Convolutional Neural Network

--- Zhongwu xie

1 . Types of layers in a convolutional network.

- -Convolution
- -Pooling
- -Fully connected

1 . Dot Product(scalar product)

The dot product of two vectors $\vec{a} = [a_1, a_2, \dots, a_n]$ and $\vec{b} = [b_1, b_2, \dots, b_n]$ is defined as:

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

e.g. $\vec{a} = [1,3,-5]$, $\vec{b} = [4,-2,-1]$

$$\vec{a} \cdot \vec{b} = \begin{bmatrix} 1 & 3 & -5 \end{bmatrix} \begin{bmatrix} 4 \\ -2 \\ -1 \end{bmatrix} = 1 \times 4 + 3 \times (-2) + (-5) \times (-1) = 3$$

2 Convolution in Neural Network

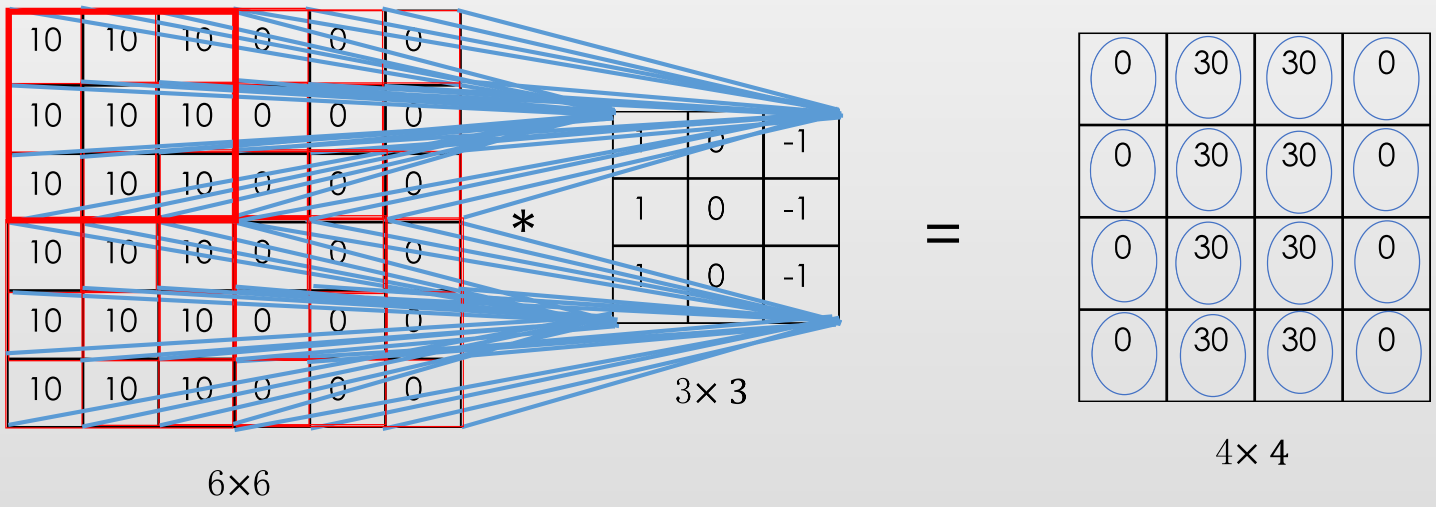
$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$

$$10\times1+10\times0+10\times(-1)+10\times1+10\times0+10\times(-1)+10\times1+10\times0+10\times(-1)=0$$

$$\begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \boxed{0}$$

Then slide the local receptive field across the entire input image.

2.1 Convolution in Neural Network



Notation:

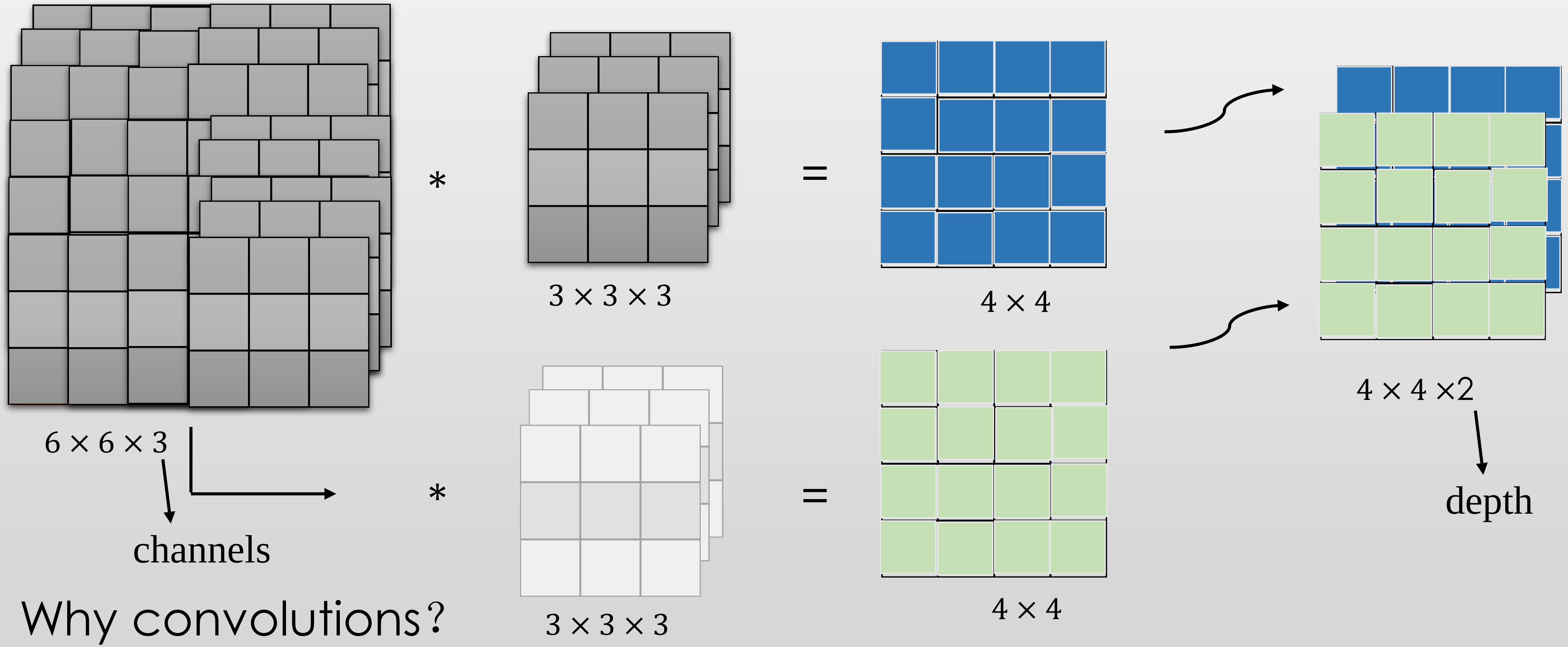
Image: n×n   filter: f×f

padding : p   stride : s

Then output:

$$\left\lfloor \frac{n+2p-f}{s}+1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s}+1 \right\rfloor$$

2.2 Multiple filters



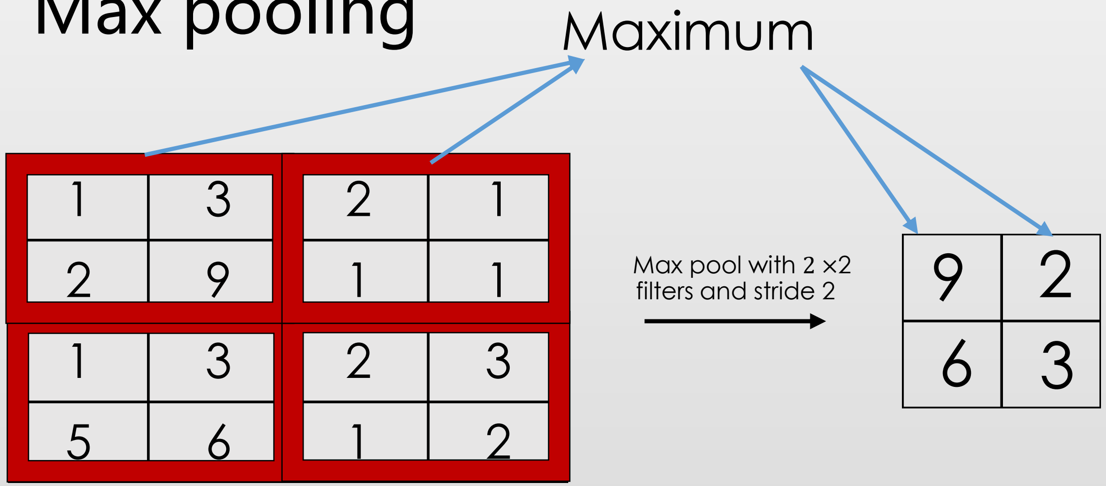
Why convolutions?

---Parameter sharing

---Sparsity of connections

3 . Pooling layers ---Shrinking the image stack

- Max pooling



Pooling:

1.Pick a window size(usually 2 or 3)

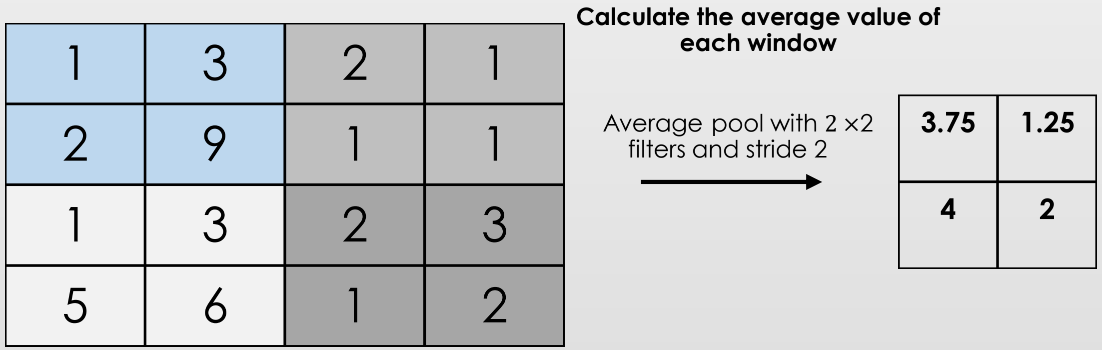
2.Pick a stride(usually 2)

3.Walk your window across your filtered images.

4.From each window , take the maximum value.

3 . Pooling layers ---Shrinking the image stack

- Average pooling



- Remove the redundancy information of convolutional layer .

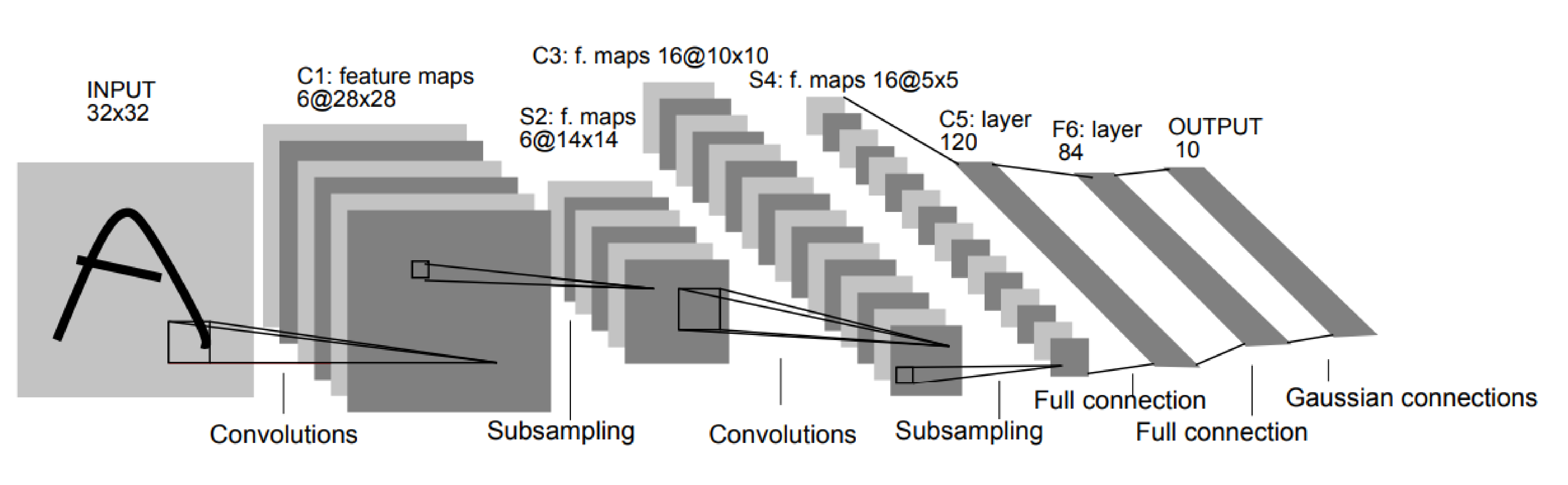
---By having less spatial information you gain computation performance

---Less spatial information also means less parameters, so less chance to over-fit

---You get some translation invariance.

3 . Full connection layer

The CNNs help extract certain features from the image , then fully connected layer is able to generalize from these features into the output-space.

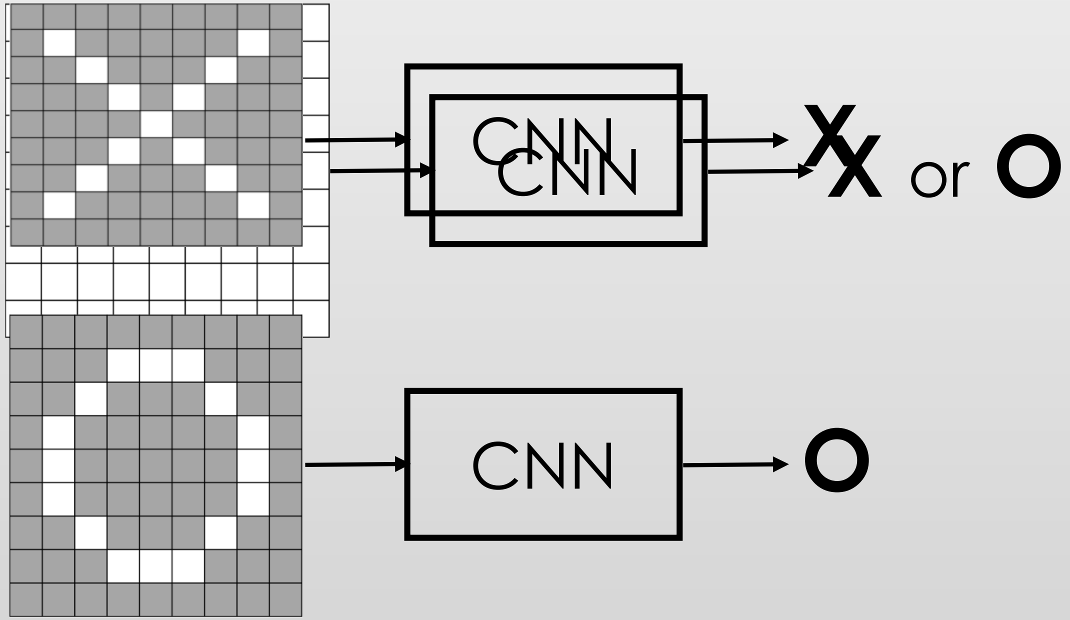


[LeCun et al.,1998.Gradient-based learning applied to document recognition.]

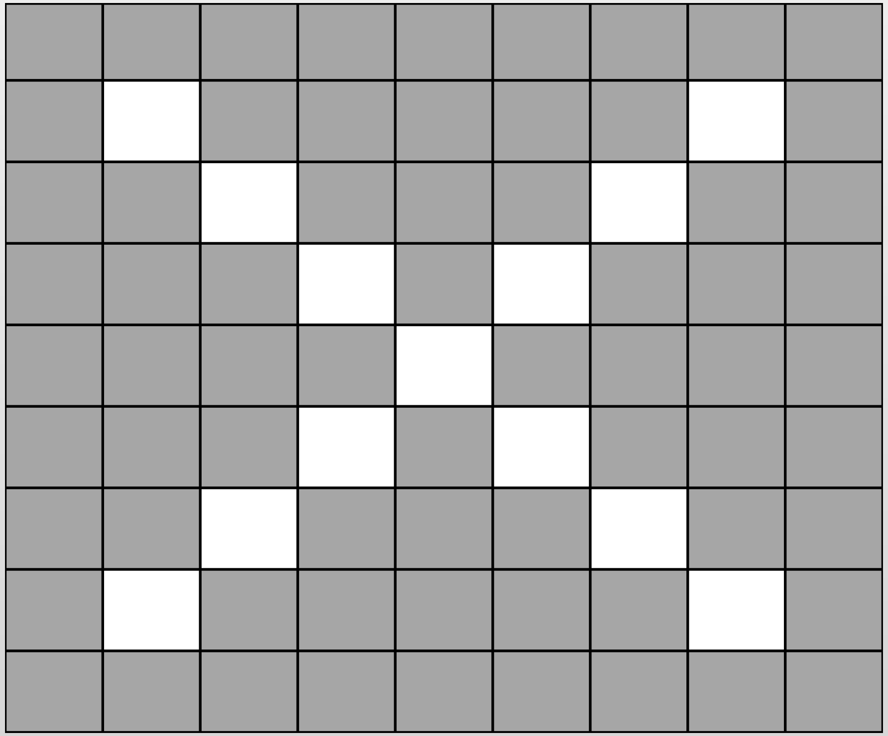
4 . For example

Say whether a picture Is of an X or O.

A two-dimensional array of pixels

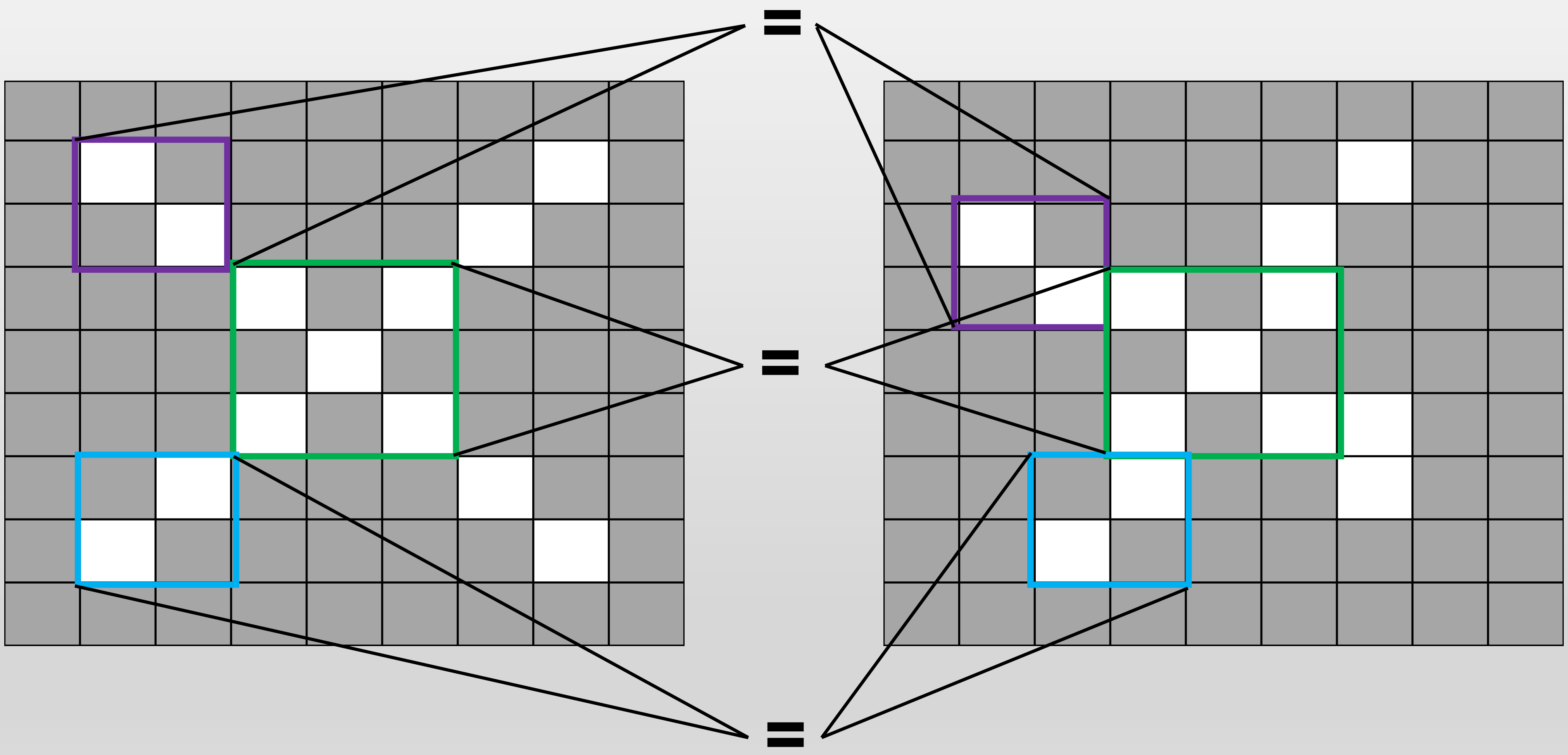


4 . 1 What the computer see

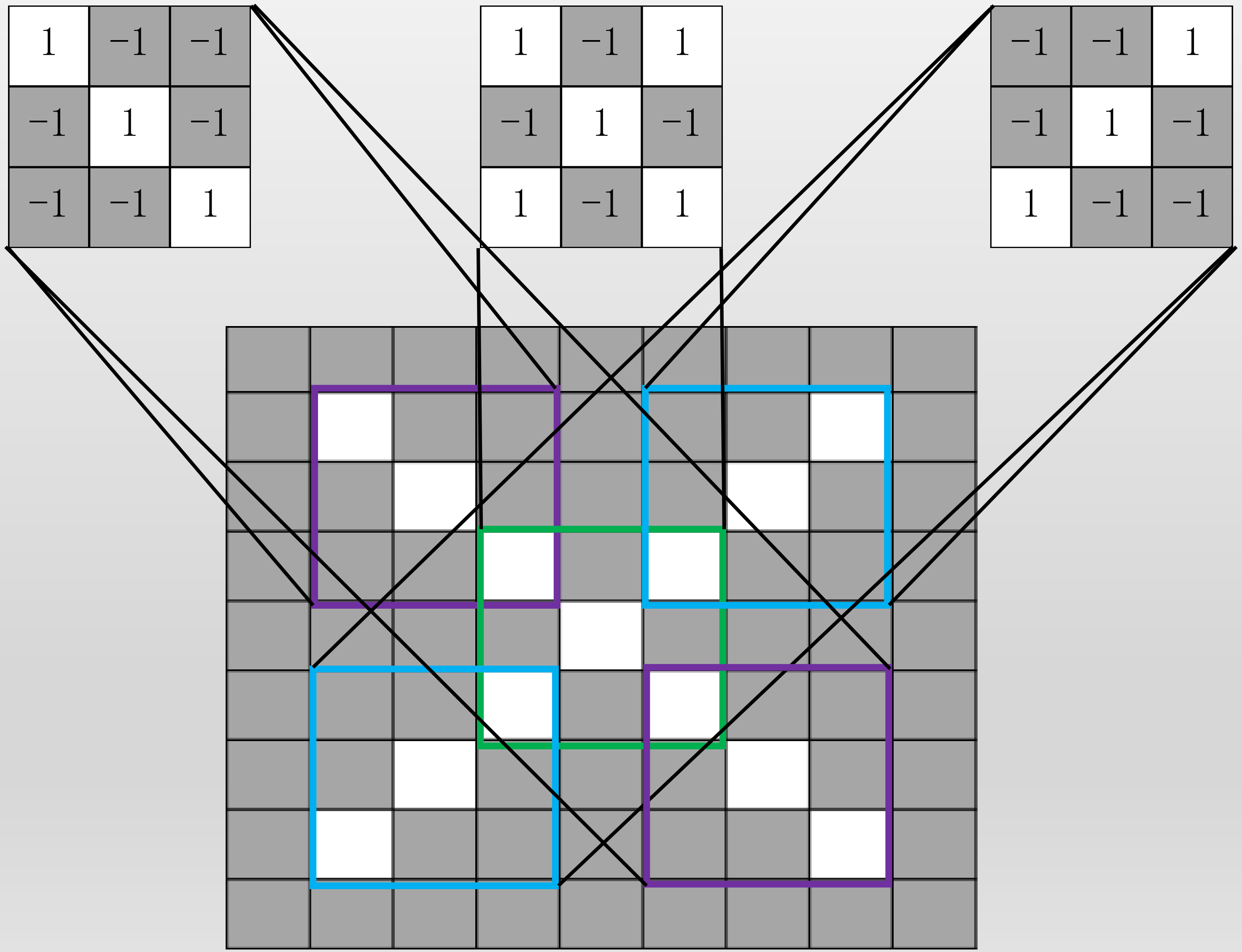


−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	1	−1	−1	−1	−1	−1	1	−1
−1	−1	1	−1	−1	−1	1	−1	−1
−1	−1	−1	1	−1	1	−1	−1	−1
−1	−1	−1	−1	1	−1	−1	−1	−1
−1	−1	−1	1	−1	1	−1	−1	−1
−1	−1	1	−1	−1	−1	1	−1	−1
−1	1	−1	−1	−1	−1	−1	1	−1
−1	−1	−1	−1	−1	−1	−1	−1	−1

4 . 1 ConvNets match pieces of the image



4 . 1 Features match pieces of the image



4 . Filtering : The math behind the match

1	−1	−1
−1	1	−1
−1	−1	1

−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	1	−1	−1	−1	−1	−1	1	−1
−1	−1	1	−1	−1	−1	1	−1	−1
−1	−1	−1	1	−1	1	−1	−1	−1
−1	−1	−1	−1	1	−1	−1	−1	−1
−1	−1	−1	1	−1	1	−1	−1	−1
−1	−1	1	−1	−1	−1	1	−1	−1
−1	1	−1	−1	−1	−1	−1	1	−1
−1	−1	−1	−1	−1	−1	−1	−1	−1

9×9

7	-1	1	3	5	-1	3
-1	9	-1	3	-1	1	-1
1	-1	9	-3	1	-1	5
3	3	-3	5	-3	3	3
5	-1	1	3	9	-1	1
-1	1	-1	3	-1	9	-1
3	-1	5	3	1	-1	7

7×7

4 . Filtering : The math behind the match

−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	1	−1	−1	−1	−1	−1	1	−1
−1	−1	1	−1	−1	−1	1	−1	−1
−1	−1	−1	1	−1	1	−1	−1	−1
−1	−1	−1	−1	1	−1	−1	−1	−1
−1	−1	−1	1	−1	1	−1	−1	−1
−1	−1	1	−1	−1	−1	1	−1	−1
−1	1	−1	−1	−1	−1	−1	1	−1
−1	−1	−1	−1	−1	−1	−1	−1	−1

$\otimes$

1	−1	−1
−1	1	−1
−1	−1	1

=

7	-1	1	3	5	-1	3
-1	9	-1	3	-1	1	-1
1	-1	9	-3	1	-1	5
3	3	-3	5	-3	3	3
5	-1	1	3	9	-1	1
-1	1	-1	3	-1	9	-1
3	-1	5	3	1	-1	7

4 . Filtering : The math behind the match

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\otimes$

1	-1	-1
-1	1	-1
-1	-1	1

=

7	-1	1	3	5	-1	3
-1	9	-1	3	-1	1	-1
1	-1	9	-3	1	-1	5
3	3	-3	5	-3	3	3
5	-1	1	3	9	-1	1
-1	1	-1	3	-1	9	-1
3	-1	5	3	1	-1	7

$\otimes$

1	-1	1
-1	1	-1
1	-1	1

=

3	-5	1	-1	1	-5	3
-5	5	-5	2	-5	5	-5
1	-5	5	-7	5	-5	1
-1	3	-7	9	7	3	-1
1	-5	5	-7	5	-5	1
-5	5	-5	3	-5	5	-5
3	-5	1	-1	1	-5	3

$\otimes$

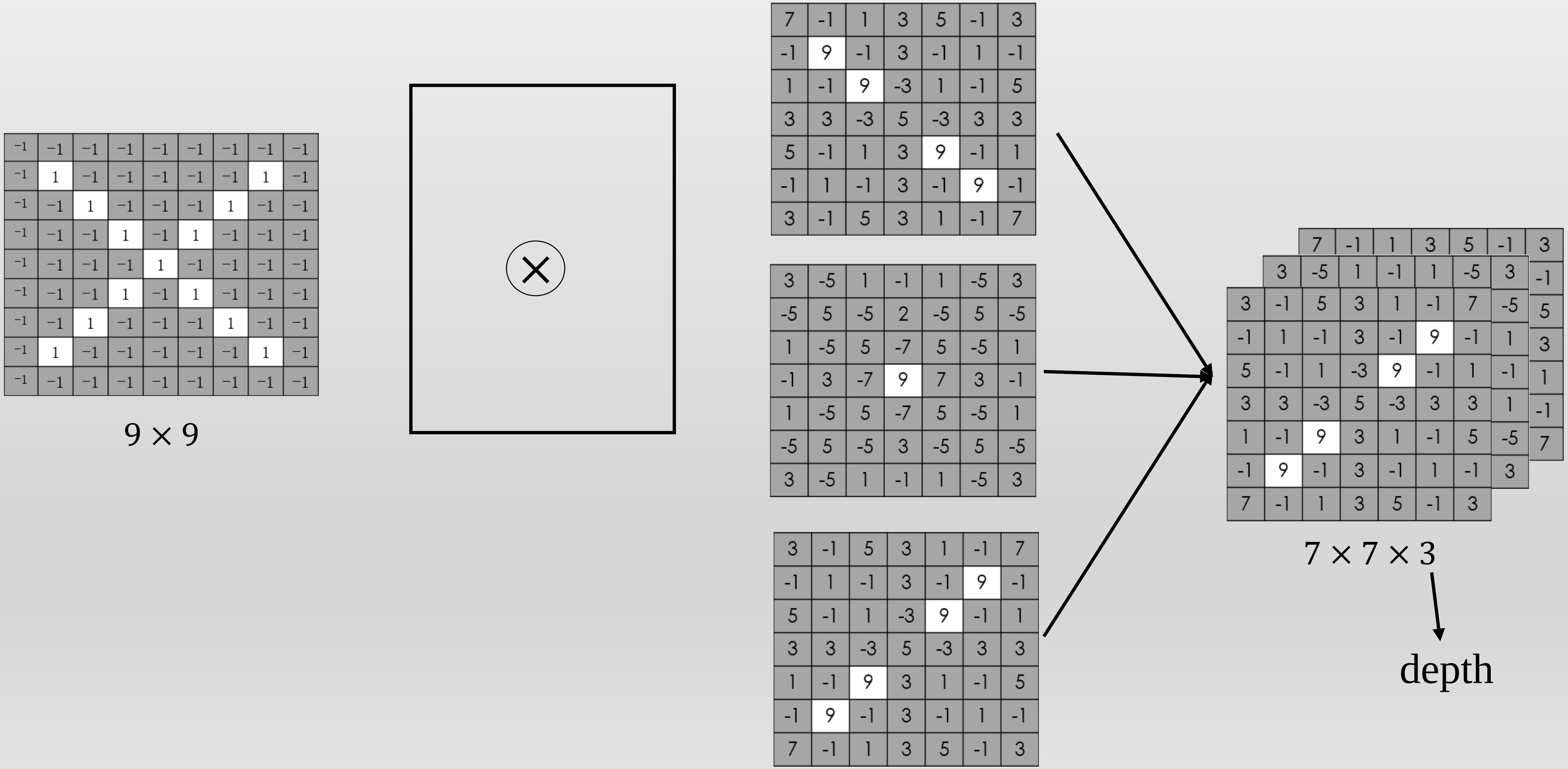
-1	-1	1
-1	1	-1
1	-1	-1

=

3	-1	5	3	1	-1	7
-1	1	-1	3	-1	9	-1
5	-1	1	-3	9	-1	1
3	3	-3	5	-3	3	3
1	-1	9	3	1	-1	5
-1	9	-1	3	-1	1	-1
7	-1	1	3	5	-1	3

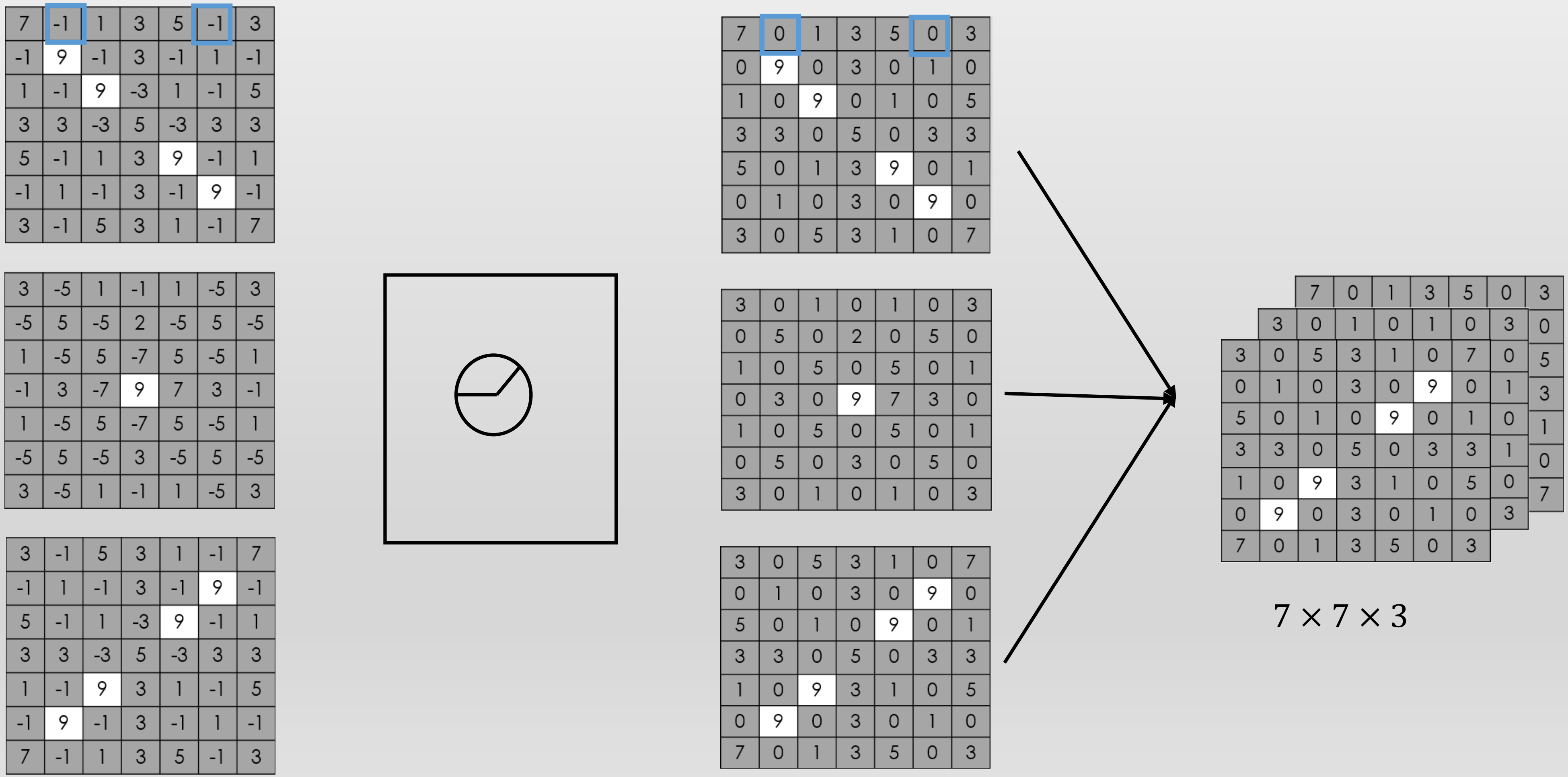
4 . Convolution layer

---One image becomes a stack of filtered images



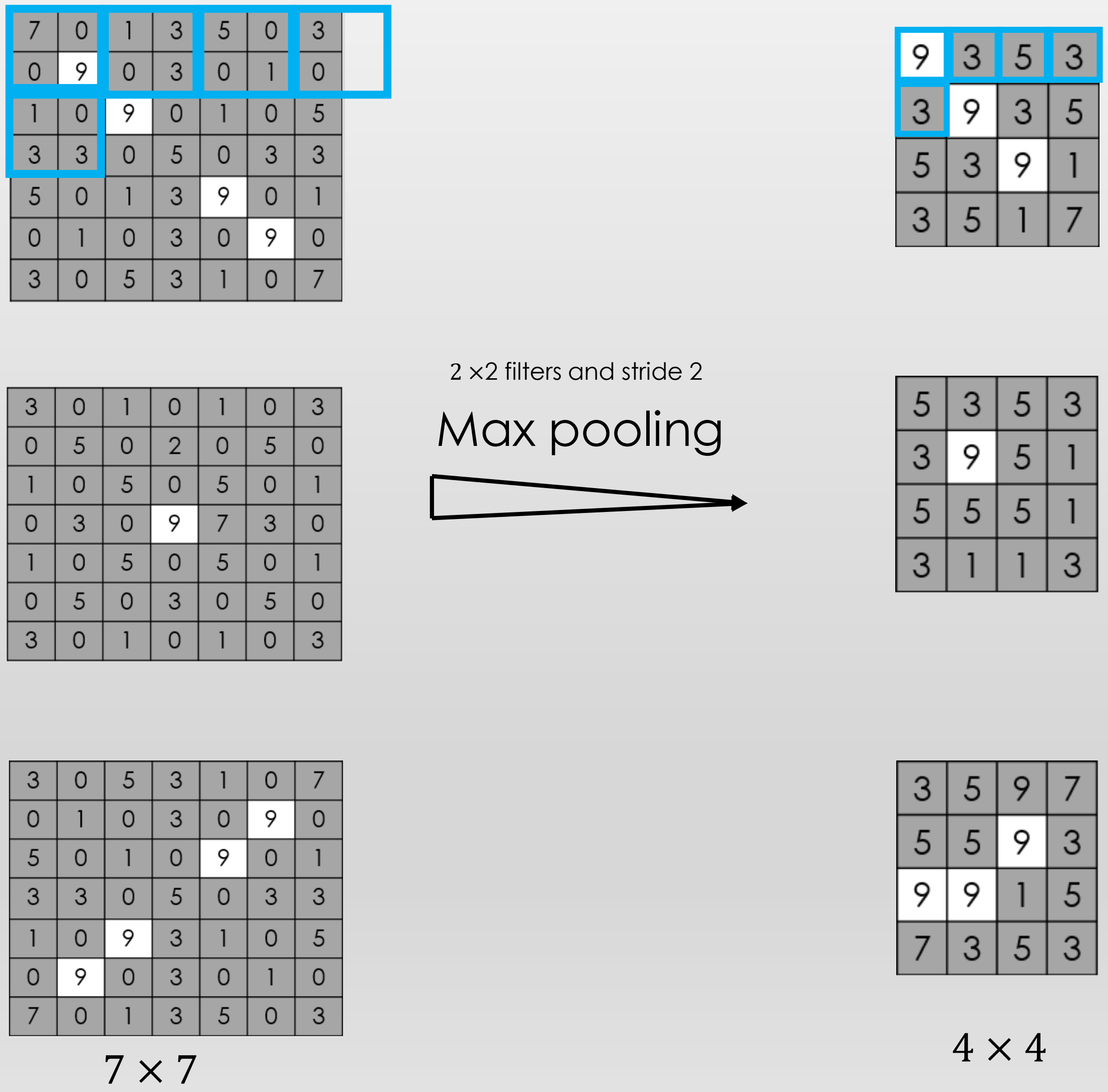
4 . Relu layer

A stack of images becomes a stack of images with no negative values.



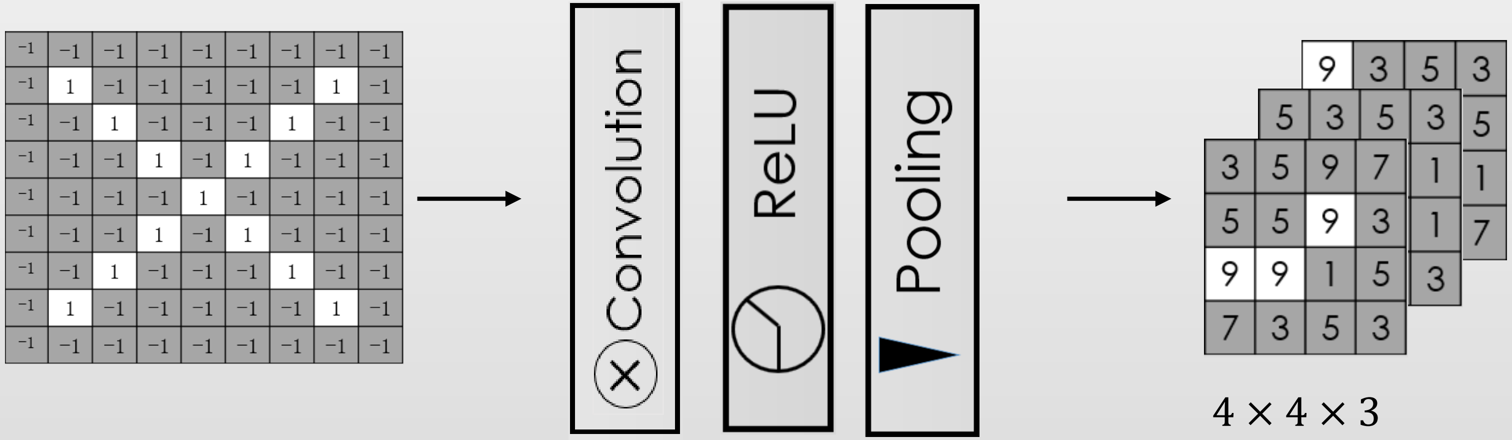
4 . Pooling layer

---A stack of images becomes a stack of smaller images

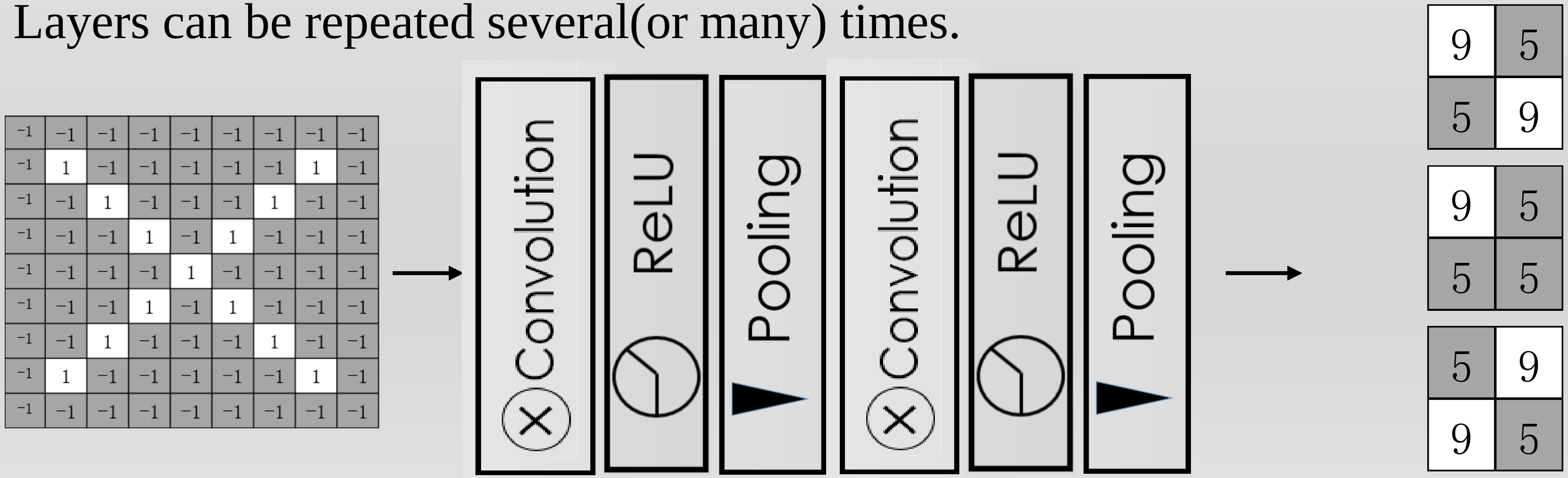


4 . Layers get stacked

The output of one becomes the input of the next.



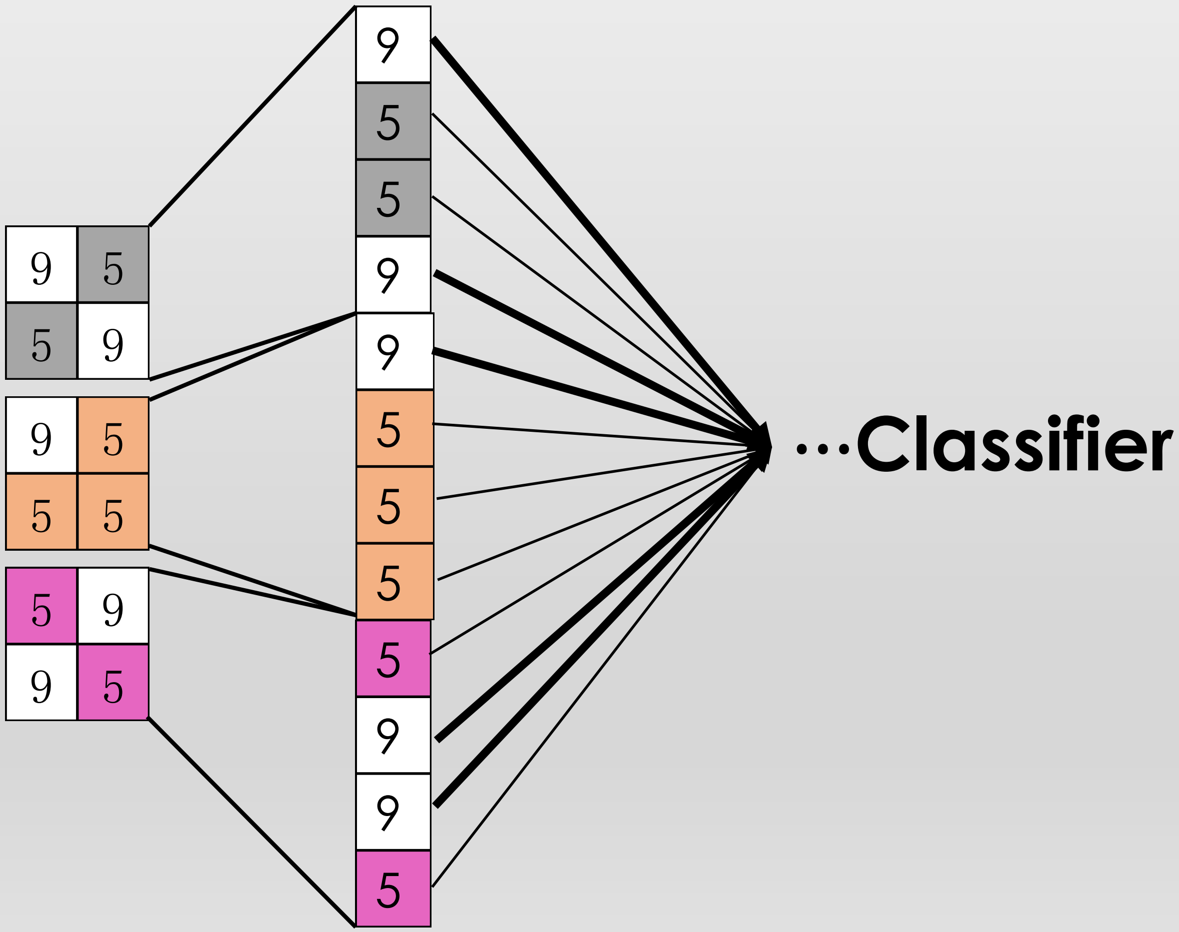
Layers can be repeated several(or many) times.



$2 \times 2 \times 3$

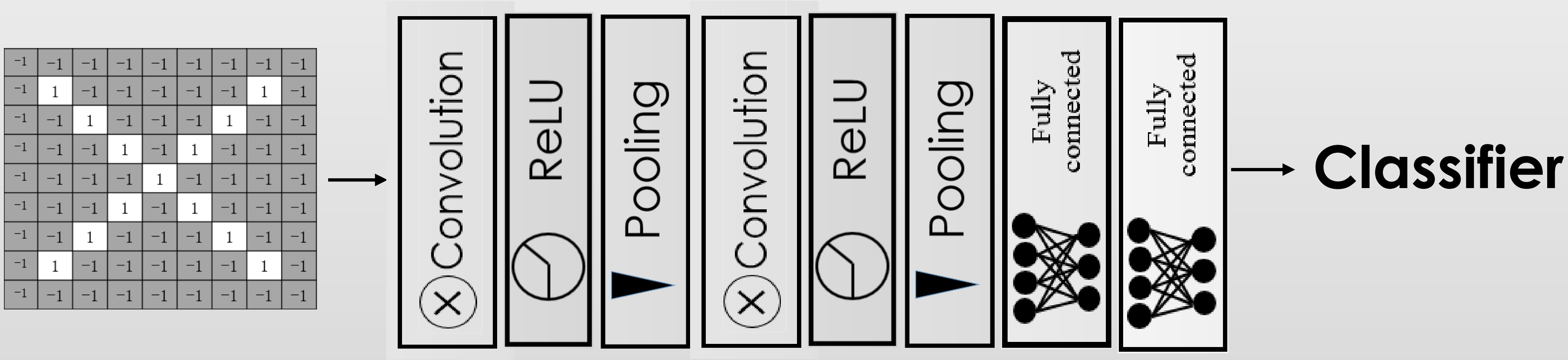
4 . Fully connected layer

Every value gets a vote---Vote depends on how strongly a value predicts X or O.



4 . Summary：Putting it all together

A set of pixels becomes a set of votes.



Learning

Q: Where do all the magic numbers come from？

Features in convolutional layers

Voting weights in fully connected layers

A:Backpropagation

Thank you