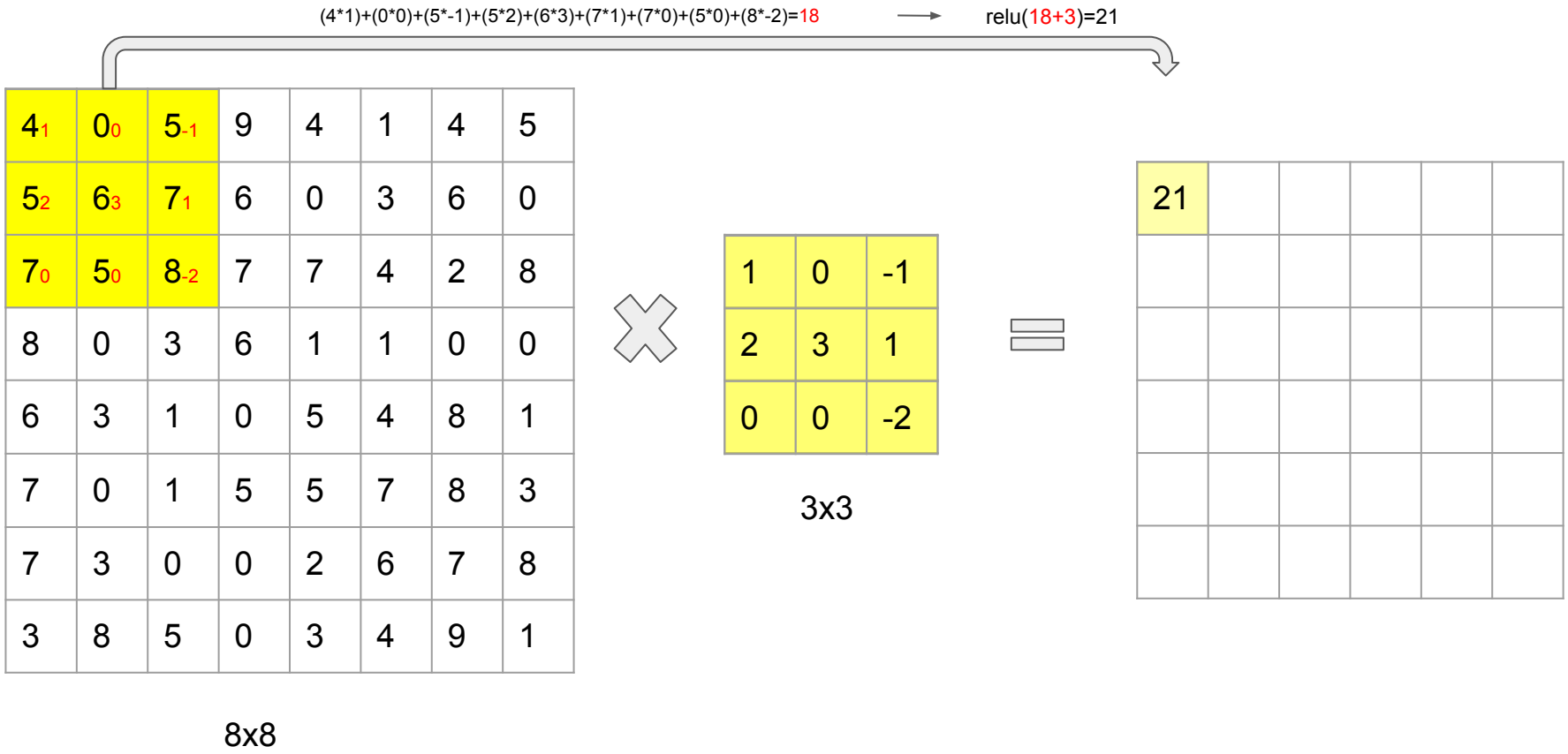
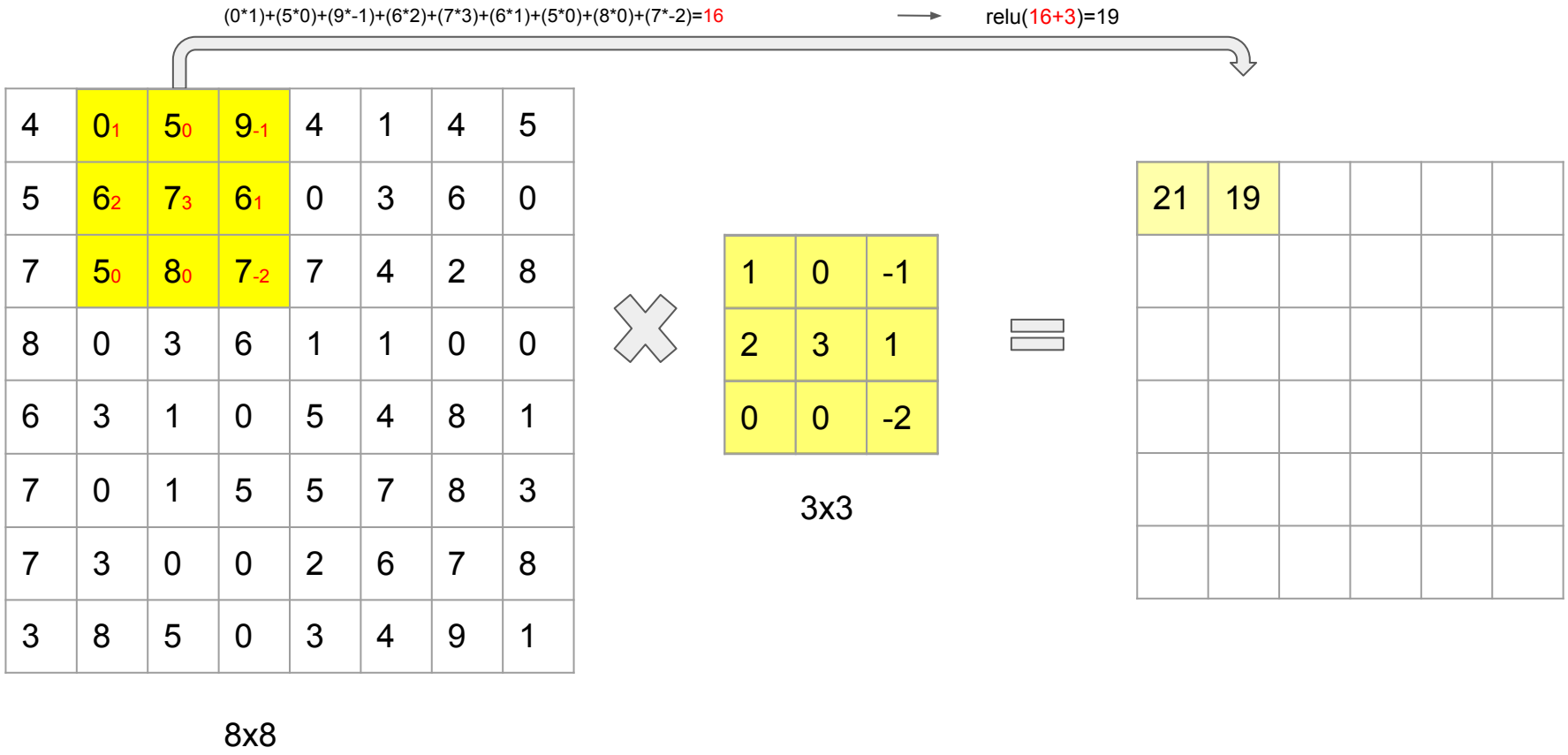
# *Convolutional neural network with application in R*
# *-- image classification --*

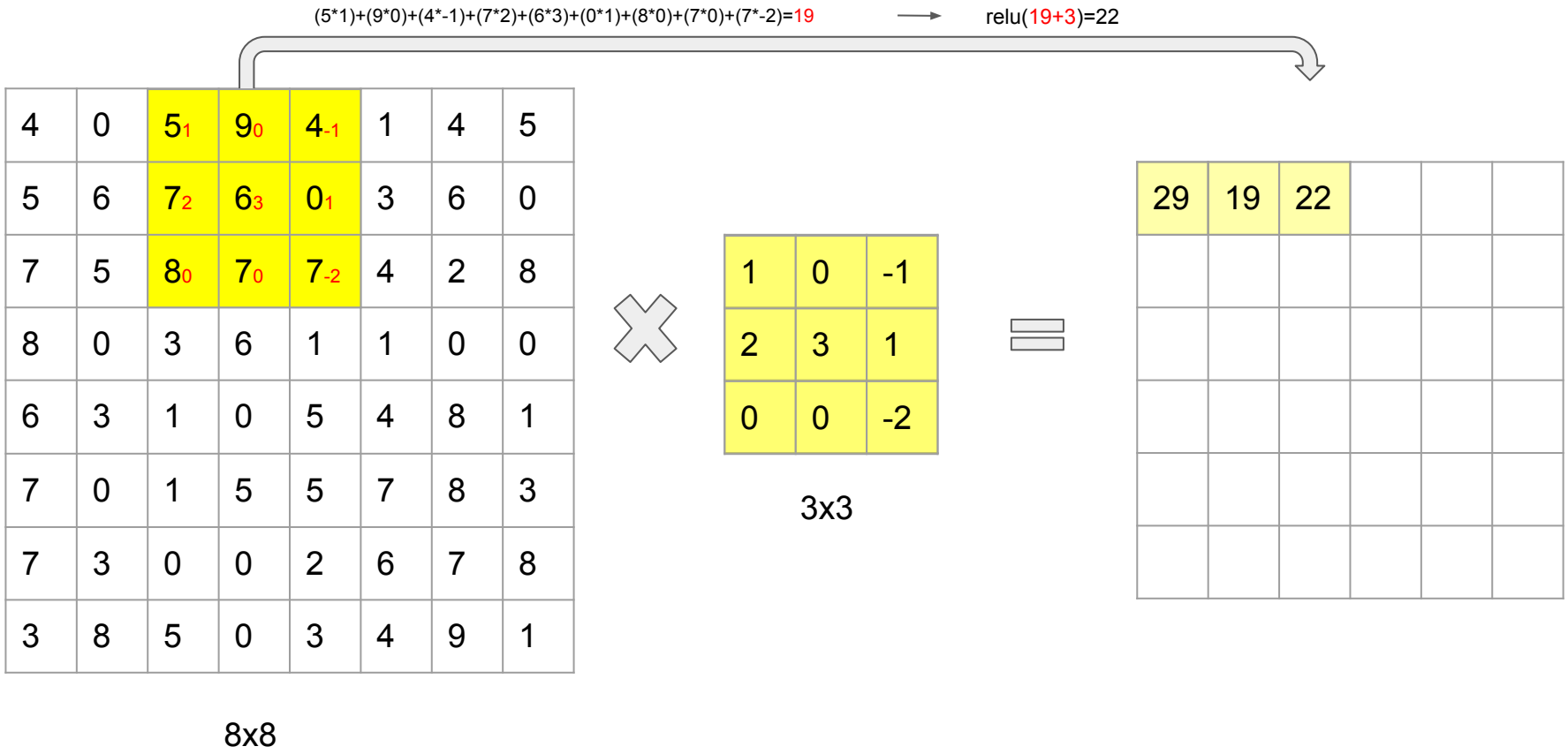
# 1- Convolutions on grey scale image



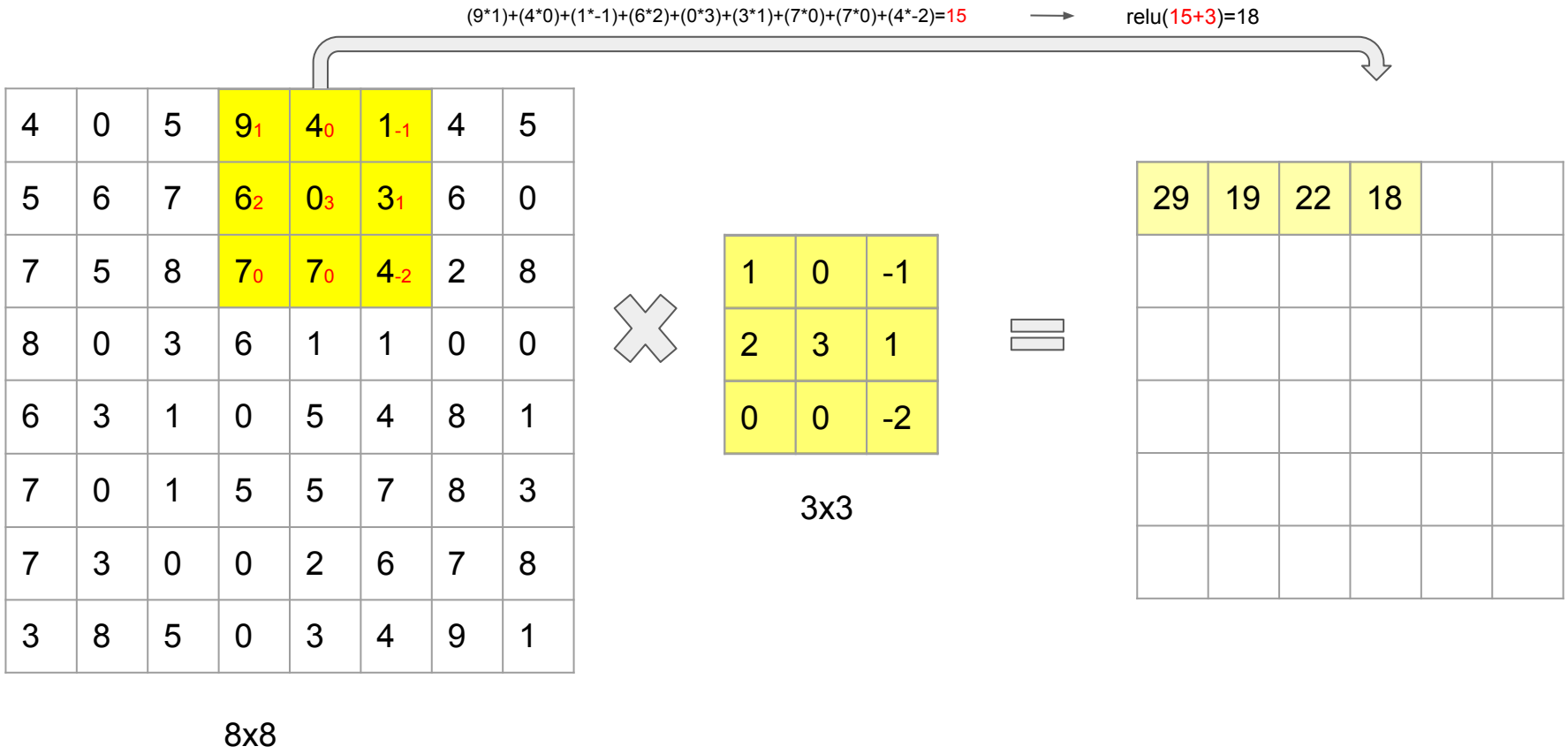
# convolution on grey scale image



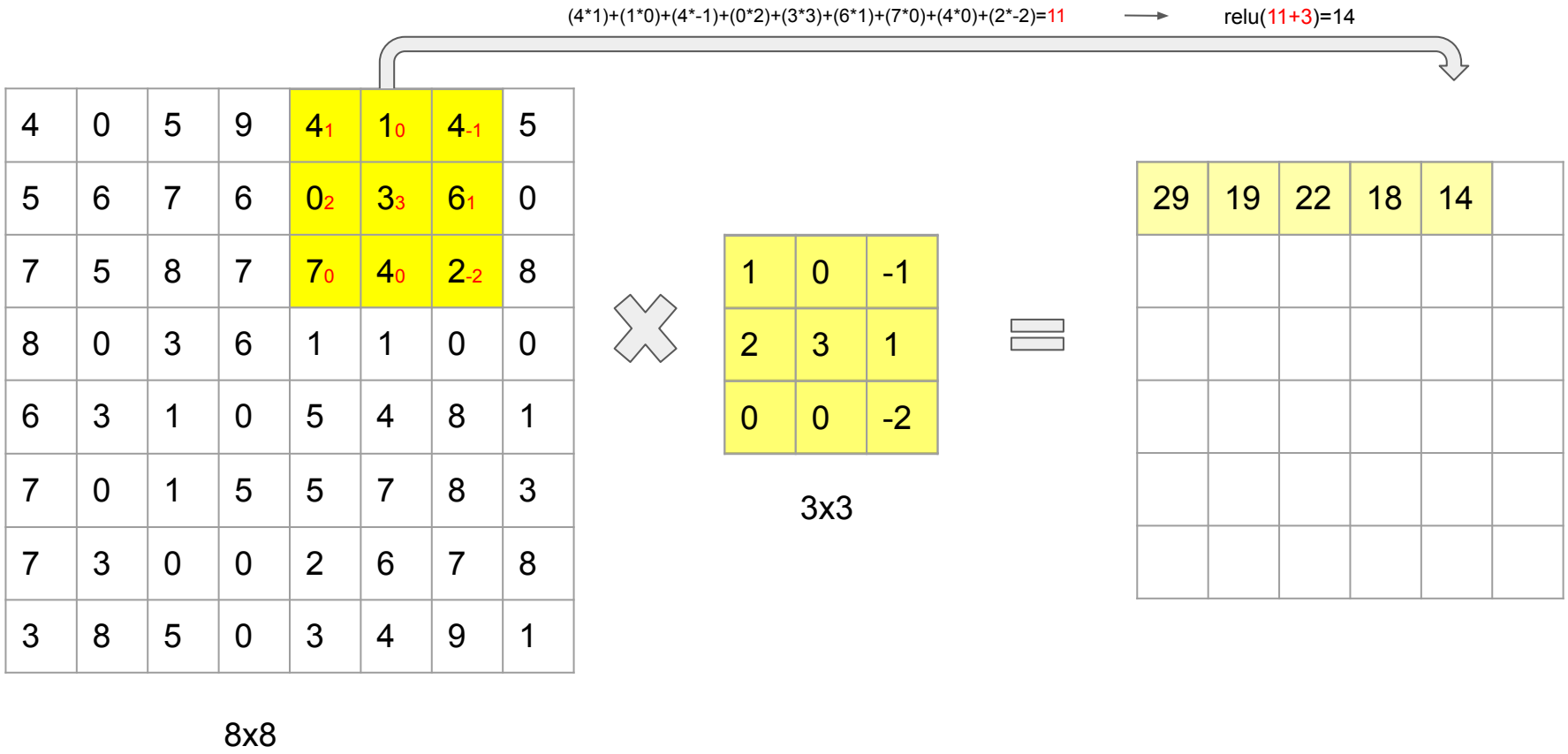
# convolution on grey scale image



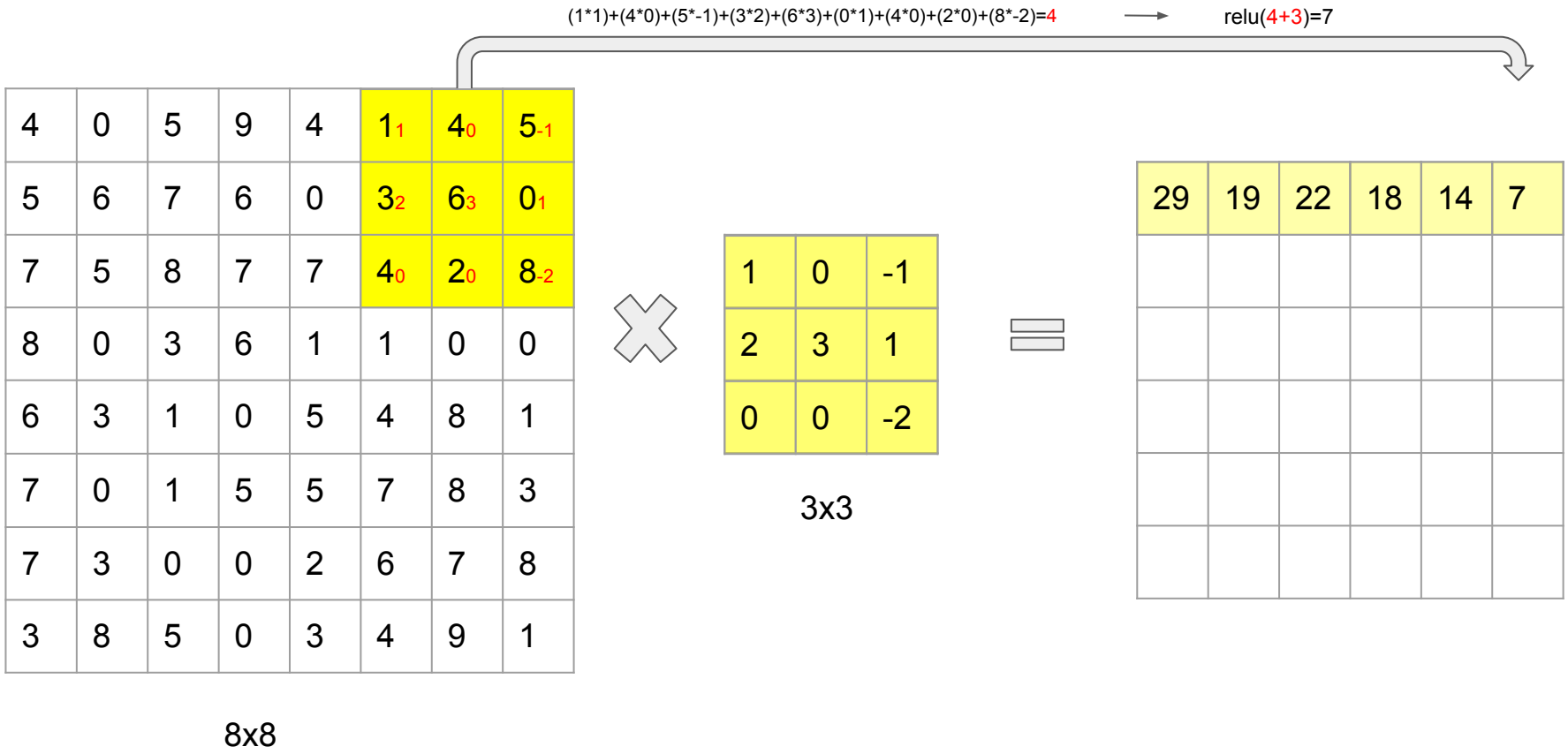
# convolution on grey scale image



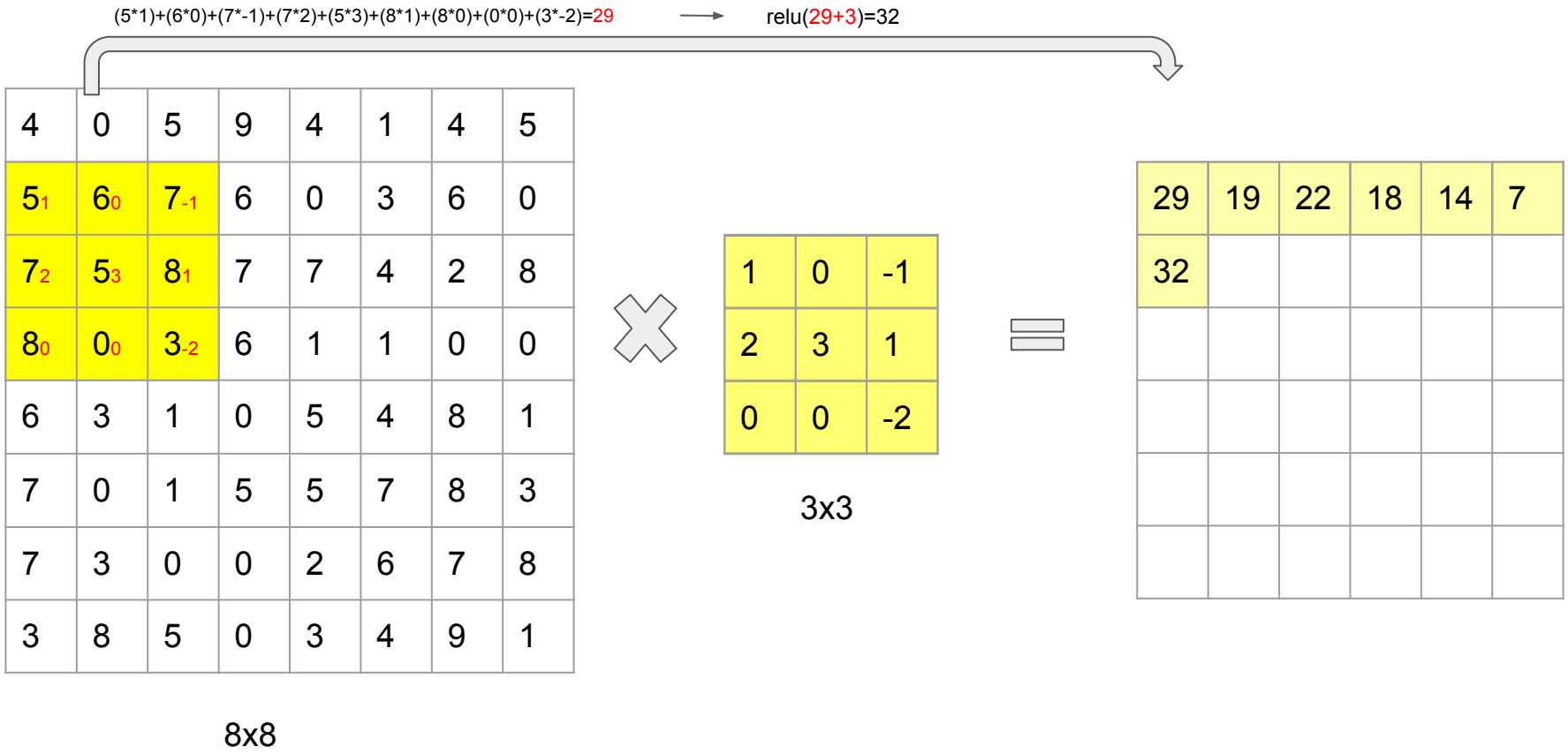
# convolution on grey scale image



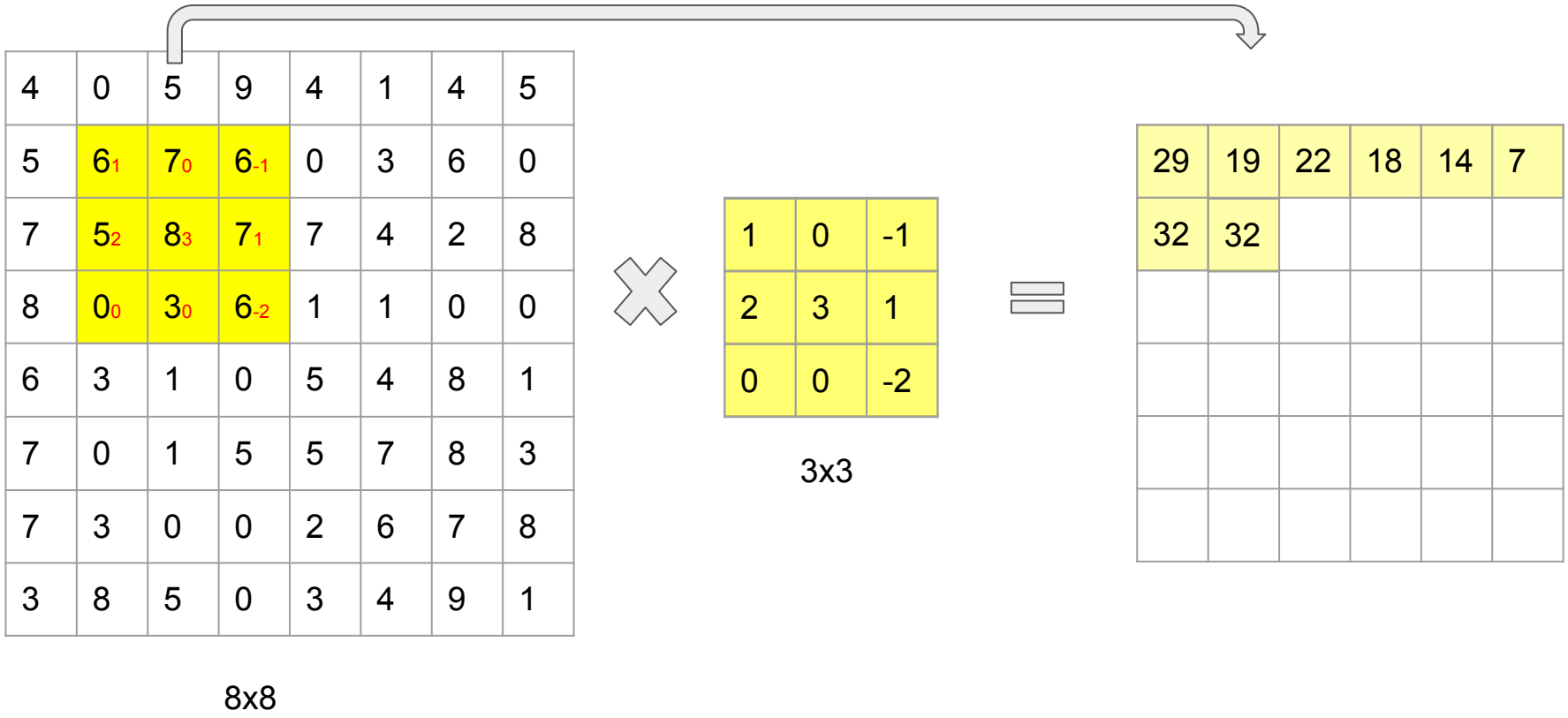
# convolution on grey scale image



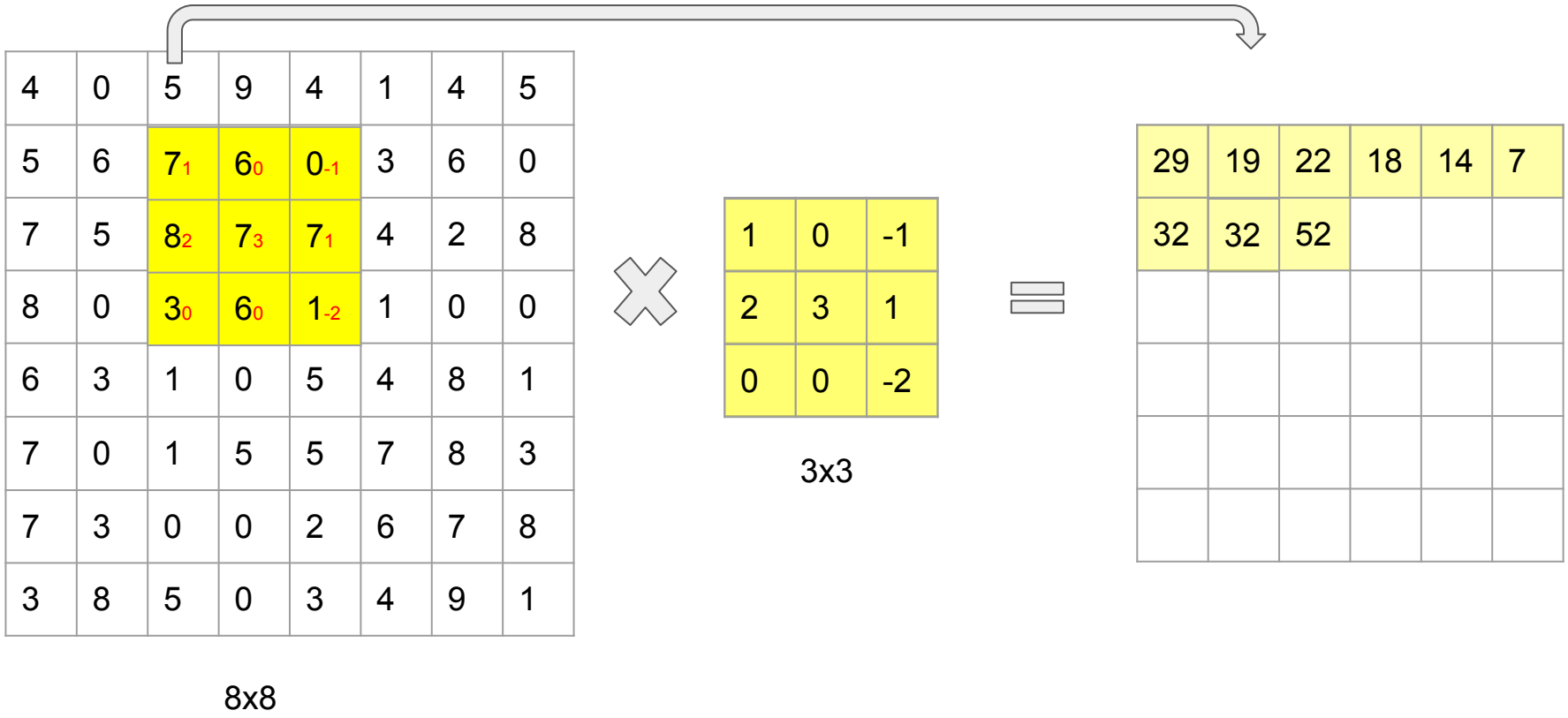
# convolution on grey scale image



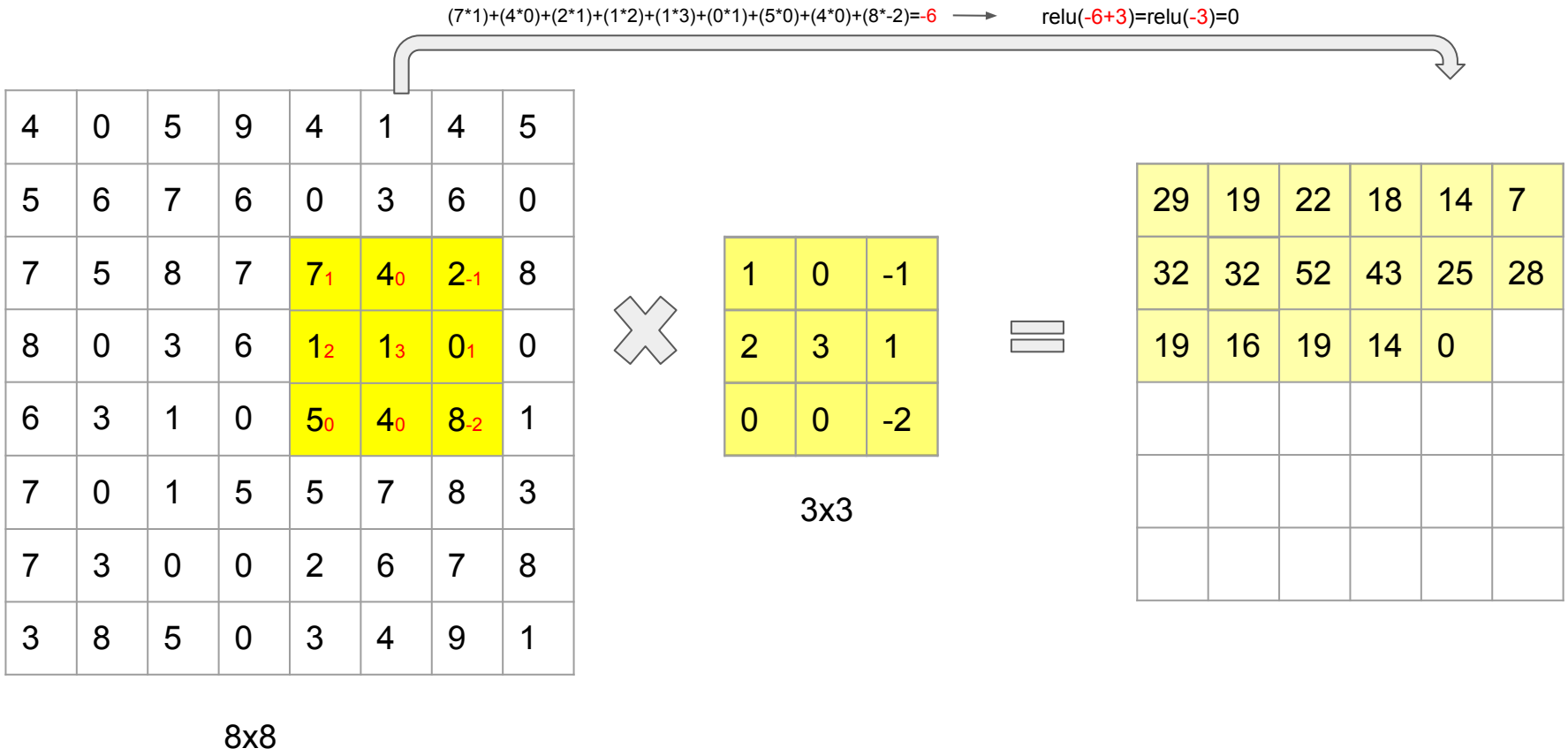
# convolution on grey scale image



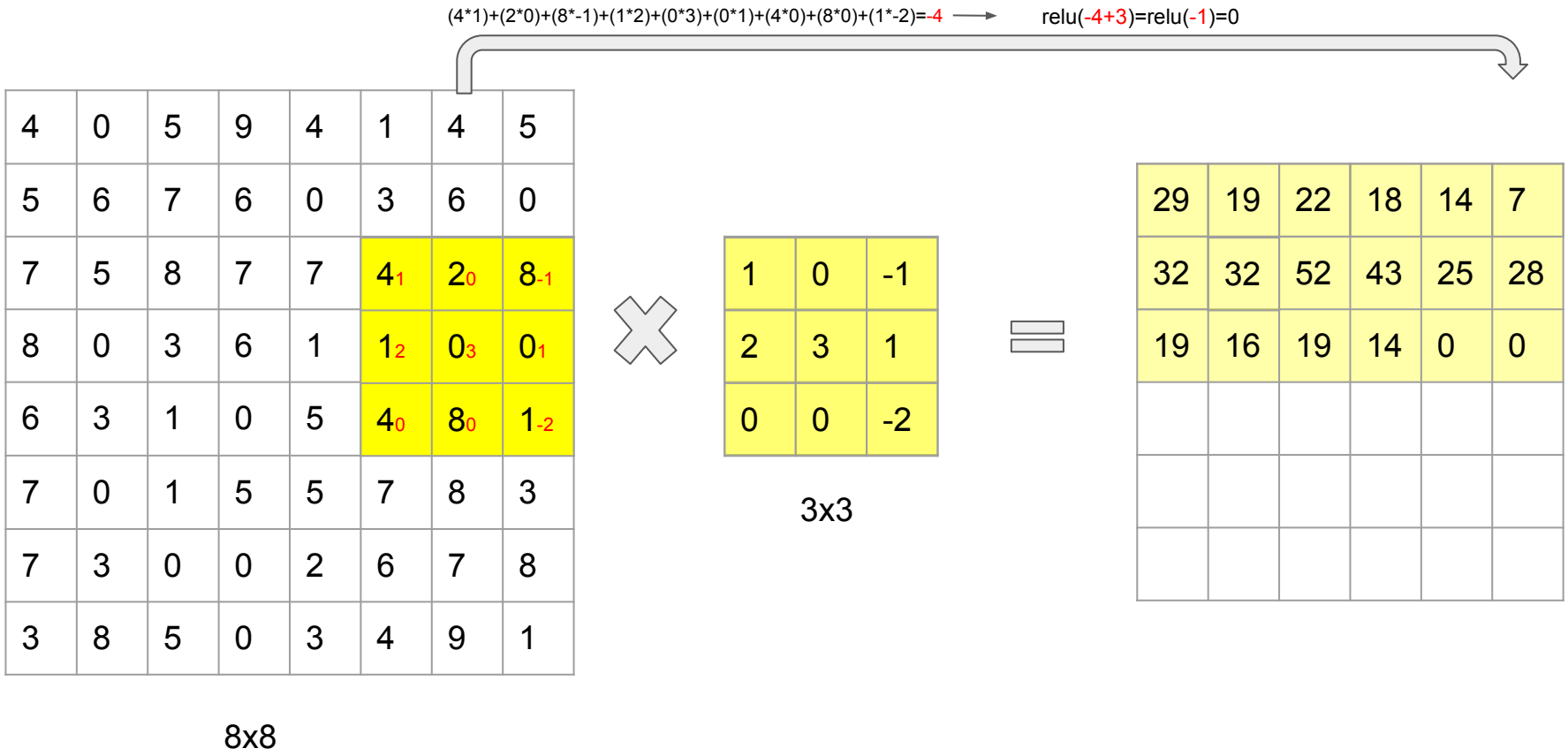
# convolution on grey scale image



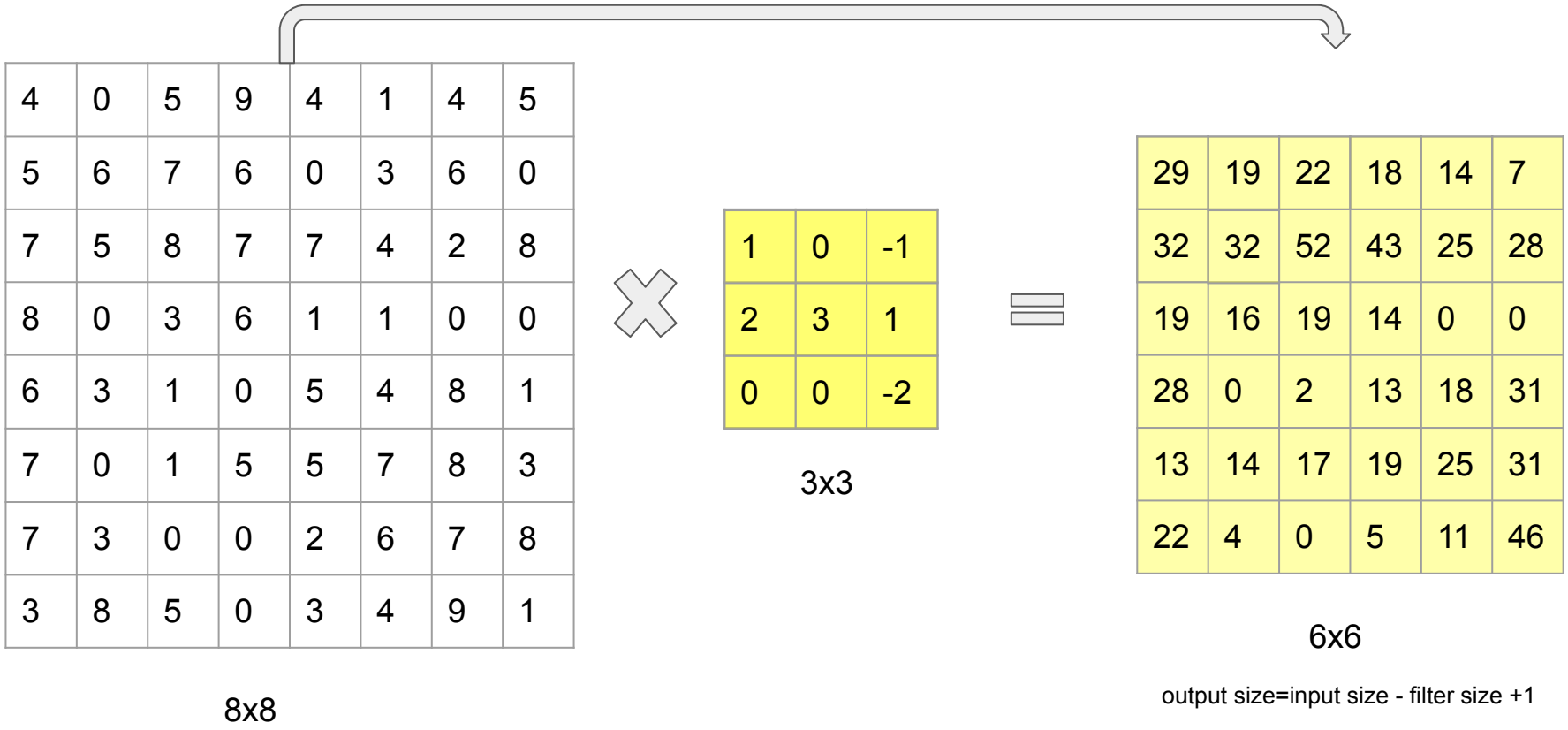
# convolution on grey scale image



# convolution on grey scale image



# convolution on grey scale image



# Using more filters

| | | | | | |
|---|---|---|---|---|---|
| 4 | 0 | 5 | 9 | 4 | 1 |
| 5 | 6 | 7 | 6 | 0 | 3 |
| 7 | 5 | 8 | 7 | 7 | 4 |
| 8 | 0 | 3 | 6 | 1 | 1 |
| 6 | 3 | 1 | 0 | 5 | 4 |
| 7 | 0 | 1 | 5 | 5 | 7 |

6x6

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 2 | 3 | 1 |
| 0 | 0 | -2 |

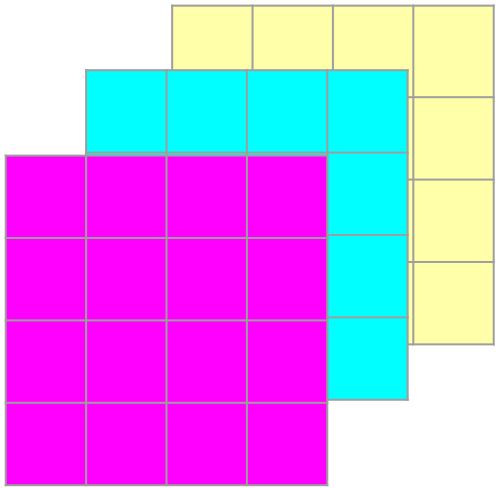
| | | |
|---|---|---|
| 1 | 0 | 1 |
| 1 | -1 | 1 |
| 1 | 0 | 2 |

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

---

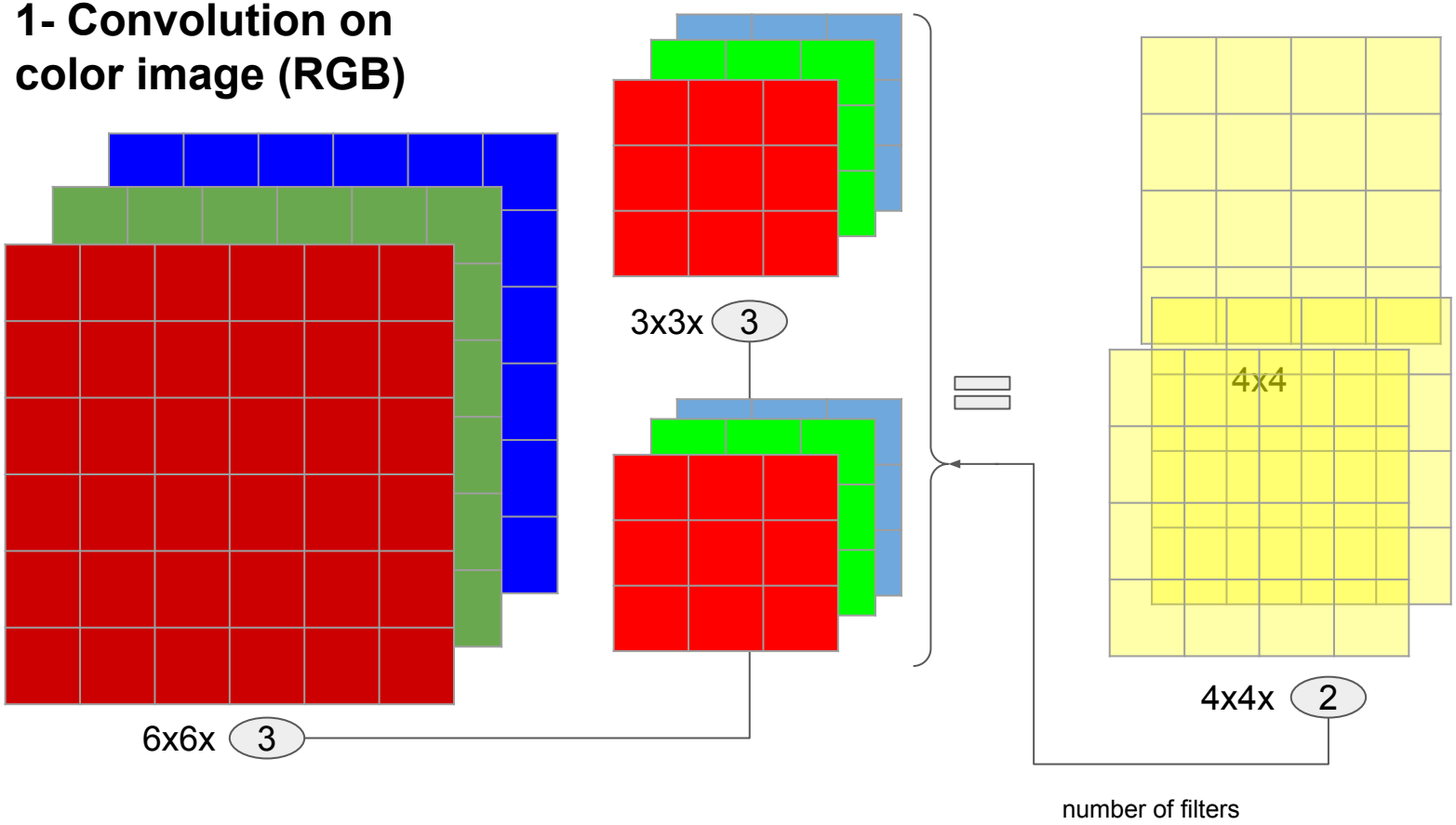
---

number of filters

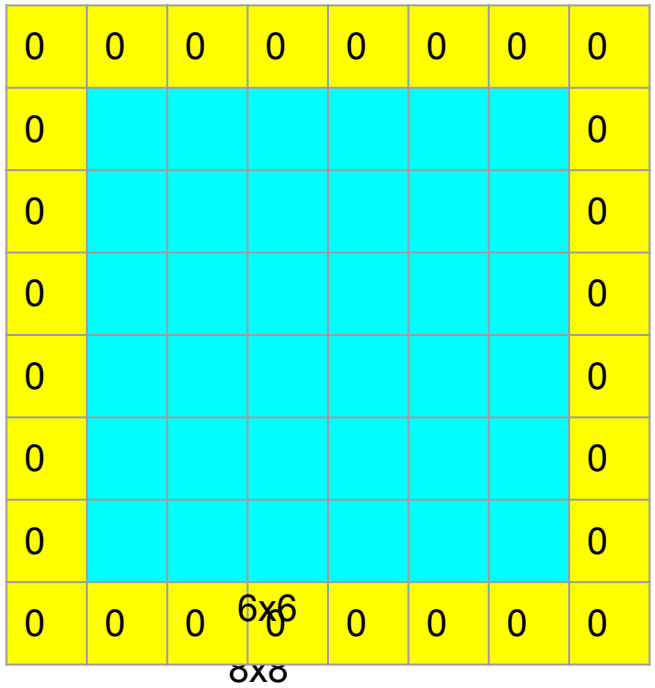


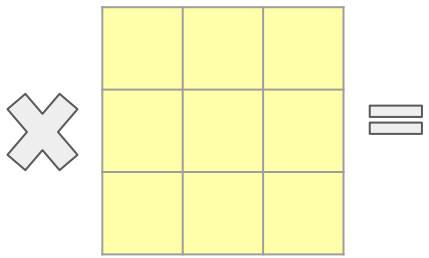
4x4x3

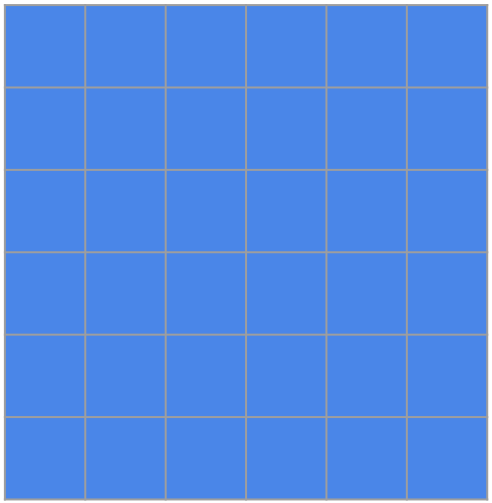
# 1- Convolution on color image (RGB)



# 3 - Padding



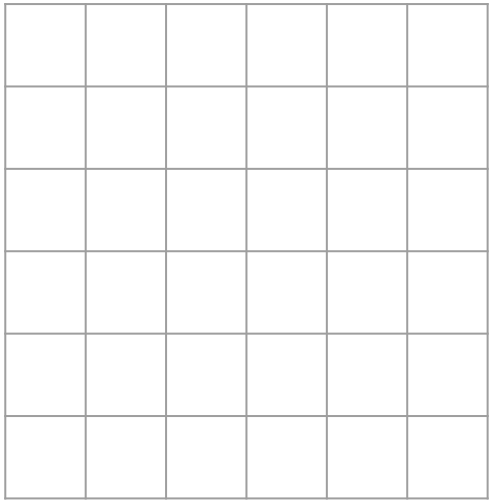



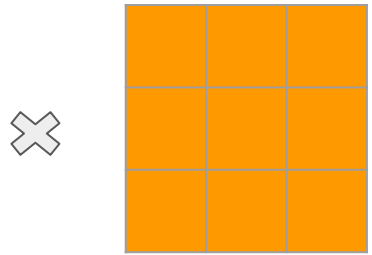

p=1

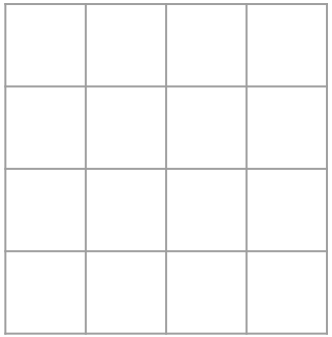
# 4 - stride

stride = 1



6x6

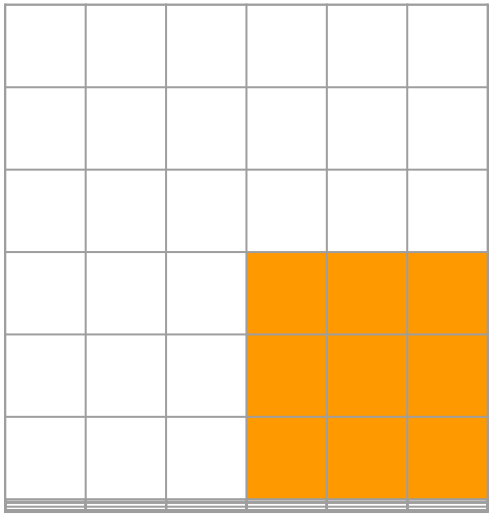




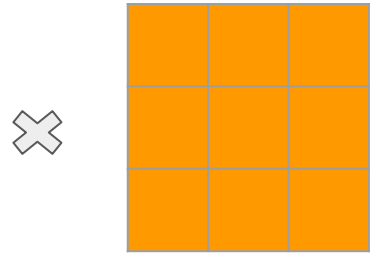
4x4

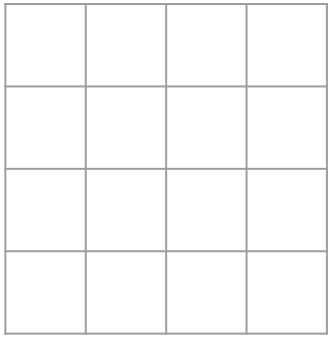
# 4 - stride

stride = 1



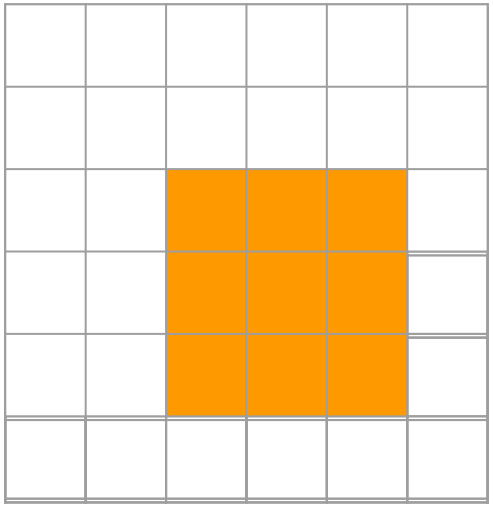
6x6





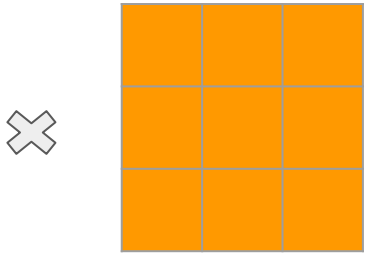
4x4

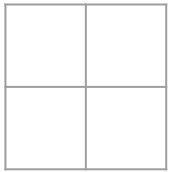
# 4 - stride



6x6

stride = 2





2x2

$$\text{output size} = \frac{\text{input size} + 2p - f}{s} + 1$$

f= filter size
p= padding size
s= strides

# 5 - pooling (empty filter)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 0 | 5 | 9 | 4 | 1 | 4 | 5 |
| 5 | 6 | 7 | 6 | 0 | 3 | 6 | 0 |
| 7 | 5 | 8 | 7 | 7 | 4 | 2 | 8 |
| 8 | 0 | 3 | 6 | 1 | 1 | 0 | 0 |
| 6 | 3 | 1 | 0 | 5 | 4 | 8 | 1 |
| 7 | 0 | 1 | 5 | 5 | 7 | 8 | 3 |
| 7 | 3 | 0 | 0 | 2 | 6 | 7 | 8 |
| 3 | 8 | 5 | 0 | 3 | 4 | 9 | 1 |

8x8

max-pooling 2x2
with stride s=2

| | | | |
|---|---|---|---|
| 6 | 9 | 4 | 6 |
| 8 | 8 | 7 | 8 |
| 7 | 5 | 7 | 8 |
| 8 | 5 | 6 | 9 |

4x4

# 6 - flattening

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 0 | 5 | 9 | 4 | 1 | 4 | 5 |
| 5 | 6 | 7 | 6 | 0 | 3 | 6 | 0 |
| 7 | 5 | 8 | 7 | 7 | 4 | 2 | 8 |
| 8 | 0 | 3 | 6 | 1 | 1 | 0 | 0 |
| 6 | 3 | 1 | 0 | 5 | 4 | 8 | 1 |
| 7 | 0 | 1 | 5 | 5 | 7 | 8 | 3 |
| 7 | 3 | 0 | 0 | 2 | 6 | 7 | 8 |
| 3 | 8 | 5 | 0 | 3 | 4 | 9 | 1 |

8x8

flattening

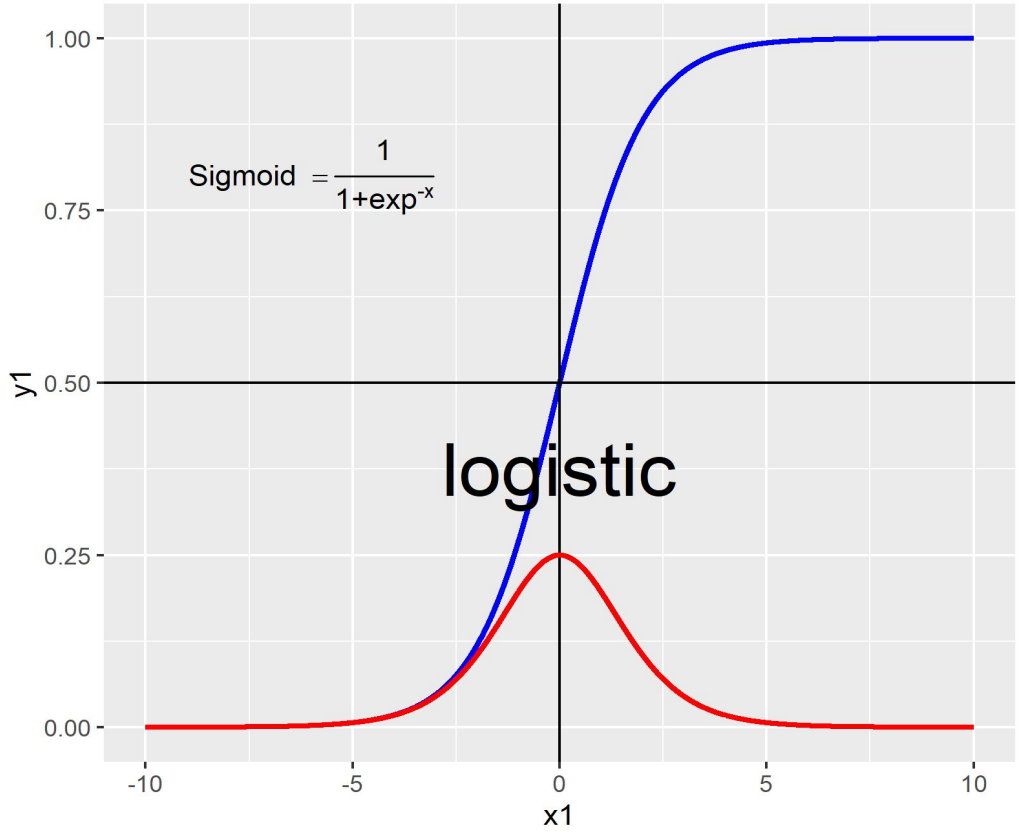
| |
|---|
| 4 |
| 0 |
| 5 |
| 9 |
| --- |
| --- |
| --- |
| 3 |
| 4 |
| 9 |
| 1 |

64x1

ANN

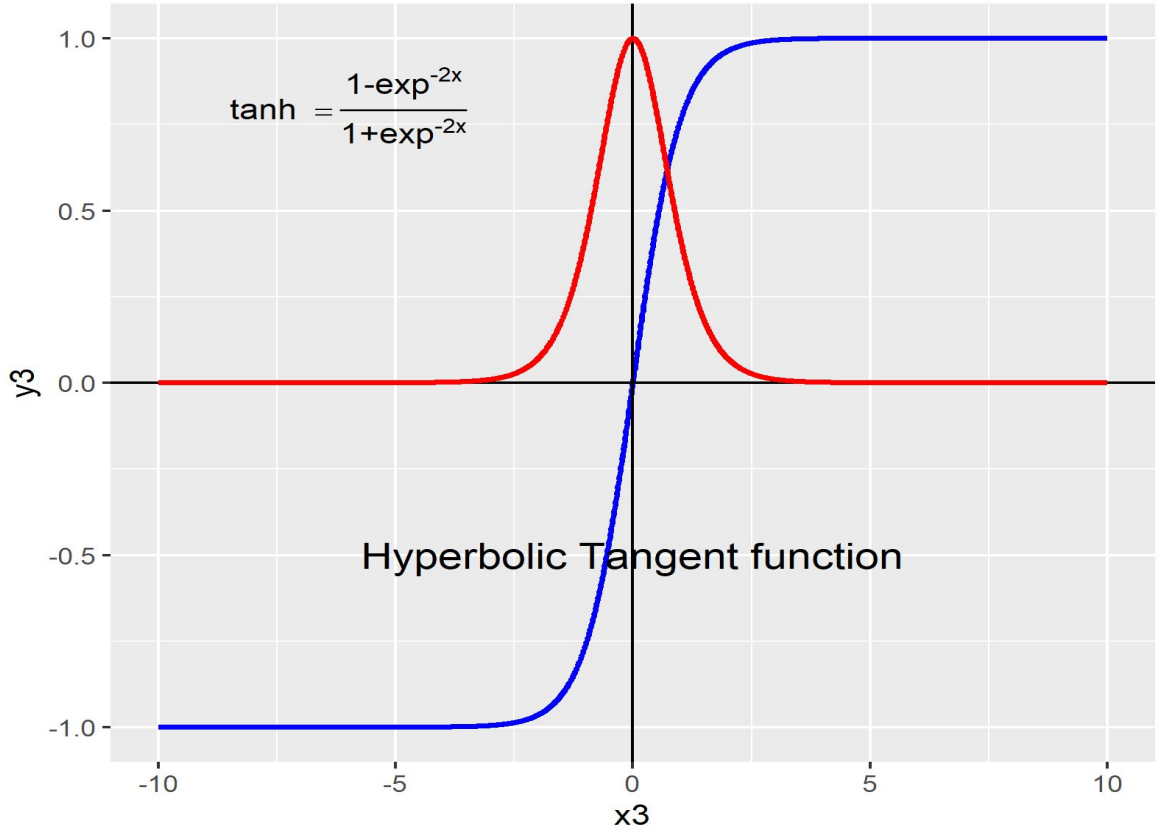
# 7 - **Activation functions**

- **Sigmoid function :**



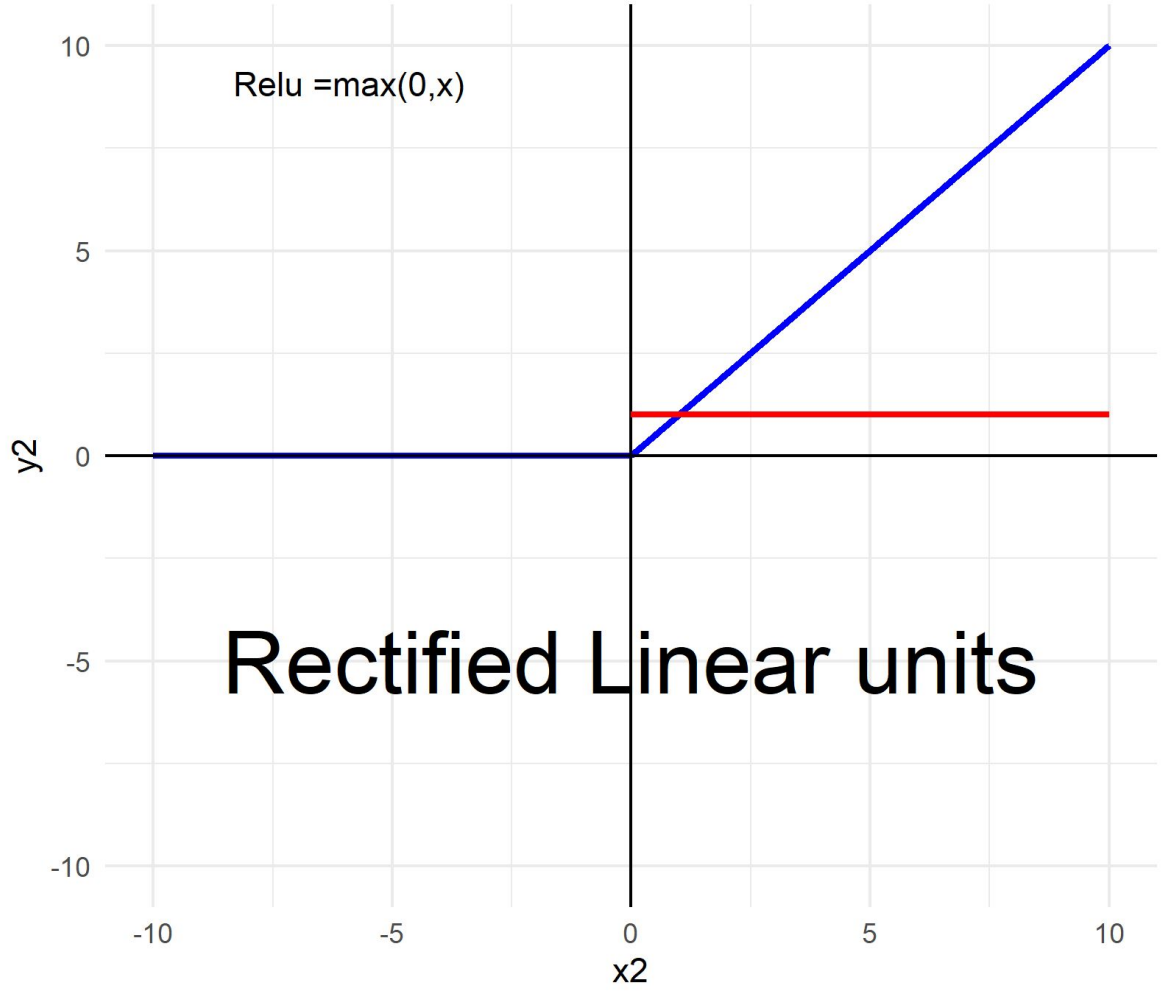
# 7 - **Activation functions**

- **Tanh function :**



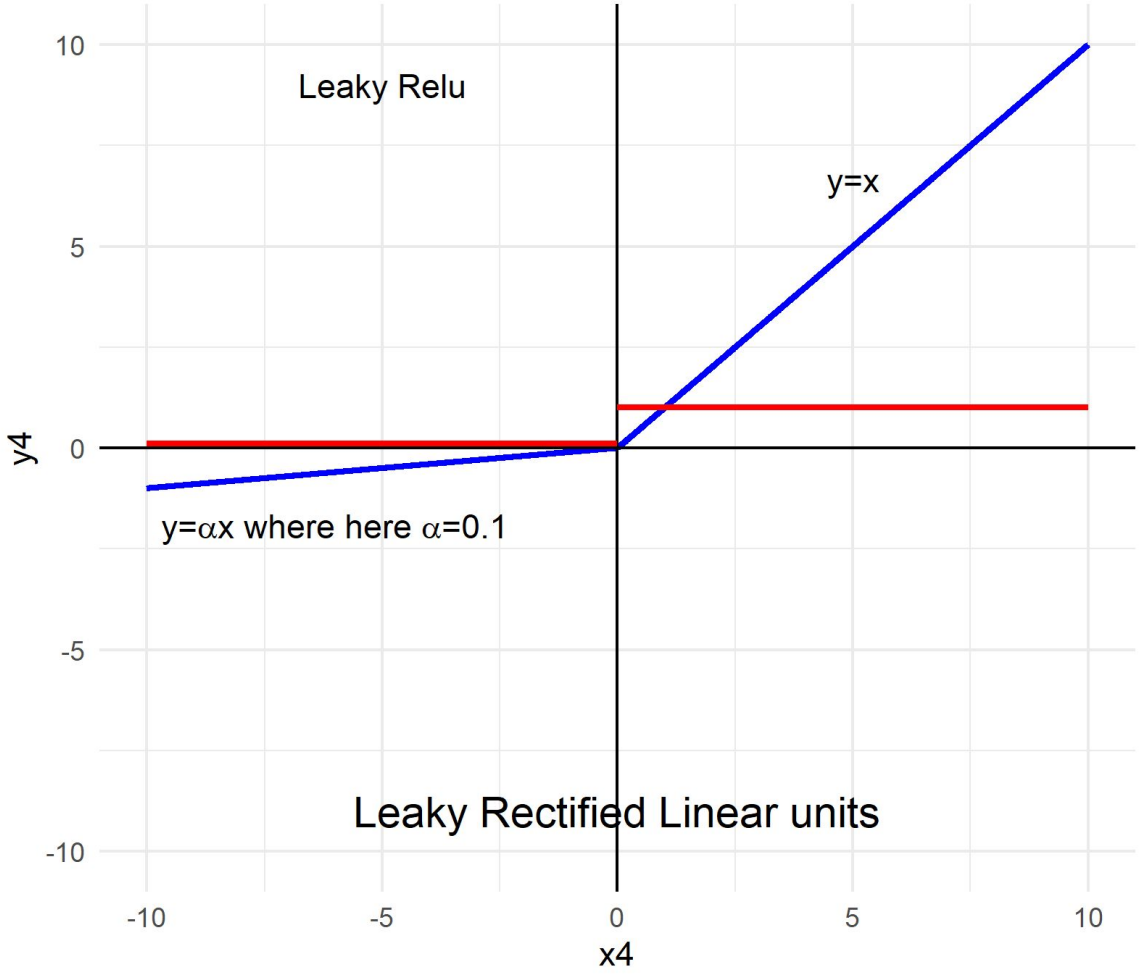
## 7 - **Activation functions**

- **Relu function:**

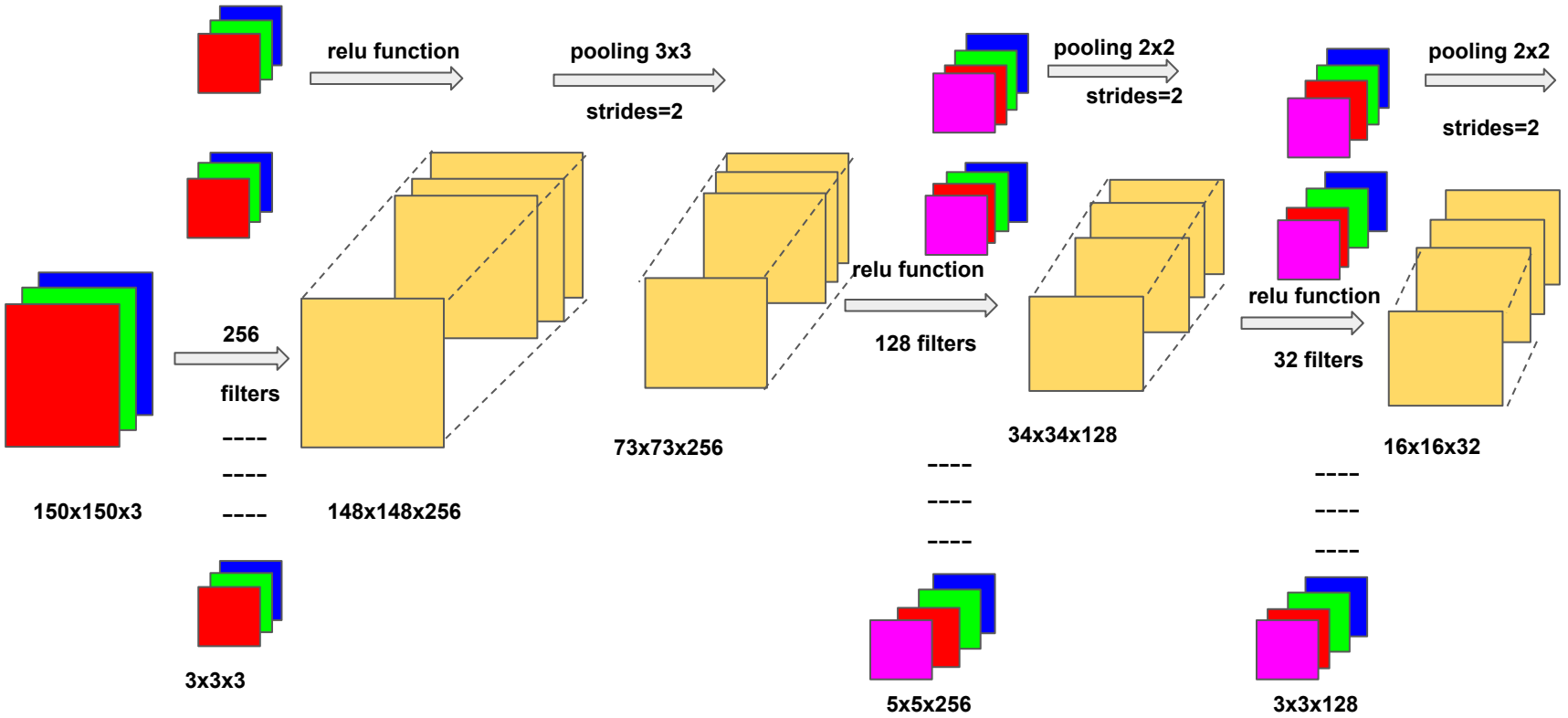


## 7 - **Activation functions**

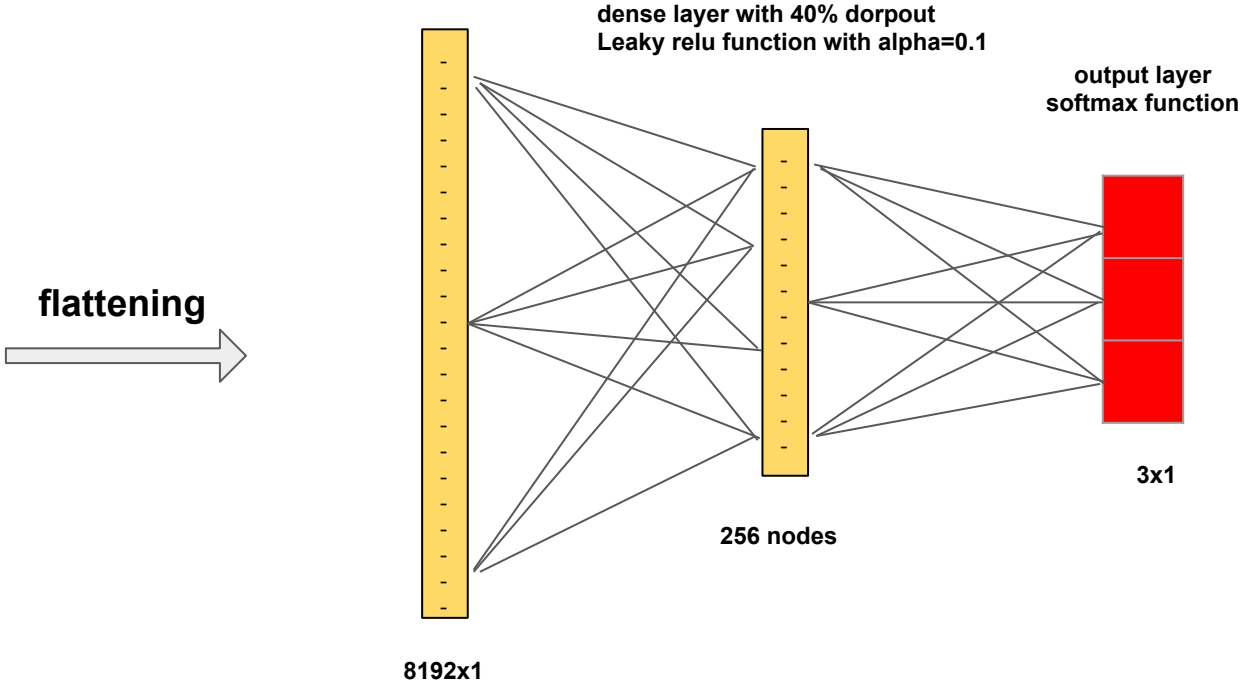
- **Leaky relu function:**



# 8- **Architecture of the CNN model**



# 8- **Architecture of the CNN model**




**Fully connected layers**