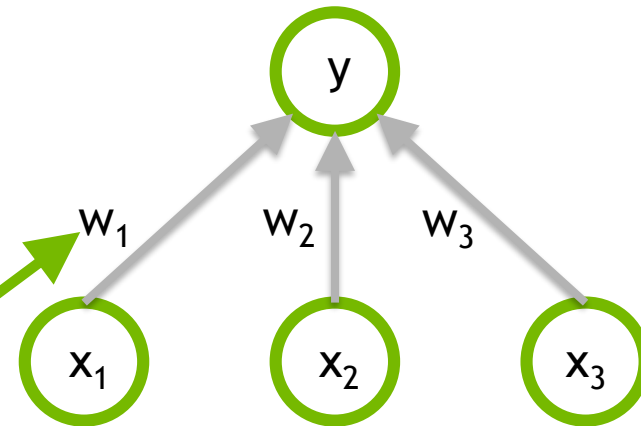
ARTIFICIAL NEURONS

Biological neuron



From Stanford cs231n lecture notes

Artificial neuron

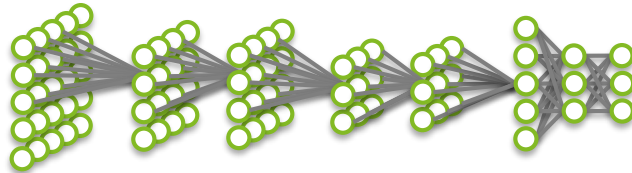


Weights (W_n)
= parameters

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

Machine Learning Software

Forward Propagation



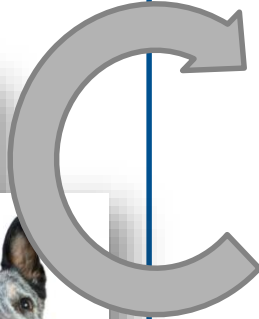
"turtle"

Backward Propagation



Compute weight update to nudge from "turtle" towards "dog"

Repeat



Tree



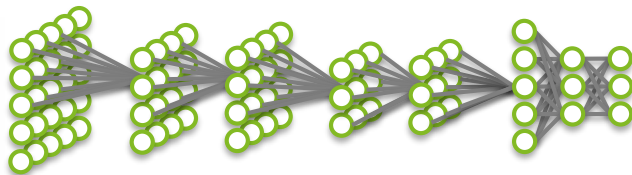
Cat



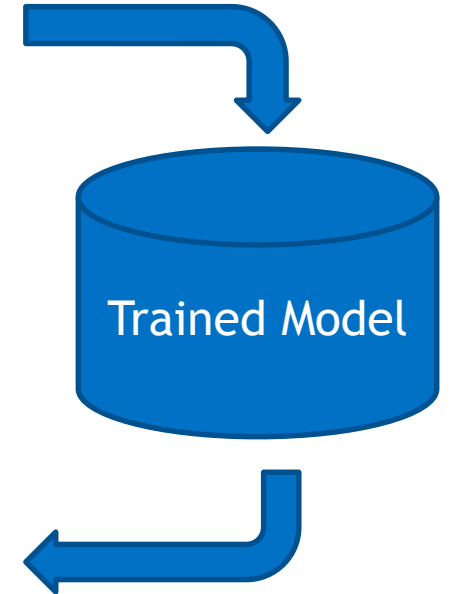
Dog

Training

Inference



"cat"



TRAINING NEURAL NETWORKS

Find a set of weights that minimizes the misfit.

Error between the target and computed output

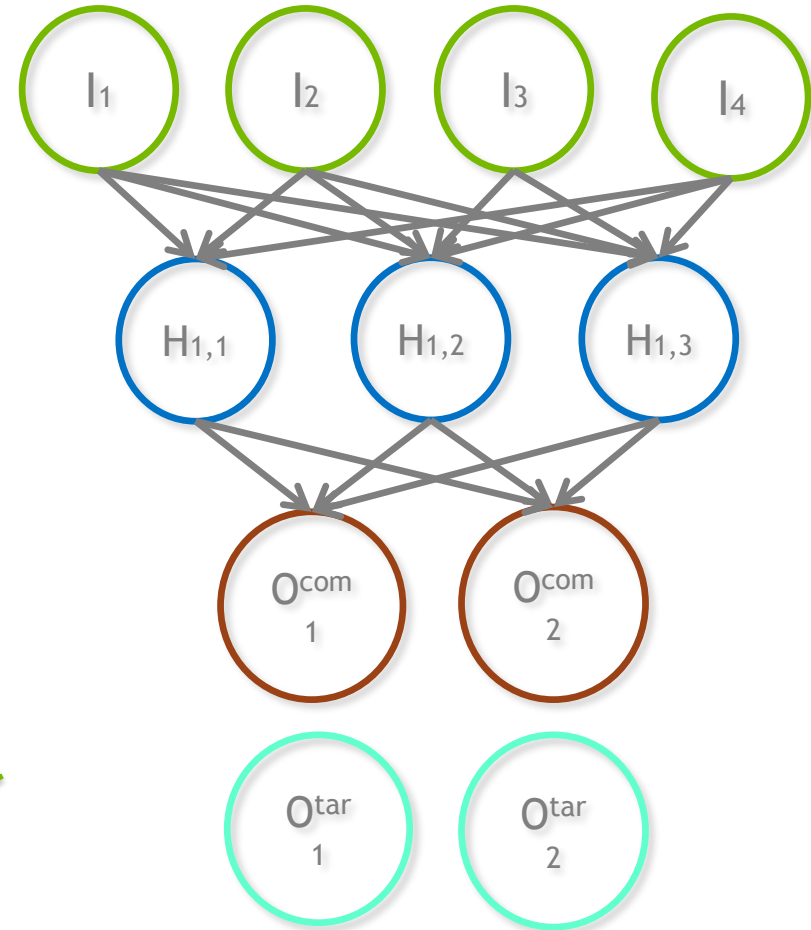
$$M(\mathbf{W}) = \sum_{i=1}^{Examples} \sum_{j=1}^{Output} (O_{comp\ i,j} - O_{target\ i,j})^2$$

Least squares optimization problem

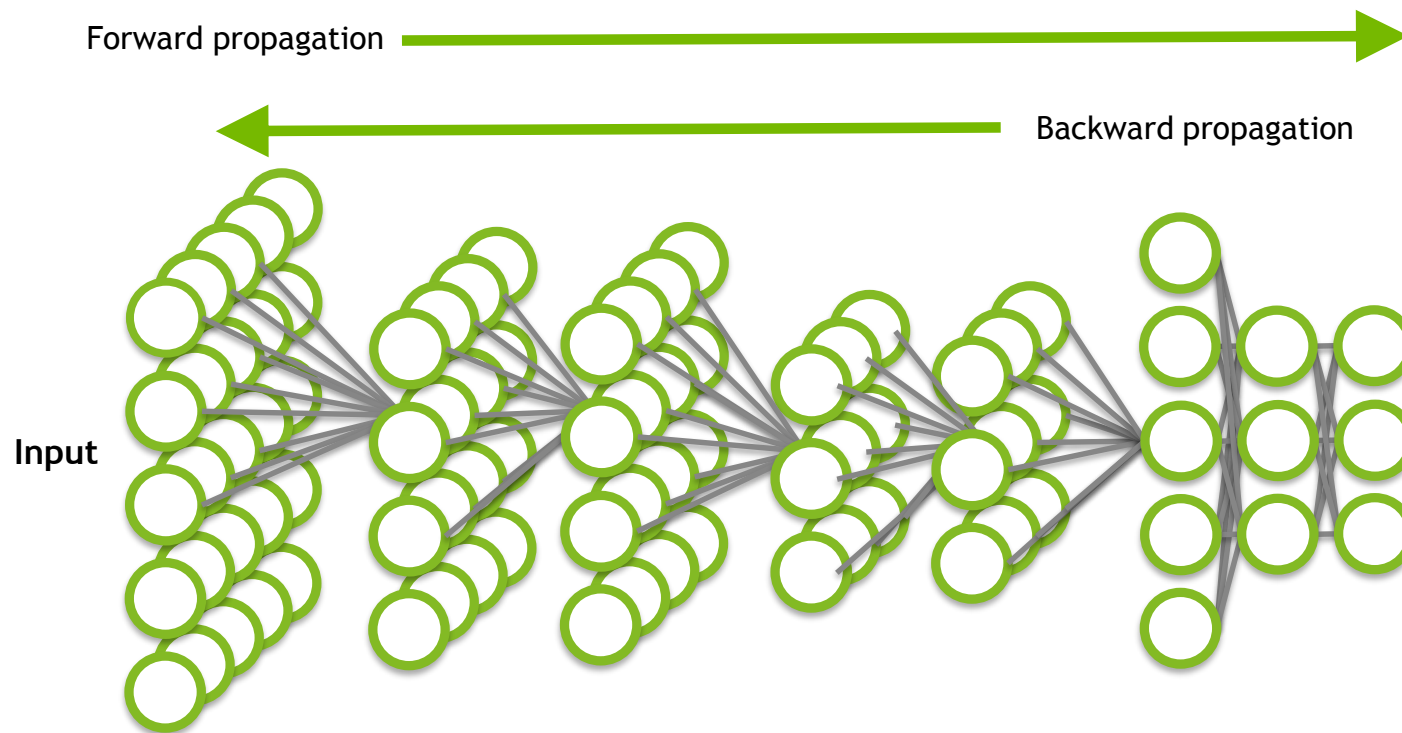
Solution by gradient descent, Monte Carlo, etc.

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

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



DEEP LEARNING APPROACH - TRAINING



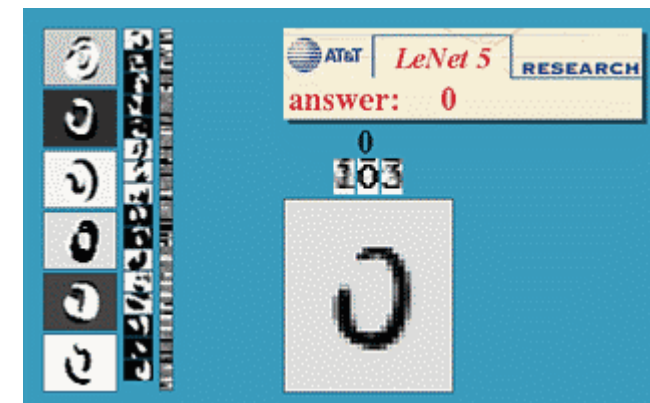
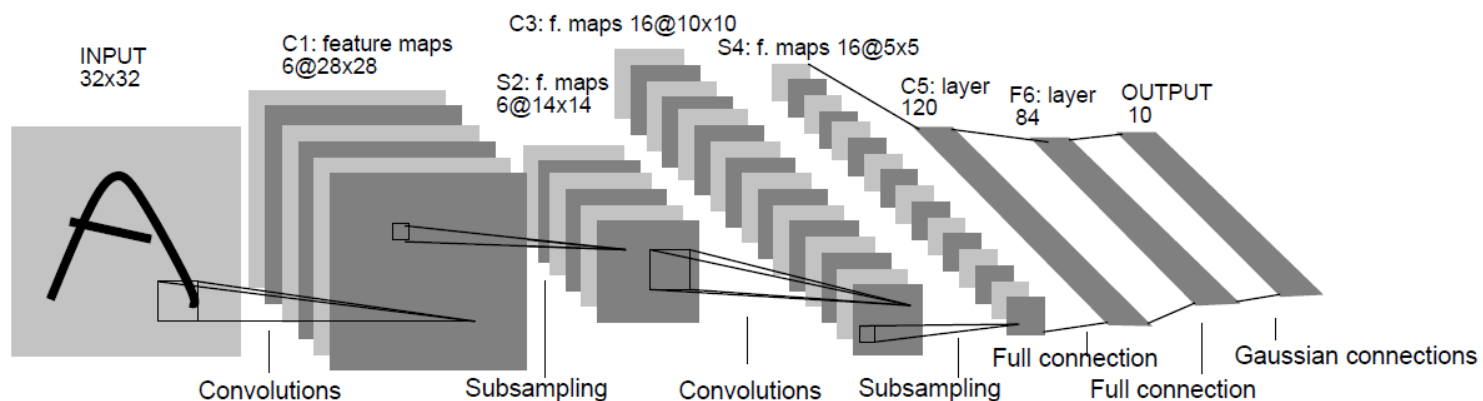
Process

- 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

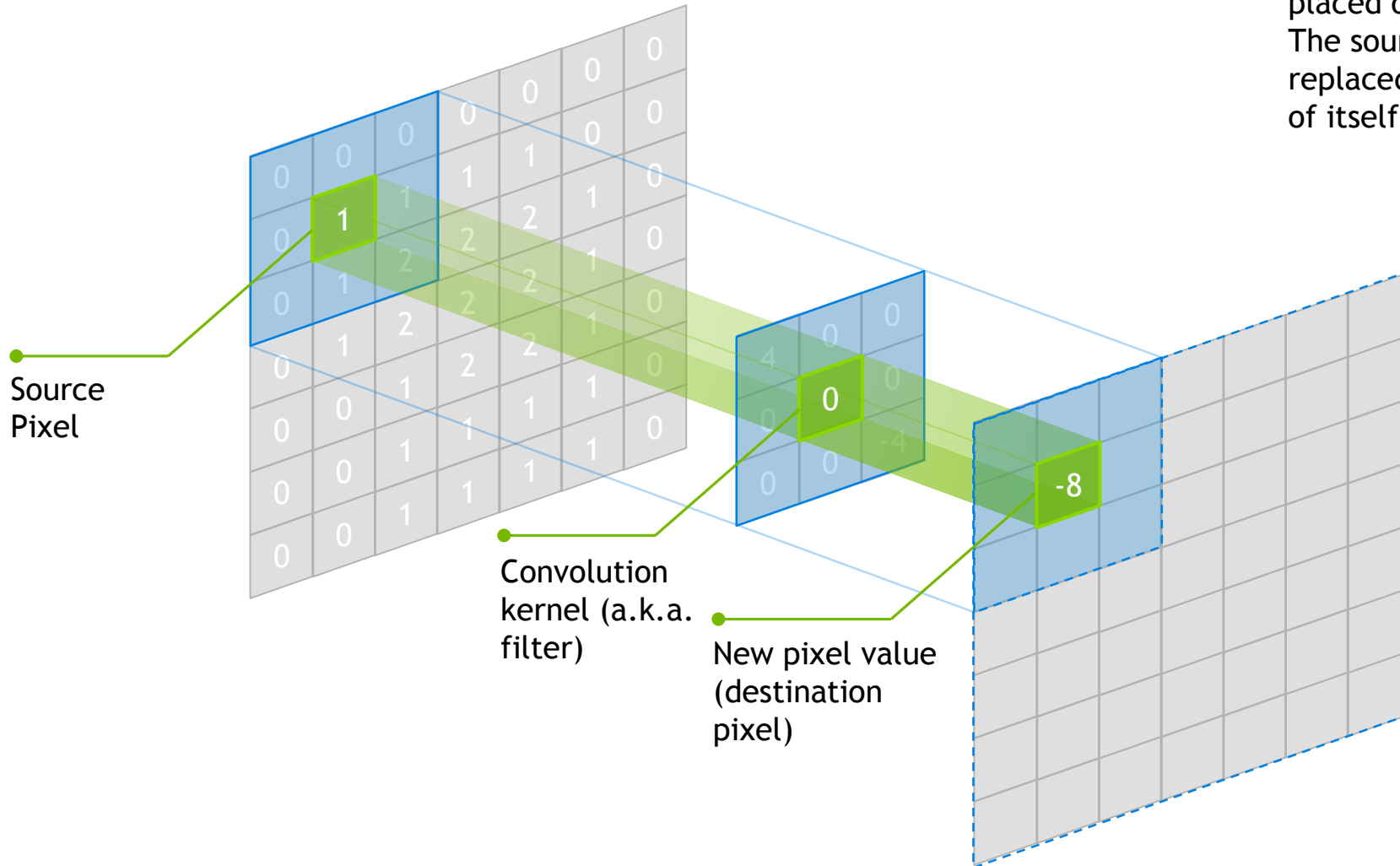
Yann LeCun et al, 1998



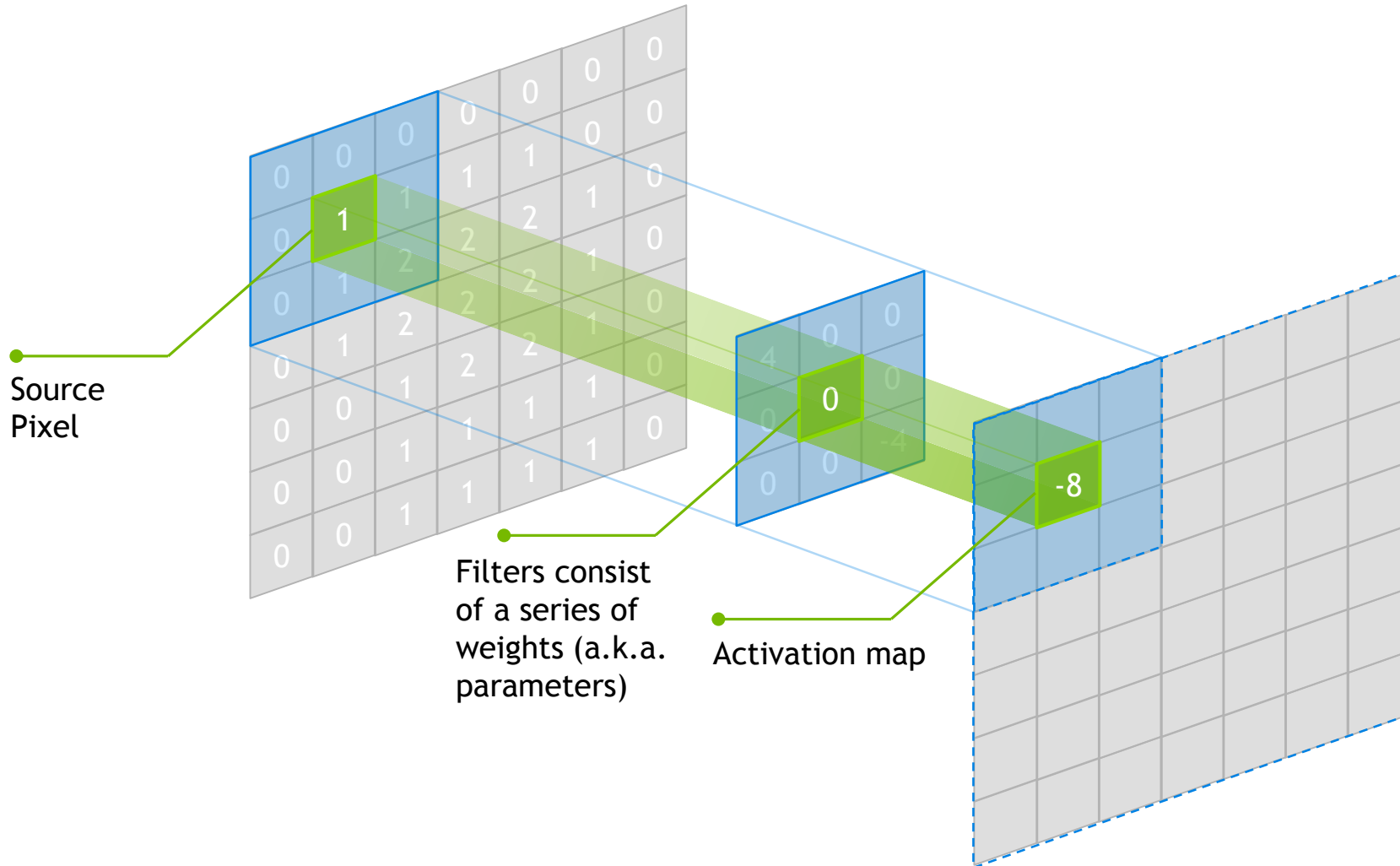
► MNIST: 0.7% error rate

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.



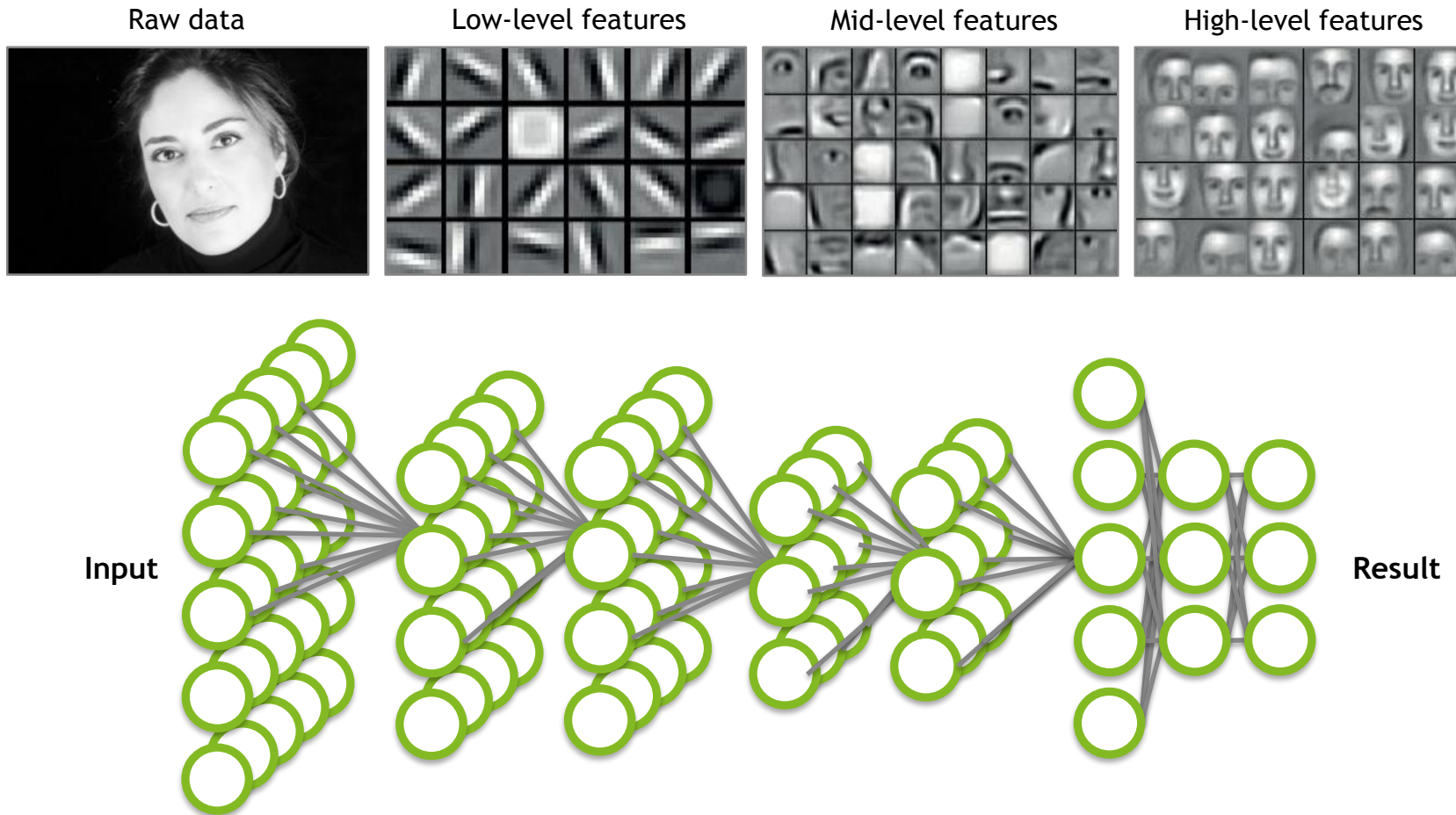
CNN TERMINOLOGY



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

The screenshot shows the DIGITS Home interface. At the top, a dark navigation bar contains the 'DIGITS' logo on the left and 'ckillam (Logout)', 'Info', and 'About' on the right. Below the navigation bar, the word 'Home' is displayed in a large font. To the right of 'Home', the text '1/1 GPU available' is shown. The main content area features a status indicator 'No Jobs Running' and a set of tabs: 'Datasets (0)', 'Models (0)', 'Pretrained Models (0)', and a 'Rectangular Snip' button. Below the tabs, there is a 'Group Jobs' section with a checked checkbox, a 'Delete' button, and a 'Group' button. A table with columns 'name', 'framework', 'status', 'elapsed', and 'submitted' is visible, currently showing 'No Models'. On the right side of the table, there is a 'New Model' button with a dropdown menu showing 'Images'. Green circles and arrows highlight the 'DIGITS' logo, the 'Home' text, the 'Datasets (0)' and 'Models (0)' tabs, and the 'New Model' button.

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

New Object Detection Dataset

Object Detection Dataset Options

Images can be stored in any of the supported file formats (.png, .jpg, .jpeg, .bmp, .ppm).

Training image folder

Label files are expected to have the .txt extension. For example if an image file is named foo.png the corresponding label file should be foo.txt.

Training label folder

Validation image folder

Validation label folder

Pad image (Width x Height)

 x

Resize image (Width x Height)

 x

Channel conversion

Minimum box size (in pixels) for validation set

Custom classes

New Image Classification Dataset

Use Image Folder Use Text Files

Image Type

Image size (Width x Height)

 x

Resize Transformation

[See example](#)

Training Images

Minimum samples per class

Maximum samples per class

% for validation

% for testing

Separate validation images folder
 Separate test images folder

DB backend

Image Encoding

Group Name

Dataset Name

[Create](#)

Different options will be presented based upon the task

DIGITS - MODEL

New Object Detection Model

Select Dataset

Python Layers

Server-side file

Use client-side file

Solver Options

Training epochs: 30

Snapshot interval (in epochs): 1

Validation interval (in epochs): 1

Random seed: [none]

Batch size: [network defaults] multiples allowed

Batch Accumulation: [empty]

Solver type: Stochastic gradient descent (SGD)

Base Learning Rate: 0.01 multiples allowed

Show advanced learning rate options

Data Transformations

Subtract Mean: Image

Crop Size: none

New Image Classification Model

Select Dataset

Python Layers

Server-side file

Use client-side file

Solver Options

Training epochs: 30

Snapshot interval (in epochs): 1

Validation interval (in epochs): 1

Random seed: [none]

Batch size: [network defaults] multiples allowed

Batch Accumulation: [empty]

Solver type: Stochastic gradient descent (SGD)

Base Learning Rate: 0.01 multiples allowed

Show advanced learning rate options

Data Transformations

Subtract Mean: Image

Crop Size: none

Define custom layers with Python

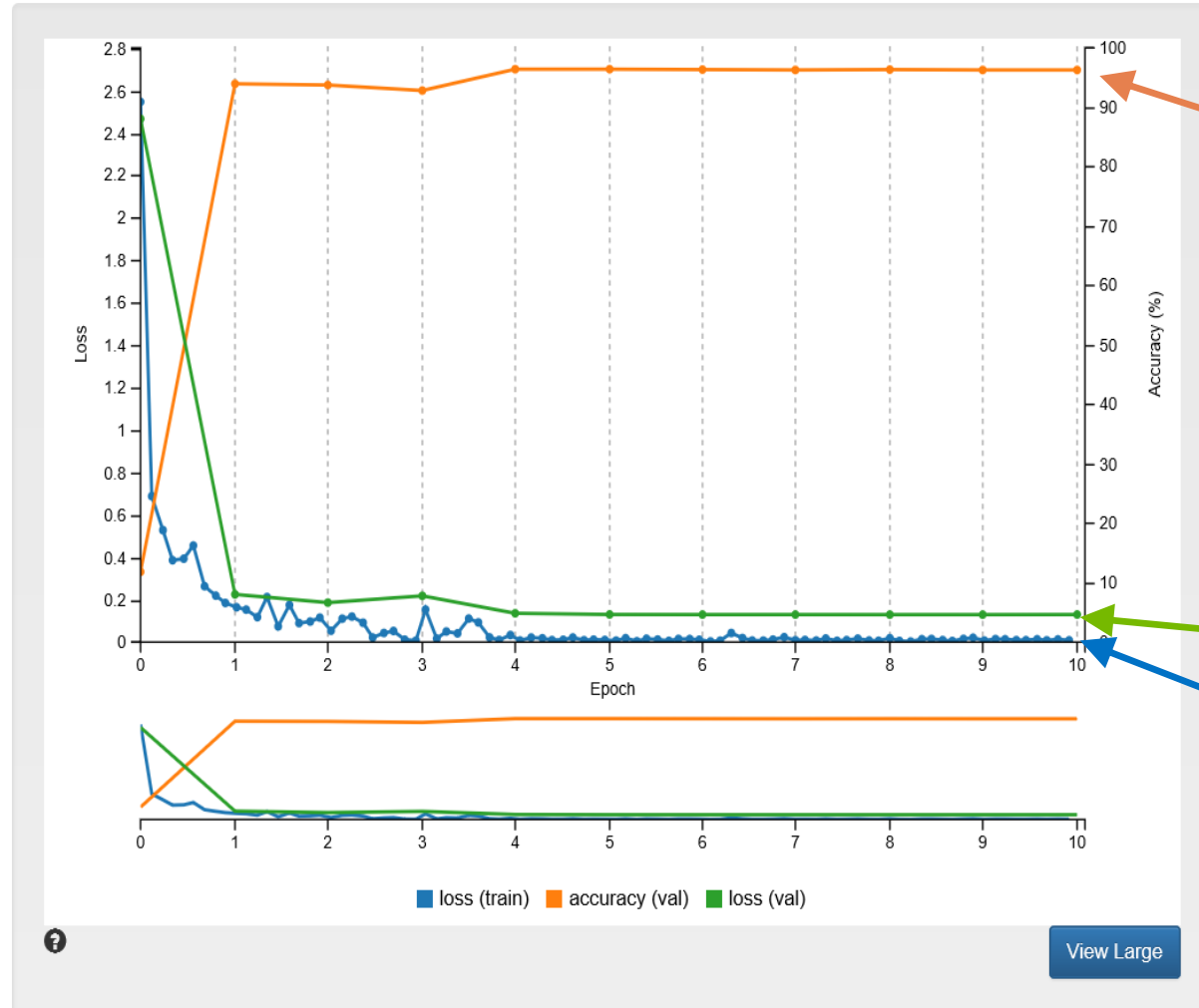
Can anneal the learning rate

Network	Details	Intended image size
LeNet	Original paper [1998]	28x28 (gray)

Network	Details	Intended image size
LeNet	Original paper [1998]	28x28 (gray)

Differences may exist between model tasks

EVALUATE THE MODEL



Accuracy
obtained from
validation dataset

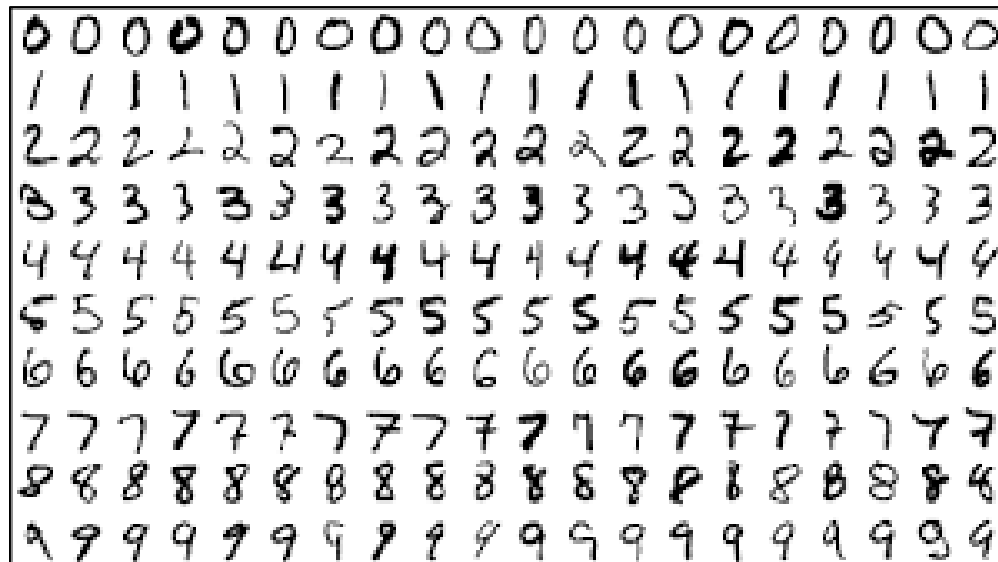
Loss function
(Validation)

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

















DIGITS Image Classification Dataset smorino (Logout) Info

Exploring MNIST invert (train_db) images

Show all images or filter by class: 0 1 2 3 4 5 6 7 8 9

Items per page: 10 - 25 - 50 - 100







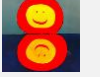
« 0 1 2 3 4 5 ... 3600 »

			
2	9	7	3
			
1	4	6	5
			
5	3	8	2
			
3	1	8	6

- $\text{Pixel}(\text{Inverted}) = 255 - \text{Pixel}(\text{original})$
- 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"
}

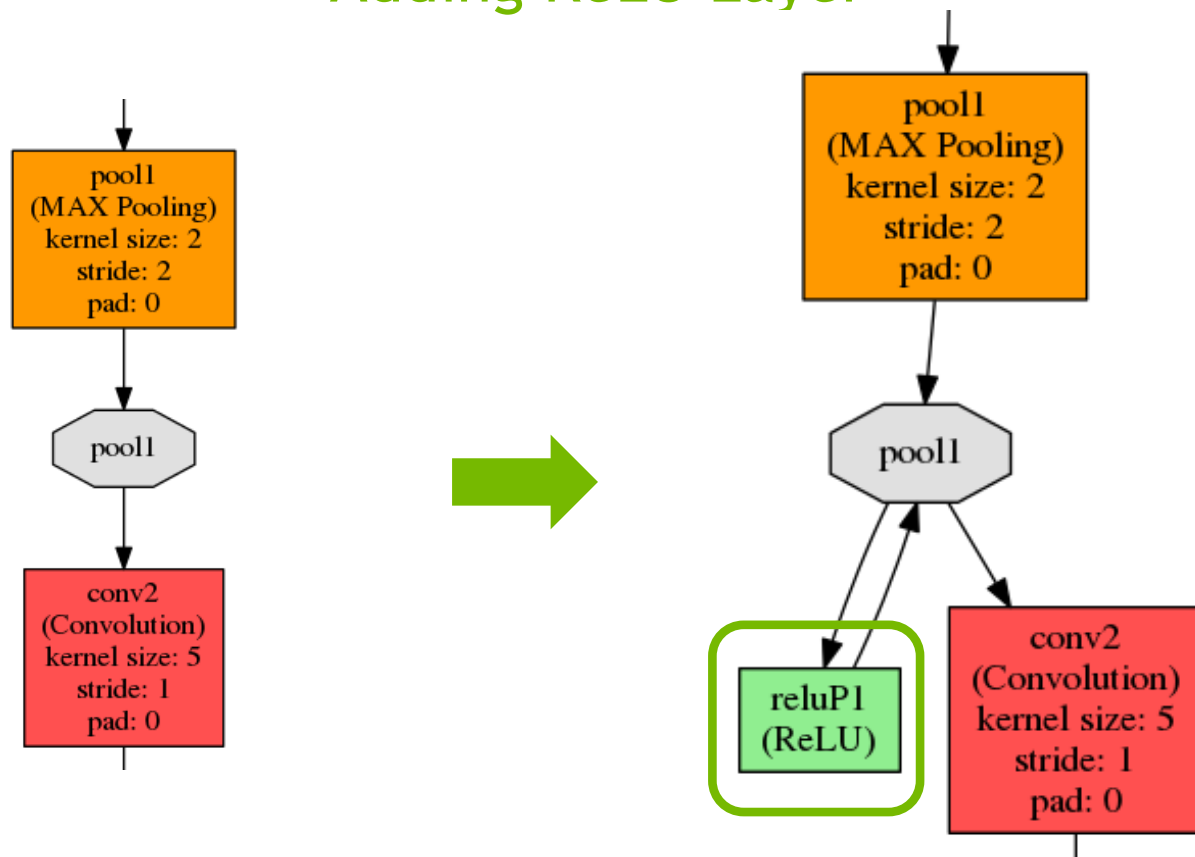
layer {
  name: "reluP1"
```

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

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







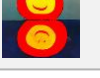
MODIFY THE NETWORK

Adding ReLU Layer



MODIFIED NETWORK

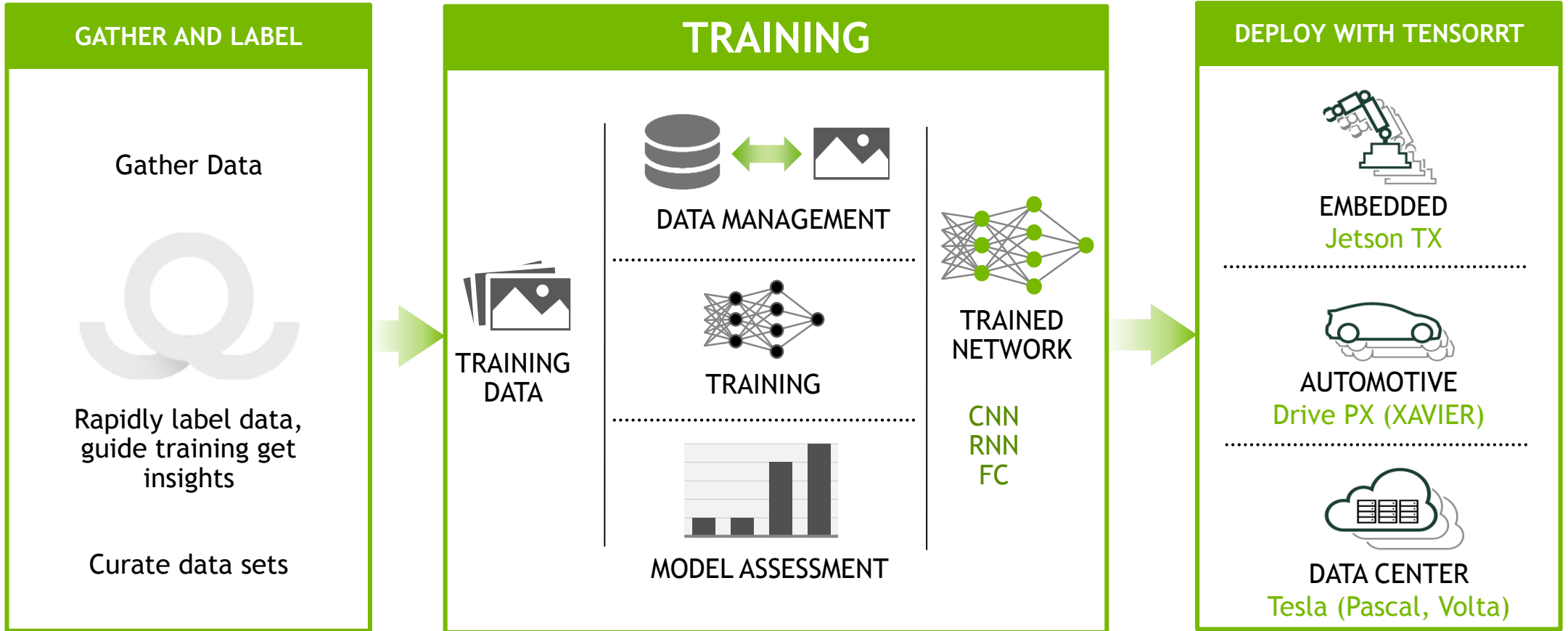
Adding filters and ReLU layer (10 epochs)

	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 %

The background features a complex network of glowing green lines and nodes. The nodes are small, bright green circles of varying sizes, and the lines are thin, semi-transparent green strands that crisscross the dark blue and black background, creating a sense of depth and connectivity. The overall aesthetic is futuristic and technical.

DEEP LEARNING SDK

NVIDIA DEEP LEARNING SOFTWARE PLATFORM



NVIDIA DEEP LEARNING SDK

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

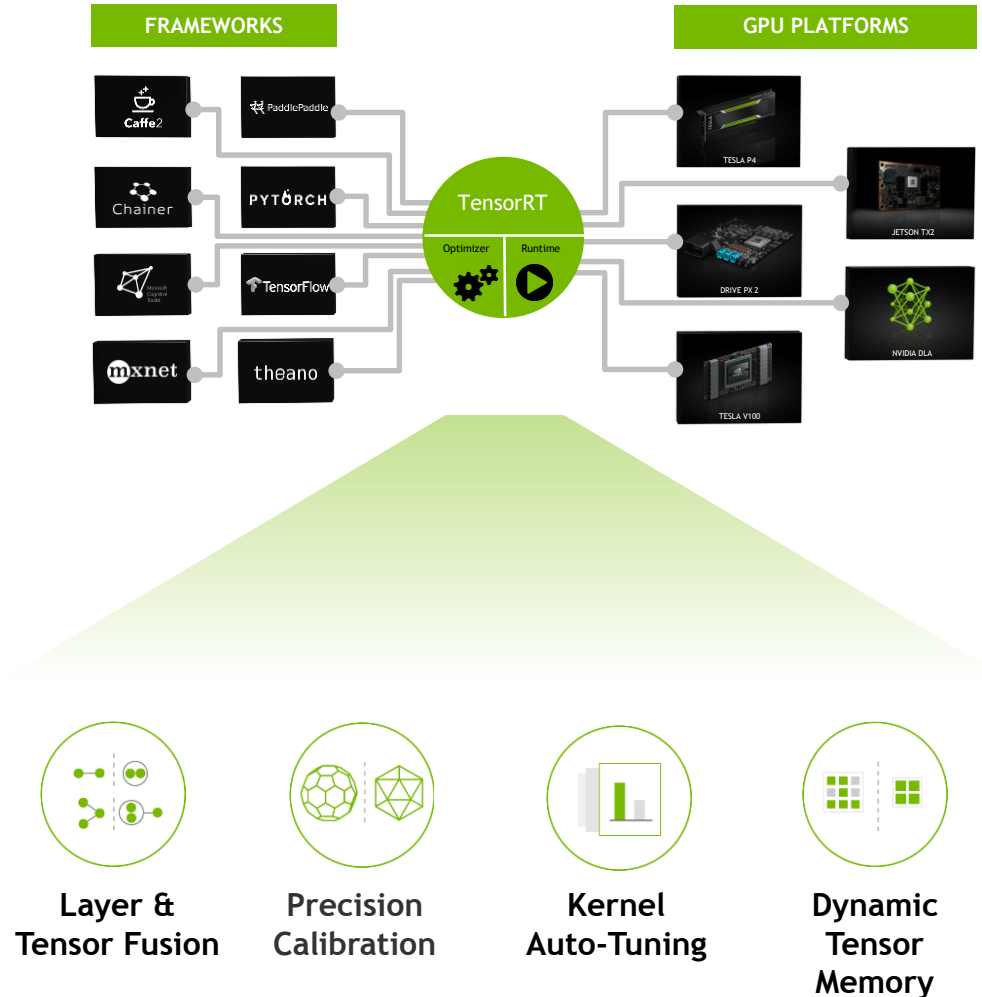
High performance neural network inference optimizer and runtime engine for production deployment

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

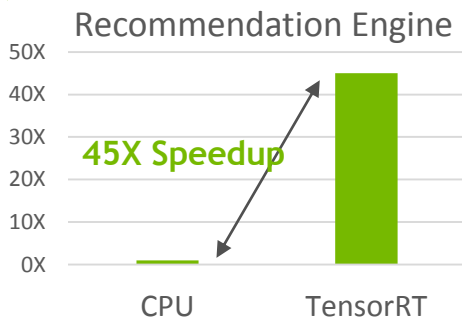
developer.nvidia.com/tensorrt



NVIDIA TENSORRT 4

RNN and MLP Layers • ONNX Import • NVIDIA DRIVE Support

Maximize RNN and MLP Throughput



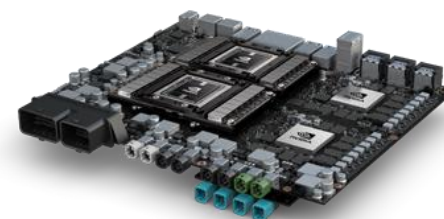
Speed up speech, audio and recommender app inference performance through new layers and optimizations

Optimize and Deploy ONNX Models



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

The background features a complex network of glowing green lines and nodes. The nodes are small, bright green circles of varying sizes, some appearing as larger, fainter bokeh-like shapes. The lines are thin, green, and crisscross the dark space, creating a sense of interconnectedness and data flow. The overall aesthetic is futuristic and technological.

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, DeepSpeech2



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



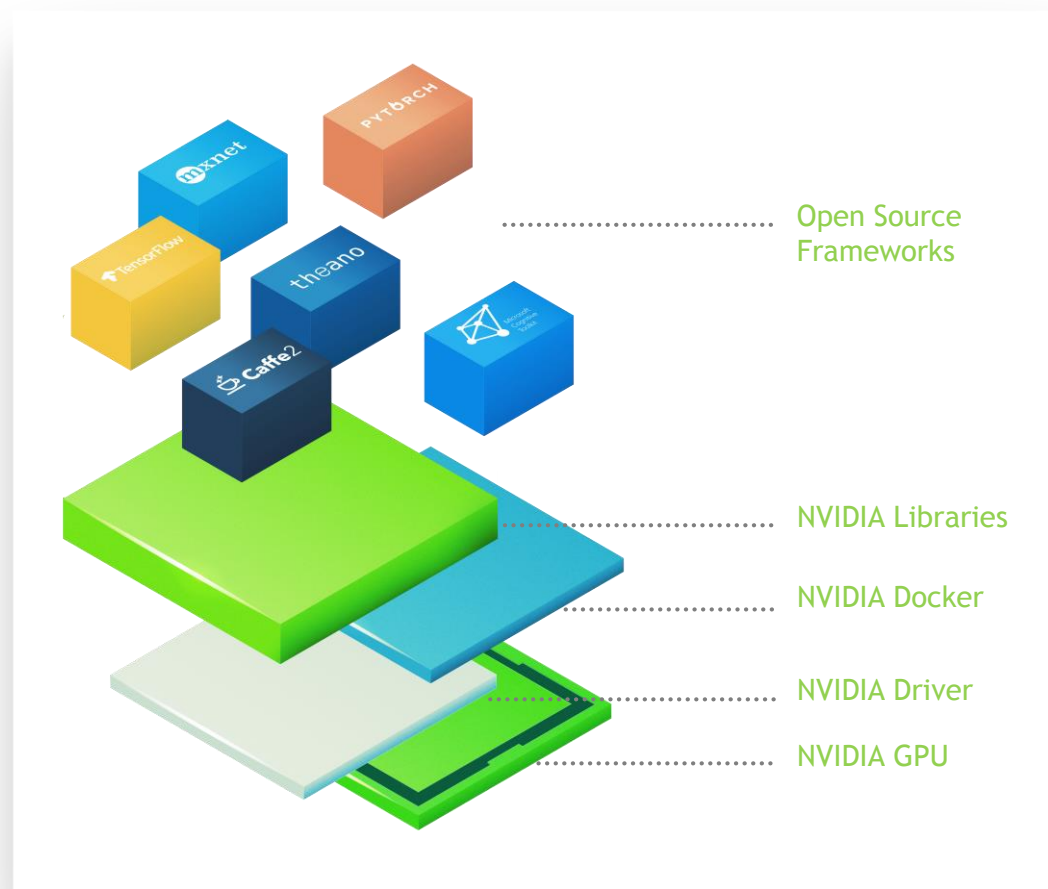
CONTAINER

CHALLENGES

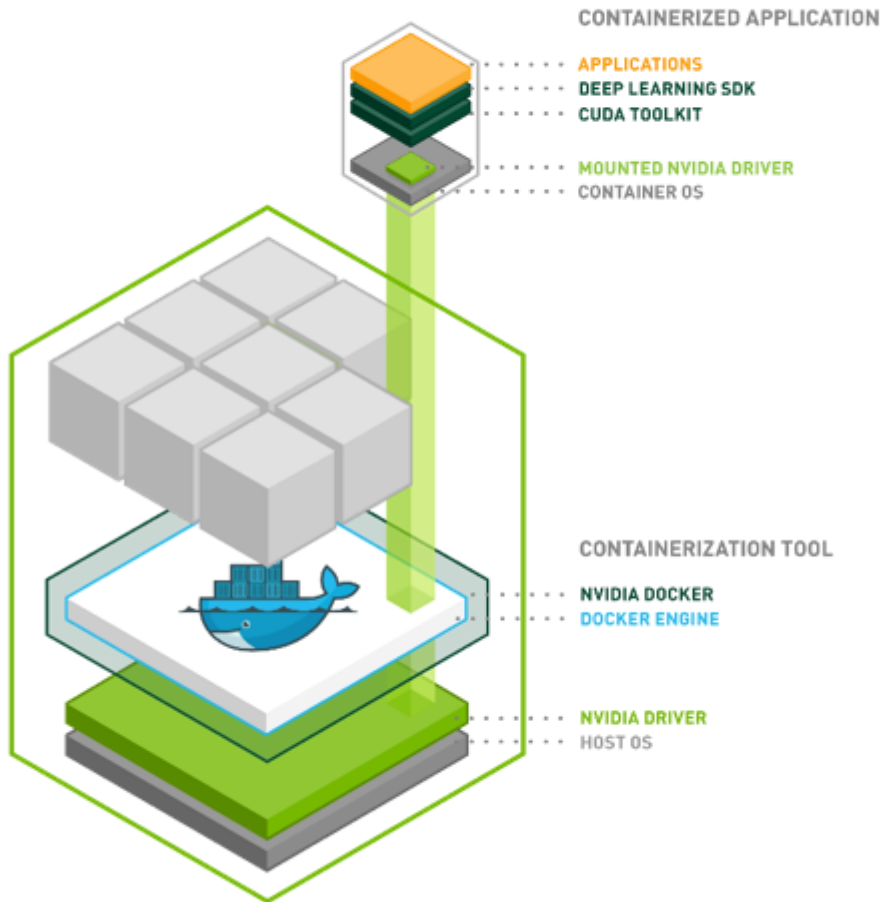
Current DIY deep learning environments are **complex** and **time consuming** to build, test and maintain

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

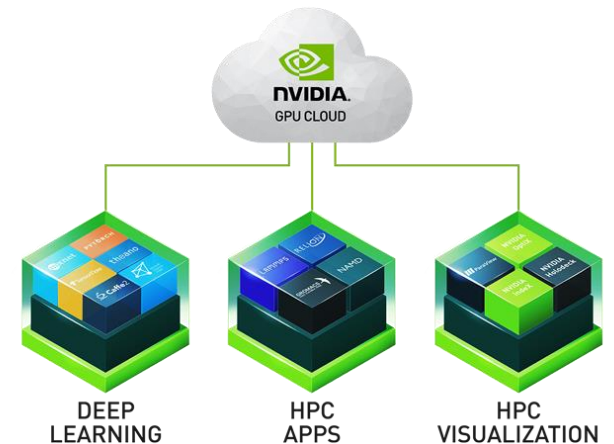
NAMD, Gromacs, LAMMPS, GAMESS, Relion, Chroma, MILC

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



AWS-EC2 | GCP | Azure | DGX

KUBERNETES

NVIDIA CONTAINER
RUNTIME

NVIDIA GPU CLOUD

NVIDIA GPUs

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?

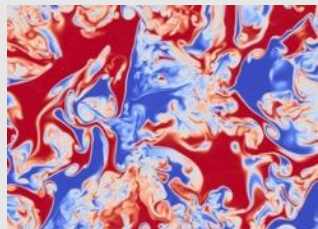


- Select/classify/augment/distribute input data
- Control job parameters

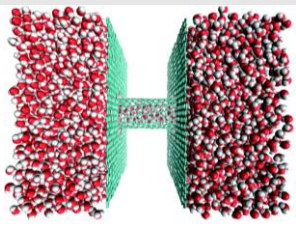
- Analyze/reduce/augment output data
- Act on output data

AI Supercomputing is The New computing model

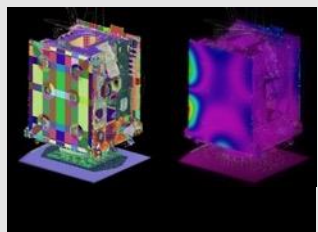
Extending The Reach of HPC By Combining Computational & Data Science



Turbulent Flow



Molecular Dynamics



Structural Analysis

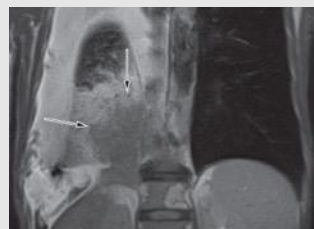


N-body Simulation

COMPUTATIONAL SCIENCE



“What’s happening?”



“Is there cancer?”

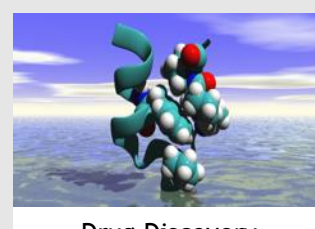


“Next move?”



“What does she mean?”

DATA SCIENCE



Drug Discovery



Clean Energy



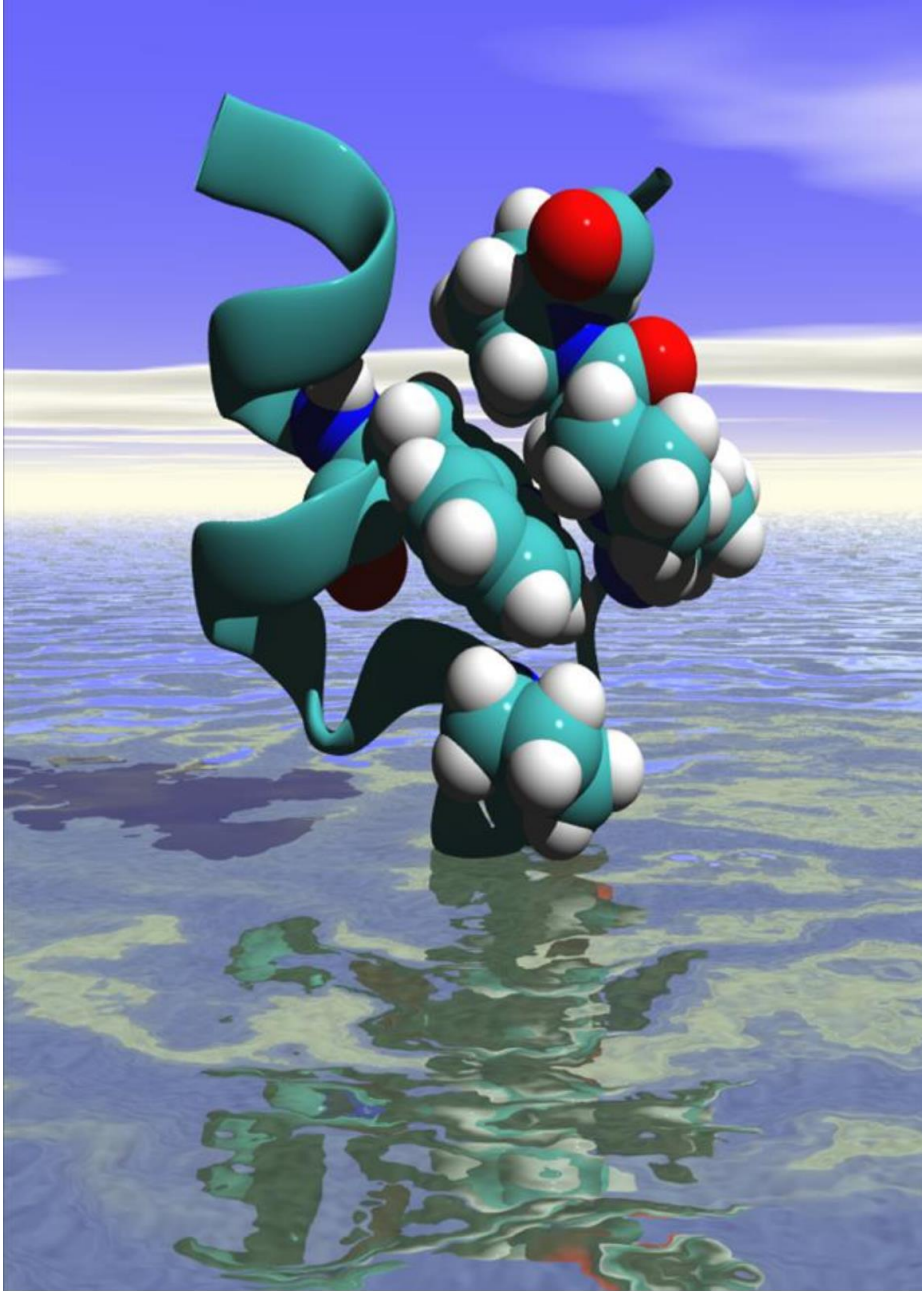
Understanding Universe



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.

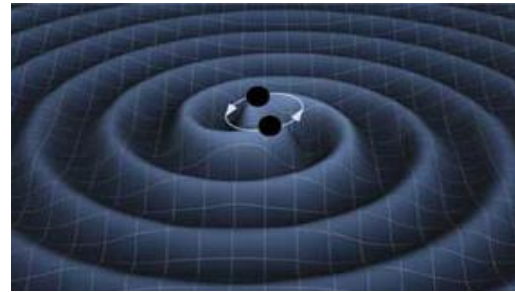
Essentially the DL model is trained to learn Hamiltonian of the Schrodinger equation.

Impact

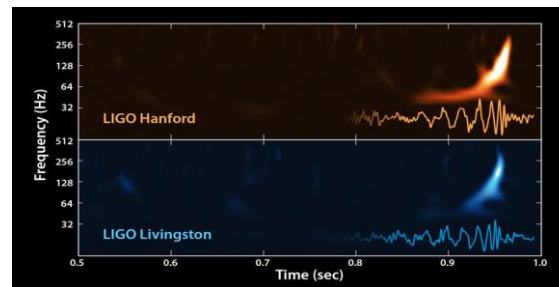
Faster, more accurate screening at far lower cost

DEEP LEARNING FOR GRAVITATIONAL WAVE DETECTION

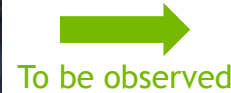
Deep learning method named deep filtering was used in the first detection of gravitational wave. Numerical simulated data was used for training deep filtering, a convolutional neural network to replace matched filtering. It provided 20X speed up on single core and potential to be accelerated further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



Gravitational wave due to black hole collide and merge



Actual Signal Caused by Gravitational Wave



To be observed



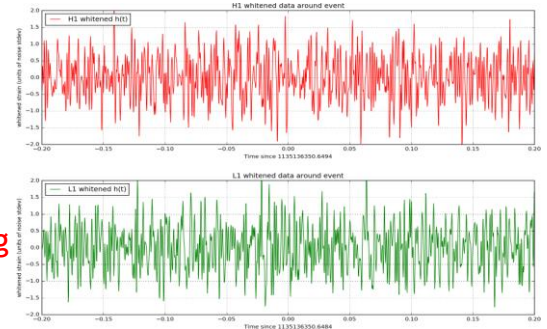
LIGO facility



How to find The signal???



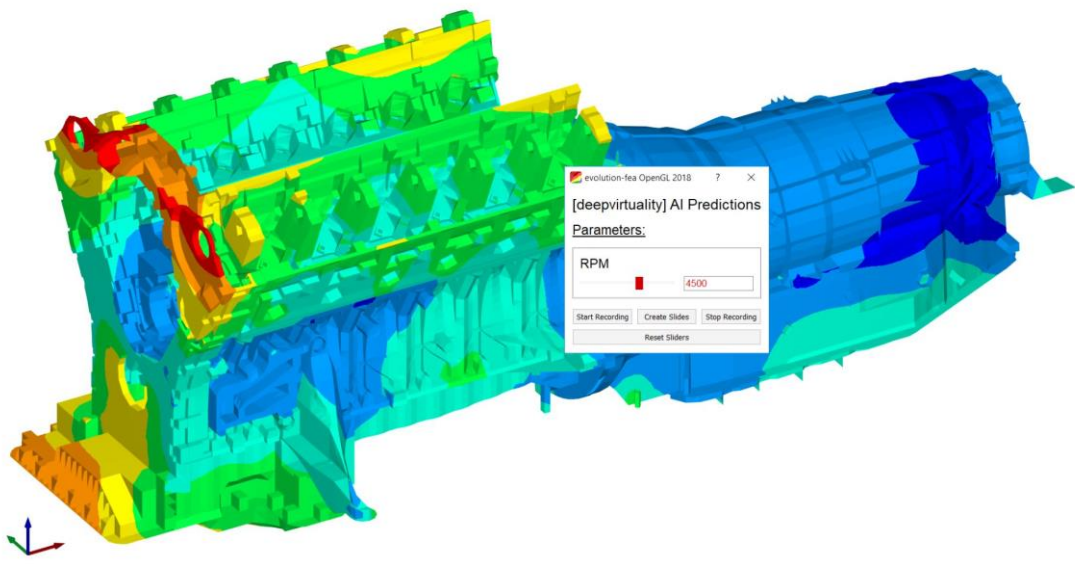
Deep Learning



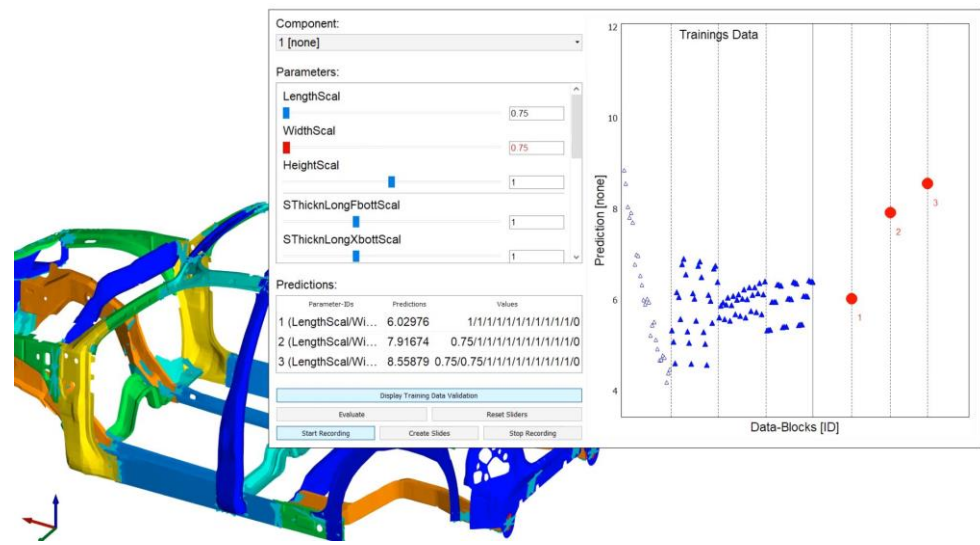
Actual observed data

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.

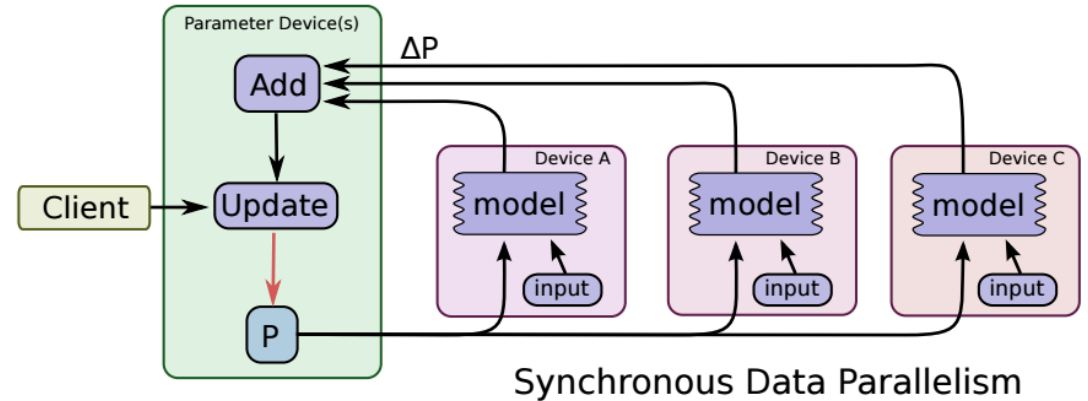


An demonstration of Structure Born Noise of a V12 Engine with Deepvirtuality



Torsional Frequencies of a Car Body by Deepvirtuality

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.”

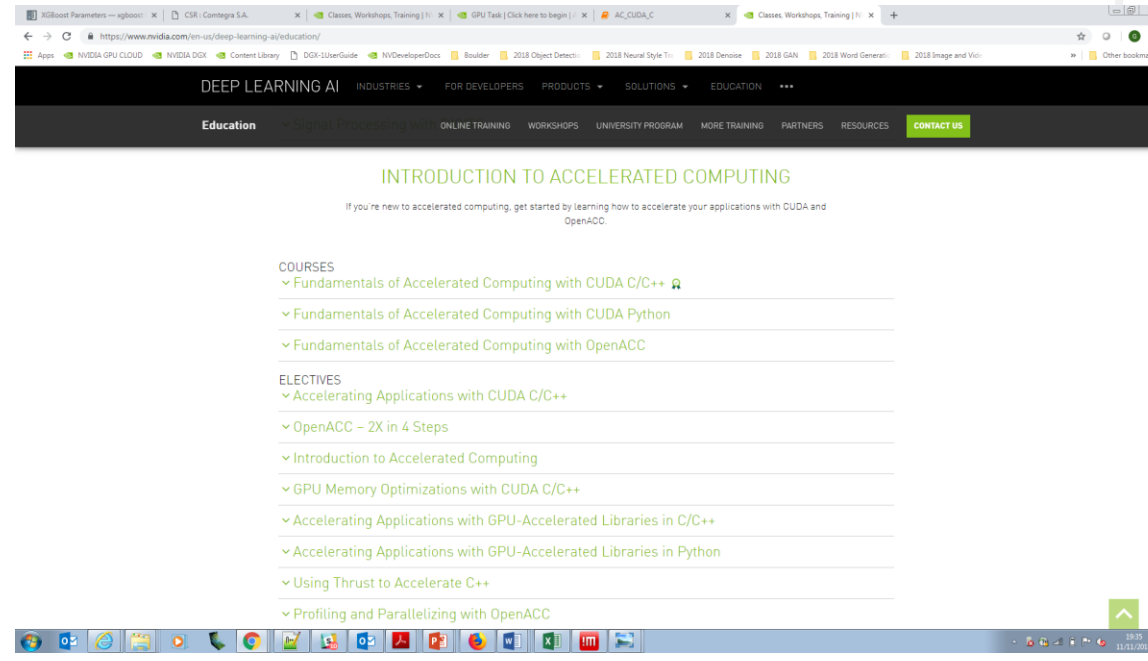
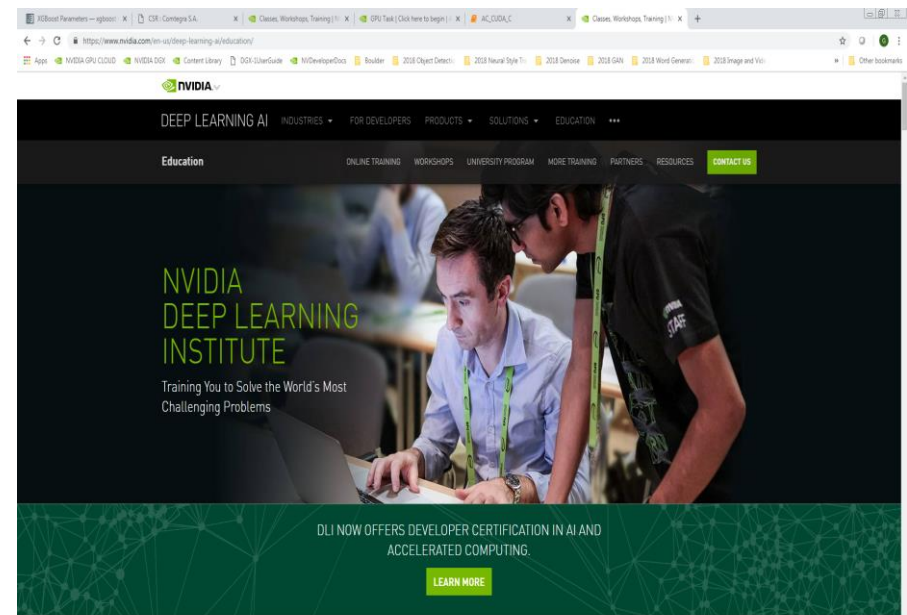
Leverage Tensorflow + MPI + NCCL2 for a simplified and performant API to enable synchronous multigpu + multinode Tensorflow.

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®