NENS 230: Analysis Techniques for Neuroscience using MATLAB
Fall 2011

Assignment Seven: Improving a Spike-Triggered Average Function
DUE DATE: 9am, Wednesday, November 30th

Please submit this assignment to nens230@gmail.com, with the subject line "hw7". A detailed assignment description is below, but in short, you will submit everything in your Assignment7 directory. We will be looking at your improved *SpikeTriggeredAverageStimulus.m* function as well as any additional functions you wrote that your function calls. Also include a figure *STA.fig* generated by the function after you've called it, and mention in the email your best guess of what type of RGC cell was recorded according to the classification scheme in Figure 1. You should send these files in a single (zipped) folder attached to your email called hw7_SUID where SUID is your SuNET ID (e.g. sstavisk).

In this assignment you apply the MATLAB best practices that were covered in Lecture 7 to improve a working, but poorly-written, function that computes the spike-triggered average (STA) stimulus when provided a.) data about the spike times recorded from a cell and b.) the stimulus that was being delivered at the same time. You will run the function with the provided data to see how it works, and then go through and improve the function's clarity, modularity, and performance.

The goal of this assignment is to practice enforcing good style, using descriptive names and comments, and writing computationally efficient code. By seeing the same function before and after you improve it, you should better appreciate what a difference these changes make both in clarity and performance. Furthermore, it is useful to learn how to read through bad code others have written and figure out how it works; this is unfortunately the situation you will likely frequently encounter in doing science. Now you can be part of the solution, not the problem.

Brief neuroscience background

In characterizing properties of neurons of sensory systems, a useful concept is that of the neuron's **receptive field**, which describes what kind of sensory stimuli evoke a response in the neuron. In the early visual system, neurons show a spatial receptive field which is the region of visual space (or more precisely, a region of the retina onto which light from the scene falls) where changing the stimulus will tend to cause a change in the neuron's response. For example, retinal ganglion cells (RGCs) will fire action potentials in response to a time-varying change in light intensity within their spatial receptive fields. The particular time-varying pattern is also important, and the response of a neuron to different time-varying patterns of stimulus activity is modeled by a temporal receptive field, also known as a **response function**.

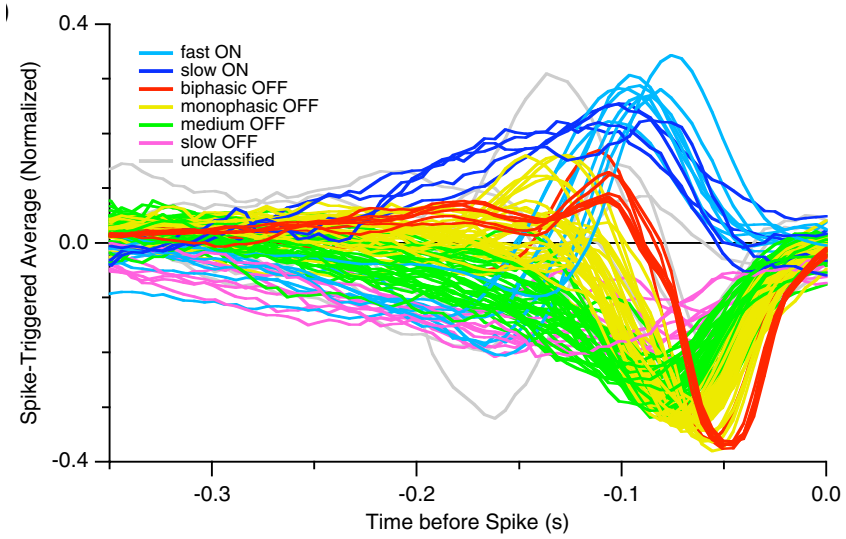

Figure 1: Spike-triggered averages serving as empirical estimates of the temporal receptive fields (also known as response functions) of different retinal ganglion cells of the larval tiger salamander. The cells' responses have been colored according to their broad functional classification as determined by the spike-triggered average.
(Adapted from Segev et al, Journal of Neurophysiology, 2006)

Intuitively, we can think of the response function as the temporal pattern of stimulus that the cell is best tuned for. So the monophasic OFF RGCs whose response functions are shown in yellow in figure 1 are most likely to fire action potentials when the illumination of their receptive field becomes briefly darker (negative deflection in the stimulus). Mathematically, we can say that the firing probability of the neuron over time is (non-linearly) dependent on the convolution of the stimulus intensity over time with the response function.

We can estimate the response function of a neuron using a simple technique called **spike-triggered averaging (STA).** To do this, a stimulus is applied to the experimental preparation (e.g., a recently dissected retina onto which light is projected) and the spiking activity of the neuron is recorded. Figure 2 demonstrates this with a snippet of the data used in this assignment. The visual stimulus intensity projected onto the retina (which covers the whole retina with the same brightness) over the course of the experiment is shown in red and the recorded spikes are shown as ticks above.

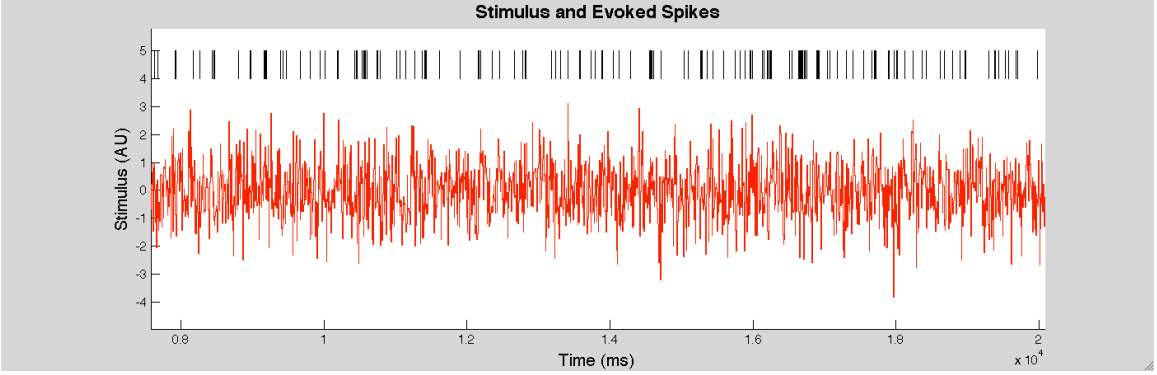

Figure 2: Snippet of the data used in this assignment. The full-field visual stimulus projected onto the retina (in red) and simultaneously recorded firing of one retinal ganglion cell (black ticks). The spike-triggered average analysis yields a response function similar to one of those in Figure 1.

To compute the spike-triggered average, we just average what the stimulus was immediately preceding every spike. Although there is considerable noise and non-linearity in the system, by averaging over what the stimulus was preceding thousands of spikes, we get an accurate estimate of the response function. This is because spikes tend to happen when the preceding stimulus was somewhat similar to the "optimal" stimulus, i.e. the response function. Because we present a random (Gaussian, in this case) stimulus it is extremely unlikely that the actual response function-like stimulus ever preceded a spike, but the average approaches this optimal stimulus. So for example if we are recording from an OFF-cell, then the stimulus will tend to have decreased just before the spike, and our STA will show a negative deflection shortly before the spike (the spike time occurs at t=0ms in these plots).

It is also important to consider how far before each spike we want to look at when computing the STA. While it is possible that whether or not a neuron spikes is dependent on what the stimulus was many seconds beforehand, in the retina this is not thought to be the case. If we look at the STA more than 300ms before each spike, the normalized stimulus tends to approach zero (i.e. the mean intensity throughout the experiment). This is what we expect to see if the stimulus that far back does not affect the likelihood of a spike because there would be approximately equal number of spikes with positive and negative stimulus intensity 300+ms before the spike. Adopting the notation of response functions and convolution from electrical engineering, we refer to the delays between the time of a spike and the preceding stimulus as tau (τ). So τ=0ms is the time of the spike, and τ=10ms refers to 10ms preceding the spike. When we compute the STA of a temporally-variable stimulus we are asking what the average stimulus was across the range of tested τ lags preceding the stimulus. In Figure 1, τ ranges from 0 to 350ms.

Detailed Instructions

You are going to fix up a working but poorly-written function that computes the spike-triggered average stimulus given the stimulus and the spike times.

Copy the Assignment 7 directory to your MATLAB directory. You are provided the STA-computing and results plotting function *SpikeTriggeredAverageStimulus.m* as well as a development helper script MAKESTA.m that just calls the function with the appropriate arguments and also times how long execution took. The spike time data is contained in the file RGCspikes.mat, and the stimulus is described in the file VisualStim.mat

Run MAKESTA.m to see the spike-triggered average be computed and displayed. When you are finished with the assignment, running MAKESTA.m should give the same result, but faster.

You should edit *SpikeTriggeredAverageStimulus.m* and possibly break it into subfunctions. Follow the best practices lecture to improve the following aspects of the code:

1. Add proper documentation, headers, and comments
2. Use sensible function and variable names.
3. Use good style; make the code neat and readable
4. Break the single function down into useful, logically-divided functions.
5. Improve the performance. Preallocation and vectorization are the lowest hanging fruit.
6. If you feel ambitious, there is a slight change in the algorithm that can really speed things up. Hint: do you really need all the intermediate results? Note: this makes only a marginal performance improvement.
7. Another hint on how to speed things up: do you need to even look at all the spikes if there isn't stimulus accompanying it? And if you didn't, would you need the conditional check that's currently in the main loop?
8. The last two input arguments, <tau> and <ignoreFirstNms>, should be optional. You can make up reasonable defaults for them. The idea behind the latter argument is that sometimes we want to not use data from the first few seconds of the experiment if we believe that the retina is still adapting to the stimulus statistics (for example, brightness and contrast adaptation).

Through optimization you can considerably speed up the execution time of this program. On Sergey's laptop the poorly-written code runs in 14.5 seconds using the data and input parameters provided, whereas the improved solution code runs in < 2s.

The *plotStimAndSpikes.m* function is not part of the intentionally badly-written code in this assignment . **YOU DO NOT NEED TO IMPROVE *plotStimAndSpikes.m.*** It's just there to help you see what the raw data you're working with looks like. When you run MAKESTA.m it will plot the stimulus intensity over time and the recorded spikes for you. You will want to zoom in using the plot tools to see what's going on.

Seeing the raw data should reinforce the take-home message that doing STA over thousands of spikes has a seemingly magic effect: you can pull out a meaningful

response function of a neuron from a seemingly random jumble of stimulus and spikes.

Extra Credit

Whoever writes the *SpikeTriggeredAverageStimulus.m* that executes the fastest when run on the grader's own machine will get a small prize. Most likely an item of delicious food or drink of the champion's choosing.