BMI Spring 2019
Neural Decoding Competition Rulebook

Description of the Competition:

In the coming weeks you will design a neural decoder designed to drive a hypothetical prosthetic device. This is a realistic and difficult task in brain-machine interfacing. The task will take the form of a competition, to make it even more enjoyable and to encourage creative solutions.

You will be provided with spike trains recorded from a monkey's brain, as he repeatedly performs an arm movement task. Your algorithm will estimate, from these data, the precise trajectory of the monkey's hand as he reaches for the target.

This is a continuous estimation task. Over the course of each spike train, you must estimate the X & Y position of the monkey's hand at each moment in time (while the training data includes the Z position, you are not asked to estimate it). Caveats: “time travel” is banned – i.e. your decoder must be causal, it may not use information from the future to estimate the position of the hand at the present. When your algorithm is tested, it will be given data in 20 ms chunks in order to enforce this.

Data:

You will be given training data in the form of a MATLAB data file. The training data is currently online on Blackboard, titled monkeydata_training.mat. A detailed description of the data is also on Blackboard, under "Description of the data".

Implementation:

The competition will be run in MATLAB. While you are welcome to use other programming languages for your own edification, non-MATLAB submissions will not be graded.

Your submission for each task will take the form of a MATLAB function. The function will accept two arguments: a set of training data (that includes reaching angles), and a set of test data (which only has spike trains, none of which are included in your training data). Your function will then return the appropriate classification or estimation, respectively.

You will be given a template, including a function declaration with a description of the arguments it must accept and the data structure it must return. You may change the variable names but not the function name nor the order of the arguments.

Algorithm submission process:

- Please submit your training function along with your position estimator function (i.e. two separate functions) as two separate files, preferably in a zip file. Please name the .zip file with your team name.
- You will need to make sure your functions work. You can test this by running it through the test functions provided on Blackboard. The format required is further described in the template functions.
- Your algorithm must take no longer than 5 minutes to run, as tested.

Teams:

This competition will be a team effort. Each team will consist of 3-4 students (we will accept exceptionally groups of 5 if well justified). You are encouraged to form the teams on your own but we are happy to provide help in forming the teams. Any serious grievances or disputes regarding teams should be addressed to the GTAs or lecturers as early as possible.

Each team will submit only ONE final file for each task. Each file will have ALL of the team members' names written at the very TOP of the file.

Competition Scoring & Prizes:

There will be a small prize in this contest, which the team with the lowest error will win. The prizes are highly desirable though as-yet unspecified.

The following error metric will be used to evaluate the performance of the submissions:

Your error will be the root mean squared (rms) error of the trajectories you compute. The mean will be taken across both dimensions, over all trajectories. The units of this error is in cm.

Inviolable Laws:

- You MUST cite every resource you use.
- Your submissions MUST implement the interfaces in the MATLAB templates exactly as specified
- You MUST put the names of ALL team members on the top of EVERY file you submit
- While you may use existing (cited) algorithms, all submitted MATLAB code MUST be entirely your own. Exceptions are the MATLAB template files, any code posted by the lecturers or GTAs on Blackboard, and the calling of functions in standard MATLAB toolboxes.
- Late submissions will be ignored. Your submission may be updated online, so it is suggested that you submit early and then continue updating your submission until the deadline if you so desire.

Submission Deadlines:

Algorithm – 11 am GMT, Thursday 21st March 2019.
Group report – 5pm 29 April 2019

Advice:

Feel free to use whatever algorithms you like. In particular, you are encouraged to attempt to design your own. You have also been exposed to several algorithmic ideas throughout the lectures and journal clubs, and you may mine research papers, professors, textbooks, etc. The sky is the limit. You may NOT, however, used copied code. All code submitted must be your own.

Be warned, however, that more mathematical artillery doesn't necessarily result in a better solution. Consider implementing a simple idea early on and then taking time to improve on it, rather than using up all your time designing a complicated but untested idea. Remember also that the ideas described in research papers rarely work exactly as specified.