Prep Course
Module 2
Numpy Broadcasting

Chao-Jen Chen

ccj@uchicago.edu

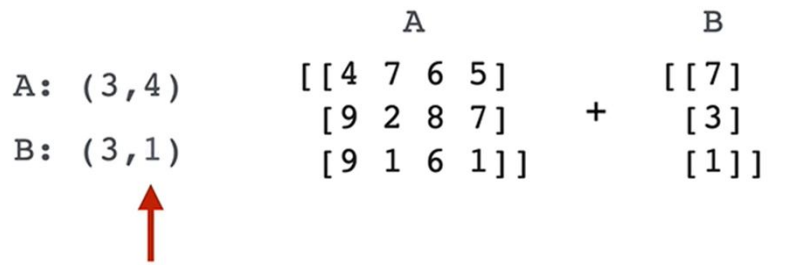
Broadcasting Rules: ***compatible or not?!***

Given two input arrays A and B to a numpy operation or function that performs broadcasting:

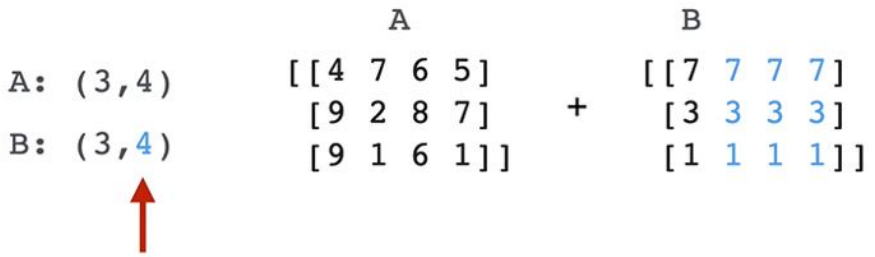
- Moving backwards from the last dimension of each array, we check if their dimensions are compatible.
- Dimensions are compatible if they are equal or either of them is one.
- If all of A's dimensions are compatible with B's dimensions, or vice versa, they are compatible arrays.
- The size of the output array is the size that is not 1 along each dimension of the inputs.

Example 1:

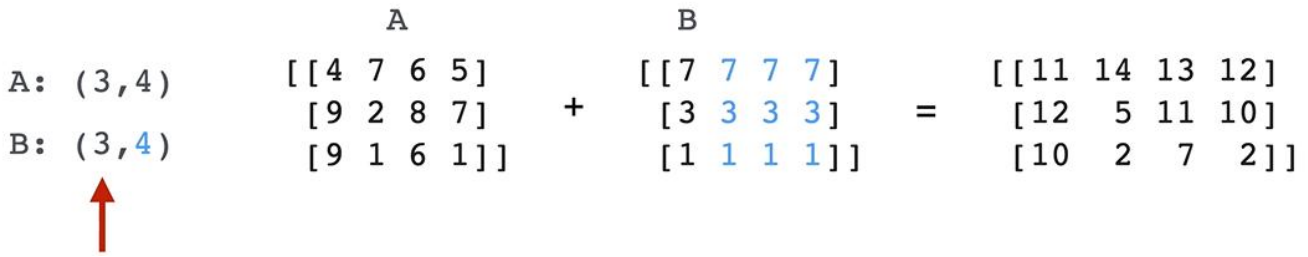
Step 1: Compatible? Yes



Step 2: Match last dimension



Step 3: Match 2nd last dimension



Example 2:

Step 1: Compatible?
No

```
A: (4,4)
B: (2,1)

      A                B
[[3 9 3 2]        [[7]
 [8 6 3 5]    +     [2]]
 [7 1 9 7]
 [6 4 2 2]]
```

Step 2: Even if we can match the last dimension, there is no rule to match the 2nd last dimensions as those dimensions are not equal and both are not one. Thus, a Value Error is thrown out.

```
A: (4,4)
B: (2,4)

      A                B
[[3 9 3 2]        [[7 7 7 7]
 [8 6 3 5]    +     [2 2 2 2]]
 [7 1 9 7]
 [6 4 2 2]]
```

```
ValueError:
operands could not be broadcast
together with shapes (4,4) (2,1)
```

Example 3:

```
1  x = np.arange(4)
2  print(x)
3  print()
4
5  y = np.ones(5)
6  print(y)
7  print()
8
9  print(x.shape)
10 print(y.shape)
11 print()
12
13 print(x + y)
```

```
[0 1 2 3]

[1. 1. 1. 1. 1.]

(4,)
(5,)


---------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-14-c4bc8c28d366> in <module>
     11 print()
     12
---> 13 print(x + y)

ValueError: operands could not be broadcast together with shapes (4,) (5,)
```

Step 1: Compatible?
No

Step 2:, a Value Error is thrown out.

Example 4:

```
1  x = np.arange(4)
2  X = x.reshape(4, 1)
3  print(X)
4  print()
5
6  y = np.ones(5)
7  print(y)
8  print()
9
10 print(X.shape)
11 print(f'( , {y.shape[0]})')
12 print()
13
14 print(X + y)
```

```
[[0]
 [1]
 [2]
 [3]]

[1. 1. 1. 1. 1.]

(4, 1)
( , 5)

[[1. 1. 1. 1. 1.]
 [2. 2. 2. 2. 2.]
 [3. 3. 3. 3. 3.]
 [4. 4. 4. 4. 4.]]
```

Step 1: Compatible?
Yes.

Step 2: Match both dimensions.