

3/10/23

Day → 3

Date 129

Numpy → II

Agenda →

- ↳ Element wise Operation
- ↳ Matrix Multiplication
- ↳ Sorting
- ↳ Vectorization
- Broadcasting

a = np.arange(12).reshape(-1, 4)

b = np.arange(12).reshape(-1, -1)

Ambiguity

Sorting → np.sort(a)

↳ Return a new array  
which is sorted & it  
is Not In-place

(ii): a.sort() → In place & sort the original  
array

Element wise Operations

Given a = [1, 2, 3, 4]

a \* 2 → [2, 4, 6, 8]

↳ Not In place Spiral

3/10/23

Date 10.0

$$b = [6, 7, 8, 9, 10]$$

$$a * b = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \end{bmatrix} = [6, 14, 24, 36, 50]$$

Note  $\rightarrow$  Both Array Must have Same Shape

i) Array & Single Element  $\rightarrow \checkmark$

(ii) Arrays have same shape  $\rightarrow \checkmark$

(iii) Arrays with diff shape  $\rightarrow \times$

$\Rightarrow$  Also we can this can applicable on 2-D Array

Matrix Multiplication  $\rightarrow$

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

$2 \times 2$      $m \times n$  equal     $n \times 3$

first elem of 1<sup>st</sup> Row & first elem of 1<sup>st</sup> Col

$$\begin{bmatrix} 21 & 24 & 7 \\ 63 & 54 & 21 \end{bmatrix}$$

$3 \times 3$

we

Interpret it as

No of column in array 1 = No of Rows in

Spiral

Date 19/1

3/10/93

## Different ways to do Matrix multiplication

(1)  $\rightarrow \text{np.dot}(a, b) \rightarrow \text{np.dot}(a, 5) \rightarrow \text{can work}$   
↳ flexible

(2)  $\rightarrow \text{np.matmul}(a, b)$  ]  $\rightarrow a \& b \text{ must have}$   
matrix

(3)  $\rightarrow a @ b$

$\rightarrow a @ 5 \rightarrow \text{Does not work}$

Why There are 8 ways to do same

→ Python introduced np.dot first &  
after that other methods. But Python doesn't  
want to deprecate these methods.

Extra (all o/p context info)

|     |     |     |
|-----|-----|-----|
| 123 | 456 | 626 |
| 100 | 192 | 303 |
| 901 | 392 | 256 |
| 052 | 003 | 122 |

$(4, 3) \leftarrow \text{Image Res}$

$\rightarrow 4 \times 3 \text{ pixels}$

(RGB)

each  $\rightarrow$  pixel  $\rightarrow$  color value

↓

Contain code  
of that cell

apply on all  
cell

My code  $\rightarrow$

if color < 250:  
    color += 50  
else:  
    color -= 50

H/W

Spiral

Date 19/8

5/10/23

So, element wise operation  
we are applying

## Vectorize | Vectorization

↳ without loops code are also

Array → Function → Return Modified Array

I have to use loops

→ send every value of array  
through loop to function

exs \*, +, - → works on element wise  
because numpy supports

But want a function to work on  
every element without loops

⇒ np.vectorize() ← function that works  
on single elements

Returns new function which can work  
with arrays

8/10/23

Date 8/10/23

# Broadcasting →

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 10 & 10 & 10 \\ \hline 20 & 20 & 20 \\ \hline 30 & 30 & 30 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline 10 & 11 & 12 \\ \hline 20 & 21 & 22 \\ \hline 30 & 31 & 32 \\ \hline \end{array}$$

Now -

→ Not Possible

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 10 & 10 & 10 \\ \hline 20 & 20 & 20 \\ \hline 30 & 30 & 30 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline 10 & 11 & 21 \\ \hline 20 & 21 & 21 \\ \hline 30 & 31 & 21 \\ \hline \end{array}$$

→ Sadded

Two not compatible  
to add → Not same  
shape

but Python Try to  
reach the same either horizontally  
or vertically

Broadcast Array to match shape  
to perform operation

ex:-

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 10 & 10 & 10 \\ \hline 20 & 20 & 20 \\ \hline 30 & 30 & 30 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline \end{array}$$

Broadcasting  
Added  
→ Smaller shape

8/10/23

Date 194

Ans of all 3 matrix plus will be same

repetition

now

title  $\rightarrow$  nptitle ([0, 10, 20, 30], (3, 1))

nr

↓  
Repeat

things

repetition

|   |    |    |    |
|---|----|----|----|
| 0 | 10 | 20 | 30 |
| 0 | 10 | 20 | 30 |
| 0 | 10 | 20 | 30 |

$\Rightarrow$  One of array should be 1-D to  
Broadcast

data array of float type

matrix multiplication

Spiral

3/10/28

## Assignment

Date... 195

## (1) → Matrix Element

$$x = \text{np.ones}((5, 5))$$

$$x[1:3, 1:-1] = 0$$

Excluded

|   | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 |

1 to 3 in Row

1 to 3 in Col

→ All elements become zero

## (2) → Mapping in numpy

|    |    |    |    |
|----|----|----|----|
| 2  | 3  | 4  | 5  |
| 3  | 7  | 3  | 5  |
| 2  | 8  | 6  | 9  |
| 11 | 23 | 12 | 19 |

def func(x, y)

return x \* y

vec = np.vectorize(func)

vec(arr1, arr1)

(A) → arr1 = [2, 2, 2, 2] ↓ Broadcasting Vertically  
 $\text{axis} = 0$

(B) → arr1 = [2, 2, 2, 2] ↓ Broadcasting Horizontally  
 $\text{axis} = 1$

(C) → arr1 = [2, 2, 2, 2] ↓ Broadcasting at Vertical & Horizontal Level  
 $\text{axis} = 0$

3/10/23

Date 196

(8)  $\rightarrow$  Specific elements  $\rightarrow$  col of first matrix

$m1 = \text{mat1. shape}[1]$

$m2 = \text{mat2. shape}[0]$

row of second matrix

If ( $m1 \leq m2$ )

return -1

$\Rightarrow \text{a} = \text{m1} \cdot \text{dot}(m1, m2)$

$\Rightarrow : \text{a}[x1:x2, c1:c2]$

(9)  $\rightarrow$  Sort the Birds

We will use argsort

$a = [1, 3, 7, 5, 2]$

$\text{nb.argsort}(a) \rightarrow [0, 4, 1, 3, 2]$

element 1 2 3 5 7

$\hookrightarrow$  Return Index in sorted manner

Name = [-, -, -, -, -]

Age = [-, -, -, -, -]

sort Name to Age

Ind = nb.argsort(Age)

Name[Ind]

Spiral

3/10/23

Date 19/7

Q5 → Comparing in Numpy

(a) → Ram, Astha, Raghavendra

Sort

Astha, Raghavendra, Ram

→ Last

Option 1

b) - arr1(Ram, Astha, Brahma)

] → False

arr2 (Shyam, kalyan, Naveen)

option 2 → (False, False, False)

## Additional Problems

Q1 → Reshape me

np.arange(10,22).reshape(3,4)

Option → A 3x4 matrix filled with 10 to 21 number range

Q2 → Vectorized codes without loop

A) → for i in range(len(arr1)):  
arr[3\*i] = arr[i]\*arr2[i]

loop → Not Vectorized Spiral

Date: 19/8

3/10) Q3

(B)  $\rightarrow$

Q3 & 1 & Q3 & 2

C  $\rightarrow$

Ans 1 = [1, 2, 3, 6, 3, 2]

Ans 2 = [4, 2, 1, 3, 3, 3, 2]

Ans 3 = nbo zeroes (len(Ans 1))

(2)  $\rightarrow$  for i in range(len(Ans 1)):

if (Ans 1[i] > 0):

Ans[i] = -1

else:

Ans 1[i] = 1

(C)  $\rightarrow$  nbowhere (Ans 1, >0, 1, -1)

Vectorised means which will be applied  
on each element without using loop

(3)  $\rightarrow$  Dot Dot Dash  $\rightarrow$

A  $\geq$  [1, 2, 3]  $\quad$  [9, 8, 7]  
 $3 \times 3$   $3 \times 3$

L  $\rightarrow$  1-D Array  $\rightarrow$  NOT Throw  
error

(B)  $\rightarrow$

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$   $\quad$   $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$2 \times 2$

$2 \times 1$

No of cols = 2 = No of Rows = 2

No T, Throw error

Spiral

3/10/23

Date 199

C →  $[1, 2, 3] \rightarrow 1 \times 3$   
 $k = 3$   
mbodot (arr1, k) → NOT Throw error

D →  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$   $[1 \ 1]_{2 \times 2}$  Throw Error  
(D)  
No of cols = 2 ≠ No of Row = 1

(4) → Swappers point A[0, 0:-1]

A →  $\begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & 9 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|} \hline \downarrow & & \\ \hline 3 & 2 & 1 \\ \hline 6 & 5 & 4 \\ \hline 9 & 8 & 7 \\ \hline \end{array}$

Option (2) → Reverse the columns of 2-D Array

(5) → Calculate Age

mask = birds == 6 Cranes

crange-age = age[mask]

mean-age = np.mean(crage-age)

rounding off = np.round(mean age, 2)

Spiral

3/30/23

Date 2/21

⑥  $\rightarrow$  oldest bird

max-age-index =  $\text{mboarg max}(\text{bird})$

old-bird = birds [max-age-index]

return old-bird

function

PE =  $\frac{\text{PE}}{\text{PE} + \text{P}}$

SI =  $\frac{\text{SI}}{\text{SI} + \text{P}}$

birds for hunting with reasoning < 3 months

old-bird = birds

new-bird = birds

old-bird = birds

old-bird = birds

old-bird = birds

Spiral