# NumPy入门培训

讲师:林应

微博:<u>http://weibo.com/u/2607195824</u>

最后更新:2016/09/02

# NumPy简介

- 官网链接: <a href="http://www.numpy.org/">http://www.numpy.org/</a>
- NumPy是Python语言的一个扩充程序库。支持高级大量的维度数组与矩阵运算,此外也针对数组运算提供大量的数学函数库。
- 作者介绍
  - Jim Hugunin: <a href="http://www.linkedin.com/in/jimhugunin">http://www.linkedin.com/in/jimhugunin</a>
  - Travis Oliphant: <a href="http://www.linkedin.com/in/teoliphant">http://www.linkedin.com/in/teoliphant</a>

### 基本功能

- 快速高效的<mark>多维数组对象ndarray</mark>
- 用于对数组执行元素级计算以及直接对数组执行数学运算的函数
- 用于读写硬盘上基于数组的数据集的工具
- 线性代数运算、傅里叶变换,以及随机数生成
- 用于将C、C++、Fortran代码集成到Python的工具
- ·除了为Python提供快速的数组处理能力,NumPy在数据分析方面还有另外一个主要作用,即作为在算法之间传递数据的容器。

## 效率对比

- 三种数据结构: list / array / numpy.array
- 三种方法求和: for / sum / numpy.sum
- 例<del>了</del>代码: extra/perf\_compare.py

# NumPy的ndarray 创建ndarray

#### • 数组创建函数

类型	说明
array	将输入数据(列表、元组、数组或其它序列类型)转换为ndarray。要么推断出dtype, 要么显示指定dtype。默认直接复制输入数据。
asarray	将输入转换为darray,如果输入本身就是一个ndarray就不进行复制。
arange	类似于内置的range,但返回一个ndarray而不是列表。
ones, ones_like	根据指定形状和dtype创建一个全1数组。ones_like以另一个数组为参数,并根据其形状和dtype创建一个全1数组。
zeros, zeros_like	类似于ones和ones_like,只不过产生的是全0数组而已。
empty, empty_like	创建数组,只分配内存空间但不填充任何值。
eye, identity	创建一个正方的N * N单位矩阵

• 例子代码: the numpy ndarray/creating ndarray.py

# NumPy的ndarray NumPy数据类型

#### • NumPy数据类型 I

类型	说明
int8, uint8 - i1, u1	有 / 无符号的8位整型
int16, uint16 - i2, u2	有 / 无符号的16位整型
int32, uint32 - i4, u4	有 / 无符号的32位整型
int64, uint64 - i8, u8	有 / 无符号的64位整型
float16 - f2	半精度浮点数
float32 - f4 or f	标准的单精度浮点数,与C的float兼容。
float64 - f8 or d	标准的双精度浮点数。与C的double和Python的float兼容。
float128 - f16 or g	扩展精度浮点数

# NumPy的ndarray NumPy数据类型

• NumPy数据类型 II

类型	说明
complex64/128/256 - c8/16/32	分别用两个32位,64位或128位浮点数表示的复数。
bool - ?	存储True和False值的布尔类型
object - O	Python对象类型
string S	固定长度的字符串类型。S10代表长度为10的字符串。
unicode U	固定长度的unicode类型

- 创建ndarray时指定dtype类型
- 使用astype显示转换类型
- 例<del>了</del>代码: the\_numpy\_ndarray/creating\_ndarray.py

# NumPy的ndarray 数组和标量之间的运算

- 不用编写循环即可对数据执行批量运算
- 大小相等的数组之间的任何算术运算都会将运算应用到元素级
- 数组与标量的算术运算也会将那个标量值传播到各个元素
- 例<del>了</del>代码:

the\_numpy\_ndarray/operations\_between\_arrays\_and\_scalars.py

# NumPy的ndarray 基本的索引和切片

- 索引原理
- 切片原理
- 例子代码: the\_numpy\_ndarray/basic\_indexing\_and\_slicing.py

# NumPy的ndarray 布尔型索引

- 布尔型数组的长度必须跟被索引的轴长度一致。
- 可以将布尔型数组跟切片、整数(或整数序列)混合使用
- 例子代码: the\_numpy\_ndarray/boolean\_indexing.py

# NumPy的ndarray 花式索引

- 花式索引(Fancy indexing)是一个NumPy术语,它指的是<mark>利用整数数组</mark>进 行索引。
- 一次传入多个索引数组会有一点特别。它返回的是一个一维数组,其中的元素 对应各个索引元组。
- 例子代码: the\_numpy\_ndarray/fancy\_indexing.py

# NumPy的ndarray 数组转置和轴对换

- 一维 / 二维数组转置
- 高维数组轴对换
- 例子代码:

the\_numpy\_ndarray/transposing\_arrays\_and\_swapping\_axes.py

#### • — 元函数 I

类型	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
sqare	计算各元素的平方。相当于arr ** 2
exp	计算各元素的e^x
log, log10, log2, log1p	分别为自然对数、底数为10的log、底数为2的log和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小整数。

#### • 一元函数 II

类型	说明
rint	将各元素值四舍五入到最接近的整数,保留dtype。
modf	将数组的小数部分与整数部分以两个独立数组的形式返还。
isnan	返回一个表示"哪些值是NaN(这不是一个数字)"的布尔型数组
isfinite, isinf	分别返回一个表示"哪些元素是有限的(非inf,非NaN)"或"哪些元素是 无穷的"的布尔型数组
cos, cosh, sin, sinh, tan, tanh	普通型或双曲型三角函数
arccos, arccosh, arcsin, arcsinh, arctan, arctanh	反三角函数
logical_not	计算各元素not x的真值。相当于-arr。

#### • 二元函数 I

类型	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	除法或向下取整除法
power	对第一个数组中的元素A和第二个数组中对应位置的元素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。
mod	元素级的求模计算

• 二元函数 ||

类型	说明
copysign	将第二个数组中的符号复制给第一个数组中的值
greater, greater_equal, less, less_equal,equal, not_equal	执行元素级的比较,最终产生布尔型数组。
logical_and, logical_or, logical_xor	执行元素级的真值逻辑运算,最终产生布尔型数组。

• 例子代码: universal\_functions.py

### 利用数组进行数据处理 简介

- NumPy数组使你可以将许多种数据处理任务表述为简洁的数组表达式(否则需要编写循环)。用数组表达式代替循环的做法,通常被称为矢量化。
- <del>矢量化数组运算</del>要比等价的纯Python方式快上<mark>一两个数量级</mark>
- 例子代码: data\_processing\_using\_arrays/intro.py

### 利用数组进行数据处理 将条件逻辑表述为数组运算

- 列表推导的局限性
  - 纯Python代码,速度不够快。
  - 无法应用于高维数组
- where和where的嵌套
- 例子代码:

data\_processing\_using\_arrays/expressing\_conditional\_logic\_as\_array\_op erations.py

## 利用数组进行数据处理 数学和统计方法

#### • 数学和统计方法

类型	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	分别为最大值和最小值的索引
cumsum	所有元素的累计和
cumprod	所有元素的累计积

### 利用数组进行数据处理 数学和统计方法

- 标准差和方差的解释
- cumsum和cumprod的解释
- 带axis参数的统计函数
- 例子代码:

data\_processing\_using\_arrays/mathematical\_and\_statistical\_methods.py

### 利用数组进行数据处理 用于布尔型数组的方法

- sum对True值计数
- any和all测试布尔型数组,对于非布尔型数组,所有非0元素将会被当做True。
- 例子代码:

data\_processing\_using\_arrays/methods\_for\_boolean\_arrays.py

### 利用数组进行数据处理 排序

- 直接排序
- 指定轴排序
- 例<del>了</del>代码:data\_processing\_using\_arrays/sorting.py

### 利用数组进行数据处理 去重以及其它集合运算

• 去重以及其它集合运算

类型	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)	得到一个表述"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)	集合的异或,即存在于一个数组中但不同时存在于两个数组中的元素。

• 例子代码: data\_processing\_using\_arrays/unique\_and\_other\_set\_logic.py

### 数组文件的输入输出

- 将数组以二进制格式保存到磁盘
- 存取文本文件
- 例子代码
  - file\_input\_and\_output\_with\_arrays/saving\_and\_loading\_text\_files.py
  - file\_input\_and\_output\_with\_arrays/storing\_arrays\_on\_disk\_in\_binary\_format.py

# 线性代数

• 常用的numpy.linalg函数 I

类型	<b>说明</b>
diag	以一维数组的形式返回方阵的对角线(或非对角线元素),获将一维数组转换为方阵(非对角线元素为0)。
dot	矩阵乘法
trace	计算对角线元素的和
det	计算矩阵行列式
eig	计算方阵的特征值和特征向量
inv	计算方阵的逆

## 线性代数

• 常用的numpy.linalg函数 II

类型	说明
pinv	计算矩阵的Moore-Penrose伪逆
qr	计算QR分解
svd	计算奇异值分解
solve	解线性方程Ax = b, 其中A为一个方阵。
Istsq	计算Ax = b的最小二乘解

• 例子代码: linear\_algebra.py

# 随机数生成

• 部分numpy.random函数 I

类型	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返回
shuffle	对一个序列就地随机乱序
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)
binomial	产生二项分布的样本值

## 随机数生成

• 部分numpy.random函数 II

类型	说明
normal	产生正态(高斯)分布的样本值
beta	产生Beta分布的样本值
chisquare	产生卡方分布的样本值
gamma	产Gamma分布的样本值
uniform	产生在[0, 1]中均匀分布的样本值

• 例子代码: random\_number\_generation.py

### 高级应用 数组重塑

- reshape重塑数组
- -1自动推导维度大小
- 例<del>了</del>代码:advanced\_array\_manipulation/reshaping\_arrays.py

## 高级应用 数组的合并和拆分

#### • 数组连接函数

类型	说明
concatenate	最一般化的连接,沿一条轴连接一组数组
vstack, row_stack	以面向行的方式对数组进行堆叠(沿轴0)
hstack,	以面向行的方式对数组进行堆叠(沿轴1)
column_stack	类似于hstack,但是会先将一维数组转换为二维列向量。
dstack	以面向"深度"的方式对数组进行堆叠(沿轴2)
split	沿指定轴在指定的位置拆分数组
hsplit, vsplit, dsplit	split的便捷化函数,分别沿着轴0、轴1和轴2进行拆分。

### 高级应用 数组的合并和拆分

- \_r对象
- \_c对象
- 例<del>了</del>代码:

advanced\_array\_manipulation/concatenating\_and\_splitting\_arrays.py

### 高级应用 元素的重复操作

- \_tile
- \_repeat
- 例<del>了</del>代码:advanced\_array\_manipulation/repeating\_elements.py

### 高级应用 花式索引的等价函数

take

• put

• 例子代码: advanced\_array\_manipulation/fancy\_indexing\_equivalents.py

给定m × n阶矩阵X,满足X =  $[x_1, x_2, ... x_n]$ ,这里第i列向量是m维向量。

求n × n矩阵 , 使得D<sub>ij</sub> = ||x<sub>i</sub> - x<sub>j</sub>||<sup>2</sup>

- 方法1:标准方法计算D<sub>ij</sub>
  - D[i, j] = numpy.linalg.norm(X[:, i], X[:, j) \*\* 2
- 方法2:利用dot计算D<sub>ij</sub>
  - d = X[:, i] X[:, j]
  - D[i, j] = numpy.dot(d, d)

- 方法3:减少dot调用次数
  - $D_{ij} = (x_i x_j)^T(x_i x_j) = x_i^Tx_i 2x_i^Tx_j + x_j^Tx_j$
  - G = numpy.dot(X.T, X)
  - $D_{ij} = G_{ii} 2G_{ij} + G_{jj}$

- 方法4:利用重复操作替代外部循环
  - 在方法3的基础上,将D表达为H+K-2G
  - $H_{ij} = G_{ii}$ ,  $K_{ij} = G_{jj}$
  - H = numpy.title(np.diag(G), (n, 1))
  - K = H<sup>T</sup>
  - $D = H + H^{T} 2G$
- 例<del>了</del>代码: extra/dist\_matrix.py