



Python—NumPy类库使用培训课程

讲师 郭剑峰

NumPy概述

NumPy(Numerical Python) 是 Python 语言的一个扩展程序库，支持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。

NumPy 是一个运行速度非常快的数学库，主要用于数组计算，包含：

- 1.一个强大的N维数组对象 ndarray
- 2.广播功能函数
- 3.整合 C/C++/Fortran 代码的工具
- 4.线性代数、傅里叶变换、随机数生成等功能

NumPy 应用

NumPy 通常与 SciPy (Scientific Python) 和 Matplotlib (绘图库) 一起使用, 这种组合广泛用于替代 MatLab, 是一个强大的科学计算环境, 有助于我们通过 Python 学习数据科学或者机器学习。

SciPy 是一个开源的 Python 算法库和数学工具包。(今后学习)

SciPy 包含的模块有最优化、线性代数、积分、插值、特殊函数、快速傅里叶变换、信号处理和图像处理、常微分方程求解和其他科学与工程中常用的计算。

Matplotlib 是 Python 编程语言及其数值数学扩展包 NumPy 的可视化操作界面。它为利用通用的图形用户界面工具包, 如 Tkinter, wxPython, Qt 或 GTK+ 向应用程序嵌入式绘图提供了应用程序接口 (API) 。

NumPy安装

Anaconda 已自带NumPy库无需安装

Pycharm安装NumPy库

Window环境:

`pip install numpy pip -i https://pypi.douban.com/simple`

```
C:\Users\Administrator>pip install numpy pip -i https://pypi.douban.com/simple
Looking in indexes: https://pypi.douban.com/simple
Requirement already satisfied: numpy in c:\python38\lib\site-packages (1.18.1)
Requirement already satisfied: pip in c:\python38\lib\site-packages (20.0.2)
WARNING: You are using pip version 20.0.2; however, version 20.1 is available.
You should consider upgrading via the 'c:\python38\python.exe -m pip install --upgrade pip' command.
```

Ubuntu & Debian环境:

`sudo apt-get install python-numpy`

CentOS/Fedora环境:

`sudo dnf install numpy`

Mac环境:

`python -m pip install numpy`

```
C:\Users\Administrator>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from numpy import *
>>> eye(4)
array([[1., 0., 0., 0.],
       [0., 1., 0., 0.],
       [0., 0., 1., 0.],
       [0., 0., 0., 1.]])
>>> exit()
```

eye(4) 生成对角矩阵

安装验证

NumPy Narray对象

NumPy 最重要的一个特点是其 N 维数组对象 ndarray，它是一系列同类型数据的集合，以 0 下标为开始进行集合中元素的索引。

ndarray 对象是用于存放同类型元素的多维数组。

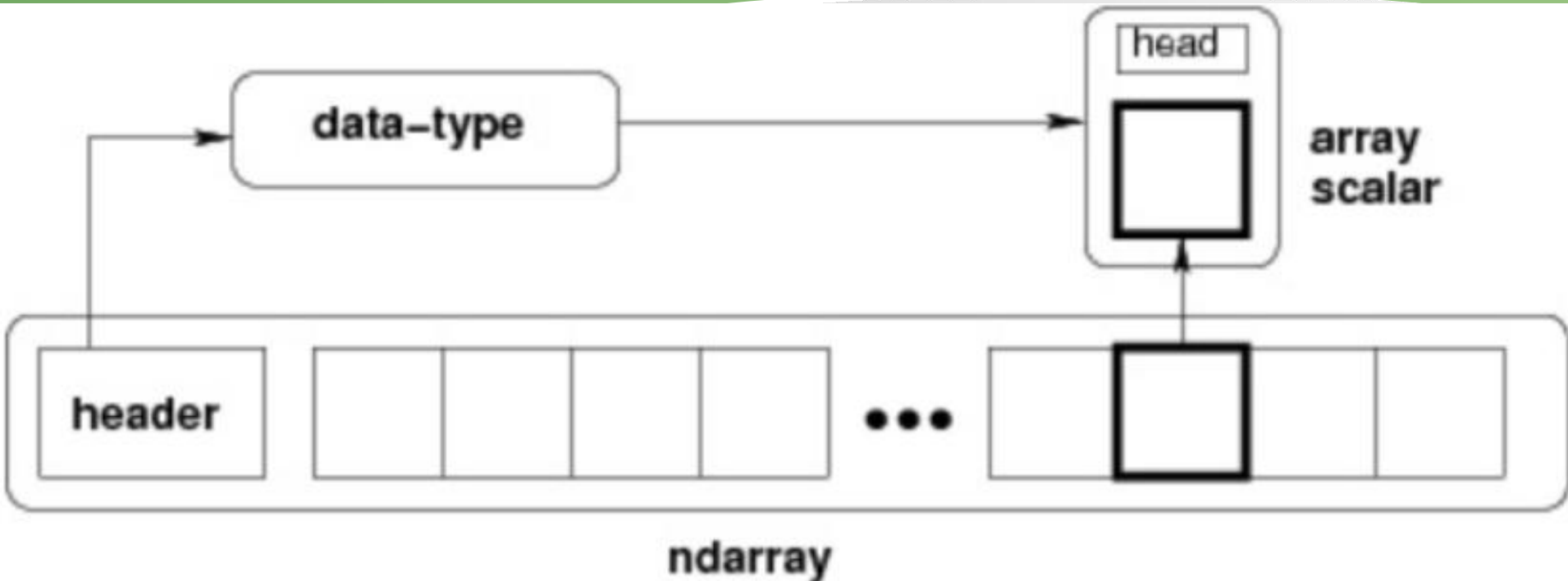
ndarray 中的每个元素在内存中都有相同存储大小的区域。

ndarray 内部由以下内容组成：

1. 一个指向数据（内存或内存映射文件中的一块数据）的指针
2. 数据类型或 dtype，描述在数组中的固定大小值的格子
3. 一个表示数组形状（shape）的元组，表示各维度大小的元组
4. 一个跨度元组（stride），其中的整数指的是为了前进到当前维度下一个元素需要"跨过"的字节数

NumPy结构和函数声明

01_Ndarray演示



`numpy.array(object, dtype = None, copy = True, order = None, subok = False, ndmin = 0)`

名称	描述
object	数组或嵌套的数列
dtype	数组元素的数据类型，可选
copy	对象是否需要复制，可选
order	创建数组的样式，C为行方向，F为列方向，A为任意方向（默认）
subok	默认返回一个与基类类型一致的数组
ndmin	指定生成数组的最小维度

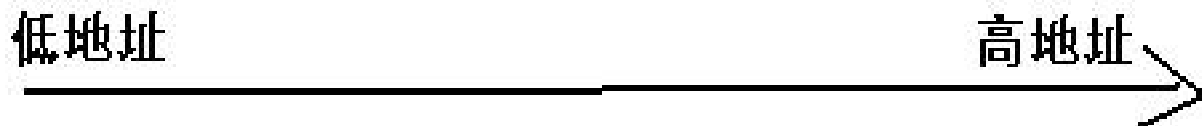
NumPy数据类型

名称	描述
bool_	布尔型数据类型 (True 或者 False)
int_	默认的整数类型 (类似于 C 语言中的 long, int32 或 int64)
intc	与 C 的 int 类型一样, 一般是 int32 或 int 64
intp	用于索引的整数类型 (类似于 C 的 ssize_t, 一般情况下仍然是 int32 或 int64)
int8	字节 (-128 to 127)
int16	整数 (-32768 to 32767)
int32	整数 (-2147483648 to 2147483647)
int64	整数 (-9223372036854775808 to 9223372036854775807)
uint8	无符号整数 (0 to 255)
uint16	无符号整数 (0 to 65535)
uint32	无符号整数 (0 to 4294967295)
uint64	无符号整数 (0 to 18446744073709551615)
float_	float64 类型的简写
float16	半精度浮点数, 包括: 1 个符号位, 5 个指数位, 10 个尾数位
float32	单精度浮点数, 包括: 1 个符号位, 8 个指数位, 23 个尾数位
float64	双精度浮点数, 包括: 1 个符号位, 11 个指数位, 52 个尾数位
complex_	complex128 类型的简写, 即 128 位复数
complex64	复数, 表示双 32 位浮点数 (实数部分和虚数部分)
complex128	复数, 表示双 64 位浮点数 (实数部分和虚数部分)

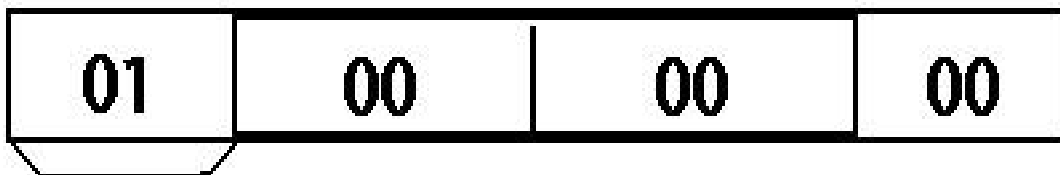
NumPy数据类型

数据类型(dtype)对象是用来描述与数组对应的内存区域如何使用，这依赖如下几个方面：

- 1.数据的类型（整数，浮点数或者 Python 对象）
- 2.数据的大小（例如， 整数使用多少个字节存储）
- 3.数据的字节顺序（小端法或大端法）



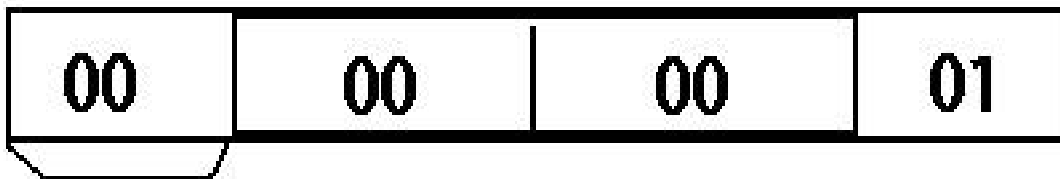
小端法:



低字节

低字节存在低地址
高字节存在高地址

大端法:



高字节

低字节存在高地址
高字节存在低地址

NumPy数据类型

在结构化类型的情况下，字段的名称、每个字段的数据类型和每个字段所取的内存块的部分

如果数据类型是子数组，它的形状和数据类型

字节顺序是通过对数据类型预先设定"<"或">"来决定的。"<"意味着小端法(最小值存储在最小的地址，即低位组放在最前面)。">"意味着大端法(即高位组放在最前面)。

dtype 对象是使用以下语法构造的：

```
numpy.dtype(object, align, copy)
```

- 1.object - 要转换为的数据类型对象
- 2.align - 如果为 true，填充字段使其类似 C 的结构体。
- 3.copy - 复制 dtype 对象，如果为 false，则是对内置数据类型对象的引用

02_dtype.py演示

NumPy数组属性

03_ndarrayAttr.py演示

属性	说明
<code>ndarray.ndim</code>	秩，即轴的数量或维度的数量
<code>ndarray.shape</code>	数组的维度，对于矩阵，n 行 m 列
<code>ndarray.size</code>	数组元素的总个数，相当于 <code>.shape</code> 中 $n*m$ 的值
<code>ndarray.dtype</code>	<code>ndarray</code> 对象的元素类型
<code>ndarray.itemsize</code>	<code>ndarray</code> 对象中每个元素的大小，以字节为单位
<code>ndarray.flags</code>	<code>ndarray</code> 对象的内存信息
<code>ndarray.real</code>	<code>ndarray</code> 元素的实部
<code>ndarray.imag</code>	<code>ndarray</code> 元素的虚部
<code>ndarray.data</code>	包含实际数组元素的缓冲区，由于一般通过数组的索引获取元素，所以通常不需要使用这个属性。

NumPy创建数组

04_ndarrayCreate.py演示

ndarray 数组除了可以使用底层 ndarray 构造器来创建外，也可以通过以下几种方式来创建：

```
numpy.empty(shape, dtype = float, order = 'C')
```

#创建一个指定形状（shape）、数据类型（dtype）且未初始化的数组

参数	描述
shape	数组形状
dtype	数据类型，可选
order	有"C"和"F"两个选项,分别代表，行优先和列优先，在计算机内存中的存储元素的顺序。

```
numpy.zeros(shape, dtype = float, order = 'C')
```

#创建指定大小的数组，数组元素以 0 来填充

参数	描述
shape	数组形状
dtype	数据类型，可选
order	'C' 用于 C 的行数组，或者 'F' 用于 FORTRAN 的列数组

```
numpy.ones(shape, dtype = None, order = 'C')
```

#创建指定形状的数组，数组元素以 1 来填充

#参数和描述同numpy.zeros

NumPy从已有数组创建

05_ndarrayExistCreate.py演示

```
numpy.asarray(a, dtype = None, order = None)
```

参数	描述
a	任意形式的输入参数, 可以是, 列表, 列表的元组, 元组, 元组的元组, 元组的列表, 多维数组
dtype	数据类型, 可选
order	可选, 有"C"和"F"两个选项, 分别代表, 行优先和列优先, 在计算机内存中的存储元素的顺序。

```
numpy.frombuffer(buffer, dtype = float, count = -1, offset = 0)
```

#用于实现动态数组

参数	描述
buffer	可以是任意对象, 会以流的形式读入。
dtype	返回数组的数据类型, 可选
count	读取的数据数量, 默认为-1, 读取所有数据。
offset	读取的起始位置, 默认为0。

```
numpy.fromiter(iterable, dtype, count=-1)
```

#从可迭代对象中建立 ndarray 对象, 返回一维数组

参数	描述
iterable	可迭代对象
dtype	返回数组的数据类型
count	读取的数据数量, 默认为-1, 读取所有数据

NumPy从数值范围创建数组

numpy.arange(start, stop, step, dtype)

06_ndarrayValueRangeCreate.py演示

参数	描述
start	起始值，默认为0
stop	终止值（不包含）
step	步长，默认为1
dtype	返回ndarray的数据类型，如果没有提供，则会使用输入数据的类型。

np.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)

创建一个一维数组，数组是一个等差数列

参数	描述
start	序列的起始值
stop	序列的终止值，如果endpoint为true，该值包含于数列中
num	要生成的等步长的样本数量，默认为50
endpoint	该值为 true 时，数列中包含stop值，反之不包含，默认是True。
retstep	如果为 True 时，生成的数组中会显示间距，反之不显示。
dtype	ndarray 的数据类型

np.logspace(start, stop, num=50, endpoint=True, base=10.0, dtype=None)

创建一个等比数列

参数	描述
start	序列的起始值为：base ** start
stop	序列的终止值为：base ** stop。如果endpoint为true，该值包含于数列中
num	要生成的等步长的样本数量，默认为50
endpoint	该值为 true 时，数列中中包含stop值，反之不包含，默认是True。
base	对数 log 的底数，默认10
dtype	ndarray 的数据类型

NumPy切片

ndarray对象的内容可以通过索引或切片来访问和修改，与 Python 中 list 的切片操作一样。

ndarray 数组可以基于 0 - n 的下标进行索引，切片对象可以通过内置的 slice 函数，并设置 start, stop 及 step 参数进行，从原数组中切割出一个新数组。

07_ndarraySlice.py演示

NumPy高级索引

NumPy 比一般的 Python 序列提供更多的索引方式。除了之前看到的用整数和切片的索引外，数组可以由整数数组索引、布尔索引及花式索引。

08_ndarrayIndex.py演示

NumPy广播

广播是 numpy 对不同形状(shape)的数组进行数值计算的方式，对数组的算术运算通常在相应的元素上进行。

如果两个数组 a 和 b 形状相同，即满足 `a.shape == b.shape`，那么 `a*b` 的结果就是 a 与 b 数组对应位相乘。这要求维数相同，且各维度的长度相同。

09_ndarrayBroadcast.py演示

NumPy迭代

NumPy 迭代器对象 `numpy.nditer` 提供了一种灵活访问一个或者多个数组元素的方式。迭代器最基本的任务的可以完成对数组元素的访问。

[10_ndarrayNditer.py](#)演示

NumPy修改数组形状

函数	描述
reshape	不改变数据的条件下修改形状
flat	数组元素迭代器
flatten	返回一份数组拷贝，对拷贝所做的修改不会影响原始数组
ravel	返回展开数组

11_ndarrayModify1.py演示

NumPy翻转数组

函数	描述
transpose	对换数组的维度
ndarray.T	和 self.transpose() 相同
rollaxis	向后滚动指定的轴
swapaxes	对换数组的两个轴

12_ndarrayModify2.py演示

NumPy修改数组维度

维度	描述
broadcast	产生模仿广播的对象
broadcast_to	将数组广播到新形状
expand_dims	扩展数组的形状
squeeze	从数组的形状中删除一维条目

13_ndarrayModify3.py演示

NumPy连接数组

函数	描述
concatenate	连接沿现有轴的数组序列
stack	沿着新的轴加入一系列数组。
hstack	水平堆叠序列中的数组（列方向）
vstack	竖直堆叠序列中的数组（行方向）

14_ndarrayModify4.py演示

NumPy分割数组

函数	描述
split	将一个数组分割为多个子数组
hsplit	将一个数组水平分割为多个子数组（按列）
vsplit	将一个数组垂直分割为多个子数组（按行）

15_ndarrayModify5.py演示

NumPy数组元素添加与删除

函数	元素及描述
resize	返回指定形状的新数组
append	将值添加到数组末尾
insert	沿指定轴将值插入到指定下标之前
delete	删掉某个轴的子数组，并返回删除后的新数组
unique	查找数组内的唯一元素

16_ndarrayModify6.py演示

NumPy位运算

- The integer *bitwise* operators are:

~ - Complement

^ - XOR

& - AND

| - OR

- Byte-sized examples:

$$\begin{array}{r} \sim \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline \end{array} \end{array}$$

$$\begin{array}{r} \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array} \\ \wedge \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array} \end{array}$$

$$\begin{array}{r} \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array} \\ \& \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array} \end{array}$$

$$\begin{array}{r} \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array} \\ | \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array} \end{array}$$

NumPy位运算

`128 >> 1` returns $128/2^1 = 64$

`256 >> 4` returns $256/2^4 = 16$

`-256 >> 4` returns $-256/2^4 = -16$

`128 << 1` returns $128 * 2^1 = 256$

`16 << 2` returns $16 * 2^2 = 64$

NumPy位运算

函数	描述
<code>bitwise_and</code>	对数组元素执行位与操作
<code>bitwise_or</code>	对数组元素执行位或操作
<code>invert</code>	按位取反
<code>left_shift</code>	向左移动二进制表示的位
<code>right_shift</code>	向右移动二进制表示的位

也可以使用 "&"、 "~"、 "|" 和 "^" 等操作符进行计算

17_ndarrayBitwise.py演示

函数	描述
<code>add()</code>	对两个数组的逐个字符串元素进行连接
<code>multiply()</code>	返回按元素多重连接后的字符串
<code>center()</code>	居中字符串
<code>capitalize()</code>	将字符串第一个字母转换为大写
<code>title()</code>	将字符串的每个单词的第一个字母转换为大写
<code>lower()</code>	数组元素转换为小写
<code>upper()</code>	数组元素转换为大写
<code>split()</code>	指定分隔符对字符串进行分割，并返回数组列表
<code>splitlines()</code>	返回元素中的行列表，以换行符分割
<code>strip()</code>	移除元素开头或者结尾处的特定字符
<code>join()</code>	通过指定分隔符来连接数组中的元素
<code>replace()</code>	使用新字符串替换字符串中的所有子字符串
<code>decode()</code>	数组元素依次调用 <code>str.decode</code>
<code>encode()</code>	数组元素依次调用 <code>str.encode</code>

NumPy数学、算术函数

NumPy 包含大量的各种数学运算的函数，包括三角函数，算术运算的函数，复数处理函数等。

NumPy 算术函数包含简单的加减乘除: `add()`, `subtract()`, `multiply()` 和 `divide()`。

需要注意的是数组必须具有相同的形状或符合数组广播规则。

19_ndarrayMath.py演示

NumPy统计函数

NumPy 提供了很多统计函数，用于从数组中查找最小元素，最大元素，百分位标准差和方差等。

20_ndarrayStat.py演示

NumPy排序、条件筛选函数

NumPy 提供了多种排序的方法。这些排序函数实现不同的排序算法，每个排序算法的特征在于执行速度，最坏情况性能，所需的工作空间和算法的稳定性。

种类	速度	最坏情况	工作空间	稳定性
'quicksort' (快速排序)	1	$O(n^2)$	0	否
'mergesort' (归并排序)	2	$O(n \cdot \log(n))$	$\sim n/2$	是
'heapsort' (堆排序)	3	$O(n \cdot \log(n))$	0	否

21_ndarraySort_Filter.py演示

NumPy副本和视图

副本是一个数据的完整的拷贝，如果我们对副本进行修改，它不会影响到原始数据，物理内存不在同一位置。

视图是数据的一个别称或引用，通过该别称或引用亦便可访问、操作原有数据，但原有数据不会产生拷贝。如果我们对视图进行修改，它会影响到原始数据，物理内存存在同一位置。

视图一般发生在：

22_ndarrayCopy_View.py演示

- 1.numpy 的切片操作返回原数据的视图。
- 2.调用 ndarray 的 view() 函数产生一个视图。

副本一般发生在：

- 1.Python 序列的切片操作，调用deepCopy()函数。
- 2.调用 ndarray 的 copy() 函数产生一个副本。

NumPy矩阵库

NumPy 中包含了一个矩阵库 `numpy.matlib`，该模块中的函数返回的是一个矩阵，而不是 `ndarray` 对象。

一个 $m \times n$ 的矩阵是一个由 m 行 (row) n 列 (column) 元素排列成的矩形阵列。
矩阵里的元素可以是数字、符号或数学式。

$$\begin{bmatrix} 1 & 9 & -13 \\ 20 & 5 & -6 \end{bmatrix}$$

23_numpy_matlib.py演示

NumPy线性代数

NumPy 提供了线性代数函数库 linalg, 该库包含了线性代数所需的所有功能。

函数	描述
dot	两个数组的点积, 即元素对应相乘。
vdot	两个向量的点积
inner	两个数组的内积
matmul	两个数组的矩阵积
determinant	数组的行列式
solve	求解线性矩阵方程
inv	计算矩阵的乘法逆矩阵

24_numpy_linalg.py演示

NumPy IO

Numpy 可以读写磁盘上的文本数据或二进制数据。

NumPy 为 ndarray 对象引入了一个简单的文件格式：.npz。

.npz 文件用于存储重建 ndarray 所需的数据、图形、dtype 和其他信息。

常用的 IO 函数有：

- 1.load() 和 save() 函数是读写文件数组数据的两个主要函数，默认情况下，数组是以未压缩的原始二进制格式保存在扩展名为 .npz 的文件中。
- 2.savetxt() 函数用于将多个数组写入文件，默认情况下，数组是以未压缩的原始二进制格式保存在扩展名为 .npz 的文件中。
- 3.loadtxt() 和 savetxt() 函数处理正常的文本文件(.txt 等)

25_numpy_io.py演示

NumPy&Matplotlib

Matplotlib 是 Python 的绘图库。它可与 NumPy 一起使用，提供了一种有效的 MatLab 开源替代方案。它也可以和图形工具包一起使用，如 PyQt 和 wxPython。

Windows安装:

`pip install matplotlib pip -i https://pypi.douban.com/simple`

```
C:\Users\Administrator>pip install matplotlib pip -i https://pypi.douban.com/simple
Looking in indexes: https://pypi.douban.com/simple
Requirement already satisfied: matplotlib in c:\python38\lib\site-packages (3.2.0)
Requirement already satisfied: pip in c:\python38\lib\site-packages (20.0.2)
Requirement already satisfied: cyclar>=0.10 in c:\python38\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\python38\lib\site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: numpy>=1.11 in c:\python38\lib\site-packages (from matplotlib) (1.18.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\python38\lib\site-packages (from matplotlib) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\python38\lib\site-packages (from matplotlib) (2.4.6)
Requirement already satisfied: six in c:\python38\lib\site-packages (from cyclar>=0.10->matplotlib) (1.14.0)
Requirement already satisfied: setuptools in c:\python38\lib\site-packages (from kiwisolver>=1.0.1->matplotlib) (41.2.0)
WARNING: You are using pip version 20.0.2; however, version 20.1 is available.
You should consider upgrading via the 'c:\python38\python.exe -m pip install --upgrade pip' command.
```

Debian/Ubuntu:

`sudo apt-get install python-matplotlib`

Centos/Fedora/Redhat:

`sudo yum install python-matplotlib`

```
C:\Users\Administrator>python -m pip list
Package            Version
-----
asgiref            3.2.5
cyclar             0.10.0
Django            3.0.4
graphviz          0.13.2
h5py              2.10.0
joblib            0.14.1
Keras             2.3.1
Keras-Applications 1.0.8
Keras-Preprocessing 1.1.0
kiwisolver        1.1.0
matplotlib        3.2.0
mysqlclient       1.4.6
nose              1.3.7
```

NumPy&Matplotlib

作为线性图的替代，可以通过向 plot() 函数添加格式字符串来显示离散值

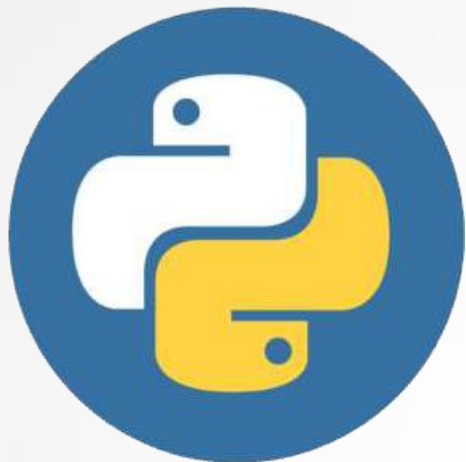
字符	描述	字符	描述
'-'	实线样式	's'	正方形标记
'--'	短横线样式	'p'	五边形标记
'-.'	点划线样式	'*'	星形标记
':'	虚线样式	'h'	六边形标记 1
'.'	点标记	'H'	六边形标记 2
','	像素标记	'+'	加号标记
'o'	圆标记	'x'	X 标记
'v'	倒三角标记	'D'	菱形标记
'^'	正三角标记	'd'	窄菱形标记
'<'	左三角标记	'|'	竖直线标记
'>'	右三角标记	'_'	水平线标记
'1'	下箭头标记		
'2'	上箭头标记		
'3'	左箭头标记		
'4'	右箭头标记		

NumPy&Matplotlib

颜色缩写

字符	颜色
'b'	蓝色
'g'	绿色
'r'	红色
'c'	青色
'm'	品红色
'y'	黄色
'k'	黑色
'w'	白色

26_numpy_matplotlib.py演示



感谢您的聆听！

