Laboratory Assignment 1 (LAB1)
Bonus Track Assignment#2: Implementing Liquid State Machines (LSMs)

The goal of this assignment is to implement a Liquid State Machine using the Izhikevich model as a liquid. The learning system shall be experimentally validated on a small version of the Laser dataset (a common benchmark for learning in the temporal domain).

Izhikevich's Model

[1] E.M. Izhikevich, "Simple model of spiking neurons." IEEE Transactions on neural networks 14.6 (2003): 1569-1572.
Available online at: http://izhikevich.org/publications/spikes.pdf
Web page: http://www.izhikevich.org/publications/spikes.htm

IMPORTANT (use it, customize it)

The Liquid of the LSM should be implemented by using the pattern of connectivity and the parameters (at least initially) following the code by Izhikevich, which can be found at
http://www.izhikevich.org/publications/net.m
and a variant to it that you can find at
https://www.dropbox.com/s/7lkcskfdzb1cfw3/modified_lsm.m?dl=0
(you will need to vary, e.g., the number of excitatory and inhibitory neurons)

[2] E.M. Izhikevich, "Which model to use for cortical spiking neurons?." IEEE transactions on neural networks 15.5 (2004): 1063-1070.
Available online at: http://izhikevich.org/publications/whichmod.pdf
Web page: http://izhikevich.org/publications/whichmod.htm

Matlab documentation

Matlab User's Guide https://www.mathworks.com/help/index.html
Matlab documentation using the help command (e.g. `help train`)

Assignment – Laser Task (reduced)

The Laser task consists in a next-step prediction (autoregressive, a particular case of transduction) on a time series obtained by sampling the intensity of a far-infrared laser in a chaotic regime.

Import the dataset from MATLAB (`load laser_dataset`). You will need to convert the data format from cell to mat using the command cell2mat.

Split the data into input and target signals. As this is an autoregressive task you will need to refer to the same time-series. For example, if the laser data is in the variable whole_dataset, you can set input_dataset = whole_dataset(:,1:end-1), and target_dataset = whole_dataset(:,2:end).

Properly separate input and target data, then split the available data in

- training (first 1500 time steps), and
- test set (the subsequent 500 time steps).

Note that this leads to use a total number of 2000 time-steps (while the original dataset contains 10093 time-steps). Note that for model selection you will use the data in the training set, with a further split in training and validation. Try to plot the time series data using the command `plot`.

Solve this regression task with a LSM, where:

- The liquid is given by a layer of interconnected Izhikevich neurons, where the input is added as external applied current
- The readout is a single neuron, trained by pseudo-inversion. Suppose that the liquid states for the training samples are stored in the variables trainStates, and the corresponding target output values are in the variable trainTargets, then the readout weight matrix can be computed in one go as:
  `Wout = trainTargets * pinv(trainStates)`
- Now it is possible to compute the output of the LSM as, for example:
  `trainOutput = Wout * trainStates`
  and analogously for the test time-steps.

Use a hold-out model selection scheme to choose a suitable number of excitable and inhibitory neurons in the liquid (and other hyper-parameters such as the scalings of the input and recurrent connections).

The output of this bonus track assignment should then consist in the following data, pertaining only to the selected hyper-parametrization:

- The script .m file(s)
- Training, validation and test errors
  The error function to consider should be the Mean Absolute Error (MAE)
  e.g. `mean(abs(outputTraining-targetTraining))`
- Target vs output plot (i.e., 2 lines one for the target and one for the output for increasing time), both for training and test data (.fig or .png file, a total number of 2 figure files)