

实验 3

NumPy机器学习库

一、实验目的

1. 掌握 NumPy 机器学习库的使用。
2. 掌握 ndarray 数组的创建方法。
3. 掌握 ndarray 数组的常用计算方法。

二、实验内容

1. 创建数组

- (1) 创建一个 4 行 5 列的全 1 数组 arr1,将类型定义为 int,并调整数组的大小为 5 行 4 列。
- (2) 生成 0~5、步长为 0.2 的数组 arr2。
- (3) 使用随机函数创建数组 arr3,数组为 4 行 5 列、数值在[1,5)范围内、满足正态分布的随机数组。

结果如下：

```
arr1:
[[1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]]
arr2:
[[0.  0.2 0.4 0.6 0.8]
 [1.  1.2 1.4 1.6 1.8]
 [2.  2.2 2.4 2.6 2.8]
 [3.  3.2 3.4 3.6 3.8]
 [4.  4.2 4.4 4.6 4.8]]
arr3:
[[1.19441258 3.50538082 2.59865547 2.57314607 4.54954027]
 [2.90148267 2.64995707 1.76515199 3.42691343 4.67548853]
 [3.66184558 4.42726054 3.03596664 3.76324214 3.91804399]
 [4.82643848 4.46458572 3.51332447 1.60390088 2.84252714]]
```

2. NumPy 数组条件运算

代码补全：现有一幅 8×8 大小的灰度图，如图 3-1 所示。图 3-1(a) 是原图，各像素的灰度值如图 3-1(b) 所示。

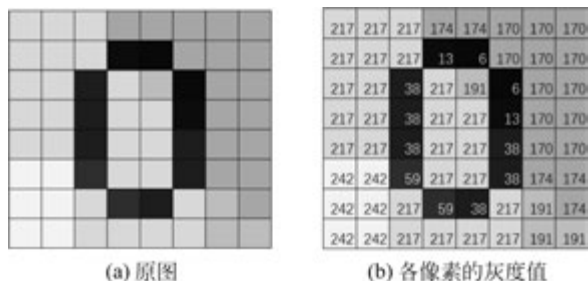


图 3-1 待二值化的灰度图

- (1) 将图像数据保存成一维 ndarray 数组 arrImg 中。
- (2) 对 arrImg 数组进行处理，完成图像的二值化，规则是将灰度值小于 128 的像素转化为 1(黑色)、灰度值大于或等于 128 的像素转化为 0(白色)。

补全代码，完成以上要求。

```
import numpy as np
arrImg=np.array([
217,217,217,174,174,170,170,170,
217,217,217,13,6,170,170,170,
217,217,38,217,191,6,170,170,
217,217,38,217,217,13,170,170,
217,217,38,217,217,38,170,170,
242,242,59,217,217,38,174,174,
242,242,217,59,38,217,191,174,
242,242,217,217,217,217,191,191
])
... 此处代码需要补充完整 ...

print('二值化后的 arrImg 数组为:\n',arrImg)
```

运行结果如下：

```
二值化后的arrImg数组为:
[0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0
 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

可以看出，原灰度图经过二值化处理后转化为黑白图片，效果如图 3-2 所示。

3. NumPy 数组改变形状

- (1) 使用数字 1~9 构造一维 ndarray 数组，将其变形为 3 行 3 列的二维数组，然后对新数组进行转置。

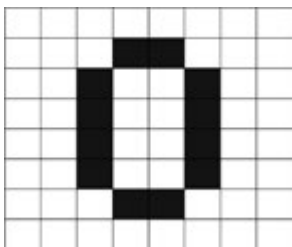


图 3-2 二值化后的图像

```
import numpy as np
arrTest=np.array([1,2,3,4,5,6,7,8,9])
arrTest.shape=_____ (填空)
# 或 arrTest=arrTest.reshape(3,3)
print('数组变形为 3 行 3 列后: \n',arrTest)
print('数组转置后: \n',arrTest.T)
```

运行结果如下：

```
数组变形为3行3列后:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
数组转置后:
[[1 4 7]
 [2 5 8]
 [3 6 9]]
```

(2) 代码补全：

- ① 将图像二值化后的 arrImg 数组变形为 8×8 大小的二维数组。
- ② 将数组进行转置,观察结果。

```
import numpy as np
arrImg=np.array([
0,0,0,0,0,0,0,0,
0,0,0,1,1,0,0,0,
0,0,1,0,0,1,0,0,
0,0,1,0,0,1,0,0,
0,0,1,0,0,1,0,0,
0,0,1,0,0,1,0,0,
0,0,1,0,0,1,0,0,
0,0,0,1,1,0,0,0,
0,0,0,0,0,0,0,0])

... 此处代码需要补充完整 ...

print('将 arrImg 变形为  $8 * 8$  二维数组:\n',arrImg)
# 将 arrImg 进行转置,保存到 newImg 中
newImg=_____ (填空)
print("转置后的数组为: \n",newImg)
```

运行结果如下：

将arrImg变形为8*8二维数组:

```
[[0 0 0 0 0 0 0 0]
 [0 0 0 1 1 0 0 0]
 [0 0 1 0 0 1 0 0]
 [0 0 1 0 0 1 0 0]
 [0 0 1 0 0 1 0 0]
 [0 0 1 0 0 1 0 0]
 [0 0 0 1 1 0 0 0]
 [0 0 0 0 0 0 0 0]]
```

转置后的数组为:

```
[[0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 1 1 1 1 0 0]
 [0 1 0 0 0 0 1 0]
 [0 1 0 0 0 0 1 0]
 [0 0 1 1 1 1 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]]
```

4. NumPy 数组切片

代码补全: 对 newImg 数组进行切片, 保留第 3 行到第 6 行、第 2 列到第 7 列。

```
import numpy as np
newImg=np.array(
[[0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 1, 1, 1, 1, 0, 0],
 [0, 1, 0, 0, 0, 0, 1, 0],
 [0, 1, 0, 0, 0, 0, 1, 0],
 [0, 0, 1, 1, 1, 1, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0]])
```

```
print('newImg 第 3 行到第 6 行、第 2 列到第 7 列的切片为:')
print(newImg[_____(填空)_____])
```

运行结果如下:

```
newImg第3行到第6行、第2列到第7列的切片为:
[[0 1 1 1 1 0]
 [1 0 0 0 0 1]
 [1 0 0 0 0 1]
 [0 1 1 1 1 0]]
```

5. NumPy 数组的统计计算

代码补全: 小美计划本周锻炼身体, 已知她的体重为 60kg, 跑步速度是 8km/h。小美一周的跑步时间和食物热量如表 3-1 所示。

表 3-1 小美一周的跑步时间和食物热量

	星期一	星期二	星期三	星期四	星期五	星期六	星期日
跑步时间/min	45	0	40	0	35	50	30
每天食物的热量/cal	1300	1200	1400	1200	1500	1600	1700

(1) 跑步消耗的热量=体重(kg)×运动时间(小时)×指数 K,其中指数 $K=30 \div 3$, 计算小美每天跑步消耗的热量,保留一位小数。

(2) 将每天食物的热量减去跑步消耗的热量,计算小美每天留存的摄入热量。

(3) 计算小美一周留存的摄入热量总和。

```
import numpy as np
weight=60
K=30/3
t=np.array([45,0,40,0,35,50,30])           # 跑步时间
foodCalor=np.array([1300,1200,1400,1200,1500,1600,1700]) # 食物热量

runCalor = np.around(weight * t/60 * K,1)    # 计算跑步消耗的热量

# 1. 输出标题
print("=====燃烧你的卡路里=====")
print(30 * "#")
print("食物的热量:", _____, '卡路里')    # 食物的热量
print("跑步消耗热量:", np.around(_____,1), '卡路里') # 跑步消耗的热量
CalorLeft=foodCalor-runCalor
print("留存的热量:", CalorLeft, '卡路里')      # 显示留存的摄入热量

# 2. 构造二维数组
tmp=np.zeros(_____ )
tmp[0]=foodCalor
tmp[1]=runCalor
tmp[2]=CalorLeft
# 3. 计算一周热量总和
print(30 * "#")
print('本周汇总(食物热量|运动消耗热量|留存热量): \n', _____)
# 显示 tmp 中各项热量之和
```

运行结果如下:

```
=====燃烧你的卡路里=====
食物的热量: [1300 1200 1400 1200 1500 1600 1700] 卡路里
跑步消耗热量: [450  0 400  0 350 500 300] 卡路里
留存的热量: [ 850 1200 1000 1200 1150 1100 1400] 卡路里
=====
本周汇总 (食物热量|运动消耗热量|留存热量):
[9900 2000 7900]
```