Laboratory Assignment: Echo State Networks for temporal data processing (Lab3-2, 2019)

Solve the following assignment, whose completion is required to access the oral examination. Upload the assignments all-together in the Moodle platform of the course (once you have completed all the labs, not only this single one) as a compressed folder including one subfolder for each laboratory.

The subfolder for this lab should be called "LAB3_2" and should include the Matlab scripts and the other files as requested in the assignment below. You can organize the code as you wish, implementing all the helper functions that you need, provided that these are included in the subfolder and are appropriately called in the scripts.
Properly organize the files requested for the different assignments into different sub-folders. E.g. "LAB3_2/Assignment1" for the first assignment, "LAB3_2/BonusTrack1", etc.

Bonus track assignments are meant to be for those who finish early, but they are not formally required for completing the Lab Assignment.

Useful documentation:

Notes on Echo State Networks and Reservoir Computing (course lecture): file "Part3 Lecture Reservoir Computing.pdf" in the Moodle platform (section Lecture Notes).

Additional material for this assignment: file "Lab3_2-AdditionalMaterial.pdf" (containing info on initialization and training of ESNs)

Matlab Neural Network Toolbox User's Guide http://it.mathworks.com/help/pdf_doc/nnet/index.html
Matlab documentation using the help command (e.g. `help train`)

Assignment 1 – NARMA10 Task

This task consists in predicting the output of a 10-th order non-linear autoregressive moving average (NARMA) system. Further information on this task can be found in the paper *Gallicchio, Claudio, and Alessio Micheli. "Architectural and markovian factors of echo state networks." Neural Networks 24.5 (2011): 440-456.*

The input of the system is a sequence of elements $u(t)$ randomly chosen according to a uniform distribution over [0, 0.5]. The output of the target system is computed as follows:

$d(t) = 0.3\, d(t-1) + 0.05 d(t-1) \sum_{i=1,\dots,10} d(t-i) + 1.5\, u(t-10) u(t-1) + 0.1.$

Given the input value $u(t)$, the task is to predict the corresponding value of $d(t)$.

Import the dataset from the MATLAB file NARMA10timeseries.mat. You can find the input and target time-series respectively in the fields *input* and *target* of the imported structure. Split the data into training (the first 5000 time steps), and test set (remaining time steps). Note that for model selection you will use the data in the training set, with a further split in training (first 4000 samples) and validation (last 1000 samples). Try to plot the time- series data using the command `plot`.

- Create (implementing from scratch the equations) and train Echo State Networks using different values of the hyper-parameters (input scaling, number of reservoir units, spectral radius, readout regularization for ridge regression; optional: percentage of connectivity among reservoir units, etc.). As regards the hyper-parameters, you can either (manually) choose a set of values to consider, either (systematically) use a grid, or a random search method.
- For each hyper-parameterization that you consider, the performance (on training, validation and test sets) should be averaged over a number of *reservoir guesses*, i.e. different random instances of ESNs with the same values of the hyper-parameters (only the seed for random initialization changes among the guesses).
- Select the best network hyper-parameters on the validation set, the hyper-parameterization with the smallest Mean Squared Error (MSE), e.g. by using the command `immse`.
- Train the selected network on all the training data and evaluate the MSE of such network on the training set and on the test set.

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

- The script .m file(s)
  - This file should also include a specification of the set of the hyper-parameters values considered for the model selection (if not already in the grid).
- All the Echo State Network structures (variables) corresponding only to the selected hyper-parametrization (e.g. a structure for the reservoir and a structure for the readout). Such structures must include all the weight matrices of the Echo State Network architecture, thereby including input-to-reservoir weight matrix `Win`, recurrent reservoir weight matrix `Wr` and reservoir-to-readout weight matrix `Wout`.
- All the hyper-parameters values of the selected Echo State Network, thereby including: number of reservoir units, input scaling parameter, spectral radius, readout regularization parameter (and the values of all the other hyper-parameters that you considered).
- Training, validation and test Mean Squared Error (e.g. by `immse` command).
- A plot (fig or png) with the target and output signal (use the hold on command to have a comparative plot). This type of plot is required for both training and test.

Bonus Track Assignment #1– Laser Task

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`), rescale values to [-1,1], properly separate input and target data, then split the available data in training (first 5000 time steps), and test set (remaining time steps). Note that for model selection you will use the data in the training set, with a further split in training (first 4000 samples) and validation (last 1000 samples). Try to plot the time series data using the command `plot`.

- Create (implementing from scratch the equations) and train Echo State Networks using different values of the hyper-parameters (input scaling, number of reservoir units, spectral radius, readout regularization for ridge regression, etc.). As regards the hyper-parameters, you can either (manually) choose a set of values to consider, either (systematically) use a grid, or a random search method.
- For each hyper-parameterization that you consider, the performance (on training, validation and test sets) should be averaged over a number of *reservoir guesses*, i.e. different random instances of ESNs with the same values of the hyper-parameters (only the seed for random initialization changes among the guesses).
- Select the best network hyper-parameters <u>on the validation set</u>, the hyper-parameterization with the smallest Mean Squared Error (MSE), e.g. by using the command `immse`.
- Train the selected network on all the training data and evaluate the MSE of such network on the training set and on the test set.

Notes:

- You can use cell2mat for manipulating input and target data that has been loaded as cell arrays.
  E.g.
  load laser_dataset; %load the laser dataset
  allData = cell2mat(laserTargets); %converts the cell structured data into a row matrix.

- For next -step prediction tasks input and target data can be separated by an appropriate shift of indexing, e.g. `inputData = allData(1:end-1); targetData = allData(2:end);`
- To give a practical insight into the next-step prediction type of tasks, suppose that the input sequence is as follows:

time step	*1*	*2*	*3*	*4*	*5*	…
input value	3	5	9	12	8	…

  then: the desired output (i.e. the target) at step 1 is 5, at step 2 is 9, at step 3 is 12, and so on…

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

- The script .m file(s)
    - This file should also include a specification of the set of the hyper-parameters values considered for the model selection (if not already in the grid).
- All the Echo State Network structures corresponding only to the selected hyper-parametrization (e.g. a structure for the reservoir and a structure for the readout). Such structures must include all the weight matrices of the Echo State Network architecture, thereby including input-to-reservoir weight matrix `Win`, recurrent reservoir weight matrix `Wr` and reservoir-to-readout weight matrix `Wout`.
- All the hyper-parameters values of the selected Echo State Network, thereby including: number of reservoir units, input scaling parameter, spectral radius, readout regularization parameter (and the values of all the other hyper-parameters that you considered).
- Training, validation and test Mean Squared Error (e.g. by `immse` command).
- A plot (fig or png) with the target and output signal (use the hold on command to have a comparative plot). This type of plot is required for both training and test.

Bonus Track Assignment #2– Mackey-Glass Task

The Mackey–Glass (MG) time series is a standard benchmark for chaotic time series prediction models. The task of interest for this assignment is a next-step prediction task (autoregressive, a particular case of transduction) on the MG time series.

Import the dataset from the file MGtimeseries.mat (available on the Moodle platform in the “LAB3_1” folder), properly separate input and target data, then split the available data in training (first 5000 time steps), and test set (remaining time steps). Note that for model selection you will use the data in the training set, with a further split in training (first 4000 samples) and validation (last 1000 samples). Try to plot the time series data using the command `plot`.

- Create (implementing from scratch the equations) and train Echo State Networks using different values of the hyper-parameters (input scaling, number of reservoir units, spectral radius, readout regularization for ridge regression, percentage of connectivity among reservoir units, etc.). As regards the hyper-parameters, you can either (manually) choose a set of values to consider, either (systematically) use a grid, or a random search method.
- For each hyper-parameterization that you consider, the performance (on training, validation and test sets) should be averaged over a number of *reservoir guesses*, i.e. different random instances of ESNs with the same values of the hyper-parameters (only the seed for random initialization changes among the guesses).
- Select the best network hyper-parameters <u>on the validation set</u>, the hyper-parameterization with the smallest Mean Squared Error (MSE), e.g. by using the command `immse`.
- Train the selected network on all the training data and evaluate the MSE of such network on the training set and on the test set.

Notes:

- You can use cell2mat for manipulating input and target data that has been loaded as cell arrays.
  E.g.
  load laser_dataset; %load the laser dataset
  allData = cell2mat(laserTargets); %converts the cell structured data into a row matrix.

- For next -step prediction tasks input and target data can be separated by an appropriate shift of indexing, e.g. `inputData = allData(1:end-1); targetData = allData(2:end);`
- To give a practical insight into the next-step prediction type of tasks, suppose that the input sequence is as follows:

time step	*1*	*2*	*3*	*4*	*5*	…
input value	3	5	9	12	8	…

  then: the desired output (i.e. the target) at step 1 is 5, at step 2 is 9, at step 3 is 12, and so on…

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

- The script .m file(s)
    - This file should also include a specification of the set of the hyper-parameters values considered for the model selection (if not already in the grid).
- All the Echo State Network structures corresponding only to the selected hyper-parametrization (e.g. a structure for the reservoir and a structure for the readout). Such structures must include all the weight matrices of the Echo State Network architecture, thereby including input-to-reservoir weight matrix `Win`, recurrent reservoir weight matrix `Wr` and reservoir-to-readout weight matrix `Wout`.
- All the hyper-parameters values of the selected Echo State Network, thereby including: number of reservoir units, input scaling parameter, spectral radius, readout regularization parameter (and the values of all the other hyper-parameters that you considered).
- Training, validation and test Mean Squared Error (`immse` command).
- A plot (fig or png) with the target and output signal (use the hold on command to have a comparative plot). This type of plot is required for both training and test.

Bonus Track Assignment #3 – Leaky Integrator ESN

Apply Leaky Integrator Echo State Networks with different values of the leaking rate parameter a to the previous tasks in Assignments #1 and Bonus Track Assignment #1,2.