Training Report Day-19

27 June 2024

Introduction to NumPy

NumPy (Numerical Python) is a powerful, open-source library in Python used for numerical computations. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy is fundamental for scientific computing in Python and serves as the foundation for many other libraries like SciPy, Pandas, and Matplotlib.

Key Features of NumPy

1. **N-Dimensional Arrays**:
    - The core feature of NumPy is its ndarray object, which is a multi-dimensional array of elements, typically of the same type. These arrays allow for efficient storage and manipulation of large datasets.
2. **Mathematical Functions**:
    - NumPy provides a wide array of mathematical functions for operations such as trigonometry, statistics, linear algebra, and more, all optimized for performance.
3. **Broadcasting**:
    - Broadcasting allows NumPy to perform operations on arrays of different shapes without needing to copy data. It simplifies the implementation of mathematical operations.
4. **Integration with C/C++ and Fortran**:
    - NumPy can interface with code written in C, C++, or Fortran, allowing for high-performance numerical computation.
5. **Linear Algebra**:
    - NumPy includes functions for linear algebra operations such as matrix multiplication, eigenvalue decomposition, and singular value decomposition.
6. **Random Number Generation**:

- The library provides tools for generating random numbers, including random sampling from different probability distributions.

Practical Applications of NumPy

- **Data Analysis**: Efficiently handle and manipulate large datasets, performing complex calculations with ease.
- **Scientific Computing**: Solve problems in physics, chemistry, biology, engineering, and other fields requiring numerical methods.
- **Artificial Intelligence, Machine Learning & Data Science**: Form the backbone of many machine learning algorithms and frameworks, providing fast computation of matrix operations and statistical functions.
- **Image Processing**: Manipulate and analyze images, which are represented as arrays of pixel values.

Example:-

```
import numpy as np
l1 = [1, 2, 3, 4]
l2 = [5, 6, 7, 8]
type(l1)
arr1 = np.array(l1)
arr1
arr1.shape
type(arr1)
arr2 = np.array([l1, l2])
arr2
```

```
import numpy as np
id1 = np.identity(4, dtype = int)
id1
mat2 = np.linspace(10,100, 1000)
mat2
```