PROGRAMMING IN PYTHON II

Project Design and Outline

Michael Widrich
Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

Outline

PROJECT DESIGN

PROJECT DESIGN

MOTIVATION

Motivation (2)

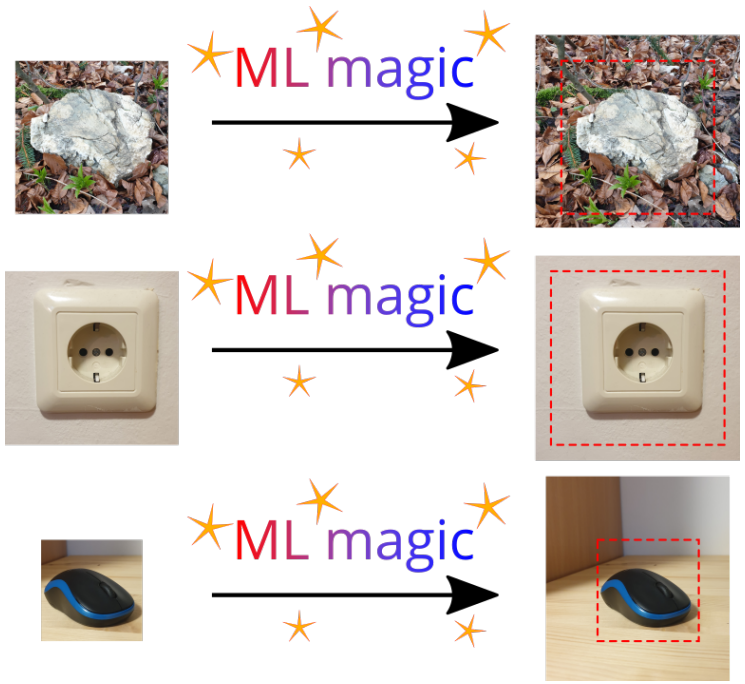
- When designing or implementing an ML project, you have to consider and constantly re-evaluate multiple aspects

Motivation (2)

- When designing or implementing an ML project, you have to consider and constantly re-evaluate multiple aspects
- Spoiler alert: The choice of the ML method itself is only one aspect of many

Motivation (2)

- In this unit we will go through the outline of the project design
- We will cover the details during the semester
- We will use our ML project as example



PROJECT DESIGN

OVERVIEW

Project Design

- Common important aspects (in my experience) as checklist:
  1. What is the project goal?
  2. What data do you have? What data do you need? What does the data look like?
  3. What hardware do you have? What hardware could you have?
  4. What ML method(s) should you use?
  5. How to evaluate the methods/models?

Project Design

- Common important aspects (in my experience) as checklist:
  1. What is the project goal?
  2. What data do you have? What data do you need? What does the data look like?
  3. What hardware do you have? What hardware could you have?
  4. What ML method(s) should you use?
  5. How to evaluate the methods/models?
- There is no one-fits-all solution! Specific tasks require specific considerations!

PROJECT DESIGN

PROJECT GOAL

Project goal

– What is the project goal?

Project goal

- What is the project goal?

- Very important aspect and often overlooked
- Requires communication with people from different fields, including management
- DO NOT make simplifications here! Make sure you are aware of the real (end) goal and communicate this!
- Rinse and repeat to overcome language barriers

PROJECT DESIGN

DATA

Data (1)

– What data do you have? What data do you need? What does the data look like?

Data (1)

- What data do you have? What data do you need? What does the data look like?

- Data is money. Big data is big money.

Data (1)

– What data do you have? What data do you need? What does the data look like?

- Data is money. Big data is big money.
- Sometimes the goals will follow from sufficiently large existing data
  - Best case but rather rare (our hunger for data is only limited by computational restrictions!)

Data (1)

– What data do you have? What data do you need? What does the data look like?

- Data is money. Big data is big money.
- Sometimes the goals will follow from sufficiently large existing data
  - Best case but rather rare (our hunger for data is only limited by computational restrictions!)
- Sometimes the goals will follow from existing but insufficiently large data
  - Common case
  - Has influence on choice of ML method
  - Allows for educated guesses at sufficiently large data size
  - Can be starting point for collecting more data

Data (2)

- Sometimes the goals are not backed up by any data
  - Very tricky and potentially dangerous!
  - You would have to make guesses about how much and which data would be needed
  - You would have to make guesses about the ML method performance in advance
  - You will need to interface with the data collection process
    - → First get small dataset, then collect more
  - You might waste a lot of time and money

Data (2)

- Sometimes the goals are not backed up by any data
  - Very tricky and potentially dangerous!
  - You would have to make guesses about how much and which data would be needed
  - You would have to make guesses about the ML method performance in advance
  - You will need to interface with the data collection process
    - → First get small dataset, then collect more
  - You might waste a lot of time and money
- A dataset might be unsuitable for your purposes
  - Biases, artifacts, labeling errors, . . .

Data (3)

- Make the most of your data
  - Talk to experts in the field of application/read up on the topic
  - Perform analysis of the data (e.g. clustering) and look for possible issues (e.g. biases, batch-effects)
  - Check if there is auxilliary data available
    - Pre-training on similar data, unused sorted-out data, data that is not suitable for training but for evaluation, . . .
  - Perform data preprocessing and augmentation
    - Normalization, oversampling, cross-validation splits, data augmentation, . . .

PROJECT DESIGN

HARDWARE/SOFTWARE

Hardware/Software

– What hardware/software do you have? What hardware/software could you have?

Hardware/Software

- What hardware/software do you have? What hardware/software could you have?

- CPU, GPU, or TPU based?
- Size of RAM and disk storage?
- Hardware compatible with ML software? Software restrictions from company/collaborations?
- Short term or long project?
  - Rent or own? Little compute over long time or lots of compute over short term?
  - My recommendation: First design/implement/experiment on owned hardware, then perform final tuning on rented hardware if needed

PROJECT DESIGN

ML METHODS

Methods

– What ML method(s) should you use?

Methods

– What ML method(s) should you use?

- Depends on goal, data, and hardware
- You will need a theoretical understanding of the methods to judge which ones to consider
  - Literature research
  - Later semesters of AI study
- Start with baselines/less complex methods and models
  - Statistics, logistic regression, SVM, . . .
  - Check Supervised Learning before Reinforcement Learning and Unsupervised Learning

PROJECT DESIGN

EVALUATION

Evaluation

– How to evaluate the methods/models?

Evaluation

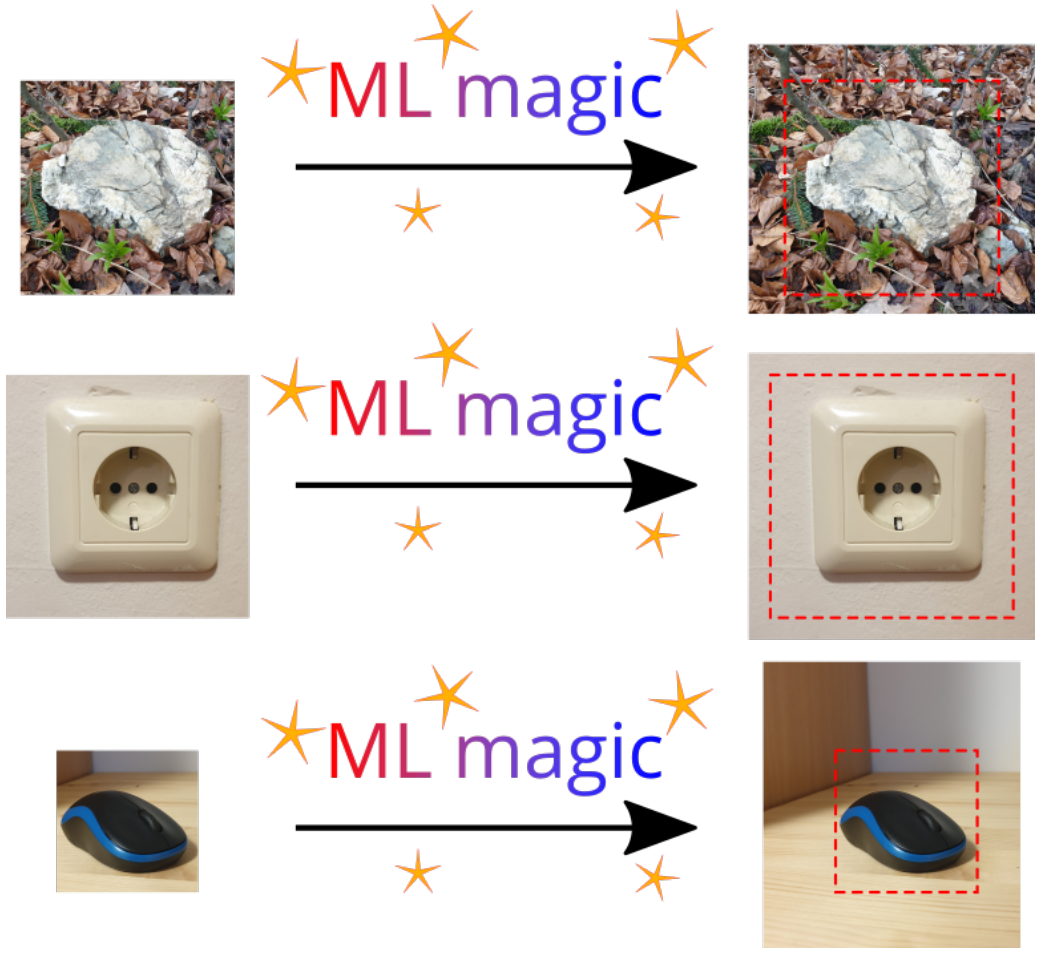
- How to evaluate the methods/models?
- Which score/performance measure?
- Do you need to correct for biases?
- Which aspects of the goal are more important?
- What do you want to generalize to?

PROJECT DESIGN

PYTHON II PROJECT

Python II Project: Goal (1)

- Extrapolate image data



Python II Project: Goal (2)

- “Extrapolate image data”
- End goal: High score on challenge server leader-board
  - Image is fed into model and model extends image borders with new plausible pixels
  - Size of created pixel borders should be freely selectable
  - Images should be grayscale
  - Should work on all kinds of grayscale images

Python II Project: Goal (2)

- “Extrapolate image data”
- End goal: High score on challenge server leader-board
  - Image is fed into model and model extends image borders with new plausible pixels
  - Size of created pixel borders should be freely selectable
  - Images should be grayscale
  - Should work on all kinds of grayscale images
- What is “plausible”?
  - Luckily, the challenge server decides for us: “Plausibility” is measured by average negative squared error per pixel

Python II Project: Data (1)

- We will create our own dataset
- We will have the following data:
  - JPG images up to 850kB
  - 100 images per student
  - Assumption: We collect 30k valid images

Python II Project: Data (2)

- We will crop out small images and pretend they are the original images
  - → we do not need to collect labels!
- This is probably a case with sufficient data for training our methods
  - We can use data augmentation to increase the dataset size
  - We could use additional data from the internet but it will not be necessary
- We will have to
  - Clean up the raw data (exclude invalid files)
  - Preform analysis and preprocessing
  - Perform data augmentation

Python II Project: Hardware/Software and Methods

- Hardware/Software
  - Harware is up to you (see introduction slides)
  - Python 3.6 or higher (recommended: 3.7)
  - PyTorch
- Methods
  - Simple Convolutional Neural Network (CNN)
  - You may also use other NN types/more complex settings
  - Design and fine-tuning is up to you

Python II Project: Evaluation

- Challenge server score determines the evaluation method
  - Will use the squared error per pixel

- “Should work on all kinds of grayscale images”
  - → Will haunt us at evaluation section later in the semester but we will do our best

Assignment, UE lesson, next lecture

- Next lesson we will learn about
  - resources for ML,
  - version control using git,
  - hash values and hashing in Python