

Miller's ihome

One number,one world ! One heart , one world !

numpy中的ndarray方法和属性

发表于六月 3, 2013

NumPy数组的维数称为秩（rank），一维数组的秩为1，二维数组的秩为2，以此类推。在NumPy中，每一个线性的数组称为是一个轴（axes），秩其实是描述轴的数量。比如说，二维数组相当于是一个一维数组，而这个一维数组中每个元素又是一个一维数组。所以这个一维数组就是NumPy中的轴（axes），而轴的数量——秩，就是数组的维数。

Numpy库中的矩阵模块为ndarray对象，有很多属性：T, data, dtype, flags, flat, imag, real, size,

itemsize, nbytes, ndim, shape, strides, ctypes, base等等。

```
>>> import numpy as np
>>> x=np.array([[1,2,3],[9,8,7],[6,5,4]])
>>> x.T      #获得x的转置矩阵
array([[1, 9, 6],
       [2, 8, 5],
       [3, 7, 4]])
>>> print x.flags      #返回数组内部的信息
C_CONTIGUOUS : True
F_CONTIGUOUS : False
OWNDATA : True
WRITEABLE : True
ALIGNED : True
UPDATEIFCOPY : False
>>> x.flat[2:6]      #将数组变为1维数组，并获取其中的一部分数据
array([3, 9, 8, 7])
>>> x.flat = 4; x      #将值赋给1维数组，再转化成有原有数组的大小形式
array([[4, 4, 4],
       [4, 4, 4],
       [4, 4, 4]])
>>> x
array([[4, 4, 4],
       [4, 4, 4],
       [4, 4, 4]])
```

ndarray.imag # 为复变函数中含有虚部的数组，如下：

```
>>> x = np.sqrt([2+3j, 5+0j])      # 创建一个复数
>>> x
array([ 1.67414923+0.89597748j,  2.23606798+0.j          ])
>>> x.imag      #获得复数的虚部
array([ 0.89597748,  0.          ])
>>> x.real      #获得复数的实部
array([ 1.67414923,  2.23606798])
>>> x=np.arange(10)      #随机生成一个数组，并重新命名一个空间的数组
>>> x.reshape(2,5)
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
>>> x.size      #获得数组中元素的个数
10
>>> x.ndim      #获得数组的维数
2
>>> x.shape      #获得数组的（行数，列数）
(2, 5)
```

```
>>> y=x.reshape(5,2)
>>> y
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7],
       [8, 9]])
>>> y.base      #获得该数组基于另外一个对象数组而来，如下，y是根据x而来
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Ndarray对象的方法

ndarray.ptp(axis=None, out=None) : 返回数组的最大值—最小值或者某轴的最大值—最小值

ndarray.clip(a_min, a_max, out=None) : 小于最小值的元素赋值为最小值，大于最大值的元素变为最大值。

ndarray.all() : 如果所有元素都为真，那么返回真；否则返回假

ndarray.any() : 只要有一个元素为真则返回真

ndarray.swapaxes(axis1, axis2) : 交换两个轴的元素，如下

```
>>> z.swapaxes(0,1)
array([[2, 4, 6, 8],
       [3, 5, 7, 9]])
```

■ 下面为改变数组维度和大小的方法：

ndarray.reshape(shape[, order]) : 返回重命名数组大小后的数组，不改变元素个数。

ndarray.resize(new_shape[, refcheck]) : 改变数组的大小（可以改变数组中元素个数）。

ndarray.transpose(*axes) : 返回矩阵的转置矩阵

ndarray.swapaxes(axis1, axis2) : 交换两个轴的元素后的矩阵。

ndarray.flatten([order]) : 复制一个一维的array出来。

ndarray.ravel([order]) : 返回为展平后的一维数组。

ndarray.squeeze([axis]) : 移除长度为1的轴。

ndarray.tolist() : 将数组转化为列表

ndarray.take(indices, axis=None, out=None, mode='raise') : 获得数组的指定索引的数据，如：

```
>>> a=np.arange(12).reshape(3,4)
>>> a
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
>>> a.take([1,3],axis=1) #提取1, 3列的数据
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11]])
```

numpy.put(a, ind, v, mode='raise') : 用v的值替换数组a中的ind（索引）的值。Mode可以为raise/wrap/clip。Clip：如果给定的ind超过了数组的大小，那么替换最后一个元素。

numpy.repeat(a, repeats, axis=None) : 重复数组的元素，如：

```
>>> x = np.array([[1,2],[3,4]])
>>> np.repeat(x, 2)
array([1, 1, 2, 2, 3, 3, 4, 4])
>>> np.repeat(x, 3, axis=1)
array([[1, 1, 1, 2, 2, 2],
       [3, 3, 3, 4, 4, 4]])
>>> np.repeat(x, [1, 2], axis=0)
array([[1, 2],
       [3, 4],
       [3, 4]])
```

numpy.tile(A, reps) : 根据给定的reps重复数组A，和repeat不同，repeat是重复元素，该方法是重复数组。

ndarray.var(axis=None, dtype=None, out=None, ddof=0) : 返回数组的方差，沿指定的轴。

ndarray.std(axis=None, dtype=None, out=None, ddof=0) : 沿给定的轴返回数则的标准差

ndarray.prod(axis=None, dtype=None, out=None) : 返回指定轴的所有元素乘机

ndarray.cumprod(axis=None, dtype=None, out=None) : 返回指定轴的累积和 如下：

`ndarray.cumprod(axis=None, dtype=None, out=None)` : 返回指定轴的累积积, 如下:

```
>>> a
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
>>> a.cumprod(axis=1) #得到竖轴的累积
array([[ 0,  0,  0,  0],
       [ 4, 20, 120, 840],
       [ 8, 72, 720, 7920]])
```

`ndarray.mean(axis=None, dtype=None, out=None)` : 返回指定轴的数组元素均值

`ndarray.cumsum(axis=None, dtype=None, out=None)` : 返回指定轴的元素累计和。如:

```
>>> a
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
>>> a.cumsum(axis=1)
array([[ 0,  1,  3,  6],
       [ 4,  9, 15, 22],
       [ 8, 17, 27, 38]])
```

`ndarray.sum(axis=None, dtype=None, out=None)` : 返回指定轴所有元素的和

`ndarray.trace(offset=0, axis1=0, axis2=1, dtype=None, out=None)` : 返回沿对角线的数组元素之和

`ndarray.round(decimals=0, out=None)` : 将数组中的元素按指定的精度进行四舍五入, 如下:

```
>>> np.around([0.37, 1.64])
array([ 0.,  2.])
>>> np.around([0.37, 1.64], decimals=1)
array([ 0.4,  1.6])
>>> np.around([.5, 1.5, 2.5, 3.5, 4.5]) # rounds to nearest even value
array([ 0.,  2.,  2.,  4.,  4.])
>>> np.around([1,2,3,11], decimals=1) # ndarray of ints is returned
array([ 1,  2,  3, 11])
>>> np.around([1,2,3,11], decimals=-1)
array([ 0,  0,  0, 10])
```

`ndarray.conj()` : 返回所有复数元素的共轭复数, 如:

```
>>> b=np.array([[1+2j,3+0j],[3+4j,7+5j]])
>>> b
array([[ 1.+2.j,  3.+0.j],
       [ 3.+4.j,  7.+5.j]])
>>> b.conj()
array([[ 1.-2.j,  3.-0.j],
       [ 3.-4.j,  7.-5.j]])
```

`ndarray.argmin(axis=None, out=None)`: 返回指定轴最小元素的索引。

`ndarray.min(axis=None, out=None)` : 返回指定轴的最小值

`ndarray.argmax(axis=None, out=None)` : 返回指定轴的最大元素索引值

`ndarray.diagonal(offset=0, axis1=0, axis2=1)` : 返回对角线的所有元素。

`ndarray.compress(condition, axis=None, out=None)` : 返回指定轴上条件下的切片。

`ndarray.nonzero()` : 返回非零元素的索引

0

此条目由Miller Wu发表在Python分类目录, 并贴了array、ndarray、numpy标签。将[固定链接](http://www.idataskys.com/numpy%E4%B8%AD%E7%9A%84ndarray%E6%96%B9%E6%B3%95%E5%92%8C%E5%B1%9E%E6%80%A7/)

[<http://www.idataskys.com/numpy%E4%B8%AD%E7%9A%84ndarray%E6%96%B9%E6%B3%95%E5%92%8C%E5%B1%9E%E6%80%A7/>] 加入收藏夹。