NumPy

Introdução

Indexação sofisticada (*fancy indexing*)

Indexação sofisticada

Indexação sofisticada (*fancy indexing*) é o termo adotado pelo *NumPy* para descrever a indexação usando *arrays* de inteiros.

NumPy

Indexação sofisticada

Primeiro vamos importar a biblioteca Numpy. Em seguida criaremos um array 5 x 3 preenchido com zeros. Depois vamos alimentar cada linha do array com valores de 0 a 4.

```
In [21]: import numpy as np

In [22]: arr = np.zeros((5,3))

In [23]: arr
Out[23]:
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])

In [24]: for i in range(5):
    ...:     arr[i] = i
    ...:
    ...:

In [25]: arr
Out[25]:
array([[0., 0., 0.],
       [1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.],
       [4., 4., 4.]])
```

NumPy

Indexação sofisticada

Veja a utilização de uma lista e de um ndarray para indexar um array.

```
IPython: C:Users/evaldo
Out[4]:
array([[0., 0., 0.],
       [1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.],
       [4., 4., 4.]])

In [5]: lista = [4,3,1,0,2]

In [6]: arr[lista]
Out[6]:
array([[4., 4., 4.],
       [3., 3., 3.],
       [1., 1., 1.],
       [0., 0., 0.],
       [2., 2., 2.]])
```

```
In [7]: arr2 = np.array([4,1,3,2,0])

In [8]: arr[arr2]
Out[8]:
array([[4., 4., 4.],
       [1., 1., 1.],
       [3., 3., 3.],
       [2., 2., 2.],
       [0., 0., 0.]])

In [9]:
```

NumPy

Indexação sofisticada

Para selecionar um subconjunto das linhas em uma ordem em particular, podemos passar uma lista ou um ndarray de inteiros especificando a ordem desejada. Nesse exemplo, estamos retornando as linhas 2 e 4 de nosso array.

```
In [25]: arr
Out[25]:
array([[0., 0., 0.],
       [1., 1., 1.],
       [2., 2., 2.],
       [3., 3., 3.],
       [4., 4., 4.]])
```

```
In [32]: arr[[2,4]]
Out[32]:
array([[2., 2., 2.],
       [4., 4., 4.]])
```

NumPy

Indexação sofisticada

É possível utilizar índices negativos, onde -1 é a última linha.

```
In [33]: arr[[-1, -2]]
Out[33]:
array([[4., 4., 4.],
       [3., 3., 3.]])
```

NumPy

Indexação sofisticada

Vamos criar agora um novo array 5x3 contendo um sequencial de 0 a 14.

```
In [41]: arr = np.arange(15).reshape(5, 3)

In [42]: arr
Out[42]:
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11],
       [12, 13, 14]])
```

arange: é uma versão da função range de Python com valor de array.

O array vai possuir 15 elementos com valores de 0 a 14.

O método reshape altera o "shape" do array. Passamos uma tupla definindo o valor de cada dimensão. Estamos defindo aqui que "arr" será um array 5 x 3.

NumPy

Indexação sofisticada

Ao passarmos vários índices de array, teremos como resultado um array unidimensional de elementos correspondentes a cada tupla de índices. Veja um exemplo.

```
In [43]: arr[[0, 2, 4], [2, 1, 0]]
Out[43]: array([ 2,  7, 12])
```

Neste exemplo estamos retornando os elementos (0, 2), (2, 1) e (4, 0).

NumPy

Indexação sofisticada

Agora veja como obter uma região retangular formada pela seleção de um subconjunto das linhas e colunas da matriz.

```
In [44]: arr[[1, 2, 3]][:,[2, 0, 1]]
Out[44]:
array([[ 5,  3,  4],
       [ 8,  6,  7],
       [11,  9, 10]])
```

[[1,2,3]] -> Estamos selecionando as linhas 1, 2 e 3.

[[1,2,3]][:] -> Usando fatia para retornar todas as linhas.

[[1,2,3]][:, [2,0,1]] -> Mudando a ordem das colunas.

FIM