به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی برق

علوم اعصاب یادگیری، شناخت و حافظه

پروژه‌ی پایانی درس: بررسی

استاد: **دکتر کربلایی آقاجان**

امیرحسین برقراری شماره دانشجویی
مازیار شمسی‌پور ۹۸۱۰۱۸۴۴

۲۶ تیر ۱۴۰۰

۱ مقدمه

توضیحات کلی در مورد کدها و پروژه و مقاله.

۲ آشنایی با مقاله‌ی پژوهش

۱.۲ هدف پژوهش

هدف این مقاله مدل سازی receptive field نورون‌های پیچیده غشر بصری[۱] می‌باشد. در واقع با اعمال تحریک‌های تصادفی spatiotemporal از توزیع $p(s)$ و مشاهده‌ی پاسخ نورون‌ها می‌خواهیم راستایی را بدست بیاوریم که در آن $p(s|r)$ تفاوت معناداری با توزیع اولیه‌ی تحریک‌ها داشته باشد.

تفاوت این مقاله با پژوهش‌های قبل خود آن است که به دلیل رفتار غیرـ خطی نورون‌های پیچیده‌ی موجود در Visual cortex نمی‌توانیم از آنالیزهای خطی spike-triggered average که پیش از این کارگشا بودند استفاده کنیم. این مقاله روش spike-triggered correlation analyses را پیشنهاد می‌دهد که بر پایه‌ی Wiener Kernel طراحی شده است.

نهایتا با این پژوهش با بررسی‌هایی که انجام می‌دهد ادعا می‌کند که می توان با این روش پایه‌ای برای تحریک‌ها ارائه داد که تعداد کمی از آن‌ها مشخص کننده‌ی ویژگی‌های مرتبط و تعداد زیادی مربوط به ویژگی‌های پوچ می‌باشند.

۲.۲ نورون‌های «پیچیده»

این مفهوم اولین بار در مقاله‌ی نوبلیست Hubel and Wiesel مطرح شده است. نورون‌های ساده نورون‌هایی هستند که رابطه‌ی تحریکـ پاسخ آنها نسبت به زمان خطی می‌باشد. این موضوع باعث به وجود آمدن مناطق ON-OFF در Receptive Field نورون مورد نظر می‌شود. به عکس دست نوشته‌ی موجود در مقاله‌ی مذکور برای جزئیات بیشتر توجه کنید.

شکل ۱: مناطق ON-OFF حوزه‌ی دریافتی نورون‌های ساده[۲]

همچنین در بخش Material and Methods آمده است که اگر نسبت هارمونیک اول به مقدار DC حوزه‌ی دریافتی بزرگتر از یک باشد نورون را ساده می‌نامیم. روشن است که نورون‌هایی که از تعاریف بالا پیروی نکنند ساده نبوده و پیچیده‌اند.

[۱] Visual cortex

[۲] EVOLUTION OF IDEAS ON THE PRIMARY VISUAL CORTEX, 1955-1978. BY DAVID H. HUBEL

۳.۲ STC analysis

می‌دانیم که اگر یک ویژگی در تحریک باعث تغییر احتمال اسپایک زدن شود، می‌دانیم که اگه تحریک‌ها را به فضای آن spike-triggered منتقل کنیم باید تفاوت قابل توجهی بین مقادیر $p(s|r)$ و $p(s)$ باشد.[۳]

روش‌های مبتنی بر correlation با هدف بررسی تغییر این توزیع احتمالات با توجه به تغییرات گشتاور مرتبه‌ی دوم آنها(واریانس) طراحی می‌شوند.

در این بین روش PCA به این خاطر که پایه‌ای در اختیار ما قرار می‌دهد که راستاهای بیشترین تا کمترین واریانس می‌باشند، بسیار مناسب کار ما خواهد بود و می‌تواند ویژگی‌هایی را که در آن واریانس تحریک‌ها بسیار بالا می‌باشد را در اختیار ما بگذارد.

روش spike-triggered analysis مبتنی بر موارد فوق مراحل زیر را انجام می‌دهد:

۱. ابتدا با فرض اینکه حافظه‌ی نورون‌ها بیشتر از ۱۶ فریم یا ۲۶۸ میلی‌ثانیه نخواهد بود؛ پترن‌های تحریک را متشکل از ۱۶ نوار رنگی در ۱۶ فریم و در فضای ۲۵۶ بعدی تعریف می‌کنیم.

۲. ماتریس spike-triggered correlation را به صورت زیر تعریف می‌کنیم:

$$C = \frac{۱}{N} \sum_{i=۱}^{N} S(i)^T S(i)$$

که در آن $S(i)$ بردار ۲۵۶ بعدی در iامین باریست که نورون در طول آزمایش اسپایک زده است و N تعداد کل اسپایک‌ها در طول آزمایش است.

۳. برداررویژه‌ها و مقدارویژه‌های این ماتریس محاسبه شده و با توجه به اندازه‌ی مقدارویژه‌ها رتبه‌بندی می‌شوند.

با توجه به PCA می‌دانیم که حاصل راستاهایی خواهد بود که در آن واریانس تحریک‌هایی که موجب اسپایک می‌شوند از زیاد به کم مرتب شده اند.

۴.۲ اعتبارسنجی نتایج

برای اعتبارسنجی مشاهدات روش spike-triggered correlation روشی که مقاله استفاده می‌کند این است که دنباله‌ای از اسپایک‌های تصادفی با تعداد اسپایک‌های مساوی با N تولید می‌کند و ماتریس correlation تحریک‌هایی که موجب این اسپایک‌های فرضی تصادفی شدند را ایجاد می‌کنیم. هدف این است که control correlation matrix را محاسبه کنیم که به این معناست

[۳] $p(\text{response} \mid \text{stimulus})$

که روشی که در قبل پیش گرفتیم را برای داده‌هایی که می‌دانیم مستقل از اسپایک زدن هستند(تمام داده‌ها)نیز اعمال کنیم. با توجه به حجم بالای داده‌ها N مساوی تعداد اسپایک‌ها را در نظر می‌گیریم و ۵ بار فرایند تولید control correlation matrix را انجام می‌دهیم و میانگین مقدارویژه‌ها را محاسبه می‌کنیم.با توجه به متن مقاله حاصل این میانگین‌گیری $\pm$۵٫۲SD بازه‌ی اطمینان ما با $p < ۱۰^{-۴}$ خواهد بود. و بنابراین اگر مقدارویژه‌ای خارج از این بازه‌ی اطمینان باشد؛ آن مقدارویژه و بردارویژه دارای تفاوت آشکار خواهند بود.(در واقع آن بردار ویژه راستاییست که تحریک‌های موجب اسپایک در آن واریانس و در نتیجه توزیع احتمال کاملا متفاوتی با توزیع اولیه دارند)

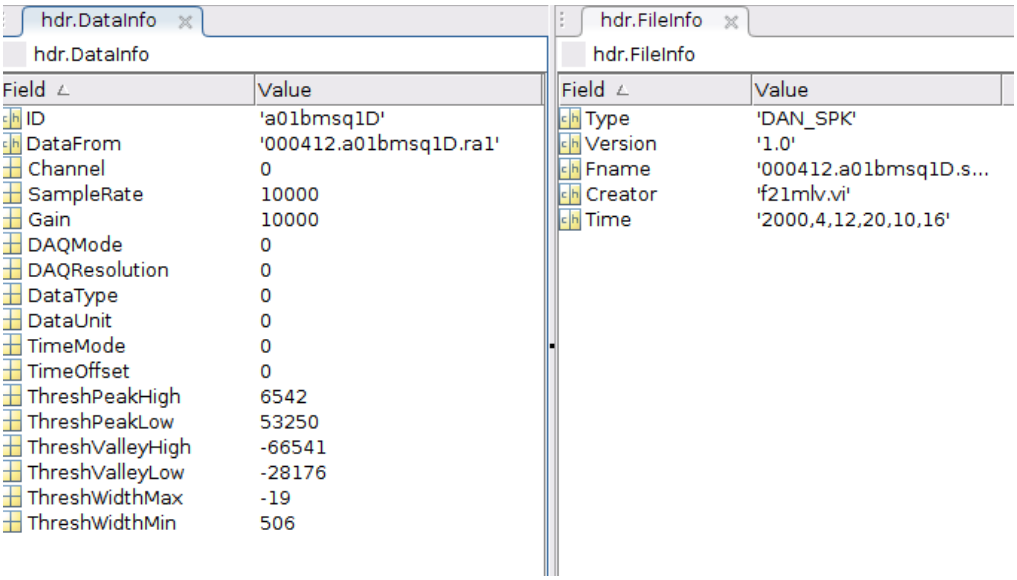
۵.۲ نتایج پژوهش

همانطور که در بخش‌های قبل اشاره کردیم هدف پیدا کردن eigenvalueهایی بود که خارج از بازه‌ی اطمینان باشند که آن‌ها به عنوان ویژگی‌های اصلی موجود در receptive-field گزارش شود.نتایج مقاله نشان می‌دهد که برای تعداد زیادی از سلول‌های پیچیده دو بردارویژه پیدا شده‌است که این بردار ویژه‌ها ناحیه‌ی ON-OFF مشخص و مجزا ازهمی دارند و همچنین correlation بسیار پایینی دارند.همچنین می‌بینیم که این راستاها به خوبی سبب تحریک نورون‌ها می‌شوند با این حال برای eigenvalueهایی که در بازه‌ی اطمینان مذکور قرار دارند تاثیر بسیار کمتری در تحریک نورون‌ها مشاهده شده است.

۳ آشنایی با دیتاست

۱.۳ بخش اول

با استفاده از تابع تعبیه شده‌ی `fget_hdr.m` یکی از فایل‌های `sa0`. را باز می‌کنیم و به محتویات آن توجه می‌کنیم.



شکل ۲: محتویات فایل `hdr`

می‌توانیم ببینیم که در این فایل اطلاعاتی اطلاعاتی مربوط به خود فایل از جمله زمان ایجاد دیتاست و سازنده‌ی آن و همچنین اطلاعاتی در مورد داده‌ها از جمله نرخ نمونه‌گیری وجود دارد. همچنین در فایل `log`. نیز اطلاعاتی مربوط به نورونی که از آن نمونه‌گیری می‌شود وجود دارد.

نوع اطلاعات	برخی از اطلاعات موجود
اطلاعات مربوط به فایل	زمان ایجاد فایل، سازنده‌ی فایل، ورژن فایل
اطلاعات مربوط به تحریک	نوع تحریک، نرخ بروزرسانی مانیتور، نرخ تغییر فریم‌ها
اطلاعات مربوط به tuning curve	زاویه‌ی ترجیحی نورون، طول و عرض ترجیحی نورون
اطلاعات مربوط به نمونه‌گیری	نرخ نمونه‌گیری، نوع کارگذاری الکترودها، تعداد چنل‌ها

و اطلاعات دیگری از این دست می‌باشد.

۲.۳ بخش دوم

تابع `Func_ReadData` را به صورتی که خواسته شده پیاده‌سازی می‌کنیم. ایده‌ی کلی در پیاده‌سازی این تابع این بود که ابتدا با استفاده از تابع `dir` فایل‌های موجود در دایرکتوری نورون مورد نظر را می‌خوانیم و فایل‌های با `mds1d.sa0` را پیدا می‌کنیم و اطلاعات آن را با استفاده از `fget_spk.m` می‌خوانیم.

همچنین با توجه به اینکه در بخش بعد به Spike-count rate احتیاج پیدا می‌کنیم یکی از خروجی‌های Optional را برابر با آن تعریف می‌کنیم، برای محاسبه‌ی این مقدار نیز تنها کافیست توجه کنیم که:

$$\langle r \rangle = \langle \frac{\#spikes}{time} \rangle_{trials}$$

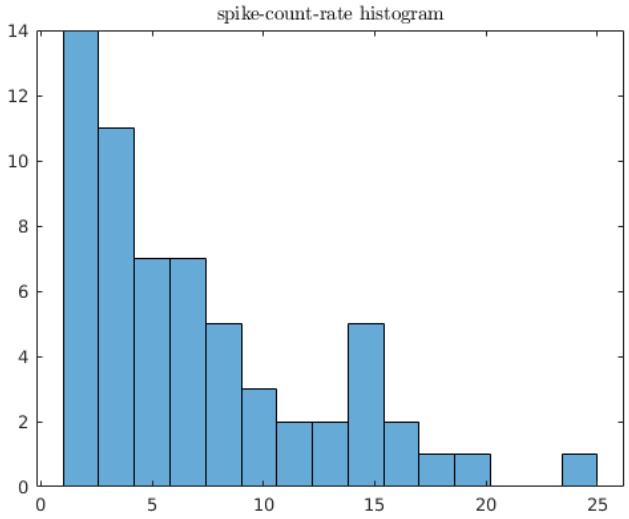
نهایتا تابع مورد نظر به صورت زیر خواهد بود:

۱.۲.۳ قطعه کد: تابع Func_ReadData

```
% Recomend: select Data/ and MatlabFucntions/
% directories and add them to matlab PATH
function [Output, spike_count_rate] = Func_ReadData(NeuronCode)
  Path = [pwd, '/Data', '/Spike_and_Log_Files/', NeuronCode];
  Listing = dir(Path);
  Index_Counter = 0;
  events = {};
  hdrs = {};
  frame_rate = 59.72; %Hz
  frame_number = 32767;
  spike_count = [];
  for i = 1:length(Listing)
    if (contains(Listing(i).name, 'msq1d.sa0', 'IgnoreCase', true
       ) && ...
      ~contains(Listing(i).name, 'sa0.sub', 'IgnoreCase', true)
         && ...
      ~contains(Listing(i).name, 'sa0.', 'IgnoreCase', true))
      Index_Counter = Index_Counter + 1;
      [events{Index_Counter}, hdrs{Index_Counter}] = fget_spk(
         Listing(i).name, 'get');
      spike_count(Index_Counter) = length(events{Index_Counter});
    end
  end
  Output = struct('events', events, 'hdr', hdrs);
  spike_count_rate = mean(spike_count) * frame_rate/frame_number;
end
```

۳.۳ بخش سوم

در این بخش ابتدا با استفاده از تابع dir در پوشه‌ی مربوط به اطلاعات اسپایک‌ها نام تمام نورون‌ها را استخراج می‌کنیم. سپس با پیمایش بر روی آن‌ها و با استفاده از تابعی که در بخش قبل نوشتیم نرخ اسپایک هر نورون را نگهداری می‌کنیم. نهایتا هیستوگرام نرخ اسپایک‌ها به صورت زیر خواهد بود:



شکل ۳: هیستوگرام نرخ اسپایک‌های نورون‌ها

سپس به سادگی می‌توانیم نورون‌هایی که نرخ اسپایک کمتر از ۲ دارند را با استفاده از قطعه کد زیر از لیست اولیه‌ای که برای کد نورون‌ها تهیه کردیم حذف کنیم.

۱.۳.۳ قطعه کد: حذف نورون‌ها با نرخ اسپایک پایین

```
index_counter = 0;
for i = 1:61
  if SCRA(i) < 2
    index_counter = index_counter + 1;
    index(index_counter) = i;
  end
end
fprintf('Exluded Neurons:\n')
disp(neuron_codes(index)')
neuron_codes(index) = [];
```

خروجی این قطعه کد شماره‌ی نورون‌های حذف شده را نمایش می‌دهد که به صورت زیر است:

```
Exluded Neurons:
"000413.b03"
"000413.b04"
"000413.b05"
"000418.a01"
"000420.b02"
"000524.c01"
"000907.f07"
```

۴.۳ بخش چهارم

در این قسمت تابع Func_StimuliExtraction به گونه‌ای پیاده می‌کنیم که با دریافت دنباله‌ای از زمان اسپایک‌ها تحریک‌هایی با ابعاد ۱۶*۱۶ که سبب این اسپایک‌ها شدند را خروجی بدهد. برای اینکار باید توجه کنیم که با توجه به متن مقاله نرخ تغییر فریم‌های $۶۰Hz$ و به طور دقیق با توجه به فایل‌های لاگ $۵۹/۷۲۱۳Hz$ می‌باشد. همچنین از آنجایی که نرخ نمونه برداری $۰/۱ms$ می‌باشد می‌توان دید که با استفاده از کد زیر می‌تواند اندیس‌هایی از فریم‌ها که در آن‌ها اسپایک رخ داده است را ببینیم.(از آنجایی که تمایل داریم ابعاد ماتریس ۱۶*۱۶ باشد اندیس‌های کمتر از ۱۶ را نادیده در نظر می‌گیریم)

۱.۴.۳ قطعه کد: یافتن اندیس‌های زمان اسپایک و نادیده گرفتن اندیس‌های کمتر ۱۵

```
1 indices = ceil((events * frame_rate) / 10000);
2 indices = indices(indices>15);
```

سپس می‌توانیم ماتریس‌های ۱۶*۱۶ که سبب اسپایک شدند را تشکیل دهیم.

برای راحت‌تر شدن کار در مراحل بعد اولا ورودی دوم ذکر شده در صورت سوال یعنی msq1D را اختیاری در نظر گرفتیم که مقدار دیفالت آن همان فایلی خواهد بود که در پوشه‌ی Data قرار دارد. همچنین در مراحل بعد از آنجایی که می‌خواهیم دنباله‌ای تصادفی از اسپایک‌ها تولید کنیم راحت‌تر خواهد بود که فقط اندیس‌ها را به عنوان ورودی به این تابع بدهیم به این منظور یک ورودی option تعریف کردیم که اگر مقدار آن 'random' باشد مقدار events در ورودی را به عنوان اندیس‌ها در نظر می‌گیرد و در غیر این صورت همان منطق قبلی را اجرا می‌کند.