Artificial Neural Networks

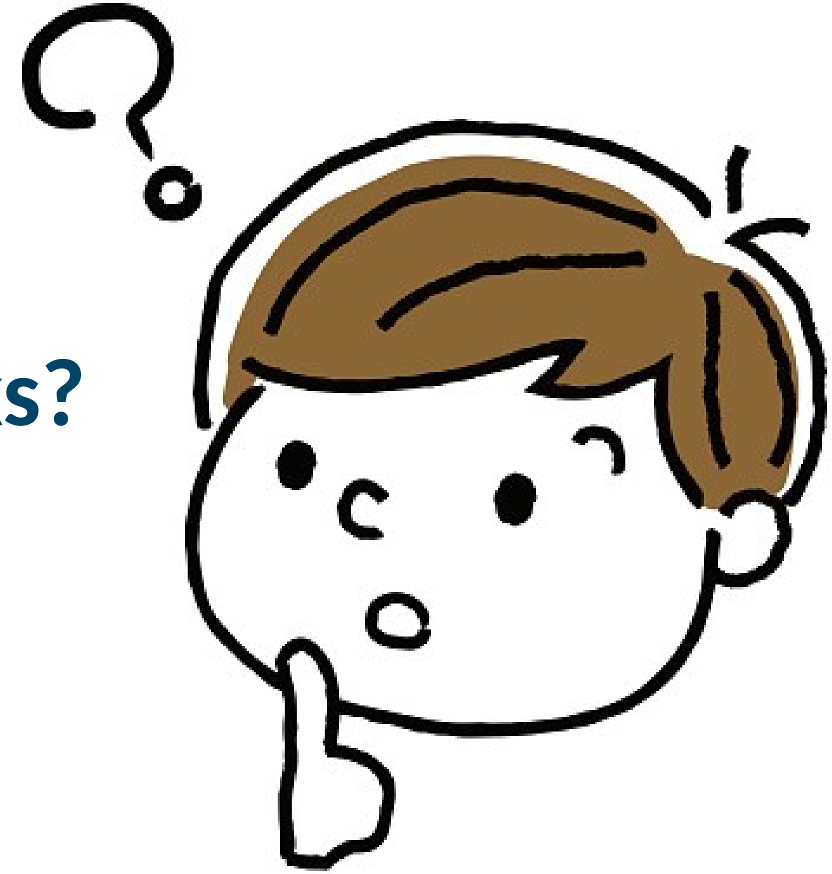
[A Classical Perspective]

José Oramas

University
of Antwerp

Fundamentally speaking…

Why Artificial Neural Networks?



Solving Intelligence

What is Intelligence?

"A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects."

Robert A Heinlein
Science Fiction Author

What is Intelligence?

More Formally…

… ability to achieve goals in a wide range of environments

$$\Upsilon(\pi) := \sum_{\mu \in E} 2^{-K(\mu)} V_{\mu}^{\pi}.$$

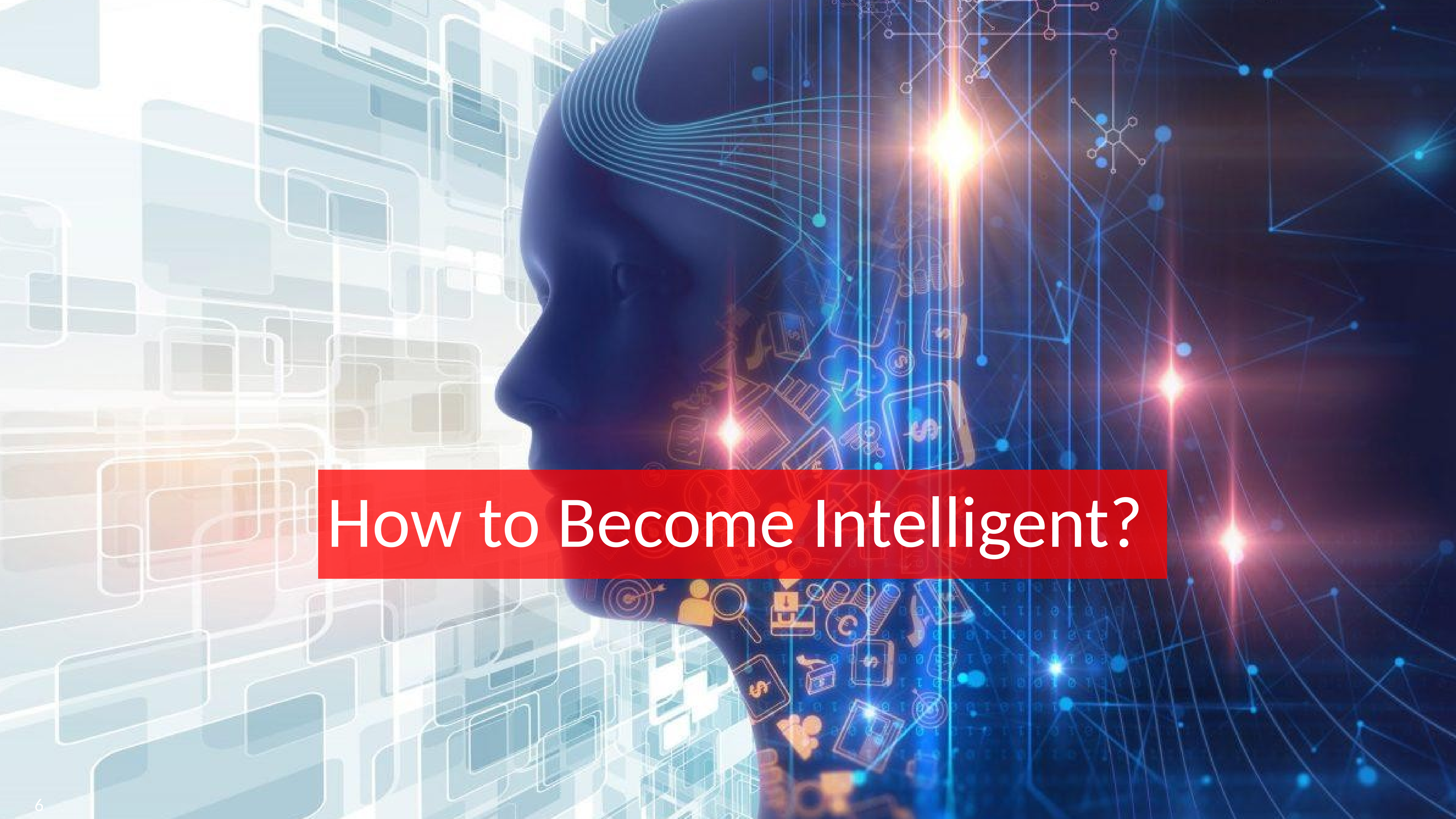
Measure of Intelligence

Over possible environments

Complexity penalty

achieved Value

Universal Intelligence: A Definition of Machine Intelligence, Legg & Hutter 2007

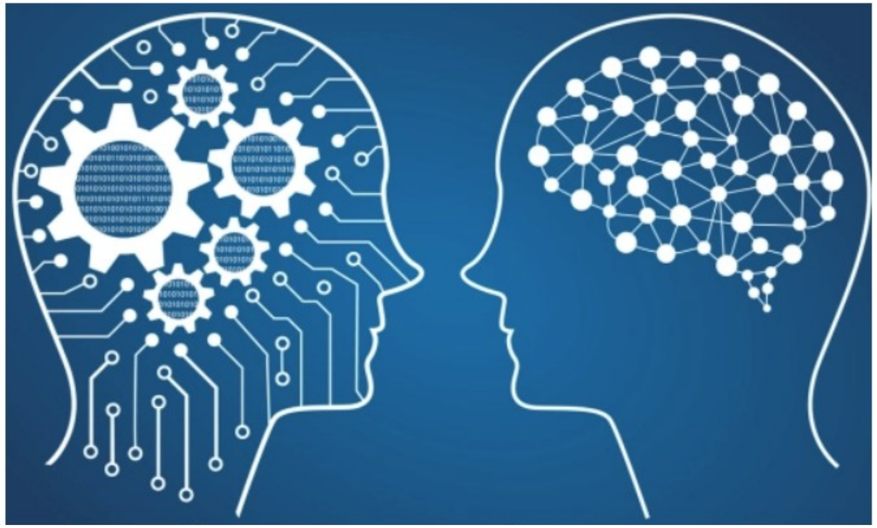


How to Become Intelligent?

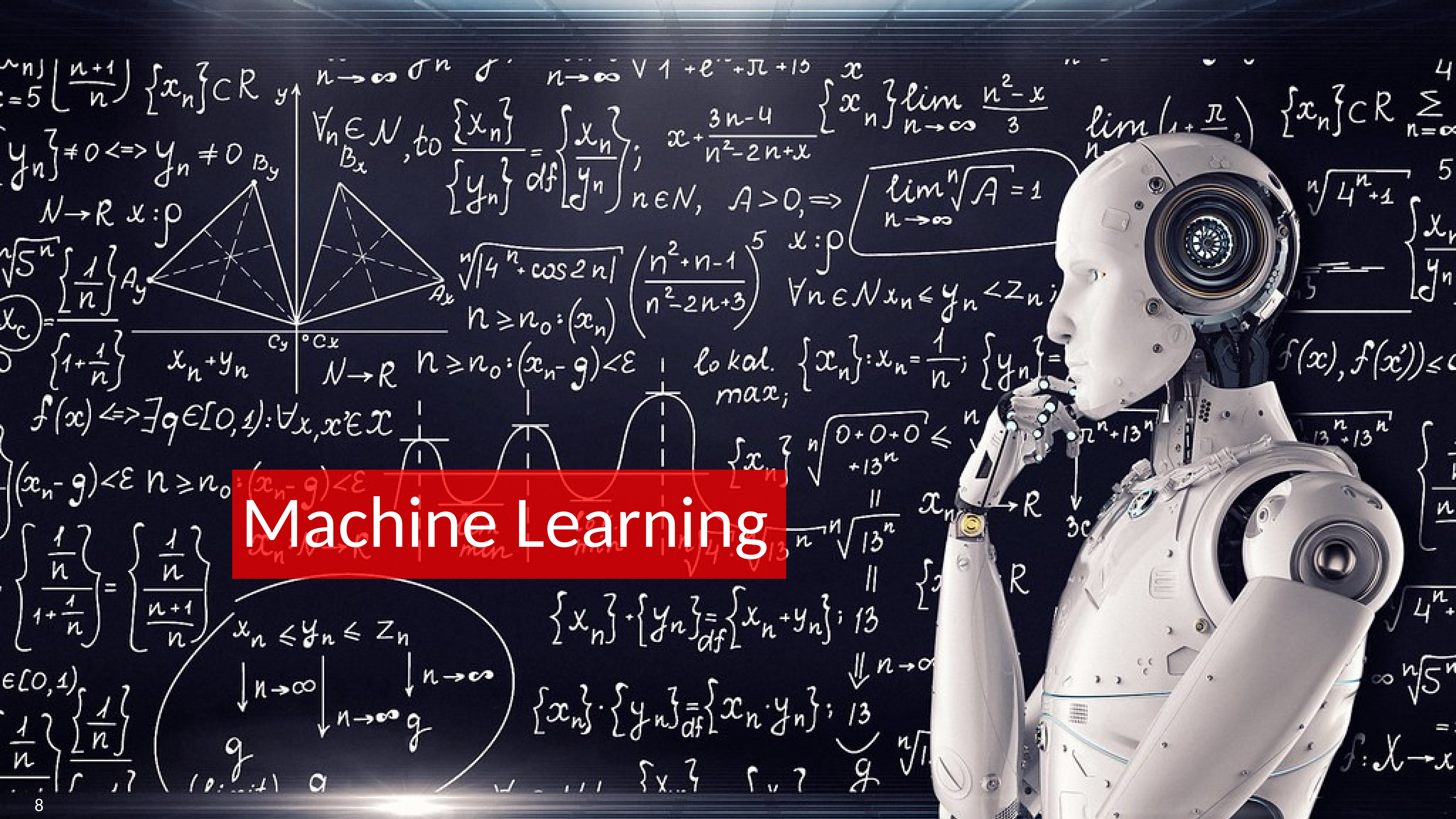
Solving Intelligence

How?

- Become intelligent through learning



Artificial Intelligence - Machine Learning

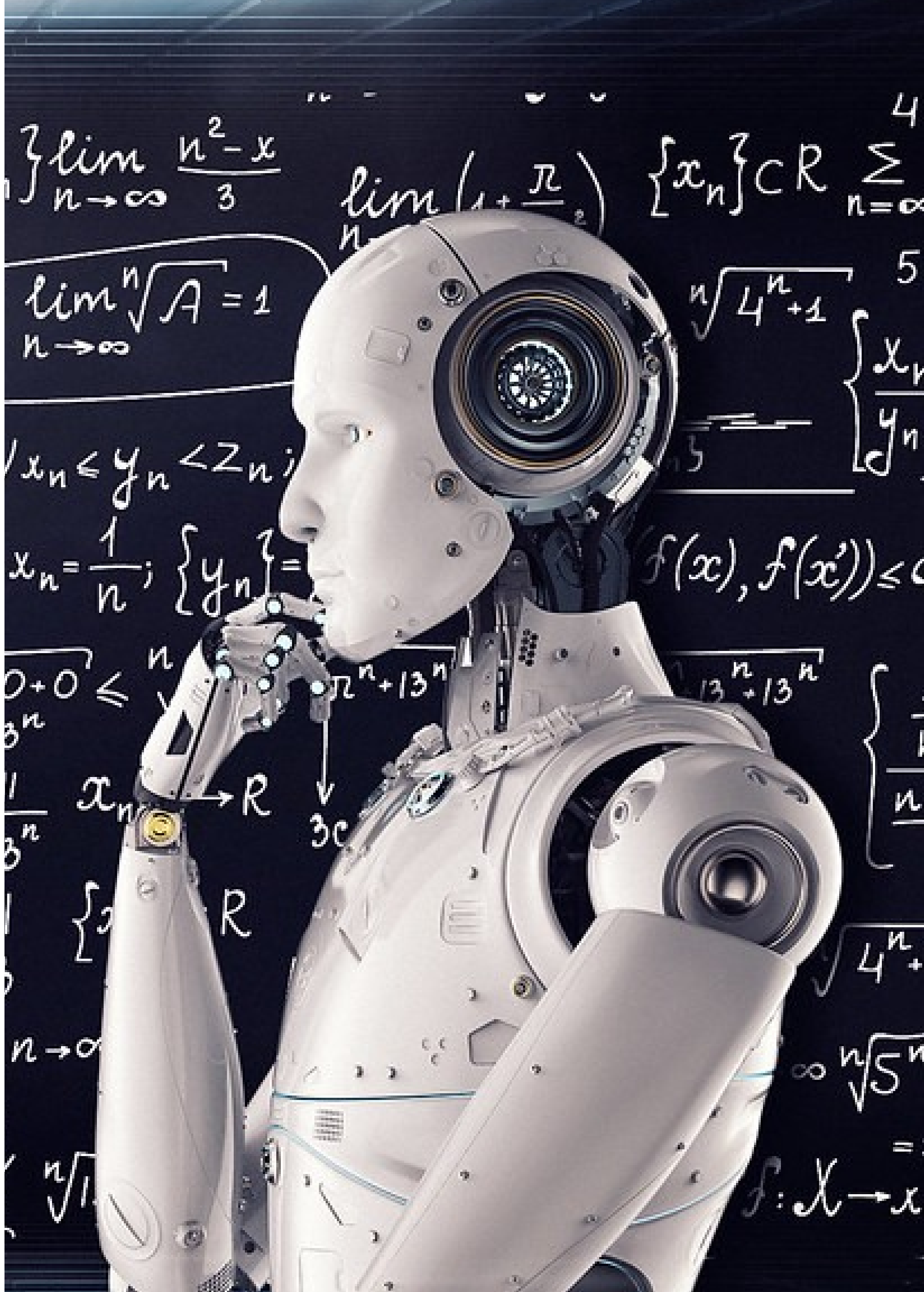


Machine Learning

Machine Learning

“Field of study that gives computers the ability to learn without being explicitly programmed”

Arthur Samuel (1959)
Professor at Stanford
Creator of the first self-learning program

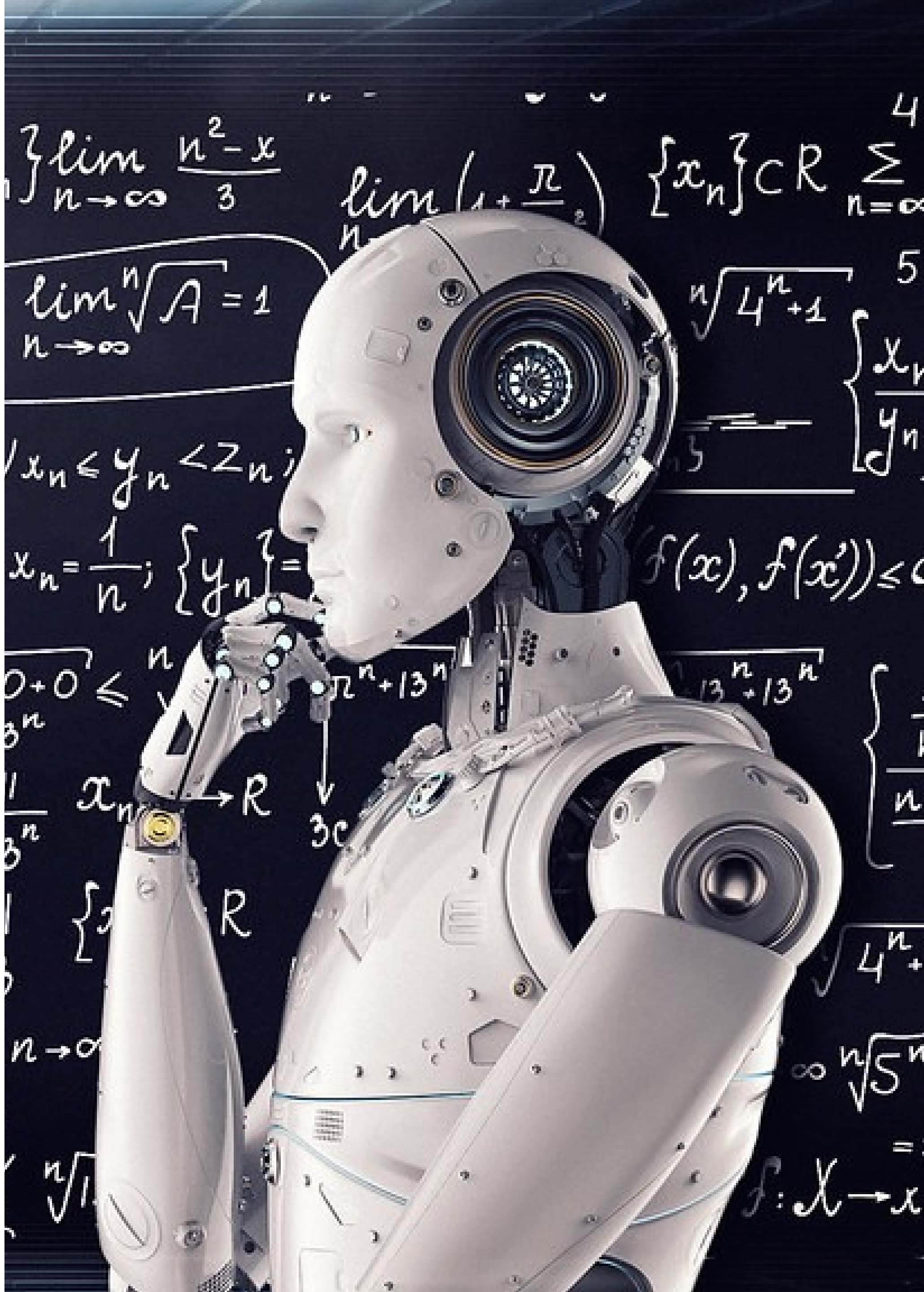


Machine Learning

“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.”

Tom Mitchell (1998)

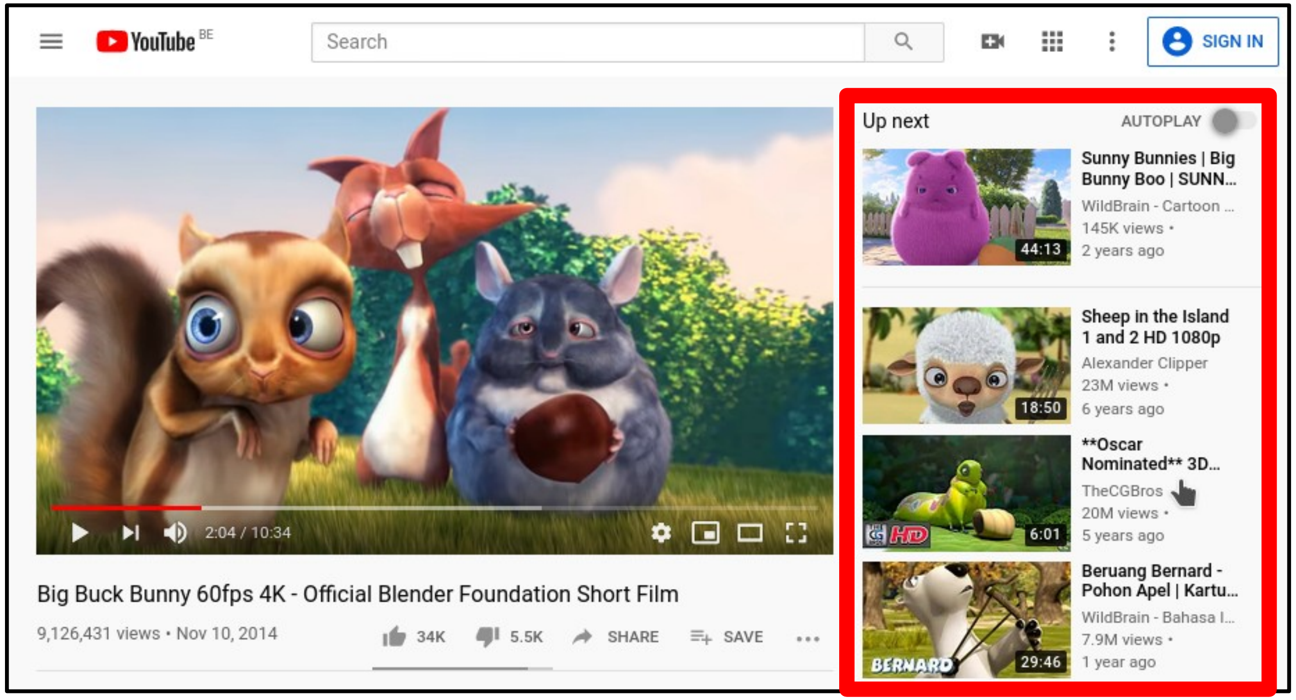
Professor at Carnegie Mellon University



Machine Learning

Exercise

- Given: the current setting where the "Up next" video has to be suggested

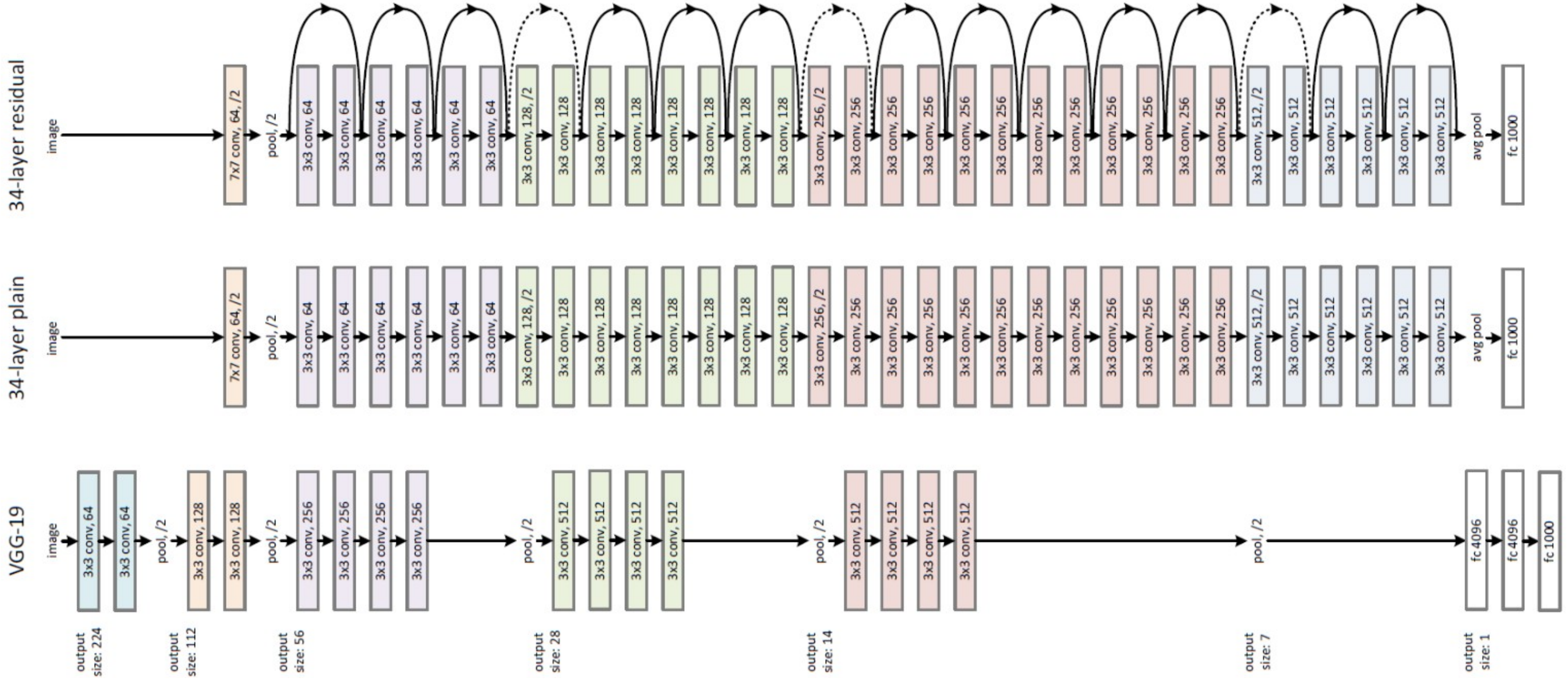


Indicate: Task (T), Experience (E), Metric (M)

Neural Networks

Artificial Neural Networks

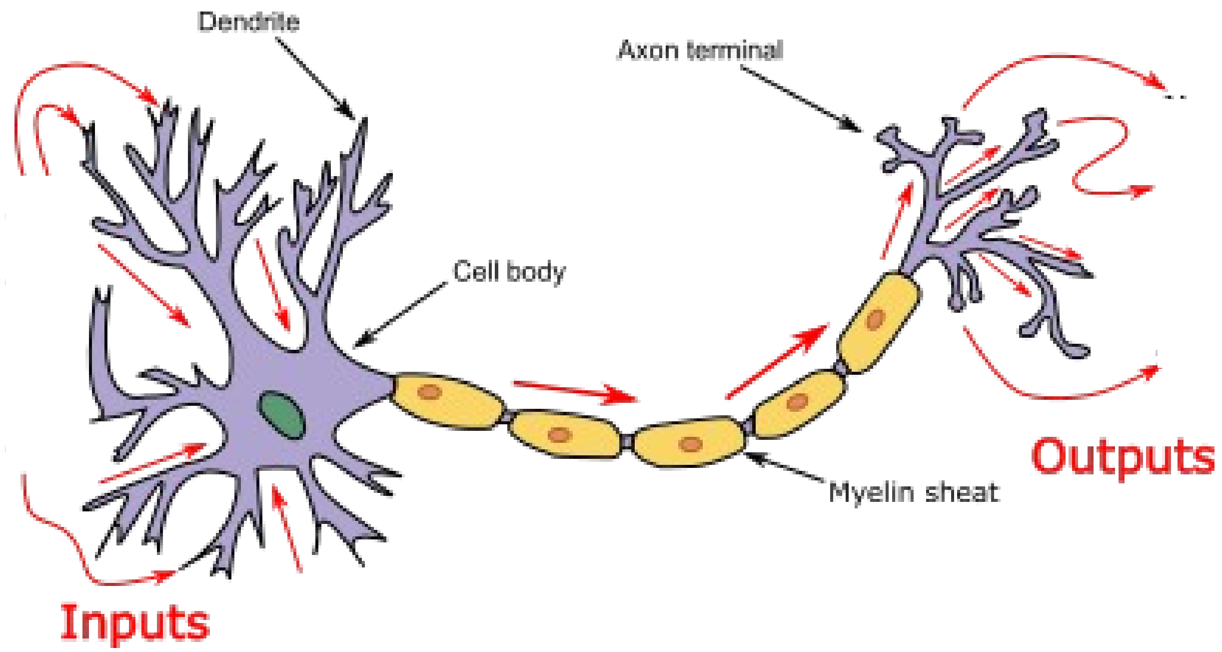
Let’s zoom in



Real Neurons

Note:

- Around 86k million in the human brain.

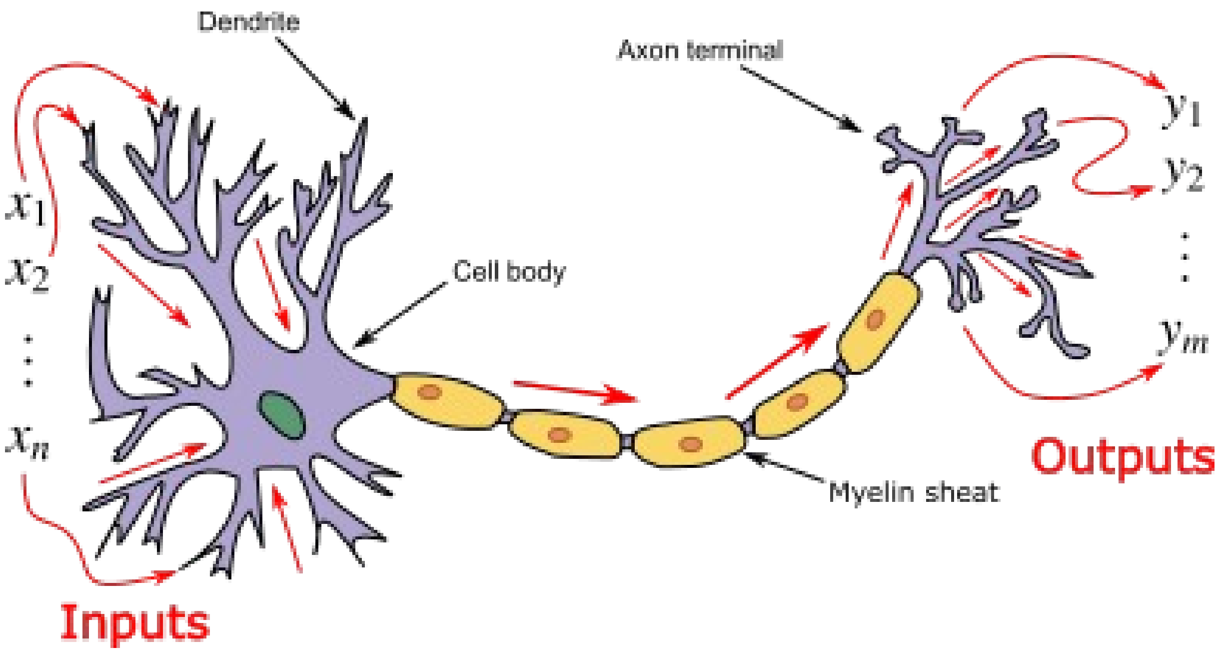


Characteristics:
- Basic computation
- Highly interconnected
- Has inhibition/excitation connections
- Possesses a state
- Outputs spikes

Real Neurons

Note:

- Around 86k million in the human brain.

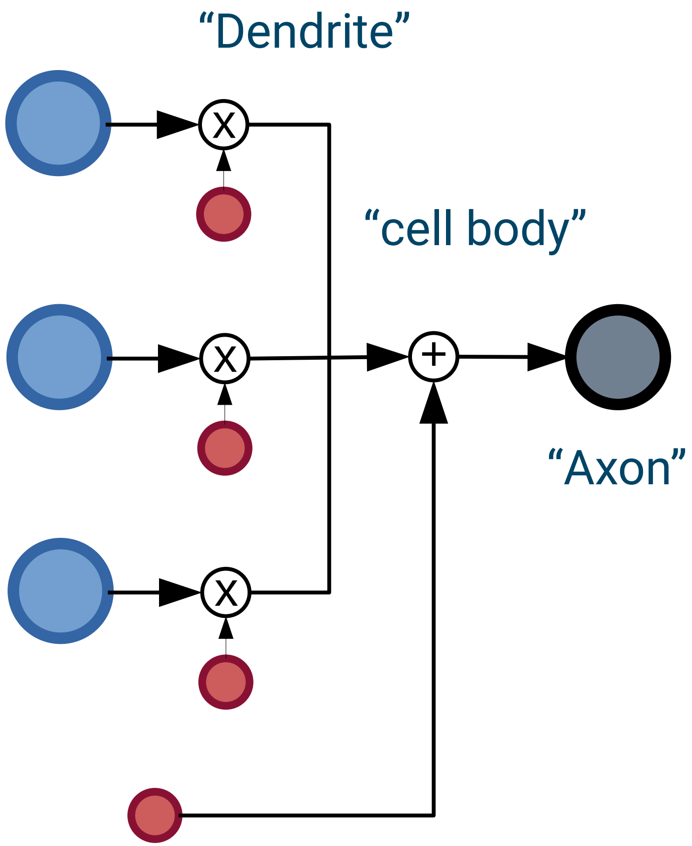


Characteristics:

- Basic computation
- Highly interconnected
- Has inhibition/excitation connections
- Possesses a state
- Outputs spikes

Artificial Neurons

How to build an artificial counterpart?



$$\sum_{i=1}^{d} w_i x_i + b$$

$$\sum_{i=0}^{d} w_i x_i, \quad x_0 := 1$$

Characteristics:

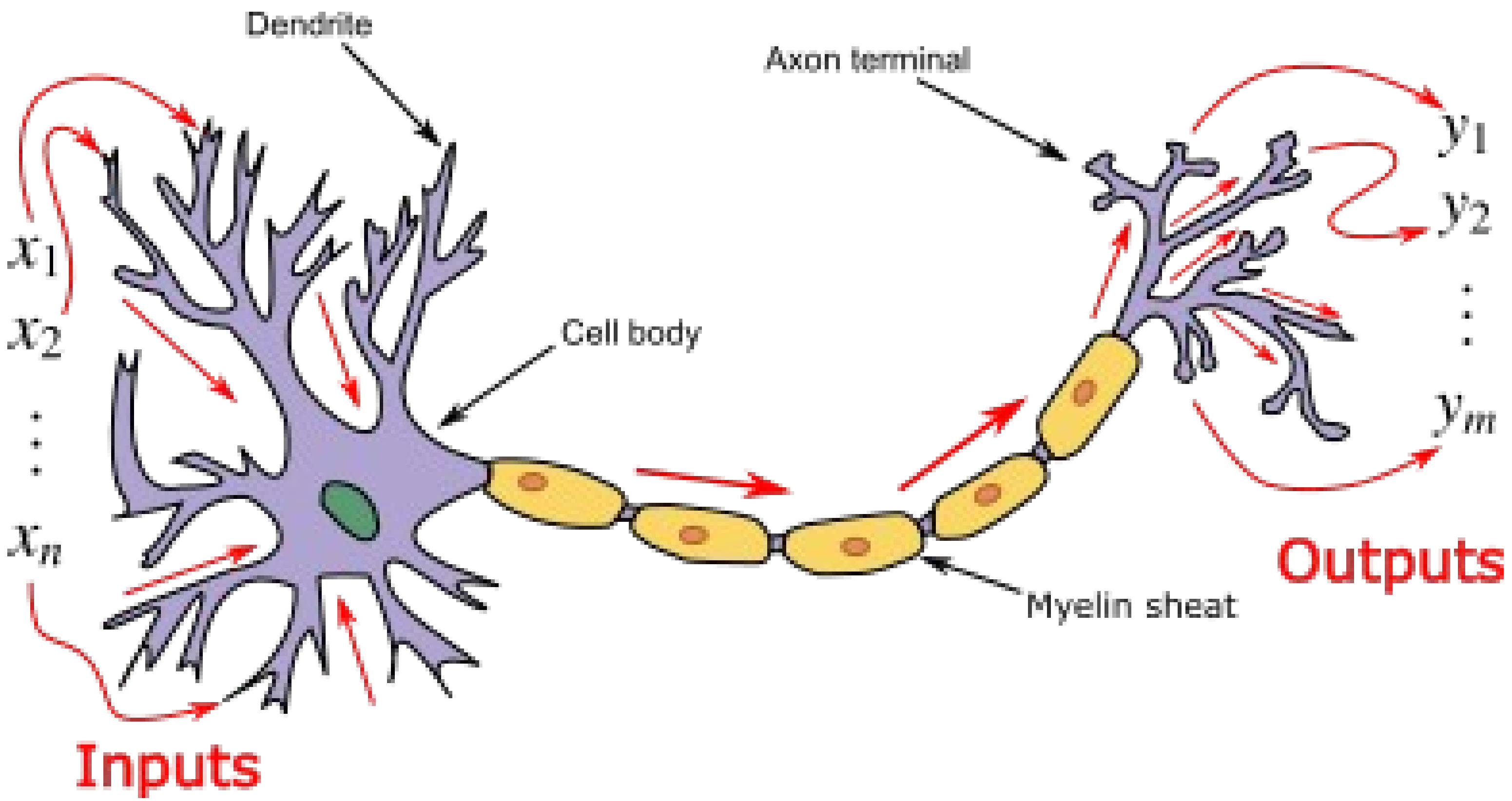
- Basic computation
- Has inhibition/excitation connections
- Building block
- Time-independent state
- Outputs real values

Interesting, but…

How to design an artificial neuron?

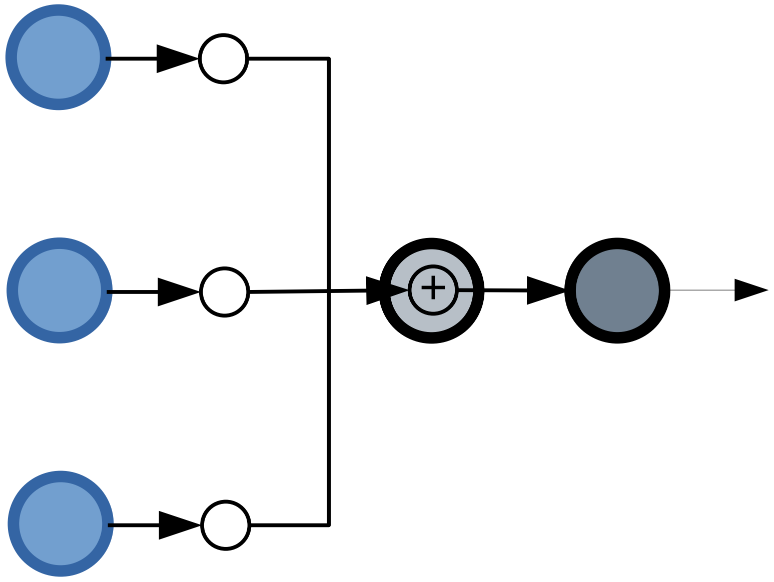
How to Design an Artificial Neuron?

Let’s define some notation…



1943: McCulloch-Pitts Neuron

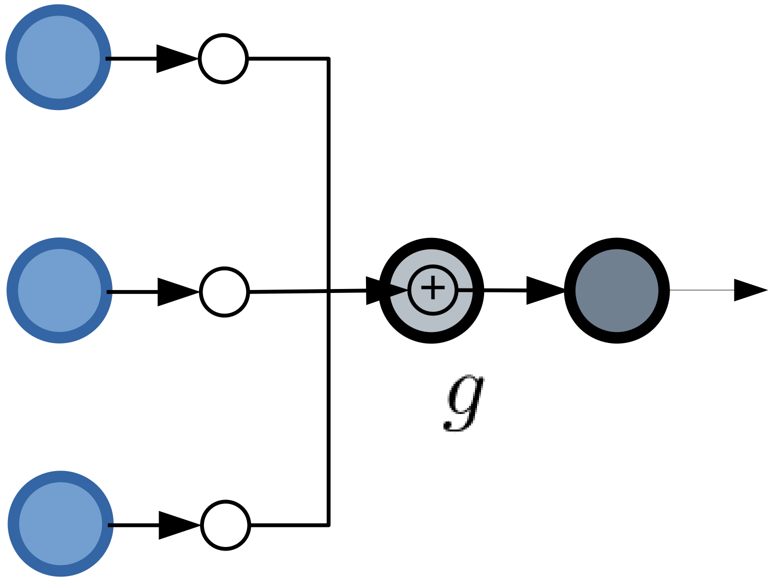
How to build an artificial counterpart?



[MuCulloch & Piitts, 1943]

1943: McCulloch-Pitts Neuron

How to build an artificial counterpart?

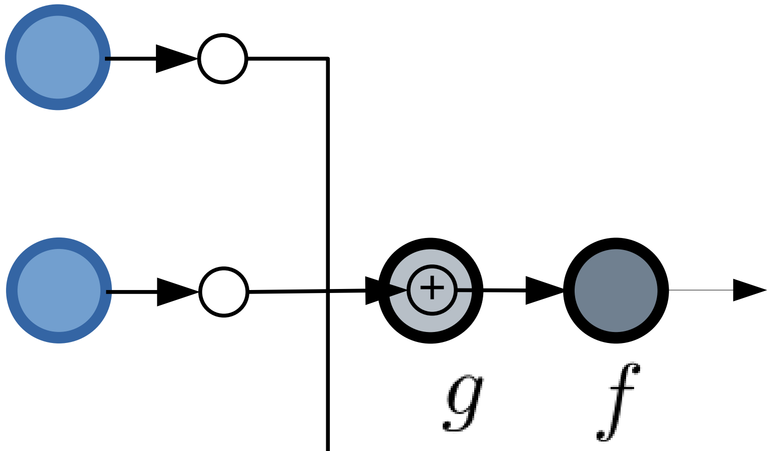


$$g(x_1, x_2, \ldots, x_n) = g(x) = \sum_{i=1}^{n} x_i$$

[MuCulloch & Piitts, 1943]

1943: McCulloch-Pitts Neuron

How to build an artificial counterpart?

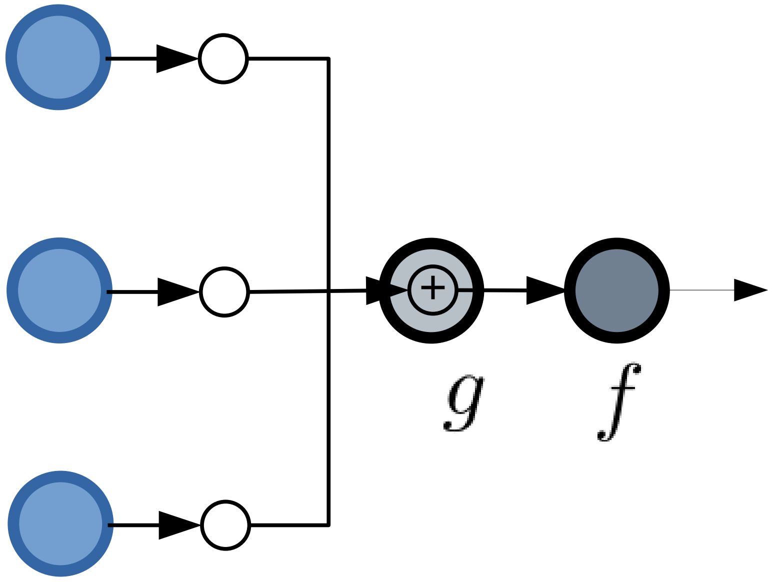


$$g(x_1, x_2, ..., x_n) = g(x) = \sum_{i=1}^{n} x_i$$

[MuCulloch & Piitts, 1943]

1943: McCulloch-Pitts Neuron

How to build an artificial counterpart?



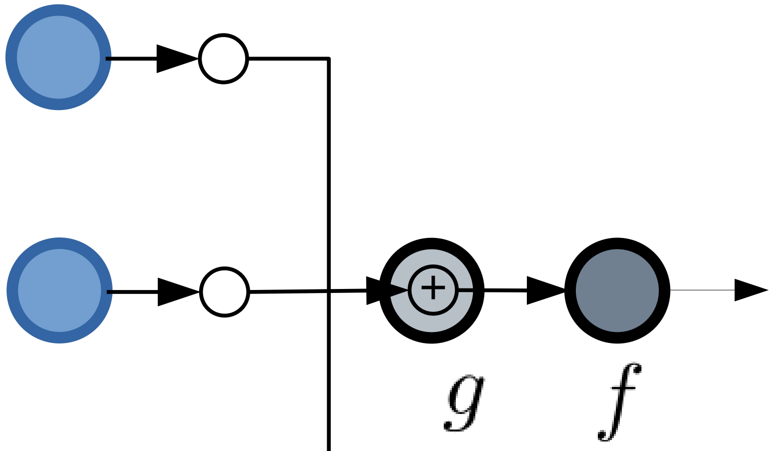
$$g(x_1, x_2, ..., x_n) = g(x) = \sum_{i=1}^{n} x_i$$

$$\begin{aligned} y = f(g(x)) &= 1 \quad if \quad g(x) \geq \theta \\ &= 0 \quad if \quad g(x) < \theta \end{aligned}$$

[MuCulloch & Piitts, 1943]

1943: McCulloch-Pitts Neuron

How to build an artificial counterpart?



[MuCulloch & Piitts, 1943]

$$g(x_1, x_2, ..., x_n) = g(x) = \sum_{i=1}^{n} x_i$$

$$\begin{aligned} y = f(g(x)) &= 1 \quad if \quad g(x) \geq \theta \\ &= 0 \quad if \quad g(x) < \theta \end{aligned}$$

Characteristics:

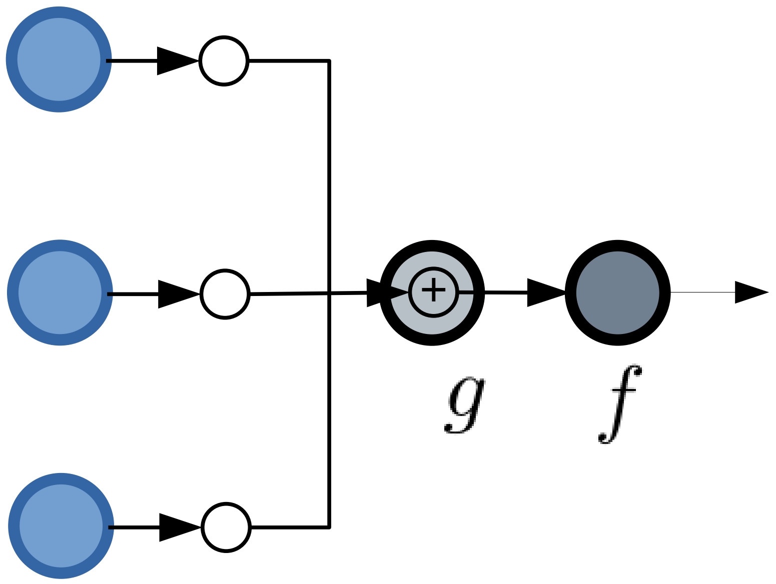
- Inputs are boolean values (inhibition/excitation)
- Two internal functions (*g* and *f*)
- Threshold parameter (*theta*)
- Output is binary

1943: McCulloch-Pitts Neuron

An example:

- Decide whether a movie should be recommended?

  *Given: **I like science fiction movies** and **horror movies** and **I do not like funny movies***

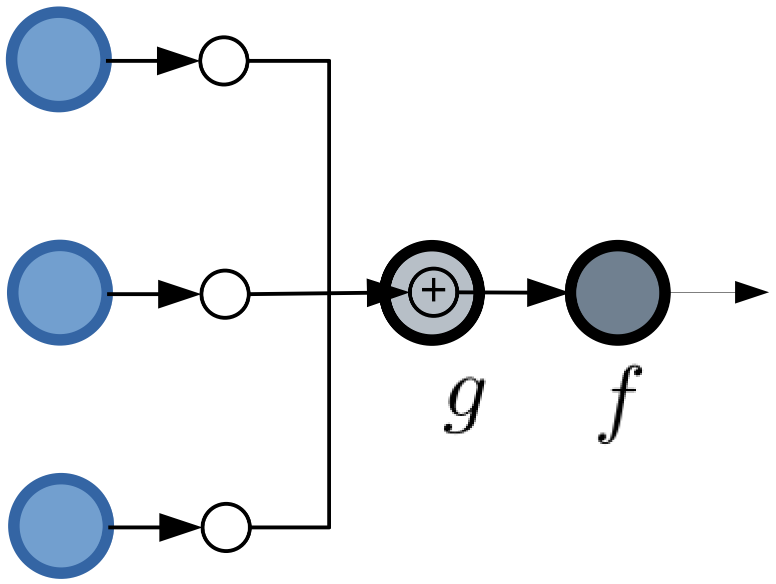


1943: McCulloch-Pitts Neuron

An example:

- Decide whether a movie should be recommended?

  *Given: **I like science fiction movies** and **horror movies** and **I do not like funny movies***

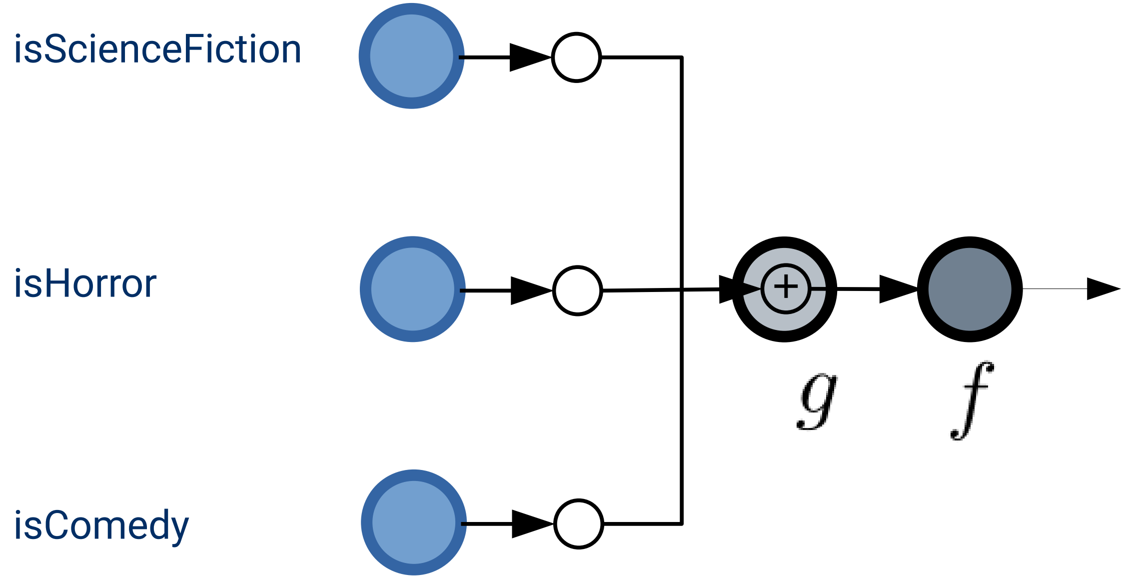


1943: McCulloch-Pitts Neuron

An example:

- Decide whether a movie should be recommended?

  Given: ***I like science fiction movies*** *and* ***horror movies*** *and* ***I do not like funny movies***

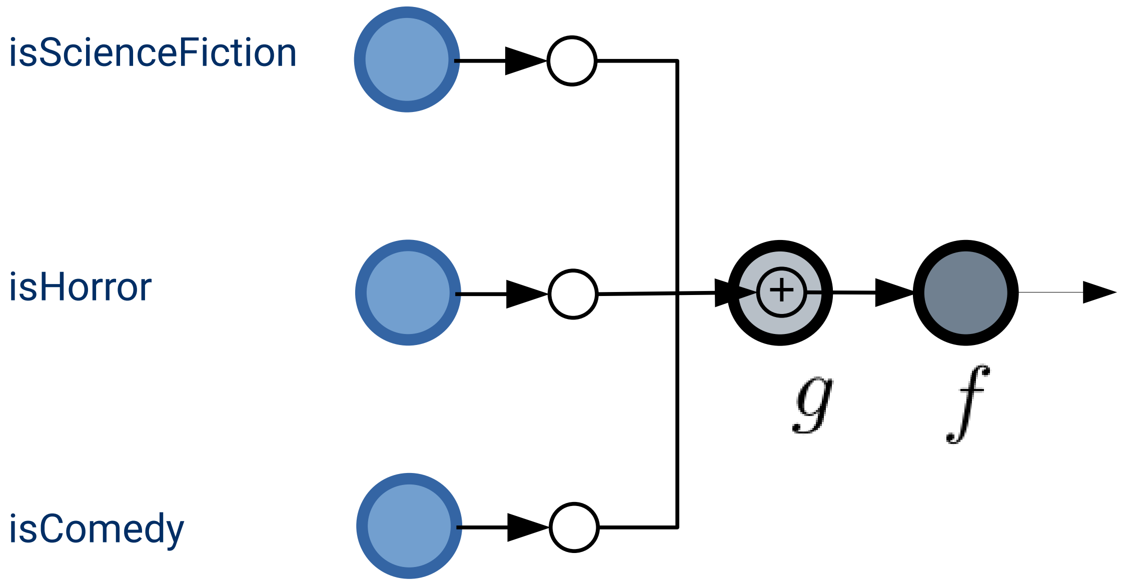


1943: McCulloch-Pitts Neuron

An example:

- Decide whether a movie should be recommended?

  *Given: **I like science fiction movies** and **horror movies** and **I do not like funny movies***

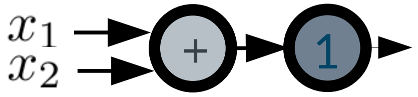


Inhibitory VS. Excitatory Inputs

1943: McCulloch-Pitts Neuron

An Example: **Logical Functions**

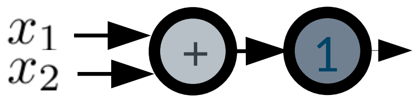
OR Function $(x_1 \vee x_2)$



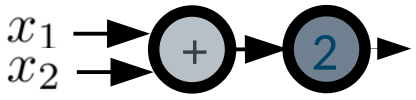
1943: McCulloch-Pitts Neuron

An Example: **Logical Functions**

OR Function $(x_1 \vee x_2)$



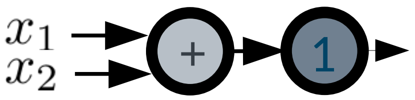
AND Function $(x_1 \wedge x_2)$



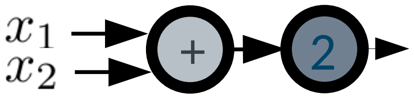
1943: McCulloch-Pitts Neuron

An Example: **Logical Functions**

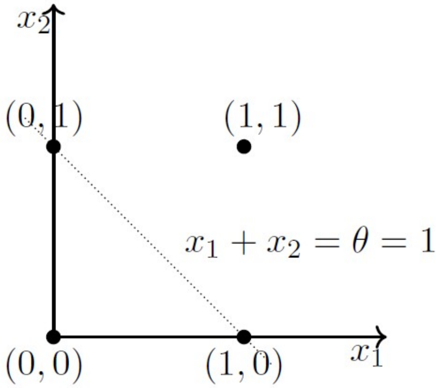
OR Function $(x_1 \vee x_2)$



AND Function $(x_1 \wedge x_2)$



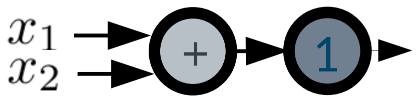
Graphically speaking



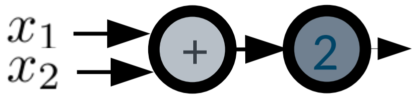
1943: McCulloch-Pitts Neuron

An Example: **Logical Functions**

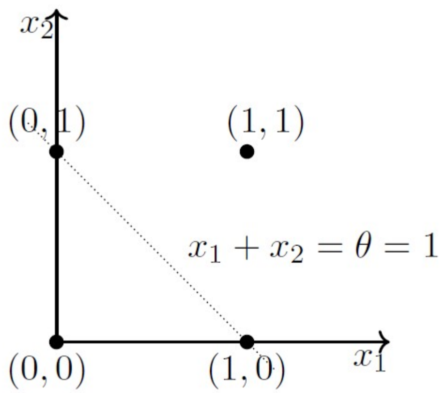
OR Function $(x_1 \vee x_2)$

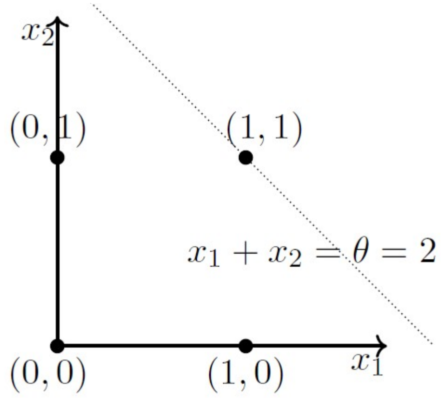


AND Function $(x_1 \wedge x_2)$



Graphically speaking

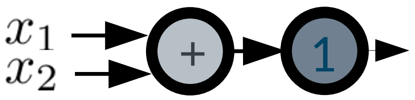




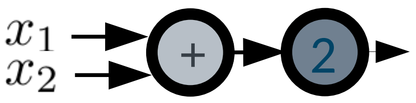
1943: McCulloch-Pitts Neuron

An Example: **Logical Functions**

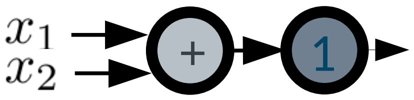
OR Function $(x_1 \vee x_2)$



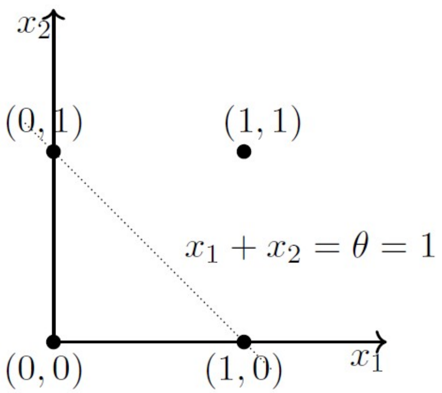
AND Function $(x_1 \wedge x_2)$

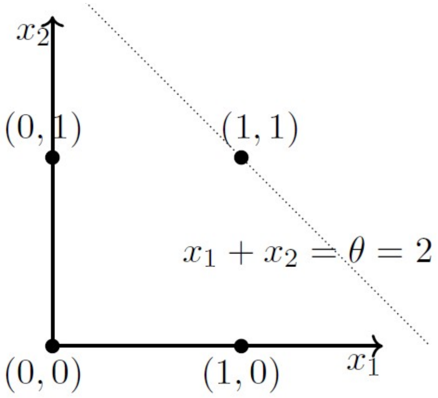


Inhibitory Input $(x_1 \wedge \neg x_2)$



Graphically speaking

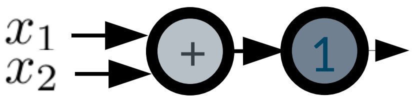




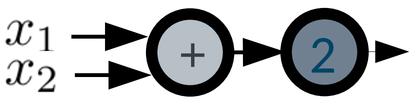
1943: McCulloch-Pitts Neuron

An Example: **Logical Functions**

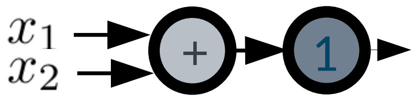
OR Function $(x_1 \vee x_2)$



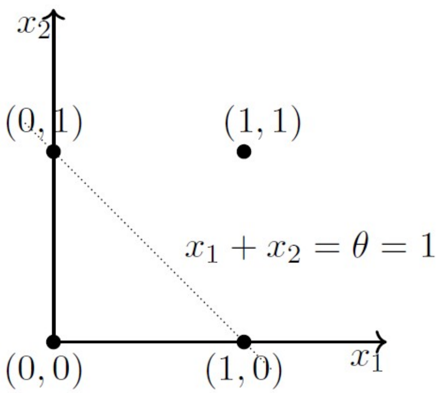
AND Function $(x_1 \wedge x_2)$

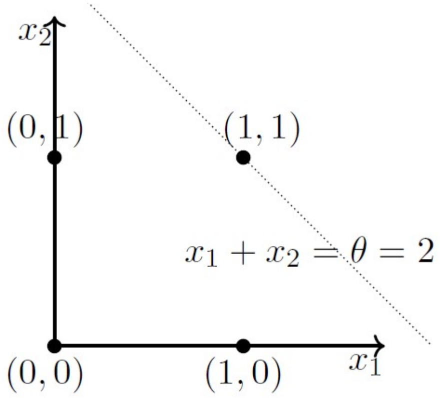


Inhibitory Input $(x_1 \wedge \neg x_2)$



Graphically speaking

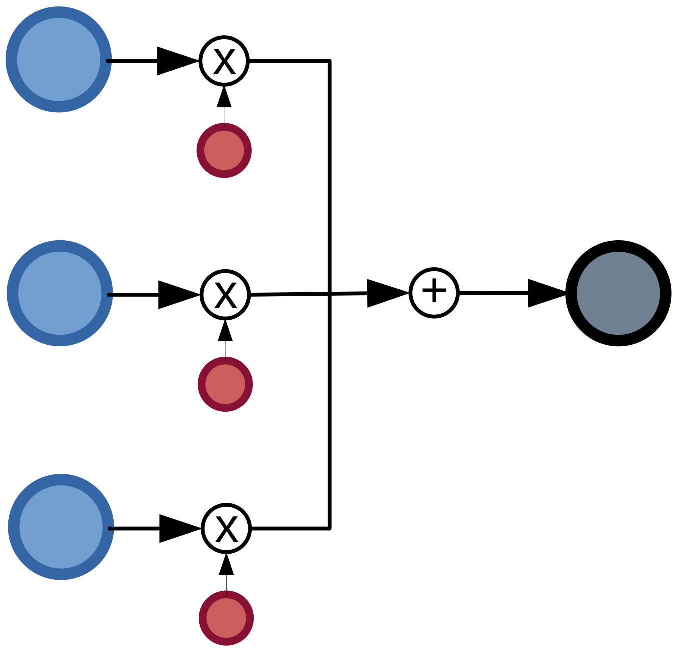




Limitations?
- What about real inputs
- Threshold is pre-defined
- Inputs have same relevance
- What about functions that are not linearly separable?

1969: Perceptron Model

What was new?



$$y = 1 \quad if \sum_{i=1}^{d} w_i x_i \geq \theta$$
$$= 0 \quad if \sum_{i=1}^{d} w_i x_i < \theta$$

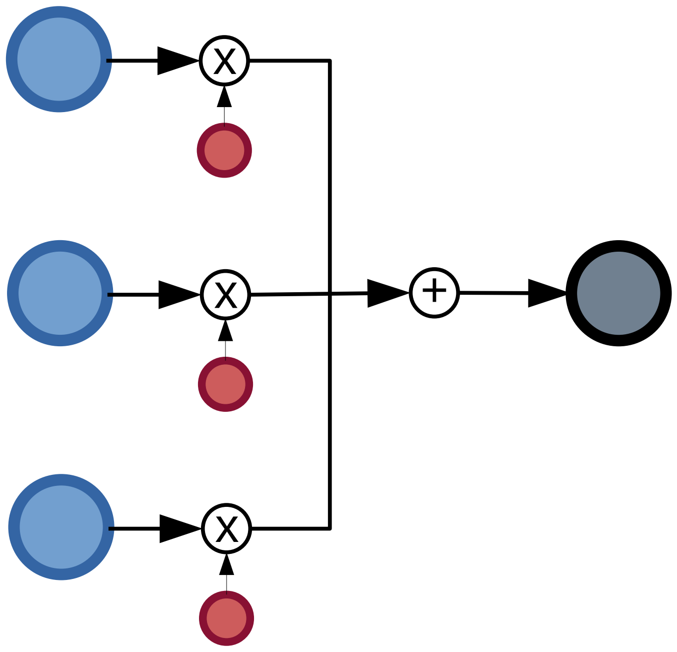
Extensions:

- Weights (w) indicate relevance
- Real inputs are supported

[Rossenblat, 1958]
[Minsky & Papert, 1969]

1969: Perceptron Model

What was new?



$$y = 1 \quad if \sum_{i=1}^{d} w_i x_i - \theta \geq 0$$
$$= 0 \quad if \sum_{i=1}^{d} w_i x_i - \theta < 0$$

Extensions:

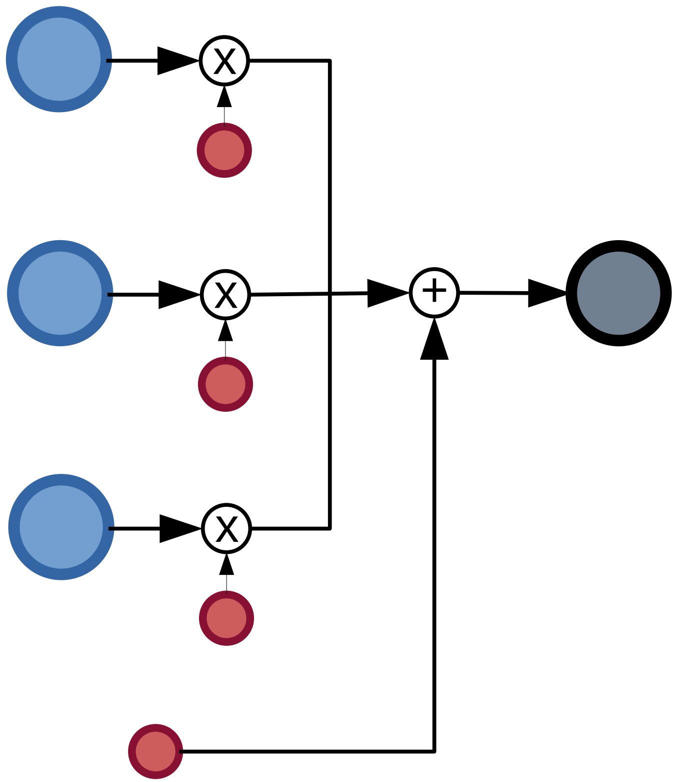
- Weights (w) indicate relevance
- Real inputs are supported

[Rossenblat, 1958]
[Minsky & Papert, 1969]

University of Antwerp

1969: Perceptron Model

What was new?



[Rossenblat, 1958]
[Minsky & Papert, 1969]

$$y = 1 \quad if \sum_{i=0}^{d} w_i x_i \geq 0$$

$$= 0 \quad if \sum_{i=0}^{d} w_i x_i < 0$$

$$x_0 = 1, w_o = -\theta$$

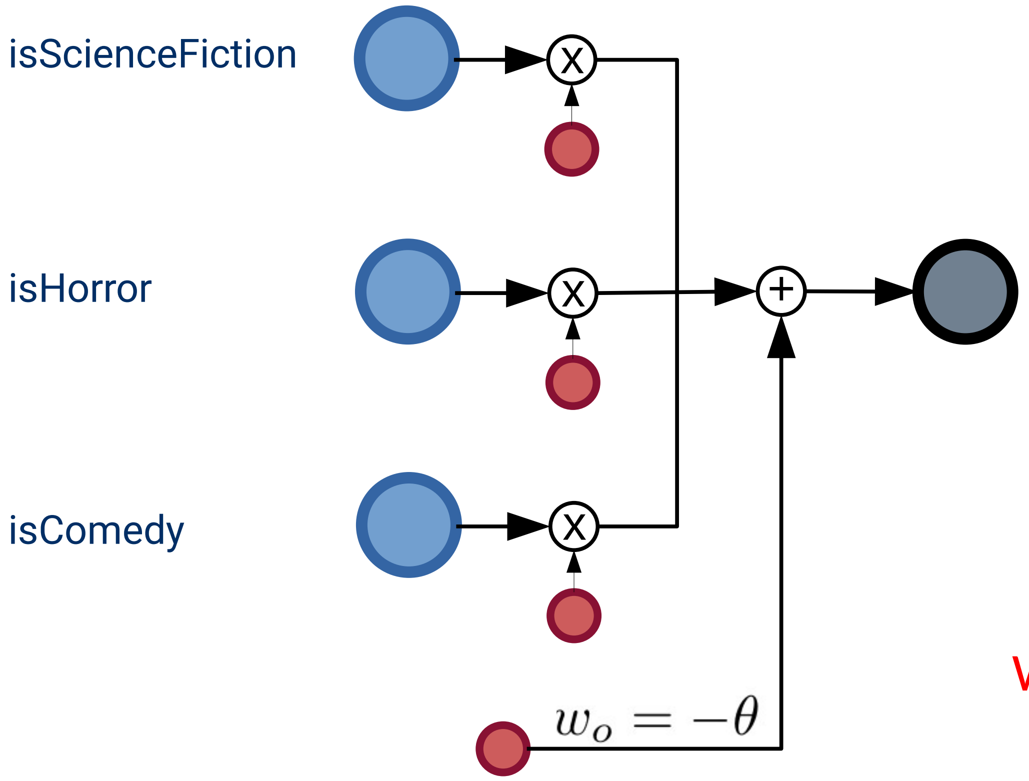
Extensions:

- Weights (w) indicate relevance
- Real inputs are supported
- Threshold (bias) is learnable

1969: Perceptron Model

Revisiting the Previous Example:

- Decide whether a movie should be recommended?



W_0 encodes Prior knowledge/Bias

1969: Perceptron Model

How do we learn the weights?

Algorithm: Perceptron Learning Algorithm

$P \leftarrow inputs \quad with \quad label \quad 1;$
$N \leftarrow inputs \quad with \quad label \quad 0;$
Initialize $\mathbf{w}$ randomly;
while $!convergence$ **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P \quad and \quad \mathbf{w}.\mathbf{x} < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
  end
  if $\mathbf{x} \in N \quad and \quad \mathbf{w}.\mathbf{x} \geq 0$ **then**
    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
  end
end
//the algorithm converges when all the inputs are classified correctly

Extensions:
- Weights are initialized randomly
- Iterate over the training data
- Convergence proof

1969: Perceptron Model

How do we learn the weights?

Algorithm: Perceptron Learning Algorithm

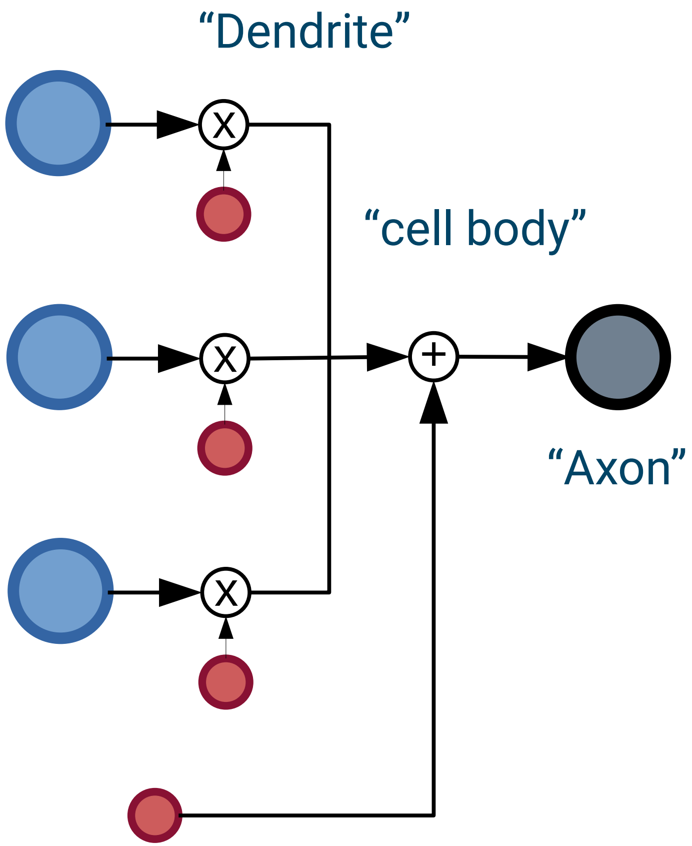
$P \leftarrow inputs \quad with \quad label \quad 1;$
$N \leftarrow inputs \quad with \quad label \quad 0;$
Initialize $\mathbf{w}$ randomly;
while $!convergence$ **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P \quad and \quad \mathbf{w}.\mathbf{x} < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
  end
  if $\mathbf{x} \in N \quad and \quad \mathbf{w}.\mathbf{x} \geq 0$ **then**
    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
  end
end
//the algorithm converges when all the inputs are classified correctly

Extensions:
- Weights are initialized randomly
- Iterate over the training data
- Convergence proof

Q: What about the task(T), experience(E) and Metric(M) ?

Artificial Neurons

How to build an artificial counterpart?



$$\sum_{i=1}^{d} w_i x_i + b$$

$$\sum_{i=0}^{d} w_i x_i, \quad x_0 := 1$$

Characteristics:
- Basic computation
- Has inhibition/excitation connections
- Building block
- Time-independent state
- Outputs real values

Summarizing

[Finally :D]

Summarizing

- **A Work in Progress**
  - [Deep] Artificial Neural Networks is continuously moving field.
  - Terminology can be confusing

Summarizing

- **A Work in Progress**
  - [Deep] Artificial Neural Networks is continuously moving field.
  - Terminology can be confusing

- **Neurons (perceptrons) are the most granular elements**
  - Neurons → Layers → Networks

Summarizing

- **A Work in Progress**
  - [Deep] Artificial Neural Networks is continuously moving field.
  - Terminology can be confusing
- **Neurons (perceptrons) are the most granular elements**
  - Neurons → Layers → Networks
- **Initially Heuristic → Now Learnable.**

Pay Attention to...

- Relevant factors when defining a learning problem
- Evolution of artificial neurons designs / motivations
  - Strengths and weaknesses
- Difference w.r.t. natural neurons.

References

- ## Evolution of Perceptron Design
  - **McCulloch & Pitts (1943) A Logical Calculus of the Ideas Immanet In Nervous Activity***
    https://www.cs.cmu.edu/~./epxing/Class/10715/reading/McCulloch.and.Pitts.pdf
  - **Minsky & Papert (1987) Perceptrons: An Introduction to Computational Geometry**
    https://1lib.eu/book/2777515/cf7392?regionChanged=&redirect=161050537
- ## Perceptron Learning Algorithm
  - **Rossenblatt (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain**
    http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf
  - **Collins - Convergence Proof for the Perceptron Algorithm**
    http://www.cs.columbia.edu/~mcollins/courses/6998-2012/notes/perc.converge.pdf

Questions?

Artificial Neural Networks

[A Classical Perspective]

José Oramas

University of Antwerp