大数据基础与应用
BIG DATA FOUNDATION AND APPLICATION

多维数组ndarray

北京理工大学计算机学院　高玉金

2019年3月

数组对象ndarray

- 多维数组ndarray是一个在NumPy库中定义的通用多维同类型数据容器
- Ndarray的出现是为了弥补Python自身没有类C数组存在的问题，提高数据容器的操作效率
- ndarray要求所有元素的数据类型必须一致
- Numpy会自动识别ndarray中的数据类型，如果数据类型不一致Numpy会将所有元素自动转换成一个合适的数据类型
- 可以为ndarray指定类型

```
In [33]: arr = np.array([2,5.0,"6"])

In [34]: arr
Out[34]: array(['2', '5.0', '6'], dtype='<U32')
```

```
arr2=np.array([2,5.0,"6"],dtype=np.float64)

In [36]: arr2
Out[36]: array([2., 5., 6.])
```

生成ndarray数组对象

- array()函数可以接收任意的序列型对象，如：np.array([1,2,3,4,5])和 np.array([[1,2],[3,4],[5,6]])
- 常用生成函数
  - zeros()函数
  - ones()函数
  - arange()函数：内置range()函数的数组版
  - empty()函数：创建没有初始化数值的数组
- 生成高维数组
  - 直接给生成函数传递一个合适的元组
  - 低维数组进行reshape操作

```
>>> np.zeros(5)
array([ 0.,  0.,  0.,  0.,  0.])
>>> np.zeros((3,5))
array([[ 0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.]])
```

arr = np.zeros(15).reshape((3,5))

ndarray的基本属性

- ndim属性：表示数组的维度
- shape属性：表示每一个维度的数量（元组）
- dtype属性：描述数组元素的数据类型
- Itemsize属性：表示单个元素的大小（字节）
- size属性：表示数据量的大小

```
In [47]: np.arange(10).shape
Out[47]: (10,)
```

arr = np.zeros(15).reshape((3,5))

```
: arr.ndim
: 2

: arr.size
: 15

: arr.dtype
: dtype('float64')

: arr.itemsize
: 8

: arr.shape
: (3, 5)
```

ndarray的基本方法

- reshape(shape)，返回一个新数组，原数组不变
- flatten()，降为一维数组，原数组不变
- rezise(shape)，修改原数组
- tolist()，将数组转换为列表

```
In [56]: arr.resize((5,3))

In [57]: arr
Out[57]:
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

```
In [58]: arr.tolist()
Out[58]:
[[0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0]]
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

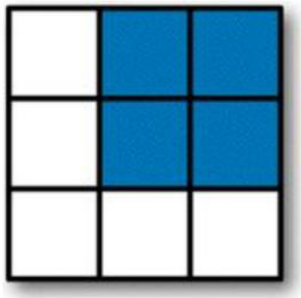
数组的索引和切片

- 一维数组索引，如arr[3]，获取特定位置的元素
- 二维书索引，如arr2d[0][2]或arr2d[0,2]
- 多维数组索引，如arr3d[1,2,3]
- 各维度分别切片arr[m:n:k]
- 对多维数组的切片赋值时，整个切片都会被重新赋值

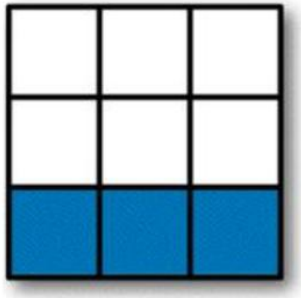
```
>>> arr = np.arange(10)
>>> arr[5:8]
array([5, 6, 7])
>>> arr[5:8] = 10
>>> arr
array([ 0,  1,  2,  3,  4, 10, 10, 10,  8,  9])
```

数组的切片

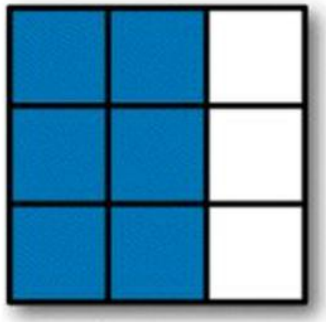
- 不同于Python的列表，数组的切片是原数组的视图而非拷贝，任何对视图的修改都会反映到原数组上（原因：大数据的效率）
- 需要拷贝时，应显式复制数组，如arr[5:8].copy()
- 分析数组的切片：arr=np.arange(9).reshape((3,3)) **行列切片重叠区**



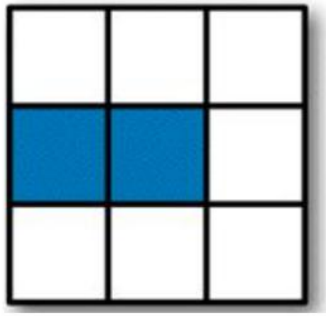
arr[:2, 1:]



arr[2]
arr[2, :]
arr[2:, :]



arr[:, :2]



arr[1, :2]
arr[1:2, :2]

多维数组切片示例

- ar2d[::-2,::-2]

ar2d=np.arange(20).reshape((4,5))

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])
```

```
array([[19, 17, 15],
       [ 9,  7,  5]])
```

- 请大家找出：ar2d[:2:-2, 1::-2]