

# گزارش سری دوم تمرین علوم اعصاب محاسباتی

محمد زیاری - 97222047

برای این تمرین از پکیج brain2 استفاده کرده ایم. این تمرین شامل 2 بخش است که قابل ذکر است که برای آن از معادله LIF استفاده شده است.

## بخش اول

در ارتباط با ساختار کلی کد ، اول از همه من 3 بخش جمعیت نورونی مجزا در نظر گرفتم و در هر کدام از آنها یک سری تغییرات اعمال کردم و نمودارها در پایین آمده است. اما در ابتدا معادله مورد نظر را تعریف کرده ایم که همانطور که در بالا ذکر شده LIF است. سپس جمعیت نورونی را تعریف کردیم و inh از exc جدا شده است. پس از آن هم سیناپس ها یا نحوه اتصال و همچنین مانیتورها تعریف شده اند. در آخر نمودار جریانی که به صورت رندوم داده شد ، نمودار رستر پلات و همچنین V-I برای یک نورون inh و یک نورون exc ترسیم شده است.

این مقادیر اولیه ای است که به احتمال اتصال ، reset ، threshold و ... داده ایم :

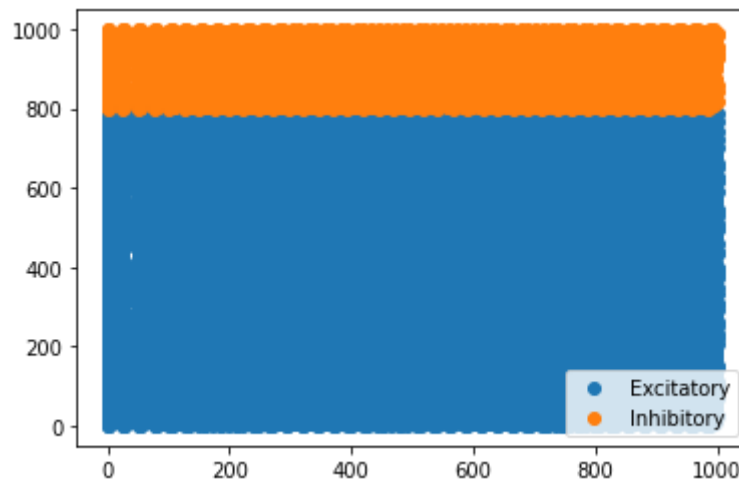
```
Pop = NeuronGroup(1000, eqs, threshold='v>-50', reset='v=-65' , method='linear')

Pop.I = I
Pop.tau = tau*ms
Pop_E = Pop[:800]
Pop_I = Pop[800:]

Syn_E = Synapses(Pop_E, Pop, on_pre='v_post += 0.05')
Syn_I = Synapses(Pop_I, Pop, on_pre='v_post -= 0.1')

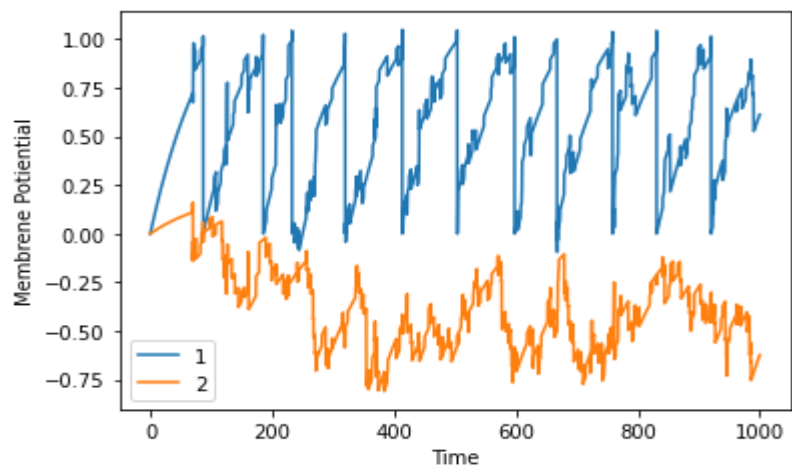
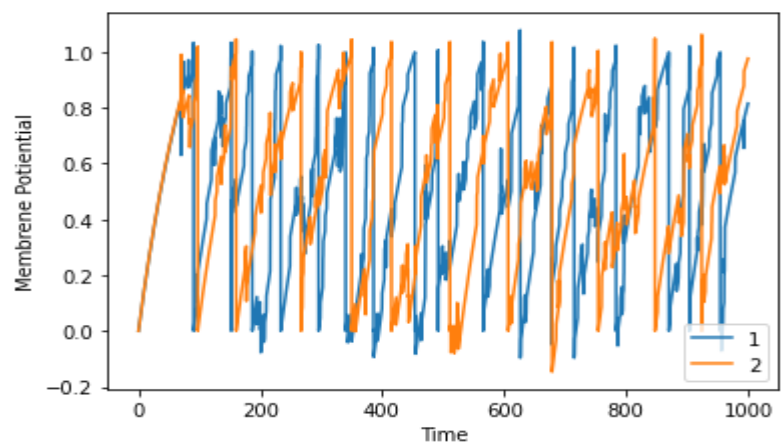
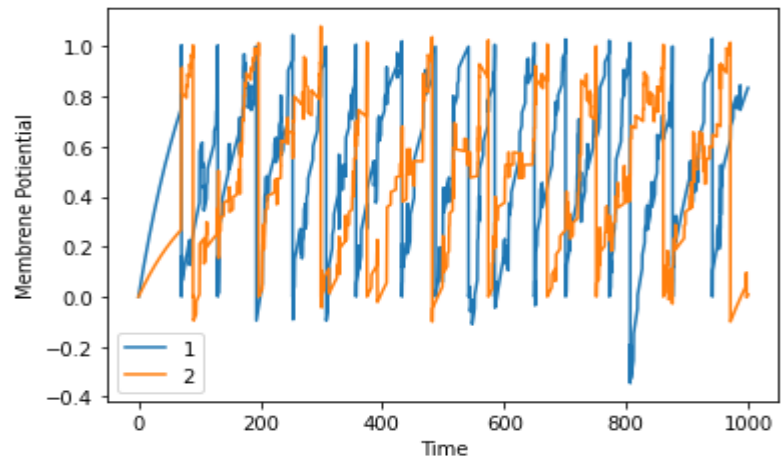
Syn_E.connect(p=0.05)
Syn_I.connect(p=0.1)
```

و تعداد اسپایک ها را اندازه گیری میکنیم :

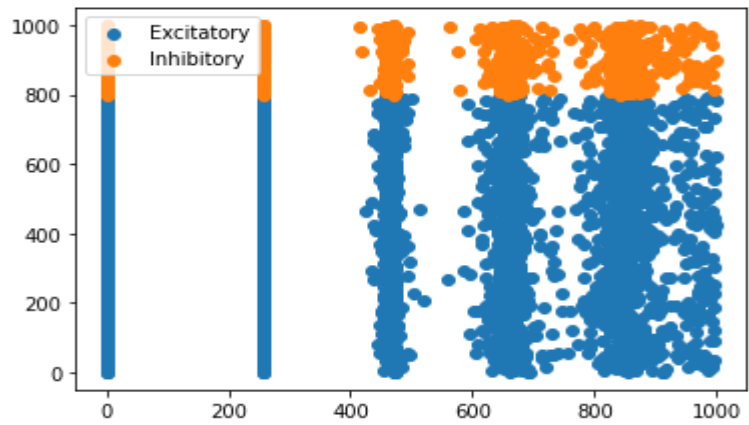
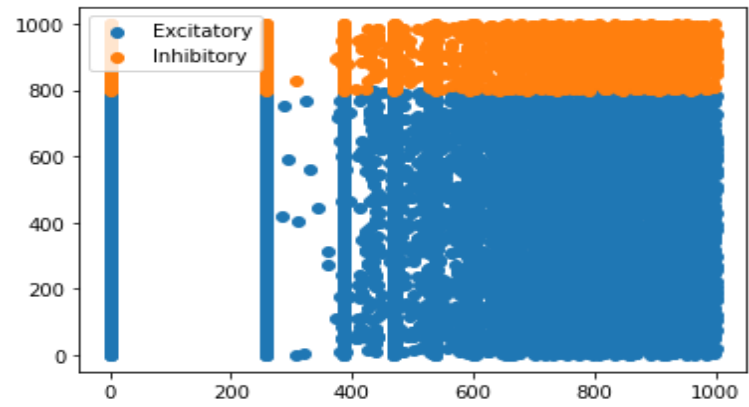


کاملاً مشخص است که اسپایک‌های زیادی در آن دیده می‌شود. حتی برای اینکار مقادیر  $\tau$  و  $\text{method}$  را عوض کردم اما کارساز نبود. برای مثال نمودار با  $\text{method}=\text{exact}$  در کد اصلی موجود می‌باشد.

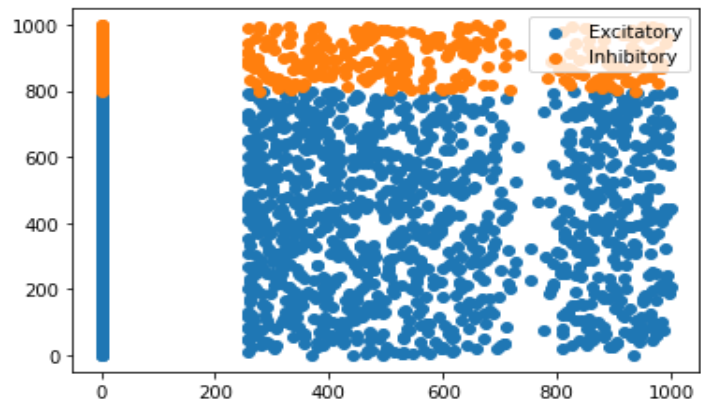
برای اینکار تغییری که می‌شود انجام داد آن است که زمان یا  $\tau$  را بالا ببریم یا آن که  $\text{threshold}$  و  $\text{reset}$  را به مقادیر 1 و 0 تغییر بدهیم که مقادیر  $\text{default}$  این تابع بودند. در  $\text{group2}$  برای نمایش نمودارها و تست از روش دوم و  $\text{group3}$  از روش سوم استفاده کردیم. حالا به سراغ جمعیت دوم می‌رویم که در آن بیشتر با متدها سر و کار داریم. نکته‌ای درباره  $\text{method}$  ها هم آن است که اگر مقداری به آن‌ها داده نشود به صورت  $\text{default}$  هم در نظر گرفته می‌شود که معمولاً از  $\text{exact}$  استفاده می‌شود. نمودارهای  $V-I$  برای 3 حالت مختلف متدها در پایین مشاهده می‌کنیم. (به ترتیب  $\text{linear}$  ،  $\text{euler}$  ،  $\text{exact}$ )

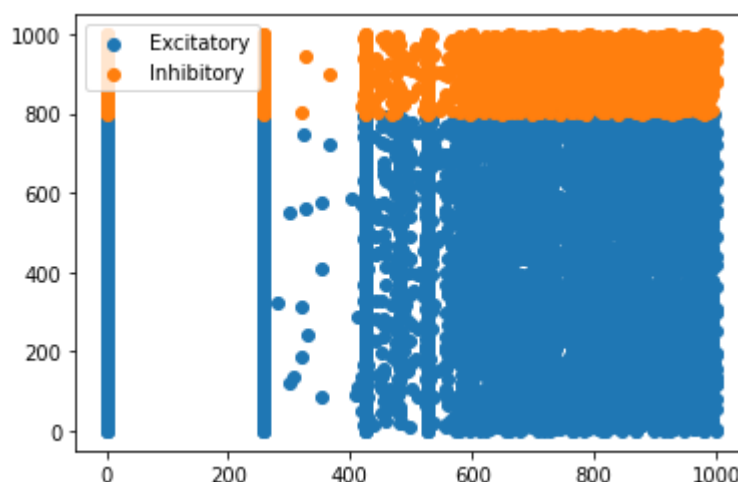


از این پس کاری با متد ها نداریم و از متد پیشفرض استفاده خواهیم کرد.  
 وزن و اتصال ما که توسط سیناپس ها معین میشوند . در ابتدا تاثیر  $v\text{-post}$  سیناپس های  
 اینهیبتوری را بررسی میکنیم. نمودار اول  $v\text{-post}+=0.2$  و دومی  $v\text{-post}=-0.2$  است.



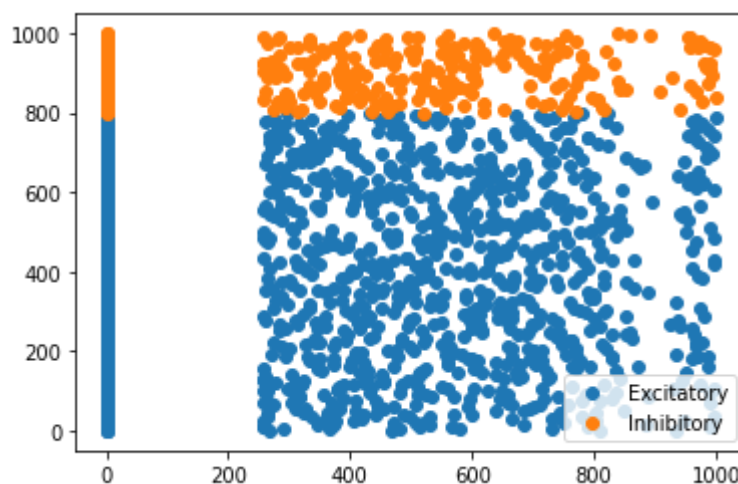
مشخص است میزان اسپایک های نورونها درحالتی که وزن بین نورون های inh افزایشی است به مقدار بیشتری نسبت به حالت کاهششی دارد. همین فرایند را برای نورونهای exc هم آزمایش میکنیم

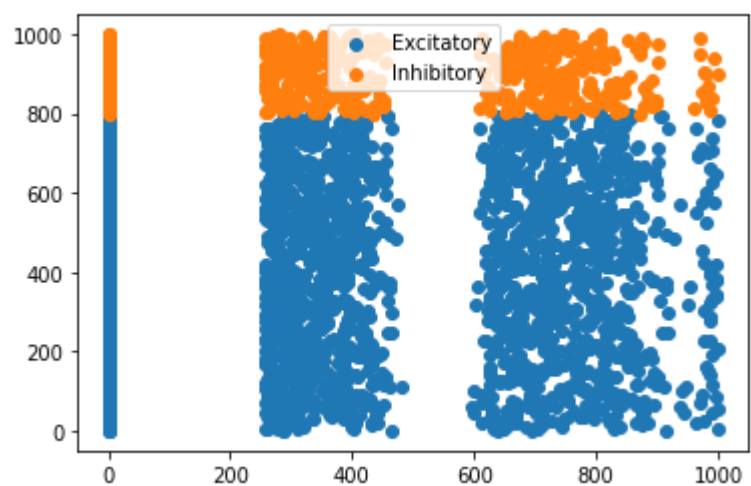




مشخص است اینبار هم در حالتی که وزن افزایشی است تعداد اسپایکها بیشتر است. نکته ای که در چند نمودار مشاهده کردم آن بود که پراکندگی اسپایک ها با تغییر بیشتر وزن نورونهای **exc** تغییر میکند. در آخر نوبت آن است  $p$  یا احتمال برخورد را بررسی کنیم.

اول از **exc** شروع میکنیم. زمانی که حالت وزن افزایشی دارد ، احتمال اتصال بالاتر باشد ، تعداد اسپایک ها بیشتر است اما زمانی که کاهشی است با بیشتر کردن اتصال ها اسپایکها پراکنده و کمتر میشوند. در واقع تصورم این است که دیگر پشت سر هم اسپایک نداریم. 2 نمودار پایین در حالت وزن کاهشی و  $p=0.05$  و  $p=0.3$  می باشد.





در آخر تاثیر  $p$  اینهیبتوری است که تاثیرش تقریباً مشابه به  $exc$  بود و فکر میکنم تاثیر  $p_{exc}$  در اسپایکها واضح تر بود. تغییر  $p_{inh}$  در کد موجود است.

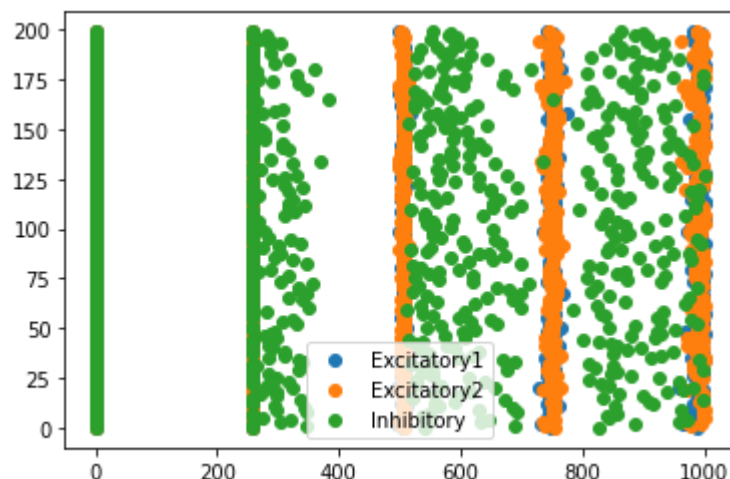
## بخش دوم

مانند بخش قبلی است با این تفاوت که این بار برای هر جمعیت نورونی پارامترهای جداگانه داریم و تعریف میکنیم و 3 گروه با 200 نورون بر این اساس تعریف میشوند و مانند قبل سیناپس ها و مانیتورها تعریف میشوند . حال میخواهیم با مقادیر مختلف برای پارامترها دریابیم چه نتیجه ای خواهیم داشت. در واقع decision چه خواهد بود. اینبار همه تغییرات در همان جمعیت اول لحاظ میکنیم و تنها در اینجا نمودارها را بررسی میکنیم که ببینیم کدام جمعیت **exc** برنده میشود یا در واقع اسپایک بیشتری میزند. این مقادیر اولیه ای است که نسبت دادیم که برای هر دو جمعیت **exc** مشابه است.

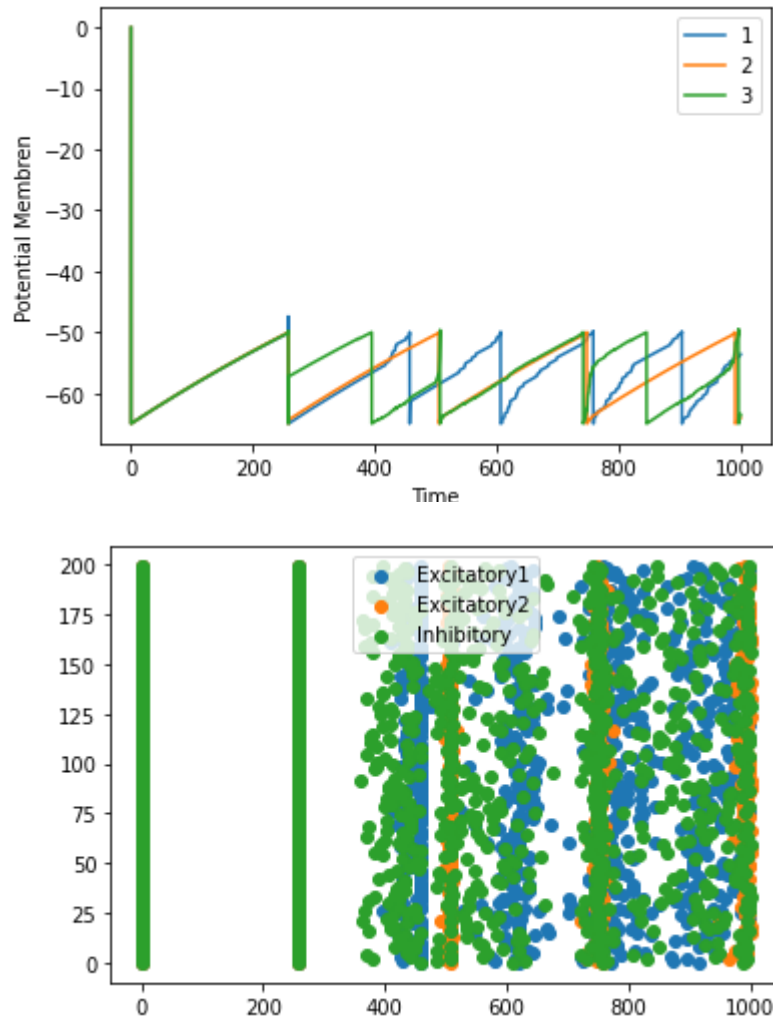
```
Syn_1 = Synapses(Pop_E1, Pop_E1, on_pre='v_post += 0.1')
Syn_2 = Synapses(Pop_E2, Pop_E2, on_pre='v_post += 0.1')
Syn_3 = Synapses(Pop_E1, Pop_I, on_pre='v_post += 0.05')
Syn_4 = Synapses(Pop_E2, Pop_I, on_pre='v_post += 0.05')
Syn_5 = Synapses(Pop_I, Pop_I, on_pre='v_post -= 0.2')

Syn_1.connect(p=0.05)
Syn_2.connect(p=0.05)
Syn_3.connect(p=0.05)
Syn_4.connect(p=0.05)
Syn_5.connect(p=0.1)
```

اما جمعیت نورونی دومی بیشتر اسپایک زده که احتمالاً به علت جریانی است که به صورت رندوم به آنها داده میشود.

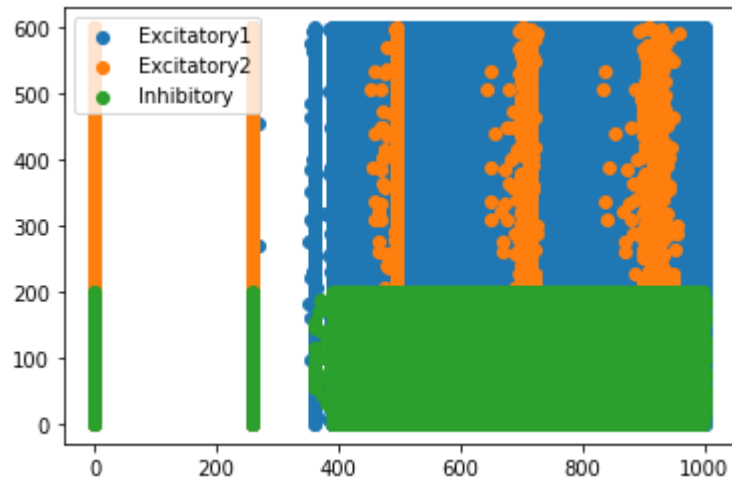


حالا سعی میکنیم با تغییر وزن و اتصالات کاری کنیم **exc1** بیشتر اسپایک بزند. برای اینکار طبق بررسی که کردیم اگر اتصال **exc2** با **inh** با **v-post** بیشتری صورت گیرد با احتمال اتصال بیشتر ، همچنین اتصال و وزن **exc1** با خودش بالاتر برود ، تعداد اسپایک های **exc1** بیشتر خواهد شد. در پایین نمودار پتانسیل 3 نورون از هر جمعیت و همچنین راستر پلات را می بینیم.

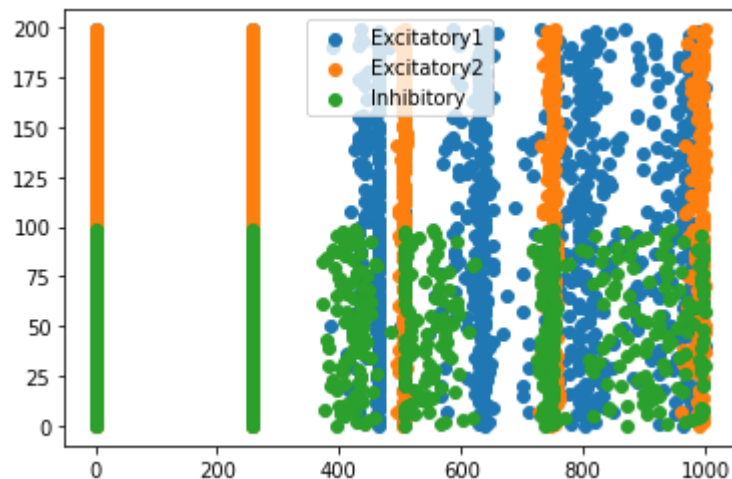


تقریباً با بررسی برای بیشتر کردن اسپایک جمعیت اول مشخص شد افزایش یا کاهش وزن هریک از اتصالات چه رخ میدهد که در بالا گفته شد. اما تنها چیزی که بررسی نشد اتصالات inh به inh بود. چیزی که واضح است آن است که اسپایک های inh خیلی زیاد است. فکر کنم دلایلش برابر بودن تعداد نورون ها در هر 3 جمعیت است. از این رو تعداد نورونها exc را به 600 افزایش میدهیم تا تفاوت را مشاهده کنیم.





به علت افزایش تعداد نورونها و اتصالات زیاد پر از اسپایک شده است برای همه جمعیت ها. تصمیم گرفتیم exc ها را 200 , inh را 100 کنیم.



حالا میزان اسپایک ها کمتر شد. تنها بررسی باقیمانده وزن و اتصالات inh باهم است. در حالتی که **v-post** کاهشی است هرچه میزان اتصالات را افزایش دهیم ، میزان اسپایک ها کمتر میشود و البته در حالت افزایشی بالعکس این قضیه برقرار است. برای مثال نمودار پایین در حالت **v-post=0.4** میباشد و **p=0.3** . در کد **p=0.1** هم موجود است و مشخص است تعداد اسپایکها افزایش یافته است.

