

Python Tutorial part 3 - Brian introduction

January 21, 2016

1 Python Tutorial part 3 - Brian introduction

<http://briansimulator.org/>

Brian is a simulator for spiking neural networks available on almost all platforms. The motivation for this project is that a simulator should not only save the time of processors, but also the time of scientists.

Brian is easy to learn and use, highly flexible and easily extensible. The Brian package itself and simulations using it are all written in the Python programming language, which is an easy, concise and highly developed language with many advanced features and development tools, excellent documentation and a large community of users providing support and extension packages.

The following code defines a randomly connected network of integrate and fire neurons with exponential inhibitory and excitatory currents, runs the simulation and makes the raster plot on the right.

In [1]: `%matplotlib inline`

```
from pylab import *
from brian2 import *                               # import brian 2
eqs = '''
dv/dt = (ge+gi-(v+49*mV))/(20*ms) : volt         # membrane potential equation, unit after ':'
dge/dt = -ge/(5*ms) : volt                        # exponential excitatory synapse, unit after ':'
dgi/dt = -gi/(10*ms) : volt                       # exponential inhibitory synapse, unit after ':'
'''
P = NeuronGroup(4000, eqs, threshold='v>-50*mV', reset='v=-60*mV') # initiates a group of 4000
P.v = -60*mV                                       # all neurons are initialized at -60 mV
Pe = P[:3200]                                       # subdivide into 3200 excitatory neurons ...
Pi = P[3200:]                                       # ... and 800 inhibitory neurons
Ce = Synapses(Pe, P, pre='ge+=1.62*mV')            # initiate exc->entire population synapses, a pre-sy
Ce.connect(True, p=0.02)                            # connect the exc connections with a probability of
Ci = Synapses(Pi, P, pre='gi-=9*mV')              # initiate inh->entire population synapses, a pre-sy
Ci.connect(True, p=0.02)                            # connects the inh connections with a probability of
M = SpikeMonitor(P)                                 # records the spikes of the NeuronGroup P
run(1*second)                                       # runs the network for simulating 1 sec
plot(M.t/ms, M.i, '.')                             # plot the spike times in a raster
show()                                              # show the raster plot
```

