Courseguide Machine Learning

Grouls, Raoul
raoul.grouls@businessdecision.nl

November 17, 2022

1 Common Information

Period	Nov 22 - Jan 23
Subject	Masterclass Machine Learning
Credits	5 ECTS
Github	raoulg/ML22
Language	English and Dutch
Prerequisite	Python programming and visualisation, statistics
Module leader	Raoul Grouls
Phone	06-26710051
Email	raoul.grouls@businessdecision.nl

Lectures (7 times 3 hr)	21
Lab (7 times 3 hr)	21
Private study (minimum)	56

2 Context

When you have learned to automate things with python, and you are familiar with data cleaning (pandas/polars), data visualisation (matplotlib/seaborn) and you are comfortable with statistics (mean, median, boxplots, histograms, probability in general, continuous / discrete distributions, normal distributions, simpsons paradox) it is quite natural to start creating models. All these items have been covered in the Data Mining & Exploration (DME) course.

The field of machine learning is really too big to cover everything. We have:

I "classic" machine learning from the 80s-00s (linear regression, random forrests, Support Vector Machines)

II probabilistic programming (PyMC, Turing.jl)

III deep learning: Neural networks and their variations (CNNs, RNNs, Transformers)

The theory of the classic machine learning algorithms is covered in 'Datascience for Business". This are the models you can store on a floppy disk, and run on a Commodore 64 (or, if you insist, excel). Probabilistic Programming is interesting if you need explainable models. It is definitely usefull, but this course will focus on deep learning. We will use the Pytorch framework, which is very usefull for learning the concepts and frequently used in a business context.

3 Environment

3.1 Library management

Reproducing results is really important. Especially on other machines than your own (unless you are planning to hand over you laptop to every client). We will be using poetry, which has been used in DME course. If you are not familiar, read the documents!

3.2 Code style guidelines

We will be following professional code style guidelines. A lot will be familiar from DME, and most of the code in this course follows these guidelines, so you can get familiar by paying close attention to what is happening. However, make sure to read through it. If you follow these guidelines, everyone that needs to read your code at a later time will thank you, including yourself 6 months from now...

3.3 Git

Because emailing code is not an option (yes I know you *can*, but that doesnt mean it is a good idea). we will need git, which is industry standard. Again, you have seen this in DME. Please try to practice git, I know it doesn't has the best user interface, but you will need to get used to it. If you really struggle, try one of the graphic interfaces or plugins.

3.4 Linting

We will use linters. This is just an additional tool to get clean code. Especially black

3.5 Gin and Ray

We will need to do a lot of experimentation. It is really to only way to develop an intuition about what sane settings are for a model. gin-config will help us not to loose track of everything we are doing, by storing configuration in a small

textfile. In addition to that, we will use ray tune to automate searching. Ray offers algorithms that help us searching through settings-spaces that are too vast to search manually.

4 Lessons

We will use the book 'Programming PyTorch for Deep Learning'. While it will help us with a lot of lessons, but not all lessons will be covered by the book.

I added a 0-baseline lesson. It has excercises and answers. I created these lessons as additional stuff during the years, but it is all usefull.

Lessons 1-3 will be covered by the book, lesson 4 will be fully focused on working with ray.

Since the 2017 paper 'Attention is all you need', the transformers architecture is the state-of-the-art. Even though the book doesn't cover it, we will take a full lesson to look at the architecture.

I will also spend some time on unsupervised problems, which is very usefull if you don't have a labeled dataset available.

And finally, while pytorch is pretty fast, due to it's c++ backend, it can be problematic to use python. During the DME lessons, we have used polars as a fast alternative for pandas. For the machine learning course, I explored the trax library from google, but they seem to have deprecated the project. That's why I will change that into a lesson to show you how to build neural networks with julia. Yes, thats another language, but I think you will find the syntax very readable, and I hope you can appreciate the factor 10-100 speedup compared to python.