ALSO can be used [it you can.

7 sensors of electricity will use only one.

EEL 6504
Due April 25, 2023

Project 2 -

Check if it linear or Non-linear system

Prepare a 7 pages report in scientific paper format, using the template from IEEE Trans. Signal Processing. The goal is to compare implementations of the LMS, KLMS and QKLMS trained with MSE and MCC to predict the next sample of the ETTm2 time series, which is a real-world data and has been used for the study of long-sequence time series forecasting. It contains two years data from an electricity transformer station in China. Each data point is recorded every hour on 7 features, HUL (**H**igh **U**se**F**ul **L**oad), HULL (**H**igh **U**se**L**ess **L**oad), MUFL (**M**iddle **U**se**F**ul **L**oad), MULL (**M**iddle **U**se**L**ess **L**oad), LUFL (**L**ow **U**se**F**ul **L**oad), LULL (**L**ow **U**se**L**ess **L**oad) and OT (**O**il **T**emperature). In the project, start using just the single OT channel. There are ~17k data points and you should use the first 11k of the data as your training (10k)/crossvalidation (1k)set and the last 6k as your testing set. Normalize each channel to zero mean, norm one. [Dataset all the large values]

Conventional prediction. As you recall, we delay the input to the model by one sample and the current sample will be used for the desired response. Use an input layer with 7 delays to map the data to RKHS (i.e., an input embedding vector of size 7). As a baseline for the comparison, please implement the linear filter with LMS filter trained with the same two cost functions.

Linear will be your baseline

- Provide an analysis of how to estimate the three free model parameters (kernel size, quantization, learning rate) using the prediction error for KLMS and QKLMS trained in the training/crossvaliation sets. Please show weight tracks.
- Quantify the convergence, and the final prediction error in the test data set. [where you should stop or start training]
- For the KLMS and QKLMS, show the tradeoff accuracy versus number of samples in two figures as demonstrated in class.
- Please create the histogram of the errors in the test set for the two cost functions to compare the MSE and MCC training. Remember to select appropriately the kernel size in MCC (the 4th hyper-parameter) in the crossvalidation set and show its effect in a figure. Explain the difference from first principles (i.e. the theory of MCC).
- Test the performance of the conventional predictor when trained to predict 1 sample ahead, in predicting 1, 10 and 50 samples ahead. Present the results in a table. Explain the results from first principles.
- Generate the trajectory from a trained model to estimate how many samples the trained model is able to predict up to a lower limit of the error (see explanation below).
- For the single channel input, redo the training to predict up to 10 samples ahead, and compare predictions at 1, 10 and 50 samples. There are multiple ways, and I suggest two for your consideration: (1) create multiple outputs (one for each advance in time), or change the delay operator (or equivalently the RKHS embedding) from z^{-1} to z^{-10}. Compare them with the previous approach on a table and explain from first principles (the theory) the results you obtain.
- Extra credit: Learning the trajectory. Recursive prediction trains the model in the same way as it is going to be used for the trajectory generation. This method requires pre training of the parameters in the first 5,000 samples as trained above for sample prediction and the error is still created between the system output and the next sample. But then, keep training the filters in the training set with the delayed outputs instead of with the input samples until the error stabilizes at small values. Expect training to be slower than before and reuse the training dataset if needed (starting over from sample 1). In a table show the number of samples (horizon of predictability) that each of the different trained systems can predict,

Verify if performance same for MSE and MCC.

* gain in computation and decrease in performance

* histogram of errors on training set for two cost function can be done on test set

MCC kernel size good accuracy and decreases outliers.

[Best comparison is histogram]

Show weight tracks on LMS.

with an instantaneous normalized error s.d. < 0.2. Explain the reason for the results obtained.

Note. To get the data use the API pandas.read_csv(filename). To normalize the data use StandScaler from sklearn.preprocessing.

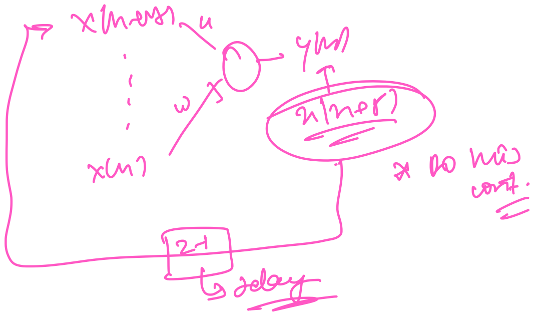
Generating the trajectory. The final test that you will be doing is to **generate** with the different trained models (LMS, KLMS, QKLMS) the OT time series. With the parameters fixed, create a recurrent system i.e., use the trained model and feedback its output to the model input with a delay of one sample. The initialization of this recurrent model must be done carefully to align the inputs and the delayed output. For the 7 delay embedding, take 8 samples of the time series and place them in a first in-last out buffer (FILO) to create the first input embedding vector to the model; compute the model output and append it to the top of the FILO to create the next input to the model; keep iterating this procedure by feeding predicted samples instead of inputs. Since you know the next samples of the time series that follow the 8 samples, you can compute the error between the true next sample and the model output. When the instantaneous deviation from the true OT data and the filter output reaches 1/5 of the standard deviation of the OT series stop the prediction and count how many samples the model takes to meet the error threshold. Use 20 different initial conditions and average the prediction steps for reporting.

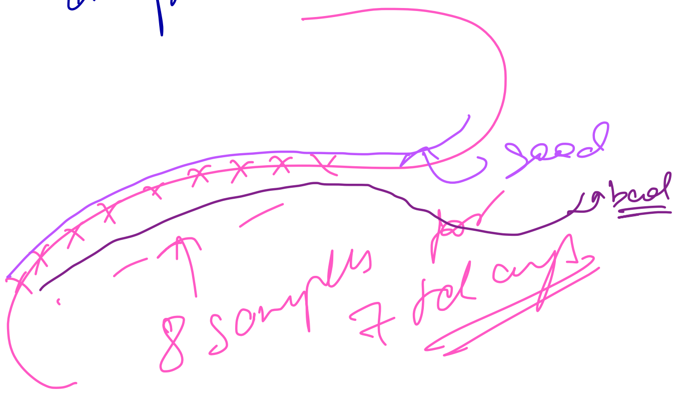
* Find Best possible predictor ;
* pushing the system 10 samples, 50 samples ahead.

jump,

what expectation

* Generating the trajectory given an initial condition, how the prediction evolves.
start from a predictor evolves.





way of selecting / Implementing:

10 samples ahead

Lo mean trajectory

for 10 samples

and find diff b/w last sample

To want 50 samples ahead

* initial condition on the trajectory.

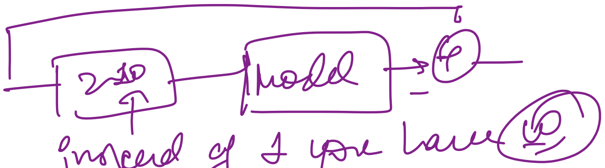
* train models

10 sample ahead

2 possibilities.

instead of one you can have 10 O/P.

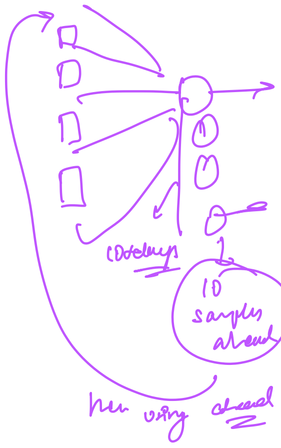
you can use sample ahead as input to your models

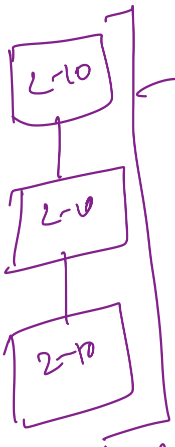


instead of 1 you have 10

so you predict 10 samples

from 8 weights.





This will look into further values taking values at diff sampling rates.

* After implementation then result.

Extra credit →

Making system aware of the error and using that to change weights :

① part
true time series (5000)

lots conveyed

② part

train with the error and cost

* signal generator learns trajectory.