

به نام خدا

پروژه نهایی علوم اعصاب محاسباتی

شبکه‌های بازگشتی اسپایکی

طبقه‌بندی متون اخبار در دسته‌های خبری

بازنمایی لغات به کمک شبکه‌های عصبی اسپایکی

بهزاد شایق بروجنی

610098032

فهرست مطالب

2	فهرست مطالب
3	شرح مسئله و توضیحات روند گزارش
4	جمعیت‌ها
4	ابزار
4	مسئله طبقه‌بندی: Logistic Regression
5	ساخت بازنمایی متن: فرکانس spike
5	ورودی شبکه: one-hot encoder
5	اتصالات
6	همزمانی نورون‌های جمعیت ادراک: کم‌پشت‌سازی اتصال ورودی به ادراک
6	بازخورد مثبت کنترل نشده نورون‌های جمعیت ادراک: مقداردهی اولیه‌ی وزن‌ها
6	خروجی یکسان: اقدام به آموزش بازنمایی لغات
7	اتصالات مسئله جایگزین
7	همزمانی نورون‌های جمعیت ادراک: k-winners-take-all
8	انحصار فعالیت جمعیت ادراک توسط بعضی از نورون‌ها: constant-summation-of-linear-coefficients
8	انحصار فعالیت جمعیت ادراک توسط بعضی از لغات: constant-summation-of-axons-utilizations
8	عدم فعالیت جمعیت ادراک: k-random-clamps
9	الهام از روش‌های سنتی: استفاده از Flat STDP و قدرمطلق std
10	نمونه نتایج
10	حدس عامل شکست
11	کدها

شرح مسئله و توضیحات روند گزارش

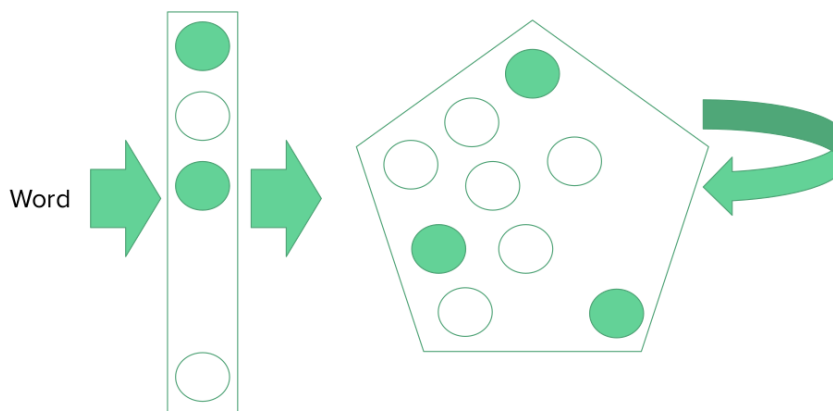
هدف اصلی از این آزمایش، شبیه‌سازی شبکه‌های عصبی بازگشتی به وسیله شبکه‌های عصبی اسپایکی برای در نظر گرفتن ترتیب مشاهده کلمات در مسائل پردازش متن بود. از همین بابت، مسئله‌ای که به عنوان ارزیابی برونی مدل، در صدد حل آن برآمدیم، مسئله‌ی طبقه‌بندی اخبار در دسته‌های خبری بود. از دادگانی شامل ۳ دسته خبری استفاده کرده و به کمک «خلاصه خبر» سعی بر پیشبینی دسته خبری کردیم. در طول این شبیه‌سازی با مسئله‌ی ضعف بازنمایی لغات روبرو شدیم که یادگیری شبکه را مختل می‌کرد. بنابراین در میانه راه اقدام به تغییر مسئله به آموزش بازنمایی لغات به کمک شبکه‌های عصبی اسپایکی کردیم.

تمام اقدامات انجام شده با هدف حل مسائل بالا منجر به شکست شد که اقدامات و نتایج را با هم بررسی می‌کنیم. به دلیل ناموفق بودن اقدامات، همه‌ی نتایج ثبت نشده‌اند و در نتیجه در گزارش آورده نخواهند شد. اما نتایج مشاهده شده توصیف می‌شوند.

به دلیل فرصت کوتاه انجام آزمایش و محدودیت سرویس GPU در بستر google colab، حل مسئله دشوارتر نیز می‌شد. بنابراین، علی‌رغم تعداد زیاد موارد آزمایش شده، تعدادی راهکار احتمالی آزمایش نشده باقیمانده که در انتهای گزارش آورده خواهد شد.

ساختار گزارش به شکل تاریخچه‌ای خواهد بود و **مهم‌ترین** آزمایش‌های انجام شده، مشاهدات، حدس‌ها و فرضیات از ریشه مشکل و راهکار ارائه شده به ترتیب زمان آورده خواهند شد.

جمعیت‌ها



شکل فوق ساختار کلی شبکه مورد بررسی را نشان می‌دهد. جمعیت‌های نورونی را به ترتیب از چپ به راست به عنوان جمعیت ورودی و جمعیت ادراک می‌شناسیم. نورون‌های جمعیت ادراک از نوع LIF هستند. اندازه جمعیت‌ها به ترتیب حدود ۱۱ هزار و ۱۱۰ می‌باشد.

ابزار

برای طبقه‌بندی در انتهای مسئله از کتابخانه‌ی sikitlearn و برای ساخت شبکه‌ی اسپایکی از کتابخانه‌ی شخصی خود یعنی spiral استفاده شد. ماژول‌های خاص این پروژه نیز در این کتابخانه پیاده‌سازی و قابل استفاده شده است.

مسئله طبقه‌بندی: Logistic Regression

با یک مسئله‌ی طبقه‌بندی متن به عنوان معیار برونی ارزیابی روبرو هستیم. هدف اصلی مدل اسپایکی، ارائه یک بازنمایی از متن خبر با هدف طبقه‌بندی است. پس از آنکه شبکه این بردار را مهیا کرد، اقدام به طبقه‌بندی بردارها به کمک مدل Logistic Regression خواهیم کرد.

ساخت بازنمایی متن: فرکانس spike

مطلوب است بازنمایی ارائه شده به عنوان خروجی شبکه اسپایکی دارای ابعاد ثابتی باشد. با این منطق که هر نورون از جمعیتی که فهم شبکه را مدل می‌کند، ناجی یک مفهوم طبیعی است و هر بار spike زدن این نورون حاکی از نزدیکی کلیت متن به آن مفهوم می‌باشد، فرکانس اسپایک نورون‌های جمعیت اصلی شبکه را به عنوان بازنمایی متن خروجی دادیم. با توجه به اینکه جملات با طول متفاوت به شبکه ورودی داده می‌شوند، برای محاسبه‌ی این فرکانس، تعداد اسپایک‌های هر نورون در طول مشاهده یک خبر را به طول آن خبر تقسیم کردیم. بنابراین بازنمایی ما یک ماتریس با ابعاد ثابت با مقادیر بین صفر و یک خواهد بود.

ورودی شبکه: one-hot encoder

برای تغذیه شبکه از لغات، از یک one-hot encoder استفاده شد که یک جمعیت به ابعاد لغتنامه در ابتدای شبکه قرار داده و با مشاهده هر لغت، یک نورون مختص به آن لغت فعال خواهد شد. برای کاهش ابعاد مسئله، پیشپردازش‌هایی بر روی متون انجام شده و تنوع لغات و در نتیجه ابعاد لایه‌ای encoder کاهش پیدا کرده است. در طول مشاهده هر متن خبری، کلمات و علائم نگارش متن خبر، هر کدام در یک گام زمانی و به ترتیب وارد مدل خواهند شد. برای پشتیبانی از batch training، برای یکسان‌سازی ابعاد ورودی، یک کلمه <PAD> به انتهای متون اضافه می‌شود که در صورتی که در یک batch، یکی از اخبار کوتاه‌تر باشد، مادامی که دیگر اخبار هنوز در حال بررسی هستند، از طرف این خبر، فقط لغت <PAD> به شبکه وارد خواهد شد. این لغت خاص باعث می‌شود هیچ یک از نورون‌های one-hot encoder فعالیتی از خود نشان ندهند.

اتصالات

در شروع آزمایش‌ها، دو اتصال در شبکه برقرار کردیم. اتصال اول برای متصل کردن one-hot encoder به جمعیت ادراک و اتصال دوم برای پیوندهای درونی جمعیت ادراک. **در ابتدا** فقط اقدام به آموزش اتصال درونی

جمعیت ادراک به کمک قانون یادگیری STDP کردیم و اتصال اول را به صورت تصادفی برقرار کرده و ثابت نگه داشتیم. این شرایط در ادامه تغییر خواهد کرد.

همزمانی نورون‌های جمعیت ادراک: کمپشت‌سازی اتصال ورودی به ادراک

در آزمایش‌های اول مشاهده شد که نورون‌های جمعیت ادراک فاقد هیچ فعالیت هستند. برای حل این مشکل، اتصال بین جمعیت ورودی با جمعیت ادراک را تقویت کردیم. این کار باعث شد همه‌ی نورون‌های جمعیت ادراک به صورت همزمان و مکرر فعالیت کنند. برای حل این مسئله، از تعدد اتصالات موجود بین جمعیت ورودی و ادراک (که پیش‌تر fully connected بود) کاستیم و این مشکل حل شد. نرخ این اتصالات را 0.1 قرار دادیم.

بازخورد مثبت کنترل نشده نورون‌های جمعیت ادراک: مقداردهی اولیه‌ی وزن‌ها

در ادامه با این مشکل روبرو شدیم که در طول مشاهده‌ی یک خبر، فعالیت جمعیت ادراک به سرعت افزایش یافته و به حداکثر مقدار ممکن می‌رسید و در طول مشاهده خبر ثابت می‌ماند. ریشه‌یابی ما از این مشاهده، بازخورد (feedback) مثبت این جمعیت به خودش بود. بنابراین تلاش کردیم فعالیت ابتدایی این جمعیت را به نحوی تغییر دهیم که با leakage مدل نورونی سازگار باشد. این کار با مقداردهی اولیه‌ی وزن‌های اتصال درونی این جمعیت ممکن و موفق بود.

خروجی یکسان: اقدام به آموزش بازنمایی لغات

پس از حل مشکلات مطرح شده در بالا، مشاهده کردیم که خروجی مدل به ازای تمامی اخبار یکسان است. حدس ما از این مشاهده، سادگی بیش اندازه برخورد ما با لغات متفاوت بود. ما ۱۱ هزار لغت متنوع را به ۱۱۰ نورون ترجمه می‌کردیم. این ترجمه به وسیله‌ی اتصالاتی تصادفی با نرخ 0.1 صورت می‌گرفت. به این صورت ما میزان زیادی از مفاهیم لغات را از دست می‌دادیم. بنابراین مشابه آن بود که ورودی تصادفی را آموزش دهیم و در

نتیجه، در ابعاد بزرگ، نتایج همگرا می‌شدند. بنابراین احساس شد به یک بازنمایی بهتر از لغات نیاز داریم که قابل اتکا باشد. مشکل اصلی این بود که چنین بازنمایی وجود ندارد. بنابراین اقدام به تغییر مسئله به آموزش بازنمایی لغات کردیم.

اتصالات مسئله جایگزین

هدف از بازنمایی لغات آن است که لغات مشابه بازنمایی مشابهی داشته باشند. لغاتی مشابه هستند که در محیط‌های یکسانی مشاهده شوند (نقل از مقالات مطرح این زمینه). بنابراین نیاز بود تا به نحوی لغاتی که در کنار هم مشاهده می‌شوند مشابه در نظر گرفته شوند. برای این منظور، باز هم نورون‌های جمعیت ادراک را نمایان‌کننده‌ی مفاهیم در نظر گرفتیم اما این بار درون آن‌ها اتصالی برقرار نکردیم تا آموزش مختص به بازنمایی لغات باقی بماند. این بار اتصال بین جمعیت ورودی و جمعیت ادراک را به کمک STDP آموزش دادیم. در این بخش، ابتدا وزن‌های سیناپسی همه اتصالات را کوچک انتخاب کرده و std را نیز غیر فعال کردیم تا به مرور، مشاهده ترکیب‌های تکراری موجب تقویت آن‌ها شود.

همزمانی نورون‌های جمعیت ادراک: k-winners-take-all

با اقدام به آموزش اتصالات بین جمعیت ورودی و ادراک، دوباره با مسئله همزمانی نورون‌های جمعیت ادراک روبرو شدیم. البته این بار در ابتدا بخشی از نورون‌ها همزمانی داشتند اما با ادامه پیدا کردن آموزش، کل جمعیت یک‌دست می‌شد. اینبار ناچار به استفاده از اتصالات کامل (dense) بودیم. بنابراین راهکار قبلی چاره‌ساز نبود. بنابراین اقدام به استفاده از تکنیک k-winners-take-all کردیم تا تعداد نورون‌های فعال در جمعیت ادراک را در هر گام زمانی محدود کرده و به مدل کمک کند تا فعالیت خود را بین نورون‌ها تقسیم کند. این کار برای حل این مشکل موفق بود اما کمکی به بازدهی نکرد.

انحصار فعالیت جمعیت ادراک توسط بعضی از نورون‌ها: constant-summation-of-linear-coefficients

مشکل بعدی که با آن روبرو شدیم این بود که تعداد محدودی از نورون‌های جمعیت ادراک همیشه فعال بوده و با توجه به قانون یادگیری STDP باز هم بیشتر تقویت می‌شدند. همچنین با توجه به تکنیک k-winners-take-all مانع تقویت دیگر نورون‌ها می‌شدند. برای حل این مشکل، تکنیکی ارائه دادیم (از وجود آن اطلاعی ندارم، حاصل ابتکار برای حل مسئله بود) که طی آن، مدل را مجبور می‌کنیم وزن‌های سیناپسی را به نحوی تغییر دهد که مجموع وزن‌های سیناپسی هر نورون post-synaptic مقدار ثابتی بماند. به این شکل، یک نورون اگر توسط چند لغت ورودی تقویت شود دیگر نمی‌تواند برای دیگر لغات نیز فعالیت نشان دهد. با استفاده از این تکنیک، مشکل مطرح شده حل شد اما بازدهی مدل همچنان پایین بود.

انحصار فعالیت جمعیت ادراک توسط بعضی از لغات: constant-summation-of-axons-utilizations

مشکل بعدی مشابه مشکل قبلی بود. اینبار همه‌ی وزن‌های سیناپسی برای تعداد محدودی از لغات آموزش داده می‌شد. این اتفاق دور از انتظار نبود. مشکل stop words در مسائل پردازش متن یک مشکل رایج است. راهکار ساده حذف این لغات بود چرا که اطلاعات غنی ندارند. اما این کار مانع استفاده از گرامر در آینده می‌شود. بنابراین از این کار پرهیز کردیم. برای حل این مشکل، راهکاری مانند راهکار قبلی ارائه دادیم. این بار مدل را اجبار کردیم که اثری که هر لغت بر جمعیت ادراک می‌گذارد را به مقدار ثابتی محدود کند. این کار نیز مشکل را حل کرد اما بازدهی مدل همچنان پایین بود.

عدم فعالیت جمعیت ادراک: k-random-clamps

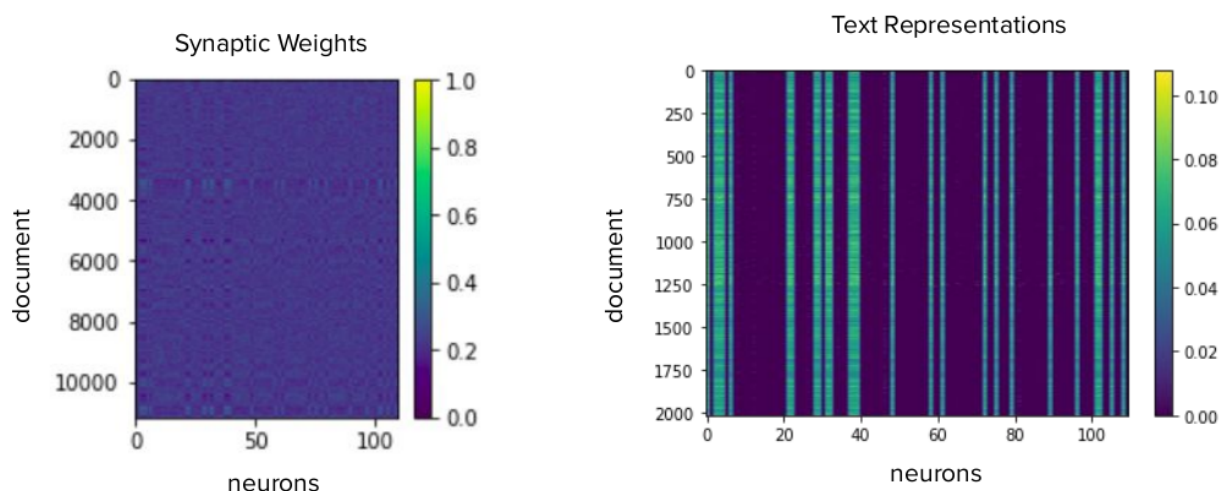
پیش‌تر اشاره کردیم که جمعیت ادراک فاقد هیچ فعالیتی بود و برای حل این مشکل، اتصالات جمعیت ورودی به ادراک را تقویت کردیم. این تقویت بیش از اندازه بزرگ بود چرا که در ابتدای آموزش، وزن‌های سیناپسی مقادیر بسیار کوچکی داشته و نیازمند ضریب بزرگی برای فعال کردن نورون post-synaptic داشتند. این ضریب بزرگ باعث می‌شد پس از پیشروی کوتاهی در آموزش، بعد از آن که وزن‌های سیناپسی مقادیر بزرگی اتخاذ می‌کردند،

فعالیت مدل را مختل کند. برای حل این مشکل، ضریب مذکور را کاهش داده و این بار برای حل عدم فعالیت جمعیت ادراک، تکنیکی ارائه کردیم که مدل را در انتهای مشاهده هر خبر مجبور می‌کند حداقل k نورون از نورون‌های ادراک را با فعال کند. انتخاب این k نورون تصادفی و مبتنی بر احتمالاتی محاسبه شده به کمک پتانسیل نورون‌ها است. به این شکل مشکل عدم فعالیت در ابتدای مسئله حل شده و به مرور، با افزایش وزن‌های سیناپسی مشکل عدم فعالیت به صورت کلی حل می‌شود.

الهام از روش‌های سنتی: استفاده از Flat STDP و قدرمطلق std

پس از آنکه با وجود تمام فعالیت‌های بالا، مدل همچنان قادر به یافتن بازنمایی مناسب از لغات نبود، اقدام به شبیه‌سازی روش‌های قدیمی کردیم. یکی از روش‌های ساده که تا حدی پاسخگو است آن است که لغاتی که در یک سند داده ظاهر می‌شوند را به هم نزدیک می‌کند. به همین منظور، با استفاده از Flat STDP، تکنیک k -random-clamps با $k=1$ و همچنین استفاده از std با ضریب -1 برای از بین بردن اهمیت ترتیب حضور لغات، تلاش کردیم اتصال بین یک نورون از جمعیت ادراک را به تمام لغات مشاهده شده در یک سند تقویت کنیم. این کار در نهایت منجر به افزایش بازدهی نشد. حدس ما از علت این امر، محدودیت منابع بود که ما را مجبور به استفاده از گام‌های آموزشی بزرگ کرده بود.

نمونه نتایج



دو نمودار بالا یکی نمونه‌ای از عملکرد مدل (بدون استفاده از دو بخش آخر ارائه شده) است. مشاهده می‌کنیم که تلاش ما برای استفاده توزیع‌شده از وزن‌های سیناپسی **موفق** بوده اما خروجی مطلوبی نداشتیم. بازدهی مدل در امر طبقه‌بندی دادگان در سه دسته، به **۵۷.۷۵ درصد** رسید. این بازدهی در صورت استفاده از بازنمایی‌های تصادفی حدود **۴۹ درصد** می‌باشد که حاکی از شکست ما در حل این مسئله است.

حدس عامل شکست

ابعاد این مسئله بزرگ است و همچنین ابعاد و جنبه‌های متنوعی دارد که باعث می‌شود تفکر درباره آن را دشوار کند. به همین دلیل احتمال آنکه تصمیمات اشتباهی گرفته شده باشد زیاد است. یکی از مشکلات اصلی که باعث کندی پیشرفت در این زمینه می‌شود نبود مدل‌های موجه از نظر بیولوژیکی است که ما را به آزمون و خطا وادار می‌کند. این امر باعث می‌شود سرعت نزدیک شدن به جواب در مسائل پردازش متن بسیار کم باشد. همچنین به دلیل ابعاد بزرگ مسئله، زمان محدود و محدودیت سرویس GPU بستر google colab باعث شد فرایند آموزش را بسیار محدود کنیم به نحوی که آموزش بردارهای بازنمایی را به کمک تنها ۱/۱۶ دادگان و آموزش طبقه‌بند نهایی را به کمک تنها ۱/۱۰ دادگان انجام دادیم. همچنین ناچار به انتخاب **گام‌های آموزشی بسیار بزرگ** بودیم که باعث می‌شود تفاوت یکبار مشاهده دو لغت در کنار هم با چندین بار مشاهده دو لغت در کنار هم از بین برود.

کدها

مدل‌ها، encoder و استراتژی‌های مطرح شده همگی در [کتابخانه spiral](#) پیاده‌سازی شده‌اند. اسکرپیت پیش‌پردازش و اسکرپیت مادر که بستر آزمایش‌های بازگو شده بود نیز ضمیمه شده‌اند. توجه کنید که اسکرپیت مادر کلیت کد را شامل می‌شود و برای آزمون‌های جزئی، نیاز به تغییرات جزئی دارد. به دلیل تعدد زیاد آزمایش‌ها، همه آن‌ها ضمیمه نمی‌شوند.