SAND2024-05575C

Sandia National Laboratories

Exceptional service in the national interest

Strategic & Large-Scale Considerations of Neuromorphic Computing

PRESENTED BY

CRAIG M. VINEYARD [CMVINEY@SANDIA.GOV]

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Outline

Strategic Considerations of Neuromorphic

- Hardware lottery
- Game theory
- Computing game
- Co-design

Large-Scale Considerations of Neuromorphic

- Role of scale in computing
- Historical neuromorphic examples
- Hala Point

Hardware Lottery

- “History tells us that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists away from some ideas and towards others.”

- “...to describe when a research idea wins because it is compatible with available software and hardware, not because the idea is superior to alternative research directions”

- “Hardware design has prioritized delivering on commercial use cases, while built-in flexibility to accommodate the next generation of ideas remains a secondary consideration.”

contributed articles

DOI:10.1145/3467017

After decades of incentivizing the isolation of hardware, software, and algorithm development, the catalysts for closer collaboration are changing the paradigm.

BY SARA HOOKER

The Hardware Lottery

HISTORY TELLS US that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists away from some ideas and towards others.[24] This adds noise to the marketplace of ideas and often means there is inertia in recognizing promising directions of research. In the field of artificial intelligence (AI) research, this article posits that it is tooling which has played a disproportionately large role in deciding which ideas succeed and which fail.

What follows is part position paper and part historical review. I introduce the term “hardware lottery” to describe when a research idea wins because it is compatible with available software and hardware, not because the idea is superior to alternative research directions. The choices about software and hardware have often played decisive roles in deciding the winners and losers in early computer science history.

These lessons are particularly salient as we move into a new era of closer collaboration between the hardware, software, and machine-learning research communities. After decades of treating hardware, software, and algorithm as separate choices, the catalysts for closer collaboration include changing hardware economics, a “bigger-is-better” race in the size of deep-learning architectures, and the dizzying requirements of deploying machine learning to edge devices.

Closer collaboration is centered on a wave of new-generation, “domain-specific” hardware that optimizes for the commercial use cases of deep neural networks. While domain specialization creates important efficiency gains for mainstream research focused on deep neural networks, it arguably makes it even more costly to veer off the beaten path of research ideas. An increasingly fragmented hardware landscape means that the gains from progress in computing will be increasingly uneven. While deep neural networks have clear commercial use cases, there are early warning signs that the path to the next breakthrough in AI may require an entirely different combination of algorithm, hardware, and software.

This article begins by acknowledging a crucial paradox: machine-learning researchers mostly ignore hardware despite the role it plays in determining which ideas succeed. The siloed evolution of hardware, software, and algorithm has played a critical role in early hardware and software lotteries. This article considers the ramifications of this siloed evolution

» key insights

- The term hardware lottery describes a research idea that wins due to its compatibility with available software and hardware, not its superiority over alternative research directions.
- We may be in the midst of a present-day hardware lottery. Hardware design has prioritized delivering on commercial use cases, while built-in flexibility to accommodate the next generation of ideas remains a secondary consideration.
- Any attempt to avoid future hardware lotteries must be concerned with making it cheaper and less time consuming to explore different hardware/software/algorithm combinations.

IMAGE BY ANDRIJ BORYS ASSOCIATES

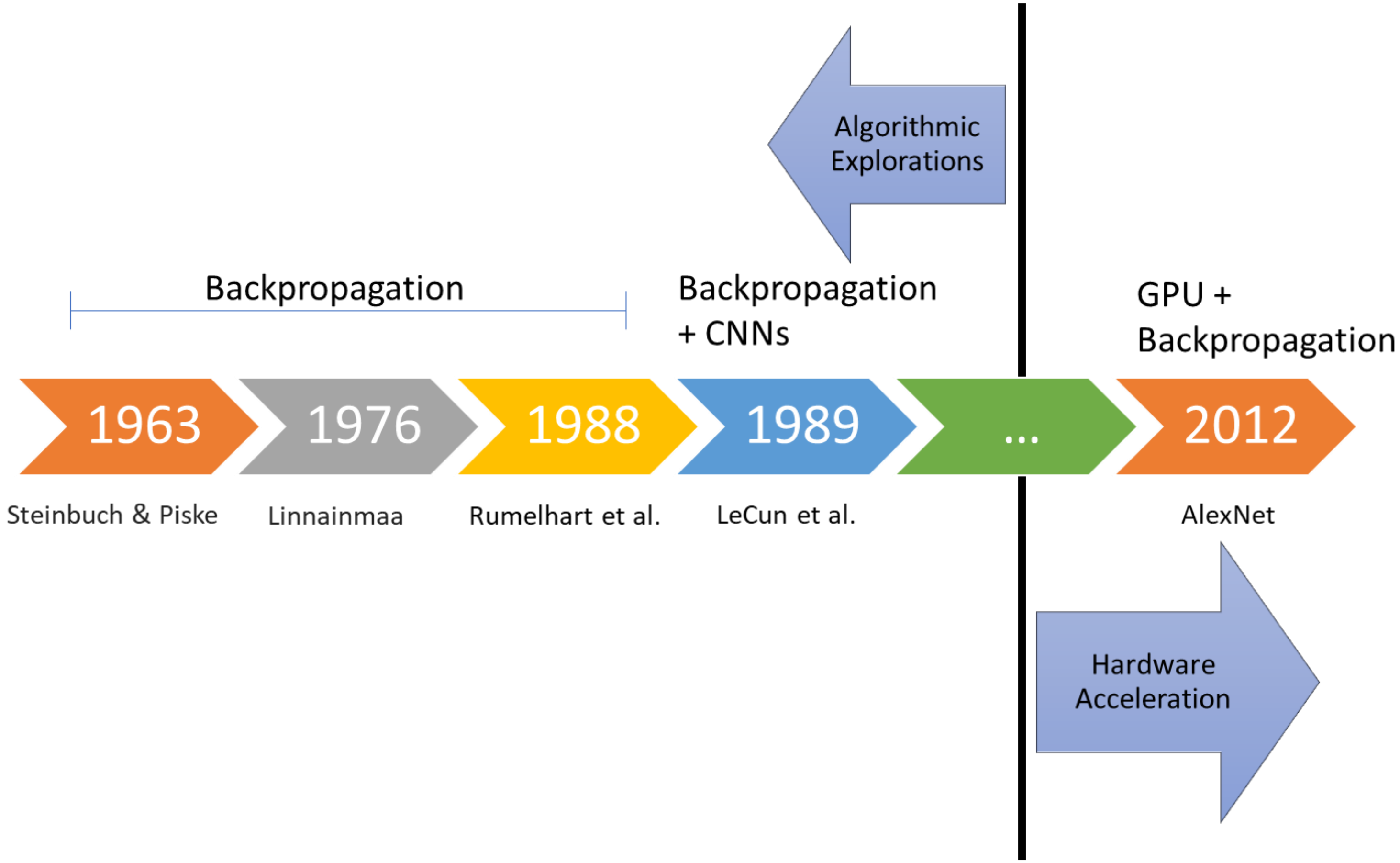
58 COMMUNICATIONS OF THE ACM | DECEMBER 2021 | VOL. 64 | NO. 12

https://dl.acm.org/doi/pdf/10.1145/3467017

Hardware Lottery

Deep Neural Network Example –

- “Backpropagation was invented in 1963, reinvented in 1976 and then again in 1988 and was paired with deep convolutional neural networks in 1989”
- “three decades later - deep neural networks were widely accepted as a promising research direction”
- “The gap between these algorithmic advances and empirical success is due in large part to incompatible hardware”



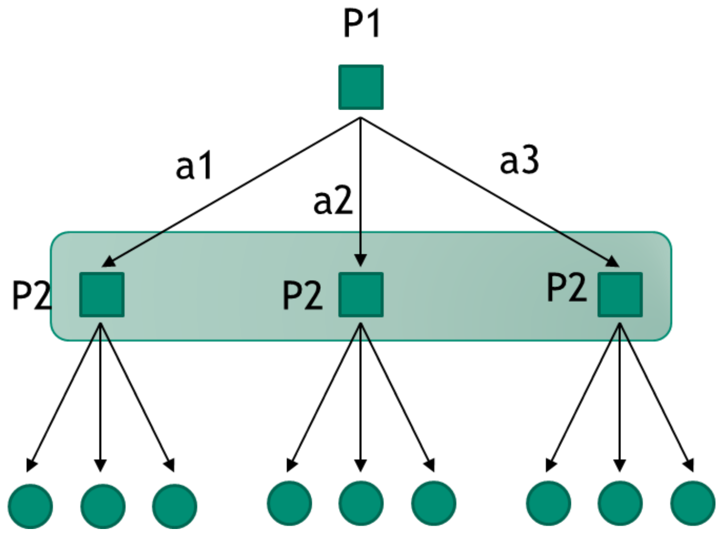
Game Theory

- Game theory – mathematics of strategic actions among agents/players
  - 1921 Emile Borel
  - 1944 von Neumann & Morgenstern

- Example - Prisoner's Dilemma

		Player 2	
		Cooperate	Defect
Player 1	Cooperate	4,4	1,5
	Defect	5,1	2,2

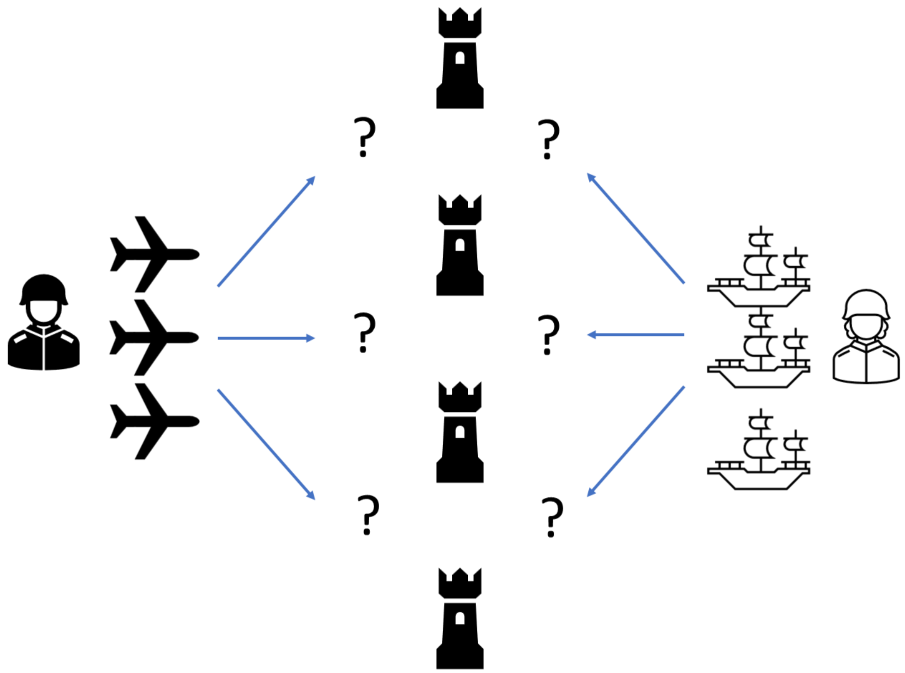
While both players are better off cooperating – Nash equilibria solution is defect-defect

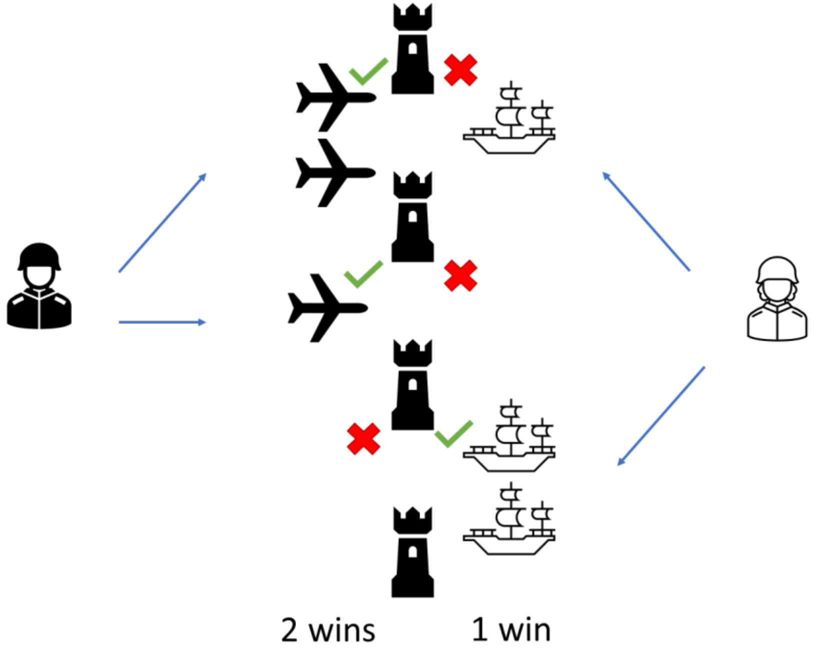


Strategic Perspective 1: Colonel Blotto Game

- Zero-sum resource allocation game
- First introduced by Borel in 1921

- Despite simplicity applicable to model many scenarios & can add nuances





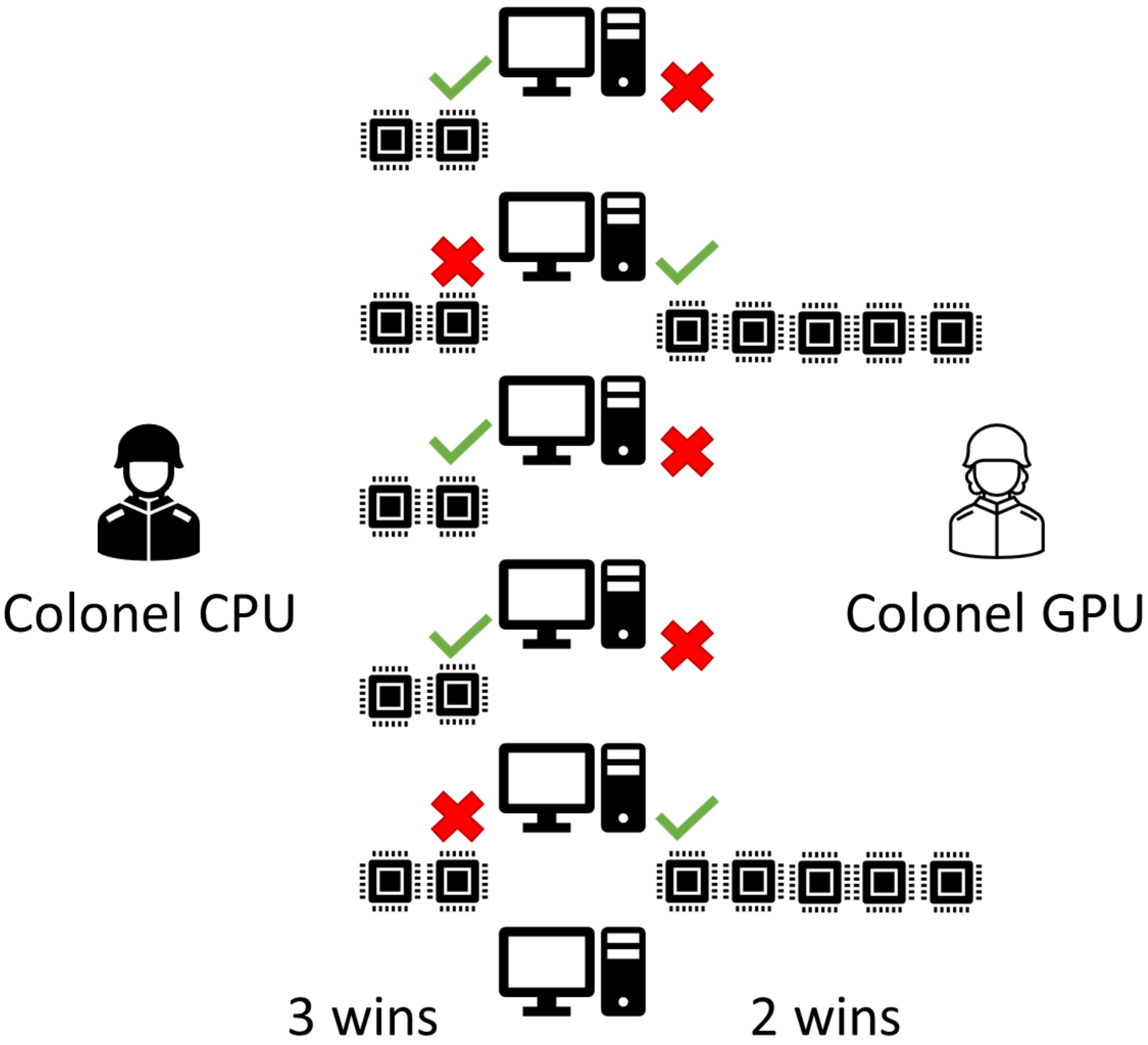
Colonel Blotto – Computing Game

At a high level – consider computing

- Battlefields are computations/workloads

Comparing different architectures

- CPU generality vs GPU specialized performance



Colonel Blotto – Computing Game

At a high level – consider computing

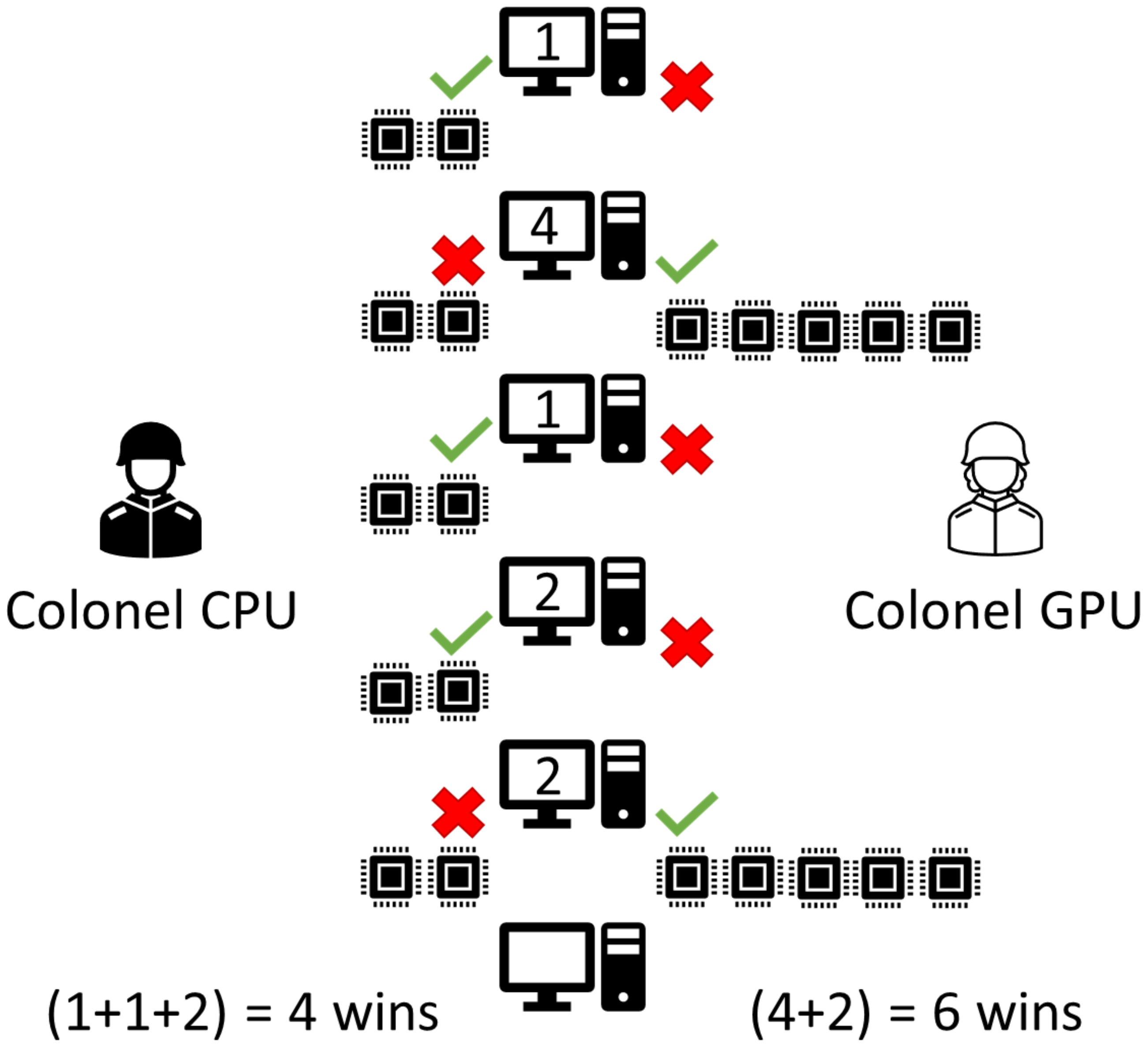
- Battlefields are computations/workloads

Comparing different architectures

- CPU generality vs GPU specialized performance

Weighted Battlefields

- GPU specialization of desirable computations advantageous



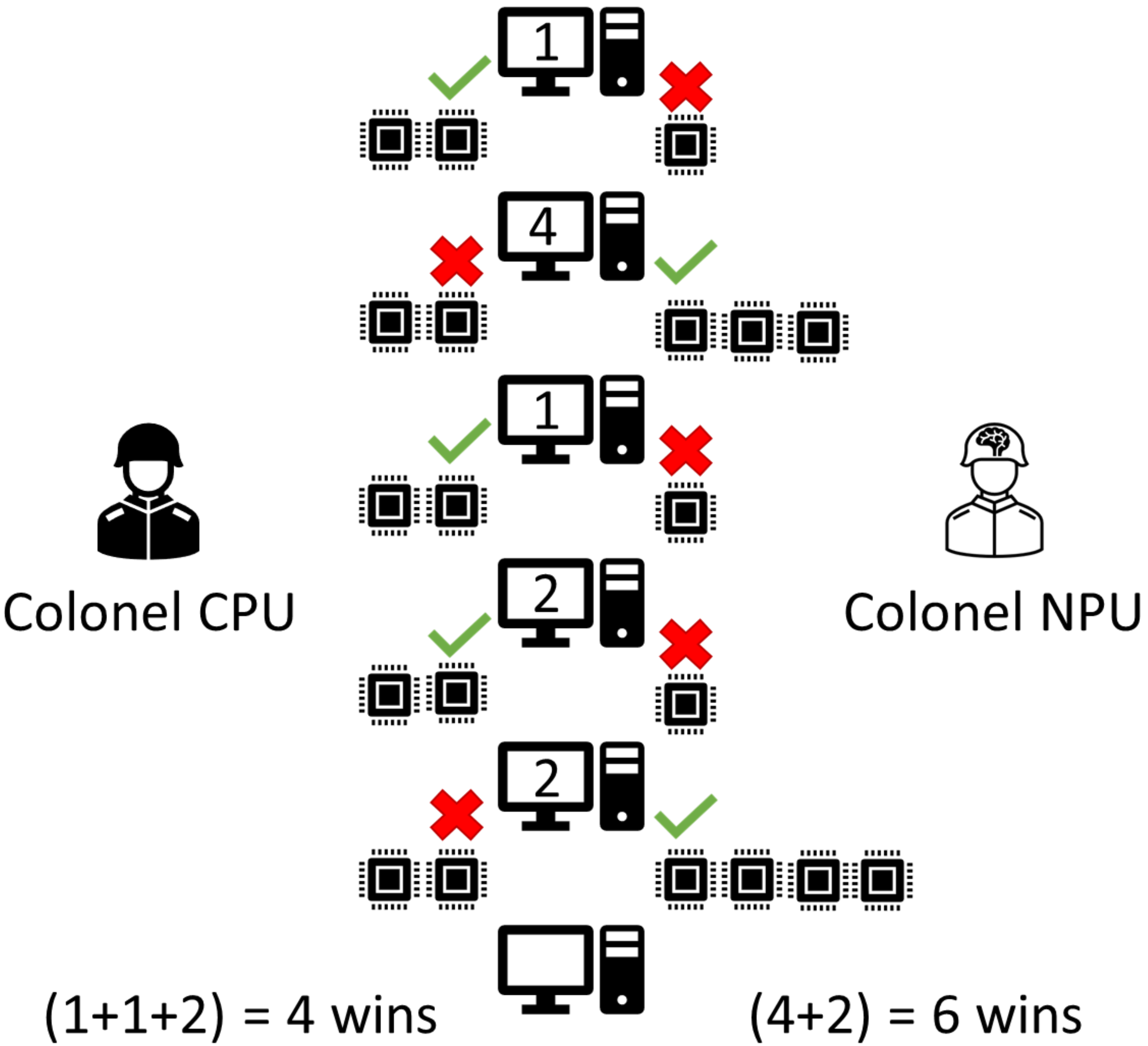
Colonel Blotto – Computing Game

What about Neuromorphic (NPU)?

- We can agree neural isn't better at every computation
  - No Free Lunch Theorem
- Like a GPU, excel as some things

Compared with CPU

- Can show same strategic outcome



Colonel Blotto – Computing Game

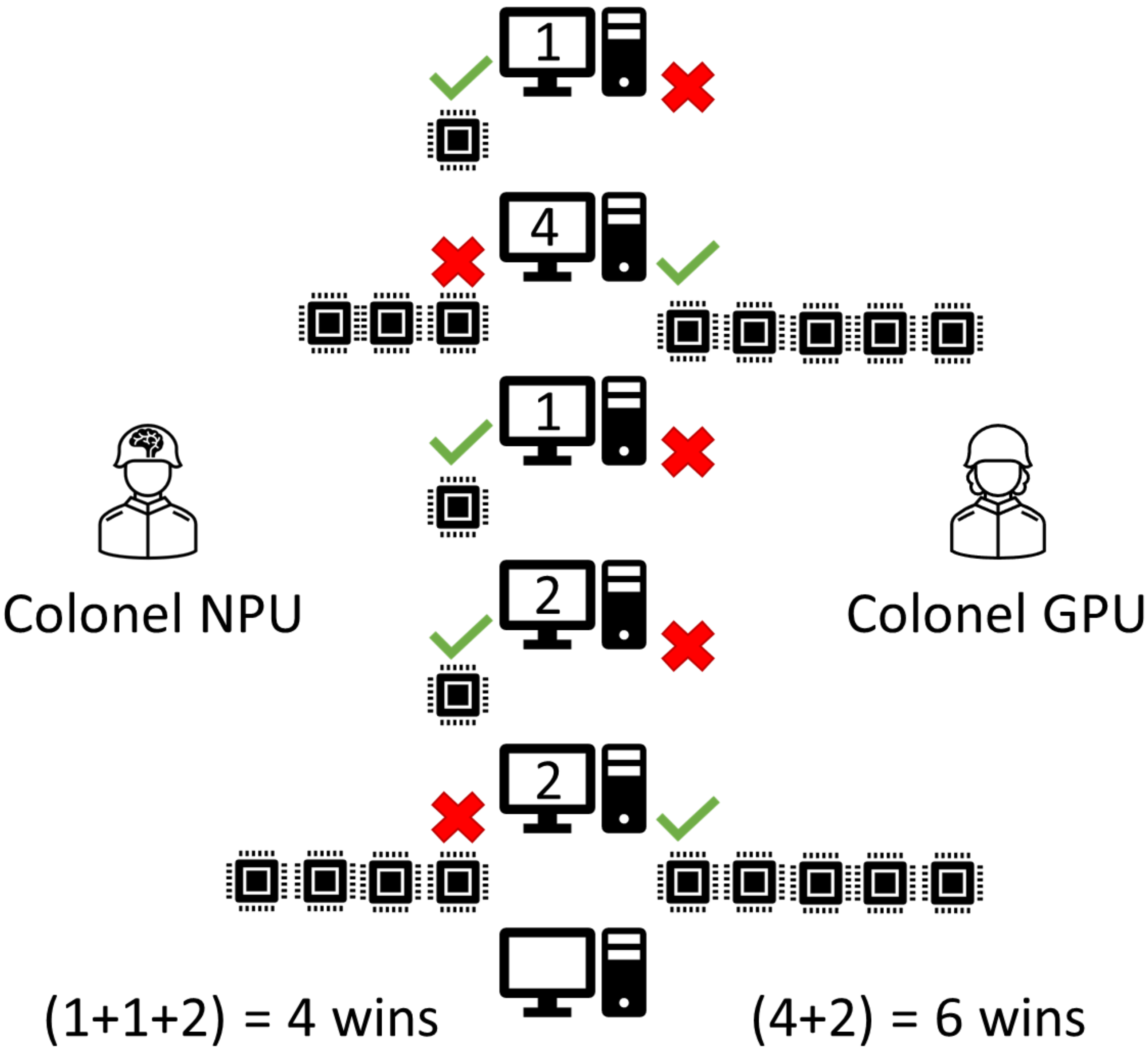
BUT

Compared with GPU

- Similar strategy – but worse is not key to success

Like playing the lottery numbers that won previously & hoping to win again

What GPU+DNN has been great at doesn't mean it's the right strategy for neuromorphic (just because it is a neural network)



Strategic Perspective 2: Neuromorphic Co-Design as a Game

Co-Design – mutual benefit of coordinating design choices of layers in technology stack

- Decisions are interdependent (otherwise optimization)

Stag Hunt

- Two hunters can team up to bring in the higher-value stag, or independently can secure a less rewarding hare

		Player 2	
		Stag	Hare
Player 1	Stag	a,a	c,b
	Hare	b,c	d,d

$$a > b \geq d > c$$

Vineyard, Craig, William Severa, and James Aimone. "Neuromorphic Co-Design as a Game." *Machine Learning with New Compute Paradigms*. 2023.

Strategic Perspective 2: Neuromorphic Co-Design as a Game

- Focus upon the co-design of algorithms and architectures

- Do we need known spiking neural algorithms whose theoretical promise can justify architectural instantiation?

- Can novel architectures precede algorithmic theory and spur innovation?

- Can either algorithms or architectures be pursued without being a lottery for the other?

Co-Design Mixed Strategy Dilemma

Do HW & SW designers pursue SNN architectures and algorithms together or individually pursue ANNs?

		Algorithm Player	
		SNN	ANN
Architecture Player	SNN	5,5	1,3
	ANN	3,1	2,2

- Mixed strategy game solution: Both player’s action distributions are 0.333 SNN and 0.667 ANN

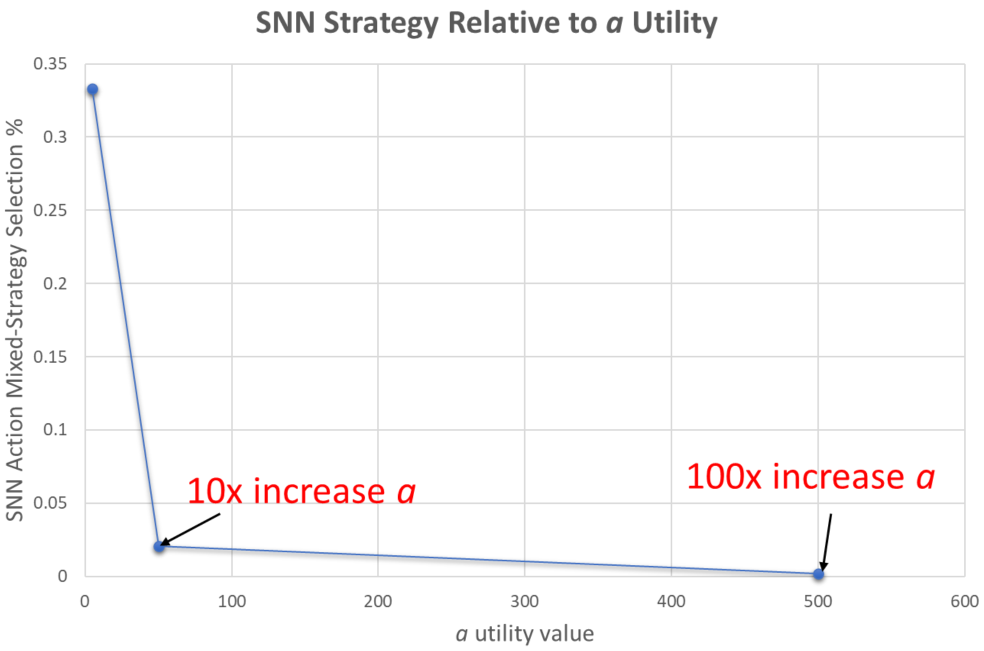
- While promise of SNN may be large, strategy favors pursuing the less risky independent action

Increasing SNN Value

Can a SNN breakthrough (algorithm or architecture) drive the field forwards?

Algorithm Player

Architecture Player		SNN	ANN
	SNN	↑,↑	1,3
	ANN	3,1	2,2

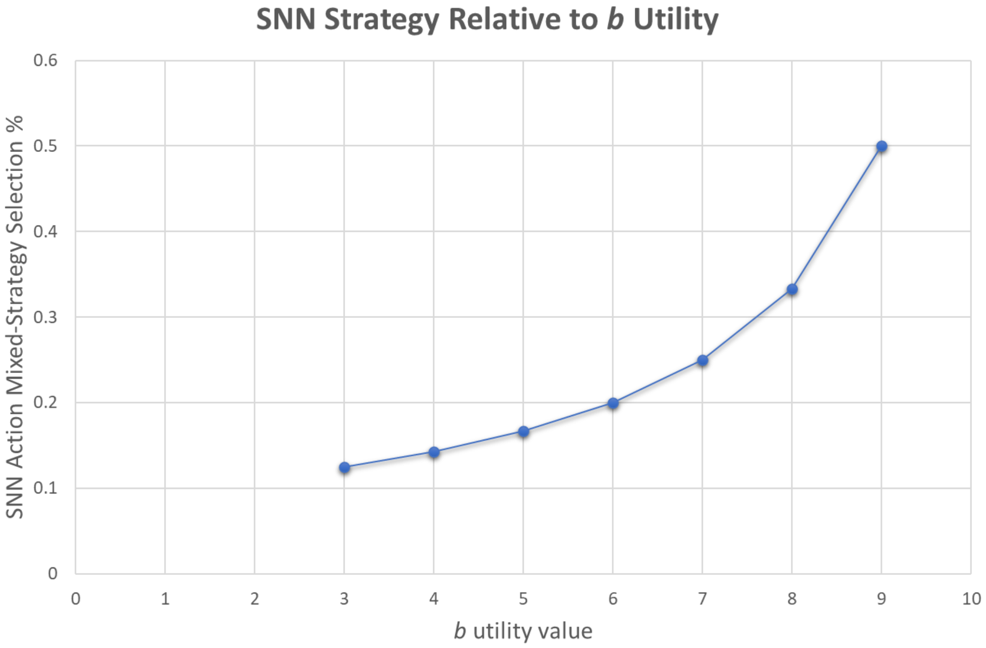


Coordination risk actually drives SNN innovation pursuit down

Compromise – SNN & ANN

Rather than solely seeking to advance SNN research support known ANNs along the way

		Algorithm Player	
		SNN	ANN
Architecture Player	SNN	10,10	1,<3:9>
	ANN	<3:9>,1	2,2



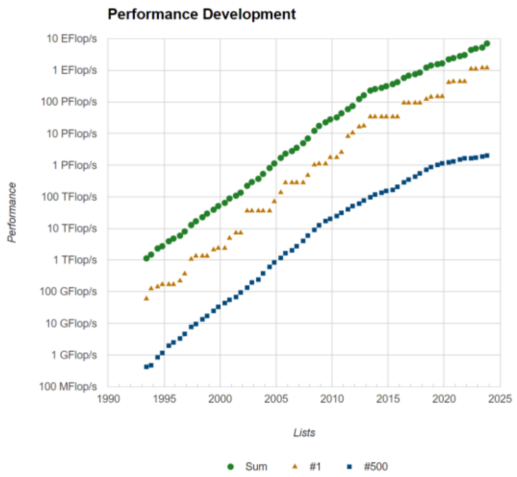
Risk dominance dynamics prevent pursuing SNN from exceeding 0.5

Neuromorphic Scaling

Scale & Computing

Top 500

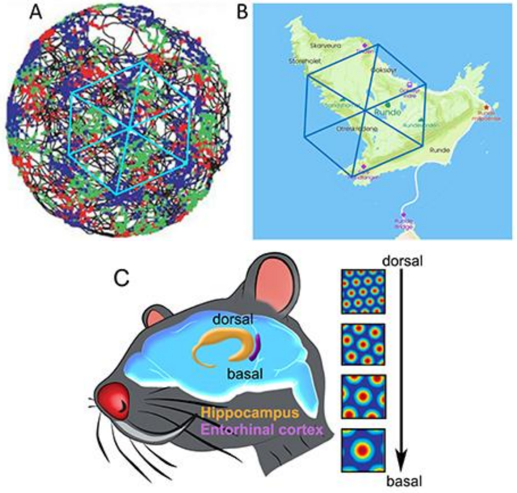
- Established 1993
- Performance measured by LINPACK Benchmark
- Solve a dense system of linear equations



Source: https://www.top500.org/statistics/perfdevel/

Neuroscience - Grid Cells

- 1971 hippocampus has "place cells"
- Expanded recording environment to determine structure of multiple firing fields
- Earlier EC recordings missed because pattern too large for conventional recording boxes



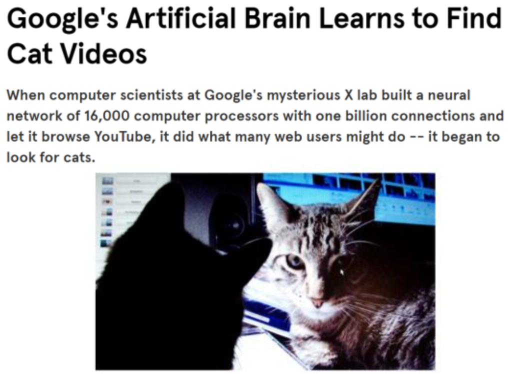
Source: https://kids.frontiersin.org/articles/10.3389/frym.2021.678725

F117 Stealth Fighter

- 1964 theory
- 1975 DARPA program
- 2D computer modeling

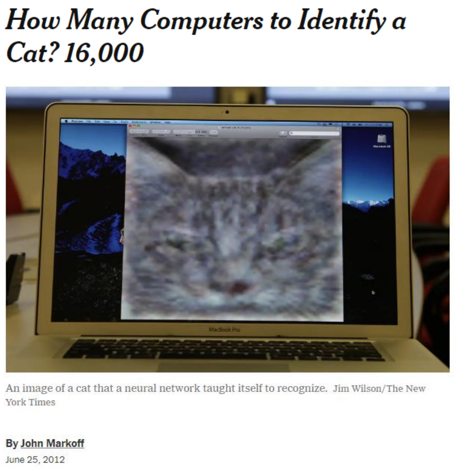
Source: https://nationalinterest.org/blog/buzz/americas-first-stealth-fighter-story-f-117-nighthawk-26231?page=0%2C1

Deep Neural Networks



Google's Artificial Brain Learns to Find Cat Videos

When computer scientists at Google's mysterious X lab built a neural network of 16,000 computer processors with one billion connections and let it browse YouTube, it did what many web users might do -- it began to look for cats.



How Many Computers to Identify a Cat? 16,000

An image of a cat that a neural network taught itself to recognize. Jim Wilson/The New York Times

By John Markoff
June 25, 2012

Sources: https://www.wired.com/2012/06/google-x-neural-network/
https://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html

“The Neural Computing phenomenon is truly amazing:

- Over the past three years, there has been a veritable explosion of interest in neural networks and neurocomputers, even though its foundations have been around since the 1940s;
- And an unusual character of neural computing is its interdisciplinary nature, spanning neurosciences, cognitive sciences, psychologists, computer science, electronics, physics and mathematics. This spectrum of disciplines engaged in neural computing research means that much current literature is scattered over many sources.”

“The surge of interest in neural network applications and models over the past three years has led to the development of special neurocomputers and neural programming environments designed to support the execution and programming of artificial neural networks.“

“The Neural Computing phenomenon is truly amazing:

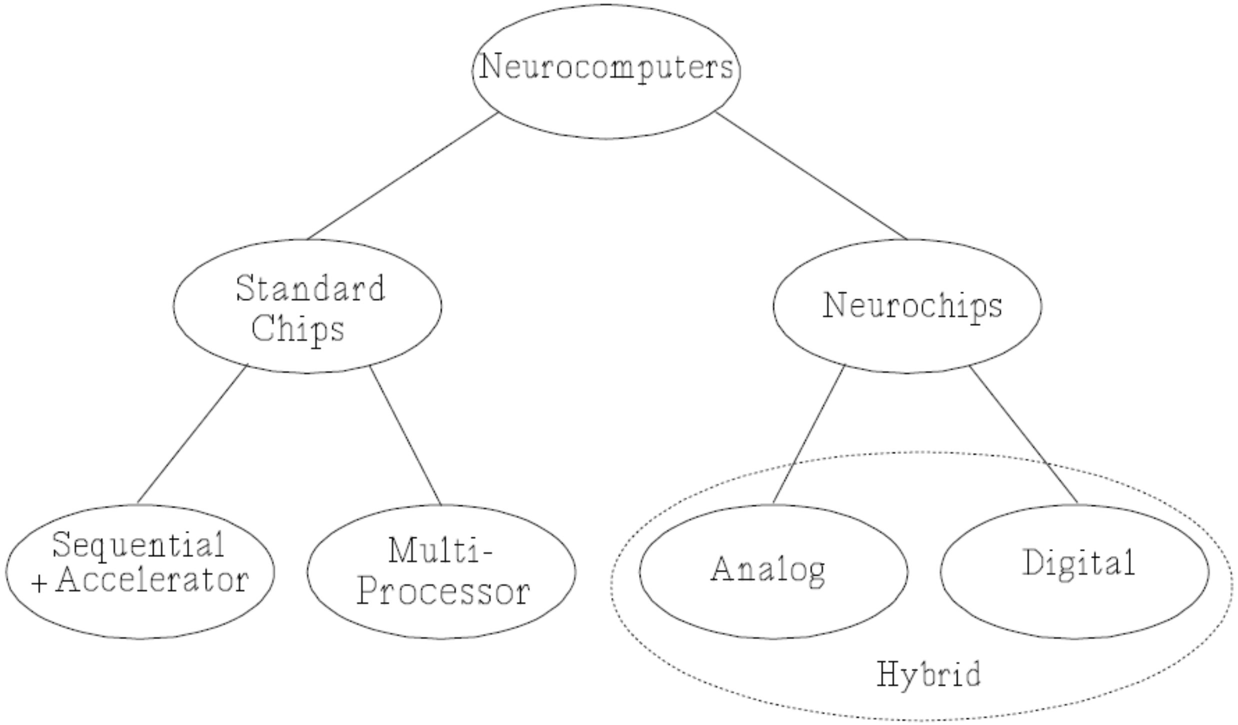
- Over the past three years, there has been a veritable explosion of interest in neural networks and neurocomputers, even though its foundations have been around since the 1940s;
- And an unusual character of neural computing is its interdisciplinary nature, spanning neurosciences, cognitive sciences, psychologists, computer science, electronics, physics and mathematics. This spectrum of disciplines engaged in neural computing research means that much current literature is scattered over many sources.”

“The surge of interest in neural network applications and models over the past three years has led to the development of special neurocomputers and neural programming environments designed to support the execution and programming of artificial neural networks.“

➢ **Treleaven, P. C**. "Hardware and software tools for neural networks." (**1990**)

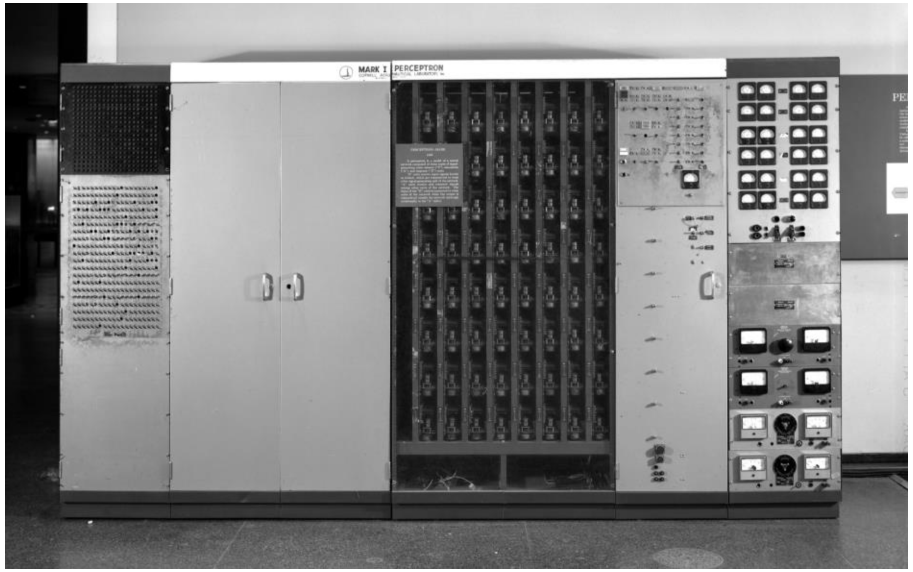
"Since the very beginning of the neural network era, there has been the belief that to fully exploit the potential of this technology it would be necessary to also develop efficient hardware implementation techniques. In the last two decades, we have witnessed how the neural networks community is managing to acquire a more profound understanding of what neural networks can do and how."

Linares Barranco, Bernabé, et al. "Guest editorial-Special issue on neural networks hardware implementations." *IEEE Transactions on Neural Networks, 14 (5), 976-977.* (2003).



Heemskerk, Jan NH. "Overview of neural hardware." *Neurocomputers for brain-style processing. Design, implementation and application* (1995).

Mark I Perceptron

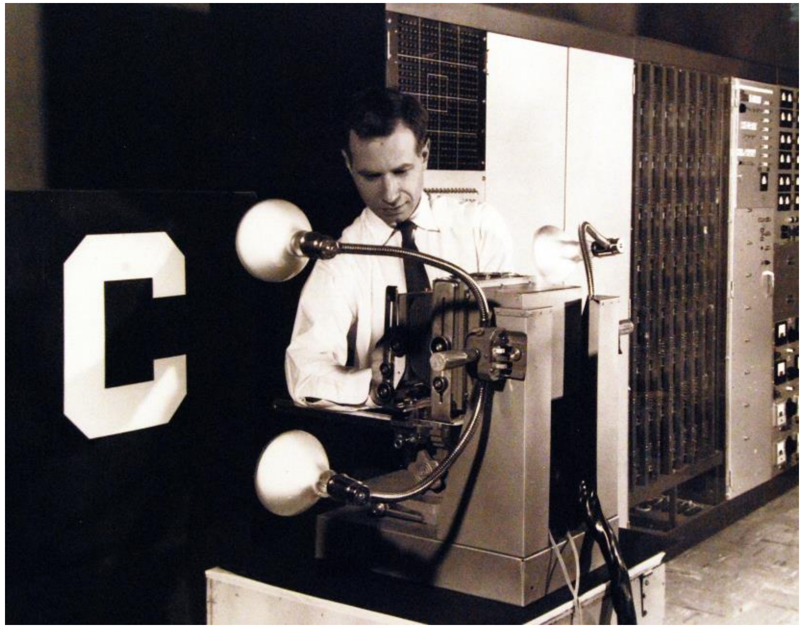


Source: https://americanhistory.si.edu/collections/nmah_334414

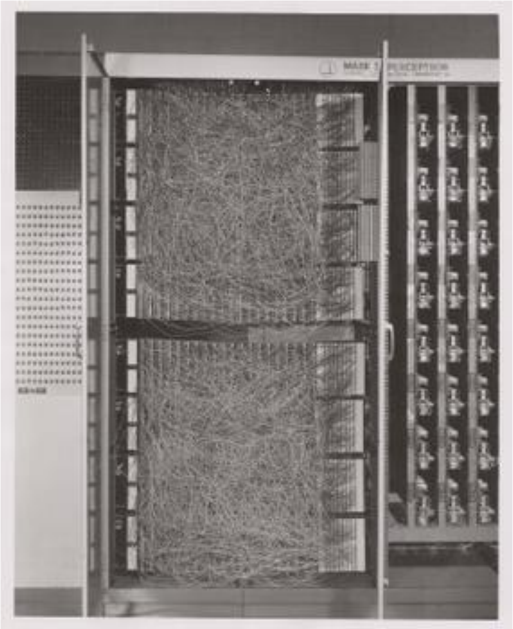
Date: 1958

Fun facts:

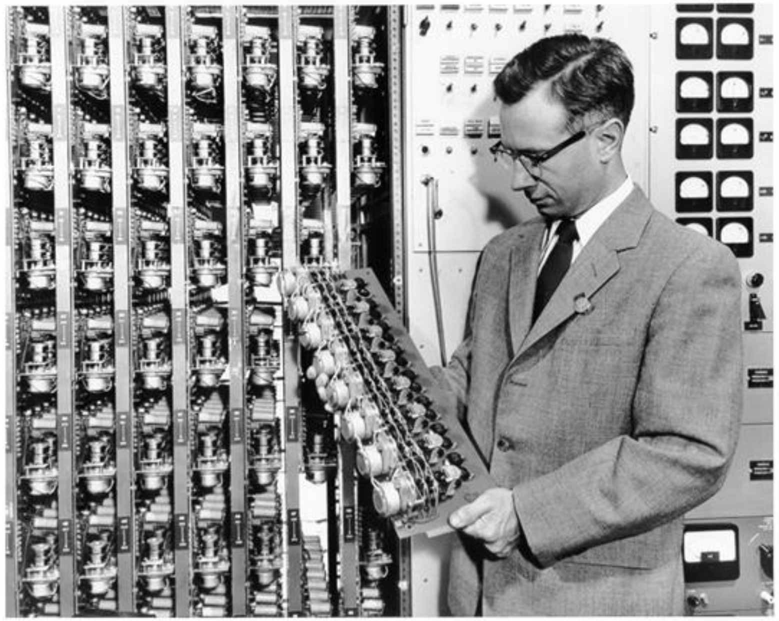
- 20×20 cadmium sulfide photocells to make a 400-pixel input image
- 8 output neurons
- Motor/Potentiometer pairs set weights



Source: https://upload.wikimedia.org/wikipedia/commons/1/1a/330-PSA-80-60_%28USN_710739%29_%2820897323365%29.jpg



Source: https://upload.wikimedia.org/wikipedia/en/5/52/Mark_I_perceptron.jpeg



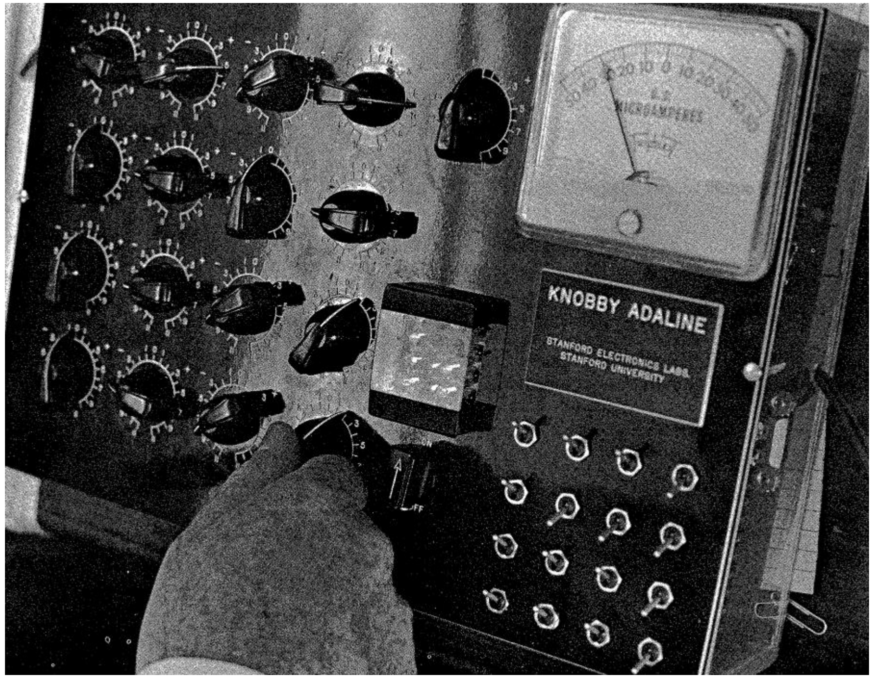
Source: https://fiascodata.blogspot.com/2018/05/a-computer-program-is-said-tolearn-from.html

ADALINE/MADALINE

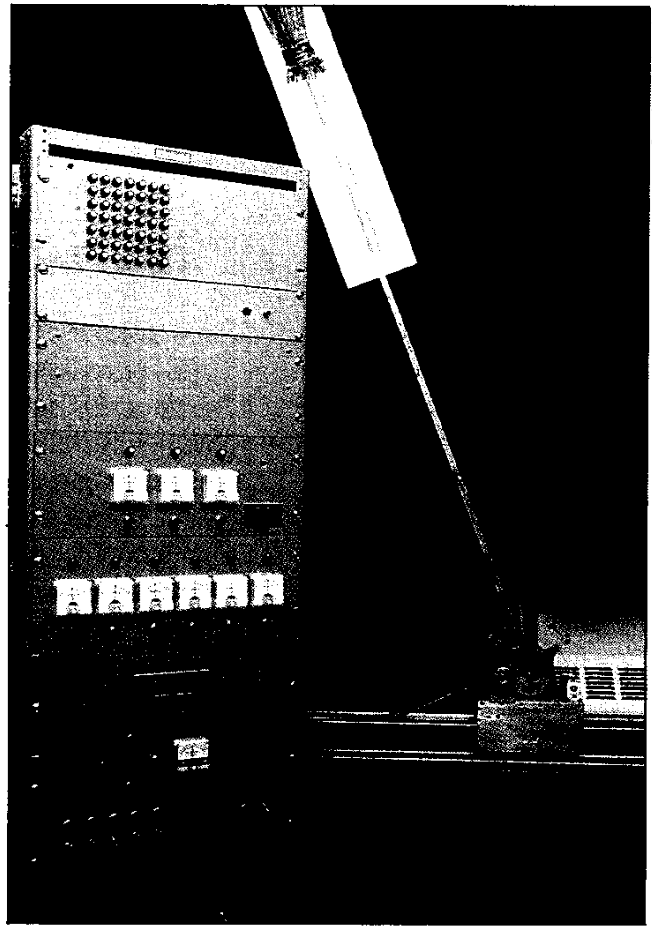
Date: 1960/62

Fun facts:

- Electronically adjustable resistor
- Memistor Corporation offered commercially 1962-1965



Source: https://en.wikipedia.org/wiki/ADALINE#/media/File:Knobby_ADALINE.jpg



Bernard Widrow, Stanford University

Broomstick balancing has become a classic test of a neural network's performance in adaptive control. The original experiment, conducted in 1962 by Bernard Widrow at Stanford University (now professor of electrical engineering), used his Madaline (Multiple ADAptive LINear Elements) neurocomputer (left), *with sensors on the broomstick and cart indicating position, angle, velocity, and acceleration.*

Source: Hecht-Nielsen, Robert. "Neurocomputing: picking the human brain." IEEE spectrum 25.3 (1988): 36-41.

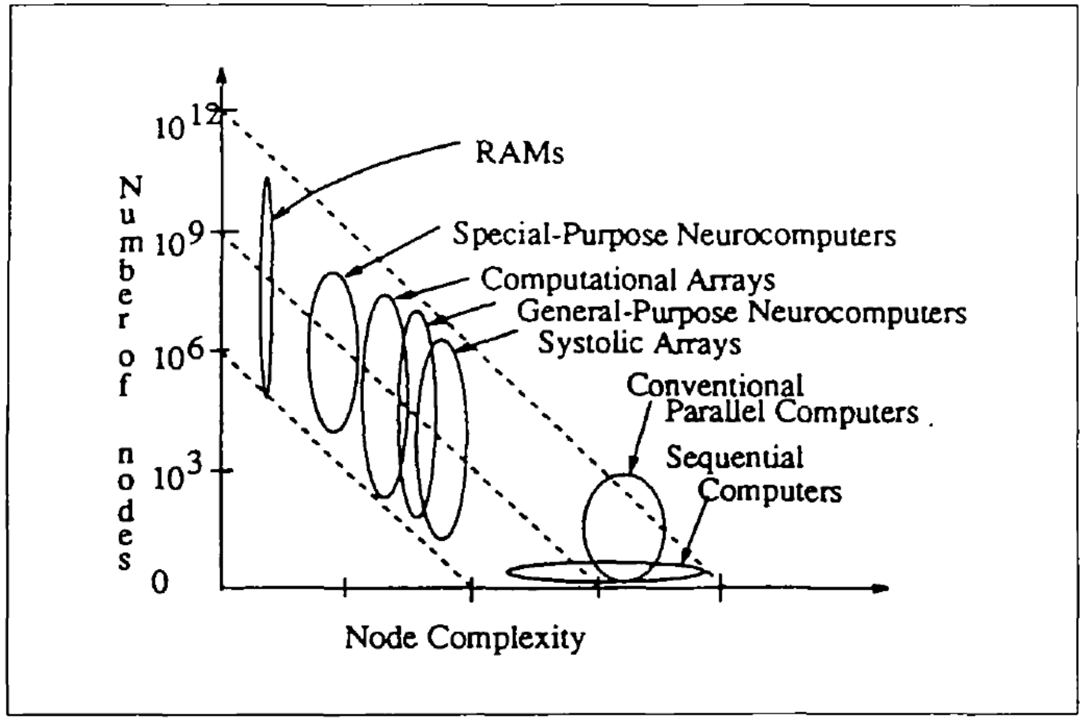
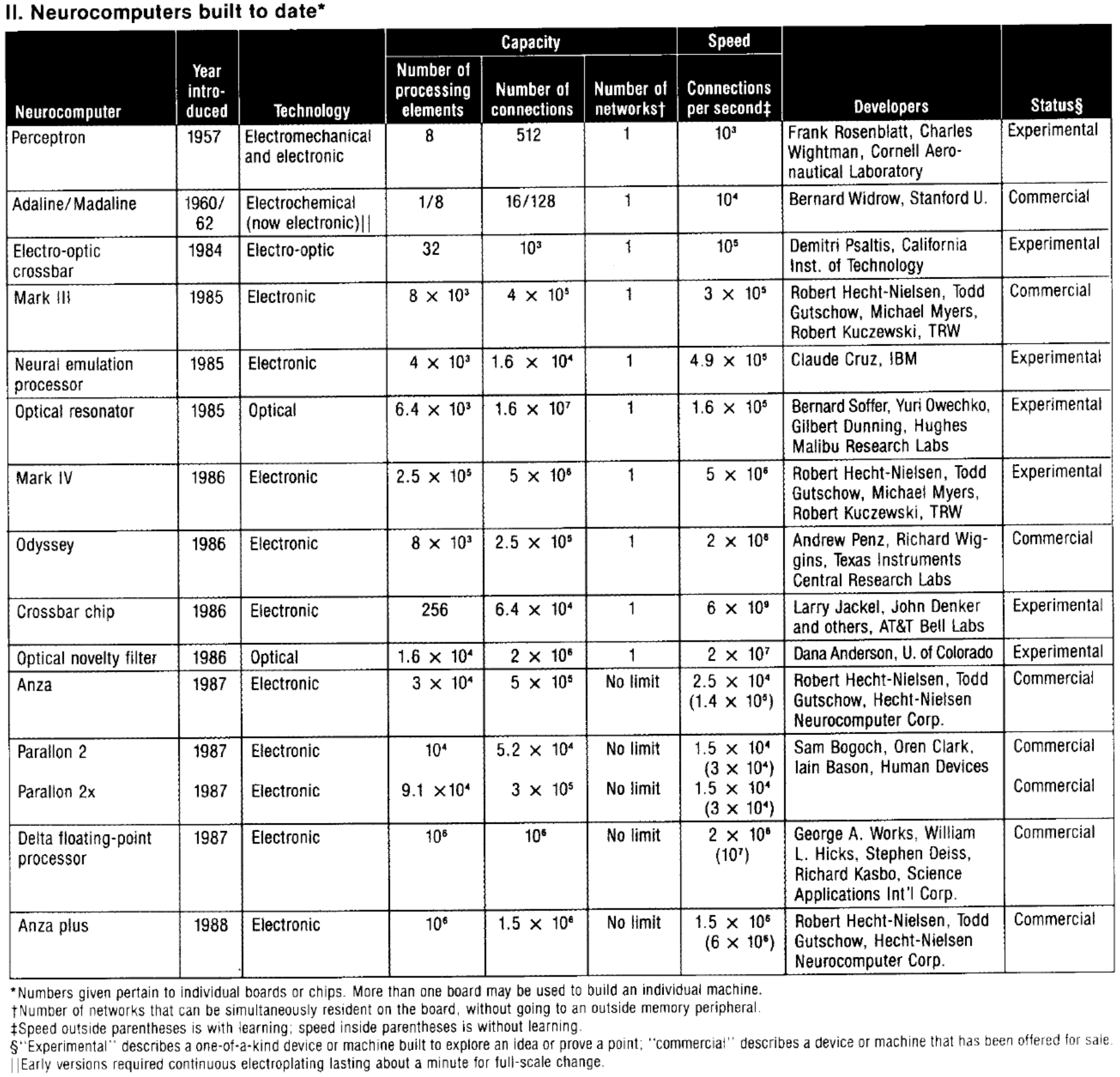


Figure 10: Spectrum of Neurocomputer Architecture (based on Seitz[32])



II. Neurocomputers built to date*

Neurocomputer	Year introduced	Technology	Capacity			Speed	Developers	Status§
			Number of processing elements	Number of connections	Number of networks†	Connections per second‡		
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	2×10^6	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

*Numbers given pertain to individual boards or chips. More than one board may be used to build an individual machine.
†Number of networks that can be simultaneously resident on the board, without going to an outside memory peripheral.
‡Speed outside parentheses is with learning; speed inside parentheses is without learning.
§"Experimental" describes a one-of-a-kind device or machine built to explore an idea or prove a point; "commercial" describes a device or machine that has been offered for sale.
‖Early versions required continuous electroplating lasting about a minute for full-scale change.

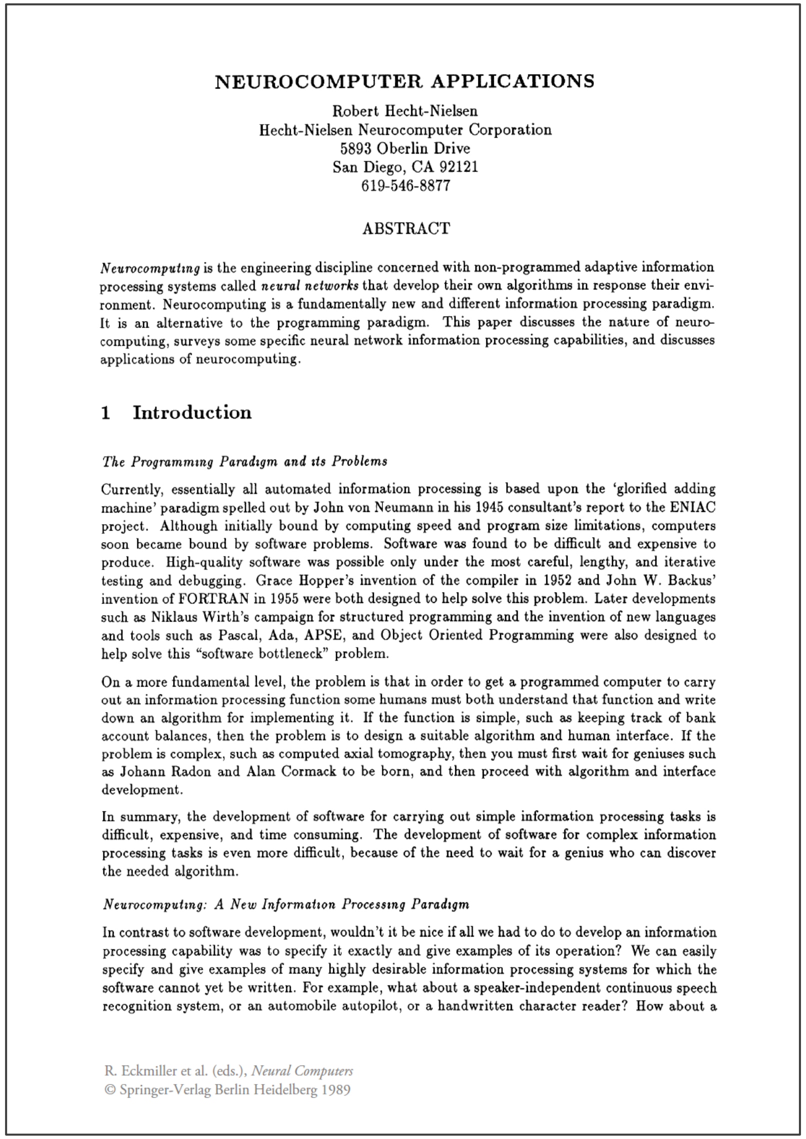
Source: Treleaven, Philip C. "Neurocomputers." Neurocomputing 1.1 (1989): 4-31.

Source: Hecht-Nielsen, Robert. "Neurocomputing: picking the human brain." IEEE spectrum 25.3 (1988): 36-41.

Neuromorphic Applications?

“Although it is still too early to predict which, if any, of these projects will succeed, the fact that they are underway is itself significant. Some examples of real-world applications currently being explored by various industries are presented below. Some of these applications (such as real-time translation of spoken language) might take a decade or more to develop, while others (such as credit application scoring) might be put into use before 1990.

- Finance - credit application scoring, credit line use analysis, new product analysis and optimization, corporate financial analysis, customer set characterization.
- Banking - marketing studies, check reading, physical security enhancement, loan evaluation, customer credit scoring.
- Insurance - insurance policy application evaluation, payout trend analysis, new product analysis and optimization.
- Defense - radar/sonar/image processing (noise reduction, data compression, feature extraction, pattern recognition), opposing force models, weapons aiming and steering, novel sensor systems.
- Entertainment - market analysis and forecasting, special effects, animation, restoration.
- Automotive - assembly jig control, warranty repair analysis, automobile autopilot.
- Transportation - waybill processing, vehicle scheduling and routing, airline fare management.
- Telecommunications - speech and image compression, automated information services, realtime translation of spoken language, customer payment processing systems.
- Retail Franchise - outlet site location selection
- Securities - stock and commodity trading advisor systems, technical market/ company / commodity analysis, customer credit analysis.
- Robotics - vision systems, appendage controllers, tactile feedback gripper control.
- Manufacturing - low cost visual inspection systems, nondestructive testing, fabrication plan development.
- Electronics - VLSI chip layout, process control, chip inspection.
- Aerospace - avionics fault detection, aircraft/spacecraft control systems, autopilot enhancements.”



NEUROCOMPUTER APPLICATIONS

Robert Hecht-Nielsen
Hecht-Nielsen Neurocomputer Corporation
5893 Oberlin Drive
San Diego, CA 92121
619-546-8877

ABSTRACT

Neurocomputing is the engineering discipline concerned with non-programmed adaptive information processing systems called *neural networks* that develop their own algorithms in response their environment. Neurocomputing is a fundamentally new and different information processing paradigm. It is an alternative to the programming paradigm. This paper discusses the nature of neurocomputing, surveys some specific neural network information processing capabilities, and discusses applications of neurocomputing.

1 Introduction

The Programming Paradigm and its Problems

Currently, essentially all automated information processing is based upon the ‘glorified adding machine’ paradigm spelled out by John von Neumann in his 1945 consultant’s report to the ENIAC project. Although initially bound by computing speed and program size limitations, computers soon became bound by software problems. Software was found to be difficult and expensive to produce. High-quality software was possible only under the most careful, lengthy, and iterative testing and debugging. Grace Hopper’s invention of the compiler in 1952 and John W. Backus’ invention of FORTRAN in 1955 were both designed to help solve this problem. Later developments such as Niklaus Wirth’s campaign for structured programming and the invention of new languages and tools such as Pascal, Ada, APSE, and Object Oriented Programming were also designed to help solve this “software bottleneck” problem.

On a more fundamental level, the problem is that in order to get a programmed computer to carry out an information processing function some humans must both understand that function and write down an algorithm for implementing it. If the function is simple, such as keeping track of bank account balances, then the problem is to design a suitable algorithm and human interface. If the problem is complex, such as computed axial tomography, then you must first wait for geniuses such as Johann Radon and Alan Cormack to be born, and then proceed with algorithm and interface development.

In summary, the development of software for carrying out simple information processing tasks is difficult, expensive, and time consuming. The development of software for complex information processing tasks is even more difficult, because of the need to wait for a genius who can discover the needed algorithm.

Neurocomputing: A New Information Processing Paradigm

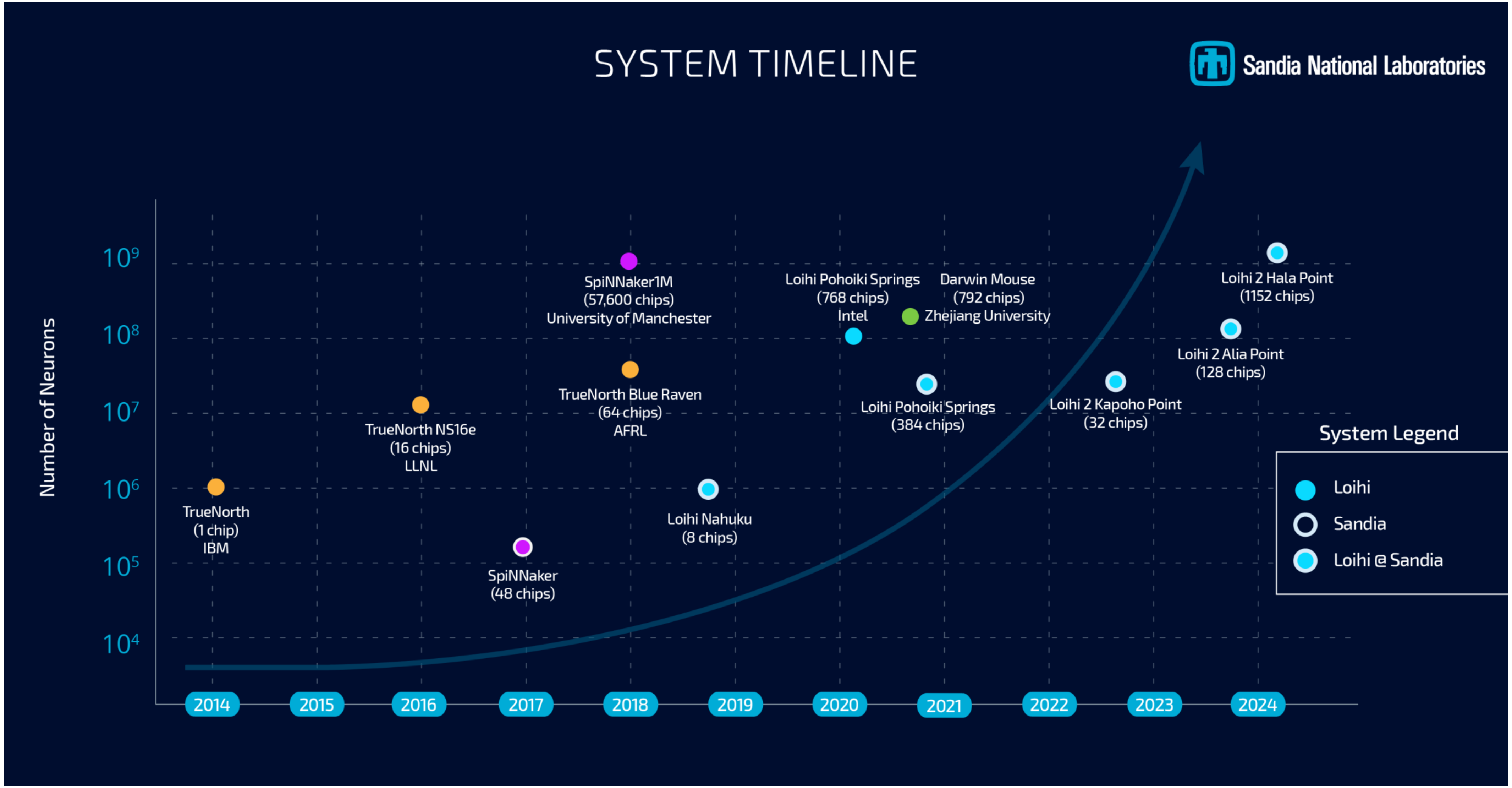
In contrast to software development, wouldn’t it be nice if all we had to do to develop an information processing capability was to specify it exactly and give examples of its operation? We can easily specify and give examples of many highly desirable information processing systems for which the software cannot yet be written. For example, what about a speaker-independent continuous speech recognition system, or an automobile autopilot, or a handwritten character reader? How about a

R. Eckmiller et al. (eds.), *Neural Computers*
© Springer-Verlag Berlin Heidelberg 1989

Source: Hecht-Nielsen, R. (1989). Neurocomputer Applications. In: Eckmiller, R., v.d. Malsburg, C. (eds) Neural Computers. Springer Study Edition, vol 41. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-83740-1_45

And many more

Today – Billion+ Neuron Neuromorphic



Sandia Labs & Intel - Hala Point

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point

1.15 Billion Neurons

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point

1.15 Billion Neurons

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Hala Point Specs & Performance

System

- 1152 Loihi 2 chips
- 140,544 neuromorphic cores
- 2,304 x86 cores
- 6U data center chassis
- 2600 Watts power (max)

Capacity

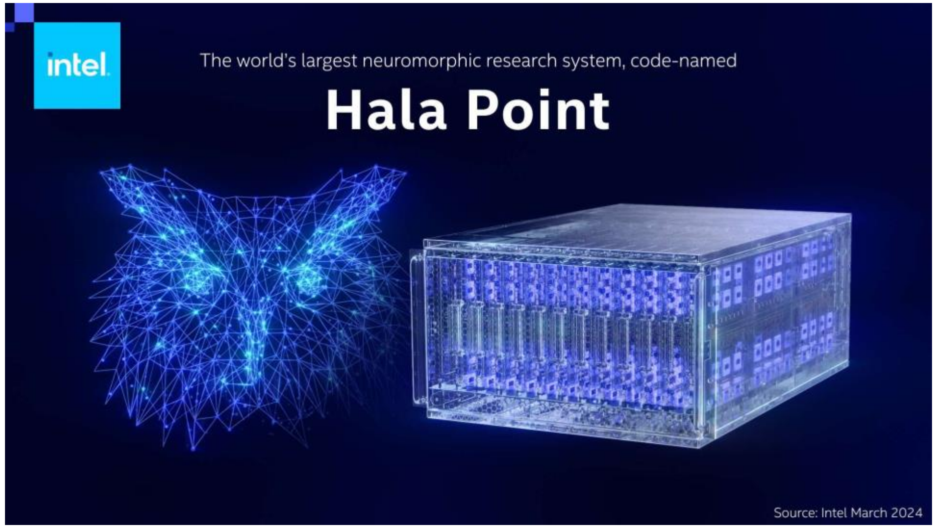
- 1.15 billion neurons
- 128 billion synapses

Speed

- 380 trillion synaptic ops/second
- 240 trillion neuron ops/second
- 16 petabytes/sec memory bandwidth
- 3.5 PB/s inter-core communication bandwidth
- 5 TB/s inter-chip communication bandwidth

Performance Characterization

- Up to 20 quadrillion operations per second (or 20 petaops)
- 15 trillion 8-bit operations per second per watt (TOPS/W)
  - 10:1 sparse connectivity & event-driven activity via sigma-delta neuron model
  - MLP network with 14,784 layers; 2048 neurons/layer, 8-bit weights; random-noise activity

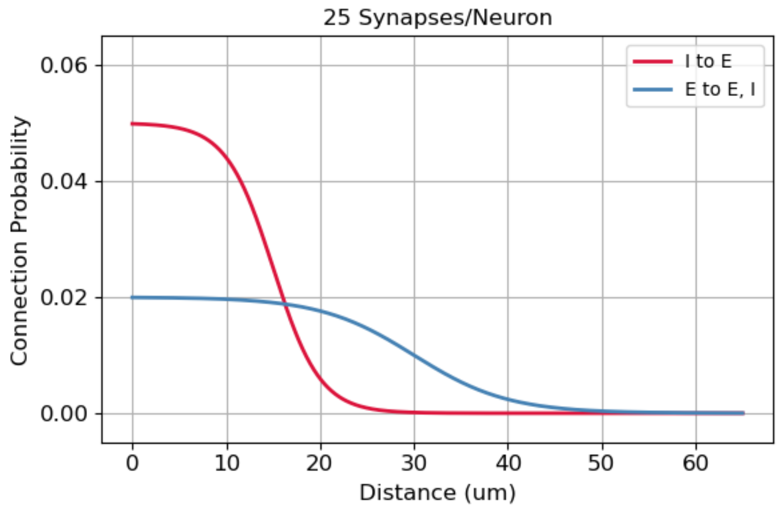

intel
The world's largest neuromorphic research system, code-named
Hala Point
Source: Intel March 2024

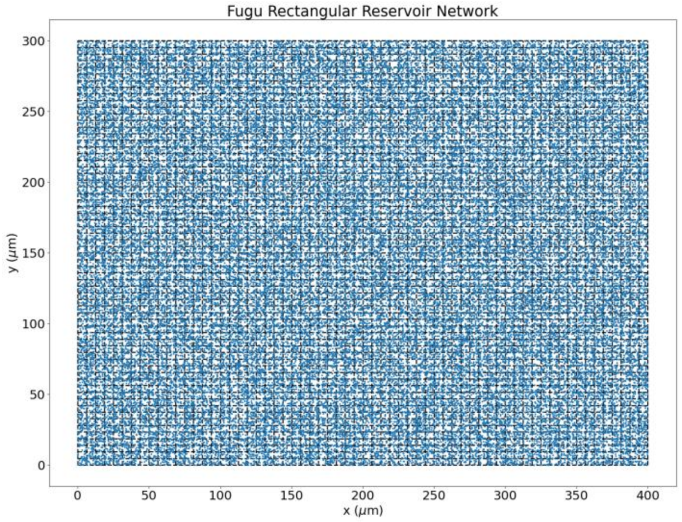
Source: https://www.intel.com/content/dam/www/central-libraries/us/en/images/2024-04/newsroom-intel-hala-point-owl.jpg.rendition.intel.web.1920.1080.jpg

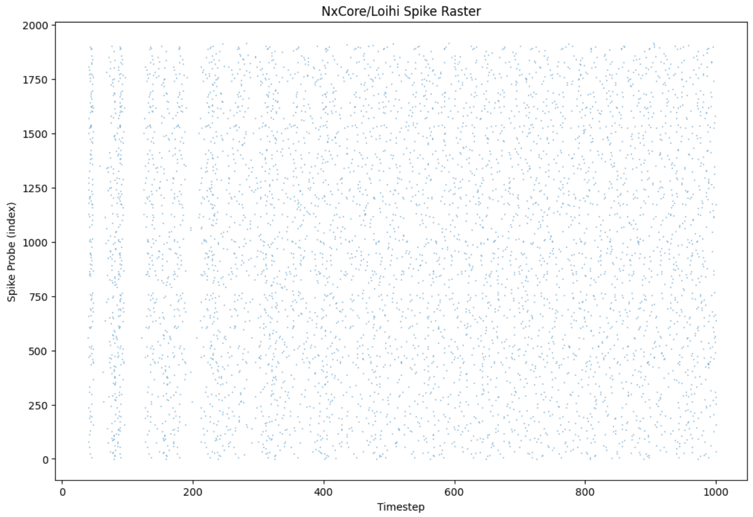
Hala Point Workload Example – Scalable Reservoir Network

- Spatially dependent connectivity
- Fan-in/out of ~25 synapses/neuron
- Split into 4096 partitions
  - Equivalent to number of neurocores mapped to
  - Vary number cores/chips

- Neurons (blue dots) randomly placed & black lines are partition boundaries
- STACS mapping to Hala Point
- Sub-sampled spike raster (60k neurons)







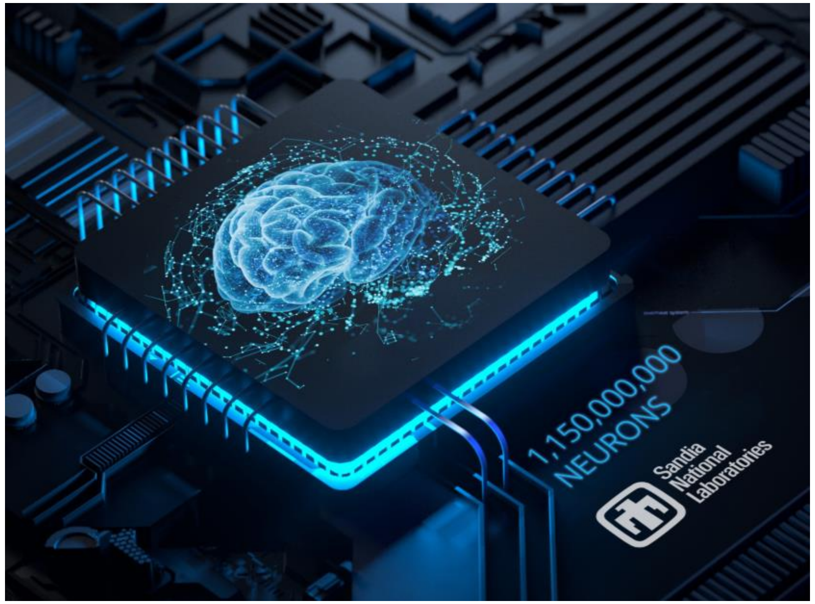
Conclusions

The Neural Computing Phenomenon is Truly Amazing

- "Over the past three years, there has been a veritable explosion of interest in neural networks and neurocomputers, even though its foundations have been around since the 1940s." Treleaven

- To deliver on that promise
  - Scaling advances are supporting: neuromorphic hardware, algorithm innovation, simulation, etc.
  - Game theoretic view illustrates why some of the best intentioned technical pursuits may not have the impact they desire

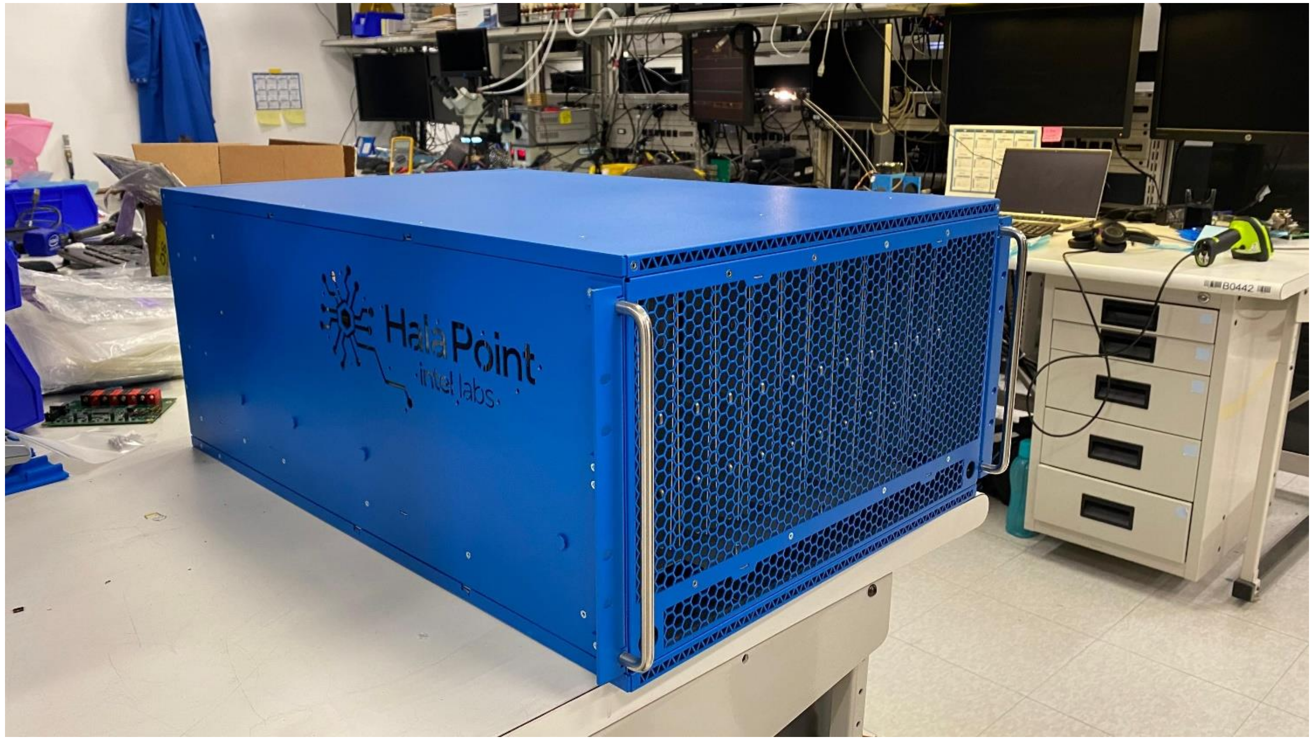
- Neuromorphic computing offers exciting research opportunities exploring - Which applications? How? When?



SANDIA LABS TUTORIALS				
SIMULATION TOOL FOR ASYNCHRONOUS CORTICAL STREAMS (STACS)	BUILDING SCALABLE, COMPOSABLE SPIKING NEURAL ALGORITHMS WITH FUGU (AN INTRODUCTION)	CROSSSIM: A HARDWARE/SOFTWARE CO-DESIGN TOOL FOR ANALOG IN-MEMORY COMPUTING	N2A – NEURAL PROGRAMMING LANGUAGE AND WORKBENCH	SANA-FE: SIMULATING ADVANCED NEUROMORPHIC ARCHITECTURES FOR FAST EXPLORATION

Thank You!

ASCI RED – 1st TFLOPS & Largest HPC



Questions?

Thank You!

Questions?