

AI VIET NAM – COURSE 2024

Numpy - Exercise

Ngày 5 tháng 7 năm 2024

Giới thiệu về bài tập: Ở phần bài tập này các bạn sẽ được ôn tập cách sử dụng thư viện numpy để thao tác trên 1D và 2D array. Dựa vào đó để giải quyết các bài toán về xử lý ảnh như chuyển ảnh màu thành ảnh xám và phân tích dữ liệu dạng bảng.

Bài tập 1: Numpy Cơ Bản

Câu hỏi 1: Câu nào sau đây là đúng để tạo mảng 1 chiều từ 0 đến 9

- a) `import numpy as np`
`arr = np.arange(0, 10, 1)`
- b) `import numpy as np`
`arr = np.arange(1, 10, 1)`
- c) `import numpy as np`
`arr = np.arange(0, 9, 1)`
- d) `import numpy as np`
`arr = np.arange(1, 11, 1)`

Câu hỏi 2: Cách tạo một mảng boolean 3x3 với tất cả giá trị là True

- a) `arr = np.ones((3,3)) > 0`
- b) `arr = np.ones((3,3), dtype=bool)`
- c) `arr = np.full((3,3), fill_value=True, dtype=bool)`
- d) a, b and c

Câu hỏi 3: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2 arr = np.arange(0,10)
3 print(arr[arr%2 == 1])
```

- a) `[1 3 5 7 9]`
- b) `[1 2 3 4 5]`
- c) `[2 4 6 8 9]`
- d) `[1 7 5 7 9]`

Câu hỏi 4: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2 arr = np.arange(0,10)
3 arr[arr%2 ==1] = -1
4 print(arr)
```

- a) [-1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
- b) [0 -1 2 -1 4 -1 6 -1 8 -1]
- c) [-1 0 -1 -0 -1 0 -1 0 -1 0]
- d) [-1 1 -1 1 -1 1 -1 1 -1 1]

Câu hỏi 5: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2 arr = np.arange(10)
3 arr_2d = arr.reshape(2,-1)
4 print(arr_2d)
```

- a) [[0 1 2 3 4 5 6 7 8 9]]
- b) [[0 1 2 3 4]
[5 6 7 8 9]]
- c) [[0 1 2 3 4]]
- d) [[0 1 2 3]
[5 6 7 8]]

Câu hỏi 6: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2 arr1 = np.arange(10).reshape(2,-1)
3 arr2 = np.repeat(1,10).reshape(2,-1)
4 c = np.concatenate([arr1,arr2],axis =0)
5 print("Result: \n", c)
```

- a) [[0 1 2 3 4]
[5 6 7 8 9]
[1 1 1 1 1]
[1 1 1 1 1]]
- b) [[0 1 2 3 4]
[5 6 7 8 9]]
- c) [[0 1 2 3 4]
[5 6 7 8 9]
[1 1 1 1 1]
[5 6 7 8 9]]
- d) [[0 1 2 3 4]]

Câu hỏi 7: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2 arr1 = np.arange(10).reshape(2,-1)
3 arr2 = np.repeat(1,10).reshape(2,-1)
4 c = np.concatenate([arr1,arr2],axis =1)
5 print("C = ", c)
```

a) $\begin{bmatrix} 0 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\ 5 & 6 & 7 & 8 & 9 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 6 & 7 & 8 & 9 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 1 & 1 & 1 & 1 & 1 \\ 5 & 6 & 7 & 8 & 9 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

d) $\begin{bmatrix} 0 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

Câu hỏi 8: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2 arr = np.array([1,2,3])
3 print(np.repeat(arr,3))
4 print(np.tile(arr,3))
```

a) $\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix}$

d) $\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix}$

Câu hỏi 9: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2 a = np.array([2,6,1,9,10,3,27])
3 index = np.where((a>=5)&(a<=10))
4 print("result", a[index])
```

a) $[3 \quad 9 \ 10]$

b) $[1 \quad 9 \ 10]$

c) $[6 \quad 9 \ 10]$

d) $[2 \quad 1 \ 10]$

Câu hỏi 10: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2
3 def maxx(x,y):
4     if x >= y :
5         return x
6     else:
7         return y
8
9 a = np.array([5,7,9,8,6,4,5])
10 b = np.array([6,3,4,8,9,7,1])
11
12 pair_max = np.vectorize(maxx,otypes=[float])
13 print(pair_max(a,b))
```

- a) [5, 7, 9, 8, 6, 4, 5]
- b) [6, 3, 4, 8, 9, 7, 1]
- c) [6. 8. 9. 8. 7. 7. 5.]
- d) [6. 7. 9. 8. 9. 7. 5.]

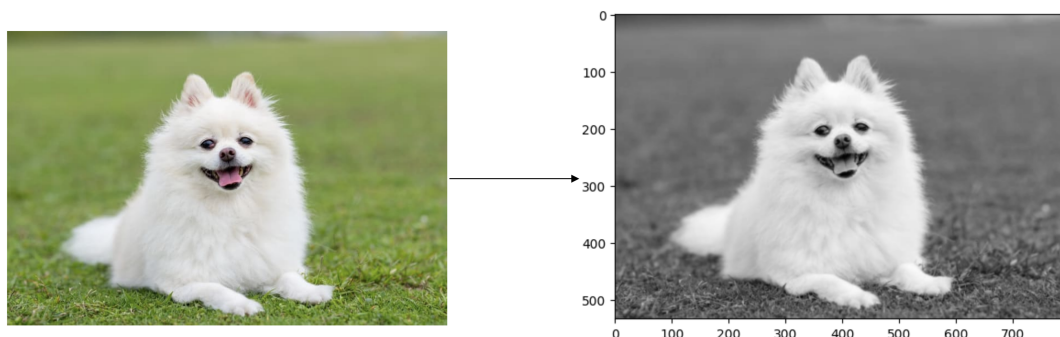
Câu hỏi 11: Kết quả của đoạn code sau đây:

```
1 import numpy as np
2
3 a = np.array([5,7,9,8,6,4,5])
4 b = np.array([6,3,4,8,9,7,1])
5
6 print("Result",np.where(a<b, b, a))
```

- a) [6. 7. 9. 8. 9. 7. 5.]
- b) [5, 7, 9, 8, 6, 4, 5]
- c) [6, 3, 4, 8, 9, 7, 1]
- d) [6. 8. 9. 8. 7. 7. 5.]

Bài tập 2: Xử lý ảnh

Các câu hỏi phần này giải quyết bài toán chuyển ảnh màu về ảnh xám. Ví dụ được mô tả như hình sau:



Hình 1: Chuyển ảnh màu thành ảnh xám

Ảnh sẽ được biểu diễn bởi tập các điểm ảnh (Pixel) có giá trị từ 0 đến 255. Ảnh màu sẽ được biểu diễn thành ma trận có kích thước (Height, Width, Channel); trong đó số Channel là 3 tương ứng là giá trị biểu diễn cho các kênh màu R-Red, G-Green, B-Blue. Ảnh xám sẽ được biểu diễn thành ma trận có kích thước (Height, Width). Vì vậy, để chuyển ảnh màu thành ảnh xám, chúng ta sẽ tổng hợp từ 3 kênh màu RGB thành 1 giá trị duy nhất trong khoảng 0 đến 255. Có 3 phương pháp chính để tính giá trị chuyển đổi:

- Lightness: Tính giá trị trung bình của giá trị lớn nhất và nhỏ nhất cho các kênh màu: $(\max(R,G,B) + \min(R,G,B))/2$
- Average: Tính giá trị trung bình của 3 kênh màu: $(R+G+B)/3$
- Luminosity: Nhân hệ số tương ứng của 3 kênh màu như sau: $0.21*R + 0.72*G + 0.07*B$

Câu hỏi 12: Hoàn thành đoạn code sau đây để chuyển ảnh màu sang ảnh xám dựa vào phương pháp Lightness:

```
1 # Download image
2 !gdown 1i9dqa21DjQoG5Q_VEvm0LrVwA1XD0vB
3
4 import matplotlib.image as mpimg
5 img = mpimg.imread('/content/dog.jpeg')
6 gray_img_01 = # Your Code Here
7 gray_img_01[0, 0]
```

- a) [102.5]
- b) [92.5]
- c) [82.5]
- d) [72.5]

Câu hỏi 13: Hoàn thành đoạn code sau đây để chuyển ảnh màu sang ảnh xám dựa vào phương pháp Average:

```
1 # Download image
2 !gdown 1i9dqa21DjQoG5Q_VEvm0LrVwAlXD0vB
3
4 import matplotlib.image as mpimg
5 img = mpimg.imread('/content/dog.jpeg')
6 gray_img_02 = # Your Code Here
7 gray_img_02[0, 0]
```

- a) [107.7]
- b) [97.7]
- c) [87.7]
- d) [77.7]

Câu hỏi 14: Hoàn thành đoạn code sau đây để chuyển ảnh màu sang ảnh xám dựa vào phương pháp Luminosity:

```
1 # Download image
2 !gdown 1i9dqa21DjQoG5Q_VEvm0LrVwAlXD0vB
3
4 import matplotlib.image as mpimg
5 img = mpimg.imread('/content/dog.jpeg')
6 gray_img_03 = # Your Code Here
7 gray_img_03[0, 0]
```

- a) [96.2]
- b) [116.2]
- c) [126.2]
- d) [136.2]

Bài tập 3: Phân tích dữ liệu dạng bảng

Các câu hỏi phần này ứng dụng numpy trong phân tích dữ liệu dạng bảng, được mô tả gồm các cột và một số hàng như sau:

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

Hình 2: Dữ liệu Advertising dạng bảng.

Dựa vào đoạn code sau để tải về dữ liệu và thực hiện các đoạn code cho các câu hỏi sau để chọn đáp án đúng:

```

1 # Download data
2 !gdown 1iA0WmVfW88HyJvTBSQDI5vesf-pgKabq
3
4 import pandas as pd
5 df = pd.read_csv('/content/advertising.csv')
6
7 data = df.to_numpy()

```

Câu hỏi 15: Lấy giá trị lớn nhất và chỉ mục tương ứng của nó trên cột Sales:

- a) Max: 25 - Index: 30
- b) Max: 27 - Index: 30
- c) Max: 27 - Index: 175
- d) Max: 25 - Index: 175

Câu hỏi 16: Giá trị trung bình của cột TV là:

- a) 146.0
- b) 147.0
- c) 148.0
- d) 149.0

Câu hỏi 17: Số lượng bản ghi có giá trị tại cột Sales lớn hơn hoặc bằng 20 là:

- a) 40

b) 41

c) 42

d) 43

Câu hỏi 18: Tính giá trị trung bình của cột Radio thoả mãn điều kiện giá trị tương ứng trên cột Sales lớn hơn hoặc bằng 15:

a) 25.2

b) 26.2

c) 27.2

d) 28.2

Câu hỏi 19: Tính tổng các giá trị của cột Newspaper sao cho giá trị của nó lớn hơn giá trị trung bình của cột Newspaper:

a) 1403.1

b) 1404.1

c) 1405.1

d) 1406.1

Câu hỏi 20: Gọi giá trị trung bình của cột Sales là A. Tạo ra mảng mới scores chứa các giá trị Good, Average và Bad sao cho: nếu giá trị hiện tại $> A \Rightarrow$ giá trị trong mảng mới là Good, $< A$ thì sẽ là Bad và $= A$ sẽ là Average. Sau đó in ra kết quả scores[7:10]

a) ['Bad ', 'Good ', 'Good ']

b) ['Good ', 'Bad ', 'Good ']

c) ['Bad ', 'Bad ', 'Good ']

d) ['Bad ', 'Bad ', 'Bad ']

Câu hỏi 21: Gọi giá trị trên cột Sales gần nhất với giá trị trung bình cũng chính cột Sales là A. Tạo ra mảng mới scores chứa các giá trị Good, Average và Bad sao cho: nếu giá trị hiện tại $> A \Rightarrow$ giá trị trong mảng mới là Good, $< A$ thì sẽ là Bad và $= A$ sẽ là Average. Sau đó in ra kết quả scores[7:10]

a) ['Bad ', 'Good ', 'Good ']

b) ['Good ', 'Bad ', 'Good ']

c) ['Bad ', 'Bad ', 'Good ']

d) ['Bad ', 'Bad ', 'Bad ']