

Numpy-03

Numpy介绍

- 定义 是一个科学计算库,用于快速进行任意维度数组间运算
- 核心类 ndarray, 可以描述任意维度的数组
- 内存风格
  - ndarray元素是内置, 速度快, 但是要求元素类型必须一致
  - python list 元素外置, 速度慢, 但是可以存储任意类型的数据
- 优势
  - ndarray支持向量化并行运算
  - 底层是使用C语言实现, 避免GIL锁限制

N维数组

- 属性
  - 形状: shape(重要)
  - 维度: ndim
  - 元素个数: size
  - 元素长度(字节): itemsize
  - 元素数据类型: dtype(重要)
- 数据类型
  - 1. 指定数据类型语法格式: np.array([], dtype)
    - 整数: 默认 np.int64
    - 浮点数: 默认: np.float64
    - 字符串: np.string\_ (多使用)

基本操作

- 生成数组的方法
  - 生成0和1的数组
    - 根据形状生成全1的数组: np.ones(shape, dtype)
    - 根据已知数组的形状生成全1的数组: np.ones\_like(arr[], dtype)
    - 根据形状生成全0的数组: np.zeros(shape, dtype)
    - 根据已知数组的形状生成全0的数组: np.zeros\_like(arr[], dtype)
  - 从现有数据生成
    - np.array() 深拷贝
    - np.asarray() 类似浅拷贝
  - 生成固定范围的数组
    - 生成等间隔数组(重点): np.linspace(start, end, nums, endpoint=True)
    - 根据起始值,结束值以及步长生成数组(类似于Python的range方法)(重点): np.arange(start, end, step)
    - 从10^start, 10^end生成num个等比序列: np.logspace(start, end, nums)
  - 均匀分布
    - 定义 每个数出现概率的都是一样
    - API: np.random.uniform(low, high, size)
      - low: 下限
      - high: 上限
      - size: 生成数组的形状
  - 正态分布
    - 概念
      - 1. 均值决定了: 正态分布的位置
      - 2. 方差决定了: 正态分布形状, 瘦高(方差小), 矮胖(方差大)
      - 3. 方差与标准差意义: 决定数据离散程度; 方差越小数据越集中, 方差越大数据越离散
    - 标准正态分布 均值为0, 方差为1的正态分布
    - API np.random.normal(loc, scale, size)
      - loc: 均值
      - scale: 标准差
      - size: 形状
- 数组的索引和切片
  - 切片: [start:end, start:end, ...]
  - 索引: [值1, 值2, ...]
- 形状修改
  - 返回相同元素新形状ndarray: 对象.reshape(新形状)
  - 修改原数组的形状: 对象.resize(新形状)
  - 数组转置: 行列互换 对象.T
- 类型修改
  - 获取新类型的ndarray: 对象.astype(新类型)
  - 把ndarray转为二进制数据: 对象.tostring()
- 数组去重
  - API: np.unique(arr)
  - 返回值: 去重后的一维数组.

ndarray运算

- 逻辑运算
  - 比较运算符: >, >=, <=, <, ==, !=
  - 原理: 数组每一个元素与指定数进行比较
  - 修改满足条件数据: 数组[条件] = 值
- 通用判断函数
  - np.all(): 全True, 才True
  - np.any(): 只要有True就是True
- np.where
  - np.where(条件, 值1, 值2) 条件对应为True, 这个位置结果就是值1, 否则结果是值2
  - 逻辑与(并且) np.logical\_and(条件1, 条件2)
  - 逻辑或(或者) np.logical\_or(条件1, 条件2)
- 统计运算
  - 最小值: 对象.min(axis) axis=0, 按列算, axis=1 按行算, 如果没有指定就求所有元素的最小值
  - 最大值: 对象.max(axis)
  - 中位数: np.median(), 注意;只有np上有这个方法, 但是对象上没有这个方法
  - 中位数: 概念: 一组数据按从小到大顺序排列, 如果是元素个数是奇数中间那个数, 如果元素个数是偶数那就是中间那个数的平均值
  - 平均值: mean
  - 方差: var
  - 标准差: std
  - 最大值所在的索引: argmax
  - 最小值所在的索引: argmin
- 矩阵
  - 矩阵: 二维数组
  - 向量: 一维数组, 默认一般列向量
  - 矩阵加法: 对应位置的元素相加
  - 矩阵与标量相乘: 标量和矩阵中每一个元素相乘
  - 矩阵相乘\* 规则: (M行, N列) × (N行, L列) = (M行, L列)
  - 矩阵乘法性质:
    - 1. 满足结合律
    - 2. 不满足交换律
  - 单位矩阵:
    - 主对角线上元素全为1, 其他位置的元素全为0
    - 主对角线: 从左上角到右下角的那条线
    - 性质: 单位矩阵乘以任何矩阵还是这个矩阵.
  - 矩阵的逆: A × A的逆 = A的逆 × A = 单位矩阵
  - 矩阵转置: 行列互换.
- 数组间运算
  - 1. 数组与数的运算
    - 支持运算: +, -, \*, /
    - 数组中每个元素与数进行运算
  - 2. 数组与数组的运算
    - 广播机制:
      - 1. 维度一样
      - 2. shape对应位置参与运算一方为1
  - 3. 矩阵运算
    - 矩阵乘法
      - np.matmul(a, b)
      - np.dot(a, b)
    - 相同点: 都可以进行矩阵乘法
    - 不同点:
      - matmul: 不能进行矩阵与标量的运算
      - dot: 可以进行矩阵与标量的运算