FROM RESEARCH TO INDUSTRY

cea tech

ARTIFICIAL INTELLIGENCE: PAST, PRESENT AND FUTURE

Marc Duranton | Commissariat à l’énergie atomique et aux énergies alternatives|

QUOTES….

"As soon as it works, no one calls it AI anymore"

John McCarthy

“The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.”

Edsger W. Dijkstra

THE TWO PARTS OF THE BRAIN

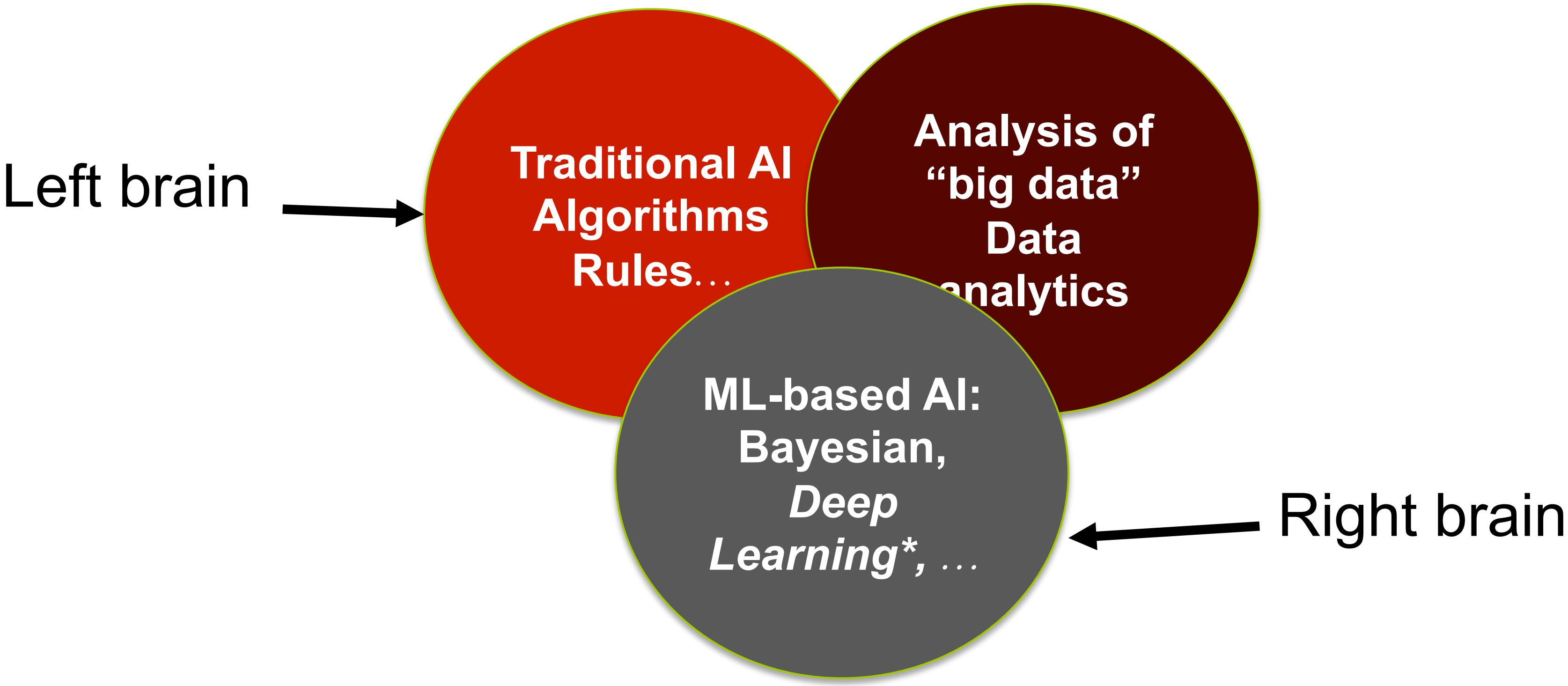
Left brain

Formalism
Rules
Abstract world

Right brain

Uncertainty
Ambiguity
Real world

KEY ELEMENTS OF ARTIFICIAL INTELLIGENCE



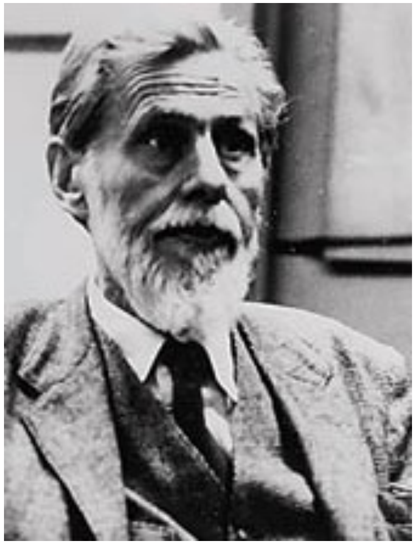
*** Reinforcement Learning, One-shot Learning, Generative Adversarial Networks, etc…**

From Greg. S. Corrado, Google brain team co-founder:

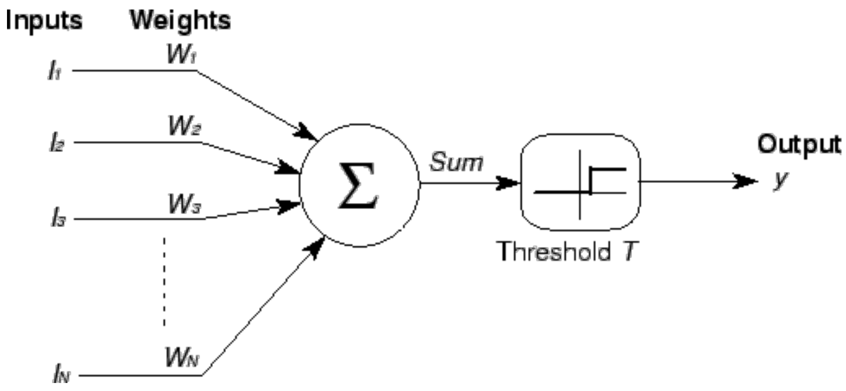
– *“Traditional AI systems are* ***programmed*** *to be clever*

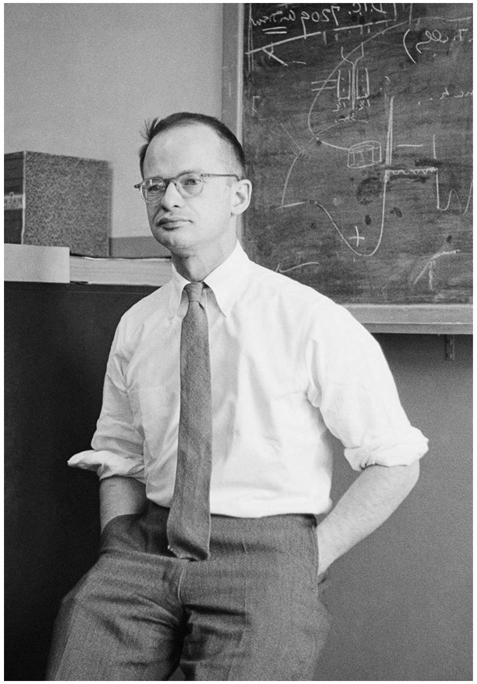
– *Modern ML-based AI systems* ***learn*** *to be clever.*

1943: MCCULLOCH AND PITTS



Neurophysiologist and cybernetician





Logician workingin the field of computational neuroscience

They laid the foundations of formal Neural Networks

1943: MCCULLOCH AND PITTS

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

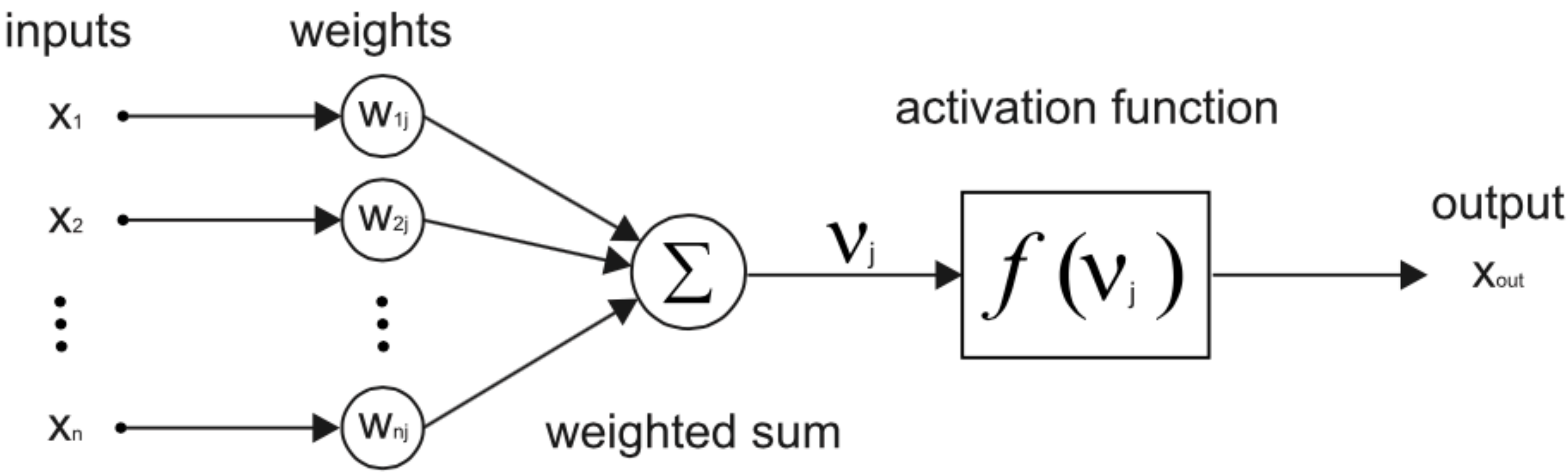
WARREN S. MCCULLOCH AND WALTER PITTS

FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

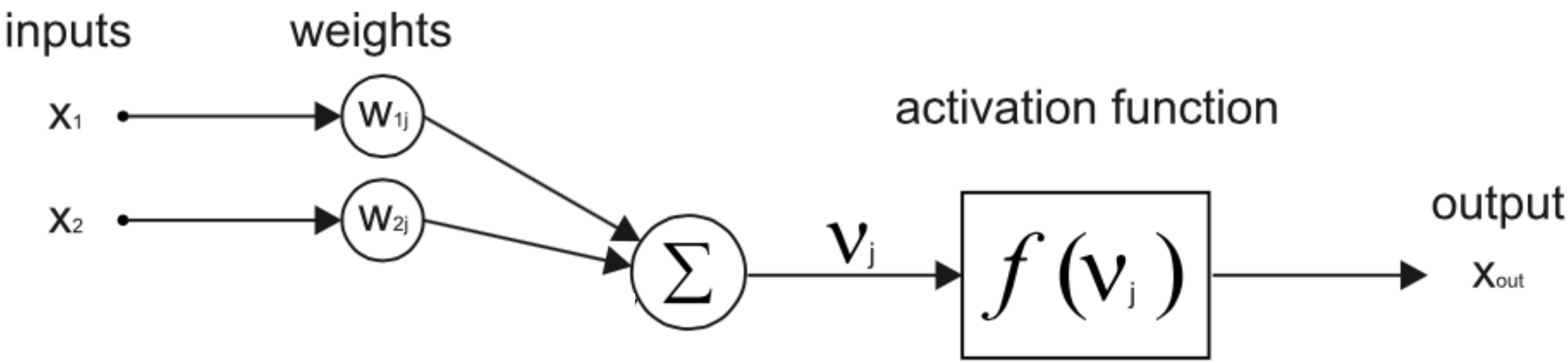
WHAT IS A NEURAL NETWORK?

A « formal » neuron:



WHAT IS A NEURAL NETWORK?

The « formal » neuron:



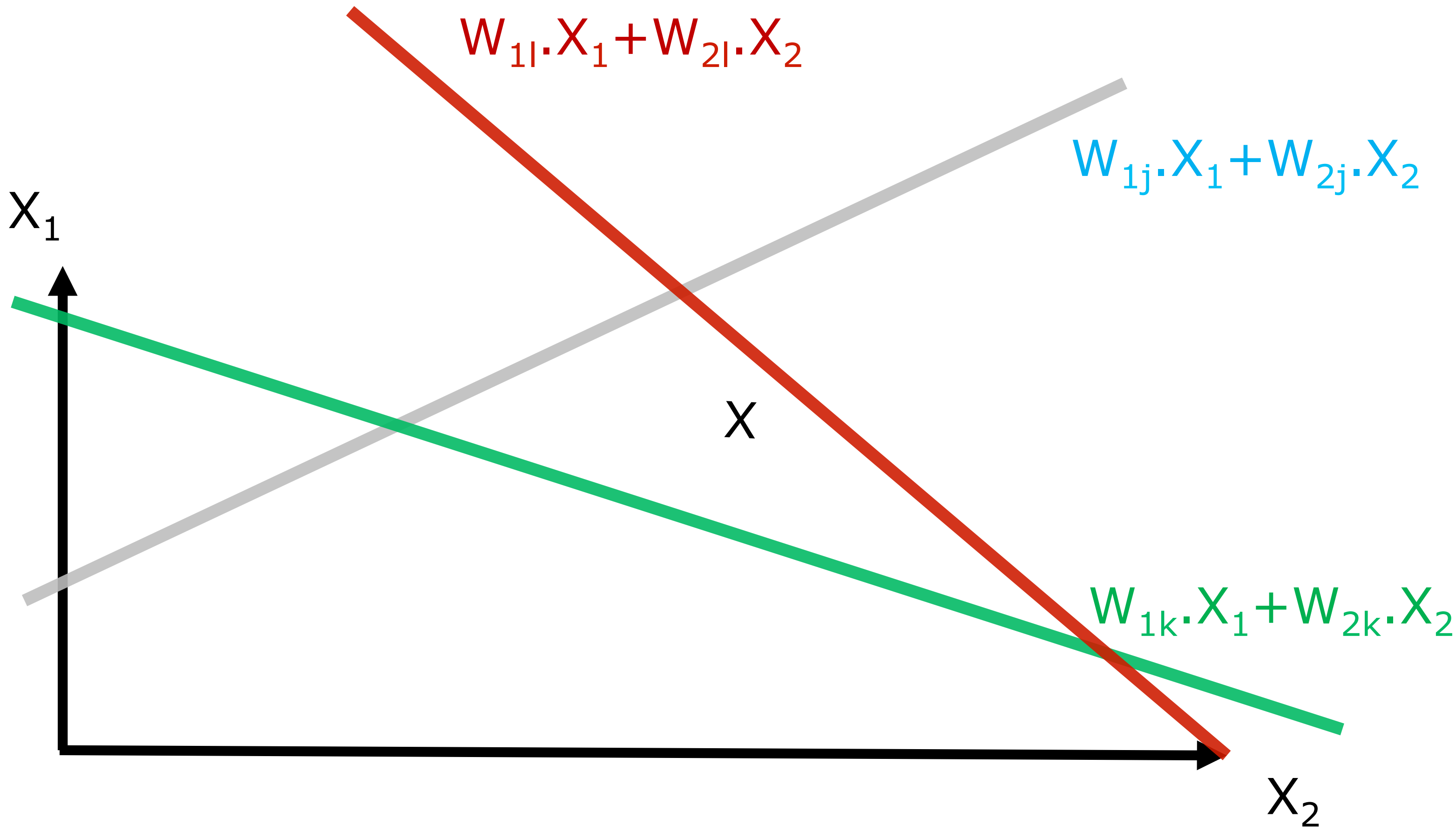
$V_j = W_{1j}.X_1 + W_{2j}.X_2$

It is the definition of an hyperplane

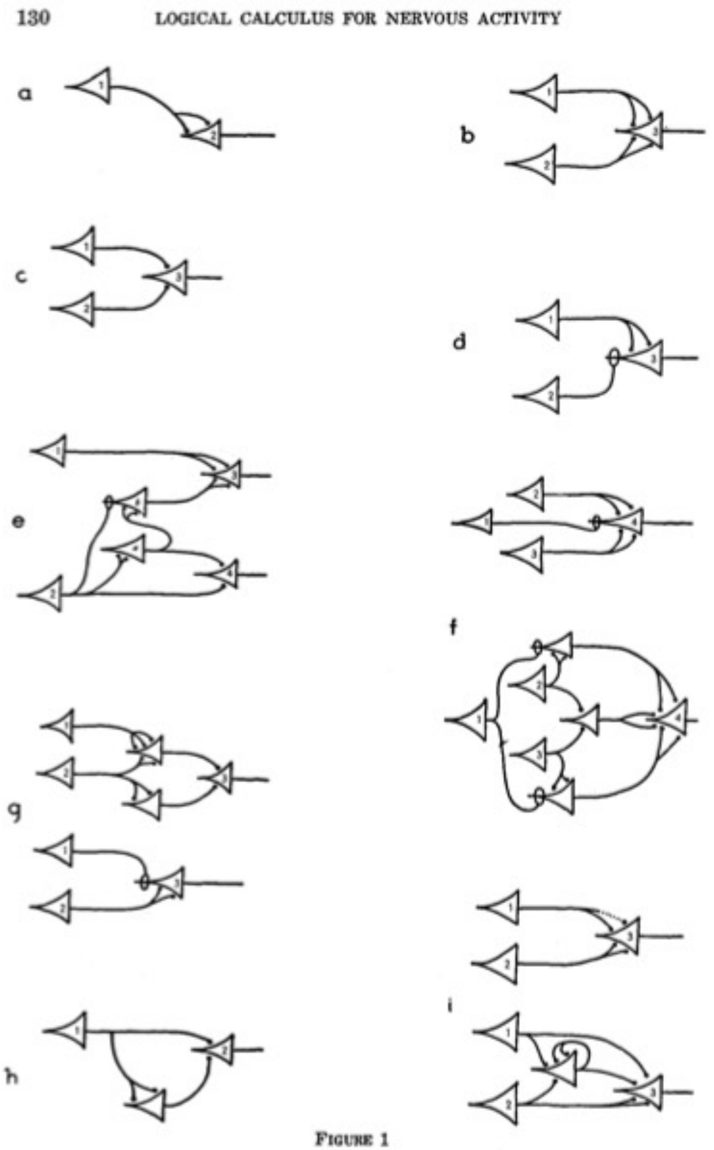
$F(V_j)$ non linear $\in \{-1,1\}$ e.g. sign() function

$X(X_1,X_2)$ is "above" or "below" the hyperplane

WHAT IS A NEURAL NETWORK?



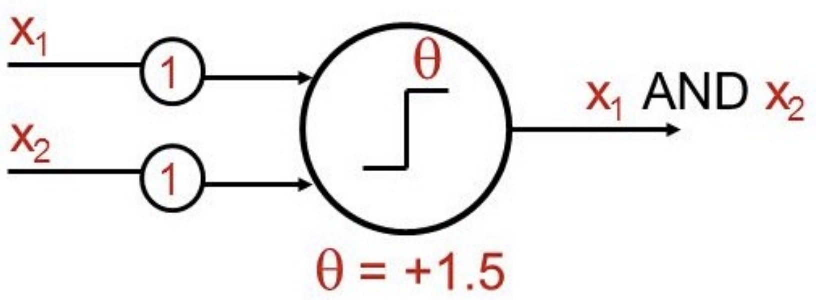
WHAT IS A NEURAL NETWORK?



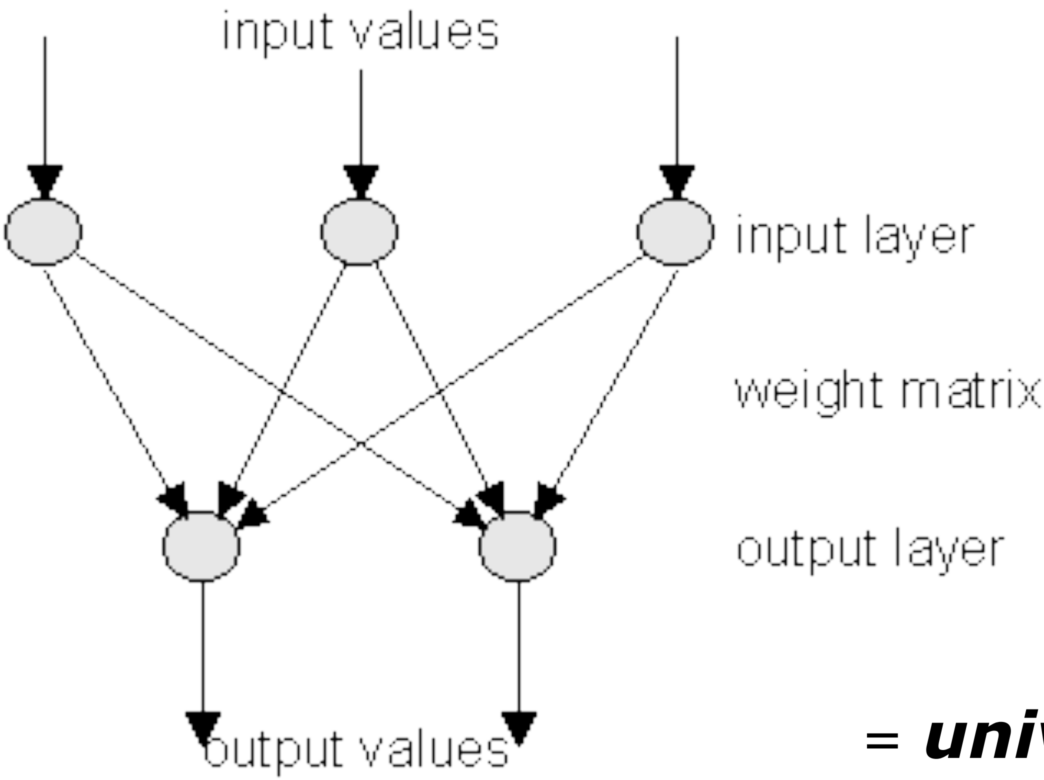
Association of neurons to make logical functions.

Example: AND gate

IN 1	IN 2	OUT
0	0	0
0	1	0
1	0	0
1	1	1



MULTILAYER NETWORK



Hyperplane separation

"logic" composition
Warren McCulloch and
Walter Pitts, 1943

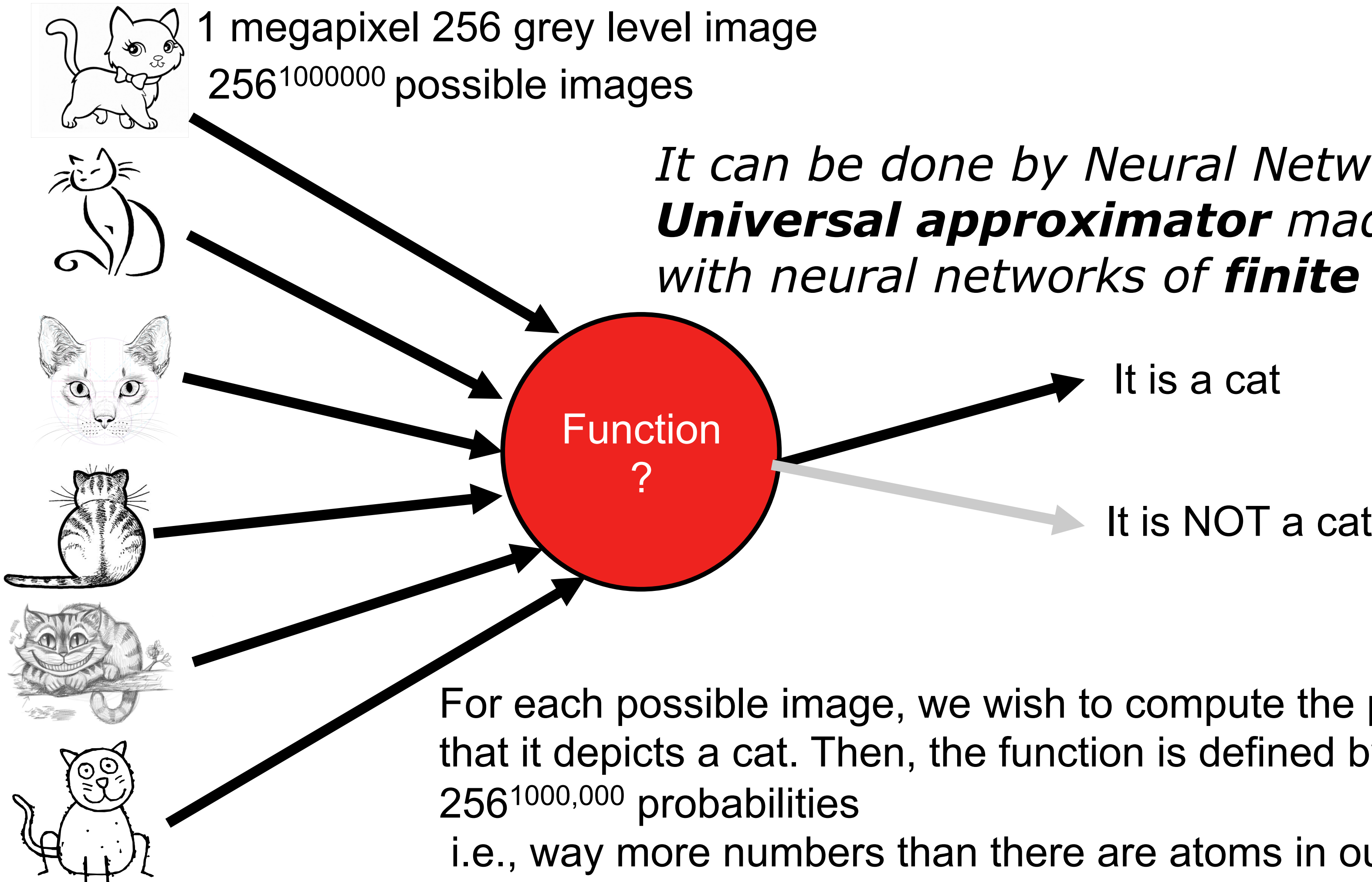
= ***universal approximator***

WHY DOES DEEP LEARNING WORK SO WELL?*

1 megapixel 256 grey level image

$256^{1000000}$ possible images

It can be done by Neural Networks:
Universal approximator *made*
with neural networks of ***finite*** *size*



For each possible image, we wish to compute the probability that it depicts a cat. Then, the function is defined by a list of $256^{1000,000}$ probabilities
i.e., way more numbers than there are atoms in our universe (about 10^{78} to 10^{82} <<< $10^{2,408,240}$).

- Work of Henry W. Lin (Harward) , Max Tegmark (MIT), and David Rolnick (MIT)
https://arxiv.org/abs/1608.08225

WHY DOES DEEP LEARNING WORK SO WELL?*

But a picture of a cat is not an arbitrary set of random pixels:

“For reasons that are still not fully understood, our universe can be accurately described by **polynomial Hamiltonians of low order**,”

The laws of physics have other important properties. For example, they are usually **symmetrical** when it comes to **rotation and translation**.

There is another property of the universe that neural networks exploit. This is the **hierarchy of its structure**.

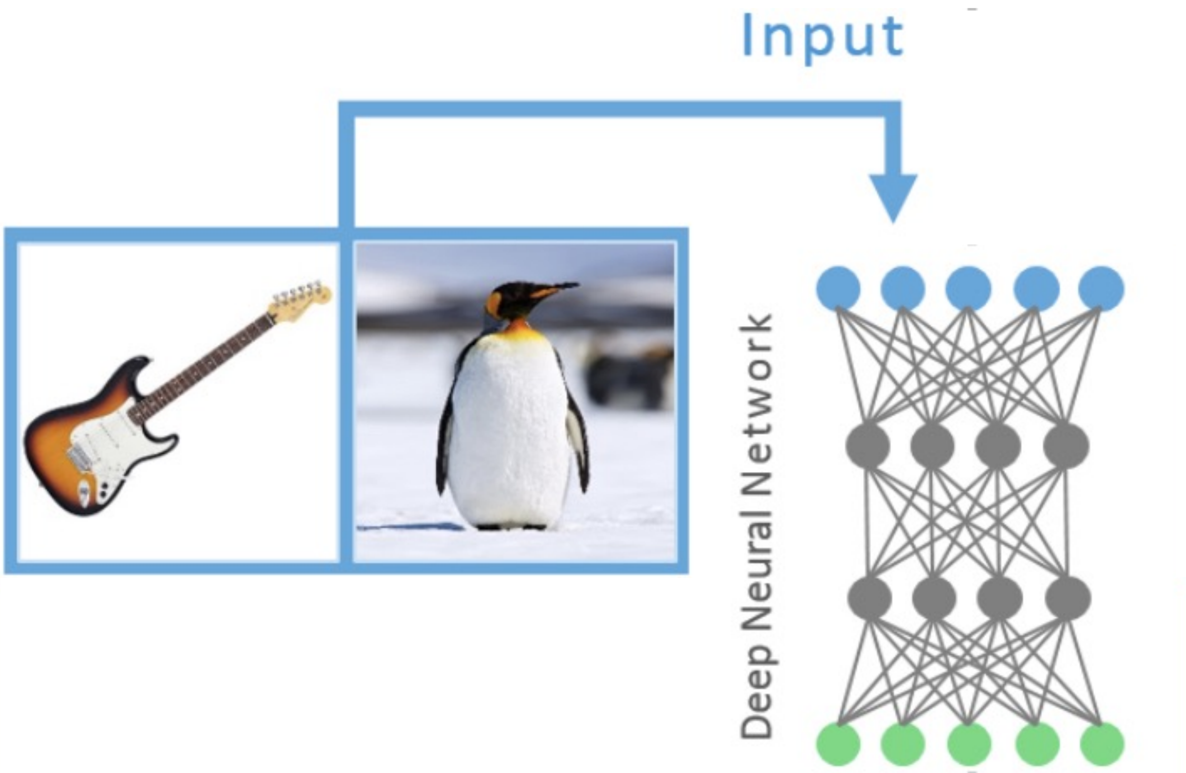
This is why the structure of neural networks is important too: the layers in these networks can approximate each step in the causal sequence.

- **These properties mean that neural networks do not need to approximate an infinitude of possible mathematical functions but only a tiny subset of the simplest ones.** – ***because they are inspired from biological systems that were developed in the context of the real world.***

* Work of Henry W. Lin (Harward) , Max Tegmark (MIT), and David Rolnick (MIT)

WHY DOES DEEP LEARNING WORK SO WELL?

OR NOT….



WHY DOES DEEP LEARNING WORK SO WELL?

OR NOT….

- Non natural images or adding noise
  ⇒ train the neural network to recognize fakes
- Problem of bad (incomplete) specifications
  ⇒ Create a learning data set including ”noisy” inputs

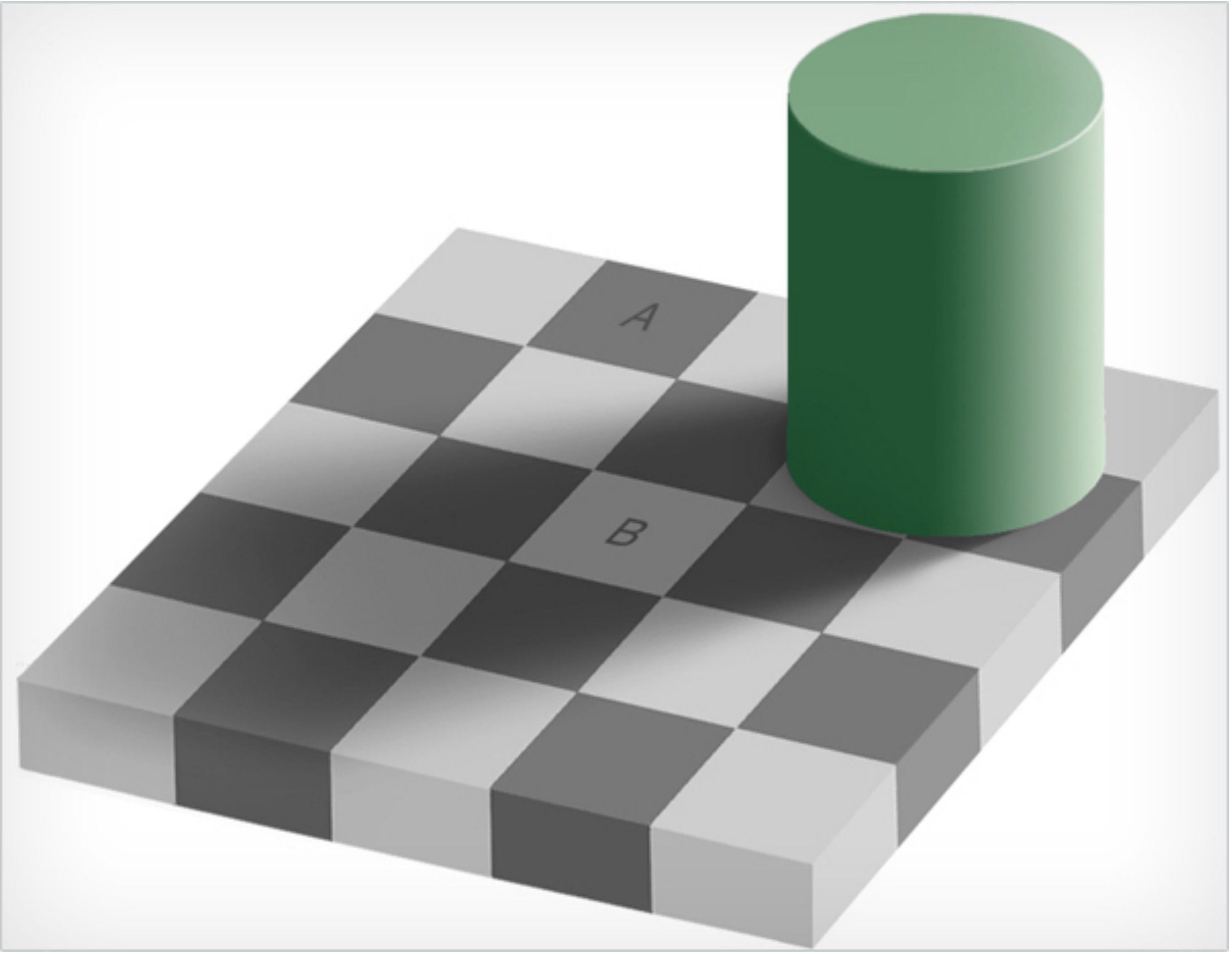
But it is and will remain a problem (like bugs in software)

hosted DNN with no such knowledge. Indeed, the only capability of our black-box adversary is to observe labels given by the DNN to chosen inputs. Our attack strategy consists

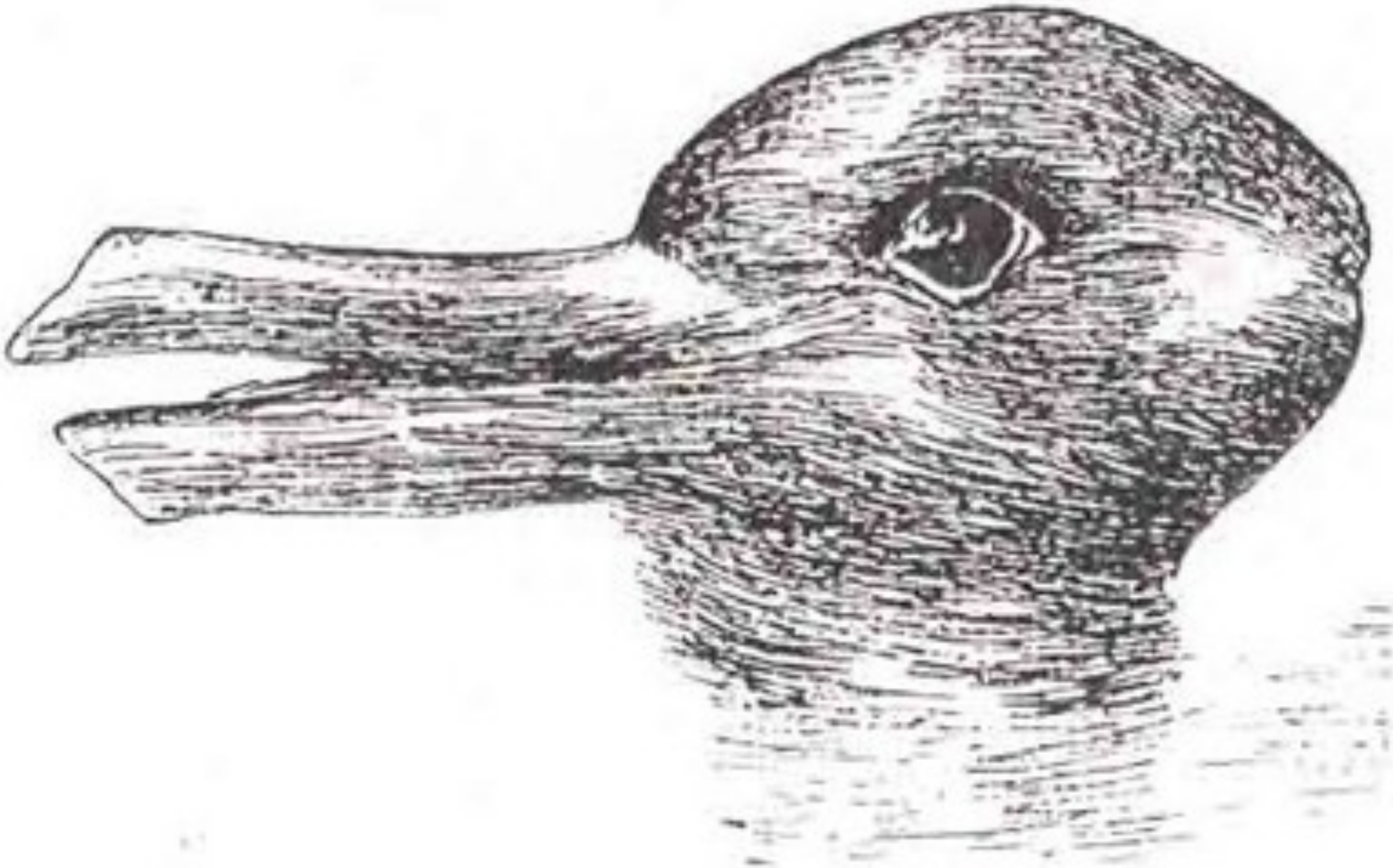
To humans, these images appear to be the same: our bio-

WHY OUR BRAIN DOES NOT ALWAYS WORK

WHY OUR BRAIN DOES NOT ALWAYS WORK



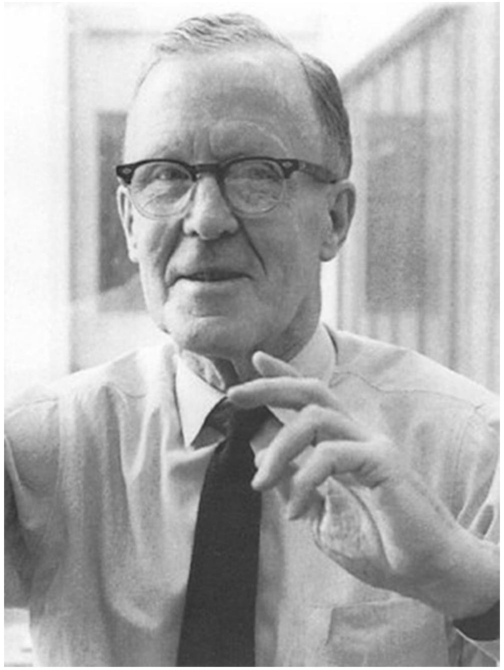
WHY OUR BRAIN DOES NOT ALWAYS WORK



1949: DONALD HEBB

Hebb’s rule or Hebbian theory: an explanation for the adaptation of neurons in the brain during the learning process

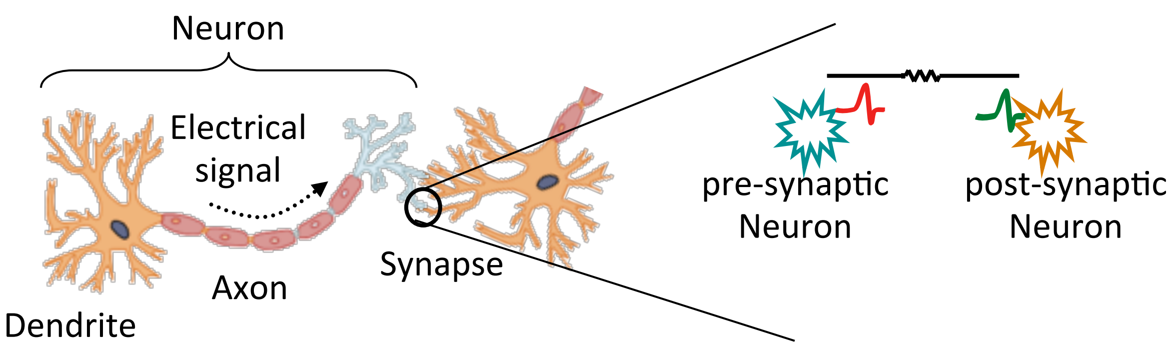
Basic mechanism for synaptic plasticity: an increase in synaptic efficacy arises from the presynaptic cell's repeated and persistent stimulation of the postsynaptic cell.

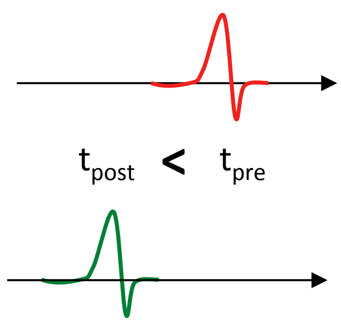


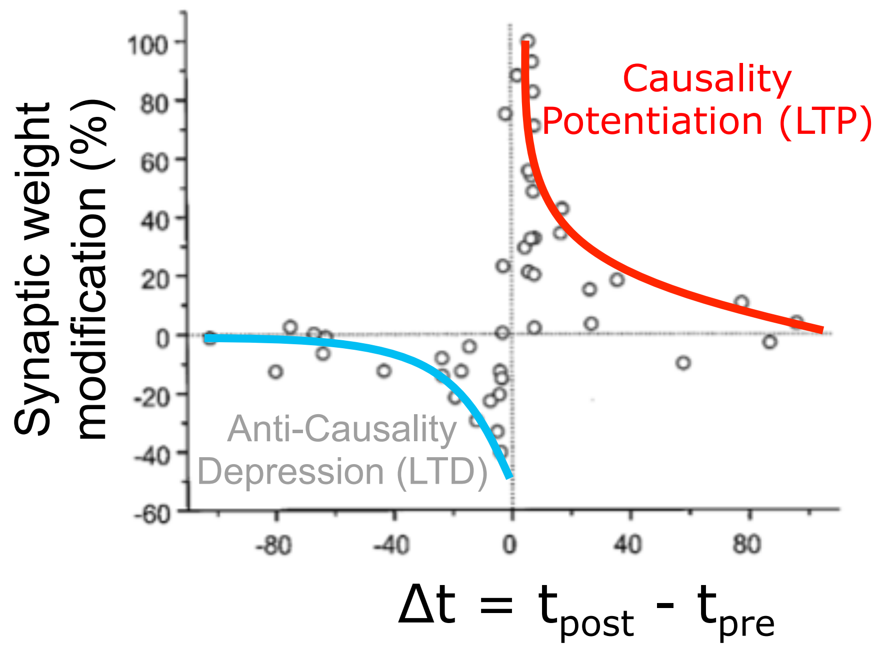
Psychologist, working in the area of neuropsychology

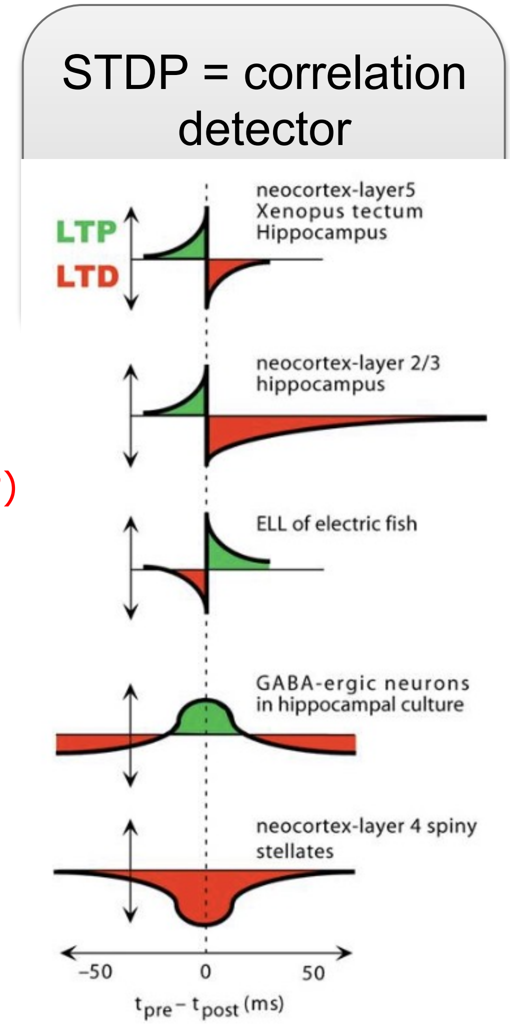
Introduced by Donald Hebb in his 1949 book « *The Organization of Behavior* »

DERIVED FROM HEBB'S RULE: STDP (SPIKE TIMING DEPENDENT PLASTICITY)





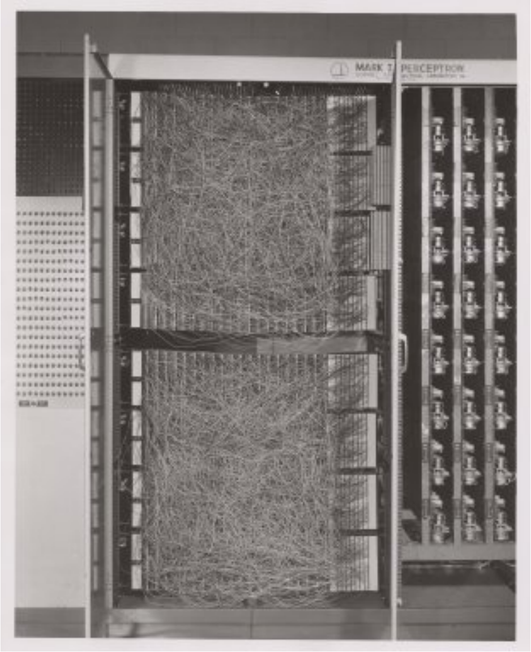




1957: THE PERCEPTRON AND F. ROSENBLATT

The perceptron algorithm was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt.

The perceptron was intended to be a machine, rather than a program, and while its first implementation was in software for the IBM 704, it was subsequently implemented in custom-built hardware as the "Mark 1 perceptron". This machine was designed for image recognition: it had an array of 400 photocells, randomly connected to the "neurons". Weights were encoded in potentiometers, and weight updates during learning were performed by electric motors.



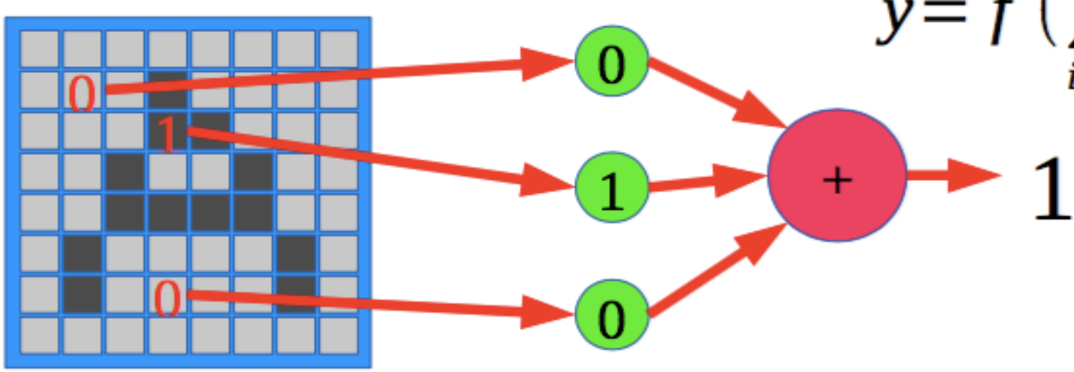
The Perceptron Learning Algorithm

- **Training set:** $(X^1,Y^1),(X^2,Y^2),\ldots\ldots,(X^p,Y^p)$
- **Take one sample** (X^k,Y^k)**, if the desired output is +1 but the actual output is -1**
  - Increase the weights whose input is positive
  - Decrease the weights whose input is negative
- **If the desired is -1 and actual is +1, do the converse.**
- **If desired and actual are equal, do nothing**

$$\boxed{w_i(t+1) = w_i(t) + (y_i^p - f(W'X^p))x_i^p}$$

1986: David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams

$$y = f\left(\sum_{i=1} w_i x_i + w_0\right) = f\left(W'X\right)$$

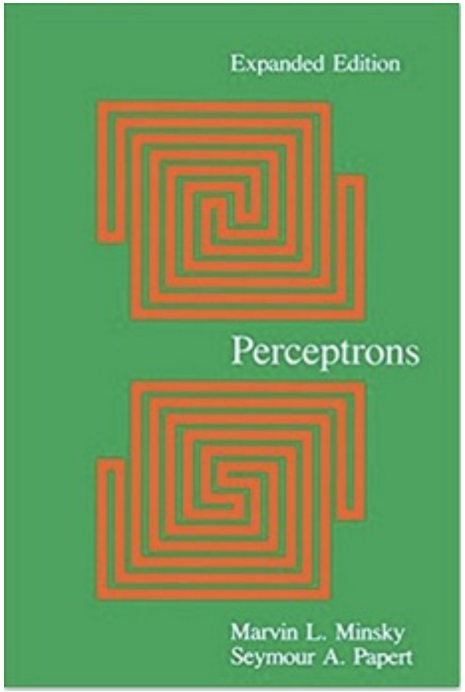


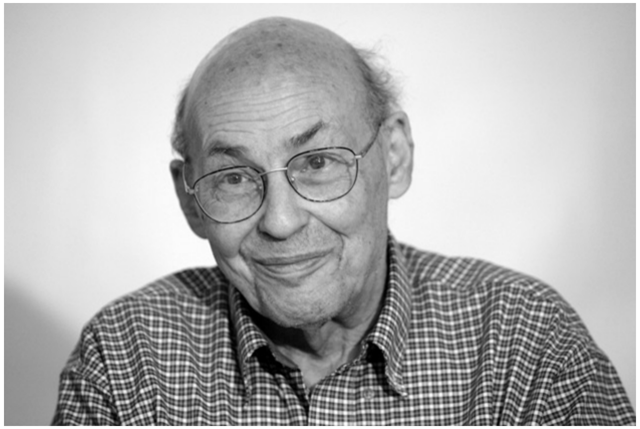
Supervised Learning

1969: MARVIN MINSKY

He developed, with Seymour Papert, the first Logo "turtle".
Minsky also built, in 1951, the first randomly wired neural network learning machine, SNARC.

Minsky wrote the book **Perceptrons** (with Seymour Papert), which became the foundational work in the analysis of artificial neural networks. This book is the center of a controversy in the history of AI, as some claim it to have had great importance in discouraging research of neural networks in the 1970s, and contributing to the so-called ”**First AI winter**".



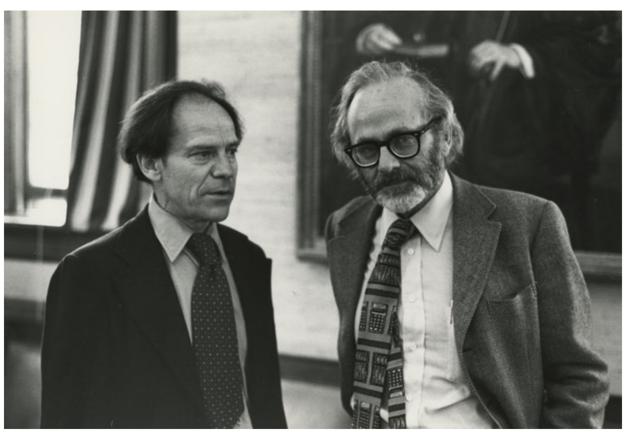


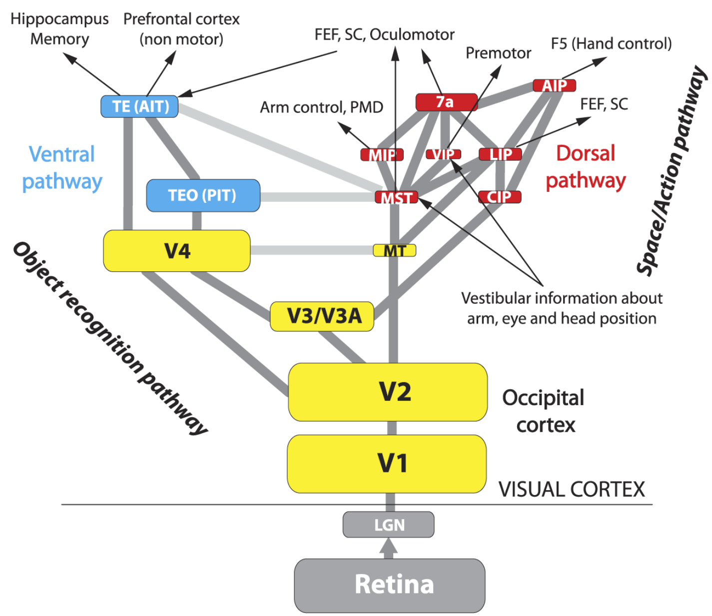
1981: DAVID MARR, DAVID HUBEL ET TORSTEN WIESEL

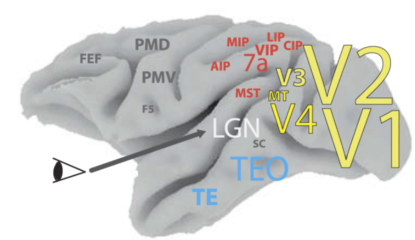
Better understanding how the biological visual system works:

- David Marr: Vision: A computational investigation into the human representation and processing of visual information, which was finished mainly on 1979 summer, was published in 1982 after his death

- Hubel and Wiesel were awarded the Nobel Prize in 1981 for their work on ocular dominance columns in the 1960s and 1970s.







1980: KUNIHIKO FUKUSHIMA

The first Deep Neural Network, inspired by the visual cortex.

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

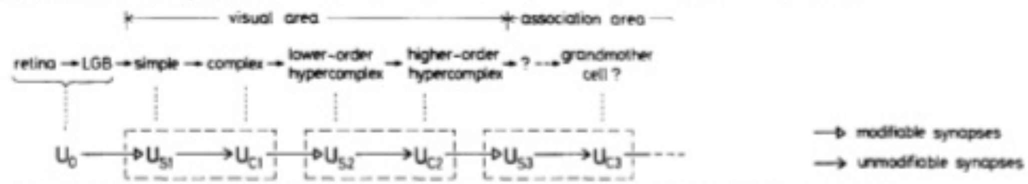


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

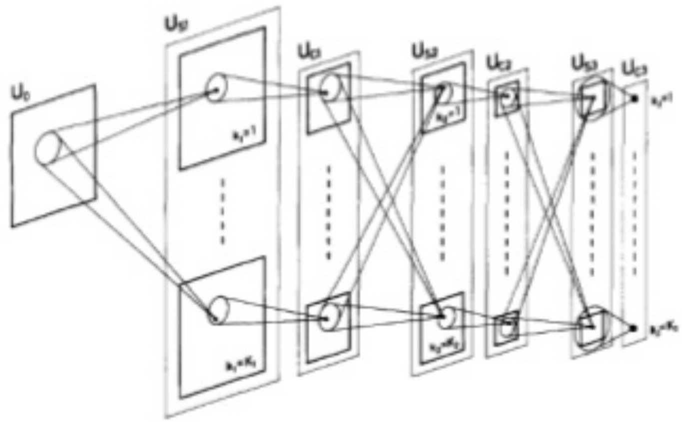


Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

Biol. Cybernetics 36, 193–202 (1980)

AROUND 1986: GEOFFREY HINTON

He was one of the first researchers who demonstrated the use of **generalized back-propagation algorithm** for training multi-layer neural networks.

He co-invented **Boltzmann machines** with David Ackley and Terry Sejnowski.

His other contributions to neural network research include distributed representations, time delay neural network, mixtures of experts, Helmholtz machines and Product of Experts

He is now working for Google.

Cognitive psychologist and computer scientist

AROUND 1985: YANN LE CUN

In 1985, he proposed and published (in French), an early version of the learning algorithm known as **error backpropagation**

Near 1989, he developed a number of new machine learning methods, such as a biologically inspired model of image recognition called **Convolutional Neural Networks**, the "Optimal Brain Damage" regularization methods, and the Graph Transformer Networks method which he applied to handwriting recognition and OCR.

The **bank check recognition system** that he helped develop was widely deployed by NCR and other companies, reading over 10% of all the checks in the US in the late 1990s and early 2000s.

In 2013, LeCun became the first director of Facebook AI Research in New York City.

1990'S NEUROCOMPUTERS...

Adaptive Solutions : CNAPS-1064 (about 1990)

- SIMD // machine based on a 64 PE chip (80 in total).
- 0.8micron, 2 metal CMOS (1inch on a side), 11million transistors
- 4W @ 25MHz

CNAPS/VMEbus Accelerator Board

- Up to ten billion MACS
- 64 to 256 CNAPS processors per board
- Up to 512 processors with optional expansion board
- Standard 6U X 160 mm VMEbus form factor

1990'S NEUROCOMPUTERS...

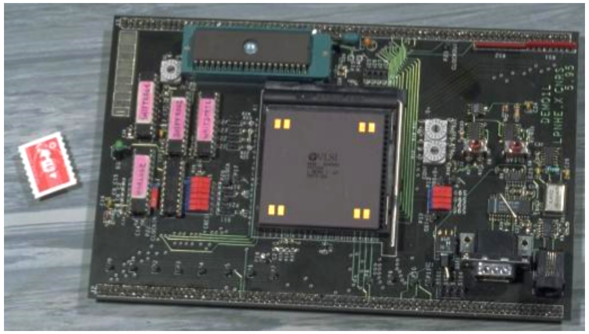
Siemens : MA-16 Chips (SYNAPSE-1 Machine 1994)

- Synapse-1, neurocomputer with 8xMA-16 chips
- Synapse3-PC, PCI board with 2xMA-16 (1.28 Gpcs)
- about 8,000 times as fast as a Sun Workstation (Sparc-2)

1990'S NEUROCOMPUTERS...

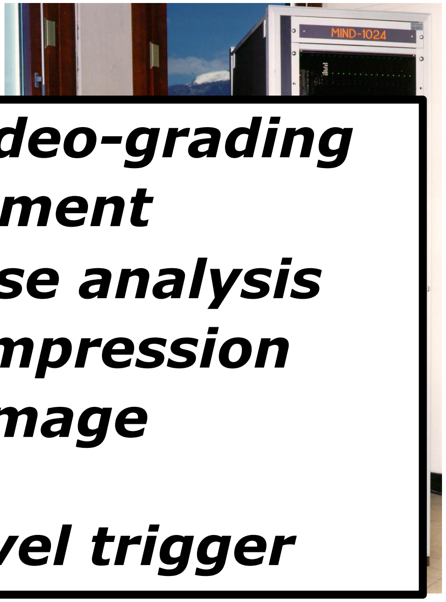
Philips : L-Neuro

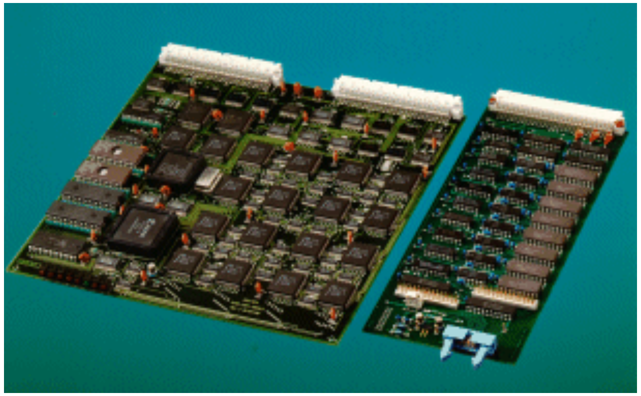
- 1st Gen 16 PEs 26 MCps (1990)
- 2nd Gen 12 PEs 720 MCps (1994)
- ➢ Used in satellite, fruit sorting, PCB inspection, sleep analysis, …



CEA's MIND machine

- Hybrid analog/digital: MIND-128 (
- Fully digital: MIND-1024 (1991)



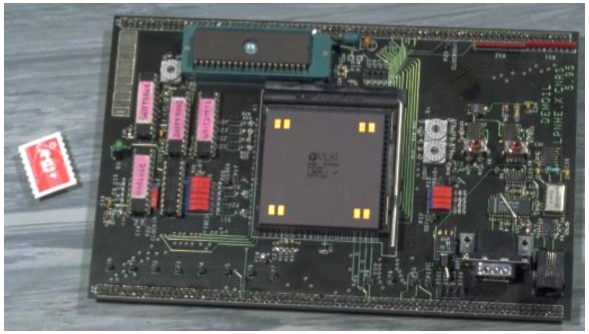


- ☐ ***Orange video-grading***
- ☐ ***Chip alignment***
- ☐ ***Sleep phase analysis***
- ☐ ***Image compression***
- ☐ ***Satellite image analysis***
- ☐ ***LHC 1st level trigger***

1990'S NEUROCOMPUTERS...

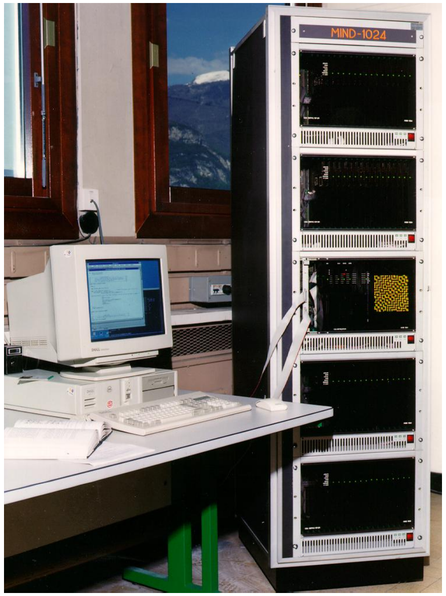
Philips : L-Neuro

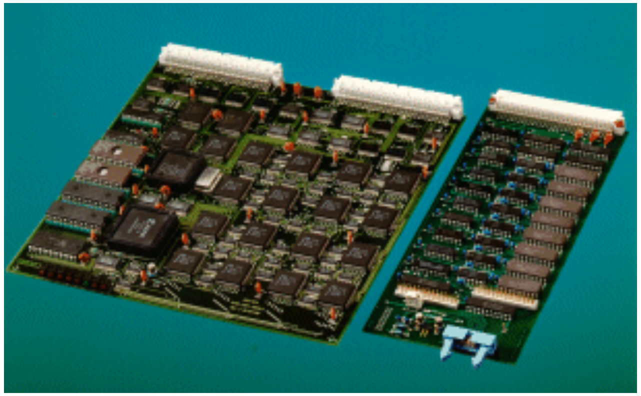
- 1st Gen 16 PEs 26 MCps (1990)
- 2nd Gen 12 PEs 720 MCps (1994)
- ➢ Used in satellite, fruit sorting, PCB inspection, sleep analysis, …

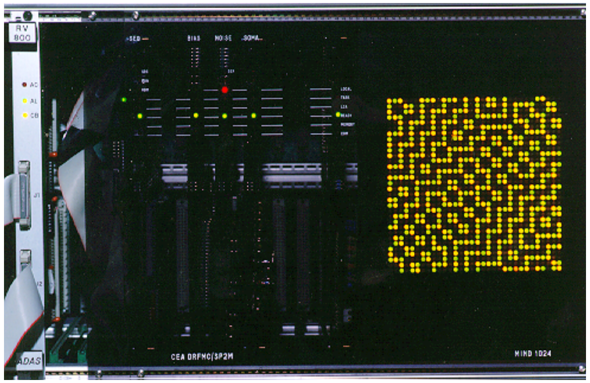


CEA's MIND machine

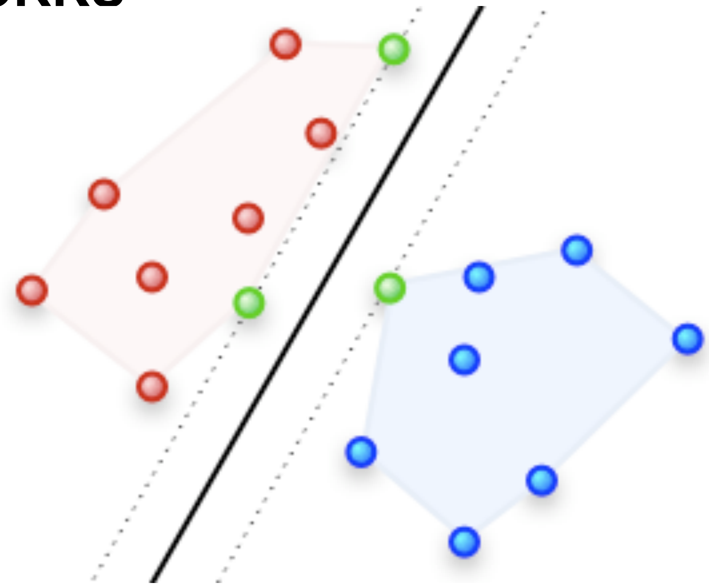
- Hybrid analog/digital: MIND-128 (1986)
- Fully digital: MIND-1024 (1991)







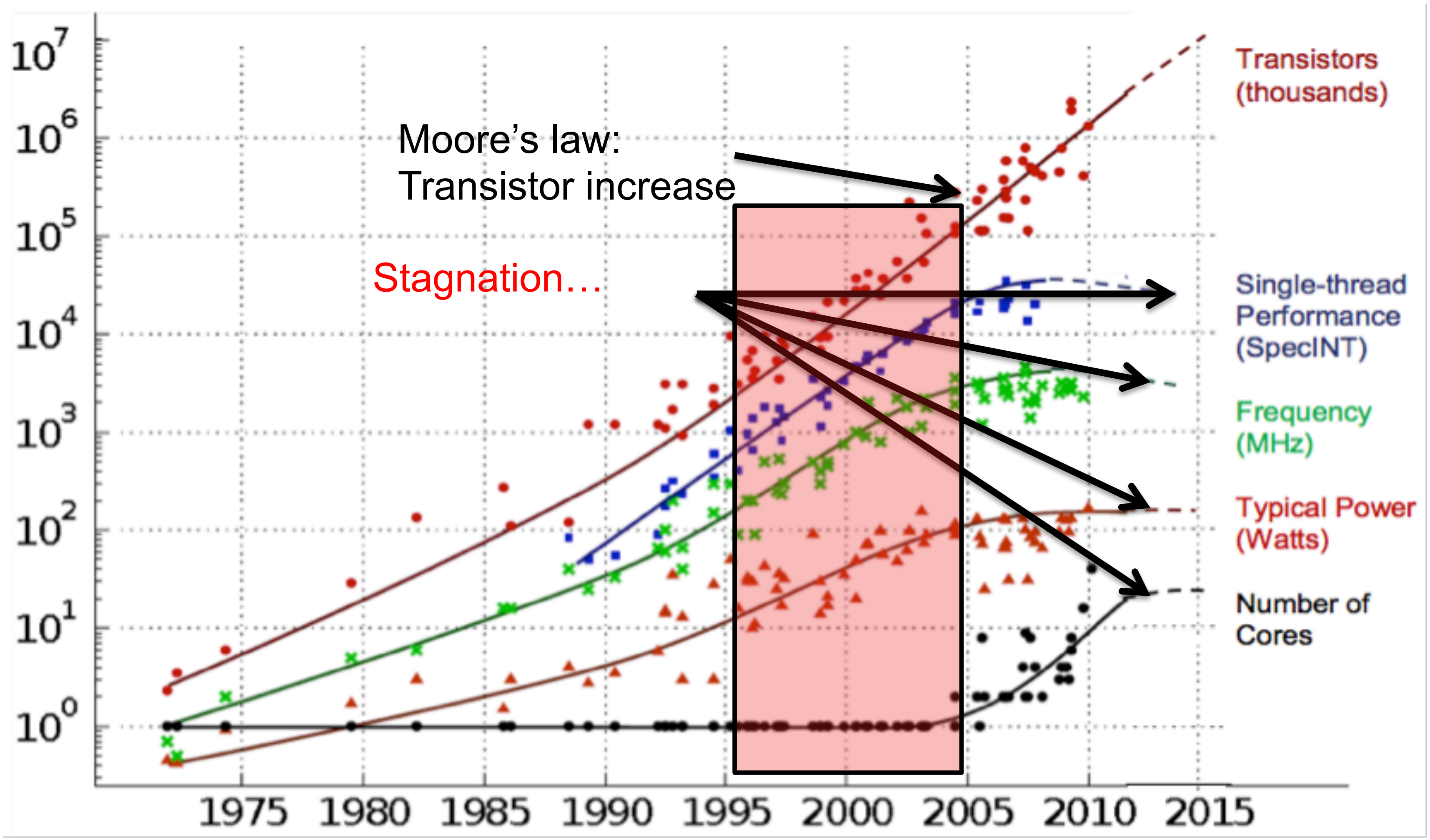
1995: SVM OR THE 2ND WINTER OF NEURAL NETWORKS



Support Vector Machines (SVMs)
The original SVM algorithm was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963.

In 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick to maximum-margin hyperplanes. The current standard incarnation (soft margin) was proposed by Corinna Cortes and Vapnik in 1993 and published in 1995.

MOORE 'S LAW AND DENNARD SCALING

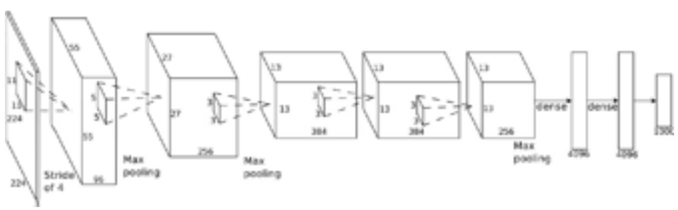


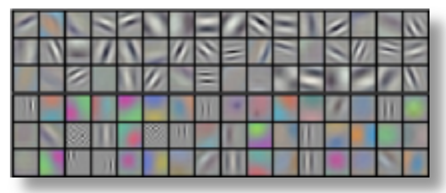
Source from C Moore, « Data Processing in ExaScale-Class Computer Systems », Salishan, April 2011

2012: DEEP NEURAL NETWORKS RISE AGAIN

They give the ***state-of-the-art performance*** e.g. in image classification

- **ImageNet classification (Hinton's team, hired by Google)**
  - 14,197,122 images, 1,000 different classes
  - Top-5 17% error rate (huge improvement) in 2012 (now ~ 3.5%)





"**Supervision**" network
Year: 2012
650,000 neurons
60,000,000 parameters
630,000,000 synapses

- **Facebook's 'DeepFace' Program (labs headed by Y. LeCun)**
  - 4.4 million images, 4,030 identities
  - 97.35% accuracy, vs. 97.53% human performance

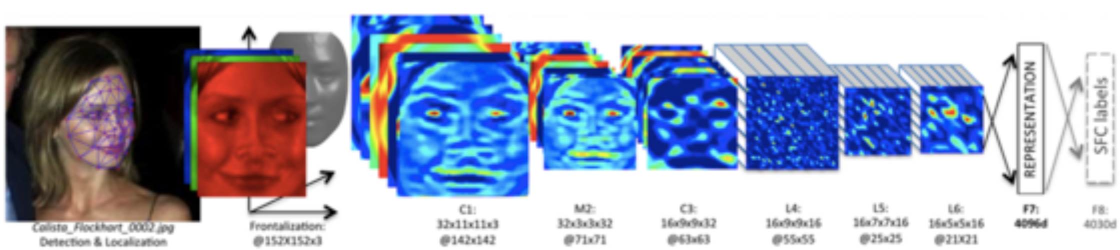


Figure 2. **Outline of the *DeepFace* architecture.** A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

From: Y. Taigman, M. Yang, M.A. Ranzato, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification"

ImageNet: Classification

- Give the name of the dominant object in the image
- Top-5 error rates: if correct class is not in top 5, count as error
  - Black:ConvNet, Purple: no ConvNet

2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

2013 Teams	%error
Clarifai (NYU spinoff)	11.7
NUS (singapore)	12.9
Zeiler-Fergus (NYU)	13.5
A. Howard	13.5
OverFeat (NYU)	14.1
UvA (Amsterdam)	14.2
Adobe	15.2
VGG (Oxford)	15.2
VGG (Oxford)	23.0

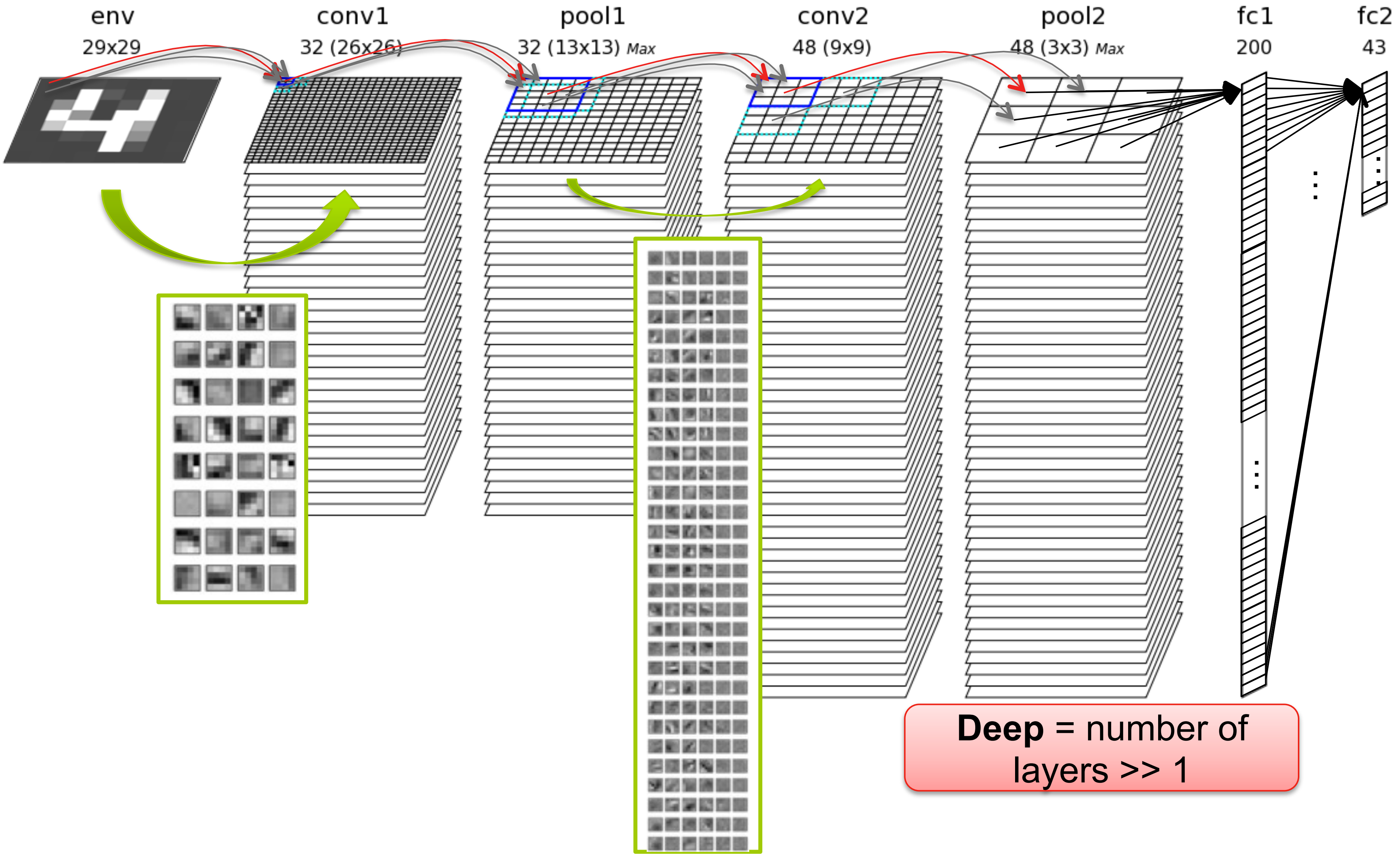
2014 Teams	%error
GoogLeNet	6.6
VGG (Oxford)	7.3
MSRA	8.0
A. Howard	8.1
DeeperVision	9.5
NUS-BST	9.7
TTIC-ECP	10.2
XYZ	11.2
UvA	12.1

COMPETITION ON IMAGENET !

Nom de l'algorithme	Date	Erreur sur le jeu de test
Supervision	2012	15.3%
Clarifai	2013	11.7%
GoogLeNet	2014	6.66%
Niveau humain		5%
Microsoft	05/02/2015	4.94%
Google	02/03/2015	4.82%
Baidu/ Deep Image	10/05/2015	4.58%
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences	10/12/2015 (le CNN a 152 couches!)	3.57%
Google Inception-v3 (Arxiv)	2015	3.5%
	Maintenant	?

EXAMPLES OF RESULTS (IMAGENET)

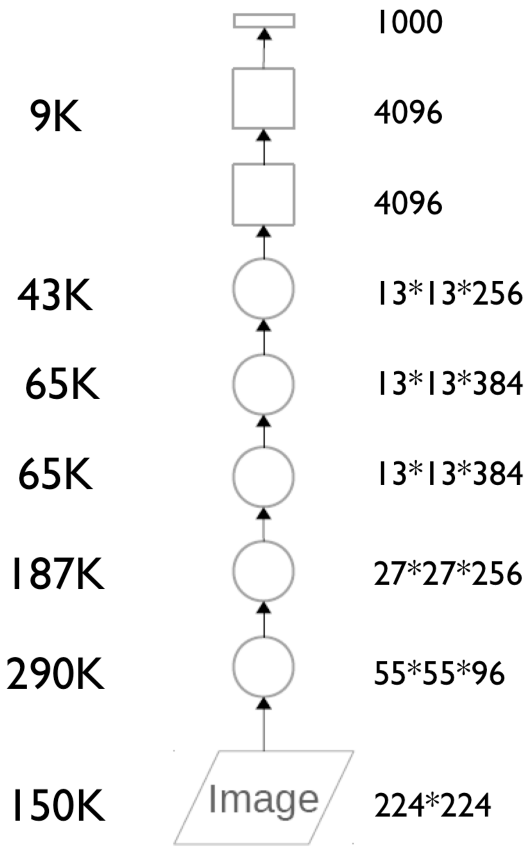
WHAT IS A CNN?



Deep = number of layers >> 1

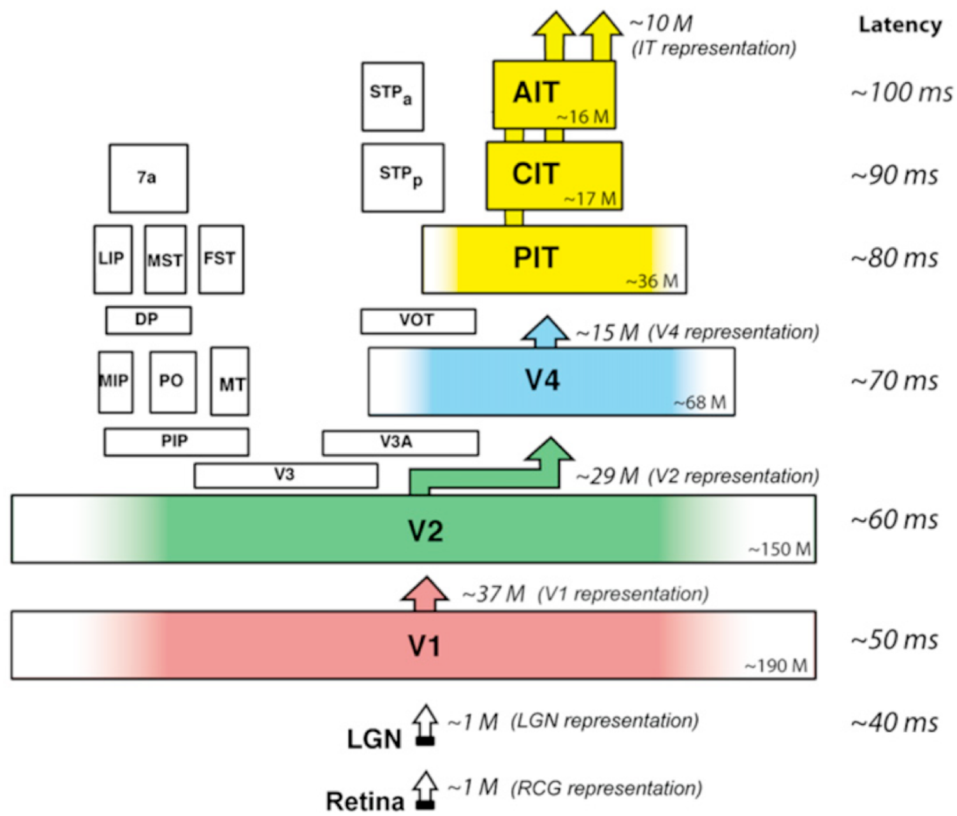
SUPERVISION VS PRIMATE VISION

Supervision



Total 650 K neurons

Primate Visual System

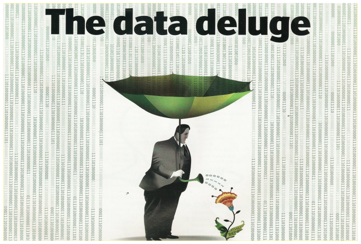


Total 478 M neurons

From Simon Thorpe

WHY NEURAL NETWORKS ARE BACK?

Application needs – "data deluge" of unstructured data



- Images, video, natural signals, …

Algorithmic progress

- "Training" of ***Deep*** Neural Networks (DNN) that outperform classical approaches

Availability of "big data" sets

- Terabyte of (labelled) data

Large amount of (parallel) processing power

- GPU are well suited for the learning phase

Software crisis

- Explicitly programming a large set of processors is difficult, Neural Networks replace imperative programming by a "programming" by examples.

Deep Learning is Everywhere (ConvNets are Everywhere)

- **Lots of applications at Facebook, Google, Microsoft, Baidu, Twitter, IBM…**
  - Image recognition for photo collection search
  - Image/Video Content filtering: spam, nudity, violence.
  - Search, Newsfeed ranking
- **People upload 800 million photos on Facebook every day**
  - (2 billion photos per day if we count Instagram, Messenger and Whatsapp)
- **Each photo on Facebook goes through two ConvNets within 2 seconds**
  - One for image recognition/tagging
  - One for face recognition (not activated in Europe).
- **Soon ConvNets will really be everywhere:**
  - self-driving cars, medical imaging, augemnted reality, mobile devices, smart cameras, robots, toys…..

PIXEL WISE IMAGE SEGMENTATION

- **DNN technic: Fully-CNN + Unpooling (for high resolution segmentation)**

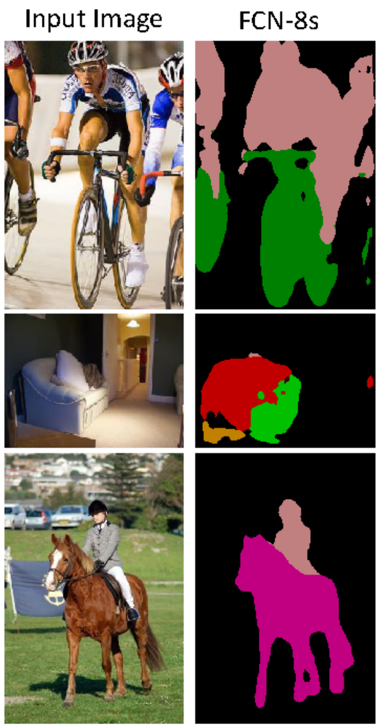
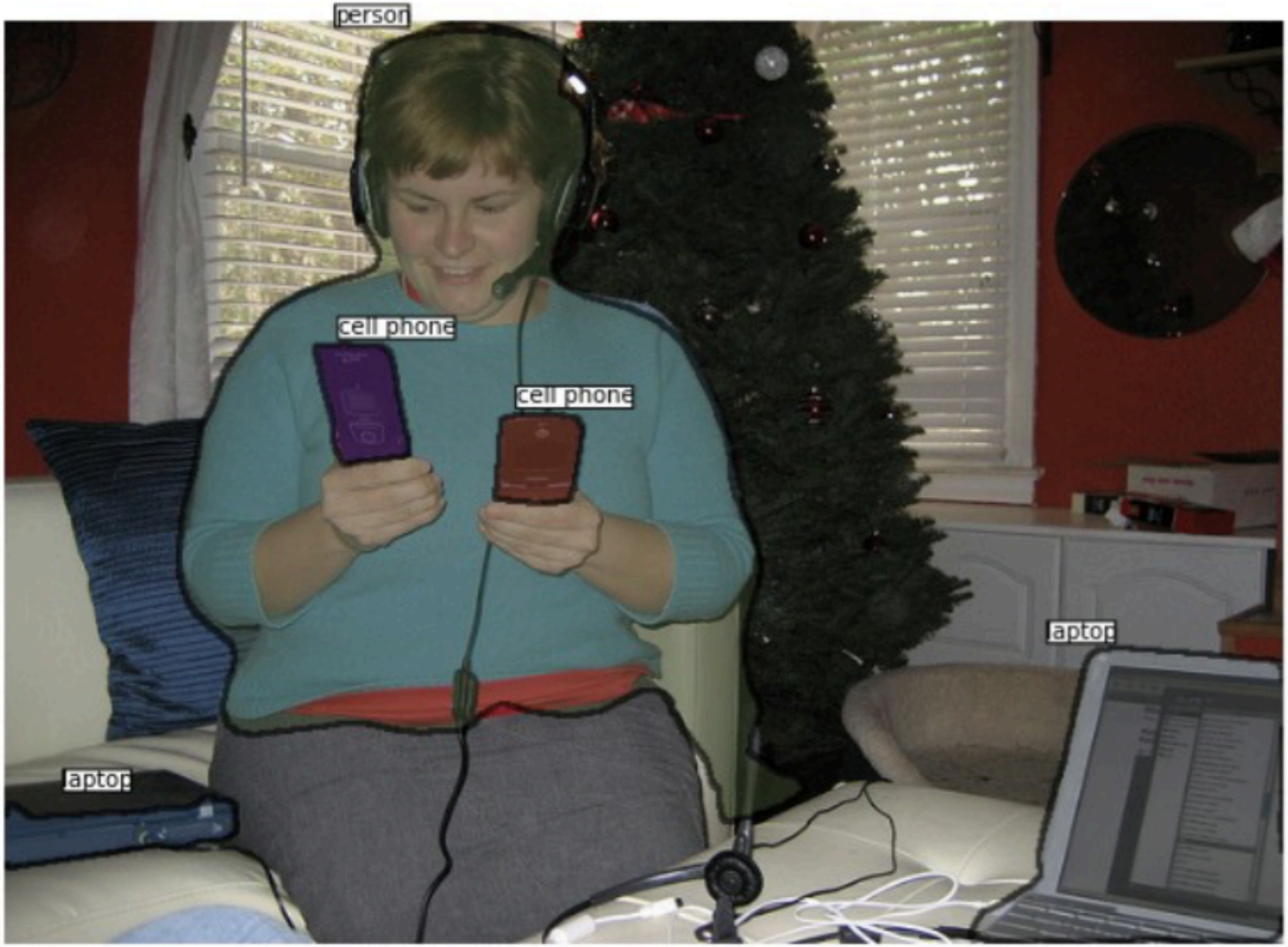


IMAGE ROI EXTRACTION AND CLASSIFICATION

- DNN technic: Faster-RCNN (or similar: YOLO, SSD…)

Simultaneous face detection and pose estimation

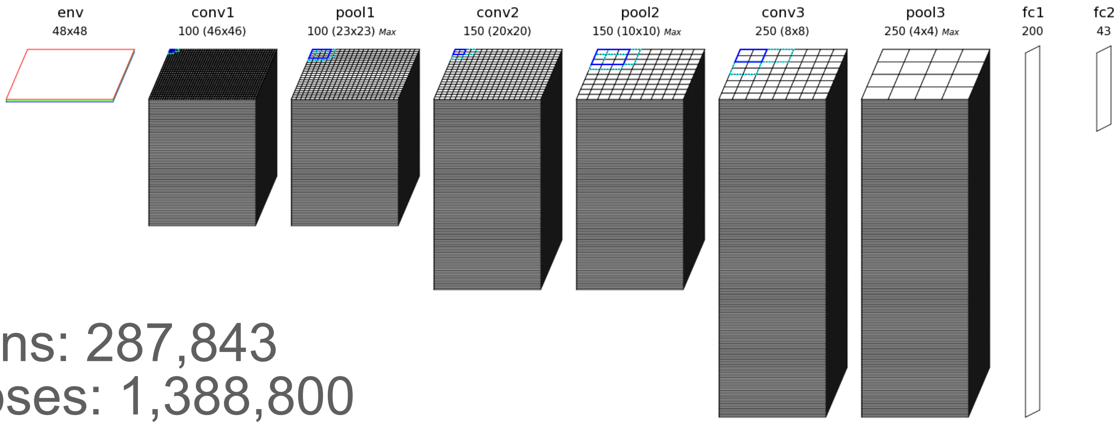
Results



EXAMPLE OF SIZE OF A TYPICAL CNN

The German Traffic Sign Recognition Benchmark (GTSRB)

43 traffic sign types
> 50,000 images



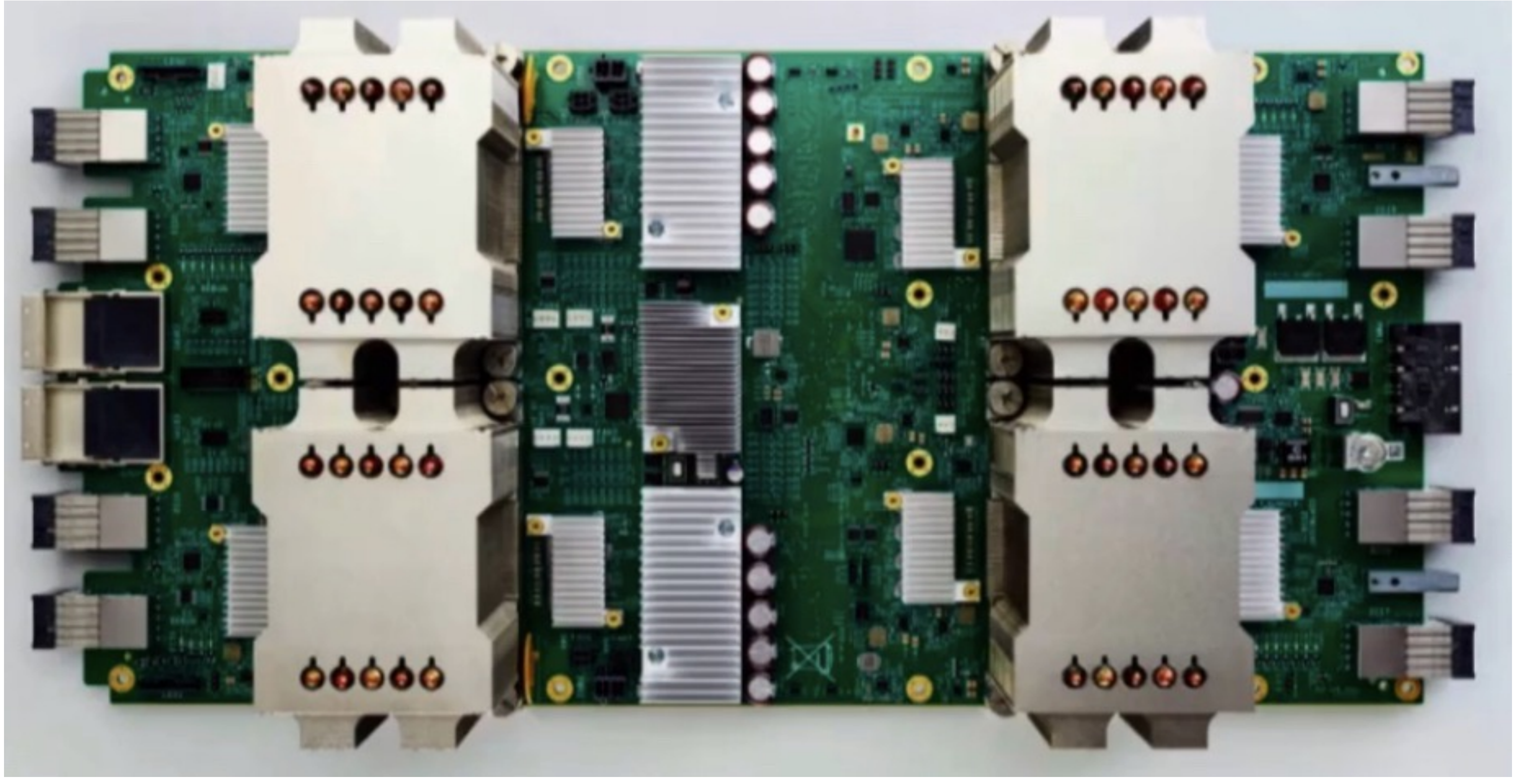
- Neurons: 287,843
- Synapses: 1,388,800
  - Total memory: 1.5MB (with 8 bits synapses)
- Connections: 124,121,800

From: D. Ciresan, U. Meier, J. Masci, J. Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Networks (32), pp. 333-338, 2012

Near human recognition (> 98%)

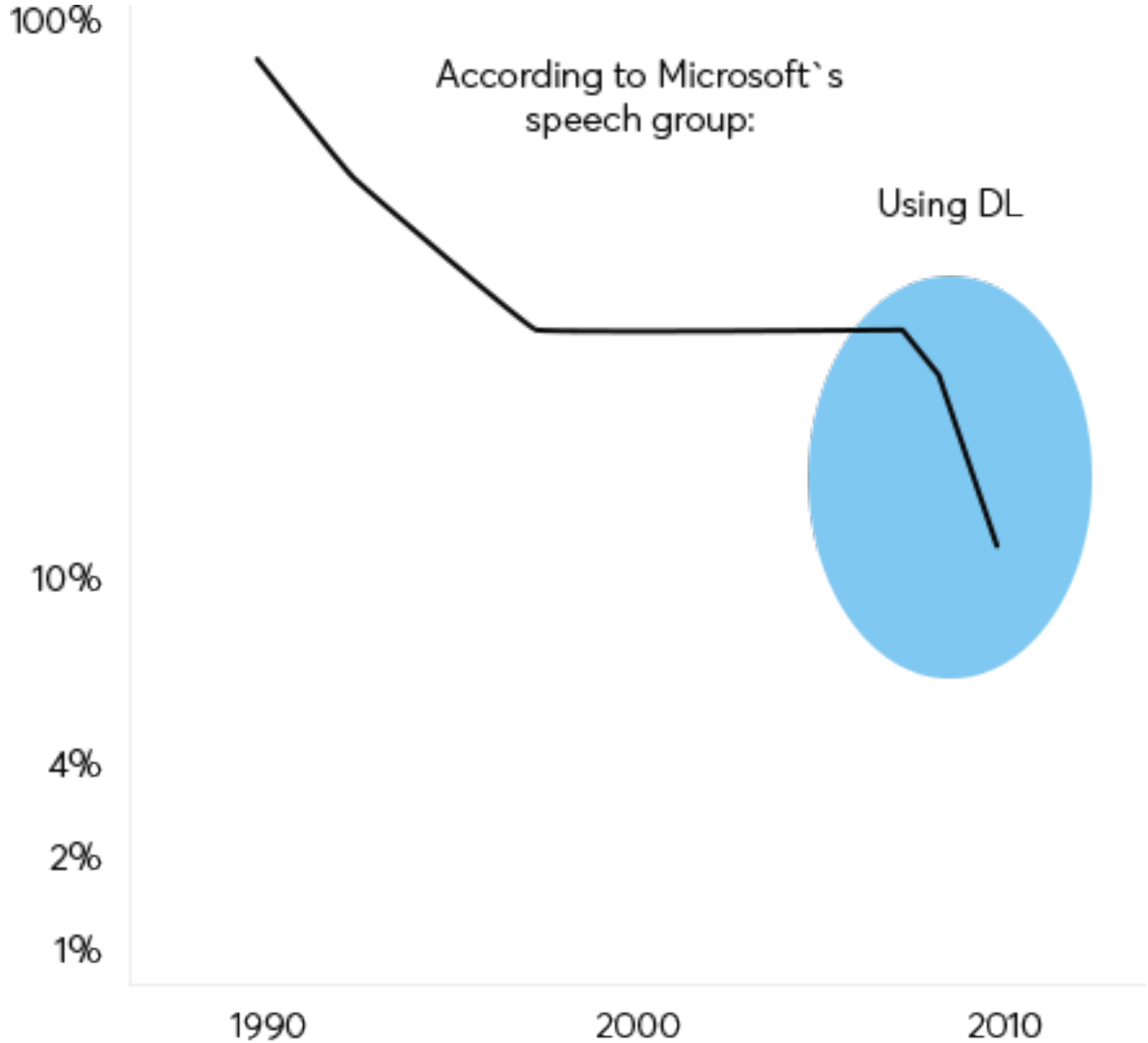
2017: GOOGLE'S CUSTOMIZED HARDWARE…

… required to increase energy efficiency
with accuracy adapted to the use (e.g. float 16)



Google's TPU2 : training and inference in a **180 teraflops$_{16}$** board
(over 200W per TPU2 chip according to the size of the heat sink)

DEEP LEARNING AND VOICE RECOGNITION



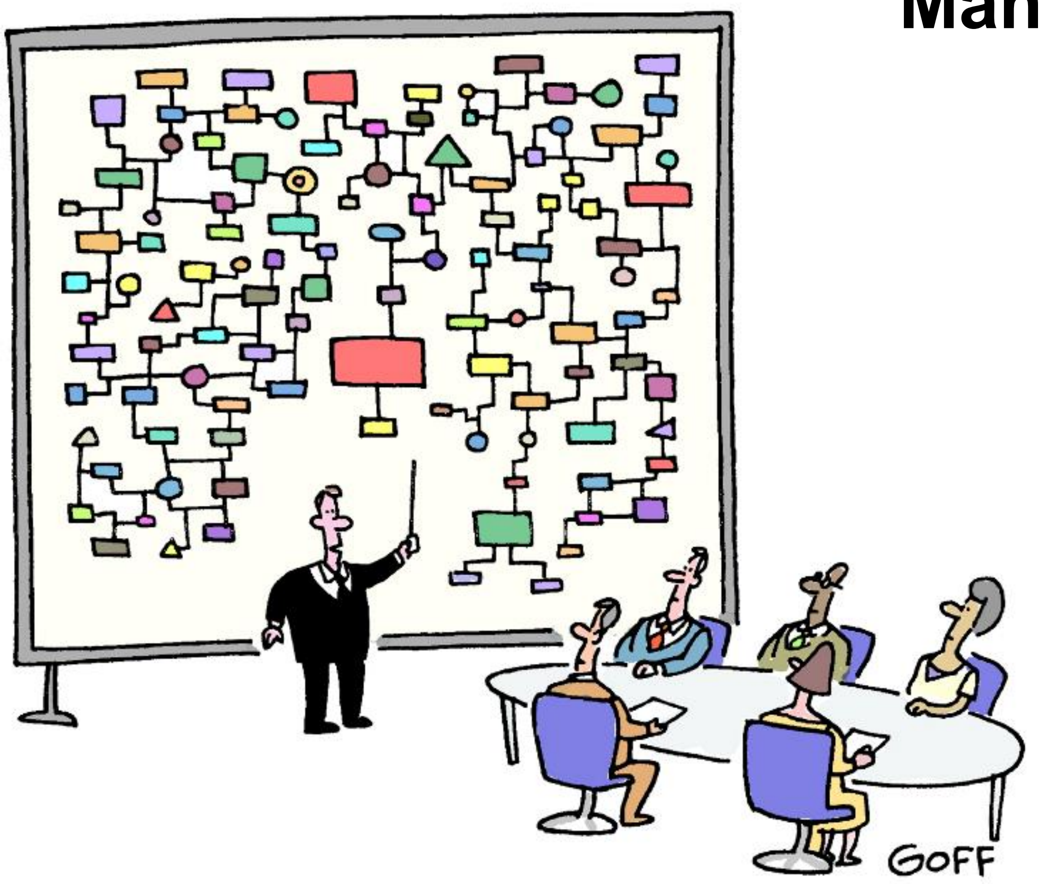
Deep Learning in **Speech Recognition**

2017: GOOGLE'S CUSTOMIZED HARDWARE…

… required to increase energy efficiency
with accuracy adapted to the use (e.g. float 16)

Google's TPU2 : 11.5 petaflops16 of machine learning number crunching
(and guessing about 400+ KW…, 100+ GFlops16/W)

Managing complexity



"And that's why we need a computer."

Cognitive solutions for complex computing systems:

- Using **AI techniques for computing systems**
  - Creating new hardware
  - Generating code
  - Optimizing systems
- Similar to ***Generative design*** for mechanical engineering

AI FOR MAKING COMPUTING SYSTEMS: "GENERATIVE DESIGN" APPROACH

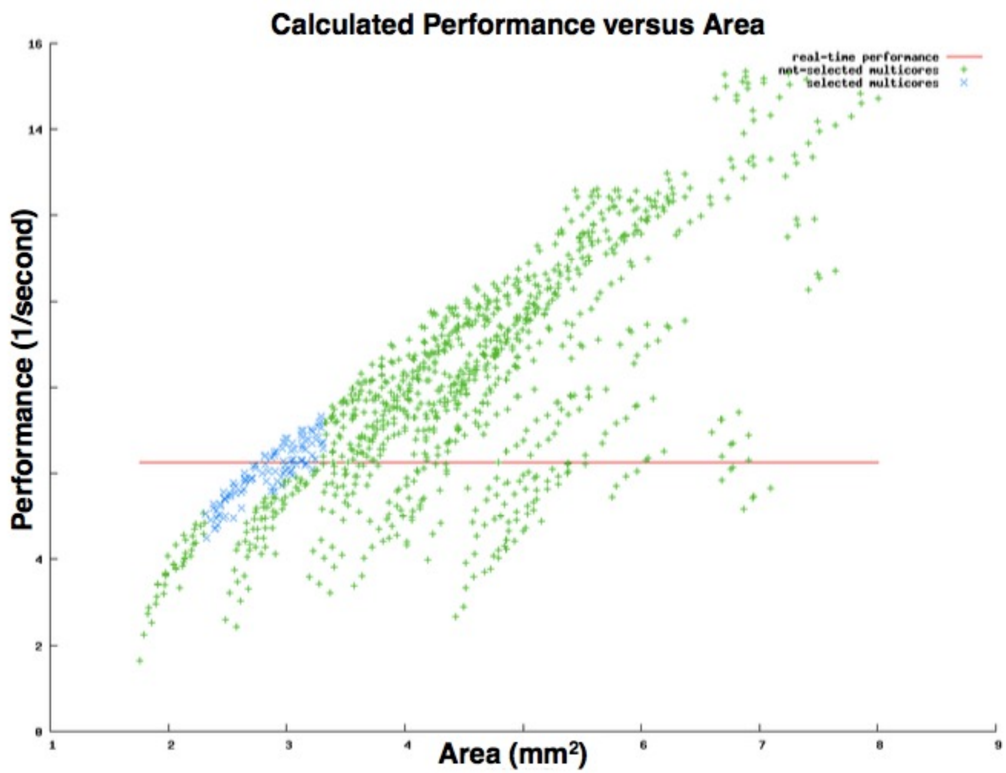
The user ***only states desired goals and constraints***
-> The ***complexity wall*** might ***prevent explaining*** the solution

Motorcycle swingarm: the piece that hinges the rear wheel to the bike's frame

EXAMPLE: DESIGN SPACE EXPLORATION FOR DESIGN MULTI-CORE PROCESSORS[1] (2010)

- **Ne-XVP project – Follow-up of the TriMedia VLIW (https://en.wikipedia.org/wiki/Ne-XVP)**
- **1,105,747,200 heterogeneous multicores in the design space**
- **2 millions years to evaluate all design points**
- **AI inspired techniques allowed to reduce the induction time to only few days**

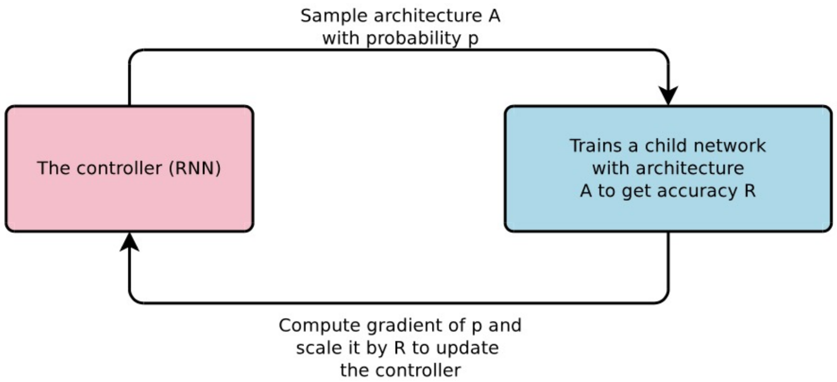
=> *x16 performance increase*



[1] M. Duranton et all., “Rapid Technology-Aware Design Space Exploration for Embedded HeterogeneousMultiprocessors” in Processor and System-on-Chip Simulation, Ed. R. Leupers, 2010

2017: GOOGLE; USING DEEP LEARNING TO DESIGN DEEP LEARNING

“*Neural Architecture Search*”, using a recurrent neural network to compose neural network architectures using reinforcement learning on CIFAR-10 (character recognition)

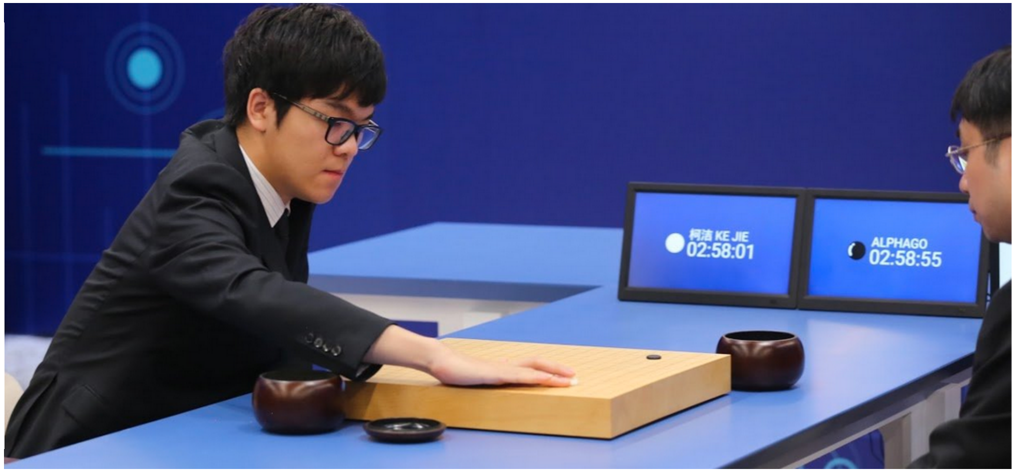


Model	Depth	Parameters	Error rate (%)
Network in Network (Lin et al., 2013)	-	-	8.81
All-CNN (Springenberg et al., 2014)	-	-	7.25
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97
Highway Network (Srivastava et al., 2015)	-	-	7.72
Scalable Bayesian Optimization (Snoek et al., 2015)	-	-	6.37
FractalNet (Larsson et al., 2016)	21	38.6M	5.22
with Dropout/Drop-path	21	38.6M	4.60
ResNet (He et al., 2016a)	110	1.7M	6.61
ResNet (reported by Huang et al. (2016c))	110	1.7M	6.41
ResNet with Stochastic Depth (Huang et al., 2016c)	110	1.7M	5.23
	1202	10.2M	4.91
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet ($L = 40, k = 12$) Huang et al. (2016a)	40	1.0M	5.24
DenseNet($L = 100, k = 12$) Huang et al. (2016a)	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) Huang et al. (2016a)	100	27.2M	3.74
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65

From arXiv:1611.01578v2, Barret Zoph, Quoc V. Le
Google Brain

2017: THE GAME OF GO

Ke Jie (human world champion in the “Go” game), after being defeated by AlphaGo on May 27th 2017, will work with Deepmind to make a tool from AlphaGo to further help Go players to enhance their game.



ALPHAGO ZERO: SELF-PLAYING TO LEARN

ALPHA ZERO: SELF-PLAYING TO LEARN

The program started from random play given no domain knowledge except the game rules according to an arXiv paper by DeepMind researchers published Dec. 5.

"I always wondered how it would be if a superior species landed on Earth and showed us how they played chess," grandmaster Peter Nielsen told BBC.

r himself. "It plays

mpion program in

practicing against

"Now I know."


Max Tegmark
@tegmark

"What we're seeing here is a model free from human bias and presuppositions. It can learn whatever it determines is optimal, which may indeed be more nuanced that our own conceptions of the same," MIT computer scientist Nick Hynes told Gizmodo following the October victory.

"AlphaZero was not 'taug
endgame tables, and ap
side pawns. This would

"It's like an alien civilisation inventing its own mathematics."

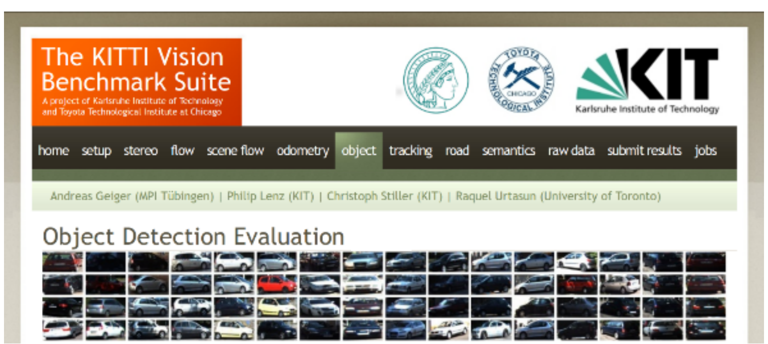
a combustion engine, then it experiments numerous times with every combination possible until it builds a Ferrari. …
The program had four hours to play itself many, many times, thereby becoming its own teacher."

DEEP MANTA

MANY-TASK DEEP NEURAL NETWORK FOR VISUAL OBJECT RECOGNITION

Applications

Driving assistance, autonomous driving
Smart city
Video-protection
Advanced Manufacturing



Technology

1. Object detection
2. Fine-grained recognition
3. Accurate pose estimation
4. 2D/3D localisation
5. Part localisation
6. Part visibility characterization

Performance

KITTI Benchmark:

- 1st rank in vehicle orientation estimation
- Top-10 in object detection

Runs at 10 Hz on Nvidia Gtx 1080

CVPR 2017 : F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière and T. Château
Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image.

DEEP MANTA

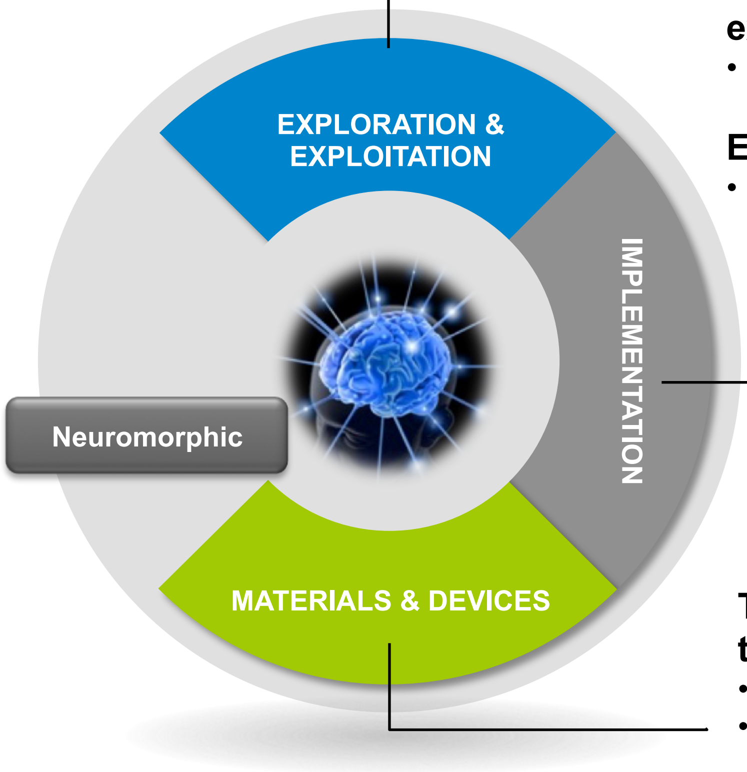
MANY-TASK DEEP NEURAL NETWORK
FOR VISUAL OBJECT RECOGNITION

FROM RESEARCH TO INDUSTRY
cea tech

What are we doing at CEA/DRT/DACLE on Deep Learning?

Centre de Grenoble
17 rue des Martyrs
38054 Grenoble Cedex

DEEP LEARNING AND NEUROMORPHIC SYSTEMS IN CEA/DRT/DACLE



Tools for fast and accurate Neural Network (NN) exploration Architecture benchmarking

- Neural Network exploration (including with spike coding and new materials)

Exploitation of Deep neural Networks

- Image recognition, annotation and indexing

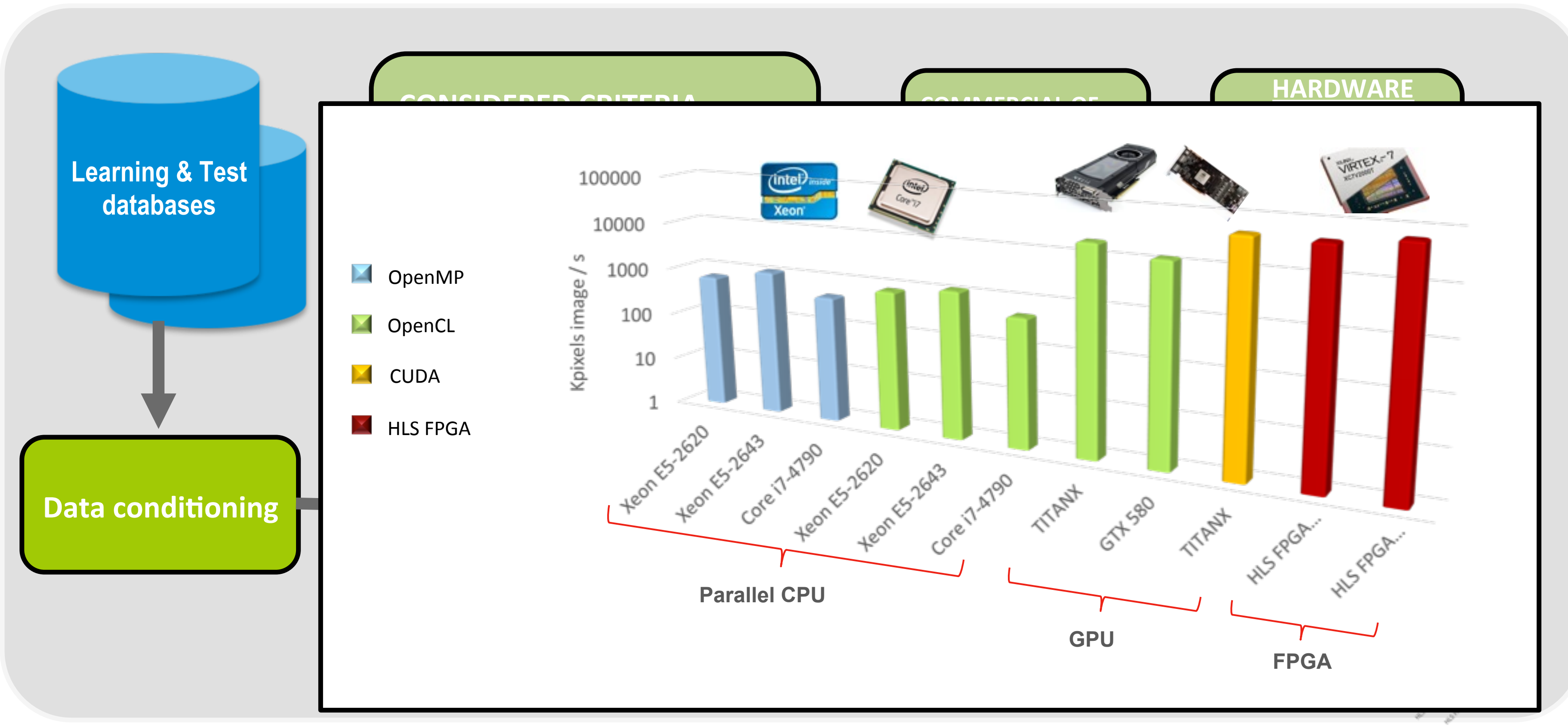
Diversity of implementations:

- Software solution / GPU
- Reconfigurable devices / FPGA
- Dedicated implementations
  - Full CMOS and binary coding
  - Full CMOS and "spike coding"
  - Using new materials

Take full advantage of advanced devices to break the density and power issues:

- 3D integration, CoolCube^{TM}.
- RRAM, PCM and new devices,

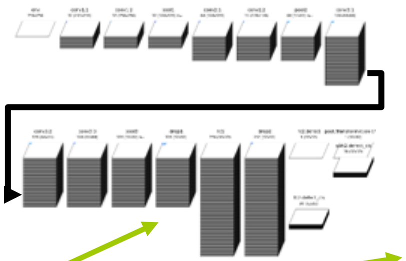
N2D2: NEURAL NETWORK DESIGN & DEPLOYMENT



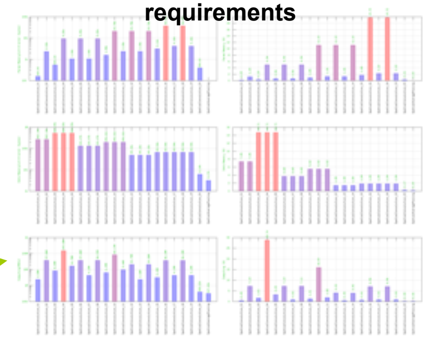
N2D2 INI network description file

```
; Database
[database]
Type=MNIST_IDX_Database
Validation=0.2

; Environment
[env]
SizeX=24
SizeY=24
BatchSize=128

[env.Transformation]
Type=PadCropTransformation
Width=[env]SizeX
Height=[env]SizeY

[env.OnTheFlyTransformation]
Type=DistortionTransformation
ApplyTo=LearnOnly
ElasticGaussianSize=21
ElasticSigma=6.0
ElasticScaling=36.0
Scaling=10.0
Rotation=10.0

; First layer (convolutionnal)
[conv1]
Input=env
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=6
Stride=2
ConfigSection=common.config

; Second layer (convolutionnal)
[conv2]
Input=conv1
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=12
Stride=2
ConfigSection=common.config

; Third layer (fully connected)
[fc1]
Input=conv2
Type=Fc
NbOutputs=100
ConfigSection=common.config

; Output layer (fully connected)
[fc2]
Input=fc1
Type=Fc
NbOutputs=10
ConfigSection=common.config

; Softmax layer
[soft]
Input=fc2
Type=Softmax
NbOutputs=10
WithLoss=1
ConfigSection=common.config

; Common solvers config
[common.config]
WeightsSolver.LearningRate=0.05
WeightsSolver.Decay=0.0005
Solvers.LearningRatePolicy=StepDecay
Solvers.LearningRateStepSize=[sp]_EpochSize
Solvers.LearningRateDecay=0.993
```

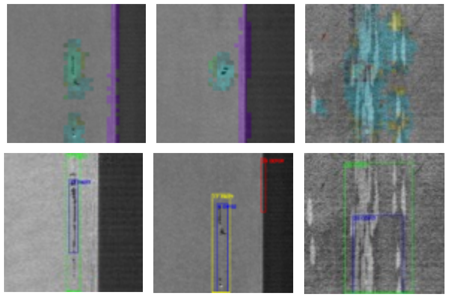
Dataflow visualization

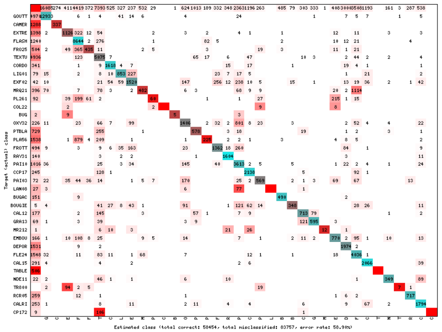


Layer-wise detailed memory and computing requirements

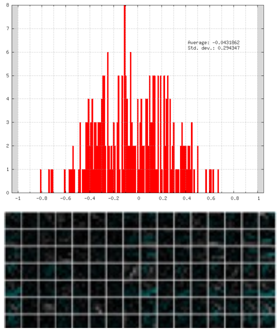


Results visualization:
- **Pixel-wise segmentation**
- **ROI bounding box extraction and classification**





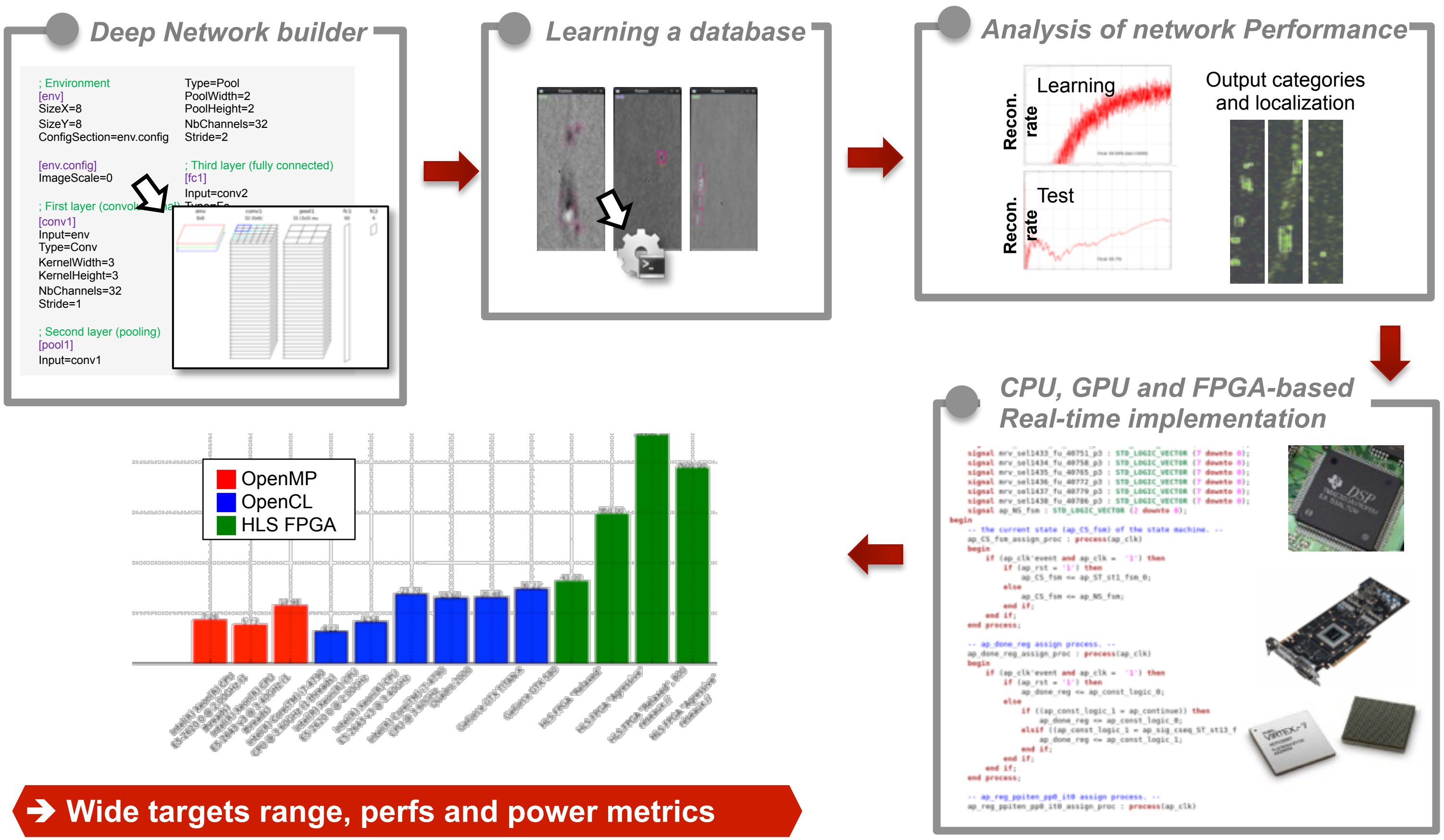
Layer-wise weights and kernels visualization, distribution and data-range analysis



Layer-wise output visualization and data-range analysis

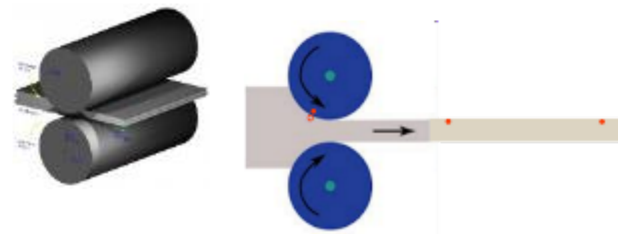
Pixel-wise and object wise confusion matrix reporting

FAST AND ACCURATE DNN EXPLORATION



➔ **Wide targets range, perfs and power metrics**

EXAMPLE OF INDUSTRIAL APPLICATION of N2D2: ROLLING MILL



CONSTRAINTS

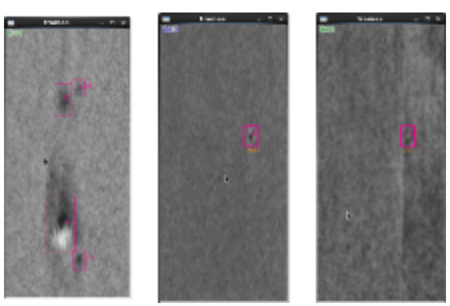
- Real time with very high throughput (20m/s)
- Tiny defect (~mm) with low contrast
- Complex environment (oil vapor, few space for inspection..)

SOLUTION

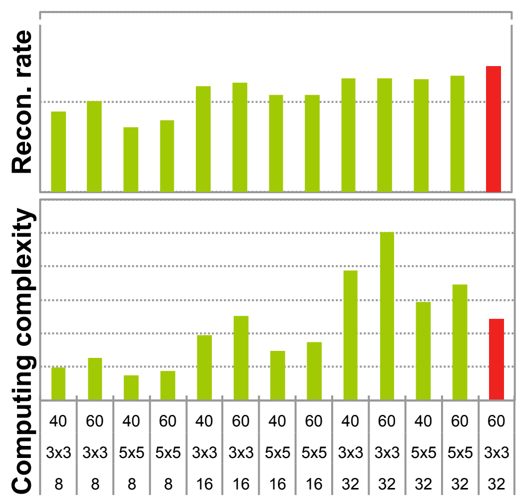
- Database labelling and Processing
- Fast NN topology Exploration
- Performance vs complexity analysis

➔ Real time performance achievable on FPGA (direct code generation)

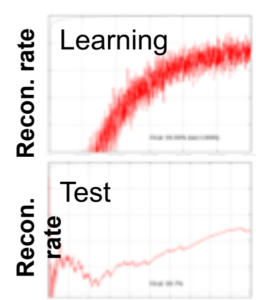
1) Defects labeling and visualization

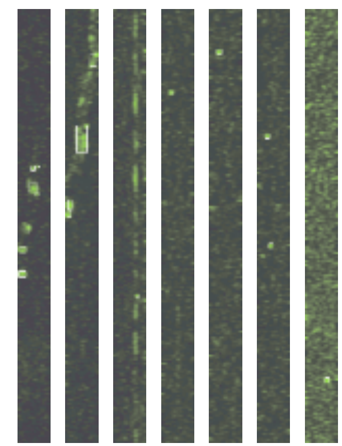


2) NN Exploration and benchmarking

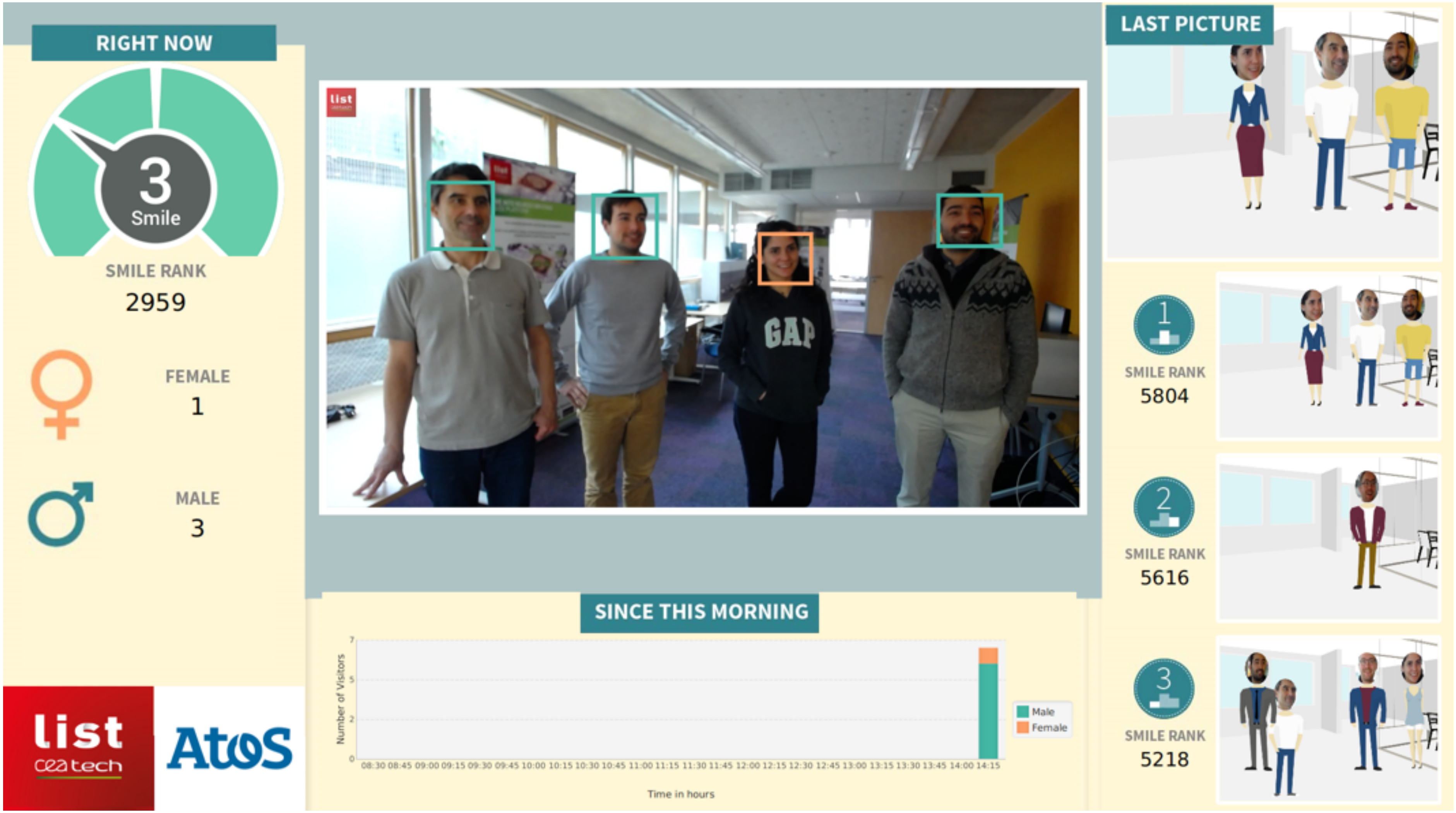


3) Defects identifications after NN learning





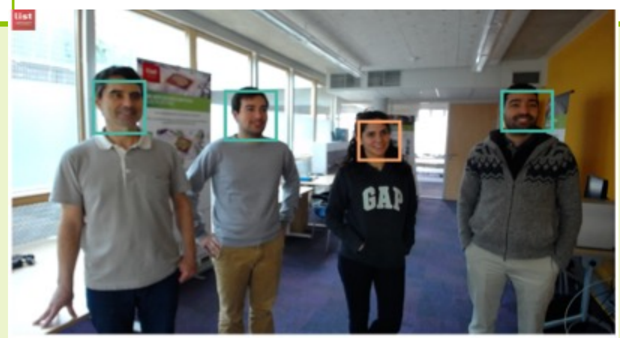
APPLICATION: REAL-TIME FACES DETECTION WITH GENDER & EMOTION

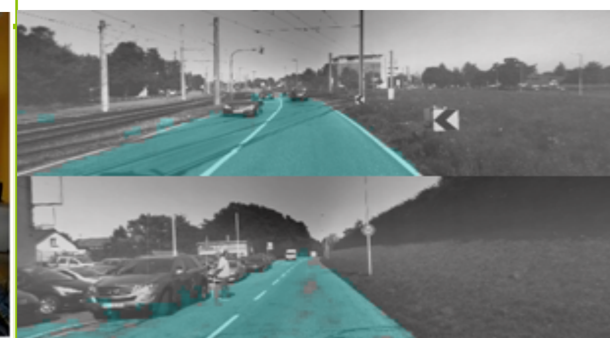


EXAMPLE OF USE OF N2D2

N2D2

AppObjectRecognition/	AppFaceDetection/	AppRoadDetection/
Live object recognition application *based on ILSVRC2012 (ImageNet) dataset*	**Live face detection application, with gender recognition** *based on the IMDB-WIKI dataset*	**Simple road segmentation application** *based on the KITTI Road dataset*



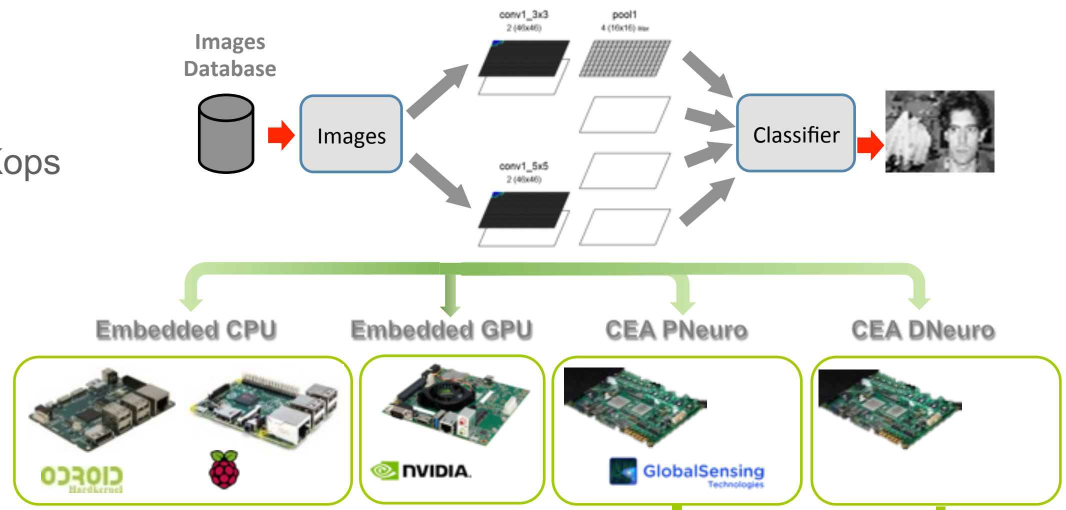


N2D2 is available at https://github.com/CEA-LIST/N2D2/

- Smallest dependencies and requirements among major frameworks:
  GCC 4.4 or Visual Studio 12 (2013) / OpenCV 2.0.0
- Easily extendable with a “plug-and-play” modular system for user-made modules

PNEURO ACCELERATOR BENCHMARKING

- **Benchmark application:**
  - Face extraction on a database of 18,000 images
  - 60 neurons on the hidden layer, 450 Kops
  - Recognition rate 97%
- **Optimized code for 5 architectures:**
  - Embedded CPU: Quad Arm A7 & A15
  - Embedded GPU: NVidia Tegra K1
  - PNeuro Quad Neuro-Cores / DNeuro



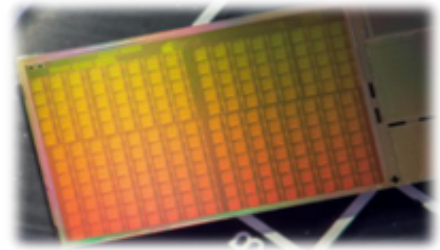
Target	Quad ARM A7 900 MHz	Quad ARM A15 2 GHz	Tegra K1 850 MHz	PNeuroV2 (FPGA) 100 MHz	PNeuroV2 (ASIC) 4 cores - 500 MHz	DNeuro (FPGA) 100 MHz
Performance	480 images/s	870 images/s	3 550 images/s	7 000 images/s	25 000 images/s	45 000 images/s
Energy Efficiency	380 images/s/W	350 images/s/W	600 images/s/W	2 800 images/s/W	125 000 images/s/W	18 000 images/s/W

- **PNeuro and DNeuro performance comparison vs Tegra K1 with N2D2:**

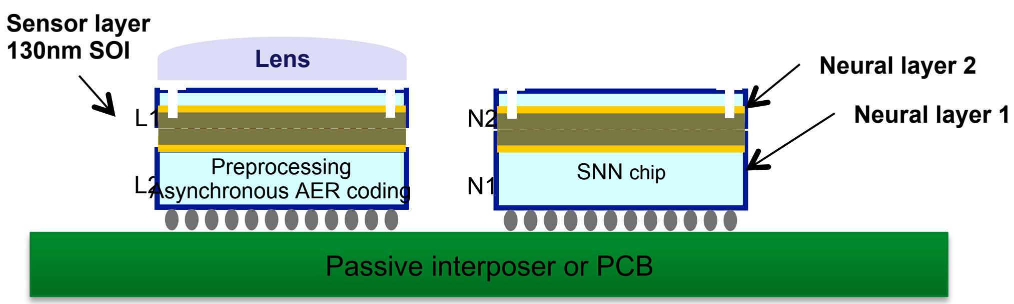
– Faster	x 2	x 7	x 12.5
– More Energy Efficient	x 4.5	x 200	x 30

3D STACKED RETINA WITH SPIKING NEURAL NETWORKS

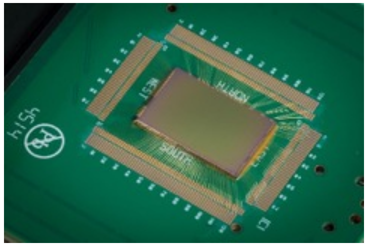
- **RETINE: image sensor + 3D stacked SIMD processors**
  - Image sensor: 70% fill factor, 12 µm pixel, >1000 fps
  - SIMD processors: 3072 units, distributed memory, 11.7 MOPS/mW
  - Feed SNN with Asynchronous Event Representation (AER) after pre-processing



Processor array die



Retine Chip
ALTIS 130nm, CuCu bonding

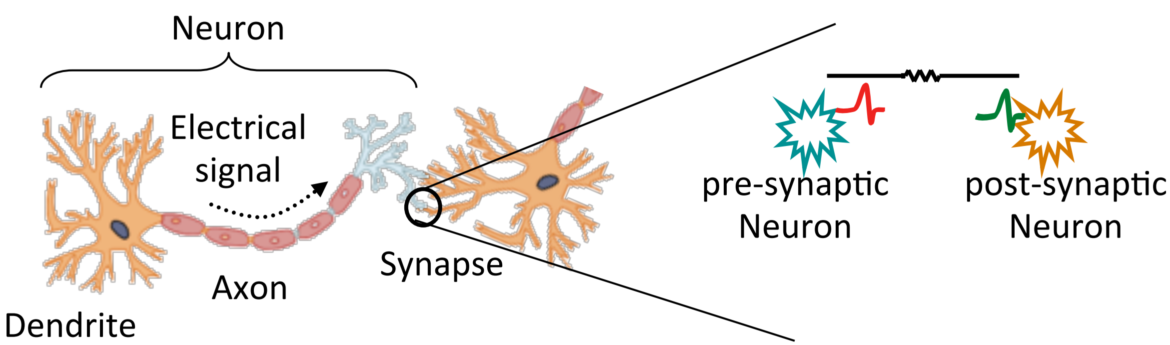


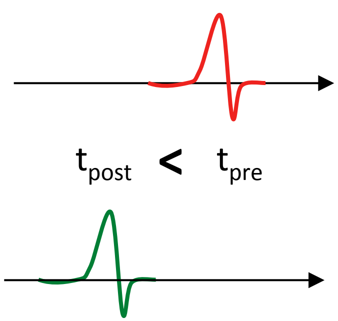
- **Pre-processing performances: (L1+L2 stacked retina)**

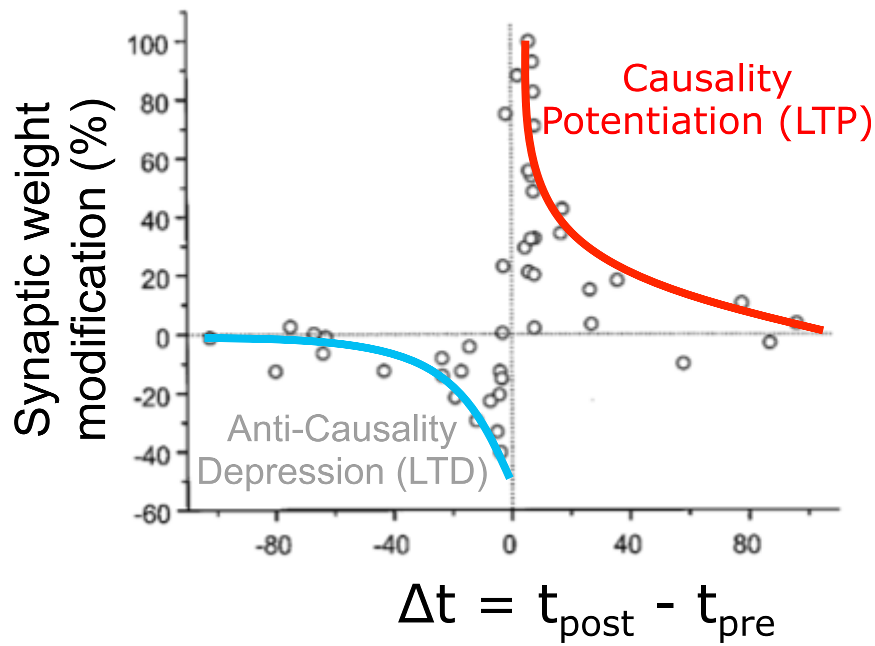
	RETINE	ARM cortex A9 +NEON	STxP70
Frequency (Mhz)	150	400	350
Performance (GOPS)	**72**	0,67	0,28
Power consumption (W)	4,8	0,25	0,08
Energy / frame (mJ)	2,74	0,68	5,6
Energy efficiency (normalized, GOPS/W)	**45**	2,68	5,25

➔ x100 computing power, x10 energy efficiency, /15 processing latency

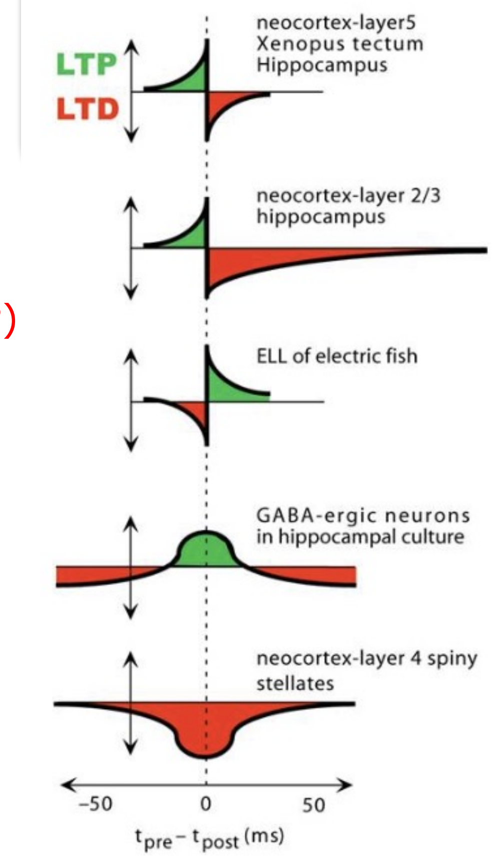
DERIVED FROM HEBB'S RULE: STDP (SPIKE TIMING DEPENDENT PLASTICITY)



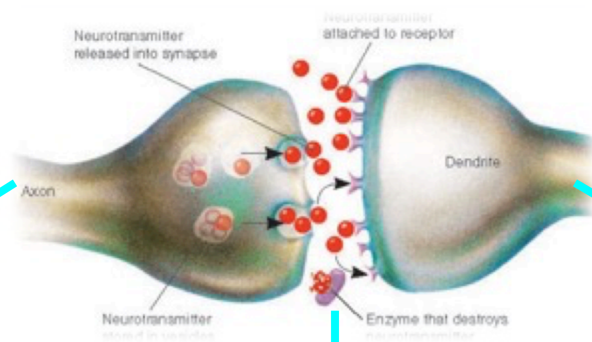




STDP = correlation detector



NEW ELEMENT: RRAM AS SYNAPSES



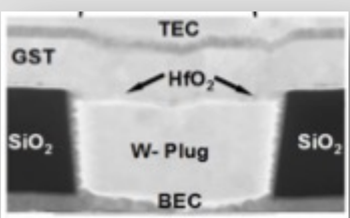
Thermal effect

Electrochemical effect

Electronic effect oxygen vacancies

PCM

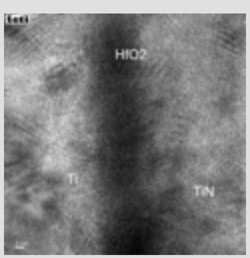
GST
GeTe
GST + HfO_2



M.Suri, et. al, IEDM 2011
M.Suri, et. al, IMW 2012 , JAP 2012
O.Bichler et al. IEEE TED 2012
M.Suri et al., EPCOS 2013
D.Garbin et al., IEEE Nano 2013

OXRAM

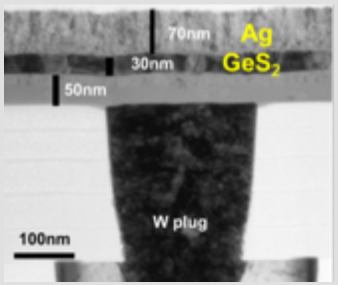
$TiN/HfO_2/Ti/TiN$



D.Garbin et al. IEDM 2014
D.Garbin et al., IEEE TED 2015

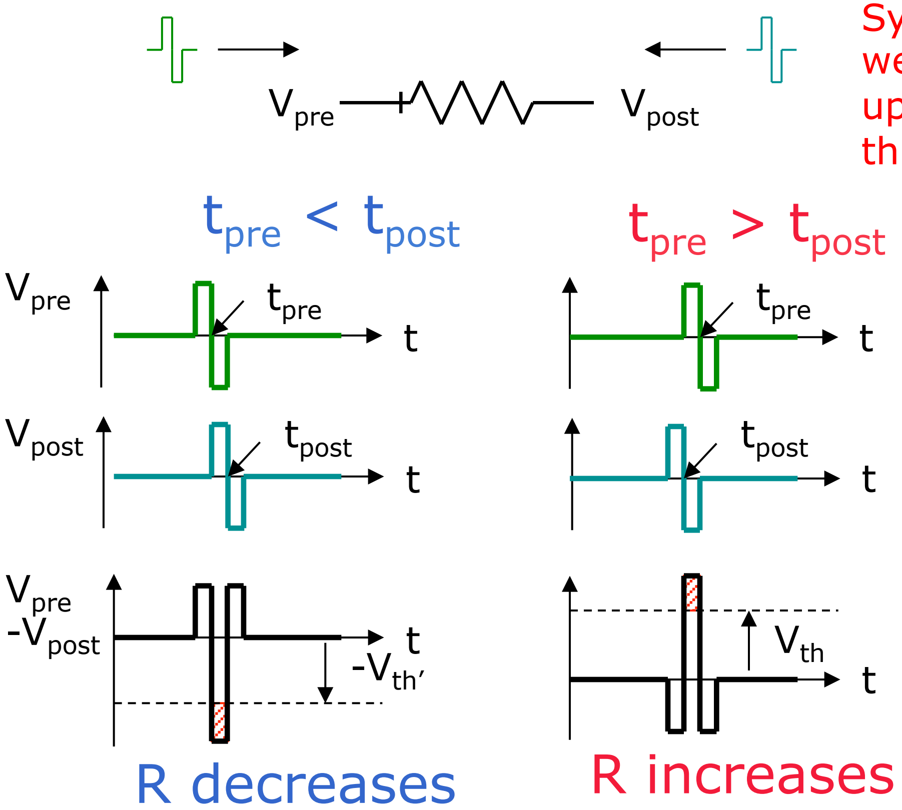
CBRAM

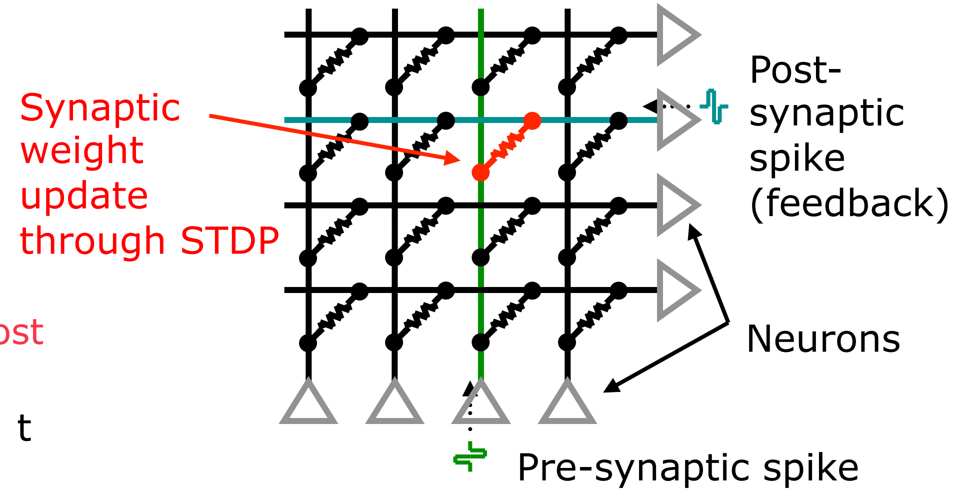
Ag / GeS_2

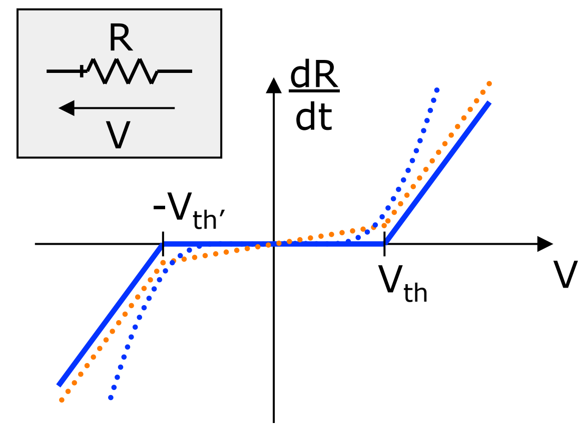


PRINCIPLE CROSSBARS OF MEMRISTORS

First Proposed by Snider(1)



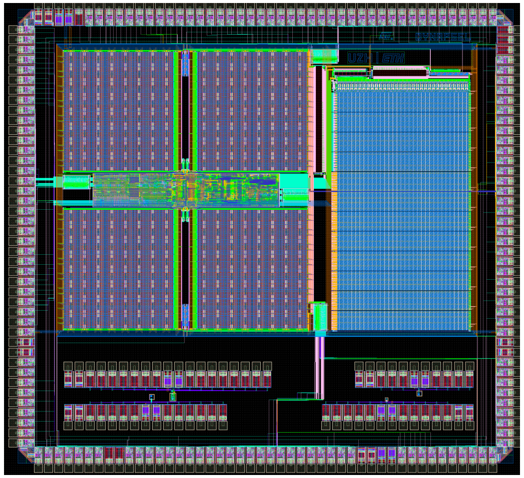




1. G. Snider, *Nanoscale Architectures*, 2008
2. B. Linares-Barranco et al, *Nature Precedings*, 2009

1^{ST} DIGITAL CHIP ARRIVED IN SUMMER 2017

	Neuram3 1st chip	IBM True North
Technology	28 nm FDSOI	28nm CMOS
Supply Voltage	1 V	0.7V
Neuron Type	Analog	Digital
Neurons per core	256	256
Core Area	0.36 mm^2	0.094 mm^2
Computation	Parallel processing	Time multiplexing
Fan In/Out	2k/8k	256/256
Synaptic Operation per Second per Watt	**300 GSOPS/W**[*1]	46 GSOPS/W
Energy per synaptic event	**<2 pJ**[*2]	10 pJ
Energy per spike	<0.375 nJ[*3]	3.9 nJ



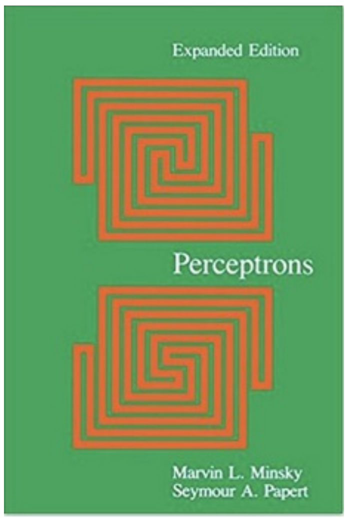
∗ 1 At 100Hz mean firing rate, by appending 4 local-core destinations per spike, 400 k events will be broadcast to 4 cores with 25% connectivity per event. 400 k x 1 k x 25% / 300 μ W = 300 GSOPS/W

∗ 2 In case of 25% match in each core, energy per synaptic event = energy per broadcast / (256*25%) =120pJ/64 = 2 pJ

∗ 3 Energy per spike = total power consumption / spikes numbers = 300 uW/800 k = 0.375 nJ

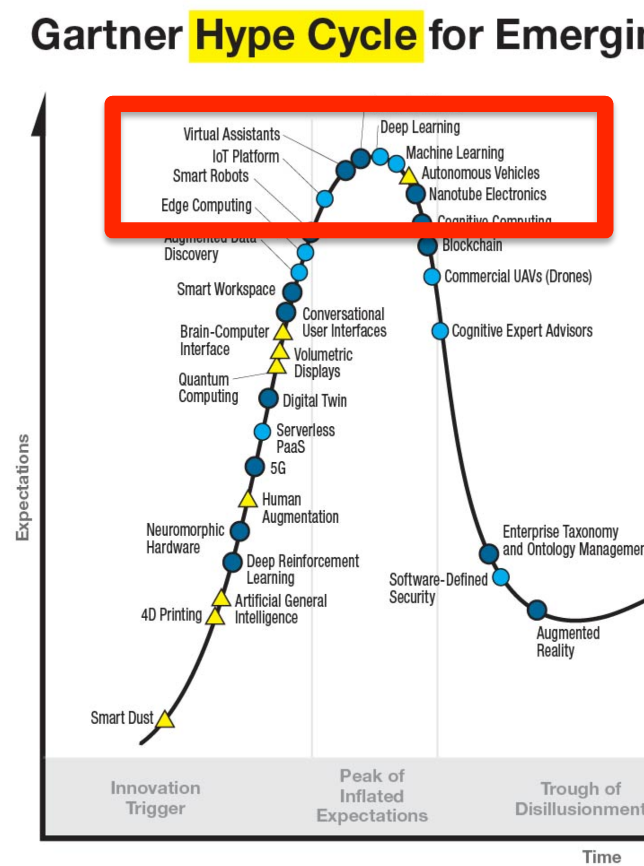
WHAT'S NEXT FOR DEEP LEANING AND AI?

1st Winter: 1987
Perceptrons
Minsky & Papert



2st Winter: 1993
SVM
Vapnik & Cortes (1963)

3rd Winter or
Plateau of Productivity?



- **Deep Learning**
- **Machine learning**
- **Autonomous vehicles**
- **Virtual assistants**
- **Smart robots**
- **Edge computing**
- **IoT platforms**
- **Connected home**

gartner.com/SmarterWithGartner

Source: Gartner (July 2017)

Thank you for your attention

Special thank you to Olivier Bichler, Christian Gamrat
And Yann LeCun for their slides I borrowed.

marc.duranton@cea.fr

INSTITUT CARNOT CEA LETI

INSTITUT CARNOT CEA LIST

MINATEC

digiteo

leti

Centre de Grenoble
17 rue des Martyrs
38054 Grenoble Cedex

list

Centre de Saclay
Nano-Innov PC 172
91191 Gif sur Yvette Cedex