

گزارش پروژه سوم (تمرین یک و دو)

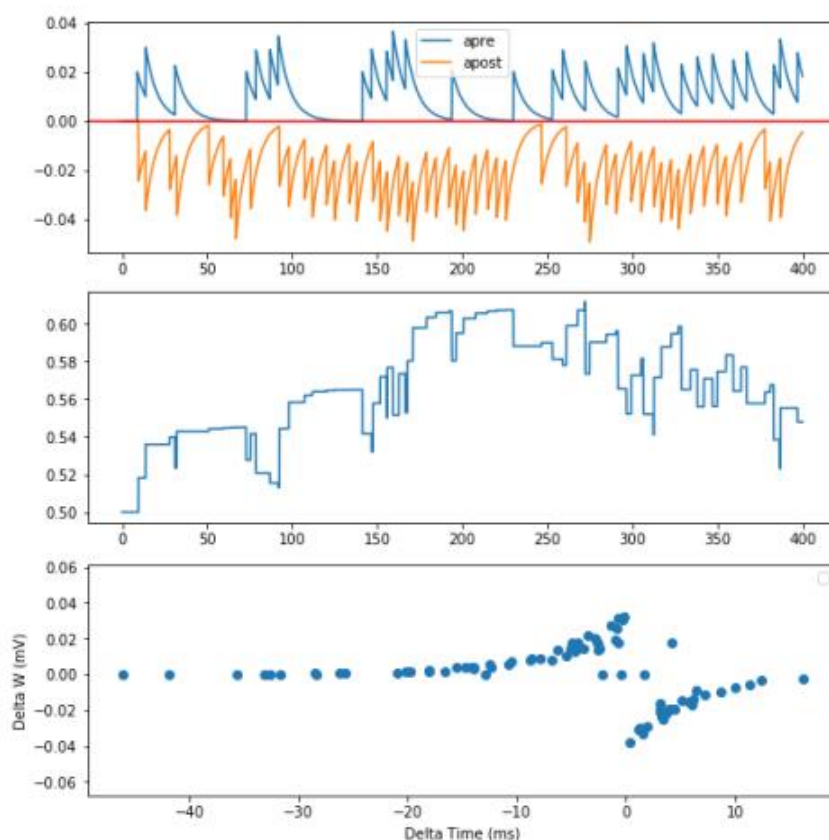
امیرحسین برهانی-96222022

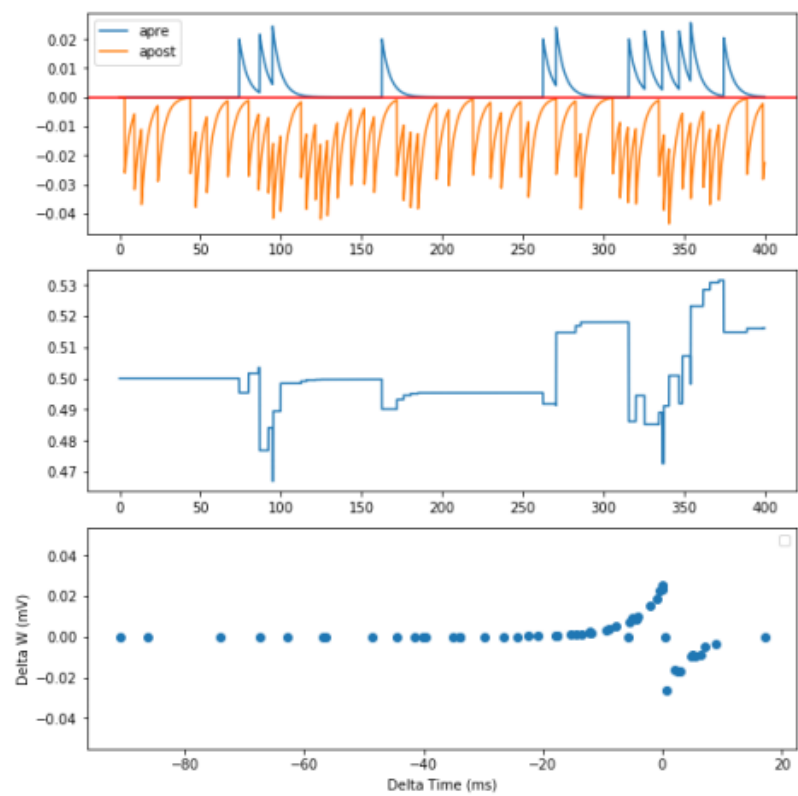
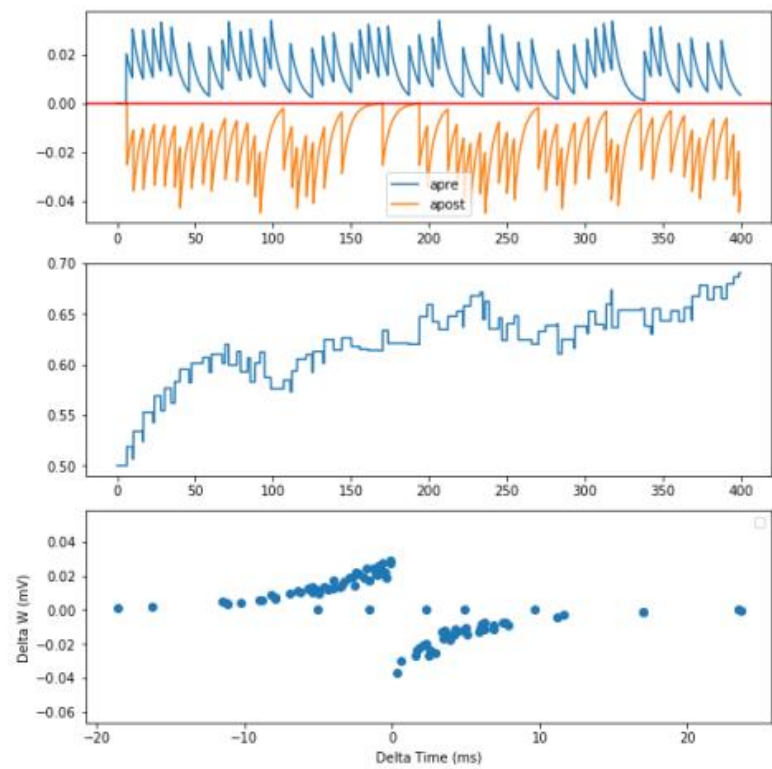
در این نمودار ها خطوط عمودی قرمز نشاندهنده حضور الگوی 1، خطوط آبی نشان دهنده حضور الگوی 2 و آخرین خطی که به رنگ نارنجی عالمت گذاری شده است نشاندهنده حضور نویز است، نویز در زمان آموزش نیز وجود دارد اما در آن جا با خطوط رنگی مشخص نشده است و هر جایی که آبی یا قرمز نباشد نویز قرار دارد. انتظار داریم در نمودار های 4 و 5 مقدار پتانسیل در حضور الگوی یک و دو خیلی بیشتر از حضور نویز باشد اما چنین نیست و تقریباً در یک سطح قرار دارند در این پروژه قصد داریم مدل یادگیری stdp را پیاده سازی کنیم .

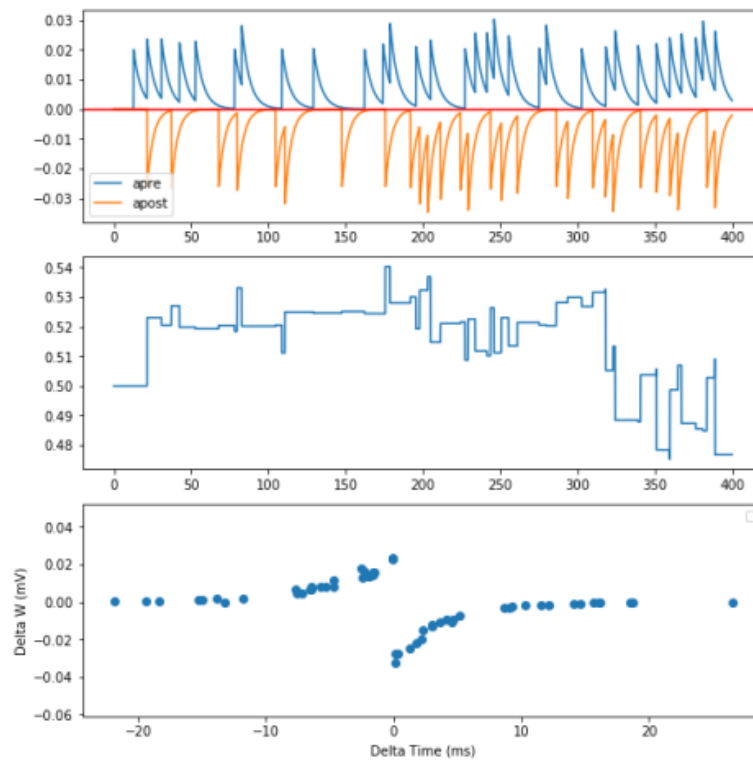
برای سوال اول از دو نورون استفاده میکنیم که به هر دو نورون جریان ورودی پالسی وارد میکنیم تا به پتانسیل شان به حد آستانه برسد و سپس بر اساس زمان spike زدن نورون ها وزن اتصال بین آن ها را تنظیم میکنیم.

برای اینکار از مدل نورونی LIF استفاده کردیم و پتانسیل نورون pre و post در بخش اول نمودار ها قابل مشاهده است که در آن پتانسیل نورون pre با رنگ آبی در بالا و پتانسیل نورون post با رنگ نارنجی و برعکس در پایین قرار دارد و این نمایش به این دلیل است که در یادگیری stdp در زمان spike زدن نورون pre باید به اندازه پتانسیل نورون post ضربدر یک ضریب ثابت $-A$ مقدار w را کاهش دهیم، همینطور در زمان spike زدن نورون post به اندازه پتانسیل نورون pre ضربدر یک ضریب ثابت $+A$ مقدار w را افزایش دهیم که از رابطه زیر پیروی میکند.

در بخش دوم نمودار نیز تغییرات وزن را مشاهده میکنید و در بخش سوم نمودار نیز تغییرات w نسبت به فاصله بین اسپایک نورون pre و post را مشاهده میکنید. در زیر چند نمونه از این آزمایش هارا مشاهده میکنید.







متاسفانه موفق به پیاده سازی درست تمرین شماره 2 نشدم، توضیح مختصری از کد تمرین دوم در زیر موجود است . کد شامل 3 تابع است:

: Make_poisson_pattern_current

ورودی ها :

N : تعداد نورون های pre

Patterns : الگوهای موجود برای این نورون ها

Durratıon : زمان شبیه سازی

Delta_time : کوچک ترین بازه زمانی

Possibility : احتمال وجود الگو در زمان اسپایک های نورون pre

از خروجی این تابع در SpikeGeneratorGroup استفاده میشود و قصد داریم زمان تمام اسپیک های نورون های pre در طول آزمایش را در این تابع مشخص کنیم . برای اینکار ابتدا بزرگترین عدد موجود در الگو ها را پیدا میکنیم و آن را L مینامیم . سپس کل بازه مورد نظرمان را با گام هایی به طول L پیشروی میکنیم و در هر گام با توجه به مقدار possibility تصمیم میگیریم که آیا باید در این گام از الگو ها استفاده بکنیم یا از نویز ، اگر قرار شد از یک الگو استفاده کنیم باز هم بین الگو هایی که در patterns وجود داد یکی را به صورت تصادفی انتخاب میکنیم و زمان اسپایک ها را با توجه به آن محاسبه و ذخیره میکنیم و اگر قرار شد از نویز استفاده کنیم.

مقداری random برای هر یک از نورون ها در بازه مربوطه مشخص میکنیم به صورتی که وارد گام بعدی نشود . از 400 ثانیه آخر برای مرحله test استفاده میکنیم ، در ثانیه 300 -duration الگوی اول را وارد میکنیم، در ثانیه

200-duration الگوی دوم را وارد میکنیم و در ثانیه 100-duration نویز وارد میکنیم تا در آخر بتوانیم از روی این سه مرحله درستی یادگیری را بررسی کنیم.

:pulse_input_current_generator

ورودی ها:

Max_I : مقدار جریان در زمان پالس زدن

Duration : زمان شبیه سازی

Delta_time : کوچک ترین بازه زمانی

Possibility : احتمال وجود جریان پالسی

این تابع به ازای تمام لحظات شبیه سازی یکی از دو مقدار 0 و **max_i** را به عنوان جریان ورودی در آن لحظه انتخاب میکند که احتمال انتخاب **max_i** برابر **possibility** و احتمال انتخاب 0 برابر **1-possibility** است.

از این تابع برای تولید جریان ورودی در نورون های **post** استفاده میشود و در آن **max_i** برابر 0 در نظر گرفته میشود یعنی عملاً به ازای تمام لحظات جریان 0 وارد نورون های **post** میشود و یعنی تنها عامل افزایش پتانسیل در این نورون ها اسپایک زدن نورون های **pre** میباشد.

Simulate

ورودی ها:

Coef : : در این شبیه سازی وزن ها بین 0 تا 1 قرار دارند و چون مقدار کوچکی است، در زمان اسپایک زدن نورون های **pre** ضریب ثابت **coef** ابتدا در **w** ضرب میشود و سپس مقدار آن به پتانسیل نورون **post** اضافه میشود.

Duration : زمان شبیه سازی

Apré : ضریب **A+** در فرمول STDP

I_1,I_2,P_1,P_2 : مقادیری که برای دو نورون **post** در تابع **pluse_input_current_generator** استفاده میشود و همانطور که گفته شد همه برابر 0 در نظر گرفته میشوند.

P_pattern : مقدار **possibility** در تابع **make_poisson_pattern_current**

Tau_pre : ثابت زمانی نورون های **pre synaptic**

اندیس 1 و 2 اشاره به نورون **post synaptic** اول و دوم دارد.

U_rest_post : پتانسیل **rest** نورون مربوطه

Tau_post : ثابت زمانی نورون مربوطه

Threshold_post : پتانسیل آستانه نورون مربوطه

Resistance_post : مقاومت نورون مربوطه-از آنجا که مقدار جریان ورودی در این دو نورون برابر 0 است و مقدار مقاومت در آن ضرب میشود این مقدار اهمیتی ندارد.

در این تابع پس از فراخوانی توابع بالایی و ابتدایی از خروجی تابع اول یک spikegeneratorgroup برای نوروں های pre میسازیم و نوروں های post را نیز با یک مدل lif و به کمک مقادیر ورودی میسازیم و این نوروں ها را به یکدیگر متصل میکنیم و stdp را در بخش synapses پیاده سازی میکنیم.

نمودار ها شامل 8 بخش است:

1: مقدار پتانسیل نوروں pre و post برای یکی از سیناپس های دلخواه .

2: مقدار پتانسیل نوروں اول در طول کل زمان آموزش و تست

3: مقدار پتانسیل نوروں دوم در طول کل زمان آموزش و تست

4: مقدار پتانسیل نوروں اول در زمان تست

5: مقدار پتانسیل نوروں دوم در زمان تست

6: تغییرات وزن ها در زمان آزمایش

7: زمان اسپایک نوروں های post

8: زمان اسپایک نوروں های pre

در این نمودار ها خطوط عمودی قرمز نشاندهنده حضور الگوی 1 ، خطوط آبی نشان دهنده حضور الگوی 2 و آخرین خطی که به رنگ نارنجی علامت گذاری شده است نشاندهنده حضور نویز است، نویز در زمان آموزش نیز وجود دارد اما در آن جا با خطوط رنگی مشخص نشده است و هر جایی که آبی یا قرمز نباشد نویز قرار دارد. انتظار داریم در نمودار های 4 و 5 مقدار پتانسیل در حضور الگوی یک و دو خیلی بیشتر از حضور نویز باشد اما چنین نیست و تقریبا در یک سطح قرار دارند.