NumPy var()

Summary: in this tutorial, you'll learn how to use the `var()` function to calculate the variances of elements in an array.

Introduction to the NumPy var() function

The variance is a measure of the spread of a distribution. To manually calculate the variance of numbers, you follow these steps:

- First, calculate the average of all numbers.
- Second, calculate the squared difference of each number by subtracting it from the mean and square the result.

- Third, calculate the average of those squared differences.

For example, to calculate the variance of three numbers 1, 2, and 3:

First, calculate the average (or mean (https://www.pythontutorial.net/python-numpy/numpy-mean/)):

(1+2+3) / 3 = 2.0

Second, calculate the squared difference of each number with the mean:

$(1-2)^2 + (2-2)^2 + (3-2)^2 = 2$

Third, calculate the average of these squared differences:

2 / 3 ~ 0.667

To calculate the variances of numbers in an array, you can use the `var()` function:

```
numpy.var(a, axis=None, dtype=None, out=None, ddof=0, keepdims=<no value>, *, where=<no value>)
```

For example:

```
import numpy as np


a = np.array([1, 2, 3])
result = np.var(a)
print(round(result,3))
```

Output:

```
0.667
```

Summary

- Use the numpy `var()` function to calculate the variance of elements in an array.