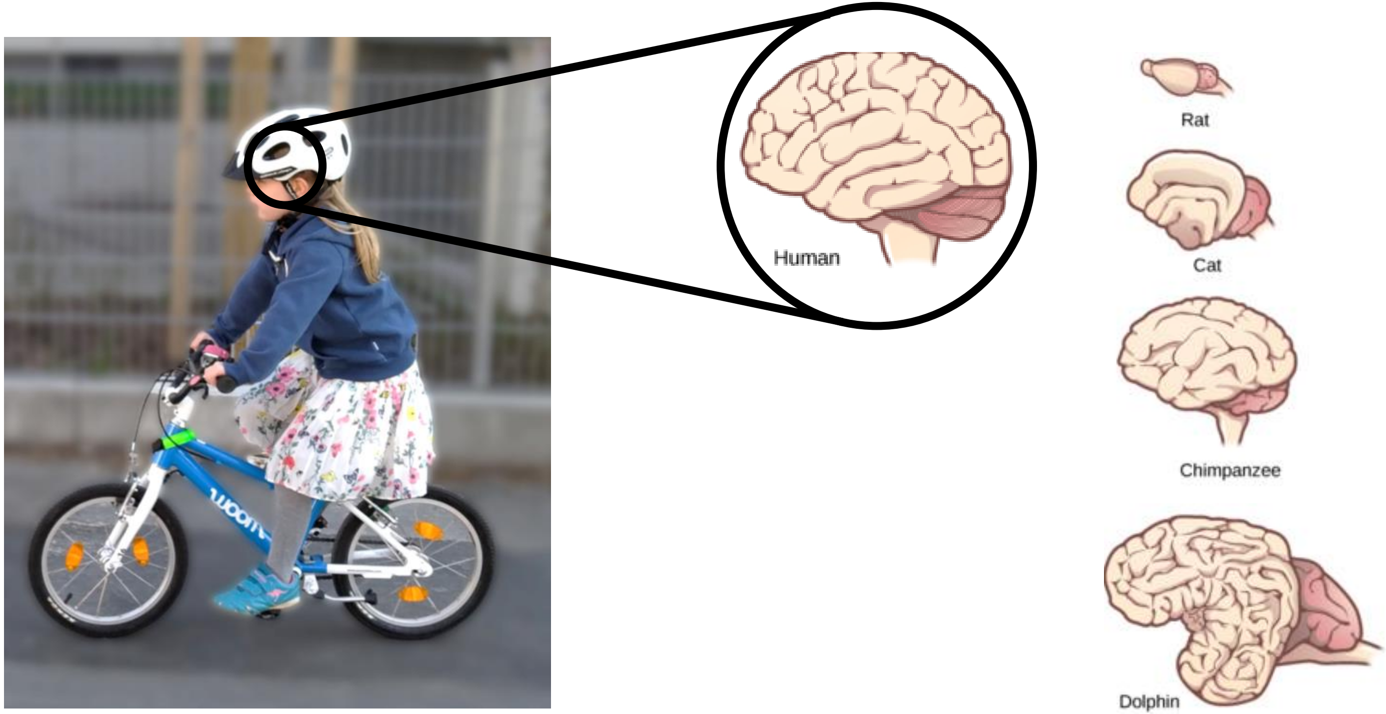
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Cognitive Neuroscience for AI Developers

Structure and Function of the Nervous System: Neurons and Glia

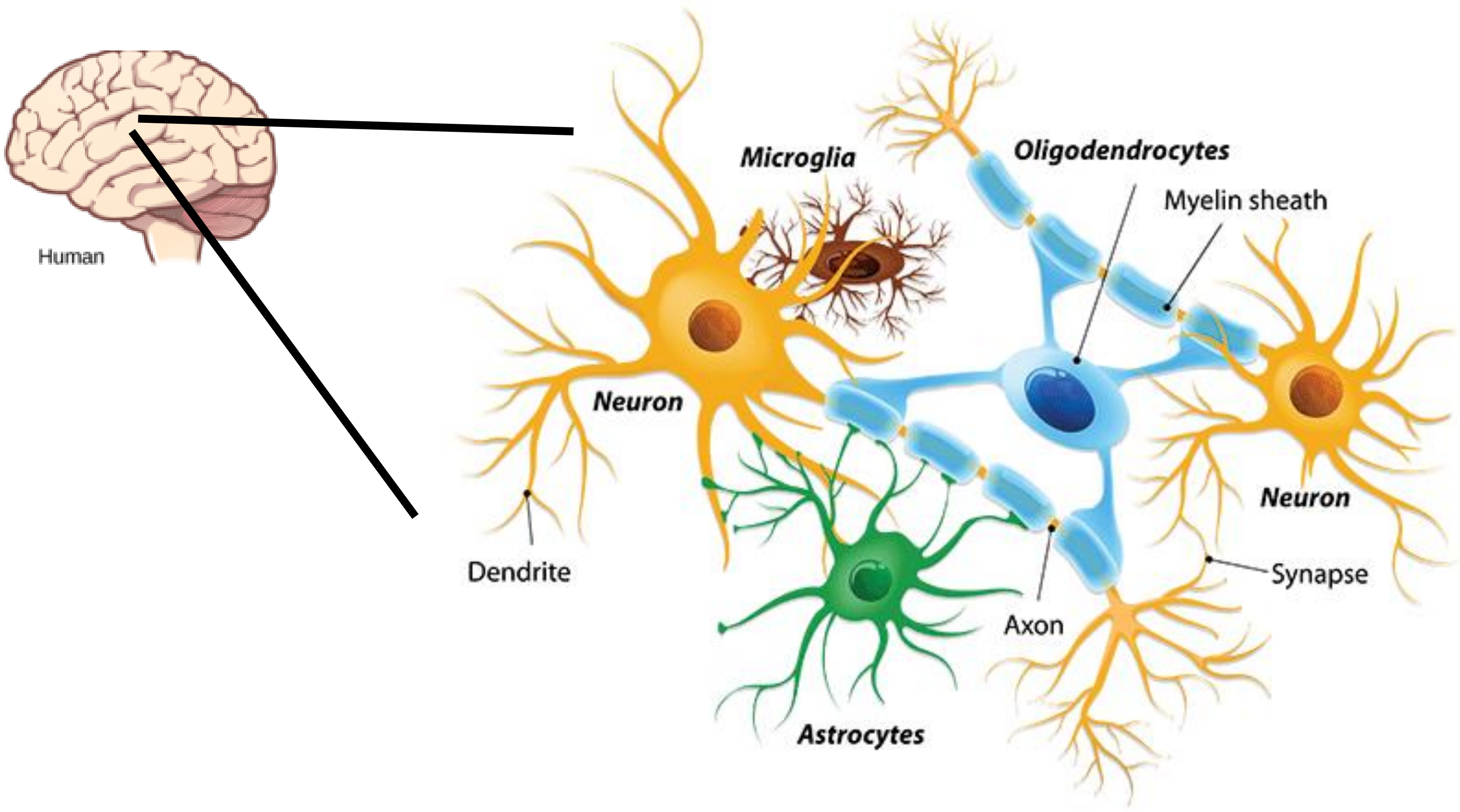
SENSORY → THE BRAIN → BEHAVIOR



https://wiki2.org/en/Brain_size

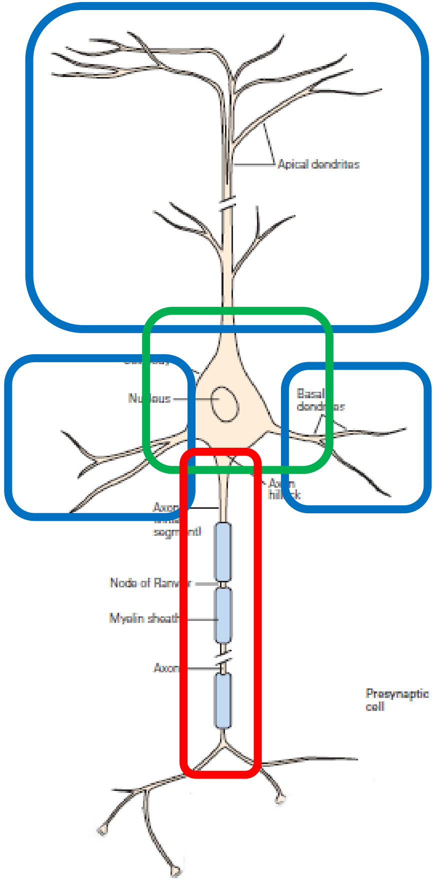
But what have all these brains in common?

CELLS.



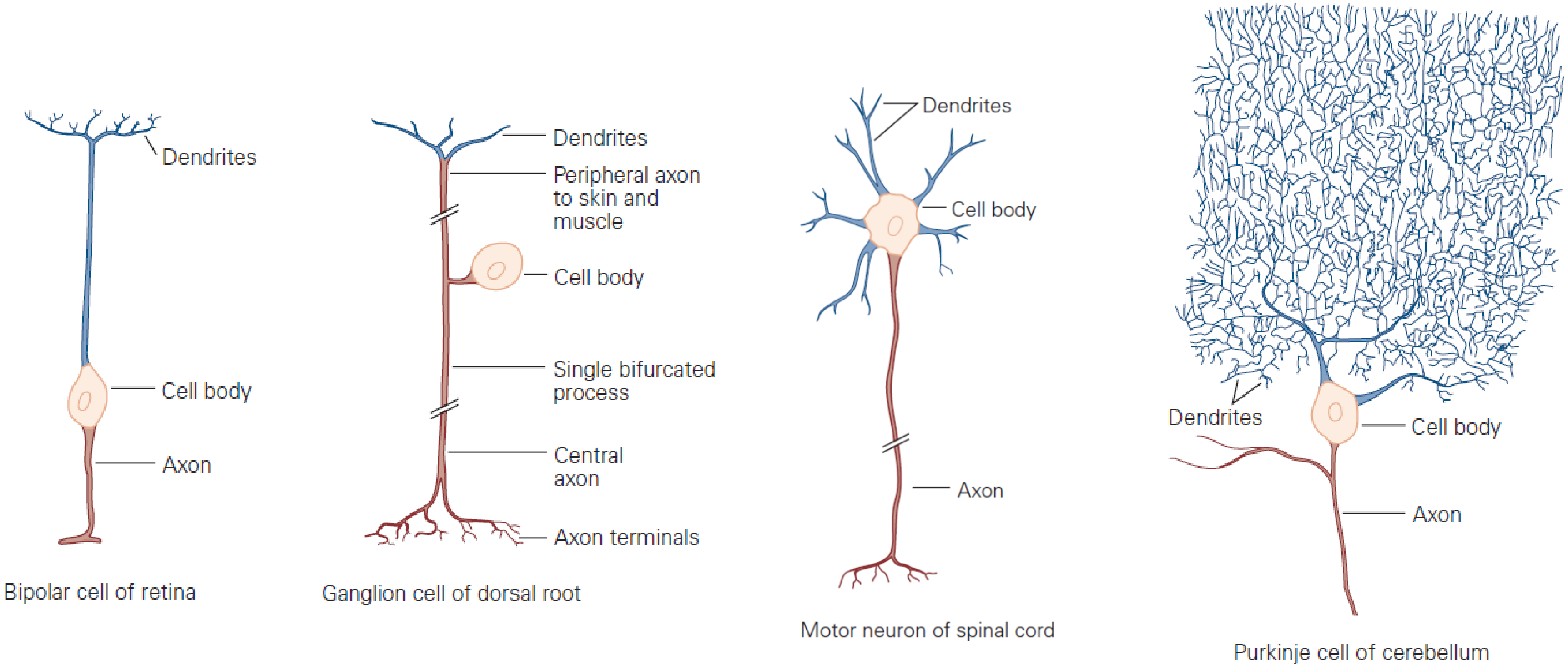
https://www.sciencenewsforstudents.org/article/scientists-say-glia

A neuron

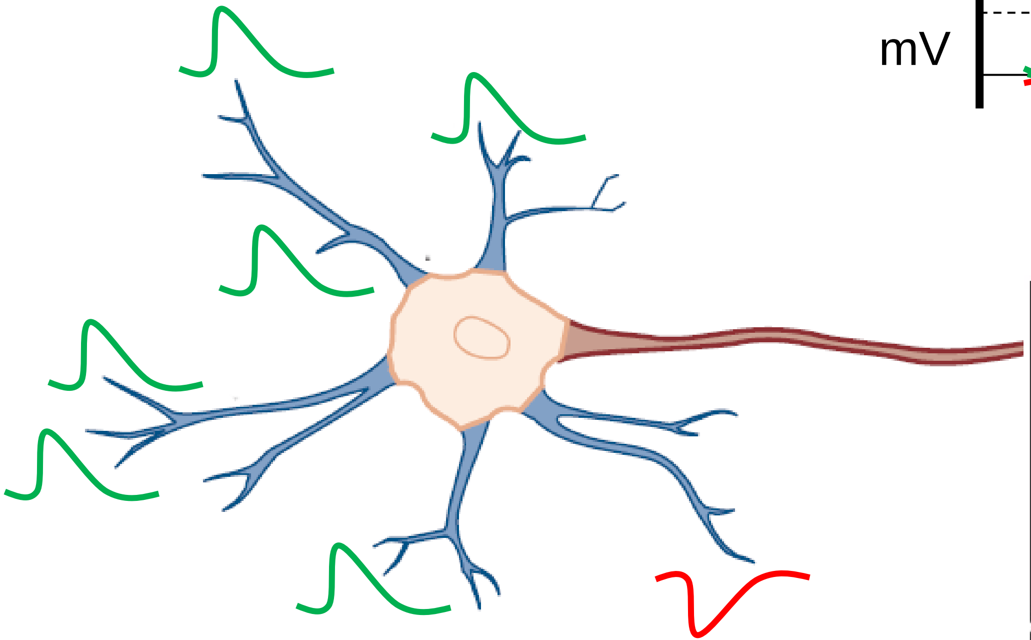


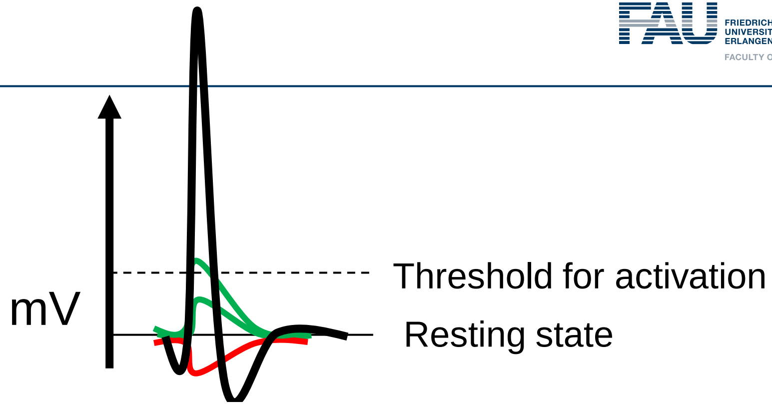
Apical dendrites
Nucleus
Basal dendrites
Axon hillock
Axon initial segment
Node of Ranvier
Myelin sheath
Axon
Presynaptic cell

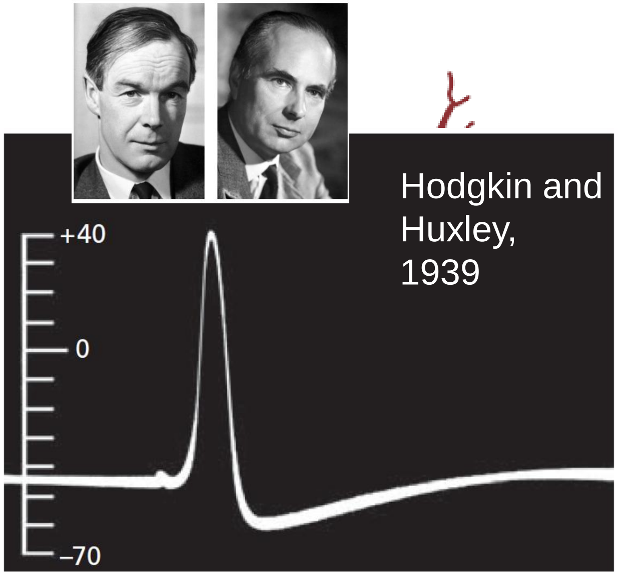
Neuron types



Neural computation





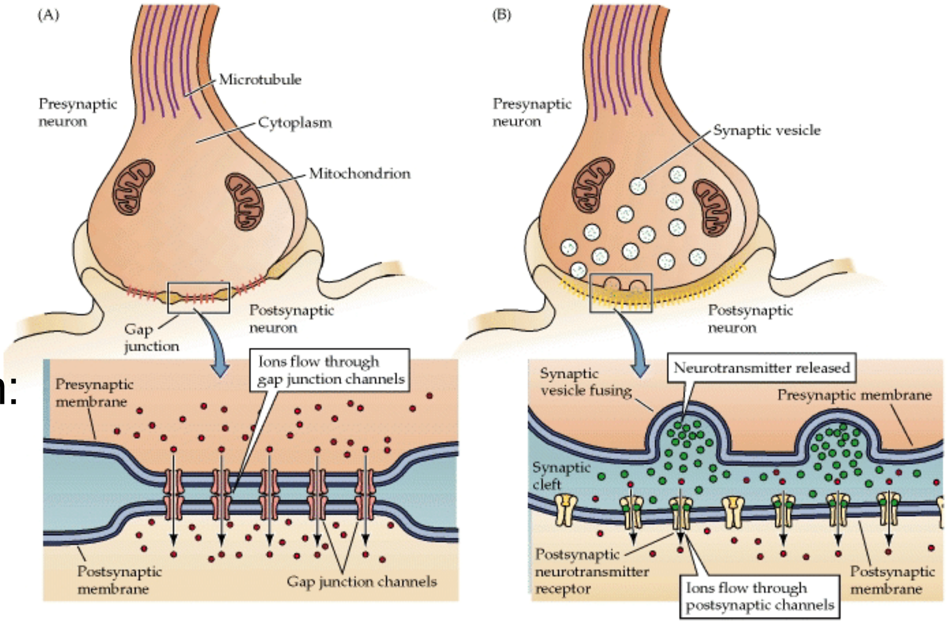


How do we transport information?

https://www.nanotec.com/eu/en/products/283-connection-cables

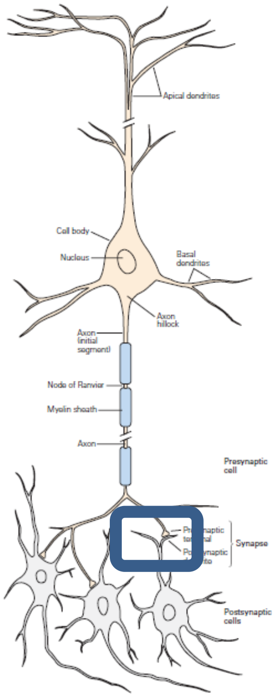
Electrical vs chemical coupling

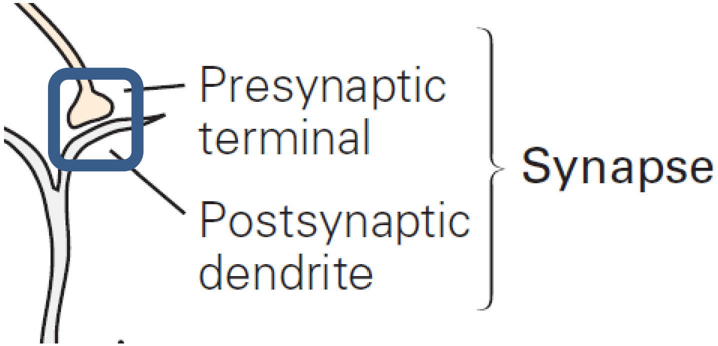
Electrical connection:
Immediate,
Same sign
Less modifiable

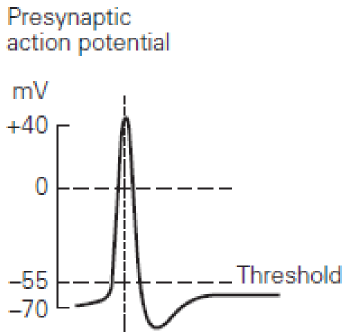


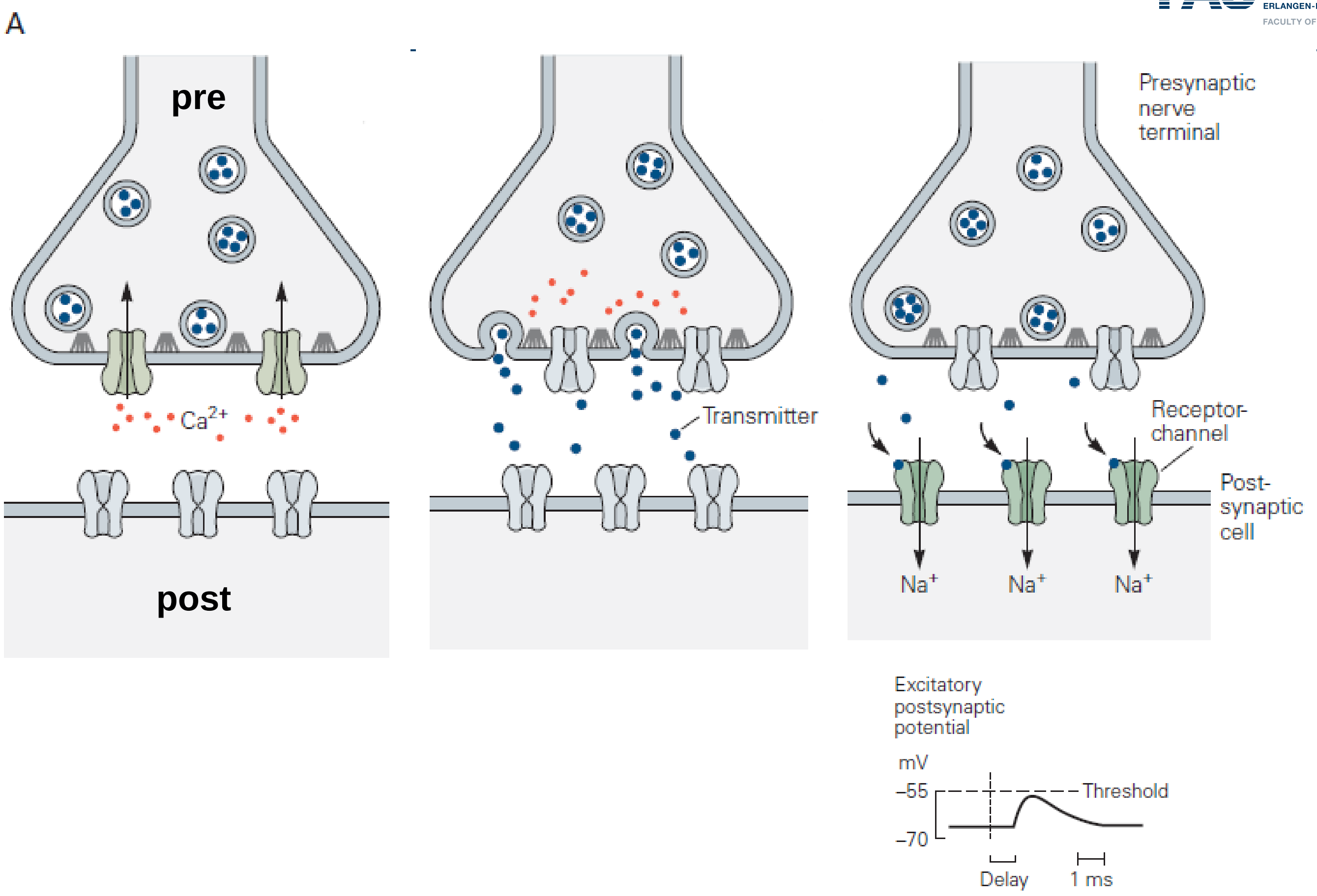
Chemical connection:
Delay,
Any sign
Very nice modifiable

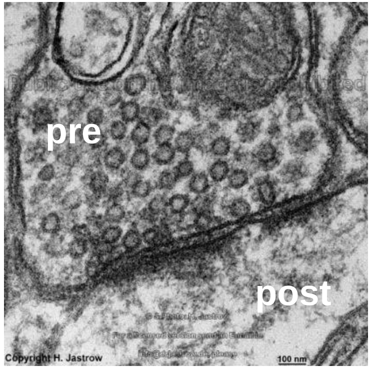
The connection between two neurons





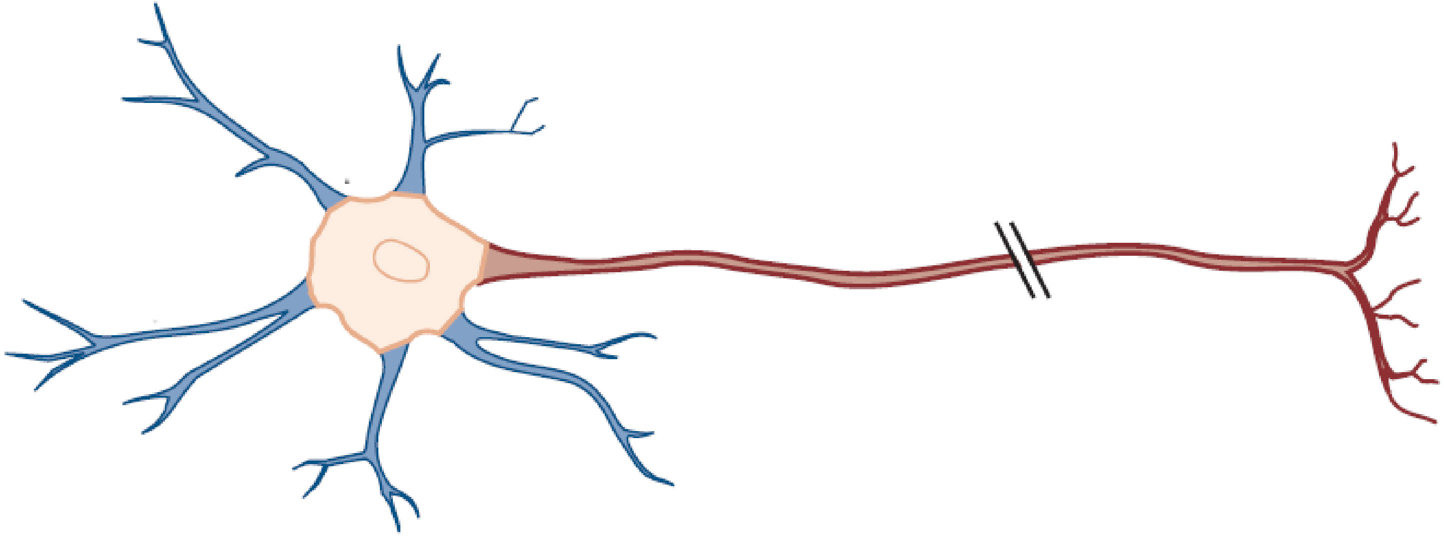




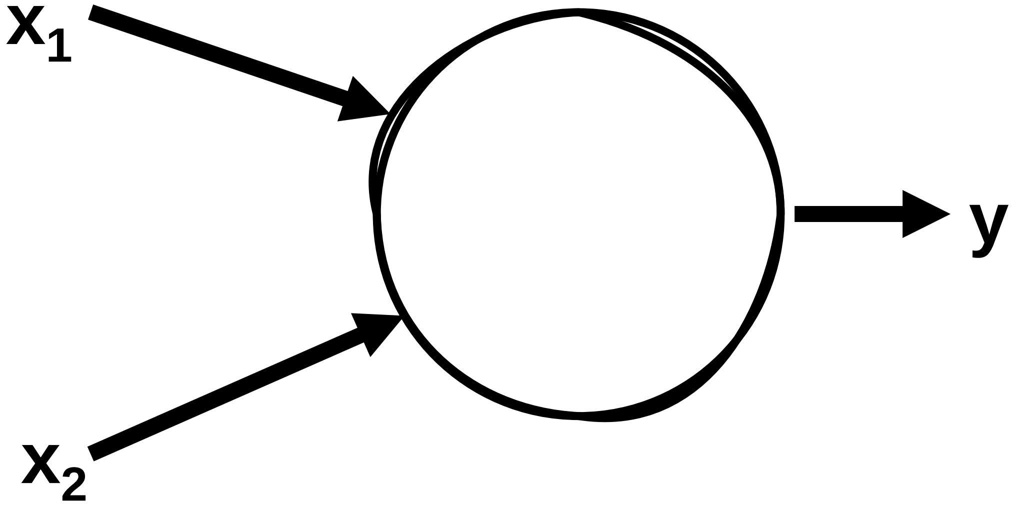


http://www.drjastrow.de/WAI/EM/EMSynapse.html

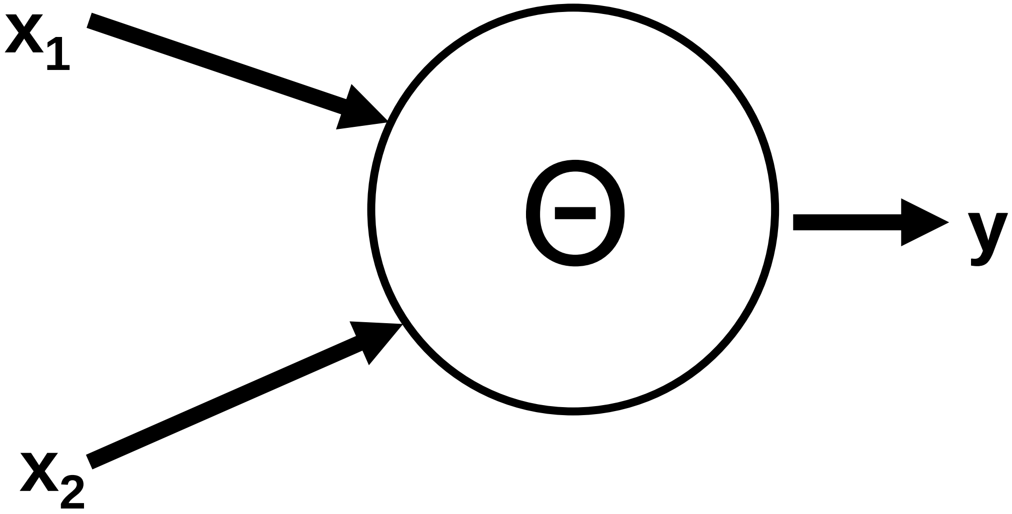
Mathematically simplifying a neuron

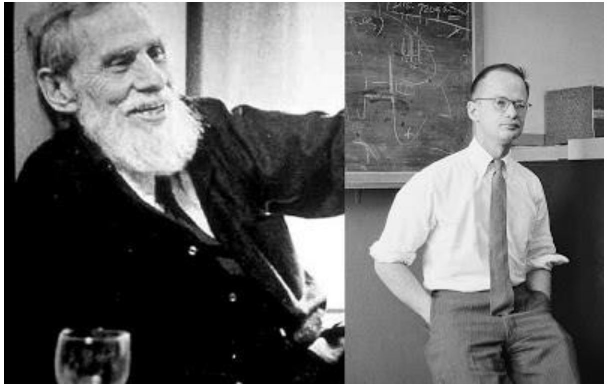


A theoretical neuron

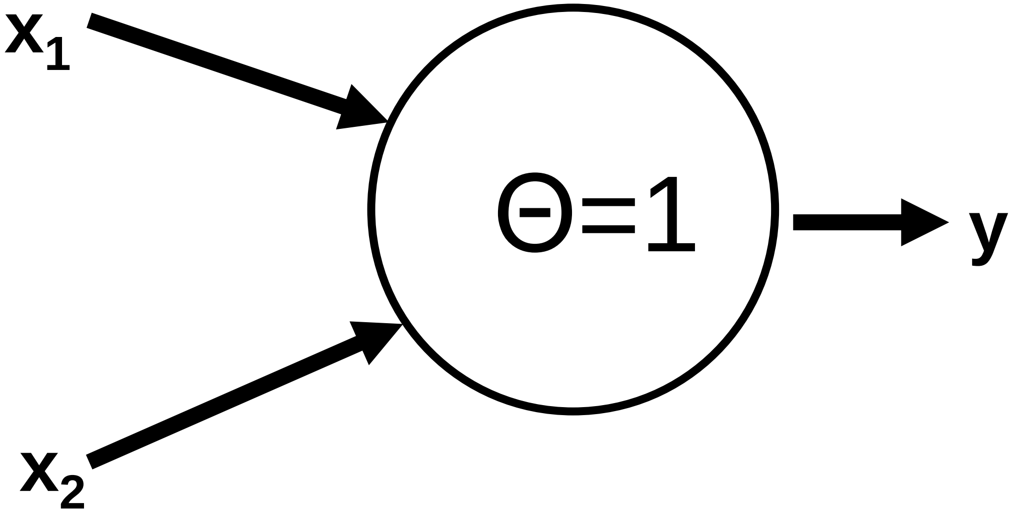


The McCulloch-Pitts neuron





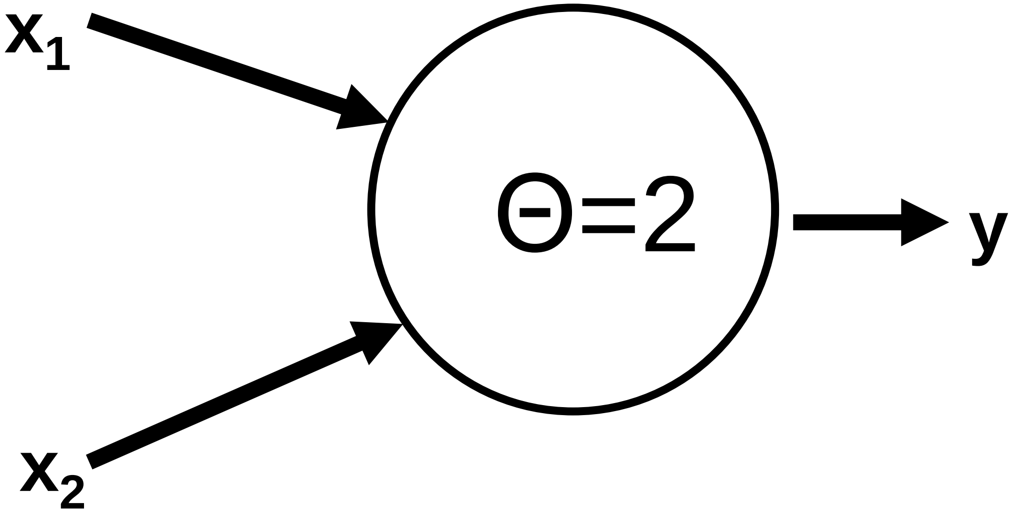
Logical operations with the McCulloch-Pitts neuron



x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1

LOGICAL OR

Logical operations with the McCulloch-Pitts neuron



x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1

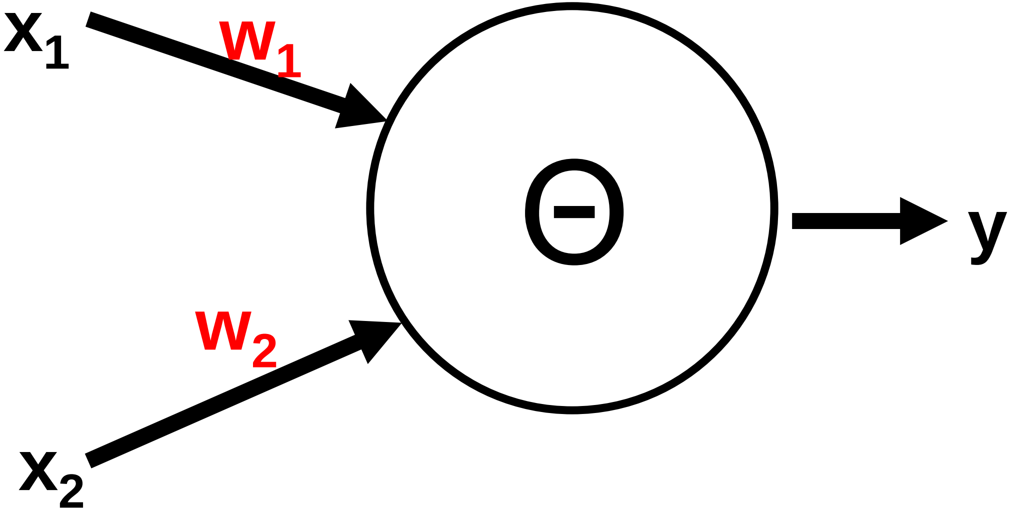
LOGICAL AND

McCulloch Pitts Neuron
(assuming no inhibitory inputs)

$$y = 1 \quad if \sum_{i=0}^{n} x_i \geq \Theta$$
$$= 0 \quad if \sum_{i=0}^{n} x_i < \Theta$$

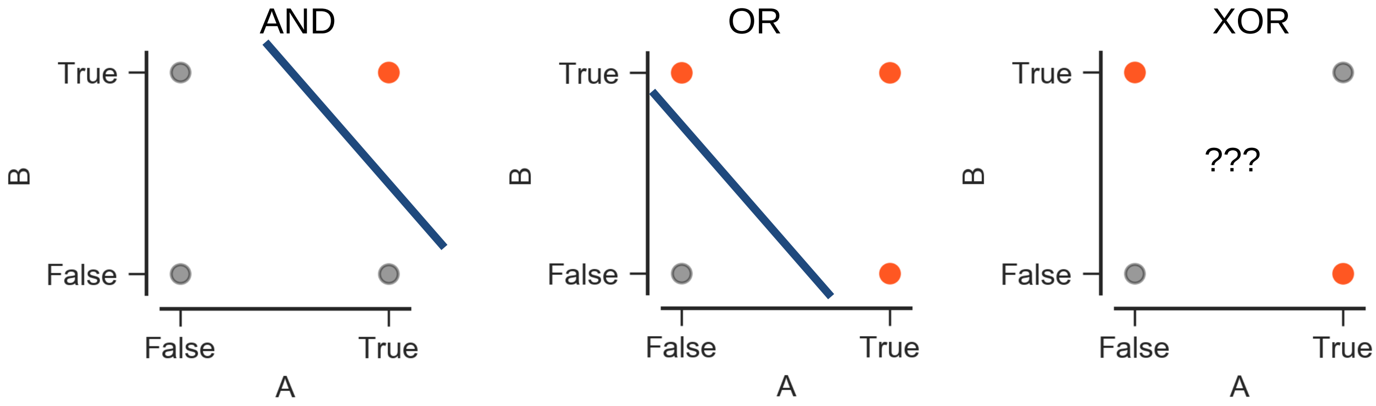
Perceptron

$$y = 1 \quad if \sum_{i=0}^{n} w_i * x_i \geq \Theta$$
$$= 0 \quad if \sum_{i=0}^{n} w_i * x_i < \Theta$$



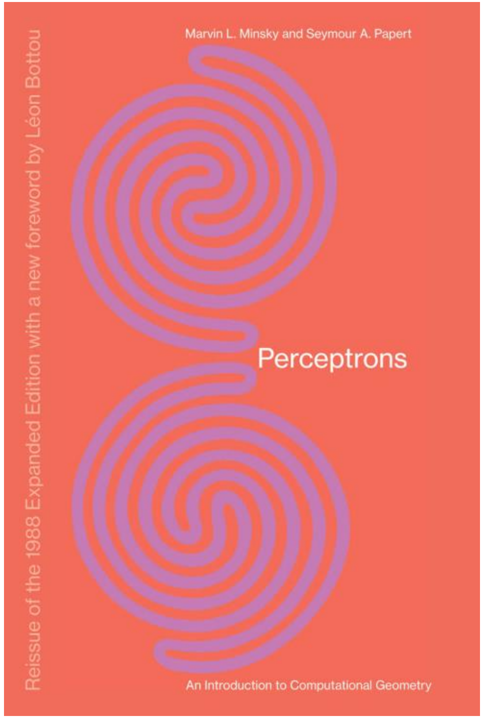
https://towardsdatascience.com/perceptron-the-artificial-neuron-4d8c70d5cc8d

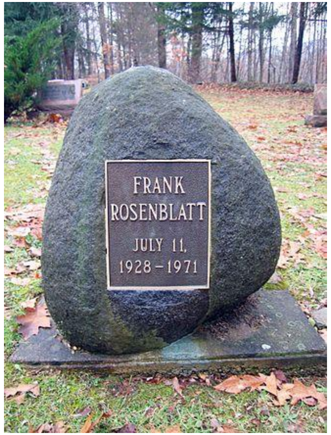
The XOR affair



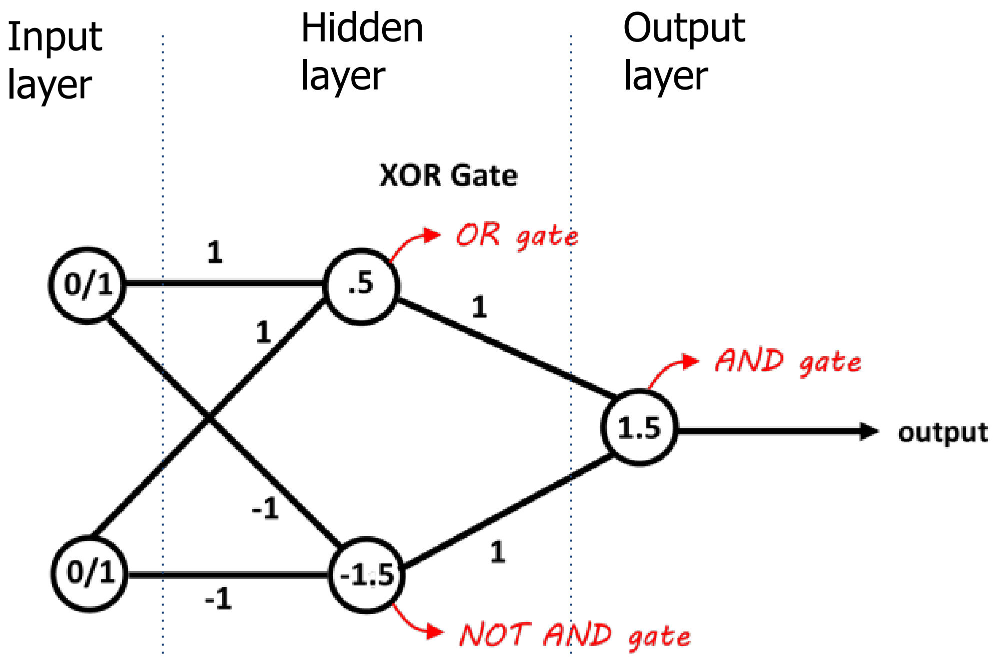
$\Theta=2$ $\Theta=1$ $\Theta=?$

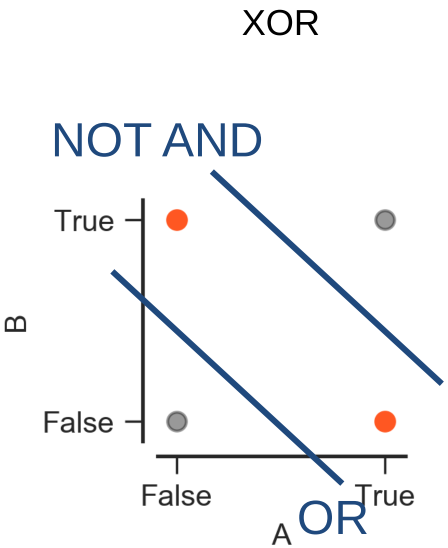
The XOR affair in
Perceptrons (1969)



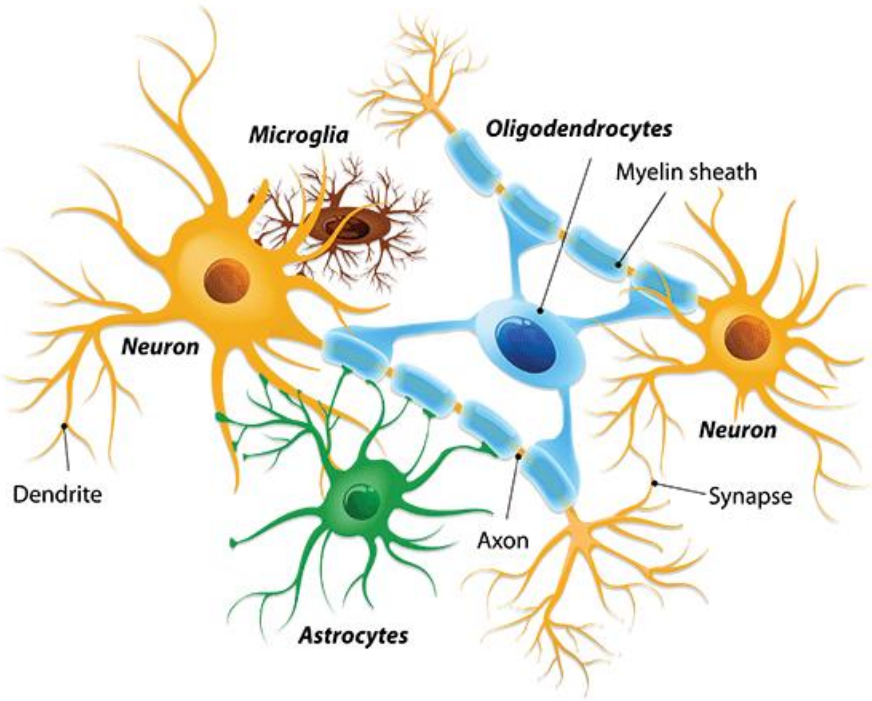


ROSENBLATT
died in a boat accident in 1971

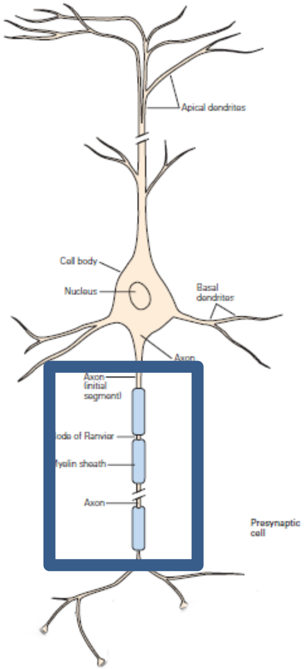


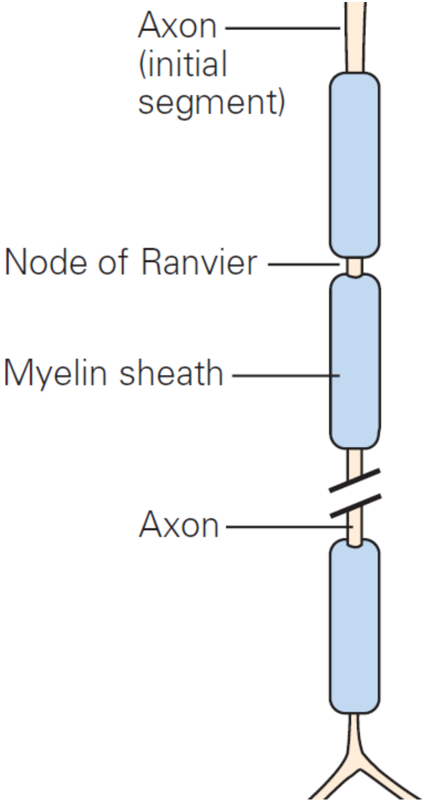


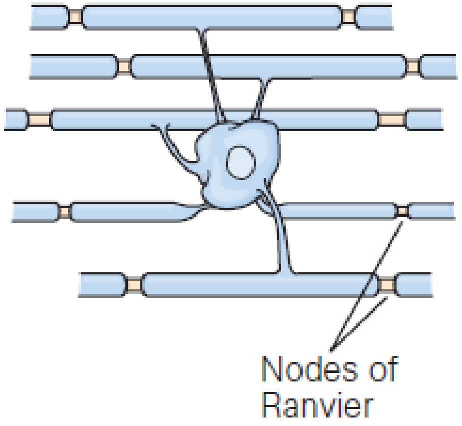
https://medium.com/@vnohitha13/neural-networks-52bec25688eb

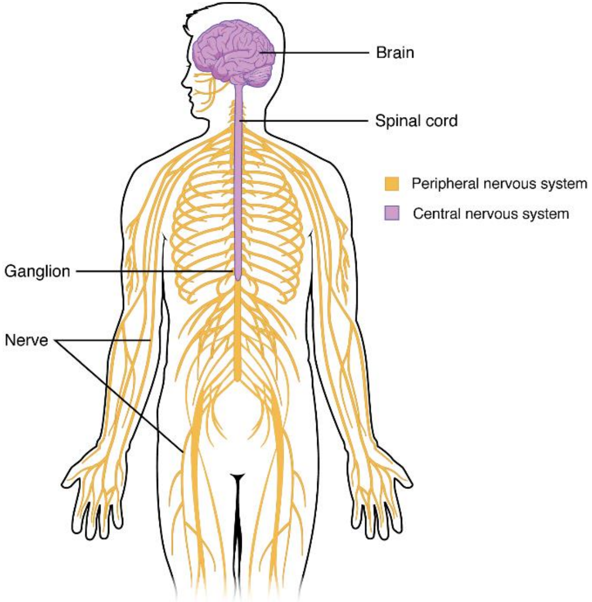


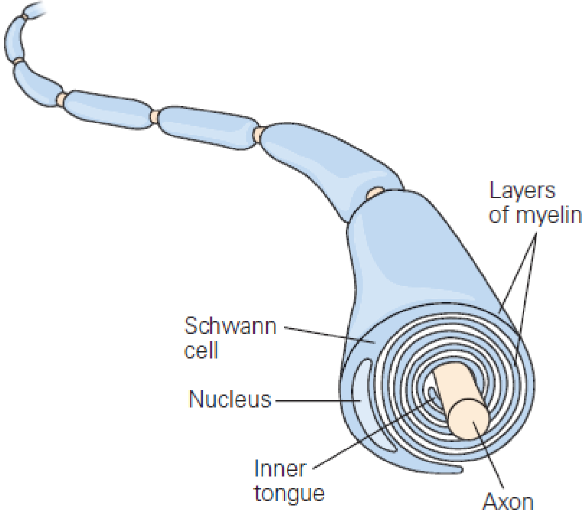
https://www.sciencenewsforstudents.org/article/scientists-say-glia


Apical dendrites
Cell body
Nucleus
Basal dendrites
Axon
Axon (initial segment)
ode of Ranvier
yelin sheath
Axon
Presynaptic cell


Axon (initial segment)
Node of Ranvier
Myelin sheath
Axon





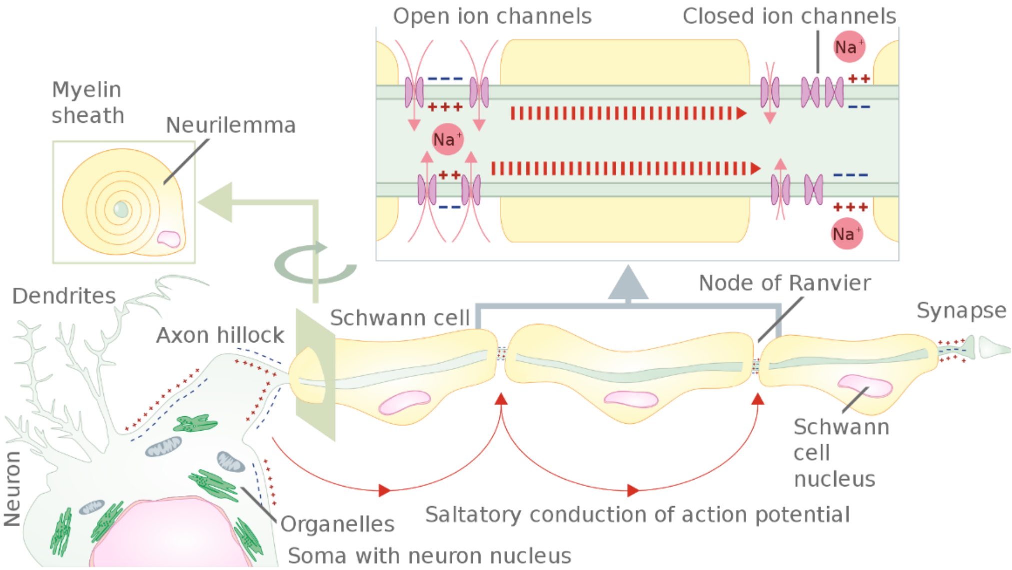


Oligodendrocytes
(Central nervous system)

Schwann cells
(Peripheral nervous system)

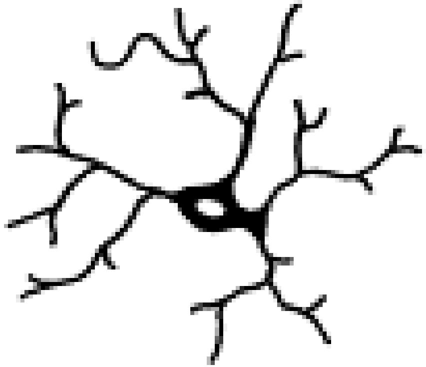
https://open.oregonstate.education/aandp/chapter/12-1-structure-and-function-of-the-nervous-system/

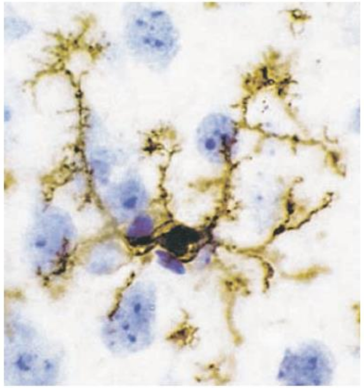
Saltatory conduction

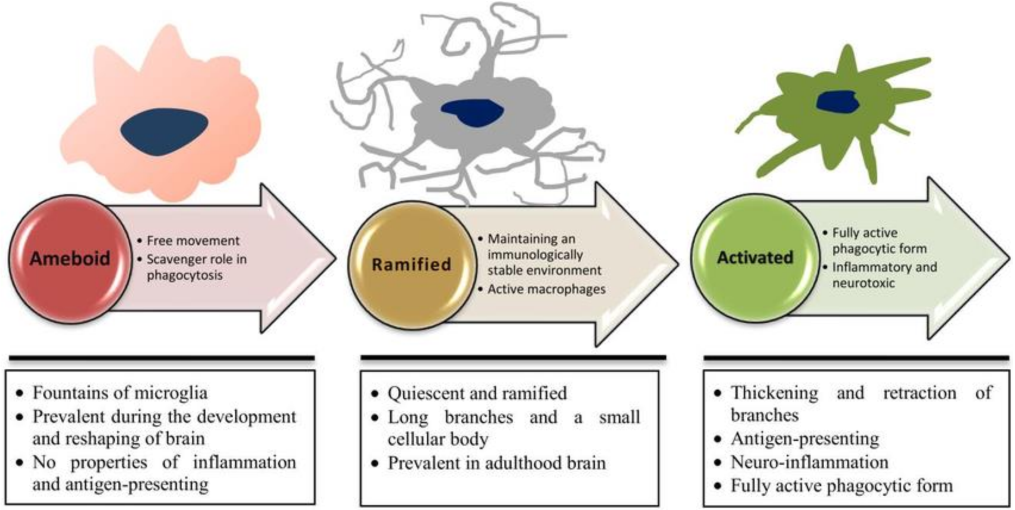


Microglia

- 10-15% of cells in the brain
- Resident macrophage cells:
  first and main active immune defense in CNS
- Scavenger for plaques and debris
- 3 forms: Ameboid, Ramified/Quiescent, Active



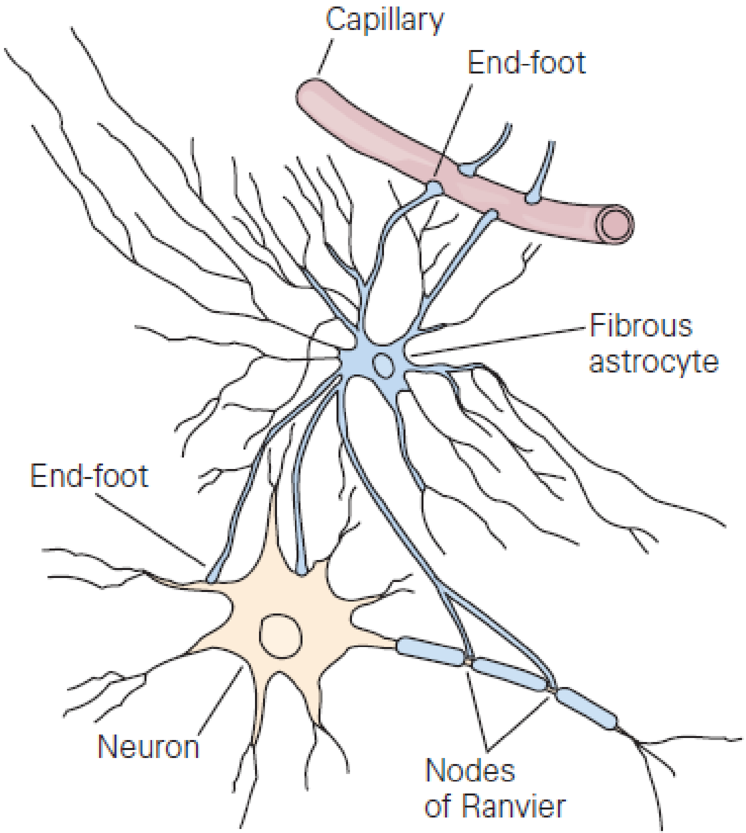




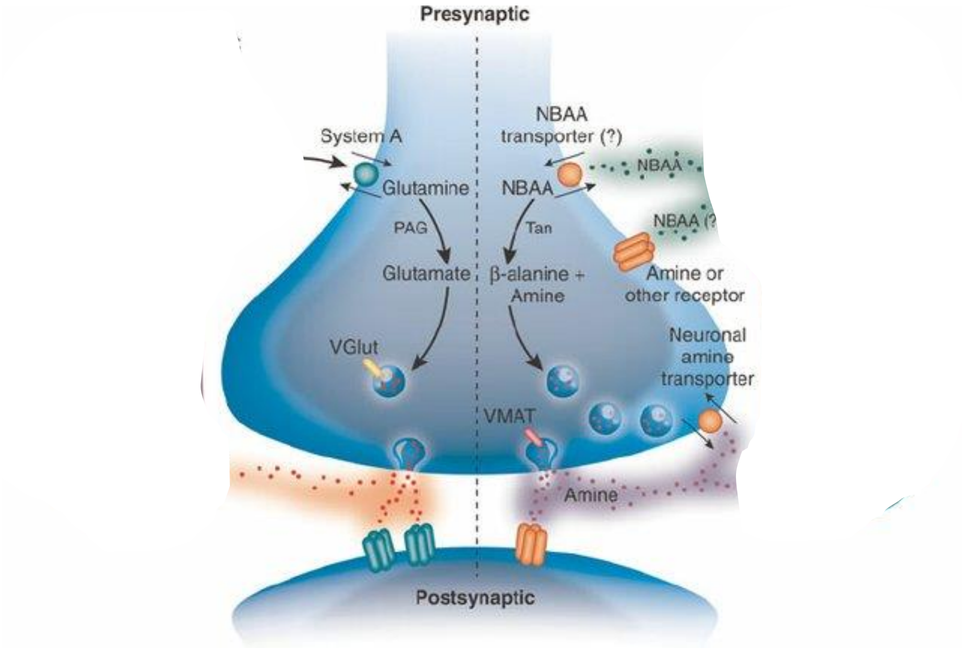
Zhiyou Cai et al., IJN 2013

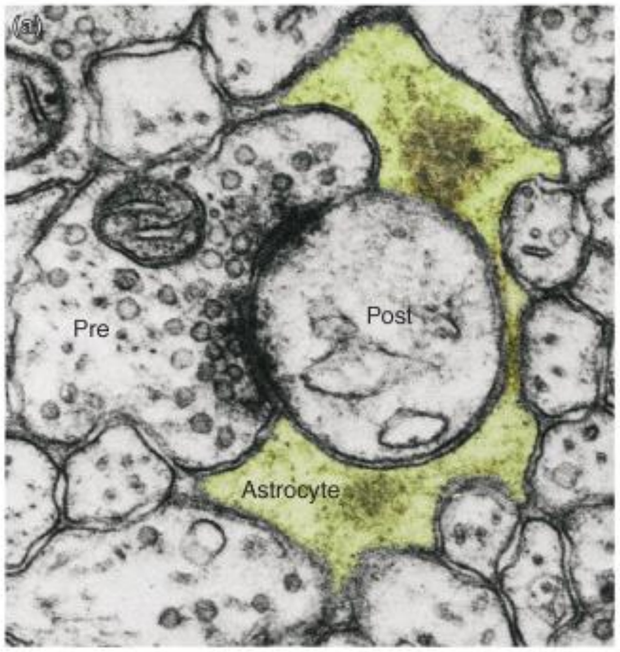
Cuadros and Navascués, Progress in Neurobiology 1998

Astrocytes


Capillary
End-foot
Fibrous astrocyte
End-foot
Neuron
Nodes of Ranvier

The tripartite synapse





Jackson, F. R. & Haydon, P. G. Glial cell regulation of neurotransmission and behavior in Drosophila. Neuron Glia Biol. 4, 11-17

Halassa et al., Trends in Molecular Medicine 2007

Building ANNs w/ glial support

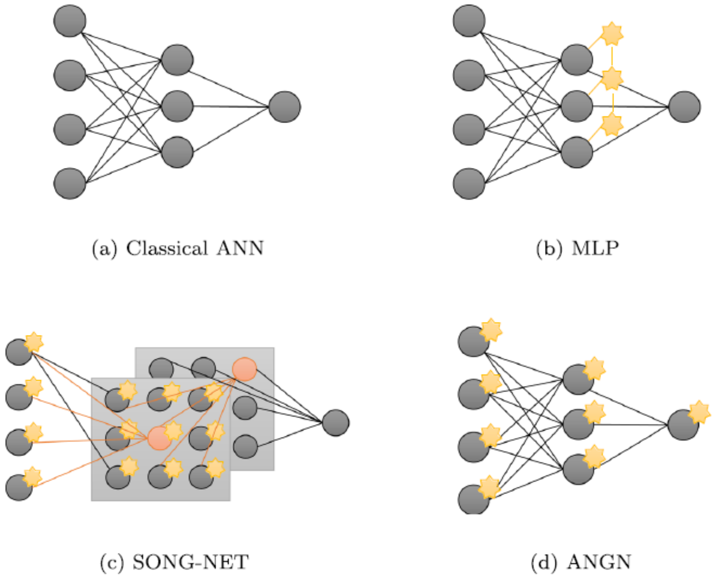


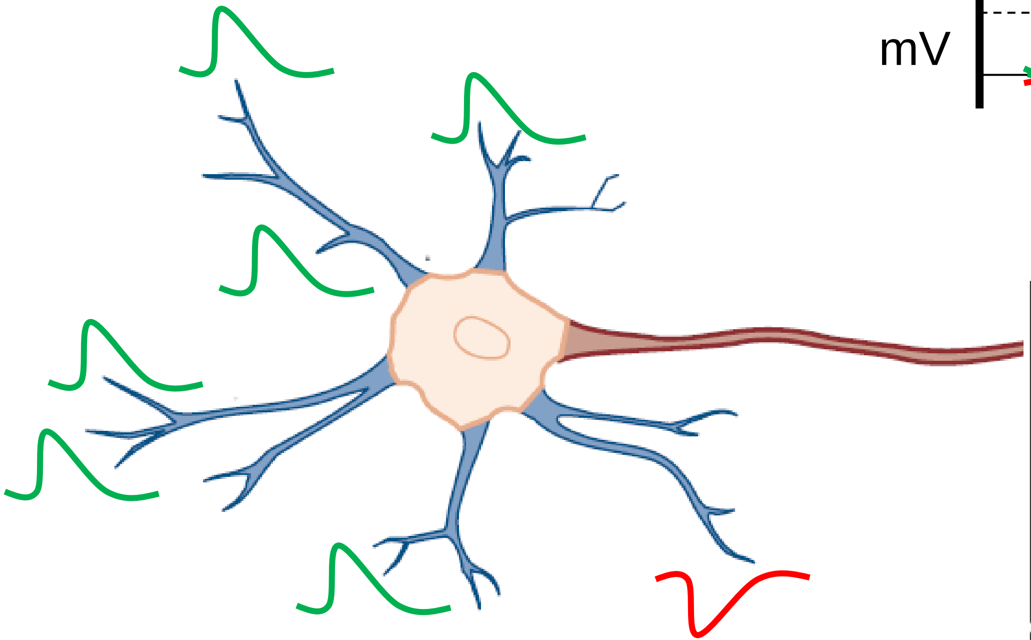
Fig. 2 Subdivision carried out on the basis of the papers selected according to the method in which the studies included the artificial astrocyte: **a** Classical feed-forward ANN, **b** MLPs with integrated astrocytes, **c** Multilayer Perceptron and Self-Organizing Maps or SONG-NETs and **d** Artificial Neuro-Glial Networks

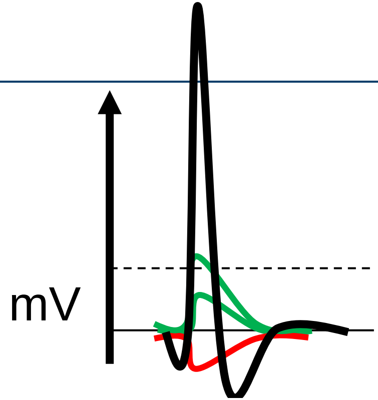
Takeaways:

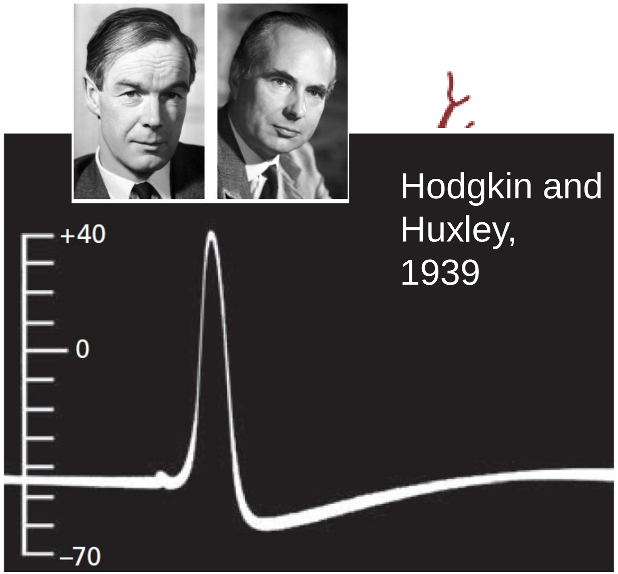
- Sometimes harder to train (no Gradient Descent possible, parameters using Genetic Algorithms)
- Inclusion of astrocytes avoided local minima and greater complexity (Ikuta et al., 2010/2011/2012, Gergel and Farkas 2018)
- Robustness to noise
- Depending on implementation: Faster to execute

Mesejo et al., Int J Neural Syst 2015

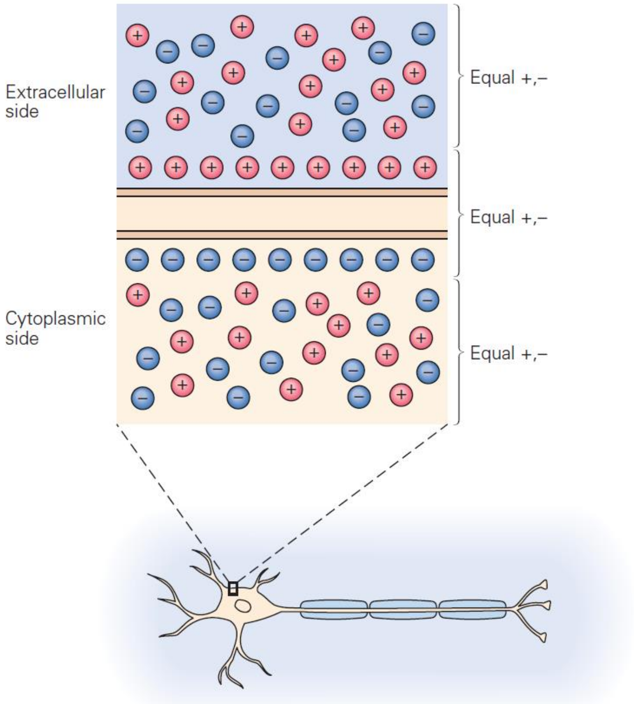
Neural computation



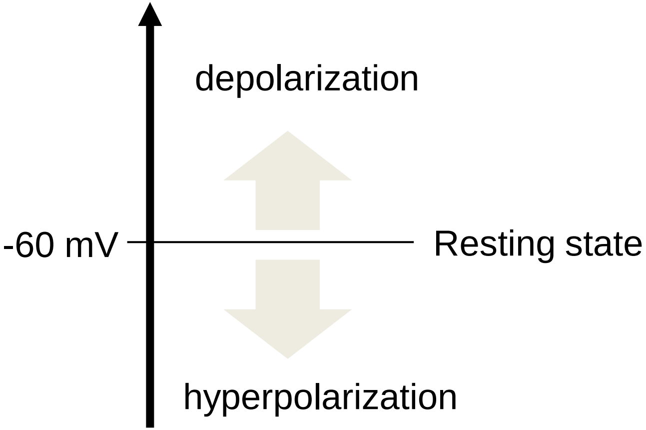


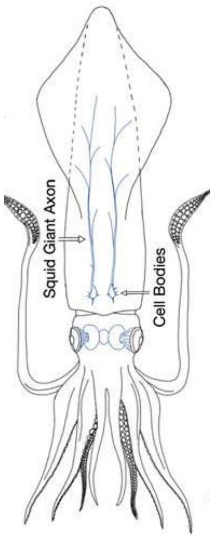


Ions and ion channels



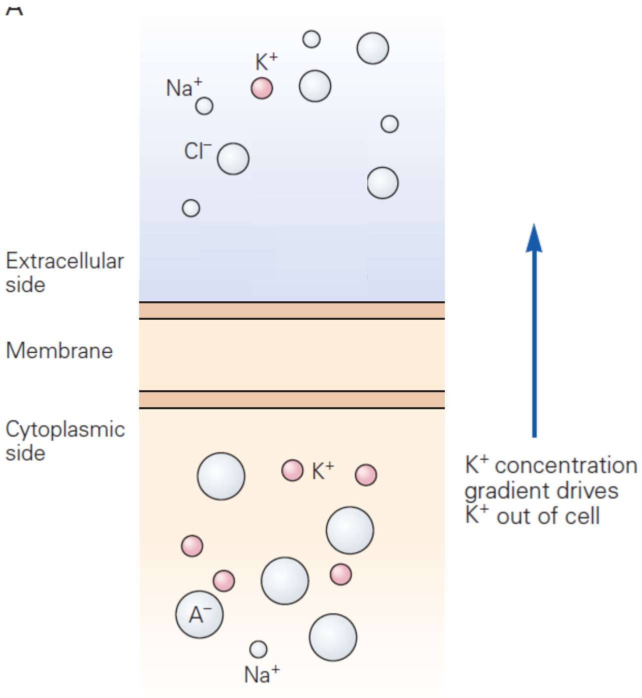
$$V_m = V_{in} - V_{out}$$

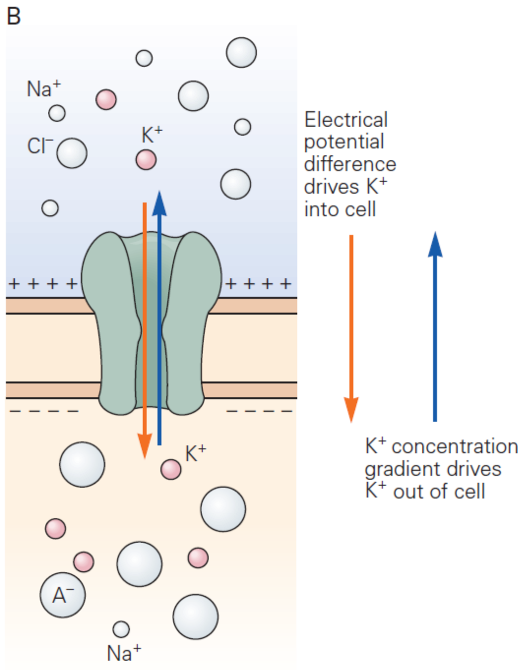




Species of ion	Concentration in cytoplasm (mM)	Concentration in extracellular fluid (mM)
K^+	400	20
Na^+	50	440
Cl^-	52	560
A^- (organic anions)	385	none

[1]The membrane potential at which there is no net flux of the ion species across the cell membrane.





Reaching equilibrium: The Nernst Equation

$$E_x = \frac{RT}{zF} \ln \frac{[X]_o}{[X]_i},$$ **Nernst Equation**

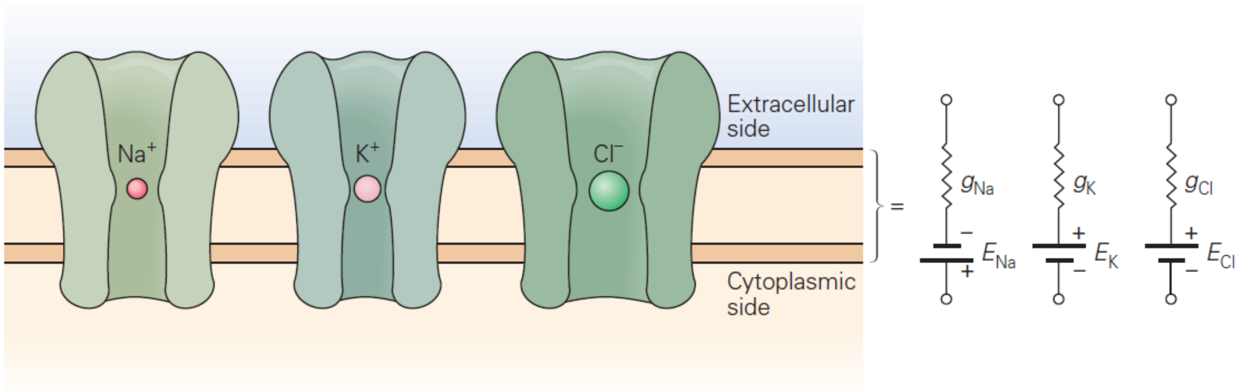
$$E_x = \frac{58\,\mathrm{mV}}{z} \log \frac{[X]_o}{[X]_i}.$$

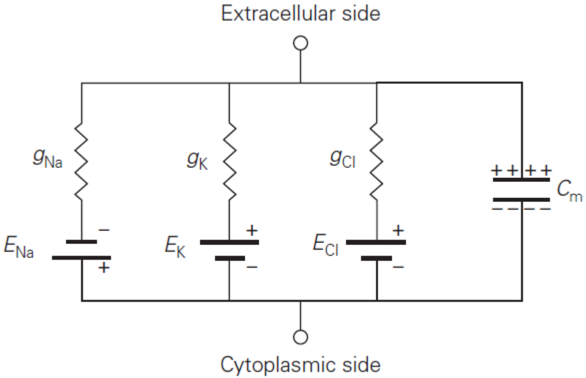
$$E_k = \frac{58\,\mathrm{mV}}{1} \log \frac{[20]}{[400]} = -75\,\mathrm{mV}.$$

Species of ion	Concentration in cytoplasm (mM)	Concentration in extracellular fluid (mM)	Equilibrium potential[1] (mV)
K^+	400	20	–75
Na^+	50	440	+55
Cl^-	52	560	–60
A^- (organic anions)	385	none	none

[1]The membrane potential at which there is no net flux of the ion species across the cell membrane.

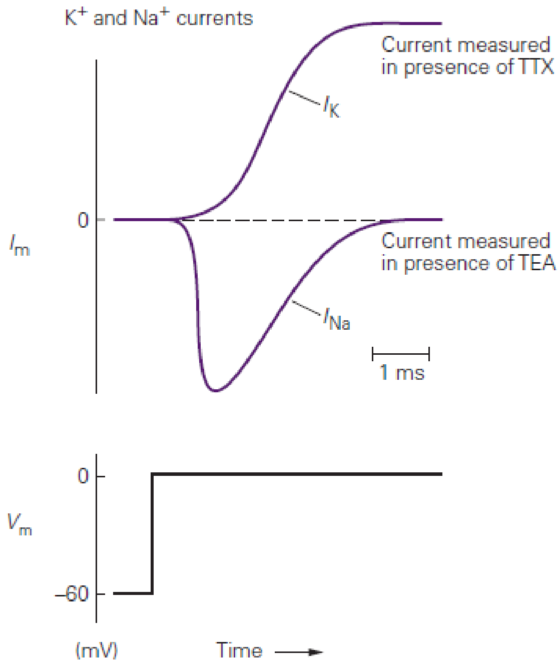
Biology explained as electrical circuit

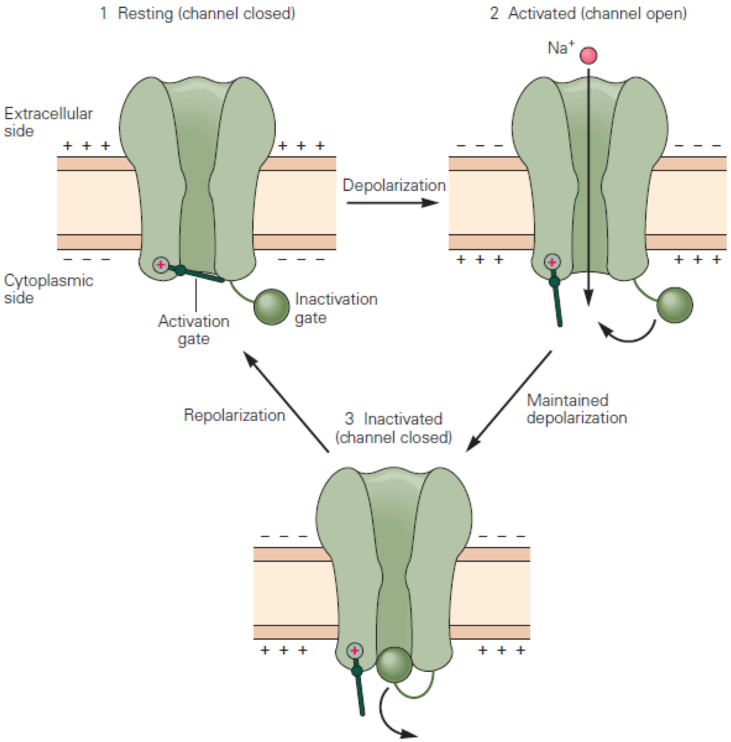


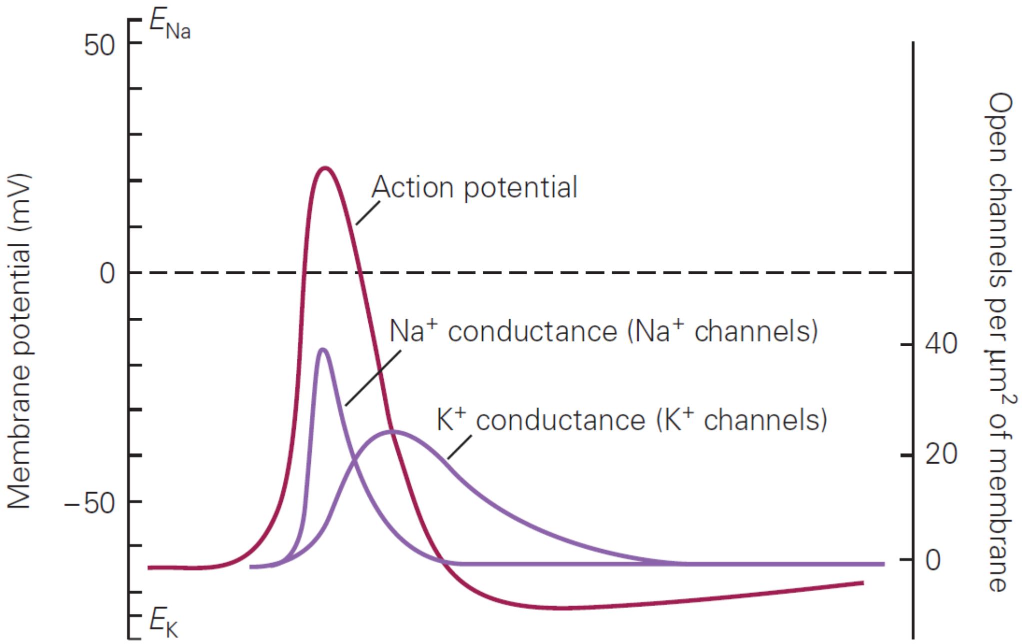


$$I_{Na} + I_K = 0.$$

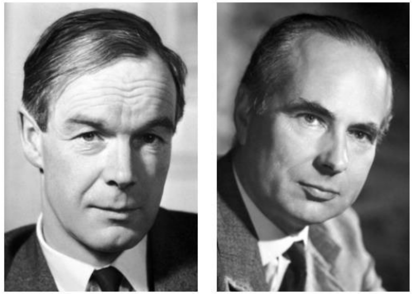
$$V_m = \frac{(E_{Na} \times g_{Na}) + (E_K \times g_K)}{g_{Na} + g_K}$$

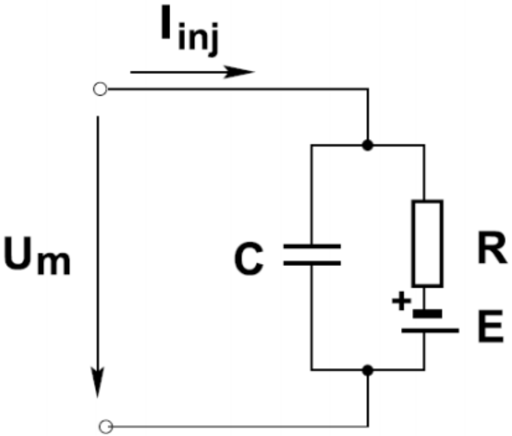

K+ and Na+ currents
Current measured in presence of TTX
I_K
0
I_m
Current measured in presence of TEA
I_{Na}
1 ms
0
V_m
–60
(mV)
Time


1 Resting (channel closed)
2 Activated (channel open)
Na+
Extracellular side
Cytoplasmic side
Activation gate
Inactivation gate
Depolarization
Repolarization
Maintained depolarization
3 Inactivated (channel closed)


E_{Na}
50
0
−50
E_K
Membrane potential (mV)
Action potential
Na⁺ conductance (Na⁺ channels)
K⁺ conductance (K⁺ channels)
40
20
0
Open channels per μm² of membrane

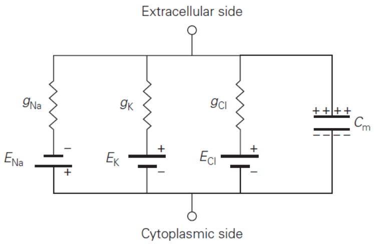
The Hodgkin-Huxley model





$$I_K = g_K(V_m - E_K)$$

Kirchhoff‘s law: conservation of electrical charge



$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -\sum_{\mathrm{ion}} I_{\mathrm{ion}} + I(t).$$

Hodgkin and Huxley found three dynamic variables: **n, m** and **h**

$$I = C_m \frac{\mathrm{d}V_m}{\mathrm{d}t} + \bar{g}_{\mathrm{K}} n^4 (V_m - V_K) + \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{Na}) + \bar{g}_l (V_m - V_l),$$

Solving the unknown

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

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

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

$$\alpha_n(V_m) = \frac{0.01(10+V_m)}{\exp\left(\frac{10+V_m}{10}\right)-1} \qquad \alpha_m(V_m) = \frac{0.1(25+V_m)}{\exp\left(\frac{25+V_m}{10}\right)-1} \qquad \alpha_h(V_m) = 0.07\exp\left(\frac{V_m}{20}\right)$$

$$\beta_n(V_m) = 0.125\exp\left(\frac{V_m}{80}\right) \qquad \beta_m(V_m) = 4\exp\left(\frac{V_m}{18}\right) \qquad \beta_h(V_m) = \frac{1}{\exp\left(\frac{30+V_m}{10}\right)+1}$$

Plugging it in Python

Steps:

1) Pre-define variables
2) Iterate over time
    1) Solve alpha, beta
    2) Solve partial differential equation using Euler‘s method
    3) Calculate conductances
    4) Calculate current
    5) Calculate membrane voltage
3) Plot result

$$\alpha_n(V_m) = \frac{0.01(10+V_m)}{\exp\left(\frac{10+V_m}{10}\right)-1} \qquad \alpha_m(V_m) = \frac{0.1(25+V_m)}{\exp\left(\frac{25+V_m}{10}\right)-1} \qquad \alpha_h(V_m) = 0.07\exp\left(\frac{V_m}{20}\right)$$

$$\beta_n(V_m) = 0.125\exp\left(\frac{V_m}{80}\right) \qquad \beta_m(V_m) = 4\exp\left(\frac{V_m}{18}\right) \qquad \beta_h(V_m) = \frac{1}{\exp\left(\frac{30+V_m}{10}\right)+1}$$

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

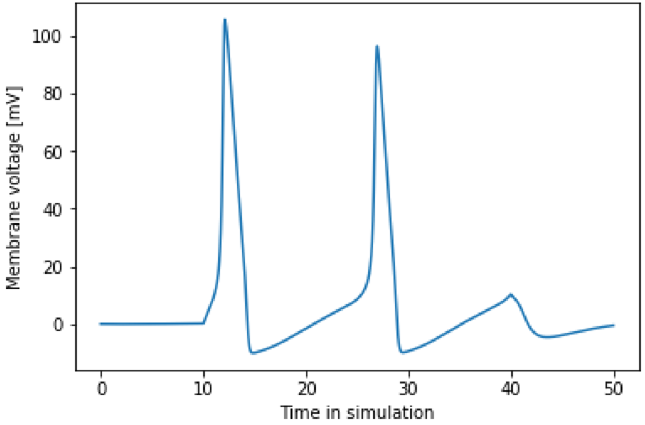
$$\frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m$$

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

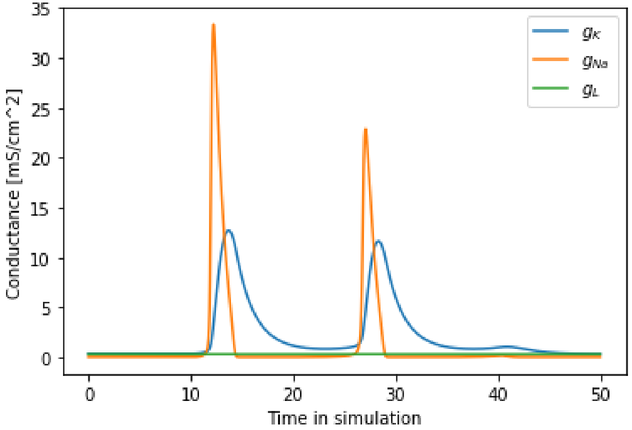
$$\bar{g}_{\mathrm{K}} n^4 (V_m - V_K) + \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{Na}) + \bar{g}_l (V_m - V_l),$$

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -\sum_{\mathrm{ion}} I_{\mathrm{ion}} + I(t).$$

[redacted] Applied external current



[redacted]

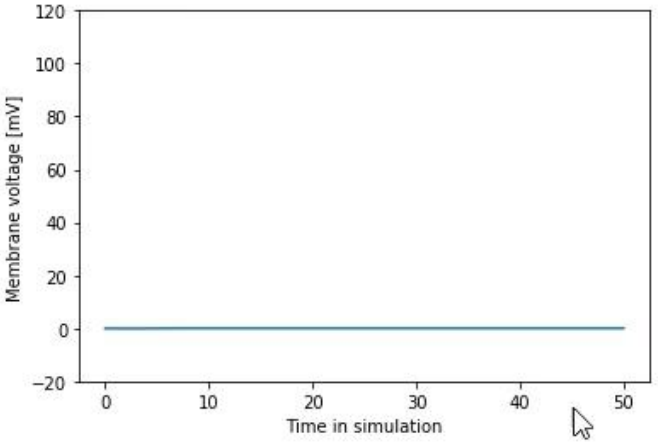


(code will be on StudOn that you can play with)

Changing the injected current

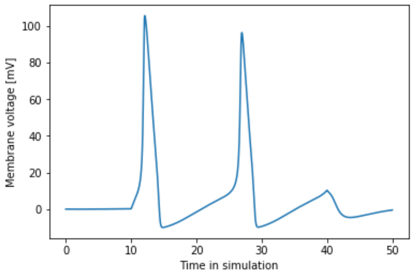
```python
@interact
def applyCurrent(I_ext_applied:(0,100,3)=0):
    plt.figure()
    plt.plot(ts, HHs[I_ext_applied])
    plt.xlabel("Time in simulation")
    plt.ylabel("Membrane voltage [mV]")
    plt.ylim([-20,120])
```

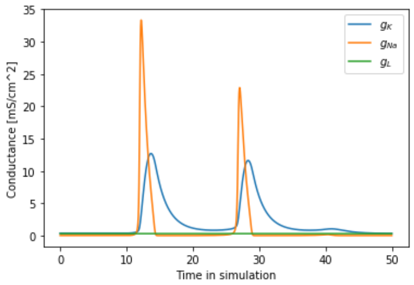
I_ext_applied 0



Hodgkin-Huxley „drawbacks“

Already for this easy task we need to estimate **20 parameters**





ARE WE ABLE TO SIMPLIFY THIS?

Integrate-and-Fire-Models

$$C_m \frac{dV}{dt} = \frac{V_{rest} - V}{R_m} + I$$

Simplifications:

1) Linear membrane current and membrane potential relationship
2) Fires action potentials through a threshold-crossing rule

Retains:

1) Membrane capacitance
2) Membrane resistance
3) Temporal dependence

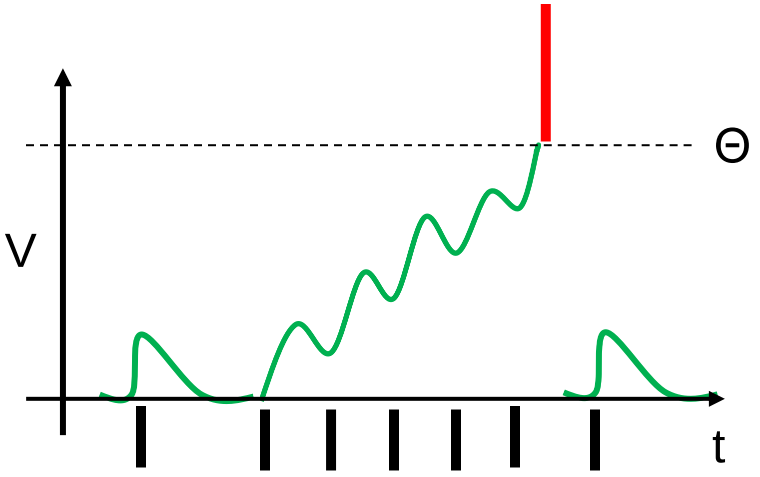
Integrate-and-Fire-Models

$$C_m \frac{dV}{dt} = \frac{V_{rest} - V}{R_m} + I \quad \Big| \cdot R$$

$$\tau_m = RC$$

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = (V_{rest} - V) + \boxed{R_m I(t)}$$

External input,
e.g. via synapse



Simulation

Steps:

1) Define constants
2) Define input signals
3) Perform simulation
    1) Reset neuron to Vm if AP was elicited
    2) Integrate signal
4) Plot results

```python
# Constants
tau = 10 # ms
Vm = -65 # mV
theta = -55 # mV
V = -65 # mV

# Input strength
RI = 100 # mV

# Simulate for 40 s, time step 0.1
dt = 0.1
```

```python
# Set input at given ts
inputs = [10, 20, 22, 24, 26, 26.5, 28, 29, 29.4]
durations = [.2, .2, .2, .2, .2, .2, .2, .2, .2]

# Construct signal
input_signal = np.zeros_like(ts)

for i,d in zip(inputs, durations):
    input_signal[(ts >= i) & (ts <= i+d)] = RI
```

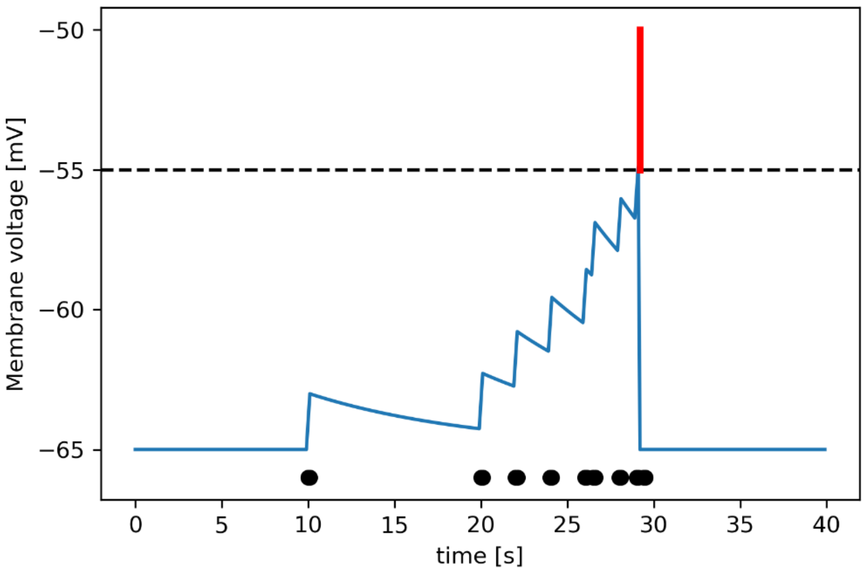
```python
# Check if action potential is elicited
s = V > theta

if s:
    V = Vm
```

```python
V = V-dt/tau * ((V - Vm) - RI_)
```

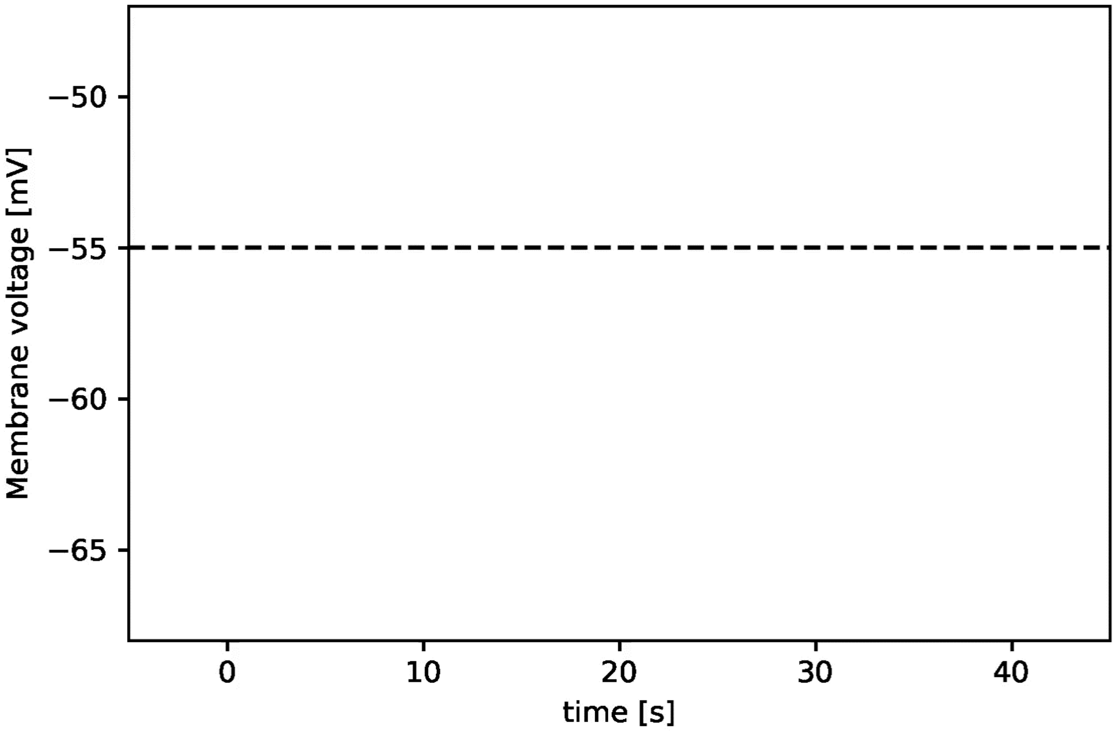
Simulation

Steps:

1) Define constants
2) Define input signals
3) Perform simulation
    1) Reset neuron to Vm if AP was elicited
    2) Integrate signal
4) Plot results

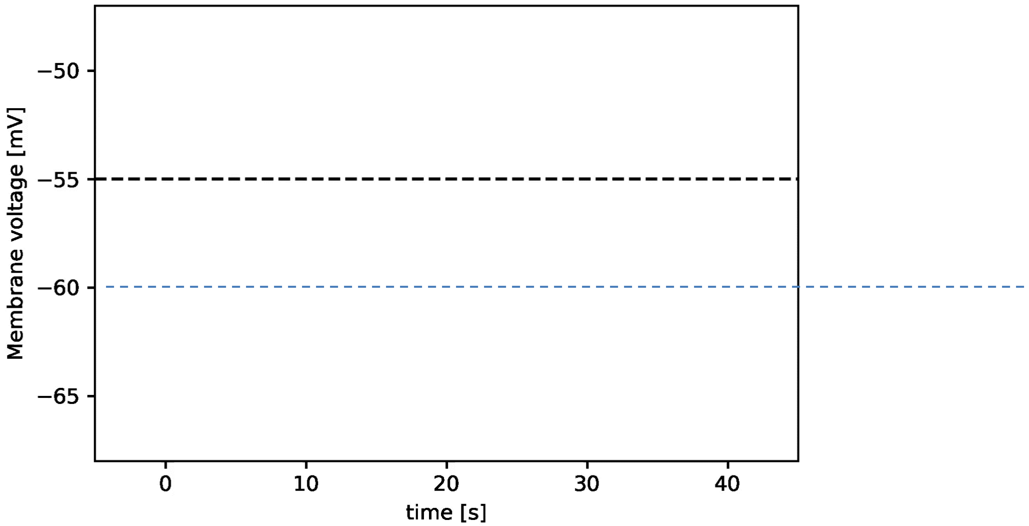


Temporal integration



Constant input current

5 mV

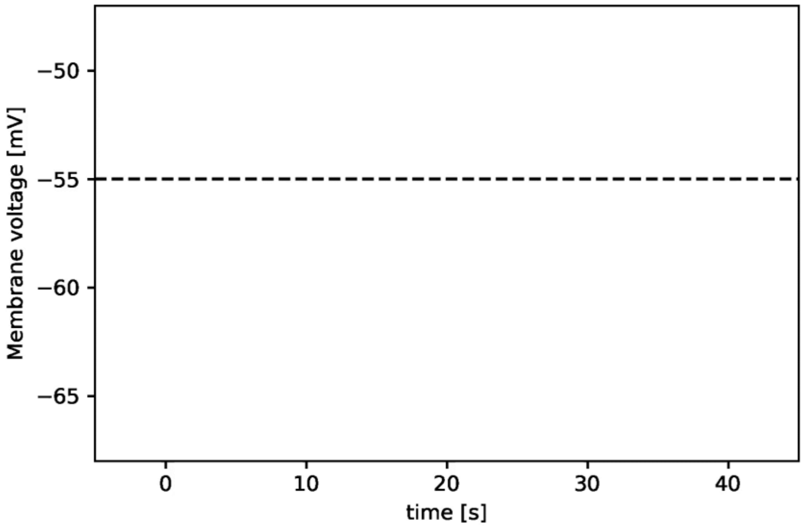


approaches
Vm+RI

with time
constant τ

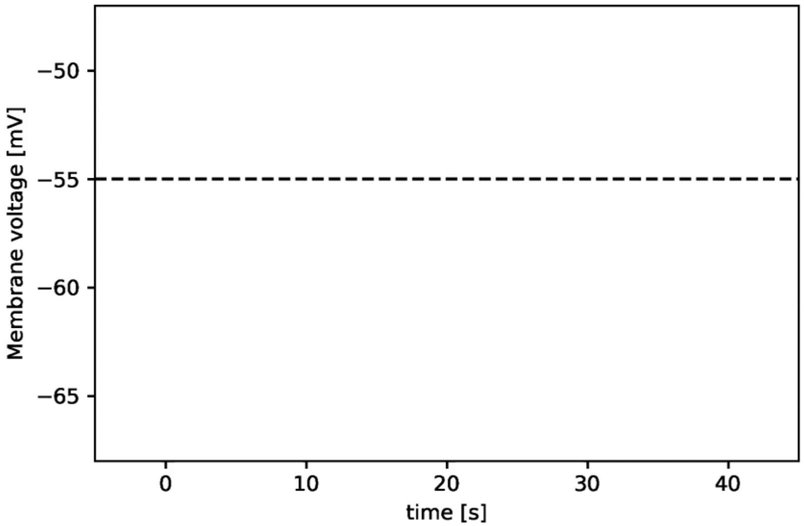
Constant input current

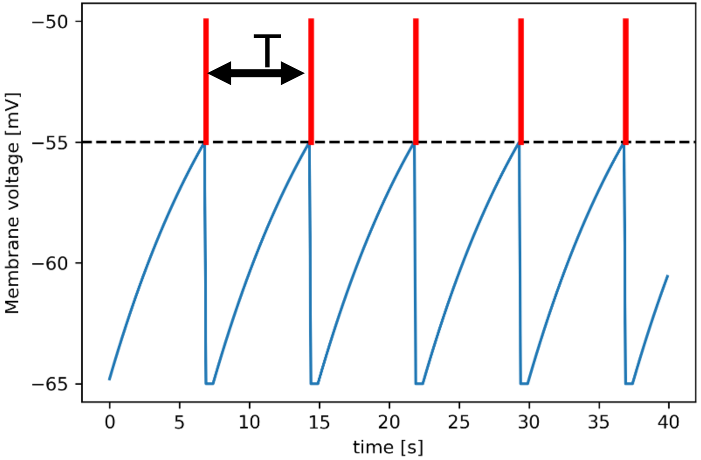
20 mV



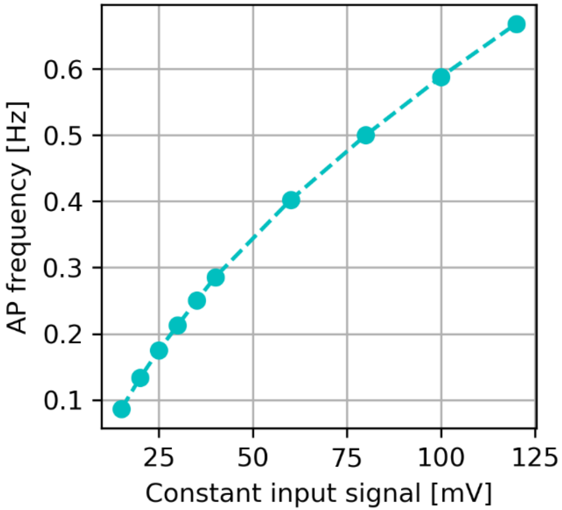
Constant input current

40 mV



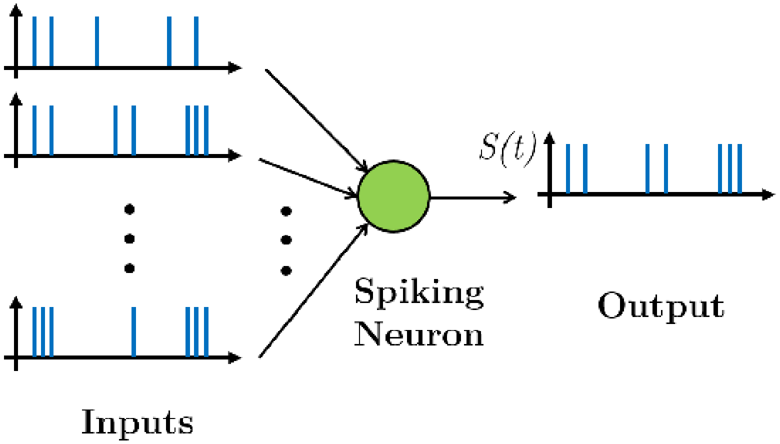

−50
−55
−60
−65
Membrane voltage [mV]
T
0
5
10
15
20
25
30
35
40
time [s]

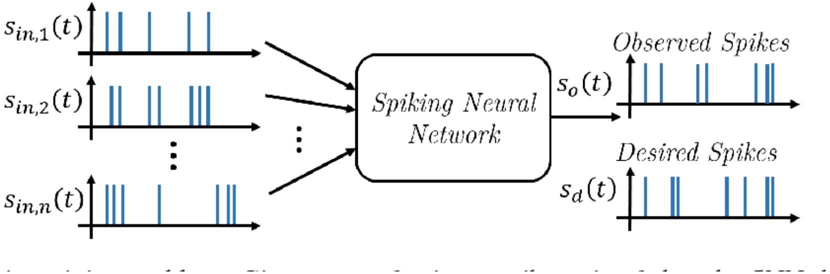
f = 1/T


0.6
0.5
0.4
0.3
0.2
0.1
AP frequency [Hz]
25
50
75
100
125
Constant input signal [mV]

Why is this important?

Spiking Neural Networks



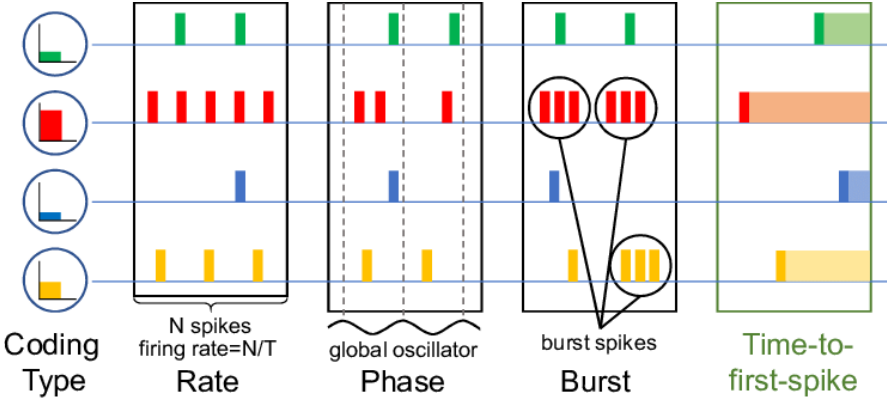


Ways of training SNNs

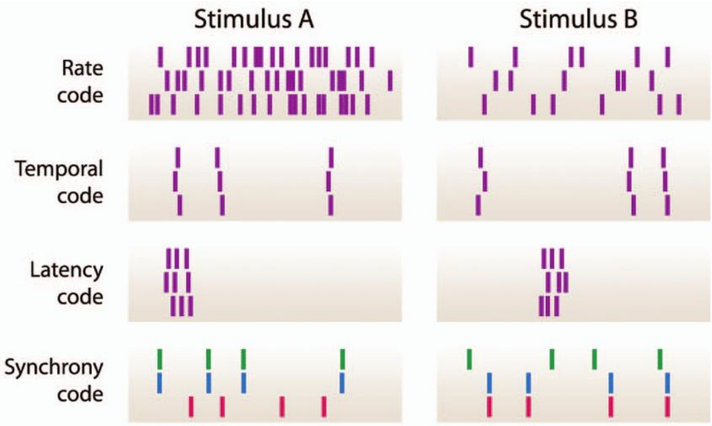
1. Binarization of ANNs: Conventional DNNs are trained with binary activations, but maintain their synchronous mode of information processing.
2. Conversion from ANNs: Conventional DNNs are trained with backpropagation, and then all analog neurons are converted into spiking ones.
3. Training of constrained networks: Before conversion, conventional DNN training methods are used together with constraints that model the properties of the spiking neuron models.
4. Supervised learning with spikes: Directly training SNNs using variations of error backpropagation.
5. Local learning rules at synapses, such as STDP (Bi and Poo, 1998; Song et al., 2000), are used for more biologically realistic training.

Pfeiffer and Pfeil, 2018

Neural coding



Park et al., 2020



Gollisch, 2008

Neuromorphic hardware

Table 1: Capacity of some recent neuromorphic systems [18].

	ODIN [19]	μBrain [20]	DYNAPs [21]	BrainScaleS [22]	SpiNNaker [23]	Neurogrid [24]	Loihi [25]	TrueNorth [26]
# Neurons/core	256	336	256	512	36K	65K	130K	1M
# Synapses/core	64K	38K	16K	128K	2.8M	8M	130M	256M
# Cores/chip	1	1	1	1	144	128	128	4096
# Chips/board	1	1	4	352	56	16	768	4096
# Neurons	256	336	1K	4M	2.5B	1M	100M	4B
# Synapses	256	336	65K	1B	200B	16B	100B	1T