

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

۲۷ تیر ۱۴۰۰

۱ مقدمه

توضیحات کلی در مورد کدها و پروژه و مقاله.

۲ آشنایی با مقاله پژوهش

۱.۲ هدف پژوهش

هدف این مقاله مدل سازی receptive field نوروهای پیچیده غش بصری^۱ می باشد. در واقع با اعمال تحریک های تصادفی spatiotemporal از توزیع $p(s)$ و مشاهده پاسخی نوروها می خواهیم راستایی را بدست بیاوریم که در آن $p(s|r)$ تفاوت معناداری با توزیع اولیه تحریک ها داشته باشد. تفاوت این مقاله با پژوهش های قبل خود آن است که به دلیل رفتار غیر خطی نوروهای پیچیده موجود در Visual cortex نمی توانیم از آنالیزهای خطی spike-triggered average که پیش از این کارگشا بودند استفاده کنیم. این مقاله روش spike-triggered correlation analyses را پیشنهاد می دهد که بر پایه Wiener Kernel طراحی شده است.

نهایتا با این پژوهش با بررسی هایی که انجام می دهد ادعا می کند که می توان با این روش پایه ای برای تحریک ها ارائه داد که تعداد کمی از آن ها مشخص کننده ویژگی های مرتبط و تعداد زیادی مربوط به ویژگی های پوچ می باشند.

۲.۲ نوروهای «پیچیده»

این مفهوم اولین بار در مقاله نوبلیست Hubel and Wiesel مطرح شده است. نوروهای ساده نوروهایی هستند که رابطه تحریک-پاسخ آنها نسبت به زمان خطی می باشد. این موضوع باعث به وجود آمدن مناطق ON-OFF در Receptive Field نورو مورد نظر می شود. به عکس دست نوشته موجود در مقاله مذکور برای جزئیات بیشتر توجه کنید.



شکل ۱: مناطق ON-OFF حوزه دریافتی نوروهای ساده^۲

همچنین در بخش Material and Methods آمده است که اگر نسبت هارمونیک اول به مقدار DC حوزه دریافتی بزرگتر از یک باشد نورو را ساده می نامیم. روشن است که نوروهایی که از تعاریف بالا پیروی نکنند ساده نبوده و پیچیده اند.

^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 اگر 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:

$$C = \frac{1}{N} \sum_{i=1}^N S(i)^T S(i)$$

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

۳. بردارویژه‌ها 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 شده‌اند.

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 $p(\text{response} | \text{stimulus})$

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 $\text{control correlation matrix}$ را انجام می‌دهیم و میانگین مقدارویژه‌ها را محاسبه می‌کنیم. با توجه به متن مقاله حاصل این میانگین‌گیری $\pm 5/2 SD$ بازه‌ی اطمینان ما با $p < 10^{-4}$ 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 راستایست که تحریک‌های موجب اسپایک در آن واریانس و در نتیجه توزیع احتمال کاملاً متفاوتی با توزیع اولیه دارند)

۵.۲ نتایج پژوهش

همانطور که در بخش‌های قبل اشاره کردیم هدف پیدا کردن eigenvalue هایی 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 از همی دارند و همچنین correlation بسیار پایینی دارند. همچنین می‌بینیم که این راستاها به خوبی سبب تحریک نورون‌ها می‌شوند با این حال برای eigenvalue هایی 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 شده `fget_hdr.m` یکی از فایل‌های `sa0` را باز می‌کنیم و به محتویات آن توجه می‌کنیم.

hdr.DataInfo		hdr.FileInfo	
Field	Value	Field	Value
ID	'a01bmsq1D'	Type	'DAN_SPK'
DataFrom	'000412.a01bmsq1D.ra1'	Version	'1.0'
Channel	0	Fname	'000412.a01bmsq1D.s...
SampleRate	10000	Creator	'f21mlv.vi'
Gain	10000	Time	'2000,4,12,20,10,16'
DAQMode	0		
DAQResolution	0		
DataType	0		
DataUnit	0		
TimeMode	0		
TimeOffset	0		
ThreshPeakHigh	6542		
ThreshPeakLow	53250		
ThreshValleyHigh	-66541		
ThreshValleyLow	-28176		
ThreshWidthMax	-19		
ThreshWidthMin	506		

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 نرونی 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 بروزسانی مانیتور، نرخ تغییر فریم‌ها
اطلاعات مربوط به <code>tuning curve</code>	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 `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:

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

نهایتاً تابع مورد نظر به صورت زیر خواهد بود:

۱.۲.۳ قطعه کد: تابع Func_ReadData

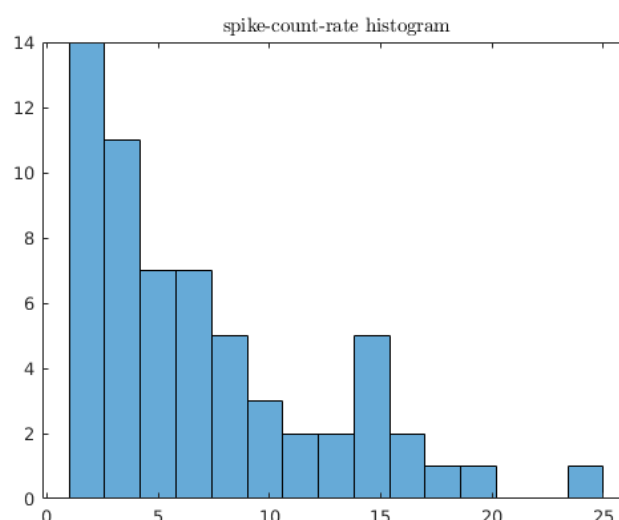
```

1 % Recomend: select Data/ and MatlabFucntions/
2 % directories and add them to matlab PATH
3 function [Output, spike_count_rate] = Func_ReadData(NeuronCode)
4     Path = [pwd, '/Data', '/Spike_and_Log_Files/', NeuronCode];
5     Listing = dir(Path);
6     Index_Counter = 0;
7     events = {};
8     hdrs = {};
9     frame_rate = 59.72; %Hz
10    frame_number = 32767;
11    spike_count = [];
12    for i = 1:length(Listing)
13        if (contains(Listing(i).name, 'msq1d.sa0', 'IgnoreCase', true)
14            && ...
15            ~contains(Listing(i).name, 'sa0.sub', 'IgnoreCase', true)
16            && ...
17            ~contains(Listing(i).name, 'sa0.', 'IgnoreCase', true))
18            Index_Counter = Index_Counter + 1;
19            [events{Index_Counter}, hdrs{Index_Counter}] = fget_spk(
20                Listing(i).name, 'get');
21            spike_count(Index_Counter) = length(events{Index_Counter});
22        end
23    end
24    Output = struct('events', events, 'hdr', hdrs);
25    spike_count_rate = mean(spike_count) * frame_rate/frame_number;
26 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. نهایتاً 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

```

1 index_counter = 0;
2 for i = 1:61
3     if SCRA(i) < 2
4         index_counter = index_counter + 1;
5         index(index_counter) = i;
6     end
7 end
8 fprintf('Excluded Neurons:\n')
9 disp(neuron_codes(index)')
10 neuron_codes(index) = [];

```

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

Excluded Neurons :

"000413.b03"

"000413.b04"

"000413.b05"

"000418.a01"

"000420.b02"

"000524.c01"

"000907.f07"

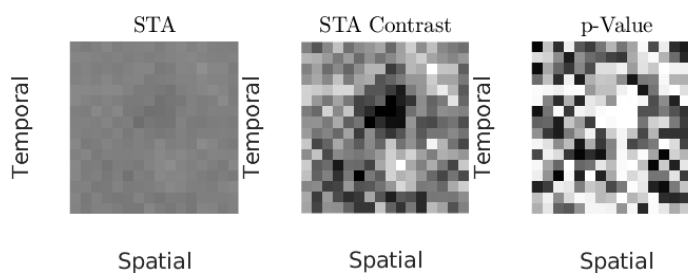
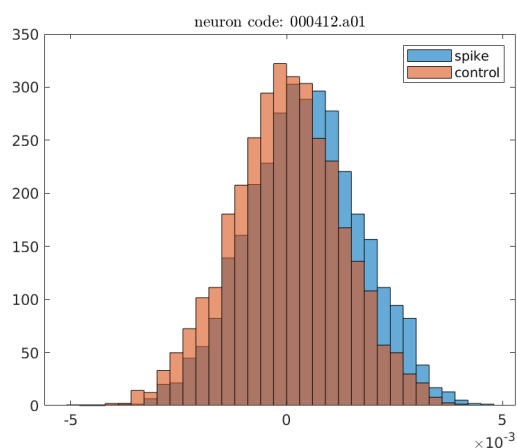
۴.۳ بخش چهارم

در این قسمت تابع `Func_StimuliExtraction` به گونه‌ای پیاده می‌کنیم که با دریافت دنباله‌ای از زمان اسپایک‌ها تحریک‌هایی با ابعاد 16×16 که سبب این اسپایک‌ها شدند را خروجی بدهد. برای اینکار باید توجه کنیم که با توجه به متن مقاله نرخ تغییر فریم‌های 60 Hz و به طور دقیق با توجه به فایل‌های لاگ $59/7213\text{ Hz}$ می‌باشد. همچنین از آنجایی که نرخ نمونه برداری 1 ms 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 16×16 باشد اندیس‌های کمتر از ۱۶ را نادیده در نظر می‌گیریم)

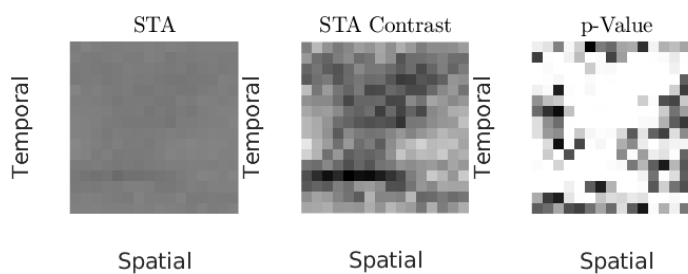
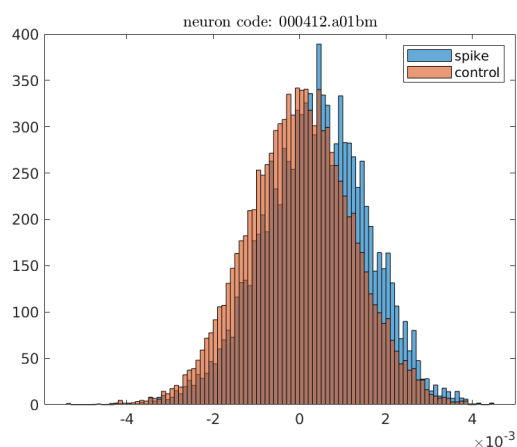
۱.۴.۳ قطعه کد: یافتن اندیس‌های زمان اسپایک و نادیده گرفتن اندیس‌های کمتر ۱۵

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

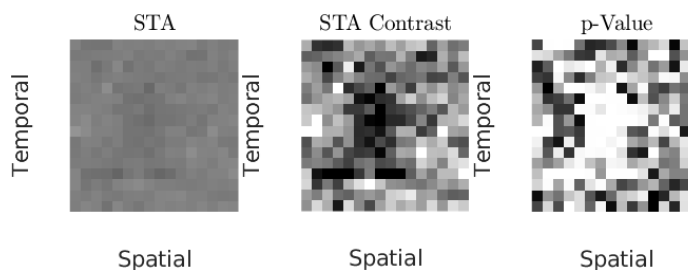
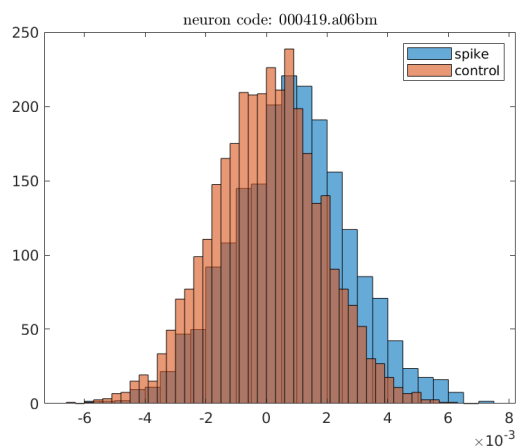
سپس می‌توانیم ماتریس‌های 16×16 که سبب اسپایک شدند را تشکیل دهیم. برای راحت‌تر شدن کار در مراحل بعد اولاً ورودی دوم ذکر شده در صورت سوال یعنی `msqlD` را اختیاری در نظر گرفتیم که مقدار دیفالت آن همان فایلی خواهد بود که در پوشه‌ی `Data` قرار دارد. همچنین در مراحل بعد از آنجایی که می‌خواهیم دنباله‌ای تصادفی 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.



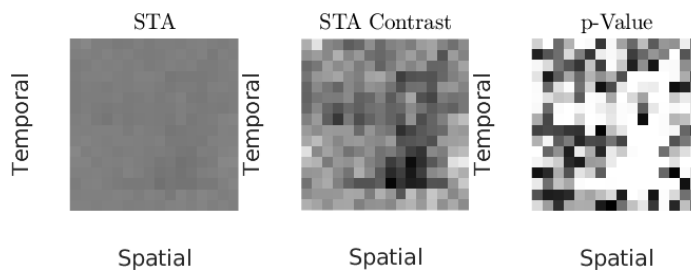
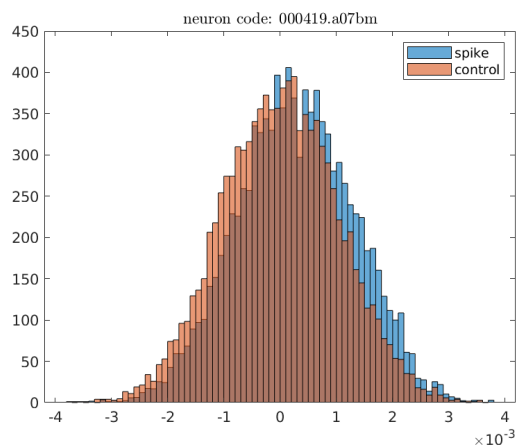
Neuron Code: 000412.a01bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 52.892081



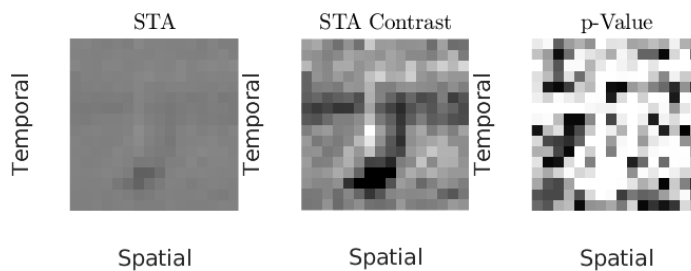
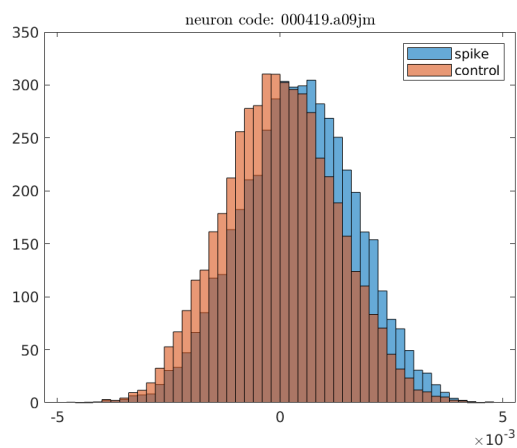
Neuron Code: 000419.a06bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 56.325007



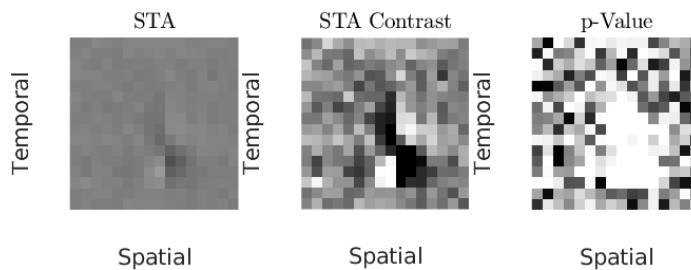
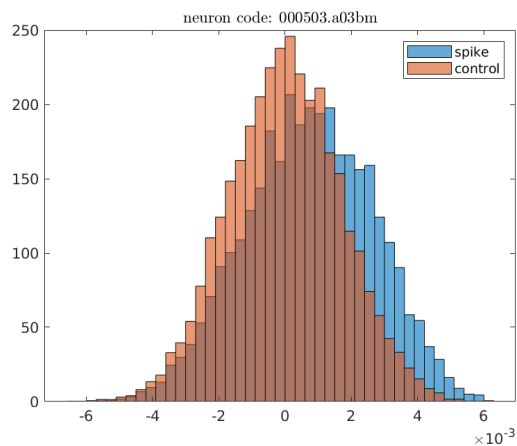
Neuron Code: 000419.a07bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 51.407035



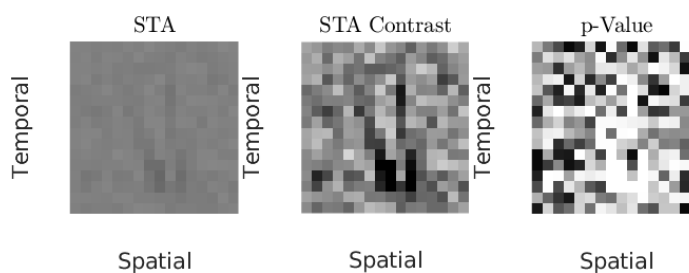
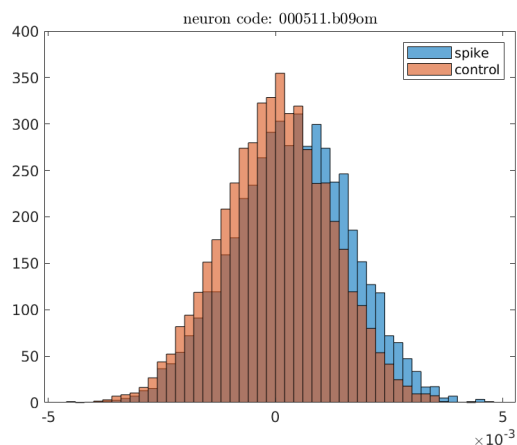
Neuron Code: 000419.a09jm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 54.320207



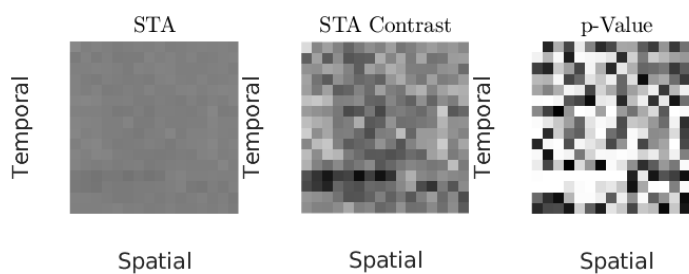
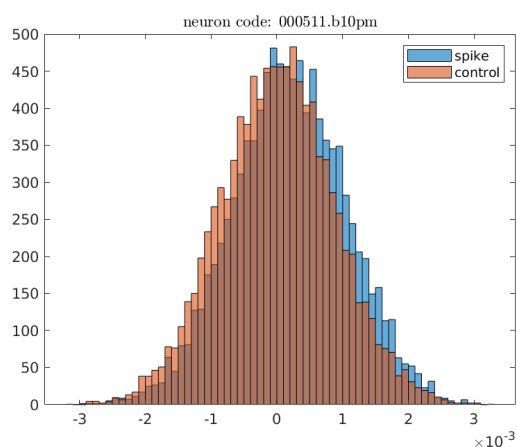
Neuron Code: 000503.a03bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 54.584792



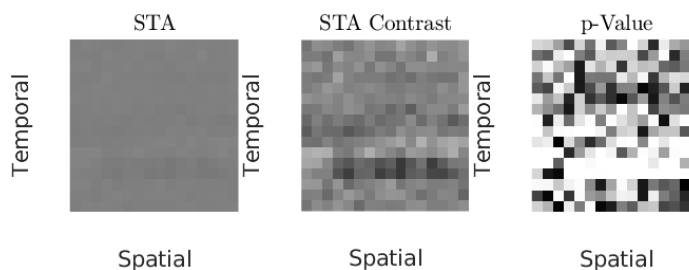
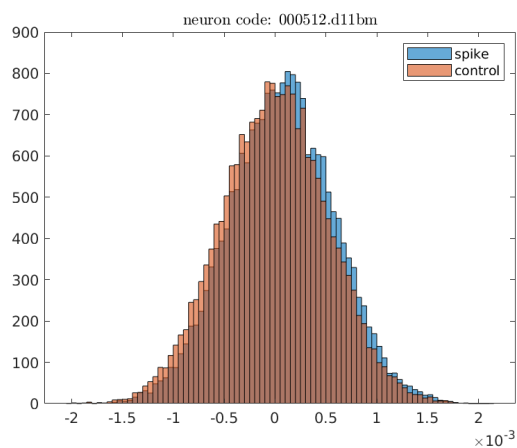
Neuron Code: 000511.b09om
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 47.656837



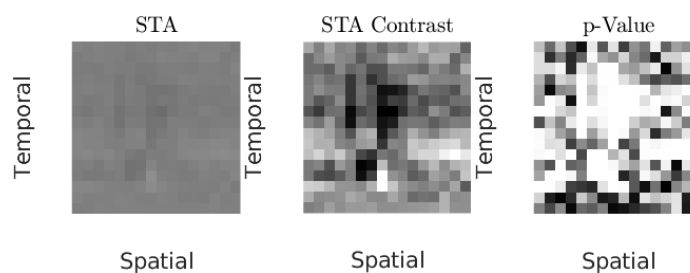
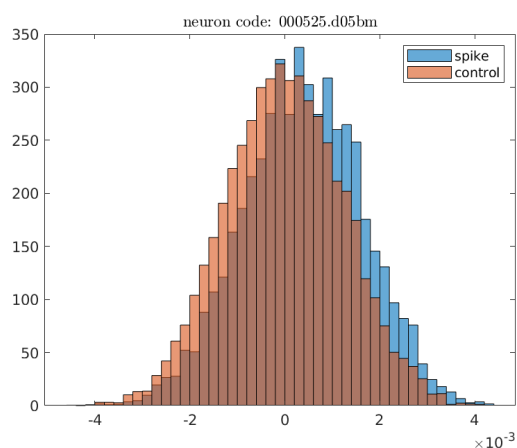
Neuron Code: 000511.b10pm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 47.605911



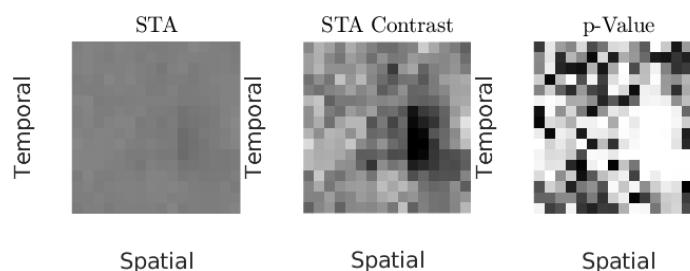
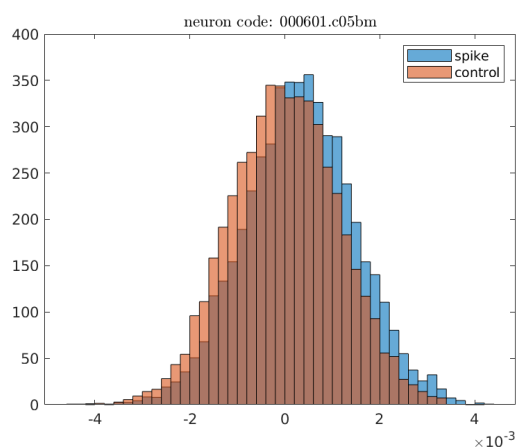
Neuron Code: 000512.d11bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 53.174895



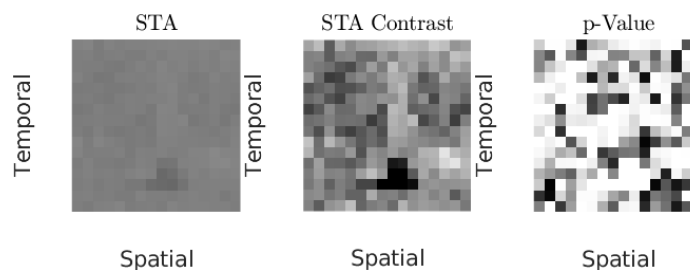
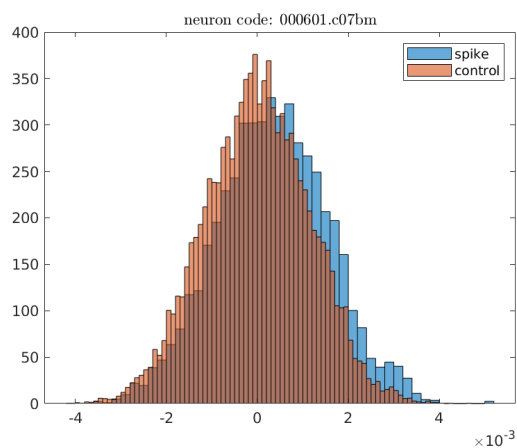
Neuron Code: 000525.d05bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 57.558372



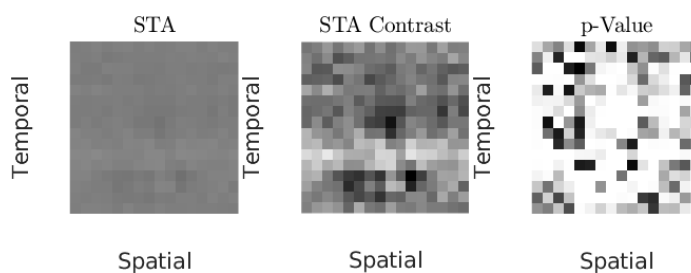
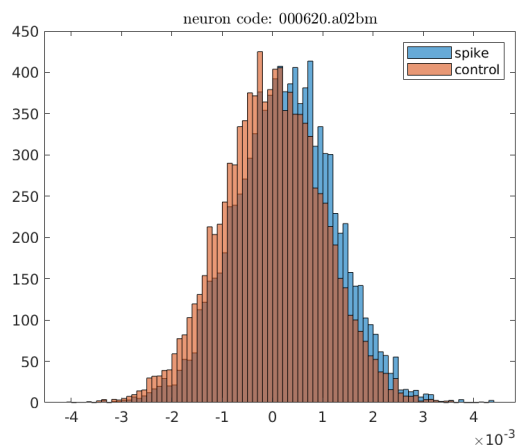
Neuron Code: 000601.c05bm
 Control/Spike p-Value: 0.000000
 t-test result: null-hypothesis rejected, means are different.
 Threshold accuracy: 55.149864



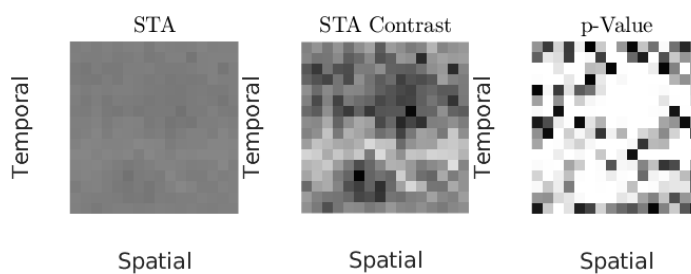
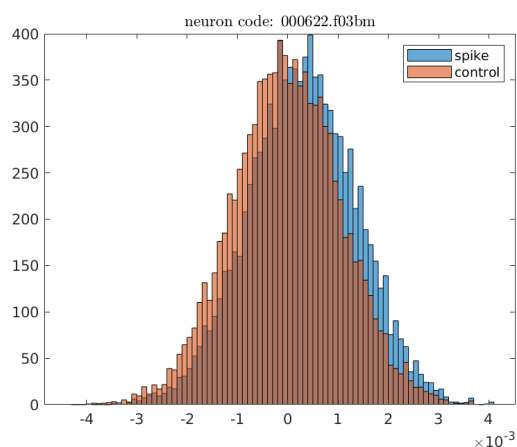
Neuron Code: 000601.c07bm
 Control/Spike p-Value: 0.000000
 t-test result: null-hypothesis rejected, means are different.
 Threshold accuracy: 53.374446



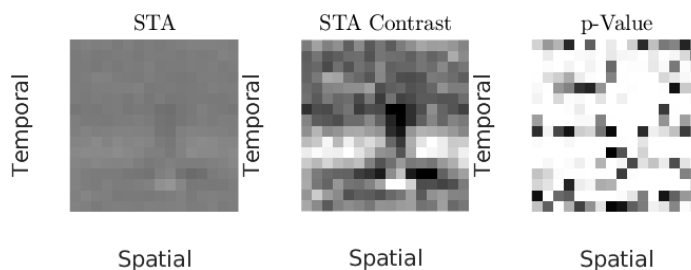
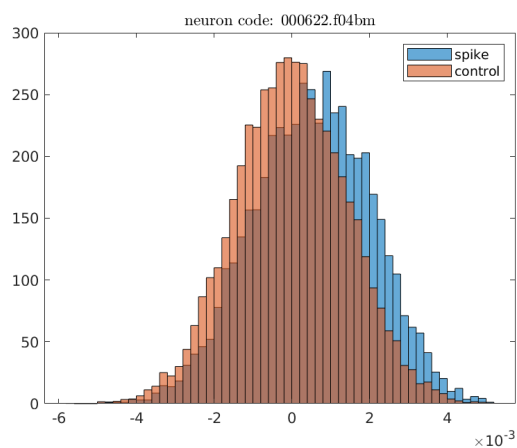
Neuron Code: 000620.a02bm
 Control/Spike p-Value: 0.000000
 t-test result: null-hypothesis rejected, means are different.
 Threshold accuracy: 47.303482



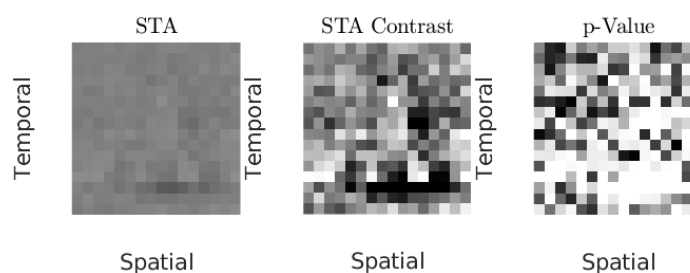
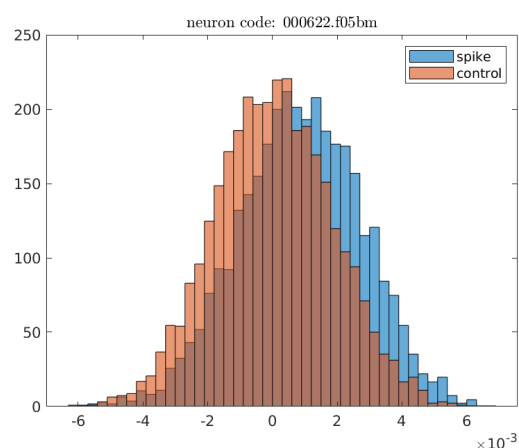
Neuron Code: 000622.f03bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 56.416876



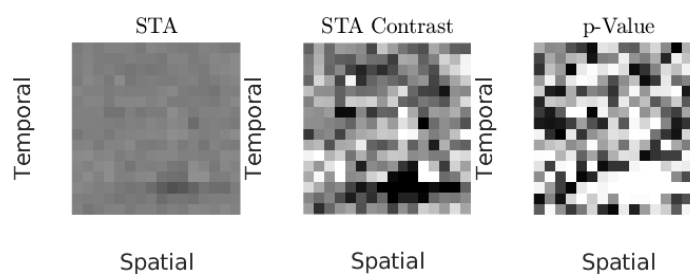
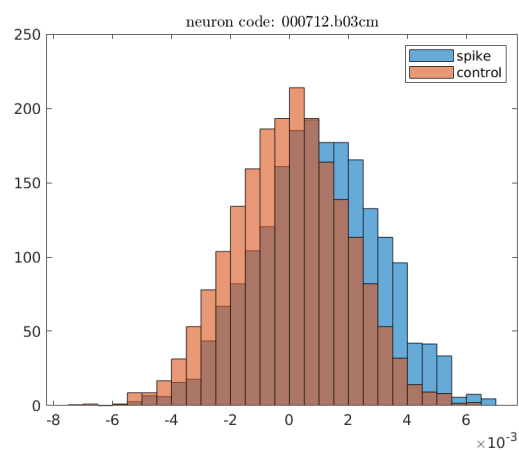
Neuron Code: 000622.f04bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 55.549866



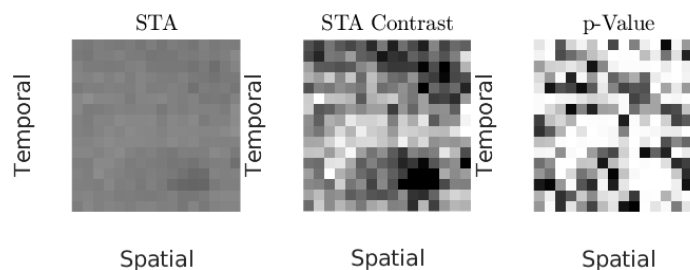
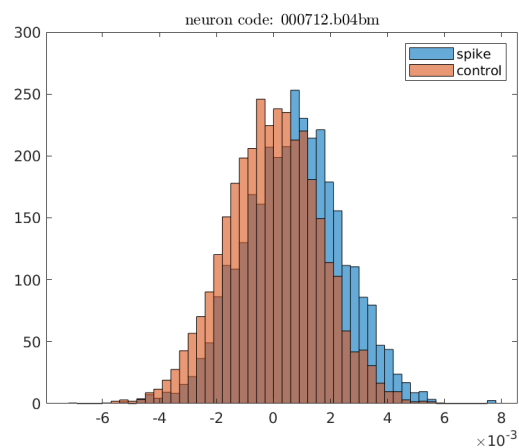
Neuron Code: 000622.f05bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 51.561590



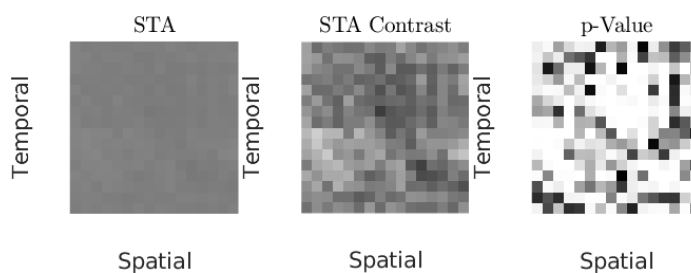
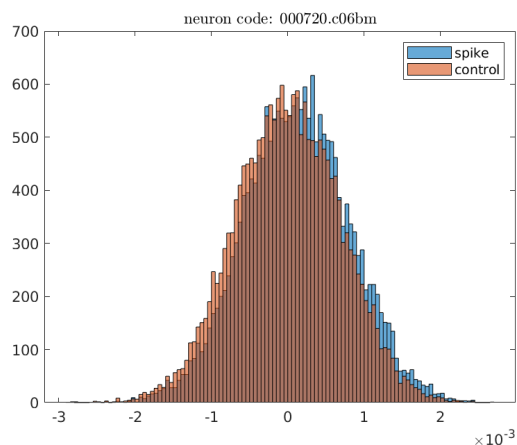
Neuron Code: 000712.b03cm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 55.392022



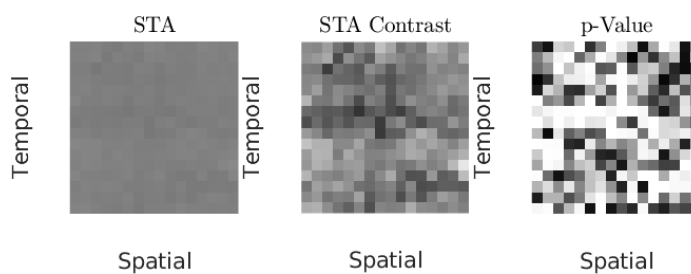
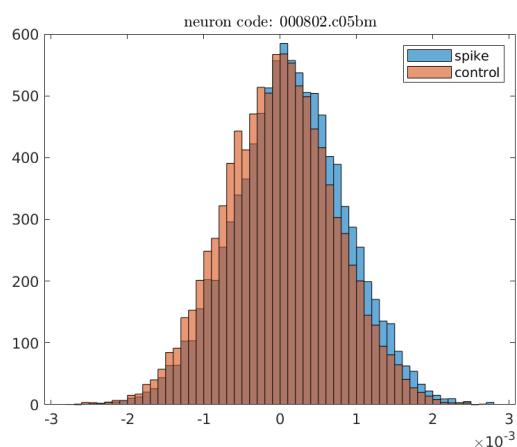
Neuron Code: 000712.b04bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 54.046535



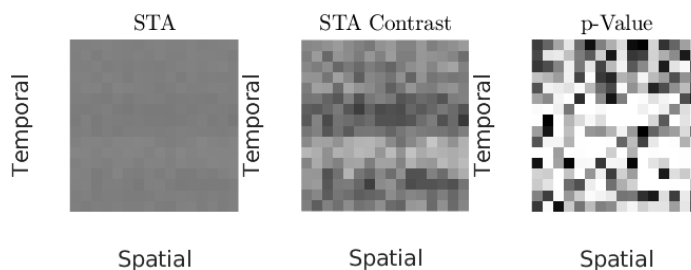
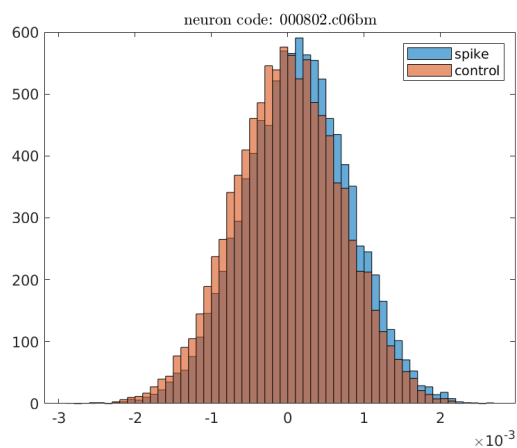
Neuron Code: 000720.c06bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 55.528077



Neuron Code: 000802.c05bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 52.037733



Neuron Code: 000802.c06bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 51.346731



Neuron Code: 000802.c07bm
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Threshold accuracy: 56.155956