

用python做数据分析2|numpy库下进行数组计算的10点tips



是蓝先生 (/u/001f10fcffb8) [+ 关注](#)

2016.04.28 16:30* 字数 857 阅读 942 评论 0 喜欢 0

(/u/001f10fcffb8)

今天是4.28号。

天气又开始变阴，雨持续了一整个月。

之前学习了用python写爬虫和python与mongodb数据库、文件之间的简单操作。想想自己还是想用python去做数据分析，或者能够在证券量化工作中用到相应的知识，所以接下来会主要学习数据分析相关的库numpy,pandas,matplotlib等，至于机器学习的库scikit-learn则等掌握了前三个库的使用后再根据需要进行学习。

```
import numpy as np
```

可以认为numpy就是一个多维的列表(list)：

```
例如
a=[[1, 2, 3],
   [3, 4, 5],
   ]
a.shape能够得到a是n×n维数组；
a.dtype能够得到数组中的元素类型是什么；同时注意ndarray中所有元素的类型必须是一致的。
取数组a中特定位置的值也是用索引，一维直接用a[x]，二维用a[x,y]即第x行第y列的数据。
```

(1) arange函数在numpy中用于创建等差数组，使用频率非常高。arange非常类似range函数，在没有学习numpy时在for循环中，几乎都用到了range，二者的区别仅仅在于是arange返回的是一个arr数组，而range返回的是list。

```
range()函数的用法：
range(1, 5) #代表从1到5(不包含5)
[1, 2, 3, 4]
range(1, 5, 2) #代表从1到5，间隔2(不包含5)
[1, 3]
range(5) #代表从0到5(不包含5)
[0, 1, 2, 3, 4]
```

(2) 花式索引，即利用整数数组进行索引。

```
例如
arr=[[1,1,2,3],
      [1,4,5,6],
      [3,2,5,0],
      [2,3,1,9]]
那么arr[[1,3]]得到的结果为arr([[1,4,5,6],[2,3,1,9]])
```

(3) 进行矩阵计算时常用到转置(T)和内积：

arr.T，可以直接表示转置

np.dot()函数可以计算内积，如np.dot(a,a.T)

(4) arr是一个数组，有一些函数可以快速对数组中所有的元素进行相应的计算操作：

用法例子为：np.abs(arr),np.sqrt(arr)

函数	说明
abs、fabs	计算整数、浮点数或复数的绝对值。对于非复数值，可以使用更快的fabs
sqrt	计算各元素的平方根。相当于arr ** 0.5
square	计算各元素的平方。相当于arr ** 2
exp	计算各元素的指数e ^x
log、log10、log2、log1p	分别为自然对数（底数为e）、底数为10的log、底数为2的log、log(1 + x)
sign	计算各元素的正负号：1（正数）、0（零）、-1（负数）
ceil	计算各元素的ceiling值，即大于等于该值的最小整数
floor	计算各元素的floor值，即小于等于该值的最大整数
rint	将各元素值四舍五入到最接近的整数，保留dtype

1.PNG

对两个数组的操作为：add(x,y),multiply(x,y)等，执行方法是均是对应位置元素进行计算

。



函数	说明
add	将数组中对应的元素相加
subtract	从第一个数组中减去第二个数组中的元素
multiply	数组元素相乘
divide、floor_divide	除法或向下圆整除法（丢弃余数）
power	对第一个数组中的元素A，根据第二个数组中的相应元素B，计算 A^B
maximum、fmax	元素级的最大值计算。fmax将忽略NaN
minimum、fmin	元素级的最小值计算。fmin将忽略NaN

2.png

(5)条件方法 **np.where(cond,x,y)**三元表达式的使用，其中cond表示条件;x、y可以是单独的数字，可以是数组。在数据分析中，它常用于根据一个数组产生新的数组。

例如假设现在有两个数值数组和一个布尔型值数组：

```
xarr=np.array([2,2,3,5,6])
yarr=np.array([5,6,9,0,1])
con=np.array([False,True,True,False])
```

现在想根据con的值选取xarr、yarr中的值，当con中值为True时选xarr的值，否则选y的值。

用一般的方法为：

```
result=[(x if c else y) for x,y,c in zip(xarr,yarr,con)]
```

使用np.where()则能极大提高计算效率，用法为：

```
result=np.where(con,xarr,yarr)
```

(6) 数学和统计方法

可通过sum,mean,std等方法对对整个数组或某个轴上的数据进行统计计算，通常叫做约简（reduction）。

对于axis不懂的可以直接这样理解，axis=0或没有即默认为零，就竖着看；axis=1就横着看。

若x为一个数组使用方法为：

```
x.sum(), x.mean()
```

x.sum(axis=0或1)，其中axis=0 表示按行将每一列的向量相加，axis=1表示按列将每一行的向量相加。

sum也可以用来对布尔型数组中的True值计数：

```
例如：arr=randn(100)
(arr>0).sum()
```



方法	说明
sum	对数组中全部或某轴向的元素求和。零长度的数组的sum为0
mean	算术平均数。零长度的数组的mean为NaN
std、var	分别为标准差和方差，自由度可调（默认为n）
min、max	最大值和最小值
argmin、argmax	分别为最大和最小元素的索引

3.1.PNG

方法	说明
cumsum	所有元素的累计和
cumprod	所有元素的累计积

3.2.png

(7) sort()排序

若x为一个数组，x.sort()默认将数组x中的元素从小到大排序。若x是多维数组，x.sort()会默认按从小到大对每一列进行排序；x.sort(1)则会从小到大对每一行进行排序。

(8) 数组的集合运算

np.unique()与python中set()的功能类似。

例若a=np.array([1,2,3,2,3,4,6])

则np.unique(a)结果为array([1,2,3,4,6])，np.unique()在去除重复元素的同时将元素从小到大进行了排序。

其他集合运算如下，注意集合运算的结果均是数组

方法	说明
unique(x)	计算x中的唯一元素，并返回有序结果
intersect1d(x, y)	计算x和y中的公共元素，并返回有序结果
union1d(x, y)	计算x和y的并集，并返回有序结果
in1d(x, y)	得到一个表示“x的元素是否包含于y”的布尔型数组
setdiff1d(x, y)	集合的差，即元素在x中且不在y中
setxor1d(x, y)	集合的对称差，即存在于一个数组中但不同时存在于两个数组中的元素 ^{译注2}

4.png

**(9) 线性代数的相关计算**

这里暂时不作介绍，后面有需要用到时具体学习并补充进来

(10) 随机数生成

numpy.random模块对python中内置的random进行了补充。使用随机数生成函数时一般要加size=()的参数。

例如使用的方法为：

```
a). numpy.random.normal(size=(4,4))可以得到4x4维服从正太分布的数组，
其中size是normal()函数的一个参数，random的其他函数也有该参数，若该参数缺失，则默认只生成一个值。

b). randint(low,high,size=xx)
    表示从low到high范围内随机生成size个整数，且high值取不到。
    注意：当high缺失或声明high=None时，将唯一的一个数作为上限。
    例如：a=np.random.randint(3,size=10)和a=np.random.randint(3,high=None,size=(2,3))
    二者表达的意思是一样的。

c). random.randn(n,n)可以生成nxn维的服从正太分布的随机数据数组。
```

部分numpy.random函数如下：

函数	说明
seed	确定随机数生成器的种子
permutation	返回一个序列的随机排列或返回一个随机排列的范围
shuffle	对一个序列就地随机排列
rand	产生均匀分布的样本值
randint	从给定的上下限范围内随机选取整数
randn	产生正态分布（平均值为0，标准差为1）的样本值，类似于MATLAB接口
binomial	产生二项分布的样本值
normal	产生正态（高斯）分布的样本值
beta	产生Beta分布的样本值

5.png

最后再说一下reshape()函数可以把一维数组转化为自己想要的多维数组，如下：

np.random.randint(8,size=9).reshape(3,3)





是蓝先生 (/u/001f10fcffb8)

写了 109927 字，被 378 人关注，获得了 340 个喜欢 (/u/001f10fcffb8)

+ 关注

愿我长命百岁，yu你终老

如果写的对你有一点点帮助，可以手动关注一下下，多谢(*^__^*)

赞赏支持

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button)



更多分享

被以下专题收入，发现更多相似内容



生活不易 我用... (/c/8c01bfa7b98a?utm_source=desktop&utm_medium=notes-included-collection)

(http://cwb.assets.jianshu.io/note

/images/3744857

/weibo



Python随手查 (/c/d0242c1e1e81?utm_source=desktop&utm_medium=notes-included-collection)

/image_b3d8f7d3510a.jpg)

