Introduction to Matlab

Pouyan R. Fard and Dario Cuevas

13. Januar 2016

Import the data file FiringRates.mat. Each row is a different neuron; each column a time point.

1. Plot the firing rate for each neuron in a separate subplot, such that all are plotted on their own „square“. You can decide the exact arrangement of squares; however, your code should work for any number of neurons, not just the 30 in the .mat file.
2. The lines should be black
3. Every subplot should have the same range. They should all go from zero to the maximum in the data file on the Y axis.
4. Add axis labels to the first subplot. They should be 'time' and 'firing rate'.
5. You should include a red line at height y=1 (see the picture). Use the command line for this.
6. Calculate the average population activity (of all 30 neurons) and plot it at the bottom of the figure, as shown in the picture; use a thick, black line. Give this plot a fitting title and axis labels 'time' and 'Population firing rate'.
7. Getting into the advanced stuff. Using get and set, remove the axis ticks on all the subplots except the first one.

Using the same data from the previous part, do the following:

1. Create a vector vT with 4 elements. The first is 1, the last is the number of time-points in the vector Y. The two in the middle should be equally spaced. (hint: use linspace)
2. In a 1x4 subplot in figure(1), plot 4 time-slices of the Y data using imagesc. That is, 4 plots, each one of them being the activity of all 30 neurons in one time-step (given by vT). The result should be like in Figure 1.
3. Now, in figure(2), repeat what was done in figure(1), but rearrange the elements in these time-slices in a 6x5 array. See Figure 2.
4. We will now designate each neuron as either active or inactive. Any neuron whose firing rate is below 5 is inactive; above 5 and it's active. For each one of the time-slices from part 2, change the value for each neuron to 0 for inactive, 1 for active. The easiest way to do this is with logical indexing; for example, for a matrix A = magic(5), A(A>5) will give you all the values of A that are bigger than 5. A(A>5) = 1 will change all the elements of A that are bigger than 5 to 1; those smaller than 5 are left as they were. Plot the new time-slices with 0s and 1s using imagesc in figure(3). See Figure 3 to see what you should obtain.
5. Finally, in figure(4) plot the covariance matrix for these neurons, using imagesc. The covariance matrix can be obtained with the function cov. The correlation matrix can also be obtained using the function corrcov. Plot it in figure(5).