Exercise 2– spike train analysis
Please submit by March 12th 2024.

The directory /data/ contains data from neurons recorded during an eye movement direction tuning task. Each file contains the information from a single neuron, organized in a Matlab structure.
The name of the structure is data. It contains the following fields:

- spikes: a T X N sparse matrix, when N is the number of trials and T is the time in ms (milliseconds). The matrix indicates the times of the spikes in each trial. A value of 1 indicates that the neuron had an action potential in the corresponding bin.
- target_direction: a vector containing the direction in degrees of the target motion.
- target_motion: The time of target motion onset and offset in ms.

Questions (50% of the grade)

Please submit graphs with a very short explanation, do not submit the code.
Choose a single data file from the folder /data /

1. Write a script that shows the raster plot for each target motion direction separately. Present the raster plots in a single figure (Hints: Matlab function spy).
2. For each direction calculate the PSTH in a 1 millisecond (ms) bin. Smooth the PSTH with a square window of 30 ms. Normalize the PSTH to firing rate in units of spikes/s. Present all 8 PSTH in a single Matlab plot. The presented traces should not contain edge effects (Hint: Matlab function smooth).
3. Calculate the direction tuning of the neuron. Calculate the average firing rate across all the movement times as a function of the direction of movement. Present a plot of the average firing rate across trials as a function of motion direction.
4. Fit the direction tuning to a cosine (f1) and von-Mises function (f2):
   - $f_1(direction) = a_0 + a_1 \bullet \cos cos\left(direction - a_2\right)$
   - $f_2(direction) = a_0 \bullet e^{a_1 \cdot cos(direction - a_2)}$

   a. Present the fit together with the direction tuning of the cell. Use a resolution of 1 degree to present the results of the fit.
   b. How can you determine the goodness of the fit?
   c. Use the fit to estimate the preferred direction of the neuron.
   d. How could you estimate the preferred direction directly from the data without fitting data to a function? What are the advantages of using a fit and of estimating directly from the data?
   e. Which of the functions (von mises/cosine) fits the data better? Base your answer on the goodness of fit you suggested in 4b.
   f. Add a baseline term to the von mises fit i.e. fit the following function:

   $f_3(direction) = a_0 e^{a_1 cos(direction - a_2)} + a_3$.

   Present the three fits together with the direction tuning of the cell. Is the fit better than the other fit?

g. Is the comparison between the quality of f3 and the other fits fair? What might be the disadvantage of using more parameters?
h. Calculate the tuning curve of two additional neurons from other data files. Present the direction curves of each cell in a single figure. Are the tuning curves the same? In what features of the tuning curve are the cells different? (Hint: Matlab functions fittype, fit and cosd).

<u>Main project (50% of Exercise grade + 2 bonus points to final grade)</u>
<u>Please submit:</u>

1. The Matlab function by Moodle with your answers.
   Name the function classifier_[your name].m
2. A short explanation (less than half of a word page) about the algorithm you used.

The goal of this part of the exercise is to decode the direction of the target motion from the spike train. You are required to write a classifier that receives as input a training set and to classify the direction of motion in a test set. The classifier should use the information from the training set to predict the direction (in degrees) of motion in the test spike trains.

<u>The function you write should have the following specification:</u>
target_dirs = my_classifier (train_data, test_spikes)
<u>Inputs</u>: train_data : Has the same format as the structure you loaded in the first part of the assignment (spikes, target direction and target_motion, see above)
test_spikes: a T X N sparse matrix, when N is the number of test trials you need to classify and T is time in ms. Each column of the matrix is a separate unclassified trial.
<u>Outputs</u>: target_dirs: a vector of the estimated directions. The i-th entry should correspond to the i-th column in the test_spikes matrix.

We provided a few functions that might help you in the process of testing your classifier:

- [success _angle, success _id] = test_classifier (classifier_func, data)
  The inputs to the function are the classifier function you need to write and the data from a single neuron. The function uses k-leave out cross validation to test the classifier. It splits the data into train and test data sets to test the quality of the classifier and returns two values:
  - success_angle: the average angle difference between the classification output of classifier_func and the actual data.
  - success_id: The fraction of trials in which the classifier has correctly identified the motion direction.
- classifier_random.m
  You could test the function test_classifier by running a classifier that randomly assigns a direction to a spike train. The random classifier is implemented in the file classifier_random.m. Load data from one of the neurons, then run the command:
  [success_angle, success_id] = test_classifier(@classifier_random, data).
  If everything is working fine you should get values around 90 for success_angle and 0.12 for success _id.

- run_test.m
  You could (if you wish) use the script run_test.m to test your classifier on all the neurons. Your goal is to maximize the success_id and minimize the success_angle. To test your classifier insert it to a directory named /classifiers/ then add the function to the list of classifiers in the variable named “class_func” (~line 9 in the code).

  After you submit your code we will test your classifiers. Classifiers that will be better than a classifier we wrote in both measures will add a bonus of 2 points to the final course grade. Importantly, we will run the test on a dataset that you cannot see, therefore, overfitting might reduce the accuracy of your classifiers.

  Good luck,