Introducing NVIDIA® cuDNN

Sharan Chetlur, Software Engineer,
CUDA Libraries and Algorithms Group

Agenda

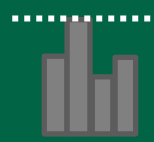
- What are CNNs?
- GPUs & CNNs
- Multi-convolve: the computational workhorse
- cuDNN
- Implementation details
- Results
- Roadmap and questions

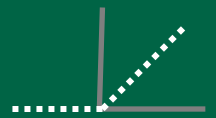
Neural Networks, briefly

- Interpret AI task as the evaluation of complex function
  - Facial Recognition: Map a bunch of pixels to a name
  - Handwriting Recognition: Image to a character
- Neural Network: Network of interconnected simple "neurons"
- Neuron typically made up of 2 stages:
  - Linear Transformation of data
  - Point-wise application of non-linear function
- In a CNN, Linear Transformation is a convolution

CNNs: Stacked Repeating Triplets

Convolution → Max-Pooling (block-wise max) → Activation (point-wise ReLU)



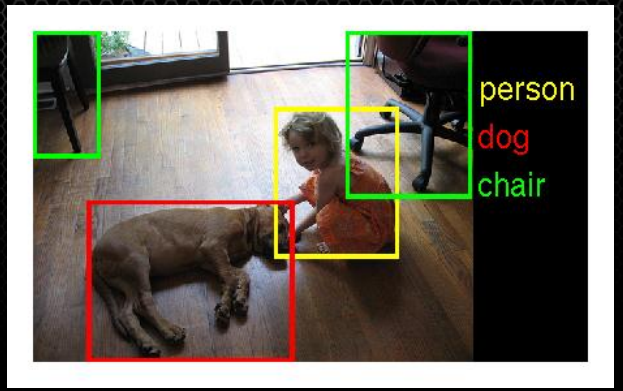


OverFeat Network, 2014

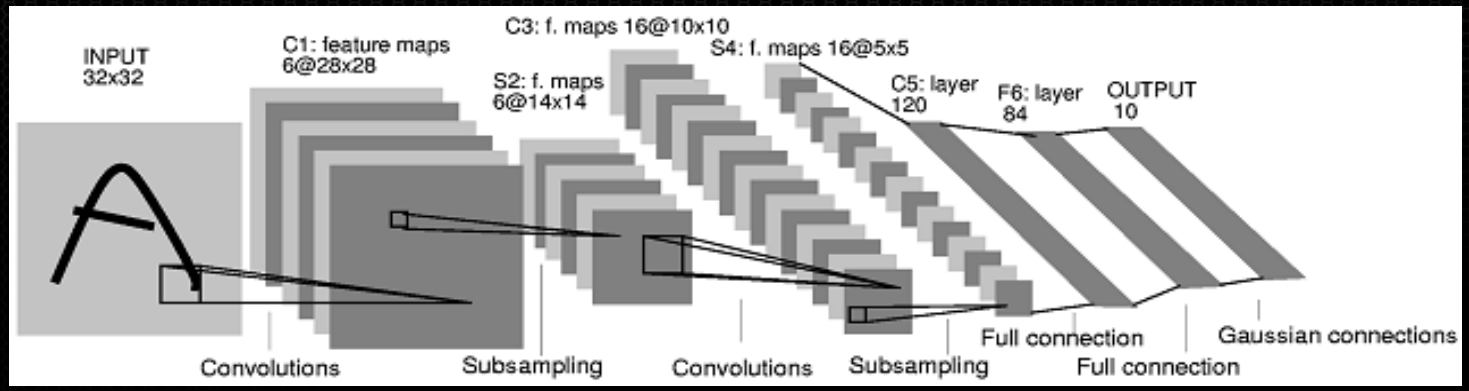
Layer	1	2	3	4	5	6	7	8	Output 9
Stage	conv + max	conv + max	conv	conv	conv	conv + max	full	full	full
# channels	96	256	512	512	1024	1024	4096	4096	1000
Filter size	7x7	7x7	3x3	3x3	3x3	3x3	-	-	-
Conv. stride	2x2	1x1	1x1	1x1	1x1	1x1	-	-	-
Pooling size	3x3	2x2	-	-	-	3x3	-	-	-
Pooling stride	3x3	2x2	-	-	-	3x3	-	-	-
Zero-Padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input size	221x221	36x36	15x15	15x15	15x15	15x15	5x5	1x1	1x1

Sample applications

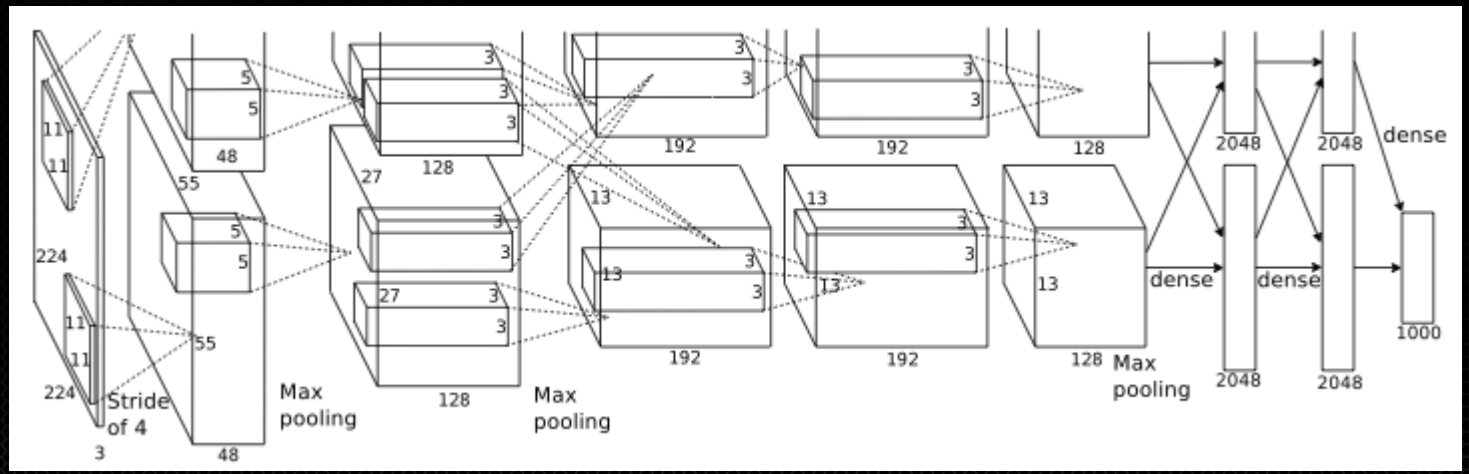
- Image
  - Classification
  - Localization
  - Segmentation
- Audio
  - Speech recognition
  - Translation



Convolutional Networks breakthrough



Y. LeCun et al. 1989-1998 : Handwritten digit reading



A. Krizhevsky, G. Hinton et al. 2012 : Imagenet classification winner

NVIDIA

GPUs for Deep Learning

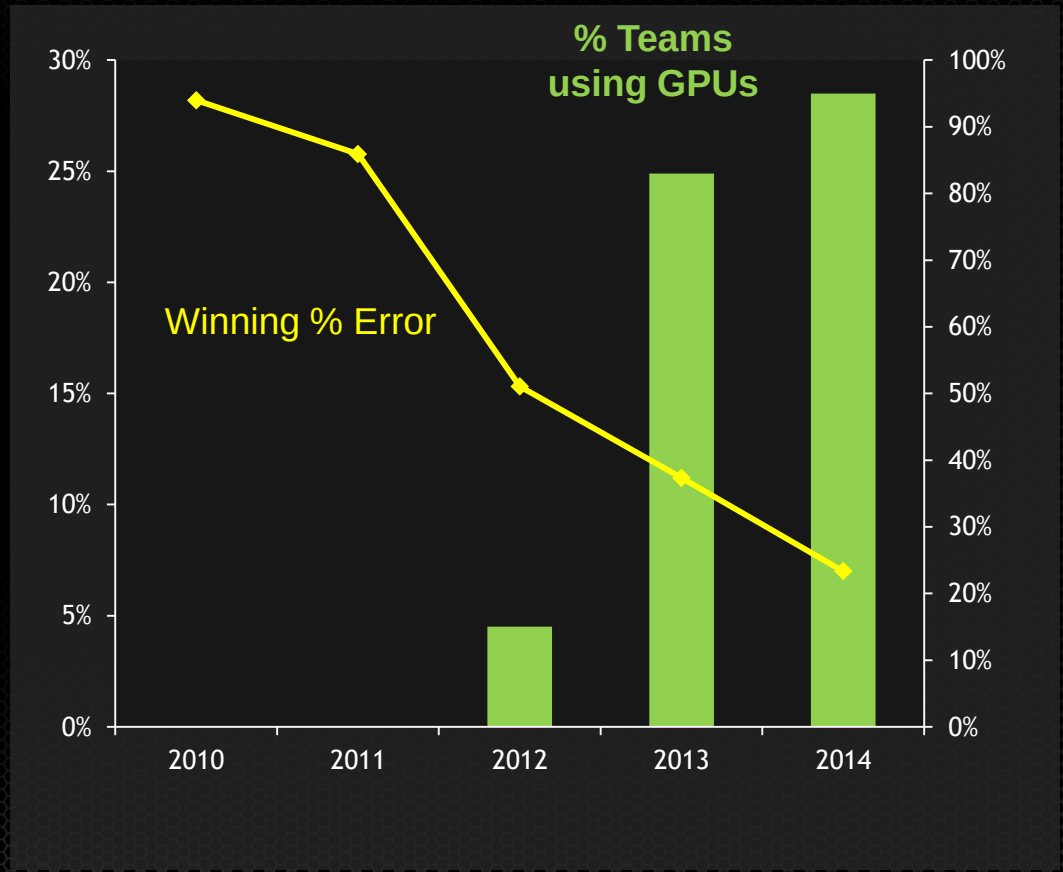
Image Recognition CHALLENGE

1.2M training images • 1000 object categories

Hosted by

IMAGENET

GPU usage for ILSVRC



NVIDIA

More on the topic …

- Convolutional Networks: A Unified Machine Learning Approach to Computer Perception (Yann LeCun, Facebook/NYU) – Video, PDF
- Large Scale Training of Deep-networks on Distributed GPUs (Ian Lane, CMU) – Video
- HYDRA – A Hybrid CPU/GPU Speech Recognition Engine for Real-Time LVCSR (Jungsuk Kim, CMU) – Video, PDF
- GPU Accelerated Model Combination for Robust Speech Recognition and Keyword Search (Wonkyum Lee, Carnegie Mellon University) – Video, PDF
- Rapid Training of Acoustic Models Using GPUs (Jike Chong, CMU) – Video
- GPU-Optimized Deep Learning Networks for Automatic Speech Recognition (Jessica Ray, MIT Lincoln Laboratory) – Video
- Visual Object Recognition Using Deep Convolutional Neural Networks (Rob Fergus, Facebook/NYU) – Video
- 10 Billion Parameter Neural Networks in Your Basement (Adam Coates, Stanford University) – Video, PDF
- Deep Neural Networks for Visual Pattern Recognition (Dan Ciresan, IDSIA) – Video
- Clarifai: Enabling Next Generation Intelligent Applications (Matthew Zeiler, Clarifai) – Video
- Beyond Pedestrian Detection: Deep Neural Networks Level-Up Automotive Safety (Ikuro Sato, Hideki Niihara, Denso IT Laboratory) – Video, PDF
- Using GPUs to Accelerate Learning to Rank (Alexander Shchekalev, Yandex) – Video, PDF

Multi-convolve overview

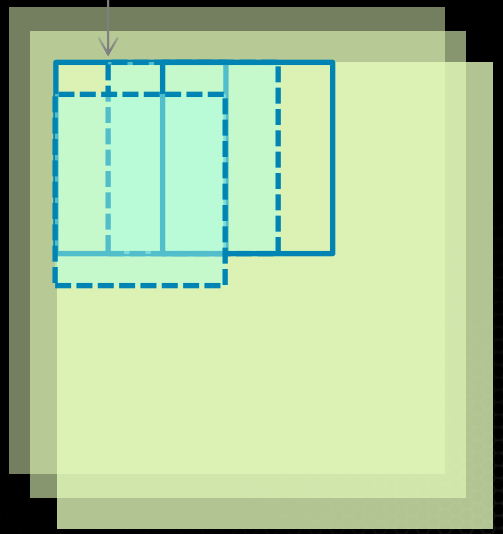
- Linear Transformation part of the CNN neuron
  - Main computational workload
  - 80-90% of execution time
- Generalization of the 2D convolution (a 4D tensor convolution)
- Very compute intensive, therefore good for GPUs
- However, not easy to implement efficiently

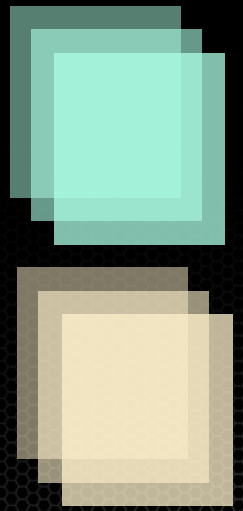
Multi-convolve, pictorially

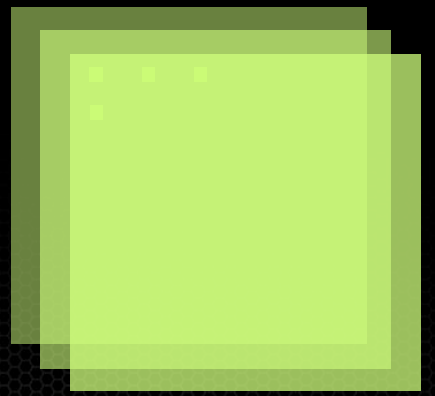
Pointwise multiply and sum, scalar output

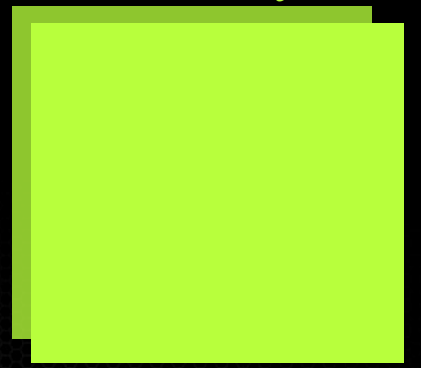
$$int[c,p,q] = \sum_{i,j \in filter} Im[c][istart+i, jstart+j] \,.\, Filt[c][i,j]$$

$$output[p,q] = \sum_{c} int[c,p,q]$$









Input Image

Input Filter

Intermediate output

Final Output

Why do it once if you can do it n times ? Batch the whole thing.

cuDNN

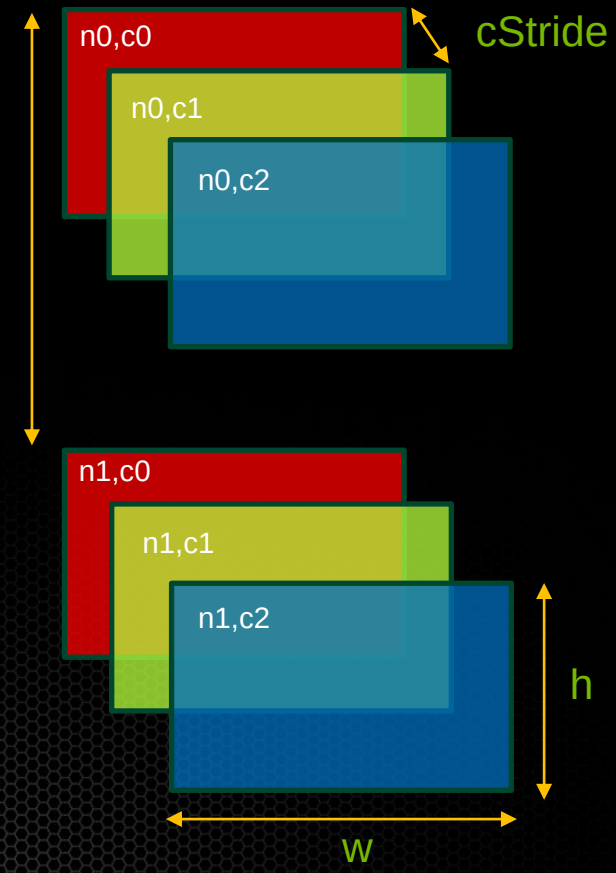
- Low-level Library of GPU-accelerated routines; similar in intent to BLAS
- Out-of-the-box speedup of Neural Networks
- Developed and maintained by NVIDIA
- Optimized for current and future NVIDIA GPU generations
- First release focused on Convolutional Neural Networks

cuDNN Features

- Flexible API : arbitrary dimension ordering, striding, and sub-regions for 4d tensors
- Less memory, more performance : Efficient forward and backward convolution routines with zero memory overhead
- Easy Integration : black box implementation of convolution and other routines – ReLu, Sigmoid, Tanh, Pooling, Softmax

Tensor-4d

- Image Batches described as 4D Tensor [n, c, h, w] with stride support [nStride, cStride, hStride, wStride]
- Allows flexible data layout
- Easy access to subsets of features (Caffe's "groups")
- Implicit cropping of sub-images
- Plan to handle negative strides - allows implicit mirroring of images



Example - OverFeat Layer 1

```
/* Allocate memory for Filter and ImageBatch, fill with data */
cudaMalloc( &ImageInBatch , ... );
cudaMalloc( &Filter , ... );
...

/* Set descriptors */
cudnnSetTensor4dDescriptor( InputDesc, CUDNN_TENSOR_NCHW, 128, 96, 221, 221);
cudnnSetFilterDescriptor( FilterDesc, 256, 96, 7, 7 );
cudnnSetConvolutionDescriptor( convDesc, InputDesc, FilterDesc,
    pad_x, pad_y, 2, 2, 1, 1, CUDNN_CONVOLUTION);

/* query output layout */
cudnnGetOutputTensor4dDim(convDesc, CUDNN_CONVOLUTION_FWD, &n_out, &c_out, &h_out, &w_out);

/* Set and allocate output tensor descriptor */
cudnnSetTensor4dDescriptor( &OutputDesc, CUDNN_TENSOR_NCHW, n_out, c_out, h_out, w_out);
cudaMalloc(&ImageBatchOut, n_out * c_out * h_out * w_out * sizeof(float));

/* launch convolution on GPU */
cudnnConvolutionForward( handle, InputDesc, ImageInBatch, FilterDesc, Filter, convDesc,
    OutputDesc, ImageBatchOut, CUDNN_RESULT_NO_ACCUMULATE);
```

Implementation 1: 2D conv as a GEMV

Image

I1	I2	I3	I4	I5	I6
I7	I8	I9	I10	I11	I12
I13	I14	I15	I16	I17	I18
I19	I20	I21	I22	I23	I24
I25	I26	I27	I28	I29	I30
I31	I32	I33	I34	I35	I36

I1	I2	I3	I7	I8	I9	I13	I14	I15
I2	I3	I4	I8	I9	I10	I14	I15	I16
I3	I4	I5	I9	I10	I11	I15	I16	I17

F1
F2
F3
F4
F5
F6
F7
F8
F9

A lot of data duplication!

F1	F2	F3
F4	F5	F6
F7	F8	F9

Filter

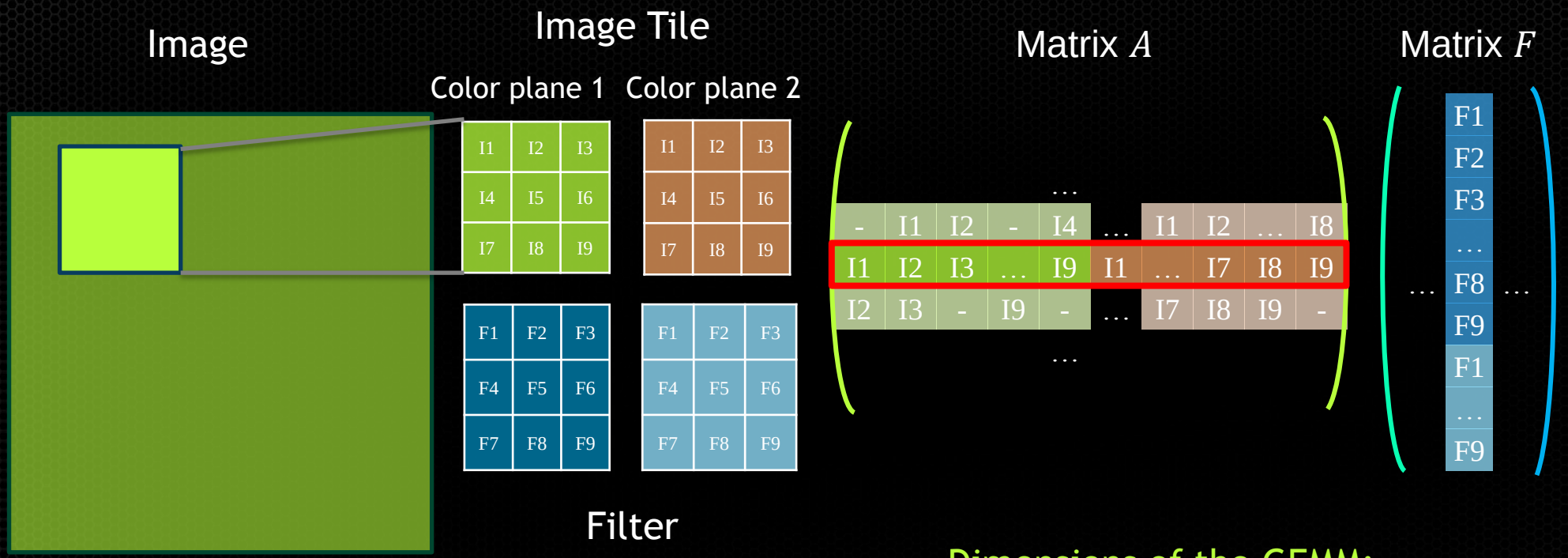
Multi-convolve

- More of the same, just a little different
  - Longer dot products
  - More filter kernels
  - Batch of images, not just one
- Mathematically:

$$out[k,p,q] = \sum_{c \in \text{input color planes}} \left(\sum_{i,j \in \text{filter}} Im[c][istart+i, jstart+j] \,.\, Filt[k][c][i,j] \right)$$

$$\forall k \in \text{output color planes}, (p,q) \in \text{output image}$$

Implementation 2: Multi-convolve as GEMM



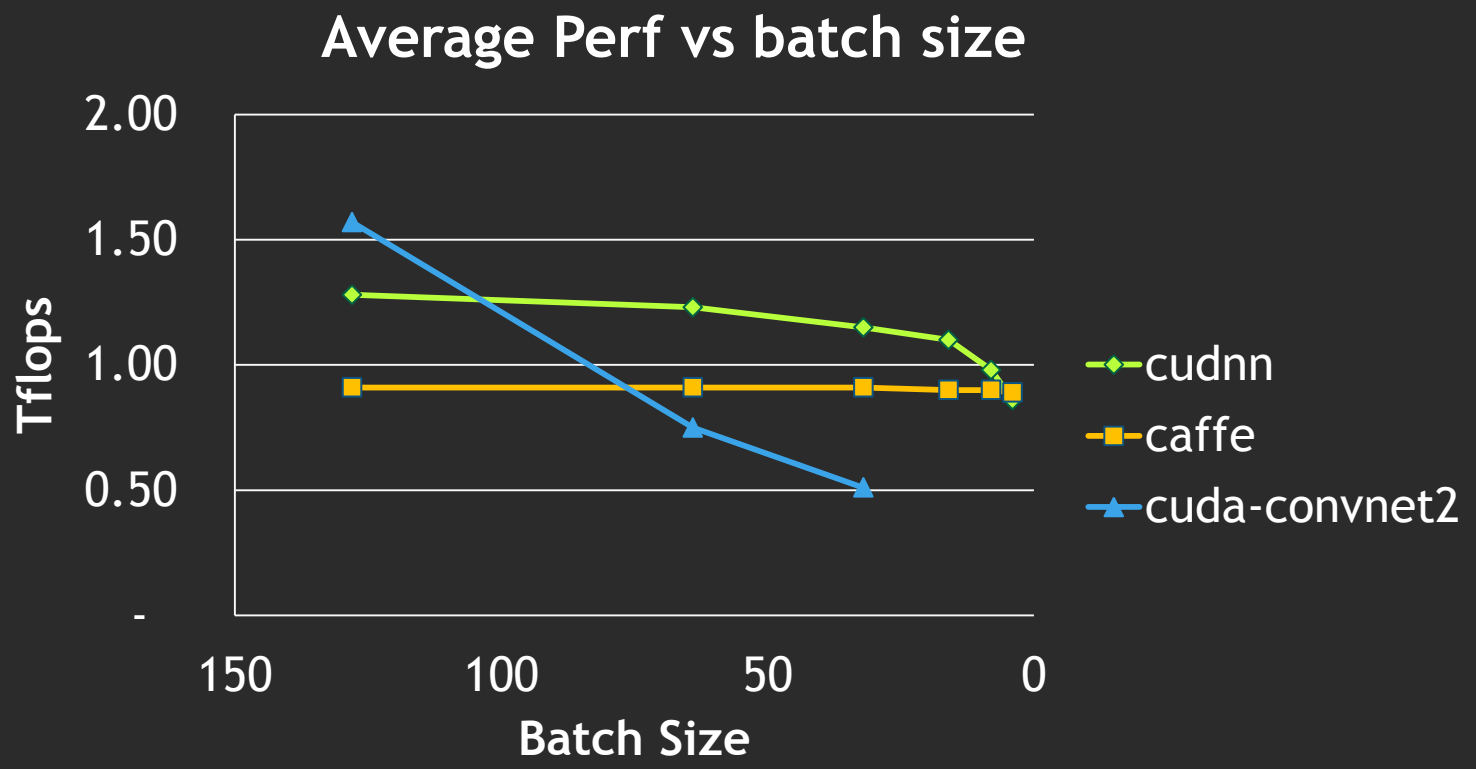
Dimensions of the GEMM:

m (rows of A) = N * P * Q

n (cols of F) = K

k (cols of A) = C * R * S

Performance



* Using **convnet-benchmarks** published by Soumith Chintala
  https://github.com/soumith/convnet-benchmarks
** Perf on a K40 GPU

cuDNN Integration

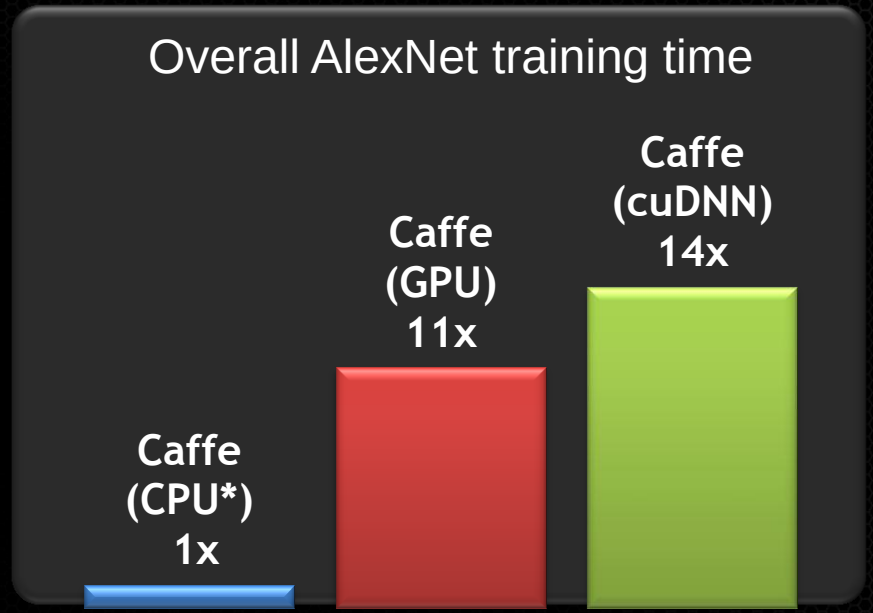
- cuDNN is already integrated in major open-source frameworks
  - Caffe
  - Torch
  - Theano (coming soon)

Yann LeCun:

"It is an awesome move on NVIDIA's part to be offering direct support for convolutional nets."

Using Caffe with cuDNN

- Accelerate Caffe layer types by 1.2 – 3x
- On average, 36% faster overall for training on Alexnet
- Integrated into Caffe dev branch today!
  (official release with Caffe 1.0)
- Seamless integration with a global switch



Baseline Caffe compared to Caffe accelerated by cuDNN on K40

*CPU is 24 core E5-2697v2 @ 2.4GHz
Intel MKL 11.1.3

NVIDIA

NVIDIA® cuDNN Roadmap

Release 1
September 2014

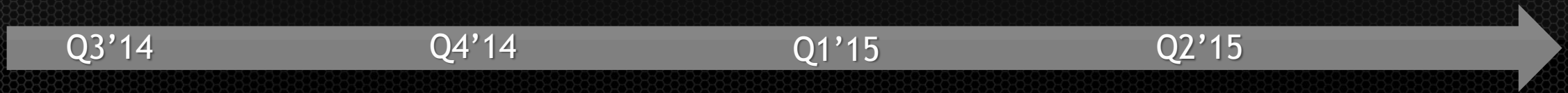
- Layers (foward & backprop)
  - Convolutional
  - Pooling
  - Softmax
  - ReLu/Sigmoid/Tanh
- High performance convolution

Release 2

- Layers
  - Local receptive field
  - Contrast normalization
  - Fully-connected
  - Recurrent
- Faster convolution routines

Release 3

- Support for multiple GPUs per node
- Tuning for future chips



Features

Performance

NVIDIA

cuDNN availability

- Free for registered developers!
- Release 1
  - available on Linux/Windows 64bit
  - GPU support for Kepler and newer
- Upcoming:
  - Tegra K1 (Jetson board)
  - Mac OSX support

Download: https://developer.nvidia.com/cuDNN
Paper: http://arxiv.org/pdf/1410.0759v2.pdf
Contact: cudnn@nvidia.com

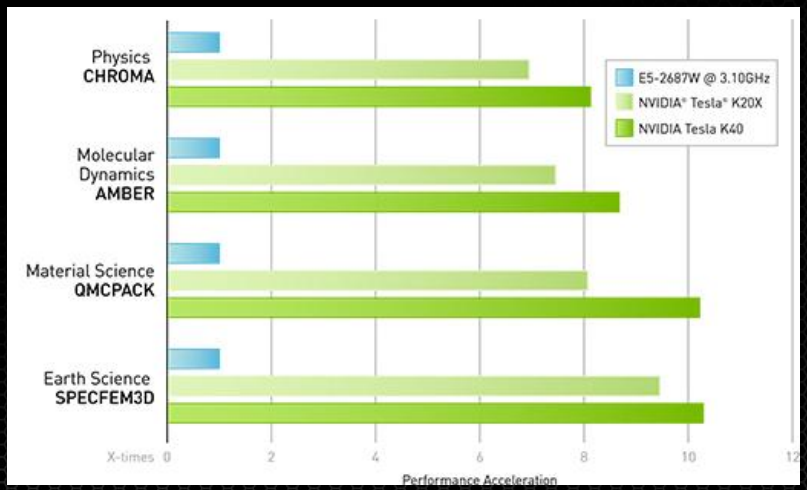
Questions?

Test drive GPU accelerators today

Accelerate your scientific discoveries:

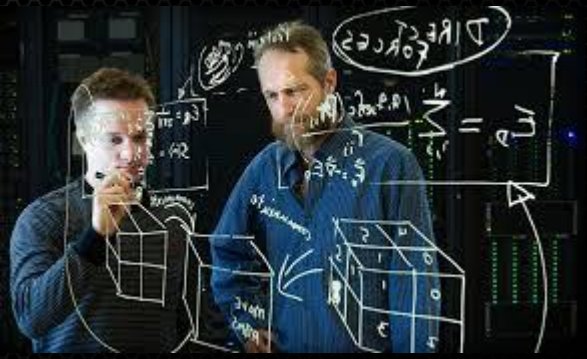
- ✓ Reducing simulation time from hours to minutes
- ✓ Using the latest Tesla K40 GPUs

FREE GPU Trial at:
www.nvidia.com/GPUTestDrive



NVIDIA

Upcoming GTC Express webinars



Tuesday, November 11

Heterogeneous CPU+GPU Molecular Dynamics Engine in CHARMM with Biofuels Applications

Antti-Pekka Hynninen and Mike Crowley, NREL

Wednesday, December 3

DIY Deep Learning for Vision: A Tutorial with Caffe

Evan Shelhamer, UC Berkeley

GPU TECHNOLOGY CONFERENCE

March 17-20, 2015 | San Jose, CA
www.gputechconf.com #GTC15

REGISTRATION IS OPEN!

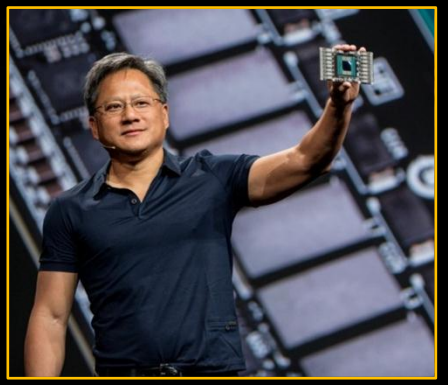
20% OFF
GM15WEB

CONNECT

Connect with experts from NVIDIA and other organizations across a wide range of fields

LEARN

Get key learnings and hands-on training in the 400+ sessions and 150+ research posters



DISCOVER

Discover the latest technologies shaping the GPU ecosystem

INNOVATE

Hear about disruptive innovations as early-stage start-ups present their work

4 Days | 3400+ Attendees | 400+ Sessions | 150+ Research Posters
40+ Countries | 180+ Press & Analytics | 100+ Exhibitors