Practical ML for Engineers

What is Learning?

Weekly Plan

Tuesday

- 8h - 10h: Introduction
- 10h - 11h: Mathematical Foundations
- 11h - 12h: **Project**
- 13h - 15h: **Project**
- 15h - 16h: Model Evaluation

Friday

- 8h - 10h: Optimization
- 10h - 14h: **Project**
- 14h - 15h: Selected students presentations and discussion.
- 15h - 16h: Practical ML and Wrap up

Part 1: What is Learning?

ML is everywhere!

ChatGPT

LLMs

Self Driving Cars

Intrusion Detection

Stock Market Prediction

Central Ideas

What do all these applications have in common?

- Learning means getting better at some given task
- Machine learning is about writing computer programs that get better at given tasks by using **data**.

> ML is the “field of study that gives computers the ability to learn without being explicitly programmed”
>
> *(A. Samuel)*

Learning Algorithms

You have already seen ML (more or less):

- Algorithmics (BFS, …)
- Statistics (regression, PCA, ANOVA)

The main goal of ML is to go beyond **rule-based** algorithms.

Why ML?

- We live in a **big data world** -> might as well use it
- It beats any other approach on several types of problems
- Gives insights into human learning

And … Its’s **fun**.

A brief (and partial) history of ML

Mathematical pre-history

- **1763**: Bayes theorem (T. Bayes)
- **1805**: Least squares fitting (A. M. Legendre, G. F. Gauss)

Early days

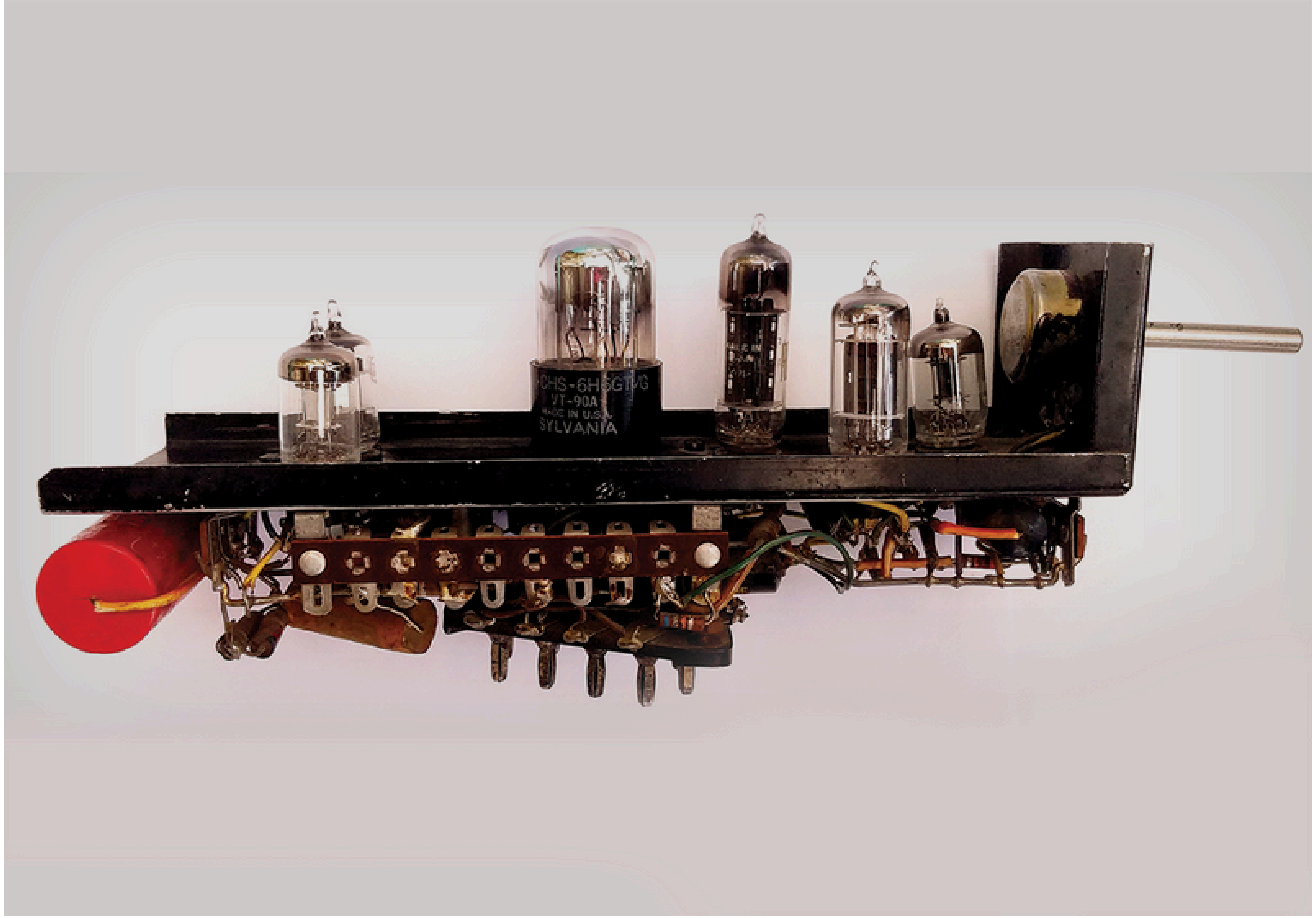
- **1943** Mathematical model of artifical neuron (Pitts and Mc Culloch)
- **1951** First implementation of a neural network (M. Minsky)
- **1952** First “intelligent” program (A. Samuel)
- **1957**: The *Perceptron* (F. Rosenblatt)
- **1974 - 1980**: First AI winter

The AI spring

- **1982 - 1985**: Backpropagation algorithm (S. Linnainmaa, P. Werbos, D. Rumelhart, G. Hinton, R. Williams)
- **1989**: Start of modern RL (C. Watkins)
- **1995**: Random Forest (T. K. Ho)
- **1997**: Deep Blue (IBM)

Modern Boom

- **2011 - 2012**: *AlexNet* (A. Krizhevsky, I. Sutskever, G. Hinton), ML becomes good at vision
- **2016**: *AlphaGo* (DeepMind), ML beats humans at complex games
- **2017**: Transformer Architecture
- **2018**: *AlphaFold*, ML performs a breakthrough in natural science
- **2022**: ChatGPT, ML becomes good at natural language



Single neuron from the _SNARC

Part 2: Learning in Practice

The ML Lifecycle

1. **Collect some data**
2. **Write a program that “learns” from it (training)**
3. **Use the program to “predict” what happens for new, unseen data**

The 3 types of ML

Traditionally, ML has been split into 3 types, depending on how data is collected and what kind of data is available:

- **supervised**
- **unsupervised**
- **reinforcement learning**

Supervised Learning

In this lecture, we will focus on supervised learning.

- In supervised learning, we have **labelled data** consisting of **training examples**.
- The dataset consists of collections of **inputs** and **targets**.

$$\mathcal{D} = \left\{ \left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)} \right) | i = 1, \ldots, n \right\}$$

Our goal is to find a "model" f that is able to predict $\boldsymbol{y}$ for unseen values of $\boldsymbol{x}$.

Remark on notation

- **Boldface** is used to denote vectors
- $\boldsymbol{x}$ denotes a single training example, the associated response being denoted by y
- Individual components $\boldsymbol{x}_j$ denote single features of a given example
- Superscript denote different training examples: $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots$

Lab 1: A first supervised learning algorithm

Your first supervised learning algorithm: Linear Regression

In this small lab, you will fit a linear model by hand to predict CO2 emissions of cars.

Linear Model

A linear model is a model which, given some data point x predicts the respons y using a linear function:

$$\hat{y} = f(x) = a \cdot x + b.$$

Preparation

1. Clone the week 1 repository:

```
git clone https://github.com/isc-hei-classrooms/301-Week1-ML-project
git lfs pull
```

2. Create Python environment:

```
uv sync
```

3. Write all your code in the notebook at

```
notebooks/mini_lab_1_fuel_consumption.ipynb
```

Your mission

The Canadian government wants to predict how much CO2 a given type of car will emit by looking at the characteristics of the car (fuel type, consumption, …).

Your task is to:

> - identify the main drivers of CO2 emission in a car
> - build a small predictive model for CO2 emission

Tasks

1. Load the dataset in Pandas.
2. Explore the dataset (plots, statistics, …).
   - are there any obvious correlations? What are the main drivers of CO2 emission?
   - are there outliers? How can we check?
3. Lets say we want to predict CO2 emission with the following simple model:

$$y = a \cdot \text{Combined (L/100km)}$$

   - Plot the model for 3 values of the parameter: $a = 10, 20, 30$. What do you observe?
   - Can you find a way to quantify with a single number how well each of the above models performs?
4. Propose a way to find the "best" value of a (the definition of best is up to you). Implement it.
5. Produce plots and/or numbers to convince the Canadian government that the model you proposed in (5) is good.
6. (optional) Propose an extension of you model that contains more parameters.

References

- The Pandas cheatsheet
- Guide on working on Python projects with uv
- The classic reference on Probabilistic Machine Learning by K. Murphy
- Code Repository for S. Raschka's *Machine Learning with PyTorch and Scikit-Learn* book