



دانشگاه تهران  
دانشکده ریاضی و علوم کامپیوتر

پروژه سوم درس علوم اعصاب محاسباتی  
آشنایی با انواع روش‌های کدگذاری و روش‌های یادگیری بدون ناظر و تقویتی

محمد زمانی

شماره دانشجویی: ۶۱۰۳۹۹۱۳۵

# فهرست مطالب

۲	۱	مقدمه
۲	۲	کدگذاری
۲	۱.۲	Time To First Spike
۳	۲.۲	کدگذاری مقادیر عددی
۵	۳.۲	کدگذاری به کمک توزیع پواسون
۷	۳	یادگیری بدون ناظر STDP
۷	۱.۳	STDP
۷	۲.۳	ساختار کلی شبکه
۸	۳.۳	ورودی‌های شبکه
۸	۱.۳.۳	جریان ثابت با نویز
۸	۲.۳.۳	ورودی تصادفی با توزیع پواسون
۹	۴.۳	تغییرات وزن سیناپسی
۹	۱.۴.۳	تغییرات وزن سیناپسی با جریان ثابت نویز دار
۱۰	۲.۴.۳	تغییرات وزن سیناپسی با ورودی تصادفی پواسون
۱۱	۵.۳	شباهت کسینوسی
۱۱	۱.۵.۳	شباهت کسینوسی با جریان ثابت نویز دار
۱۲	۲.۵.۳	شباهت کسینوسی با ورودی تصادفی پواسون
۱۲	۶.۳	تاثیر نوروں بدون فعالیت
۱۲	۱.۶.۳	تاثیر نوروں بدون فعالیت با جریان ورودی ثابت نویز دار
۱۳	۲.۶.۳	تاثیر نوروں بدون فعالیت با ورودی تصادفی پواسون
۱۳	۷.۳	بررسی پارامترهای مهم
۱۳	۱.۷.۳	$a_+$ و $a_-$
۱۳	۲.۷.۳	محدودیت‌های وزنی
۱۴	۴	یادگیری تقویتی RSTDP
۱۴	۱.۴	RSTDP
۱۴	۲.۴	ساختار کلی شبکه
۱۵	۳.۴	ورودی‌های شبکه
۱۵	۱.۳.۴	جریان ثابت با نویز
۱۵	۲.۳.۴	ورودی تصادفی با توزیع پواسون
۱۶	۴.۴	تغییرات وزن سیناپسی
۱۶	۱.۴.۴	تغییرات وزن سیناپسی با جریان ثابت نویز دار
۱۷	۲.۴.۴	تغییرات وزن سیناپسی با ورودی تصادفی پواسون
۱۸	۵.۴	شباهت کسینوسی
۱۸	۱.۵.۴	شباهت کسینوسی با جریان ثابت نویز دار
۱۹	۲.۵.۴	شباهت کسینوسی با ورودی تصادفی پواسون
۱۹	۶.۴	تاثیر نوروں بدون فعالیت
۱۹	۱.۶.۴	تاثیر نوروں بدون فعالیت با جریان ورودی ثابت نویز دار
۲۰	۲.۶.۴	تاثیر نوروں بدون فعالیت با ورودی تصادفی پواسون

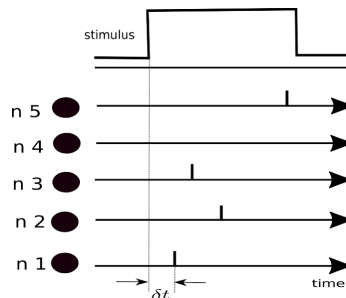
## ۱ مقدمه

تا به اینجای کار ما انواع مدل‌های نورونی را مورد بررسی قرار دادیم و سپس جمعیت‌های نورونی و ارتباطات میان آن‌ها را شبیه سازی کردیم و رفتار آن‌ها را در شرایط مختلف تحلیل کردیم. حال قصد داریم که فرآیندهای کدگذاری در هنگام ورودی دادن به شبکه را شبیه سازی کنیم و سپس می‌خواهیم ببینیم که چگونه شبکه ما یاد خواهد گرفت و آن را شبیه سازی خواهیم کرد و تحلیل می‌کنیم.

## ۲ کدگذاری

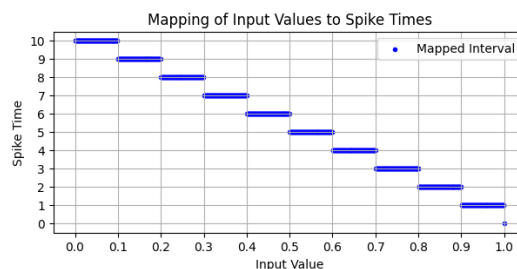
### ۱.۲ Time To First Spike

در روش کدگذاری TTFS ما به دنبال زمانی هستیم که نورون اولین ضربه خود را می‌زند، به عبارت دیگر از لحظه‌ای که ورودی به نورون داده می‌شود تا زمانی که نورون ضربه می‌زند را به عنوان چیزی که اطلاعات در آن وجود دارد ذخیره و کد می‌کنیم. که شماتیک کلی آن را می‌توانید در شکل ۱ مشاهده کنید.



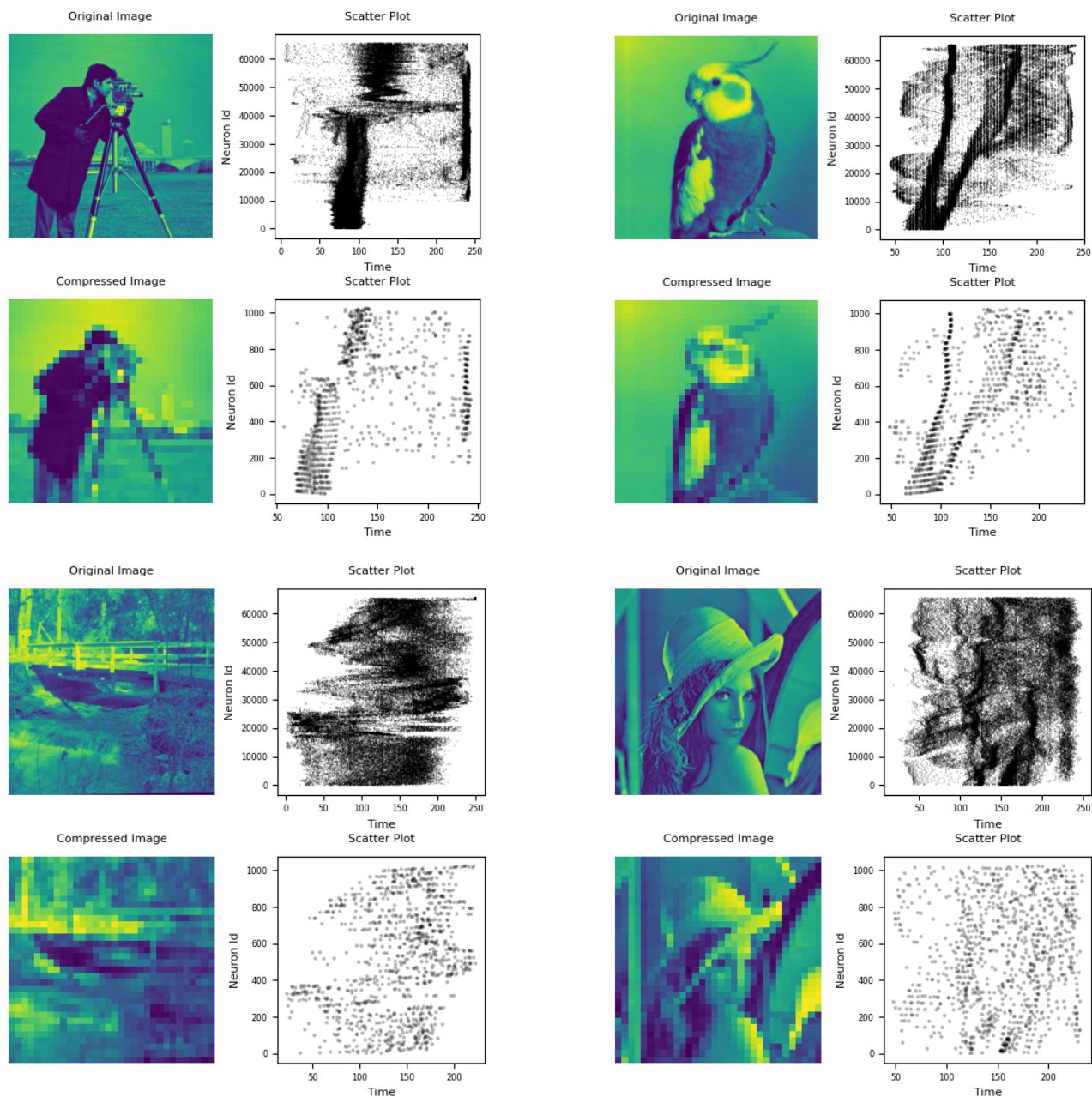
شکل ۱: Time To First Spike

حال برای پیاده سازی این روش ما ورودی‌ها را نرمال می‌کنیم در یک بازه مثل صفر تا یک و یک پارامتر  $T$  داریم، این بازه صفر تا یک را به  $T$  قسمت مساوی تقسیم خواهیم کرد و با توجه به اینکه ورودی در بازه چندم قرار می‌گیرد در آن زمان ضربه خواهد زد، شکل ۲.



شکل ۲: نحوه شبیه‌سازی Time To First Spike

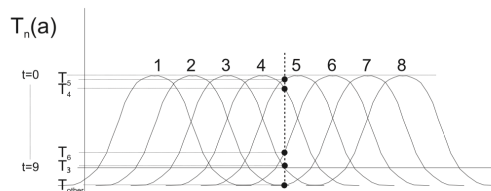
حال با استفاده از کدگذاری پیاده شده چهار عکس متفاوت را هم به صورت اورجینال هم به صورت فشرده شده کدگذاری می‌کنیم و می‌توانید نتایج آن و raster plot را در شکل ۳ مشاهده کنید.



شکل ۳: raster plot برای ضربه‌های تولید شده با استفاده از روش کدگذاری TTFS

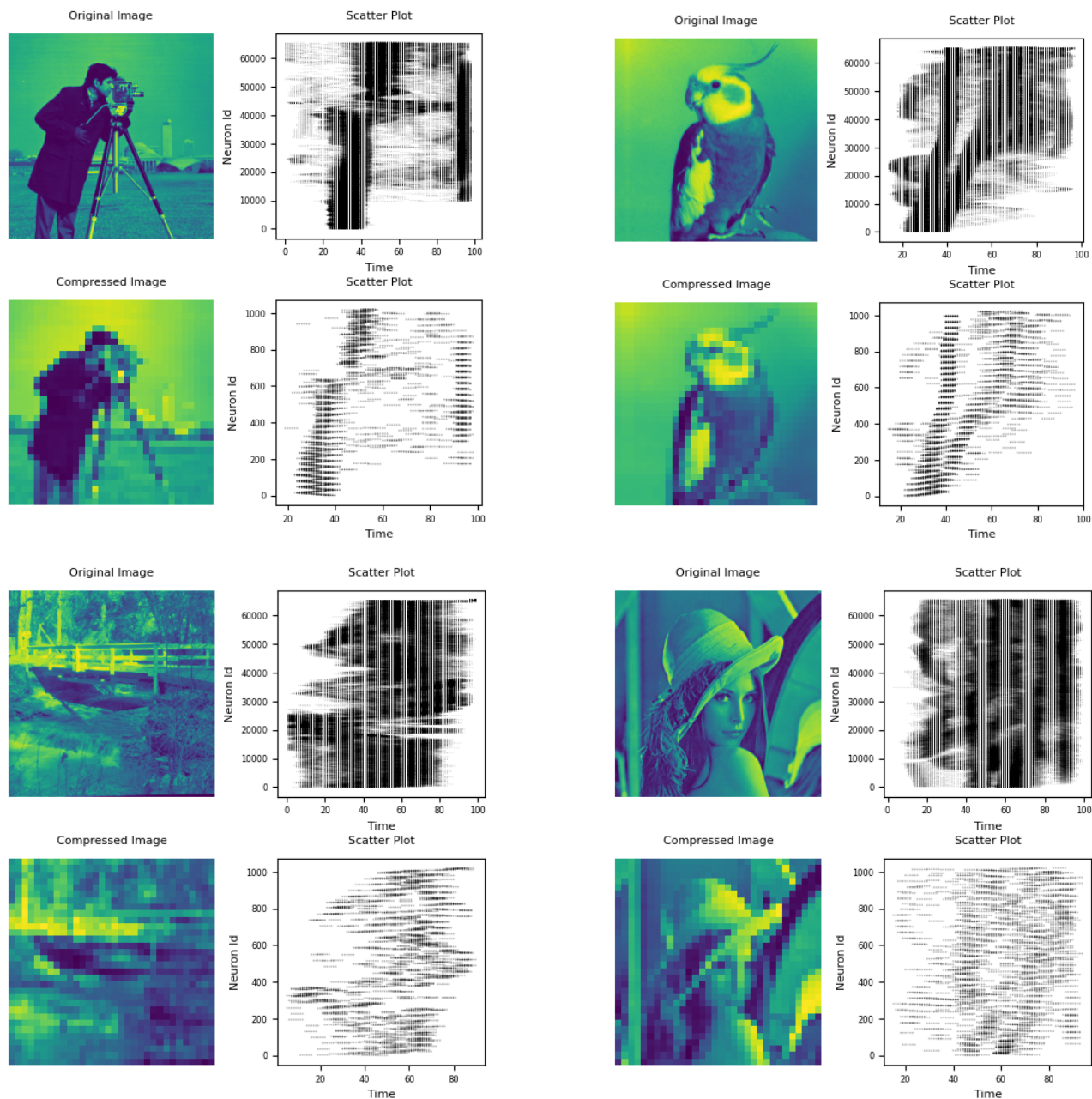
## ۲.۲ کدگذاری مقادیر عددی

در این روش بازه ۰ تا ۲۵۵ را که ورودی ما در این بازه قرار دارد به  $T$  قسمت مساوی تقسیم می‌کنیم که این  $T$  یک پارامتر است، سپس توزیع‌های نرمالی با میانگین‌هایی که در این تقسیم بندی بدست آوردیم با یک انحراف معیار ثابت که پارامتر ما است رسم می‌کنیم، حال برای هر ورودی ما نقاط تلاقی آن نقطه با این توزیع‌ها را به عنوان زمان ضربه نوروں در نظر می‌گیریم، به این صورت برای هر ورودی می‌توانیم یک دنباله‌ای از ضربه‌ها به صورت کد شده ارائه دهیم، شکل ۴.



شکل ۴: کدگذاری مقادیر عددی

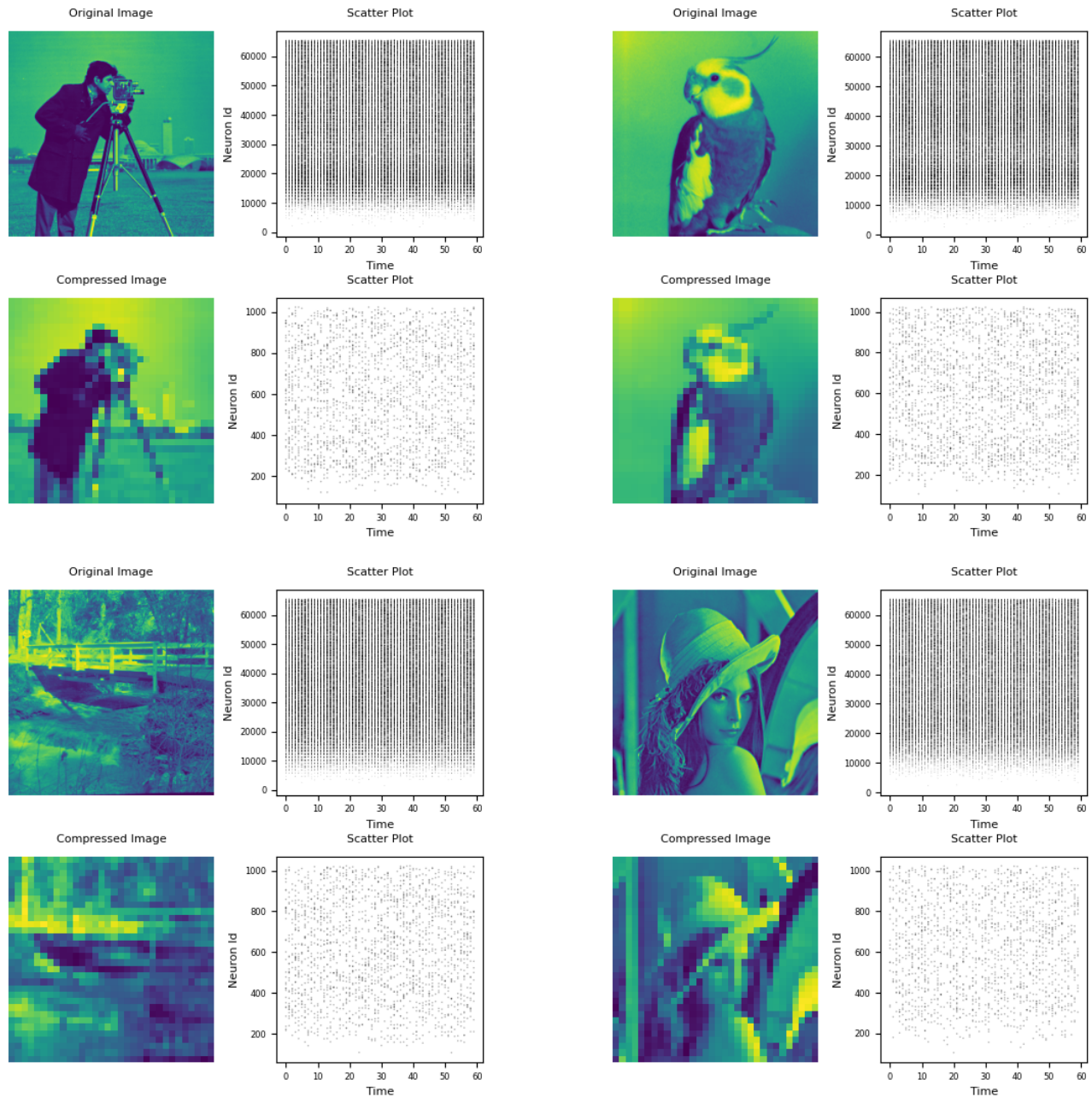
حال با توجه به توضیحات داده شده مانند قسمت قبل ورودی‌ها را در مدت زمان  $T = 100\text{ms}$  ورودی می‌دهیم و raster plot حاصل را می‌توانید در شکل ۵ مشاهده کنید.



شکل ۵: raster plot برای ضربه‌های تولید شده با استفاده از روش کدگذاری مقادیر عددی

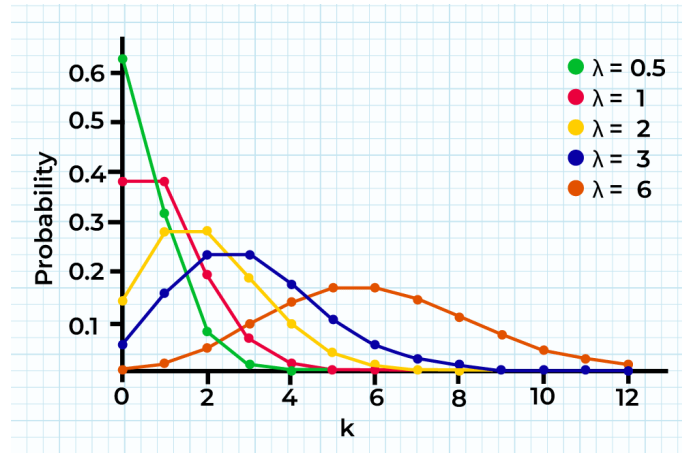
## ۳.۲ کدگذاری به کمک توزیع پواسون

در این روش از کدگذاری زمان میان ضربه‌های تولید شده را با استفاده از توزیع پواسون بدست می‌آوریم، شکل توزیع پواسون را می‌توانید در شکل ۹ مشاهده کنید، همانطور که می‌بینید در این توزیع یک پارامتر به نام  $\lambda$  نیز وجود دارد که ما آن را در آزمایشمان ۰/۰۹ قرار دادیم. در نهایت مانند قسمت‌های قبل ورودی را در مدت زمان  $T = 60\text{ms}$  ورودی دادیم و نتیجه را می‌توانید در شکل ۶ مشاهده کنید.



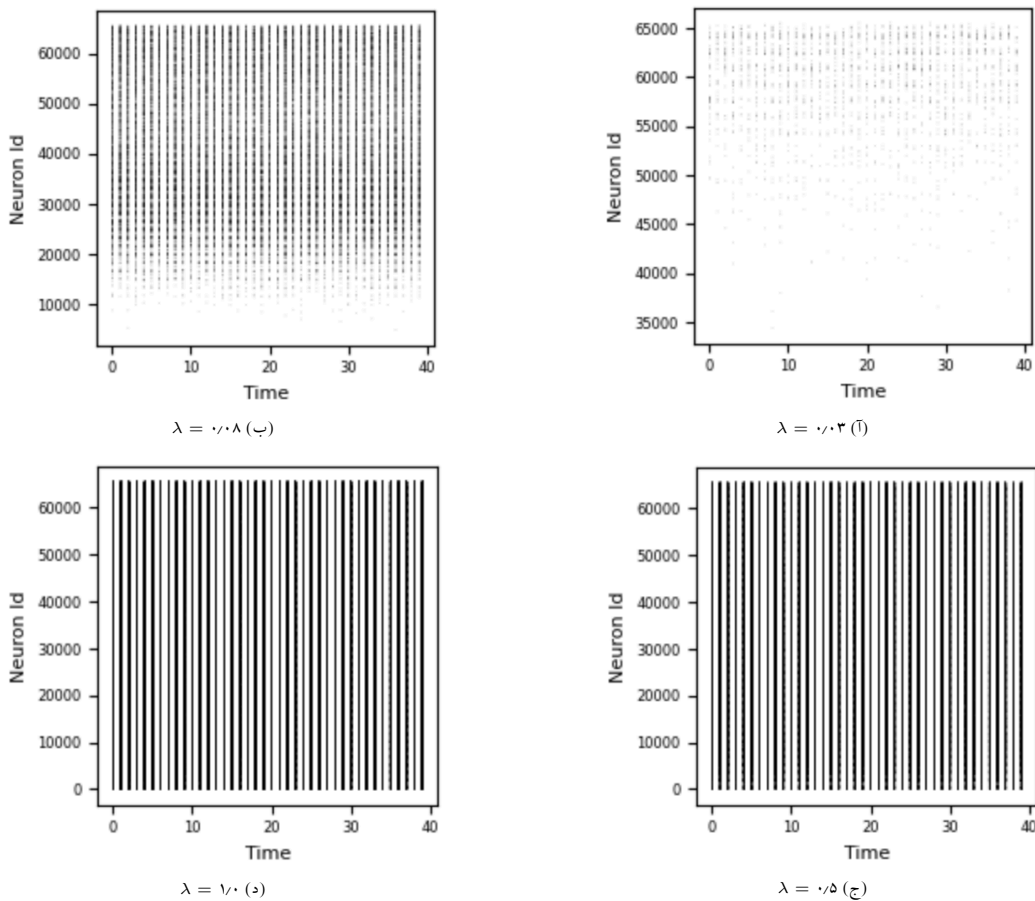
شکل ۶: raster plot برای ضربه‌های تولید شده با استفاده از روش کدگذاری به کمک توزیع پواسون





شکل ۷: توزیع پواسون

در اینجا در توزیع پواسون پارامتر  $\lambda$  در واقع نشان دهنده نسبت یا میزان ضربه‌ها می‌باشد که هر چه آن را عدد بزرگتری قرار دهیم تعداد ضربه‌هایی که نورون می‌زند بیشتر خواهد شد زیرا با افزایش مقدار  $\lambda$  مقدار شانس ضربه را افزایش می‌دهیم، به عنوان مثال برای تصویر پرنده این آزمایش را با مقادیر مختلف  $\lambda$  تست می‌کنیم و نتیجه آن را می‌تواند در شکل ۸ مشاهده کنید.



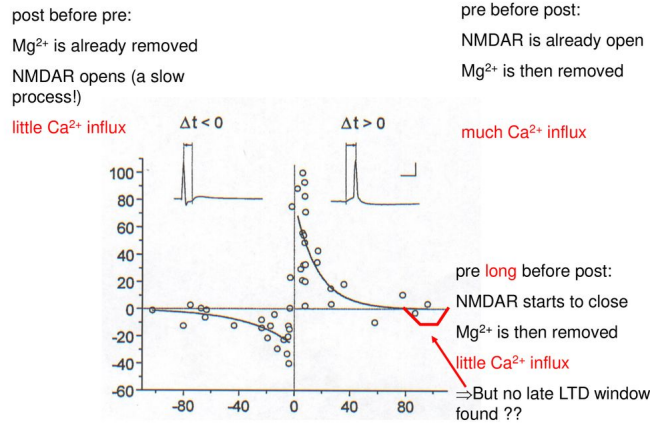
شکل ۸: نقش پارامتر  $\lambda$  در میزان ضربه‌های نورون

## ۳ یادگیری بدون ناظر STDP

### ۱.۳ STDP

اگر که ما دو الکتروود داشته باشیم به طوری که الکتروودها درون غشای نورونی قرار گیرند و از یکی از آنها برای تحریک نورون استفاده کنیم و از دیگری برای اندازه گیری Neuron Response استفاده کنیم، می‌توانیم مشاهده کنیم که اگر که نورون قبلی فعال شود و سپس با یک فاصله زمانی کوتاهی نورون بعدی متصل به قبلی فعال شود قدرت سیناپسی میان این دو نورون بیشتر می‌شود که این نشان می‌دهد تغییر وزن سیناپسی می‌تواند تابعی از  $t_j^f - t_i^f$  باشد.

### STDP



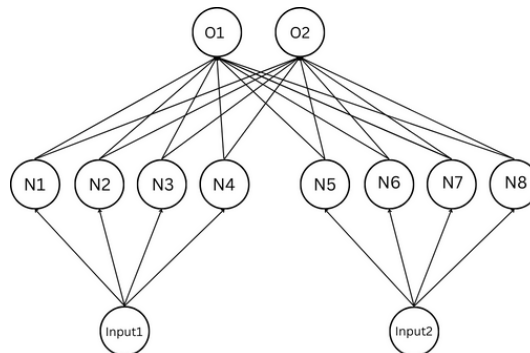
شکل ۹: فرآیند یادگیری STDP، به فرآیند افزایش وزن در این قانون یادگیری LTP و به فرآیند کاهش وزن LTD می‌گویند.

به طور کلی می‌توانیم تغییرات وزن بر حسب زمان را با این معادله شبیه سازی کنیم:

$$\frac{dw_{ij}}{dt} = -A_-(w_{ij})y_i(t)\Sigma_f\delta(t - t_j^f) + A_+(w_{ij})x_j(t)\Sigma_f\delta(t - t_i^f)$$

### ۲.۳ ساختار کلی شبکه

برای انجام آزمایشات شبکه ما به طور کلی متشکل از سه جمعیت نورونی خواهد بود که دو جمعیت نورونی متشکل از چهار نورون داریم که به یک جمعیت نورونی متشکل از دو نورون متصل هستند، در واقع این جمعیت دو نورونی نقش خروجی‌های ما را دارند، ورودی‌ها نیز به دو جمعیت نورونی چهار نورونی داده می‌شود که این ساختار را می‌تواند در شکل ۱۰ مشاهده کنید.



شکل ۱۰: ساختار شبکه ساخته شده

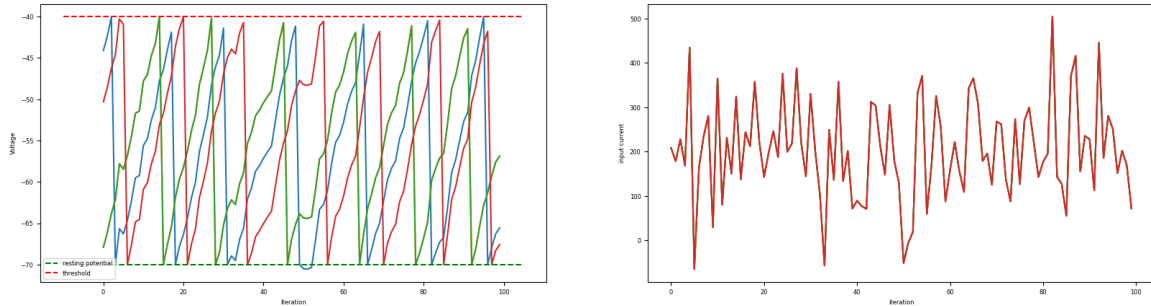


### ۳.۳ ورودی‌های شبکه

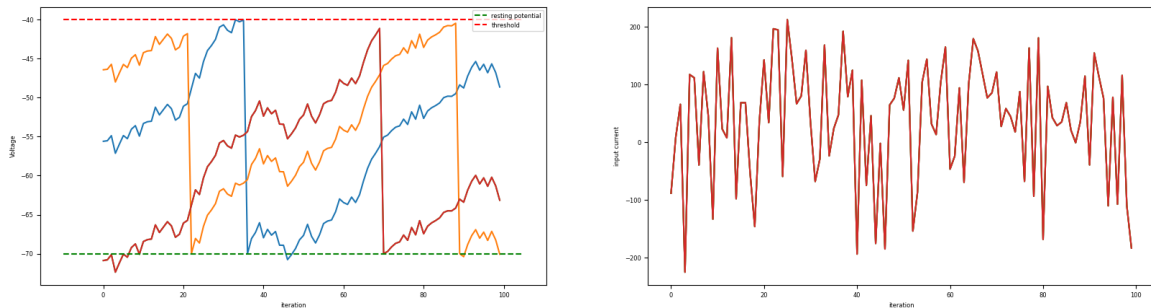
#### ۱.۳.۳ جریان ثابت با نویز

در این حالت به هر یک از جمعیت‌های نورونی چهار نورونی یک جریان مجزا وارد می‌کنیم، این جریان یک مقدار ثابت خواهد داشت که به آن مقدار ثابت با توزیع نرمال نویز وارد می‌کنیم.

به عنوان مثال به جمعیت نورونی اول جریان ثابت ۲۰۰ با نویز وارد می‌کنیم و به جمعیت نورونی دوم جریان ثابت ۵۰ با نویز وارد می‌کنیم که می‌توانید نمودار جریان‌های ورودی و فعالیت شبکه را در شکل ۱۱ مشاهده کنید.

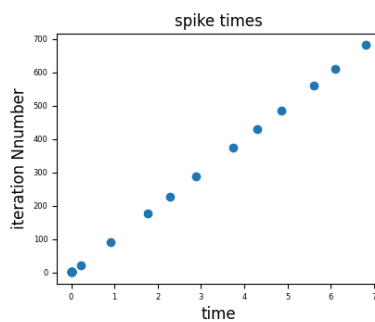


(آ) جمعیت نورونی اول

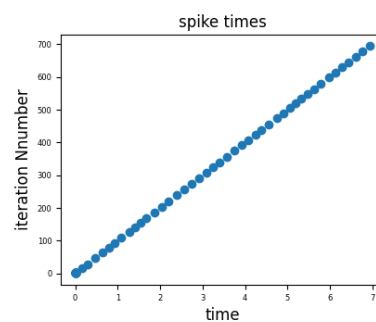


(ب) جمعیت نورونی دوم

شکل ۱۱: نمودار جریان ورودی و تغییرات پتانسیل نورون‌های جمعیت‌های نورونی



(ب) جمعیت نورونی دوم



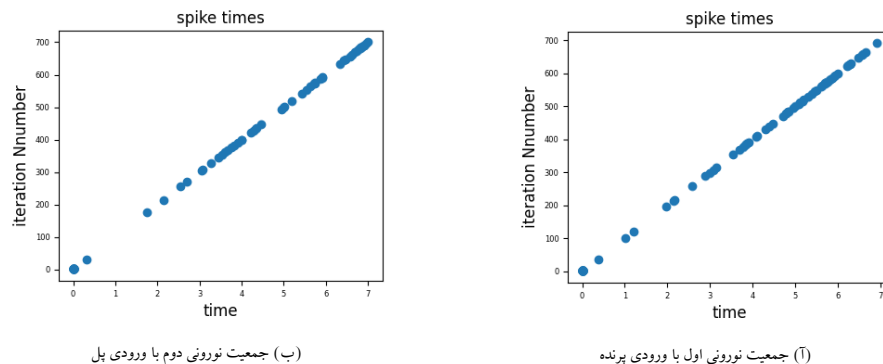
(آ) جمعیت نورونی اول

شکل ۱۲: نمودار زمان فعالیت جمعیت‌های نورونی

### ۲.۳.۳ ورودی تصادفی با توزیع پواسون

در این قسمت از کدگذاری پواسون که در بخش کدگذاری پواسون توضیح دادیم استفاده می‌کنیم، یک عکس را انتخاب می‌کنیم و آن را کد می‌کنیم و ضربه‌های تولید شده را به عنوان ورودی به شبکه می‌دهیم و جمعیت نورونی را مجبور به آن رفتار می‌کنیم.

در این بخش از آزمایش ما دو عکس پرنده و پل انتخاب شده را به عنوان ورودی به دو جمعیت نورونی می‌دهیم به این صورت که به جمعیت اول ضربه‌های تولید شده توسط عکس پرنده و به جمعیت دوم ضربه‌های تولید شده توسط عکس پل را می‌دهیم، ضربه‌های این دو جمعیت نورونی را می‌توانید در شکل ۱۳ مشاهده کنید.

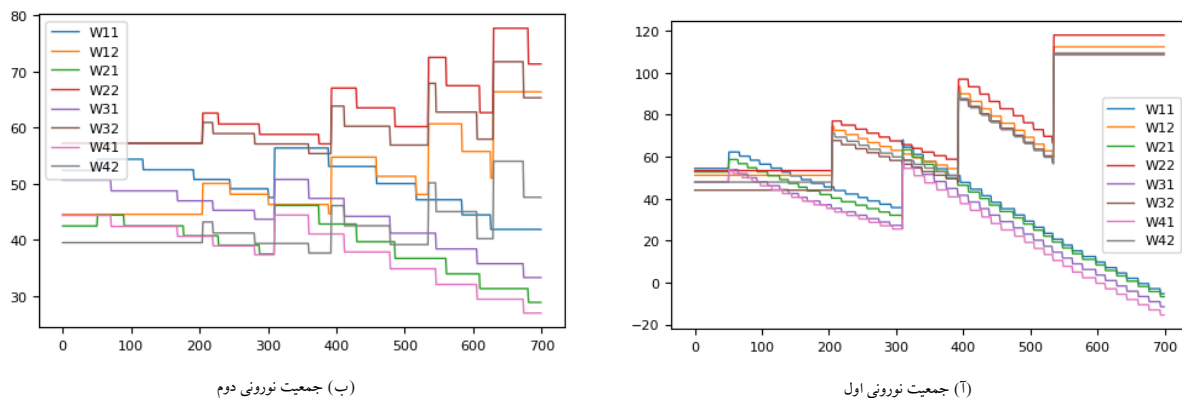


شکل ۱۳: نمودار زمان فعالیت جمعیت‌های نورونی

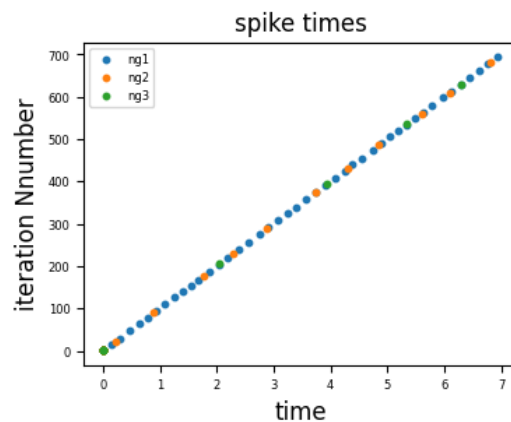
## ۴.۳ تغییرات وزن سیناپسی

اگر نورون‌های هر یک از جمعیت‌های نورونی را از یک تا  $n$  شماره گذاری کنیم، نشان دهنده وزن میان نورون  $i$ ام از جمعیت قبلی به نورون  $j$ ام جمعیت بعدی است. در این بخش می‌خواهیم تغییرات وزن سیناپسی را نشان دهیم.

### ۱.۴.۳ تغییرات وزن سیناپسی با جریان ثابت نویز دار



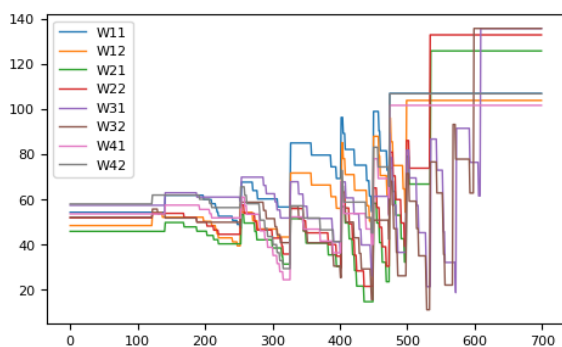
شکل ۱۴: نمودار تغییرات وزن‌های سیناپسی جمعیت‌های نورونی



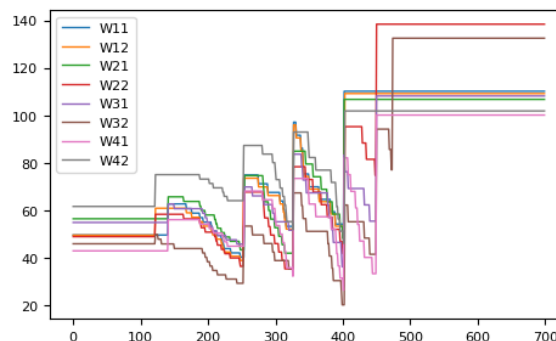
شکل ۱۵: زمان ضربه هر سه جمعیت نوروئی موجود در شبکه

همانطور که در شکل ۱۵ مشاهده می‌کنید فواصل زمانی ضربه بین نوروئ‌های جمعیت اول با ضربه‌های جمعیت خروجی نسبت به جمعیت دوم و جمعیت خروجی کمتر می‌باشد و در نتیجه این مورد طبق تعریف قانون یادگیری  $STDP$  باعث می‌شود که افزایش وزن در جمعیت اول بیشتر باشد و همچنین شدت کاهش وزن نیز بیشتر باشد.

## ۲.۴.۳ تغییرات وزن سیناپسی با ورودی تصادفی پوآسون

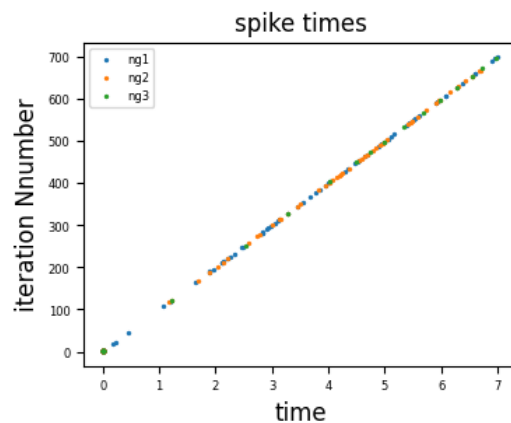


(ب) جمعیت نوروئی دوم با ورودی پل



(آ) جمعیت نوروئی اول با ورودی پرنده

شکل ۱۶: نمودار تغییرات وزن‌های سیناپسی جمعیت‌های نوروئی



شکل ۱۷: زمان ضربه هر سه جمعیت نوروئی موجود در شبکه

همانطور که مشاهده می‌کنید در این حالت که هر دو جمعیت نورونی ورودی ما ضربه‌هایی قبل از ضربه‌های جمعیت خروجی داشتند باعث شده است که وزن‌های سیناپسی افزایش پیدا کنند، البته نکته مهمی که وجود دارد این است که ما برای کنترل افزایش وزن‌ها از *hard bound* نیز استفاده کرده‌ایم که باعث می‌شود به وزن‌ها اجازه صعود بدون حد ندهد و این رفتار را در نمودارهای ۱۶ می‌توانید مشاهده کنید.

## ۵.۳ شباهت کسینوسی

The cosine of two non-zero vectors can be derived by using the [Euclidean dot product](#) formula:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \|\mathbf{B}\| \cos \theta$$

Given two  $n$ -dimensional [vectors](#) of attributes,  $\mathbf{A}$  and  $\mathbf{B}$ , the cosine similarity,  $\cos(\theta)$ , is represented using a [dot product](#) and [magnitude](#) as

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}},$$

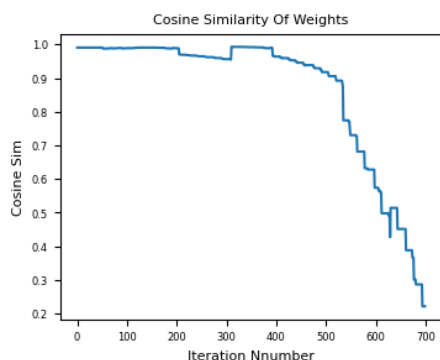
where  $A_i$  and  $B_i$  are the  $i$ th [components](#) of vectors  $\mathbf{A}$  and  $\mathbf{B}$ , respectively.

شکل ۱۸: تعریف شباهت کسینوسی

همانطور که در شکل ۱۰ نشان داده شده است، به هر نورون خروجی هشت نورون متصل است که این به معنای وجود هشت وزن سیناپسی می‌باشد، حال برای هر مرحله از شبیه سازی برای هر نورون خروجی یک بردار هشت تایی از وزن‌ها در آن لحظه خواهیم داشت که می‌توانیم برای این بردارها شباهت کسینوسی را اندازه گیری کنیم.

## ۱۰.۵.۳ شباهت کسینوسی با جریان ثابت نويز دار

همانطور که بالاتر توضیح دادیم در این قسمت شباهت کسینوسی را در هر مرحله از شبیه سازی محاسبه می‌کنیم که نتیجه آن را در نمودار ۱۹ می‌توانید مشاهده کنید.



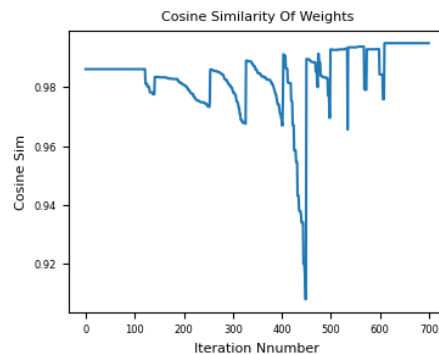
شکل ۱۹: نمودار شباهت کسینوسی وزن‌های دو نورون خروجی

اگر که وظیفه ما یک *classification* باشد در آن زمان هر نورون خروجی ما بازنمایی یک کلاس خواهد بود، بنابراین زمانی این مدل به خوبی عمل می‌کند که شباهت میان این نورون‌های خروجی به کمینه خود برسند چون کاملاً در این صورت الگوهای متفاوتی را آموخته‌اند، در اینجا هم همانطور که مشاهده می‌کنید به مرور زمان وقتی شبیه سازی را انجام می‌دهیم شباهت میان وزن‌ها کمتر می‌شود که این نشان دهنده یادگیری الگوهای متفاوت توسط نورون‌های خروجی

می‌باشد که این مطابق انتظار ما رفتار خوبی است چونکه ما دوجمعیت با ورودی‌های کاملاً متفاوت داشتیم و انتظار داشتیم که بتوانیم تفاوت آن‌ها را در نورون‌های خارجی نیز مشاهده کنیم.

### ۲.۵.۳ شباهت کسینوسی با ورودی تصادفی پوآسون

همانطور که ابتدای این بخش توضیح دادیم در این قسمت شباهت کسینوسی را در هر مرحله از شبیه سازی محاسبه می‌کنیم که نتیجه آن را در نمودار ۲۰ می‌توانید مشاهده کنید.



شکل ۲۰: نمودار شباهت کسینوسی وزن‌های دو نورون خروجی

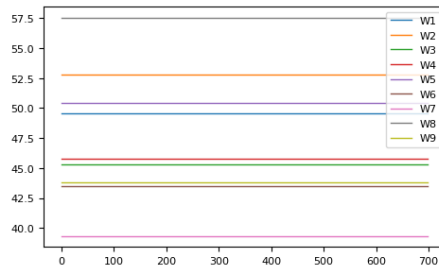
اگر که وظیفه ما یک *classification* باشد وقتی می‌توانیم به خوبی عمل کنیم که شباهت به میزان کمینه خود برسد و این البته در حالتی خوب است که الگوهای ورودی هم کاملاً متفاوت باشند، برای مثال در بخش ورودی با جریان ثابت نويز دار که الگوها تفاوت زیادی داشتند این شباهت خیلی کم بود و نورون‌های خروجی رفتارهای متفاوتی را آموختند اما در این بخش همانطور که در شکل ۲۰ مشاهده می‌کنید نورون‌های خروجی ما الگوهای خیلی متفاوتی را نیاموخته‌اند و تقریباً می‌توان گفت که الگوها با هم مشابه بوده‌اند اما با این حال می‌توان از آن‌ها برای جداسازی کلاس‌ها استفاده کرد.

### ۶.۳ تاثیر نورون بدون فعالیت

در این بخش یک جمعیت نورونی با یک نورون به دو جمعیت نورونی ورودی اضافه می‌کنیم که به آن ورودی نمی‌دهیم تا ضربه ای نزنند همچنین یک جمعیت نورونی نیز به خروجی اضافه می‌کنیم که در آن نیز با افزایش حد آستانه کاری می‌کنیم که آن نورون ضربه ای نزنند، بنابراین در این شبکه جدید نه نورون در لایه ورودی داریم که از سه جمعیت تشکیل شده‌اند و دو جمعیت در لایه خروجی داریم که در کل سه نورون دارند.

### ۱.۶.۳ تاثیر نورون بدون فعالیت با جریان ورودی ثابت نويزدار

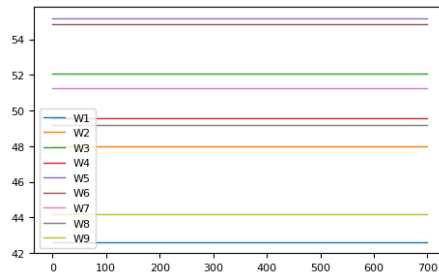
در این حالت از آنجایی که نورون خروجی‌ای که اضافه کردیم هیچ ضربه ای نمی‌زنند بنابراین وزن سیناپسی متصل به آن نورون نیز تغییری ندارد، ما نه نورون متصل به این نورون داریم که تغییرات وزن‌های سیناپسی را می‌توانید در شکل ۲۱ مشاهده کنید و همانطور که می‌بینید تغییری مشاهده نمی‌شود.



شکل ۲۱: تغییر وزن‌های سیناپسی نورون بدون فعالیت خروجی

### ۲.۶.۳ تاثیر نورون بدون فعالیت با ورودی تصادفی پوآسون

همانطور که در قسمت قبلی توضیح داده شد از آنجایی که هیچ ضربه ای این نورون خروجی اضافه شده نمی‌زند بنابراین طبق تعریف قانون یادگیری STDP هیچ تغییری در وزن‌های سیناپسی متصل به این نورون رخ نمی‌دهد و این موضوع را می‌توانید در شکل ۲۲ مشاهده کنید.



شکل ۲۲: تغییر وزن‌های سیناپسی نورون بدون فعالیت خروجی

## ۷.۳ بررسی پارامترهای مهم

از مهمترین پارامترهایی که در این یادگیری موثر هستند می‌توان به این پارامترها اشاره کرده:

### ۱.۷.۳ $a_+$ و $a_-$

این دو هر کدام ضرایبی برای افزایش و کاهش وزن می‌باشد که هر چه برای مثال  $a_+$  را افزایش دهیم سرعت افزایش وزن‌ها بیشتر می‌شود و اگر  $a_-$  را افزایش دهیم سرعت کاهش وزن افزایش می‌یابد. در آزمایشات انجام شده ما  $a_- = a_+ = 2$  قرار داده‌ایم.

### ۲.۷.۳ محدودیت‌های وزنی

در کل می‌توانیم از محدودیت‌های وزنی متفاوتی برای کنترل وزن‌ها استفاده کنیم که هر کدام از این روش‌ها نیز پارامترهای خاص خود را دارند، برای مثال در این آزمایشات ذکر شده از hard bound استفاده کردیم و بیشینه وزن را مقدار ۱۰۰ قرار دادیم اما می‌توان آن را تغییر داد و از پارامترهای متفاوت دیگری استفاده کرد، از آنجایی که این پارامتر مفهوم خاصی ندارد و فقط برای کنترل وزن‌ها می‌باشد آن را با پارامتر دیگری تست نمی‌کنیم.



## ۴ یادگیری تقویتی RSTDP

### ۱.۴ RSTDP

در یادگیری بدون ناظر STDP ما کنترلی بر روند یادگیری نداریم و ماهیت این نوع یادگیری به این صورت است که چیزی که تکرار می‌شود و فرکانس آن زیاد است تقویت می‌شود و یادگرفته می‌شود اما در یادگیری تقویتی ما به دنبال این هستیم که با دادن مجازات و پاداش بر روی روند یادگیری کنترل داشته باشیم و در انتهای یادگیری کارهایی که منجر به پاداش می‌شوند را یاد بگیریم و از طرفی از کارهایی که منجر به مجازات می‌شوند دوری کنیم.

به نظر می‌رسد درون مغز ما یادگیری تقویتی به کمک ترشح دوپامین<sup>۱</sup> در مغز انجام می‌شود، دوپامین یک Neuro-modulator است که دورن مغز وجود دارد و بر روی روند یادگیری مغز نقش دارد. مطالعات نشان می‌دهد که زمانی که ما کاری انجام می‌دهیم که برای ما پاداشی به همراه دارد غلظت دوپامین در مغز ما بیشتر می‌شود و در صورتی که کار ما باعث مجازات شود غلظت دوپامین کم می‌شود.

یکی از مسائل مهمی که در این نوع یادگیری با آن مواجه هستیم مسئله distal problem است، وقتی که ما کاری انجام می‌دهیم ممکن است پس مدتی مجازات یا پاداش آن را دریافت کنیم بنابراین تشخیص اینکه هر پاداش مربوط به چه واکنشی بوده است بسیار مهم است.

آزمایشات نشان می‌دهند که دوپامین به دو صورت می‌تواند بر روی STDP تاثیر بگذارد:

• changing STDP's window

• changing STDP's polarity

اگر به هر کدام از این موارد یک متغیر نسبت دهیم آنگاه می‌توانیم دینامیک آن‌ها را به این صورت بیان کنیم:

$$\frac{dc}{dt} = -\frac{c}{\tau_c} + STDP(I)\sigma(t - \frac{t_{pre}}{t_{post}})$$

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

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

که  $s$  نشان دهنده وزن سیناپسی است،  $c$  نیز نشان دهنده همان فعالیت آهسته‌ای است که وقتی نورون قبلی فعال می‌شود و این فعالیت به نورون بعدی منتقل می‌شود، که در حین انجام این انتقال یک سری مکانیزم‌ها و path way هایی فعال می‌شوند که به صورت لحظه‌ای نیستند و آهسته به وجود می‌آیند و آهسته از بین می‌روند، بنابراین  $c$  به طور کلی فعالیت آنزیم‌هایی که در plasticity مهم هستند را نشان می‌دهد. در نهایت  $d$  نیز غلظت دوپامین را نشان می‌دهد.

### ۲.۴ ساختار کلی شبکه

در این بخش نیز شبکه ما همانطور که در شکل ۱۰ نشان دادیم خواهد بود از سه جمعیت نورونی تشکیل شده است.

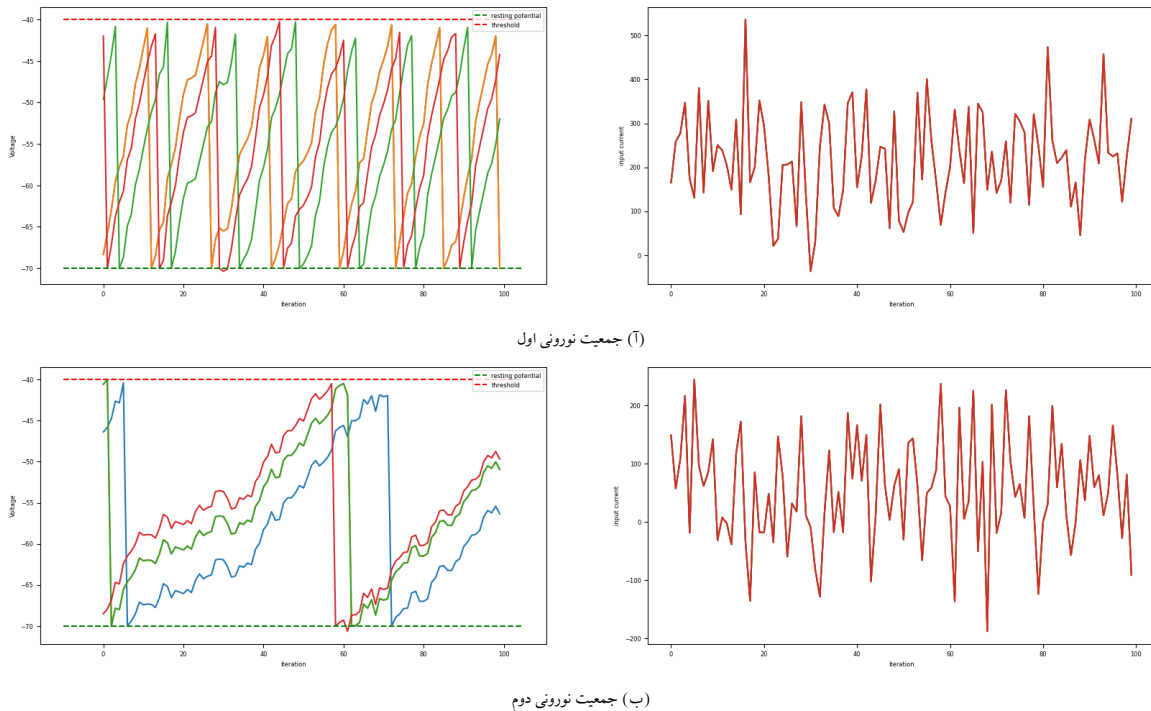
<sup>1</sup>Dopamine

## ۳.۴ ورودی‌های شبکه

### ۱.۳.۴ جریان ثابت با نویز

در این حالت به هر یک از جمعیت‌های نورونی چهار نورونی یک جریان مجزا وارد می‌کنیم، این جریان یک مقدار ثابت خواهد داشت که به آن مقدار ثابت با توزیع نرمال نویز وارد می‌کنیم.

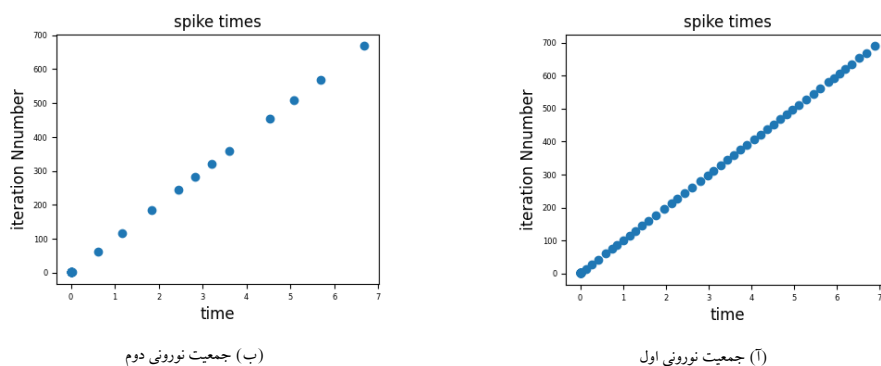
به عنوان مثال به جمعیت نورونی اول جریان ثابت ۲۰۰ با نویز وارد می‌کنیم و به جمعیت نورونی دوم جریان ثابت ۵۰ با نویز وارد می‌کنیم که می‌توانید نمودار جریان‌های ورودی و فعالیت شبکه را در شکل ۲۳ مشاهده کنید.



(آ) جمعیت نورونی اول

(ب) جمعیت نورونی دوم

شکل ۲۳: نمودار جریان ورودی و تغییرات پتانسیل نورون‌های جمعیت‌های نورونی



(ب) جمعیت نورونی دوم

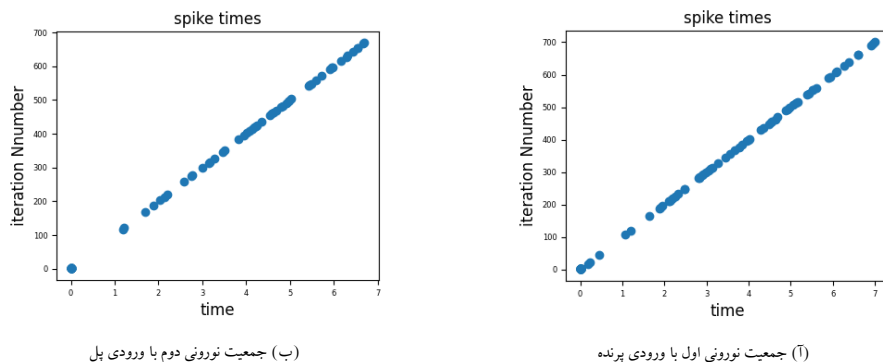
(آ) جمعیت نورونی اول

شکل ۲۴: نمودار زمان فعالیت جمعیت‌های نورونی

## ۲.۳.۴ ورودی تصادفی با توزیع پواسون

در این قسمت از کدگذاری پواسون که در بخش کدگذاری پواسون توضیح دادیم استفاده می‌کنیم، یک عکس را انتخاب می‌کنیم و آن را کد می‌کنیم و ضربه‌های تولید شده را به عنوان ورودی به شبکه می‌دهیم و جمعیت نورونی را مجبور به آن رفتار می‌کنیم.

در این بخش از آزمایش ما دو عکس پرنده و پل انتخاب شده را به عنوان ورودی به دو جمعیت نورونی می‌دهیم به این صورت که به جمعیت اول ضربه‌های تولید شده توسط عکس پرنده و به جمعیت دوم ضربه‌های تولید شده توسط عکس پل را می‌دهیم، ضربه‌های این دو جمعیت نورونی را می‌توانید در شکل ۲۵ مشاهده کنید.

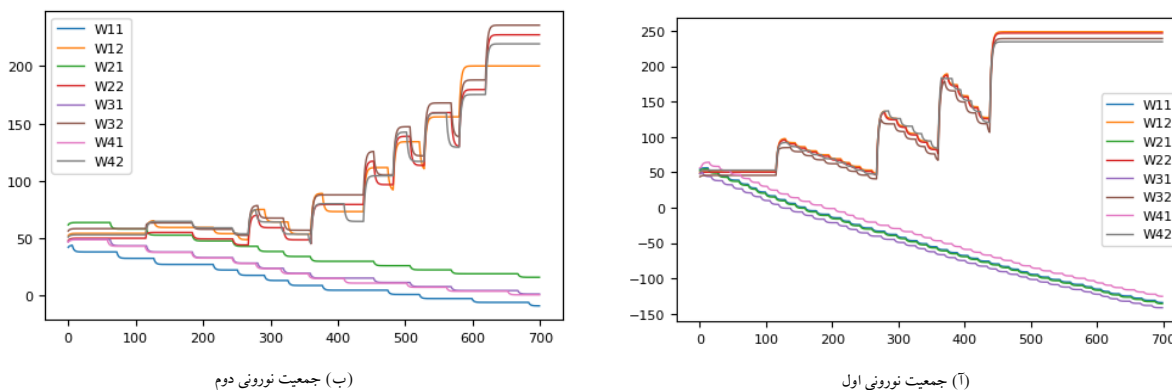


شکل ۲۵: نمودار زمان فعالیت جمعیت‌های نورونی

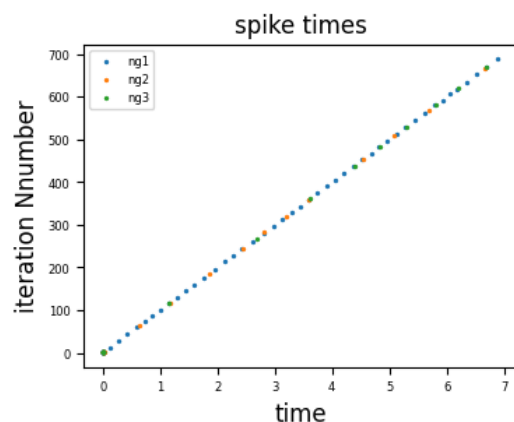
## ۴.۴ تغییرات وزن سیناپسی

اگر نورون‌های هر یک از جمعیت‌های نورونی را از یک تا  $n$  شماره گذاری کنیم، نشان دهنده وزن میان نورون  $i$ ام از جمعیت قبلی به نورون  $j$ ام جمعیت بعدی است. در این بخش می‌خواهیم تغییرات وزن سیناپسی را نشان دهیم.

### ۱.۴.۴ تغییرات وزن سیناپسی با جریان ثابت نویز دار



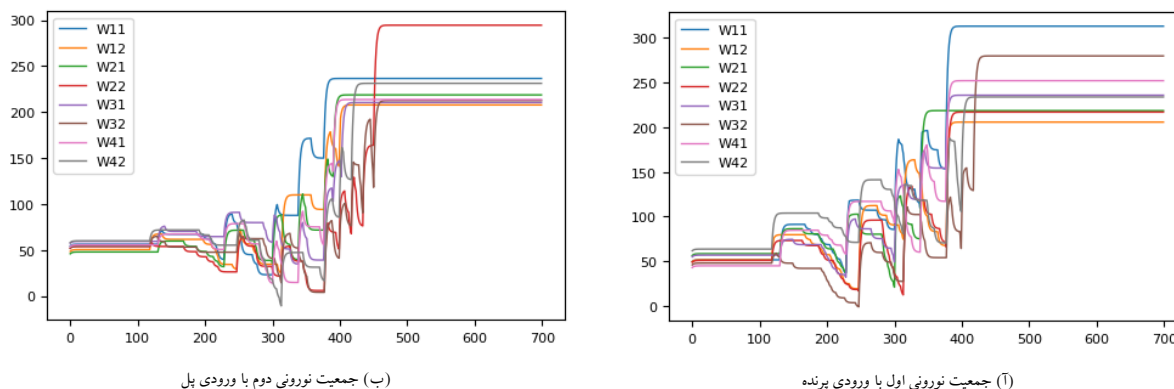
شکل ۲۶: نمودار تغییرات وزن‌های سیناپسی جمعیت‌های نورونی



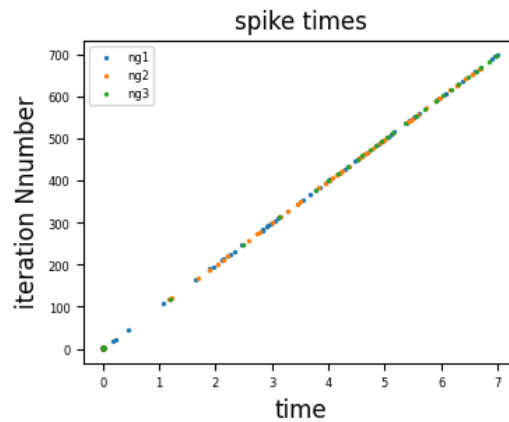
شکل ۲۷: زمان ضربه هر سه جمعیت نورونی موجود در شبکه

همانطور که در نمودارهای ۲۶ نشان داده شده است در هر دو جمعیت نورونی وزن‌های متصل به خروجی اول به مرور زمان کاهش یافته اند و وزن‌های خروجی به جمعیت دوم افزایش که این رفتار به این دلیل است که دو جمعیت ما کاملاً الگوهای مختلفی را به عنوان ورودی می‌گیرند و طبقاً انتظار داریم که فعالیت یک از خروجی‌ها بیشتر و دیگری کمتر باشد تا شباهت به کمینه خود برسد و به عبارت دیگر در این نوع یادگیری ما برای یکی از الگوها پاداش می‌دهیم و برای دیگری مجازات می‌کنیم، در نهایت این امر باعث می‌شود که یکی از خروجی‌ها که نشان دهنده فعالیت با پاداش است تقویت شود و دیگری تضعیف می‌شود.

## ۲۰.۴.۴ تغییرات وزن سیناپسی با ورودی تصادفی پوآسون



شکل ۲۸: نمودار تغییرات وزن‌های سیناپسی جمعیت‌های نورونی



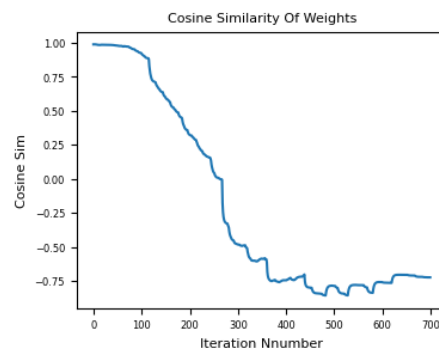
شکل ۲۹: زمان ضربه هر سه جمعیت نورونی موجود در شبکه

در این آزمایش همانطور که در نمودارهای ۲۸ مشاهده می‌کنید از آنجایی که هر دو الگو ورودی مشابه هم بودند و از طرفی ما به هر دو ورودی پاداش دادیم و غلظت دوپامین را افزایش دادیم باعث شد که همه وزن‌ها افزایش داشته باشند و خروجی‌های ما الگوهای متفاوتی را یاد نمی‌گیرند، می‌توانیم این موضوع را در شباهت کسینوسی نیز مشاهده کنیم.

## ۵.۴ شباهت کسینوسی

### ۱.۵.۴ شباهت کسینوسی با جریان ثابت نویز دار

همانطور که بالاتر توضیح دادیم در این قسمت شباهت کسینوسی را در هر مرحله از شبیه سازی محاسبه می‌کنیم که نتیجه آن را در نمودار ۳۰ می‌توانید مشاهده کنید.

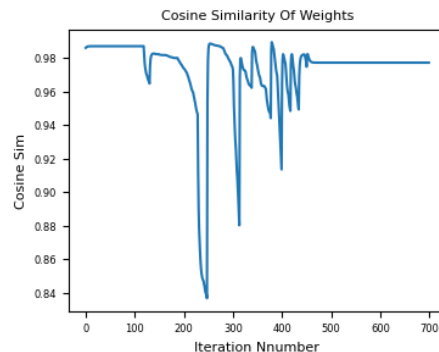


شکل ۳۰: نمودار شباهت کسینوسی وزن‌های دو نورون خروجی

اگر که وظیفه ما یک *classification* باشد در آن زمان هر نورون خروجی ما بازنمایی یک کلاس خواهد بود، بنابراین زمانی این مدل به خوبی عمل می‌کند که شباهت میان این نورون‌های خروجی به کمینه خود برسند چون کاملاً در این صورت الگوهای متفاوتی را آموخته‌اند، در اینجا هم همانطور که مشاهده می‌کنید به مرور زمان وقتی شبیه سازی را انجام می‌دهیم شباهت میان وزن‌ها کمتر می‌شود که این نشان دهنده یادگیری الگوهای متفاوت توسط نورون‌های خروجی می‌باشد که این مطابق انتظار ما رفتار خوبی است چونکه ما دوجمعیت با ورودی‌های کاملاً متفاوت داشتیم و انتظار داشتیم که بتوانیم تفاوت آن‌ها را در نورون‌های خارجی نیز مشاهده کنیم. در این قسمت با توجه به اینکه به یک الگو پاداش و دیگری مجازات دادیم و کنترل بر یادگیری داشتیم نسبت به یادگیری STDP کاهش بیشتری در شباهت کسینوسی مشاهده می‌کنیم.

## ۲.۵.۴ شباهت کسینوسی با ورودی تصادفی پوآسون

همانطور که ابتدای این بخش توضیح دادیم در این قسمت شباهات کسینوسی را در هر مرحله از شبیه سازی محاسبه می‌کنیم که نتیجه آن را در نمودار ۳۱ می‌توانید مشاهده کنید.



شکل ۳۱: نمودار شباهت کسینوسی وزن‌های دو نورون خروجی

اگر که وظیفه ما یک *classification* باشد وقتی می‌توانیم به خوبی عمل کنیم که شباهت به میزان کمینه خود برسد و این البته در حالتی خوب است که الگوهای ورودی هم کاملاً متفاوت باشند، برای مثال در بخش ورودی با جریان ثابت نويز دار که الگوها تفاوت زیادی داشتند این شباهت خیلی کم بود و نورون‌های خروجی رفتارهای متفاوتی را آموختند اما در این بخش همانطور که در شکل ۳۱ مشاهده می‌کنید نورون‌های خروجی ما الگوهای خیلی متفاوتی را نیاموخته‌اند و تقریباً می‌توان گفت که الگوها با هم مشابه بوده‌اند اما با این حال می‌توان از آن‌ها برای جداسازی کلاس‌ها استفاده کرد به عبارت دیگر ما به هر دو ورودی در این آزمایش پاداش دادیم و پس الگوهای خیلی متفاوتی نیاموختند و شباهت زیادی قابل مشاهده است.

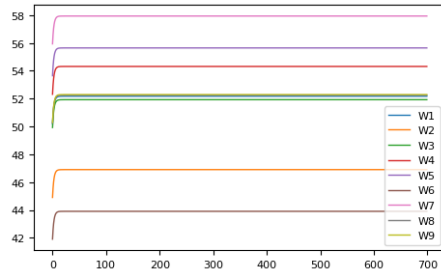
## ۶.۴ تاثیر نورون بدون فعالیت

در این بخش یک جمعیت نورونی با یک نورن به دو جمعیت نورونی ورودی اضافه می‌کنیم که به آن ورودی نمی‌دهیم تا ضربه ای نزنند همچنین یک جمعیت نورونی نیز به خروجی اضافه می‌کنیم که در آن نیز با افزایش حد آستانه کاری می‌کنیم که آن نورون ضربه ای نزنند، بنابراین در این شبکه جدید نه نورون در لایه ورودی داریم که از سه جمعیت تشکیل شده‌اند و دو جمعیت در لایه خروجی داریم که در کل سه نورون دارند.

## ۱۰.۶.۴ تاثیر نورون بدون فعالیت با جریان ورودی ثابت نويزدار

در این حالت از آنجایی که نورون خروجی‌ای که اضافه کردیم هیچ ضربه ای نمی‌زنند بنابراین وزن سیناپسی متصل به آن نورون نیز تغییری ندارد، ما نه نورون متصل به این نورون داریم که تغییرات وزن‌های سیناپسی را می‌توانید در شکل ۳۲ مشاهده کنید و همانطور که می‌بینید تغییری مشاهده نمی‌شود.

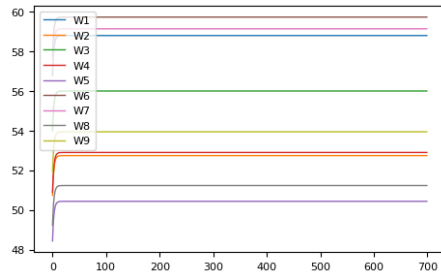




شکل ۳۲: تغییر وزن‌های سیناپسی نورون بدون فعالیت خروجی

#### ۲.۶.۴ تاثیر نورون بدون فعالیت با ورودی تصادفی پوآسون

همانطور که در قسمت قبلی توضیح داده شد از آنجایی که هیچ ضربه ای این نورون خروجی اضافه شده نمی‌زند بنابراین طبق تعریف قانون یادگیری RSTDپ هیچ تغییری در وزن‌های سیناپسی متصل به این نورون رخ نمی‌دهد و این موضوع را می‌توانید در شکل ۳۳ مشاهده کنید.



شکل ۳۳: تغییر وزن‌های سیناپسی نورون بدون فعالیت خروجی