NumPy prod()

Summary: in this tutorial, you'll learn how to use the numpy `prod()` function to calculate the product of numbers in an array.

Introduction to to the NumPy prod() function

Suppose you have three numbers n, m, and k. The product of the three numbers is nxmxk. For example, the product of 2, 3, and 4 is 2x3x4 = 24.

To calcualte the products of numbers in an aray, you use the numpy `prod()` function:

```
numpy.prod(a, axis=None, dtype=None, out=None, keepdims=<no value>, initial=<no value>, where=<no value>)
```

NumPy prod() function examples

Let's take some example of using the numpy `prod()` function.

1) Using the numpy prod() function with 1-D array example

The following example uses the `prod()` to calculate the products of numbers in a 1-D array:

```
import numpy as np

a = np.arange(1, 5)
result = np.prod(a)

print(a)
print(f'result={result}')
```

Output:

```
[1 2 3 4]
result=24
```

How it works.

First, create an array that has 4 numbers from 1 to 4 using the `arange()` function.

```
a = np.arange(1, 5)
```

Second, calculate the products of all the numbers in the array a:

```
result = np.prod(a)
```

Third, display the numbers of the array a and their product:

```
print(a)
print(f'result={result}')
```

Note that you can pass an array like object to the `prod()` function e.g., a list. For example:

```
import numpy as np

result = np.prod([1, 2, 3, 4, 5])

print(f'result={result}')
```

Output:

```
result=120
```

2) Using the numpy prod() function with multidimensional array examples

The following example uses the `prod()` to calcualte products of all numbers in a 2-D array:

```
import numpy as np

result = np.prod([
    [1, 2],
    [3, 4]
])

print(f'result={result}')
```

Output:

```
result=24
```

To calcualte product of numbers of an axis, you can specify the axis argument. For example, the following uses the `prod()` to calculate the product of numbers on axis 0:

```
import numpy as np

result = np.prod([
    [1, 2],
    [3, 4]
], axis=0)

print(f'result={result}')
```

Output:

```
result=[3 8]
```

Similarly, you can calculate the product of numbers on axis 1:

```
import numpy as np

result = np.prod([
    [1, 2],
    [3, 4]
], axis=1)

print(f'result={result}')
```

Output:

```
result=[ 2 12]
```

3) Selecting numbers to include in the product

To select specific number to include in the product, you use the where argument. For example:

```
import numpy as np

a = np.array([np.nan, 3, 4])
result = np.prod(a, where=[False, True, True])
print(result)
```

Output:

```
12.0
```

In this example, the array contains three elements np.nan, 3, and 4.

The where argument uses a boolean list to specify which element in the array a should be included in the product.

If the value of the `where` list is `True`, the corresponding element of the input array will be included in the product.

4) Special cases

Note that if you pass an array of integers to the `prod()` function that causes an overflow, the `prod()` won't raise the error. For example:

```
import numpy as np

result = np.prod(np.arange(1, 100))
print(f'result={result}')
```

Output:

```
result=0
```

The `prod()` function returns 1 if the array is empty. For example:

```
import numpy as np

result = np.prod(np.array([]))

print(f'result={result}')
```

Output:

```
1.0
```

Summary

- Use numpy `prod()` function to calculate the product of numbers in an array.