

Numpy库的使用

讲师：萨缪尔 Samuel



萨缪尔老师

网易数据分析教研负责人

前盛大游戏战略规划总监、前腾讯游戏商业智能中心Leader

上海交通大学高金硕士、《哈佛管理导师》外部导师

- 知乎大V：「萨缪尔」主要聚集与商业分析、行业与战略分析、Python数据分析等
- BAT互联网巨头商业洞察分析 & 咨询公司战略咨询背景
- 擅长行业趋势研究和战略管理咨询工具，为20多家上市公司提供战略发展决策建议

课程

亮点

1

Numpy库简单介绍

2

Numpy常用数据类型

3

Numpy生成数组的函数

4

Numpy的N维数组

课程

亮点

5

Numpy的切片与索引

6

Numpy的广播机制

7

Numpy的常用函数

Numpy库 简单介绍

1

Numpy库简介

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 的前身 Numeric 最早是由 Jim Hugunin 与其它协作者共同开发，2005 年，Travis Oliphant 在 Numeric 中结合了另一个同性质的程序库 Numarray 的特色，并加入了其它扩展而开发了 NumPy。NumPy 为开放源代码并且由许多协作者共同维护开发。

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：

- 一个强大的N维数组对象 ndarray
- 广播功能函数
- 线性代数
- 随机数生成

Numpy库 常用数据类型



Numpy的数据类型

类型	说明
bool	布尔型（值为 True 或 False），占用 1bit
inti	其长度取决于平台的整数（通常为 int32 或者 int64）
int8	字节类型（取值范围从-128~127）
int16	整型（取值范围-32768~32767）
int32	整型（取值范围为 $-2^{31} \sim 2^{31}-1$ ）
int64	整型（取值范围为 $-2^{63} \sim 2^{63}-1$ ）
uint8	无符号整型（取值范围为 0~255）
uint16	无符号整型（取值范围为 0~65535）
uint32	无符号整型（取值范围为 $0 \sim 2^{32}-1$ ）
uint64	无符号整型（取值范围为 $0 \sim 2^{64}-1$ ）
float16	半精度浮点型：符号占用 1bit，指数占用 5 bit，尾数占用 10 bit
float32	单精度浮点型：符号占用 1 bit，指数占用 8 bit，尾数占用 23 bit
float64 或者 float	双精度浮点型：符号占用 1 bit，指数占用 11 bit，尾数占用 52 bit
complex64	复数类型，由两个 32 位浮点数（实部和虚部）表示
complex128 或者 complex	复数类型，由两个 64 位浮点数（实部和虚部）表示

切记：不能把复数类型转化成整型。当我们试图进行这种转换时，将会触发 **TypeError** 错误

Numpy库 生成数组的函数



Numpy 生成数组的函数

函数	形式	用处
<code>np.arange()</code>	<code>arange(start, stop, step, dtype)</code>	生成任意长度的数组
<code>np.linspace()</code>	<code>linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)</code>	生成任意长度的数组
<code>np.zeros()</code>	<code>zeros(shape, dtype=None, order='C')</code>	生成任意维度数值为0的矩阵
<code>np.ones()</code>	<code>ones(shape, dtype=None, order='C')</code>	生成任意维度数值为1的矩阵

Numpy库 N维数组



4

Numpy N维数组

具体案例：

```
X1=np.arange(30).reshape(2,3,5)
```

```
X2=np.arange(30).reshape(3,5,2)
```

15	16	17	18	19
0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

Shape: (2, 3, 5)



三维数组用包含3个元素的元组来表示，
以此类推...

Numpy库 切片与索引



一维数组的切片与索引

```
In [21]: #一维数组的切片与索引
import numpy as np
a = np.arange(9)
print(a)
a[3:7]
```

```
[0 1 2 3 4 5 6 7 8]
```

```
Out[21]: array([3, 4, 5, 6])
```

```
In [27]: #一般情况下一个 cell 后只能输出一个结果,多行输出的话需要加上一下两行代码
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all' # 默认为'last', 即输出最后一个结果

import numpy as np
a = np.arange(12)
a
a[3:7]
a[0:9:2] #下标每次递增2, 默认为1
a[::-1] #逆向取数
```

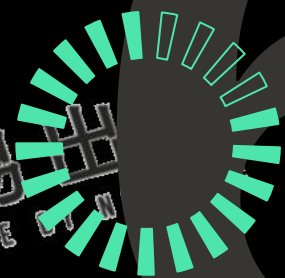
```
Out[27]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
Out[27]: array([3, 4, 5, 6])
```

```
Out[27]: array([0, 2, 4, 6, 8])
```

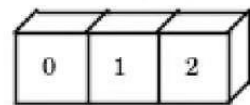
```
Out[27]: array([11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1,  0])
```


Numpy库 广播机制

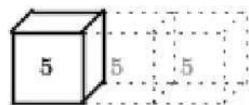


Numpy广播机制

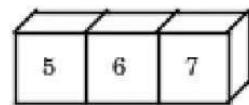
`np.arange(3) + 5`



+

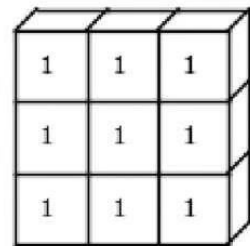


=

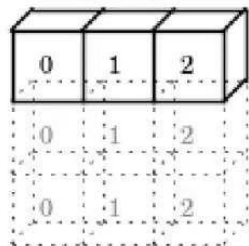


(a)

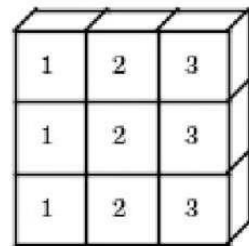
`np.ones((3, 3)) + np.arange(3)`



+

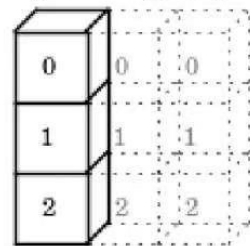


=

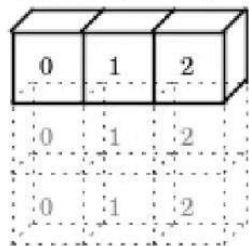


(b)

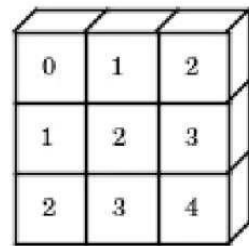
`np.arange(3).reshape((3, 1)) + np.arange(3)`



+



=



(c)

Numpy库 常用函数



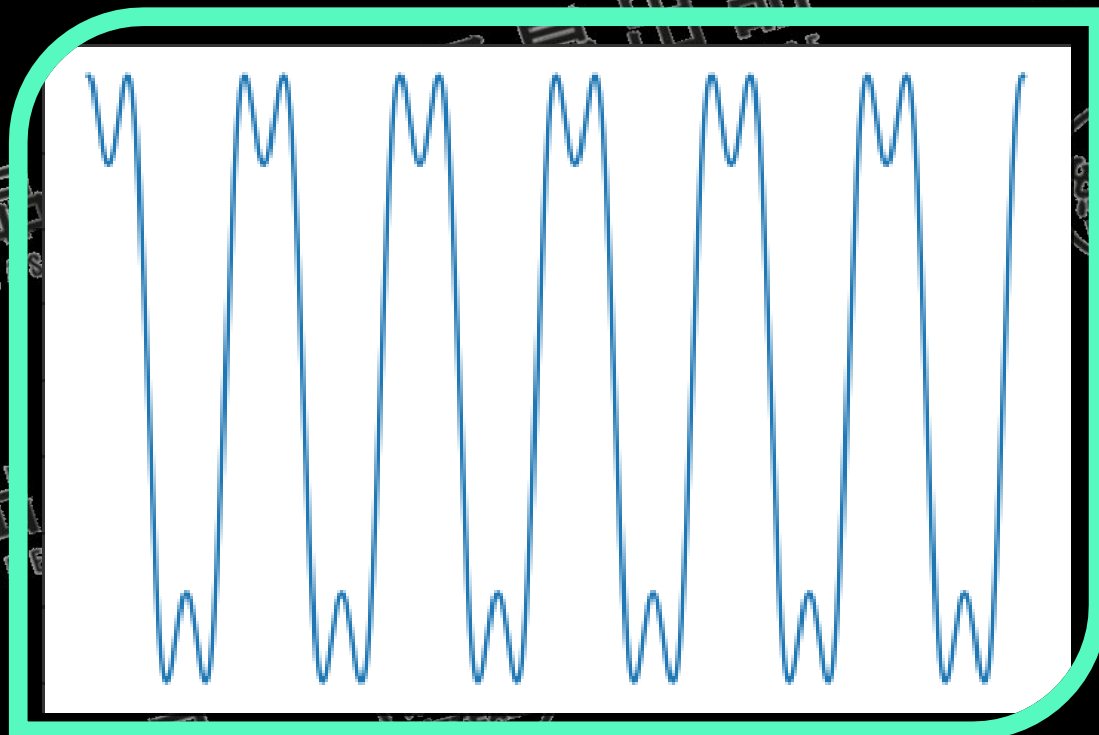
Numpy常用函数

函数类型	常用函数
数学函数	<code>np.sin()</code> 、 <code>np.cos()</code> 、 <code>np.tan()</code>
算数函数	<code>np.add()</code> <code>np.subtract()</code> <code>np.multiply()</code> <code>np.divide()</code>
统计函数	<code>np.min()</code> <code>np.max()</code> <code>np.ptp()</code> <code>np.percentile()</code> <code>np.mean()</code>
排序条件筛选函数	<code>np.sort()</code> <code>np.argsort()</code> <code>np.lexsort()</code>
随机函数	<code>np.random.random()</code> <code>np.random.rand()</code> <code>np.random.randint()</code>

案例：计算 $\cos^3 x + \sin^3 x$ 的最大值最小值

```
import matplotlib.pyplot as plt
import numpy as np
import math
```

```
pi=math.pi
x=np.linspace(0,12*pi,1000)
y=(np.cos(x))**3+(np.sin(x))**3
plt.figure()
plt.plot(x,y)
plt.show()
```



课程

总结

1

Numpy库的**发展历程**以及**主要用途**

2

Numpy的**数据类型**以及**生成数组**的方法

3

Numpy的**切片索引**以及**广播机制**

4

Numpy的常用**函数**

谢谢观看

