

NumPy - Transmissão

O termo **radiodifusão** refere-se à capacidade do NumPy de tratar matrizes de diferentes formatos durante operações aritméticas. As operações aritméticas em arrays geralmente são feitas nos elementos correspondentes. Se duas matrizes tiverem exatamente o mesmo formato, essas operações serão executadas sem problemas.

Exemplo 1

```
import numpy as np
```

[Demonstração ao vivo](#)

```
a = np.array([1,2,3,4])  
b = np.array([10,20,30,40])  
c = a * b  
print c
```

Sua saída é a seguinte -

```
[10  40  90 160]
```

Se as dimensões de duas matrizes forem diferentes, as operações elemento a elemento não serão possíveis. No entanto, operações em matrizes de formatos não semelhantes ainda são possíveis no NumPy, devido à capacidade de transmissão. A matriz menor é **transmitida** para o tamanho da matriz maior para que tenham formas compatíveis.

A transmissão é possível se as seguintes regras forem satisfeitas -

- Matriz com **ndim** menor que a outra é anexada a '1' em seu formato.
- O tamanho em cada dimensão da forma de saída é o máximo dos tamanhos de entrada nessa dimensão.
- Uma entrada pode ser usada no cálculo, se seu tamanho em uma dimensão específica corresponder ao tamanho de saída ou se seu valor for exatamente 1.
- Se uma entrada tiver tamanho de dimensão 1, a primeira entrada de dados nessa dimensão será usada para todos os cálculos nessa dimensão.

Diz-se que um conjunto de matrizes pode ser **transmitido** se as regras acima produzirem um resultado válido e uma das seguintes afirmações for verdadeira -



- As matrizes têm exatamente o mesmo formato.
- As matrizes têm o mesmo número de dimensões e o comprimento de cada dimensão é um comprimento comum ou 1.
- Uma matriz com poucas dimensões pode ter sua forma anexada a uma dimensão de comprimento 1, de modo que a propriedade declarada acima seja verdadeira.

O programa a seguir mostra um exemplo de transmissão.

Exemplo 2

```
import numpy as np
a = np.array([[0.0,0.0,0.0],[10.0,10.0,10.0],[20.0,20.0,20.0],[30.0,30.0,30.0]])
b = np.array([1.0,2.0,3.0])

print 'First array:'
print a
print '\n'

print 'Second array:'
print b
print '\n'

print 'First Array + Second Array'
print a + b
```

Demonstração ao vivo

O resultado deste programa seria o seguinte -

First array:

```
[[ 0.  0.  0.]
 [ 10. 10. 10.]
 [ 20. 20. 20.]
 [ 30. 30. 30.]]
```

Second array:

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

First Array + Second Array

```
[[ 1.  2.  3.]
 [ 11. 12. 13.]
 [ 21. 22. 23.]
 [ 31. 32. 33.]]
```



A figura a seguir demonstra como o array **b** é transmitido para se tornar compatível com **a** .

