

numpy的broadcast广播机制

维度数量

numpy中指定维度都是用元组来的，比如`np.zeros((2,3,2))`的维度数量是三维的。`np.zeros((3,))`维度数量这是1维的，因为`(3)`不是元组它只能算3加个括号而已，加个逗号`(3,)`才是元组。

某个维度大小

比如`np.zeros((2,3,4))`的维度数量是三维的。这个数组第一维的维度大小是2，第二维的维度大小是3，第三维的维度大小是4。

广播 (broadcasting)

通常只在对多个数组进行对应元素操作形状不同时，才会发生广播。
那什么是对应元素进行操作呢？比如：

```
a = np.array([1,2,3])
b = np.array([2,2,2])
a*b # a和b对应元素相乘
# a*b的结果是： [1*2,2*2,3*2]
'''
np.dot(a,b) # 这就不是对应元素操作，这是矩阵相乘。
# np.dot(a,b)的结果是a,b的点积。
'''

>>> np.dot(a,b)
>>> 12

>>> a @ b
>>> 12

>>> a * b
>>> [2, 4, 6]
```

什么叫做形状不同呢？

```
a = np.array([1,2,3])
b = 2
a*b #a是1维向量，b是标量，这就是形状不同
# 结果也是： [1*2,2*2, 3*2]
'''
这是因为发生了广播。b被填充为[2,2,2]
然后a*b的效果变成了， [1,2,3]*[2,2,2]
'''
```

前面的两个例子输入不同但运行结果相同的原因就是发生的广播(broadcast)。

可以广播的几种情况：

1. 假定只有两个数组进行操作，即A+B、A*B这种情况。

两个数组各维度大小从后往前比对均一致

举个例子：

```
A = np.zeros((2,5,3,4))
B = np.zeros((3,4))
print((A+B).shape) # 输出 (2, 5, 3, 4)

A = np.zeros((4))
B = np.zeros((3,4))
print((A+B).shape) # 输出(3,4)
```

举个反例：

```
A = np.zeros((2,5,3,4))
B = np.zeros((3,3))
print((A+B).shape)
报错：
ValueError: operands could not be broadcast together with shapes (2,5,3,4) (3,3)
为啥呢？因为最后一维的大小A是4，B是3，不一致。
```

2. 两个数组存在一些维度大小不相等时，有一个数组的该不相等维度大小为1

这是对上面那条规则的补充，虽然存在多个维大小不一致，但是只要不相等的那些维有一个数组的该大小是1就可以。

举个例子：

```
A = np.zeros((2,5,3,4))
B = np.zeros((3,1))
print((A+B).shape) # 输出： (2, 5, 3, 4)

A = np.zeros((2,5,3,4))
B = np.zeros((2,1,1,4))
print((A+B).shape) # 输出： (2, 5, 3, 4)

A = np.zeros((1))
B = np.zeros((3,4))
print((A+B).shape) # 输出(3,4)

# 下面是报错案例
A = np.zeros((2,5,3,4))
B = np.zeros((2,4,1,4))
print((A+B).shape)
ValueError: operands could not be broadcast together with shapes (2,5,3,4) (2,4,1,4)
为啥报错？因为A和B的第2维不相等。并且都不等于1。
```