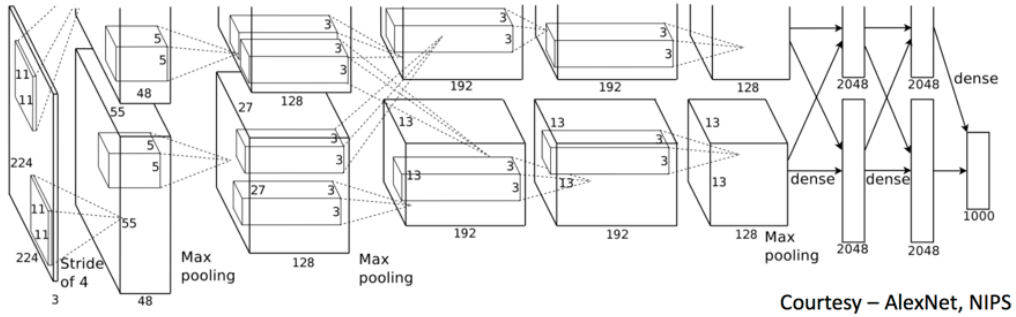
Hardware for Deep Learning

Bill Dally
Stanford and NVIDIA

Stanford Platform Lab Retreat
June 3, 2016

HARDWARE AND DATA ENABLE DNNS



Courtesy – AlexNet, NIPS 2012

THE NEED FOR SPEED

Larger data sets and models lead to better accuracy but also increase computation time. Therefore progress in deep neural networks is limited by how fast the networks can be computed.

Likewise the application of convnets to low latency inference problems, such as pedestrian detection in self driving car video imagery, is limited by how fast a small set of images, possibly a single image, can be classified.

More data ➔ Bigger Models ➔More Need for Compute

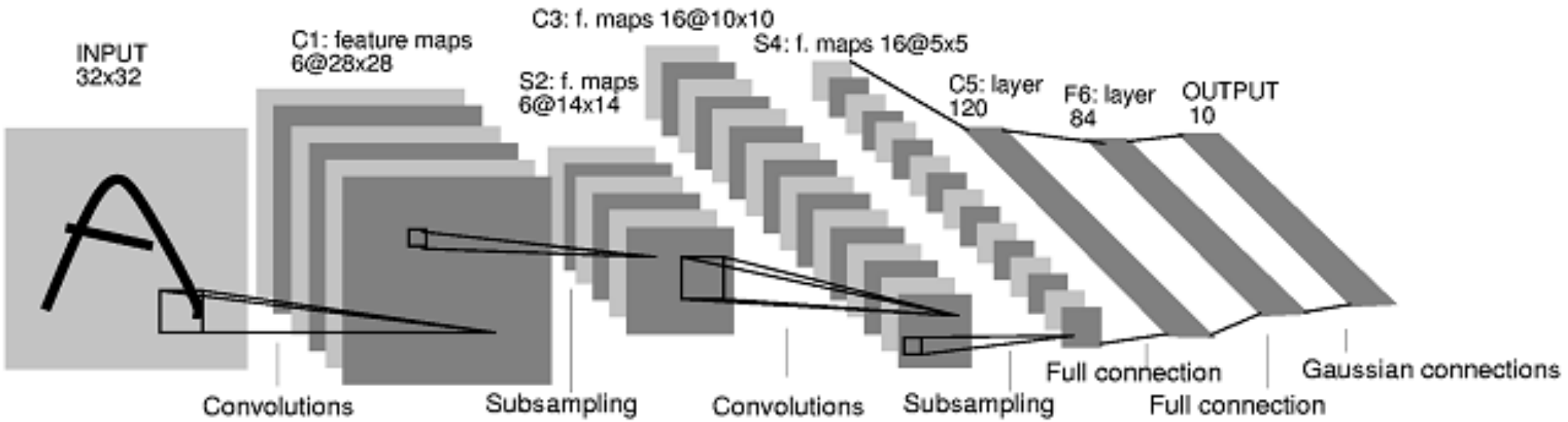
But Moore's law is no longer providing more compute…

Lavin & Gray, Fast Algorithms for Convolutional Neural Networks, 2015

Deep Neural Network

What is frames/J and frames/s/mm^2

for training & inference?



LeCun, Yann, et al. "Learning algorithms for classification: A comparison on handwritten digit recognition." *Neural networks: the statistical mechanics perspective* 261 (1995): 276.

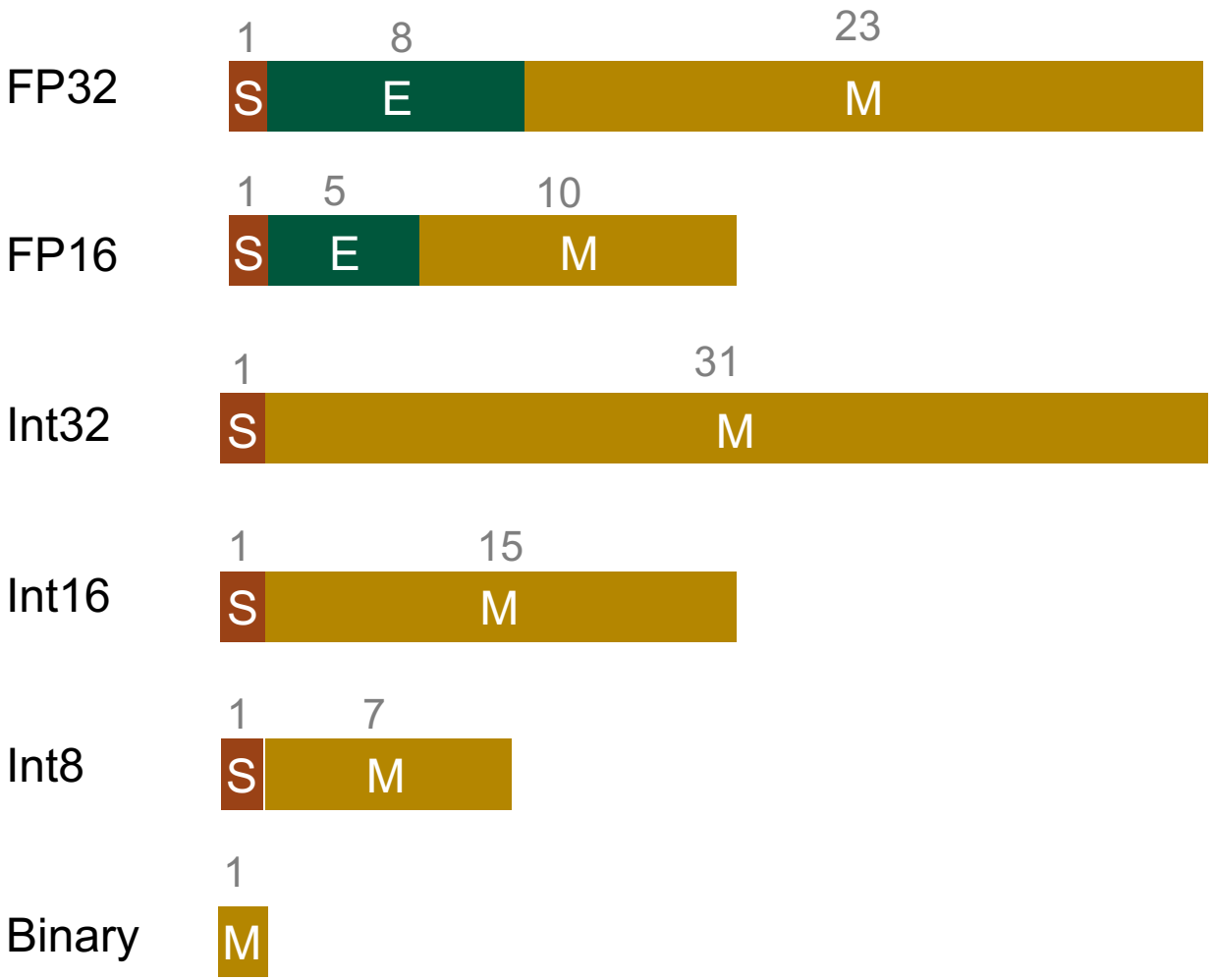
4 Distinct Sub-problems

Inference

Precision

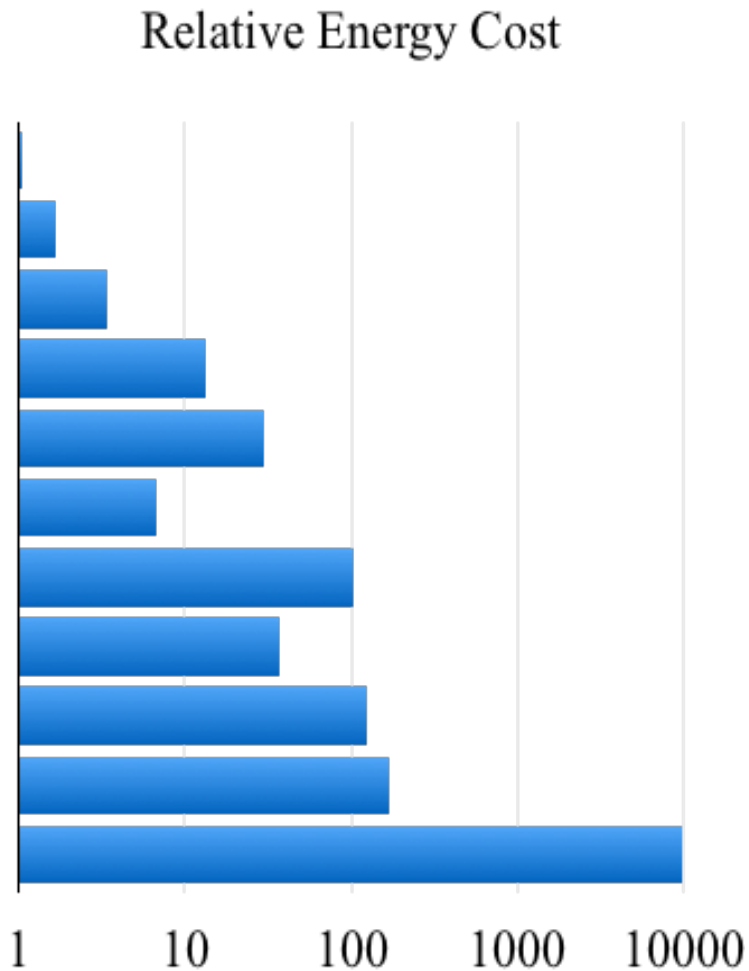
Use the “smallest” representation that doesn’t sacrifice accuracy
(32FP -> 4-6 bits quantized)

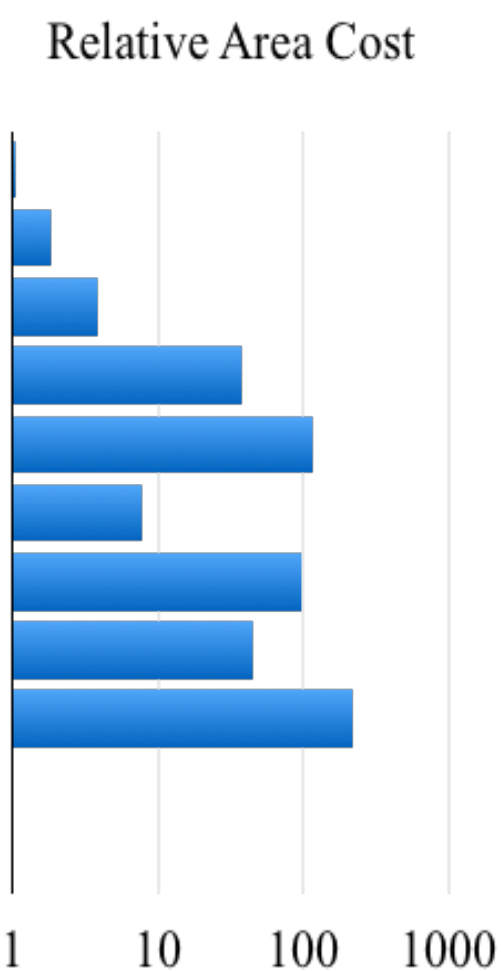
Number Representation



Cost of Operations

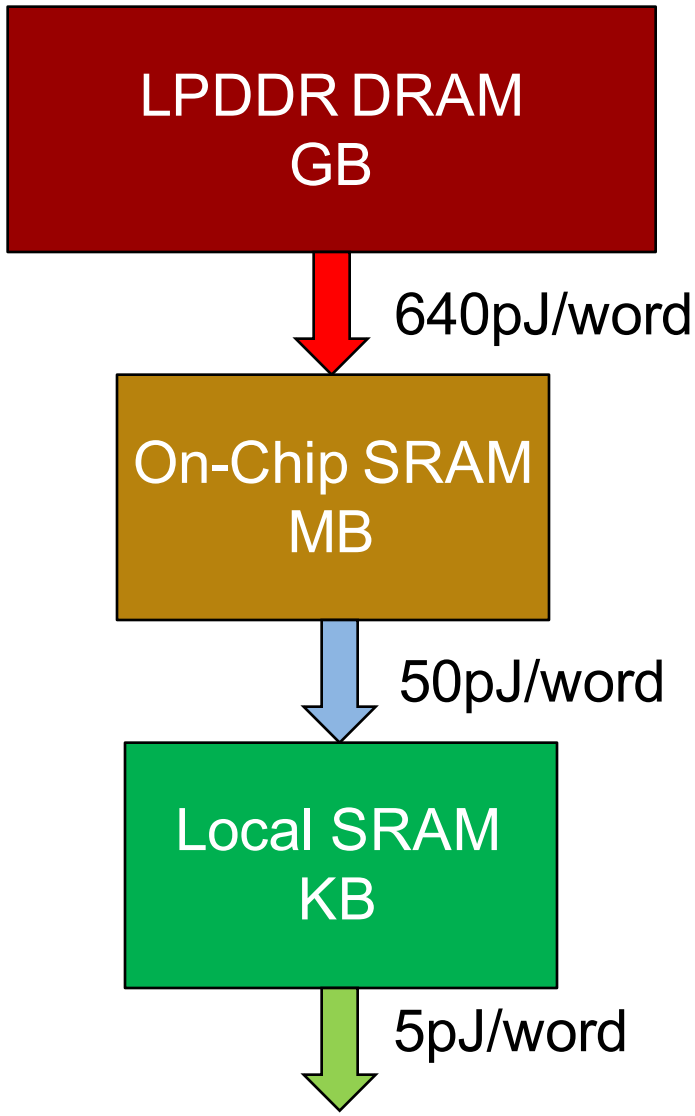
Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A





Energy numbers are from Mark Horowitz "Computing's Energy Problem (and what we can do about it)", ISSCC 2014
Area numbers are from synthesized result using Design Compiler under TSMC 45nm tech node. FP units used DesignWare Library.

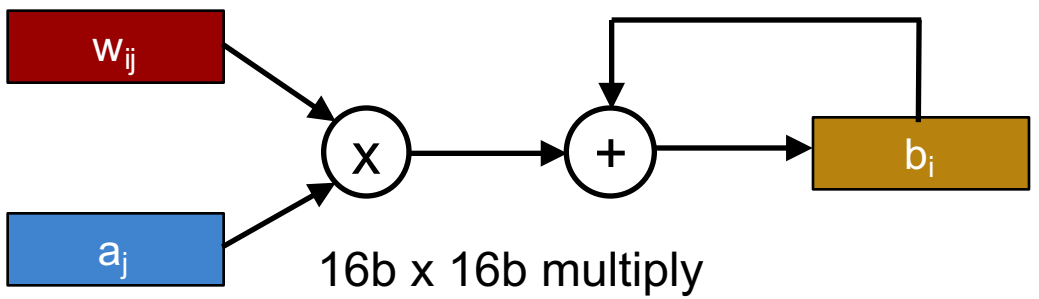
The Importance of Staying Local



Mixed Precision

Store weights as 4b using Trained quantization, decode to 16b

Store activations as 16b

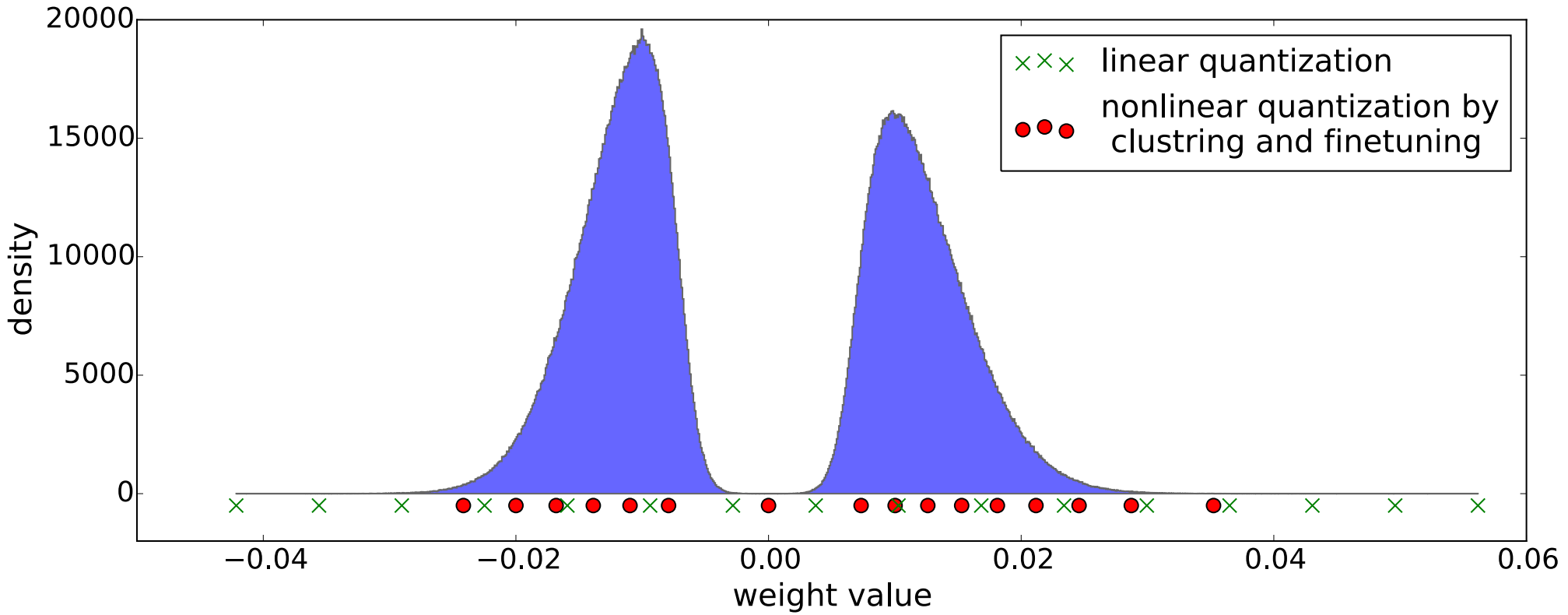


16b x 16b multiply
round result to 16b

accumulate 24b or 32b
to avoid saturation

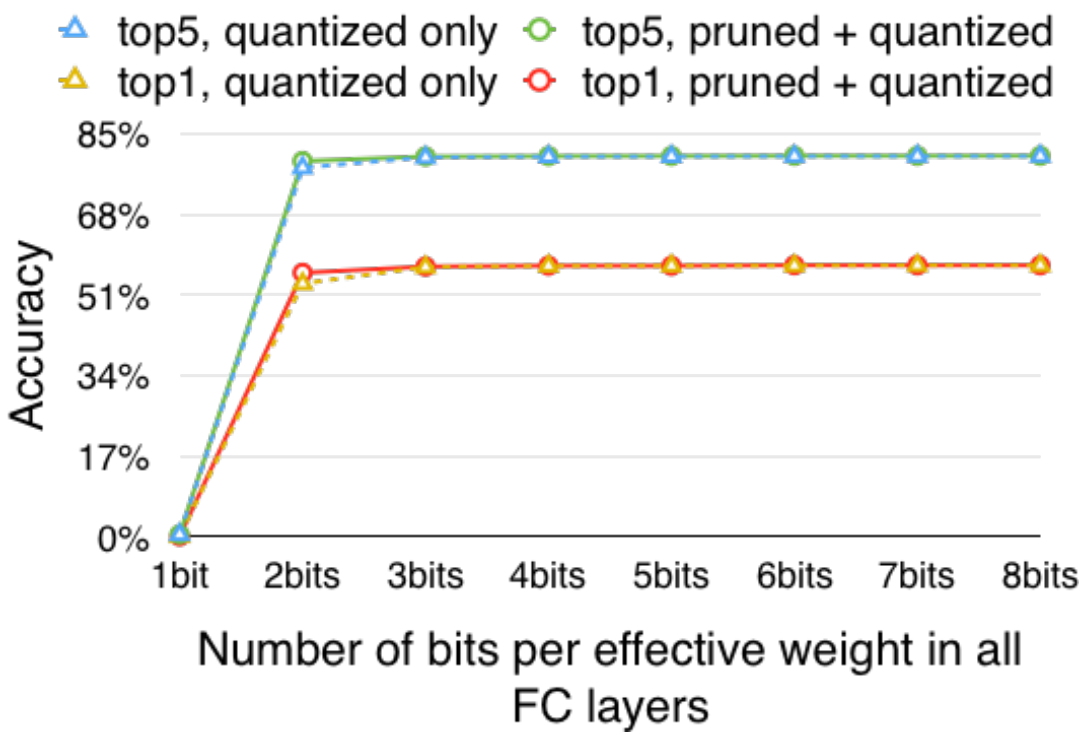
Batch normalization important to ‘center’ dynamic range

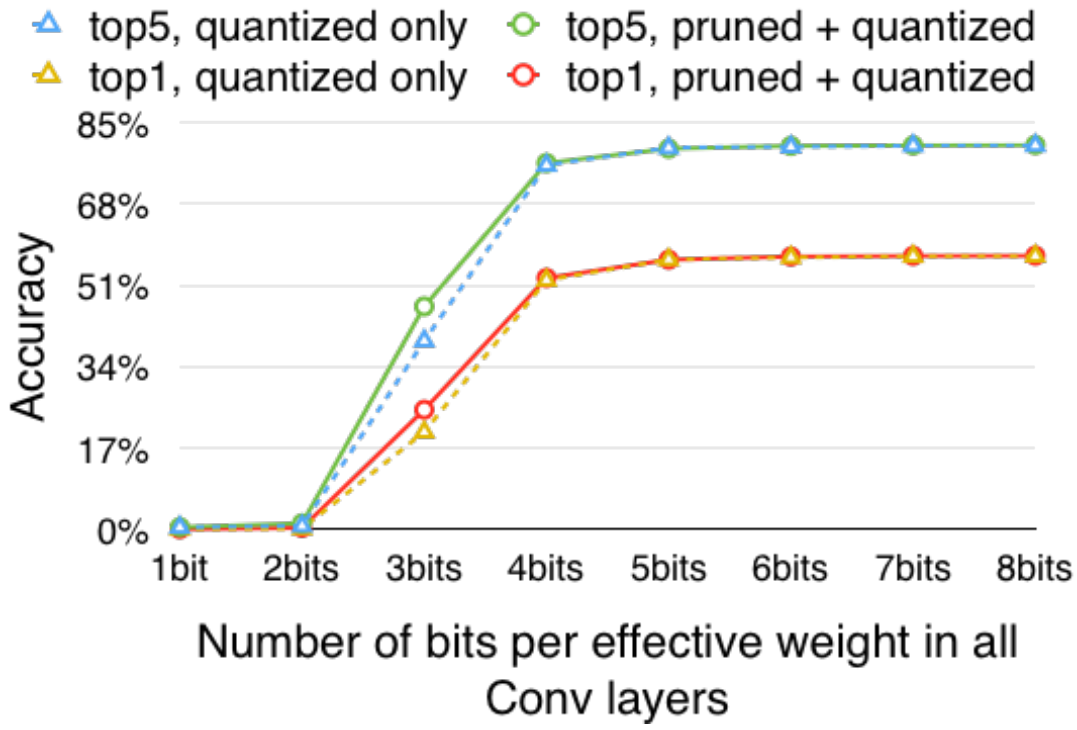
Trained Quantization



Han et al. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, ICLR 2016 (Best Paper)

2-6 Bits/Weight Sufficient

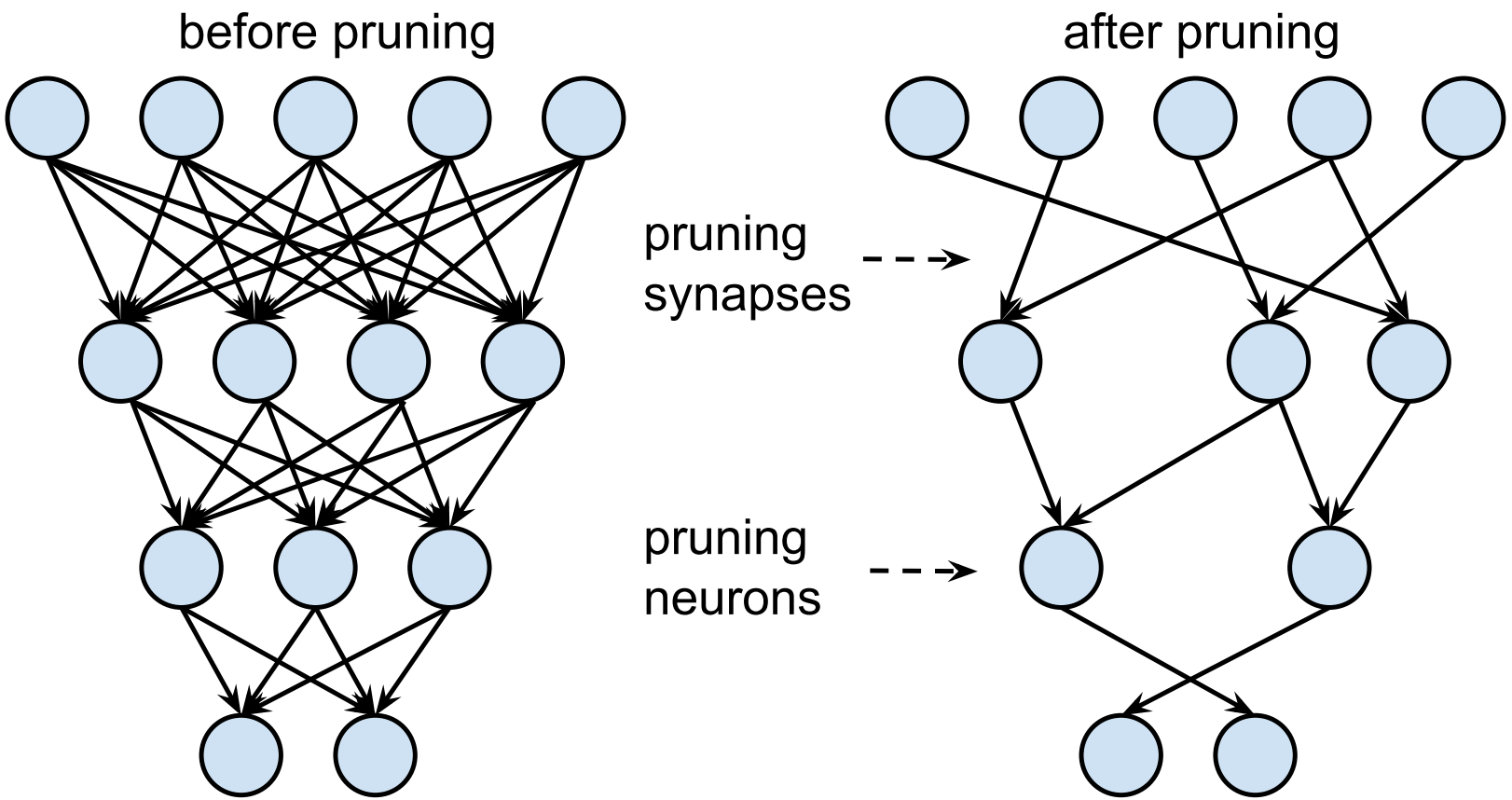




Sparsity

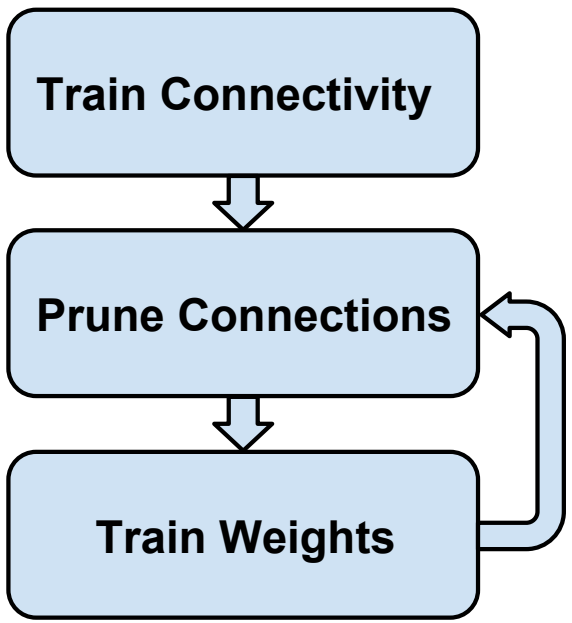
Don’t do operations that don’t matter (10x - 30x)

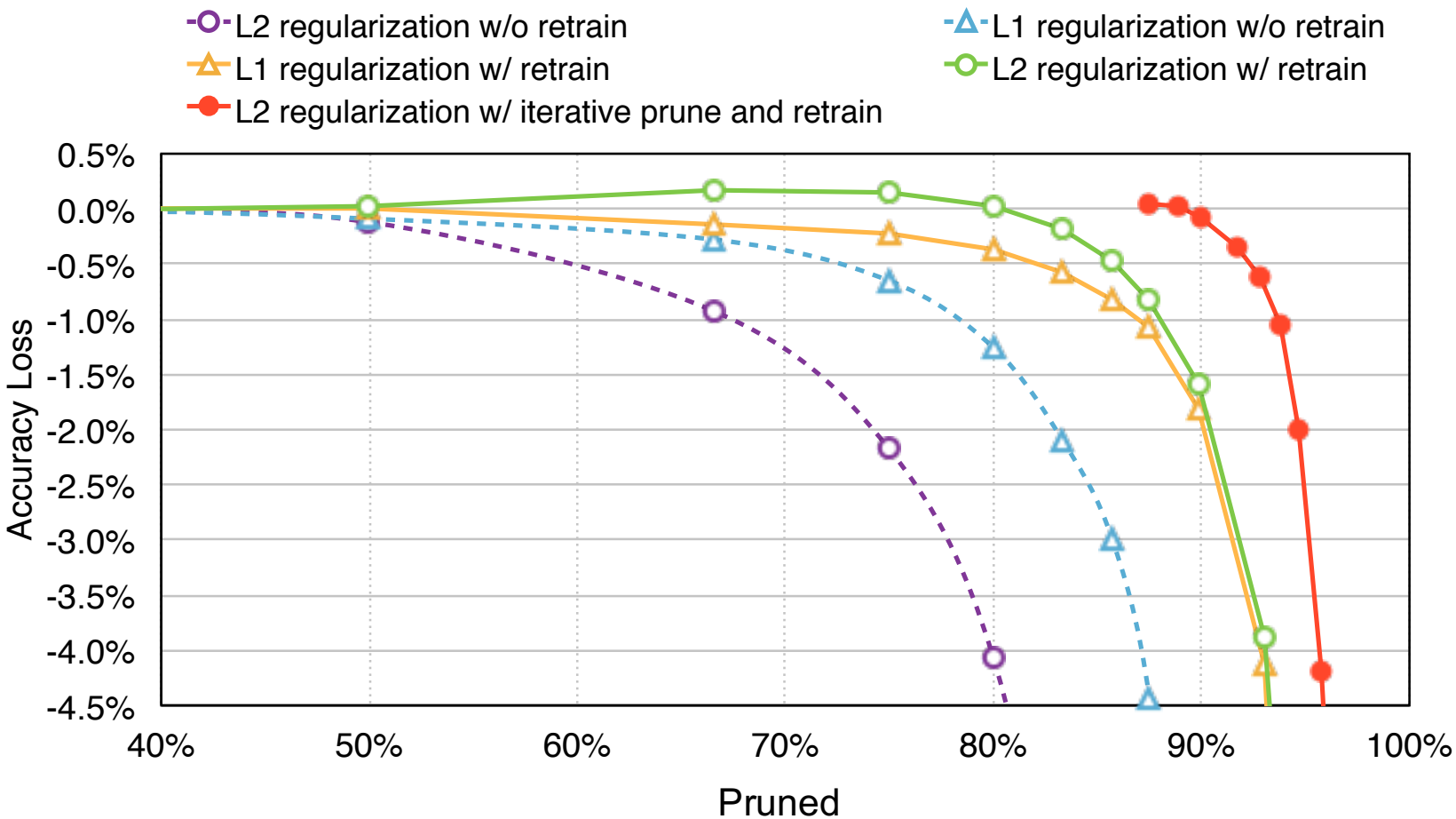
Sparsity



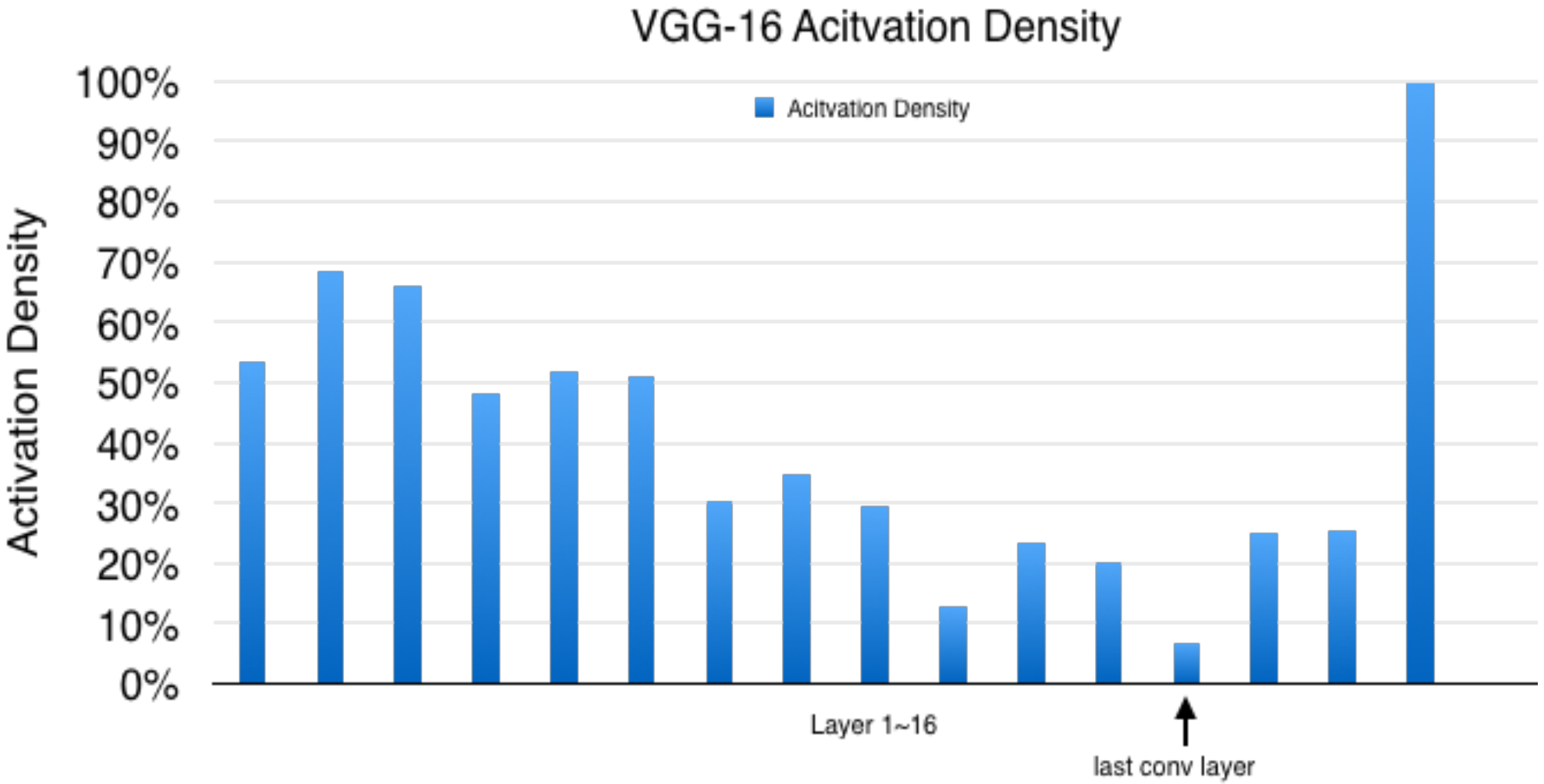
Han et al. Learning both Weights and Connections for Efficient Neural Networks, NIPS 2015

Retrain To Recover Accuracy

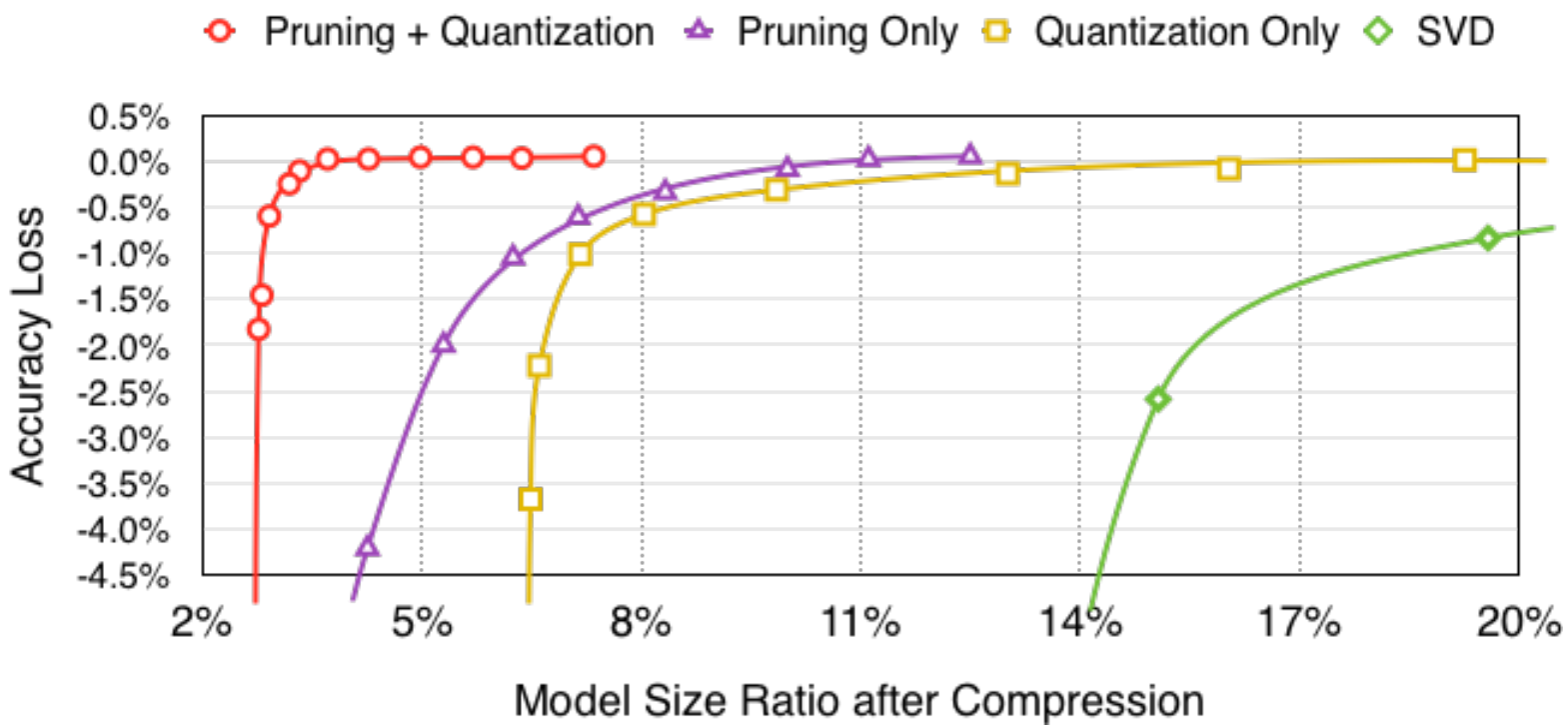




Han et al. Learning both Weights and Connections for Efficient Neural Networks, NIPS 2015

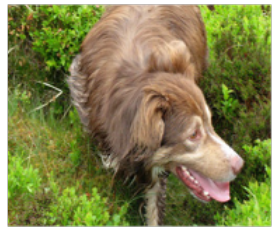

VGG-16 Acitvation Density
Acitvation Density
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Activation Density
Layer 1~16
last conv layer

Pruning + Trained Quantization



Pruning Neural Talk And LSTM

- **Original**: a basketball player in a white uniform is playing with a ball
- **Pruned 90%**: a basketball player in a white uniform is playing with a basketball



- **Original** : a brown dog is running through a grassy field
- **Pruned 90%**: a brown dog is running through a grassy area

- **Original** : a man is riding a surfboard on a wave
- **Pruned 90%**: a man in a wetsuit is riding a wave on a beach

- **Original** : a soccer player in red is running in the field
- **Pruned 95%**: a man in a red shirt and black and white black shirt is running through a field

Fixed-Function Hardware

Diannao (Electric Brain)

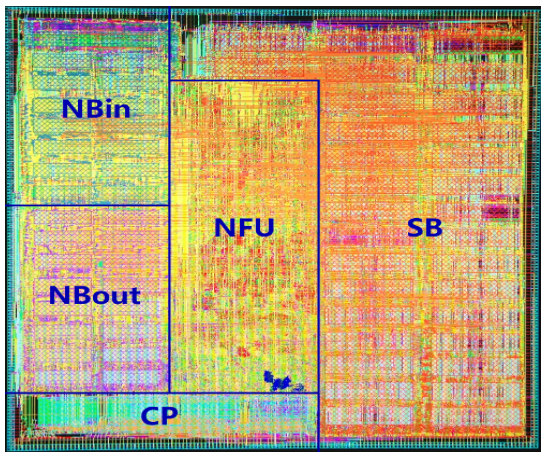


Figure 15. *Layout (65nm).*

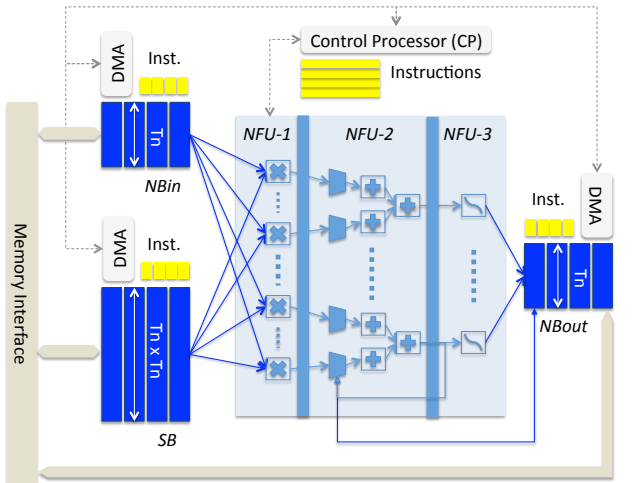


Figure 11. *Accelerator.*

Component or Block	Area in μm^2	(%)	Power in mW	(%)	Critical path in ns
ACCELERATOR	3,023,077		485		1.02
Combinational	608,842	(20.14%)	89	(18.41%)	
Memory	1,158,000	(38.31%)	177	(36.59%)	
Registers	375,882	(12.43%)	86	(17.84%)	
Clock network	68,721	(2.27%)	132	(27.16%)	
Filler cell	811,632	(26.85%)			
SB	1,153,814	(38.17%)	105	(22.65%)	
NBin	427,992	(14.16%)	91	(19.76%)	
NBout	433,906	(14.35%)	92	(19.97%)	
NFU	846,563	(28.00%)	132	(27.22%)	
CP	141,809	(5.69%)	31	(6.39%)	
AXIMUX	9,767	(0.32%)	8	(2.65%)	
Other	9,226	(0.31%)	26	(5.36%)	

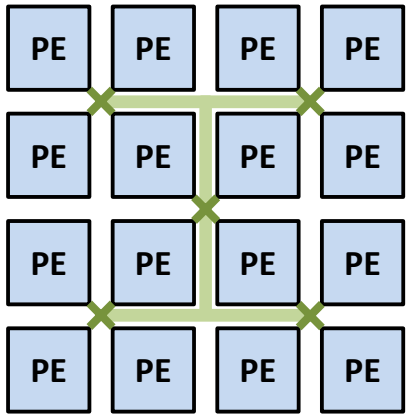
Table 6. *Characteristics of accelerator and breakdown by component type (first 5 lines), and functional block (last 7 lines).*

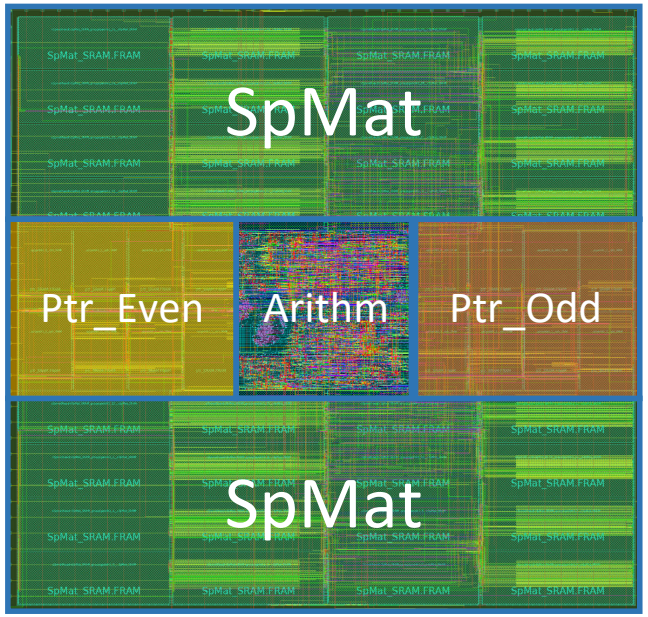
- Diannao improved CNN computation efficiency by using dedicated functional units and memory buffers optimized for the CNN workload.
- Multiplier + adder tree + shifter + non-linear lookup orchestrated by instructions
- Weights in off-chip DRAM
- 452 GOP/s, 3.02 mm^2 and 485 mW

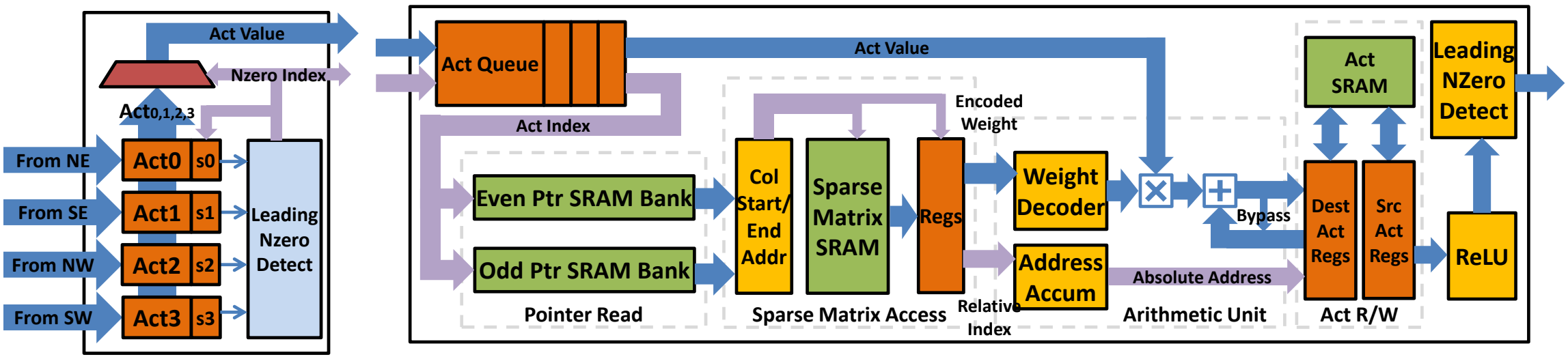
Chen et al. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning, ASPLOS 2014

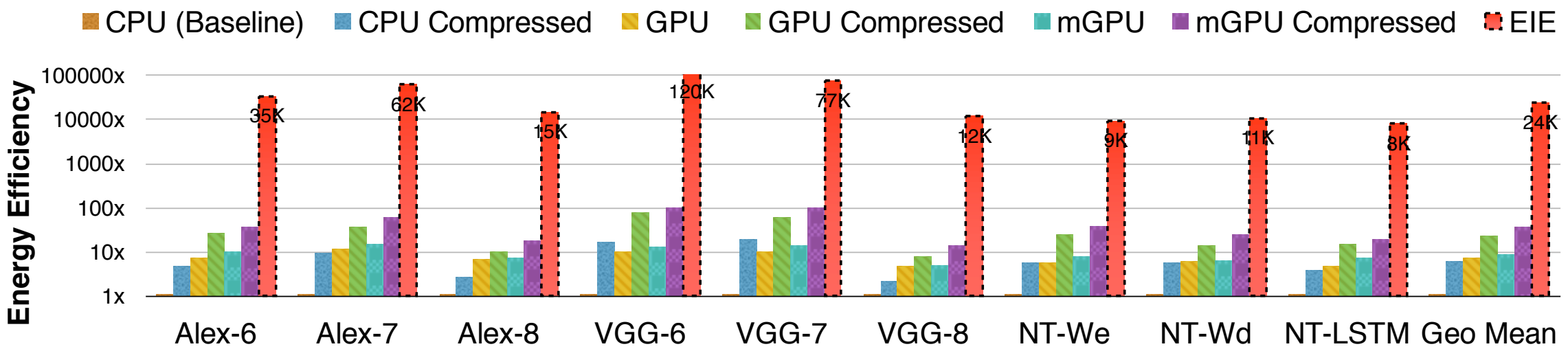
Efficient Inference Engine

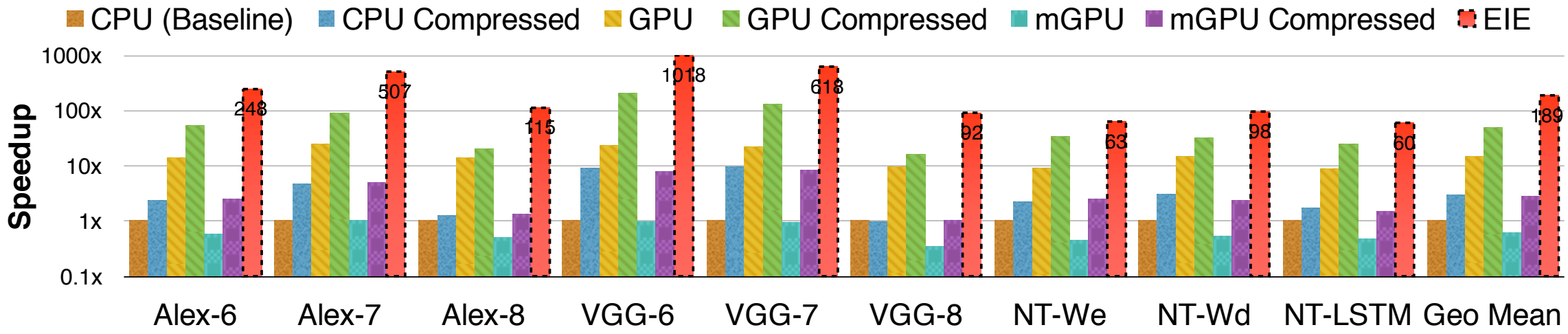
	Power (mW)	(%)	Area (μm^2)	(%)
Total	9.157		638,024	
memory	5.416	(59.15%)	594,786	(93.22%)
clock network	1.874	(20.46%)	866	(0.14%)
register	1.026	(11.20%)	9,465	(1.48%)
combinational	0.841	(9.18%)	8,946	(1.40%)
filler cell			23,961	(3.76%)
Act_queue	0.112	(1.23%)	758	(0.12%)
PtrRead	1.807	(19.73%)	121,849	(19.10%)
SpmatRead	4.955	(54.11%)	469,412	(73.57%)
ArithmUnit	1.162	(12.68%)	3,110	(0.49%)
ActRW	1.122	(12.25%)	18,934	(2.97%)
filler cell			23,961	(3.76%)

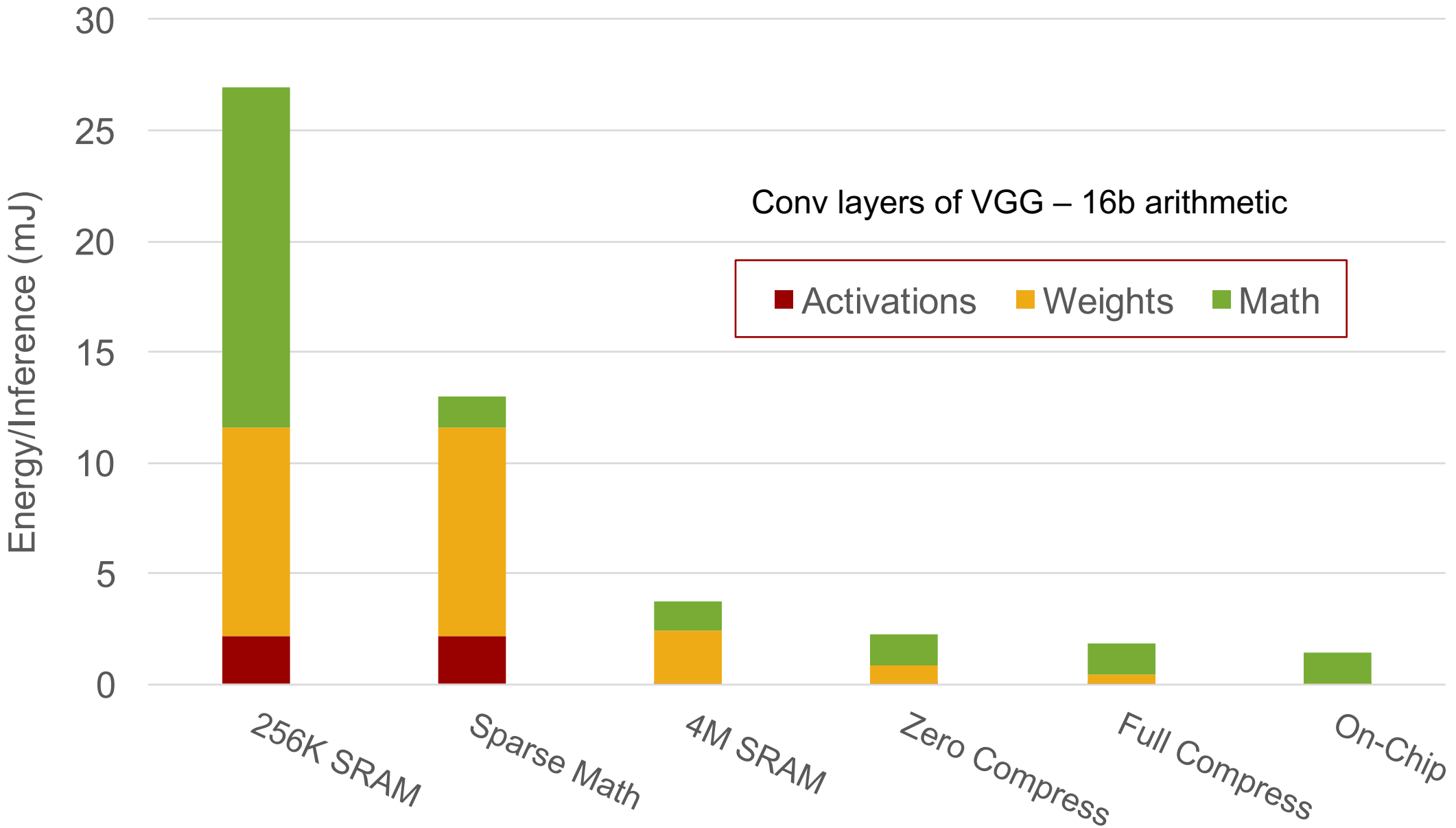


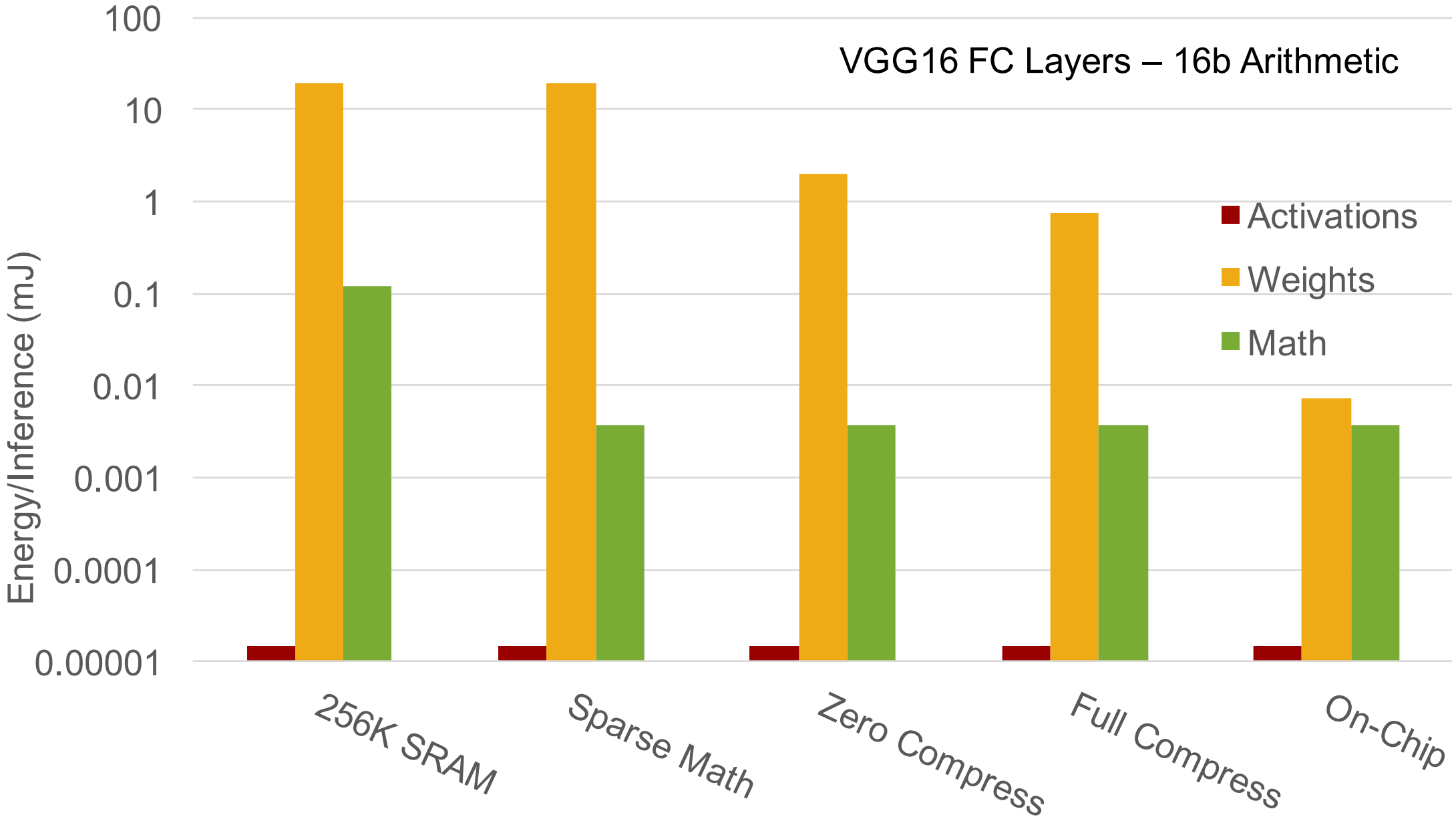





CPU (Baseline)
CPU Compressed
GPU
GPU Compressed
mGPU
mGPU Compressed
EIE
Energy Efficiency
100000x
10000x
1000x
100x
10x
1x
35K
62K
15K
120K
77K
12K
9K
11K
8K
24K
Alex-6
Alex-7
Alex-8
VGG-6
VGG-7
VGG-8
NT-We
NT-Wd
NT-LSTM
Geo Mean


CPU (Baseline)
CPU Compressed
GPU
GPU Compressed
mGPU
mGPU Compressed
EIE
Speedup
1000x
100x
10x
1x
0.1x
248
507
115
1018
618
92
63
98
60
189
Alex-6
Alex-7
Alex-8
VGG-6
VGG-7
VGG-8
NT-We
NT-Wd
NT-LSTM
Geo Mean


Energy/Inference (mJ)
30
25
20
15
10
5
0
Conv layers of VGG – 16b arithmetic
Activations
Weights
Math
256K SRAM
Sparse Math
4M SRAM
Zero Compress
Full Compress
On-Chip


VGG16 FC Layers – 16b Arithmetic
Energy/Inference (mJ)
100
10
1
0.1
0.01
0.001
0.0001
0.00001
Activations
Weights
Math
256K SRAM
Sparse Math
Zero Compress
Full Compress
On-Chip

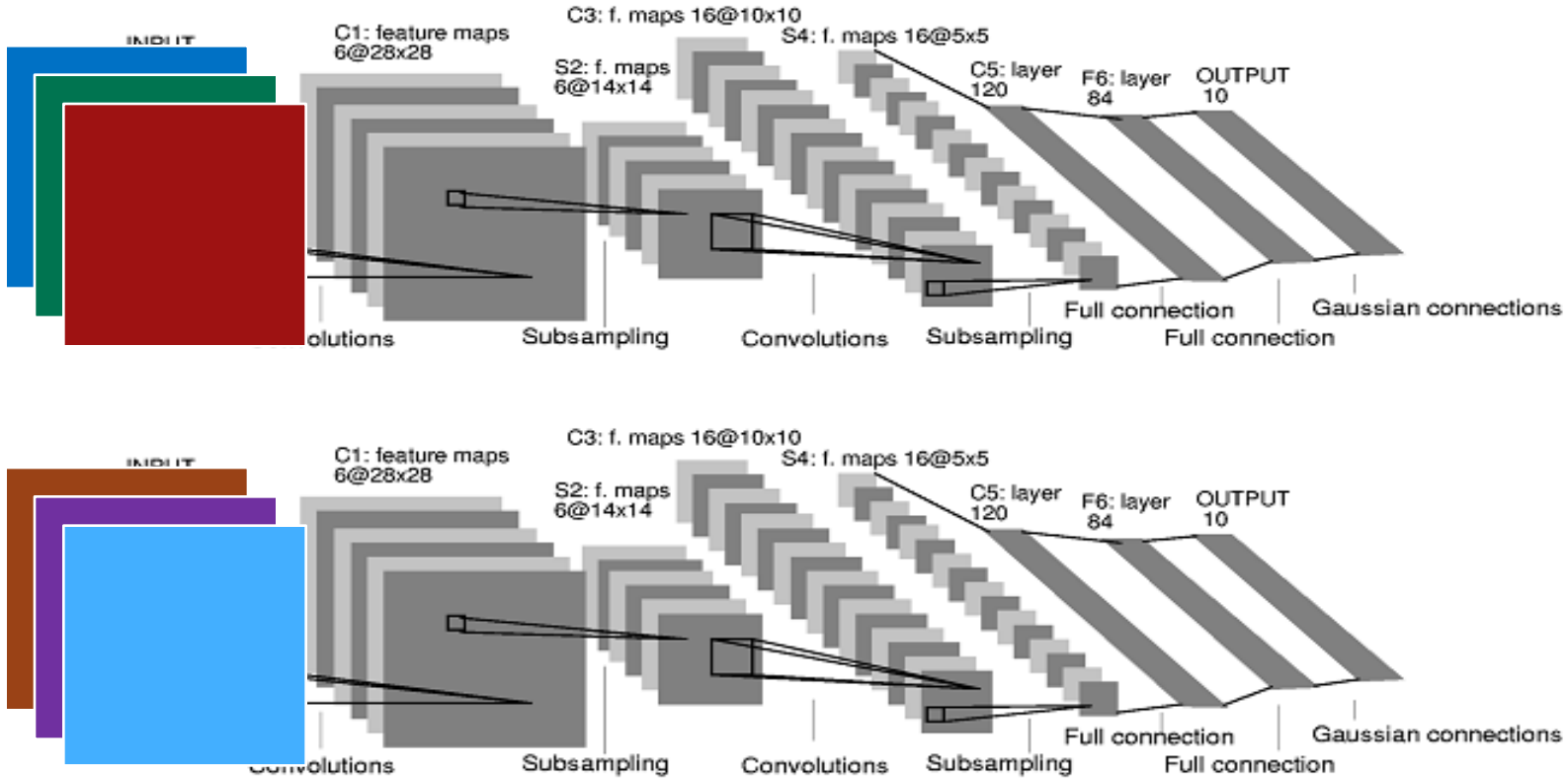
Inference Summary

- Prune 70-95% of static weights (3-20x reduction in size and ops) [3-20x]
- Exploit dynamic sparsity of activations (3x reduction in ops) [10-60x]
- Reduce precision to ~4 bits (8x size reduction) [80-480x]
- 25-150x smaller fits in smaller RAM (100x less power) [8,000-48,000x]
- Dedicated hardware to eliminate overhead
  - Facilitates compression and sparsity
  - Exploit locality

Sparsity, Precision, Locality

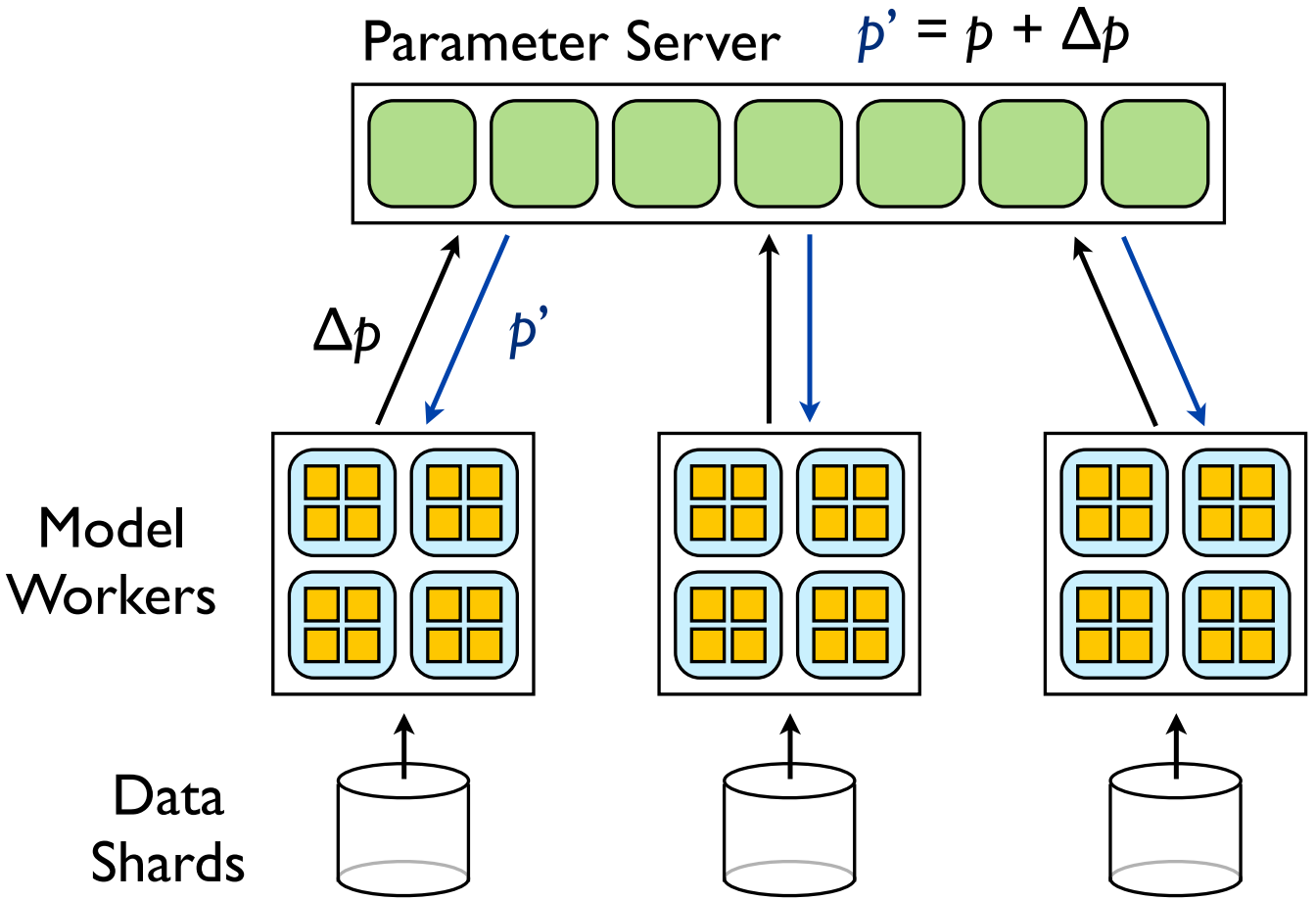
Faster Training through Parallelism

Data Parallel – Run Multiple Inputs In Parallel



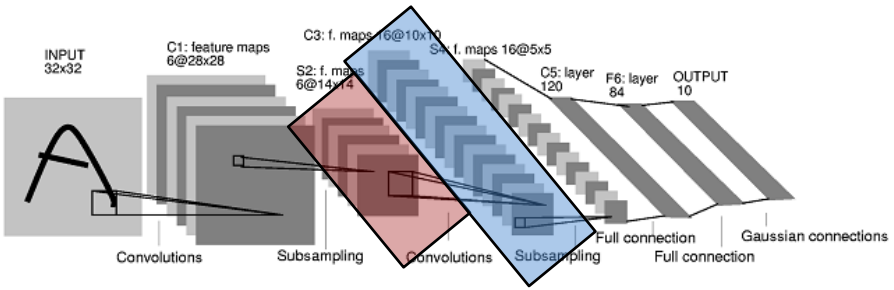
- **Doesn't affect latency for one input**
- **Requires P-fold larger batch size**
- **For training requires coordinated weight update**

Parameter Update

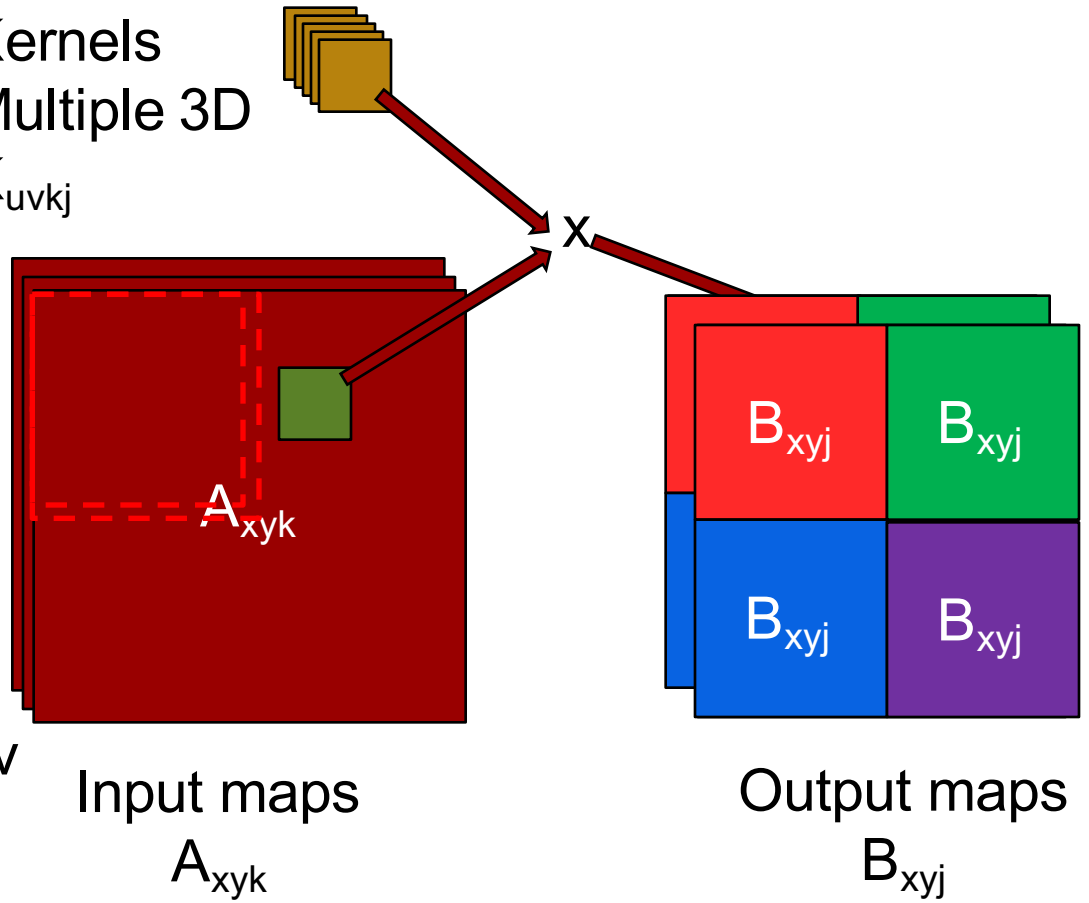


Large Scale Distributed Deep Networks, Jeff Dean et al., 2013

Model-Parallel Convolution – by output region (x,y)



Kernels
Multiple 3D
K_{uvkj}

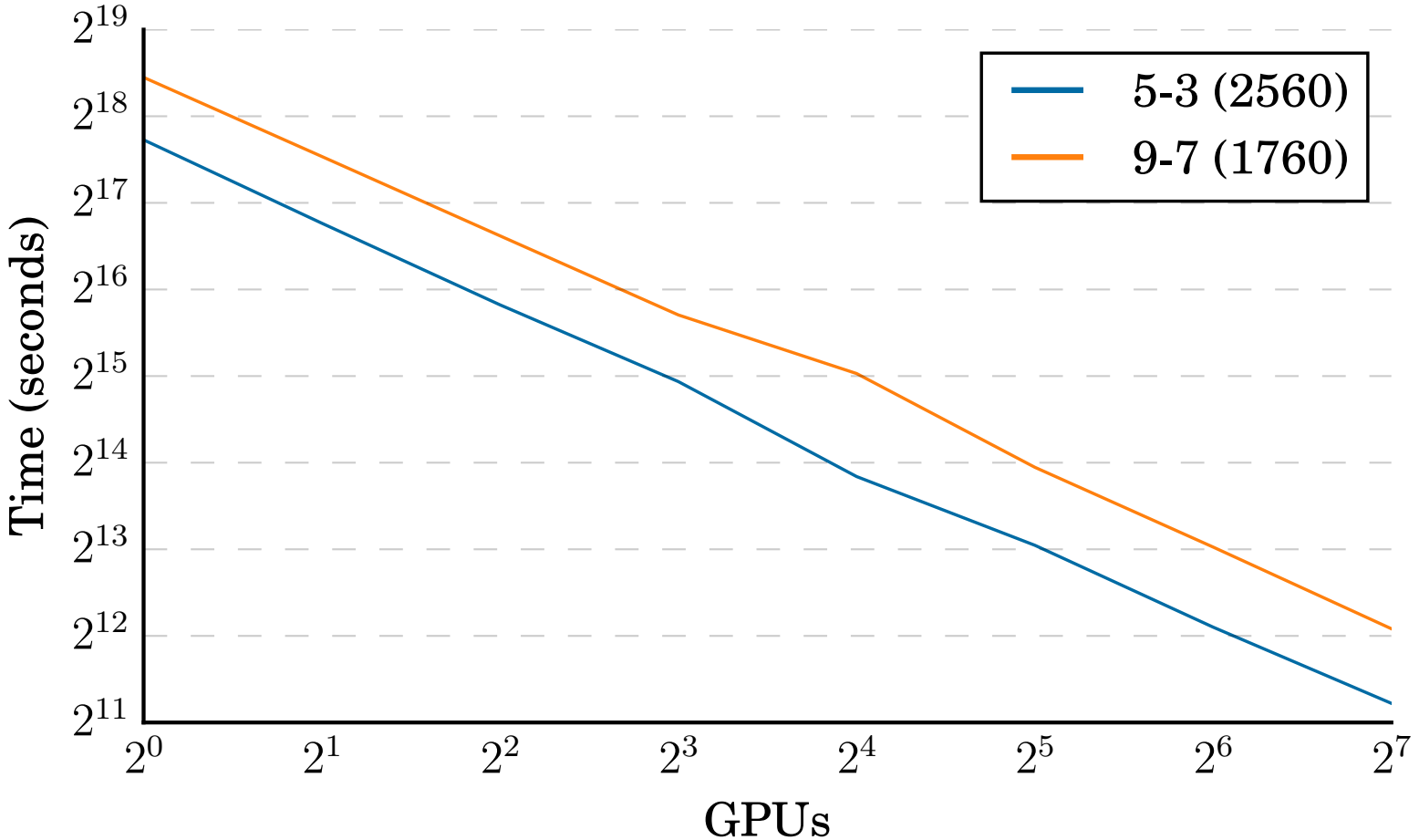


6D Loop
Forall region XY
  For each output map j
    For each input map k
      For each pixel x,y in XY
        For each kernel element u,v
          B_{xyj} += $A_{(x-u)(y-v)k}$ x K_{uvkj}

Input maps
A_{xyk}

Output maps
B_{xyj}

Parallel GPUs on Deep Speech 2



Baidu, Deep Speech 2: End-to-End Speech Recognition in English and Mandarin, 2015

4 Distinct Sub-problems

	Training	Inference	
Convolutional	32b FP Batch Activation Storage GPUs ideal Comm for Parallelism	Low-Precision Compressed Latency-Sensitive Fixed-Function HW Arithmetic Dominated	B x S Weight Reuse Act Dominated
Fully-Conn.	32b FP Batch Weight Storage GPUs ideal Comm. for Parallelism	Low-Precision Compressed Latency-Sensitive No weight reuse Fixed-Function HW Storage dominated	B Weight Reuse Weight Dominated
	32b FP – large batches Minimize Training Time Enables larger networks	8b Int – small (unit) batches Meet real-time constraint	

Summary

- Fixed-function hardware will dominate inference (100-10,000x gain)
  - Sparse, low-precision, compressed (25-150x smaller)
  - 3x dynamic sparsity
  - All weights and activations from local memory (10-100x less energy)
  - Flexible enough to track evolving algorithms

- GPUs will dominate training
  - Only dynamic sparsity (3x activations, 2x dropout)
  - Medium precision (FP16 – for weights), stochastic rounding
  - Large memory footprint (batch x retained activations)
  - Communication BW scales with parallelism