



دانشکده مهندسی برق

گزارش کار پروژه

نام درس

علوم اعصاب محاسباتی

نام دانشجو

پویا احمدپور

استاد:

دکتر محمدرضا دلیری

تیر ماه 1401

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

چکیده

در این مطلب به پیش‌بینی پاسخ نوروئی یک دسته از نوروئ‌های ناحیه‌ی قشری میانی تمپورال دو میمون می‌پردازیم می‌پردازیم. که با توجه به نقطه‌ی تمرکز بینایی خود، به یک سری از محرک‌های بینایی در پس زمینه پاسخ می‌دهند. داده‌های مرتبط، از 41 نوروئ هستند که پیش‌تر تفکیک و زمان رخداد پتانسیل عمل (اسپایک) در آن‌ها مشخص شده است. پاسخ این نوروئ‌ها همگام با نمایش محرک‌ها در پس زمینه، و در زمان‌های آزمایش (تریال)، 3 ثانیه‌ای، ضبط شده‌اند. که در بین این تریال‌ها، یک بار به صورت رندوم، نقطه‌ی تمرکز میمون‌ها تغییر پیدا می‌کند. از آنجا که با تغییر این نقطه، نوع پاسخ نوروئ‌ها به محرک‌های بینایی تغییر می‌کند و عملکرد آن‌ها متفاوت خواهد بود، در این پروژه، از دو طبقه‌بند (کلاسیفایر) استفاده کردیم، که همزمان، با تحلیل جداگانه‌ی پاسخ‌های پیش از تغییر نقطه‌ی تمرکز و پاسخ‌های بعد از آن، و ترکیب آن‌ها، پیش‌بینی دقیق‌تری از پاسخ نوروئی، با توجه به محرک‌ها ارائه خواهند داد.

واژه‌های کلیدی: نوروئ‌های ناحیه‌ی قشری بینایی، کدینگ، محرک‌های بینایی، پاسخ نوروئی،

کلاس‌بندی

فهرست مطالب

1	فصل 1: مقدمه
2	1-1- مقدمه
	Error! Bookmark not defined.
	فصل 2: مروري بر منابع
	2-1- کارهای پیشین
	Error! Bookmark not defined.
5	فصل 3: روش تحقیق
6	3-1- مقدمه
6	3-2- درباره‌ی داده‌ها
7	3-3- پیش پردازش
14	3-5- انتخاب ویژگی
15	3-6- ارزیابی و طبقه‌بندی
17	فصل 4: نتایج و تفسیر آنها
18	4-1- بحث و نتیجه‌گیری
21	فصل 5: جمع‌بندی و پیشنهادها
22	5-1- مقدمه
22	5-2- محتوا
22	5-2-1- جمع‌بندی
23	5-2-2- نوآوری
23	5-2-3- پیشنهادها
25	مراجع
26	پیوست‌ها

فهرست علائم اختصاري

فصل 1:

مقدمه

1-1- مقدمه

بررسی عملکرد نورون‌ها در مغز و خارج از آن، موضوع جدیدی است که در حوزه‌ی علوم اعصاب، عصب‌شناسی، پزشکی و مهندسی پزشکی، بسیار مورد توجه واقع شده است. در این بسته از داده‌ها (دیتاست)، پاسخ نورونی مغز دو میمون به یک سری رندوم از محرک‌های بینایی، در طی چند صد ترایال، ضبط شده است. در طول هر ترایال یک بار به صورت تصادفی (رندوم)، نقطه‌ی تمرکز بینایی برای میمون تغییر می‌کند. با توجه به این مسئله که، نورون‌ها پاسخ انتخابی به محرک‌های بینایی با توجه به مکان قرار گرفتنشان در ناحیه‌ی گیرندگی (receptive filed) نشان می‌دهند و همچنین ماهیت تصادفی و غیرایستای پاسخ نورونی به محرک‌های مشابه، می‌توان دریافت که آپدیت دائم کلاسیفایر با دیتای جدید، موجب بهبود عملکرد تشخیص و پیش‌بینی آن خواهد شد. در اینجا به شکلی ساده‌تر به استفاده از دو کلاسیفایر می‌پردازیم که یکی با داده‌های پیش از تغییر نقطه‌ی تمرکز بینایی و یکی با داده‌های پس از آن آموزش داده شده است. علی‌رغم پایین آمدن سرعت آموزش مدل در این رویکرد، دقت کلاسیفایر در پیش‌بینی پاسخ نهایی، پیشرفت قابل توجهی داشته است.

فصل 2:

مروری بر منابع

1-2- کارهای پیشین

در مقاله‌ی مرتبط با این ست از داده‌ها، طراحی مدلی پیشنهاد شده است، که با توجه به رندوم بودن محرک‌ها و تغییر شبه رندوم Saccade point در طول آزمایش و تغییر جایگاه محرک‌ها در Receptive Field نوروها، بتواند، همچنان خود را آپدیت کرده و پیش‌بینی قابل قبولی را از پاسخ نرونی ارائه دهد. در این مقاله یک مدل تغییر شکل یافته و اختصاص یافته از مدل‌های خطی ایجاد شده که با زمان تغییر کند. نام این مدل Sparse-variable Generalized linear model می‌باشد. که به سیستم اجازه می‌دهد با رزولوشن در حد میلی ثانیه خود را با تغییرات ورودی به روزرسانی کند.

فصل 3:

مواد و روش ها

1-3- مقدمه

در این قسمت به توضیح و شرح داده‌ها، روش‌های استفاده شده برای پردازش و پرداخت داده، نحوه و نوع ویژگی‌های استخراج شده از سیگنال ورودی و بررسی روش‌های طبقه‌بندی و پیش‌بینی داده‌ها خواهیم پرداخت.

2-3- درباره‌ی داده‌ها

همانطور که پیشتر اشاره شد، داده‌ها از 41 نورون و توسط یک دسته الکتروود 16 کاناله، ضبط شده‌اند که در طی یک آزمایش تاثیرات تغییر نقطه‌ی تمرکز بینایی بدست آمده‌اند. میمون‌های مورد آزمایش پیش‌تر آموزش داده شده‌اند تا به یک نقطه‌ی مشخص خیره شوند. این نقطه جایی است که اولین نقطه‌ی تمرکز ظاهر می‌شود و میمون‌ها تا زمان تغییر نقطه‌ی تمرکز (Saccade point) نباید، به چشم خود حرکت دهند. پس از طی شدن یک زمان رندوم بین 0 تا 3 ثانیه، ک مدت زمان هر تریال می‌باشد، نقطه‌ی تمرکز در چند درجه فاصله از نقطه‌ی ابتدایی، مجدداً ظاهر می‌شود. در طی این تریال، 81 محرک بینایی در قالب مربع‌های سفید رنگ، در پس زمینه‌ای سیاه، به میمون نمایش داده می‌شوند و پاسخ نوروئی مربوط به آن ضبط می‌شود.

1-2-3- مروری بر جزئیات ویژگی داده‌ها

دیتاست فوق‌الذکر به صورت دسته‌بندی شده در چند فایل، به فرمت متلب قرار گرفته‌اند که در زیر، به آن‌ها اشاره خواهیم کرد.

Stimcode:

پیش‌تر اشاره شد که 81 محرک در پس زمینه در طول یک تریال نمایش داده می‌شوند. این 81 محرک یک صفحه‌ی سیاه 9×9 را پوشش می‌دهند. در نتیجه 81 مکان (لوکیشن) مختلف برای محرک قابل تعریف است که در طول تریال با یک ترتیب شبه رندوم، نمایش داده می‌شوند. به طوری که در طول یک تریال هر 81 محرک نمایش داده شده و هیچگاه دو محرک همزمان نشوند. Stimcode یک ماتریس $T \times N$ می‌باشد

که مکان نمایش محرک در هر لحظه در طول ترايال را نشان می دهد. در این ماتریس، T ، زمان و طول ترايال است که 3000 میلی ثانیه می باشد و N ، تعداد ترايال هایی که از نورو ن مربوطه ضبط صورت گرفته است.

:Conds

همچنان که محرک ها با یک ترکیب شبه رندوم در طول ترايال نمایش داده می شوند، این ترکیب نمایش را با نام Condition، نام گذاری کرده اند. ماتریس conds یک ماتریس $N \times 1$ ، شامل condition مربوط به هر ترايال می باشد.

:Resp

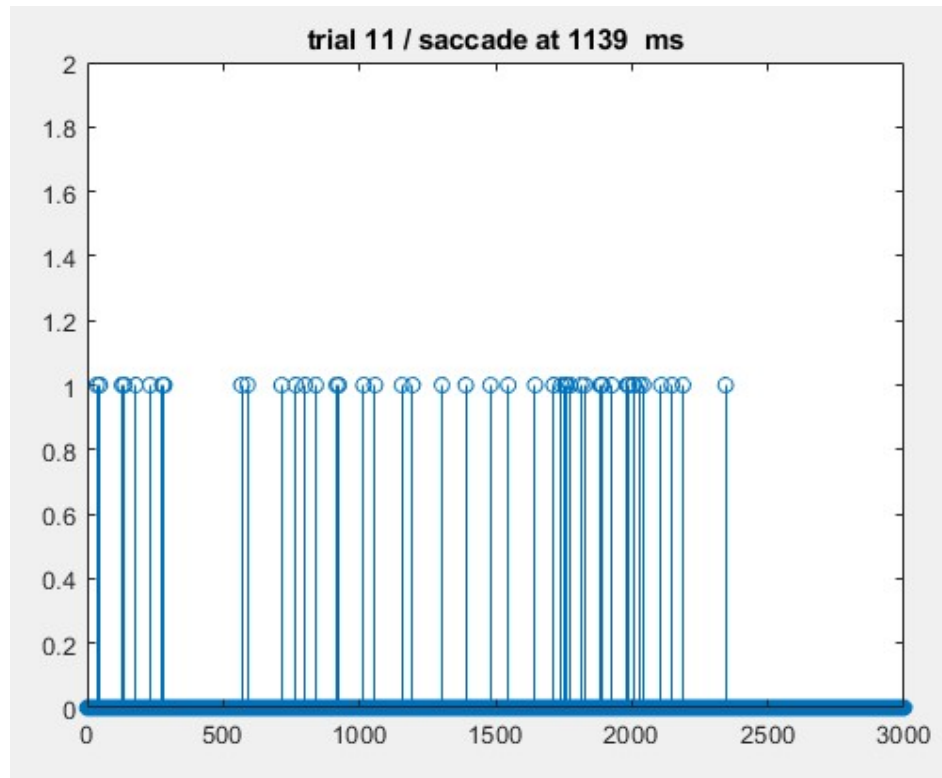
این ماتریس فعالیت اسپایکی نورو ن را در طول ترايال به ما نشان می دهد. در این ماتریس $N \times T$ ، در هر ترايال، و در هر زمان، اسپایک نورو ن را با 1 و عدم وجود فعالیت اسپایکی را با 0 مشخص کرده اند.

:Tsaccade

همانطور که ذکر شد، زمان تغییر نقطه ی تمرکز بینایی (Saccade) میمون ها در طول ترايال به صورت تقریباً رندوم انتخاب می شود. این ماتریس $N \times 1$ ، زمان تغییر Saccade را برای هر ترايال مشخص می کند.

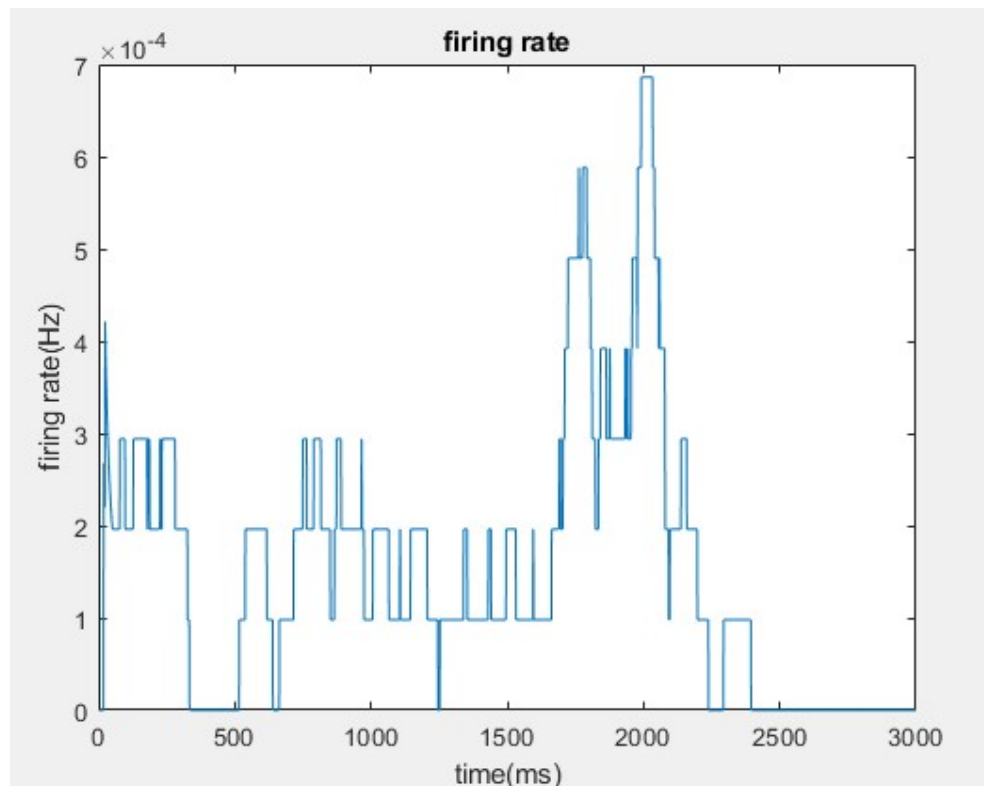
3-3- پیش پردازش

در ابتدای امر، باید توجه داشت، که قسمت اصلی پیش پردازش بر روی داده ها پیش تر انجام شده و دو عمل Clustering برای تفکیک نورو ن ها از یکدیگر و عمل Filtering برای جدا کردن رشته های اسپایکی از سیگنال ها و آر تی فکت های ضبط شده توسط الکترو د انجام شده است. اما برای توضیح آنچه در ادامه انجام داده ایم، لازم است تا نمونه ای از داده های مربوطه نمایش داده شوند.



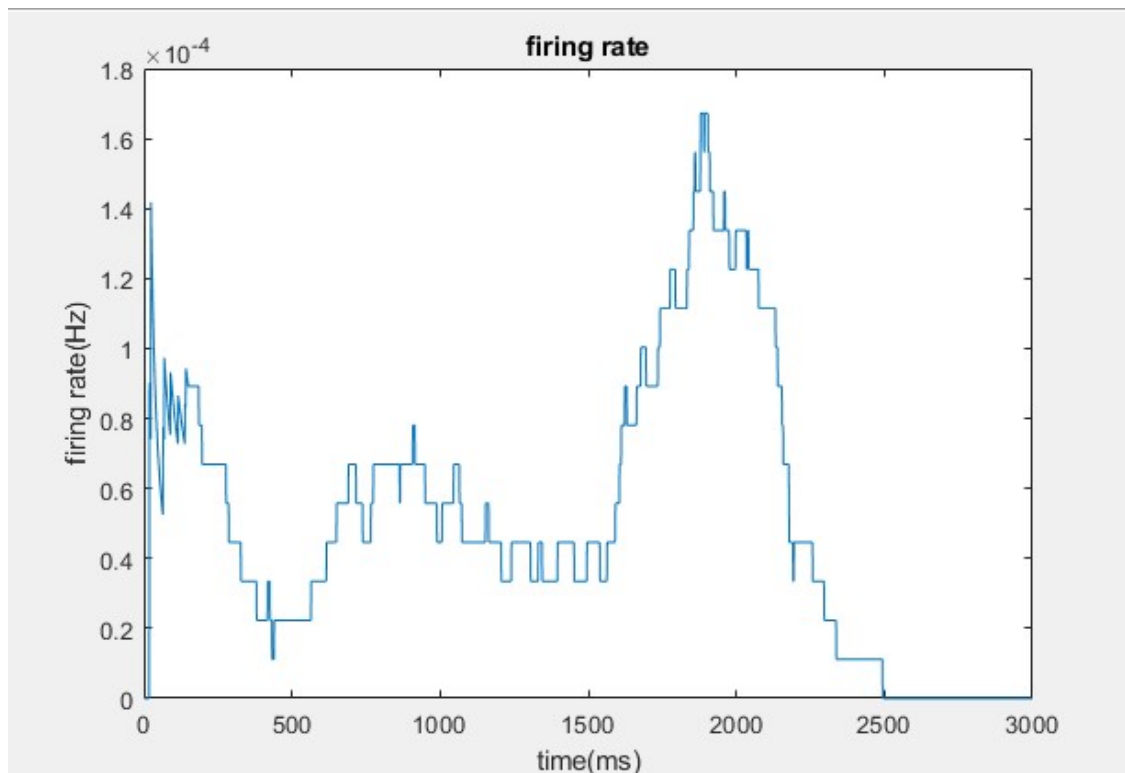
شکل 1. قطار اسپایک‌ها برای نمونه تریال 11، که زمان Saccade در آن مشخص شده است.

در صورتی که بخواهیم، تعداد رخداد‌های اسپایکی را در طول تریال نمایش دهیم، می‌توانیم از نرخ فایرینگ یا Firing rate نوروں نیز استفاده کنیم. برای این کار می‌توان سیگنال را به سگمنت‌های کوچک یا بین تقسیم کرده و یک نرخ فایرینگ برای هر بین محاسبه کنیم. پاسخ مربوطه می‌تواند اطلاعات قابل درک‌تر و راحت‌تری را برای تفسیر و بعضاً پردازش سیگنال، در اختیار ما قرار دهد. زیرا استخراج ویژگی از سیگنال پیوسته‌ی مرتبط ممکن است، برای ما راحت‌تر و سریع‌تر باشد.



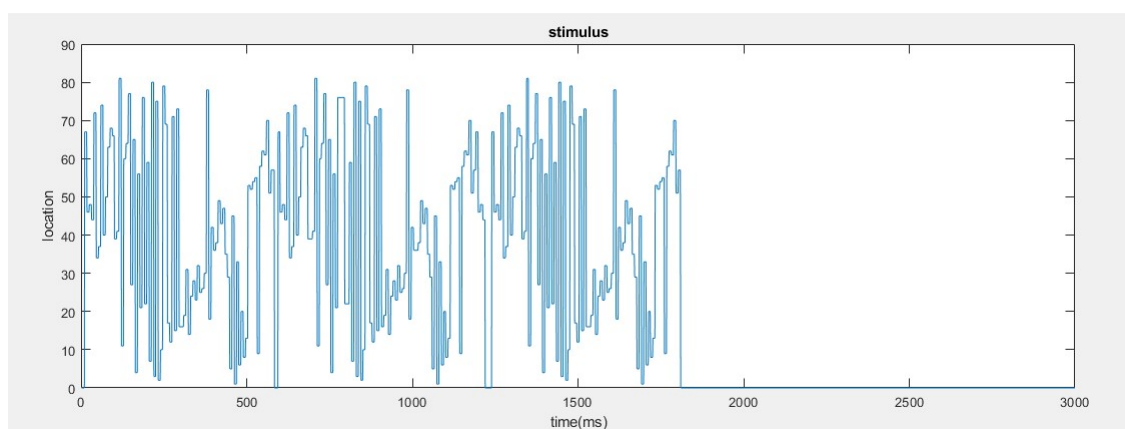
شکل 2. پاسخ نورونی پنجره گذاری شده به عنوان نمونه، با طول پنجره‌ی 101 میلی ثانیه

در صورت افزایش طول پنجره انتظار می‌رود که پاسخ نرم‌تر یا Smooth تر، شود. اگرچه با انجام این کار اطلاعات لحظه‌ای بیشتری از دست می‌رود، اما پاسخ، به یک منحنی پیوسته نزدیک‌تر می‌شود.



شکل 3. پاسخ نورونی، پنجره گذاری شده، با طول پنجره 300 میلی ثانیه

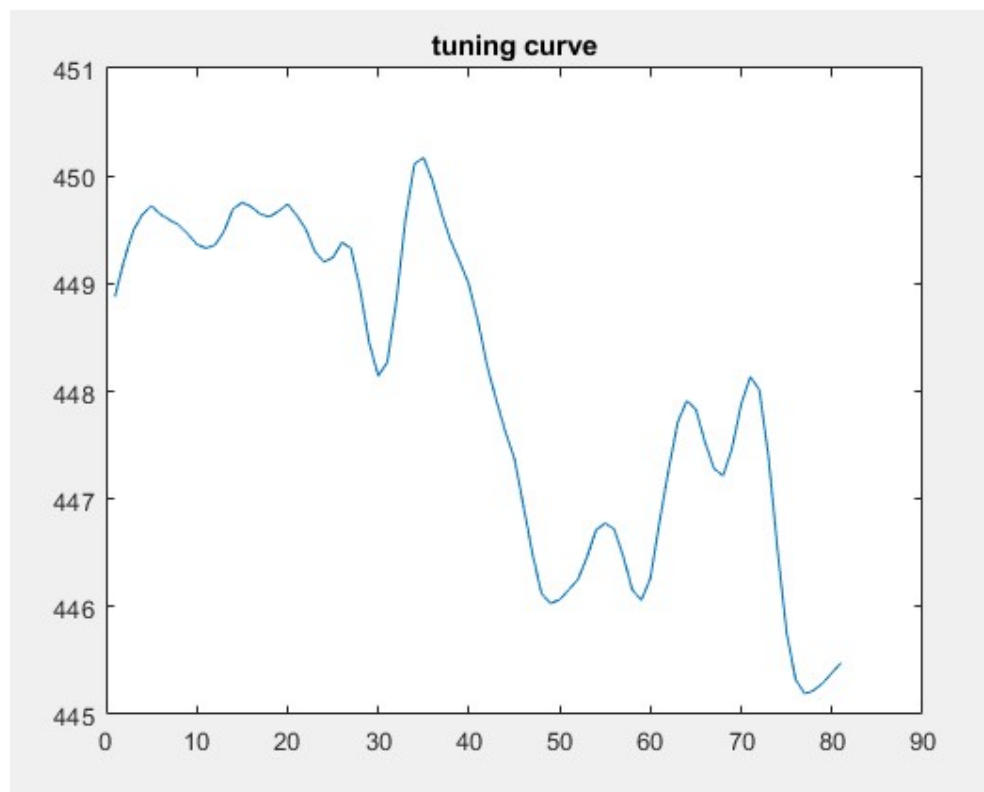
حال محرک را نیز در طول زمان ترسیم می کنیم تا نسبت به آنچه پس از این به عنوان ورودی در نظر می گیریم، واقف شویم.



شکل 4. ترسیم محرک بینایی در پس زمینه که در طول زمان کاملاً ناپیوسته تغییر می کند

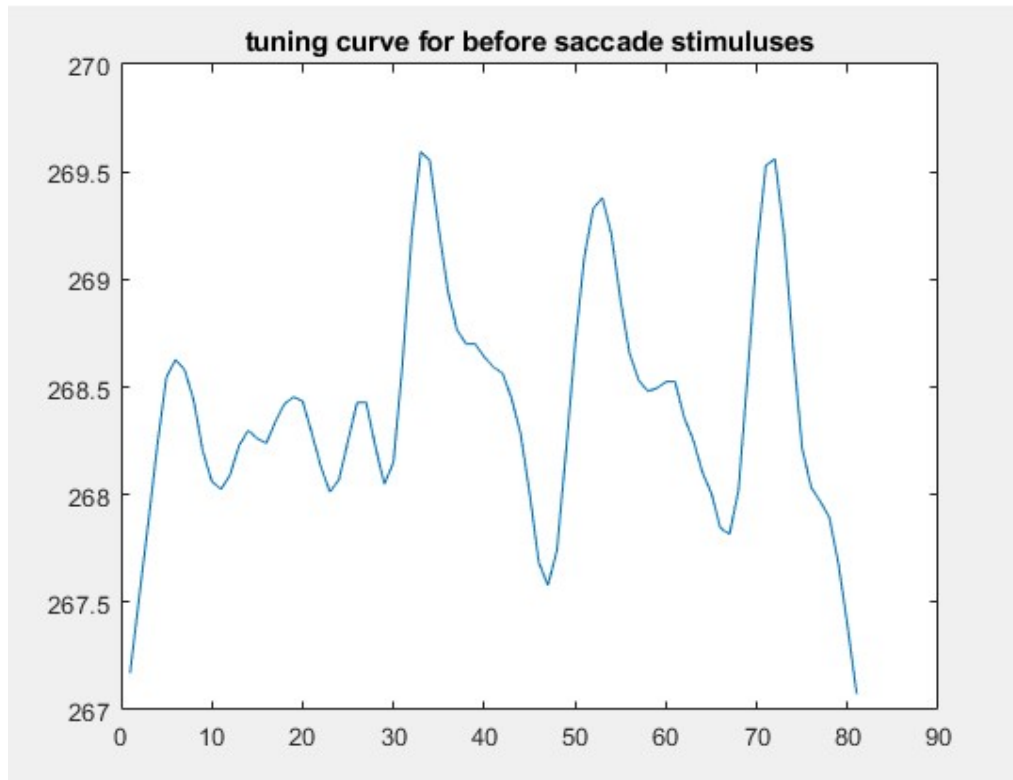
مشاهده میشود که محرک و خروجی که همان پاسخ نورون ها است، متغیرهایی با خروجی ناپیوسته

می‌باشند که به صورت غیر پریودیک و بدون وجود یک الگوی ظاهری تغییر می‌کنند. نتیجتاً برای استخراج ویژگی از محرک و پیش‌بینی خروجی نیاز به ایجاد تغییراتی روی این دو دیتا خواهیم داشت. اما پیش از آن به مقایسه‌ی پاسخ نوروئی نسبت به محرک‌ها، پیش و پس از تغییر Saccade می‌پردازیم، تا دلیل رویکردی که پس از این پیش خواهیم گرفت روشن شود. برای درک بهتری از این مهم، نمایش Tuning curve به عنوان معیاری از نمایش پاسخ نوروئی بر حسب محرک بسیار کمک کننده خواهد بود. برای یک نوروئی که به صورت تصادفی انتخاب شده، منحنی تیونینگ را مقایسه خواهیم کرد.



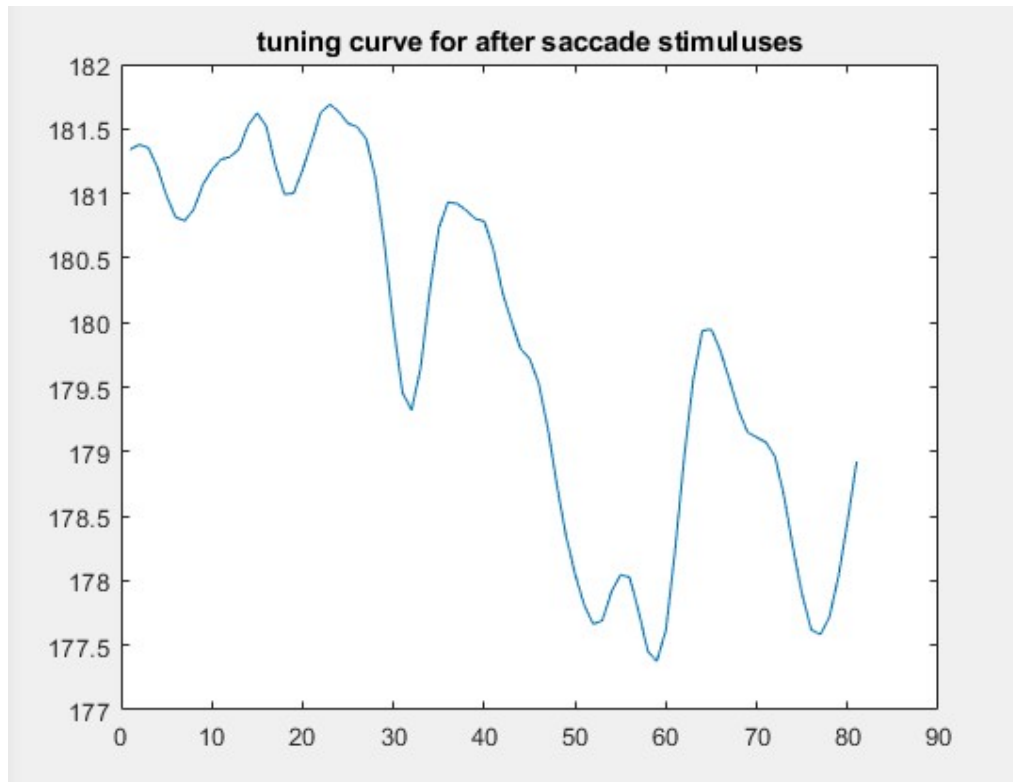
شکل 5. منحنی تیونینگ در طول تمام ترایال

همانطور که در شکل مشخص است، پاسخ نوروئی به محرک‌های موجود، در موقعیت‌های 1 تا 40 دارای فرکانس بیشتری است. اما این منحنی اطلاعاتی از تفاوت پاسخ‌دهی نوروئی پیش و پس از تغییر نقطه‌ی تمرکز بینایی سوژه به ما نمی‌دهد.



شکل 6. منحنی تیونینگ نورون پیش از Saccade

آنچه بدان اشاره شد در این شکل به وضوح دیده می‌شود. علی‌رغم اینکه در منحنی تیونینگ کل تراپال، بیشترین سطح پاسخ نورون به محرک‌های 1 تا 40 دیده می‌شد، اما پیش از تغییر Saccade پاسخ دهی نورون به این محرک‌ها بالا نبوده و حتی نسبت به برخی دیگر از محرک‌ها در حدود لوکیشن 75، پایین‌تر نیز بوده‌اند. مطابق این شکل و با توجه منحنی تیونینگ کلی، انتظار می‌رود که پاسخ نورن به محرک‌های 1 تا 40 در منحنی تیونینگ پس از تغییر Saccade به شکلی مشهود افزایش یابد.



شکل 7. منحنی تیونینگ نوروں پس از تغییر Saccade

تفاوت به سزایی که بین دو شکل آخر وجود دارد، حاکی از اهمیت به روزرسانی آموزش مدل به صورت جداگانه، پیش و پس از تغییر نقطه‌ی بینایی می‌باشد. چرا که نوروں‌ها پاسخ متفاوتی به محرک‌ها پس از تغییر این نقطه نشان خواهند داد.

به دلیل وجود زمان Saccade رندوم در بین ترایال‌ها امکان گرفتن میانگین از ترایال‌ها و بدست آوردن یک منحنی تیونینگ دقیق از پاسخ نوروئی وجود ندارد. لذا از انجام این بخش با توجه به نوع دیتاست، صرف نظر شده است.

لذا پس از این دیتا را توسط برنامه به دو قسمت پیش از Saccade و پس از Saccade تقسیم می‌کنیم. داده‌ها پس از پنجره گذاری، دسته بندی و استخراج ویژگی که جلوتر بدان اشاره خواهد شد، به دلیل وجود ناپیوستگی‌ها، دارای بخش‌هایی با عنوان Nan و قسمت‌هایی به عنوان Inf می‌شود. این قسمت‌ها در برنامه شناسایی و با مقدار میانی در هر ستون مربوط به خود جایگزین شده‌اند تا نیازی به دور ریختن هیچ بخشی از یافته‌های هرچند ناپیوسته نباشد. زیرا لحاظ کردن ماهیت ناپیوسته و تغییرات ناگهانی سیگنال نیز، بخشی از عمل پیش‌بینی سیگنال خروجی خواهد بود.

4-3- استخراج ویژگی

برای استخراج ویژگی از ورودی ما که محرک‌های بینایی هستند، و از آنجا که این ورودی را صرفاً به صورت یک سری تصادفی از اعداد 1 تا 81 در طول تریال‌های با طول 3000 میلی ثانیه در اختیار داریم. ابتدا آن را به صورت یک سیگنال یا سری زمانی در طول تریال در نظر می‌گیریم. از آنجا که این سیگنال همچنان ناپیوسته و دارای تغییرات زیاد در بازه‌های زمانی کوچک است، سرعت یادگیری سیستم را بسیار کند می‌کند. همچنین با توجه به ماهیت تصادفی قطار اسپایک‌ها در خروجی، در نظر گرفتن محرک در بازه‌ی زمانی نمایش آن، خروجی را بدست می‌دهد که در نهایت کاملاً مطابق با پاسخ واقعی در عمل نخواهد بود. با توجه به آنچه ذکر شد، سیگنال ورودی یا محرک‌ها را به بازه‌های زمانی 100 میلی‌ثانیه‌ای تقسیم می‌کنیم. در این سگمنت‌های 100 میلی‌ثانیه‌ای میانگین، انحراف معیار، واریانس، اعواج هارمونیک‌ها، نسبت سیگنال به نویز، میانگین ریشه‌ی دوم و نسبت سیگنال به اعوجاج را محاسبه می‌کنیم. هفت مورد فوق ویژگی یا فیچرهایی در حوزه‌ی زمان هستند که برای عمل کلاس‌بندی از آن‌ها استفاده می‌کنیم.

5-3- انتخاب ویژگی

از بین هفت ویژگی استخراج شده در حوزه‌ی زمان، قطعاً بخشی دارای اطلاعات یکسان خواهند بود. اگرچه استفاده از هر هفت ویژگی تاثیر مثبتی روی نتایج و دقت کلاس‌بندی دارد. اما استفاده از همه‌ی آن‌ها به نسبت کاهش سرعتی که در سیستم ایجاد می‌شود و با توجه به تعداد زیاد تریال‌هایی که در اختیار داریم، اصلاً مقرون به صرفه نیست. بنابراین به یک معیار برای مقایسه و حذف ویژگی‌هایی که دارای کورلیشن و اطلاعات مشابه می‌باشند، نیاز خواهیم داشت. برای همین، در این قسمت از روش آماری Principle component analysis و یا PCA استفاده می‌کنیم.

روش انتخاب شده، با در نظر گرفتن یک مرز بسیار کوچک برای p ، از هفت ویژگی استخراج شده در حوزه‌ی زمان، فقط سه و بعضاً دو ویژگی را انتخاب می‌کند که موجب افزایش سرعت طبقه‌بندی می‌شود. با توجه به این‌که در یک اجرا index مربوط به PCA، فیچرهای انتخاب شده را به ما نشان می‌دهد، می‌توانیم در ابتدای مسیر آن‌ها را حذف کنیم تا سرعت برنامه در طول استخراج ویژگی و ترکیب داده‌ها نیز افزایش یابد، اما از آنجا که روند آزمایش، شامل حالت آنلاین نمی‌باشد، به آنچه انجام شد، بسنده می‌کنیم.

3-6- ارزیابی و کدگذاری

پس از انتخاب ویژگی وارد مرحله‌ی کلاسبندی و پیش‌بینی می‌شویم. چنانچه قبلاً درباره‌ی آن صحبت کردیم، سیگنال‌ها به سگمنت‌های مساوی تقسیم شده‌اند. و لیبیل هر سگمنت محرک در هر تریال، سگمنت پاسخ نوروئی، مربوطه و موازی با آن در همان تریال خواهد بود. پیش شروع عمل استخراج ویژگی، داده‌های تفکیک شده پیش و پس از Saccade، مربوط به پاسخ و محرک، به صورت 75٪ داده‌های Training و 25٪ تست تقسیم شده‌اند. اما پس از استخراج ویژگی‌ها نیاز است تا داده‌های مربوط به آموزش دسته بندی شوند و به ازای هر سگمنت، ویژگی‌های یکسان در قالب ستون‌های مختلف در کنار یکدیگر قرار گیرند و لیبیل مربوط به آن‌ها که پاسخ نوروئی مرتبط با آن سگمنت است در یک ستون جداگانه موازی با سطر ویژگی‌های مربوط به خود قرار گیرد.

در ادامه می‌توان از دیتای پرداخت شده، برای امر کلاس‌بندی استفاده کرد. با توجه به آنچه ابزار متلب در اختیار ما قرار می‌دهد، میتوان دریافت که کلاسیفایرهایی که از یک توزیع خاص تبعیت کرده و یا مخصوص طبقه‌بندی سیستم‌های دو کلاسه هستند، مناسب و قابل استفاده برای حالت چندکلاسه‌ی ما که شامل تعداد زیادی از انواع پاسخ نوروئی می‌باشد، نیست. در میان آنچه در اختیار داریم، شبکه‌های عصبی، درخت تصمیم و کلاسیفایر K-nearest neighbour یا KNN، گزینه‌های مناسبی به نظر می‌رسند. همچنین از کلاسیفایر SVM به دلیل سرعت فوق‌العاده پایین در این برنامه که از راه آزمون و خطا بدست آمده است، صرف نظر می‌شود. لذا برای پیش‌بینی پاسخ نوروئی در این برنامه فقط از دو کلاسیفایر KNN و Tree decision استفاده کردیم که نتایج نسبتاً خوبی را در اختیار ما قرار داده‌اند.

اما پیش از ارزیابی کلاسبندی با داده‌های تست و برای آموزش مدل، از K-fold Cross Validation استفاده شد، و با آزمون و خطا و به ازای $K = 3, 10, 20$ مدل ارزیابی شد. با توجه به این که تاثیر چندانی در عملکرد کلاسیفایر مشاهده نشد و سرعت سیستم کاهش چشمگیری یافت، این قسمت در ادامه لحاظ نشده و فقط برای آزمون احتمالی کاربر، در برنامه قرار داده شده است. دلیل نداشتن تاثیر به سزای Cross Validation از نظر ما، وجود حجم بسیار بالای اطلاعات و تعدد تریال‌ها می‌باشد که مدل را از آموزش بیشتر و ارزیابی‌های دیگر بی‌نیاز می‌کند.

در زیر نتایج دقت در کلاسبندی و پیش‌بینی داده‌های تست را پیش از تغییر نقطه‌ی تمرکز بینایی، برای نوروئ‌های مختلف که به صورت رندوم از میان 41 نوروئ انتخاب شده‌اند، مشاهده می‌کنید.

جدول 1. دقت کلاسبندی 7 نورون پیش از تغییر نقطه‌ی تمرکز بینایی

number	Session Date	Channel number	Accuracy of Tree-decision before Saccade	Accuracy of KNN before Saccade
1	2015/09/27	10	75.9%	75.8%
2	2015/12/15	16	70.3%	69.9%
3	2015/12/15	11	67.2%	66.5%
4	2015/07/17	2	76.3%	75.4%
5	2015/08/20	5	83.1%	82.3%
6	2015/09/05	4	72.1%	71.2%
7	2015/09/05	8	65.8%	65.4%

جدول بعدی، دقت کلاسبندی همان نورون‌ها را به ازای کلاسیفایرهای ترین و تست شده با اطلاعات پس از Saccade time نشان می‌دهد. مطابق انتظار، تفاوت دقت چندانی بین این دو قسمت دیده نمی‌شود.

جدول 2. دقت کلاسبندی 7 نورون پس از تغییر نقطه‌ی تمرکز بینایی

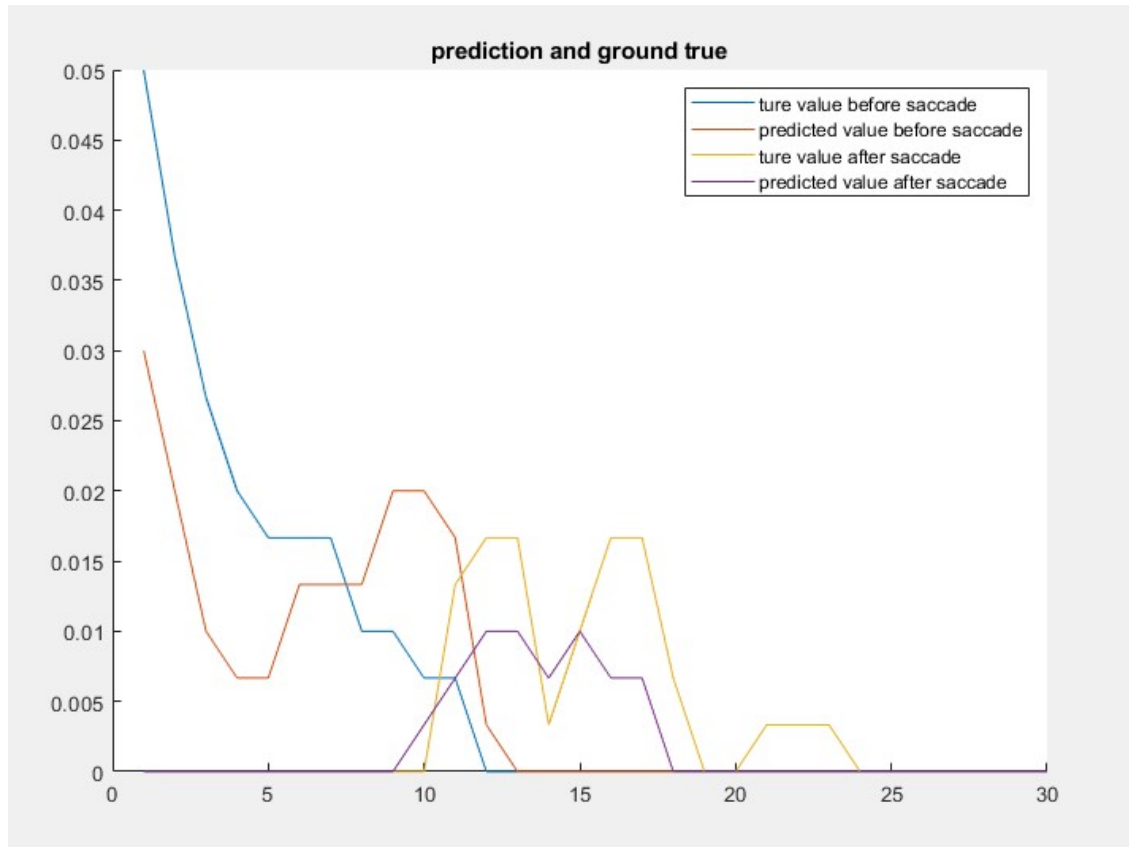
number	Session Date	Channel number	Accuracy of Tree-decision after Saccade	Accuracy of KNN after Saccade
1	2015/09/27	10	79.6%	78.0%
2	2015/12/15	16	68.6%	68.3%
3	2015/12/15	11	63.5%	63.5%
4	2015/07/17	2	68.9%	67.9%
5	2015/08/20	5	79.4%	78.6%
6	2015/09/05	4	72.3%	71.5%
7	2015/09/05	8	66.3%	65.9%

فصل 4:

نتایج و تفسیر آنها

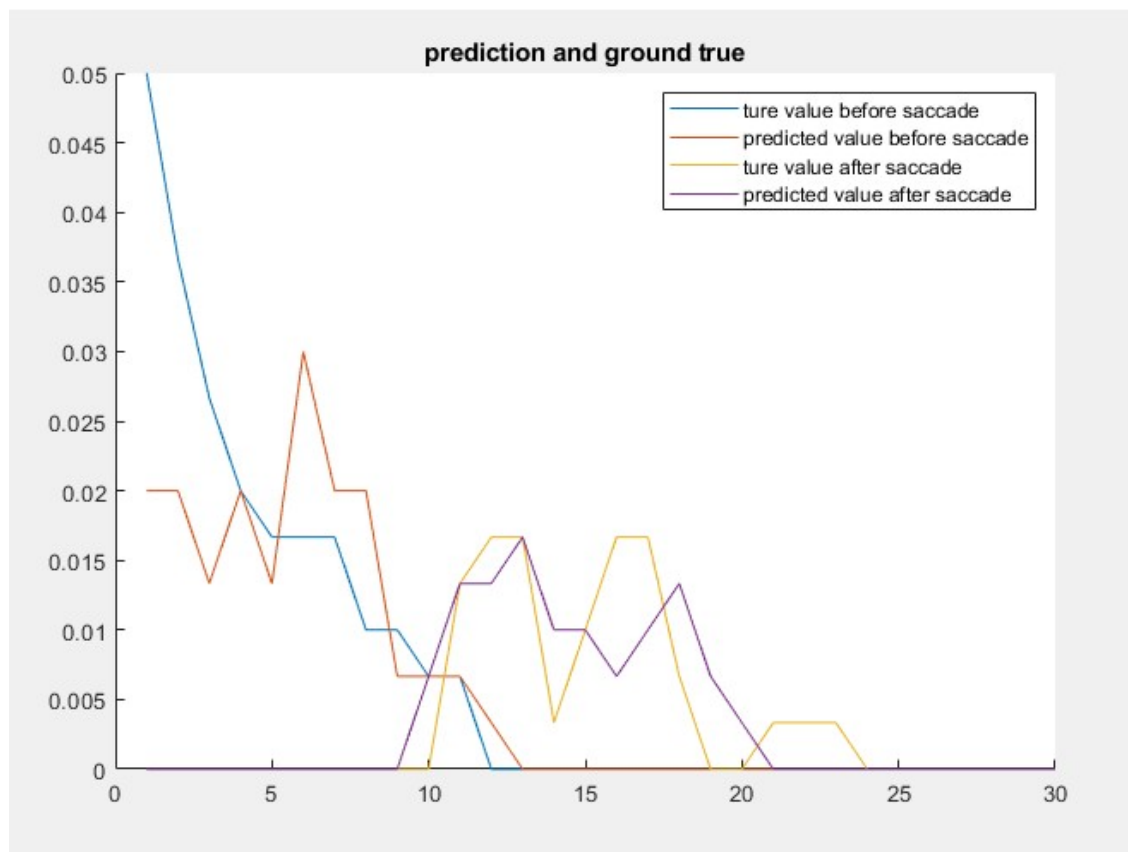
4-1- بحث و نتیجه‌گیری

همانطور که در جداول بخش قبل مشاهده می‌شود، کلاسیفایرهای مورد استفاده در این پروژه در برخی از موارد می‌توانند، دقت قابل قبولی ارائه دهند. در ادامه به بررسی گرافیکی نتیجه‌ی این تخمین‌ها و مقایسه‌ی نتایج حاصل از پیش‌بینی کلاسیفایر با داده‌های حقیقی خواهیم پرداخت.



شکل 8. مقایسه‌ی نتایج حاصل از پیش‌بینی درخت تصمیم و مقدار واقعی یک نورون پیش و پس از Saccade

کورلیشن و ارتباط پاسخ با محرک در قسمت‌هایی از شکل مشهود می‌باشد. برای مثال اگرچه مقدار میانگین پاسخ در سگمنت‌های نخست، دقیقاً مشخص نشده است، اما کلاسیفایر به خوبی روند نزولی آن را با توجه به محرک‌های نمایش داده شده، پیش‌بینی کرده است. آنچه در بالا ترسیم شده حاصل تخمین کلاسیفایر درخت تصمیم می‌باشد. که میانگین دقت 76٪ را برای کل تریال ارائه داده است. در شکل زیر نتایج واقعی و حاصل از پیش‌بینی کلاسیفایر KNN را مقایسه می‌کنیم.



شکل 9. نتایج حاصل از پیش‌بینی کلاسیفایر KNN و مقدار واقعی، پیش و پس از Saccade

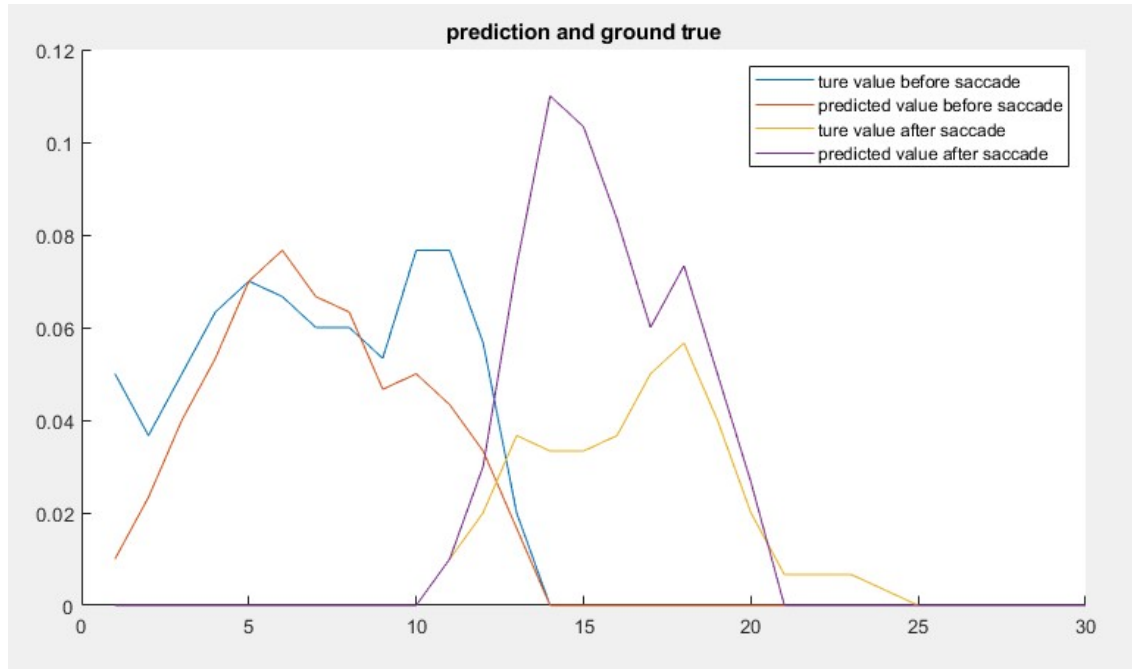
همانطور که در شکل مشهود است، نتایج کلاسیفایر KNN برای تخمین خروجی نورو، پس از Saccade به مقدار واقعی بسیار نزدیک است. اما پیش از Saccade عملکردی به خوبی درخت تصمیم نداشته است. این کلاسیفایر مقدار دقت میانگین 71.6٪ را ارائه می‌دهد.

آنچه تاکنون دیدیم نتایج حاصل از پنجره گذاری سیگنال ورودی و خروجی به ازای پنجره‌هایی با طول 100 میلی ثانیه می‌باشد. از آنجا که در طول پنجره‌ها اطلاعات میانگین گرفته می‌شوند، کاهش طول پنجره‌ها موجب افزایش دقت خواهد شد، اما این افزایش دقت، به قیمت بایاس شدن نتایج به سمت صفر و کند شدن سیستم خواهد بود. به بیان دیگر، از آنجا که بیشتر طول تراپال دارای مقدار صفر (بدون اسپایک) بوده است، بیشتر دیتا حاوی اطلاعات صفر خواهد بود و پس از آموزش مدل، نتایج به سمت صفر صفر بایاس خواهند شد که دقت کلاسیفایر را افزایش می‌دهد. اما خروجی مدل، هیچ شباهتی به خروجی واقعی به جز در نقاط نزدیک صفر نخواهد داشت.

به عکس، بزرگتر کردن طول پنجره، علی‌رغم اینکه موجب کاهش چشمگیر دقت می‌شود، باعث کمتر شدن تعداد کلاس‌ها و محدود شدن داده می‌شود که بیشترین کورلیشن میان پاسخ و نتایج پیش‌بینی را بدست

خواهد داد.

برای مثال میتوان به شکل زیر اشاره کرد که با استفاده از پیش‌بینی کلاسیفایر KNN و به ازای دقت میانگین 66.6٪ درصد بدست آمده است.



شکل 10. نتایج حاصل از پیش‌بینی و مقدار واقعی یک ترايال مربوط به پاسخ یک نورو ن با دقت 66.6٪

مشاهده می‌شود که نتایج تخمین نسبت به دو شکل بالا کورلیشن بیشتری با شکل ظاهری پاسخ نورو ن دارا است. اما آنچه باعث پایین‌تر بودن دقت آن شده است، کمتر بودن بایاس مدل به سمت پاسخ صفر می‌باشد، که بالاتر ضمن مطرح کردنف مسئله‌ی طول پنجره به آن اشاره شد. این مطلب را علاوه بر شکل می‌توان از روی میانگین نرخ فایرینگ 84.5 برای پس از Saccade و میانگین 111.2 پیش از Saccade تشخیص داد. به میزانی که نرخ فایرینگ بالاتر می‌باشد، میانگین سگمنت‌های کمتری برابر صفر بوده و مدل با اطلاعات متنوع‌تری آموزش داده می‌شود، این مسئله موجب تفکیک‌پذیری بهتر و دقت پایین‌تر می‌شود.

فصل 5:

جمع بندي و پيشنها دها

1-5- مقدمه

در این مطلب به بررسی روشی برای پیش‌بینی کدگذاری نورونی در طول انجام Saccade task، همراه با محرک‌هایی در پس زمینه پرداختیم. ابتدا داده‌ها را به دو دسته‌ی پیش و پس از تغییر نقطه‌ی تمرکز بینایی تقسیم کرده و پس از آن به سگمنت‌های 100 میلی‌ثانیه‌ای تقسیم کردیم. محرک‌ها را به عنوان ورودی و سیگنال میانگین نرخ فایرینگ در هر سگمنت را به عنوان برچسب یا لیبل داده در نظر گرفتیم. از هر پنجره سیگنال ورودی 7 ویژگی در حوزه‌ی زمان استخراج شد که در مرحله‌ی انتخاب ویژگی، 5 تا 4 تای آنها برای هر نورون حذف میشدند و با آنچه باقی ماند به پیش‌بینی خروجی با استفاده از دو کلاسیفایر درخت تصمیم و KNN، پرداختیم.

2-5- محتوا

1-2-5- جمع‌بندی

مشخصاً از اجرای این پروژه اطلاعاتی دریافت شد که اشاره به آن‌ها خالی از لطف نمی‌باشد. نخست این که افزایش طول پنجره، همانطور که در فصل قبل مطرح شد، پاسخی حدودی‌تر و با دقت پایین‌تری ارائه خواهد داد، که در طول زمان Smooth تر به نظر می‌رسد، اما به طور قطع پاسخ آن دارای کورلیشن و همبستگی بیشتری با پاسخ واقعی نورون خواهد بود. دلیل این امر، کاهش تنوع تغییرات خروجی و پیوسته‌تر شدن سیگنال و مهم‌تر از همه کمتر شدن تعداد میانگین‌های صفر، در طول ترایال‌ها است که باعث بایاس شدن تخمین به سمت خروجی صفر خواهد شد.

پس از آن این مسئله مطرح می‌شود که پاسخ نورونی، پیش و پس از Saccade نسبت به محرک‌ها دارای تغییر محسوسی خواهد بود. منحنی‌های تیونینگ رسم شده در فصل‌های گذشته به خوبی حاکی از اثبات این مسئله می‌باشند. بنابراین تقسیم داده‌ها به دو قسمت پیش و پس از Saccade برای آموزش و آزمودن کلاسیفایرها و ترکیب خروجی‌ها در پیش‌بینی سیگنال خروجی، انتخابی مناسب و ثمر بخش تلقی می‌شود.

2-2-5- نو آوری

اگرچه آنچه در این مطلب بدان پرداخته شد، دارای مطالب جدید و روشی نوآورانه نیست، ولی ظاهراً آنچه به آن پرداخته شده، قبلاً در مقالات مربوط به این داده مورد بررسی قرار نگرفته و به نوعی جدید می‌باشد. حاصل تقسیم‌بندی داده به دو بخش، پنجره‌گذاری با طول‌های مختلف در این پروژه و نتایج آن، هرچند چشمگیر نیستند، ولی می‌توانند اطلاعات مفیدی را در اختیار پژوهشگران این حوزه، در رابطه با این دیتاست قرار دهند. در مقاله‌ی مربوط به این دیتاست، به جای استفاده از دو کلاسیفایر مجزا برای زمان Saccade یک مدل متغیر با زمان در نظر گرفته شده که طبعاً تعداد بیشتری از Saccade‌ها را برای کاربردهای عملی پوشش خواهد داد، ولی استفاده از کلاسیفایرهای خطی مجزا به دلیل هزینه‌های محاسباتی پایین و سرعت بالا، همچنان می‌توانند به عنوان گزینه مطرح شوند.

3-2-5- پیشنهادات

در این مطلب، آنطور که باید و به شکل مطلوب به تلاش برای بهینه‌سازی مدل‌های پیش‌بین و یا کلاسیفایرها، جز استفاده از Cross-validation که موجب کندی و افزایش نه چندان قابل توجه دقت می‌شود، نشده است. برای کارهای مرتبط و بعدی می‌توان روش‌های متنوعی را برای بهینه‌سازی مدل‌های استفاده شده در نظر گرفت.

همچنین استفاده از شبکه‌های عصبی به عنوان یک مدل برای کلاسیفایر توصیه می‌شود. برای مثال شبکه‌ی LSTM در مقالات مرتبط اخیر به خوبی در امر Decoding و Incoding پاسخ نوروئی و محرک‌ها موفق ظاهر شده است.

در ادامه ارائه‌ی روش یا الگوریتمی برای یافتن یک طول بهینه برای پنجره‌گذاری سیگنال‌های ورودی و خروجی نیز می‌تواند عملکرد سیستم را تا حد قابل توجهی بهبود بخشد که در این مطلب به آن پرداخته نشده است.

از آنجا که تمامی ویژگی‌های استخراج شده از سیگنال در این پروژه، در حوزه‌ی زمان بوده‌اند، استفاده از فیچرهای حوزه‌ی فرکانس، و ضرایب تبدیل ویولت سیگنال‌ها قطعاً تأثیر به سزایی در بهبود و بهینه‌سازی امر پیش‌بینی و کلاسیفایر خواهد داشت.

مراجع

مراجع

- [1] Characterizing and dissociating multiple time-varying modulatory computations influencing neuronal activity by Kaiser Niknam, Amir Akbarian, Kelsey Clark, Yasin Zamani, Behrad Noudoost, Neda Nategh, PLOS Computational Biology
- [2] Capturing spike train temporal pattern with wavelet average coefficient for brain machine interface, Shixian Wen, Allen Yin, Po-He Tseng, Laurent Itti, Mikhail A. Lebedev & Miguel Nicolelis, Published: 24 September 2021 ,nature
- [3] Theoretical Neuroscience, Peter Dayan and L.F. Abbott, Published at 2002

پیوست‌ها

Abstract:

In this paper we consider a several classifier based, model which is able to encode neuron's output response, in a more practical manner. Since the data set, obtained from the middle temporal cortex of two macaque monkeys contains random changes in Saccade point, it is desired to design a model, capable of classifying the neural response in a time-varying configuration. As the time-varying models are well known for high demanding amount of training data and enormous computational costs, we propose several linear classifiers, designed and trained to encode the response depending on the Saccade point. We also do a small investigation on the effect of window size, on the data segmentation which leads to some interesting and informative results.

Keywords: DataSet, Saccade task, Neural Encoding, Linear Classification



IU | ST

**Iran University of Science and Technology
Electrical Engineering Department**

Title

An approach to encode neurons during a Saccade task

By:

Pouya Ahmadpour

Supervisor:

Dr. Mohammad Reza Daliri

Advisor:

Saeed Sang Sefidi

2022