

Python 科学计算基础

第六章 NumPy 数组和矩阵计算

2025 年 9 月 5 日

目录

创建数组

- 已有数据元素

- 没有数据元素但已知形状

- 改变数组的形状

- 数组的堆叠

- 数组的分割

数组的运算

- 基本运算

- 函数运算

索引、切片和迭代

复制和视图

矩阵计算

实验 6：NumPy 数组和矩阵计算

概述

NumPy 扩展库定义了由同类型的元素组成的多维数组 `ndarray` 及其常用运算。`ndarray` 是科学计算中最常用的数据类型，它相对列表的优势是运算速度更快和占用内存更少。`ndarray` 是一个类，它的别名是 `array`。它的主要属性包括 `ndim`(维数)、`shape`(形状，即由每个维度的长度构成的元组)、`size`(元素数量) 和 `dtype`(元素类型：可以是 Python 的内置类型，也可以是 NumPy 定义的类型，例如 `numpy.int32`、`numpy.int16` 和 `numpy.float64` 等)。

矩阵使用 `numpy.array` 类表示。SciPy 扩展库的 `scipy.linalg` 模块提供了常用的矩阵计算函数。

已有数据元素

可以从存储了数据元素的列表、元组或它们的嵌套创建数组，其类型取决于数据元素的类型，也可以使用 `dtype` 关键字实参指定类型。如果数据元素无法使用指定类型表示，可能会发生溢出或精度损失。

程序 6.1

没有数据元素但已知形状

- ▶ `np.zeros` 和 `np.ones`
- ▶ `zeros_like` 和 `ones_like`
- ▶ `np.arange`: 下界、上界和步长
- ▶ `np.linspace`: 下界、上界和数量
- ▶ `np.random.seed` 和 `np.random.rand`
- ▶ `np.fromfunction`

程序 6.2

改变数组的形状

改变形状是指改变各维度的长度，但不改变组成数组的元素。前三种方法不会改变原数组的形状。

- ▶ `flatten` 方法从一个多维数组生成一维数组。
- ▶ `reshape` 方法从原数组生成一个指定形状的新数组。
- ▶ `T` 方法从一个数组生成它的转置。
- ▶ `resize` 方法则将原数组改变为指定的形状。

程序 6.3

数组的堆叠

水平堆叠

- ▶ `np.hstack` 函数沿第二个维度将两个数组堆叠在一起形成新的数组。

垂直堆叠

- ▶ `np.vstack` 函数沿第一个维度将两个数组堆叠在一起形成新的数组。
- ▶ `np.r_` 函数将多个数组 (数值) 堆叠在一起形成新的数组, 其中语法 `start:stop:step` 等同于 `np.arange(start, stop, step)`, 语法 `start:stop:numj` 等同于 `np.linspace(start, stop, num)`。

程序 6.4

数组的分割

- ▶ `np.hsplit` 函数沿第二个维度将一个数组分割成为多个数组，可以指定一个正整数表示均匀分割得到的数组的数量或指定一个元组表示各分割点的索引值。
- ▶ `np.vsplit` 函数沿第一个维度将一个数组分割成为多个数组，参数和 `hsplit` 类似。

程序 6.5

- 1.`np.hsplit(arr, n)` 表示将数组`arr`均匀分割成`n`个数组
- 2.`np.hsplit(arr,[t1,t2])`表示将数组`arr`按照列索引`[t1,t2]`分割

基本运算

- ▶ 数组的一元运算和二元运算对于每个元素分别进行。二元运算要求两个数组具有相同的形状。
- ▶ 两个二维数组之间的矩阵乘法可通过@运算符或 dot 方法完成。
- ▶ 数组和单个数值之间的运算等同于数组的每个元素分别和数值之间进行运算。
- ▶ “+=”、“-=”、“*=”、“/=” 和 “**=” 运算符不返回新的数组，而是用运算结果替代原数组。
- ▶ 如果参与运算的多个数组的元素的类型不同，则结果的类型设为取值范围最大的类型。

函数运算

- ▶ `sum`、`max` 和 `min` 方法分别返回一个数组包含的所有元素的总和、最大值和最小值。
- ▶ `np.sort` 函数可对数组进行排序。对于二维数组，可通过 `axis` 关键字实参指定对每行或每列分别计算。
- ▶ NumPy 提供了很多数学函数 (`sin`, `cos`, `exp` 等)，这些函数可分别作用于数组的每个元素。输入函数名和问号可以获取该函数的详细说明。

程序 6.7

索引、切片和迭代

一维数组可以像列表一样进行索引、切片和迭代。多维数组的每个维度都有一个索引，这些索引用逗号分隔共同构成一个完整的索引元组。如果提供的索引的数量小于维度，则等同于将缺失的维度全部选择（即冒号“:”）。省略号（“...”）代表多个可省略的冒号。

将二维数组看成是一个矩阵，对于二维数组的索引和迭代以矩阵的行为单位。

程序 6.8

程序 6.9

数组作为索引

数组不仅可以使⽤整数和切片作为索引，也可以使⽤整数数组 (或列表) 或布尔值数组 (或列表) 作为索引。

使⽤一个整数数组 a 作为一个数组 b 的索引得到的结果是一个数组 c ， c 中的每个元素是以 a 中的每个元素作为索引值所获取的数组 b 中的对应元素。程序 6.10

使⽤一个布尔值数组 a 作为一个数组 b 的索引得到的结果是一个数组 c ， c 中仅包含以数组 a 中的值为 `True` 的元素的索引值所获取的数组 b 中的对应元素。对于多维数组，可分别对每个维度索引。程序 6.11

复制和视图

在对数组进行运算时，有时元素会复制到一个新数组中，有时不发生复制。

程序 6.12 的 Out[4] 行的输出结果是：k is h 的值为 True，并且 k 和 h 的 id 值相同。这说明 In[3] 行的赋值运算给已有数组 h 创建一个别名 k，而不发生元素的复制。Python 语言的函数调用对于可变实参是传引用而非传值，所以也不发生元素的复制。

复制和视图

`view` 方法从已有数组创建一个新的数组，可以理解为原数组的一个视图。新数组和已有数组共享元素，但可以有不同形状。从原数组切片得到的新数组也是原数组的一个视图。

`ravel` 方法从一个多维数组生成一个一维数组，也是原数组的一个视图。`reshape` 方法从原数组生成一个不同形状的视图。

`copy` 方法将已有数组的元素复制到新创建的数组中，新数组和原数组不共享元素。`copy` 方法的一个用途是复制元素以后可以用 `del` 回收原数组占用的内存空间。

程序 6.12

矩阵计算

矩阵可以使用 `numpy.array` 类的二维对象表示。SciPy 扩展库的 `scipy.linalg` 模块定义了常用的矩阵计算函数。

- ▶ `det` 函数计算一个矩阵的行列式。
- ▶ `inv` 函数计算一个矩阵的逆矩阵。
- ▶ `norm` 函数对于一个矩阵 \mathbf{A} 计算其 Frobenius 范数 $\sqrt{\text{trace}(\mathbf{A}^H \mathbf{A})}$ ，对于一个向量 \mathbf{x} 计算其欧式范数 (各分量的平方和的平方根) $\sqrt{\sum_i |x_i|^2}$ 。
- ▶ `solve(A, b)` 函数求解线性方程组 $\mathbf{Ax} = \mathbf{b}$ 。
- ▶ `eig(A)` 函数求解特征值和特征向量。

程序 6.13

矩阵计算

一个秩为 r 的 $m \times n$ 的实矩阵 \mathbf{A} 的奇异值分解 (singular value decomposition) 为 $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T = [\mathbf{u}_1, \dots, \mathbf{u}_m]\mathbf{S}[\mathbf{v}_1, \dots, \mathbf{v}_n]^T$, 其中 $\mathbf{S} = \left(\begin{array}{c|c} \text{diag}(\sigma_1, \dots, \sigma_r) & \mathbf{0}_{r, n-r} \\ \hline \mathbf{0}_{m-r, r} & \mathbf{0}_{m-r, n-r} \end{array} \right)$, 奇异值为 $\sigma_1, \dots, \sigma_r$, 奇异向量为 $\mathbf{u}_1, \dots, \mathbf{u}_m$ 和 $\mathbf{v}_1, \dots, \mathbf{v}_n$, 且

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i, \quad i = 1, \dots, r, \quad \mathbf{A}\mathbf{v}_i = 0, \quad i = r+1, \dots, n, \quad (1)$$

$$\mathbf{A}^T\mathbf{u}_i = \sigma_i\mathbf{v}_i, \quad i = 1, \dots, r, \quad \mathbf{A}^T\mathbf{u}_i = 0, \quad i = r+1, \dots, m. \quad (2)$$

`svd(A)` 返回一个元组 $(\mathbf{U}, \mathbf{s}, \mathbf{V}^T)$, 其中 \mathbf{s} 是由所有奇异值组成的向量。`diagsvd` 函数返回 \mathbf{S} 。矩阵的秩可通过计数非零奇异值的个数获得。

实验 6：NumPy 数组和矩阵计算

本实验的目的是掌握使用 NumPy 库和 `scipy.linalg` 模块进行矩阵计算。

在 Blackboard 系统提交一个文本文件 (txt 后缀)，文件中记录每道题的源程序和运行结果。

1. 求解线性方程组和矩阵运算

生成一个由实数组成的秩为 4 的 4 行 4 列的矩阵 A 和一个由实数组成的包含 4 个元素的列向量 b 。求解线性方程组 $Ax = b$ 。计算矩阵 A 的转置、行列式、秩、逆矩阵、特征值和特征向量。

2. 计算最小二乘解和矩阵分解

生成一个由实数组成的秩为 4 的 6 行 4 列的实矩阵 \mathbf{B} 和一个由实数组成的包含 6 个元素的列向量矩阵 \mathbf{b} 。计算线性方程组 $\mathbf{B}\mathbf{x} = \mathbf{b}$ 的最小二乘解 $(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{b}$ 。计算矩阵 \mathbf{B} 的奇异值分解，并验证第 16 页的奇异值分解满足的方程。计算矩阵 \mathbf{B} 的 LU 分解。