

## Broadcasting

$$A = [1 \ 2 \ 3] \quad \text{shape} = (3,) \quad \text{ndim} = 1$$

$$b = 7 \quad \text{shape} = ()$$

não tem  
 shape é  
 um  
 escalar

$$\text{res} = A * b$$

$$\text{res} \rightarrow [7 \ 14 \ 21]$$

multiplicam  
 cada elemento  
 de A por b

Algo análogo ocorreria para qualquer shape de A!

Vamos para algo mais interessante

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \text{shape} = (2, 3) \quad [[1, 2, 3], [4, 5, 6]]$$

$$b = [1 \ 10] \quad \text{shape} = (2,) \quad [1, 10]$$

$$c = [0.1 \ 1 \ 10] \quad \text{shape} = (3,) \quad [0.1, 1, 10]$$

Vamos supor que queremos dois resultados

$$\begin{bmatrix} 1 & 2 & 3 \\ 40 & 50 & 60 \end{bmatrix}$$



Cada linha

Mult. por valor diff

$$\begin{bmatrix} 0.1 & 2 & 30 \\ 0.4 & 5 & 60 \end{bmatrix}$$



Cada coluna

Mult. por valor diff

## Regras do broadcasting

mp.ndarray ou Tensor

- 2 arrays : a e b
- operador : +, -, \*, ÷

Ex: • a.shape = (5, 6) b é float

• a.shape = (3, 4) b.shape = (4, )

Regras: ① Alinha a shape a direita  
② Verifica se dimensões das compatíveis,

ou → Mesmo valor  
ou → Uma delas é 1

$$a = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & + & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \quad \text{shape: (3, 4)}$$

$$b = [10 \ 20 \ 30 \ 40] \quad \text{shape: (4, )}$$

operador: +

① Alinha a direita

(2)

$$\begin{array}{c} 3 \quad 4 \\ \hline 3 \quad 4 \end{array} \leftarrow \begin{array}{l} \text{a tem shape (3, 4)} \\ \text{b tem shape (4)} \end{array}$$

3, 4 ← saída tem shape (3, 4)

$$a = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

$$b = [10 \ 20 \ 30 \ 40]$$

II I 3, 4  
I 3, 4 } definindo  
o formato da saída

- I) Dimensões botem, entras não é alterado
- II) Segundo array e' 1, entras sera espalhados nesse dimensão

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} + \begin{bmatrix} 10 & 20 & 30 & 40 \\ 10 & 20 & 30 & 40 \\ 10 & 20 & 30 & 40 \end{bmatrix} \xrightarrow{\text{O array e' copiado na dim. do broadcast}}$$

$$= \begin{bmatrix} 11 & 22 & 33 & 44 \\ 15 & 26 & 37 & 48 \\ 19 & 30 & 41 & 52 \end{bmatrix}$$

! Essa operação é implícita, no verdade o código é otimizado para prover isso sem de fato criar a matriz intermediária!

## Exemplos

$$1) a = [[1, 2, 3, 4], \leftarrow \text{shape: } (2, 4) \\ [5, 6, 7, 8]]$$

$$b = [[1] \leftarrow \text{shape: } (2, 1) \\ [10]]$$

$$a * b = ?$$

- ① Alinha a direita
- ② Dimensões compatíveis?

↳ Igualis  
↳ Unas e = 1

linhas coluna  
2, 4  
2, 1 ← expansão das colunas  
2, 4 ← shape da saída!

$$\begin{array}{c} a \\ b(\text{implícito}) \end{array} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 1 \\ 10 & 10 & 10 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 50 & 60 & 70 & 80 \end{bmatrix}$$

Outra interpretação:

$$\begin{matrix} 2, 4 \\ 2, 1 \\ \hline 2, 4 \end{matrix} \text{ desliza} \quad \begin{matrix} [1 & 2 & 3 & 4] \\ [5 & 6 & 7 & 8] \end{matrix} * \begin{matrix} [1 & 1 & 1 & 1] \\ [10 & 10 & 10 & 10] \end{matrix} = \begin{matrix} [1 & 2 & 3 & 4] \\ [50 & 60 & 70 & 80] \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 10 \end{bmatrix} \begin{matrix} \nearrow \\ * \\ \searrow \end{matrix}$$

2)  $a = [[1, 2, 3, 4], [5, 6, 7, 8]]$ , shape = (2, 4)

~~$\begin{matrix} 2, 4 \\ 4, 1 \\ \hline 4 \end{matrix}$~~  expande

mas é possível  
fazer o broadcast  
(quebra a regra ②)

$$b = [[0.1], [1], [10], [100]] \quad \text{shape} = (4, 1)$$

3)  $a = [[1, 2, 3, 4], [5, 6, 7, 8]]$ , shape = (2, 4)

$$b = [0.1 \ 1 \ 10 \ 100] \quad \text{shape} = (4, 1)$$

expande  
 $\begin{matrix} 2, 4 \\ 1, 4 \\ \hline 2 \ 4 \end{matrix}$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} * \begin{bmatrix} 0.1 & 1 & 10 & 100 \\ 0.1 & 1 & 10 & 100 \end{bmatrix}$$

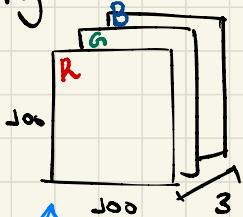
$$= \begin{bmatrix} 0.1 & 2 & 30 & 400 \\ 0.5 & 6 & 70 & 800 \end{bmatrix}$$

4)

$$\text{img.shape} = (3, 100, 100)$$

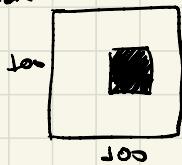
$$\text{mask.shape} = (100, 100)$$

img:



↑  
imagem RGB  
de 100 x 100  
pixels

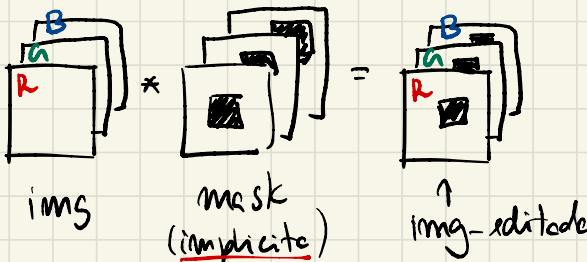
mask:



↑  
mascara me  
imagem (uma  
region que queremos  
apagar na imagem)  
array c/ 0 and queiro  
apagar e 1 no resto.

comet  
(H,W)  
(C,H,W)  
(N,C,H,W)  
↑  
numero  
imags

$$\text{img-edited} = \text{img} * \text{mask}$$



expandindo  

$$\begin{matrix} 3, 100, 100 \\ 1, 100, 100 \\ \hline 3, 100, 100 \end{matrix}$$

## Ejercicio 2

$$a = \begin{bmatrix} 0 & 1 & 2 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} \quad \xrightarrow{?} \quad \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{bmatrix} \quad \leftarrow (3, 4)$$

a: expandir  
b: expandir  
 $\begin{bmatrix} 1 & 4 \\ 3 & 1 \end{bmatrix}$   $\xrightarrow{\alpha}$  man columna  
 $\begin{bmatrix} 3 & 4 \end{bmatrix}$

$$a = \text{mp.array}(4) \leftarrow (4,)$$
$$b = \text{mp.array}(3) \leftarrow (3,)$$

$$b = b.\text{reshape}(3, 1) \leftarrow (3, 1)$$

$$r1s = a + b$$

$$\begin{bmatrix} 0 & 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \quad \downarrow \text{expandir}$$
$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{bmatrix}$$