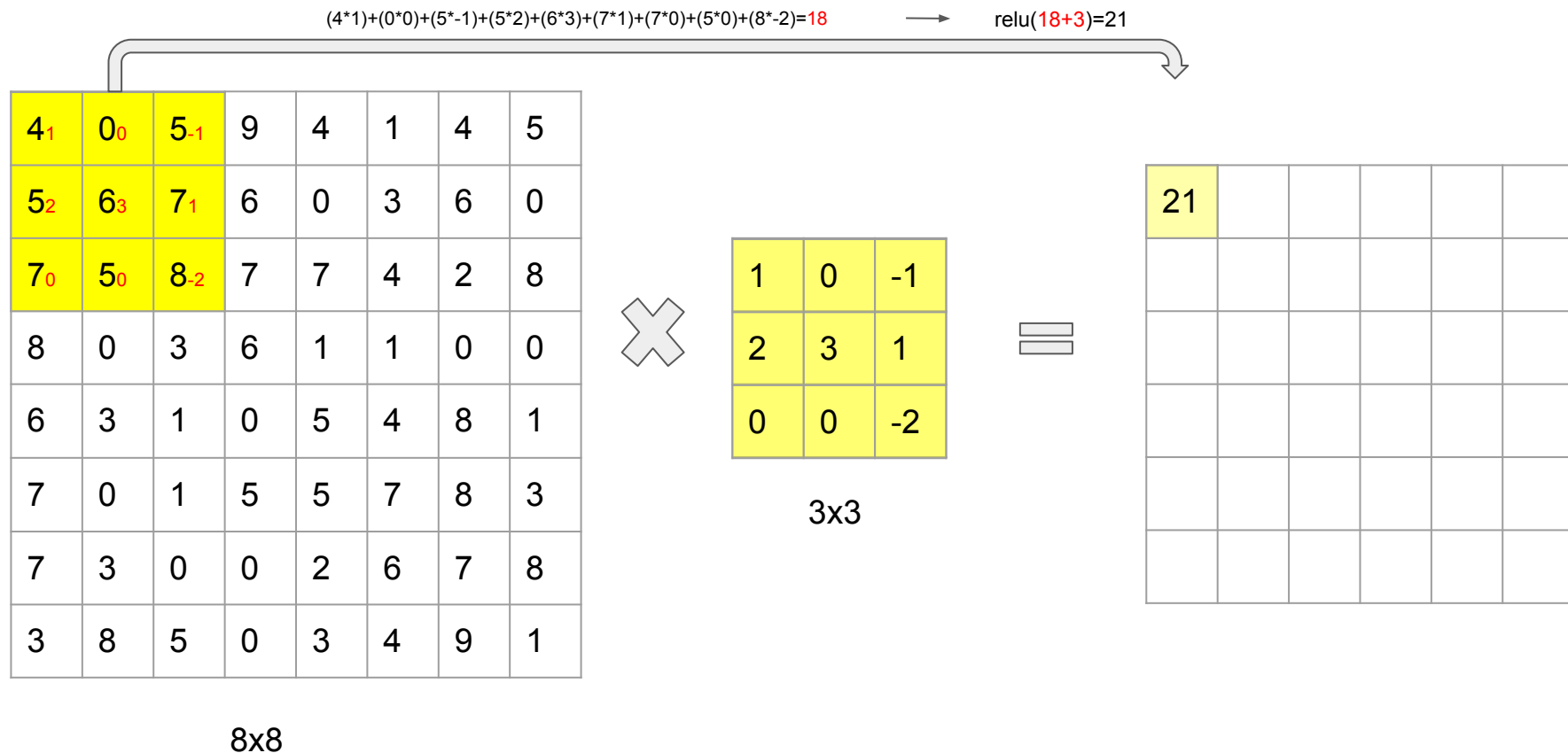
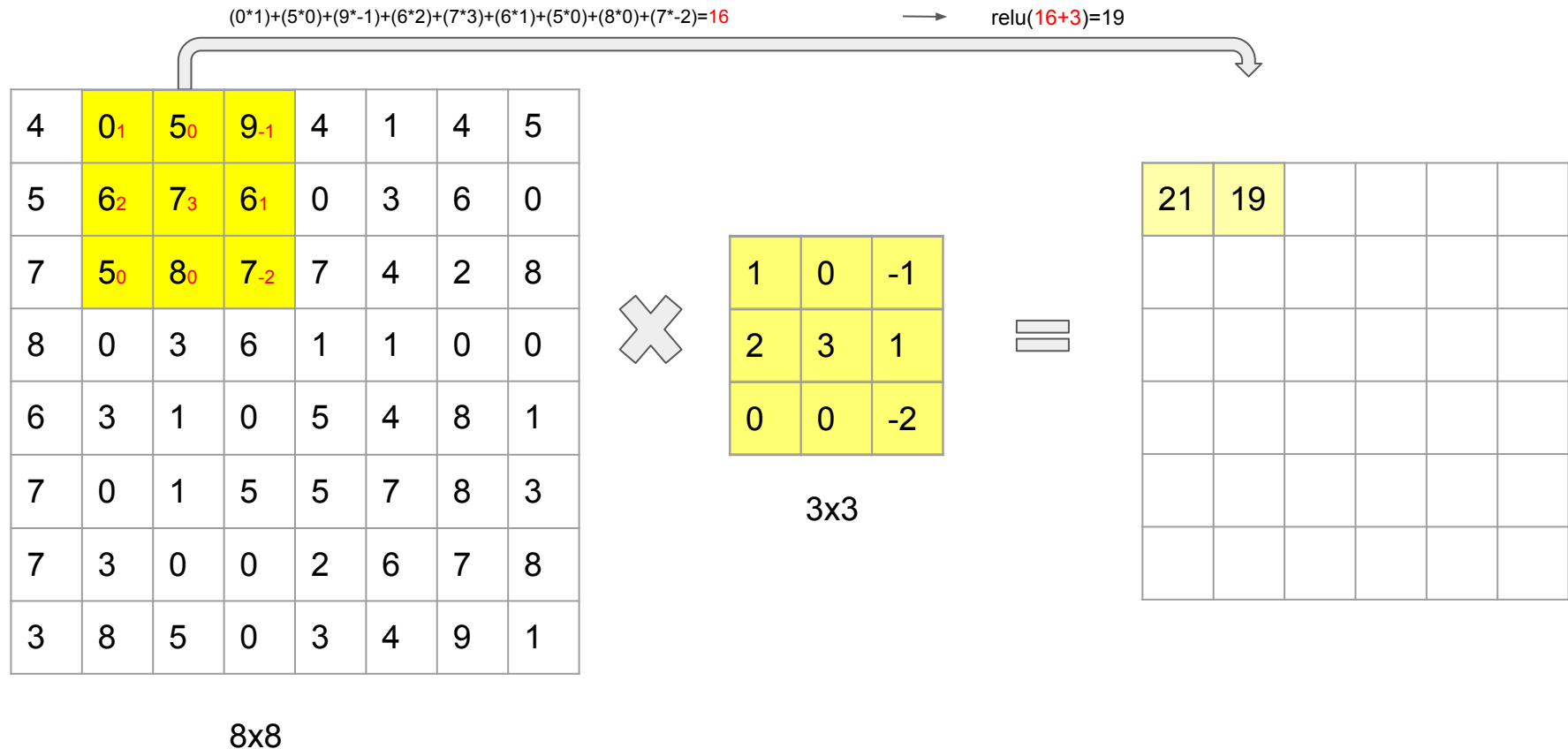


Convolutional neural network with application in R
-- image classification --

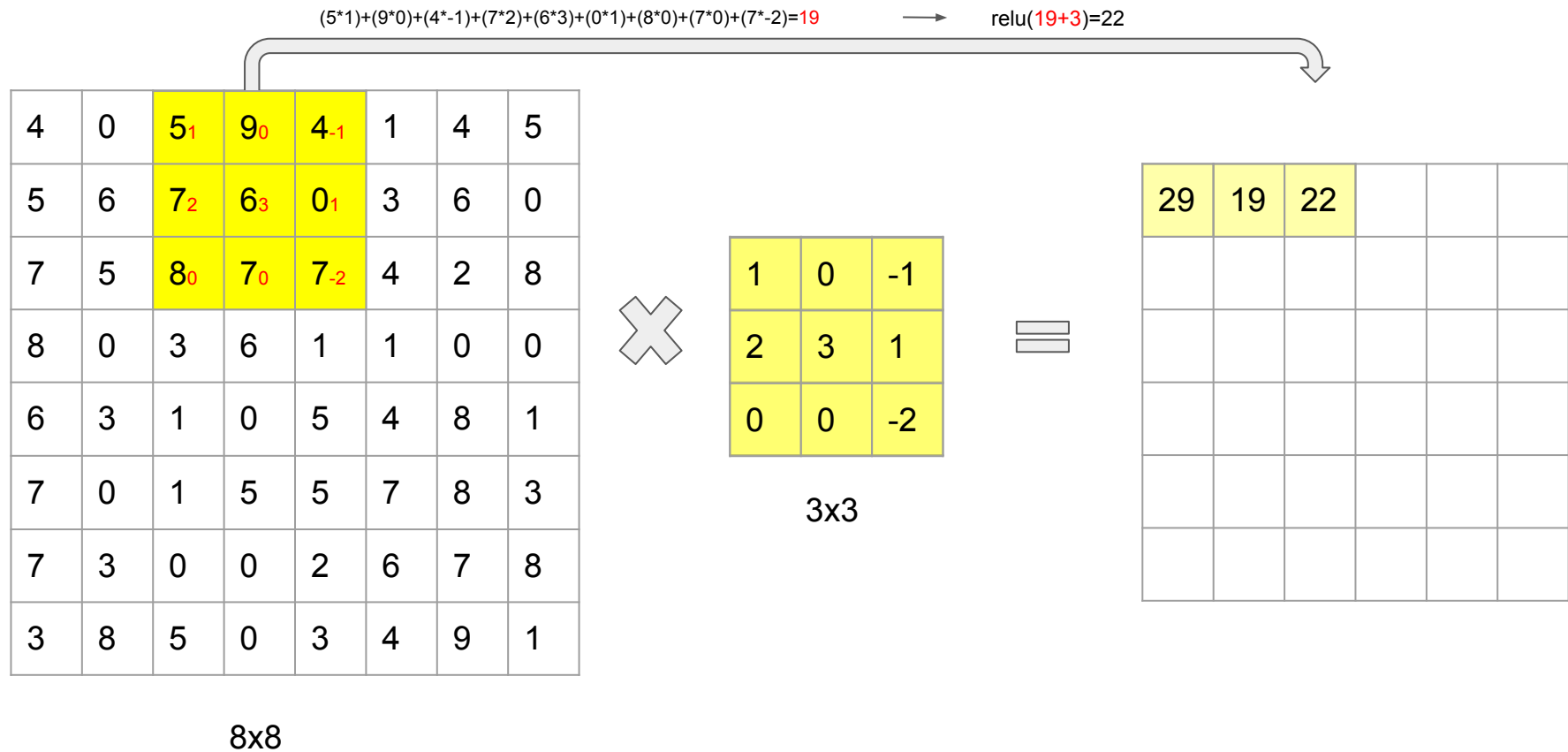
1- Convolutions on grey scale image



convolution on grey scale image



convolution on grey scale image



convolution on grey scale image

$(9*1)+(4*0)+(1*-1)+(6*2)+(0*3)+(3*1)+(7*0)+(7*0)+(4*-2)=15 \longrightarrow \text{relu}(15+3)=18$

4	0	5	9 ₁	4 ₀	1 ₋₁	4	5
5	6	7	6 ₂	0 ₃	3 ₁	6	0
7	5	8	7 ₀	7 ₀	4 ₋₂	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



1	0	-1
2	3	1
0	0	-2

3x3



29	19	22	18		

8x8

convolution on grey scale image

$(4 \times 1) + (1 \times 0) + (4 \times -1) + (0 \times 2) + (3 \times 3) + (6 \times 1) + (7 \times 0) + (4 \times 0) + (2 \times -2) = 11$ \rightarrow $\text{relu}(11 + 3) = 14$

4	0	5	9	4 ₁	1 ₀	4 ₋₁	5
5	6	7	6	0 ₂	3 ₃	6 ₁	0
7	5	8	7	7 ₀	4 ₀	2 ₋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



1	0	-1
2	3	1
0	0	-2

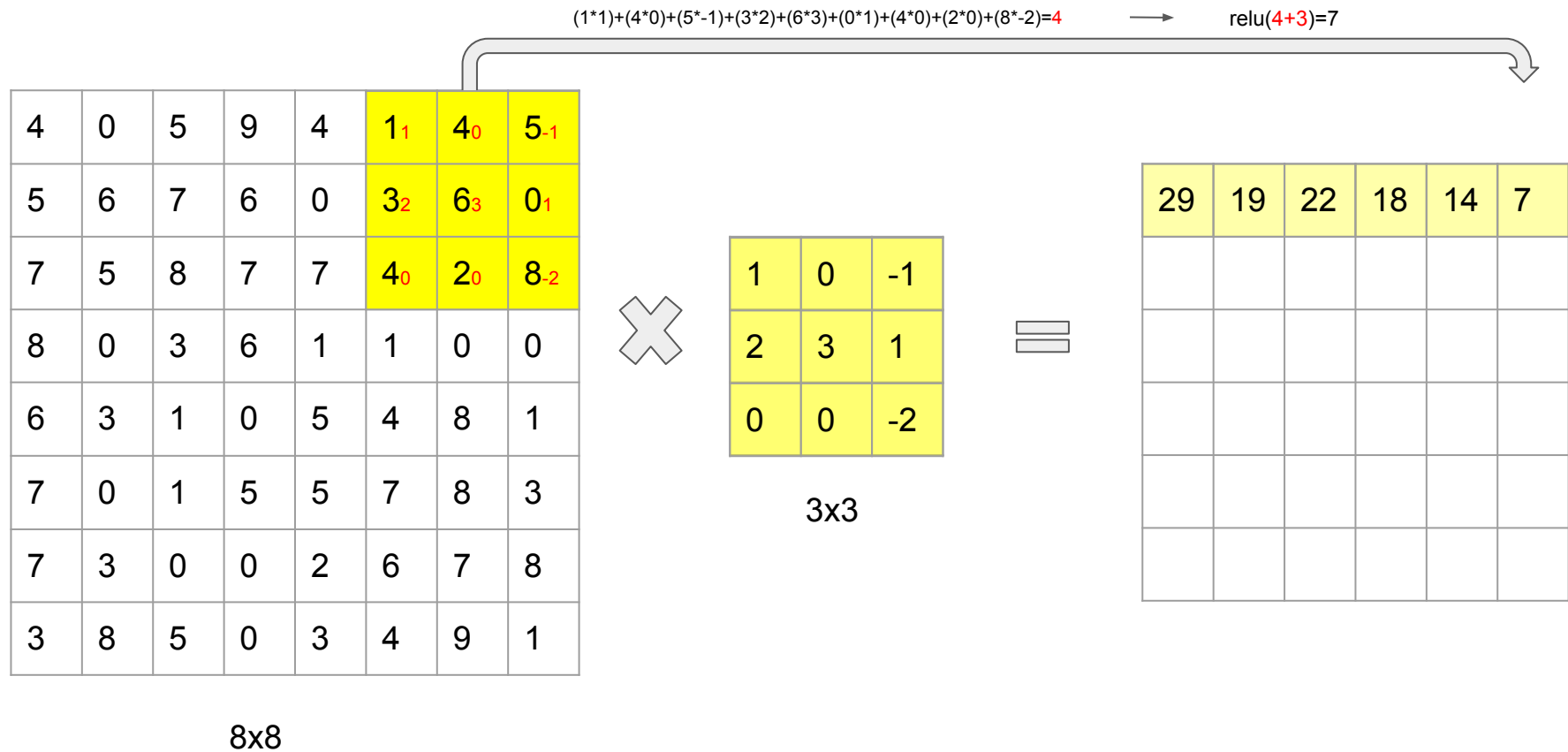
3x3



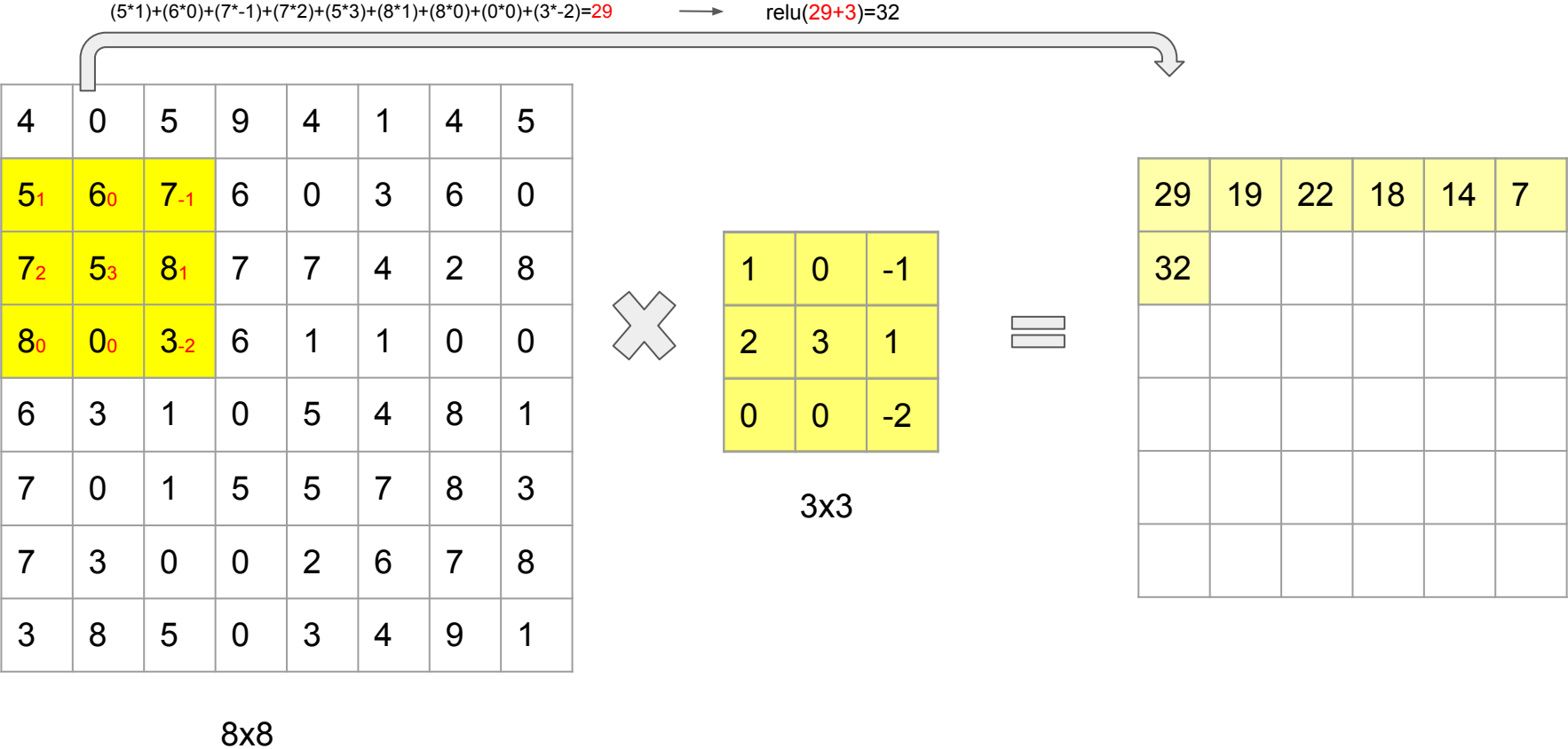
29	19	22	18	14	

8x8

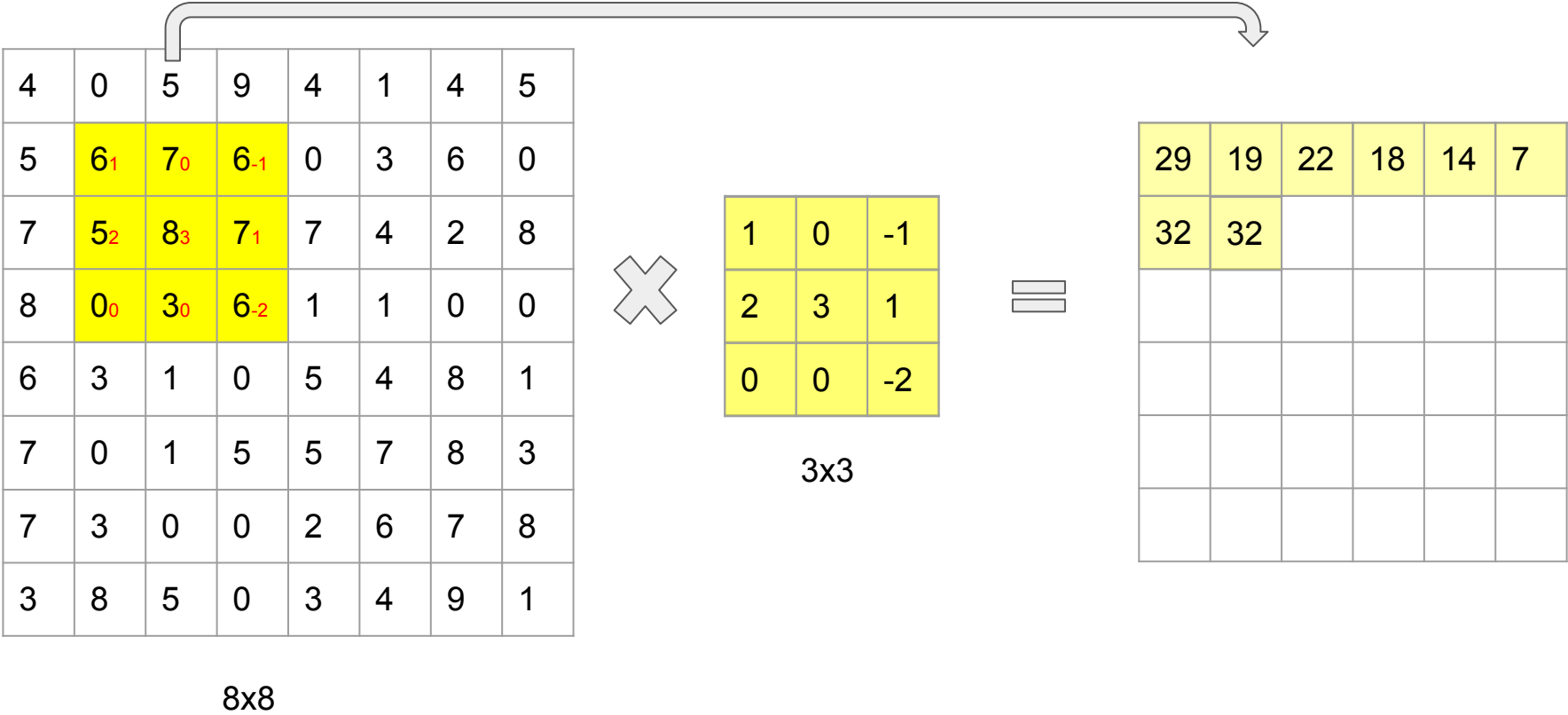
convolution on grey scale image



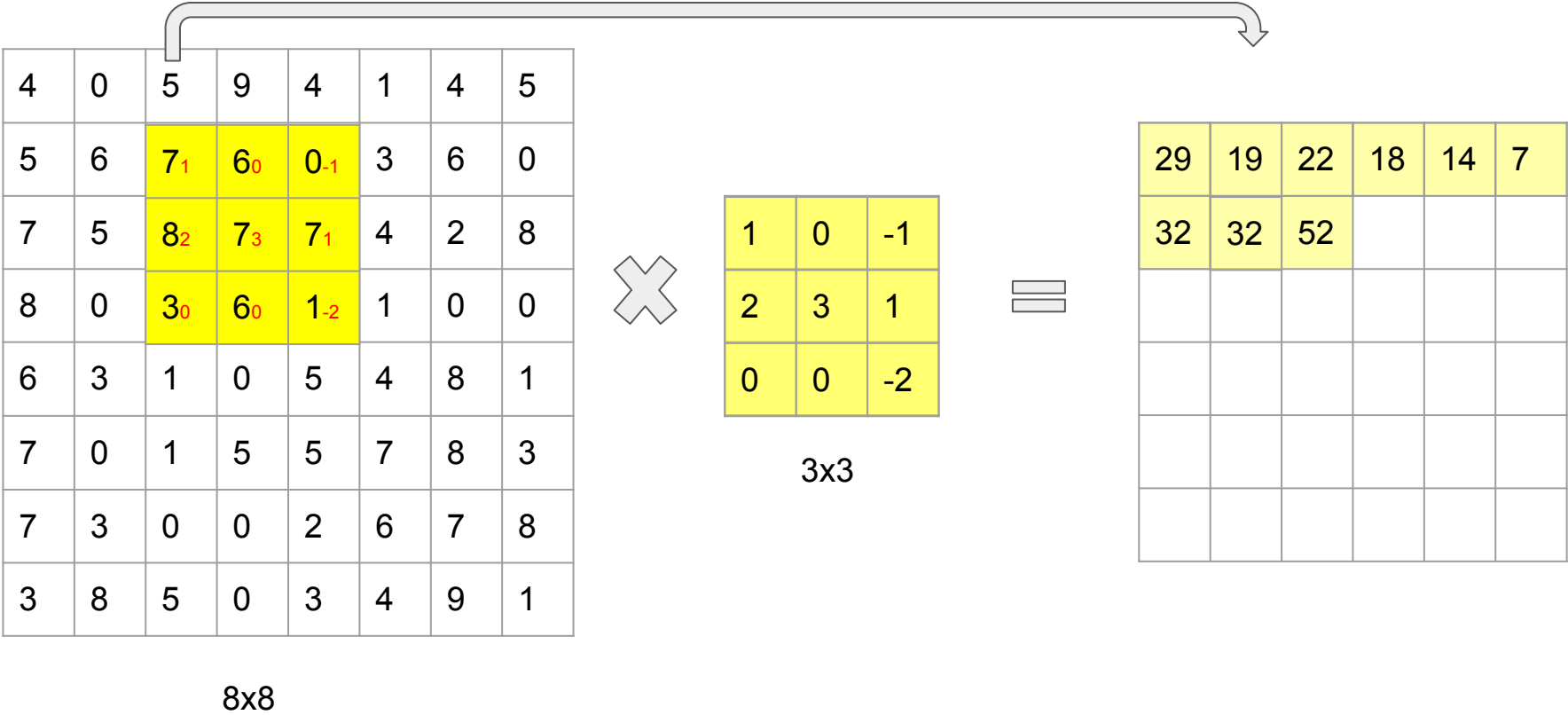
convolution on grey scale image



convolution on grey scale image



convolution on grey scale image



convolution on grey scale image

$(7*1)+(4*0)+(2*1)+(1*2)+(1*3)+(0*1)+(5*0)+(4*0)+(8*-2)=-6 \longrightarrow \text{relu}(-6+3)=\text{relu}(-3)=0$

4	0	5	9	4	1	4	5
5	6	7	6	0	3	6	0
7	5	8	7	7 ₁	4 ₀	2 ₋₁	8
8	0	3	6	1 ₂	1 ₃	0 ₁	0
6	3	1	0	5 ₀	4 ₀	8 ₋₂	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



1	0	-1
2	3	1
0	0	-2

3x3



29	19	22	18	14	7
32	32	52	43	25	28
19	16	19	14	0	

convolution on grey scale image

$(4*1)+(2*0)+(8*-1)+(1*2)+(0*3)+(0*1)+(4*0)+(8*0)+(1*-2)=-4 \longrightarrow \text{relu}(-4+3)=\text{relu}(-1)=0$

4	0	5	9	4	1	4	5
5	6	7	6	0	3	6	0
7	5	8	7	7	4 ₁	2 ₀	8 ₋₁
8	0	3	6	1	1 ₂	0 ₃	0 ₁
6	3	1	0	5	4 ₀	8 ₀	1 ₋₂
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



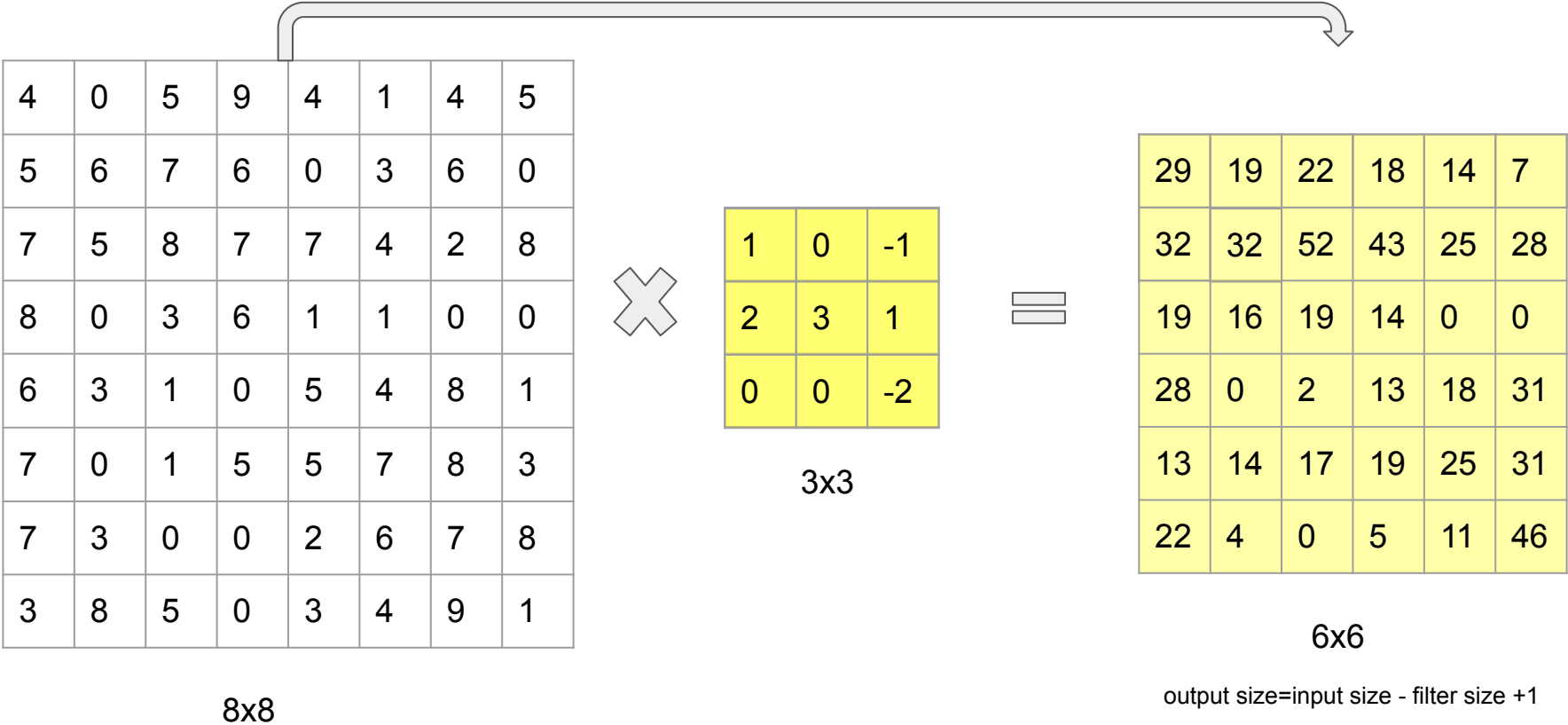
1	0	-1
2	3	1
0	0	-2

3x3



29	19	22	18	14	7
32	32	52	43	25	28
19	16	19	14	0	0

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



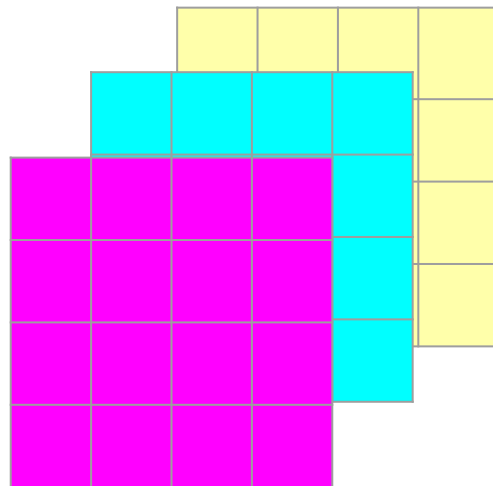
1	0	-1
2	3	1
0	0	-2

1	0	1
1	-1	1
1	0	2

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

=

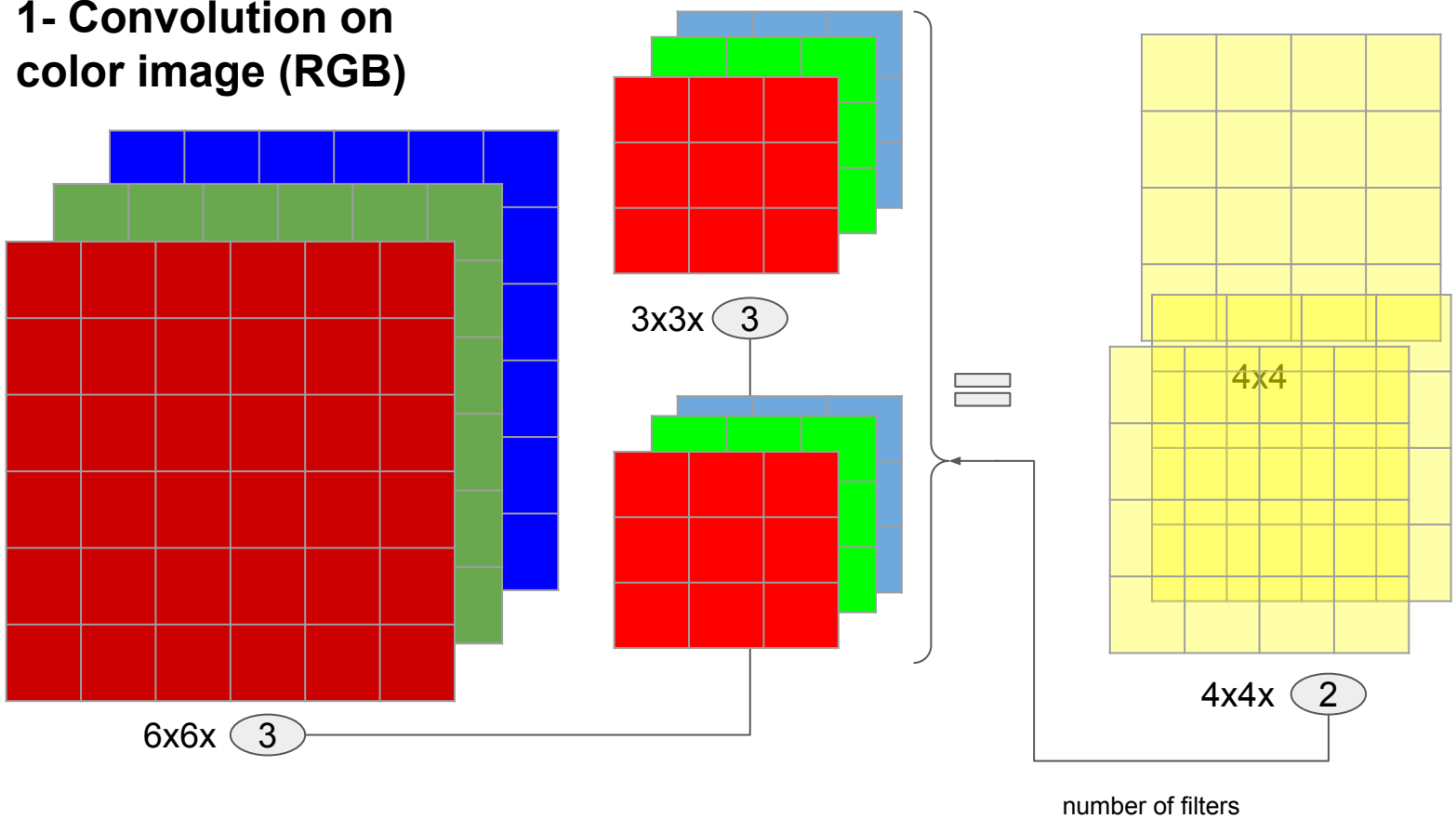
number of filters



4x4x3

3

1- Convolution on color image (RGB)



3 - Padding

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

6x6
0x0



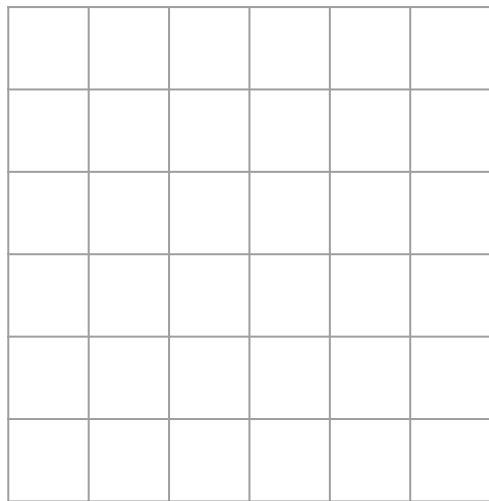
3x3



6x6

p=1

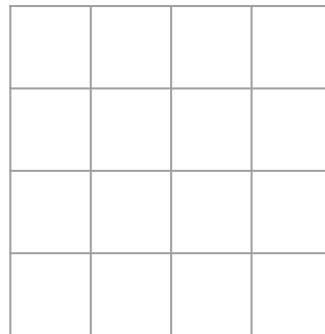
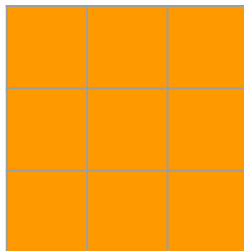
4 - stride



6x6

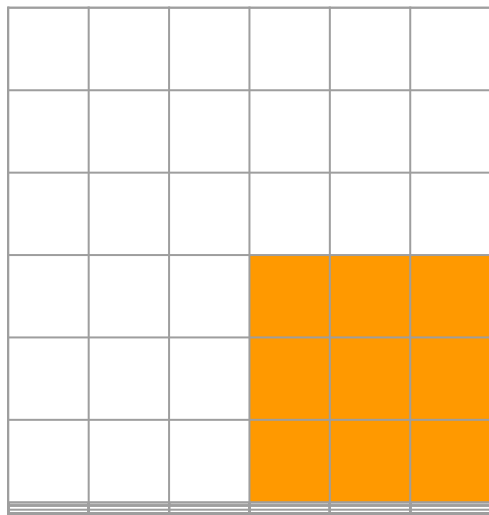


stride = 1



4x4

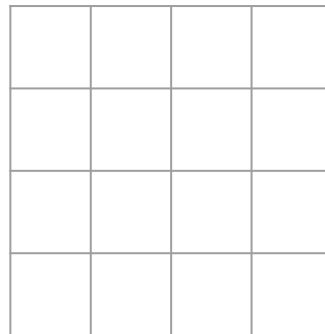
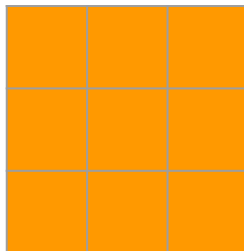
4 - stride



6x6

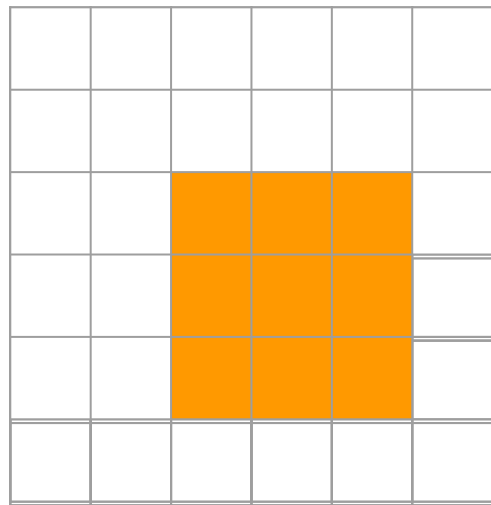


stride = 1



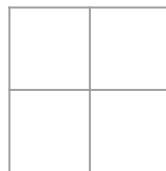
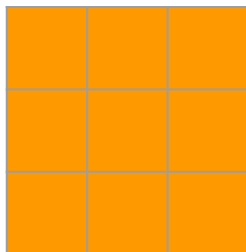
4x4

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

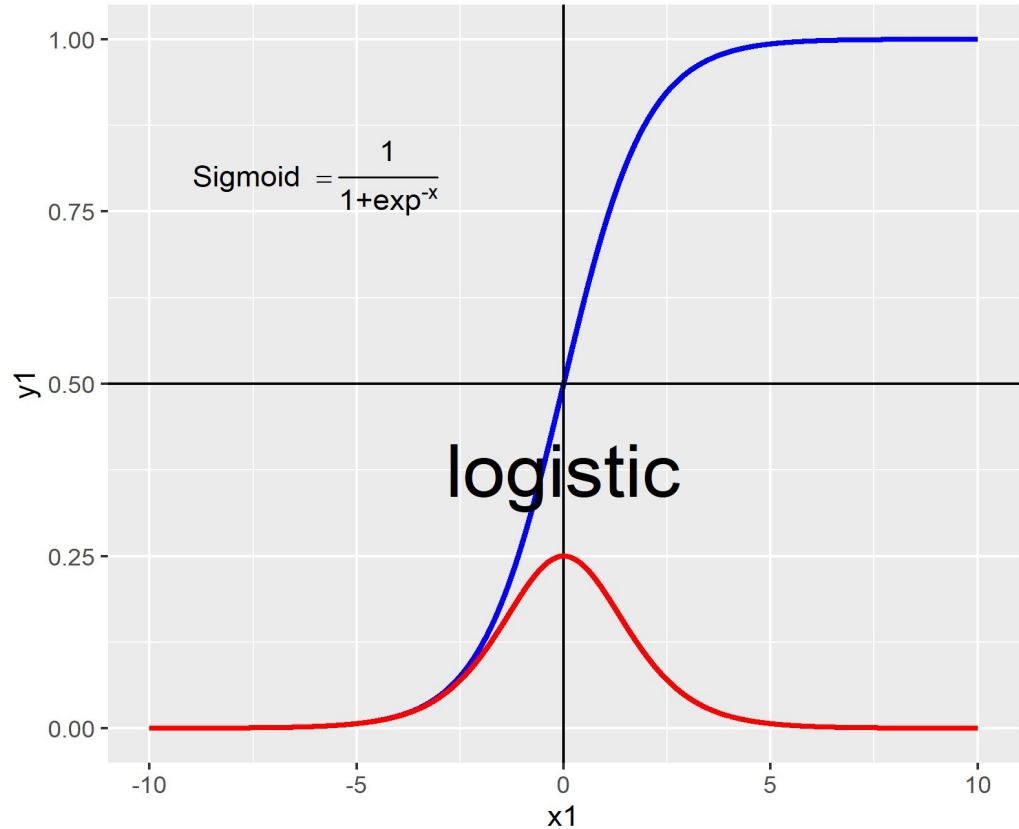
64x1



ANN

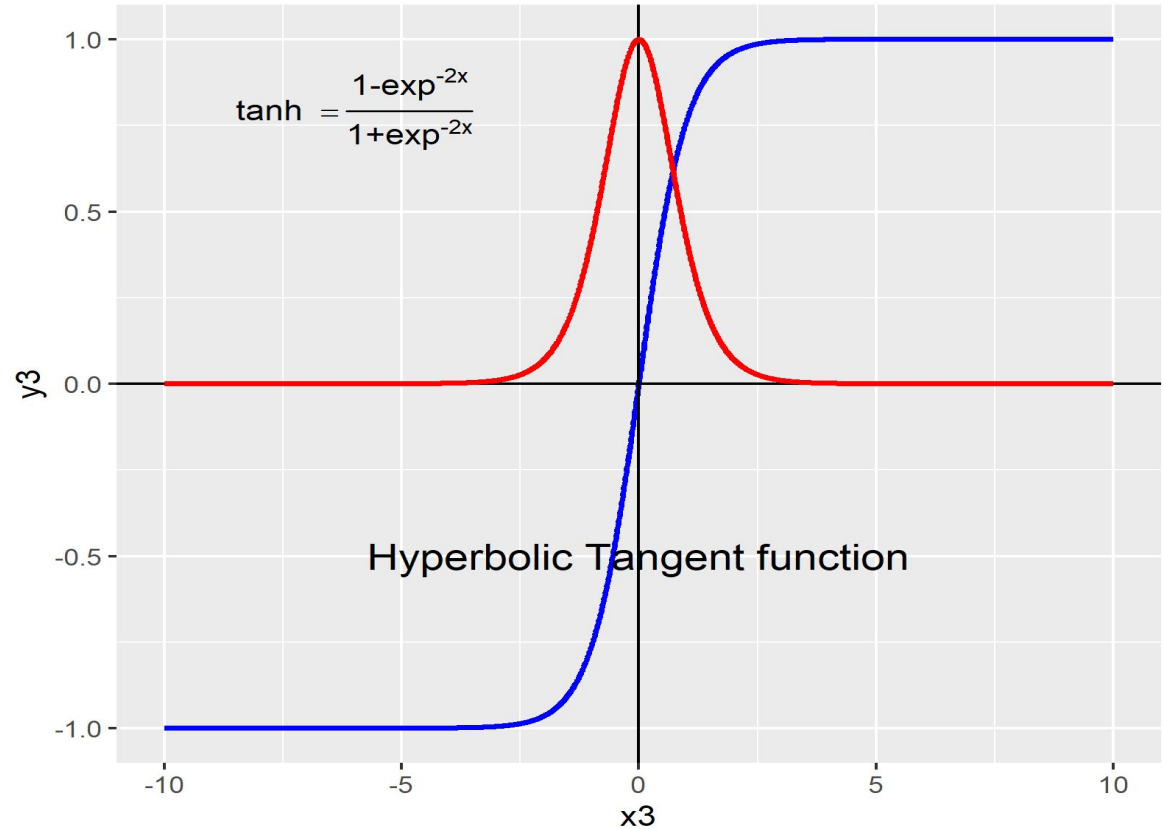
7 - Activation functions

- Sigmoid function :



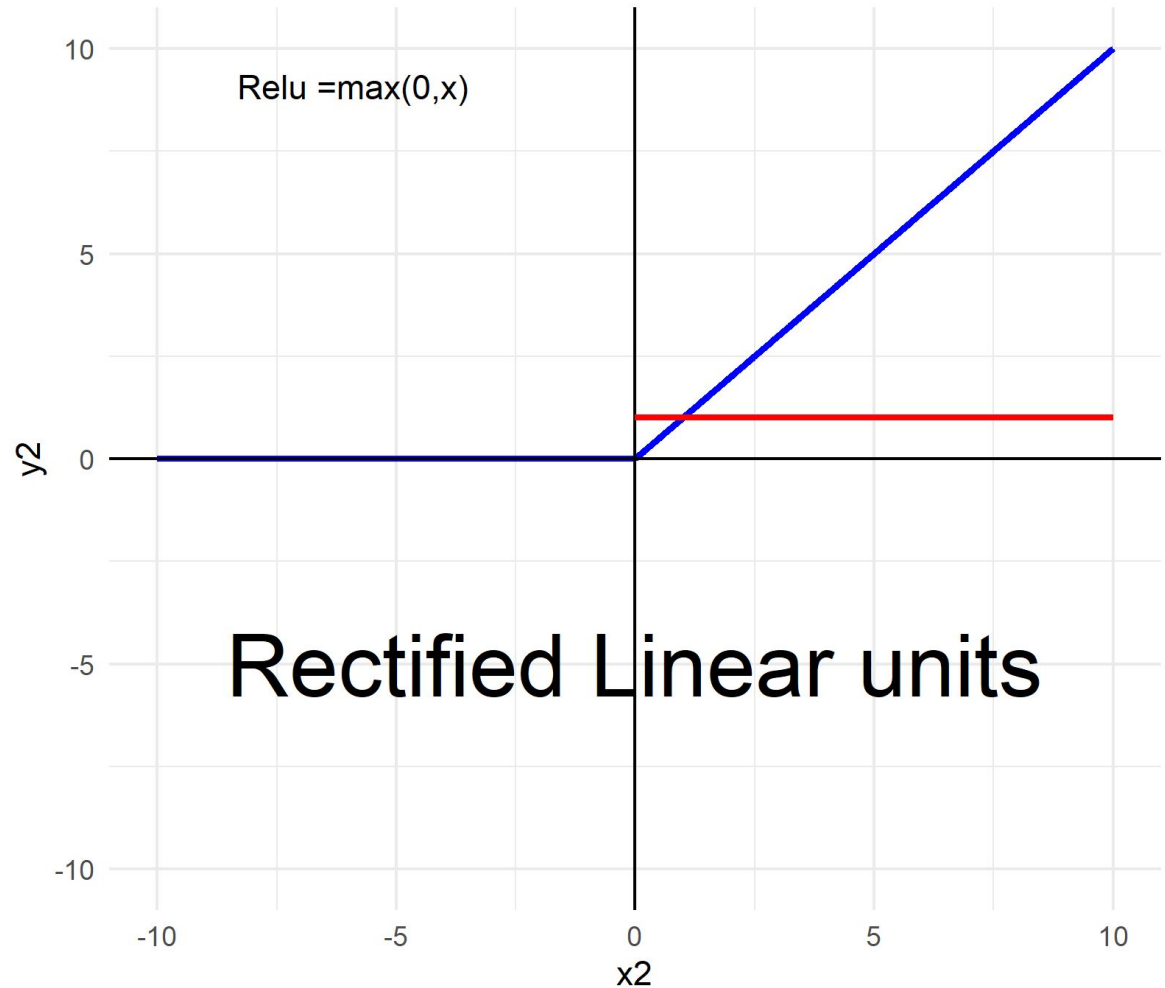
7 - Activation functions

- Tanh function :



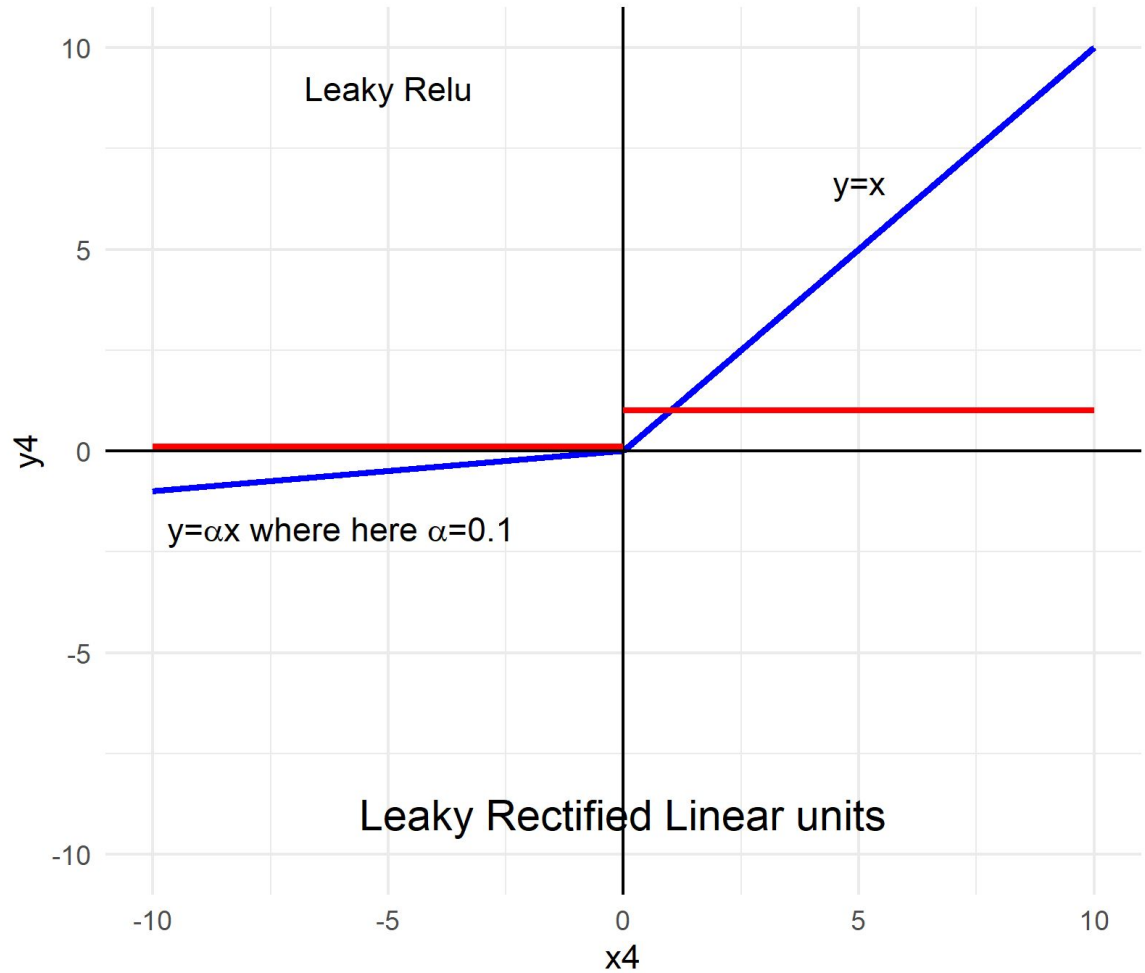
7 - Activation functions

- Relu function:

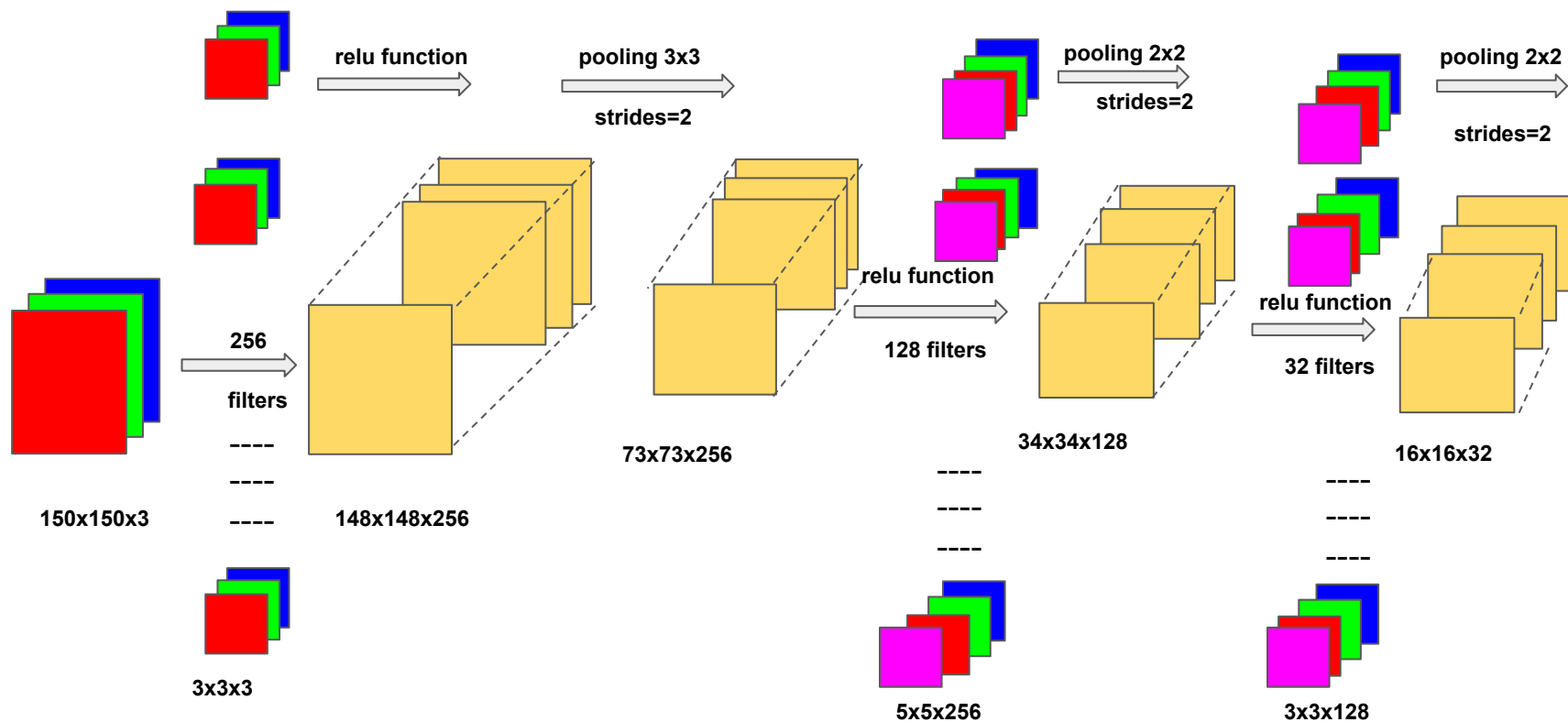


7 - Activation functions

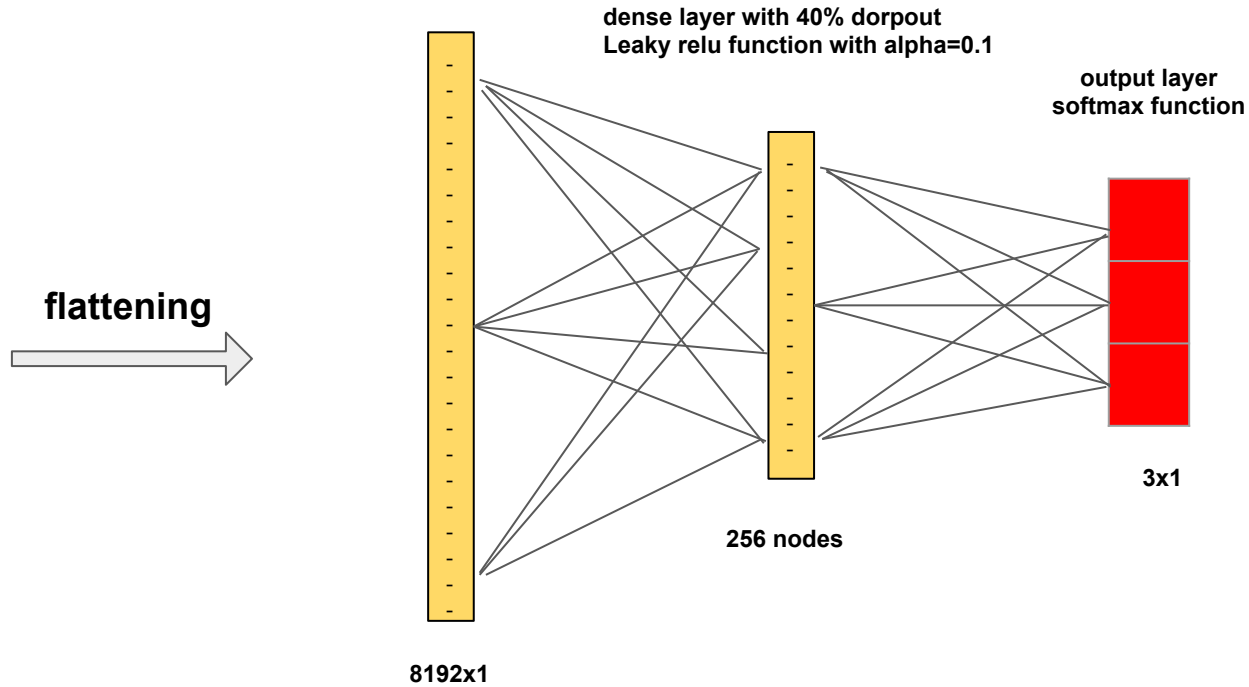
- Leaky relu function:



8- Architecture of the CNN model



8- Architecture of the CNN model



Fully connected layers