

Summary Numpy

Tuesday, December 17, 2024 1:28 AM

Lesson : Summary Numpy

NumPy:

NumPy is a fundamental library in Python for scientific computing. It offers powerful features for working with multidimensional arrays and performing various mathematical operations efficiently.

NumPy (Numerical Python) is an open source Python library that has a wide range of inbuilt functions.

Advantage of NumPy : NumPy is ideal for handling large amounts of homogeneous (same-type) data, offering significant improvements in speed and memory efficiency. It also provides high-level syntax for a wide range of numerical operations, making it a powerful tool for scientific computing and data processing on the CPU.

Key Concepts of NumPy:

1. Arrays: The Core Data Structure

What are Arrays? NumPy arrays are multidimensional collections of elements, all of the same data type. Unlike Python lists, NumPy arrays are designed for high-performance operations on large datasets.

There are several ways to create NumPy arrays:

• From Python Lists:

```
1 import numpy as np
2
3 data = [1, 2, 3, 4, 5]
4 arr = np.array(data)
5 print(arr)
```

```
[1 2 3 4 5]
```

• Using Built-in Functions:

`np.zeros(shape)`: Creates an array filled with zeros.

`np.ones(shape)`: Creates an array filled with ones.

`np.empty(shape)`: Creates an array with uninitialized elements (faster than zeros).

```
1 zeros_arr = np.zeros(5)
2 print(zeros_arr)
3
4 ones_arr = np.ones((2, 3))
5 print(ones_arr)
```

```
[0. 0. 0. 0. 0.]
```

```
[[1. 1. 1.]
```

```
[1. 1. 1.]]
```

Data Types: NumPy arrays can hold various data types like integers (int), floats (float), strings (str), etc. You can specify the data type during creation or let NumPy infer it. `dtype` attribute reveals the data type of the elements in the array

1. Array Attributes and Operations:

Shape: The shape of an array represents its dimensionality and size. Use `.shape` to get the shape as a tuple.

```
1 print(arr.shape)
```

```
(5,)
```

Indexing and Slicing: Similar to Python lists, one can access elements using square brackets `[]` and slice sub-arrays. NumPy supports advanced indexing for multidimensional arrays.

```
1 print(arr[0])
2 print(arr[1:4])
```

```
1
```

```
[2 3 4]
```

Arithmetic Operations: NumPy supports element-wise arithmetic operations between arrays of the same shape. These operations are often much faster than using Python loops.

```
1 arr1 = np.array([1, 2, 3])
2 arr2 = np.array([4, 5, 6])
3 print(arr1 + arr2)
```

```
[5 7 9]
```

Linear Algebra Operations: NumPy provides functions for linear algebra operations like matrix multiplication, dot product, finding eigenvalues, etc

```
1 # Matrix multiplication (requires 2D arrays)
2 mat1 = np.array([[1, 2], [3, 4]])
3 mat2 = np.array([[5, 6], [7, 8]])
4 product_mat = np.matmul(mat1, mat2)
5 print(product_mat)
```

```
[[19 22]
```

```
[43 50]]
```

3. Broadcasting - A Powerful Feature

- Broadcasting is a mechanism in NumPy that allows performing operations on arrays of different shapes under certain conditions. When operands have different shapes, NumPy expands the smaller array to match the larger one in a specific way to perform element-wise operations.