PROGRAMMING IN PYTHON II

Introduction

Michael Widrich
Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

Contact

Michael Widrich et al.

Institute for Machine Learning
Johannes Kepler University
Altenberger Str. 69
A-4040 Linz

E-Mail: `python@ml.jku.at`

Only mails to this address will be answered!

Institute Homepage: `https://www.ml.jku.at`
Moodle: `https://moodle.jku.at`

Outline

1. **Recap Python I**

2. **Outline Python II**

3. **Schedule for today's lecture**

RECAP PYTHON I

Recap Python I

- In Python I we have learned about programming and Python…

Recap Python I

- In Python I we have learned about programming and Python. . . a lot of it actually:
  - □ Basics about hardware and datatypes
  - □ Command line, Python Interpreter
  - □ Usage of PyCharm Editor
  - □ Python scripts
  - □ Debugging
  - □ Python syntax/style
  - □ Floats, ints, strings, lists, dictionaries
  - □ Conditions, loops, list comprehensions
  - □ Exceptions

  ⋮

Recap Python I

⋮

- □ Functions
- □ Regular expressions
- □ Classes
- □ os/sys (Python as pseudo shell-script)
- □ Matlpotlib/Pyplot (Plotting in Python)
- □ Numpy (efficient computation in Python)
- □ Multiprocessing (subprocesses in Python)
- □ Numba (compiling and speeding up Python programs)
- □ PyTorch (optimized programming for ML)

Recap Python I

⋮

- □ Functions
- □ Regular expressions
- □ Classes
- □ os/sys (Python as pseudo shell-script)
- □ Matlpotlib/Pyplot (Plotting in Python)
- □ Numpy (efficient computation in Python)
- □ Multiprocessing (subprocesses in Python)
- □ Numba (compiling and speeding up Python programs)
- □ PyTorch (optimized programming for ML)

…need a recap? Materials available here:

`https://github.com/widmi/programming-in-python`

OUTLINE PYTHON II

What awaits you in Python II?

What awaits you in Python II?

- A full-fledged Machine Learning project

What awaits you in Python II?

- A full-fledged Machine Learning project
  - □ Collection of data
  - □ Setup of a project with git integration
  - □ Analysis of the data
  - □ Preprocessing of the data
  - □ Loading of the data
  - □ Implementation of the Neural Network (inference)
  - □ Implementation of the Neural Network (training)
  - □ Implementation of data augmentation
  - □ Evaluation of performance

Goals of this course

- **Main goal:** You will be able to set up your own ML project
  - □ Implementation in Python and PyTorch
  - □ Usage of git to access resources on github
  - □ Fundamentals and pitfalls in data preparation
  - □ Fundamentals and pitfalls in design, training, and evaluation of a ML model
  - □ Knowledge about where theory and math comes in (we will keep it on the practical side!)
  - □ Practical tools and knowledge on how to implement a ML project

Lecture style (1)

- Interactive lecture style

  → Please bring your laptops or share one if possible!

Lecture style (1)

- Interactive lecture style

  → Please bring your laptops or share one if possible!
- Attendance is not compulsory

Lecture style (1)

- Interactive lecture style

  → Please bring your laptops or share one if possible!
- Attendance is not compulsory
- Main platform: `https://moodle.jku.at`
  - □ Video streams and chat for questions during stream
  - □ Course materials, slides, source code
  - □ Forums for announcements, assignment related questions, general questions
  - □ Assignment sheets and submission of exercises
  - □ Multiple-Choice exams

Lecture style (2)

- Questions?

  → During lectures: Use dedicated moodle chat or ask us during lecture breaks

  → After lectures: Ask your collegues, use the student help-desk, ask in the moodle forum, or write us an informal email (`python@ml.jku.at`)

Lecture style (2)

- Questions?

  → During lectures: Use dedicated moodle chat or ask us during lecture breaks

  → After lectures: Ask your collegues, use the student help-desk, ask in the moodle forum, or write us an informal email (`python@ml.jku.at`)

- Stick to the moodle forum rules!

Lesson structure (1)

- This course is structured in multiple units
- Each unit addresses a specific topic

Lesson structure (1)

- This course is structured in multiple units
- Each unit addresses a specific topic
- 3 Python code files per unit:
  1. Explanation/Demonstration: Python file that explains a topic and demonstrates solutions in Python
  2. Tasks: Short voluntary example tasks
  3. Solutions: Example solutions for the tasks

Lesson structure (2)

- For each unit in the lecture there are two parts:
    1. Theoretical part: Explanation/Demonstration file will be shown and discussed
    2. Practical part: Students work on solving the tasks
- A preliminary schedule will be available via moodle

Grading (1)

- Assignment 1 (35 points):
  - Data collection (5 points)
  - Data analysis (15 points)
  - Data preprocessing/loading (15 points)
- Assignment 2 (ML challenge) (55 points + 10 bonus points):
  - Participation in ML challenge (points based on your ML model performance)
- 1 Multiple-choice exam (10 points):
  - Online multiple-choice exam via Moodle at fixed date/time (see KUSSS for dates)

Grading (2)

- Assignment 1 consists of multiple exercises
- Exercises will be graded **automatically**
  - □ Stick to the **Instructions for submitting homework** in moodle
  - □ You will receive unit-testing scripts to test your submission on a sub-set of input files at home
  - □ The unit-testing script does not guarantee your points!
- $\geq 40\%$ on the exam is required to pass the course

Grading (3)

- Start working on exercises and project in due time
- Automated plagiarism checks – don’t copy code from your colleagues or the internet (or you receive 0 points)
  - You will need to submit your code for the ML challenge
- 2+ weeks time per exercise (after required topics have been covered)
  - You will receive example solutions for the exercises after each exercise deadline

Grading (3)

- Start working on exercises and project in due time
- Automated plagiarism checks – don’t copy code from your colleagues or the internet (or you receive 0 points)
  - □ You will need to submit your code for the ML challenge
- 2+ weeks time per exercise (after required topics have been covered)
  - □ You will receive example solutions for the exercises after each exercise deadline
- Project should be possible with laptop using CPU and 4GB RAM but GPU and more RAM will be faster
  - → you may also use cloud services (google, amazon, lambda labs, . . .)

SCHEDULE FOR TODAY’S LECTURE

Schedule for today's lecture

1. ML project design
2. Information about data collection task