

# DATA.ML.200 Pattern Recognition and Machine Learning

## Exercise Set 1: Single neuron gradient descent

### 1. *Install Python.*

You should know how to do that. Recommended procedure is to use Anaconda and make own environment for the course (e.g. “dataml200”).

Lecturer uses Python version 3.9 in his code examples, and therefore it is recommended.

You may use your favourite IDE (PyCharm, VS Code, Spyder, Emacs etc), but make sure to always return exercise answers in the requested formats.

### 2. **pen&paper** *Derive gradients for a single two input perceptron.*

This was actually your Homework 2, but you will need it in this exercise so you must do it before proceeding. In gradient descent the gradients will be used to update the perceptron parameters (weights)  $w_0$  and  $w_1$ .

### 3. **python** *Single neuron gradient descent. (10 pts)*

During the lectures you were given a Python example for the “hobbit vs. elf” classification task. The classifier was a single neuron (perceptron) trained with the Hebbian rule. The following two lines of code implemented the Hebbian rule:

```
w1_t = w1_t+learning_rate*(y_tr[x_ind]-y)*x
w0_t = w0_t+learning_rate*(y_tr[x_ind]-y)*1
```

In the previous question you derived the weight update rules for gradient descent. Next, use these rules to optimize the perceptron weights instead of the Hebbian rule. You may study the effect of the learning rate, and select a suitable one.

**Note:** You may use the code snippet from the lecture notebook if you wish.