ECE 598NSG/498NSU
Deep Learning in Hardware
Fall 2020

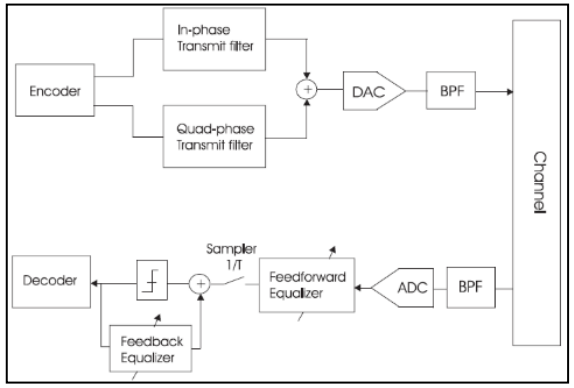
Introduction

Naresh Shanbhag
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

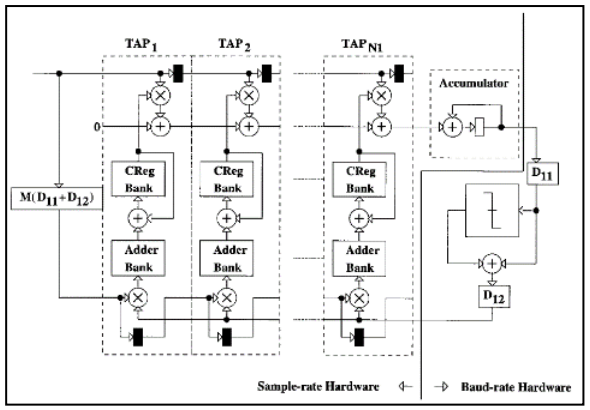
http://shanbhag.ece.uiuc.edu

Research Area – Energy-efficient Inference Systems in Silicon

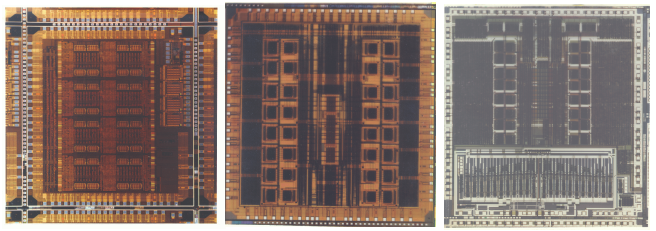
Algorithms



Architectures



Integrated Circuits



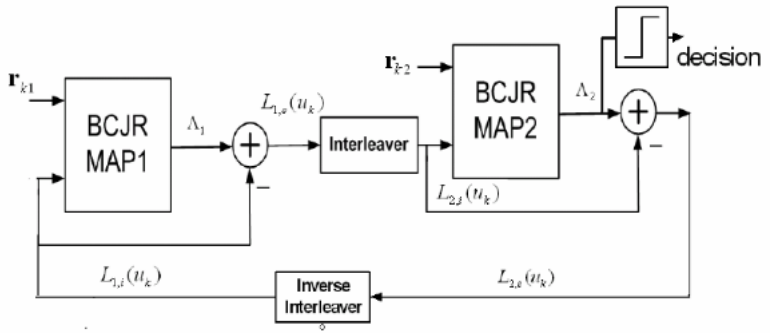
AT&T Bell Labs ['93-95]

(Lead Chip Architect)

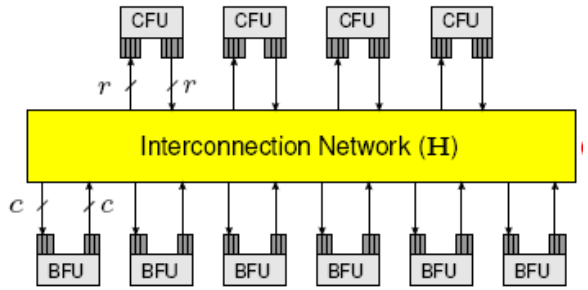
[51.84 Mb/s VDSL chip-sets]

[very high-speed digital subscriber line (VDSL)]

Algorithms

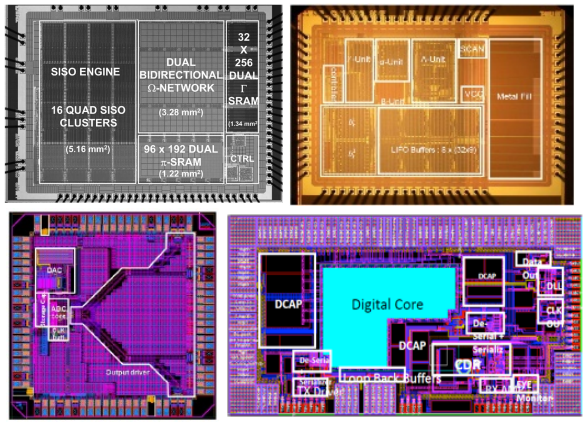


Architectures



[error control decoders, ADC-based links]

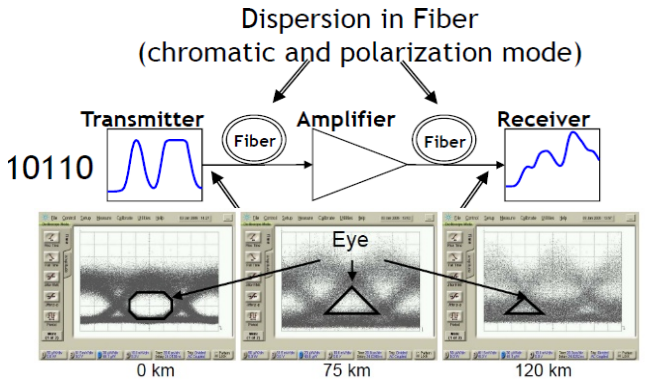
Integrated Circuits



UI ['95-'05]

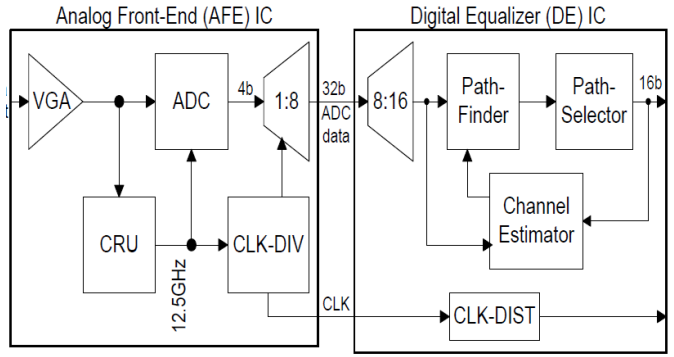
Intersymbol Communications, Inc. (2001-07)

Algorithms

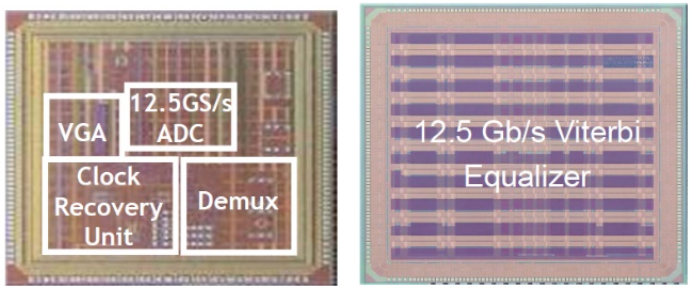


Reach extension: 80km → 200km

Architectures



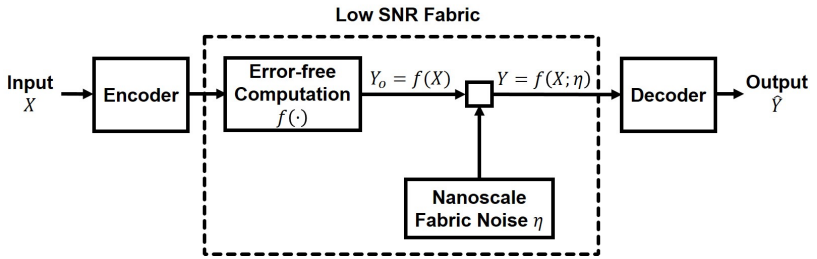
Integrated Circuits



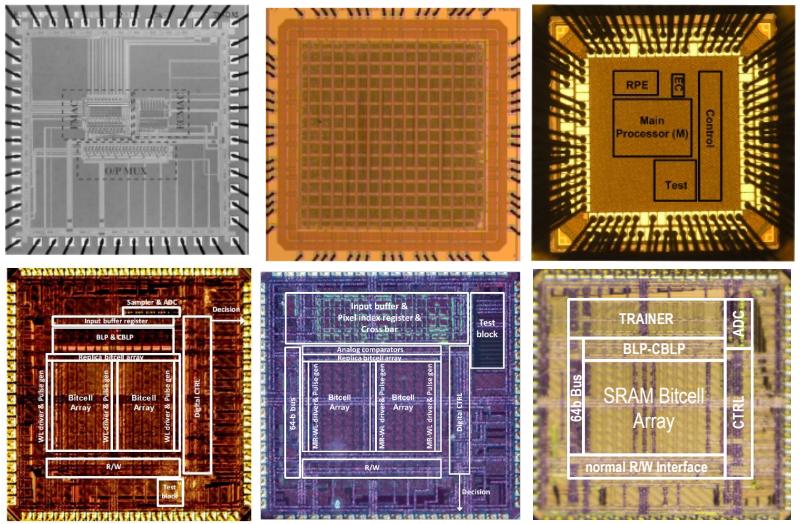
- Co-founder and CTO: fund-raising, technical leadership, customer visits & field trials
- Core technical team – Raj Hegde (Ph.D.'01); H.-M. Bae (Ph.D.'04); J. Ashbrook (M.S.'00)
- Established ADC-based receiver as an industry standard for 100G+ optical links
- 2006 IEEE Solid-State Circuits Society Best Paper Award;
- acquired by Finisar Corp (2007): currently a Finisar Design Center (Fox Dr., Champaign)

Shannon-inspired Model of Computation

Framework

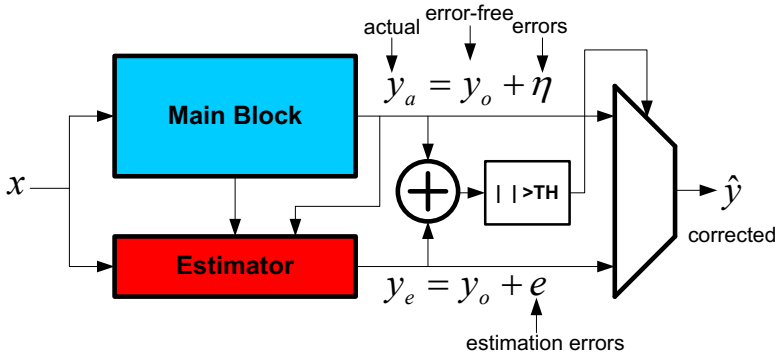


Integrated Circuits

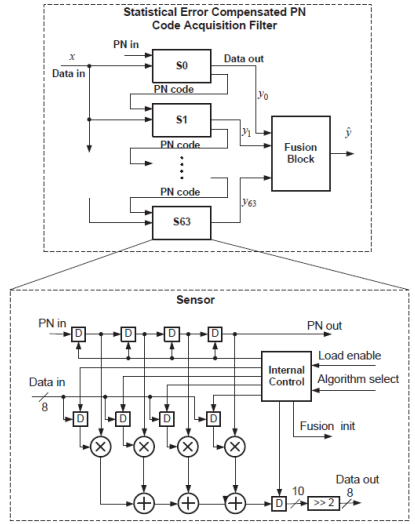


Deep In-memory Architectures For Machine Learning

Algorithms



Architectures



- IEEE Spectrum [‘10,’18]; 11 Ph.D.s
- Proceedings of IEEE, Special Issue on *Non von Neumann Computing*, January 2019.

23 Mar 2018 | 15:00 GMT

To Speed Up AI, Mix Memory and Processing

New computing architectures aim to extend artificial intelligence from the cloud to smartphones

By **Katherine Bourzac**

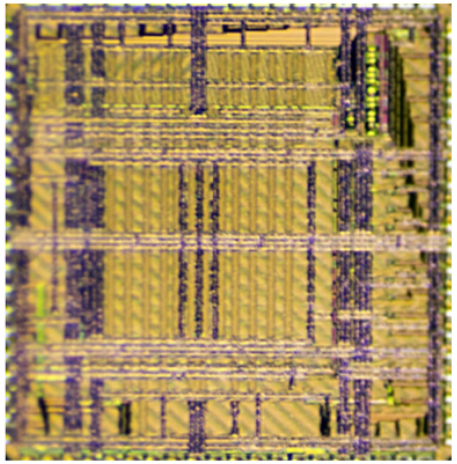


Image: Sujan Gonugondla

Tearing Down Walls: This prototype features a new chip design called deep in-memory architecture.

If John von Neumann were designing a computer today, there's no way he would build a thick wall between processing and memory. At least, that's what computer engineer Naresh Shanbhag of the University of Illinois at Urbana-Champaign believes. The eponymous von Neumann architecture was published in 1945. It enabled the first stored-memory, reprogrammable computers—and it's been the backbone of the industry ever since.

Now, Shanbhag thinks it's time to switch to a design that's better suited for today's data-intensive tasks. In February, at the International Solid-State Circuits Conference (ISSCC), in San Francisco, he and others made their case for a new architecture that brings computing and memory closer together. The idea is not to replace the processor altogether but to add new functions to the memory that will make devices smarter without requiring more power.

The Deep In-memory Architecture (DIMA)

"An Energy-efficient In-memory Architecture for Machine Learning"

[Verma, Mitra, Wong, Shanbhag]

https://spectrum.ieee.org/computing/hardware/to-speed-up-ai-mix-memory-and-processing

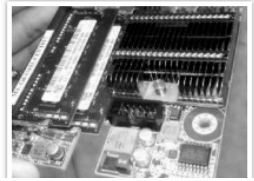
Market Buzz

THE PLATFORM

HOME COMPUTE STORE CONNECT CONTROL CODE ANALYZE HPC ENTERPRIS

MICROSOFT EXTENDS FPGA REACH FROM BING TO DEEP LEARNING

August 27, 2015 Timothy Prickett Morgan

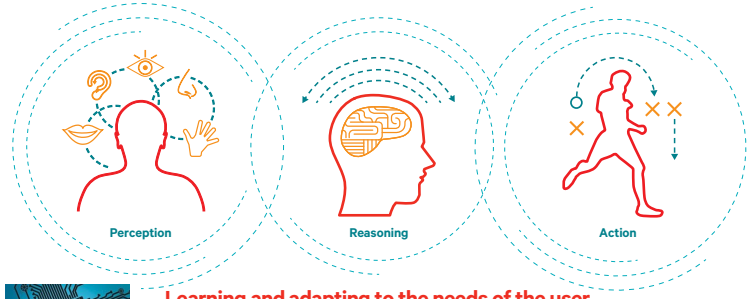


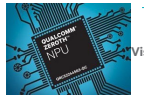
After three years of research into how it might accelerate its Bing search engine using field programmable gate arrays (FPGAs), Microsoft came up with a scheme that would let it lash Stratix V devices from Altera to the two-socket server nodes in the minimalist Open Cloud Servers that it has designed expressly for its hyperscale datacenters. These CPU-FPGA hybrids were rolled out into production earlier this year to accelerate Bing page rank functions, and Microsoft started

QUALCOMM

Introducing Qualcomm® Zeroth™ Platform

Qualcomm Technologies' first cognitive computing platform designed for on-device intelligence.





Learning and adapting to the needs of the user.

Visual perception • Intelligent connectivity • Intuitive security • Always-on awareness
Immersive multimedia • Speech and audio recognition • Natural interaction

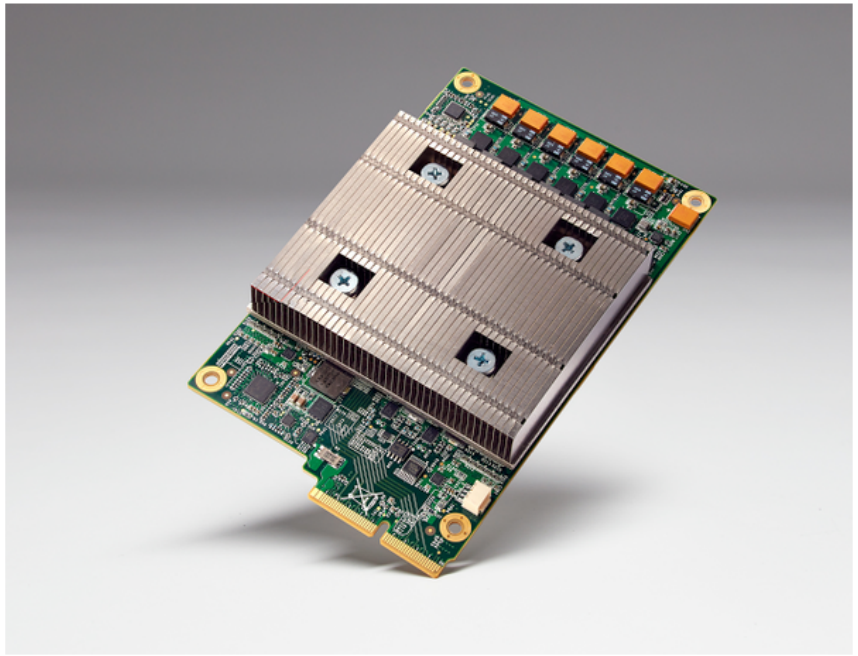
Find out more about Zeroth Platform at qualcomm.com/zeroth

Tom's Hardware › Processors › Processors News

Google's Big Chip Unveil For Machine Learning: Tensor Processing Unit With 10x Better Efficiency (Updated)

By Lucian Armasu MAY 19, 2016 9:10 AM - Source: Google Cloud Platform Blog | 18 COMMENTS

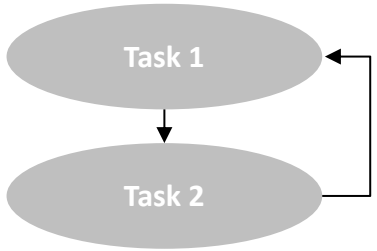
TAGS : Google + Processors +



While other companies are arguing about whether GPUs, FPGAs, or VPUs are better suited for machine learning, Google came out with the news that it has been using its own custom-built Tensor Processing Unit (TPU) for over a year, achieving a claimed 10x

DL in Hardware

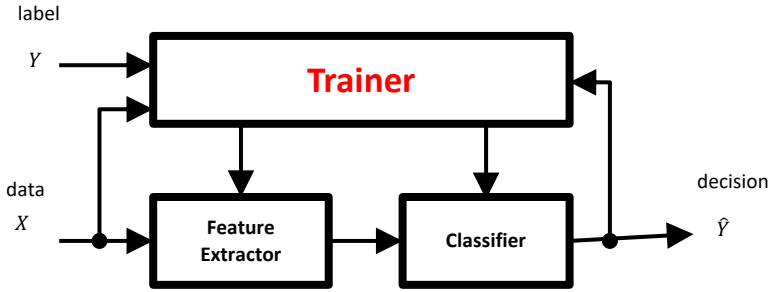
Applications



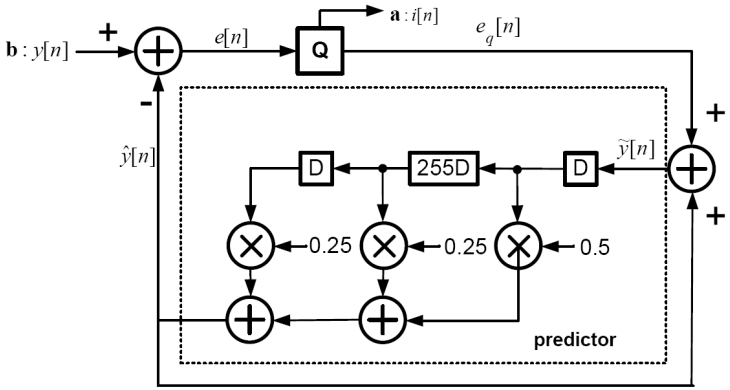
natural vision, EEG analysis, navigation, surveillance

Inference Tasks

Learning Models

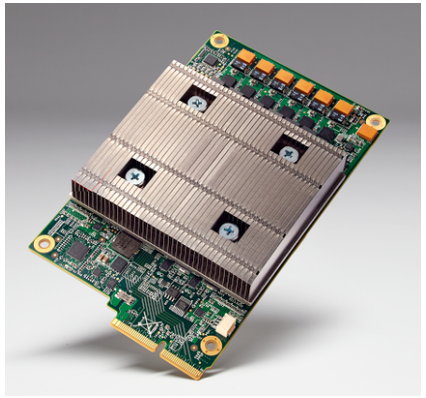


Finite-precision Architecture

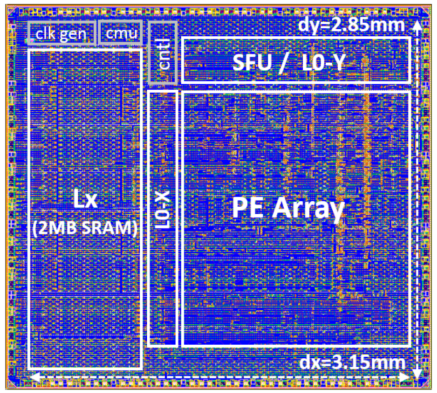


Hardware

module



integrated circuit



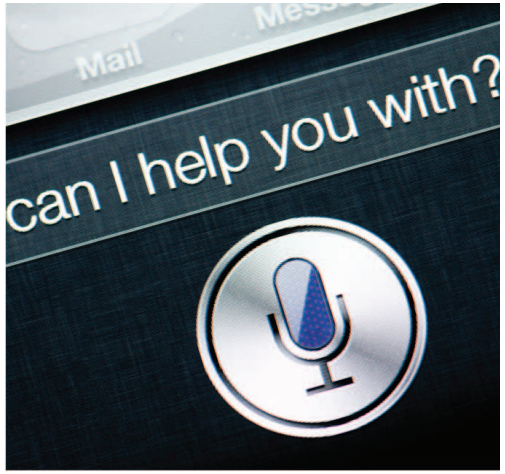
- Applications of AI and Deep Learning (DL)
- DL – A historical perspective
- DL in Hardware

DL – Applications

AI in our Lives

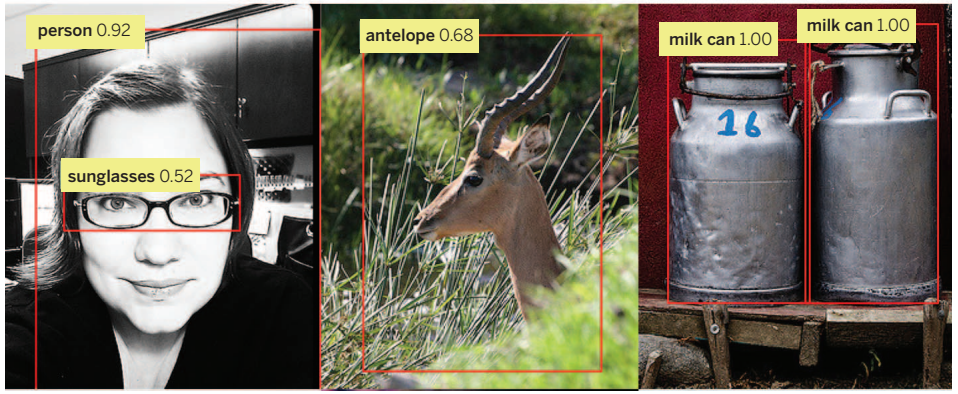
autonomous vehicles

natural language processing

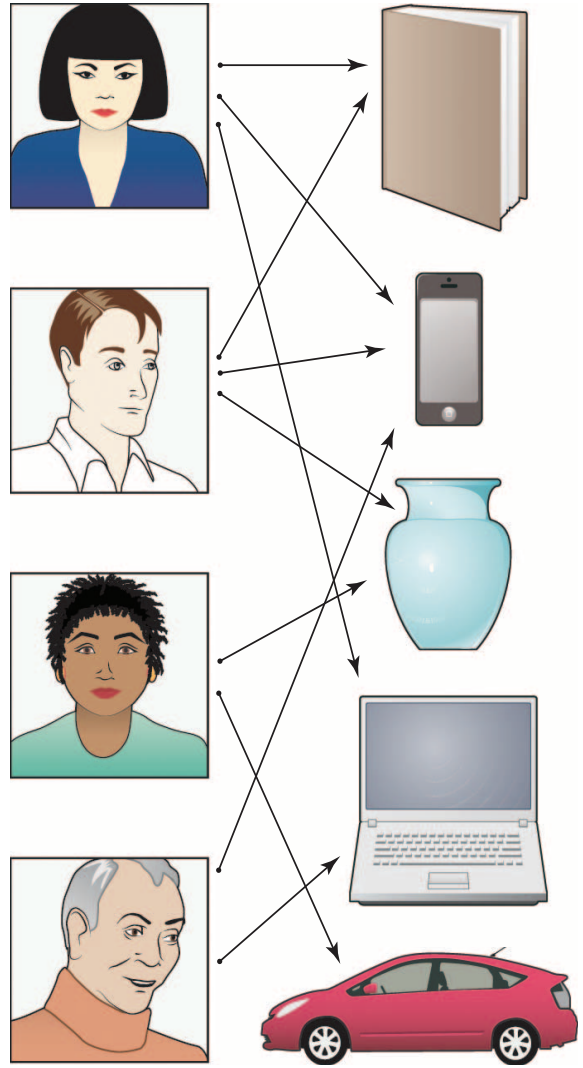


robotic surgery

computer vision – making computers see



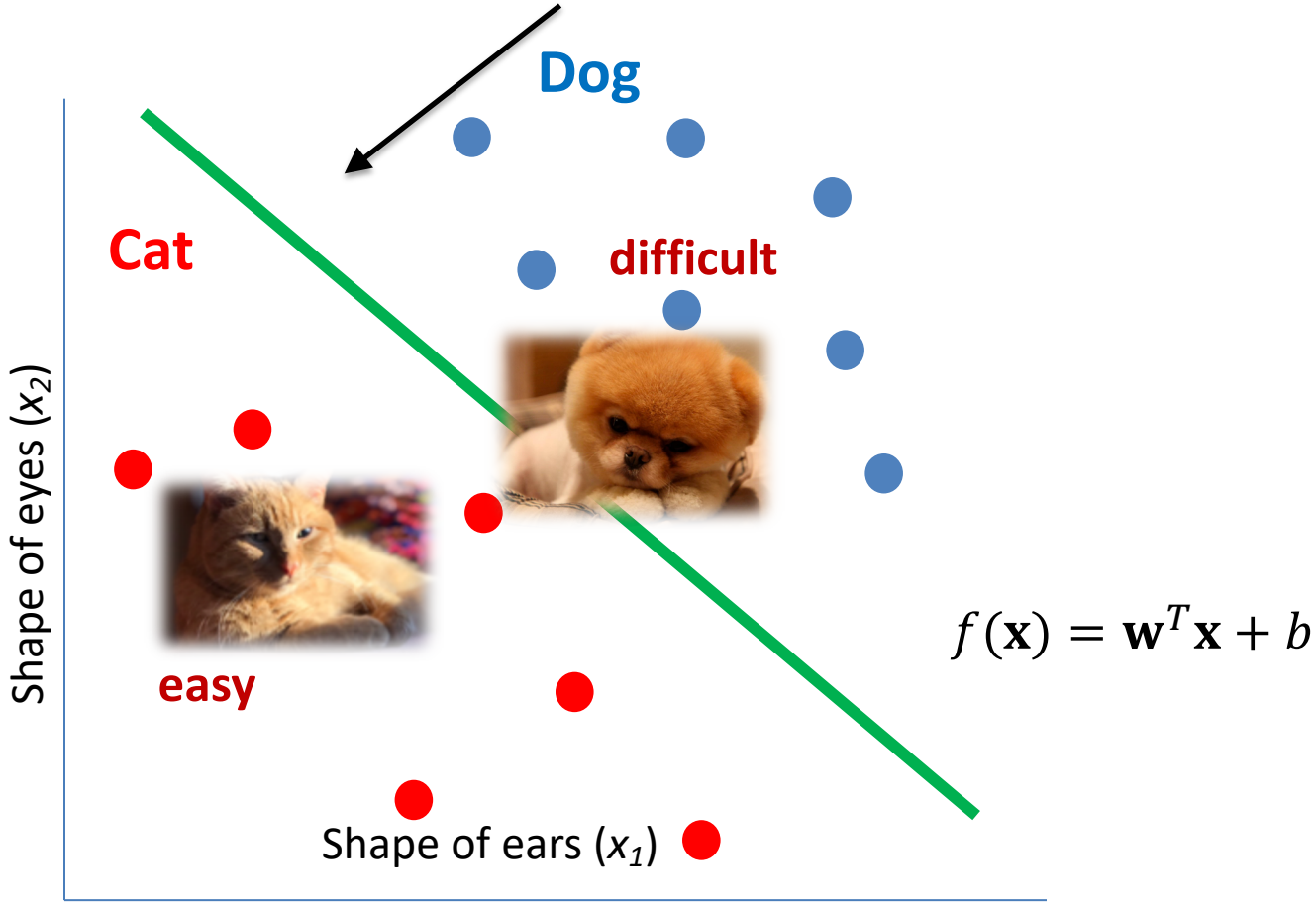
Recommendation systems



[Jordan, Mitchell, Science 2015]

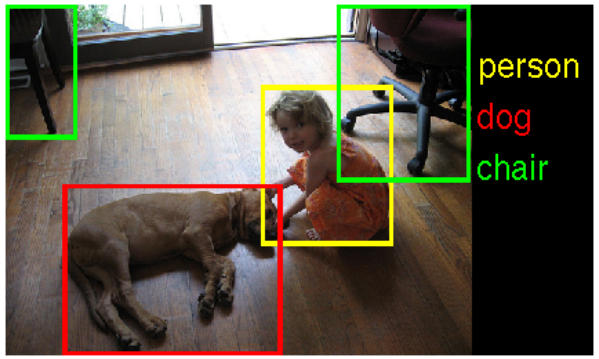
The Classification Task

machine has to learn this from features – shape of ears, eyes, whiskers…….

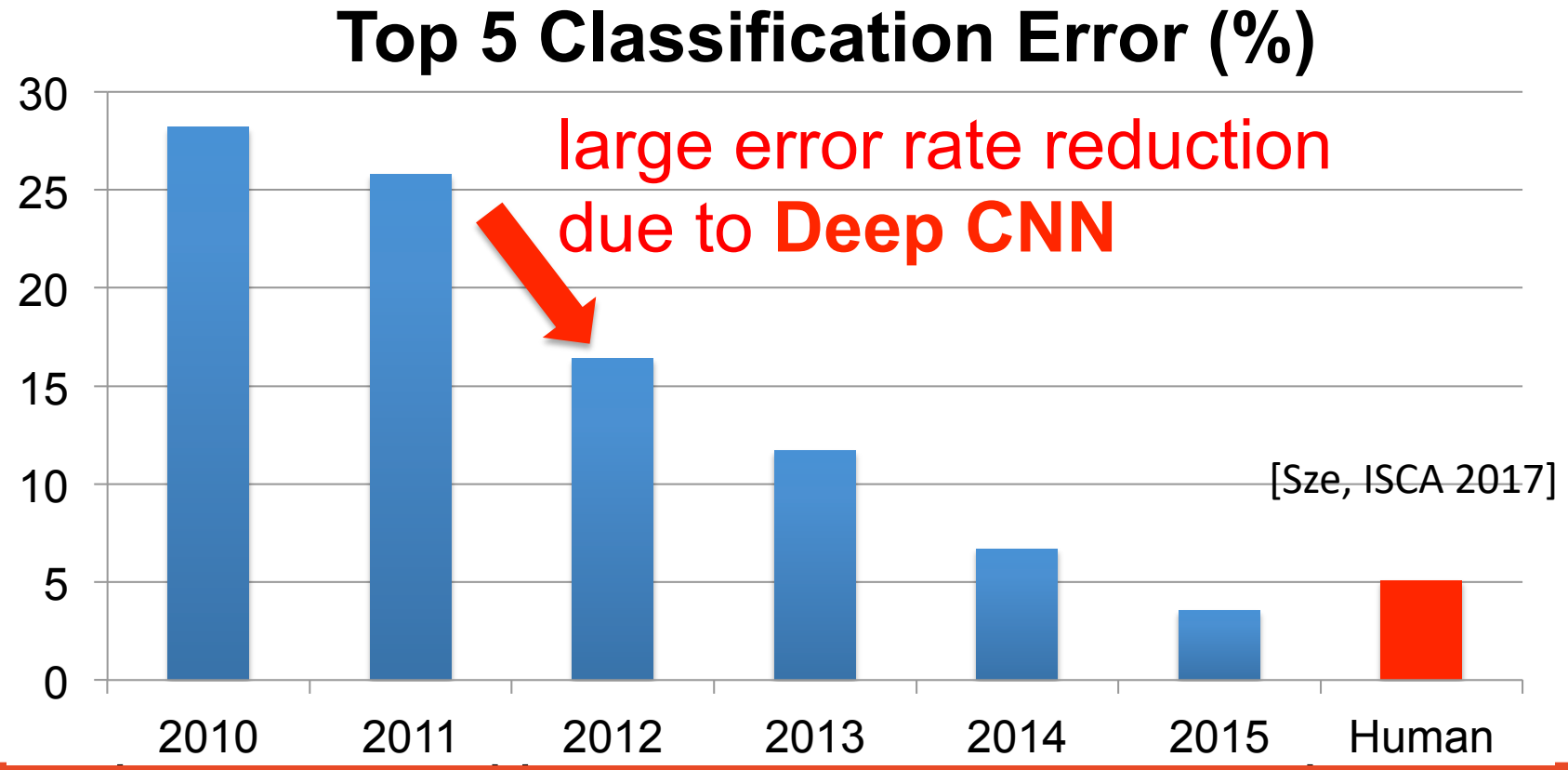


machines can make errors with difficult images

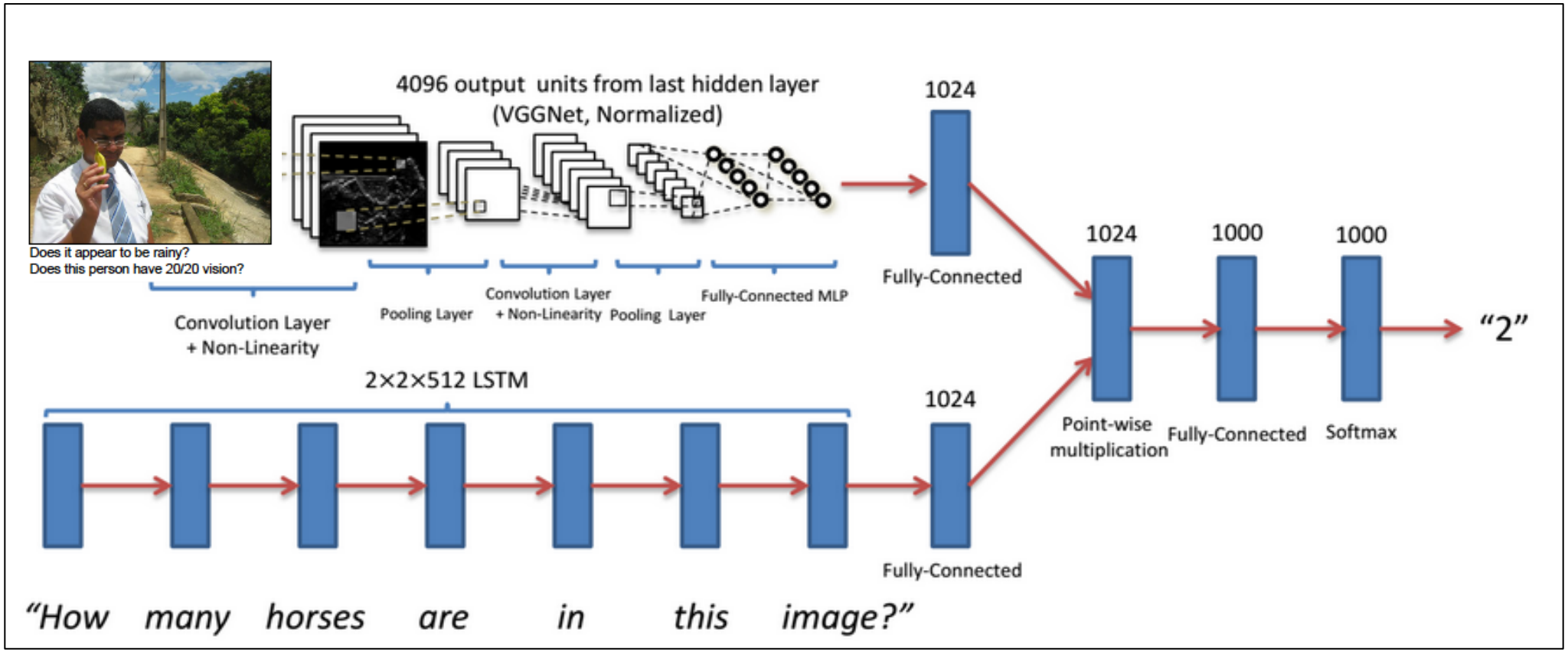
ImageNet Database - 1000 classes



ImageNet Classification Challenge



Visual Q/A – Measuring Machine Intelligence



Measuring Machine Intelligence Through Visual Question Answering — **Parikh et. Al (arXiv, 2016)**

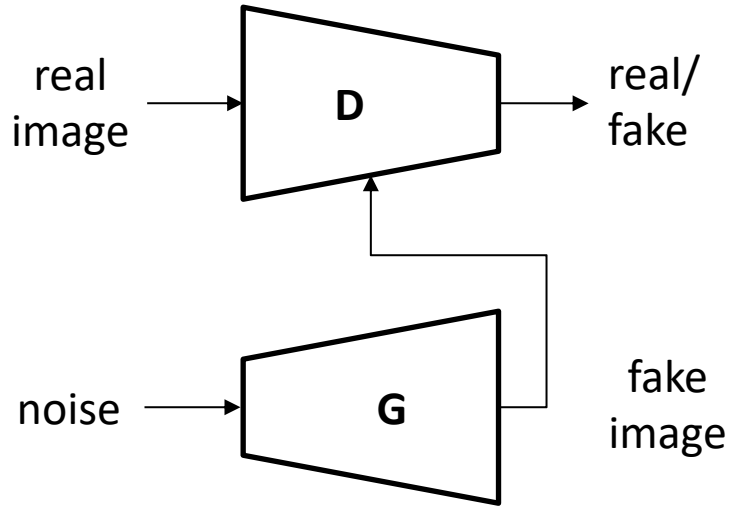
- integrate multi-modal representations: visual and speech
- DNNs for images; long short term memory (LSTM) for speech

Generative Adversarial Networks (GANs)

[NIPS 2014] **Generative Adversarial Nets**

Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, Yoshua Bengio‡

faces 'invented' by the machine

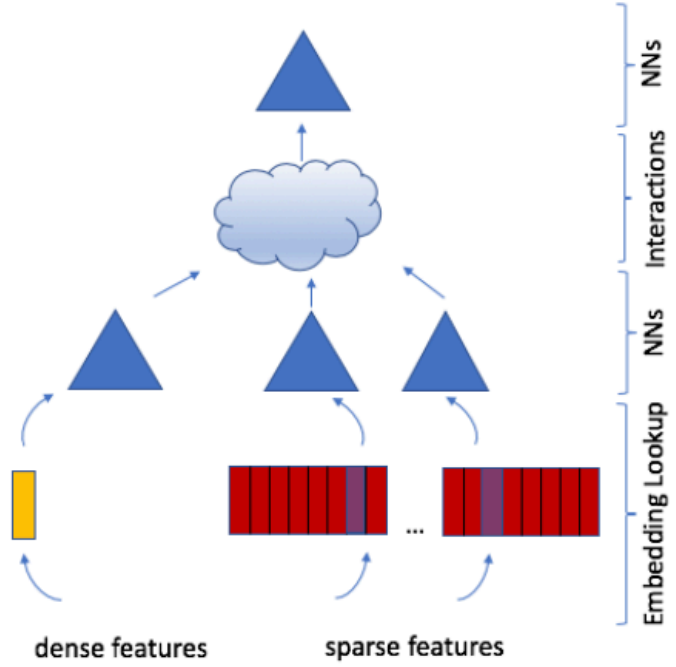


- G: generator DNN
- D: discriminator DNN
- G tries to fool D
- D tries to discriminate between real and generated samples

Recommendation Systems

Deep Learning Recommendation Model for Personalization and Recommendation Systems

Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi*, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta†, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong and Misha Smelyanskiy
Facebook, 1 Hacker Way, Menlo Park, CA 94065
{mnaumov,dheevatsa}@fb.com



- categorical data leads to 1-hot/multi-hot coded sparse features; need to map to dense features using embedding tables (> 10M cols, >10 rows)
- bottom DNNs → extract higher level features; top DNNs → predict CTR (click through rate = # of clicks per # of impressions)

- intelligence – a working definition:

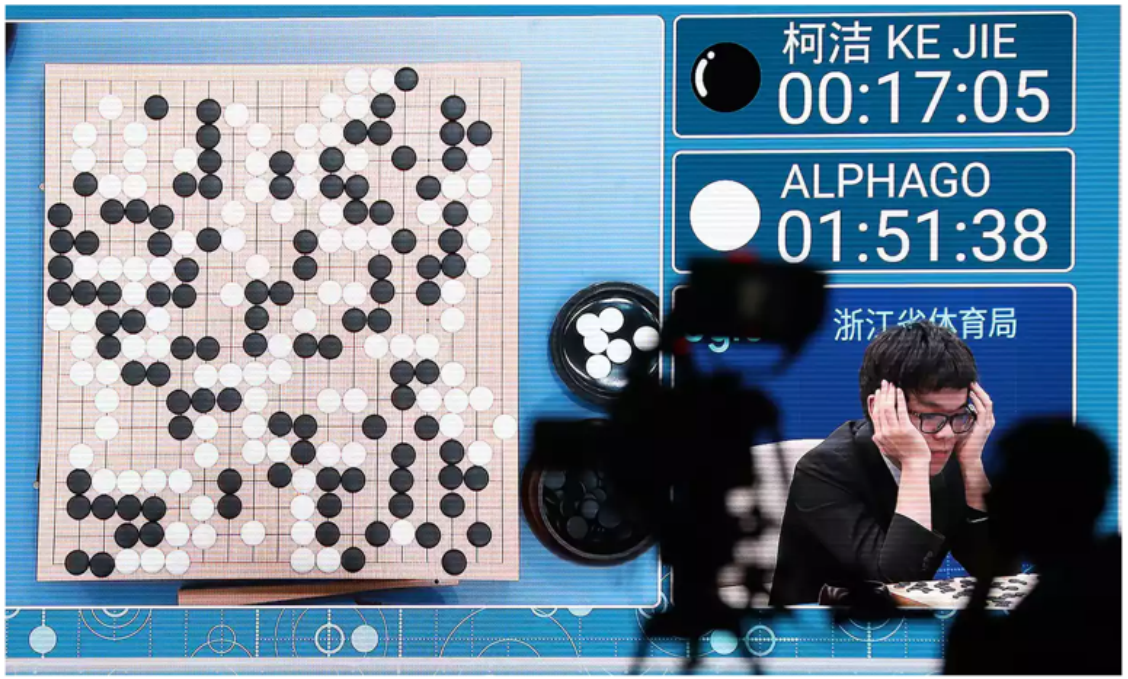
intelligent systems are those that **learn** from **examples** to make *accurate* **inferences** using *limited* **data** and **computational resources**, and are able to **adapt** to changes in its environment

- requires extracting **information** from **data**

The Energy Cost of Intelligence

World's best Go player flummoxed by Google's 'godlike' AlphaGo AI

Ke Jie, who once boasted he would never be beaten by a computer at the ancient Chinese game, said he had 'horrible experience'



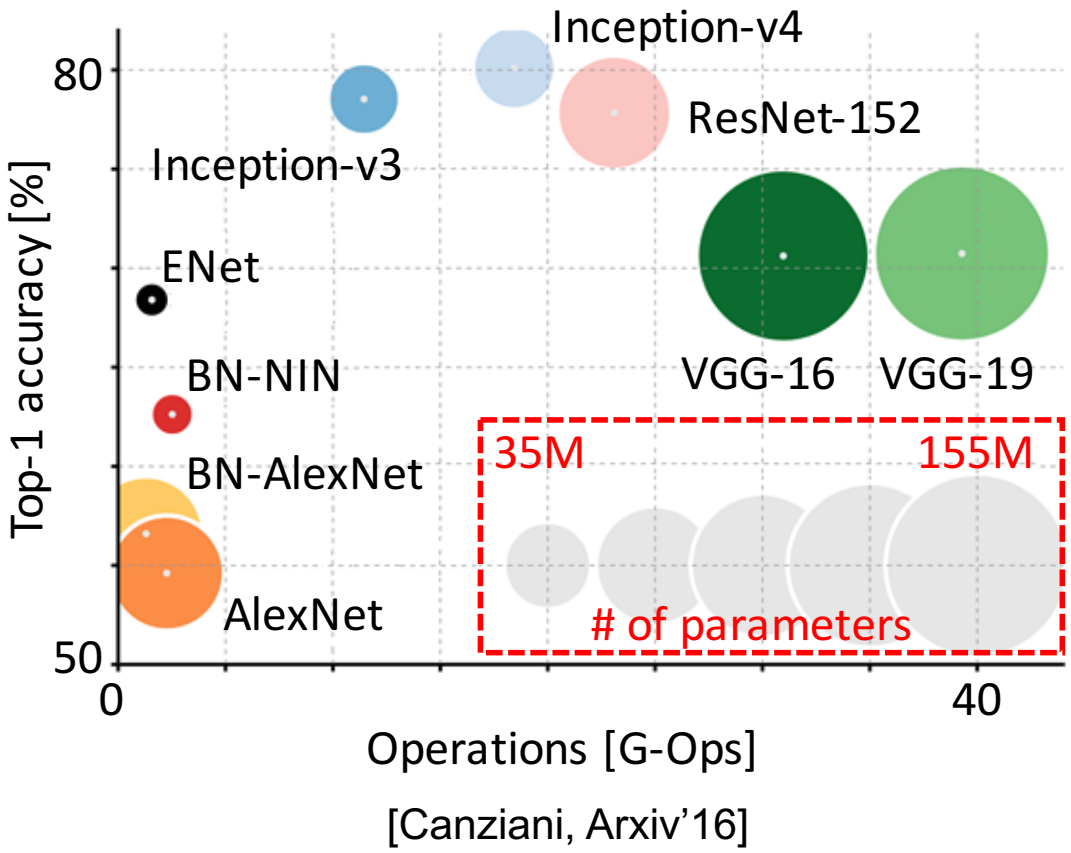
▲ A screen at the Future of Go summit in Wuzhen, China, keeps score during the match between Chinese player Ke Jie and Google's AI AlphaGo. Photograph: Wu Hong/EPA

The Guardian, May 2017

- machines can beat humans in specific cognitive tasks
- game of Go is complex → huge search space: $\sim 250^{150}$ (Go) vs. $\sim 35^{80}$ (Chess)
- AlphaGo machine: 1202 CPUs+176 GPUs
- HUGE Energy Cost $\sim 10{,}000\times$ more than human brain

The Efficiency Challenge in Machine Learning

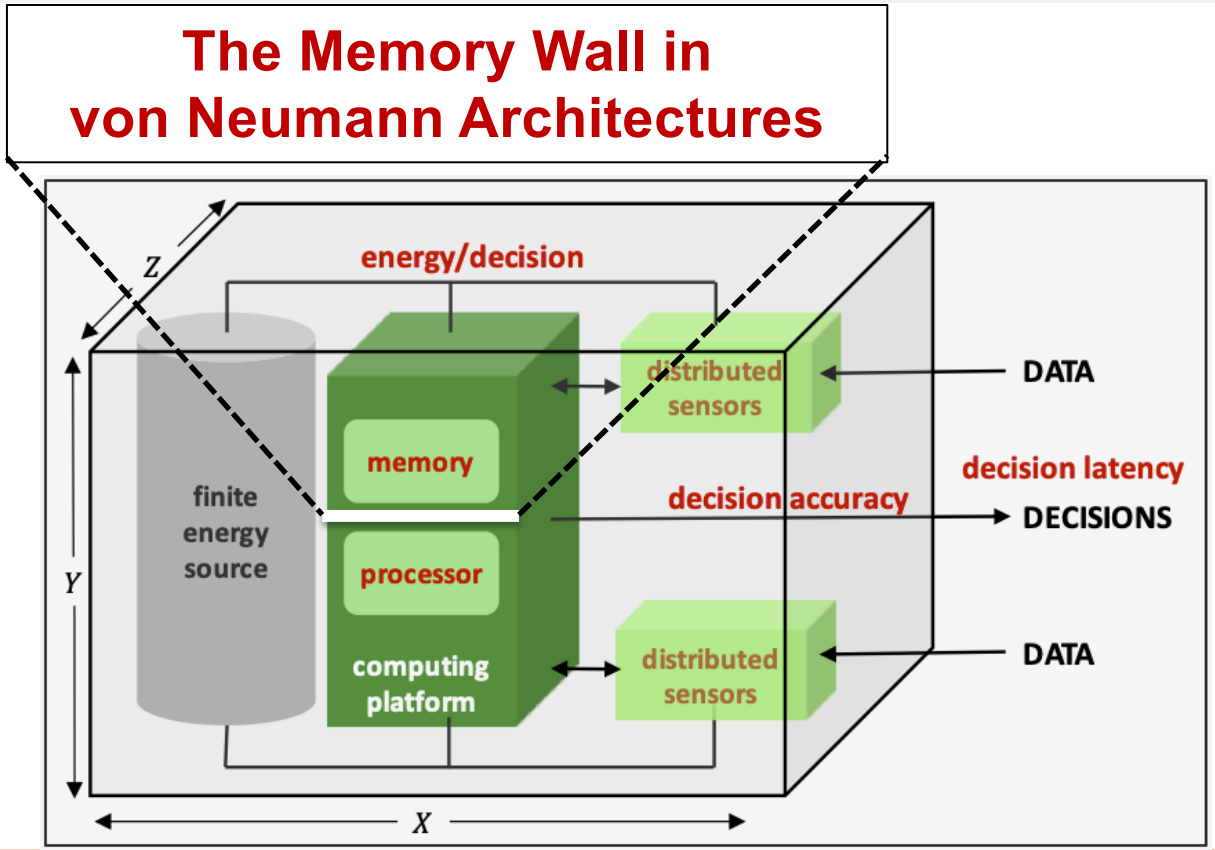
Model Complexity



[Canziani, Arxiv'16]

Data Movement Cost

$$\frac{E_{mem}}{E_{mac}} \approx \sim 100\times \text{ (SRAM)} \rightarrow \sim 500\times \text{ (DRAM)}$$



fundamental question

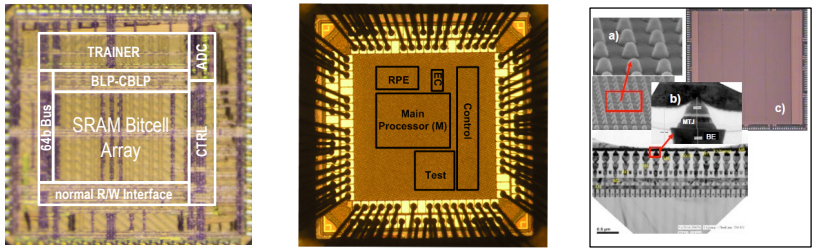
how do we design **learning machines** that **operate at the limits** of **accuracy**-**robustness**-**energy efficiency** with **guarantees**?

focus of ECE 498NSU/ECE 598NSG

Shanbhag Group Research Vectors

http://shanbhag.ece.Illinois.edu

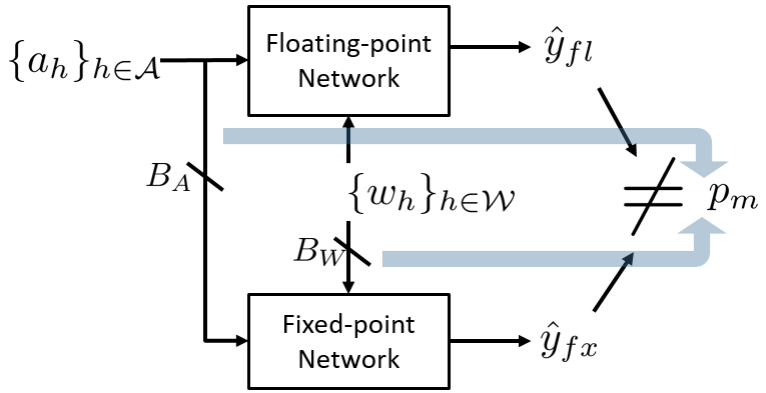
energy efficient circuit architectures



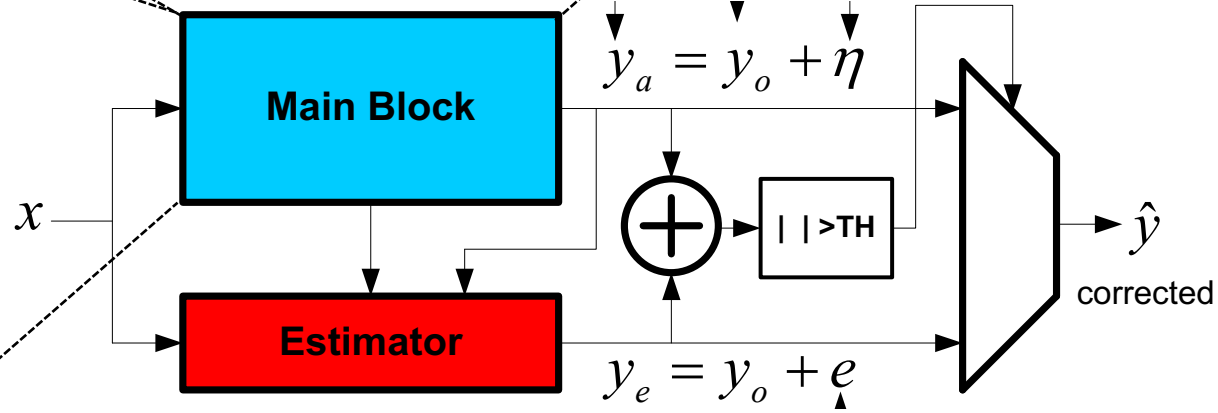
applications

- Computer Vision
- ATR
- RF Signal Processing
- Biomedical

low complexity algorithms



$$p_m \leq \Delta_A^2 E_A + \Delta_W^2 E_W$$



Shannon-inspired model of computation

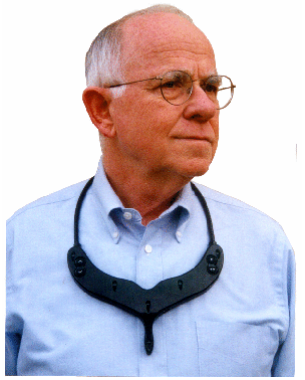
DL – A Historical Perspective

30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation

BERNARD WIDROW, FELLOW, IEEE, AND MICHAEL A. LEHR

PROCEEDINGS OF THE IEEE, VOL. 78, NO. 9, SEPTEMBER 1990

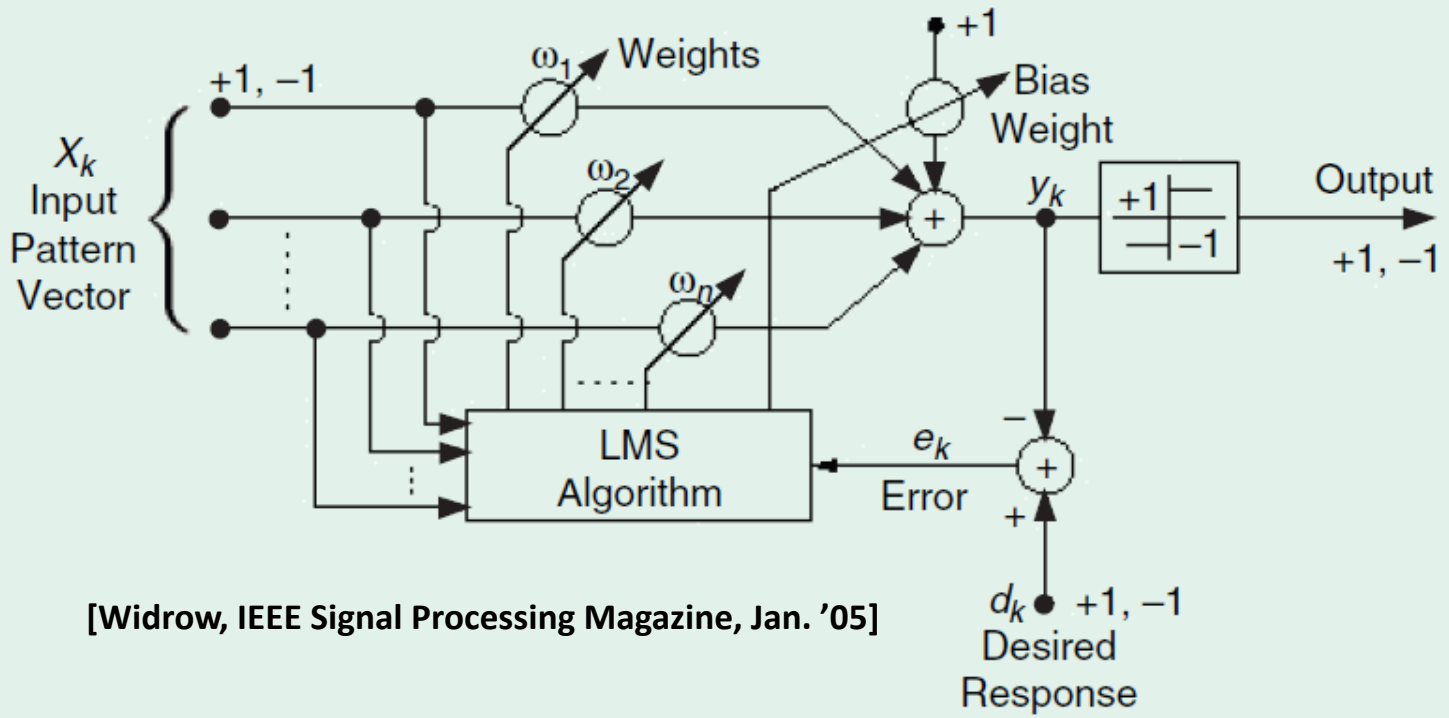
Widrow



- McCullough & Pitts neuron model (1943)
- Rosenblatt's perceptron (1958) – α-LMS learning rule (1960)
- LMS learning rule by Widrow and Hoff (1960) - μ-LMS
- ADALINE (**Ada**ptive **Li**near **Ne**uron) uses LMS to train linearly separable Boolean functions
- MADALINE (Multiple ADALINE, 1961) uses LMS to train multilayer ADALINE networks
  - gives rise to the need for backpropagation algorithm (Werbos, 1971, rediscovered in 1986 by Rumelhart, Hinton, and Williams)
- Hopfield networks (1982)
- Rumelhart, Hinton, Williams - backpropagation algorithm (1986)
- binary neurons used because multiplications were expensive

- 1969s-85s (**1st AI Winter**)
- Early 80s: Boltzmann machines and Hopfield networks (**second wave**)
- 1985: Backpropagation algorithm - needed differentiable non-linearities such as sigmoid function – unavailable until computers became powerful enough (1M FL MACs/sec)
- Lost popularity in 1990s (**2nd AI Winter**): 1) inadequate performance of computers; 2) small number of applications with large labelled data-sets; 3) difficulty in designing NN simulators; 4) lack of open source software
- Resurgence in 2013 (**third wave**): 1) improved methods; 2) large data-sets; 3) low-cost TFLOPS-class GPGPUs; 4) open source libraries (Torch, Theano, Caffe….)
- We are fortunate to be living in the third wave

The ADALINE Classifier



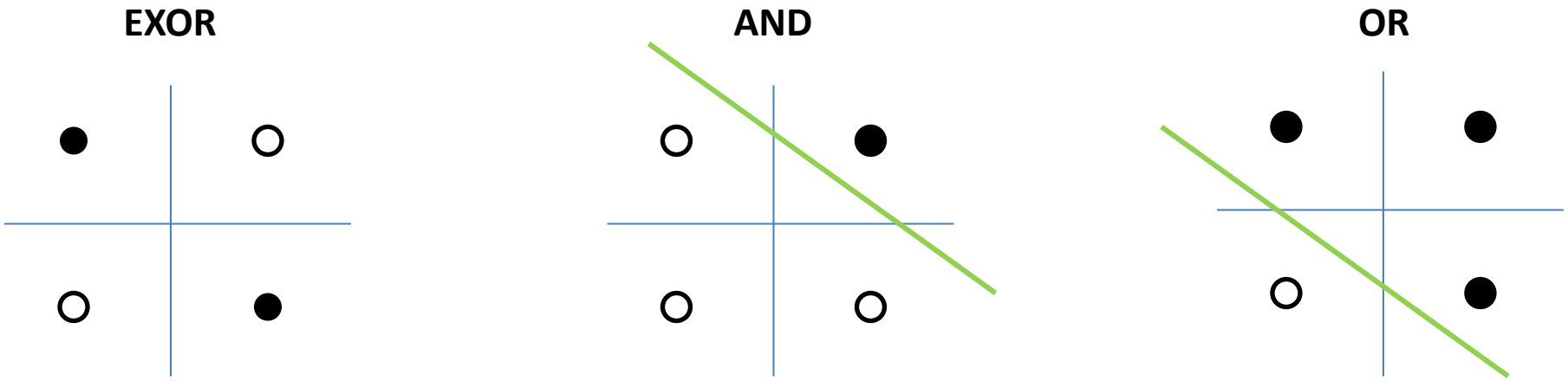
[Widrow, IEEE Signal Processing Magazine, Jan. '05]

$$\mu - LMS$$

$$W_{k+1} = W_k + \mu e_k X_k$$
$$e_k = d_k - y_k$$
$$y_k = W_k^T X_k$$

- binary inputs and outputs, continuous-valued weights and errors
  - Multiplications were prohibitively expensive (influence of H/W on Algorithms)
- can realize all ***linearly separable*** Boolean functions from 2^{2^N} possible functions of N inputs

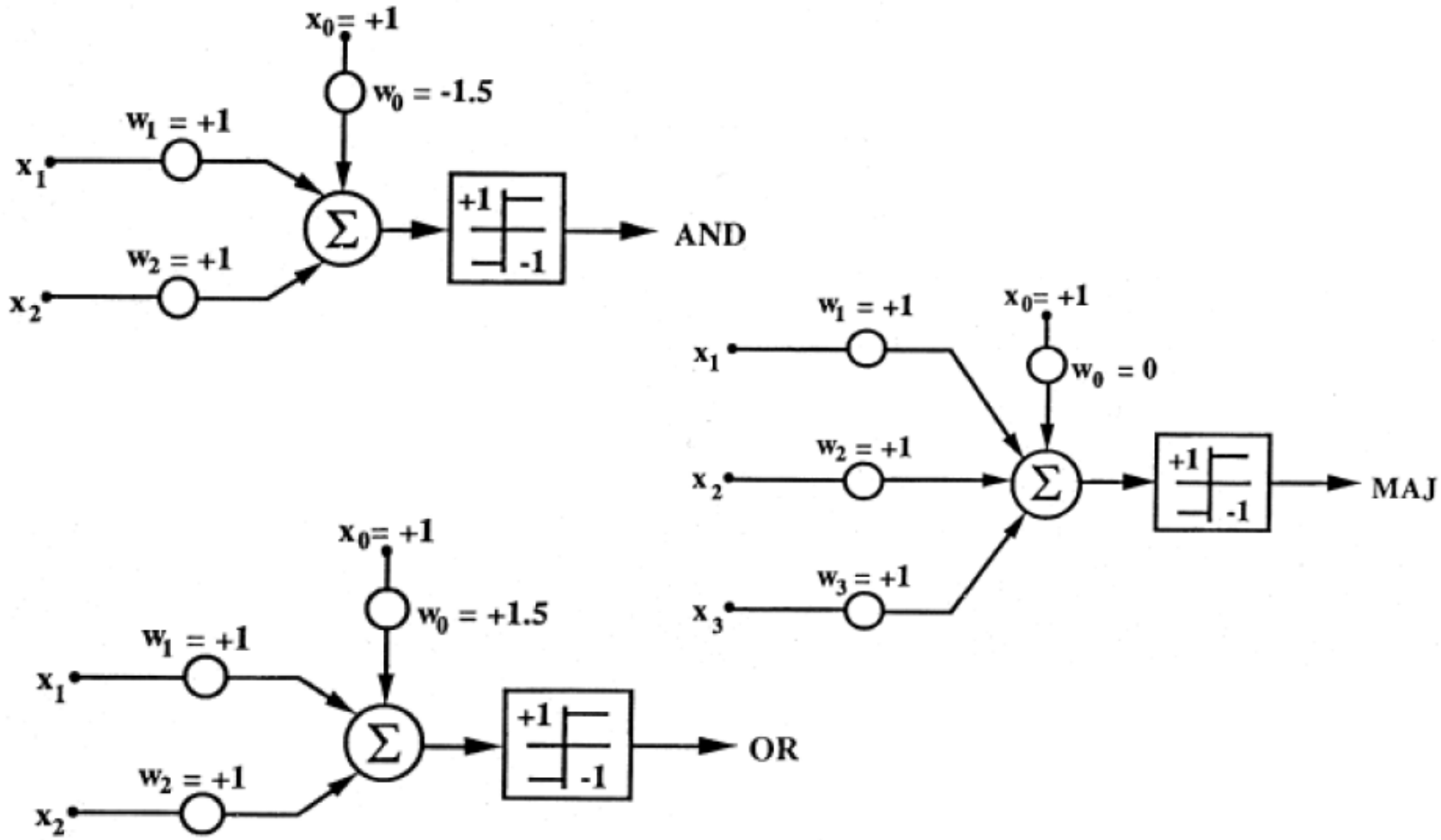
Linearly Separable Boolean Functions



linearly non-separable

- 1-set & 0-set separated by a hyperplane (linear decision boundary)
- $N = 2$: all 16 Boolean functions can be realized except EXOR/EXNOR

ADALINE – AND, OR and MAJORITY



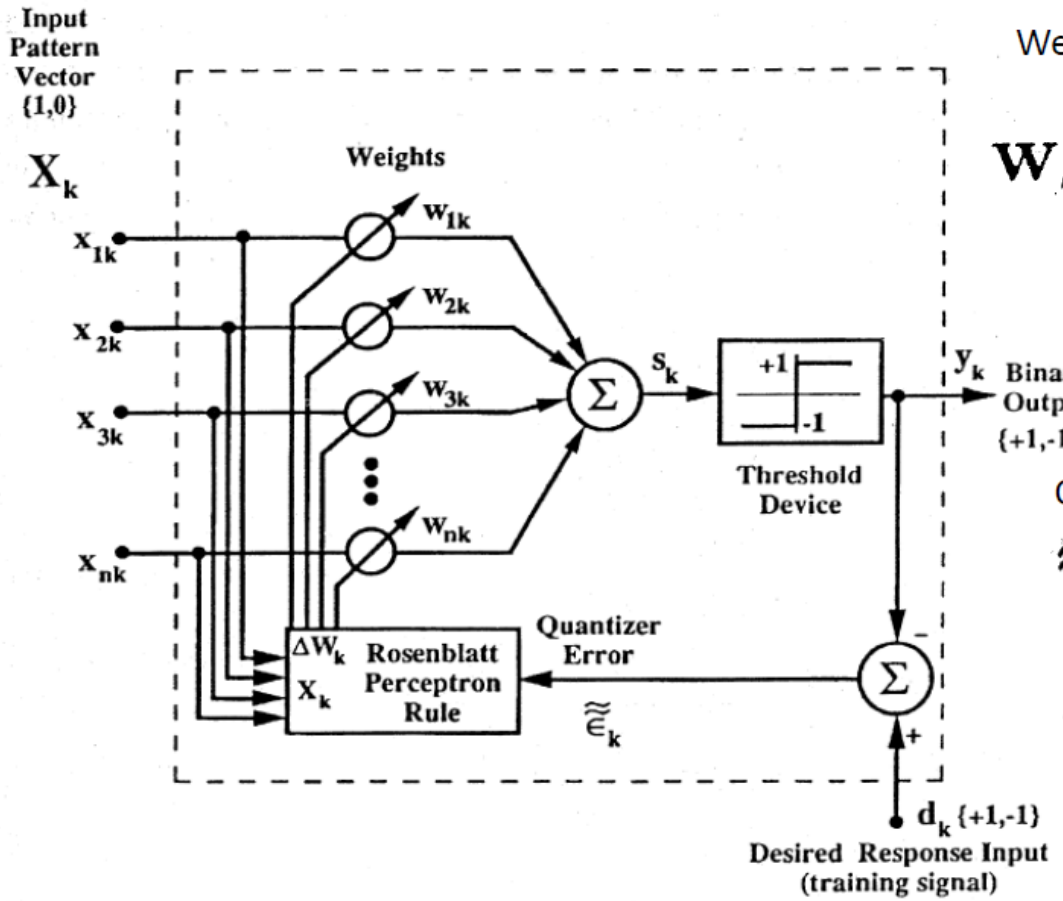
The Perceptron

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–407. (Reprinted in *Neurocomputing* (MIT Press, 1988).).

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan, New York.

- a connectionist model of computation in the brain
- input stimuli changes the connections/weights in a network
- contrast connectionist model with coded memory model → one -to-one mapping between stimuli and stored pattern
- more interested in ensemble/network behavior than individual
- relies on probability theory rather than symbolic logic (von Neumann, 1956 paper) to explain reliable emergent behavior when nodes are unreliable or exact connections are unknown

The Perceptron



Weight update:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha \frac{\approx\epsilon_k}{2} \mathbf{X}_k$$

Quantizer error:

$$\approx\epsilon_k \triangleq d_k - y_k.$$

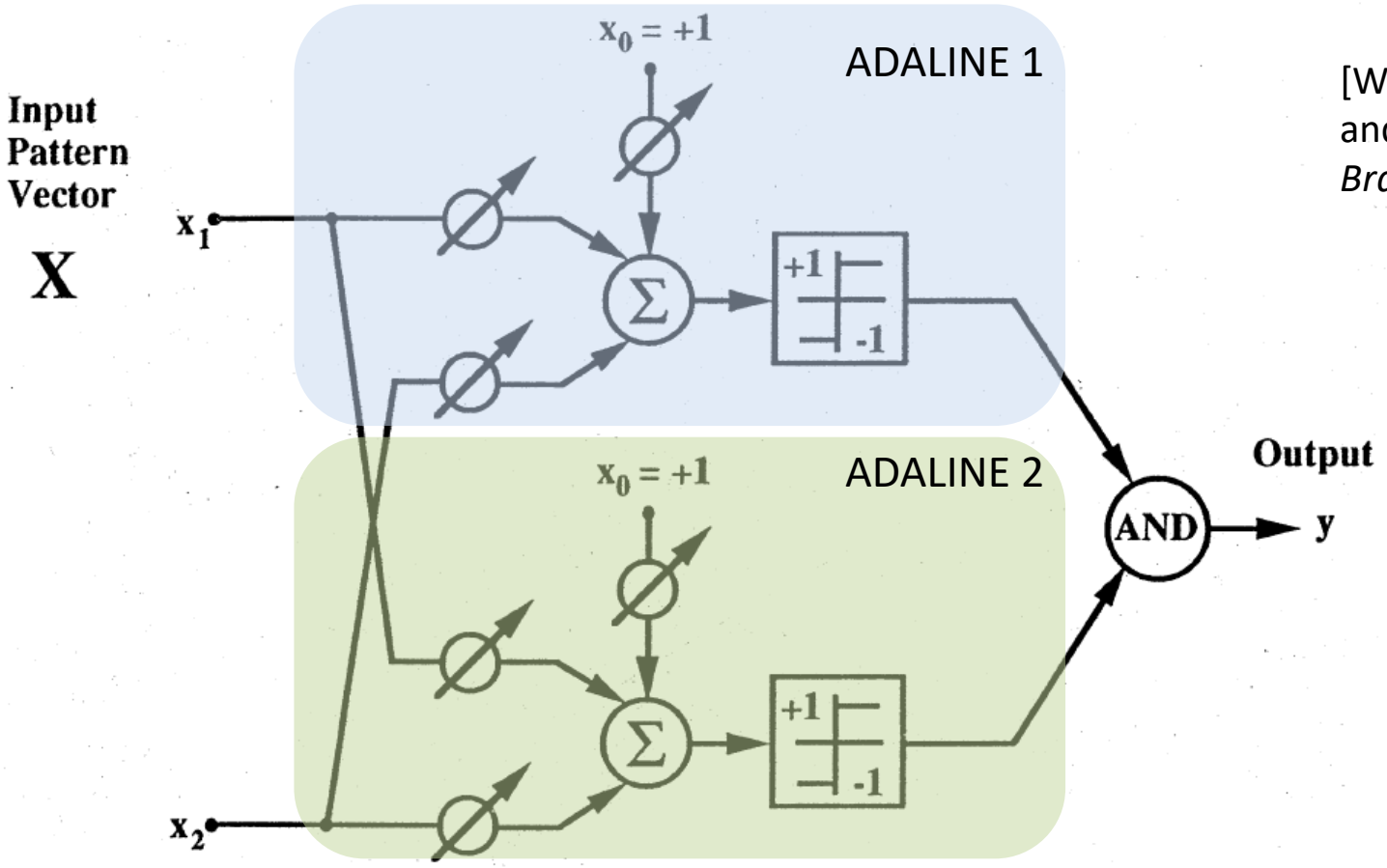
- network response is binary (unlike ADALINE)
- Can derive from SGD using the approximation $\frac{\partial y_k}{\partial s_k} = 1$, i.e., the slicer is a pass-through buffer.

- weights get modified only if there is an error ($y_k = -d_k$):

$$\mathbf{W}_{k+1} = \mathbf{W}_k + d_k \mathbf{X}_k$$

- input vector is added or subtracted from current weight vector if error occurs
- for a separable training set - perceptron learning rule will find a solution in finite number of steps (apply repeatedly until all samples are correctly classified) → many solutions exist
- smaller the margin, more number of steps
- algorithm does not converge when data are non-separable

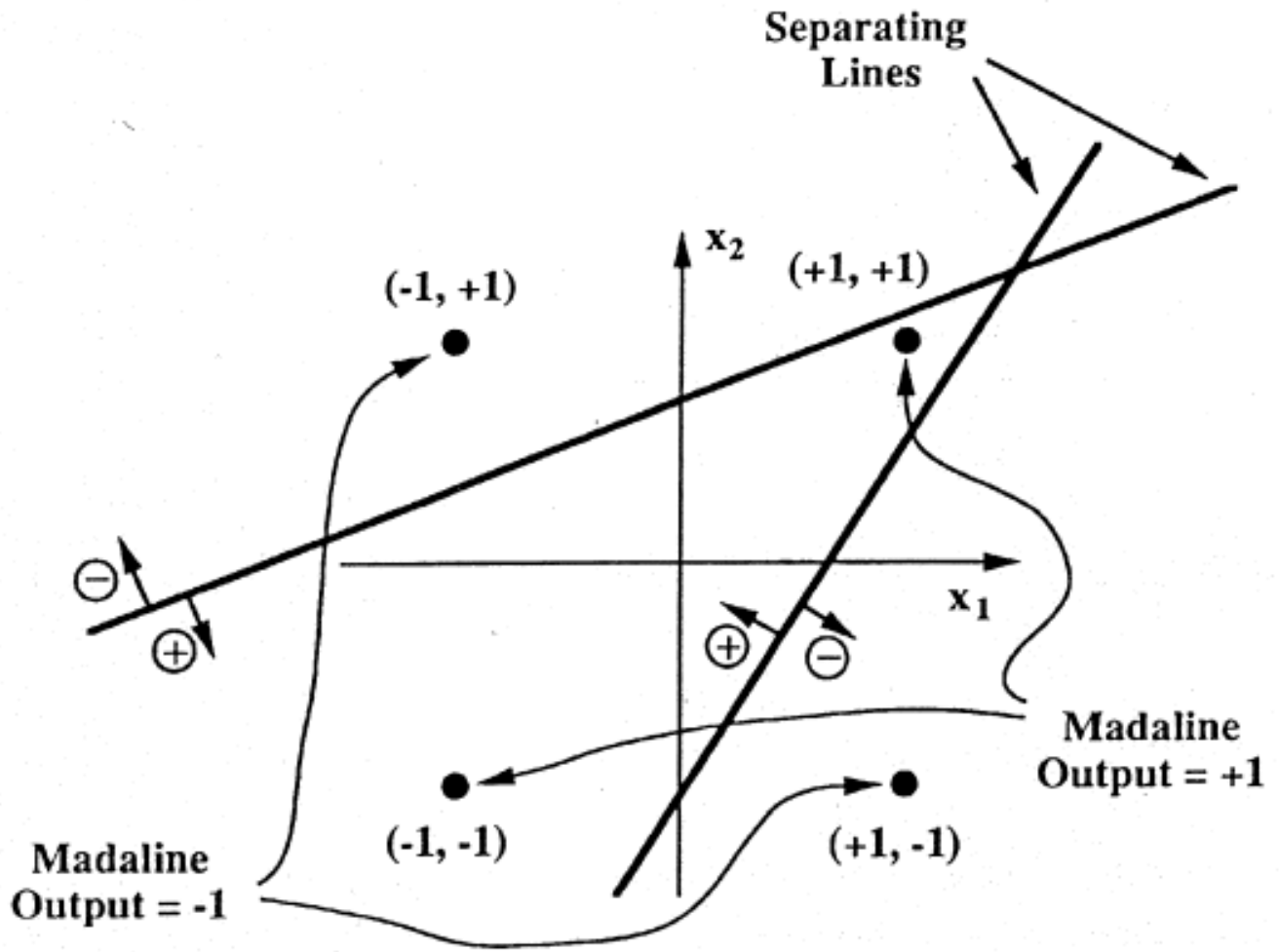
MADALINE



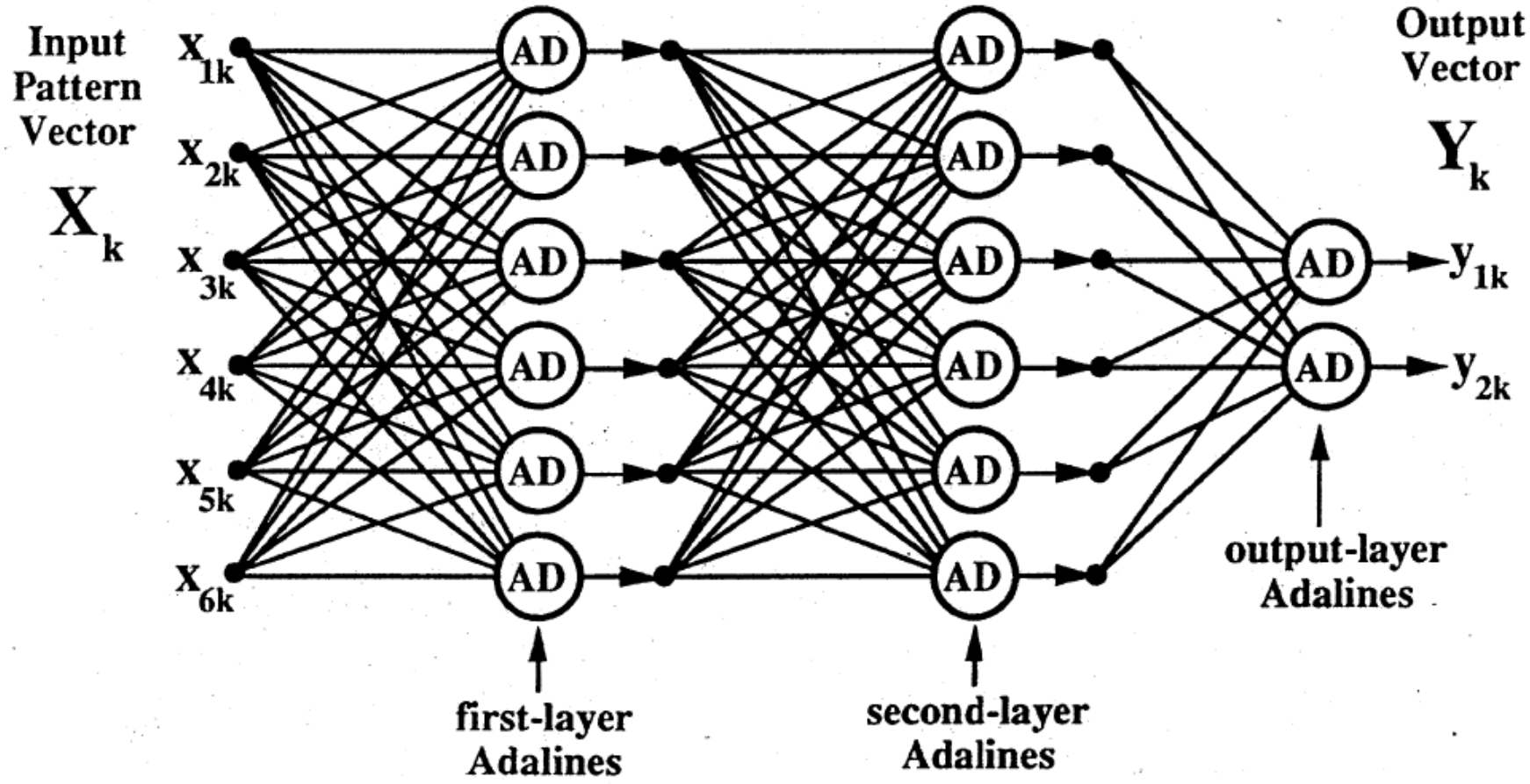
[Widrow & Lehr, Perceptron, Adalines and Back Prop., *The Handbook for Brain Theory and Neural Networks*]

- can realize all 16 Boolean functions of two binary variables

MADALINE Decision Boundary



MADALINE – 3 Layers



- *fully connected*: can train the output layer; need back propagation to train the hidden layers; precursor to deep neural networks

Hopfield Networks

Proc. Natl. Acad. Sci. USA
Vol. 79, pp. 2554–2558, April 1982
Biophysics

Neural networks and physical systems with emergent collective computational abilities

(associative memory/parallel processing/categorization/content-addressable memory/fail-soft devices)

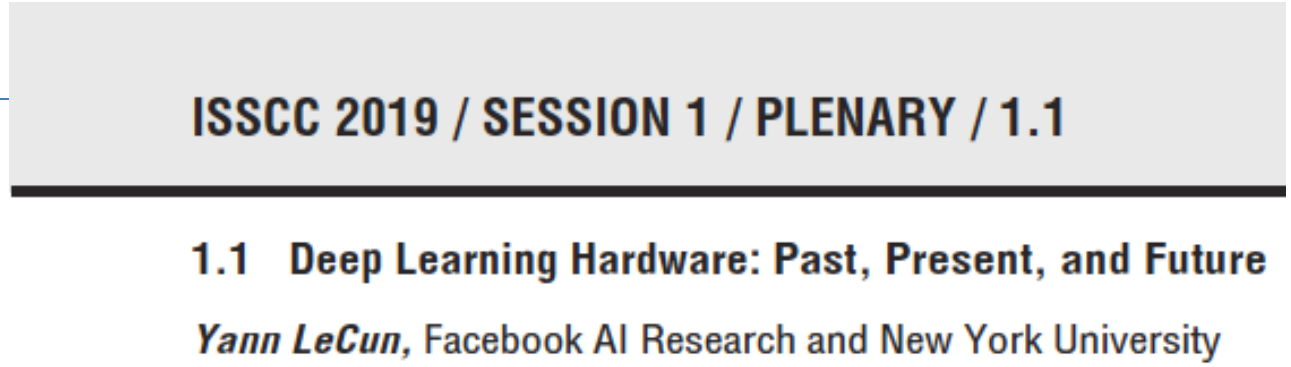
J. J. HOPFIELD

Division of Chemistry and Biology, California Institute of Technology, Pasadena, California 91125; and Bell Laboratories, Murray Hill, New Jersey 07974

- can computation be viewed as a "spontaneous collective consequence of having a large number of interacting simple neurons"?
- also uses McCullogh & Pitts (1943) model of neuron

Four Simple Ideas Underlying DL

- Origins in the 1950's


ISSCC 2019 / SESSION 1 / PLENARY / 1.1

1.1 Deep Learning Hardware: Past, Present, and Future

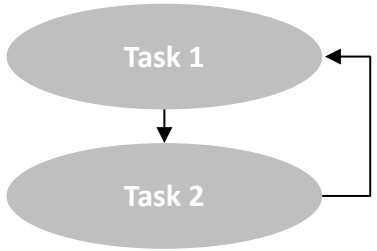
Yann LeCun, Facebook AI Research and New York University

1) compose complex functions from simple parametrized functional blocks
2) learn desired functions from examples (data) by adjusting the parameters
3) learning is an optimization procedure – minimize an objective function via gradient-based methods
4) compute gradient efficiently via back-propagation

DL in Hardware

DL in Hardware

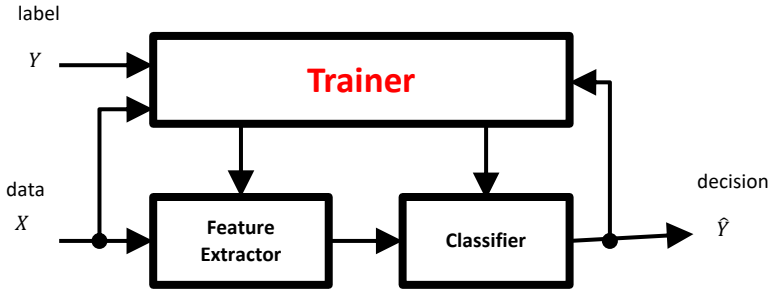
Applications



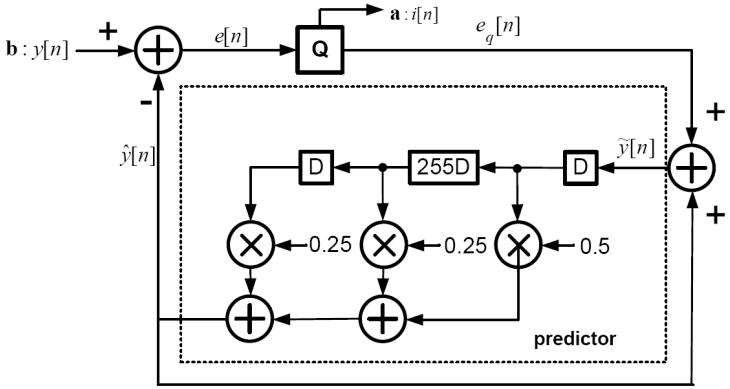
natural vision, EEG analysis, navigation, surveillance

Inference Tasks

Learning Models

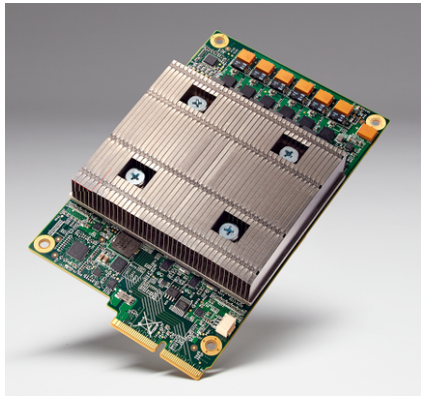


Finite-precision Architecture

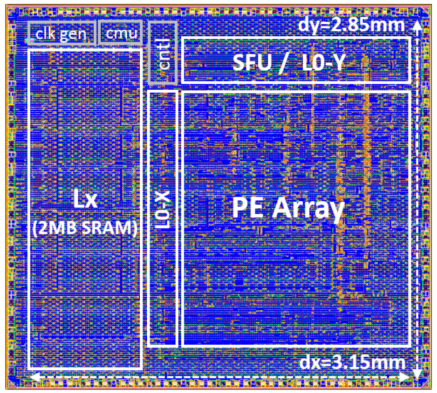


Hardware

module

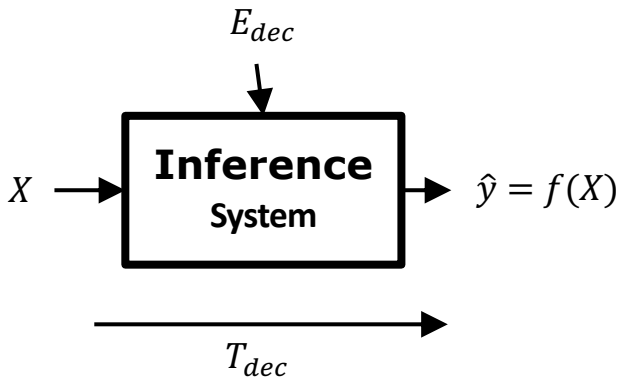


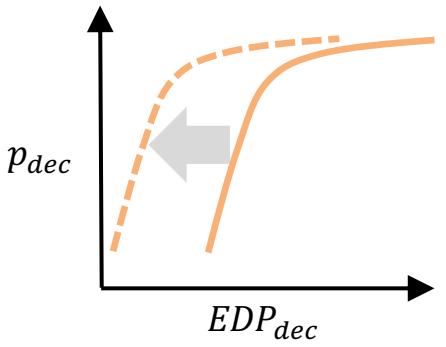
integrated circuit

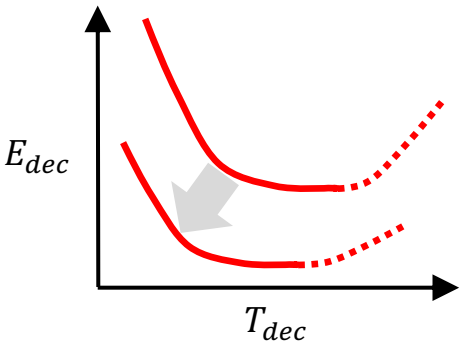


System Metrics for Inference

system-level energy vs. latency vs. accuracy trade-off







- E_{dec}: energy per decision
- T_{dec}: decision latency
- p_{dec}: decision accuracy
- $EDP_{dec} = E_{dec} \times T_{dec}$: decision EDP

- $E_{dec} \propto 1/T_{dec} \propto$ data & network size
- $p_{dec} \propto EDP_{dec}$

The Energy Cost of Intelligence

The Economist | World politics | Business & finance | Economics | Science & technology | Culture

Artificial intelligence and Go

The Economist, March 2016

A game-changing result

AlphaGo's masters taught it the game, but an electrifying match shows what the computer may have to teach humans

Mar 19th 2016 | From the print edition

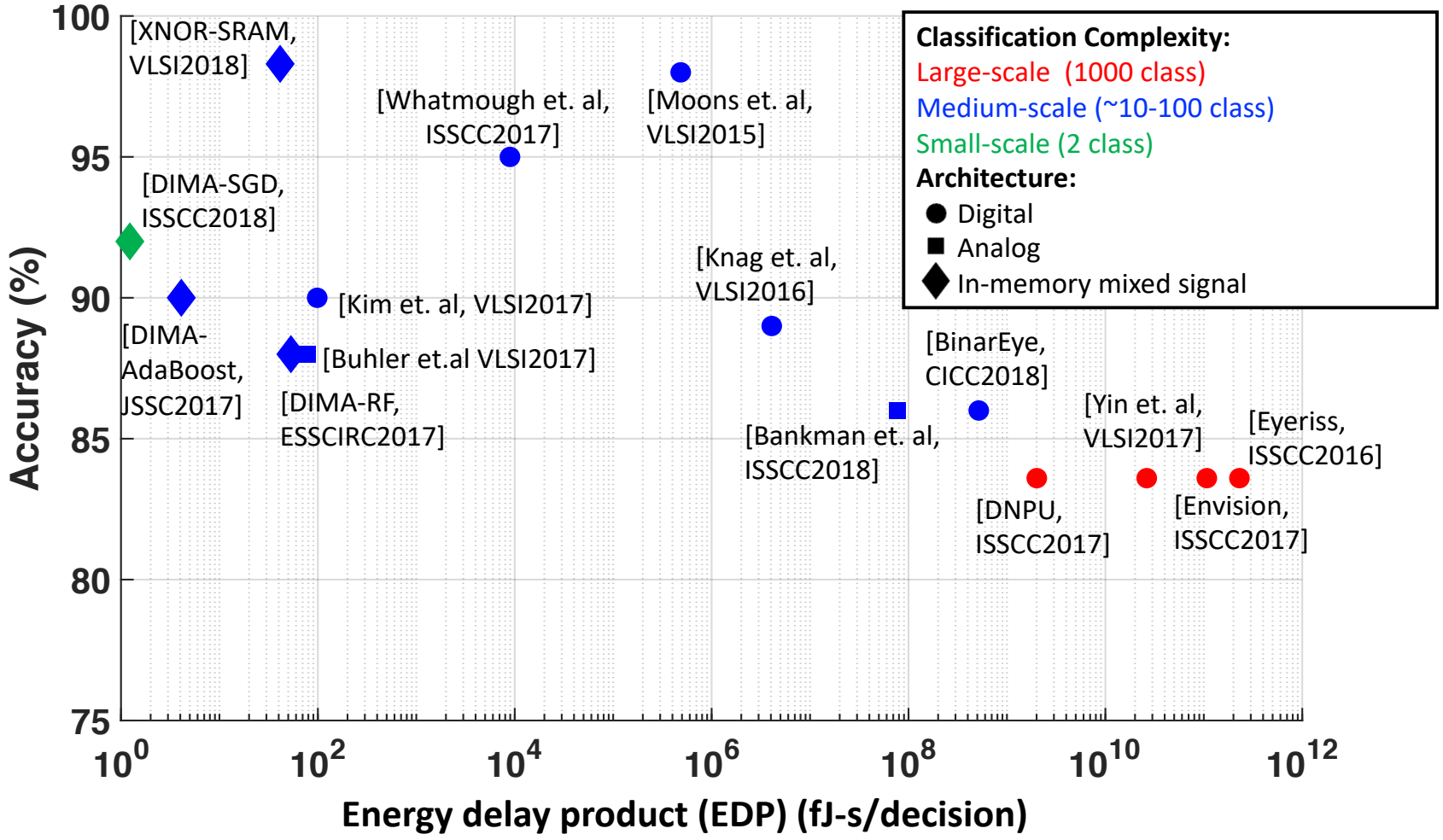
Timekeeper | Like 2.9K | Tweet

IT WAS not quite a whitewash, but it was close. When DeepMind, a London-based artificial intelligence (AI) company bought by Google for $400m in 2014, challenged Lee Sedol to a five-game Go match, Mr Lee—one of the best human players of that ancient and notoriously taxing board game—confidently predicted that he would win 5-0, or maybe 4-1.

He was right about the score, but wrong about the winner. The match, played in Seoul to

- machines can beat humans in specific cognitive tasks
- game of Go is complex → huge search space: $\sim 250^{150}$ (Go) vs. $\sim 35^{80}$ (Chess)
- AlphaGo machine: 1202 CPUs+176 GPUs
- HUGE Energy Cost $\sim 10{,}000\times$ more than human brain

System (Decision) Level –Accuracy vs. EDP

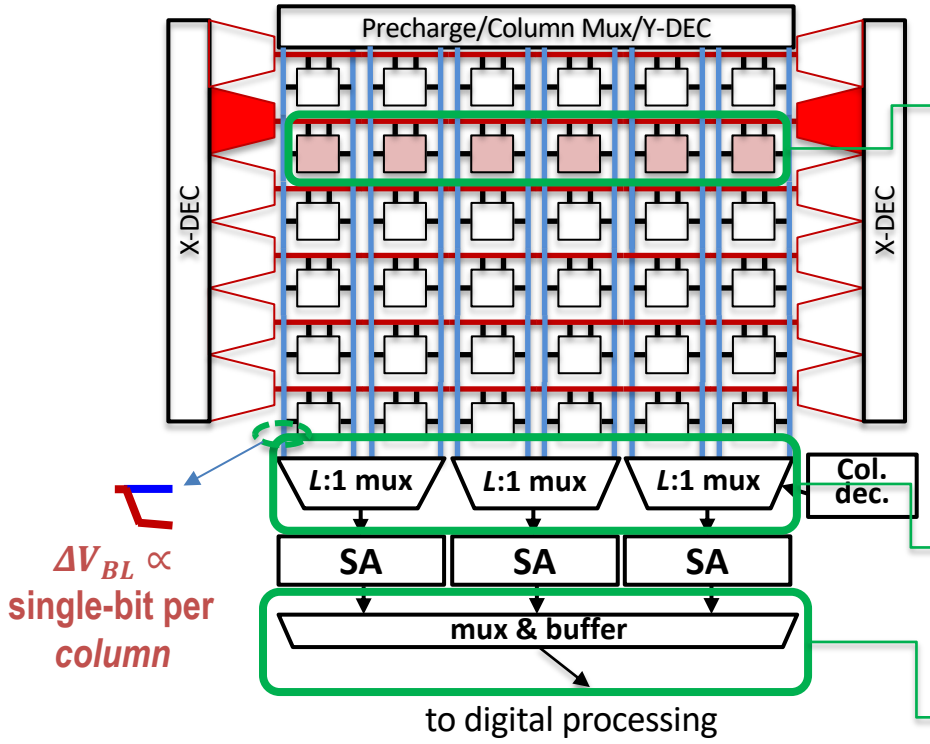


All EDP numbers scaled for 65nm node

Fundamental Question

How do we design **intelligent machines** that **operate at the limits** of **energy-delay-accuracy**?

Von Neumann (Digital) Architecture



single row read per access
energy inefficient, throughput bottleneck

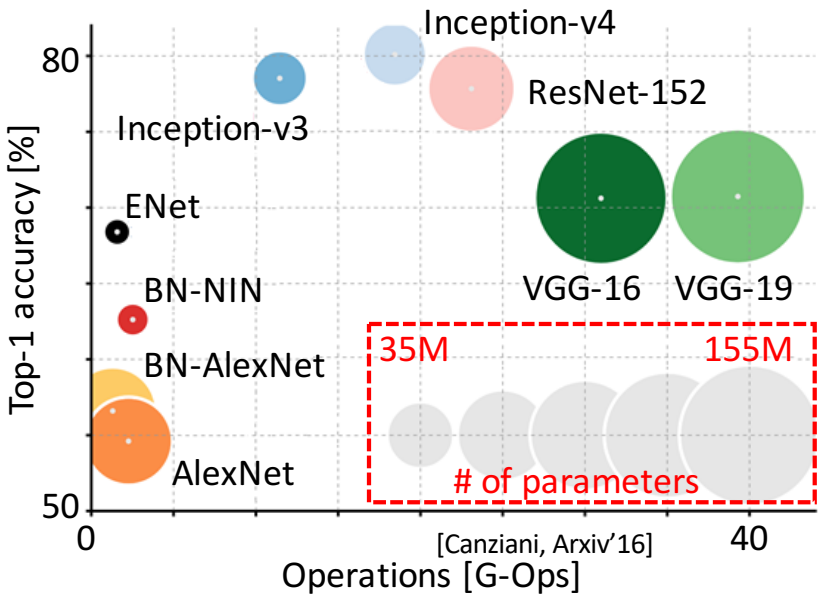
The Memory Wall

column muxing
degrades throughput

large I/O bandwidth requirement
large latency, energy consumption

The Energy Cost of Data Movement

$$\frac{E_{mem}}{E_{mac}} \approx \sim 100\times \text{ (SRAM)} \rightarrow \sim 500\times \text{ (DRAM)} \rightarrow \sim 1000\times \text{ (Flash)}$$



Memory Energy (45nm)

Memory	
Cache	(64bit)
8KB	10pJ
32KB	20pJ
1MB	100pJ
DRAM	1.3-2.6nJ

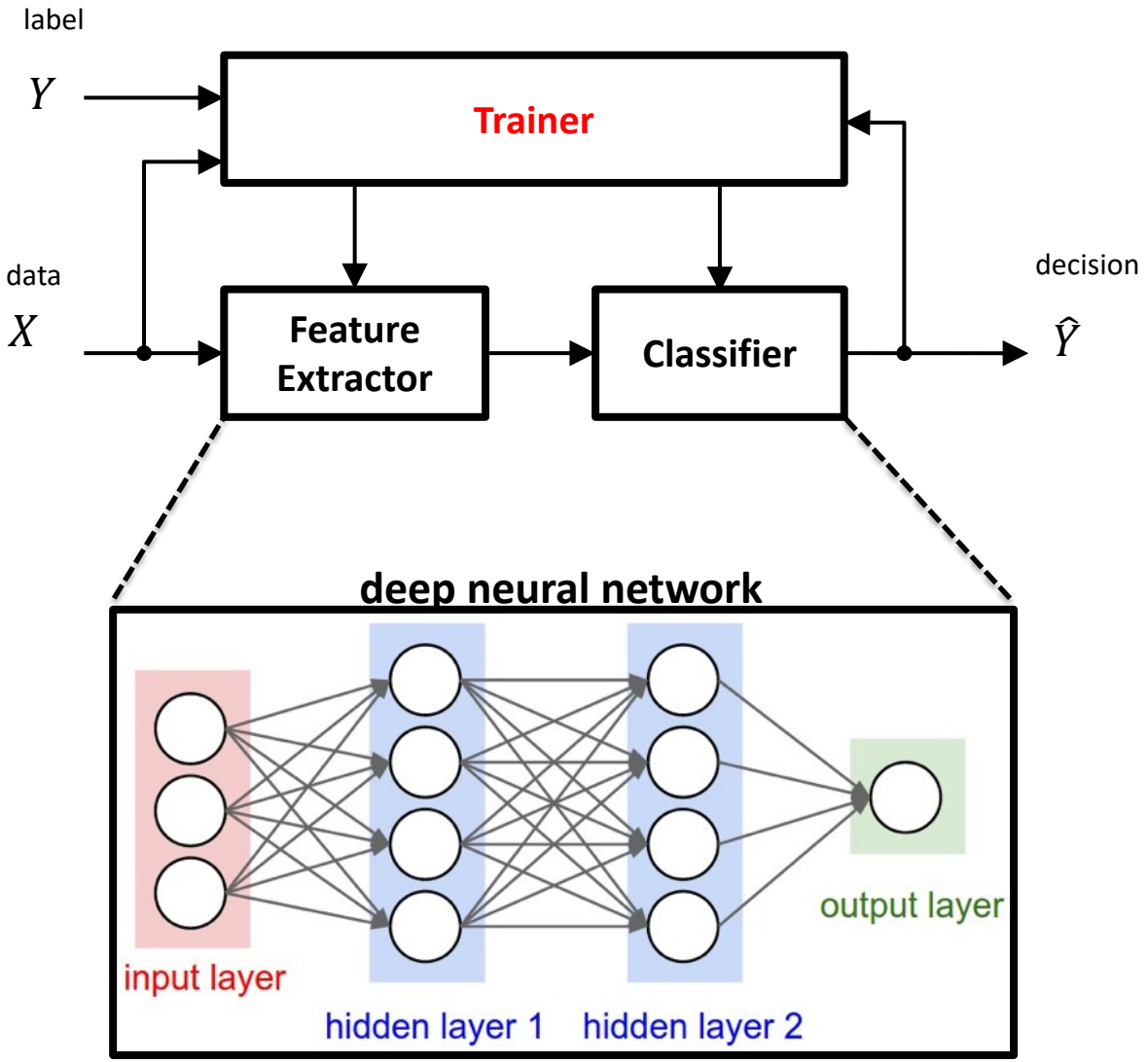
Computation Energy (45nm)

Integer	
Add	
8 bit	0.03pJ
32 bit	0.1pJ
Mult	
8 bit	0.2pJ
32 bit	3 pJ

FP	
FAdd	
16 bit	0.4pJ
32 bit	0.9pJ
FMult	
16 bit	1pJ
32 bit	4pJ

[*Horowitz, ISSCC'14*]

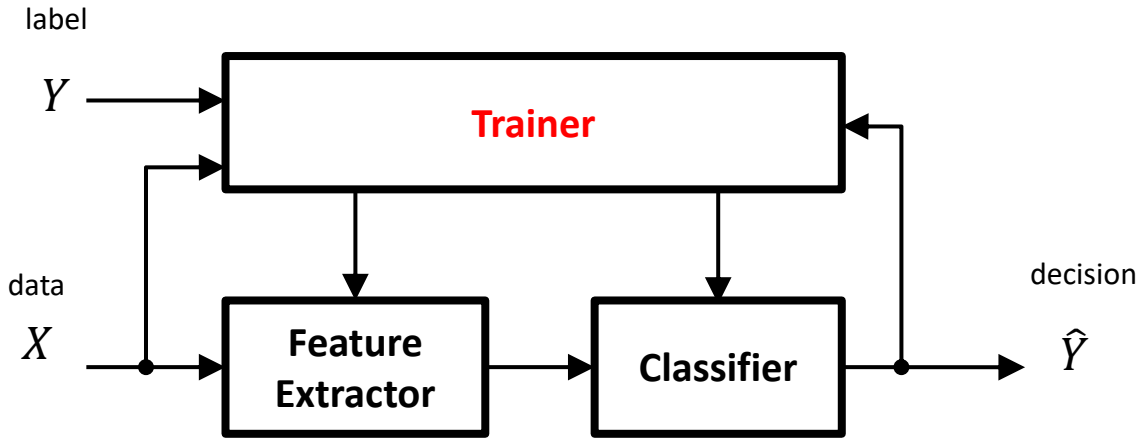
A Deep Learning System



(training phase)

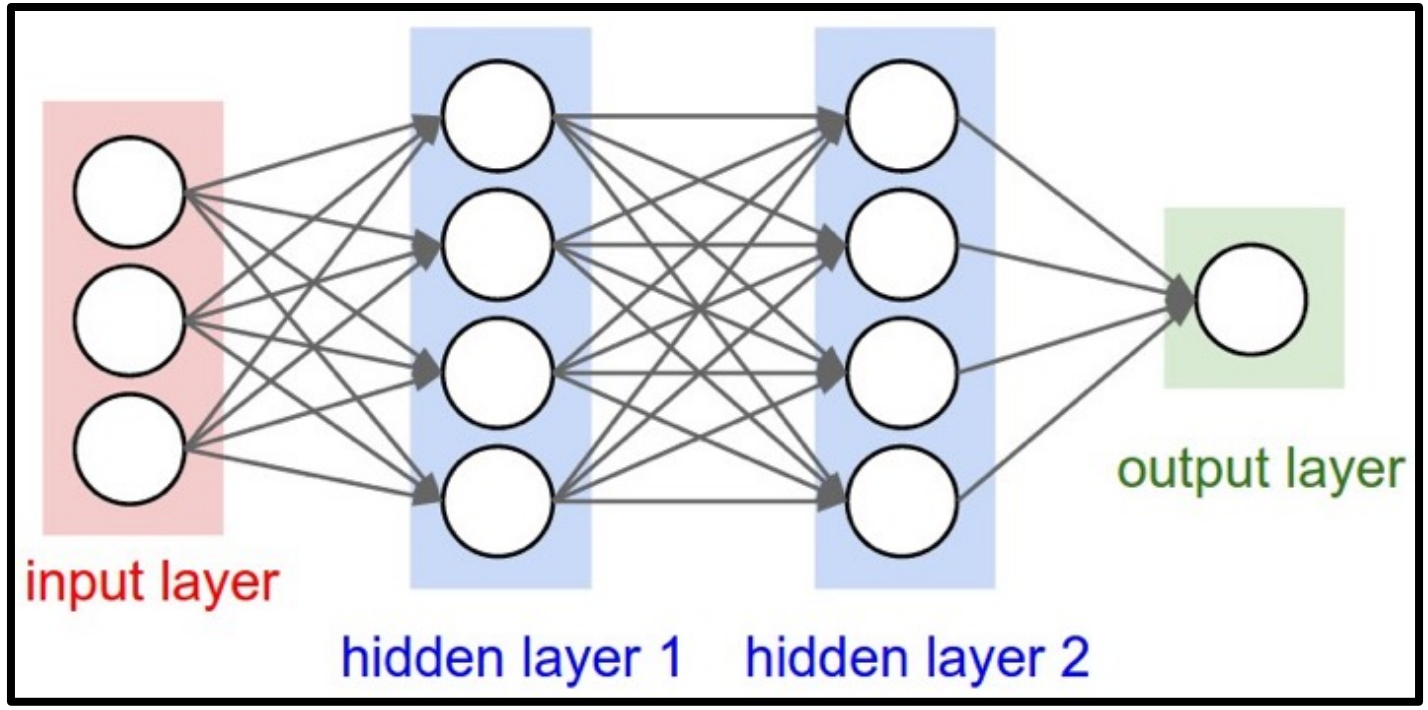
(test phase)

A Classifier Architecture



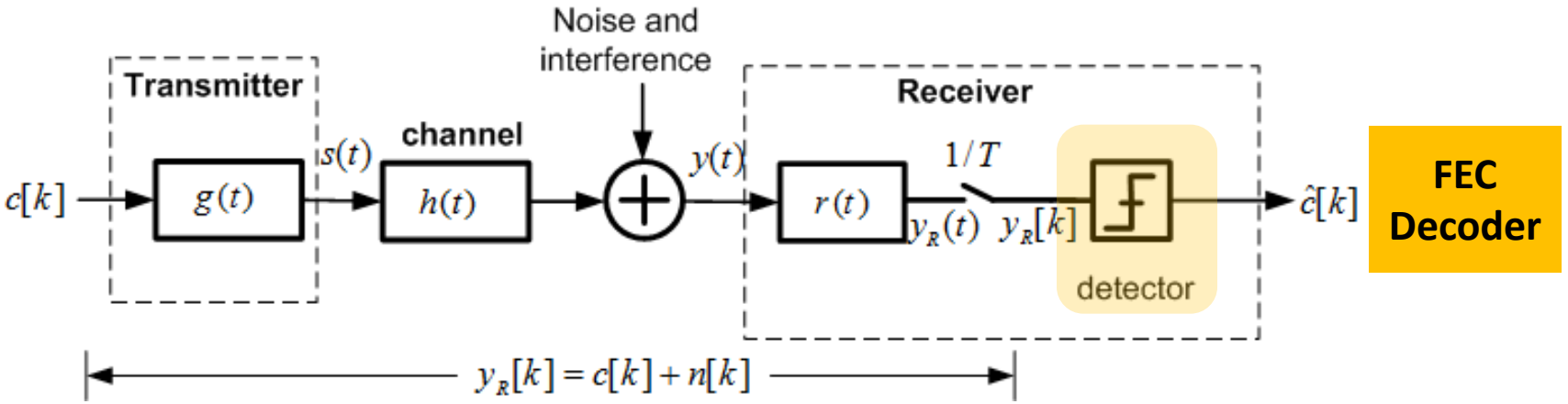
misclassification rate (accuracy):
$\Pr\{\hat{Y} \neq Y\}$

- a system for supervised learning for classification task
- key components:
  - data set: (X, Y)
  - feature extractor (FE): preprocesses X to improve accuracy & speed-up classification
  - classifier makes decisions
  - trainer: learns parameters of FE and classifier to maximize accuracy

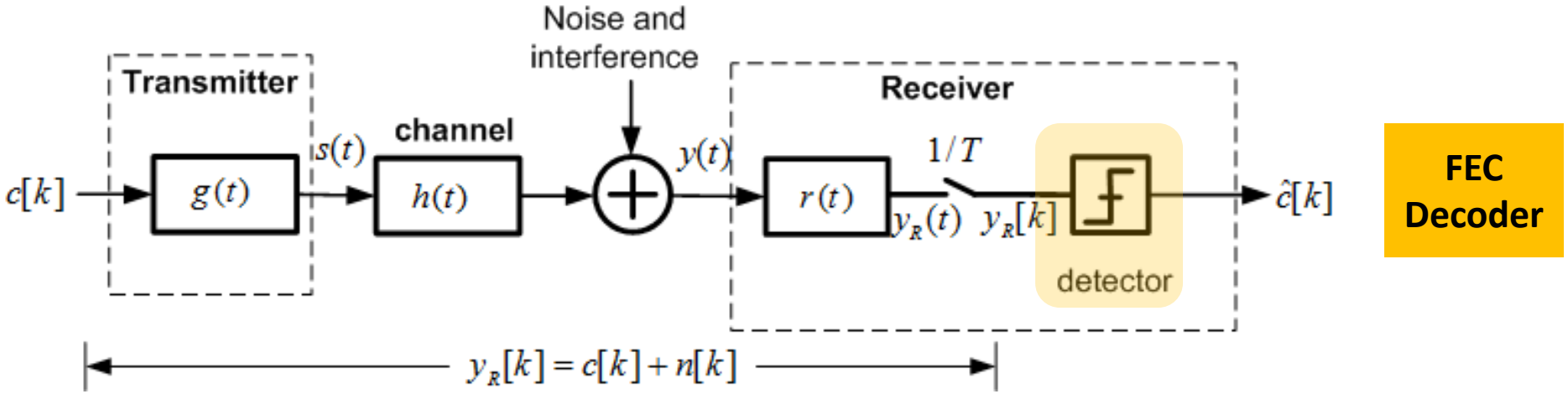


dominated by matrix-vector multiply (MVM) computations

Classifiers in Communication Links



- Classifiers used in communications links are referred to as **detectors**.
- A forward error-control (FEC) decoder is a detector that bases its decision on a block of channel symbols $\hat{c}[k]$
- A slicer is a detector that bases its decisions on a single observation $y_R[k]$.



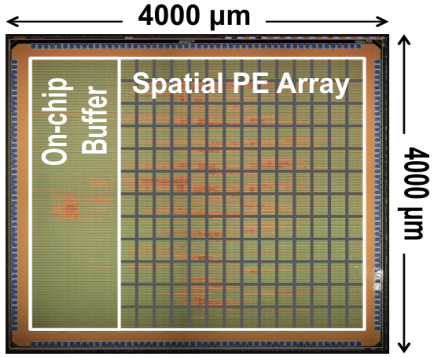
- transmitter imposed structure on models of the input data (signal) and noise, e.g. additive noise model:

$$y_R[k] = c[k] + n[k]$$

- imposed structure on data makes the classifier computationally efficient and accurate → key distinction between machine learning and communications

Machine Learning in Finite Precision

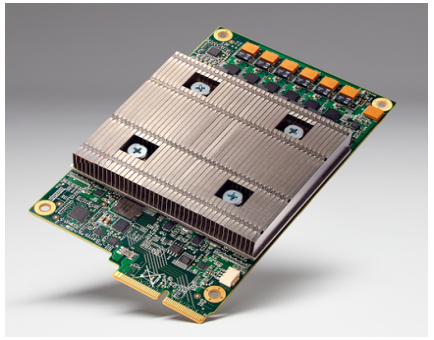
MIT's Eyeriss



[ISSCC'16]

16b fixed-point
(inference)

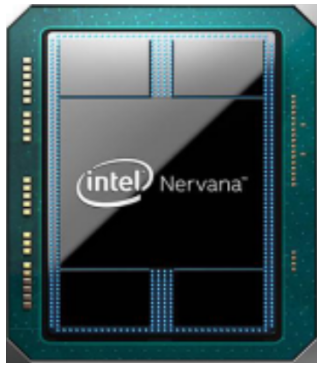
Google's TPU



[ISCA'17]

8b fixed-point
(inference)
16b floating-point
(training)

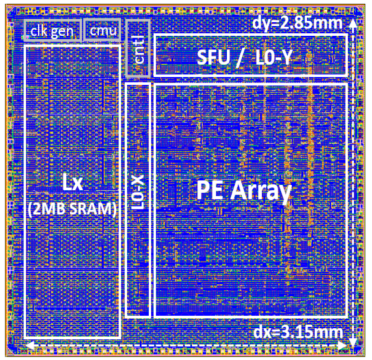
Intel's NNP



[NIPS'17]

16b flexpoint
(training)

IBM's AI Core



[VLSI'18]

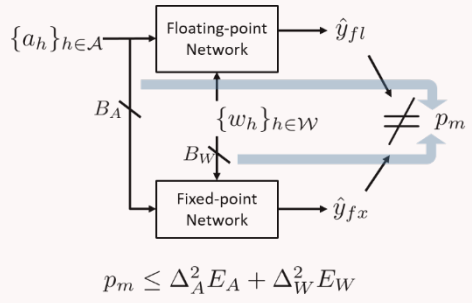
16b floating-point
(training)

Are these the minimum precisions required?

How to find minimum precision requirements of DNNs?

Machine Learning with Minimum Precision

Fixed-point inference with theoretical guarantees

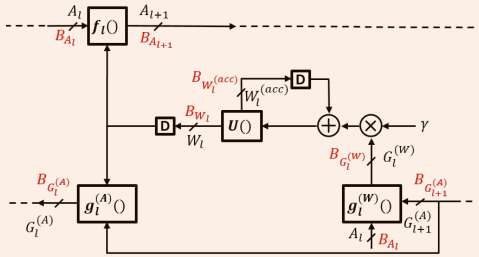


$$p_m \leq \Delta_A^2 E_A + \Delta_W^2 E_W$$

Sakr, Kim, Shanbhag **ICML 2017**

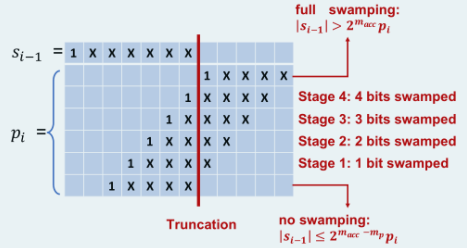
Sakr & Shanbhag **ICASSP 2018**

Fixed-point training with close-to-minimal precisions



Sakr & Shanbhag **ICLR 2019**

Fl.-pt. training with accumulation bit-width scaling

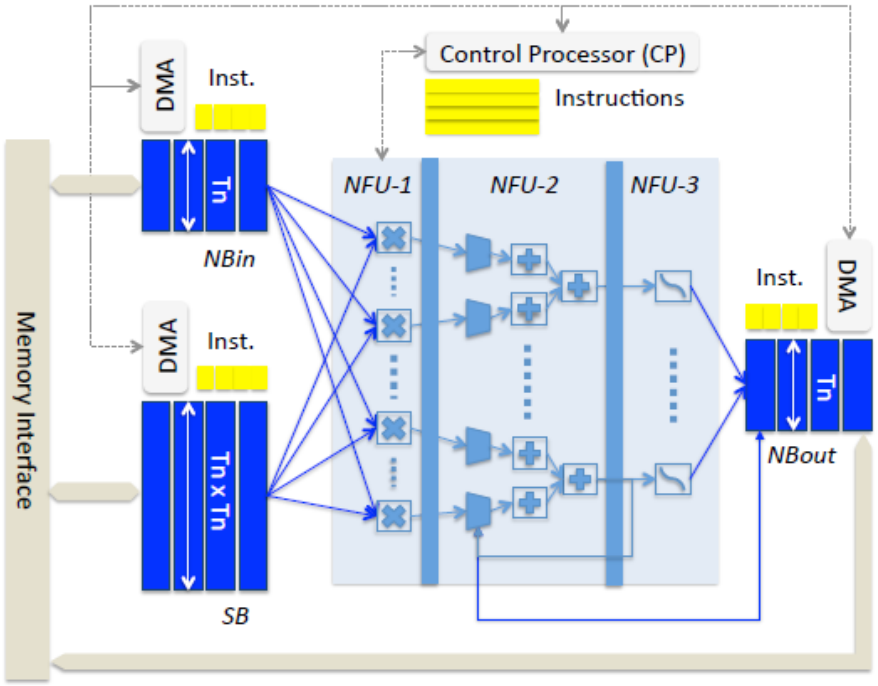


Sakr & Shanbhag (*with IBM*)

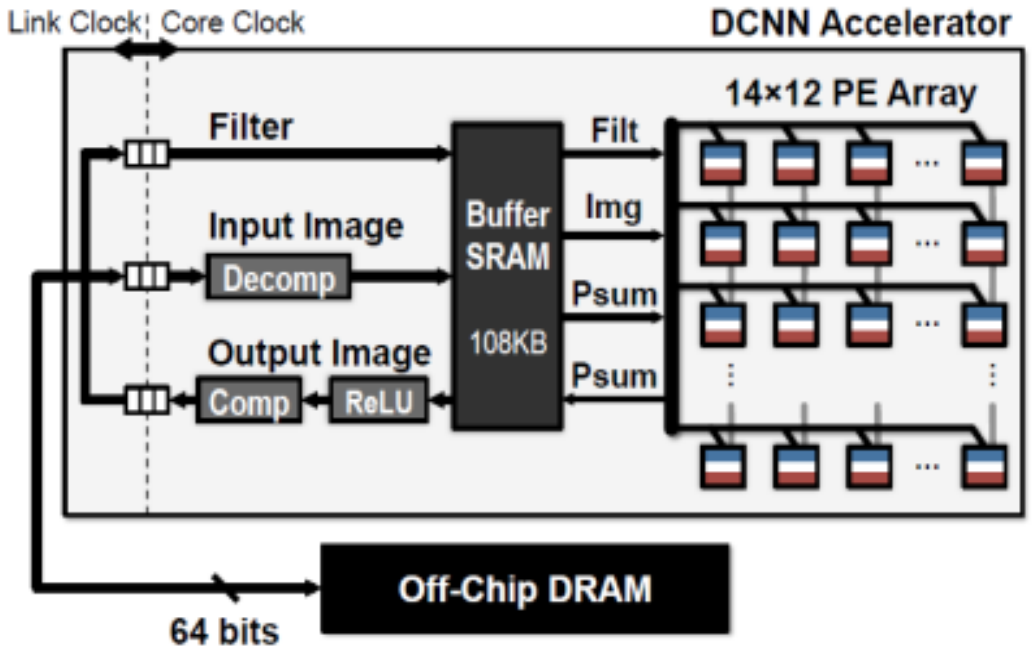
ICLR 2019

Digital DL Accerators

DianNao Family

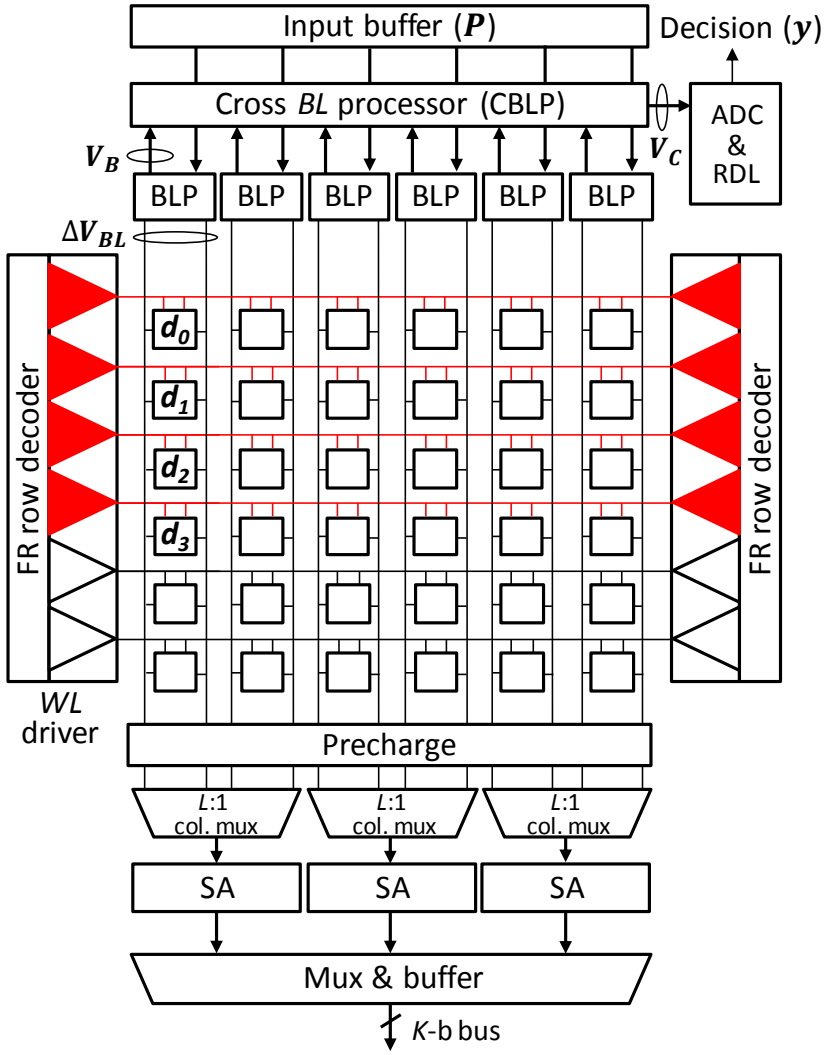


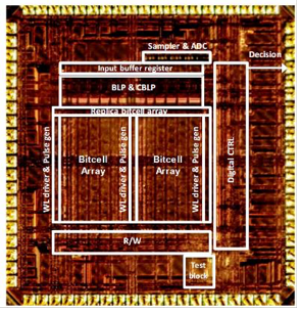
Eyeriss



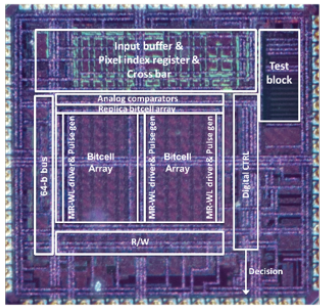
- exploiting data and weight reuse opportunities
- explore energy-latency-accuracy trade-offs

In-memory DL/ML Architectures

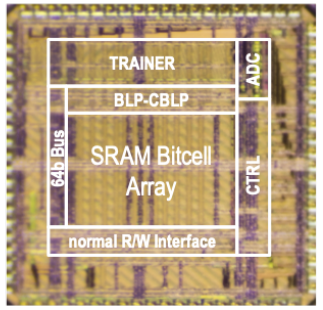




[JSSC'18]



[ESSCIRC'17 JSSC'18]

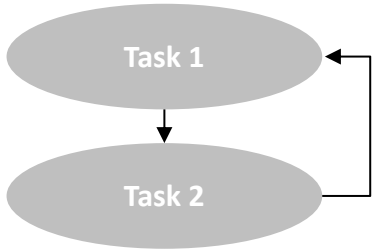


[ISSCC'18, JSSC'18]

- embed analog computations into memory
- maximize row and column parallelism
- trade-off energy-latency-compute SNR
- > 100X gains in lab prototypes

DL in Hardware

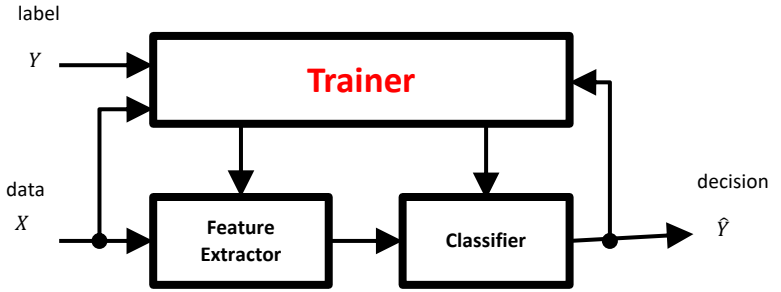
Applications



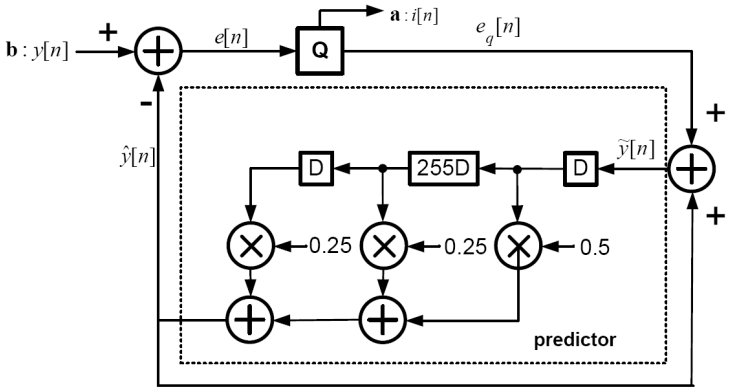
natural vision, EEG analysis, navigation, surveillance

Inference Tasks

Learning Models

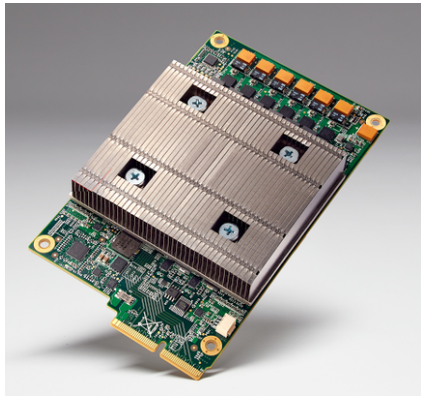


Finite-precision Architecture

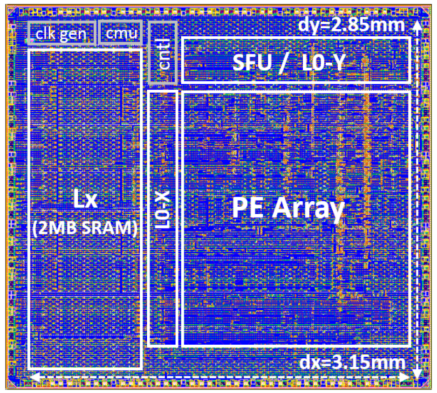


Hardware

module

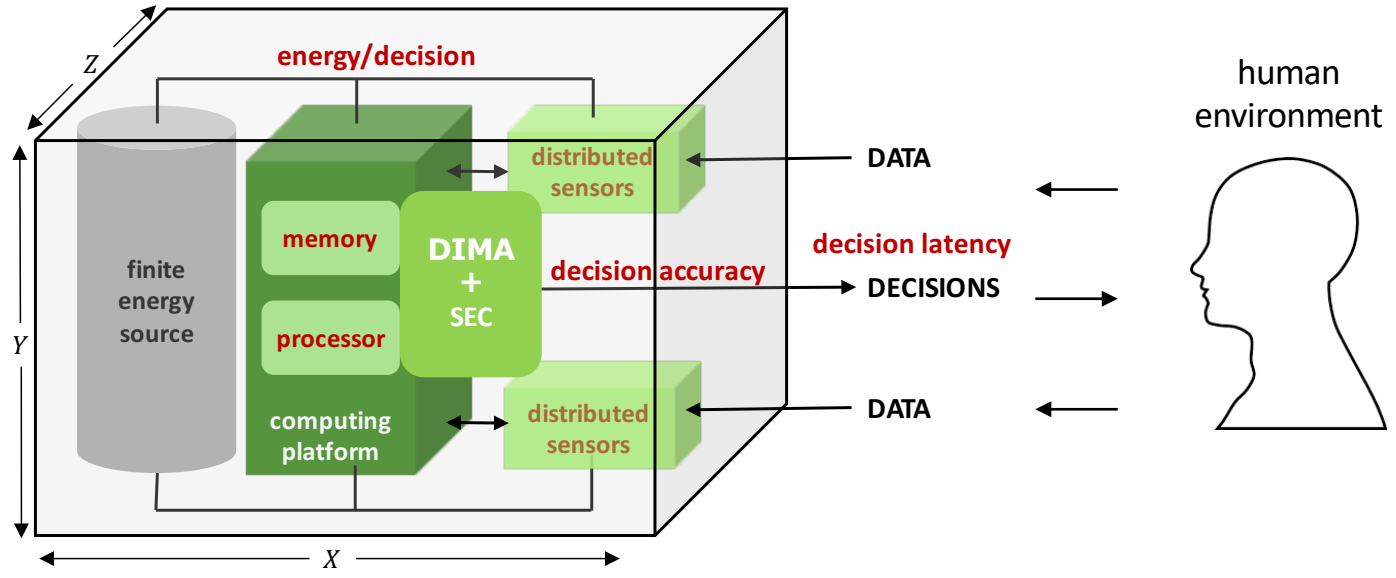


integrated circuit



The Future – Human-Centric AI

Human-centric Cognitive Agent



fully autonomous functionality via **in-situ** information processing of **rich multimodal** sensory inputs under
severely constrained volume, energy & computational resources

Some Thoughts in Closing

- Today's question: **what can machines do**?
- the real question: **what kind of machines should we build**? **What kind of society do we want to live in**?
- needs writers, artists and engineers to **imagine the future** together!.....then build it.

Course Web Page

https://courses.grainger.illinois.edu/ece598nsg/fa2020/
https://courses.grainger.illinois.edu/ece498nsu/fa2020/

http://shanbhag.ece.uiuc.edu