

پروژه ی دوم علوم اعصاب محاسباتی :

تمرین اول :

در این تمرین یک جمعیت نرونی شامل 1000 نورون که به دو زیر گروه 800 نورون تحریکی و 200 نورون مهارى تقسیم شده است ، ساخته شده این نورون ها با معادله ی نورون های leaky integrate and fire و با چند حالت پارامتر های اولیه متفاوت شبیه سازی شده اند و به آنها به دو روش مختلف random noise و input random اعمال شده است ، در نهایت خروجی raster plot و نمودار شدت جریان نسبت به زمان یا اختلاف پتانسیل نسبت به زمان نشان دهنده ی این جمعیت خواهد بود .

در تمام آزمایشات از پکیج brian2 استفاده شده است و بعضی قسمتهای کدها یا تعیین پارامتر ها از سایت brian2 برداشته شده است .

آزمایش اول :

شبیه سازی با معادله ی LIF و random noise و اتصال کامل :

```
eqs = '''
dv/dt = -(v-Vr)/taum + xi/sqrt(ms)*mV : volt (unless refractory)
'''
```

یک جمعیت نرونی شامل 1000 نورون با پارامتر های مشخص شده میسازیم و آنرا به 2 زیر گروه مهارى و تحریکی تقسیم میکنیم:

```
taum = 20*ms
Vt = -50*mV
Vr = -60*mV
j0=1000
N=1000

eqs = '''
dv/dt = -(v-Vr)/taum + xi/sqrt(ms)*mV : volt (unless refractory)
'''
#xi =random input noise

P = NeuronGroup(N, eqs, threshold='v>Vt', reset='v = Vr',
                refractory=5*ms, method='euler')
group_exc = P[:800]
group_inh = P[800:]
```

بین تمام نورون ها با وزن w سیناپس داریم (این وزن ها باید به نوعی تعریف شوند که تمام اتصالات قدرت برابر داشته باشند و راهی مناسب برای ان به صورت $w_{ij}=j_0/N$ است) و تمام آنها به یکدیگر متصلند ، نورون های مهارى با وزن منفی به یکدیگر و به نورونهای تحریکی ، نورون های تحریکی با وزن مثبت به یکدیگر و به نورون های مهارى متصلند :

```
# connections
c=Synapses(group_exc,group_exc,'w:1')#all ex ,all in are conected to them selves and eachother
c.connect()
c.w=j0/N
c=Synapses(group_exc,group_inh,'w:1')#all ex ,all in are conected to them selves and eachother
c.connect()
c.w=j0/N
c=Synapses(group_inh,group_exc,'w:1')#all ex ,all in are conected to them selves and eachother
c.connect()
c.w=-j0/N
c=Synapses(group_inh,group_inh,'w:1')#all ex ,all in are conected to them selves and eachother
c.connect()
c.w=-j0/N
```

حال اسپایک های هر گروه را جداگانه و membrane potential مکل گروه را مانیتور میکنیم :

```
#Monitor spikes
s_mon = SpikeMonitor(group_exc)
S_mon=SpikeMonitor(group_inh)

#Monitor membrane potential
v_mon = StateMonitor(P, variables='v', record = [0])
```

سپس شبکه را برای زمان مشخص شده run میکنیم و خروجی ها را visualize میکنیم :

```
run(.5 * second)

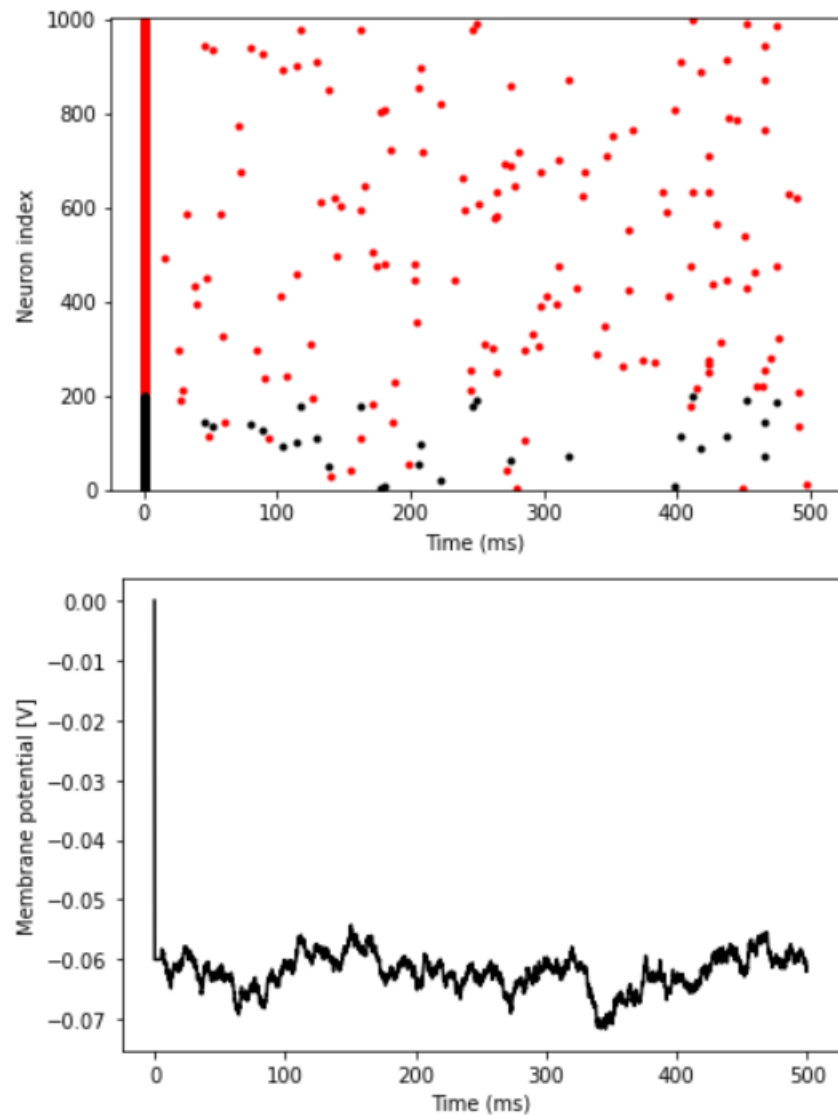
plot(s_mon.t/ms, s_mon.i, '.r')
plot(S_mon.t/ms, S_mon.i, '.k')
xlabel('Time (ms)')
ylabel('Neuron index')
tight_layout()
ylim([-1, len(P)+1])
show()

plot(v_mon.t/ms, v_mon.v[0], 'k')
xlabel('Time (ms)')
ylabel('Membrane potential [V]')
tight_layout()

show()
```

خروجی نمودار هایی به شکل زیر خواهد بود که به ما نشان میدهند کدام نورون ها در چه زمانی اسپایک میزنند و اختلاف پتانسیل چگونه در حال تغییر است مثلا در لحظه ی ۱۰ اختلاف پتانسیل بیشترین مقدار خود را دارد و تمامی نورون ها در حال اسپایک زدن هستند .

چون تمامی نورون ها به یکدیگر با وزن های تعریف شده متصلند باهم برهمکنش دارند و این باعث میشود در لحظات بعدی میزان پتانسیل و اسپایک ها متفاوت شود :

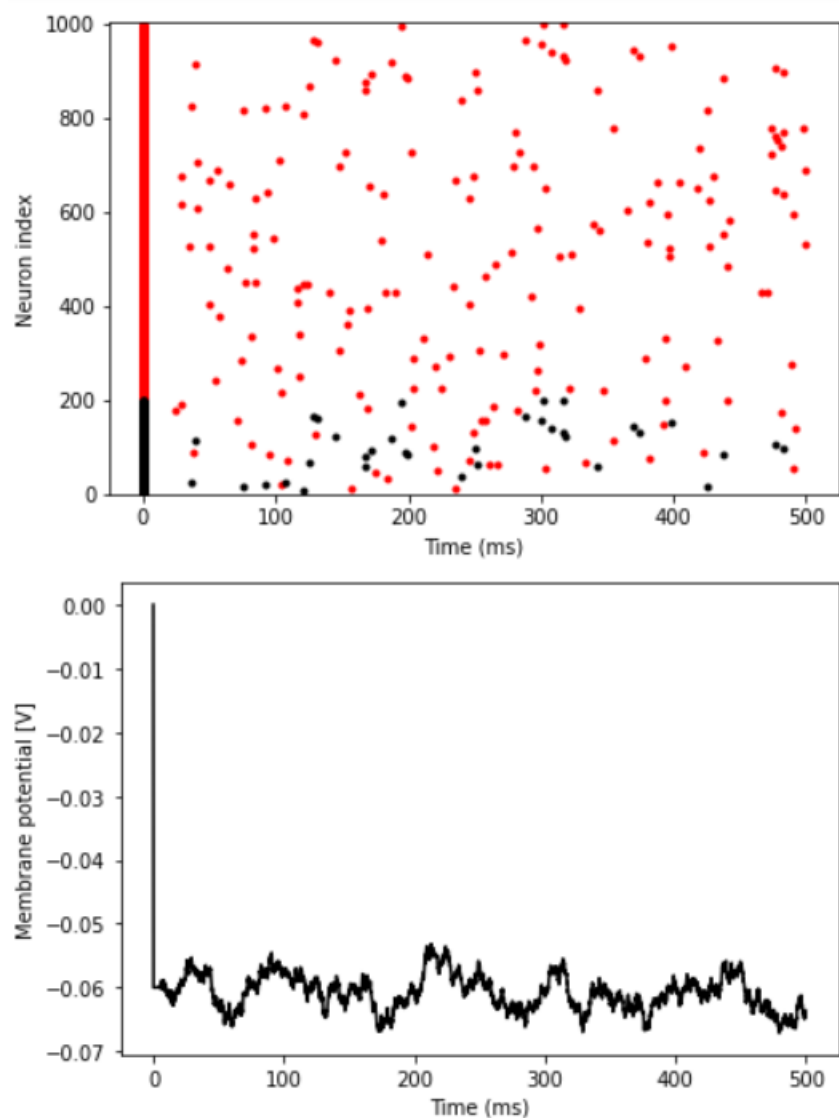


ازمایش 2:

این آزمایش دقیقاً مانند حالت قبلی است با این تفاوت که نورون ها با احتمال p به یکدیگر متصل شده اند ($w_{ij}=j_0/pN$):

```
# connections
c=Synapses(group_exc,group_exc,'w:1')#
c.connect("i<800",p=0.02)
c.w=j0/N
c=Synapses(group_exc,group_inh,'w:1')#
c.connect("i<800",p=0.02)
c.w=j0/N
c=Synapses(group_inh,group_exc,'w:1')#
c.connect("i>=800",p=0.02)
c.w=-j0/N
c=Synapses(group_inh,group_inh,'w:1')#
c.connect("i>=800",p=0.02)
c.w=-j0/N
```

خروجی به صورت زیر خواهد شد:



ازمایش 3:

در این آزمایش به جای استفاده از noise رندوم از جریان I رندوم استفاده کردیم و برای اینکار از ارایه ها استفاده کردم همچنین پارامتر ها را نیز تغییر دادم به طوری که با شدت جریان همخوانی داشته باشد:

```

start_scope()
N=1000
A = 2.5
f = 10*Hz
tau = 5*ms
# Let's create an array
num_samples = int(200*ms/defaultclock.dt)
I_arr = zeros(num_samples)
for _ in range(100):
    a = randint(num_samples)
    I_arr[a:a+100] = rand()
I_recorded = TimedArray(A*I_arr, dt=defaultclock.dt)

eqs = '''
dv/dt = (I-v)/tau : 1
I = I_recorded(t) : 1
'''

```

سپس مانند آزمایش 1 یک جمعیت نورونی با دو زیر گروه ایجاد شده و connection ها نیز مانند حالت اول خواهند بود و برای مانیتور کردن جریان از statemonitor استفاده شده است :

```

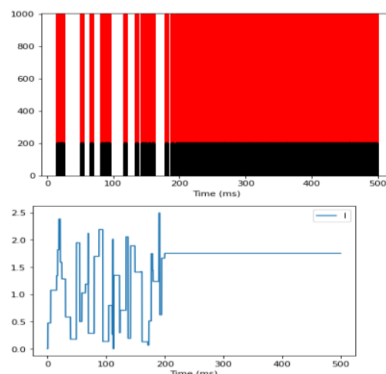
#monitor spikes
s_mon = SpikeMonitor(group_exc)
S_mon=SpikeMonitor(group_inh)

M = StateMonitor(G, variables=True, record=True)

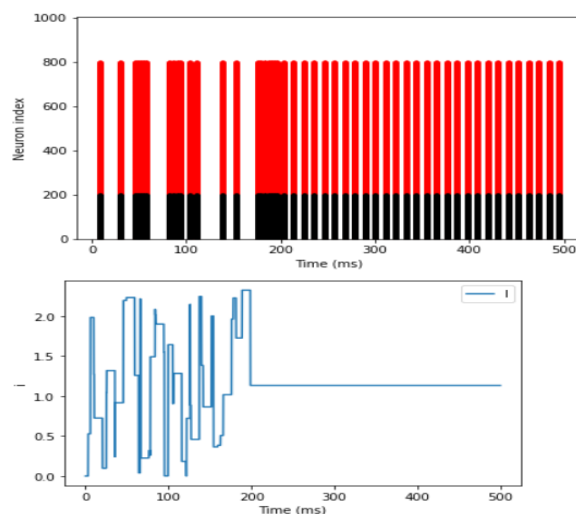
run(.5 * second)
|
plot(s_mon.t/ms, s_mon.i, '.r')
plot(S_mon.t/ms, S_mon.i, '.k')
xlabel('Time (ms)')
ylabel('Neuron index')
tight_layout()
ylim([-1, len(P)+1])
show()
plot(M.t/ms, M.I[0], label='I')
xlabel('Time (ms)')
ylabel('i')
legend(loc='best');

```

و خروجی به شکل زیر خواهد شد(احتمالا به دلیل اثر جریان ورودی اسپایک های زیادی داشته ام و اگر سایر پارامتر ها تغییر کنند یا وزن نورون های مهارى بیشتر شود این اتفاق رخ نخواهد داد) :



همین آزمایش برای حالتی که وزن ها با احتمال p یکدیگر متصلند نیز انجام شده و خروجی به شکل زیر است :



آزمایش 4:

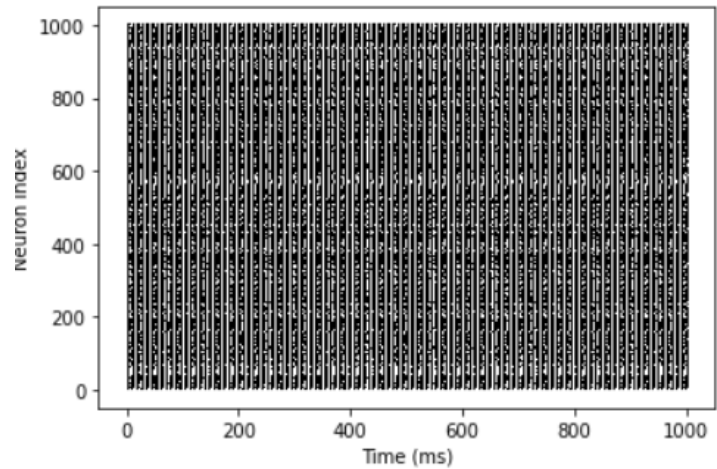
در این آزمایش معادله را به گونه ای تغییر دادیم که پتانسیل به صورت رندوم تغییر کند همچنین تنها بین اعضایی هر گروه سیناپس تعریف کردیم و هنگامی که یک نورون $presynaptic$ اسپایک بزند $v+=w$ خواهد شد ، مقدار تاخیر را نیز اثر داده ایم و پارامتر های اولیه نیز اندکی تغییر داده شده اند :

```
start_scope()
N=1000
taum = 20*ms
Vt = -50*mV
Vr = -60*mV
El = -49*mV
j0=1000

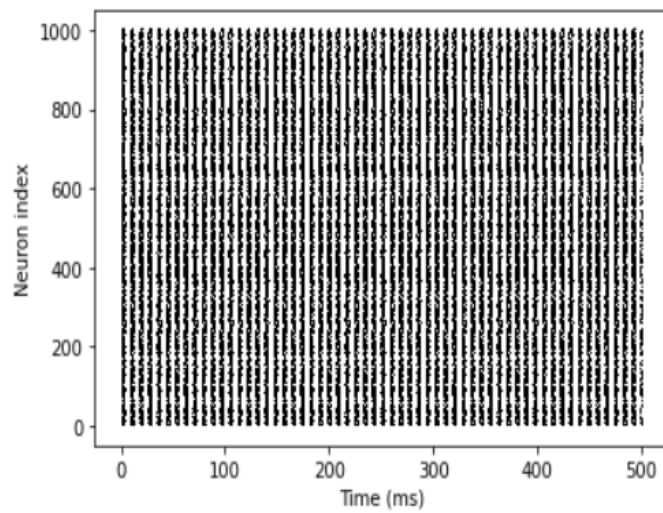
eqs='''
dv/dt = (v0 - v) / taum : volt (unless refractory)
v0 : volt
'''
P = NeuronGroup(N, eqs, threshold='v>Vt', reset='v = Vr', refractory=5*ms, method='exact')
group_exc = group[:800]
group_inh = group[800:]
P.v = 'Vr + Rand()' * (Vt - Vr)'

#connectivity
C1 = Synapses(group_exc, group_exc, 'w:1', on_pre='v += w', delay=1*ms)
C1.connect()
C2 = Synapses(group_inh, group_inh, 'w:1', on_pre='v += w', delay=1*ms)
C2.connect()
C1.connect()
C2.w = f10/N # excitatory synaptic weight
```

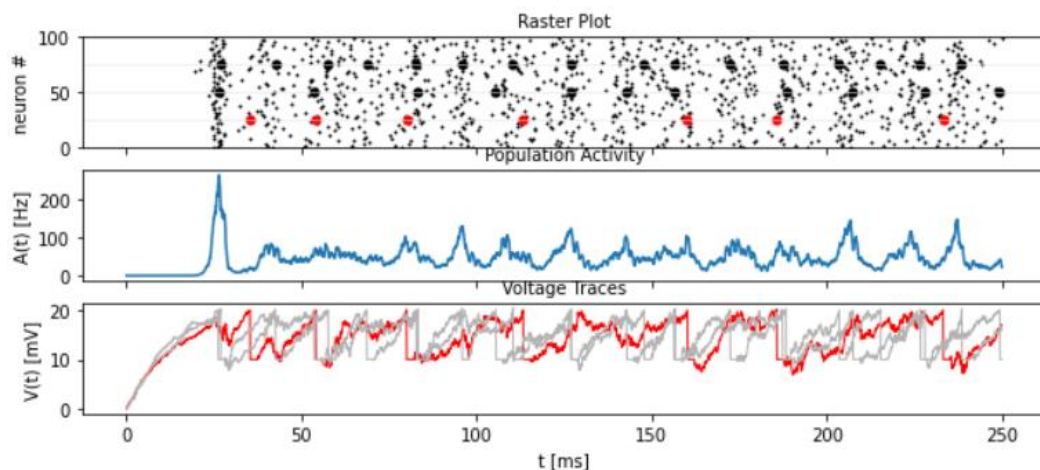
خروجی به صورت زیر خواهد شد:



همین آزمایش برای حالتی که نوروں ها با احتمال p به یکدیگر متصلند خروجی زیر را خواهد داشت:



نمودار زیر خروجی کد آماده ایست که از کتابخانه ی `neurodynex` برداشته شده است برای نشان دادن `rasterplot`، فعالیت جمعیت و میزان پتانسیل ان :



150 نورون به صورت رندوم انتخاب شده و 3 قطار اسپایکی مشاهده میشود .

تمرین دوم :

ازمایش 1:

در این تمرین باید 2 جمعیت نورون تحریکی و 1 جمعیت نورون مهاري ساخته و پدیده ی perceptual decision making را در آنها بررسی کنیم ، این سه جمعیت با یکدیگر برهمکنش دارند شبکه ای که بررسی میکنم شبکه ای ایست که توسط Wong & Wang در سال 2006 بعنوان مدلی از تصمیم گیری ارایه شده است ، پارامتر های استفاده شده از پارامتر های نتورک اصلی متفاوتند ، ابتدا ساختار شبکه و روش دستیابی به subpopulation ها را بررسی میکنیم ، درون جمعیت تحریکی وزن های $w^+ > w^0, w^-$ ، تمامی نورون ها یک ورودی poisson را دریافت میکنند

```
neurodynex.competing_populations.decision_making.sim_decision_making_network(N_Excit=384,
N_Inhib=96, weight_scaling_factor=5.33, t_stimulus_start=100. * msecond, t_stimulus_duration=9.999
* second, coherence_level=0.0, stimulus_update_interval=30. * msecond,
mu0_mean_stimulus_Hz=160.0, stimulus_std_Hz=20.0, N_extern=1000, firing_rate_extern=9.8 * hertz,
w_pos=1.9, f_Subpop_size=0.25, max_sim_time=1. * second, stop_condition_rate=None,
monitored_subset_size=512)
```

پارامترها:

N_Excit (int) – total number of neurons in the excitatory population

N_Inhib (int) – nr of neurons in the inhibitory populations

weight_scaling_factor – When increasing the number of neurons by 2, the weights should be scaled down by 1/2

t_stimulus_start (Quantity) – time when the stimulation starts

t_stimulus_duration (Quantity) – duration of the stimulation

coherence_level (int) – coherence of the stimulus. Difference in mean between the PoissonGroups “left” stimulus and “right” stimulus

stimulus_update_interval (Quantity) – the mean of the stimulating PoissonGroups is re-sampled at this interval

mu0_mean_stimulus_Hz (float) – maximum mean firing rate of the stimulus if $c=+1$ or $c=-1$. Each neuron in the populations “Left” and “Right” receives an independent poisson input.

stimulus_std_Hz (float) – std deviation of the stimulating PoissonGroups.

N_extrn (int) – nr of neurons in the stimulus independent poisson background population

firing_rate_extrn (int) – firing rate of the stimulus independent poisson background population

w_pos (float) – Scaling (strengthening) of the recurrent weights within the subpopulations “Left” and “Right”

f_Subpop_size (float) – fraction of the neurons in the subpopulations “Left” and “Right”. #left = #right = $\text{int}(f_Subpop_size * N_Extrn)$.

max_sim_time (Quantity) – simulated time.

stop_condition_rate (Quantity) – An optional stopping criteria: If not None, the simulation stops if the firing rate of either subpopulation “Left” or “Right” is above stop_condition_rate.

monitored_subset_size (int) – max nr of neurons for which a state monitor is registered.

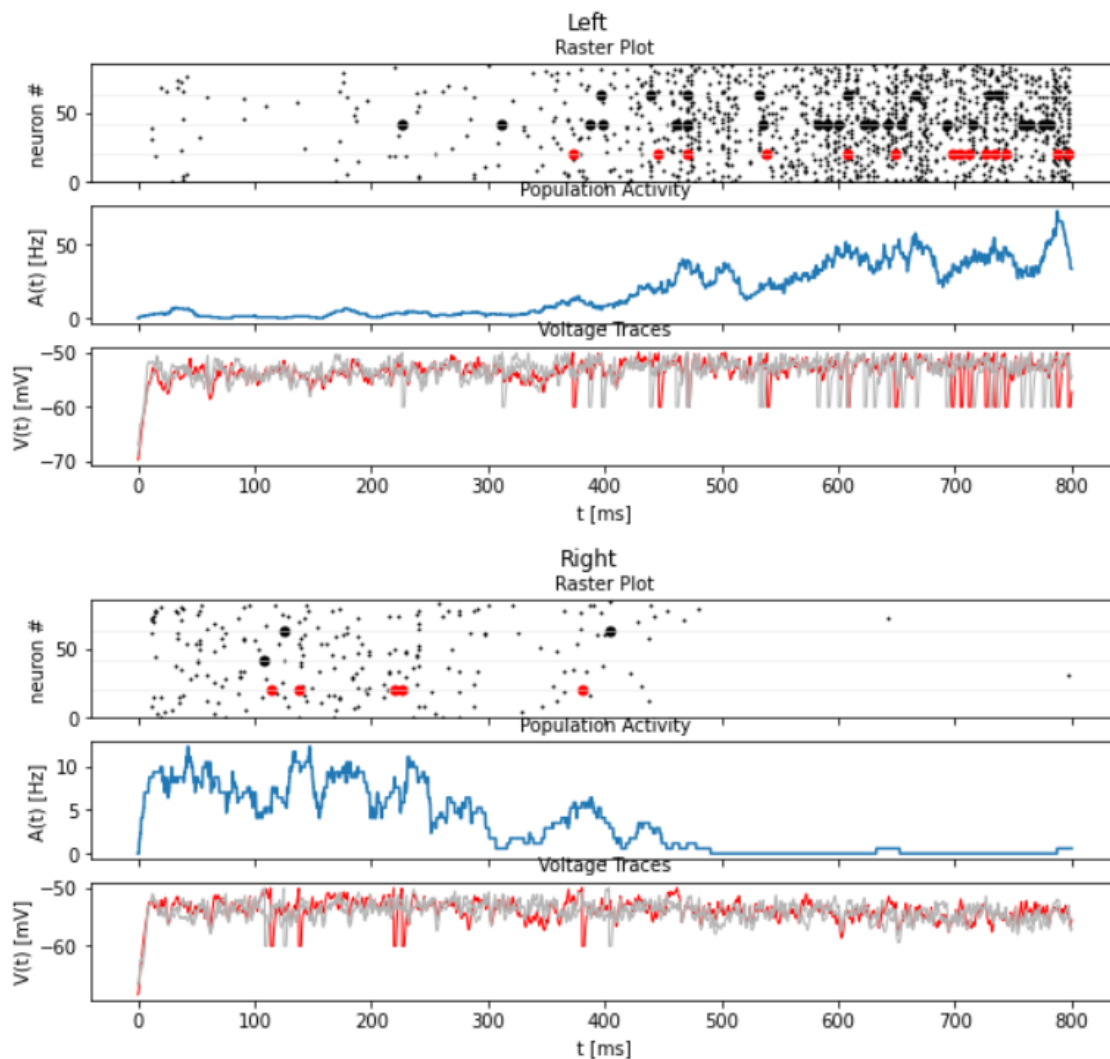
محرك ورودی توسط دو poissonGroup اجرا میشود ، هر 30ms ، ν_{left} و ν_{right} هر کدام از poissonGroup ها از توزیع نرمال زیر بدست میآیند :

$$\begin{aligned}\nu_{left} &\sim \mathcal{N}(\mu_{left}, \sigma^2) \\ \nu_{right} &\sim \mathcal{N}(\mu_{right}, \sigma^2) \\ \mu_{left} &= \mu_0 * (0.5 + 0.5c) \\ \mu_{right} &= \mu_0 * (0.5 - 0.5c) \\ c &\in [-1, +1]\end{aligned}$$

The coherence level c , the maximum mean μ_0 and the standard deviation σ are parameters of `sim_decision_making_network()`.

خروجی:

“rate_monitor_A”, “spike_monitor_A”, “voltage_monitor_A”, “idx_monitored_neurons_A”,
 “rate_monitor_B”,



ازمایش 2:

در این آزمایش تلاش کردم بدون استفاده از کتابخانه آزمایش قبل ، 3 جمعیت نورونی بسازم که با یکدیگر برهمکنش دارند ، دو جمعیت تحریکی و یک جمعیت مهارى است و جمعیت های تحریکی از دو poisson group و ورودی دریافت میکنند که ورودی ها با یکدیگر برابر نیستند تا یکی از آنها بتواند جمعیت برنده بشود ، برای آنکه شبکه در تعادل باشد نورون های مهارى از threshold کمتری برخوردارند(همچنین اگر وزن آنها 2 برابر نورون های تحریکی بود هم این اتفاق میافتاد) ، تمامی نورون ها با وزن های مشخص به یکدیگر متصل شده اند

```

eqs = '''
dv/dt = -(v-Vr)/taum : volt (unless refractory)
'''

P = NeuronGroup(N1, eqs, threshold='v>Vt', reset='v = Vr',
               refractory=5*ms, method='euler')

G=NeuronGroup(N2, eqs, threshold='v>Vt', reset='v = Vr',
               refractory=5*ms, method='euler')

inh=NeuronGroup(N3, eqs, threshold='v>Vt', reset='v = Vr',
                refractory=5*ms, method='euler') # since the network must be balanced the threshold of the inhibitory pop is 2 times smaller than exc pops
poissonG1=PoissonGroup(500,np.arange(500)*Hz + 10*Hz)
poissonG2=PoissonGroup(500,np.arange(500)*Hz + 0.1*Hz)

```

```

S = Synapses(poissonG1, G)
S.connect()
S = Synapses(poissonG2, P)
S.connect()
# connections
c=Synapses(P,P,'w:1')#all ex ar
c.connect()
c.w=j0/N1

c=Synapses(G,G,'w:1')#all ex ar
c.connect()
c.w=j0/N2

c=Synapses(inh,inh,'w:1')#all i
c.connect()
c.w=j0/N3

c=Synapses(P,G,'w:1')# ex pops
c.connect()
c.w=j0/N1

c=Synapses(P,inh , 'w:1')#ex pop
c.connect()
c.w=j0/N3

c=Synapses(G,inh , 'w:1')#ex pop
c.connect()
c.w=j0/N3

```

```

c=Synapses(inh,P , 'w:1')#ex p
c.connect()
c.w=-j0/N3

c=Synapses(inh,G , 'w:1')#ex p
c.connect()
c.w=-j0/N3

```

زمانی که جریان ها به جمعیت های نورونی تحریکی میرسند هر دوی آنها شروع به افزایش فعالیت میکنند و چون با جمعیت مهاری برهمکنش دارند باعث میشوند آن جمعیت نیز شروع به فعالیت کند و اسپایکهایی با وزن منفی به دو جمعیت تحریکی ارسال کند بدین ترتیب بعد از مدتی یکی از جمعیت های تحریکی بیشتر از دیگری فعالیت میکند و به عنوان جمعیت برنده شناخته خواهد شد برای اندازه گیری این فعالیت ها و برهمکنش ها نیاز است معادلات زیر پیاده شده و جمعیت برنده مشخص گردد:

$$\begin{aligned}
 \tau_E \frac{dh_{E,1}}{dt} &= -h_{E,1} + w_{EE}g_E(h_{E,1}) + w_{EI}g_{inh}(h_{inh}) + RI_1, \\
 \tau_E \frac{dh_{E,2}}{dt} &= -h_{E,2} + w_{EE}g_E(h_{E,2}) + w_{EI}g_{inh}(h_{inh}) + RI_2, \\
 \tau_{inh} \frac{dh_{inh}}{dt} &= -h_{inh} + w_{IE}g_E(h_{E,1}) + w_{IE}g_E(h_{E,2}),
 \end{aligned}$$

اما من نتونسم این معادلات رو پیاده سازی کنم :))