

第 5 章

1. 和 Python 列表相比, Numpy 数组在科学计算领域具有哪些优势?

答案:

NumPy 数组是 Python 中用于科学计算的核心数据结构, 相比 Python 列表, 它具有以下优势:

(1) NumPy 数组在内存中是连续存储的, 而 Python 列表存储的是对象的引用, 这使得 NumPy 数组的访问速度更快, 尤其是在处理大型数据集时。

(2) NumPy 提供了丰富的向量化操作函数, 可以对整个数组进行操作, 而无需使用循环, 这大大提高了计算效率。

(3) NumPy 的广播机制允许对不同形状的数组进行算术运算, 简化了代码并提高了效率。

2. Pandas 中的 Series 和 Python 中的列表有什么区别?

答案:

Pandas 中的 Series 和 Python 中的列表(list)虽然在某些方面有相似之处, 但它们在功能、用途和底层实现上有许多重要区别。以下是它们的主要区别:

(1) 是 Python 的内置数据结构, 属于动态数组。可以存储任意类型的数据(如整数、浮点数、字符串、对象等)。没有固定的类型限制, 元素类型可以不同。Series 是 pandas 库中的一维数据结构, 类似于表格中的一列。底层基于 NumPy 数组, 通常存储同类型的数据(如全部是整数、浮点数或字符串)。

如果存储不同类型的元素, 会被强制转换为统一的类型(通常是 object 类型)。

(2) Python 列表只能通过整数索引(从 0 开始)访问元素, 不支持自定义索引。Series 默认索引是整数索引(从 0 开始), 但可以自定义索引, 支持通过索引标签快速访问数据, 类似于字典的键值对。

(3) Python 列表主要用于存储和操作简单的数据集, 并提供了一些基本操作, 如添加、删除、排序等。Series 专为数据分析设计, 支持更多高级操作, 如数据筛选、聚合、缺失值处理等。

3. 如何使用 sklearn 中的线性模型进行非线性函数的拟合?

答案:

Sklearn 支持多元拟合, 可以通过非线性变换, 将需要拟合的目标公式线性化, 转化为线性形式进行拟合。例如使用 $y = ax + bx^2 + cn$ 拟合 $y = \sin(x)$ 时, 可以用以下代码实现:

```
1. import numpy as np
2. import matplotlib.pyplot as plt
3.
4. import sklearn.linear_model as skl
5.
6. # Origin Data with random err
```

```

7. xOrigin=np.linspace(0, np.pi/2.0, 11)
8. yOrigin=np.sin(xOrigin)+(np.random.rand(11)- 0.5)* 0.5
9.
10. # Linear Model
11.
12. regrModel=skl.LinearRegression()
13.
14. # Train
15. xIn1 =xOrigin.reshape(- 1,1)
16. xIn2=xIn1* * 2
17.
18. xIn=np.concatenate((xIn1, xIn2), axis=1)
19. yIn=yOrigin.reshape(- 1, 1)
20.
21. regrModel.fit(xIn, yIn)
22.
23. # Test
24.
25. yOut=regrModel.predict(xIn)
26.
27. plt.plot(xOrigin, yOrigin, ' o' )
28. plt.plot(xIn1, yOut, ' - - ' )

```

code32. py

运行结果如图：

