# Neural Networks and Categorization

Ecole Centrale de Nantes

october-november 2011

Pierre Andry
Université Cergy-Pontoise
andry@ensea.fr
ETIS UMR CNRS 8051

# Global introduction

Traditional opposition : *classical IA / connectionism*

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- Logics (inference, deduction, etc... )
- Reasoning
- Symbol manipulation
- Solution discovery
- Rules (rules based system, expert systems)
- Advanced Algorithmics
- State space search (depth first, A*, ...)

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- Logics (inference, deduction, etc... )
- Reasoning
  - LISP [MIT,McCarthy,58]
  - Prolog [Colmerauer,60]
- Symbol manipulation
- Solution discovery
- Rules (rules based system, expert systems)
- Advanced Algorithmics
- State space search (depth first, A*, ...)

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- Logics (inference, deduction, etc... )
- Reasoning
- Symbol manipulation
  - “computers can be used to manipulate symbols” [McCarthy 60]
  - “dialog with a computer ELIZA [Weiezbaum, 60]
- Solution discovery
- Rules (rules based system, expert systems)
- Advanced Algorithmics
- State space search (depth first, A*, ...)

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- Logics (inference, deduction, etc... )
- Reasoning
- Symbol manipulation
- Solution discovery
- Rules (rules based system, expert systems)
- Advanced Algorithmics
- State space search (depth first, A*, ...)

- 80's
- Scripts language
- PLANNER : goal generation for program solving
- GPS [Newel & Simon, 78]

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- Logics (inference, deduction, etc... )
- Reasoning
- Symbol manipulation
- Solution discovery
- Rules (rules based system, expert systems)
- Advanced Algorithmics
- State space search (depth first, A*, ...)
    - A star [Hart & Al. 68]

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

Achievements ?

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

Achievement :

In 1997, Deep Blue super computer (IBM) beat world chess champion Gary Kasparov

... and Chess is *one* expression of intelligence (memory, symbol manipulation, adaptation...)

# Global introduction

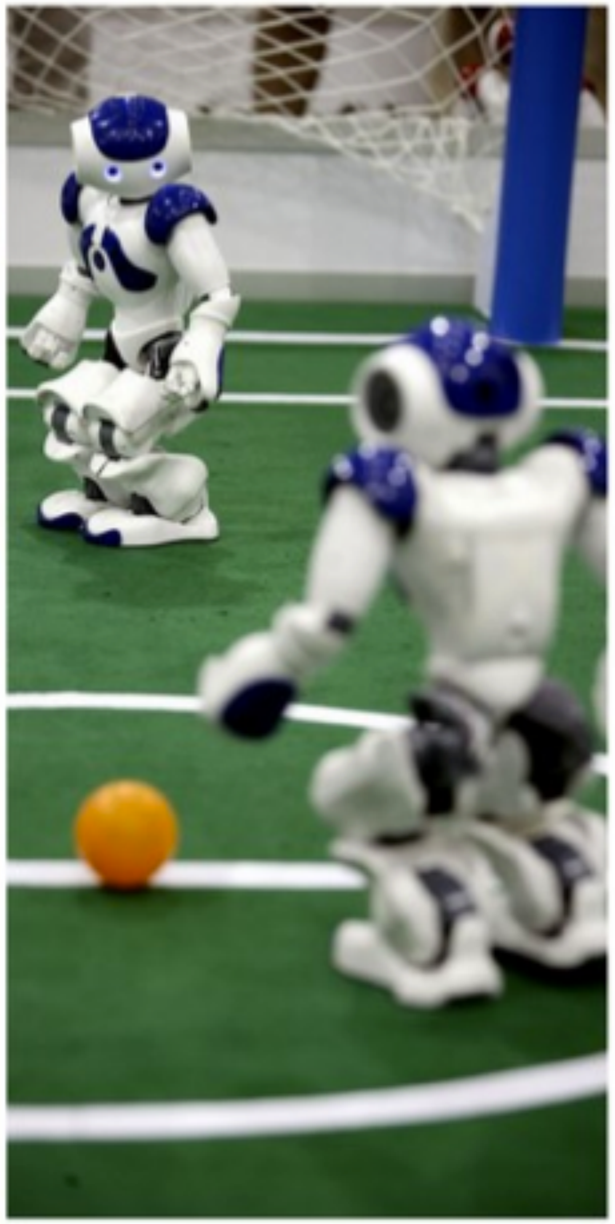
Traditional opposition : *classical IA* / *connectionism*

...but

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

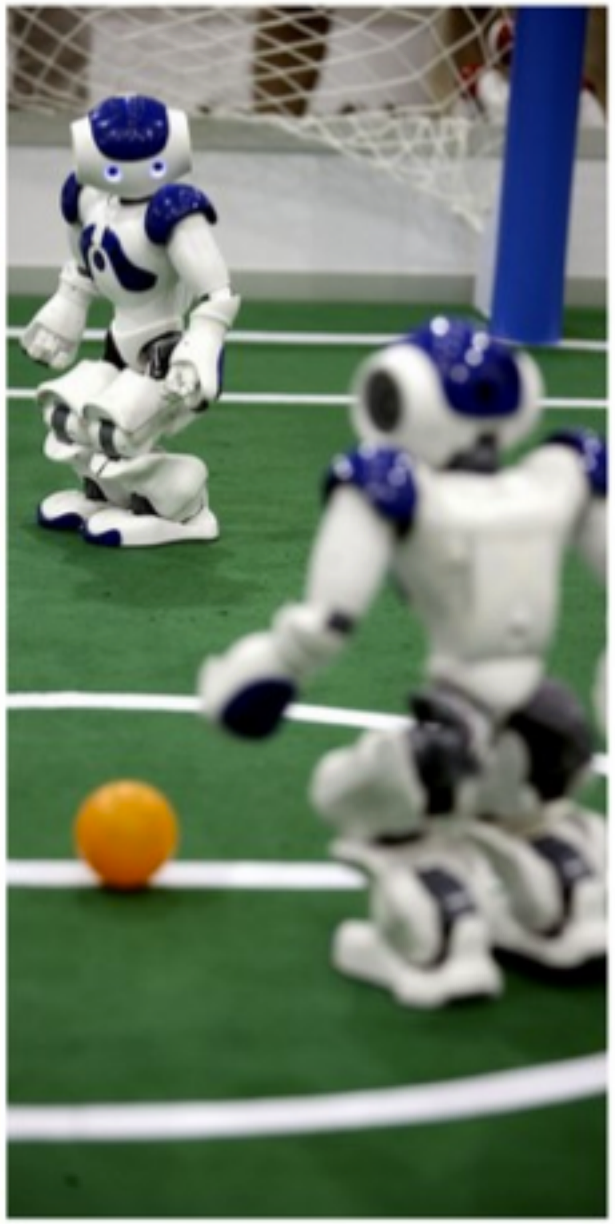
...but



Nao humanoid robot is still not able to compete with a 3 years-old child...

# Global introduction

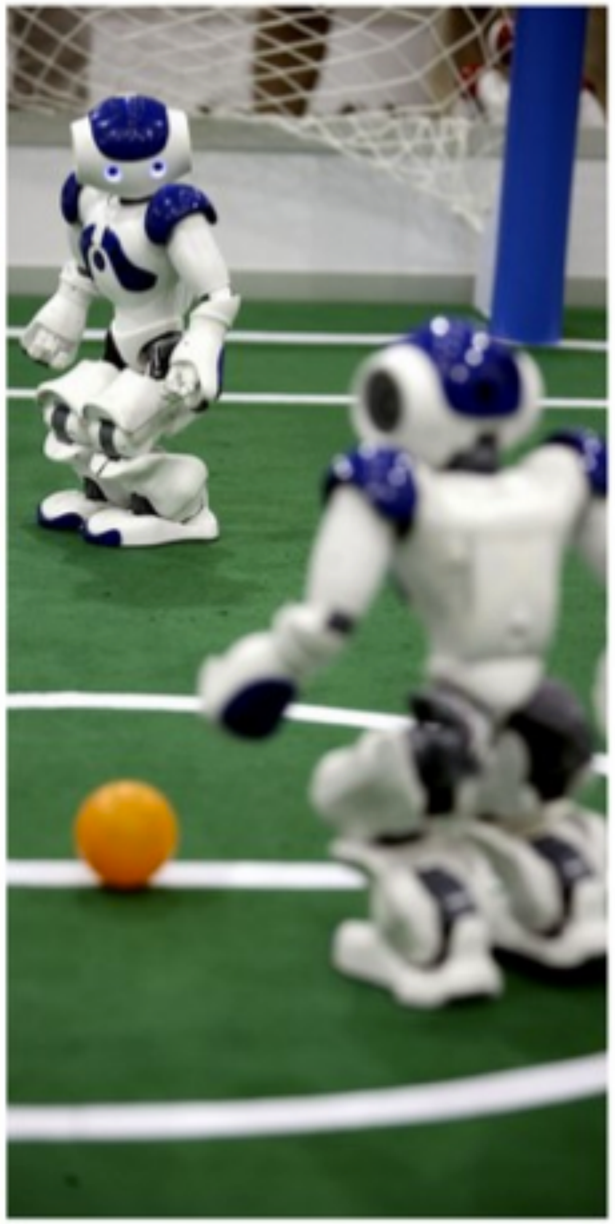
Traditional opposition : *classical IA* / *connectionism*



- we are able to beat the chess world champion....
- ...but not to play elementary ball game such as football
- Why ?

# Global introduction

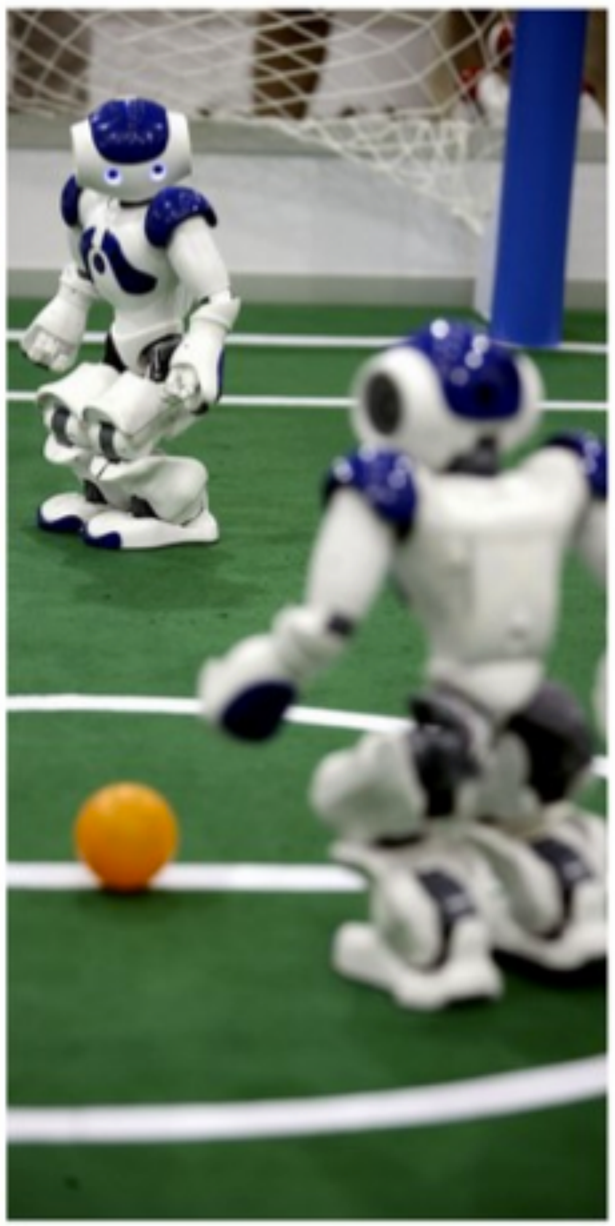
Traditional opposition : *classical IA* / *connectionism*



- perception
- action
- sensorimotor coordination
- memory
- fast decision making
- adaptation (“on line”)
- prediction / anticipation
- theory of mind ? strategy ?
- collective ?

# Global introduction

Traditional opposition : *classical IA* / *connectionism*



- perception
- action
- sensorimotor coordination
- memory
- fast decision making
- adaptation (“on line”)
- prediction / anticipation
- theory of mind ? strategy ?
- collective ?

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- perception
- action
- sensorimotor coordination
- memory
- fast decision making
- adaptation ("on line")
- prediction / anticipation
- theory of mind ? strategy ?
- collective ?

All simultaneously...

...with one brain

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- We are successful when the system (a computer) manipulates symbols...
- when the states are well *defined*
- *well identified*
- when the concepts are correctly *framed*
- when the world is easily *segmented*
- *Chess is perfect for that*

# Global introduction

Traditional opposition : *classical IA* / *connectionism*


- We are successful when the system (a computer) manipulates symbols...
- when the states are well *defined*
- *well identified*
- when the concepts are correctly *framed, formalised*
- when the world is easily *segmented*
- *Chess is perfect for that*

# Global introduction

Traditional opposition : *classical IA* / *connectionism*


A. de Rengervé, J. Hirel, P Andry, M. Quoy, P. Gaussier -ETIS-Cergy [BioRob2011]

- We fail when the system is immersed in the real world...
- always changing
- unpredictable
- things, peoples, objects, concepts are undefined
- noisy
- *A nightmare for Intelligent robotics*

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- To bypass that these issues, we need :
  - *to let the system adapt*
  - *to let the system learn*
  - *build its own categories*
  - *to exploit the perception-action dynamics*

# Global introduction

Traditional opposition : *classical IA* / *connectionism*

- To bypass that these issues, we need :
  - *to let the system adapt*
  - *to let the system learn*
  - *build its own categories*
  - *to exploit the perception-action dynamics*

The brain does it well : connectionism

# Global introduction

One more example : let's try to define an object : a chair

- Let's do it classical :
  - *object (legs, 4) and object(back,1) and object (seat,1)*

# Global introduction

One more example : let's try to define an object : a chair

- Let's do it classical :
  - *(object (legs, 4) or (object (legs,3)) and object(back,1) and object (seat,1)...*

# Global introduction

One more example : let's try to define an object : a chair

- Let's do it classical :
    - *(object (legs, 4) or (object (legs,3)) ~~and object(back,1)~~ and object (seat,1)...*

# Global introduction

One more example : let's try to define an object : a chair

- Let's do it classical :
  - *(object (legs, 4) or (object (legs,3)) ~~and object(back,1)~~ and object (seat,1)...*
  - *well...*

# Global introduction

One more example : let's try to define an object : a chair

- Let's do it classical :
  - *(object (legs, 4) or (object (legs,3)) ~~and object(back,1)~~ and object (seat,1)...*
  - *well...*
  - *..or (object, legs, 1) ?*

# Global introduction

One more example : let's try to define an object : a chair

(one of the world most known chair : Eames rocking chair)

- Let's do it classical :
  - *(object (legs, 4) or (object (legs,3)) ~~and object(back,1)~~ and object (seat,1)...*
  - *well...*
  - *..or (object, legs, 1) ?*
  - *..or 2 ?*

# Global introduction

One more example : let's try to define an object : a chair

- Let's do it classical :
  - *(object (legs, 4) or (object (legs,3)) ~~and object(back,1)~~ and object (seat,1)...*
  - *well...*
  - *..or (object, legs, 1) ?*
  - *..or 2 ? .....?*

# Global introduction

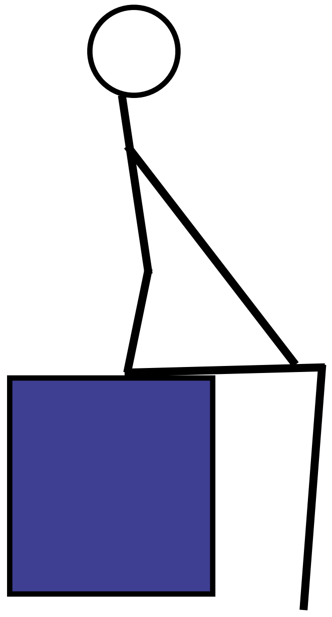
One more example : let's try to define an object : a chair

- Let's do it classical :
  - ~~*(object (legs, 4) or (object (legs,3)) and object(back,1) and object (seat,1)...*~~
  - ~~*well...*~~
  - ~~*..or (object, legs, 1) ?*~~
  - ~~*..or 2 ? .....?*~~
  - *....?*

# Global introduction

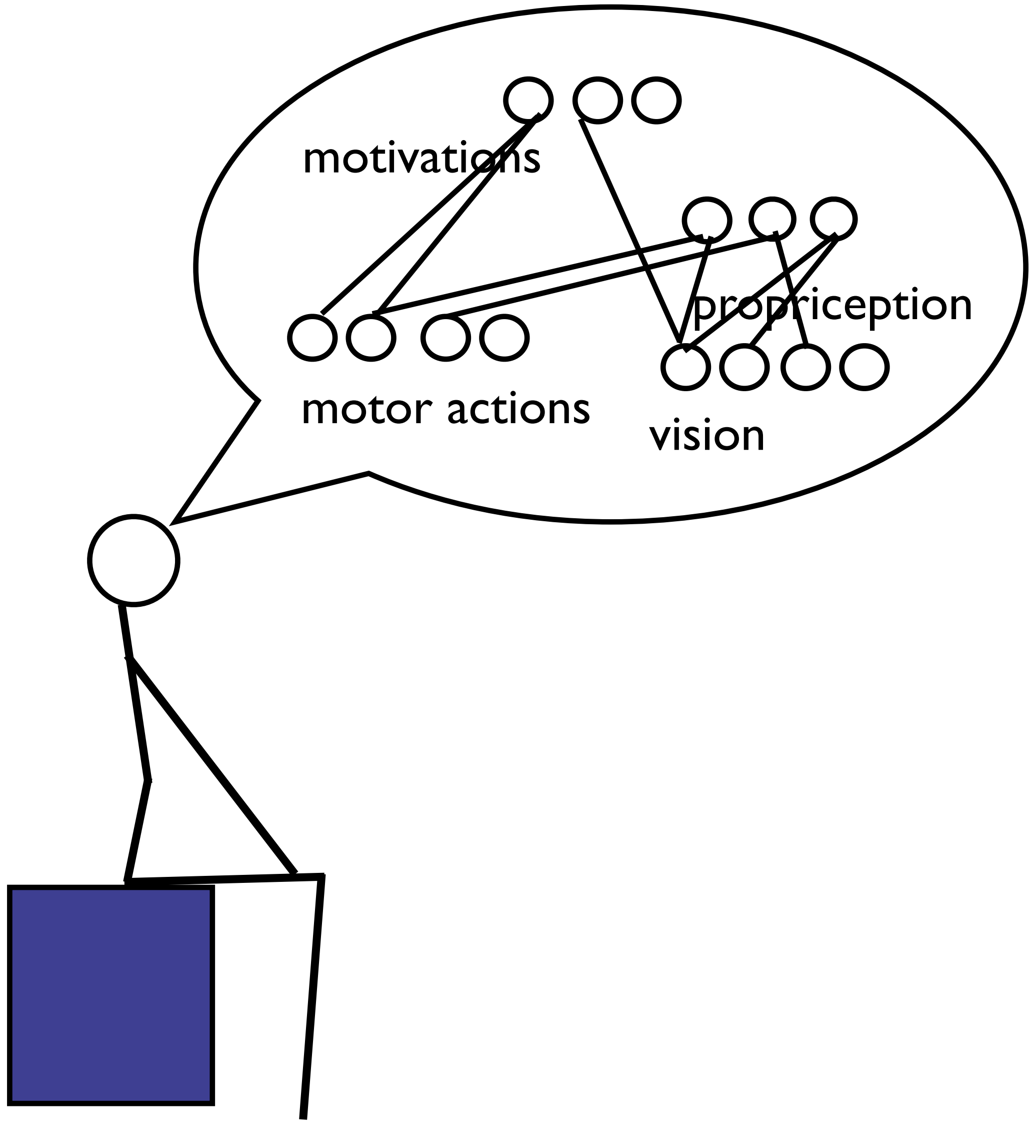
If you want to recognize a chair, you need to :

- **Build** the experience of seating
- to **categorize** it
  - *you need to have legs*
  - *you need to need to seat*



# Global introduction

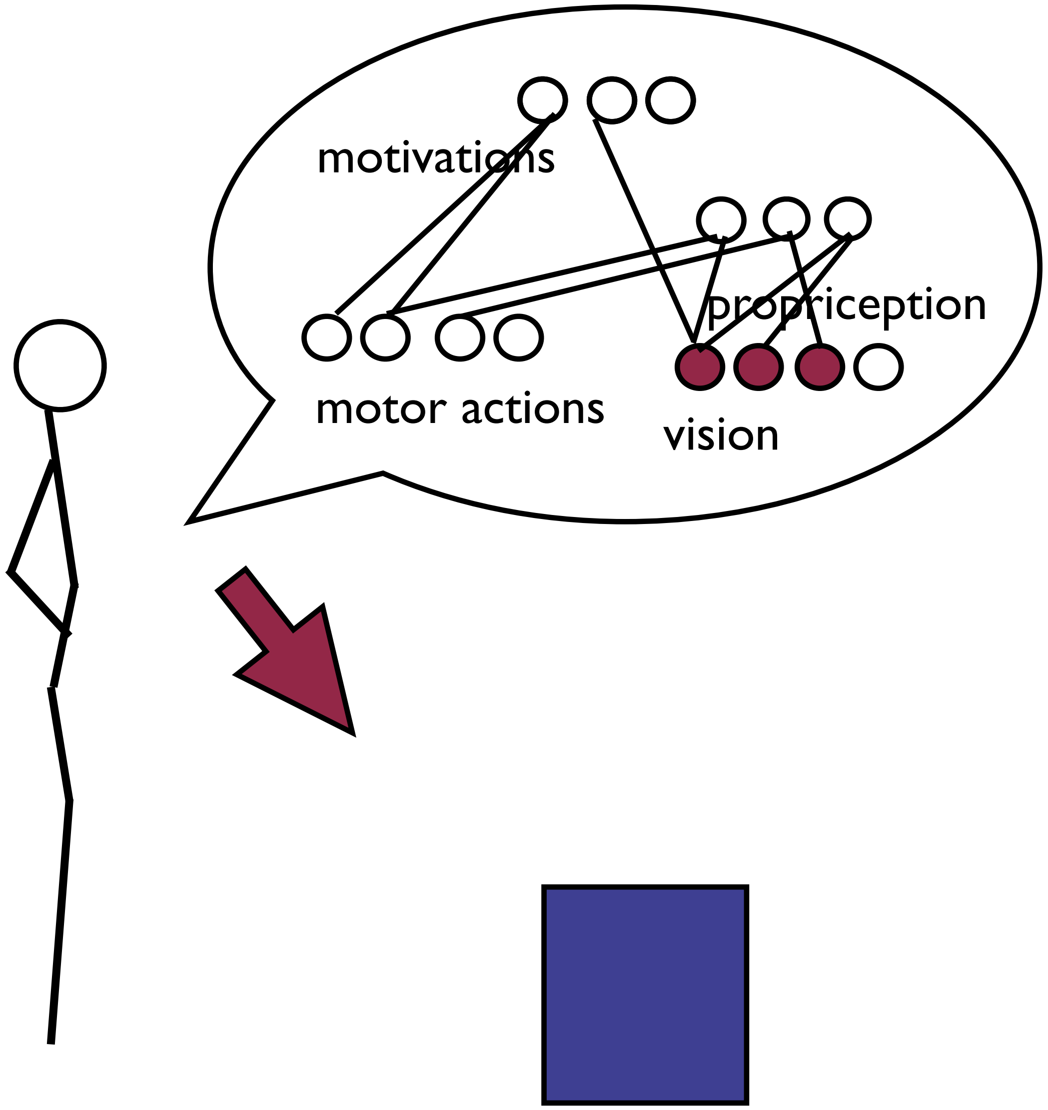
If you want to recognize a chair, you need to :




- **Build** the experience of seating
- to **categorize** it
  - *you need to have legs*
  - *you need to need to seat*
  - *merge the perceptions, the motor actions*
  - *to associate the objects vision with the whole*

# Global introduction

If you want to recognize a chair, you need to :




- **Build** the experience of seating
- to **categorize** it
  - *you need to have legs*
  - *you need to need to seat*
  - *merge the perceptions, the motor actions*
  - *to associate the objects vision with the whole*
  - *in order to generalize and recognize*

# Global introduction

Connectionism is a part of this perception-action philosophy...

and neural networks is a tool to achieve such adaptive categorization

# Overview

## Introduction : from biology to the formal neuron

**Part I : supervised learning**

- perceptron
    - simple rule
    - Widrow Hoff rule
    - limitations
- associative memories
- multi-layer perceptron
- backpropagation

**Part II : unsupervised learning**

- brain mechanisms
    - competition and cooperation
    - WTA
- Self Organizing Maps
    - Kohonen maps
    - K-means (analogy)
- Let’s put it all together
    - ART

# From biology...

The brain is the main unit for information processing.

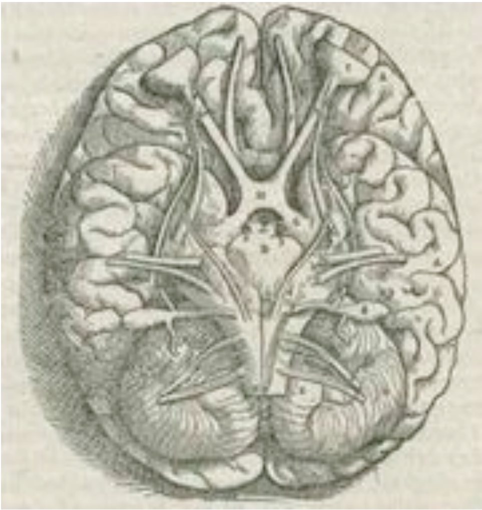
It is composed of 100 billions of nervous cells: the neurons

The neurons are connected in networks in order to :

- monitor
- regulate
- modulate

all the function of the organism.

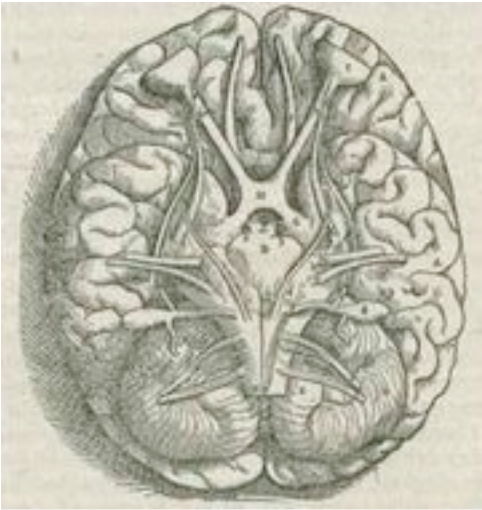
Moreover : the “organ” of intelligence



# From biology...

Cognition : all the mental processes involved in the scaffolding of *our* reality, et the basis of reasoning and all the high-level functions.

- perception
- memories
- building categories
- learning
- inhibition
- action selection
- *representations*



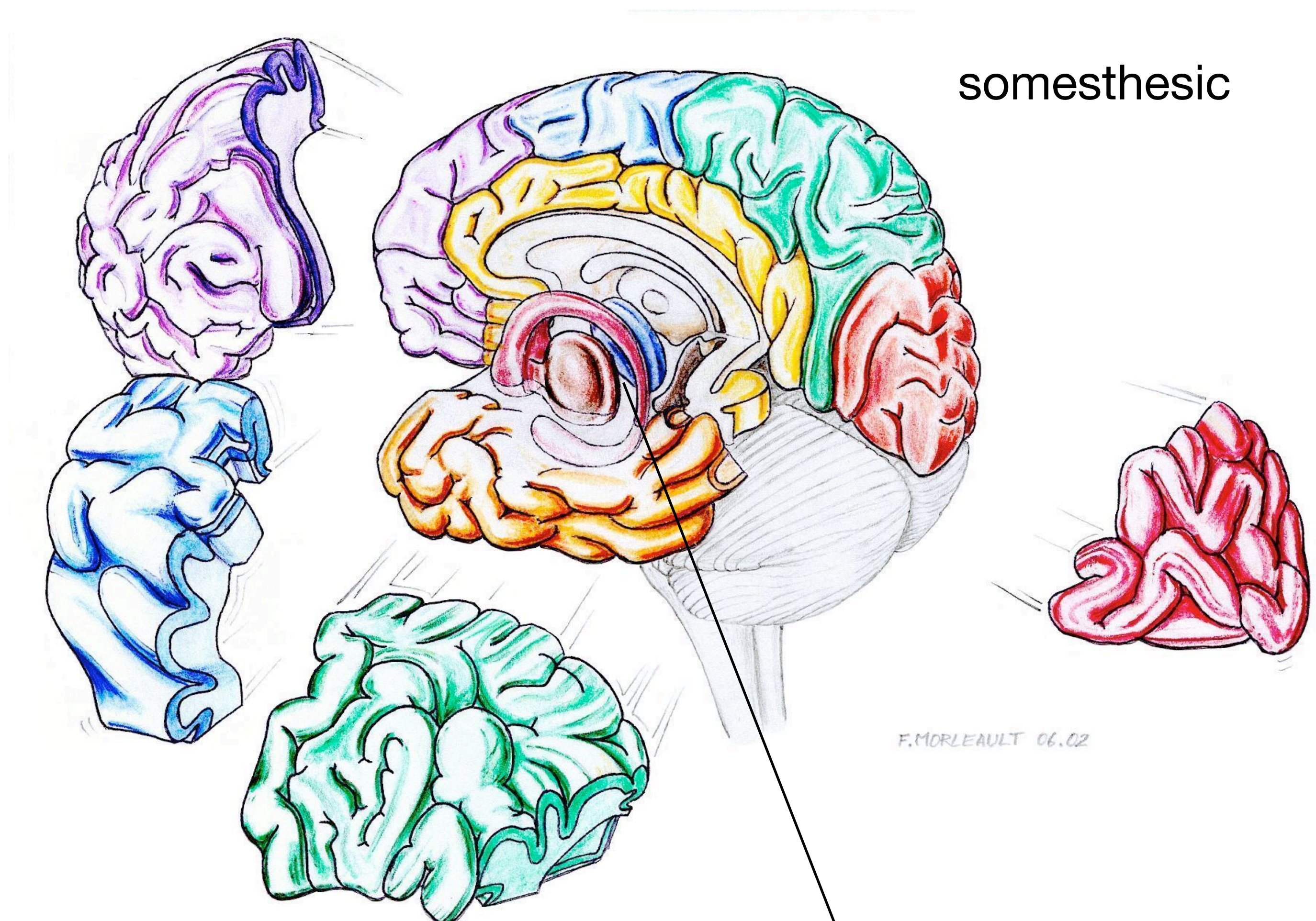
Information processing : input -> evaluation->decision->action

# From biology...

cortical areas

somesthesic

PFC




visual areas

motor

hippocampus

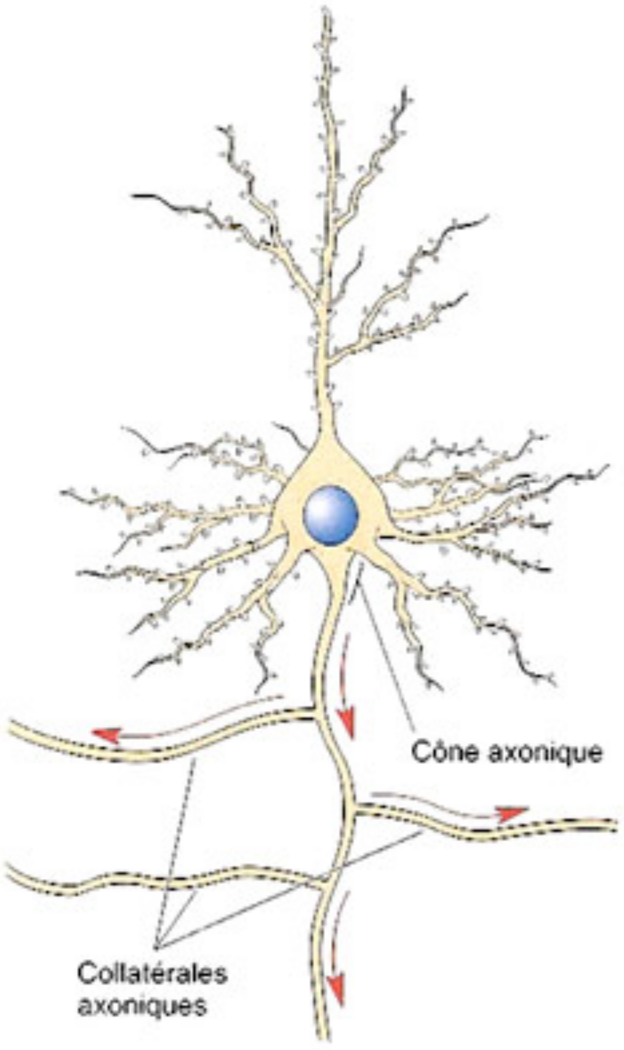
basal ganglia

amygdalia

# From biology...

Neuron : the main unit for information processing

- Dendrit
- Axon
- Soma
- Synapses

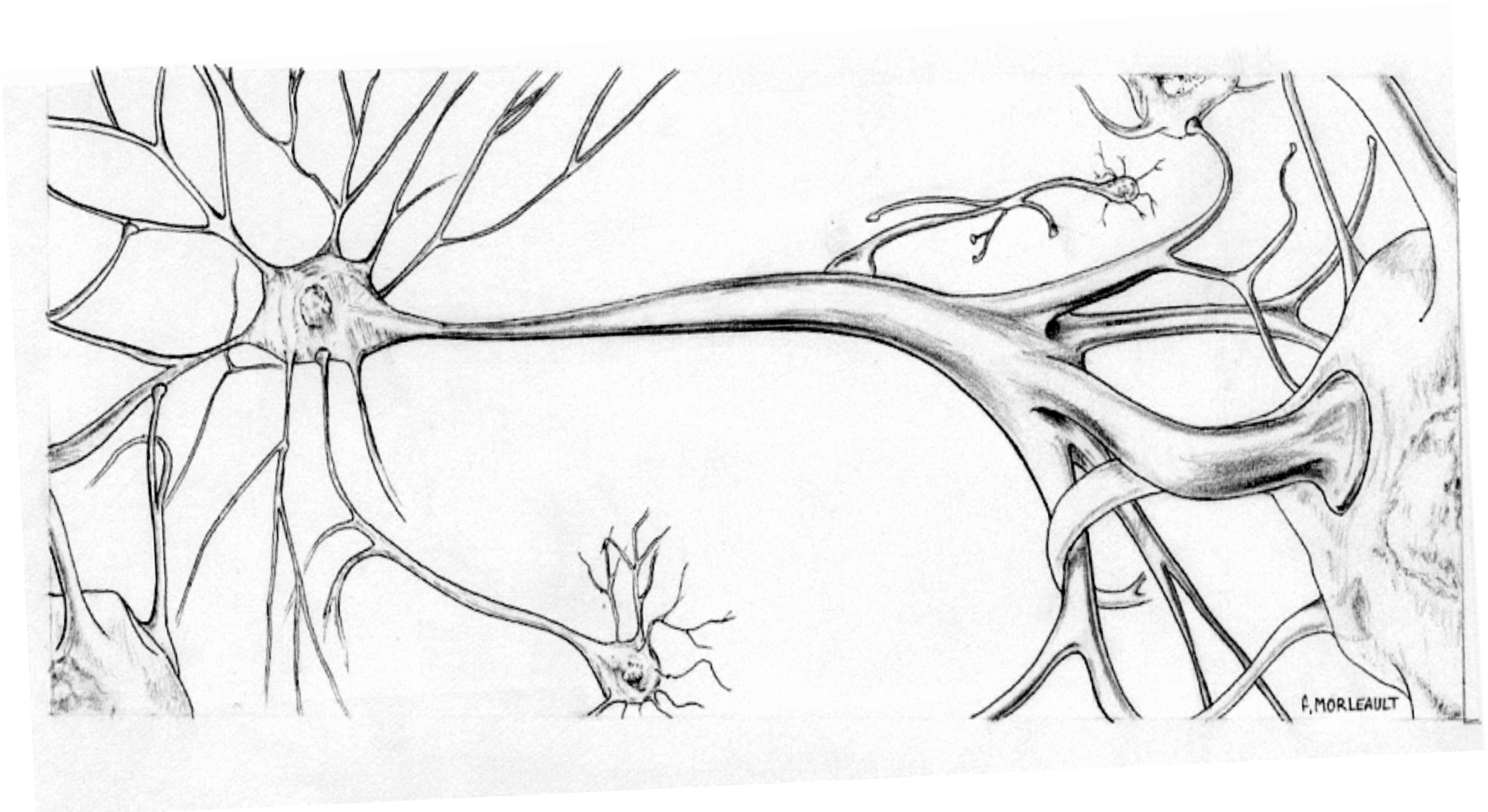



Electric information propagation

uni-directional

from one unit to the other

# From biology...



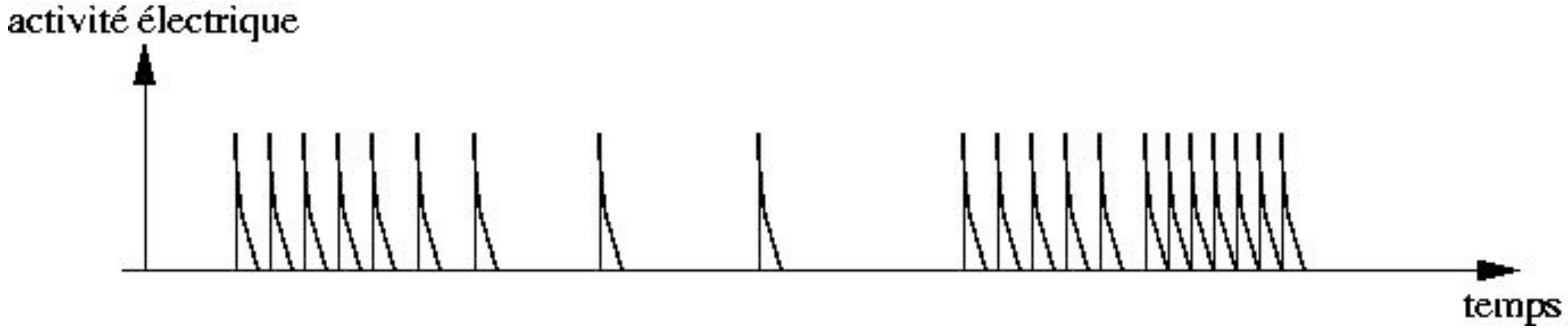

A single neuron can be 1m long

A single neuron can have up to 10 000 connections with other ones

(average of 1000 connections per neurons)

# From biology...

But very slow

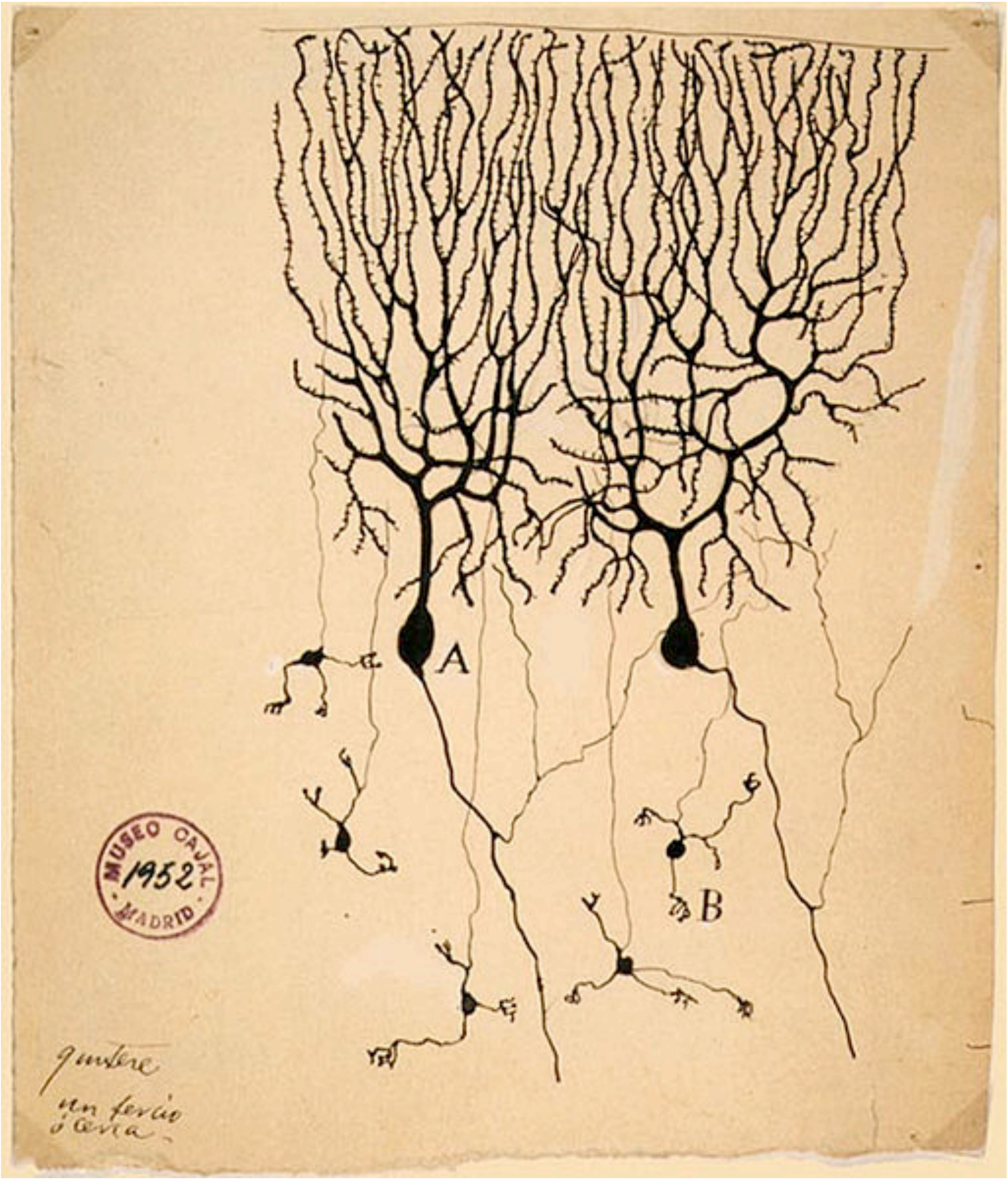



- If you measure the speed of information transmission, you get a rate of 300HZ
- 300HZ = frequency of the action potential : Spike train
- very slow, when compared to modern computer buses (circa 1GHZ)

# From biology...

But...

- Multiple connections
- massive parallelism : continuous flow of information tat can be processed by multiples areas at the same time (asynchronous calculation - no main clock-)
- Adaptation : the synapse can adapt to modify the information intensity transmitted by a neuron to the other: learning



# From biology...

comparison brain/ modern computer

| Von Neumann Architecture | Brain |
|---|---|
| Computation and memory:<br>separated and centralized | Computation and memory<br>integrated and distributed |
| program = sequence<br>of instructions | calculus = multiple constraints<br>satisfaction |
| execution  of one process<br>at a time | permanent combination of<br>multiples different information sources |
| 1 to 8 very fast processors<br>t | hundreds of billions of slow connected units |

# ...to information processing...

# ...to information processing...

At the basis any behavior, is information processing

2 different kinds :

- Chemical, long term (LT) “message sending” **: hormones.** chemical and not restricted to a given receptor, diffusion in all the body, transported by the blood. Slow process.

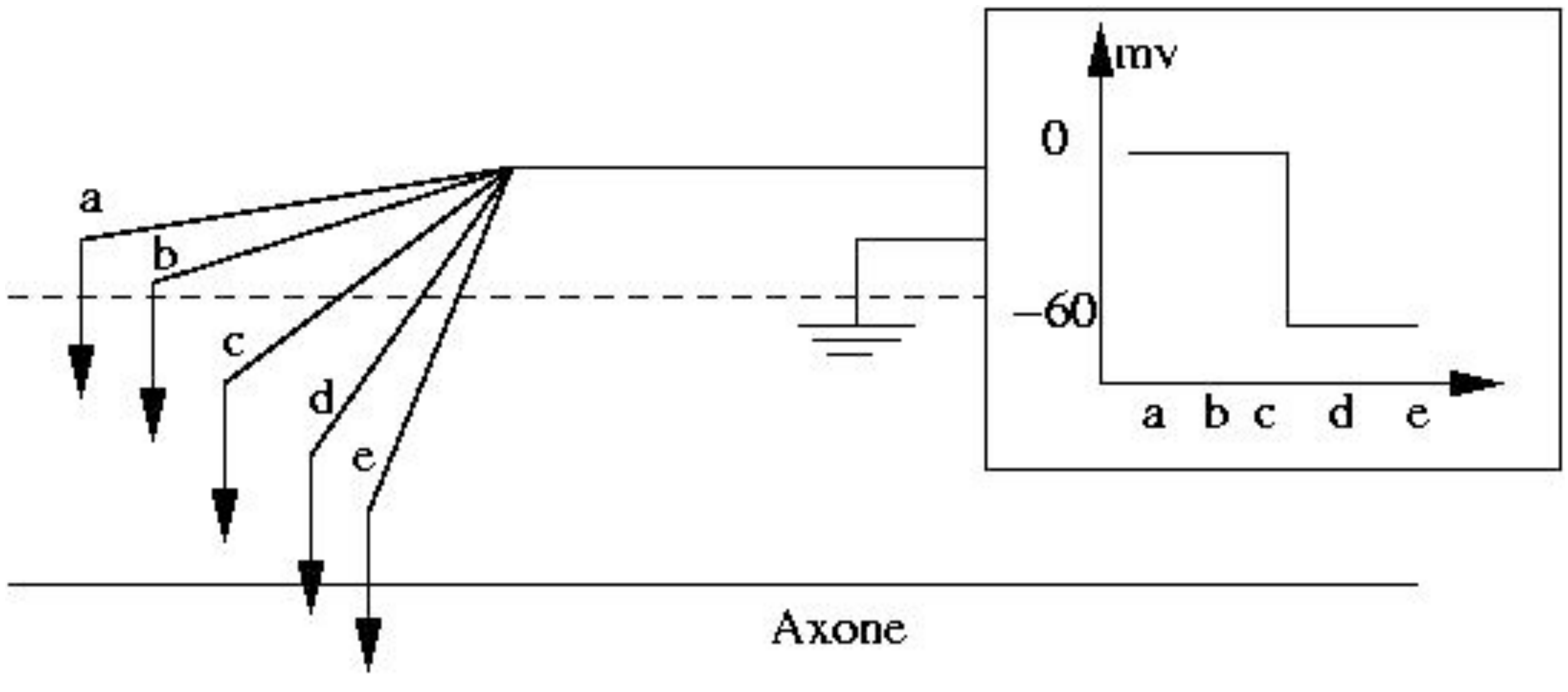
- Electro-chemical, short term, **potential trains** sent by neurons. “Fast”

  - *One neuron produces electrical potential changes, and the changes (the potential variation) are transmitted all along the membranes : emission of electrical signals*

  - *The communication int the inter-neural space (between the synapse and the dendrite) is made by production of chemical substances, the neurotransmitters*

# ...to information processing...

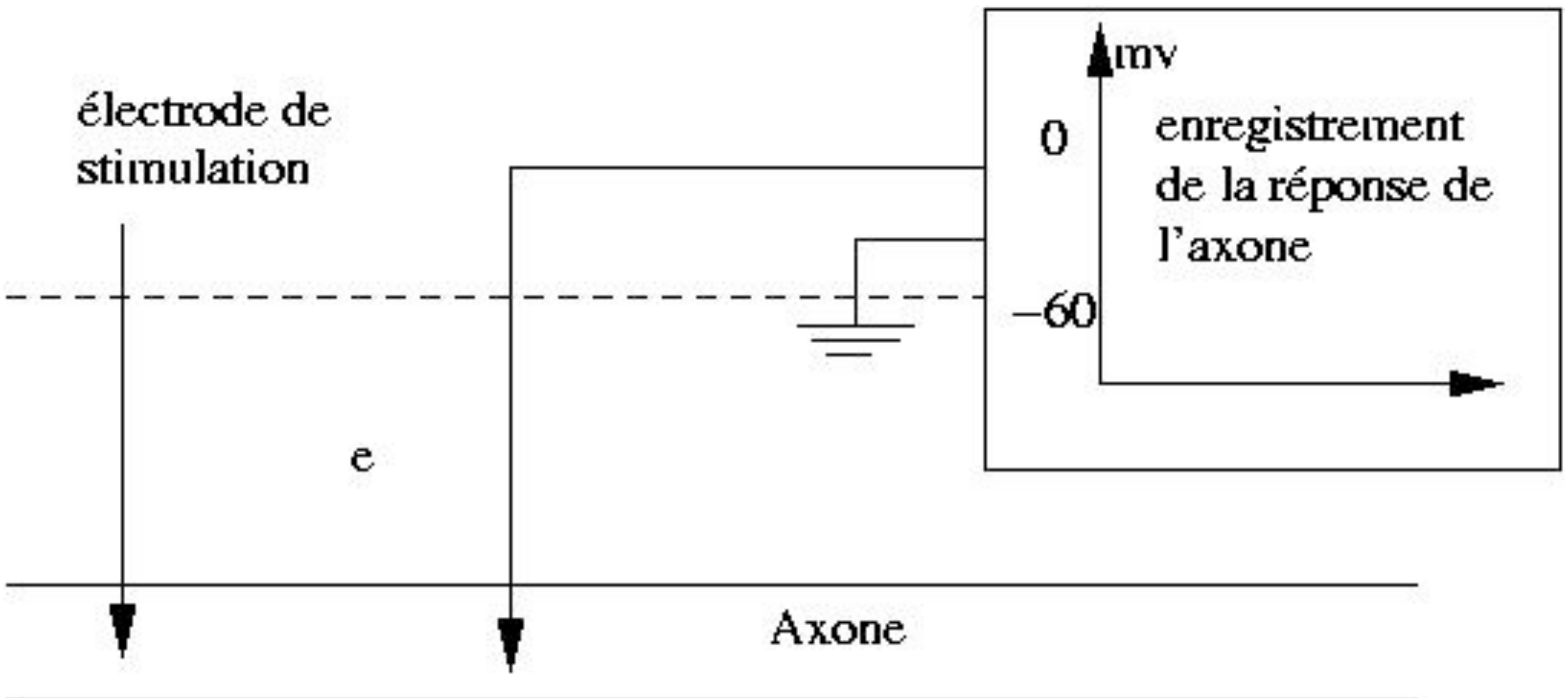
- suppose you have a multimeter to measure the value of the electrical potential inside and outside the axon membrane




- default resting potential (of the neuron): -66 mv : the inside of the axone is negatively charged (difference between a,b,c, and d, e )

# ...to information processing...

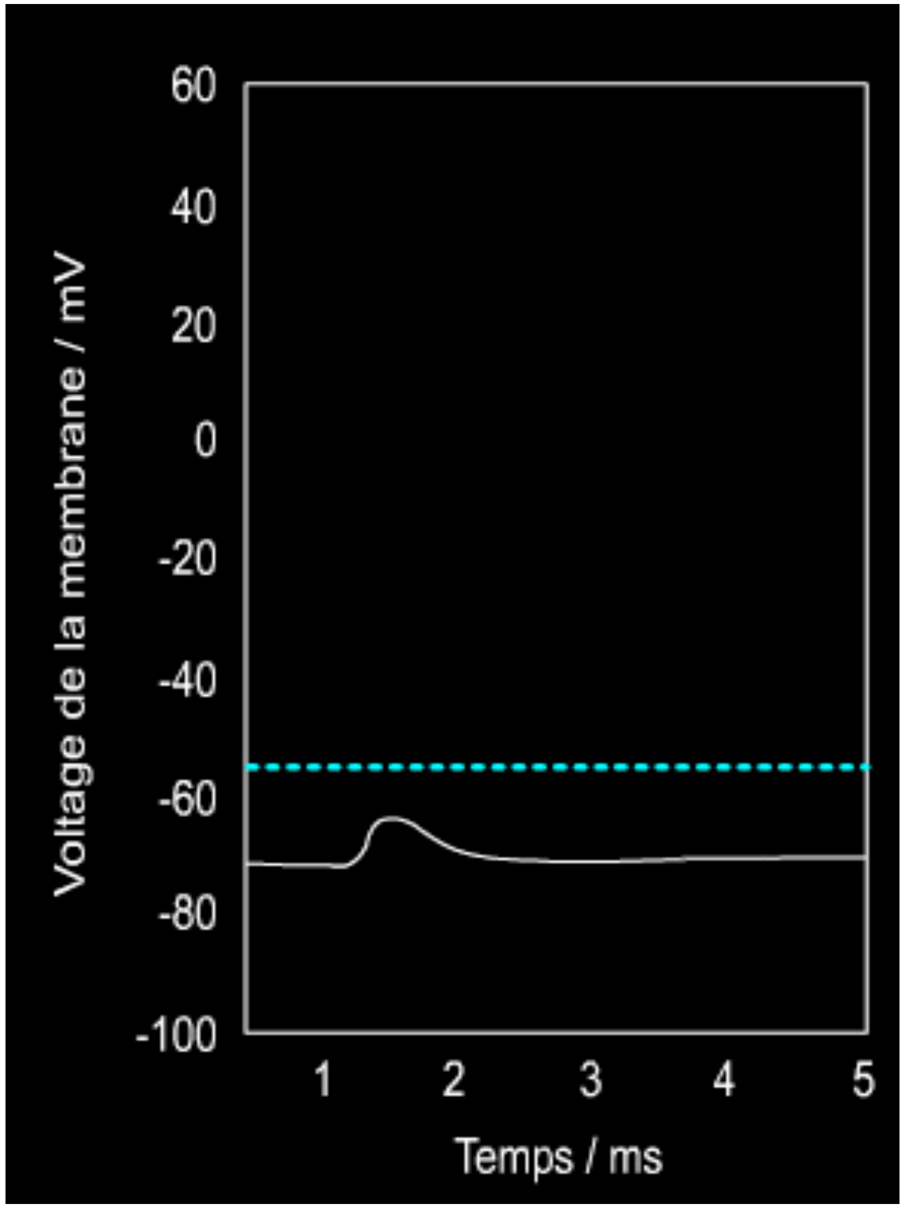
- Suppose you plug an electrode and send a given voltage in the membrane of the axone...




- ...and you record the axon's membrane reaction at a given point

# ...to information processing...

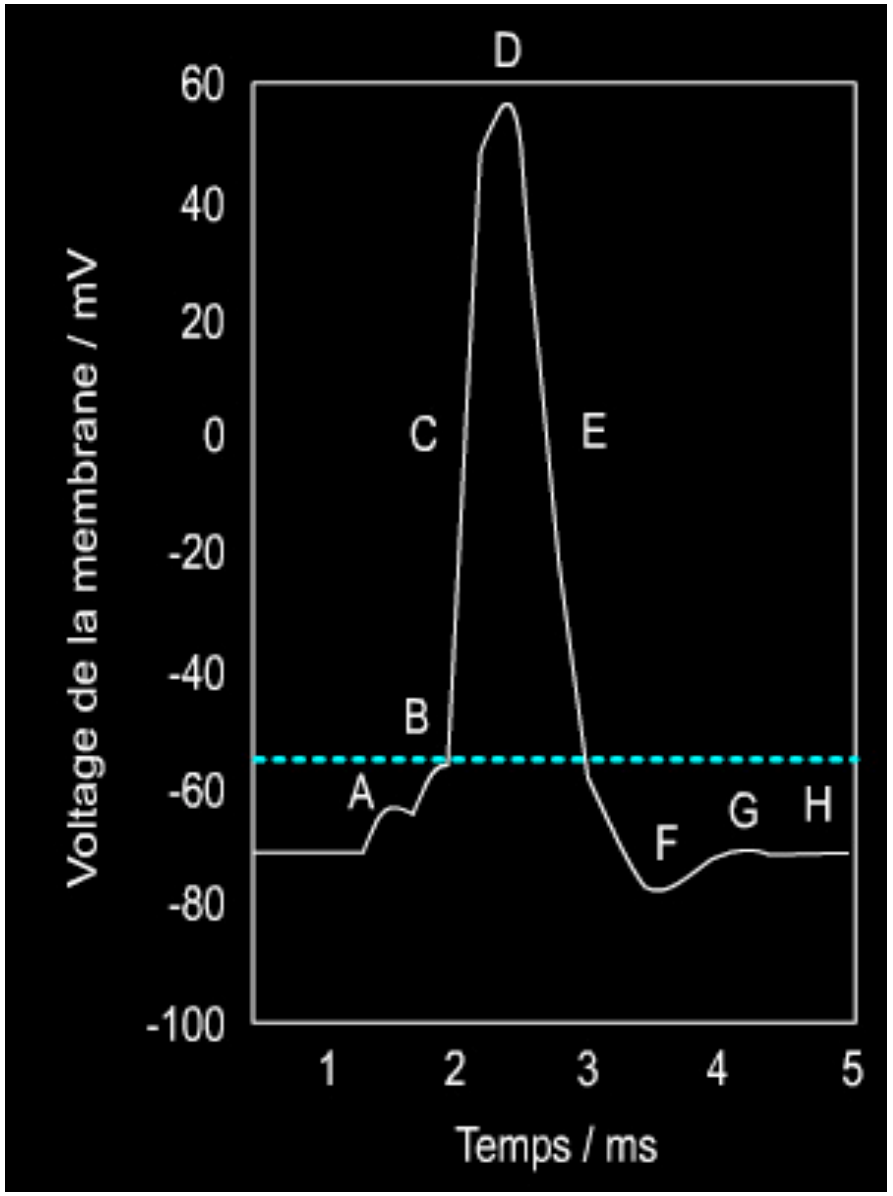
- Stimuli under 10 or 15mv : no significative response from the membrane



# ...to information processing...

- Depolarization > 10 or 15 mv : strong reaction called action potential .

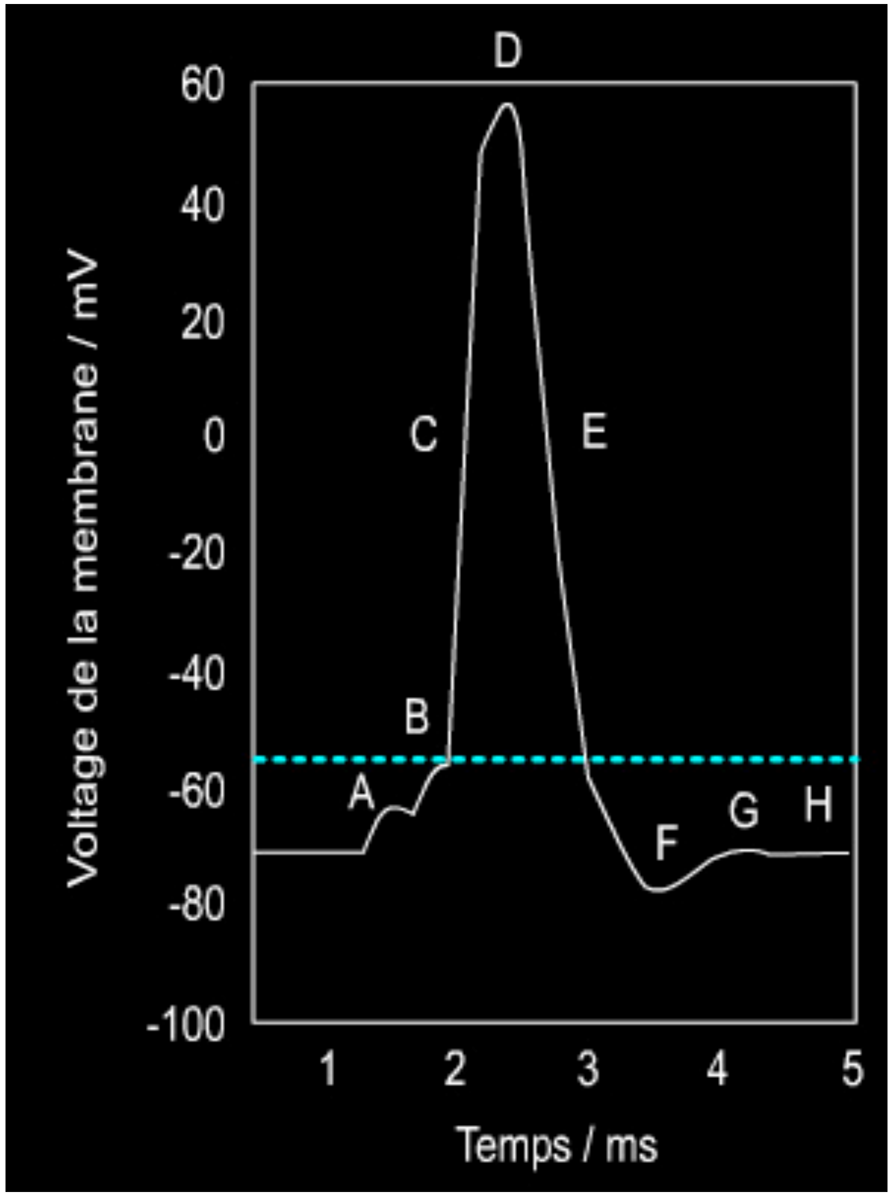
- spike reaching +60mv (C-D)
- overshoot
- whatever the stimulation is (>15mn), the spike is the same
- stereotyped
- followed by a short undershoot (F-G) during no new stimulation is possible: explain the 300Hz limit



# ...to information processing...

- Depolarization > 10 or 15 mv : strong reaction called action potential .

- The spike will propagate all along the axon with conservation of shape and amplitude
- traveling electrical wave
- 30 m/s



# ...to information processing...

- If the intensity of the trigger has no effect on the shape of the peak, it will nevertheless influence the amount of peaks generated.

- the stronger the stimulus is, the greater is the number of peaks (frequency max : 300hz)

- The axon is providing a frequency code for the intensity of the stimulus

- potential trains

- naturally, the trigger of the pikes potential will depend on the activity of the neuron’s nucleus : the soma

# ...to information processing...

- The neuron's nucleus will act as an *adder*, summing the incoming potentials (arriving potentials from the dendrites).

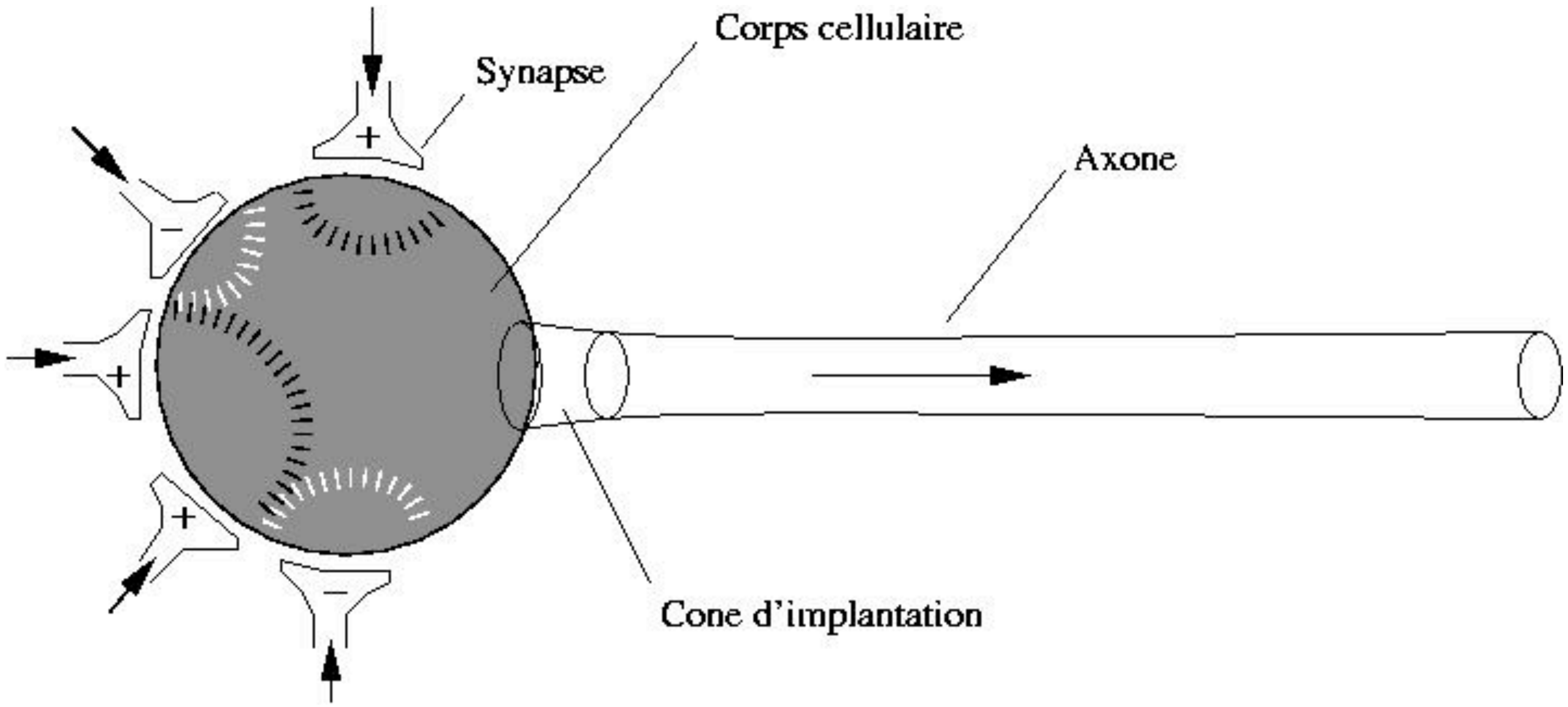
- it is called the soma (soma = sum in latin)

- temporal en spatial sum according to :
  - the synapses positions on the dendrite
  - the frequency of the peaks

- The soma will trigger the axon if a given threshold is overshoot

- non-linearity : thresholds + saturation

# ...to information processing...



# ...to a simple formal neuron

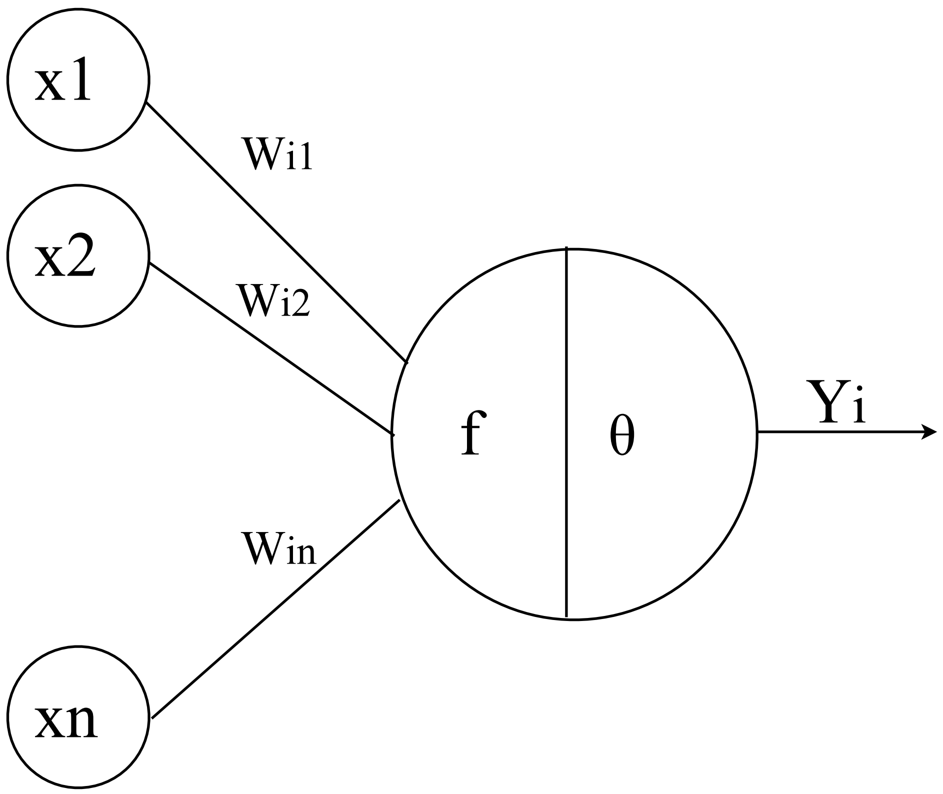
Formal neuron  i

- main processing unit
- combined in networks
- calculate the incoming potentials, the Xi [-1,1]:
- summation in an internal potential Poti [-1,1]
- Threshold $\theta$ [-1,1]
- deliver one output Yi [-1,1]
- For our introduction we won't use the frequency coding : just a value expressing the mean number of emitted potential
- McCulloc & Pitts model [MacCullocs & Pitts48]

See the works of Thorpe & Al. for models of neurons coding the frequency of the action potential : spiking neurons

# ...to a simple formal neuron

Formal neuron i



# ...to a simple formal neuron

Formal neuron i

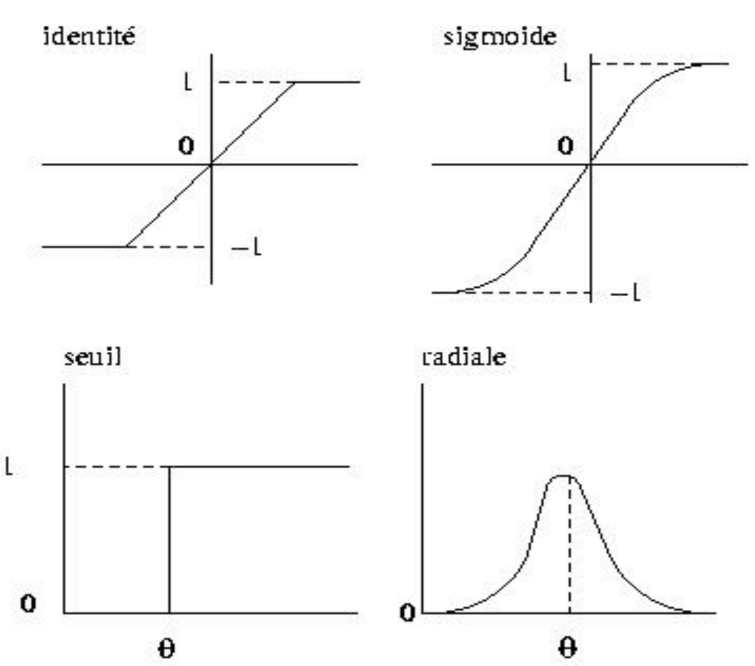
- calculate the incoming potentials, the Xi [-1,1]:
- summation in an internal potential Poti [-1,1]
- Threshold [-1,1]
- deliver one output Yi [-1,1]
- we have seen that the information transmission between two neurons is made by emission of neurotransmitters.
- Coefficient W [-1,1] of the incoming potential

# ...to a simple formal neuron

Formal neuron i

- Internal potential of neuron i : $pot_i = \sum w_{ij}.x_j$
- activation of output : $S_i = f(pot_i)$
- with f, the transfer function according to the model

For example :




identity function

$$f(x) = \begin{cases} 0 \text{ si } x < \alpha \\ 1 \text{ si } x > \beta \\ x \text{ si } x \in [\alpha,\beta] \end{cases}$$

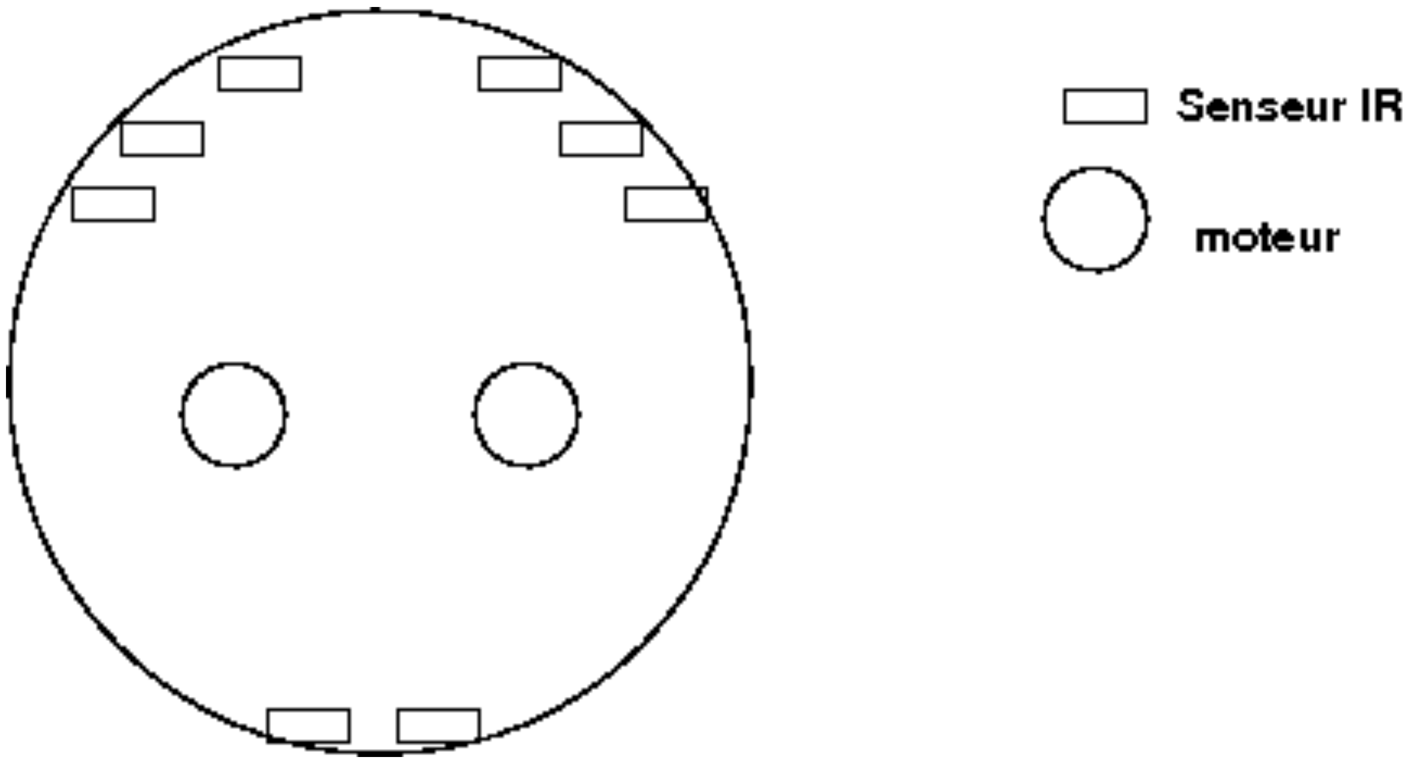
with $\theta = \beta - \alpha$

# ...to a simple formal neuron

Topology of the network ?

# example : vehicles

- Braitenberg's vehicles [Braitenberg80]
- You have to build a roomba robot that avoid obstacles.




- Sensors [0,1024] saturation when contact to obstacle
- motors : speed control [-1,1]

# example : vehicles

- Classical analysis:

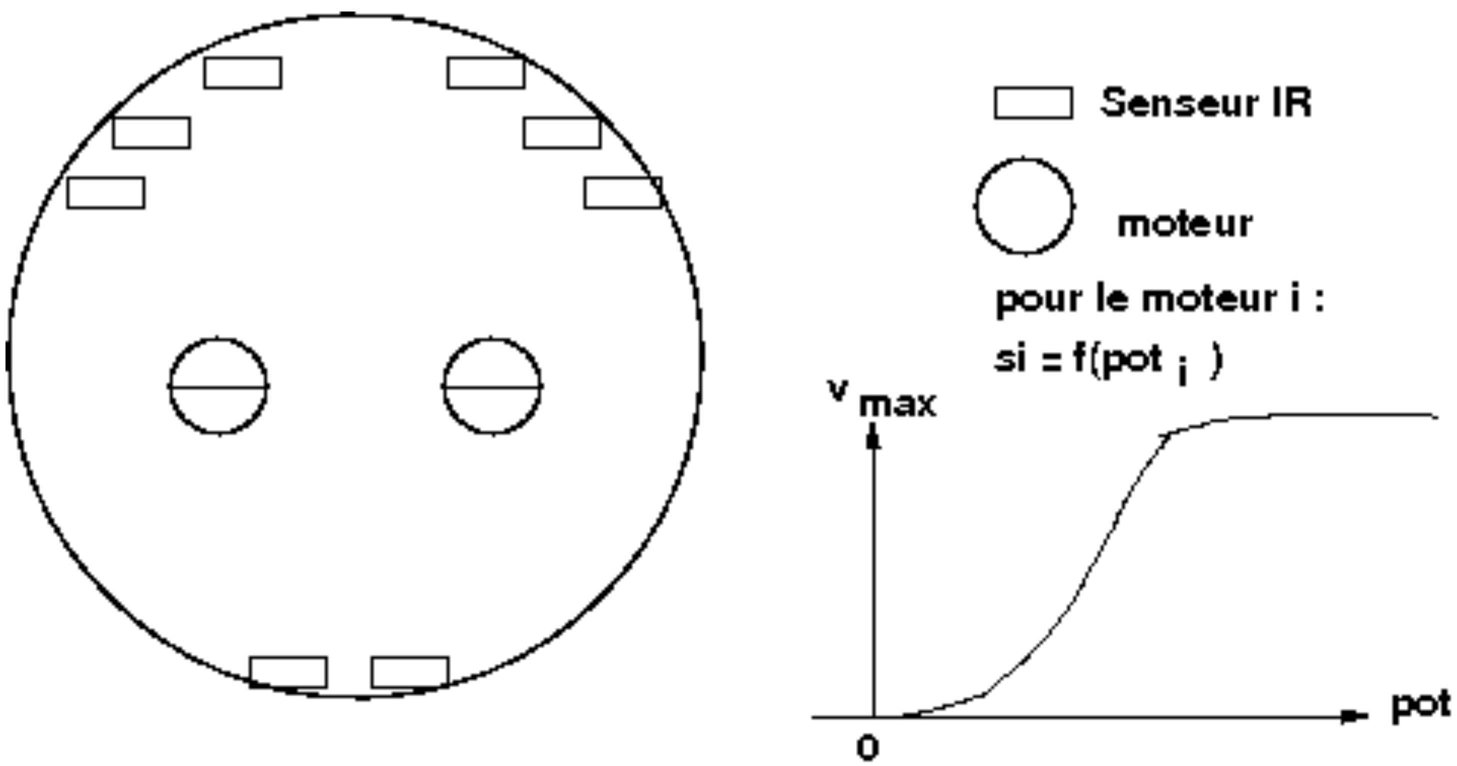
```
if obstacle_detected == left
       turn (angle_right)
if obstacle_detected == right
      turn (angle_left)
 if obstacle_detected == right&left
      ...
else ...
```

obstacle is an important symbol

Difficult to recognize, frame, (remember the chair...)

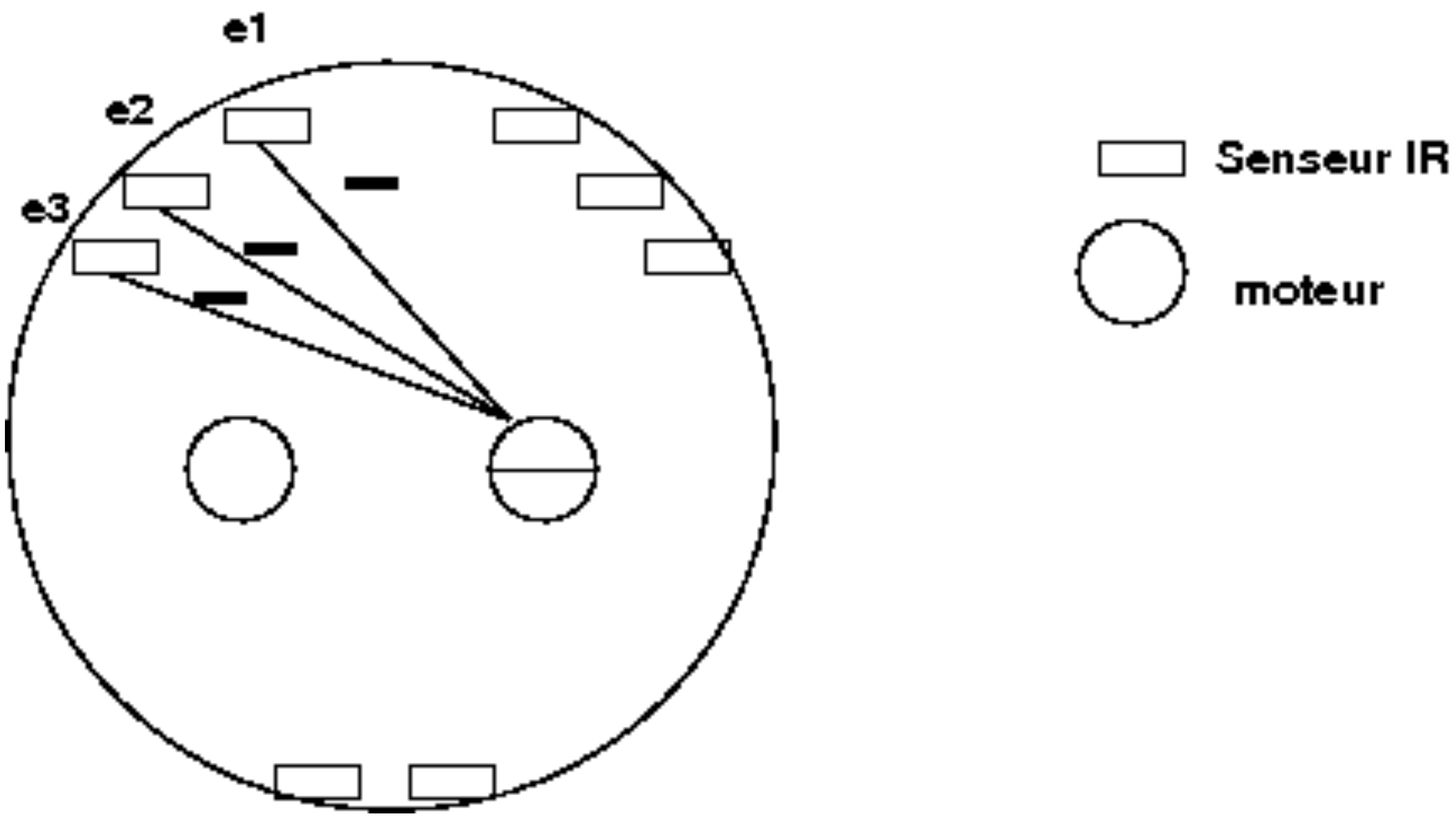
# example : vehicles

- normalize sensor activities [0,1024] -> [0,1]



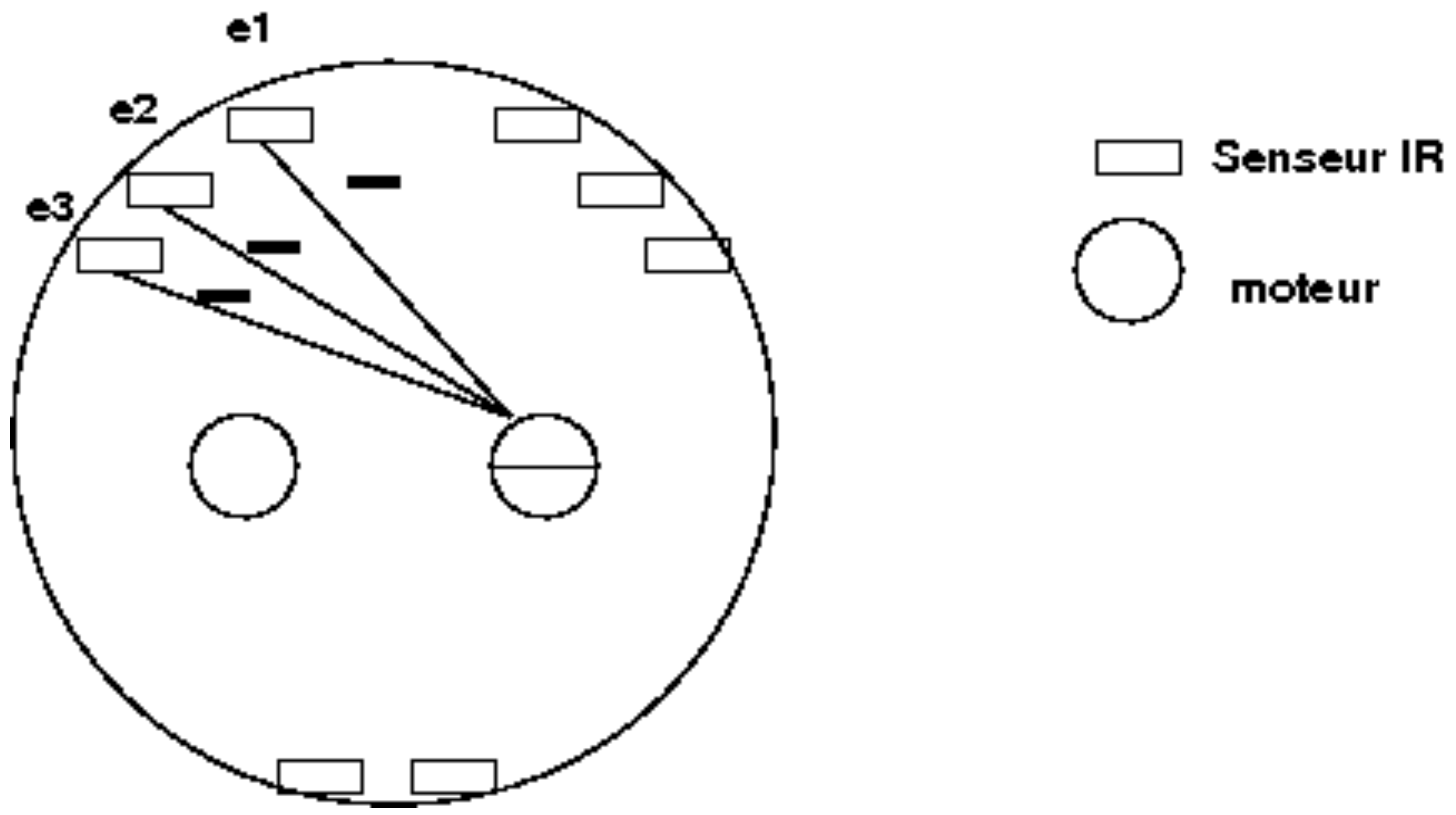
# example : vehicles

- build connection (weight 1/3 to normalize) to the opposite motor.
- suppose the motor is activated by a neuron
- use identity transfer function



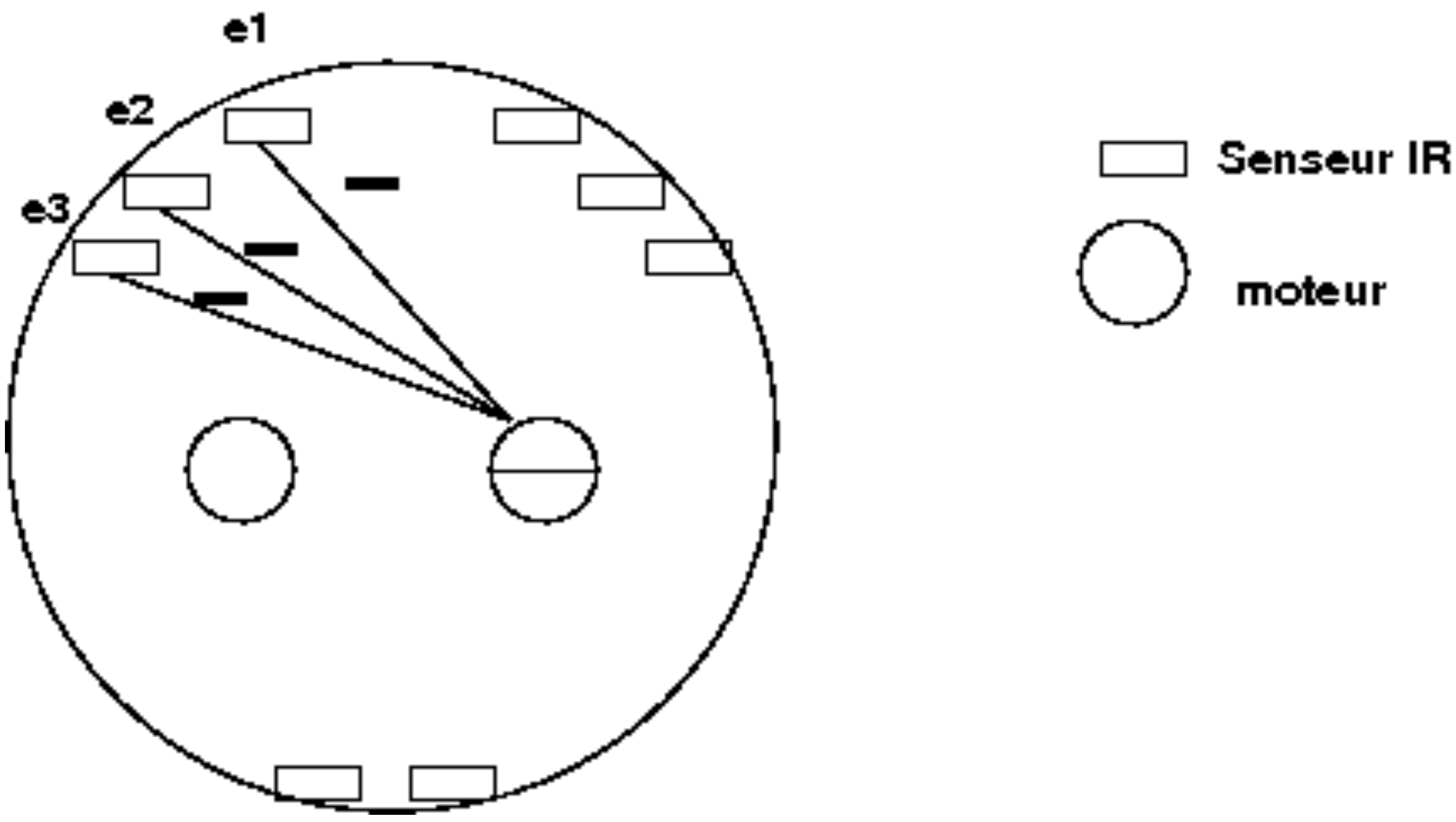
# example : vehicles

- sensor saturation induce a diminution of the neuron potential : the opposite motor runs slower, the robot turn



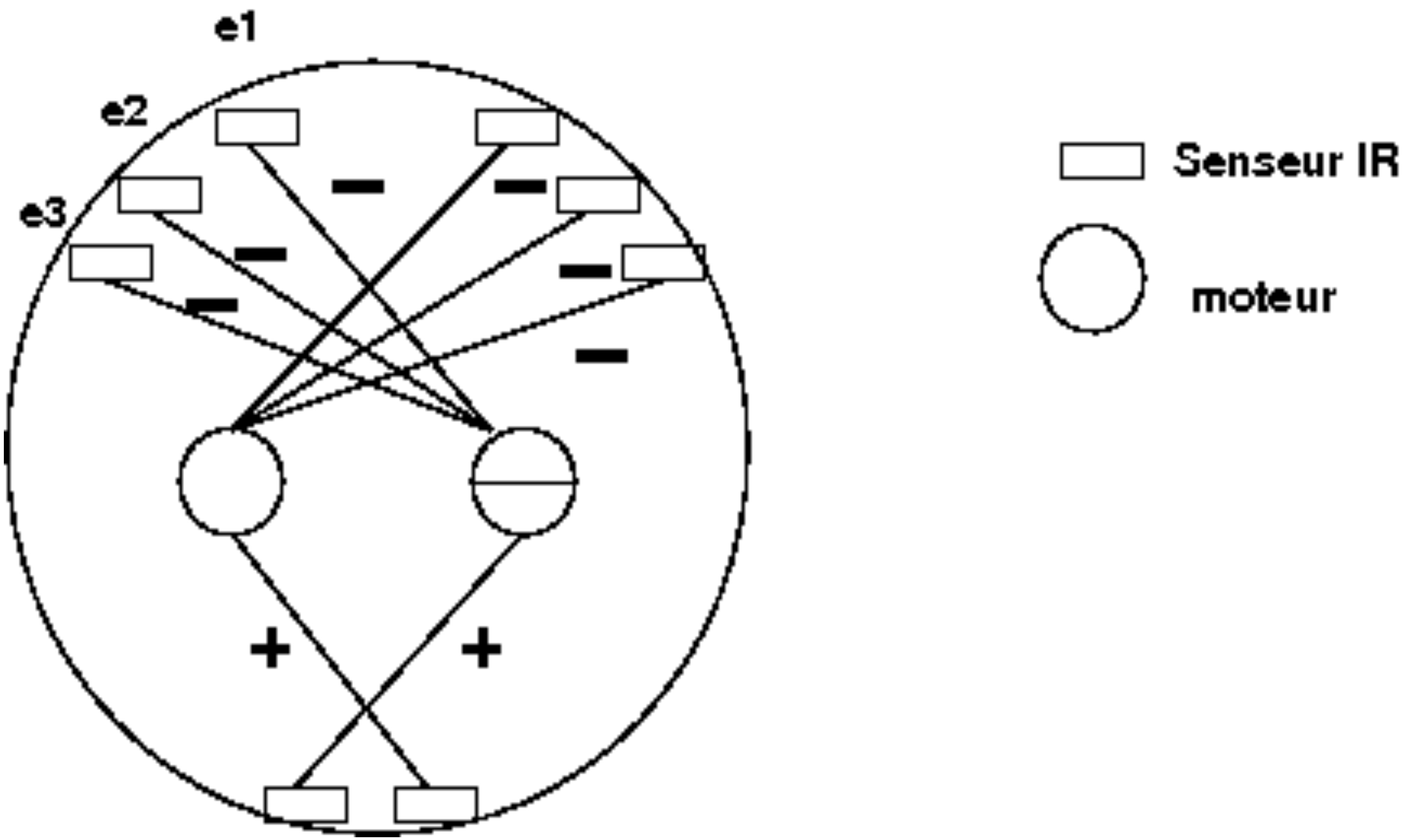
# example : vehicles

- Code : only simple operations: pot(i) = e1*w1+e2*w2+e3*w3



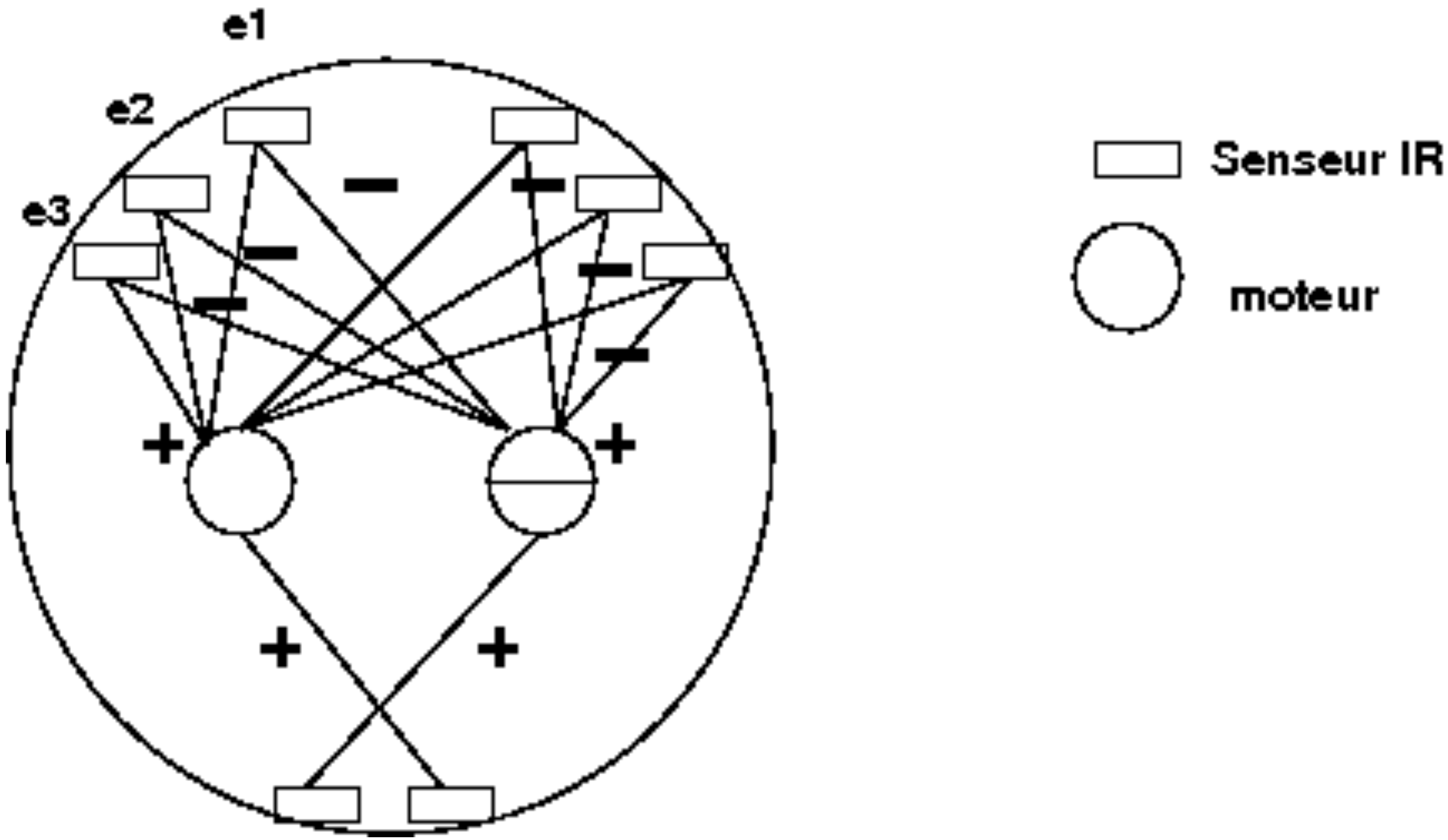
# example : vehicles

- Complete architecture:



# example : vehicles

- enhancement :

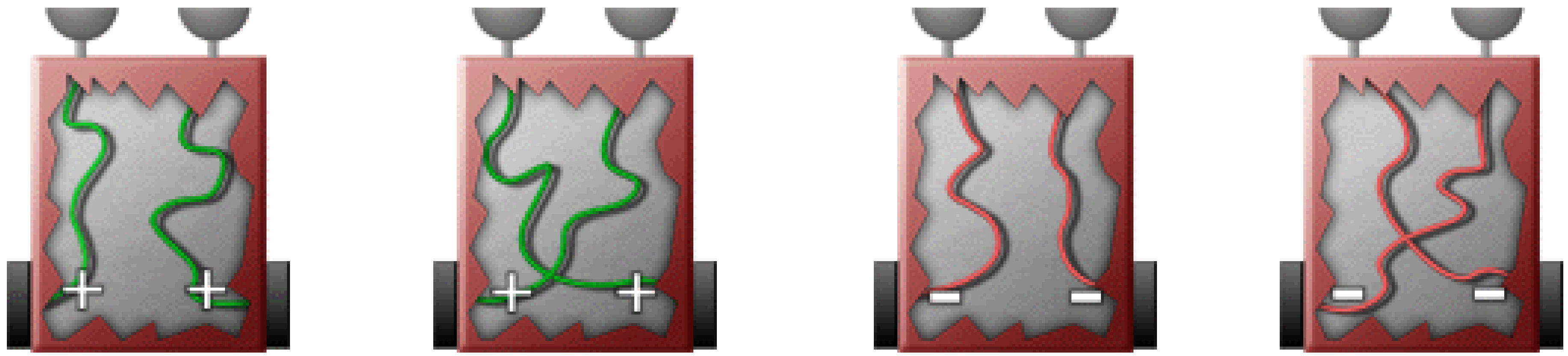


# example : vehicles

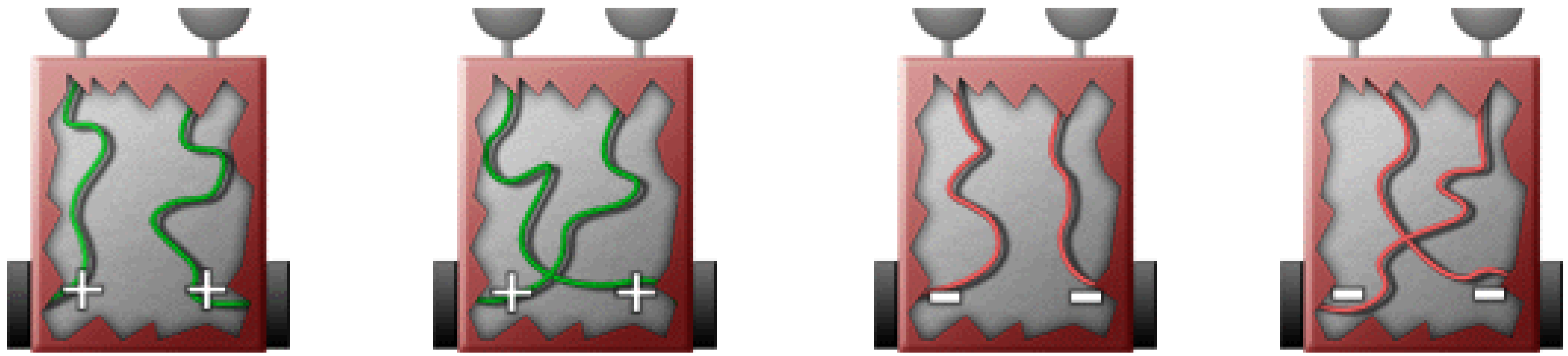
- only 2 neurons
- topology matters
- No “if”, just a vector product : light calculation.
- No notion of obstacle

- behaviors can be stacked with more sensors :
  - obstacle avoidance and phototaxis =
  - stacking circuits

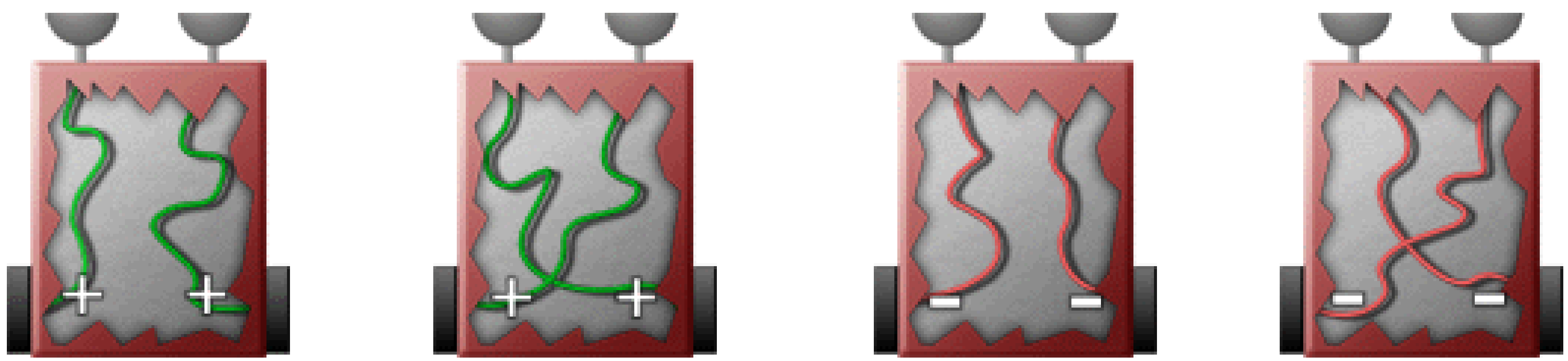
# example : vehicles



# example : vehicles

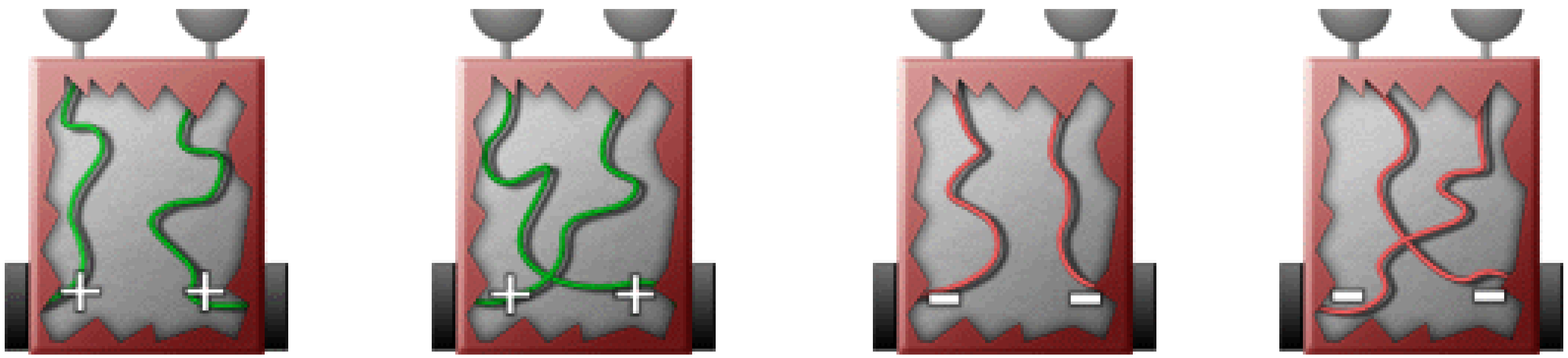


# example : vehicles



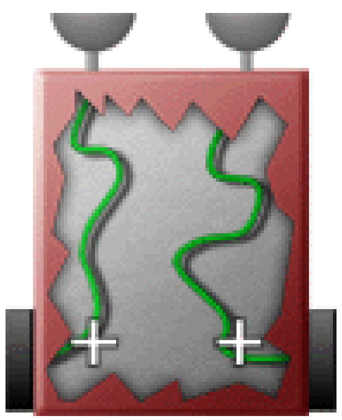
fear

# example : vehicles

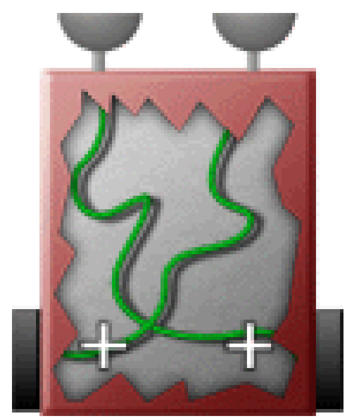


fear angry

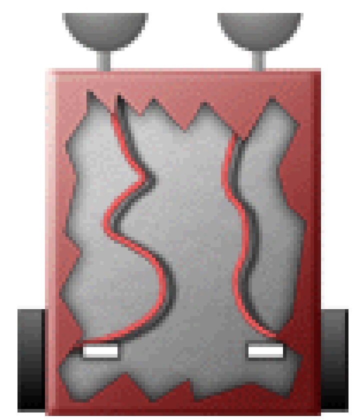
# example : vehicles



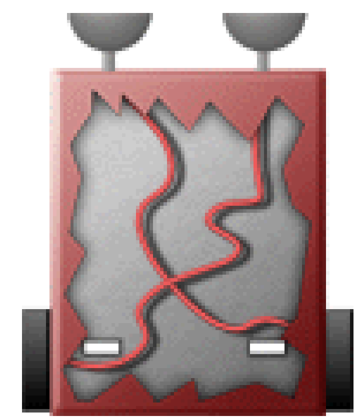
fear



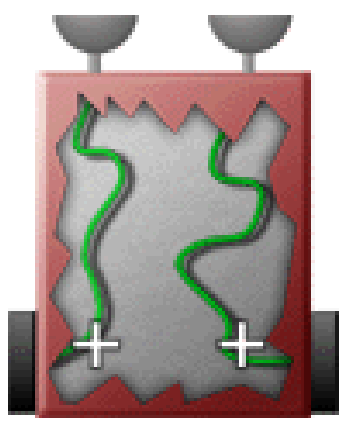
angry



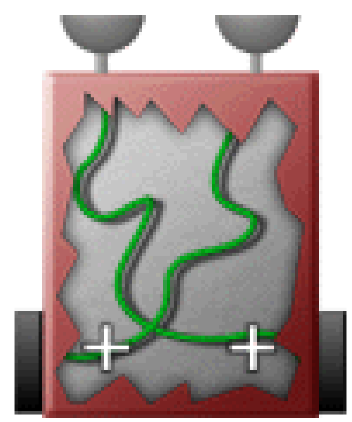
love



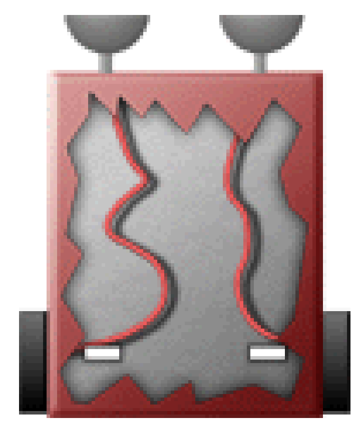
# example : vehicles



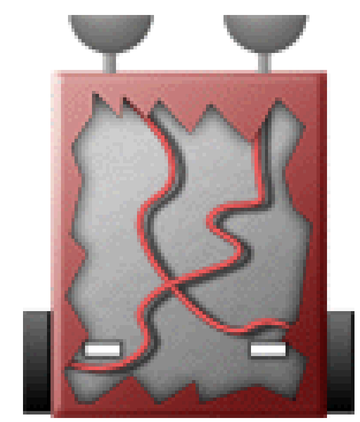
fear



angry

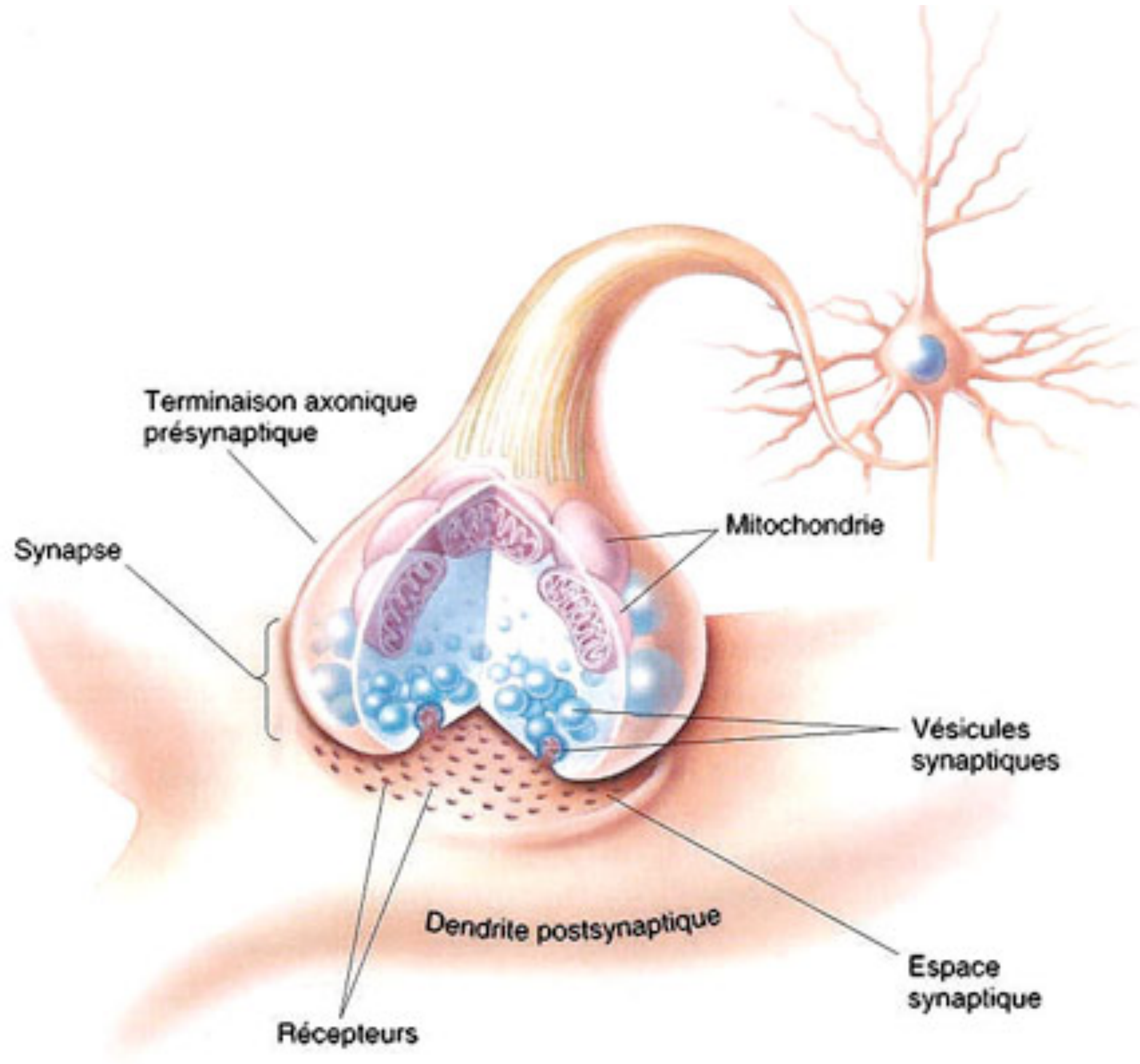


love



exploration

# Learning


Terminaison axonique
présynaptique
Synapse
Mitochondrie
Vésicules
synaptiques
Dendrite postsynaptique
Espace
synaptique
Récepteurs

# Learning

- nervous circuits : also crucial for memory (no dedicated centralised structure).
- Observation : an informal experience is correlated to measurable neuro-chemical and neuro-anatomical modifications in the brain
- Physiological modification of  synapses:
  - Pré-synaptic : increase release of neurotransmitters
  - Post-synaptic : increase sensitiveness of the receptive membrane
- Structural modifications :
  - the frequent “use” of a circuit induce an increase of the synaptic contacts

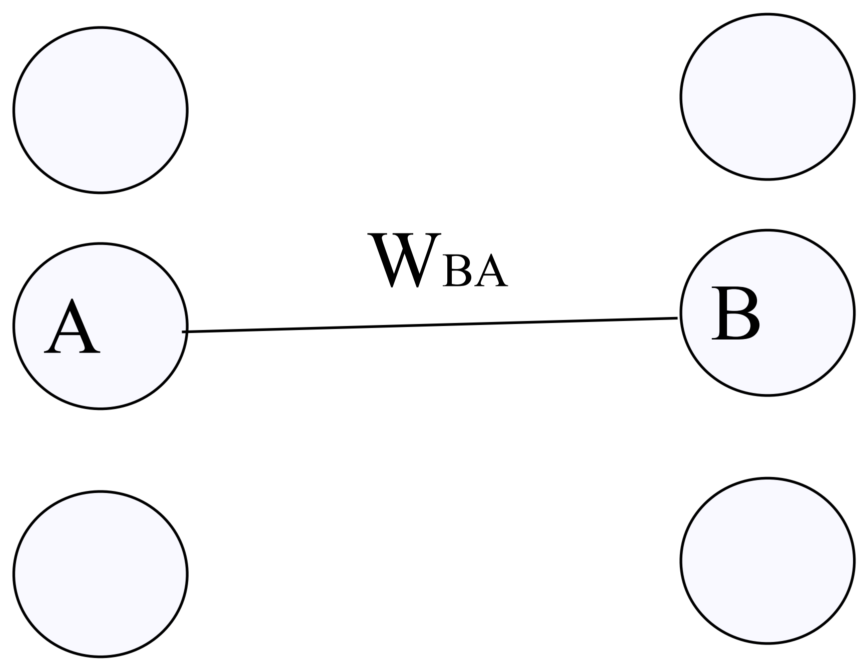
# Learning

Hebb Rule [Hebb 49] :

"*when cell A excites by its axon cell B and, in a repeated and persistent manner, it triggers impulsion of B, a process of metabolic change happens in one or two of both cells, driving to an significant increase of the efficiency of A to generate an impulsion in B, among the other cells*"

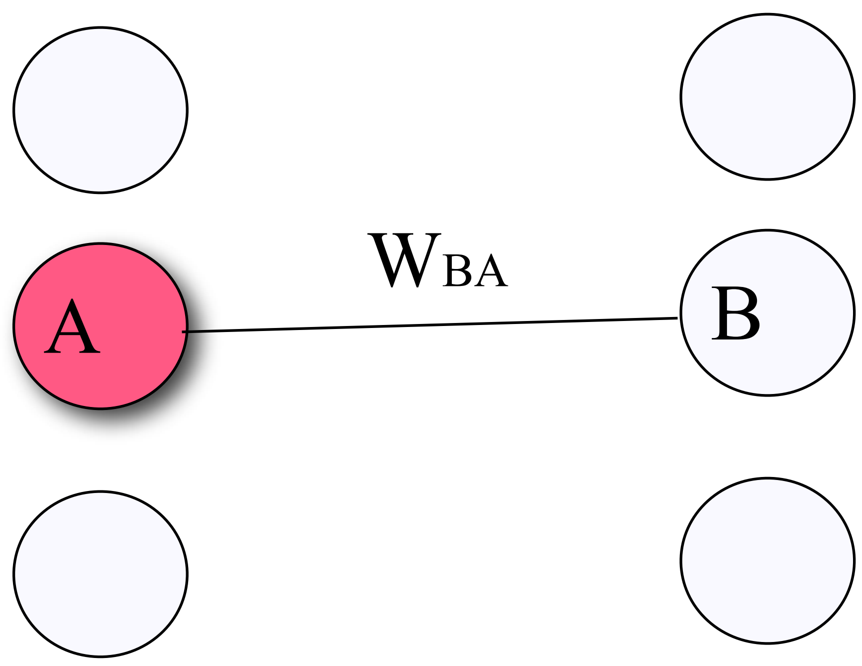
# Learning

Hebb Rule [Hebb 49] :



# Learning

Hebb Rule [Hebb 49] :

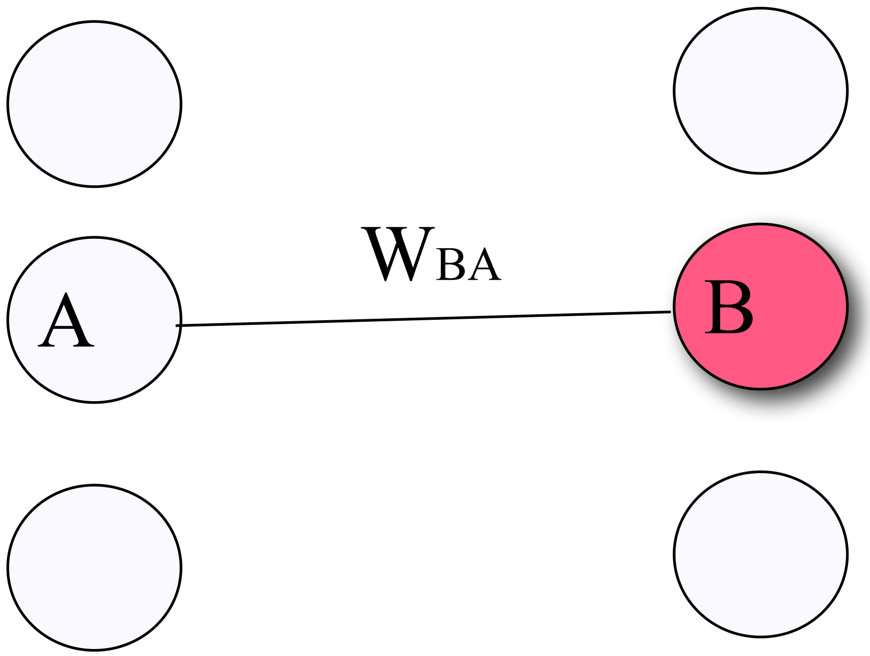



The sole A activity is not enough  to induce B activation

$A*W_{BA} < \theta_B$

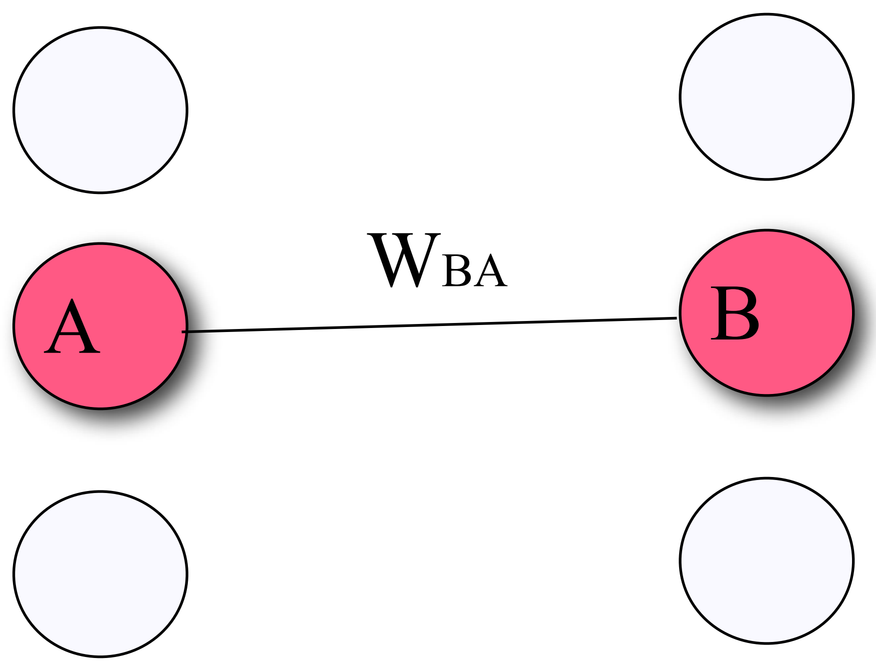
# Learning

Hebb Rule [Hebb 49] :



# Learning

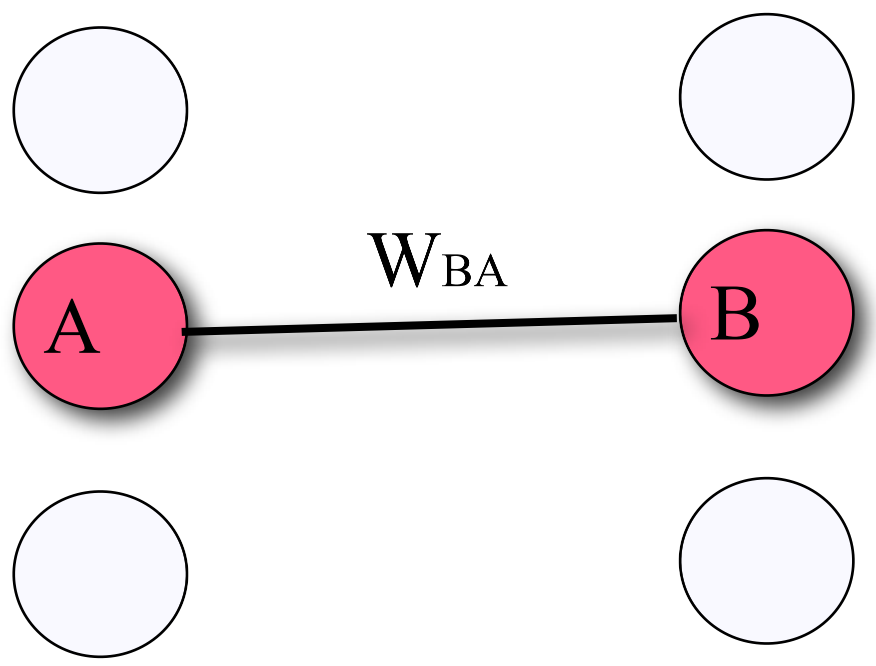
Hebb Rule [Hebb 49] :




For some reason... coactivation of A and B...

# Learning

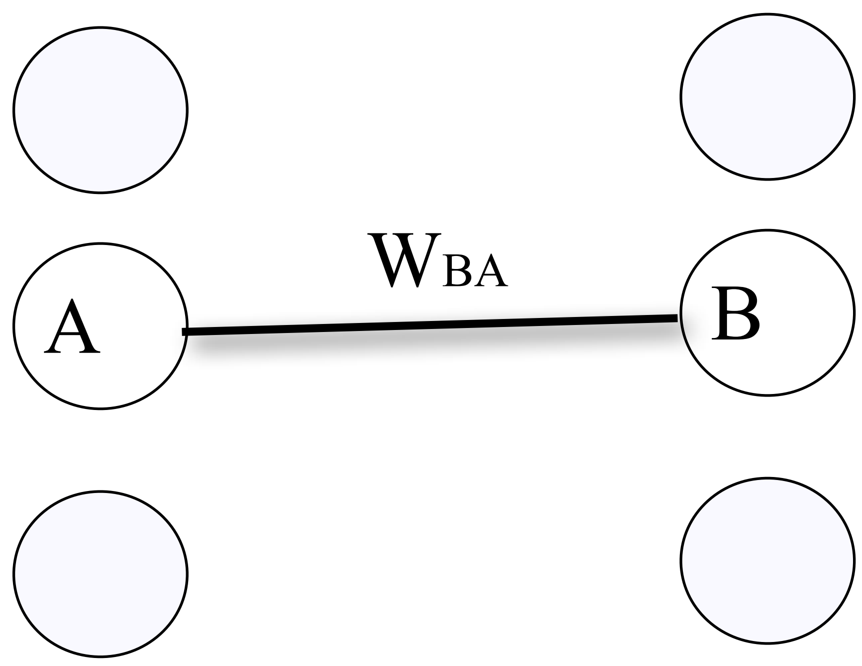
Hebb Rule [Hebb 49] :




For some reason... coactivation of A and B...repeatedly
increase of the connection $W_{BA}$

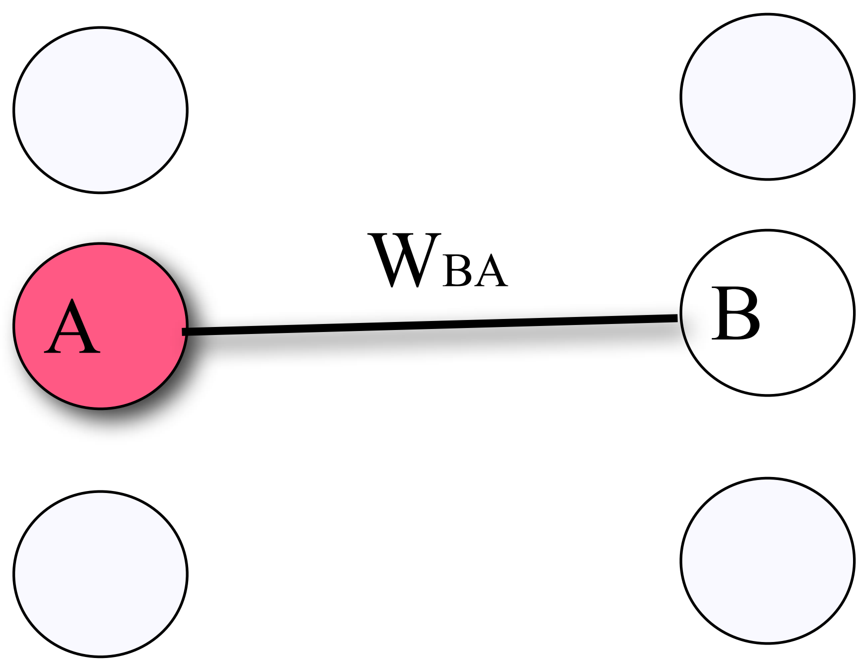
# Learning

Hebb Rule [Hebb 49] :



# Learning

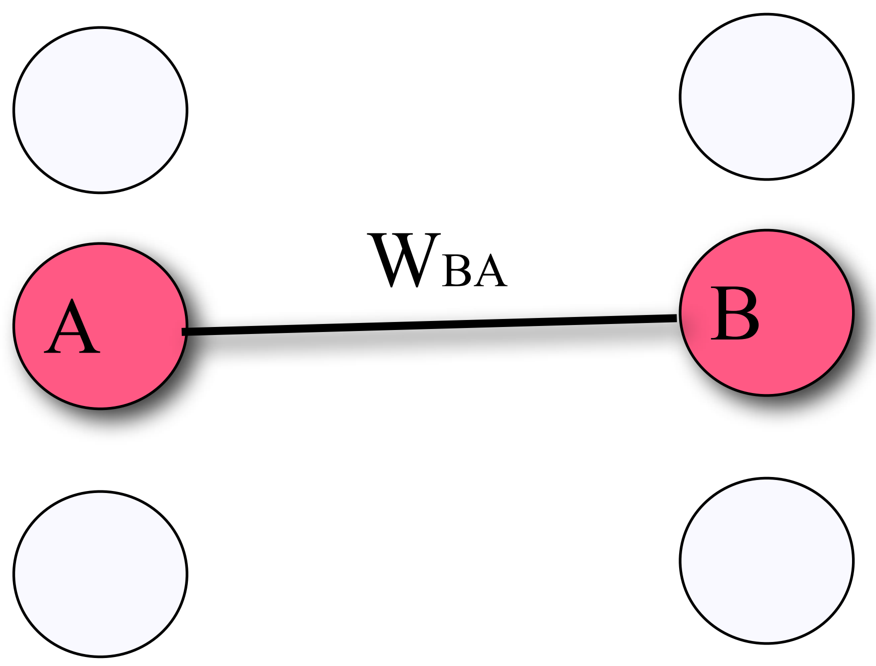
Hebb Rule [Hebb 49] :




The sole A activity is now enough ...

# Learning

Hebb Rule [Hebb 49] :




The sole A activity is not enough  to overshoot B threashold

$A*W_{BA} > \theta_B$

# Learning

- Hebb rule

$$W_{ij}\,(t+1) = w_{ij}\,(t) + eps.Y_j.Y_i$$

i.e :

$$\Delta W_{ij} = eps.Y_j.Y_i$$

with :

eps : learning speed

# Learning


UNIVERSITÉ
de Cergy-Pontoise
DGA
Training mobile robots to perform
path following by an intuitive
human–robot interaction
C. Giovannangeli
P. Gaussier
ETIS Laboratory
CNRS UMR 8051
Cergy–Pontoise University

# Overview

## Introduction : from biology to the formal neuron

**Part I : supervised learning**

- perceptron
    - simple rule
    - Widrow Hoff rule
    - limitations
- associative memories
- multi-layer perceptron
- backpropagation

**Part II : unsupervised learning**

- brain mechanisms
    - competition and cooperation
    - WTA
- Self Organizing Maps
    - Kohonen maps
    - K-means (analogy)
- Let’s put it all together
    - ART

# Supervised learning

- goal :
- make the network learn to categorize inputs
    - pattern recognition
    - class separation
- method :
    - train the network from different inputs
    - test the network generalization

# Supervised learning

- goal :
- make the network learn to categorize inputs
    - pattern recognition
    - class separation
- method :
    - train the network from different inputs
    - test the network generalization

# Supervised learning

The network will learn

- structural changes
- modification of Wij in order to obtain the correct output
- the expected result is a correct categorization
- learning is iterative: not too fast, not too slow
- learning rate epsilon
- test the generalization

# Supervised learning

How to guide the learning ?

- fundamental notion of **error**
- supervised : we expect a given answer
- at each time step, we calculate the error
- error : (desired ouput - output)
- while there is an error : we change the weights

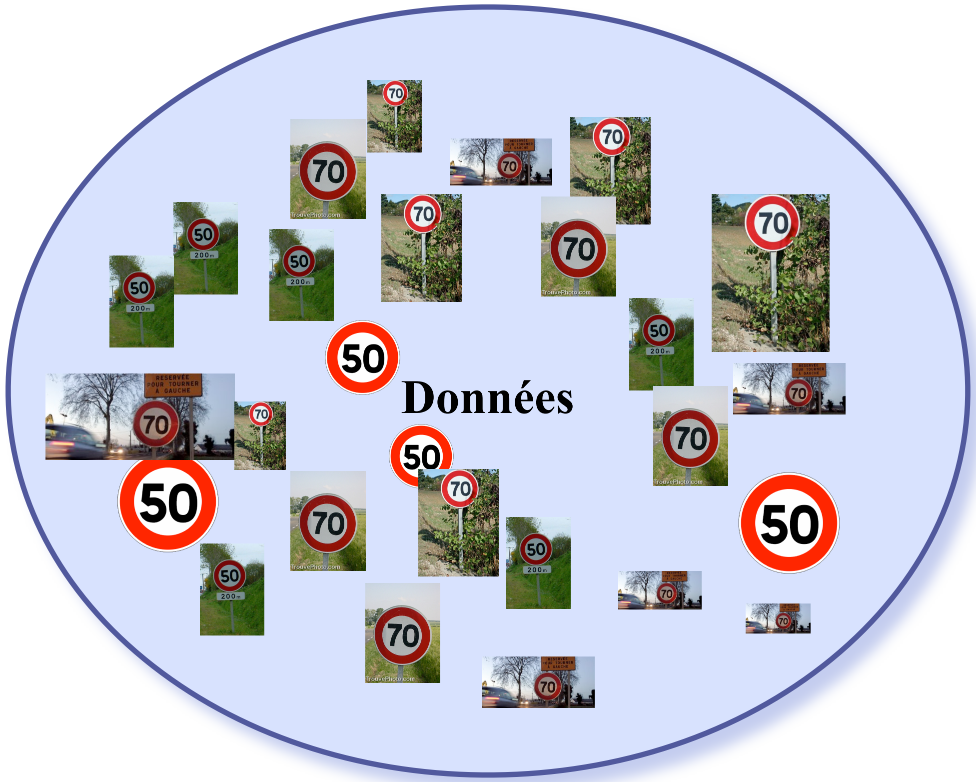
# Supervised learning

Example

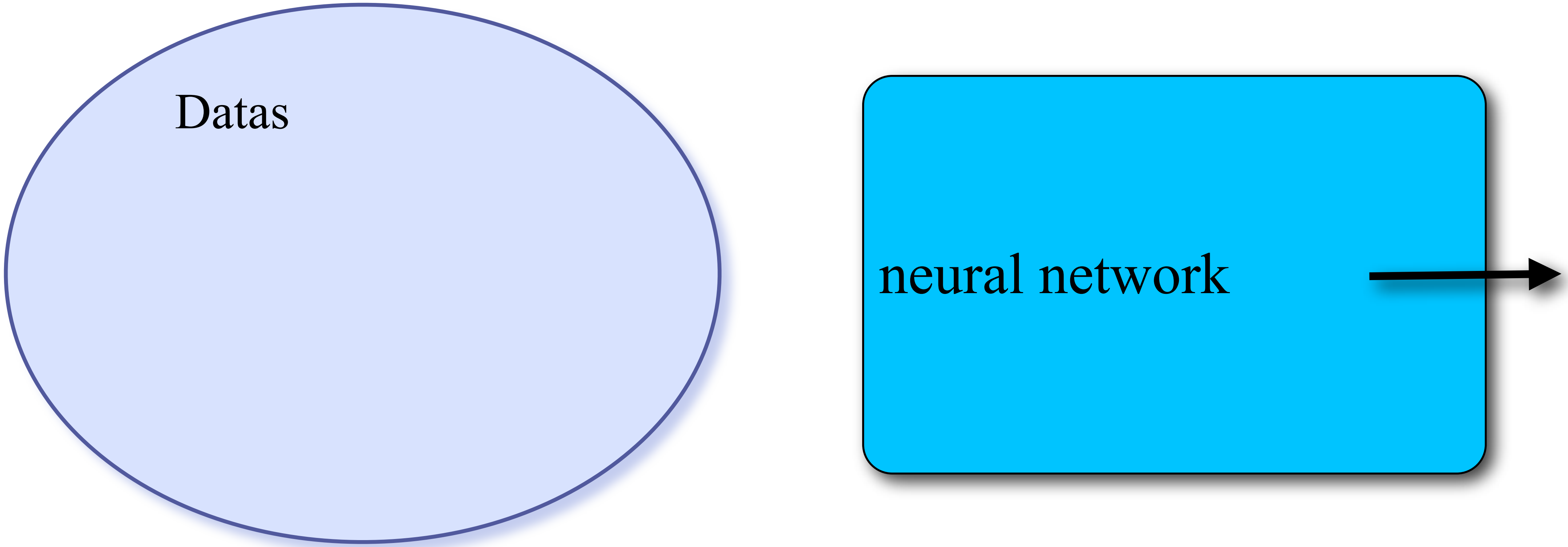




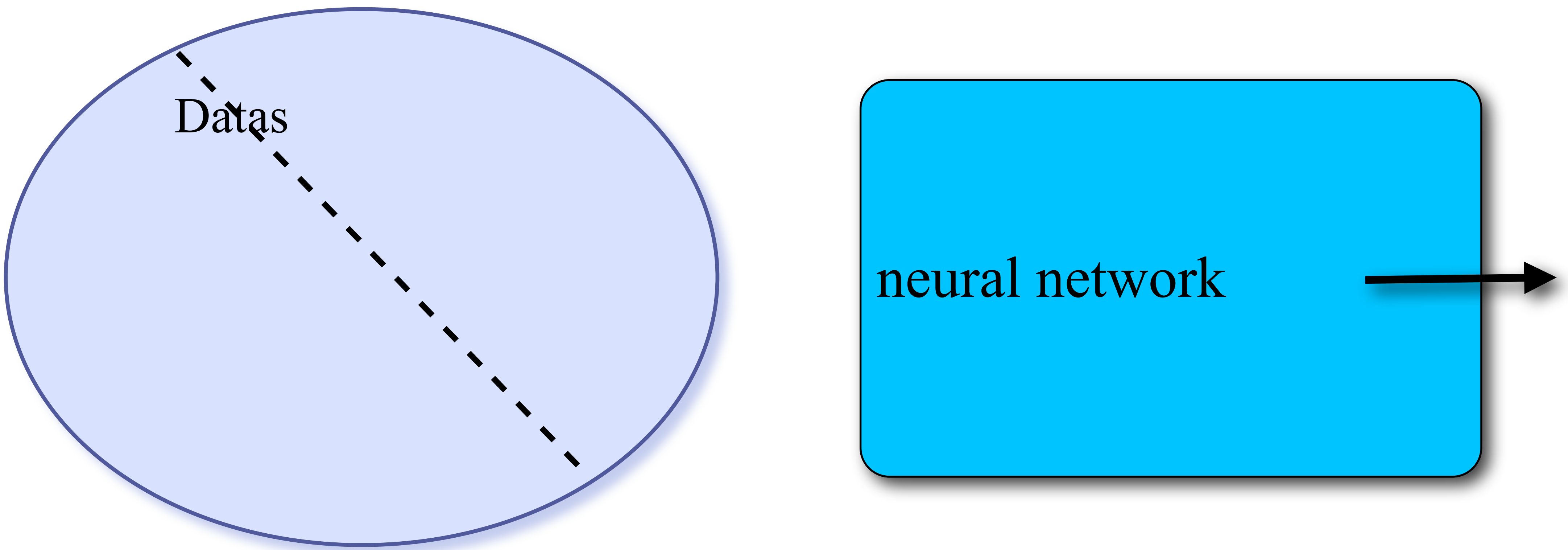
# Supervised learning



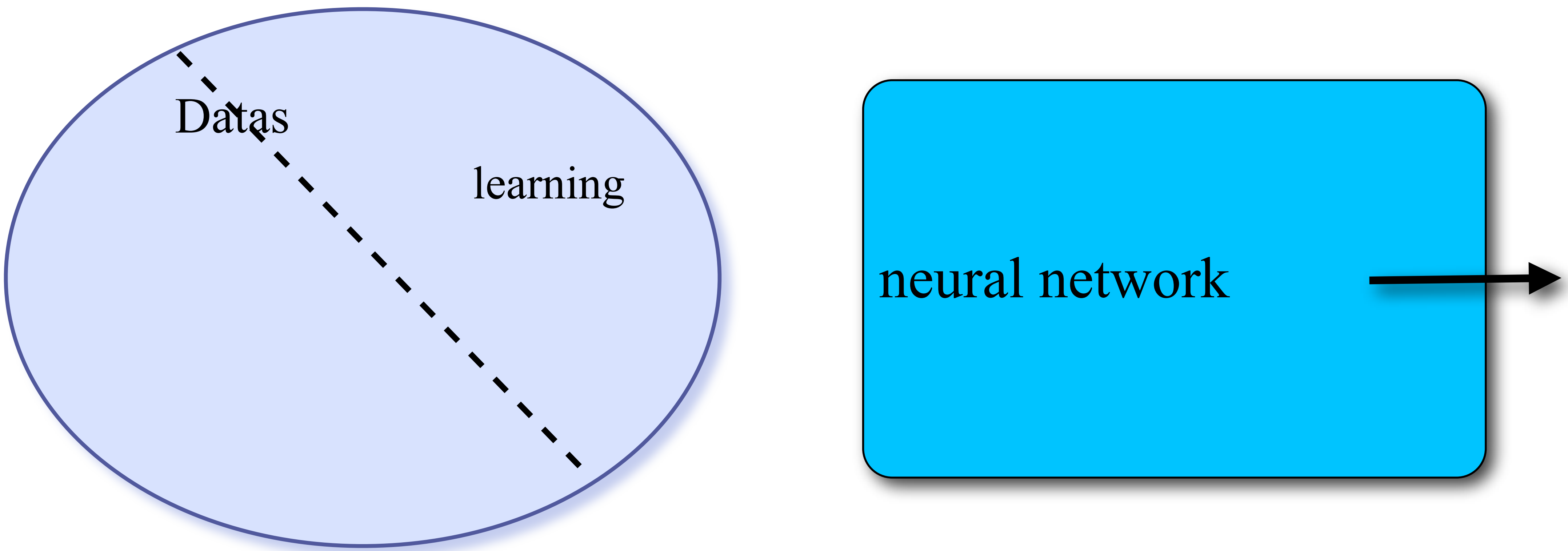
# Supervised learning



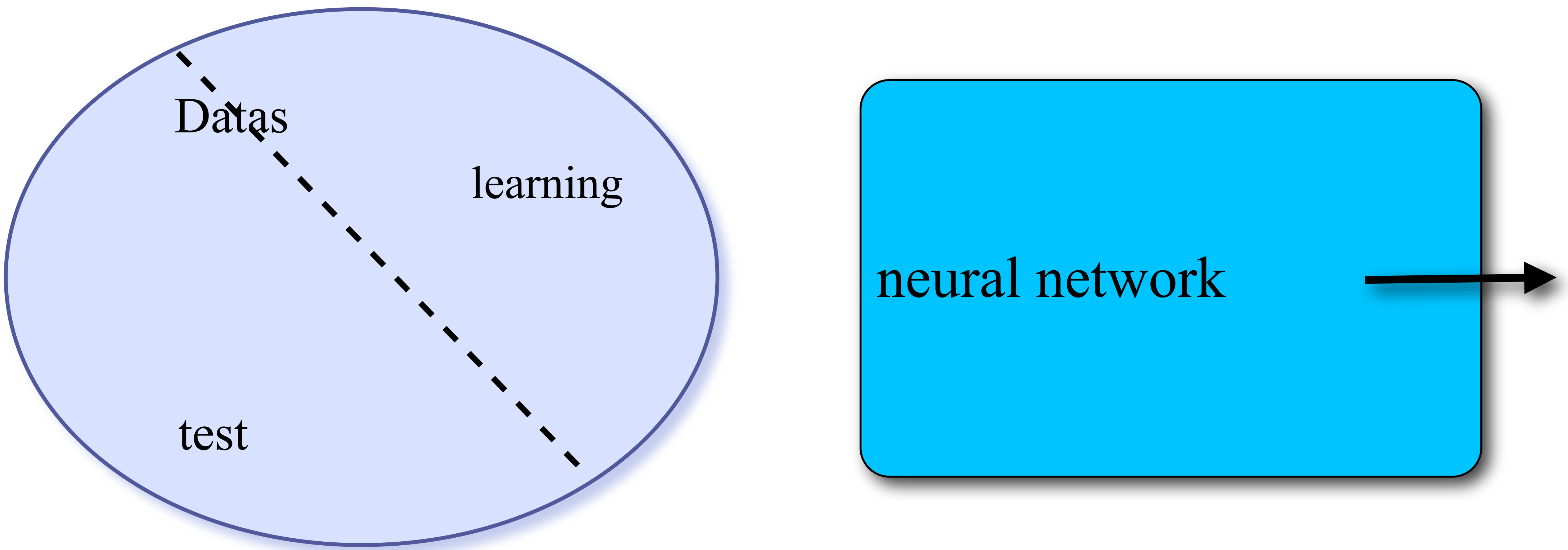
# Supervised learning



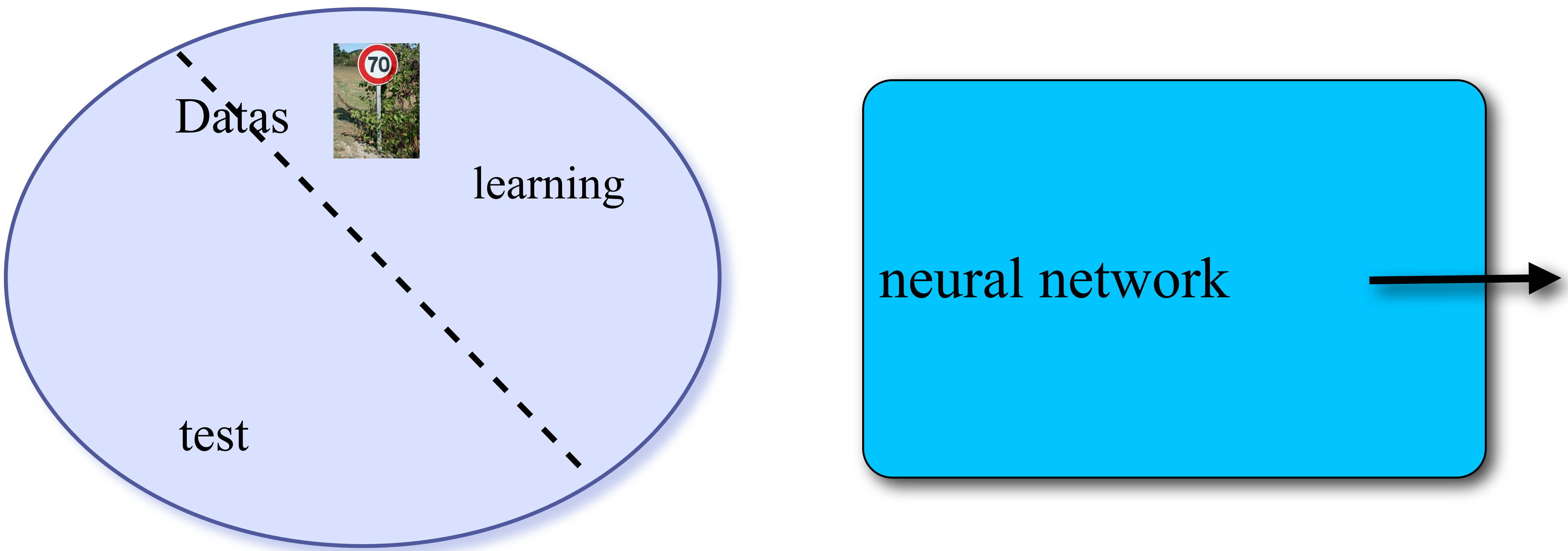
# Supervised learning



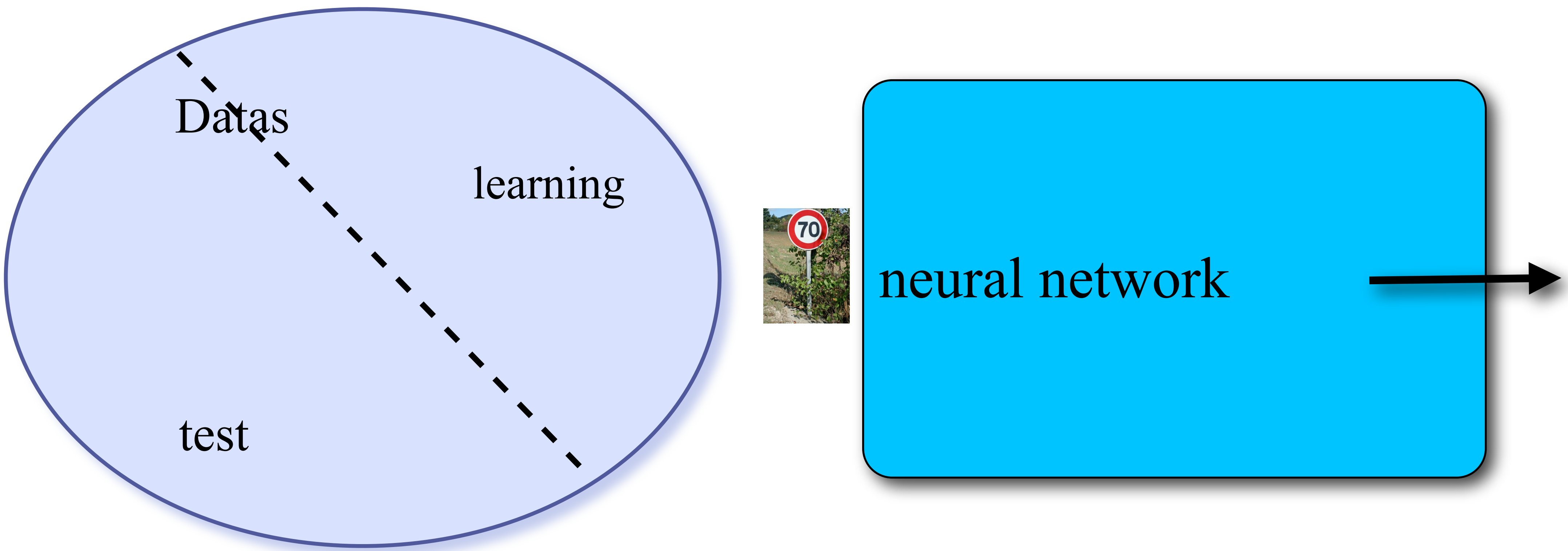
# Supervised learning



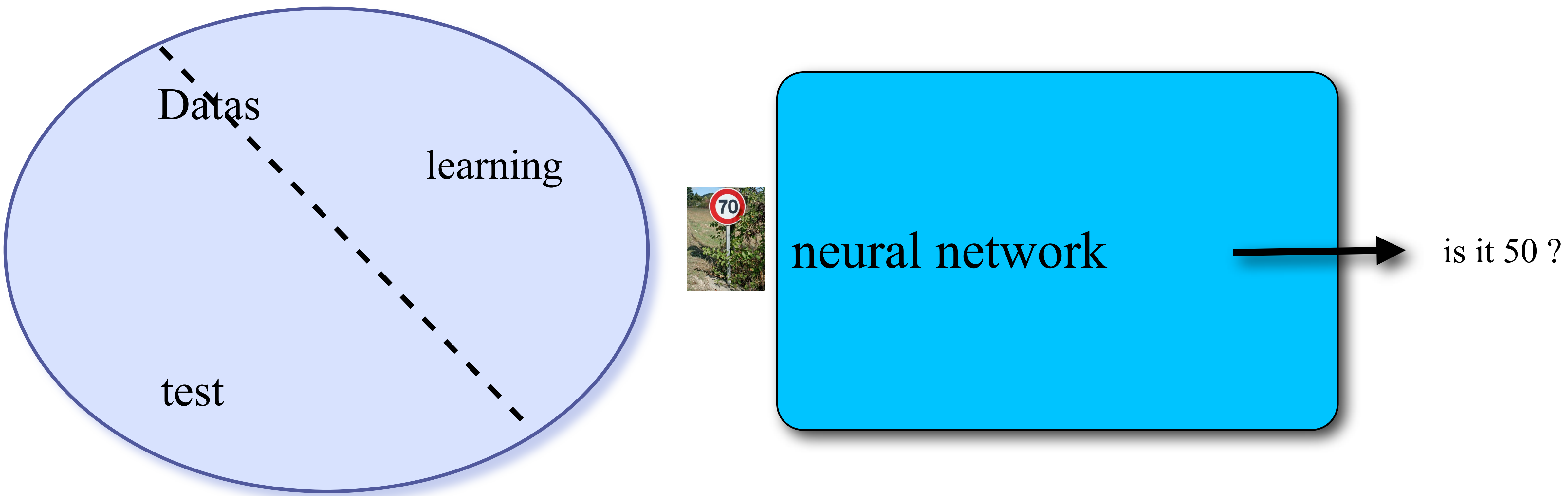
# Supervised learning



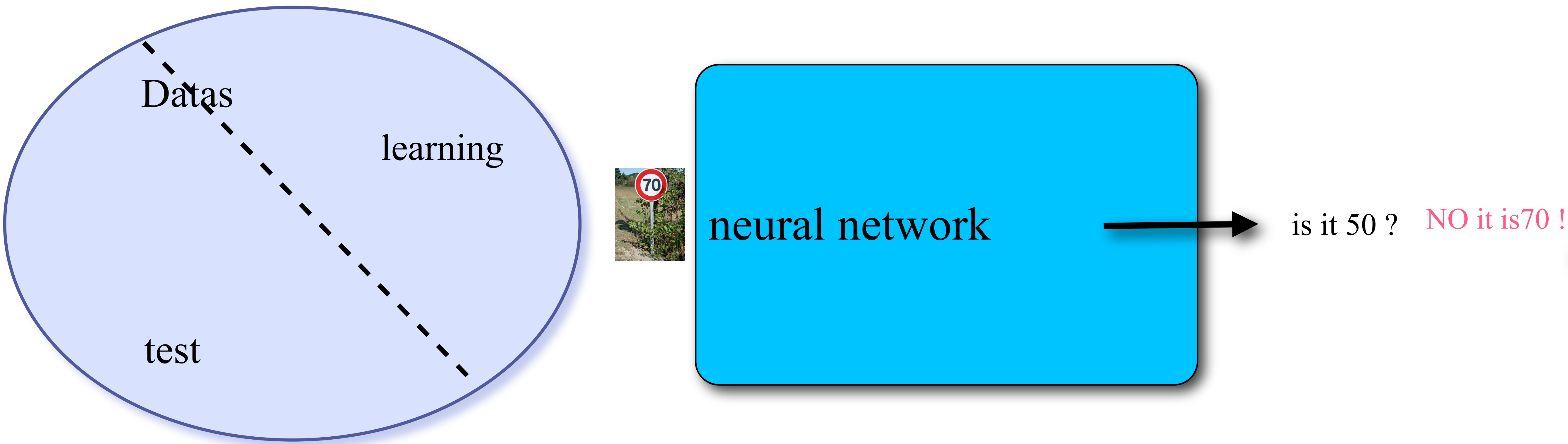
# Supervised learning



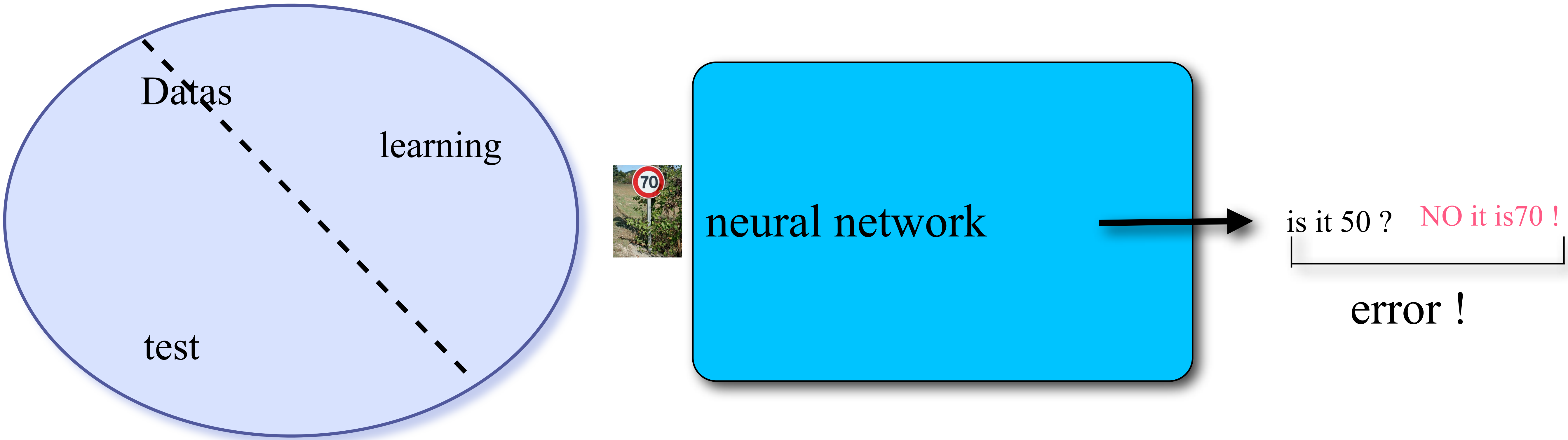
# Supervised learning



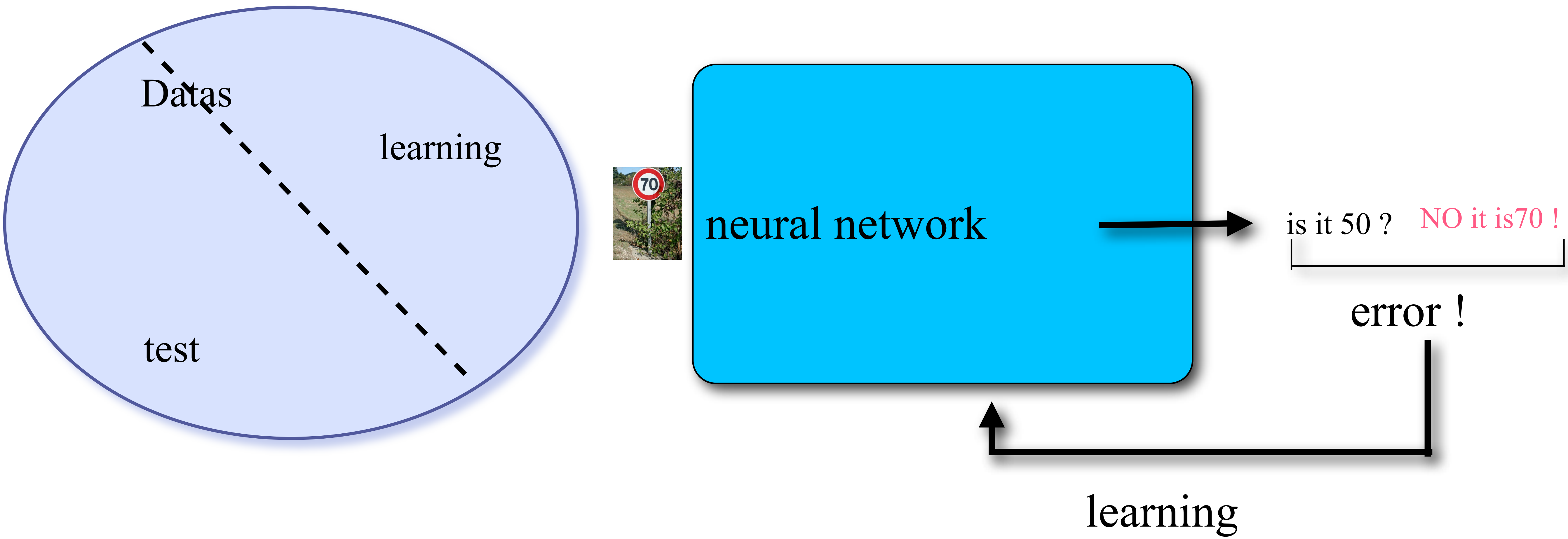
# Supervised learning



# Supervised learning



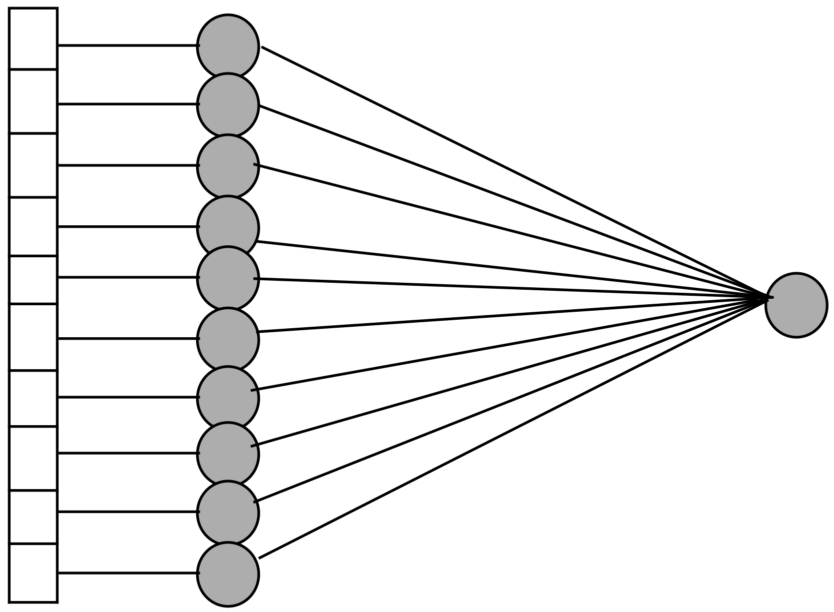
# Supervised learning



# Perceptron

## Architecture

- As input, the retina : raw numerical information
- A first layer of neurons : one-one connections with the rétina (normalization only)
- A last layer, called decision layer : the output of the system.



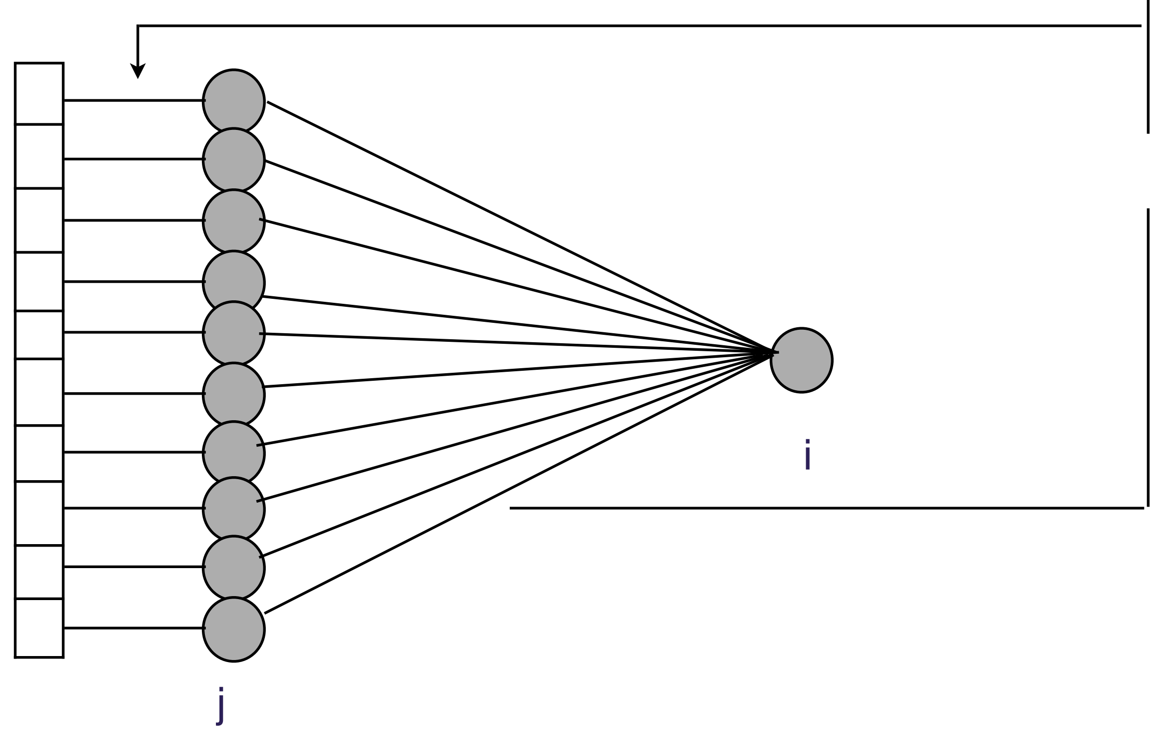
# Perceptron

## Architecture

- As input, the retina : raw numerical information

- A first layer of neurons : one-one connections with the rétina (normalization only)

- A last layer, called decision layer : the output of the system.




- non-modifiable weight. direct transmission, W=1/(raw-max)

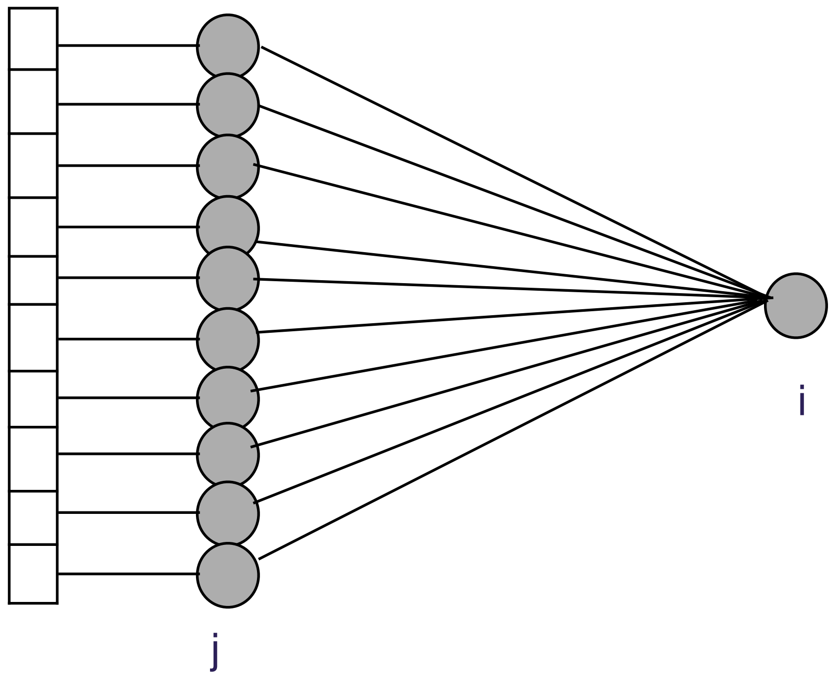
- modifiable weights. learning !

- Wij= random between [-1/n,1/n]

# Perceptron

Learning : the simple rule

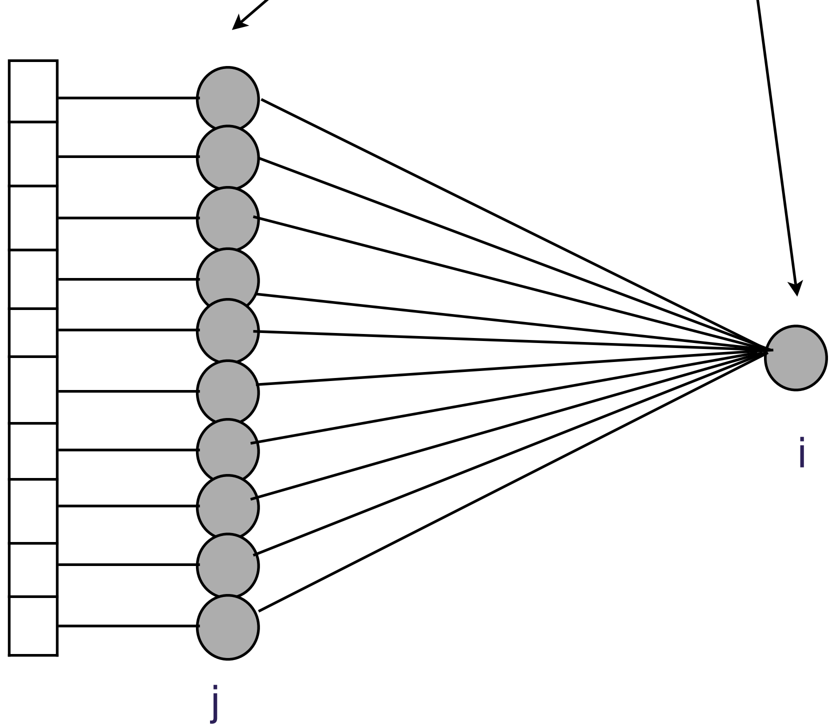
- $w_{ij}(t+i) = w_{ij}(t) + eps.(Y_d - Y_i).X_j$



# Perceptron

Learning : the simple rule

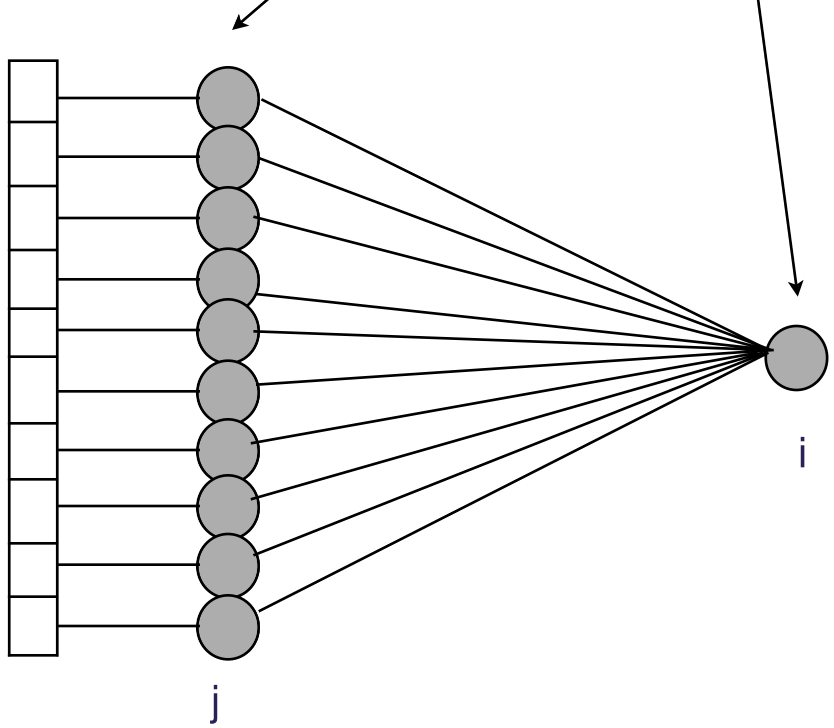
- $w_{ij}(t+i) = w_{ij}(t) + eps.(Y_d - Y_i).Y_j$



# Perceptron

Learning : the simple rule

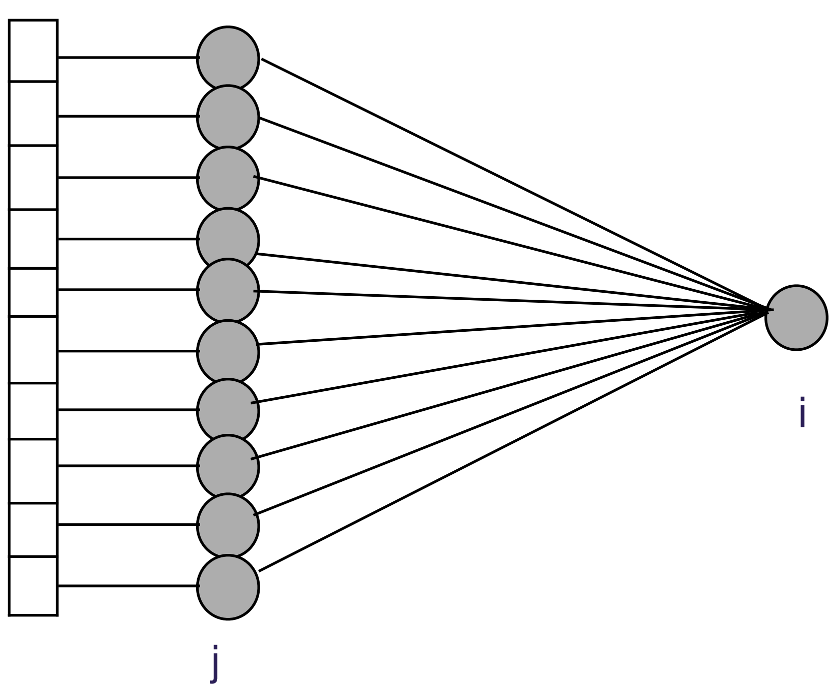
- $w_{ij}(t+i) = w_{ij}(t) + eps.(Y_d - Y_i).Y_j$



# Perceptron

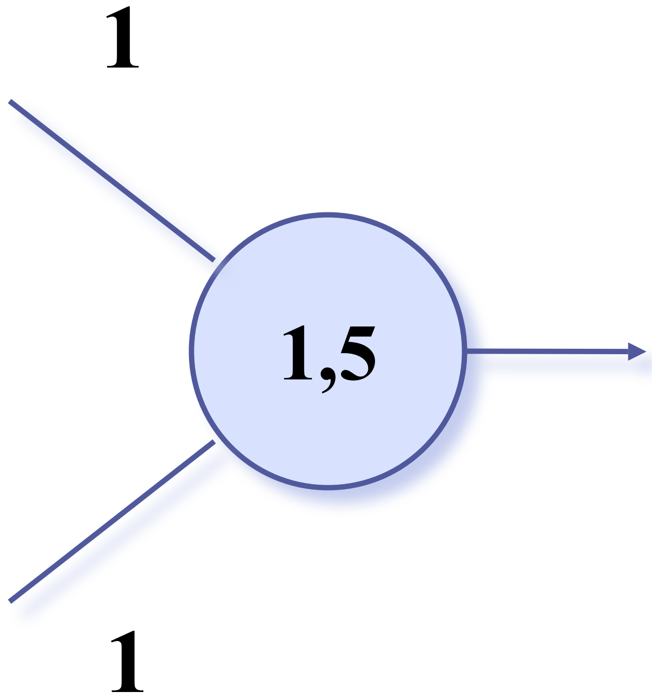
Learning : the simple rule

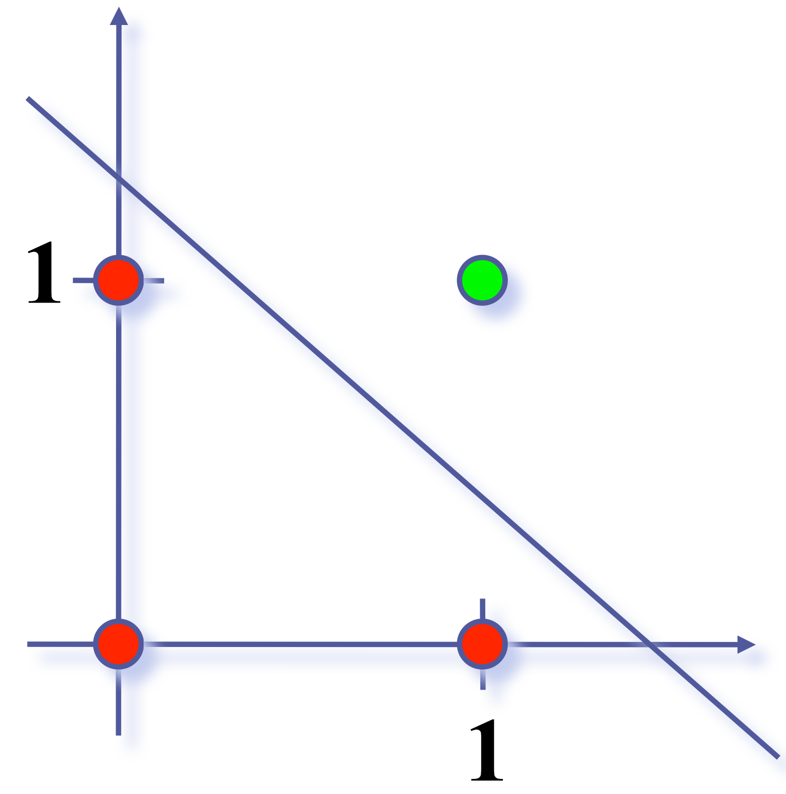
$$w_{ij}(t+i) = w_{ij}(t) + eps.\underbrace{(Y_d - Y_i)}_{\text{error}}.Y_j$$



# Perceptron

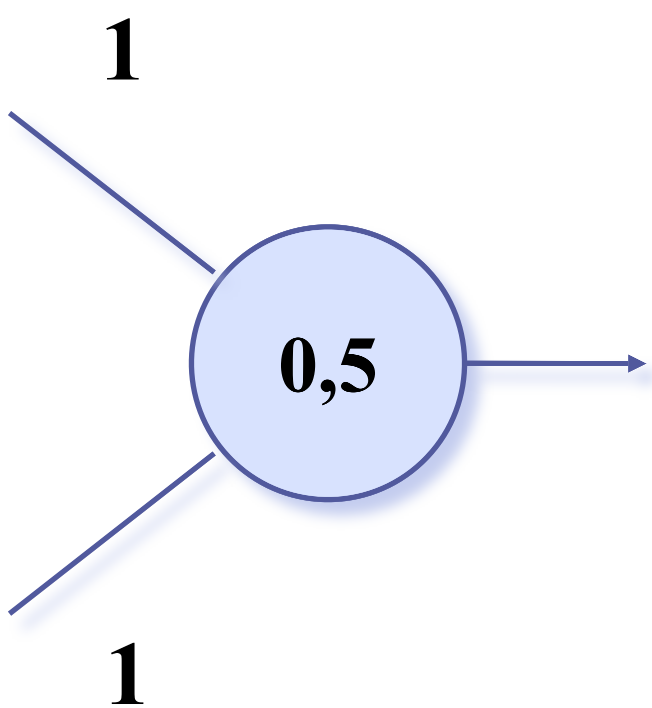
- logical AND

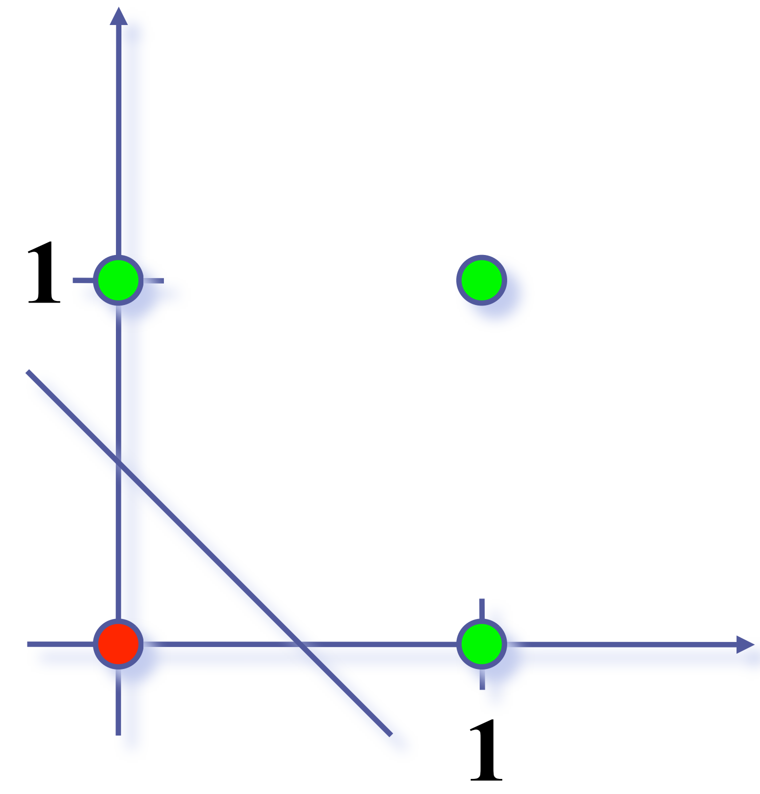


# Perceptron

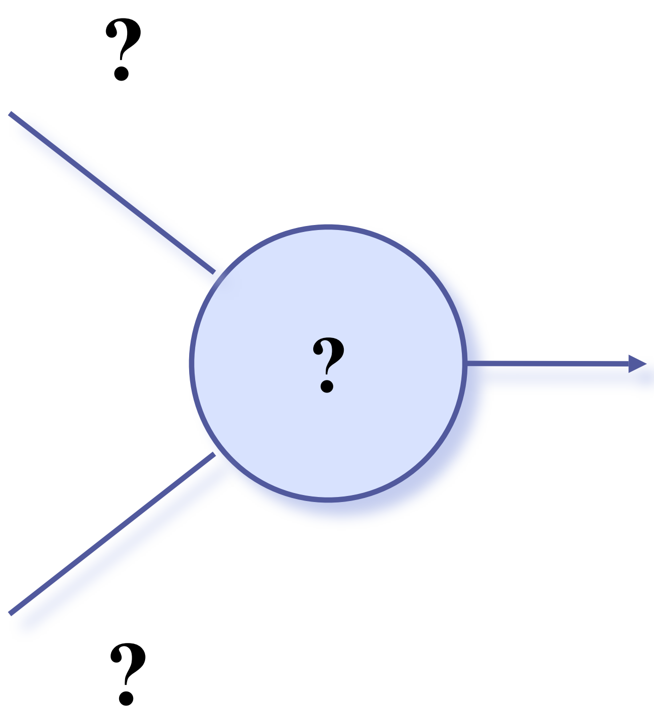
- logical OR

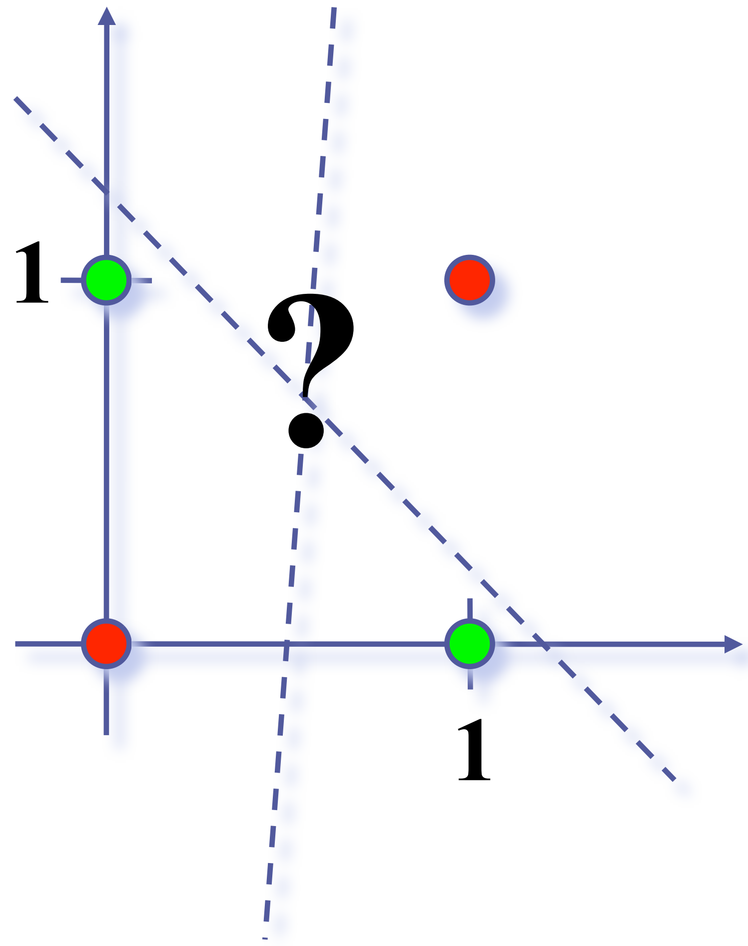


# Perceptron

- XOR ?



# Perceptron

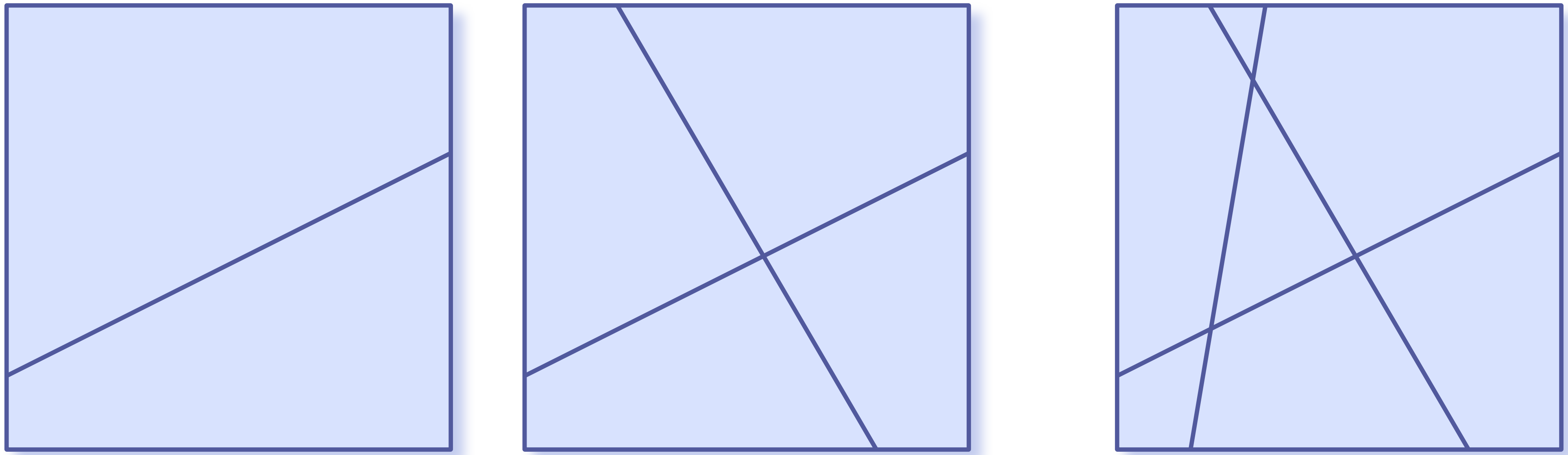
- limited to linear separation

- **what to ?**
  - increase the number of layers ad combine the outputs

# Perceptron

- limited to linear separation

- **what to ?**
  - increase the number of layers ad combine the outputs



**1 layer** **2 layers** **3 layers** …