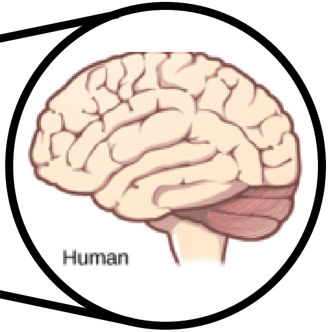
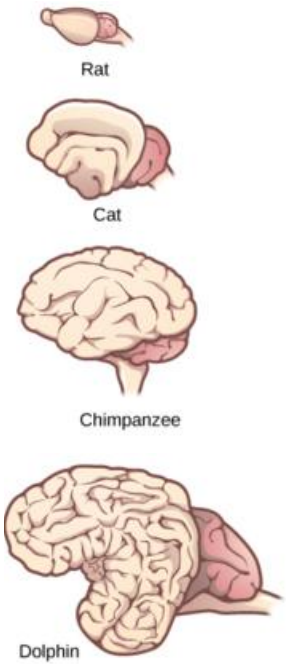
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Cognitive Neuroscience for AI Developers

Structure and Function of the Nervous System: Neurons and Glia

SENSORY → THE BRAIN → BEHAVIOR

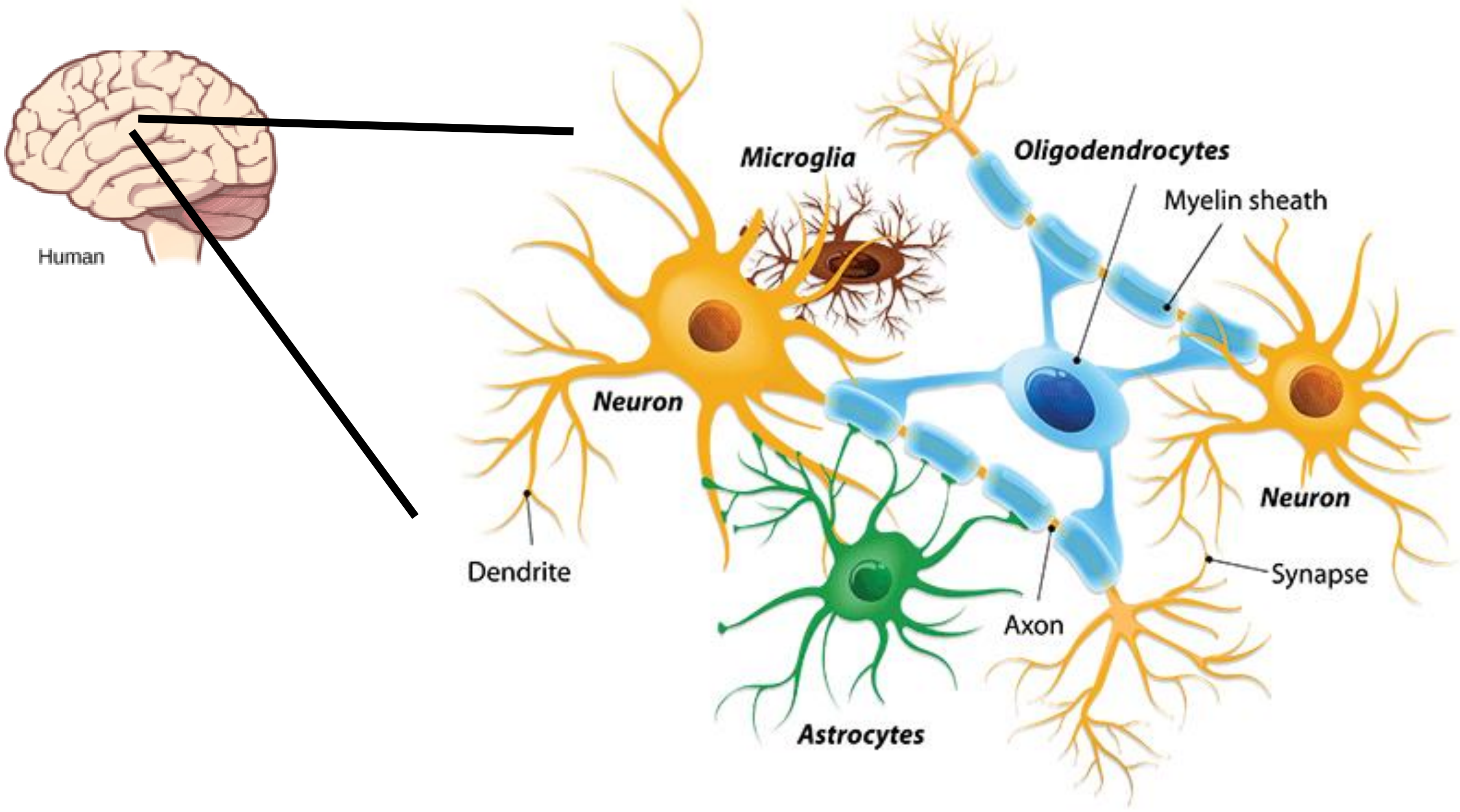




https://wiki2.org/en/Brain_size

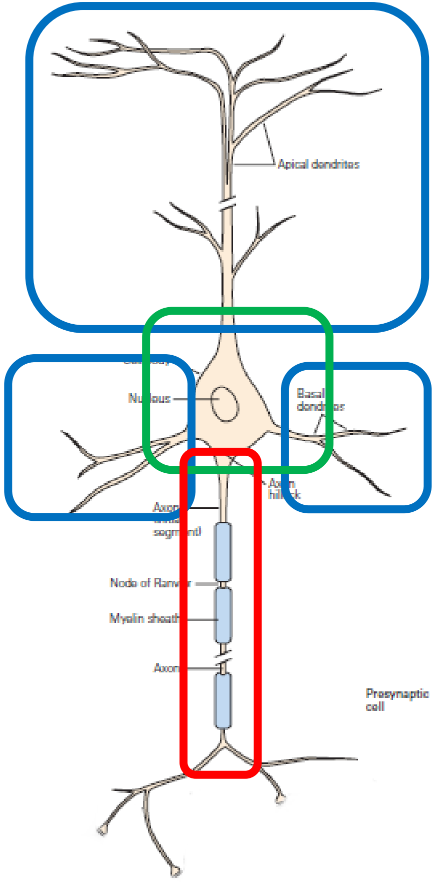
But what have all these brains in common?

CELLS.

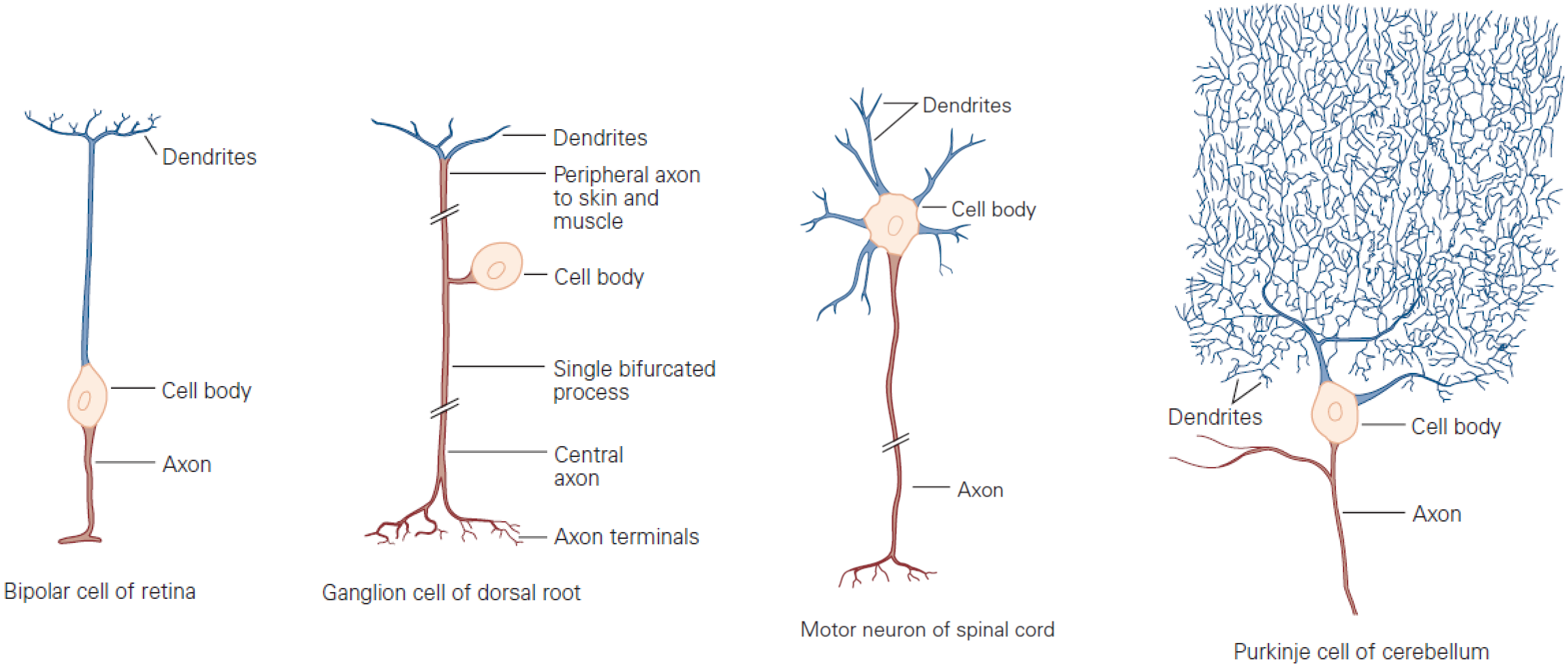


https://www.sciencenewsforstudents.org/article/scientists-say-glia

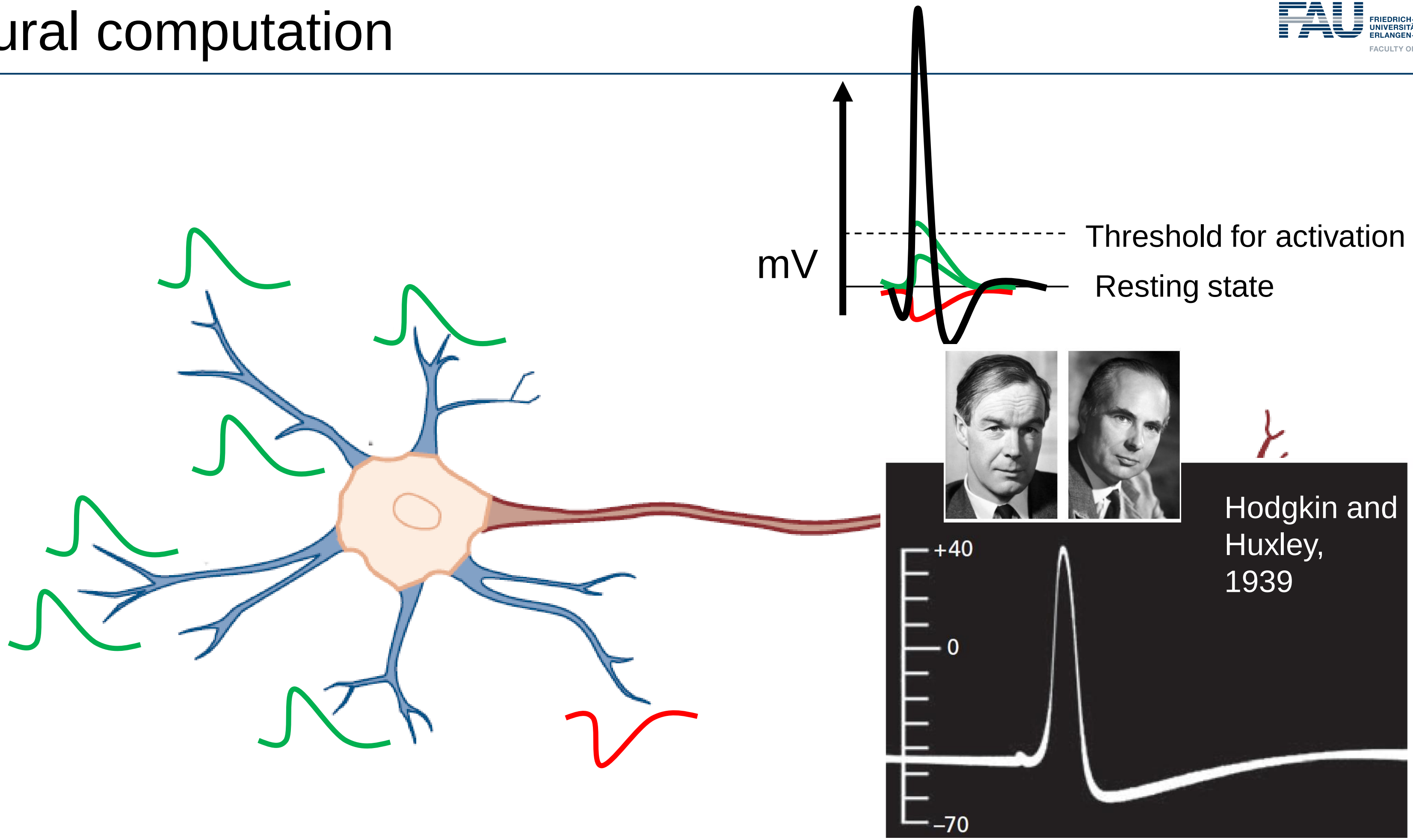
A neuron



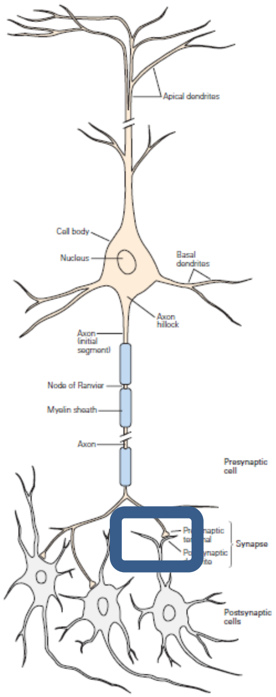
Neuron types

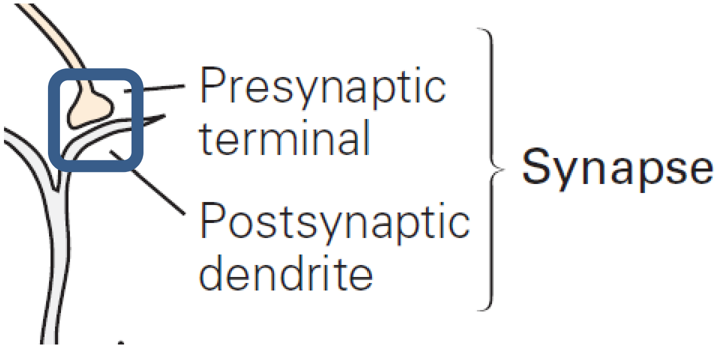


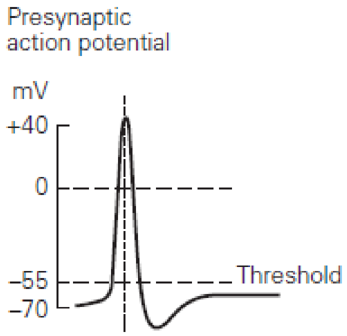
Neural computation

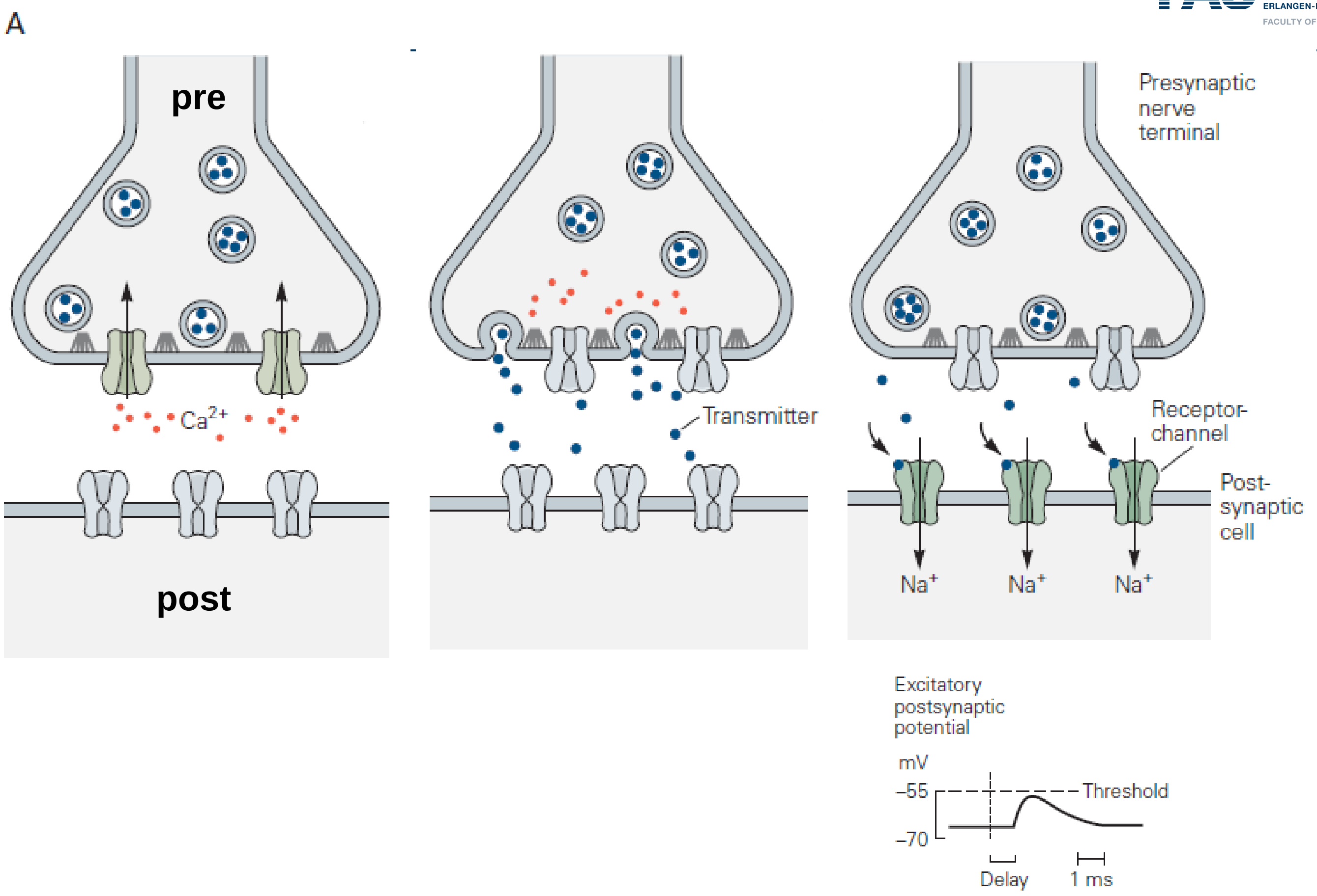


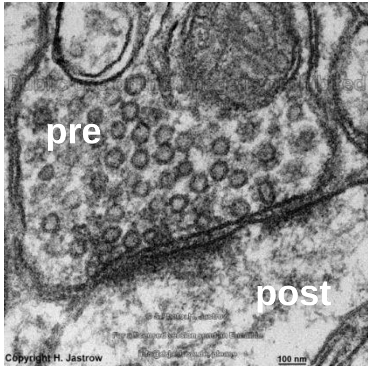
The connection between two neurons: the synapse





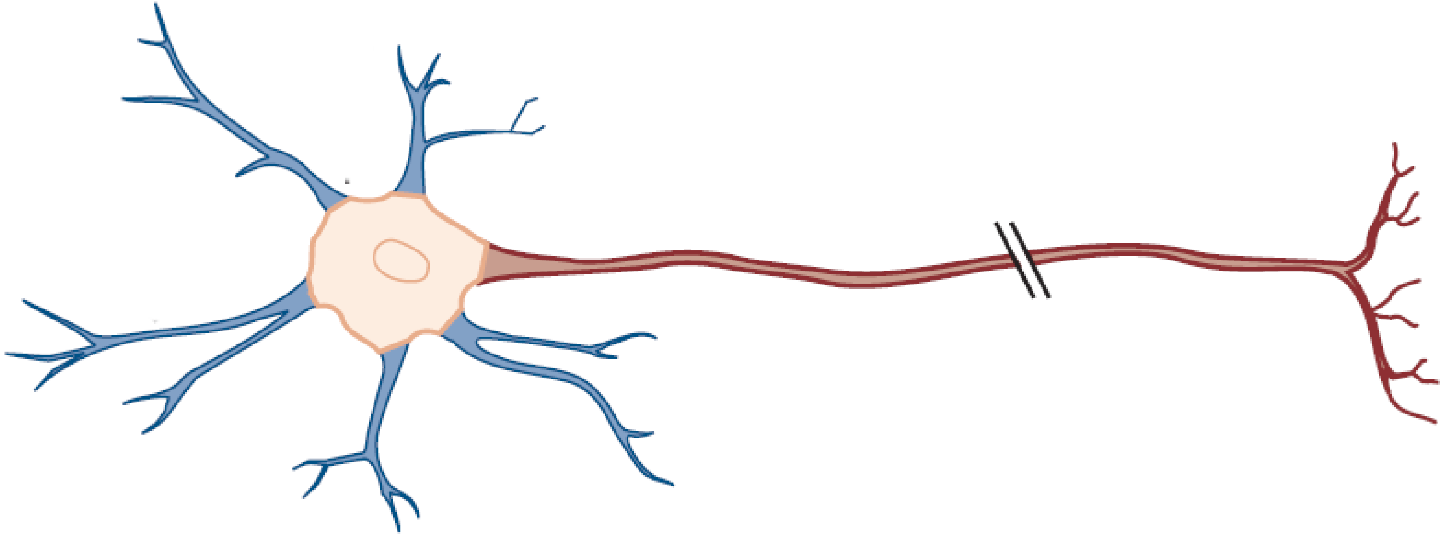




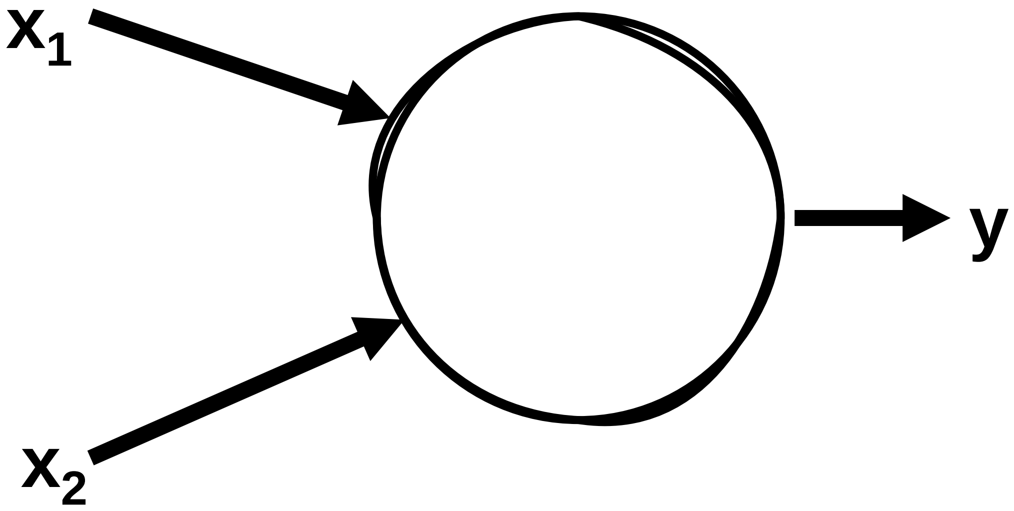


http://www.drjastrow.de/WAI/EM/EMSynapse.html

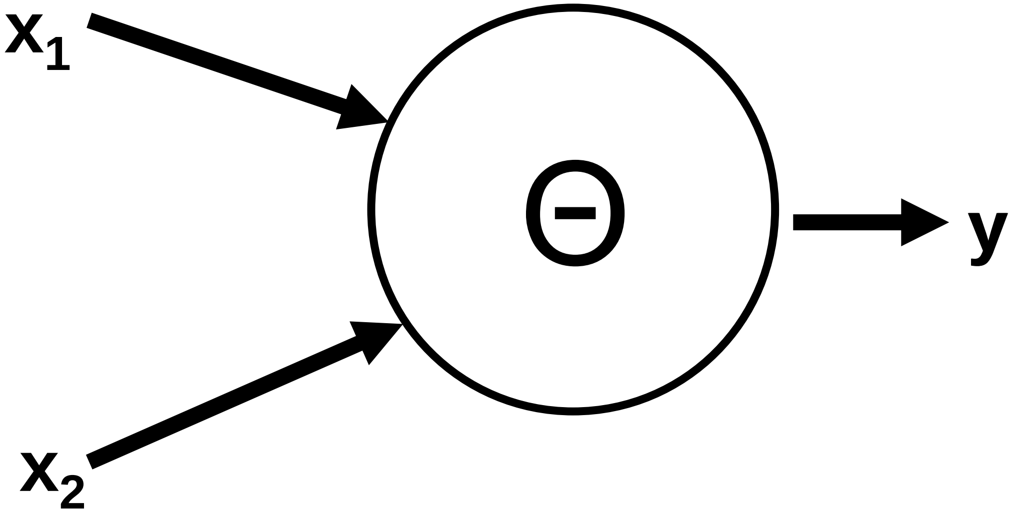
Mathematically simplifying a neuron

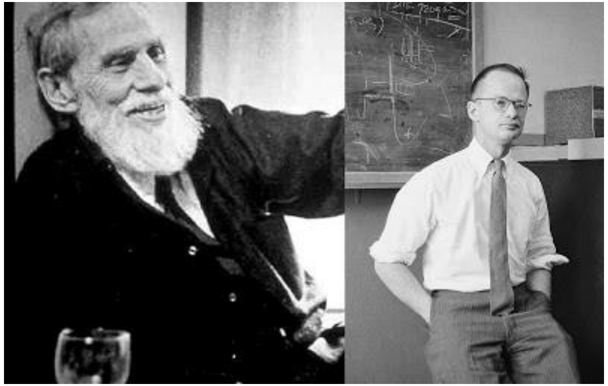


A theoretical neuron

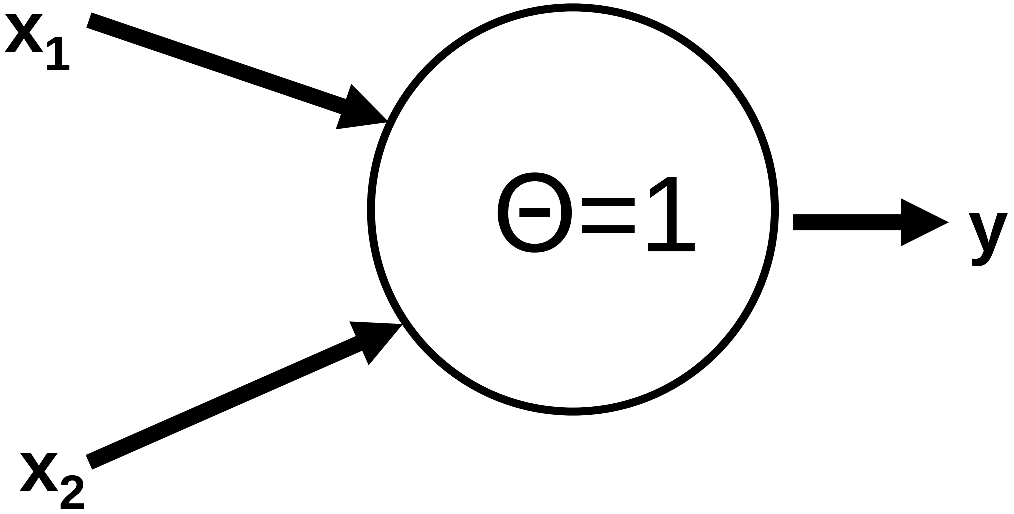


The McCulloch-Pitts neuron





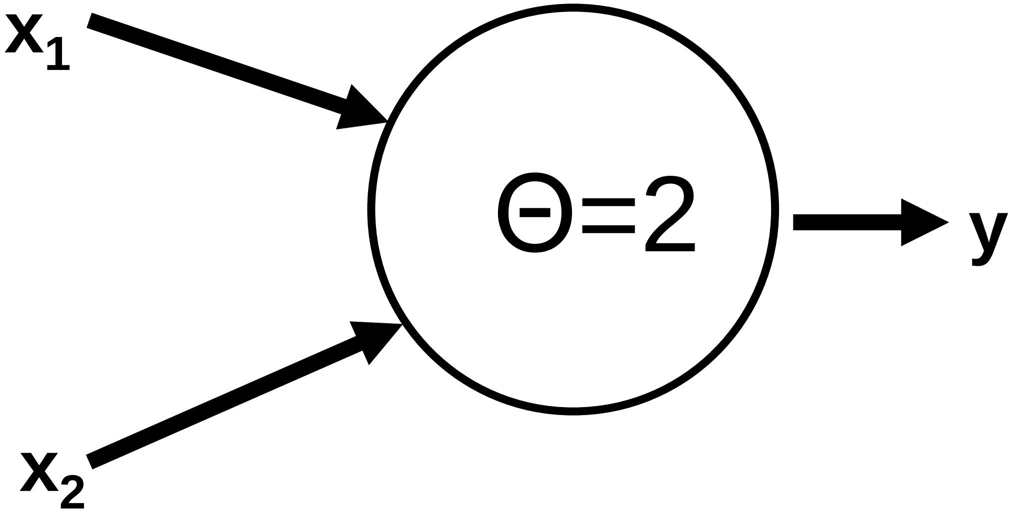
Logical operations with the McCulloch-Pitts neuron



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

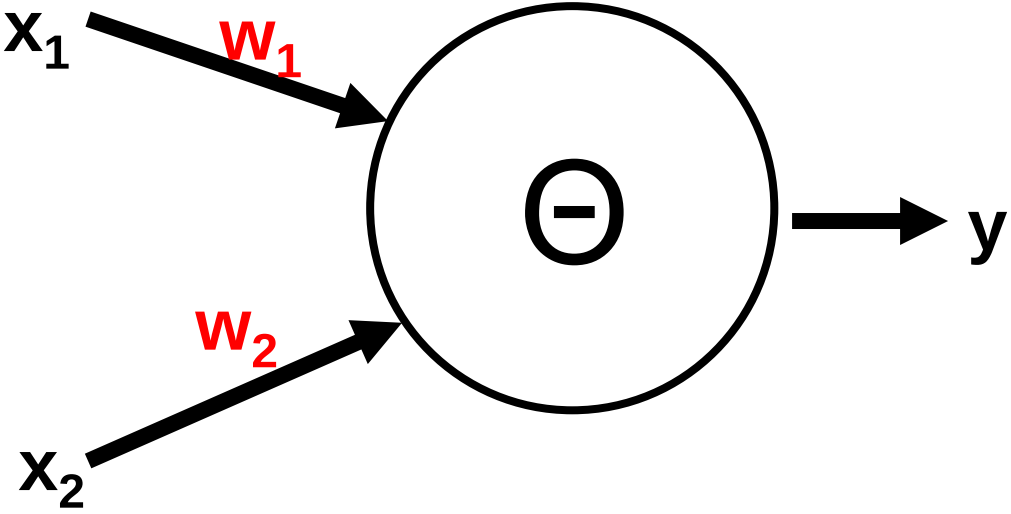
LOGICAL AND

McCulloch Pitts Neuron
(assuming no inhibitory inputs)

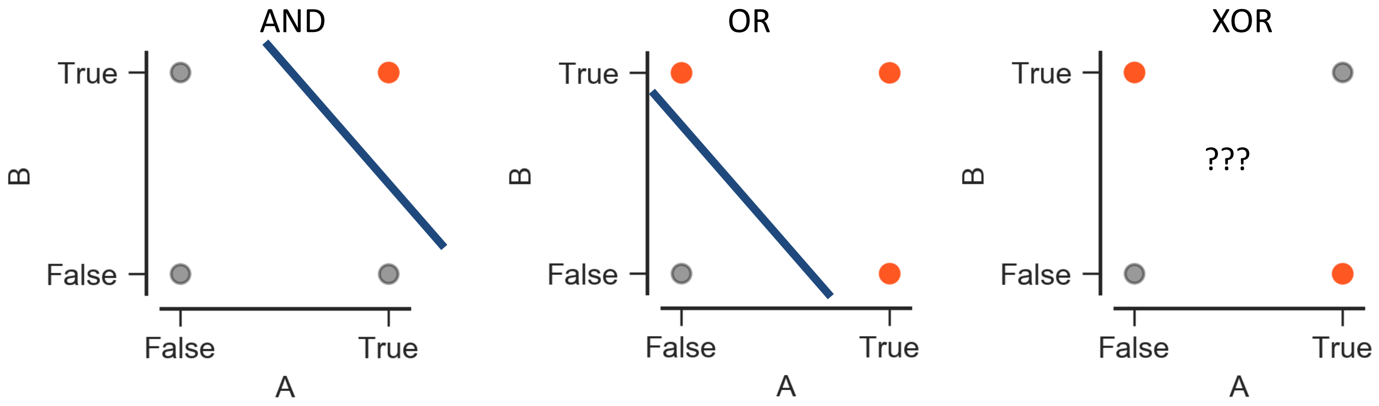
$$y = 1 \quad if \sum_{i=0}^{n} x_i \geq \Theta$$
$$= 0 \quad if \sum_{i=0}^{n} x_i < \Theta$$

Perceptron

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



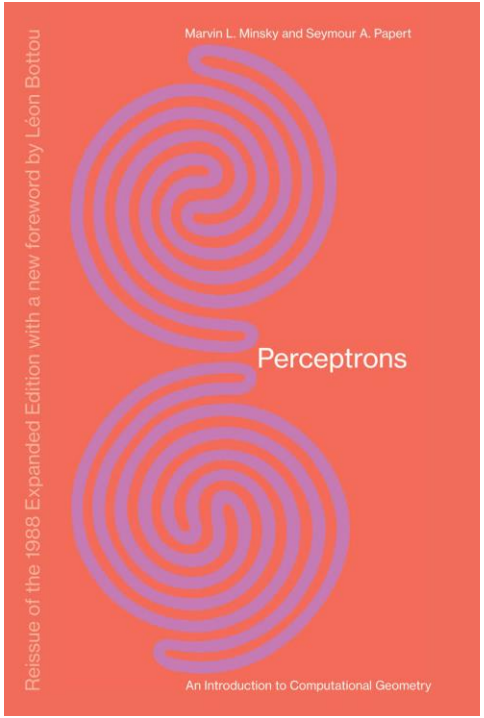
The XOR affair

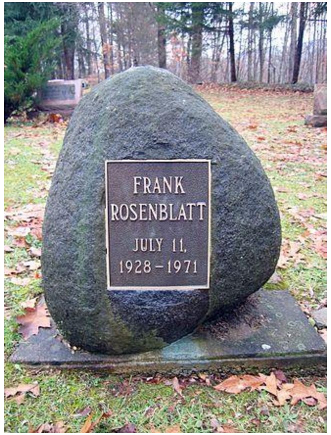


$\Theta=2$ $\Theta=1$ $\Theta=?$

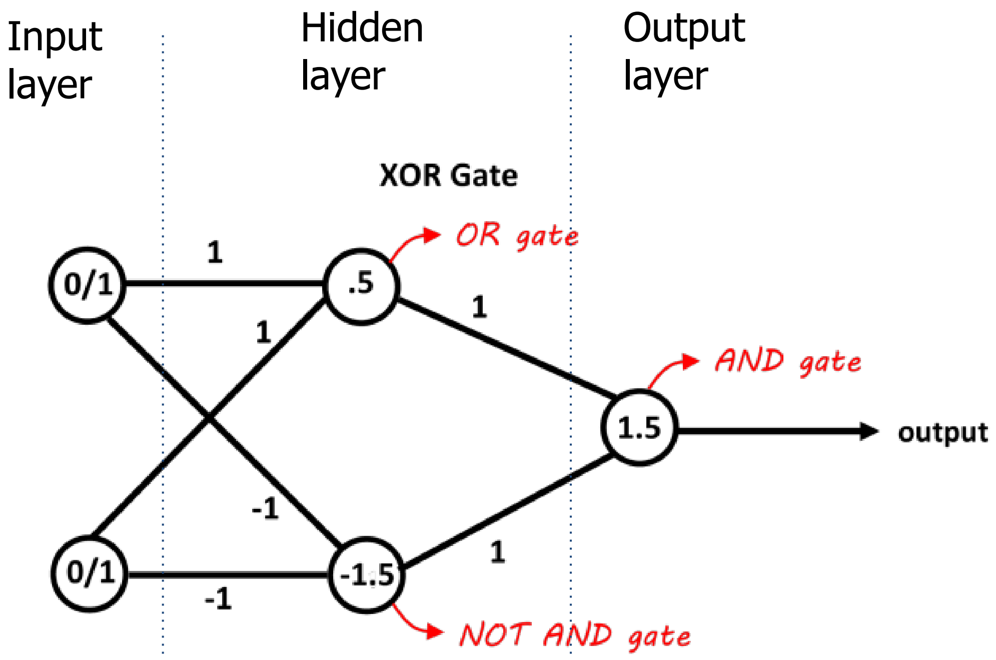
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

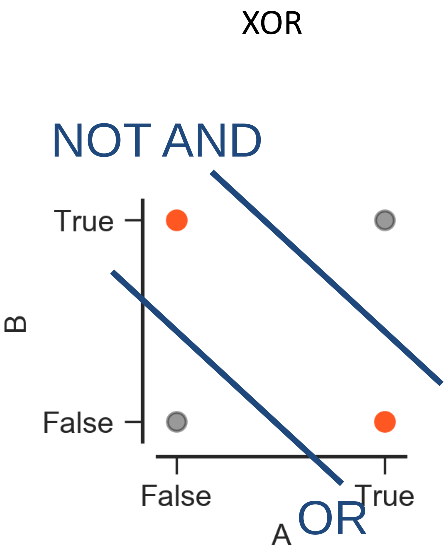
The XOR affair in
Perceptrons (1969)



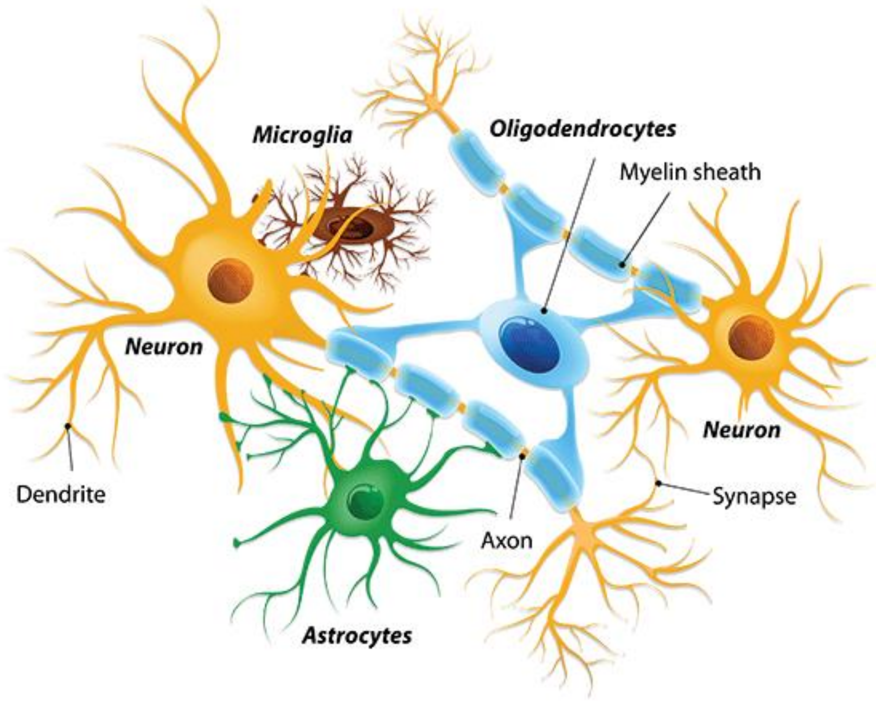


ROSENBLATT
died in a boat accident in 1971

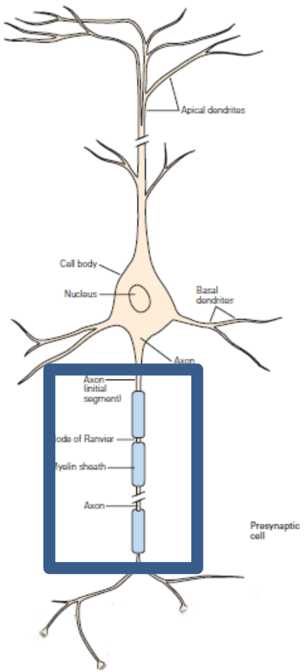


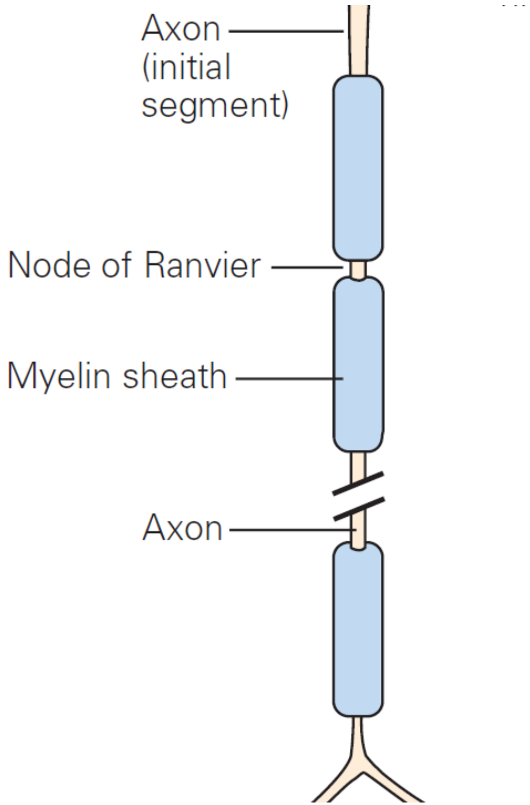


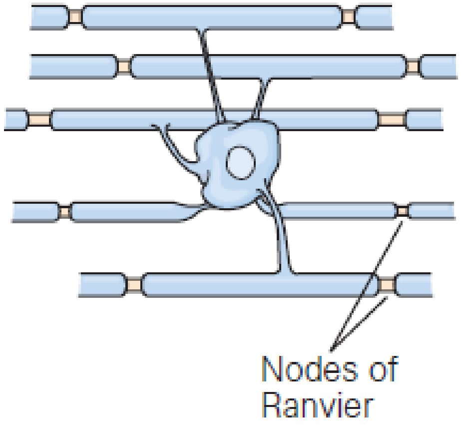
https://medium.com/@vnohitha13/neural-networks-52bec25688eb

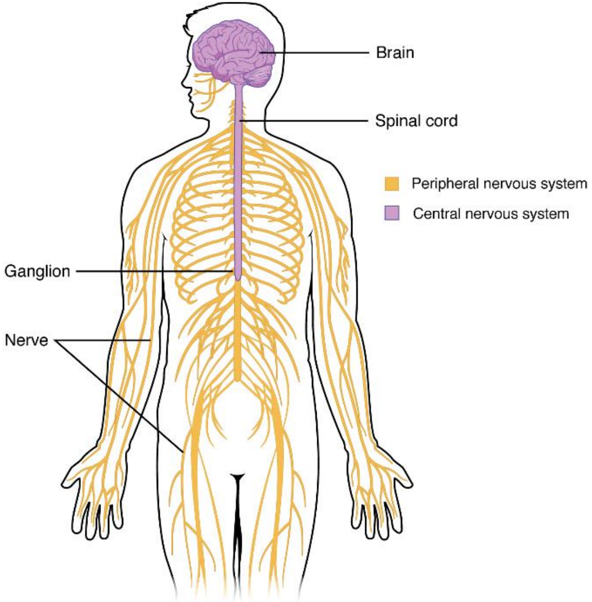


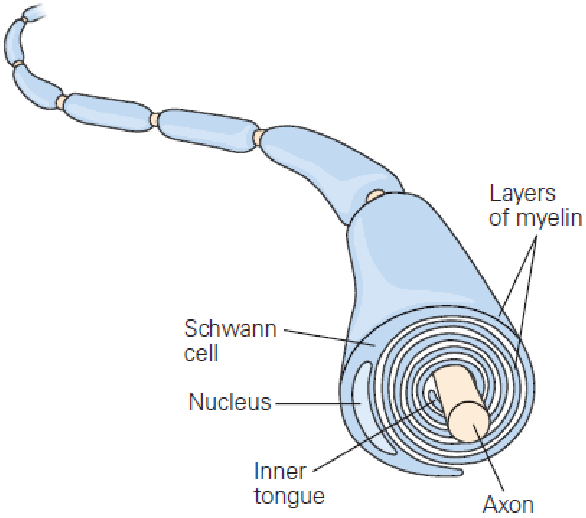
https://www.sciencenewsforstudents.org/article/scientists-say-glia


Apical dendrites
Cell body
Nucleus
Basal dendrites
Axon
Axon (initial segment)
Axon
Presynaptic cell


Axon (initial segment)
Node of Ranvier
Myelin sheath
Axon





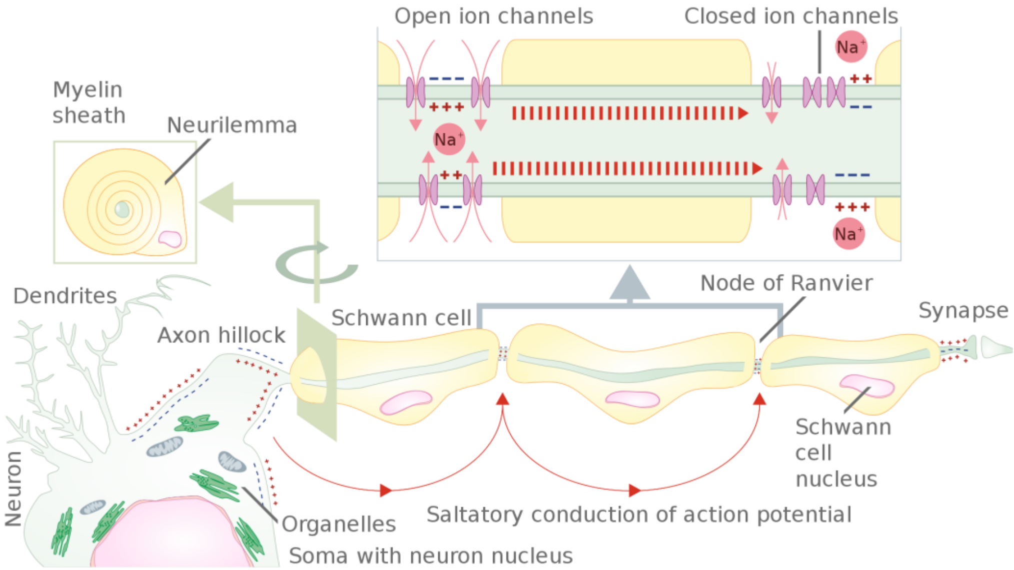


Oligodendrocytes
(Central nervous system)

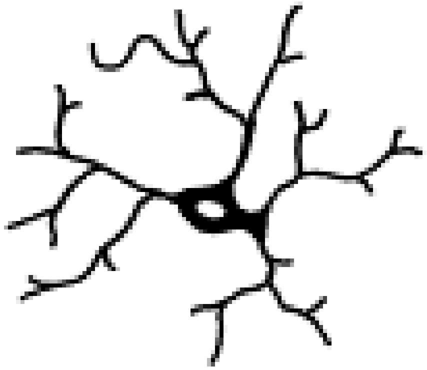
Schwann cells
(Peripheral nervous system)

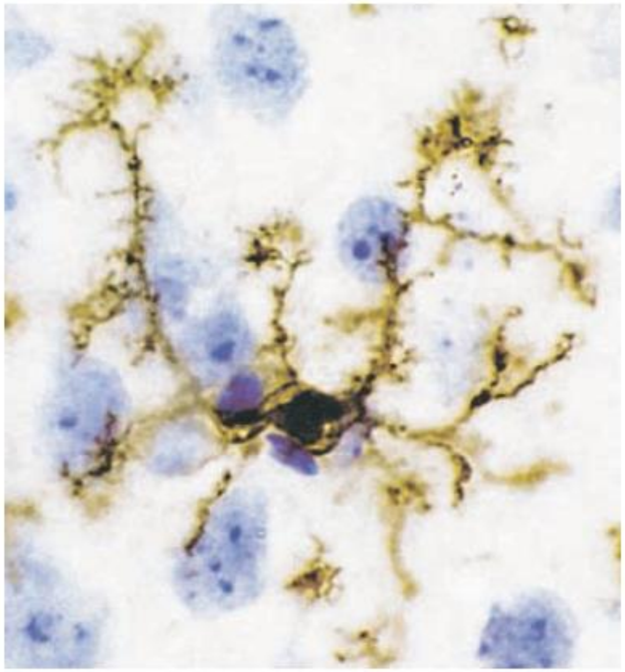
https://open.oregonstate.education/aandp/chapter/12-1-structure-and-function-of-the-nervous-system/

Saltatory connection



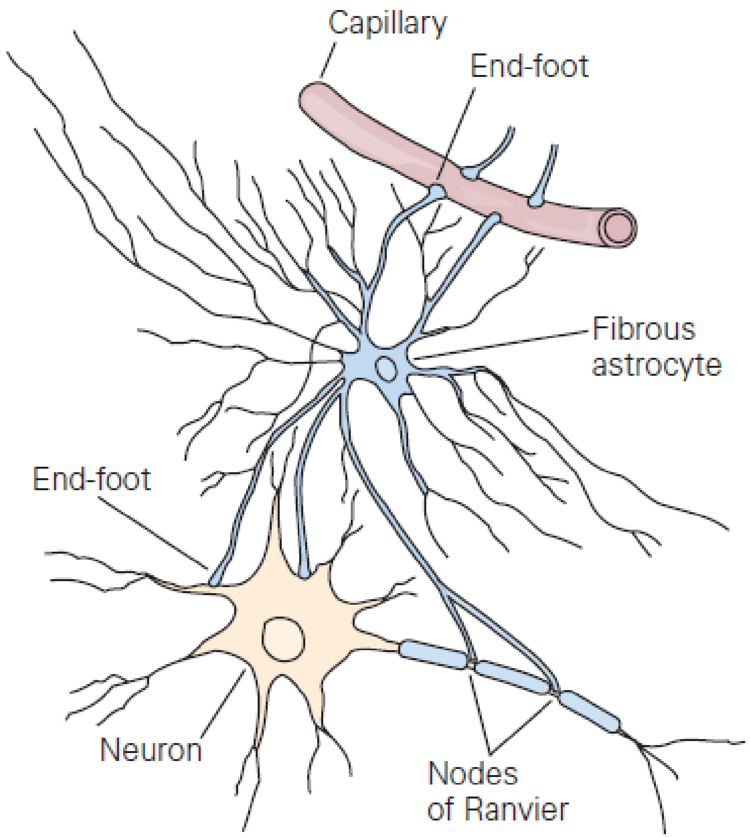
Microglia



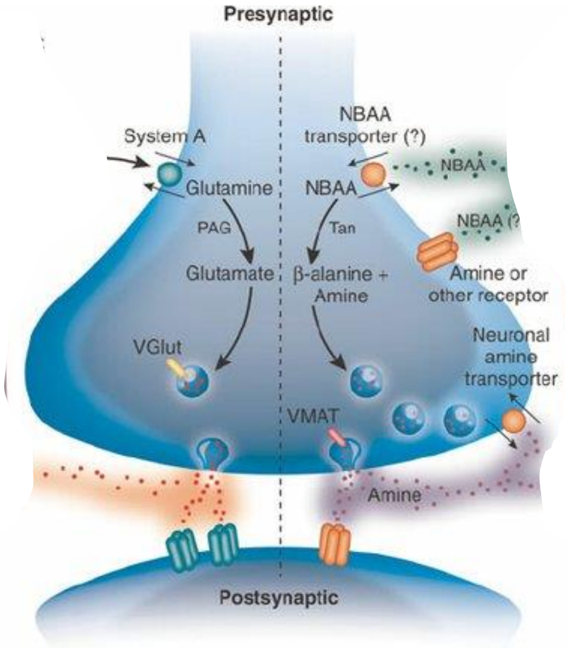


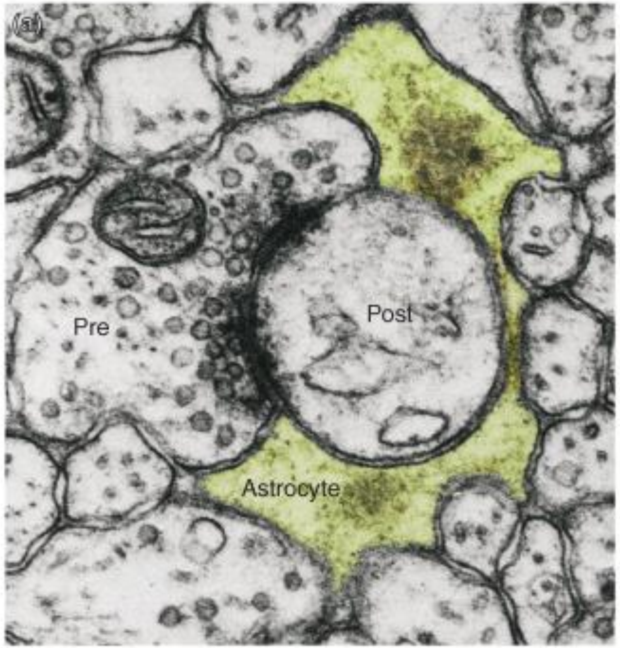
Cuadros and Navascués, Progress in Neurobiology 1998

Astrocytes


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

The tripartite synapse

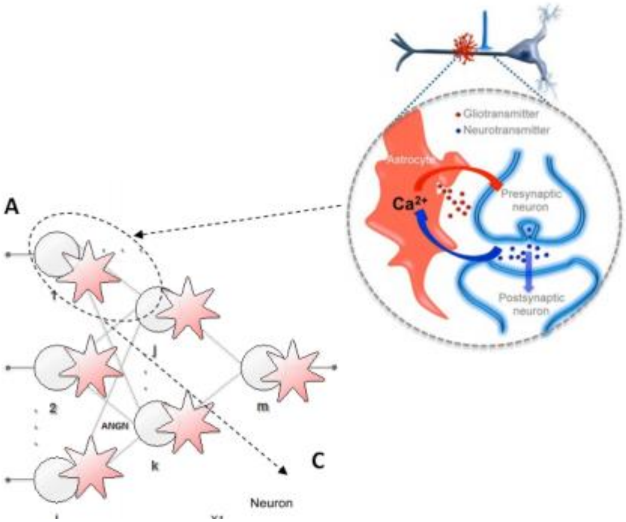


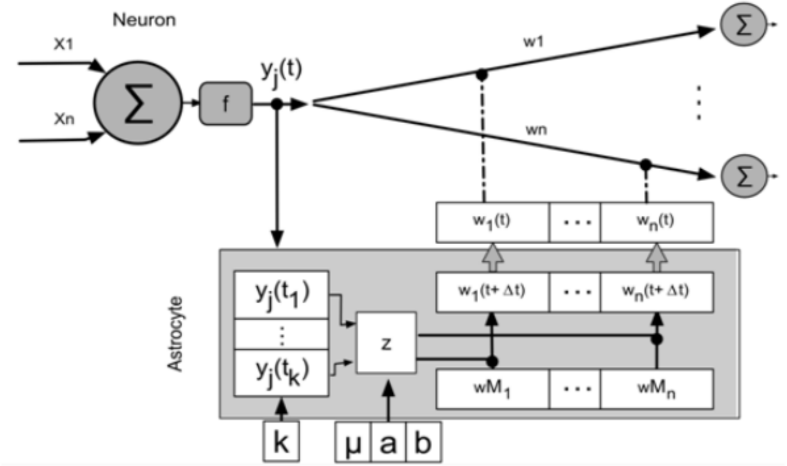


Jackson, F. R. & Haydon, P. G. Glial cell regulation of neurotransmission and behavior in Drosophila. Neuron Glia Biol. 4, 11-17

Halassa et al., Trends in Molecular Medicine 2007

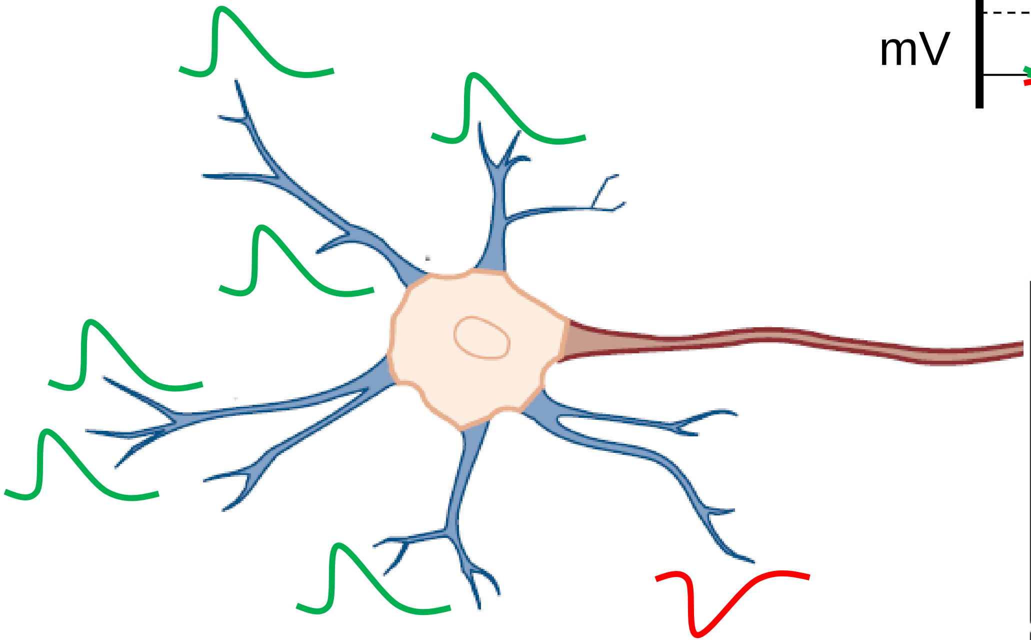
Using the tripartite synapse in Deep Learning

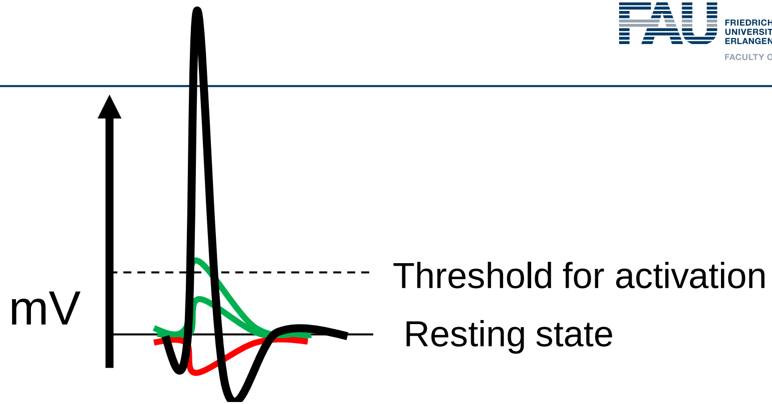


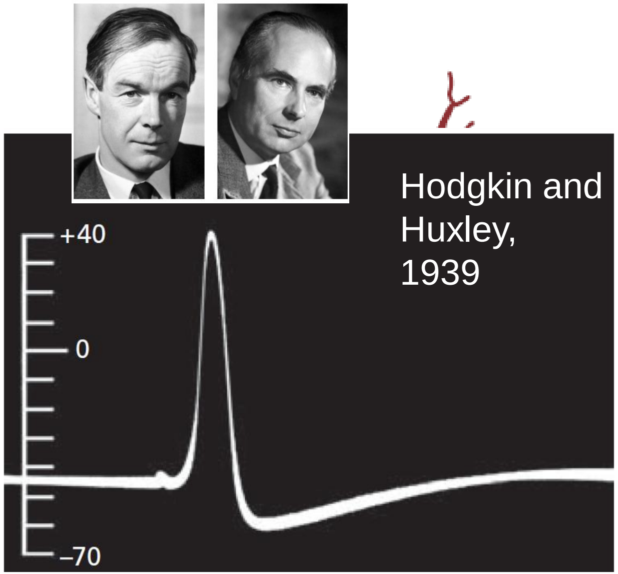


Mesejo et al., Int J Neural Syst 2015

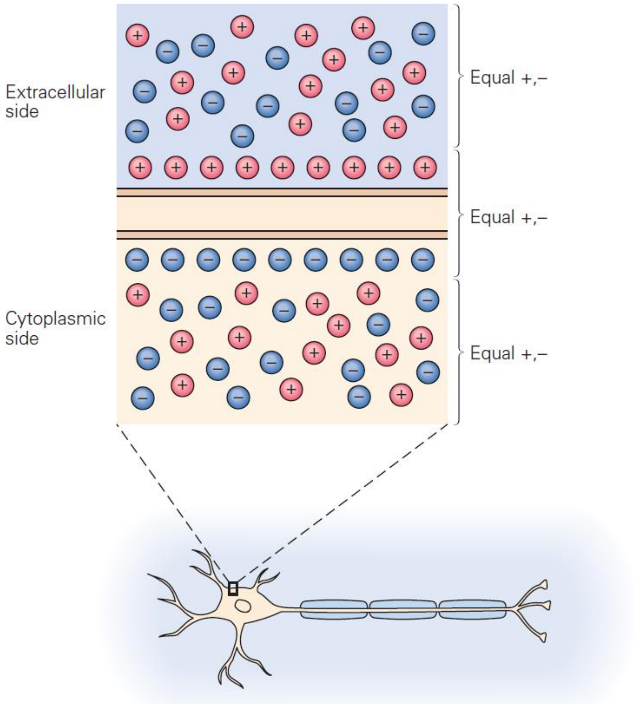
Neural computation



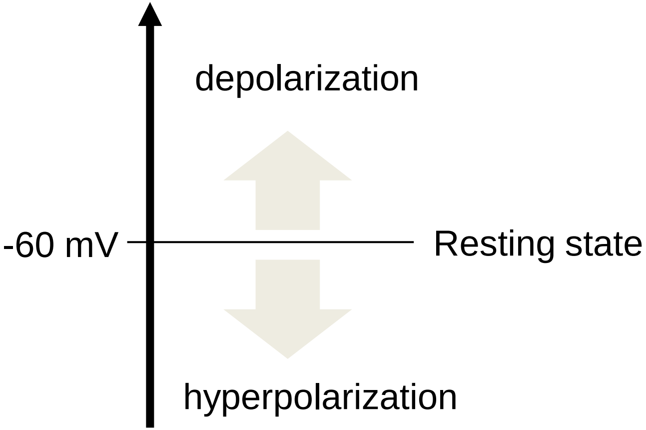


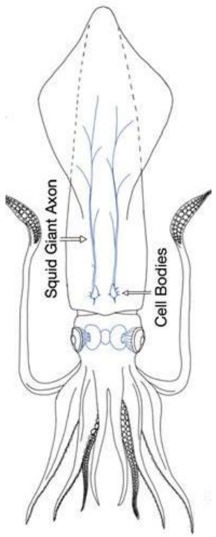


Ions and ion channels



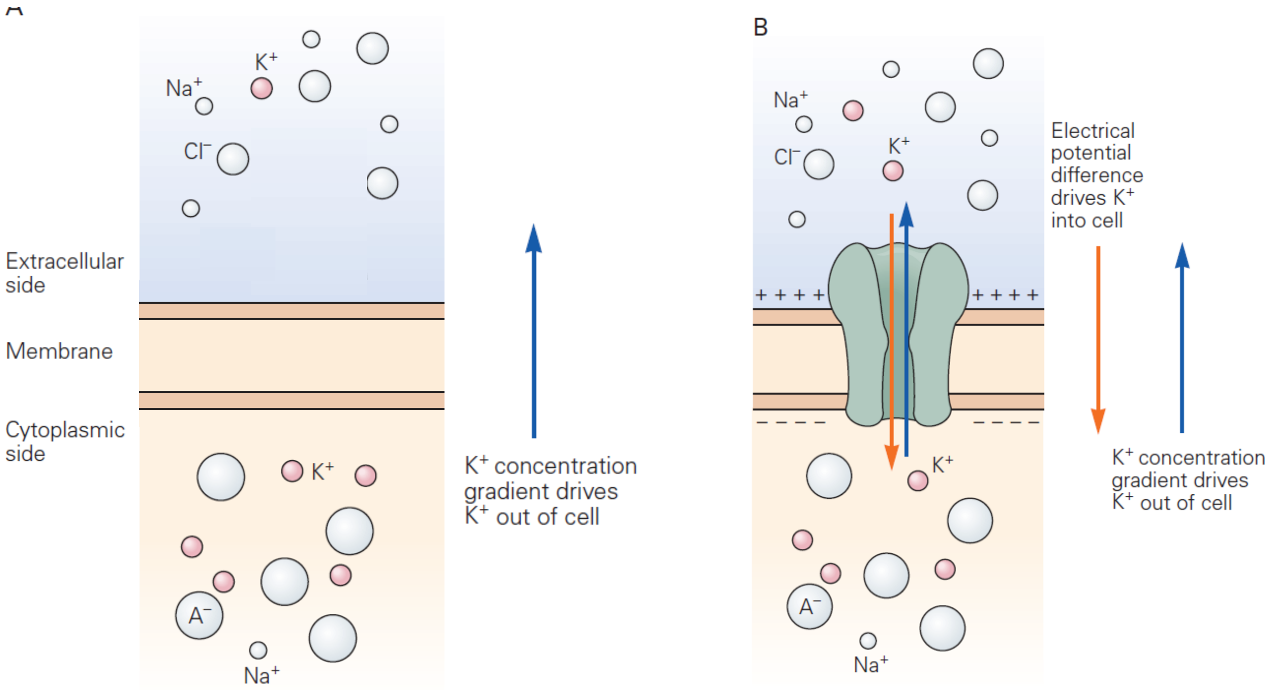
$$V_m = V_{in} - V_{out}$$





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

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

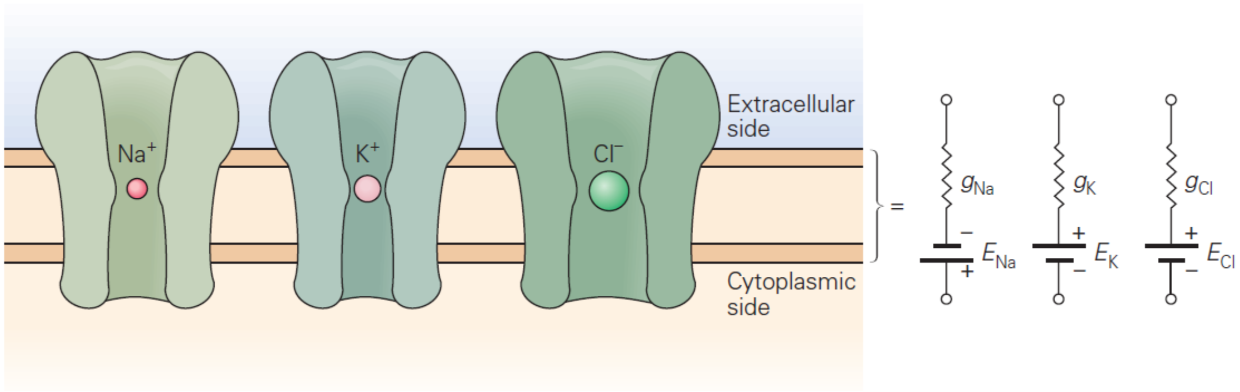
$$E_x = \frac{58\,\text{mV}}{z} \log \frac{[X]_o}{[X]_i}.$$

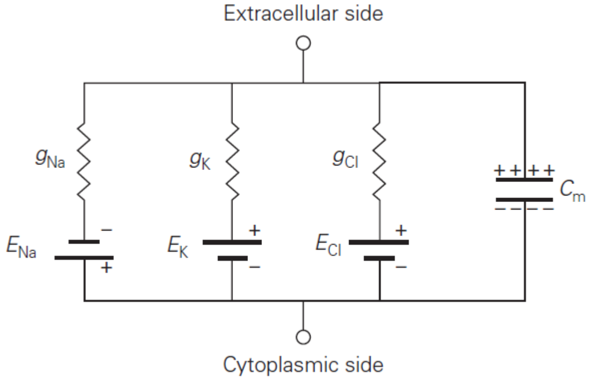
$$E_k = \frac{58\,\text{mV}}{1} \log \frac{[20]}{[400]} = -75\,\text{mV}.$$

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.

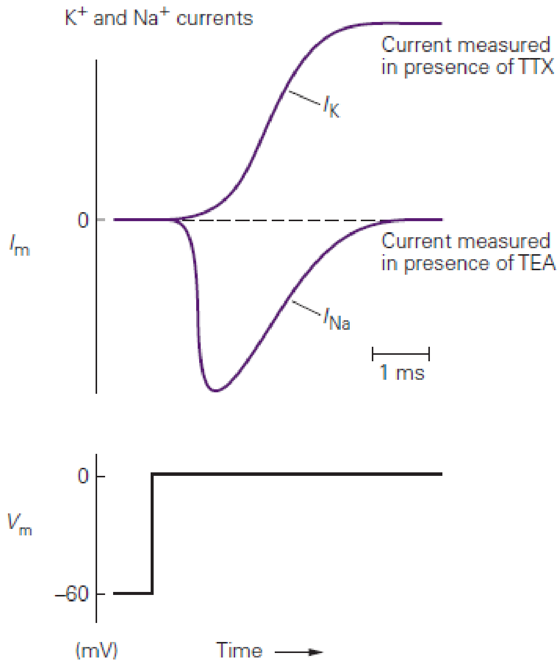
Biology explained as electrical circuit



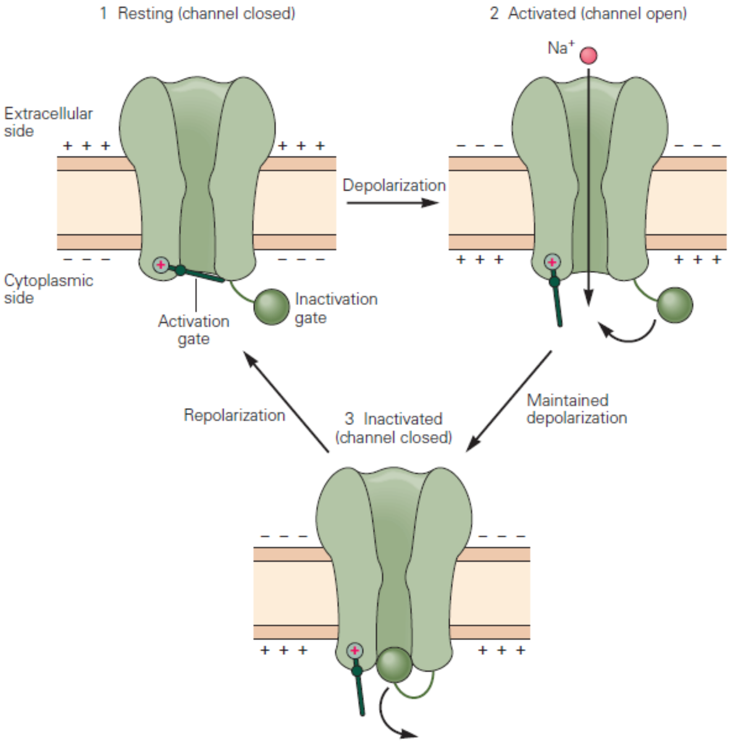


$$I_{\mathrm{Na}} + I_{\mathrm{K}} = 0.$$

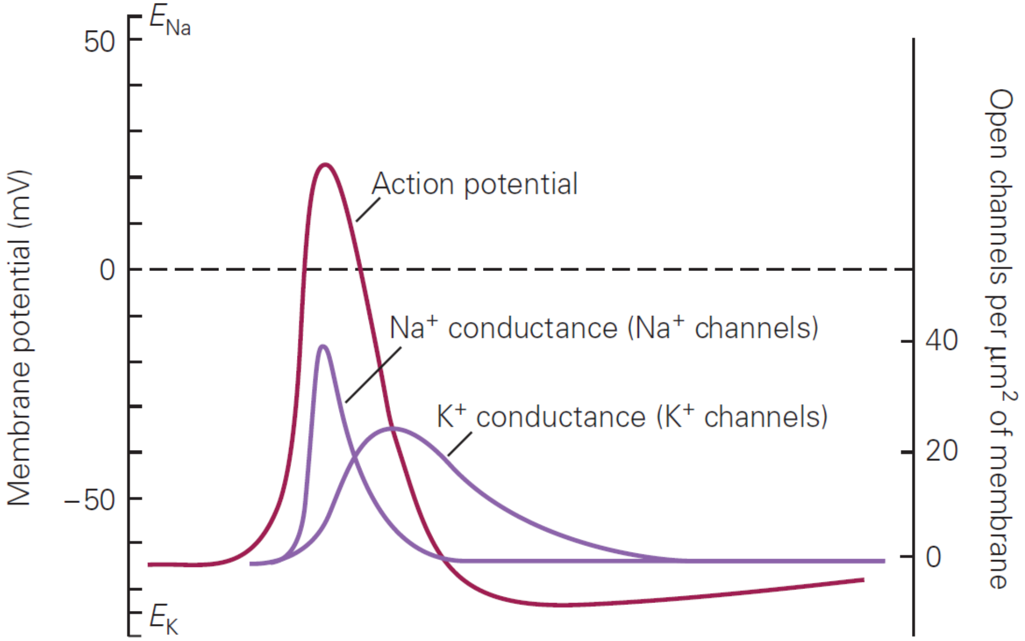
$$V_{\mathrm{m}} = \frac{(E_{\mathrm{Na}} \times g_{\mathrm{Na}}) + (E_{\mathrm{K}} \times g_{\mathrm{K}})}{g_{\mathrm{Na}} + g_{\mathrm{K}}}$$



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

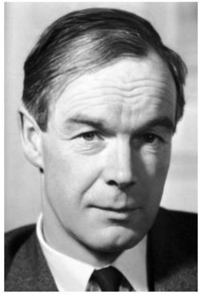


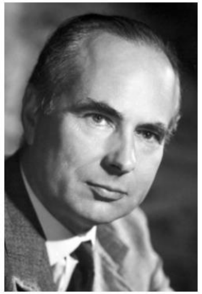
1 Resting (channel closed)
2 Activated (channel open)
Na+
Extracellular side
Cytoplasmic side
Depolarization
Activation gate
Inactivation gate
Repolarization
3 Inactivated (channel closed)
Maintained depolarization

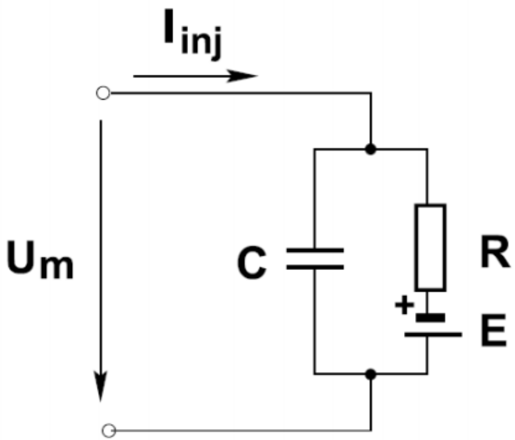


E_{Na}
50
0
−50
E_K
Membrane potential (mV)
Action potential
Na^+ conductance (Na^+ channels)
K^+ conductance (K^+ channels)
40
20
0
Open channels per µm^2 of membrane

The Hodgkin-Huxley model

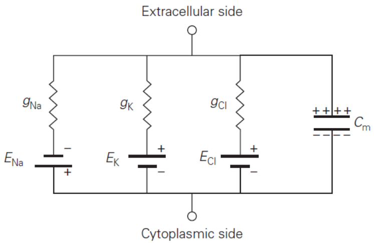






$$I_K = g_K(V_m - E_K)$$

Kirchhoff‘s law: conservation of electrical charge



$$C\frac{dV}{dt} = -\sum_{ion} I_{ion} + I(t).$$

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

$$I = C_m \frac{dV_m}{dt}$$

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

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

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

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

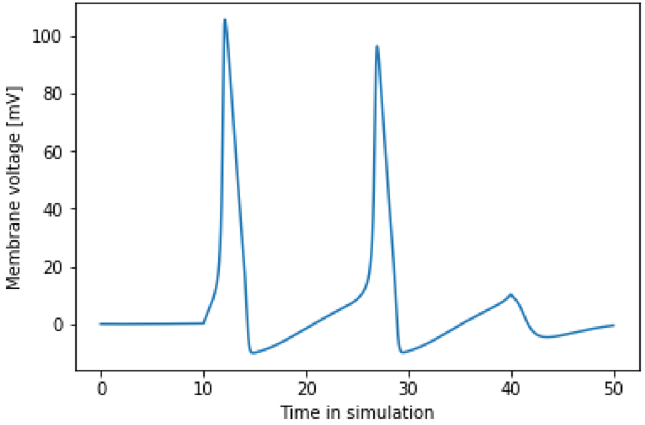
$$\frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m$$

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

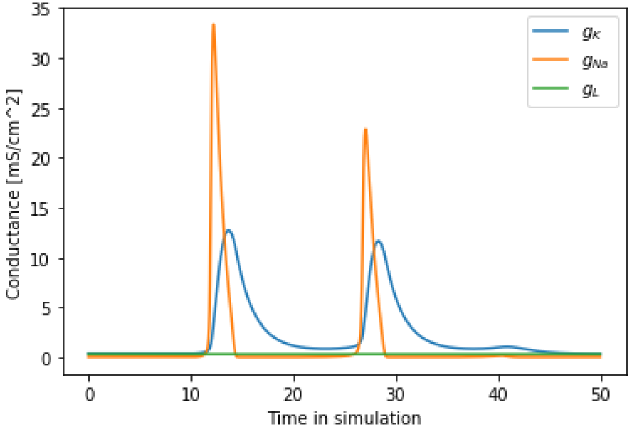
$$\bar{g}_{\mathrm{K}} n^4 (V_m - V_K) + \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{Na}) + \bar{g}_l (V_m - V_l),$$

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

[redacted] Applied external current



[redacted]

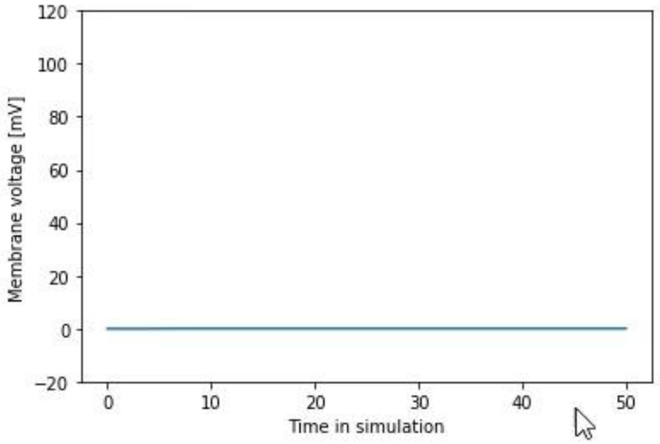


(code will be on StudOn that you can play with)

Changing the injected current

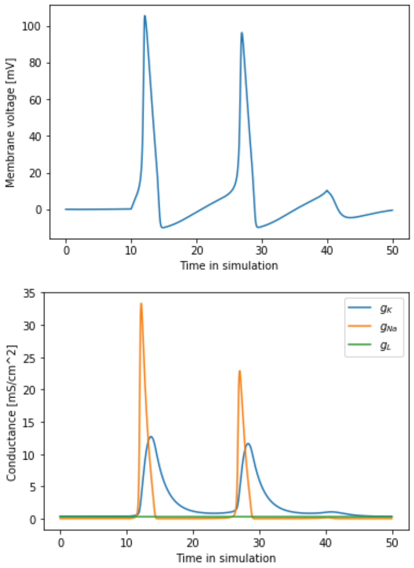

```python
@interact
def applyCurrent(I_ext_applied:(0,100,3)=0):
    plt.figure()
    plt.plot(ts, HHs[I_ext_applied])
    plt.xlabel("Time in simulation")
    plt.ylabel("Membrane voltage [mV]")
    plt.ylim([-20,120])
```

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

$$C_{\mathrm{m}} \frac{dV}{dt} = \frac{V_{\mathrm{rest}} - V}{R_{\mathrm{m}}} + I$$

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

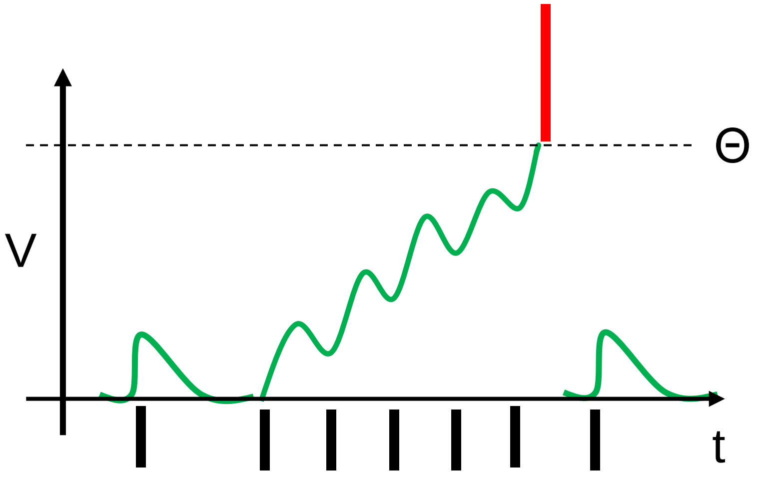
Integrate-and-Fire-Models

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

$$\tau_m = RC$$

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

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
```

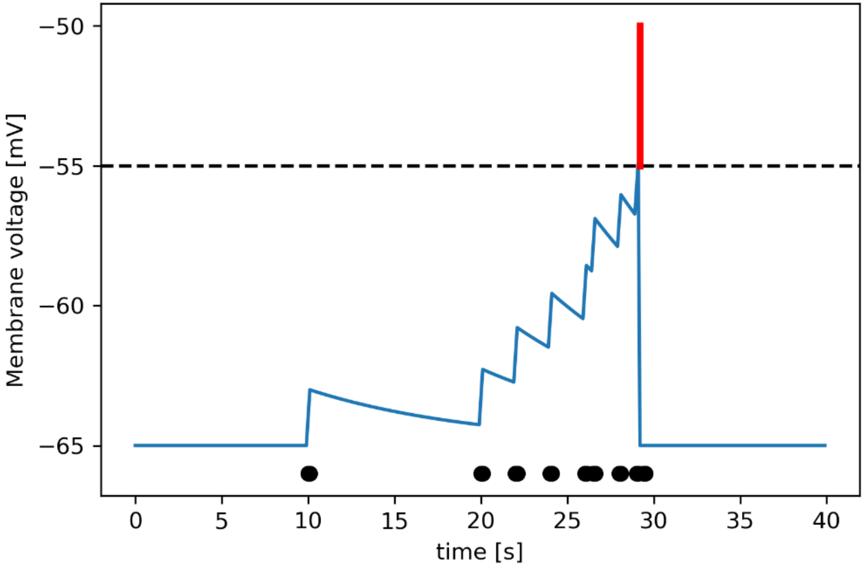
```python
# Check if action potential is elicited
s = V > theta

if s:
    V = Vm
```

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

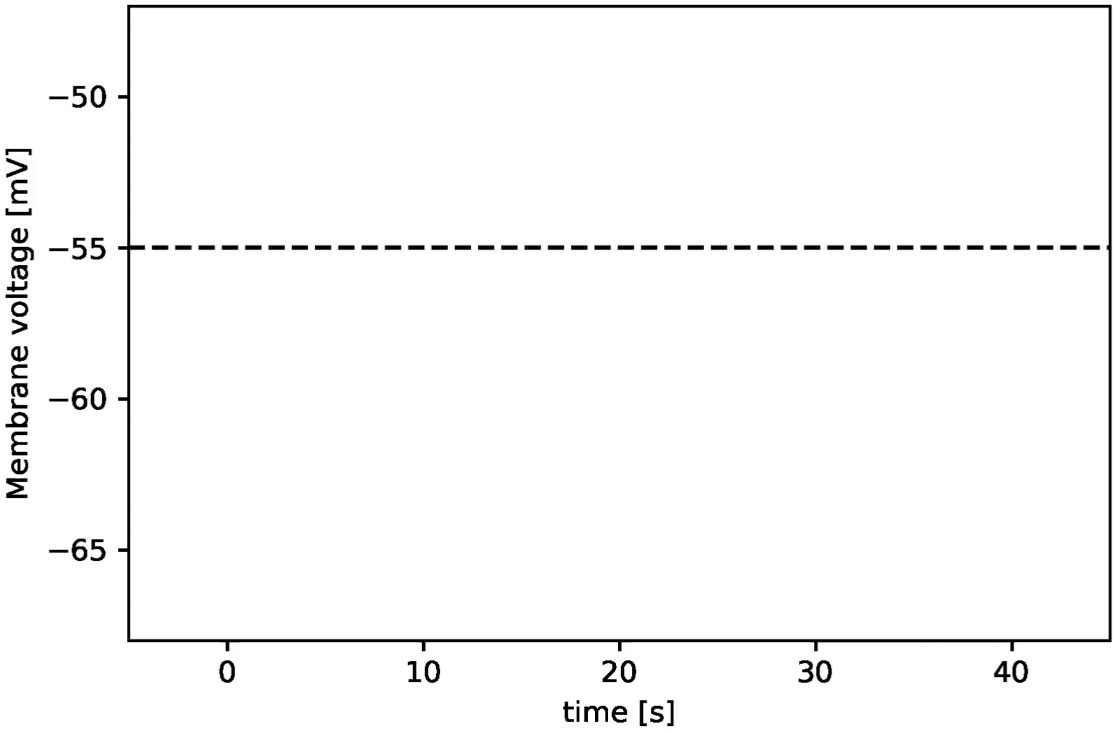
Simulation

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

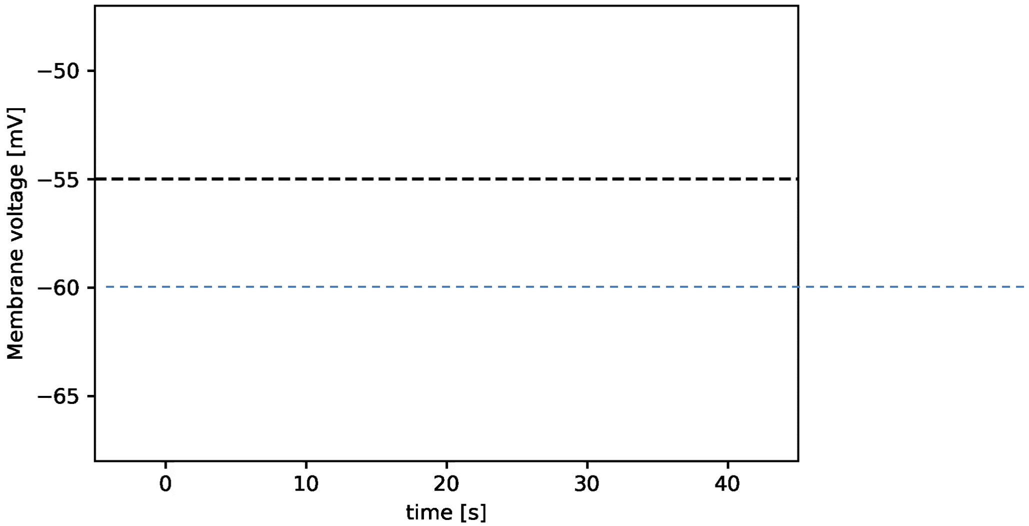


Temporal integration



Constant input current

5 mV

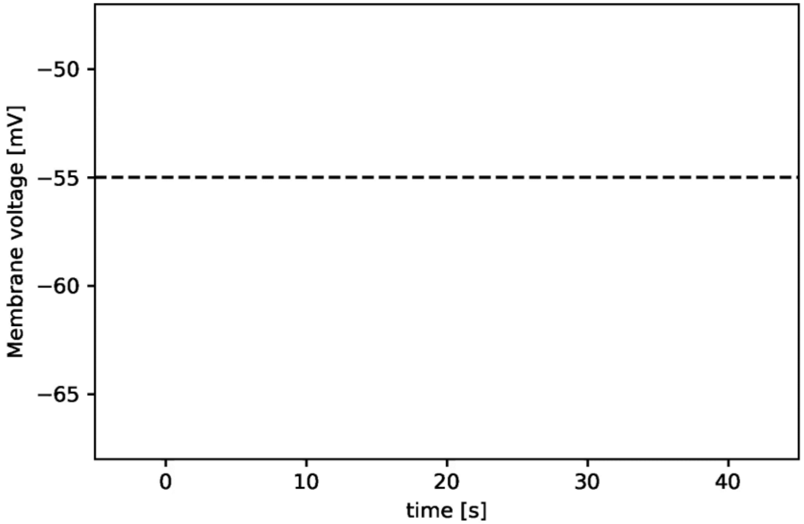


approaches
Vm+RI

with time
constant τ

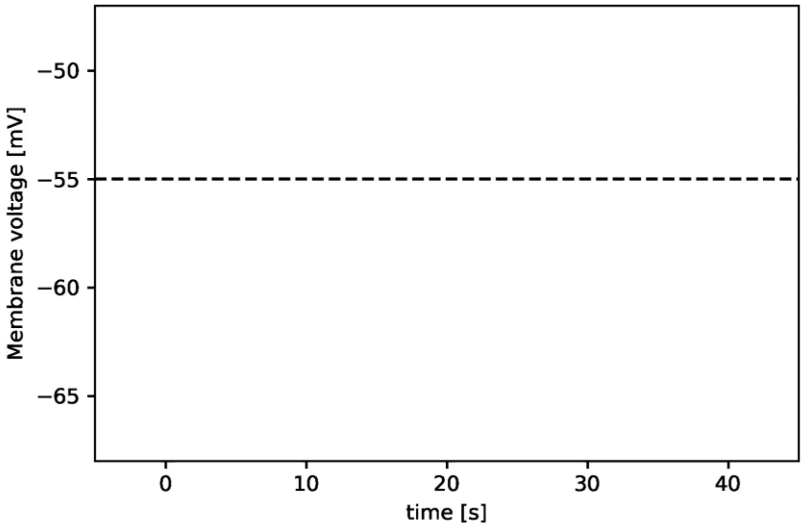
Constant input current

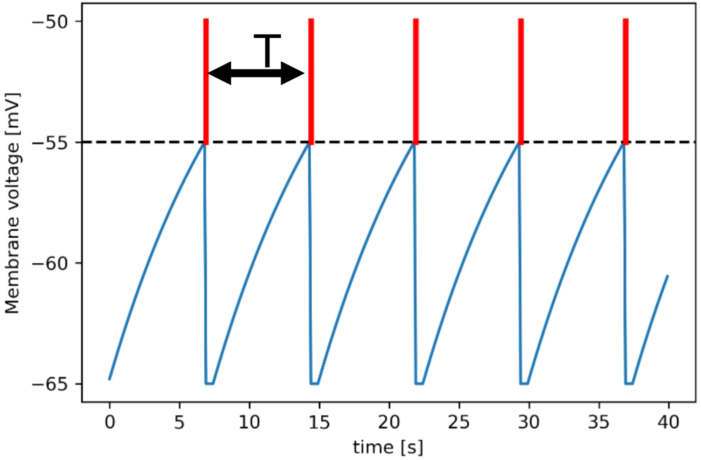
20 mV



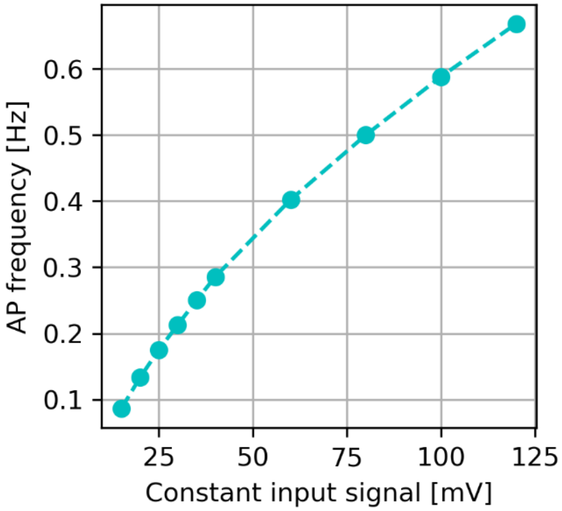
Constant input current

40 mV



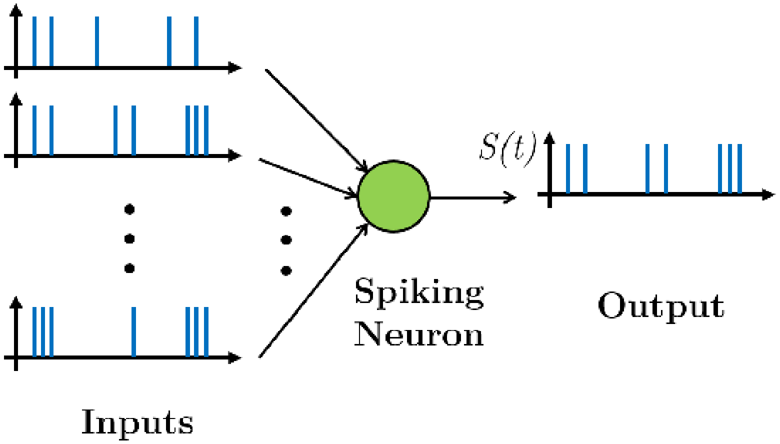

T
−50
−55
−60
−65
Membrane voltage [mV]
0
5
10
15
20
25
30
35
40
time [s]

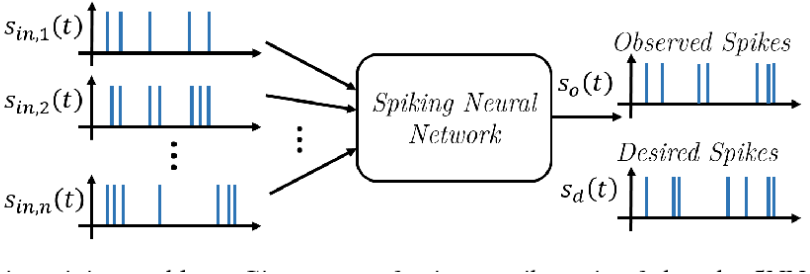
$f = 1/T$


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

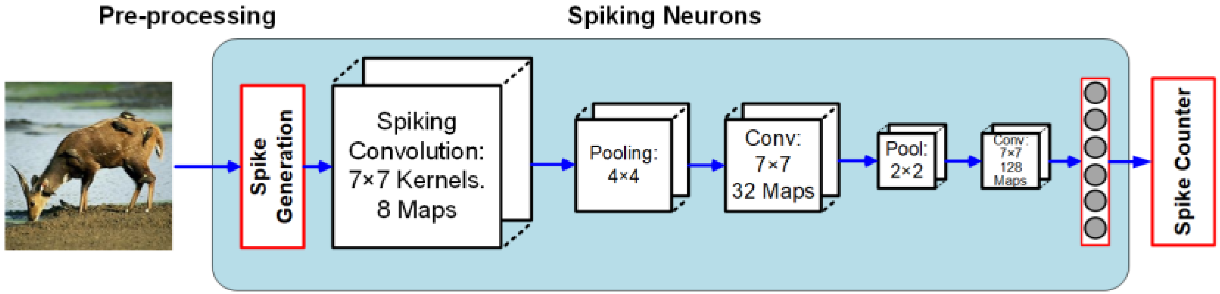
Why is this important?

Spiking Neural Networks

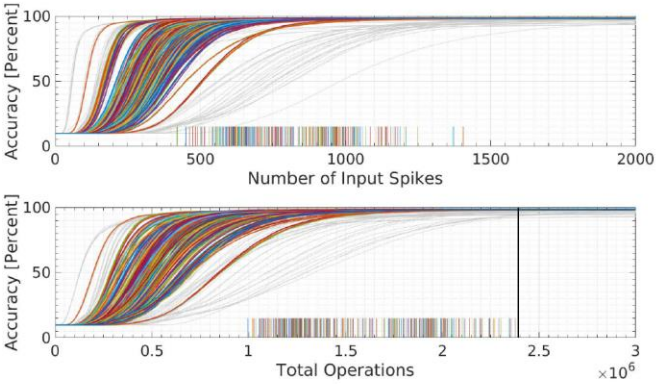




Spiking neural networks



Cao et al., Int J Comp Vision 2015



Neil et al., ACM Symposium on Applied Computing 2016