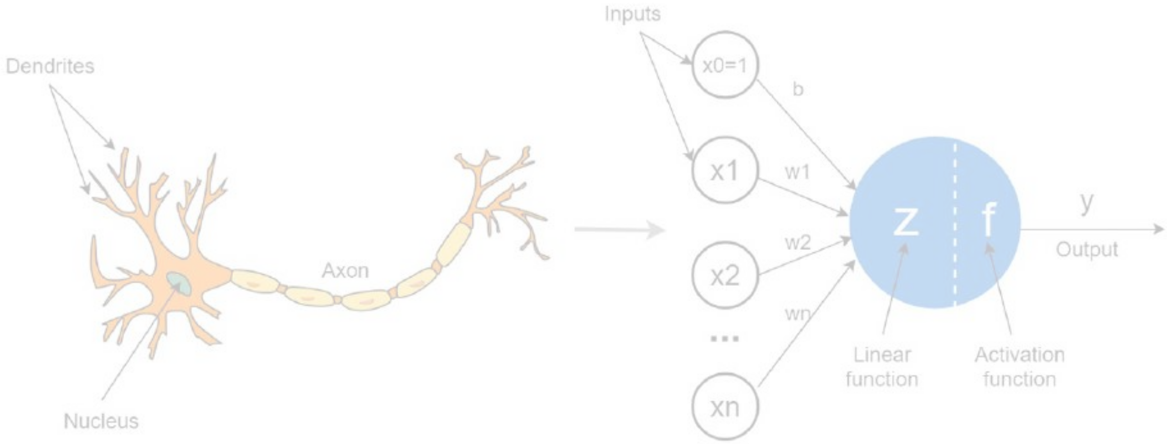


Binary neurons and networks

Olesia Dogonasheva
odogonasheva@hse.ru

Class outline

1. Biological neurons
2. Binary neurons
    a. The perceptron learning rule
    b. Limitations
3. Associative networks
    a. Attractors
    b. Hopfield learning rule
4. Summary

Class outline

1. **Biological neurons**
2. Binary neurons
    a. The perceptron learning rule
    b. Limitations
3. Associative networks
    a. Attractors
    b. Hopfield learning rule
4. Summary

What does the hardware look like?

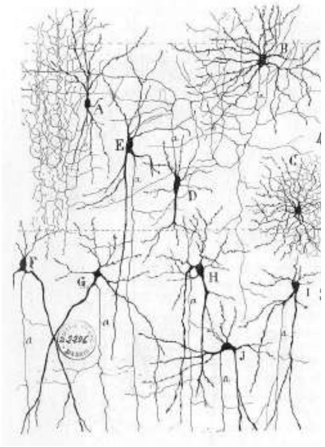
1. Reticular theory (up to 1900)
    - «Protoplasmic reticulum»

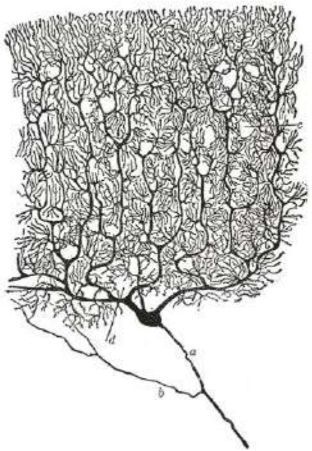
Joseph von Gerlach

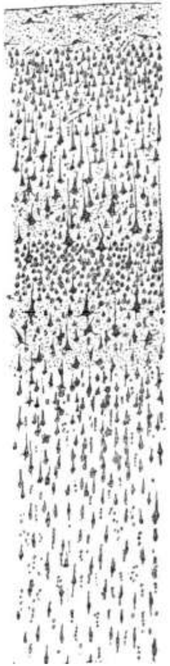
Camillo Golgi

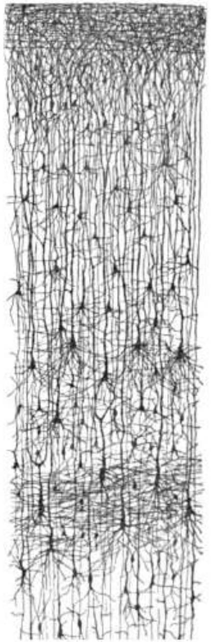
What does the hardware look like?

1. Reticular theory (up to 1900)
   - ○ «Protoplasmic reticulum»
2. Neuron doctrine









Santiago Ramón y Cajal

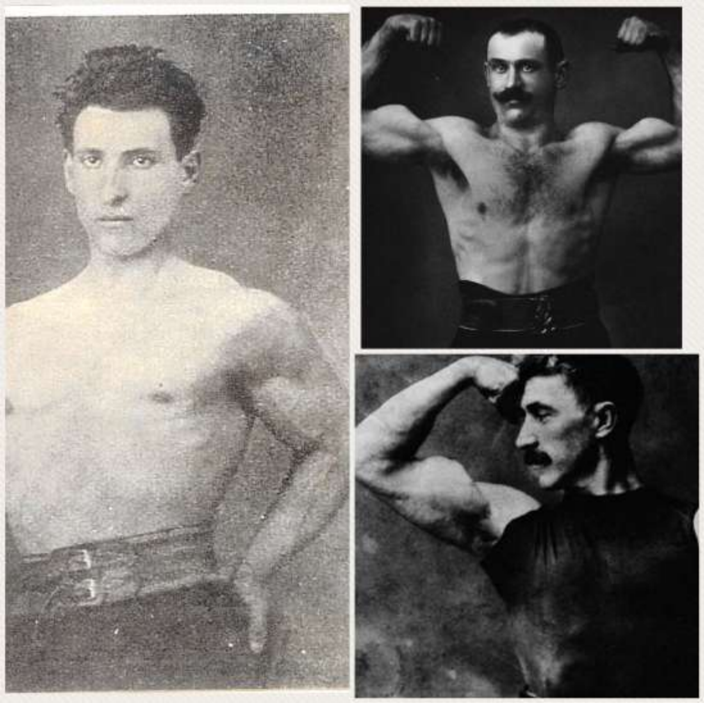
What does the hardware look like?

1. Reticular theory (up to 1900)
   - «Protoplasmic reticulum»
2. Neuron doctrine
   - Neural units
   - Neurons are cells
   - Specialization
   - Nucleus is key
   - Nerve fibers are cell processes
   - Cell division
   - Nerve cells are connected by sites of contact and not cytoplasmic continuity.
   - Law of dynamic polarization
   - Synapse
   - Unity of transmission
   - Dale's law

Santiago Ramón y Cajal

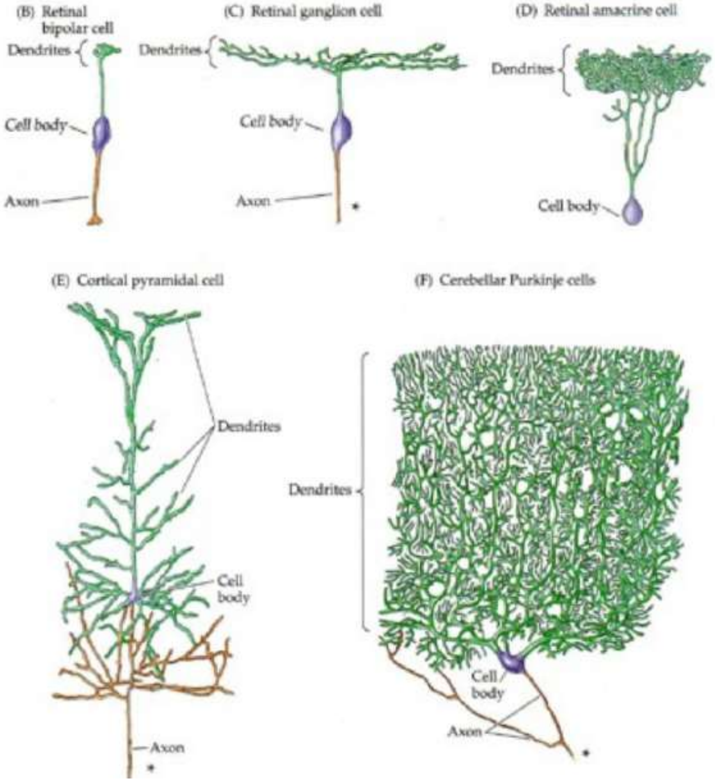
What does the hardware look like?

1. Reticular theory (up to 1900)
    - «Protoplasmic reticulum»
2. Neuron doctrine
    - Neural units
    - Neurons are cells
    - Specialization
    - Nucleus is key
    - Nerve fibers are cell processes
    - Cell division
    - Nerve cells are connected by sites of contact and not cytoplasmic continuity.
    - Law of dynamic polarization
    - Synapse
    - Unity of transmission
    - Dale's law



Santiago Ramón y Cajal

Neurons = basic units of computation

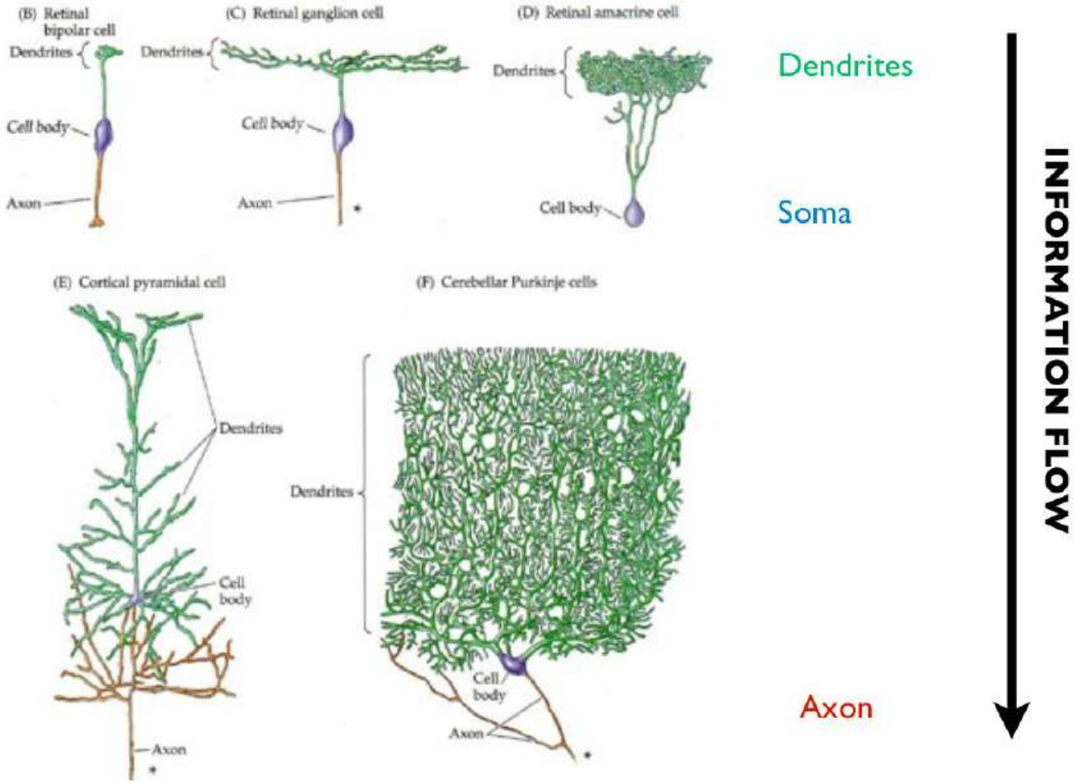


Dendrites

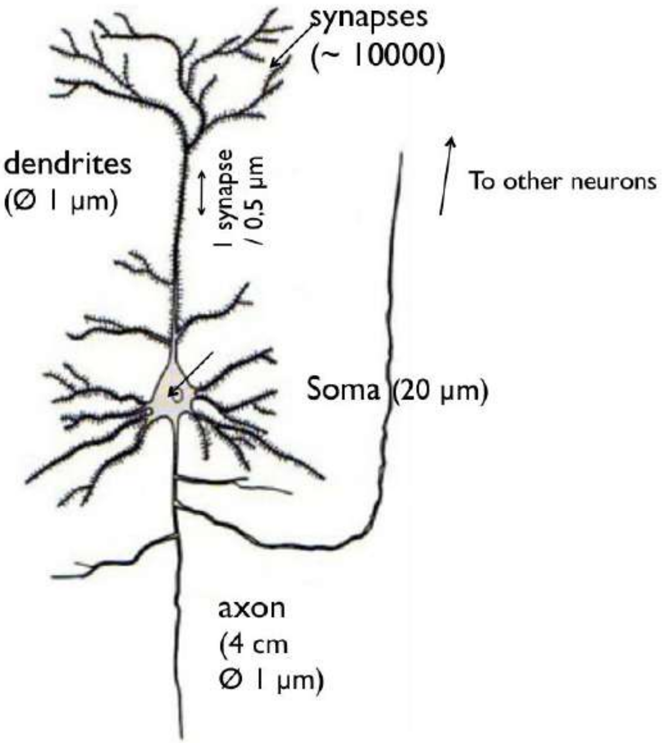
Soma

Axon

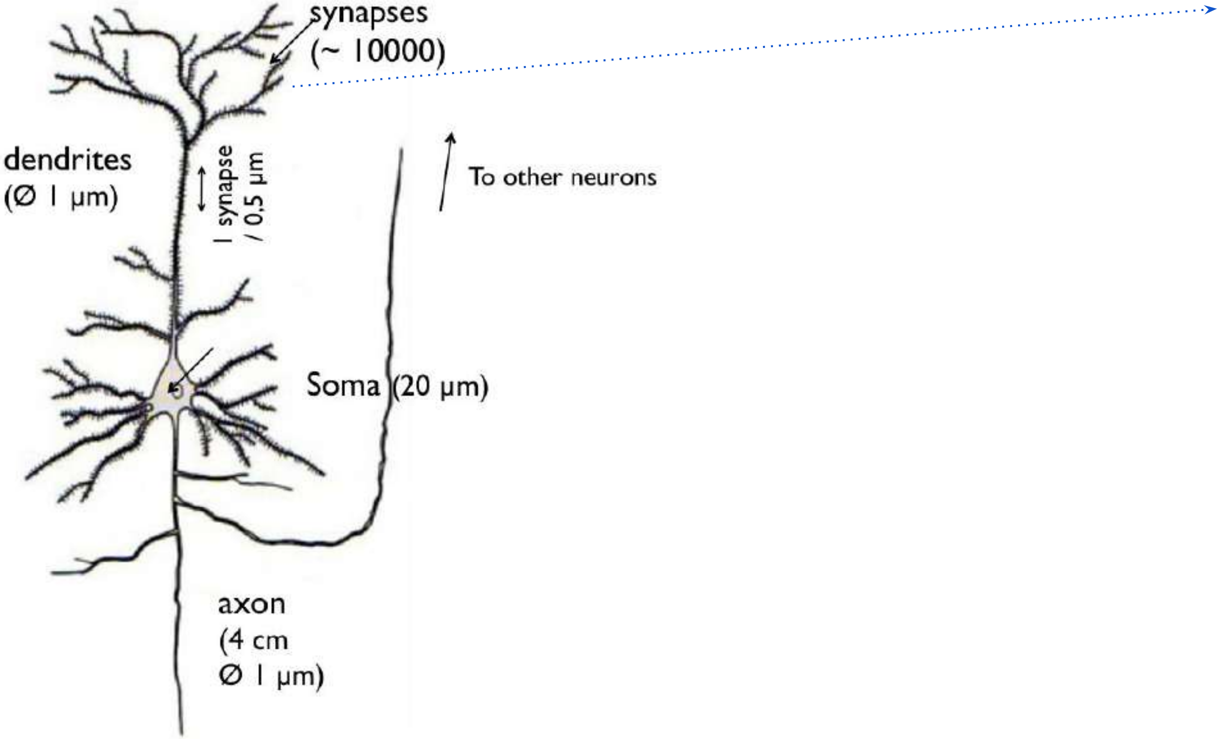
Neurons = basic units of computation

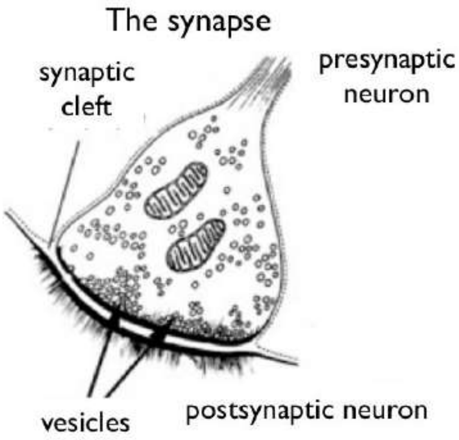


The Typical Cortical Neuron

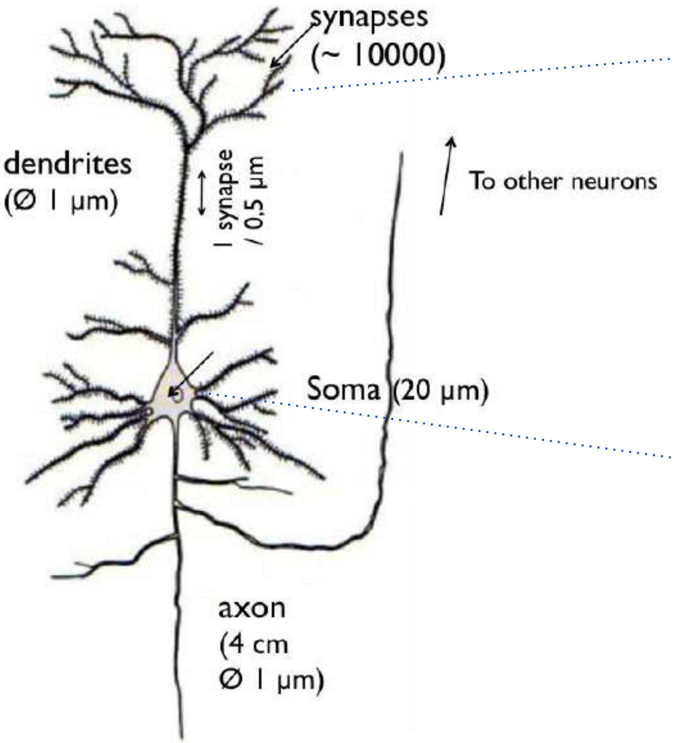


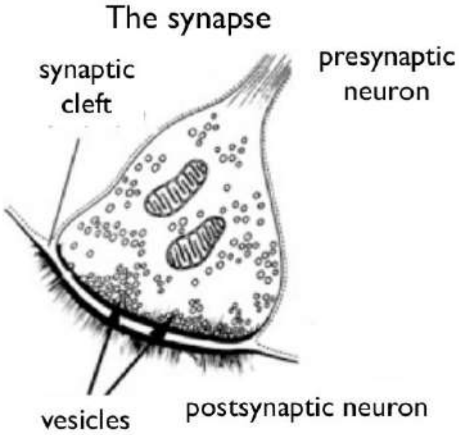
The Typical Cortical Neuron

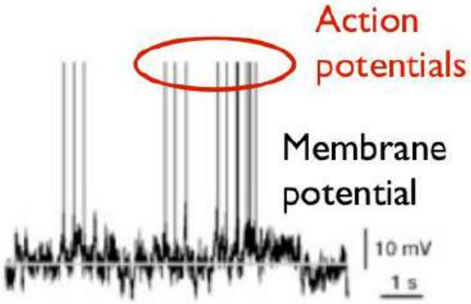




The Typical Cortical Neuron







Class outline

1. Biological neurons
2. **Binary neurons**
   a. The perceptron learning rule
   b. Limitations
3. Associative networks
   a. Attractors
   b. Hopfield learning rule
4. Summary

Brain = digital machine

1930-1950: birth of first computers

- **Shannon**: information theory of digital signals
- **Turing** : universal capabilities of digital machines
- **Von Neumann**: architecture of universal computers

Can we construct an electronic brain?
Birth of *Artificial Intelligence*

Claude Shannon

Alan Turing

John von Neumann

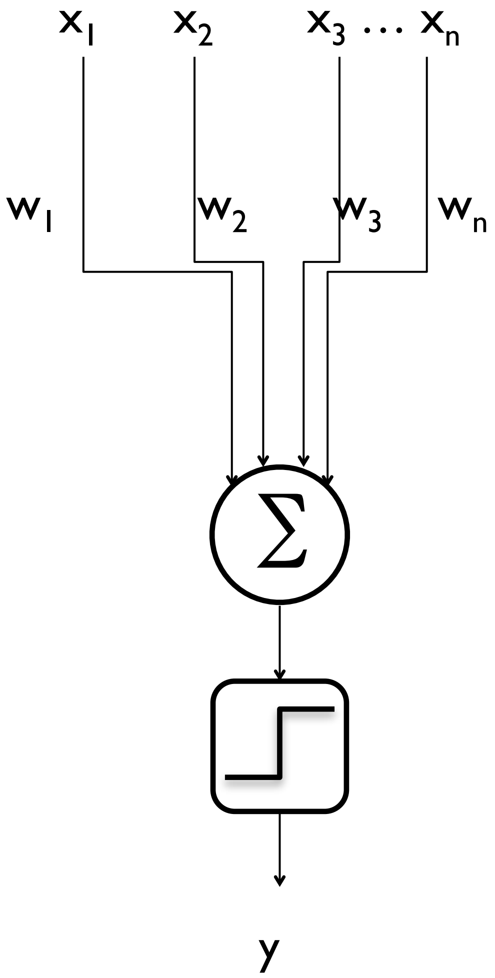
Revolution of psychology and Cognitive Science

“Symposium on Information Theory” MIT (September 11, 1956)

Experimental psychology + Information theory + theoretical linguistic

- ❖ G. Miller (1956) “The Magical Number Seven, Plus or Minus Two”
- ❖ N. Chomsky (1957) “Syntactic Structures”
- ❖ B.F. Skinner (1959) “Verbal Behavior”
- ❖ Джон Маккарти, Марвин Мински, Аллен Ньюэлл и Герберт Саймон

The Binary Neuron



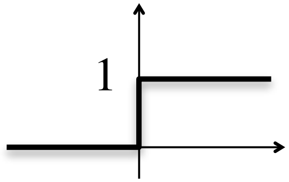
Synaptic Inputs

Synaptic weights

Summation

$$\sum_{k=1}^{N} w_k x_k$$

Threshold

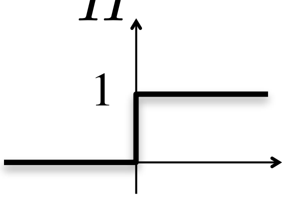


Output

$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right)$$

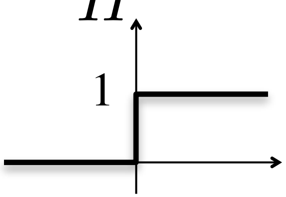
McCulloch and Pitts (1943)

$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

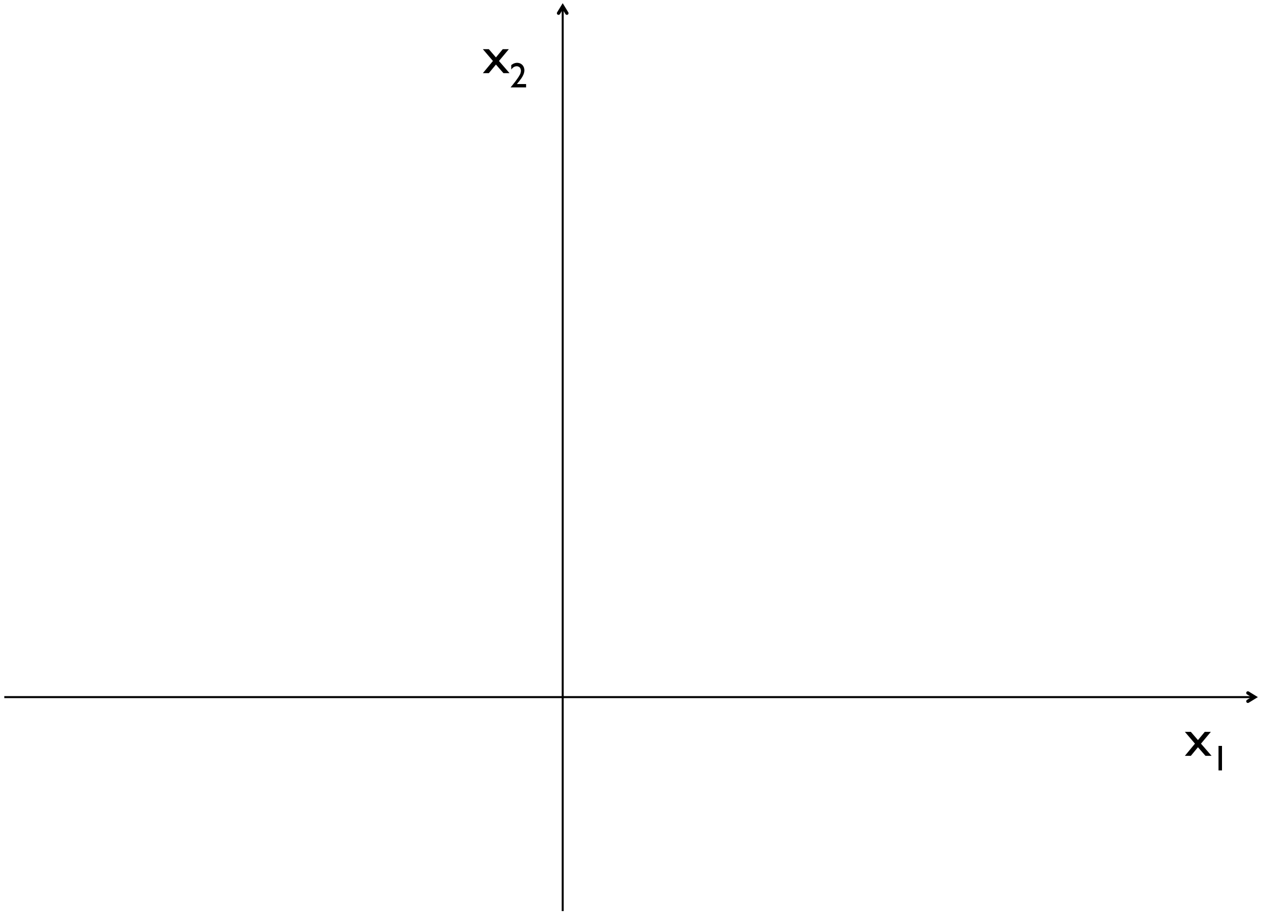


Two synaptic inputs $\vec{x} = (x_1, x_2)$ $\vec{w} = (w_1, w_2)$

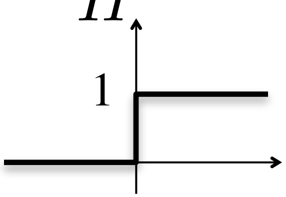
$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$



Two synaptic inputs $\vec{x} = (x_1, x_2)$ $\vec{w} = (w_1, w_2)$

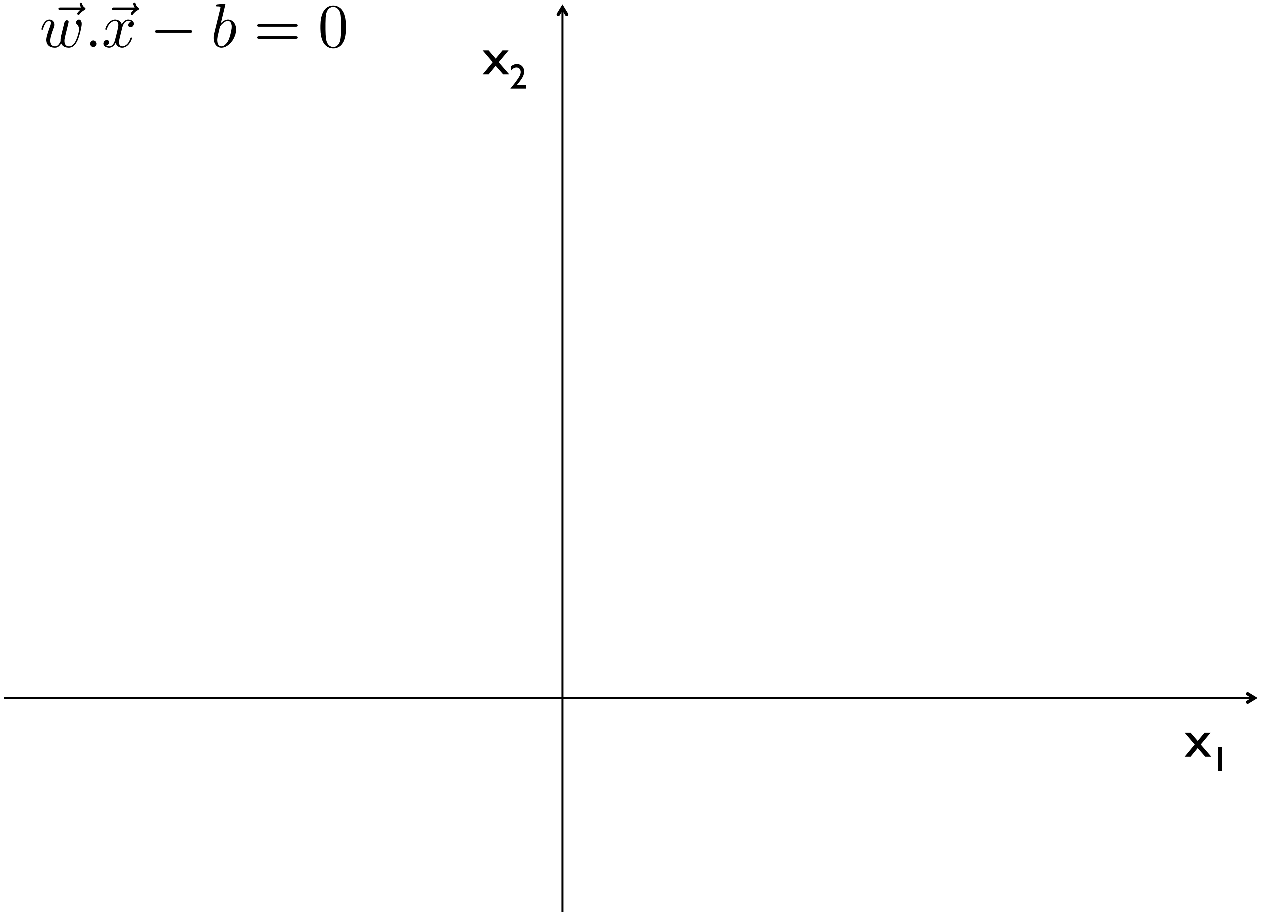


$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

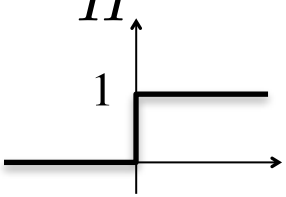


Two synaptic inputs $\vec{x} = (x_1, x_2)$ $\vec{w} = (w_1, w_2)$

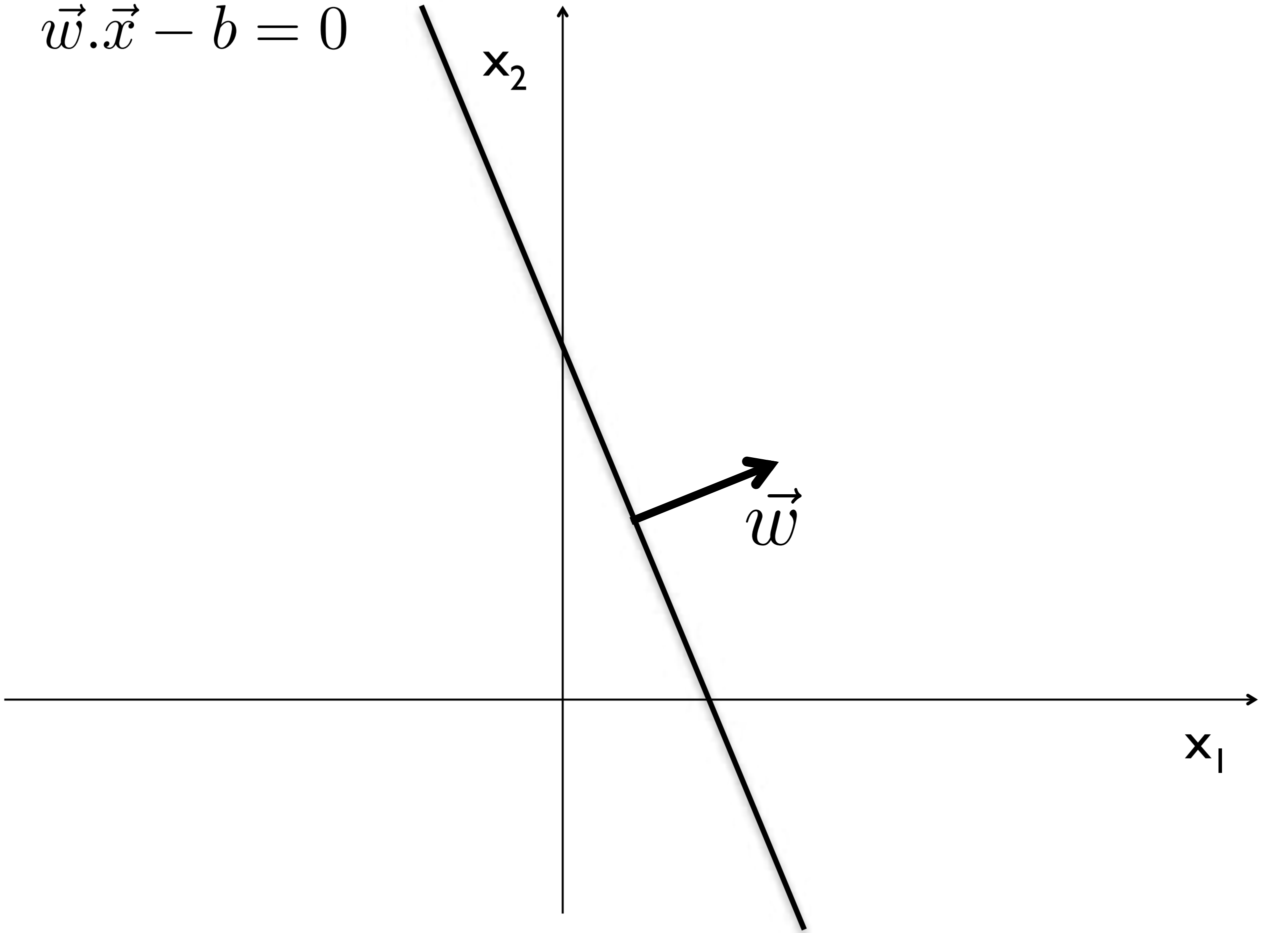
$$\vec{w}.\vec{x} - b = 0$$



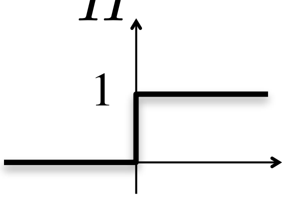
$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$



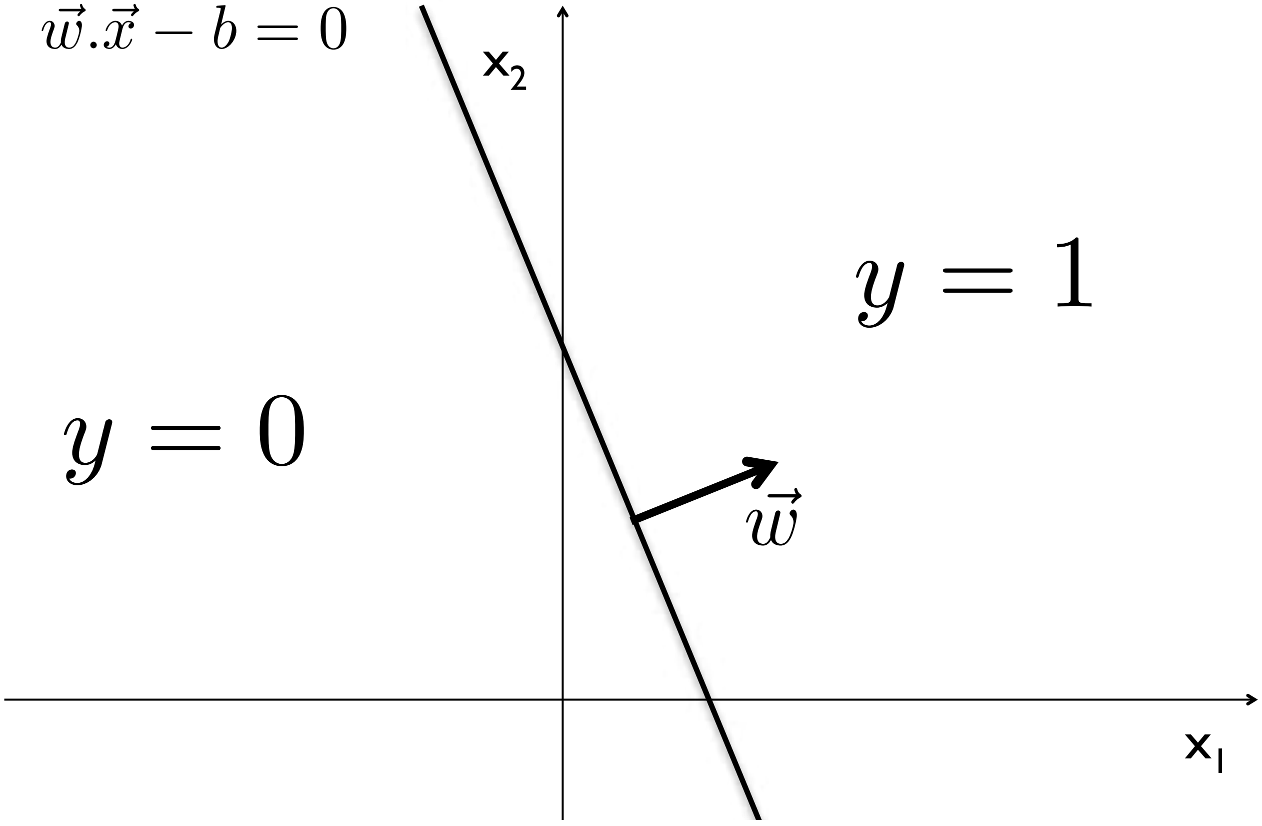
Two synaptic inputs $\vec{x} = (x_1, x_2)$ $\vec{w} = (w_1, w_2)$



$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

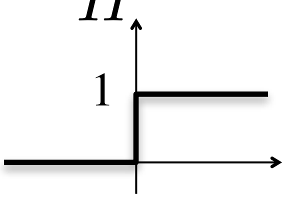


Two synaptic inputs $\vec{x} = (x_1, x_2)$ $\vec{w} = (w_1, w_2)$



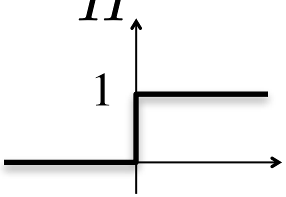
→ Separates the plane in two regions

$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

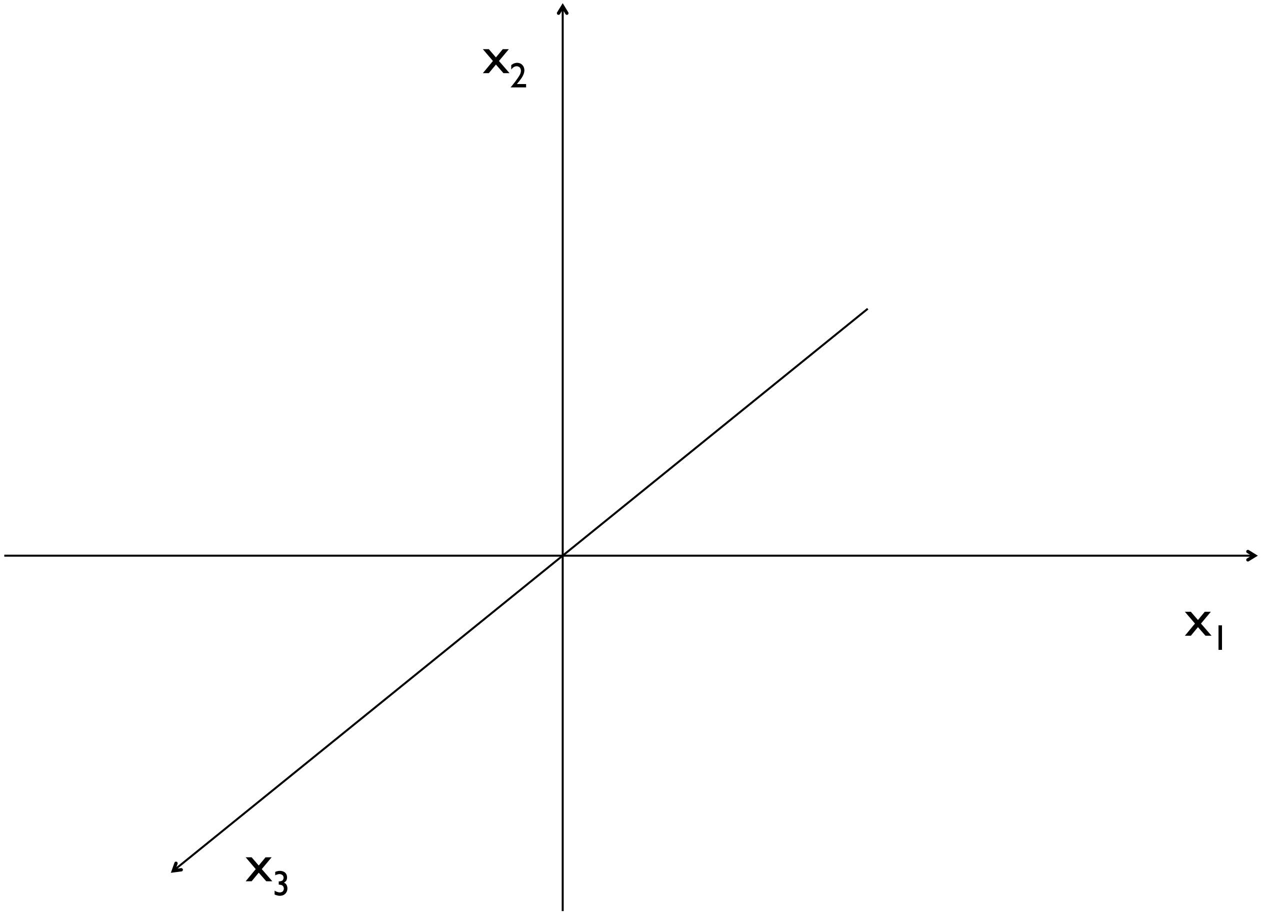


Three synaptic inputs $\vec{x} = (x_1, x_2, x_3)$ $\vec{w} = (w_1, w_2, w_3)$

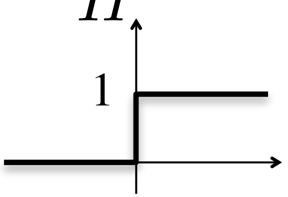
$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$



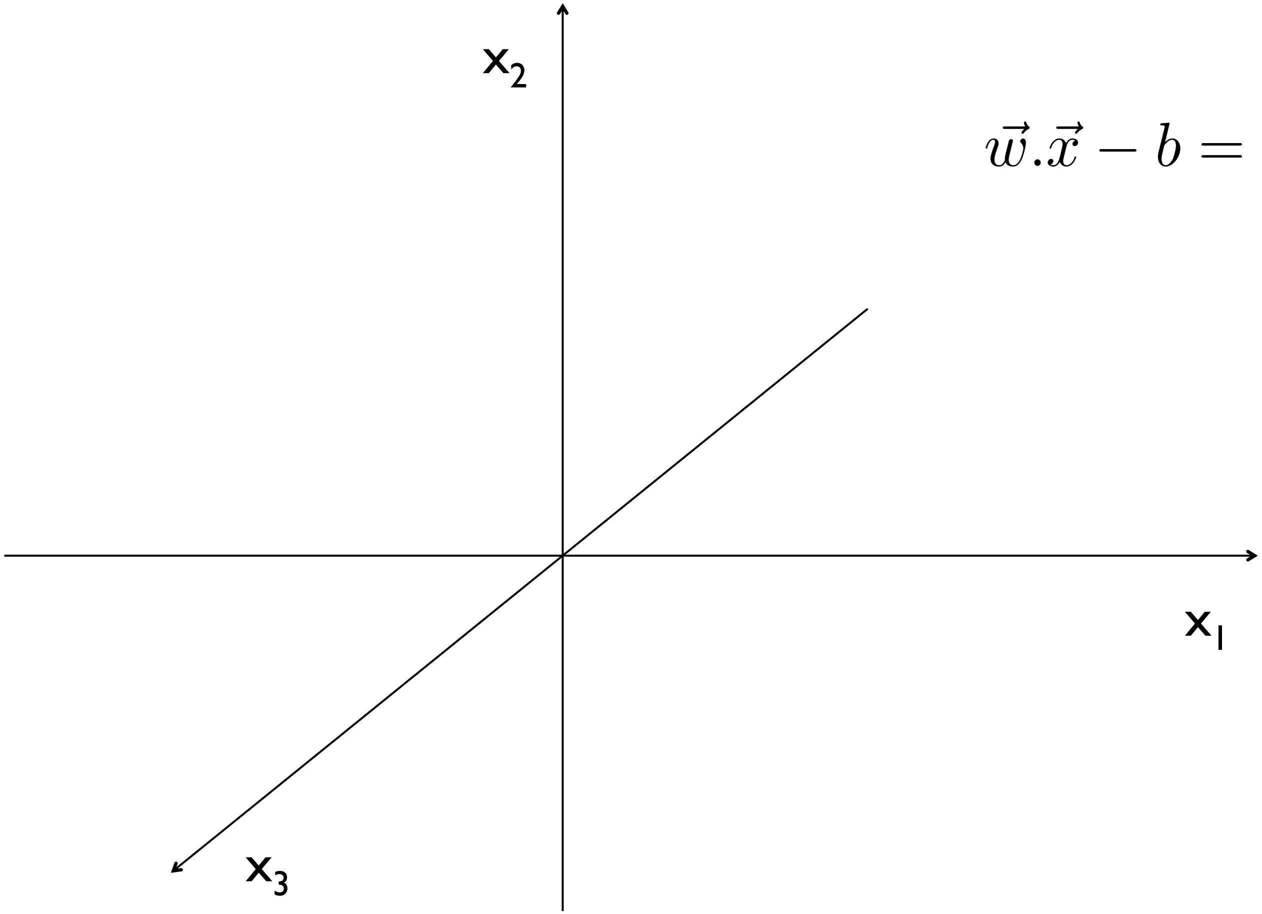
Three synaptic inputs $\vec{x} = (x_1, x_2, x_3)$ $\vec{w} = (w_1, w_2, w_3)$



$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

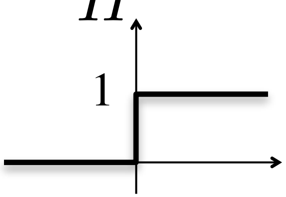


Three synaptic inputs $\vec{x} = (x_1, x_2, x_3)$ $\vec{w} = (w_1, w_2, w_3)$

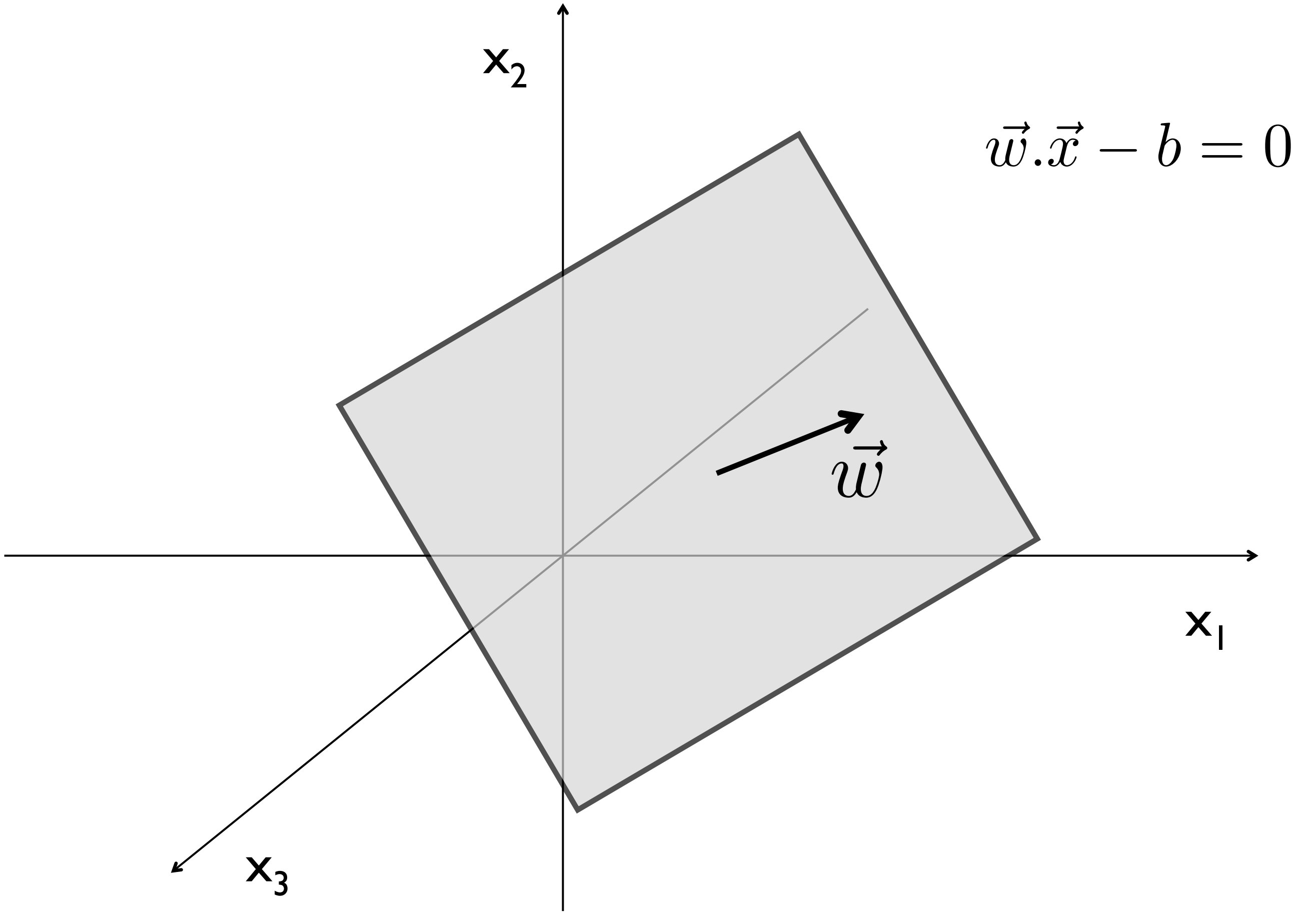


$$\vec{w}.\vec{x} - b = 0$$

$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

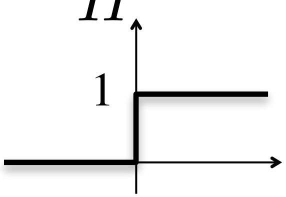


Three synaptic inputs $\vec{x} = (x_1, x_2, x_3)$ $\vec{w} = (w_1, w_2, w_3)$

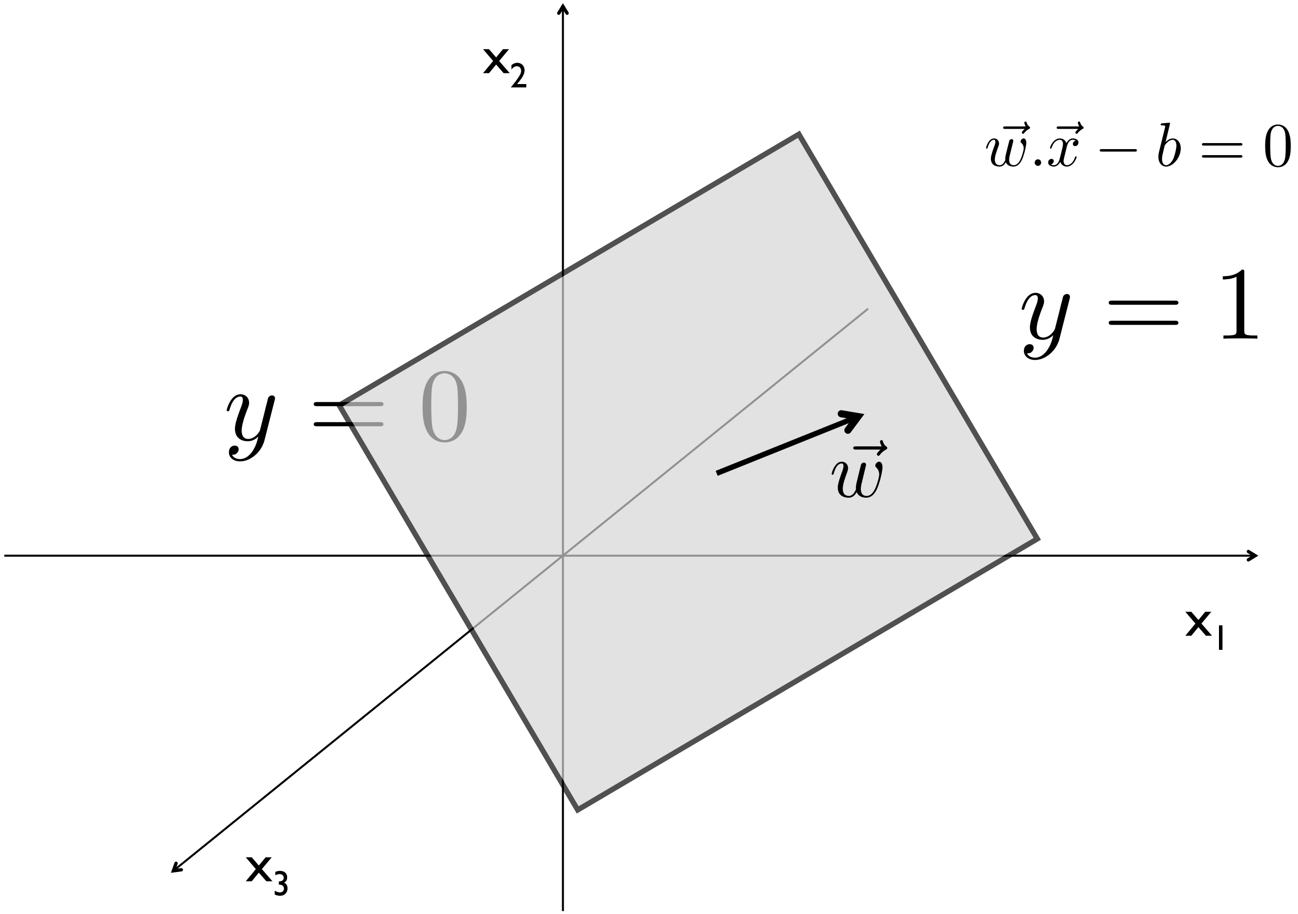


$$\vec{w}.\vec{x} - b = 0$$

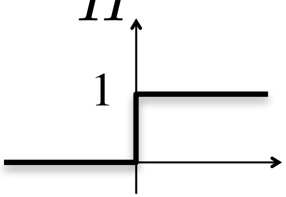
$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$



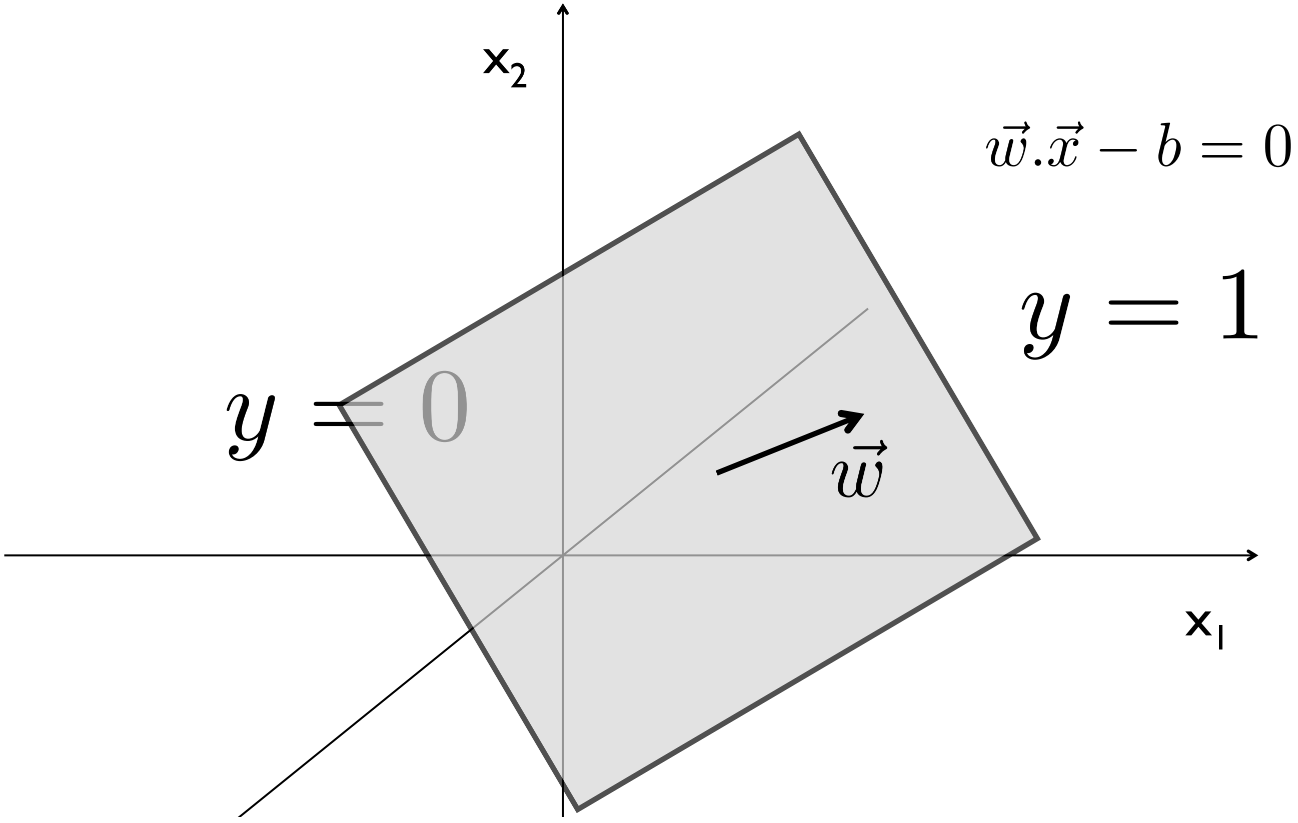
Three synaptic inputs $\vec{x} = (x_1, x_2, x_3)$ $\vec{w} = (w_1, w_2, w_3)$



$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

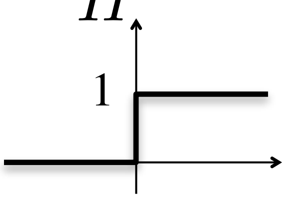


Three synaptic inputs $\vec{x} = (x_1, x_2, x_3)$ $\vec{w} = (w_1, w_2, w_3)$



→ Separates the space in two regions

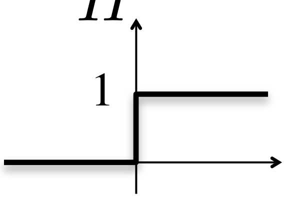
$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$



N synaptic inputs $\vec{x} = (x_1, \ldots, x_N)$ $\vec{w} = (w_1, \ldots, w_N)$

$$\vec{w}.\vec{x} - b = 0$$

$$y = H\left(\sum_{k=1}^{N} w_k x_k - b\right) = H\left(\vec{w}.\vec{x} - b\right)$$

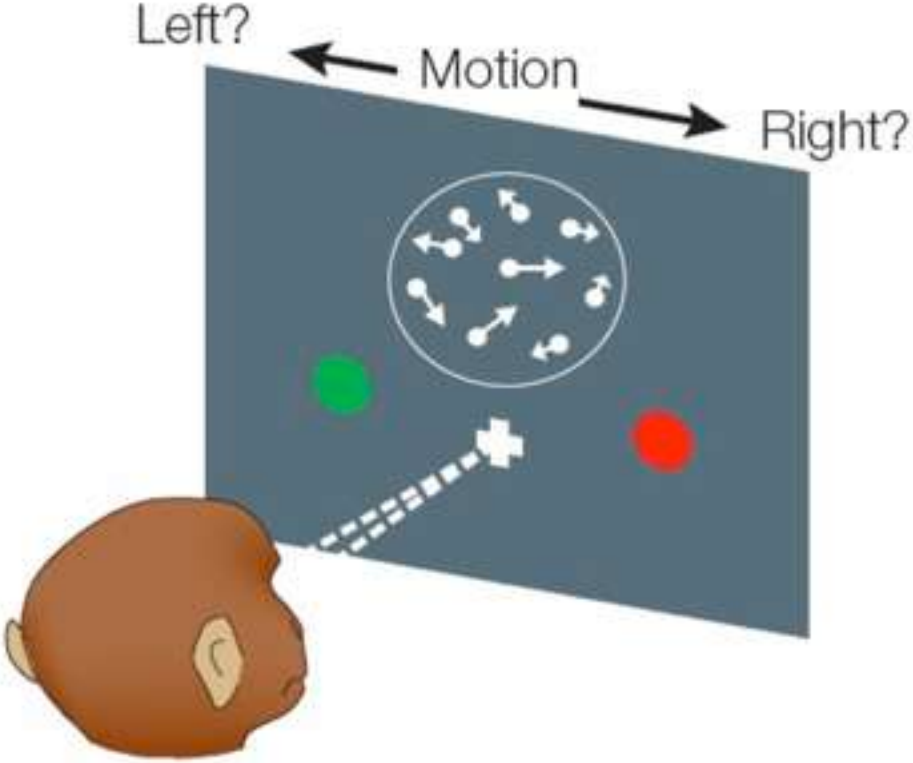


N synaptic inputs $\vec{x} = (x_1, \ldots, x_N)$ $\vec{w} = (w_1, \ldots, w_N)$

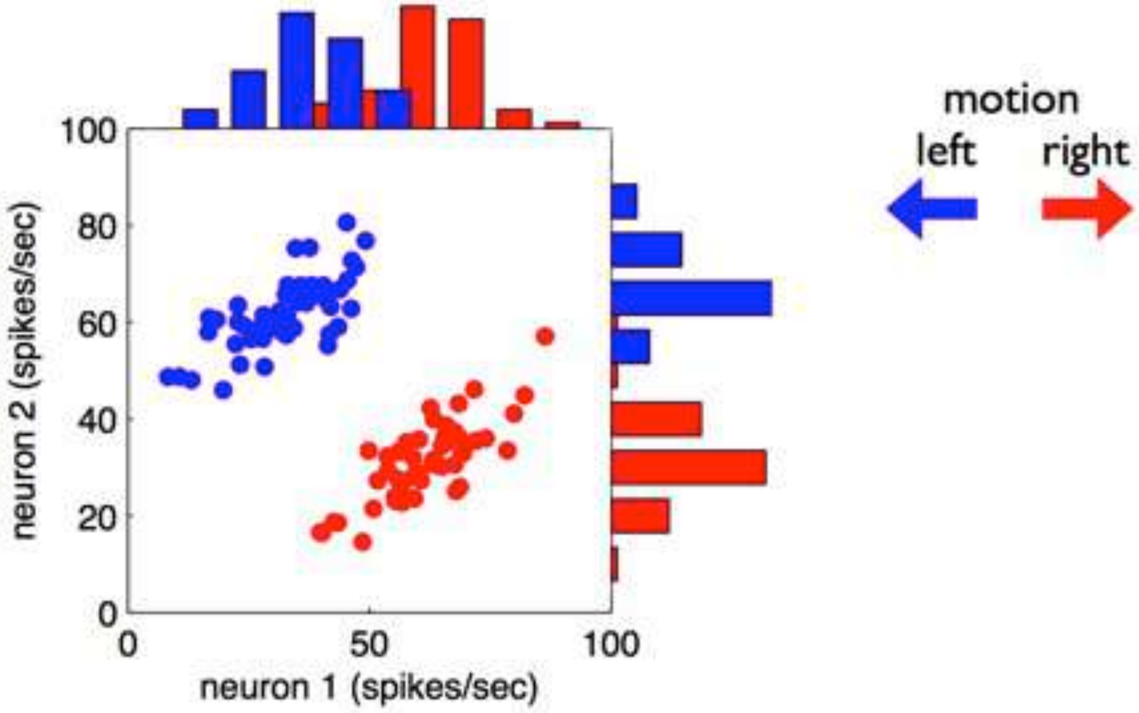
$\vec{w}.\vec{x} - b = 0$ defines a **hyperplane**

→ Separates the space in two regions
binary classifier

Binary classification task

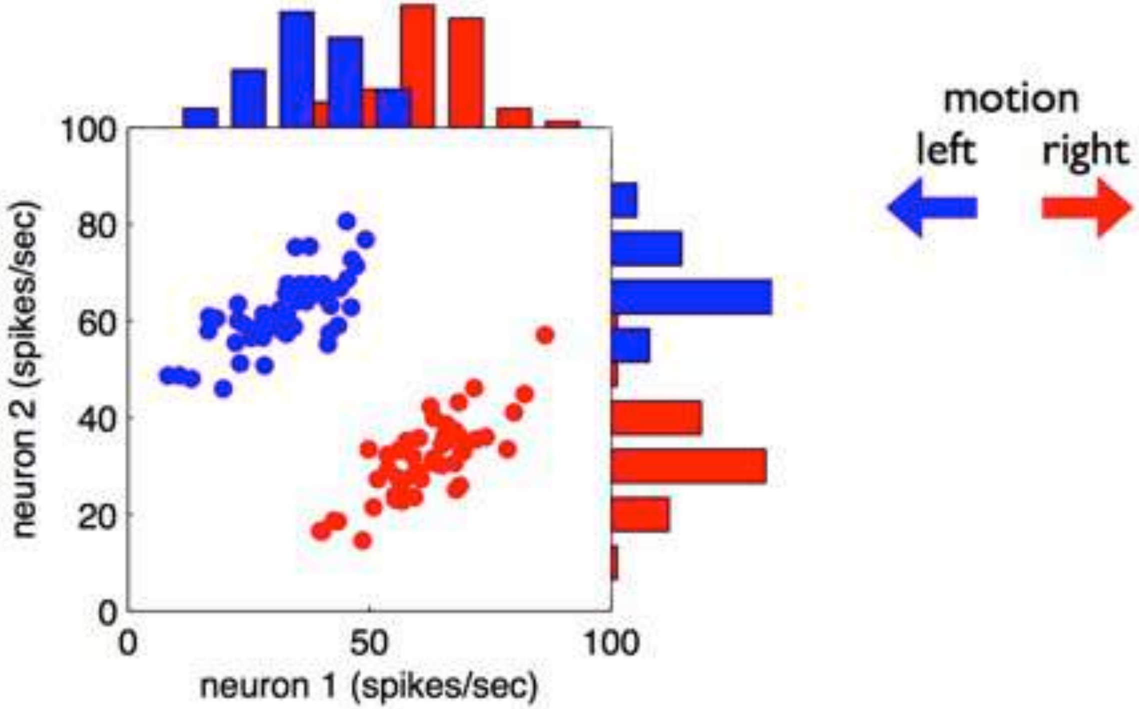


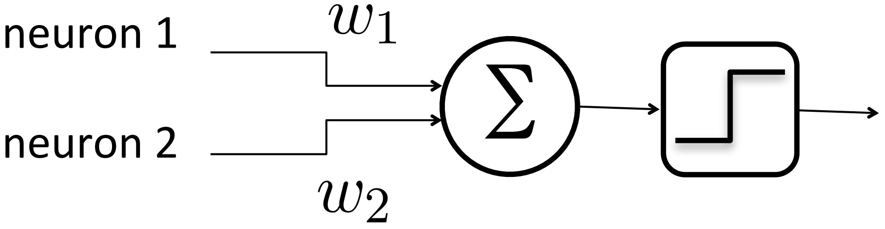
Activity of neurons in MT



→ Can an upstream neuron read out the motion direction?

Neural readout of motion direction





neuron 1

w_1

neuron 2

w_2

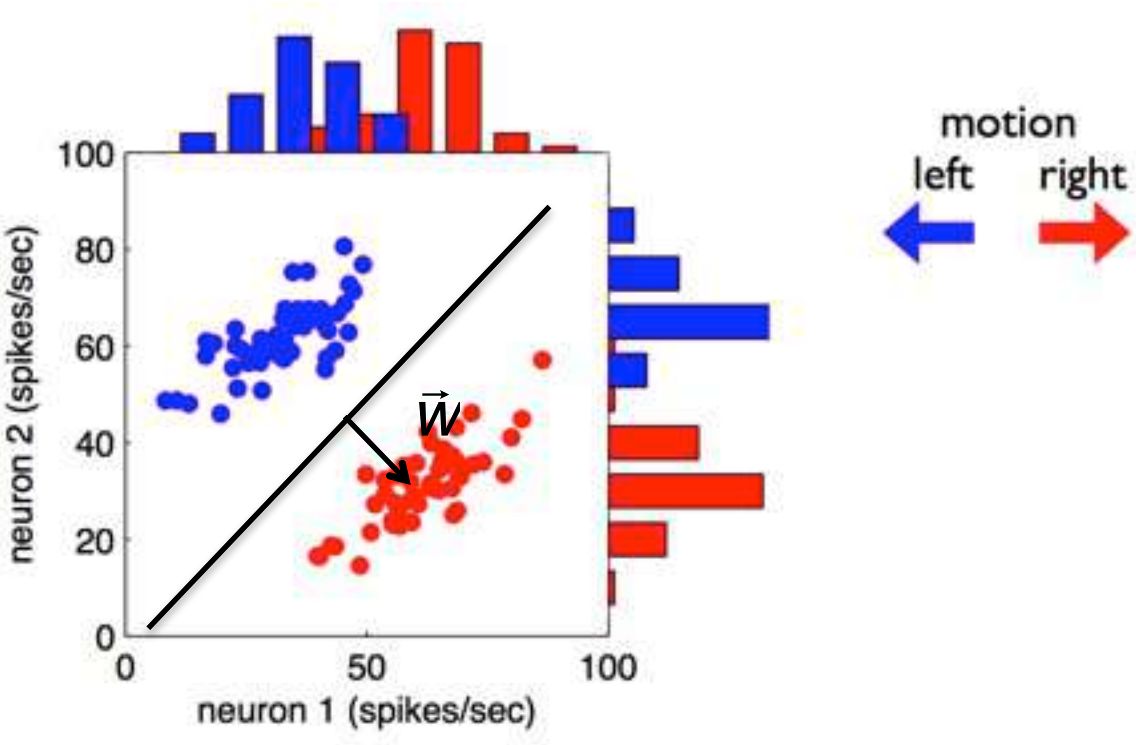
Σ

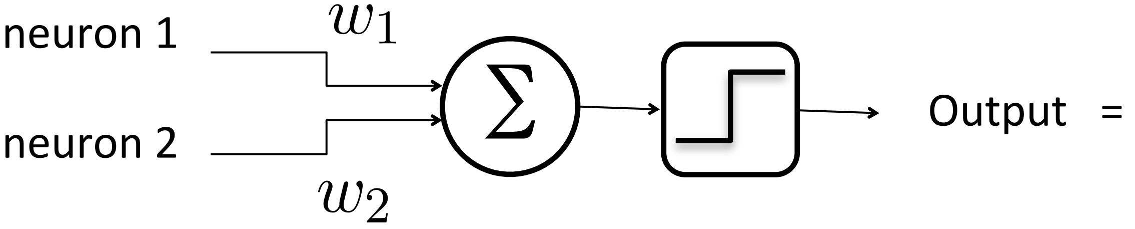
Output =

1 if right

0 if left

Neural readout of motion direction





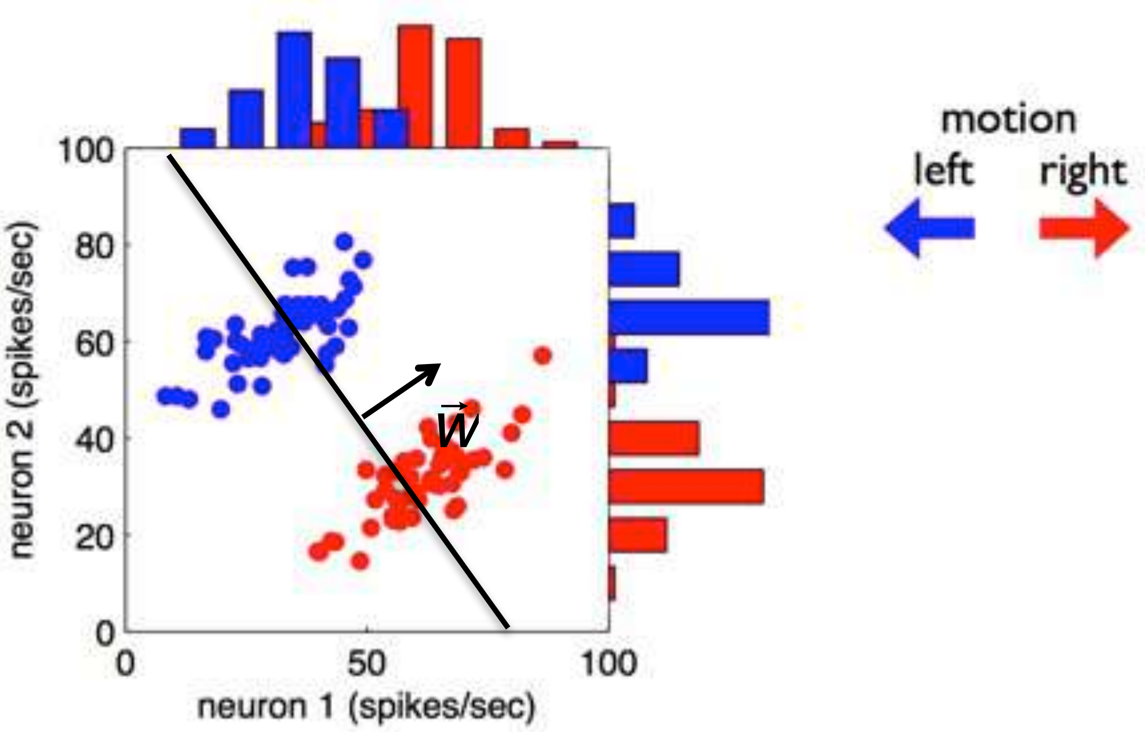
Output =

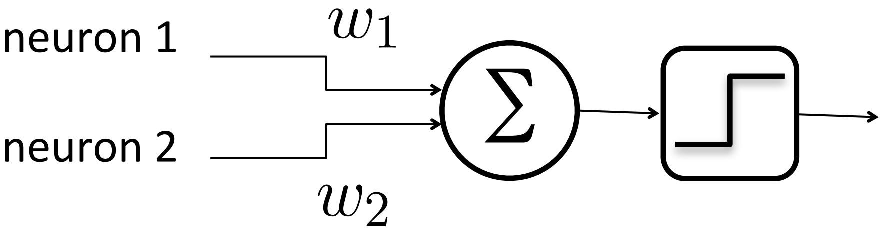
1 if right

0 if left

→ Correct readout!

Neural readout of motion direction

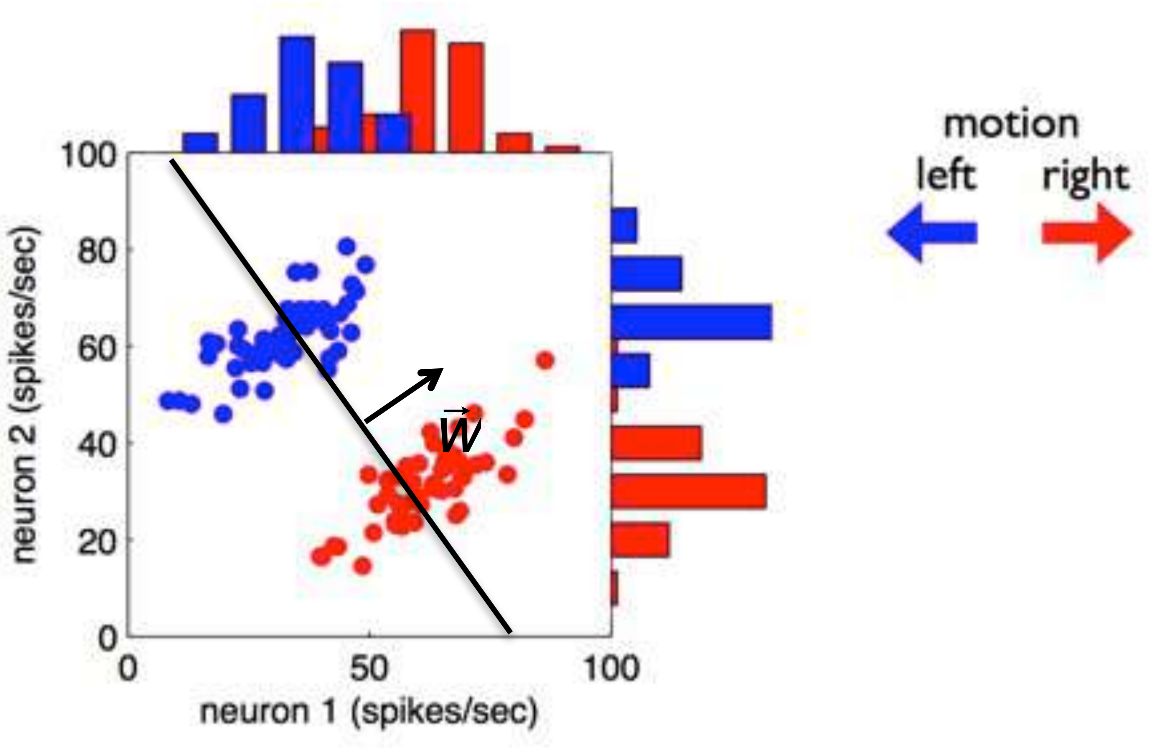




Output = 1 if right

0 if left

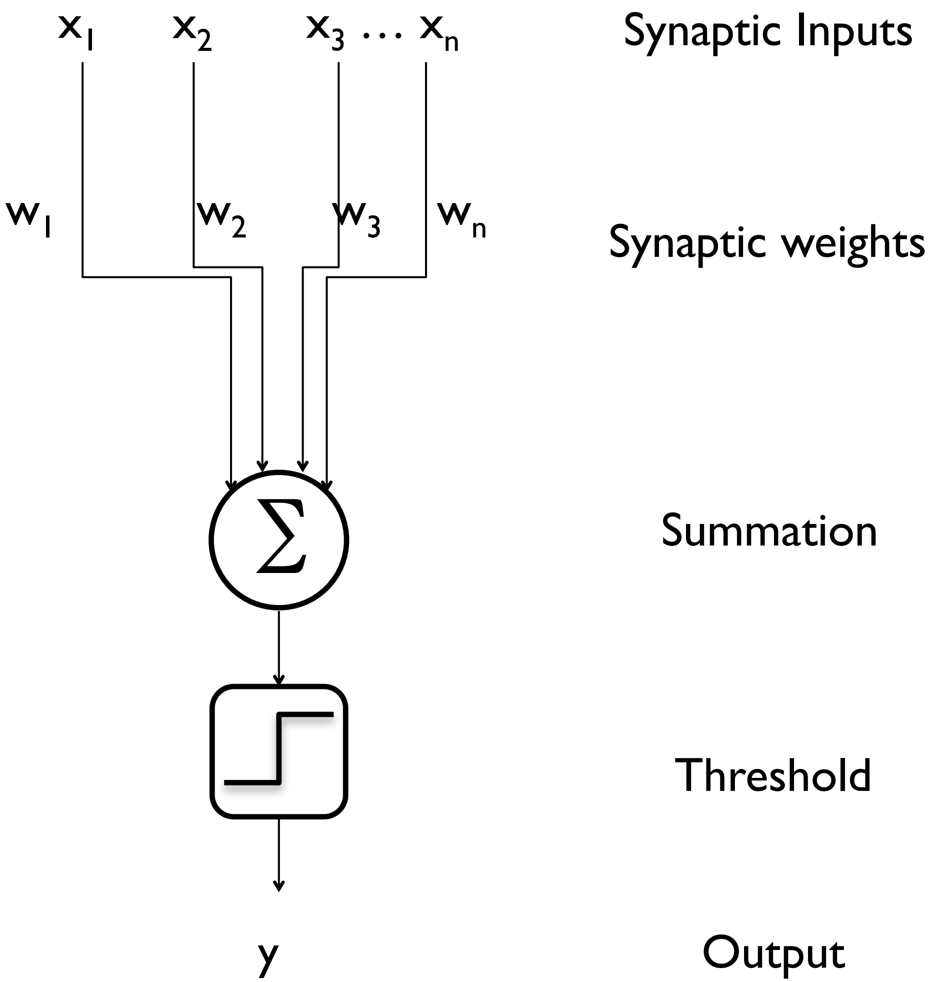
Binary neuron = linear classifier



Need to adjust synaptic weights!!

→ Learning = modification of synaptic weights

Learning in the Binary Neuron



x_1 x_2 $x_3 \ldots x_n$ Synaptic Inputs

w_1 w_2 w_3 w_n Synaptic weights

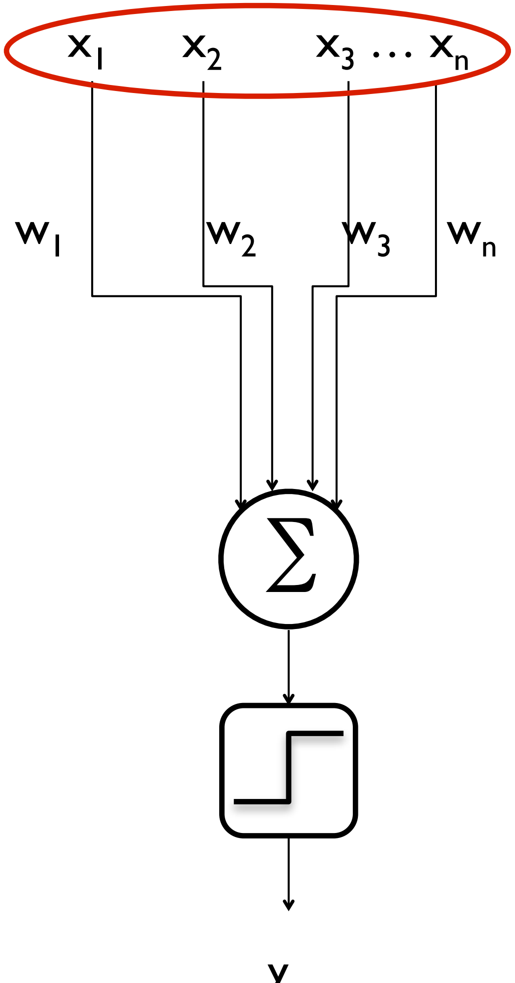
Σ Summation

Threshold

y Output

Learning in the Binary Neuron

given



Synaptic Inputs

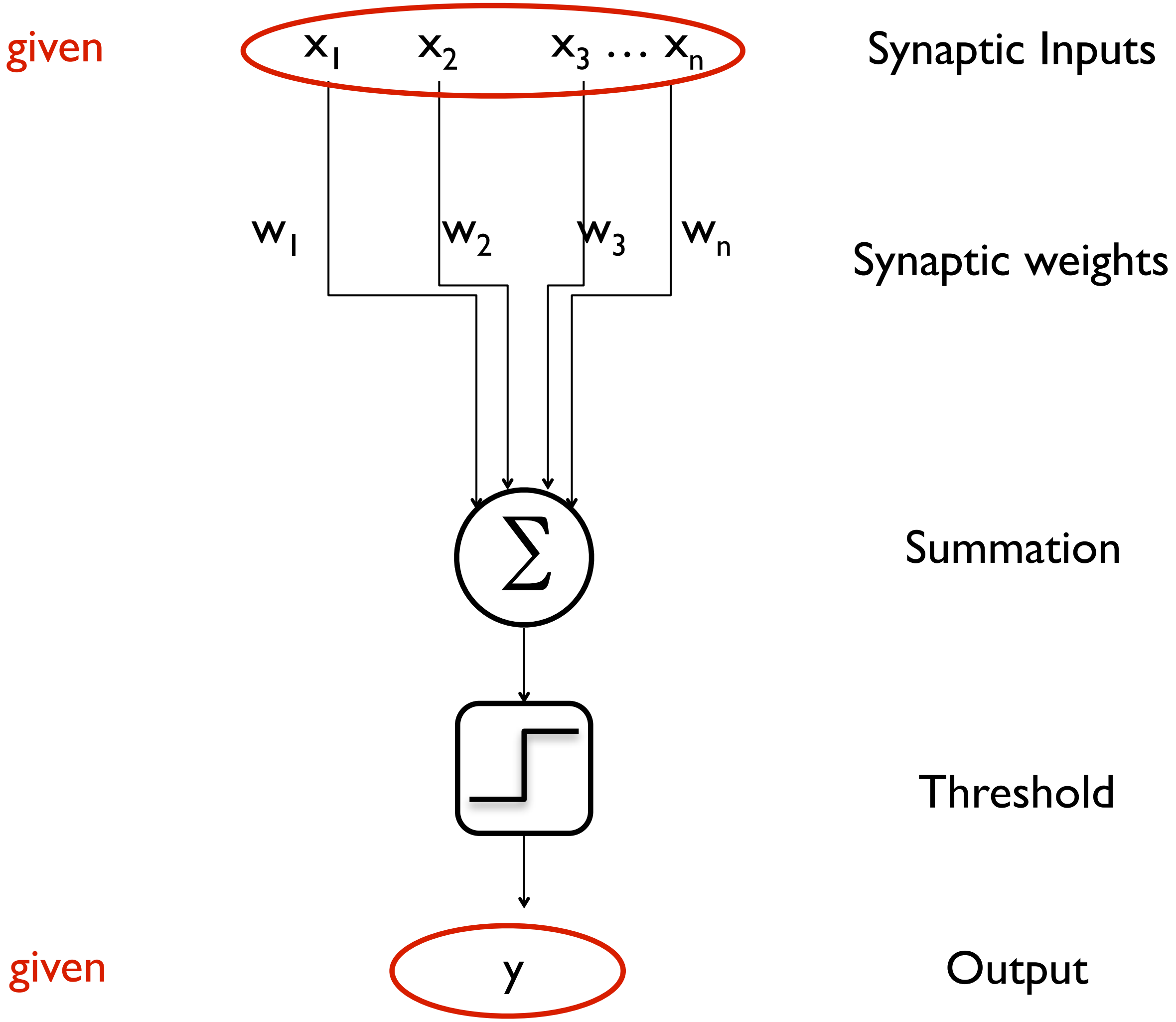
Synaptic weights

Summation

Threshold

Output

Learning in the Binary Neuron



given x_1 x_2 $x_3 \ldots x_n$ Synaptic Inputs

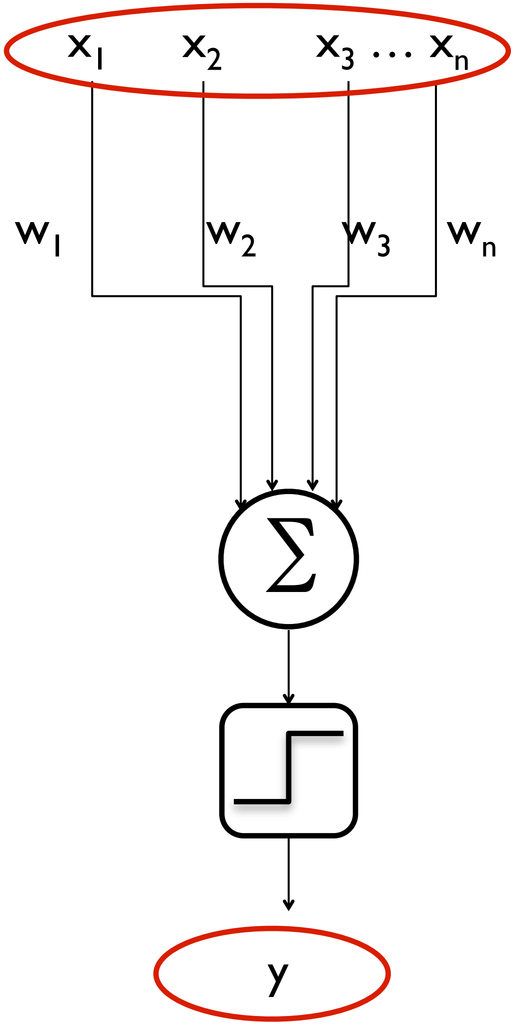
w_1 w_2 w_3 w_n Synaptic weights

Σ Summation

Threshold

given y Output

Learning in the Binary Neuron



given — Synaptic Inputs

Synaptic weights → adjust

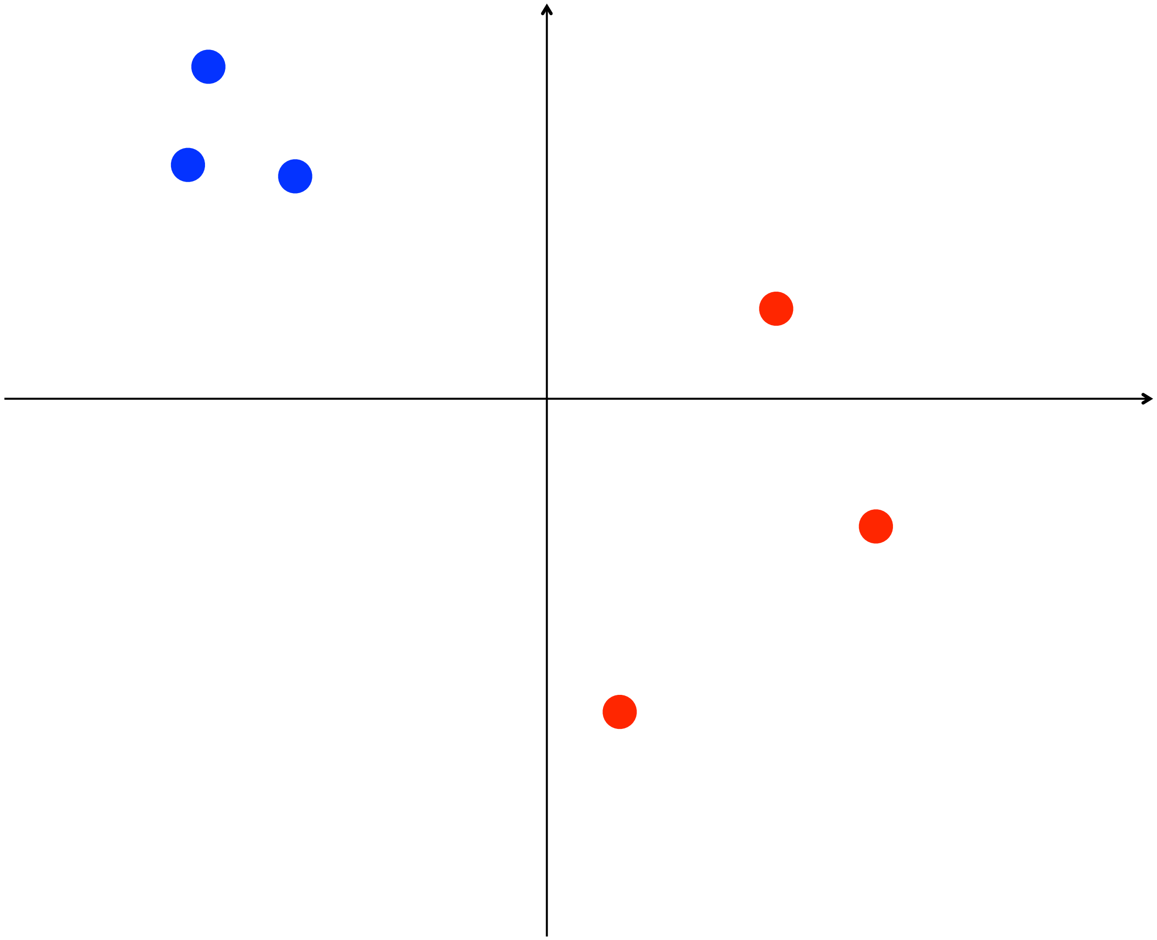
Summation

Threshold

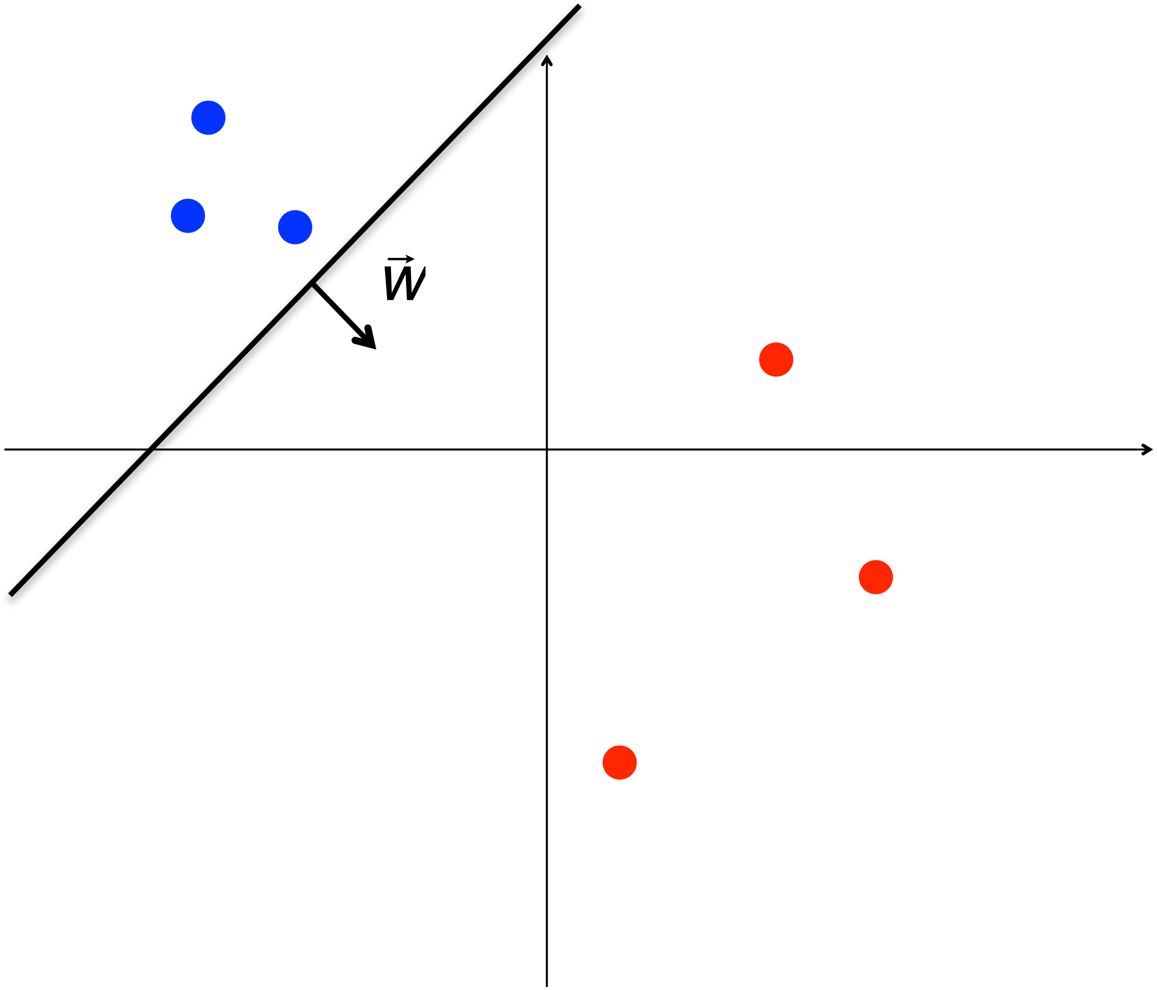
given — Output

→ synaptic plasticity underlies learning

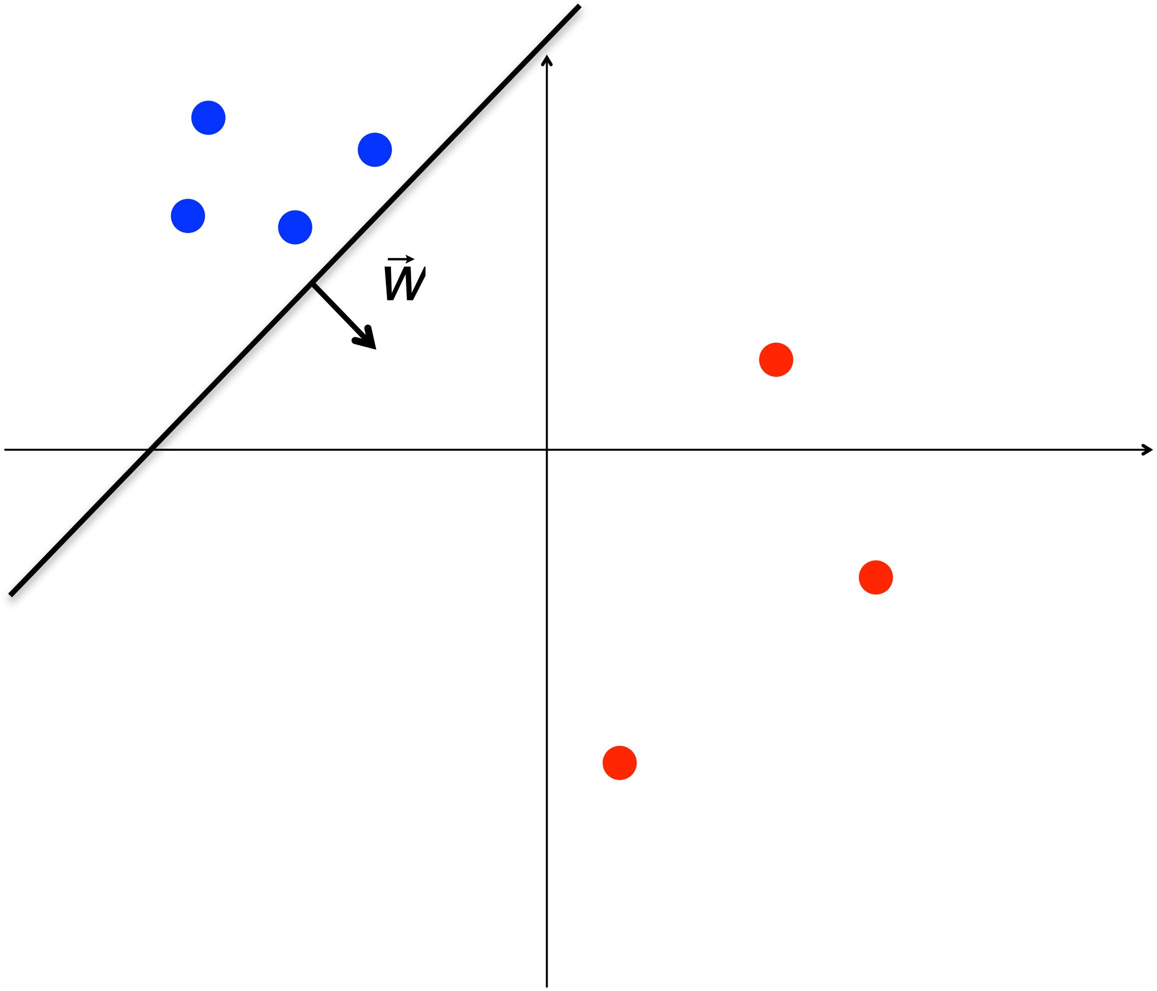
Exemple: classify red points as 1 and blue points as 0



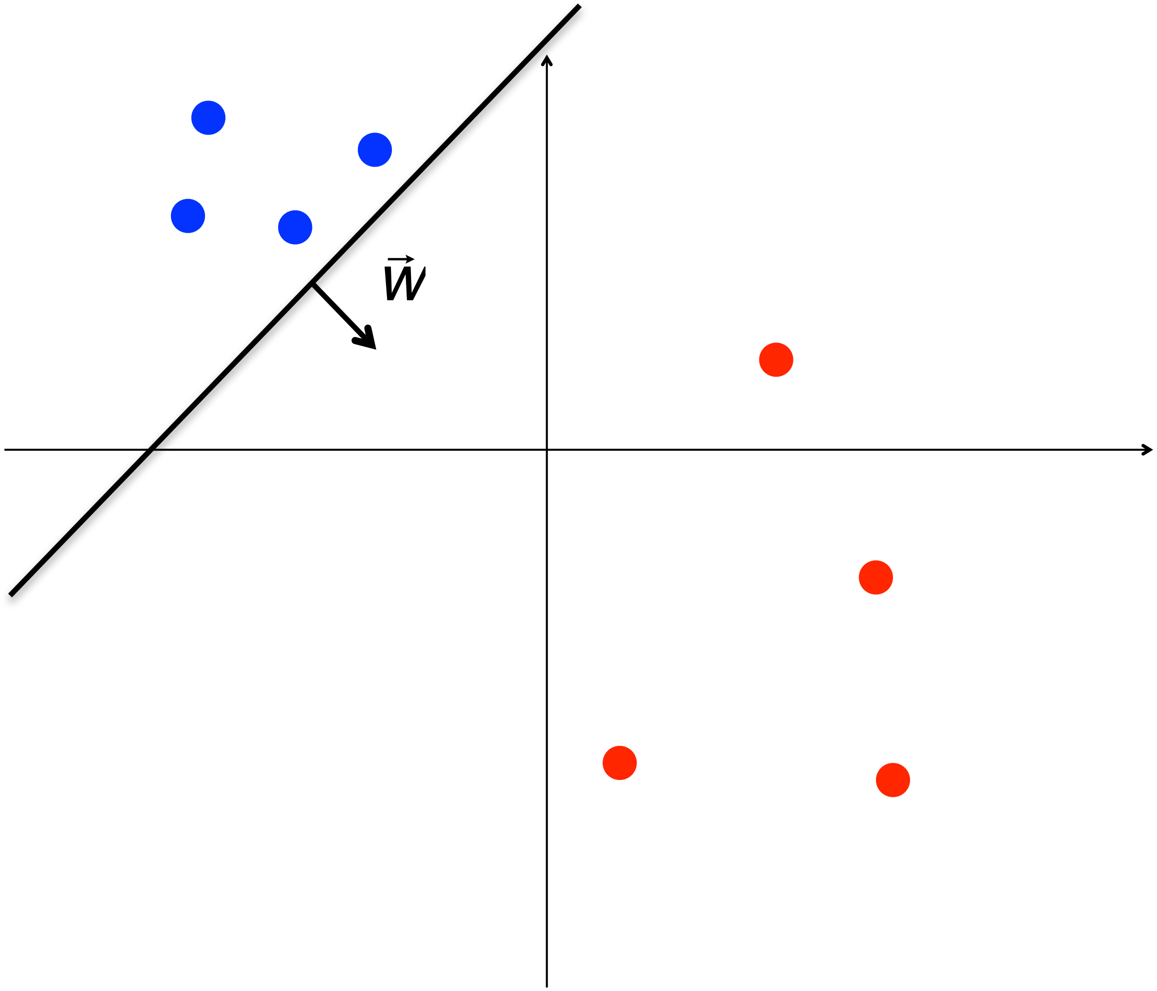
Exemple: classify red points as 1 and blue points as 0



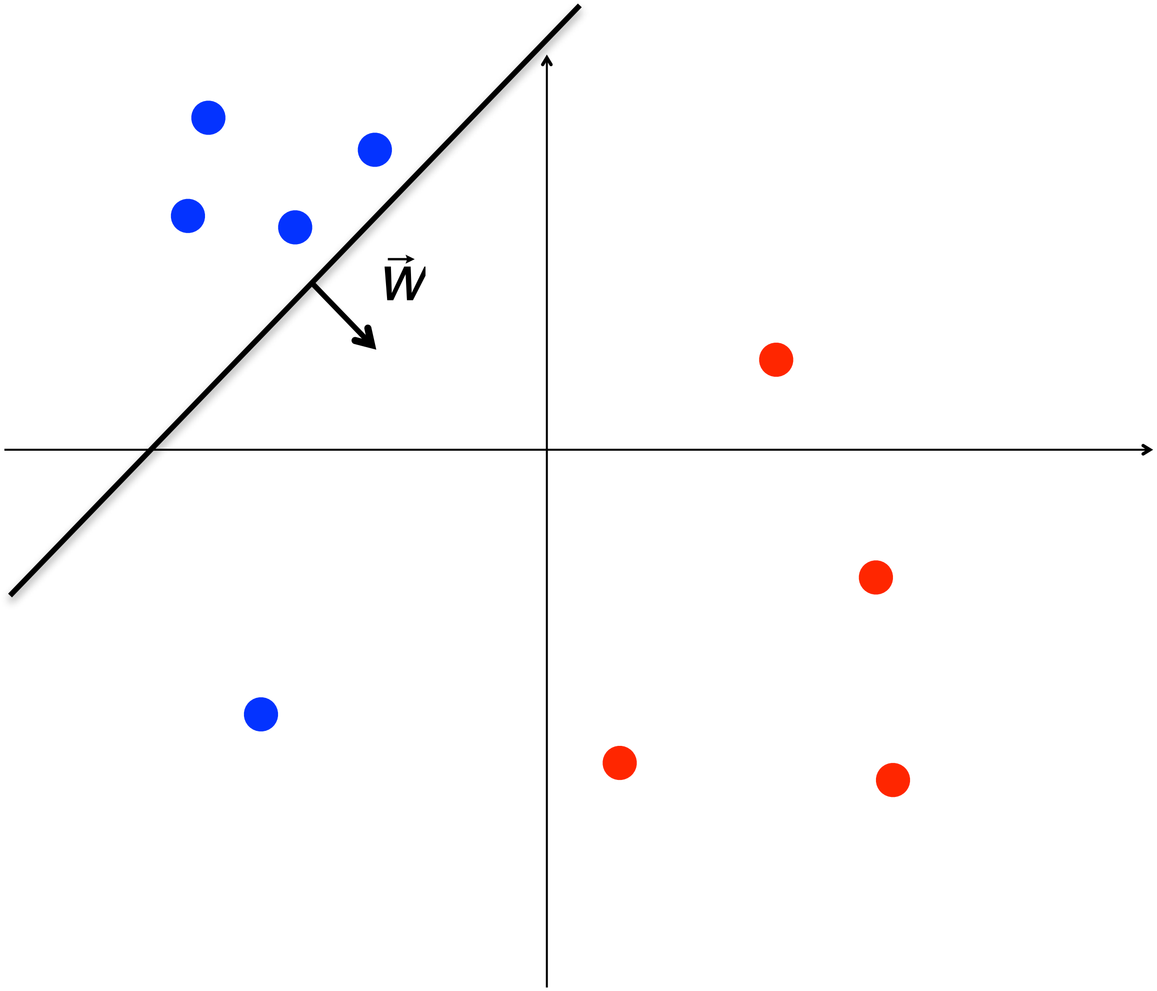
Exemple: classify red points as 1 and blue points as 0



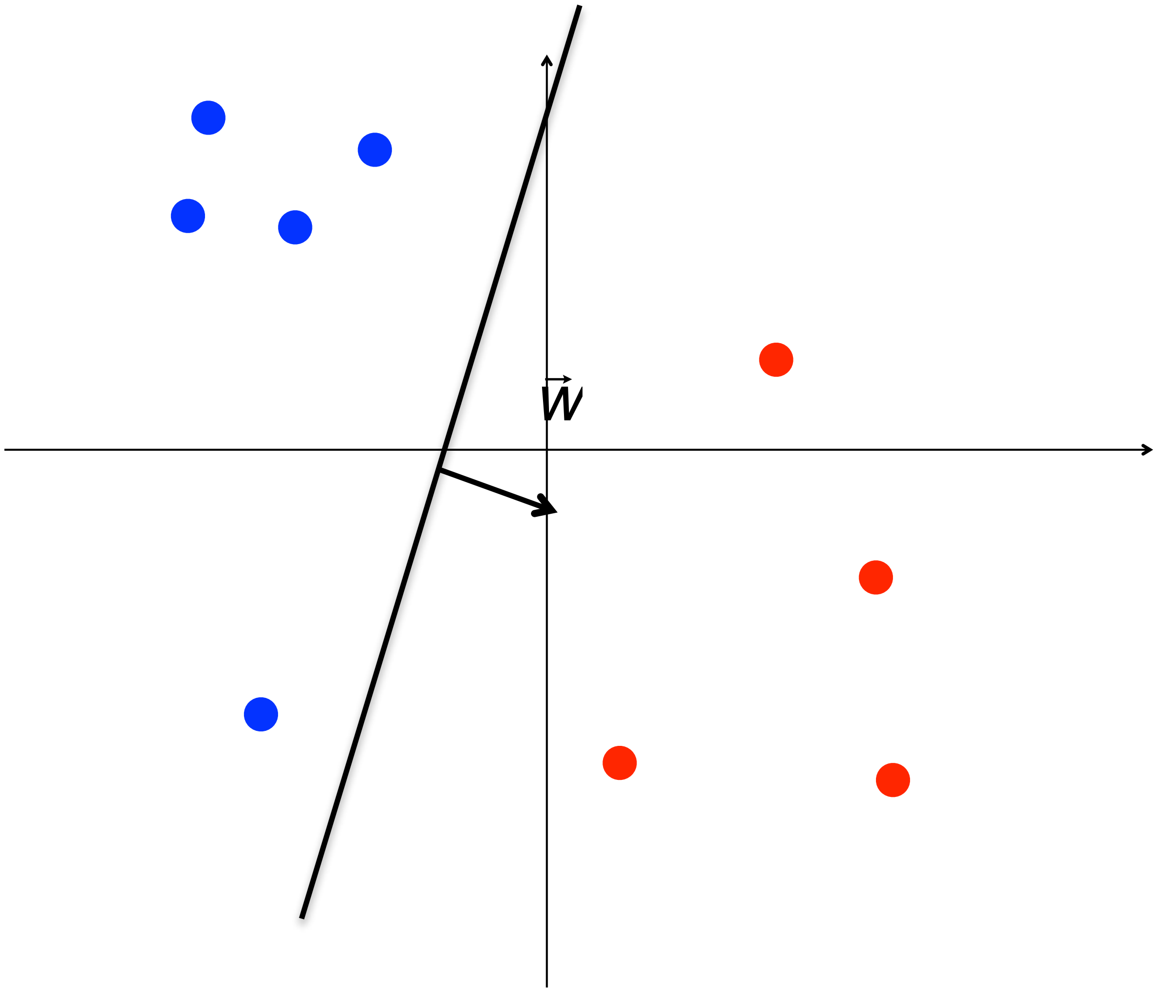
Exemple: classify red points as 1 and blue points as 0



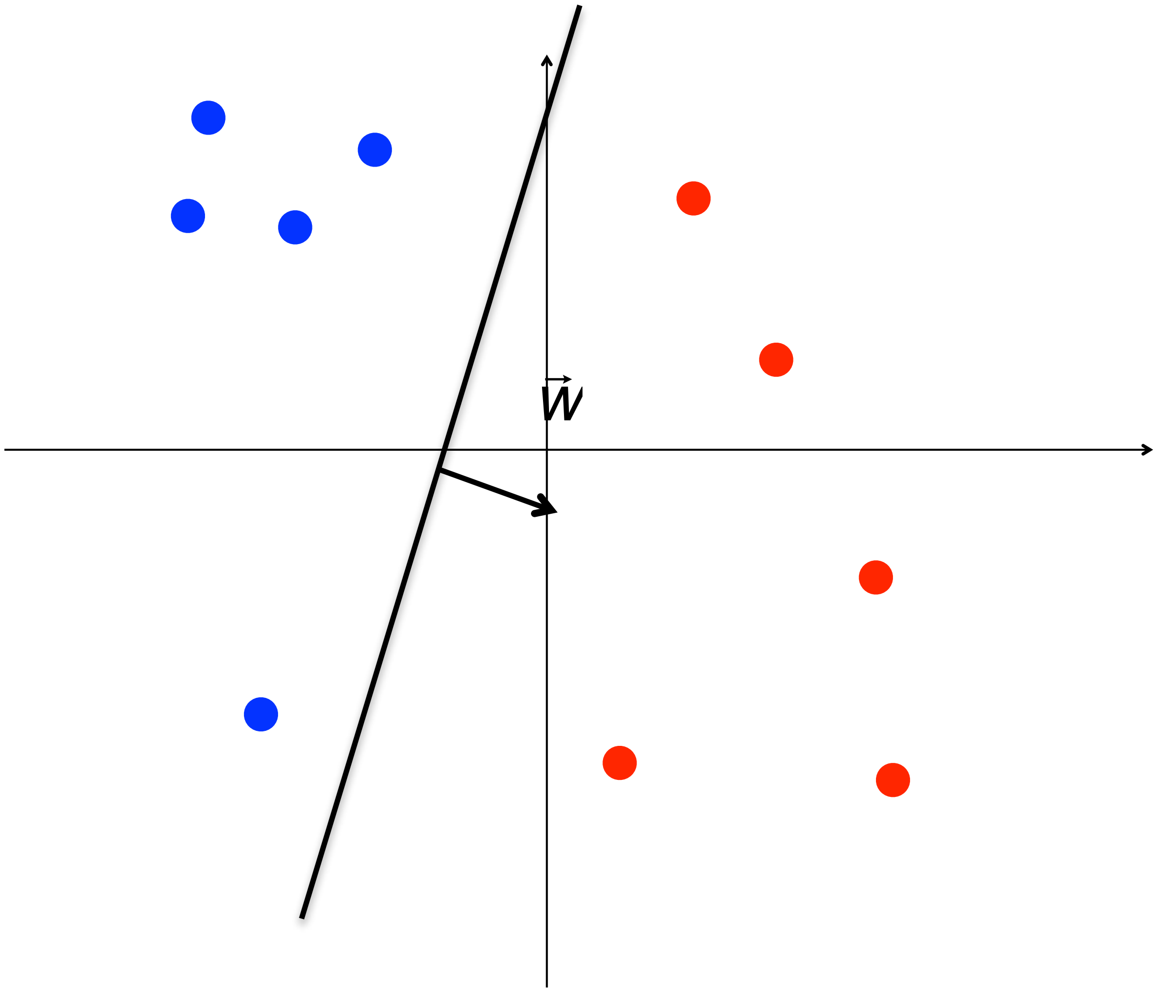
Exemple: classify red points as 1 and blue points as 0



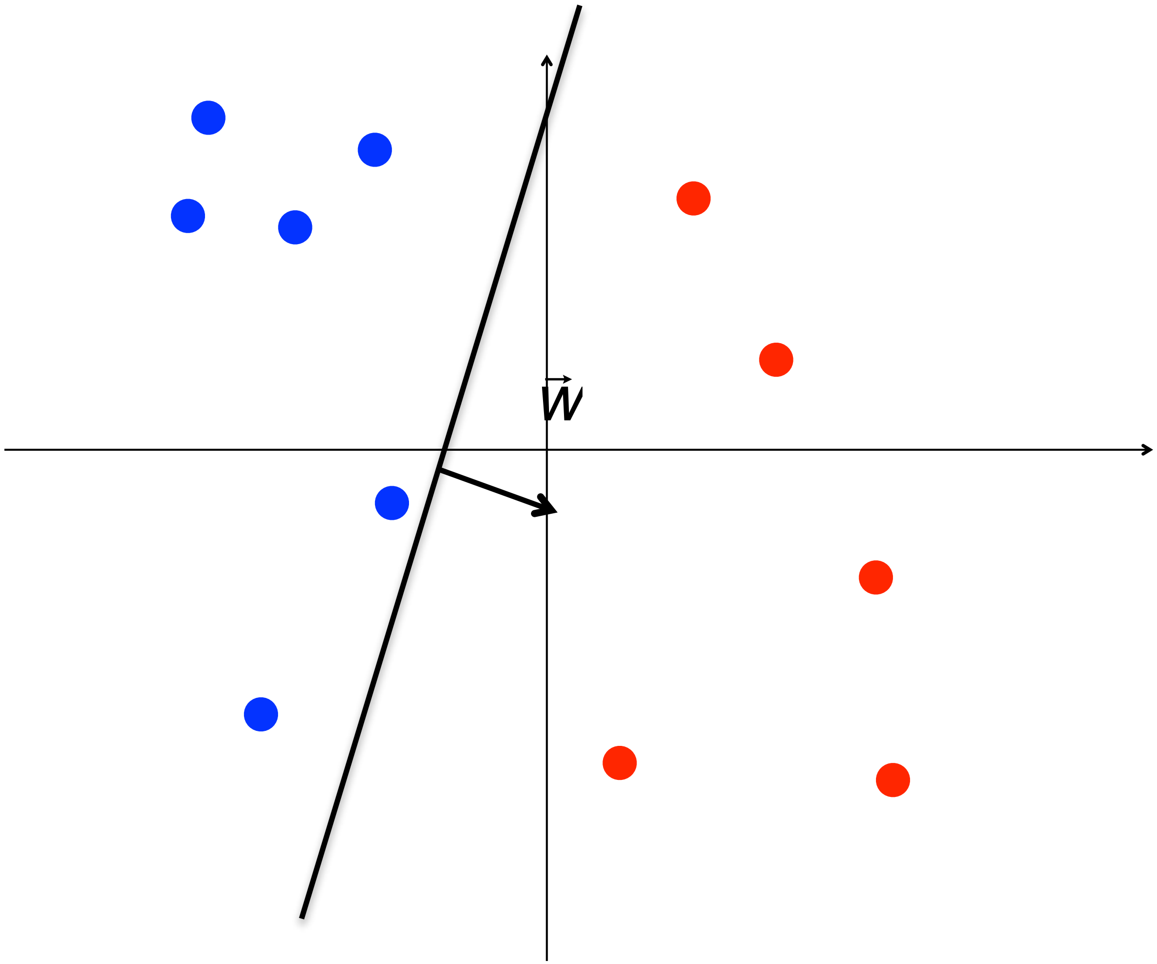
Exemple: classify red points as 1 and blue points as 0



Exemple: classify red points as 1 and blue points as 0



Exemple: classify red points as 1 and blue points as 0



→ automatic rule for updating synaptic weights?

The perceptron learning rule

Rosenblatt (1958)

Training set of p patterns: $\{(x^{(0)}, d_0), (x^{(1)}, d_1) \ldots (x^{(p)}, d_p)\}$

where $x^{(k)}$ is an input vector

$d_k = 0 \text{ or } 1$ is a desired output

The perceptron learning rule

Rosenblatt (1958)

Training set of p patterns: $\{(x^{(0)}, d_0), (x^{(1)}, d_1) \ldots (x^{(p)}, d_p)\}$

where $x^{(k)}$ is an input vector

$d_k = 0 \text{ or } 1$ is a desired output

On every step:

for each pattern k

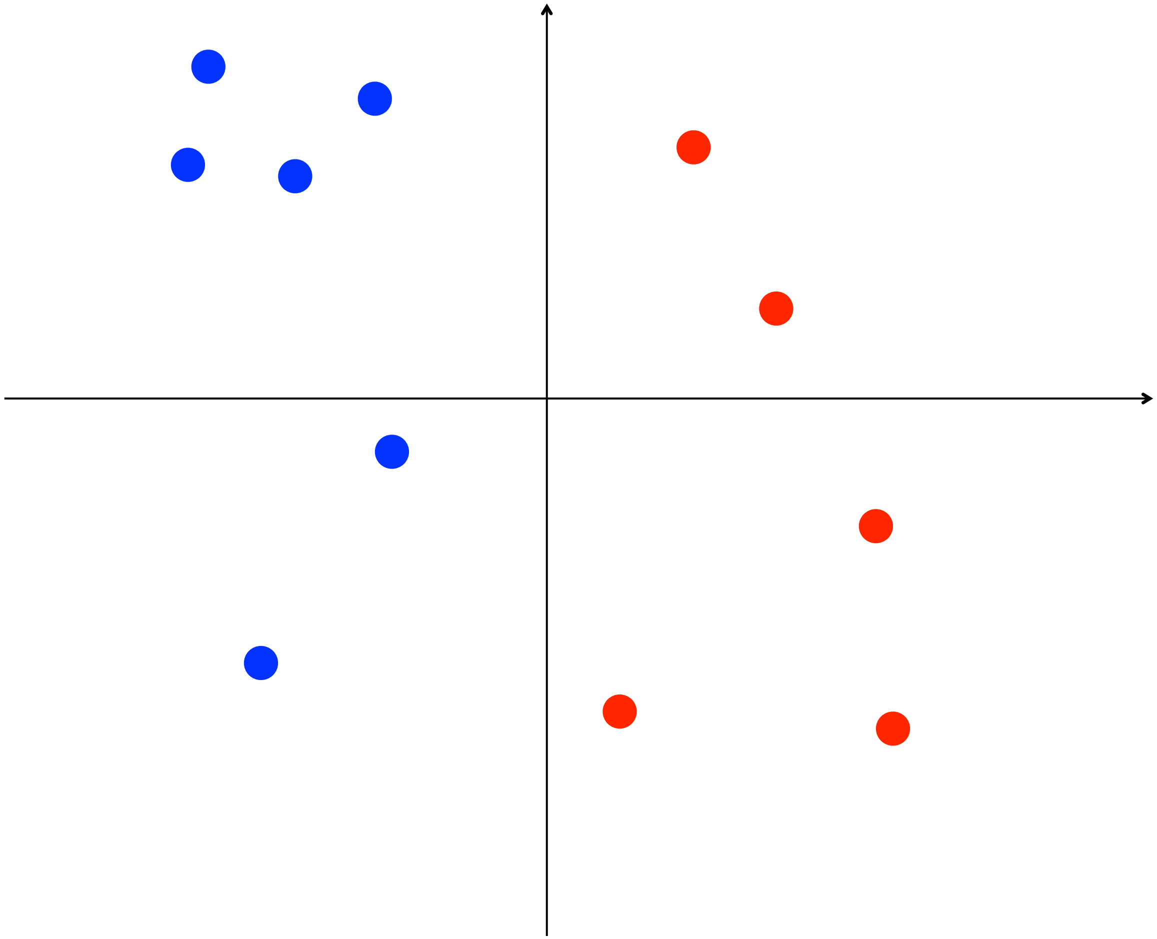
1. compute the output $y_k = H(\sum_{i=1}^{N} w_i x_i^{(k)})$
2. if $y_k \neq d_k$ update the weights:

$$w_i(t+1) = w_i(t) + (d_k - y_k)x_i^{(k)}$$

→ **Converges in a finite number of steps if a solution exists**

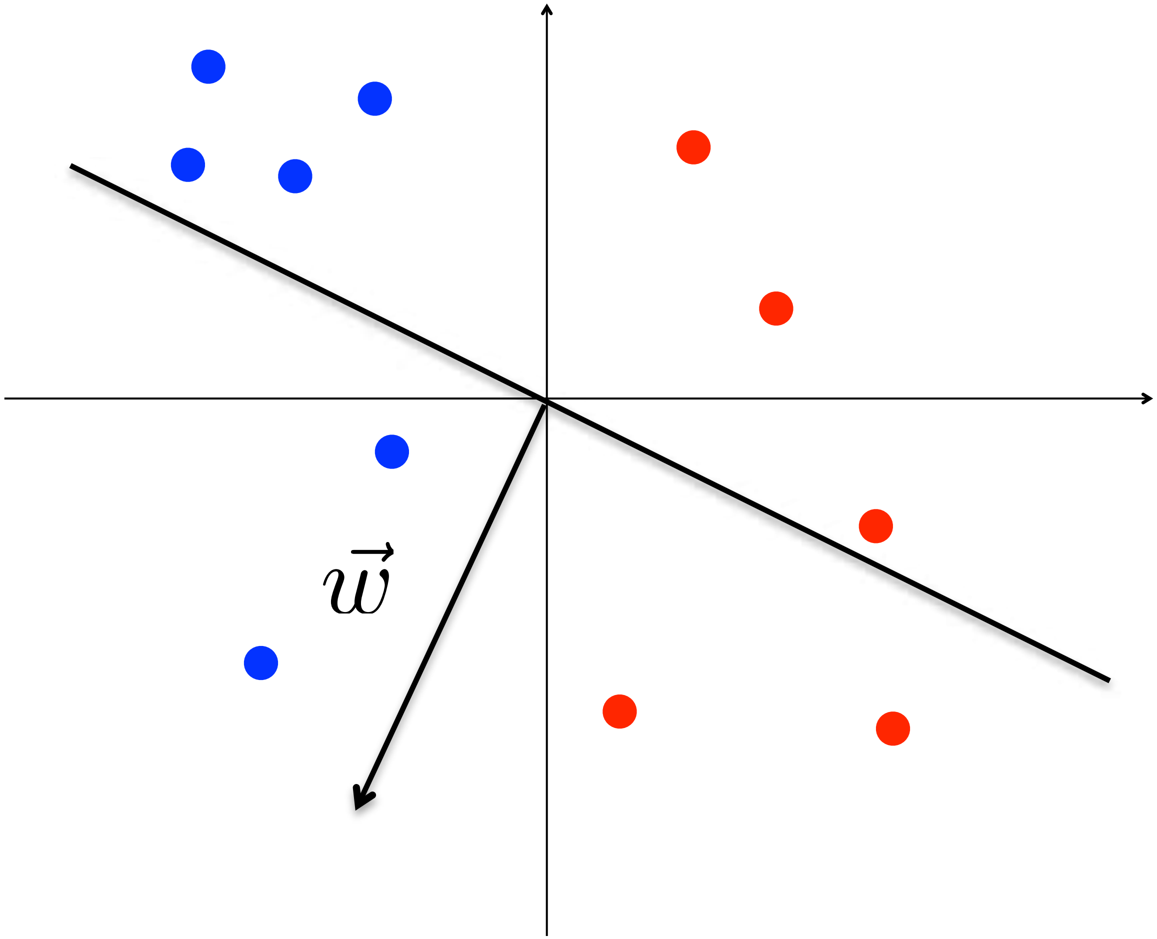
The perceptron learning rule

Aim: classify red points as 1 and blue points as 0



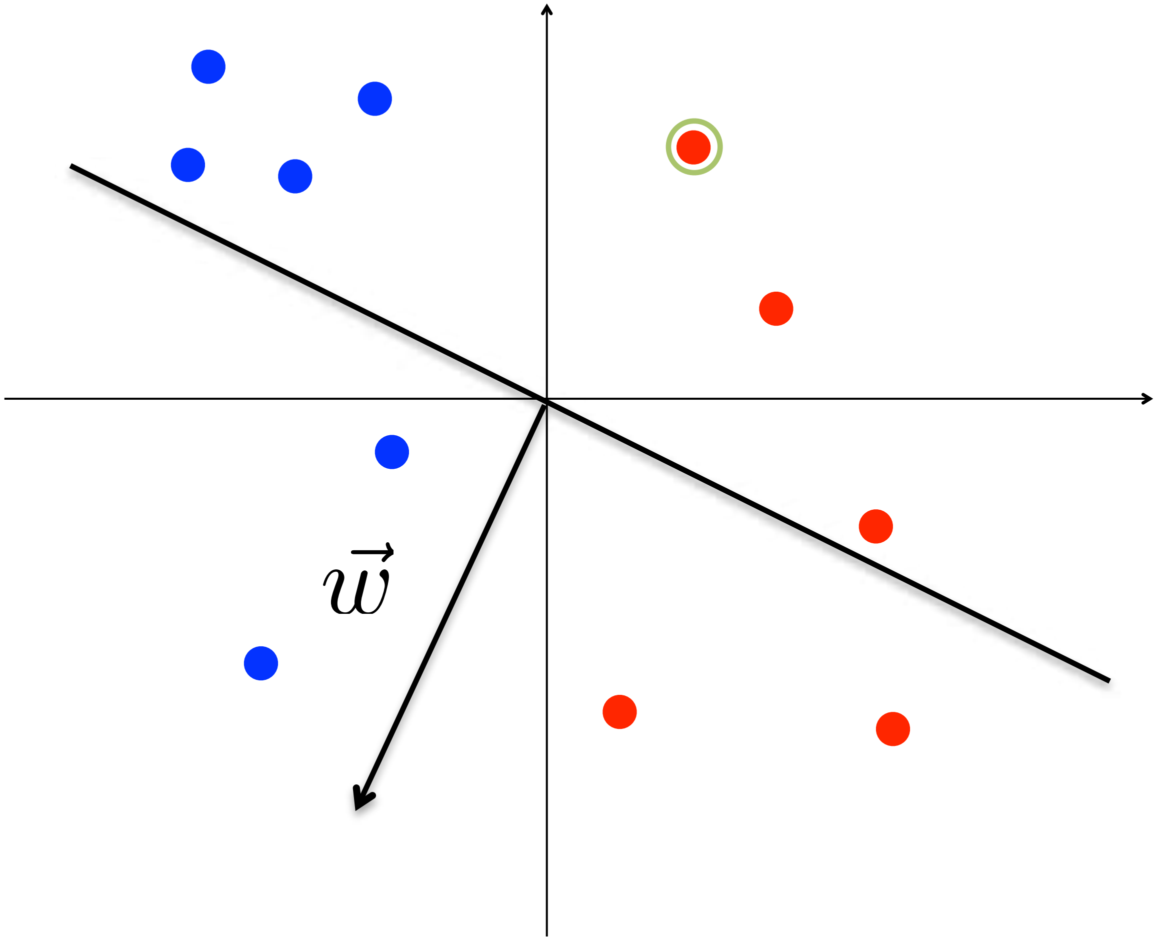
The perceptron learning rule

Start with a random set of synaptic weights



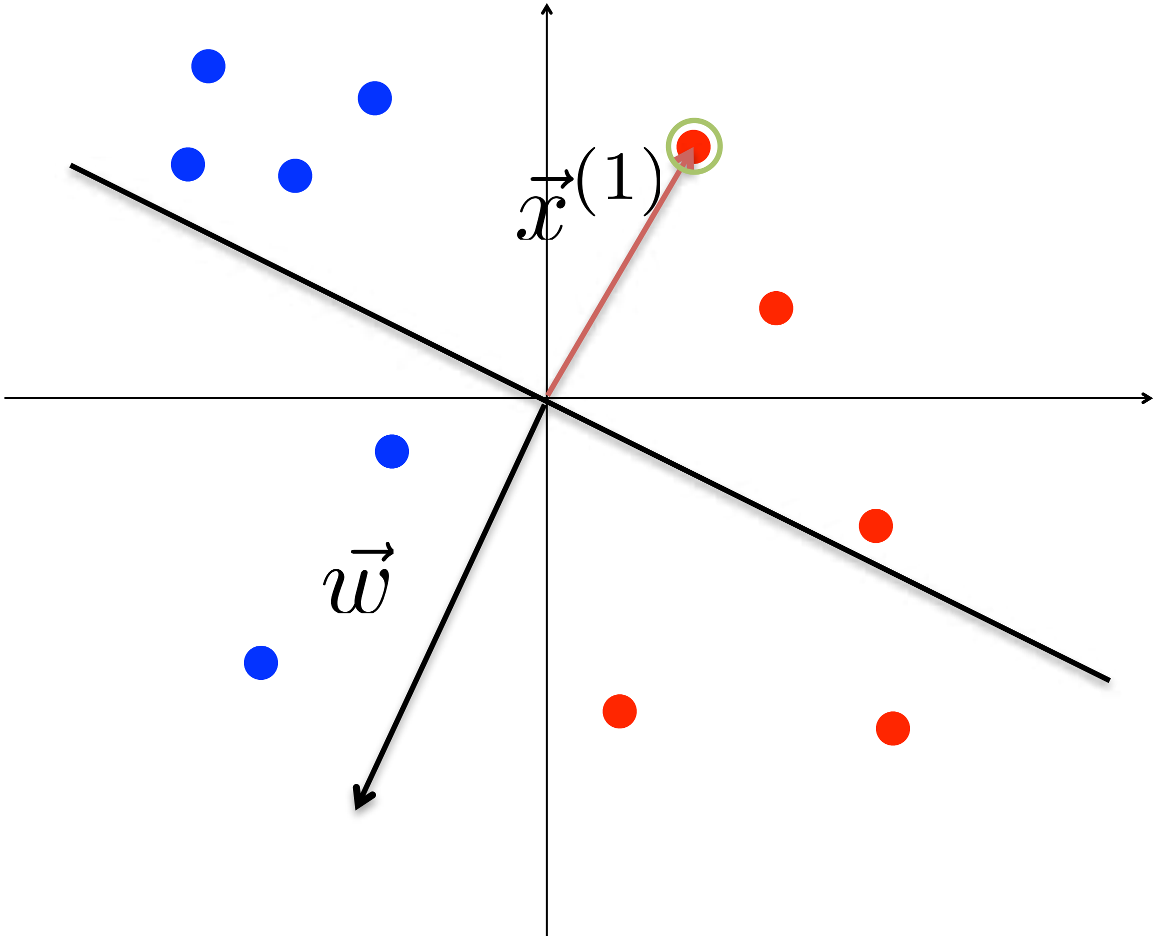
The perceptron learning rule

Choose a misclassified pattern



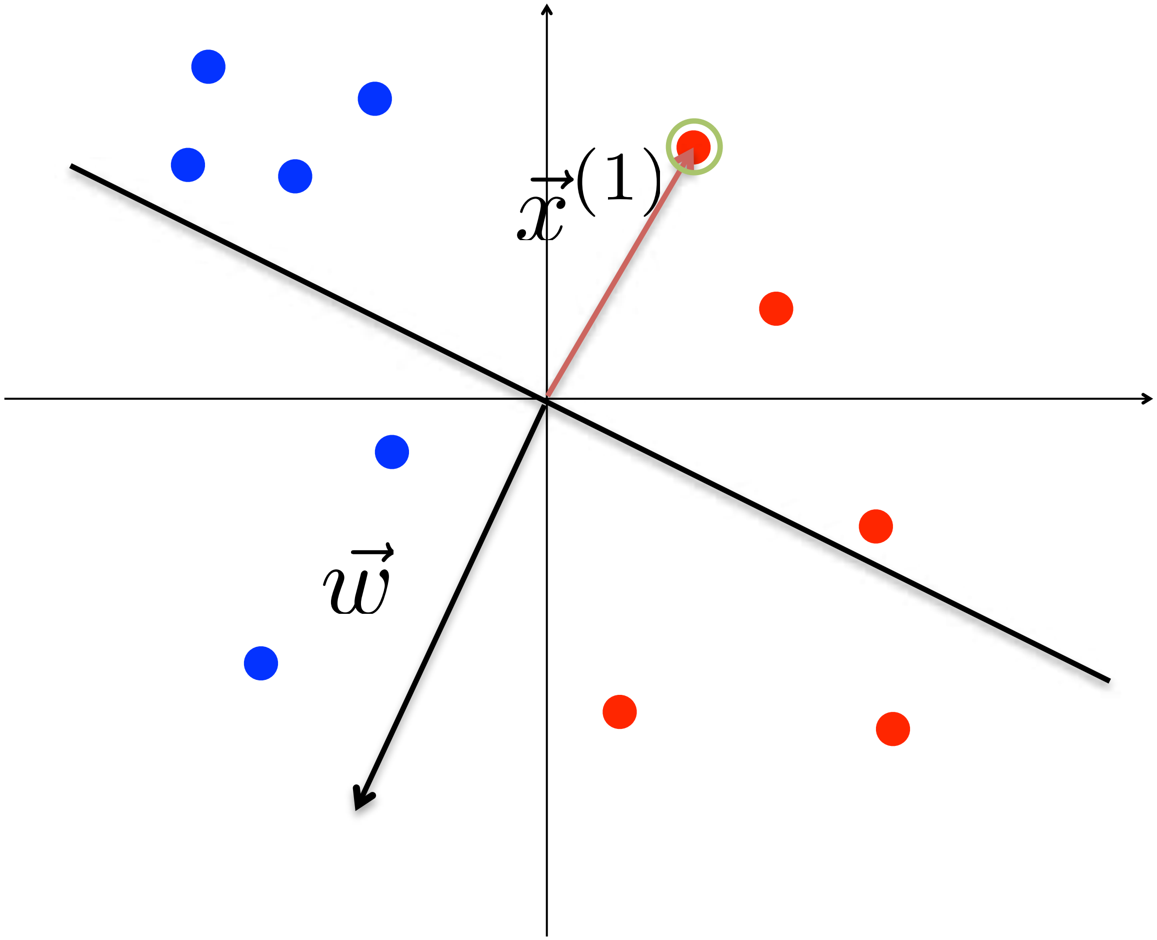
The perceptron learning rule

Choose a misclassified pattern



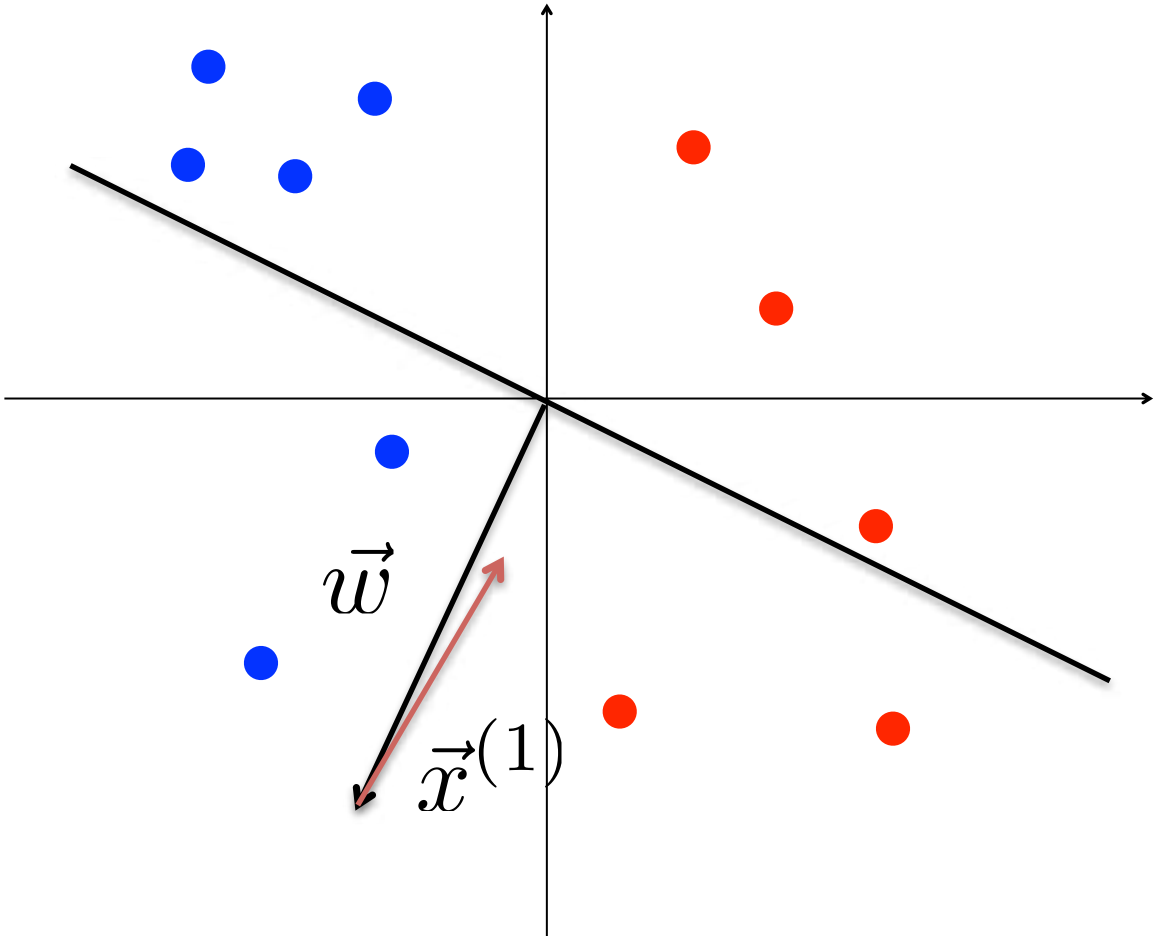
The perceptron learning rule

Update weight vector $\vec{w} = \vec{w} + \vec{x}^{(1)}$



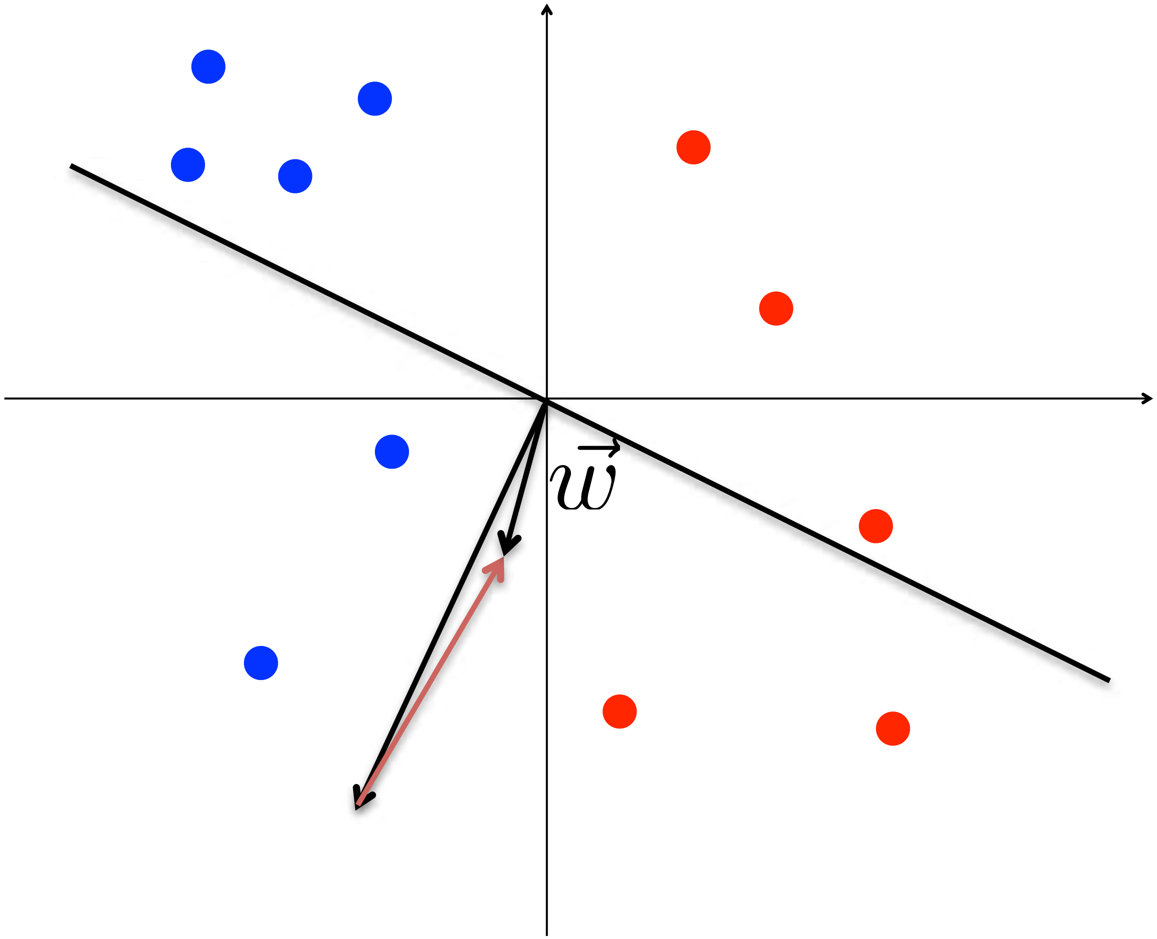
The perceptron learning rule

Update weight vector $\vec{w} = \vec{w} + \vec{x}^{(1)}$



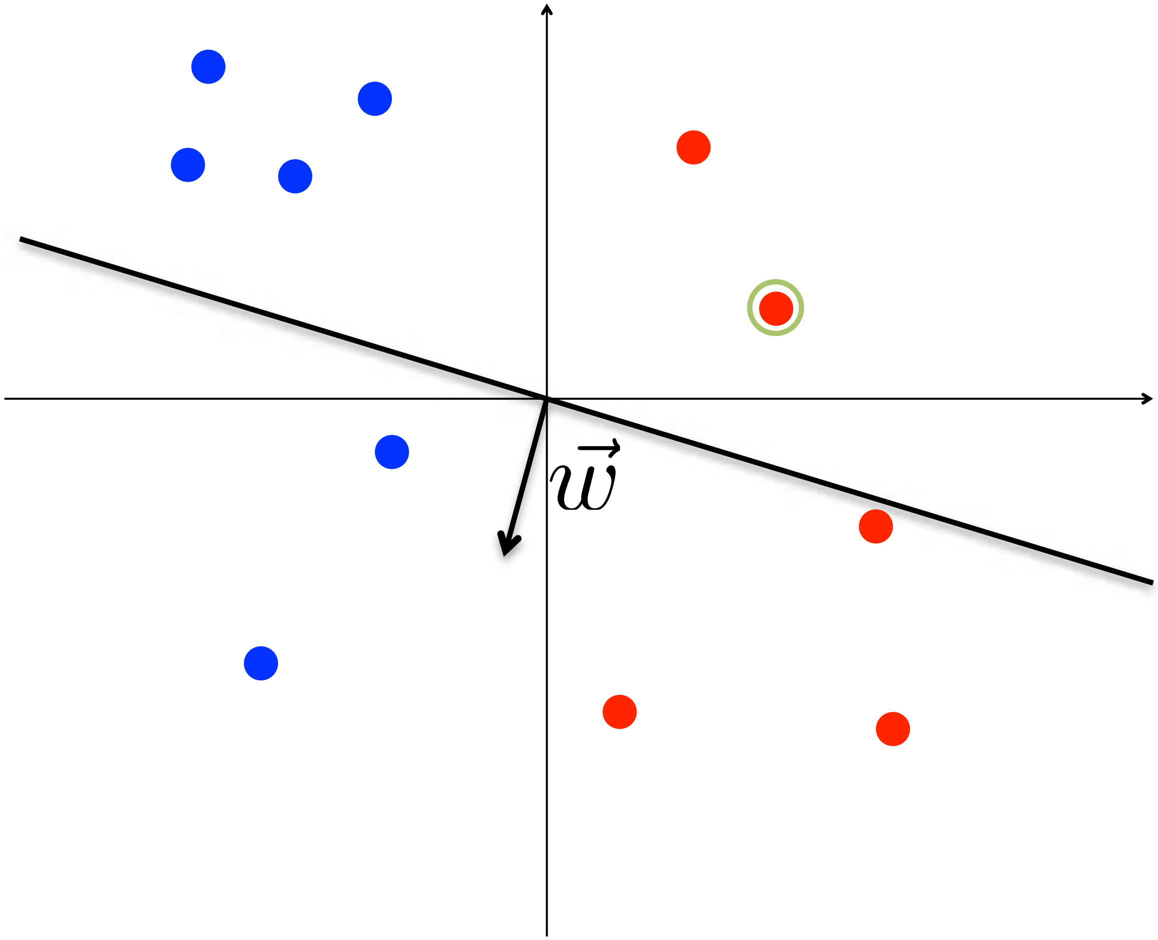
The perceptron learning rule

Update weight vector $\vec{w} = \vec{w} + \vec{x}^{(1)}$



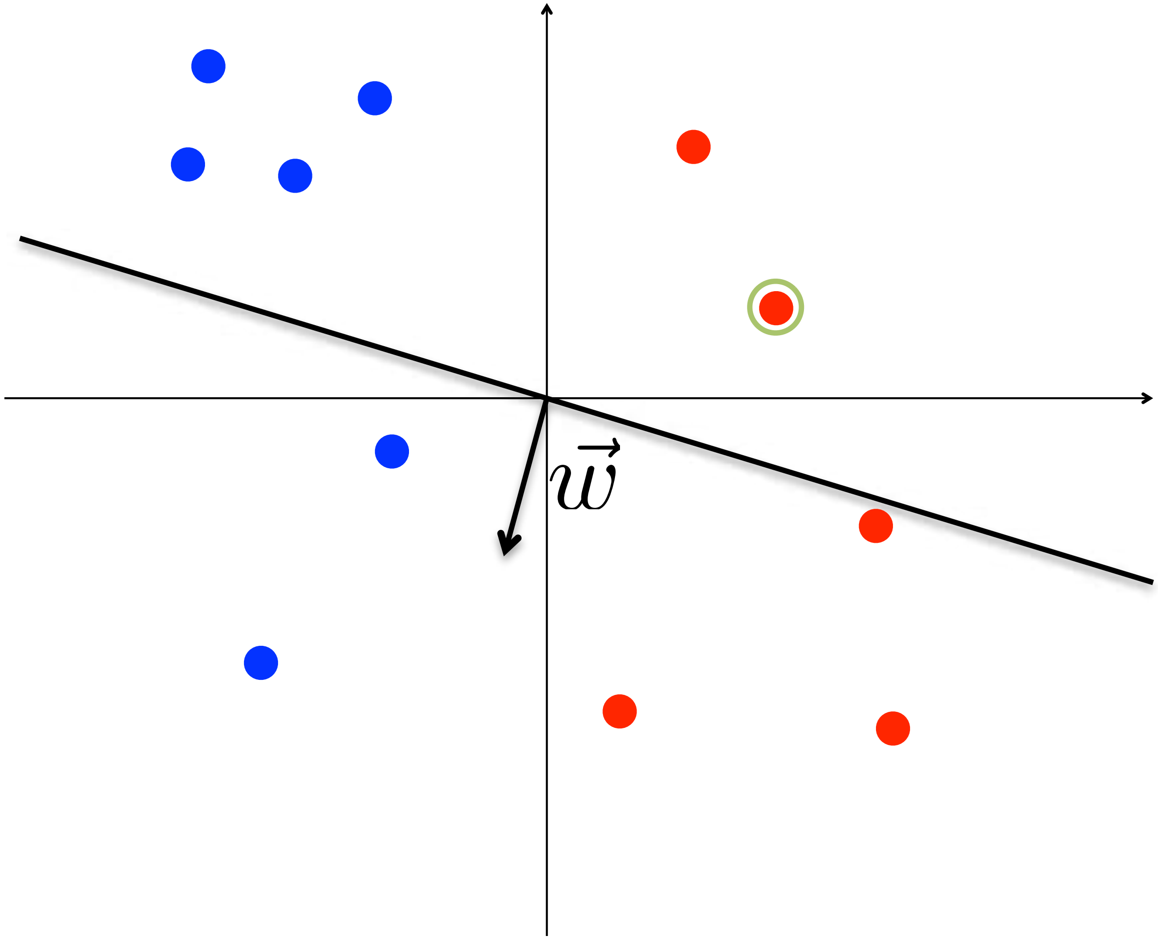
The perceptron learning rule

Update weight vector



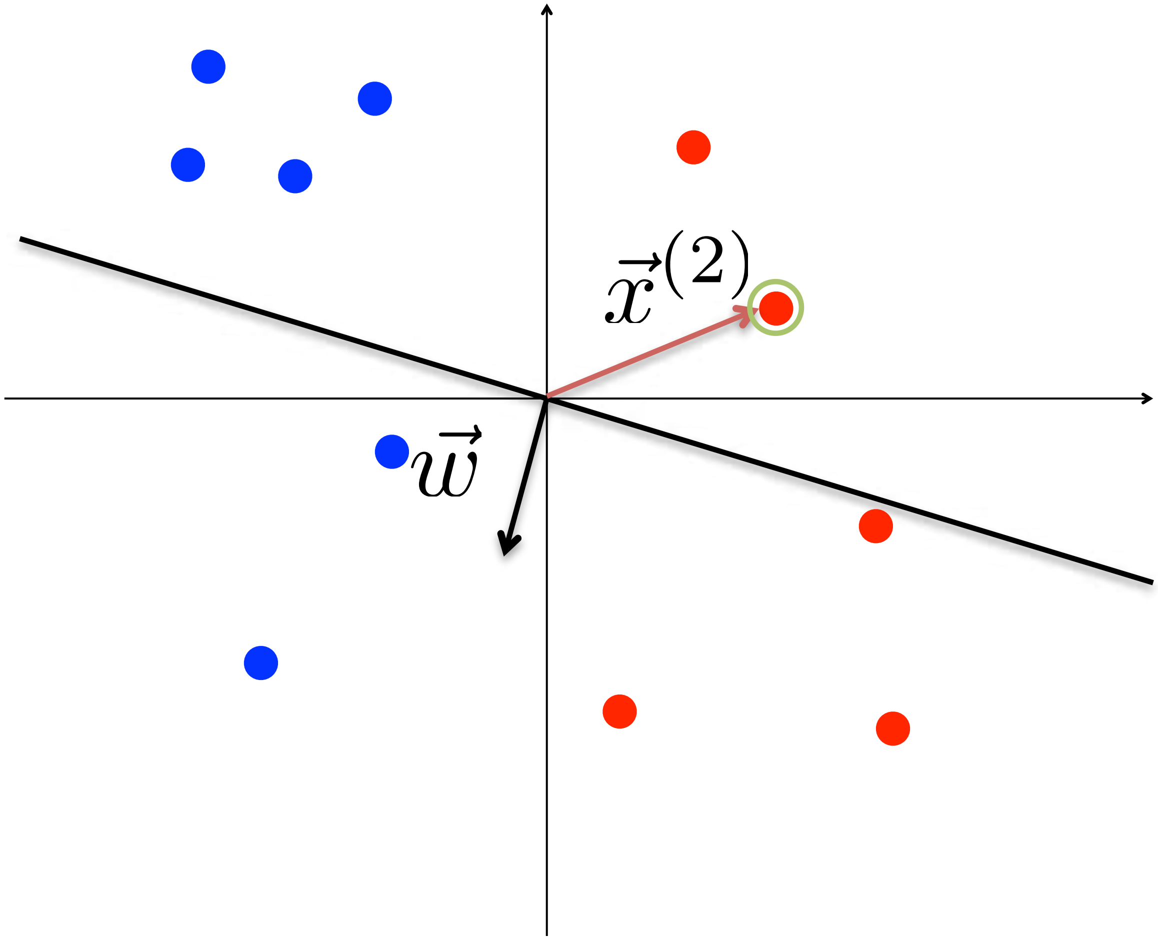
The perceptron learning rule

Choose next misclassified pattern



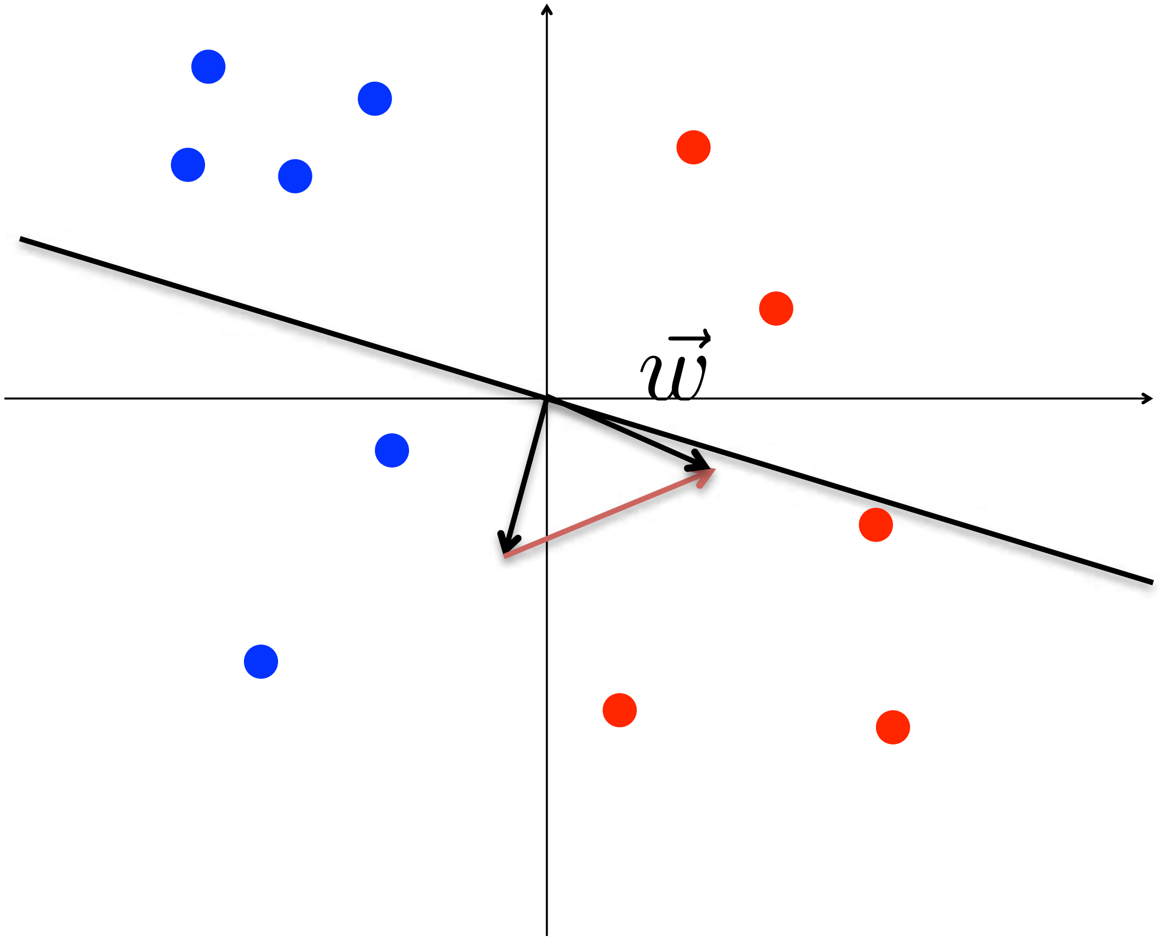
The perceptron learning rule

Choose next misclassified pattern



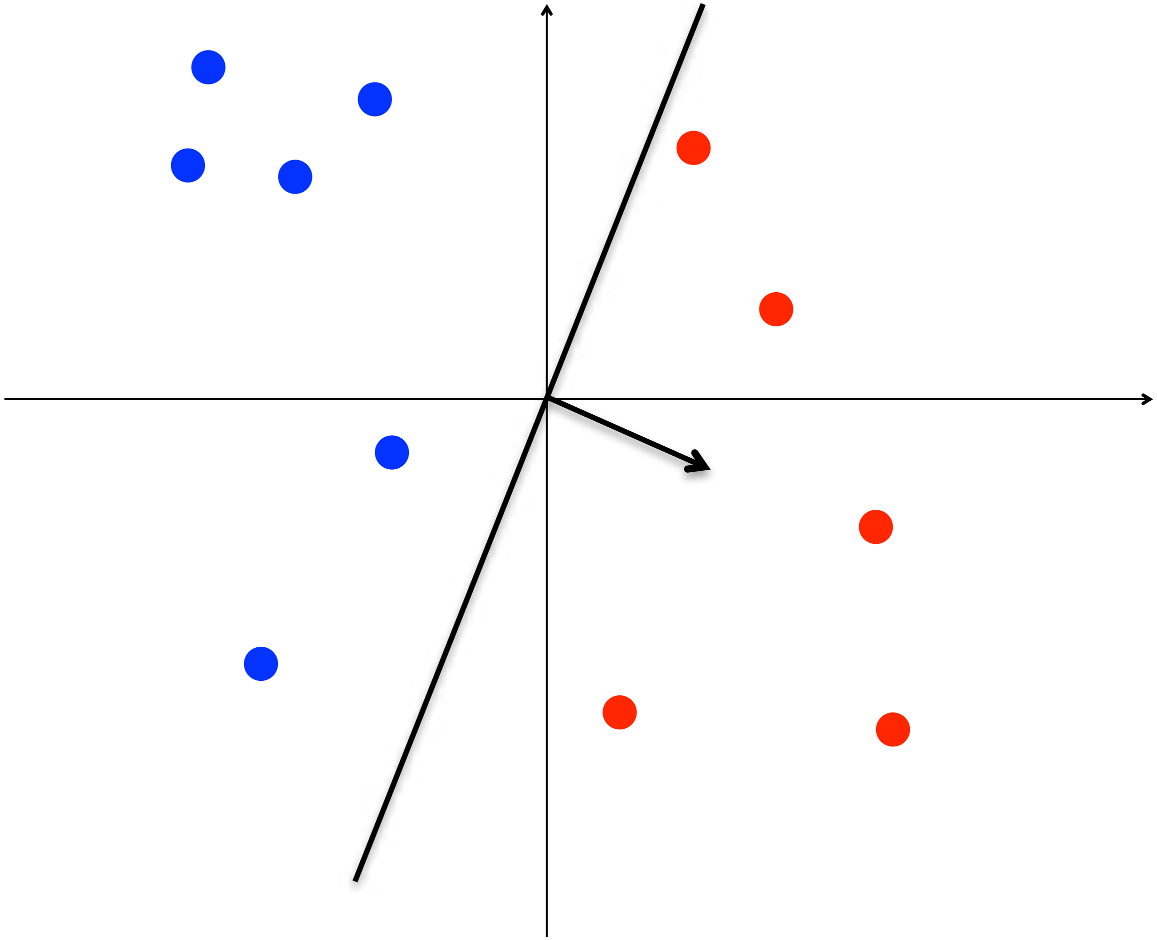
The perceptron learning rule

Update weight vector $\vec{w} = \vec{w} + \vec{x}^{(2)}$



The perceptron learning rule

Correct classification: learning terminates



What can a perceptron do?

Rosenblatt: « The perceptron may eventually be able to learn, make decisions,
and translate languages »

What can a perceptron do?

Rosenblatt: « The perceptron may eventually be able to learn, make decisions, and translate languages »

Exemple: train neurons to recognize hand-written digits

What can a perceptron do?

Rosenblatt: « The perceptron may eventually be able to learn, make decisions, and translate languages »

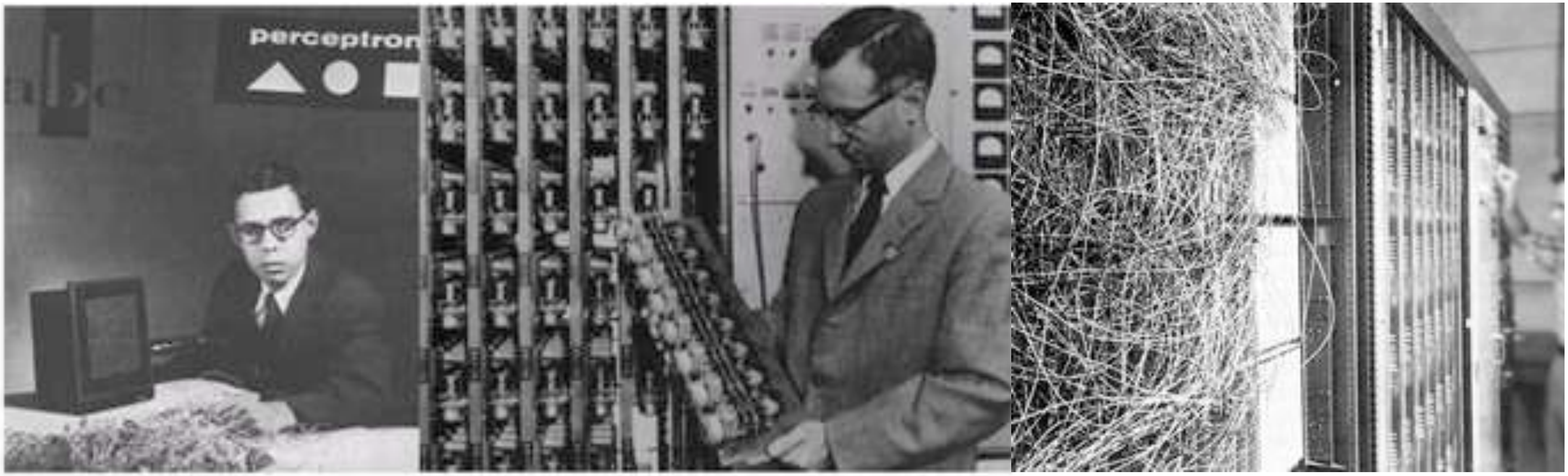
Exemple: train neurons to recognize hand-written digits

Train ten binary neurons

Inputs: vector of pixel values corresponding to digits

Neuron 3: output = 1 if input is the digit 3
output = 0 otherwise

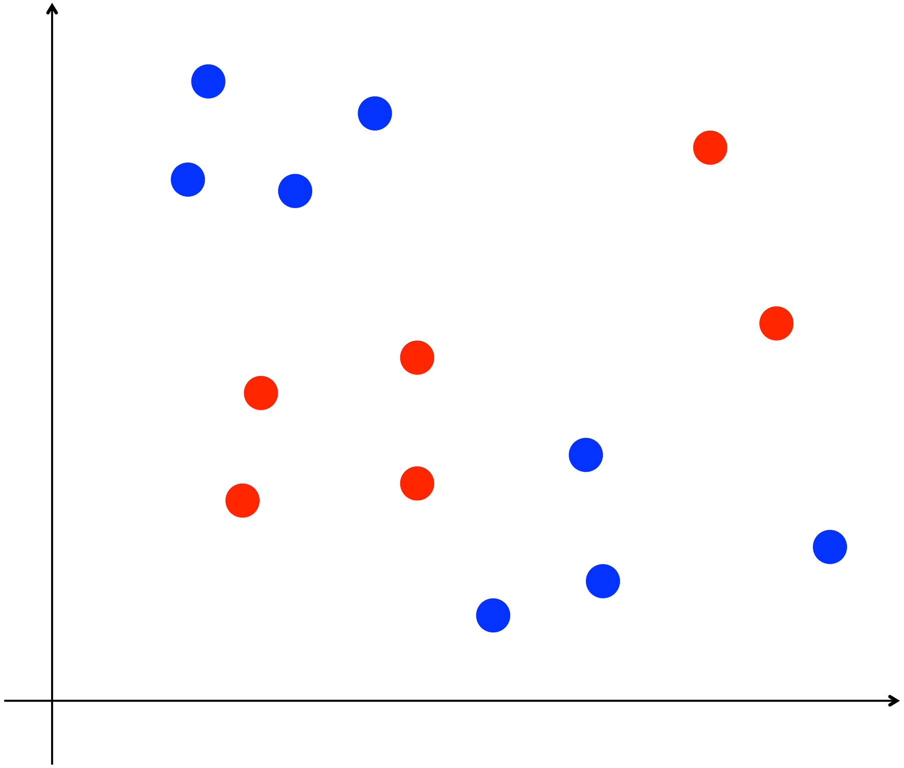
The perceptron



Frank Rosenblatt

A binary neuron can only implement linearly separable functions

Two sets are linearly separable if there exists a hyperplane separating them



Minsky and Pappert, *Perceptrons* (1969)

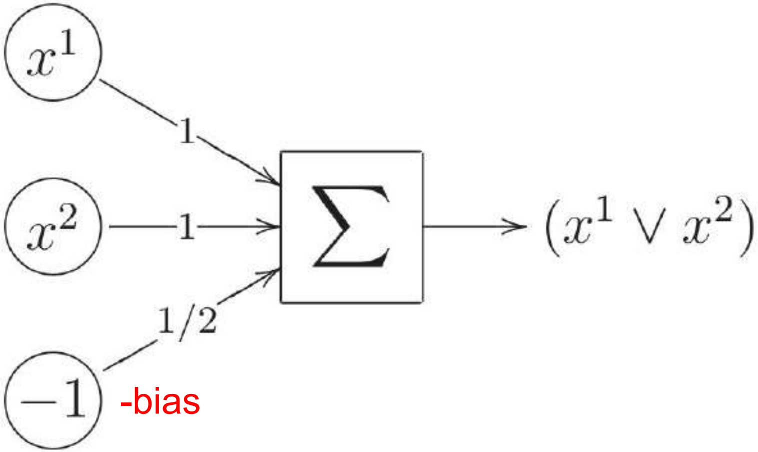
A binary neuron can only implement linearly separable functions

Marvin Minsky and Seymour Papert
Perceptrons (1969)

→ **AI winter: halt in research and funding during 10 years**

Neuron model example

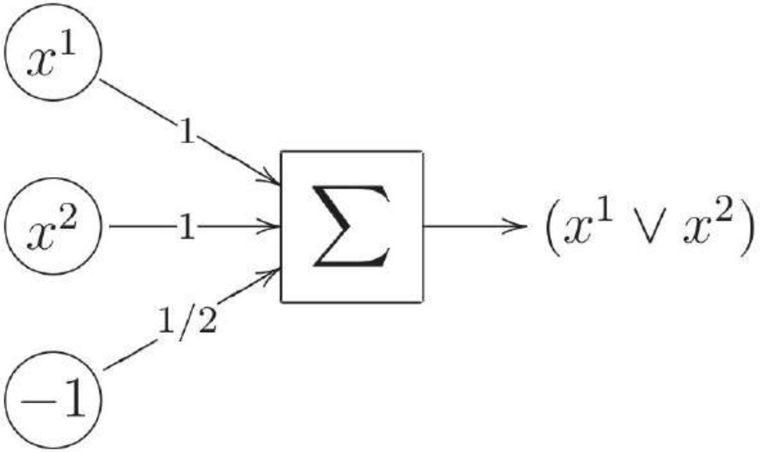
$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$

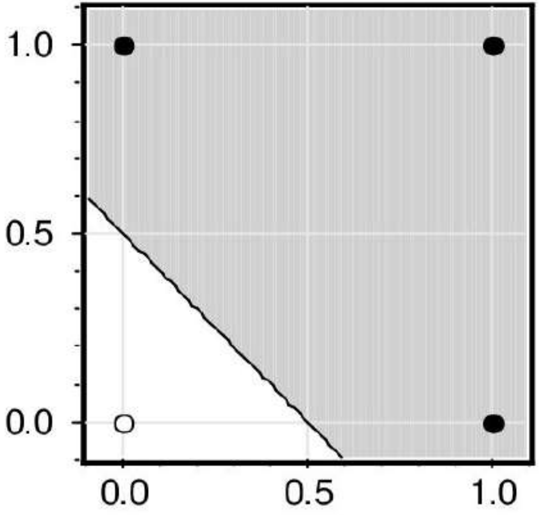


Credit: *http://www.ccas.ru/voron*

Neuron model decision boundary

$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$



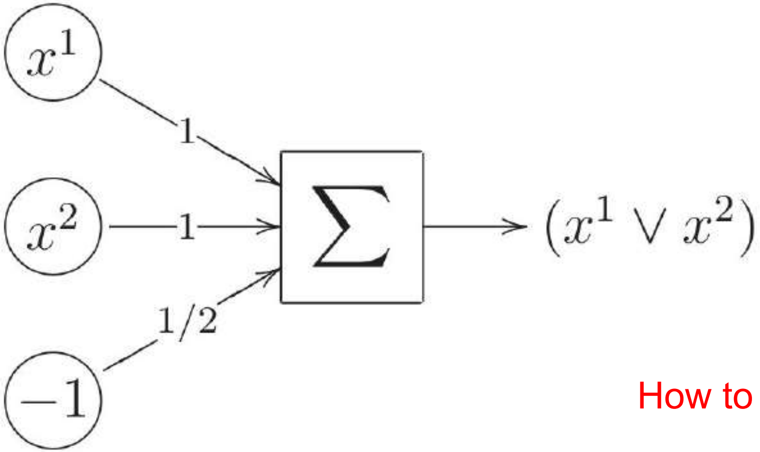


$$x^1 \vee x^2 = \left[x^1 + x^2 - \frac{1}{2} > 0\right]$$

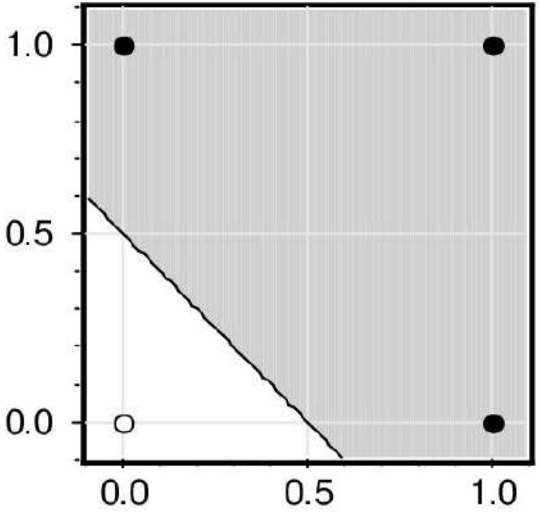
Credit: http://www.ccas.ru/voron

Neuron model decision boundary

$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$



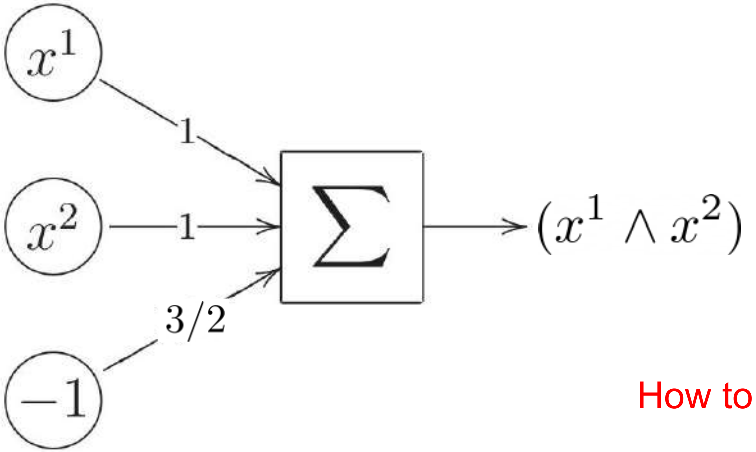
How to make **AND**?



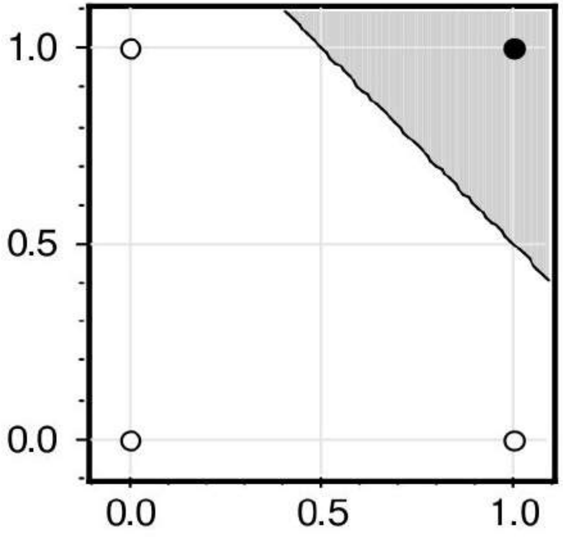
Credit: http://www.ccas.ru/voron

Neuron model decision boundary

$$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$$



How to make **AND**?

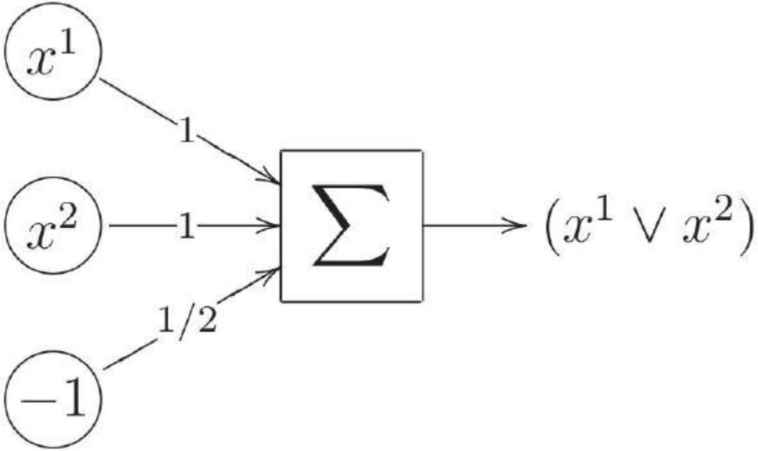


$$x^1 \wedge x^2 = \left[x^1 + x^2 - \frac{3}{2} > 0\right]$$

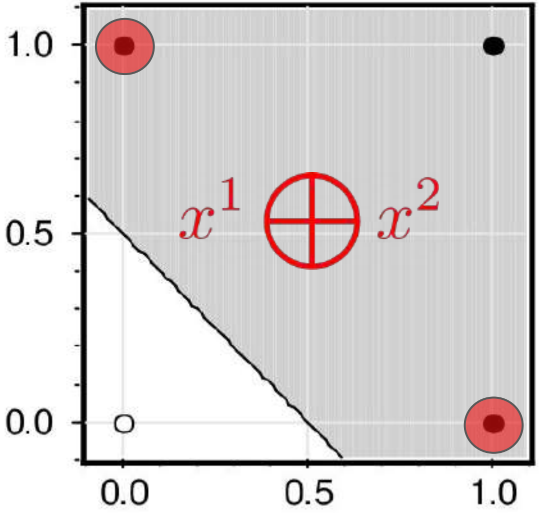
Credit: *http://www.ccas.ru/voron*

Example when it fails

$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$



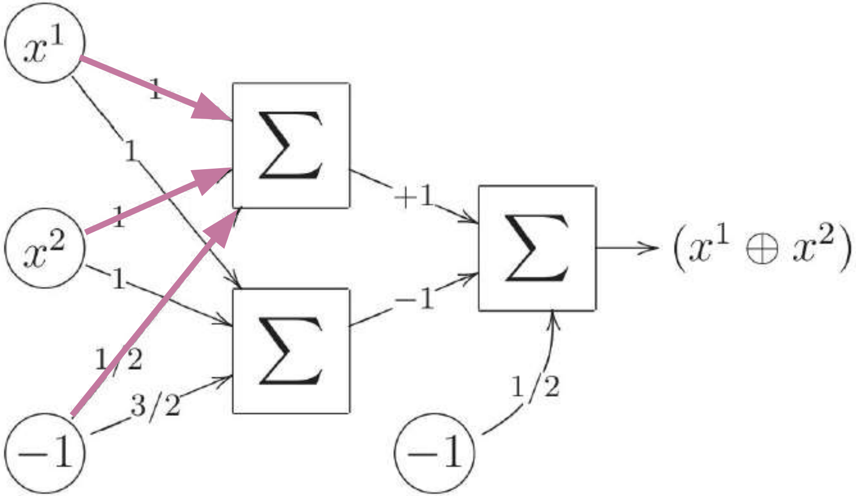
What if we want separate them?
XOR

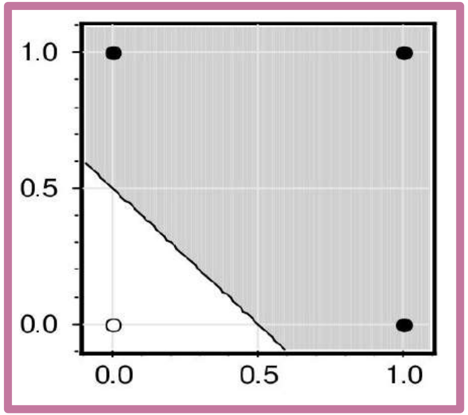


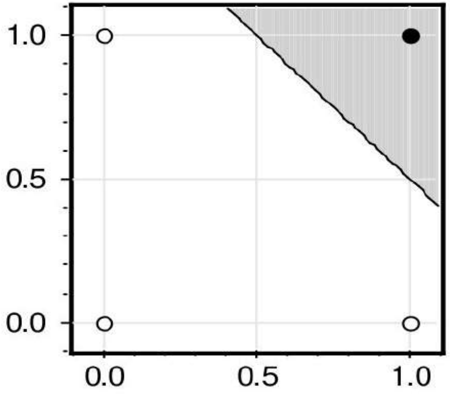
Credit: *http://www.ccas.ru/voron*

Multilayer network: hierarchical decision boundary

$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$



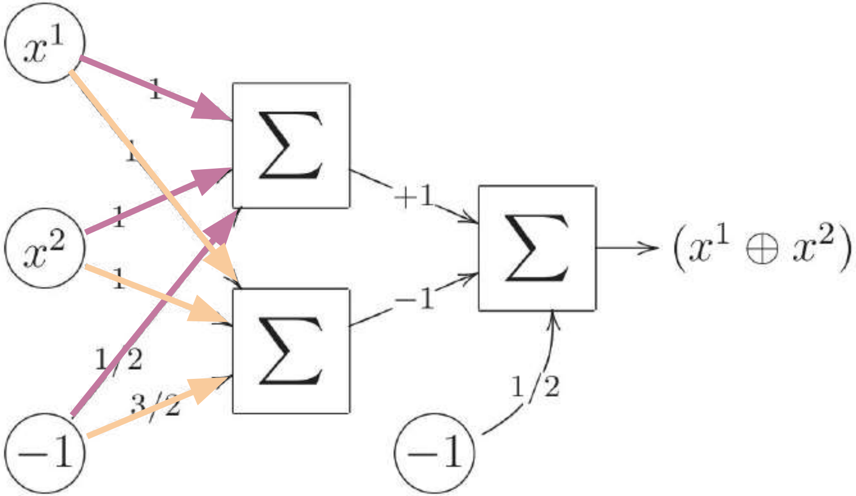


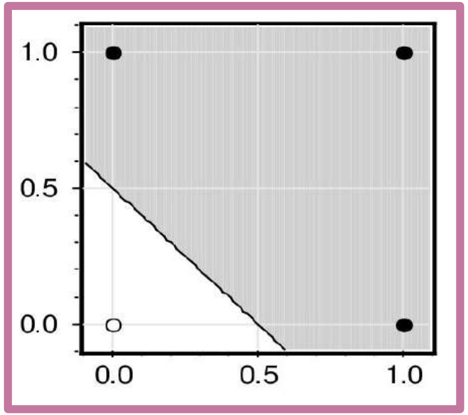


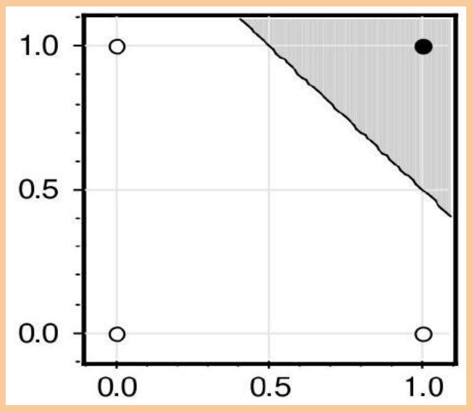
Credit: *http://www.ccas.ru/voron*

Multilayer network: hierarchical decision boundary

$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$



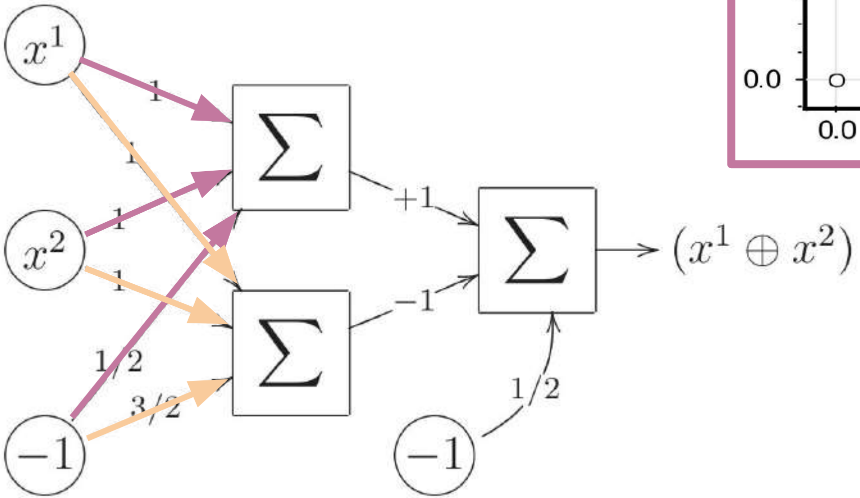


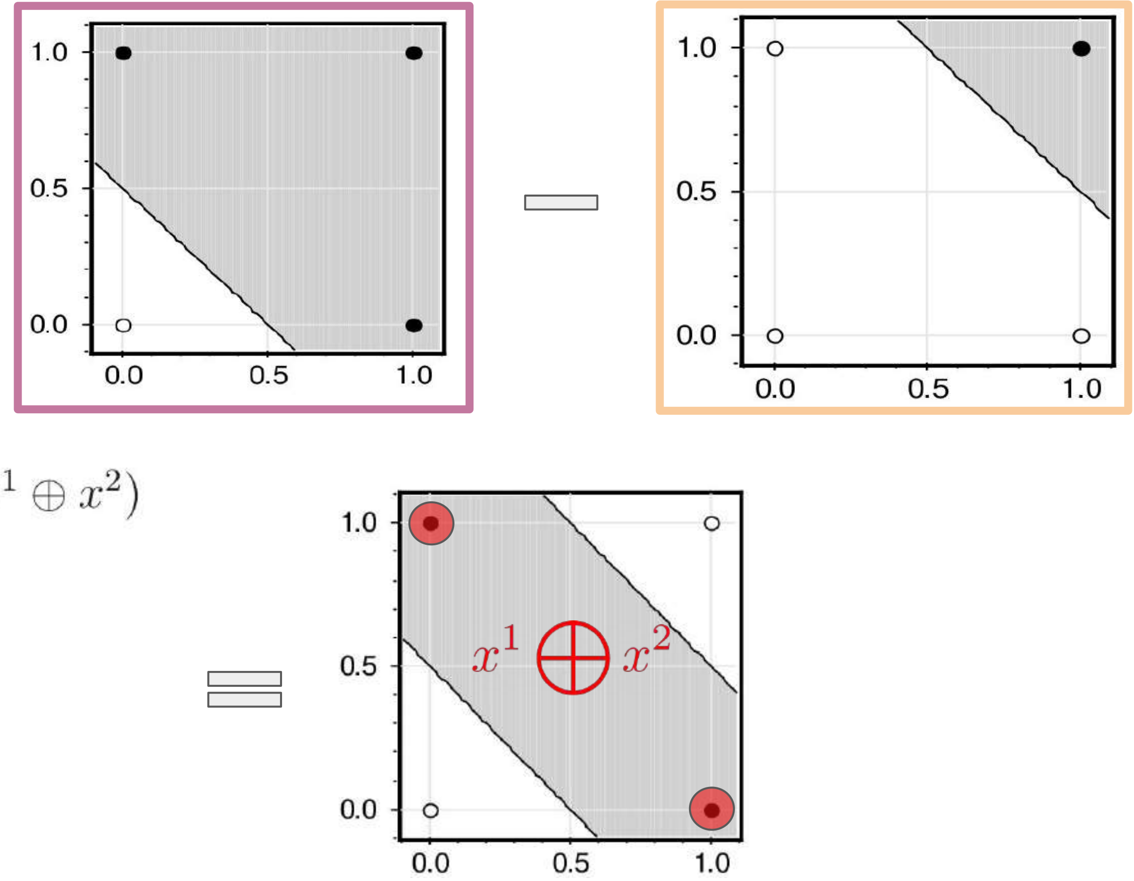


Credit: *http://www.ccas.ru/voron*

Problem fixed!

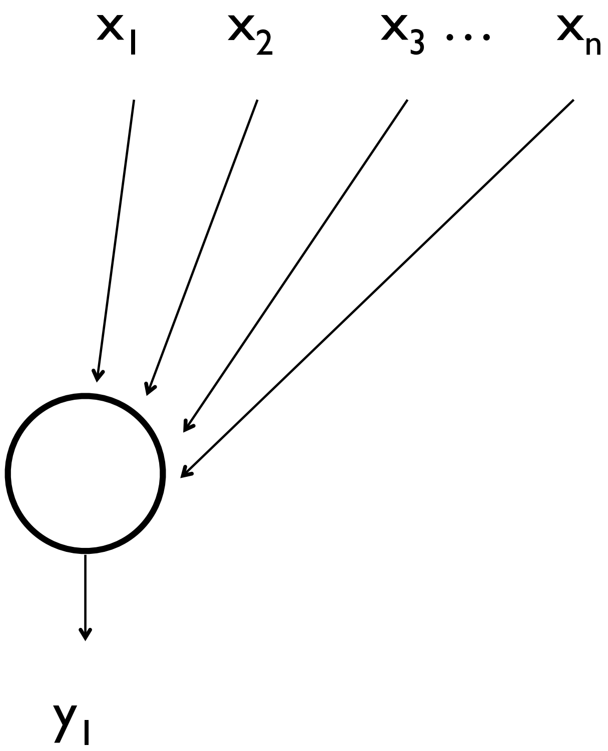
$x \in \mathbb{R}^2 \quad x^i \in \{0, 1\}$





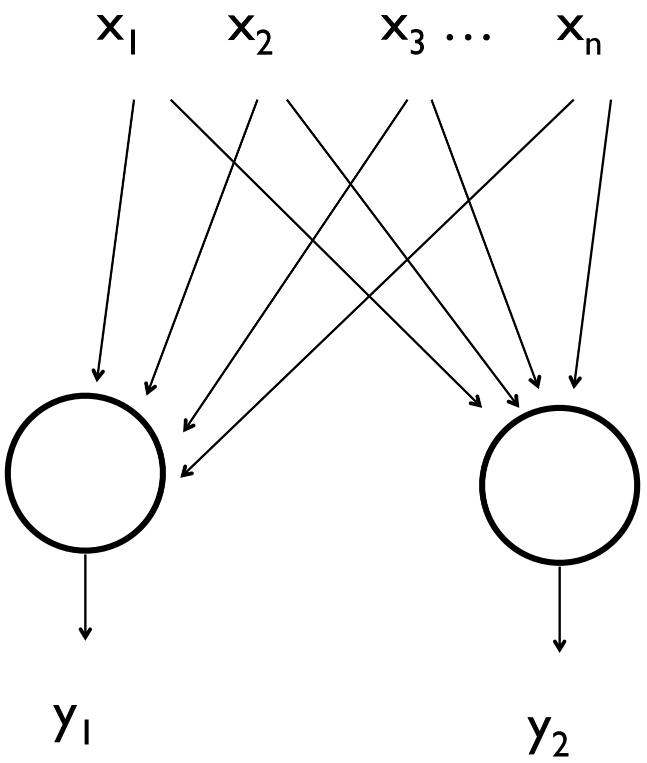
Credit: *http://www.ccas.ru/voron*

What can binary neurons compute?

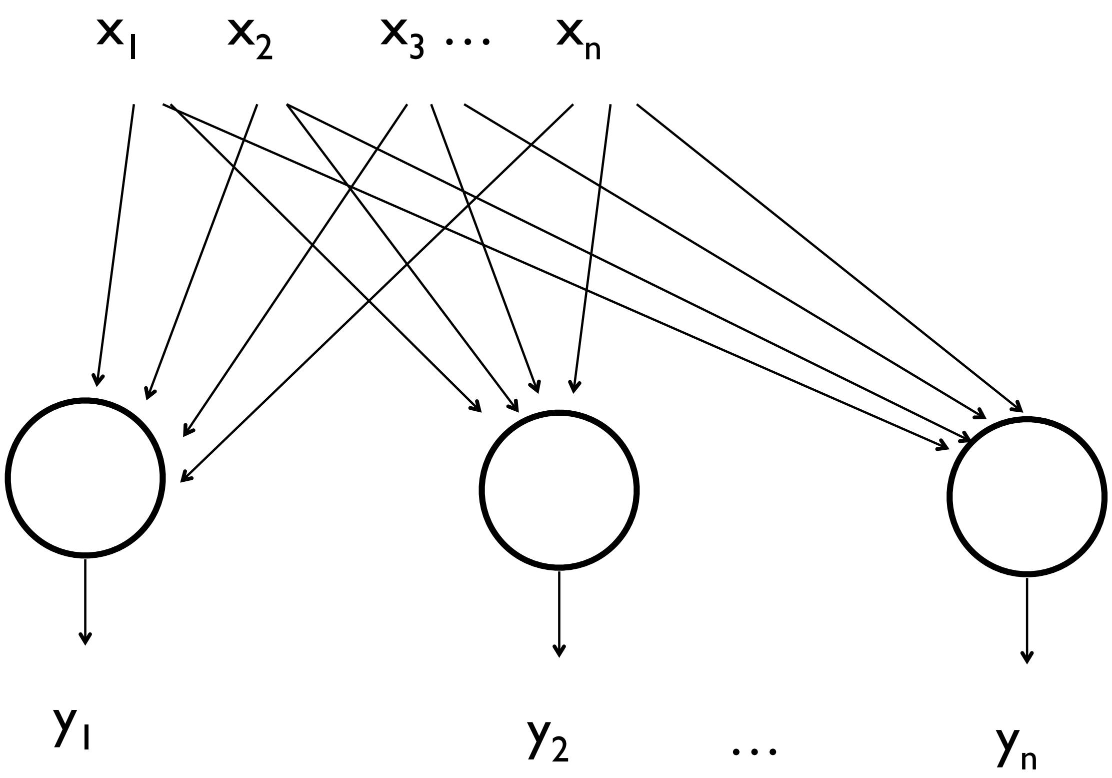


→ **Single binary neurons can compute only linearly separable functions**

What can feedforward networks compute?

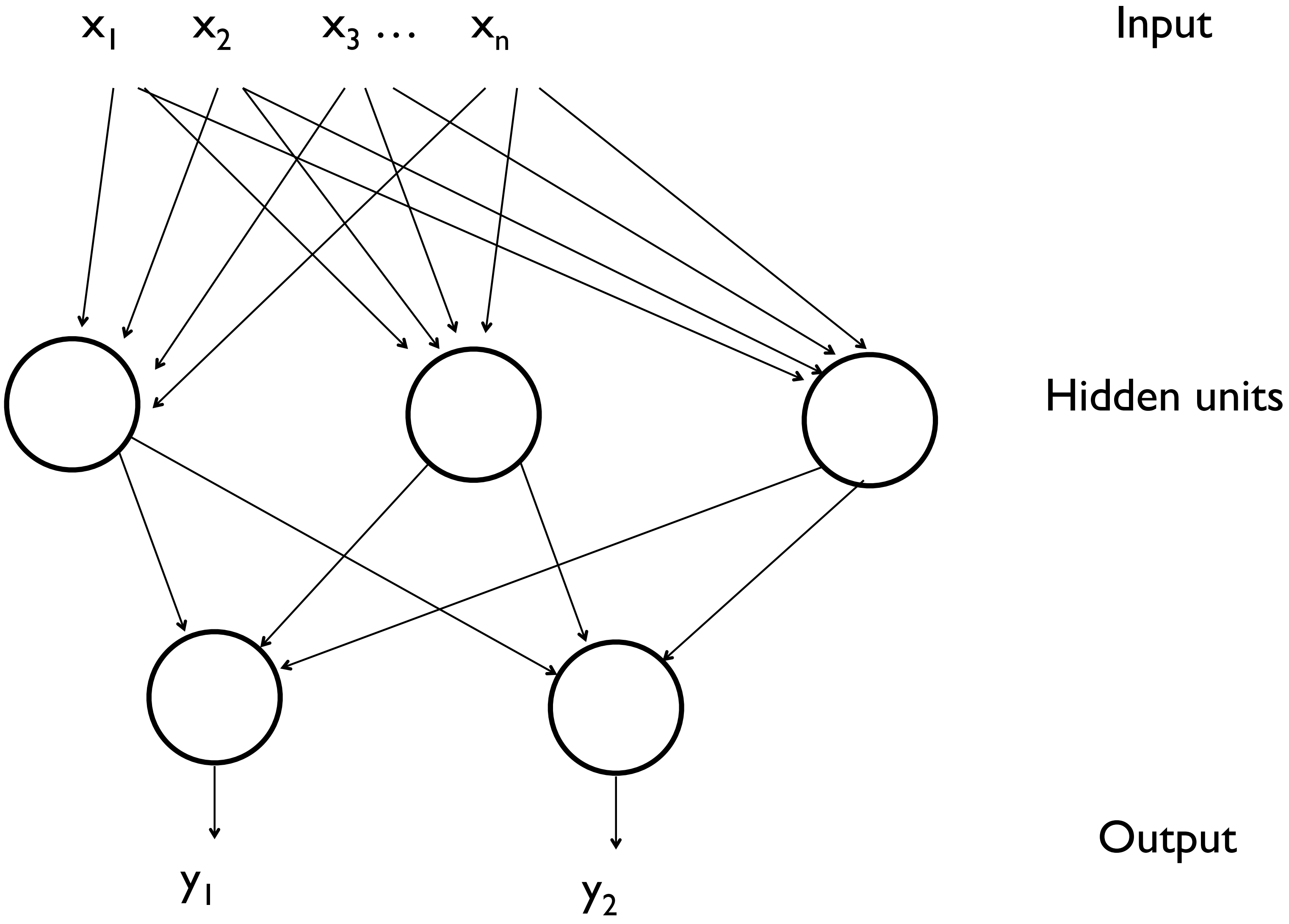


What can feedforward networks compute?



⟶ **Single layer networks can compute only linearly separable functions**

What can multilayer feedforward networks compute?



Multilayer networks can compute any binary function!

Any binary function $f : \{0,1\}^n \rightarrow \{0,1\}$

can be represented using only ANDs and ORs

[disjunctive normal form and conjunctive normal form]

⟶ **Multilayer networks have universal computational properties**

Multilayer networks can compute any binary function!

Any binary function $f : \{0,1\}^n \rightarrow \{0,1\}$

can be represented using only ANDs and ORs

[disjunctive normal form and conjunctive normal form]

→ **Multilayer networks have universal computational properties**

... but how to train them?

Training multilayer networks

Set of p training patterns $\{(x^{(0)}, d_0), (x^{(1)}, d_1) \ldots (x^{(p)}, d_p)\}$

Aim: minimize cost function $$E = \sum_{k=1}^{p} ||y_k - d_k||^2$$

by changing the synaptic weights in the network

⟶ Backpropagation algorithm (Rumelhart, Hinton and Williams 1986)

David Rumelhart

Geoff Hinton

Backpropagation

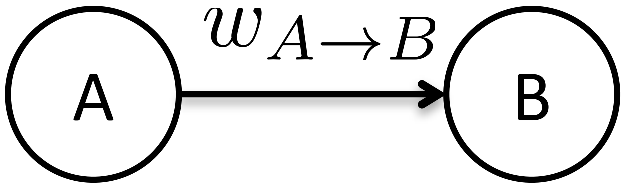
- Renaissance of Artificial Neural Networks since the 80’s, but…
- Backpropagation suggests a **retrograde propagation** along axons and synapses and would require “precise error signals that are different for each neuron, which are not accepted as likely candidates for learning processes in the brain” (Mazzoni et al. 1991)
- In other words:

  ➡ **Not compatible with biology**

Hebb's postulate

Donald Hebb

When an axon of **cell A** is near enough to excite **cell B** and repeatedly or persistently **takes part in firing it**, some growth process or metabolic change takes place in one or both cells such that A's **efficiency**, as one of the cells firing B, **is increased**. (1949)



If A and B are active at the same time, $w_{A \to B}$ increases.

The perceptron learning rule

Rosenblatt (1958)

Training set of p patterns: $\{(x^{(0)}, d_0), (x^{(1)}, d_1) \ldots (x^{(p)}, d_p)\}$

where $x^{(k)}$ is an input vector

$d_k = 0 \text{ or } 1$ is a desired output

On every step:

for each pattern k

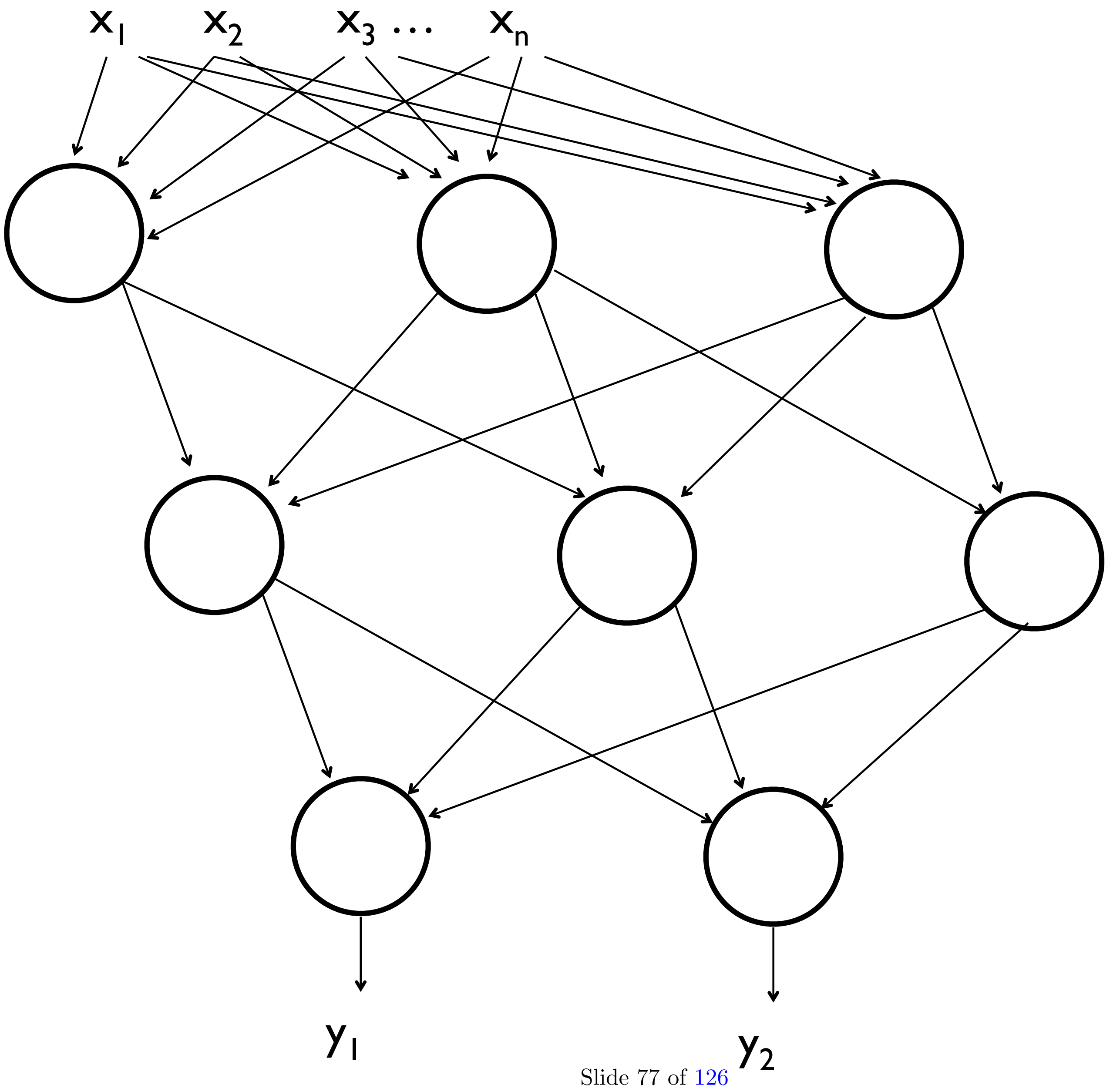
1. compute the output $y_k = H(\sum_{i=1}^{N} w_i x_i^{(k)})$

2. if $y_k \neq d_k$ update the weights:

$$w_i(t+1) = w_i(t) + \underbrace{(d_k - y_k)x_i^{(k)}}_{\textbf{input x output}}$$

→ **hebbian learning**

So far: feedforward networks

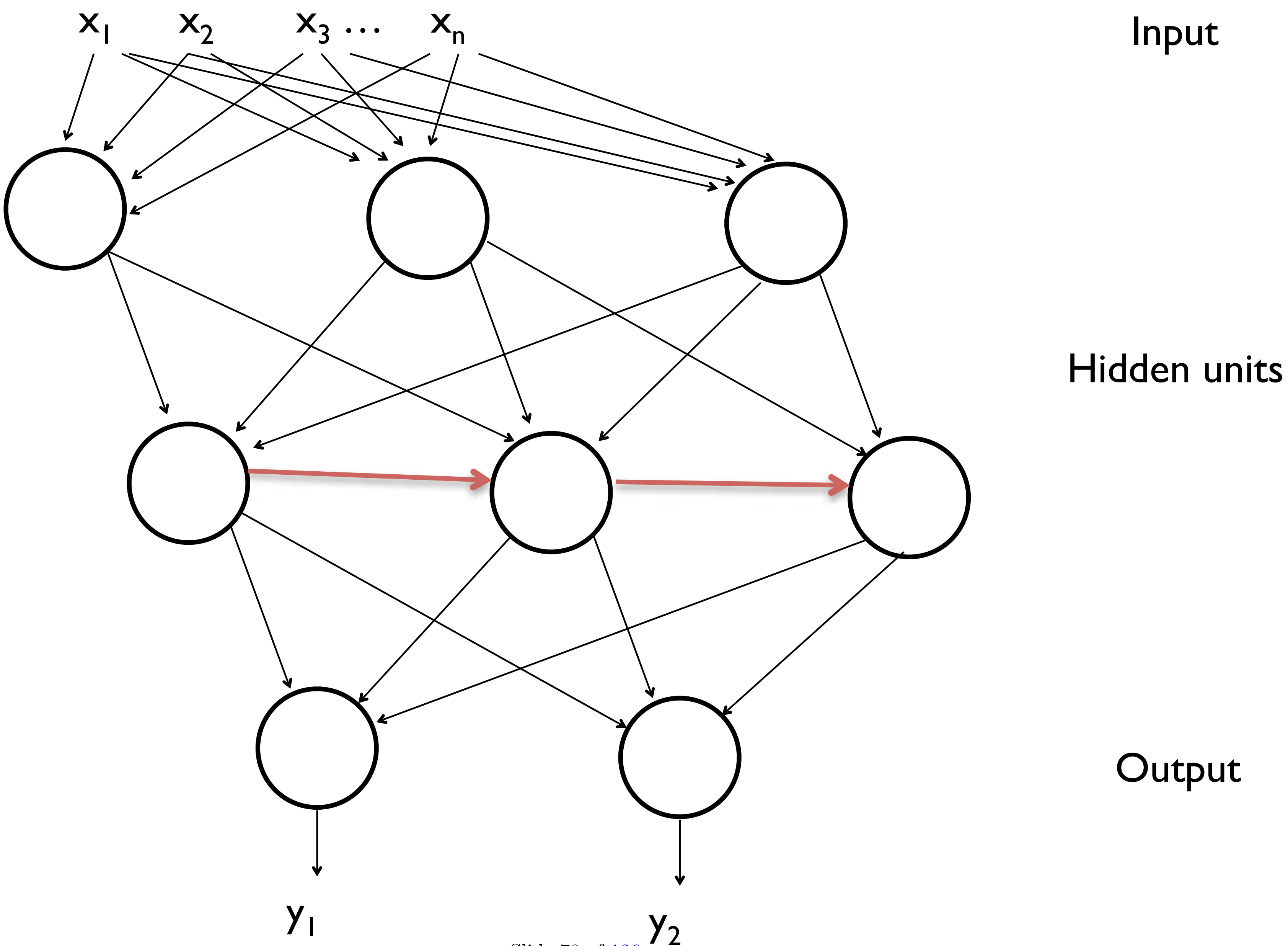


Input

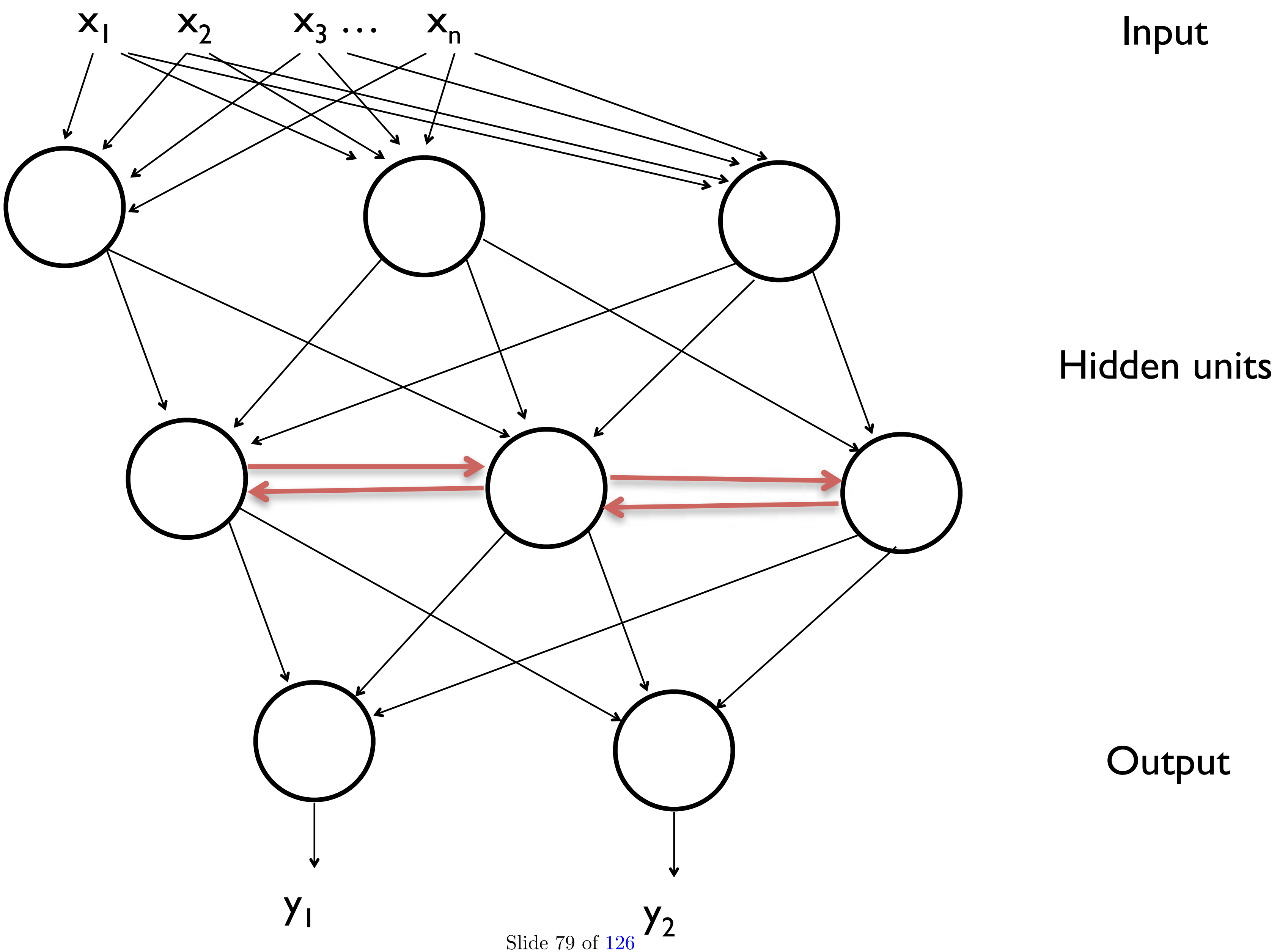
Hidden units

Output

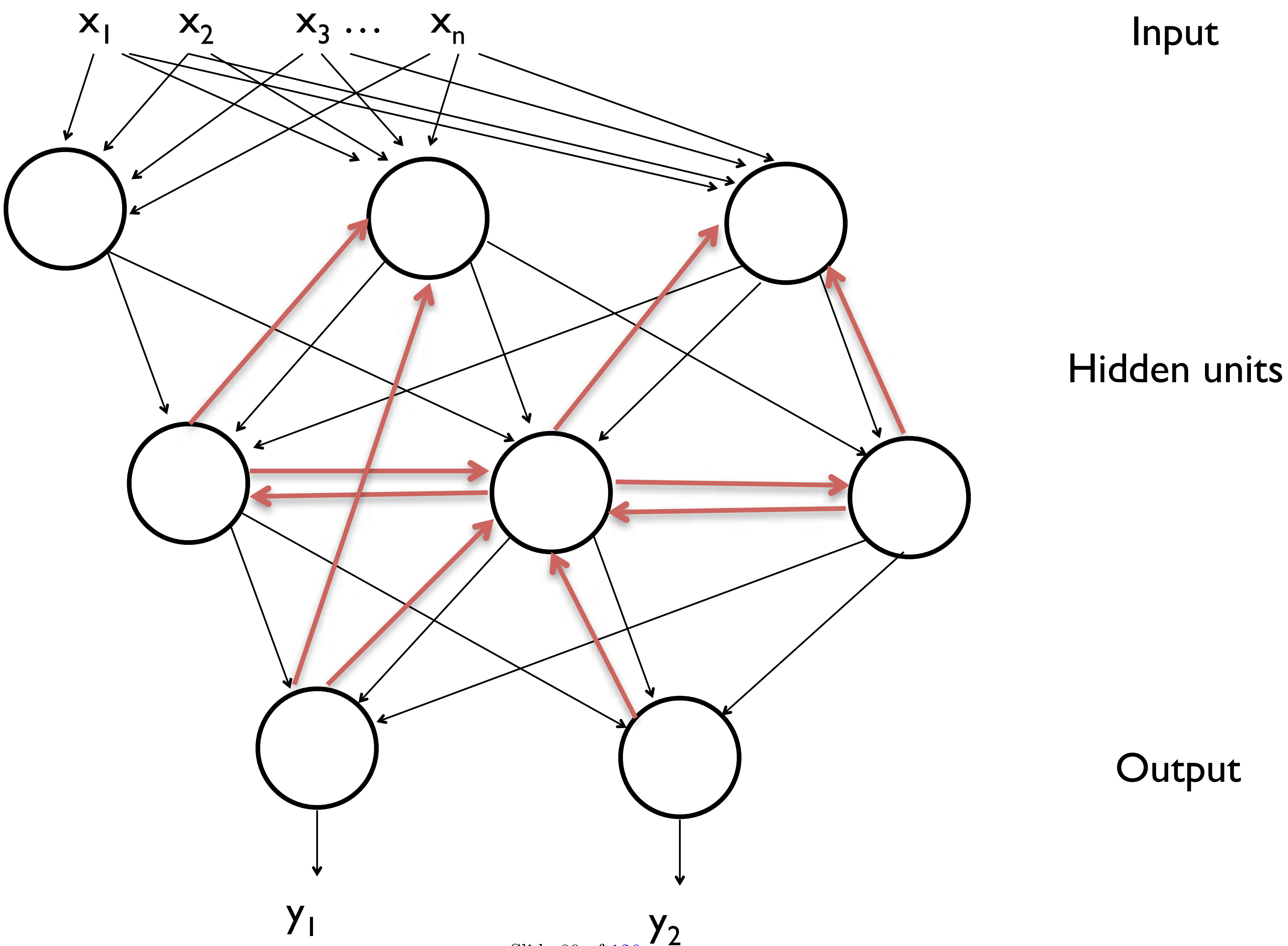
More general: recurrent connections



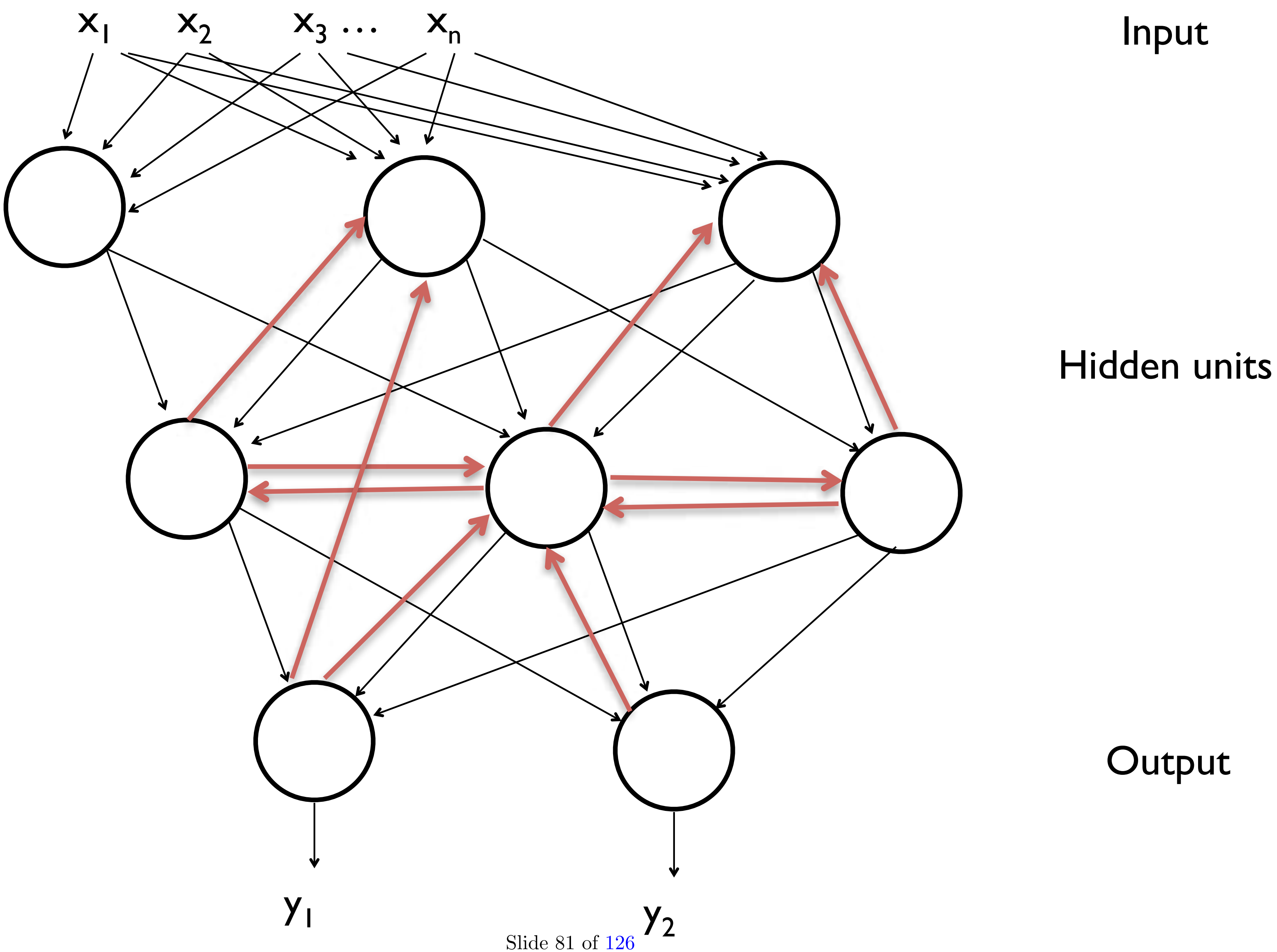
More general: recurrent connections



More general: recurrent connections



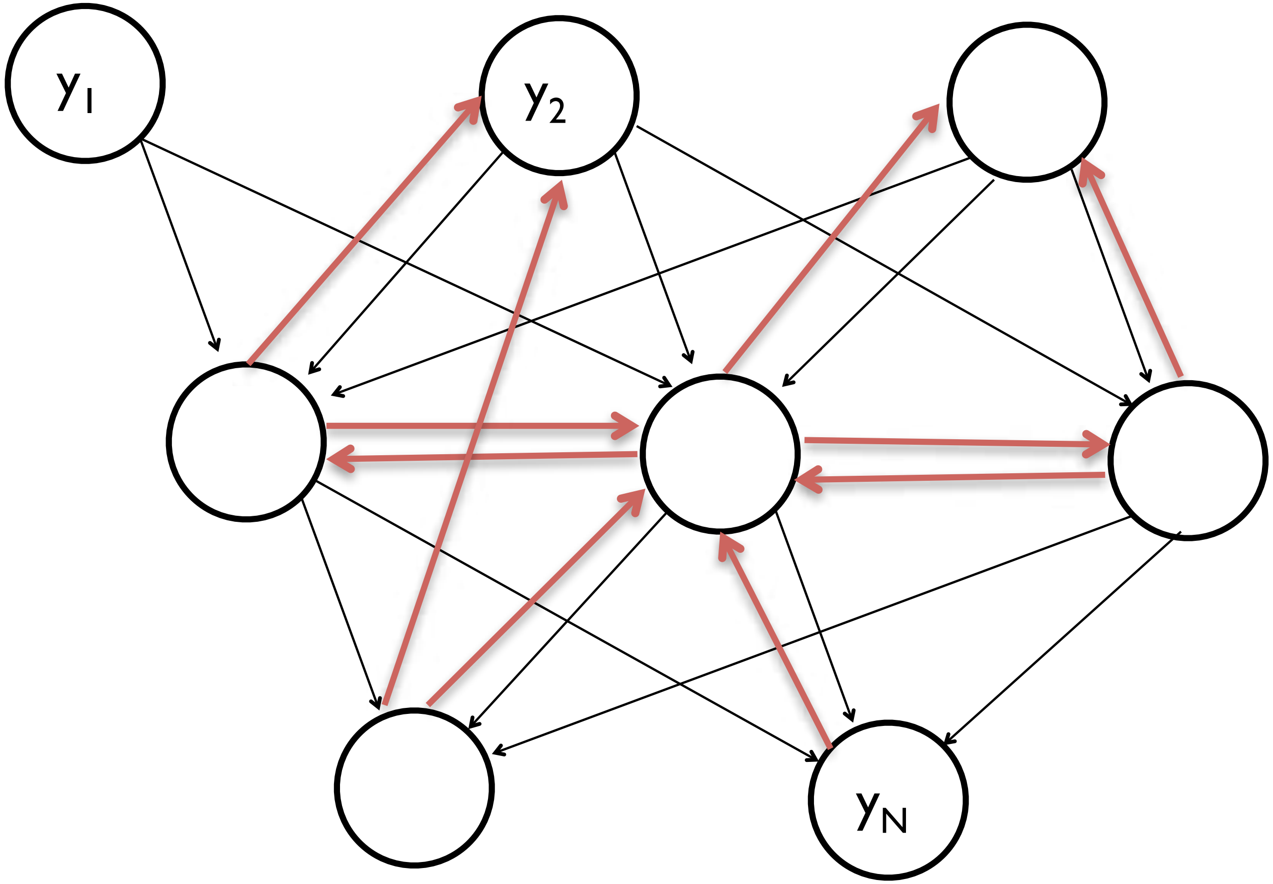
Recurrent networks: need to look at the dynamics!



Recurrent networks: need to look at the dynamics!

x_1 x_2 $x_3 \dots$ x_N

External Inputs



y_1 y_2 $y_3 \dots$ y_N

Activities

Network dynamics in discrete time

Network of N units

$$y_i(t+1) = H[\sum_{j=1}^{N} w_{ij} y_j(t) + x_i(t)]$$

Network dynamics in discrete time

Network of N units

$$y_i(t+1) = H[\sum_{j=1}^{N} w_{ij} y_j(t) + x_i(t)]$$

$$\uparrow$$

Activity of neuron I
at next timestep

Network dynamics in discrete time

Network of N units

$$y_i(t+1) = H[\underline{\sum_{j=1}^{N} w_{ij} y_j(t)} + x_i(t)]$$

Activity of neuron I
at next timestep

Total input
from network
at present timestep

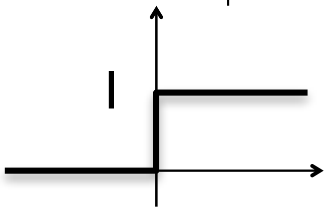
External input

Network dynamics in discrete time

Network of N units

$$y_i(t+1) = H[\underline{\sum_{j=1}^{N} w_{ij}y_j(t)} + x_i(t)]$$

Activity of neuron I
at next timestep



Total input
from network
at present timestep

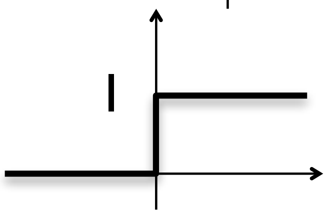
External input

Network dynamics in discrete time

Network of N units

$$y_i(t+1) = H[\underline{\sum_{j=1}^{N} w_{ij} y_j(t)} + x_i(t)]$$

Activity of neuron I
at next timestep



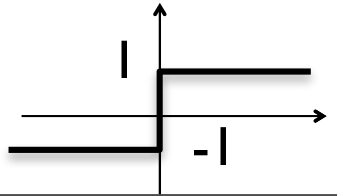
Total input
from network
at present timestep

External input

Change of notations:

$$y_i = \begin{cases} 1 & \text{active} \\ -1 & \text{inactive} \end{cases}$$

$$H[x] = \mathrm{sgn}[x]$$



Example

Network of N = 3 units

activity: $\vec{y}(t) = (y_1(t), y_2(t), y_3(t))$

synaptic matrix: $$W = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix}$$

dynamics: $\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$

Example

Network of N = 3 units

initial condition: $\vec{y}(t=1) = (1,1,1)$

synaptic matrix: $$W = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

dynamics: $$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

Dynamics - first time step

$$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

Dynamics - first time step

$$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

$$W.\vec{y}(1) = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Dynamics - first time step

$$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

$$W.\vec{y}(1) = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

$$\vec{y}(2) = sgn \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Dynamics - second time step

$$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

$$W.\vec{y}(2) = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Dynamics - second time step

$$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

$$W.\vec{y}(2) = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

$$\vec{y}(3) = sgn \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Dynamics - second time step

$$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

$$W.\vec{y}(2) = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

$$\vec{y}(3) = sgn\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \vec{y}(2)!$$

→ the activity does not evolve anymore = fixed point

Fixed points of network dynamics

External input only on first step = set initial conditions

Dynamics stop when

$$y_i(t+1) = y_i(t)$$

Fixed points of network dynamics

External input only on first step = set initial conditions

Dynamics stop when

$$y_i(t+1) = y_i(t)$$

$$y_i = sgn\left(\sum_{j=1}^{N} w_{ij} y_j\right)$$

→ **fixed point = output of the network**

Start from different initial conditon

Network of N = 3 units

initial condition: $\vec{y}(t=1) = (1, -1, -1)$

synaptic matrix: $$W = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

dynamics: $\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$

Dynamics - first time step

$$\vec{y}(t+1) = sgn\left[W.\vec{y}(t)\right]$$

$$W.\vec{y}(1) = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

$$\vec{y}(2) = sgn \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

→ same fixed point!

Attractors

Given an input (=initial condition),
the network dynamics will evolve to the closest fixed point.

Attractors

Given an input (=initial condition),
the network dynamics will evolve to the closest fixed point.

Different initial conditions can lead to the same fixed point.

Attractors

Given an input (=initial condition),
the network dynamics will evolve to the closest fixed point.

Different initial conditions can lead to the same fixed point.

→ **fixed points = attractors for the dynamics**

Attractors

Given an input (=initial condition),
the network dynamics will evolve to the closest fixed point.

Different initial conditions can lead to the same fixed point.

⟶ **fixed points = attractors for the dynamics**

attractor neural networks:
store patterns as fixed points = memories

Attractors

Given an input (=initial condition),
the network dynamics will evolve to the closest fixed point.

Different initial conditions can lead to the same fixed point.

→ **fixed points = attractors for the dynamics**

attractor neural networks:
store patterns as fixed points = memories

Fixed points depend on synaptic weights.

how to set weights to encode desired patterns?

Hopfield learning rule for recurrent networks

Set of p desired outcomes: $\{\xi^{(1)}, \xi^{(2)}, \ldots \xi^{(p)}\}$

John Hopfield

Set weights to (Hopfield 1982):

$$w_{ij} = \frac{1}{N} \sum_{k=1}^{p} \xi_i^{(k)} \xi_j^{(k)}$$

Pseudo-hebbian rule
Symmetric connections: the network possesses an energy function
Network dynamics minimize the energy function

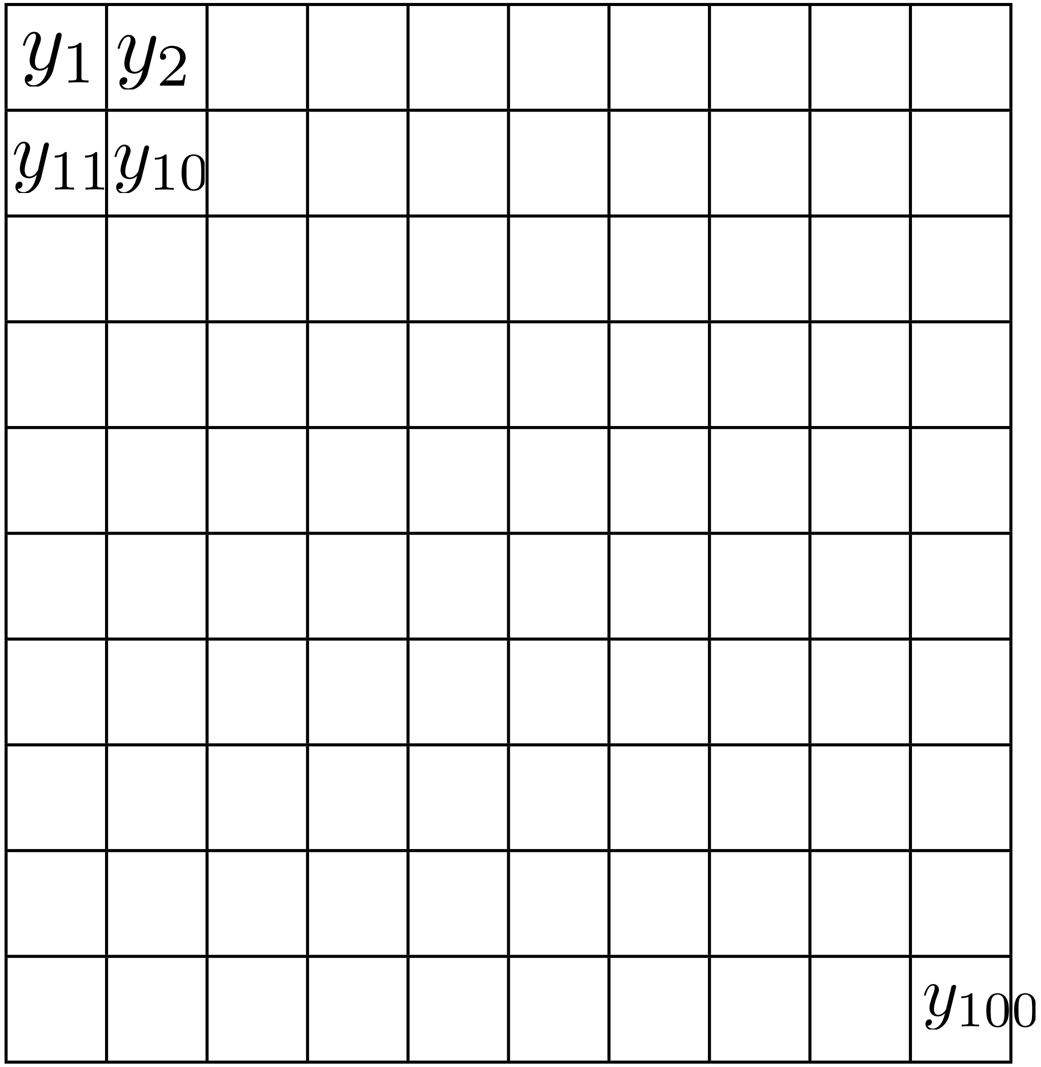
→ **Stored patterns are located at the minima of the energy function**

Exemple: network of n=100 neurons

Every box represents a neuron

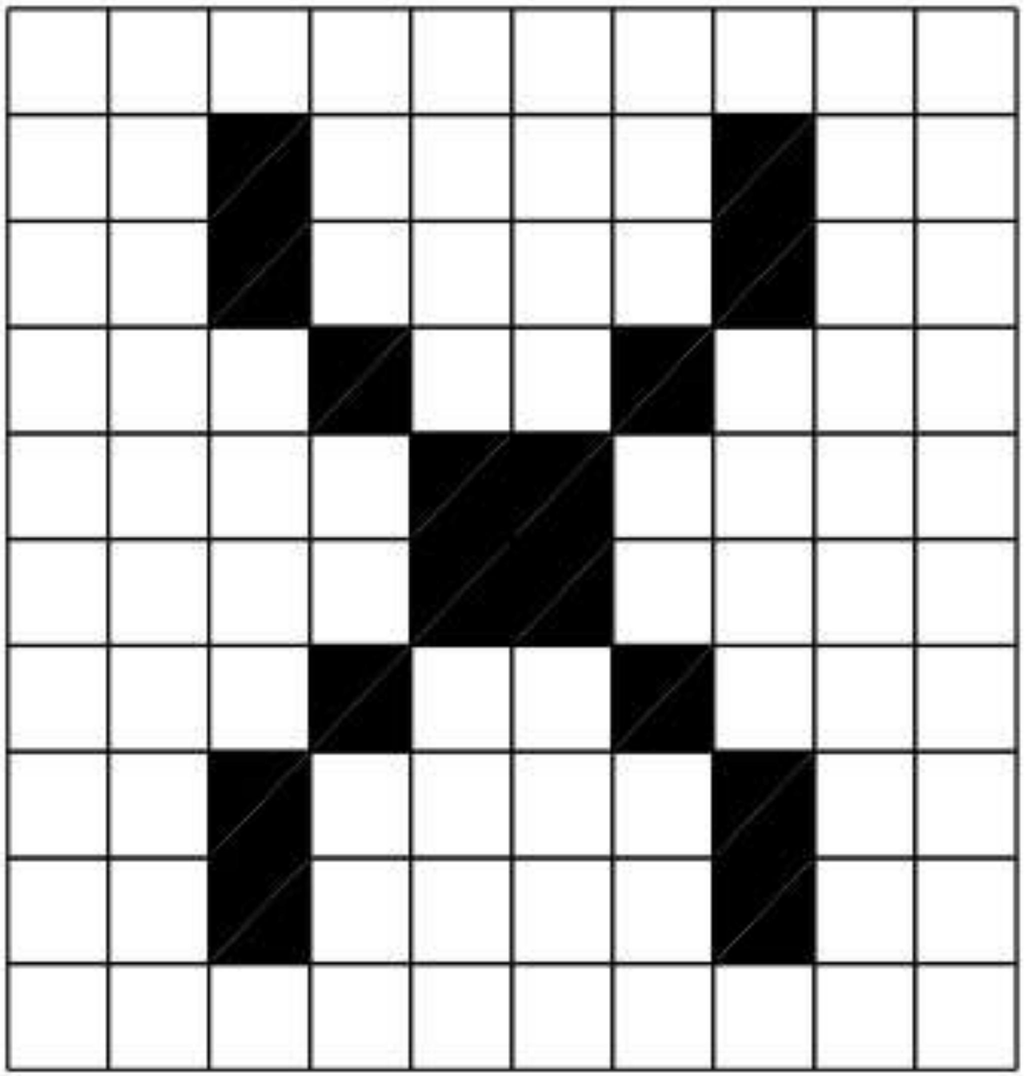
All neurons are interconnected

$$y_i = \begin{cases} 1 & \text{Black} \\ -1 & \text{White} \end{cases}$$



Exemple of learning in a recurrent network

Desired ouput $\xi^{(1)}$:

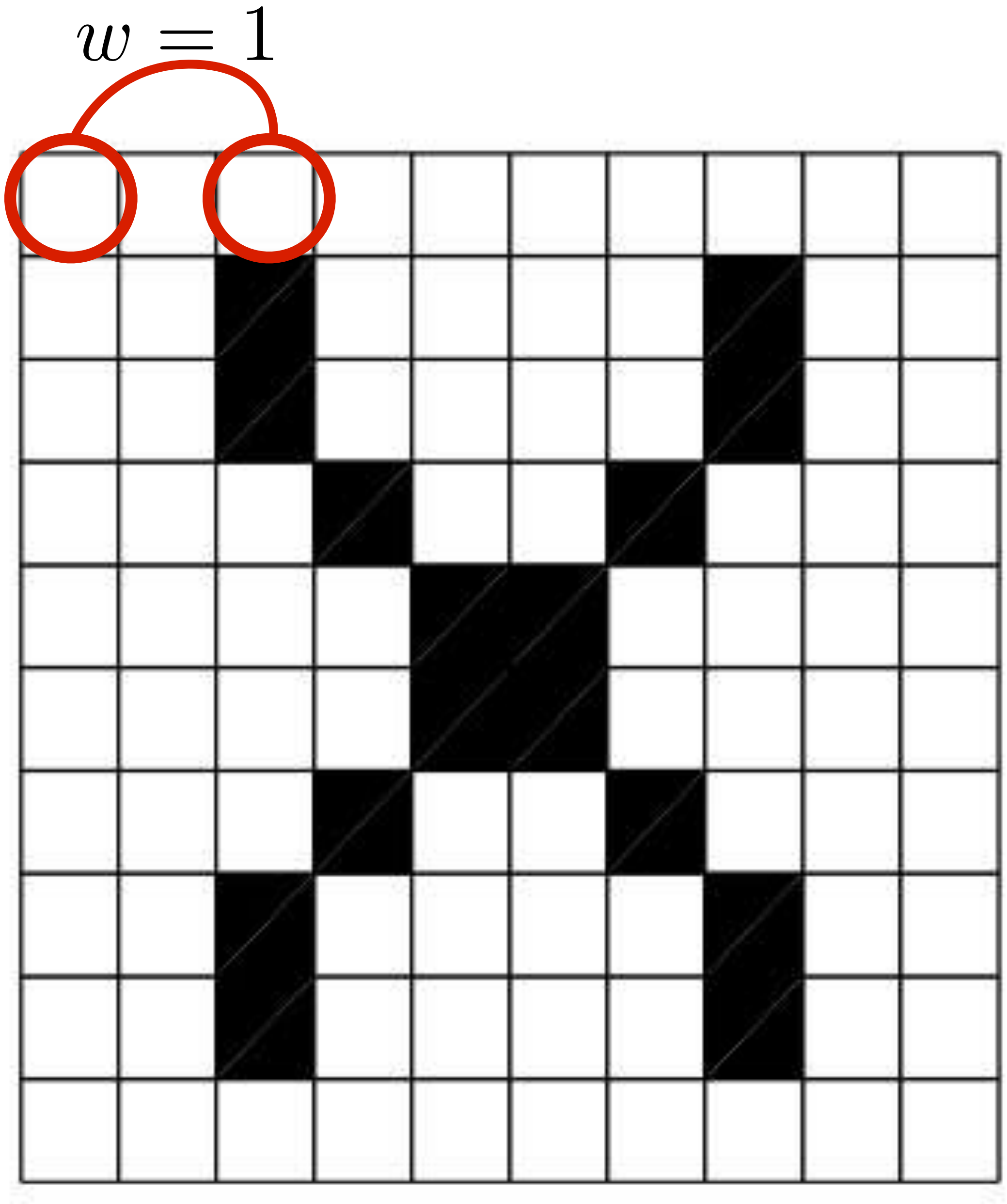


Synaptic weights:

$$w_{ij} = \frac{1}{N}\xi_i^{(1)}\xi_j^{(1)}$$

Exemple of learning in a recurrent network

Desired ouput $\xi^{(1)}$:

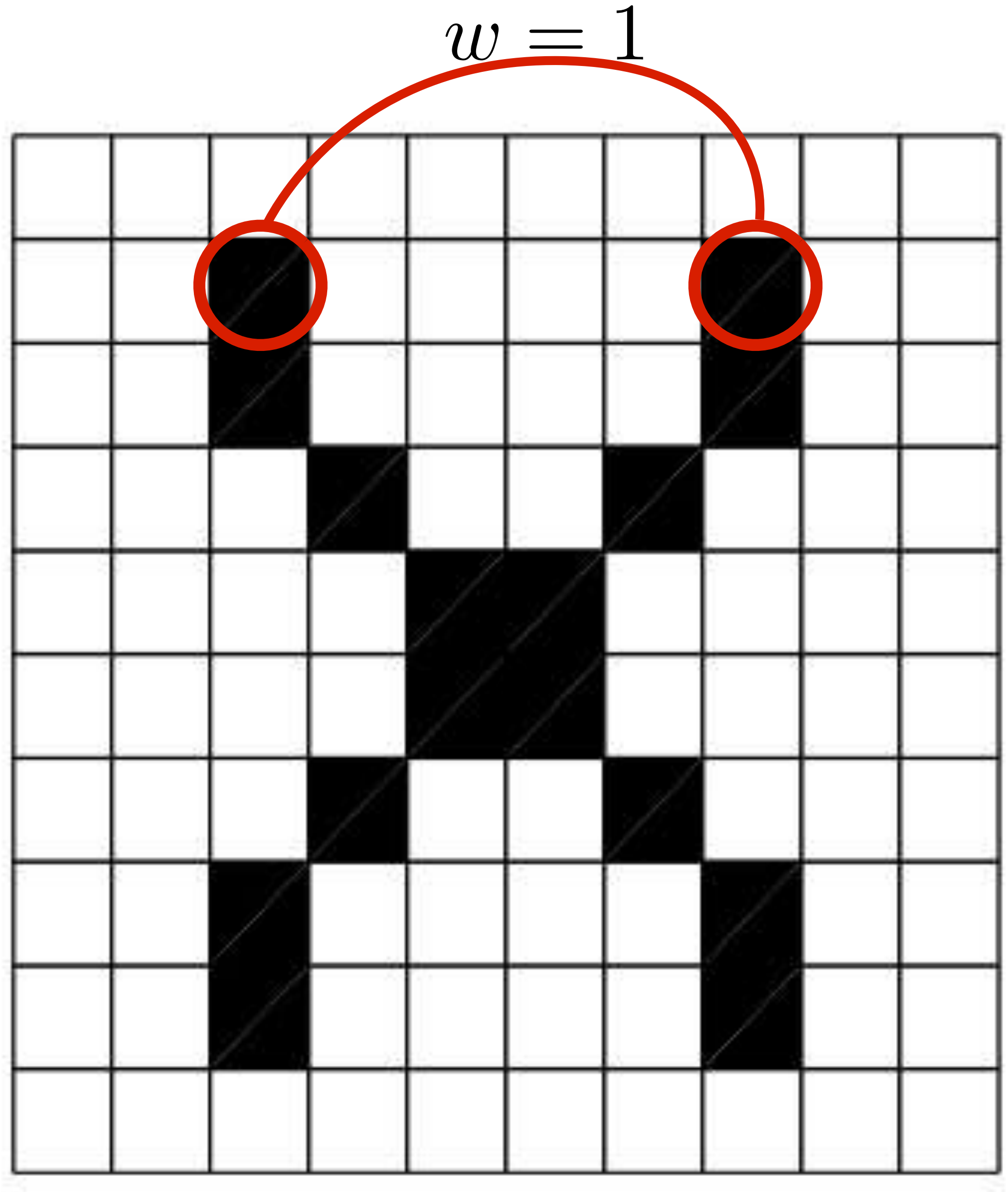


Synaptic weights:

$$w_{ij} = \frac{1}{N}\xi_i^{(1)}\xi_j^{(1)}$$

Exemple of learning in a recurrent network

Desired ouput $\xi^{(1)}$:

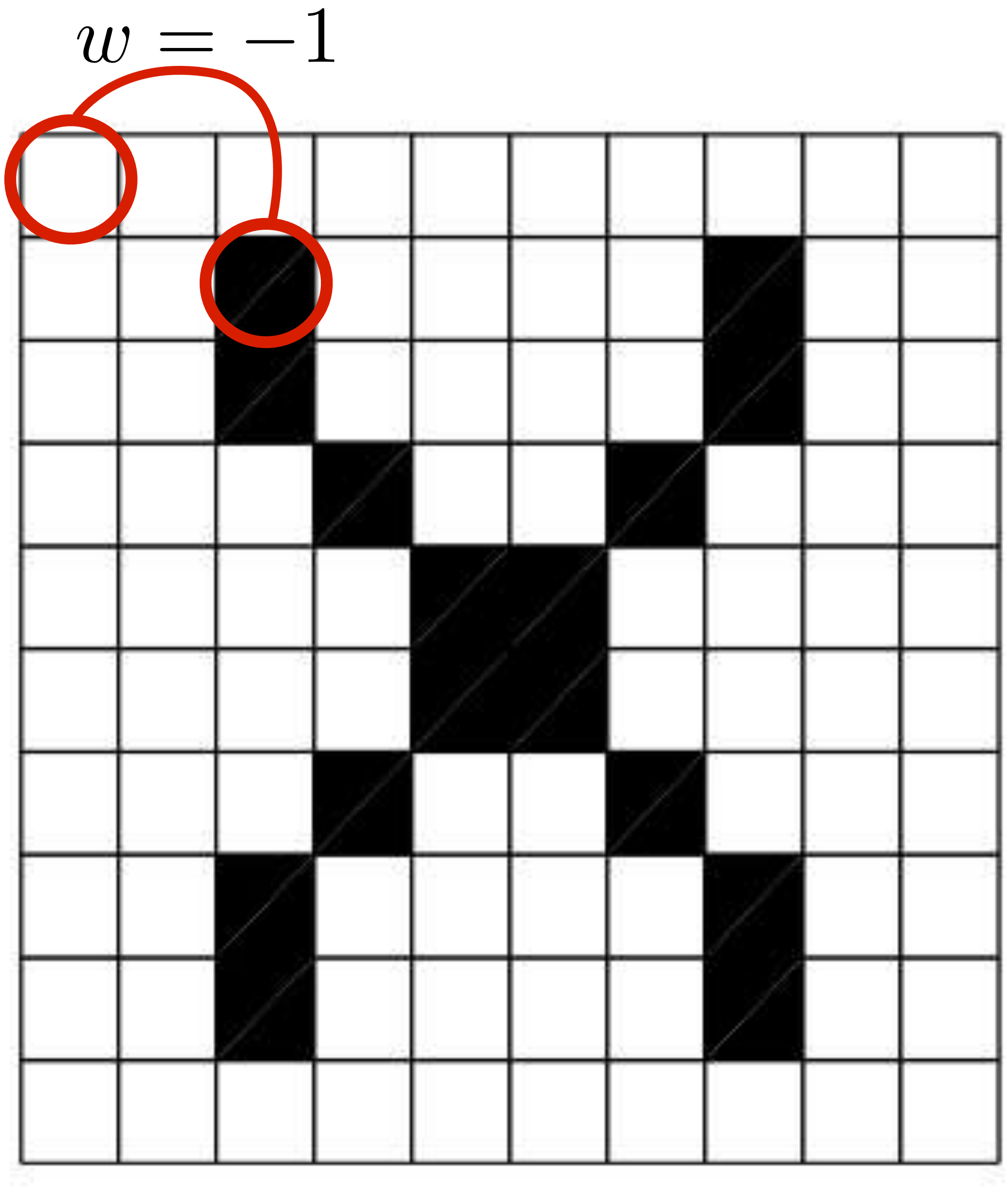


Synaptic weights:

$$w_{ij} = \frac{1}{N}\xi_i^{(1)}\xi_j^{(1)}$$

Exemple of learning in a recurrent network

Desired ouput $\xi^{(1)}$:



Synaptic weights:

$$w_{ij} = \frac{1}{N}\xi_i^{(1)}\xi_j^{(1)}$$

$\xi^{(1)}$ is a fixed point of the dynamics

Fixed point equation:

$$y_i = \mathrm{sgn}[\sum_{j=1}^{n} w_{ij} y_j(t)]$$

with:

$$w_{ij} = \frac{1}{N} \xi_i^{(1)} \xi_j^{(1)}$$

$\xi^{(1)}$ is a fixed point of the dynamics

Fixed point equation:

$$y_i = \mathrm{sgn}[\sum_{j=1}^{n} w_{ij} y_j(t)] \xrightarrow{y_i = \xi_i} = \mathrm{sgn}(\sum_{j=1}^{N} w_{ij} \xi_j)$$

with:

$$w_{ij} = \frac{1}{N} \xi_i^{(1)} \xi_j^{(1)}$$

$\xi^{(1)}$ is a fixed point of the dynamics

Fixed point equation:

$$y_i = \mathrm{sgn}[\sum_{j=1}^{n} w_{ij} y_j(t)] \xrightarrow{y_i = \xi_i} = \mathrm{sgn}(\sum_{j=1}^{N} w_{ij}\xi_j)$$

$$= \mathrm{sgn}(\sum_{j=1}^{N} \frac{\xi_i \xi_j}{N} \xi_j)$$

with:

$$w_{ij} = \frac{1}{N} \xi_i^{(1)} \xi_j^{(1)}$$

$\xi^{(1)}$ is a fixed point of the dynamics

Fixed point equation:

$$y_i = \mathrm{sgn}[\sum_{j=1}^{n} w_{ij} y_j(t)] \xrightarrow{y_i = \xi_i} = \mathrm{sgn}(\sum_{j=1}^{N} w_{ij}\xi_j)$$

$$= \mathrm{sgn}(\sum_{j=1}^{N} \frac{\xi_i \xi_j}{N} \xi_j)$$

$$= \mathrm{sgn}(\frac{\xi_i}{N} \sum_{j=1}^{N} \xi_j \xi_j)$$

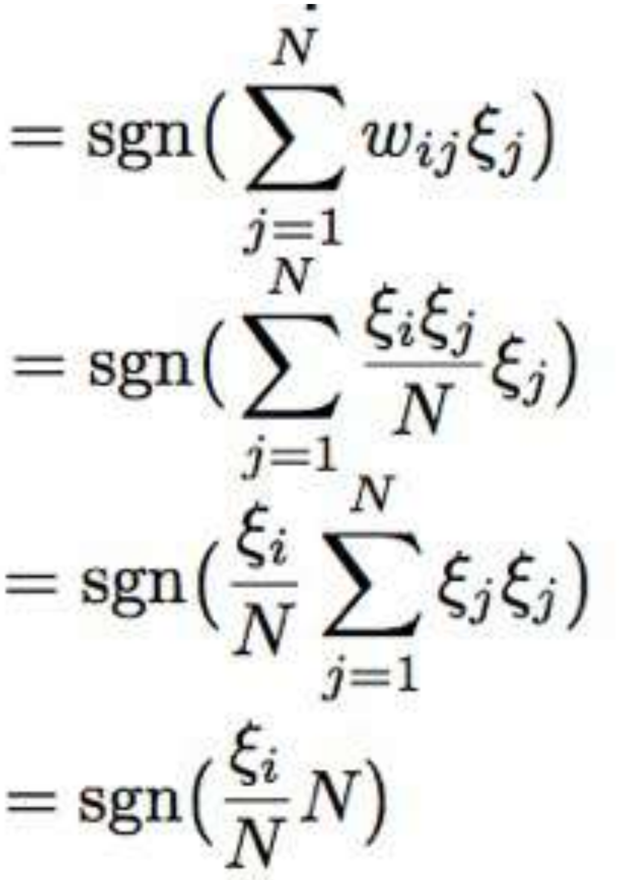
with:

$$w_{ij} = \frac{1}{N} \xi_i^{(1)} \xi_j^{(1)}$$

$\xi^{(1)}$ is a fixed point of the dynamics

Fixed point equation:

$$y_i = \mathrm{sgn}[\sum_{j=1}^{n} w_{ij} y_j(t)] \xrightarrow{y_i = \xi_i}$$



$$= \mathrm{sgn}(\sum_{j=1}^{N} w_{ij}\xi_j)$$

$$= \mathrm{sgn}(\sum_{j=1}^{N} \frac{\xi_i \xi_j}{N} \xi_j)$$

$$= \mathrm{sgn}(\frac{\xi_i}{N} \sum_{j=1}^{N} \xi_j \xi_j)$$

$$= \mathrm{sgn}(\frac{\xi_i}{N} N)$$

with:

$$w_{ij} = \frac{1}{N}\xi_i^{(1)}\xi_j^{(1)}$$

$\xi^{(1)}$ is a fixed point of the dynamics

Fixed point equation:

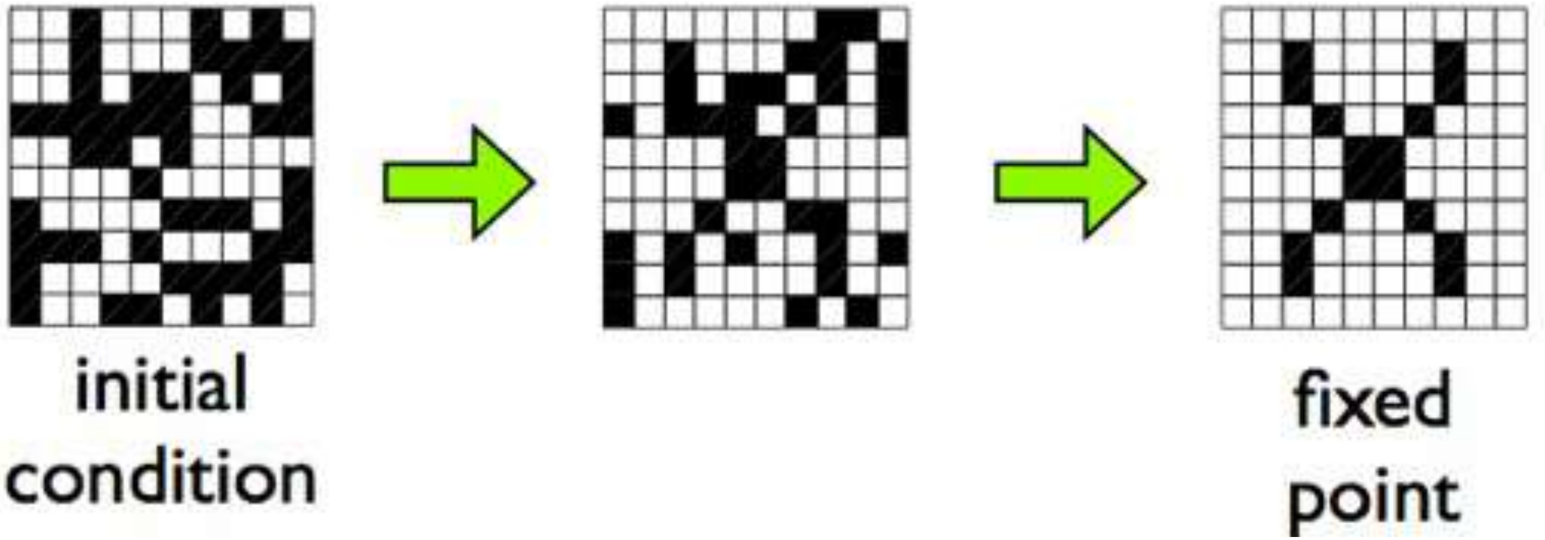
$$y_i = \mathrm{sgn}[\sum_{j=1}^{n} w_{ij} y_j(t)] \xrightarrow{y_i = \xi_i}$$

$$\begin{aligned}
&= \mathrm{sgn}(\sum_{j=1}^{N} w_{ij}\xi_j) \\
&= \mathrm{sgn}(\sum_{j=1}^{N} \frac{\xi_i \xi_j}{N} \xi_j) \\
&= \mathrm{sgn}(\frac{\xi_i}{N} \sum_{j=1}^{N} \xi_j \xi_j) \\
&= \mathrm{sgn}(\frac{\xi_i}{N} N) \\
&= \mathrm{sgn}(\xi_i) \\
&= \xi_i
\end{aligned}$$

with:

$$w_{ij} = \frac{1}{N} \xi_i^{(1)} \xi_j^{(1)}$$

Exemple of learning in a recurrent network

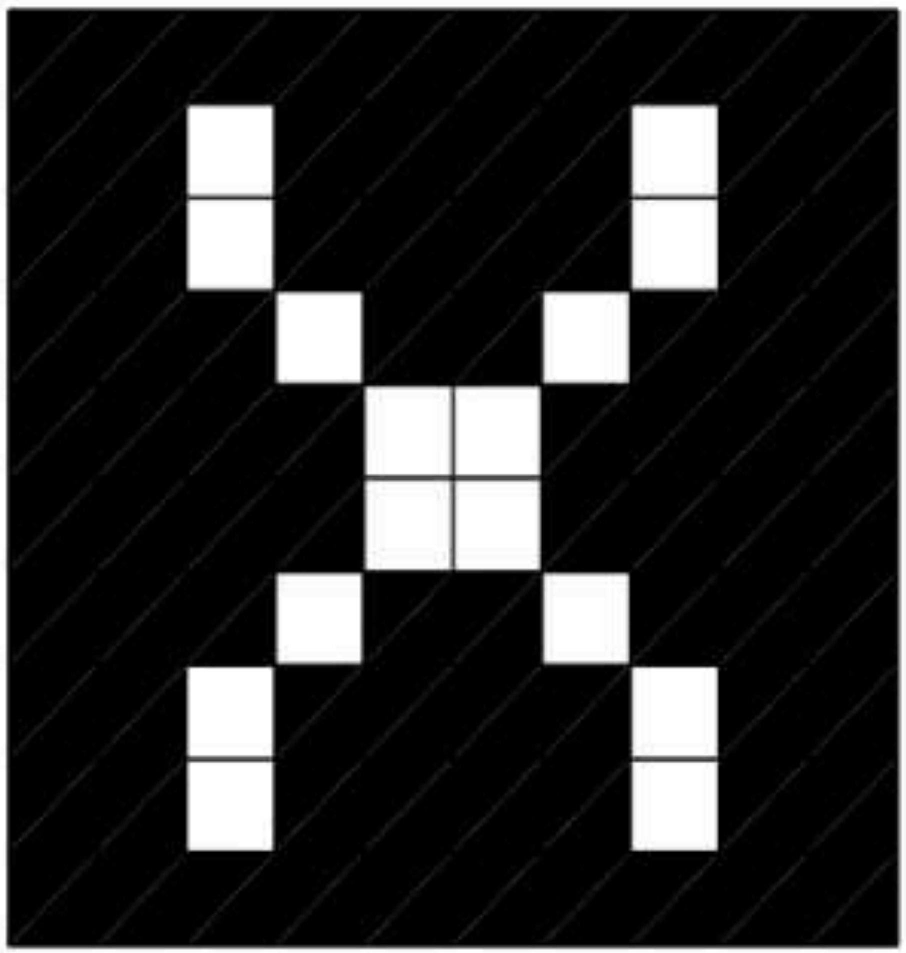


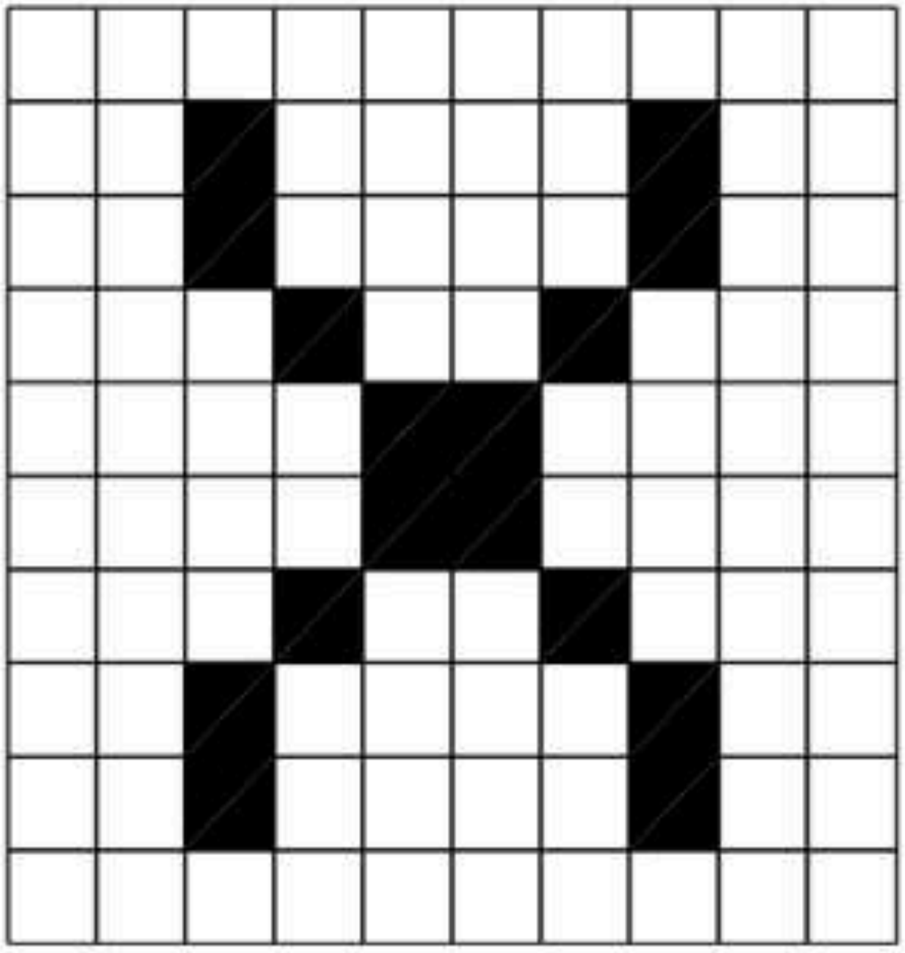
→ **Pattern is stored in the network and recovered from suitable initial conditions**

Content-addressable, associative memory

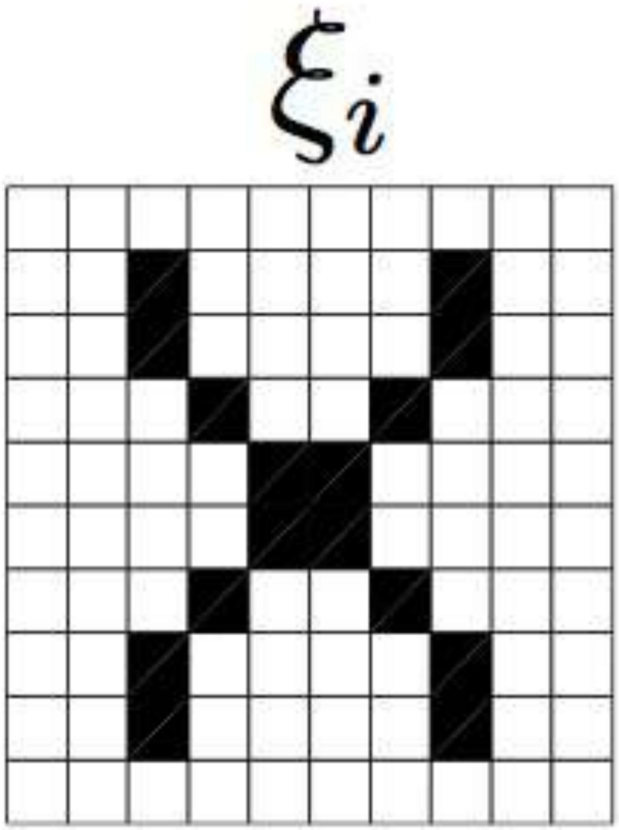
Content-addressable, associative memory

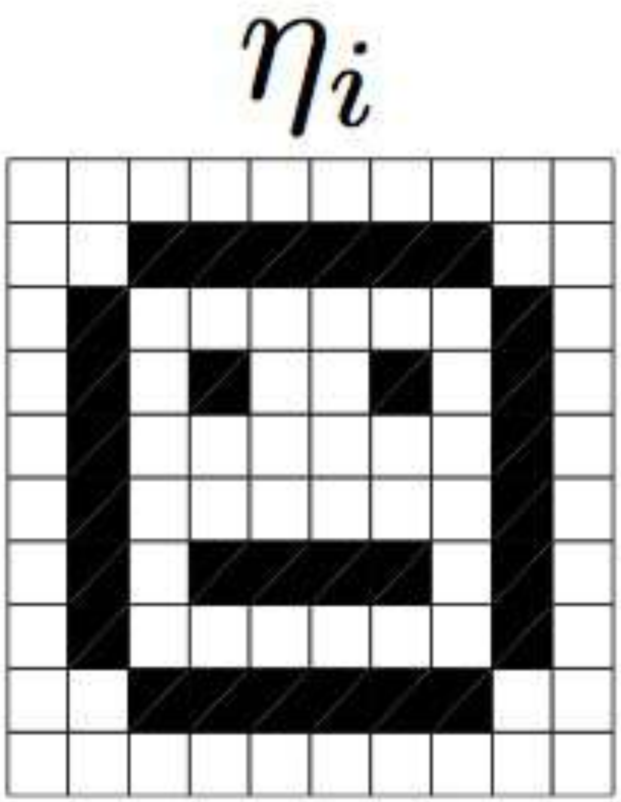
Inverse pattern is stored too!





Storing two patterns





$$w_{ij} = \frac{\xi_i^{(1)}\xi_j^{(1)} + \eta_i^{(1)}\eta_j^{(1)}}{N}$$

Storing multiple patterns

An schematic of a network with 4 attractors:

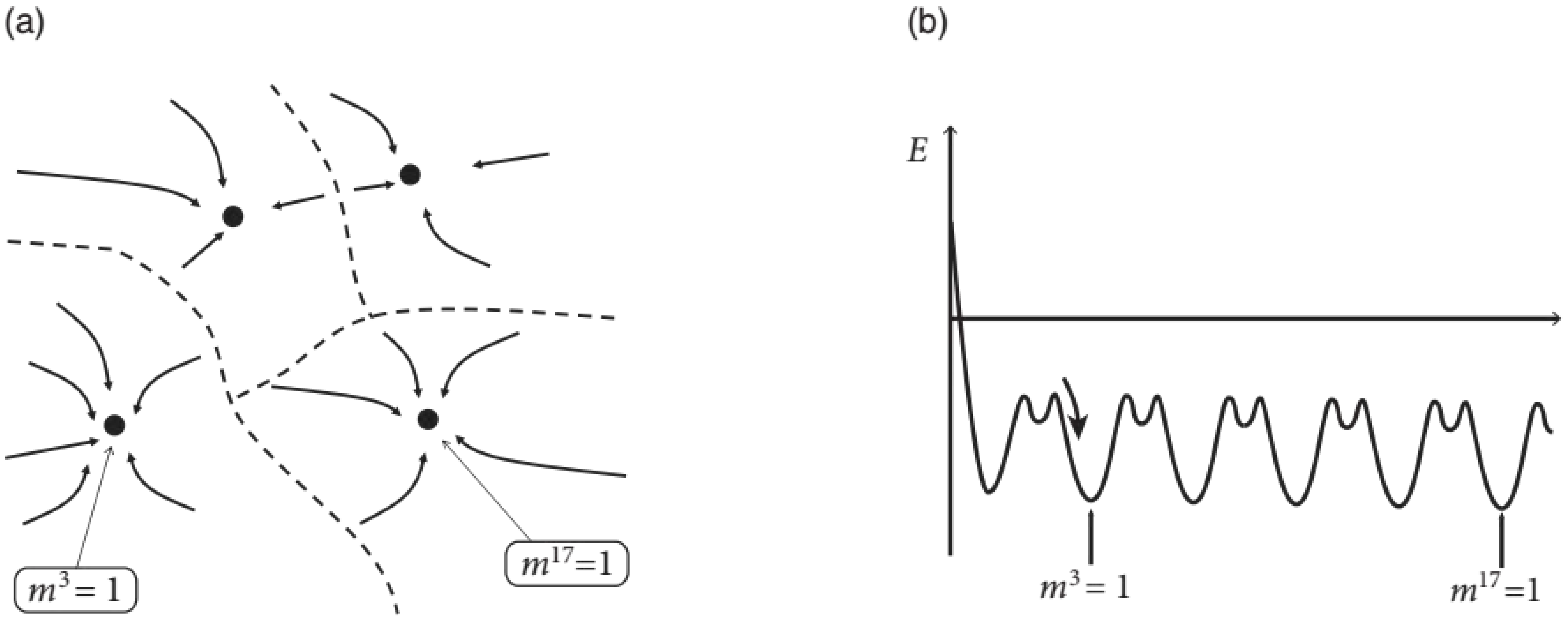


Fig. 17.9 Attractor picture and energy landscape. (a) The dynamics are attracted toward fixed points corresponding to memory states (overlap $m^\nu = 1$). Four attractor states are indicated. The dashed lines show the boundaries of the basin of attraction of each memory. (b) The Hopfield model has multiple equivalent energy minima, each one corresponding to the retrieval (overlap $m^\nu = 1$) of one pattern. Between the main minima, additional local minima (corresponding to mixtures of several patterns) may also exist.

Figure from Gerstner et al. (2014)

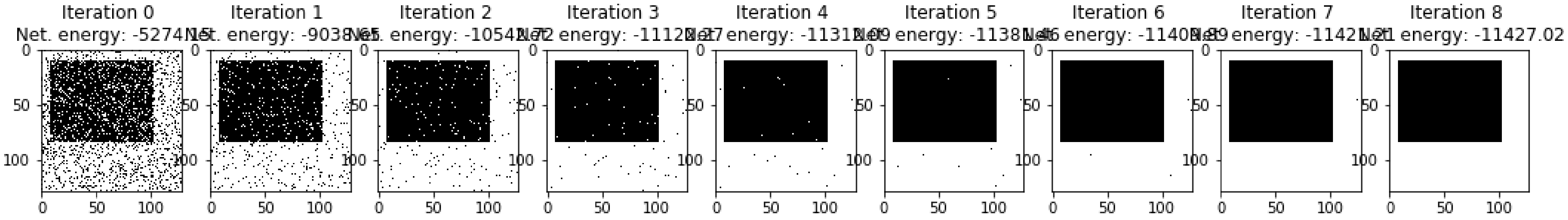
Another source for Hopfield networks

A nice tutorial related to Hopfield networks:

https://towardsdatascience.com/hopfield-networks-are-useless-heres-why-you-should-learn-them-f0930ebeadcd

(Last accessed on 25/04/2022)

There is a very nice example of de-noising using a Hopfield network with three attractors:



Binary neurons and networks: summary

- Binary neurons act as binary linear classifiers
- Learning rules can be used to train the neurons to produce the desired output
- Single layer feedforward networks can compute linearly separable operations [PERCEPTRON – HEBBIAN learning rule]

- Multilayer feedforward networks can compute any binary function

- Recurrent networks can memorize and recall patterns [ATTRACTOR networks]

Outlook

BINARY NEURONS AND NETWORKS

ARTIFICIAL NEURAL NETWORKS

aim: solve machine-learning problems

→ **inspired by biology**

NEUROSCIENCE

aim: understand how the brain works

→ **constrained by biology**

References

Georgopoulos AP, Georgopoulos A. The Beautiful Brain and the Influence of Santiago Ramón y Cajal on Medicine. JAMA. 2017;318(6):502–504. https://doi.org/10.1001/jama.2017.9858

Gerstner, W., Kistler, W., Naud, R., Paninski, L. (2014). Neuronal Dynamics. Oxford Oxford University Press.

Hebb, D. (1949). The organization of behavior; a neuropsychological theory. Wiley.

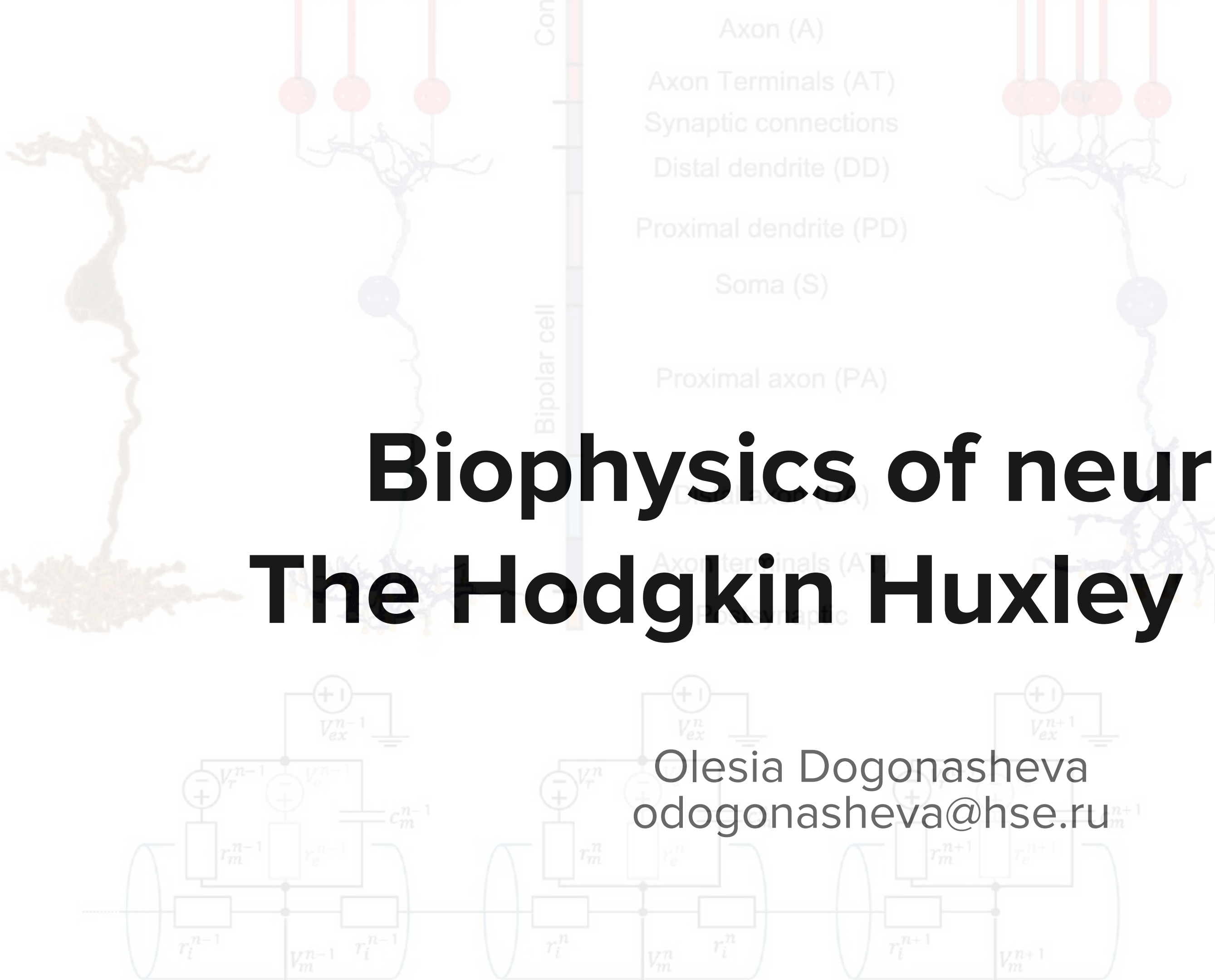
Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*. 79(8), 2554-2558. https://doi.org/10.1073/pnas.79.8.2554

Mazzoni, P., Andersen, R. Jordan, M. (1991) A more biologically plausible learning rule for neural networks. *Proceedings of the National Academy of Sciences.* 88 (10) 4433-4437 https://doi.org/10.1073/pnas.88.10.4433

McCulloch, W.S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133 (1943). https://doi.org/10.1007/BF02478259

Minsky, M., Papert S. (1969). Perceptrons: An introduction to computational geometry. M.I.T. Press.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. https://doi.org/10.1037/h0042519



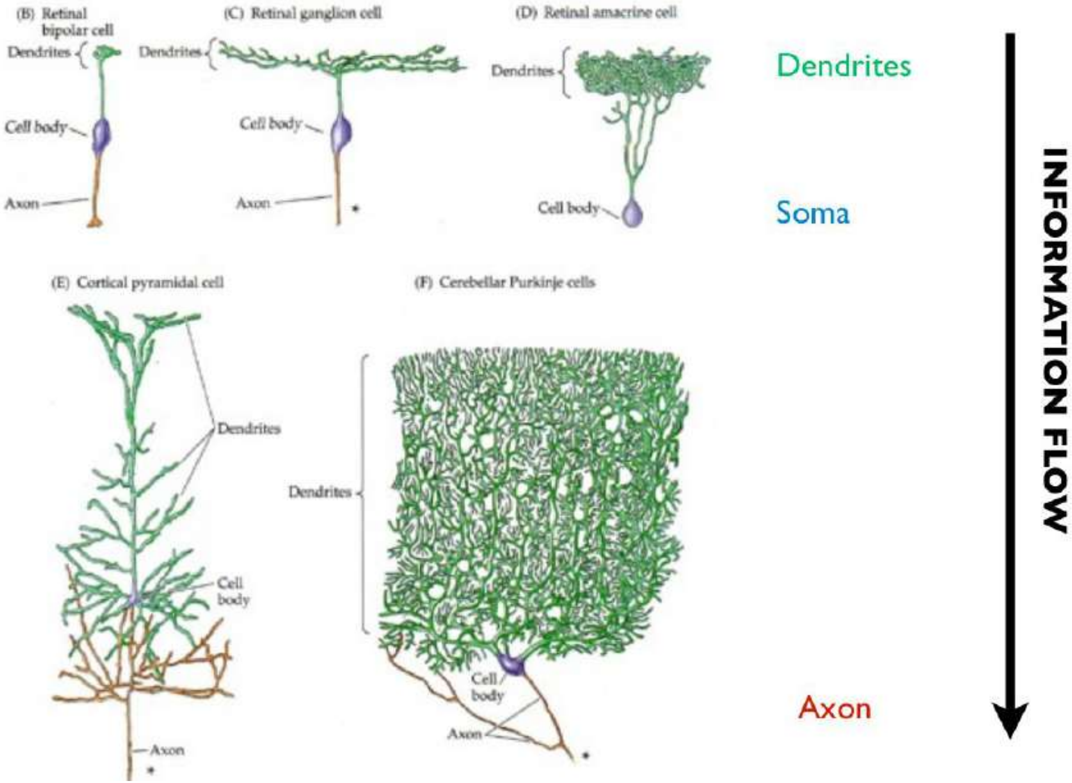
Biophysics of neurons
The Hodgkin Huxley model

Olesia Dogonasheva
odogonasheva@hse.ru

Outline

1. Neural electricity
    a. The resting potential
    b. The action potential
    c. Electrodiffusion and the Nerst potential
    d. The membrane equation
2. The Hodgkin Huxley model
3. Synapses

Neurons = basic units of computation



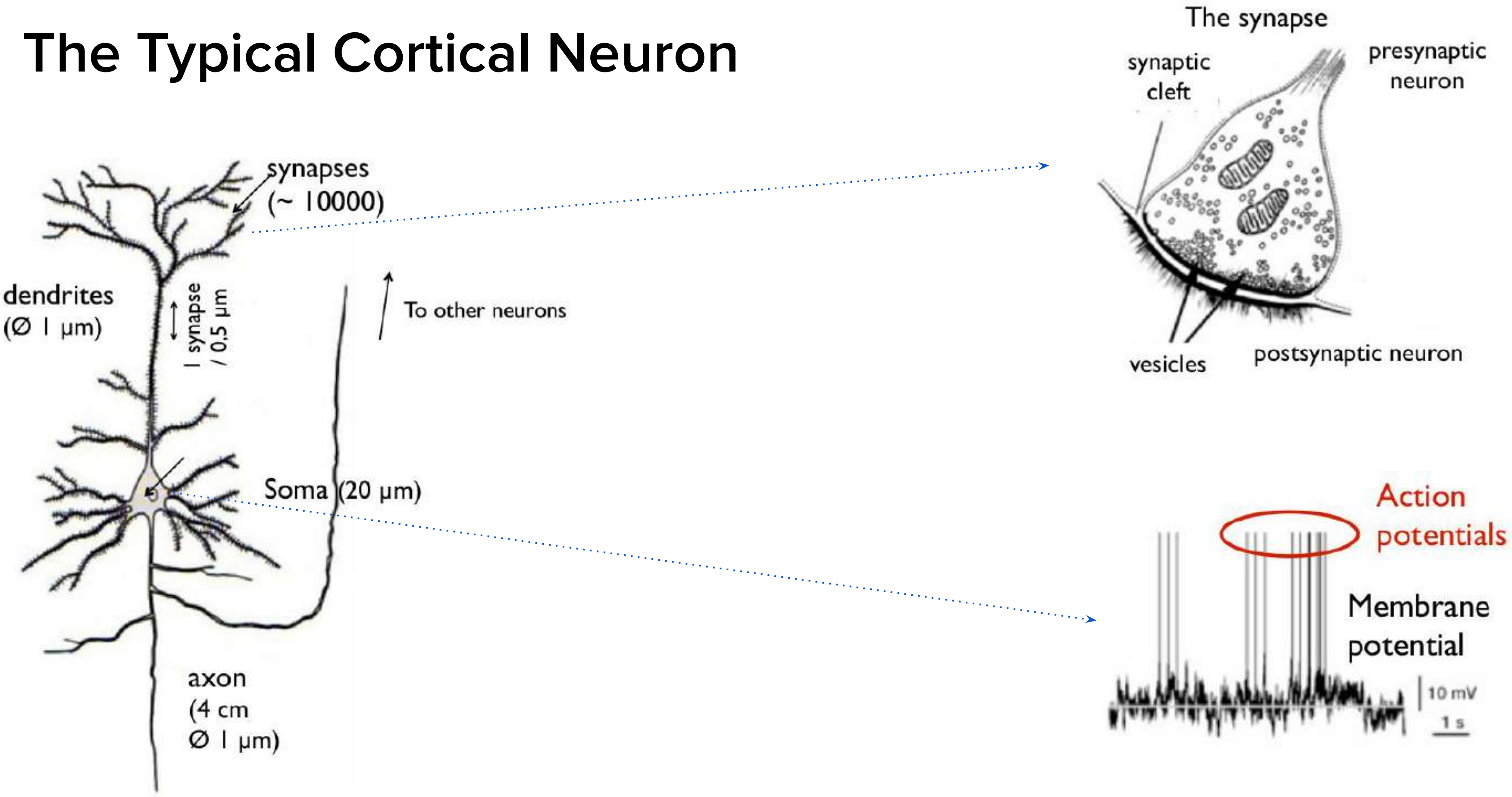
Dendrites

Soma

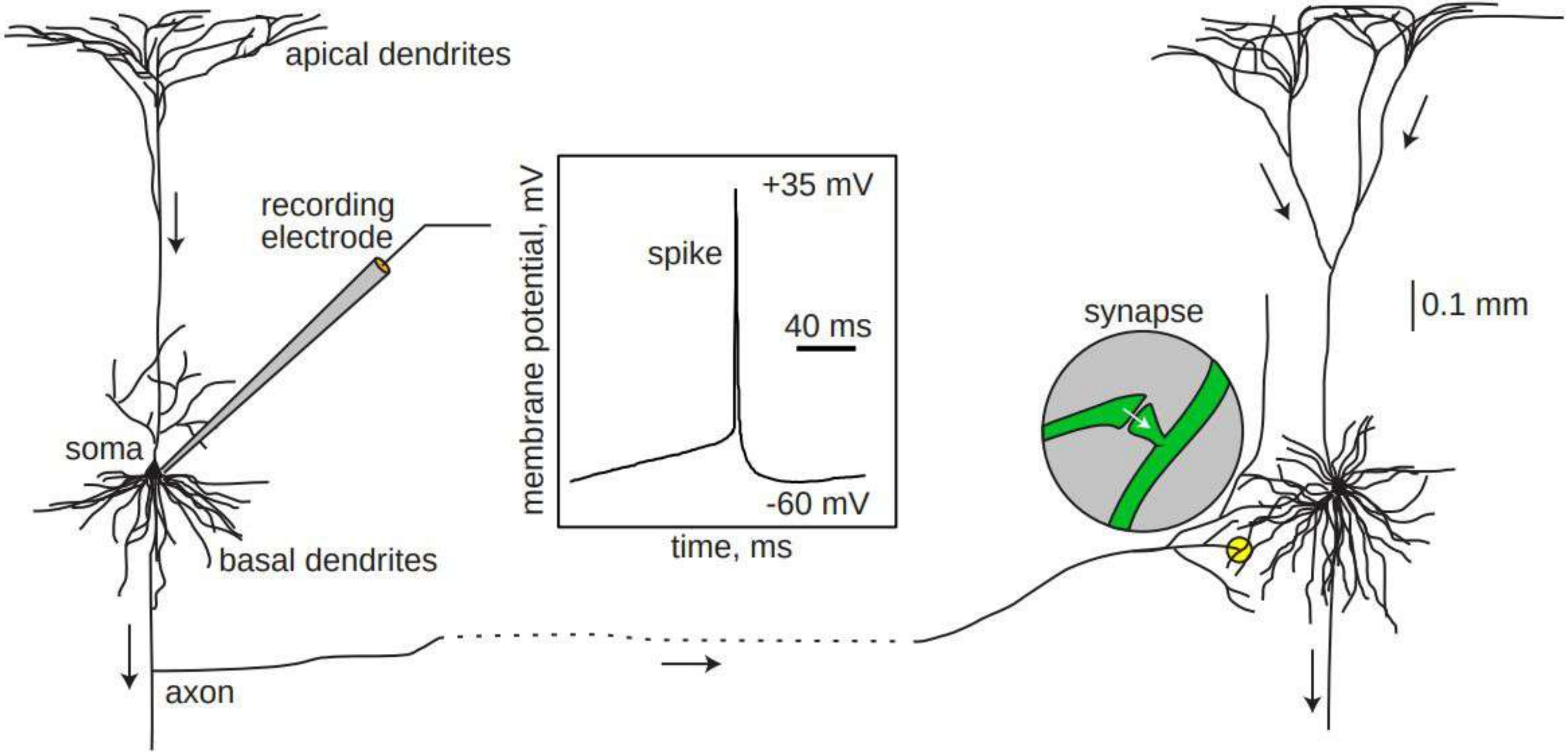
Axon

INFORMATION FLOW

The Typical Cortical Neuron



The action potential



The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

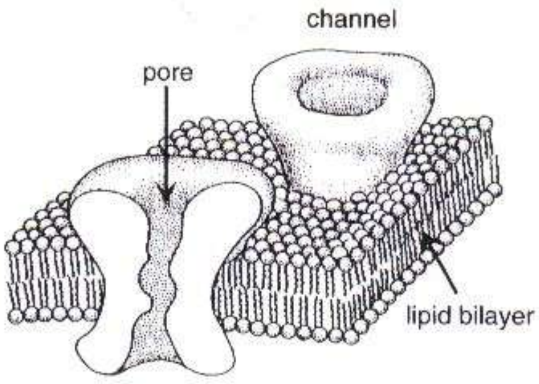
1. K^+-channels genes turns on:

The resting potential

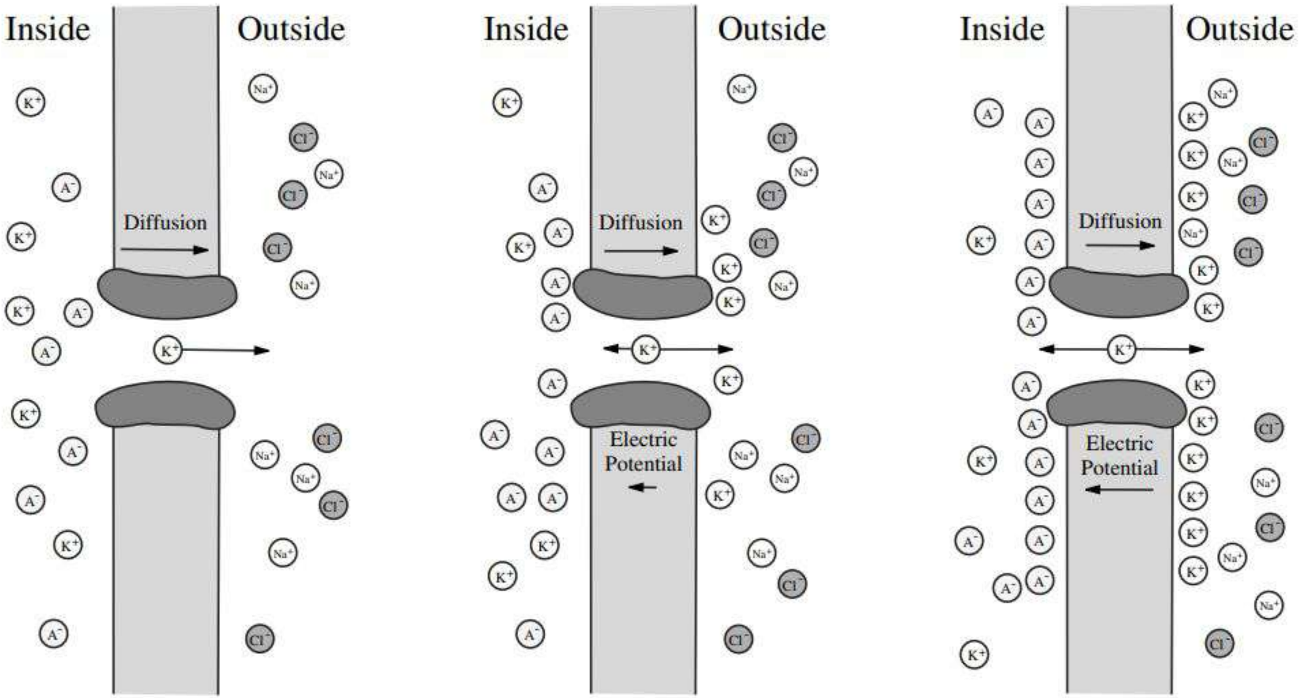
Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

1. K^+-channels genes turns on:



K+-channels genes and ion-current equilibrium



The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, -PO_4^-, some aminoacids

1. K^+-channels genes turns on:

The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

1. K^+-channels genes turns on:

$$\boxed{E_{\text{ion}} = \frac{RT}{zF} \ln \frac{[\text{Ion}]_{\text{out}}}{[\text{Ion}]_{\text{in}}}}$$

The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

1. K^+-channels genes turns on:

$$\boxed{E_{\text{ion}} = \frac{RT}{zF} \ln \frac{[\text{Ion}]_{\text{out}}}{[\text{Ion}]_{\text{in}}}}$$

z = ion valency
R = universal gas constant (8, 315 mJ/(K◦·Mol)
T = temperature (in degrees Kelvin)
F = Faraday’s constant (96,48 coulombs/Mol

The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

1. K^+-channels genes turns on:

$$\boxed{E_{\text{ion}} = \frac{RT}{zF} \ln \frac{[\text{Ion}]_{\text{out}}}{[\text{Ion}]_{\text{in}}}}$$

z = ion valency
R = universal gas constant (8, 315 mJ/(K◦·Mol)
T = temperature (in degrees Kelvin)
F = Faraday’s constant (96,48 coulombs/Mol

2. Na^+-channels genes turns on: sodium enters the cell

The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

1. K^+-channels genes turns on:

$$\boxed{E_{\text{ion}} = \frac{RT}{zF} \ln \frac{[\text{Ion}]_{\text{out}}}{[\text{Ion}]_{\text{in}}}}$$

z = ion valency
R = universal gas constant (8, 315 mJ/(K∘·Mol)
T = temperature (in degrees Kelvin)
F = Faraday's constant (96,48 coulombs/Mol

2. Na^+-channels genes turns on: sodium enters the cell

$\#_{\text{channels}}$ varies in cells,
if a lot ⇒ $|V_{rest}|$ ↓ ⇒ the neuron is more excitable
(more sensitive)

The resting potential

Initial state of neurons (≈ up to 3 months of the embryonic development): $V_{rest} = 0$

- Positively charged ions: K^+ и Na^+
- Negatively charged ions: Cl^-, $-PO_4^-$, some aminoacids

1. K^+-channels genes turns on:

$$\boxed{E_{\text{ion}} = \frac{RT}{zF} \ln \frac{[\text{Ion}]_{\text{out}}}{[\text{Ion}]_{\text{in}}}}$$

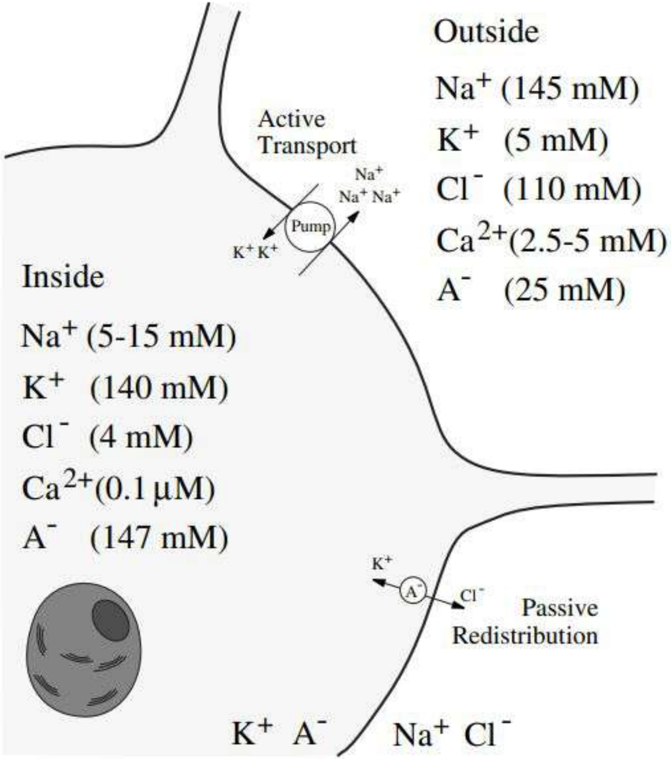
z = ion valency
R = universal gas constant (8, 315 mJ/(K◦·Mol)
T = temperature (in degrees Kelvin)
F = Faraday’s constant (96,48 coulombs/Mol

2. Na^+-channels genes turns on: sodium enters the cell

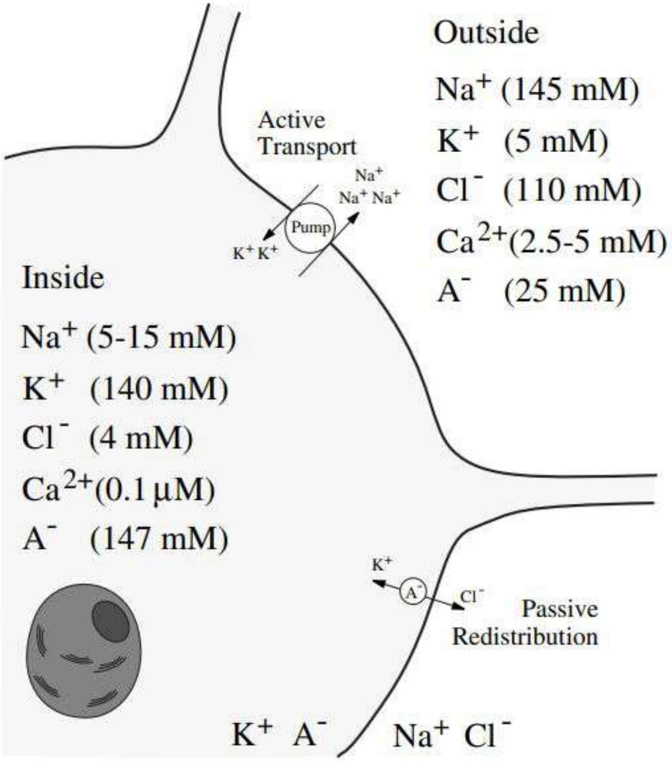
$\#_{\text{channels}}$ varies in cells,
if a lot ⇒ $|V_{rest}|$ ↓ ⇒ the neuron is more excitable (more sensitive)

3. Na^+-K^+-ATPase performs the process reverse to 1-2.

Main ionic currents through neuron membranes



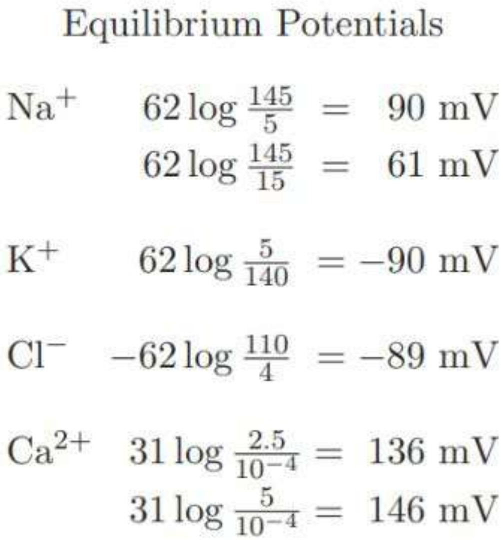
Main ionic currents through neuron membranes



Equilibrium Potentials

$$
\begin{array}{llcl}
\mathrm{Na}^{+} & 62\log\frac{145}{5} & = & 90\ \mathrm{mV} \\
 & 62\log\frac{145}{15} & = & 61\ \mathrm{mV} \\
\mathrm{K}^{+} & 62\log\frac{5}{140} & = & -90\ \mathrm{mV} \\
\mathrm{Cl}^{-} & -62\log\frac{110}{4} & = & -89\ \mathrm{mV} \\
\mathrm{Ca}^{2+} & 31\log\frac{2.5}{10^{-4}} & = & 136\ \mathrm{mV} \\
 & 31\log\frac{5}{10^{-4}} & = & 146\ \mathrm{mV}
\end{array}
$$

The action potential

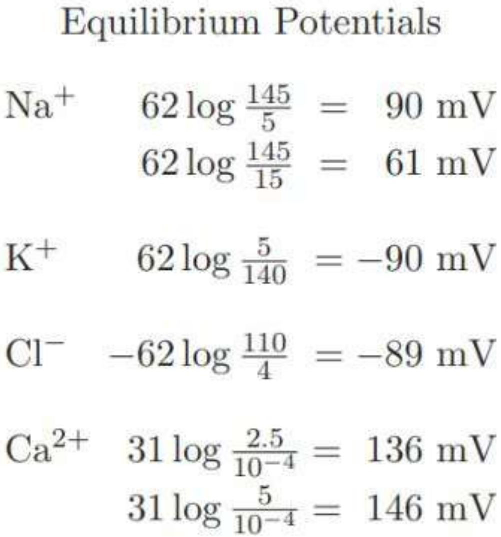


Equilibrium Potentials

Na^+	$62\log\frac{145}{5}$	$= 90$ mV
	$62\log\frac{145}{15}$	$= 61$ mV
K^+	$62\log\frac{5}{140}$	$= -90$ mV
Cl^-	$-62\log\frac{110}{4}$	$= -89$ mV
Ca^{2+}	$31\log\frac{2.5}{10^{-4}}$	$= 136$ mV
	$31\log\frac{5}{10^{-4}}$	$= 146$ mV

The action potential

At rest, the neuron is polarised: $V_m \approx -70$ mV



Equilibrium Potentials

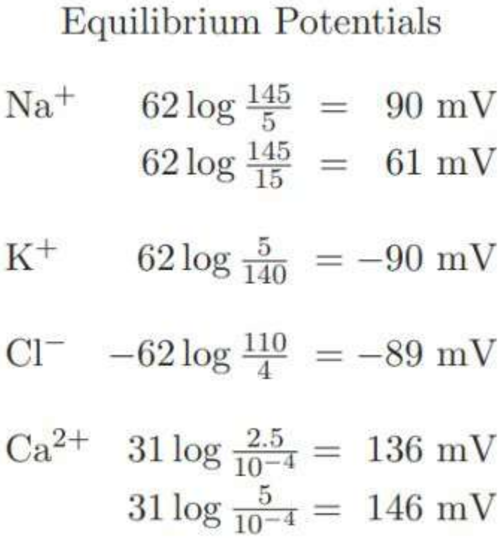
$$\begin{array}{lrcr}
\mathrm{Na}^+ & 62\log\frac{145}{5} & = & 90\ \mathrm{mV} \\
 & 62\log\frac{145}{15} & = & 61\ \mathrm{mV} \\
 & & & \\
\mathrm{K}^+ & 62\log\frac{5}{140} & = & -90\ \mathrm{mV} \\
 & & & \\
\mathrm{Cl}^- & -62\log\frac{110}{4} & = & -89\ \mathrm{mV} \\
 & & & \\
\mathrm{Ca}^{2+} & 31\log\frac{2.5}{10^{-4}} & = & 136\ \mathrm{mV} \\
 & 31\log\frac{5}{10^{-4}} & = & 146\ \mathrm{mV}
\end{array}$$

The action potential

At rest, the neuron is polarised: $V_m \eqsim -70$ mV

Terminology:

- depolarised: V_m increases
- hyperpolarised: V_m decreases



Equilibrium Potentials

$$
\begin{array}{lrcr}
\mathrm{Na}^+ & 62 \log \frac{145}{5} & = & 90\ \mathrm{mV} \\
 & 62 \log \frac{145}{15} & = & 61\ \mathrm{mV} \\
\\
\mathrm{K}^+ & 62 \log \frac{5}{140} & = & -90\ \mathrm{mV} \\
\\
\mathrm{Cl}^- & -62 \log \frac{110}{4} & = & -89\ \mathrm{mV} \\
\\
\mathrm{Ca}^{2+} & 31 \log \frac{2.5}{10^{-4}} & = & 136\ \mathrm{mV} \\
 & 31 \log \frac{5}{10^{-4}} & = & 146\ \mathrm{mV}
\end{array}
$$

The action potential

At rest, the neuron is polarised: $V_m \approx -70$ mV

Terminology:

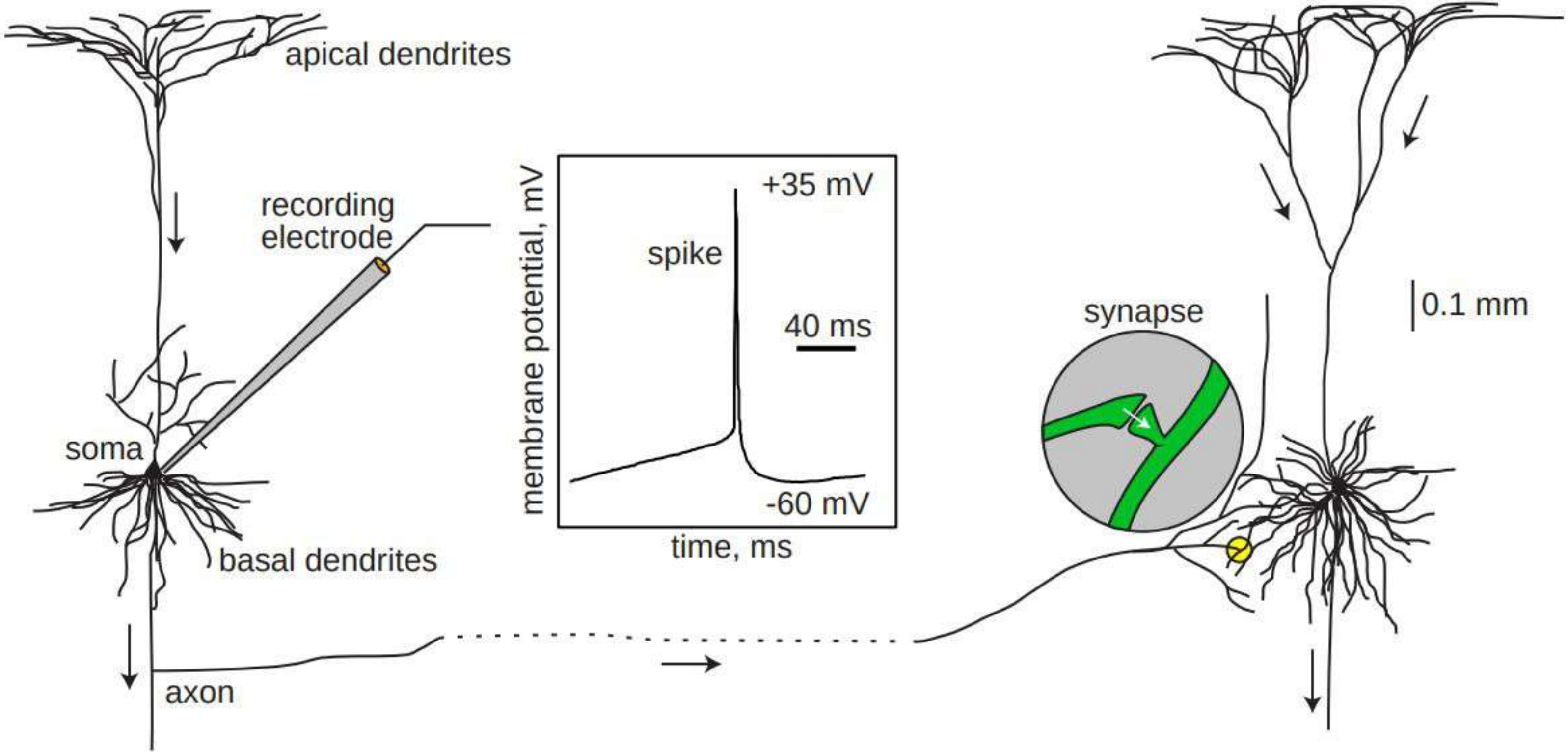
- depolarised: V_m increases
- hyperpolarised: V_m decreases

$$E_{\mathrm{K}} < E_{\mathrm{Cl}} < V_{(\mathrm{at\ rest})} < E_{\mathrm{Na}} < E_{\mathrm{Ca}}$$

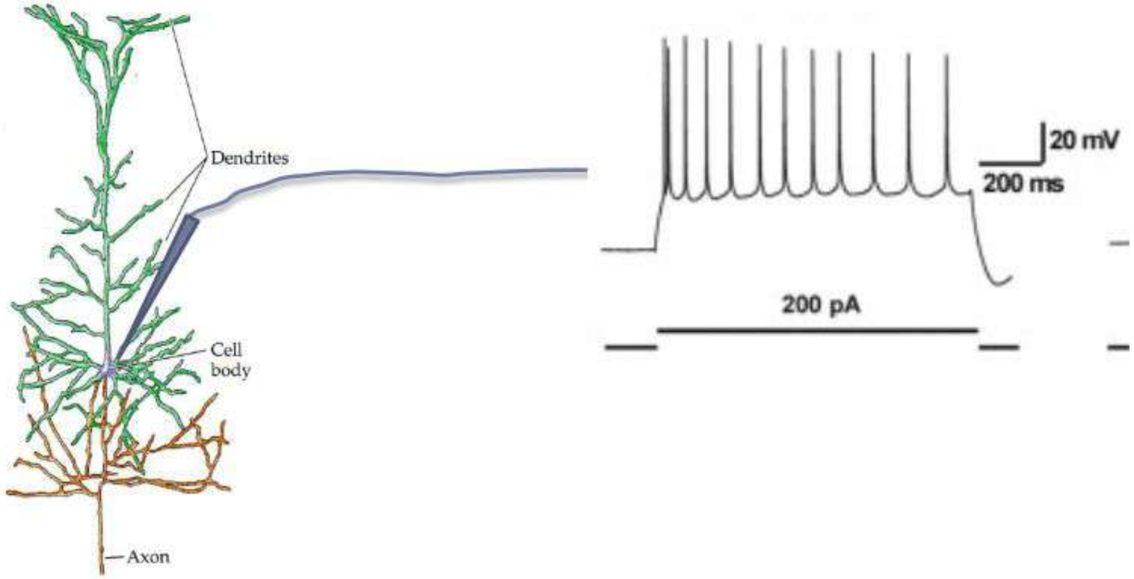
Equilibrium Potentials

Na^+	$62 \log \frac{145}{5}$	$= 90$ mV
	$62 \log \frac{145}{15}$	$= 61$ mV
K^+	$62 \log \frac{5}{140}$	$= -90$ mV
Cl^-	$-62 \log \frac{110}{4}$	$= -89$ mV
Ca^{2+}	$31 \log \frac{2.5}{10^{-4}}$	$= 136$ mV
	$31 \log \frac{5}{10^{-4}}$	$= 146$ mV

The action potential



Electrical activity in vitro



Membrane potential

Injected current

The action potential

The action potential is a reply on a stimultation of a neuron

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell

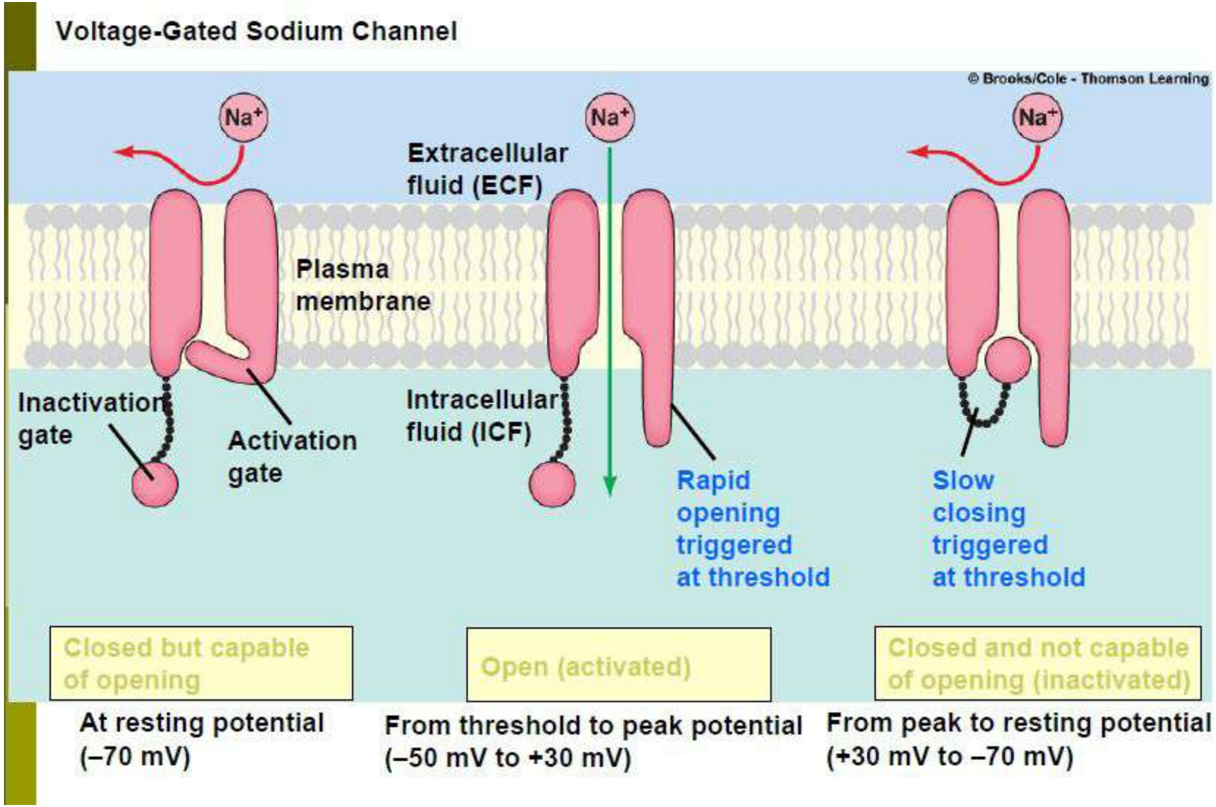
The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na$^+$-channels open
   b. Na$^+$-ions flood in the cell

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

Na$^+$-channels


Voltage-Gated Sodium Channel
© Brooks/Cole - Thomson Learning
Na+
Na+
Na+
Extracellular fluid (ECF)
Plasma membrane
Inactivation gate
Activation gate
Intracellular fluid (ICF)
Rapid opening triggered at threshold
Slow closing triggered at threshold
Closed but capable of opening
Open (activated)
Closed and not capable of opening (inactivated)
At resting potential (–70 mV)
From threshold to peak potential (–50 mV to +30 mV)
From peak to resting potential (+30 mV to –70 mV)

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell
3. Repolarization phase:

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na$^+$-channels open
   b. Na$^+$-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close
   b. Voltage-gated K^+-channels open

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close
   b. Voltage-gated K^+-channels open
   c. K^+ -ions flood out of the cell

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close
   b. Voltage-gated K^+-channels open
   c. K^+ -ions flood out of the cell

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

Na^+-channels close after 0.5 ms
K^+-channels are slow
K^+-channels open after 0.5 ms

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
    a. Voltage-gated Na^+-channels open
    b. Na^+-ions flood in the cell
3. Repolarization phase:
    a. Voltage-gated Na+-channels close
    b. Voltage-gated K^+-channels open
    c. K^+ -ions flood out of the cell
4. Hyperpolarization phase

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

Na^+-channels close after 0.5 ms
K^+-channels are slow
K^+-channels open after 0.5 ms

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close
   b. Voltage-gated K^+-channels open
   c. K^+ -ions flood out of the cell
4. Hyperpolarization phase
   a. K^+ -ions still flood out of the cell

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

Na^+-channels close after 0.5 ms
K^+-channels are slow
K^+-channels open after 0.5 ms

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close
   b. Voltage-gated K^+-channels open
   c. K^+ -ions flood out of the cell
4. Hyperpolarization phase
   a. K^+ -ions still flood out of the cell
   b. Voltage-gated K+-channels close

Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

Na^+-channels close after 0.5 ms
K^+-channels are slow
K^+-channels open after 0.5 ms

The action potential

The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na^+-channels open
   b. Na^+-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close
   b. Voltage-gated K^+-channels open
   c. K^+ -ions flood out of the cell
4. Hyperpolarization phase
   a. K^+ -ions still flood out of the cell
   b. Voltage-gated K+-channels close

Re-established

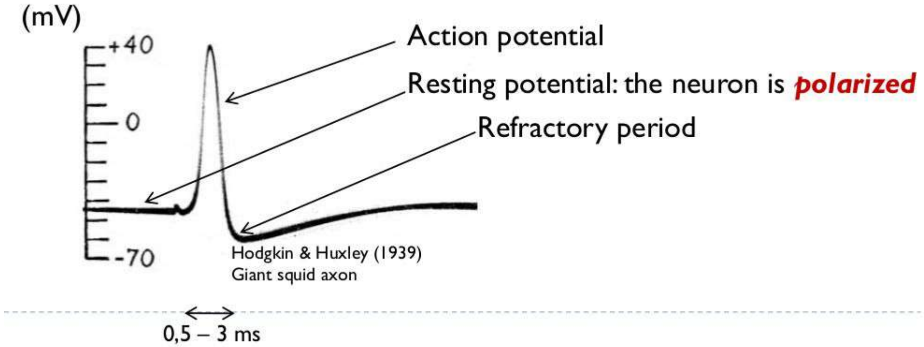
Na+-channels are ionotropic and fast
Na+-channels open when V > -50 mV

Na^+-channels close after 0.5 ms
K^+-channels are slow
K^+-channels open after 0.5 ms

The action potential

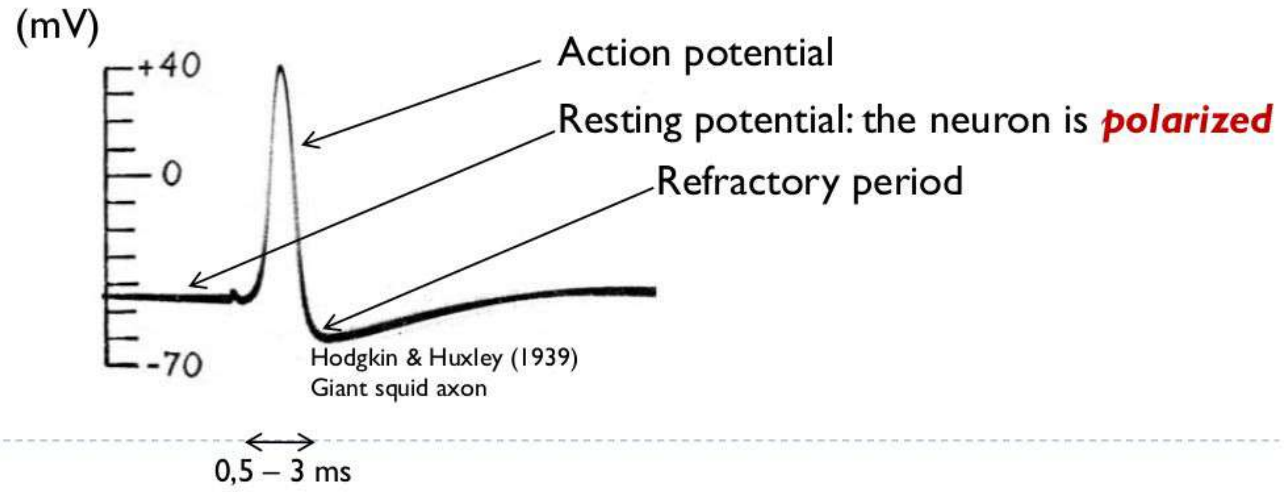
The action potential is a reply on a stimultation of a neuron

1. Stimulation
2. Depolarization phase:
   a. Voltage-gated Na$^+$-channels open
   b. Na$^+$-ions flood in the cell
3. Repolarization phase:
   a. Voltage-gated Na+-channels close
   b. Voltage-gated K$^+$-channels open
   c. K$^+$ -ions flood out of the cell
4. Hyperpolarization phase
   a. K$^+$ -ions still flood out of the cell
   b. Voltage-gated K+-channels close



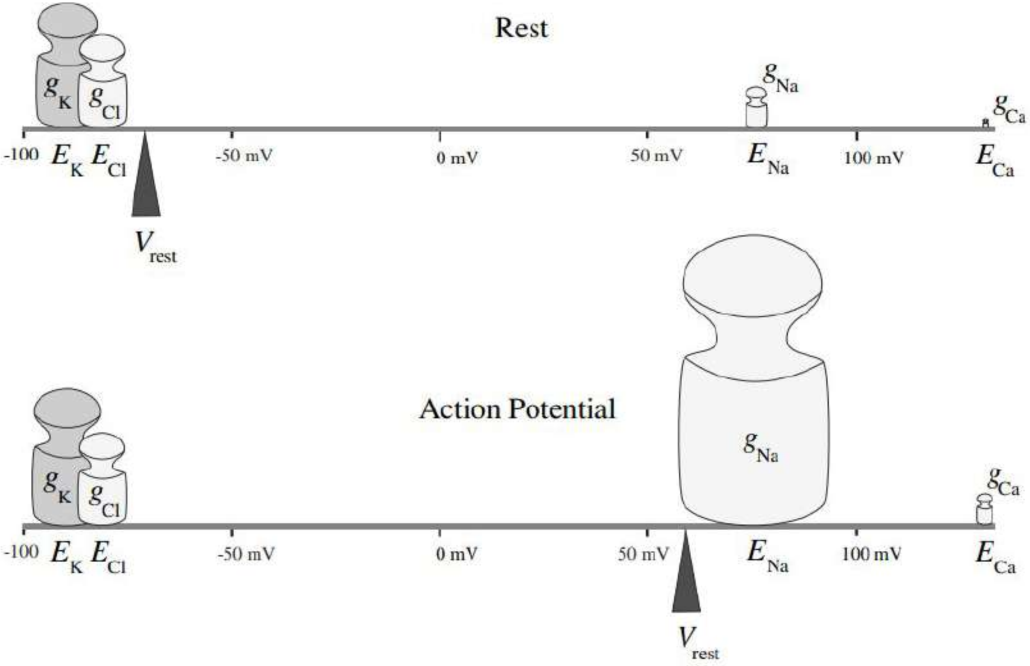
Re-established

The action potential



Hodgkin & Huxley (1939)
Giant squid axon

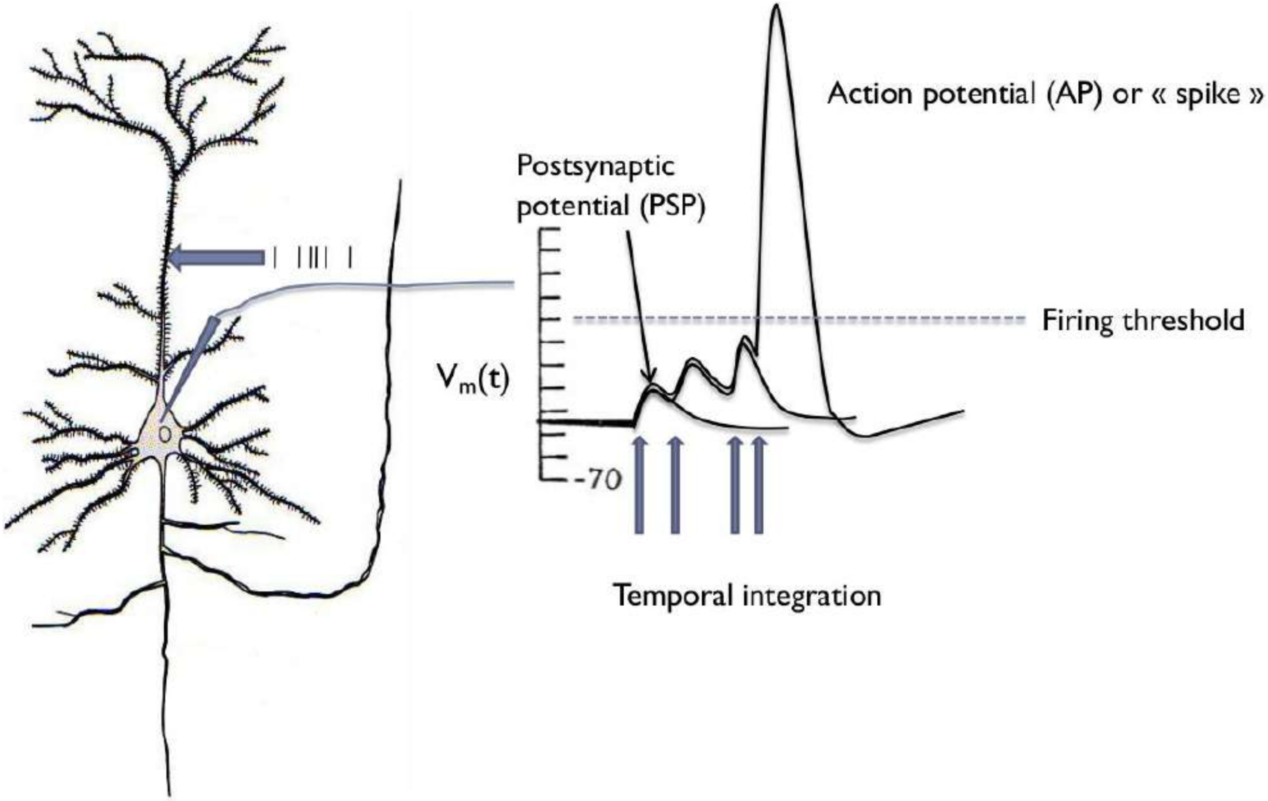
The action potential



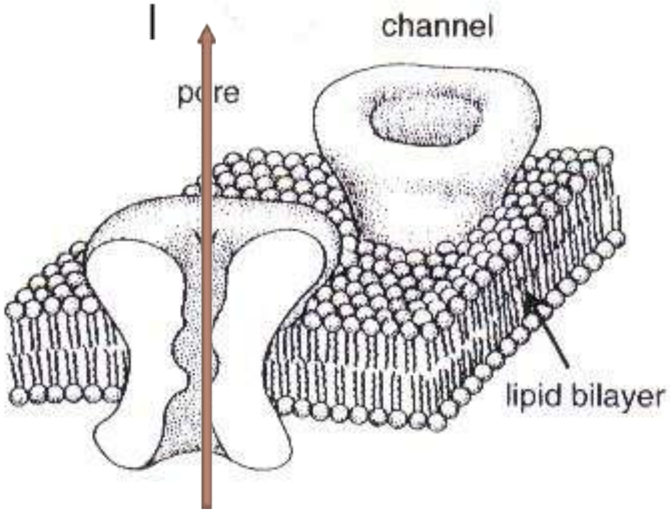
Equilibrium Potentials

Na^+	$62\log\frac{145}{5}$	$= 90$ mV
	$62\log\frac{145}{15}$	$= 61$ mV
K^+	$62\log\frac{5}{140}$	$= -90$ mV
Cl^-	$-62\log\frac{110}{4}$	$= -89$ mV
Ca^{2+}	$31\log\frac{2.5}{10^{-4}}$	$= 136$ mV
	$31\log\frac{5}{10^{-4}}$	$= 146$ mV

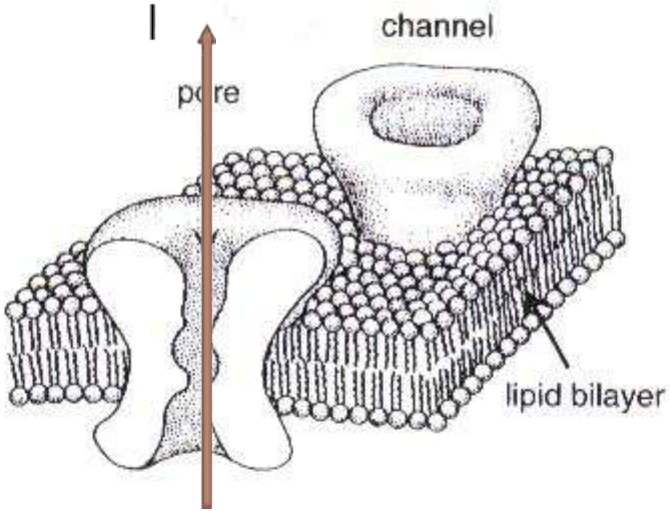
How does a neuron process spike trains?

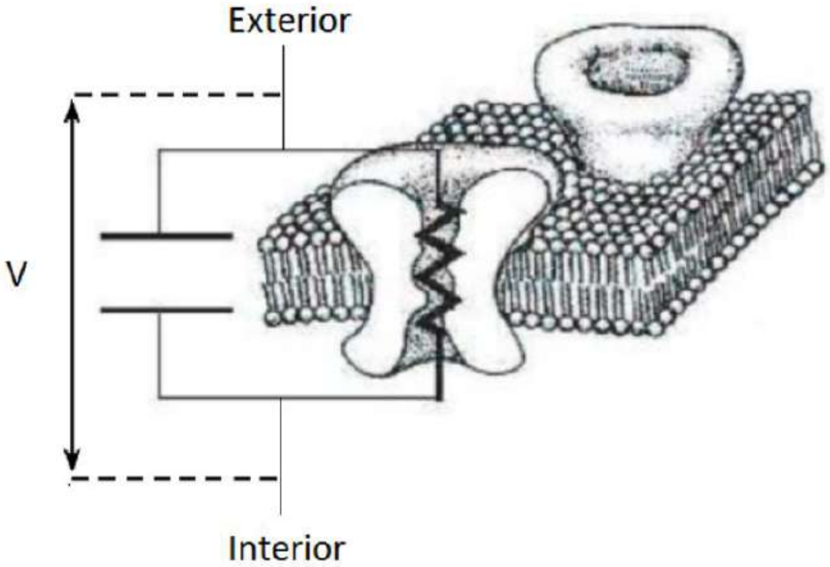

Action potential (AP) or « spike »
Postsynaptic potential (PSP)
Firing threshold
$V_m(t)$
-70
Temporal integration

Equivalent electrical circuits



Equivalent electrical circuits





The membrane equation

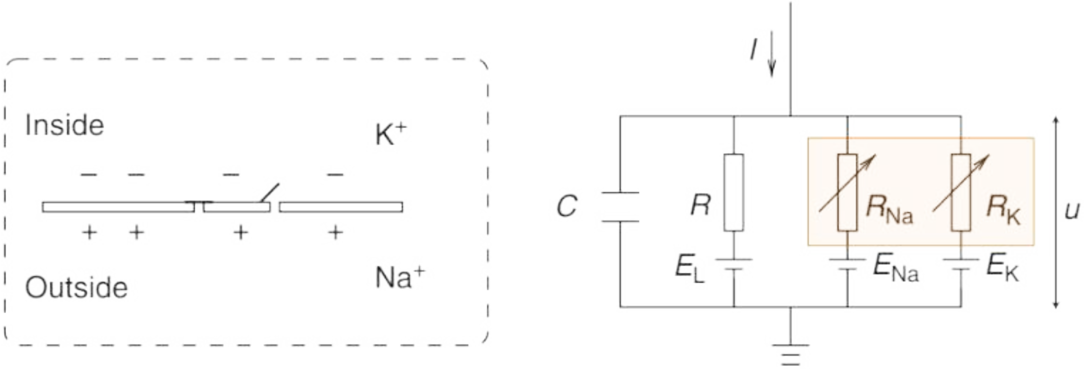


Fig. 2.2 Schematic diagram for the Hodgkin–Huxley model.

The membrane equation

$I = I_C + I_L + I_{Na} + I_K$

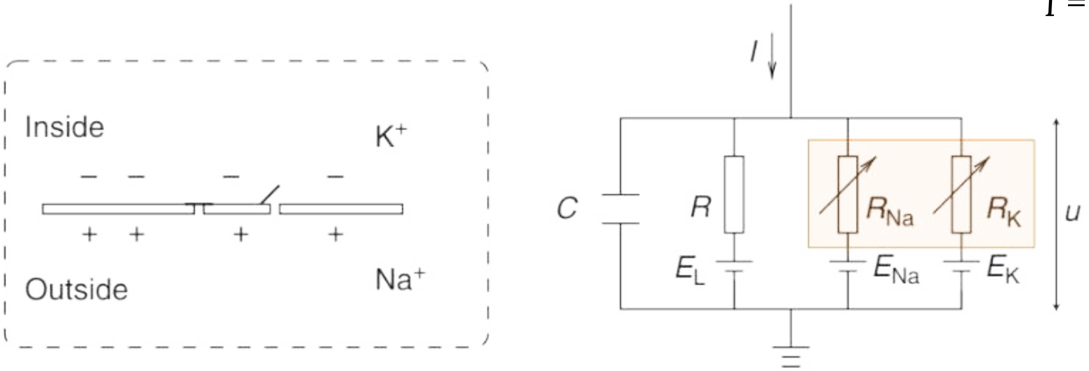
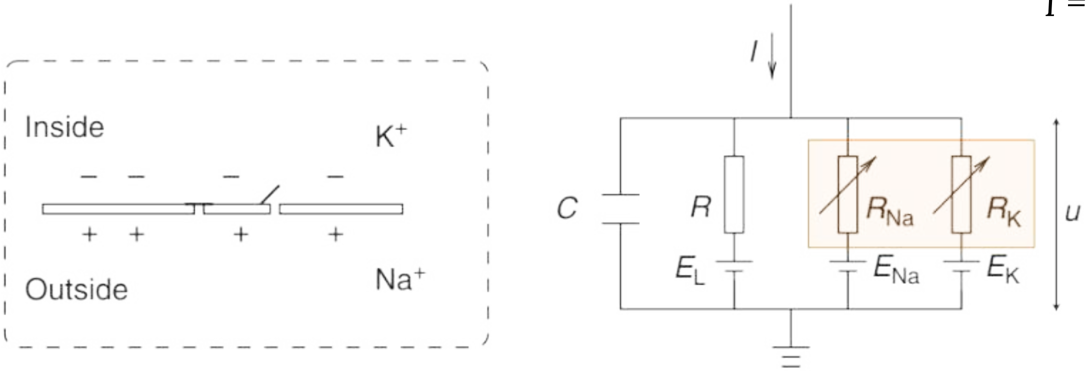


Fig. 2.2 Schematic diagram for the Hodgkin–Huxley model.

The membrane equation

$$I = I_C + I_L + I_{Na} + I_K$$

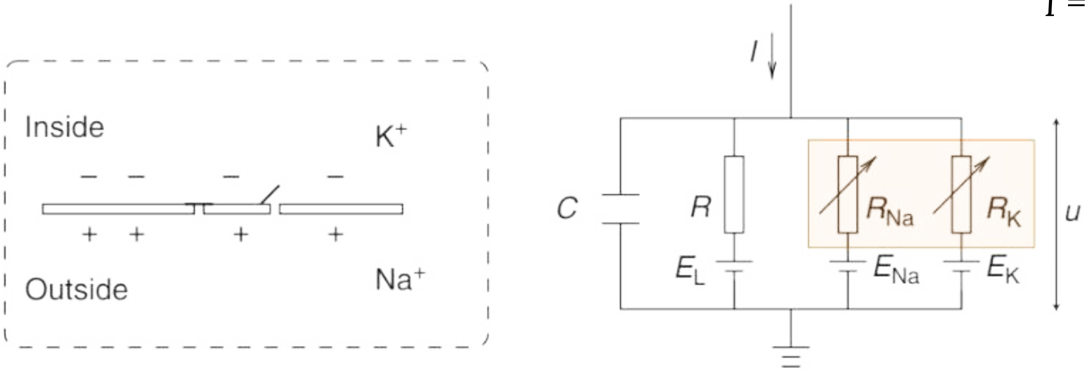


Voltage-gated resistances

Fig. 2.2 Schematic diagram for the Hodgkin–Huxley model.

The membrane equation

$$I = I_C + I_L + I_{Na} + I_K$$



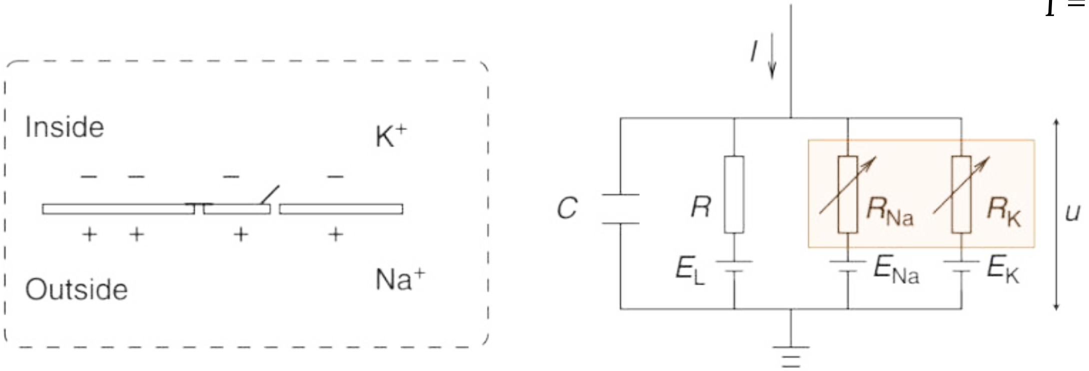
Voltage-gated resistances

Fig. 2.2 Schematic diagram for the Hodgkin–Huxley model.

More open: less R or more g = 1/R

The membrane equation

$$I = I_C + I_L + I_{Na} + I_K$$



Voltage-gated resistances

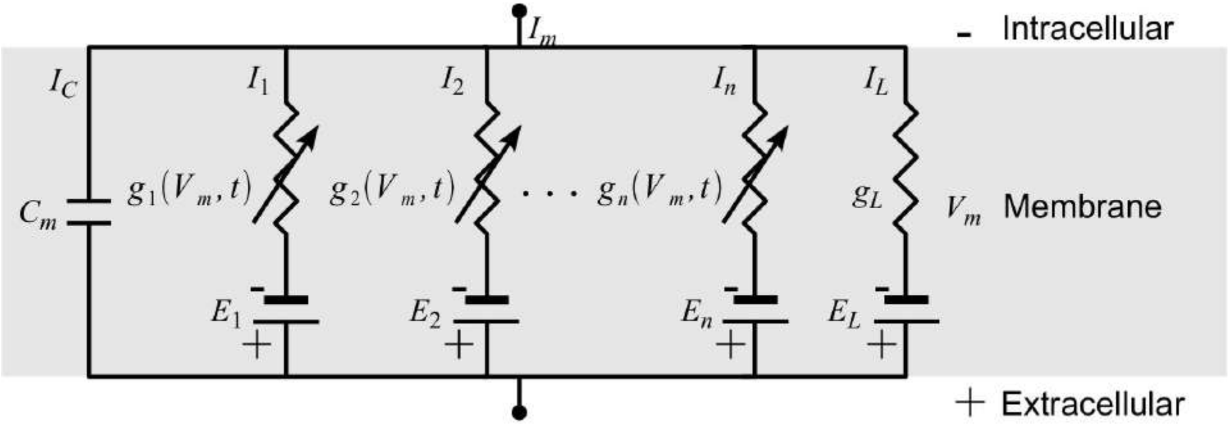
Fig. 2.2 Schematic diagram for the Hodgkin–Huxley model.

More open: less R or more g = 1/R

Variable resistance and therefore variable conductance

Firing: Mainly due to the sodium channels

The membrane equation



$I = I_C + I_L + I_{Na} + I_K$

Voltage-gated resistances

More open: less R or more g = 1/R

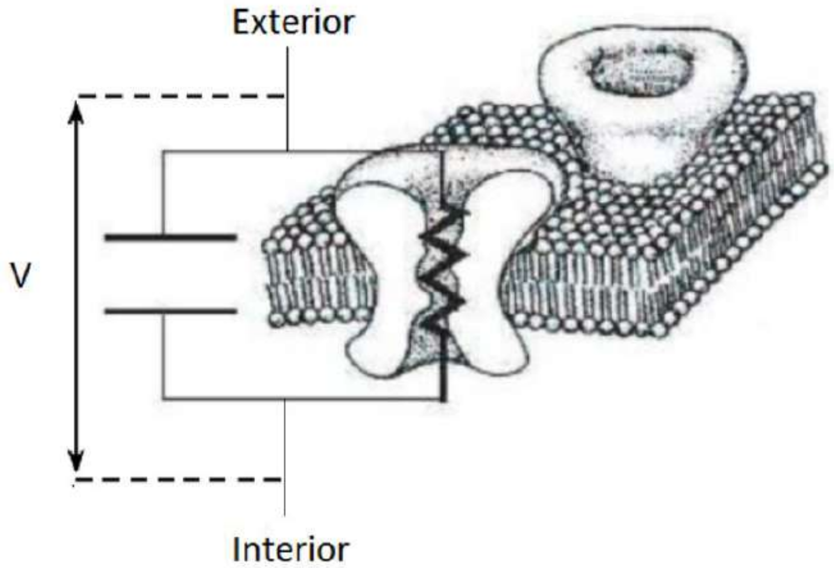
Variable resistance and therefore variable conductance

Firing: Mainly due to the sodium channels

The membrane equation

Using Kirchhoff's law we can write down the membrane equation:

$$C\dot{V} = I - I_{\mathrm{Na}} - I_{\mathrm{Ca}} - I_{\mathrm{K}} - I_{\mathrm{Cl}}$$



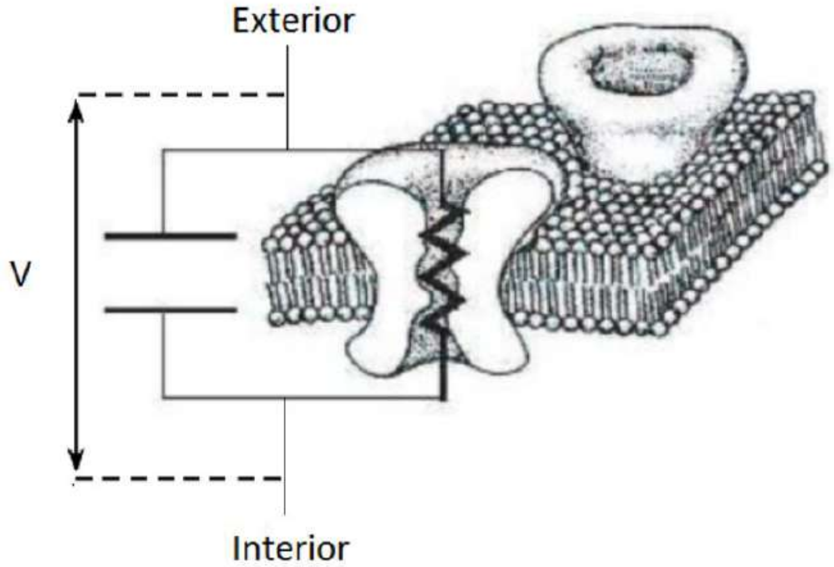
The membrane equation

Using Kirchhoff's law we can write down the membrane equation:

$$C\dot{V} = I - I_{\mathrm{Na}} - I_{\mathrm{Ca}} - I_{\mathrm{K}} - I_{\mathrm{Cl}}$$

Currents:

$$I = -\, g(V_m - E)$$



The membrane equation

Using Kirchhoff’s law we can write down the membrane equation:

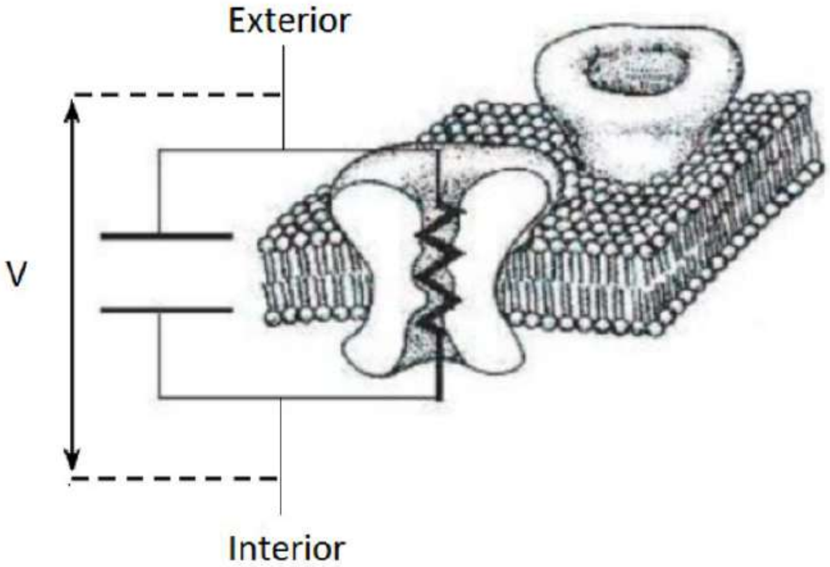
$$C\dot{V} = I - I_{\mathrm{Na}} - I_{\mathrm{Ca}} - I_{\mathrm{K}} - I_{\mathrm{Cl}}$$

Currents:

$$I = -\ g(V_m - E)$$

Membrane conductance

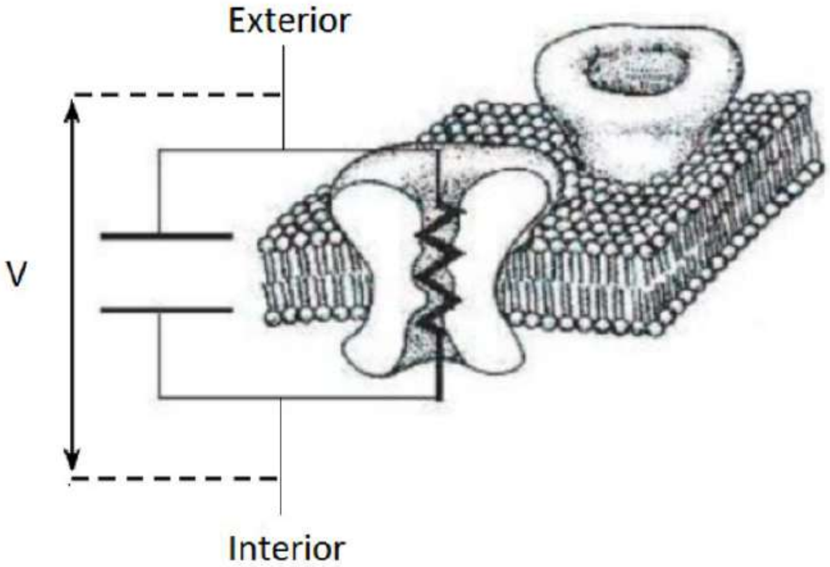
Resting potential



The membrane equation

Using Kirchhoff's law we can write down the membrane equation:

$$C\dot{V} = I - I_{\mathrm{Na}} - I_{\mathrm{Ca}} - I_{\mathrm{K}} - I_{\mathrm{Cl}}$$



Currents:

$$I = - g(V_m - E)$$

Membrane conductance

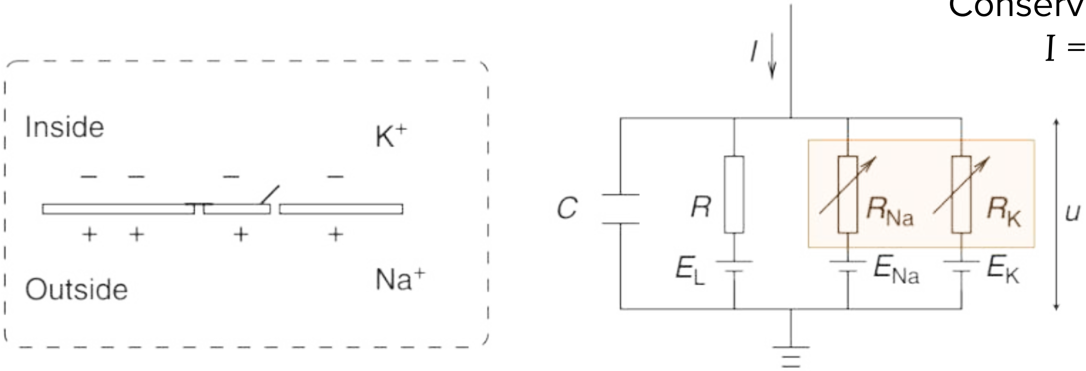
Resting potential

$$C\dot{V} = I - g_{\mathrm{Na}}\left(V - E_{\mathrm{Na}}\right) - g_{\mathrm{Ca}}\left(V - E_{\mathrm{Ca}}\right) - g_{\mathrm{K}}\left(V - E_{\mathrm{K}}\right) - g_{\mathrm{Cl}}\left(V - E_{\mathrm{Cl}}\right)$$

The membrane equation

Conservation of energy:

$$I = I_C + I_L + I_{Na} + I_K$$



Voltage-gated resistances

Fig. 2.2 Schematic diagram for the Hodgkin–Huxley model.

More open: less R or more g = 1/R

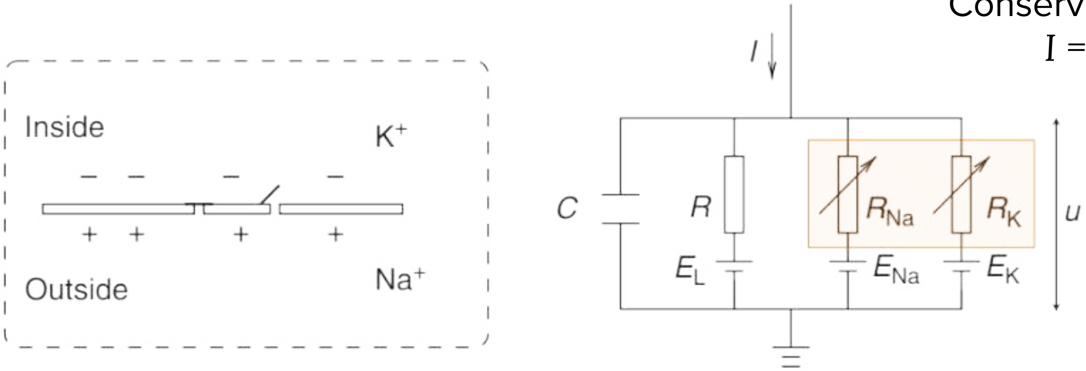
Variable resistance and therefore variable conductance

Firing: Mainly due to the sodium channels

The membrane equation

Conservation of energy:

$I = I_C + I_L + I_{Na} + I_K$



Voltage-gated resistances

Fig. 2.2 Schematic diagram for the Hodgkin–Huxley model.

$$C_m \frac{dV}{dt} = - g_L (V - E_L) - \sum_{k=1}^{n} g_k (V - E_k) + I_e$$

More open: less R or more g = 1/R

Variable resistance and therefore variable conductance

Firing: Mainly due to the sodium channels

Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

The gates may be sensitive to the following factors:

Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

The gates may be sensitive to the following factors:

- Membrane potential.

Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

The gates may be sensitive to the following factors:

- Membrane potential.
  - Example: voltage-gated Na^+ or K^+ channels

Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

The gates may be sensitive to the following factors:

- Membrane potential.
  - Example: voltage-gated Na^+ or K^+ channels
- Intracellular agents (second-messengers).

Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

The gates may be sensitive to the following factors:

- Membrane potential.
  - Example: voltage-gated Na^+ or K^+ channels
- Intracellular agents (second-messengers).
  - Example: Ca^{2+}-gated K^+ channels

Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

The gates may be sensitive to the following factors:

- Membrane potential.
    - Example: voltage-gated Na^+ or K^+ channels
- Intracellular agents (second-messengers).
    - Example: Ca^{2+}-gated K^+ channels
- Extracellular agents (neurotransmitters and neuromodulators).

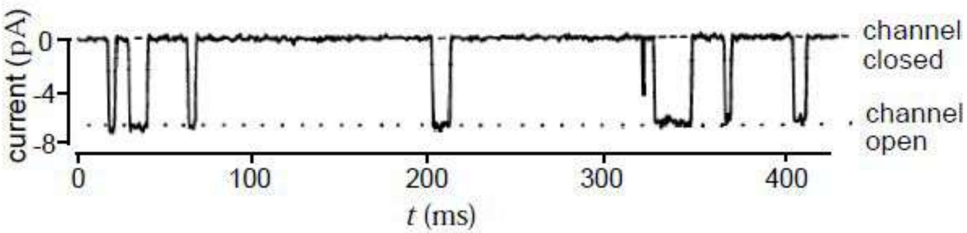
Conductances

Ionic channels are large transmembrane proteins having aqueous pores through which ions can flow down their electrochemical gradients. The electrical conductance of individual channels may be **controlled** by gating particles (**gates**), which switch the channels between open and closed states.

The gates may be sensitive to the following factors:

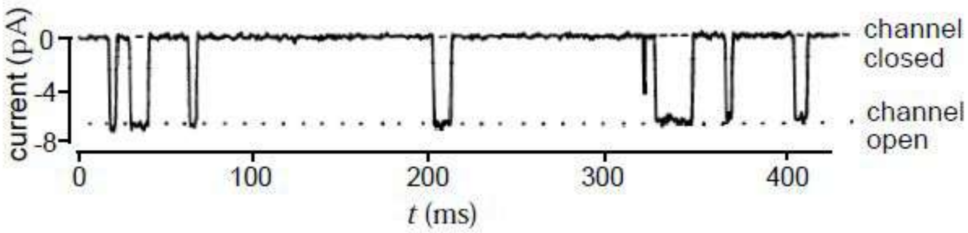
- Membrane potential.
  - Example: voltage-gated Na^+ or K^+ channels
- Intracellular agents (second-messengers).
  - Example: Ca^{2+}-gated K^+ channels
- Extracellular agents (neurotransmitters and neuromodulators).
  - Example: $AMPA_R$, $NMDA_R$, $GABA_R$

Dynamics of Ion Channel State



Dynamics of Ion Channel State

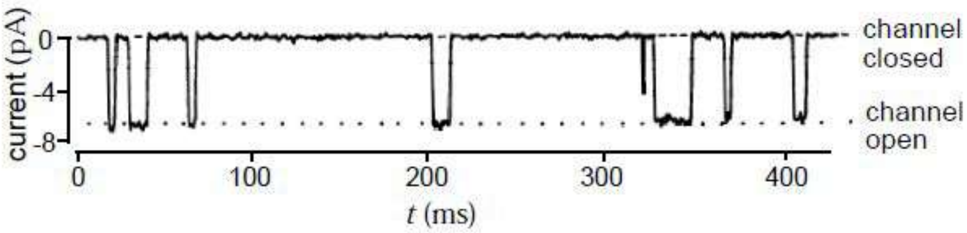
$$I = \bar{g}\, p\, (V - E)$$



Dynamics of Ion Channel State

$$I = \bar{g}\, p\, (V - E)$$

g = maximal conductance
E = the reverse potential of the current
p = the average proportion of channels in the open state

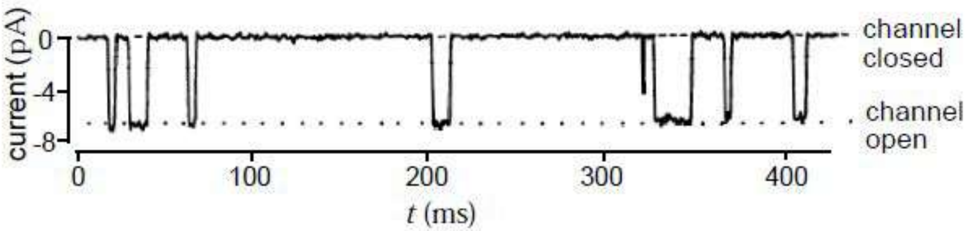


Dynamics of Ion Channel State

g = maximal conductance
E = the reverse potential of the current
p = the average proportion of channels in the open state

$$I = \bar{g}\, p\, (V - E)$$

$$p = m^a\, h^b$$



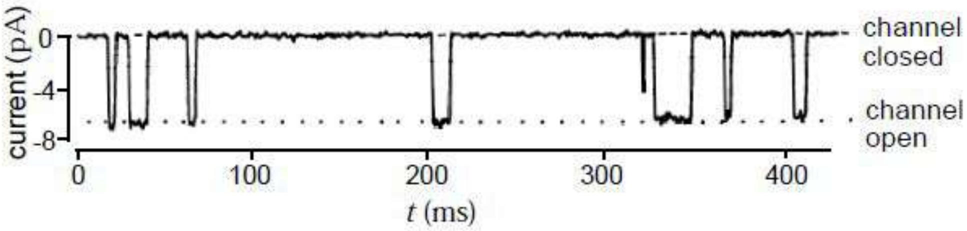
Dynamics of Ion Channel State

$$I = \bar{g}\, p\, (V - E)$$

$$p = m^a\, h^b$$

g = maximal conductance
E = the reverse potential of the current
p = the average proportion of channels in the open state

m = the probability of an activation
h = the probability of an inactivation
a = the number of activation gates
b = the number of inactivation gates



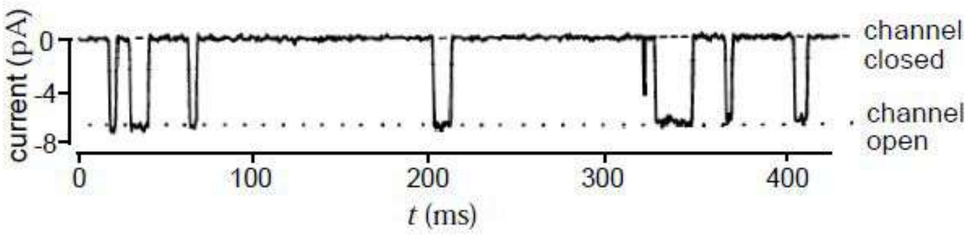
Dynamics of Ion Channel State

$$I = \bar{g}\, p\,(V - E)$$

$$p = m^a\, h^b$$

g = maximal conductance
E = the reverse potential of the current
p = the average proportion of channels in the open state

m = the probability of an activation
h = the probability of an inactivation
a = the number of activation gates
b = the number of inactivation gates



➔	$0 < m < 1$	partially activated
➔	$m = 1$	completely activated
➔	$m = 0$	not activated or deactivated
➔	$h = 0$	inactivated
➔	$h = 1$	released from inactivation or deinactivated
➔	$b = 0$	channels do not have inactivation gates

Dynamics of Ion Channel State

Dynamics of Ion Channel State

$$\dot{m} = (m_\infty(V) - m)/\tau(V)$$

Dynamics of Ion Channel State

$$\dot{m} = (m_\infty(V) - m)/\tau(V)$$

m_∞= the voltage-sensitive steady-state activation function

τ = time constant (from experiments)

Dynamics of Ion Channel State

$$\dot{m} = (m_\infty(V) - m)/\tau(V)$$

m_∞= the voltage-sensitive steady-state activation function
τ = time constant (from experiments)

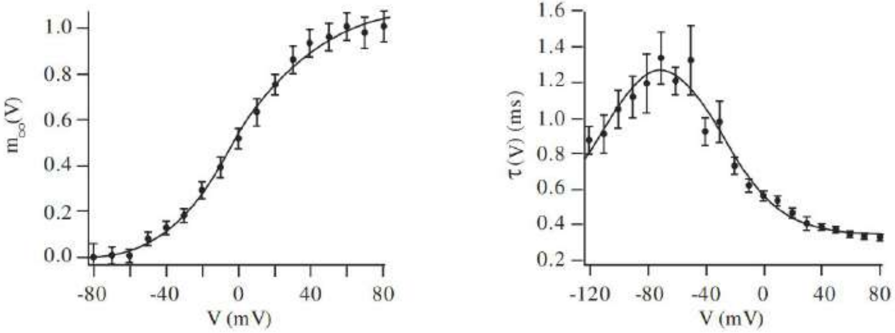


Figure 2.9: The activation function $m_\infty(V)$ and the time constant $\tau(V)$ of the fast transient K^+ current in layer 5 neocortical pyramidal neurons. (Modified from Korngreen and Sakmann 2000.)

Dynamics of Ion Channel State

$$\dot{m} = (m_\infty(V) - m)/\tau(V)$$

$$\dot{h} = (h_\infty(V) - h)/\tau(V)$$

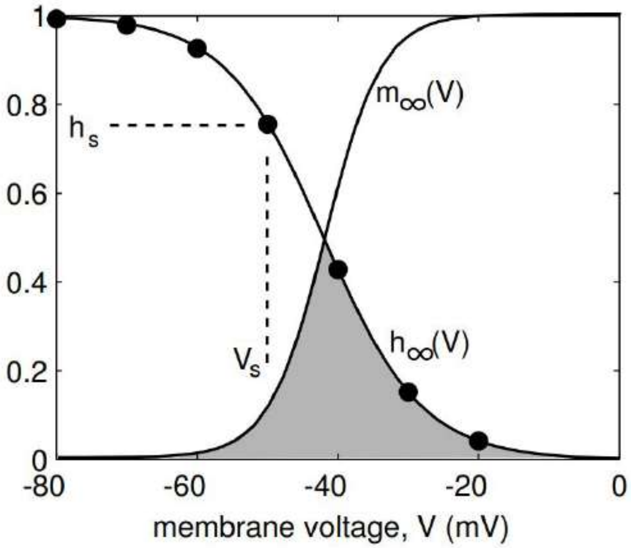
m_∞= the voltage-sensitive steady-state activation function
h_∞= the voltage-sensitive steady-state inactivation function
τ = time constant (from experiments)

Dynamics of Ion Channel State

$$\dot{m} = (m_\infty(V) - m)/\tau(V)$$

$$\dot{h} = (h_\infty(V) - h)/\tau(V)$$

m_∞= the voltage-sensitive steady-state activation function
h_∞= the voltage-sensitive steady-state inactivation function
τ = time constant (from experiments)



The Hodgkin-Huxley model

A.L. Hodgkin

A. Huxley

The Hodgkin-Huxley model

Nobel prize 1963

A.L. Hodgkin

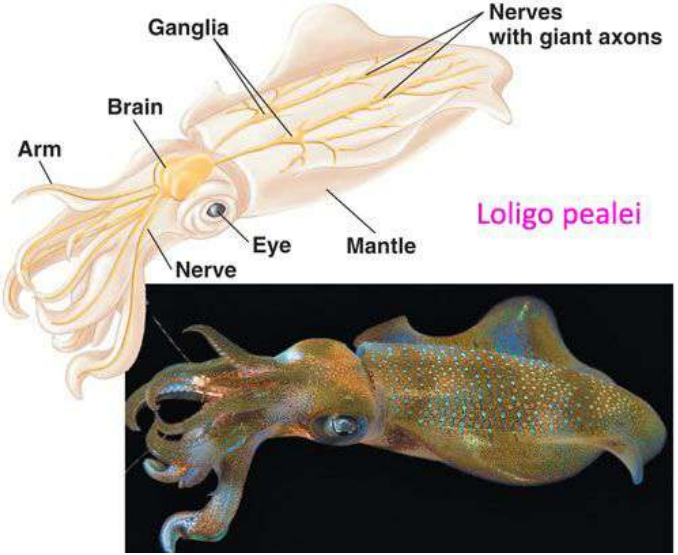
A. Huxley

The Hodgkin-Huxley model

Nobel prize 1963

A.L. Hodgkin

A. Huxley



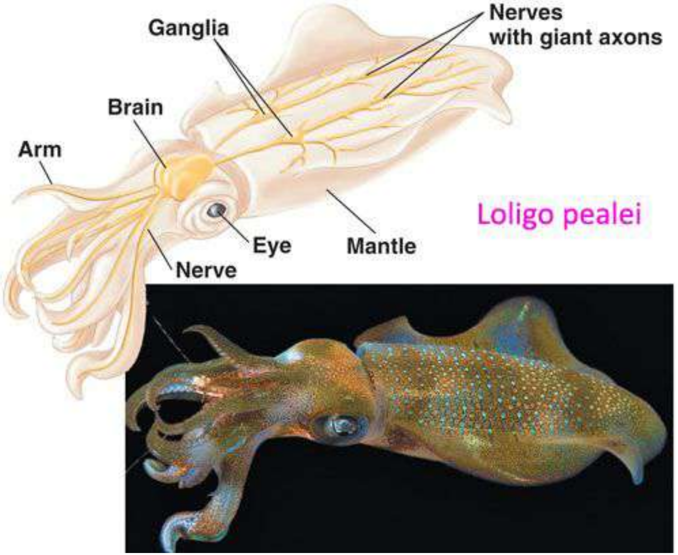
The Hodgkin-Huxley model

$$C\dot{V} = I - \overbrace{\bar{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}})}^{I_{\mathrm{K}}} - \overbrace{\bar{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}})}^{I_{\mathrm{Na}}} - \overbrace{g_{\mathrm{L}} (V - E_{\mathrm{L}})}^{I_{\mathrm{L}}}$$

Nobel prize 1963

A.L. Hodgkin

A. Huxley



Loligo pealei

The Hodgkin-Huxley model

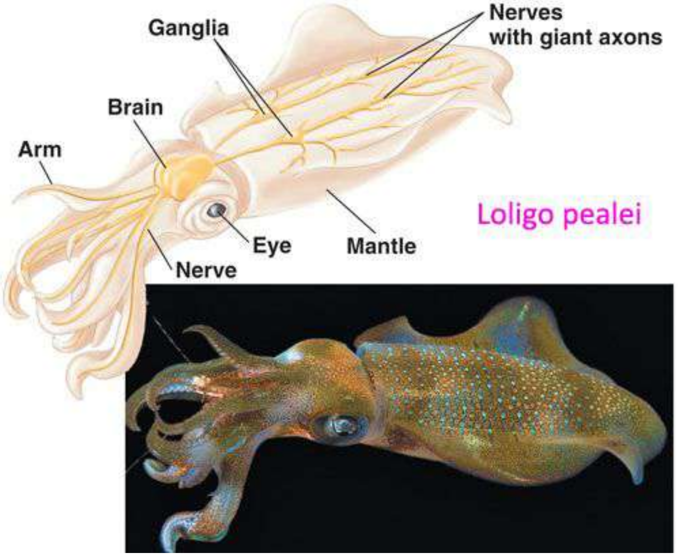
$$C\dot{V} = I - \overbrace{\bar{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}})}^{I_{\mathrm{K}}} - \overbrace{\bar{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}})}^{I_{\mathrm{Na}}} - \overbrace{g_{\mathrm{L}} (V - E_{\mathrm{L}})}^{I_{\mathrm{L}}}$$

$$\begin{aligned}
\dot{n} &= (n_\infty(V) - n)/\tau_n(V)\ , \\
\dot{m} &= (m_\infty(V) - m)/\tau_m(V)\ , \\
\dot{h} &= (h_\infty(V) - h)/\tau_h(V)\ ,
\end{aligned}$$

Nobel prize 1963

A.L. Hodgkin

A. Huxley



Loligo pealei

The Hodgkin-Huxley model

Nobel prize 1963

A.L. Hodgkin

A. Huxley

$$C\dot{V} = I - \overbrace{\bar{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}})}^{I_{\mathrm{K}}} - \overbrace{\bar{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}})}^{I_{\mathrm{Na}}} - \overbrace{g_{\mathrm{L}} (V - E_{\mathrm{L}})}^{I_{\mathrm{L}}}$$

$$\begin{aligned}
\dot{n} &= (n_\infty(V) - n)/\tau_n(V) \, , \\
\dot{m} &= (m_\infty(V) - m)/\tau_m(V) \, , \\
\dot{h} &= (h_\infty(V) - h)/\tau_h(V) \, ,
\end{aligned}$$

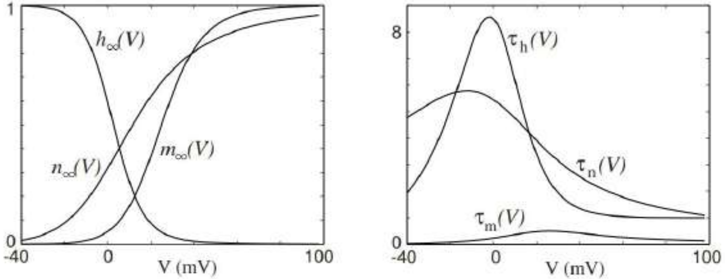
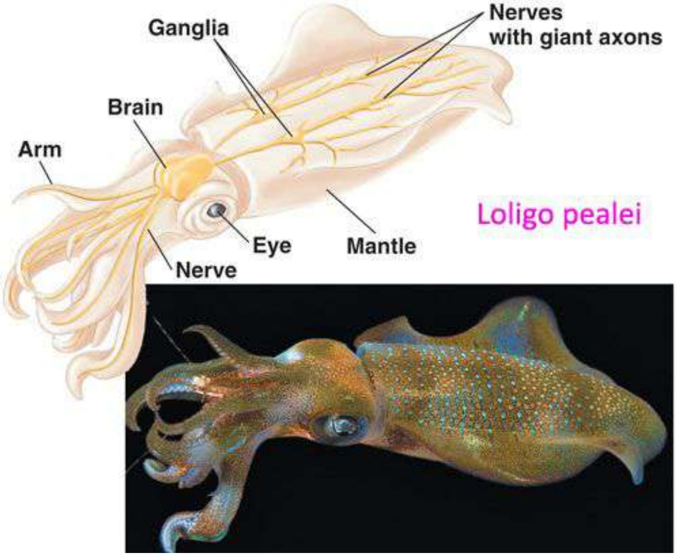


Figure 2.13: Steady-state (in)activation functions (left) and voltage-dependent time constants (right) in the Hodgkin-Huxley model.



Loligo pealei

The Hodgkin-Huxley model

$$C_m \frac{dV}{dt} = -g_L(V - E_L) - \bar{g}_K n^4 (V - E_K) - \bar{g}_{Na} m^3 h (V - E_{Na}) + I_e$$

$$\frac{dn}{dt} = (1 - n) \cdot \alpha_n(V) - n \cdot \beta_n(V)$$

$$\frac{dm}{dt} = (1 - m) \cdot \alpha_m(V) - m \cdot \beta_m(V)$$

$$\frac{dh}{dt} = (1 - h) \cdot \alpha_h(V) - h \cdot \beta_h(V)$$

$$\alpha_n = \frac{0.01 \cdot (V + 55)}{1 - \exp(-0.1 \cdot (V + 55))} \qquad \beta_n = 0.125 \cdot \exp(-0.0125 \cdot (V + 65))$$

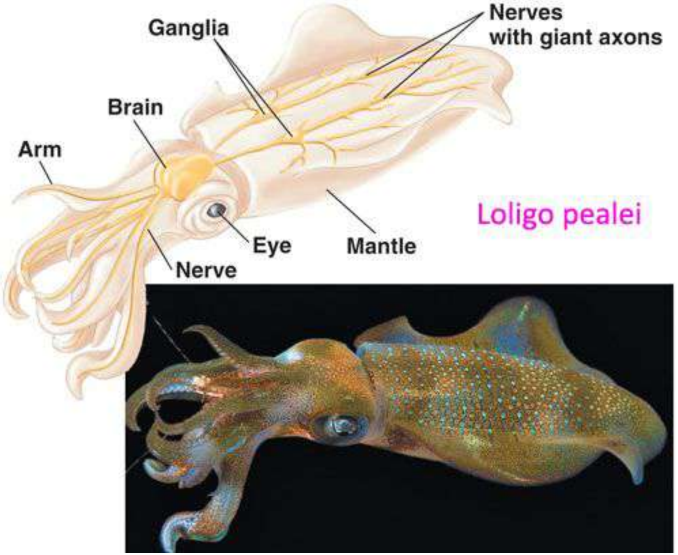
$$\alpha_m = \frac{0.1 \cdot (V + 40)}{1 - \exp(-0.1 \cdot (V + 40))} \qquad \beta_m = 4 \cdot \exp(-0.0556 \cdot (V + 65))$$

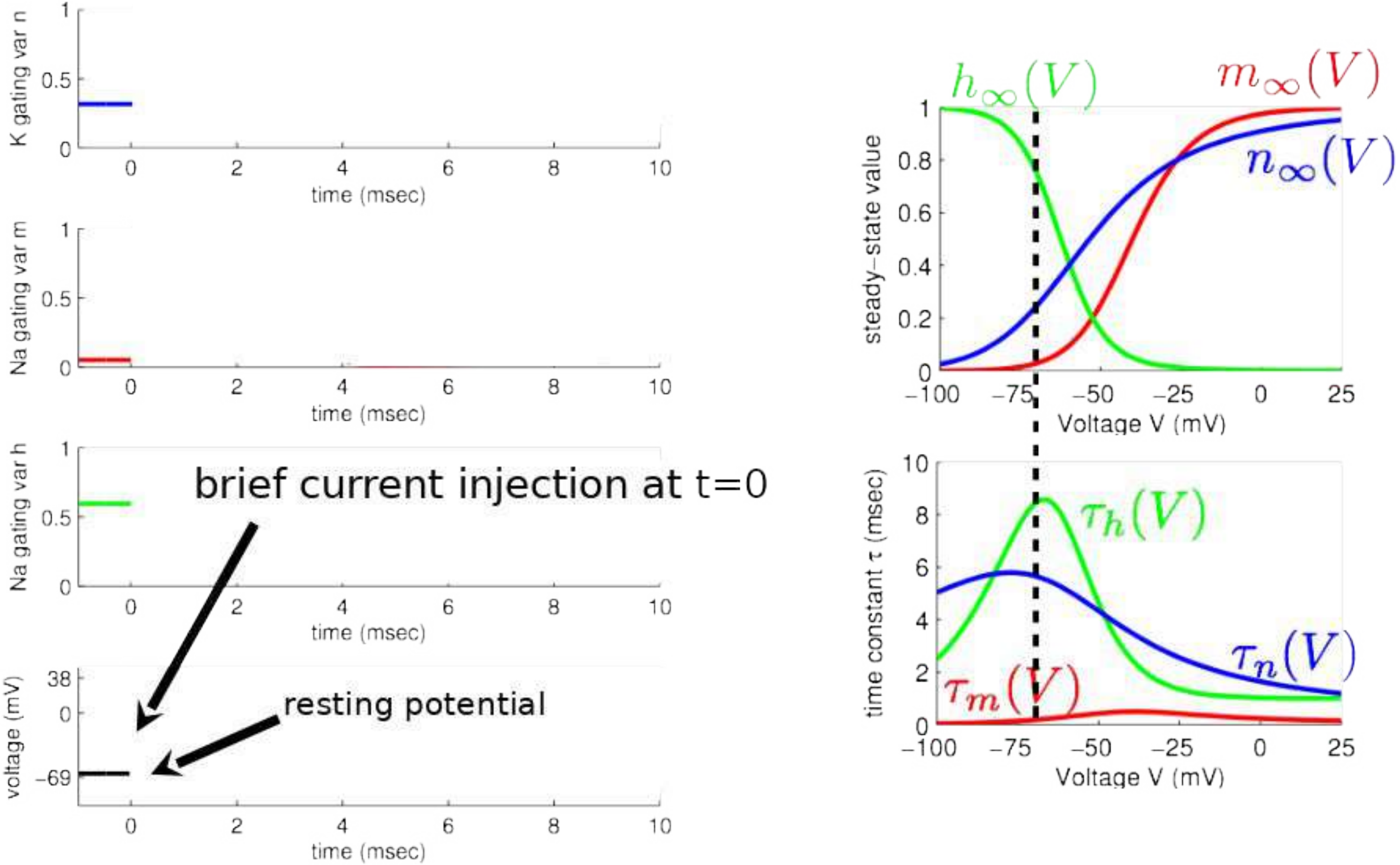
$$\alpha_h = 0.07 \cdot \exp(-0.05 \cdot (V + 65)) \qquad \beta_h = \frac{1}{1 + \exp(-0.1 \cdot (V + 35))}$$

Nobel prize 1963

A.L. Hodgkin

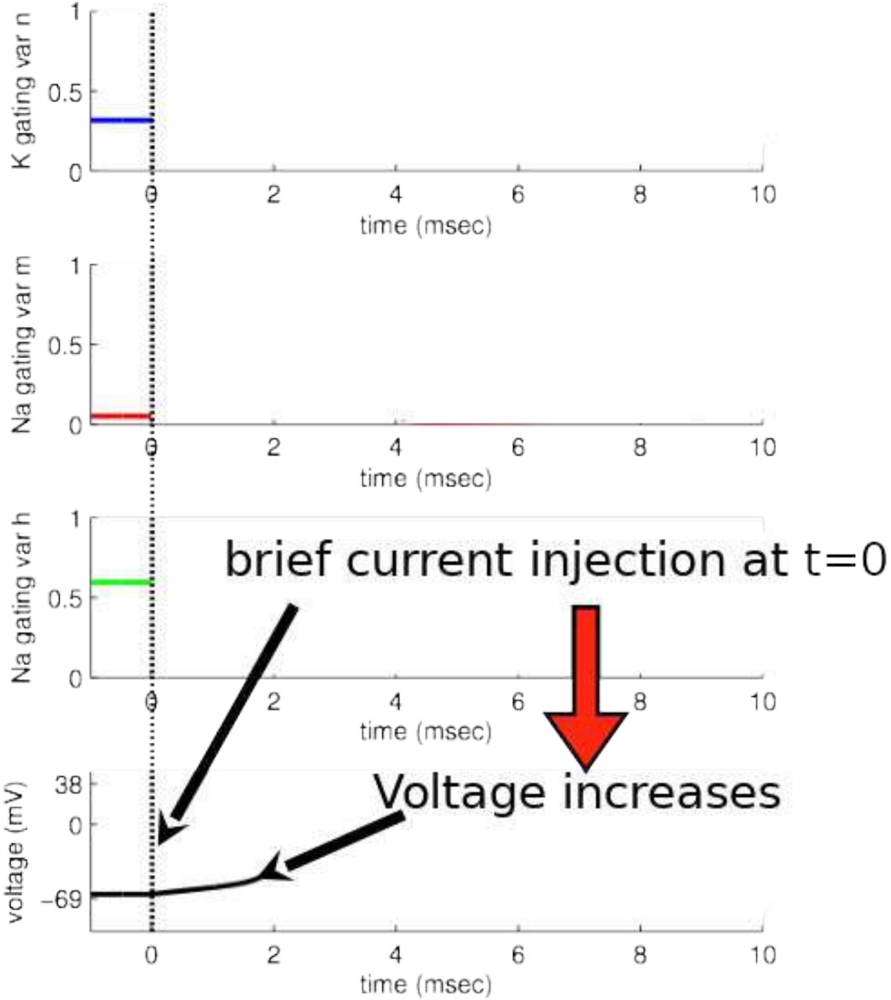
A. Huxley





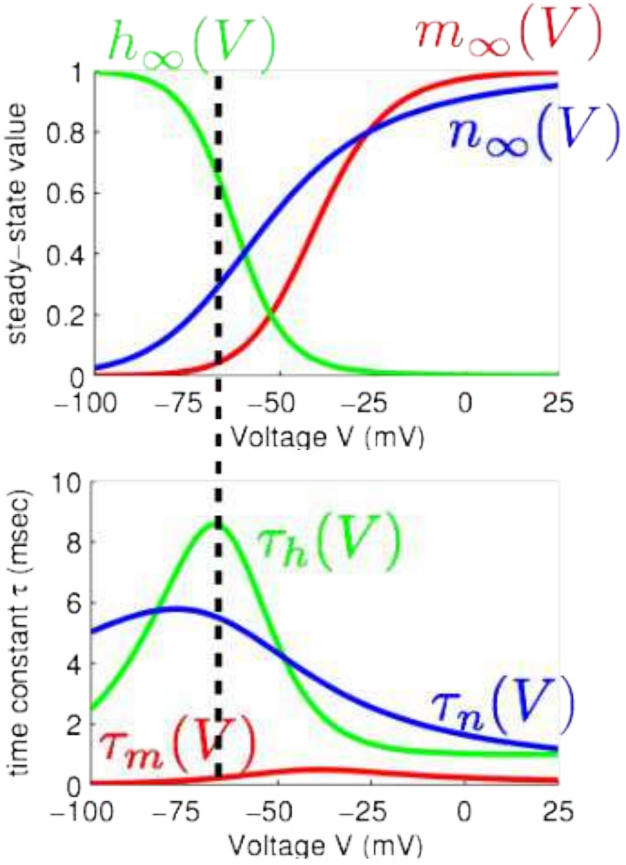
brief current injection at t=0

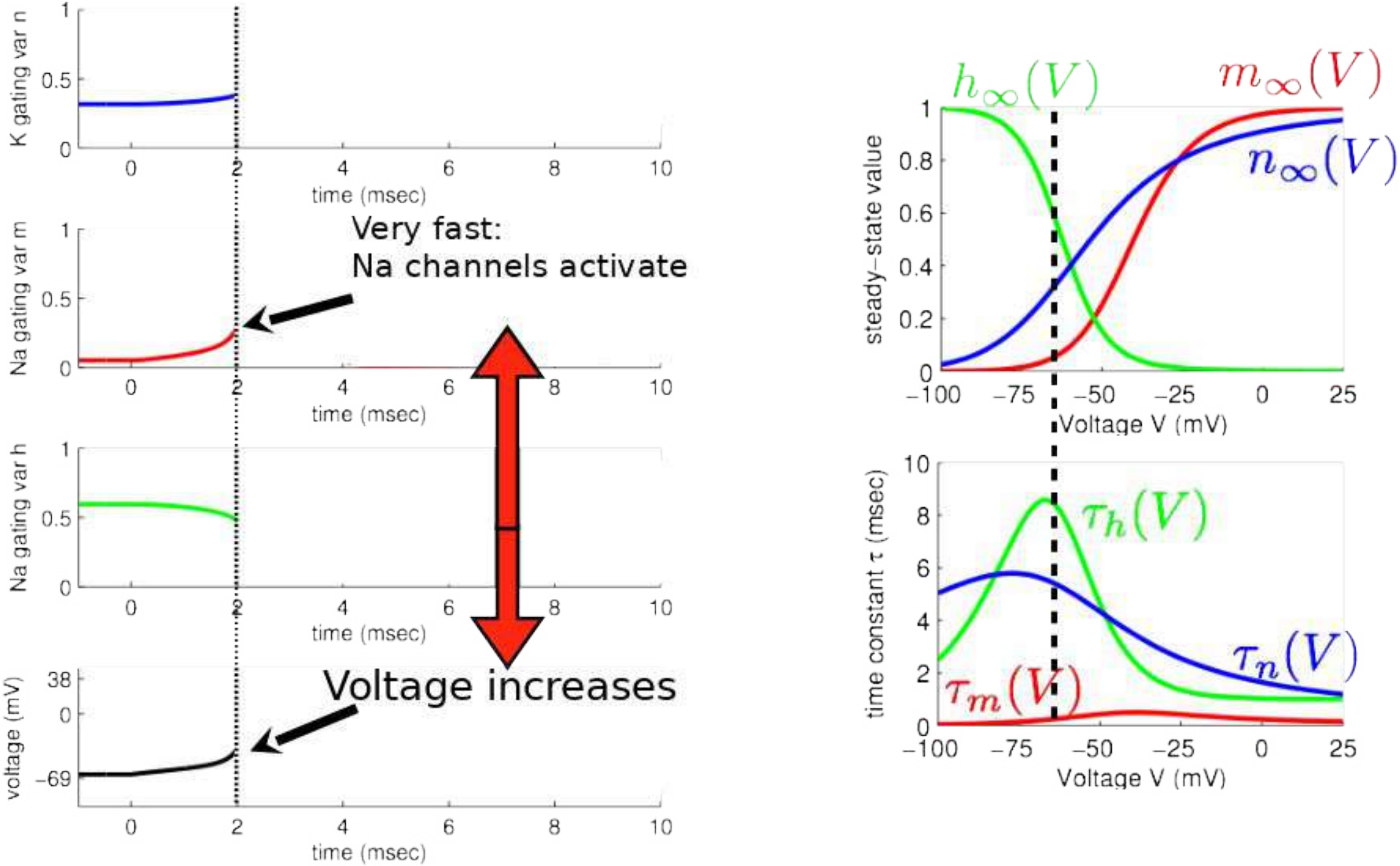
resting potential



brief current injection at t=0

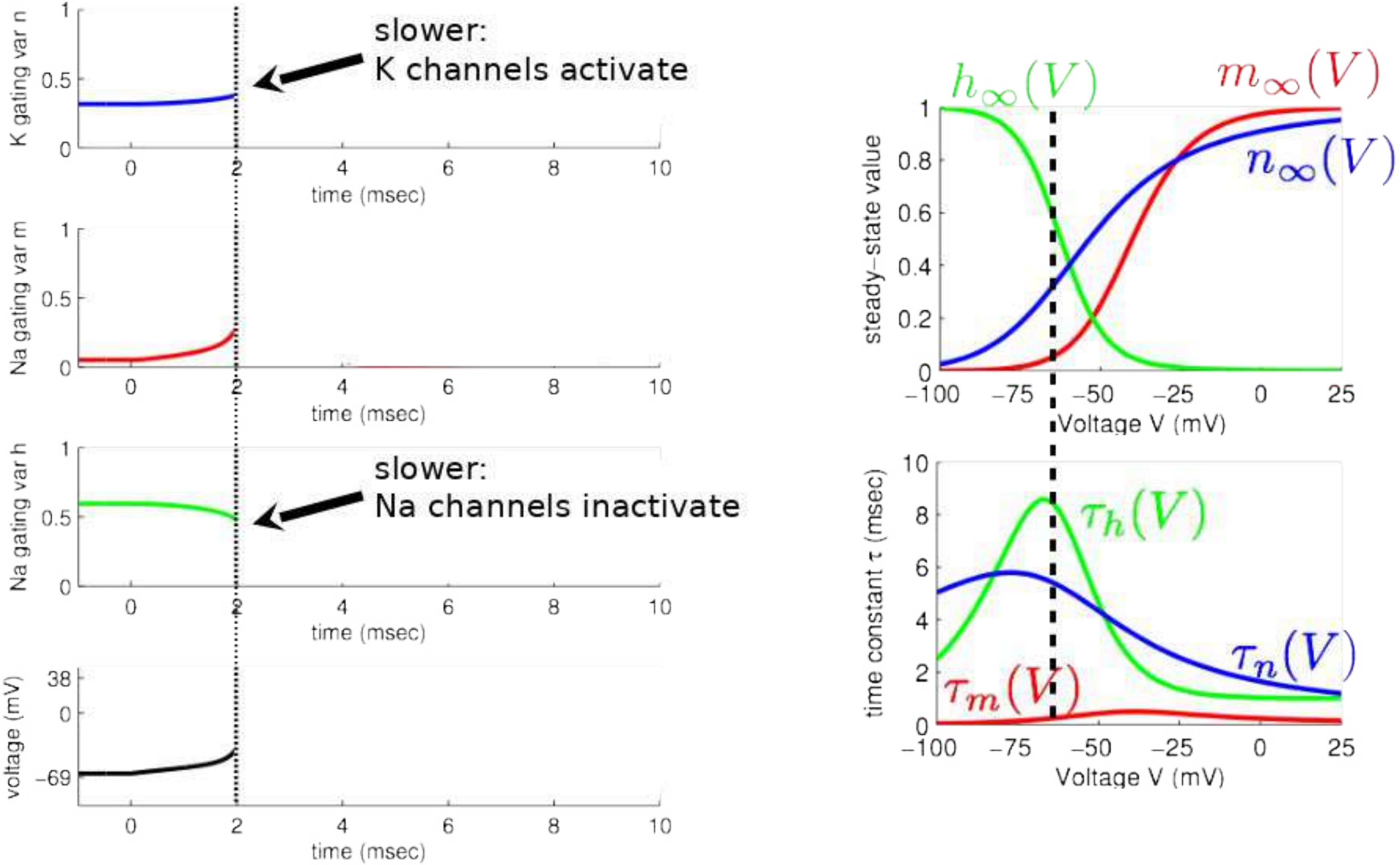
Voltage increases





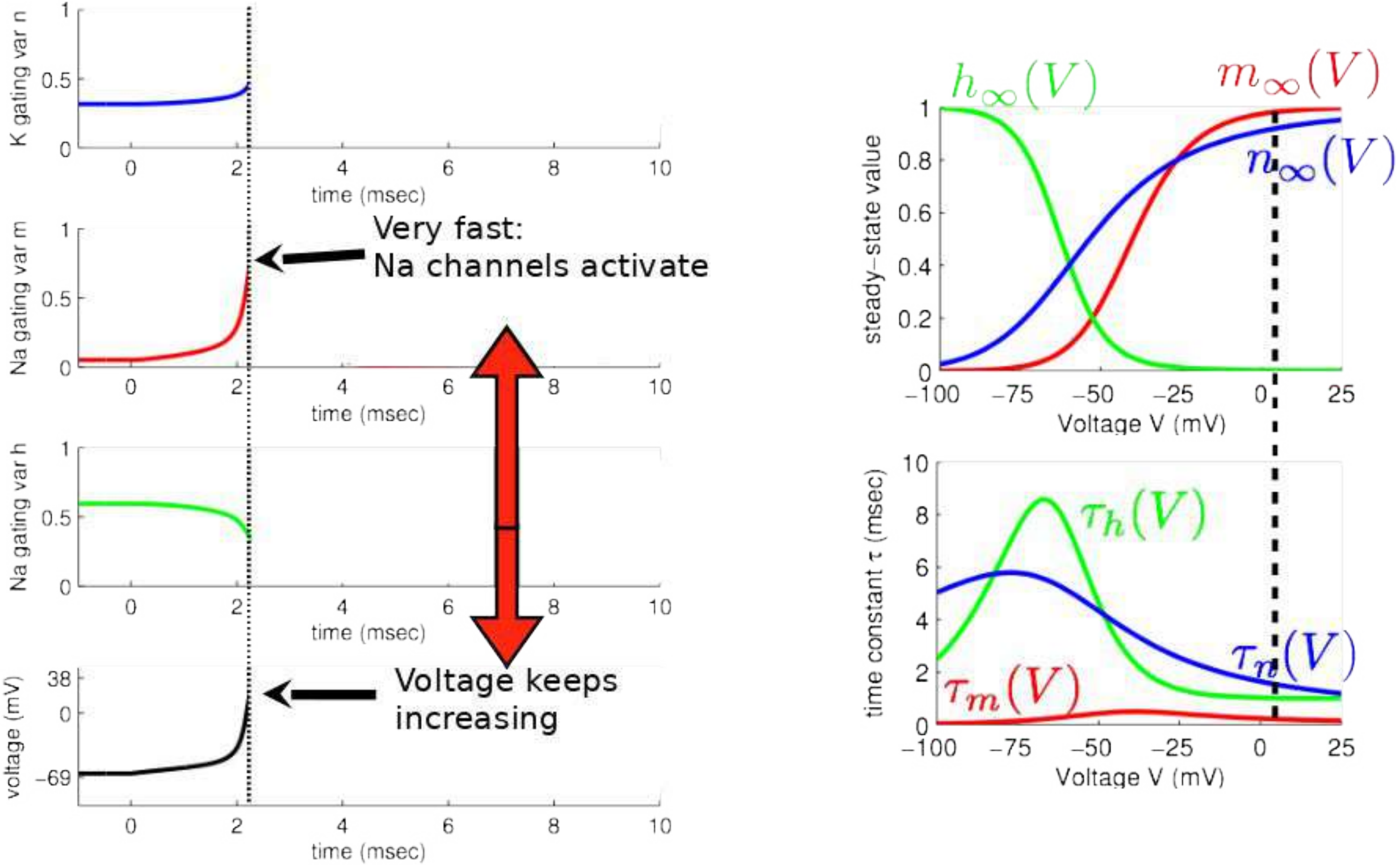
Very fast:
Na channels activate

Voltage increases



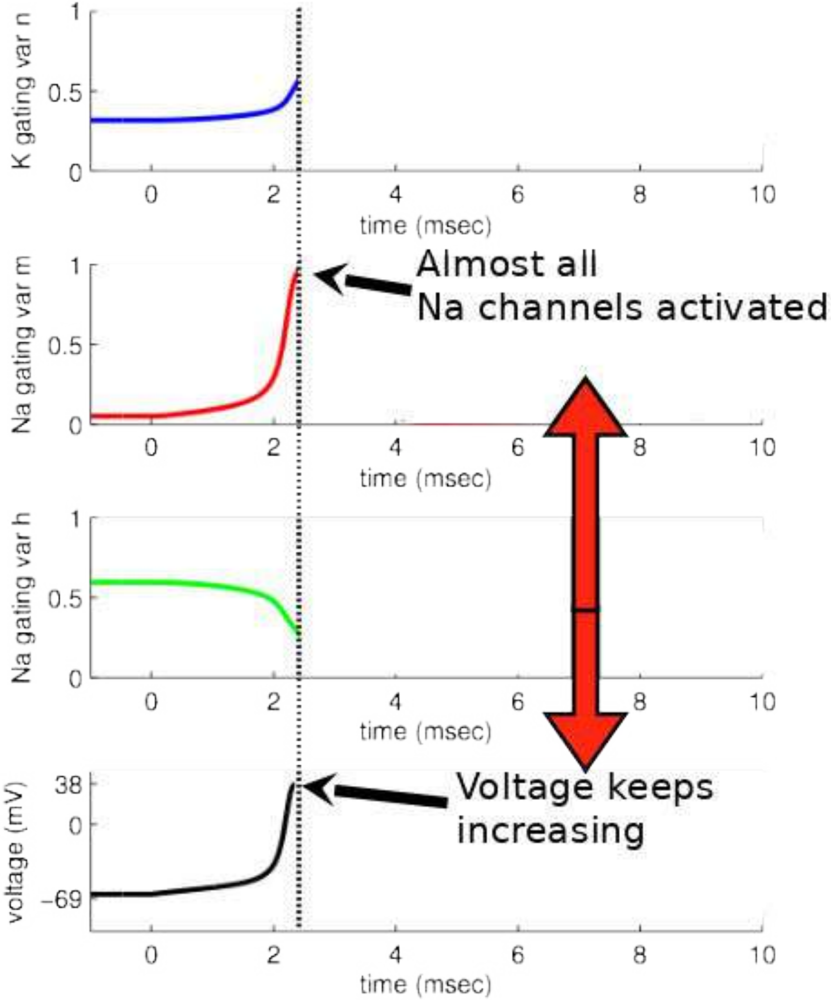
slower:
K channels activate

slower:
Na channels inactivate



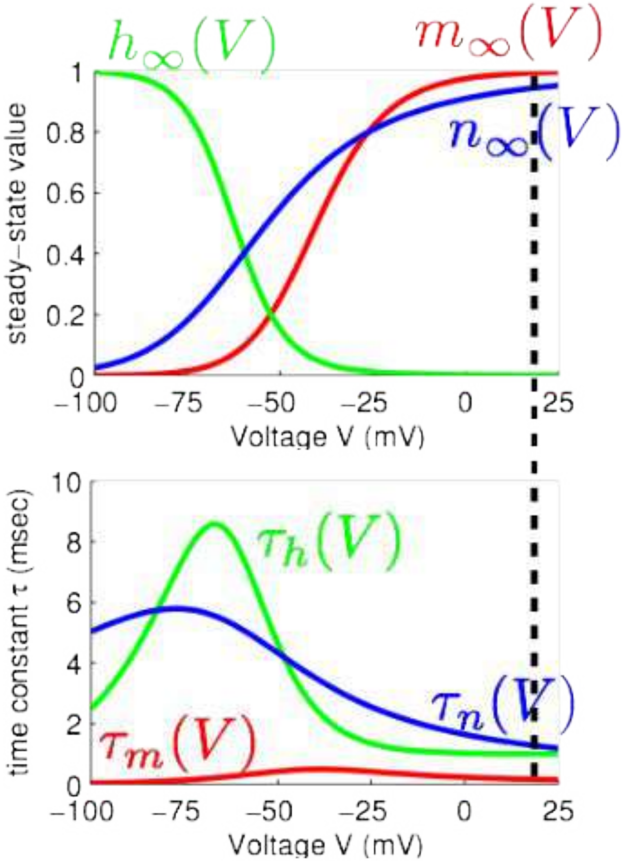
Very fast:
Na channels activate

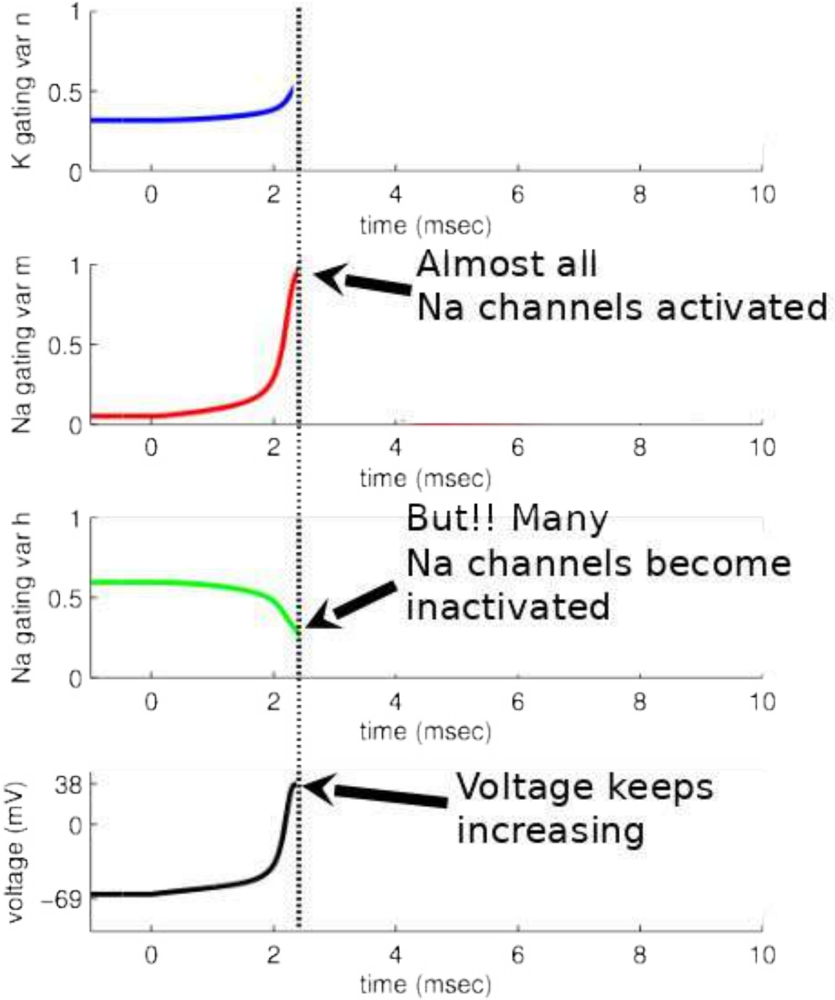
Voltage keeps
increasing



Almost all
Na channels activated

Voltage keeps
increasing

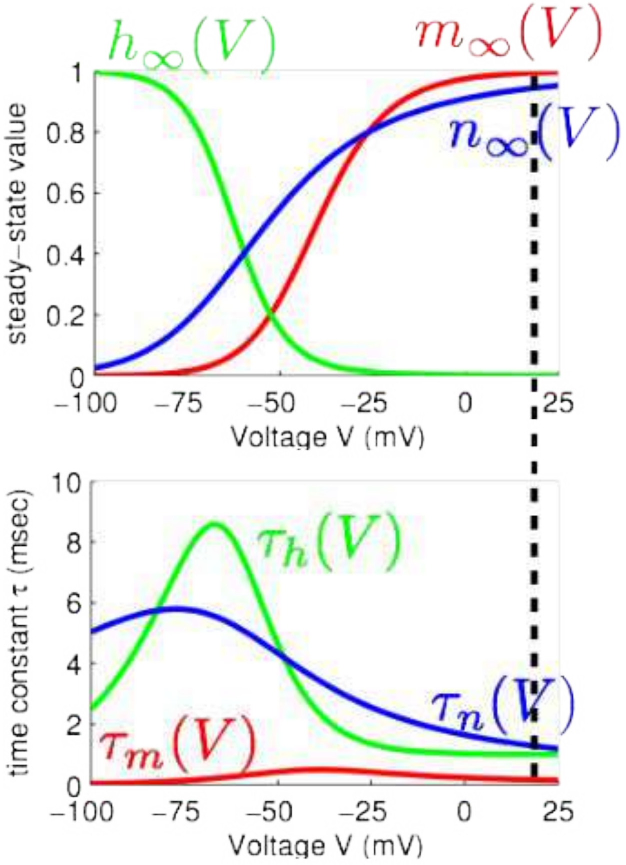


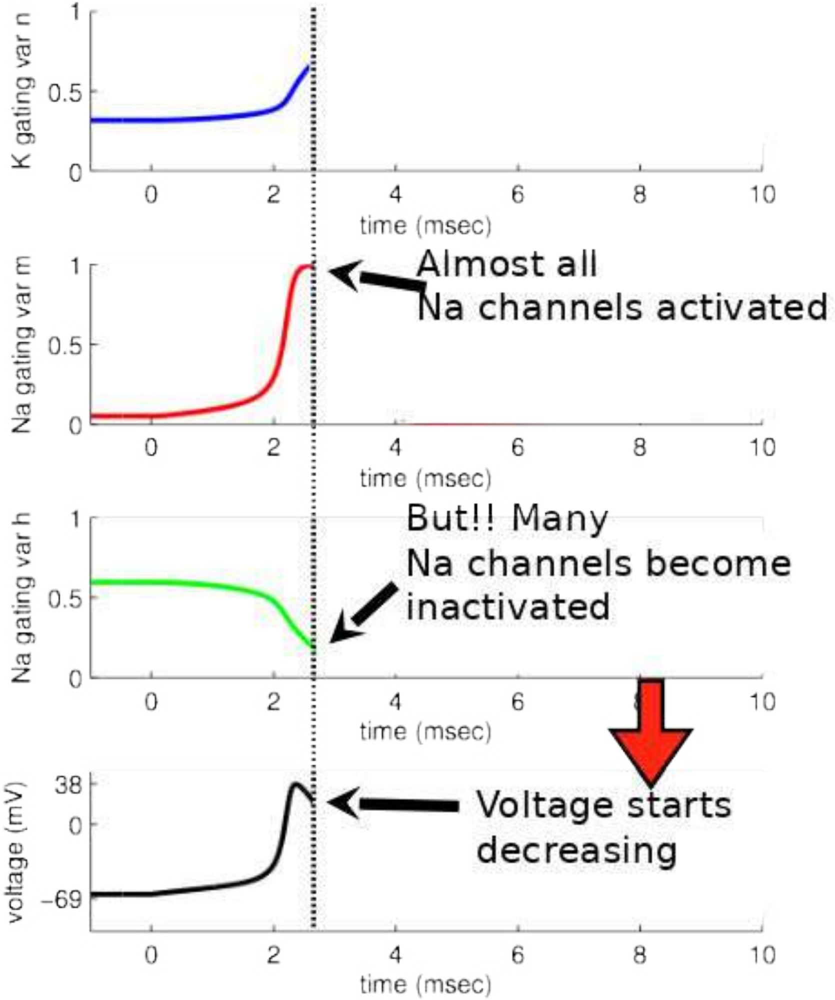


Almost all
Na channels activated

But!! Many
Na channels become
inactivated

Voltage keeps
increasing

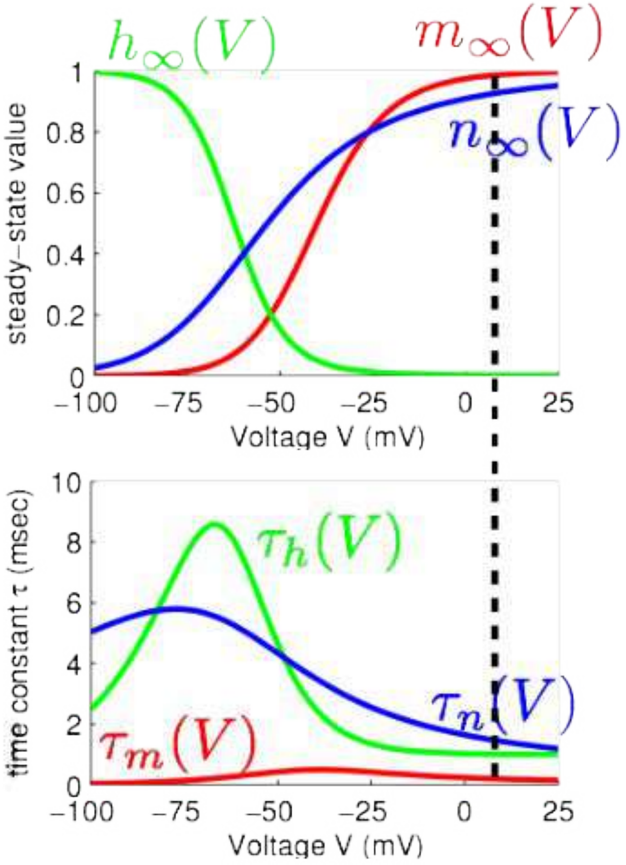


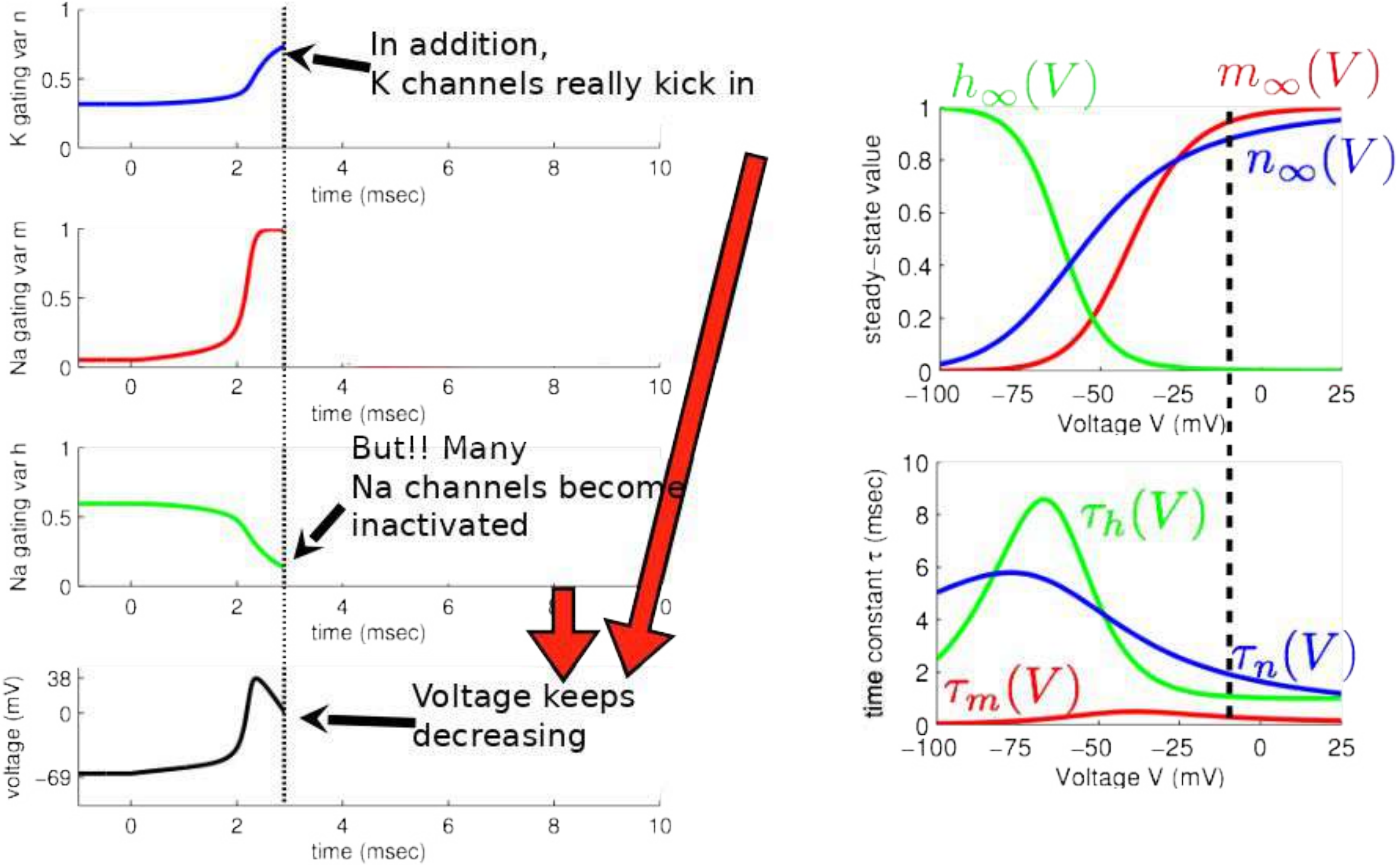


Almost all
Na channels activated

But!! Many
Na channels become
inactivated

Voltage starts
decreasing

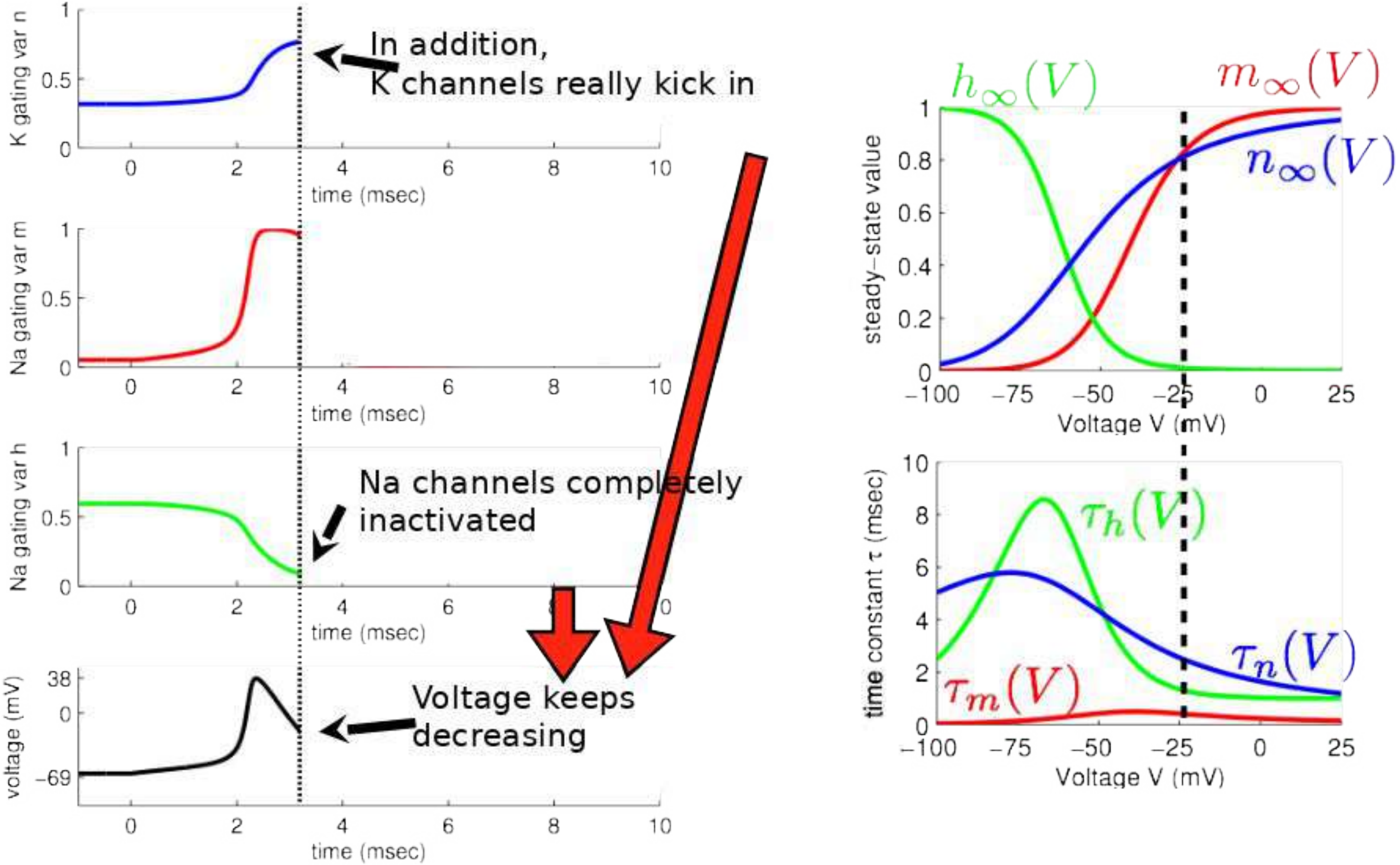




In addition,
K channels really kick in

But!! Many
Na channels become
inactivated

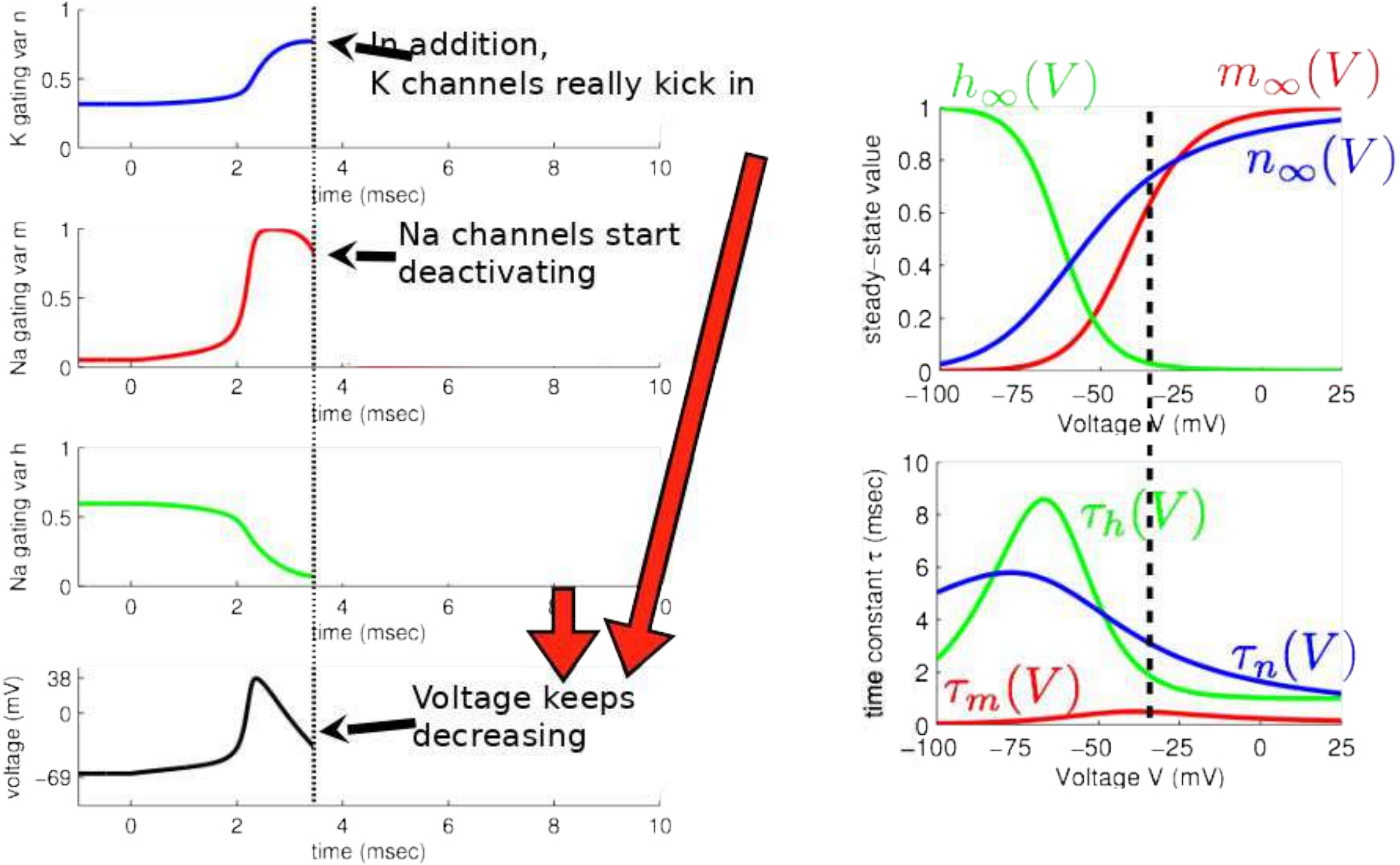
Voltage keeps
decreasing



In addition,
K channels really kick in

Na channels completely
inactivated

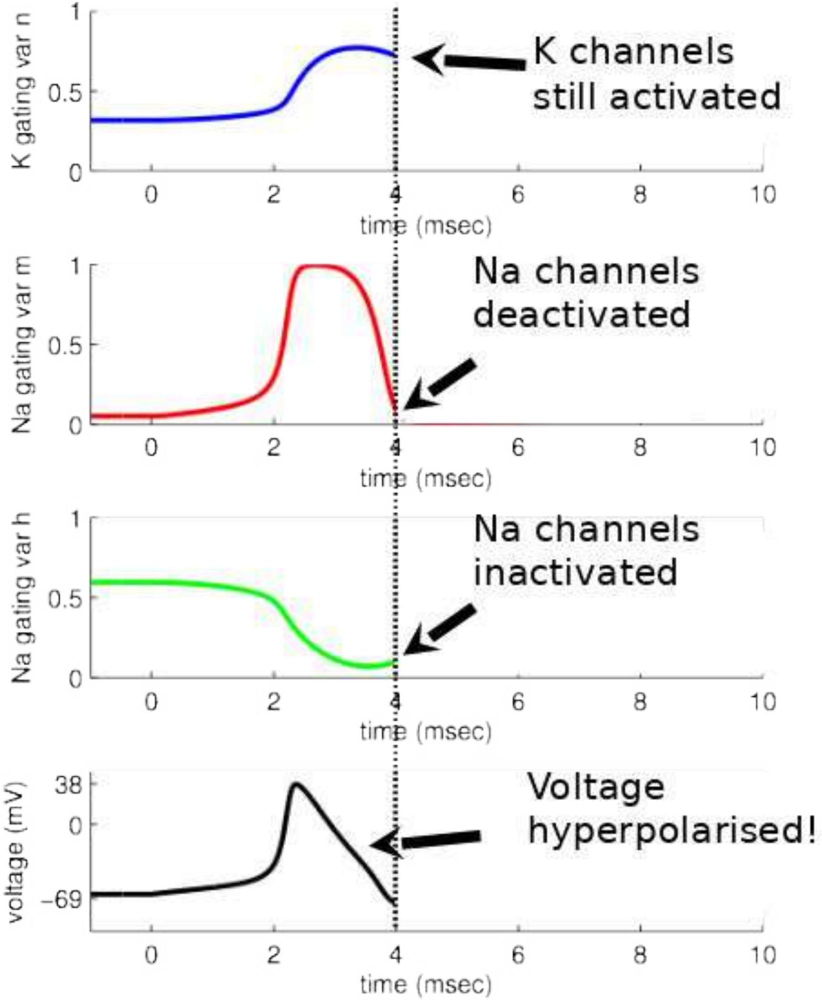
Voltage keeps
decreasing



In addition,
K channels really kick in

Na channels start
deactivating

Voltage keeps
decreasing

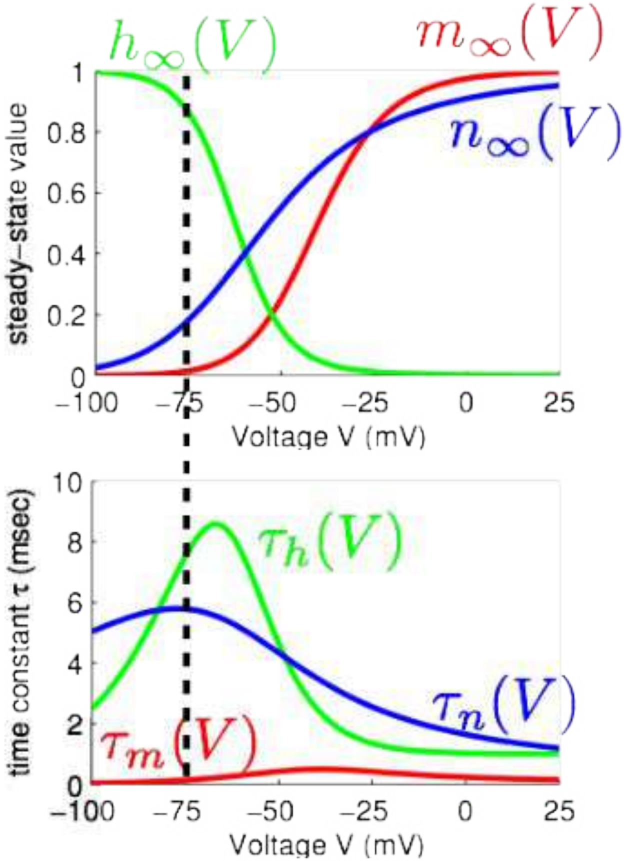


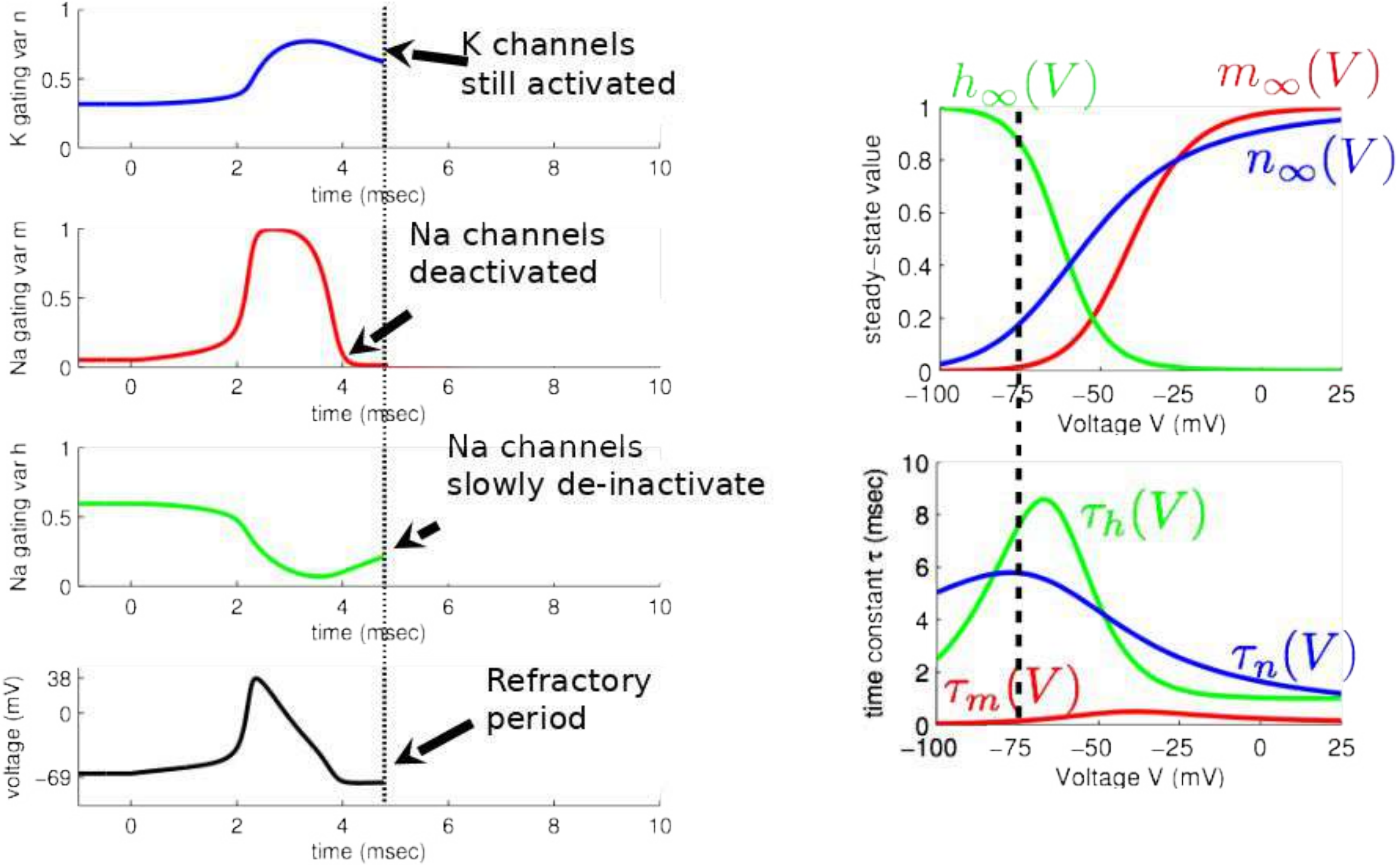
K channels still activated

Na channels deactivated

Na channels inactivated

Voltage hyperpolarised!



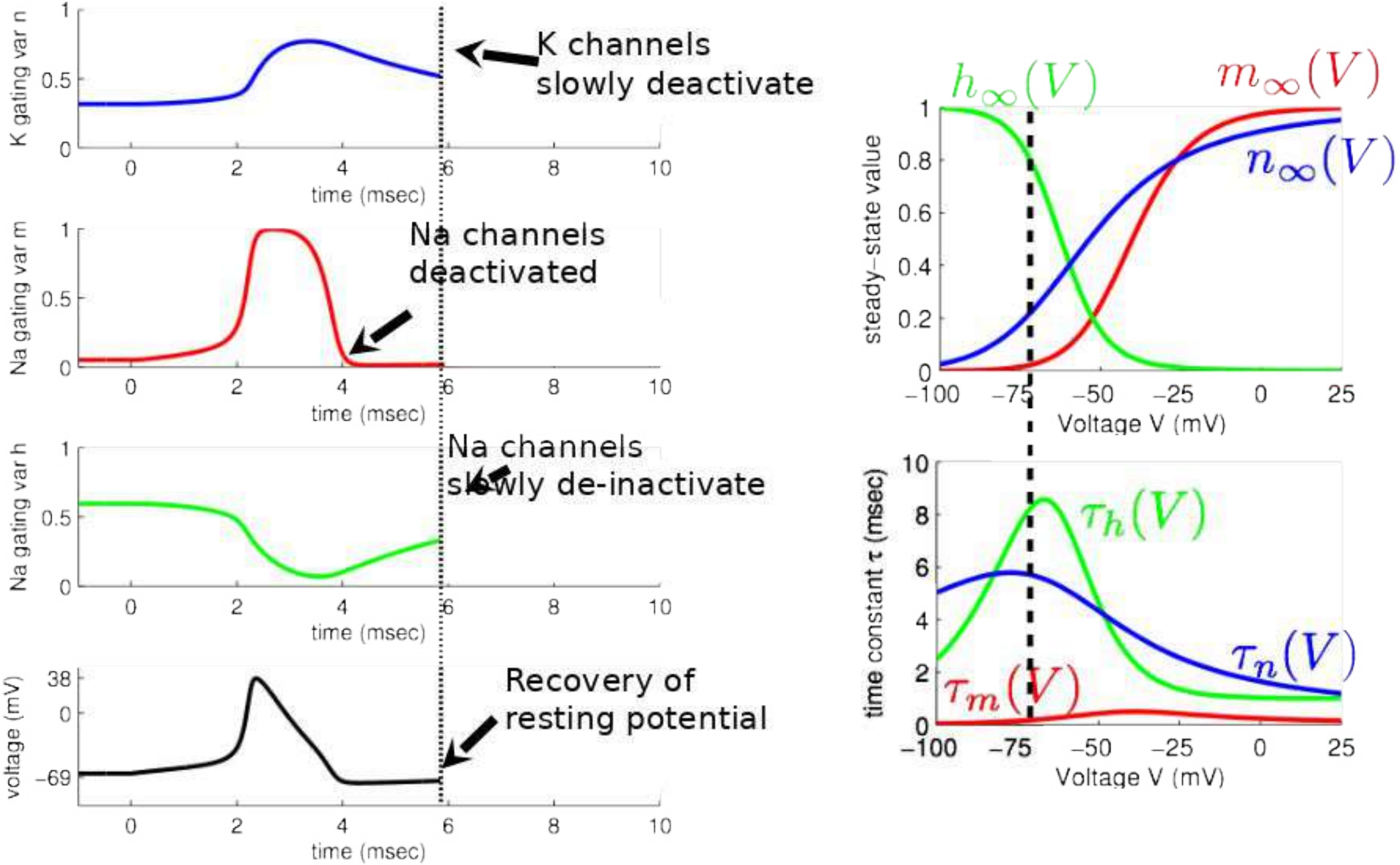


K channels still activated

Na channels deactivated

Na channels slowly de-inactivate

Refractory period

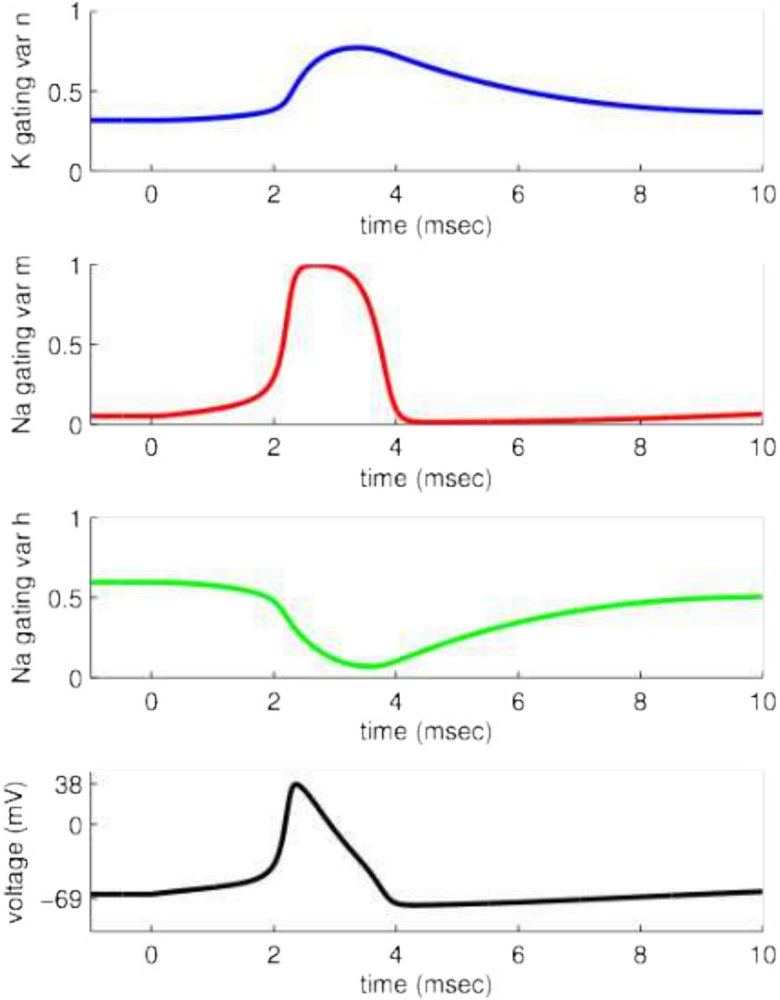


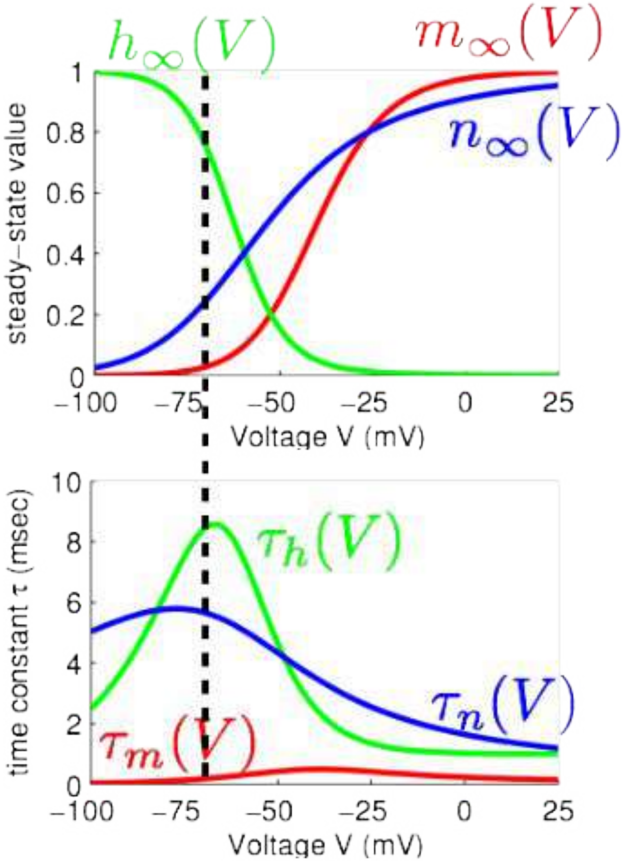
K channels slowly deactivate

Na channels deactivated

Na channels slowly de-inactivate

Recovery of resting potential


K gating var n
Na gating var m
Na gating var h
voltage (mV)
38
0
-69
1
0.5
0
0
2
4
6
8
10
time (msec)

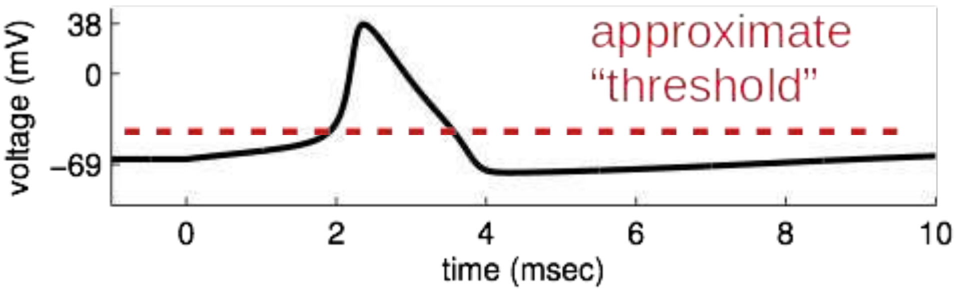

$h_\infty(V)$
$m_\infty(V)$
$n_\infty(V)$
steady-state value
1
0.8
0.6
0.4
0.2
0
−100
−75
−50
−25
0
25
Voltage V (mV)
$\tau_h(V)$
$\tau_n(V)$
$\tau_m(V)$
time constant τ (msec)
10
8
6
4
2
0
Voltage V (mV)

The threshold in the HH model

- The threshold is embedded in the use of these activation variables (m, n, and h)
- The activation variables need to be measured
- i.e., the threshold is not a fitting parameter and there is no V_{reset} that needs to be added

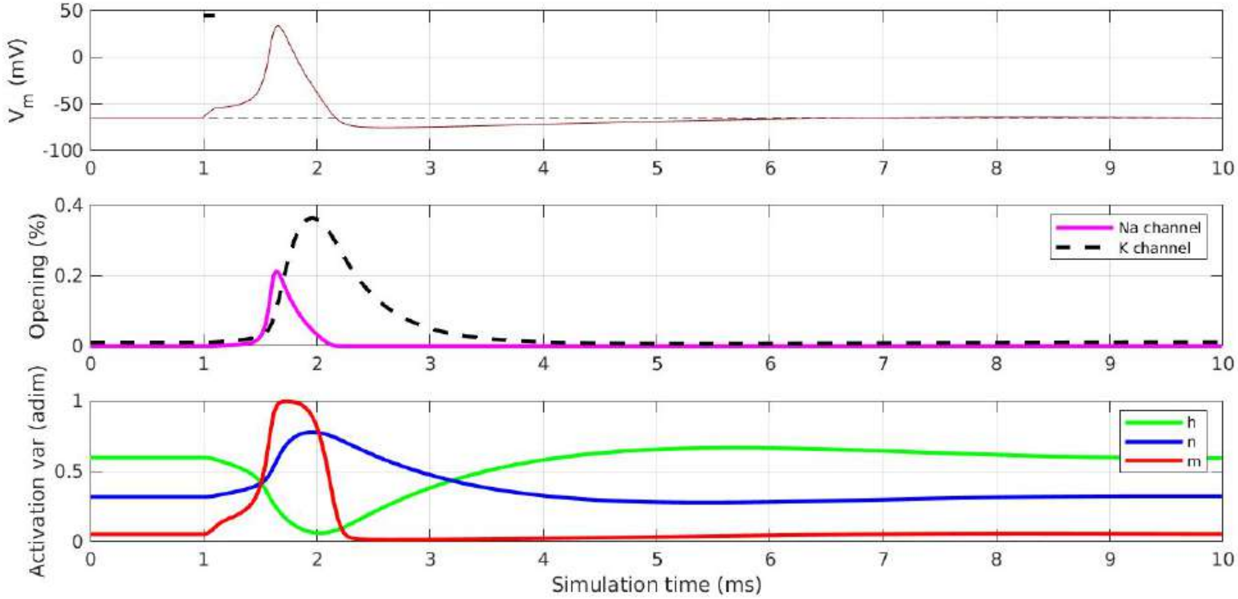
The threshold in the HH model

- The threshold is embedded in the use of these activation variables (m, n, and h)
- The activation variables need to be measured
- i.e., the threshold is not a fitting parameter and there is no V_{reset} that needs to be added



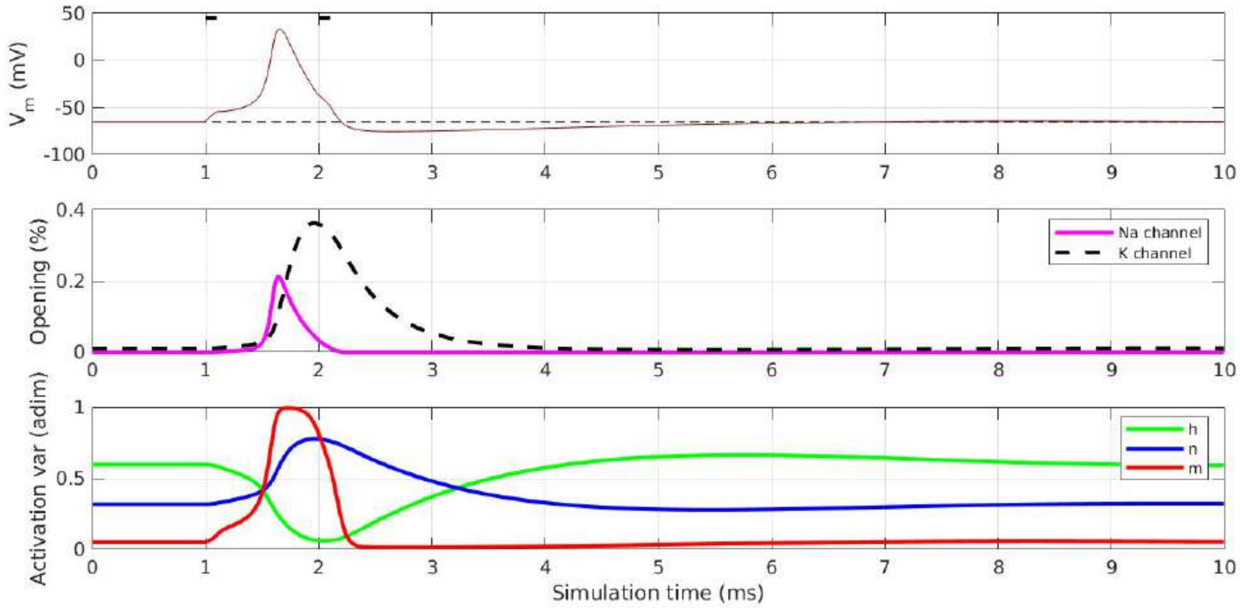
Simulations: Extra insights into the HH model

Injected current pulse of 0.1 ms at t = 1 ms, I_0=100 uA



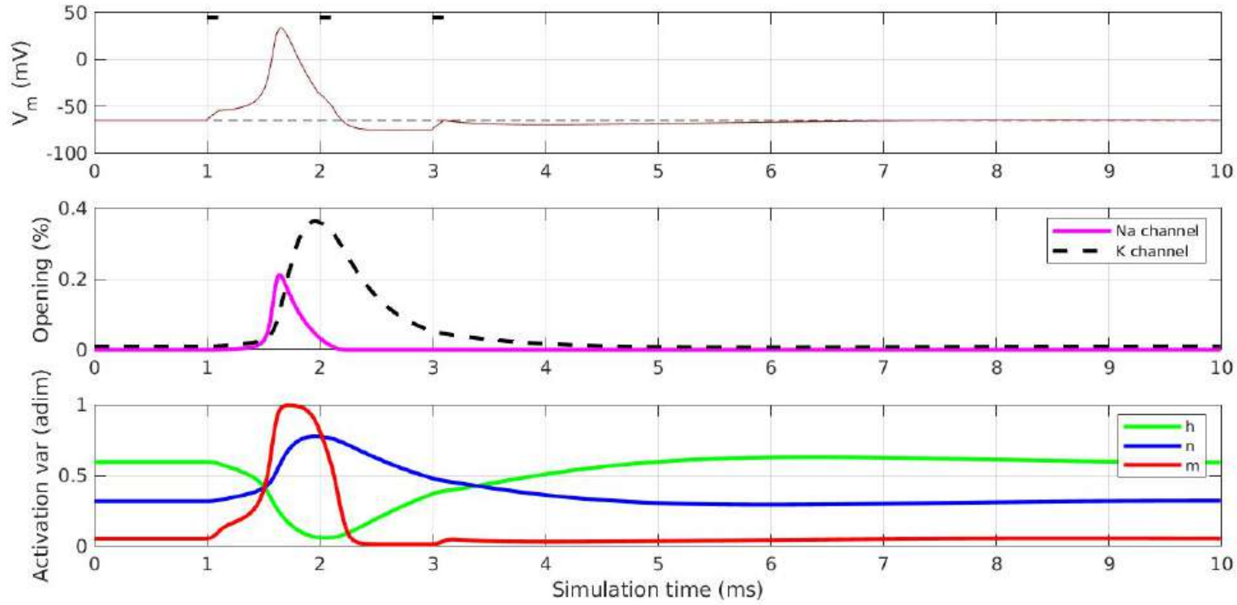
Simulations: Extra insights into the HH model

Injected current pulse of 0.1 ms at t = 1 and 2 ms, I_0=100 uA



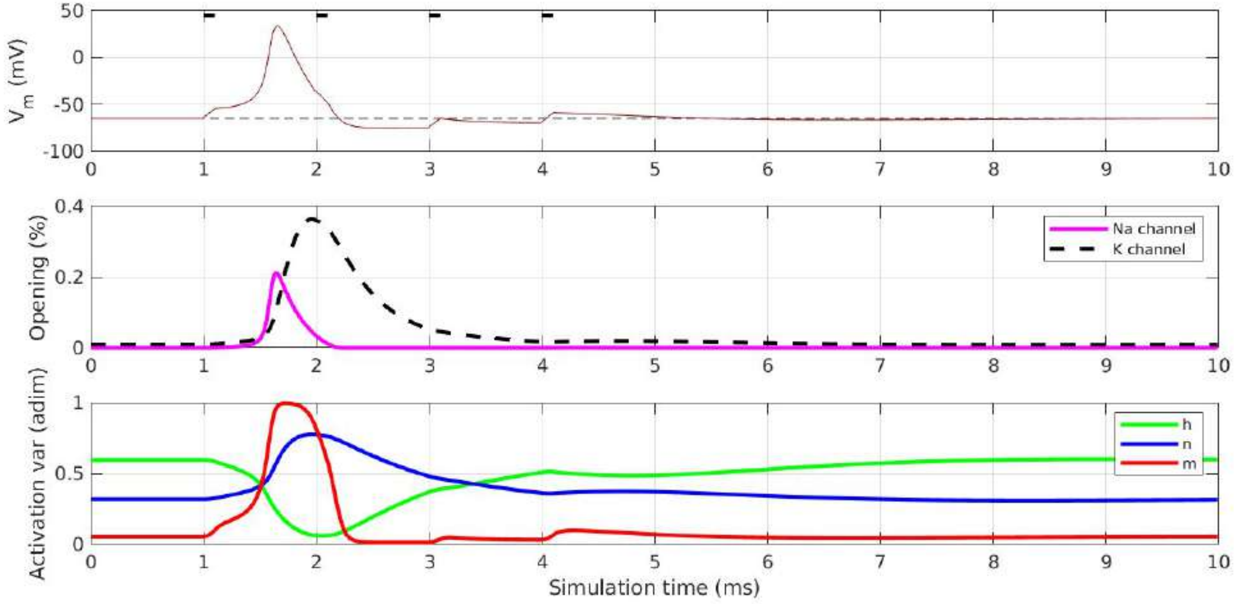
Simulations: Extra insights into the HH model

Injected current pulse of 0.1 ms repeated every 1 ms, I_0=100 uA



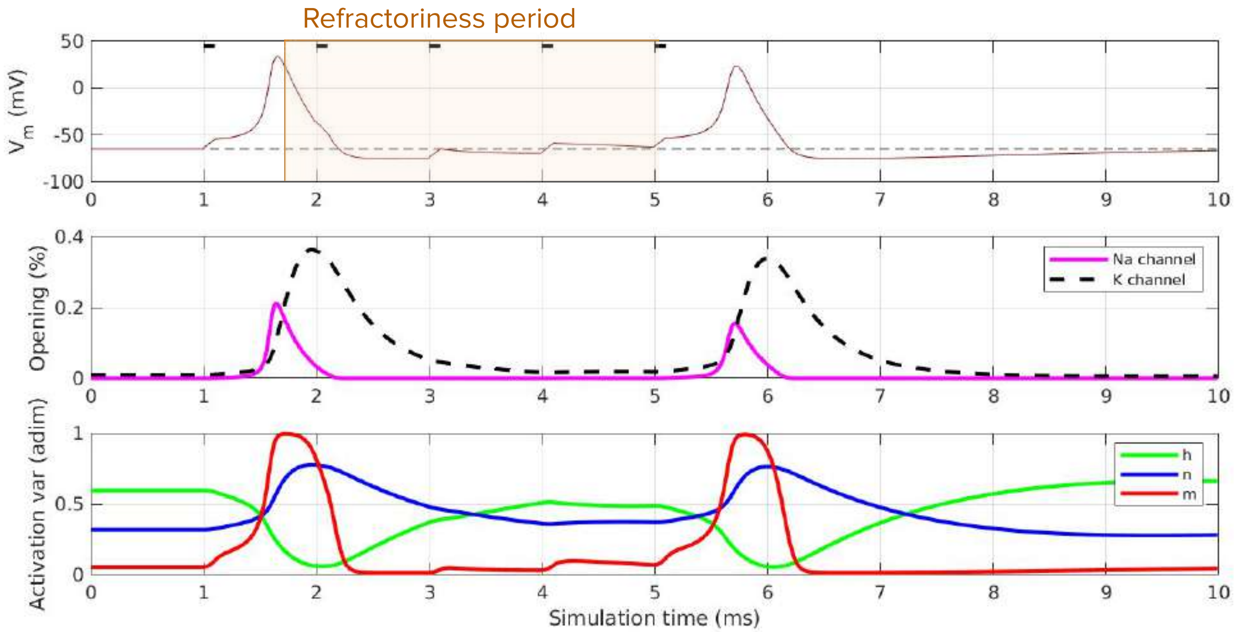
Simulations: Extra insights into the HH model

Injected current pulse of 0.1 ms repeated every 1 ms, I_0=100 uA



Simulations: Extra insights into the HH model

Injected current pulse of 0.1 ms repeated every 1 ms, I_0=100 uA



The refractory period

Just after a spike, it is harder to trigger another one.

Two causes:

- Inactivation of sodium channels (fast):
  - absolute refractory period (impossible to spike)
- Opening of potassium channels (slower):
  - relative refractory period (harder to spike)

The refractory period

The refractory period

Just after a spike, it is harder to trigger another one.

The refractory period

Just after a spike, it is harder to trigger another one.

Two causes:

The refractory period

Just after a spike, it is harder to trigger another one.

Two causes:

- Inactivation of sodium channels (fast):
    - absolute refractory period (impossible to spike)

The refractory period

Just after a spike, it is harder to trigger another one.

Two causes:

- Inactivation of sodium channels (fast):
    - absolute refractory period (impossible to spike)
- Opening of potassium channels (slower):
    - relative refractory period (harder to spike)

The refractory period

Just after a spike, it is harder to trigger another one.

Two causes:

- Inactivation of sodium channels (fast):
  - absolute refractory period (impossible to spike)
- Opening of potassium channels (slower):
  - relative refractory period (harder to spike)

In contrast to the integrate-and-fire models we don’t need to tweak the model to account for refractoriness

The need for stochasticity

The need for stochasticity

Gating isn't deterministic, depicted using m, n, and h

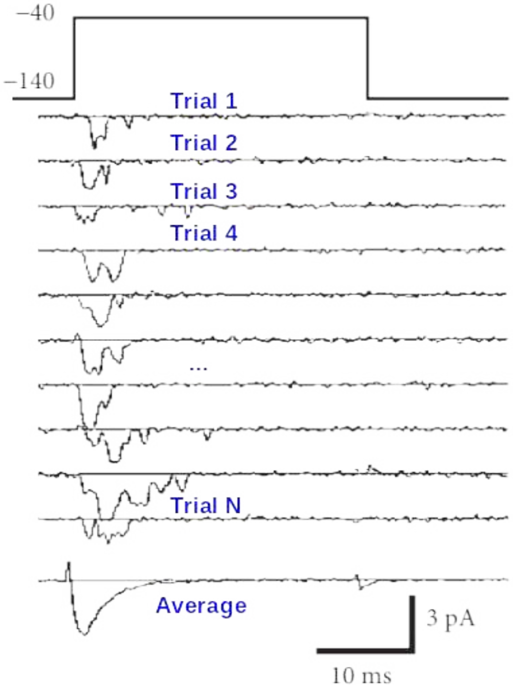


Fig. 2.5 Stochastic channel activation. The current flowing through a small patch of membrane after application of a voltage step (top row) shows step-like changes and is different in each trial (subsequent traces). Averaging over many trials yields the bottom trace. Adapted from Patlak and Ortiz (1985). ©1985 Rockefeller University Press. Originally published in *Journal of General Physiology*, **86**: 89–104.

The need for stochasticity

Gating isn’t deterministic, depicted using m, n, and h

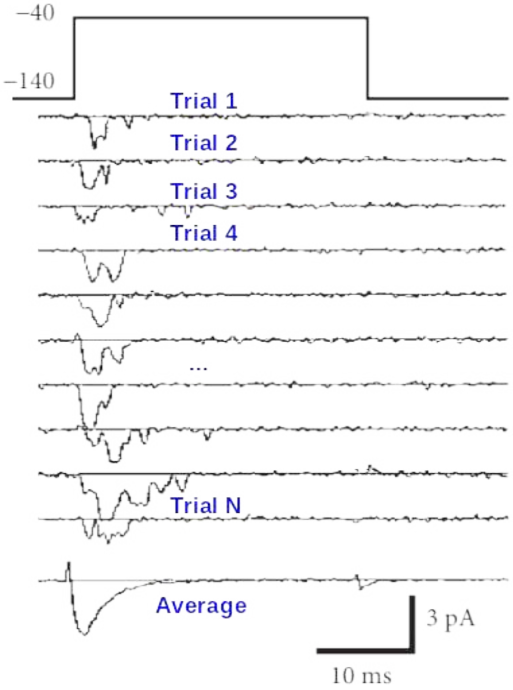


Fig. 2.5 Stochastic channel activation. The current flowing through a small patch of membrane after application of a voltage step (top row) shows step-like changes and is different in each trial (subsequent traces). Averaging over many trials yields the bottom trace. Adapted from Patlak and Ortiz (1985). ©1985 Rockefeller University Press. Originally published in *Journal of General Physiology*, **86**: 89–104.

- Neurons receive input from several presynaptic neurons

The need for stochasticity

Gating isn’t deterministic, depicted using m, n, and h

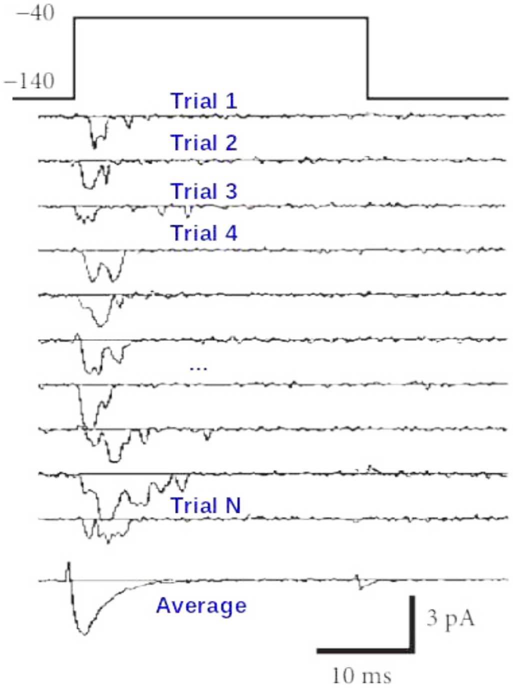


Fig. 2.5 Stochastic channel activation. The current flowing through a small patch of membrane after application of a voltage step (top row) shows step-like changes and is different in each trial (subsequent traces). Averaging over many trials yields the bottom trace. Adapted from Patlak and Ortiz (1985). ©1985 Rockefeller University Press. Originally published in *Journal of General Physiology*, **86**: 89–104.

- ❖ Neurons receive input from several presynaptic neurons
- ❖ Neurons have an spontaneous activity

The need for stochasticity

Gating isn't deterministic, depicted using m, n, and h

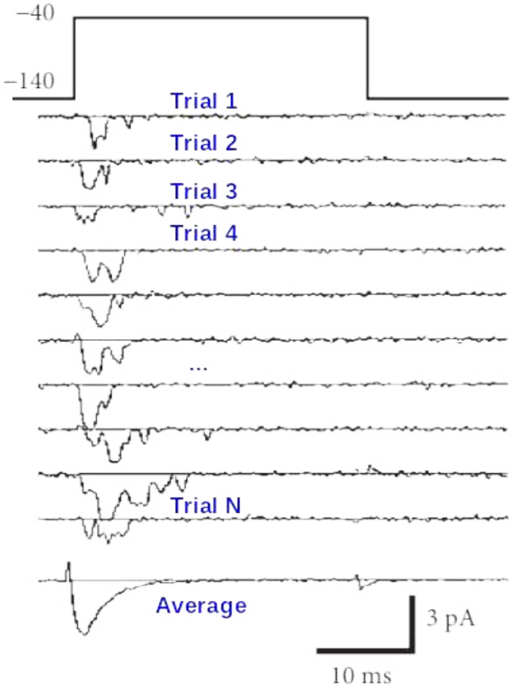


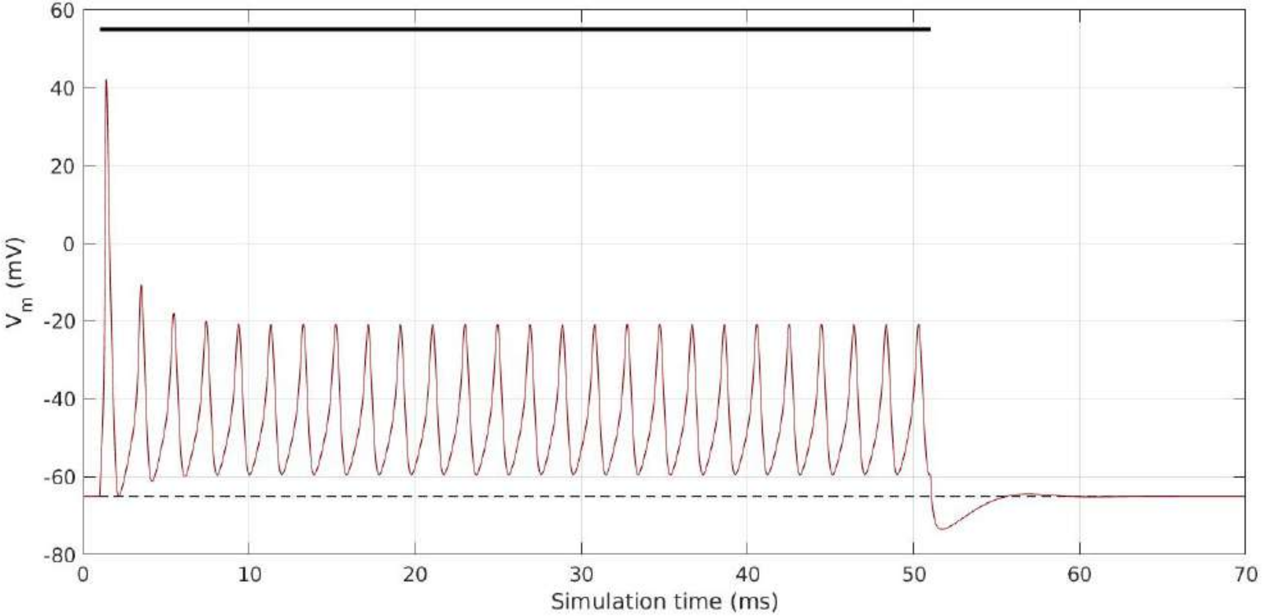
Fig. 2.5 Stochastic channel activation. The current flowing through a small patch of membrane after application of a voltage step (top row) shows step-like changes and is different in each trial (subsequent traces). Averaging over many trials yields the bottom trace. Adapted from Patlak and Ortiz (1985). ©1985 Rockefeller University Press. Originally published in *Journal of General Physiology*, **86**: 89–104.

- ❖ Neurons receive input from several presynaptic neurons
- ❖ Neurons have an spontaneous activity

Let's see how the HH model behaves to a stochastic input current

Response to a long deterministic pulse

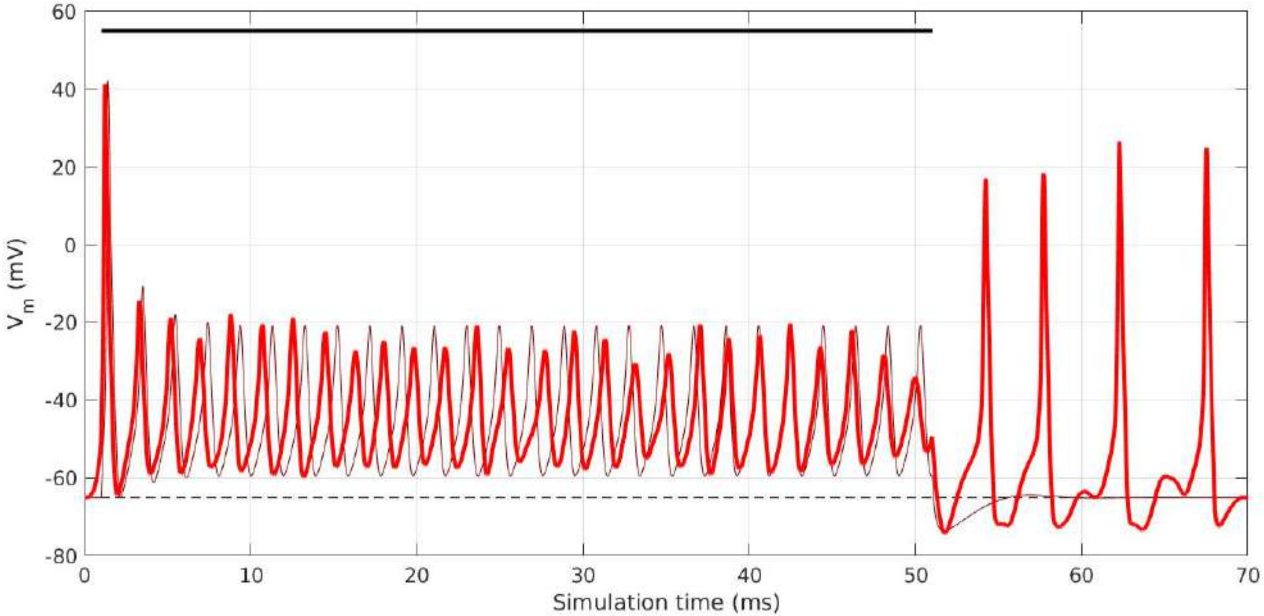
Injected current pulse of 50 ms at t = 1 ms, I_0=100 uA



Response to a noisy long pulse

Injected current pulse of 50 ms at t = 1 ms, I_0=100 uA

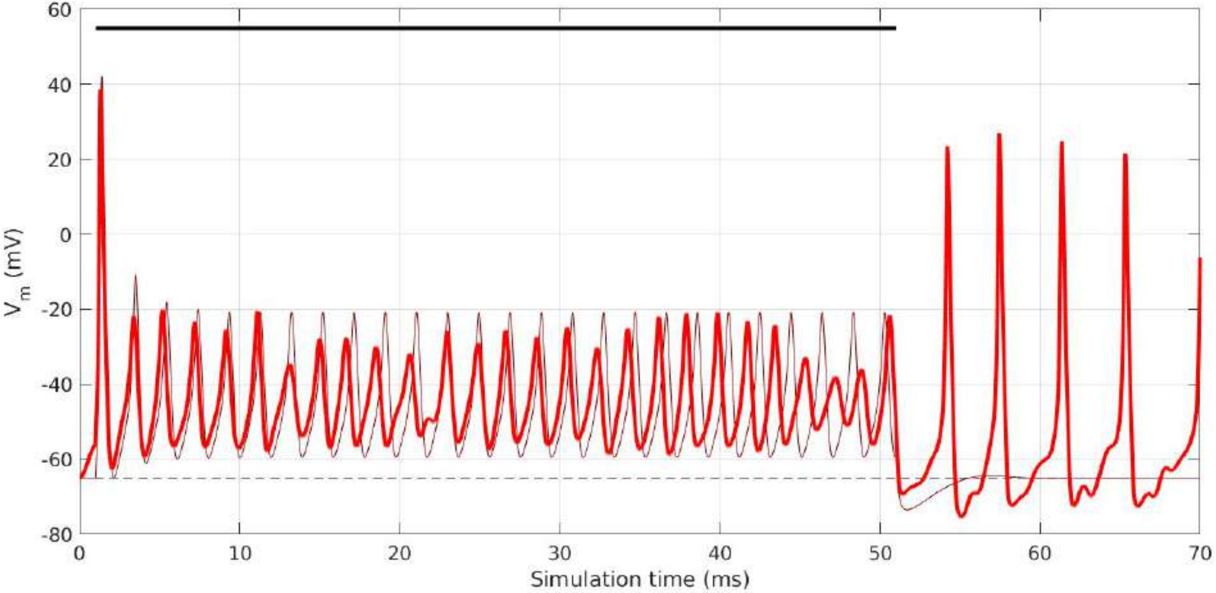
Same pulse as in the deterministic simulation but adding a Gaussian noise with standard deviation of 10 uA



Response to a noisy long pulse («retest»)

Injected current pulse of 50 ms at t = 1 ms, I_0=100 uA

Same pulse as in the deterministic simulation but adding a Gaussian noise with standard deviation of 10 uA



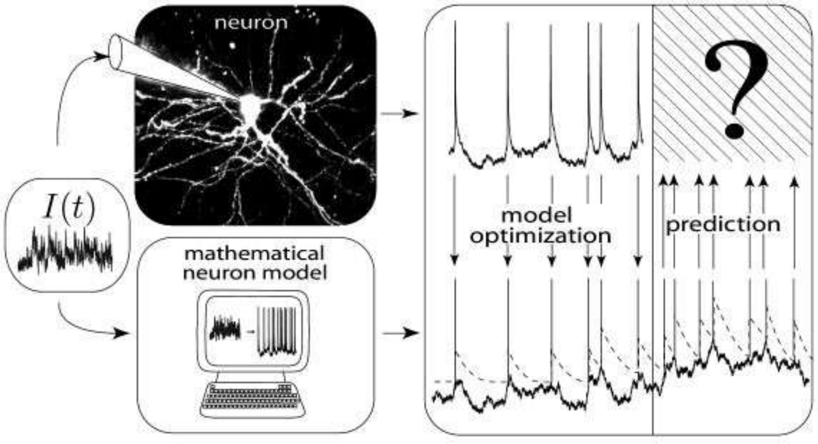
Comparison with data

Comparison with data

Gerstner W, Naud R: How good are neuron models? Science, (2009)

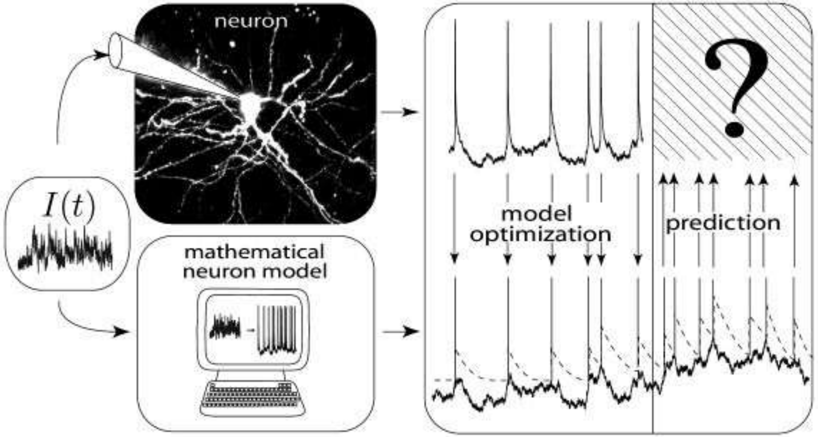
Comparison with data

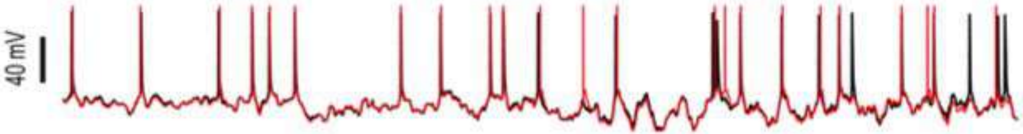
Gerstner W, Naud R: How good are neuron models? Science, (2009)



Comparison with data

Gerstner W, Naud R: How good are neuron models? Science, (2009)





Comparison with data

Comparison with data

Model of Dopaminergic
midbrain neurons

Comparison with data

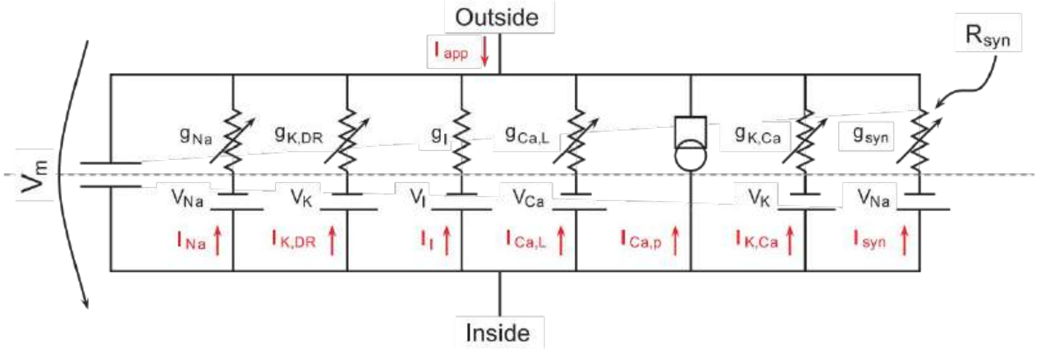
Model of Dopaminergic
midbrain neurons

Drion et al. (2011, PLoS)

Comparison with data

Model of Dopaminergic midbrain neurons

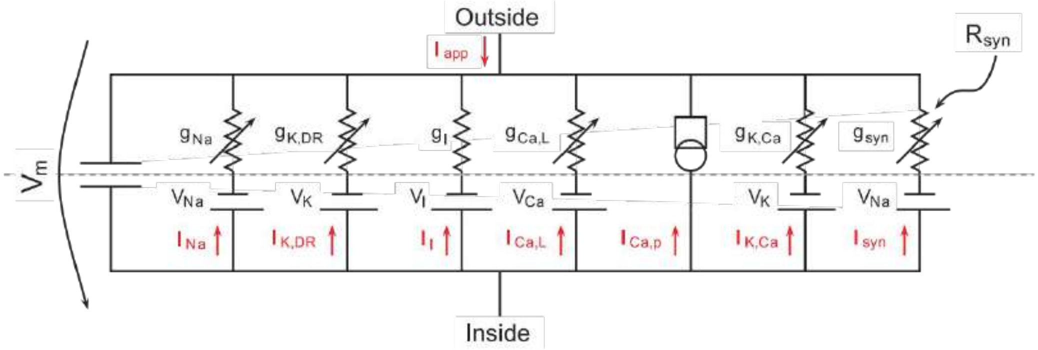
Drion et al. (2011, PLoS)



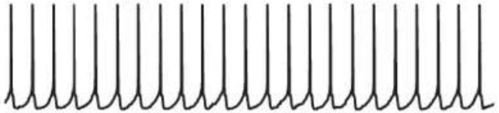
Comparison with data

Model of Dopaminergic midbrain neurons

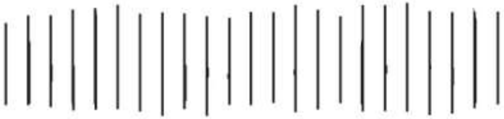
Drion et al. (2011, PLoS)



low synaptic noise



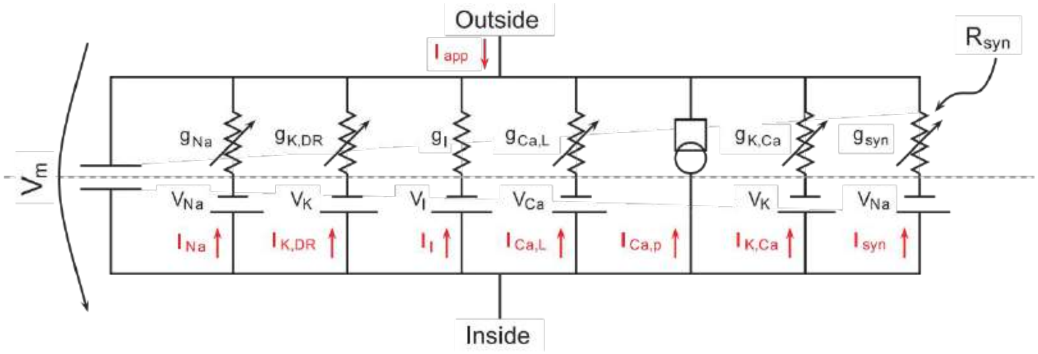
in vitro

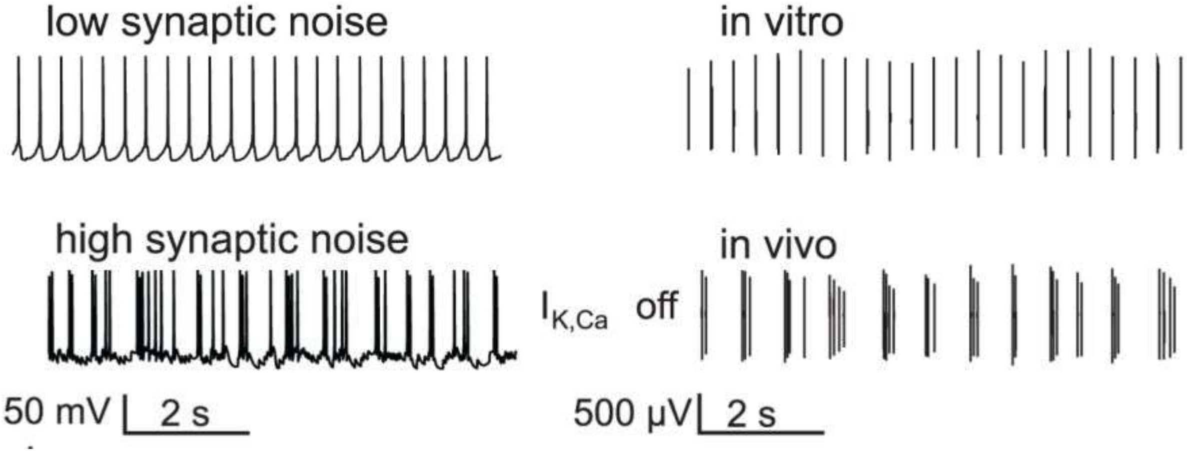


Comparison with data

Model of Dopaminergic midbrain neurons

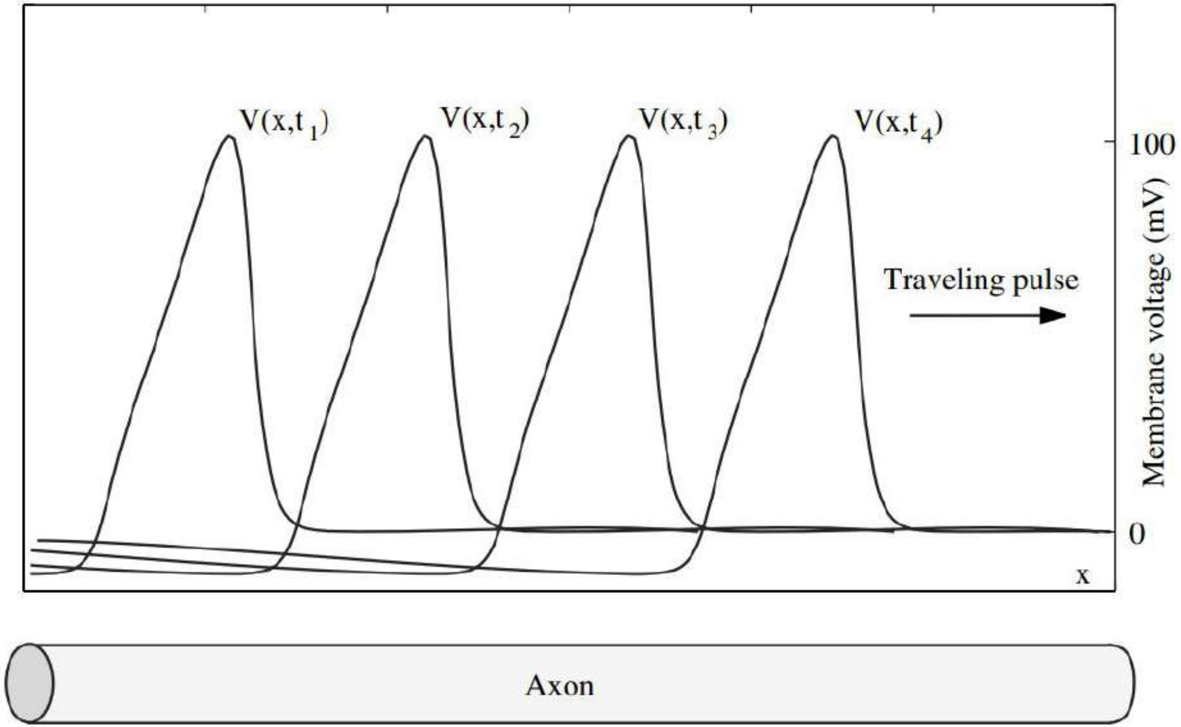
Drion et al. (2011, PLoS)





Propagation of the Action Potentials

Propagation of the Action Potentials



Propagation of the Action Potentials

Propagation of the Action Potentials

1. Excitation in some region ➜ membrane depolarization $V_0 + dV$

Propagation of the Action Potentials

1. Excitation in some region ➔ membrane depolarization $V_0 + dV$
2. Under the potential difference between the region of excitation and the neighboring area, in the axoplasm a current i_a flows

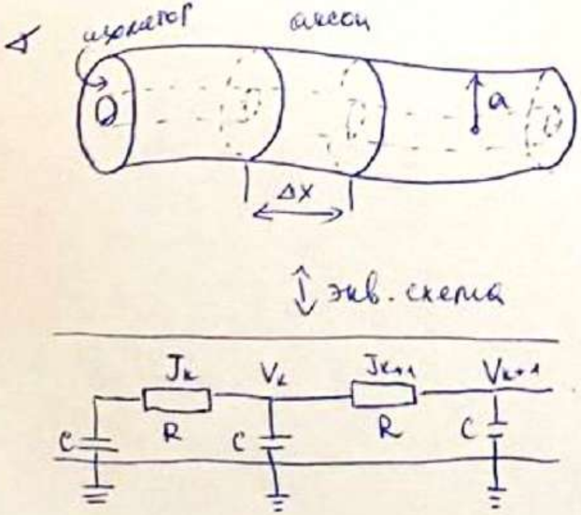
Propagation of the Action Potentials

1. Excitation in some region ➜ membrane depolarization V_0 + dV
2. Under the potential difference between the region of excitation and the neighboring area, in the axoplasm a current i_a flows
3. It leads to decreasing of membrane potential for dV

Propagation of the Action Potentials

1. Excitation in some region ➜ membrane depolarization $V_0 + dV$
2. Under the potential difference between the region of excitation and the neighboring area, in the axoplasm a current i_a flows
3. It leads to decreasing of membrane potential for dV
4. If polarization is enough for threshold ➜ excitation

Propagation of the Action Potentials



∃ ёмкость м/у центр. проводником и обмоткой

∃ сопротивление вдоль центрального проводника

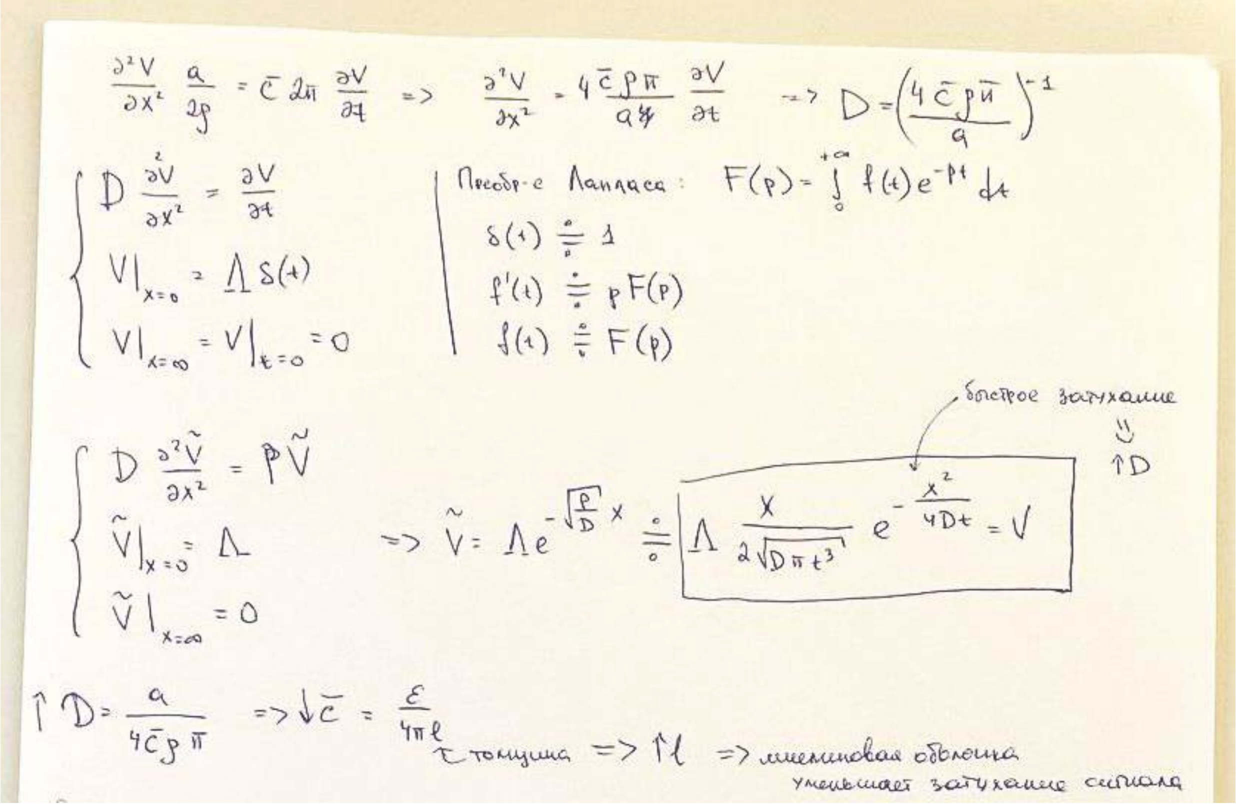
$$1),2)\quad V_{k+1}-V_k = J_{k+1}R = J_{k+1}\underbrace{\frac{\overset{\text{уд-ть}}{\rho}\,dx}{\pi a^2}}_{\text{сопр-е малого уч-ка аксона}} \qquad // R=\rho\frac{\overset{=dx}{l}}{\underset{=\pi a^2}{S}}$$

Плотность тока: $j_{k+1} = \dfrac{J_{k+1}}{\underbrace{2\pi a}_{\text{длина окружн-ти аксона}}}$

$$V_{k+1}-V_k = j_{k+1}\rho\frac{2dx}{a} \Rightarrow \boxed{\frac{\partial V}{\partial x} = \frac{2\rho}{a}j}$$

$$J_{k+1}-J_k = C\dot{V}_k = \underset{\uparrow\text{уд. ёмкость}}{\bar{c}}\,dx\cdot 2\pi a\,\dot{V} \Rightarrow \boxed{\frac{\partial j}{\partial x} = \bar{c}\,2\pi\dot{V}}$$

Propagation of the Action Potentials



$$\frac{\partial^2 V}{\partial x^2}\frac{a}{2\rho} = \bar{C}\,2\pi\,\frac{\partial V}{\partial t} \Rightarrow \frac{\partial^2 V}{\partial x^2} = \frac{4\bar{C}\rho\pi}{a\cancel{4}}\frac{\partial V}{\partial t} \Rightarrow D = \left(\frac{4\bar{C}\rho\pi}{a}\right)^{-1}$$

$$\begin{cases} D\dfrac{\partial^2 V}{\partial x^2} = \dfrac{\partial V}{\partial t} \\ V\big|_{x=0} = \Lambda\,\delta(t) \\ V\big|_{x=\infty} = V\big|_{t=0} = 0 \end{cases}$$

Преобр-е Лапласа: $F(p) = \int_0^{+\infty} f(t)e^{-pt}\,dt$

$\delta(t) \overset{\cdot}{\underset{\cdot}{=}} 1$

$f'(t) \overset{\cdot}{\underset{\cdot}{=}} pF(p)$

$f(t) \overset{\cdot}{\underset{\cdot}{=}} F(p)$

быстрое затухание

$\Downarrow$

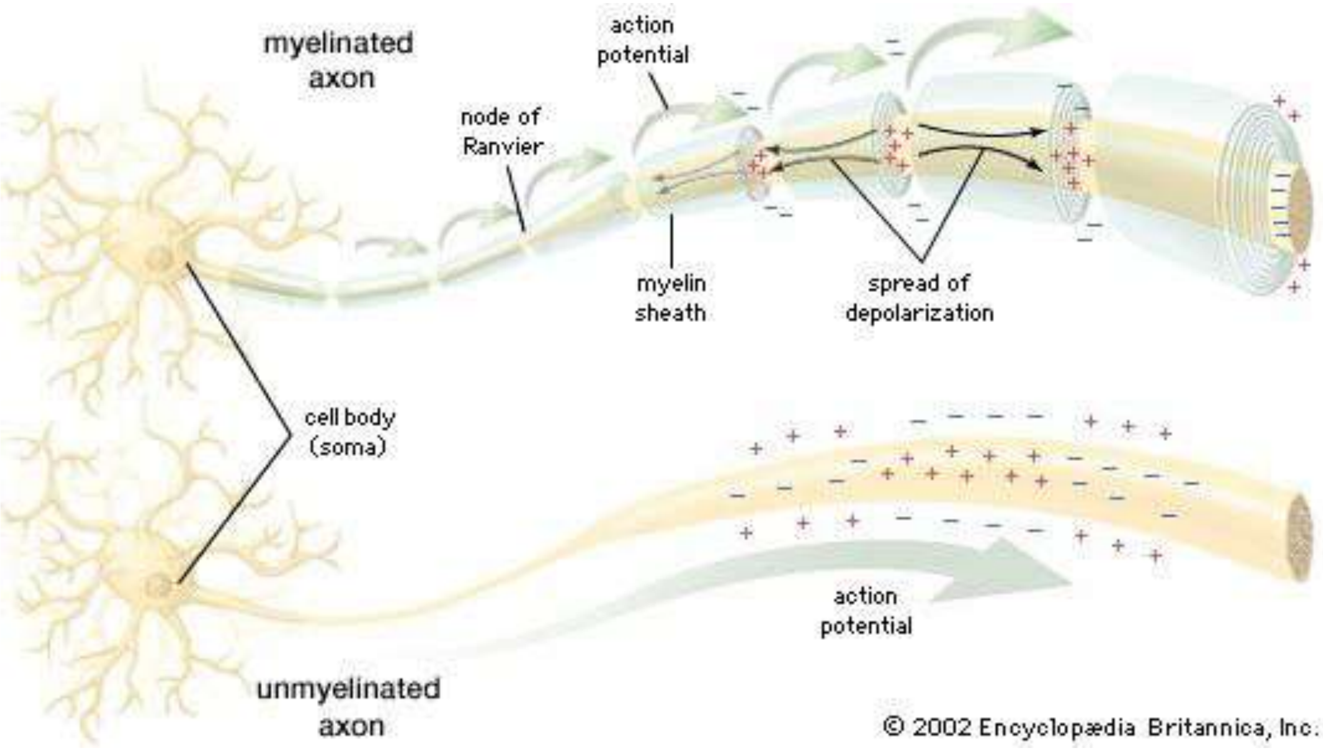
$\uparrow D$

$$\begin{cases} D\dfrac{\partial^2 \tilde{V}}{\partial x^2} = p\tilde{V} \\ \tilde{V}\big|_{x=0} = \Lambda \\ \tilde{V}\big|_{x=\infty} = 0 \end{cases} \Rightarrow \tilde{V} = \Lambda e^{-\sqrt{\frac{p}{D}}x} \overset{\cdot}{\underset{\cdot}{=}} \boxed{\Lambda\frac{x}{2\sqrt{D\pi t^3}}e^{-\frac{x^2}{4Dt}} = V}$$

$$\uparrow D = \frac{a}{4\bar{C}\rho\pi} \Rightarrow \downarrow\bar{C} = \frac{\varepsilon}{4\pi l}$$

l толщина $\Rightarrow \uparrow l \Rightarrow$ миелиновая оболочка уменьшает затухание сигнала

Propagation of the Action Potentials



References

Bean (2007). The action potential in mammalian central neurons. Nat. Rev. Neurosci, 8, 451-465.

Benna, M. & Fusi, S. (2016) Computational principles of synaptic memory consolidation. Nat. Neurosci., 19, 1697-1706.

Dayan, P, & Abbott, L. (2005). Theoretical neuroscience: computational and mathematical modeling of neural systems. MIT press.

Drion G., Massotte L., Sepulchre R., & Seutin V. (2011). How modeling can reconcile apparently discrepant experimental results: The case of pacemaking in dopaminergic neurons. PLoS Computational Biology 7: e1002050. doi: 10.1371/journal.pcbi.1002050

Gerstner, W., & Naud, R. (2009). How good are neuron models? Science, 326, 379–380. doi:10.1126/science.1181936

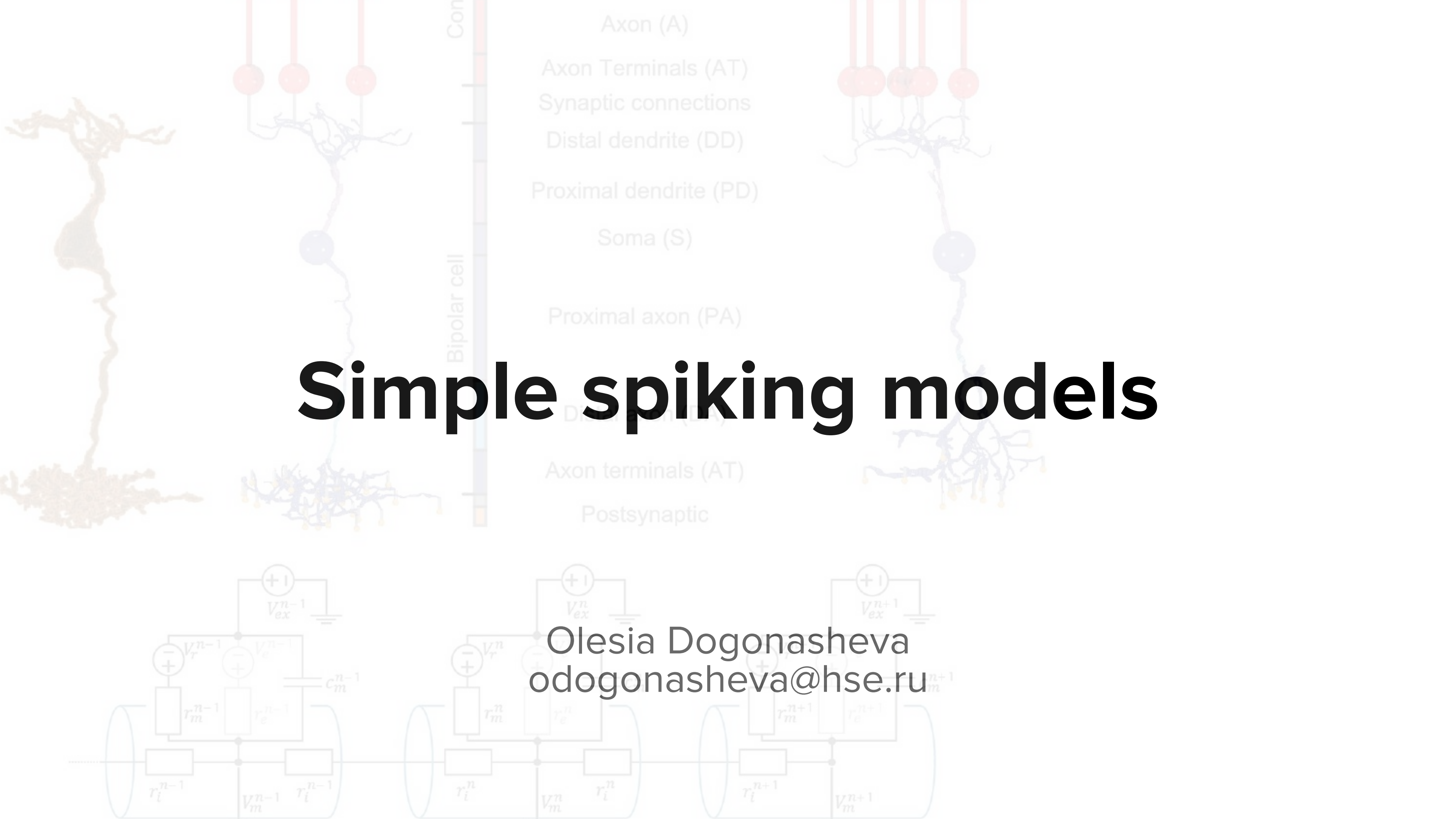
Hodgkin, A., & Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol., 117, 500–44.

Patlak, J. & Ortiz, M. (1985) Slow currents through single sodium channels of the adult rat heart. J. Gen. Phys., 86:89–104.

Patolsky F., Timko B., Yu G., Fang Y., Greytak A., Zheng G., & Lieber C. (2006). Detection, stimulation, and inhibition of neuronal signals with high-density nanowire transistor arrays. Science, 313:1100-4. doi: 10.1126/science.1128640.

Parekh, R., Ascoli, G. (2013). Neuronal morphology goes digital: A research hub for cellular and system neuroscience. Neuron, 77. 10.1016/j.neuron.2013.03.008

Thank you!



Simple spiking models

Olesia Dogonasheva
odogonasheva@hse.ru

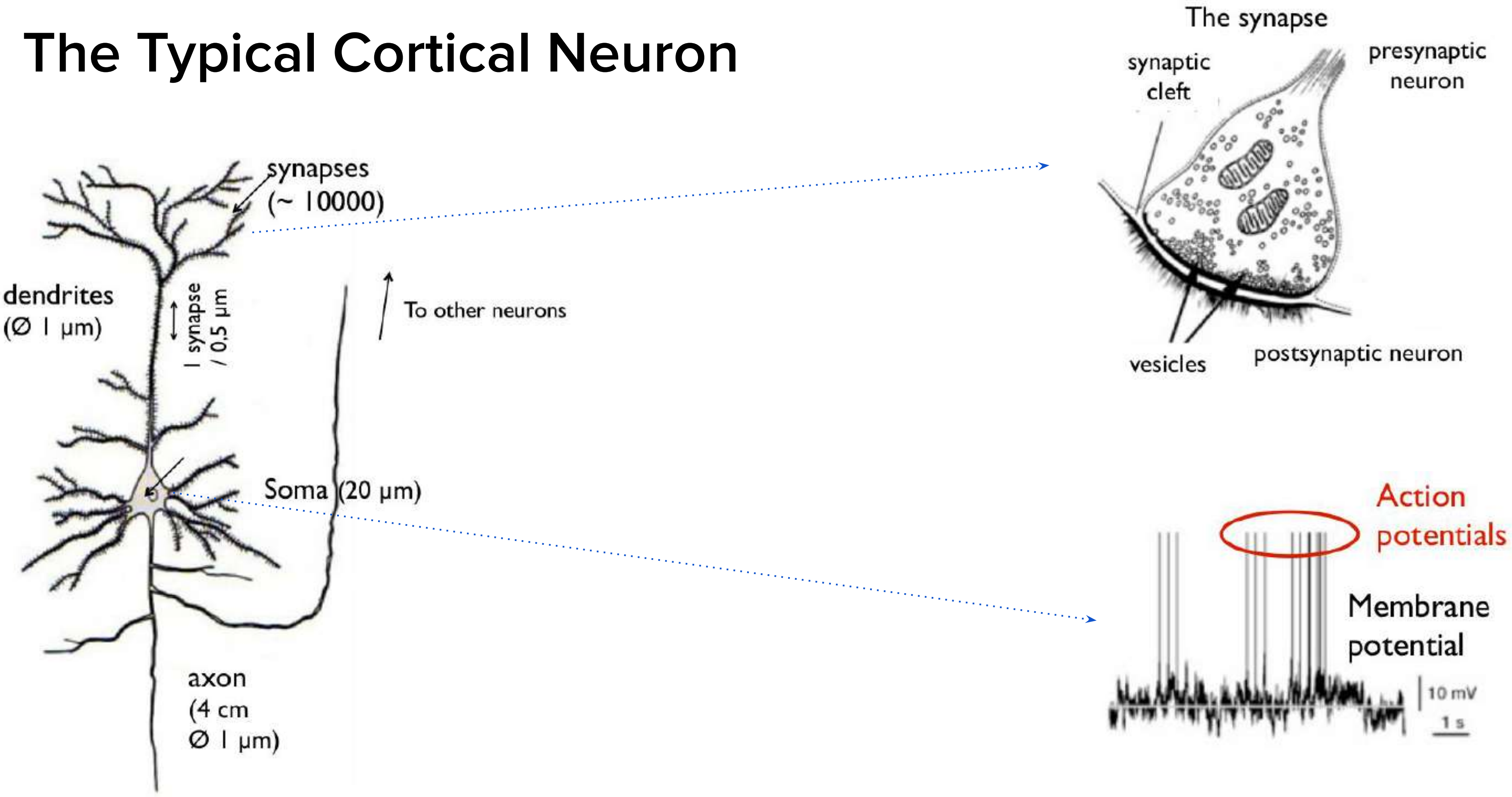
Outline

1. Code for Hopfield neural network
2. Code for Hodgkin-Huxley model
3. Modification of HH model
    a. Addition of different ion channels
    b. Example: Morris-Lecar model
4. Simple spiking models
    a. LIF
    b. QIF
    c. EIF
    d. GIF
5. Synapses
    a. Classification
    b. Simple model

Open *p.86* in

lecture_2022-11-14.pdf

The Typical Cortical Neuron



HH model

$$C\dot{V} = I - \overbrace{\bar{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}})}^{I_{\mathrm{K}}} - \overbrace{\bar{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}})}^{I_{\mathrm{Na}}} - \overbrace{g_{\mathrm{L}} (V - E_{\mathrm{L}})}^{I_{\mathrm{L}}}$$

$$\begin{aligned} \dot{n} &= (n_\infty(V) - n)/\tau_n(V)\,, \\ \dot{m} &= (m_\infty(V) - m)/\tau_m(V)\,, \\ \dot{h} &= (h_\infty(V) - h)/\tau_h(V)\,, \end{aligned}$$

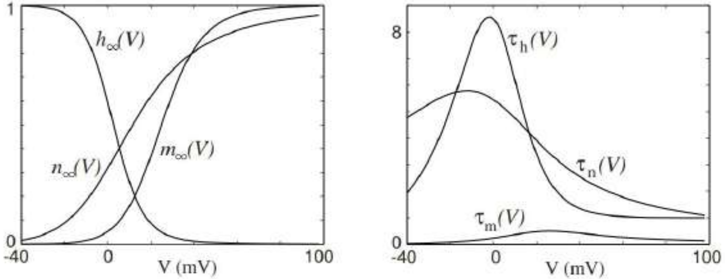
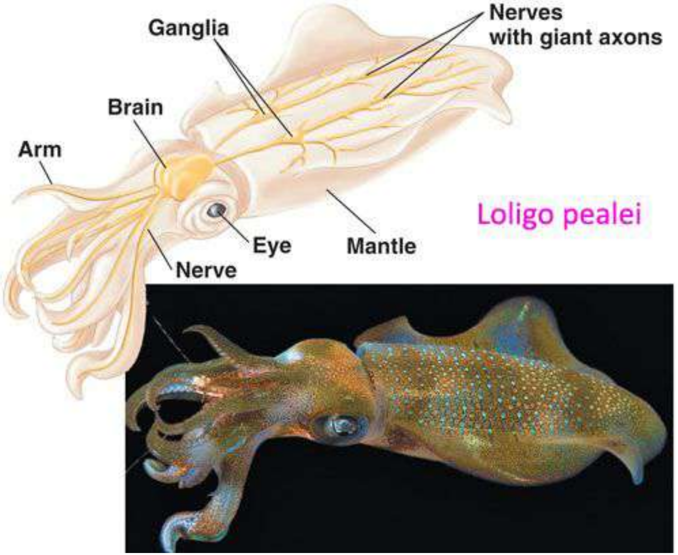


Figure 2.13: Steady-state (in)activation functions (left) and voltage-dependent time constants (right) in the Hodgkin-Huxley model.

Nobel prize 1963

A.L. Hodgkin

A. Huxley



Loligo pealei

Code for HH model

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_{k} I_k\left(t\right) + I\left(t\right)$$

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_{k} I_k\left(t\right) + I\left(t\right)$$

$$I_k\left(t\right) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k}\, h^{q_k}\, \left(u - E_k\right)$$

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_{k} I_k(t) + I(t)$$

$$I_k(t) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k}\, h^{q_k}\ (u - E_k)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_{k} I_k\left(t\right) + I\left(t\right)$$

$$I_k\left(t\right) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k}\, h^{q_k}\, \left(u - E_k\right)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

It is possible to classify an ion channel using

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_k I_k(t) + I(t)$$

$$I_k(t) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k} h^{q_k} \; (u - E_k)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

It is possible to classify an ion channel using

- its genetic sequence;

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_k I_k(t) + I(t)$$

$$I_k(t) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k}\, h^{q_k}\; (u - E_k)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

It is possible to classify an ion channel using

- its genetic sequence;
- the ion type (sodium, potassium, calcium ...) that can pass through the open channel;

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_k I_k(t) + I(t)$$

$$I_k(t) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k} h^{q_k} \; (u - E_k)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

It is possible to classify an ion channel using

- its genetic sequence;
- the ion type (sodium, potassium, calcium ...) that can pass through the open channel;
- its voltage dependence;

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_k I_k(t) + I(t)$$

$$I_k(t) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k}\, h^{q_k}\, \left(u - E_k\right)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

It is possible to classify an ion channel using

- its genetic sequence;
- the ion type (sodium, potassium, calcium ...) that can pass through the open channel;
- its voltage dependence;
- its sensitivity to second-messengers such as intra-cellular calcium;

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_{k} I_k\left(t\right) + I\left(t\right)$$

$$I_k\left(t\right) = g_k\left(\left[Ca^{++}\right], \ldots\right) m^{p_k}\, h^{q_k}\ \left(u - E_k\right)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

It is possible to classify an ion channel using

- its genetic sequence;
- the ion type (sodium, potassium, calcium ...) that can pass through the open channel;
- its voltage dependence;
- its sensitivity to second-messengers such as intra-cellular calcium;
- its presumed functional role;

Framework for biophysical neuron models

In fact, we have more than just sodium and potassium channels:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_k I_k\,(t) + I\,(t)$$

$$I_k\,(t) = g_k\left(\left[Ca^{++}\right],\ldots\right) m^{p_k}\, h^{q_k}\ \left(u - E_k\right)$$

m and h describe activation and inactivation of the channel,
p_k and q_k are empirical parameters,
E_k is the reversal potential,
g_k is the maximum conductance which may depend on secondary variables such as the concentration of calcium, magnesium, dopamine or other substances

It is possible to classify an ion channel using

- its genetic sequence;
- the ion type (sodium, potassium, calcium ...) that can pass through the open channel;
- its voltage dependence;
- its sensitivity to second-messengers such as intra-cellular calcium;
- its presumed functional role;
- its response to pharmacological drugs or to neuromodulators such as acetylcholine and dopamine.

Morris-Lecar model

Morris-Lecar model

Catherine Morris

Harold Lecar

Morris-Lecar model

Catherine Morris

Harold Lecar

Balanus nubilus

Morris-Lecar model

$$\begin{cases} C\dot{V} &= I - g_K w(V - E_K) - g_{Ca} m_{inf}(V - E_{Ca}) - g_L(V - E_L), \\ \dot{w} &= \lambda_w(V)(w_{inf}(V) - w), \end{cases}$$

Catherine Morris

Harold Lecar

Balanus nubilus

Morris-Lecar model

$$\begin{cases} C\dot{V} & = I - g_K w(V - E_K) - g_{Ca} m_{inf}(V - E_{Ca}) - g_L(V - E_L), \\ \dot{w} & = \lambda_w(V)(w_{inf}(V) - w), \end{cases}$$

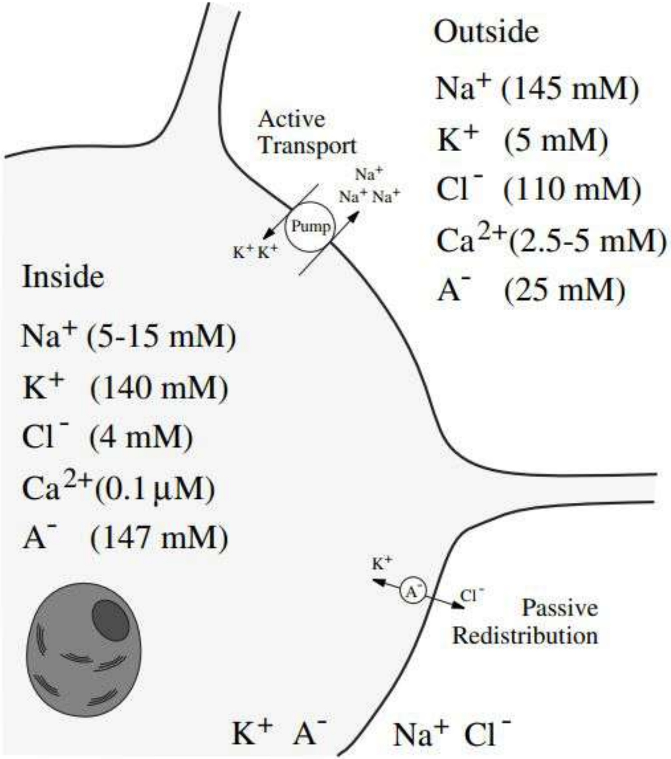
Catherine Morris

Harold Lecar

Balanus nubilus

Morris, C. and Lecar, H. (1981) Voltage oscillations in the barnacle giant muscle fiber. Biophys. J. 35: 193 - 213.

Main ionic currents through neuron membranes



Equilibrium Potentials

$$
\begin{array}{lrcl}
\mathrm{Na}^{+} & 62\log\frac{145}{5} & = & 90\ \mathrm{mV} \\
 & 62\log\frac{145}{15} & = & 61\ \mathrm{mV} \\
\mathrm{K}^{+} & 62\log\frac{5}{140} & = & -90\ \mathrm{mV} \\
\mathrm{Cl}^{-} & -62\log\frac{110}{4} & = & -89\ \mathrm{mV} \\
\mathrm{Ca}^{2+} & 31\log\frac{2.5}{10^{-4}} & = & 136\ \mathrm{mV} \\
 & 31\log\frac{5}{10^{-4}} & = & 146\ \mathrm{mV}
\end{array}
$$

Morris-Lecar model

$$\begin{cases} C\dot{V} &= I - g_K w(V - E_K) - g_{Ca} m_{inf}(V - E_{Ca}) - g_L(V - E_L), \\ \dot{w} &= \lambda_w(V)(w_{inf}(V) - w), \end{cases}$$

Catherine Morris

Harold Lecar

Balanus nubilus

Morris, C. and Lecar, H. (1981) Voltage oscillations in the barnacle giant muscle fiber. Biophys. J. 35: 193 - 213.

Morris-Lecar model

$$\begin{cases} C\dot{V} & = I - g_K w(V - E_K) - g_{Ca} m_{inf}(V - E_{Ca}) - g_L(V - E_L), \\ \dot{w} & = \lambda_w(V)(w_{inf}(V) - w), \end{cases}$$

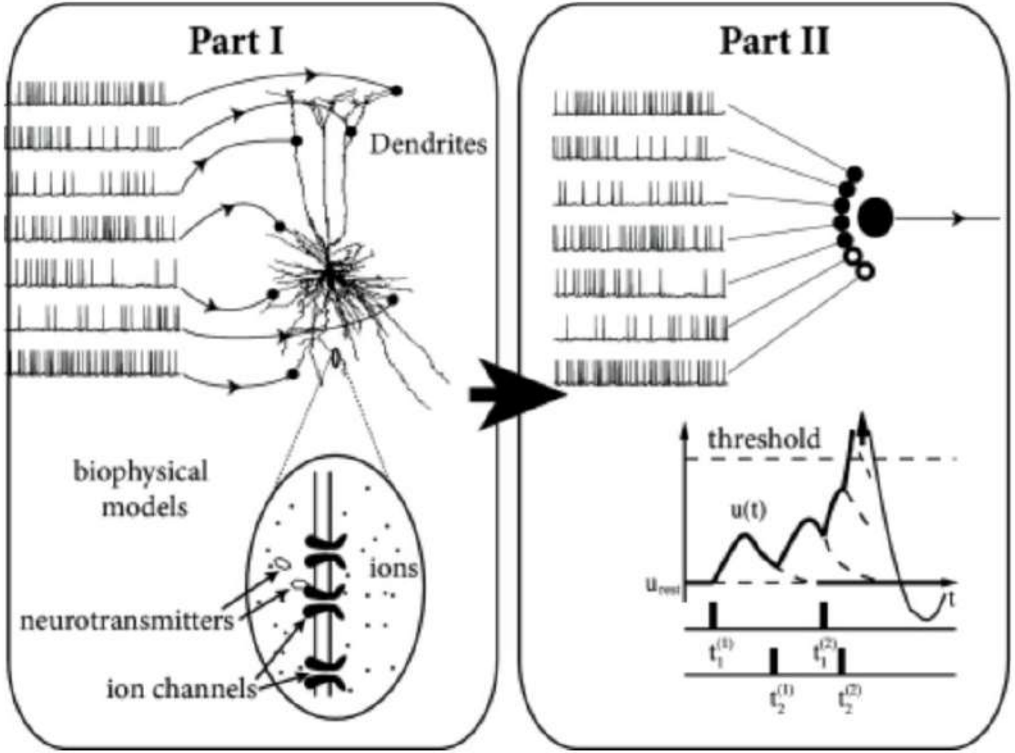
- The two-dimensional model can be analyzed using phase-plane methods
- Morris–Lecar neurons exhibit both class I and class II of excitability
- The Morris-Lecar equations are particularly useful for modelling fast-spiking neurons, such as the pyramidal neurons of the neocortex
- A model employing Morris-Lecar oscillators of different frequencies has been used to explain quite complex bursting phenomena of coupled neurons

Catherine Morris

Harold Lecar

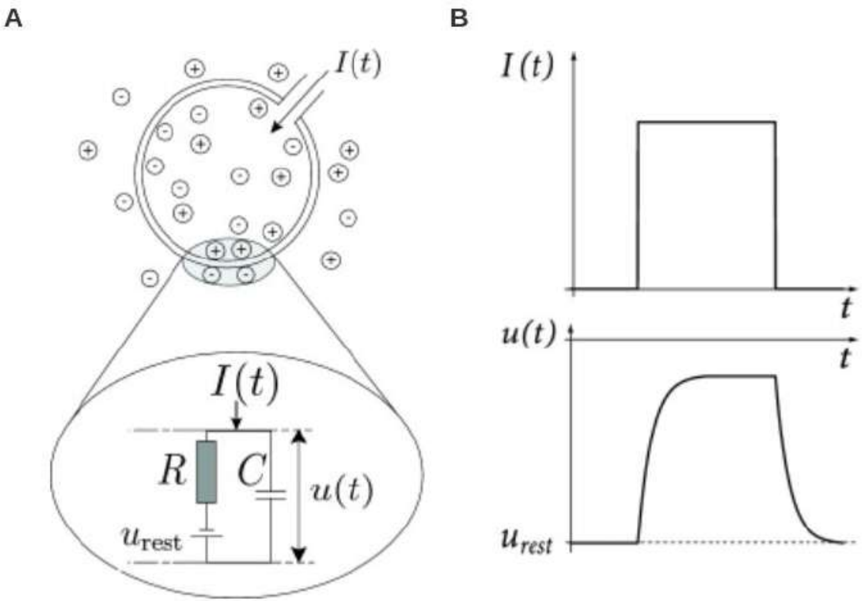
Balanus nubilus

Morris, C. and Lecar, H. (1981) Voltage oscillations in the barnacle giant muscle fiber. Biophys. J. 35: 193 - 213.


Part I
Dendrites
biophysical models
ions
neurotransmitters
ion channels
Part II
threshold
u(t)
u_{rest}
t
$t_1^{(1)}$
$t_1^{(2)}$
$t_2^{(1)}$
$t_2^{(2)}$

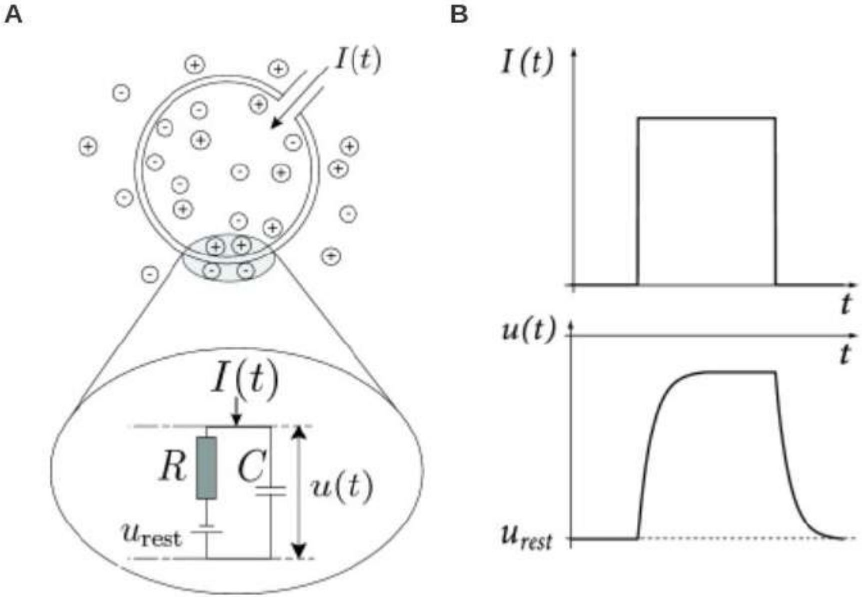
Leaky integrate-and-fire model

Leaky integrate-and-fire model



Leaky integrate-and-fire model

- u is the instantaneous membrane potential
- u_{rest} the resting potential (in the absence of any input)
- I(t) is an injected current
- u(t) –> u_{rest}



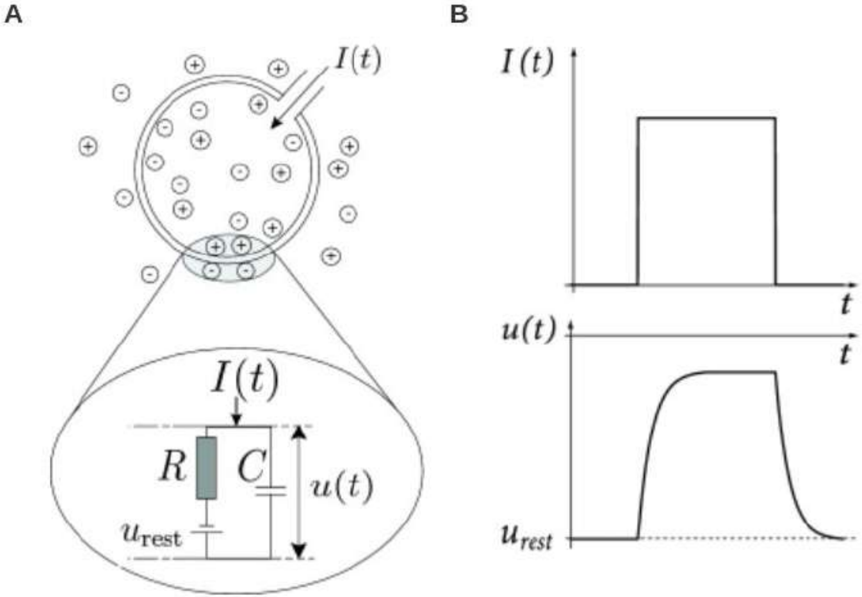
Leaky integrate-and-fire model

- u is the instantaneous membrane potential
- u_{rest} the resting potential (in the absence of any input)
- I(t) is an injected current
- u(t) –> u_{rest}

A neuron is surrounded by a cell membrane, which is a rather **good insulator**. If a short current pulse I(t) is injected into the neuron, the additional electrical charge

$$q=\int I(t)dt$$

will charge the cell membrane.



Leaky integrate-and-fire model

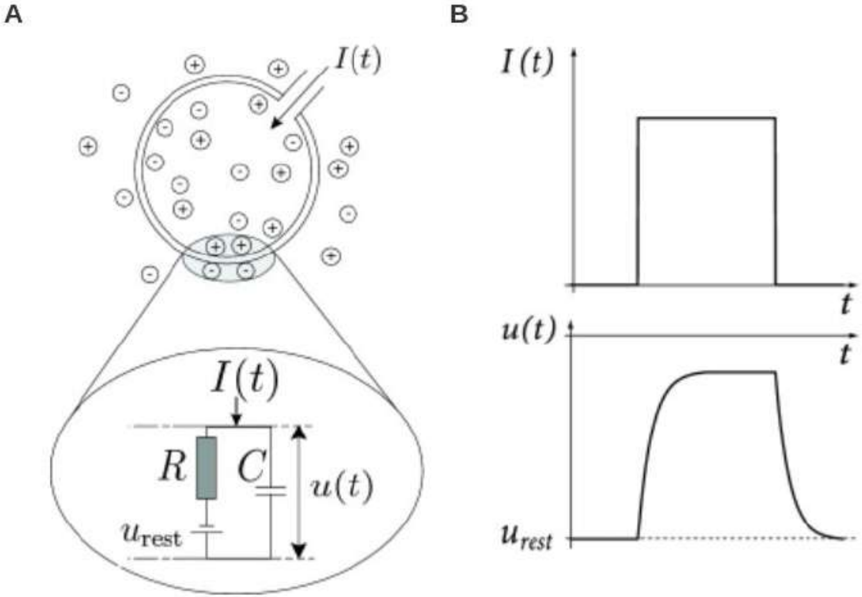
- u is the instantaneous membrane potential
- u_{rest} the resting potential (in the absence of any input)
- I(t) is an injected current
- u(t) –> u_{rest}

A neuron is surrounded by a cell membrane, which is a rather **good insulator**. If a short current pulse I(t) is injected into the neuron, the additional electrical charge

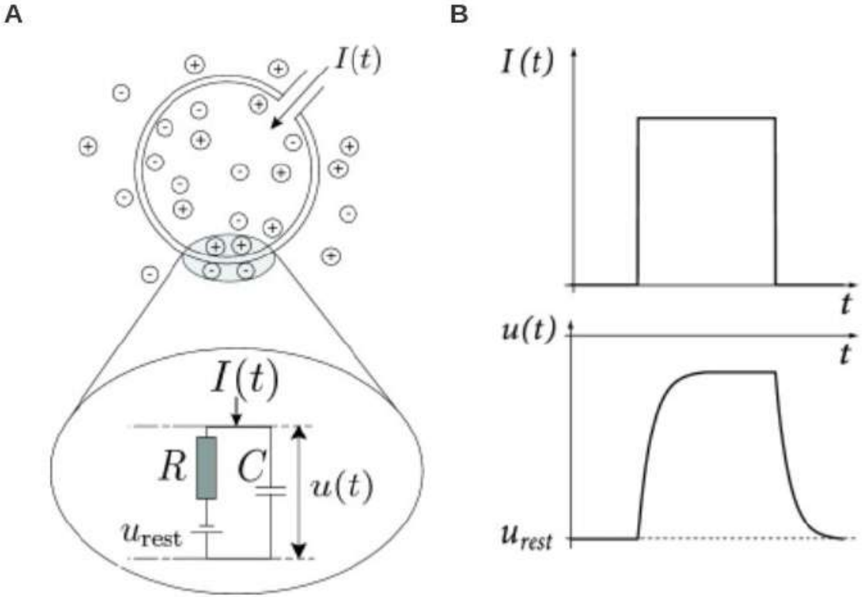
$$q=\int I(t)dt$$

will charge the cell membrane.

The cell membrane acts like a **capacitor**. This insulator is not perfect, the charge will, over time, slowly **leak** through the cell membrane. The cell membrane can therefore be characterized by a finite leak resistance R.

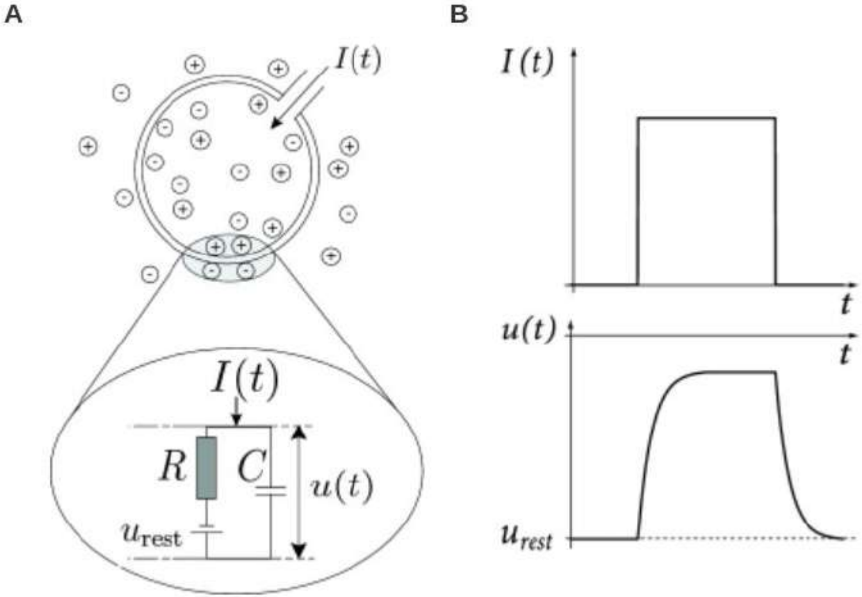


Leaky integrate-and-fire model



Leaky integrate-and-fire model

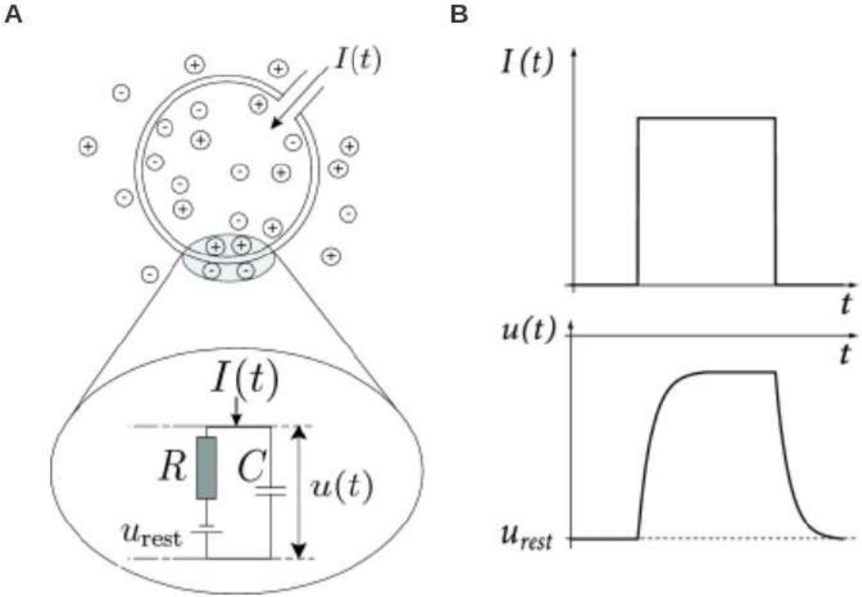
The simplest electrical circuit consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I(t)*:



Leaky integrate-and-fire model

The simplest electrical circuit consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I(t)*:

$$I\left(t\right) = I_R + I_C$$

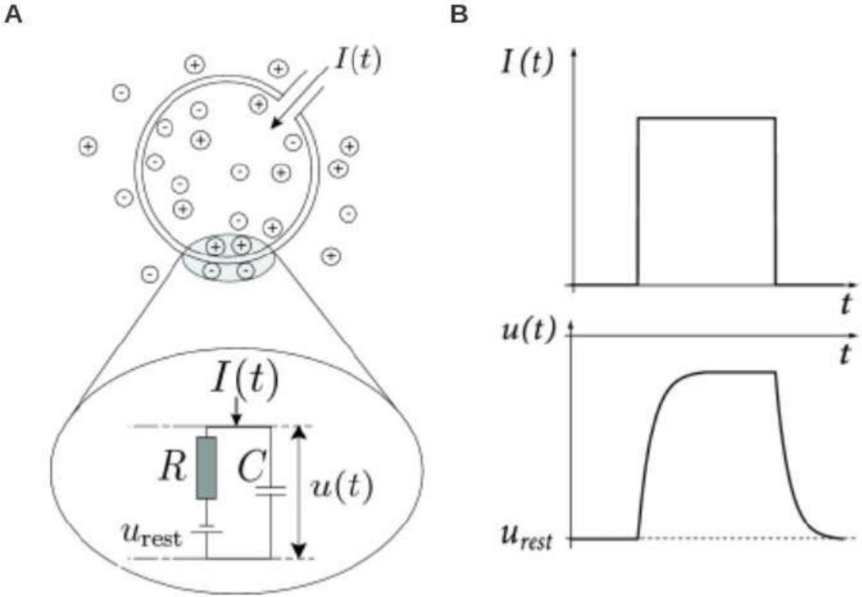


Leaky integrate-and-fire model

The simplest electrical circuit consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I(t)*:

$$I(t) = \boxed{I_R} + I_C$$

the resistive current which passes through the linear resistor R. Ohm's law as $I_R = u_R/R$ where $u_R = u - u_{rest}$ is the voltage across the resistor



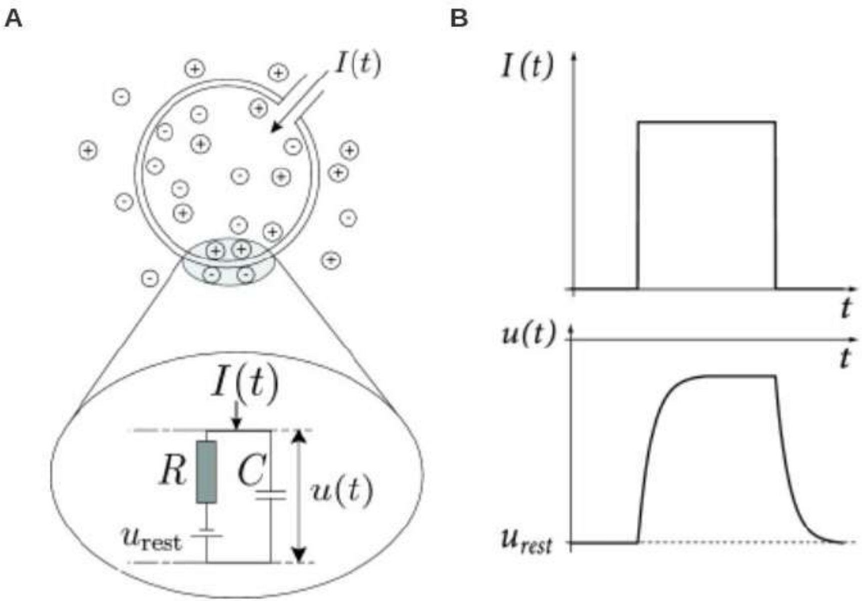
Leaky integrate-and-fire model

The simplest electrical circuit consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I(t)*:

$$I\left(t\right) = \boxed{I_R} + \boxed{I_C}$$

the resistive current which passes through the linear resistor R. Ohm's law as $I_R = u_R / R$ where $u_R = u - u_{rest}$ is the voltage across the resistor

charges the capacitor C. From the definition C = q / u a capacitive current $I_C = dq / dt = C\,du / dt$.



Leaky integrate-and-fire model

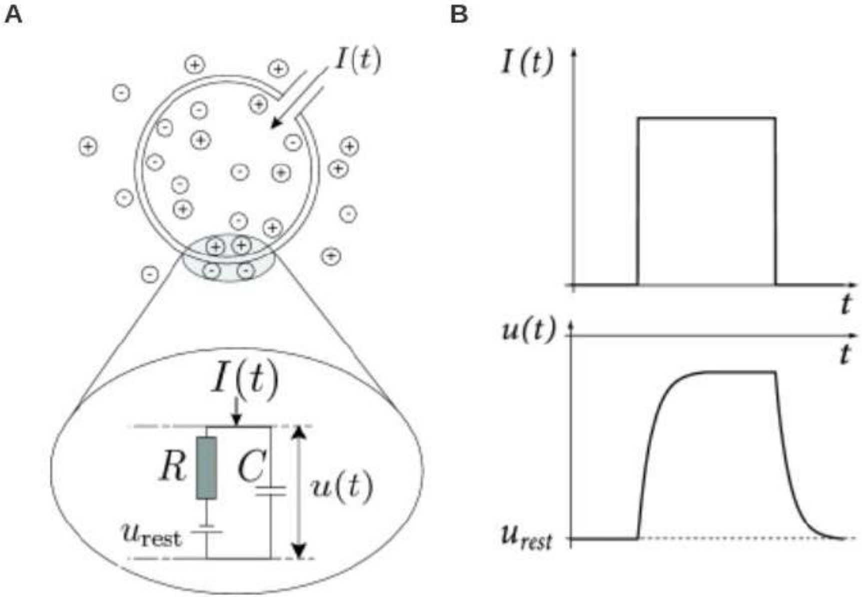
The simplest electrical circuit consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I(t)*:

$$I\left(t\right) = \boxed{I_R} + \boxed{I_C}$$

the resistive current which passes through the linear resistor R. Ohm’s law as I_R=u_R/R where u_R = u − u_{rest} is the voltage across the resistor

charges the capacitor C. From the definition C = q / u a capacitive current I_C = dq / dt = C du / dt.

$$I\left(t\right) = \frac{u\left(t\right) - u_{\mathrm{rest}}}{R} + C\,\frac{\mathrm{d}u}{\mathrm{d}t}$$



Leaky integrate-and-fire model

The simplest electrical circuit consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I(t)*:

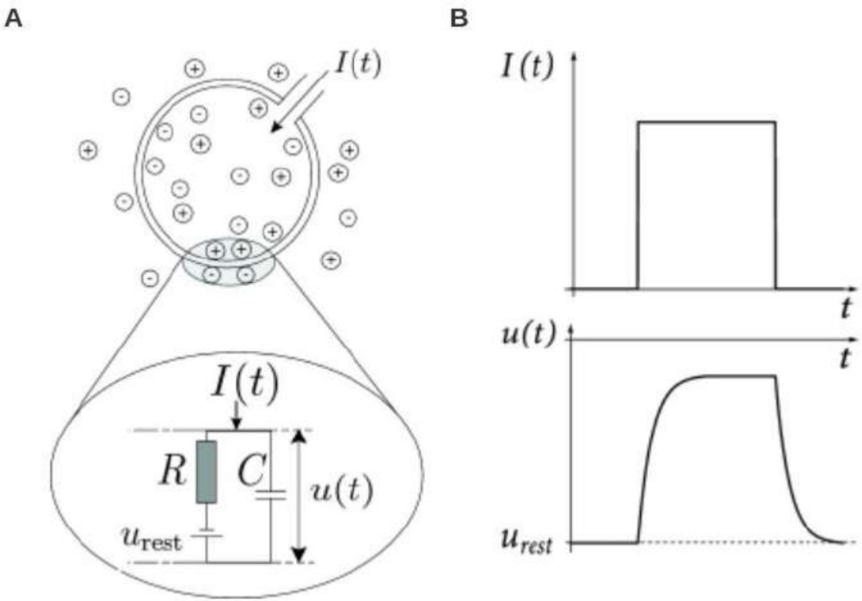
$$I\left(t\right) = \boxed{I_R} + \boxed{I_C}$$

the resistive current which passes through the linear resistor R. Ohm's law as $I_R=u_R/R$ where $u_R = u - u_{rest}$ is the voltage across the resistor

charges the capacitor C. From the definition C = q / u a capacitive current $I_C = dq / dt = C\ du / dt$.

$$I\left(t\right) = \frac{u\left(t\right) - u_{\mathrm{rest}}}{R} + C\,\frac{\mathrm{d}u}{\mathrm{d}t}$$

$$\tau_m\,\frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\,I\left(t\right)$$



Leaky integrate-and-fire model

The simplest electrical circuit consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I(t)*:

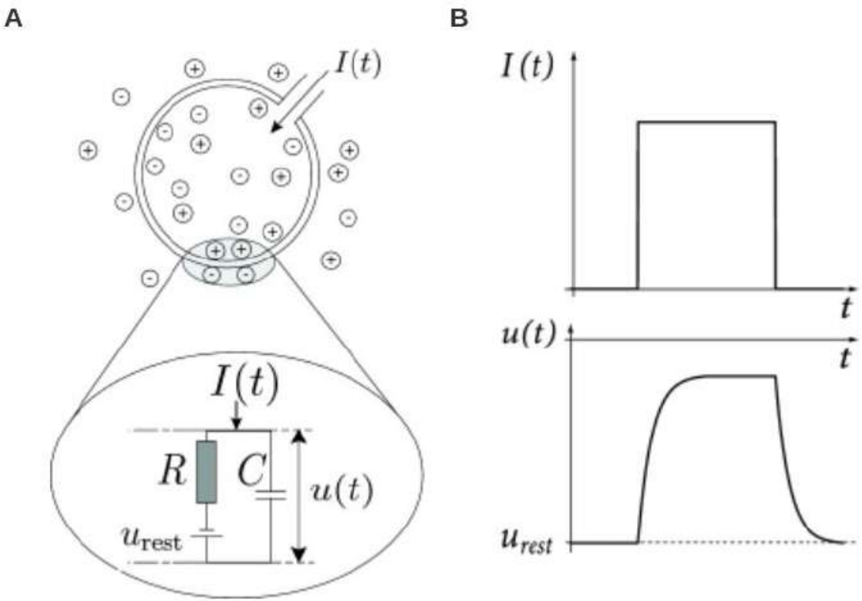
$$I(t) = \boxed{I_R} + \boxed{I_C}$$

the resistive current which passes through the linear resistor R. Ohm's law as $I_R = u_R / R$ where $u_R = u - u_{rest}$ is the voltage across the resistor

charges the capacitor C. From the definition C = q / u a capacitive current $I_C = dq / dt = C\, du / dt$.

$$I(t) = \frac{u(t) - u_{\text{rest}}}{R} + C \frac{\mathrm{d}u}{\mathrm{d}t}$$

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\text{rest}}\right] + R\, I(t)$$



u is the membrane potential
τ_m is the membrane time constant
R is the resistance
I(t) is an external current

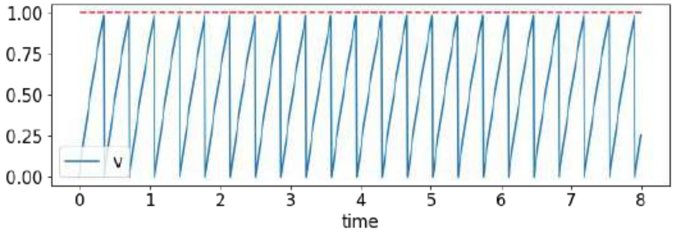
Code for LIF

Limitations of the Leaky Integrate-and-Fire Model

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\, I\left(t\right)$$

Limitations of the Leaky Integrate-and-Fire Model

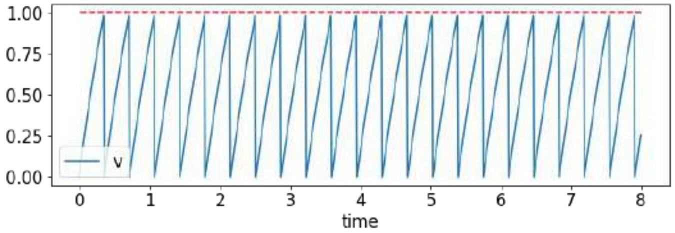
$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\text{rest}}\right] + R\,I(t)$$



Limitations of the Leaky Integrate-and-Fire Model

1. Adaptation, Bursting, and Inhibitory Rebound

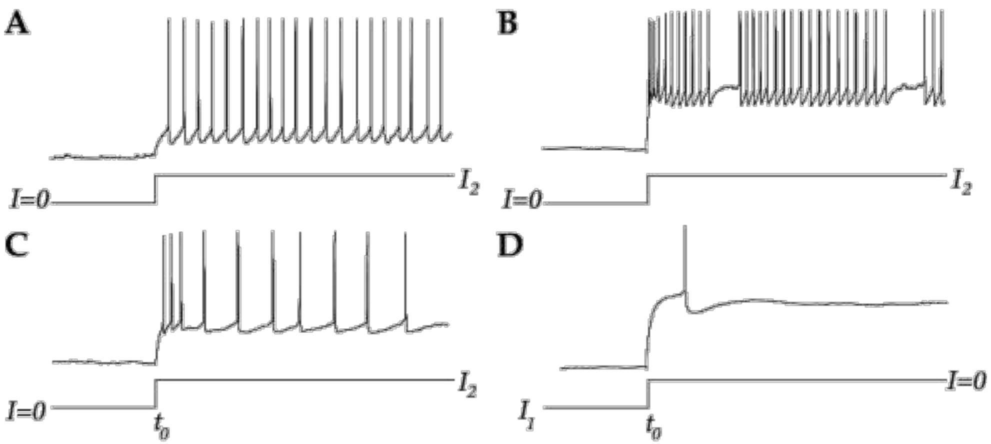
$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\,I\left(t\right)$$

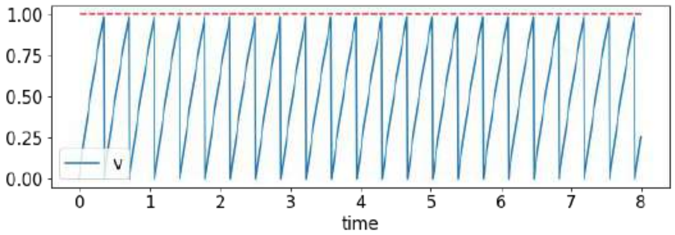


Limitations of the Leaky Integrate-and-Fire Model

1. Adaptation, Bursting, and Inhibitory Rebound

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\mathrm{rest}}\right] + R\,I(t)$$

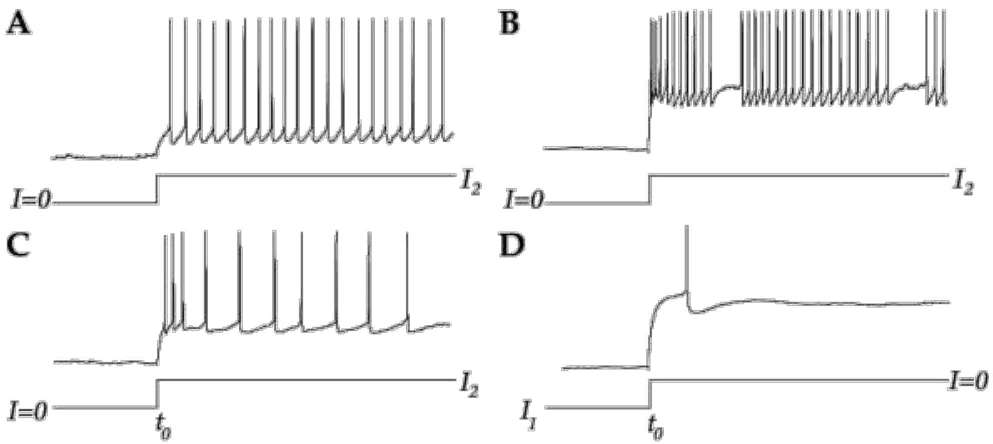




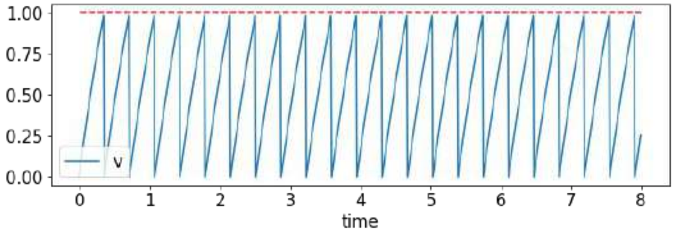
Limitations of the Leaky Integrate-and-Fire Model

1. Adaptation, Bursting, and Inhibitory Rebound

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\text{rest}}\right] + R\,I(t)$$



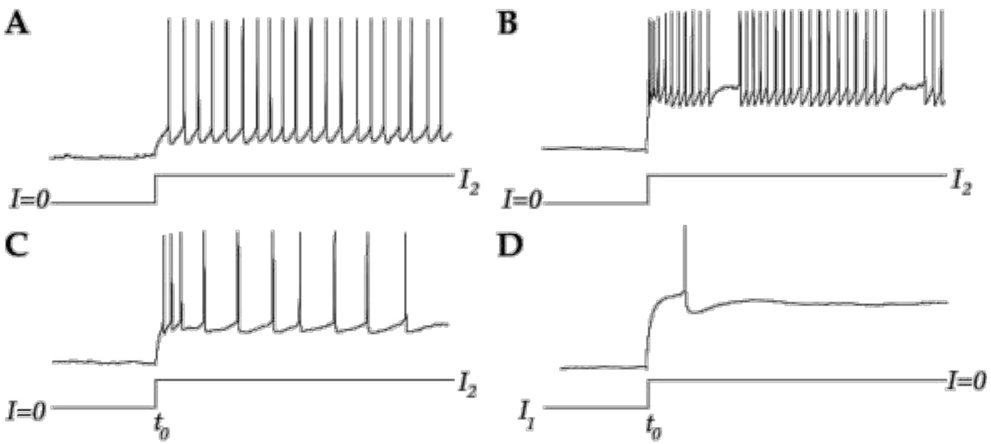
Response to a current step. In **A** - **C**, the current is switched on at t=t_0 to a value I_2>0. Fast-spiking neurons (**A**) have short interspike intervals without adaptation while regular-spiking neurons (**C**) exhibit adaptation, visible as an increase in the duration of interspike intervals. An example of a stuttering neuron is shown in **B**. Many neurons emit an inhibitory rebound spike (**D**) after an inhibitory current I_1< 0 is switched off. Data [1-2]

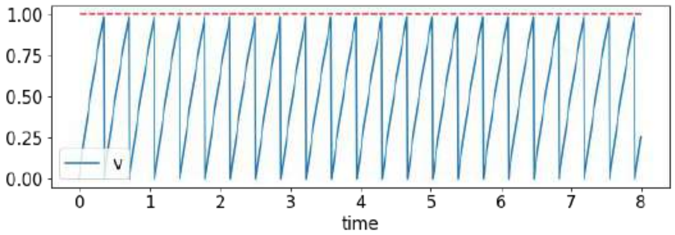


Limitations of the Leaky Integrate-and-Fire Model

1. Adaptation, Bursting, and Inhibitory Rebound

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\text{rest}}\right] + R\,I(t)$$





Response to a current step. In **A** - **C**, the current is switched on at $t=t_0$ to a value $I_2>0$. Fast-spiking neurons (**A**) have short interspike intervals without adaptation while regular-spiking neurons (**C**) exhibit adaptation, visible as an increase in the duration of interspike intervals. An example of a stuttering neuron is shown in **B**. Many neurons emit an inhibitory rebound spike (**D**) after an inhibitory current $I_1< 0$ is switched off. Data [1-2]

[1] H. Markram, M. Toledo-Rodrgiguez, Y. Wang, A. Gupta, G. Silberberg and C. Wu (2004) *Interneurons of the neocortical inhibitory system. Nature Review Neuroscienc* **5**, pp. 793–807.

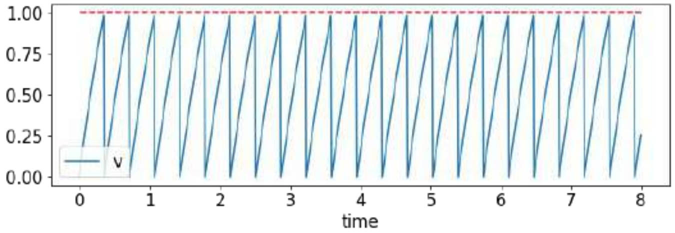
[2] M. Toledo-Rodriguez, B. Blumenfeld, C. Wu, J. Luo, B. Attali, P. Goodman and H. Markram (2004) *Correlation maps allow neuronal electrical properties to be predicted from single-cell gene expression profiles in rat neocortex. Cerebral Cortex* **14**, pp. 1310–1327.

Limitations of the Leaky Integrate-and-Fire Model

2. Conductance Changes after a Spike:

The shape of the postsynaptic potentials does not only depend on the level of depolarization but, more generally, on the internal state of the neuron, e.g., on the timing relative to previous action potentials.

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\,I\left(t\right)$$

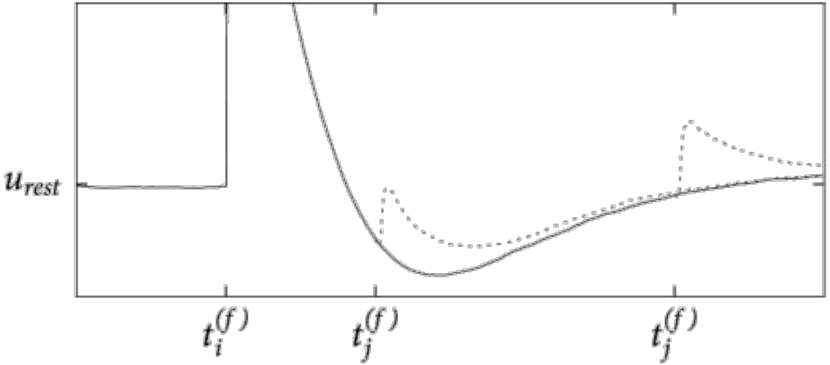


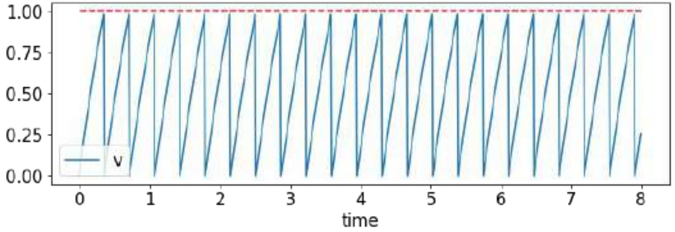
Limitations of the Leaky Integrate-and-Fire Model

2. Conductance Changes after a Spike:

The shape of the postsynaptic potentials does not only depend on the level of depolarization but, more generally, on the internal state of the neuron, e.g., on the timing relative to previous action potentials.

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\,I\left(t\right)$$



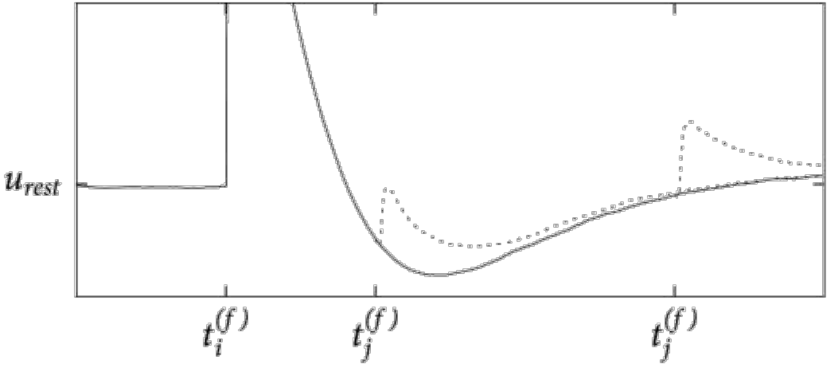


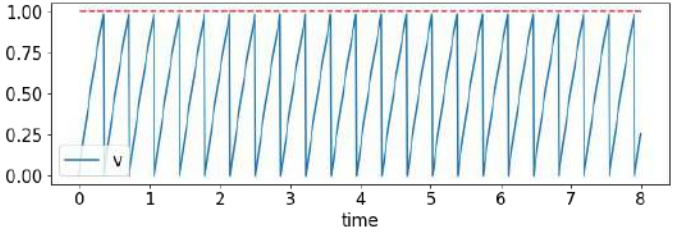
Limitations of the Leaky Integrate-and-Fire Model

2. Conductance Changes after a Spike:

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\mathrm{rest}}\right] + R\,I(t)$$

The shape of the postsynaptic potentials does not only depend on the level of depolarization but, more generally, on the internal state of the neuron, e.g., on the timing relative to previous action potentials.





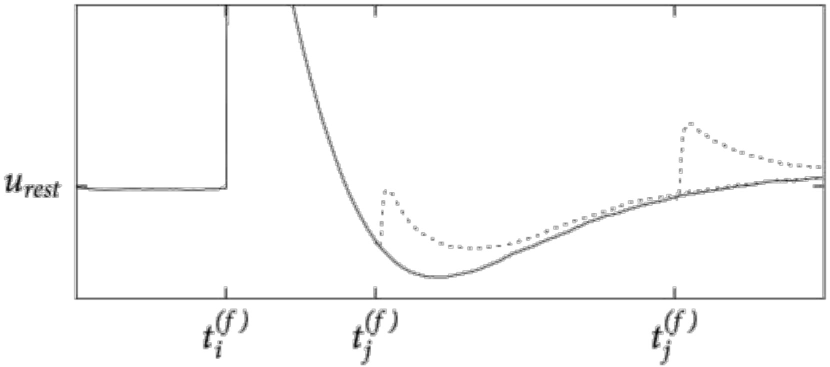
The shape of postsynaptic potentials (dashed lines) depends on the time t - $t_i^{(f)}$ that has passed since the last output spike of neuron i. The postsynaptic spike has been triggered at time $t_i^{(f)}$. A presynaptic spike that arrives at time $t_i^{(f)}$ shortly after the spike of the postsynaptic neuron has a smaller effect than a spike that arrives much later.

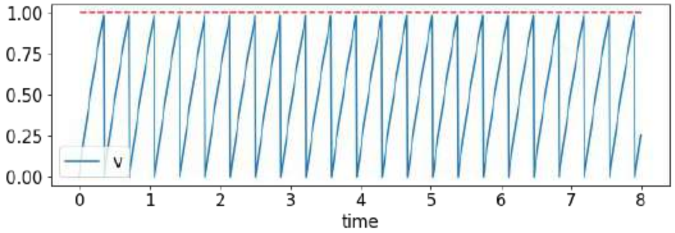
Limitations of the Leaky Integrate-and-Fire Model

2. Conductance Changes after a Spike:

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\text{rest}}\right] + R\,I(t)$$

The shape of the postsynaptic potentials does not only depend on the level of depolarization but, more generally, on the internal state of the neuron, e.g., on the timing relative to previous action potentials.





The shape of postsynaptic potentials (dashed lines) depends on the time t - $t_i^{(f)}$ that has passed since the last output spike of neuron i. The postsynaptic spike has been triggered at time $t_i^{(f)}$. A presynaptic spike that arrives at time $t_i^{(f)}$ shortly after the spike of the postsynaptic neuron has a smaller effect than a spike that arrives much later.

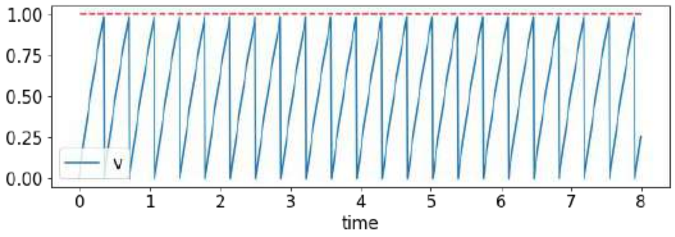
T. K. Berger, R. Perin, G. Silberberg and H. Markram (2009) *Frequency-dependent disynaptic inhibition in the pyramidal network: a ubiquitous pathway in the developing rat neocortex. The Journal of Physiology* **587** (22), pp. 5411–5425.

Limitations of the Leaky Integrate-and-Fire Model

3. Spatial Structure

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\text{rest}}\right] + R\,I(t)$$

The form of postsynaptic potentials also depends on the location of the synapse on the dendritic tree. Synapses that are located far away from the soma are expected to evoke a smaller postsynaptic response at the soma than a synapse that is located directly on the soma. If several inputs occur on the same dendritic branch within a few milliseconds, the first input will cause local changes of the membrane potential that influence the amplitude of the response to the input spikes that arrive slightly later. This may lead to saturation or, in the case of so-called ‘active’ currents, to an enhancement of the response. Such nonlinear interactions between different presynaptic spikes are neglected in the leaky integrate-and-fire model. Whereas a purely linear dendrite can be incorporated in the ‘filter’ description of the model, nonlinear interactions cannot. Small regions on the dendrite where a strong nonlinear boosting of synaptic currents occurs are sometimes called dendritic ’hot spots’. The boosting can lead to dendritic spikes which, in contrast to normal somatic action potentials last for tens of milliseconds.



Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\,I\left(t\right)$$

Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\, I\left(t\right)$$

u is the membrane potential
τ_m is the membrane time constant
R is the resistance
I(t) is an external current

Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\text{rest}}\right] + R\, I\left(t\right)$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
τ_m is the membrane time constant
R is the resistance
I(t) is an external current

Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u(t) - u_{\mathrm{rest}}\right] + R\,I(t)$$

u is the membrane potential
τ_m is the membrane time constant
R is the resistance
$I(t)$ is an external current

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\,I\left(t\right)$$

u is the membrane potential
τ_m is the membrane time constant
R is the resistance
I(t) is an external current

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\text{rest}}\right] + R\,I\left(t\right)$$

u is the membrane potential
τ_m is the membrane time constant
R is the resistance
I(t) is an external current

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\mathrm{rest}}\right] + R\, I\left(t\right)$$

u is the membrane potential
τ_m is the membrane time constant
R is the resistance
I(t) is an external current

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = f\left(u\right) + R\left(u\right)\, I$$

Integrate-and-fire models

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -\left[u\left(t\right) - u_{\text{rest}}\right] + R\,I\left(t\right)$$

u is the membrane potential
τ_m is the membrane time constant
R is the resistance
I(t) is an external current

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = f\left(u\right) + R\left(u\right)\, I$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = f(u) + R(u)\ I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = \boxed{f(u)} + R(u)\; I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = \boxed{f(u)} + R(u)\ I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

f(u)??

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = \boxed{f(u)} + R(u)\; I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

f(u)??

Extracting the Nonlinearity from Data!

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = \boxed{f(u)} + R(u)\; I$$

$$t^{(f)} : \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

f(u)??

Extracting the Nonlinearity from Data! Consider:

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = \boxed{f\left(u\right)} + R\left(u\right)\ I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

f(u)??

Extracting the Nonlinearity from Data! Consider: $\tilde{f}\left(u\right) = f\left(u\right)/\tau$

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = \boxed{f(u)} + R(u)\; I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

f(u)??

Extracting the Nonlinearity from Data! Consider:

$$\tilde{f}(u) = f(u)/\tau$$

$$\tilde{f}(u(t)) = \frac{1}{C} I(t) - \frac{\mathrm{d}}{\mathrm{d}t} u(t)$$

General integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = \boxed{f(u)} + R(u)\; I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

f(u)??

$$\tilde{f}(u) = f(u)/\tau$$

Extracting the Nonlinearity from Data! Consider:

$$\tilde{f}(u(t)) = \frac{1}{C} I(t) - \frac{\mathrm{d}}{\mathrm{d}t} u(t)$$

In order to determine the function f˜(u), an experimentalist injects a time-dependent current I(t) into the soma of a neuron while measuring with a second electrode the voltage u(t). From the voltage time course, one finds the voltage derivative du/dt.

For each voltage u there are many data points.
At the end, we average across all points at a given voltage u to find the empirical function

$$\tilde{f}(u(t)) = \left\langle \frac{1}{C} I(t) - \frac{\mathrm{d}}{\mathrm{d}t} u(t) \right\rangle$$

How to choose f(u)?

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = f(u) + R(u)\ I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

For example, Exponential Integrate-and-Fire Model:

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -(u - u_{\text{rest}}) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\, I$$

u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current

Exponential integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -\left(u - u_{\mathrm{rest}}\right) + \Delta_T \, \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\, I$$

Exponential integrate-and-fire model

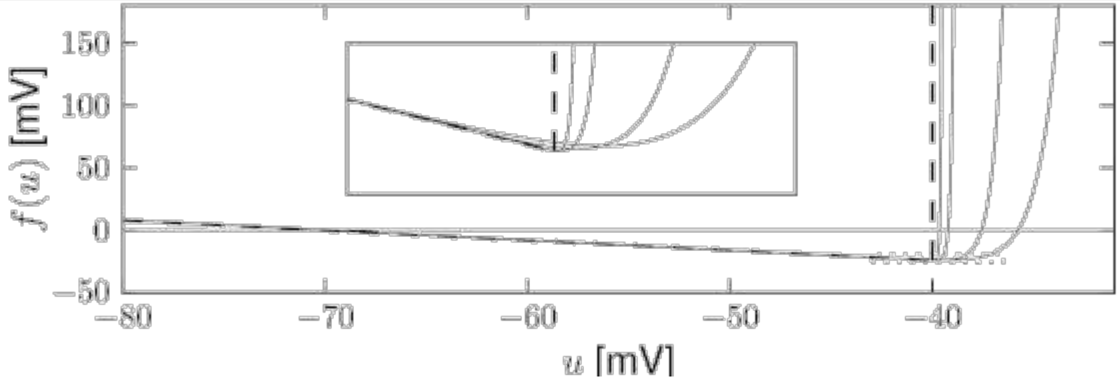
$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -\left(u - u_{\mathrm{rest}}\right) + \Delta_T \, \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\, I$$

u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current

Exponential integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -(u - u_{\mathrm{rest}}) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\,I$$

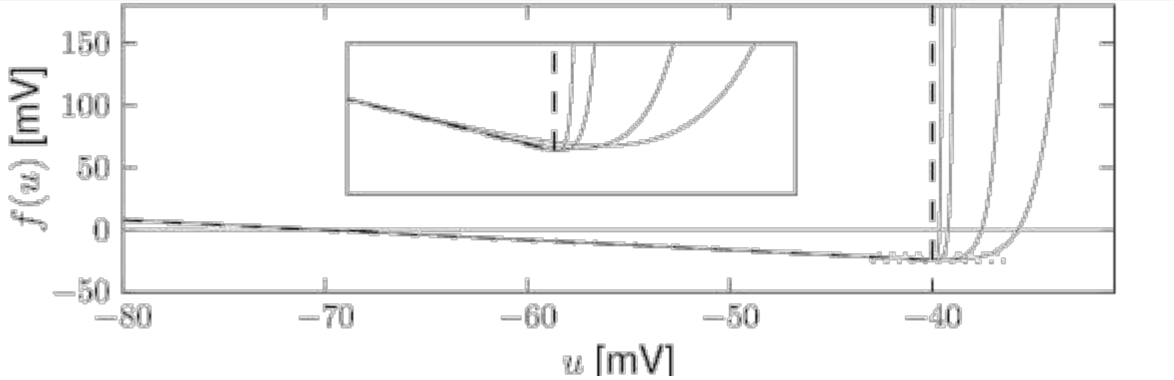
u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current



Exponential integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -(u - u_{\mathrm{rest}}) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\,I$$

u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current



The function f(u) is plotted for different choices of the ’sharpness’ of the threshold (Δ_T = 1, 0.5, 0.25, 0.05 mV) In the limit Δ_T → 0 the EIF model becomes equivalent to LIF model (dashed line). The inset shows a zoom onto the threshold region (dotted box).

Exponential integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -(u - u_{\mathrm{rest}}) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\,I$$

u is the membrane potential

u_{rest} is the resting potential

τ is the membrane time constant

Δ_T is a “sharpness” parameter

ϑ_{rh} is a threshold

R(u) is the resistance

I is an external current

Exponential integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -(u - u_{\text{rest}}) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\, I$$

u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current

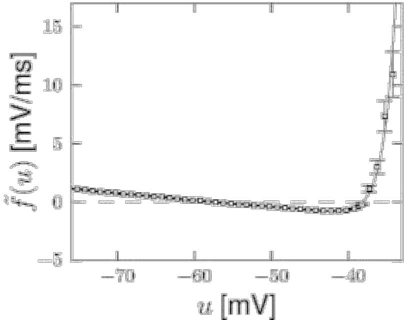
Extracting nonlinear integrate-and-fire models from data. The function f(u) characterizing the nonlinearity of an IF-model is derived from experimental data using random current injection into neurons.

Exponential integrate-and-fire model

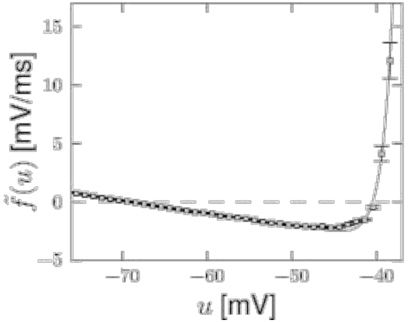
$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -(u - u_{\mathrm{rest}}) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\,I$$

u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current

Extracting nonlinear integrate-and-fire models from data. The function f(u) characterizing the nonlinearity of an IF-model is derived from experimental data using random current injection into neurons.



Cortical pyramidal cells.



Inhibitory interneurons

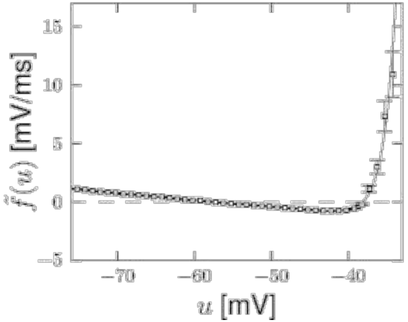
Exponential integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -(u - u_{\mathrm{rest}}) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\,I$$

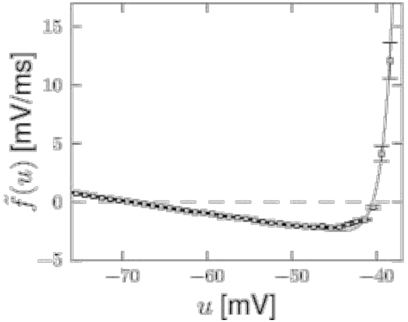
u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current

Extracting nonlinear integrate-and-fire models from data. The function f(u) characterizing the nonlinearity of an IF-model is derived from experimental data using random current injection into neurons.

$$\tilde{f}\,(u\,(t)) = \left\langle \frac{1}{C}\,I\,(t) - \frac{\mathrm{d}}{\mathrm{d}t}u\,(t) \right\rangle$$



Cortical pyramidal cells.



Inhibitory interneurons

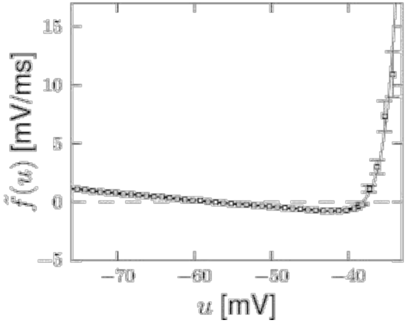
Exponential integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = -\left(u - u_{\mathrm{rest}}\right) + \Delta_T \exp\left(\frac{u - \vartheta_{rh}}{\Delta_T}\right) + R\,I$$

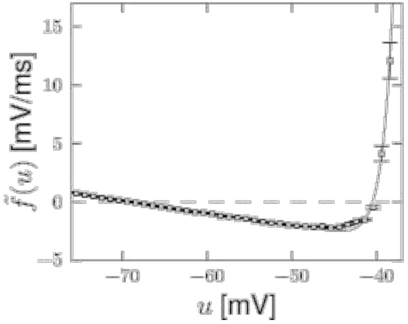
u is the membrane potential
u_{rest} is the resting potential
τ is the membrane time constant
Δ_T is a “sharpness” parameter
ϑ_{rh} is a threshold
R(u) is the resistance
I is an external current

Extracting nonlinear integrate-and-fire models from data. The function f(u) characterizing the nonlinearity of an IF-model is derived from experimental data using random current injection into neurons.

$$\tilde{f}\,\left(u\left(t\right)\right) = \left\langle \frac{1}{C}\, I\left(t\right) - \frac{\mathrm{d}}{\mathrm{d}t} u\left(t\right) \right\rangle$$



Cortical pyramidal cells.



Inhibitory interneurons

L. Badel, S. Lefort, T.K. Berger, C. Petersen, W. Gerstner and M.J.E. Richardson (2008) *Extracting non-linear integrate-and-fire models from experimental data using dynamic i-v curves*. *Biological Cybernetics* **99** (4-5), pp. 361–370.

How to choose f(u)?

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = f(u) + R(u)\ I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

How to choose f(u)?

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = f(u) + R(u)\ I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\mathrm{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

For example, Quadratic Integrate-and-Fire Model:

How to choose f(u)?

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = f\left(u\right) + R\left(u\right)\ I$$

$$t^{(f)}: \quad u\left(t^{(f)}\right) = \theta_{\text{reset}} \quad \text{and} \quad \left.\frac{\mathrm{d}u\left(t\right)}{\mathrm{d}t}\right|_{t=t^{(f)}} > 0$$

u is the membrane potential
f(u) is a nonlinear function of u
τ is the membrane time constant
R(u) is the resistance
I is an external current

$t^{(f)}$ is a firing time
θ_{reset} is the firing threshold

For example, Quadratic Integrate-and-Fire Model:

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = a_0\ \left(u - u_{\text{rest}}\right)\ \left(u - u_c\right) + RI$$

Quadratic integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = a_0 \ (u - u_{\mathrm{rest}}) \ (u - u_c) + RI$$

u is the membrane potential
u_c is the critical voltage for spike initiation by a short current pulse
τ is the membrane time constant
$R(u)$ is the resistance
I is an external current

Quadratic integrate-and-fire model

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = a_0 \ (u - u_{\mathrm{rest}}) \ (u - u_c) + RI$$

u is the membrane potential
u_c is the critical voltage for spike
initiation by a short current pulse
τ is the membrane time constant
R(u) is the resistance
I is an external current

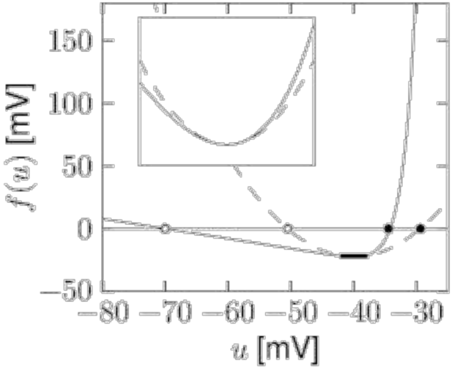
The quadratic integrate-and-fire model (dashed line), compared to an exponential integrate-and-fire model (solid line):

Quadratic integrate-and-fire model

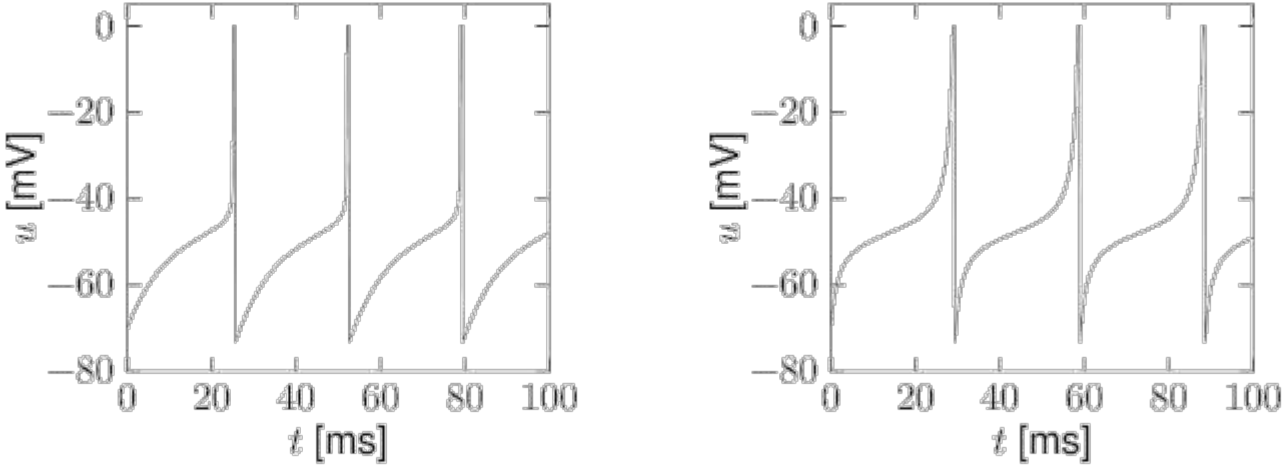
$$\tau \frac{\mathrm{d}}{\mathrm{d}t} u = a_0 \ (u - u_{\text{rest}}) \ (u - u_c) + RI$$

u is the membrane potential
u_c is the critical voltage for spike initiation by a short current pulse
τ is the membrane time constant
R(u) is the resistance
I is an external current

The quadratic integrate-and-fire model (dashed line), compared to an exponential integrate-and-fire model (solid line):



Comparing QIF and EIF models



Repetitive firing in Nonlinear integrate-and-fire.
Left: Exponential Integrate-and-Fire Model and
Right Quadratic Integrate-and-Fire Model receiving a constant current sufficient to elicit repetitive firing.

Note the comparatively slow upswing of the action potential in the quadratic integrate-and-fire model.

Advantages and disadvantages of IF models

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.
- Nonlinear integrate-and-fire neurons, however, can account for the fact that in real neurons the effective voltage threshold for repetitive firing is different than the voltage threshold found with short current pulses.

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.
- Nonlinear integrate-and-fire neurons, however, can account for the fact that in real neurons the effective voltage threshold for repetitive firing is different than the voltage threshold found with short current pulses.
- Once the membrane potential is above the intrinsic threshold, the upswing of the membrane potential starts. The integration is stopped at a numerical threshold θreset which is much higher and conceptually very different than the intrinsic firing threshold of the model. In fact, the exact value of the numerical threshold does not matter, since, without such a threshold, the membrane potential would go to infinity in finite time.

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.
- Nonlinear integrate-and-fire neurons, however, can account for the fact that in real neurons the effective voltage threshold for repetitive firing is different than the voltage threshold found with short current pulses.
- Once the membrane potential is above the intrinsic threshold, the upswing of the membrane potential starts. The integration is stopped at a numerical threshold θreset which is much higher and conceptually very different than the intrinsic firing threshold of the model. In fact, the exact value of the numerical threshold does not matter, since, without such a threshold, the membrane potential would go to infinity in finite time.
- EIF model has the ‘correct’ nonlinearity

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.
- Nonlinear integrate-and-fire neurons, however, can account for the fact that in real neurons the effective voltage threshold for repetitive firing is different than the voltage threshold found with short current pulses.
- Once the membrane potential is above the intrinsic threshold, the upswing of the membrane potential starts. The integration is stopped at a numerical threshold θreset which is much higher and conceptually very different than the intrinsic firing threshold of the model. In fact, the exact value of the numerical threshold does not matter, since, without such a threshold, the membrane potential would go to infinity in finite time.
- EIF model has the ‘correct’ nonlinearity
- QIF model is too nonlinear in the subthreshold regime and too slow in the superthreshold regime.

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.
- Nonlinear integrate-and-fire neurons, however, can account for the fact that in real neurons the effective voltage threshold for repetitive firing is different than the voltage threshold found with short current pulses.
- Once the membrane potential is above the intrinsic threshold, the upswing of the membrane potential starts. The integration is stopped at a numerical threshold θreset which is much higher and conceptually very different than the intrinsic firing threshold of the model. In fact, the exact value of the numerical threshold does not matter, since, without such a threshold, the membrane potential would go to infinity in finite time.
- EIF model has the ‘correct’ nonlinearity
- QIF model is too nonlinear in the subthreshold regime and too slow in the superthreshold regime.
- IF-models show a frequency-current curve of type I.

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.
- Nonlinear integrate-and-fire neurons, however, can account for the fact that in real neurons the effective voltage threshold for repetitive firing is different than the voltage threshold found with short current pulses.
- Once the membrane potential is above the intrinsic threshold, the upswing of the membrane potential starts. The integration is stopped at a numerical threshold θreset which is much higher and conceptually very different than the intrinsic firing threshold of the model. In fact, the exact value of the numerical threshold does not matter, since, without such a threshold, the membrane potential would go to infinity in finite time.
- EIF model has the ‘correct’ nonlinearity
- QIF model is too nonlinear in the subthreshold regime and too slow in the superthreshold regime.
- IF-models show a frequency-current curve of type I.
- IF-models are very easy to implement and compute

Advantages and disadvantages of IF models

- LIF model is rather limited in scope, since it has one universal voltage threshold.
- Nonlinear integrate-and-fire neurons, however, can account for the fact that in real neurons the effective voltage threshold for repetitive firing is different than the voltage threshold found with short current pulses.
- Once the membrane potential is above the intrinsic threshold, the upswing of the membrane potential starts. The integration is stopped at a numerical threshold θreset which is much higher and conceptually very different than the intrinsic firing threshold of the model. In fact, the exact value of the numerical threshold does not matter, since, without such a threshold, the membrane potential would go to infinity in finite time.
- EIF model has the ‘correct’ nonlinearity
- QIF model is too nonlinear in the subthreshold regime and too slow in the superthreshold regime.
- IF-models show a frequency-current curve of type I.
- IF-models are very easy to implement and compute
- QIF model is used in the construction of the mean field reduction (population activity)

Synapses

Olesia Dogonasheva
odogonasheva@hse.ru

Outline

1. Synapses
   a. Classification
   b. Simple model
2. Networks

Synaptic connections in neural ensembles

Synaptic connections in neural ensembles

Main types:

- electrical;

Synaptic connections in neural ensembles

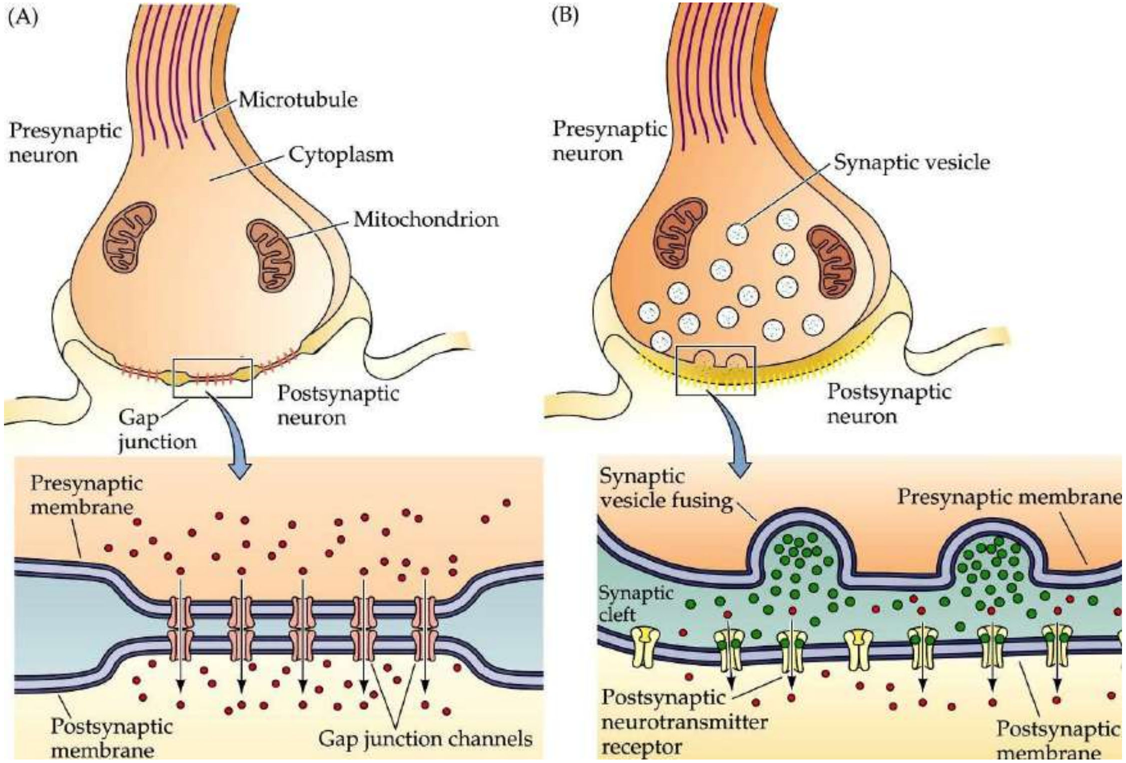
Main types:

- electrical;
- chemical.

Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.



Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.

Classification by effect on the postsynaptic membrane:

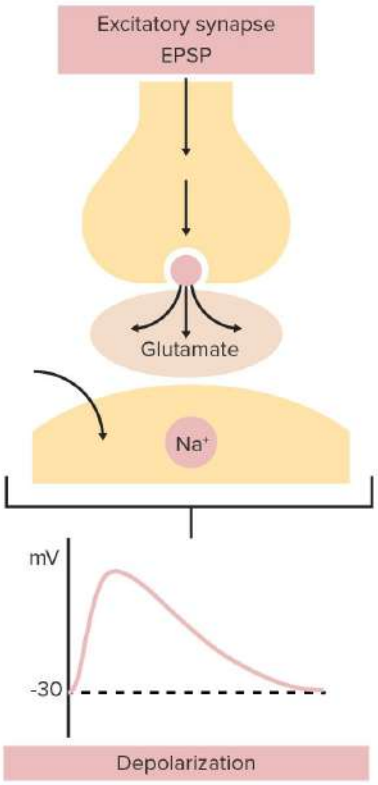
Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.

Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;



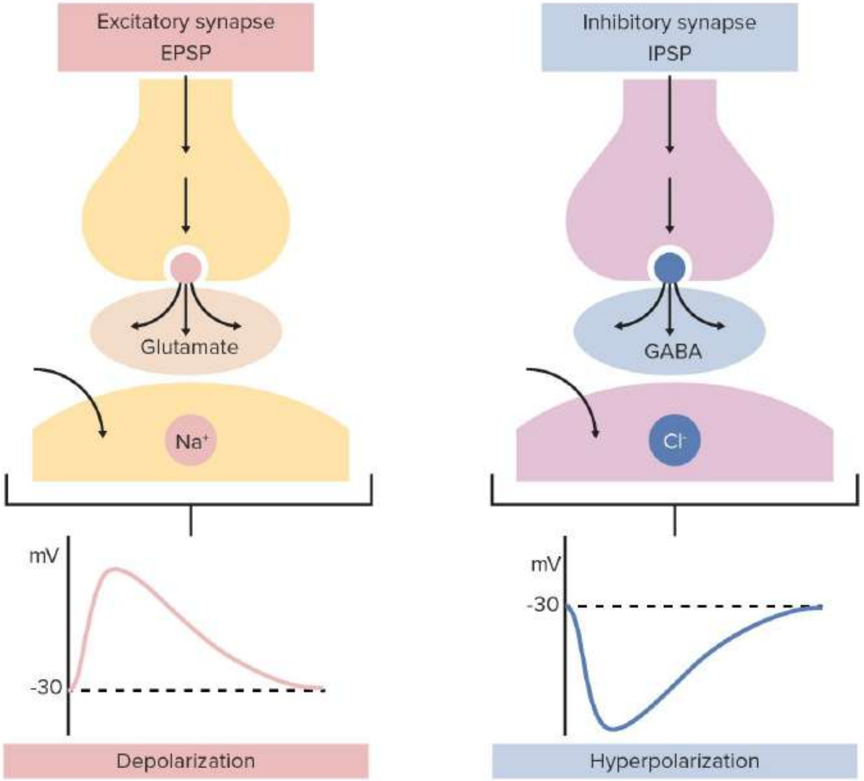
Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.

Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;
- inhibitory: hyperpolarized post-synaptic membrane.



Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.

Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;
- inhibitory: hyperpolarized post-synaptic membrane.

Cell attachment types:

Synaptic connections in neural ensembles

Main types:

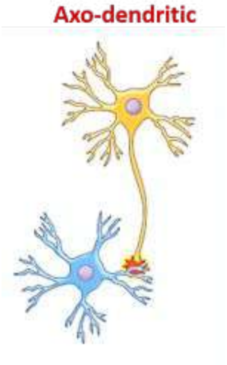
- electrical;
- chemical.

Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;
- inhibitory: hyperpolarized post-synaptic membrane.

Cell attachment types:

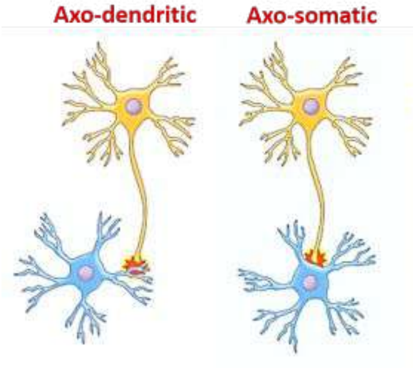
- Axodendritic: the most common synapse in the human body.



Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.



Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;
- inhibitory: hyperpolarized post-synaptic membrane.

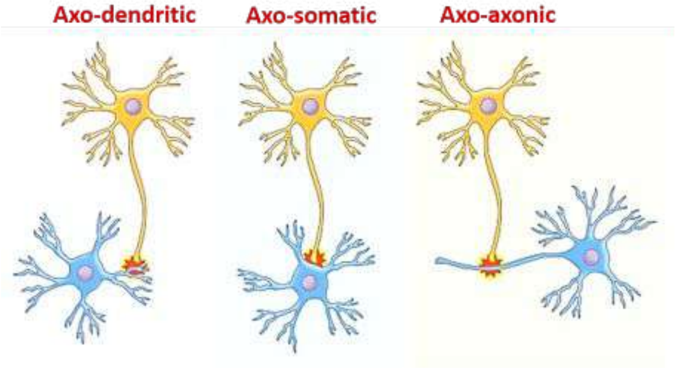
Cell attachment types:

- Axodendritic: the most common synapse in the human body.
- Axosomatic: connects to the cell membrane of the body or soma of another cell.

Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.



Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;
- inhibitory: hyperpolarized post-synaptic membrane.

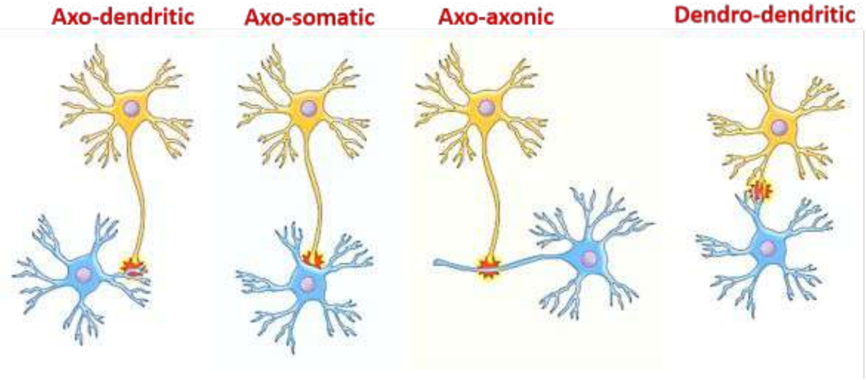
Cell attachment types:

- Axodendritic: the most common synapse in the human body.
- Axosomatic: connects to the cell membrane of the body or soma of another cell.
- Axo-axonic: usually, these are inhibitory synapses.

Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.



Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;
- inhibitory: hyperpolarized post-synaptic membrane.

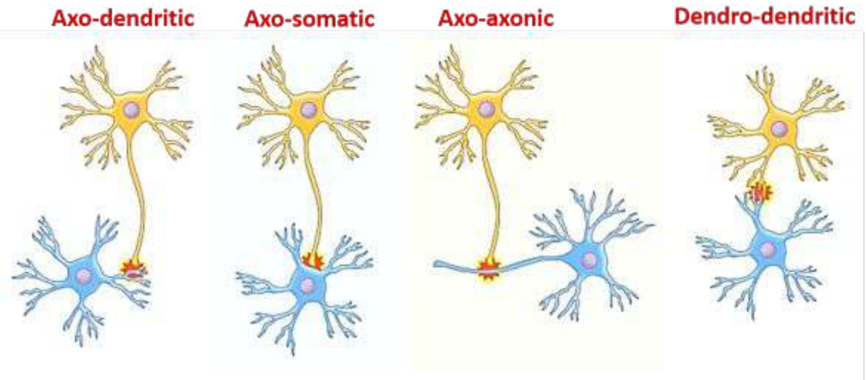
Cell attachment types:

- Axodendritic: the most common synapse in the human body.
- Axosomatic: connects to the cell membrane of the body or soma of another cell.
- Axo-axonic: usually, these are inhibitory synapses.
- Dendro-dendritic: these are dendritic connections between two different neurons.

Synaptic connections in neural ensembles

Main types:

- electrical;
- chemical.



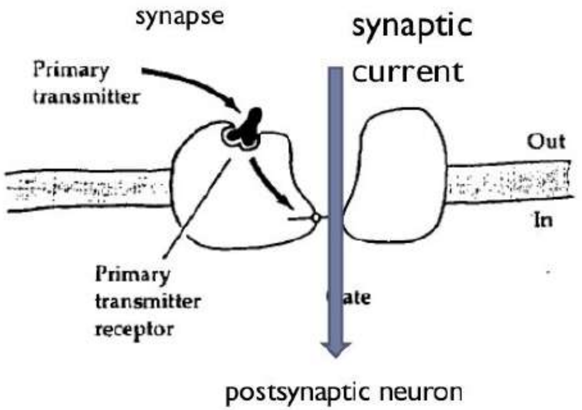
Classification by effect on the postsynaptic membrane:

- excitatory: depolarized post-synaptic membrane;
- inhibitory: hyperpolarized post-synaptic membrane.

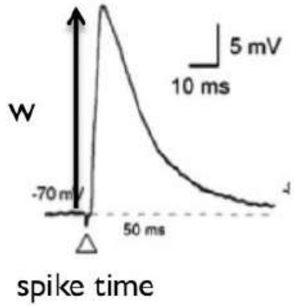
Cell attachment types:

- Axodendritic: the most common synapse in the human body.
- Axosomatic: connects to the cell membrane of the body or soma of another cell.
- Axo-axonic: usually, these are inhibitory synapses.
- Dendro-dendritic: these are dendritic connections between two different neurons.
- Neuromuscular: these types of synapses are highly specialised. Usually, these are large synapses that convert the electrical impulses in the motor neuron into the electrical activity that causes muscle contractions. All neuromuscular junctions use acetylcholine as a neurotransmitter.

Idealized synapse (instantaneous)



« post-synaptic potential (PSP) »

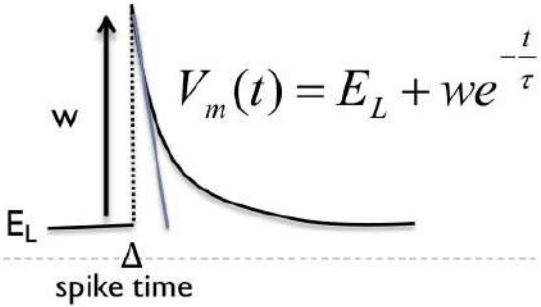


w = « synaptic weight »

Membrane equation:

$$\tau \frac{dV_m}{dt} = E_L - V_m$$

$$V_m \to V_m + w$$ **at spike time**



$$V_m(t) = E_L + we^{-\frac{t}{\tau}}$$

A more realistic synapse model

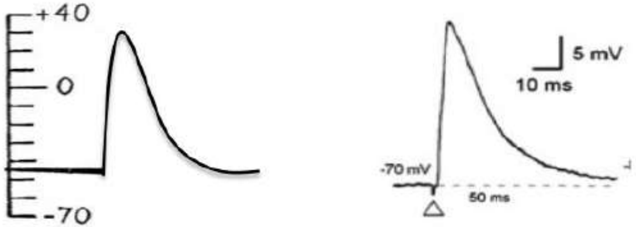
Synaptic current

$$I_s = g_s(E_s - V_m)$$

Ionic channel conductance

Synaptic reversal potential

Post Synaptic potential



Computational principles of synaptic memory consolidation, MK Benna, S Fusi, Nature neuroscience, (2016)

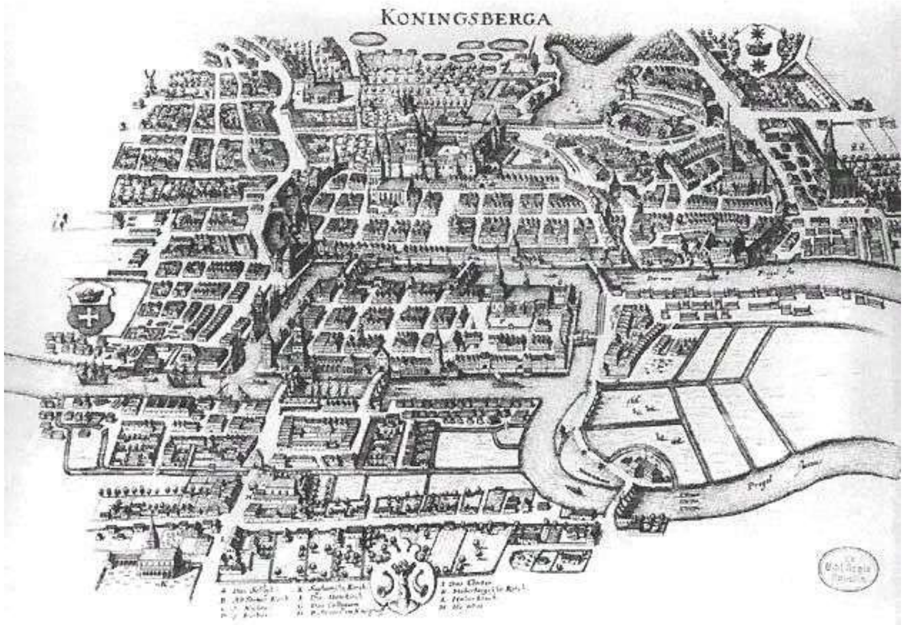
Code for network of LIF

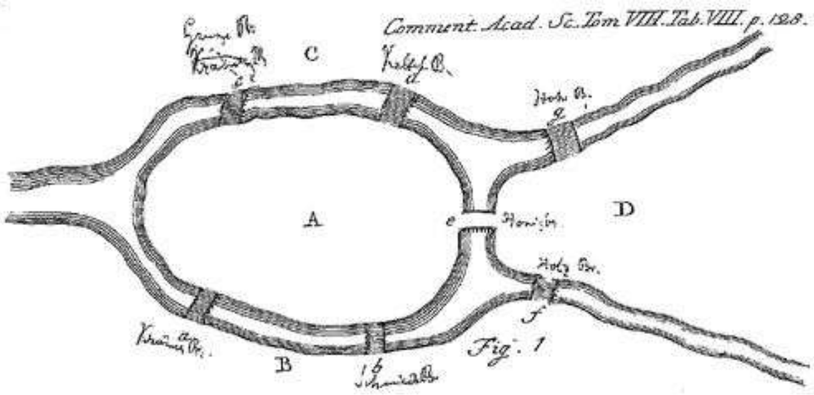
Questions

Graph theory

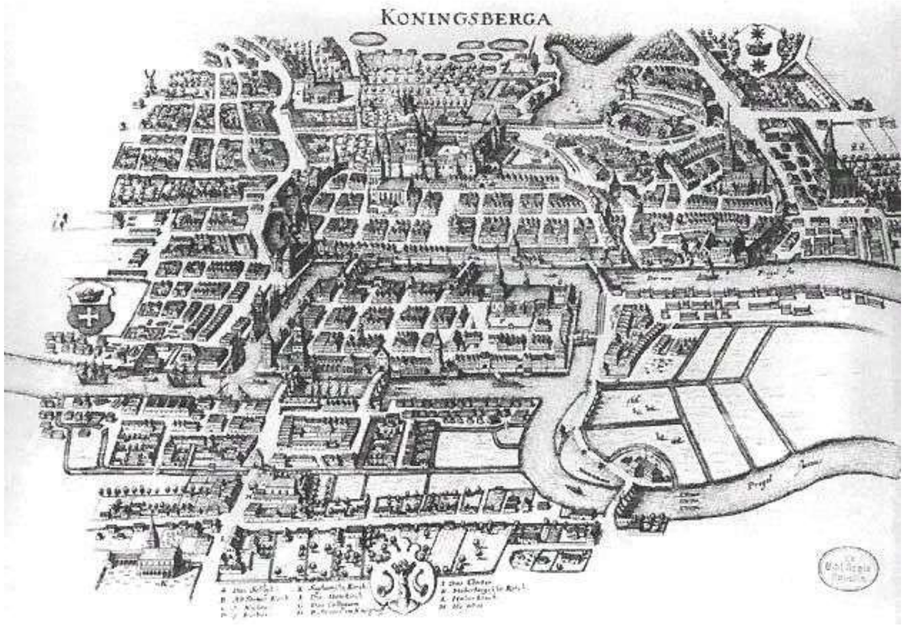
Introduction

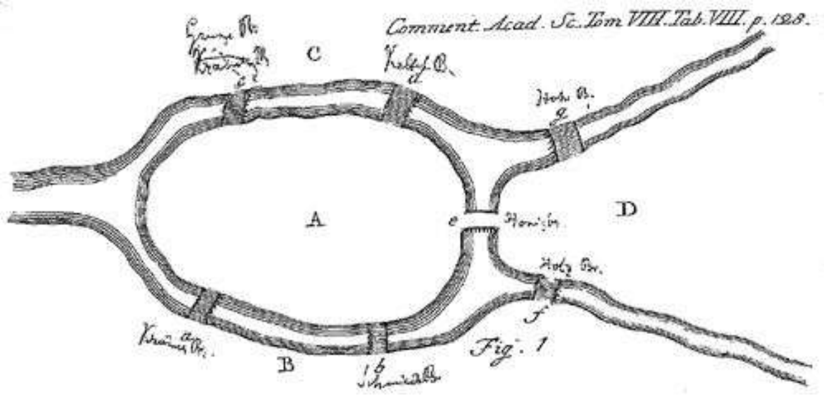
Seven Bridges of Königsberg





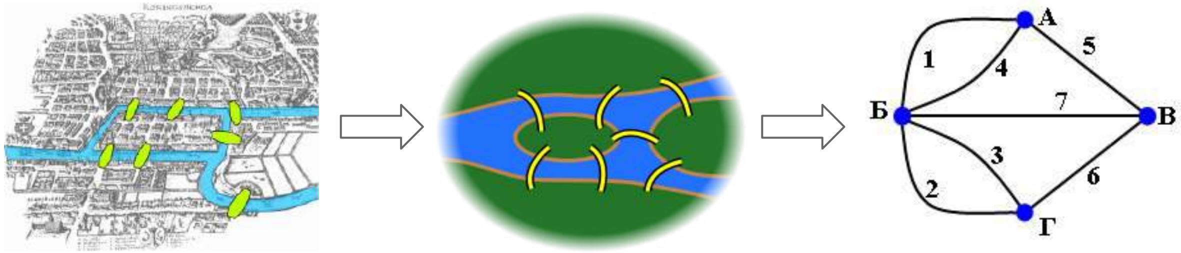
Seven Bridges of Königsberg



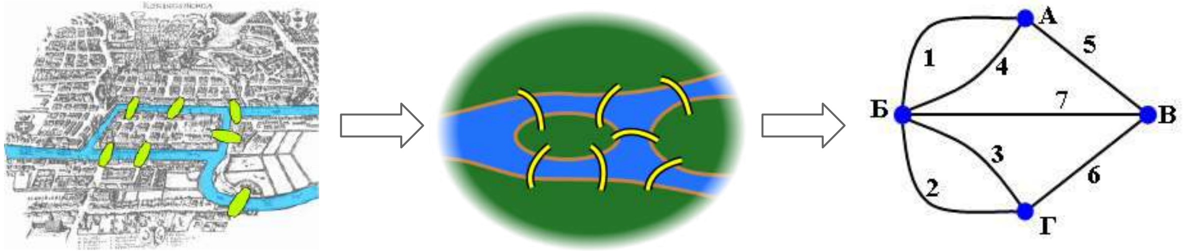


The problem is to devise a walk through the city that would cross each of those bridges once and only once.

1. Construct a **graph**:



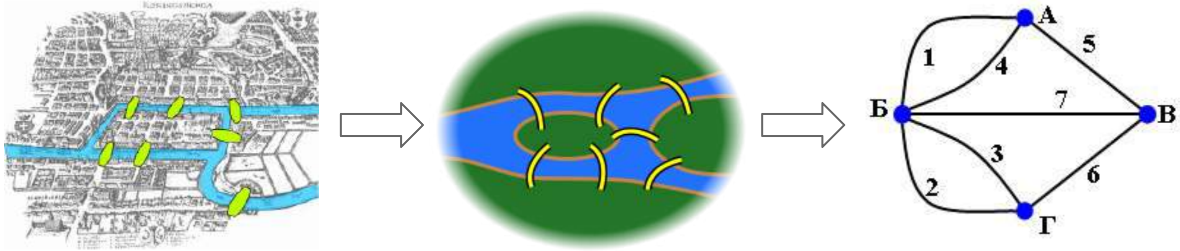
1. Construct a **graph**:



2. Define

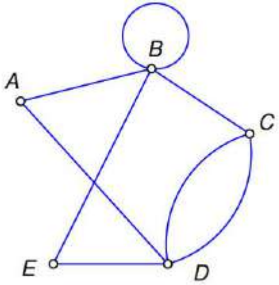
   - **Euler path (trail)** as a path that uses every edge of a graph exactly once.
   - Define **Euler circuit** as a circuit that uses every edge of a graph exactly once.
   - Vertex degree is a number edges with that vertex as an end-point

1. Construct a **graph**:

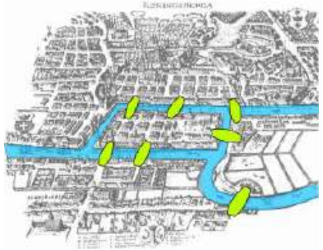


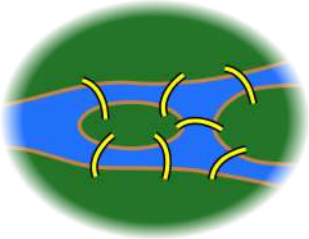
2. Define

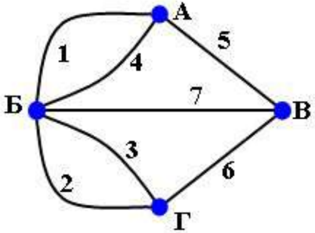
   - **Euler path (trail)** as a path that uses every edge of a graph exactly once.
   - Define **Euler circuit** as a circuit that uses every edge of a graph exactly once.
   - Vertex degree is a number edges with that vertex as an end-point



1. Construct a **graph**:





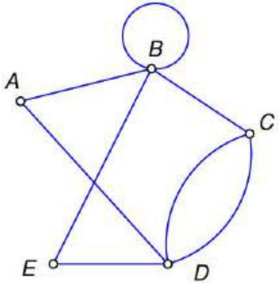


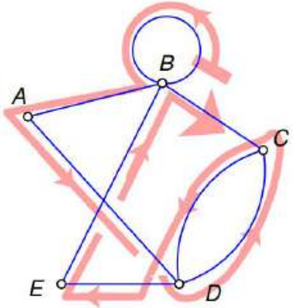
2. Define

○ **Euler path (trail)** as a path that uses every edge of a graph exactly once.

○ Define **Euler circuit** as a circuit that uses every edge of a graph exactly once.

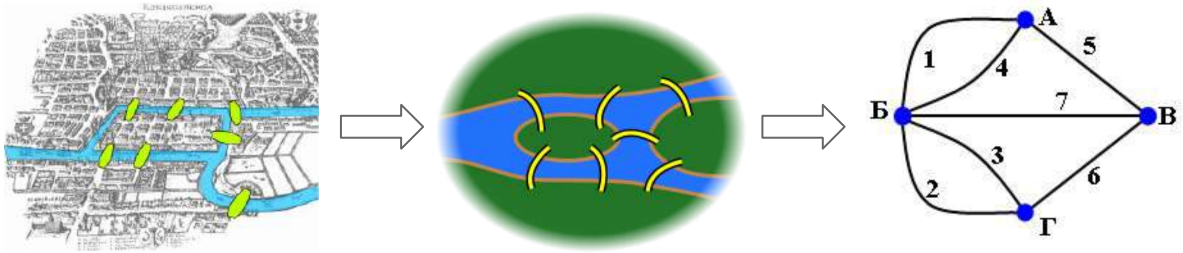
○ Vertex degree is a number edges with that vertex as an end-point





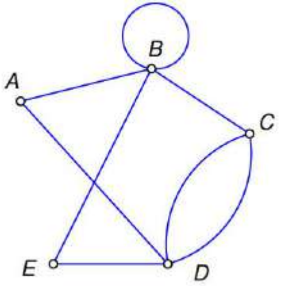
An Euler path: BBADCDEBC

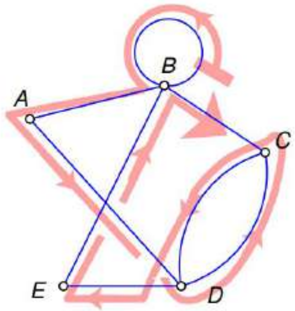
1. Construct a **graph**:



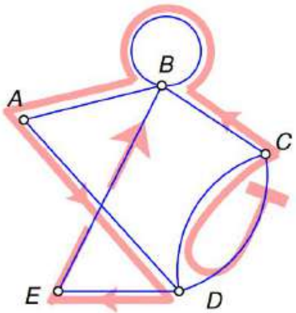
2. Define

   - **Euler path (trail)** as a path that uses every edge of a graph exactly once.
   - Define **Euler circuit** as a circuit that uses every edge of a graph exactly once.
   - Vertex degree is a number edges with that vertex as an end-point



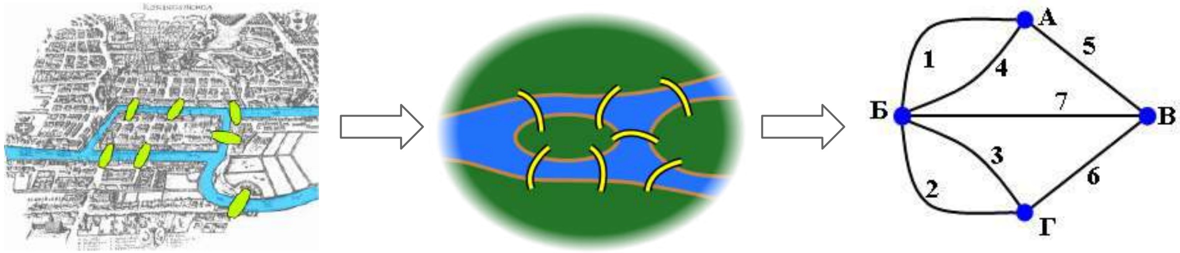


An Euler path: BBADCDEBC



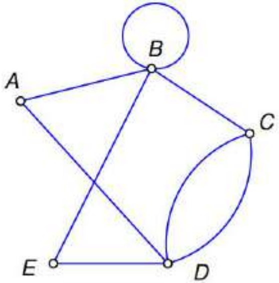
Another Euler path: CDCBBADEB

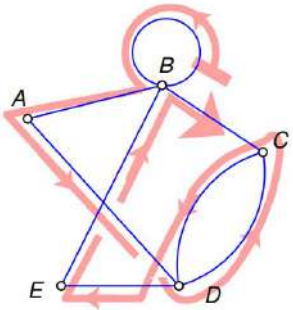
1. Construct a **graph**:



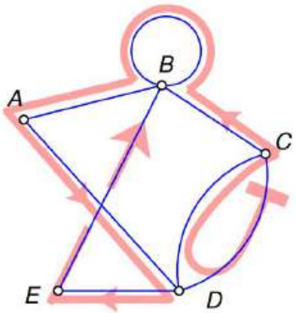
2. Define

- **Euler path (trail)** as a path that uses every edge of a graph exactly once.
- Define **Euler circuit** as a circuit that uses every edge of a graph exactly once.
- Vertex degree is a number edges with that vertex as an end-point

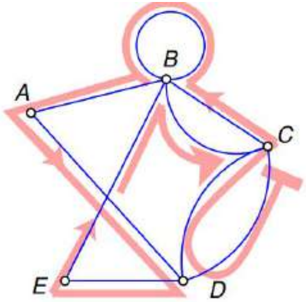




An Euler path: BBADCDEBC

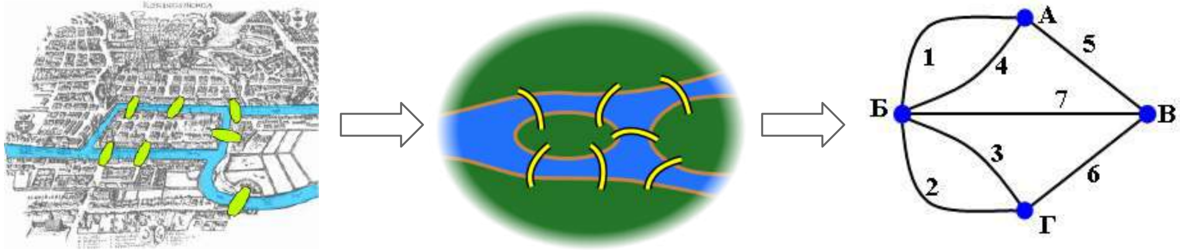


Another Euler path: CDCBBADEB



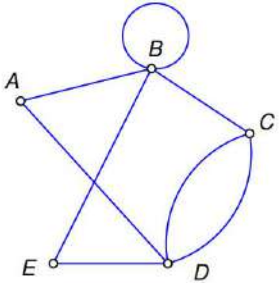
An Euler circuit: CDCBBADEBC

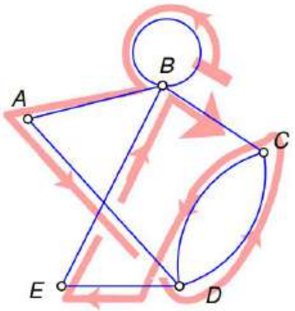
1. Construct a **graph**:



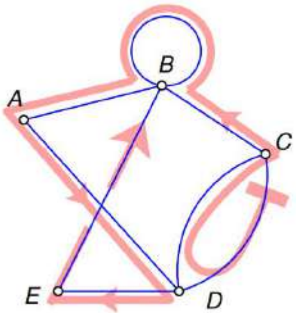
2. Define

- **Euler path (trail)** as a path that uses every edge of a graph exactly once.
- Define **Euler circuit** as a circuit that uses every edge of a graph exactly once.
- Vertex degree is a number edges with that vertex as an end-point

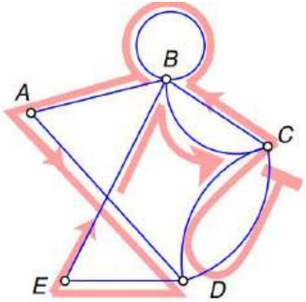




An Euler path: BBADCDEBC



Another Euler path: CDCBBADEB



An Euler circuit: CDCBBADEBC

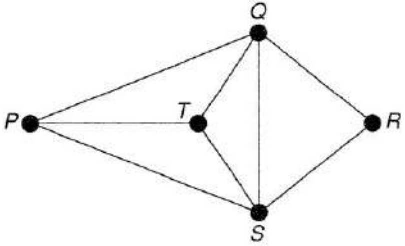
3. **Euler's theorem**:

★ If a graph G has an **Euler path**, then it must have **exactly two odd** vertices.

★ If a graph G has an **Euler circuit**, then all of its vertices must be **even vertices**.

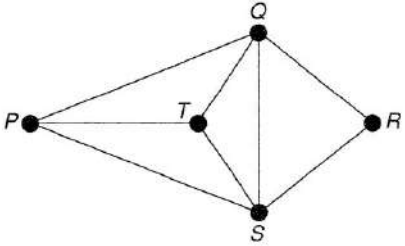
Glossary

- Simple graph



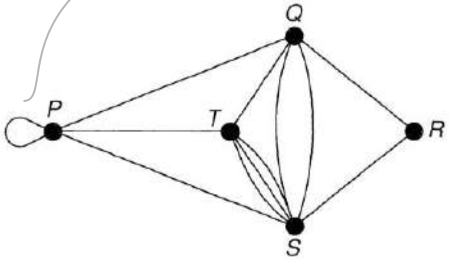
Glossary

- ➔ Simple graph



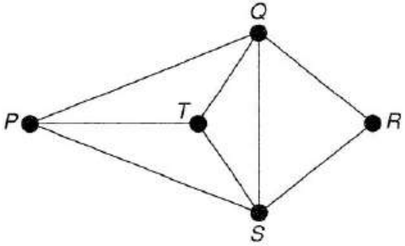
- ➔ General graph may have loops

Loop is an edge
from some
vertex to itself

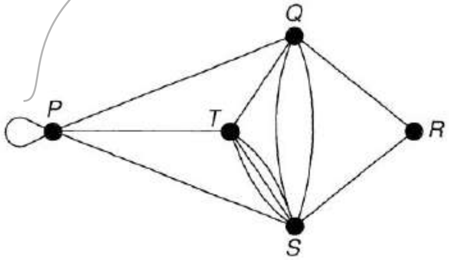


Glossary

- ➔ Simple graph

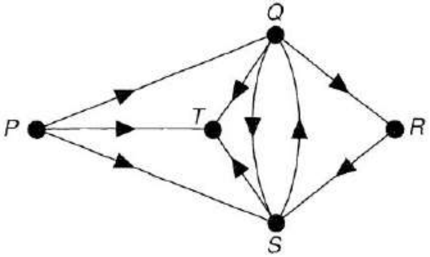


- ➔ General graph may have loops



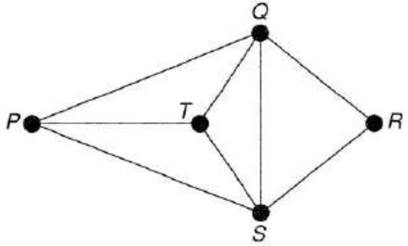
Loop is an edge from some vertex to itself

- ➔ Directed graph



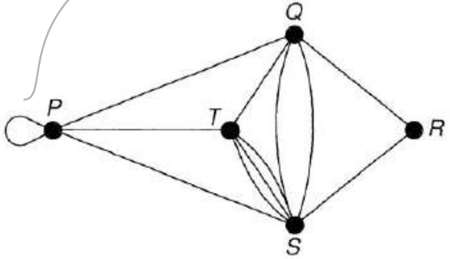
Glossary

- ➔ Simple graph

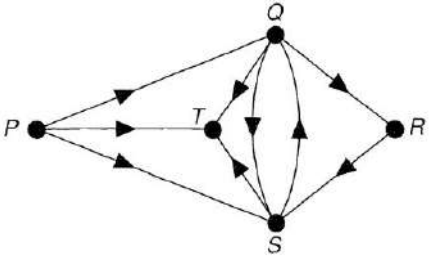


- ➔ General graph may have loops

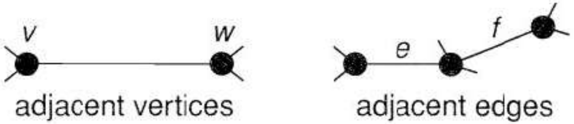
Loop is an edge from some vertex to itself



- ➔ Directed graph

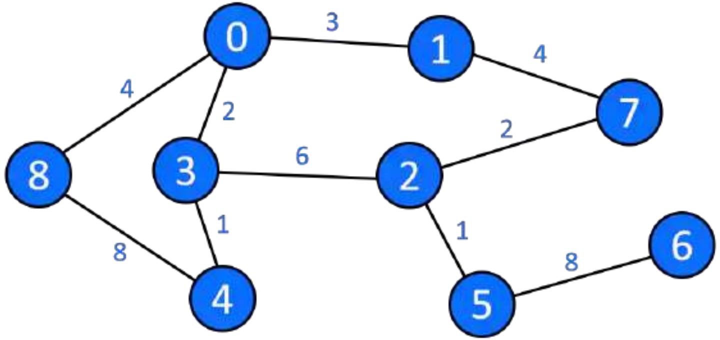


- ➔ Adjacency:



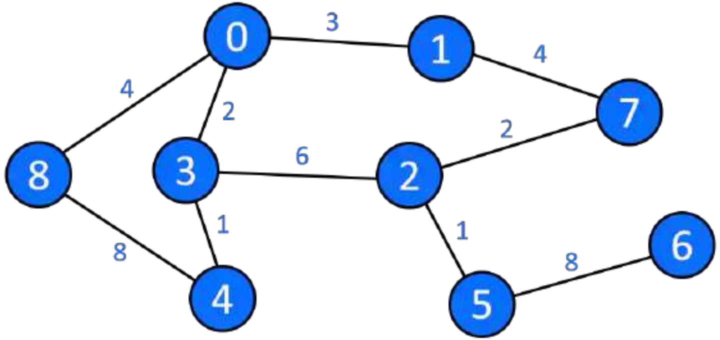
Glossary

- ➔ Weighted graph := graph which edges are associated with some numbers

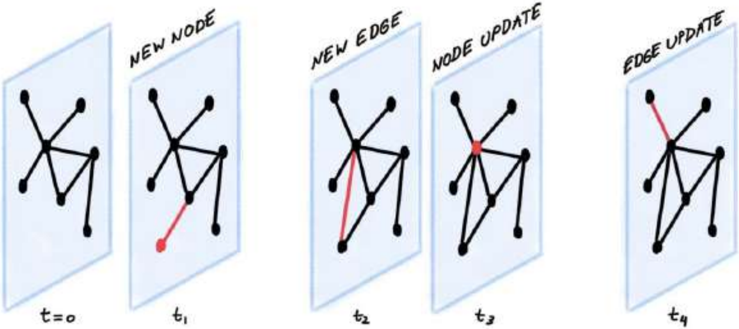


Glossary

➔ Weighted graph := graph which edges are associated with some numbers



➔ Dynamic graph := graph that can change during time



Glossary

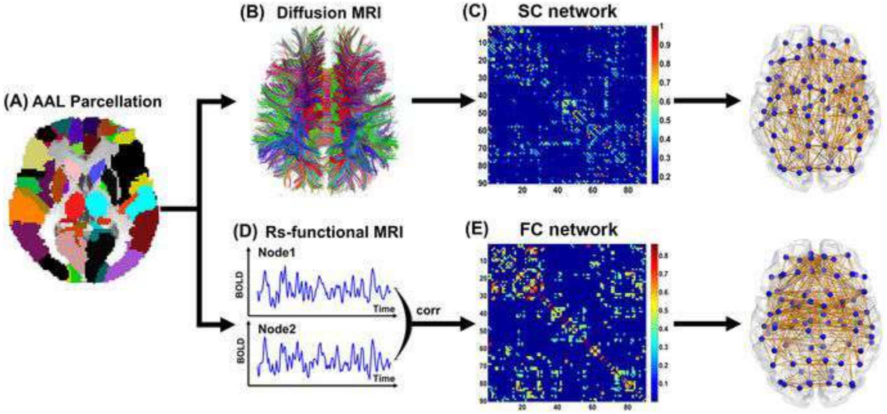
- ➔ Weighted graph := graph which edges are associated with some numbers

- ➔ Dynamic graph := weighted graph where weight can change during time

Structural connectome:

Nodes := *brain regions*
Edges := *synaptic path between regions*
Weights := *Maybe synaptic strength*

It is a weighted or unweighted static graph!

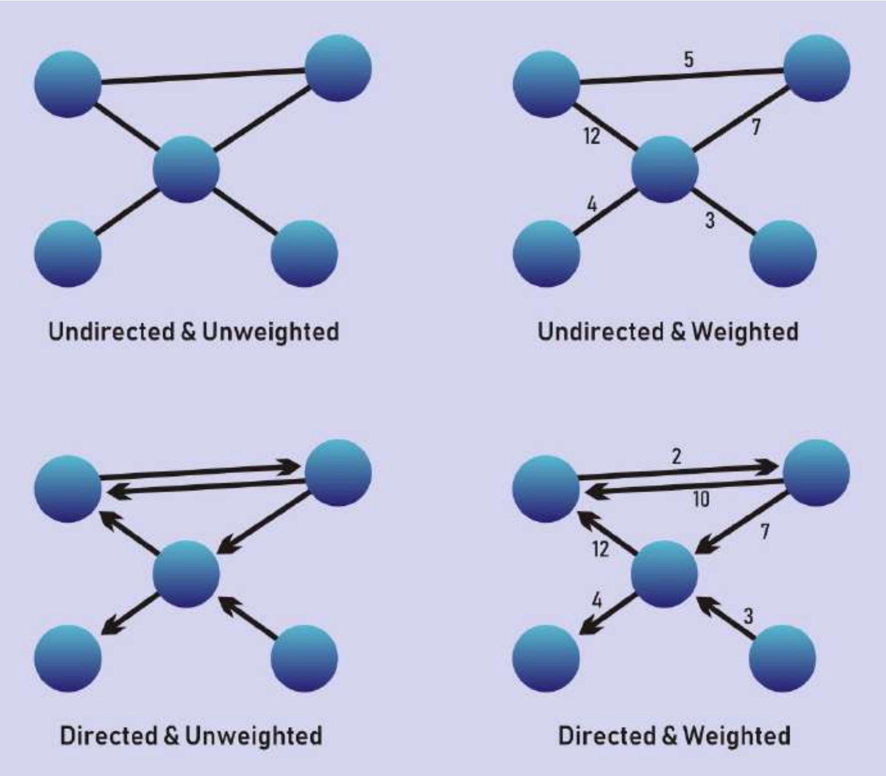


Functional connectome:

Nodes := *brain regions*
Edges := *Partial synchronization between regions*
Weights := *Correlation of signals*

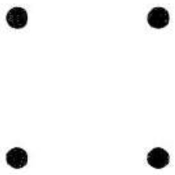
It is a weighted dynamic graph!

Base classification



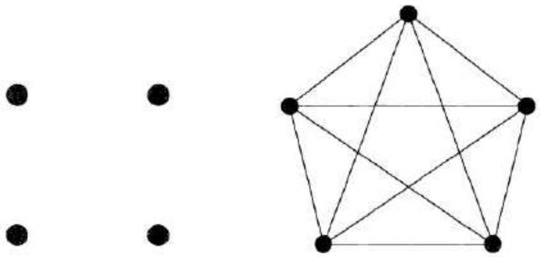
Topologies

- ➔ Null graph and fully connected graph



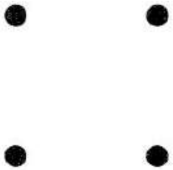
Topologies

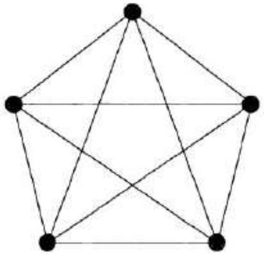
- ➔ Null graph and fully connected graph



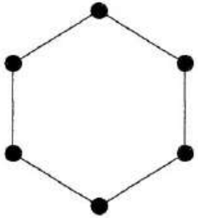
Topologies

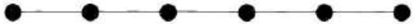
- ➔ Null graph and fully connected graph

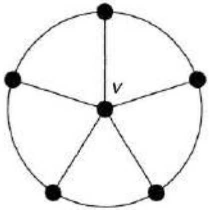




- ➔ Cycle graph, path and wheel

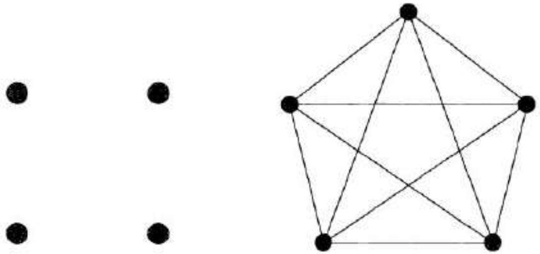




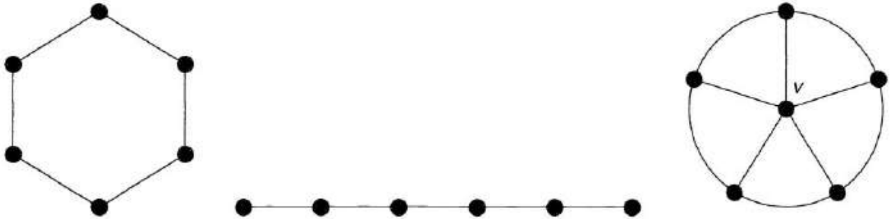


Topologies

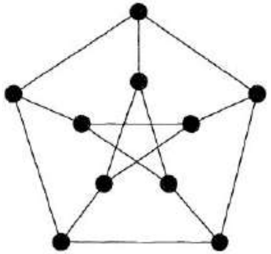
- ➔ Null graph and fully connected graph



- ➔ Cycle graph, path and wheel



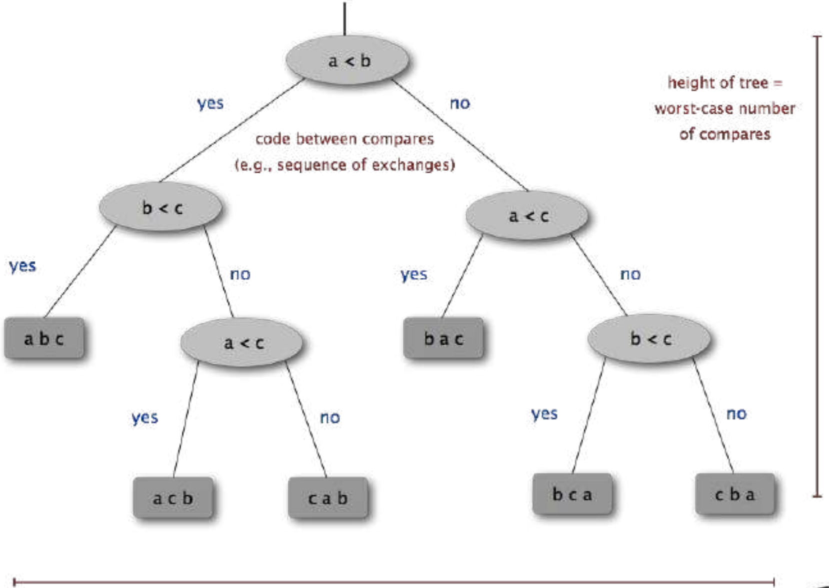
- ➔ Regular graphs (each vertex has the same *degree*)

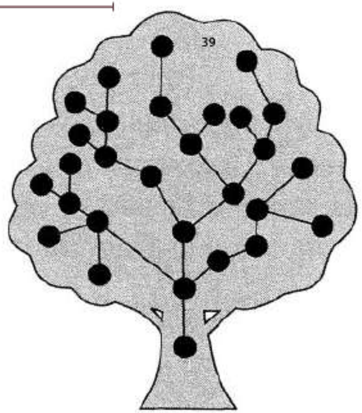


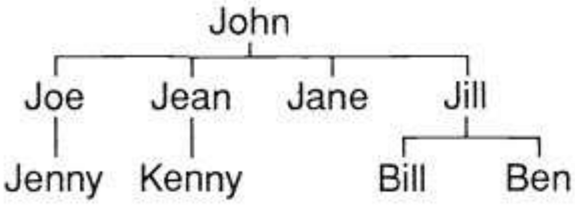
Topologies

- ➔ Tree – connected graph without cycles

- ➔ Random graphs
  - ◆ Erdos-Renyi,
  - ◆ Barabash-Albert (Scale free),
  - ◆ Watts-Strogatz (Small world)





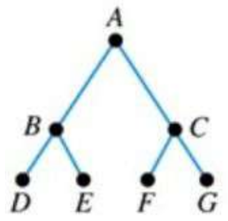


Topologies

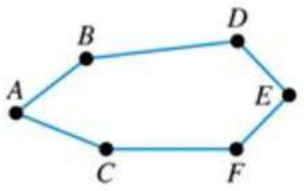
- ➔ Tree – connected graph with only one path between each pair of vertices

- ➔ Random graphs
  - ◆ Erdos-Renyi,
  - ◆ Barabash-Albert (Scale free),
  - ◆ Watts-Strogatz (Small world)

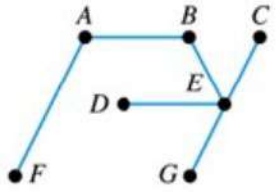
1.



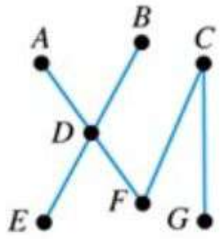
2.



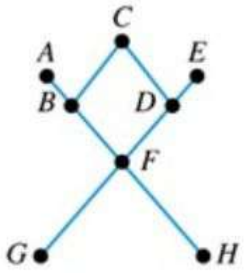
3.



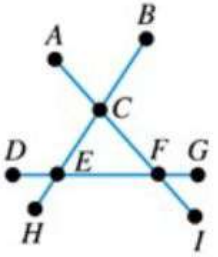
4.



5.



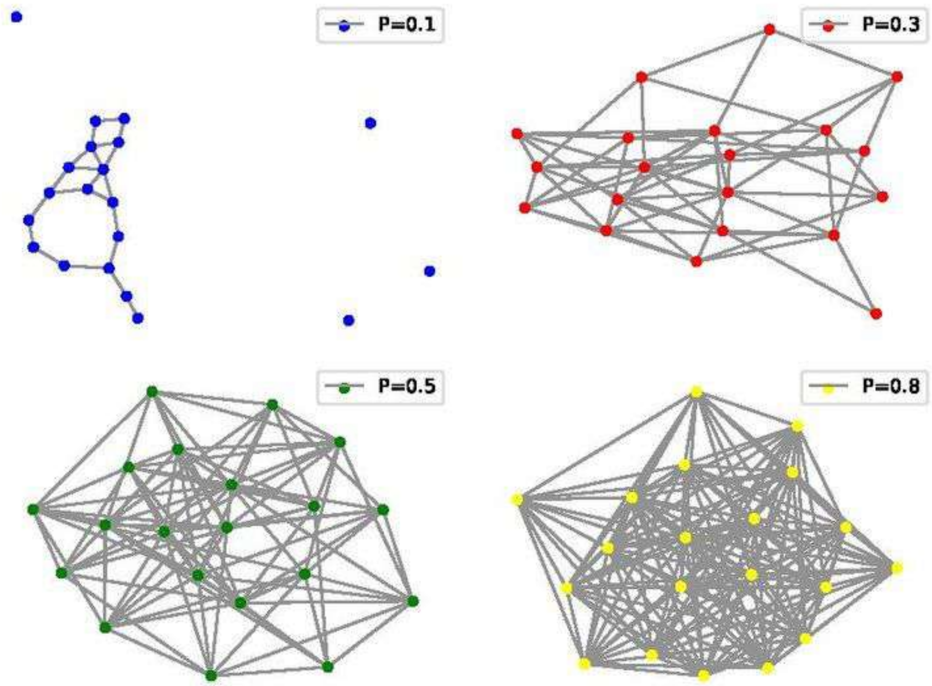
6.



Topologies

- Tree – connected graph with only one path between each pair of vertices

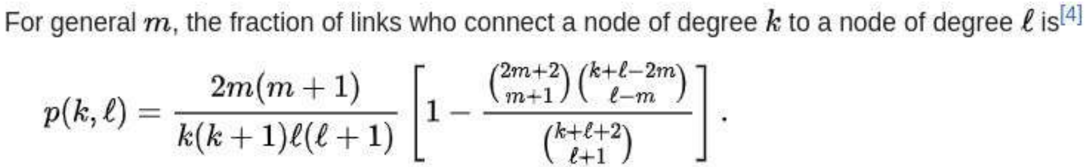
- Random graphs
  - Erdos-Renyi,
  - Barabash-Albert (Scale free),
  - Watts-Strogatz (Small world)



$$p^M (1-p)^{\binom{n}{2}-M}.$$

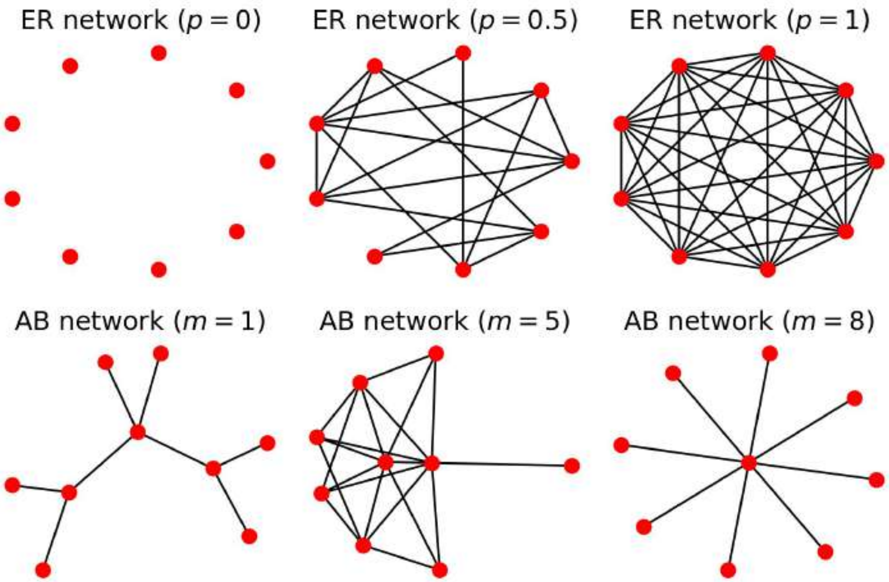
Topologies

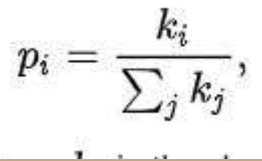
- ➔ Tree – connected graph with only one path between each pair of vertices
- ➔ Random graphs
  - ◆ Erdos-Renyi,
  - ◆ Barabash-Albert (Scale free),
  - ◆ Watts-Strogatz (Small world)



For general m, the fraction of links who connect a node of degree k to a node of degree ℓ is[4]

$$p(k,\ell) = \frac{2m(m+1)}{k(k+1)\ell(\ell+1)}\left[1 - \frac{\binom{2m+2}{m+1}\binom{k+\ell-2m}{\ell-m}}{\binom{k+\ell+2}{\ell+1}}\right].$$





$$p_i = \frac{k_i}{\sum_j k_j},$$

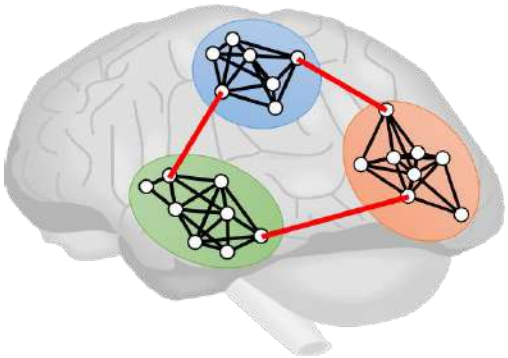
Important property: Heavily linked nodes ("hubs") tend to quickly accumulate even more links, while nodes with only a few links are unlikely to be chosen as the destination for a new link. The new nodes have a "preference" to attach themselves to the already heavily linked nodes.

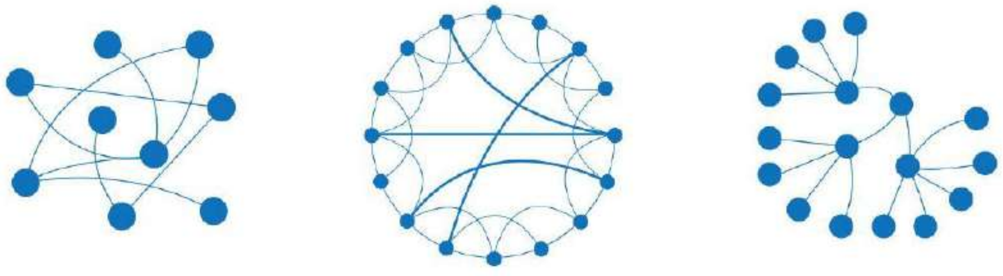
Topologies

- ➔ Tree – connected graph with only one path between each pair of vertices

- ➔ Random graphs
  - ◆ Erdos-Renyi,
  - ◆ Barabash-Albert (Scale free),
  - ◆ Watts-Strogatz (Small world)

Human brain structural and functional networks follow small-world configuration.



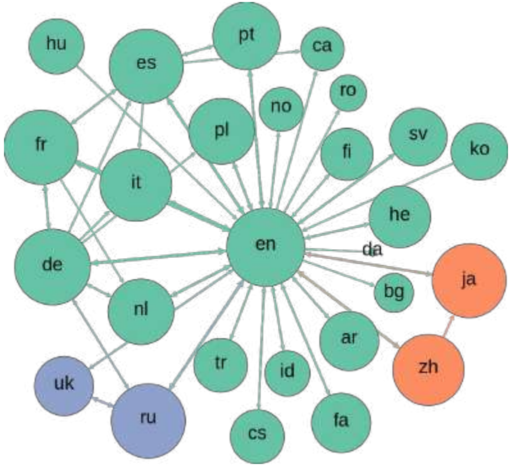


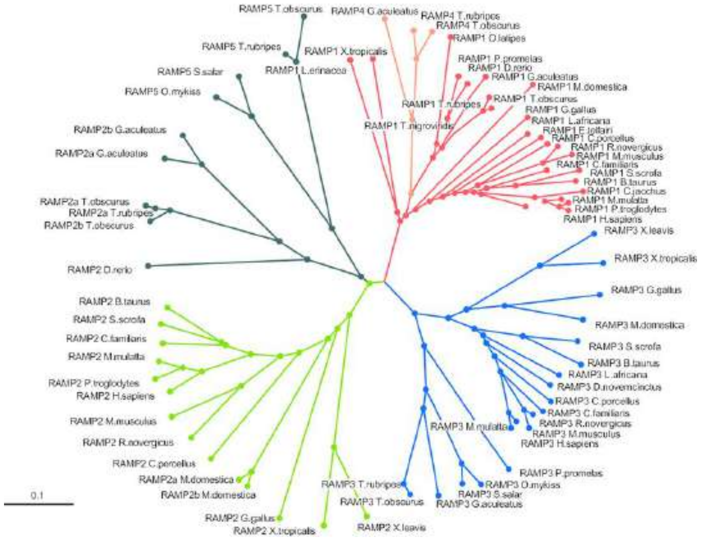
Random
Average distributions.
No structure or hierarchal patterns.

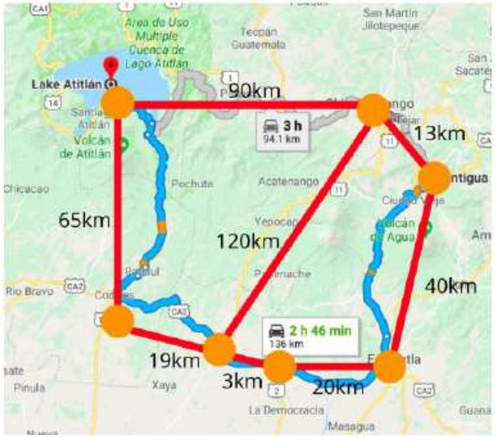
Small-World
High local clustering and short average path lengths.
Hub-and-spoke architecture.

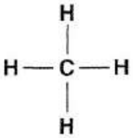
Scale-Free
Hub-and-spoke architecture preserved at multiple scales.
High power law distribution.

More examples:

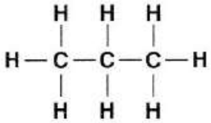




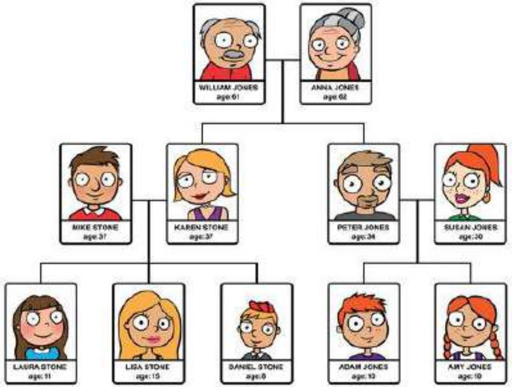


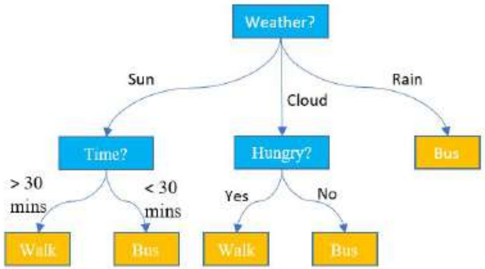


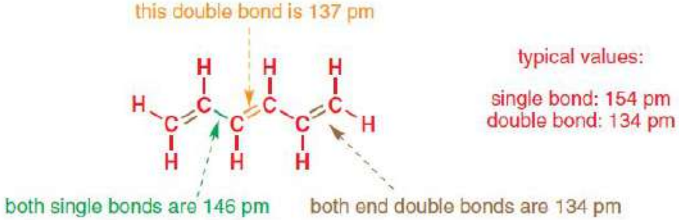
methane



propane







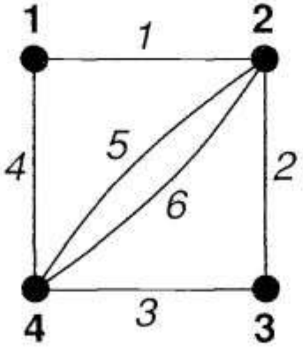
this double bond is 137 pm
typical values:
single bond: 154 pm
double bond: 134 pm
both single bonds are 146 pm
both end double bonds are 134 pm

Graph presentation

- Diagram
- List of edges:

  [(1, 2), (2, 3), (3, 4), (1, 4), (2, 4), (2, 4)]

- Adjacency and incident matrices:

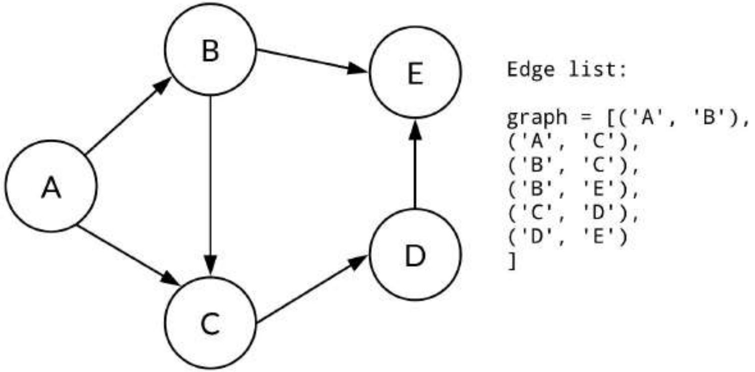


$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

size(A) = (#v, #v)
A_{ij} = #e between v_i and v_j

size(M) = (#v, #e)
M_{ij} = 1 if e_j from v_i exist

Graph presentation



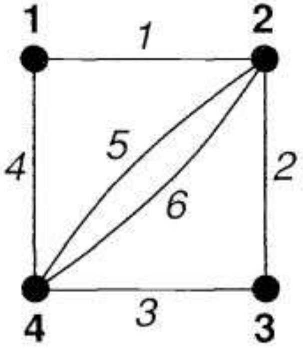
```
Edge list:

graph = [('A', 'B'),
('A', 'C'),
('B', 'C'),
('B', 'E'),
('C', 'D'),
('D', 'E')
]
```

- ➔ Diagram
- ➔ List of edges:

  [(1, 2), (2, 3), (3, 4), (1, 4), (2, 4), (2, 4)]

- ➔ Adjacency and incident matrices:



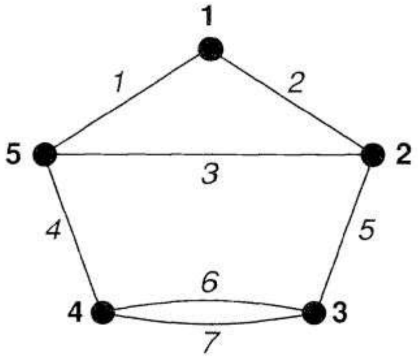
$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

size(A) = (#v, #v)
A_{ij} = #e between v_i and v_j

size(M) = (#v, #e)
M_{ij} = 1 if e_j from v_i exist

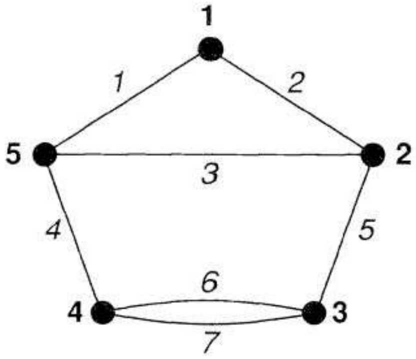
Graph presentation: your turn!

- ➔ Diagram
- ➔ List of edges: [?]
- ➔ Adjacency and incident matrices:



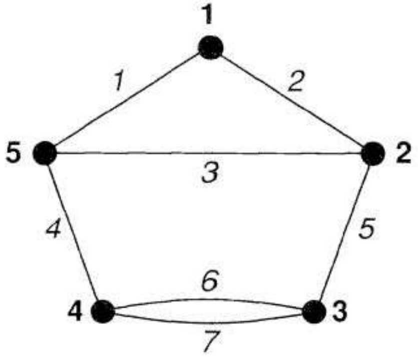
Graph presentation: your turn!

- ➔ Diagram
- ➔ List of edges: [(1, 5), (1, 2), (2, 5), (4, 5), (2, 3), (3, 4), (3, 4)]
- ➔ Adjacency and incident matrices:



Graph presentation: your turn!

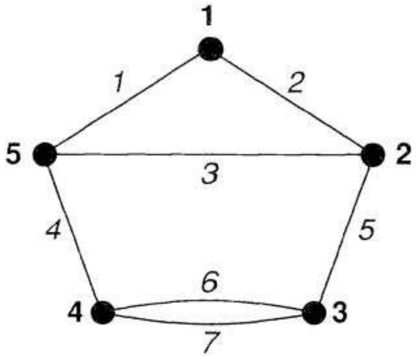
- ➔ Diagram
- ➔ List of edges: [(1, 5), (1, 2), (2, 5), (4, 5), (2, 3), (3, 4), (3, 4)]
- ➔ Adjacency and incident matrices:



$$\begin{pmatrix} 0 & 1 & 1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Graph presentation: your turn!

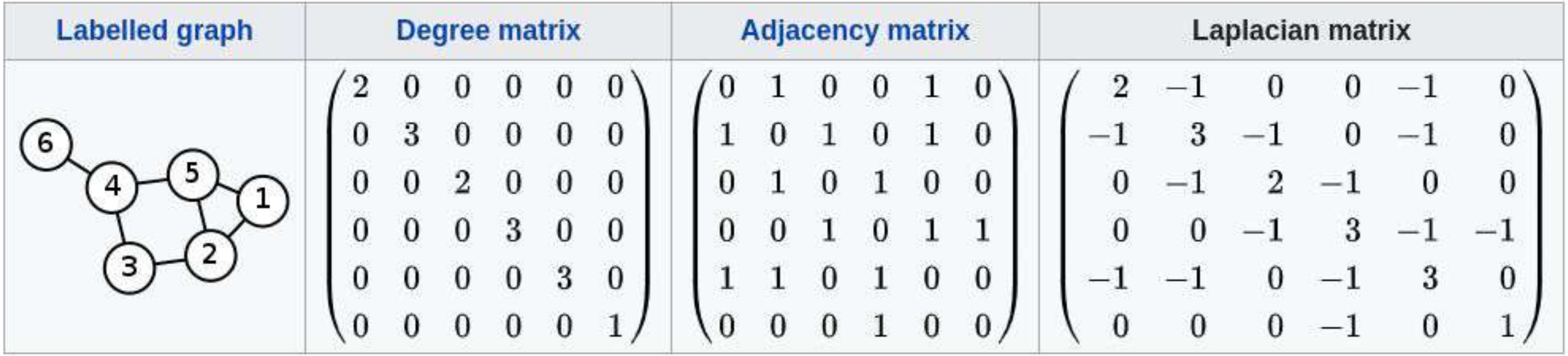
- Diagram
- List of edges: [(1, 5), (1, 2), (2, 5), (4, 5), (2, 3), (3, 4), (3, 4)]
- Adjacency and incident matrices:



$$\begin{pmatrix} 0 & 1 & 1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Laplacian matrix



Labelled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

L = D - A,

L := Laplacian matrix
D := degree matrix
A := adjacency matrix

Eigenvalues

$$A = \begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix}$$

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

$$\det(A - \lambda I) = 0$$

$$\det\left(\begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$\det\left(\begin{bmatrix} 1 - \lambda & 4 \\ 3 & 2 - \lambda \end{bmatrix}\right) = 0$$

$$(1 - \lambda)(2 - \lambda) - 12 = 0$$

$$\lambda^2 - 3\lambda - 10 = 0$$

$$(\lambda - 5)(\lambda + 2) = 0$$

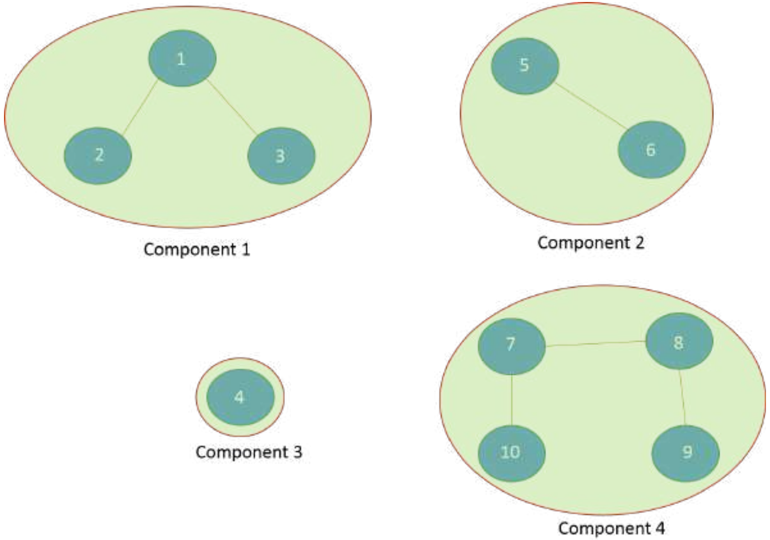
$$\lambda = 5, -2$$

Graph characteristics

1. Connected components

subgraphs where any two vertices are connected by paths, and which are connected to no additional vertices in the rest of the graph

Number of connected components equals to the ***number of eigenvalues = 0*** *in the Laplacian matrix*



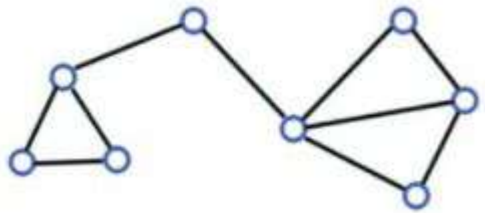
Graph characteristics

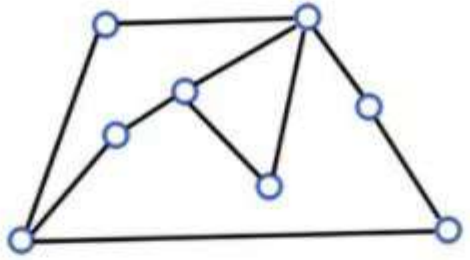
2. Algebraic connectivity

- *reflects how well connected the overall graph is*
- *how easy this network goes to synchronization (!)*

We can calculate it as the ***second-smallest eigenvalue*** *of the* ***Laplacian*** *matrix*

Alg. connectivity = 0.238



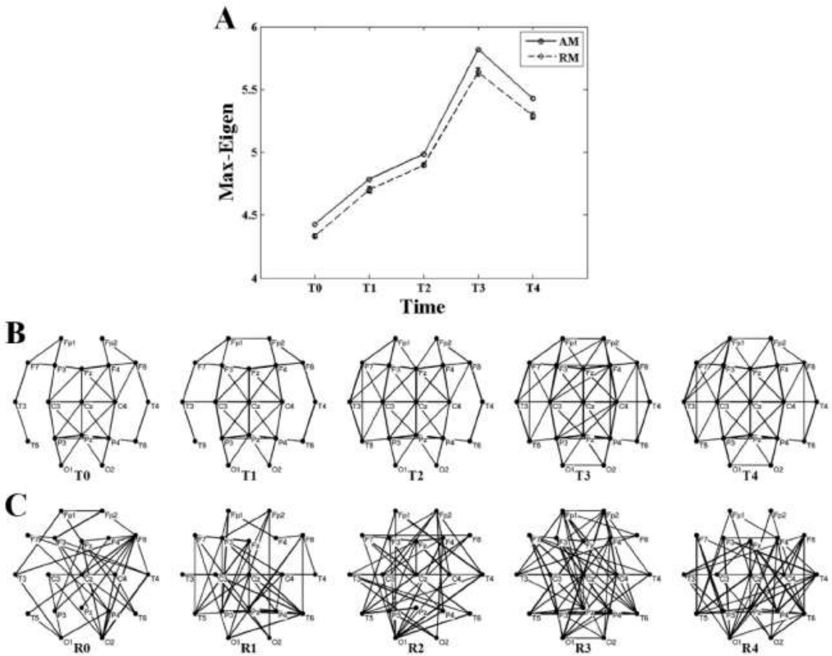


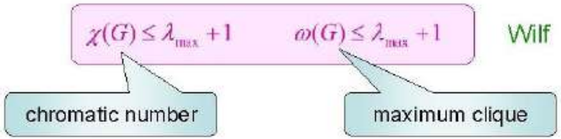
Alg. connectivity = 0.925

Graph characteristics

3. Maximum eigenvalue of adjacency matrix

- *a measure of how small changes to the graph structure influence flows on the graph,*
- *defines the transition to synchronization,*
- *important in percolation on directed networks*





$\chi(G) \le \lambda_{\max} + 1$ $\quad$ $\omega(G) \le \lambda_{\max} + 1$ $\quad$ Wilf

chromatic number

maximum clique

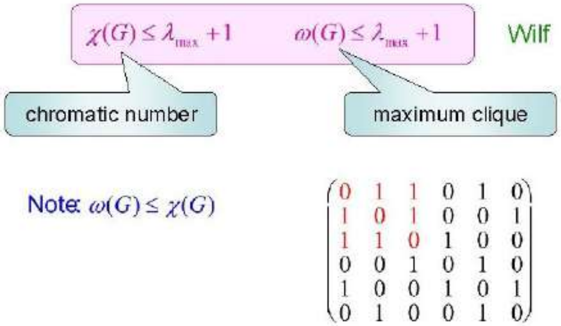
Note: $\omega(G) \le \chi(G)$

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Graph characteristics

3. Maximum eigenvalue of adjacency matrix

- *a measure of how small changes to the graph structure influence flows on the graph,*
- *defines the transition to synchronization,*
- *important in percolation on directed networks*

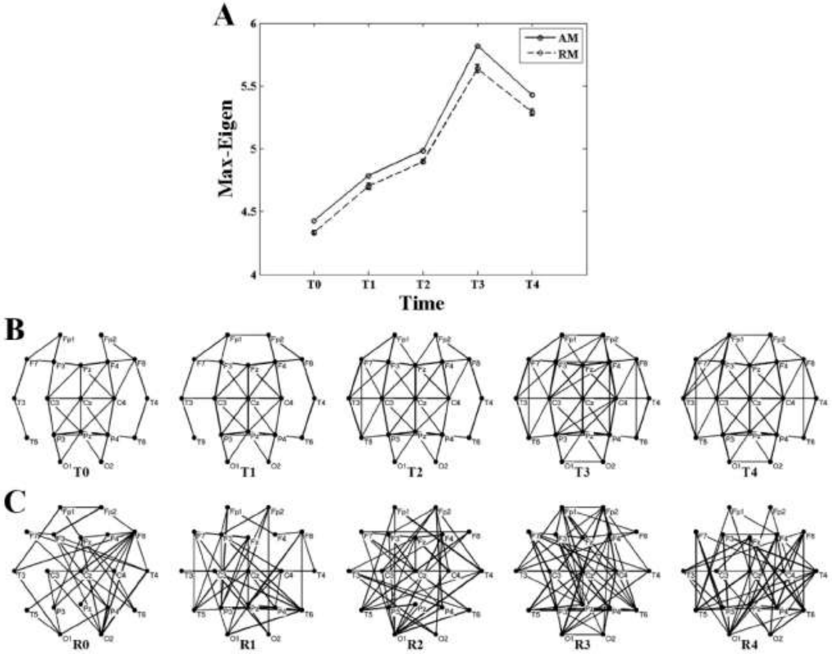


$\chi(G) \le \lambda_{\max} + 1$ $\qquad$ $\omega(G) \le \lambda_{\max} + 1$ $\qquad$ Wilf

chromatic number $\qquad$ maximum clique

Note: $\omega(G) \le \chi(G)$

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



- For a large class of networks, there is a transition to synchrony at a critical coupling constant determined by the maximum eigenvalue of the adjacency matrix.
- A larger maximum eigenvalue of the adjacency matrix favors a lower threshold for synchronization.
- Heterogeneity in the degree distribution, randomness in the couplings, and positive degree correlations favors synchronization.

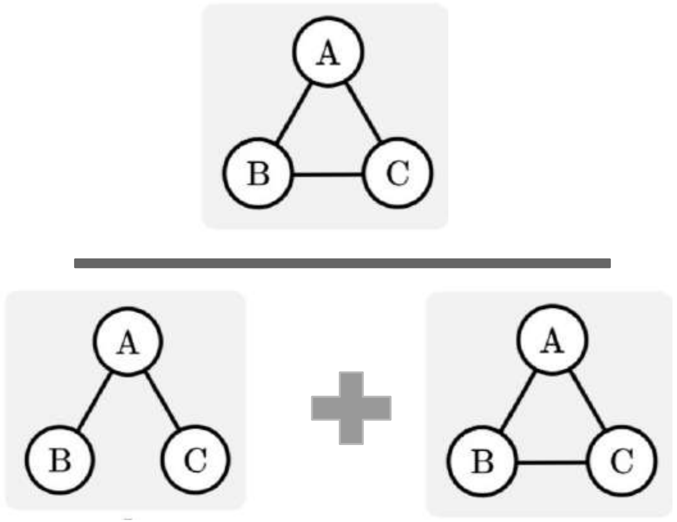
Our papers can be obtained from: http://www.chaos.umd.edu/umdsyncnets.html

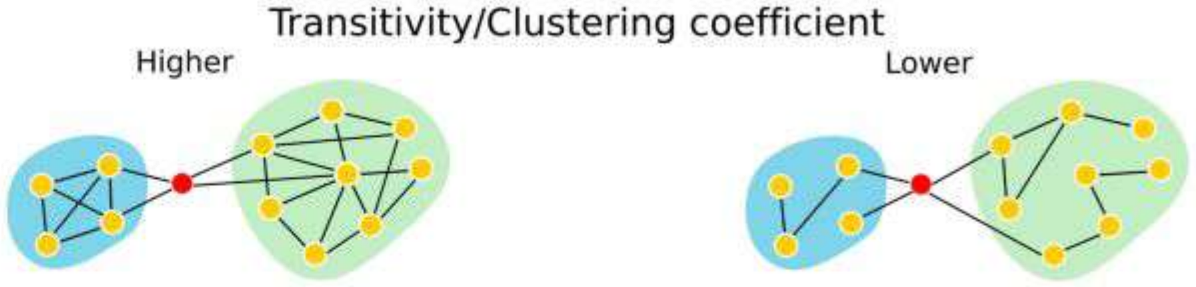
Graph characteristics

4. Transitivity

the relative number of triangles, compared to the number of triades

represent How dense the network is

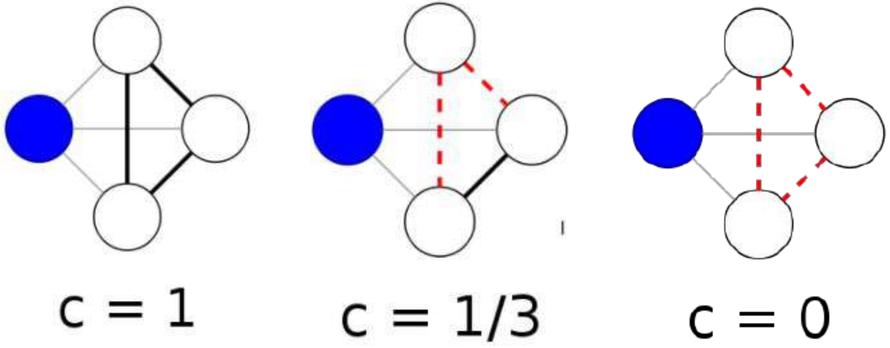




Graph characteristics

5. Average clustering measure of the degree to which nodes in a graph tend to cluster together

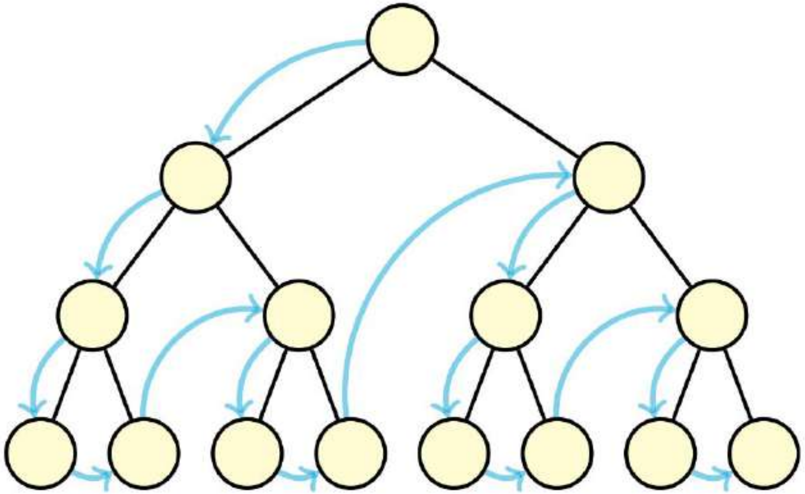
a proportion of the number of links between the vertices within its neighbourhood divided by the number of links that could possibly exist between them.



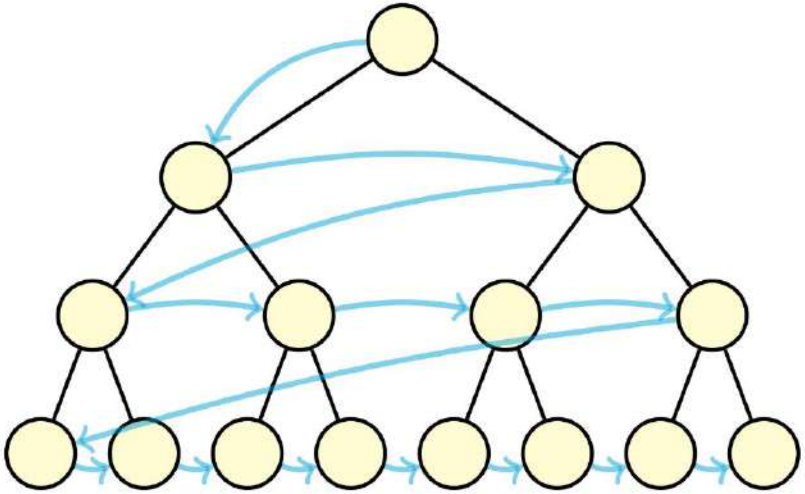
Graph characteristics

1. Connected components *subgraphs where any two vertices are connected by paths, and which is connected to no additional vertices in the rest of the graph*
2. Algebraic connectivity *reflects how well connected the overall graph is*
3. Maximum eigenvalue of adjacency matrix *a measure of how small changes to the graph structure influence flows on the graph*
4. Transitivity *the relative number of triangles, compared to the number of triades*
5. Average clustering measure of the degree to which nodes in a graph tend to cluster together
6. Chromatic number measure of criticality

Popular problems and algorithms



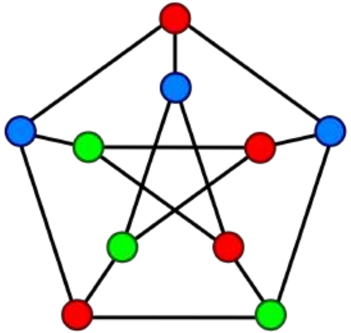
Depth-First Search



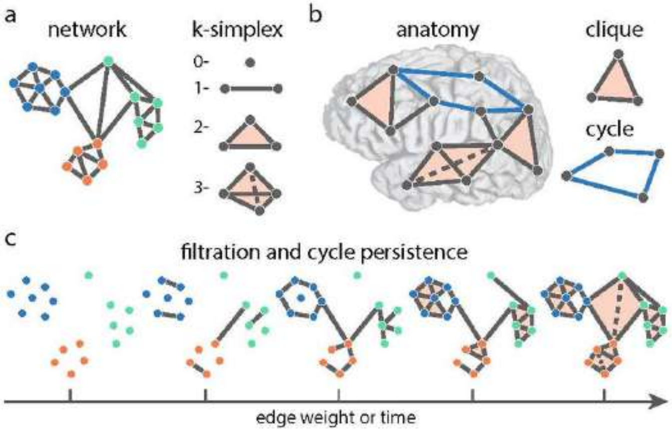
Breadth-First Search

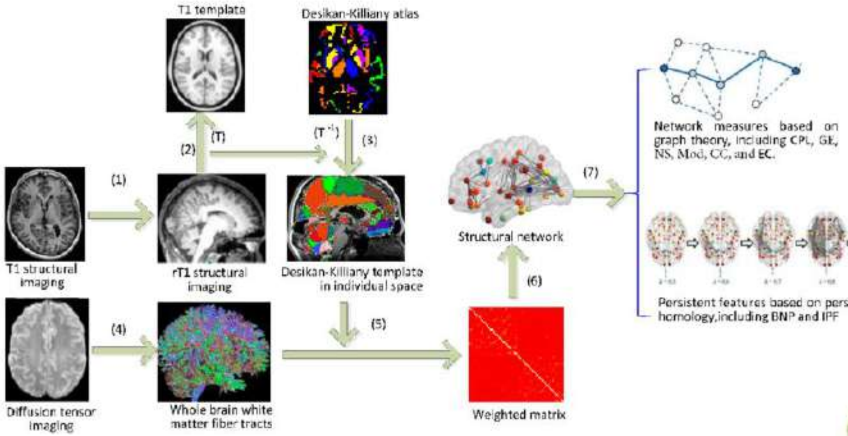
Popular problems and algorithms

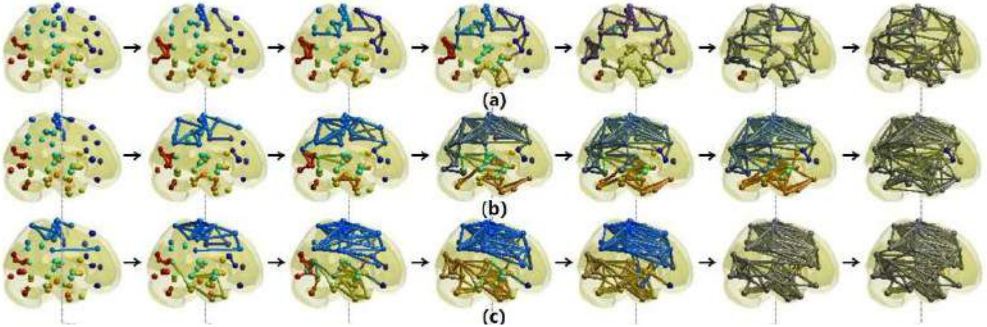
- Shortest path problem:
  - Dijkstra’s algorithm,
  - Bellman–Ford algorithm
- Travelling salesman problem:
  - Exact algorithms,
  - Nearest neighbour algorithm (greedy algorithms),
  - Ant colony optimization
  - etc.

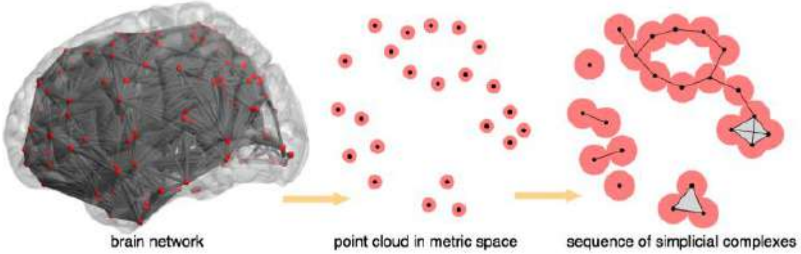


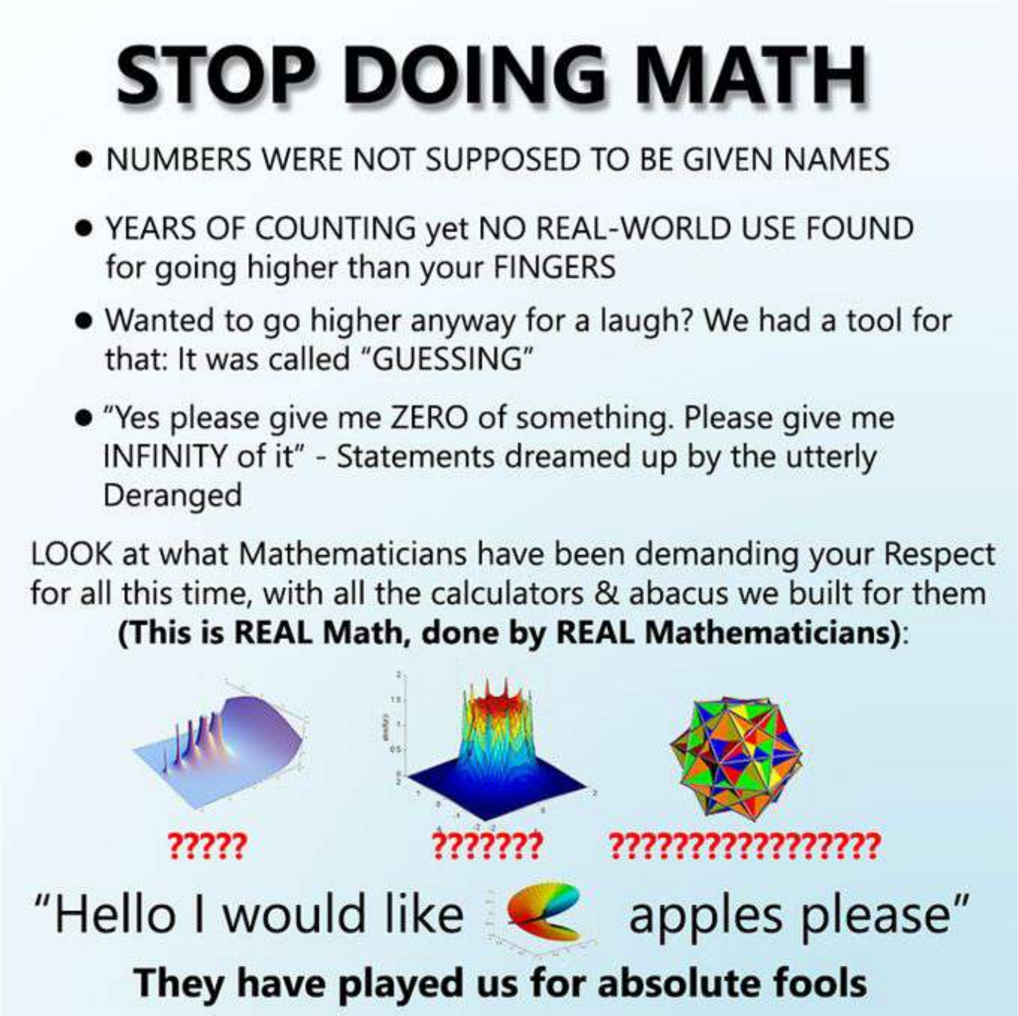
Graph theory for neuroscience




T1 template
Desikan-Killiany atlas
(2)
(T)
(T⁻¹)
(3)
(1)
T1 structural imaging
rT1 structural imaging
Desikan-Killiany template in individual space
Structural network
(7)
Network measures based on graph theory, including CPL, GE, NS, Mod, CC, and EC.
Persistent features based on persistent homology,including BNP and IPF
(6)
(4)
(5)
Diffusion tensor imaging
Whole brain white matter fiber tracts
Weighted matrix







STOP DOING MATH

- NUMBERS WERE NOT SUPPOSED TO BE GIVEN NAMES
- YEARS OF COUNTING yet NO REAL-WORLD USE FOUND for going higher than your FINGERS
- Wanted to go higher anyway for a laugh? We had a tool for that: It was called "GUESSING"
- "Yes please give me ZERO of something. Please give me INFINITY of it" - Statements dreamed up by the utterly Deranged

LOOK at what Mathematicians have been demanding your Respect for all this time, with all the calculators & abacus we built for them **(This is REAL Math, done by REAL Mathematicians)**:

????? ??????? ?????????????????

"Hello I would like apples please"

They have played us for absolute fools

Let’s practice!