Deep Learning Project

Robert Mateescu

1 Introduction

In Deep Learning we design systems that work on data, and their behavior and performance depend on the implementation details and on the data itself. The theoretical and conceptual ideas that we discuss in class are the tools and background that will help you when you actually implement these algorithms. The project is an opportunity to test your understanding and further explore on your own. Use your curiosity and creativity to think about the problems, and propose solutions and ideas. I encourage you to develop a habit of asking questions, to yourselves and to your peers. Don't be afraid to come up with questions, there are no stupid questions, and in time you will sharpen your intuition about what and how to ask, and how to figure out the essential points.

2 Teams

You will work in teams. We will randomly decide the teams in class. The grading will be done as described in the syllabus. There are maximum 15 points: up to 12 points for the collective team effort and up to 3 points for individual effort. You are like a sports team now: work hard together and collaborate, share your knowledge and help each other. Use your organizational skills to decide how everyone will contribute. Even if you split the work without overlap, each one of you should still understand what your team members did, and be able to present the entire work.

3 What Will You Do?

1. **Choose a dataset and a task** (e.g., classification, prediction).
2. **Implement a deep learning system, train, validate and test it.** This is the part that will take most of the time. You may explore various neural network architectures (e.g., MLP, CNN, LSTM, RNN). You will do a hyper-parameter search, which includes but is not limited to: network size (number of layers, number of nodes per layer), activation functions, learning rate, batch size, number of epochs, optimizer, loss function, dropout rate, regularization method, random restarts, data split etc. I encourage you to use PyTorch, but you are welcome to use any other framework.
3. **Write a report describing what you did.** The report is a mini research paper. It contains title, author names, abstract, and several sections that cover: introduction to the problem, what you set out to achieve, methodology, experimental setup, explanation of results, conclusion that summarizes what we learn from your work. Include useful plots of the data you collected during training (e.g. accuracy, loss, convergence time, memory usage etc)., images, diagrams or other visuals that facilitate your exposition. Do not include too much code, unless it is really necessary.
4. **Give a presentation in class** (e.g., 15-20 minutes for presentation + 5 minutes for questions or discussion). During the presentation, each team member should present at least one slide. I suggest you prepare 10-15 slides.
5. **Have some fun while doing the project.**

4 Datasets

There are many publicly available datasets, some of which are mentioned below. You are encouraged to search for and pick other datasets that you are interested in.

Some of the more popular datasets are MNIST, CIFAR-10, Iris, Wine, ImageNet. Try something with thousands of examples, the more the better so that your algorithm can learn something meaningful. The large ImageNet has 14M examples, and that is too big. You can use the one from ILSVRC (ImageNet Large Scale Visual Recognition Challenge), that has 1,281,167 training images, 50,000 validation images, and 100,000 test images. You may decide to use only a part of that dataset, for example just a few classes (e.g., cats and dogs).

You are encouraged to explore the repositories below, as well as many others.

- UC Irvine Machine Learning Repository:
  https://archive.ics.uci.edu/
- The Kaggle competition:
  https://www.kaggle.com/datasets
- Google dataset search:
  https://datasetsearch.research.google.com/