

# Numpy基础

---

## 1.理解矢量化操作

- python之所以流行，是因为写程序确实方便。执行python程序的时候，程序交给python解释器执行。开发人员并不需要处理内存分配和清理等工作。
- python属于高级语言，书写快，但是执行慢；C语言属于低级语言，书写慢，但是执行快。
- Pandas和numpy能实现程序书写快，执行也快。主要的原因就是使用矢量化操作
- 矢量化利用了一个被称为单指令多数据(SIMD)的处理器特性来更快地处理数据

## 2. 纽约机场出租数据

- pickup\_year: 行程年份
- pickup\_month: 行程月份
- pickup\_day : 行程日
- pickup\_location\_code: 接驳地点
- dropoff\_location\_code : 到达地点
- trip\_distance: 行程距离
- trip\_length: 行程花费时间
- fare\_amount: 行程费用
- total\_amount: 总费用

[nyc\\_taxi.csv下载地址](#)

```
1  import csv
2  import numpy as np
3
4  # import nyc_taxi.csv as a list of lists
5  f = open('nyc_taxi.csv', 'r')
6
7  # 去除首行
8  taxi_list = list(csv.reader(f))[1:]
9
10 # 将里面的数值转化为浮点型
11 converted_taxi_list = []
12 for row in taxi_list:
13     converted_row = []
14     for each in row:
15         converted_row.append(float(each))
16     converted_taxi_list.append(converted_row)
17
18 # 生成ndarray对象
19 taxi = np.array(converted_taxi_list)
20
```

## 3. 理解Nddarray对象

- ndarray对应'N-dimensional array' --N维数组
- 主要处理1维数组和2维数组。
- Narray里面的每个元素不一定是相同的类型
- **ndarray属性**

```
1 # 定义一个元组, (x, y)
2 # 其中x表示数组有多少行, y表示数组有多少列
3 taxi.shape()
```

- **行切片**

```
1 # 行切片
2 taxi[1:]
```

```
1 taxi_ten = taxi[:10]
2 print(taxi_ten)
```

## 4. Narray元素选择和切片

- 通用公式

```
1 # 通用公式
2 taxi[row, column]
3 # row 和column可以是单个数字, 也可以是一个切片, 也可以是一个列表比如【1, 3, 7】
4 # 比如row-->3:5, column-->4,6, 表示选择3-5行, 4-6列的数据
```

```
1 # 选择第一行
2 row_0 = taxi[0]
3 # 选择391到500行 (包括500行)
4 rows_391_to_500 = taxi[391:501]
5 # 选择21行5列的元素
6 row_21_column_5 = taxi[21,5]
7 # 示例
8 taxi[1:3, :3]
9 # 示例
10 taxi[0:3, 2:7]
```

## 5. 矢量计算

- numpy比python list 快30倍。
- 矢量计算：

```
1 import numpy as np
2 # 现有列表
3 my_numbers = [
4     [6, 5],
5     [1, 3],
```

```

6         [5, 6],
7         [1, 4],
8         [3, 7],
9         [5, 8],
10        [3, 5],
11        [8, 4]
12    ]
13    # 计算求和
14    # 1 先将list of list转化为numpy ndarray
15
16    data = np.array(my_numbers)
17    # 2 获取第一列和第二列的为1D数组
18    column_01 = data[:, 0]
19    column_02 = data[:, 1]
20    # 第一列和第二列求和
21    sum = column_01 + column_02

```

- 矢量计算可以使用加减乘除

## 6. Numpy算术运算

[numpy常用算术方法链接](#)

- 1D数组常见计算

```

1  # 最大值
2  trip.max()
3  # 最小值
4  trip.min()
5  # 平均值
6  trip.mean()
7  # 求和
8  trip.sum()

```

- 2D数组常见计算

```

1  # 假如需要计算所有的数据
2  ndarray.max() # --->> 返回一个数，所有数据中的最大值
3  # 假如需要计算每一行的最大值
4  ndarray.max(axis=1) # --->>返回一个1D数组
5  # 计算每一列的最大值
6  ndarray.max(axis=0) # --->> 返回一个1D数组

```

## 7. 添加行或者列到数组

- 两个数组要相加，shape必须一致

```

1  # 可以使用expand_dim()函数更改shape
2  zeros_2d = np.expand_dims(zeros, axis=0)
3  # 使用np.concatenate()函数实现合并
4  combined = np.concatenate([ones, zeors_2d], axis=0)

```

## 8. 数组排序

- 使用np.argsort()函数，np.argsort()返回的是一个该数组排序的一个索引。比如：

```
1 # 对一下数组对象进行排序
2 fruit = np.ndarray(['orange', 'banana', 'apple', 'grape', 'cherry'])
3 # 排序, 返回一个索引
4 sorted_order = np.argsort(fruit)
5 # 获取排序后的数组
6 sorted_fruit = fruit[sorted_order]
```