

به نام خدا

### گزارش پروژه ۳

امیرحسین میرزاده / ۹۶۲۲۲۰۸۲

در قسمت اول پروژه، الگوریتم STDP برای آپدیت وزن ها بر اساس زمان اسپایک ها پیاده سازی شده است. مدل نوروئی مورد استفاده ما در این قسمت، LIF می باشد، که در فایل کد، دو کلاس LIF و LIF2 وجود دارد که کلاس LIF2، با تابع جریان single step کار می کند.

کلاس NeuronType در واقع یک enumerator است که تحرکی با مهارتی بودن نوروئ را مشخص می کند.

کلاس NeuronGroups، چند نوروئ را به هم متصل می کند.

تابع initiate داخل این کلاس، وزن نوروئ ها را بر اساس این که کدام نوروئ پیش یا پس سیناپسی اسپایک زده، آپدیت می کند.

سپس، کلاس STDP، که آرگومان های اشیای آن نوروئ ها، اتصالات آن ها، جدول وزن ها، میزان تاخیر آپدیت وزن بعد از اسپایک، و تعداد تکرار الگوریتم است. Amplify\_pos\_val کران بالای ما برای تغییرات وزن ها و Amplify\_neg\_val کران پایین تغییرات وزن، برای کنترل تغییرات وزن است.

تابع initiate در این کلاس، وظیفه آپدیت وزن ها بر اساس ترتیب اسپایک نوروئ ها را برعهده دارد.

کلاس SNN، وظیفه پیاده سازی بخش دوم پروژه را برعهده دارد که یک شبکه عصبی اسپایکی است، که reward-based بوده و در صورت رخ دادن حالت مطلوب (اسپایک نوروئ پیشین قبل از پسین)، باعث افزایش وزن می شود. آرگومان های این کلاس شامل ثابت تاو، نوروئ ها، ابعاد شبکه و میزان دوپامین و reward در نظر گرفته شده است.

تابع connect\_network، اتصالات سیناپسی بین لایه های ما را برقرار می کند. تابع fit، شروع به ترین کردن شبکه ما از طریق تابع train می کند تا زمانی که تمام داده ها به تعداد epoch مان به خورد شبکه داده شده باشند.

تابع predict، برای پیشبینی داده های تست طراحی شده است.

تابع جریان داده شده به شبکه ما:

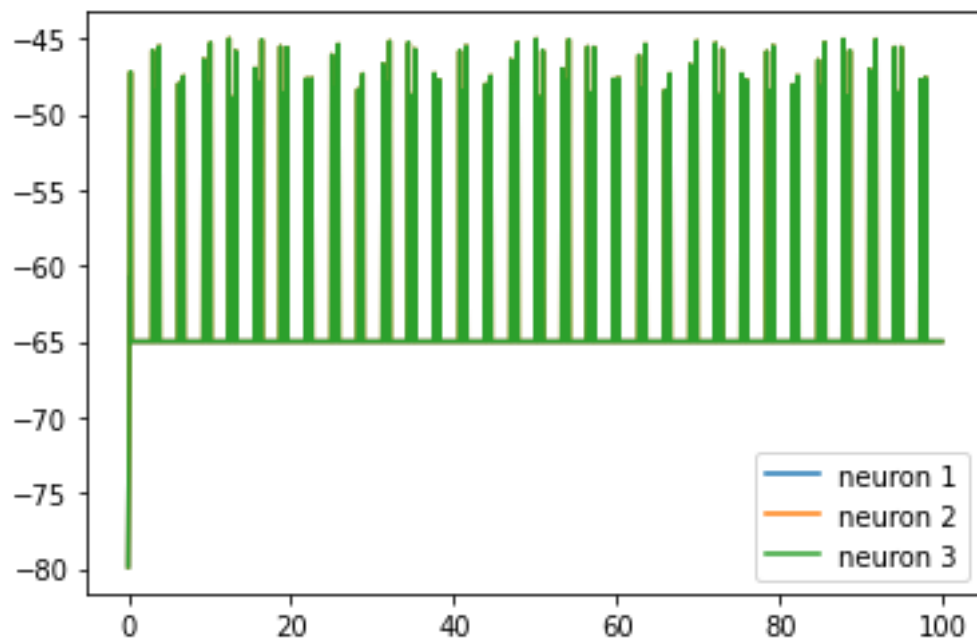
```
const = lambda x: 200 * (math.sqrt(math.fabs(math.sin(x))))
```

نوروئ ها و اتصالات آن ها:

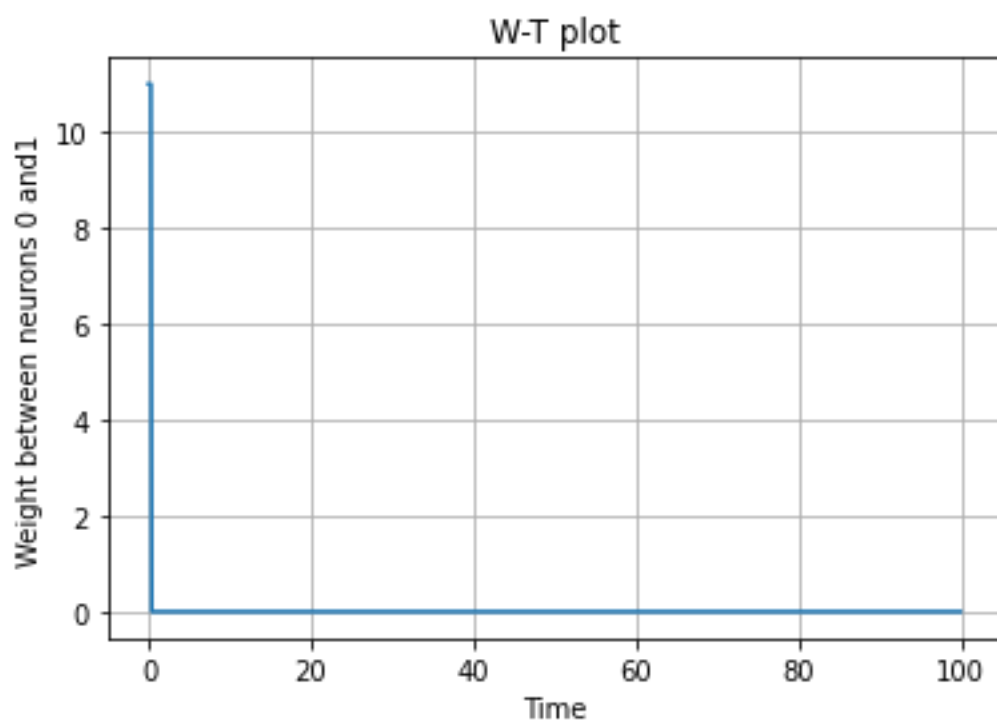
```
neuron1 = LeakyIntegrateAndFire(input_current=const, threshold=-60)
neuron2 = LeakyIntegrateAndFire(input_current=const)
neuron3 = LeakyIntegrateAndFire(input_current=const)

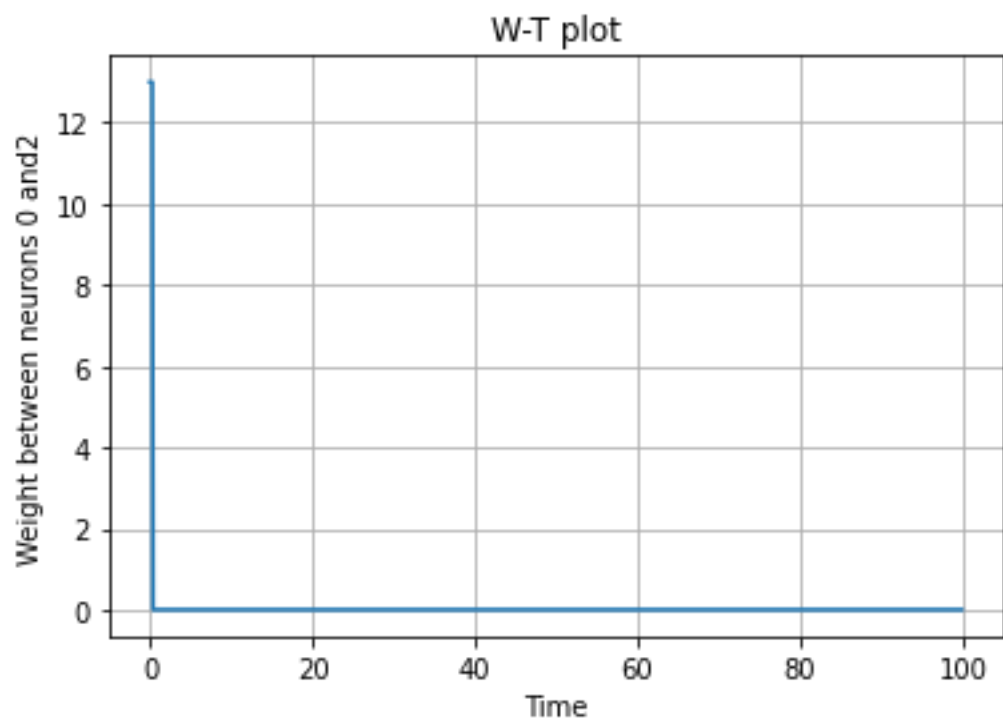
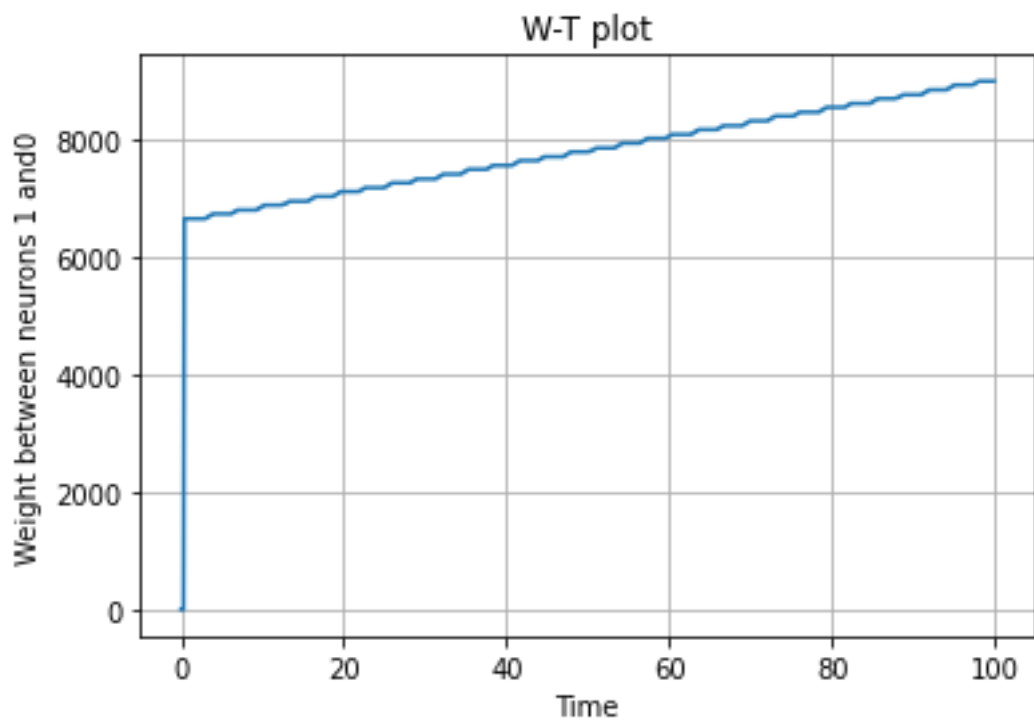
connections = [[1, 2], [0, 2], [0, 1]]
neurons = [neuron1, neuron2, neuron3]
```

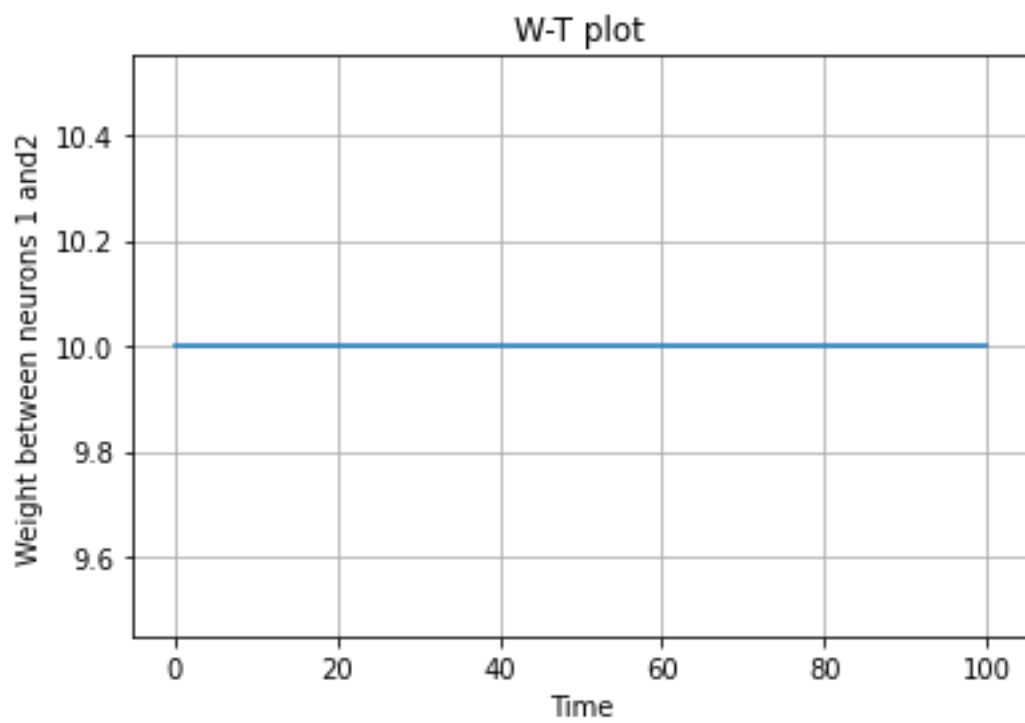
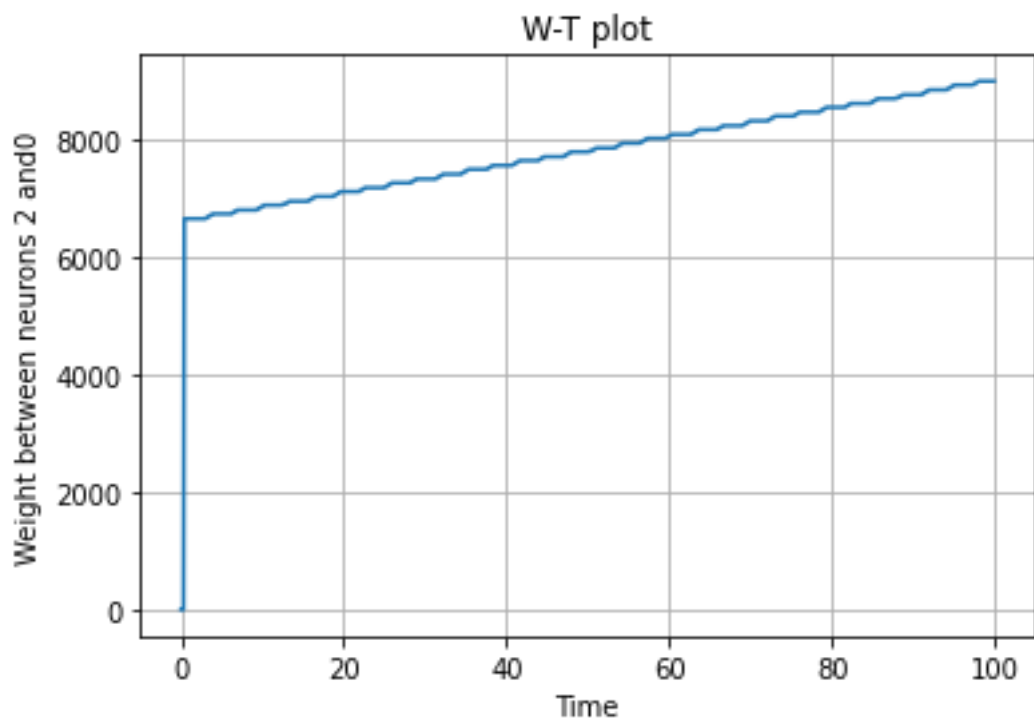
نمودار زیر، نمودار اسپایک های هر ۳ نوروئ ما در طول زمان است. این اسپایک ها در لیست activity\_history ذخیره می شوند.

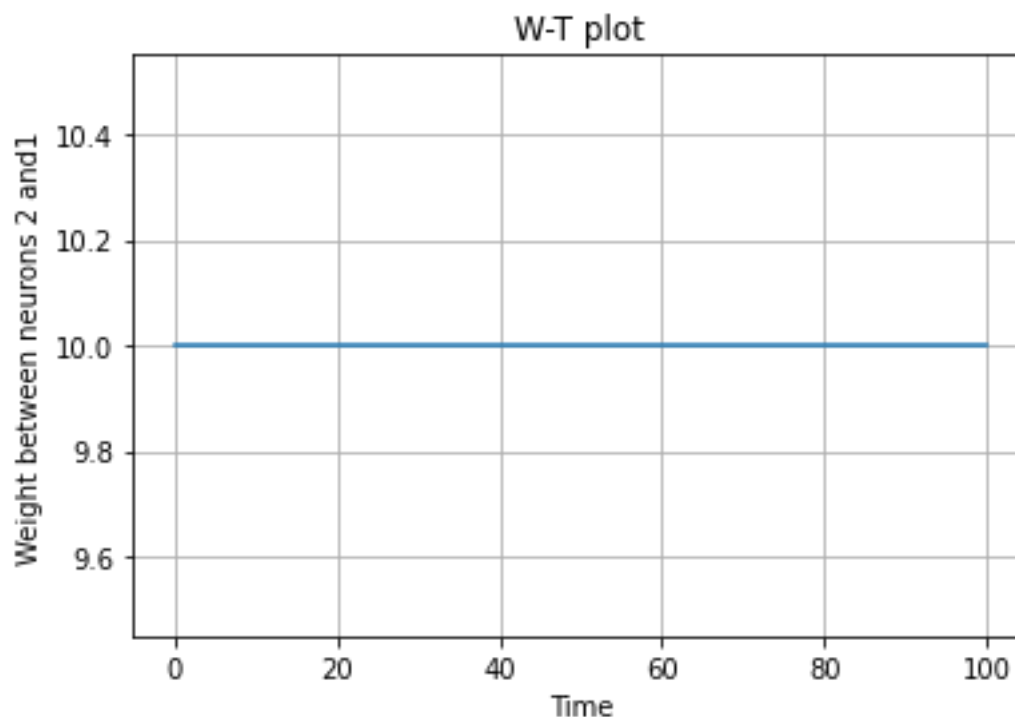


نمودارهای زیر، تغییرات وزن های نورون های ما در طول زمان را نشان می دهد (بین نورون های ۰ و ۱، ۰ و ۲، ۱ و ۲).









از آنجایی که مقادیر مقاومت، ظرفیت خازن، پتانسیل استراحت و اسپایک و... رندوم به مدل LIF داده می شوند، در این اجرای برنامه، آپدیت وزنی بین نورون های ۱ و ۲ اتفاق نیفتاده است. در مورد دیگر جفت ها، مشخص می شود که هر گاه نورون پیشین قبل از پسین اسپایک بزند، وزن بینشان افزایش و در غیر این صورت، کاهش پیدا می کند.

در قسمت بعدی، داده از از فایل اکسل خوانده شده و به گروه های فیچرها و نتیجه های آموزشی و تست افراز می شوند.

مدل خود را ابعاد موردنظر تشکیل داده، و داده ها را برای آن fit می کنیم. در این حالت خاص از اجرای برنامه، دقت پیشبینی برای داده های train، ۶۰ و برای داده های تست نیز 60 درصد می باشد.