

## گزارش تمرین 4

سحر برکاتی 96222021

یکی از رویکردهای STDP،  $R - STDP$  است که برخلاف STDP که شباهت ها و ویژگی هایی که به طور مداوم تکرار میشد را یاد میگرفت، در اینجا تفاوت بین دو ویژگی را یاد میگیریم. پس  $R - STDP$  عملکرد بهتری نسبت به STDP دارد. هر دو ویژگی feature extraction و classification که با spike ها انجام میشد، اینبار فقط با حداکثر یک spike در هر نورون رخ میدهد. و بعلاوه چنین شبکه ای مصرف انرژی کمتری دارد. نورون ها در مغز به وسیله یسناپس هایی بهانعطافم متصل اند، که در طول زمان یا قوی تر میشوند یا ضعیف. STDP نوعی انعطاف پذیری است که در سطوح بالا مغز انجام میشود، و در سیستم بینایی با در نظر گرفتن اختلاف زمانی بین spike زدن دو نورون pre - post synaptic کار میکند. میدانیم که STDP با پیدا کردن شباهت ها و ویژگی هایی که به طور مداوم تکرار میشد یک spike تولید میکرد و باعث میشد هرگاه آن الگو را مشاهده کرد پتانسیلش افزایش پیدا کند.

STDP در پیدا کردن ویژگی های مشابه، عملکرد خوبی دارد اما به عنوان یک الگوریتم *unsupervised learnin* در تشخیص ویژگی های کم تکرار تر با مشکل رو به رو است.

سیستم reward دادن در مغز روی عملکرد تصمیم گیری و شکل گیری رفتار تاثیر زیادی دارد و به عنوان reinforcement learning شناخته شده و کاری که میکند این است که افراد را ترغیب میکند که کارهایی که بابت آنها reward دریافت میکنند را تکرار کند و کارهایی که در نتیجه آن punishment اتفاق افتاده، تکرار نشود. دوپامین در مغز یک ماده شیمیایی است که در سیستم reward نقش بسیار مهمی دارد. دوپامین روی تغییرات قطبی و پنجره زمانی STDP تاثیر میگذارد.

$R - STDP$ ، نشانی سیناپس هایی که شرایط STDP را دارند ذخیره میکند و تغییرات وزن را در زمان دریافت سیگنال اعمال میکند که یا سیگنال reward است یا سیگنال punishment.

همچنین نشان داده شد که،  $R - STDP$  با کدگذاری ورودی rate - input coding و نرخ firing مسئله XOR را حل کند.

گفتیم که STDP تمایل دارد ویژگی های مشابه یا مکرر را پیدا کند هرچند که ممکن است نیازی نباشد. در این جا یک شبکه SNN که دارای  $R - STDP$  است را برای تشخیص اشیا طبیعی بدون طبقه بندی خارجی در نظر میگیریم. به جای آن ما یک سری نورون ها را مخصوص تشخیص کلاس مورد نظر تعیین میکنیم که با دیدن شی مورد نظر یک spike بزنند.  $R - STDP$  به شبکه ما کمک میکند که ویژگی مورد نظرش پیدا کند و بدین ترتیب مصرف انرژی کاهش میابد. شبکه ما دارای چهار لایه است.

لایه اول، براساس لبه ها، تصویر ورودی را به تاخیری spike ای تبدیل میکند. روی این قطار spike ها در لایه دوم عملیات pooling اعمال میشود. در لایه سوم نورون های integrate and fire وجود دارند، که اطلاعات دریافتی مربوط به لبه ها و زوایا را دریافت میکند و بر اساس آنها ویژگی پیچیده ای را میسازد. این لایه تنها لایه ای است که در آن learning در آن رخ میدهد و از  $R - STDP$  استفاده میشود. سیگنال های reward و punishment در لایه چهارم فرستاده میشود که شبکه ما بر اساس آن تصمیم گیری میکند. شبکه ما فقط با اولین spike ای که در لایه سوم زده میشود، بدون استفاده از classification تصمیم گیری میکند. و همانطور که گفته شد اگر تصمیم خوبی باشد یک سیگنال reward و اگر تصمیم خوبی نباشد یه سیگنال punishment تولید میشود. برای این که شبکه از نظر محاسباتی بهتر عمل کند، هر نورون فقط یکبار کل تصویر را میبند.

برای نشان دادن کارایی  $R - STDP$ ، آزمایش اسباب بازی را انجام دادیم. میبینیم که  $R - STDP$  با این که منابع

آموزشی کمتری دارد اما نسبت به STDP عملکرد بهتری دارد. بعلاوه این که STDP – R میتواند با فرستادن سیگنال reward رفتار یک نورون را تغییر دهد.

حالا میخواهیم هر لایه را جداگانه بررسی کنیم:

- لایه اول: لایه S1 ساده ترین لایه است، نورون های این لایه لبه ها و خطوط را با زوایا مختلف را در تصویر خاکستری با مقیاسی که به آن تبدیل شده تشخیص میدهد و متناسب با آنها با تاخیر spike تولید میکند. برای همین اینجا از فیلترهای gabor با چهار جهت متفاوت باهم ترکیب میشوند. پس این لایه شامل 4 صفحه است که هر کدام یک زاویه را تشخیص میدهند.
- لایه دوم: لایه C1، این لایه پیچیده است. در اینجا 4 شبکه عصبی دو بعدی داریم که مربوط به یکی از جهت گیری های لایه قبلی است. هر نورون در این لایه عملیات pooling را در پنجره هایی به اندازه  $w_{c1} \times w_{c1}$  و stride ای برابر با  $r_{c1}$  روی نورون های S1 انجام میدهد و بلافاصله با دریافت اولین spike یک spike تولید میکند. عملیات pooling باعث کاهش نورون ها در لایه S1 میشود. یک local invariance به لبه های زاویه دار اضافه میشود.
- لایه سوم: لایه S2، ویژگی های ورودی از لایه های قبل را باهم ترکیب کرده و یک ویژگی پیچیده معنی دار تولید میکند. همانطور که نورون ها fire میکنند، وزن های سیناپسی براساس ترتیب spike نورون pre – post synaptic و همچنین بر اساس سیگنال های reward و punishment، update میشوند. این سیگنال از فعالیت لایه بعدی گرفته میشود که نشان دهنده تصمیم شبکه است. وزن اولیه به صورت تصادفی و براساس میانگین 0.8 و انحراف از معیار 0.05 انتخاب شده.
- لایه آخر (چهارم): لایه C2، این لایه شامل n نورون میشود. و هر کدام به نوعی به یکی از شبکه های عصبی S2 اختصاص میابد. نورون C2 فقط اولین spike دریافتی از شبکه عصبی مربوط به آن دریافت کرده است را گسترش میدهد. فعالیت های این لایه نمایانگر تصمیم گرفته شده است. برای این منظور، نورون های این لایه به چند گروه تقسیم میشوند و هر گروه را به دسته خاصی از محرک های ورودی اختصاص میدهیم. فرض بر این است که تصمیم شبکه درباره گروه محرک ورودی، همان گروهی باشد که اولین spike را در لایه C2 گسترش میدهد.

ما یک reinforcement learning داریم، که از آن برای update کردن وزن نورون های pre synaptic لایه S2 استفاده میکنیم. مقدار تغییر وزن توسط سیگنال های reward و punishment انجام میشود که این سیگنال ها براساس درستی یا نادرستی تصمیم شبکه است. ما از یک برنده استفاده میکنیم تا یادگیری را در بین تمام نورون های S2 انجام دهد که در نتیجه این کار یکی از اولین spike ها برنده است و میتواند وزن های سیناپسی خود را update کند. و این نورون یکی از نورون های تصمیم گیرنده شبکه است.

احتمال گیر افتادن در local optima ها یا overfitting بر اثر کسب بیشترین مقدار reward وجود دارد. ما برای جلوگیری از این اتفاق از دو تکنیک برای تشخیص اشیا استفاده میکنیم:

#### 1- Adaptive learning rate:

از آنجایی که وزن های اولیه در ابتدا به صورت تصادفی تعیین میشوند اشتباهات در ابتدا آموزش زیاد است. یعنی شانس است. با ادامه آزمایشات نسبت نمونه های طبقه بندی درست بیشتر از نمونه های طبقه بندی غلط میشود. در صورتی که نرخ طبقه بندی غلط بیشتر باشد، شبکه سیگنال punishment بیشتری دریافت میکند، که در نتیجه آن وزن های سیناپس ها کاهش می یابد. و همچنین در صورتی که نرخ طبقه بندی درست بیشتر باشد، شبکه سیگنال reward بیشتری دریافت میکند، که در نتیجه آن وزن های سیناپسی تقویت میشوند. در

این حالت شبکه سعی میکند طبقه بندی های نادرست را انجام ندهد و در نتیجه آن سیگنال punishment دریافت نکند و برای بقیه در حالت silent باقی بمانند. در دو حالت overfitting به دلیل بالانس نبودن reward و punishment ها رخ میدهد. برای حل این مشکل، ما یک عامل adjustment را برای اصلاح مقدار وزن ضرب میکنیم، که توسط آن تاثیر نمونه های درست و نادرست آموزشی متعادل میشود.

-2 Drop out :

در یک reinforcement learning، هدف این است reward ها بیشترین حالت را داشته باشند. از آنجایی که شبکه ما فقط داده های آموزشی را میبیند، ممکن است ویژگی هایی را پیدا کنند که برای طبقه بندی همه نمونه های آزمایشی کافی است. بنظر میرسد که این مسئله در مقایسه با مشکلات پیچیده، دچار overfitting شده و شبکه ترجیح میدهد که برخی از نورون ها را آموزش ندهد. ما برای حل این مشکل از drop out استفاده میکنیم ف که باعث میشود یک نورون C2 با احتمال P، موقتا خاموش شود. این تکنیک کمک میکند که هم شان پیدا کردن ویژگی های نامشابه بیشتر شود، و نرخ blind firing را کاهش میدهد.

همانطور که دیدیم، فعالیت آخرین لایه به ویژه اولین spike در لایه C2، تنها اطلاعاتی است که شبکه ما برای تصمیم گیری نهایی در مورد ورودی استفاده میکند.

پس بدین ترتیب نه تنها به طبقه بندی خارجی نیاز نداریم بلکه، همزمان قابلیت تطبیق پذیر بیولوژیکی شبکه را هم افزایش میدهیم.

منبع استفاده شده :

First – spike based visual categorization using reward – modulated STDP