Slide set 3 exercises

NOTE : you can merge the subprojects analyses into the same document, zip can contain all the exercise codes, there is no need to do separate zip packages, just one.

NOTE2: The classifier/predictor codes are expected to have training and testing phases and thus the data must be divided (close to) 80% / 20% ratio for training and testing.

Exercise 1

Look at the page:

https://timingai.com/finance/using-fourier-transforms-with-pytorch-for-data-analysis/

and browse the slide set 3a.

Explain verbally how the Fourier transform can be used for time series data manipulation in machine learning and especially with neural networks.

Explain verbally why it is not worth training and computing the frequency conversion (Fourier transform) using NN's.

Max 2 p.

Exercise 2

Install noise reduction pack:

https://github.com/timsainb/noisereduce

and download **two** speech samples from:

https://github.com/microsoft/MS-SNSD/tree/master/noise_train

run the noise reduction to these two speech samples. Show spectrum clips 300-500 samples from the original and noise reduced outputs from both speech samples (four images total). Copy the spectrogram parts from "spectrogram_sound.py" and modify it to your needs.

You may have to adjust the parameters to make the system work.

Include these four figures (aligned in pairs) in your analysis answer.

Max 3 p.

(This is not working properly)

~~Exercise 3~~

~~Load the "simple_autoencoder.py" and the dataset "CS2_35.csv" from code folder. Verify that the code is running smoothly.~~

~~Verbally (examine the original code):~~

- ~~How the autoencoder is working, what are the input and output (target) time series?~~
- ~~Which one the outcome of autoencoder is compared to in MSE process?~~

~~Modify the code to train/clean your own time series (sound, noisy time series, stock exchange, etc.). Do not use the dataset included in the original code.~~

~~You might look from this page:~~

~~https://machinelearningmastery.com/time-series-datasets-for-machine-learning/~~

~~If you cannot find noisy time series, you can add noise to your own time series with the method presented in code.~~

~~If the cleaning process is not working you may alter the architecture from input_size-32-16--16-32 input size to much larger and consider changing the values of "input_size", "hidden_size" and "batch_size", by verifying the results VISUALLY. Try to match the original data to reconstruted data as close as possible.~~

~~Include triplet figure containing original, noisy and cleaned time series in your analysis containing 200-500 aligned samples in each subfigure.~~

~~Max 5 p.~~

EXTRA: Exercise 4

Download the time series classification database 'pip install aeon'
https://www.timeseriesclassification.com/dataset.php

Choose one of the sets from the list and modify 'Exercise_train_test.py' code to classify the selected dataset (train/test). Include confusion matrix in code.

Test the effect of **standardization**, by copying/mimicking the code from 'Normalization_snippet.py' with and without the standardization. Remember to make the normalization to both train and test data.

How much the accuracy result changed in test data between no-normalization and normalization?
Show the results also using confusion matrices. Can you see the effect in matrices?

Explain verbally: Why the X is usually normalized but not y in prediction?

Include figures to your detailed analysis answer as well as zipped code.

Max 11 p.