

vncoder.vn

Bài 14: Hàm Toán Học

5-6 phút

Nói một cách dễ hiểu, NumPy chứa một số lượng lớn các phép toán khác nhau. NumPy cung cấp các hàm lượng giác tiêu chuẩn, các hàm cho phép toán số học, xử lý số phức, v.v.

1. Hàm lượng giác :

NumPy có các hàm lượng giác tiêu chuẩn trả về tỷ số lượng giác cho một góc nhất định tính bằng radian.

Ví dụ :

```
import numpy as np
a = np.array([0, 30, 45, 60, 90])

print 'Sine of different angles:'
# Convert to radians by multiplying with pi/180
print np.sin(a*np.pi/180)
print '\n'

print 'Cosine values for angles in array:'
print np.cos(a*np.pi/180)
print '\n'

print 'Tangent values for given angles:'
print np.tan(a*np.pi/180)
```

Kết quả :

Sine of different angles:

```
[ 0.          0.5          0.70710678  0.8660254
 1.          ]
```

Cosine values for angles in array:

```
[ 1.00000000e+00  8.66025404e-01
 7.07106781e-01  5.00000000e-01
 6.12323400e-17]
```

Tangent values for given angles:

```
[ 0.00000000e+00  5.77350269e-01
 1.00000000e+00  1.73205081e+00
 1.63312394e+16]
```

Các hàm arcsin, arcos và arctan trả về nghịch đảo lượng giác của sin, cos và tan của góc đã cho. Kết quả của các hàm này có thể được xác minh bởi hàm `numpy.degrees()` bằng cách chuyển đổi radian sang độ.

Ví dụ :

```
import numpy as np
a = np.array([0, 30, 45, 60, 90])

print 'Array containing sine values:'
sin = np.sin(a*np.pi/180)
print sin
print '\n'

print 'Compute sine inverse of angles. Returned
values are in radians.'
inv = np.arcsin(sin)
```

```
print inv
print '\n'

print 'Check result by converting to degrees:'
print np.degrees(inv)
print '\n'

print 'arccos and arctan functions behave
similarly:'
cos = np.cos(a*np.pi/180)
print cos
print '\n'

print 'Inverse of cos:'
inv = np.arccos(cos)
print inv
print '\n'

print 'In degrees:'
print np.degrees(inv)
print '\n'

print 'Tan function:'
tan = np.tan(a*np.pi/180)
print tan
print '\n'

print 'Inverse of tan:'
inv = np.arctan(tan)
print inv
print '\n'
```

```
print 'In degrees:'  
print np.degrees(inv)
```

Kết quả :

Array containing sine values:

```
[ 0.          0.5          0.70710678  0.8660254  
1.          ]
```

Compute sine inverse of angles. Returned values are in radians.

```
[ 0.          0.52359878  0.78539816  
1.04719755  1.57079633]
```

Check result by converting to degrees:

```
[ 0.  30.  45.  60.  90.]
```

arccos and arctan functions behave similarly:

```
[ 1.00000000e+00  8.66025404e-01  
7.07106781e-01  5.00000000e-01  
6.12323400e-17]
```

Inverse of cos:

```
[ 0.          0.52359878  0.78539816  
1.04719755  1.57079633]
```

In degrees:

```
[ 0.  30.  45.  60.  90.]
```

Tan function:

```
[ 0.00000000e+00  5.77350269e-01  
1.00000000e+00  1.73205081e+00  
1.63312394e+16]
```

Inverse of tan:

```
[ 0.          0.52359878  0.78539816
 1.04719755  1.57079633]
```

In degrees:

```
[ 0.  30.  45.  60.  90.]
```

2. Các hàm để làm tròn

a. `numpy.around()`:

Đây là một hàm trả về giá trị được làm tròn đến độ chính xác mong muốn. Hàm nhận các tham số sau.

1. **a** : Dữ liệu đầu vào
2. **decimals** : Số lượng số thập phân làm tròn đến. Mặc định là 0. Nếu âm, số nguyên được làm tròn đến vị trí bên trái của dấu thập phân

Ví dụ :

```
import numpy as np
a = np.array([1.0, 5.55, 123, 0.567, 25.532])
```

```
print 'Original array:'
print a
print '\n'
```

```
print 'After rounding:'
print np.around(a)
print np.around(a, decimals = 1)
print np.around(a, decimals = -1)
```

Kết quả :

Original array:

```
[ 1.      5.55 123.      0.567 25.532]
```

After rounding:

```
[ 1.      6.    123.      1.    26. ]
```

```
[ 1.      5.6   123.      0.6   25.5]
```

```
[ 0.      10.   120.      0.    30. ]
```

b. numpy.floor() :

Hàm này trả về số nguyên lớn nhất không lớn hơn tham số đầu vào. Tầng của vô hướng x là số nguyên i lớn nhất, sao cho $i \leq x$. Lưu ý rằng trong Python, ván sán luôn được làm tròn từ 0.

Ví dụ :

```
import numpy as np
a = np.array([-1.7, 1.5, -0.2, 0.6, 10])
```

```
print 'The given array:'
```

```
print a
```

```
print '\n'
```

```
print 'The modified array:'
```

```
print np.floor(a)
```

Kết quả :

The given array:

```
[ -1.7   1.5  -0.2   0.6  10. ]
```

The modified array:

```
[ -2.    1.   -1.    0.   10.]
```

c. numpy.ceil()

Hàm ceil () trả về giá trị trần của giá trị đầu vào, tức là ceil của vô hướng x là số nguyên i nhỏ nhất, sao cho $i \geq x$

Ví dụ :

```
import numpy as np
a = np.array([-1.7, 1.5, -0.2, 0.6, 10])

print 'The given array:'
print a
print '\n'

print 'The modified array:'
print np.ceil(a)
```

Kết quả :

```
The given array:
[ -1.7   1.5  -0.2   0.6  10. ]

The modified array:
[ -1.   2.  -0.   1.  10.]
```