Exercise Sheet 0 - Machine Learning with Python
Prof. Dr. Paul Swoboda

Note:
For the practical part of the course, we will utilize Python to implement various Machine Learning techniques. We recommend using the Anaconda distribution `https://www.anaconda.com/download` for environment and package management.
To avoid version conflicts and simplify installation, Python programs are often added to environment files, in which the packages used and their versions are specified. Package managers such as Conda can use such a file to automatically install the required packages in an encapsulated environment. Familiarize yourself with the use of Conda environments `https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html`.

Exercise 1: Download Numpy (1 Point)

For this task you need numpy `https://numpy.org/`. If you have installed miniconda, you may need to install numpy. You can install packages via the terminal with the following command:

```
conda install numpy
```

numpy should already be pre-installed in Anaconda. Open a Jupyter Notebook and calculate the binary logarithm of 65536 using following command:

```
numpy.log2(65536)
```

Exercise 2: Guessing Single Layer Perceptron (4 Points)

For the next exercise, create a Jupyter Notebook using only NumPy and Pandas as the imported modules, and record your answers to the questions within the notebook.
Download the dataset from Ilias: Data.csv. The objective is to determine whether a given house is classified as expensive. To accomplish this, we will build a model to predict whether a house falls into the "too expensive" category. For this task, we employ a logistic regression model, which maps the input features, such as square footage, number of bedrooms, location, and other relevant attributes, to the probability of the house being classified as expensive or not given our data. The process is carried out in two steps: first, an affine transformation is applied to the feature data, and then the resulting output is passed through a sigmoid function.

To start, we need to implement the equation below:

$$X' = X \cdot W + B$$

Given the dataset, what are the dimensions of the feature tensor X, the weight tensor W, and the bias tensor B?
The target $\hat{y}$ is a binary variable (classifying whether the housing is too expensive or not), in order to transform our model X' into the interval [0,1], we use the sigmoid function given by:

$$\sigma(X') = \frac{1}{1 + e^{-X'}}$$

What is the dimension of $\sigma(X')$?
The sigmoid function is a monotonically increasing, differentiable function with an output range of [0,1]. Its graphical representation is shown in Figure 1.

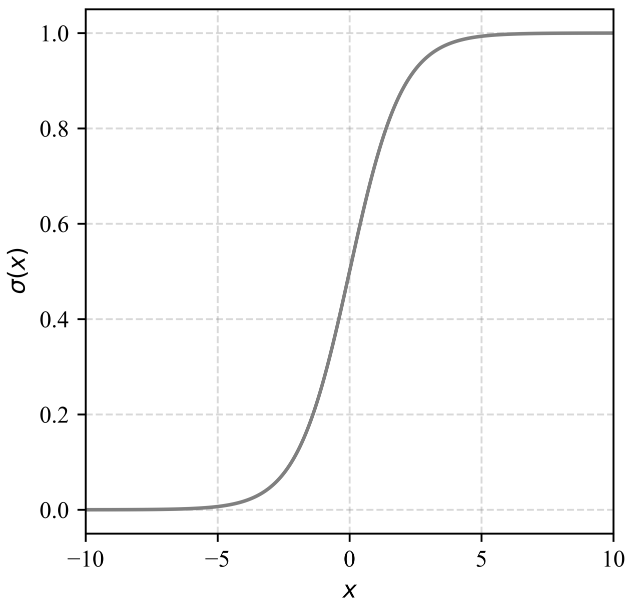

Figure 1: The figure displays the sigmoid function plot.

Next, we want to estimate suitable values for our weights. Implement a random search using the numpy.random functionality to minimize the following expression:

$$\min_{W,B} \left\{ \sum_i \left[\hat{y}_i - \sigma(X')_i \right]^2 \right\}$$