Deep Learning

Chapter 3 Convolutional Neural Network

Dr. Van-Toi NGUYEN

EEE, Phenikaa University

Chapter 3: Convolutional Neural Network

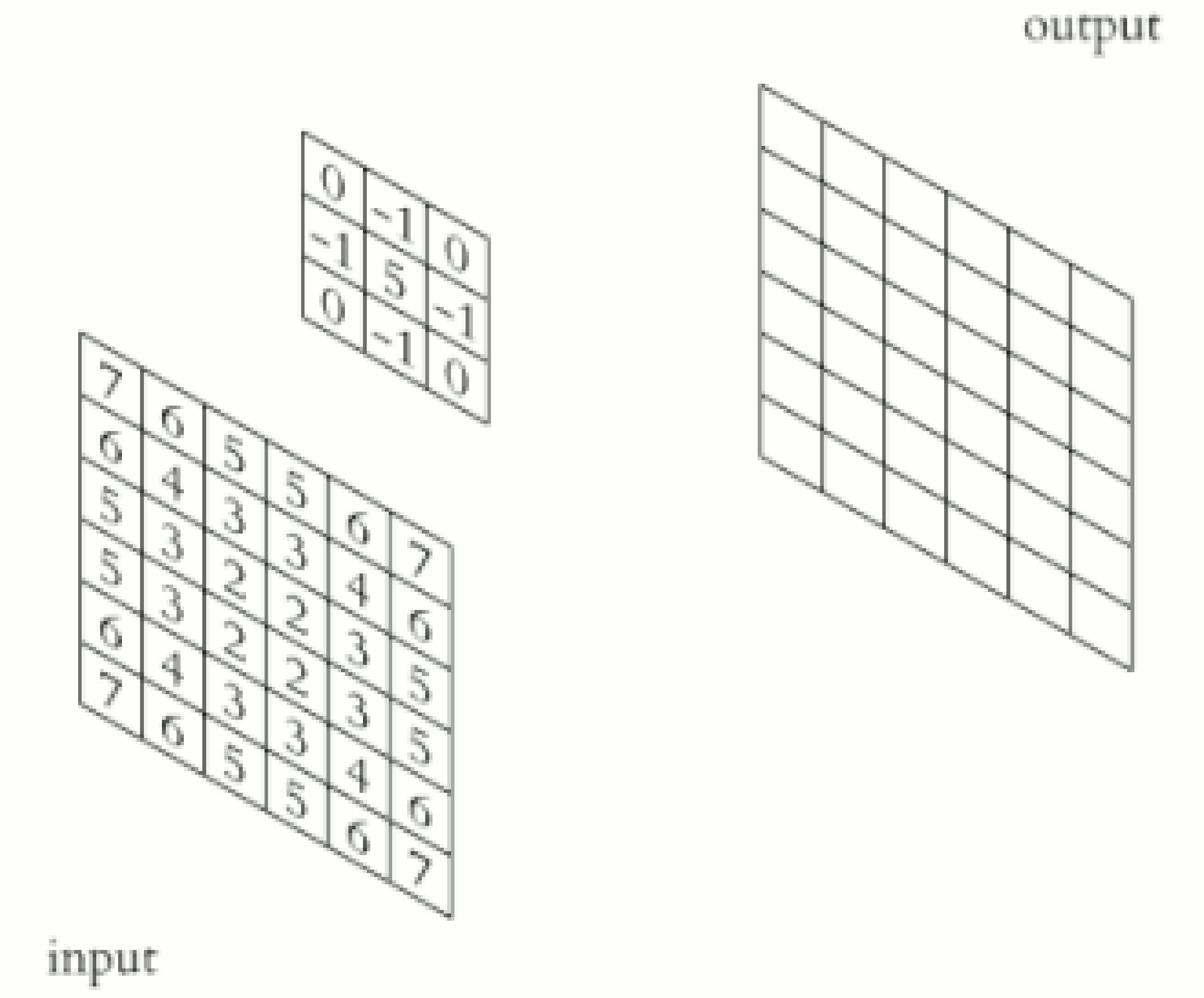
1. Convolutional operator
2. History of CNN
3. Deep Convolutional Models
4. Layers in CNN
5. Applications of CNN
6. Practice

3.1 Convolutional Operator

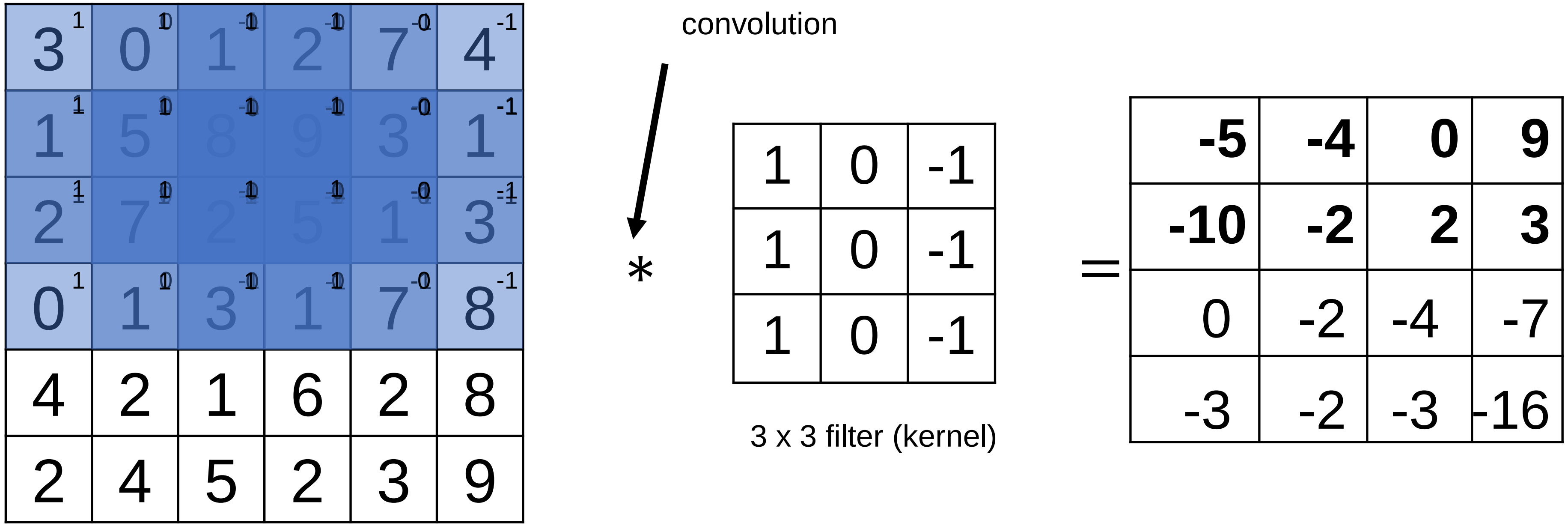
The general form for matrix convolution is

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} * \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{(m-i)(n-j)} y_{(1+i)(1+j)}$$

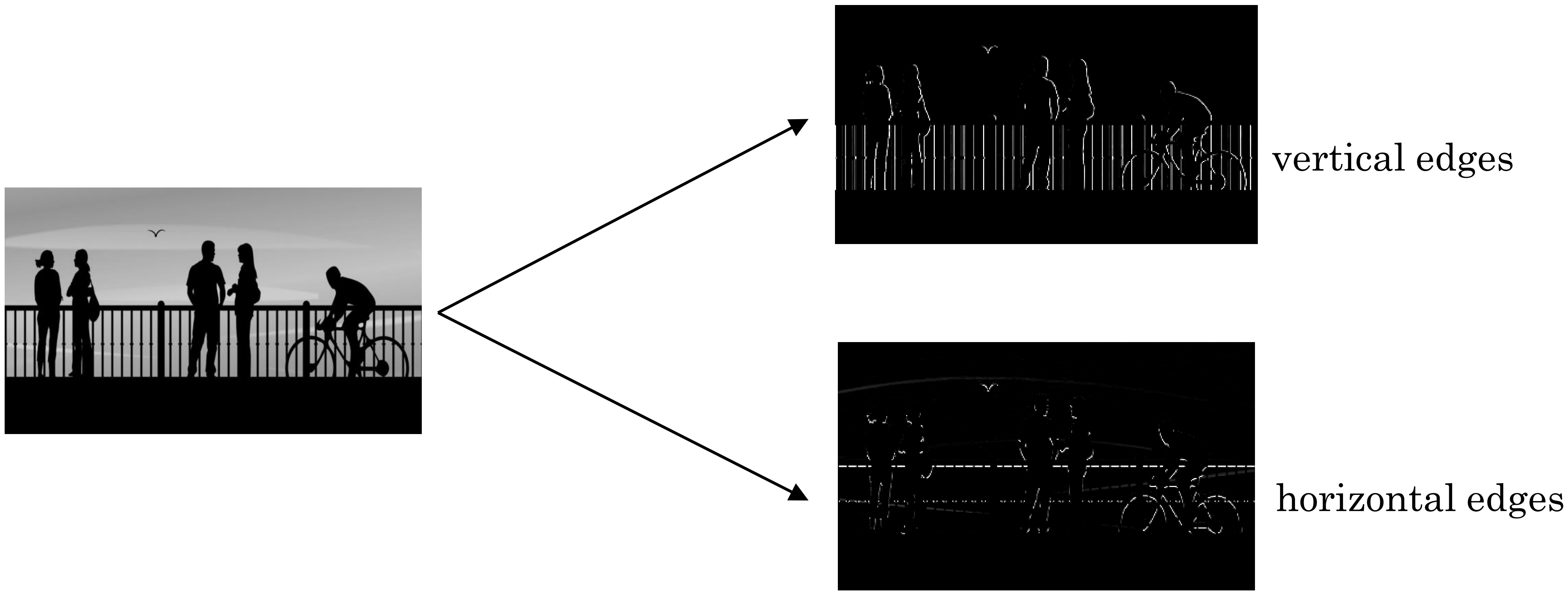
3.1 Convolutional Operator



3.1 Convolutional Operator



To Extract Edges in an Image



3.1 Convolutional Operator

To Extract Edges in an Image

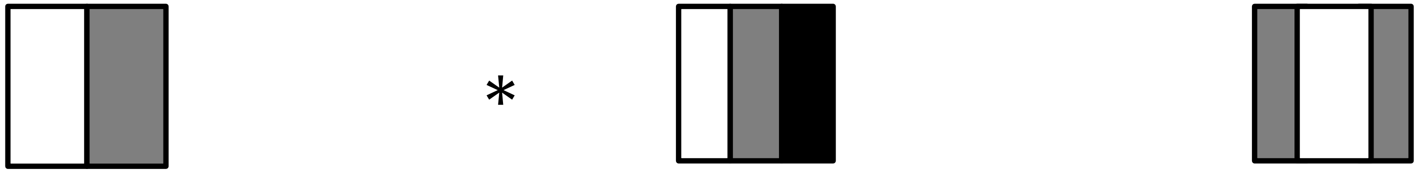
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

1	0	-1
1	0	-1
1	0	-1

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



3.1 Convolutional Operator

To Extract Edges in an Image

1	0	-1
1	0	-1
1	0	-1

Vertical

1	1	1
0	0	0
-1	-1	-1

Horizontal

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

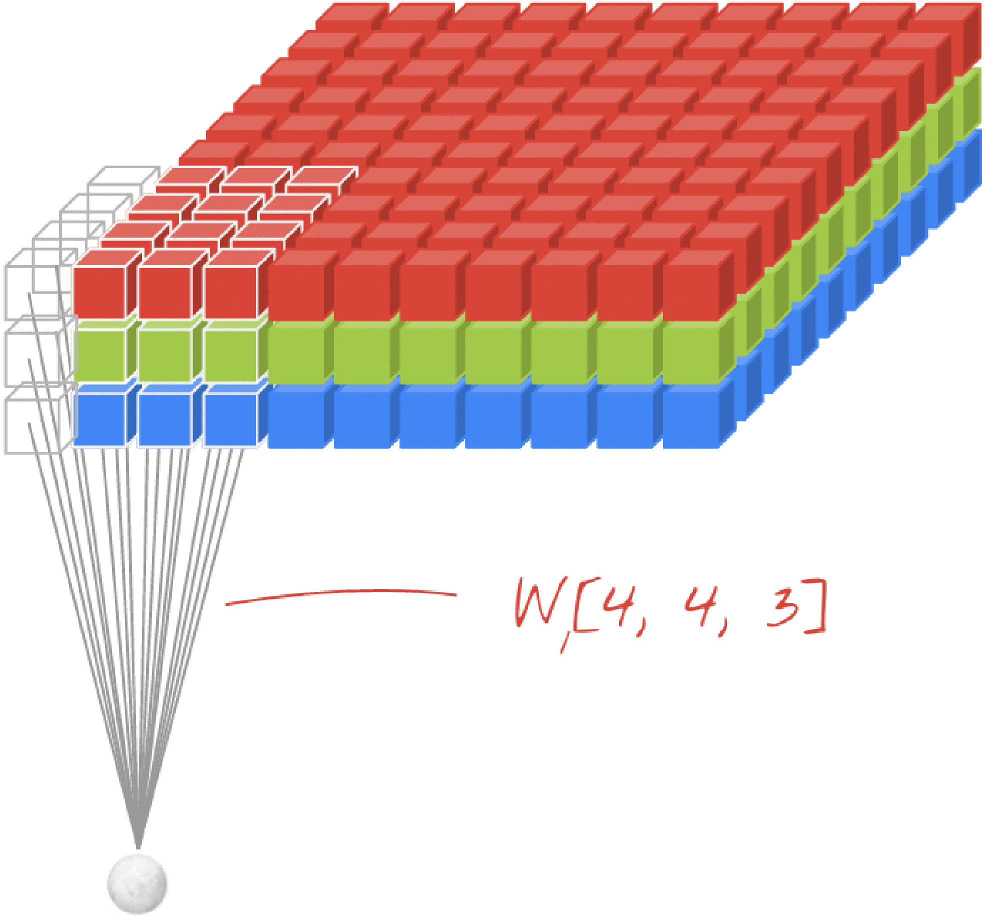
*

1	1	1
0	0	0
-1	-1	-1

=

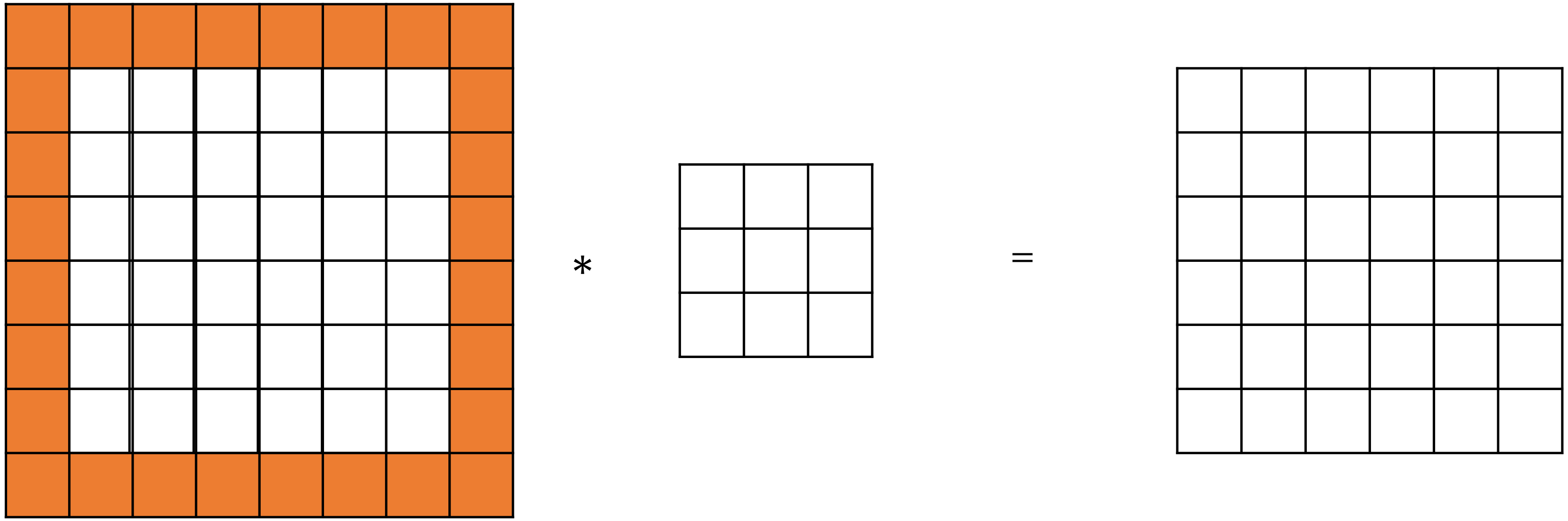
0	0	0	0
30	10	-10	-30
30	10	-10	-30
0	0	0	0

Convolutional layer (Conv)



3.4 Convolutional layer

Padding



Size of input data: $n \times n$, Size of filter: $f \times f$
Size of output: $(n - f + 1) \times (n - f + 1)$
Example: $6 - 3 + 1 = 4$, hence 4×4, size is reduced!

Use padding, extra border of 1 all around, ($p = 1$) gives output of
$(n + 2p - f + 1) \times (n + 2p - f + 1) = 6 \times 6$ (same as original data size)

Padding

- "**Valid**" (no padding): $n \times n * f \times f \to (n-f+1) \times (n-f+1)$
  - $6 \times 6 * 3 \times 3 \to 4 \times 4$
- "**Same**": Pad so that output size is the same as the input size
  - $n + 2p - f + 1 = n \Rightarrow p = \frac{f-1}{2}$
  - $f = 3, p = \frac{3-1}{2} = 1,\ or\ f = 5, p = 2$

PHENIKAA UNIVERSITY

Stride

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

7 x 7

$*$

3	4	4
1	0	2
-1	0	3

3 x 3

Stride = 2

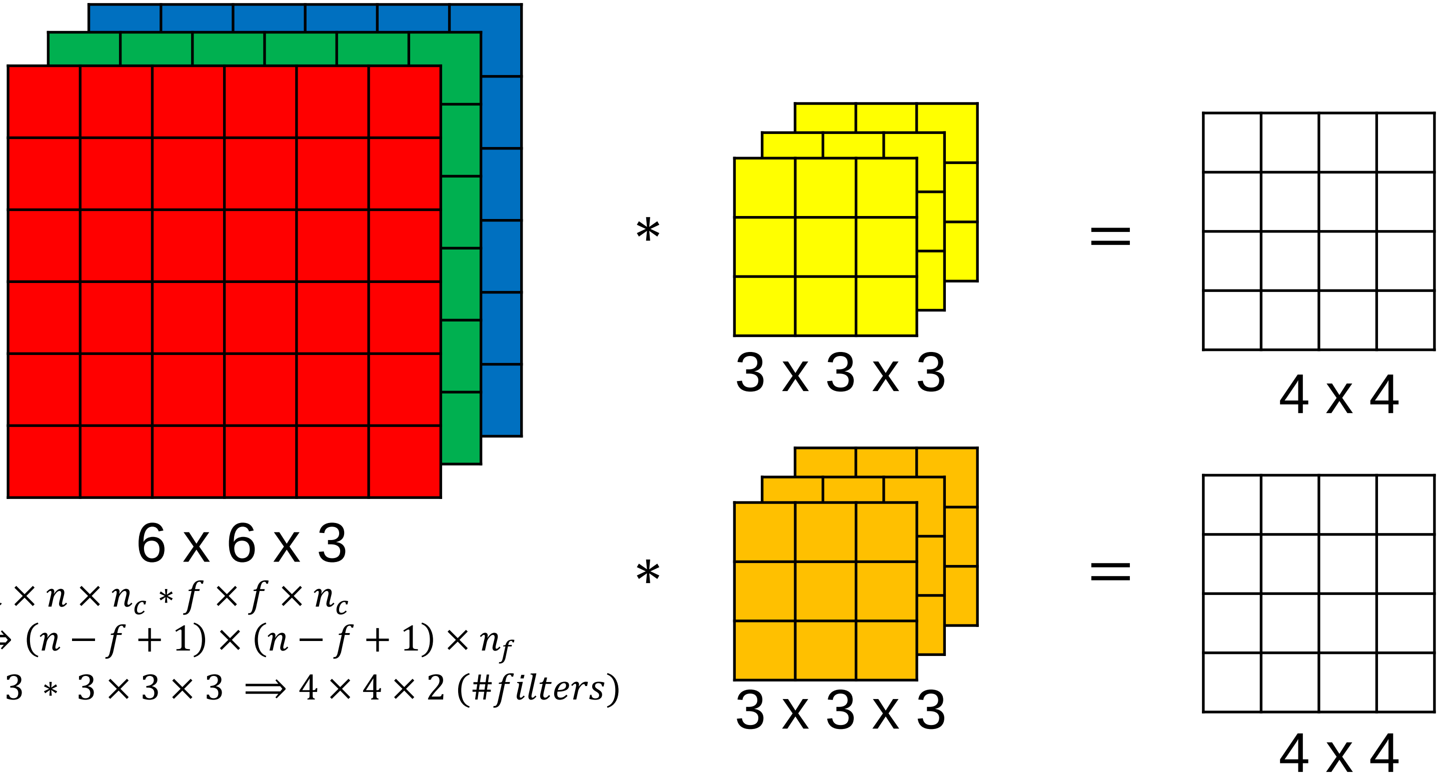
$=$

91	100	83
69	91	127
44	71	74

$$n \times n * f \times f \ (padding\ p, stride\ s) \Rightarrow \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

$$(7+0-3)/2+1 = 4/2+1 = 3$$

Multiple Filter (Convs)

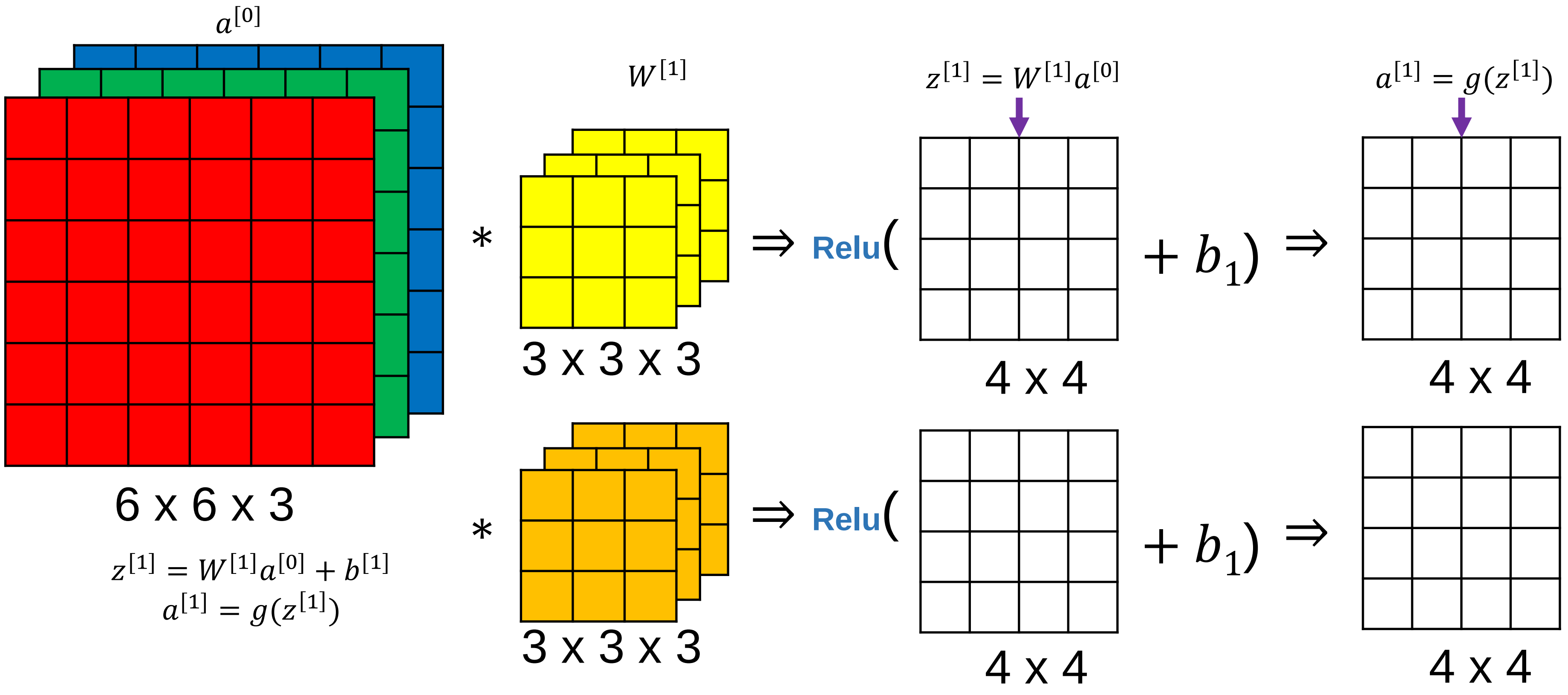


$n \times n \times n_c * f \times f \times n_c$

$\Rightarrow (n - f + 1) \times (n - f + 1) \times n_f$

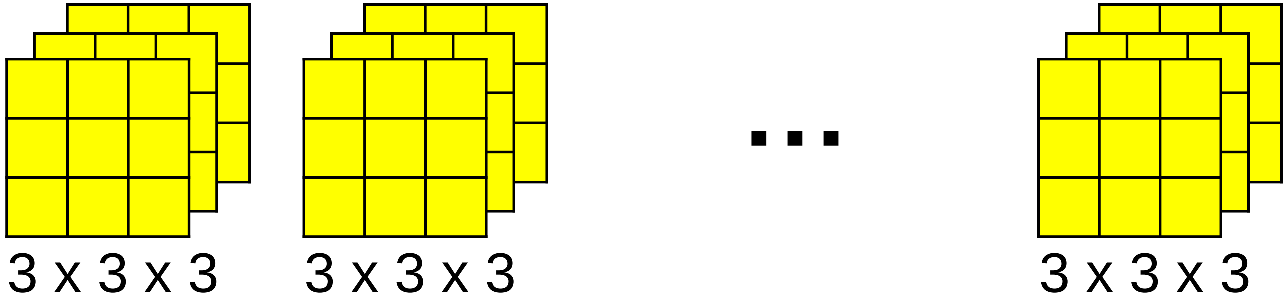
$6 \times 6 \times 3 \; * \; 3 \times 3 \times 3 \; \Longrightarrow 4 \times 4 \times 2 \; (\#filters)$

Multiple Filter (Convs)



No. Parameters

- If you have 10 filters that are 3 x 3 x 3 in one layer of a neural network, how many parameters does that layer have?



- (3 x 3 x 3 + 1) x 10 = 280 parameters
- No matter how large the input image is, the number of parameters is fixed to 280 for 10 filters of size 3 x 3 x 3.

3.4 Convolutional layer

Summary

If layer 1 is a convolution layer:

Activations: $\boldsymbol{a}^{[l]} \Rightarrow \boldsymbol{n}_H^{[l]} \times \boldsymbol{n}_W^{[l]} \times \boldsymbol{n}_C^{[l]}$

- $f^{[l]} = filter\ size$

  Weights: $\boldsymbol{f}^{[l]} \times \boldsymbol{f}^{[l]} \times \boldsymbol{n}_C^{[l-1]} \times \boldsymbol{n}_C^{[l]}$

- $p^{[l]} = padding$

  Bias: $\boldsymbol{n}_C^{[l]}$

- $s^{[l]} = stride$
- $n_C^{[l]} = number\ of\ filters$
- Each filter is: $f^{[l]} \times f^{[l]} \times n_C^{[l-1]}$
- Input Size: $n_H^{[l-1]} \times n_W^{[l-1]} \times n_C^{[l-1]}$
- Output Size: $n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$
- $n_H^{[l]} = \left\lfloor \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$
- $n_W^{[l]} = \left\lfloor \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$

Summary

If layer 1 is a convolution layer:

Activations: $\boldsymbol{a}^{[l]} \Rightarrow \boldsymbol{n}_H^{[l]} \times \boldsymbol{n}_W^{[l]} \times \boldsymbol{n}_C^{[l]}$

- $f^{[l]} = filter\ size$

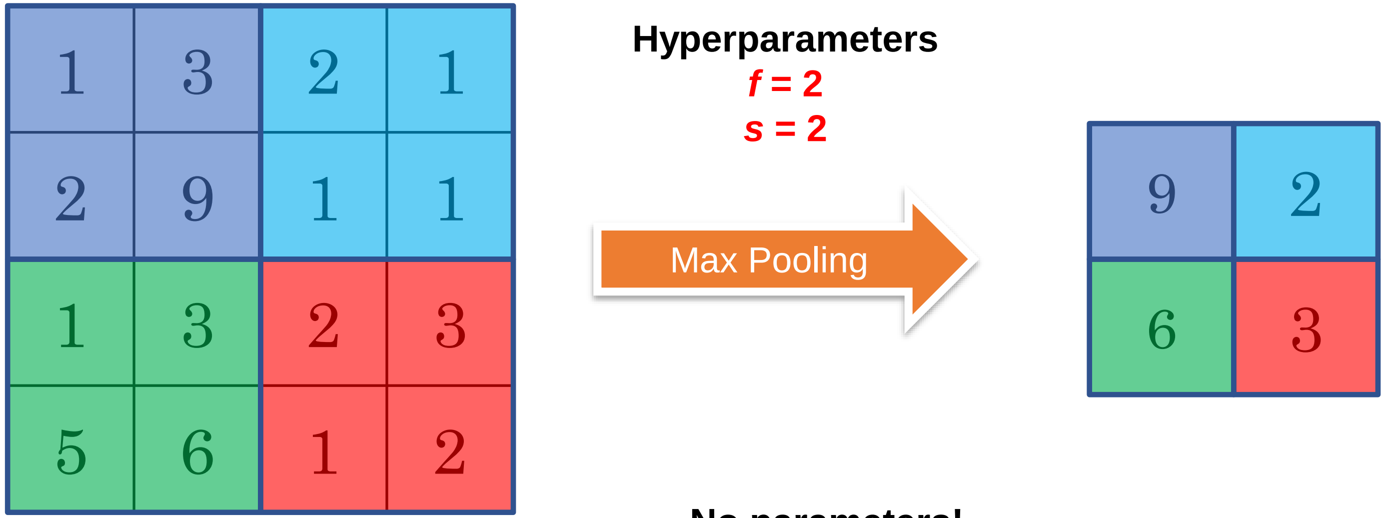
Weights: $\boldsymbol{f}^{[l]} \times \boldsymbol{f}^{[l]} \times \boldsymbol{n}_C^{[l-1]} \times \boldsymbol{n}_C^{[l]}$

- $p^{[l]} = padding$

Bias: $\boldsymbol{n}_C^{[l]}$

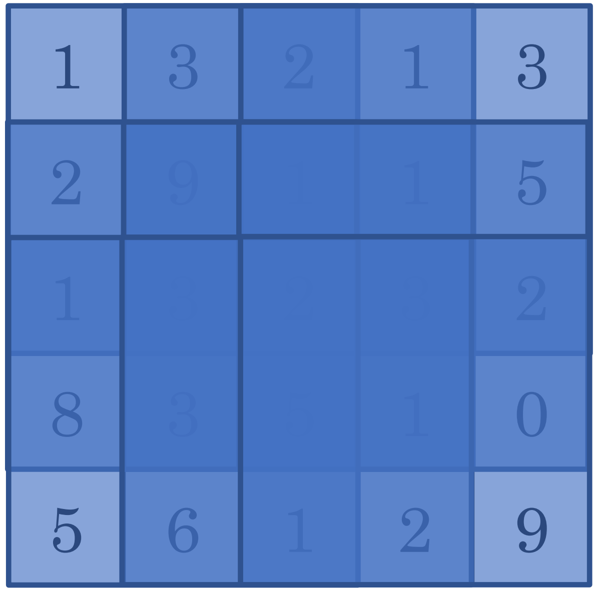
- $s^{[l]} = stride$
- $n_C^{[l]} = number\ of\ filters$
- Each filter is: $f^{[l]} \times f^{[l]} \times n_C^{[l-1]}$
- Input Size: $n_H^{[l-1]} \times n_W^{[l-1]} \times n_C^{[l-1]}$
- Output Size: $n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$
- $n_H^{[l]} = \left\lfloor \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$
- $n_W^{[l]} = \left\lfloor \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$

3.4 Max Pooling



No parameters!

3.4 Max Pooling



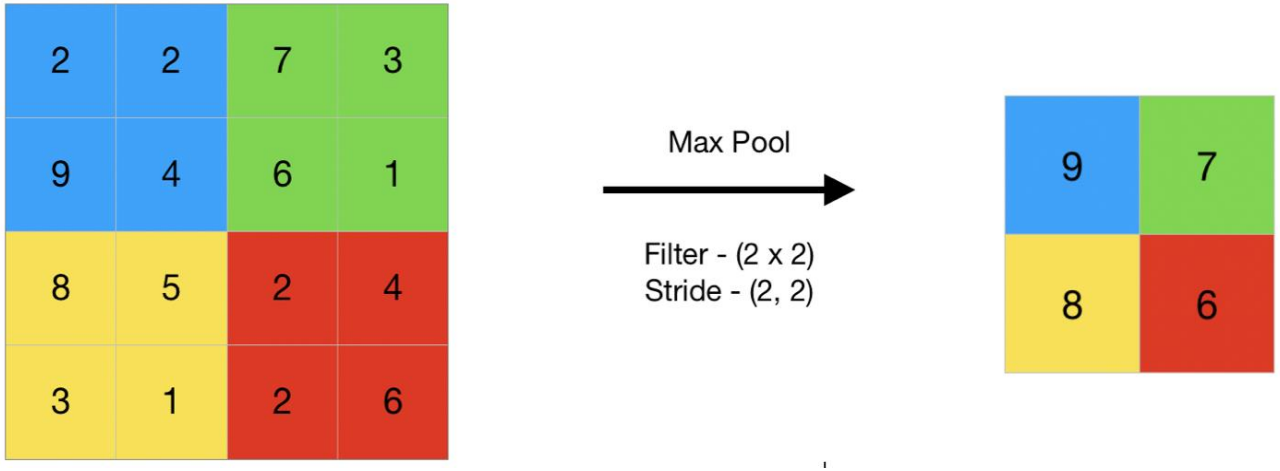
5 x 5

Hyperparameters
***f* = 3**
***s* = 1**

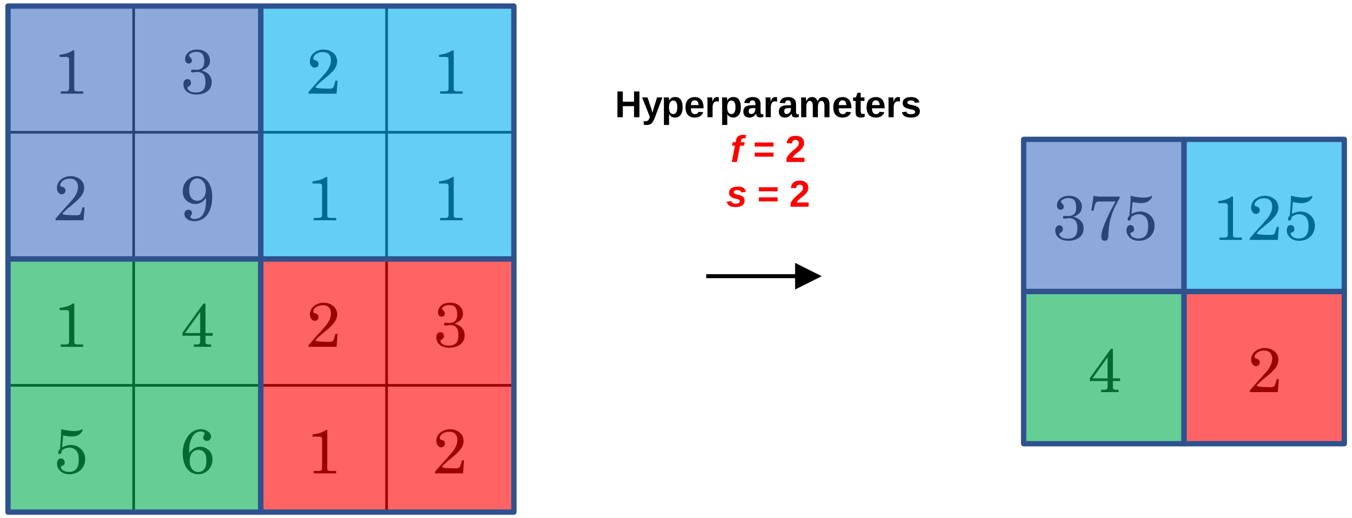
9	9	5
9	9	5
8	6	9

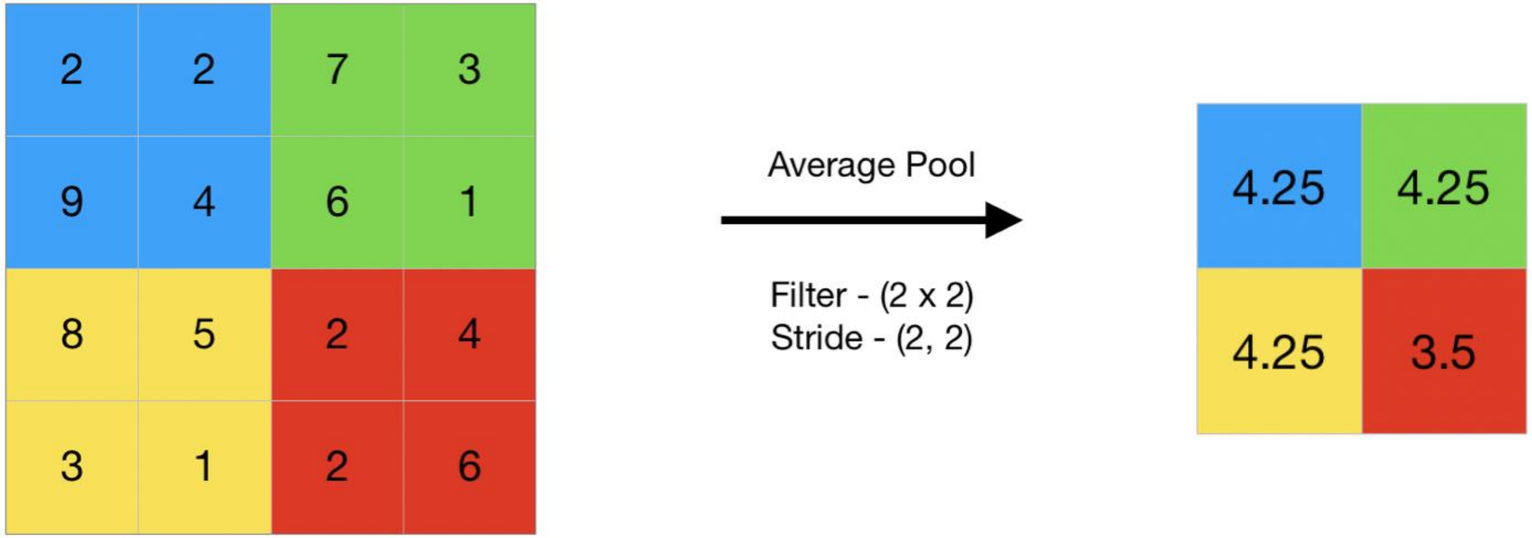
3 x 3

Note: For multiple channels, the above max pooling is done for each channel

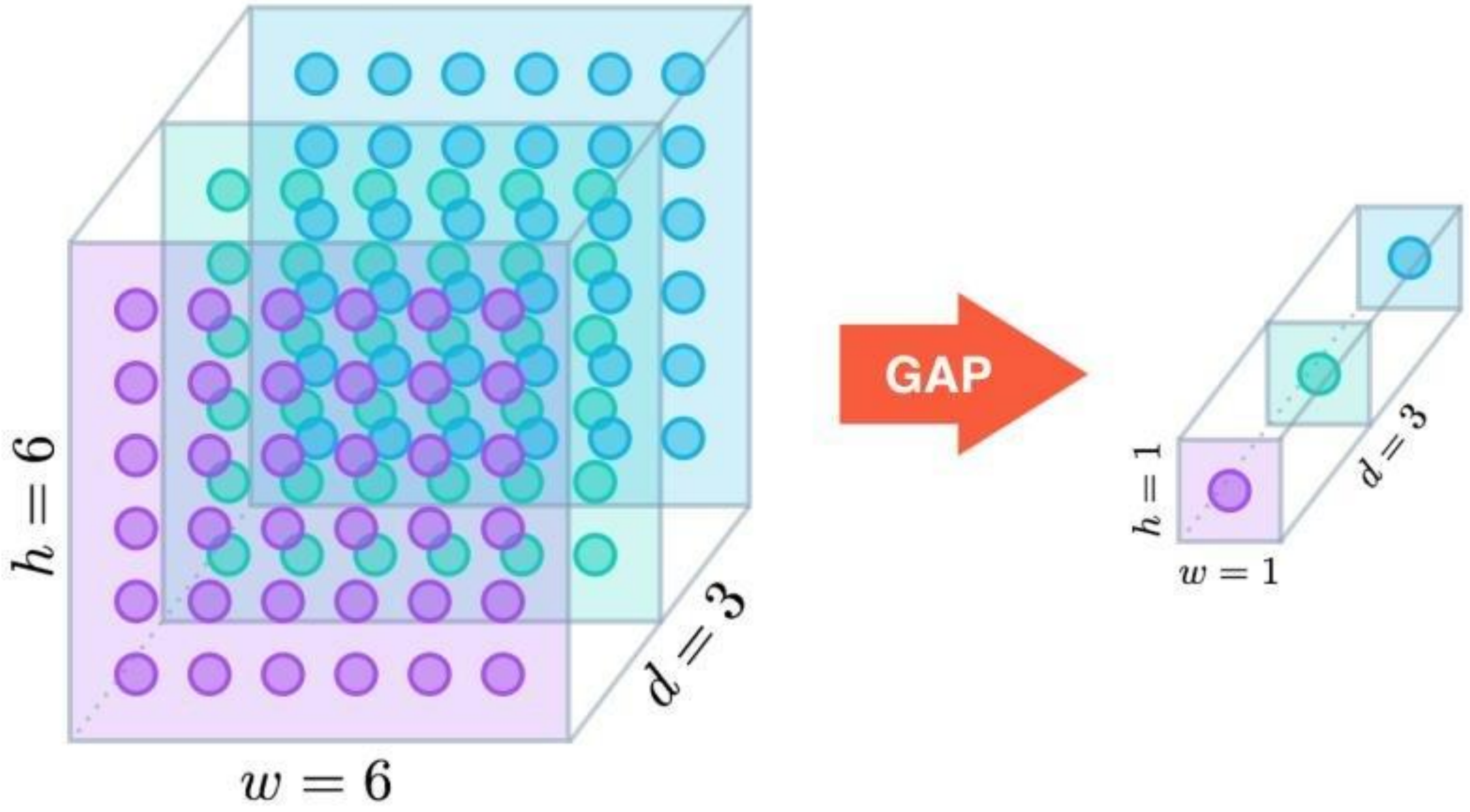


3.4 Average Pooling





3.4 Global Pooling



3.4 Pooling

Hyperparameters

- f: filter size
- s: stride
- Max or average pooling
- Usually, p = 0, no padding

$$n_H \times n_W \times n_C \rightarrow \left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times n_C$$