Spike-triggered average – discovering the image features that make a neuron fire

This example was loosely taken from: Dayan & Abbot, Theoretical Neuroscience, MIT Press, Chapters 1-2.

MATLAB concepts covered: 1. Image plotting with subplots 2. using 'find' to compute events to be used as trigger for further analyses 3. event-triggered averages to find receptive fields 4. Random Permutation statistics to find the significant parts of images or results

In this exercise, we are dealing with the problem of discovering in an objective way what are the features of the external world that make a sensory neuron fire. Suppose for example that we have a visual neuron and we would like to discover what features of an image that we present on a computer screen make a visual neuron fire. In this regard, the simplest question to answer is to discover the receptive field, that is the part of the screen that the neuron responds to.

One experimental possibility to find the receptive field is to manually map it, by presenting some small image (for example, a little spot of bright light) in some location of the screen and listen to the neural response, then move the location around until one finds all the spots that make the neuron fire and the spots than do not. This approach may work well when the number of possible alternatives is restricted (for example, we have so a priori information about the approximate location of the receptive field and we can limit our trial-and-error exploration of the screen positions to a small region), but it may not work well in other cases when we have no prior knowledge or intuition about what these neurons may do.

In such cases, the best approach is to present a sequence of random images and then look at the neural responses that they provoke. The advantage of using a lot of random images is that these images (being random) span a wide range of possible stimulus combinations very rapidly and in an unbiased way. This approach is therefore efficient and not influenced by our own assumptions.

In our exercise, we will try to discover what image features make a simulated visual neuron fire using the technique of Spike Triggered Average (STA).
The STA addresses the question; “What, on average, did the sensory stimulus do shortly before or when a spike was fired?” Mathematically, the STA is defined as the average stimulus that precedes a spike by a given time (sum of all stimuli preceding a spike by a given time lag, divided by the total number of spikes).

In real life, a real neuron will respond to the stimulus with some latency (something like 50 ms for a primary visual cortical neuron), but for simplicity we will begin by analyzing a

simulated neuron that responds instantaneously (i.e. with zero latency) to the image.

First load in some image frames presented to the visual system and the simulated neural responses measured during the presentation of a sequence of the images:

load Images_and_spikes_nodelay.mat;
This file contains 2 variables:
'images' is the collection of the 10000 5 by 5 images black and white images that were presented to the simulated neuron. The first index of 'Images' is pixel number (from 1 to 5) on the x-axis. The second index is pixel number (from 1 to 5) on the y-axis of the image. The third index is the sequential number of the presented image frames (from 1 to 10000). The value of "Images" encodes the luminance level of each pixel (from 0 'black' to 1 'white').
'spikes' is the collection of the neural responses to ech of the presented images. A value of one means that the neuron fired one spike in response to the presented image. A value of 0 means that the neuron did not fire in response to the presented image.

Ex. 1: Make a figure showing examples of images that made the neuron fire, and make another figure showing example of images that did not.

The way to understand which feature of the image makes the neuron fire is to compare images that make the neuron fire and images that do not.
First find the image time frames that contain a spike (hint: use find to find the elements of spikes that are larger than 0). Then plot a figure with many of the images that led to a spike (hint: use subplot to plot many images together, use image plotting functions like imagesc with gray scale color map (function colormap) to plot black&white images). Then plot another figure with many images that did not lead to a spike.
What do you see? Can you build an intuition of what makes the neuron fire and where in the image is the receptive field located?

Ex. 2: Compute ad plot the Spike Triggered Average (STA).

To formalize the intuition you formed from the visual inspection of the images that the neuron likes, compute the STA by taking the mean of all the images that led to a spike. What do you see? Where is the RF is located?

Ex. 3: Compute and plot the STA at different latencies.

As we mentioned above, real neurons respond to the presented image after some time (latency). Moreover, sometimes the neurons can change or invert their selectivity depending on the time preceding the spike. To study how the selectivity of neurons evolves in time, people compute the STA as a function of the time lapsed before the spike.
The previous dataset was unrealistic because the simulated neuron responded to the

image with zero latency. We are now going to upload a new dataset of 'images' and 'spikes' that contained neural responses that are delayed with respect to the presentation of the image and have a complex dependence on the history of stimulus presentation.:

load Images_and_spikes_withdelay.mat;

You will then compute and plot the STA not only in correspondence to the time of spike generation, but you will also compute the STA a time interval before the spike was fired. Compute the STA 1 image frame before the spike was fired, 2 frames before the spike was fired,…n frames before the spike was fired…. From this plot, estimate the range of past time frames that the neuron is sensitive to, and how neural selectivity to the image changes over a range of times before the spike was fired.
Optional: define the pixels belonging to the RF by visual inspection of the STA images. Then, for each time before the spike, plot the average luminance of the pixels of the STA inside the RF. This is called the "time function" of the RF. What do you see? How does this time function help you to understand the time range in which the visual stimuli can influence neural activity?

Ex. 4: Determine quantitatively the exact location of the Receptive Field by non-parametric statistics

If you look at the STA calculated for example in Exercise 3, you will find a clear image region that looks like belonging to the receptive field, but you will also find less clear region that may or may not belong to the real receptive field. How do we separate the "real" receptive field from the noise?

One way to determine what is real and what is noise in the STA is to use a random permutation test. If we randomly permute the spike times, then any gray level value and any image structure that we find in the STA after permutation must just be noise. This is because randomly permuting the spike times destroys any relationship between images and spikes. The distribution of values of gray levels of each pixel across different random permutations can be used to estimate the values expected under the null hypothesis that that pixel has no bearing on the neural response. If the STA values in a given pixel are larger than the vast majority of the permuted values, then we can confidently conclude that this pixel influences the neuron and belongs to the RF.

Here is a description of the procedure.
Create a set of (say 1000) different random permutations of spike times. (Hint: random permutations are created by function randperm.)
Computed STA of each of the 1000 permuted spike sequences.
Sort (from lowest to highest) the values of the pixel luminance across all permutations

(hint: use function sort).
For each pixel, take the value of the 99 out of 100 (or 990 out of 1000) sorted permutation: this is the threshold for significance at p=0.99.
For each pixel check if the real STA value (computed from the original, non-permuted data) is above or below the threshold for significance. If it is above, the pixel belongs to the RF. If it is below, then it doesn't belong.

What RF do you find with this procedure? Does it make sense to you? Did this help you to separate "signal" from "noise" in your results?

Ex. 5: Create your own simulated neuron with a receptive field of choice and then discover its receptive field

Now create a sequence of random black and white images. Then select a region of these images that you would like to be the receptive field, and associate a spike to each image whose average luminance inside the receptive field is larger than a certain threshold value. Store the so created sequence of images and spikes exactly with the format that was used for the files you loaded earlier in the exercise. Then apply the analysis of Ex. 2 and 4 to these data. Can your algorithm retrieve the correct receptive field that you used to generate the data?