Introduction to Artificial Intelligence

Dr. Thomas Keller
C. Büchner, C. Grundke, A. Kauffmann

University of Zürich
Spring Semester 2022

Exercise Sheet 10

Due: May 18, 2022
Points total: 20 marks

Exercise 10.1 – Learning **4 marks**

Consider the problem faced by an infant learning to speak and understand a language. Explain how this process fits the general learning model and describe the following aspects:

- percepts and actions of the infant,
- types of learning the infant must do, and
- available example data.

Exercise 10.2 – Decision Trees **9 marks**

(a) Consider the following data set comprised of three binary input attributes (A_1,A_2, and A_3 where 0 denotes "false" and 1 denotes "true") and one binary output:

Examples	A_1	A_2	A_3	Output y
x_1	1	0	0	0
x_2	1	0	1	0
x_3	0	1	0	0
x_4	1	1	1	1
x_5	1	1	0	1

Use the Learn-Decision-Tree algorithm from the book (Figure 19.5) to learn a decision tree for the above data. Provide the final decision tree and the computations you made to determine the attribute split at each node. (5 marks)

(b) Construct and provide a data set (set of examples with attributes and classifications) that causes the decision-tree learning algorithm to find a non-minimal-sized tree.

Note: By the size of a tree we mean the length of the longest path from the root to any leaf, i.e., the depth of the tree.

Hint: It is possible to construct such a data set with no more than 3 input attributes. (3 marks)

(c) We never test the same attribute twice along one path in a decision tree. Why would this never be beneficial? (1 mark)

Exercise 10.3 – Programming 7 marks

Download the archive `sheet10-programming.zip` from MOODLE. It contains two files:

`data.py` contains utilities to generate the training data. You should not change this file.

`learning.py` contains the skeleton for your implementation of the following exercises. This file is executable and plots the data along with the ground truth function f from which the (noisy) data was generated.

(a) Implement the function `linear_regression` to find the optimal w_0 and w_1 for the given data (using the L_2 loss-function).

Hint: You can run `learning.py` with the option `--linear-regression` to test your implementation. (3 marks)

(b) Implement the function `update_weights` to execute one iteration of gradient-descent (using the L_2 loss-function). Then implement the function `gradient_descent` to iteratively optimize w_0 and w_1 for the given data. Instead of iterating until *converged* (as outlined in the book), we would like you to iterate `iterations` times, where `iterations` is a parameter of the function.

Hint: You can run `learning.py` with the option `--gradient-descent` to test your implementation. (4 marks)

Submission rules:

- Exercise sheets must be submitted in groups of three students. Please submit a single copy of the exercises per group (only one member of the group does the submission).
- Create a single PDF file (ending `.pdf`) for all non-programming exercises. Use a file name that does not contain any spaces or special characters other than the underscore "_". If you want to submit handwritten solutions, include their scans in the single PDF. Make sure it is in a reasonable resolution so that it is readable, but ensure at the same time that the PDF size is not astronomically large. Put the names of all group members on top of the first page. Make sure your PDF has size A4 (fits the page size if printed on A4). Submit your single PDF file to the corresponding exercise assignment in MOODLE.
- For programming exercises, only create those code text files required by the exercise. Put your names in a comment on top of each file. Make sure your code compiles and test it. Code that does not compile or which we cannot successfully execute will not be graded. Create a ZIP file (ending `.zip`, `.tar.gz`, or `.tgz`; *not* `.rar` or anything else) containing the code text file(s) (ending `.py`) and nothing else. Do not use directories within the ZIP, i.e., zip the files directly.
- Do not upload several versions to MOODLE, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.