Numpy库

NumPy是 Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

1. 使用numpy生成数组，得到ndarray的类型

```
import numpy as np
import random

# 方法1:
t1 = np.array([1, 2, 3])
print(t1)
print(type(t1))

# 方法2:
t2 = np.array(range(10))
print(t2)
print(type(t2))

# 方法3:
t3 = np.arange(4, 10, 2)
print(t3)
print(type(t3))
```

2. numpy中的数据类型

```
# numpy创建ndarray同时指定数据类型
t4 = np.array(range(1, 4), dtype=float)  # dtype=float  dtype="float32" 都行
print(t4)

# 查看ndarray中每一个数据的类型
print(t4.dtype)

# numpy中的布尔类型
t5 = np.array([1, 1, 0, 0, 0], dtype=bool)
print(t5)
print(t5.dtype)

# 手动改变创建好的数组的数据类型
t6 = t5.astype("int8")
print(t6)
print(t6.dtype)

# numpy中的小数
t7 = np.array([random.random() for i in range(10)])
print(t7)
print(t7.dtype)

# 小数保留指定的位数
t8 = np.round(t7, 2)
print(t8)
```

3. 数组的形状

数组的形状主要的方法是shape和reshape方法

```
import numpy as np

# shape方法：判断数组是几行几列
a = np.array([[3, 4, 5], [6, 7, 8]])
print(a.shape)      # (2, 3)

# reshape方法将数组转换为指定的维度
a = np.array(range(24))
print(a)    # [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23]
b = a.reshape((2, 3, 4))
print(b)
# b的值如下：即分为两组，每组分别是三行四列
# [[[ 0  1  2  3]
#   [ 4  5  6  7]
#   [ 8  9 10 11]]
#
#  [[12 13 14 15]
#   [16 17 18 19]
#   [20 21 22 23]]]
print(b.shape)      # (2, 3, 4) 因此当数组为三维时，三个数分别代表：组，行，列

# 将数组转换为一维数组的两种方式
# 1.reshape方法
c = np.array(range(12)).reshape(3, 4)
print(c)    # c是一个二维数组
d = c.reshape(12,)
print(d)    # 此时d即为一维数组，上面参数的12是3*4得来的，这里需要注意的是，不能传入(1, 12)来进行转换，否则会多一层[]

# 2.flatten方法
e = c.flatten()
print(e)
```

4. 数组的计算和numpy数组中的轴

首先任何维度的数组和数字计算是没有问题的，加减乘除都可以，会对多维数组的每一个元素都和数进行相应的运算。若二维数组间进行计算，需要这两个数组间有一个维度是相同的，才可以进行计算，比如一个2行6列的数组和一个2行1列的数组进行计算会对每一个列进行相关的行操作。三维数组的计算则需要有一个面相同，即两个维度是相同的，比如（3，3，2）和（3，2）两个数组可以计算。

numpy数组中的轴
在numpy中可以理解为方向,使用0,1,2...数字表示,对于一个一维数组,只有一个0轴,对于2维数组(shape(2,2)),有0轴和1轴,对于三维数组(shape(2,2, 3)),有0,1,2轴。

5. numpy读取数据，以及数据的索引、切片、转置

```
import numpy as np

us_file_path = "./youtube_video_data/US_video_data_numbers.csv"
uk_file_path = "./youtube_video_data/GB_video_data_numbers.csv"
```

```
# delimiter是文件中的分隔符，dtype指定读取后的数据类型，unpack表示将读取后的数据转置
# t1 = np.loadtxt(us_file_path, delimiter=",", dtype="int", unpack=True)
t2 = np.loadtxt(us_file_path, delimiter=",", dtype="int")

print(t2)
print(t2[2]) # 取第三行
print(t2[2:]) # 取连续的多行
print(t2[[2, 8, 10]]) # 取不连续的多行，记住多加一个[]

# 取行数据另一种形式，逗号前面的表示行，后面的表示列。列取“:” 说明不管列，只对行做出约束
print(t2[1,:])
print(t2[2:,:])
print(t2[[2,10,3],:])
print(t2[:,0]) # 取列
print(t2[:,2:]) # 取连续的多列
print(t2[:,[0,2]]) # 取不连续的多列

# 取行和列：取第3行，第四列的值
a = t2[2,3]
print(a)
print(type(a))

# 数组的转置的几种方式,t是数据
# 1. t.transpose
# 2. t.swapaxes(1,0)
# 3. t.T
```

6. 将nan处理为平均值

```
import numpy as np
# 该方法将数组中的nan取列的平均值
def fill_ndarray(t1):
    for i in range(t1.shape[1]):    # 遍历每一列
        temp_col = t1[:, i]         # 当前的一列
        nan_num = np.count_nonzero(temp_col != temp_col)    # nan的个数
        if nan_num != 0:            # 不为0，说明该列中有nan
            # print(temp_col[temp_col == temp_col])  [ 2. 10.]、[ 3. 11.]
            # print(temp_col[temp_col != temp_col])   [nan]、[nan]

            # 取当前一列不为nan的其他数，如果这一列是[2, nan, 10]，那么结果就是[2, 10]
            temp_not_nan_col = temp_col[temp_col == temp_col]

            # 选中当前的nan的位置，把均值赋值给nan
            temp_col[np.isnan(temp_col)] = temp_not_nan_col.mean()
    return t1

if __name__ == '__main__':
    t1 = np.array(range(12)).reshape(3, 4).astype("float")
    t1[1, 2:] = np.nan
    print(t1)
    t1 = fill_ndarray(t1)
    print(t1)
```