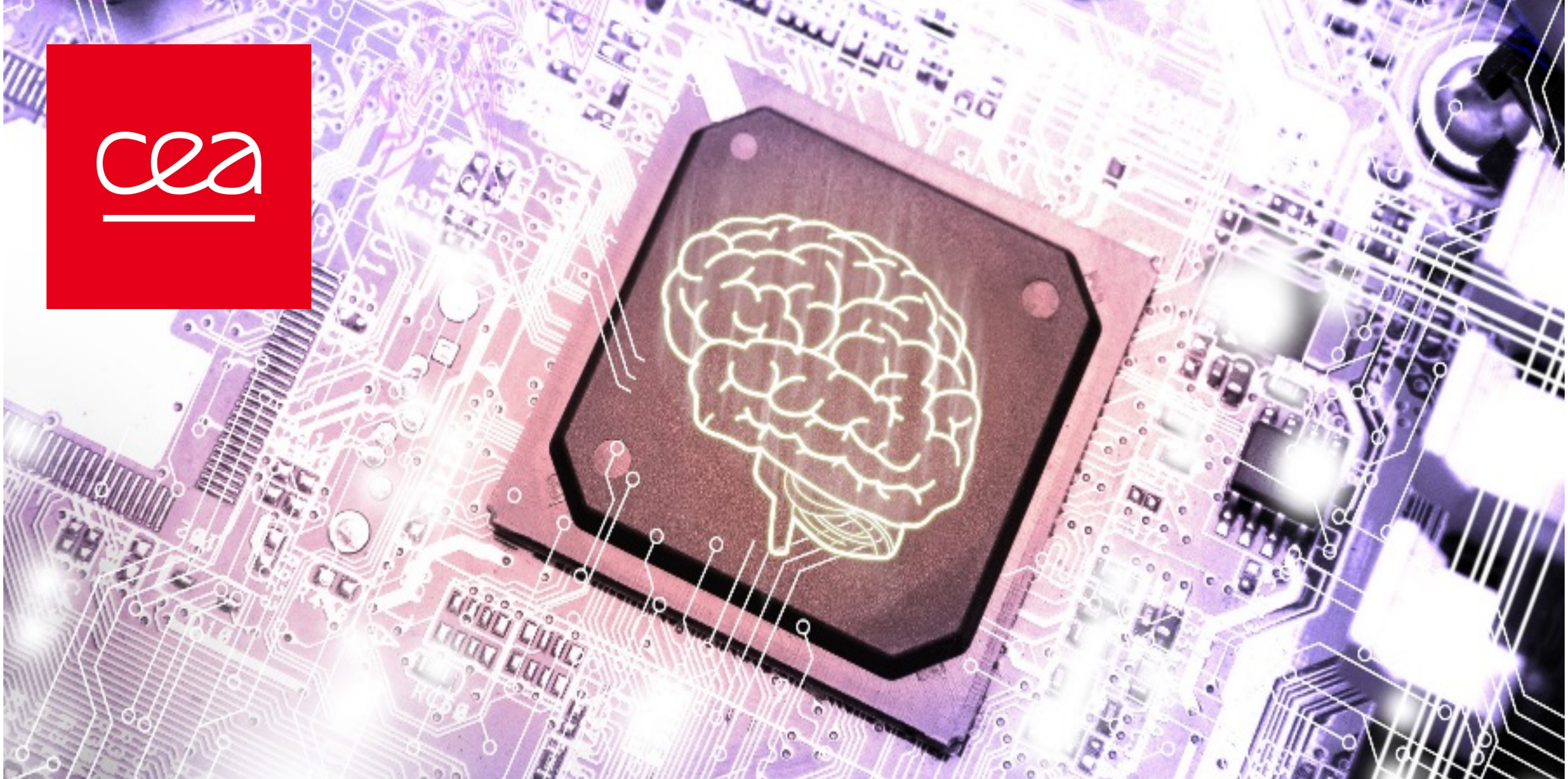


A SHORT STORY OR ARTIFICIAL INTELLIGENCE AND DEEP LEARNING

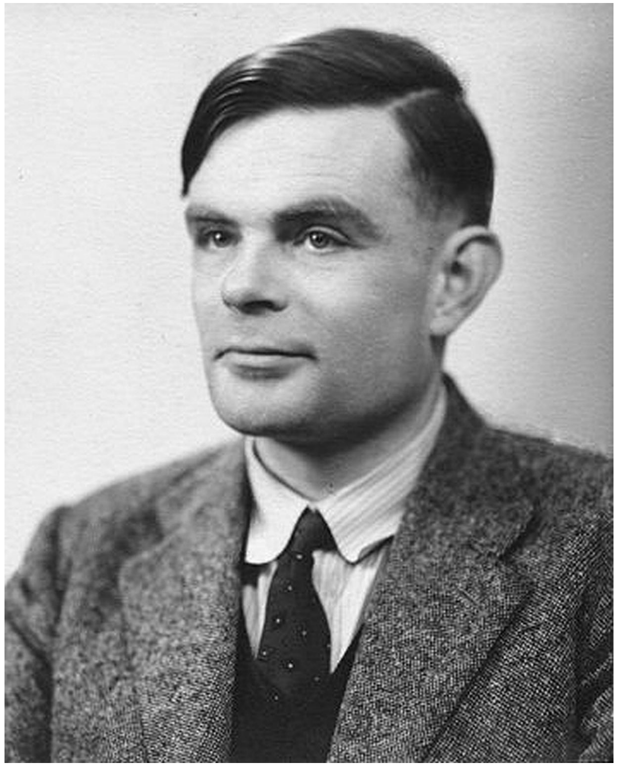
Part 1: From origins to perceptive AI

Marc Duranton
Commissariat à l'énergie atomique et aux énergies alternatives

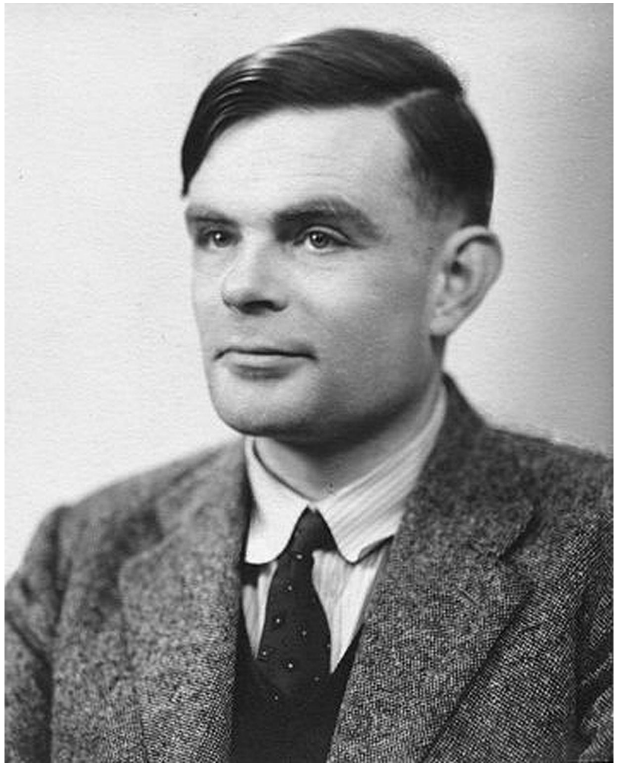
June 4th, 2025

1942: ALAN TURING

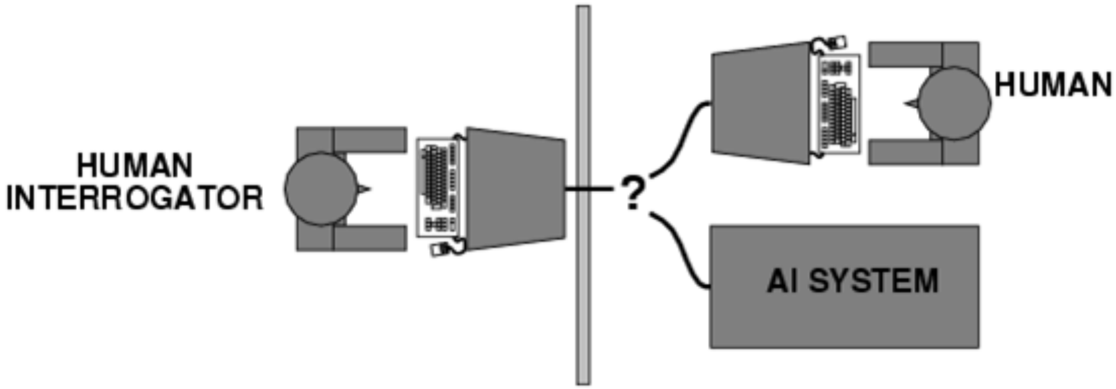
1942: Any form of mathematical reasoning can be made by a machine.



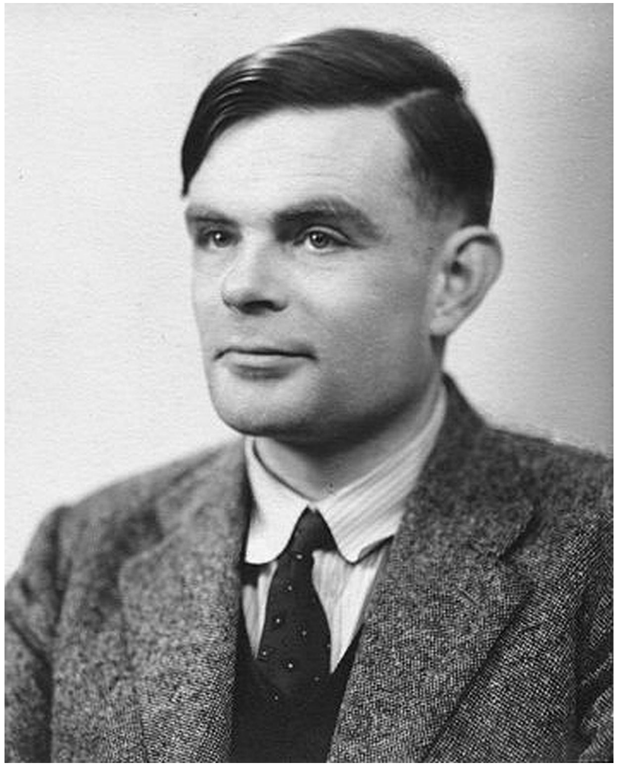
1942: ALAN TURING



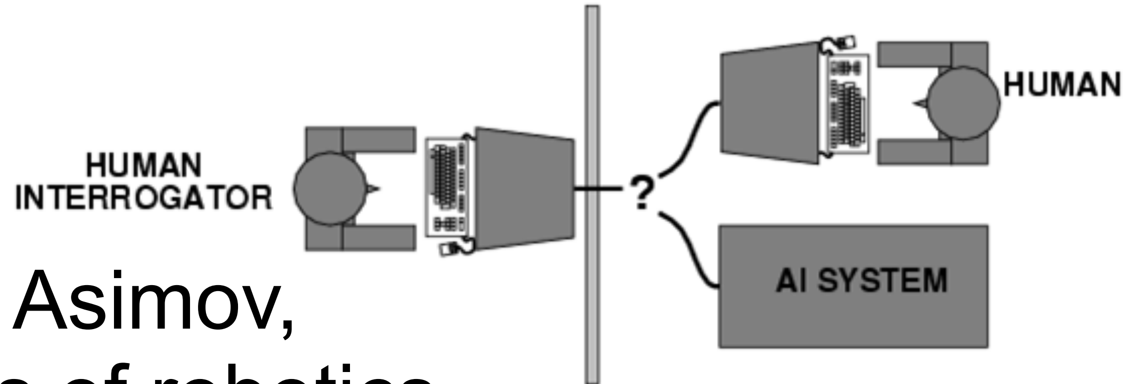
1942: Any form of mathematical reasoning can be made by a machine.
1950: He invented the “Turing test” to check if a system is “intelligent”, i.e. undisguisable from a human



1942: ALAN TURING

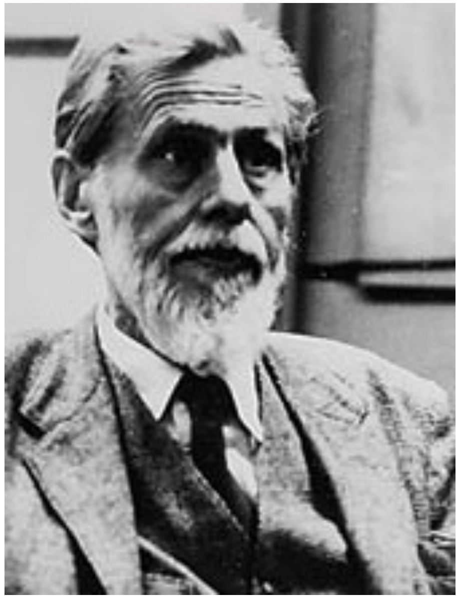


1942: Any form of mathematical reasoning can be made by a machine.
1950: He invented the "Turing test" to check if a system is "intelligent", i.e. undisguisable from a human

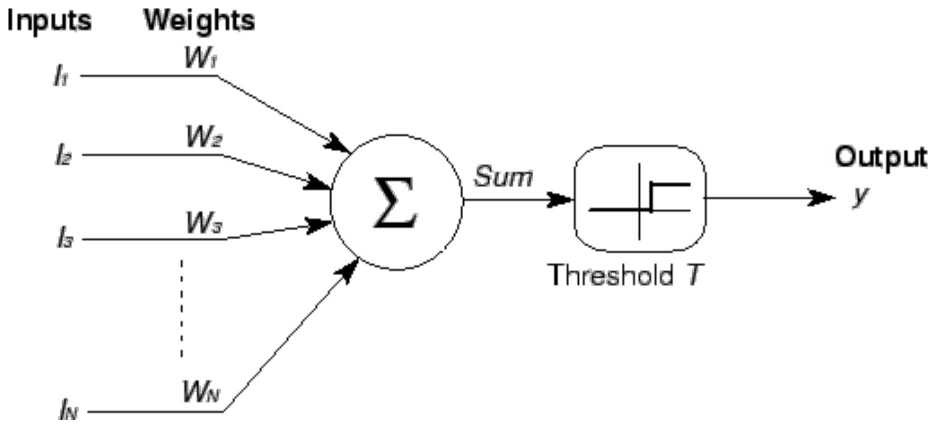


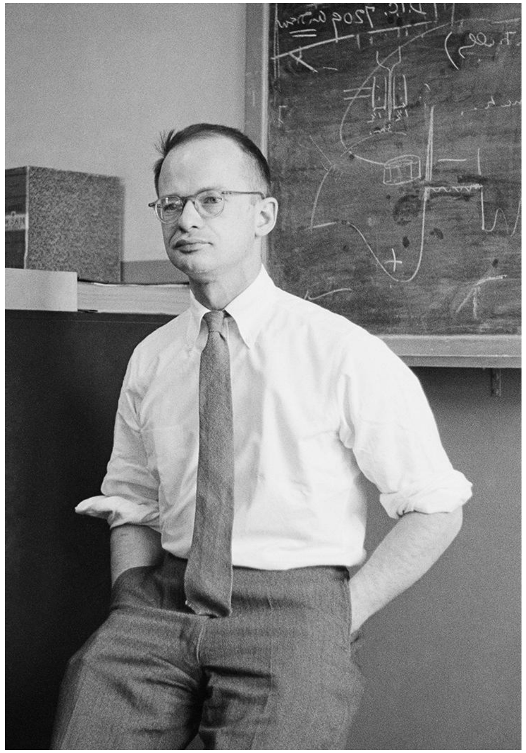
The same year, Isaac Asimov, invented the 3 (4) laws of robotics

1943: MCCULLOCH AND PITTS



Neurophysiologist and cybernetician



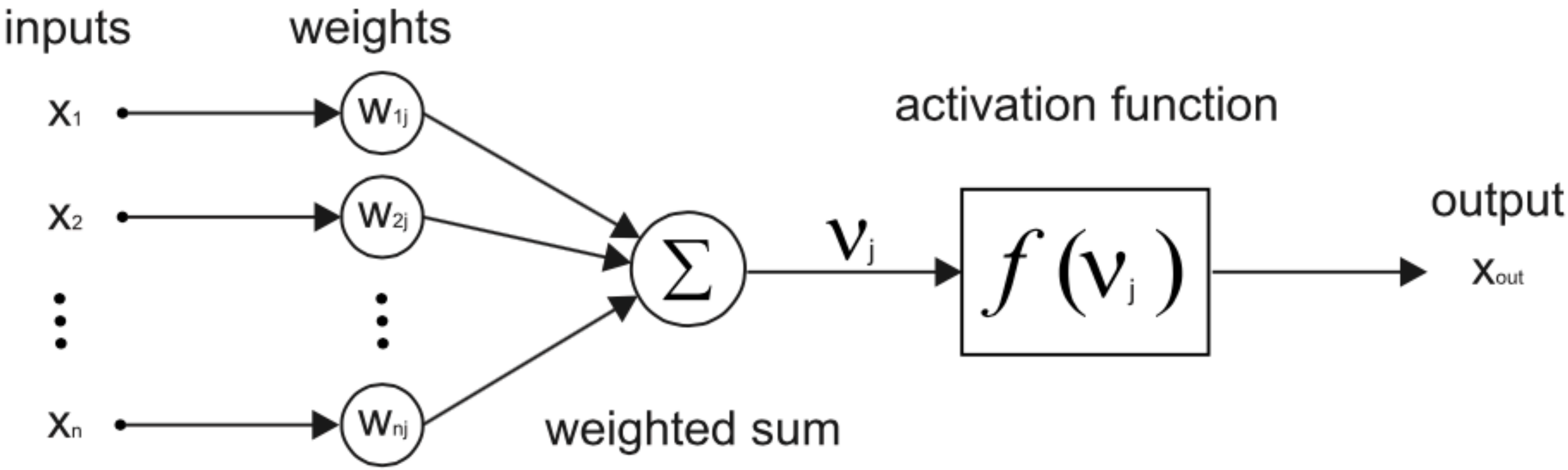


Logician workingin the field of computational neuroscience

They laid the foundations of formal Neural Networks

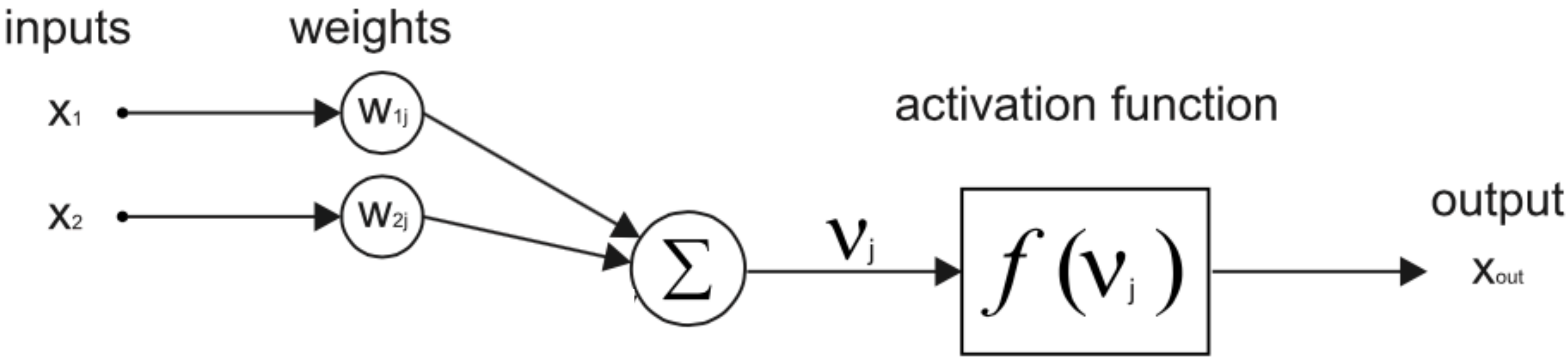
WHAT IS A NEURAL NETWORK?

A « formal » neuron:



WHAT IS A NEURAL NETWORK?

The « formal » neuron:



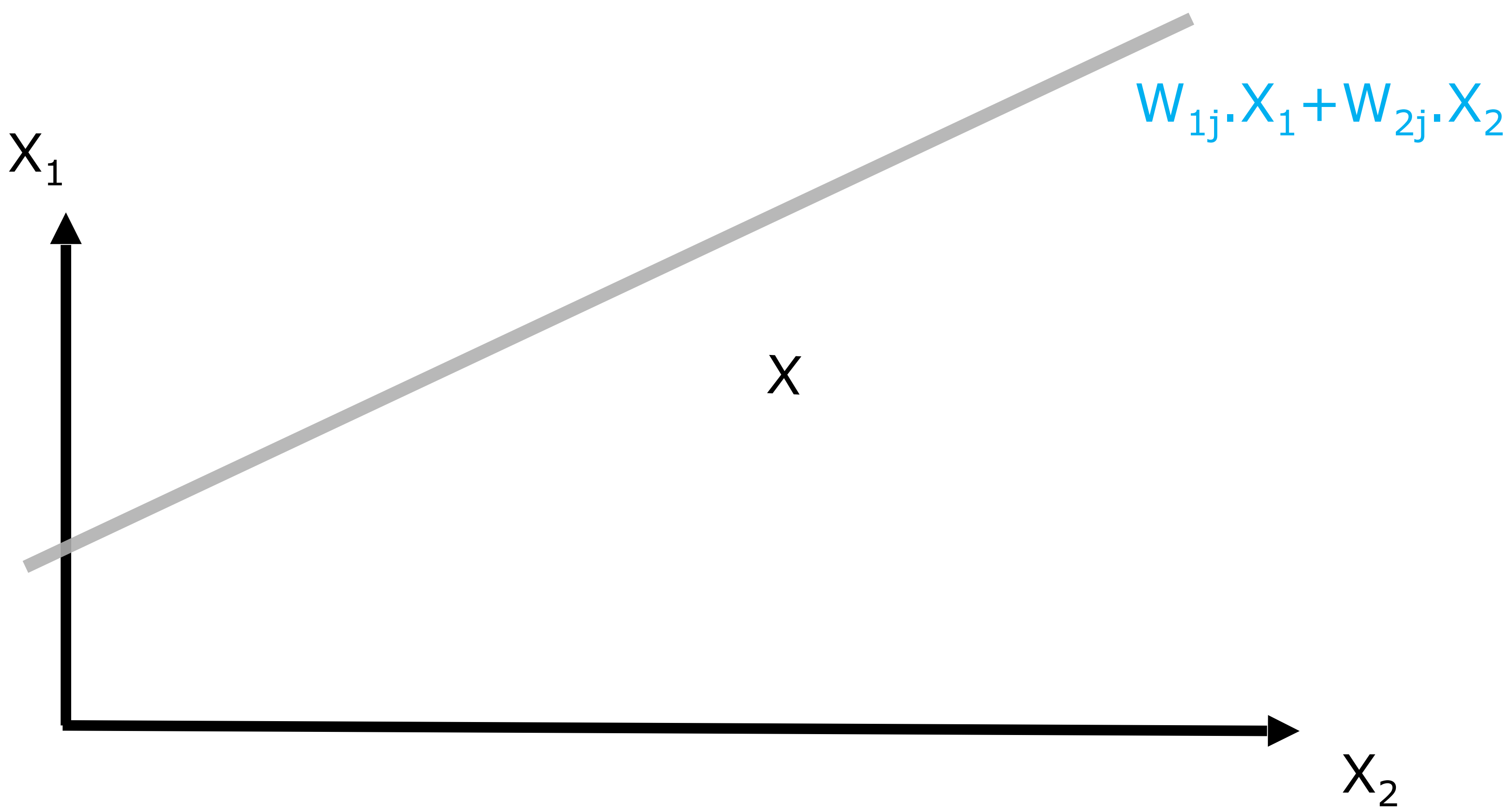
$$V_j = W_{1j}.X_1 + W_{2j}.X_2$$

It is the definition of an hyperplane

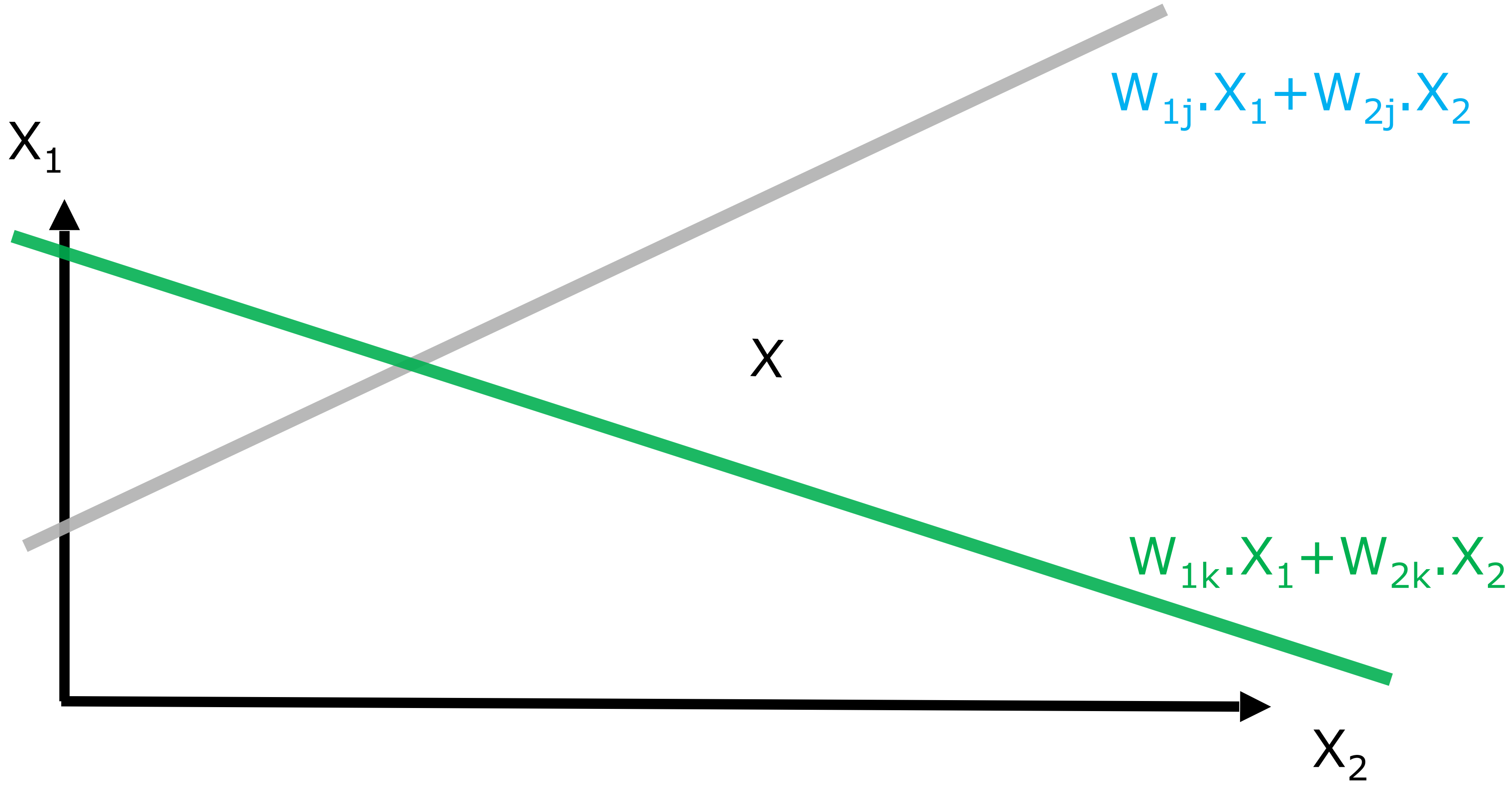
$F(V_j)$ non linear $\in\{-1,1\}$ e.g. sign() function

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

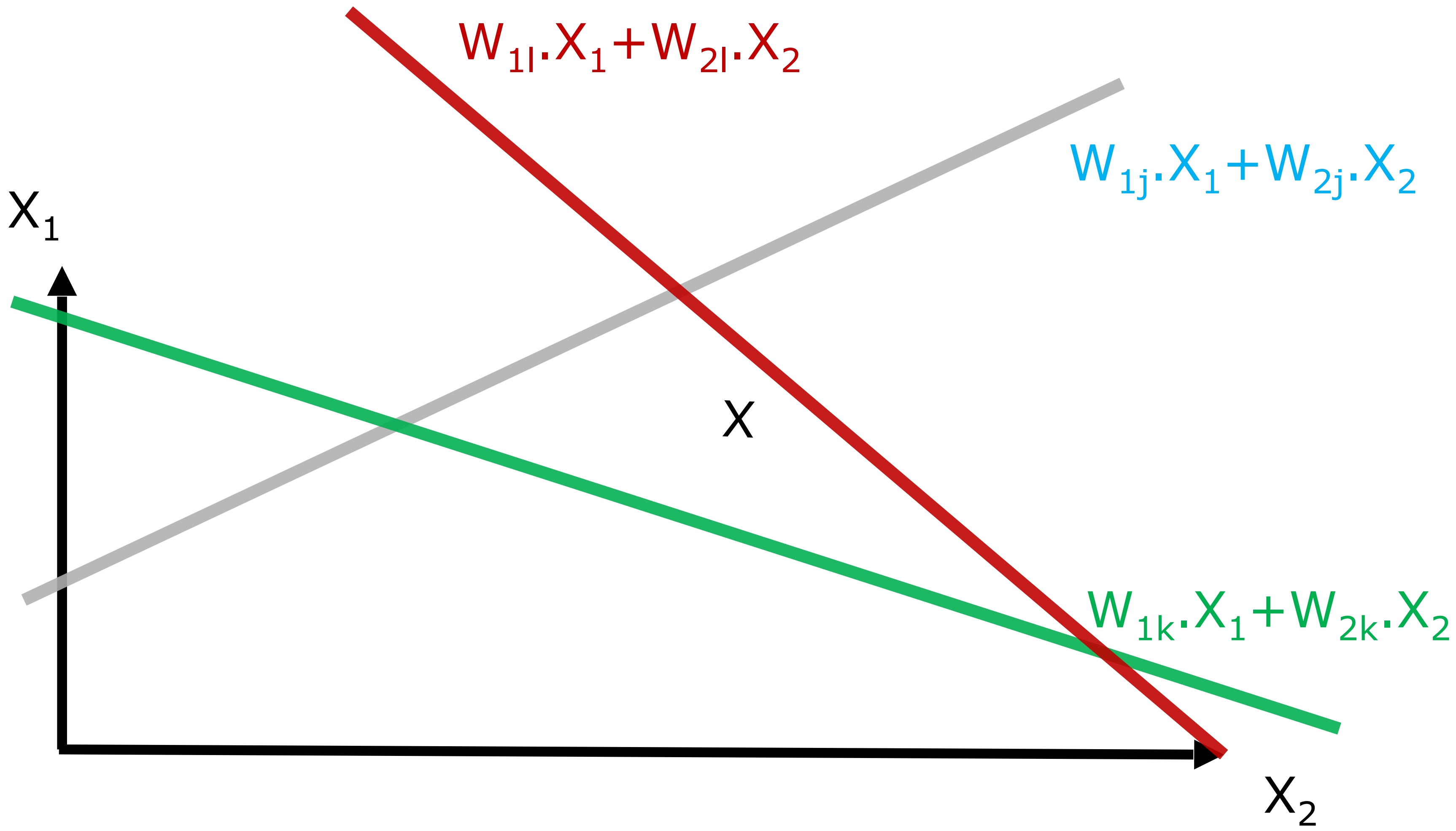
WHAT IS A NEURAL NETWORK?



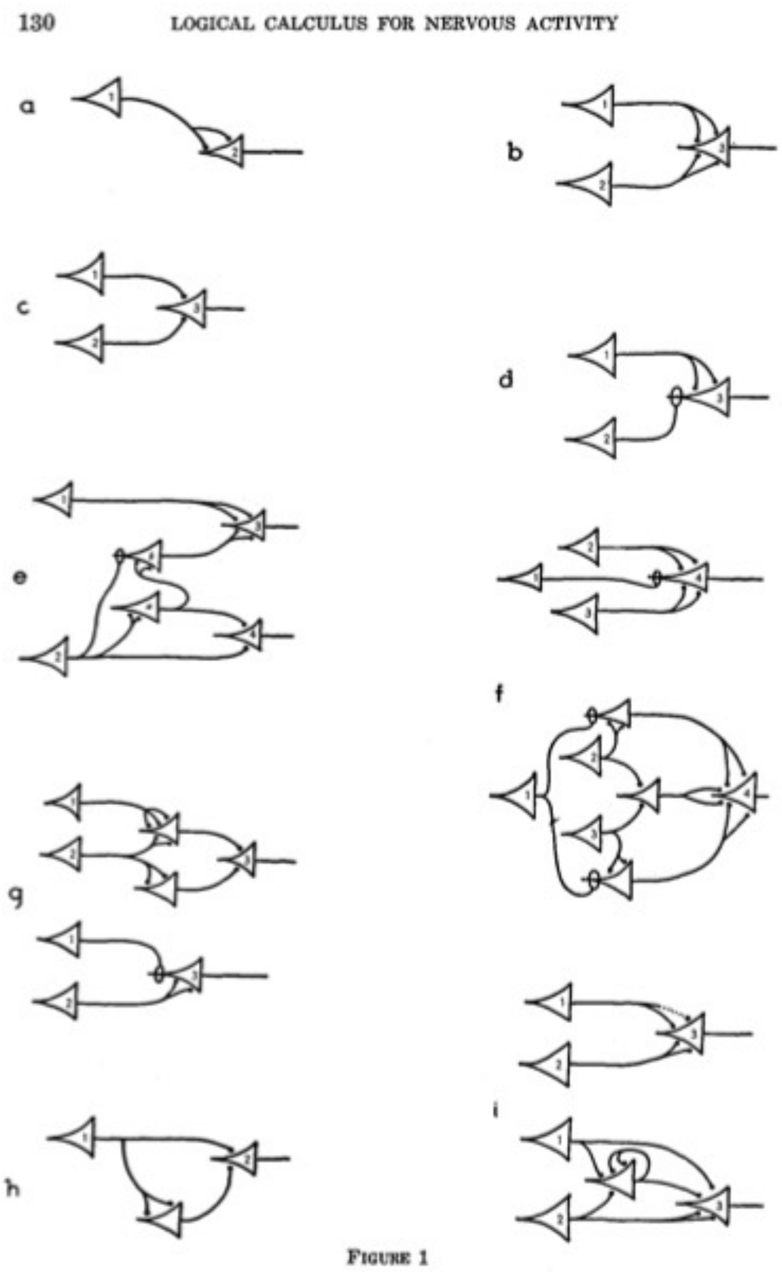
WHAT IS A NEURAL NETWORK?



WHAT IS A NEURAL NETWORK?

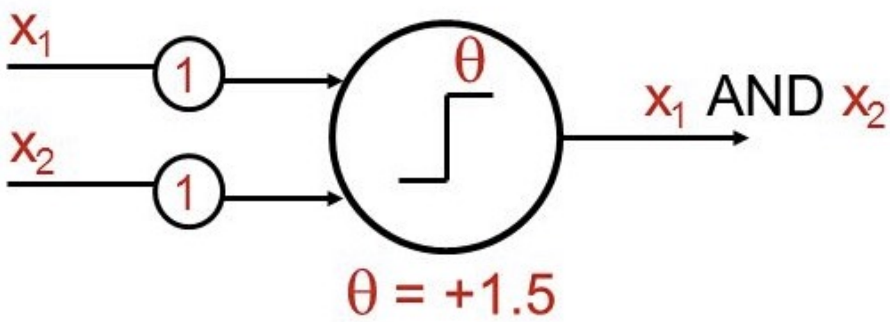


WHAT IS A NEURAL NETWORK?

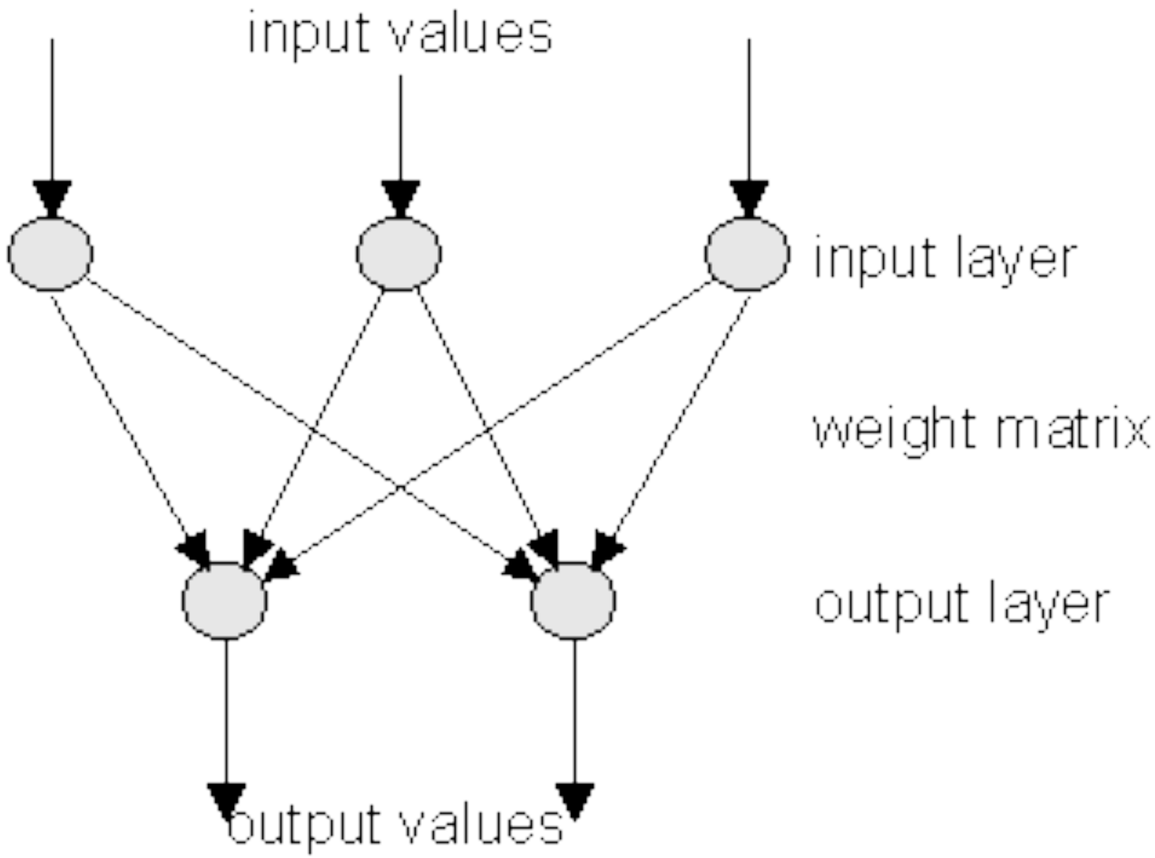


Association of neurons to make logical functions.
Example: AND gate

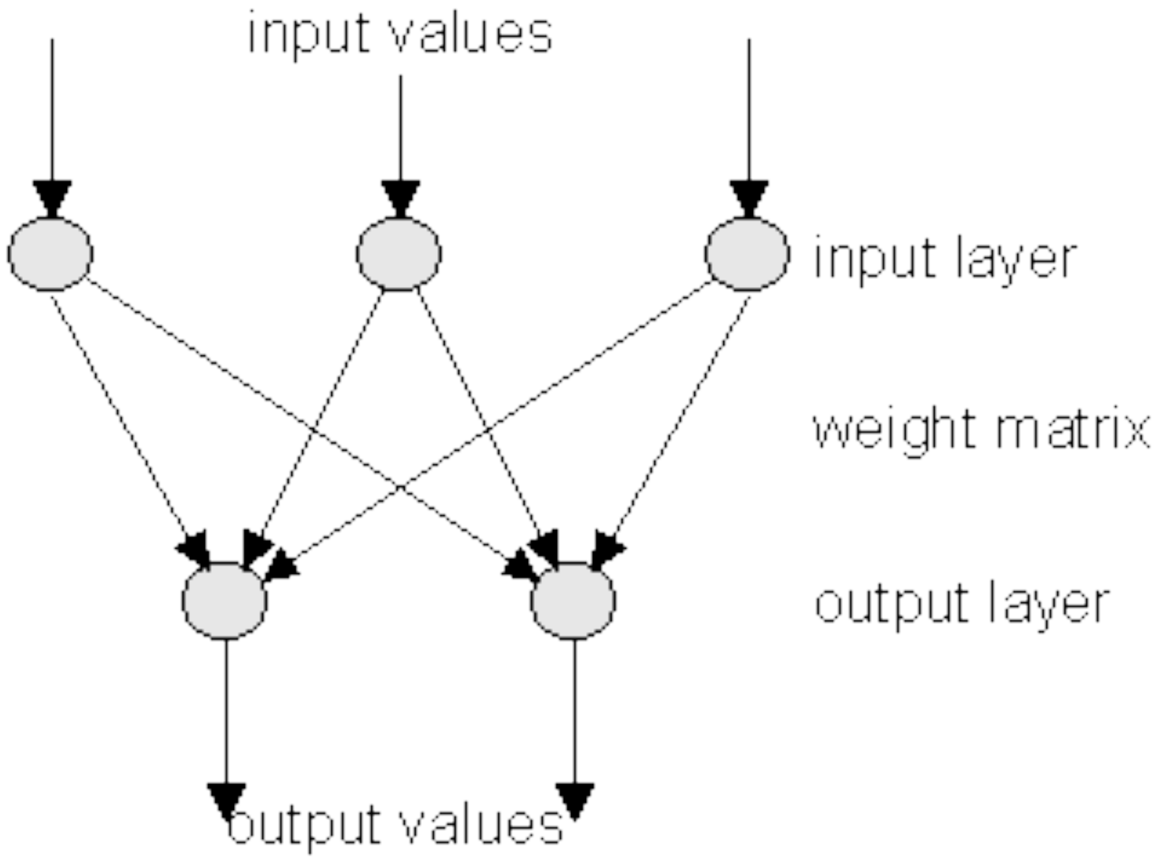
IN 1	IN 2	OUT
0	0	0
0	1	0
1	0	0
1	1	1



MULTILAYER NETWORK

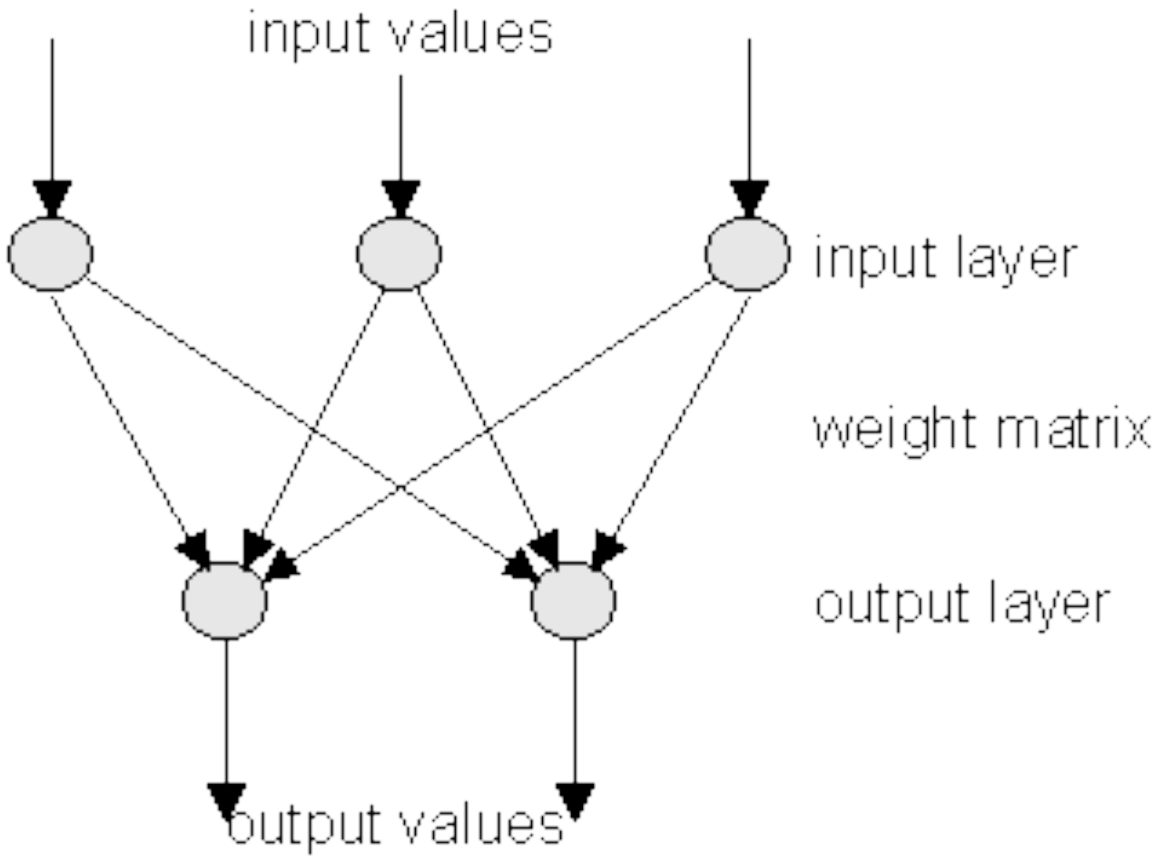


MULTILAYER NETWORK



Hyperplane separation

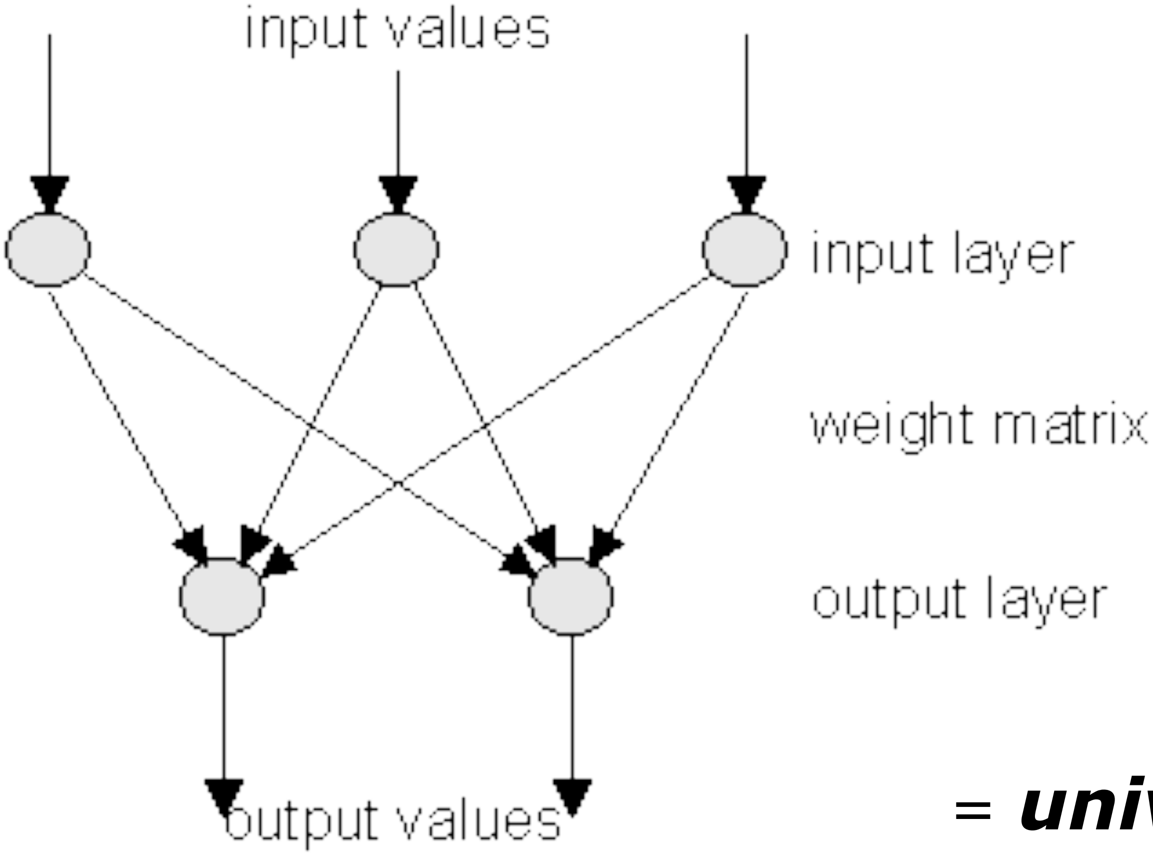
MULTILAYER NETWORK



Hyperplane separation

“logic” composition
Warren McCulloch and
Walter Pitts, 1943

MULTILAYER NETWORK



Hyperplane separation

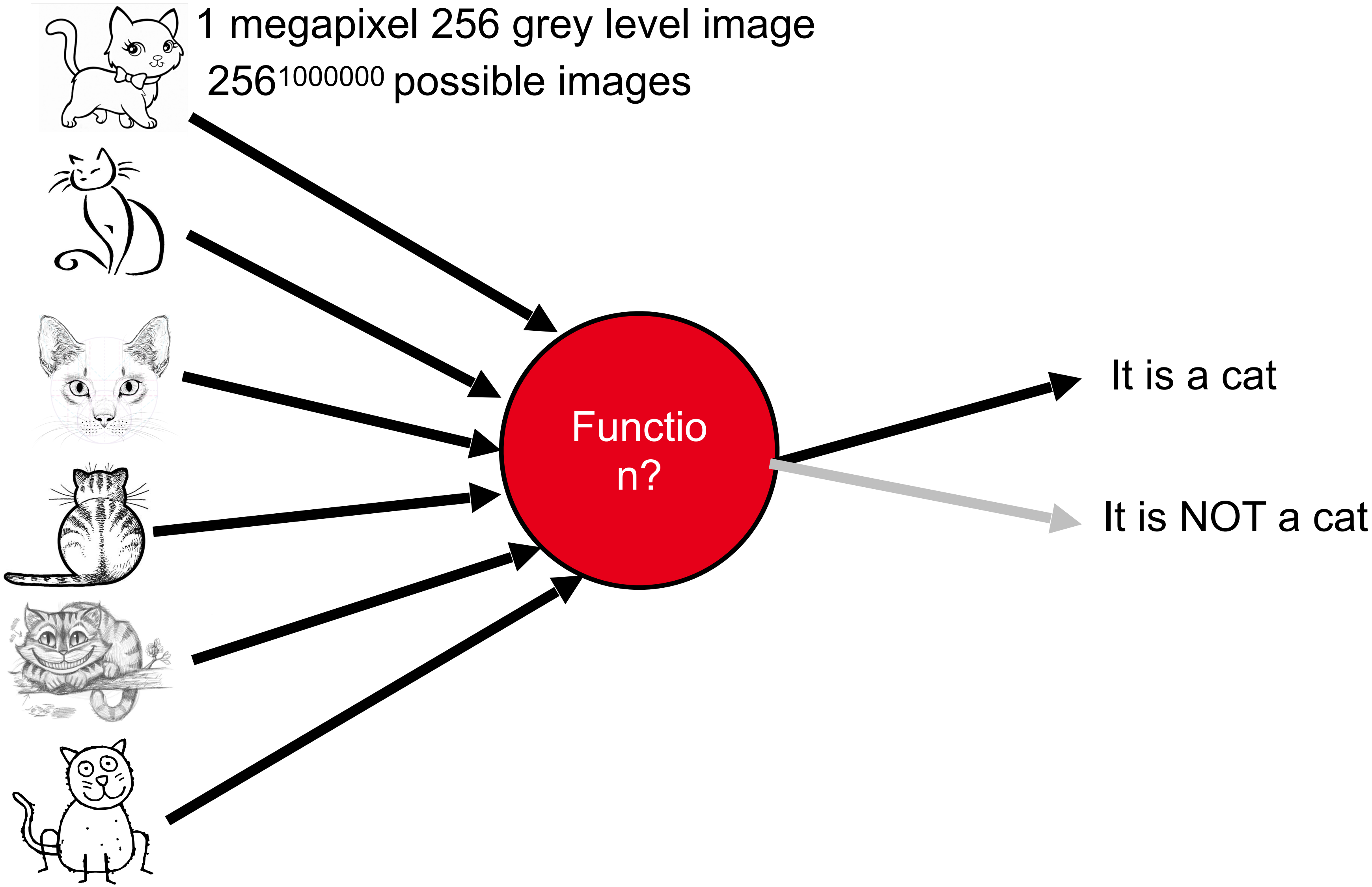
"logic" composition
Warren McCulloch and
Walter Pitts, 1943

= ***universal approximator***

WHY DOES DEEP LEARNING WORK SO WELL?*

1 megapixel 256 grey level image

$256^{1000000}$ possible images

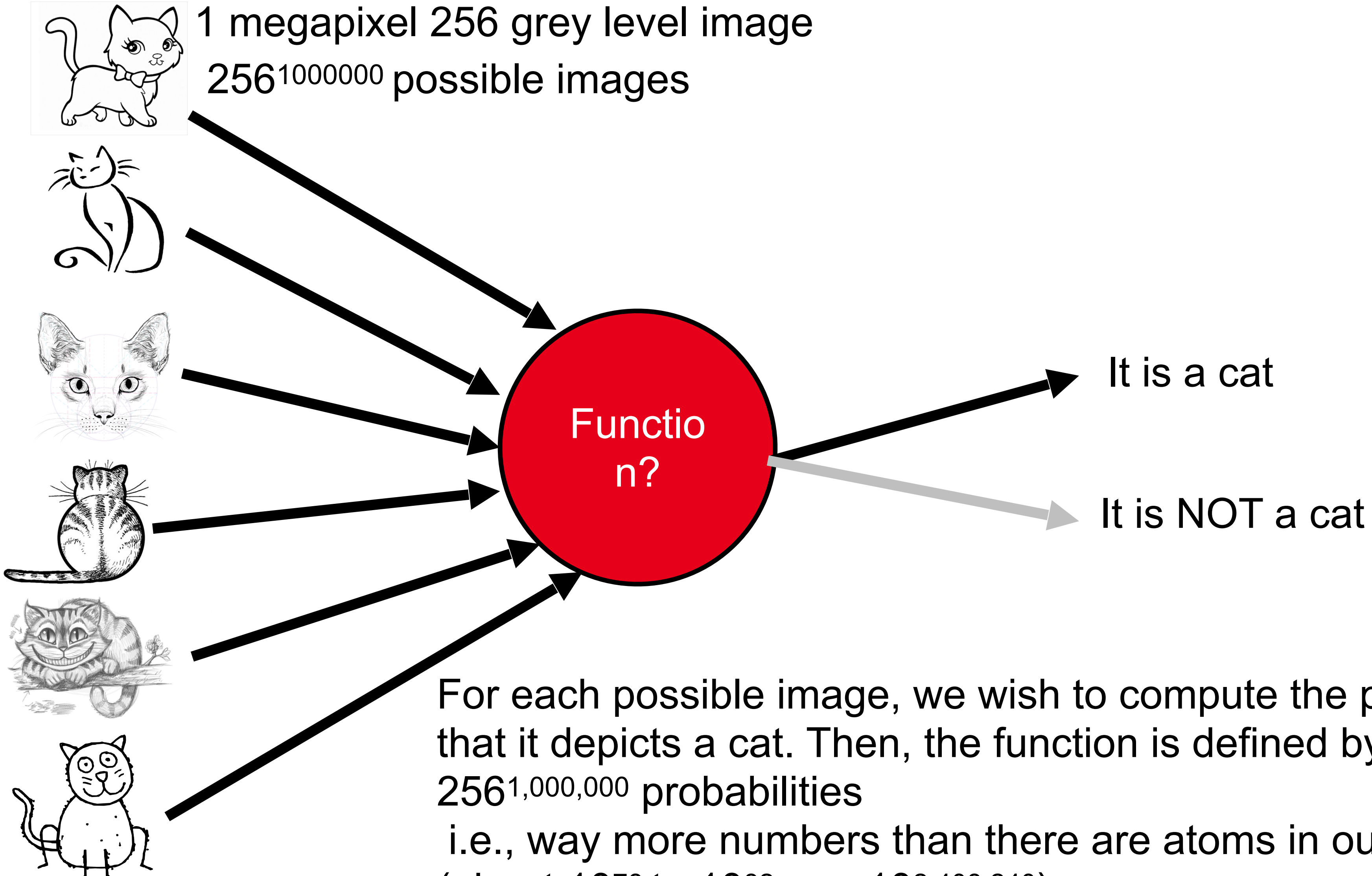


- Work of Henry W. Lin (Harward) , Max Tegmark (MIT), and David Rolnick (MIT) https://arxiv.org/abs/1608.08225

WHY DOES DEEP LEARNING WORK SO WELL?*

1 megapixel 256 grey level image

$256^{1000000}$ possible images



For each possible image, we wish to compute the probability that it depicts a cat. Then, the function is defined by a list of $256^{1,000,000}$ probabilities

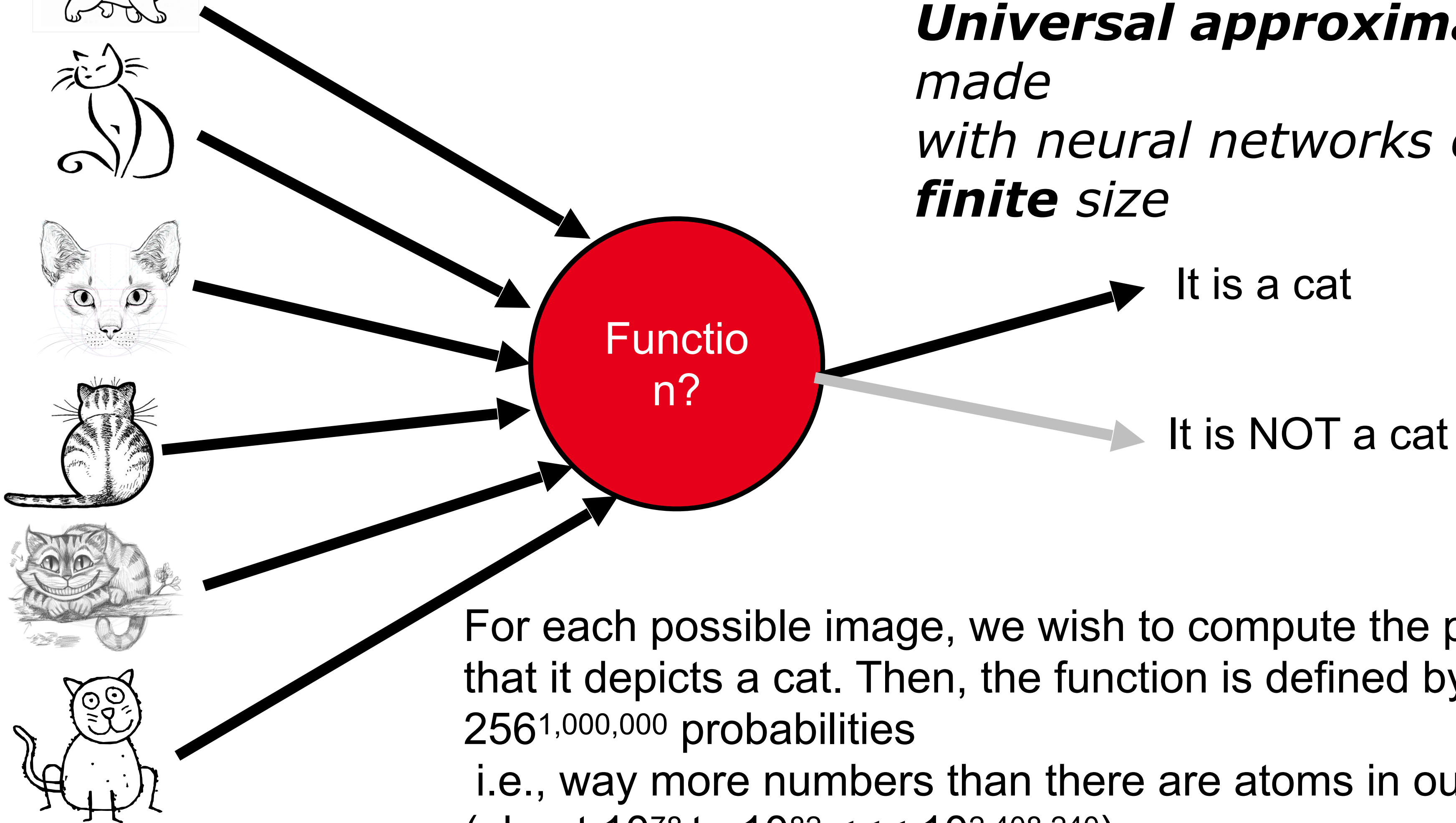
i.e., way more numbers than there are atoms in our universe (about 10^{78} to 10^{82} <<< $10^{2,408,240}$).

- 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?*

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 $256^{1,000,000}$ probabilities
i.e., way more numbers than there are atoms in our universe (about 10^{78} to 10^{82} <<< $10^{2,408,240}$).

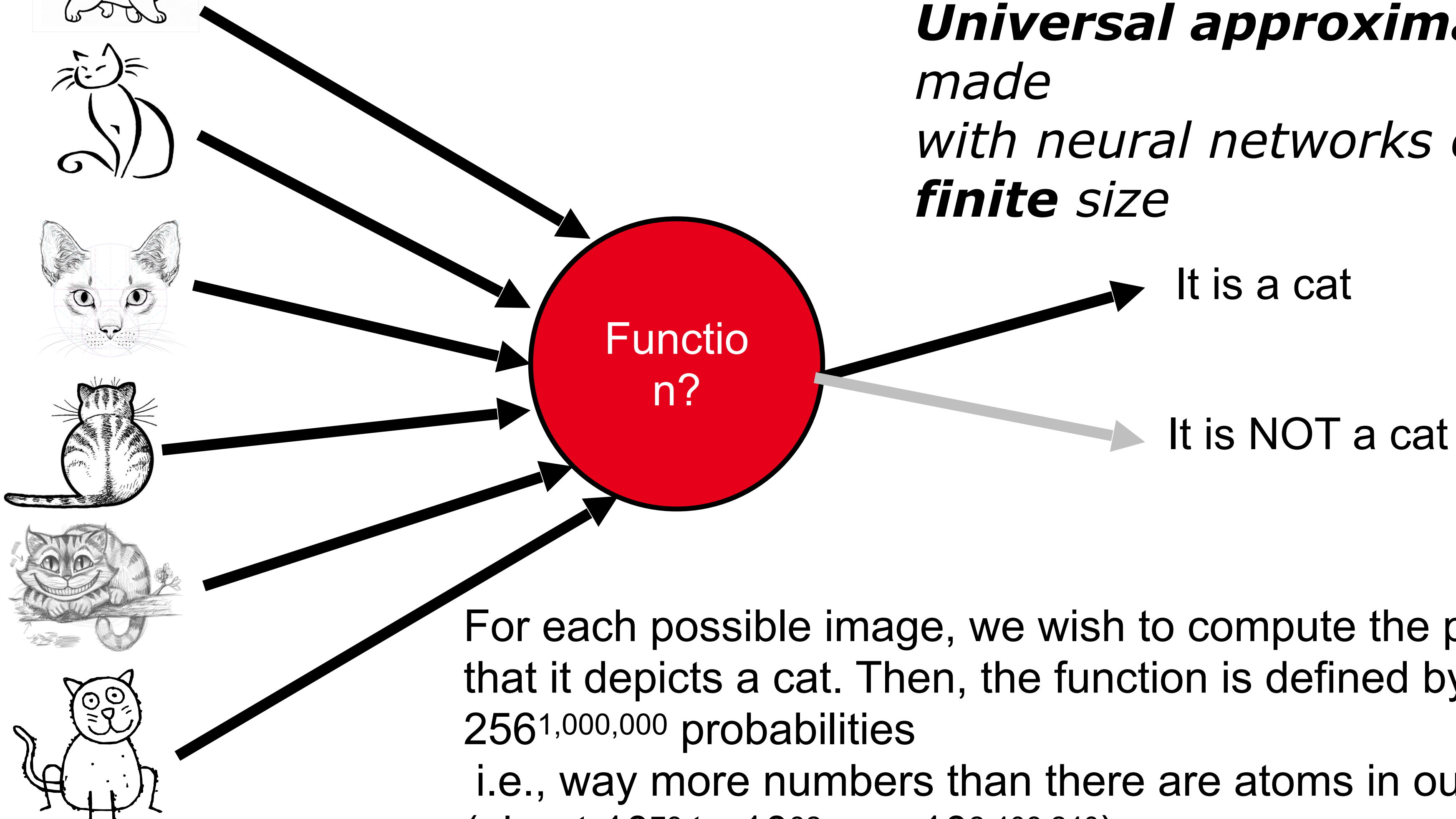
- Work of Henry W. Lin (Harward) , Max Tegmark (MIT), and David Rolnick (MIT) https://arxiv.org/abs/1608.08225

WHY DOES DEEP LEARNING WORK SO WELL?*

1 megapixel 256 grey level image

$256^{1000000}$ possible images

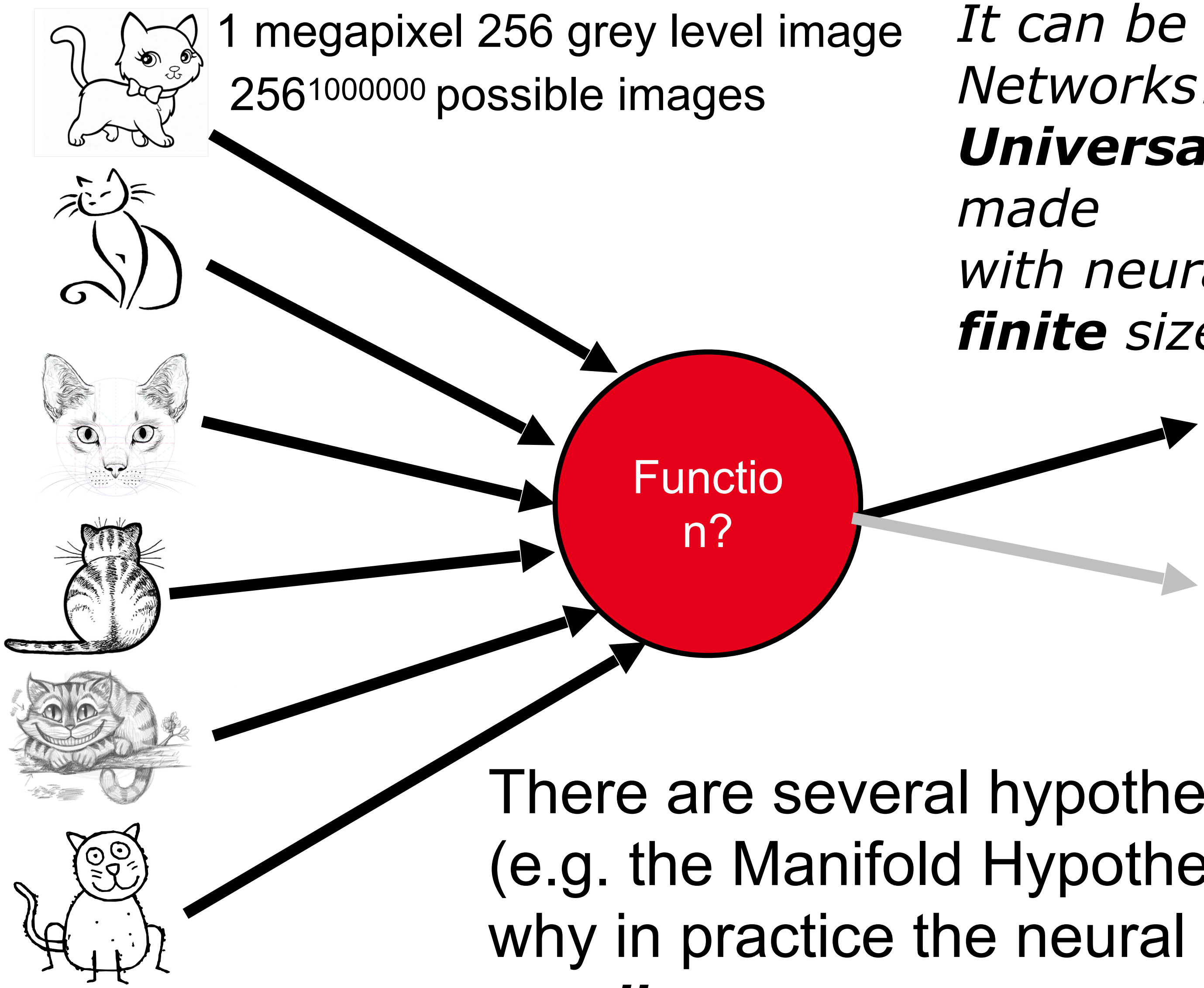
It can be done by Neural Networks:
Universal approximator *made with neural networks of* ***finite*** *size*



For each possible image, we wish to compute the probability that it depicts a cat. Then, the function is defined by a list of $256^{1,000,000}$ 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?*

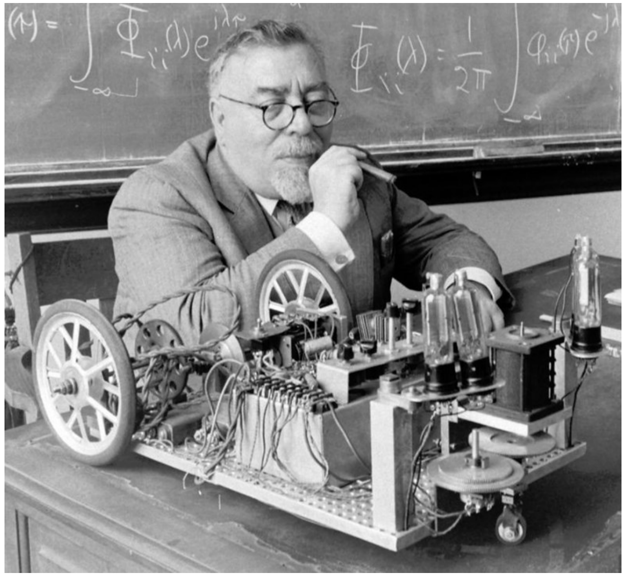


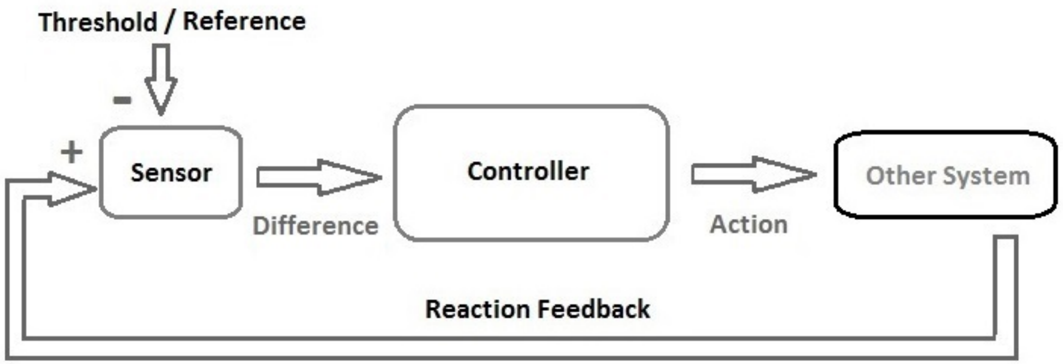
It can be done by Neural Networks:
Universal approximator *made with neural networks of* ***finite*** *size*

There are several hypotheses (e.g. the Manifold Hypothesis, …) why in practice the neural network ***is very small***

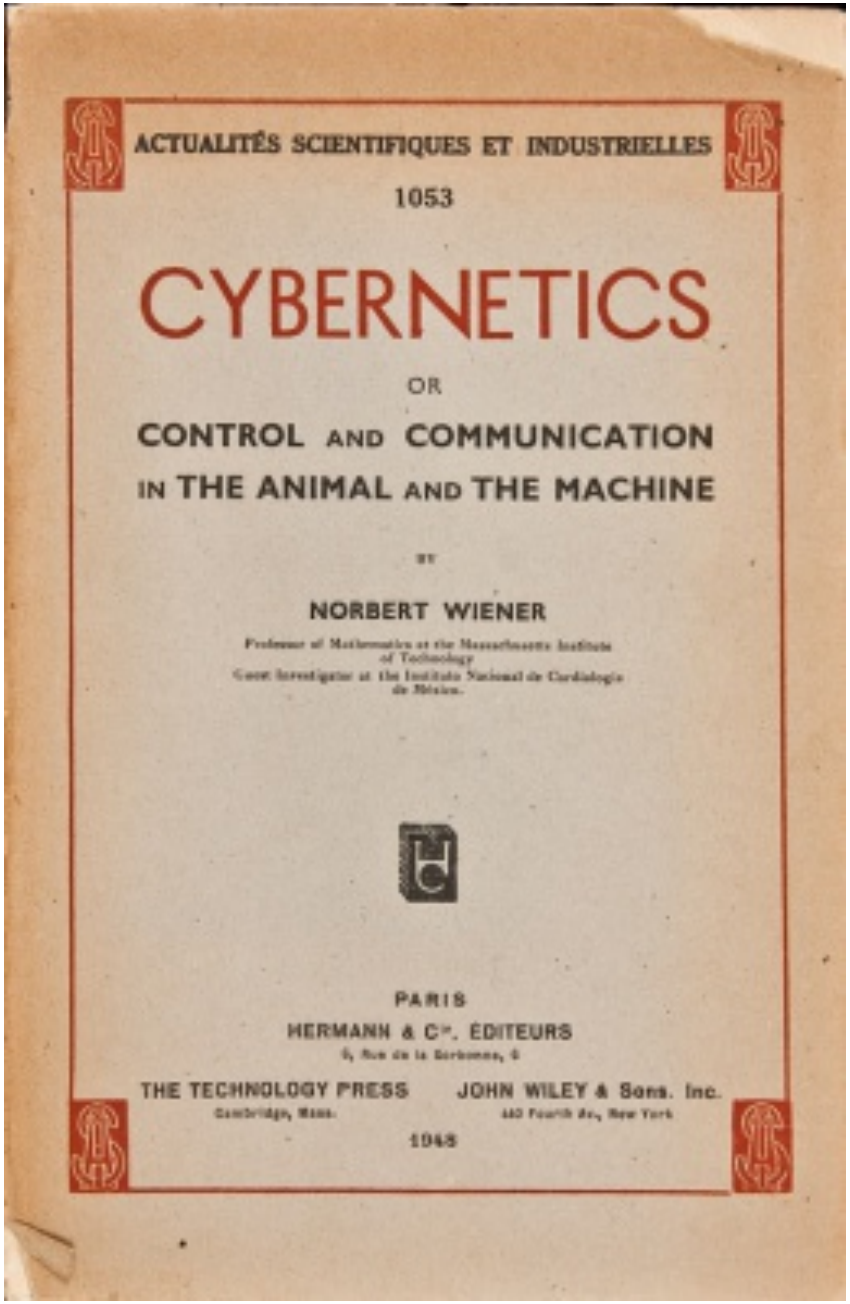
- Work of Henry W. Lin (Harward) , Max Tegmark (MIT), and David Rolnick (MIT) https://arxiv.org/abs/1608.08225

1948: NORBERT WIENER

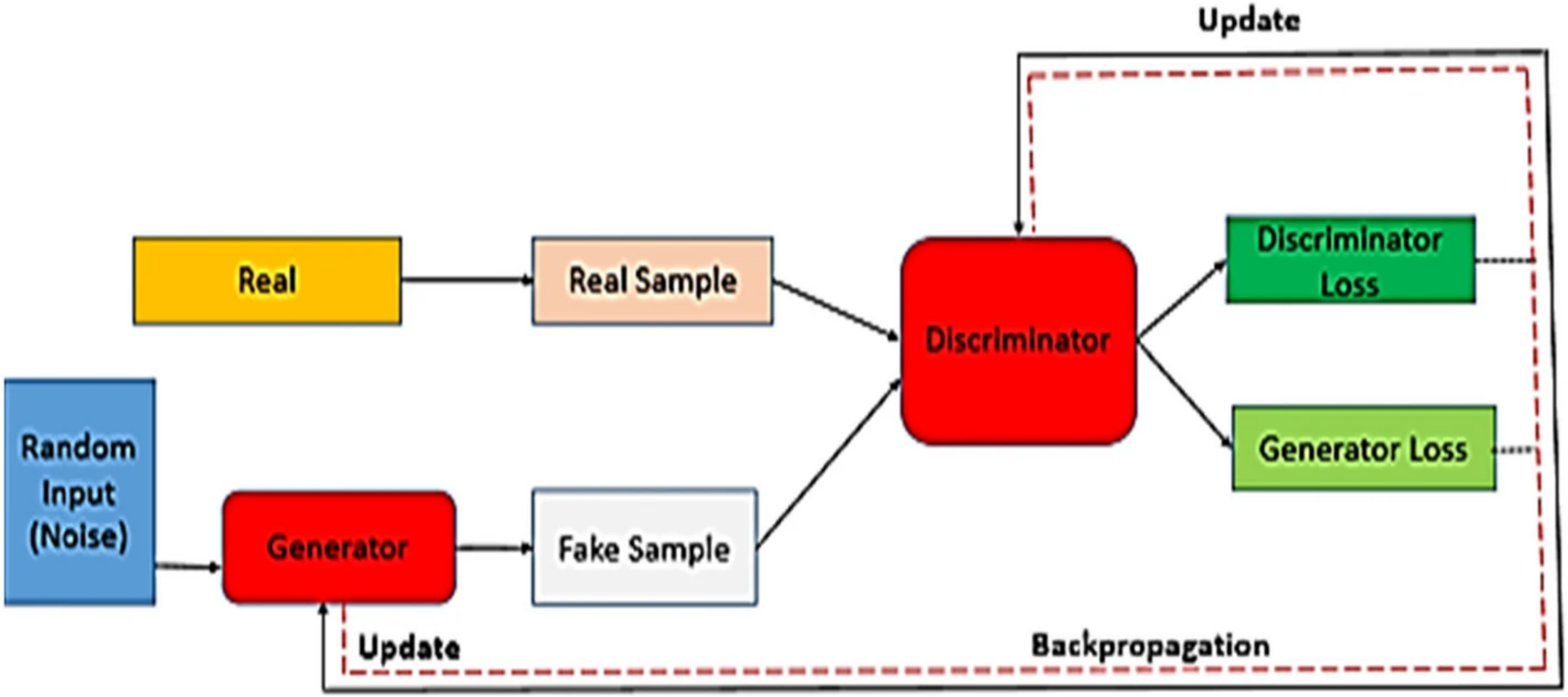




A Cybernetic Loop

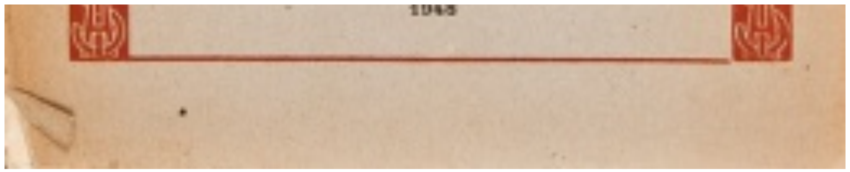


1948: NORBERT WIENER



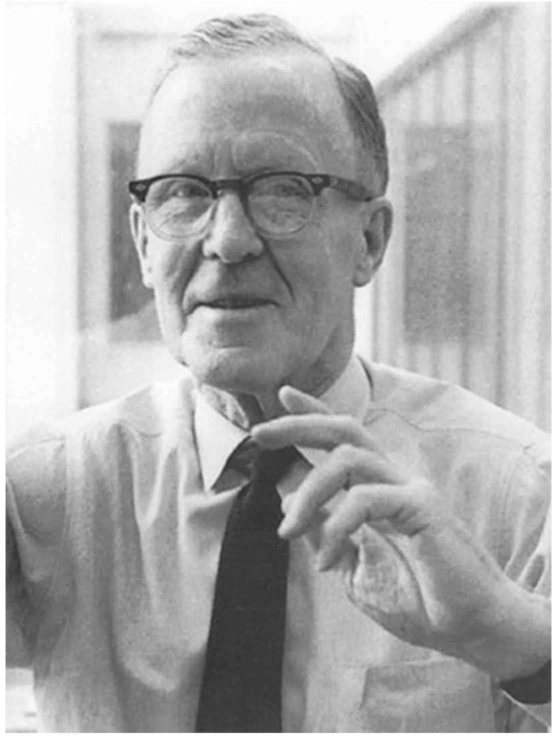
Basic Generative adversarial networks (GAN)
from https://link.springer.com/article/10.1007/s11042-024-18767-y

A Cybernetic Loop



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

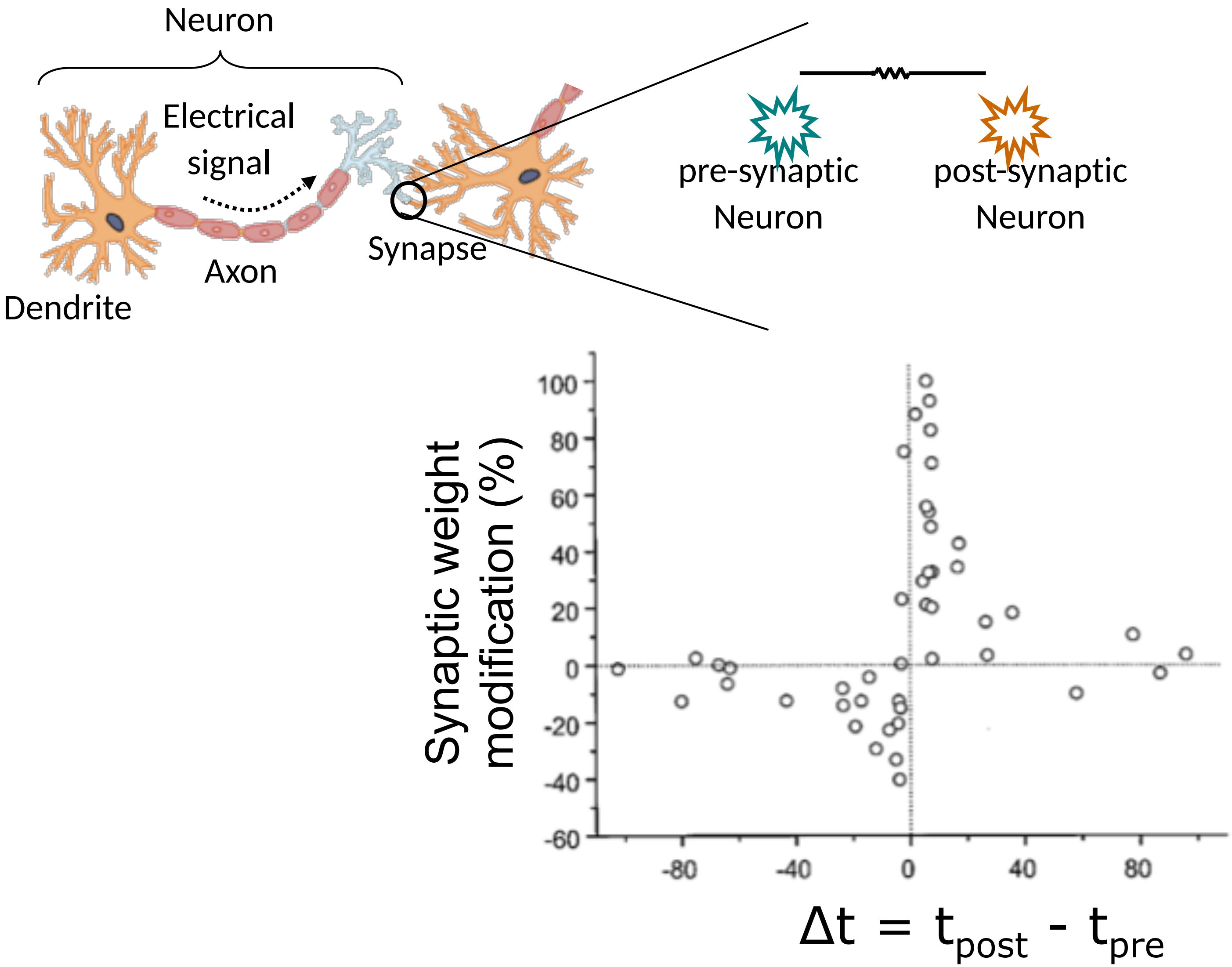
Basic mechanism for synaptic plasticity: an increase in synaptic efficacy arises from the presynaptic cell's repeated and persistent stimulation of the postsynaptic cell.



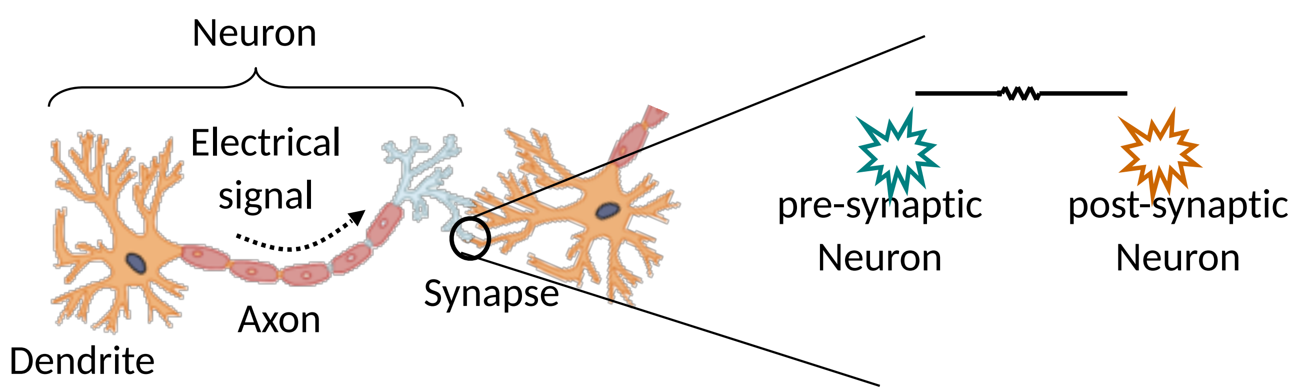
Psychologist, working in the area of neuropsychology

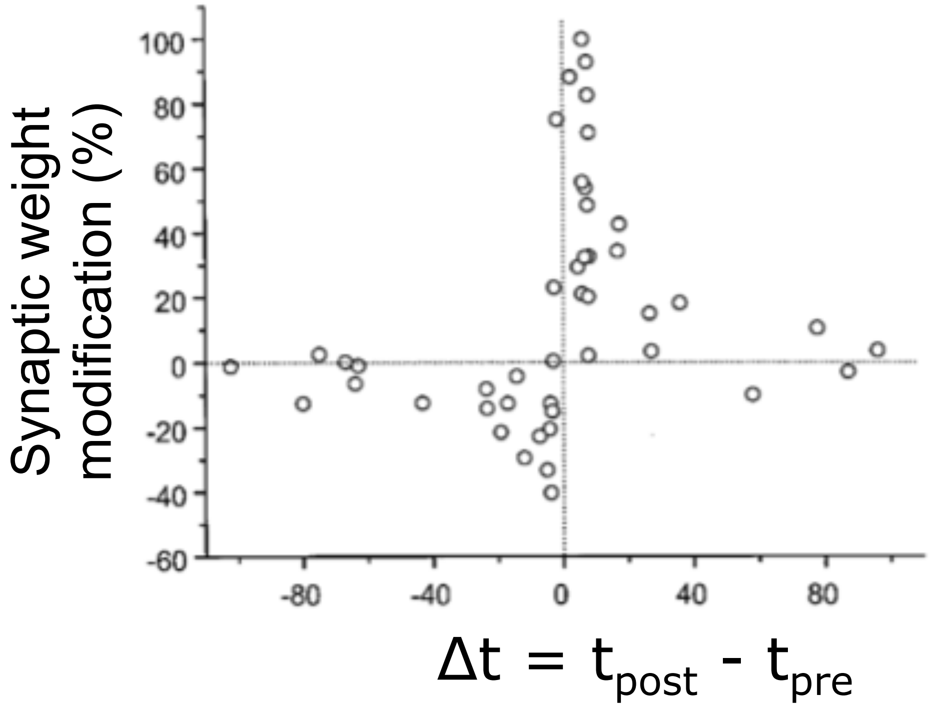
Introduced by Donald Hebb in his 1949 book « *The Organization of Behavior* »

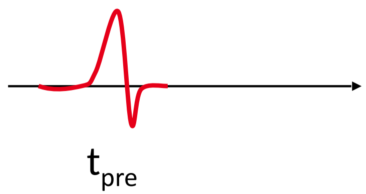
DERIVED FROM HEBB’S RULE: STDP (SPIKE TIMING DEPENDENT PLASTICITY)



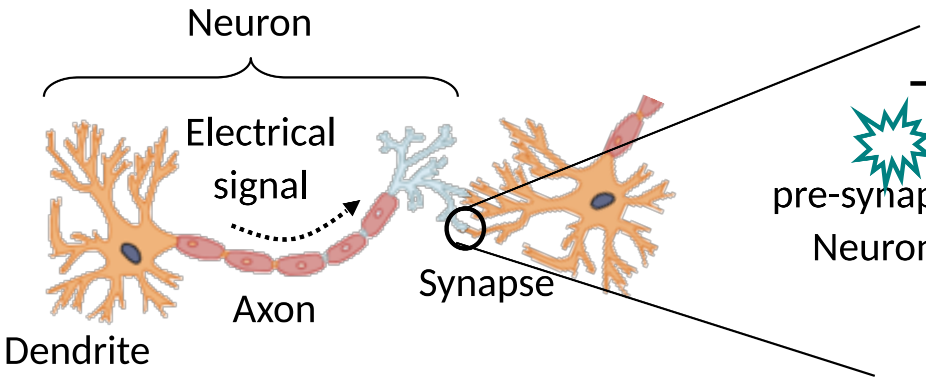
DERIVED FROM HEBB'S RULE: STDP (SPIKE TIMING DEPENDENT PLASTICITY)

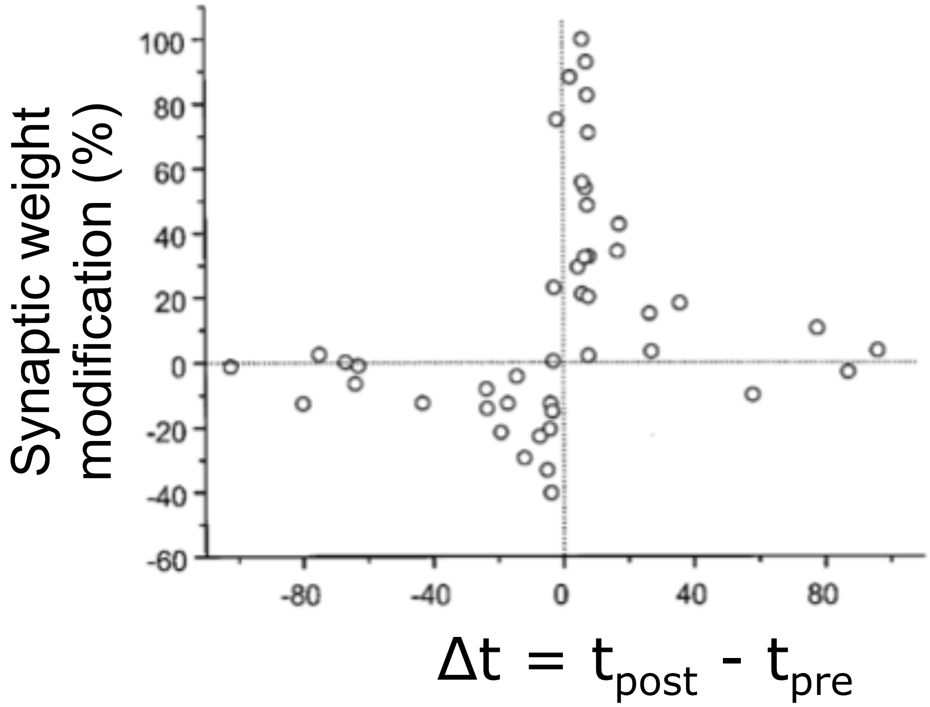


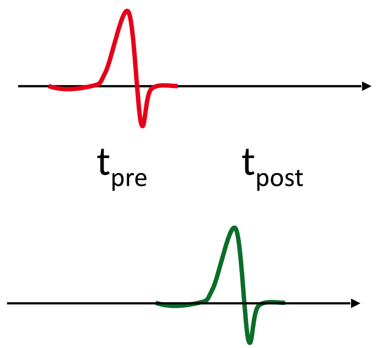




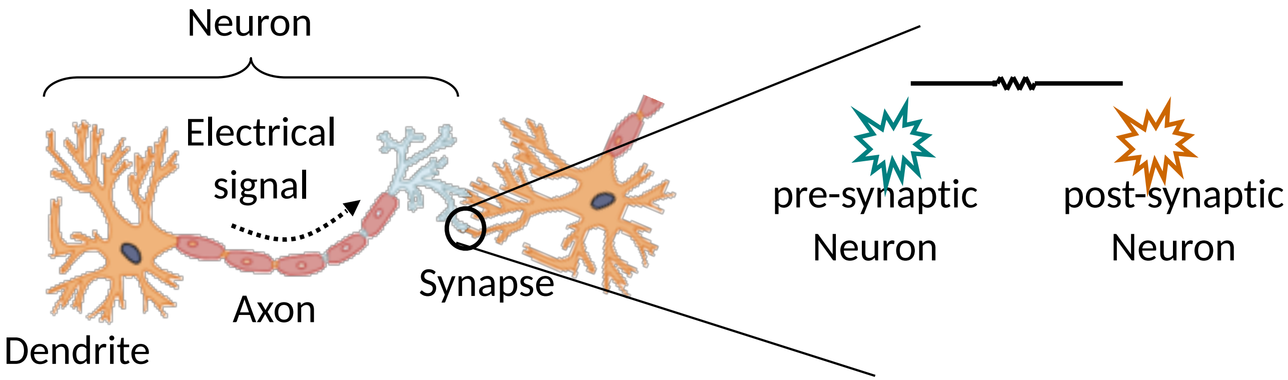
DERIVED FROM HEBB’S RULE: STDP (SPIKE TIMING DEPENDENT PLASTICITY)

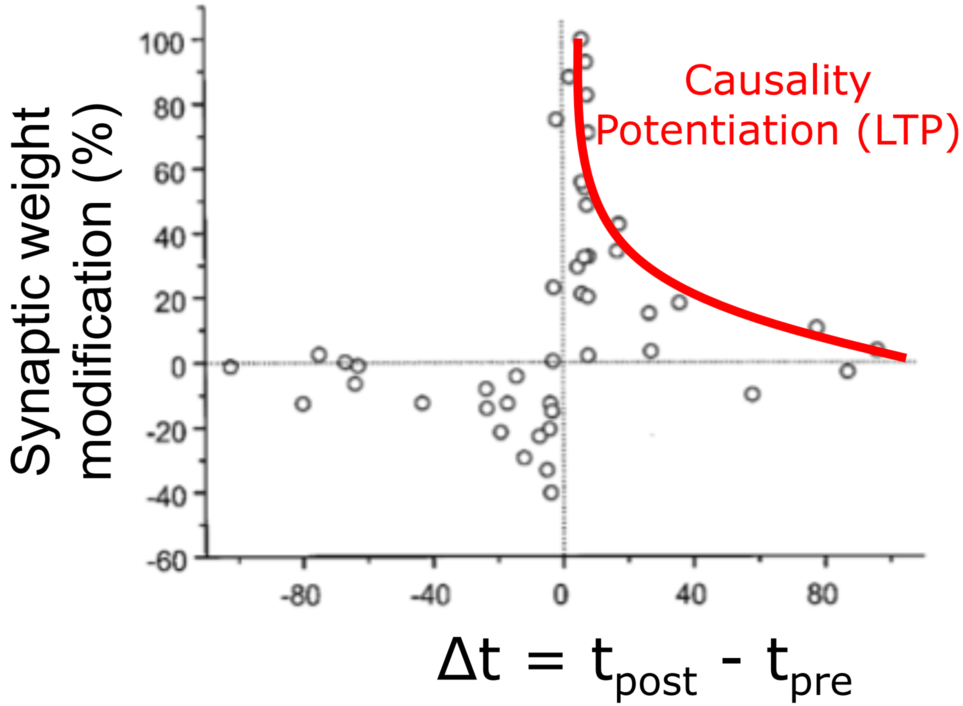


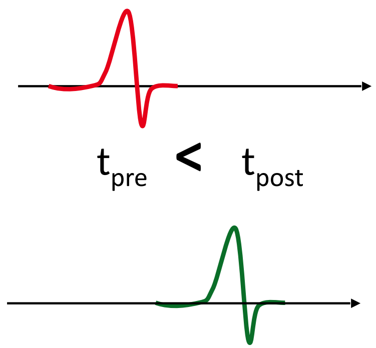




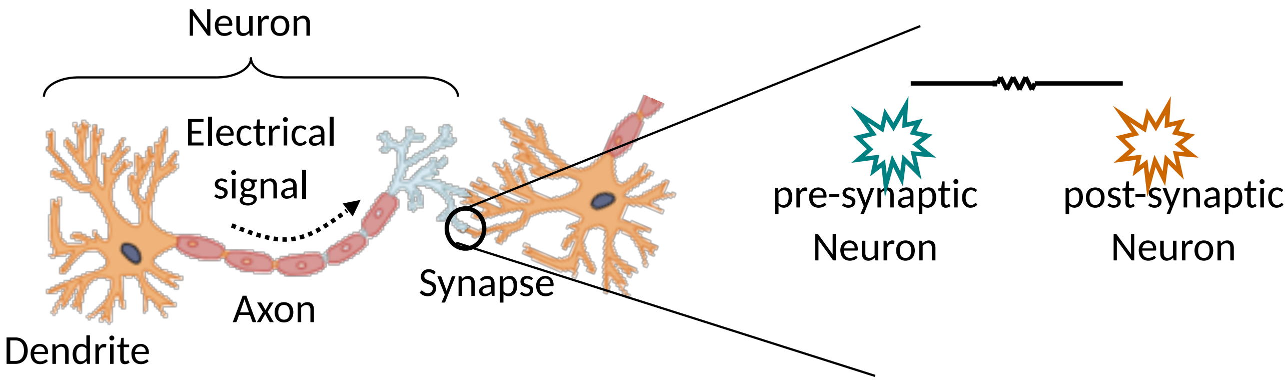
DERIVED FROM HEBB'S RULE: STDP (SPIKE TIMING DEPENDENT PLASTICITY)

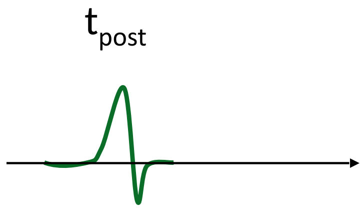


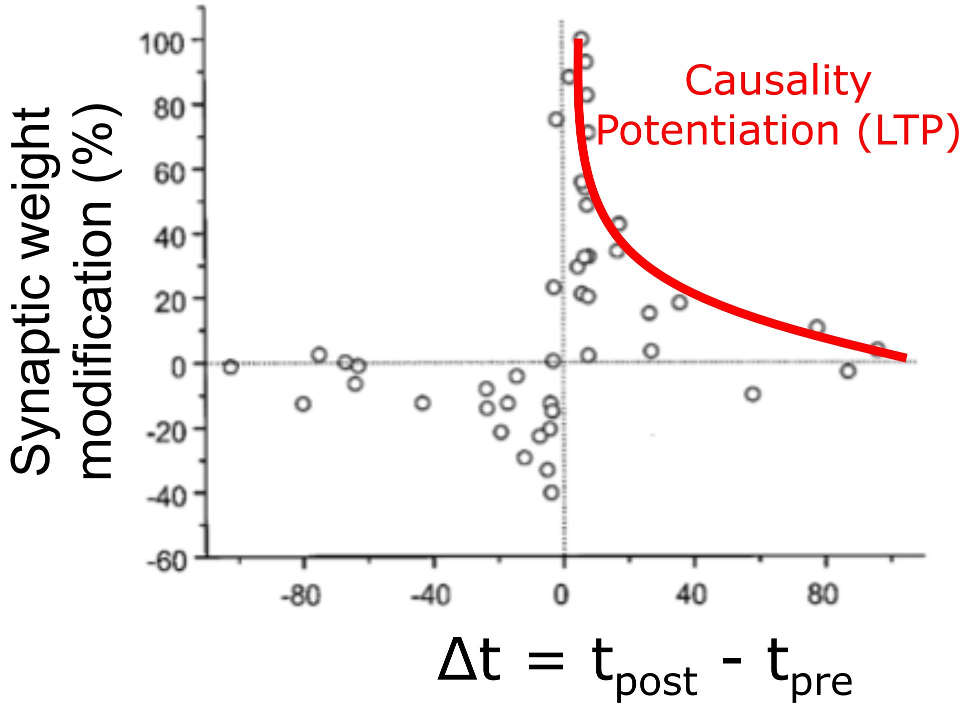


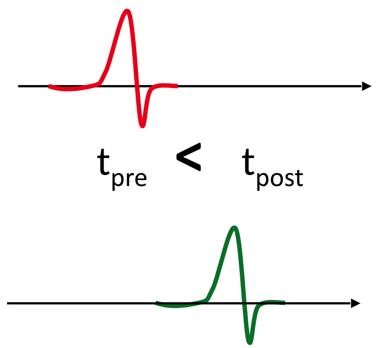


DERIVED FROM HEBB'S RULE: STDP (SPIKE TIMING DEPENDENT PLASTICITY)

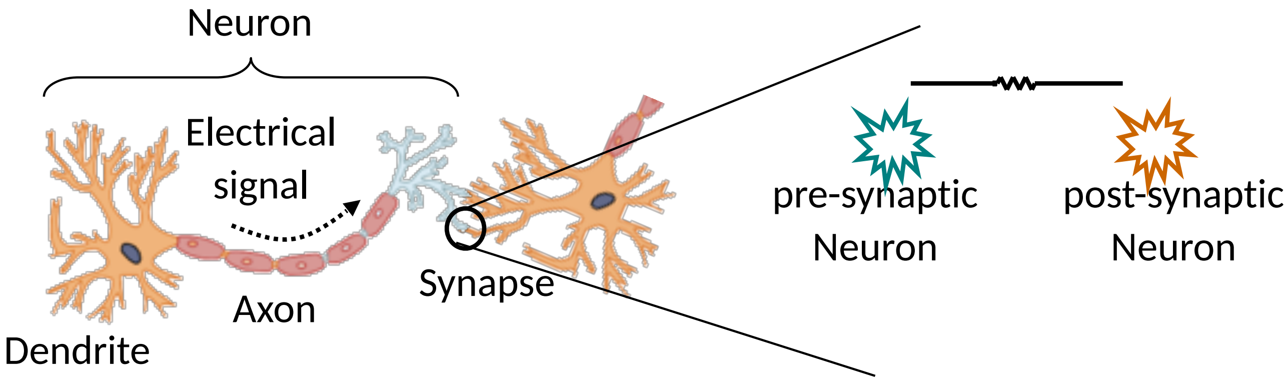


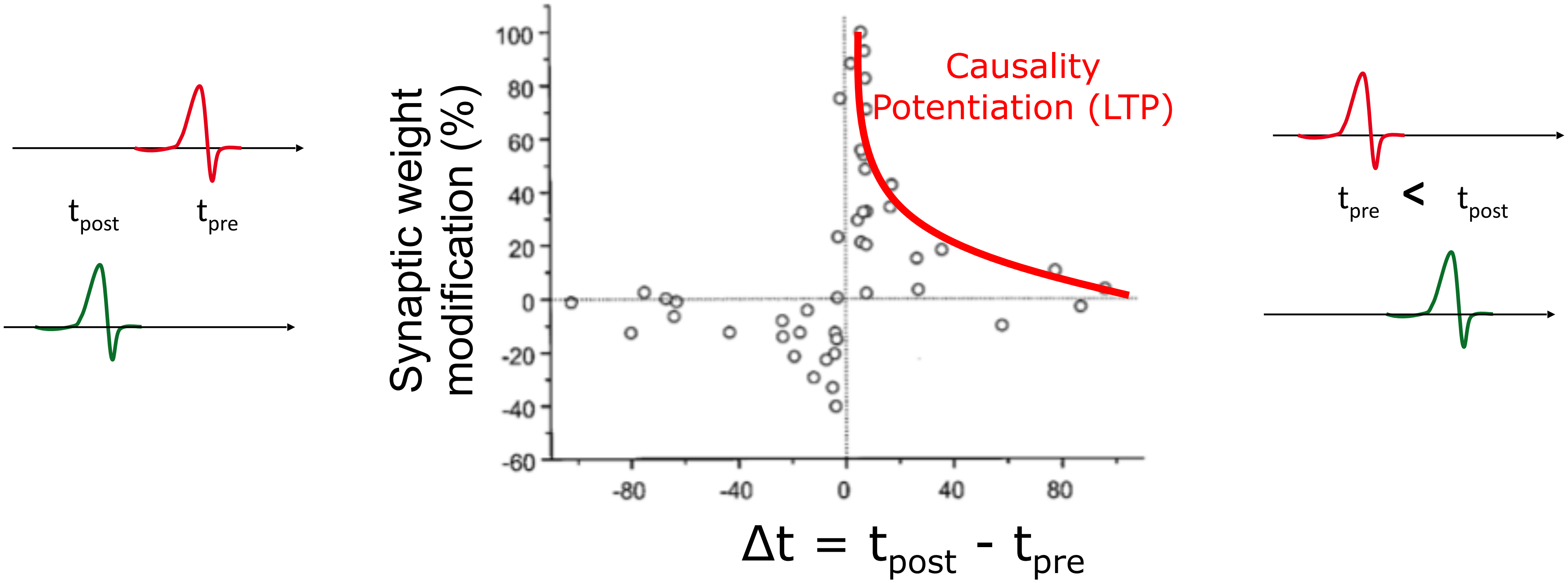




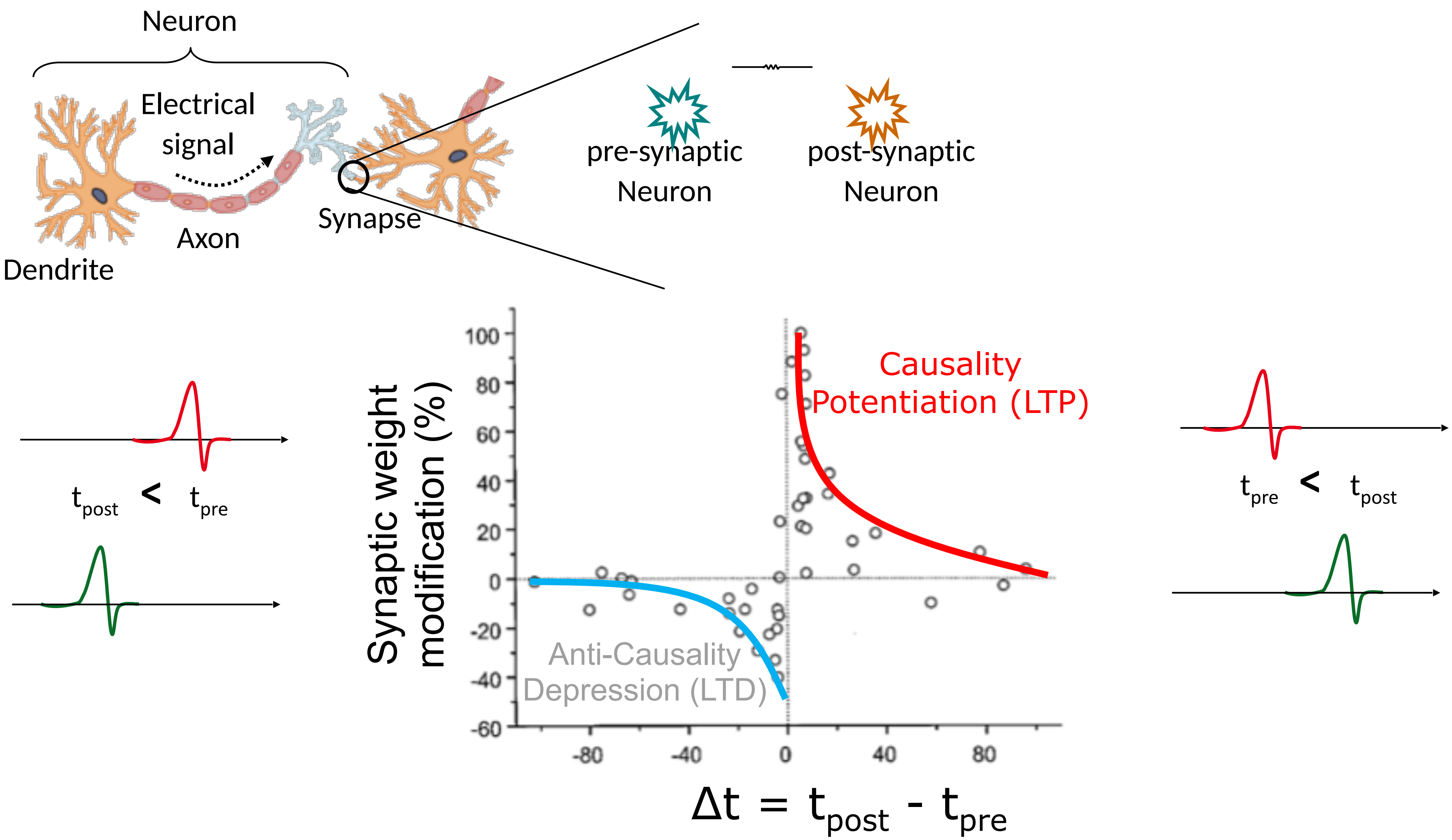


DERIVED FROM HEBB'S RULE: STDP
(SPIKE TIMING DEPENDENT PLASTICITY)

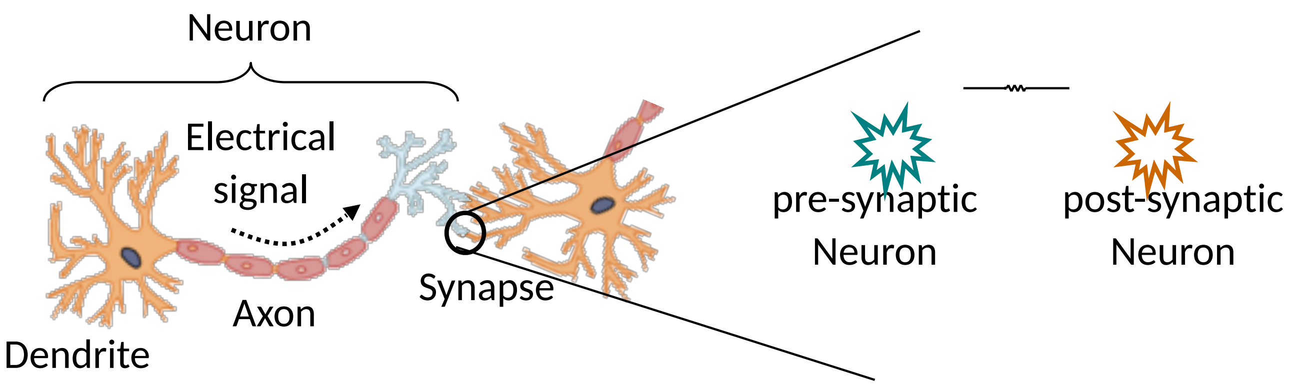




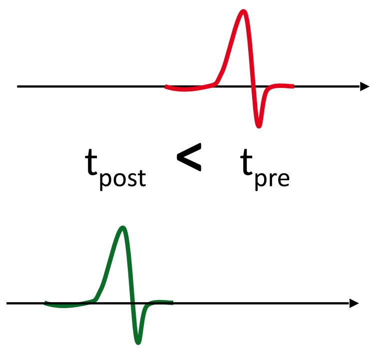
DERIVED FROM HEBB'S RULE: STDP
(SPIKE TIMING DEPENDENT PLASTICITY)

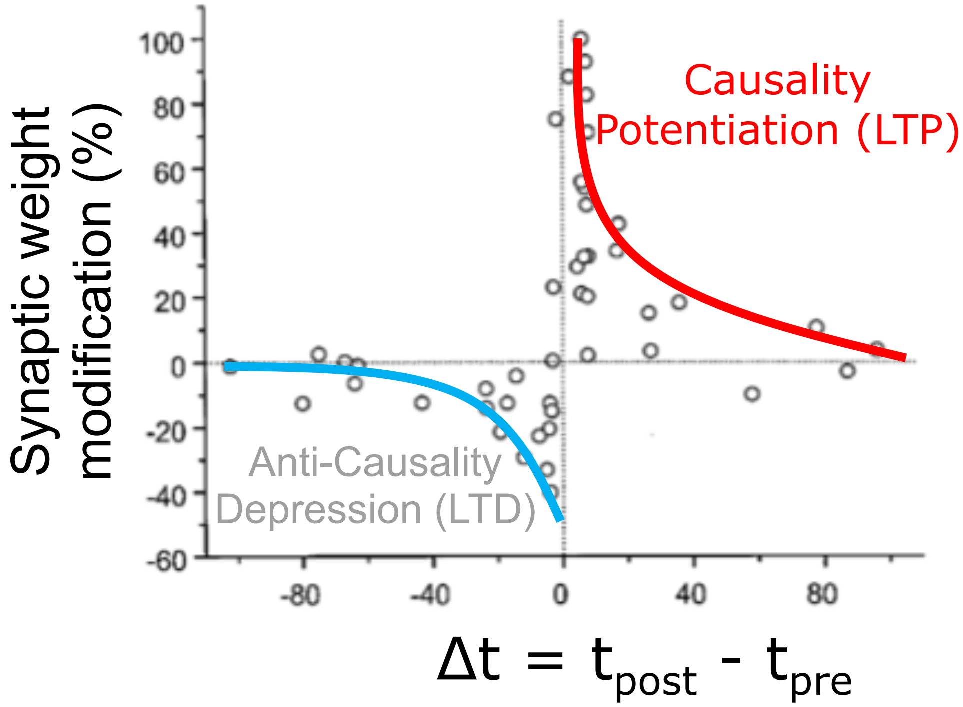


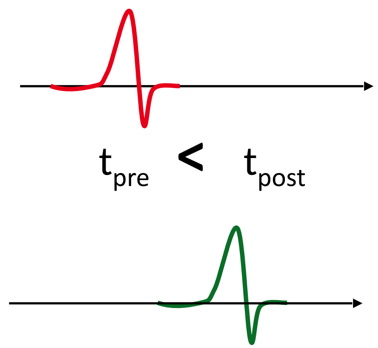
DERIVED FROM HEBB'S RULE: STDP
(SPIKE TIMING DEPENDENT PLASTICITY)



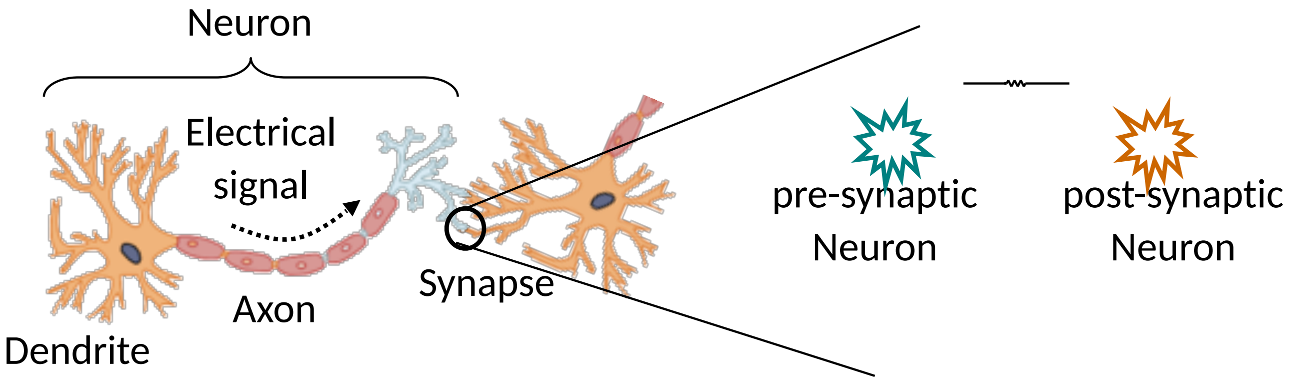
STDP = correlation detector
➔ Possible learning model of the brain?



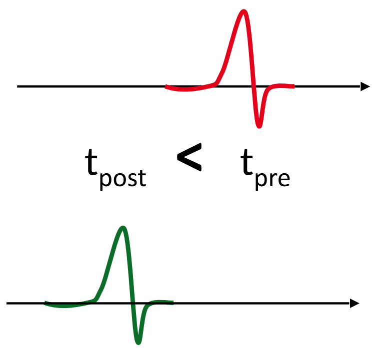


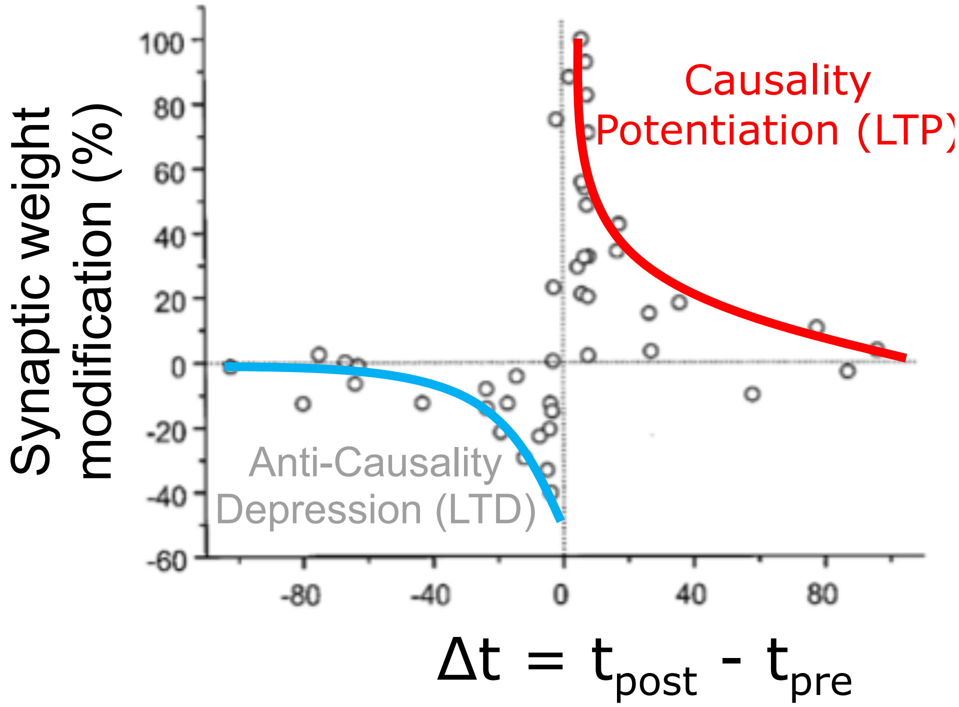


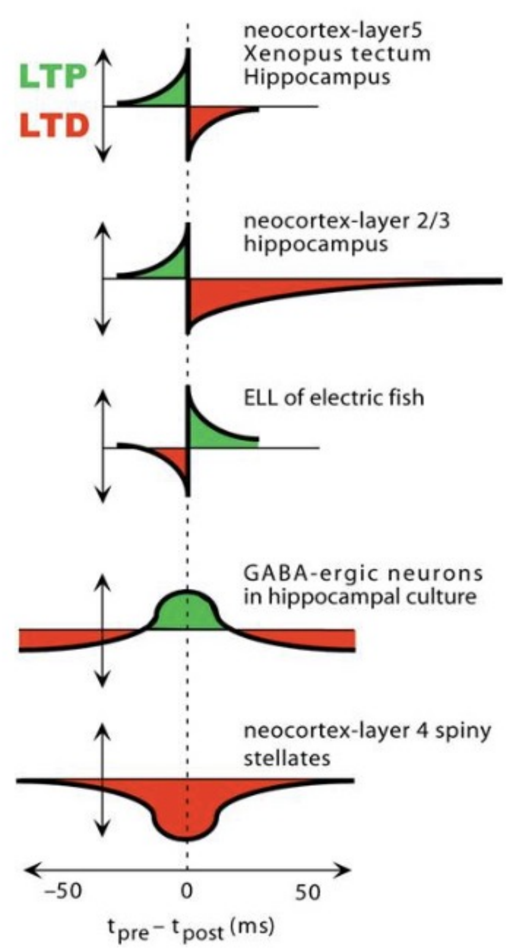
DERIVED FROM HEBB'S RULE: STDP
(SPIKE TIMING DEPENDENT PLASTICITY)



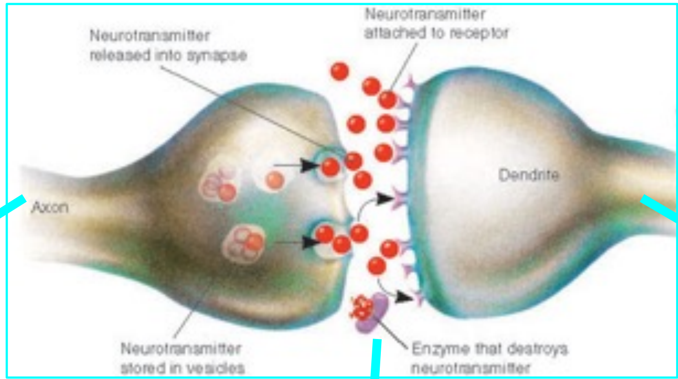
STDP = correlation detector







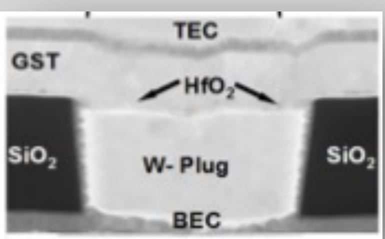
SIDE REMARK: INVESTIGATION OF RRAM AS SYNAPSES UNSUPERVISED LEARNING (INFORMATION CODED BY SPIKES)



Thermal effect

PCM

GST
GeTe
GST + HfO_2

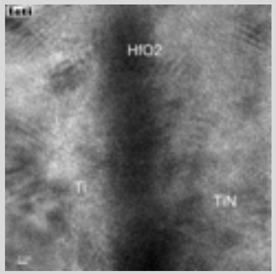


M.Suri, et. al, IEDM 2011
M.Suri, et. al, IMW 2012 , JAP 2012
O.Bichler et al. IEEE TED 2012
M.Suri et al., EPCOS 2013
D.Garbin et al., IEEE Nano 2013

Electronic effect
oxygen vacancies

OXRAM

$TiN/HfO_2/Ti/TiN$

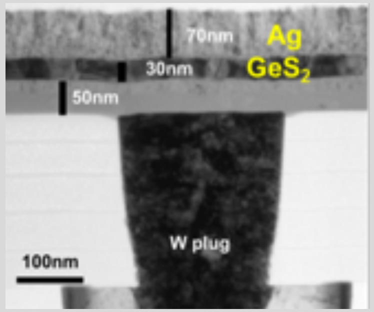


D.Garbin et al. IEDM 2014
D.Garbin et al., IEEE TED 2015

Electrochemical effect

CBRAM

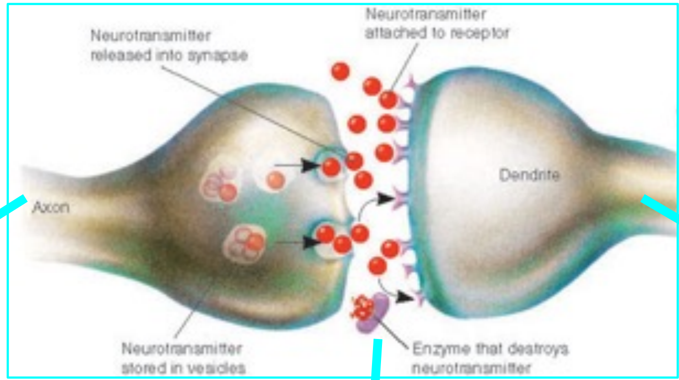
Ag / GeS_2



SIDE REMARK: INVESTIGATION OF RRAM AS SYNAPSES UNSUPERVISED LEARNING (INFORMATION CODED BY SPIKES)

Analog computing: using physical phenomenon to make computations

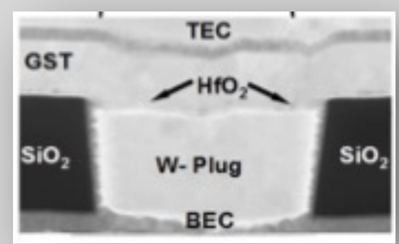
Thermal effect



Electrochemical effect

PCM

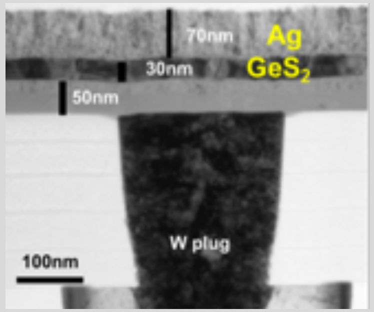
GST
GeTe
GST + HfO_2



Electronic effect oxygen vacancies

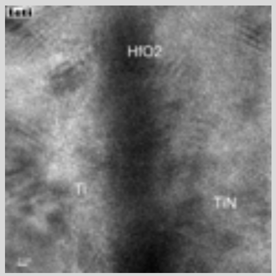
CBRAM

Ag / GeS_2



OXRAM

$TiN/HfO_2/Ti/TiN$



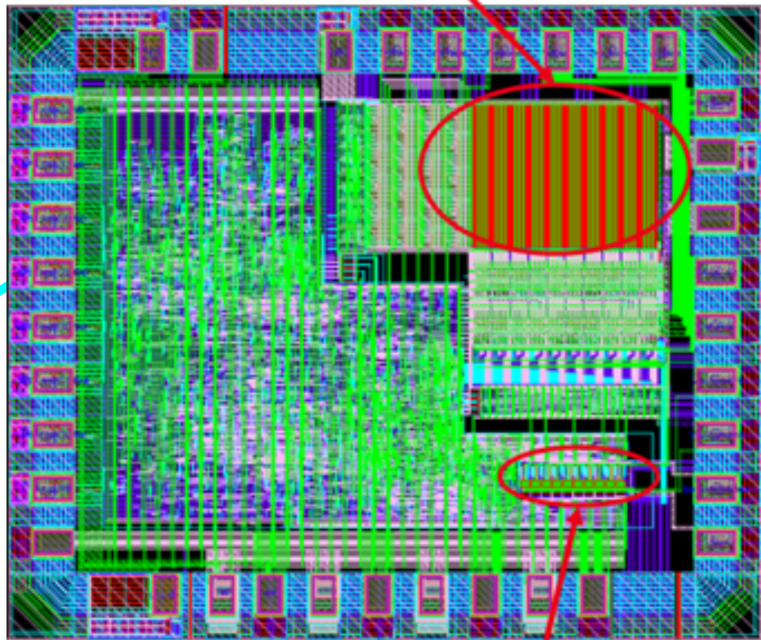
M.Suri, et. al, IEDM 2011
M.Suri, et. al, IMW 2012 , JAP 2012
O.Bichler et al. IEEE TED 2012
M.Suri et al., EPCOS 2013
D.Garbin et al., IEEE Nano 2013

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

SIDE REMARK: INVESTIGATION OF RRAM AS SYNAPSES UNSUPERVISED LEARNING (INFORMATION CODED BY SPIKES)

Analog computing: using physical phenomenon to make computations

OxRAMs

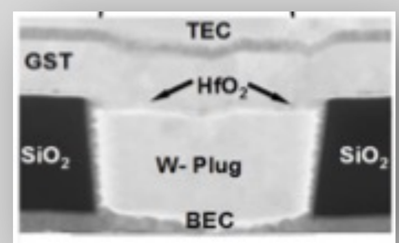


Neurons

Thermal effect

PCM

GST
GeTe
GST + HfO_2

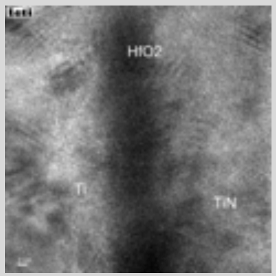


M.Suri, et. al, IEDM 2011
M.Suri, et. al, IMW 2012 , JAP 2012
O.Bichler et al. IEEE TED 2012
M.Suri et al., EPCOS 2013
D.Garbin et al., IEEE Nano 2013

Electronic effect oxygen vacancies

OXRAM

TiN/HfO_2/Ti/TiN

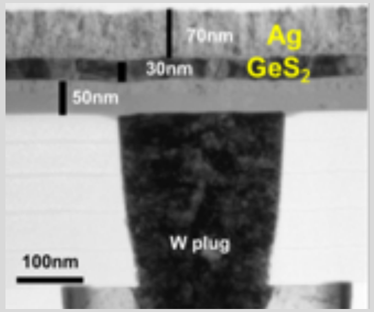


D.Garbin et al. IEDM 2014
D.Garbin et al., IEEE TED 2015

Electrochemical effect

CBRAM

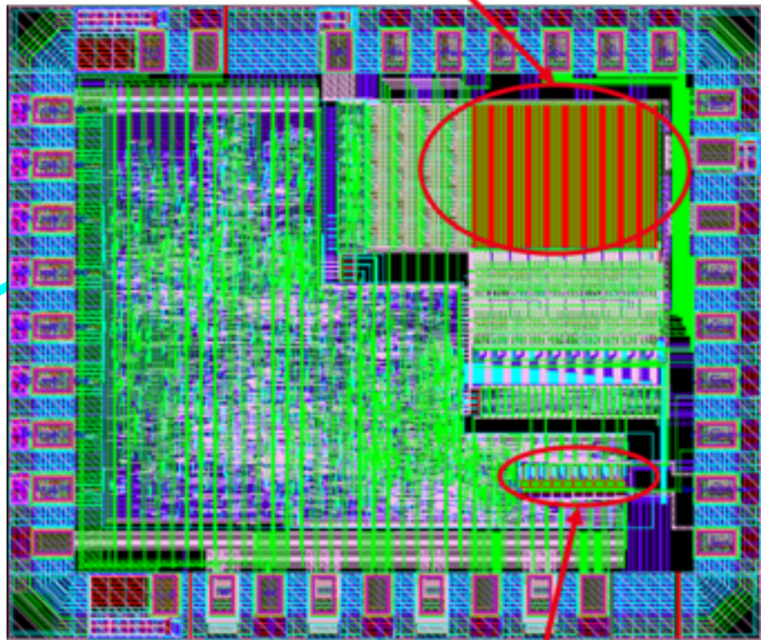
Ag / GeS_2



SIDE REMARK: INVESTIGATION OF RRAM AS SYNAPSES UNSUPERVISED LEARNING (INFORMATION CODED BY SPIKES)

Analog computing: using physical phenomenon to make computations

OxRAMs

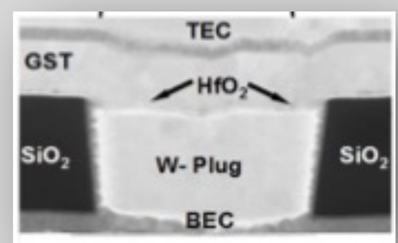


Neurons

Thermal effect

PCM

GST
GeTe
GST + HfO_2

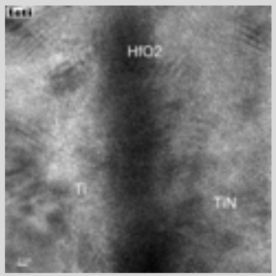


M.Suri, et. al, IEDM 2011
M.Suri, et. al, IMW 2012 , JAP 2012
O.Bichler et al. IEEE TED 2012
M.Suri et al., EPCOS 2013
D.Garbin et al., IEEE Nano 2013

Electronic effect oxygen vacancies

OXRAM

TiN/HfO_2/Ti/TiN

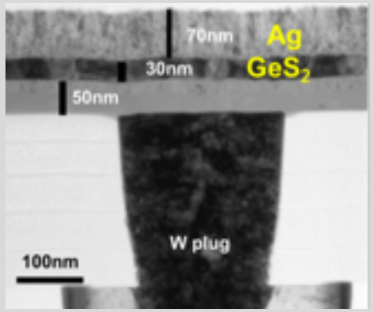


D.Garbin et al. IEDM 2014
D.Garbin et al., IEEE TED 2015

Electrochemical effect

CBRAM

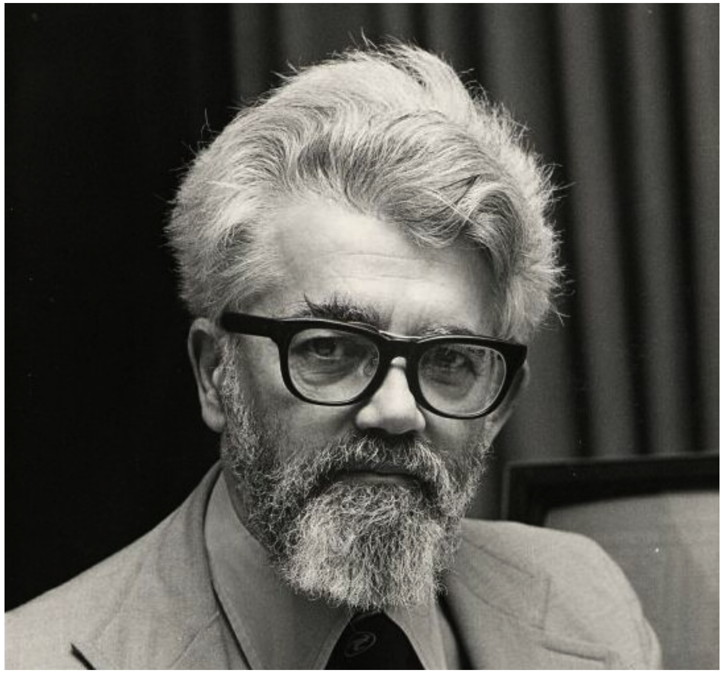
Ag / GeS_2



Leading to ***neuromorphic*** chips

1955: JOHN MCCARTHY

John McCarthy is one of the "founding fathers" of artificial intelligence, together with Marvin Minsky, Allen Newell and Herbert A. Simon.



McCarthy coined the term "artificial intelligence" in 1955, and organized the famous **Dartmouth Conference** in Summer 1956. This conference started AI as a science field.

While at MIT, McCarthy developed the programming language ***LISP*** in 1950, one of the two oldest programming language

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1))) ) )
```

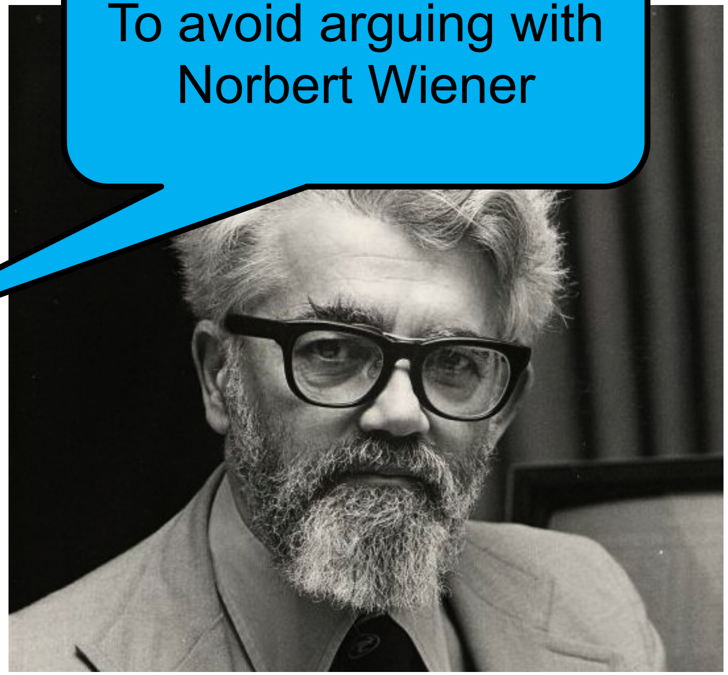
1955: JOHN MCCARTHY

John McCarthy is one of the "founding fathers" of artificial intelligence, together with Marvin Minsky, Allen Newell and Herbert A. Simon.

McCarthy coined the term "artificial intelligence" in 1955, and organized the famous **Dartmouth Conference** in Summer 1956. This conference started AI as a science field.

While at MIT, McCarthy developed the programming language ***LISP*** in 1950, one of the two oldest programming language

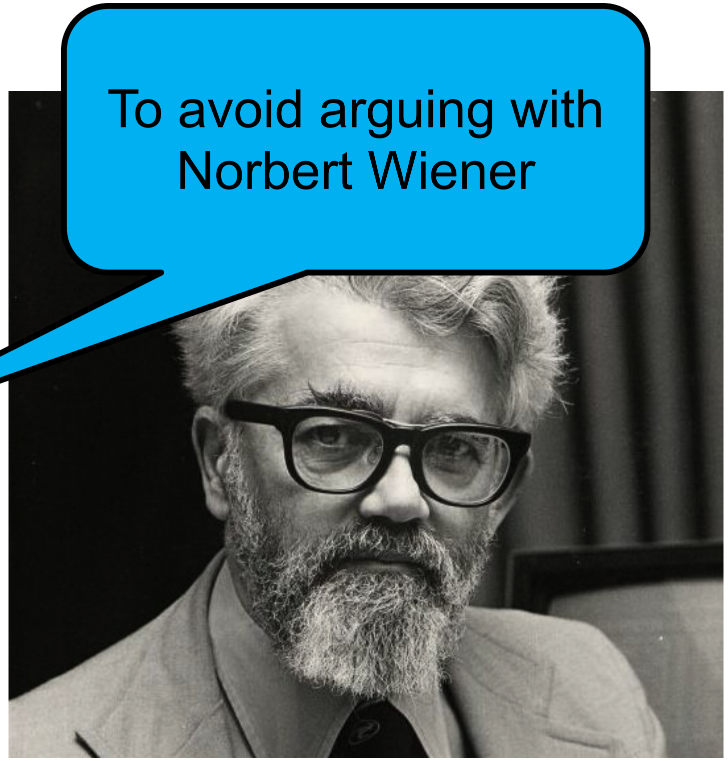
To avoid arguing with Norbert Wiener



```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1))) ) )
```

1955: JOHN MCCARTHY

John McCarthy is one of the "founding fathers" of artificial intelligence, together with Marvin Minsky, Allen Newell and Herbert A. Simon.



McCarthy coined the term "artificial intelligence" in 1955, and organized the famous **Dartmouth Conference** in Summer 1956. This conference started AI as a science field.

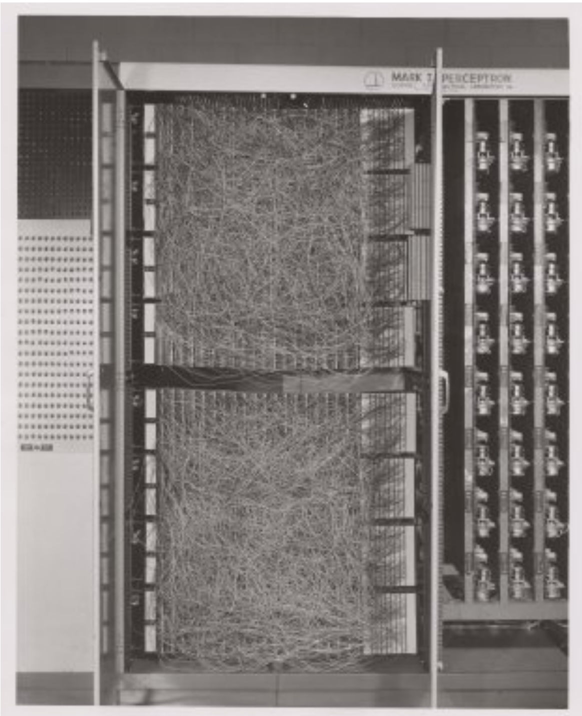
While at MIT, McCarthy developed the programming language ***LISP*** in 1950, one of the two oldest programming language

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1))) ) )
```

Recursive definition of a factorial

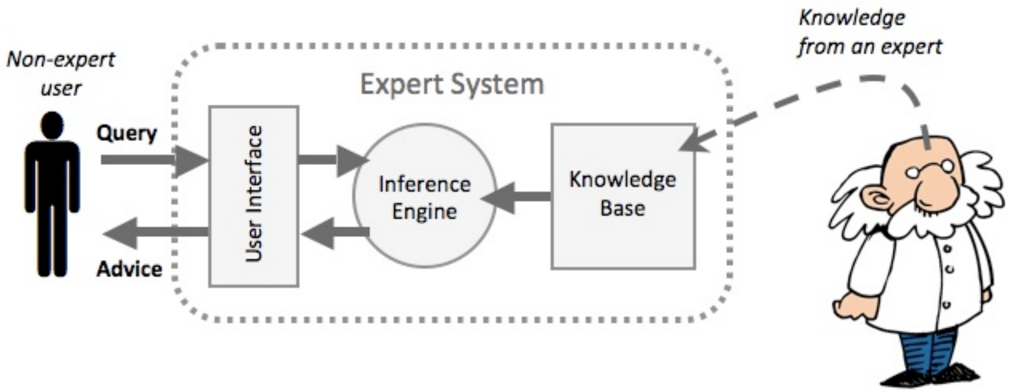
The perceptron algorithm was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt.

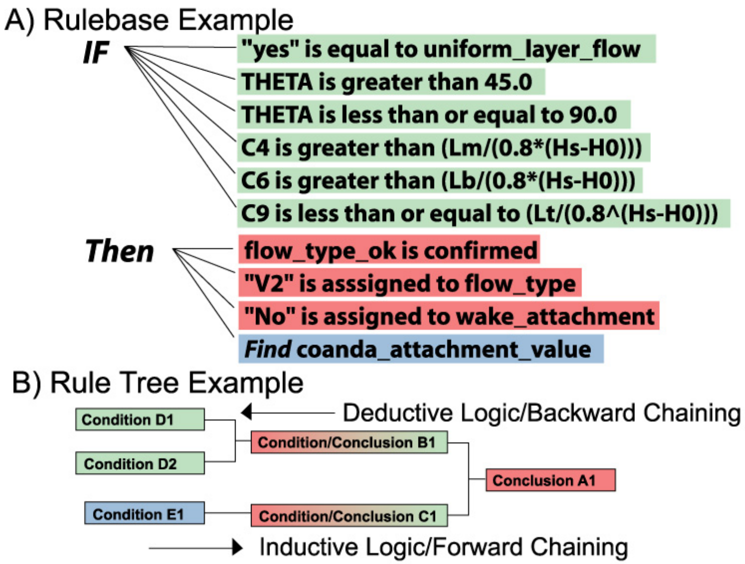
The perceptron was intended to be a machine, rather than a program, and while its first implementation was in software for the IBM 704, it was subsequently implemented in custom-built hardware as the "Mark 1 perceptron". This machine was designed for image recognition: it had an array of 400 photocells,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



1965: EXPERT SYSTEMS

Expert systems were introduced by the Stanford Heuristic Programming Project led by Edward Feigenbaum,
Can also use predicate logic or even Fuzzy Logic

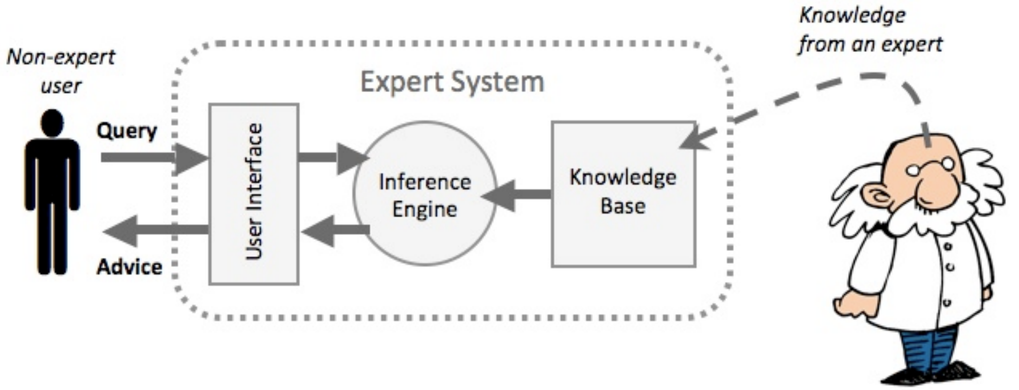


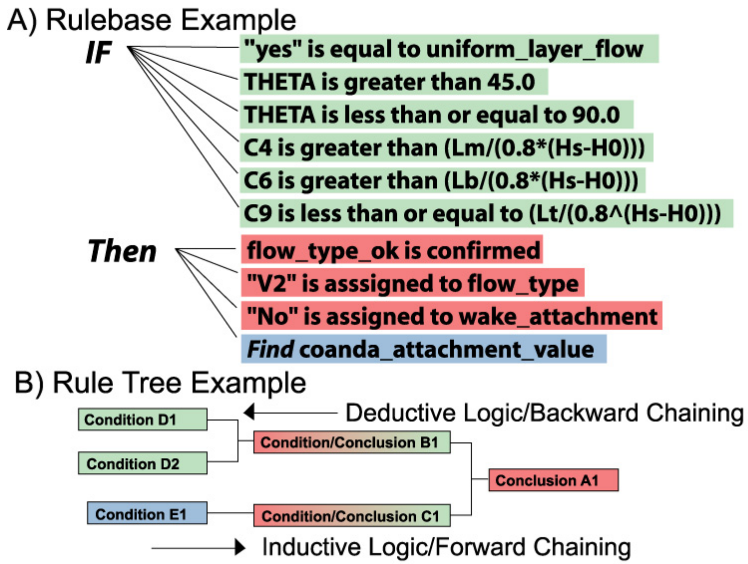


A Lisp machine

1965: EXPERT SYSTEMS

Expert systems were introduced by the Stanford Heuristic Programming Project led by Edward Feigenbaum,
Can also use predicate logic or even Fuzzy Logic





A Lisp machine

Decoupling:
- The (inference) engine
- The knowledge base

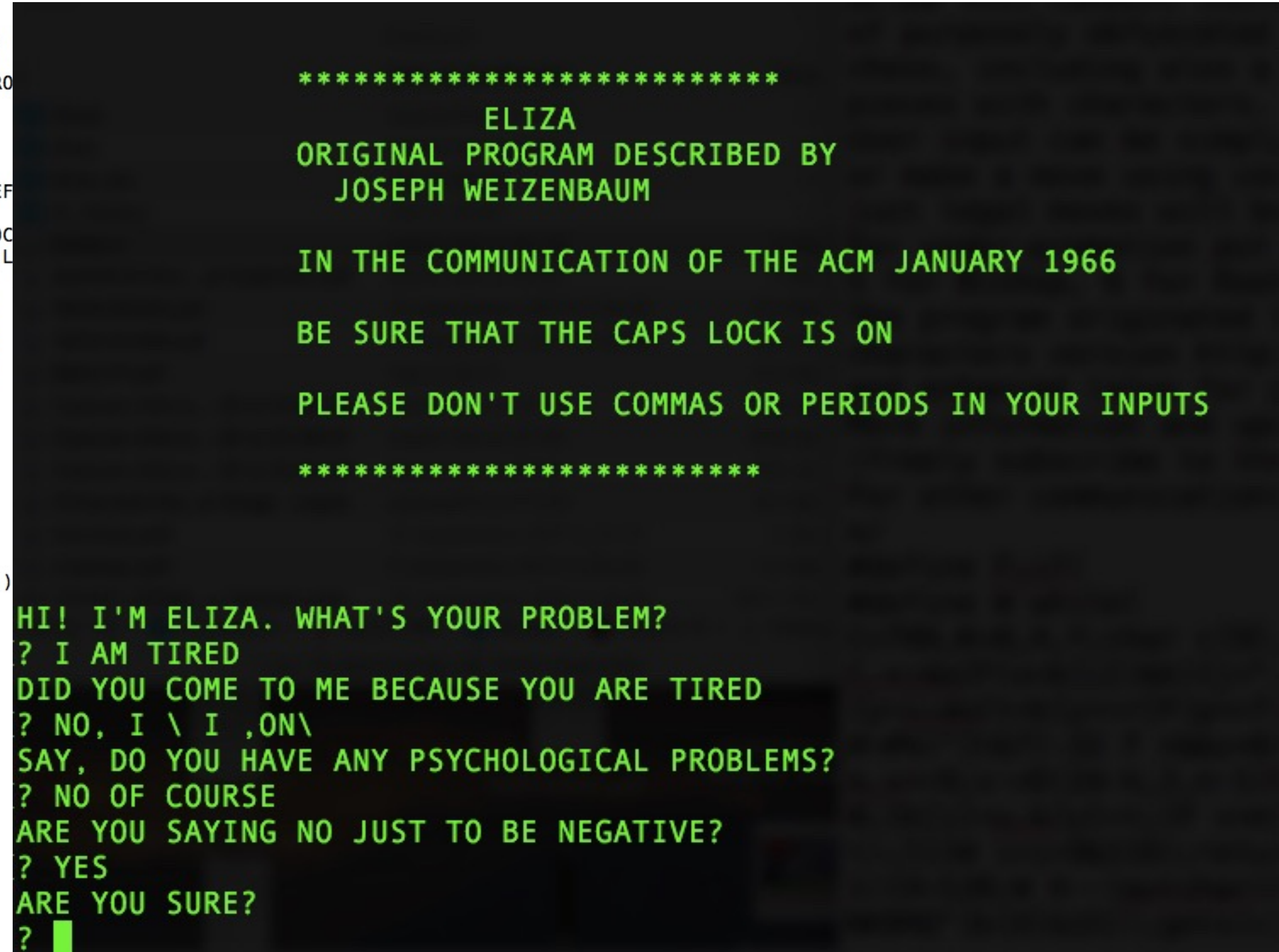
1966: ELIZA THE CHAT BOT !

```
;;; /DOCFNS/   31 JULY 1969  1007:42

(PROGN (PRIN1 (QUOTE FILE" CREATED ")
              T)
       (PRIN1 (QUOTE 08/22/68" 1522:26")
              T)
       (TERPRI T))
(DEFINEQ

(DOCTOR
  (LAMBDA NIL
    (PROG (SENTENCE KEYSTACK MEMSTACK TIMON)
;;           (SETSEPR 109 106 0)
;;           (SETBRK 14 12 31 1 13 8 9 27 26 3)
;;           (CONTROL T)
;;           (GCGAG NIL)
          (SETQ FLIPFLOP 0)
          (SETQ TIMON (QUOTIENT (CLOCK)
              60))
          (RECONSTRUCT (QUOTE (TELL ME YOUR PROBLEMS "."
                  PLEASE TERMINATE INPUT WITH A PERIOD OR A
                  QUESTION MARK ".")))
            T)
          (SETNONE)
      A   (PRIN1 (QUOTE "
*"))
          (COND
            ((NULL (SETQ SENTENCE (MAKESENTENCE)))
              (GO A)))
          (SETQ KEYSTACK (CDR SENTENCE))
          (SETQ SENTENCE (CAR SENTENCE))
          (COND
            ((EQUAL SENTENCE (QUOTE (GOODBYE)))
              (RETURN (RECONSTRUCT (APPEND (QUOTE (IT'S BEEN
                            MY PLEASURE "," THAT'S))
                      (CONS (PACK (LIST (QUOTE $)
                            (REMAINDER (PLUS (QUOTIENT
```

ELIZA is an early natural language processing computer program created from 1964 to 1966 at the MIT Artificial Intelligence Laboratory by Joseph Weizenbaum.

Created to demonstrate the superficiality of communication between man and machine, Eliza simulated conversation by using a 'pattern matching' and substitution methodology that gave users an illusion of understanding on the part of the program.

1966: ELIZA THE CHAT BOT !



ed
al

ciality
d
on by
itution
sion of
gram.

1966: ELIZA THE CHAT BOT !

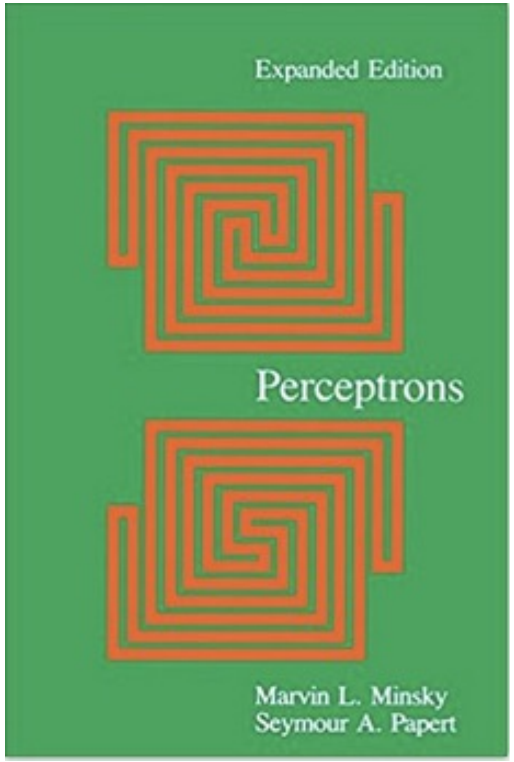
```
;;; /DOCFNS/   31 JULY 1969  1007:42

(PROGN (PRIN1 (QUOTE FILE" CREATED ")
              T)
       (PRIN1 (QUOTE 08/22/68" 1522:26")
              T)
       (TERPRI T))
(DEFINEQ

(DOCTOR
  (LAMBDA NIL
    (PROG (SENTENCE KEYSTACK MEMSTACK TIMON)
;;          (SETSEPR 109 106 0)
;;          (SETBRK 14 12 31 1 13 8 9 27 26 3)
;;          (CONTROL T)
;;          (GCGAG NIL)
          (SETQ FLIPFLOP 0)
          (SETQ TIMON (QUOTIENT (CLOCK)
              60))
          (RECONSTRUCT (QUOTE (TELL ME YOUR PROBLEMS "."
                PLEASE TERMINATE INPUT WITH A PERIOD OR A
                QUESTION MARK "."))
            T)
          (SETNONE)
      A   (PRIN1 (QUOTE "
*"))
          (COND
            ((NULL (SETQ SENTENCE (MAKESENTENCE)))
              (GO A)))
          (SETQ KEYSTACK (CDR SENTENCE))
          (SETQ SENTENCE (CAR SENTENCE))
          (COND
            ((EQUAL SENTENCE (QUOTE (GOODBYE)))
              (RETURN (RECONSTRUCT (APPEND (QUOTE (IT'S BEEN
                        MY PLEASURE "," THAT'S))
                    (CONS (PACK (LIST (QUOTE $)
                        (REMAINDER (PLUS (QUOTIENT
```

ELIZA is an early natural language processing computer program created from 1964 to 1966 at the MIT Artificial Intelligence Laboratory by Joseph Weizenbaum.

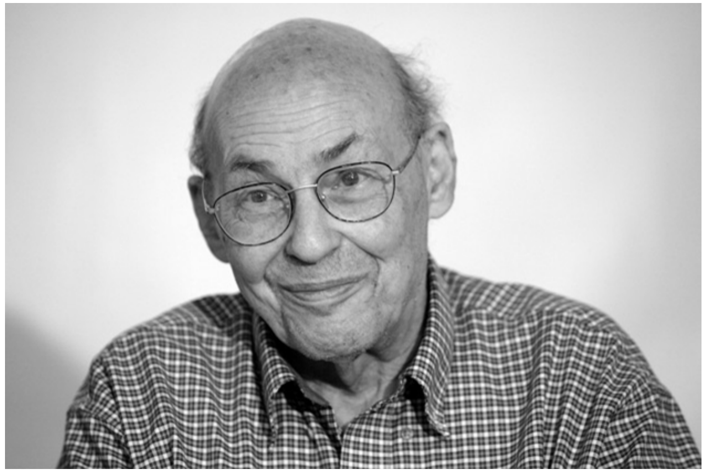
Created to demonstrate the superficiality of communication between man and machine, Eliza simulated conversation by using a 'pattern matching' and substitution methodology that gave users an illusion of understanding on the part of the program.

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**".

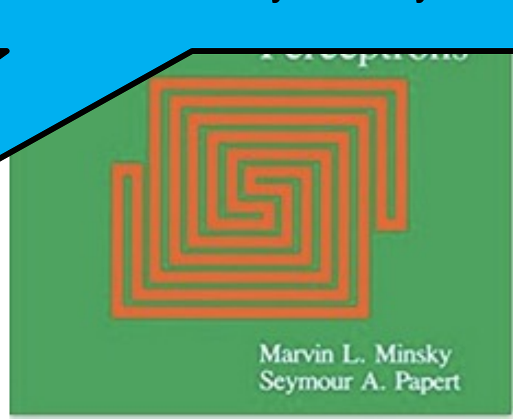


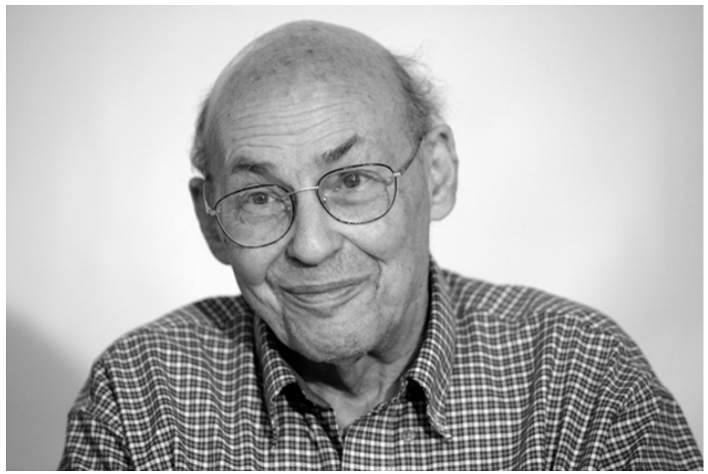
1969: MARVIN MINSKY

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**".

On the surface, XOR appears to be a very simple problem, however, Minksy and Papert (1969) showed that this was a big problem for neural network architectures of the 1960s, known as Perceptrons which are only one layer.





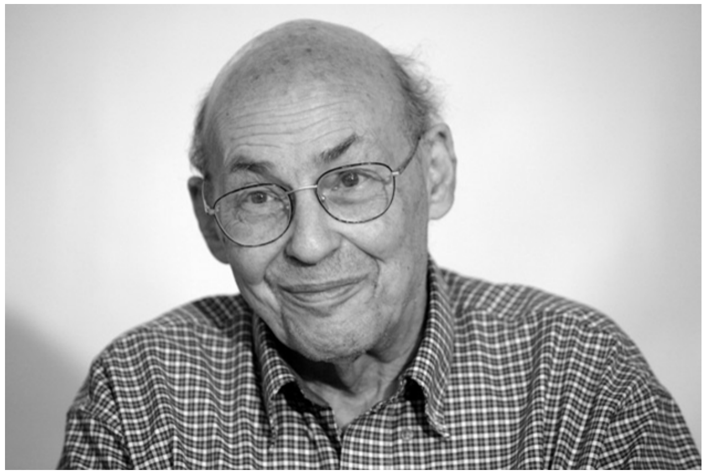
1969: MARVIN MINSKY

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**".

On the surface, XOR appears to be a very simple problem, however, Minksy and Papert (1969) showed that this was a big problem for neural network architectures of the 1960s, known as Perceptrons which are only one layer.

Input 1	Input 2	Output
0	0	0
0	1	1
1	1	0
1	0	1



1969: MARVIN MINSKY

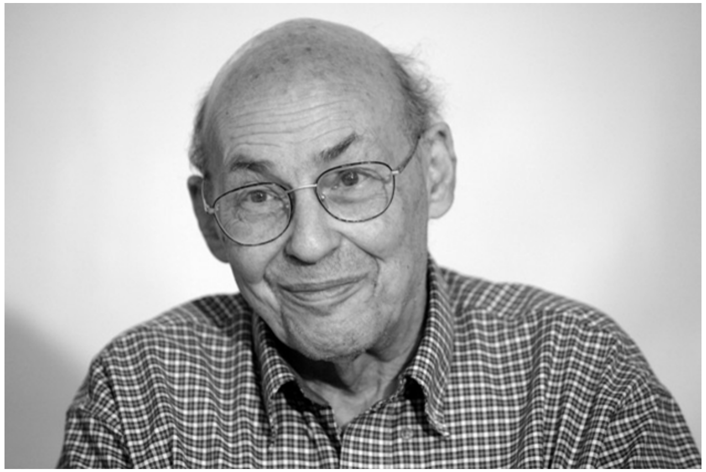
He developed, with Seymour Papert, the first Logo "turtle".
Minsky also built, in 1951, the first randomly wired neural network learning machine, SNARC.

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**".

On the surface, XOR appears to be a very simple problem, however, Minksy and Papert (1969) showed that this was a big problem for neural network architectures of the 1960s, known as Perceptrons which are only one layer.

Input 1	Input 2	Output
0	0	0
0	1	1
1	1	0
1	0	1

$$\begin{aligned} p \oplus q &= (p \land \neg q) \lor (\neg p \land q) \\ &= (p \lor q) \land (\neg p \lor \neg q) \\ &= (p \lor q) \land \neg(p \land q) \end{aligned}$$



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

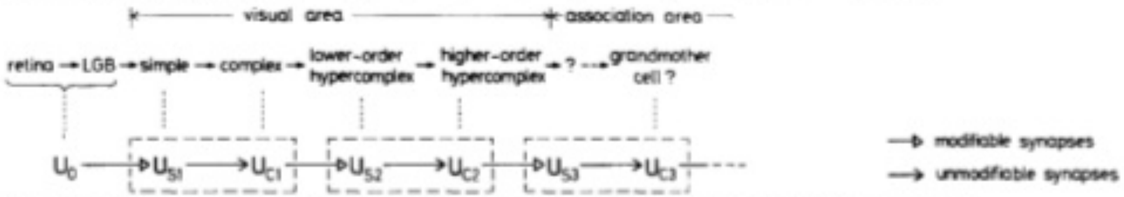


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

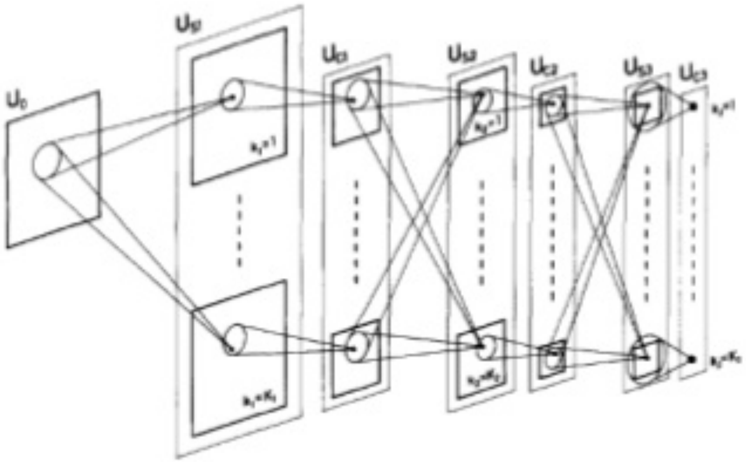


Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

Biol. Cybernetics 36, 193–202 (1980)

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

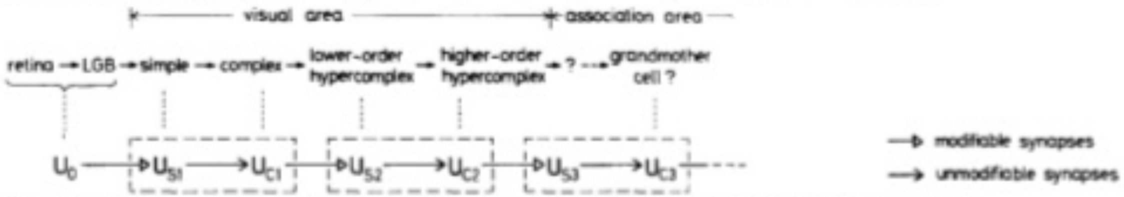


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

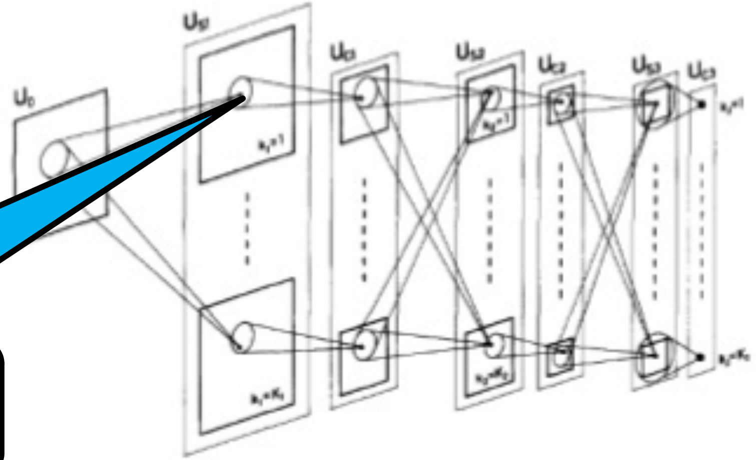


Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

But no real algorithms to set the values of the synaptic weights

Biol. Cybernetics 36, 193–202 (1980)

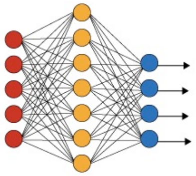
He was one of the first researchers who demonstrated the use of **generalized back-propagation algorithm** for training multi-layer neural networks.

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



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.

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.

COGNITIVA 85

Paris, 4-7 Juin 1985

A LEARNING SCHEME FOR ASSYMETRIC THRESHOLD NETWORK.

UNE PROCEDURE D'APPRENTISSAGE POUR RESEAU A SEUIL ASSYMETRIQUE.

YANN LE CUN

Ecole Supérieure d'Ingénieurs en Electrotechnique et Electronique, 91 rue Falguière 75015 Paris
and Laboratoire de Dynamique des Réseaux, 1 rue Descartes 75005 Paris.

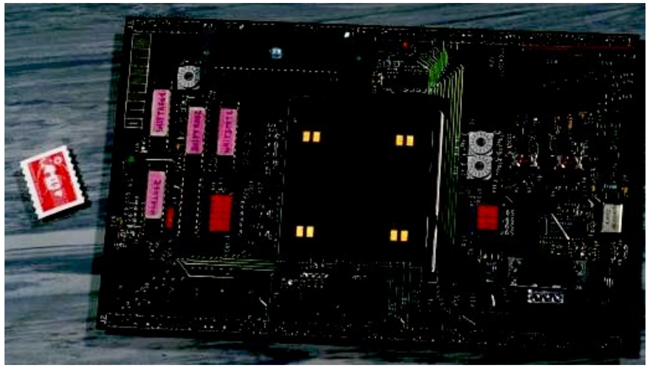
RESUME

Une nouvelle méthode paramétrique d'apprentissage supervisé utilisant un réseau parallèle d'automates à seuil est proposée. Le modèle est constitué de trois types d'éléments: les cellules d'entrée, les cellules de sortie, et les cellules internes, ces dernières n'ayant aucune interaction directe avec l'extérieur. L'apprentissage est un processus itératif local qui minimise une fonction de coût en modifiant les interactions entre cellules. L'utilisation d'une matrice de connexions assymetrique ainsi que la modification par l'apprentissage des paramètres des cellules internes constituent les principales particularités de ce modèle. Ceci permet l'apprentissage de discriminations dans le cas non linéairement séparable ainsi que la synthèse de prédicats d'ordre élevé. Des simulations effectuées sur un réseau hiérarchique de quelques centaines d'éléments mettent en évidence les capacités de *généralisation* du réseau (production d'une réponse correcte pour une forme non apprise) dans le cas de la reconnaissance d'images bruitées de basse résolution avec réponse invariante par faible translation et distortion. Des simulations en conditions d'*auto-apprentissage* (avec une sortie désirée auto-générée) ont également été effectuées pour modéliser l'apprentissage Pavlovien et les associations objet-symbole.

SUMMARY

A new parametric method for supervised learning is presented which is based on a threshold network structure. The model is composed of three types of units: input units, output units, and hidden units, the last group having no interaction with the outside world. The learning process is a local iterative scheme which minimizes a particular cost function by modifying the interactions between units. The non-symetric nature of the weight matrix as well as the modification of the hidden units weights by the learning process constitute the main particularities of this model. This system can learn high order predicates and discrminations in the non-linearly separable case. Simulations have been performed using hierarchical networks containing several hundred cells. The network exhibits generalization abilities (i.e. production of a correct output for a non-learned input pattern) on a low-resolution noisy picture recognition task. Other simulations have been done in self-learning conditions (i.e. with self-generated desired output) that modelize Pavlovian learning and object-symbol associations.

1990'S NEUROCOMPUTERS...

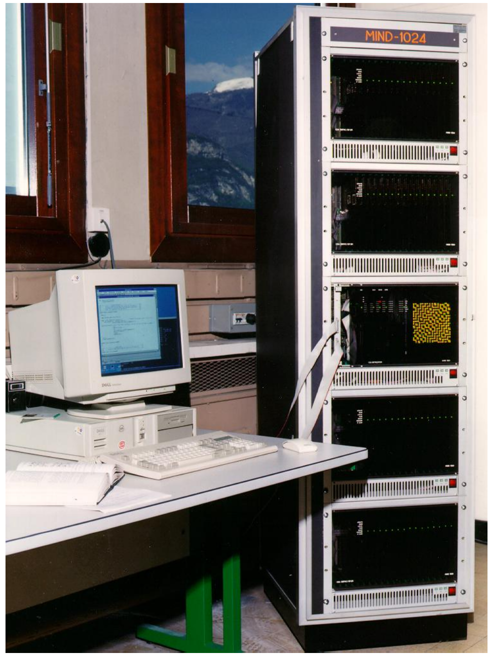


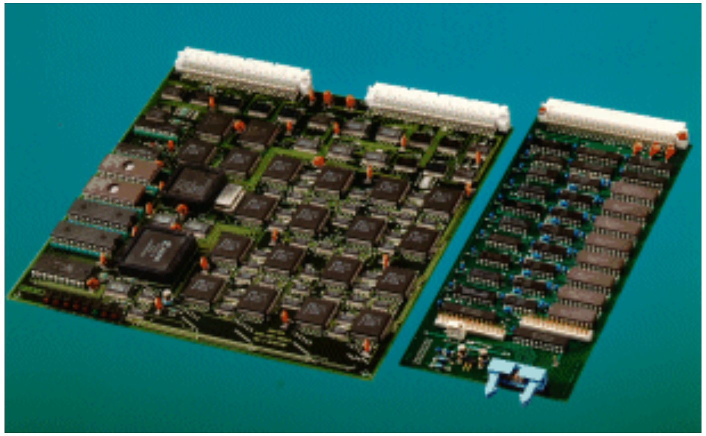
Philips : L-Neuro

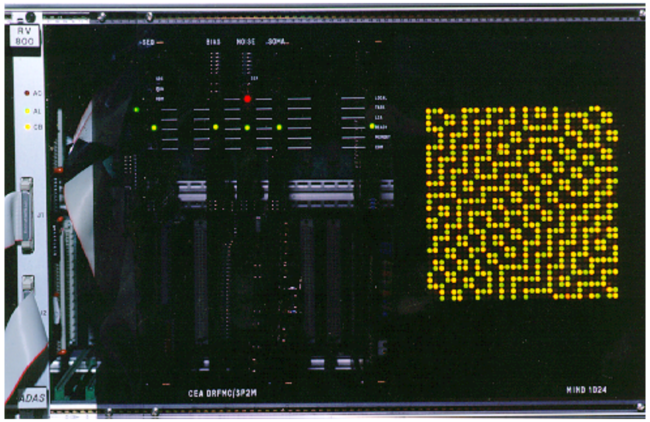
- 1st Gen 16 PEs 26 MCps (1990)
- 2nd Gen 12 PEs 720 MCps (1994)
- ➢ Used in satellite, fruit sorting, PCB inspection, sleep analysis, …

CEA's MIND machine

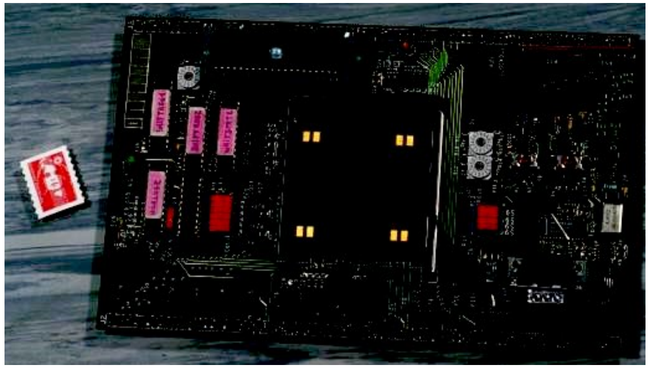
- Hybrid analog/digital: MIND-128 (1986)
- Fully digital: MIND-1024 (1991)







1990'S NEUROCOMPUTERS...



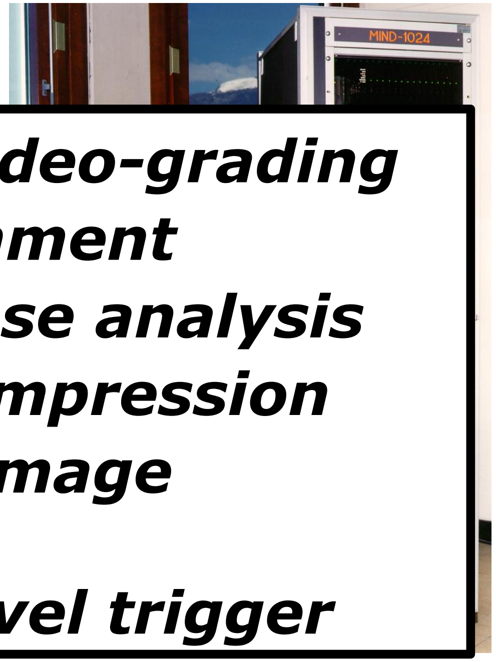
Philips : L-Neuro

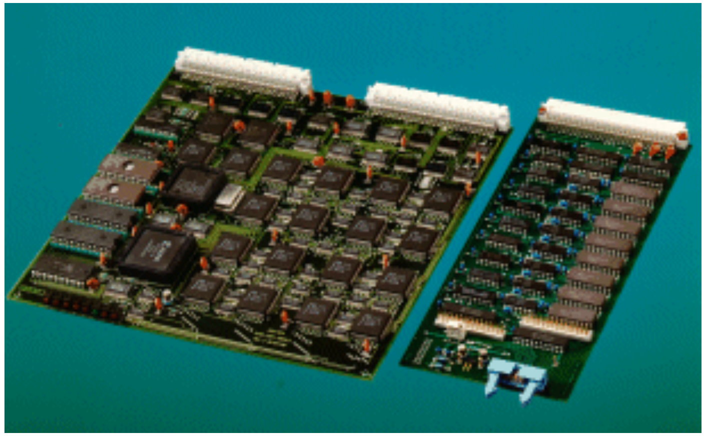
- 1st Gen 16 PEs 26 MCps (1990)
- 2nd Gen 12 PEs 720 MCps (1994)

➢ Used in satellite, fruit sorting, PCB inspection, sleep analysis, …

CEA's MIND machine

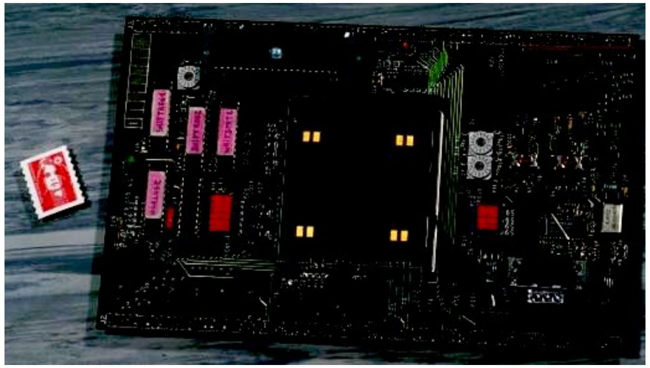
- Hybrid analog/digital: MIND-128 (…6)
- Fully digital: MIND-1024 (1991)





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

1990'S NEUROCOMPUTERS...



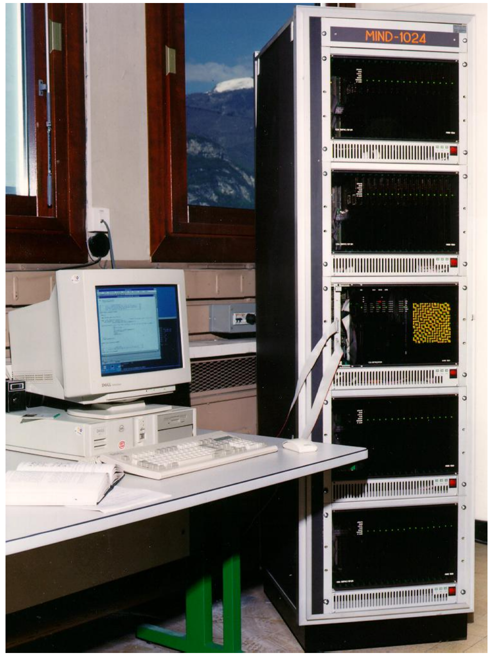
Philips : L-Neuro

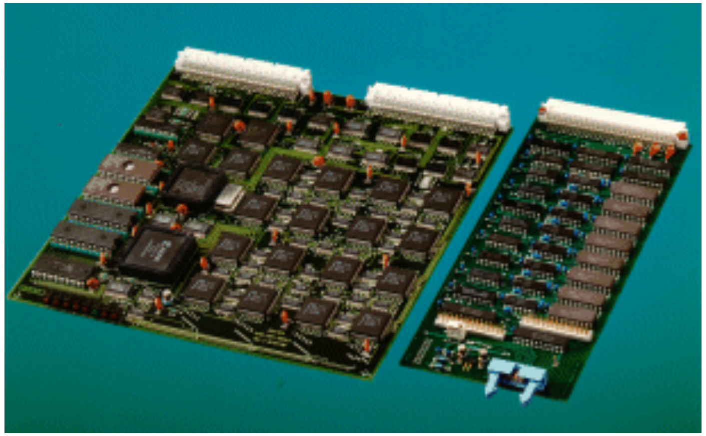
- 1st Gen 16 PEs 26 MCps (1990)
- 2nd Gen 12 PEs 720 MCps (1994)

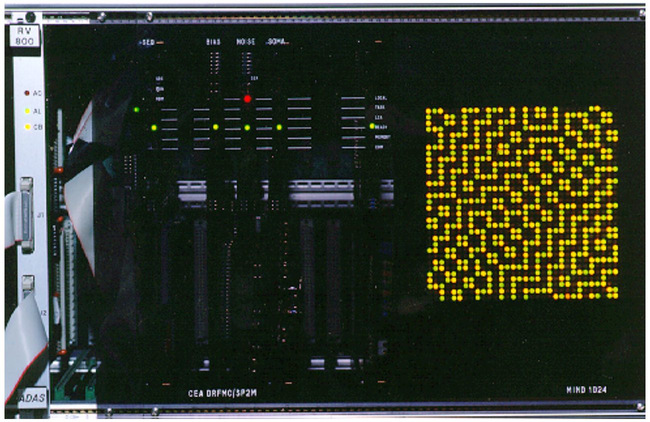
➢ Used in satellite, fruit sorting, PCB inspection, sleep analysis, …

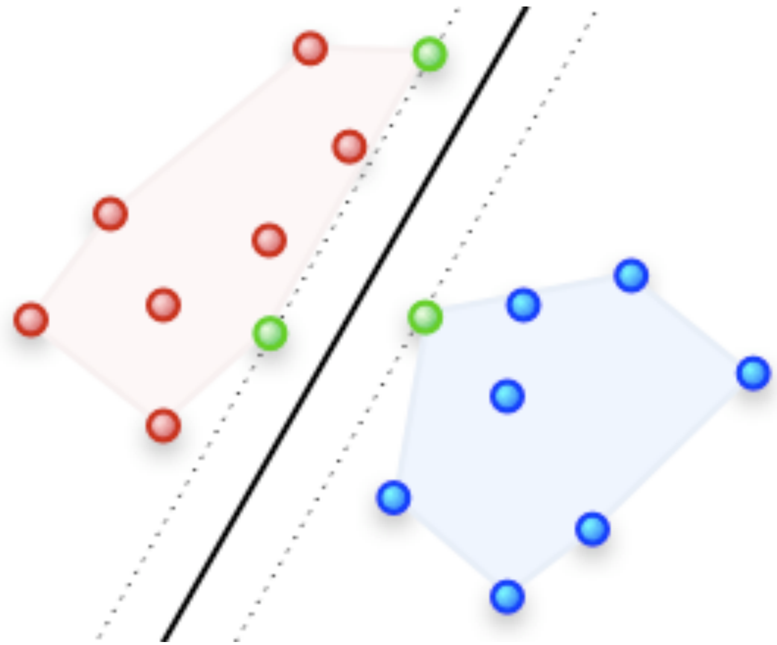
CEA's MIND machine

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









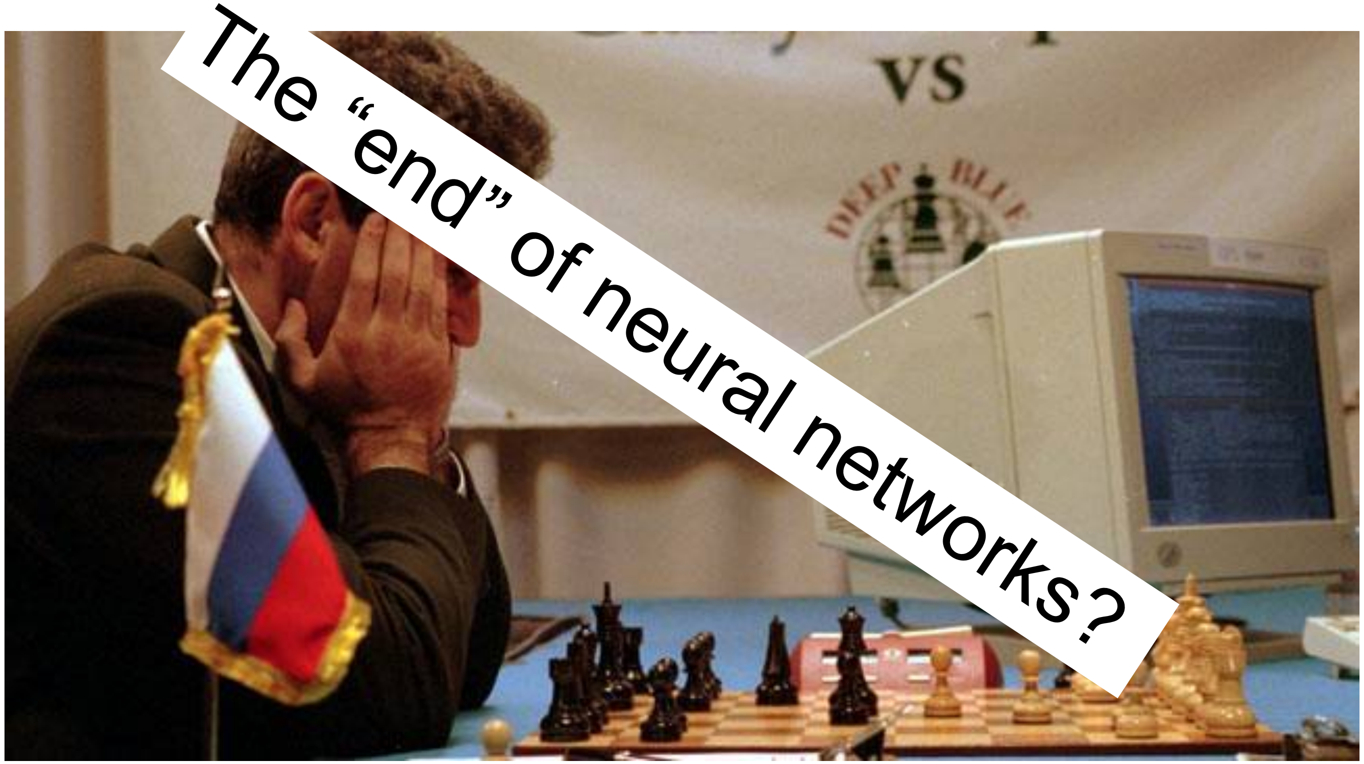
Support Vector Machines (SVMs)
The original SVM algorithm was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in **1963**.

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.

As far back as the mid-60s, chess was called the "Drosophila of artificial intelligence" – a reference to the fruit flies biologists used to uncover the secrets of genetics –
1997 – Deep Blue wins a six-game match against Garry Kasparov.

1997: CHESS AND DEEP BLUE

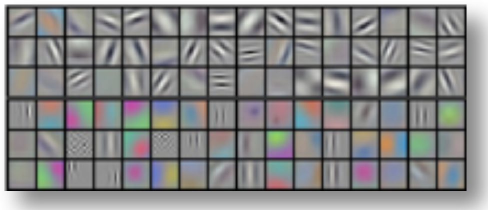
As far back as the mid-60s, chess was called the "Drosophila of artificial intelligence" – a reference to the fruit flies biologists used to uncover the secrets of genetics – 1997 – Deep Blue wins a six-game match against Garry Kasparov.



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

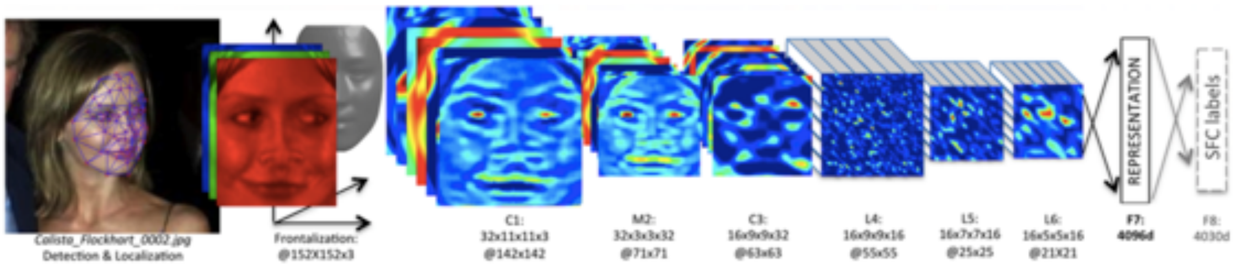


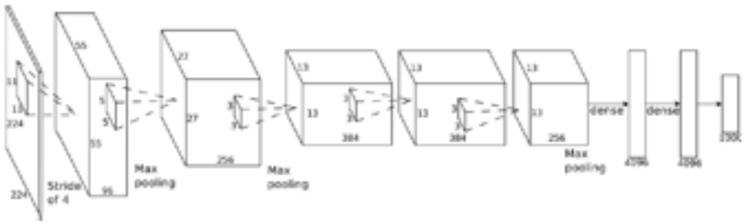
Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than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

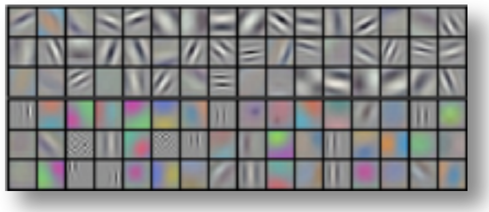
From: Y. Taigman, M. Yang, M.A. Ranzato, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification"

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

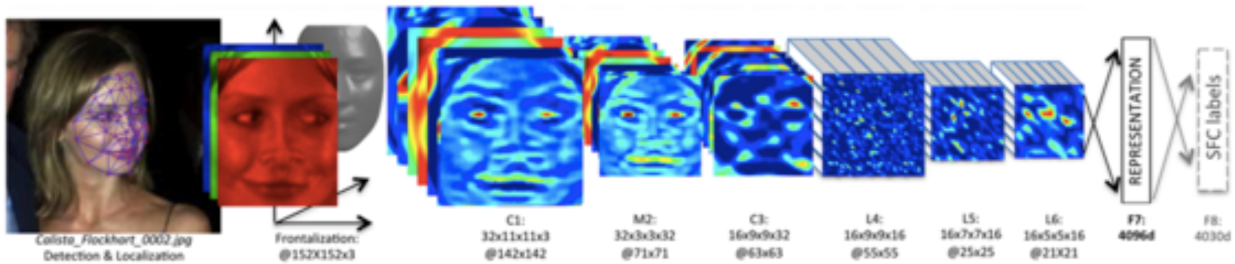


Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than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

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%)

research highlights

DOI:10.1145/3065386

ImageNet Classification with Deep Convolutional Neural Networks

By Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0%, respectively, which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully connected layers we employed a recently developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

that were widely investigated in the 1980s. These networks used multiple layers of feature detectors that were all learned from the training data. Neuroscientists and psychologists had hypothesized that a hierarchy of such feature detectors would provide a robust way to recognize objects but they had no idea how such a hierarchy could be learned. There was great excitement in the 1980s because several different research groups discovered that multiple layers of feature detectors could be trained efficiently using a relatively straight-forward algorithm called backpropagation[18, 22, 27, 33] to compute, for each image, how the classification performance of the whole network depended on the value of the weight on each connection.

Backpropagation worked well for a variety of tasks, but in the 1980s it did not live up to the very high expectations of its advocates. In particular, it proved to be very difficult to learn networks with many layers and these were precisely the networks that should have given the most impressive results. Many researchers concluded, incorrectly, that learning a deep neural network from random initial weights was just too difficult. Twenty years later, we know what went wrong: for

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

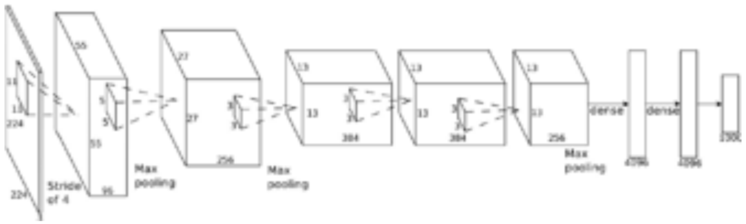
y Y. LeCun)

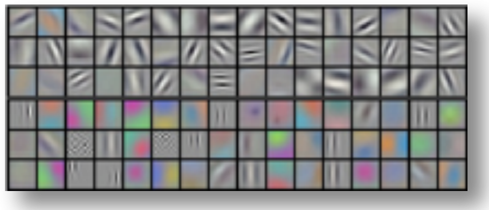
rom:Y. Taigman, M. Yang, M.A. Ranzato,
DeepFace: Closing the Gap to Human-Level
erformance in Face Verification"

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

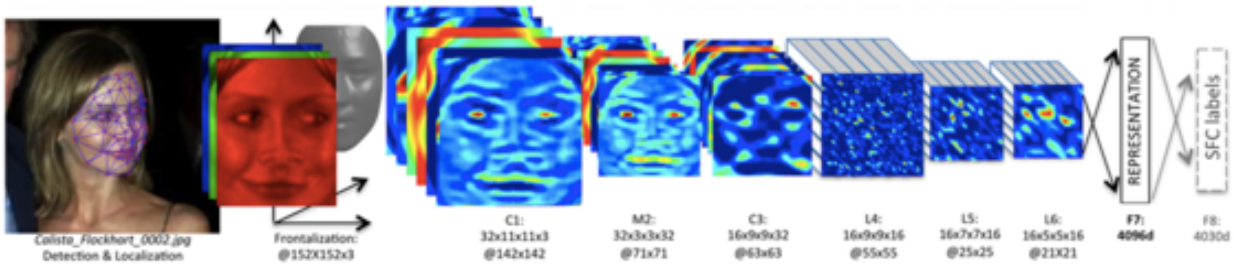


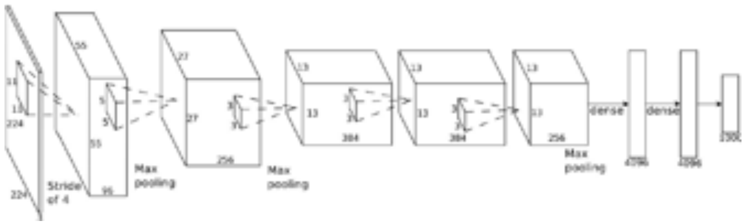
Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than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

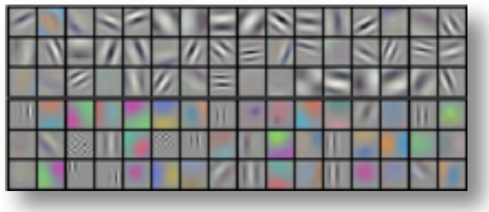
From:Y. Taigman, M. Yang, M.A. Ranzato, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification"

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)**

The 2018 **Turing Award recipients** are Google VP Geoffrey Hinton*, Facebook's Yann LeCun and Yoshua Bengio, Scientific Director of AI research center Mila.

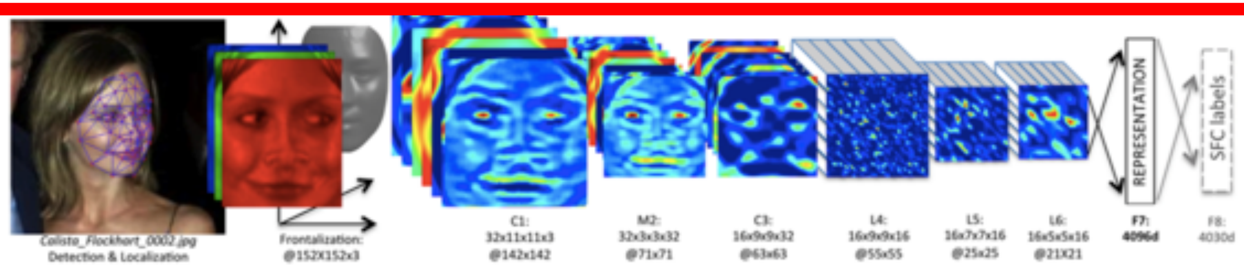


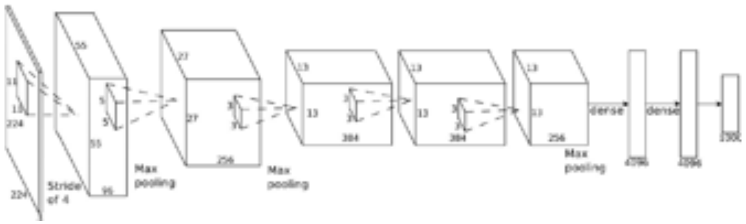
Figure 2. **Outline of the *DeepFace* architecture.** A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than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

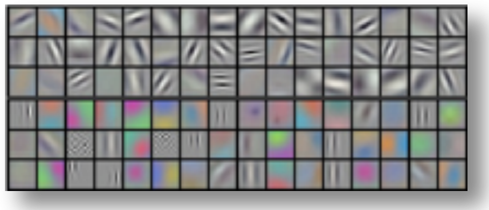
From: Y. Taigman, M. Yang, M.A. Ranzato, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification"

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)**

The 2018 **Turing Award recipients** are Google VP Geoffrey Hinton*, Facebook's Yann LeCun and Yoshua Bengio, Scientific Director of AI research center Mila.

* He was also awarded with John Hopfield the 2024 Nobel Prize in Physics for "foundational discoveries and inventions that enable machine learning with artificial neural networks"

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.

ImageNet: Classification

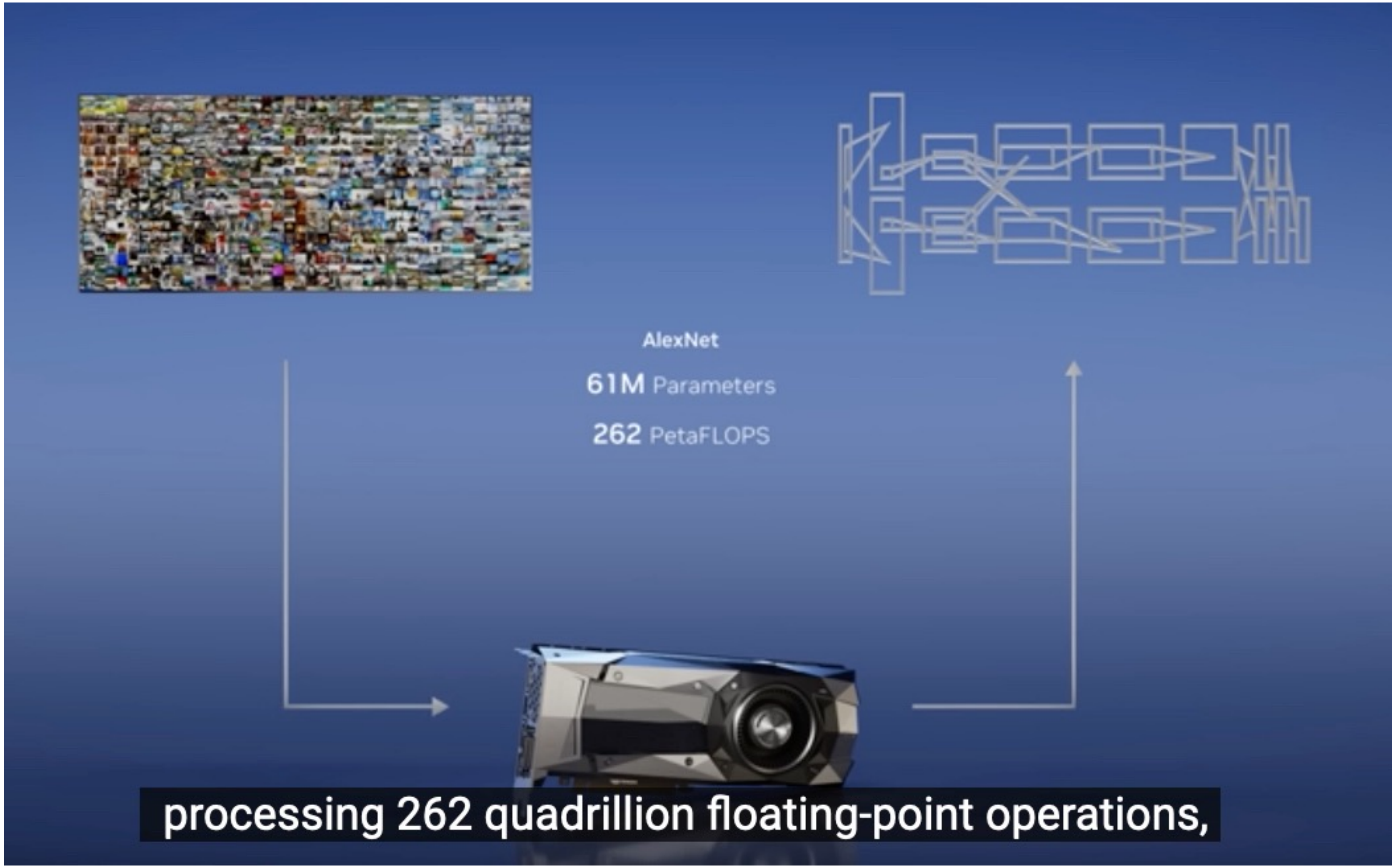
- Give the name of the dominant object in the image
- Top-5 error rates: if correct class is not in top 5, count as error
  - Black:ConvNet, Purple: no ConvNet

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

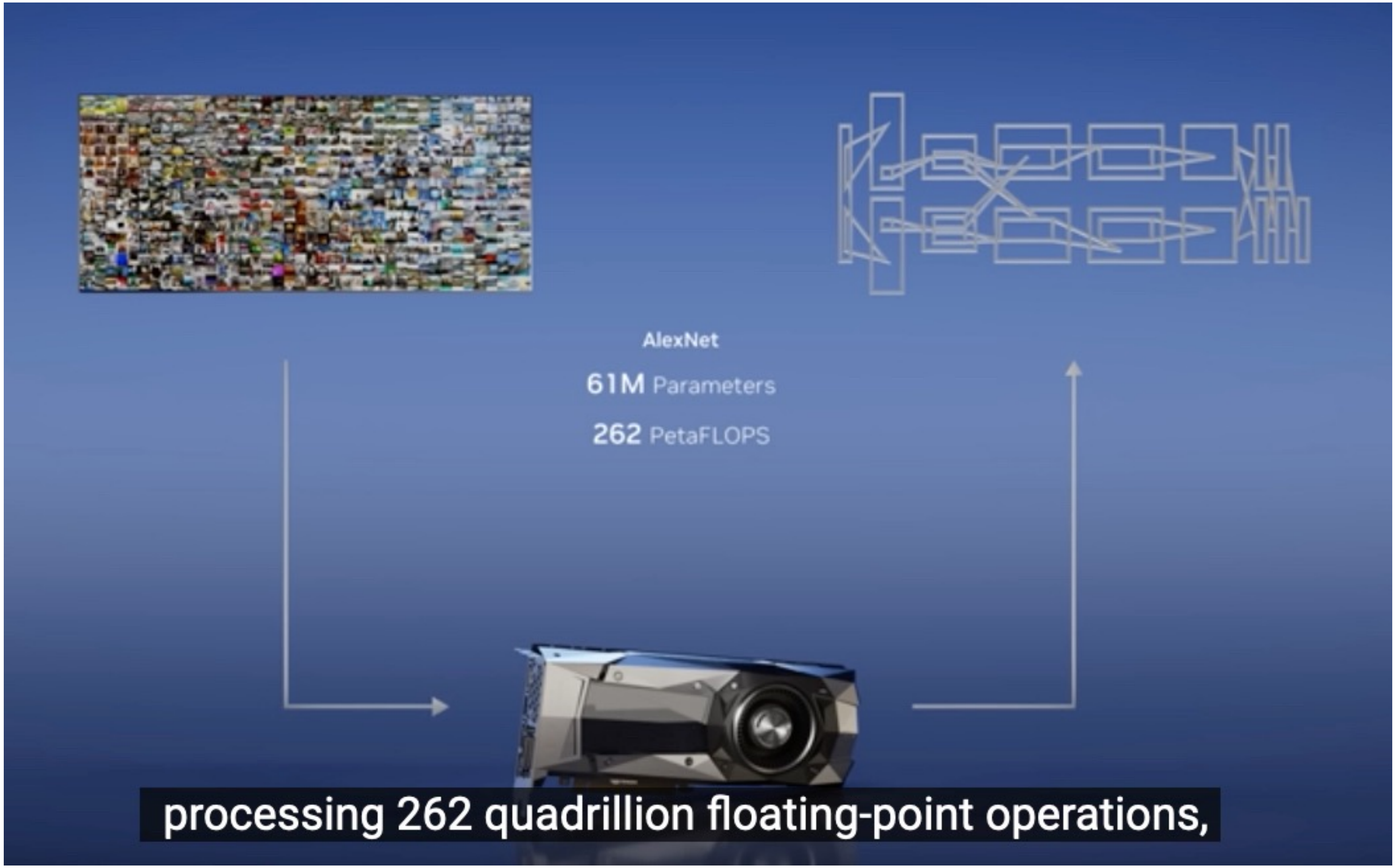
2014 Teams	%error
GoogLeNet	6.6
VGG (Oxford)	7.3
MSRA	8.0
A. Howard	8.1
DeeperVision	9.5
NUS-BST	9.7
TTIC-ECP	10.2
XYZ	11.2
UvA	12.1

Computing power is driving the advance of AI



From GTC 2023 Keynote with NVIDIA CEO Jensen Huang

Computing power is driving the advance of AI



2012: AlexNet
GeForce GTX 580
Won ImageNet Challenge
262×10^{15} FLOPS

From GTC 2023 Keynote with NVIDIA CEO Jensen Huang

IMAGE ROI EXTRACTION AND CLASSIFICATION

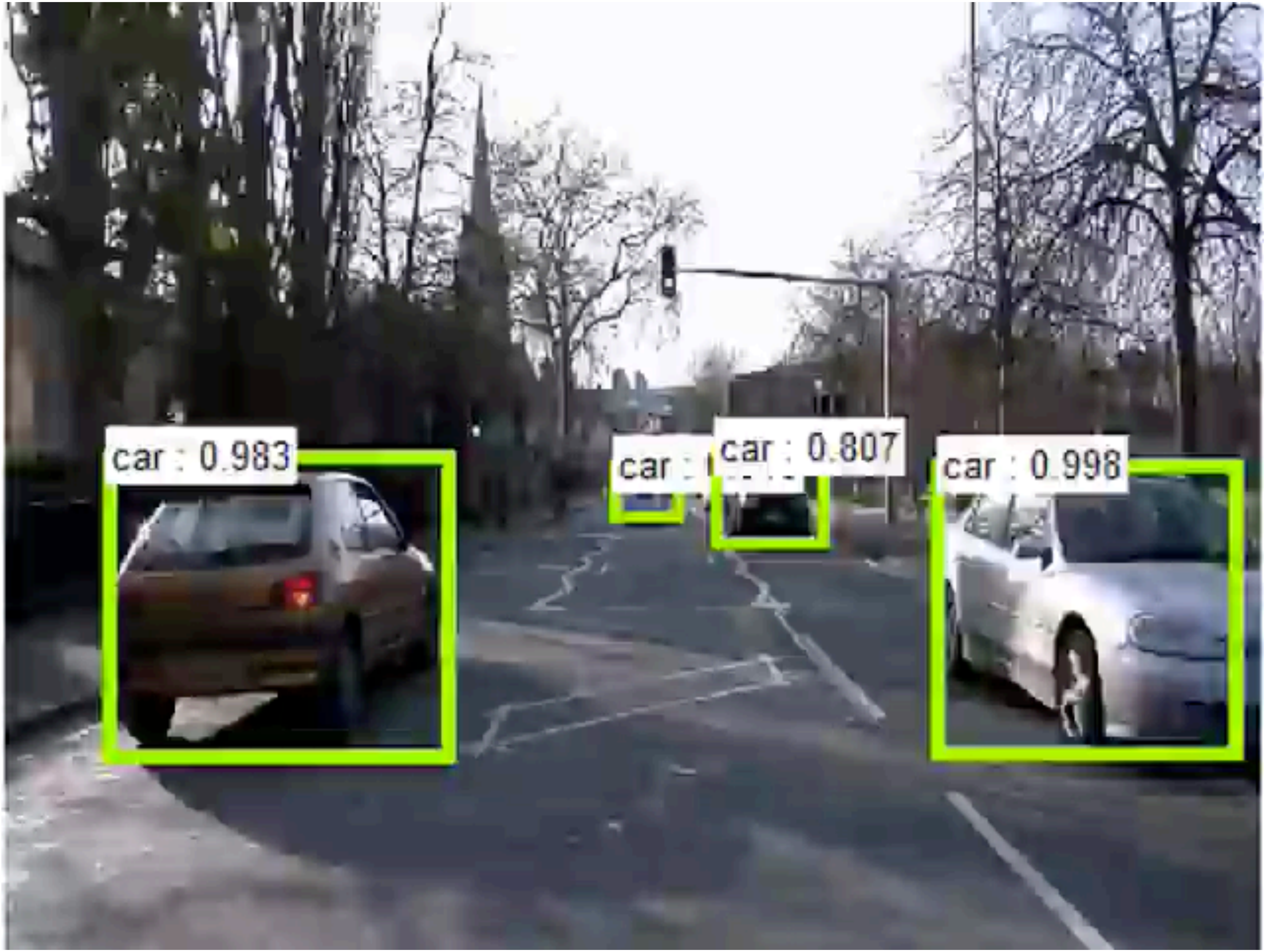


IMAGE ROI EXTRACTION AND CLASSIFICATION

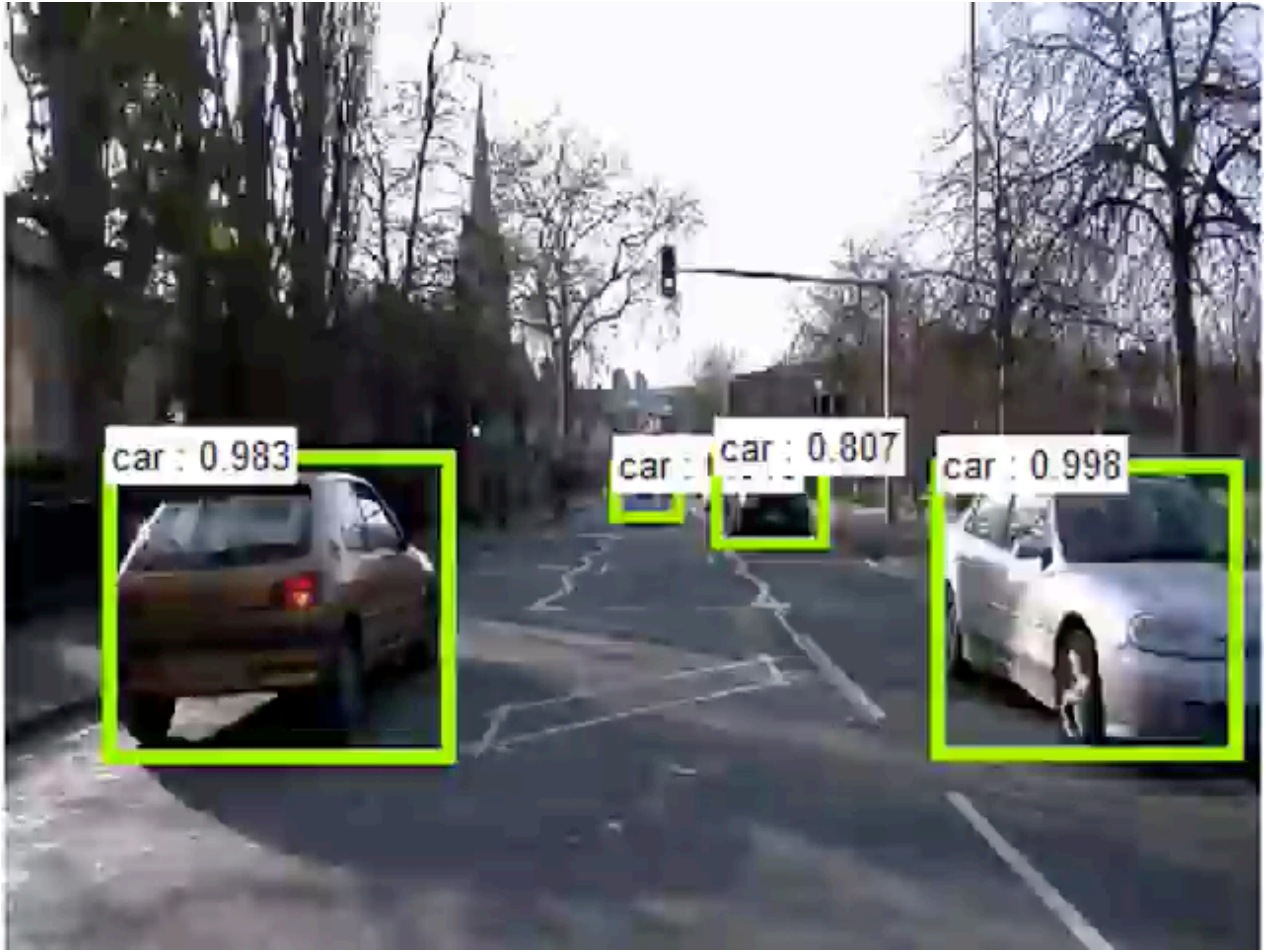
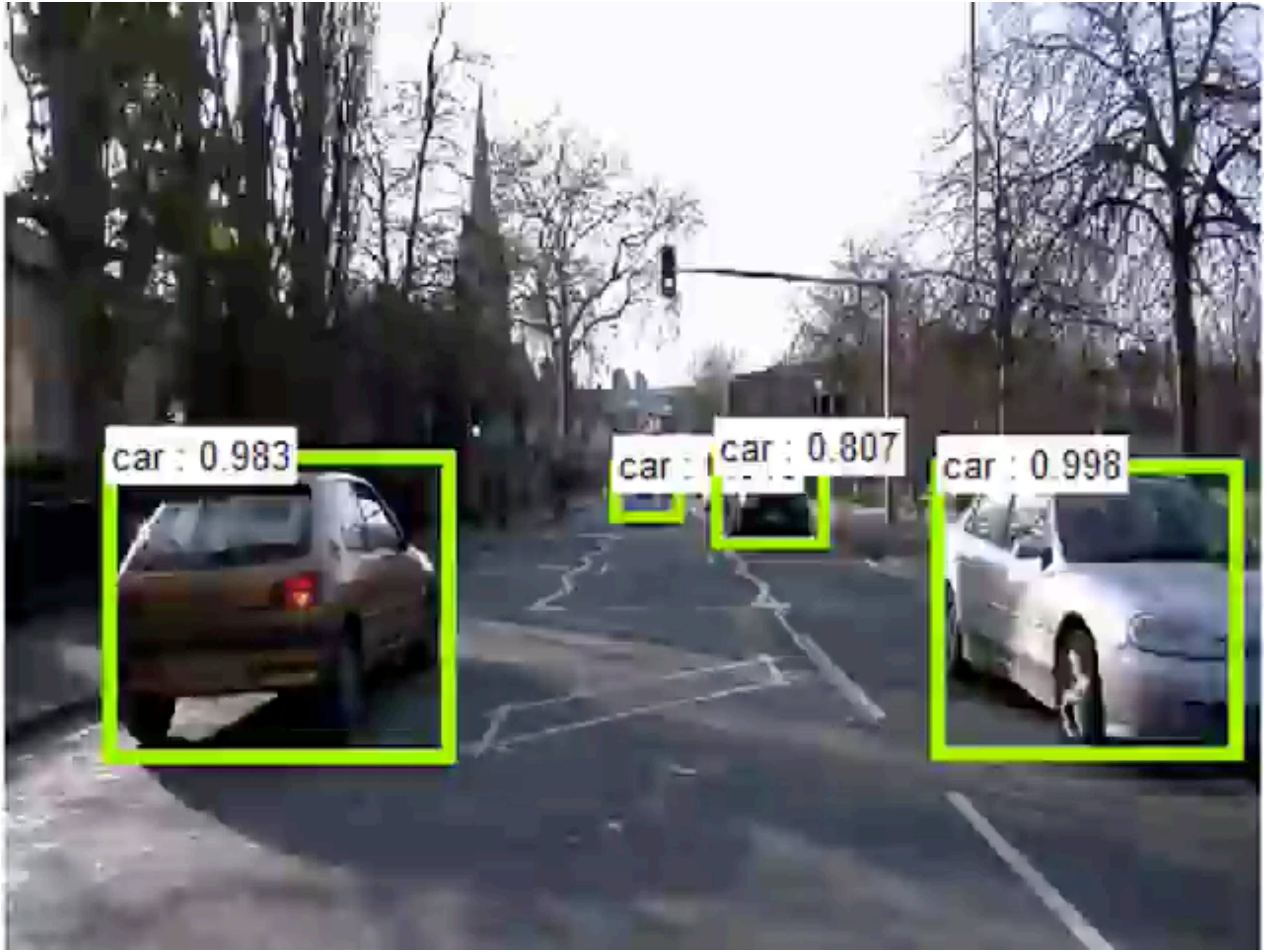


IMAGE ROI EXTRACTION AND CLASSIFICATION



FROM RESEARCH TO INDUSTRY

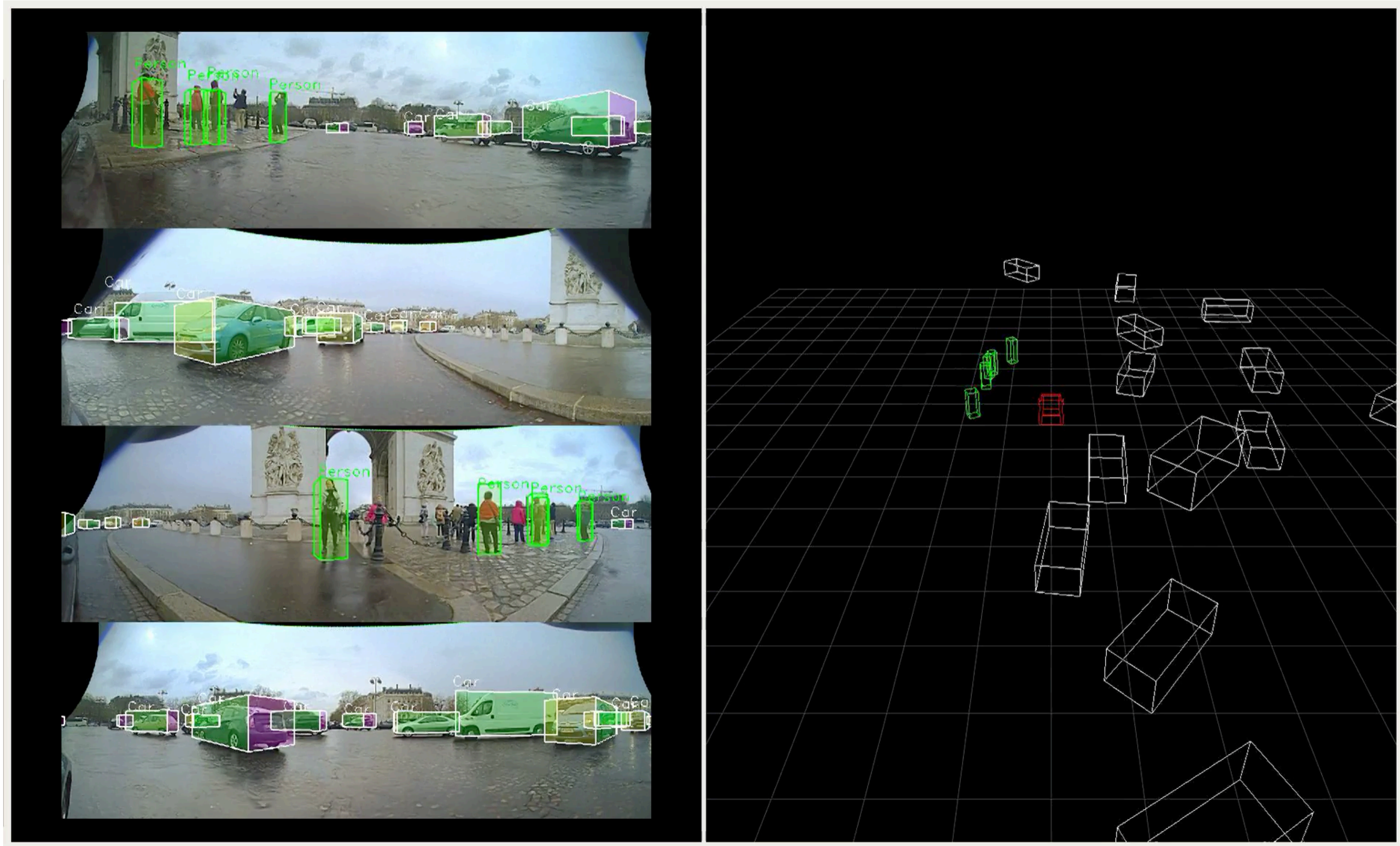
cea tech

CES 2018

Valeo

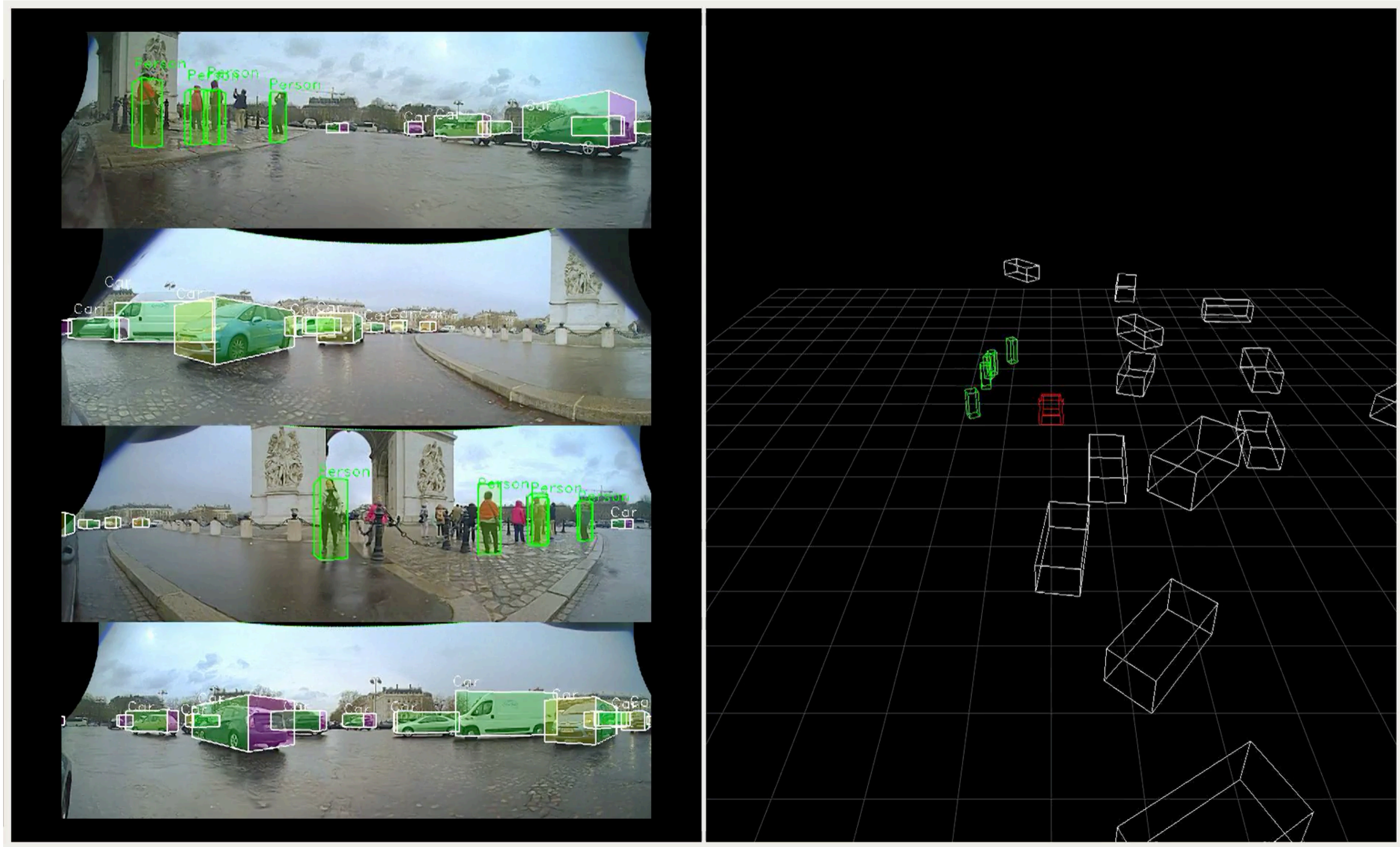
DEEP MANTA

MANY-TASK DEEP NEURAL NETWORK FOR VISUAL OBJECT RECOGNITION



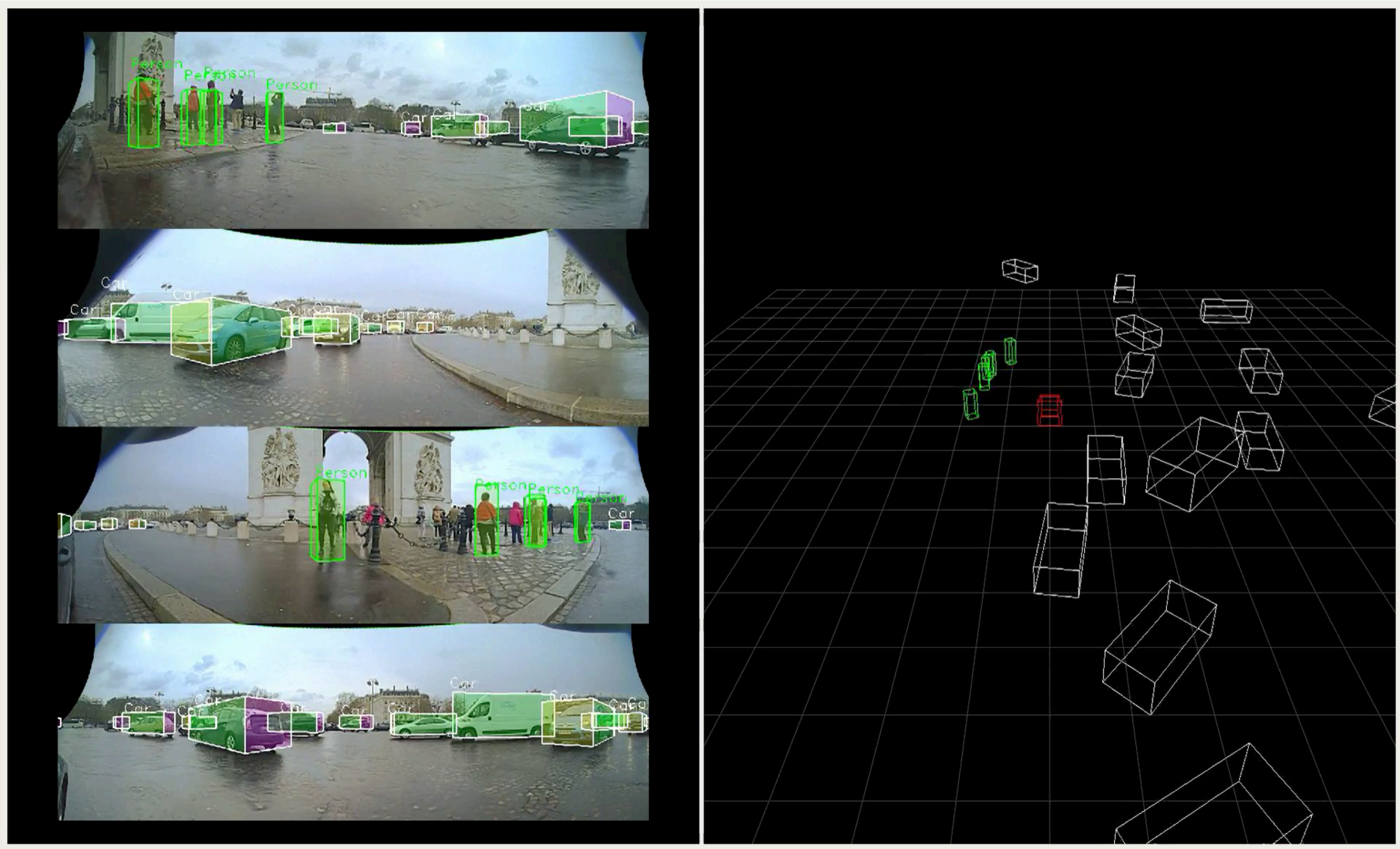
DEEP MANTA

MANY-TASK DEEP NEURAL NETWORK FOR VISUAL OBJECT RECOGNITION



DEEP MANTA

MANY-TASK DEEP NEURAL NETWORK
FOR VISUAL OBJECT RECOGNITION



2025: Tesla’s FSD on the same location

2025: Tesla’s FSD on the same location

2025: Tesla's FSD on the same location

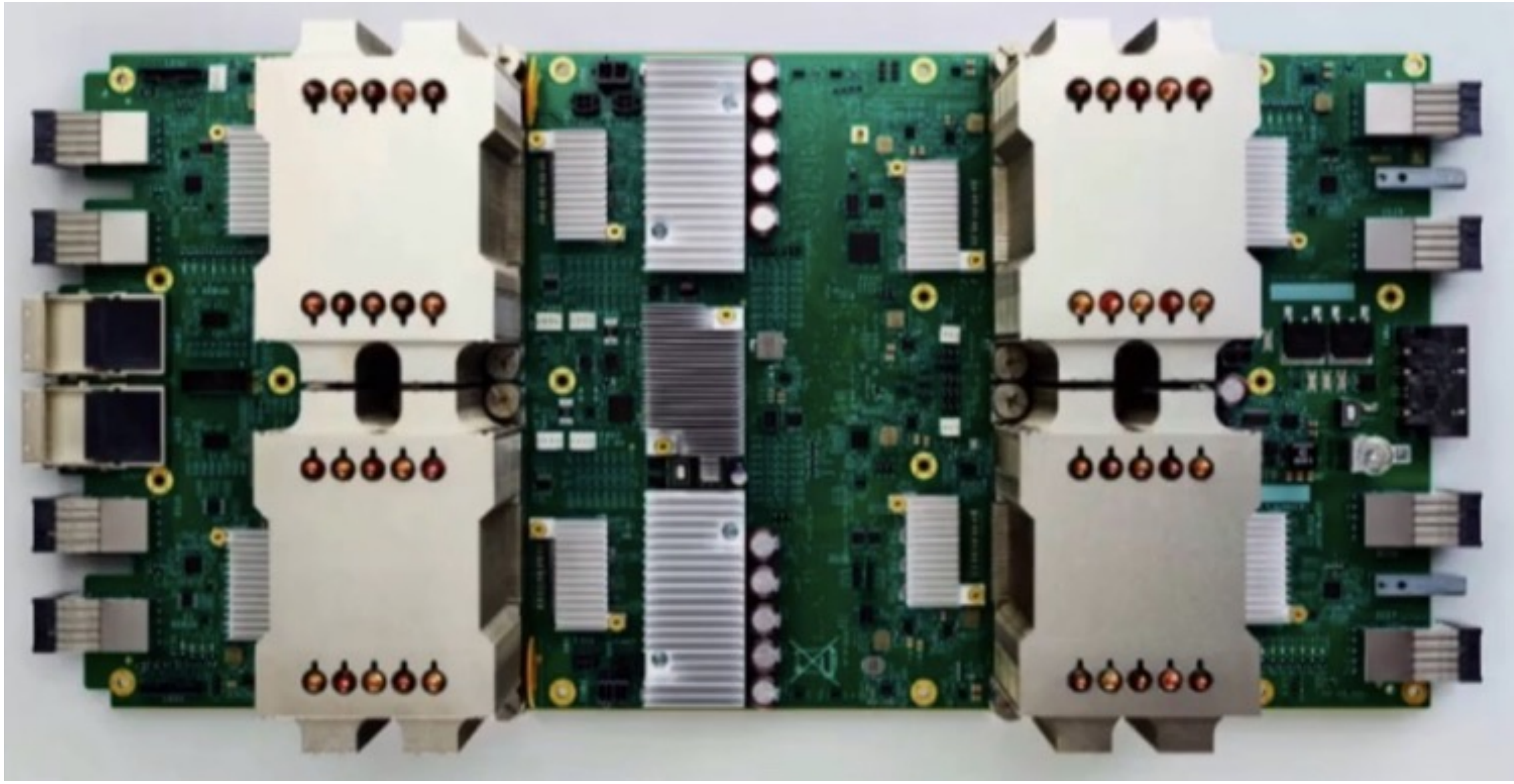
DEEP LEARNING AND VOICE RECOGNITION

" The need for TPUs really emerged about six (12) years ago, when we started using computationally expensive deep learning models in more and more places throughout our products. The computational expense of using these models had us worried. If we considered a scenario where people use Google voice search for just three minutes a day and we ran deep neural nets for our speech recognition system on the processing units we were using, we would have had to *double the number of Google data centers*!"

[https://cloudplatform.googleblog.com/2017/04/quantifying-the-performance-of-the-TPU-our-first-machine-learning-chip.html]

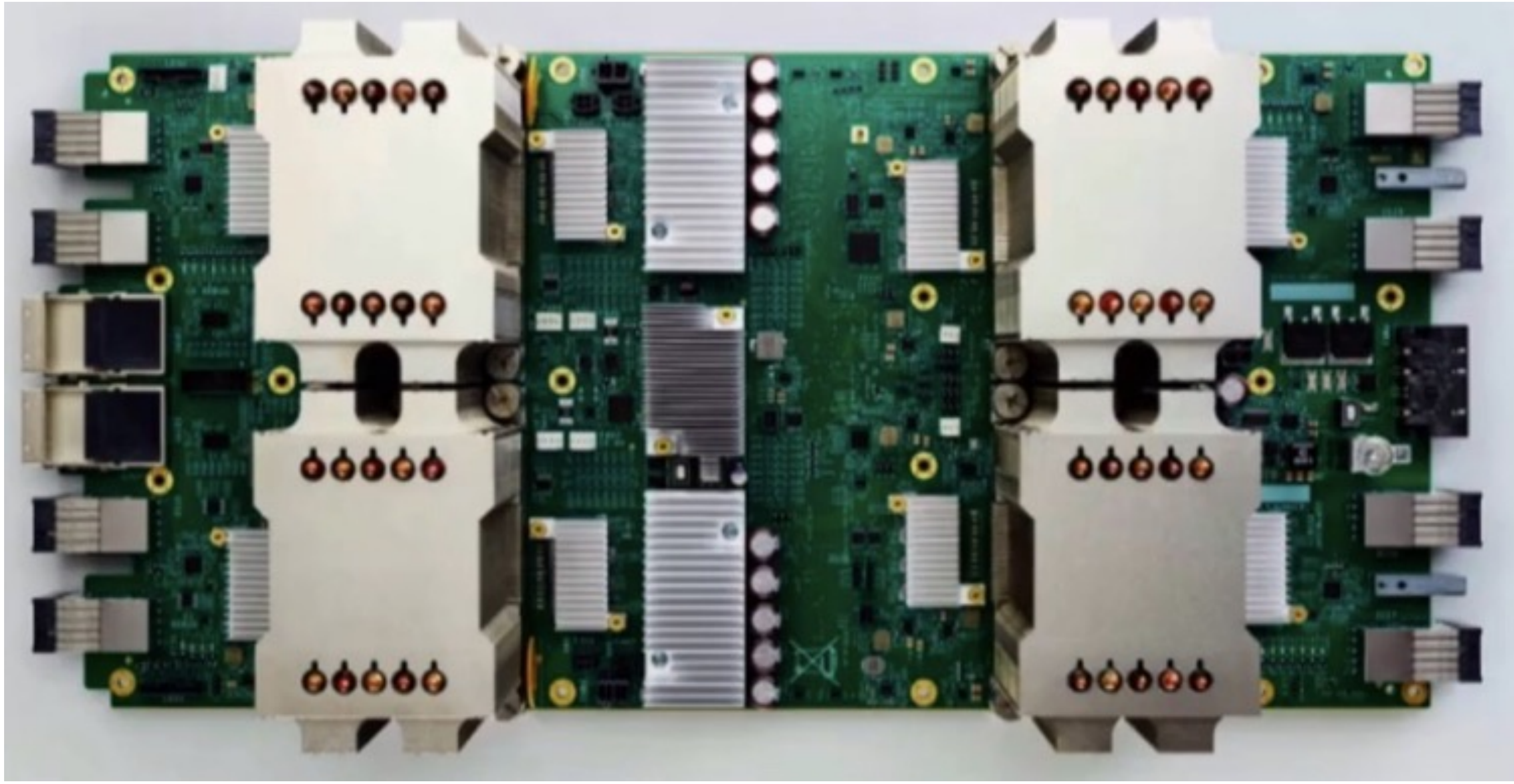
2017: GOOGLE'S CUSTOMIZED HARDWARE…

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



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)

2017: GOOGLE'S CUSTOMIZED TPU HARDWARE…

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

Peta = 10^{15} = million of milliard

2017: GOOGLE'S CUSTOMIZED TPU HARDWARE…

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

Google's TPU2 : 11.5 petaflops_{16} of machine learning number crunching (and guessing about 400+ KW…, 100+ GFlops_{16}/W)

Peta = 10^{15} = million of milliard

2017: GOOGLE’S CUSTOMIZED TPU HARDWARE…

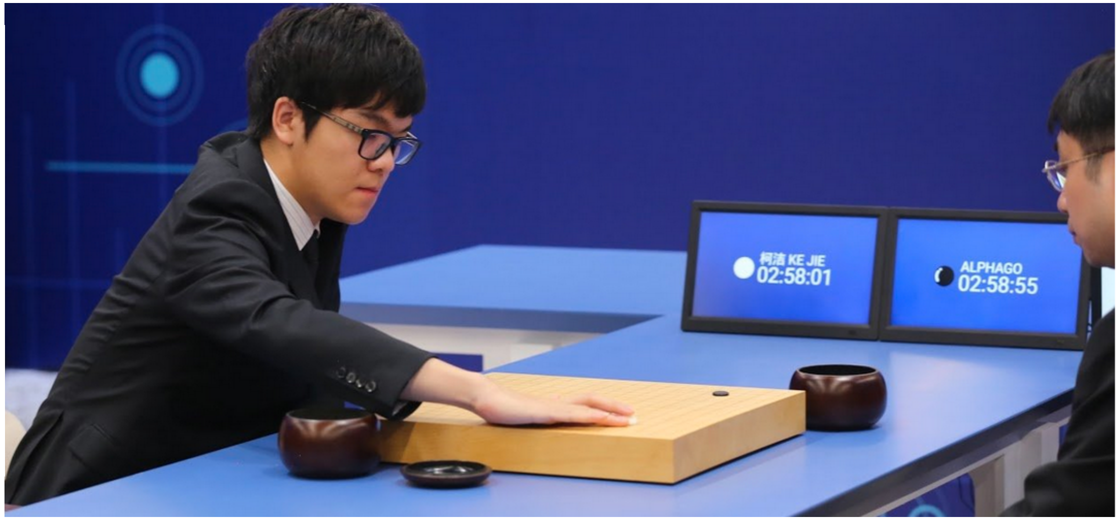
… required to increase energy efficiency
with accuracy adapted to the use (e.g. float 16)

Google’s TPU2 : 11.5 petaflops$_{16}$ of machine learning number crunching
(and guessing about 400+ KW…, 100+ GFlops$_{16}$/W)

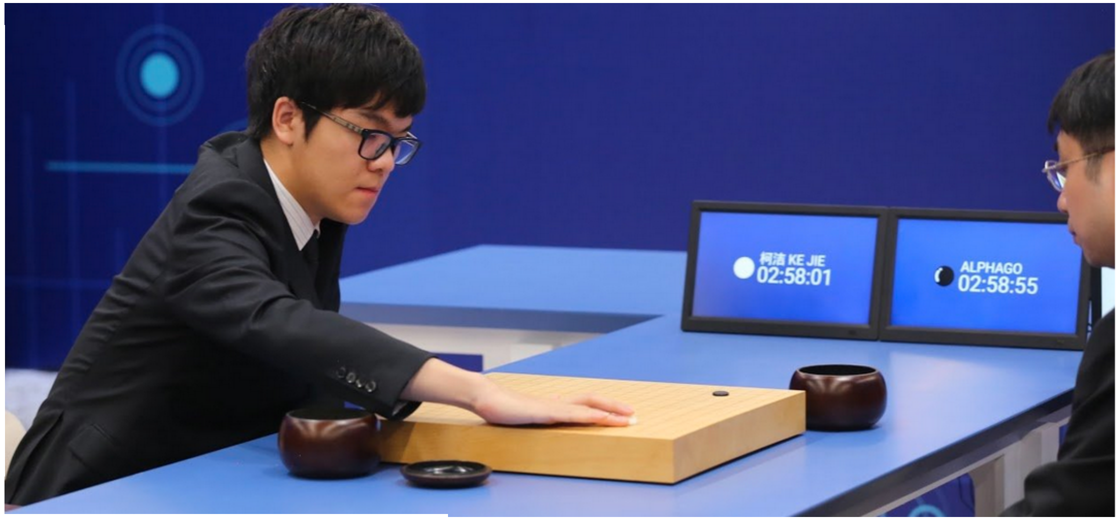
From Google

Peta = 10^{15} = million of milliard

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.



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.



ARTICLE

doi:10.1038/nature16961

Mastering the game of Go with deep neural networks and tree search

David Silver[1]*, Aja Huang[1]*, Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

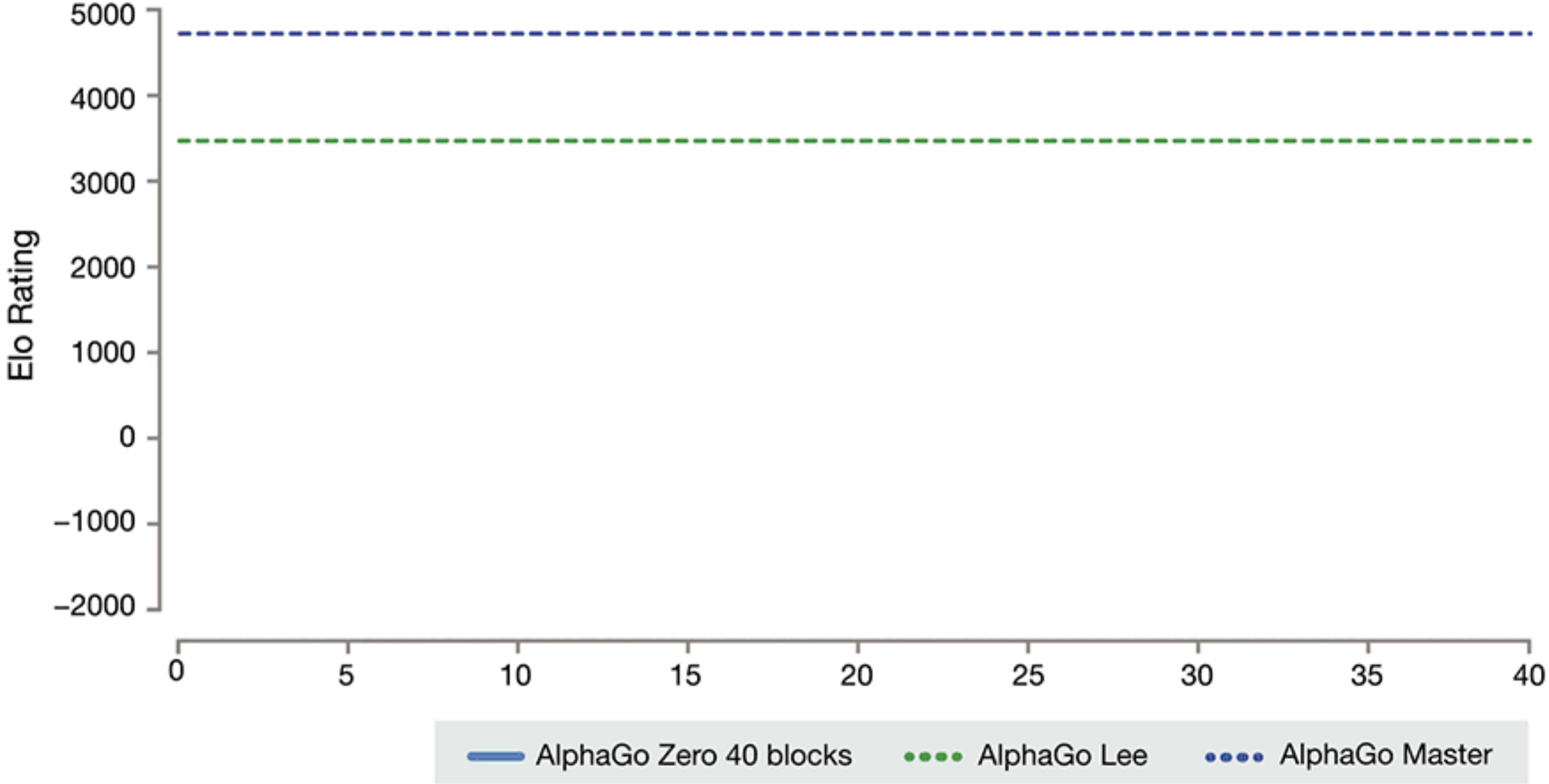
ALPHAGO ZERO: SELF-PLAYING TO LEARN

From doi:10.1038/nature24270 (Received 07 April 2017)

ALPHAGO ZERO: SELF-PLAYING TO LEARN

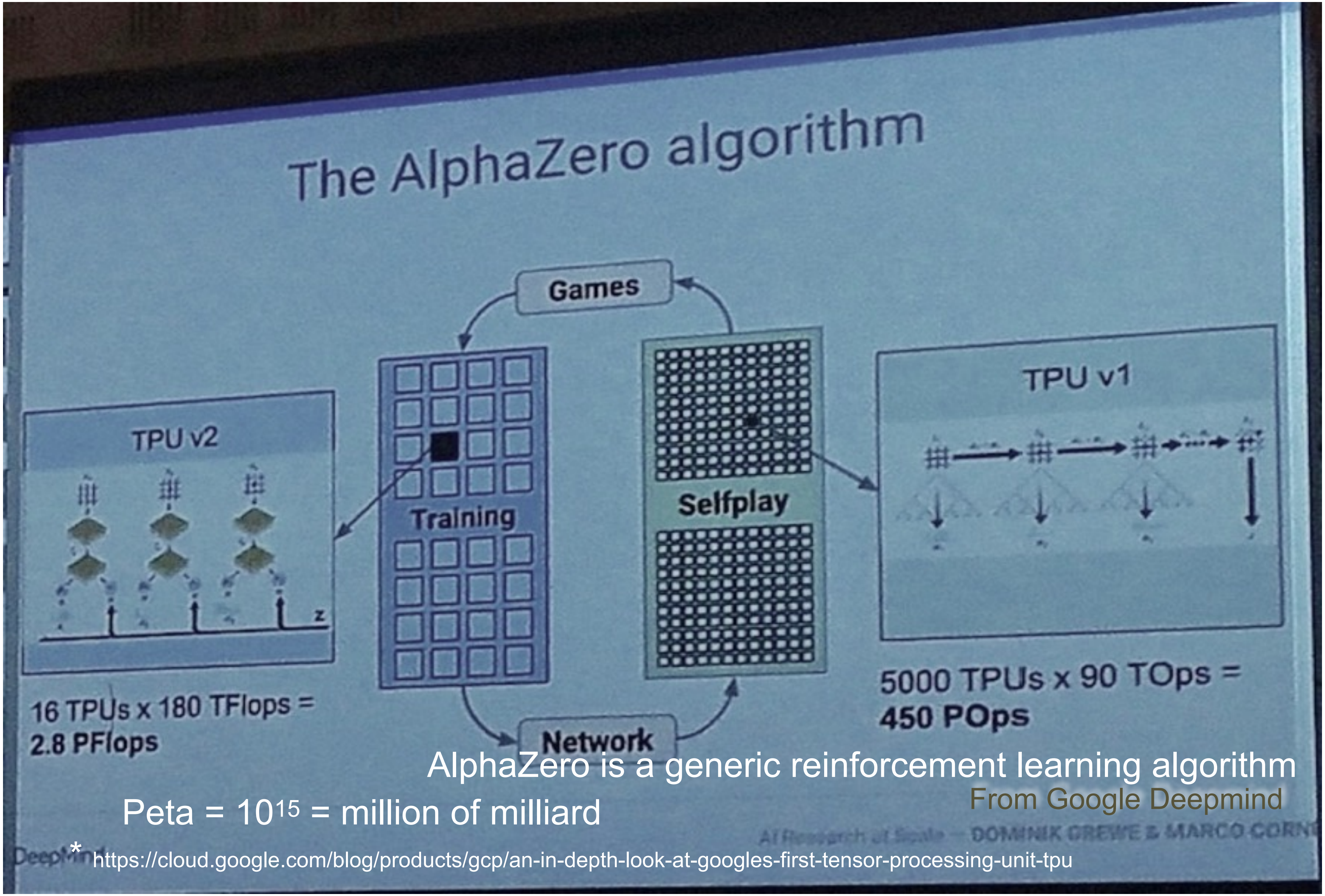
From doi:10.1038/nature24270 (Received 07 April 2017)

ALPHAGO ZERO: SELF-PLAYING TO LEARN



From doi:10.1038/nature24270 (Received 07 April 2017)

ALPHAZERO: COMPUTING RESOURCES



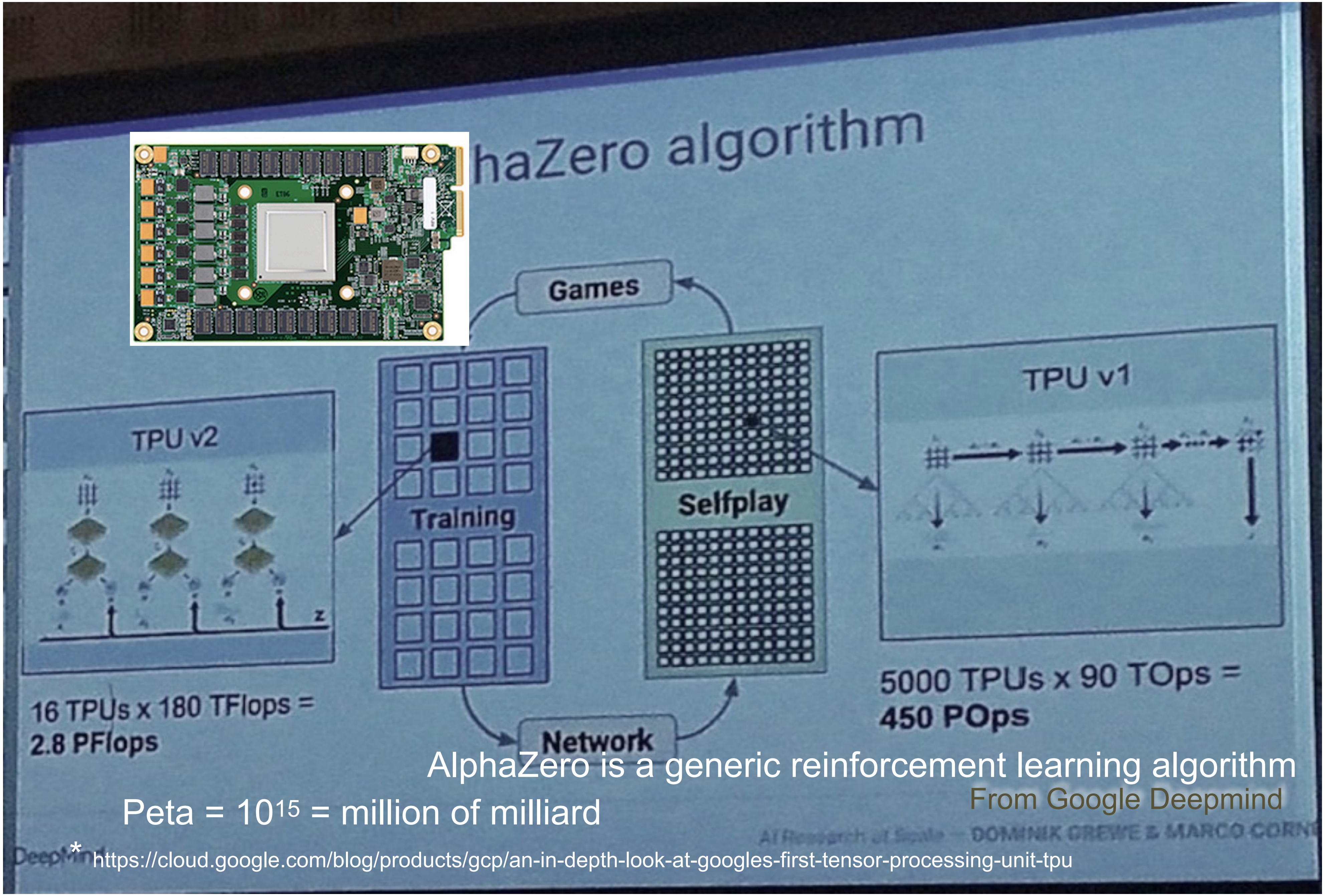
AlphaZero is a generic reinforcement learning algorithm

From Google Deepmind

Peta = 10^{15} = million of milliard

* https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

ALPHAZERO: COMPUTING RESOURCES



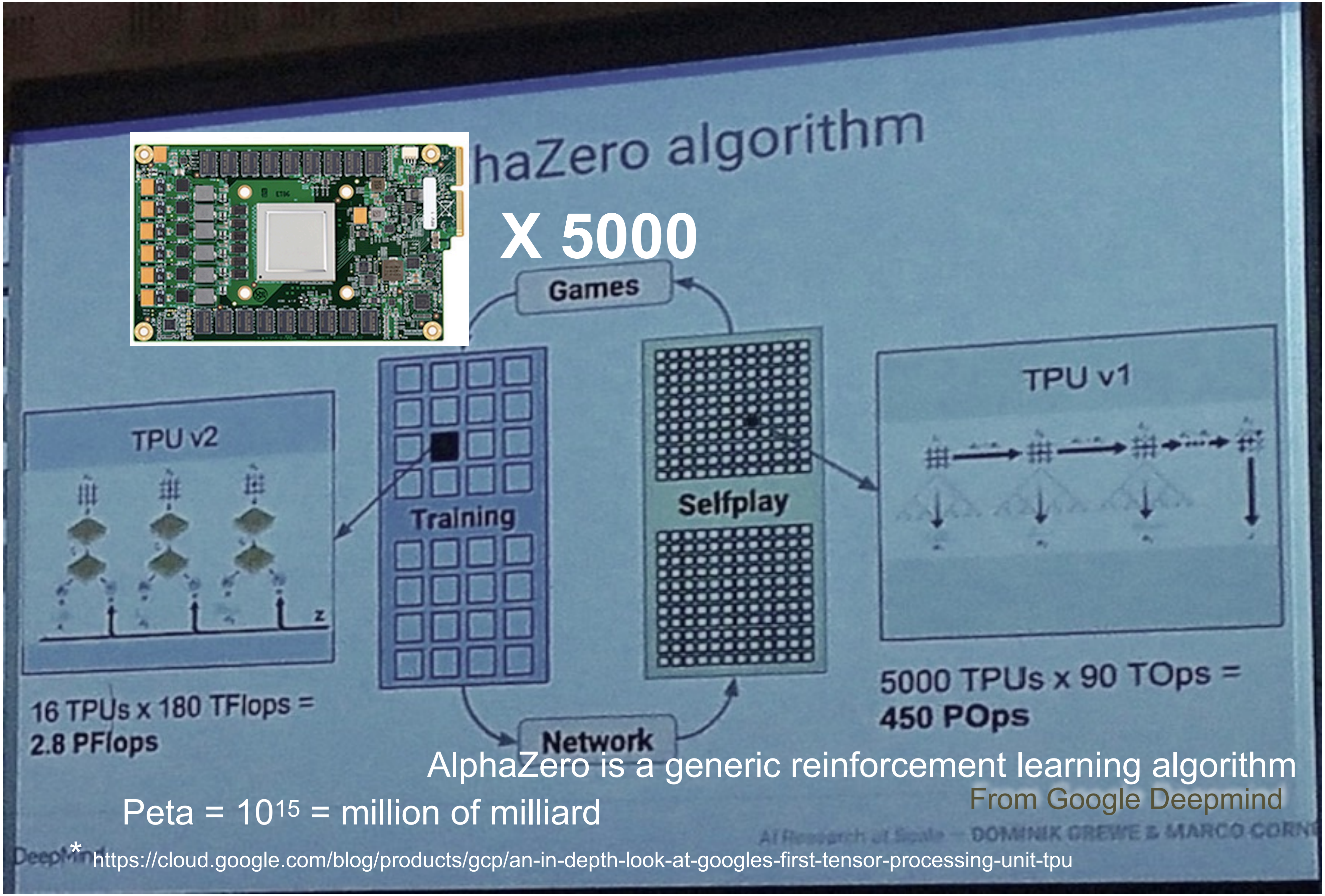
AlphaZero is a generic reinforcement learning algorithm

From Google Deepmind

Peta = 10^{15} = million of milliard

* https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

ALPHAZERO: COMPUTING RESOURCES



X 5000

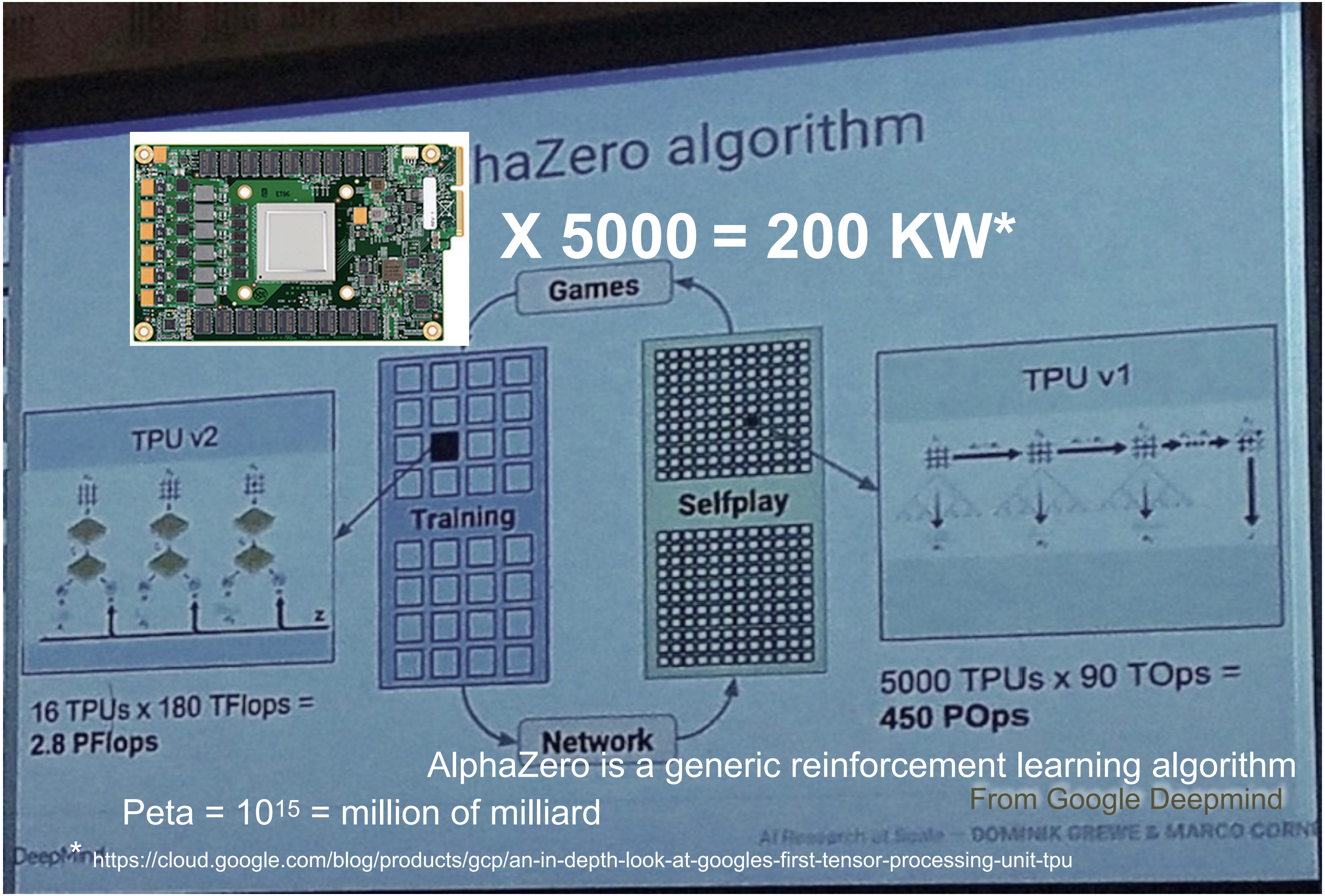
AlphaZero is a generic reinforcement learning algorithm

From Google Deepmind

Peta = 10^{15} = million of milliard

* https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

ALPHAZERO: COMPUTING RESOURCES



X 5000 = 200 KW*

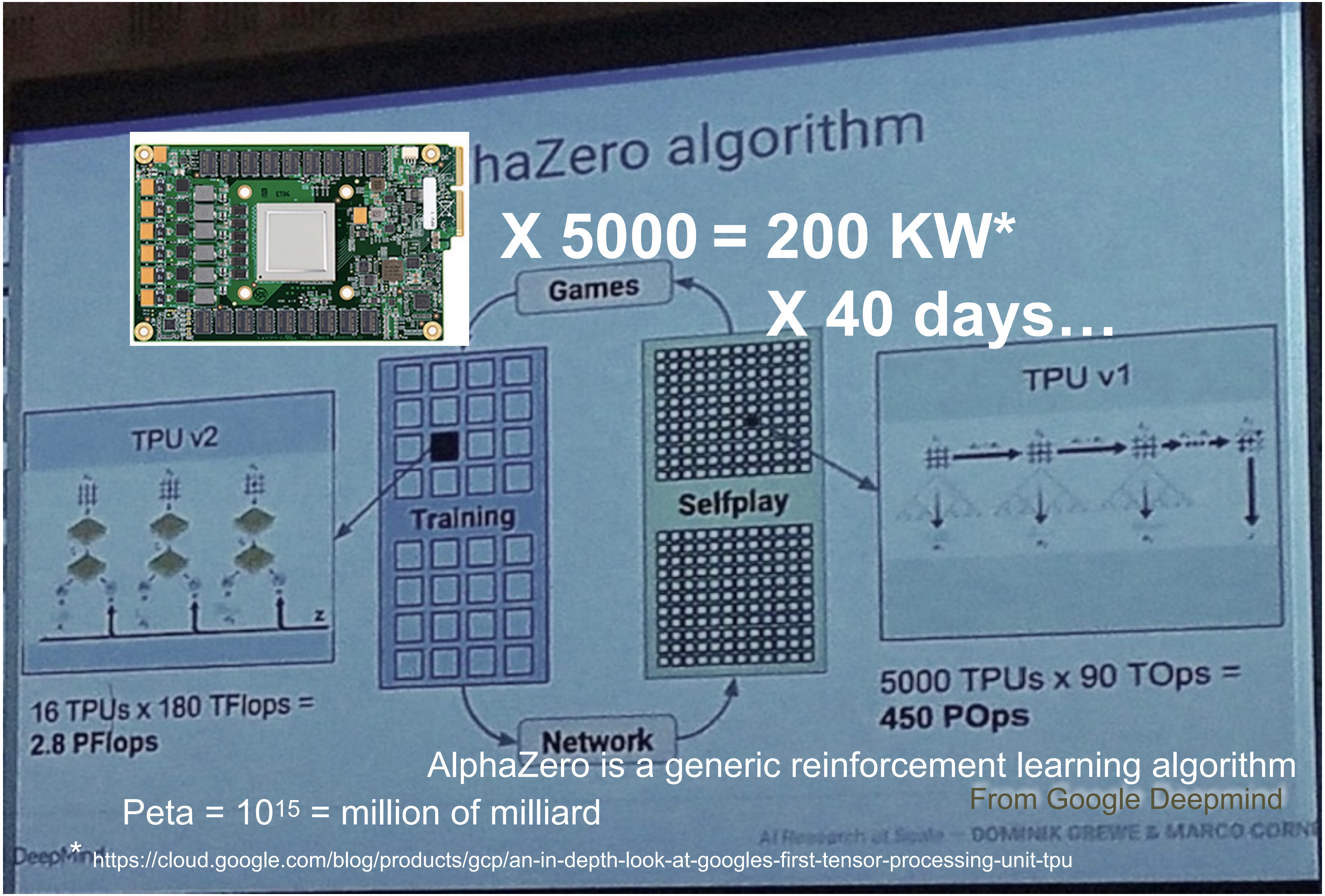
AlphaZero is a generic reinforcement learning algorithm

From Google Deepmind

Peta = 10^{15} = million of milliard

* https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

ALPHAZERO: COMPUTING RESOURCES



X 5000 = 200 KW*

X 40 days…

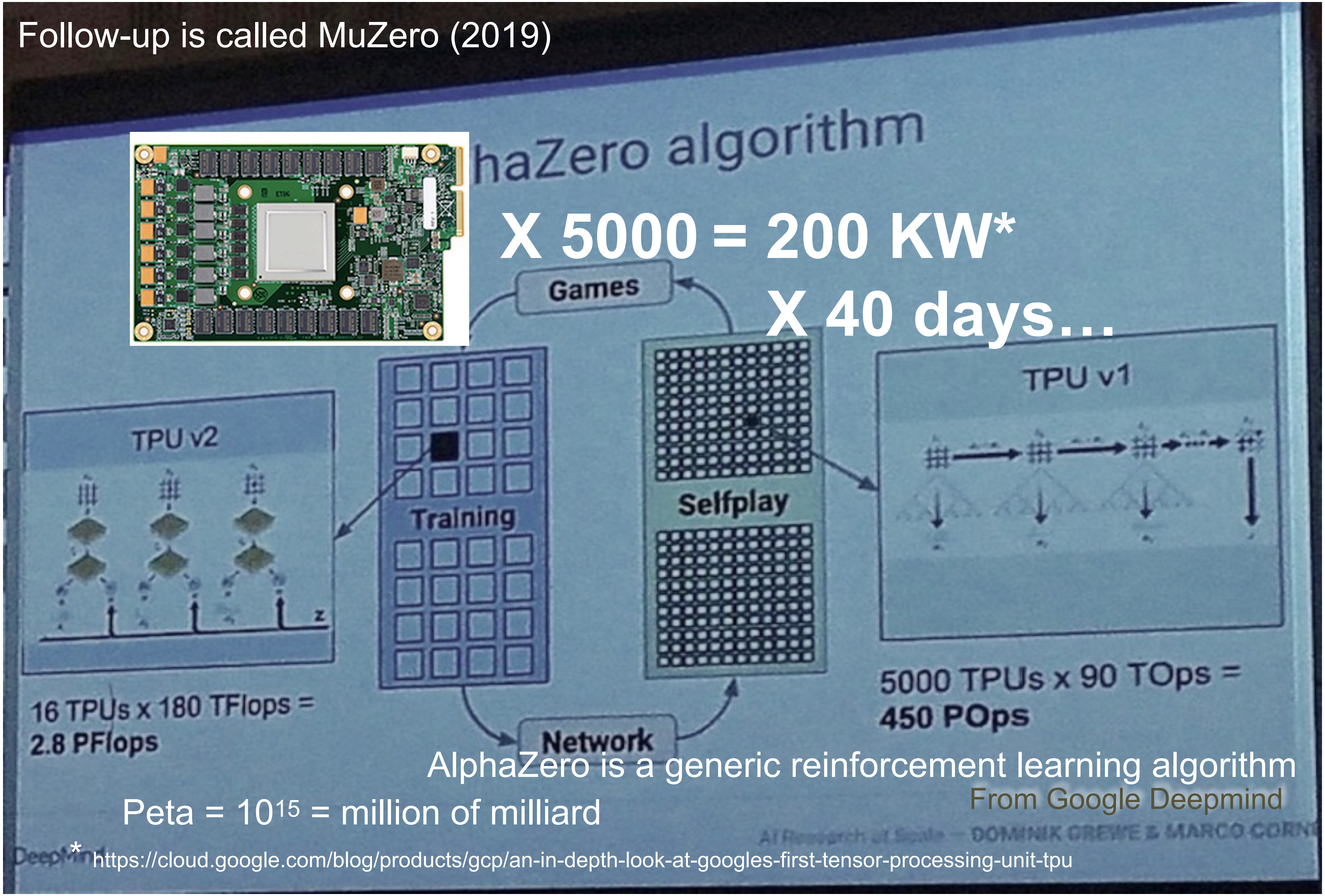
AlphaZero is a generic reinforcement learning algorithm

From Google Deepmind

Peta = 10^{15} = million of milliard

* https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

ALPHAZERO: COMPUTING RESOURCES



Follow-up is called MuZero (2019)

X 5000 = 200 KW*

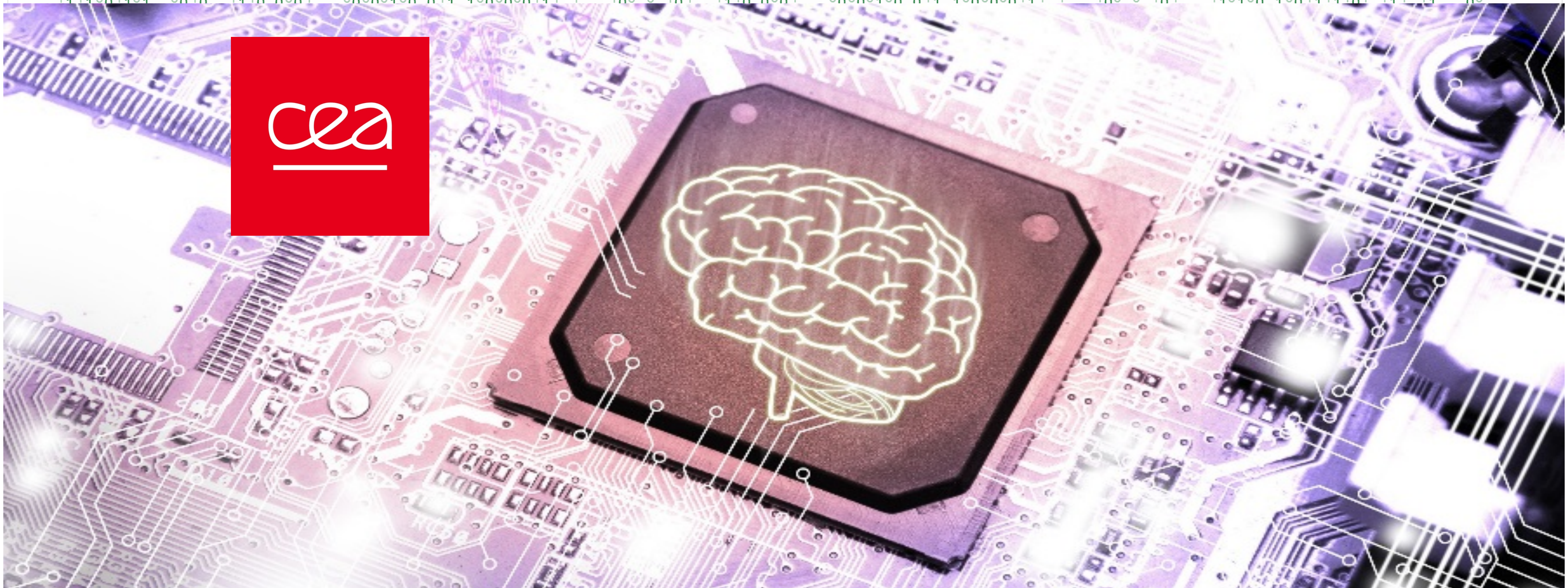
X 40 days…

AlphaZero is a generic reinforcement learning algorithm

From Google Deepmind

Peta = 10^{15} = million of milliard

* https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu



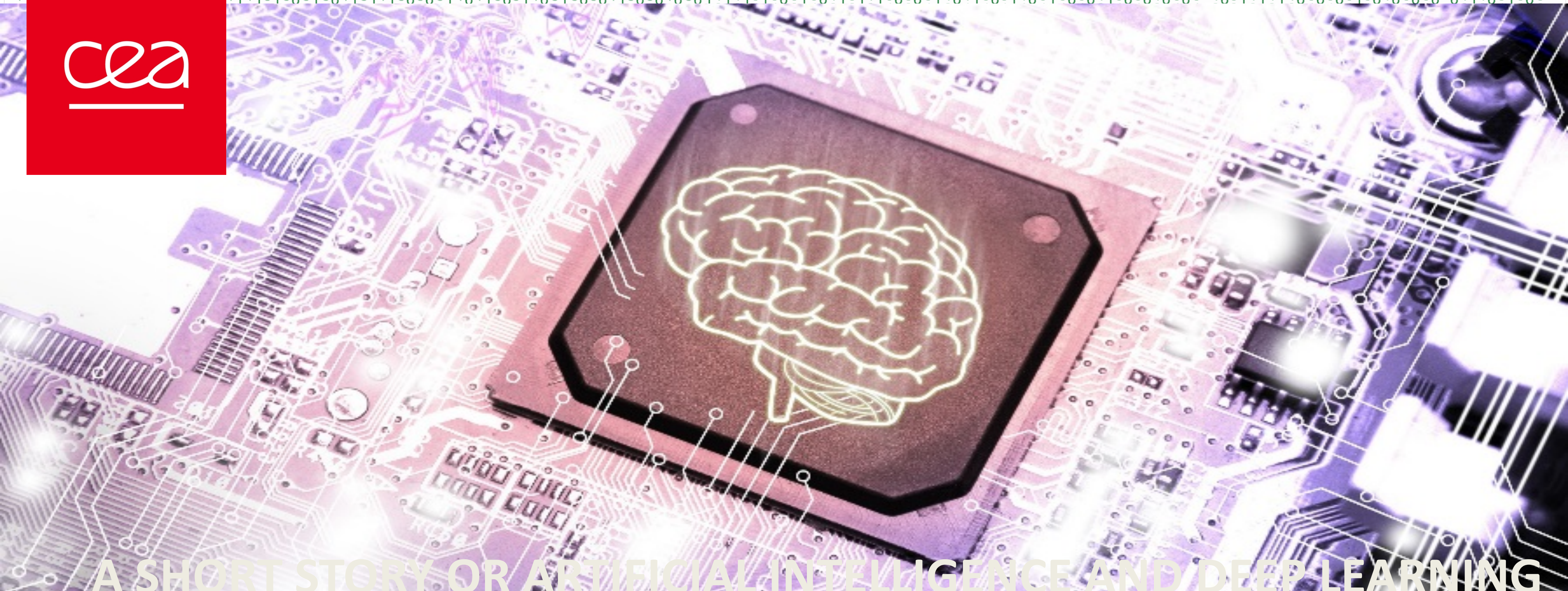
A SHORT STORY OR ARTIFICIAL INTELLIGENCE AND DEEP LEARNING

Marc Duranton

Commissariat à l’énergie atomique et aux énergies alternatives

Move to:

Part 2: from 2017: the era of generative AI



A SHORT STORY OR ARTIFICIAL INTELLIGENCE AND DEEP LEARNING

Part 2: from 2017: the era of generative AI

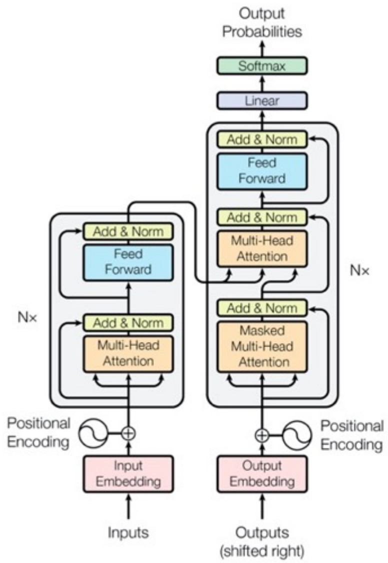
Marc Duranton

Commissariat à l’énergie atomique et aux énergies alternatives

June 4th, 2025

The origin of LLMs: Transformers (2017)

We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

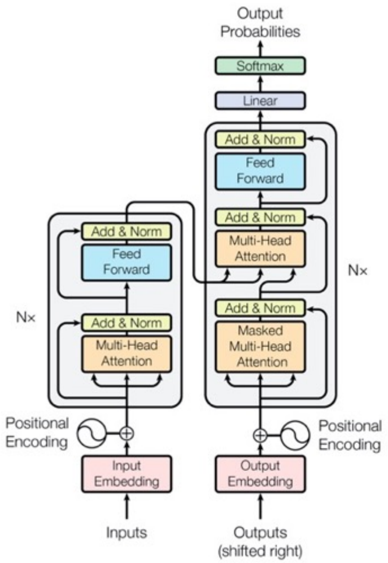
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

The origin of LLMs: Transformers (2017)

We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

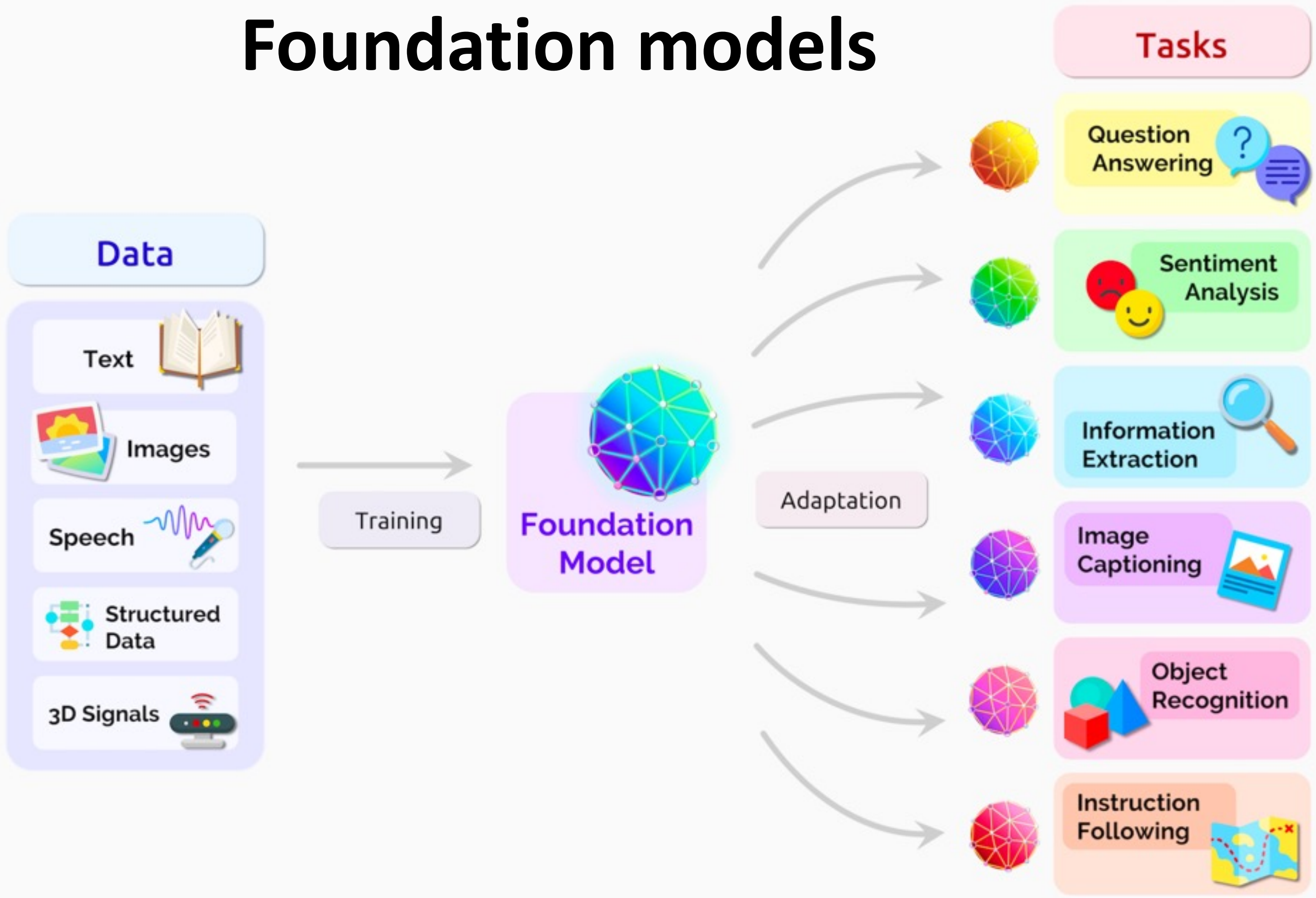
Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

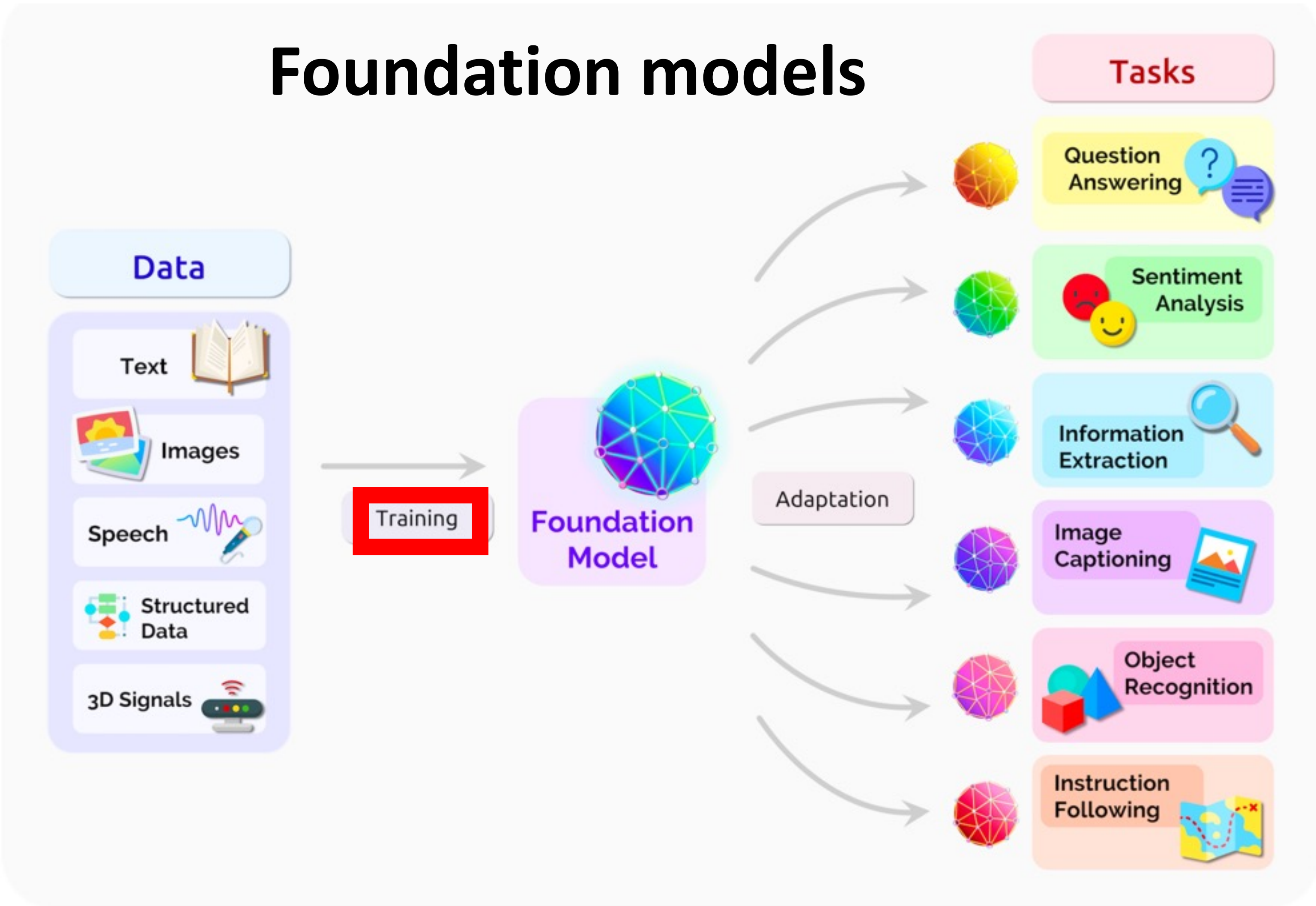
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Foundation models



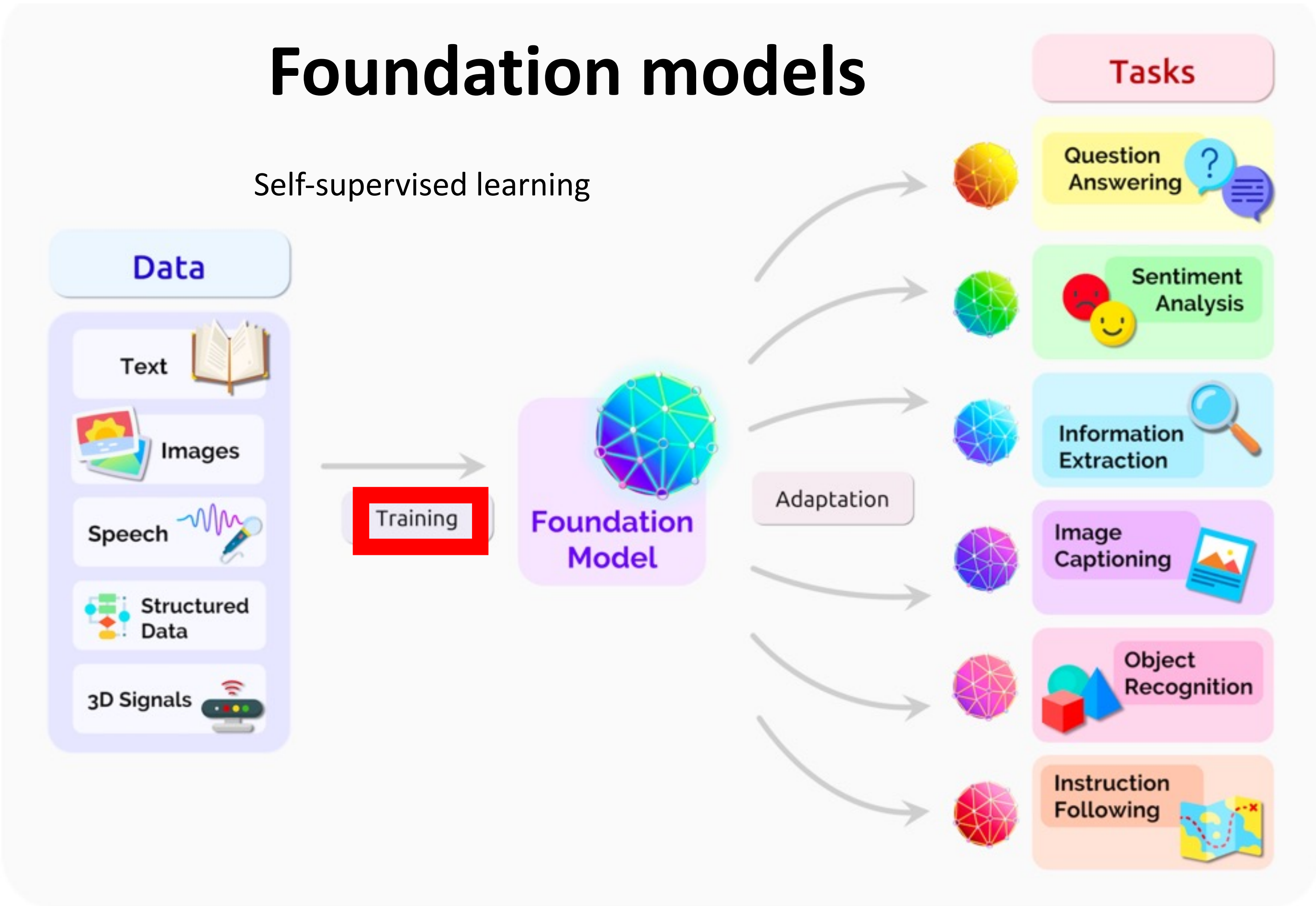
« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »

From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258



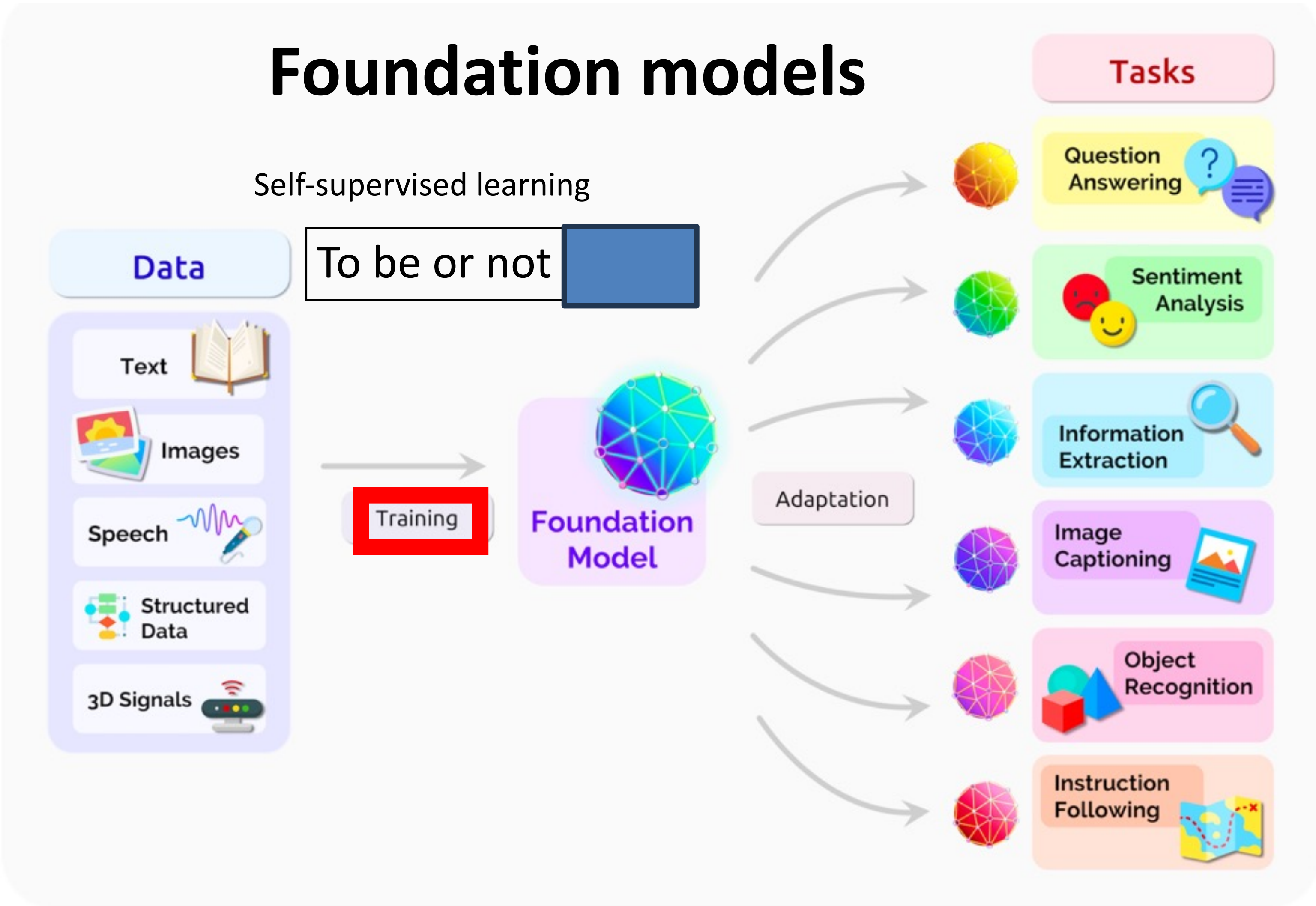
« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »
From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258

Foundation models



« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »
From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258

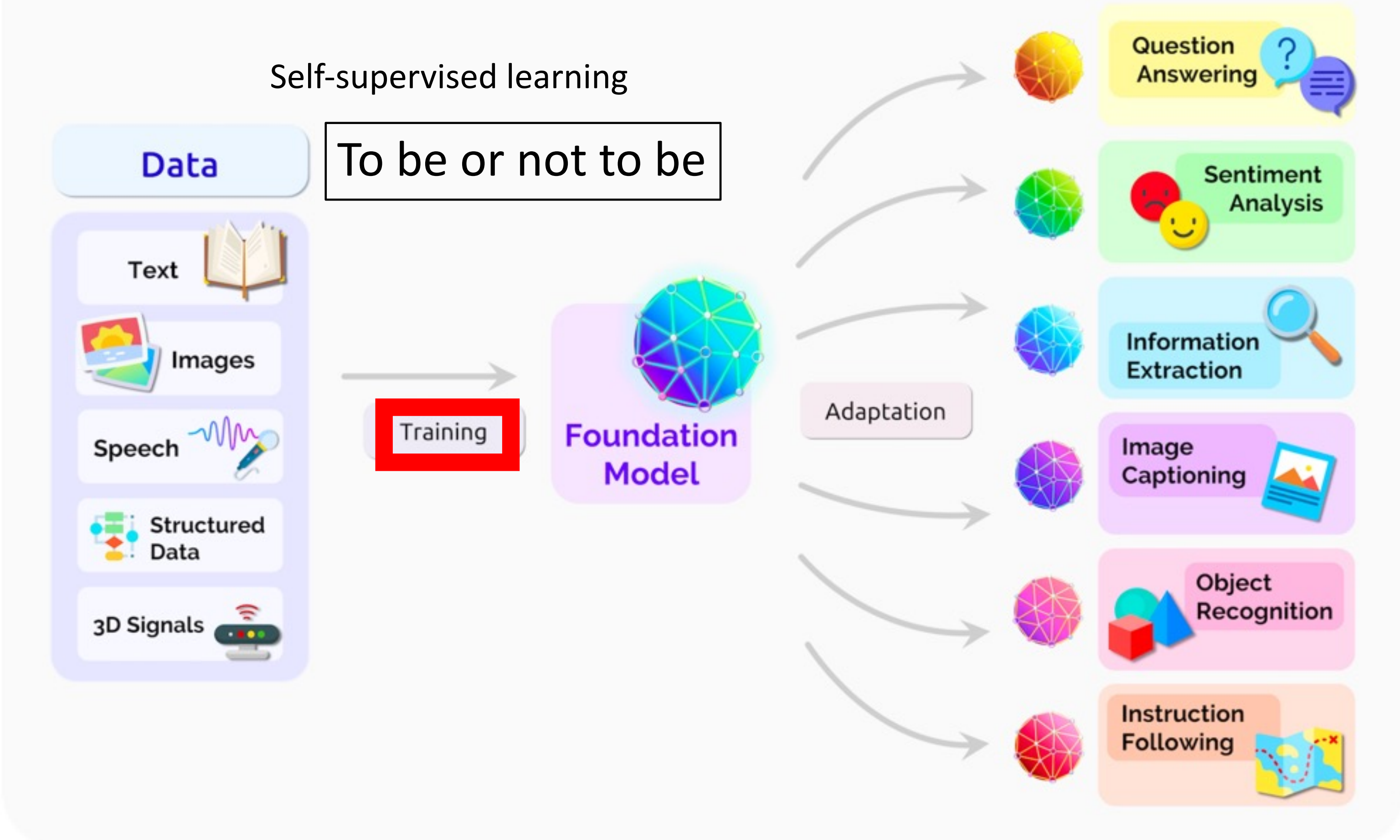
Foundation models



« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »

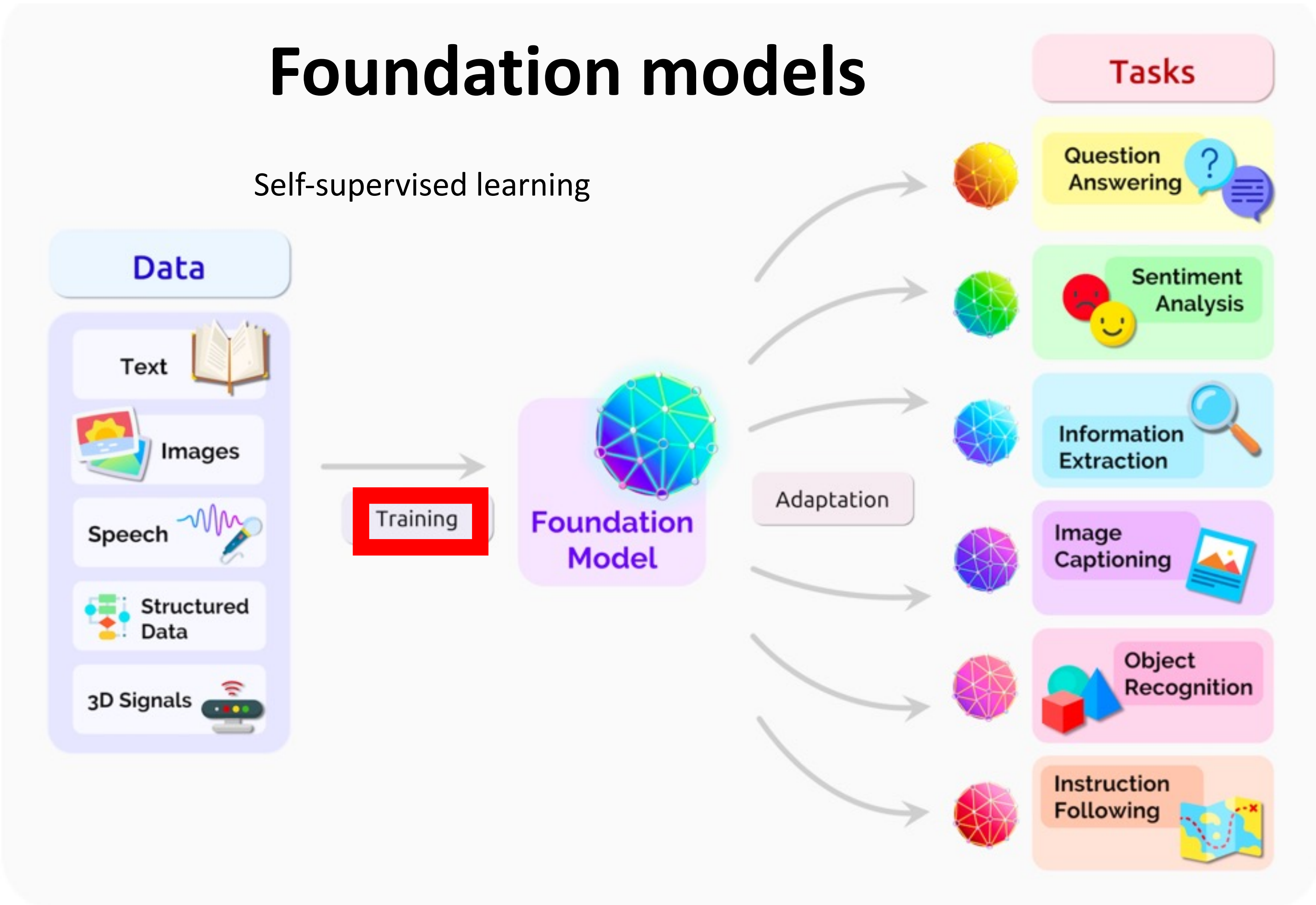
From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258

Foundation models



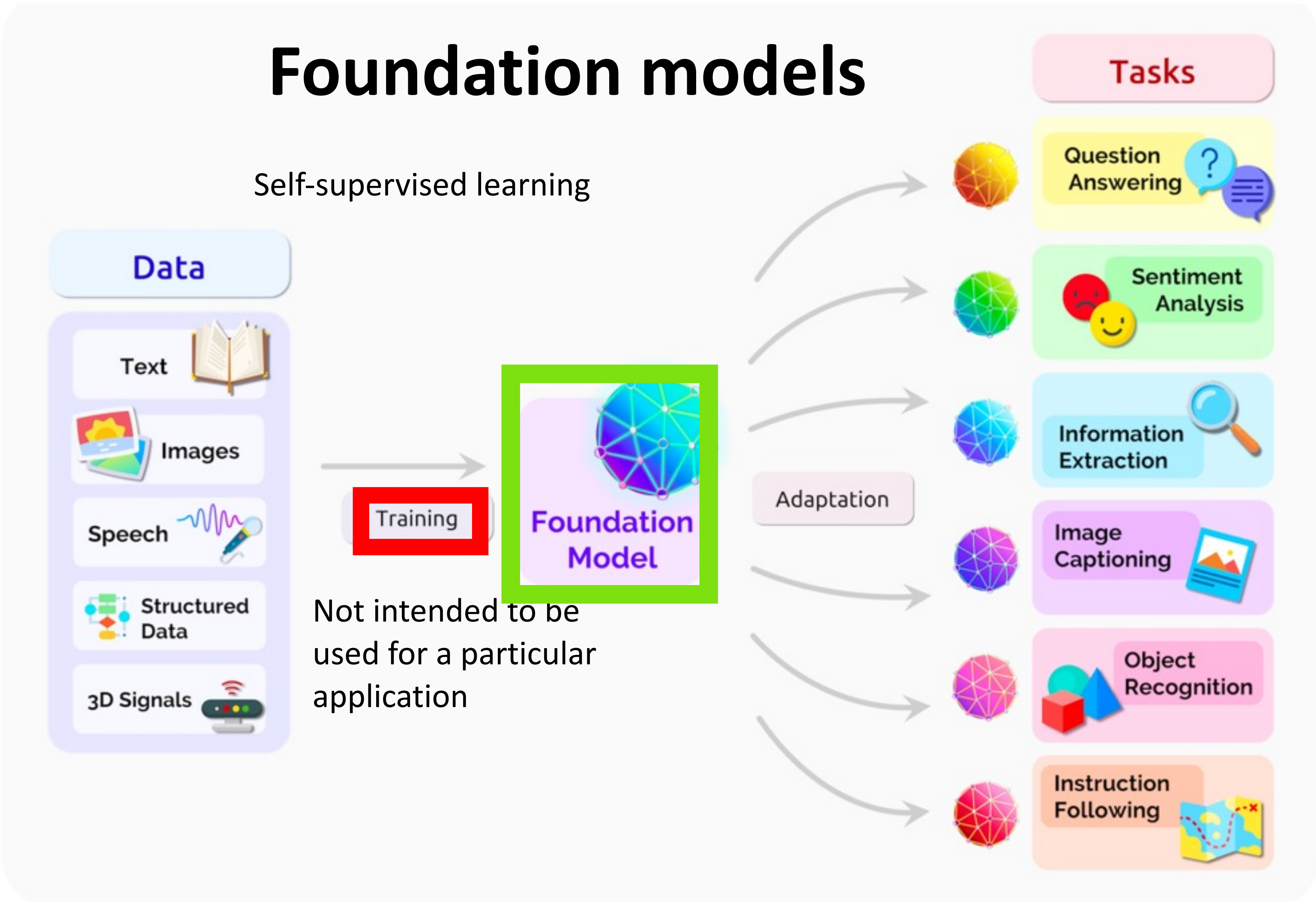
« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »
From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258

Foundation models



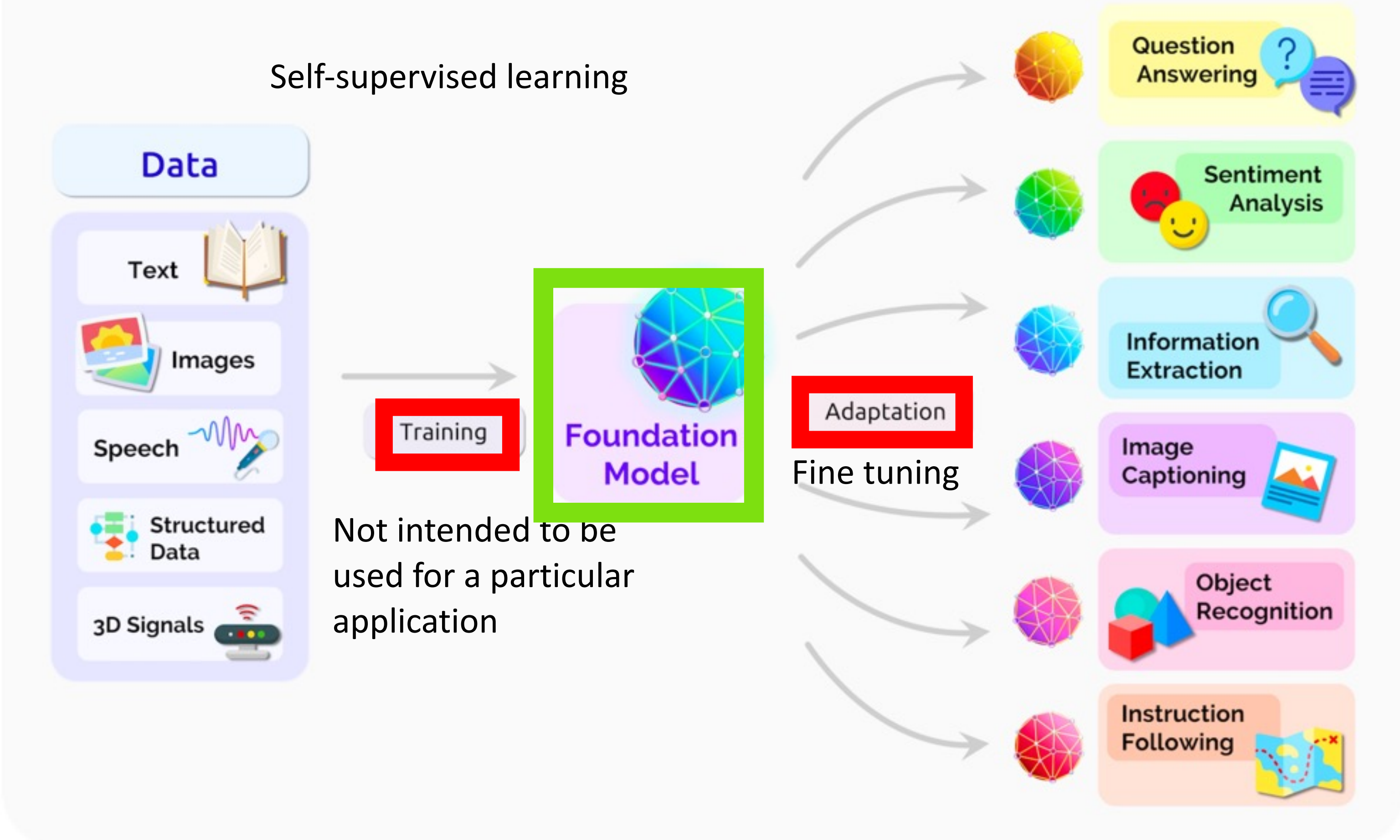
« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »
From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258

Foundation models

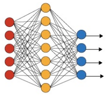


« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »
From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258

Foundation models



« A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks. »
From « On the Opportunities and Risks of Foundation Models » https://arxiv.org/abs/2108.07258



Evolution of Generative Pre-trained Transformers (GPT) in OpenAI

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus: 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, **2018**	"1 month on 8 GPUs", or 1.7e19 FLOP.
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, **2019** (initial/limited version) and November 5, 2019 (full version)	"tens of petaflop/s-day", or 1.5e21 FLOP.
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion	499 Billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, **2020**	3640 petaflop/s-day, or 3.2e23 FLOP.
GPT-3.5	Undisclosed	175 billion	Undisclosed	March 15, **2022**	Undisclosed
ChatGPT	Undisclosed	? (rumor 20M???)		**November 20, 2022**	
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.	Undisclosed (1.8 trillon aka 1.8e12)	Undisclosed (13 trillon tokens, aka 1.3e13)	March 14, **2023**	Undisclosed. Estimated 2.1e25 FLOP.

From https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

Evolution of Generative Pre-trained Transformers (GPT) in OpenAI

Compute requirement

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus: 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, **2018**	"1 month on 8 GPUs", or 1.7e19 FLOP.
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, **2019** (initial/limited version) and November 5, 2019 (full version)	"tens of petaflop/s-day", or 1.5e21 FLOP.
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion	499 Billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, **2020**	3640 petaflop/s-day, or 3.2e23 FLOP.
GPT-3.5	Undisclosed	175 billion	Undisclosed	March 15, **2022**	Undisclosed
ChatGPT	Undisclosed	? (rumor 20M???)		**November 20, 2022**	
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.	Undisclosed (1.8 trillon aka 1.8e12)	Undisclosed (13 trillon tokens, aka 1.3e13)	March 14, **2023**	Undisclosed. Estimated 2.1e25 FLOP.

From https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

Evolution of Generative Pre-trained Transformers (GPT) in OpenAI

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus: 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, **2018**	"1 month on 8 GPUs", or 1.7e19 FLOP.
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, **2019** (initial/limited version) and November 5, 2019 (full version)	"tens of petaflop/s-day", or 1.5e21 FLOP.
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion	499 Billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, **2020**	3640 petaflop/s-day, or 3.2e23 FLOP.
GPT-3.5	Undisclosed	175 billion	Undisclosed	March 15, **2022**	Undisclosed
ChatGPT	Undisclosed	? (rumor 20M???)		**November 20, 2022**	
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.	Undisclosed (1.8 trillon aka 1.8e12)	Undisclosed (13 trillon tokens, aka 1.3e13)	March 14, **2023**	Undisclosed. Estimated 2.1e25 FLOP.

From https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

Compute requirement

~ x 88

Evolution of Generative Pre-trained Transformers (GPT) in OpenAI

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus: 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, **2018**	"1 month on 8 GPUs", or 1.7e19 FLOP.
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, **2019** (initial/limited version) and November 5, 2019 (full version)	"tens of petaflop/s-day", or 1.5e21 FLOP.
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion	499 Billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, **2020**	3640 petaflop/s-day, or 3.2e23 FLOP.
GPT-3.5	Undisclosed	175 billion	Undisclosed	March 15, **2022**	Undisclosed
ChatGPT	Undisclosed	? (rumor 20M???)		**November 20, 2022**	
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.	Undisclosed (1.8 trillon aka 1.8e12)	Undisclosed (13 trillon tokens, aka 1.3e13)	March 14, **2023**	Undisclosed. Estimated 2.1e25 FLOP.

From https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

Compute requirement

~ x 88

~ x 213

Evolution of Generative Pre-trained Transformers (GPT) in OpenAI

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus: 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, **2018**	"1 month on 8 GPUs", or 1.7e19 FLOP.
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, **2019** (initial/limited version) and November 5, 2019 (full version)	"tens of petaflop/s-day", or 1.5e21 FLOP.
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion	499 Billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, **2020**	3640 petaflop/s-day, or 3.2e23 FLOP.
GPT-3.5	Undisclosed	175 billion	Undisclosed	March 15, **2022**	Undisclosed
ChatGPT	Undisclosed	? (rumor 20M???)		**November 20, 2022**	
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.	Undisclosed (1.8 trillon aka 1.8e12)	Undisclosed (13 trillon tokens, aka 1.3e13)	March 14, **2023**	Undisclosed. Estimated 2.1e25 FLOP.

From https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

Compute requirement

~ x 88

~ x 213

~ x 65

Evolution of Generative Pre-trained Transformers (GPT) in OpenAI

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus: 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, **2018**	"1 month on 8 GPUs", or 1.7e19 FLOP.
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, **2019** (initial/limited version) and November 5, 2019 (full version)	"tens of petaflop/s-day", or 1.5e21 FLOP.
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion	499 Billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, **2020**	3640 petaflop/s-day, or 3.2e23 FLOP.
GPT-3.5	Undisclosed	175 billion	Undisclosed	March 15, **2022**	Undisclosed
ChatGPT	Undisclosed	? (rumor 20M???)		**November 20, 2022**	
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.	Undisclosed (1.8 trillon aka 1.8e12)	Undisclosed (13 trillon tokens, aka 1.3e13)	March 14, **2023**	Undisclosed. Estimated 2.1e25 FLOP.

From https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

Compute requirement

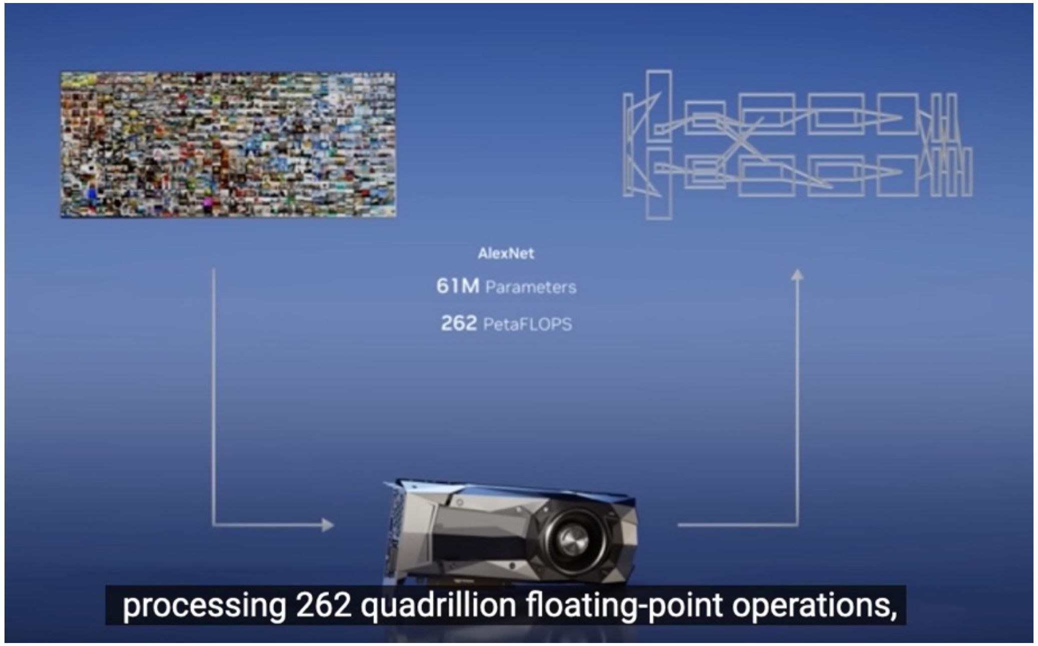
~ x 88

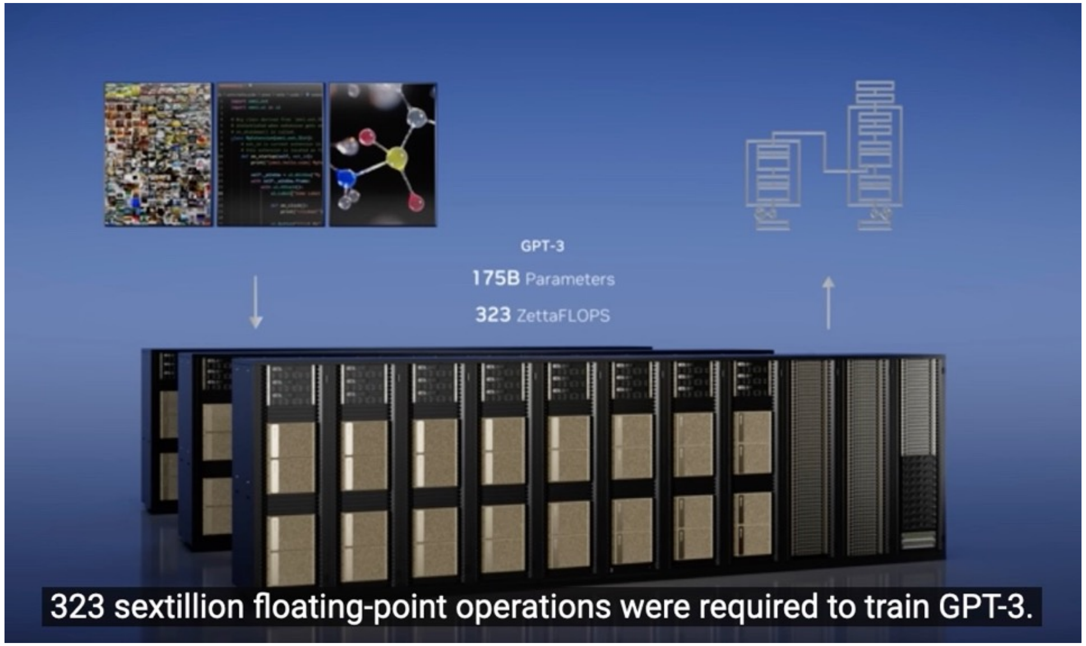
~ x 213

~ x 65

~ x 1 218 360

Computing power is driving the advance of AI

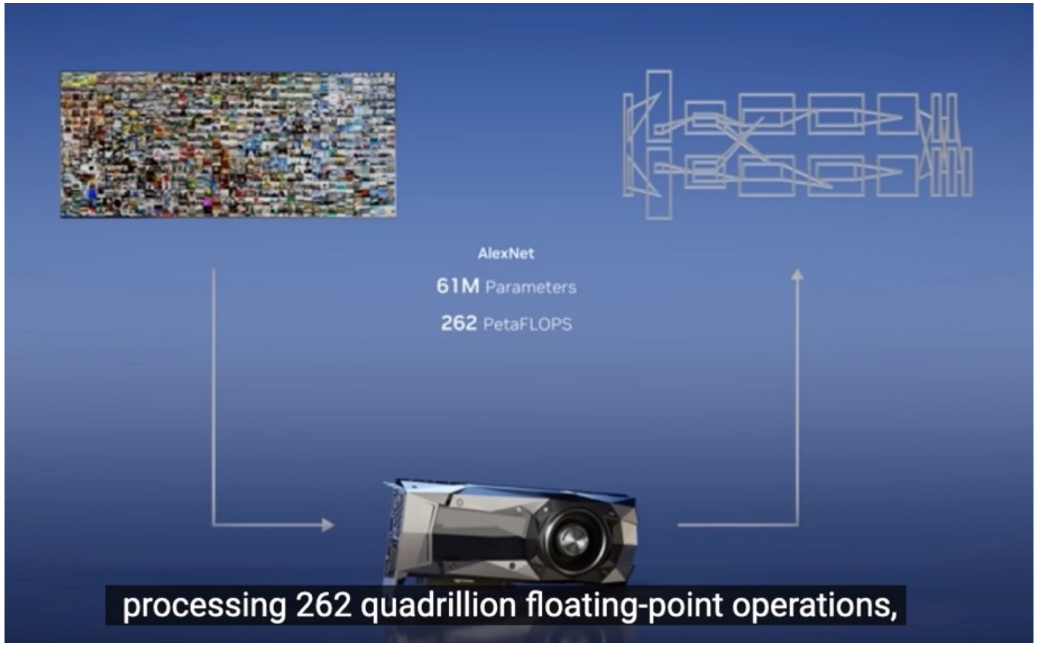




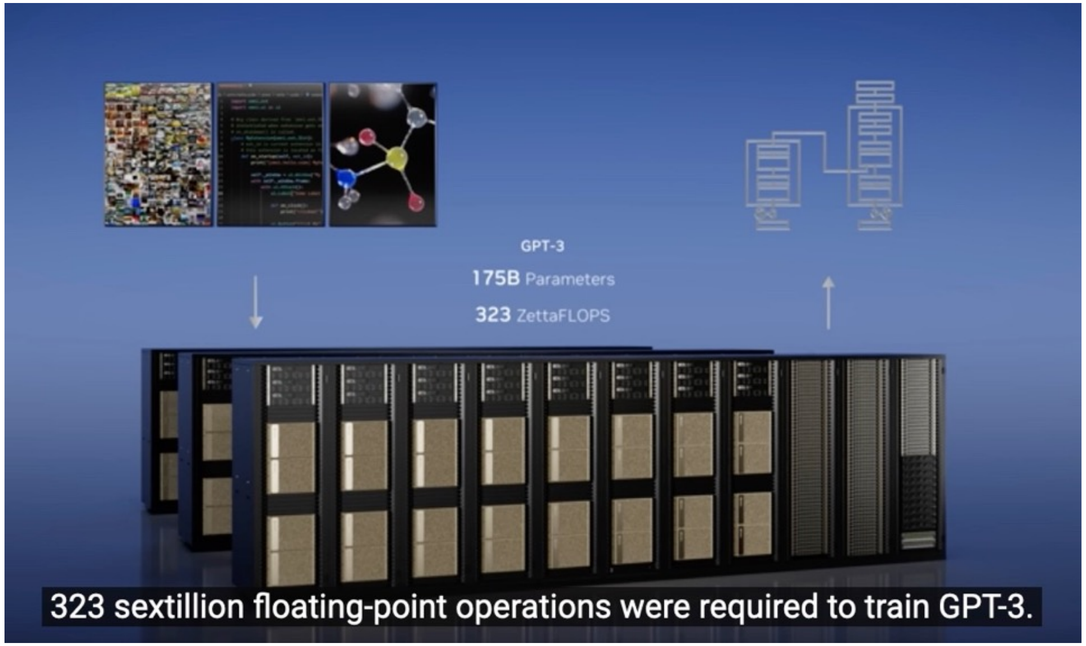
2012: AlexNet
GeForce GTX 580
Won ImageNet Challenge
262 x 10^{15} FLOPS (262 PetaFLOPS)

From GTC 2023 Keynote with NVIDIA CEO Jensen Huang

Computing power is driving the advance of AI



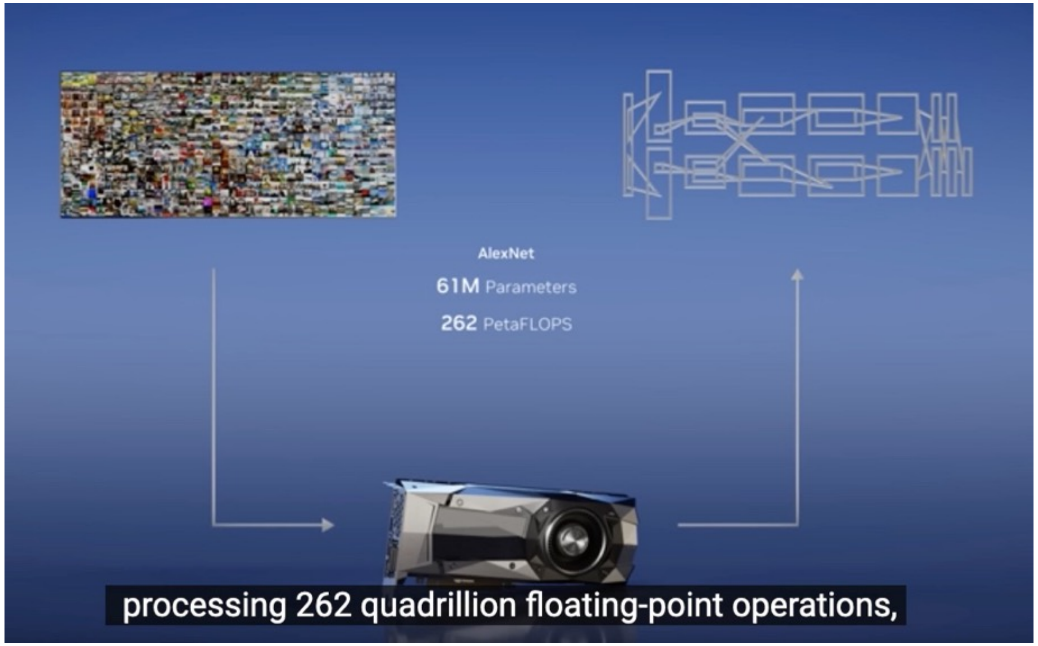
2012: AlexNet
GeForce GTX 580
Won ImageNet Challenge
262 x 10^{15} FLOPS (262 PetaFLOPS)



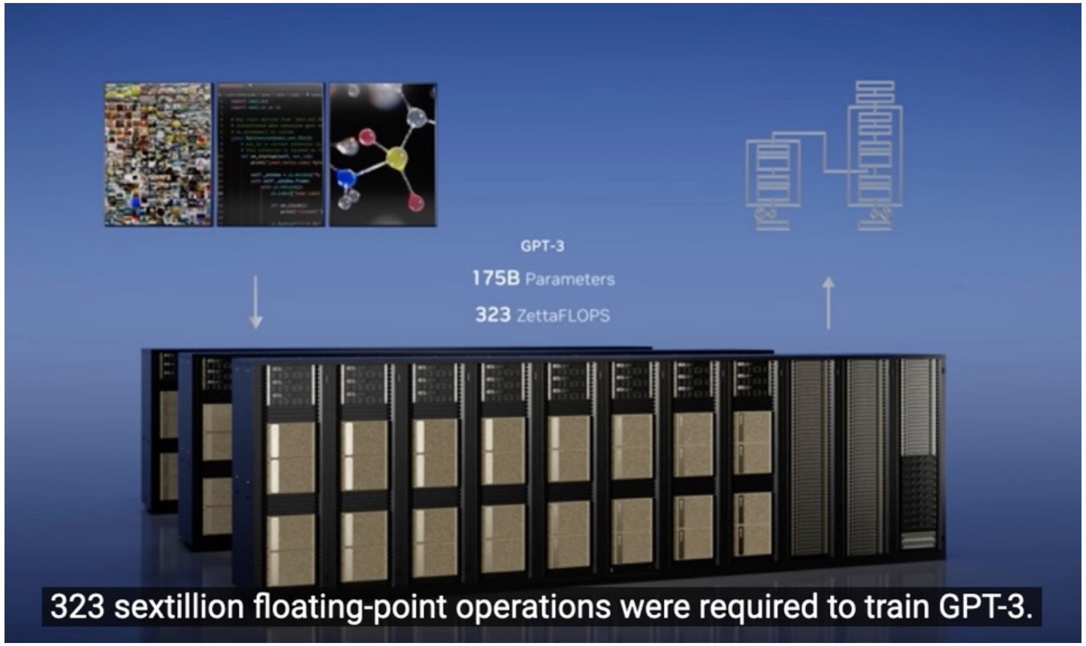
2020: GPT-3
323 x 10^{21} FLOPS (323 ZerraFLOPS)
X 1 000 000 more floating point operations

From GTC 2023 Keynote with NVIDIA CEO Jensen Huang

Computing power is driving the advance of AI



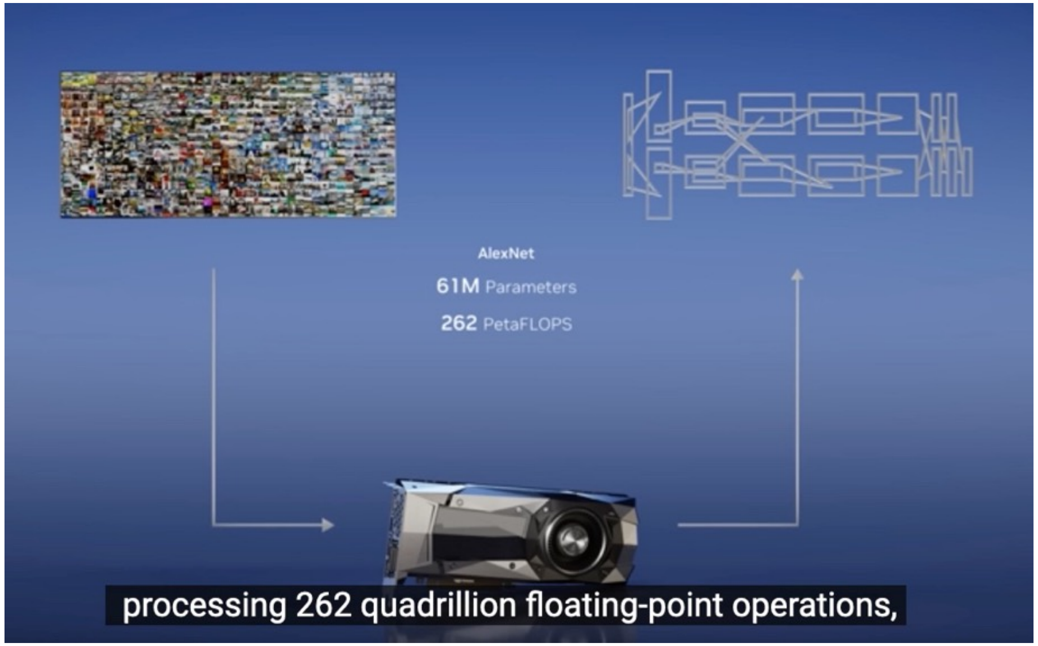
2012: AlexNet
GeForce GTX 580
Won ImageNet Challenge
262 x 10^{15} FLOPS (262 PetaFLOPS)



2020: GPT-3
323 x 10^{21} FLOPS (323 ZerraFLOPS)
X 1 000 000 more floating point operations

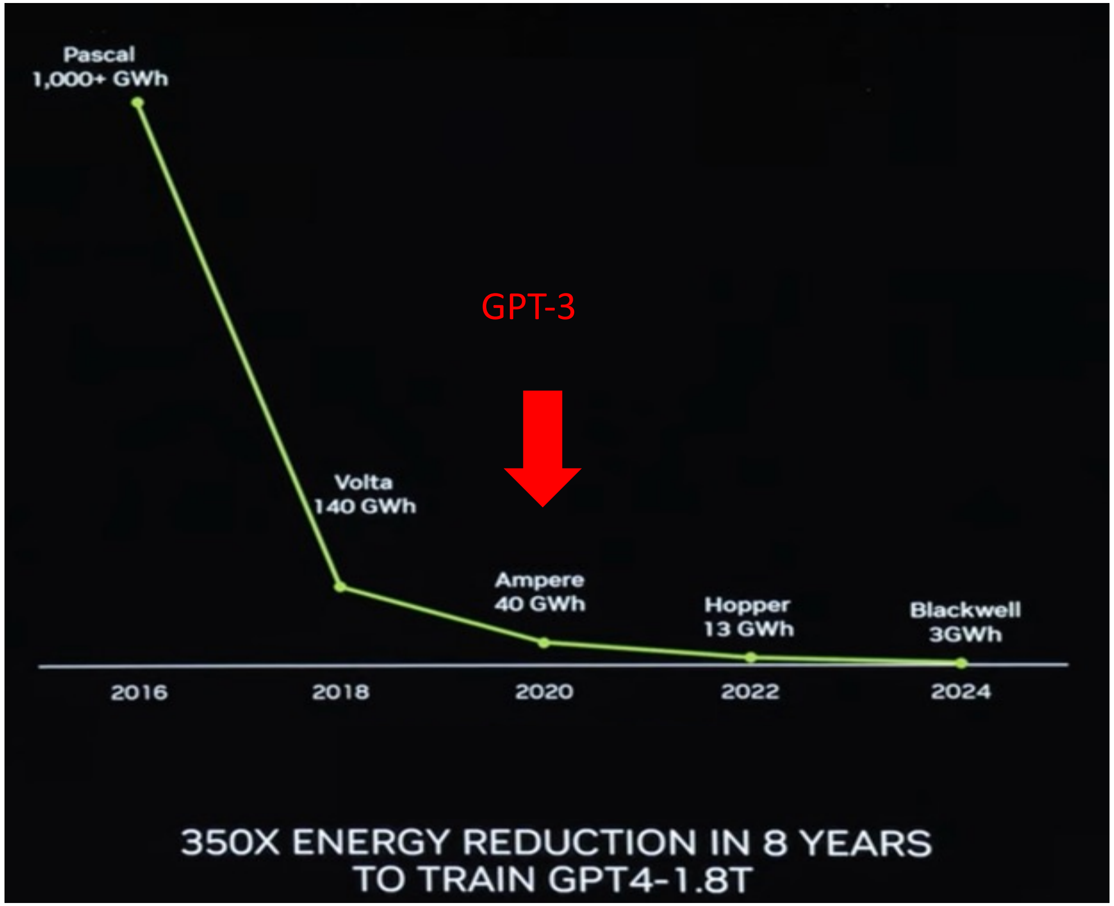
From GTC 2023 Keynote with NVIDIA CEO Jensen Huang

Computing power is driving the advance of AI

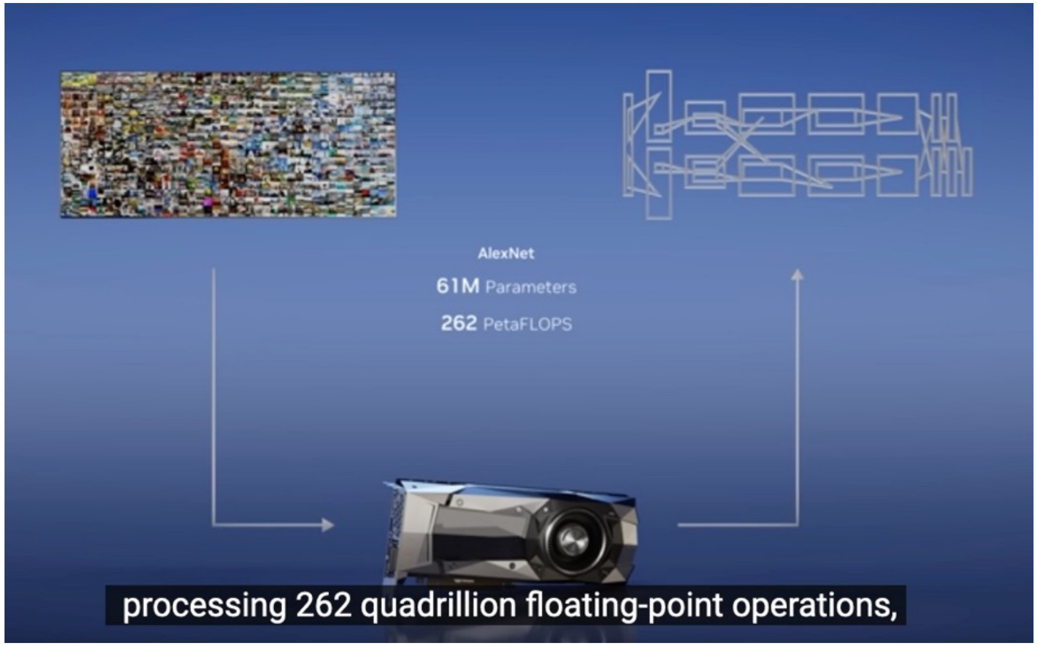


2012: AlexNet
GeForce GTX 580
Won ImageNet Challenge
262 x 10^{15} FLOPS (262 PetaFLOPS)

From GTC 2023 Keynote with NVIDIA CEO Jensen Huang

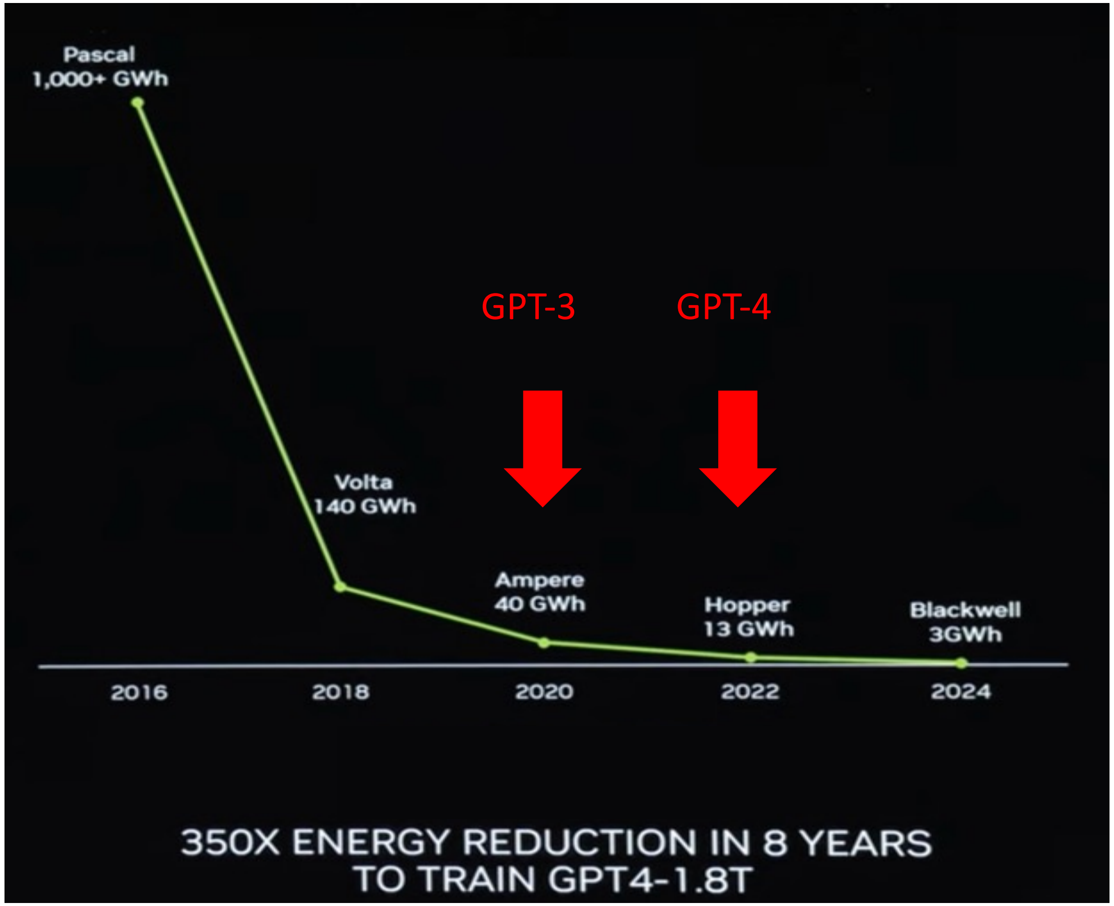


Computing power is driving the advance of AI

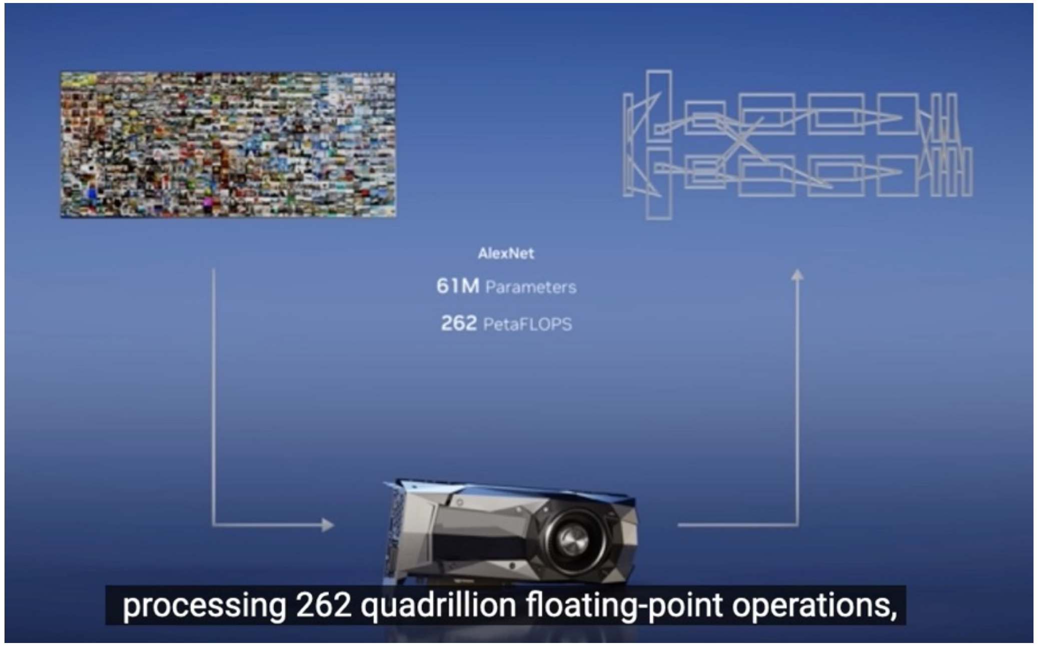


2012: AlexNet
GeForce GTX 580
Won ImageNet Challenge
262 x 10^{15} FLOPS (262 PetaFLOPS)

From GTC 2023 Keynote with NVIDIA CEO Jensen Huang

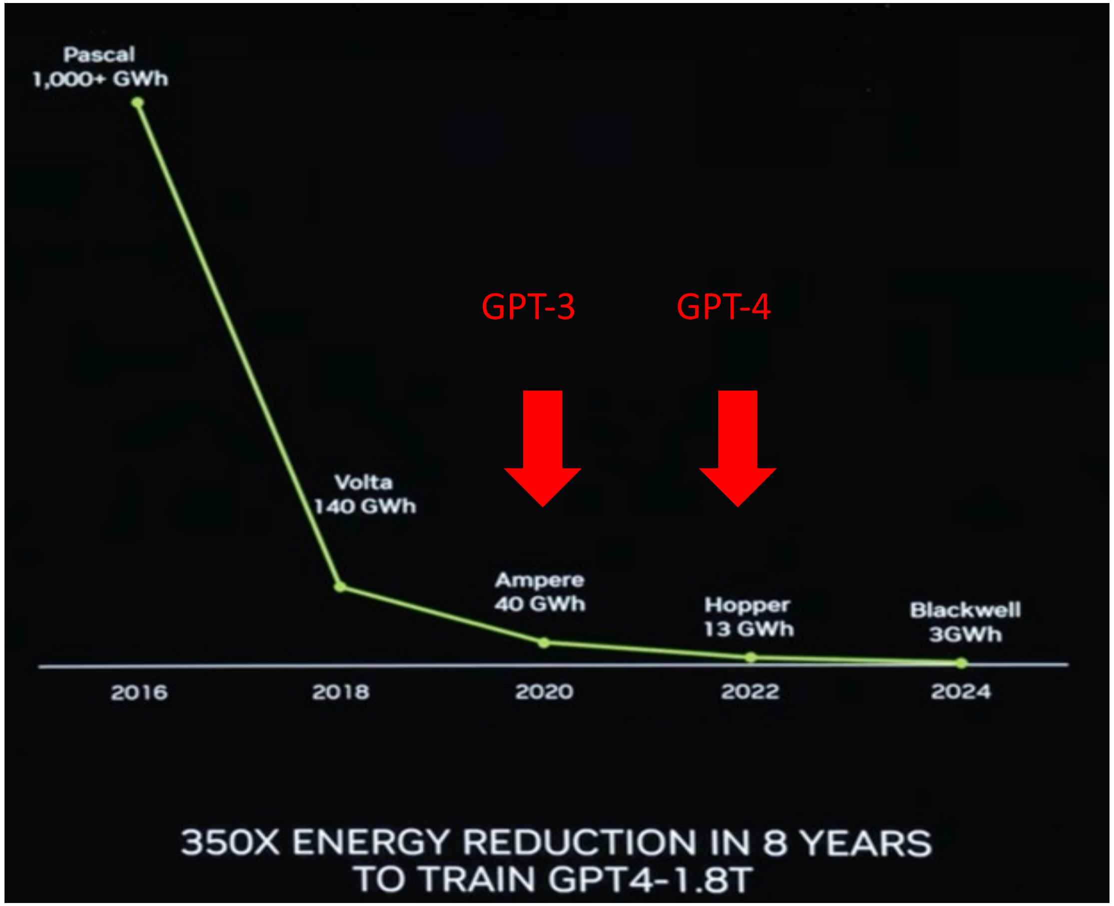


Computing power is driving the advance of AI



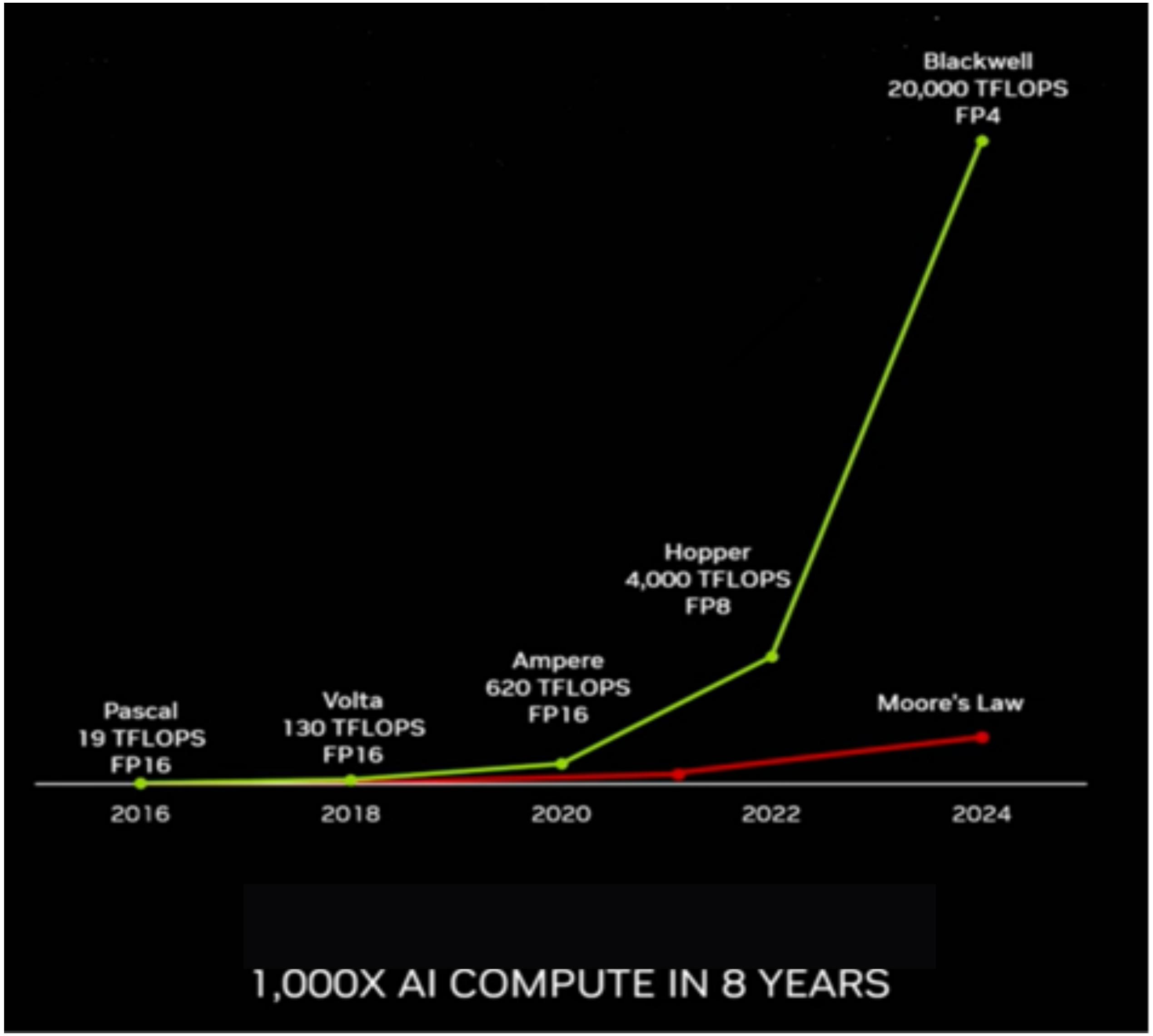
2012: AlexNet
GeForce GTX 580
Won ImageNet Challenge
262 x 10^{15} FLOPS (262 PetaFLOPS)

From GTC 2023 Keynote with NVIDIA CEO Jensen Huang



Cost of energy for training is a limiting factor!

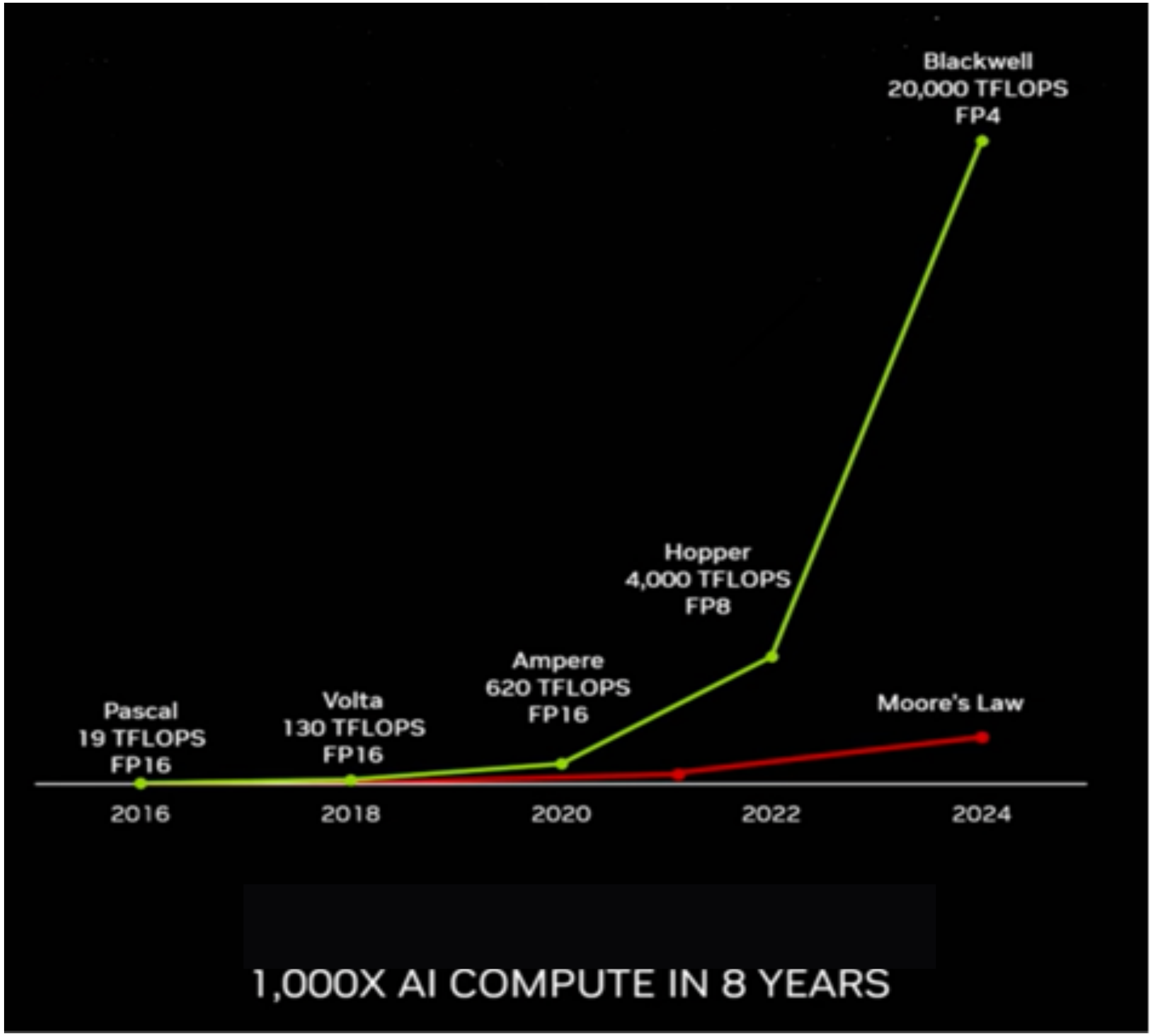
Exponential increase of AI performances



From Nvidia, Computex 2024

Exponential increase of AI performances

Thanks to advances in *architecture* and data coding (moving from float 64/32 to FP4) (but it is a one shot!)



From Nvidia, Computex 2024

One of the early Open Source LLM (March-July 2022)

BigScience

BLOOM: open-source alternative to GPT-3

a BigScience initiative

BLOOM

176B params · 59 languages · Open-access

https://bigscience.huggingface.co

https://huggingface.co/bigscience/bloom

1.5TB of text, 350B tokens

43 languages, 16 programming languages

118 days of training on 384 A100 GPUs

More details at https://huggingface.co/blog/bloom-megatron-deepspeed

Smaller versions are available : 560M, 1.1B, 1.7B, 3B, 7.1B

BLOOMZ models (same sizes) are fine-tuned for **instruction following**
https://huggingface.co/bigscience/bloomz

One of the early Open Source LLM (March-July 2022)

BigScience

a BigScience initiative

BLOOM

176B params · 59 languages · Open-access

BLOOM: open-source alternative to GPT-3

https://bigscience.huggingface.co

https://huggingface.co/bigscience/bloom

1.5TB of text, 350B tokens

43 languages, 16 programming languages

118 days of training on 384 A100 GPUs

Estimated cost of training: Equivalent of $2-5M in cloud
Server training location: Île-de-France, France
Environmental Impact: The training supercomputer, Jean Zay, **uses mostly nuclear energy**. The **heat generated by it is reused** for heating campus housing.

More details at https://huggingface.co/blog/bloom-megatron-deepspeed

Smaller versions are available : 560M, 1.1B, 1.7B, 3B, 7.1B

BLOOMZ models (same sizes) are fine-tuned for **instruction following**
https://huggingface.co/bigscience/bloomz

One of the early Open Source LLM (March-July 2022)

BigScience

a BigScience initiative

BLOOM

176B params · 59 languages · Open-access

BLOOM: open-source alternative to GPT-3

https://bigscience.huggingface.co

https://huggingface.co/bigscience/bloom

1.5TB of text, 350B tokens

43 languages, 16 programming languages

118 days of training on 384 A100 GPUs

Estimated cost of training: Equivalent of $2-5M in cloud
Server training location: Île-de-France, France
Environmental Impact: The training supercomputer, Jean Zay, **uses mostly nuclear energy**. The **heat generated by it is reused** for heating campus housing.

More details at https://huggingface.co/blog/bloom-megatron-deepspeed

Smaller versions are available : 560M, 1.1B, 1.7B, 3B, 7.1B

BLOOMZ models (same sizes) are fine-tuned for **instruction following**
https://huggingface.co/bigscience/bloomz

2022: Reinforcement with simulation in the loop

The reinforcement technique with simulation in the loop allow to learn and adapt with minimum numbers of real data (from S. Abeyruwan et al., “i-Sim2Real: Reinforcement Learning of Robotic Policies in Tight Human-Robot Interaction Loops (pre-print), Arxiv, 22 November 2022. Available: https://arxiv.org/abs/2207.06572.

2022: Reinforcement with simulation in the loop

The reinforcement technique with simulation in the loop allow to learn and adapt with minimum numbers of real data (from S. Abeyruwan et al., “i-Sim2Real: Reinforcement Learning of Robotic Policies in Tight Human-Robot Interaction Loops (pre-print), Arxiv, 22 November 2022. Available: https://arxiv.org/abs/2207.06572.

2022: Reinforcement with simulation in the loop

The reinforcement technique with simulation in the loop allow to learn and adapt with minimum numbers of real data (from S. Abeyruwan et al., “i-Sim2Real: Reinforcement Learning of Robotic Policies in Tight Human-Robot Interaction Loops (pre-print), Arxiv, 22 November 2022. Available: https://arxiv.org/abs/2207.06572.

AI for making new structures: “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

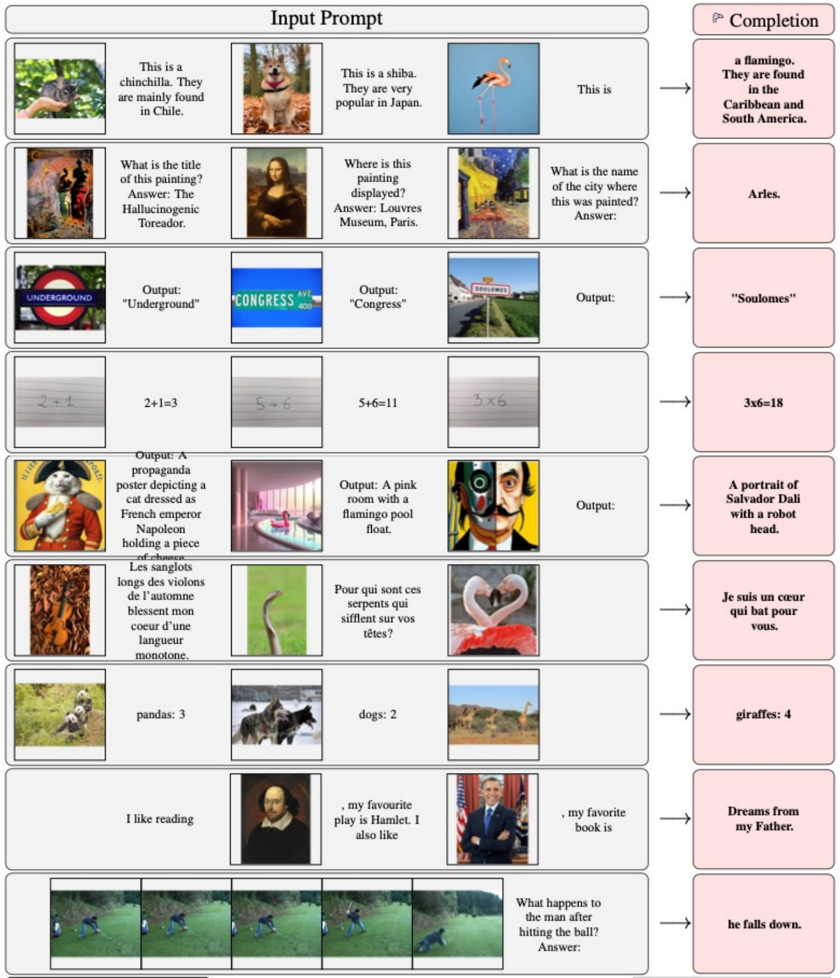
2022: Flamingo (Deepmind) : image + text + few shot learning

Flamingo: a Visual Language Model for Few-Shot Learning

Jean-Baptiste Alayrac[*,‡] Jeff Donahue[*] Pauline Luc[*] Antoine Miech[*]
Iain Barr[†] Yana Hasson[†] Karel Lenc[†] Arthur Mensch[†] Katie Millican[†]
Malcolm Reynolds[†] Roman Ring[†] Eliza Rutherford[†] Serkan Cabi Tengda Han
Zhitao Gong Sina Samangooei Marianne Monteiro Jacob Menick
Sebastian Borgeaud Andrew Brock Aida Nematzadeh Sahand Sharifzadeh
Mikolaj Binkowski Ricardo Barreira Oriol Vinyals Andrew Zisserman
Karen Simonyan[*,‡]

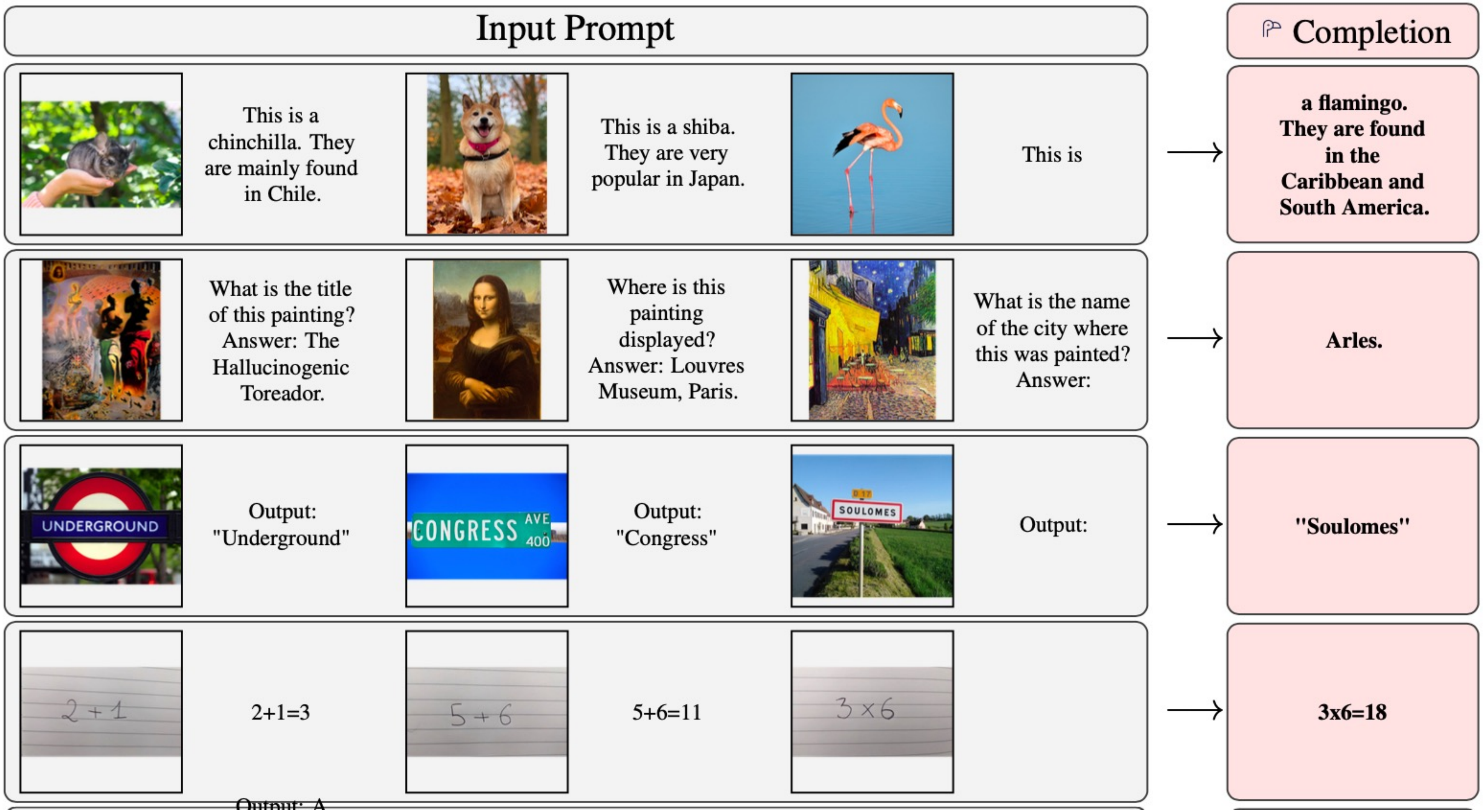
[*] Equal contributions, ordered alphabetically, [†] Equal contributions, ordered alphabetically, [‡] Equal senior contributions

DeepMind



From https://arxiv.org/abs/2204.14198

2022: Flamingo (Deepmind) : image + text + few shot learning



From https://arxiv.org/abs/2204.14198

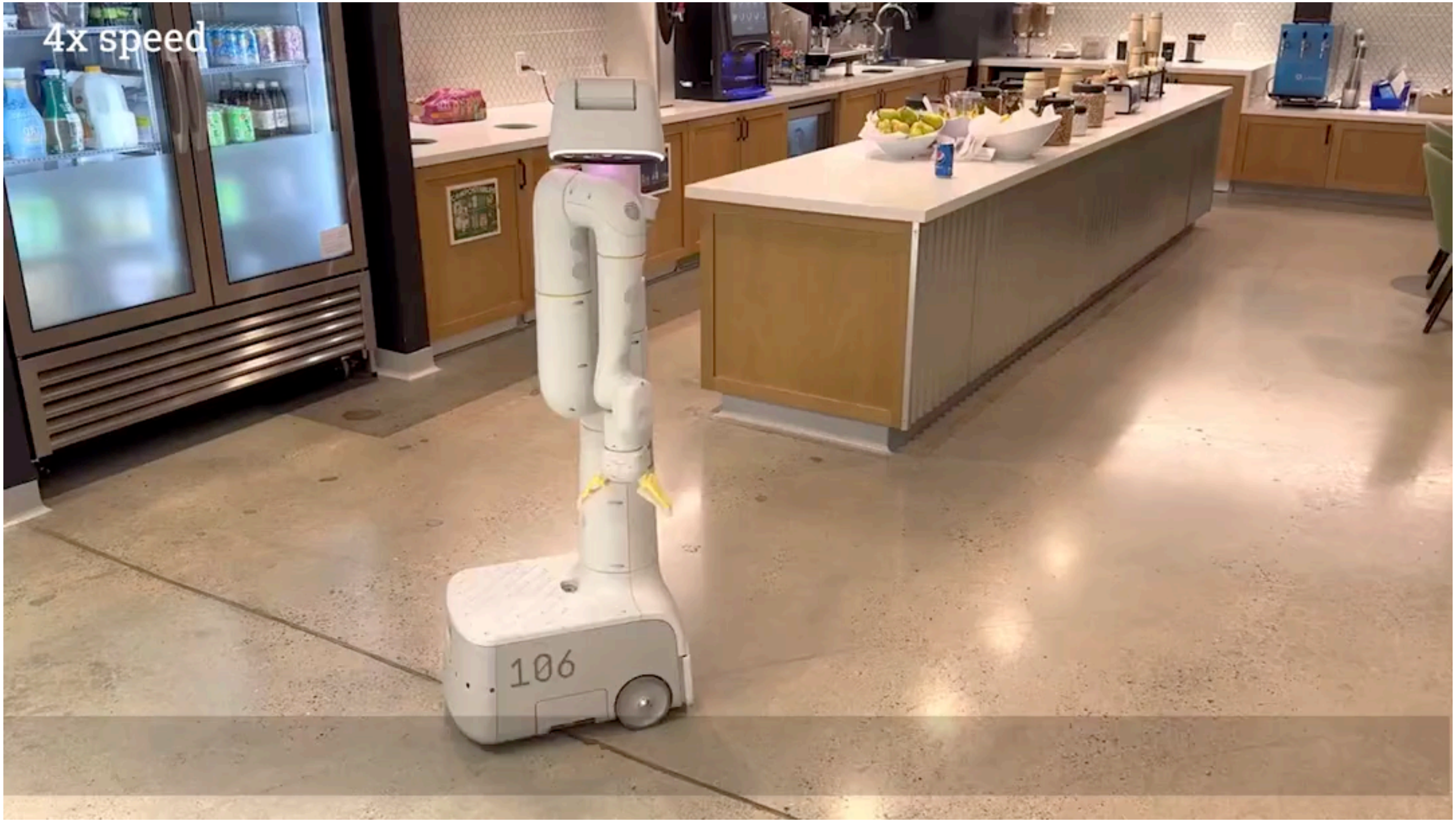
2023: PaLM-E: An Embodied Multimodal Language Model

Robot: I am a robot operating in a kitchen. Given **img**, When a human asks me to do a task, I will respond with the sequence of actions I would do to accomplish the task with only the items I see. Human: Use all of the ingredients you see to make a cake batter.

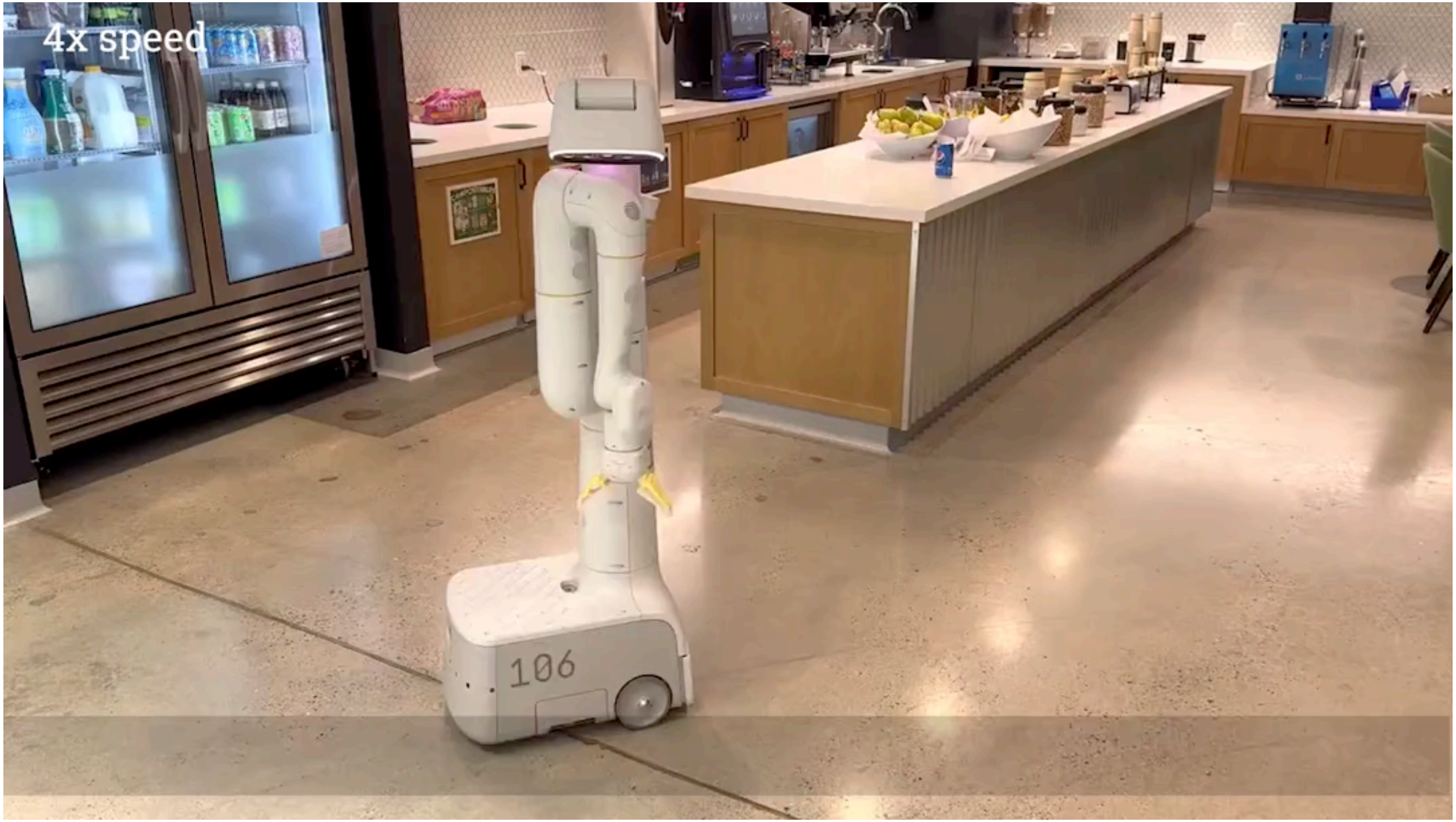
1. crack egg. 2. put egg in bowl. 3. put flour in bowl. 4. put sugar in bowl. 5. mix. 6. put in pan. 7. bake. 8. eat. 9. clean up.

From https://palm-e.github.io/

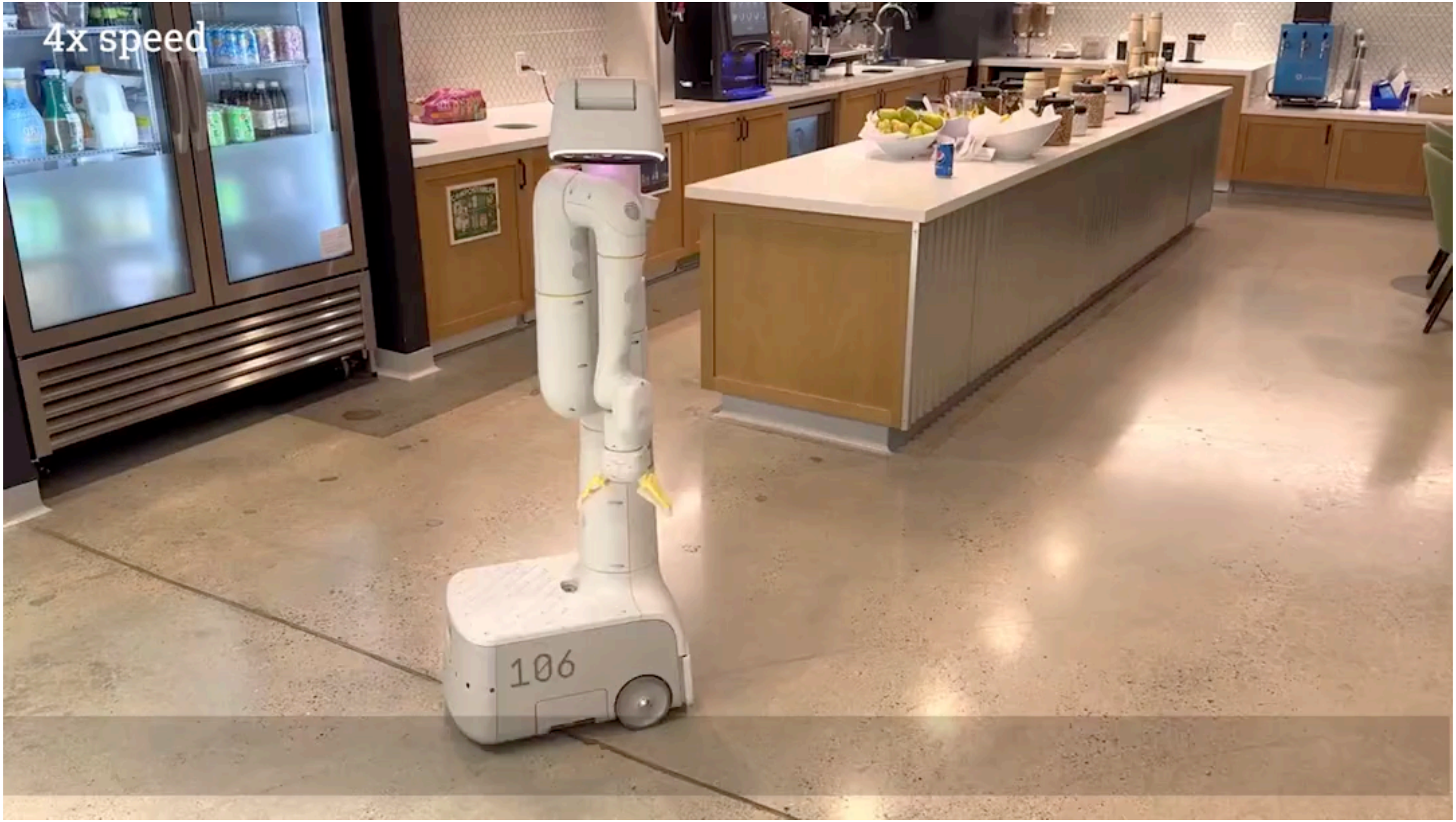
2023: PaLM-E: An Embodied Multimodal Language Model



2023: PaLM-E: An Embodied Multimodal Language Model



2023: PaLM-E: An Embodied Multimodal Language Model



2023: The trigger of “open weights” revolution: Alpaca from Stanford, derived from LLaMa from Meta

Alpaca: A Strong, Replicable Instruction-Following Model

Authors: Rohan Taori* and Ishaan Gulrajani* and Tianyi Zhang* and Yann Dubois* and Xuechen Li* and Carlos Guestrin and Percy Liang and Tatsunori B. Hashimoto

---

*We introduce **Alpaca 7B**, a model fine-tuned from the LLaMA 7B model on 52K instruction-following demonstrations. On our preliminary evaluation of single-turn instruction following, Alpaca behaves qualitatively similarly to OpenAI's text-davinci-003, while being surprisingly small and easy/cheap to reproduce (<600$). Checkout our code release on GitHub.*

Update: The public demo is now disabled. The original goal of releasing a demo was to disseminate our research in an accessible way. We feel that we have mostly achieved this goal, and given the hosting costs and the inadequacies of our content filters, we decided to bring down the demo.

Stanford Alpaca

From https://crfm.stanford.edu/2023/03/13/alpaca.html

2023: The trigger of “open weights” revolution: Alpaca from Stanford, derived from LLaMa from Meta

Alpaca: A Strong, Replicable Instruction-Following Model

Authors: Rohan Taori* and Ishaan Gulrajani* and Tianyi Zhang* and Yann Dubois* and Xuechen Li* and Carlos Guestrin and Percy Liang and Tatsunori B. Hashimoto

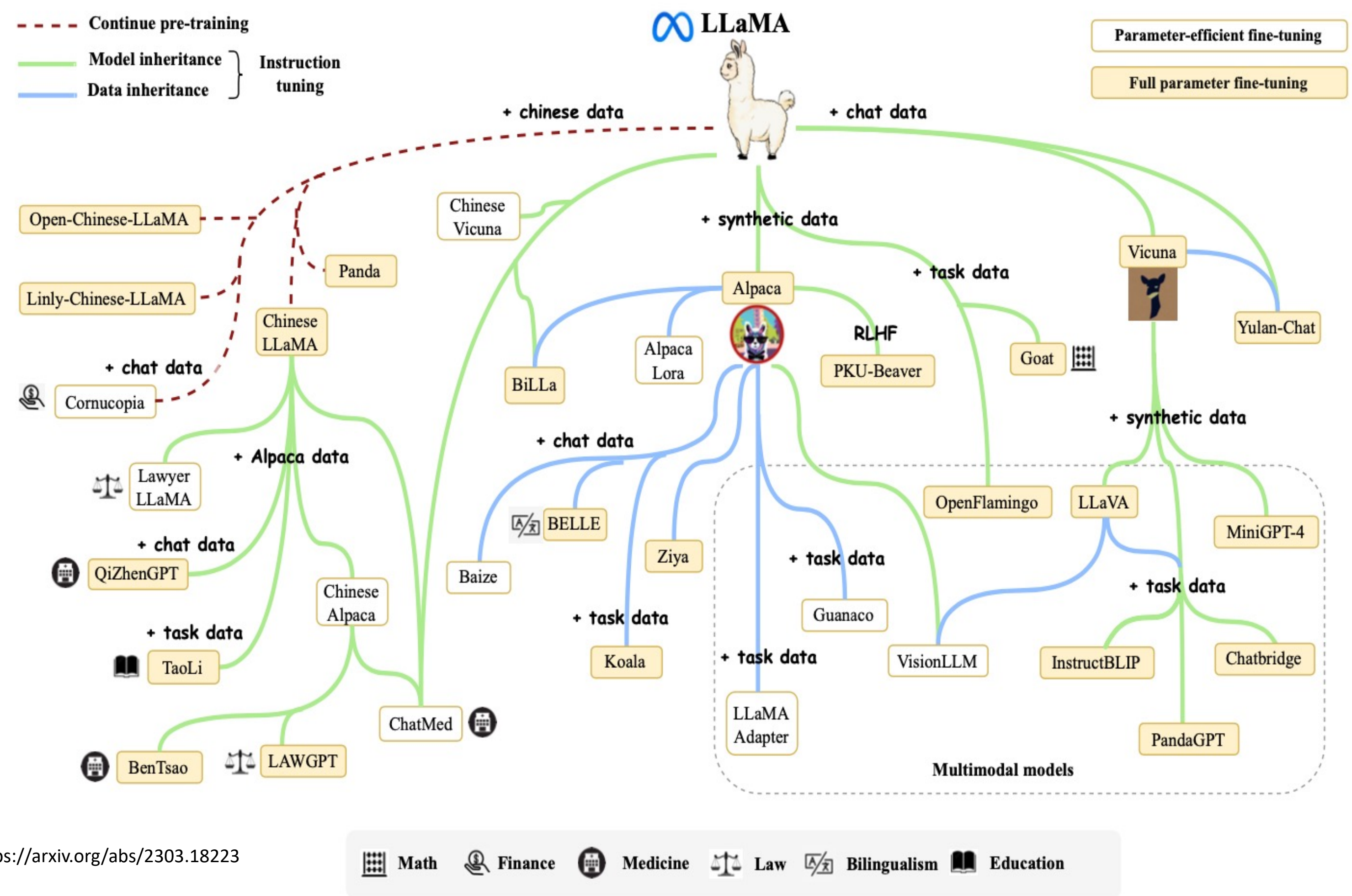
*We introduce **Alpaca 7B**, a model fine-tuned from the LLaMA 7B model on 52K instruction-following demonstrations. On our preliminary evaluation of single-turn instruction following, Alpaca behaves qualitatively similarly to OpenAI's text-davinci-003, while being surprisingly small and easy/cheap to reproduce (<600$). Checkout our code release on GitHub.*

Update: The public demo is now disabled. The original goal of releasing a demo was to disseminate our research in an accessible way. We feel that we have mostly achieved this goal, and given the hosting costs and the inadequacies of our content filters, we decided to bring down the demo.

Stanford Alpaca

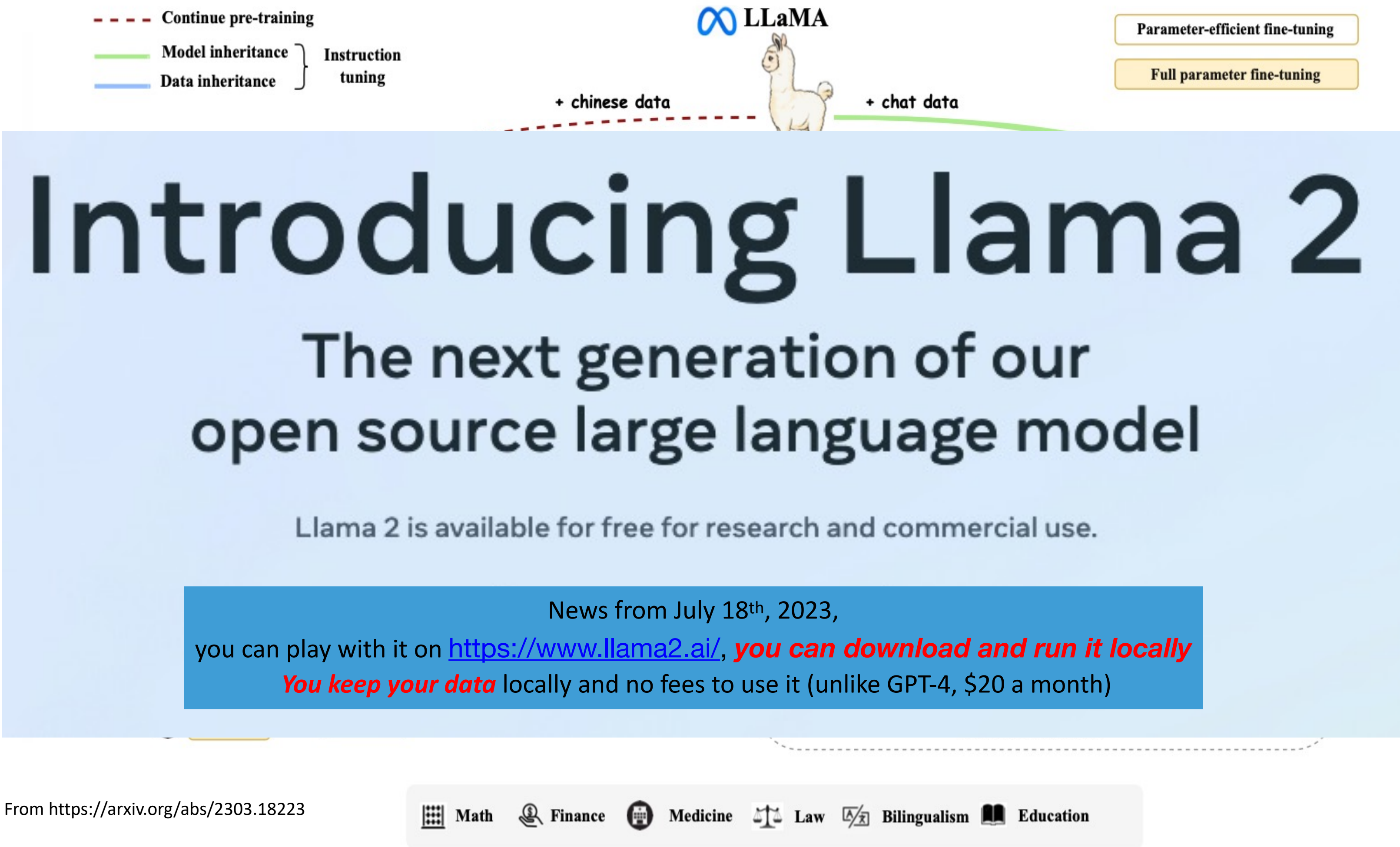
From https://crfm.stanford.edu/2023/03/13/alpaca.html

2023: The trigger of "open weights" revolution: Alpaca from Stanford, derived from LLaMa from Meta



From https://arxiv.org/abs/2303.18223

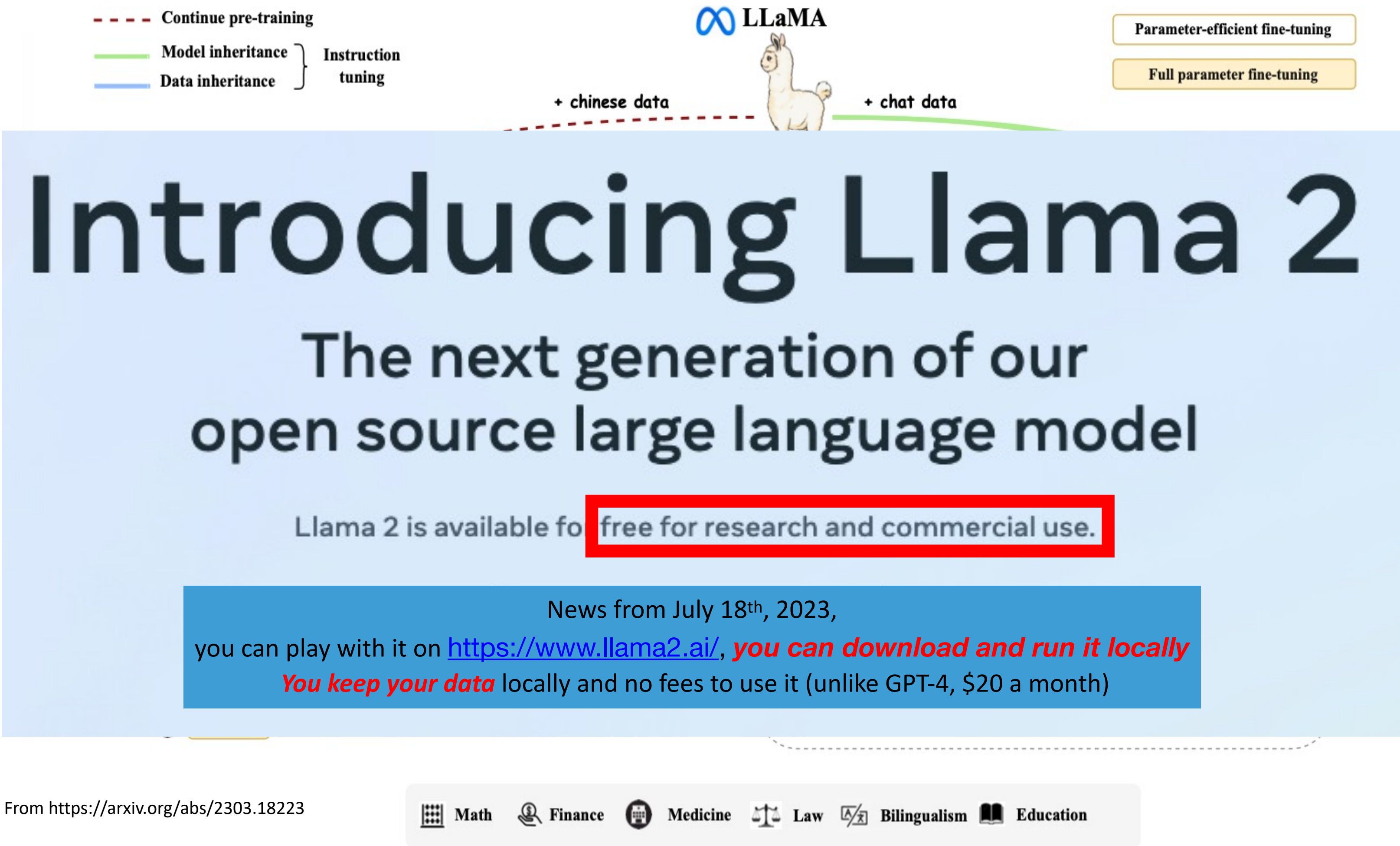
2023: The trigger of “open weights” revolution: Alpaca from Stanford, derived from LLaMa from Meta



News from July 18th, 2023,
you can play with it on https://www.llama2.ai/, ***you can download and run it locally***
You keep your data locally and no fees to use it (unlike GPT-4, $20 a month)

From https://arxiv.org/abs/2303.18223

2023: The trigger of "open weights" revolution: Alpaca from Stanford, derived from LLaMa from Meta



News from July 18th, 2023,
you can play with it on https://www.llama2.ai/, ***you can download and run it locally***
You keep your data locally and no fees to use it (unlike GPT-4, $20 a month)

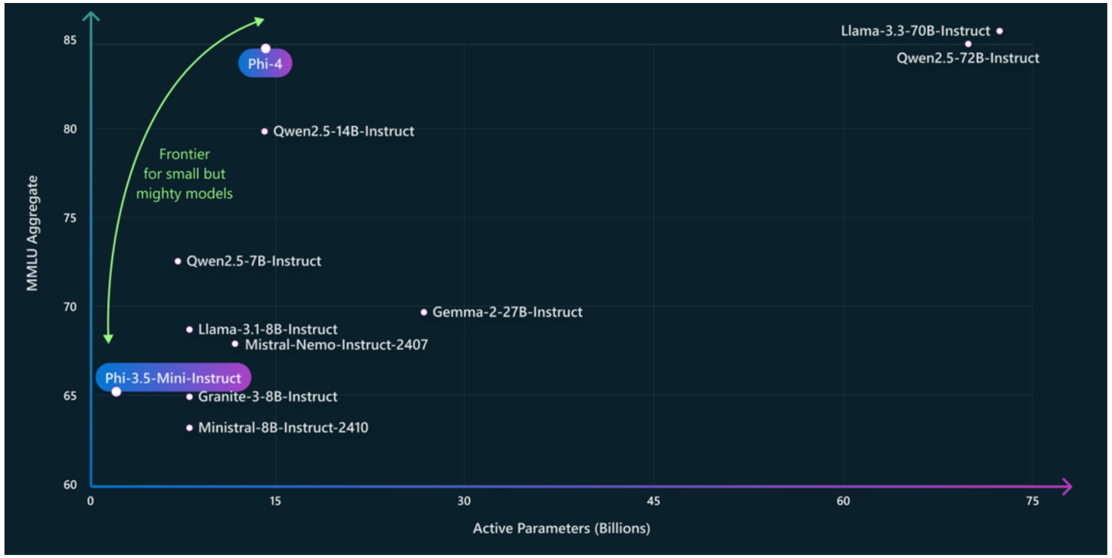
From https://arxiv.org/abs/2303.18223

2024: Smaller LLM models get more powerful

- Current models of about 10B parameters have better performances than the original ChatGPT of 2022

Model name	Announced	MMLU *
ChatGPT (gtp-3.5-turbo) 175B	November 2022	70
GPT-4 (gpt-4-0314) 1.76T?	March 2023	86.4
GPT-4o ?	May 2024	88.7
Llama 3.1 405B	July 2024	88.6
O1?	September 2024	92.3
Pixtral-12B	September 2024	69.2
Qwen 2.5 14B	September 2024	80
Llama 3.2 70B	December 2024	86.0
Phi-4 14B	December 2024	84.8
Deepseek-R1 671B / 37B MoE	January 2025	90.8

*Massive Multitask Language Understanding



From https://techcommunity.microsoft.com/blog/aiplatformblog/introducing-phi-4-microsoft%E2%80%99s-newest-small-language-model-specializing-in-comple/4357090

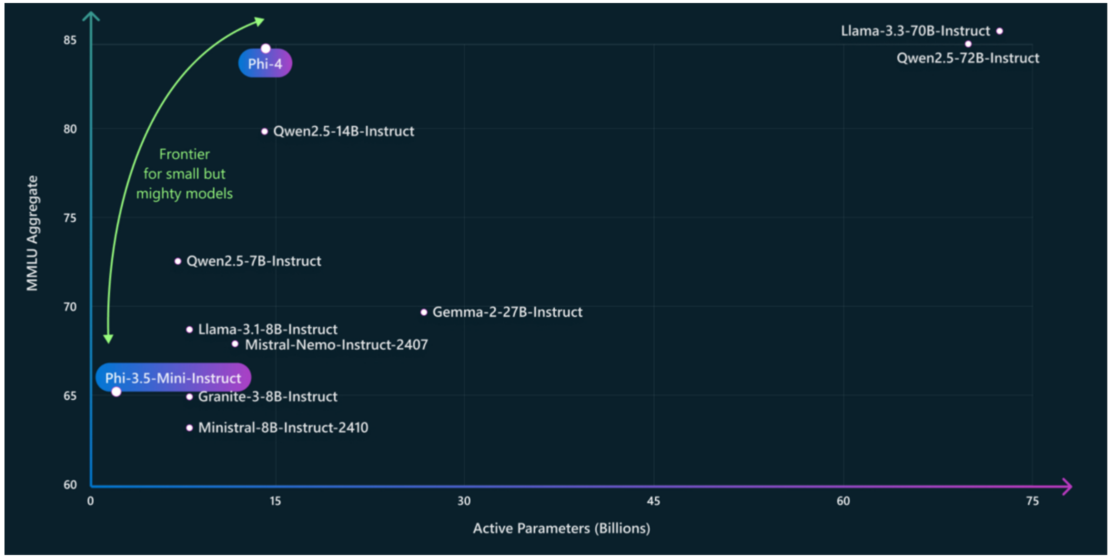
2024: Smaller LLM models get more powerful

- Current models of about 10B parameters have better performances than the original ChatGPT of 2022

Model name	Announced	MMLU *
ChatGPT (gtp-3.5-turbo) 175B	November 2022	70
GPT-4 (gpt-4-0314) 1.76T?	March 2023	86.4
GPT-4o ?	May 2024	88.7
Llama 3.1 405B	July 2024	88.6
O1?	September 2024	92.3
Pixtral-12B	September 2024	69.2
Qwen 2.5 14B	September 2024	80
Llama 3.2 70B	December 2024	86.0
Phi-4 14B	December 2024	84.8
Deepseek-R1 671B / 37B MoE	January 2025	90.8

*Massive Multitask Language Understanding

"Open source" models are catching up closed models with few months delay



From https://techcommunity.microsoft.com/blog/aiplatformblog/introducing-phi-4-microsoft%E2%80%99s-newest-small-language-model-specializing-in-comple/4357090

Distillation: Smaller Models Can Be Powerful Too*

*“We demonstrate that the **reasoning patterns of larger models can be distilled into smaller models**, resulting in better performance compared to the reasoning patterns discovered through RL on small models.*

- *Using the reasoning data generated by DeepSeek-R1, we fine-tuned several dense models that are widely used in the research community. The evaluation results demonstrate that the distilled smaller dense models perform exceptionally well on benchmarks. We open-source distilled 1.5B, 7B, 8B, 14B, 32B, and 70B checkpoints based on Qwen2.5 and Llama3 series to the community.”*

Model	AIME 2024 pass@1	AIME 2024 cons@64	MATH-500 pass@1	GPQA Diamond pass@1	LiveCodeBench pass@1	CodeForces rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481

* From https://github.com/deepseek-ai/DeepSeek-R1?tab=readme-ov-file#distilled-model-evaluation

The American Invitational Mathematics Examination (AIME) is a selective and prestigious 15-question 3-hour test given since 1983 to those who rank in the top 2.5% on the AMC 10.

Distillation: Smaller Models Can Be Powerful Too*

*"We demonstrate that the **reasoning patterns of larger models can be distilled into smaller models**, resulting in better performance compared to the reasoning patterns discovered through RL on small models.*

- *Using the reasoning data generated by DeepSeek-R1, we fine-tuned several dense models that are widely used in the research community. The evaluation results demonstrate that the distilled smaller dense models perform exceptionally well on benchmarks. We open-source distilled 1.5B, 7B, 8B, 14B, 32B, and 70B checkpoints based on Qwen2.5 and Llama3 series to the community."*

Model	AIME 2024 pass@1	AIME 2024 cons@64	MATH-500 pass@1	GPQA Diamond pass@1	LiveCodeBench pass@1	CodeForces rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481

* From https://github.com/deepseek-ai/DeepSeek-R1?tab=readme-ov-file#distilled-model-evaluation

The American Invitational Mathematics Examination (AIME) is a selective and prestigious 15-question 3-hour test given since 1983 to those who rank in the top 2.5% on the AMC 10.

What’s new (end 2024)

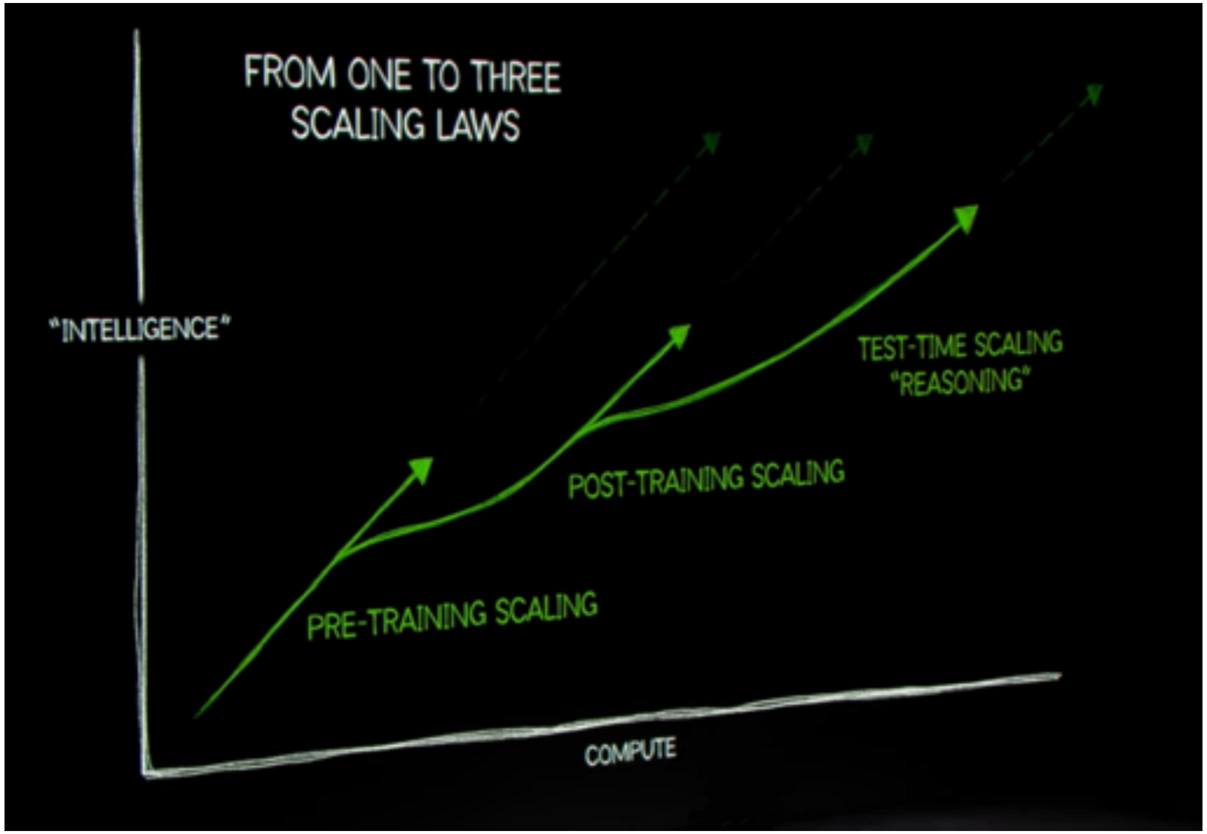
Scaling is no more the only way to increase performances

- Small (specialized) models gets performances of (older) larger LLMs
  - Training by artificial data / larger models

What's new (end 2024)

Scaling is no more the only way to increase performances

- Small (specialized) models gets performances of (older) larger LLMs
  - Training by artificial data / larger models
- **Test-time compute** / inference time scaling
  - Pioneering by o1/ o3 from OpenAI
  - Democratized by DeepSeek-R1 (Open weights)

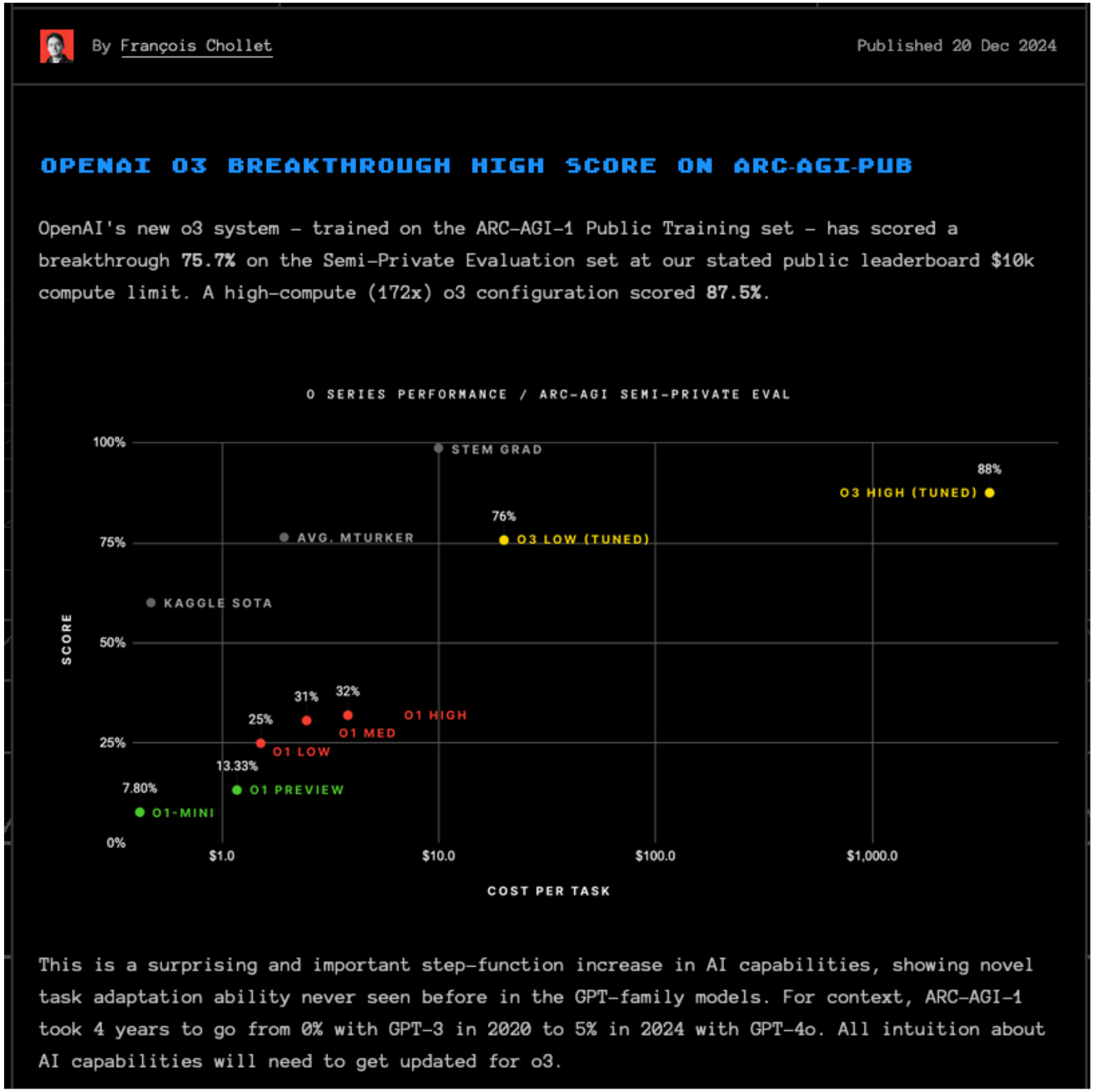


From Jensen Huang keynote at CES 2025

The test-time compute (re)volution

"***Test-time compute***": additional computational power is used during the inference stage to improve the quality of the response, rather than just relying on the pre-trained model's capabilities.

It allows the LLM to "think harder", "**think step-by-step**" on a problem by performing extra calculations at runtime to produce better results, often involving techniques like generating multiple solutions and selecting the best one.



By François Chollet Published 20 Dec 2024

OPENAI O3 BREAKTHROUGH HIGH SCORE ON ARC-AGI-PUB

OpenAI's new o3 system - trained on the ARC-AGI-1 Public Training set - has scored a breakthrough **75.7%** on the Semi-Private Evaluation set at our stated public leaderboard $10k compute limit. A high-compute (172x) o3 configuration scored **87.5%**.

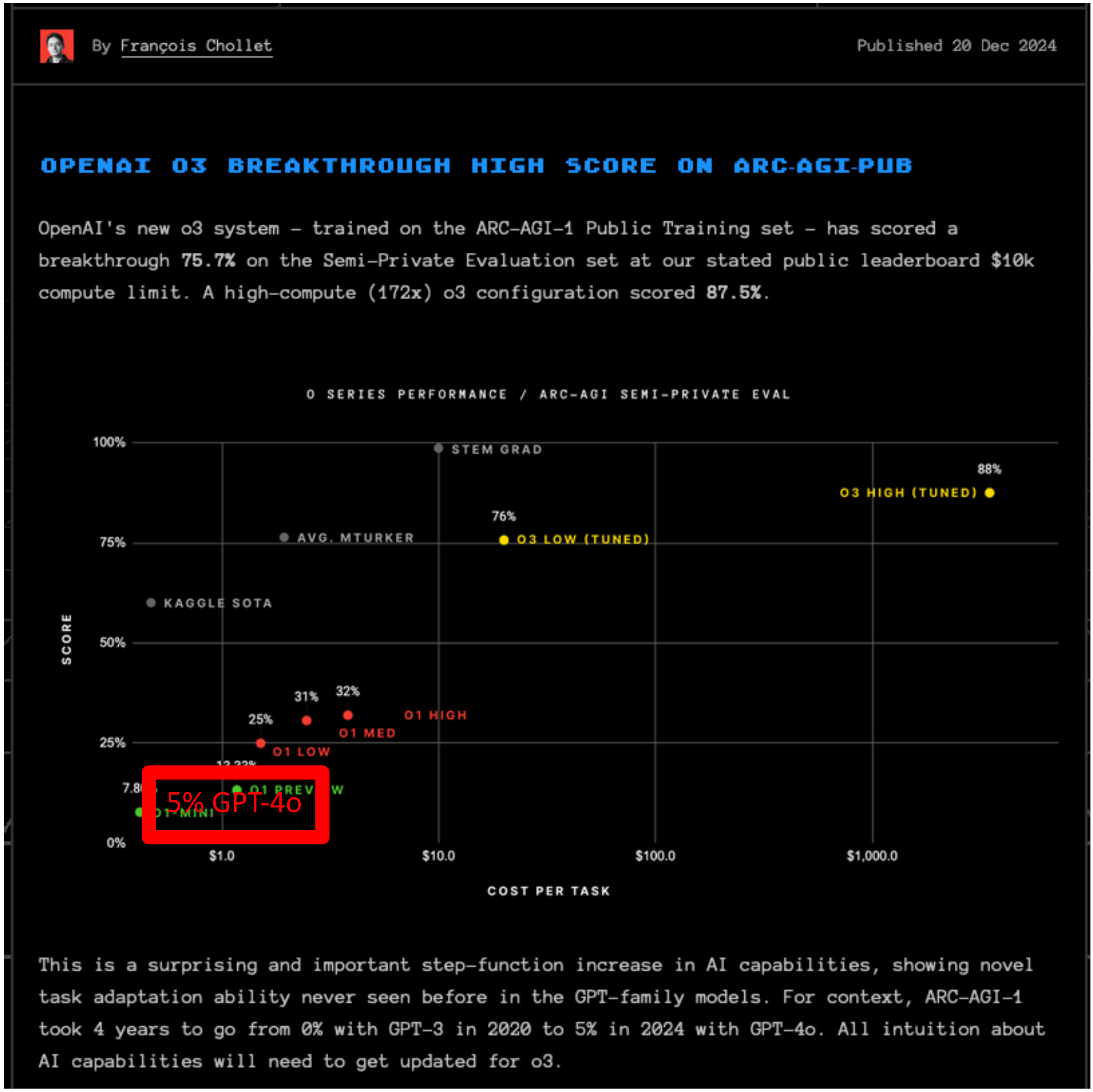
This is a surprising and important step-function increase in AI capabilities, showing novel task adaptation ability never seen before in the GPT-family models. For context, ARC-AGI-1 took 4 years to go from 0% with GPT-3 in 2020 to 5% in 2024 with GPT-4o. All intuition about AI capabilities will need to get updated for o3.

From https://arcprize.org/blog/oai-o3-pub-breakthrough

The test-time compute (re)volution

"***Test-time compute***": additional computational power is used during the inference stage to improve the quality of the response, rather than just relying on the pre-trained model's capabilities.

It allows the LLM to "think harder", "**think step-by-step**" on a problem by performing extra calculations at runtime to produce better results, often involving techniques like generating multiple solutions and selecting the best one.

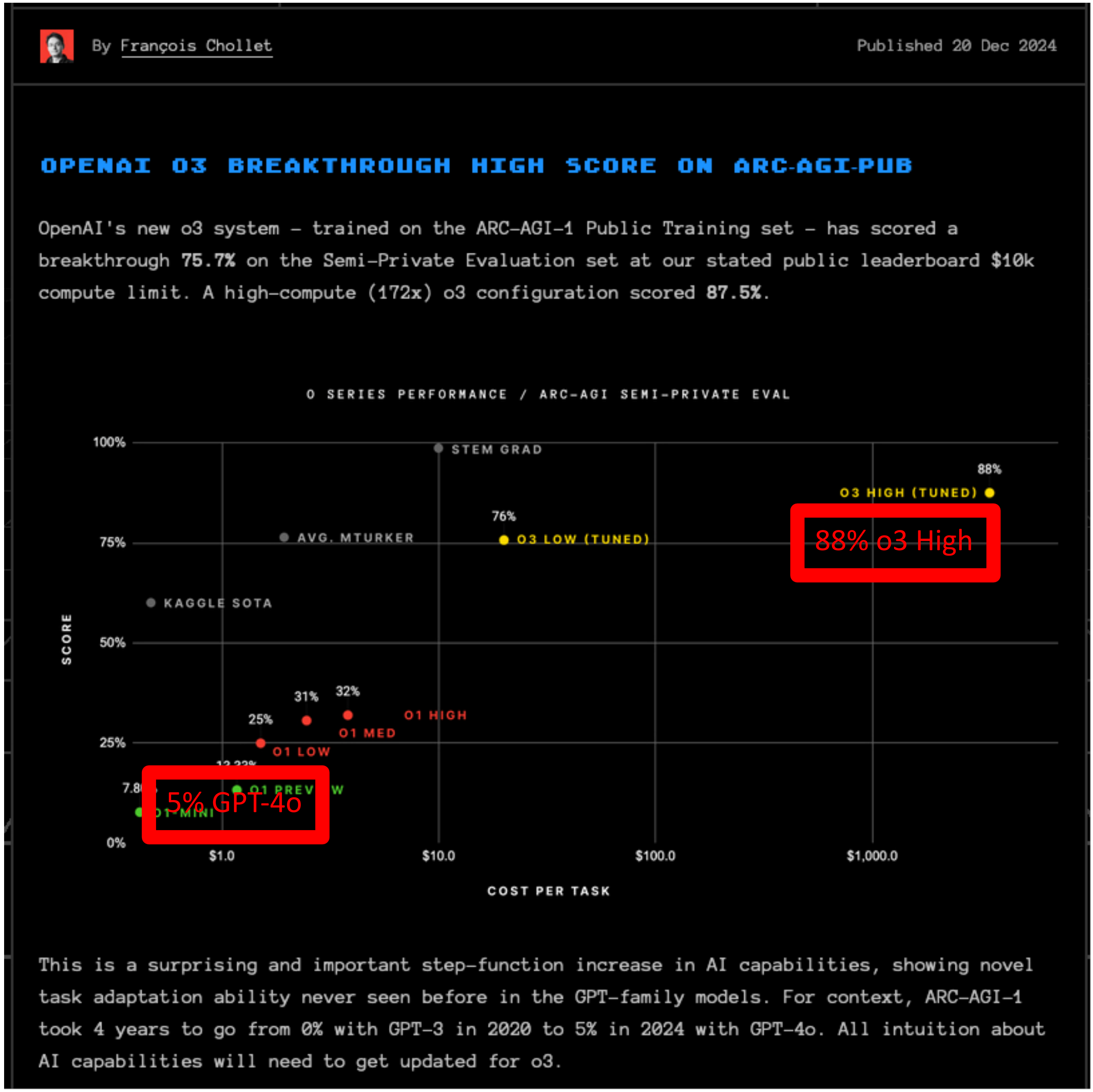


From https://arcprize.org/blog/oai-o3-pub-breakthrough

The test-time compute (re)volution

"***Test-time compute***": additional computational power is used during the inference stage to improve the quality of the response, rather than just relying on the pre-trained model's capabilities.

It allows the LLM to "think harder", "**think step-by-step**" on a problem by performing extra calculations at runtime to produce better results, often involving techniques like generating multiple solutions and selecting the best one.



From https://arcprize.org/blog/oai-o3-pub-breakthrough

What's new (end 2024)

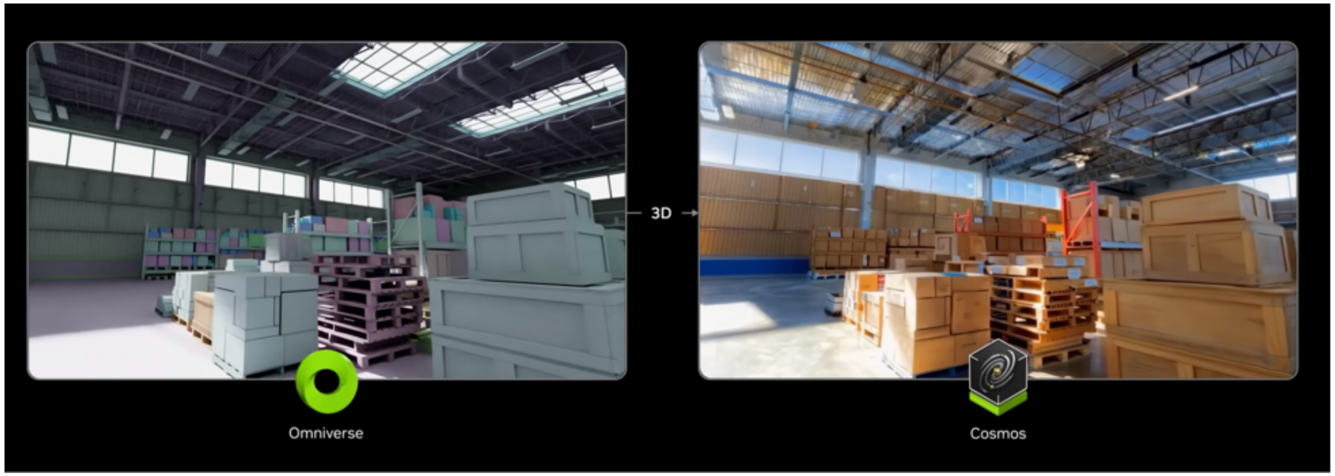
"World" foundation models:

"World models are generative AI models that understand the dynamics of the real world, including physics and spatial properties.

…

They understand the physical qualities of real-world environments by learning to represent and predict dynamics like motion, force, and spatial relationships from sensory data."*

Digital twin



World model

* From https://www.nvidia.com/en-us/glossary/world-models/

Agentic AI the future of AI?

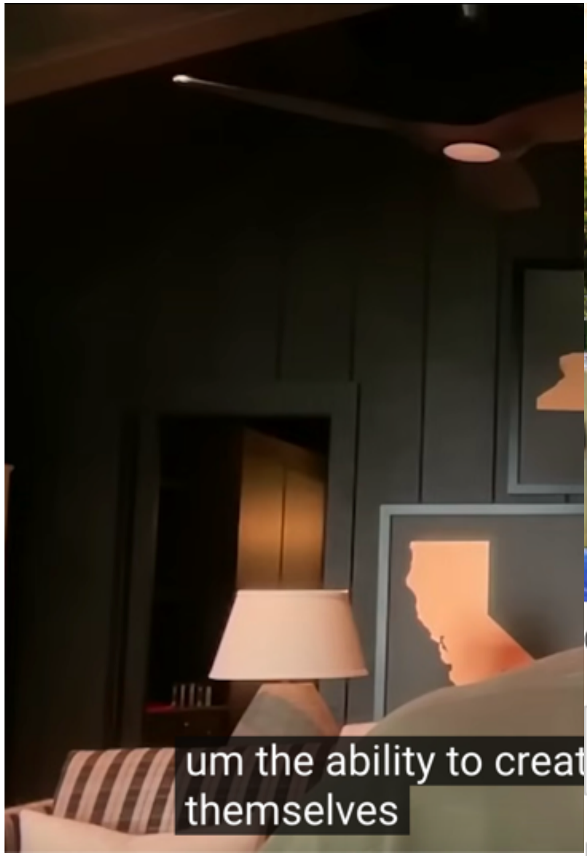
- Using a set of small specialized LLMs can have similar performances than of a large LLM
- Only a subset of the LLM are activated simultaneously (Mixture of Experts)

Agentic AI the future of AI?

- Using a set of small specialized LLMs can have similar performances than of a large LLM
- Only a subset of the LLM are activated simultaneously (Mixture of Experts)

Agentic AI the future of AI?

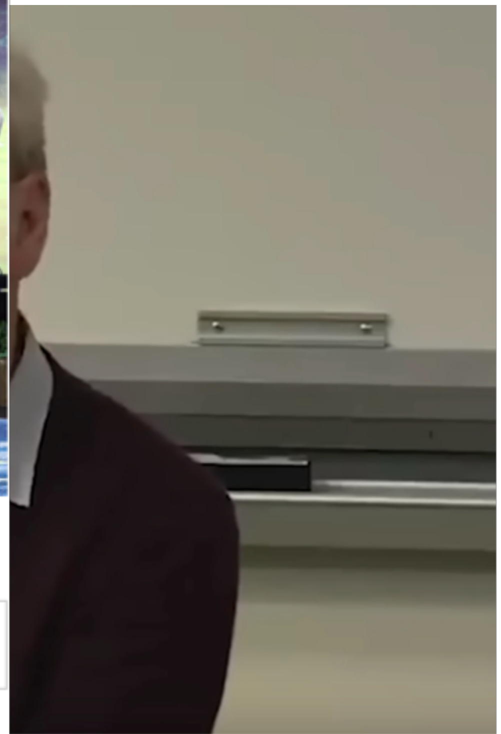
- Using a set of small specialized LLMs can have similar performances than of a large LLM
- Only a subset of the LLM are activated simultaneously (Mixture of Experts)



(Image credit: Getty Images / Justin Sullivan)

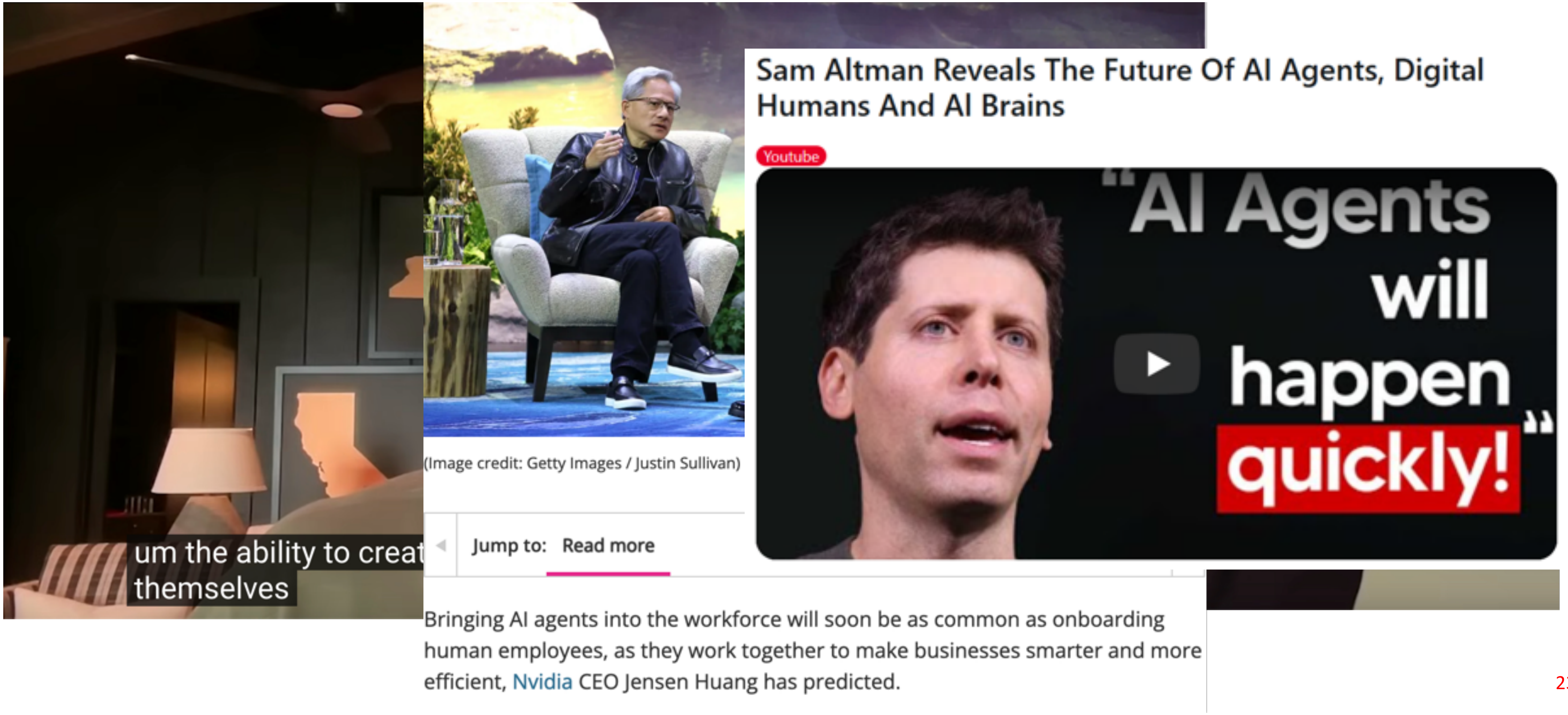
Jump to: Read more

Bringing AI agents into the workforce will soon be as common as onboarding human employees, as they work together to make businesses smarter and more efficient, Nvidia CEO Jensen Huang has predicted.

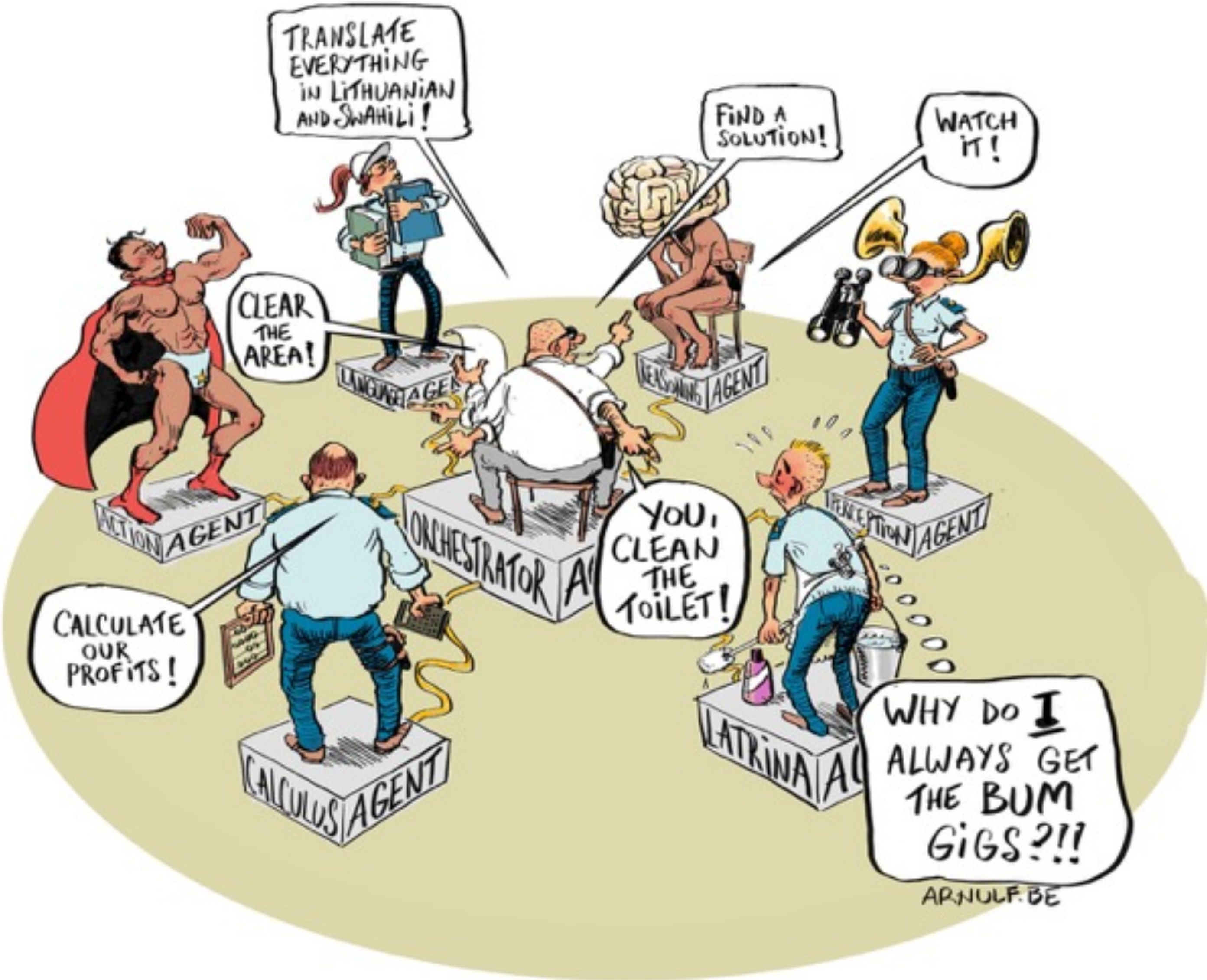


Agentic AI the future of AI?

- Using a set of small specialized LLMs can have similar performances than of a large LLM
- Only a subset of the LLM are activated simultaneously (Mixture of Experts)



Agentic AI in one slide



AI can run at the edge…

AI data center: 5 MW

AI can run at the edge...

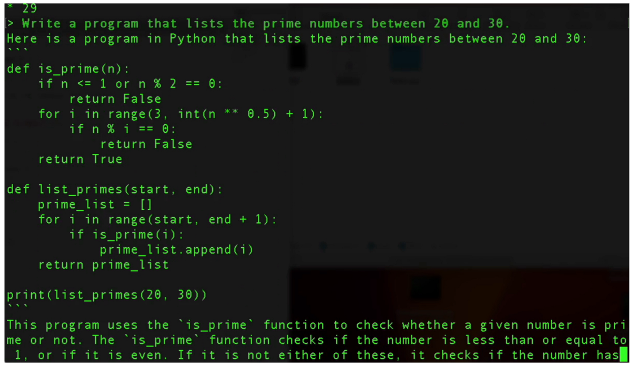
AI data center: 5 MW

Project Digits: 150W ???

AI can run at the edge…

LLM running locally on Mac mini: about 20W

AI data center: 5 MW



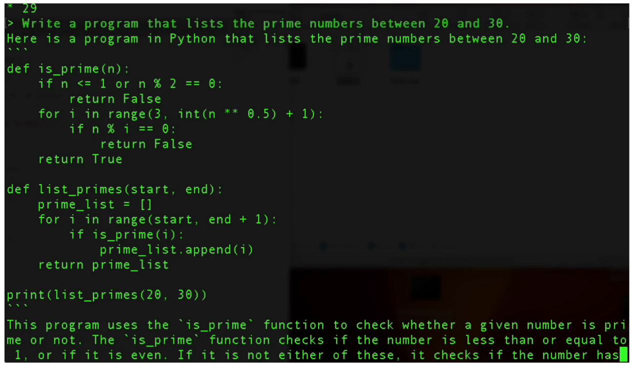
````
* 29
> Write a program that lists the prime numbers between 20 and 30.
Here is a program in Python that lists the prime numbers between 20 and 30:
```
def is_prime(n):
    if n <= 1 or n % 2 == 0:
        return False
    for i in range(3, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

def list_primes(start, end):
    prime_list = []
    for i in range(start, end + 1):
        if is_prime(i):
            prime_list.append(i)
    return prime_list

print(list_primes(20, 30))
```
This program uses the `is_prime` function to check whether a given number is pri
me or not. The `is_prime` function checks if the number is less than or equal to
 1, or if it is even. If it is not either of these, it checks if the number has
````

Project Digits: 150W ???

AI can run at the edge…

LLM running locally on Mac mini: about 20W

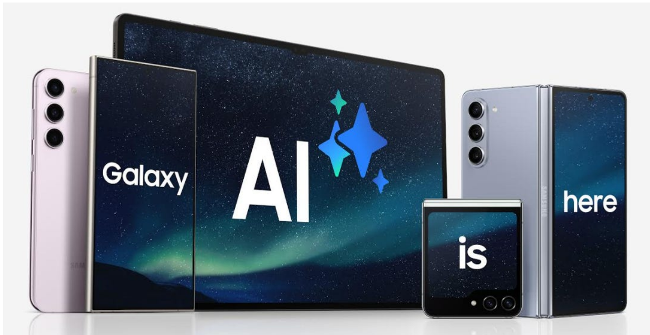
AI data center: 5 MW



```
> Write a program that lists the prime numbers between 20 and 30.
Here is a program in Python that lists the prime numbers between 20 and 30:
```
def is_prime(n):
    if n <= 1 or n % 2 == 0:
        return False
    for i in range(3, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

def list_primes(start, end):
    prime_list = []
    for i in range(start, end + 1):
        if is_prime(i):
            prime_list.append(i)
    return prime_list

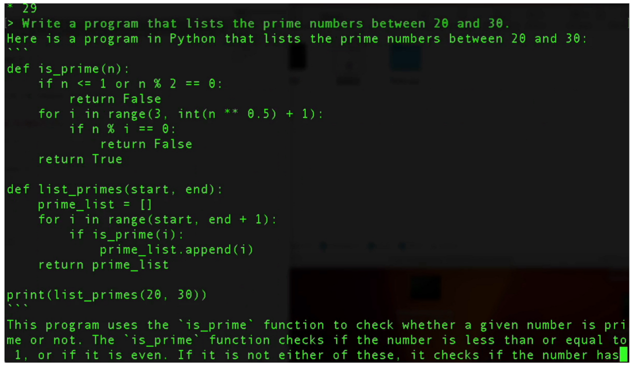
print(list_primes(20, 30))

This program uses the `is_prime` function to check whether a given number is pri
me or not. The `is_prime` function checks if the number is less than or equal to
 1, or if it is even. If it is not either of these, it checks if the number has
```

Project Digits: 150W ???




AI smartphones: 5-10W


```

AI can run at the edge…

LLM running locally on Mac mini: about 20W

AI data center: 5 MW

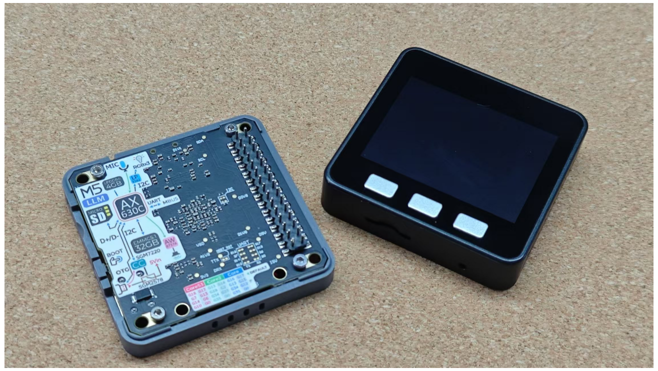
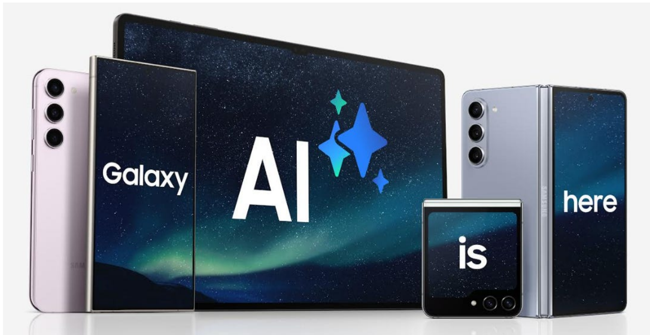


```
> Write a program that lists the prime numbers between 20 and 30.
Here is a program in Python that lists the prime numbers between 20 and 30:
```

def is_prime(n):
    if n <= 1 or n % 2 == 0:
        return False
    for i in range(3, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

def list_primes(start, end):
    prime_list = []
    for i in range(start, end + 1):
        if is_prime(i):
            prime_list.append(i)
    return prime_list

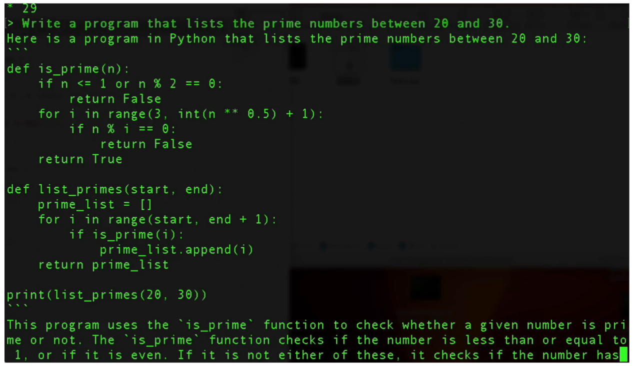
print(list_primes(20, 30))

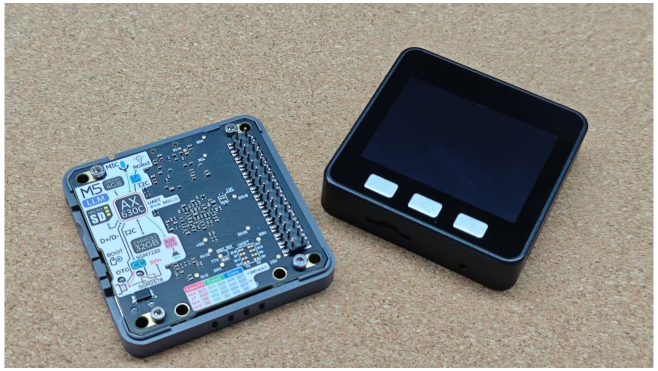
This program uses the `is_prime` function to check whether a given number is pri
me or not. The `is_prime` function checks if the number is less than or equal to
 1, or if it is even. If it is not either of these, it checks if the number has
```



IoT LLM box: 1-2 W

Project Digits: 150W ???




AI smartphones: 5-10W


```

AI can run at the edge…

LLM running locally on Mac mini: about 20W

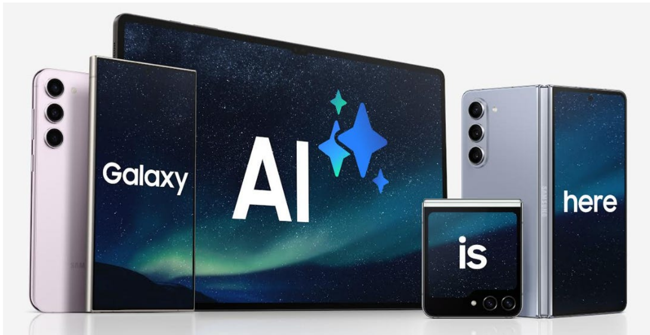
AI data center: 5 MW



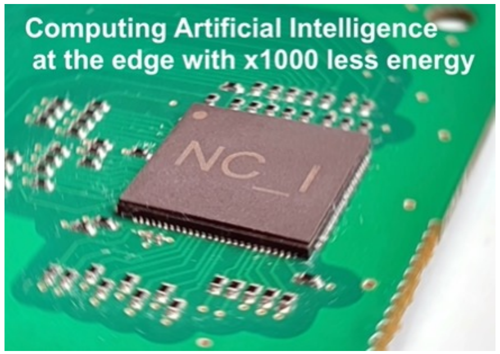


IoT LLM box: 1-2 W

Project Digits: 150W ???

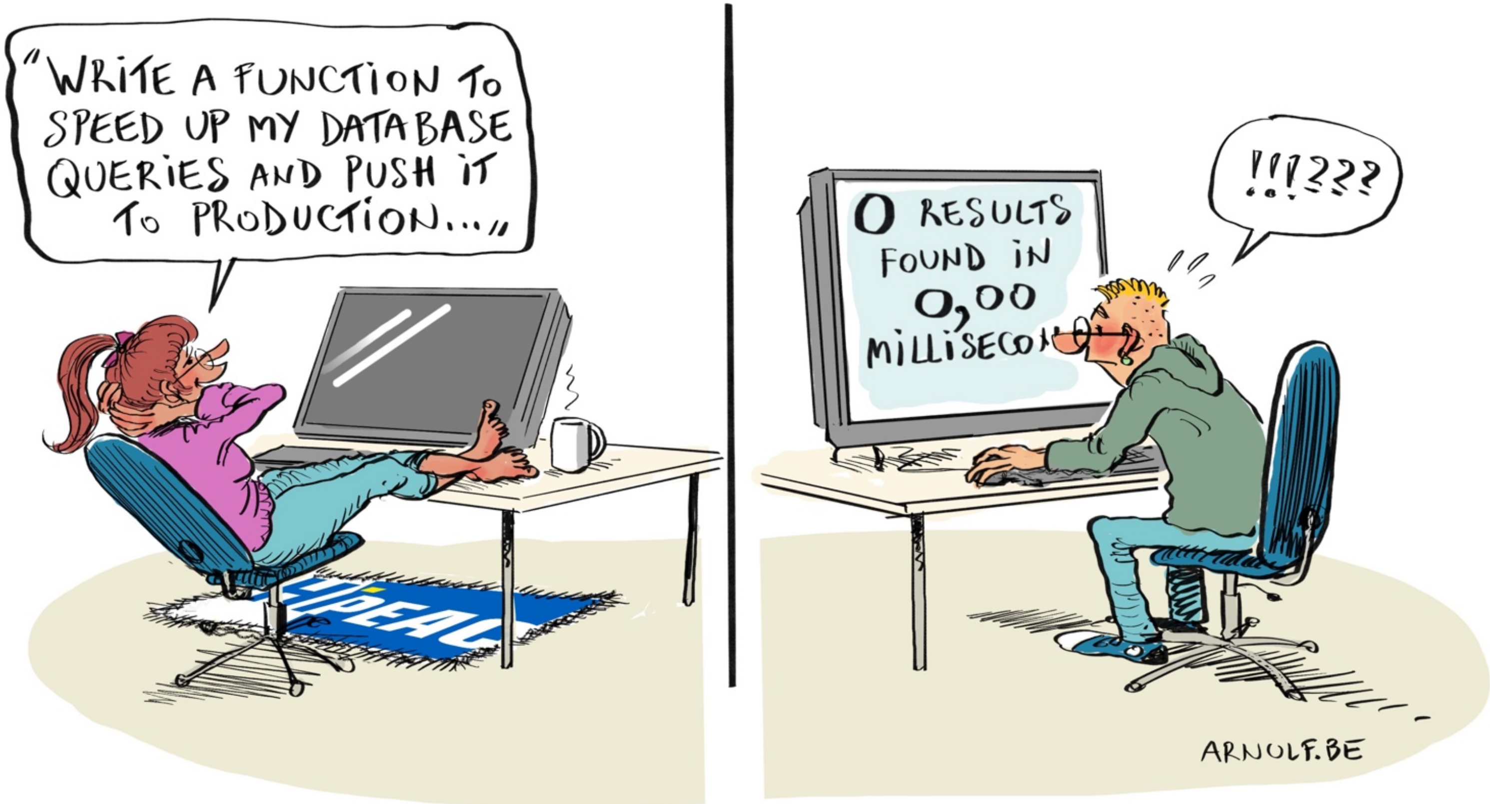


AI smartphones: 5-10W



Object detection on
HD images at 30FPS for 23mW

Various applications of current AI: code generation



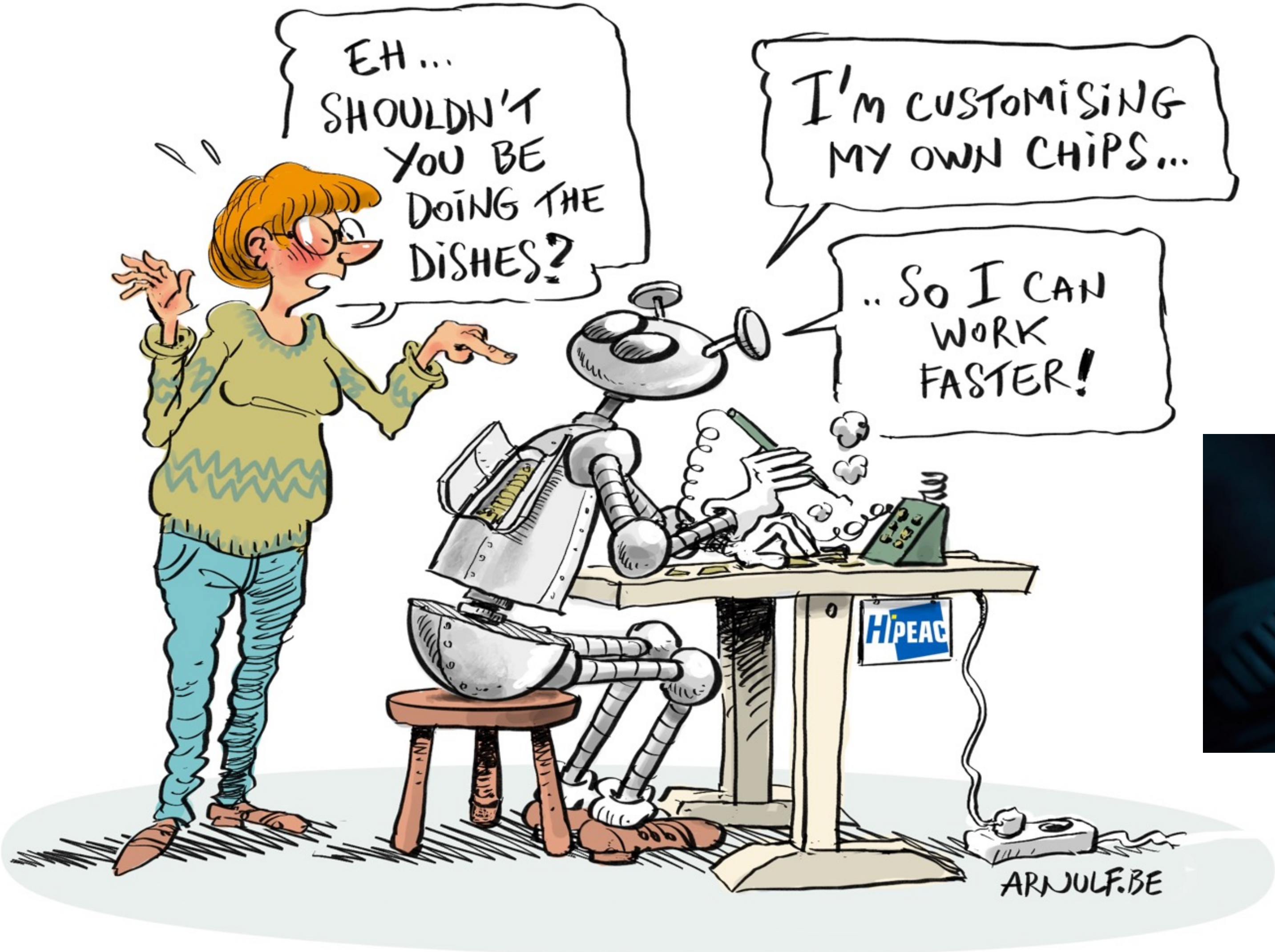
Video and sound generation (Veo3 from Google)

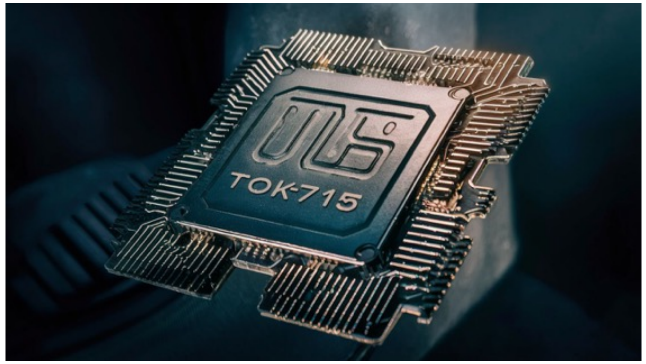
Video and sound generation (Veo3 from Google)

Video and sound generation (Veo3 from Google)

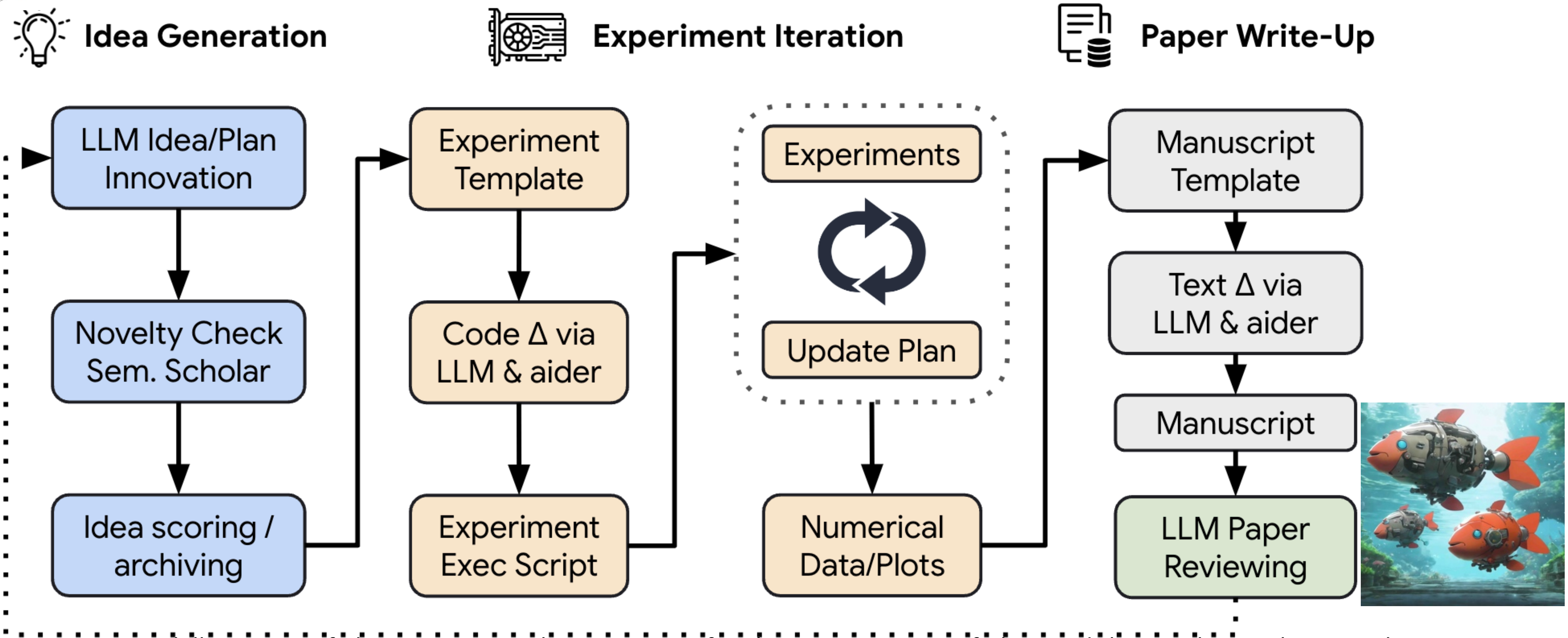
cybersecurity attack and protection

AI help design chips for AI…





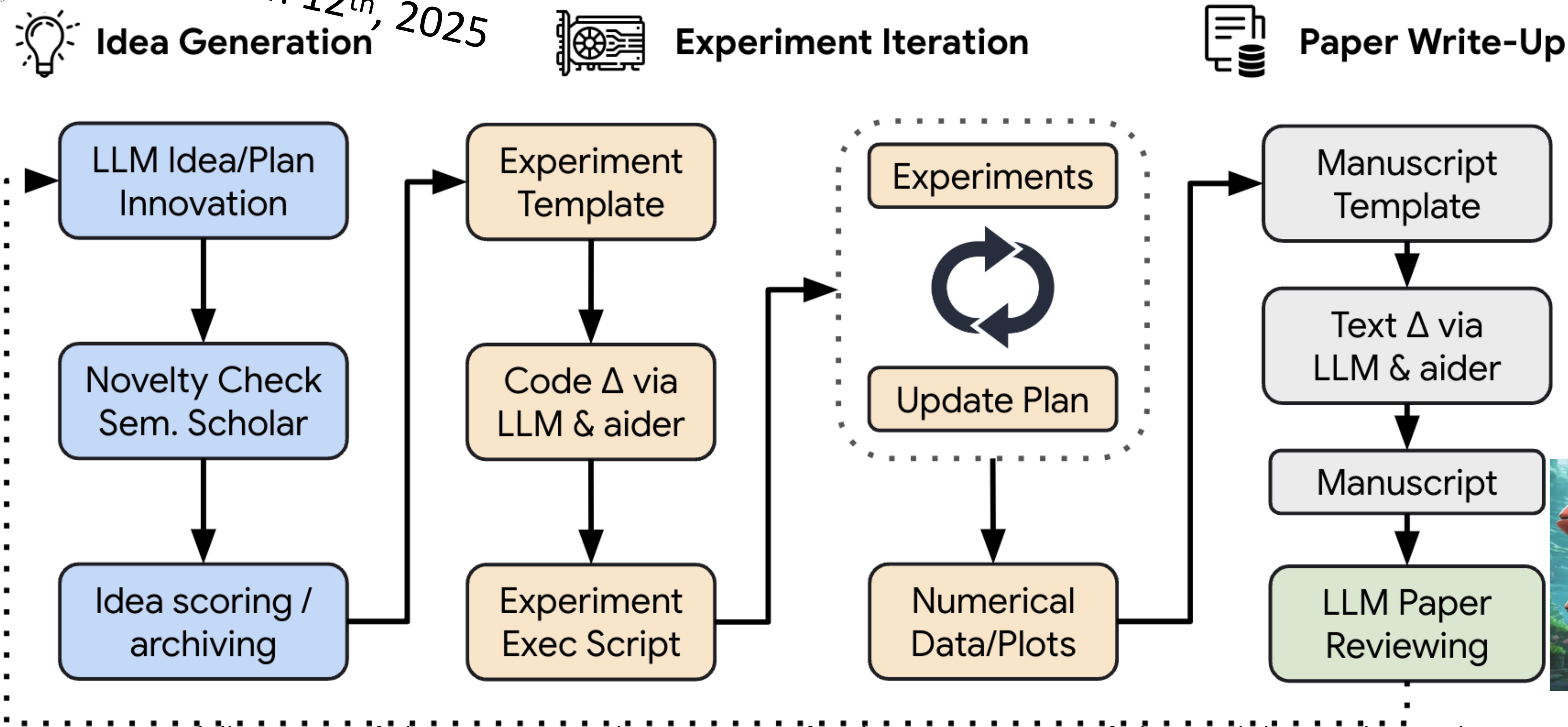
The AI scientist from Sakana.ai



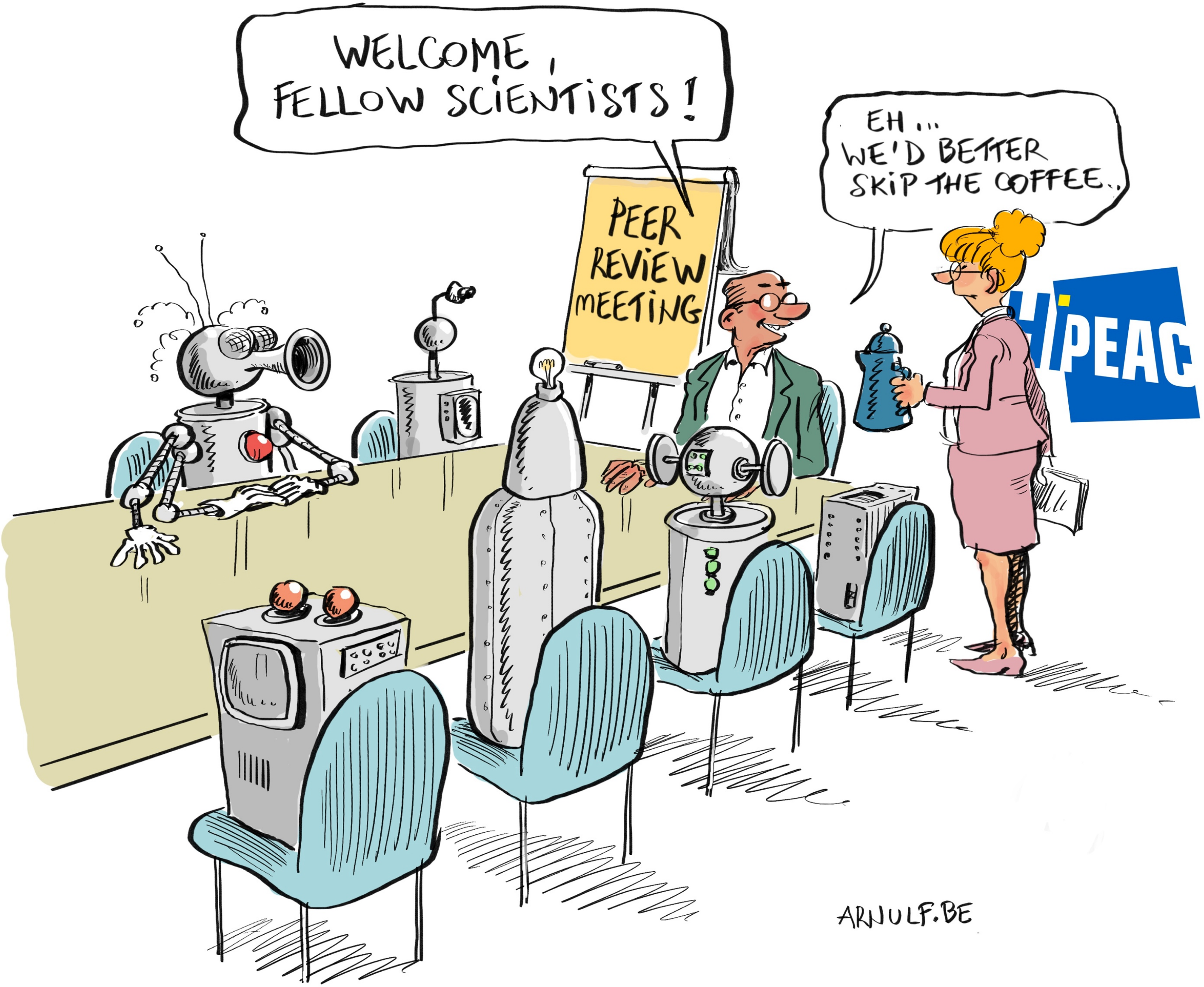
Conceptual illustration of The AI Scientist. The AI Scientist first brainstorms a set of ideas and then evaluates their novelty. Next, it edits a codebase powered by recent advances in automated code generation to implement the novel algorithms. The Scientist then runs experiments to gather results consisting of both numerical data and visual summaries. It crafts a scientific report, explaining and contextualizing the results. Finally, the AI Scientist generates an automated peer review based on top-tier machine learning conference standards. This review helps refine the current project and informs future generations of open-ended ideation. From https://sakana.ai/ai-scientist/

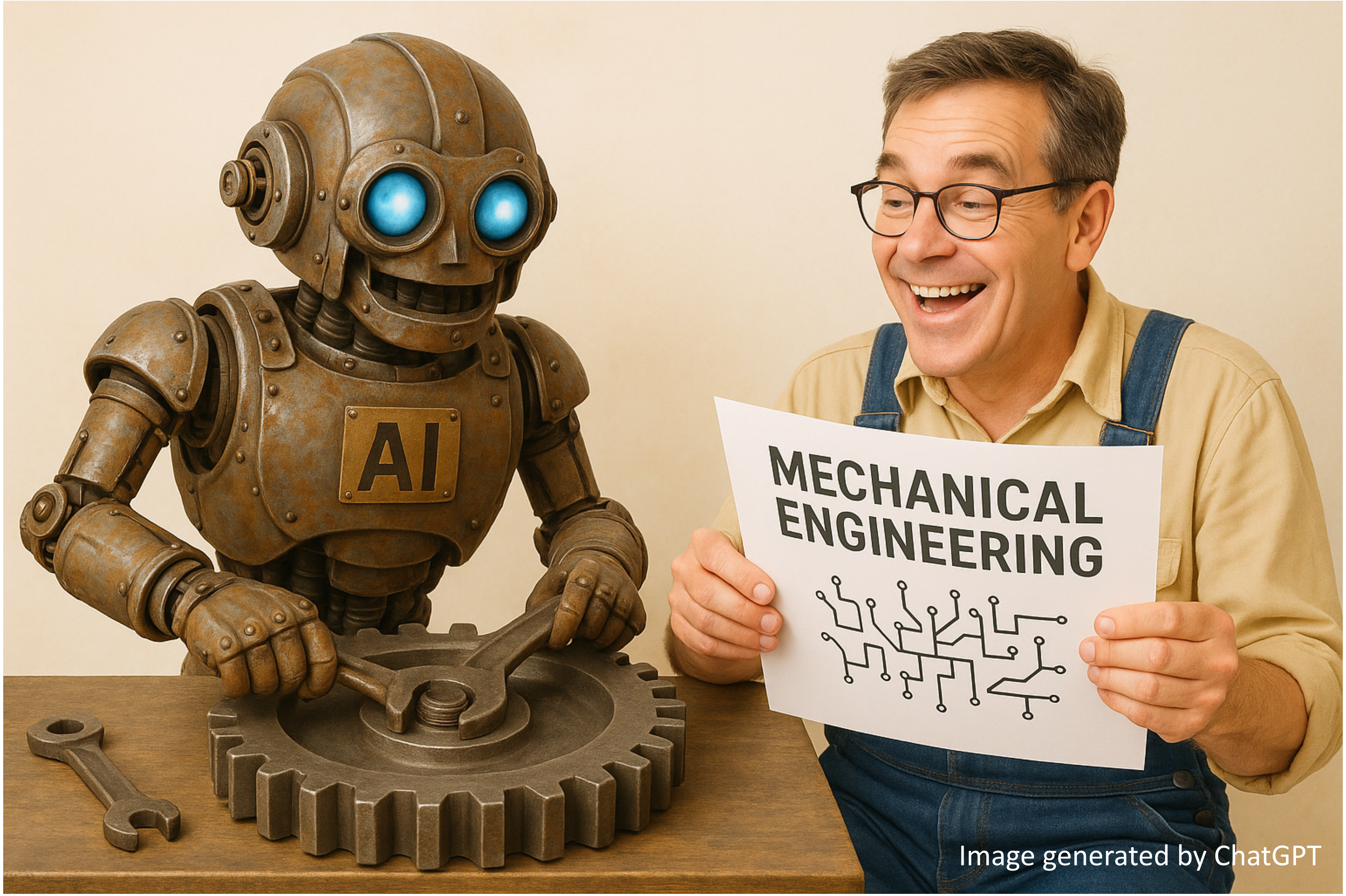
The AI scientist from Sakana.ai

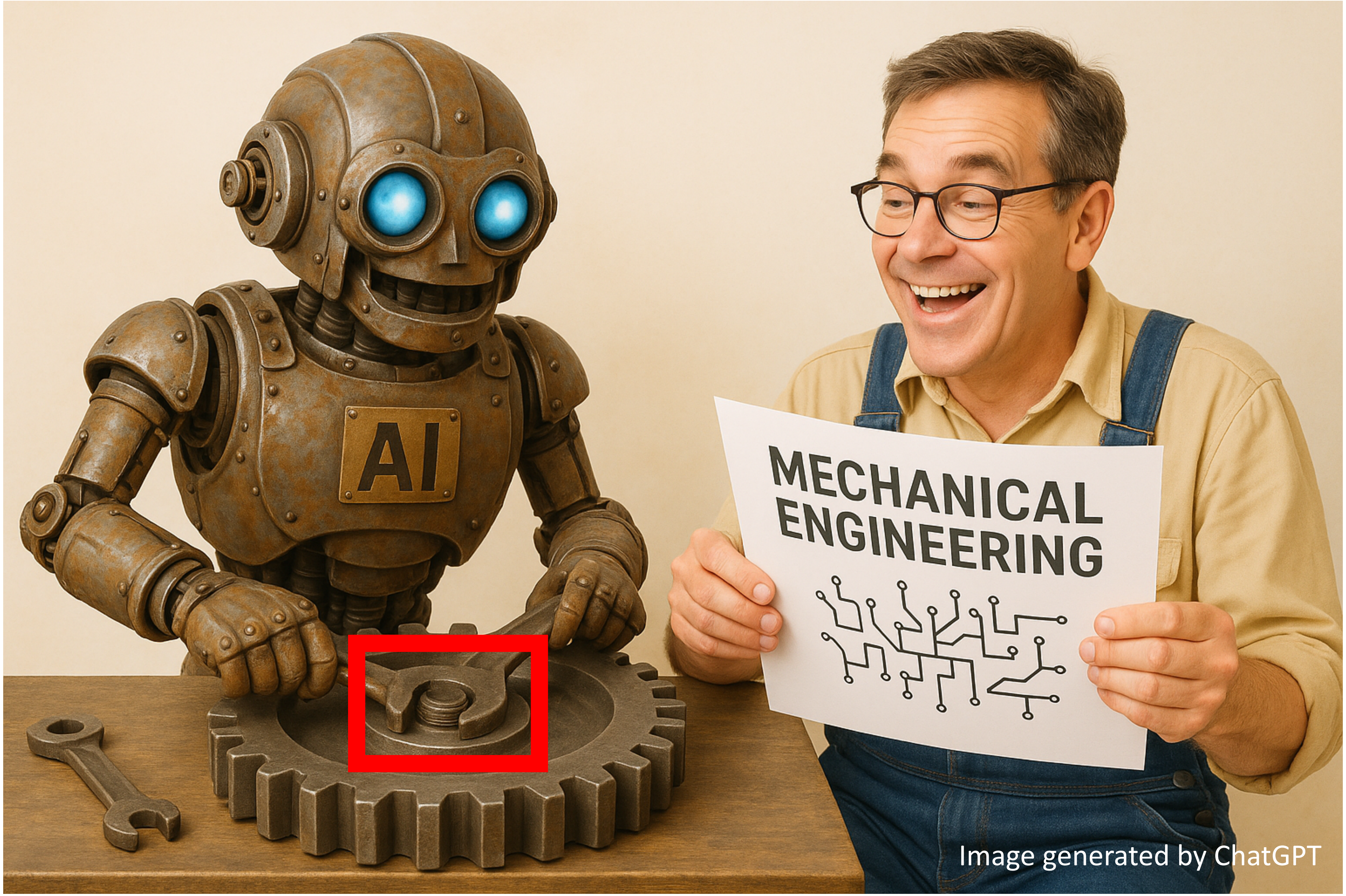
March 12th, 2025



Conceptual illustration of The AI Scientist. The AI Scientist first brainstorms a set of ideas and then evaluates their novelty. Next, it edits a codebase powered by recent advances in automated code generation to implement the novel algorithms. The Scientist then runs experiments to gather results consisting of both numerical data and visual summaries. It crafts a scientific report, explaining and contextualizing the results. Finally, the AI Scientist generates an automated peer review based on top-tier machine learning conference standards. This review helps refine the current project and informs future generations of open-ended ideation. From https://sakana.ai/ai-scientist/


WELCOME, FELLOW SCIENTISTS!
PEER REVIEW MEETING
EH... WE'D BETTER SKIP THE COFFEE..
HiPEAC
ARNULF.BE


AI
MECHANICAL ENGINEERING
Image generated by ChatGPT


AI
MECHANICAL
ENGINEERING
Image generated by ChatGPT

CONCLUSION: WE LIVE AN EXCITING TIME!

CONCLUSION: WE LIVE AN EXCITING TIME!