Dynamical Systems Theory in Machine Learning & Data Science

lecturers: Daniel Durstewitz, Zahra Monfared
tutors: Manuel Brenner, Daniel Kramer, Janik Fechtelpeter, Max Thurm, Jonas Mikhaeil, Unai Fischer Abaigar
WS2021/22

Final Project

To be uploaded before or on March 9th, 2022

For organizational purposes, please include your matriculation numbers when handing in the project. Please tell us if you need a grade, and if you handed in the exercise sheets as a group please check if you received a score on all the sheets. Lastly, also let us know if you need your grade for the lecture before the 9th of March.

The aim of the final project is to make use of the concept and ideas you have learnt in the last couple of months in a practical context by training a model on a time series, using it to generate a completely new time series, and gauging the quality of the reconstruction by a power spectrum correlation measure.

As we have seen during the lecture, there are many different approaches that deal with dynamical systems reconstruction. During the final project, you are asked to select a method from the literature where code is in some form provided by the authors, and use it to reconstruct two time series derived from benchmark dynamical systems. To make things a little easier, we have compiled a list of suggestions. In case you have problems with the selection or are stuck with one of the approaches, feel free to reach out to us.

- Neural ODEs (e.g.Chen et al.. For this approach, you can also feel free to implement a Neural ODE from scratch, using e.g. the pytorch package torchdiffeq package. In that case, you can summarize the Neural ODE paper from Chen et al..
- RNN-ODEs (Rubanova et al.) This approach generalizes RNNs by combining them with NODEs.
- LSTMs (Vlachas et al.).
- Reservoir Computing (Pathak et al.). Both the LSTM and the Reservoir Computing approach are covered in the same code base (a fair warning: the code is a little messy to wrap your head around initially).
- Next Generation Reservoir Computing (Gauthier et al.). We've had mixed success with this model but you are free to try it and point out potential downsides.

Task 1. Write a short summary

Familiarize yourself with the content and theoretical background of the paper you selected and write up a short summary (1-2 pages) of its content in Latex.

Task 2. Implement a power spectrum metric

On moodle, we provide a code snippet called *psc.py*. This computes a power-spectrum correlation (PSC) between two input time series. It features two hyperparameters: σ, which smoothes the spectra with a Gaussian kernel with width σ, and the cutoff, which excludes all frequencies above the threshhold frequency from the calculation of the PSC. Integrate this metric into your code by implementing a routine which calls the model, draws a random initial condition and generates a time series of length T. Then use this freely generated time series to compare the power spectra between the ground truth time series and the generated time series.

Task 3. Testing

On moodle, we provide you with two datasets. One is generated from the Lorenz-63 system that you have already encountered frequently in the tutorials, and the other from the Lorenz-96 model.Both datasets are split into 100 time series of length 1000.
Train the model on the training data. If reconstructions are not successful, play around with the respective hyperparameters of the algorithm (depending on the approach, it is of course not guaranteed that reconstructions will be successful in the end, but try not to give up too soon). Test the quality of reconstruction on an appropriate test set by computing the power spectrum correlation between ground truth and reconstructed time series. Find values for the power spectrum smoothing factor and the cutoff which make sense for the given dataset.

Good luck!