Creating spike trains using MATLAB for (Java) pyramidal neuron simulation.

LSS 19 February 2019.

The Java pyramidal neuron simulation expects spike trains(both driving and contextual) to consist of csv files, 3 values per line, read as doubles: neuron number, synapse number, time. This is how to create such files, for multiple input spike trains with different neurons and synapse numbers using MATLAB. The .m files used reside in Documents/MATLAB/Spikes.

1: Decide on the spike trains to be created. For each you need a minimum and a maximum spike rate (can be the same), a period of time before they start (default is 0.1 seconds), a duration (in seconds) and whether or not there is to be temporal jitter. Use createSpikeTrain.m to create each of these.

Example:

d2 = createSpikeTrain(40,50,5, 'jitter', 0.005) ;

creates a 1-D array of times of spikes starting at 40Hz, finishing at 50 Hz, lasting (about) 5 seconds after the initial 0.1 second silence, with the output times jittered by a Gaussian with standard deviation 0.005 seconds. Note that the spikes will always be strictly increasing in time.

2: Turn each spike train into a .csv file with appropriate neuron number and synapse number, using makeCSVarray.

Example:

d2c = makeCSVarray(d2, 1, 2) ;

turns the 1-D spike train d2 into an N by 3 array

<1 (neuron number), 2 (synapse number), spike\_time>

3: Put all the N by 3 arrays together, and sort them into spike time order using vertcat and sortrows.

Example: suppose we have

>> d1c = makeCSVarray(d1, 1, 1) ;

>> d2c = makeCSVarray(d2, 1, 2) ;

>> d4c = makCSVarray(d4, 2, 1) ;

creating 3 N\_i by 3 arrays

We turn this into a single sorted array by

>> dfinal = vertcat(d1c, d2c, d4c) ;

>> dfinalsorted = sortrows(dfinal, 3) ;

First lines creates the concatenated array, and the second one sorts it by time

4: Write the concatenated array out to a .csv file to be read by the Java spiking network simulator.

>> fp = '/Users/lss/Documents/workspace/PyramidalCells/Test\_nov2018/' ; % directory to be written to

>> csvwrite([fp 'temp\_c.csv'], dfinalsorted) ; % write the file out to temp\_c.csv in that directory