



Engineering Challenges of Deep Learning

feat. C++



Peter Goldsborough

Facebook

November 9, 2017

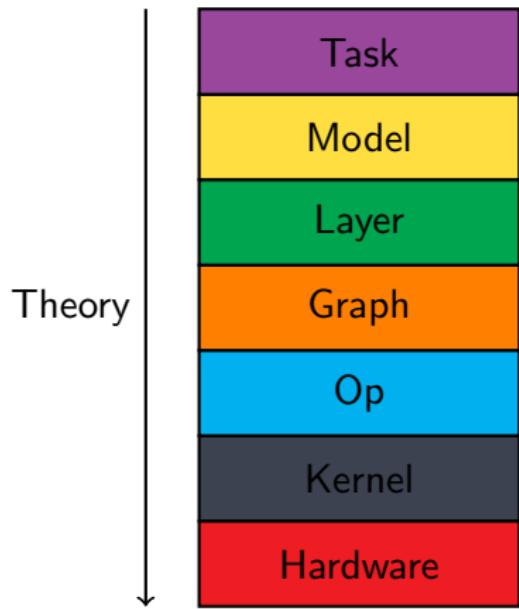
Strategy

Strategy



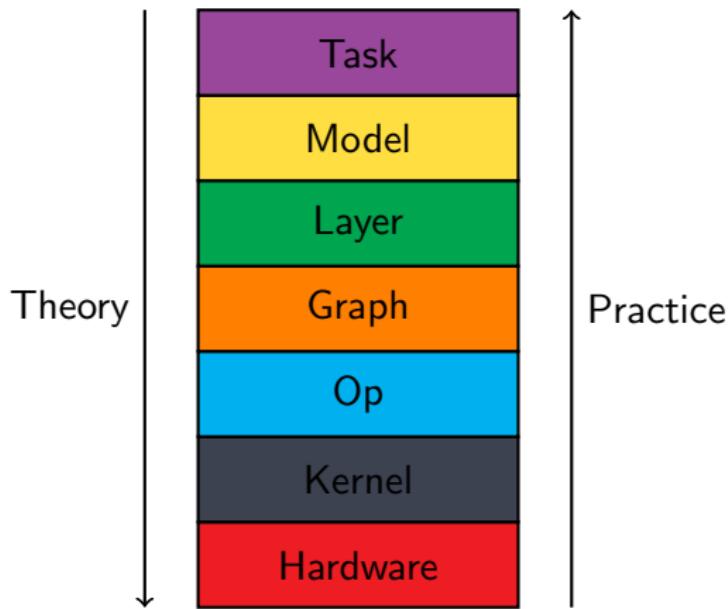
OSI Model of Machine Learning

Strategy



OSI Model of Machine Learning

Strategy



OSI Model of Machine Learning

Task Layer

Task Layer

$$f \left(\text{Teapot} \right) = \text{Teapot}$$

Classification^{*}

* Supervised

Task Layer

$$f \left(\text{Teapot} \right) = \text{Teapot}$$

Classification^{*}

$$f \left(\text{Cloud} \right) = \begin{matrix} \text{Cloud 1} & \text{Cloud 2} & \text{Cloud 3} \\ \text{Cloud 4} & \text{Cloud 5} & \text{Cloud 6} \\ \text{Cloud 7} & \text{Cloud 8} & \text{Cloud 9} \end{matrix}$$

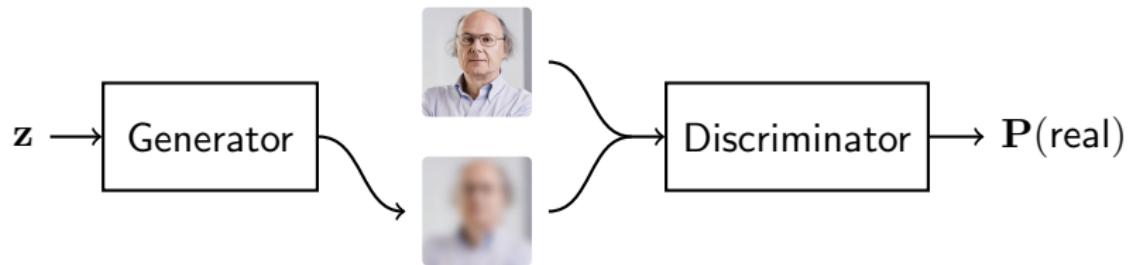
Generative[†]

* Supervised

† Unsupervised

Model Layer

Model Layer



Generative Adversarial Network (GAN)

Generative Adversarial Networks



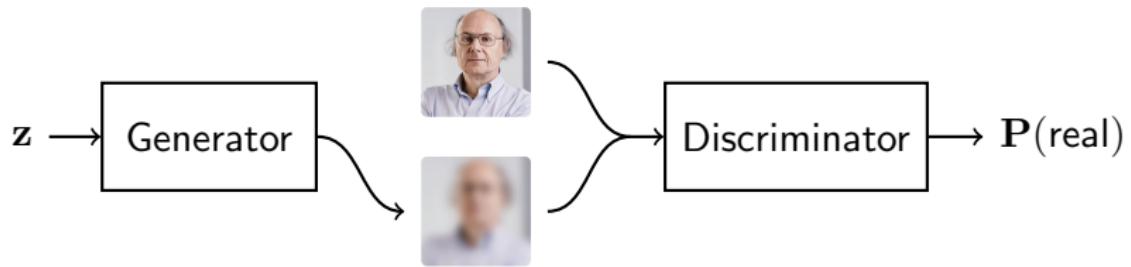
Progressive GAN: Karras et al. (2017)

Generative Adversarial Networks



BEGAN: Berthelot et al. (2017)

Layer Layer

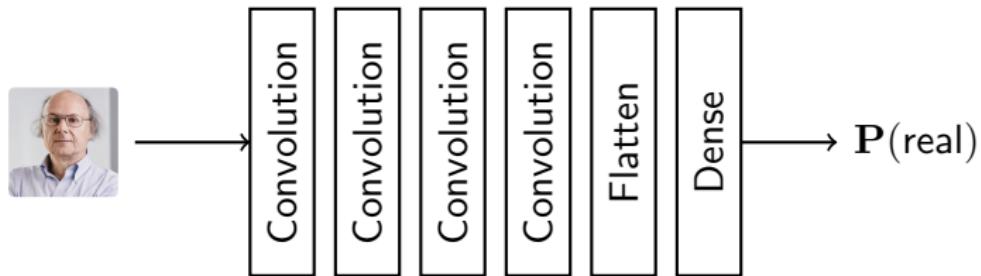


Generative Adversarial Networks (GAN)

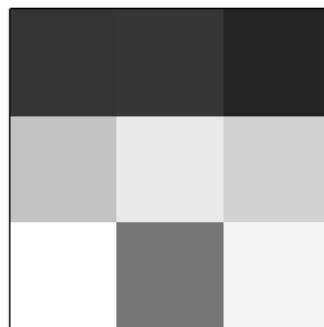
Layer Layer



Layer Layer



Layers: Convolution



Image

Layers: Convolution

0.4	0.9	0.1
0.7	0.2	0.6
0.8	0.3	0.5

Image

Layers: Convolution

0.4	0.9	0.1
0.7	0.2	0.6
0.8	0.3	0.5

Image

5.7	2.4
3.1	0.9

Kernel

Layers: Convolution

$5.7 \cdot 0.4$	$2.4 \cdot 0.9$	0.1
$3.1 \cdot 0.7$	$0.9 \cdot 0.2$	0.6
0.8	0.3	0.5

Image

Layers: Convolution

$5.7 \cdot 0.4$	$2.4 \cdot 0.9$	0.1
$3.1 \cdot 0.7$	$0.9 \cdot 0.2$	0.6
0.8	0.3	0.5

Image

6.79

Output

Layers: Convolution

0.4	$5.7 \cdot 0.9$	$2.4 \cdot 0.1$
0.7	$3.1 \cdot 0.2$	$0.9 \cdot 0.6$
0.8	0.3	0.5

Image

6.79

Output

Layers: Convolution

0.4	$5.7 \cdot 0.9$	$2.4 \cdot 0.1$
0.7	$3.1 \cdot 0.2$	$0.9 \cdot 0.6$
0.8	0.3	0.5

Image

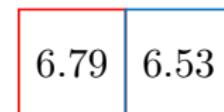
6.79	6.53
------	------

Output

Layers: Convolution

0.4	0.9	0.1
$5.7 \cdot 0.7$	$2.4 \cdot 0.2$	0.6
$3.1 \cdot 0.8$	$0.9 \cdot 0.3$	0.5

Image

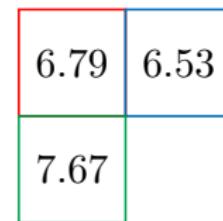


Output

Layers: Convolution

0.4	0.9	0.1
$5.7 \cdot 0.7$	$2.4 \cdot 0.2$	0.6
$3.1 \cdot 0.8$	$0.9 \cdot 0.3$	0.5

Image



Output

Layers: Convolution

0.4	0.9	0.1
0.7	$5.7 \cdot 0.2$	$2.4 \cdot 0.6$
0.8	$3.1 \cdot 0.3$	$0.9 \cdot 0.5$

Image

6.79	6.53
7.67	

Output

Layers: Convolution

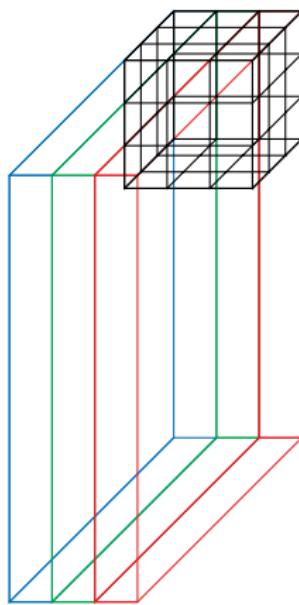
0.4	0.9	0.1
0.7	$5.7 \cdot 0.2$	$2.4 \cdot 0.6$
0.8	$3.1 \cdot 0.3$	$0.9 \cdot 0.5$

Image

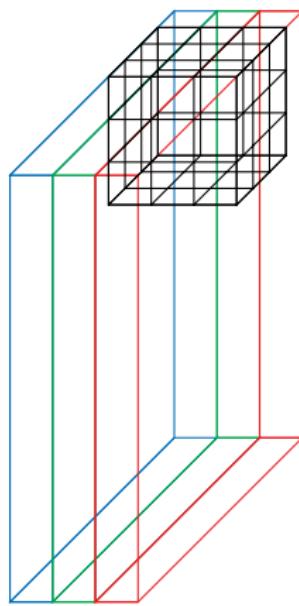
6.79	6.53
7.67	3.96

Output

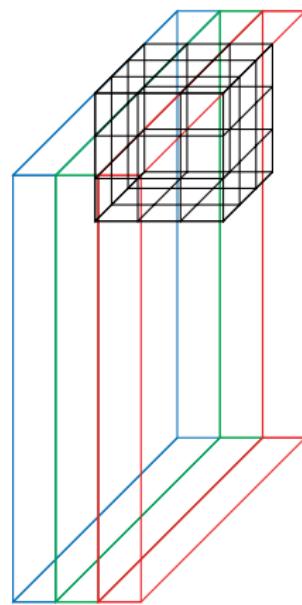
Layers: Convolution



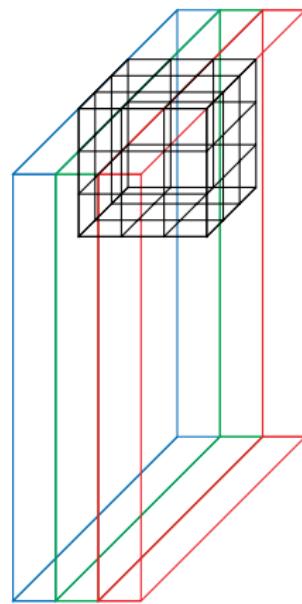
Layers: Convolution



Layers: Convolution



Layers: Convolution



Layers: Convolution

0	1	0
0	1	0
0	1	0

0	0	0
-1	1	0
0	0	0

Layers: Convolution

0	1	0
0	1	0
0	1	0

0	0	0
-1	1	0
0	0	0

Patterns

Layers: Convolution

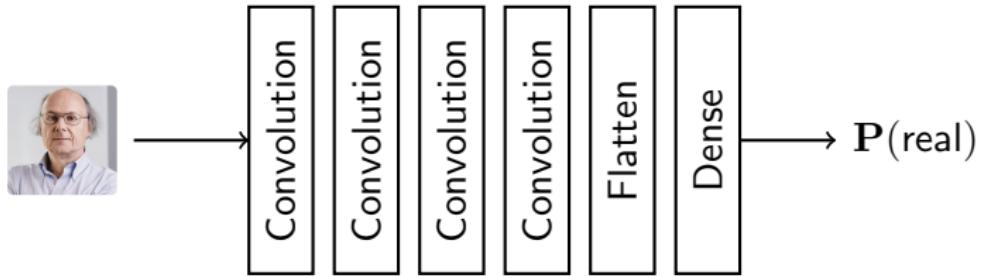
0	1	0
0	1	0
0	1	0

Patterns

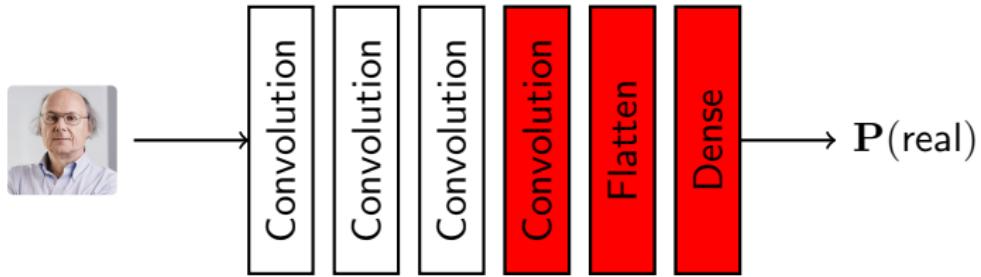
0	0	0
-1	1	0
0	0	0

Edges

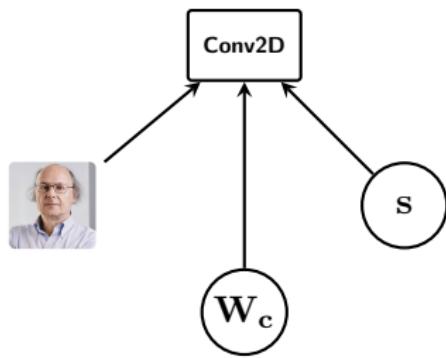
Graph Layer



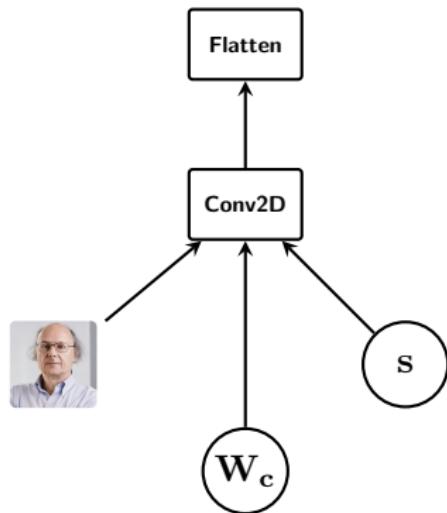
Graph Layer



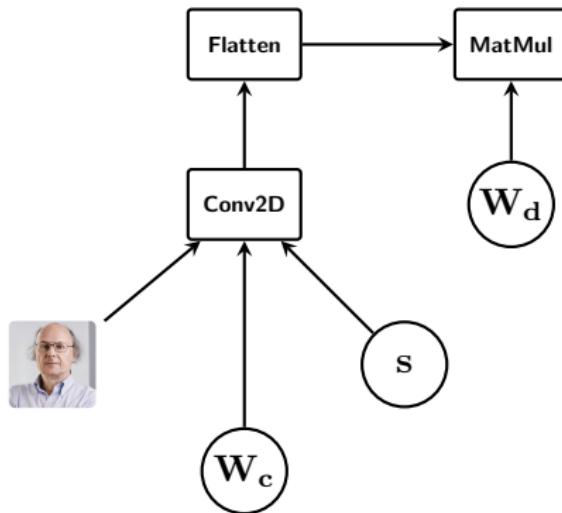
Graph Layer



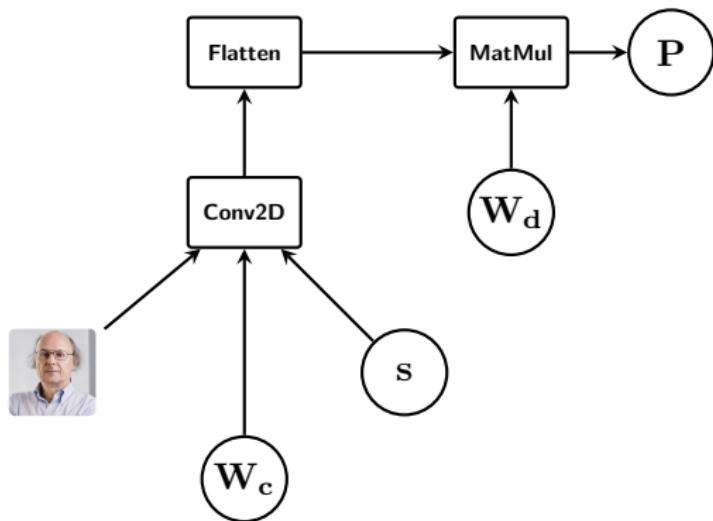
Graph Layer



Graph Layer

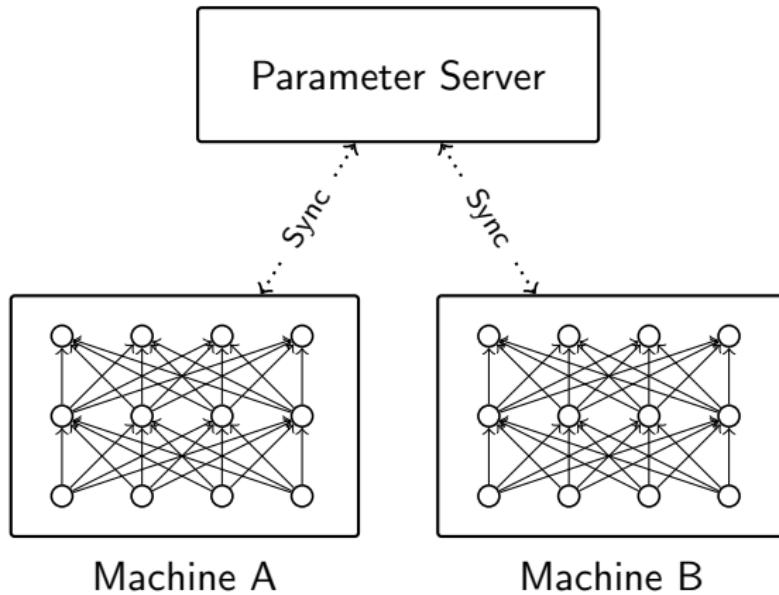


Graph Layer



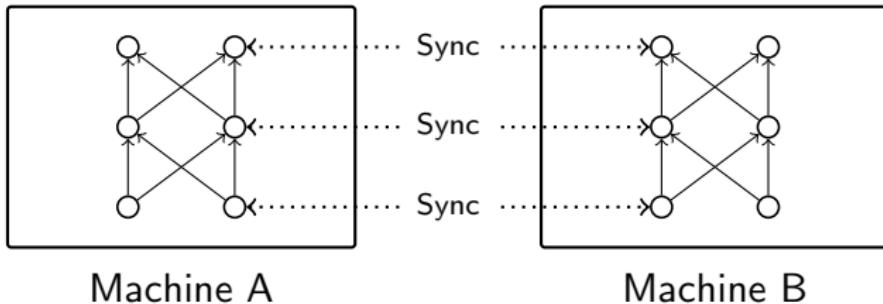
Graph Layer Parallelism

Graph Layer Parallelism



Data Parallelism

Graph Layer Parallelism



Model Parallelism

Static Graphs

Static Graphs

```
Matrix<6, 9> a = ...;
Matrix<6, 9> b = ...;
Matrix<16, 6> c = ...;
scalar s = 7;

d := s * a + b;
e := matmul(c, d);

x := if_clause(condition, d, e);

result := evaluate(x);
```

Dynamic Graphs

```
Matrix a = ...;
Matrix b = ...;
Matrix c = ...;
scalar s = 7;

if s > 0 {
    d := s * a + b;
} else {
    d := a - b;
}

while s > 0 {
    x = input();
    c = matmul(c, x);
}

result := c;
```

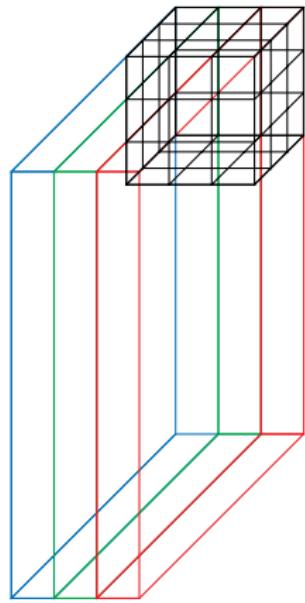
Op Layer

Op Layer

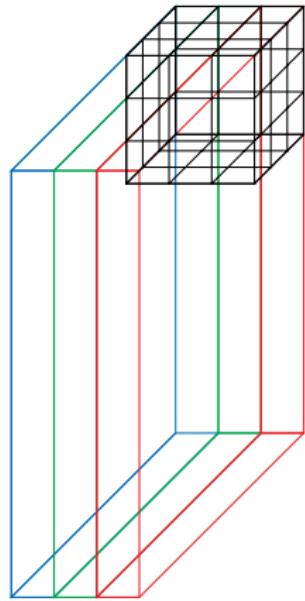


Bob

im2col

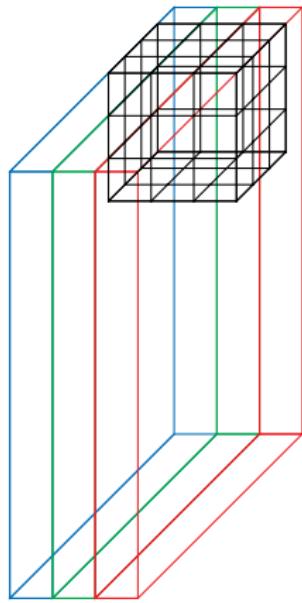


im2col

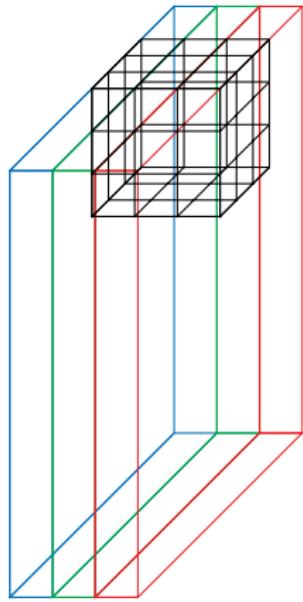


$$\begin{matrix} x_0 & x_1 & x_2 \\ \dots & & \\ x_{24} & x_{25} & x_{26} \end{matrix}$$

im2col

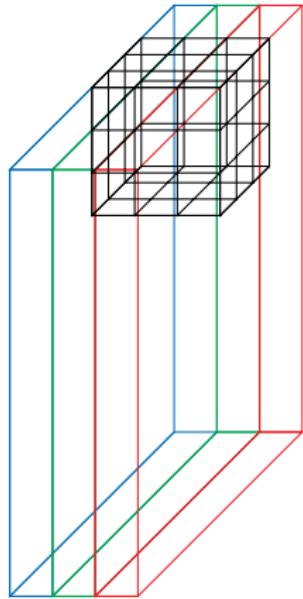

$$\begin{array}{|c|c|c|} \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline \end{array} \quad \cdots \quad \begin{array}{|c|c|c|} \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline \end{array}$$

im2col

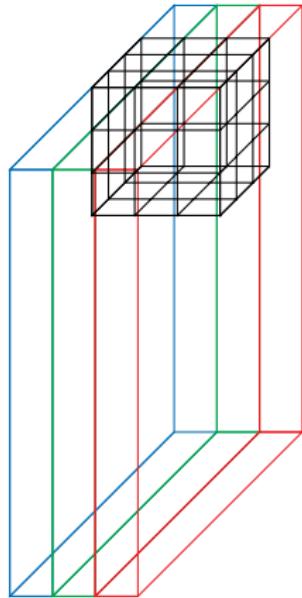


x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}

im2col

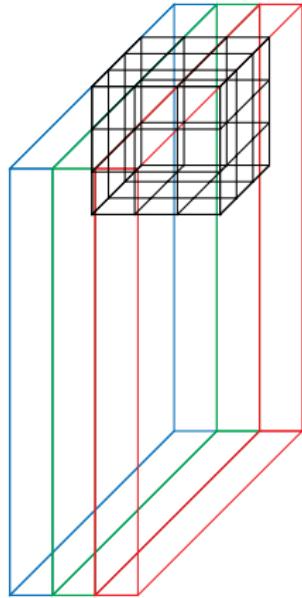

$$\begin{array}{|c|c|c|} \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array} \quad \cdots \quad \begin{array}{|c|c|c|} \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array}$$
$$\begin{array}{|c|c|c|} \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array} \quad \cdots \quad \begin{array}{|c|c|c|} \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array}$$

im2col



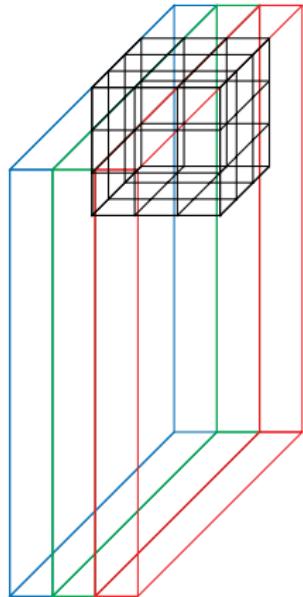
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_0
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_1
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_{24}
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_{25}
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_{26}

im2col



x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_0	k_0
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_1	k_1
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_2	k_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_{24}	k_{24}
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_{25}	k_{25}
x_0	x_1	x_2	⋮	x_{24}	x_{25}	x_{26}	k_{26}	k_{26}

im2col



$$\begin{matrix} x_0 \\ x_0 \\ x_0 \end{matrix}$$

$$\begin{matrix} x_1 \\ x_1 \\ x_1 \end{matrix}$$

$$\begin{matrix} x_2 \\ x_2 \\ x_2 \end{matrix}$$

⋮

⋮

⋮

$$\begin{matrix} x_{24} \\ x_{24} \\ x_{24} \end{matrix}$$

$$\begin{matrix} x_{25} \\ x_{25} \\ x_{25} \end{matrix}$$

$$\begin{matrix} x_{26} \\ x_{26} \\ x_{26} \end{matrix}$$

$$\begin{matrix} k_0 \\ k_1 \\ k_2 \end{matrix}$$

$$\begin{matrix} k_0 \\ k_1 \\ k_2 \end{matrix}$$

$$\begin{matrix} k_0 \\ k_1 \\ k_2 \end{matrix}$$

$$\begin{matrix} x_0 \\ x_0 \\ x_0 \end{matrix}$$

$$\begin{matrix} x_1 \\ x_1 \\ x_1 \end{matrix}$$

$$\begin{matrix} x_2 \\ x_2 \\ x_2 \end{matrix}$$

⋮

⋮

⋮

$$\begin{matrix} x_{24} \\ x_{24} \\ x_{24} \end{matrix}$$

$$\begin{matrix} x_{25} \\ x_{25} \\ x_{25} \end{matrix}$$

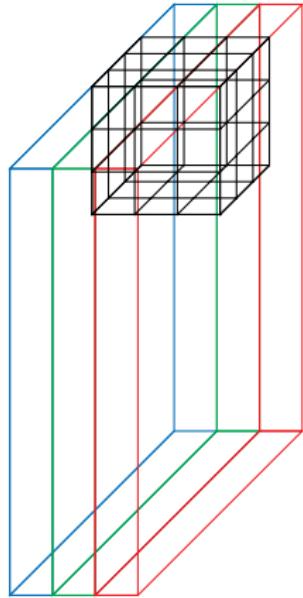
$$\begin{matrix} x_{26} \\ x_{26} \\ x_{26} \end{matrix}$$

$$\begin{matrix} k_{24} \\ k_{25} \\ k_{26} \end{matrix}$$

$$\begin{matrix} k_{24} \\ k_{25} \\ k_{26} \end{matrix}$$

$$\begin{matrix} k_{24} \\ k_{25} \\ k_{26} \end{matrix}$$

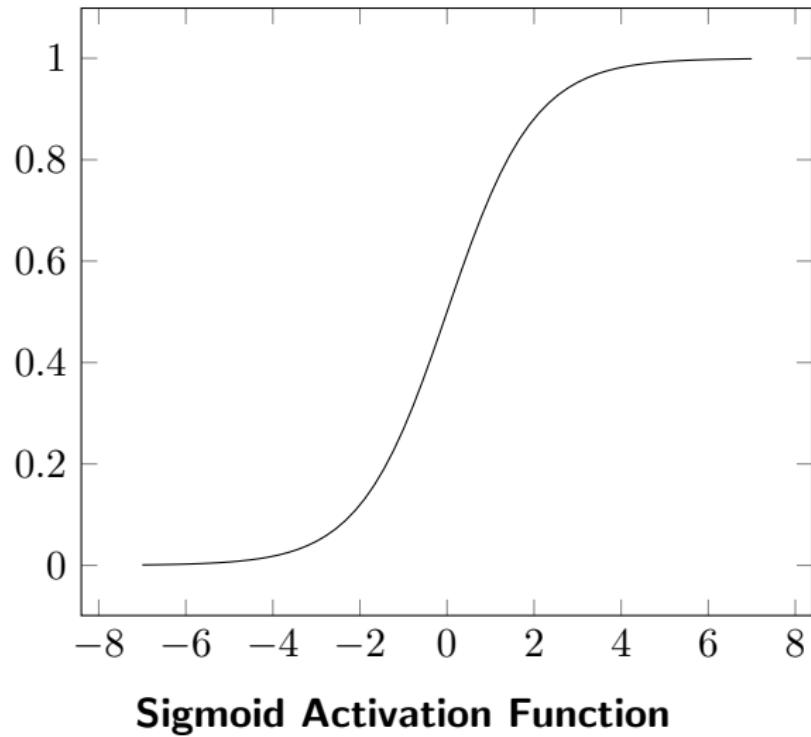
im2col



$$\begin{array}{|c|c|c|} \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array} \cdots \begin{array}{|c|c|c|} \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline k_0 & k_0 & k_0 \\ \hline k_1 & k_1 & k_1 \\ \hline k_2 & k_2 & k_2 \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array}$$
$$\begin{array}{|c|c|c|} \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline x_0 & x_1 & x_2 \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array} \cdots \begin{array}{|c|c|c|} \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline x_{24} & x_{25} & x_{26} \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline k_{24} & k_{24} & k_{24} \\ \hline k_{25} & k_{25} & k_{25} \\ \hline k_{26} & k_{26} & k_{26} \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array}$$

Kernel Layer

Kernel Layer



Kernel Layer

Kernel Layer

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Forward Pass

Kernel Layer

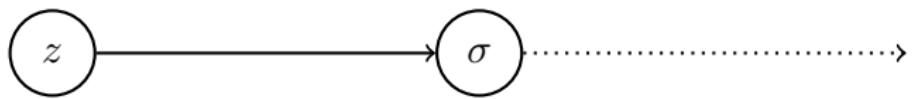
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Forward Pass

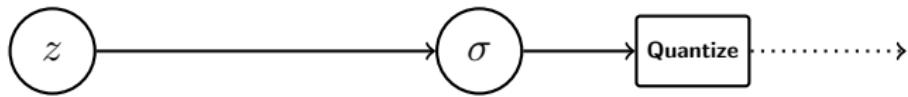
$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

Backward Pass

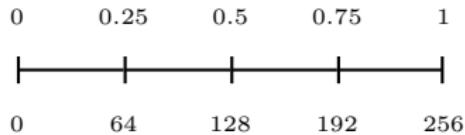
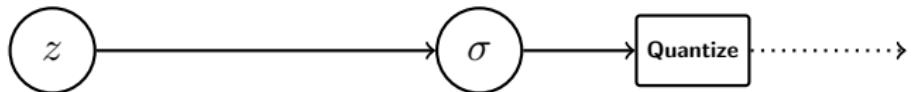
Quantization



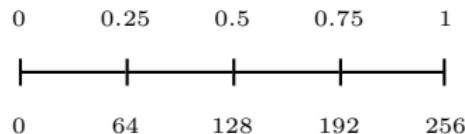
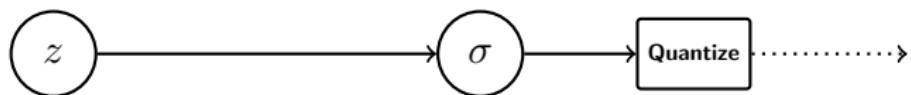
Quantization



Quantization



Quantization



github.com/google/gemmlowp

Hardware Layer

Hardware Layer



CPU

Hardware Layer



CPU



GPU

Hardware Layer



CPU



GPU



ASIC

THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG
PILE OF LINEAR ALGEBRA, THEN COLLECT
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL
THEY START LOOKING RIGHT.



Hardware: GPU



Titan X

- ▶ 3840 cores
- ▶ 12GB on-chip memory
- ▶ 547.7GB/s bandwidth
- ▶ 12 TFLOPS

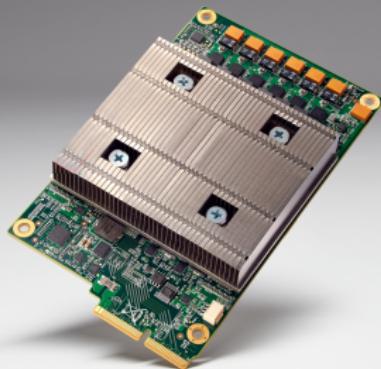
Hardware: Big Basin



Big Basin

- ▶ 8 NVIDIA Tesla P100 GPUs
- ▶ NVLink (3x faster than PCIe)
- ▶ 128GB RAM
- ▶ 84.8 TFLOPS
- ▶ Reduced Precision Computing

Hardware: TPU

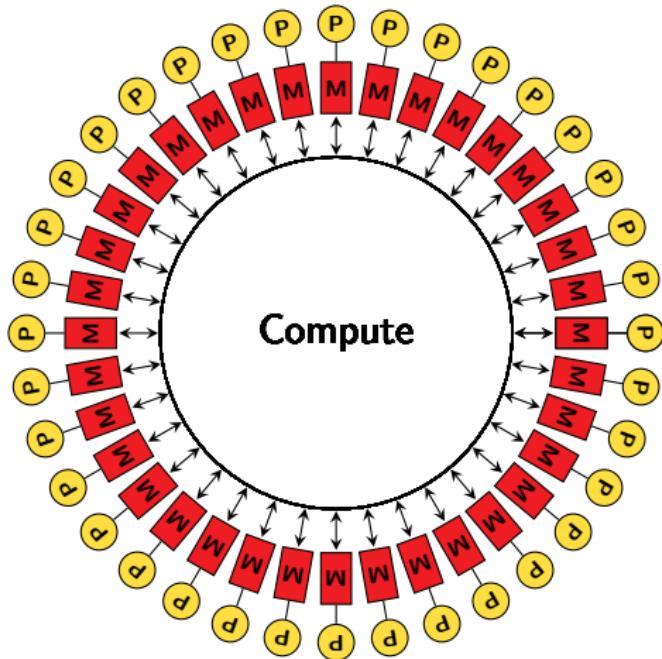


TPU

- ▶ *Coprocessor*
- ▶ 92 TOPS for 8-bit int
- ▶ 42 TFLOPS (TPU2)
- ▶ 24 MB Memory

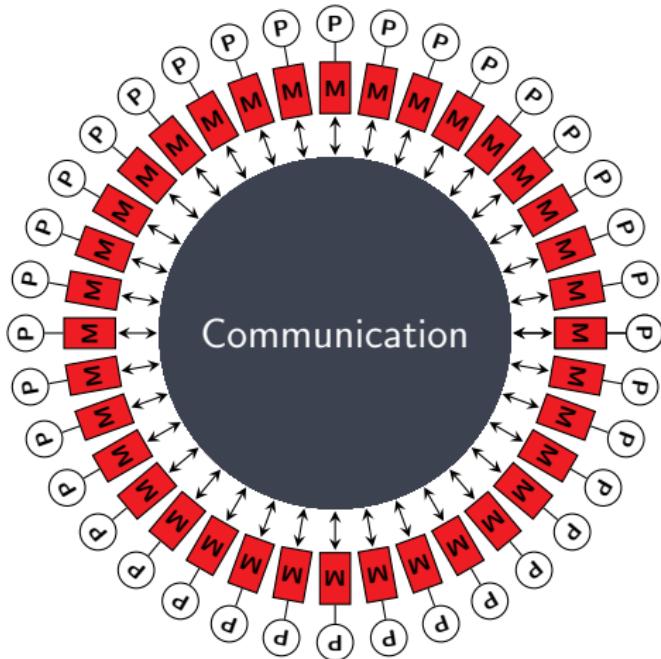
Hardware: IPU

Hardware: IPU



Bulk Synchronous Parallelism

Hardware: IPU



Bulk Synchronous Parallelism

Hardware: IPU

- ▶ Startup from Bristol (UK)
- ▶ Graphcore *Colossus*
- ▶ TBR later this year
- ▶ >1000 Processors/Chip
- ▶ Mixed Precision Arithmetic
- ▶ 2x – 200x faster than GPUs



Graphcore

Kernel Layer

Kernel Layer

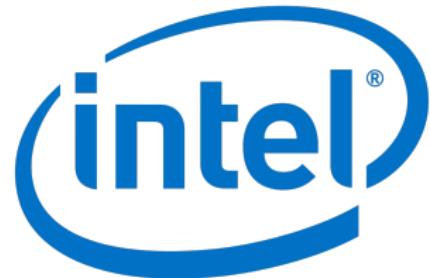


cuDNN/cuBLAS

Kernel Layer



cuDNN/cuBLAS



MKL

Demo: Convolution with cuDNN

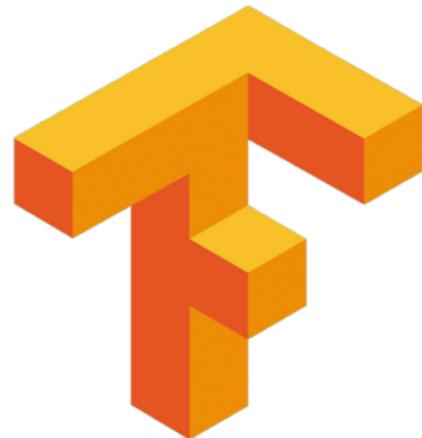
github.com/peter-can-talk/meeting-cpp-2017

Demo: Custom TensorFlow Op

Graph Layer

Graph Layer: TensorFlow

- ▶ Google (2015)
- ▶ Static Graphs*
- ▶ GPUs ✓
- ▶ Distributed ✓
- ▶ C++ API ✗
- ▶ Very “Deployable”
- ▶ Often Too Low-Level



* Dynamic with *TF Eager* (Oct. 2017)

Graph Layer: PyTorch/Caffe2



- ▶ Facebook (2016)
- ▶ Dynamic Graphs
- ▶ For Research
- ▶ GPUs ✓
- ▶ Distributed ✓
- ▶ Pure Python



- ▶ Facebook (2017)
- ▶ Static Graphs
- ▶ For Deployment
- ▶ GPUs ✓
- ▶ Distributed ✓
- ▶ C++ API ✓

Graph Layer: MXNet

- ▶ Community, then Amazon
- ▶ Apache Incubator
- ▶ Highly Modular
- ▶ GPUs ✓
- ▶ Distributed ✓
- ▶ Python, R, Scala, Julia, C++ ✓



Graph Layer: Dlib

- ▶ No Corporate Backing
- ▶ Mature and Reliable (2002)
- ▶ Pure C++ ✓
- ▶ GPUs ✓
- ▶ Distributed ✗



Layer Layer

Demo: MXNet Classifier

Layer Layer

Demo: Dlib Classifier

Q & A

github.com/peter-can-talk/meeting-cpp-2017