Seyed Hossein Mohammadi (96222096)

Project 3 – STDP Learning

Part 1

After some modification on synapse are simulated, I was able to simulate synaptic input with delays (every synapse is actually a few synapses, each with a certain delay).

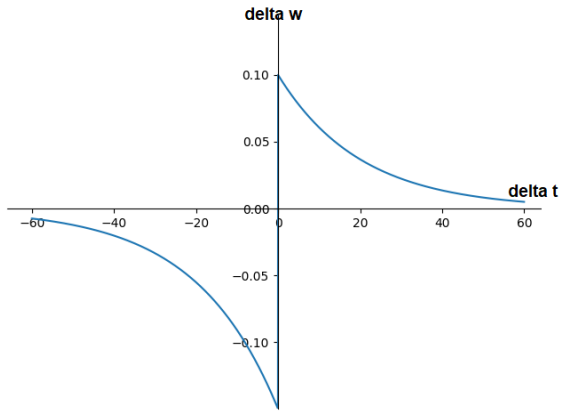
Afterwards I implemented STDP as a class that has 2 important functions: “train_pre_to_post_syn” and “train_post_to_pre_syn” which do as their name suggests.

I tested the STDP alg. with this configuration:

```
(phi_p_e=0.01, phi_p_i=0.03, rate=20, phi_n_e=0.015, phi_n_i=0.045, w_max=1, tau_p=20, tau_n=20)
```

And this delta w function:

```
def delta_w_func(self, x, pre_w, is_excretory):
    if x >= 0:
        phi = self.phi_p_e if is_excretory else self.phi_p_i
        a = phi * (self.w_max - pre_w); tau = -self.tau_p
    else:
        phi = self.phi_n_e if is_excretory else self.phi_n_i
        a = -phi * pre_w; tau = self.tau_n
    delta_w = self.rate * a * exp(x / tau)
```



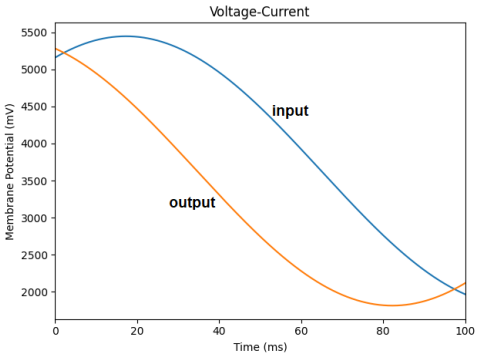
Note that the negative portion of delta_w is more than 50% stronger than the positive section.

The configuration of the 2 AELIF neuron is as follows:

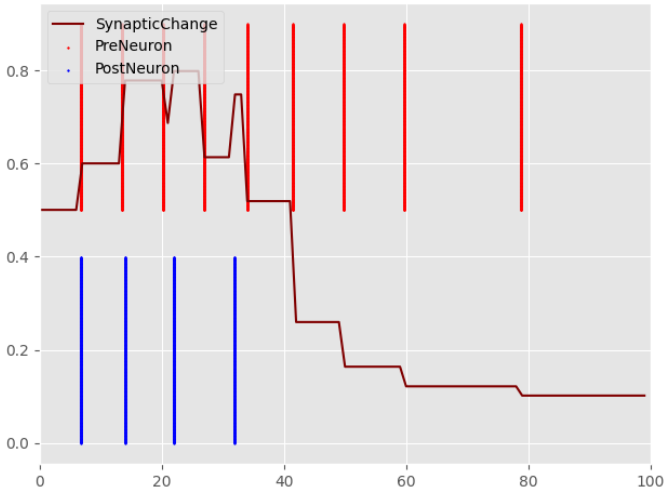
```
n_config="dt=0.03125, R=10, tau=8, theta=-40, U_rest=-75, U_reset=-65, U_spike=5, "
"weight_sens=11,ref_period=2, ref_time=0, theta_rh=-45, delta_t=2, a=0.01, b=500, tau_k=100"
```

The connection model has J=1 with only one connection:

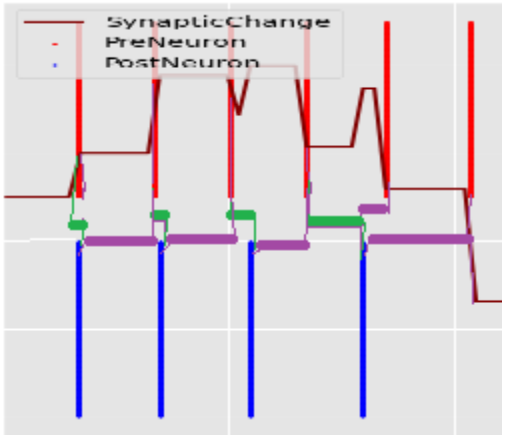
The current used for the 2 neuron:



Part 1 results:



This graph clearly shows a how STDP is functioning.



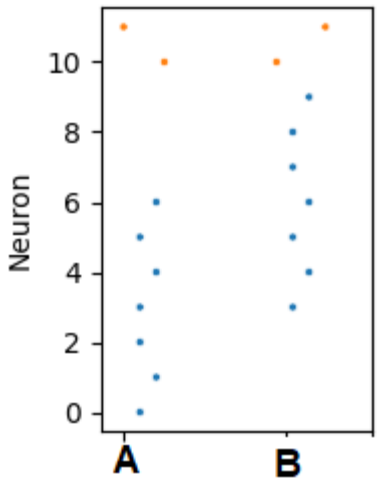
Here, green lines show positive delta_t which should cause a slight increase in w and purple shows negative delta_t which should cause a significant decrease in w also note that shorter lines, cause a larger change in w.

Part 2

Here I used the function “set_neurons_state” in population class to manually cause (or suppress) neuron firing.
The neurons are manually configured in 50ms intervals so to give time for simulation to take place.
This act much like supervised learning where input and output are fixed.

If a pattern is set for input, the output neuron will fire a few ms after wards so to strengthen the weights with STDP but this will ONLY strengthen the connection (a.k.a increase weights).

to weaken unrelated and harmful connection to a output neuron A (which responds to pattern A), I force neuron A to fire right before pattern B as input. This will force STDP to decrease neuron A sensitivity to pattern B.
here are 2 examples:



Note that for pattern A, neuron B (11) fires right before the pattern is shown.

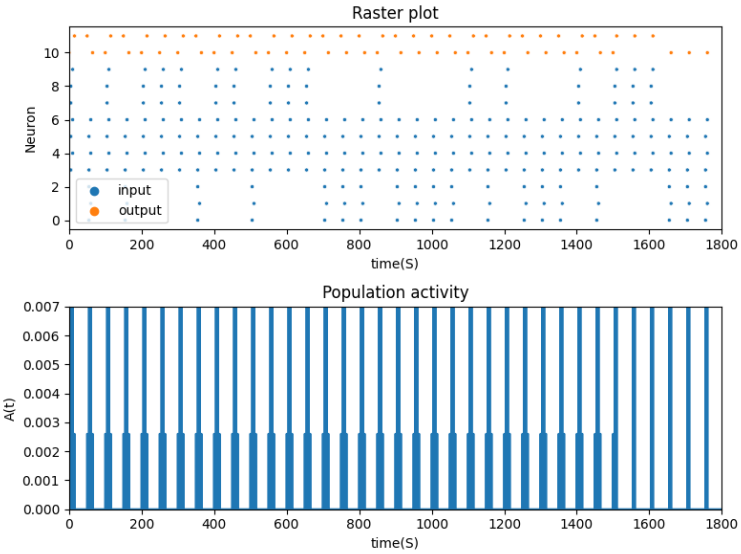
Notes about the patterns:

1. These patterns will be used for the final results.
2. Neuron 10 should respond to Neuron 0 to 6 and neuron 11 to neurons 3 to 9
3. Neurons 3 to 6 are common in both input pattern A and B.

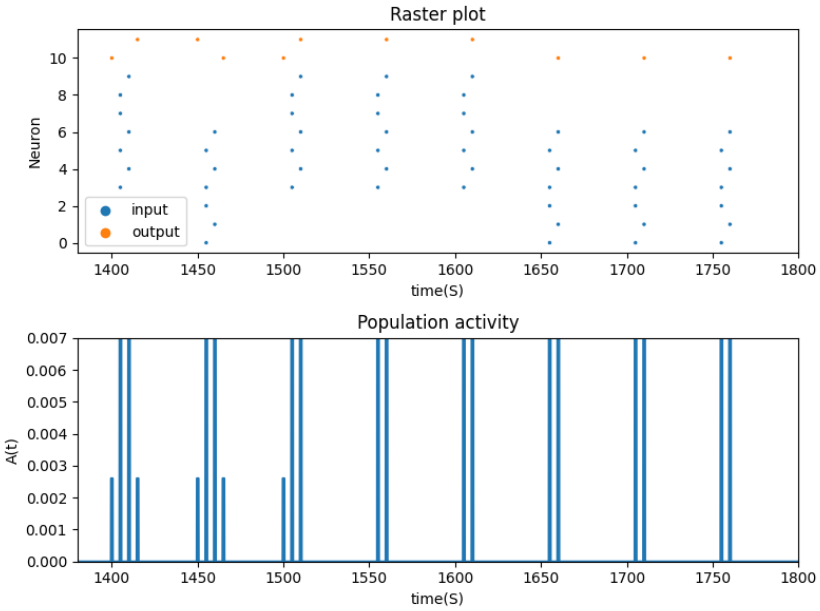
Part 2 results:

A network of 12 excitatory ELIF neurons (2 output, 10 input) are paired, each with 1 instantaneous synapse (has no delay).

First, I use the above-mentioned method to train the network for 1500ms with 50ms between each pattern. Image below shows both train and test results (last 300ms is the test time)

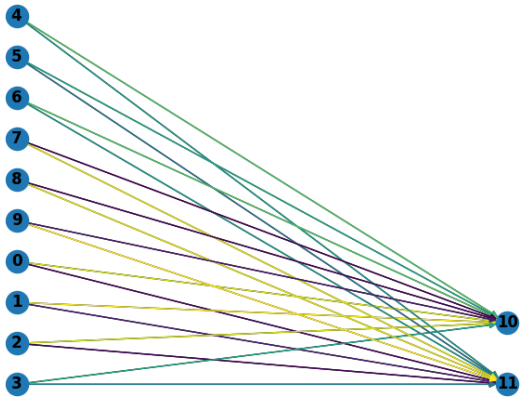


The test section (last 300ms)
the STDP leaning was disabled and only the input was fixed:



As you see, output neurons (orange ones) have learned the patterns quite well.

The final weights of the network are as follows (note: the numbers are not in order):



As you see, common input neurons in both patterns are not as relevant to the main difference. This is because every time, those weights would first see a decrease and then an increases. Overall causing a relatively weak, above zero, connection.

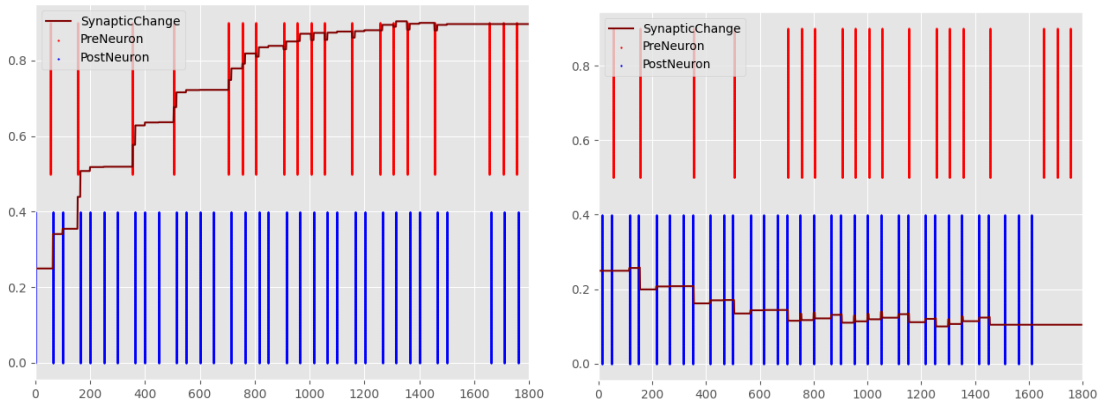
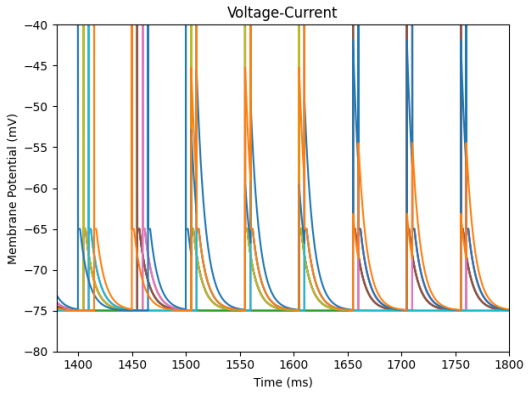
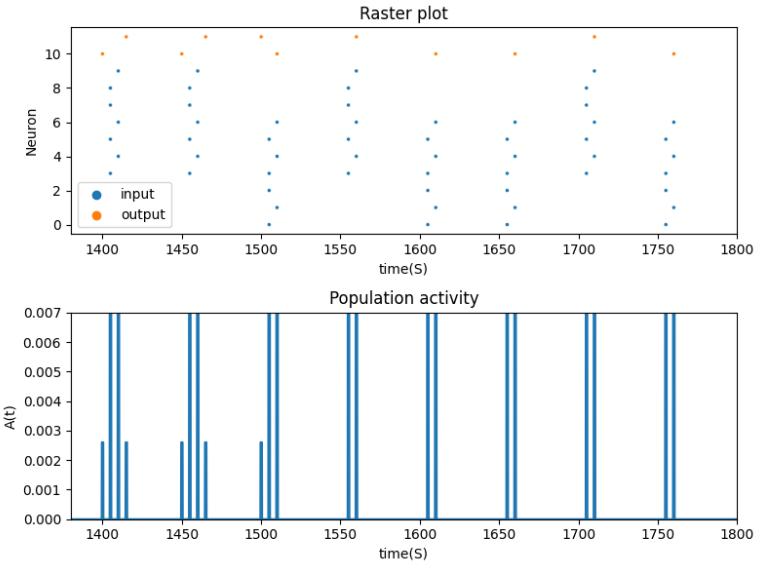


Image to the left is the weight change of the connection from neuron 0 to neuron 10 and Image to the right is from neuron 0 to neuron 11. The red and blue bars show spikes for each neuron.

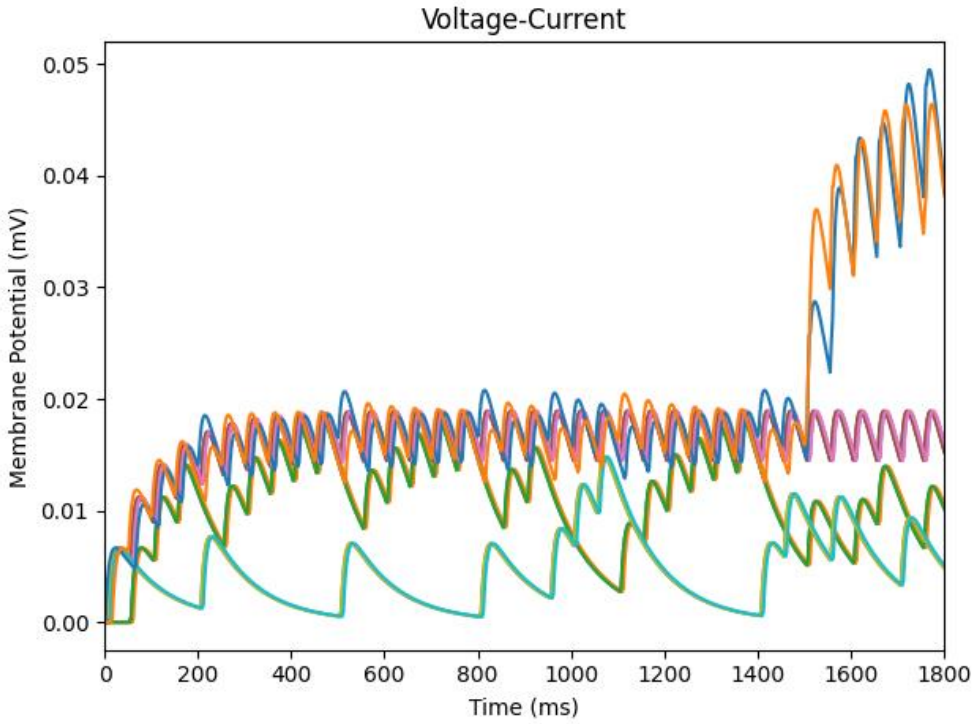
Voltage of each neuron:



The same tests were done with an AELIF network and the final result is as follows:



AELIF would stop working after some time duo to internal current rising to high:

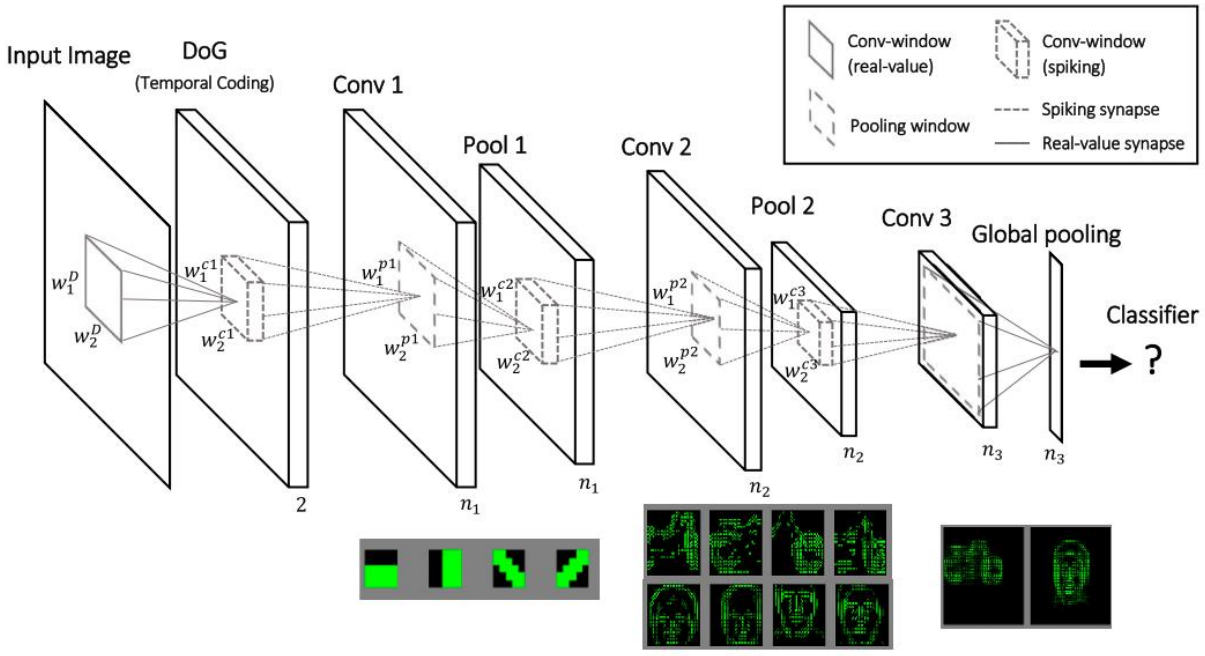


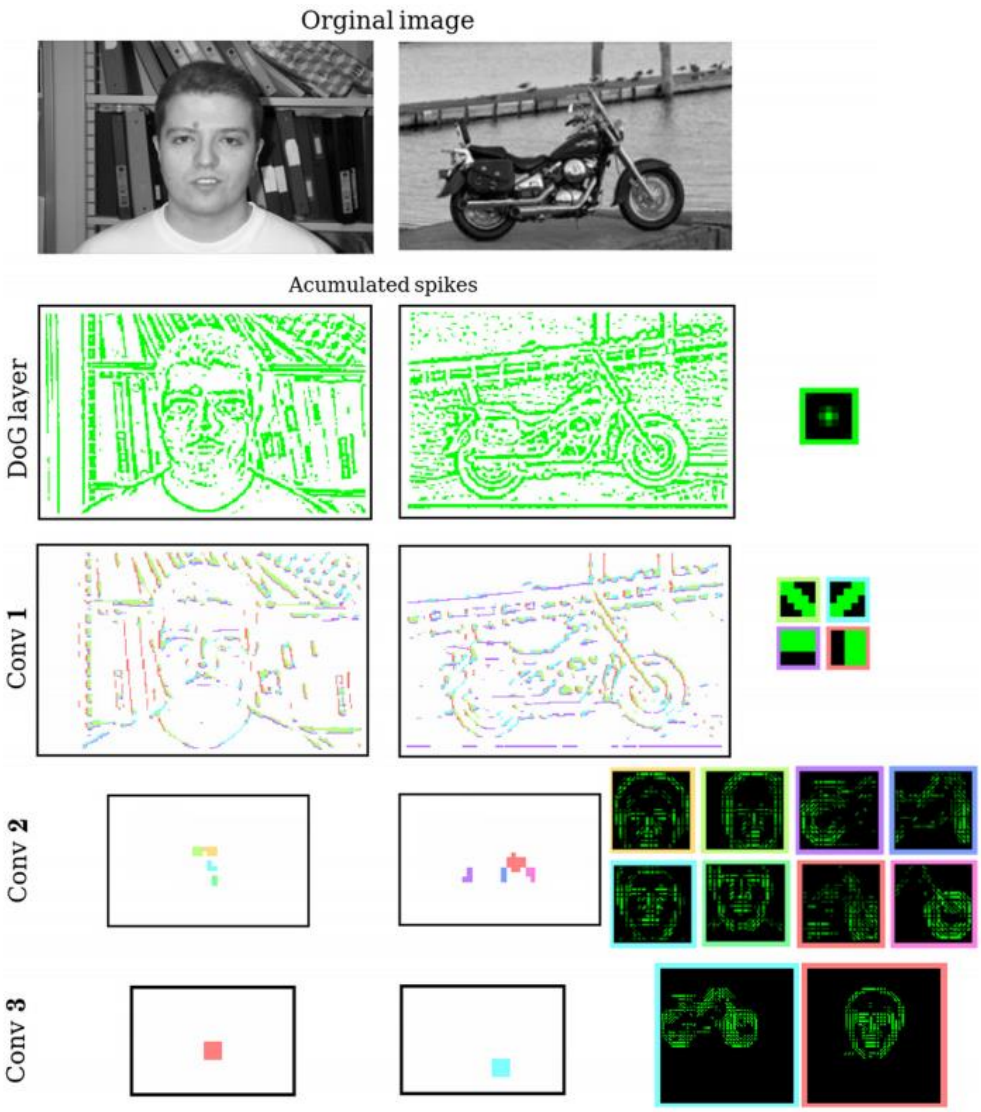
Part 3

Key takeaways:

As the first paper to use STDP in a deep SNN network, the proposed model archives an Accuracy: 99.1% on face/motorbike task and 82.8% on ETH-80 which is not bad, but not great.
A Temporal rank coding is used to both reduce total spikes (energy efficiency) and to help with the winner takes all learning

The network is comprised of first a DoG to simulate the ganglion cells which detect contrast. Then we have the first convolution layer which acts much like the V1 in brain (recognize and map lines and their angle). Afterwards a max polling layer (which simply propagating the first spike) helps with invariance of the network by mapping similar features to one point all the while helping to compress the visual data. The next layer is a convolution layer along with a pooling layer is added that acts much like the V2 and V4 region of the brain (recognize and map shapes and objects).
Again, another convolution layer along with a pooling layer is added, this time to simulate the behavior of IT region of the brain that recognizes the actual thing in an image like a face or a motorbike.
A final global pooling is also used to map the overall image into one point for classification by SVM.

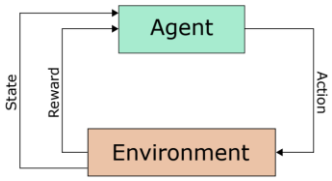




Training will only happen in weights coming to conv layers. Each neuron in the convolution layer has an inhibitory effect on its nearby neurons to help with the winner takes all mechanism; also, the first neuron to spike, triggers STDP. After weight adjustment, the new value is shared to all nearby neurons.

Part 4

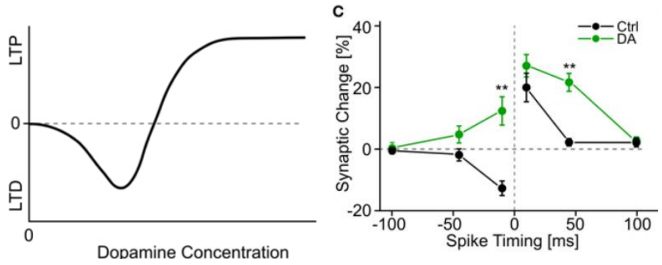
-STDP یک یادگیری unsupervised هست اما RSTDP یک یادگیری reinforcment هست به این معنا که عملی توسط شبکه انجام میشود و محیط به عنوان واکنش تشویق یا تنبیه میکند (که ممکن هست کمی دیر تر ظاهر شود) که باعث تقویت یا تضعیف اون عمل میشود.



-دو نوع تقویت ممکن هست:

- Classical Conditioning که در آن یک عامل مثل زنگوله و یک reward یا punishment با هم مرتبط میشوند.
- Operant Conditioning که در آن یک عمل به همراه یک عامل به یک reward یا punishment مرتبط میشوند.

-یک مشکل با این روش، تاخیر بین عمل و reward هست و اینکه چگونه یک شبکه بداند که کدام چیز باعث reward شده (Distal Reward Problem). یک را حل، دوپامین هست که میتواند یک الگو را ثانیه ها بعد از ایجاد شدن reinforce کند. به زبان دیگر، غلظت دوپامین نحو عملکرد STDP را تغییر میدهد. غلظت کم دوپامین باعث تشدید شدن LTD و غلظت بالا لاعث تشدید شدن LTP میشود



<u>Reward modulated STDP</u>: دوپامین روی polarity و بازه STDP تاثیر میگذارد. برای مدل کردن دوپامین، از متغییر c استفاده میکنیم که مدلی برای Synaptic Eligibility Trace (Synaptic Tag) خواهد بود. (s همان وزن نورون هست و d غلظت دوپامین و τc بازه زمانی تاثیر گذار را مشخص میکند)

- Synaptic Tag فرایندی هست که در آن تاثیر اسپایک کردن نورونی برای مدتی باقی میماند و هنگامی که دوپامین وارد شود، به نسبت آنکه غلظت کم یا زیاد باشد، نورونی هایی که eligibility شان بالاتر هست، تقویت یا تضعف میشوند.

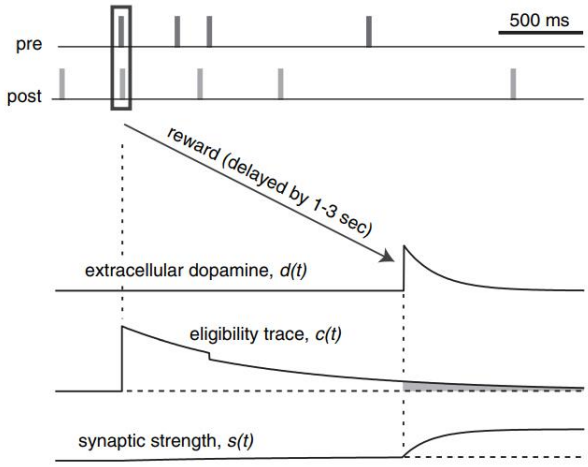
$$\frac{dc}{dt} = -\frac{c}{\tau_c} + STDP(\tau)\delta(t - t_{pre/post}),$$
$$\frac{ds}{dt} = cd,$$

$$\tau = t_{post} - t_{pre}$$

همچنین، d در طول زمان کم میشود که توسط رابطه زیر مدل شده که در آن DA مقدار دوپامینی هست که وارد میشود. (τd بازه زمانی تاثیر گذار را مشخص میکند)

$$\frac{dd}{dt} = -\frac{d}{\tau_d} + DA(t)$$

مثالی از عملکرد این روابط



برای چه مسئله ای مناسب تر هست و چرا؟ این یادگیری از نوع reinforcement learning هست و به همین دلیل در این یادگیری برای مسائلی بهتر هست که داده به صورت آماده وجود ندارد و دسترسی به یک محیط تعاملی موجود هست و باید عملی را برای شرایط خاصی در محیط انتخاب کرد.

چون این یادگیری، میتواند مدتی بعد از شروع اسپایک اولیع صورت گیرد، برای مسائلی که محیط در آن ها با تاخییر، جایز/جریمه میدهد بسیار مفید هست.