

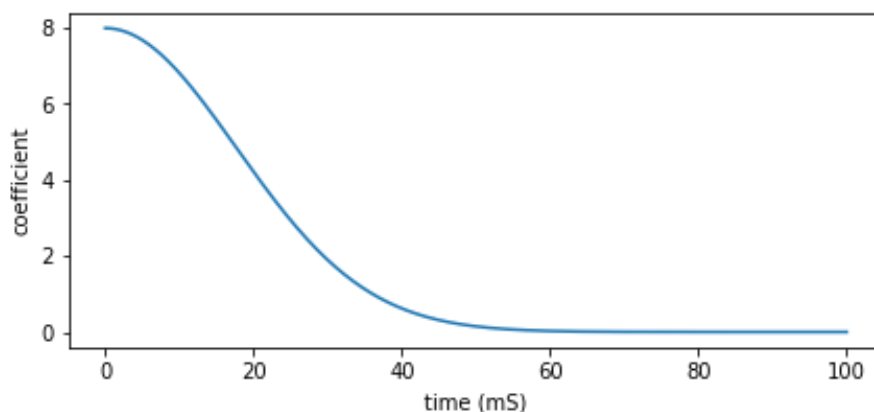
در این سری از تمرین هدف پیاده سازی متد های پیشنهادی برای فرایند های یادگیری در مغز است . در سوال اول پیاده سازی روش *STDP* و در سوال دوم پیاده سازی روش *Reward based STDP* خواسته شده است.

در کنار این فایل که گزارش تمرین است سه فایل دیگر نیز به نام های *PyNeuron.py* و *CNS_ErfanKarami_98222079_3.ipynb* و *neuronpopulation.py* هم آمده اند که فایل اول مدلسازی هایی از تک نورون است که در گزارش سری قبلی تمرین در مورد آن توضیحات کامل داده شده است. فایل دوم فایل نوتبوک تمرین است که خروجی اجرای مدل ها در آن قرار دارد و فایل سوم فایلی است که مدلسازی جمعیت نورونی و شبکه ای از جمعیت های نورونی را نشان می دهد، در این فایل تغییراتی در کلاس ها داده شده به گونه ای که بتوان یادگیری *STDP* را برای جمعیت های نورونی ایجاد کرد و در نهایت شبکه عصبی اسپایکی *SNN* با یادگیری *Reward base STDP* در این فایل پیاده سازی شده است.

برای کدینگ اطلاعات از روش *rate coding* استفاده شده و جریان ورودی به هر نورون پس سیناپسی با استفاده از یک تابع *time course* و به صورت تجمیعی با توجه به فعالیت نورون پیش سیناپسی تعیین میشود.

البته در تمرین سری قبل هم این تابع *time course* مورد استفاده قرار گرفته بود.

time course function for excitatory population

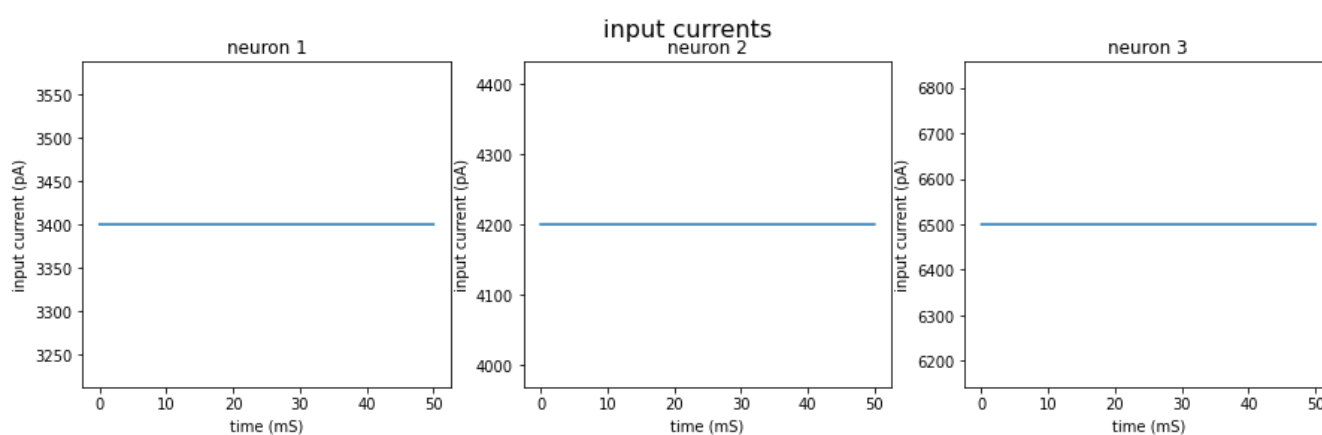


در واقع در این کدینگ فعالیت نورون های پیش سیناپسی در طول زمان تجمیع شده و با ضربی که با توجه به تابع بالا تعیین میشود به نورون پس سیناپسی داده میشود.

۱. در این سوال خواسته شده است که سه نورون را به صورت *fully connected* در نظر بگیریم و با دادن سه دسته مختلف جریان ورودی به این نورون ها با روش *STDP* وزن اتصالات را آپدیت کنیم.

• سه جریان ثابت متفاوت:

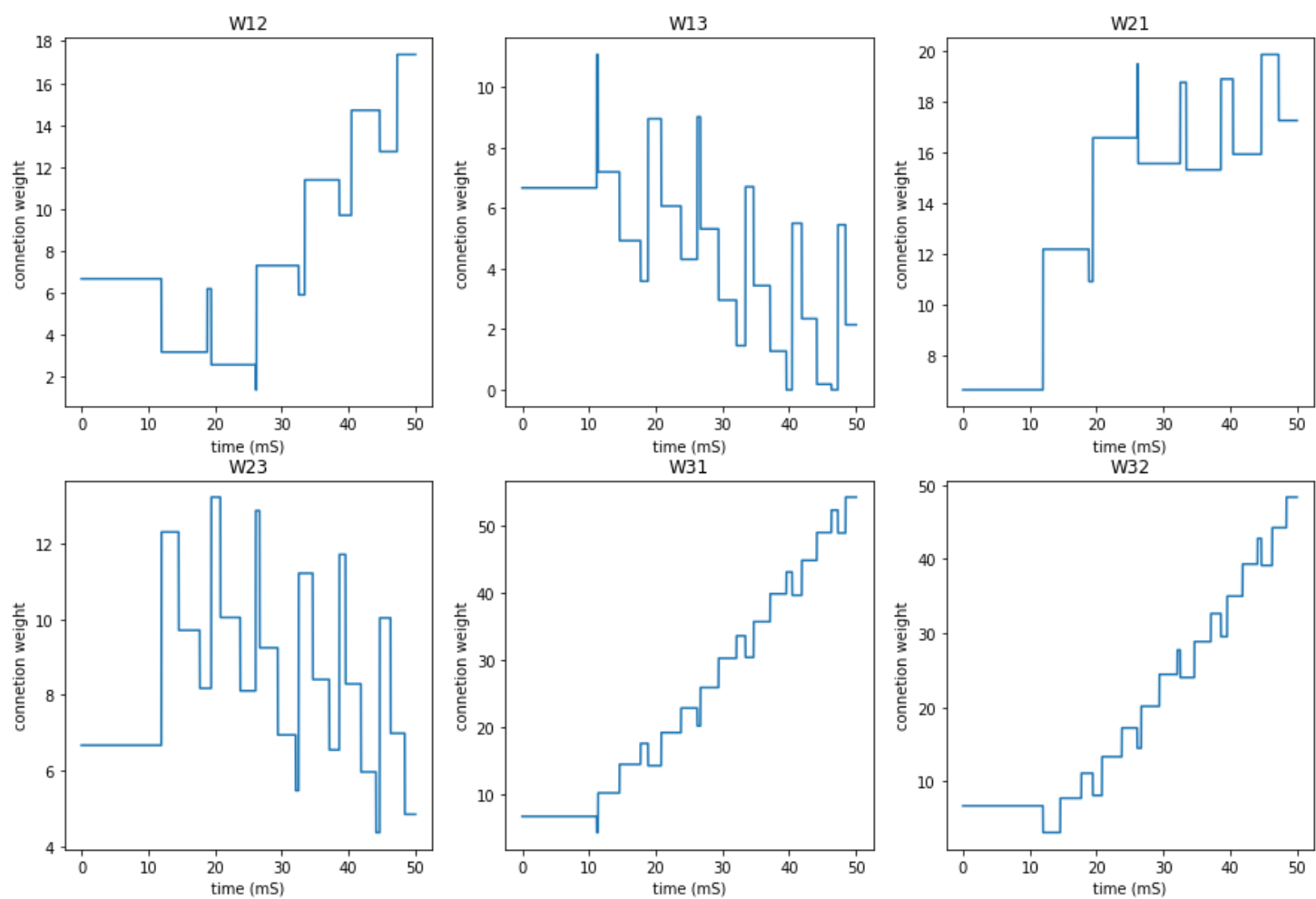
شکل توابع جریان ورودی در زیر آورده شده و هر کدام به یکی از نورون های موجود در جمعیت داده شده اند:



همانطور که مشخص است جریان ها در واقع سه جریان 3400 pA , 4200 pA , 6500 pA بوده اند.

حال شبیه سازی را انجام داده و تغییرات وزن ها به صورت زیر هستند:

Synaptic weight changes according to STDP learning rule



parameters

neurons: 3 * LIF(10+np.random.rand()*3,8+np.random.rand()*4,-79,-50,5,-68)

learning_rule: STDP

time_interval: 50 mS

dt: 0.3125 mS

dt_minus: 6 mS

dt_plus: 10 mS

a_minus: -4

dt_plus: +6

time course thresholds: 0.3

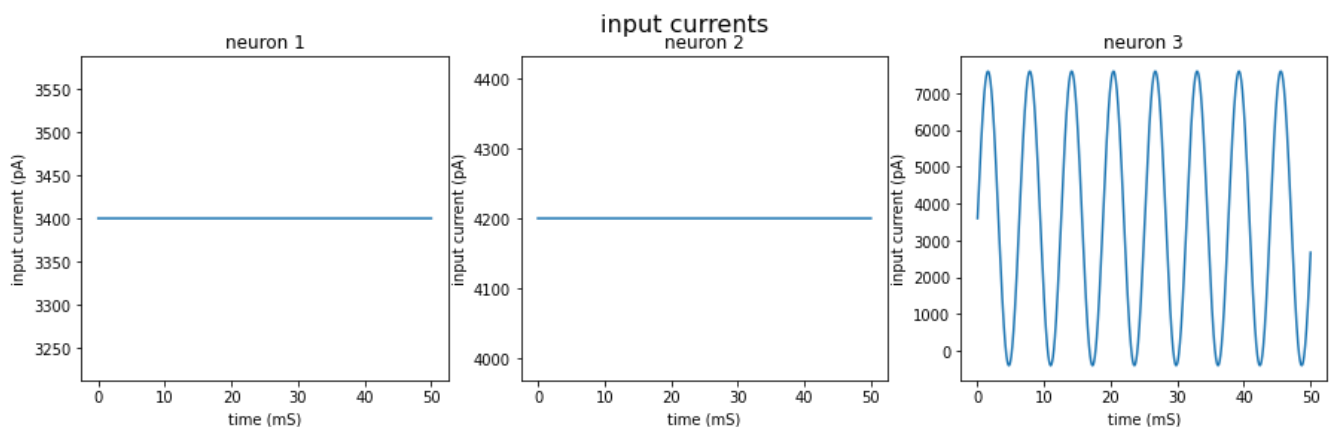
دقت داشته باشید که مطابق قرار داد کتاب w_{ij} به معنای اتصال بین دو نورون i و j است که نورون j پیش سیناپسی و نورون i پس سیناپسی است. با توجه به این نکته و نمودار ها تنها اتصالات از نورون سوم به دوم و سوم به اول در طول یادگیری نسبت به شروع کاهش وزن سیناپسی را تجربه کرده اند و تغییرات وزن سیناپسی برای بقیه اتصالات مثبت بوده.

*** برای پیاده سازی $STDP$ هم از $pair\ based\ STDP$ و از روابط زیر استفاده شده. برای افزایش سرعت اجرا و سادگی هم تنها نزدیک ترین اسپایک ها مدنظر قرار گرفته اند.

$$\begin{aligned}\Delta w_+ &= A_+(w) \cdot \exp(-|\Delta t|/\tau_+) \text{ at } t_{\text{post}} \text{ for } t_{\text{pre}} < t_{\text{post}}, \\ \Delta w_- &= A_-(w) \cdot \exp(-|\Delta t|/\tau_-) \text{ at } t_{\text{pre}} \text{ for } t_{\text{pre}} < t_{\text{post}},\end{aligned}\quad (19.10)$$

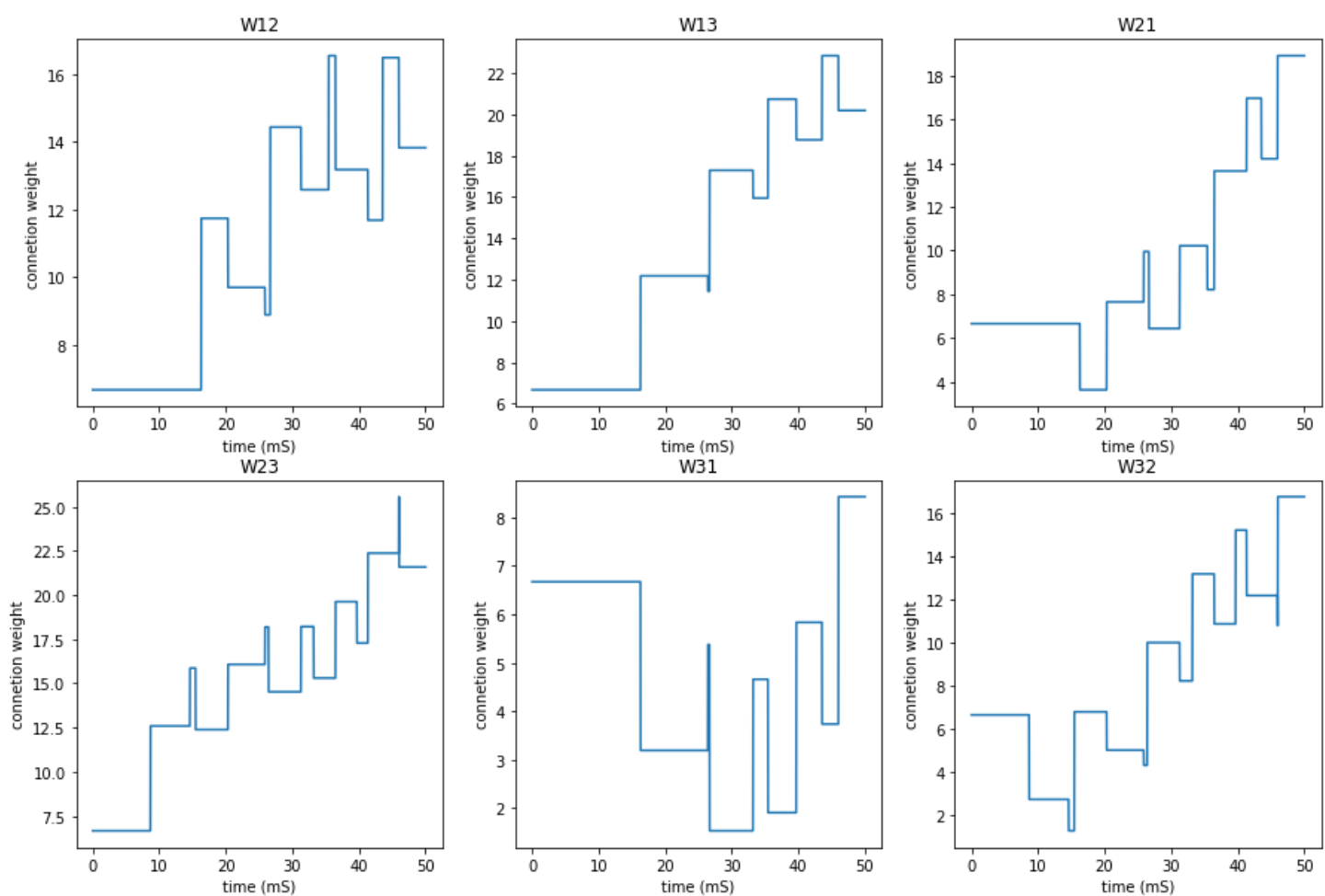
• دو جریان ثابت متفاوت و یک جریان سینوسی:

شکل توابع جریان ورودی در زیر آورده شده و هر کدام به یکی از نورون های موجود در جمعیت داده شده اند:



جریان های ورودی به صورت دو جریان ثابت 3400 pA , 4200 pA و یک جریان سینوسی با ضابطه $4000 * (\sin(\text{time}) + 0.9)$ تعریف شده اند و با این جریان های ورودی و قاعده یادگیری *STDP* تغییرات وزن های سیناپسی به صورت زیر اند:

Synaptic weight changes according to STDP learning rule



parameters

neurons: 3 * LIF(10+np.random.rand()*3,8+np.random.rand()*4,-79,-50,5,-68)

learning_rule: STDP

time_interval: 50 mS

dt: 0.3125 mS

dt_minus: 6 mS

dt_plus: 10 mS

a_minus: -4

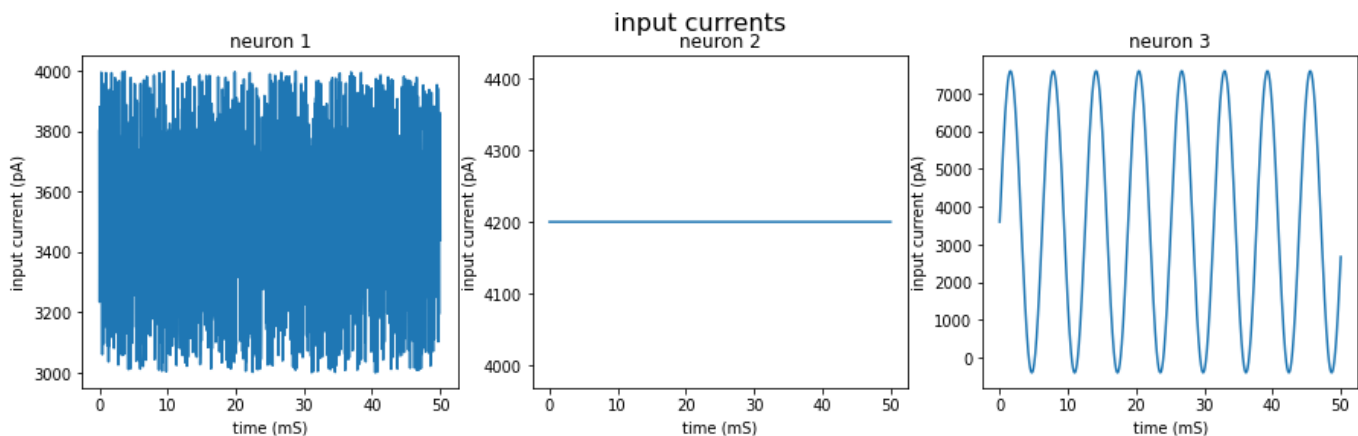
dt_plus: +6

time course thresholds: 0.3

همانگونه که مشخص است تغییرات تمام وزن ها نسبت به نقطه شروع مقداری مثبت بوده است.

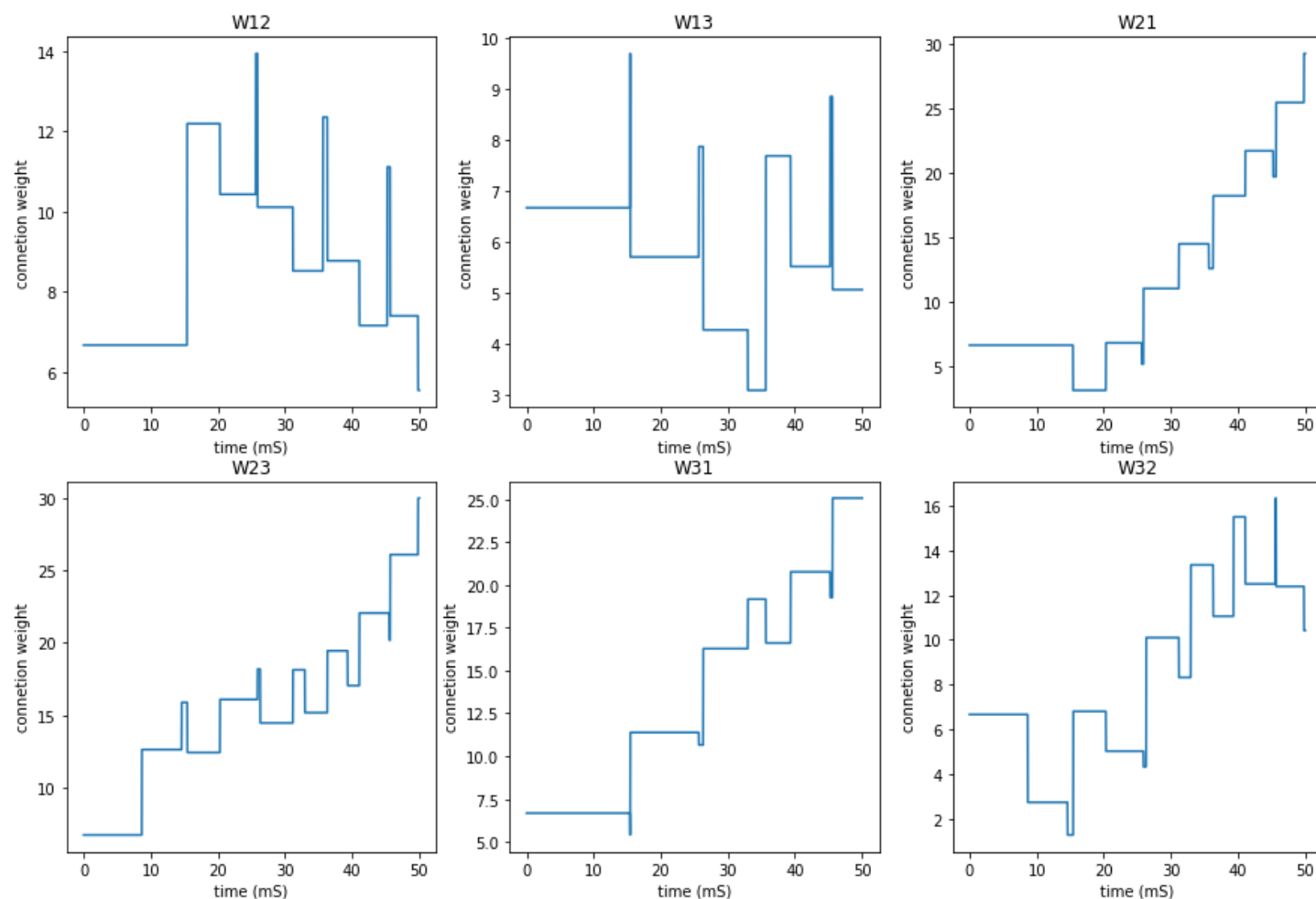
- یک جریان ثابت، یک جریان سینوسی و یک جریان پله ای:

شکل توابع جریان ورودی در زیر آورده شده و هر کدام به یکی از نورون های موجود در جمعیت داده شده اند:



جریان های ورودی به صورت یک جریان ثابت 4200 pA و یک جریان سینوسی با ضابطه $4000 * (\sin(\text{time}) + 0.9)$ و یک جریان پله ای با ضابطه $3000 + 1000 * \text{random}()$ تعریف شده اند که عدد تصادفی عددی بین ۰ و ۱ است و با این جریان های ورودی و قاعده یادگیری $STDP$ تغییرات وزن های سیناپسی به صورت زیر اند:

Synaptic weight changes according to STDP learning rule



و پارامترهای تنظیم شده هم عبارتند از:

parameters

neurons: 3 * LIF(10+np.random.rand()*3,8+np.random.rand()*4,-79,-50,5,-68)

learning_rule: STDP

time_interval: 50 mS

dt: 0.3125 mS

dt_minus: 6 mS

dt_plus: 10 mS

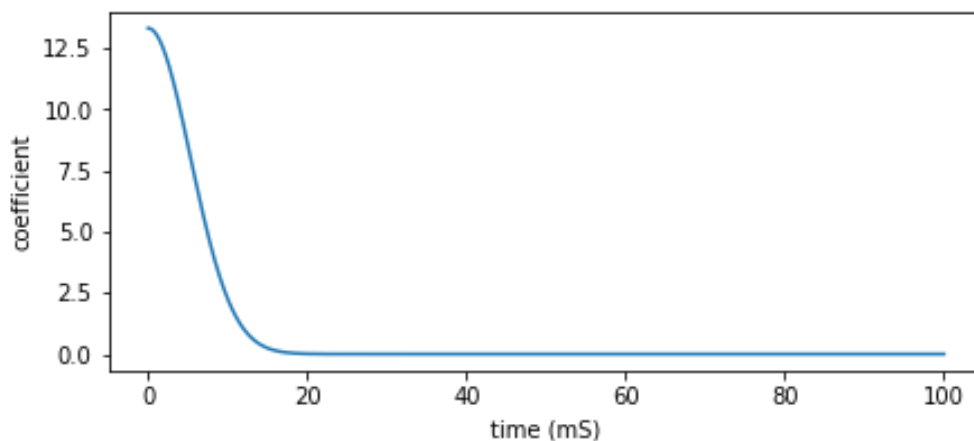
a_minus: -4

dt_plus: +6

time course thresholds: 0.3

۲. برای این قسمت به منظور افزایش سرعت اجرا تابع *time course* کمی تغییر کرده که تصویر نمودار آنرا در زیر مشاهده میکنید.

time course function for SNN



** به دلیل اینکه نورون *LIF* که در ساختار شبکه به کار رفته است به جریان هایی در ابعاد چند 1000 pA به بالا واکنش نشان میدهید به منظور رعایت مقیاس تمامی جریان های ورودی *test* و *train* در عدد 10000 ضرب شده اند.

نکته مورد اهمیت دیگر این است که شبکه اسپایکی هایپر پارامتر های زیادی دارد که تنظیم نادرست آنها میتواند موجب عدم آموزش مناسب شبکه شود این پارامتر ها در زیر نمایش داده شده اند) دقت کنید داریم که این مقادیر لزوما بهترین مقادیر ممکن نیستند):

lif_neuron_parameters:

,R':10'

,tau':8'

,u_rest':-79'

,threshold':-50'

,u_spike':5'

,u_reset':-68'

,random':True'

rand_change':2'

این پارامتر ها پارامتر نورو ن های سازنده شبکه را نشان میدهند . عبارت *random* نشان میدهد که هر کدام از این مقادیر به علاوه مقداری تصادفی به عنوان پارامتر های ورودی تعیین میشوند و *rand_change* نشان دهنده حداکثر مقدار تصادفی است.

connection_details:

, 'connection_type': 'fully_connect'

j':8'

در این بخش هم اطلاعات شبکه آورده شده که شبکه عصبی از نوع *fully_connect* و وزن های اولیه هر کدام مقدار ۸ با یک نویز هستند.

اطلاعات دیگر مربوط به شبکه از جمله مقدار *reward* و تایم کورس دوپامین و تگ های ایجاد شده بین نورو ن های *pre synaptic* و *post synaptic* هم در نوتبوک و در هنگام تعریف شبکه قرار داده شده است.

حال نتیجه دقت مدل بر روی داده های *train* و *test* را میتوانید ببینید:

• Train

Out put: [1., 1., 1., 1., 0., 0., 0., 0., 0., 0.]

accuracy of SNN on train data: 0.8

• Test

Out put: [1., 0., 0., 1., 1., 0., 0., 0., 0., 0.]

accuracy of SNN on test data: 0.7

در شکل صفحه بعد هم تغییرات وزن سیناپسی به ازای هر *epoch* که شامل مقداری آموزش شبکه بر روی یکی از داده های ورودی است نشان داده شده است:

input currents

