Lab Course Machine Learning

In-Class Exercise Sheet 2

Prof. Dr. Dr. Lars Schmidt-Thieme
Jung Min Choi

HiWi: Harish Malik

1 Matrix Computations Using `numpy`

1. Matrix Multiplication:
   a) Create two matrices A and B of size 3×3 using `numpy`. Perform matrix multiplication of A and B. Verify the result using both `np.dot()` and the `@` operator.
2. Matrix Inverse:
   a) Generate a random 3×3 matrix C using `numpy.random.randn()`.
   b) Compute the inverse of matrix C using `numpy.linalg.inv()`. Check that $C \times C^{-1} \approx I$, where I is the identity matrix.
3. Matrix Transpose:
   a) Create a 3×2 matrix D and compute its transpose using `numpy.transpose()` or the `.T` attribute.
   b) Perform matrix multiplication of D^T (the transpose of D) with D.
4. Difference Between `np.dot()` and `@`:
   a) Create two 2D matrices and multiply them using both `np.dot()` and the `@` operator. Verify that the results are the same for these 2D matrices.
   b) Create two 1D vectors and compute their dot product using both `np.dot()` and `@`. Verify that the results are the same for these 1D vectors.
   c) Create two higher-dimensional arrays (e.g., A of shape $2 \times 3 \times 4$ and B of shape $2 \times 4 \times 5$). Compute their matrix multiplication using both `np.dot()` and `@`.
   d) Create two 3×3 matrices A and B. Perform element-wise multiplication using `numpy.multiply()` and contrast this with matrix multiplication using `np.dot()` and the `@` operator.

2 Univariate Linear Regression Using `numpy`

Implement a univariate linear regression model to fit a line through a set of data points using the Least Squares Estimation method.

1. Generate a set of $n = 100$ data points where the input feature x is randomly distributed between 0 and 10, and the target variable y is generated using the linear relation $y = 2x + 1 + \text{noise}$, where noise is a small random normal error. Fit a linear regression model to this data.
2. Generate another set of $n = 100$ data points where y follows a non-linear relation $y = x^2 + 3x + 5 + \text{noise}$. Fit a linear regression model to this non-linear data and compare the results with the actual curve.
3. Plot the original data points and the best-fit line using `matplotlib` for both the linear and non-linear data. Additionally, plot the actual curve for the non-linear data alongside the linear regression line.
4. Show the computed values of the slope and intercept for both the linear and non-linear datasets.