2.1 NumPy简介

NumPy是Numerical Python的缩写，它是Python中一个用于科学计算的扩展程序库。NumPy因其包含丰富的数学函数、支持高维数组与矩阵且运行速度快等特点而得到了广泛应用。具体而言，NumPy的主要功能有：

ndarray，一种支持矢量计算和广播能力的多维数组，运行速度快且节省空间；

能快速处理整组数据的数学函数；

整合C/C++/Fortran代码的工具；

线性代数、傅立叶变换、随机数生成等功能；

磁盘数据读写工具；

内存映射文件操作工具。

2.2 NumPy的安装

和许多扩展库一样，numpy的安装可以通过pip进行。如果你已经安装了Python 3.x或以上版本，则安装Python时就已经自带了pip功能。打开命令行窗口，输入pip install numpy命令，即可自动下载并安装numpy。

安装完成后，可以打开一个命令行窗口进入Python终端会话，对之前的安装进行测试。输入import numpy as np命令，尝试导入numpy，如果没有出现任何错误信息提示，说明说明numpy已经成功安装，可以使用。

如果你已经安装了Anaconda，则Anaconda中已经集成了numpy，无需重复安装。

2.3 NumPy数组

2.3.1 NumPy数组的创建和基本属性

数组NumPy中所提供的最基本，也是最常用的一种数据类型。NumPy数组的维度数量称之为秩（rank），即n维数组的秩为n。对于多维数组而言，构成它的每一个单一维度的数组称之为一个轴（axis）或维度（dimension）。例如，一个二维数组可以看作两个一维数组；这两个一维数组中的每个元素又可以看做一个一维数组。

首先介绍如何创建一个NumPy数组并查看一些关键属性。可以使用np.array()方法创建数组。下面这条语句便创建了一个简单的一维数组：

|  |
| --- |
| >>> a = np.array([0, 1, 2, 3]) |

输入这条语句后，再键入数组名a并按回车键，就会出现array([0, 1, 2, 3])字样，表明我们已经创建了数组 [0, 1, 2, 3]。

使用type(a)函数可以查看到，对象a的类型为ndarray：

|  |
| --- |
| >>> type(a)  <class 'numpy.ndarray'> |

而具体到数组中元素的数值类型，可以用a.dtype查看：

|  |
| --- |
| >>> a.dtype  dtype('int32') |

由于NumPy的最主要作用之一就是处理矩阵数组，因此了解一个数组的维度就很关键。在NumPy中，我们可以用shape方法查看数组的维度，shape方法返回一个整数元组，元组中每一个元素代表每个维度的大小。举例说来，如果我们对一个数组b调用方法b.shape，得到一元组(3,)，表示b是一个一维数组，包含3个元素；如果得到二元组(2, 3)，表示b是一个二维数组，在两个维度上分别含有2个元素和3个元素，以此类推。

NumPy中的数组对象ndarray有如下较为常用的属性：

ndarray.shape： 数组的维度，或矩阵的行列数。它返回一个元组，元组的长度就是维度的数目。

ndarray.size： 数组中元素的总数

ndarray.drype：ndarray对象的元素类型

ndarray.itemsize：ndarray对象中单个元素大小。单位为字节。

ndarray.ndim：维度数

#end