

بخش اول پروژه: یادگیری STDP

- یادگیری STDP چیست؟ به طور خلاصه در این یادگیری تغییر وزن های سیناپسی تابعی از فاصله زمانی بین اسپایک دو نورون (pre و post) می باشد.
- به طور کلی تغییرات وزن های سیناپسی از فرمول زیر به دست می آید:

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

- ایده این فرمول ها از اینجا آمده که در مغز برای شبیه سازی یادگیری unsupervised میتوانیم از فاصله زمانی بین اسپایک دو نورون کمک کنیم. زیرا اسپایک بیشتر دو نورون در یک بازه زمانی مشخص، نشان دهنده بالا بودن ارتباط آنها و در نتیجه افزایش وزن سیناپسی آنها می باشد.
- حال برای ساده سازی از مدل نرونی LIF استفاده می کنیم.

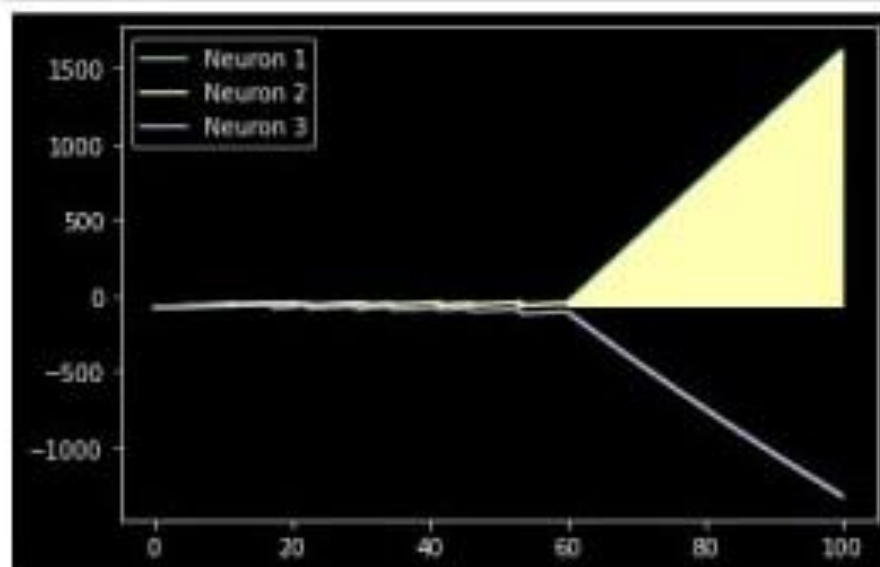
در این بخش باید به سه نورون که به هر کدام جریان متفاوتی وارد می شود را شبیه سازی کنیم. ابتدا جریان ها را به شکل مقابل وارد می کنیم:

```
I1 = lambda x: 60
I2 = lambda x: 20
I3 = lambda x: 40
neuron1 = LIF(I=I1)
neuron2 = LIF(I=I2)
neuron3 = LIF(I=I3)

neurons = [neuron1, neuron2, neuron3]
connections = [(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)]
weights = [5, 5, 5, 5, 5, 5]
```

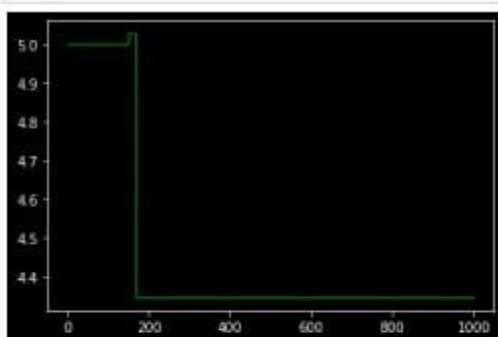
این مقادیر نشان دهنده وزن های اولیه 5 برای هر دو رابطه دلخواه می باشد. جریان های 60، 20 و 30 به ترتیب به نورون های 1 و 2 و 3 وارد می شوند.

نمودار زمان – پتانسیل نورون ها به شکل زیر می باشد:

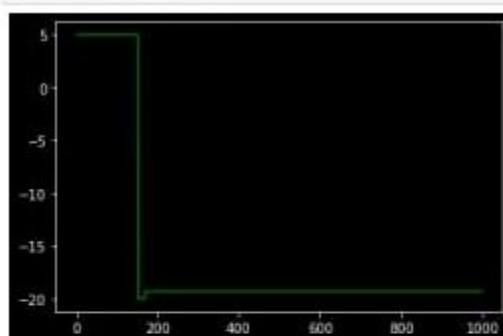


همچنین نمودار تغییرات برخی از وزن های به شکل زیر می باشد:

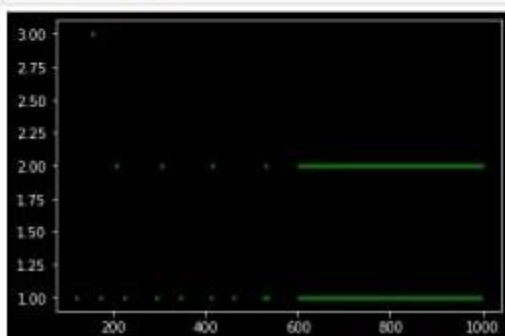
`stdp.wt_plot(3, 1)`



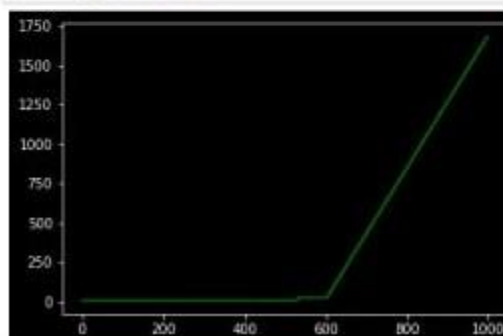
`stdp.wt_plot(1, 3)`



`stdp.scatter_plot()`



`stdp.wt_plot(1, 2)`



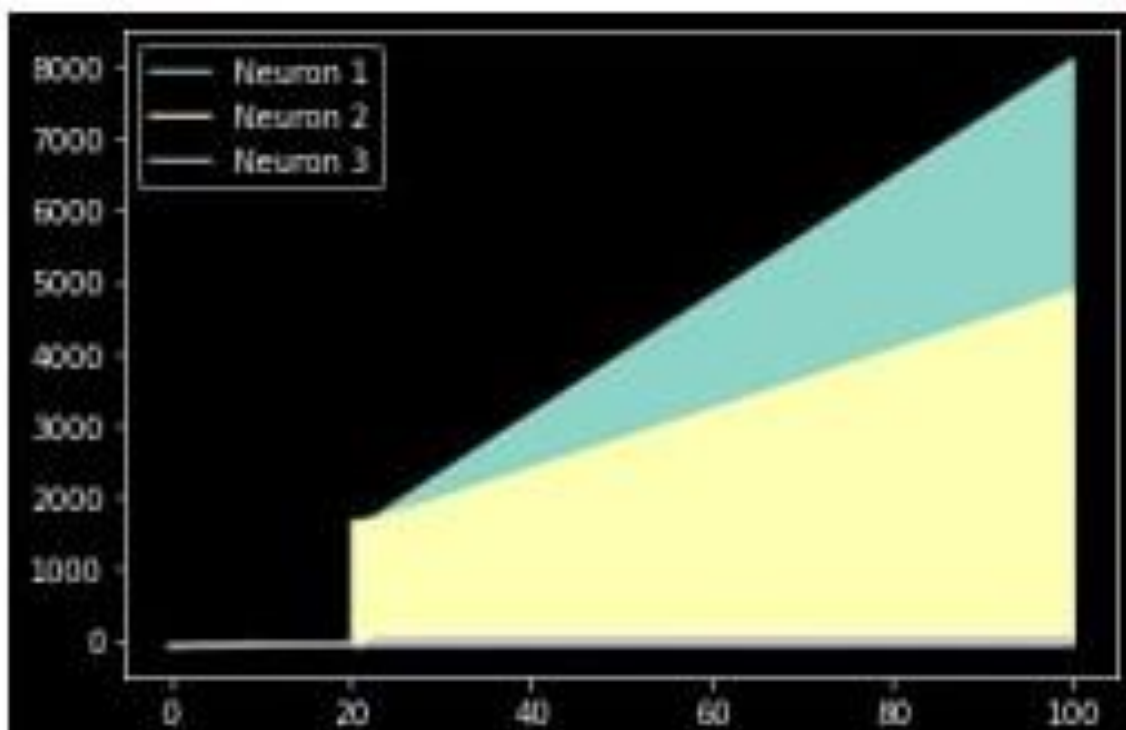
تغییرات وزن ها از [5, 5, 5, 5, 5, 5] به [1681.4980615987085, -5.000760812076495, 1715.8641925469694, 19.316886150578593, 4.999239187923505, 4.344730529274715] می باشد.

حال اگر مقادیر را به شکل زیر وارد مدل کنیم:

```
I1 = lambda x: 20 * (math.sin(x/10) + 0.6)
I2 = lambda x: 30 * (math.sin(x/10) + 0.3)
I3 = lambda x: 40 * (math.cos(x/10) + 0.9)
neuron1 = LIF(I=I1)
neuron2 = LIF(I=I2)
neuron3 = LIF(I=I3)

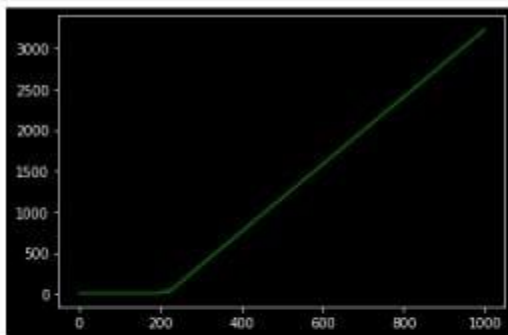
neurons = [neuron1, neuron2, neuron3]
connections = [(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)]
weights = [5, 5, 5, 5, 5, 5]
```

در این آزمایش مقادیر وزن ها مانند آزمایش قبل می باشد. همچنین مقدار جریان وارد شده به هر نورون به صورت سیسنوسی است. در این حالت نمودار تغییرات پتانسیل در طول زمان به این شکل خواهد بود:

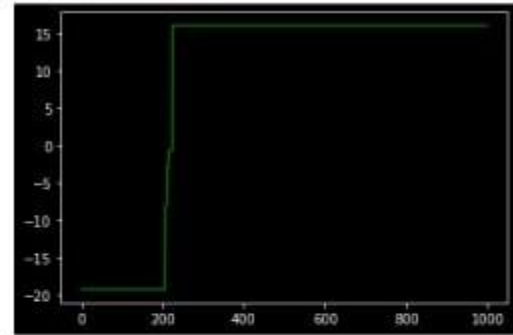


همچنین نمودار تغییرات برخی از وزن ها به شکل زیر می باشد:

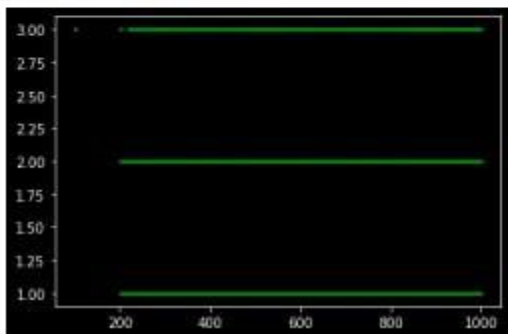
stdp.wt_plot(3, 1)



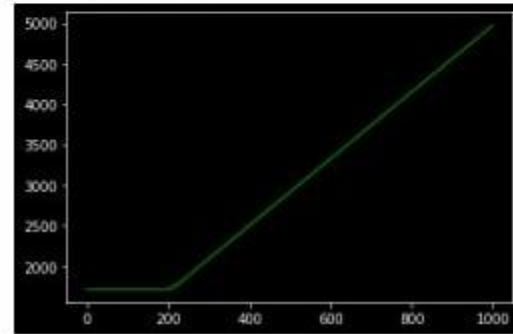
stdp.wt_plot(1, 3)



stdp.scatter_plot()



stdp.wt_plot(2, 1)



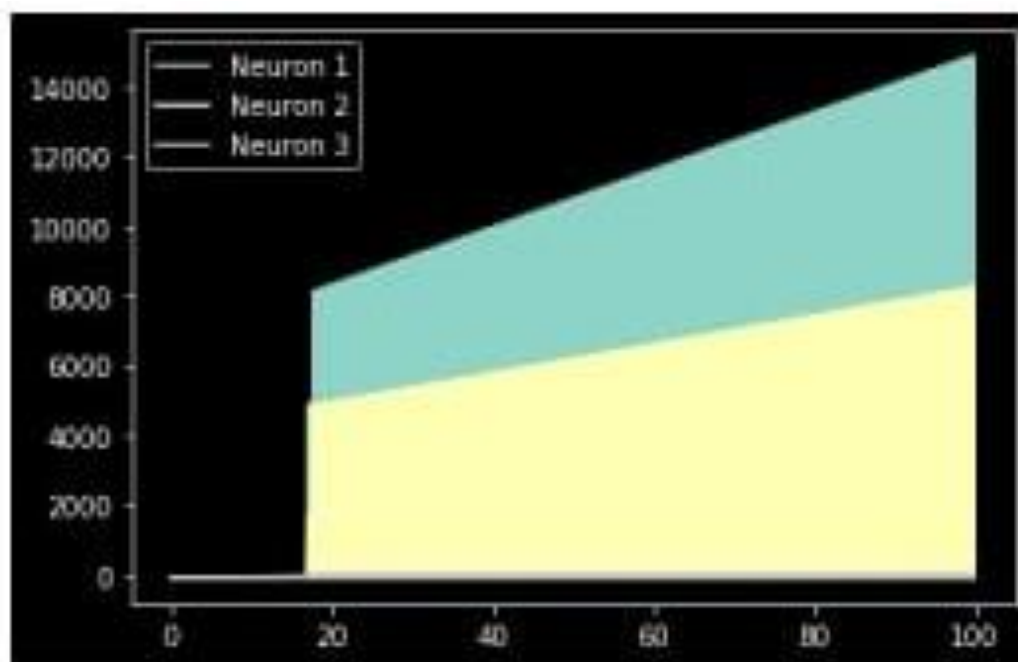
تغییرات وزن ها از [5, 5, 5, 5, 5, 5] به [18.613892092461434, 30.986161024614013, 2896.48453828127, 27.282778370512837, 2890.9597472985497, 2870.7723669462302] می باشد.
در سومین آزمایش جریان های وارد شده به مدل به شکل زیر می باشد:

```
I1 = lambda x: 5 * x
I2 = lambda x: 3 * x
I3 = lambda x: 2 * x
neuron1 = LIF(I=I1)
neuron2 = LIF(I=I2)
neuron3 = LIF(I=I3)

neurons = [neuron1, neuron2, neuron3]
connections = [(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)]
weights = [5, 5, 5, 5, 5, 5]
```

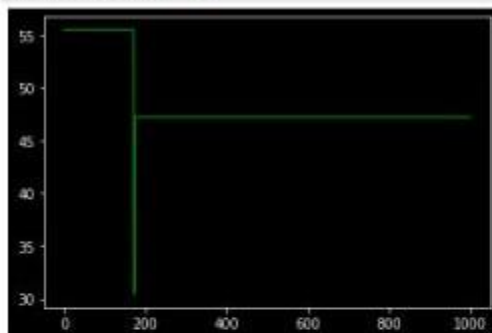
در این آزمایش نیز وزن های اولیه هر دو نورون 5 می باشد و جریان ها به شکل یک تابع خطی وارد مدل شده است.

نمودار تغییرات پتانسیل نورون ها بر حسب زمان به شکل زیر می باشد:

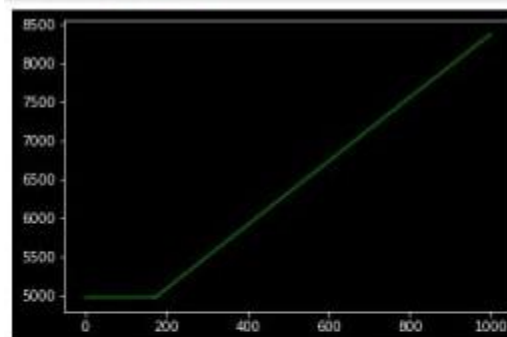


همچنین تغییرات وزن بین برخی از نورون ها به شکل زیر می باشد:

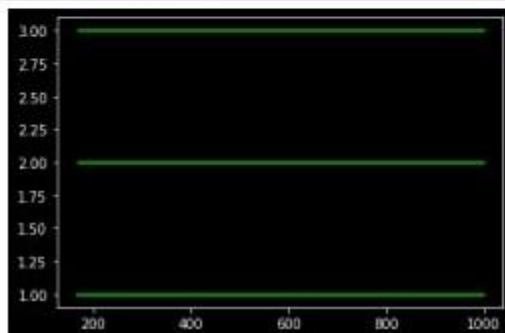
`stdp.wt_plot(2, 3)`



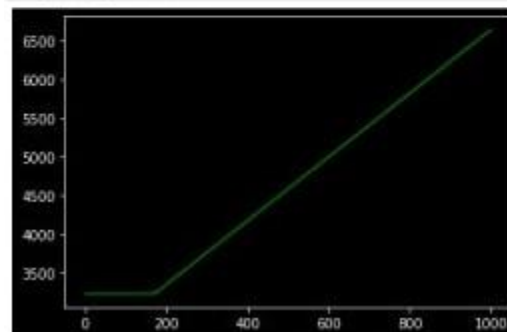
`stdp.wt_plot(2, 1)`



`stdp.scatter_plot()`



`stdp.wt_plot(3, 1)`



تغییرات وزن ها در این حالت نیز از [5, 5, 5, 5, 5] به [1528.187307396355, -19.999999996172452, 1530.4105238035152, 19.88693140606008, 21.644932556951062, 16.23322409910299] می باشد.

بخش دوم: یادگیری Reward based STDP

- یادگیری RSTDP چیست؟ در این یادگیری با شبیه سازی ترشح دوپامین در مغز می خواهیم یک مدل بسازیم.
- این یادگیری در طبقه بندی Reinforcement learning قرار می گیرد.
- زمانی که مغز در اثر یک سری اسپایک نوروئی دوپامین ترشح می کند، این بدان معنی است که این عمل برای بدن مفید بوده، پس در نتیجه وزن های سیناپسی مربوط به آن اسپایک افزایش پیدا می کند تا این Reward بیشتر به بدن برسد.
- از فرمول های زیر برای شبیه سازی این نوع یادگیری استفاده می شود:

$$\frac{dc}{dt} = -\frac{c}{\tau_c} + STDP(\tau)\delta(t - t_{pre/post}),$$

$$\frac{ds}{dt} = cd,$$

در 1000 iteration از دیتاست داده شده استفاده میکنیم و وزن های سیناپسی را تغییر میدهیم. سپس آنها را با استفاده از دیتاست تست، صحت سنجی میکنیم. دیتاست داده شده به شکل مقابل می باشد:

	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	2.0
2		2.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	2.0
3		3.0	0.0	1.0	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	1.0
5		5.0	0.0	0.0	0.0	0.0	1.0	2.0	0.0	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
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
11		2.0	0.0	1.0	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	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
14		5.0	0.0	0.0	0.0	0.0	0.0	3.0	3.0	2.0	1.0
15	output_neuron_number	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0