Time Series Augmentation Example

Last updated on 09_18_2024 by Yiming Jia

Notes:

1. This **Example** folder includes the codes and data to reproduce the convolutional variable autoencoder (CVAE) in Section 3.1 of Jia and Sasani (2025).

2. The result of Step 1 is included, which is marked as **finished**.

3. Some data are larger than 25 MB (e.g., “Aug_TimeSeries.mat” and “Aug.dat” in subfolder “Data_Lat4_Epoch50000_Aug”, the authors suggest to download them individually and move them to the corresponding folders or subfolders.

Pre-requests:

1. Users need to have access to Python 3.10.14 (TensorFlow 2.13.1 required) and Matlab R2023b.

The authors recommend running Python codes on Jupyter Notebook. Running the codes on different versions of Python, TensorFlow, or Matlab may result in compatibility issues. However, most compatibility issues can be resolved by following the suggestions in the error messages.

2. Users need to have basic knowledge of machine learning or deep learning to tune the hyperparameters of convolutional variable autoencoder (CVAE).

Step-by-step instruction:

Step 1 (finished): Prepare the time series data for augmentation as a MATLAB dataset.

The time series data are saved as “TimeSeries.mat”, which includes a 94×1 vector of cells. Each cell includes a 800×2 matrix.

Step 2: Run Python code “P1_CVAE.ipynb”.

This step is used to train the CVAE, which may take a couple of hours.

Three folders (Model_Lat4_Epoch50000, Figure_Lat4_Epoch50000, and Data_Lat4_Epoch50000) are automatically generated when “P1_CVAE.ipynb” is finished.

Step 3: Run Matlab codes “P2_Plot_Comp.m” and/or “P2_Plot_Comp_2D.m”.

These two Matlab codes are used to generate the figures to compare the original and reconstructed time series.

In “P2_Plot_Comp.m”, for each dimension of one original time series, the original and reconstructed time series are plotted versus time steps. The user can change the parameter “ID” on line 34 to plot the time series of interest or write "ID" as a vector to plot comparisons for multiple time series.

In “P2_Plot_Comp_2D.m”, the plot is generated using two dimensions of the time series (2D). The user can select which two dimensions to plot on line 33. Similarly, the parameter “ID” on line 37 can be changed, as previously mentioned.

Step 4: Run Matlab code “P3_Aug_Lat.m”.

This Matlab code is used to generate latent features based on the mean and variance learned by the trained CVAE. The “Num_Aug” is a user-defined parameter (here as 100), which is the number of augmented time series generated from a single original time series.

Step 5: Run Python code “P4_CVAE_Aug.ipynb”.

This Python code loads the trained CVAE in step 1 to augment the time series. The folder named “Data_Lat4_Epoch50000_Aug” is automatically generated when “P4_CVAE_Aug.ipynb” is finished. It includes the augmented time series as “Aug.dat”.

Note that “Aug.dat” has been uploaded by the authors. If the users does not want to run the previous steps, they can use the uploaded “Aug.dat” for the following steps.

Step 6: Run Matlab code “P5_Aug_Data.m”.

This Matlab code is used to post-process the time series data augmented from Step 5. The augmented time series data are saved as Matlab dataset “Aug_TimeSeries.mat”, which has the same structure as the original time series data “TimeSeries.mat”.

Note that “Aug_TimeSeries.mat” has been uploaded by the authors. If the users does not want to run the previous steps, they can use the uploaded “Aug_TimeSeries.mat” for Step 7.

Step 7: Run Matlab code “P6_Plot_Aug.m” and/or “P6_Plot_Aug_2D.m”.

These two Matlab codes are used to generate the figures to compare the original and augmented time series.

Similar to Step 3, “P6_Plot_Aug.m” plots the original and augmented time series versus time steps. The user can change the parameter “ID” on line 22 to plot the time series of interest or write "ID" as a vector to plot comparisons for multiple time series.

In “P6_Plot_Aug_2D.m”, the plot is generated using two dimensions of the time series (2D). The user can select which two dimensions to plot on line 21. Similarly, the parameter “ID” on line 26 can be changed, as previously mentioned.