

Introduction to Computational Neuroscience

Practice session: Data Analysis - Spiking Data

Ilya Kuzovkin, Raul Vicente

March 16, 2015

A request: Please track how long it will take to complete this set of exercises. Add this time to your final report.

The last time we had a look at the continuous brain data: strength of the EEG signal was changing over the time. Another very popular form of data, mostly associated with the single neuron recordings, is *spiking* data. The concept is very simple: we attach a sensor to the neuron and when the neuron fires we write “1” into our data file, otherwise we write “0”. The resulting data shows us for each time point whether the neuron has fired or not.

Dataset

The dataset we will be working with has the spiking data recorded while the mouse was presented with a stimulus: moving bar on the screen, which can have different orientations. On Figure 1 you can see an example: white and black bars are tilted (*orientation*) and they also move in the direction of the tilt during the experiment. The spiking activity is recorded from the 72 neurons and we want to find whether different neurons react differently on stimuli with different orientation. Have a look at Figure 2 – spiking pattern of this particular neuron seems to indicate that this neuron is more active when the bar is tilted 45° clockwise from the horizontal position. And its activity fades when the bar changes the orientation. We would like to rediscover from the data, than indeed different neurons respond differently to different orientations.



Figure 1: Example of the stimulus. During the actual experiment the bars are moving.

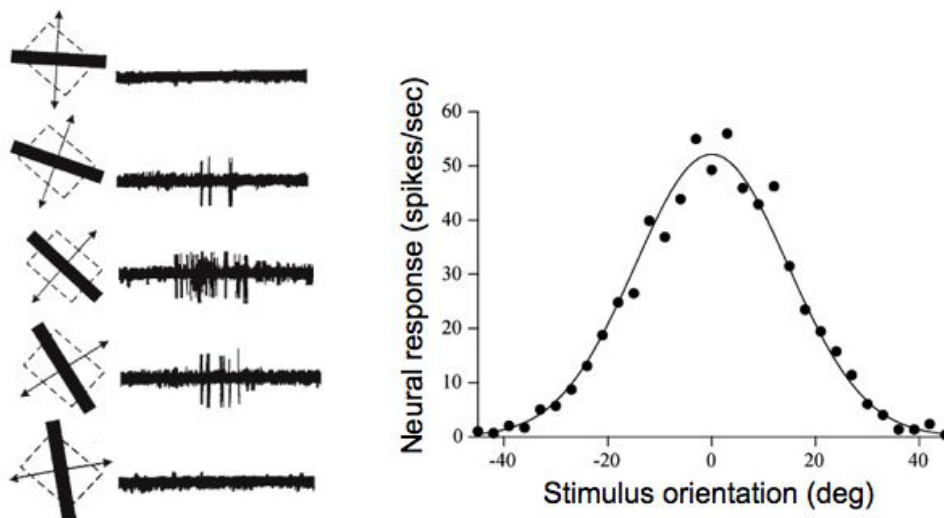


Figure 2: Neuronal response to different orientations of the bar.

NB: Note that in the dataset we have 13 stimuli (13 orientations), but the first one should be ignored (it has orientation -1°). So you should only use the other 12.

Exercise 1: Raster plots (2pt)

Let us start by plotting some spiking data. Under the `data/lgn` folder you have recordings from 72 neurons of a mouse. **Matlab/Octave** users can use files in the `matlab` folder, the same data is available in the plain text format in the `plain` folder.

Matlab files have names `mlgnori_NN.mat` where `NN` is the number of the neuron. When you load it to your environment you will find that each file has a structure with two elements: `spktimes` and `stim`. The first one is $M \times N \times T$ matrix where M is the stimulus number, N is the trial number at T is the time in milliseconds. So this is a 3-dimensional matrix with values 0 (no spike) and 1 (spike). The second file describes the stimulus. Open it and you will see what information is in there.

If you do not use **Matlab**, then it is easier for you to use the plain text files. They have names `neuron_NN_stimulus_SS.csv` where `NN` is the number of the neuron and `SS` is the number of the stimulus. Inside each file one line represents one trial. For each millisecond it has value 0 (no spike) or 1 (spike). Files with names `stimulus_SS.csv` describe the stimulus: they hold four values:

1. Time before the stimulus (in seconds).
2. Duration of the stimulus (in seconds).
3. Time after the stimulus (in seconds).
4. Orientation of the stimulus (in degrees).

Your task is to take any of the neurons and plot all trials as a raster plot (see Figure 3). You will notice that neuronal response to the stimulus varies a lot, this is why you usually need several trials.

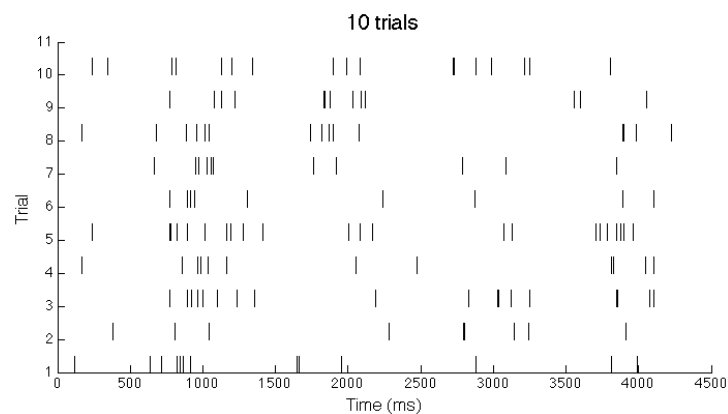


Figure 3: Raster plot of 10 trials on the same neuron.

On the X axis we have time and on the Y axis trials. A vertical bar indicates a spike. To draw a vertical line in **Matlab** you can use

```
line([t t], [tr tr+0.5], 'Color', 'k');
```

where `t` is the time and `tr` is the number of the trial.

The next step is to create two more raster plots. The first one will illustrate the behaviour of all the neurons under the same stimulus, while the other one will have responses of the same neuron but to the different stimuli.

First we need to average spiking data so that each neuron would be represented by one vector. Simplest way to do that is just to add trials together. Let us say that you have 10 trials, and each of them is a vector of 4500 time points. You just sum those vectors together to obtain one vector of length 4500. After that replace all values which are greater than 1 with 1.

Create the following two plots:

1. Raster plot, where on X axis we have time and on Y axis we have all 72 neurons. Verticals bars are the responses to the same stimulus (choose any). Please note that in our dataset different recordings have different length, but this should not be a problem.
2. Raster plot, where on Y axis we have all 13 stimuli and bars are the responses from the same neuron (choose any) to those stimuli.

Exercise 2: Tuning Curve as Rose Diagram (2pt)

From the lecture you must be familiar with the idea of a *tuning curve*. There is a really neat way to visualize a tuning curve. It fits especially well with the experiment we have at our hands. This visualization method is called *rose chart* or *angle histogram*. You can see an example on Figure 4. The idea is that the values are placed on the circumference on the circle and the length of the sector is determined by the number of times the value appears in the list. It is like a histogram drawn in circle.

In the most cases this is just a fancy way to draw a histogram, but in our case it allows to represent our data in much better way by placing orientations (stimuli) on the circle. For example on Figure 4 we can clearly see that neuron 8 reacts more to the orientation of 0° , the neuron 6 is most active in the range of 270° to 330° and so on.

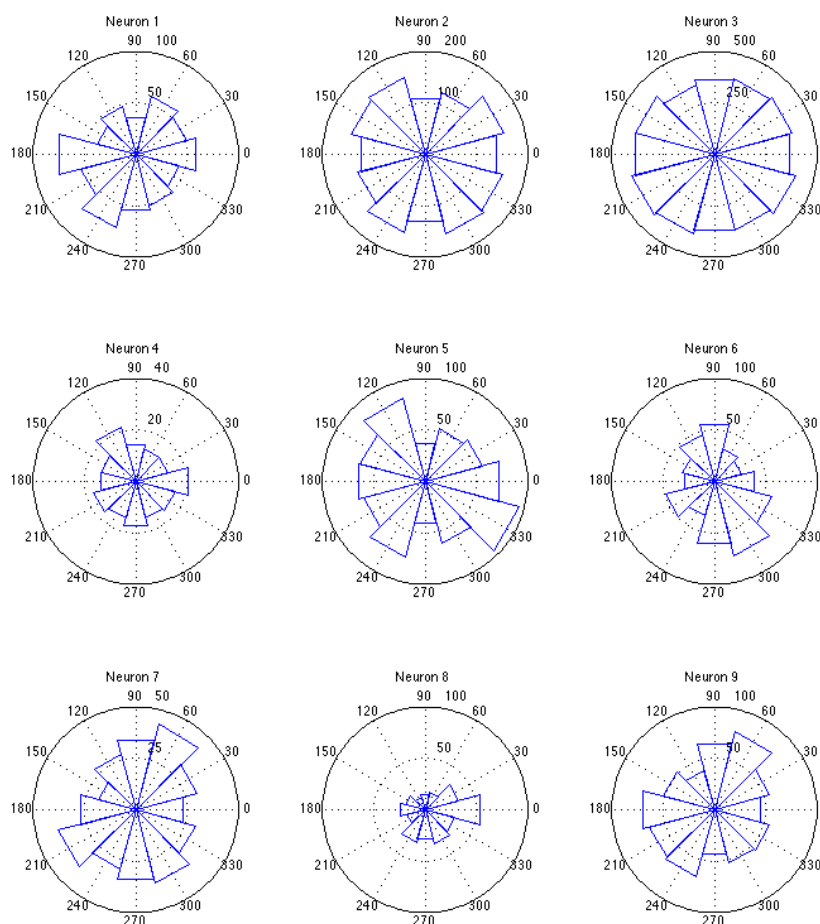


Figure 4: Rose diagram: the number of spikes for each of 12 orientations.

Your task is to produce similar plot for any 9 neurons. To produce a plot for one neuron do the following:

1. Create an array **A** where for each spike you will store the orientation when the spike occurred.
2. For each orientation (stimulus) do:
 - (a) Count the number of times the neuron fired.
 - (b) Insert current orientation (in degrees) into vector **A** as many times as the neuron has fired.
3. Draw the plot. In Matlab you can do:
 - (a) Array with the angles of bins (needed for the `rose` function):
`angles = degtorad([0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330]);`
 - (b) Call `rose(deg2rad(A), angles)`

Exercise 3: Receptive field (1pt)

In general the *receptive field* of a neuron is a region of space in which the presence of a stimulus will alter the firing of that neuron. By now you have a set of tools to analyse our spiking data and estimate the receptive fields of the neurons in the dataset we have. In our case the receptive field is the orientation of the bar on the screen (each of the neurons has a set of orientations it reacts to). Your task is to find 5 neurons with the most specific receptive fields. You can use any of the tools we have seen during this session or any other methods you find useful.

In your report please describe what you did, your findings and the list of 5 neurons with the most specific orientations.

Exercise 4: What else our neurons are tuned to? (1pt)

In this session we have seen an example of how neurons are “tuned” to respond to very specific stimuli. Your task is to find other interesting examples of stimuli our neurons are tuned to react to. Are there special neurons, which fire when you look at a human face? Neurons which react on the temperature? Hunger? Numbers?...

Exercise 5*: Post-Stimulus Time Histogram (PSTH) (bonus 1pt)

Another useful analysis tool is a histogram, where on *X* axis we have time points (or time ranges) and on *Y* axis the number of spikes that occurred during given time range. It is called *Post-Stimulus Time Histogram (PSTH)*.

1. Choose any neuron, any stimulus.
2. Take an average over all trials as we did before.
3. Choose time window, for example 20ms or 50ms.
4. Plot a histogram, where on the *X* axis we have time windows and on the *Y* axis the number of spikes that occurred during that window.

Please submit a **pdf** report with answers to the questions and comments about your solutions. Include figures, explanations and essential pieces of code. Do not include the code itself as a separate file, your report should give good understanding of what you have done. Please mark how long it took to complete this set of exercises. Upload the **pdf** to the practice session page on the course website.