Problem set 8

Analysis for Neuroscientists

1. In the next class, we will encounter the *general linear model*, in which we relate dependent variables $\mathbf{Y}$ to independent variables $\mathbf{X}$ according to the equation:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U}.$$

   Assume

$$\mathbf{Y} = \begin{pmatrix} Y_{11} \\ Y_{21} \\ Y_{31} \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \\ X_{31} & X_{32} \end{pmatrix}, \ \boldsymbol{\beta} = \begin{pmatrix} \beta_{11} \\ \beta_{21} \end{pmatrix}, \ \mathbf{U} = \begin{pmatrix} U_{11} \\ U_{21} \\ U_{31} \end{pmatrix}.$$

   Write down the expression for Y_{11}, Y_{21}, and Y_{31} in terms of the other variables using the rules of matrix multiplication and addition.

2. In prob2.mat there is a variable spikes. spikes{c} contains a 500×100 array of 0s and 1s representing the spikes of neuron c (3 neurons total) during 100 trials of 500 ms each. We're going to compute cross-correlograms (CCGs) for different pairs of neurons to see which ones are connected.

   The function calcccg(s1,s2,Nlags,dt) will compute the cross-correlogram given s1 and s2, which are vectors of 0s and 1s representing spikes. Nlags is the number of bins to the left and right of 0 to be included. For this problem, you can use Nlags=20. A peak to the right of 0 means s1 tends to fire after s2.

   (a) Compute the average (over all 100 trials) cross-correlogram for a pair of neurons by using calcccg and averaging the result over the trials. You can base your code on this snippet:

```
load('prob2.mat');

Nlags = 20;
t = dt*(-Nlags:Nlags);

ccg = zeros(length(t),1);

%fill in code here
```

```
plot(1000*t,ccg); %factor of 1000 to convert seconds to milliseconds
xlabel('Lag (ms)');
ylabel('Cross-correlation (Hz)');
```

(b) Do this for all pairs of cells. Which pairs seem to have a synaptic connection?

(c) For one of the connections you identified in the previous question, remake the CCG but by shuffling the trials of one of the neurons. You can use `randperm(Ntrials)` to get a randomly reordered vector of trial numbers. Plot the shuffled CCG on top of the true CCG.

References