Programming Practice I

Yiwen Liao
Institute of Signal Processing and System Theory

1 Overview

In this practice, you can learn the basic knowledge of Python. In particular, by doing this practice, it is expected that you can:

- get familiar with the basic syntax of Python;
- get the basic idea of NumPy;
- get the basic idea of Matplotlib for visualization.

Hint: Refer to the cheat sheets for Python/NumPy/Matplotlib available in ILIAS.

2 Tasks

In this section, you can start the programming practice task by task. Please pay attention to the hints as well as the API documents online (by clicking the corresponding hyperlinks below).

2.1 Import Packages

Please create a notebook and import NumPy and the pyplot package of Matplotlib. Specifically, the NumPy should be renamed as `np` and `pyplot` should be renamed as `plt` when you import these two packages in this task. Hint:

- refer to the tutorial for using notebook;
- refer to the API documents for NumPy;
- refer to the API documents for Matplotlib.

2.2 1D Dataset

Please create a 1D dataset $\{(x_1, y_1), (x_2, y_2), \dots, (x_{100}, y_{100})\}$, where x_i is evenly spaced w.r.t. the range $[-1, 1]$, and $y_i = 0.1 \cdot x_i + x_i^2 + x_i^3$. Subsequently, please write a function

Manuscript updated on 19.04.2022

`create_dataset(n_samples=100)` that can return the same dataset but with user-defined number of samples. Finally, please write a `Dataset` class, the initialization method of which has the parameter `n_samples=100`, and this class has a function `load_data()` that returns the same dataset but with user-defined number of samples. Hint:

- x_i denotes input data, while y_i denotes ground truth;
- 1D means that x_i has one feature only;
- consider using `np.linspace()` to generate x_i;
- consider using `np.power()`;
- how is a function defined in Python?
- how is a class defined in Python?

2.3 Plot the 1D Dataset

Please plot the 1D dataset above. The figure size should be set to 5×5. The x-axis should only consist of five ticks from -1 to $+1$. Hint:

- consider using `plt.plot()` for plotting lines and curves;
- more materials can be found here.

2.4 2D Dataset

Please create a 2D dataset $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_{100}, y_{100})\}$, where each dimension of $\boldsymbol{x}_i$ is sampled from a standard normal distribution. Define $y_i = 0$ if $||\boldsymbol{x}_i||_2^2 < 1$, else $y_i = 1$. The seed should be set to 42. Hint:

- $\boldsymbol{x}_i$ denotes input data, while y_i denotes ground truth;
- 2D means that $\boldsymbol{x}_i$ has two features;
- refer to API for `np.random` for more materials;
- what is the goal of fixing seeds?

2.5 Plot the 2D Dataset

Please plot the 2D dataset above. The figure size should be set to 5×5. The samples affiliated to class 1 should be *red triangles*, while the rest samples are *blue circles*. Add one legend to the figure to indicate the two classes. Hint:

- consider using `plt.scatter()`;
- more materials can be found here.

2.6 A Simple Convolution

Please create a matrix $X = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix}$ and a 2×2 filter W initialized with ones. Calculate the convolution output (stride=1, dilation=1, no padding) using NumPy. Subsequently, write a function `convolve(X, W)` to calculate any convolution regarding the given `X` and the filter `W` (still: stride=1, dilation=1, no padding). Hint:

- consider using `np.ones()`;
- we can assume that both `X` and `W` are matrices for simplicity;
- how to calculate convolution?
- more materials can be found here.