



گزارش پروژه چهارم علوم اعصاب محاسباتی

امیرحسین انتظاری

۱۹ خرداد ۱۴۰۳

فهرست مطالب

۱	ساز و کارهای درون یک لایه نورو	۱
۱	مقدمه	۱.۱
۱	مجموعه داده مورد استفاده	۱.۱.۱
۱	یک مدل ساده	۲.۱.۱
۳	افزودن ساز و کار مهار جانبی (Lateral Inhibition)	۲.۱
۵	آزمایش مدل با اشتراک متفاوت الگوها	۱.۲.۱
۶	k-برنده همه چیز را می‌گیرند (K-Winners-Take-All)	۳.۱
۶	آزمایش مدل با اشتراک متفاوت الگوها	۱.۳.۱
۱۰	هم‌ایستایی یا هم‌ئوستازی (Homeostasis)	۲
۱۰	ساز و کار هم‌ایستایی	۱.۲
۱۰	ساز و کار هم‌ایستایی براساس فعالیت	۲.۲
۱۰	افزایش تعداد الگوها	۱.۲.۲
۱۳	ساز و کار هم‌ئوستازی براساس ولتاژ (امتیازی)	۳.۲
۱۵	(امتیازی) آزمایش با نسبت های مختلف الگوها و لایه خروجی	۴.۲

چکیده

هدف از این پروژه، آشنایی با ساز و کارهای موجود بین نورون های یک لایه و تاثیر آن ها بر فرآیند یادگیری است. در این پروژه قصد داریم سه ساز و کار مهار جانبی^۱، برنده-همه-چیز-را-می-گیرد^۲ و هم ایستایی^۳ را بررسی کنیم. در اکثر آزمایش های این پروژه، از قانون یادگیری انعطاف پذیری وابسته به زمان ضربه ($STDP$)^۴ استفاده می کنیم. همچنین بررسی می کنیم انواع الگو ها از نظر تعداد و پیچیدگی آن ها چه تاثیری بر فرآیند یادگیری می گذارد. در ادامه پارامترهای مختلف و تاثیر آن ها را نیز آزمایش خواهیم کرد.

^۱Lateral inhibition

^۲k-winners-take-all

^۳Homeostasis

^۴Spike-Timing-Dependent Plasticity

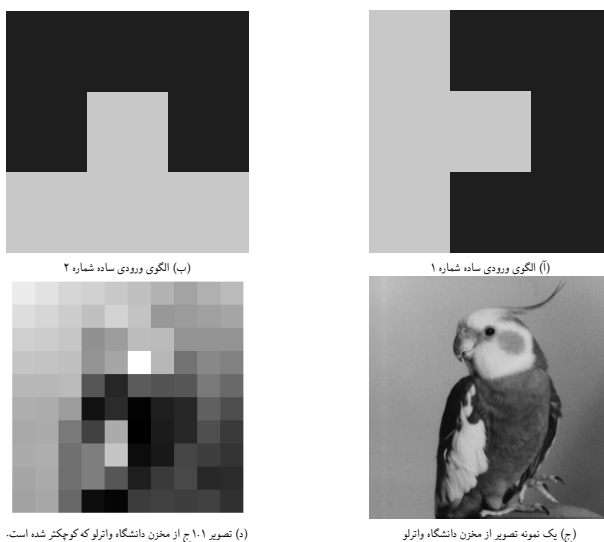
بخش ۱

ساز و کار های درون یک لایه نورونی

۱.۱ مقدمه

۱.۱.۱ مجموعه داده مورد استفاده

در این پروژه، برای الگوهای ساده، از الگوی ۱۰.۱ و ۱۰.۱ ب استفاده می‌کنیم و برای الگوهای پیچیده تر از مخزن دانشگاه واترلو^۱ با نسبت اندازه‌های مختلف استفاده می‌کنیم. این تصاویر همگی سیاه و سفید هستند. از این رو می‌توانیم کل تصویر را به صورت یک آرایه در نظر بگیریم، به طوری که سطرهای تصویر را پشت سر هم ردیف می‌کنیم. کدگذاری مورد استفاده در این پروژه نیز، کدگذاری پواسون می‌باشد.



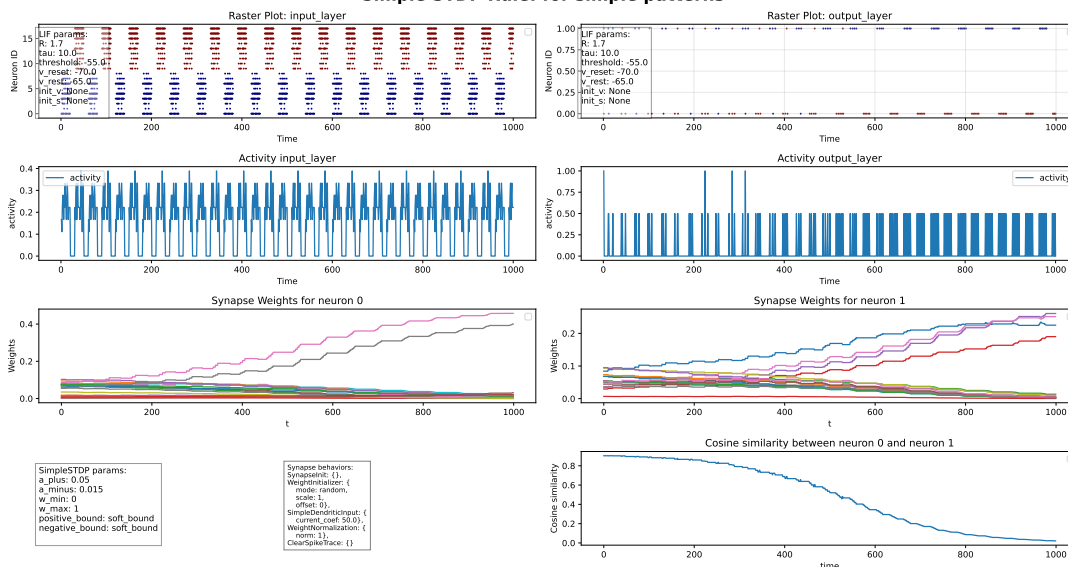
شکل ۱.۱: مجموعه داده مورد استفاده

۲.۱.۱ یک مدل ساده

در این پروژه می‌خواهیم با ساز و کارهای موجود بین نورون‌های یک لایه آشنا شویم و تاثیر آن‌ها را بر فرآیند یادگیری بررسی کنیم. برای اینکار لازم است ابتدا، یک آزمایش ساده برای حالتی که هیچ یک از این ساز و کارها درون لایه‌ها وجود ندارد انجام دهیم تا نتایج را بهتر با آن مقایسه کنیم. همانطور که در فایل پروژه نیز آمده است، در شبکه مورد نظر، تنها یک لایه ورودی و یک لایه خروجی در نظر می‌گیریم. سپس یک سیناپس با اتصال کامل و وزن‌های اولیه تصادفی بین این دو لایه ایجاد می‌کنیم. همچنین از قانون یادگیری انعطاف‌پذیری وابسته به زمان ضربه ($STDP$)^۲ برای آموزش شبکه استفاده می‌کنیم. از آنجا که این آزمایش به عنوان یک مرجع برای مقایسه در نظر گرفته می‌شود، رفتار دندریت‌ها را نیز یک رفتار ساده در نظر می‌گیریم. در نهایت، شبیه‌سازی را برای ۱۰۰۰ تکرار انجام می‌دهیم. همانطور که پروژه قبل نیز ملاحظه کردیم، قانون یادگیری $STDP$ به تنهایی نمی‌تواند در یادگیری الگوها به خوبی $RSTDP$ عمل کند. هر چند که این قانون، مطابق شکل ۲.۱ توانسته است الگوهای ورودی را به خوبی یاد بگیرد، اما این یادگیری پایدار نیست و به ازای تکرارها مختلف ممکن است به طرق متفاوت عمل کند. (شکل ۳.۱)

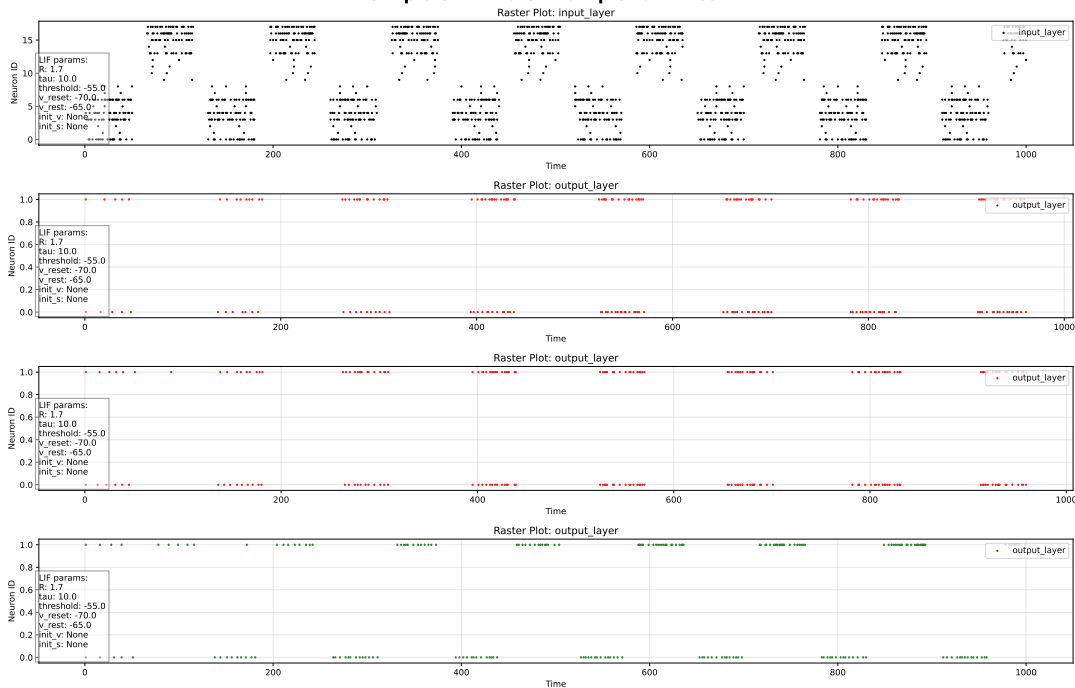
^۱مخزن تصاویر دانشگاه واترلو
^۲Plasticity Spike-Timing-Dependent

Simple STDP Rule: for simple patterns



شکل ۲.۱: یک شبکه ساده با قانون یادگیری *STDP* ساده و بدون ساز و کار اضافه. برای درک بهتر و تحلیل ساز و کار های درون یک لایه، بهتر است یک شبکه ساده در نظر گرفته و رفتار آن را با شبکه هایی که ساز و کار به آن ها اضافه می شود مقایسه کنیم. همانطور که در پروژه قبل دیدیم، قانون یادگیری *STDP* می تواند در یادگیری الگو های ساده، تا حد قابل قبولی عمل کند. هر چند این عملکرد ممکن است پایدار نباشد. (شکل ۳.۱)

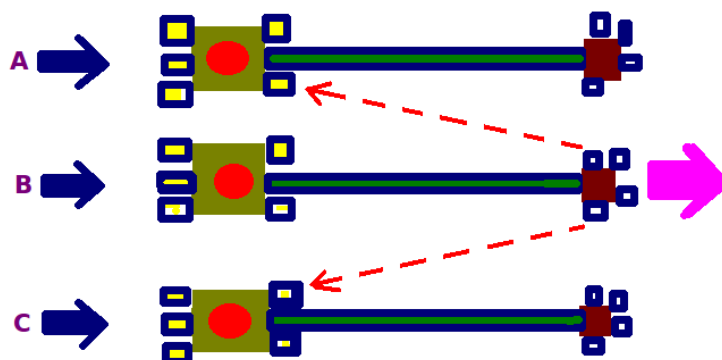
Simple STDP Rule: multiple runtimes



شکل ۳.۱: مقایسه نتایج شبیه سازی های مختلف شبکه تنها با قانون یادگیری *STDP*. همانطور که در شکل بالا نیز مشاهده می کنیم، اجرای شبیه سازی یک شبکه یکسان که تنها قانون یادگیری *STDP* را دارد، می تواند منجر به نتایج متفاوت شود. در شکل بالا، در شبیه سازی اول، شبکه نتوانسته است به خوبی الگو ها را یاد بگیرد، اما شبیه سازی های دوم و سوم عملکرد خوبی داشته اند.

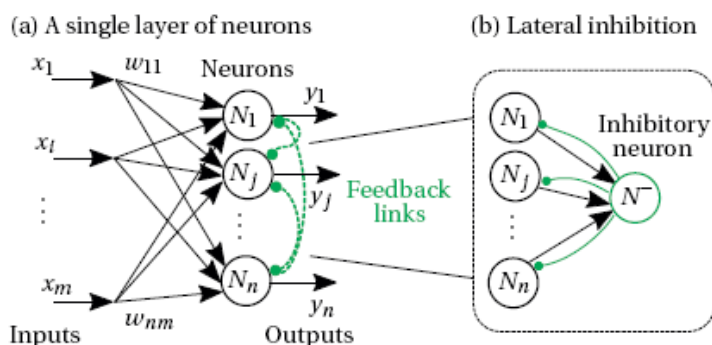
۲.۱ افزودن ساز و کار مهار جانبی (Lateral Inhibition)

در علوم اعصاب، مهار یا بازداری جانبی ظرفیت یک نورون برانگیخته برای کاهش فعالیت همسایگان خود است. مهار جانبی گسترش پتانسیل های عمل را از نورون های برانگیخته به نورون های همسایه در جهت جانبی غیرفعال می کند. این عمل یک کنتراست در تحریک ایجاد می کند که باعث افزایش ادراک حسی می شود و به آن تضاد جانبی نیز گفته می شود و عمدتاً در فرآیندهای بینایی، بلکه در پردازش لمسی، شنوایی و حتی بویایی نیز رخ می دهد. [۴]



شکل ۴.۱: محرکی که بر هر سه نورون تأثیر می گذارد، اما بر B قوی ترین یا اول تأثیر می گذارد، اگر B سیگنال های جانبی را به همسایگان A و C ارسال کند تا ضربه بزنند، در نتیجه آنها را مهار می کند. مهار جانبی در بینایی برای *sharp* کردن سیگنال های مغز (فلش صورتی) استفاده می شود.

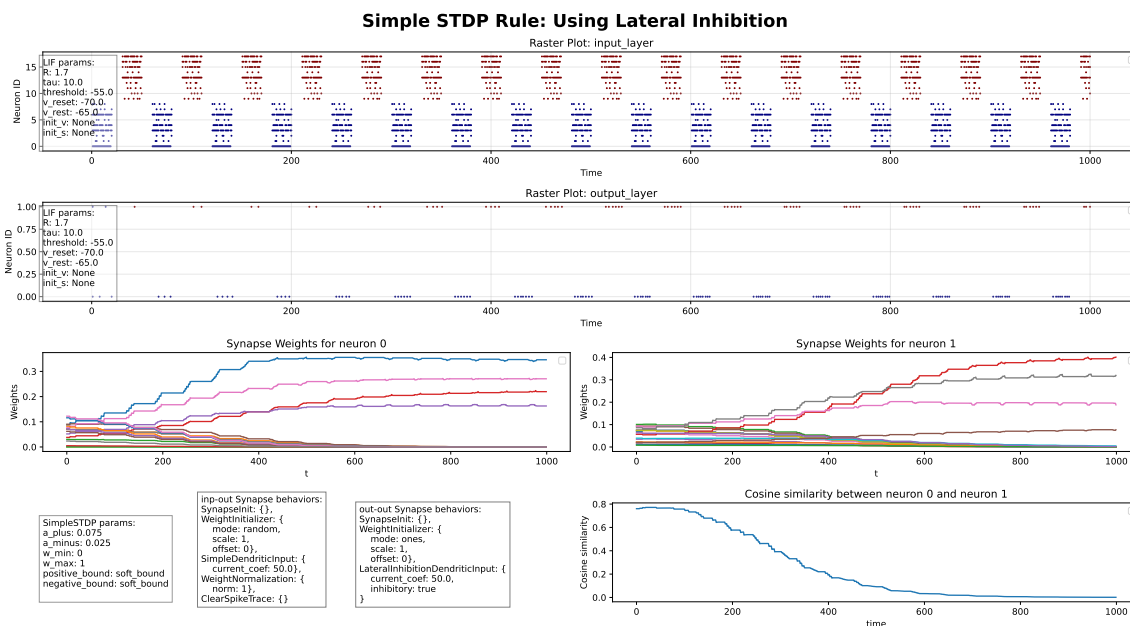
برای شبیه سازی ساز و کار مهار جانبی، ممکن است بتوان به چندین روش عمل کرد. از آنجا که می خواهیم یک نورون برانگیخته همسایگان خود را مهار کند، می توان اینکار را از طریق ایجاد یک جریان منفی از سمت این نورون به نورون های همسایه یا پایین آوردن اختلاف پتانسیل آن ها این کار را انجام داد. روشی که درون این پروژه استفاده شده است، همان روشی است که در کتابخانه *CoNeX* نیز استفاده شده است. شمای کلی این روش، در شکل ۵.۱ که از [۵] برداشته شده، آمده است.



شکل ۵.۱: یک شبکه *SNN* تک لایه با پیوندهای بازخورد (الف) نورون های تحریکی $N_1 \dots N_n$ با پیوندهای بازخوردی در ارتباط هستند که فعالیت های واگرا را در نورون ها تقویت می کند. (ب) پیوندهای بازخورد اغلب ساز و کار مهار جانبی را اجرا می کنند، که در آن یک نورون بازدارنده اضافی N^- نورون های تحریکی را مهار می کند. (رجوع کنید به [۵])

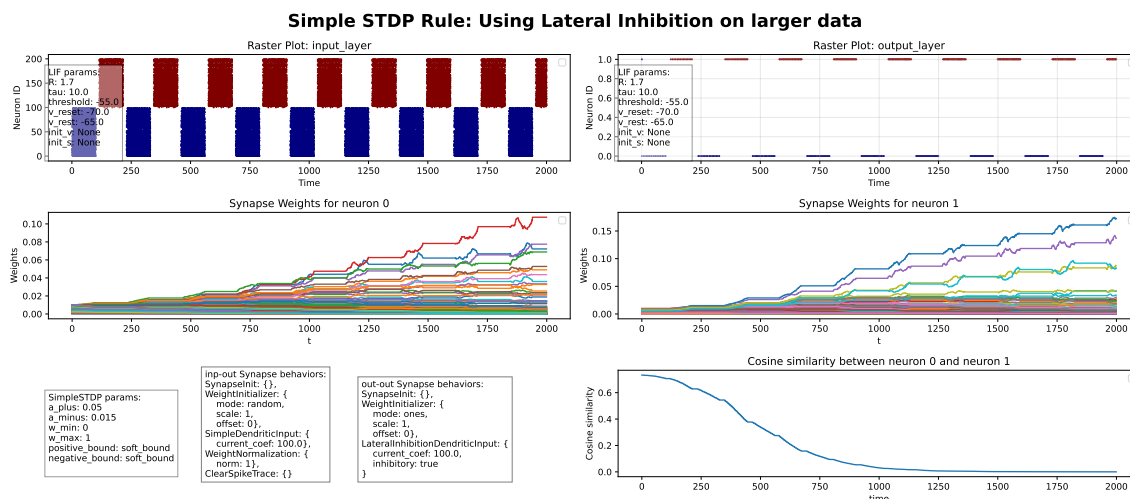
حال که با نحوه ساز و کار مهار جانبی آشنا شدیم، آن را به مدلمان افزوده و آزمایش می کنیم. برای اینکار لازم است که یک سیناپس جدید، از لایه خروجی به خودش تشکیل و دهیم، و به آن رفتار مهار جانبی را اضافه کنیم. همانطور که در شکل ۶.۱ نیز ملاحظه می کنیم افزودن ساز و کار مهار جانبی به لایه خروجی باعث می شود فرایند یادگیری مدل بهبود یابد و نه تنها در مراحل زودتری، بتواند الگوها را تشخیص دهد، بلکه با تکرار شبیه سازی، با احتمال بیشتری دقت پایداری را حفظ کند. همچنین

همانطور که ملاحظه می‌کنیم نمودار شباهت کسینوسی آن به سمت صفر میل می‌کند.



شکل ۶.۱: افزودن ساز و کار مهار جانبی به لایه خروجی. همانطور که در شکل بالا مشاهده می‌کنیم، افزودن ساز و کار مهار جانبی به لایه خروجی، باعث بهبود فرایند یادگیری مدل می‌شود. به طوری که مدل هم در مراحل زودتری الگوها را تشخیص می‌دهد، و هم با تکرار آزمایش، با احتمال بیشتری نسبت به حالتی که این ساز و کار وجود ندارد پایداری خود را حفظ می‌کند. نکته مهم دیگری که در این شکل مشاهده می‌شود، نمودار شباهت کسینوسی بین وزن های لایه خروجی است. این نمودار در حالتی که این ساز و کار وجود نداشت، پس از اینکه تا حدی کم می‌شد، نزدیک صفر نوسان میکرد، اما با افزودن ساز و کار مهار جانبی، شباهت کسینوسی بعد از حدود ۲۵۰ تکرار به مقدار صفر میل می‌کند.

تکرار آزمایش با الگوی پیچیده تر نیز (مجموعه داده دانشگاه واترلو) نتایج مشابهی به همراه خواهد داشت و مدل توانایی تشخیص الگوها را همچنان دارد. (شکل ۷.۱)



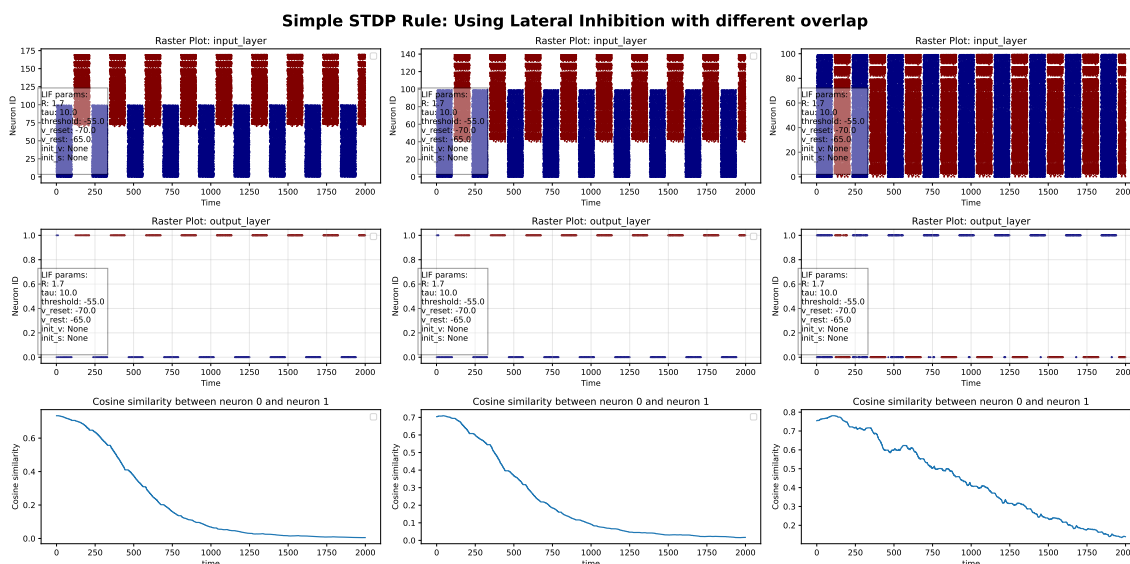
شکل ۷.۱: افزودن ساز و کار مهار جانبی به لایه خروجی: الگوهای پیچیده تر. مشاهده می‌کنیم حتی با شبیه سازی آزمایش با الگوهای پیچیده تر، یعنی تصاویر مجموعه داده دانشگاه واترلو نیز، مدل توانایی خود در تشخیص دادن را از دست نداده است.

نتیجه گیری

افزودن ساز و کار مهار جانبی به لایه خروجی، باعث می شود توانایی مدل در تشخیص الگوها بیشتر شود و نوروں ها به الگوهای متفاوت حساس تر شوند. در ساز و کار مهار جانبی، برانگیخته شدن یک نوروں، باعث می شود همسایه های خود را مهار کنند. از این رو، افزودن این ساز و کار به لایه خروجی، باعث می شود هنگامی که یک نوروں به یک الگو حساس می شود، نوروں دیگر را مهار کرده و در نتیجه، تشخیص الگوها بهتر شود.

۱.۲.۱ آزمایش مدل با اشتراک متفاوت الگوها

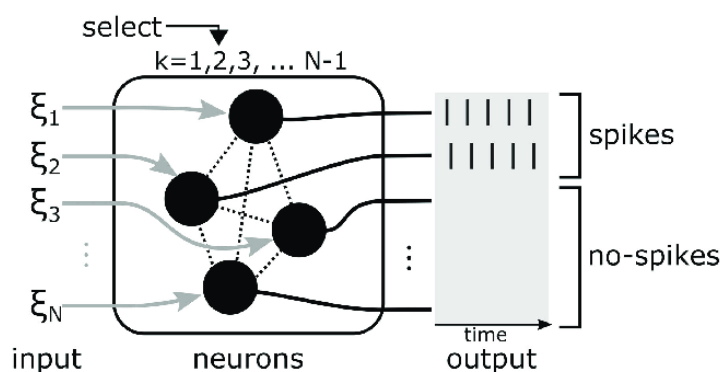
در این قسمت، آزمایش را با میزان اشتراک متفاوت الگوها تکرار می کنیم. برای اینکار، سه مقدار متفاوت 20%، 50% و 100% را بین دو الگو در نظر گرفته و برای هر کدام شبیه سازی را انجام می دهیم. مطابق شکل ۸.۱ مشاهده می کنیم که با افزودن میزان اشتراک بین الگوهای ورودی نیز همچنان مدل توانایی تشخیص دادن الگوها را دارد. هر چند در مدلی که میزان اشتراک به حداکثر خود رسیده است، مشاهده می کنیم که مدل دیرتر الگوها را تشخیص می دهد و همچنین نمودار شباهت کسینوسی آن نیز دیر به سمت صفر میل می کند.



شکل ۸.۱: ساز و کار مهار جانبی در لایه خروجی: آزمایش میزان اشتراک متفاوت الگوها مشاهده می کنیم حتی با شبیه سازی آزمایش با میزان اشتراک متفاوت الگوها نیز مدل توانایی خود در تشخیص دادن را از دست نداده است. هر چند در شکل سمت راست که مدل حداکثر اشتراک را در الگوها دارد، تشخیص دادن در مراحل دیرتری اتفاق می افتد، اما در نهایت موفق به تشخیص الگوها می شود. علاوه بر آن، نمودار شباهت کسینوسی نیز دیر به سمت صفر میل می کند. در این مدل، بقیه پارامترها همانند شکل ۷.۱ می باشد.

۳.۱ k - برنده همه چیز را می‌گیرند (K-Winners-Take-All)

ساز و کار K-winners-take-all (KwTA) در شبکه‌های عصبی ضربه‌ای (یا حتی مصنوعی) شکلی از یادگیری رقابتی است که به تعیین اینکه کدام نورون‌ها در یک شبکه به شدت در پاسخ به ورودی‌های خاص فعال می‌شوند کمک می‌کند. در زمینه شبکه‌های عصبی ضربه‌ای، که در آن سعی می‌کنیم از نحوه عملکرد نورون‌های واقعی الهام بگیریم، ساز و کار KwTA با اجازه دادن به نورون‌ها برای رقابت بر اساس پتانسیل‌های غشایی یا سرعت ضربه زدن عمل می‌کند. می‌دانیم هر نورون در یک شبکه جریان را دریافت می‌کند و این جریان ورودی بر پتانسیل آنها تأثیر می‌گذارد. نورون‌هایی که به آستانه خاصی می‌رسند ضربه می‌زنند. از این رو در پیاده سازی KwTA، از بین تمام نورون‌های یک لایه یا گروه خاص، فقط k نورونی که پتانسیل‌شان بالاتر است (یا آن‌هایی که به شدت افزایش یافته اند) مجاز به فعال ماندن یا "برنده شدن" هستند. بقیه سرکوب می‌شوند یا فعالیت آنها کاهش می‌یابد، از این رو اصطلاح "برنده ها همه چیز را می‌گیرند" برای آن به کار برده می‌شود. ساز و کار KwTA معمولاً از طریق روش‌های مختلفی مانند بازخورد مهار، که در آن نورون‌ها فعالیت نورون‌های دیگر را در مجاورت خود مهار می‌کنند، یا از طریق طراحی شبکه که خود معماری تضمین می‌کند که فقط k نورون برتر می‌توانند در هر نقطه فعال بمانند، انجام می‌شود. زمان. این رویکرد را می‌توان به عنوان شکلی از رقابت بین نورون‌ها دید که یک ویژگی اساسی سیستم‌های عصبی بیولوژیکی است و برای کارهایی مانند تشخیص الگو و فرآیندهای تصمیم‌گیری در هر دو سیستم طبیعی و مصنوعی بسیار مهم است. خلاصه این مفهوم در شکل ۹.۱ آمده است.



شکل ۹.۱: ساز و کار «k - برنده همه چیز را می‌گیرند». در هر مرحله از شبیه سازی، فقط آن نورون‌هایی که بزرگترین ورودی را دریافت می‌کنند، ضربه می‌زنند، در حالی که ضربه‌های همه نورون‌های دیگر به صورت پویا سرکوب می‌شوند.

حال می‌خواهیم این ساز و کار را به شبکه‌مان اضافه کنیم. ابتدا، حالتی را در نظر می‌گیریم که در آن شبکه، از یک لایه ورودی و خروجی و یک سیناپس از لایه ورودی به لایه خروجی استفاده می‌کند. همچنین مشابه قسمت قبل، از قانون یادگیری *STDP* برای آموزش مدل استفاده می‌کنیم. همانطور که در شکل ۱۰.۱ نیز ملاحظه می‌کنیم، افزودن ساز و کار k-winners-take-all به لایه خروجی باعث می‌شود در همان مراحل اولیه، مدل الگوها را یاد بگیرد. هر چند در این حالت نیز ممکن است در برخی شبیه سازی‌ها مدل نتواند الگوها را به خوبی یاد بگیرد، اما قرار گرفتن این ساز و کار در کنار ساز و کارهای دیگر که بررسی کردیم یا بررسی خواهیم کرد، توانایی مدل را افزایش خواهد داد. همانطور که از شکل برمی‌آید، اضافه کردن این ساز و کار باعث شده است تمایز وزن‌های بین نورون‌ها را شاهد باشیم و همچنین شباهت کسینوسی نسبت به حالت بدون ساز و کار یا با ساز و کار مهار جانبی، زودتر به صفر میل کند.

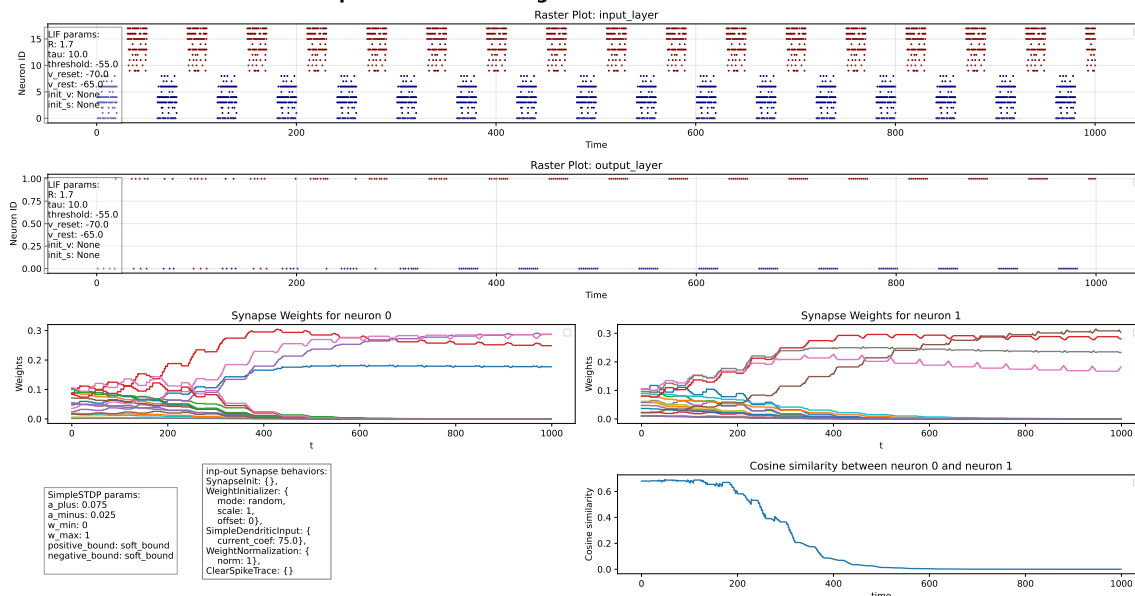
حال به لایه خروجی، ساز و کار مهار جانبی که در بخش قبل انجام دادیم را نیز اضافه می‌کنیم تا رفتار مدل را در حضور هر دو ساز و کار بررسی کنیم. مطابق شکل ۱۱.۱ مشاهده می‌کنیم که با داشتن هر دو ساز و کار k-winners-take-all و مهار جانبی، مدل به خوبی توانسته است دو الگو را تشخیص دهد. مطابق شکل، دریافت می‌شود که در مراحل اولیه، مدل در یادگیری الگوها دچار مشکل بوده است با این حال، توانسته است پس از مراحل، الگوها را یاد بگیرد. در حالی که هنگامی که این شرایط برای مدل‌های دیگر پیش می‌آید، مدل حتی پس از طی کردن مراحل بیشتر شبیه سازی نیز موفق به یادگیری الگوها نمی‌شود.

حال مدل را برای الگوهای پیچیده‌تر نیز آزمایش می‌کنیم. مطابق شکل ۱۲.۱ مشاهده می‌کنیم که مدل از عهده تشخیص الگوهای پیچیده‌تر نیز برمی‌آید. مجدداً در این حالت نیز جداسازی وزن‌ها به خوبی انجام می‌شود و شباهت کسینوسی نیز به صفر میل می‌کند. همچنین مدل پایدار است و با تکرار آزمایش نیز می‌تواند الگوها را تشخیص دهد.

۱۳.۱ آزمایش مدل با اشتراک متفاوت الگوها

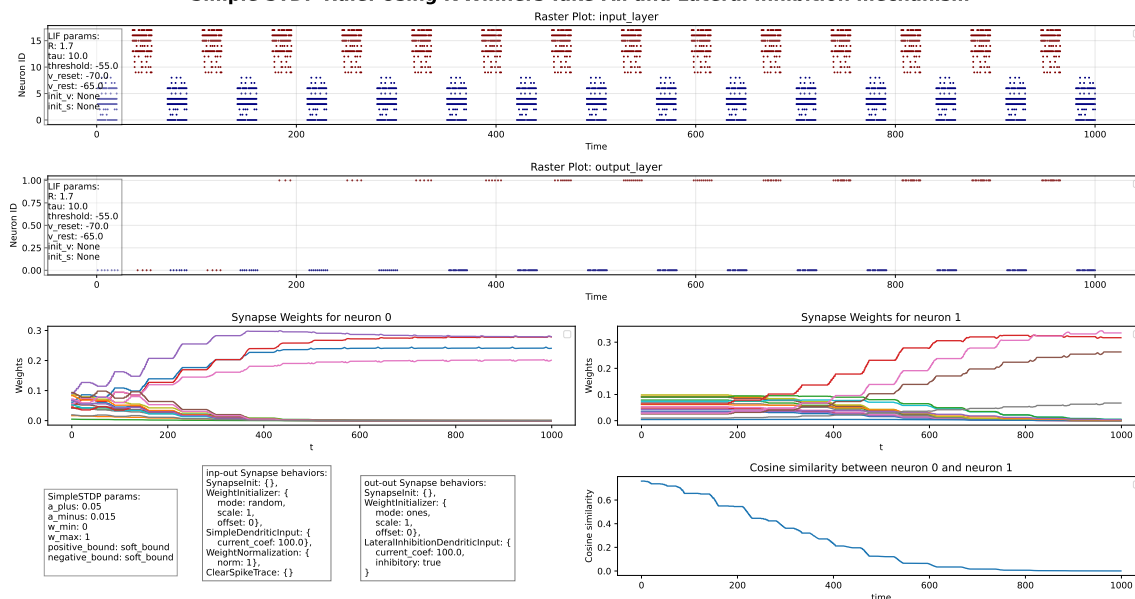
حال مشابه قسمت قبل، آزمایش‌های بالا را برای الگوها با میزان اشتراک متفاوت انجام می‌دهیم. انتظار داریم که عملکرد شبکه‌ای که هر دو ساز و کار را دارد، بهترین عملکرد را داشته باشد. شکل ۱۳.۱ این موضوع را تأیید می‌کند.

Simple STDP Rule: Using K-Winners-Take-All mechanism



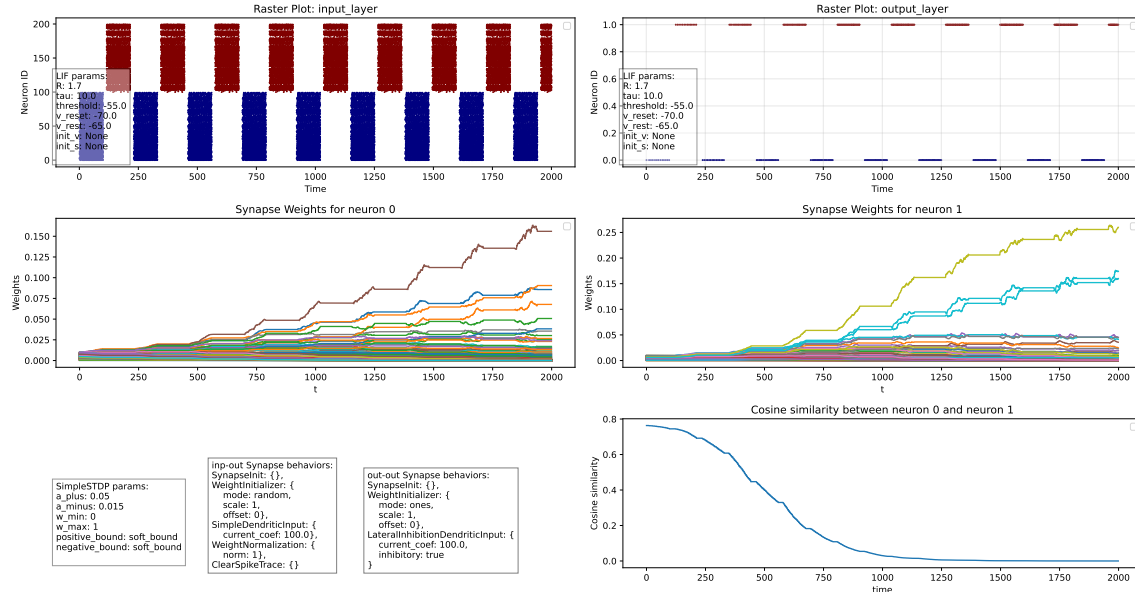
شکل ۱۰.۱: اضافه کردن ساز و کار **k-winners-take-all** به تنهایی در لایه خروجی یک شبکه ساده. مشاهده می‌کنیم که شبکه توانسته است به خوبی الگوهای ورودی را یاد بگیرد. نسبت به حالتی که ساز و کاری وجود نداشت یا حالتی که ساز و کار مهار جانبی داشتیم، یاد گرفتن الگوها زودتر اتفاق افتاده و همچنین شباهت کسینوسی زودتر به سمت صفر میل کرده است.

Simple STDP Rule: Using K-Winners-Take-All and Lateral Inhibition mechanism



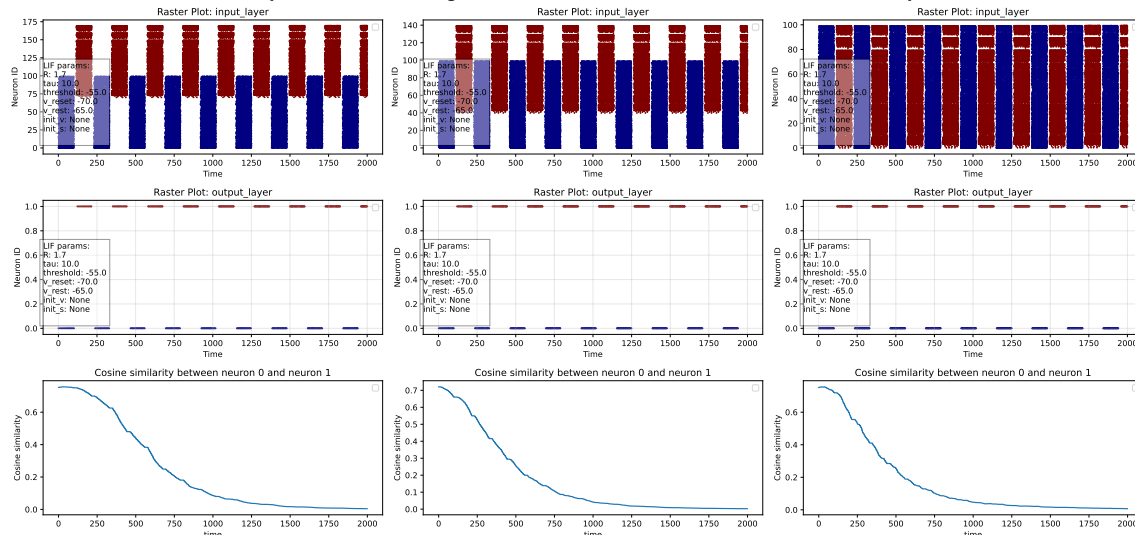
شکل ۱۰.۱: اضافه کردن ساز و کار **k-winners-take-all** به همراه ساز و کار مهار جانبی در لایه خروجی. مشاهده می‌کنیم که با داشتن هر دو ساز و کار **k-winners-take-all** و مهار جانبی، مدل به خوبی توانسته است دو الگو را تشخیص دهد. مطابق شکل، دریافت می‌شود که در مراحل اولیه، مدل در یادگیری الگوها دچار مشکل بوده است با این حال، توانسته است پس از مراحل، الگوها را یاد بگیرد. در حالی که هنگامی که این شرایط برای مدل‌های دیگر پیش می‌آید، مدل حتی پس از طی کردن مراحل بیشتر شبیه سازی نیز موفق به یادگیری الگوها نمی‌شود. همچنین پس از طی مراحل، وزن‌ها به خوبی نسبت به الگوهای ورودی تنظیم شده‌اند و مشاهده می‌کنیم که شباهت کسینوسی نیز به صفر میل کرده است. نکته مهمی که در این مدل وجود دارد این است که پایدار است و هر چقدر آن را شبیه سازی کنیم، در تشخیص الگوها موفق است.

Simple STDP Rule: Using KWTa and Lateral Inhibition mechanism on larger data



شکل ۱۲.۱: اضافه کردن ساز و کار k -winners-take-all به همراه ساز و کار مهار جانبی به لایه خروجی: داده های پیچیده تر. مطابق شکل ملاحظه می کنیم مدل از عهده تشخیص الگو های پیچیده تر نیز برمی آید. در این حالت نیز وزن ها به خوبی تنظیم می شوند و شباهت کسینوسی نیز به صفر میل می کند. نکته مهم اینکه مدل پایدار است و با تکرار آزمایش نیز می تواند الگو ها را تشخیص دهد.

Simple STDP Rule: Using KWTa and Lateral Inhibition with different overlap



شکل ۱۳.۱: اضافه کردن ساز و کار k -winners-take-all به همراه ساز و کار مهار جانبی به لایه خروجی: میزان اشتراک متفاوت. همانطور که انتظار داشتیم، شبکه ای که هر دو ساز و کار را دارد، نسبت به انواع میزان اشتراک الگو ها پایدار است و در هر سه حالت توانسته است الگو ها را تشخیص دهد. هر چند هنگامی که آزمایش می شود، برای اشتراک با میزان کم یا متوسط، همیشه مدل درست تشخیص می داد، ولی هنگامی که حداکثر اشتراک را داشتیم، در بعضی شبیه سازی ها مدل نمی توانست به درستی دو الگو را تشخیص دهد.

نتیجه گیری

مطابق آزمایش هایی که انجام دادیم، مشاهده کردیم که اضافه کردن ساز و کار k -winners-take-all به یک شبکه ساده، می تواند همانند ساز و کار مهار جانبی، قدرت مدل را در تشخیص الگو ها بهبود ببخشد. هر چند تاثیر آن اندکی بهتر از ساز و کار مهار جانبی است چرا که پایداری

مدل را نسبت به تکرار آزمایش افزایش می‌داد. اما در کل تاثیری که هر دوی این ساز و کارها روی رفتار مدل به تنهایی می‌گذارند یکسان است. ما در مدل هایمان به این علاقه داریم که هر نورون فقط هنگامی که ورودی مخصوص خود را می‌بیند ضربه بزند و در پنجره زمانی که ورودی دیگر داده می‌شود ضربه‌ای نزند. هر دوی این ساز و کارها به نحوی اینکار را سهولت می‌بخشند. یکی با دادن جریان منفی به نورون دیگر، و دیگری با سرکوب کردن نورون دیگر.

در نهایت مشاهده کردیم که هر دوی این ساز و کارها به تنهایی، تا حدودی نسبت به افزایش میزان اشتراک مقاوم هستند. هر چند اشتراک حداکثری ممکن است پایداری آن‌ها را کاهش دهد. بهترین عملکرد را هم زمانی شاهد بودیم که هر دوی این ساز و کارها به لایه خروجی اضافه شدند و مدل نسبت به تکرار آزمایش یا افزایش اشتراک الگوها پایداری نشان می‌داد. در بخش بعدی، ساز و کار دیگری به شبکه‌مان اضافه می‌کنیم.

بخش ۲

همایستایی یا هومئوستازی (Homeostasis)

۱.۲ ساز و کار همایستایی

همایستایی یا هومئوستازی^۱ در زیست‌شناسی به معنای حفظ پایداری محیط داخلی بدن و ثابت نگه داشتن شرایط فیزیکی و شیمیایی جاندار است. عملکرد بهینه جاندار در گرو این ویژگی است که متغیرهای زیادی از جمله دما و تعادل مایعات بدن را در محدوده‌ای از پیش تعیین شده نگه می‌دارد (محدوده هومئوستاتیک). پی‌اچ مایعات برون‌سلولی، غلظت یون‌های سدیم، پتاسیم و کلسیم و سطح قند خون نیز بخشی از این متغیرهاست که پیوسته کنترل می‌شود. جاندار، علی‌رغم تغییرات محیط، نوع رژیم غذایی و مقدار فعالیت بدنی، تعادل متغیرهای بدنش را، هرکدام با یک یا چند سازوکار هومئوستاتیک به‌طوری پایدار حفظ می‌کند که تمام این فرایندهای تنظیمی باهم حیات را تداوم می‌بخشد. پلاستیسته هومئوستاتیک نیز در زمینه تولیدکننده الگوهای مرکزی بسیار مهم است. در این زمینه، خواص عصبی در پاسخ به تغییرات محیطی به منظور حفظ خروجی عصبی مناسب تعدیل می‌شوند. در این قسمت، اضافه کردن هر دو نوع این ساز و کار را به لایه دوم بررسی می‌کنیم.

۲.۲ ساز و کار همایستایی براساس فعالیت

ابتدا، به عنوان اولین آزمایش، به یک شبکه ساده، این ساز و کار را اضافه کرده، و سپس به شبکه‌های بخش قبلی نیز آن را اضافه می‌کنیم. همانطور که در شکل ۱.۲ نیز مشاهده می‌کنیم، اضافه کردن این ساز و کار به تنهایی به یک شبکه ساده با قانون یادگیری STDP توانسته است نسبت به حالت ساده، بهتر الگوها را تشخیص دهد. هر چند با تکرار شبیه‌سازی، ممکن است این یادگیری انجام نشود چرا که ساز و کار هومئوستازی بیشتر در کنار ساز و کارهای دیگر برای بهبود و حفظ پایداری می‌تواند کاربرد داشته باشد.

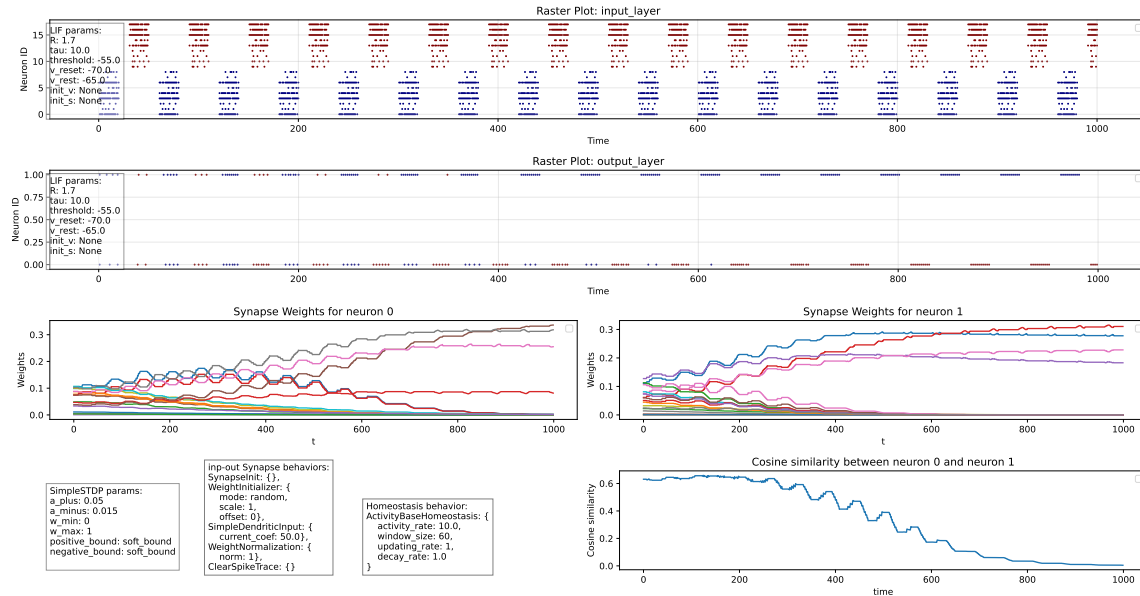
حال این ساز و کار را به شبکه قسمت قبل که دارای ساز و کارهای مهار جانبی و k -winners-take-all بود اضافه می‌کنیم. انتظار داریم که اضافه کردن این ساز و کار بتواند پایداری شبکه قبل را بهتر کند. همانطور که مطابق شکل ۲.۲ نیز مشاهده می‌کنیم، افزودن این ساز و کار به شبکه باعث می‌شود که شبکه توانایی یادگیری خود را حفظ کند و نسبت به قبل پایدارتر شود. در این مدل، ما پارامترهای ساز و کار هومئوستازی را به نحوی دادیم که در یک پنجره زمانی که ورودی داده می‌شود، فقط نصف نوروها فعالیت داشته باشند. از این رو مطابق شکل نیز ملاحظه می‌کنیم که تعداد ضربه‌ها هنگامی که یک ورودی داده می‌شود نسبت به مدل‌های قبلی بیشتر شده است و گویی که نوروها مخصوص به یک الگو، فقط و فقط در زمان ورودی دادن آن الگو فعال می‌شود.

۱.۲.۲ افزایش تعداد الگوها

حال در این مرحله، تعداد الگوهای ورودی و در نتیجه تعداد نوروهای خروجی را افزایش می‌دهیم. برای اینکار از مجموعه داده مخزن دانشگاه واترلو و همچنین یک عکس تصادفی (من در اینجا عکس خودم را به عنوان نمونه قرار داده‌ام) استفاده می‌کنیم. (شکل ۳.۲)

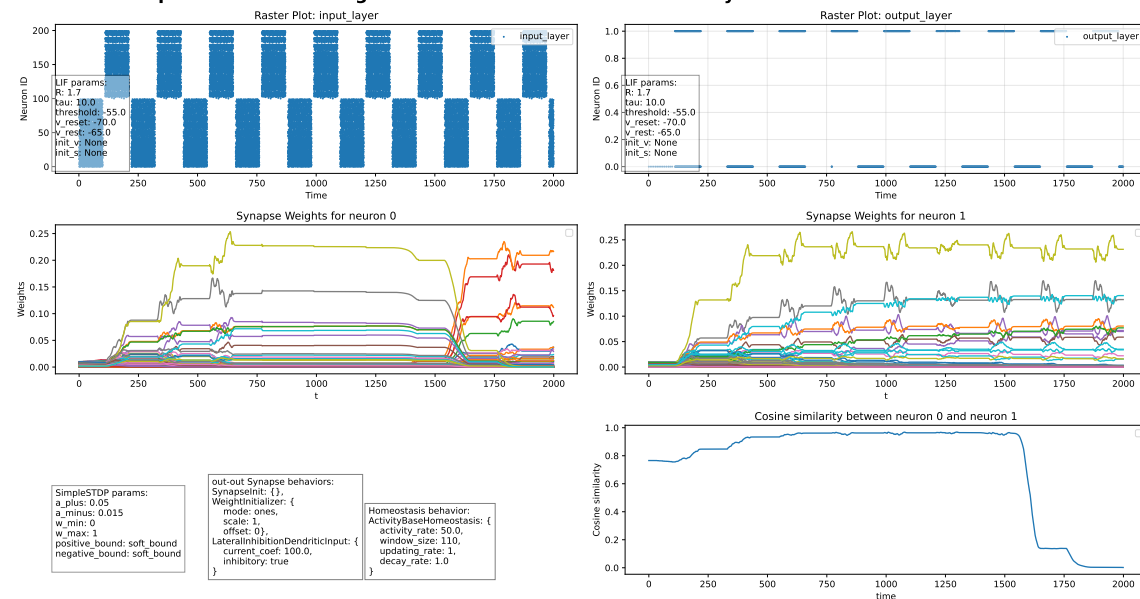
^۱Homeostasis

Simple STDP Rule: Using Activity Base Homeostasis



شکل ۱.۲: اضافه کردن ساز و کار هموستازی براساس فعالیت به یک شبکه ساده. همانطور که در شکل بالا نیز مشاهده می‌کنیم، اضافه کردن این ساز و کار به تنهایی برای یک شبکه ساده، می‌تواند نسبت به حالتی که وجود نداشته باشد، یادگیری را بهتر کن (یادگیری زودتر اتفاق بیوفتد یا تعداد نورون های فعال برای الگو بیشتر شود) اما این ساز و کار به تنهایی نمی‌تواند یادگیری مدل را تنظیم کند و بهتر است در کنار ساز و کار های دیگر شبکه مورد استفاده قرار گیرد.

Simple STDP Rule: Using KWTa + Lateral Inhibition + Activity Base Homeostasis mechanism



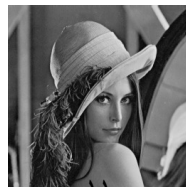
شکل ۲.۲: ساز و کار، مهار جانبی، k-winners-take-all و هموستازی براساس فعالیت در یک شبکه. همانطور که در شکل بالا نیز مشاهده می‌کنیم، اضافه کردن این ساز و کار هموستازی به شبکه قسمت قبل که هم مهار جانبی داشت و هم k-winners-take-all توانسته است الگوهای ورودی را به خوبی تشخیص دهد. همچنین ساز و کار هموستازی باعث شده که میزان فعالیت نورون ها در یک پنجره زمانی (در اینجا پنجره زمانی که ورودی داده می‌شود) متعادل شود و ما شاهد فعالیت بیشتر نورون های لایه خروجی برای الگو متعلق به خود هستیم.



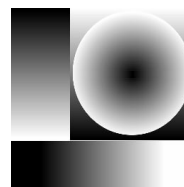
(ا)



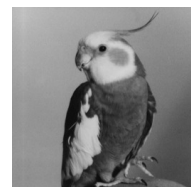
(د)



(ج)



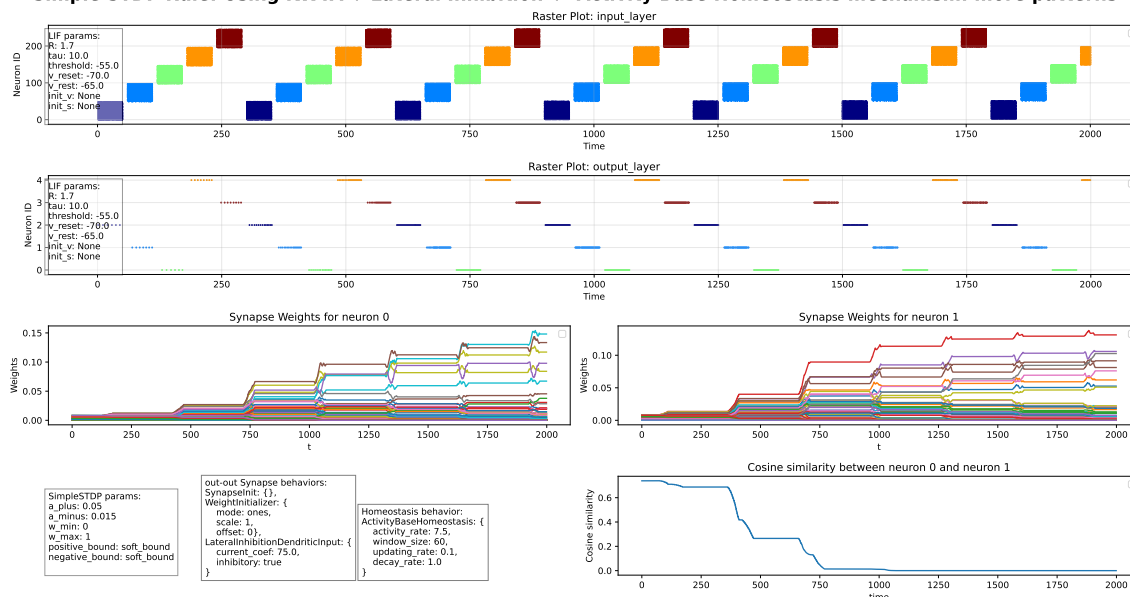
(ب)



(ل)

حال ساز و کار مورد نظر را به مدل اضافه می‌کنیم و مطابق شکل ۷.۲ ملاحظه می‌کنیم که مدل برای تعداد الگوهای بیشتر نیز می‌تواند توانایی خود در تشخیص آن‌ها را حفظ کند. حتی با تکرار شبیه سازی با دفعات زیاد نیز مدل همچنان در تشخیص الگوها موفق بوده و پایداری خود را حفظ می‌کند. فقط دقت شود که به دلیل جلوگیری از شلوغی نمودارها و تحلیل بهتر، در اینجا اندازه الگوهای ورودی را کمی کاهش دادیم.

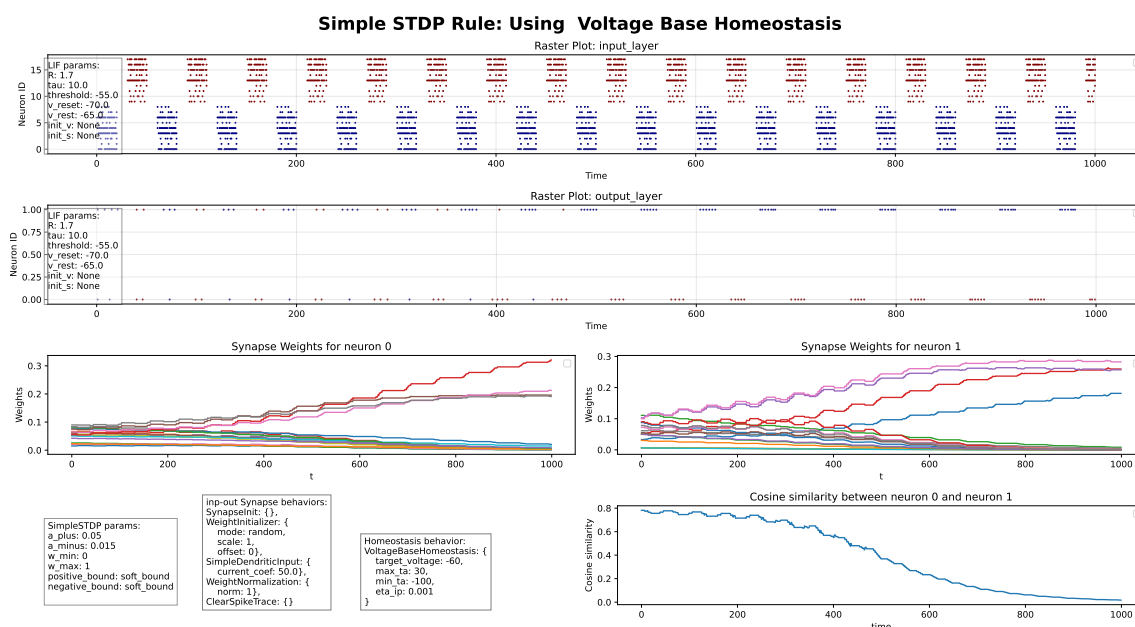
Simple STDP Rule: Using KWTa + Lateral Inhibition + Activity Base Homeostasis mechanism: more patterns



شکل ۴.۲: ساز و کار، مهار جانبی، **k-winners-take-all** و هموستازی براساس فعالیت در یک شبکه: الگوهای بیشتر مطابق شکل ملاحظه می‌کنیم که مدل توانسته است از عهده تشخیص الگوهای بیشتر نیز بربیاید. هر نوروں توانسته است یکی از الگوها را یاد بگیرد. در این آزمایش، با تکرار شبیه سازی، پایداری مدل حفظ می‌شود و هر نوروں لایه خروجی، یک الگو را تشخیص می‌دهد.

۳.۲ ساز و کار هموستازی براساس ولتاژ (امتیازی)

حال در این قسمت، ساز و کار هموستازی براساس ولتاژ را به مدل اضافه می‌کنیم. کلیت مفهوم این ساز و کار نیز همانند ساز و کار قبلی است، اما تفاوت آن در چیزی است که آن را حفظ می‌کند. در این ساز و کار، سعی می‌شود که ولتاژ نورون را در محدوده‌ای که به عنوان پارامتر می‌گیرد، تنظیم کند. ابتدا این ساز و کار را به تنهایی و به همراه ساز و کارهای دیگر به شبکه اضافه می‌کنیم و در نهایت آن را با نوع قبلی هموستازی مقایسه می‌کنیم. مطابق شکل ۵.۲ مشاهده می‌کنیم که افزودن ساز و کار هموستازی بر پایه ولتاژ توانسته است مدل را قادر سازد الگوهای ورودی را تشخیص دهد. نکته جالبی که در این آزمایش وجود داشت این است که حتی با تکرار آزمایش نیز، مدل به احتمال بیشتری نسبت به حالتی که ساز و کار وجود نداشت توانایی خود در تشخیص الگوها را حفظ می‌کرد.



شکل ۵.۲: اضافه کردن ساز و کار هموستازی براساس ولتاژ در یک شبکه ساده. مطابق شکل ملاحظه می‌کنیم که با اضافه کردن این ساز و کار نیز، مدل توانسته است الگوهای ورودی را تشخیص دهد. هر چند هنوز نیز با تکرار آزمایش مدل ممکن است در تشخیص الگوها گاهی ناموفق باشد، ولی نسبت به حالتی که شبکه ساز و کار وجود ندارد شاهد پایداری بیشتری هستیم.

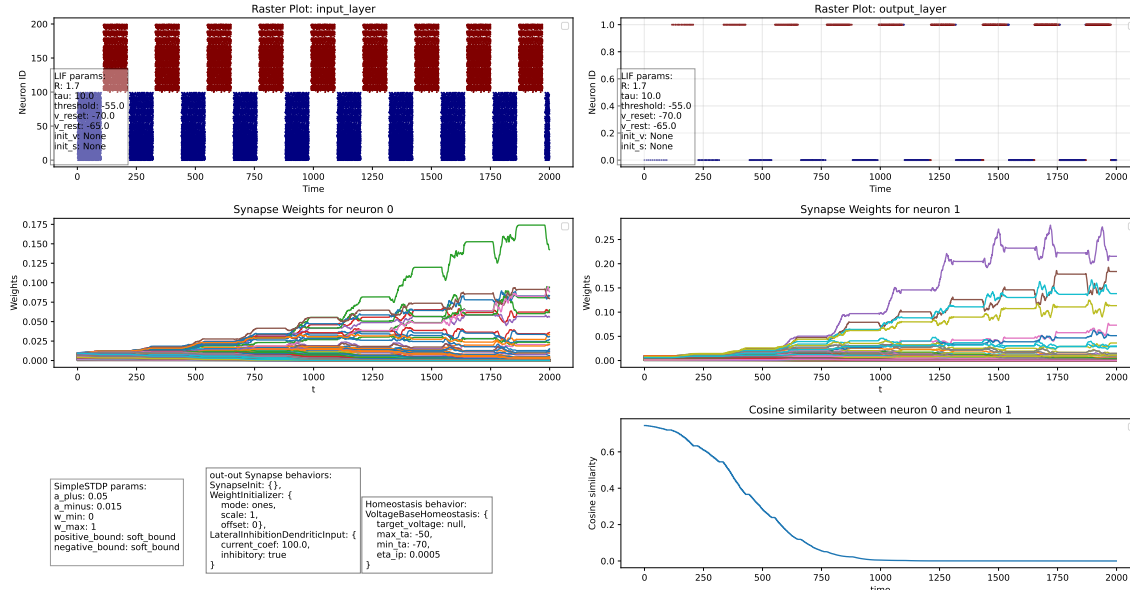
حال که تاثیر این ساز و کار را بر مدل ملاحظه کردیم، به سراغ اضافه کردن آن به شبکه‌ای که در نهایت در بخش دوم ساخته شد می‌رویم. همانطور که در شکل ۶.۲ نیز ملاحظه می‌کنیم، اضافه کردن این ساز و کار به شبکه نیز مانند قسمت قبل است و مدل توانسته است الگوها را تشخیص دهد. نکته‌ای که در مقایسه این ساز و کار با ساز و کار قبلی وجود دارد این است که ما در ساز و کار قبلی شاهد بودیم که در پنجره زمانی که ورودی‌ها داده می‌شدند، فعالیت نورون‌های لایه خروجی متراکم تر بود در حالی که چنین چیزی را در اینجا شاهد نیستیم.

مقایسه پیاده سازی

در این قسمت می‌خواهیم پیاده سازی دو ساز و کار را با یکدیگر مقایسه کنیم.

هموستازی مبتنی بر فعالیت در هموستازی مبتنی بر فعالیت، تنظیم سطح فعالیت نورون‌ها بر اساس رفتار spiking آنها در یک پنجره زمانی تعریف شده طراحی شده است. این رفتار، یک پارامتر به نام activity_rate دریافت می‌کند که نشان دهنده نرخ ضربه زدن نورون‌ها در پنجره زمانی مورد نظر است. این پنجره زمانی نیز خود با یک پارامتر به نام window_size مقدار دهی می‌شود. همچنین یک پارامتر به نام updating_rate نیز گرفته می‌شود که عامل مقیاس‌پذیری برای به‌روزرسانی آستانه‌های نورون‌ها است. علاوه بر آن پارامتر decay_rate نیز وجود دارد تا نرخ به‌روزرسانی را در طول زمان کاهش می‌دهد تا تغییرات ممکن را تثبیت کند. نحوه کارکرد این ساز و کار به این صورت است که ضربه‌ها را در یک پنجره مشخص جمع می‌کند و آستانه نورون را بر اساس انحراف از فعالیت هدف تنظیم می‌کند. اگر فعالیت خیلی زیاد باشد، آستانه افزایش می‌یابد و اگر خیلی کم باشد، کاهش می‌یابد. به‌روزرسانی آستانه‌ها با updating_rate مقیاس‌بندی می‌شود، و این نرخ پس از هر پنجره با updating_rate تغییر می‌کند.

Simple STDP Rule: Using KWTa + Lateral Inhibition + Voltage Base Homeostasis



شکل ۶.۲: ساز و کار، مهار جانبی، **k-winners-take-all** و هم‌نوسازی براساس ولتاژ در یک شبکه. همانطور که در شکل بالا نیز مشاهده می‌کنیم، اضافه کردن این ساز و کار هم‌نوسازی به شبکه قسمت قبل که هم مهار جانبی داشت و هم **k-winners-take-all** توانسته است الگوهای ورودی را به خوبی تشخیص دهد. همچنین ساز و کار هم‌نوسازی باعث شده که میزان فعالیت نورون‌ها در یک پنجره زمانی (در اینجا پنجره زمانی که ورودی داده می‌شود) متعادل شود و ما شاهد فعالیت بیشتر نورون‌های لایه خروجی برای الگو متعلق به خود هستیم.

هم‌نوسازی مبتنی بر ولتاژ این ساز و کار ولتاژ نورون‌ها را تنظیم می‌کند تا آن را در محدوده دلخواه نگه دارد. برای هدایت تنظیمات خود از آستانه‌های ولتاژ هدف، حداقل (\min_ta) و حداکثر (\max_ta) استفاده می‌کند. همچنین یک پارامتر η_{ip} نیز وجود دارد که مربوط به قدرت یا نرخ تنظیمی است که تغییرات در آن اعمال می‌شود. این ساز و کار بررسی می‌کند که آیا ولتاژ نورون از آستانه‌های تعیین‌شده فراتر می‌رود یا از آن پایین‌تر می‌آید و بر این اساس تنظیم می‌کند. تنظیمات به طور مستقیم از ولتاژ نورون کم می‌شود، (تحت تاثیر پارامتر η_{ip}).

مقایسه

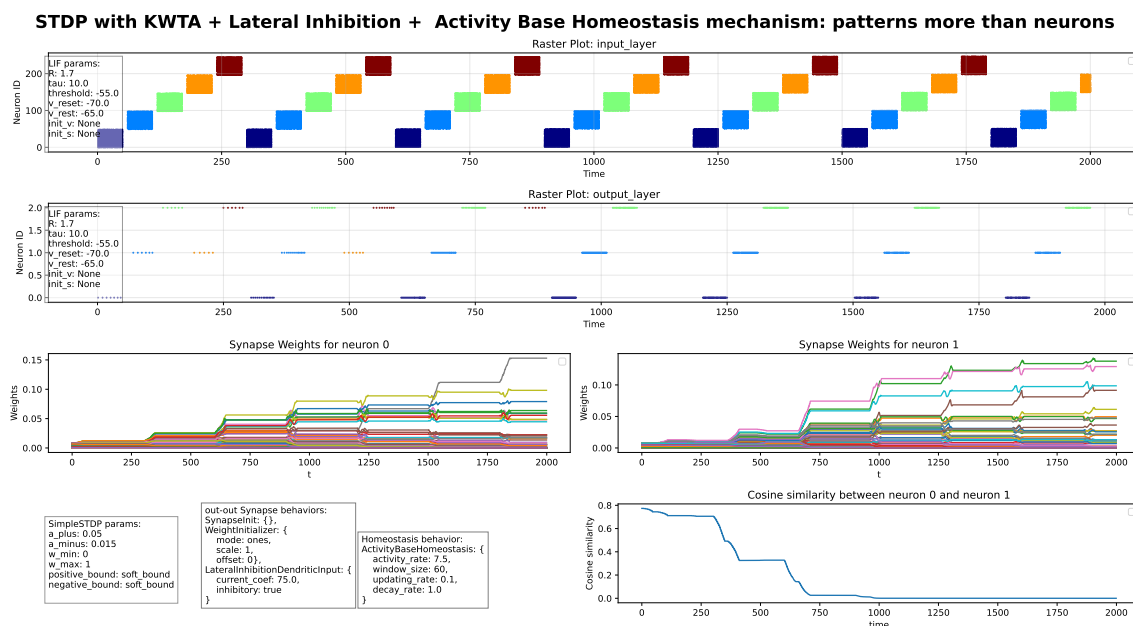
- **تمرکز:** هم‌نوسازی مبتنی بر فعالیت بر روی سطح فعالیت (نرخ زدن) نورون‌ها تمرکز دارد، در حالی که هم‌نوسازی مبتنی بر ولتاژ بر حفظ ولتاژ نورون در محدوده‌های خاص تمرکز دارد.
- **تطبیق پذیری:** هم‌نوسازی مبتنی بر فعالیت دارای یک نرخ تطبیقی است (updating_rate) که با decay_rate به‌روزرسانی می‌شود، که می‌تواند آن را پویاتر کند و به تغییرات طولانی مدت در رفتار نورون پاسخ دهد. هم‌نوسازی مبتنی بر ولتاژ از یک نرخ ثابت (η_{ip}) استفاده می‌کند، که ممکن است در طول زمان کمتر به تغییرات پاسخ دهد مگر اینکه به صورت دستی تنظیم شود. (این مورد را زمانی که شبیه سازی را تا مراحل بیشتری ادامه می‌دهیم شاهد هستیم. شکل ۶.۲ را ببینید.)
- **پیچیدگی:** رویکرد مبتنی بر فعالیت به دلیل نیاز به شمارش ضربه‌ها و محاسبه به‌روزرسانی‌ها ممکن است به طور بالقوه پیچیده‌تر باشد. رویکرد مبتنی بر ولتاژ مستقیماً ولتاژ جریان را با محدوده‌های تنظیم شده مقایسه می‌کند، که ممکن است از نظر محاسباتی ساده‌تر باشد.
- **کاربرد:** انتخاب بین این روش‌ها به نیازهای خاص شبیه سازی بستگی دارد. به عنوان مثال، اگر تمرکز بر پایداری شبکه در برابر تغییرات ناگهانی در فعالیت ضربه‌ها باشد، روش مبتنی بر فعالیت ممکن است ترجیح داده شود. اگر اطمینان از عملکرد نورون‌ها در محدوده ولتاژ حیاتی‌تر باشد (احتمالاً برای جلوگیری از اشباع یا نورون‌های غیرفعال)، روش مبتنی بر ولتاژ مناسب‌تر خواهد بود.

۴.۲ (امتیازی) آزمایش با نسبت های مختلف الگو ها و لایه خروجی

در این قسمت، آزمایش را به ازای نسبت های مختلف تعداد نورون های لایه خروجی و تعداد الگو ها انجام می دهیم. در حالت کلی، سه حالت داریم:

- تعداد الگو ها بیشتر از تعداد نورون های لایه خروجی باشد
- تعداد الگو ها با نورون های لایه خروجی برابر باشد. (این آزمایش به صورت ضمنی در بخش های قبل انجام شد)
- تعداد الگو ها کمتر از تعداد نورون های لایه خروجی باشد

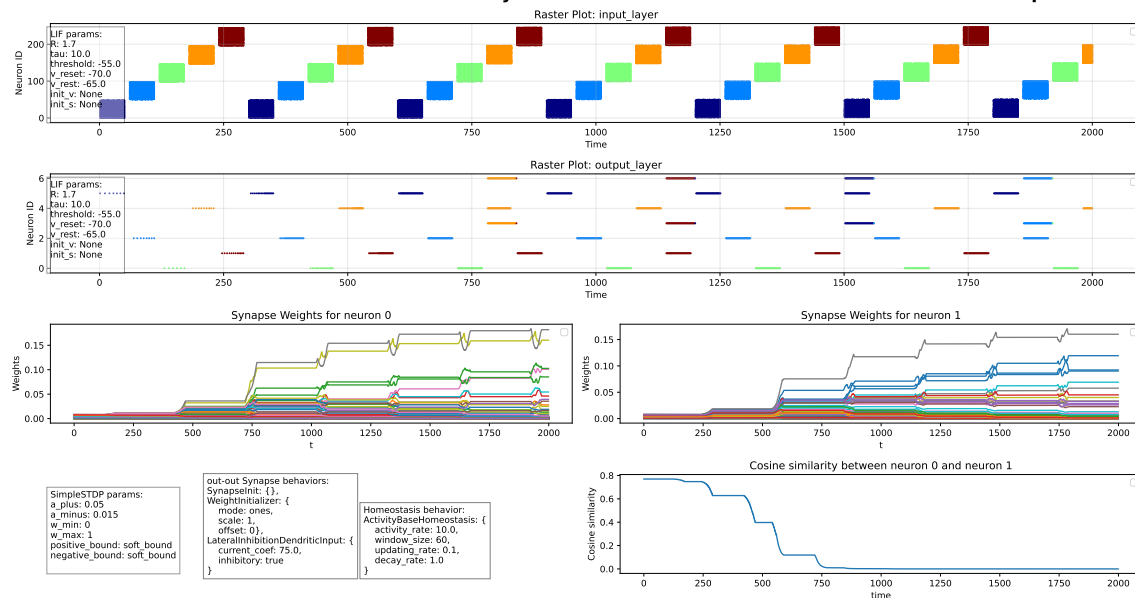
ابتدا حالتی را بررسی می کنیم که تعداد الگو ها کمتر از تعداد نورون های لایه خروجی باشد. برای این آزمایش، از یک شبکه با دو لایه و یک سیناپس از لایه ورودی به خروجی که از قانون یادگیری STDP استفاده می کند و همچنین ساز و کار مهار جانی و k -winners-take-all دارد استفاده می کنیم. ساز و کار هموستازی مبتنی بر فعالیت نیز در لایه خروجی استفاده می شود. مطابق شکل ۷.۲ مشاهده می کنیم که در این حالت، در ابتدا بعضی نورون ها به ازای دو الگو ضربه می زنند و رفته رفته برای یکی از این دو الگو حساس تر شده و فقط هنگامی که آن الگو ورودی داده می شود فعال می شوند.



شکل ۷.۲: ساز و کار، مهار جانی، k -winners-take-all و هموستازی مبتنی بر فعالیت: تعداد الگو ها ۵ عدد و تعداد نورون های خروجی ۳ عدد است. مطابق شکل مشاهده می کنیم که در این حالت، در مراحل اولیه شبیه سازی بعضی از نورون های لایه خروجی به ازای دو الگو فعال می شوند و هر چه مراحل بیشتری می گذرد نسبت به یکی از این دو الگو حساس تر شده و در نهایت فقط هنگامی که آن الگو ورودی داده می شود ضربه می زنند.

حال آزمایش را برای حالتی که تعداد الگو ها از تعداد نورون های لایه خروجی کمتر است تکرار می کنیم و نتیجه را بررسی می کنیم. در این حالت مطابق شکل ۸.۲ مشاهده می کنیم که بیشتر بودن تعداد نورون ها نسبت به الگو ها باعث می شود که به اندازه تعداد الگو ها، نورون های خروجی به الگو های متناظر حساس شوند و نورون های دیگر نیز ممکن است به یک الگوی دیگر مجددا حساس شده یا کلا ضربه ای نزنند و غیر فعال شوند.

STDP with KWTa + Lateral Inhibition + Activity Base Homeostasis mechanism: neurons more than patterns



شکل ۸.۲: ساز و کار، مهار جانبی، **k-winners-take-all** و هم‌مؤسزازی مبتنی بر فعالیت: تعداد الگوها ۵ عدد و تعداد نورون‌های خروجی ۷ عدد است. مطابق شکل مشاهده می‌کنیم که بیشتر بودن تعداد نورون‌ها نسبت به الگوها باعث می‌شود که به اندازه تعداد الگوها، نورون‌های خروجی به الگوهای متناظر حساس شوند و نورون‌های دیگر نیز ممکن است به یک الگوی دیگر مجدداً حساس شده یا کلاً ضربه‌ای نزنند و غیر فعال شوند.

کتاب نامه

- [۱] Computational Neuroscience Course, School of computer science, University of Tehran
- [۲] PymoNNtorchPytorch-adapted version of PymoNNto
- [۳] Neuronal Dynamics, Wulfram Gerstner, Werner M. Kistler, Richard Naud and Liam Paninski
- [۴] Lateral inhibition. Wikipedia [Link]
- [۵] Unsupervised Learning of Phase-Change-Based Neuromorphic Systems. Wozniak, Stanislaw Andrzej