Exercise session 13

Python's ecosystem for scientific computing.

Advanced Programming - SISSA, UniTS, 2024-2025

Giuseppe Alessio D'Inverno

17 Dec 2024

Exercise 1: NumPy

1. **Array creation and manipulation**
    i. Create a 2D NumPy array of shape 5×5 filled with random integers between 1 and 10.
    ii. Extract the second row, third column element, and the diagonal elements.
    iii. Reshape it into a 1D array of shape 1×25.
2. **Linear algebra operations**
    i. Generate two 3x3 matrices with random integers from 1 to 10 and perform element-wise and matrix-matrix multiplication.
    ii. Create a 3x3 matrix with random values, compute its inverse and determinant.
3. **Statistical analysis**
    i. Generate a 1D NumPy array with 20 random integers between 1 and 100.
    ii. Calculate the mean, median, standard deviation, and variance.

Exercise 2: SciPy (1/2)

1. **Solving a linear system of equations**
   i. Define a 100×100 *sparse* tridiagonal matrix A, with 2 over the main diagonal, and -1 over the first lower and upper diagonals.
   ii. Let $b = Ax_{\text{ex}}$ where $x_{\text{ex}} = [1, 1, \ldots, 1]^T \in \mathbb{R}^{100}$
   iii. Solve the linear system $Ax = b$ and compute the residual $\|b - Ax\|$ and the error $\|x - x_{\text{ex}}\|$ in norm 1, 2 and infinity.
2. **Function optimization**
   i. Consider the function $f(x) = \sin(\pi x)\exp(-x/10)$ over the interval $[-2, 4]$.
   ii. Plot the function using Matplotlib to visually identify potential minima.
   iii. Use `scipy.optimize.minimize` with different initial guesses to find these minima.

Exercise 2: SciPy (2/2)

3. **Data interpolation and integration**
   i. An electric vehicle charging station erogates the following series of energy measurements over time:

```
time = np.arange(0, 46, 3) # Hours.
energy = np.array([27.29, 23.20, 24.93, 28.72, 27.60, 19.06, 24.85, 21.54, 21.69, 23.23, 22.43, 26.36, 24.28, 22.36, 23.33, 23.00]) # kW.
```

   ii. Use SciPy to build a cubic interpolator of these data points.
   iii. Evaluate the interpolator over 1000 equispaced nodes between 0 and 45 and plot the values obtained.
   iv. Integrate the interpolant over $(0, 45)$.

Exercise 3: pandas (1/2)

1. **DataFrame operations and visualization**
    i. Import the `sales_data.csv` dataset as a pandas DataFrame.
    ii. Extract data from the 'South' region, sort them by descending 'Quantity' and add a new column 'Total revenue' $=$ 'Quantity' $\times$ 'Price'.
    iii. Visualize trends of 'Total revenue' by 'Date' (line plot) and by 'Product' (bar plot).
2. **Exploratory data analysis with the *iris* dataset**
    i. Load the `iris` dataset from seaborn.
    ii. Group the data by 'species' and compute summary statistics for `sepal_length` and `sepal_width`.
    iii. Use seaborn to plot the histogram of the sepal length distribution for each species.
    iv. Use seaborn to generate a scatter plot of sepal width vs. sepal length.

Exercise 3: pandas (2/2)

3. **Time series analysis with real data**
    i. Import the `weather_data.csv` dataset.
    ii. Resample the dataset to compute monthly averages.
    iii. Computing a 7-day rolling mean.
    iv. Visualize the original data and the rolling mean using line plots.

Exercise 4: designing a Neural Network for binary classification

1. Import the `XORDataset` from `dataset.py`.
2. Design a neural network in PyTorch that correctly classify the samples.
3. As a loss function, use `nn.BCEWithLogitsLoss()`.

Exercise 5: code obfuscation

Without executing the program, determine the output of the code contained in `wish.cpp`.