vncoder.vn

Bài 15: Các phép tính toán học

5-6 phút

Các mảng đầu vào để thực hiện các phép toán số học như cộng (), trừ (), nhân () và chia () phải có cùng hình dạng hoặc phải tuân theo quy tắc phát mảng.

Ví dụ :

```
import numpy as np
a = np.arange(9, dtype =
np.float_).reshape(3,3)

print 'First array:'
print a
print '\n'

print 'Second array:'
b = np.array([10,10,10])
print b
print '\n'

print 'Add the two arrays:'
print np.add(a,b)
print '\n'

print 'Subtract the two arrays:'
print np.subtract(a,b)
```

```
print '\n'

print 'Multiply the two arrays:'
print np.multiply(a,b)
print '\n'

print 'Divide the two arrays:'
print np.divide(a,b)
```

Kết quả :

```
First array:
[[ 0. 1. 2.]
 [ 3. 4. 5.]
 [ 6. 7. 8.]]

Second array:
[10 10 10]

Add the two arrays:
[[ 10. 11. 12.]
 [ 13. 14. 15.]
 [ 16. 17. 18.]]

Subtract the two arrays:
[[-10. -9. -8.]
 [ -7. -6. -5.]
 [ -4. -3. -2.]]

Multiply the two arrays:
[[ 0. 10. 20.]
 [ 30. 40. 50.]
 [ 60. 70. 80.]]
```

```
Divide the two arrays:
[[ 0.  0.1 0.2]
 [ 0.3 0.4 0.5]
 [ 0.6 0.7 0.8]]
```

Bây giờ chúng ta hãy thảo luận về một số hàm số học quan trọng khác có sẵn trong NumPy.

1. numpy.reciprocal()

Hàm này trả về nghịch đảo của đối số, theo phần tử. Đối với các phần tử có giá trị tuyệt đối lớn hơn 1, kết quả luôn là 0 do cách Python xử lý phép chia số nguyên. Đối với số nguyên 0, cảnh báo tràn được đưa ra.

Ví dụ :

```
import numpy as np
a = np.array([0.25, 1.33, 1, 0, 100])

print 'Our array is:'
print a
print '\n'

print 'After applying reciprocal function:'
print np.reciprocal(a)
print '\n'

b = np.array([100], dtype = int)
print 'The second array is:'
print b
print '\n'
```

```
print 'After applying reciprocal function:'
print np.reciprocal(b)
```

Kết quả :

```
Our array is:
[   0.25    1.33    1.      0.    100.  ]

After applying reciprocal function:
main.py:9: RuntimeWarning: divide by zero
encountered in reciprocal
  print np.reciprocal(a)
[ 4.         0.7518797  1.                inf
0.01     ]

The second array is:
[100]

After applying reciprocal function:
[0]
```

2. numpy.power()

Hàm này coi các phần tử trong mảng đầu vào đầu tiên là cơ sở và trả về giá trị của nó được nâng lên thành lũy thừa của phần tử tương ứng trong mảng đầu vào thứ hai.

Ví dụ :

```
import numpy as np
a = np.array([10,100,1000])

print 'Our array is:'
print a
print '\n'
```

```
print 'Applying power function:'
print np.power(a,2)
print '\n'

print 'Second array:'
b = np.array([1,2,3])
print b
print '\n'

print 'Applying power function again:'
print np.power(a,b)
```

Kết quả :

```
Our array is:
[  10  100 1000]

Applying power function:
[    100   10000 1000000]

Second array:
[1 2 3]

Applying power function again:
[        10      10000 1000000000]
```

3. numpy.mod()

Hàm này trả về phần còn lại của phép chia của các phần tử tương ứng trong mảng đầu vào. Hàm numpy.remainder () cũng cho kết quả tương tự.

Ví dụ :

```
import numpy as np
a = np.array([10,20,30])
b = np.array([3,5,7])

print 'First array:'
print a
print '\n'

print 'Second array:'
print b
print '\n'

print 'Applying mod() function:'
print np.mod(a,b)
print '\n'

print 'Applying remainder() function:'
print np.remainder(a,b)
```

```
First array:
[10 20 30]

Second array:
[3 5 7]

Applying mod() function:
[1 0 2]

Applying remainder() function:
[1 0 2]
```

Các hàm sau được sử dụng để thực hiện các phép toán trên mảng số phức.

- **numpy.real()** − trả về phần thực của đối số kiểu dữ liệu phức.
- **numpy.imag()** − trả về phần ảo của đối số kiểu dữ liệu phức.
- **numpy.conj()** −trả về liên hợp phức, có được bằng cách thay đổi dấu của phần ảo.
- **numpy.angle()** − trả về góc của đối số phức. Hàm có tham số độ. Nếu đúng, góc tính bằng độ được trả về, ngược lại góc tính bằng radian.

```
import numpy as np
a = np.array([-5.6j, 0.2j, 11. , 1+1j])

print 'Our array is:'
print a
print '\n'

print 'Applying real() function:'
print np.real(a)
print '\n'

print 'Applying imag() function:'
print np.imag(a)
print '\n'

print 'Applying conj() function:'
print np.conj(a)
print '\n'

print 'Applying angle() function:'
print np.angle(a)
print '\n'
```

```
print 'Applying angle() function again (result
in degrees)'
print np.angle(a, deg = True)
```

Kết quả :

```
Our array is:
[ 0.-5.6j 0.+0.2j 11.+0.j 1.+1.j ]

Applying real() function:
[ 0. 0. 11. 1.]

Applying imag() function:
[-5.6 0.2 0. 1. ]

Applying conj() function:
[ 0.+5.6j 0.-0.2j 11.-0.j 1.-1.j ]

Applying angle() function:
[-1.57079633 1.57079633 0. 0.78539816]

Applying angle() function again (result in
degrees)
[-90. 90. 0. 45.]
```