The Explosion in Neural Network Hardware

Arm Summit,
Cambridge, September 17th, 2018

Trevor Mudge

Bredt Family Professor of Computer Science and Engineering

The University of Michigan, Ann Arbor

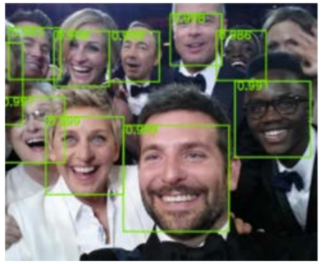
What Just Happened?

- For years the common wisdom was that Hardware was a bad bet for a venture
- That has changed
- More than 45 start-ups are designing chips for image processing, speech, and self-driving cars
- 5 have raised more than $100 million
- Venture capitalists have invested over $1.5 billion in chip start-ups last year

Driving Factors

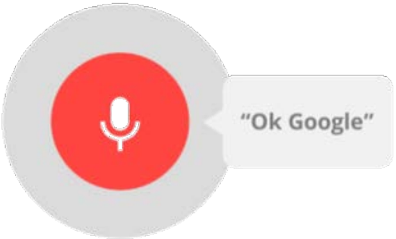
- Pragmatism— “unreasonable” success of neural nets
- Slowing of Moore’s Law has made accelerators more attractive
- Algorithms similar to an existing paradigm
- Existing accelerators could easily be repurposed—GPUs and DSPs
- Orders of magnitude increase in the size of data sets
- Independent Foundries—TSMC is perhaps the best known

What Are Neural Networks Used For?



Computer vision

Self-driving cars



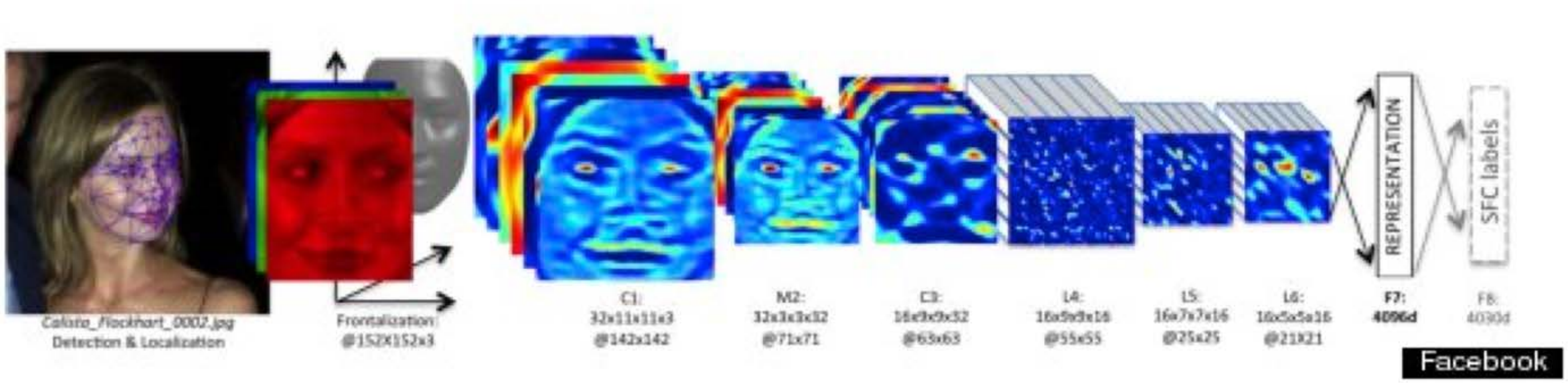
Keyword Spotting

Seizure Detection

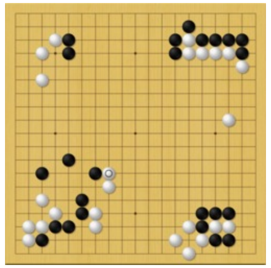
- A unifying approach to “understanding” –in contrast to an expert guided set of algorithms to recognize faces for example
- Their recent success is based on the availability of enormous amounts of training data

Notable Successes

- Facebooks Deep Face is 97.35% accurate on the “Labeled Faces in the Wild” (LFW) dataset—as good than a human in some cases

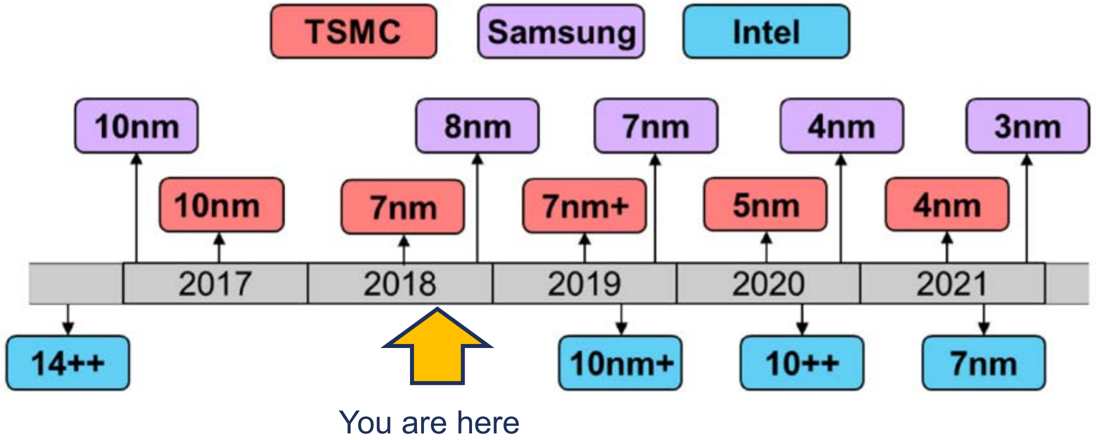


- Recent attention grabbing application—DeepMind’s AlphaGO
  - It beat European Go champion Fan Hui in October 2015
  - It was powered by Google’s Tensor Processing Unit (TPU 1.0)
  - TPU 2.0 beat Ke Jie, the world no. 1 GO player May 2017
  - AlphaZero improved on that by playing itself
  - More than just NNs



Slowing of Moore’s Law ⇒ Accelerators

- Power scaling—ended a long time ago
- Cost per transistor scaling—more recently
- Technical limits—still has several nodes to go
- 2nm may not be worth it—EE Times 3/23/18
- Time between nodes increasing

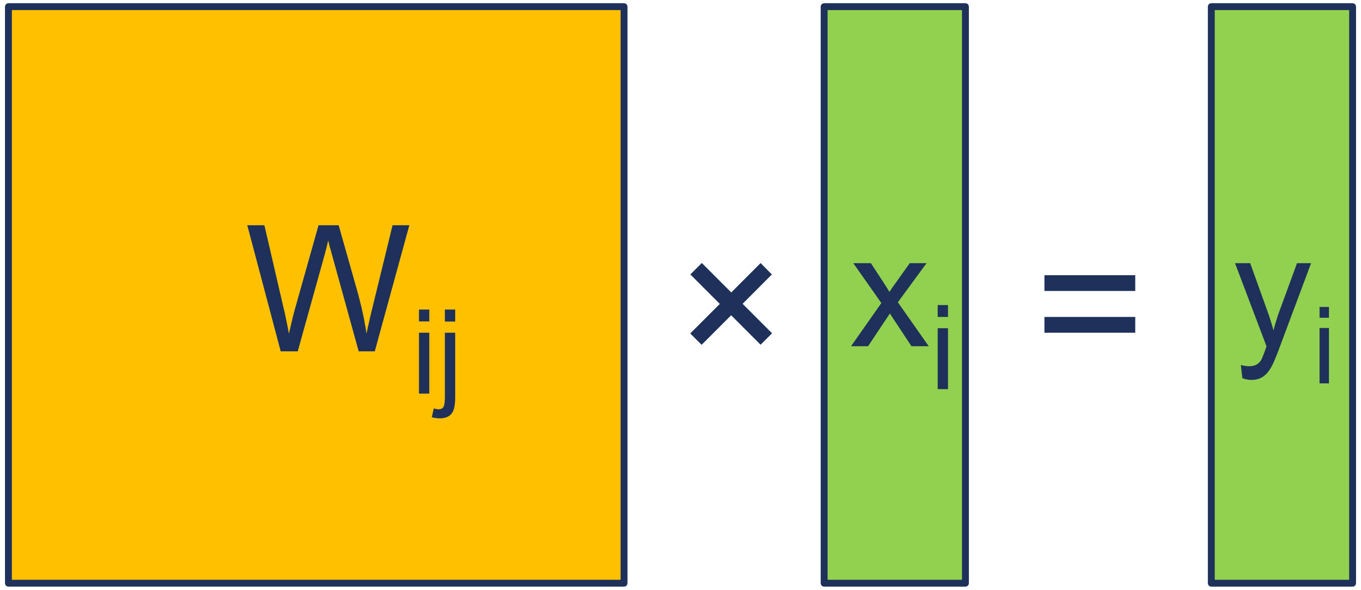


(Source: The Linley Group with future nodes being Linley estimates)

- ROI a show stopper—8/28/18 GlobalFoundries halts 7nm work
- Next FinFET node would have cost $2-4B

Algorithms Fit Existing Paradigm

- Algorithms fitted an existing paradigm—variations on dense matrix-vector multiply



$$W_{ij} \times x_i = y_i$$

Orders of Magnitude Increase in Data

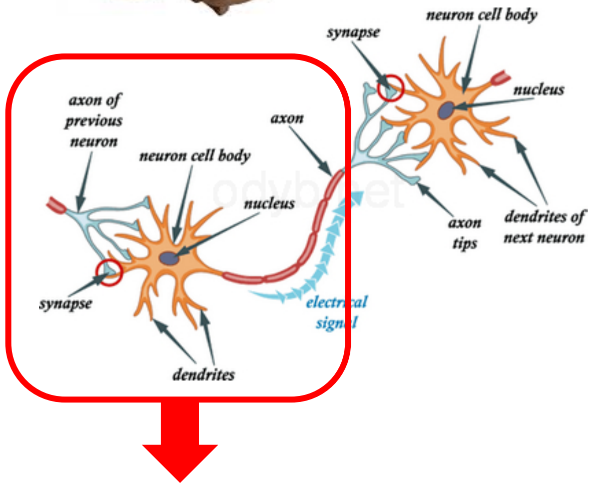
- Orders of magnitude increase in the size of data sets
- Google /Facebook / Baidu / etc. have access to vast amounts of data and this has been the game changer
- FAANGs (Facebook/Amazon/Apple/Netflix/Google) have access to vast amounts of data and this has been the game changer
- Add to that list: Baidu/Microsoft/Alibaba/Tencent/FSB (!)
- Available to 3rd parties—Cambridge Analytica (deceased!)
- Open Source
    - AlexNet—image classification (CNN)
    - VGG-16—large-scale image recognition (CNN)
    - Deep Residual Network—Microsoft
    - Proposed MLPerf—Google/Biadu led consortium

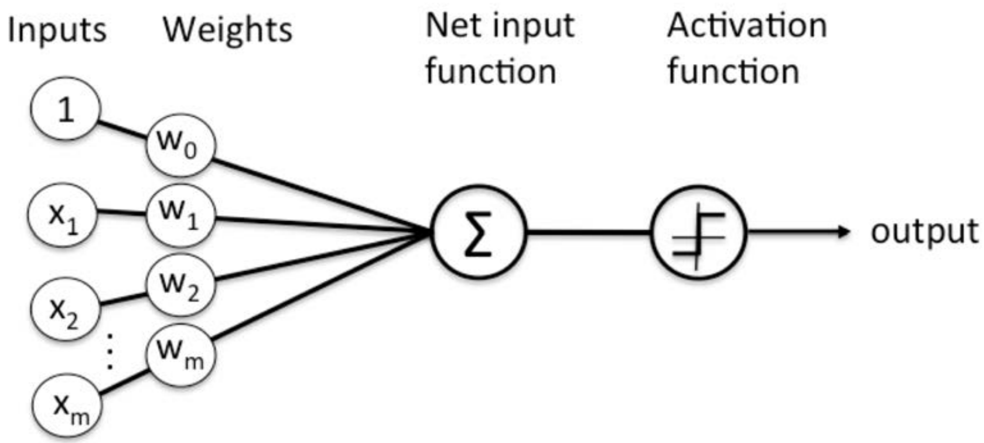
What are Neural Nets—NNs

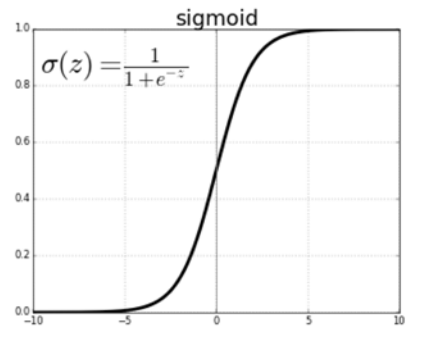
NEURON

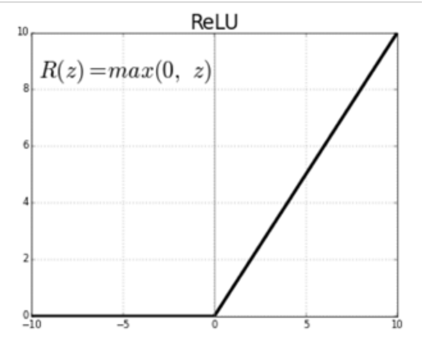
- Unfortunate anthropomorphization!
- Only a passing relationship to the neurons in your brain
- Neuron shown with (synaptic) weighted inputs feeding dendrites!
- The net input function is just a **dot-product**
- The "activation" function is a **non-linear** function
- Often simplified to the rectified linear unit—ReLU

mandatory brain picture









What are Neural Nets—5 Slide Introduction!

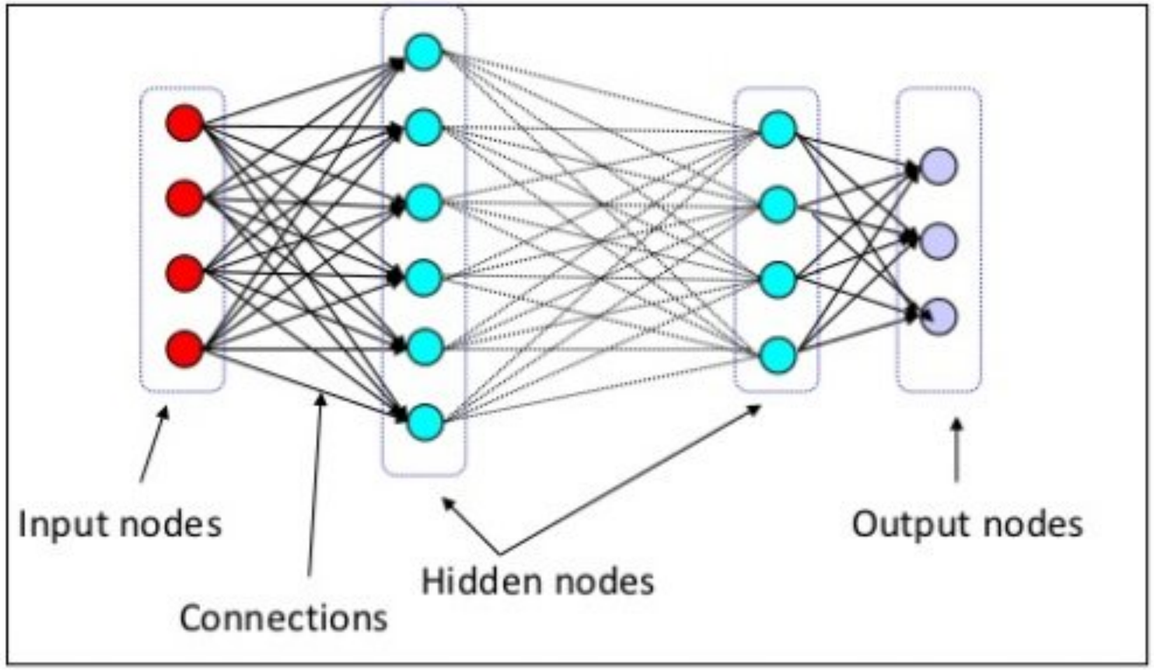
NEURAL NETS

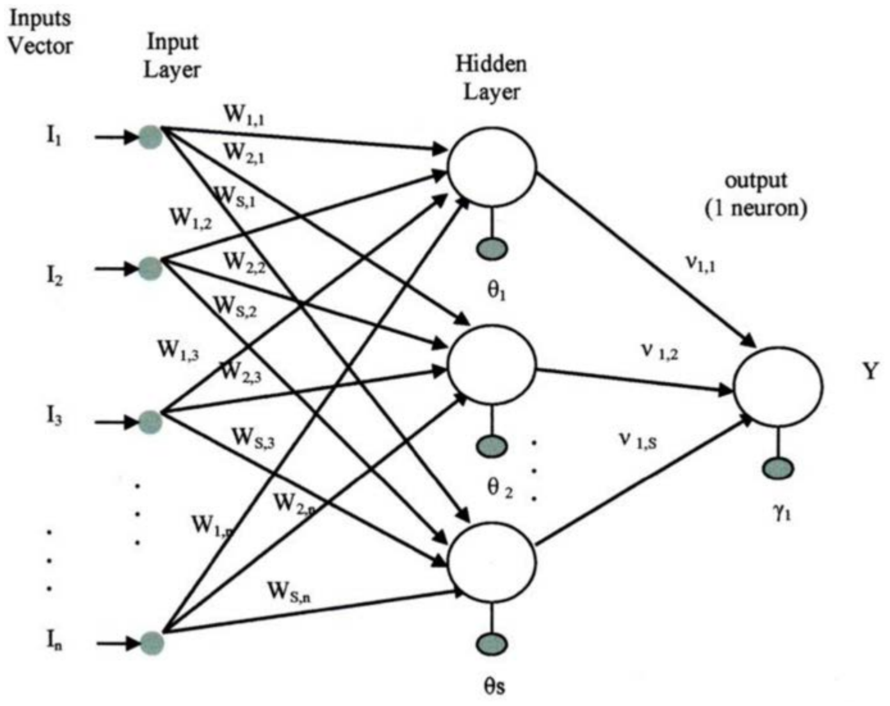
- From input to first hidden layer is a matrix-vector multiply with a weight matrix

  $$W \otimes \underline{\text{input}} = \underline{V}$$

- Deep Neural Nets (DNNs) have multiple hidden layers

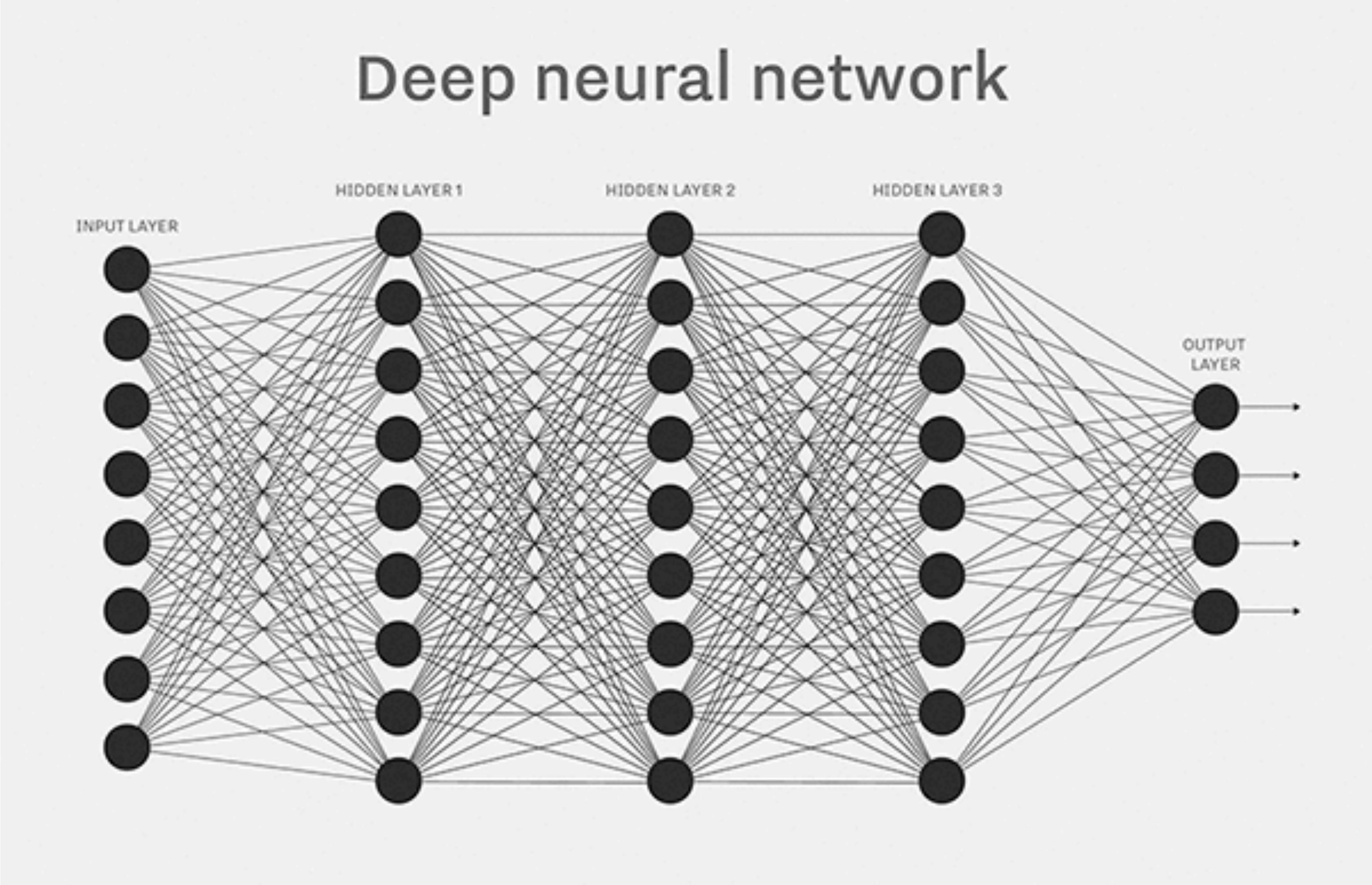
  $$\text{output} = \ldots \otimes W_3 \otimes W_2 \otimes W_1 \otimes \underline{\text{input}}$$





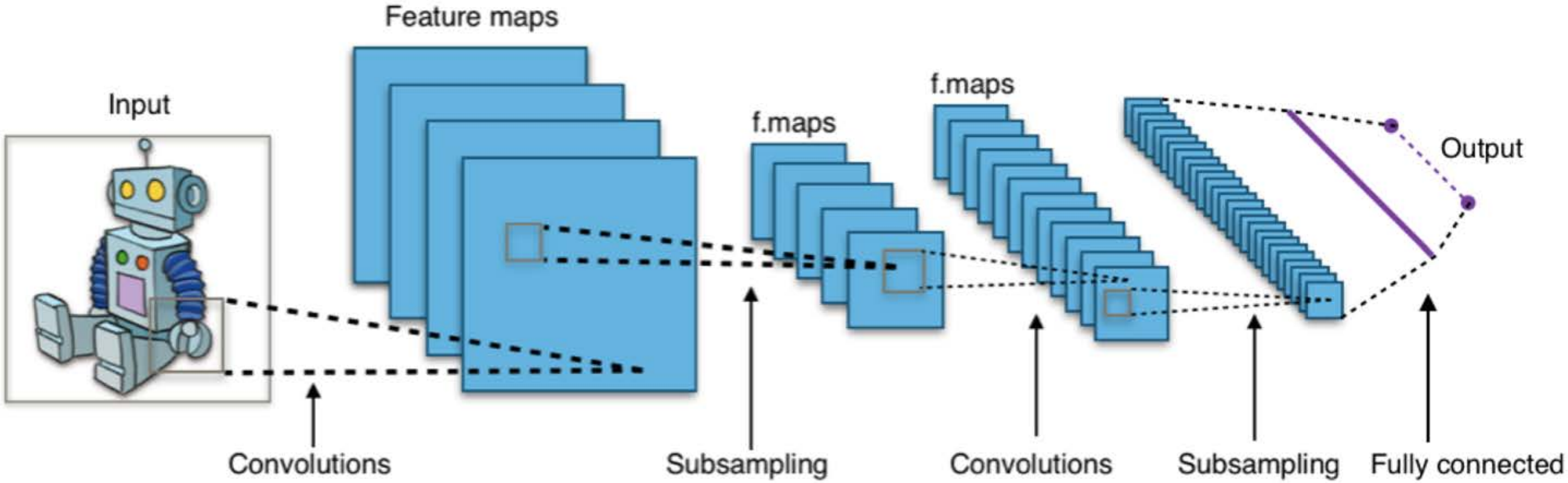
DNN—deep neural networks

- DNNs have more than two levels that are “fully connected”
- Bipartite graphs
- Dense matrix operations
- Other varies of NNs that depend on fast dot products:
  - CNNs—convolutional NNs
  - RNN—recurrent NNs
  - LSTM—long short-term memory



CNN—convolutional neural networks

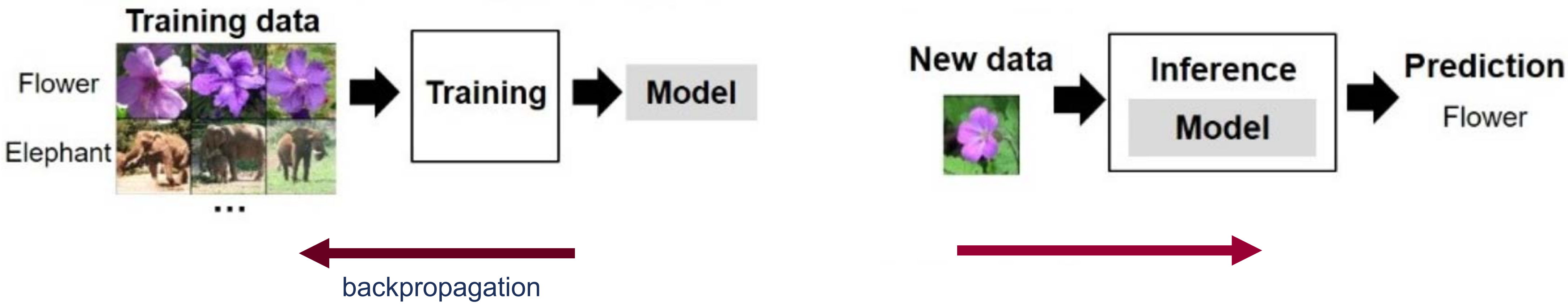
- Borrowed an idea from signal processing
- Used typically in image applications
- Cuts down on dimensionality



- The 4 feature maps are produced as a result of 4 convolution kernels being applied to the image array

Training and Inference

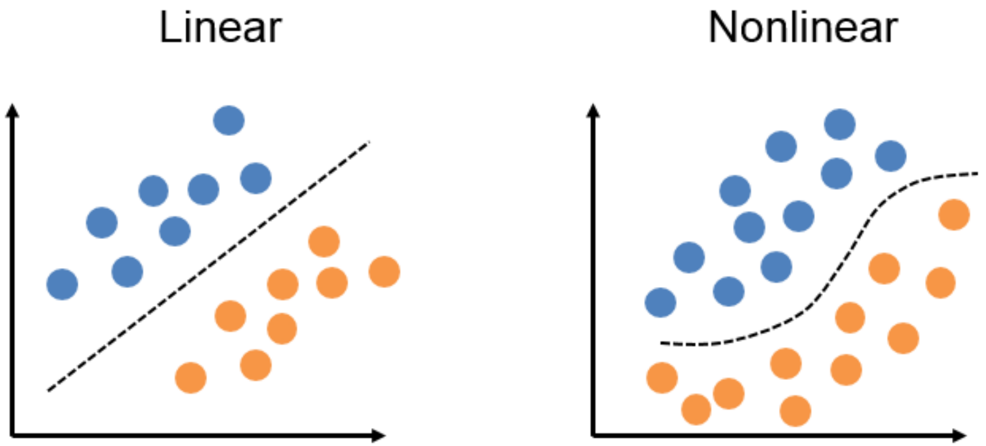
- The weights come from the learning or training phase
- Start with randomly assigned weights and “learn” through a process of successive approximation that typically involves back propagation with (stochastic) gradient descent
- Both processes involve matrix-vector multiplication
- Inference is done much more frequently
- Often inference uses fixed point and training uses floating point



Summary

- Basic Algorithm is a vector-matrix multiply

$$\ldots \otimes W_3 \otimes W_2 \otimes W_1 \otimes \underline{\text{input}}$$



- The number of weigh matrices corresponds to the depth of the network—the rank of the matrices can be in the millions
- The non-linear operator ⊗ prevents us from pre-evaluating the matrix products—a **significant** inefficiency
- BUT it makes possible non-linear separation in classification space
- The basic operation is a dot product followed by a non-linear operation—a MAC operation and some sort of thresholding

$$threshold(\sum W_{ik} \times input_k)$$

Summary—Note on pre-evaluation

- Basic Algorithm is a vector-matrix multiply

$$\ldots \otimes W_3 \otimes W_2 \otimes W_1 \otimes \underline{\text{input}}$$

- The product is a function of $\underline{\text{input}}$
- If $\otimes$ were simply normal matrix multiply $\bullet$ then

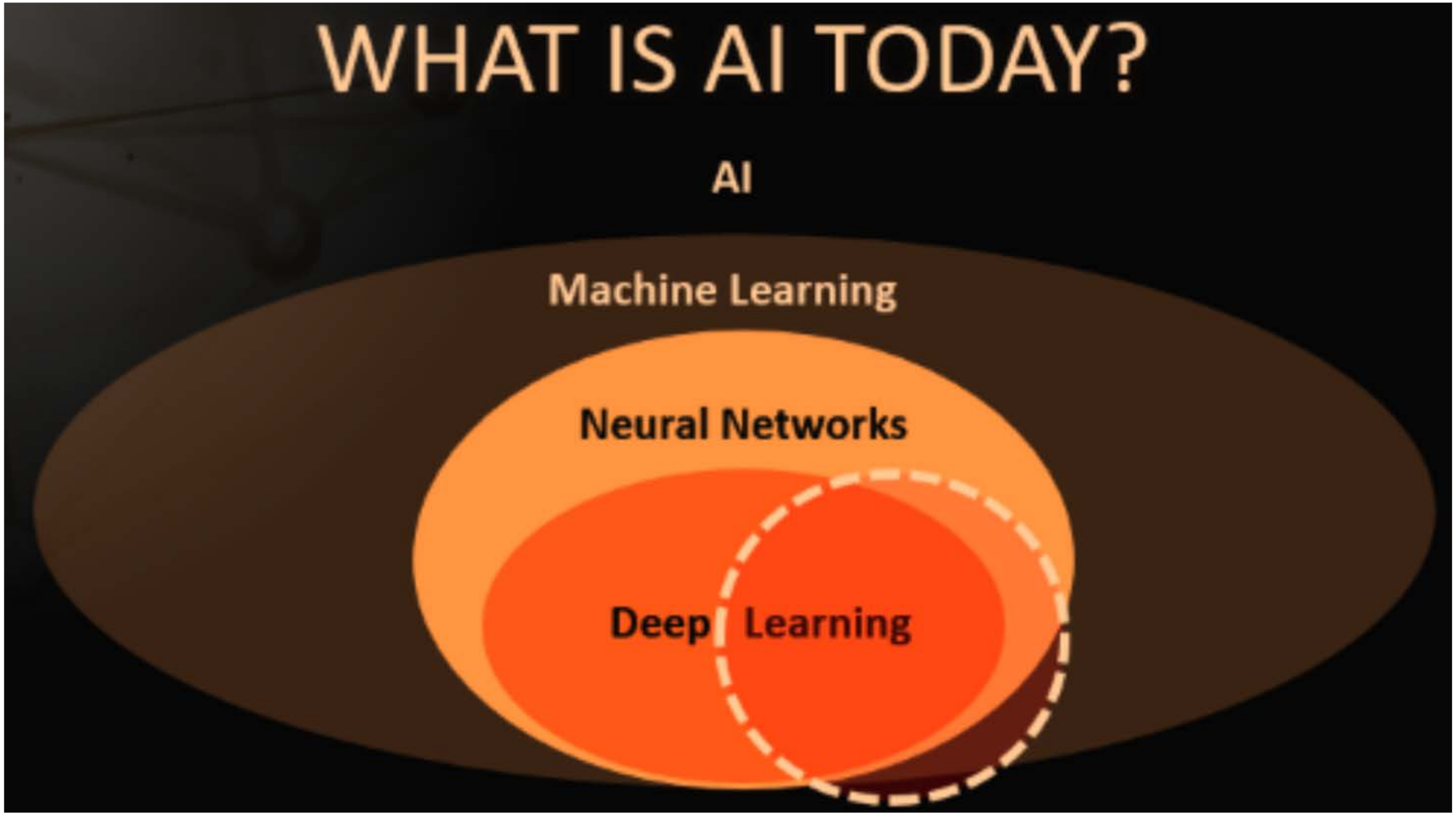
$$\ldots W_3 \bullet W_2 \bullet W_1 \bullet \underline{\text{input}}$$

Can be written $W \bullet \underline{\text{input}}$

Where $W = \ldots W_3 \bullet W_2 \bullet W_1$

- The inference step would be just ONE matrix multiply
- Question: Can we use $(W_2 \otimes W_1 \otimes \underline{\text{input}} - W_2 \bullet W_1 \bullet \underline{\text{input}})$ for representative samples of $\underline{\text{input}}$ as an approximate correction

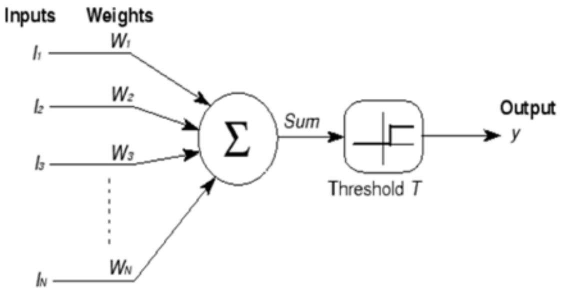
Classification—often mischaracterized as AI



(Source: Intel)

What’s Changed?

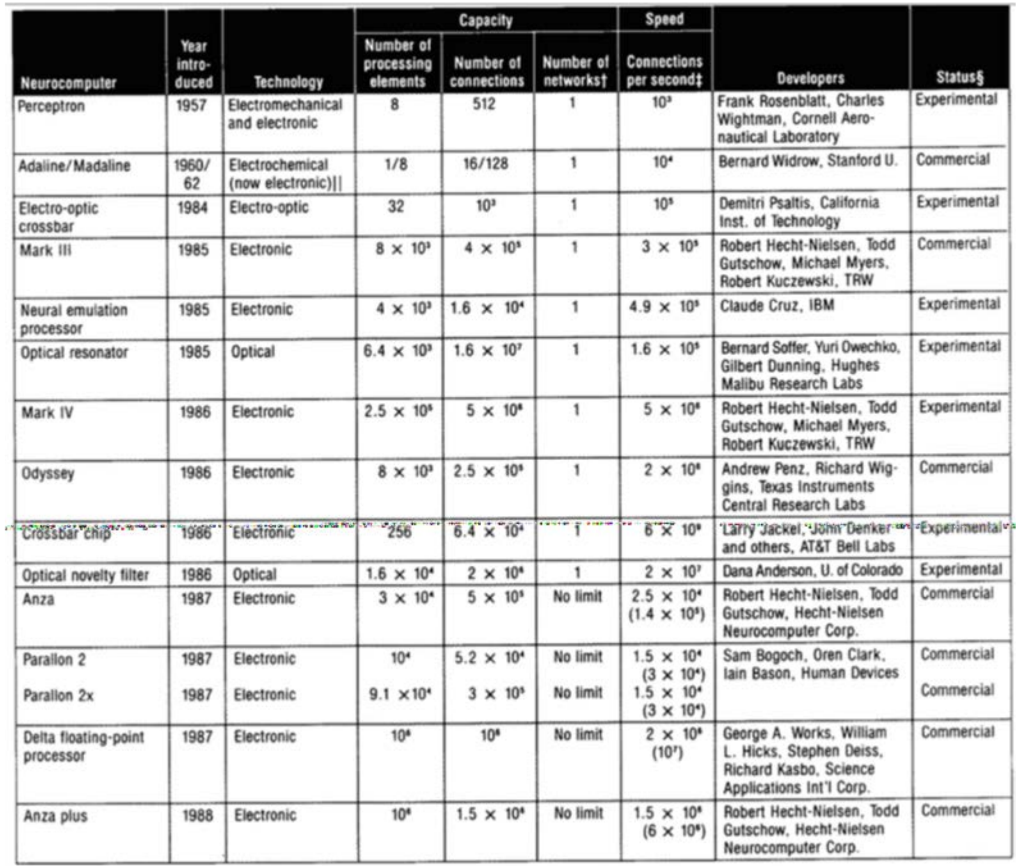
- Neural nets have been around for over 70 years—eons in computer-evolution time
  - McCulloch–Pitts Neurons—1943



- Countless innovations but the basic idea is quite old
  - Notably back propagation to learn weights in supervised learning
  - Convolutional NN—nearest neighbor convolution layer
  - Recurrent NN—feedback added
  - Long short-term memory—state added

- Massive improvements in Compute Power & More Data
- Larger, deeper, better
  - AlexNet
    - 8 layers, 240MB weights
  - VGG-16
    - 16 layers, 550MB weights
  - Deep Residual Network
    - 152 layers, 229MB weights

Best NNs in 1988—Hecht-Nielsen (?)



			Capacity			Speed		
Neurocomputer	Year introduced	Technology	Number of processing elements	Number of connections	Number of networks†	Connections per second‡	Developers	Status§
Perceptron	1957	Electromechanical and electronic	8	512	1	10^3	Frank Rosenblatt, Charles Wightman, Cornell Aeronautical Laboratory	Experimental
Adaline/Madaline	1960/62	Electrochemical (now electronic)‖	1/8	16/128	1	10^4	Bernard Widrow, Stanford U.	Commercial
Electro-optic crossbar	1984	Electro-optic	32	10^3	1	10^5	Demitri Psaltis, California Inst. of Technology	Experimental
Mark III	1985	Electronic	8×10^3	4×10^5	1	3×10^5	Robert Hecht-Nielsen, Todd Gutschow, Michael Myers, Robert Kuczewski, TRW	Commercial
Neural emulation processor	1985	Electronic	4×10^3	1.6×10^4	1	4.9×10^5	Claude Cruz, IBM	Experimental
Optical resonator	1985	Optical	6.4×10^3	1.6×10^7	1	1.6×10^5	Bernard Soffer, Yuri Owechko, Gilbert Dunning, Hughes Malibu Research Labs	Experimental
Mark IV	1986	Electronic	2.5×10^5	5×10^6	1	5×10^6	Robert Hecht-Nielsen, Todd Gutschow, Michael Myers, Robert Kuczewski, TRW	Experimental
Odyssey	1986	Electronic	8×10^3	2.5×10^5	1	2×10^6	Andrew Penz, Richard Wiggins, Texas Instruments Central Research Labs	Commercial
Crossbar chip	1986	Electronic	256	6.4×10^4	1	6×10^9	Larry Jackel, John Denker and others, AT&T Bell Labs	Experimental
Optical novelty filter	1986	Optical	1.6×10^4	2×10^8	1	2×10^7	Dana Anderson, U. of Colorado	Experimental
Anza	1987	Electronic	3×10^4	5×10^5	No limit	2.5×10^4 (1.4×10^5)	Robert Hecht-Nielsen, Todd Gutschow, Hecht-Nielsen Neurocomputer Corp.	Commercial
Parallon 2	1987	Electronic	10^4	5.2×10^4	No limit	1.5×10^4 (3×10^4)	Sam Bogoch, Oren Clark, Iain Bason, Human Devices	Commercial
Parallon 2x	1987	Electronic	9.1×10^4	3×10^5	No limit	1.5×10^4 (3×10^4)		Commercial
Delta floating-point processor	1987	Electronic	10^6	10^6	No limit	2×10^6 (10^7)	George A. Works, William L. Hicks, Stephen Deiss, Richard Kasbo, Science Applications Int'l Corp.	Commercial
Anza plus	1988	Electronic	10^6	1.5×10^6	No limit	1.5×10^6 (6×10^6)	Robert Hecht-Nielsen, Todd Gutschow, Hecht-Nielsen Neurocomputer Corp.	Commercial

Convergence—what is the common denominator?

- Dot product for dense matrix operations—MAC units
- Take away for computer architects:
- Dense => vector processing
- We know how to do this
- Why not use existing—repurpose
- There are still opportunities
  - Size and power
  - Systolic-type organizations
  - Tailor precision to the application

Who’s On The Bandwagon?

Recall:

- More than 45 start-ups are designing chips for image processing, speech, and self-driving cars
- 5 have raised more than $100 million
- Venture capitalists have invested over $1.5 billion in chip start-ups last year

- These numbers are conservative

Just Some of the Offerings

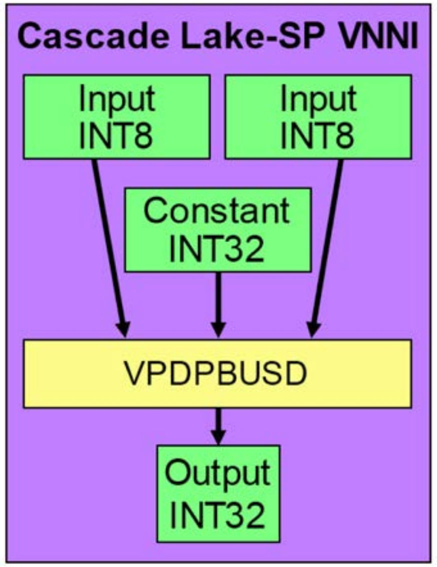
- Two Approaches
  - Repurpose a signal processing chip or a GPU—CEVA & nVidia
  - Start from scratch—Google’s TPU & now nVidia is claiming a TPU in the works
- Because the key ingredient is a dot product hardware to do this has existed for decades—DSP MACs
- Consequently everyone in the DSP space claims they have a DNN solution!
- Some of the current offerings and their characteristics
  - Intel—purchased Nervana and Movidius
    - Possible use of the Movidius accelerator in Intel’s future PC chip sets
  - Wave—45 person start up with DSP expertise
  - TPU—disagrees with M/soft FPGA solution and nVidia’s GPU solution
  - CEVA-XM6-based vision platform
  - nVidia—announced a TPU-like processor
    - Tesla for training
  - Graphcore's Intelligent Processor Unit (IPU)
    - TSMC—no details, has “very high” memory bandwidth 8 bit arithmetic
  - FIVEAI from GraphCore
  - Apple’s Bionic neural engine in the A11 SoC in its iPhone
  - The DeePhi block in Samsung’s Exynos 9810 in the Galaxy S9
  - The neural engine from China’s Cambricon in Huawei’s Kirin 970 handset

Landscape for Hardware Offerings

- Training tends to use heavy-weight GPGPUs
- Inference uses smaller engines
- Inference is now being done in mobile platforms
- Four solutions:
  - Repurposed CPUs
  - ASICs
  - FPGAs
  - Analog
  - Academia

Repurposed CPU—Intel Cascade Lake

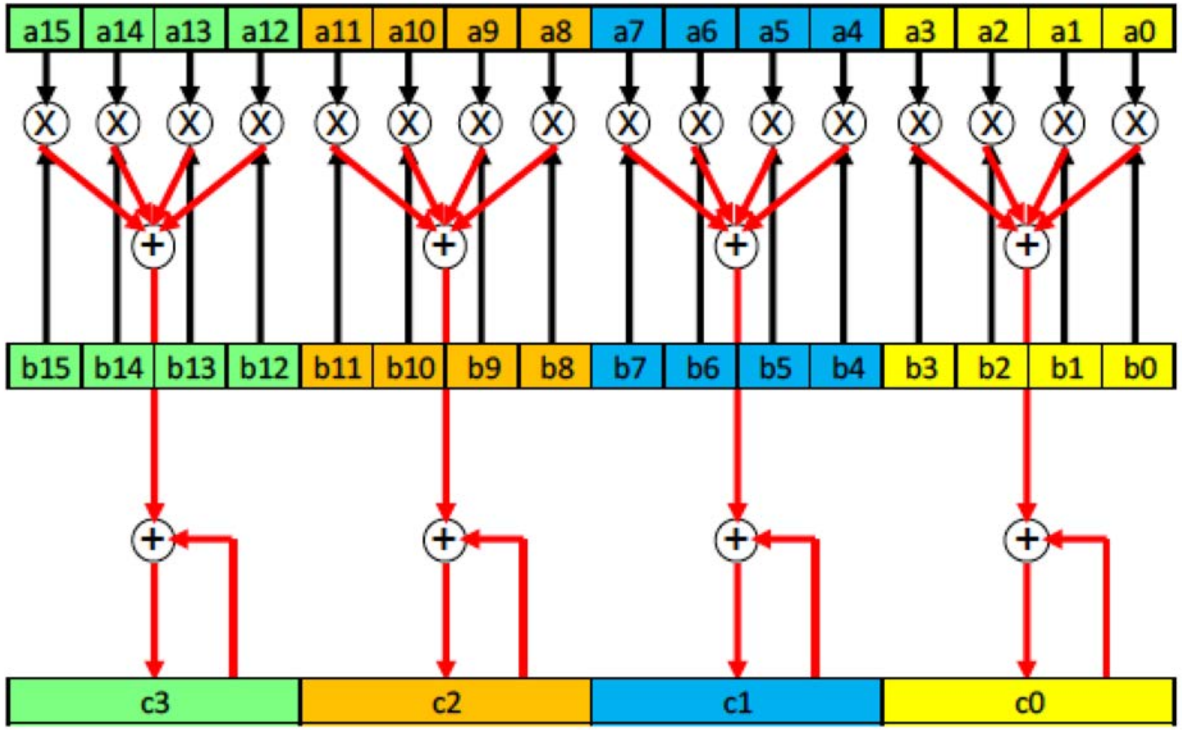
- Add the vector neural network Instruction (VNNI)
- For convolution loops operating on 8-bit integers, the new vector unit fuses three instructions into one
- Companion software MKL-DNN—math kernel library for deep neural networks
- The VPDPBUSD instruction fuses MAC operations for INT8 operands into a 32-bit accumulator to evaluate 4 terms at once
  $c_0 = a_3 \times b_3 + a_2 \times b_2 + a_1 \times b_1 + a_0 \times b_0 + c_0$
  - a_i and b_i bytes from INT32 a and b



- VPDWSSD instruction fuses MAC operations for INT16 into a 32-bit accumulator
  $c_0 = a_1 \times b_1 + a_0 \times b_0 + c_0$
- Triples INT8 over dual AVX Skylake-SP
  - Yields ~ 2 × over

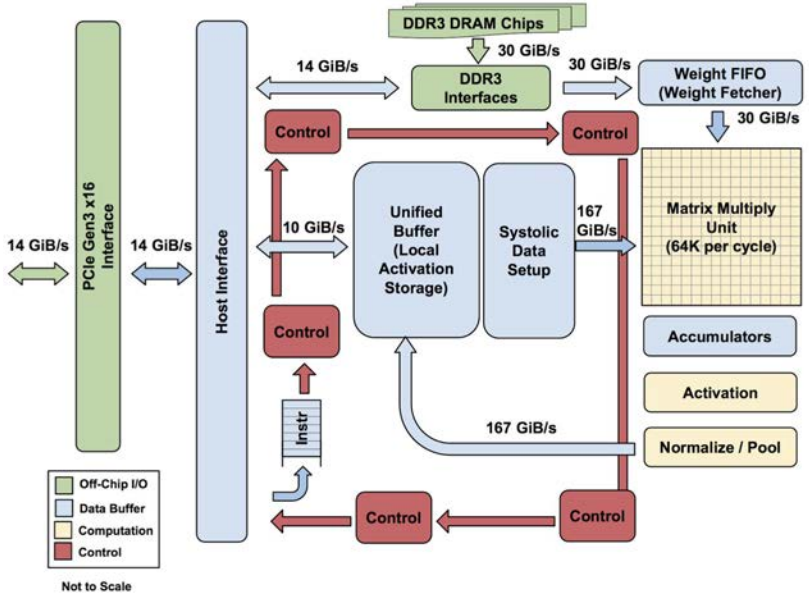
Repurposed CPU—Arm dot Product Instructions

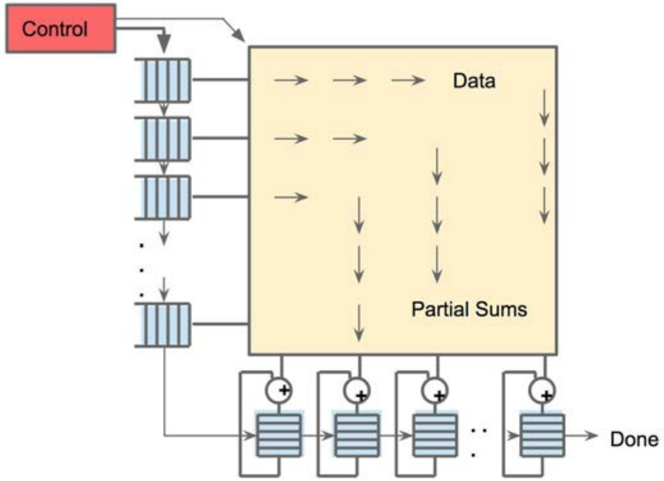
- Arm solution—add dot-product instructions in Neon
- Supported by NN libraries in Arm’s project Trillium software stack
- Similar to Intel’s approach
- 4x performance boost to CNNs on 64-bit Cortex-A CPUs
- 4 dot products at once
- 4 32-bit accumulators each evaluating
  $c_0 = a_3 \times b_3 + a_2 \times b_2 + a_1 \times b_1 + a_0 \times b_0 + c_0$
  a_i and b_i bytes from INT32 a and b
- Cortex-A76 at 2.4GHz gives 614GOP/s or 307GMAC/s



Google’s TPU 1.0*—a 3 year old technology

- Matrix multiply unit—65,536 (256x256)
  - 8-bit multiply-accumulate units
- 700 MHz clock
- Peak: 92T operations/second
  - 65,536 × 2 × 700M
  - >25 × more MACs vs GPU
  - >1000 × more MACs vs CPU
- 24 MB of on-chip Unified Buffer
- 3.5 × as much on-chip memory vs GPU
- Two 2133MHz DDR3 DRAM channels
- 8 GB of off-chip weight DRAM memory
- Control and data pipelined





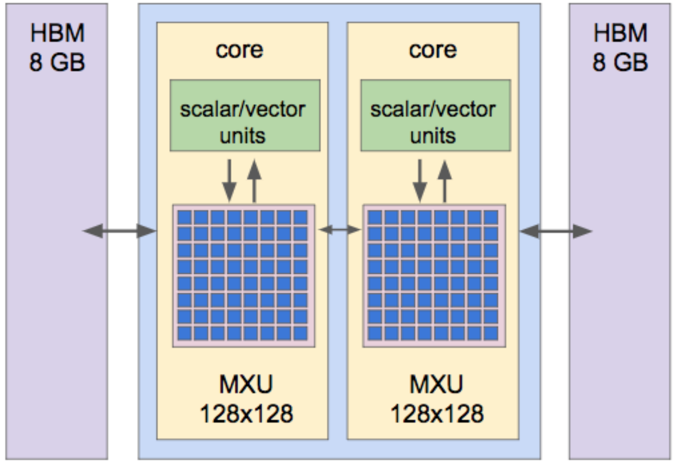
* "In-Datacenter Performance Analysis of a Tensor Processing Unit, Jouppi et al." 44th International Symposium on Computer Architecture (ISCA), Toronto, Canada, June 26, 2017.

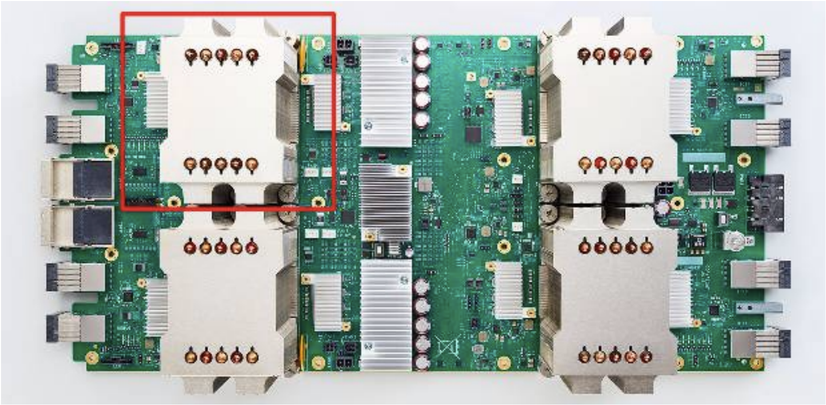
Observations

- TPU 1.0 uses 8 bit integer arithmetic to save power and area
  - A theme for others too—e.g. GraphCore
- BUT TPU 2.0 appears to be floating point
  - Ease of programming is worth something
- Systolic operation is best suited to dense matrices
- Publicly available development environment—Tensor Flow

Change of Direction for TPU 2.0

- Targeting training too
- 32 bit floating point
- 45 TFLOPS
- 16GB HBM, 600 GB/s mem BW
- Power consumption?

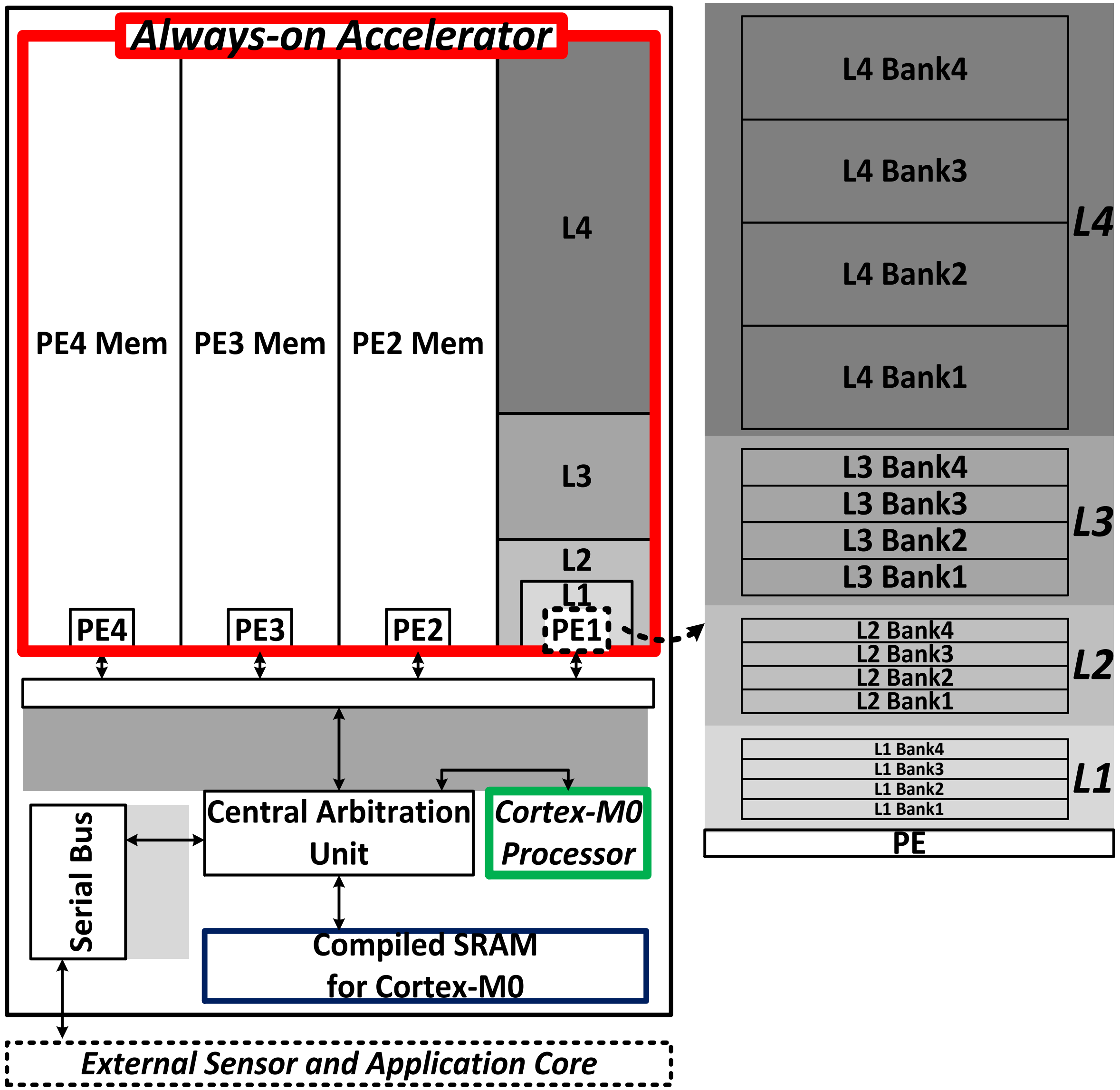




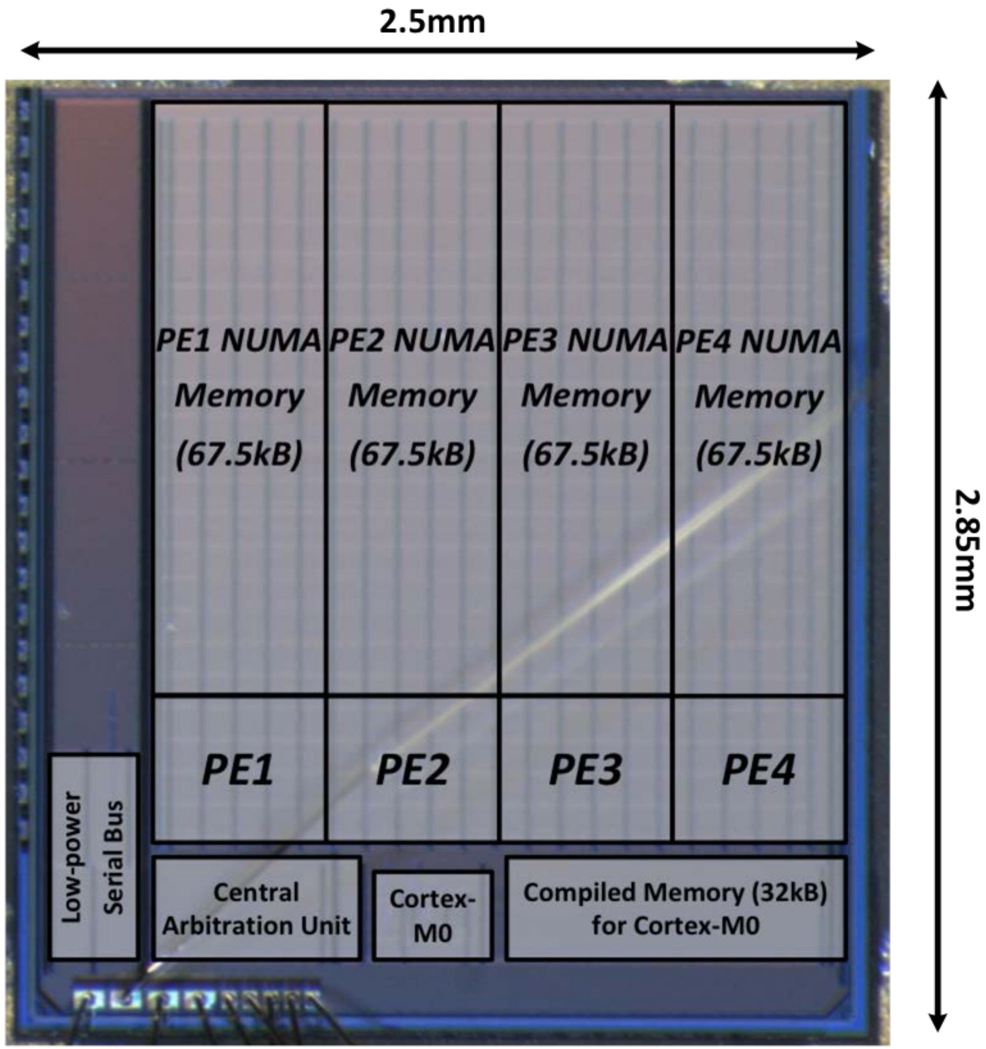
Academia—Low-Power Applications

- **Non-uniform scratchpad architecture**

- Many always-on application executes in a repeatable and deterministic fashion
- Optimal memory access can be pre-determined statically
  - Scratchpad instead of cache
  - Assign more frequently data to smaller, nearby banks



Academia—Chip implementation

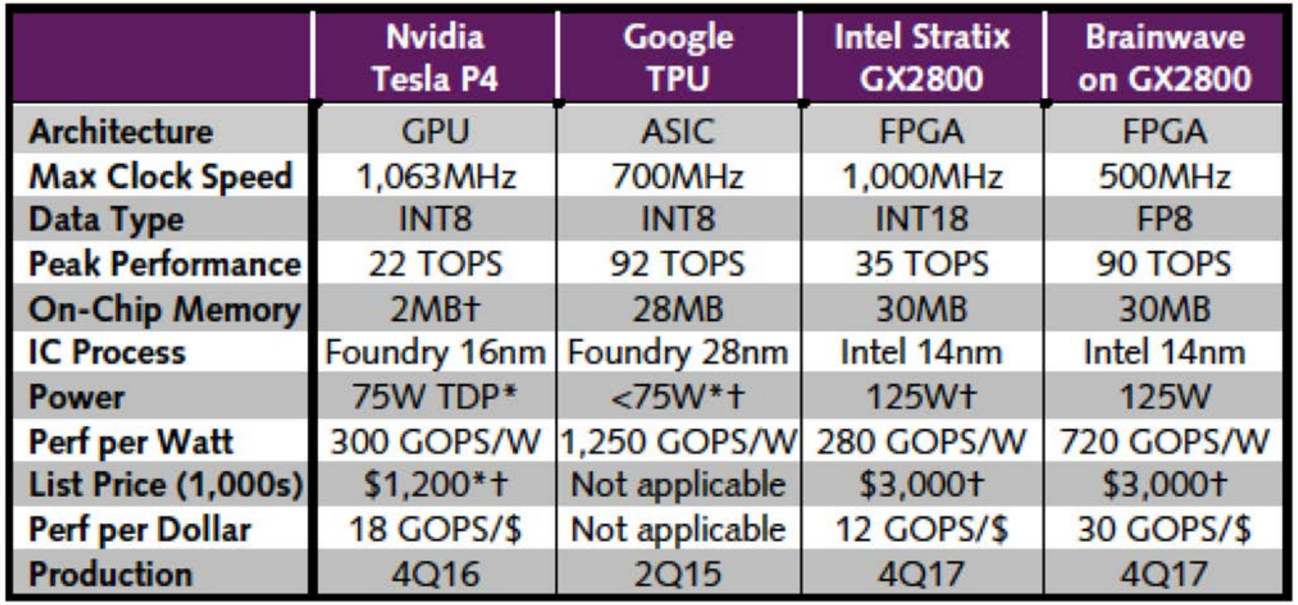


Process	40nm
Chip Area	7.1mm^2
# of PEs	4
Accelerator SRAM Size	270 KB
Available Fixed-Point Precision	6, 8, 12, 16, 24, 32 bits
Operating Power	0.288 mW
Efficiency	374 GOPs / W

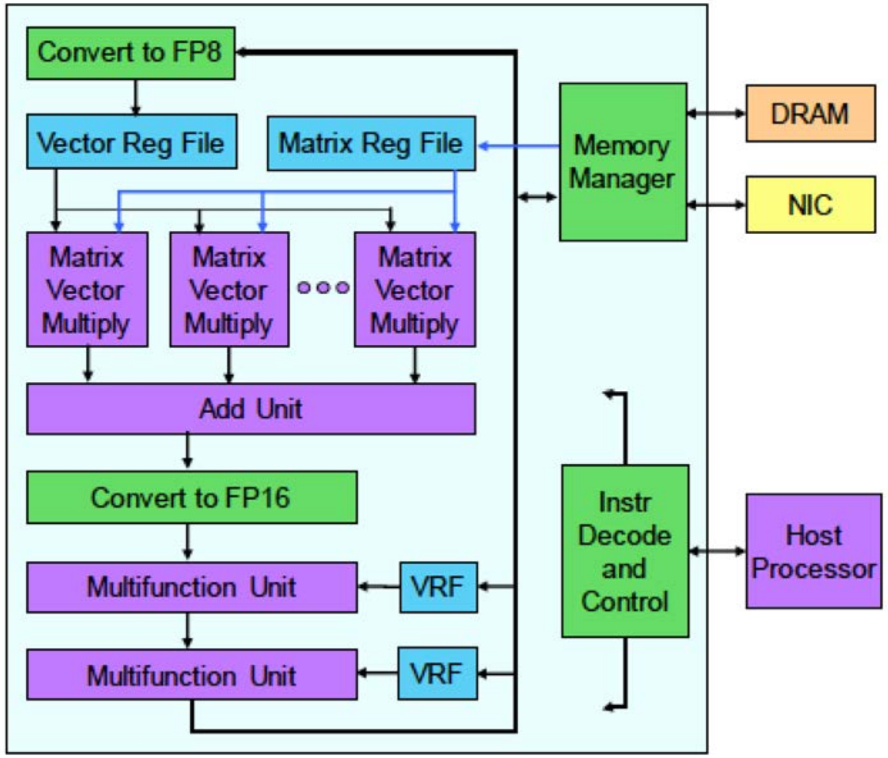
Reference: S. Bang, et.al, ISSCC 2017

FPGA—Microsoft Brainwave

- Microsoft's cloud solution—may morph into an ASIC
- Intended as a coprocessor
- Employs a large number of multiply units to accelerate DNNs.
- Number of units is depends on the FPGA size
- Performance comparisons:



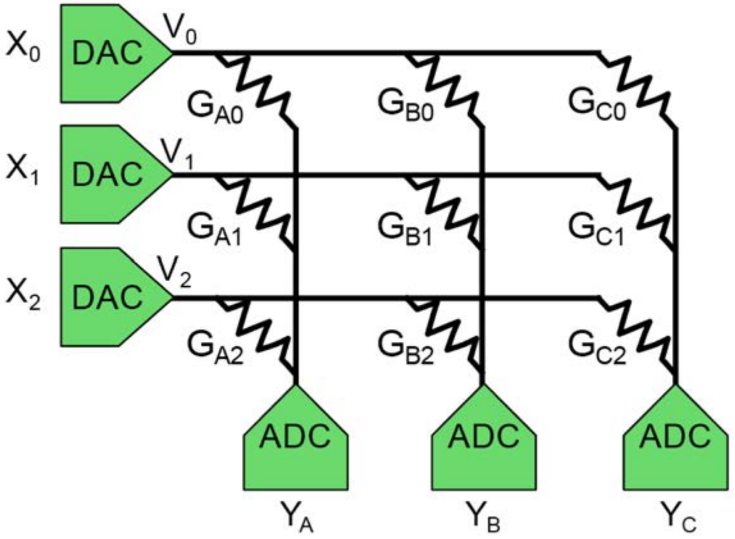
	Nvidia Tesla P4	Google TPU	Intel Stratix GX2800	Brainwave on GX2800
Architecture	GPU	ASIC	FPGA	FPGA
Max Clock Speed	1,063MHz	700MHz	1,000MHz	500MHz
Data Type	INT8	INT8	INT18	FP8
Peak Performance	22 TOPS	92 TOPS	35 TOPS	90 TOPS
On-Chip Memory	2MB†	28MB	30MB	30MB
IC Process	Foundry 16nm	Foundry 28nm	Intel 14nm	Intel 14nm
Power	75W TDP*	<75W*†	125W†	125W
Perf per Watt	300 GOPS/W	1,250 GOPS/W	280 GOPS/W	720 GOPS/W
List Price (1,000s)	$1,200*†	Not applicable	$3,000†	$3,000†
Perf per Dollar	18 GOPS/$	Not applicable	12 GOPS/$	30 GOPS/$
Production	4Q16	2Q15	4Q17	4Q17



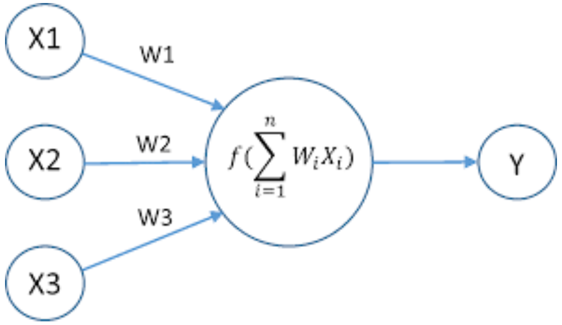
Selected deep-learning accelerators. TOPS=trillions of math op-erations per second. These designs all target inferencing. *Includes DRAM. (Source: vendors, except †The Linley Group estimate)

Analog—Mythic IPU

- Premises:
  - Use Analog In-Memory Computing to eliminate processor/memory energy
- Based on Fujitsu’s 40nm embedded-flash cell
  - Flash cell is used to store 256 different conductances—shown as Gs in diagram
  - Conductances are voltage programmed through 8-bit DACs
  - Voltage programmed conductances in memory cells represent the neural-network weights.
  - Ohm’s law and current summing
- Low power—5W
- To achieve this precision a closed loop calibration phase is required—takes one minute
- Pooling layers (may be non-linear) and activations (the ReLU function) still require digital logic

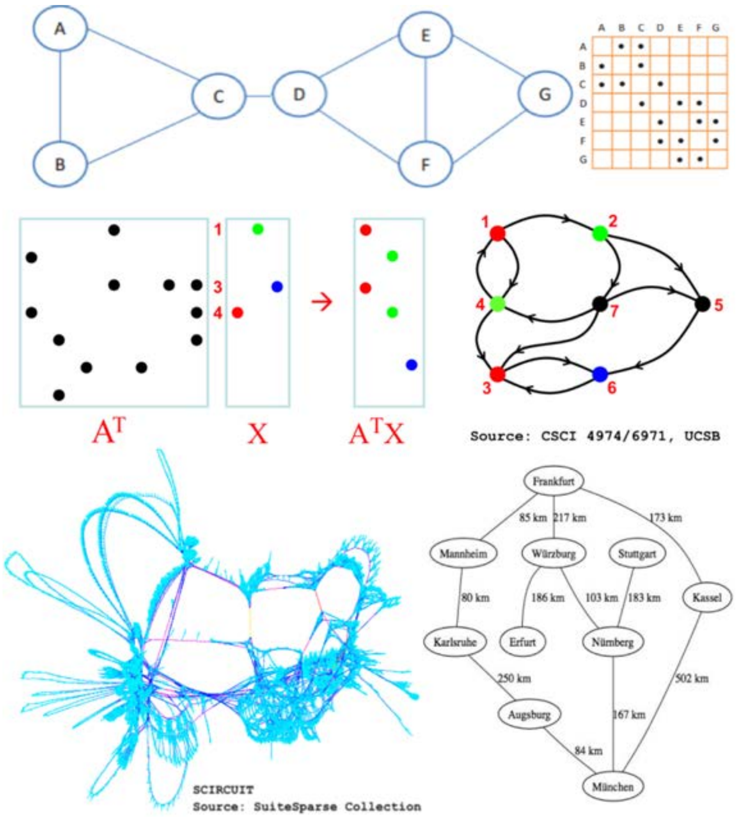


(Source: The Linley Group)



What’s in the Future

- Investment boom is tailing off—”AI fatigue”
- Recognition that many of the future ML problems will require efficient handling of sparse data structures
- Big data collected from various sources
  - Sensor feed, social media, scientific experiments
- Challenge: the nature of data is sparse
- Architecture research previously focused on improving compute
  - Sparse matrix computation: a key example of memory bound workloads
  - GPUs achieve ~100 GFLOPS for dense matrix multiply vs. ~100 MFLOPS for sparse matrices
- Change of focus to data movement & less rigid SIMD compute model

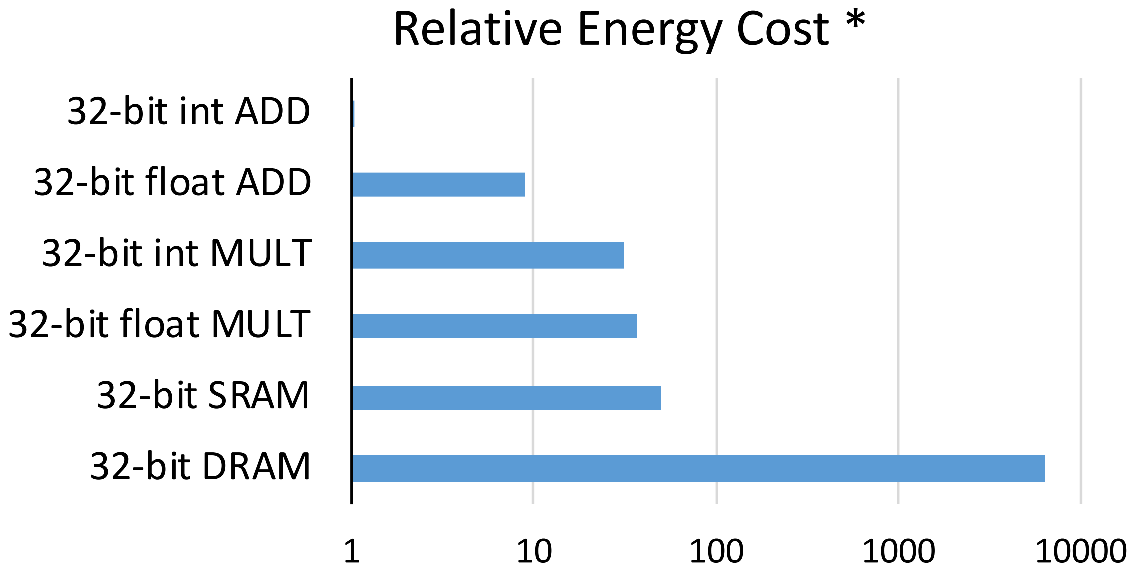


Source: CSCI 4974/6971, UCSB

SCIRCUIT
Source: SuiteSparse Collection

What Next? Reduce the DNN Size*

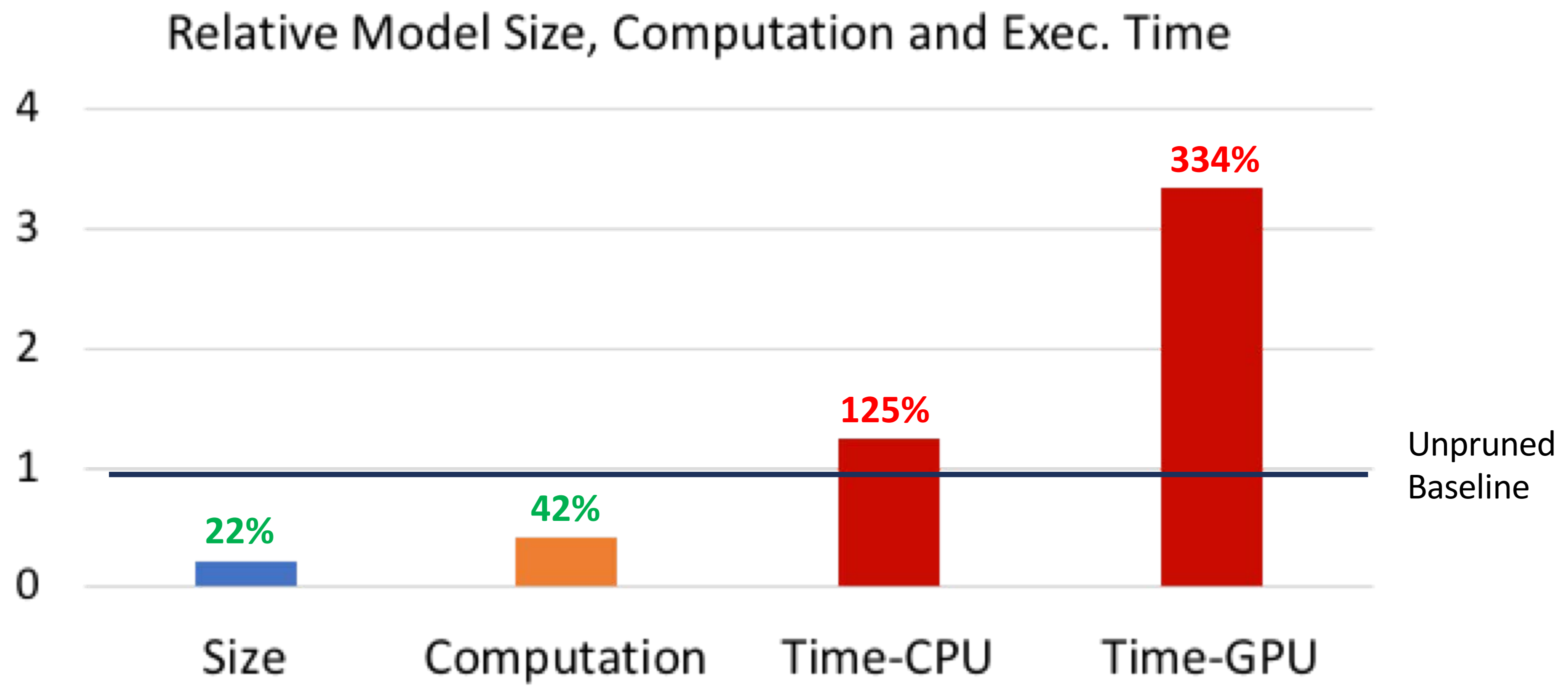
- Recall
  - AlexNet—8 layers, 240MB weights
  - VGG-16—16 layers, 550MB weights
  - Deep Residual Network—152 layers, 229MB weights
- Large model size leads to high energy cost
  - NNs cannot fit in on-chip SRAM
  - DRAM access is energy-consuming
- Precision reduction
  - Low-precision fixed-point representation
  - Need hardware support
- Weights pruning
  - Remove redundant weights
  - Sparse weights matrix
- Weight sharing
- Application Specific Accelerators †



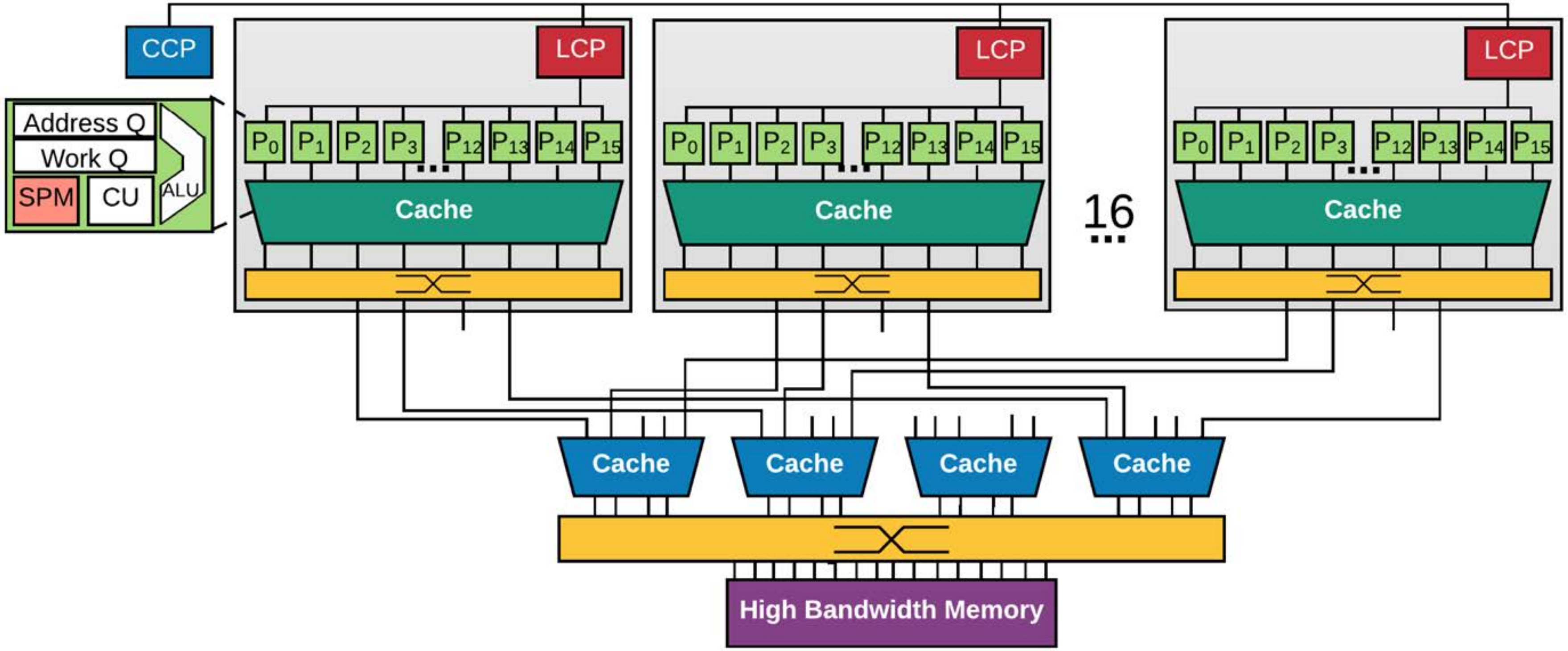
* Han *et al.* “A deep neural network compression pipeline: Pruning, quantization, huffman encoding.” arXiv preprint arXiv:1510.00149 (2015)
†Han, et al. "EIE: Efficient Inference Engine on Compressed Deep Neural Network." arXiv preprint arXiv:1602.01528 (2016).

Drawbacks—Sparsity Difficult to Vectorize

- Execution time increases
  - Computation reduction not fully utilized
  - Extra computation for decoding sparse format
- AlexNet



OuterSPACE Project



- SPMD-style Processing Elements (PEs), high-speed crossbars and non-coherent caches with request coalescing, HBM interface
- Local Control Processor (LCP): streaming instructions in to the PEs
- Central Control Processor (CCP): work scheduling and memory management

Q&A

Thank you

Questions?