Assignment 1: AMPA and NMDA

Neural Computation 2015-2016. Mark van Rossum

16th October 2015

Practical info

Organize your answers according to the questions; don't merge them. Plots should include axis labels and units (either on the plot, or mentioned in the text), see my web page link. Some answers will require units as well.

You will find that some questions are quite open-ended. In order to receive full marks for those you will need to do more than running a simulation and making a plot. Instead, you should substantiate your explanations and claims, for instance by doing additional simulations or mathematical analysis. However, core-dumping (just writing down all you can think of) is discouraged, and incorrect claims can reduce marks. It should not be necessary to consult scientific literature, but if you do use additional literature, cite it. There will be a to-be-determined normalization factor between the number of points scored and the resulting percentage mark.

Copying results is absolutely not allowed and can lead to severe punishment. It's OK to ask for help from your friends. However, this help must not extend to copying code, results, or written text that your friend has written, or that you and your friend have written together. I assess you on the basis of what you are able to do by yourself. It's OK to help a friend. However, this help must not extend to providing your friend with code or written text. If you are found to have done so, a penalty will be assessed against you as well.

Deadline will be announced via email and the website. Hand in a paper copy to ITO (if you are out of town an email with a PDF to me is fine). Late policies are strict and are stated at `www.inf.ed.ac.uk/student-services/teaching-organisation/for-taught-students/coursework-and-projects`. In case of illness etc, contact your personal tutor (CC me if you want).

Model and setup

In this assignment we look excitatory synapses that contain both AMPA and NMDA receptors, and compare the contribution of these different receptor types to the post-synaptic input. First, we examine simplified synapse models.

Create a neuron model with a soma (cylinder with $10\mu m$ length and $10\mu m$ diameter) from which one dendrites sprouts ($500\mu m$ length and $1\mu m$ diameter). Both soma and dendrite are passive; use `g_pas=0.0005` and NEURON's default value for r_i (called Ra in NEURON). Choose nseg such that each segment is $\lambda/20$, where λ is the electrotonic length. This is usually a decent choice the get sufficient numerical accuracy. Now we create a synapse on the dendrite:

```
// create simple synapse with exponential timecourse
objref esyn1 // allocate object
dend esyn1 = new ExpSyn(0.5) //position halfway
```

```
esyn1.tau=5 // time-constant in ms
esyn1.e=0 // reversal potential in mV

objref nil, nc1
nc1= new NetCon(nil, esyn1)
nc1.delay=0
nc1.weight = 0.002 // weight in μS
```

The weight of the synapse is in this case set to 2nS, but you can change it using `nc1.weight = xxx`.

We set the time of synaptic input using the following construction:

```
objref syntimes1
syntimes1 = new Vector(1)
syntimes1.x[0]= 10 // input at 10 ms

objref fih
fih = new FInitializeHandler("loadqueue()")
proc loadqueue() { local ii
for ii=0, syntimes1.size()-1{
nc1.event(syntimes1.x[ii]) }
}
```

Questions

Question 1 (5 points) Show how you calculated the value for nseg. How does nseg depend on the membrane leak?

Question 2 (5 points) First consider a case where the synapse is on the soma. Model the AMPA receptor as an exponential synapse with a 5ms timecourse, and the NMDA receptor as one with a 100ms timecourse. Although our neuron model does not spike, assume that the neuron fires whenever the voltage reaches a certain value. Consider two extreme limits: a very short and a very slow membrane time-constant. How should the conductances of AMPA and NMDA be related in either case to be equally responsible for a spike (i.e. give the same depolarization)? Compare your findings to simulations and discuss any mismatch.

Question 3 (10 points) How would the above arguments change if the synapse were on the dendrite, and the membrane time-constant were manipulated by altering the membrane leak everywhere in the neuron? Again test with simulations.

Next, we model more realistic synaptic responses. Copy `nmdamvr.mod` and `ampamvr.mod` from the course website to your directory. Feel free to browse this code. It is straightforward. Run `nrnivmodl` to compile the synapses (see below). When you start up nrngui, it should list "Additional mechanisms from files ampamvr.mod nmdamvr.mod". For example, for an AMPA synapse, code as follows:

```
objref ampasyn1
dend ampasyn1= new AMPAmvr(0.5) // location is now halfway
objref ncampa1,nil
ncampa1 = new NetCon(nil, ampasyn1)
ampasyn1.gmax = 0.1 // synaptic conductance in μS, in this case 'weight' variable has
no effect.
```

Use an AMPA conductance of 2nS (ampasyn1.gmax) and an NMDA conductance of 200nS (nmdasyn1.gmax).

Note that we now assume that all the synapses considered have both a AMPA and NMDA component, but the AMPA NMDA ratio can be varied.

Question 4 (10 points) Place the synapse at various location on the dendrite. Examine the voltage response in the soma as a function of the distance of the synapse. Explore the various components that determine the time-course and the amplitude of the response and justify your claims.

Question 5 (10 points) Make two synapses (identical, and both with AMPA and NMDA) on the dendrite, one on the far tip, one in the middle. Activate them with a variable time delay and examine the somatic response as a function of the delay. Explain your findings.

Question 6 (5 points) For which values of the AMPA and NMDA conductances is the effect(s) of the previous question the biggest. Give reasoning behind your findings.

Hint:

1. For systematic exploration of the role of a synaptic parameter it can be useful to create an array of synaptic inputs, all located at the same spot, but with different parameters and different activation times. This allows you to obtain all data for a plot in one simulation run. You can use constructions such as:
```
ns=20 //synapses 0...ns-1
objref ampasyn[ns], ampanetcon[ns]
for is=0,ns-1{
dend ampasyn[is] = new AMPAmvr(0.5)
ampasyn[is].gmax = gampa*is
....
```

2. Analysis of the simulation is easiest outside of NEURON, using matlab or similar. Voltage/current traces can be saved. For that use constructions like
```
objref data
data = new Vector();
data.record(&soma.v(0.5)) // this for voltage in soma
objref fout
fout= new File()
fout.wopen("out.dat")
data.printf(fout)
fout.close()
quit()
```

3. If nrnivmodl complains that it cannot find /usr/lib/libnrniv.la, copy nrnivmodl to your working directory (cp /usr/bin/nrnivmodl .), and edit the nrnivmodl so that it looks:
```
ARCH=x86_64
if test "x${NRNHOME}" = x ; then
prefix=/usr
exec_prefix=/usr
bindir=${exec_prefix}/bin
==>> libdir=${exec_prefix}/lib64
```
Next, compile using ./nrnivmodl