گزارش پروژه فاز چهارم نوروساینس

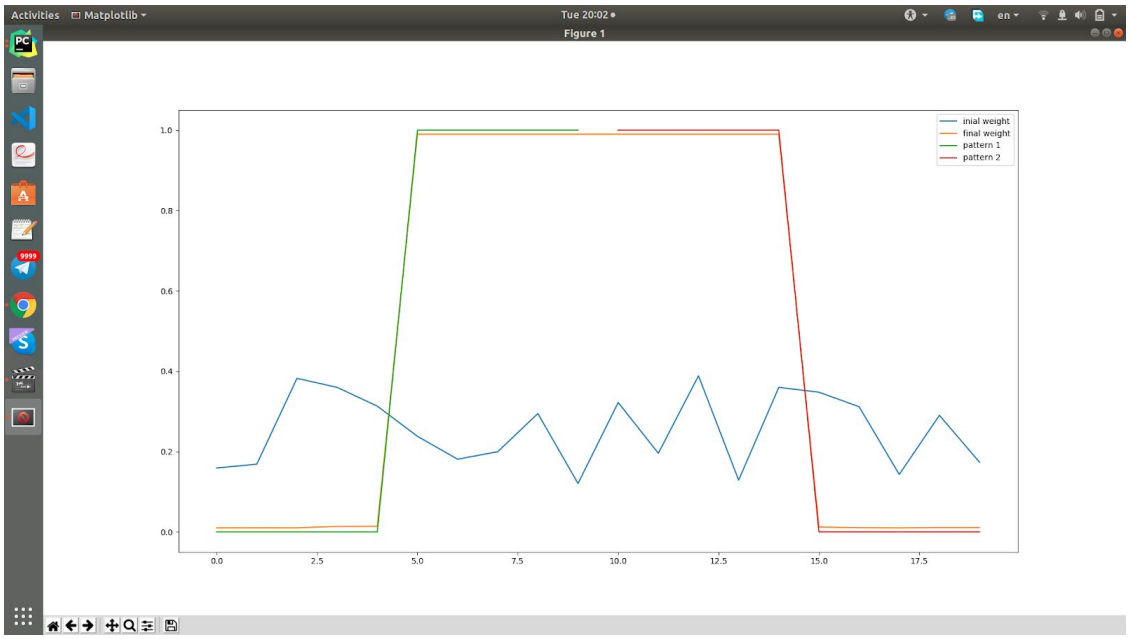
ریحانه درفشی ۶۱۰۳۹۶۰۹۸

قسمت اول:

پیادسازی قانون یادگیری تشویقی

```
def calculate_synaptic_tag(self, t):
   delta_c = -self.c / self.tau_c + self.stdp(self.pre, self.post) *
max(self.alpha(self.post, t),self.alpha(self.pre, t))
   self.c = self.c + delta_c
   self.c = min(0.3, self.c)
   self.c = max(0, self.c)
```

```
def calculate_weight(self,dopamine):
   delta_w = self.c * dopamine
   self.weight += delta_w
   self.weight = max(self.weight,0.01)
   self.weight = min(self.weight,0.99)
```



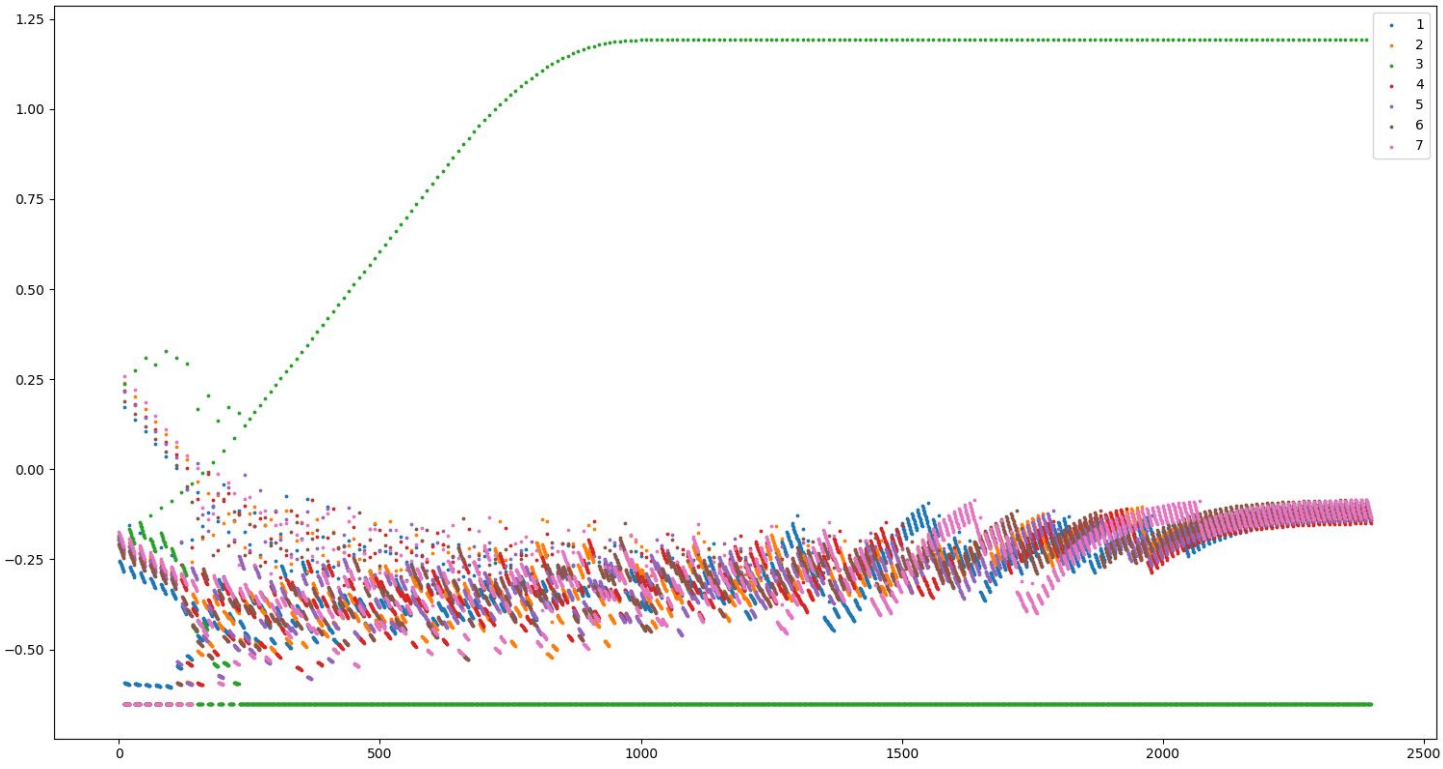
قسمت دوم:

یک network شامل ۱۰۰ نورون که شامل ۸۰ نورون تحریکی و ۲۰ نورون مهاری در نظر گرفته شده، یک جمعیت ۳۰ تایی برای لایه ورودی و هفت جمعیت ۱۰ تایی برای لایه های خروجی، سیناپس های بین این نورون ها به صورت رندوم تشکیل شده و وزن نورون ها نیز همین طور و برای هر لایه خروجی یک پترن رندوم نیز ایجاد شده، میتوان یک پترن را به عنوان پترن ورودی انتخاب کرد، در این صورت وقتی نورون های لایه متناظر با پترن بعد از هر بار اجرای پترن اسپایک میزنند به مقدار دوپامین اضافه میشود و در صورتی که نورون های لایه های دیگر بعد از پترن غیر متناظر اسپایک بزنند مقدار دوپامین کم می شود، دوپامین یک متغیر گلوبال است که در همه جای کد مقدار یکسانی دارد. مقدار synaptic tag و weight برای هر سیناپس همان طور که در قسمت اول نشان داده شده محاسبه می شود. تعداد نورون های کلی، نورون های لایه ورودی و خروجی و احتمال وجود سیناپس بین یک نورون لایه ورودی و خروجی قابل تغییر است، فعالیت هر لایه در هر یک دهم ثانیه به صورت میانگین ولتاژ نورون های آن لایه در آن زمان محاسبه شده و نمایش داده میشود.

```
self.neuron_numbers = 100
self.input_layer_size = 30
self.n = 10
self.inh_size = 20
self.p = 0.5
```

```
a = Network()
a.time(3).draw()
```

در نمودار صفحه بعد محور افقی نشان دهنده زمان است و محور عمودی نشان دهنده فعالیت، هر رنگ نشان دهنده فعالیت یکی از جمعیت های خروجی است، و چون پترن سوم به اجرا درآمده فعالیت جمعیت سوم به تدریج زیاد شده است.


1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
0
500
1000
1500
2000
2500
1
2
3
4
5
6
7