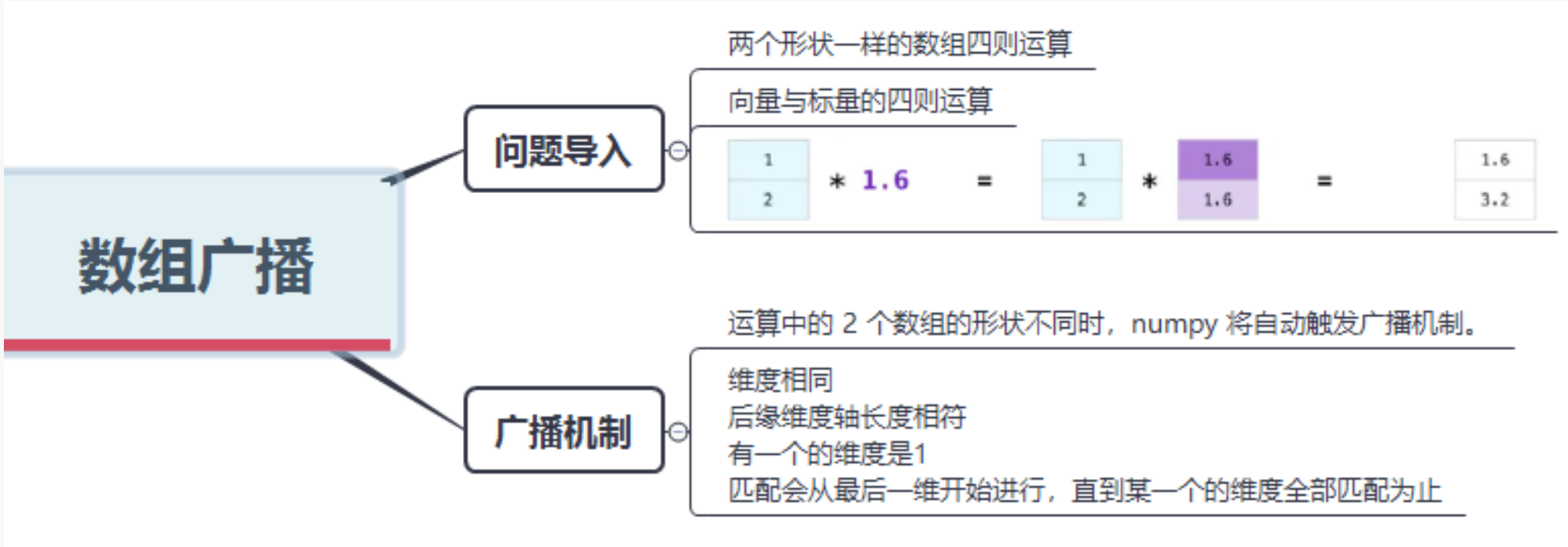
Python科学计算库Numpy之

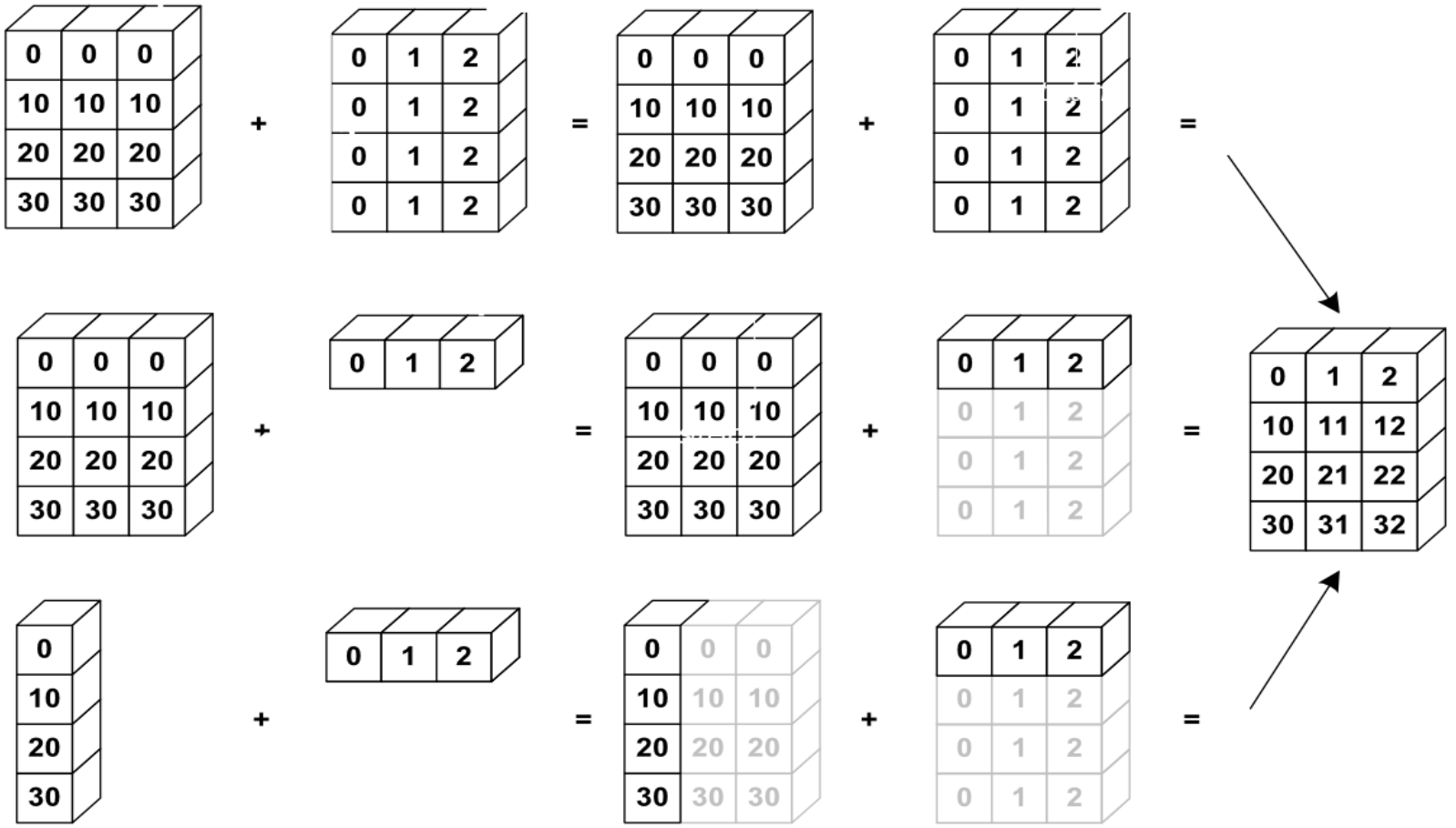
07-数组广播机制

海豚数据科学实验室

知识结构图

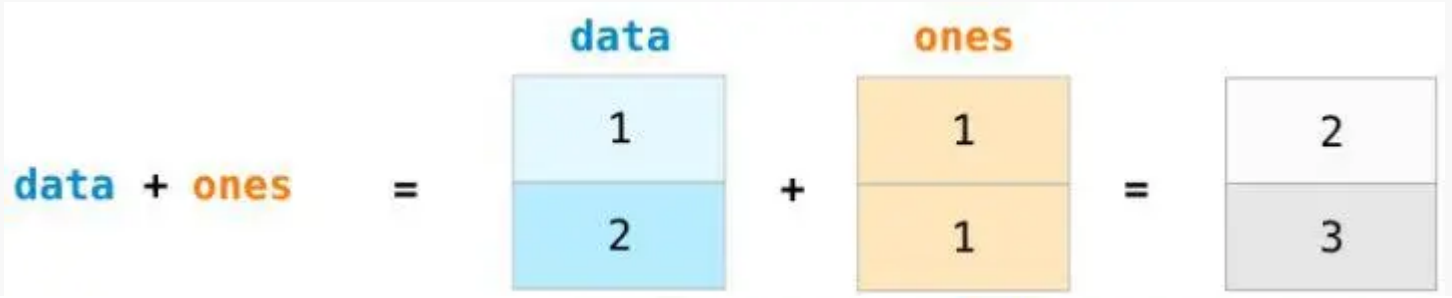

数组广播
问题导入
两个形状一样的数组四则运算
向量与标量的四则运算
1
2
* 1.6
=
1
2
*
1.6
1.6
=
1.6
3.2
广播机制
运算中的 2 个数组的形状不同时，numpy 将自动触发广播机制。
维度相同
后缘维度轴长度相符
有一个的维度是1
匹配会从最后一维开始进行，直到某一个的维度全部匹配为止

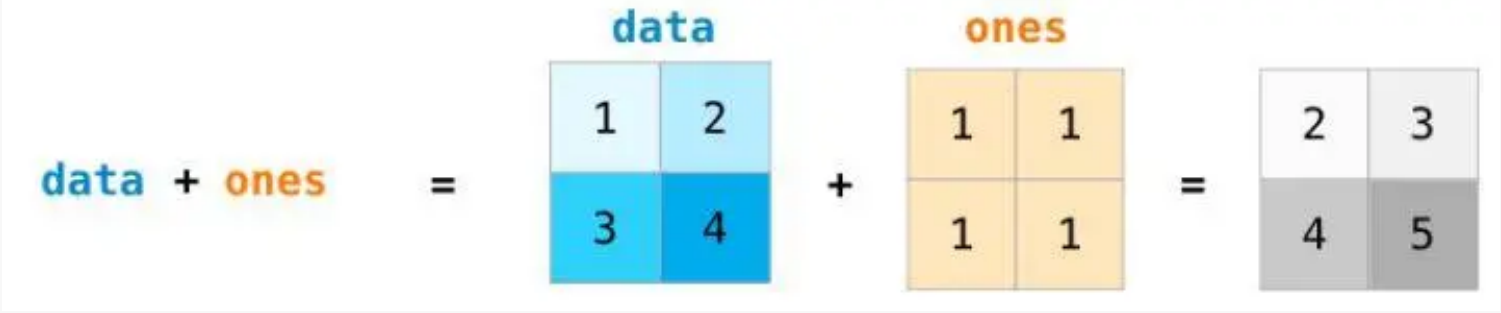
数组广播

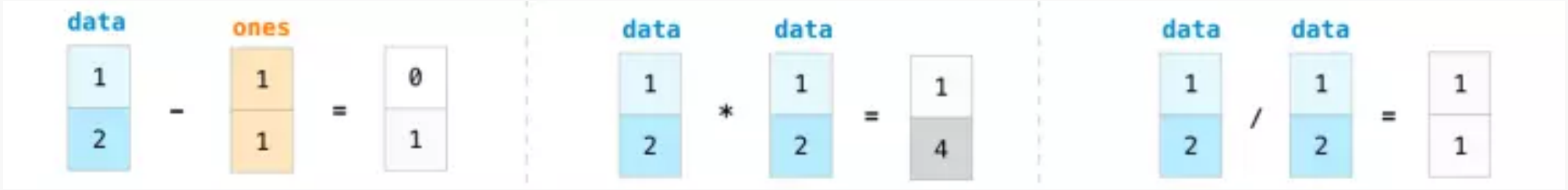

0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
=
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=
0
10
20
30
+
0 1 2
=
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=
0 1 2
10 11 12
20 21 22
30 31 32

数组广播

两个矩阵的大小相同，我们可以使用算术运算符（+ – * /）来进行矩阵计算



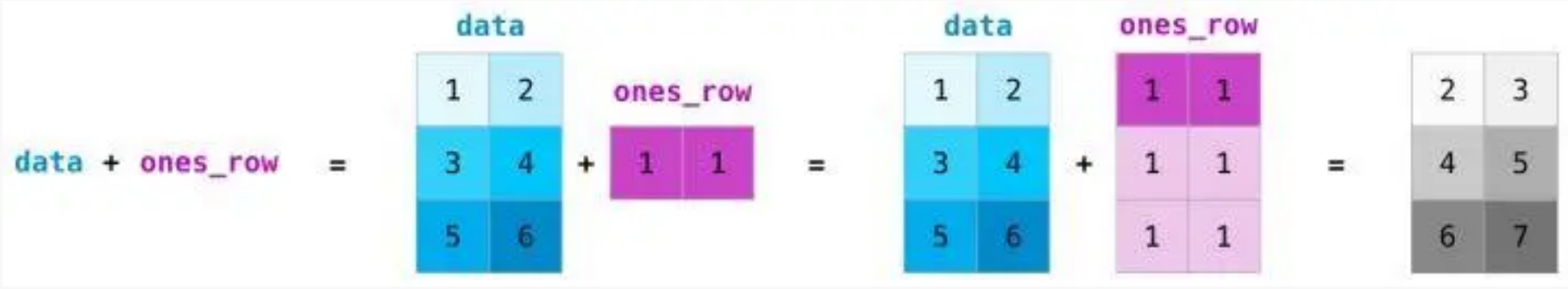




数组广播

向量和标量之间的操作

NumPy让每个单元格都会发生相乘叫做广播



数组广播

如果两个数组 a 和 b 形状相同，即满足 a.shape == b.shape，
那么 a*b 的结果就是 a 与 b 数组对应位相乘。
这要求维数相同，且各维度的长度相同。

当运算中的 2 个数组的形状不同时，numpy 将自动触发广播机制。

广播机制是Numpy让两个不同shape的数组能够做一些运算，
需要对参与运算的两个数组做一些处理或者说扩展，
最终是参与运算的两个数组的shape一样，
然后广播计算(对应位置数据进行某运算)得到结果。

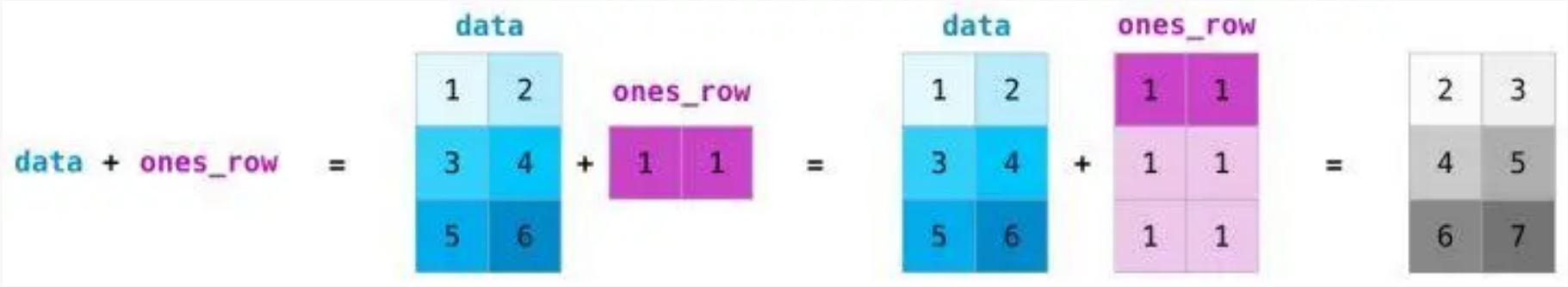
广播机制首先需要判断参与计算的两个数组能否被广播机制处理？
规则是，比较两个数组的shape，从shape的尾部开始一一比对。

(1). 如果两个数组的维度相同，对应位置上轴的长度相同或其中一个的轴长度为1，广播兼容，可在轴长度为1的轴上进行广播机制处理。
(2). 如果两个数组的维度不同，那么给低维度的数组前扩展提升一维，扩展维的轴长度为1,然后在扩展出的维上进行广播机制处理。

数组广播

向量和标量之间的操作

(2 x 1) + (1,) => (2 x 1) + (1 x 1) => (2 x 1) + (2 x 1)



(3 x 2) + (1 x 2) => (3 x 2) + (3 x 2)

数组广播

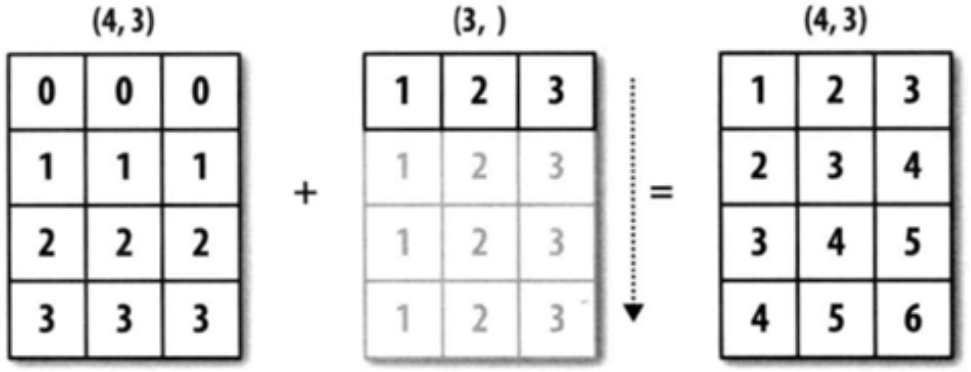

```
arr1 = np.array([[0, 0, 0],
                 [1, 1, 1],
                 [2, 2, 2],
                 [3, 3, 3]])   #arr1.shape = (4, 3)
arr2 = np.array([1, 2, 3])     #arr2.shape = (3, )
```

```
array([[1, 2, 3],
       [2, 3, 4],
       [3, 4, 5],
       [4, 5, 6]])
```

数组维度不同，后缘维度的轴长相符

```
arr1 + arr2
```

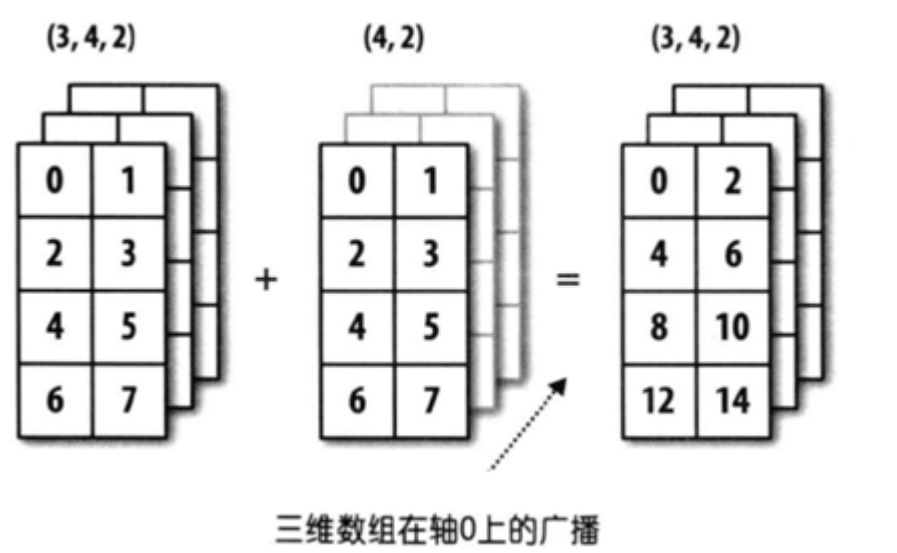
```
array([[1, 2, 3],
       [2, 3, 4],
       [3, 4, 5],
       [4, 5, 6]])
```



一维数组在轴0上的广播

数组广播

数组维度不同，后缘维度的轴长相符



（3,4,2）和（4,2）的维度是不相同的，前者为3维，后者为2维。
但是它们后缘维度的轴长相同，都为（4,2），所以可以沿着0轴进行广播。

数组广播

对于 Numpy 来说，维度匹配当且仅当：

维度相同

后缘维度轴长度相符

有一个的维度是1

匹配会从最后一维开始进行，直到某一个的维度全部匹配为止

A	B	Result
3d array: 256 x 256 x 3	1d array: 3	3d array: 256 x 256 x 3
4d array: 8 x 1 x 6 x 1	3d array: 7 x 1 x 5	3d array: 8 x 7 x 6 x 5
3d array: 5 x 4 x 3	1d array: 1	3d array: 5 x 4 x 3
3d array: 15 x 4 x 13	1d array: 15 x 1 x 13	3d array: 15 x 4 x 13
2d array: 4 x 1	1d array: 3	2d array: 4 x 3

谢 谢!

海豚数据科学实验室