Advanced Machine Learning Course Project

Deadlines.

End of day Friday, November 17th, 2023: Topic and group proposal must be emailed to ~~austin.stromme@ensae.fr~~ advanced.ml.fall.2023.ensae@gmail.com.

End of day Friday, January 19th, 2023: Written report due – instructions for submission will follow.

Project description.

This project is intended to deepen your knowledge of machine learning by asking you to dive into an aspect of machine learning which is of interest to you. The spectrum of allowable topics is broad. On the very applied side, you may identify a problem you want to solve with machine learning, and do a project centered around finding/developing the best algorithm and model for this problem. And on the very theoretical side, you may find an interesting paper about the theory of ML and study that. The only constraint is that **there must be some implementation of ML models/algorithms with real data**. And the more applied the project, the more involved the implementation needs to be. You will write a report about your project and submit it along with your code. **To be done in teams of at most 3 people (strongly encouraged to form teams of at least two).**

Written report.

The main deliverable is a written report due by **end of day Friday, January 19th, 2023**. In general, it should be written similarly to a typical ML research paper, but with the following differences:

- **Be honest.** If your method only works under special conditions or seems to be worse than other methods in every setting you've tried, make that clear. This is fine as long as you have thoroughly explored *why* it isn't working and given evidence in your writeup.
- **Keep it simple.** Please don't try and describe the most complex and involved version of your method. Keep it simple and focus on the simplest version of the method that you can think of that still keeps its important features. Use the space you've saved to
- **Teach the reader.** Write the report in a way that one of your classmates could literally pick it up, read it in half an hour, and learn several interesting things. Discuss background literature, actual settings where the topic arises (e.g. if you do a report on clustering, talk about an interesting applied problem where clustering shows up), and other methods for the problem.

It should be between 6 and 9 pages (inclusive), not counting reference, and in the NeurIPS style format (available for download here). It should include the following components:

- **Introduction.** Include thorough background discussion of the topic, where it arises, why it is important, and what previous methods do and don't accomplish.
- **Your method/topic.** Include a complete description of the specific topic under consideration in this report, a discussion of its motivation, meaning, benefits and limitations. This discussion and the empirical section should complement each other (e.g. don't claim that your method for clustering works great but then not have any empirical evidence to confirm it).
- **Empirical analysis.** Must apply the method to real data and thoroughly analyze its performance. This might include a preliminary analysis of the method on synthetic data to evaluate its correctness (e.g. does your method for clustering recover the true clusters on a synthetic dataset?), and then a subsequent analysis on real data, or could be only on real data. There should be multiple plots/tables/other figures, and it shouldn't be the same quantities each time (e.g. don't plot the training error vs. number of iterations three different times). Data should include error bars, if appropriate. Hyperparameters should be selected with cross-validation if possible.
- **Code.** At a minimum, you must personally implement at least some part of your ML model – you may not do a project that simply applies out-of-the-box ML models, or ML models copied from github, to an out-of-the-box dataset, and then makes new figures. Such a minimum standard would apply only to the most theoretical/research-level projects; the more applied the project the more involved the implementation is expected to be. Thus, a very applied project is expected to have *much more sophisticated implementation* than this minimum standard – potentially including data processing, training of complex models from scratch, and custom refinements. As explained above, the intent of the project is not to force you to re-create the minute details of existing work, but to instead encounter, learn, and implement the core *ideas* of existing work. An important part of deeply understanding a topic is knowing which parts are essential and which are not, as well as finding creative ways to deeply explore the topic with the time and computational resources you have available. In terms of deliverables for the code:
  - **There must be a link to a github repo or other location where your code and data can be downloaded**.
  - The code must be in Python and include everything you used to both collect data and to generate plots.
  - There must be a readme with detailed instructions on how to recreate your results.
  - There must be a detailed detailed description of how you created the code, clearly specifying to what extent the code is original, where code was copied from if it was copied, and what modifications you made and why.

Topic and group proposal.

You must email a brief description of your proposed project and the proposed group members to austin.stromme@ensae.fr by **end of day Friday, November 17th, 2023.** This description should include: a description of the general topic, what you plan to do for your own project, at least three references (can be books, papers, blog posts, github repos, etc.) for the topic, the dataset you will use, and a brief description of how you plan to implement your code. I will give you feedback if your project is out of scope and you can then revise your proposal and re-submit to me.

One approach to this project.

1. Find and carefully read an ML paper that introduces an important method or idea. (This step can be based on your own readings, the following list of papers, or by looking through an ML textbook and finding an interesting topic and then searching for the important papers in that area.)
2. Identify and explore a dataset to apply the method to, perhaps from the original paper itself.
3. Carefully read several prior and follow-up works to understand its context in the literature, important applications of the method, and relevant baselines.
4. Implement a minimal version of the method as well as some baselines.
5. Do preliminary empirical exploration on synthetic data.
6. Revise your implementation.
7. Do a final empirical exploration of the method on synthetic and real data.
8. Write your report.

Here are some interesting ML papers that could inspire a project.

Important ML algorithms:

- Support Vector Clustering, by Ben-Hur, Horn, Siegelmann, and Vapnik, 2001
- Visualizing data using t-SNE, by van der Maaten and Hinton, 2008
- XGBoost: A Scalable Tree Boosting System by Chen and Guestrin, 2016
- Adam: A method for stochastic optimization, by Kingma and Lei Ba, 2014
- Sinkhorn: Lightspeed Computation of Optimal Transport Distances, by Cuturi, 2013
- Convolution Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains, by Solomon et al, 2015
- A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, by Freund and Schapire, 1997
- Introduction to multi-armed bandits by Slivkins, 2017
- Efficient Estimation of Word Representations in Vector Space, by Mikolov, Chen, Corrado, and Dean, 2013
- A Survey of Distributed Optimization by Yang et al 2019

Interesting theoretical phenomena in machine learning:

- Gradient descent using neural networks typically occurs at the edge of stability, by Cohen et al, 2021
- Reconciling modern machine learning practice with the bias-variance tradeoff, by Belkin, Hsu, Ma, and Mandal, 2019

- Do Imagenet Classifiers Generalize to Imagenet? by Recht, Roelofs, Schmidt, and Shankar, 2019
- A meta-analysis of overfitting in machine learning, by Roelofs et al, 2019
- Neural Tangent Kernel: Convergence and Generalization in Neural Networks, by Jacot, Gabriel, and Hongler, 2018
- Understanding deep learning (still) requires re-thinking generalization, by Zhang et al, 2019
- Implicit Regularization in Matrix Factorization, by Gunasekar et al, 2017
- Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation, by Belkin 2021

Adversarial robustness – this is a huge topic, could start with these:

- Intriguing properties of neural networks, Szegedy et al, 2013
- Towards Deep Learning Models Resistant to Adversarial Attacks, by Madry et al, 2017
- Adversarial Examples Are Not Bugs, They Are Features by Ilyas et al 2019

Fairness and privacy – also a huge topic, could start with these:

- Inherent Trade-Offs in the Fair Determination of Risk Scores by Kleinberg, Mullainathan, and Raghavan, 2016
- A Survey on Bias and Fairness in Machine Learning, by Mehrabi et al 2021
- Membership Inference Attacks Against Machine Learning Models, by Shokri, Stronati, Song and Shmatikov, 2017

Deep learning:

- How Does Batch Normalization Help Optimization, by Santurkar, Tsipras, Ilyas, and Madry 2018
- On the difficulty of training Recurrent Neural Networks, by Pascanu, Mikolov, and Bengio, 2012
- Attention is all you need, by Vaswani et al, 2017
- BERT: Pre-training of deep bidirectional transformers for language understanding by Devlin, Chang, Lee, and Toutanova 2018
- Improving natural language understanding by generative pre-training by Radford, Narasimhan, Salimans, and Sutskever, 2018
- Language models are unsupervised multi-task learners, by Radford et al 2019
- Language models are few-shot learners, by Brown et al 2020

Resources for datasets.

- Kaggle
- UCI ML repository

Example projects from Stanford's Machine Learning Course.

Fall 2019
Spring 2021