Topological Machine Learning Library

The goal of this project is to apply all material learned in INF367 and successfully implement and apply topological machine learning methods. Please read the description carefully!

This project is a compulsory part of the course. This project contributes 75% to the final grade. The grade will be based on good choice of methods, correctness of answers, clarity of code and thoroughness and clarity of reporting.

Requirements

The work will consist of three parts:

- Topological Machine Learning Library (50 pts): This should be a library that can be installed using `pip install`. The library should be modeled on `sklearn` and contain classes for
  - graph finding (Self-Organizing Maps, Growing Neural Gas, Generative Gaussian Graphs),
  - Reeb graphs (Mapper),
  - persistent homology (filtered complexes, matrix reduction, persistence images, persistence landscapes, PersLay),
  - topological neural networks (topological autoencoder, topological layer).

  Typical classes should have a `fit` and `transform` or `predict` method. The neural network classes should instead be possible to use in place of another standard class that it can replace (e.g. a layer from `tensorflow.keras.layers`).
- Documentation (30 pts): All classes should be documented, with all parameters clearly described. For each class, there should be at least on example in the documentation.
- Application (20 pts): The purpose of the application is to create one good use-case of topological machine learning. You will use publicly available (or your own) real data. Simulated data will not be accepted for this part. The submission should be a Jupyter Notebook with the following structure:
  - Introduction, i.e. where the problem is explained.
  - Dataset, i.e. where the dataset is described.
  - Analysis, i.e. where the steps of the analysis are highlighted and the central role played by topological machine learning is explained.
  - Short benchmark with comparable non-topological methods. You can get full points even if the classical method outperforms the topological method.

Deadline

The project has to be submitted before Friday, 22.04, 23.59. Deliver at MittUIB.no/assignments. Late assignments will be penalized.

Deliverables

For the final submission, please provide one zip file that contains all relevant parts of the project including a README.md file that gives an overview of all files included.

In addition to packages from the standard library, you may use the following python packages: `numpy`, `pandas`, `scipy`, `sklearn`, `matplotlib`, `seaborn`, `plotly`, `tensorflow`, `keras`, `torch`. If you use any other packages we will not be able to run your app and you will fail the project (unless discussed with Nello ahead of time).

Code should be documented and tricks (e.g. to avoid division by zero, to make sure it takes finite time to run, etc.) should be reported. The rational behind all steps in the code should be clear from the report.

NOTE: This project is a learning experience. If we see that you have copied your answers from online resources, you will get 0 points.