

## Machine Learning and Intelligent Systems

### Project 2: The Perceptron

The perceptron is a machine learning algorithm for learning a binary classifier of the form:

$$\hat{y} = h(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

where  $\mathbf{w}$  is a vector of real-valued weights and  $b$  is denoted the bias

#### Objectives

By executing this project, you will be able to

1. Gain understanding on the principles governing the perceptron algorithm
2. Strengthen your understanding of derivation and the gradient descent algorithm.
3. Familiarize with the process of training, validation and testing by splitting the data on your own.
4. Improve your proficiency in the use of coding tools and libraries for machine learning by building a full project from scratch .

### Part I – Implementing a perceptron algorithm from scratch:

**Task 1.** Your first task will be to code the perceptron algorithm in a file called `perceptron.py`. This time, you will not be provided with a skeleton file. To be consistent with the previous project, you will need to create a class named `Perceptron`, which should have the following two methods:

1. `train` – with the necessary steps to train a perceptron
2. `predict` – with the necessary steps to predict the labels  $y$  of a group of samples  $X$ .

You may add any other functions that you consider necessary.

Make sure you debug your code, to verify that it works accordingly. You may consider comparing it against scikit-learn implementation for validation.

**Task 2.** Document your code.

### Part II – Using the perceptron:

You will now use your coded perceptron to classify a handwritten digit as 0 or 1. To this end, create a jupyter notebook denoted `experiment.ipynb`. This will contain all your experiments. We will work with the UCI ML hand-written digits datasets that is contained in scikit-learn.

**Task 3.** Load the dataset and using the `load_digits` function in scikit-learn. Check its documentation to see how you can use it ([link](#)). The dataset contains digits from 0 to 9. Discard all digits different from 0 and 1.

**Task 4.** Using the remaining dataset, split your data into training, validation and test. You can use the [train test split](#) function from scikit-learn.

**Task 5.** Train your perceptron using the training set. Use the validation set to choose a model. Report the selected model's accuracy using the testing set. Report your results.

Hint: The digits dataset is a dataset of images. You need to convert them from a 2D array to a 1D one.

## Part III – Getting familiar with the gradient:

**Task 6.** Solve exercises 5.1-5.3, 5.5-5.7 and 5.9 from Chapter 5 in [The Mathematics of Machine Learning](#)<sup>1</sup>.

**Bonus point:** Solve exercise 5.8

### Report:

You need to prepare a 1-page report explaining how your model was trained and in which ways you used the validation and test sets. Report the accuracy of your model. the results obtained in the validation set and the strategy you used to choose a model. Use as many pages as you require for Task 6. For task 6, if it is easier, you can deliver scanned copies of derivations done by hand.

### Deliverables:

Upload a zip file named <group-name>.zip containing the report, the perceptron.py, the experiment.ipynb file and any instructions required to run it. If you scanned the derivations, an additional pdf with them.

### Evaluation:

| Criteria  | Score |
|---|-------|
| Your code runs and works as expected for the given problem  | 6     |
| Your code is well-documented  | 1     |
| Your model performs better than chance on a test set for the digits problem (0/1 classification). | 2     |
| Among all submissions, your model is the fastest during testing                                   | 2     |
| Your answers to part III are correct  | 4     |
| Your report   | 5     |

### Important:

- Failing to submit a report leads to a mark of zero (0).
- If ChatGPT is used, failing to report it and explaining its use leads to a mark of zero (0).
- A group will be chosen at random to present their solution during the lecture. Failing to justify the submitted solution leads to a mark of zero (0).

---

<sup>1</sup> <https://mml-book.github.io/book/mml-book.pdf>