

CNS
PROJECT 3

Seyed Mohsen Sadeghi 99222059

STDP

هدف بخش اول این تمرین پیاده سازی الگوریتم یادگیری Stdp بود که برای پیاده سازی آن از مدل های ساخته شده در پروژه های 1 و 2 استفاده میکنیم.

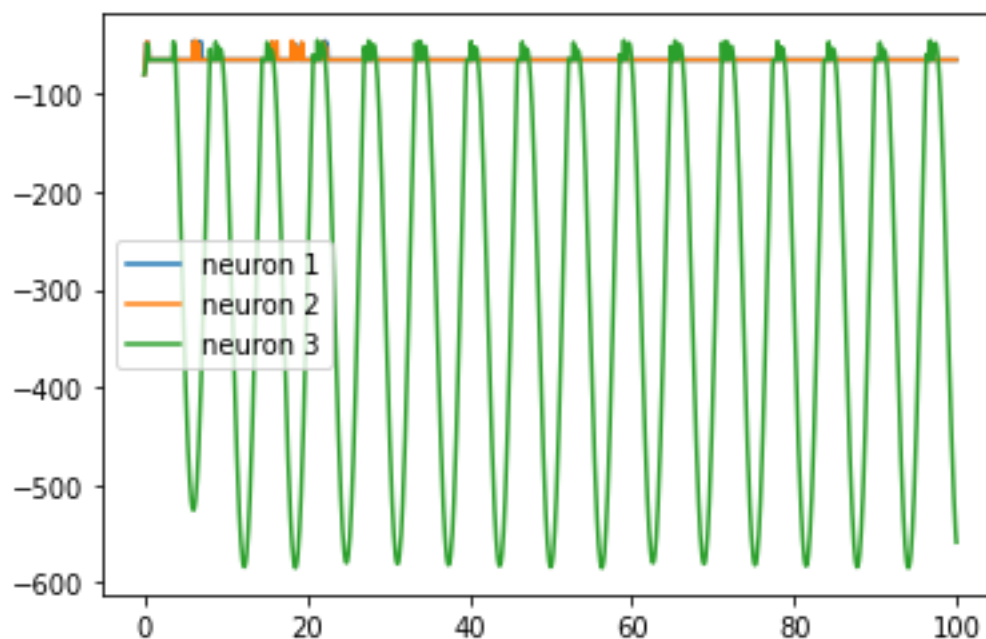
کلاس lif که همان نورون ماست و کلاس neurons group که یک گروه نورونی را نمایش میدهد که در آن اسپایک زدن یک نورون پری سیناپتیک باعث میشود نورون پست سیناپتیک ما به اندازه مقدار وزنی داده شده افزایش پتانسیل داشته باشد که البته این افزایش با اندکی تاخیر اعمال میشود.

در این بخش ابتدا نورون ها را به صورت full connective به هم وصل می کنیم و تغییر w-t ها را بسته با جریان بررسی میکنیم.

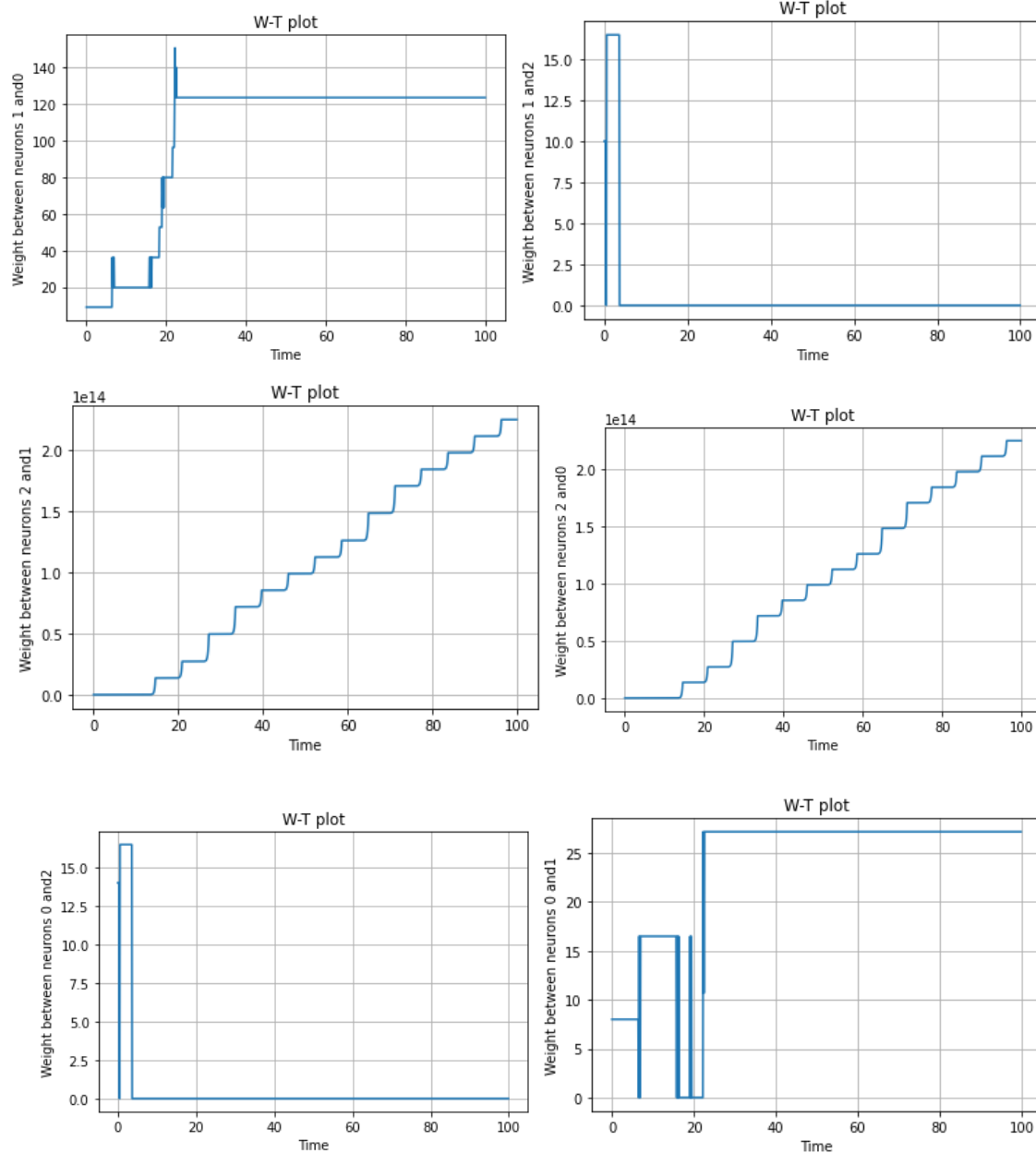
جریان نورون اول و دوم func1 و جریان نورون سوم func2 است.

```
func1 = lambda x: 200 * (math.sqrt(math.fabs(math.sin(x))))  
func2 = lambda x: 250 * math.sin(x)
```

نمودار جریان نورون ها بر اساس ورودی ها و تاثیرشان روی یک دیگر نمودار زیر را تولید میکند.



حال نمودار تغییرات وزن ها را بررسی میکنیم:



همانطور که از بررسی نمودار ها میبینیم هر گاه نورون پری اسپایک میزند مقدار وزن ها زیاد شده و هرگاه نورون پست فایر میکند مقدار وزن ها کاهش می یابد.

R-STDP

در بخش دوم پروژه می خواهیم با استفاده از یک SNN الگوریتم یادگیری را پیاده سازی کنیم که در این مدل لایه اول ما 5 نورون و لایه دوم ما 2 نورون دارد با توجه به دیتا ست داده شده باید مدل خود را طوری طراحی کنیم که ورودی های مد نظر خروجی های مد نظر را بدهند.

	test	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10
0	input_neuron_number	train_1	train_2	train_3	train_4	train_5	train_6	train_7	train_8	train_9	train_10
1	1.0	1.0	2.0	3.0	2.0	1.0	1.0	0.0	0.0	0.0	2.0
2	2.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	2.0
3	3.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
4	4.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	1.0	0.0	1.0
5	5.0	0.0	0.0	0.0	0.0	1.0	2.0	0.0	1.0	1.0	2.0
6	output_neuron_number	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	test	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	input_neuron_number	train_1	train_2	train_3	train_4	train_5	train_6	train_7	train_8	train_9	train_10
10	1.0	1.0	0.0	0.0	2.0	1.0	1.0	3.0	0.0	1.0	2.0
11	2.0	0.0	1.0	2.0	2.0	2.0	0.0	1.0	0.0	2.0	2.0
12	3.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	0.0	2.0
13	4.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	1.0	3.0
14	5.0	0.0	0.0	0.0	0.0	0.0	3.0	3.0	2.0	1.0	3.0
15	output_neuron_number	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0

نکته : برای این که ورودی ها قابل قبول تر باشند داده ها در 10000 ضرب شده اند.

سپس با دادن ورودی ها به مدل و بررسی خروجی ها میبینیم در برابر داده های ورودی تمرینی مدل ما 0.7 دقت داشته و در برابر داده تست 0.8 دقت داشته است.