

Numpy 使用教程

一、实验介绍

1.1 实验内容

如果你使用 Python 语言进行科学计算，那么一定会接触到 Numpy。Numpy 是支持 Python 语言的数值计算扩充库，其拥有强大的多维数组处理与矩阵运算能力。除此之外，Numpy 还内建了大量的函数，方便你快速构建数学模型。

1.2 实验知识点

- Numpy 安装
- Numpy 数值类型介绍

1.3 实验环境

- python2.7
- Xfce 终端
- ipython 终端

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有 Python 基础，并对使用 Numpy 进行科学计算感兴趣的用户。

二、Numpy 简介

Numpy 的英文全称为 Numerical Python，意味 Python 面向数值计算的第三方库。Numpy 的特点在于，针对 Python 内建的数组类型做了扩充，支持更高维度的数组和矩阵运算，以及更丰富的数学函数。

Numpy 是 Scipy.org 中最重要的库之一，它同时也被 Pandas，Matplotlib 等我们熟知的第三方库作为核心计算库。当你在单独安装这些库时，你会发现同时会安装 Numpy 作为依赖。

三、Numpy 安装

首先，我们需要在实验楼的在线环境或者本地安装 Numpy。最方便地，就是通过 pip 来进行安装了。我们只需要打开终端，键入下面的命令就可以完成安装了。

```
sudo pip install numpy
```

Windows 用户只需要打开系统命令行工具，输入 `pip install numpy` 即可。当然，Ubuntu & Debian 用户还可以用 apt-get 安装：

```
sudo apt-get install python-numpy
```

四、Numpy 数值类型

安装完毕之后，我们先来了解 Numpy 支持的数据类型。为了更加方便地学习和演示，以下代码在为特别注明的情况下，均在 ipython 交互式终端运行，你可以通过在线环境左下角 系统应用菜单>附件 打开。

Python 本身支持的数值类型有 `int`（整型，python2 中存在 `long` 长整型）、`float`（浮点型）、`bool`（布尔型）和 `complex`（复数型）。

而 Numpy 支持比 Python 本身更为丰富的数值类型，细分如下：

1. `bool` : 布尔类型, 1 个字节, 值为 `True` 或 `False`。
2. `int` : 整数类型, 通常为 `int64` 或 `int32` 。
3. `intc` : 与 C 里的 `int` 相同, 通常为 `int32` 或 `int64`。
4. `intp` : 用于索引, 通常为 `int32` 或 `int64`。
5. `int8` : 字节 (从 -128 到 127)
6. `int16` : 整数 (从 -32768 到 32767)
7. `int32` : 整数 (从 -2147483648 到 2147483647)
8. `int64` : 整数 (从 -9223372036854775808 到 9223372036854775807)
9. `uint8` : 无符号整数 (从 0 到 255)
10. `uint16` : 无符号整数 (从 0 到 65535)
11. `uint32` : 无符号整数 (从 0 到 4294967295)
12. `uint64` : 无符号整数 (从 0 到 18446744073709551615)
13. `float` : `float64` 的简写。
14. `float16` : 半精度浮点, 5 位指数, 10 位尾数
15. `float32` : 单精度浮点, 8 位指数, 23 位尾数
16. `float64` : 双精度浮点, 11 位指数, 52 位尾数
17. `complex` : `complex128` 的简写。
18. `complex64` : 复数, 由两个 32 位浮点表示。
19. `complex128` : 复数, 由两个 64 位浮点表示。

在 Numpy 中, 上面提到的这些数值类型都被归于 `dtype` (data-type) 对象的实例。

我们可以用 `numpy.dtype(object, align, copy)` 来指定数值类型。而在数组里面, 可以用 `dtype=` 参数。

```
import numpy as np
```

```
a = np.array([1.1, 2.2, 3.3], dtype=np.float64) # 指定 1 维数组的数值类型为 float64
```

```
In [3]: a
Out[3]: array([ 1.1,  2.2,  3.3])
```

另外, 你可以使用 `.astype()` 方法在不同的数值类型之间相互转换。

```
a.astype(int) # 将 a 的数值类型从 float64 转换为 int
```

```
In [6]: a.astype(int)
Out[6]: array([1, 2, 3])
```

最后，你可以使用 `.dtype` 来查看 `dtype` 属性。

```
a.dtype # 查看 a 的数值类型
```

```
In [10]: a.dtype
Out[10]: dtype('int64')
```

三、实验总结

了解并数组 Numpy 支持的数值类型至关重要，因为 `dtype` 参数几乎延伸到了 `numpy` 中所有与多维数组相关的方法中去。

**本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。*

下一节：[Numpy 多维数组创建及属性 \(/courses/912/labs/3406/document\)](/courses/912/labs/3406/document)