vncoder.vn

Bài 19: Copy & View

3-4 phút

Trong khi thực thi các hàm, một số hàm trả về bản sao của mảng đầu vào, trong khi một số trả về dạng xem. Khi nội dung được lưu trữ vật lý ở một vị trí khác, nó được gọi là Sao chép. Mặt khác, nếu một dạng xem khác của cùng một nội dung bộ nhớ được cung cấp, ta gọi nó là Dạng xem.

1. No copy :

Các phép gán đơn giản không tạo ra bản sao của đối tượng mảng. Thay vào đó, nó sử dụng cùng một id () của mảng ban đầu để truy cập nó. Id () trả về một định danh chung của đối tượng Python, tương tự như con trỏ trong C.

Hơn nữa, bất kỳ thay đổi nào trong một trong hai đều được phản ánh trong cái còn lại. Ví dụ: hình dạng thay đổi của một cái cũng sẽ thay đổi hình dạng của cái kia

Ví dụ :

```
import numpy as np
a = np.arange(6)

print 'Our array is:'
print a

print 'Applying id() function:'
```

```
print id(a)

print 'a is assigned to b:'
b = a
print b

print 'b has same id():'
print id(b)

print 'Change shape of b:'
b.shape = 3,2
print b

print 'Shape of a also gets changed:'
print a
```

Kết quả :

```
Our array is:
[0 1 2 3 4 5]

Applying id() function:
139747815479536

a is assigned to b:
[0 1 2 3 4 5]
b has same id():
139747815479536

Change shape of b:
[[0 1]
 [2 3]
 [4 5]]
```

```
Shape of a also gets changed:
[[0 1]
 [2 3]
 [4 5]]
```

2. View hoặc Shallow Copy

NumPy có phương thức ndarray.view () là một đối tượng mảng mới xem xét cùng một dữ liệu của mảng ban đầu. Không giống như trường hợp trước đó, thay đổi kích thước của mảng mới không thay đổi kích thước của mảng ban đầu.

Ví dụ :

```
import numpy as np
# To begin with, a is 3X2 array
a = np.arange(6).reshape(3,2)

print 'Array a:'
print a

print 'Create view of a:'
b = a.view()
print b

print 'id() for both the arrays are different:'
print 'id() of a:'
print id(a)
print 'id() of b:'
print id(b)

# Change the shape of b. It does not change the
shape of a
```

```
b.shape = 2,3

print 'Shape of b:'
print b

print 'Shape of a:'
print a
```

Kết quả :

```
Array a:
[[0 1]
 [2 3]
 [4 5]]

Create view of a:
[[0 1]
 [2 3]
 [4 5]]

id() for both the arrays are different:
id() of a:
140424307227264
id() of b:
140424151696288

Shape of b:
[[0 1 2]
 [3 4 5]]

Shape of a:
[[0 1]
 [2 3]
```

```
 [4 5]]
```

Ví dụ 2 :

```
import numpy as np
a = np.array([[10,10], [2,3], [4,5]])

print 'Our array is:'
print a

print 'Create a slice:'
s = a[:, :2]
print s
```

Kết quả :

```
Our array is:
[[10 10]
 [ 2 3]
 [ 4 5]]

Create a slice:
[[10 10]
 [ 2 3]
 [ 4 5]]
```

3. Deep Copy

Hàm ndarray.copy () tạo một bản sao sâu. Nó là một bản sao hoàn chỉnh của mảng và dữ liệu của nó, và không chia sẻ với mảng ban đầu.

Ví dụ :

```
import numpy as np
a = np.array([[10,10], [2,3], [4,5]])
```

```
print 'Array a is:'
print a

print 'Create a deep copy of a:'
b = a.copy()
print 'Array b is:'
print b

#b does not share any memory of a
print 'Can we write b is a'
print b is a

print 'Change the contents of b:'
b[0,0] = 100

print 'Modified array b:'
print b

print 'a remains unchanged:'
print a
```

Kết quả :

```
Array a is:
[[10 10]
 [ 2 3]
 [ 4 5]]

Create a deep copy of a:
Array b is:
[[10 10]
 [ 2 3]
```

```
 [ 4 5]]
Can we write b is a
False

Change the contents of b:
Modified array b:
[[100 10]
 [ 2 3]
 [ 4 5]]

a remains unchanged:
[[10 10]
 [ 2 3]
 [ 4 5]]
```