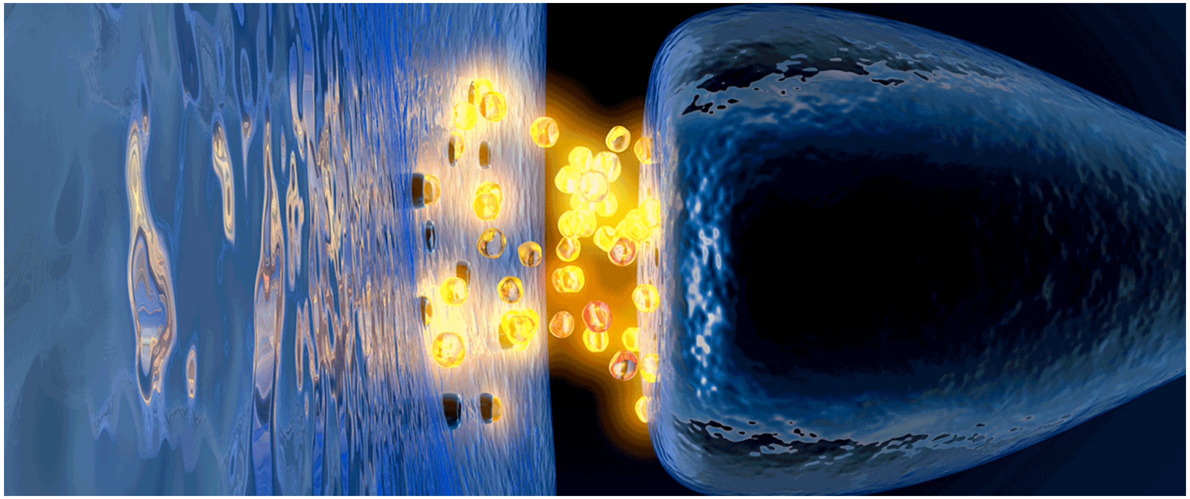
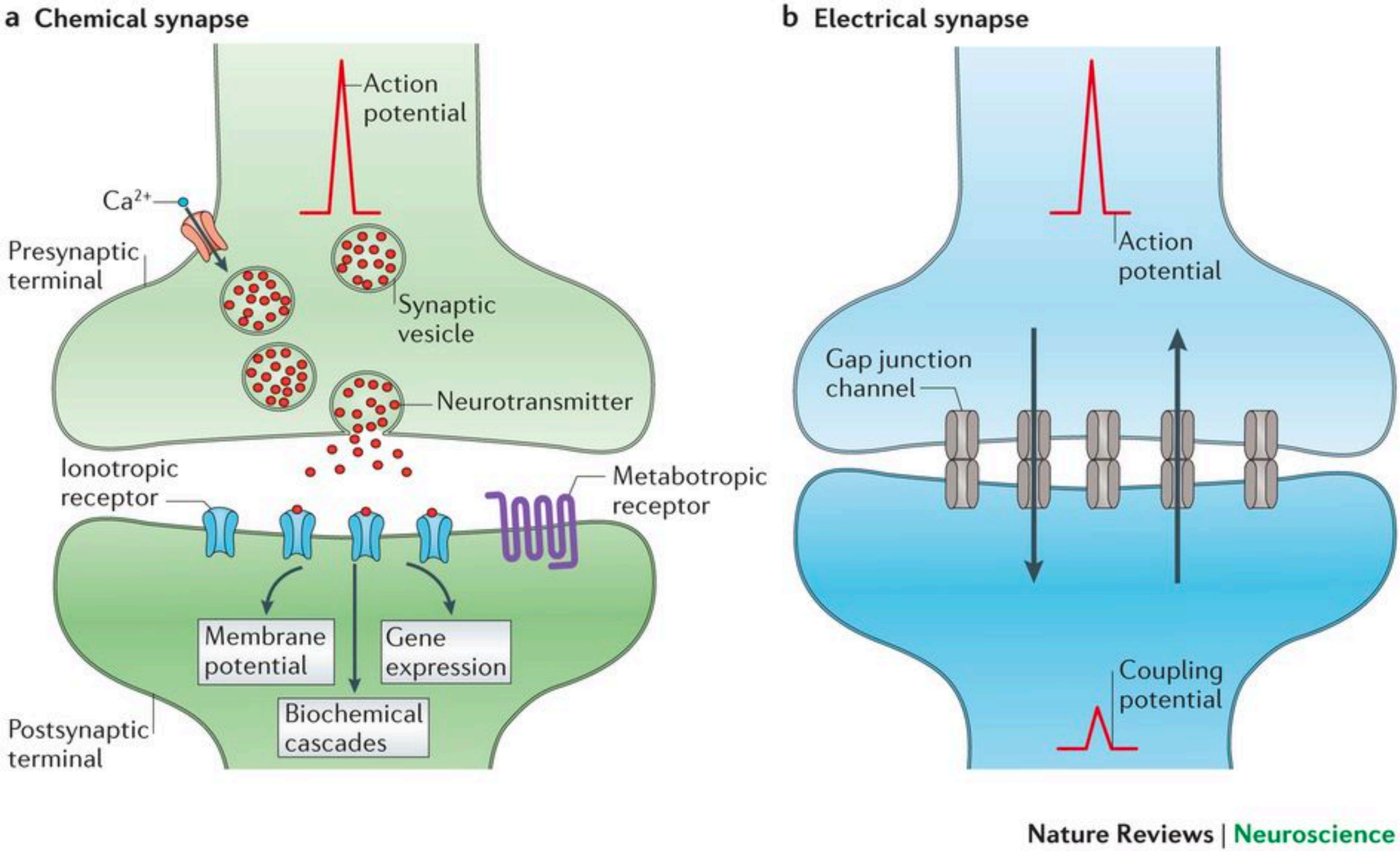
LASCON 2024

Tutorial: Modeling Synapses and Networks

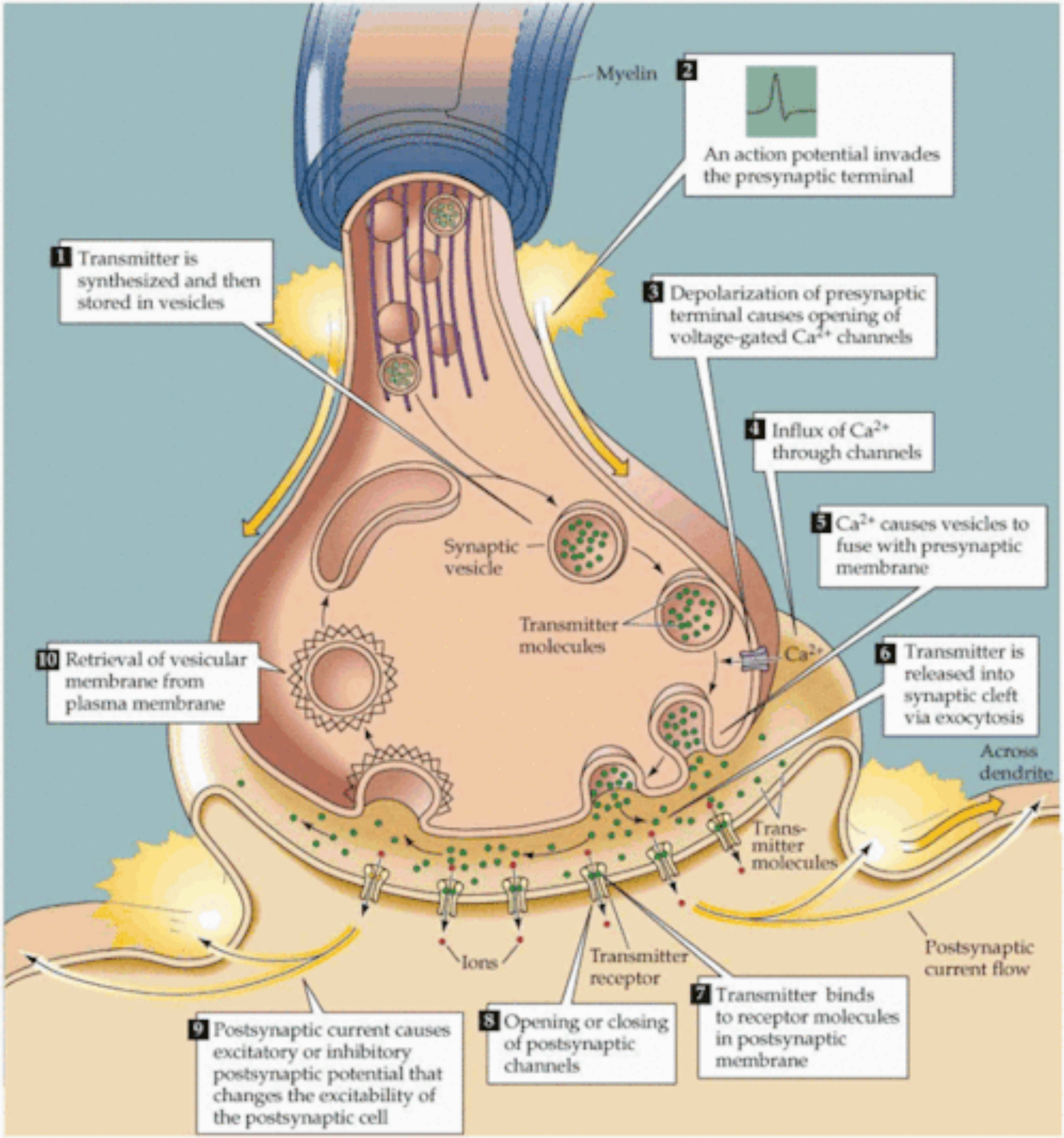


Arnd Roth and Valery Bragin

Types of Synapses



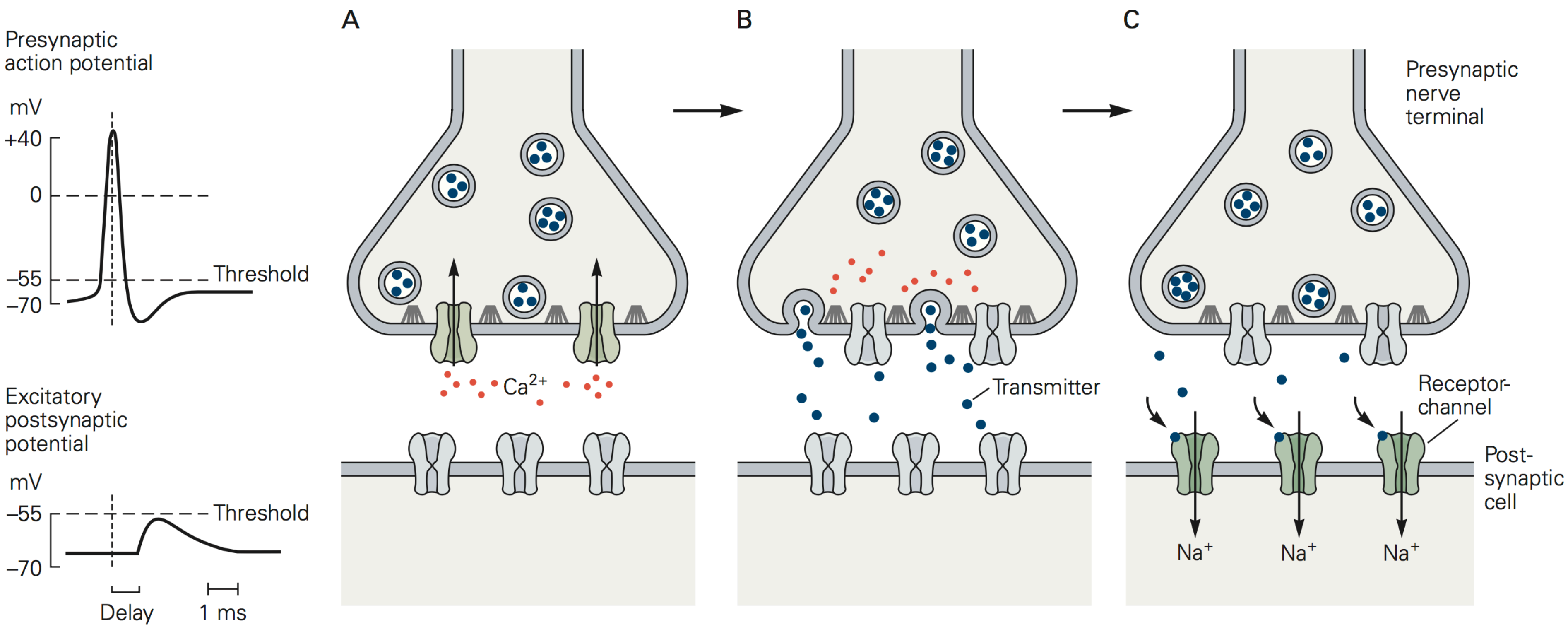
Chemical Synapse



https://www.youtube.com/watch?v=1nPGKqbeBHo

https://www.youtube.com/watch?v=mItV4rC57kM

Chemical Synapse

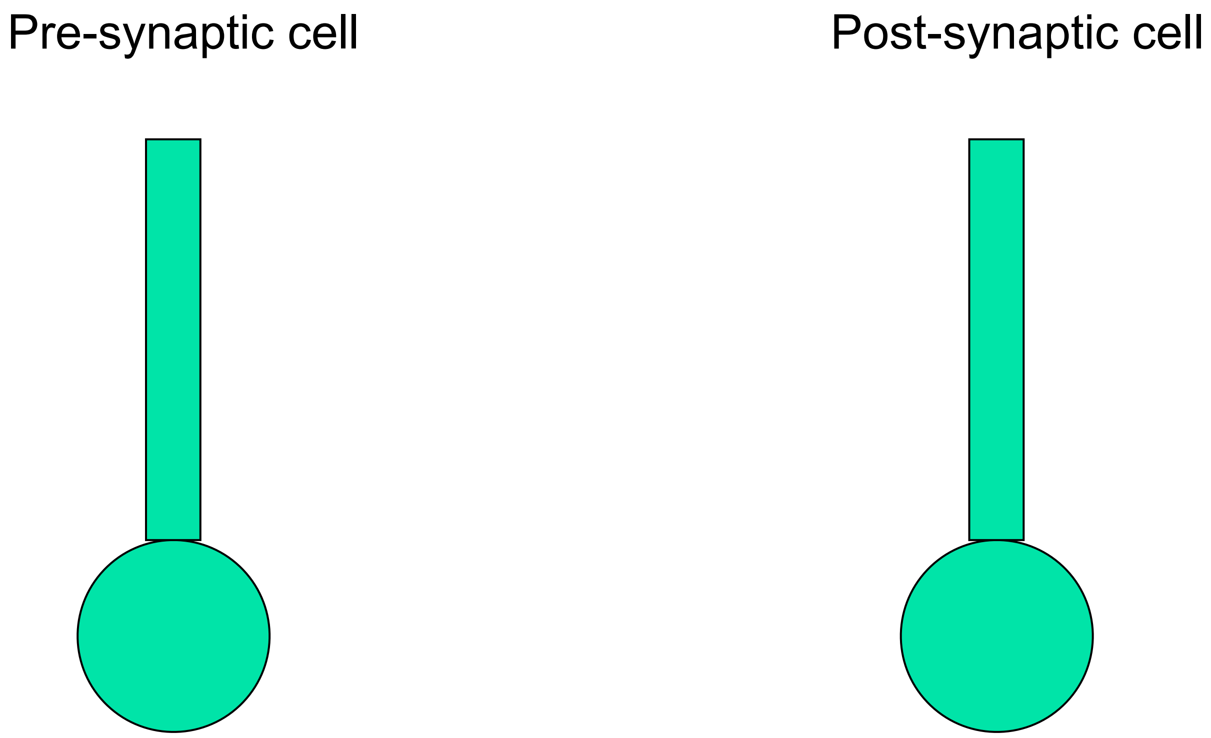


Postsynaptic potentials

- Binding of neurotransmitters to the post synaptic membrane can either be:
- **Excitatory (produces EPSP)**
  - Depolarizes membrane potential
- **Inhibitory (produces IPSP)**
  - Hyperpolarizes membrane potential

Both types of PSPs are *graded potentials* (as opposed to action potentials)

Synaptic connection model



Synaptic connection model

Pre-synaptic cell

Post-synaptic cell

NEURON Synapse
(Ionotropic receptor)

Params:

- *e*: equilibrium potential (mV)
- *tau*: time constant (ms)

Function:
Changes channel conductance (*g*) when event received (presyn spike).

Generates current that moves *V* towards *e* :

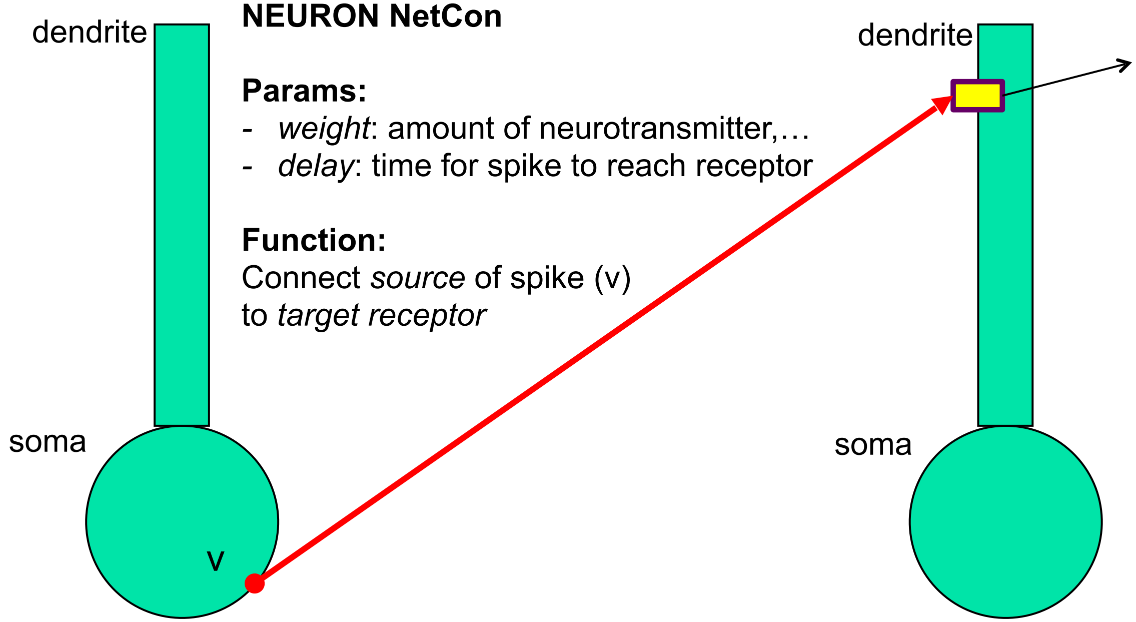
- if $e > V_{rest}$ → excitatory synapse (depolarizing)

- if $e < V_{rest}$ → inhibitory potential (hyperpolarizing)

Synaptic connection model

Pre-synaptic cell

Post-synaptic cell



NEURON NetCon

Params:
- *weight*: amount of neurotransmitter,…
- *delay*: time for spike to reach receptor

Function:
Connect *source* of spike (v)
to *target receptor*

NEURON Synapse
(Ionotropic receptor)

Params:
- *e*: equilibrium potential (mV)
- *tau*: time constant (ms)

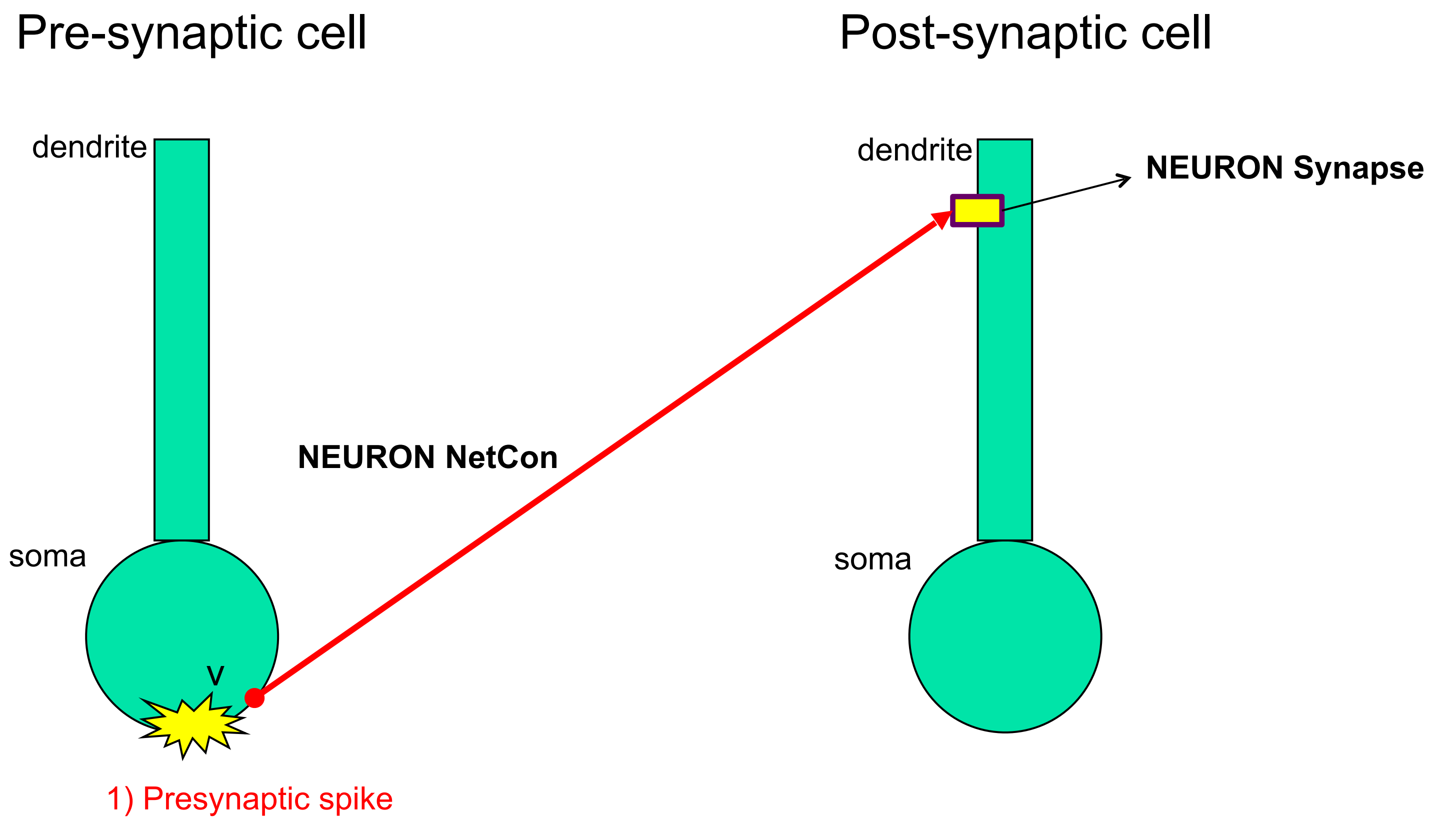
Function:
Changes channel conductance (*g*)
when event received (presyn spike).

Generates current that moves *V* towards *e* :

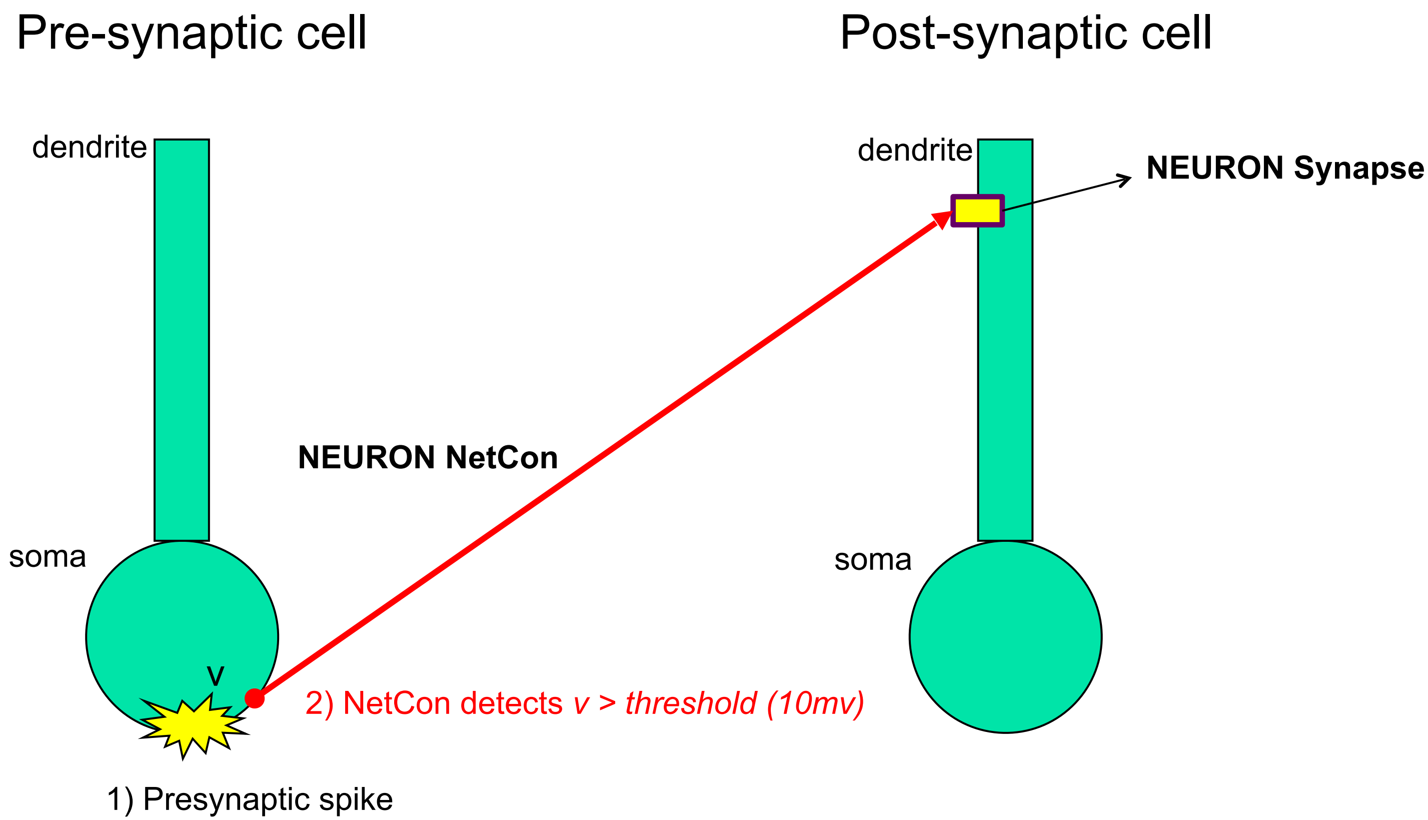
- if $e > V_{rest}$ → excitatory synapse (depolarizing)

- if $e < V_{rest}$ → inhibitory synapse (hyperpolarizing)

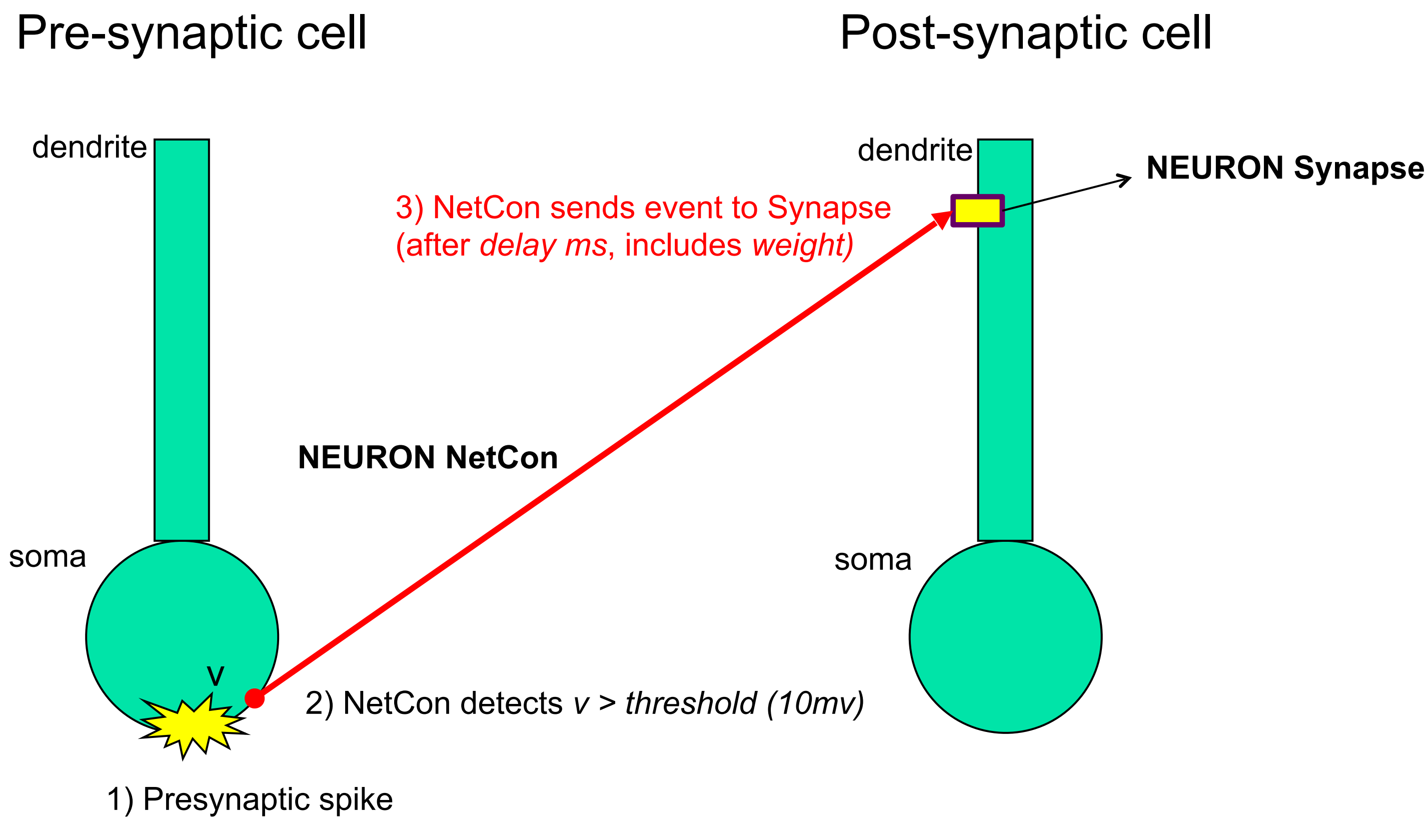
Synaptic connection model



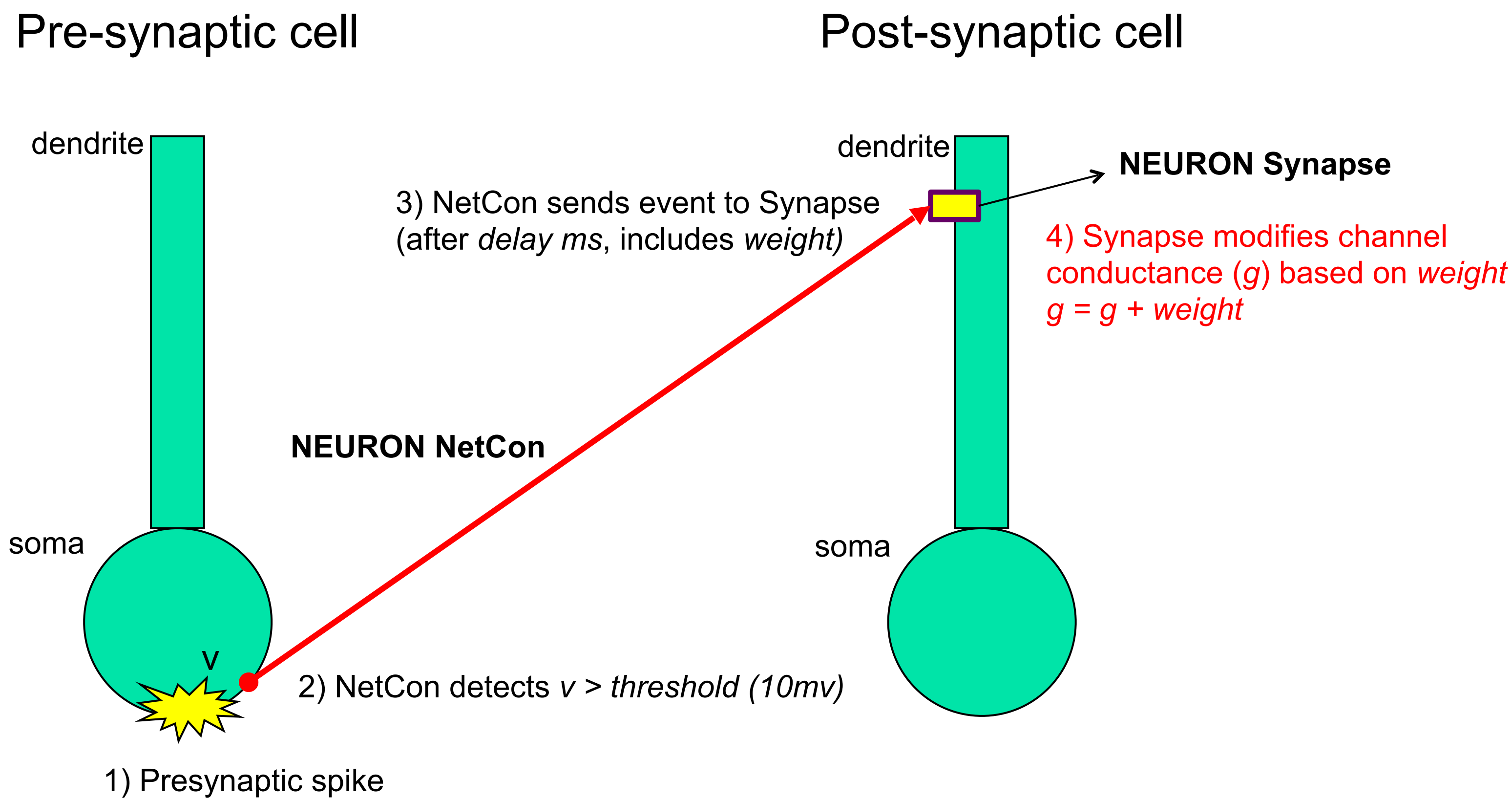
Synaptic connection model



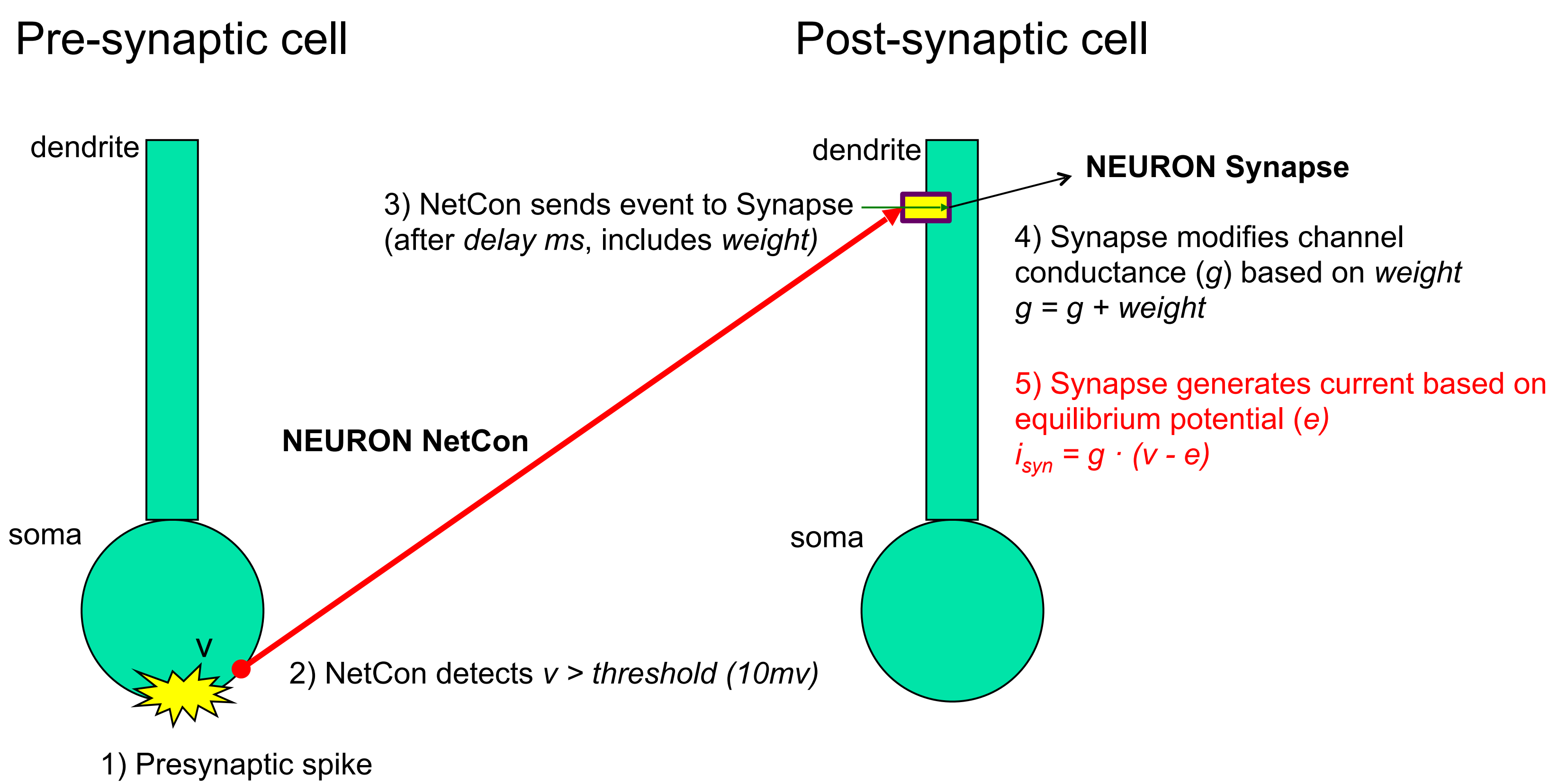
Synaptic connection model



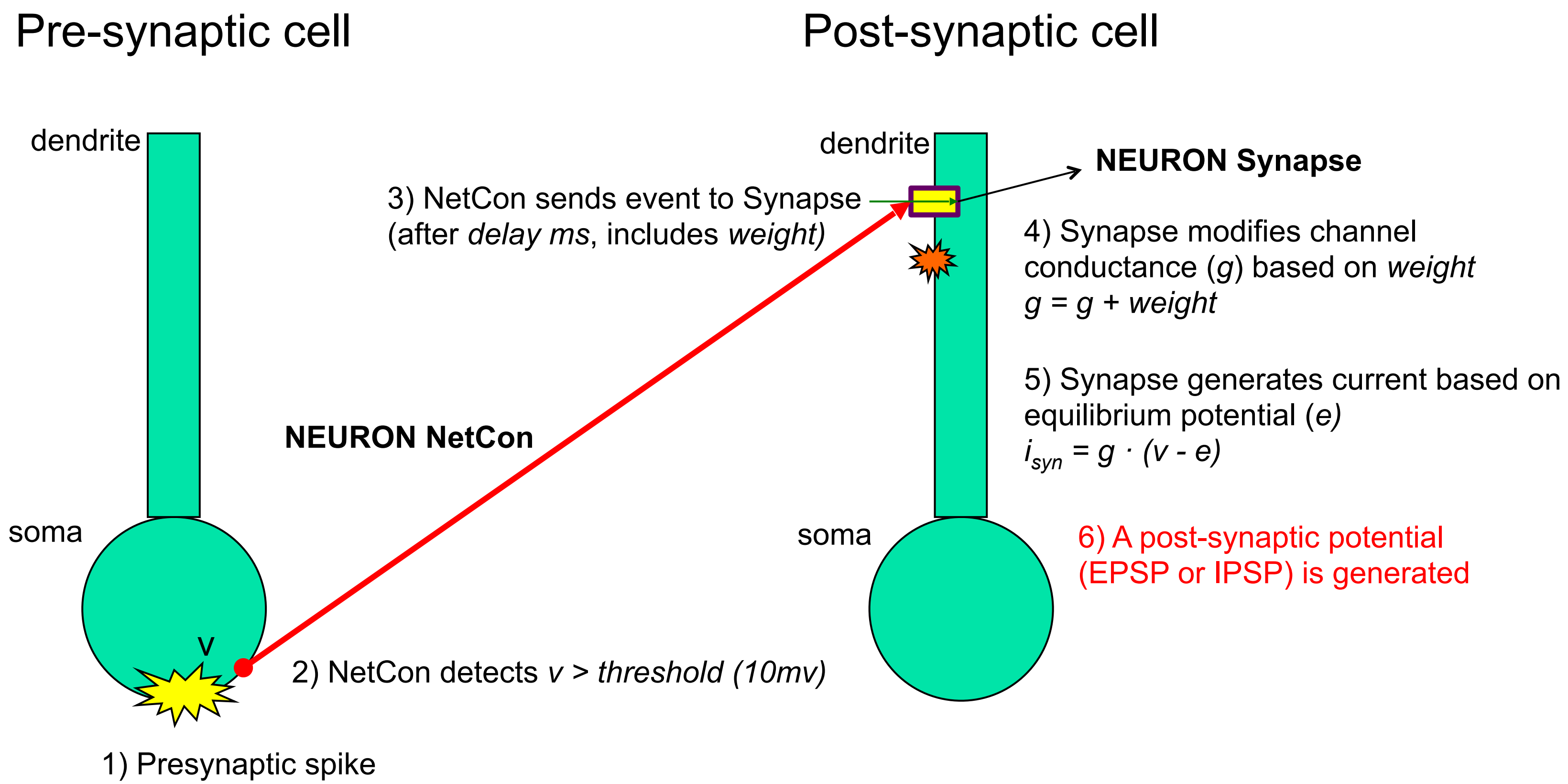
Synaptic connection model



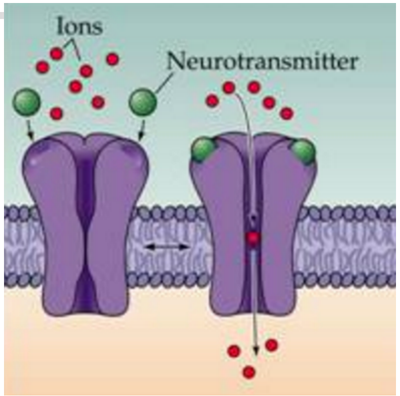
Synaptic connection model



Synaptic connection model



NEURON Synapse



- NEURON synapses correspond to the postsynaptic receptor
- Requires input event (presynaptic spike) to get activated
- If input event, modifies channel conductance (*g*) based on *weight:*
  $g = g + weight$
- Synapse generates current based on equilibrium potential (*e*)*:*
  $i_{syn} = g \cdot (v - e)$
- Synaptic current generates a post-synaptic potential:
  - if $e > V_{rest}$ → excitatory synapse (EPSP)
  - if $e < V_{rest}$ → inhibitory synapse (IPSP)

NEURON Synapses

- AlphaSynpase
  - eg. *syn = h.AlphaSynapse(dend(0.5))*
  - Artificial synapse – doesn't receive input from presynaptic cell (activated 'manually')
  - Parameters:
    - *onset*: active at this time (ms), eg. *syn.onset = 20*
    - *gmax*: max conductance reached after *onset+tau* (ms), eg. *syn.gmax = 0.1*
    - *tau*: time for conductance to reach *gmax,* eg. *syn.tau = 1*

- ExpSyn
  - eg. *syn = h.ExpSyn(dend(0.5))*
  - Connected to presynaptic cell via NetCon (requires event=spike to activate)
  - Parameters:
    - *e*: synapse equilibrium potential (mV). , eg. *syn.e = 0 or syn.e = -80*
    - *tau*: time for conductance to reach *gmax,* eg. *syn.tau = 1*

- More complex synapse models that use eg. Ca^{2+} concentrations

NEURON Netcons

Syntax:
con = h.NetCon(source voltage, target, sec = source section)

eg. *con = h.NetCon(soma(0.5)._ref_v, syn, sec=soma)*
con.weight = 0.1
con.delay = 1
con.threshold = 10

Description:
Network Connection object that defines a synaptic connection between a source and target.
When the source variable passes *threshold,* the target will receive an event at time t+*delay* along with weight information.

Source:
Normally a reference to a *membrane potential (v)* which is watched during simulation for v > threshold.

Target:
Must be a POINT_PROCESS (eg. synapse)

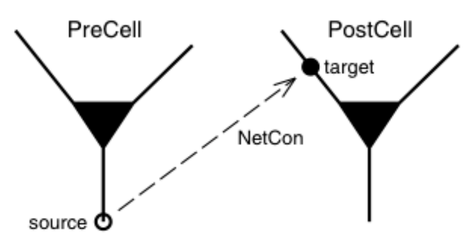


Figure 2.1. A NetCon attached to the presynaptic neuron PreCell detects spikes at the location labeled source, and delivers events to the synapse target which is attached to the postsynaptic neuron PostCell.

Netcons

Parameters:

- *delay*: time between action potential reaching axon terminal and
- *weight:* change of conductance in synaptic ionic channels (open/close channels)
  ~= amount of neurotransmitter binding to receptors * num receptors
  ~= amount of Ca^{2+} channels, concentration of Ca^{2+} ions

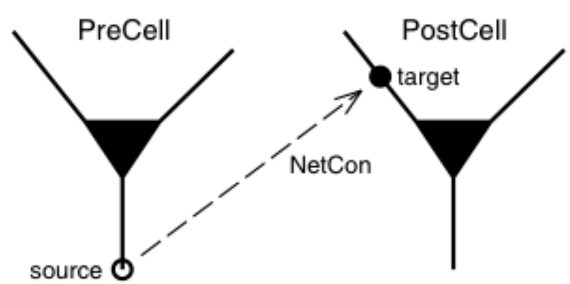
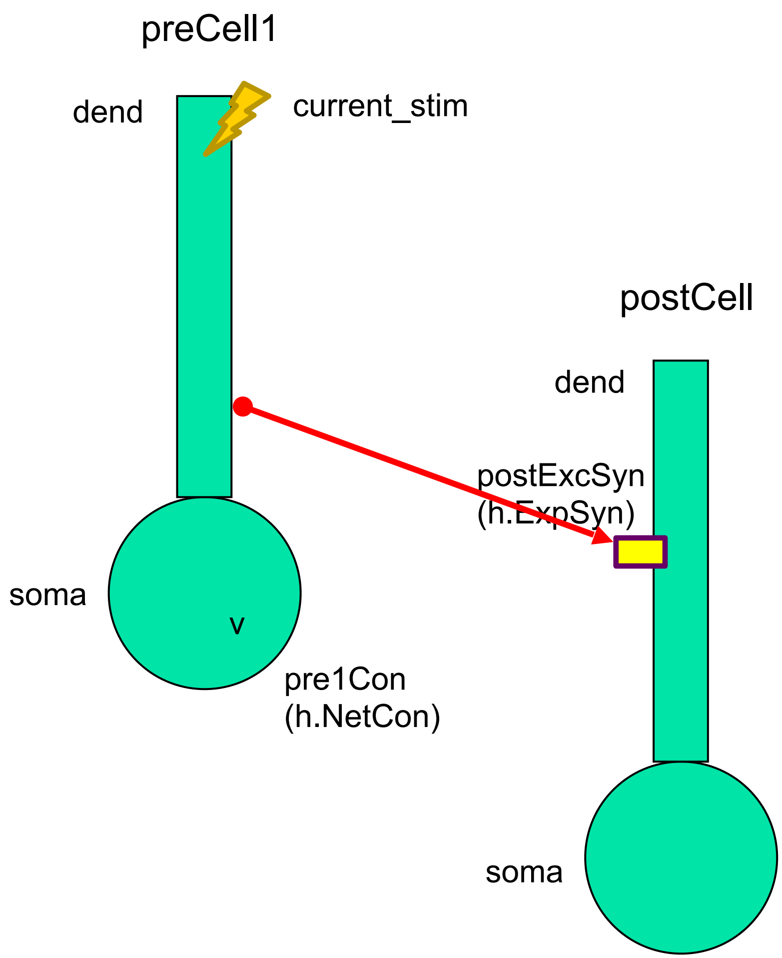


Figure 2.1. A NetCon attached to the presynaptic neuron PreCell detects spikes at the location labeled source, and delivers events to the synapse target which is attached to the postsynaptic neuron PostCell.

Connections Tutorial

1) Start with *tutorial_3.py.* It contains a single *postCell*, with no stimulation.
2) *S*et up recording and plot voltage. If needed, set limit on voltage axis, e.g.: `pyplot.ylim(-80,30)`
3) Add *preCell1* of same type *HHCell*, add a current stimulation to *preCell1* at 20 ms:
   ```
   preCell1.add_current_stim(20) # add stimulation
   ```
4) Add an excitatory synaptic receptor at the dendrite of *postCell,* with reversal potential of 0 mV and delay of 2 ms*:*
   ```
   postSyn1 = h.ExpSyn(postCell.dend(0.5)) # add exc synapse
   postSyn1.e = 0
   postSyn1.tau = 2
   ```
5) Add a connection between the *preCell1* soma and the *postCell* dendrite with weight of 0.002 and delay of 1 ms.
   ```
   pre1Con = h.NetCon(preCell1.soma(0.5)._ref_v, postSyn1, sec=preCell1.soma)
   pre1Con.weight[0] = 0.002
   pre1Con.delay = 1
   ```
6) Run simulation for 60 ms, plot voltage
7) Try different values of ExpSyn and NetCon parameters, to get (1) small EPSP, (2) EPSP that leads to spike, (3) IPSP.

Connections Tutorial



Play with the ExpSyn and Netcon parameters to get:

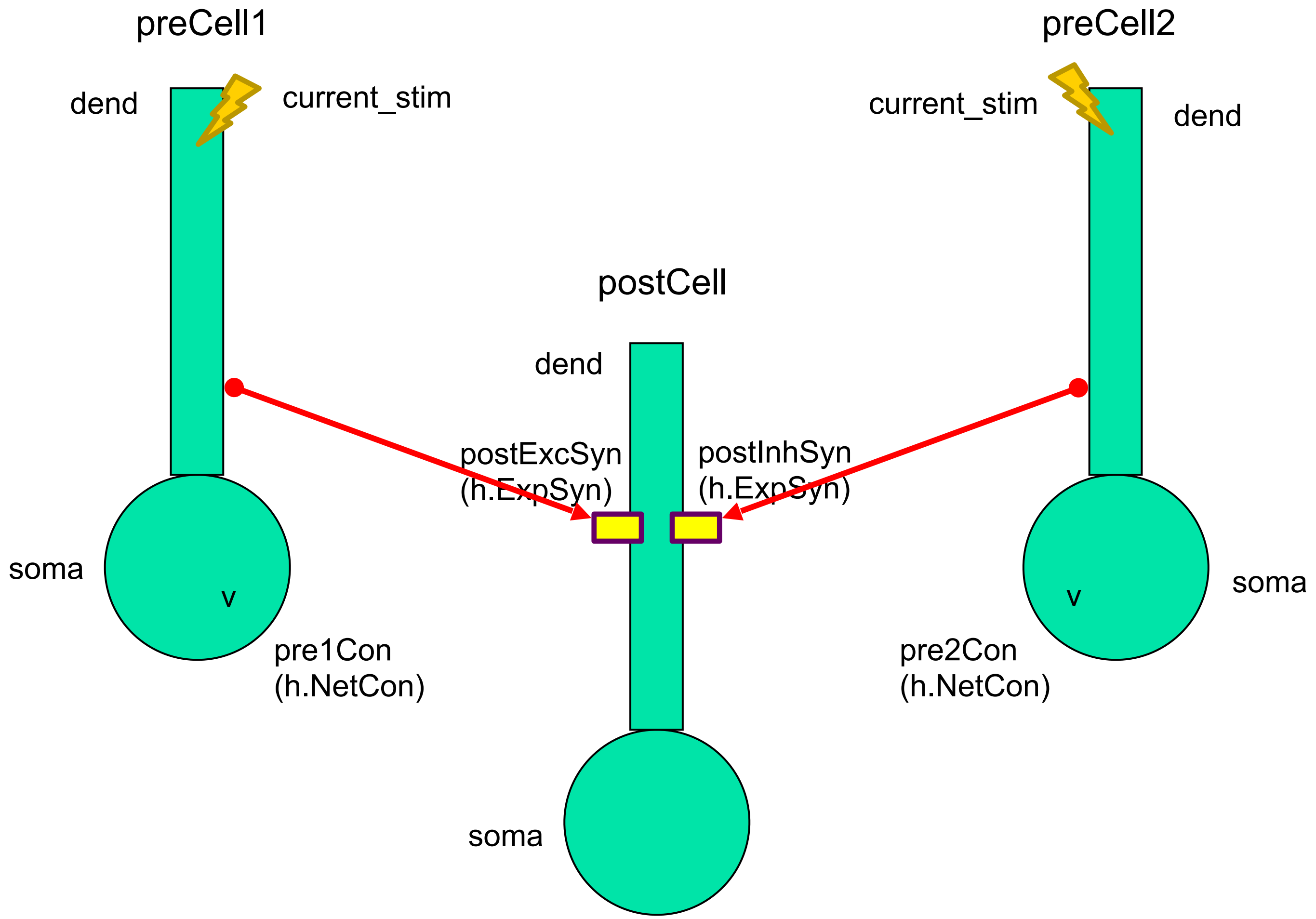
1) **Small EPSP**
2) **EPSP that leads to action potential**
3) **IPSP**

Note: if get errors when rerunning try adding *h.init()* right before *h.run()*

Connections Tutorial

1) Add a *preCell2*
2) Add another excitatory synaptic receptor *postSyn2* at the dendrite of *postCell,* with reversal potential of 0 mV and tau of 2 ms
3) Add a connection between the *preCell2* soma and the *postSyn2* synapse *postCell* dendrite with weight of 0.002 and delay of 1 ms.
4) By commenting out either *preCell1* or *preCell2,* observe that spike in one of them doesn't lead to postsynaptic spike, but spike is generated if both pre-cells are connected (spatial summation)
5) Modify *postSyn2* to be inhibitory (change reversal potential)
6) Try to show spatial cancellation (EPSP+IPSP)

Connections Tutorial

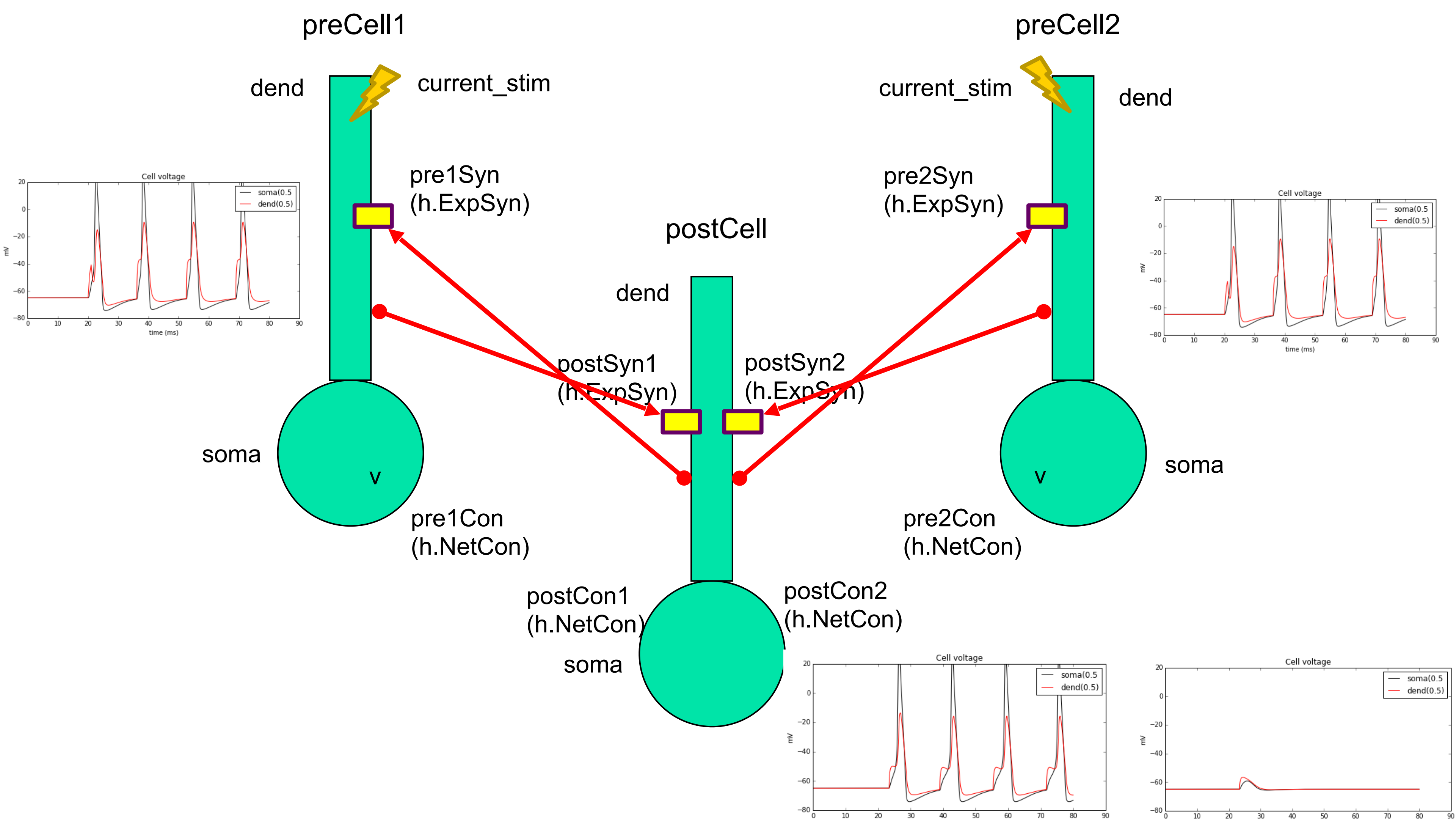


Connections - Assessment

The aim is to connect the postsynaptic cell to the 2 presynaptic cells so that we create a loop and they keep firing forever (see diagram in next slide):

1) Start from the file which contains solution to previous example: cancellation of EPSP (from *preCell1)* and IPSP (from *preCell2).*
2) Adjust the parameters to make both synapses excitatory, and so that the postsynaptic cell fires due to spatial summation (ie. requires input from BOTH presynaptic cells to fire; input from just 1 of them should generate an EPSP but not a spike)
3) Add an excitatory synapse to each of the presynaptic cells, at location 0.5 of the dendrite, with tau of 2 ms.
4) Add a connection from the postsynaptic cell soma to the each of the presynaptic cells (using the synapses created in step 3), each with delay of 10 ms.
5) Run simulation for **80 ms**
6) Adjust the weights as necessary to obtain 4 spikes in all 3 cells as shown in the next slide.
7) Make sure that step 2) still holds true: ie. if you set to 0 the weight of either of the presynpatic cells to the postsynaptic, then the 4 spikes should no longer be there.

Connections Tutorial



with input from 2 presyn cells
(spatial summation)

with input from only 1 presyn
cell (no spatial summation)