# Deep Learning

## A course about theory & practice

## Tensor Broadcasting

Marco Piastra

# The General Broadcasting Rules

- TensorFlow adopts the general broadcasting rules of NumPy

  When operating on two arrays, NumPy compares their shapes element-wise

  It starts with the **trailing** dimensions, and works its way forward


- Two dimensions are **compatible** when

  1. they are equal, or
  2. one of them is 1

- The size of the resulting array is the **maximum size** along each dimension of the input arrays


- When a tensor is broadcast, its entries are **conceptually copied**

  Broadcasting is a performance optimization, thus,
  for performance reasons, **no actual copying occurs**

# Applying the General Broadcasting Rule

```
A        (2d array):   5 x 4
B        (1d array):       1
Result (2d array):    5 x 4


A        (3d array):  15 x 3 x 1
B        (2d array):       3 x 5
Result (3d array):  15 x 3 x 5


A        (4d array):   8 x 1 x 6 x 5
B        (3d array):       7 x 1 x 5
Result (4d array):   8 x 7 x 6 x 5
```

# Broadcasting: another example

- Each channel of an RGB image can be scaled by multiplying the image by a 1-D array (vector) with 3 values.

```
Image  (3d array): 4 x 4 x 3
Scale  (1d array):         3
Result (3d array): 4 x 4 x 3
```
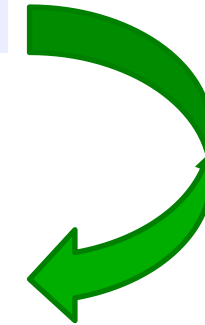
| 0.5 | 0.3 | 0.2 |

| 0.5 | 0.5 | 0.5 | 0.5 |
|-----|-----|-----|-----|
| 0.5 | 0.5 | 0.5 | 0.5 |
| 0.5 | 0.5 | 0.5 | 0.5 |
| 0.5 | 0.5 | 0.5 | 0.5 |

| 0.3 | 0.3 | 0.3 | 0.3 |
|-----|-----|-----|-----|
| 0.3 | 0.3 | 0.3 | 0.3 |
| 0.3 | 0.3 | 0.3 | 0.3 |
| 0.3 | 0.3 | 0.3 | 0.3 |

| 0.2 | 0.2 | 0.2 | 0.2 |
|-----|-----|-----|-----|
| 0.2 | 0.2 | 0.2 | 0.2 |
| 0.2 | 0.2 | 0.2 | 0.2 |
| 0.2 | 0.2 | 0.2 | 0.2 |

Broadcasting