How can I apply a function to each element of a NumPy array?

Authored by

stats writer

May 11, 2024

One can apply a function to each element of a NumPy array by using the built-in function "apply_along_axis" or by using the "vectorize" function. Both of these functions allow for the application of a user-defined function to each element of a NumPy array, making it easier to perform operations on large arrays efficiently. Additionally, using these functions helps to avoid the need for writing loops, reducing the overall code complexity and increasing its readability.

Map a Function Over a NumPy Array (With Examples)

You can use the following basic syntax to map a function over a NumPy array:

```
#define function
my_function = lambda x: x*5

#map function to every element in NumPy array
my_function(my_array)
```

The following examples show how to use this syntax in practice.

Example 1: Map Function Over 1-Dimensional NumPy Array

The following code shows how to map a function to a NumPy array that multiplies each value by 2 and then adds 5:

```
import numpy as np
```

```
#create NumPy array
data = np.array()

#define function
my_function = lambda x: x*2+5

#apply function to NumPy array
my_function(data)

array()
```

Here is how each value in the new array was calculated:

First value: 1*2+5 = 7Second value: 3*2+5 = 11Third value: 4*2+5 = 13

And so on.

Example 2: Map Function Over Multi-Dimensional NumPy Array

The following code shows how to map a function to a multi-dimensional NumPy array that multiplies each value by 2 and then adds 5:

```
import numpy as np

#create NumPy array
```

data = np.array(,])

#view NumPy array
print(data)

]

#define function
my_function = lambda x: x*2+5

#apply function to NumPy array
my_function(data)

array(,
])

Notice that this syntax worked with a multi-dimensional array just as well as it worked with a one-dimensional array.

The following tutorials explain how to perform other common operations in NumPy: