

گزارش پروژه سوم علوم اعصاب

محمدرضا صیدگر 97222055

(1)

تمرین اول میخواهد که 2 نورون را در نظر بگیریم که با یک سیناپسی بهم متصل اند و با قانون یادگیری STDP بیایم وزن این سیناپس را دچار تغییر کنیم. همانطور که می دانیم در این قانون تغییر وزن زمانی اجرا می شود یکی از نورون های پس سیناپسی یا پیش سیناپسی spike بزند.

The values of x_j and y_i determine the weight change:

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

طبق فرمول بالا می بینیم که اگر i نورون پس سیناپسی باشد و اگر j نورون پیش سیناپسی باشد ، در لحظه spike زدن نورون i سمت راست اجرا می شود و در لحظه spike زدن نورون j سمت چپ این جمع اجرا خواهد شد.

اما برای مدل سازی این قانون ما آمدم 2 تا جمعیت نورونی در نظر گرفتیم و یک نورون از هر دو جمعیت را در نظر گرفتیم و با یک سیناپس بهم این دو نورون را وصل کردیم و روی آن سیناپس قواعد تغییر وزن STDP را لحاظ کردیم.

هر کدام از جمعیت های نورونی ما دارای 10 تا نورون است و این آزمایش را در 60 میلی ثانیه بررسی کردیم.

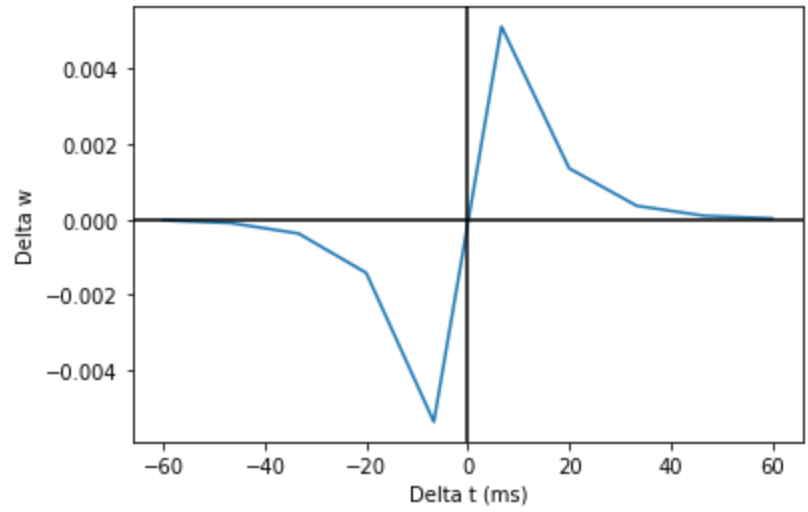
زمان های spike زدن نورون های pre و post را طبق فرمول های زیر ثابت کردیم:

Pre = 'i*time_max/(N-1) '

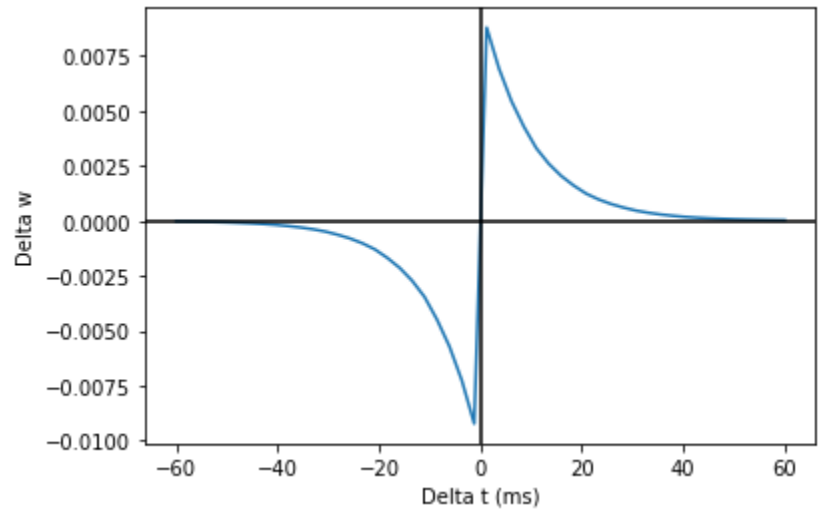
Post = ' (N-1-i) *time_max/(N-1) '

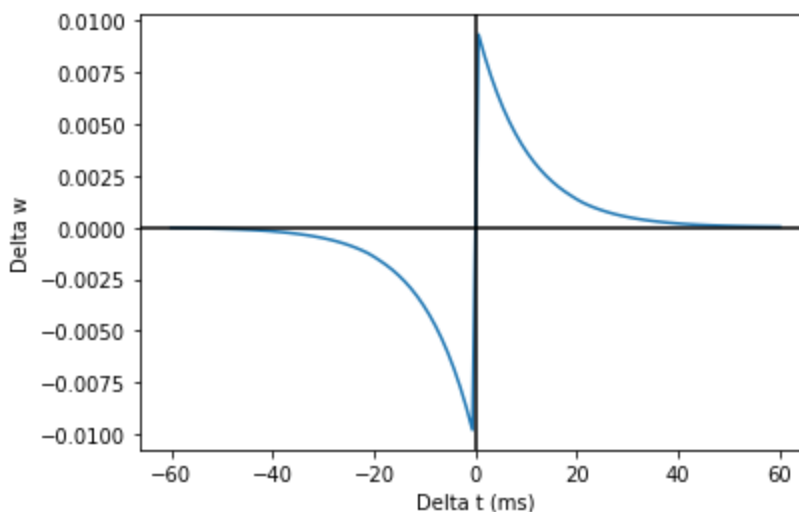
که N همان 10 نورون است.

در نهایت آمدم طبق اختلاف زمانی که دو تا نورون spike زدند نمودار تغییرات وزن را را نمایش دادیم که به شکل زیر در آمد:



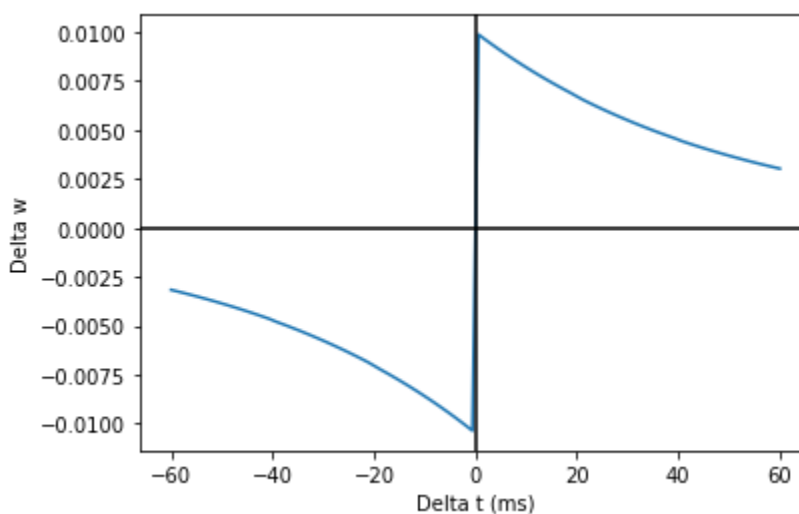
همین مدل سازی را دو بار دیگر اجرا کردیم این دفعه روی جمعیت نرونی با جمعیت های 50 و دفعه بعدی 100 نرون که نتیجه به شکل های زیر در آمد:





که می بینید نمودارهای بسیار **fit** تر نسبت به مدل قبلی داریم.

در مرحله بعدی آمدیم تاثیر هاپیر پارامتر **tau** را بررسی کردیم در مدل های قبلی برابر 10 میلی ثانیه بود ولی ایندفعه برابر 50 میلی ثانیه در نظر گرفتیم تا نمودار را دریافت کنیم:



می بینیم که افزایش این متغیر هرچند فاصله **spike** ها زیاد باشد ولی تاثیر زیادی روی وزن سیناپسی خواهد گذاشت که یعنی زیاد بود این متغیر نتیجه خوبی نخواهد داشت.

(2)

(3)

مطالعات قبلی نشان داده است که انعطاف پذیری وابسته به زمان (STDP) می تواند در شبکه های عصبی ضربه ای (SNN) برای استخراج ویژگی های بصری با پیچیدگی کم یا متوسط به روشی بدون نظارت استفاده شود.

خط دیگری از تحقیقات - با استفاده از شبکه های عصبی rate-based آموزش دیده با back-propagation است که داشتن لایه های زیاد باعث تقویت قدرت تشخیص می شود ، رویکردی که به عنوان یادگیری عمیق شناخته می شود.

در طول دهه های اخیر ، مدل های محاسباتی مختلفی از این پردازش سلسله مراتبی پیشنهاد شده است. علیرغم موفقیت های محدود مدل های اولیه ، پیشرفت های اخیر در شبکه های عصبی کانولوشن عمیق (DCNN) منجر به مدل هایی با عملکرد بالا شده است. اگرچه معماری DC ها به نوعی از سیستم بصری اولیه الهام گرفته شده است ، اما آنها کاملاً از پردازش عصبی و مکانیسم های یادگیری واقعی در قشر غافل می شوند. اگرچه معماری DC ها به نوعی از سیستم بصری اولیه الهام گرفته شده است ، اما آنها کاملاً از پردازش عصبی و مکانیسم های یادگیری واقعی در قشر غافل می شوند.

در DCNN هر بخش ، نیاز است که به بخش های دیگر floating point ارسال کنند تا دیگر بخش ها فعال شوند، در حالی که نوروں های بیولوژیکی با ارسال pulse الکتریکی به یکدیگر متصل می شوند. سرعت متوسط در سیستم های تصویری اولیه بسیار کم است . از این رو ، به نظر می رسد که نوروں ها فقط هنگامی که یک پیام مهم داشته باشند ، spike می زنند و برخی از اطلاعات می توانند در زمان spike آنها رمزگذاری شوند. چنین کدگذاری در زمان spike منجر به یک محاسبات عصبی سریع و بسیار کارآمد در مغز می شود.

DCNN های با عملکرد بالا در حال حاضر با الگوریتم نظارت شده مبتنی بر back-propagation که هیچ ریشه بیولوژیکی ندارد آموزش می بینند. اگرچه از نظر دقت خوب عمل می کند ، اما همگرایی کند است. بعلاوه ، با توجه به اینکه DCNN ها معمولاً میلیون ها پارامتر رایگان دارند ، به میلیون ها نمونه برجسب دار برای جلوگیری از over-fitting نیاز است. با این حال ، پستانداران ، به ویژه انسان ، می توانند از مثالهای بسیار دور یاد بگیرند در حالی که بیشتر اوقات هیچ برجسبی در دسترس نیست. آنها ممکن است به لطف انعطاف پذیری وابسته به زمان (STDP) ، یک مکانیسم یادگیری بدون نظارت که در قشر بینایی پستانداران رخ می دهد ، قادر به انجام این کار باشند.

تا به امروز ، شبکه های عصبی ضربه ای مختلف (SNN) برای حل وظایف شناسایی شی ارائه شده است. گروهی از این شبکه ها در واقع نسخه های تبدیل شده DCNN های سنتی هستند. ایده اصلی این است که هر واحد محاسباتی DCNN را با یک نوروں spiky که نرخ fire آن با خروجی آن واحد ارتباط ندارد جایگزین کند. هدف این شبکه ها کاهش مصرف انرژی در DCNN ها است. با این حال ، اشکال اجتناب ناپذیر در چنین کدگذاری spike rate این است که نیاز به سنسورهای مختلف در هر تصویر و زمان پردازش طولانی است. علاوه بر این ، استفاده از الگوریتم یادگیری back-propagation و داشتن هر دو خروجی مثبت (excitatory) و منفی (inhibitory) در یک نوروں از نظر زیست شناختی قابل قبول نیست. از طرفی SNN ها شبکه هایی براساس spike هستند و spiking pattern ها را یاد می گیرند.

شبکه های SDNN، کدگذاری ای بر اساس spike time neural دارند . این شبکه ها شامل لایه های به ترتیب : لایه temporal coding ، لایه convolution ای و لایه pooling . آخرین لایه که لایه convolution ای است هم یاد میگیرد و اشیا را تشخیص میدهد .

از معماری هایی برای SDNN این است که در لایه اول ما از فیلتر DoG استفاده کردیم ، که نشان دهنده تفاوت ها را در تصویر است که شدت تفاوت ها در تاخیر spike های خروجی کد میشود . در لایه های convolution، با استفاده از خروجی های لایه قبل که ویژگی های ساده ای را تشخیص داده اند استفاده می کنند ویژگی های پیچیده تری را تشخیص دهند . این لایه ها هم وقتی که ویژگی مورد نظر را پیدا کنند، یک spike می زنند. در آخر لایه های pooling وجود دارد که هدف این است که با فشرده کردن اطلاعات، حساسیت به ویژگی ها کاهش پیدا کند.

لایه Convolutional :

یک لایه convolutional شامل چندین neuronal map است. هر نورون انتخابی به یک ویژگی بصری تعیین شده توسط وزن سیناپسی ورودی آن تعیین شده. نورونها در یک نقشه خاص ویژگی بصری یکسانی را تشخیص می دهند اما در مکان های مختلف. برای این منظور ، وزن سیناپسی نورون های متعلق به یک نقشه باید از هر نظر یکسان باشد. هر نورون spike های ورودی را از سلول های عصبی واقع در یک پنجره مشخص در تمام neuronal map های لایه قبلی دریافت می کند. از این رو ، یک ویژگی بصری در لایه convolutional ترکیبی از چندین ویژگی ساده استخراج شده در لایه قبلی است.

نورون ها در تمام لایه های کانولوشن ، نورون های non-leaky integrate-and-fire هستند که spike های ورودی از سلولهای عصبی پیش سیناپسی را جمع می کنند و وقتی پتانسیل های داخلی آنها به یک threshold از پیش تعیین شده می رسد spike می زنند. هر spike پیش سیناپسی با توجه به وزن سیناپسی پتانسیل نورون را افزایش می دهد. در هر مرحله ، پتانسیل داخلی نورون i به شرح زیر به روز می شود:

$$V_i(t) = V_i(t-1) + \sum_j W_{j,i} S_j(t-1),$$

که در آن $V_i(t)$ پتانسیل داخلی نورون کانولوشن در مرحله زمانی t است ، W وزن سیناپسی بین نورون پیش سیناپسی j و نورون کانولوشن i است و S_j قطار spike ای نورون پیش سیناپسی j است. اگر V_i از threshold خود ، V_{thr} فراتر رود ، نورون یک spike منتشر می کند و V_i تنظیم مجدد می شود:

$$V_i(t) = 0 \text{ and } S_i(t) = 1, \text{ if } V_i(t) \geq V_{thr}.$$

یک مکانیسم مهار جانبی در تمام لایه های کانولوشن وجود دارد. هنگامی که یک نورون در یک مکان خاص fire می کند ، نورون های دیگر را در آن مکان متعلق به سایر neuronal map ها مهار می کند و تا زمان نمایش تصویر بعدی به آنها اجازه fire نمی دهد. علاوه بر این ، سلولهای عصبی بیش از

یک بار مجاز به fire نیستند. اینها با هم کد coding کم اما بسیار آموزنده را فراهم می کنند ، زیرا در هر مکان حداکثر یک spike وجود دارد که نشانگر وجود یک ویژگی بصری خاص در آن مکان است.

لایه های Pooling:

لایه های pooling با انجام یک عملیات max pooling غیرخطی بر روی مجموعه ای از نورون های همسایه با همان ویژگی ترجیحی ، به شبکه کمک می کنند تا invariance شود. نورون های pooling ، نورون های integrate-and-fire هستند که وزن و threshold سیناپسی ورودی آنها همه روی 1 تنظیم شده اند. از این رو ، اولین spike ورودی آنها را فعال کرده و منجر به spike خروجی می شود. هر نورون pooling حداکثر یک بار مجاز به fire است. لازم به ذکر است که هیچ گونه یادگیری در لایه های pooling رخ نمی دهد. نقش مهم دیگر لایه های pooling ، فشردن سازی اطلاعات بصری است.

یادگیری مبتنی بر STDP:

یادگیری فقط در لایه های convolutional اتفاق می افتد که باید با ترکیب ویژگی های ساده تر استخراج شده در لایه قبلی ، ویژگی های بصری را تشخیص دهد. یادگیری لایه به لایه انجام می شود ، یعنی یادگیری در یک لایه convolutional وقتی شروع می شود که یادگیری در لایه convolutional قبلی نهایی شده. نسخه ساده STDP به شرح زیر است:

$$\Delta w_{ij} = \begin{cases} a^+ w_{ij} (1 - w_{ij}), & \text{if } t_j - t_i \leq 0, \\ a^- w_{ij} (1 - w_{ij}), & \text{if } t_j - t_i > 0, \end{cases}$$

که در آن i و j به ترتیب به شاخص نورون های پس و پیش از سیناپسی اشاره می کنند ، t_j و t_i زمانهای مربوط به spike هستند ، Δw_{ij} اصلاح وزن سیناپسی است ، و a^+ و a^- دو پارامتر هستند که میزان یادگیری را مشخص می کنند. توجه داشته باشید که اختلاف زمانی دقیق بین دو spike در تغییر وزن تأثیری ندارد اما فقط علامت آن در نظر گرفته می شود. انتخاب مقادیر بزرگ برای پارامترهای یادگیری (به عنوان مثال ، a^+ و a^-) حافظه یادگیری را کاهش می دهد ، بنابراین ، نورون ها آخرین تصاویر ارائه شده را یاد می گیرند و تصاویر قبلاً دیده شده را یاد نمی گیرند. همچنین ، انتخاب مقادیر بسیار کوچک ، روند یادگیری را کند می کند.

با تنظیم a^- بزرگتر از a^+ ، وزن سیناپسی به تدریج تحلیل می رود تا جایی که نورون ها دیگر نمی توانند به threshold خود برسند تا fire کند. بنابراین ، a^+ بهتر است از a^- بزرگتر باشد ، با این حال ، با تنظیم a^+ خیلی بزرگتر از a^- ، نورون ها تمایل دارند بیش از یک الگو یاد بگیرند و به همه آنها پاسخ دهند. در مجموع بهتر است a^+ و a^- نه خیلی بزرگ و نه خیلی کوچک انتخاب شوند و بهتر است a^+ کمی بیشتر از a^- تنظیم شود.

در طول یادگیری یک لایه convolutional ، نوروں ها در همان map ، همان ویژگی را در مکان های مختلف تشخیص می دهند.
همگرایی یادگیری لایه | کانولوشن :

$$C_l = \sum_f \sum_i w_{f,i}(1 - w_{f,i})/n_w$$

$w_{f,i}$ وزن سیناپسی از ویژگی f ام است و n_w تعداد کل وزن های سیناپسی (مستقل از ویژگی ها) در آن لایه است.

لایه Global pooling و classification :

لایه global pooling فقط در مرحله طبقه بندی استفاده می شود. نوروں های آخرین لایه global max pooling را روی neuronal map های مربوطه در آخرین لایه کانولوشن انجام می دهند. برای هر ویژگی فقط یک مقدار خروجی وجود دارد که نشان دهنده وجود آن ویژگی در تصویر ورودی است. خروجی لایه global pooling روی تصاویر آموزشی برای آموزش طبقه بندی خطی SVM استفاده می شود. در مرحله آزمایش ، تصویر شی آزمایش توسط شبکه پردازش می شود و خروجی لایه global pooling برای تعیین دسته بندی به classifier ارسال می شود.

بررسی:

DCNN های supervised اخیر در چالش برانگیزترین مجموعه داده های شناسایی شی مانند Imagenet به دقت بالایی رسیده اند. معماری این شبکه ها عمدتاً از پردازش سلسله مراتبی عمیق در قشر بینایی الهام گرفته اند. با این حال مکانیسم های یادگیری و پردازش عصبی اعمال شده در DCNN ها با قشر بینایی سازگار نیست ، جایی که نوروں ها با استفاده از spike ها ارتباط برقرار می کنند و الگوهای spike ورودی را به طور عمده unsupervised یاد می گیرند. بکارگیری چنین مکانیزم هایی در DCNN می تواند مصرف انرژی آنها را بهبود بخشد و نیاز آنها را به یادگیری supervised گران قیمت با میلیون ها تصویر دارای برچسب کاهش دهد.

در مقاله یک SDNN مبتنی بر STDP با spike-time coding پیشنهاد داده شده. هر نوروں حداکثر یک بار مجاز به fire بود ، جایی که زمان spike آن اهمیت ورودی بصری آن را نشان می دهد. بنابراین ، نوروں هایی که زودتر fire می کنند اطلاعات بصری برجسته تری دارند و از این رو به آنها اجازه داده شده است که STDP را انجام دهند و الگوهای ورودی را بیاموزند.

SDNN پیشنهادی در چندین مجموعه داده تصویر مورد ارزیابی قرار گرفت و به دقت بالایی رسید.

SDNN پیشنهادی چندین مزیت بر همتایان خود دارد. اول ، SDNN پیشنهادی اولین شبکه عصبی ضربه ای با بیش از یک لایه قابل یادگیری است که می تواند تصاویر شی طبیعی را در مقیاس بزرگ پردازش کند. دوم ، به دلیل استفاده از یک کدگذاری کارآمد زمانی ، که اطلاعات بصری را در زمان spike های اول کد می کند ، می تواند تصاویر ورودی را با تعداد کمی spike و در چند مرحله زمانی پردازش کند. سوم ، SDNN پیشنهادی از قانون یادگیری STDP با الهام از زیست و کاملاً unsupervised بهره می برد که می تواند ویژگی های شی تشخیصی را بیاموزد و از زمینه های بی ربط غافل شود. همچنین بهتر از supervised deep network بود که عمدتاً از overfitting و کمبود داده آموزش کافی رنج می برد.

(4)

R-STDP رویکرد تغییر یافته STDP است ، R-STDP ویژگی های تبعیض آمیز و تفاوت ها را استخراج می کند ، در حالی که STDP کلاسیک شباهت ها و الگوهای تکراری را یاد می گیرد. در نتیجه ، R-STDP عملکرد بهتری نسبت به STDP دارد. علاوه بر این ، R-STDP برای یادگیری آنلاین مناسب است و می تواند با تغییراتی مانند label permutations سازگار شود. هم استخراج ویژگی و هم طبقه بندی با spike انجام می شود که با استفاده از حداکثر یک spike در هر نورون انجام می شود. بنابراین شبکه از نظر سخت افزاری سازگار و دارای انرژی کم است.

همانطور که گفته شد STDP در یافتن ویژگی های مکرر از نظر آماری به خوبی کار می کند ، با این حال ، به عنوان الگوریتم یادگیری بدون نظارت ، در تشخیص ویژگی های نادر اما تشخیصی برای ویژگی های مهم مانند تصمیم گیری با مشکل روبرو می شود. چندین مطالعه نشان می دهد که سیستم پاداش مغز نقشی حیاتی در تصمیم گیری و شکل گیری رفتارها دارد. این امر به عنوان یادگیری تقویتی (RL) نیز شناخته می شود ، که به وسیله آن یادگیرنده ترغیب می شود رفتارهای پاداش آور را تکرار کرده و از رفتارهایی که منجر به مجازات می شوند ، اجتناب کند. مشخص شده است که دوپامین ، به عنوان یک تنظیم کننده عصبی ، یکی از مواد شیمیایی مهم در سیستم پاداش است. همچنین نشان داده شده است که دوپامین ، و همچنین برخی دیگر از تنظیم کننده های عصبی بر synaptic plasticity تأثیر می گذارد مانند تغییر قطب یا تنظیم پنجره زمانی STDP.

یکی از ایده های خوب مطالعه شده برای مدل سازی نقش سیستم پاداش ، تعدیل یا حتی معکوس کردن تغییر وزن تعیین شده توسط STDP است که به آن reward-modulated STDP

R-STDP گفته می شود. R-STDP رد سیناپس های واجد شرایط STDP را ذخیره می کند و تغییرات وزن را در زمان دریافت سیگنال اعمال می کند. چه سیگنال پاداش یا مجازات. همچنین نشان داده شده که R-STDP قادر است با استفاده از کدگذاری ورودی زمانی یا نرخ و یادگیری *firing rate* ، مسئله XOR را حل کند. گفتیم که STDP تمایل به استخراج ویژگی های مکرر دارد که لزوماً برای کار مورد نظر مناسب نیستند. در این تحقیق ، یک SNN سلسله مراتبی مجهز به R-STDP برای حل شناسایی شی بصری در تصاویر طبیعی ، بدون استفاده از طبقه بندی کننده خارجی ارائه شده. در عوض یک سری نورون ها را مخصوص تشخیص کلاس مورد نظر تعیین شده که اگر هدف یا *target* آنها به شبکه ارائه شود ، در اسرع وقت برای *fire* کردن تقویت می شوند. بنابراین ، محرک های ورودی صرفاً براساس تأخیرهای سنبله اول به روشی سریع و قابل قبول از نظر زیست شناختی طبقه بندی می شوند. R-STDP شبکه ما را قادر می سازد تا ویژگی های تشخیصی خاص کار را پیدا کند ، بنابراین ، هزینه محاسباتی سیستم شناسایی نهایی را کاهش می دهد. R-STDP شبکه ما را قادر می سازد تا ویژگی های تشخیصی خاص را پیدا کند ، بنابراین ، هزینه محاسباتی سیستم شناسایی نهایی را کاهش می دهد. شبکه بر اساس مدل Masquelier و Thorpes با چهار لایه است. لایه اول شبکه بر اساس برجستگی لبه ها ، تصویر ورودی را به تأخیر *spike* تبدیل می کند. این قطار *spike* تحت عملیات *local pooling* در لایه دوم قرار می گیرد. لایه سوم شبکه شامل چندین نورون *integrate-and-fire* است که اطلاعات دریافتی لبه ها را ترکیب می کند و ویژگی های پیچیده را استخراج می کند. این تنها لایه قابل آموزش در شبکه ما است که برای *synaptic plasticity* از R-STDP استفاده می کند. سیگنال (پاداش / مجازات) برای *synaptic plasticity* توسط لایه چهارم ارائه می شود ، که در آن تصمیم شبکه گرفته می شود. شبکه ما فقط از اولین *spike* ساطع شده توسط نورون ها در لایه سوم برای تصمیم گیری ، بدون استفاده از طبقه بندی کننده خارجی استفاده می کند. اگر تصمیم آن درست / نادرست باشد ، یک سیگنال پاداش / مجازات جهانی تولید می شود. علاوه بر این ، به منظور افزایش بهره وری محاسباتی ، به هر سلول در شبکه اجازه داده می شود که فقط یک بار در هر تصویر بچرخد. زدن حداکثر یک *spike* در هر نورون نه تنها کارایی محاسباتی ، بلکه واقع گرایی بیولوژیکی هم است. تصمیم گیری بدون هیچ طبقه بندی با حداکثر یک *spike* در هر نورون ، روش پیشنهادی را به یک روش مناسب برای پیاده سازی سخت افزار تبدیل می کند. دو آزمایش اسباب بازی برای نشان دادن توانایی های R-STDP انجام شده . نشان داده شده شبکه ای که از R-STDP استفاده می کند با استفاده از منابع محاسباتی کمتر از STDP دارای ویژگی های آموزنده است. همچنین نشان داده شده که R-STDP می تواند در صورت لزوم ، با تشویق وی برای یادگیری آنچه قبلاً یاد گرفته است ، رفتار یک نورون را تغییر دهد. نتایج

آزمایش ها مزیت استفاده از R-STDP نسبت به STDP در یافتن ویژگی های افتراقی خاص را نشان می دهد.
حال ساختار شبکه را می خواهیم بررسی کنیم:

لایه S1:

هدف این لایه استخراج لبه های جهت دار از تصویر ورودی خاکستری است و متناسب با آنها به زمان تاخیر spike تولید می کند. برای این منظور، تصویر ورودی با فیلترهای Gabor با چهار جهت مختلف ترکیب می شود. بنابراین، این لایه شامل چهار feature map است، که هر یک، یک زاویه را تشخیص می دهند.

لایه C1:

اولین لایه پیچیده ما یک لایه local pooling روی spike های حاصل از لایه S1 است. در اینجا، 4 شبکه عصبی 2D وجود دارد که مربوط به هر یک از جهت گیری ها است. هر نورون C1 یک عملیات local pooling را از طریق پنجره ای به اندازه $wc1 \times wc1$ و گام rc1 بر روی نورون های S1 در یک شبکه خاص انجام می دهد، پس از آن، بلافاصله پس از دریافت spike ورودی اولیه خود، یک spike منتشر می کند. این عملیات pooling افزونگی لایه S1 را کاهش می دهد و تعداد نورون های مورد نیاز را کاهش می دهد، که در نتیجه کارایی محاسبات را افزایش می دهد. این همچنین یک local invariance را به موقعیت لبه های زاویه دار اضافه می کند.

لایه S2:

این لایه اطلاعات ورودی مربوط به لبه های زاویه دار را ترکیب کرده و آنها را به ویژگی های پیچیده معنی دار تبدیل می کند. همانطور که نورون ها fire می کنند، وزن سیناپسی آنها بر اساس ترتیب spike های قبل و بعد از سیناپسی و همچنین سیگنال پاداش / مجازات در حال بروزرسانی هستند. این سیگنال از فعالیت لایه بعدی گرفته شده است، که نشان دهنده تصمیم شبکه است. علاوه بر این، وزن اولیه سیناپس ها به طور تصادفی با میانگین 0.8 و انحراف معیار 0.05 انتخاب می شود. انتخاب مقادیر کم یا متوسط برای نتایج متوسط منجر به نوروهای غیرفعال و آموزش ندیده می شود. علاوه بر این، مقادیر زیاد برای واریانس تأثیر وضعیت اولیه شبکه را افزایش می دهد. بر این اساس، مقدار متوسط زیاد با واریانس کوچک انتخاب مناسبی است.

لایه C2:

این لایه دقیقاً شامل n نورون است و هر کدام به یکی از شبکه های عصبی $S2$ اختصاص داده شده اند. یک نورون $C2$ فقط اولین spike ای را که از شبکه عصبی مربوطه دریافت می کند ، گسترش می دهد. همانطور که قبلاً ذکر شد ، فعالیت سلولهای عصبی $C2$ نشان دهنده تصمیم شبکه است. برای این منظور ، سلول های عصبی $C2$ را به چندین گروه تقسیم کرده و هر گروه را به دسته خاصی از محرک های ورودی اختصاص می دهیم. سپس فرض بر این است که تصمیم شبکه در مورد گروه محرک ورودی همان کسی است که گروه آن اولین spike را در میان گروههای $C2$ دیگر منتشر می کند.

یک مکانیسم یادگیری تقویت کننده برای به روزرسانی وزنهاى پیش سیناپسی نورونهای $S2$ پیشنهاد شده. در اینجا ، مقدار تغییر وزن توسط یک سیگنال پاداش / مجازات تعدیل می شود ، که با توجه به درستی / نادرستی تصمیم شبکه دریافت می شود. همچنین از یک برنده در میان نورونهای $S2$ استفاده شده ، که در آن یکی با اولین spike برنده است و تنها کسی است که وزن سیناپسی خود را به روز می کند. این نورون یکی از تصمیم گیرنده های شبکه است.

اجتناب از overfitting:

از مشکلات یادگیری تقویتی ، ممکن است برای بدست آوردن حداکثر پاداش ممکن نسبت به نمونه های آموزش ، در local optima گیر کنیم یا overfit بشیم. به منظور کمک به شبکه ، دو مکانیسم اضافی که برای تشخیص شی استفاده می شود را بیان می کنیم:

مورد اول) Adaptive learning rate:

از آنجا که وزنهاى اولیه نورونها به طور تصادفی تنظیم می شوند ، تعداد نمونه های طبقه بندی اشتباه در ابتدای مرحله آموزش نسبتاً زیاد است (یعنی عملکرد در حد شانس است). با ادامه آزمایشات ، نسبت نمونه های طبقه بندی شده صحیح به طبقه بندی های غلط افزایش می یابد. در صورت بالا بودن طبقه بندی نادرست ، شبکه سیگنال های مجازات بیشتری دریافت میکنند ، که به سرعت آن وزن های سیناپسی را ضعیف می کند. به همین ترتیب ، وقتی نرخ طبقه بندی صحیح بالاتر می رود ، میزان کسب پاداش نیز افزایش می یابد. در این حالت ، شبکه ترجیح می دهد با انتخاب بیشتر و بیشتر برای اصلاح نمونه ها ، طبقه بندی غلط را انجام ندهد و برای سایرین در حالت silent باشد. در هر دو حالت ، overfitting به دلیل تأثیر نامتعادل پاداش و مجازات اتفاق می افتد. برای برطرف کردن این قضیه یک متغیر adjustment برای اصلاح وزن ضرب می شود که تاثیر نمونه های درست و نادرست آموزشی را متعادل می کند.

:Dropout

در یک سناریوی یادگیری تقویتی ، هدف یادگیرنده به حداکثر رساندن ارزش مورد انتظار برای کسب پاداش است. در اینجا از آنجا که شبکه فقط نمونه های آموزش را می بیند ، ممکن است تعداد مشخصاتی را پیدا کند که برای طبقه بندی صحیح تقریباً همه نمونه های آموزش کافی باشد. به نظر می رسد این مسئله در مواجهه با مشکلات پیچیده باعث بروز **overfitting** شدید می شود و شبکه ترجیح می دهد برخی از نورون ها را آموزش نبیند. این نورون ها **hit rate** شبکه را روی نمونه های **test** کاهش می دهند. در اینجا ، از روش **dropout** استفاده می شود، که باعث می شود یک نورون **C2** با احتمال p_{drop} به طور موقت خاموش شود. این روش باعث افزایش میزان درگیری کلی سلولهای عصبی می شود ، که به نوبه خود ، نه تنها احتمال یافتن ویژگی های افتراقی بیشتر را افزایش می دهد ، بلکه سرعت **blind firing** را نیز کاهش می دهد.

همانطور که قبلاً ذکر شد ، فعالیت آخرین لایه ، به ویژه اولین **spike** در لایه **C2** ، تنها اطلاعاتی است که شبکه ما برای تصمیم گیری نهایی در مورد محرک های ورودی استفاده می کند. به این ترتیب ، ما به طبقه بندی کننده های خارجی احتیاج نداریم و همزمان قابلیت زیست شناختی شبکه را افزایش می دهیم.