Project 2

FYS5429/9429, Advanced machine learning and data analysis for the physical sciences, University of Oslo, Norway

Spring semester 2024, deadline June 7

Possible paths for project 2

We discuss here several paths as well as data sets for the second project (or as parts of a larger project) Tentative deadline June 7.

The report should also be styled as a scientific report. The guidelines we have established at https://github.com/CompPhysics/AdvancedMachineLearning/tree/main/doc/Projects/EvaluationGrading could be useful in structuring your report. We have also added a lecture set by Anne Ruimy (director of EDP journals) on how to write effective titles and abstracts. See https://github.com/CompPhysics/AdvancedMachineLearning/tree/main/doc/Projects/WritingAbstracts for these lectures. Finally, at https://github.com/CompPhysics/AdvancedMachineLearning/tree/main/doc/Projects/2023/ProjectExamples you can find different examples of previous reports. See also the literature suggestions below.

For those of you who have planned to write one project only, feel free to proceed with that.

For those of you who plan to write a second project, we would like to propose that you focus on generative methods, in particular those we have discussed during the lectures. These are

1. Boltzmann machines
2. Variational autoencoders and GANs
3. Diffusion models

Here you can opt for the following paths.

1. The computational path: Here we propose a path where you develop your own code for say a Boltzmann machine or a variational autoencoder

and apply this to data of your own selection. The code should be object oriented and flexible allowing for eventual extensions by including different Loss/Cost functions and other functionalities. Feel free to select data sets from those suggested below here. You can compare your own codes with implementations using TensorFlow(Keras)/PyTorch or other libraries. The codes in the examples from the various lectures use MNIST as example dataset. Feel free to use the same dataset or other ones.

1. The differential equation path: Here you can use the same differential equations as discussed in project 1, but now solving these with either Boltzmann machines or Variational Autoencoders (see for example https://arxiv.org/abs/2203.11363), or GANs (see for example https://proceedings.neurips.cc/paper/2020/file/3c8f9a173f749710d6377d3150cf90da-Paper.pdf).

2. The application path: Here you can use the most relevant method(s) (say variational autoencoders, GANs or diffusion models) and apply this(these) to datasets relevant for your own research.

3. Variational Autoencoders (VAEs) and Bayesian statistics path: Here you can compare VAEs with Bayesian Neural Networks, see https://github.com/ibarrond/VariationalAutoencoders. For this project we recommend the pedagogical article by Kingma and Welling, An Introduction to Variational Autoencoders, see https://arxiv.org/abs/1906.02691 as background literature and the literature list below here.

Defining the data sets to analyze yourself

You can propose own data sets that relate to your research interests or just use existing data sets from say

1. Kaggle

2. The University of California at Irvine (UCI) with its machine learning repository.

3. For the differential equation problems, you can generate your own datasets, as described below.

4. If possible, you should link the data sets with existing research and analyses thereof. Scientific articles which have used Machine Learning algorithms to analyze the data are highly welcome. Perhaps you can improve previous analyses and even publish a new article?

Literature

The following articles and books (with codes) are relevant here:

1. Kingma and Welling, An Introduction to Variational Autoencoders, see https://arxiv.org/abs/1906.02691.
2. To create Boltzmann machine using Keras, see Babcock and Bali chapter 4, see https://github.com/PacktPublishing/Hands-On-Generative-AI-with-Python-and-TensorFlo blob/master/Chapter_4/models/rbm.py
3. See also Foster, chapter 7 on energy-based models at https://github.com/davidADSP/Generative_Deep_Learning_2nd_Edition/tree/main/notebooks/07_ebm/01_ebm and chapter 3 for VAEs and chapter 8 for diffusion models.
4. Du and Mordatch, Implicit generation and modeling with energy-based models, see https://arxiv.org/pdf/1903.08689.pdf
5. Calvin Luo gives an excellent link between VAEs and diffusion models, see https://calvinyluo.com/2022/08/26/diffusion-tutorial.html

Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.
- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.
- Include the source code of your program. Comment your program properly.
- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.
- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.
- Try to evaluate the reliabilty and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.
- Try to give an interpretation of you results in your answers to the problems.

- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to keep this course at the interactive level and your comments can help us improve it.
- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.

Format for electronic delivery of report and programs

The preferred format for the report is a PDF file. You can also use DOC or postscript formats or as an ipython notebook file. As programming language we prefer that you choose between C/C++, Fortran2008 or Python. The following prescription should be followed when preparing the report:

- Send us an email in order to hand in your projects with a link to your GitHub/Gitlab repository.
- In your GitHub/GitLab or similar repository, please include a folder which contains selected results. These can be in the form of output from your code for a selected set of runs and input parameters.

Finally, we encourage you to collaborate. Optimal working groups consist of 2-3 students. You can then hand in a common report.