Applied Deep Learning

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

www.bayes.city.ac.uk

Learning objectives of today

Goals: Understand the neural network foundations of computer vision, one of the most promising deep learning applications:

- Convolutional layers: hidden layers purpose-built for image analysis
- Pooling layers: keeping complexity in check

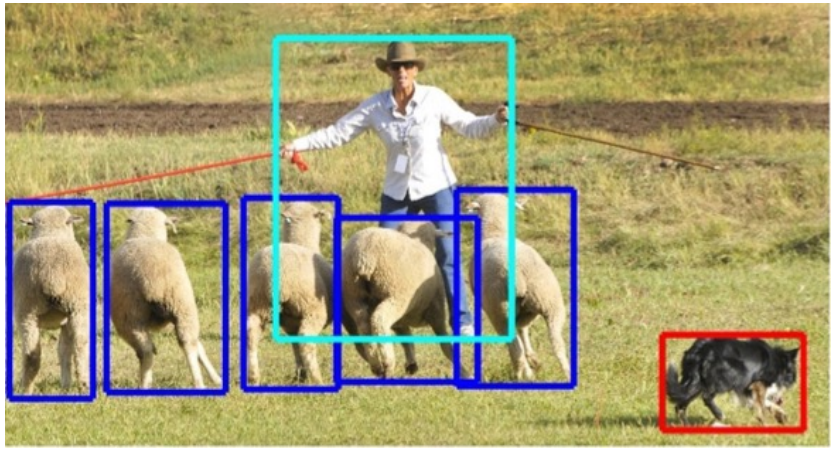
How will we do this?

- We discuss the biological motivations behind convolutional layers
- We then look at the convolution operations and how convolutional layers simplify computer vision
- We will see in class how to add convolutional layers to our neural networks

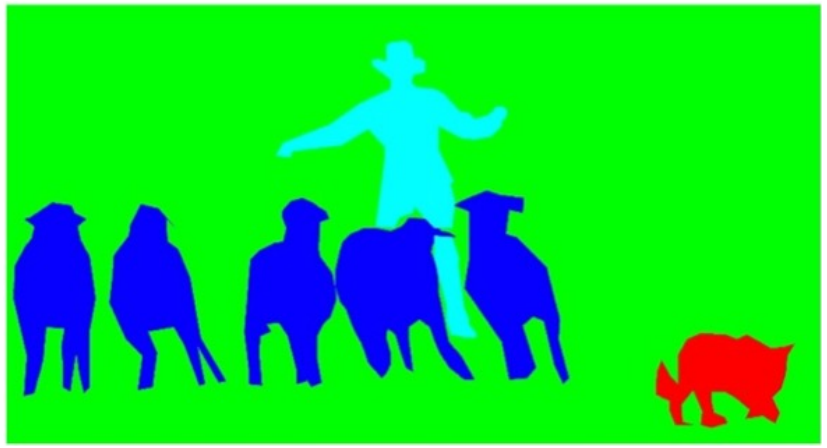
Typical computer vision problems

Image classification

Object detection



Semantic segmentation



Neural style transfer

Source: Lin, reiinakano.com

Computer vision in business processes

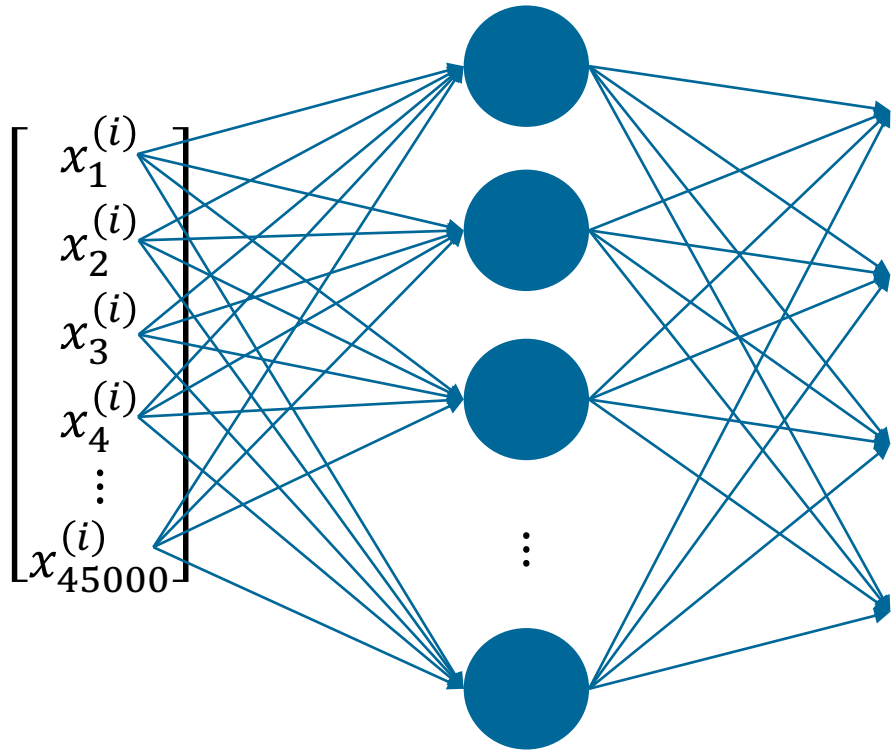
- Manufacturing:
  - Robotic control
  - Quality management
  - Equipment surveillance, preventive maintenance
  - Productivity analysis, ergonomics
- Services
  - Process analysis
  - Security systems
- Agriculture
  - Plant monitoring
  - Animal monitoring
- Transportation
  - Self-driving cars
  - Traffic/logistics flow analysis
- Healthcare
  - Medical image analysis
  - Movement analysis

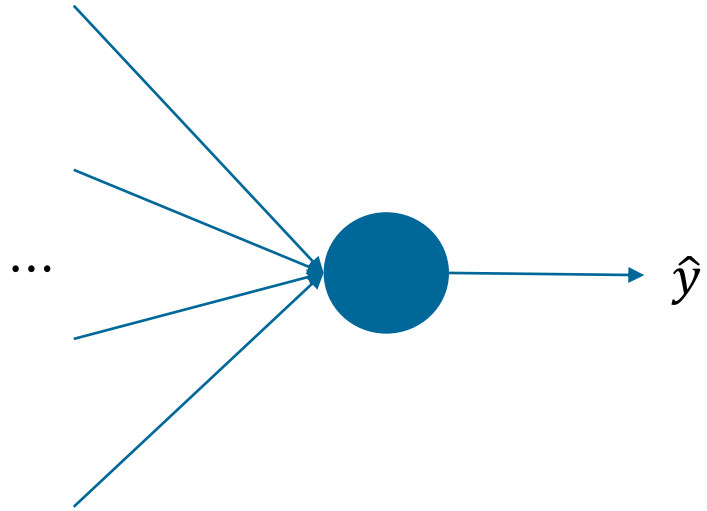
Challenges of dense layers in image analysis

Challenges to deep learning using large images

$y = 1$, if it's a cat
$y = 0$, if it's a BA student

150x100x3

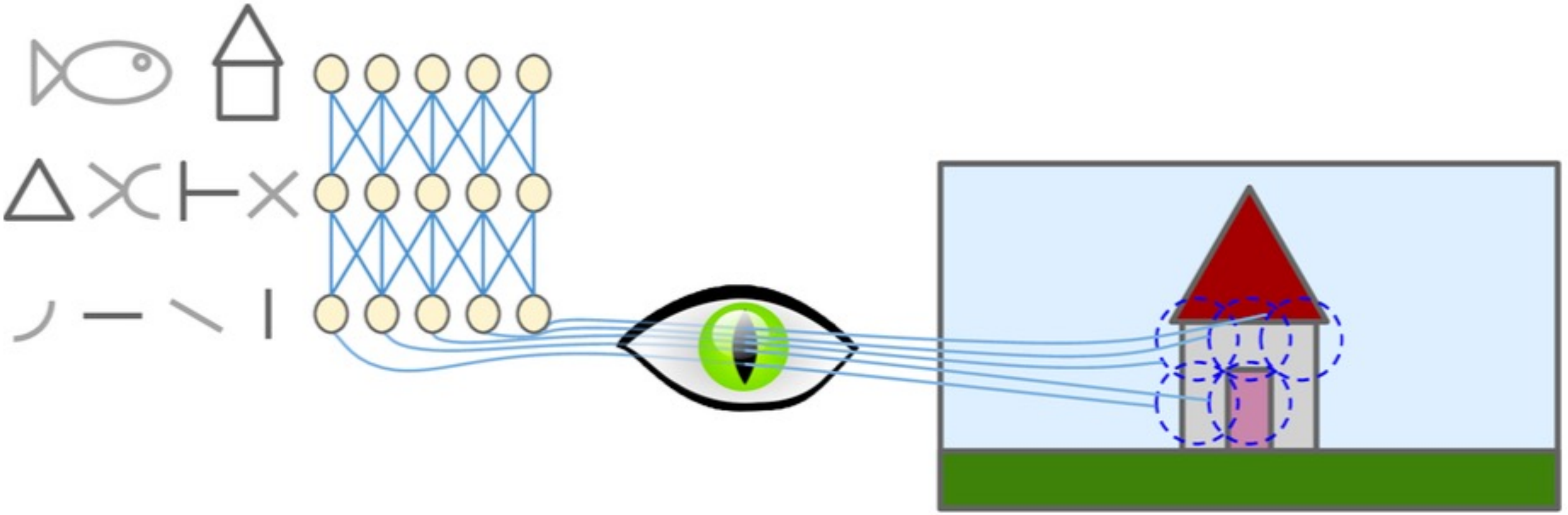




Say we have 100 neurons at the first layer.
Then we need 100*(45000+1)=4.5 mio parameters, just for the first layer!

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

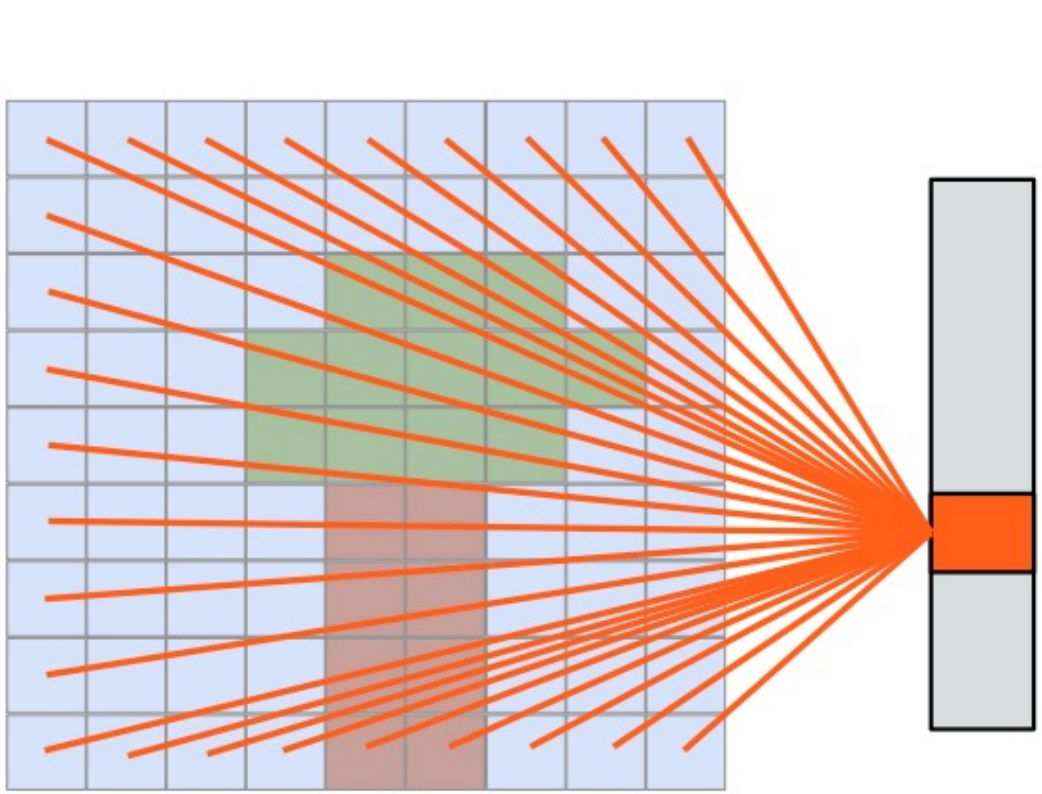
Motivation – the visual cortex

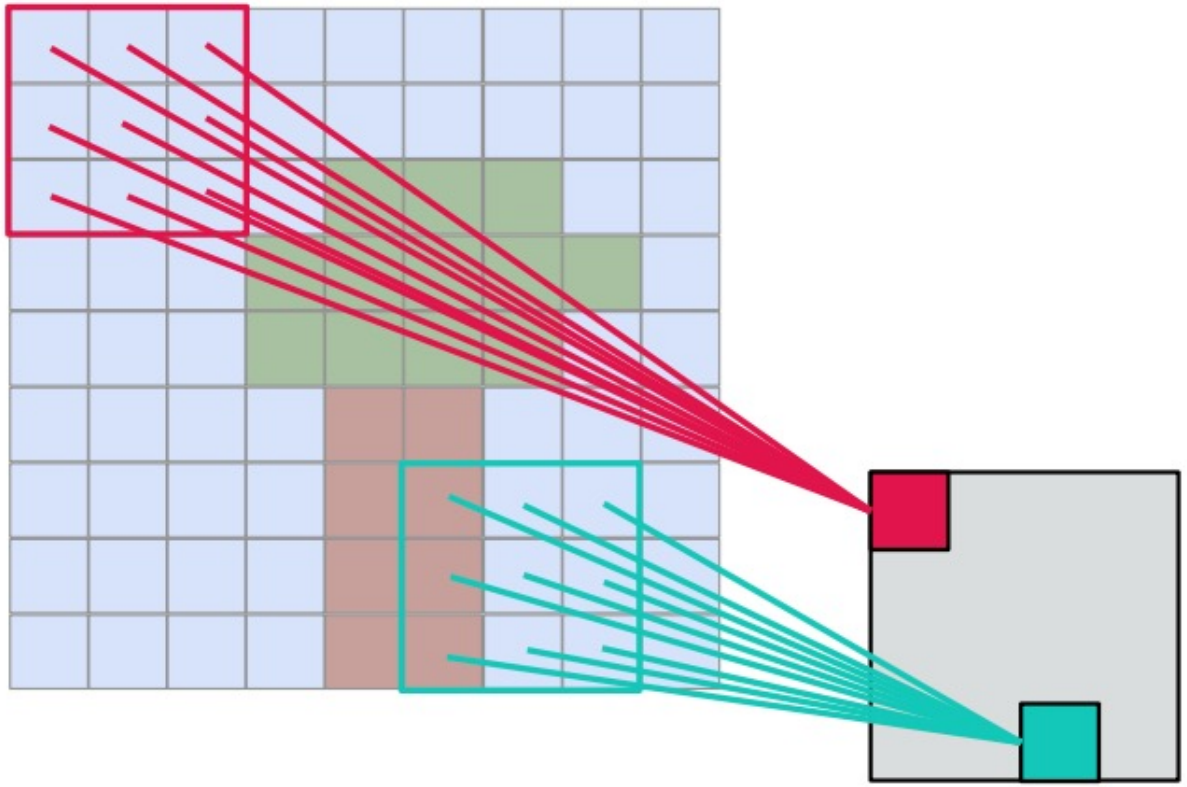


Source: Géron

Image filters

From fully connected to locally connected





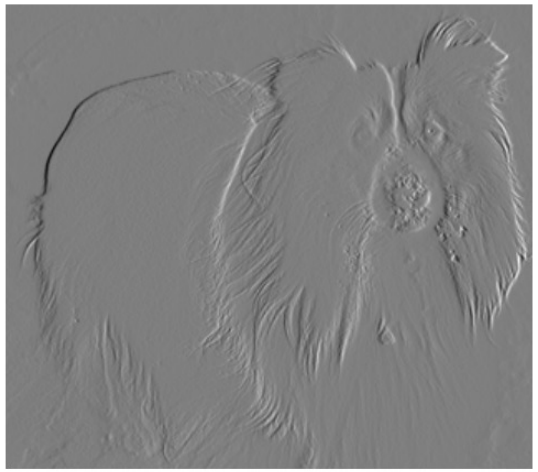
Source: Dieleman

Edge detection by convolution

Input

1	-1

Kernel



Output

Source: Goodfellow

The convolution operator

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

1	2
2	1

=

20		

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

1	2
2	1

=

20	12	

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

1	2
2	1

=

20	12	19

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

1	2
2	1

=

20	12	19
22		

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

1	2
2	1

=

20	12	19
22	21	

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

1	2
2	1

=

20	12	19
22	21	22
22	15	16

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Vertical edge detection

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			

Vertical edge detection

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		

Vertical edge detection

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	

Vertical edge detection

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

Vertical edge detection

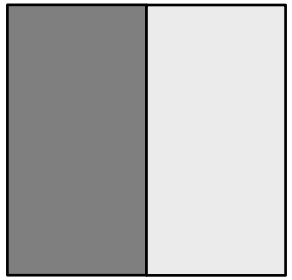
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

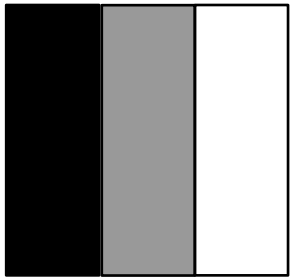
*

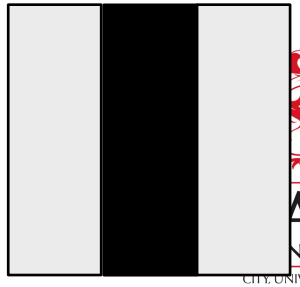
1	0	-1
1	0	-1
1	0	-1

=

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

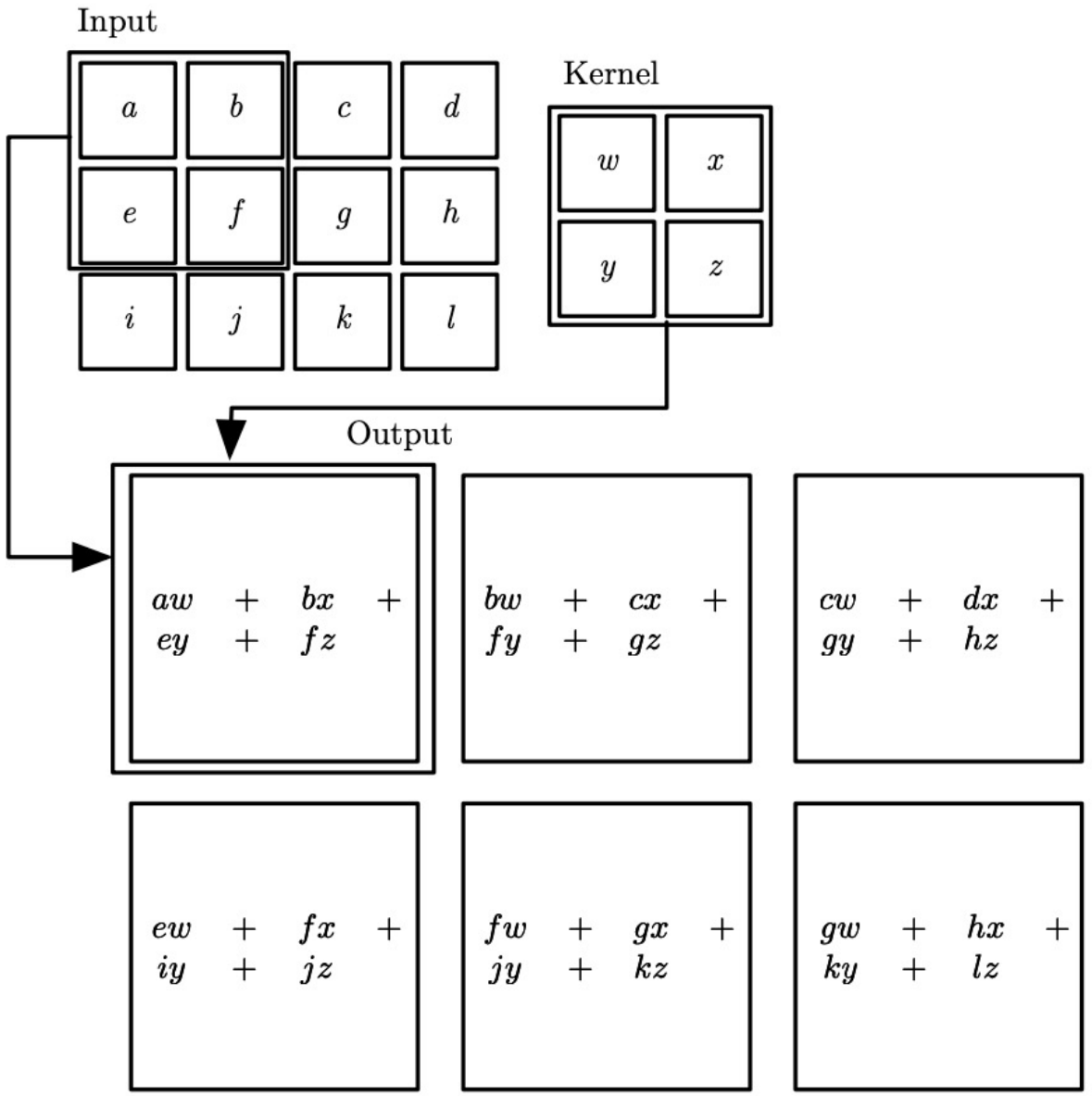






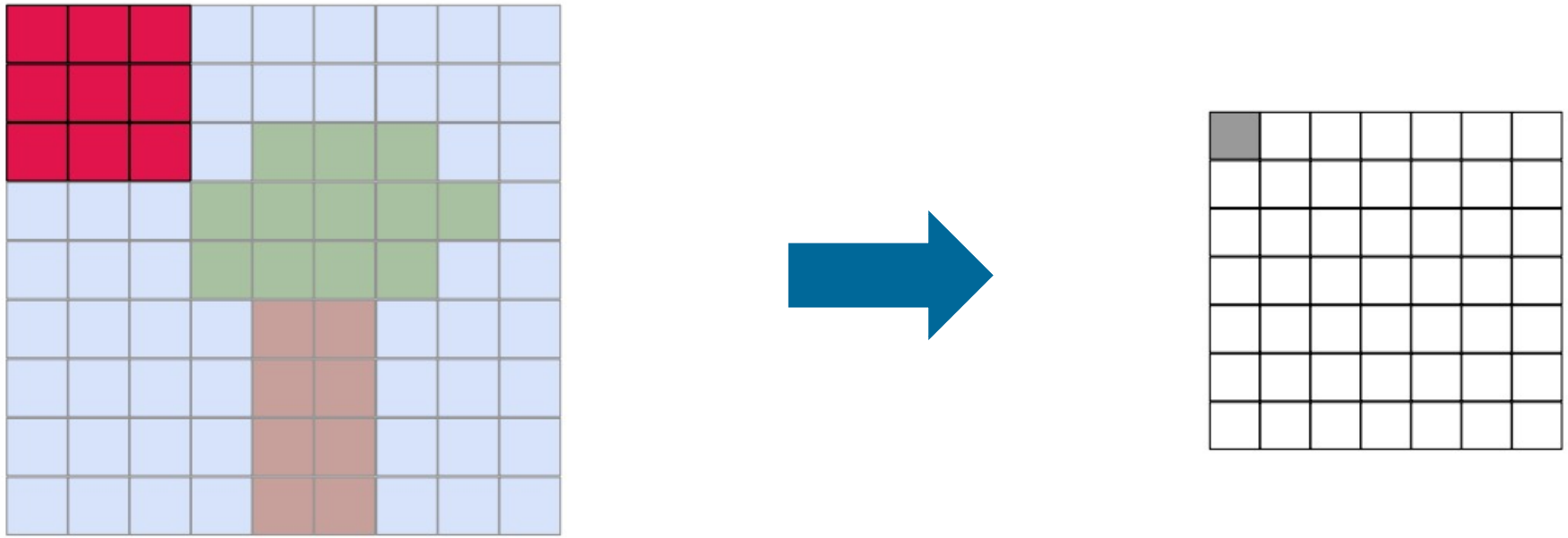
Two-dimensional convolutional layers

Learning the right filter – a convolutional layer



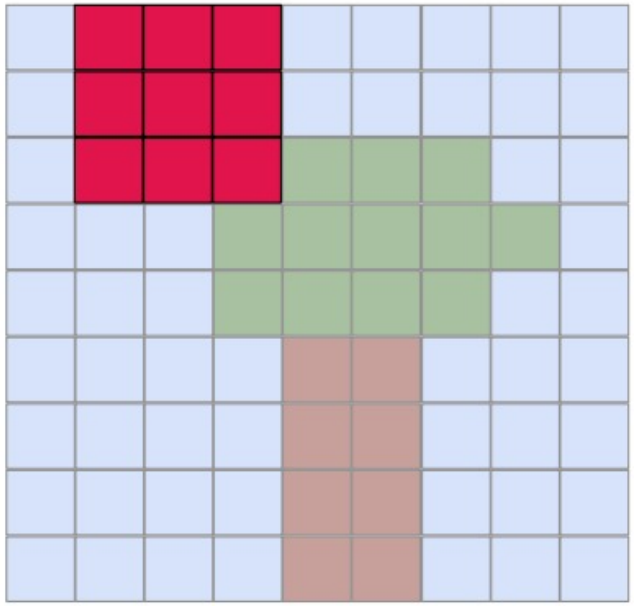
Source: Goodfellow

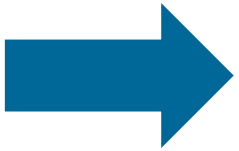
Convolution in practice

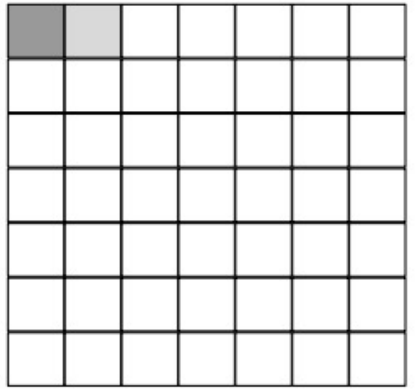


Source: Dieleman

Convolution in practice

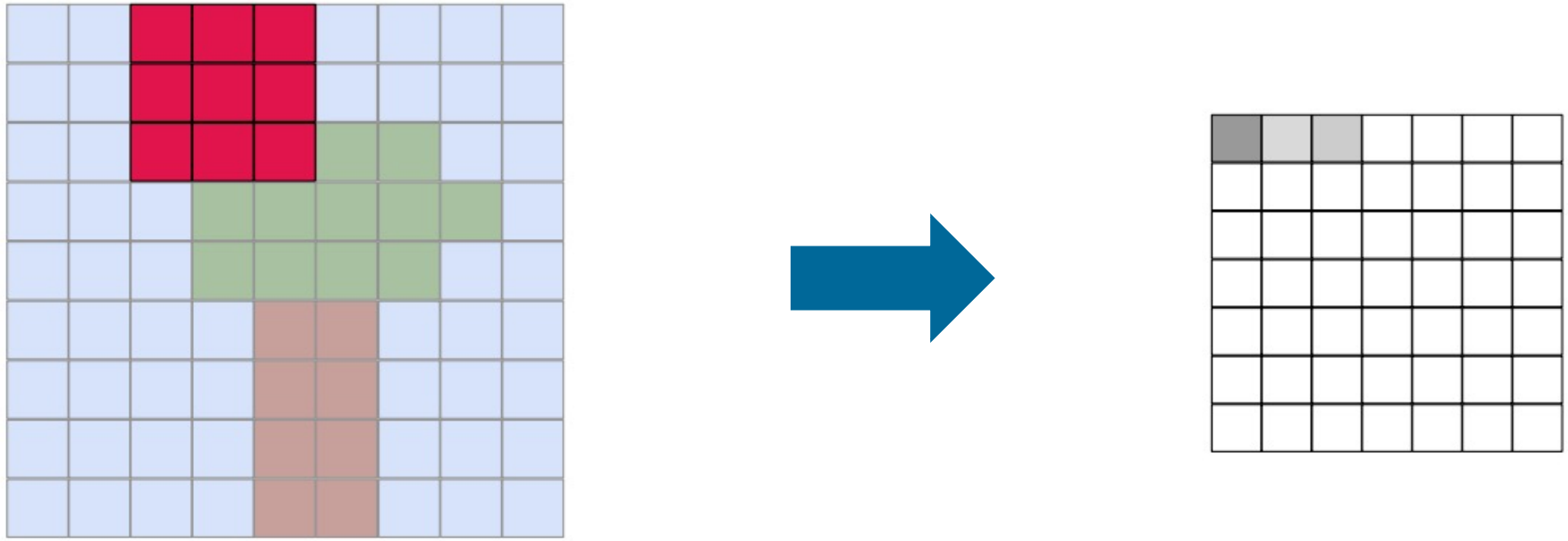






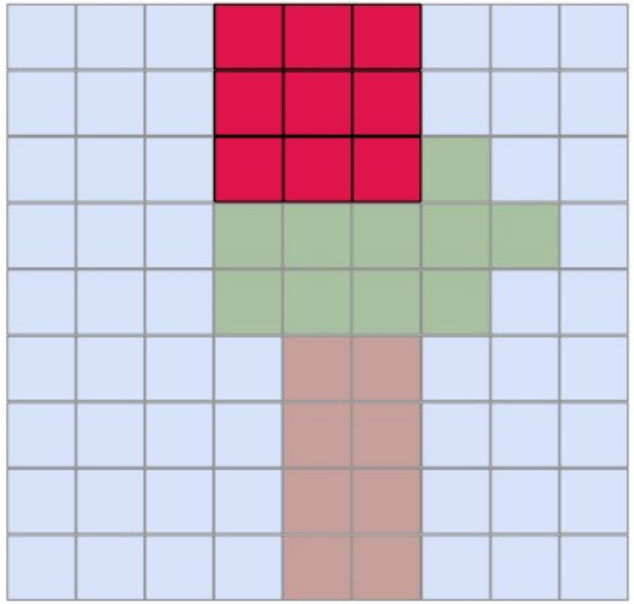
Source: Dieleman

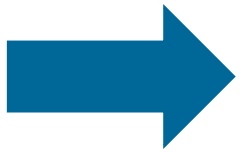
Convolution in practice

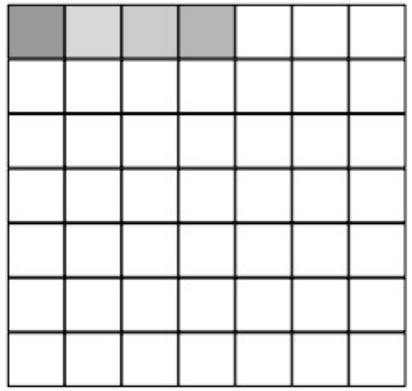


Source: Dieleman

Convolution in practice

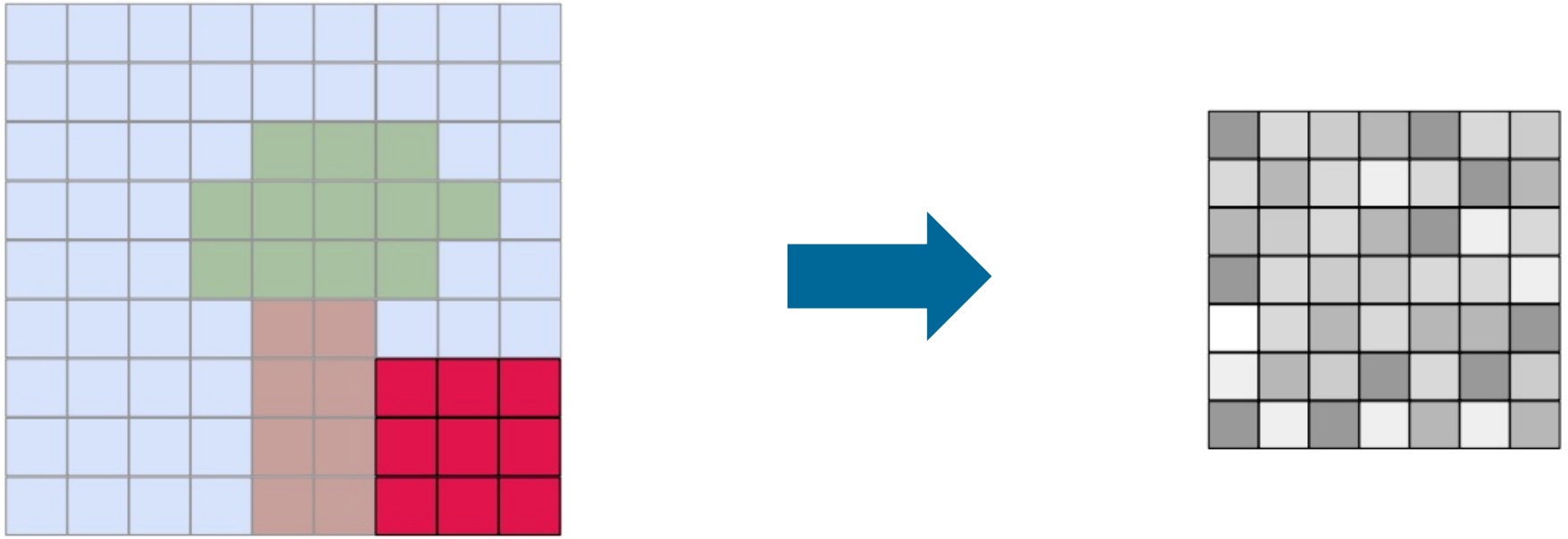






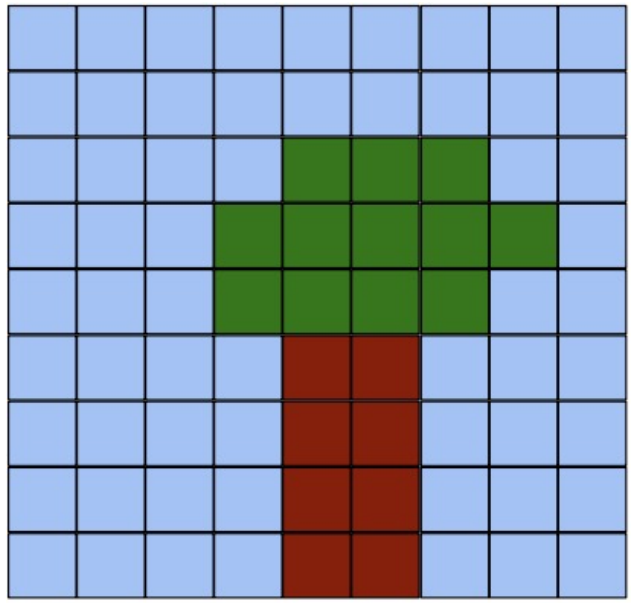
Source: Dieleman

Convolution in practice



Source: Dieleman

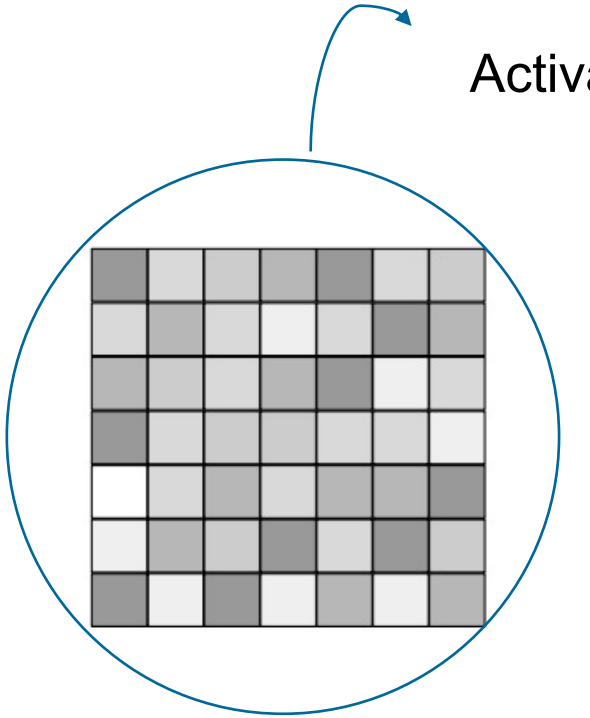
Some more details



$*$

$w_{1,1}$	$w_{1,2}$	$w_{1,3}$
$w_{2,1}$	$w_{2,2}$	$w_{2,3}$
$w_{3,1}$	$w_{3,2}$	$w_{3,3}$

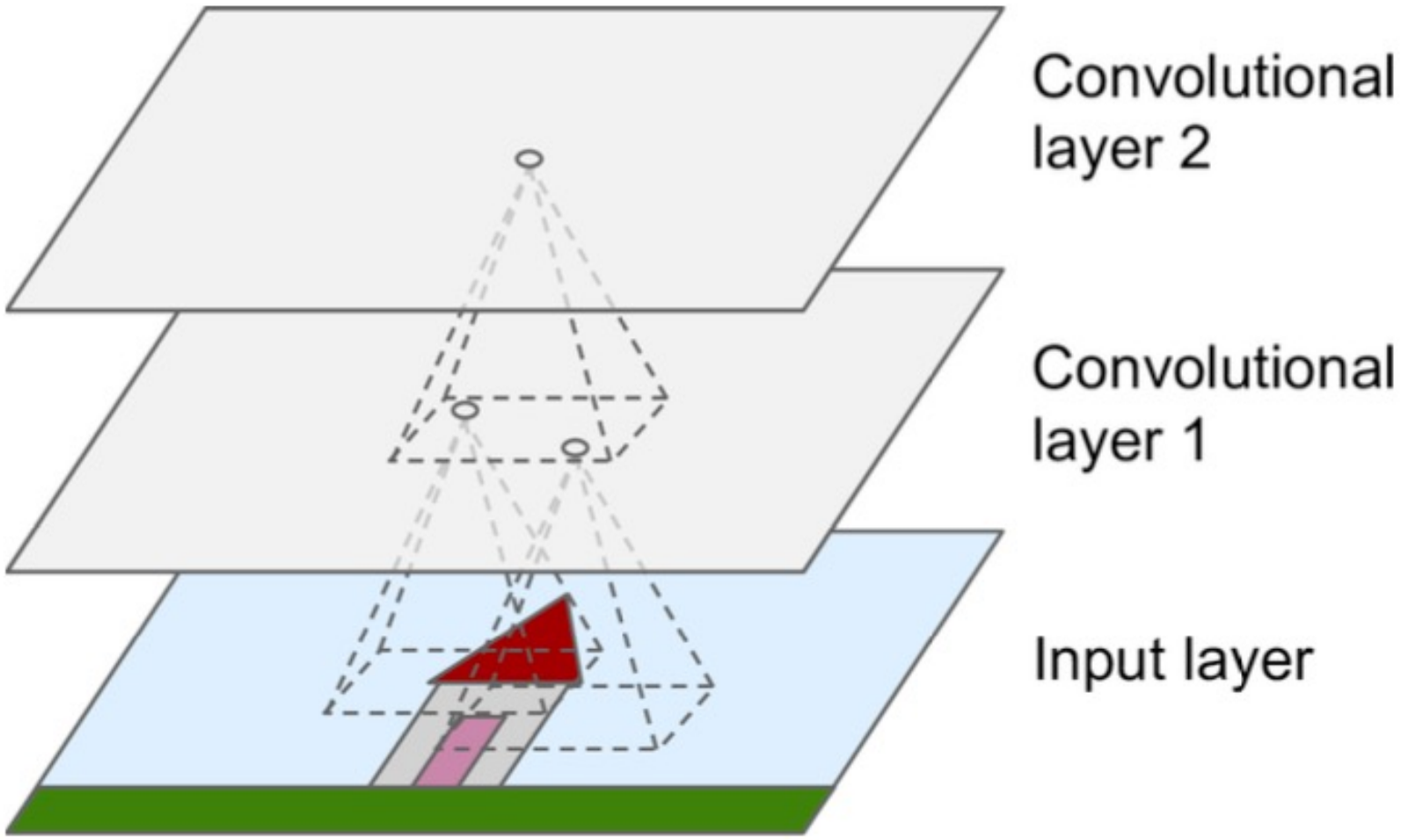
$+ \quad b \quad =$



Activation function

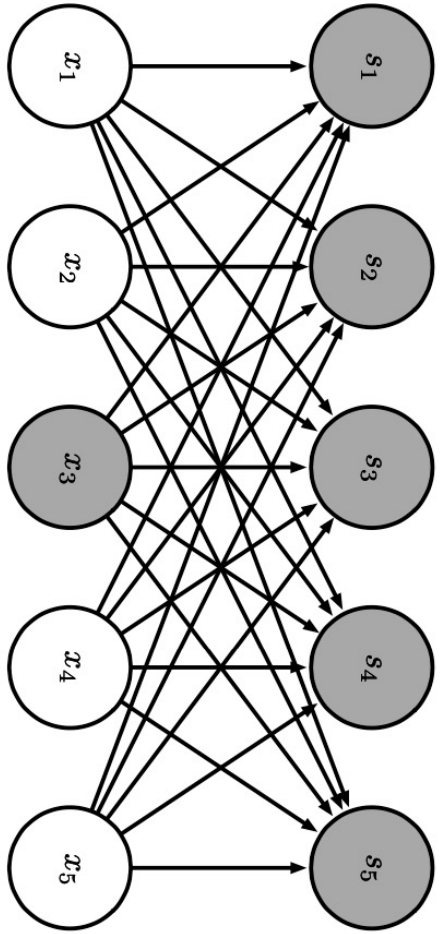
Source: Dieleman

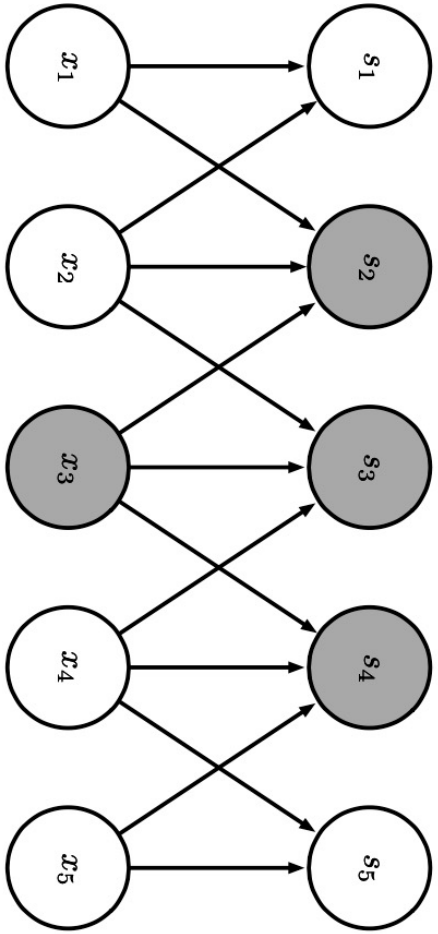
Stacking convolutional layers



Source: Géron

Less parameters required





Source: Goodfellow

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

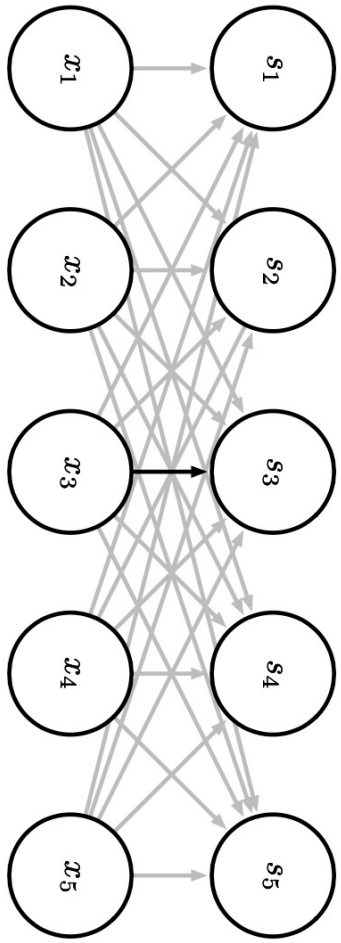
Efficiency of convolution

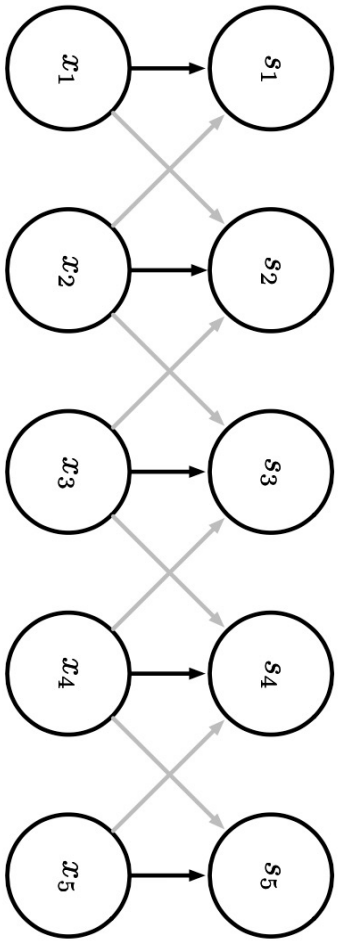
- As before: images with 150x100 (but only one channel, e.g., greyscale)
- Hidden layer with 100 neurons versus a convolutional layer that leads to a 10x10 output

Dense layer with 100 neurons	91x141 Convolution layer
100*(150x100+1)=1.5 mio	91x141+1=0.013 mio

- Usually, we make much smaller convolutions (but many of them)

Parameters are shared

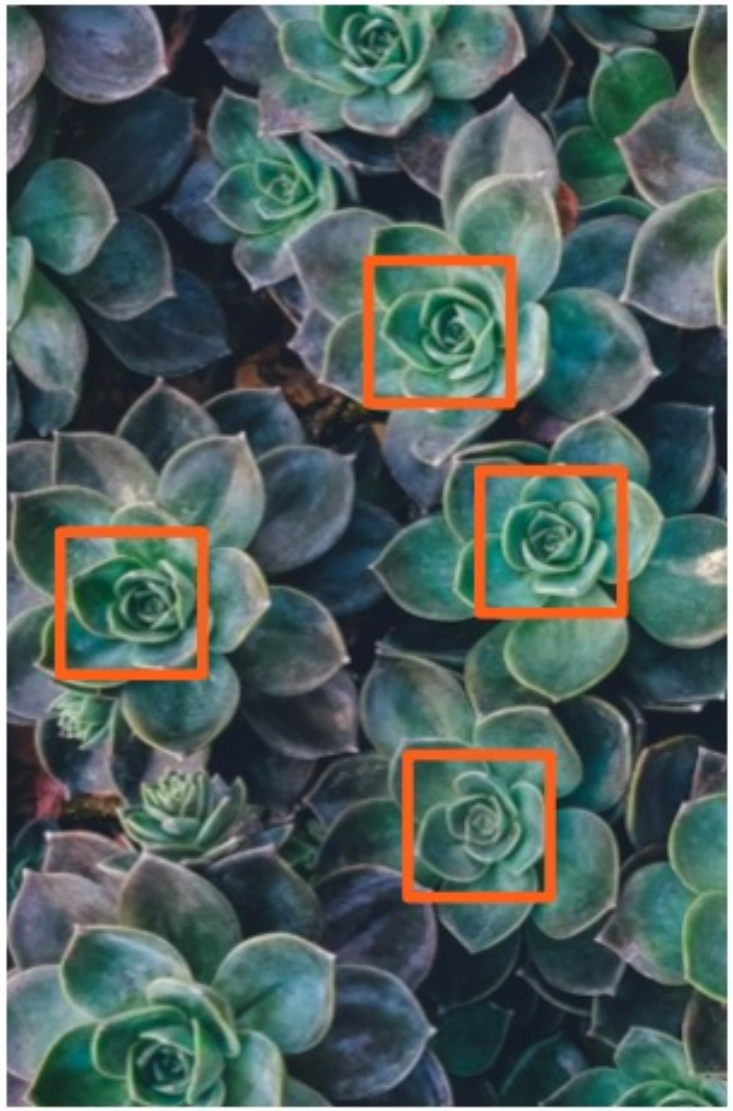




Source: Goodfellow

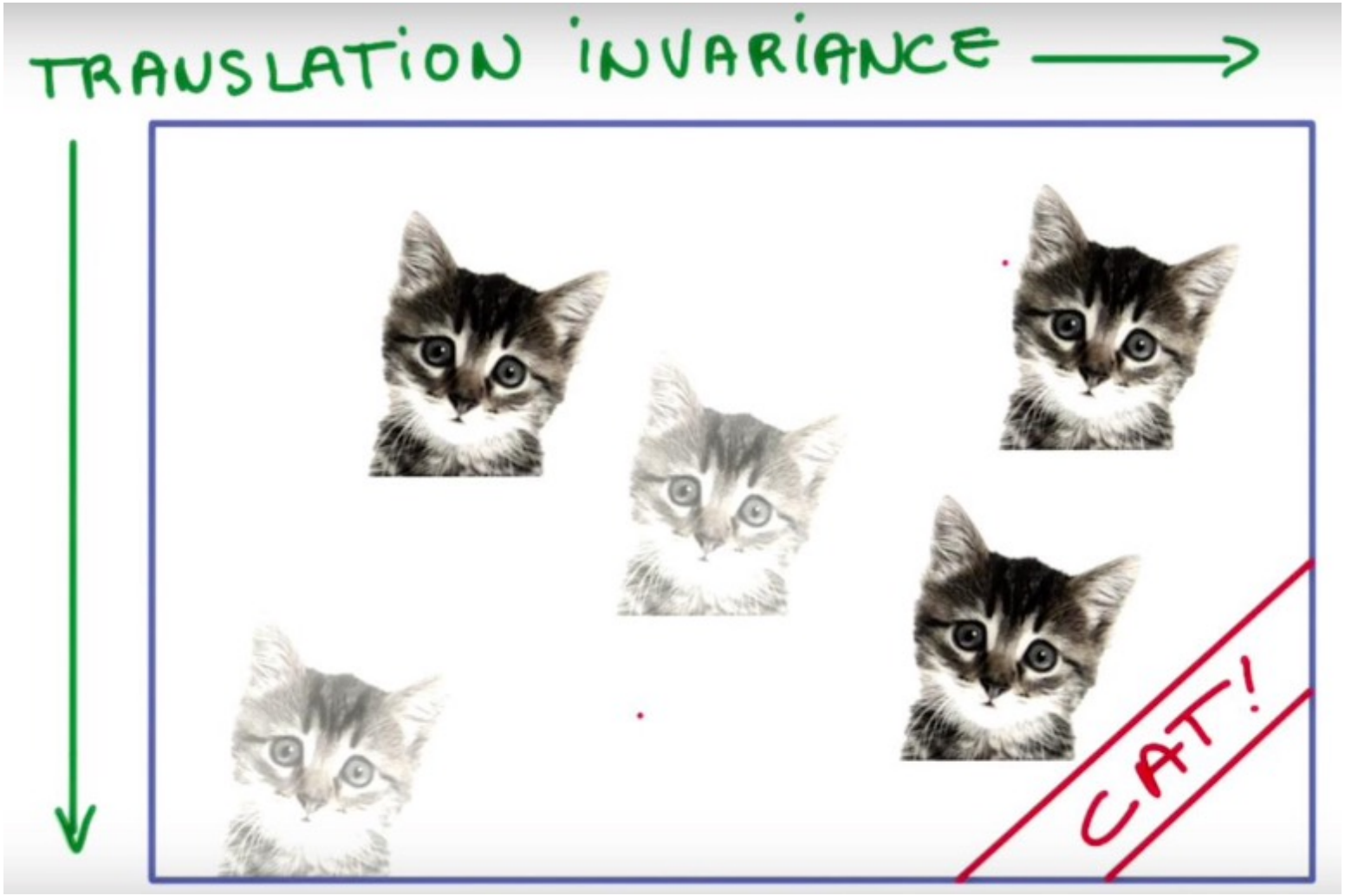
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The advantage of parameter sharing



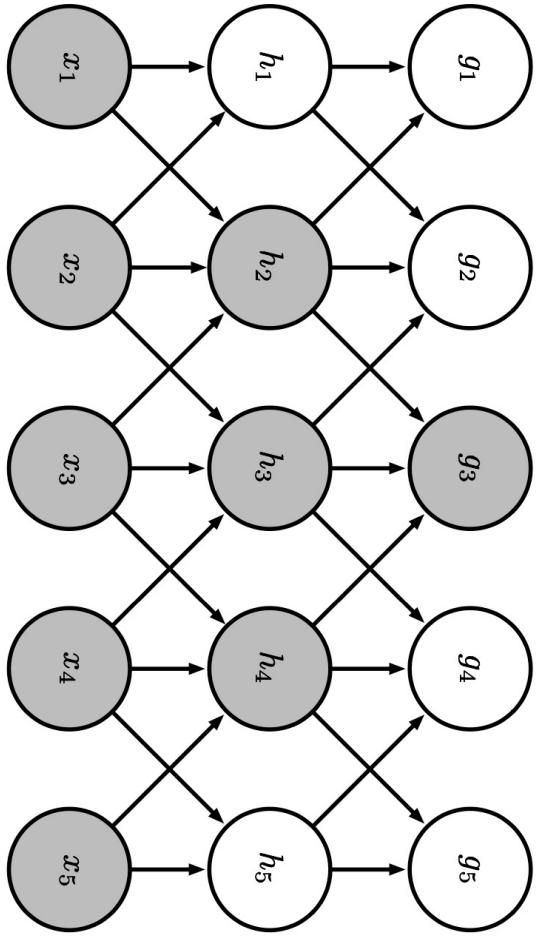
Source: Dieleman

Translation invariance



Source: Bhaskhar

Hierarchical setup



Source: Goodfellow

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The size of the output

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

4x4

$(n_H^{[0]}, n_W^{[0]})$

*

1	2
2	1

2x2

(f_H, f_W)

=

20	12	19
22	21	22
22	15	16

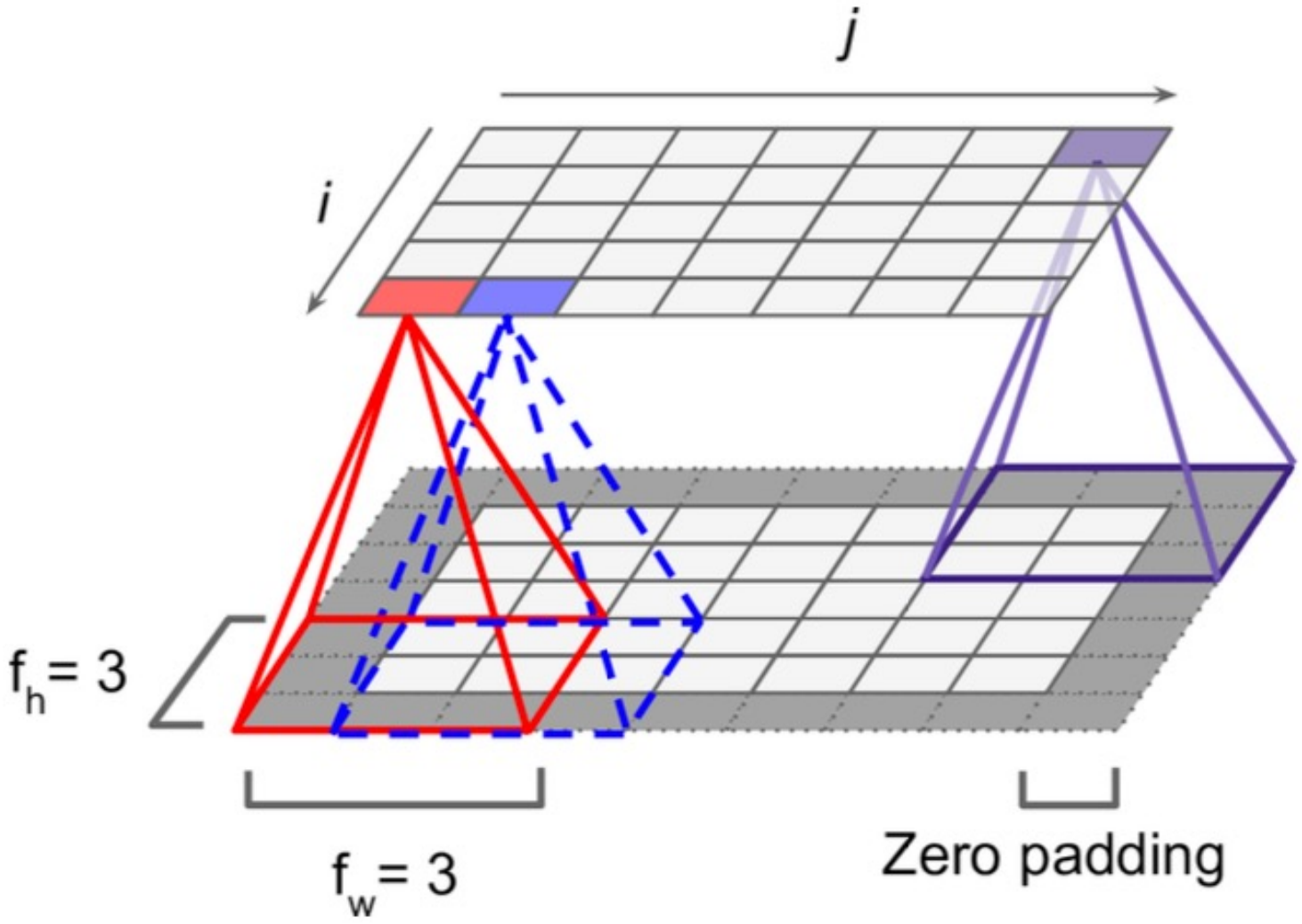
3x3

$(n_H^{[1]}, n_W^{[1]})$

$$n_H^{[1]} = n_H^{[0]} - f_H + 1$$

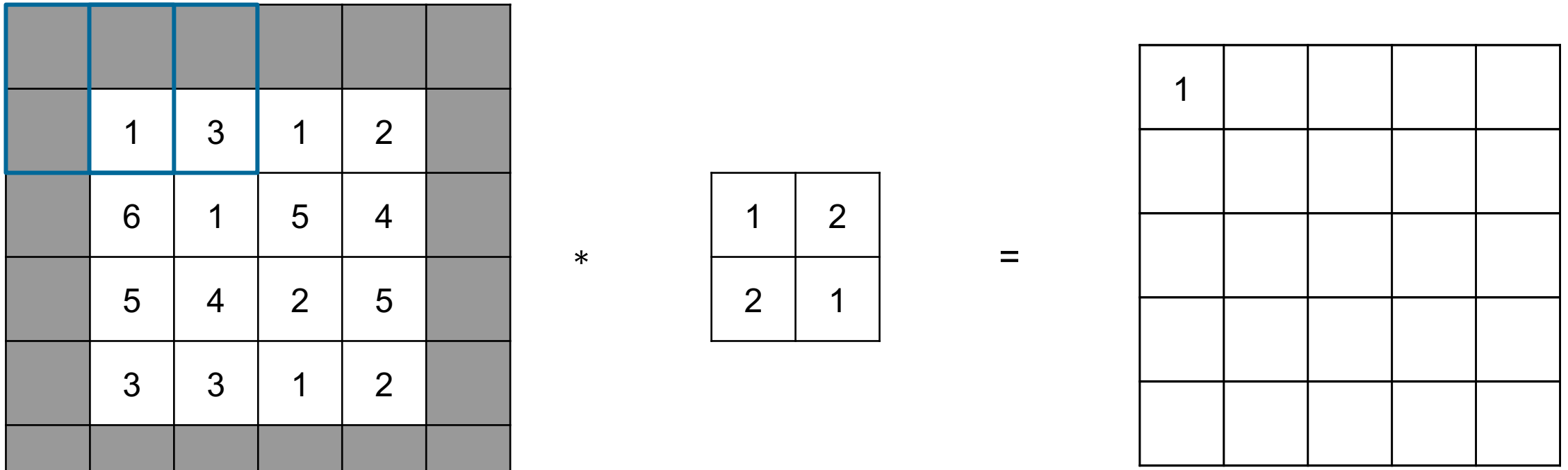
$$n_W^{[1]} = n_W^{[0]} - f_W + 1$$

Additional specification: padding

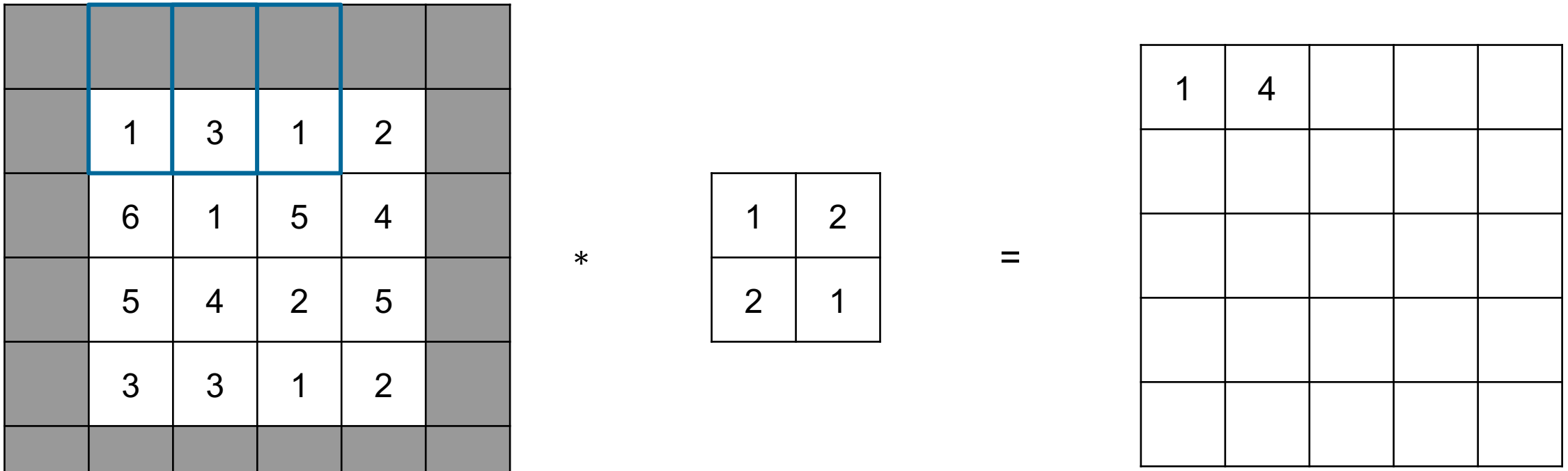


Source: Géron

The convolution operator with padding

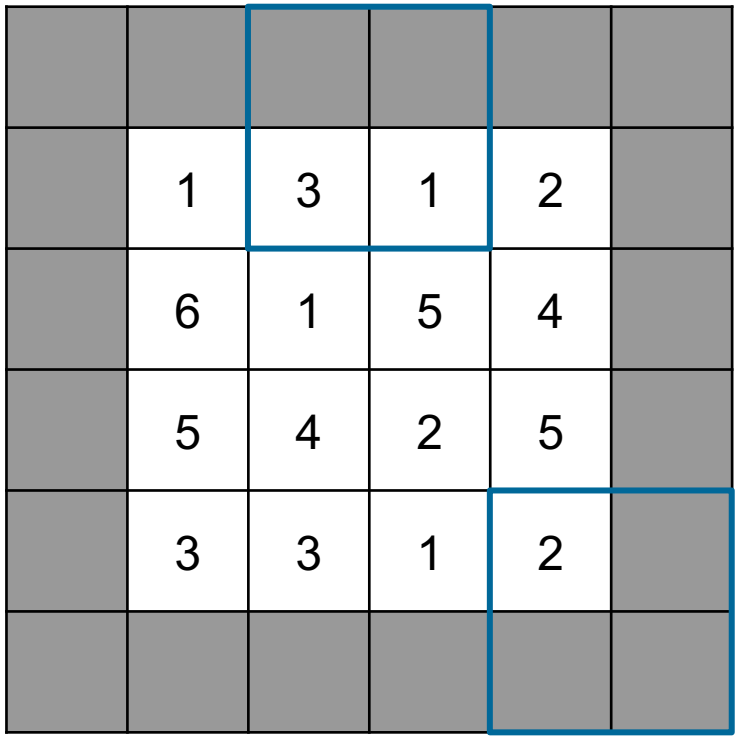


The convolution operator with padding



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator with padding



*

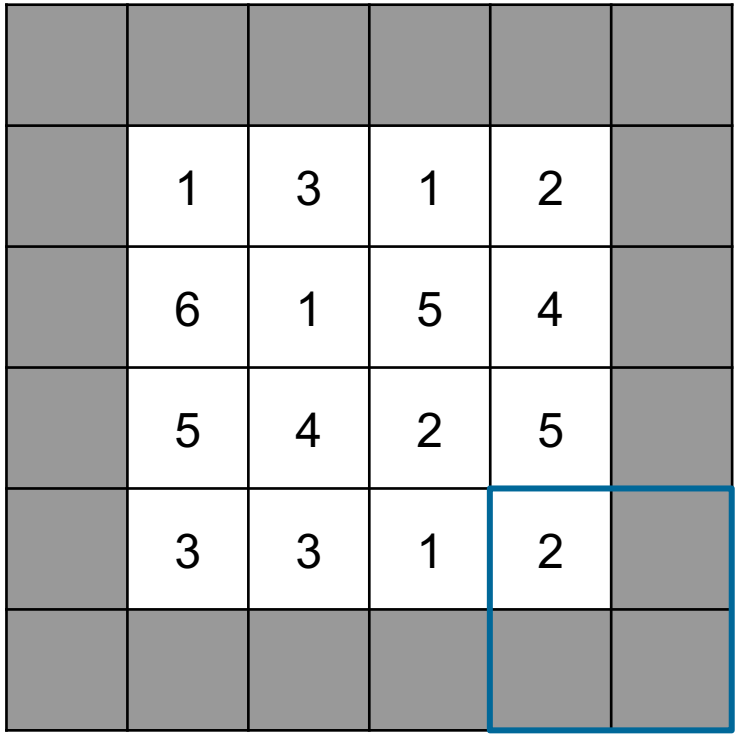
1	2
2	1

=

1	4	7		

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator with padding



4x4
$(n_H^{[0]}, n_W^{[0]})$

*

1	2
2	1

2x2 with padding 1x1
(f_H, f_W) with (p_H, p_W)

=

1	4	7	4	4
8	20	12	19	10
17	22	21	22	14
13	22	15	16	9
6	9	5	5	2

5x5
$(n_H^{[1]}, n_W^{[1]})$

$$n_H^{[1]} = n_H^{[0]} + 2p_H - f_H + 1$$

$$n_W^{[1]} = n_W^{[0]} + 2p_W - f_W + 1$$

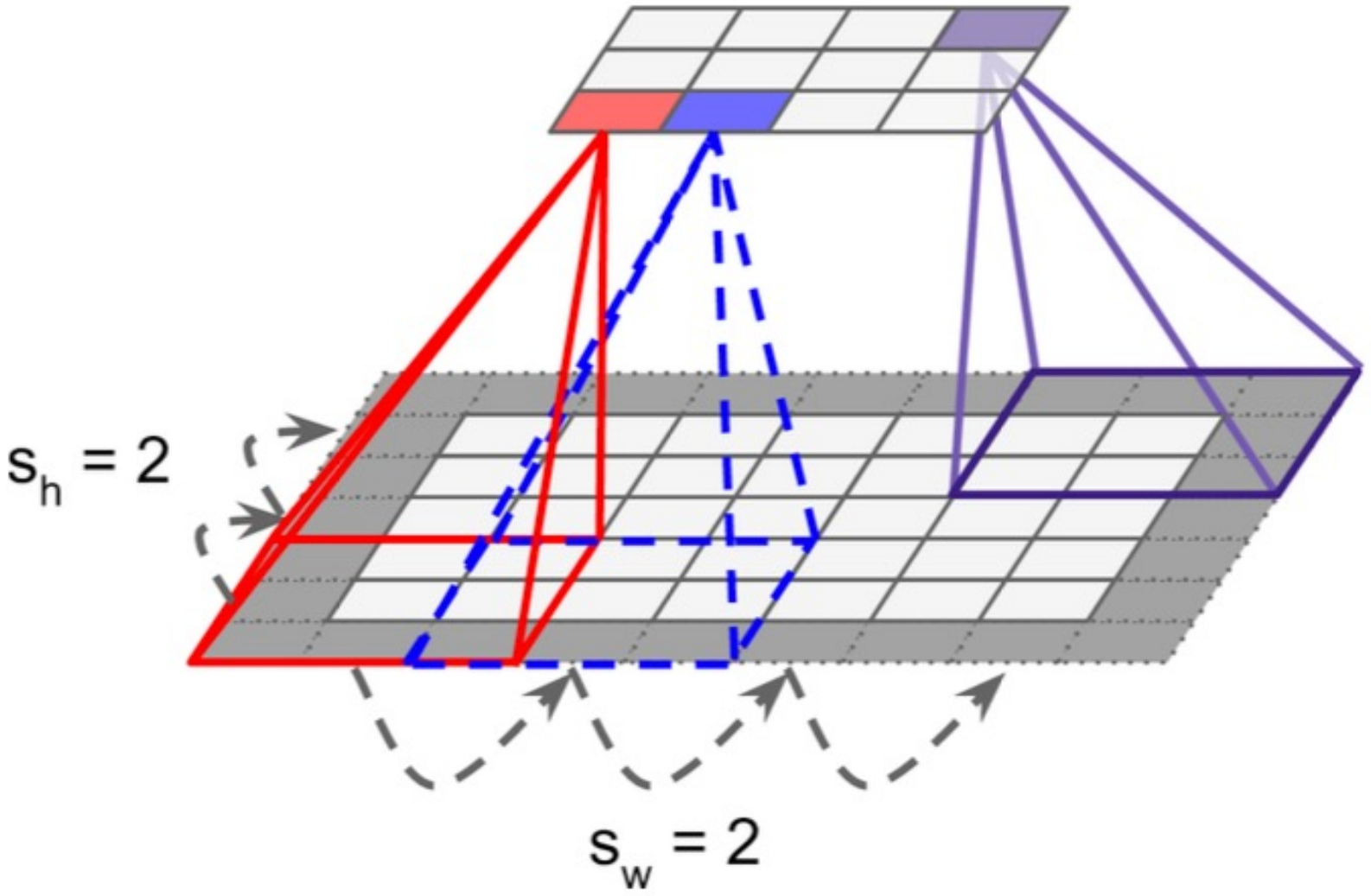
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Typical types of padding

- In principle, you can specify the number of rows/columns to add with padding

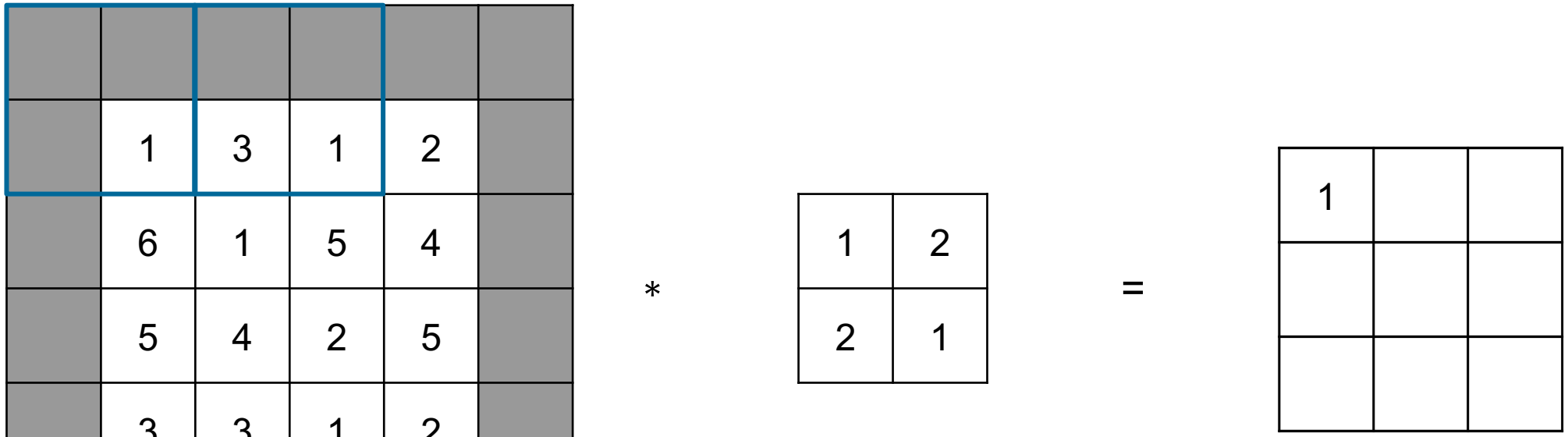
- More commonly, you will use one of “same” or “valid”
  - “same”: add padding as needed so that output size = input size
  - “valid”: no padding is used, so some rows and columns at the bottom-right may be ignored

Additional specification: stride



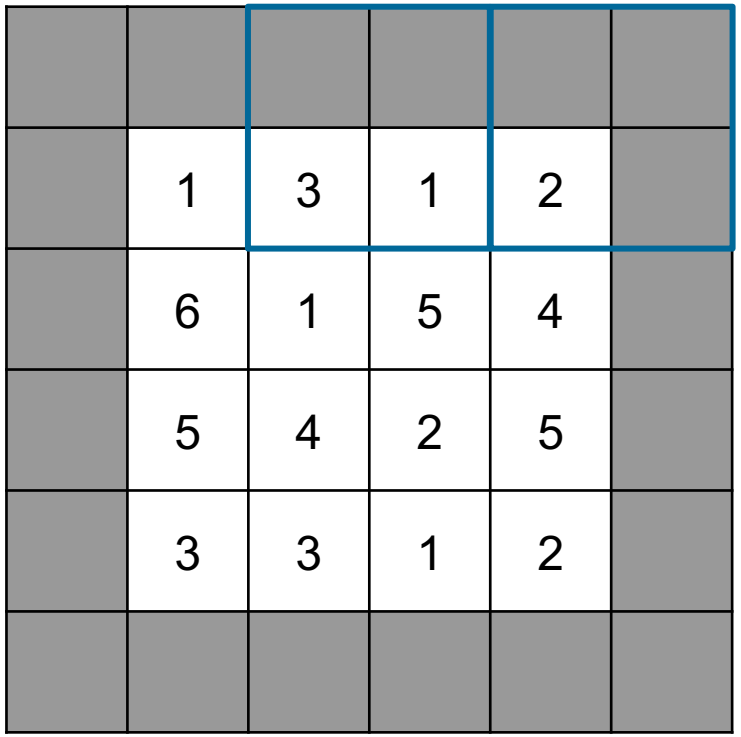
Source: Géron

The convolution operator with padding and stride



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator with padding and stride



*

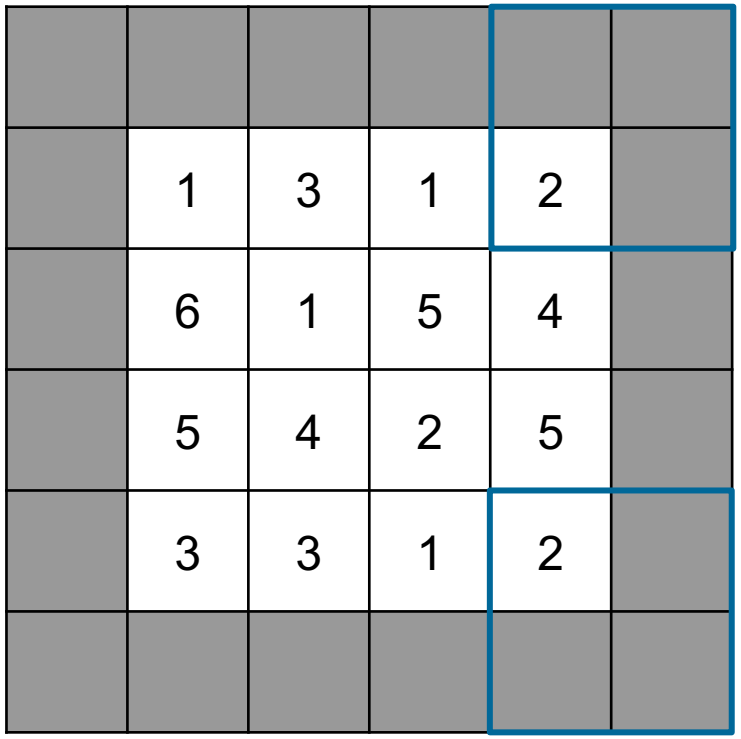
1	2
2	1

=

1	7	

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator with padding and stride



*

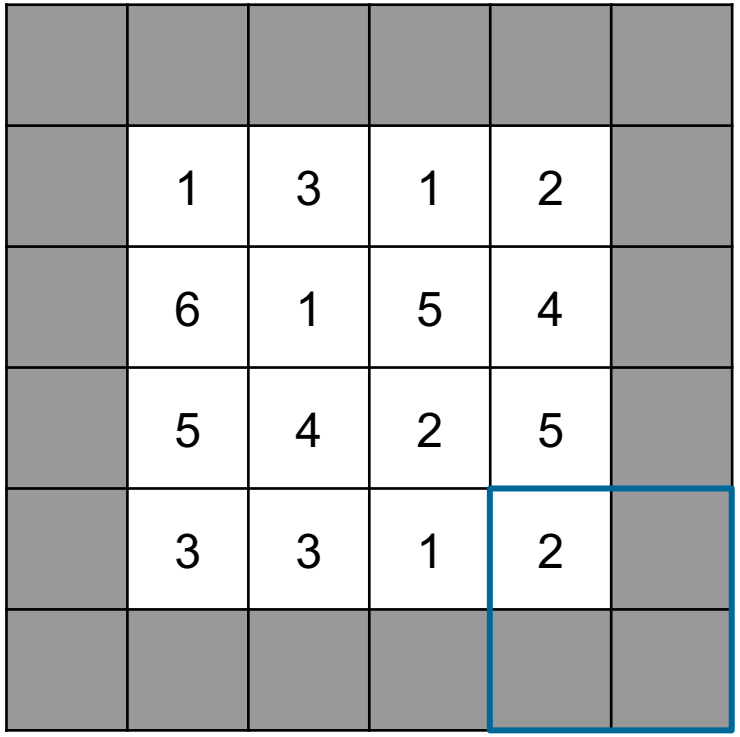
1	2
2	1

=

1	7	4

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator with padding and stride



*

1	2
2	1

=

1	7	4
17	21	14
6	5	2

4x4
$(n_H^{[0]}, n_W^{[0]})$

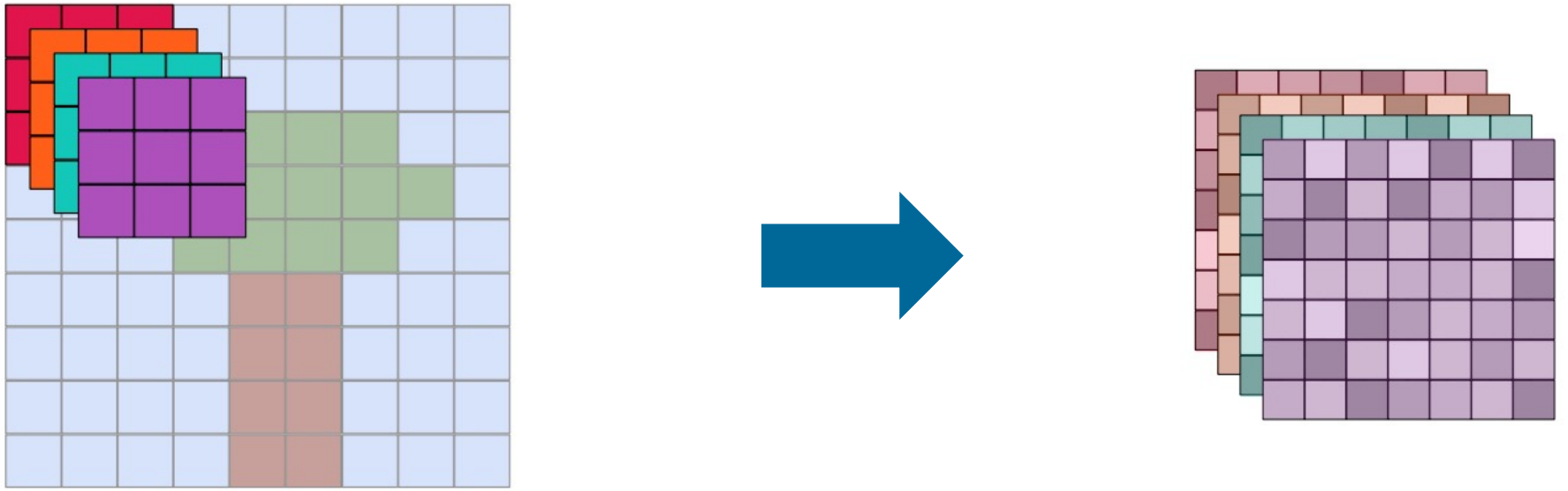
2x2 with padding 1x1 and stride 2x2
(f_H, f_W) with (p_H, p_W) and (s_H, s_W)

3x3
$(n_H^{[1]}, n_W^{[1]})$

$$n_H^{[1]} = \left\lfloor \frac{n_H^{[0]} + 2p_H - f_H}{s_H} + 1 \right\rfloor \quad n_W^{[1]} = \left\lfloor \frac{n_W^{[0]} + 2p_W - f_W}{s_W} + 1 \right\rfloor$$

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

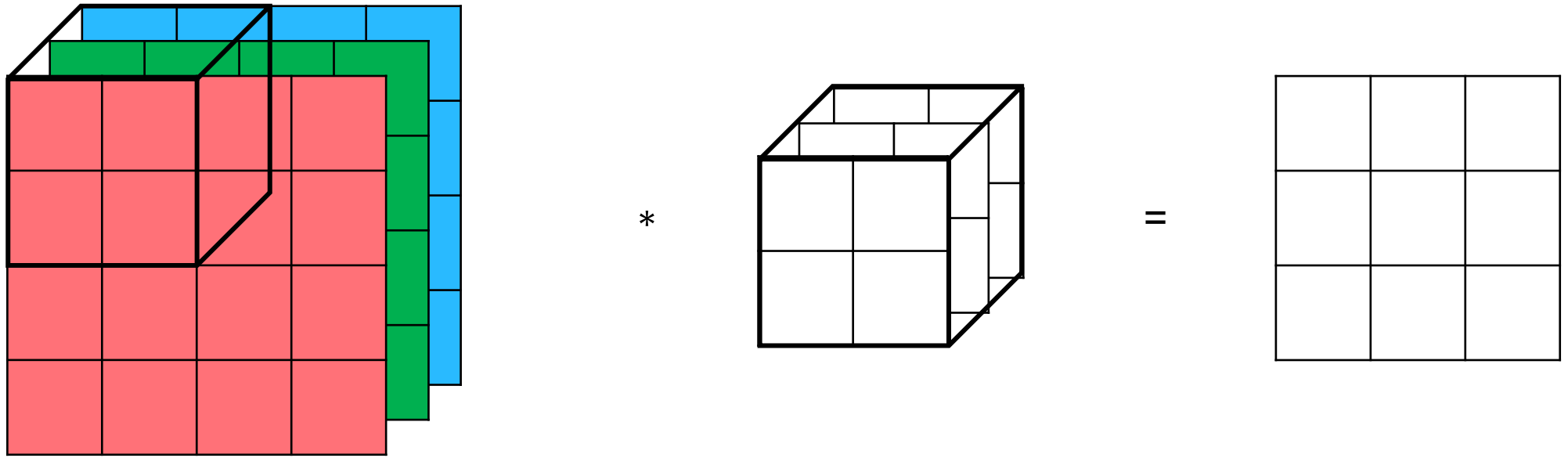
Convolution with multiple channels



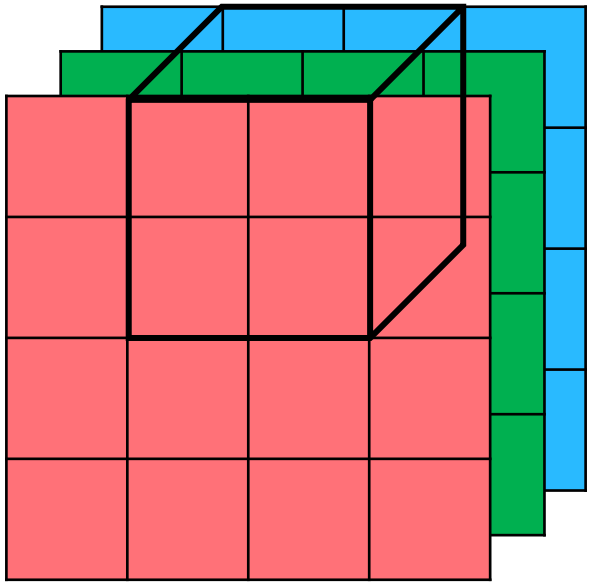
Source: Dieleman

Three-dimensional convolutional layers

Convolution on a 3D array

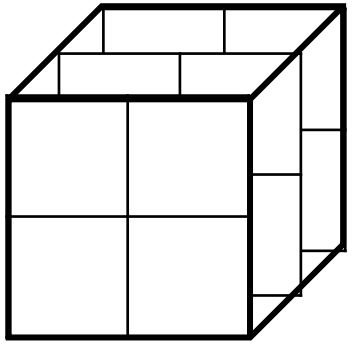


Convolution on a 3D array



4x4x3

*



2x2x3

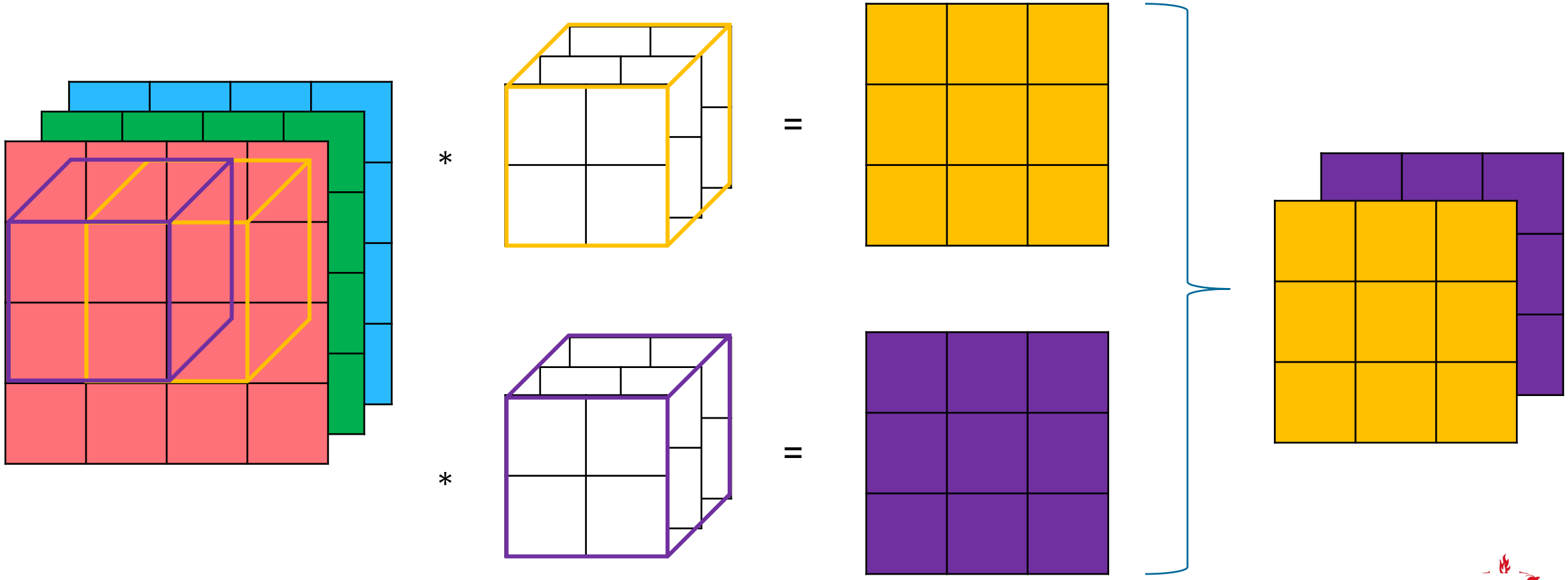
=

20		

3x3

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Multiple 3D convolutions



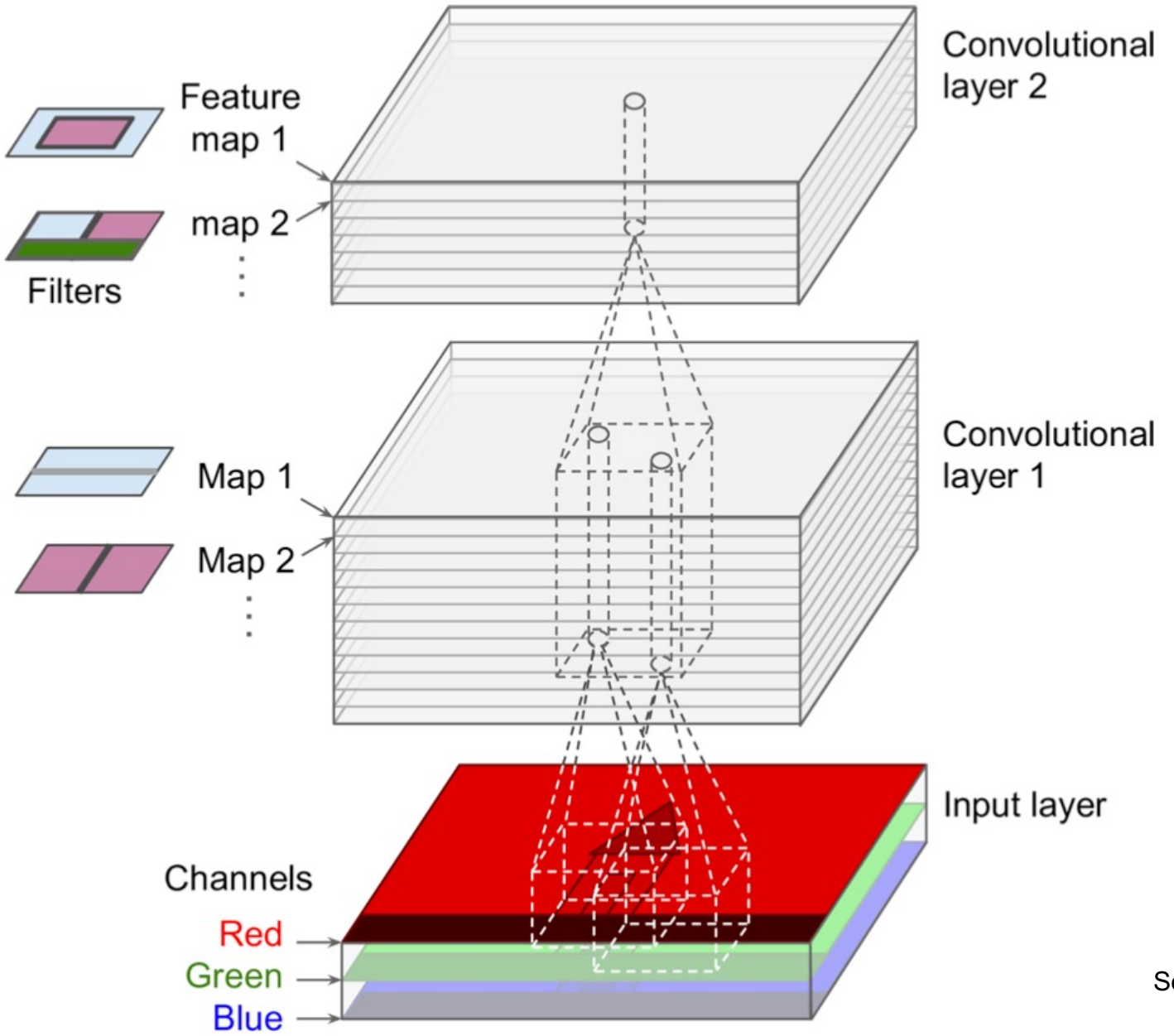
4x4x3
$(n_H^{[0]}, n_W^{[0]}, \mathrm{n}_\mathrm{C}^{[0]})$

2x2x3
$(f_H, f_W, \mathrm{n}_\mathrm{C}^{[0]})$

3x3x2
$(n_H^{[1]}, n_W^{[1]}, \mathrm{n}_\mathrm{C}^{[1]})$

$n_C^{[1]}$= number of filters

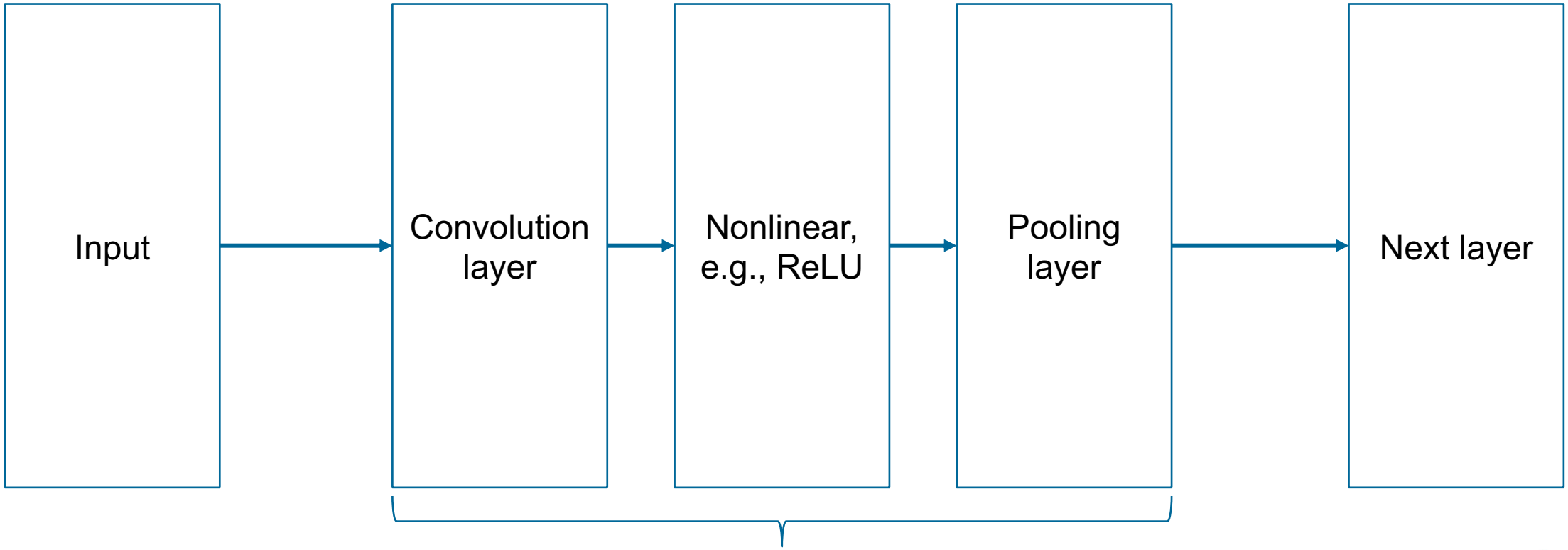
Stacking 3D convolutional layers



Source: Géron

Pooling layers

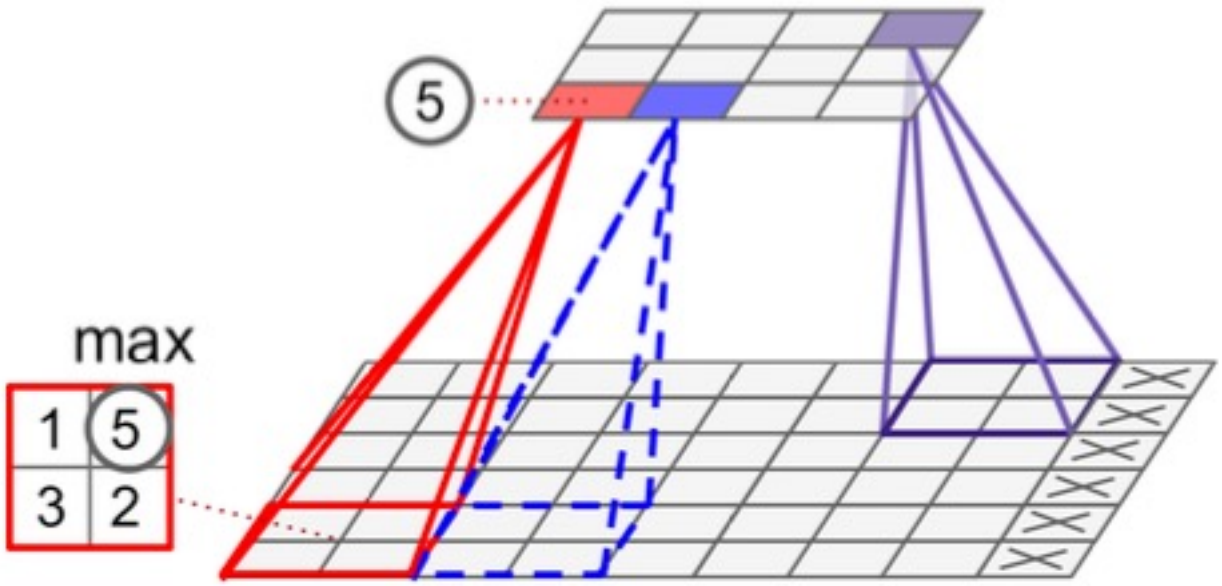
CNN Components



Objective of pooling layers

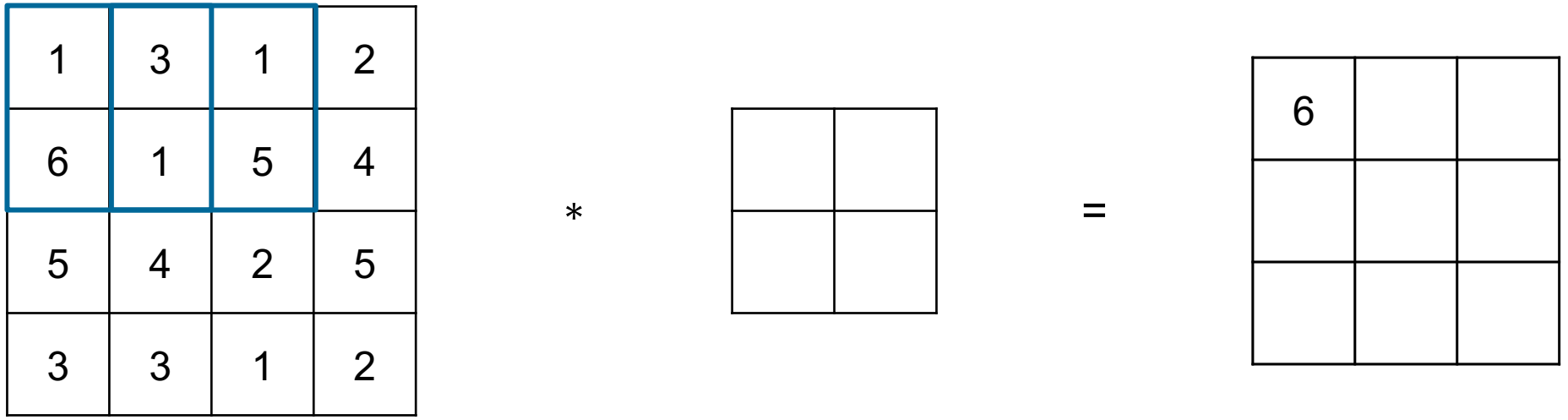
- Subsample (i.e., summarize) the input
  - Reduced computational load
  - Reduced memory usage
  - Fewer parameters (and, thus, less overfitting)

Max pooling



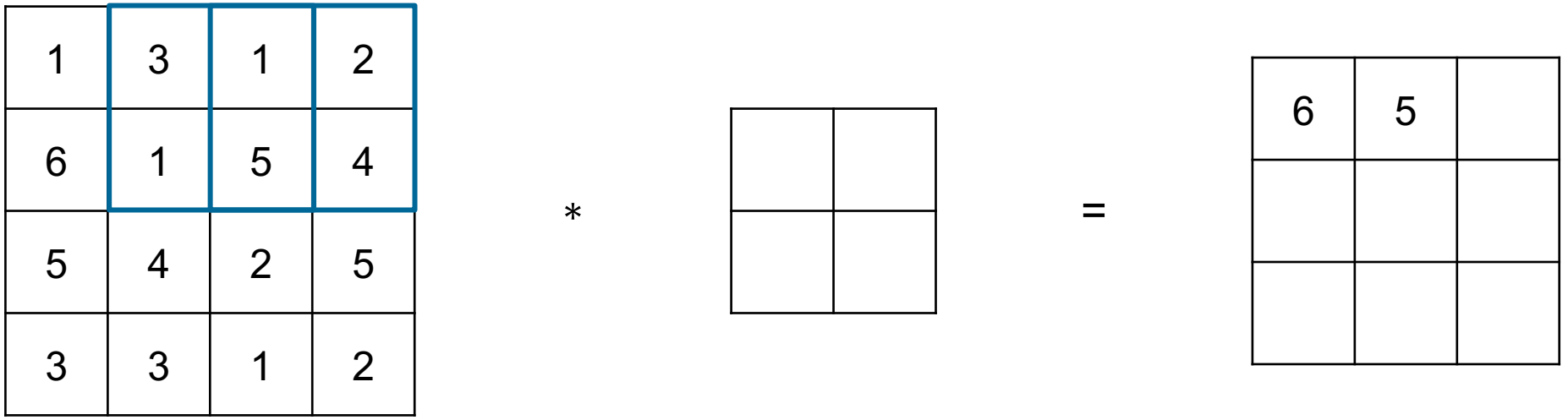
Source: Géron

Max pooling in practice



Source: Géron

Max pooling in practice



Source: Géron

Max pooling in practice

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

=

6	5	5

Source: Géron

Max pooling in practice

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

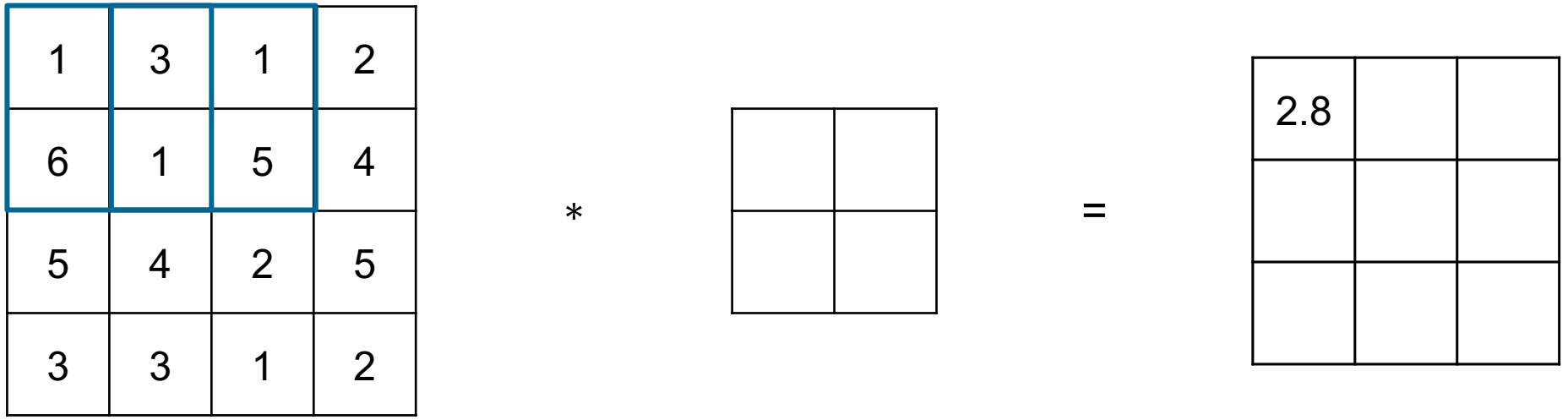
*

=

6	5	5
6	5	5
5	4	5

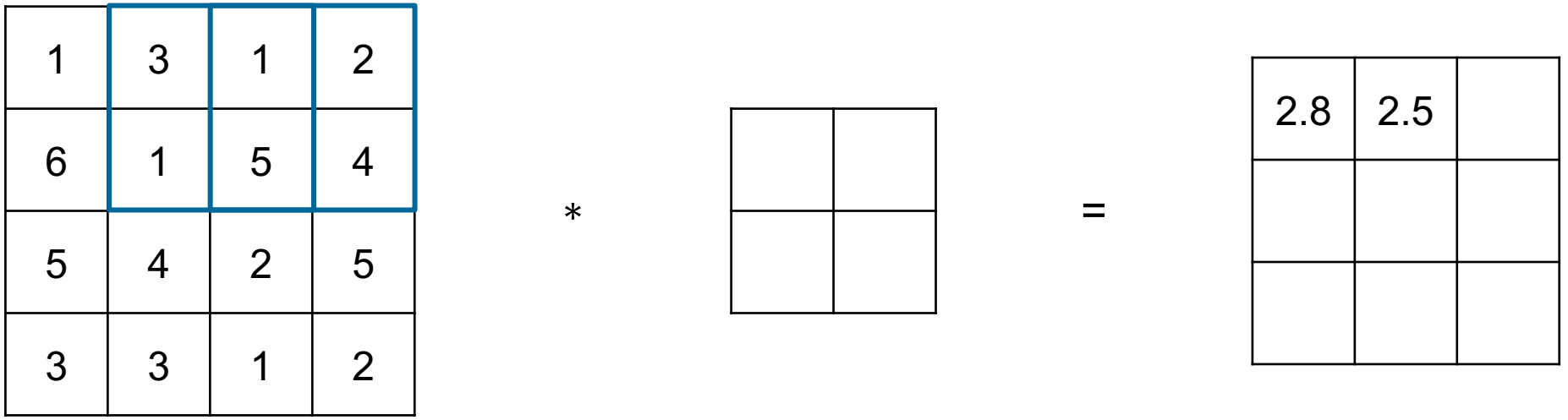
Source: Géron

Average pooling in practice



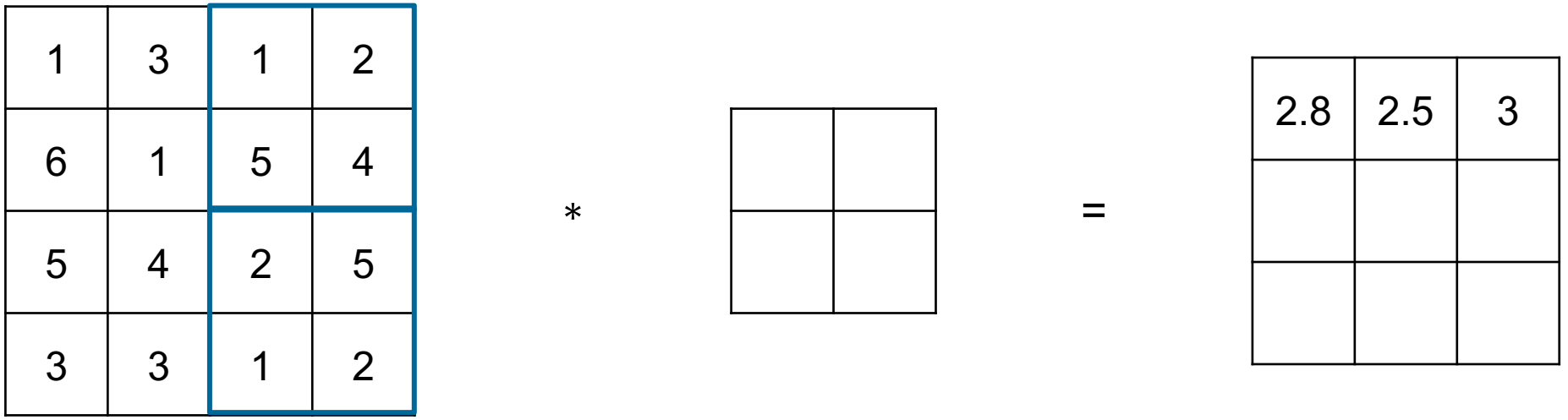
Source: Géron

Average pooling in practice



Source: Géron

Average pooling in practice



Source: Géron

Average pooling in practice

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

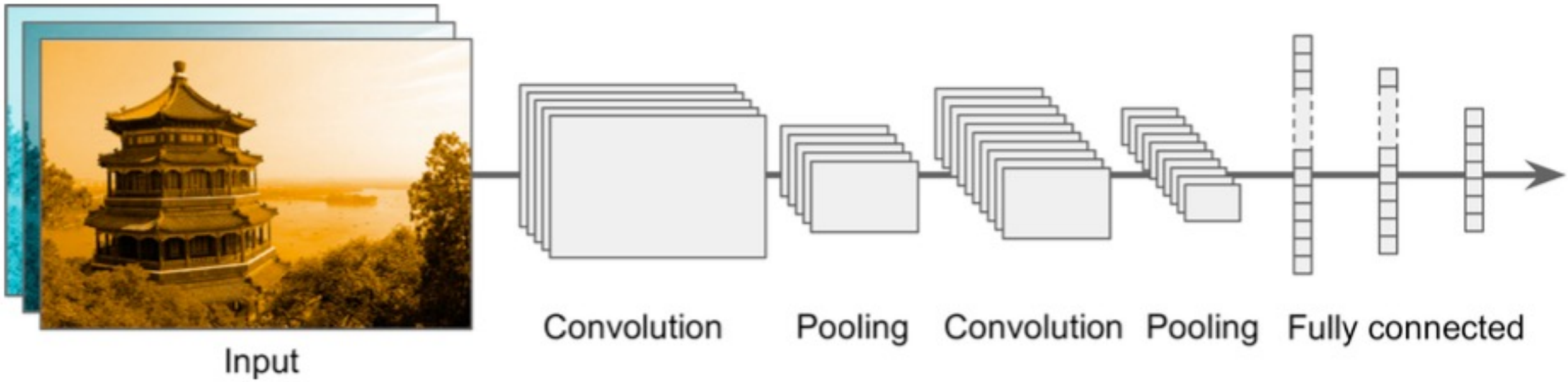
=

2.8	2.5	3
4	3	4
3.8	2.5	2.5

Source: Géron

Typical architectures

Typical architecture

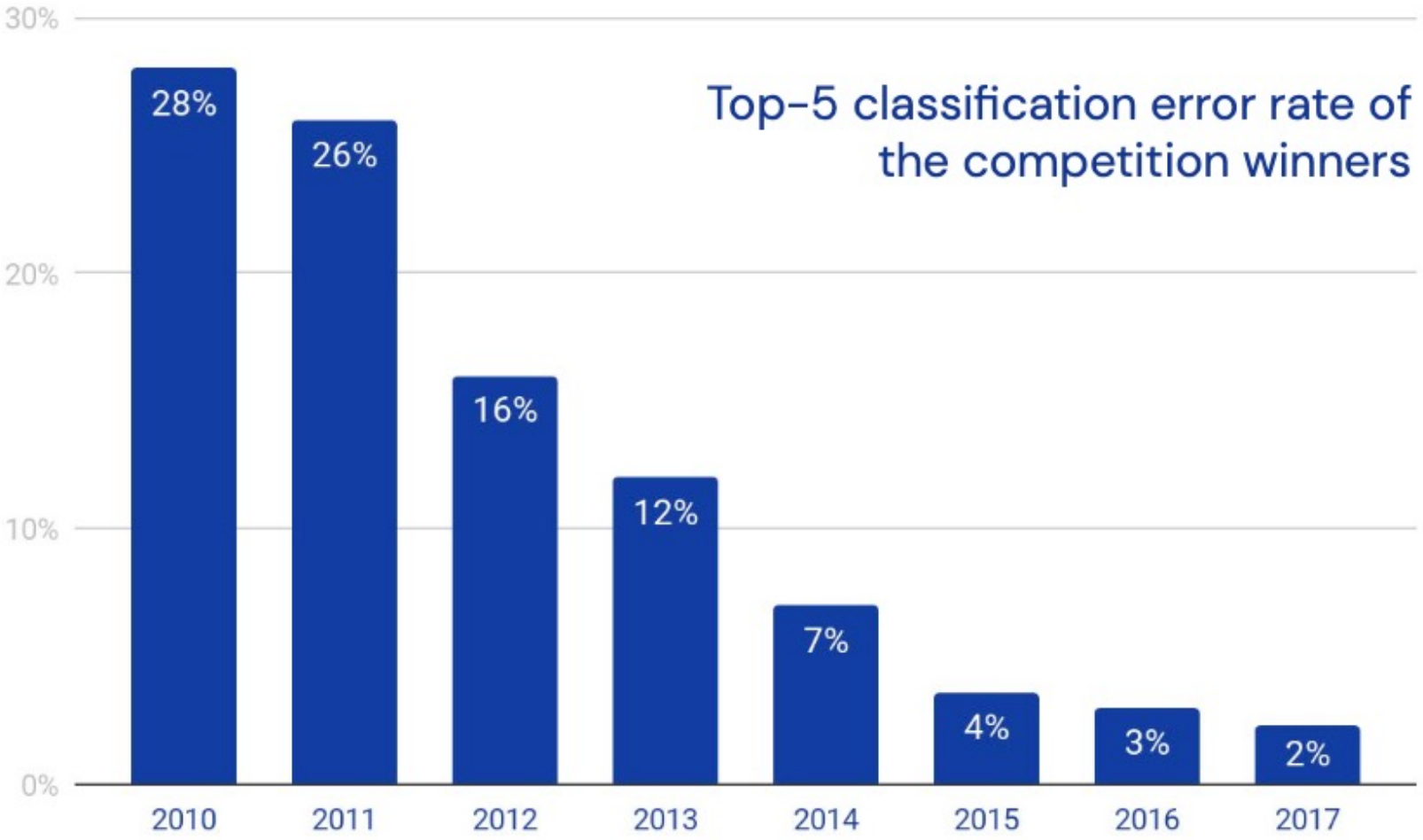


Source: Géron

The ImageNet challenge

- Major computer vision benchmark for image classification (and later, more advanced stuff)
- From 2010-2017 (now transferred to Kaggle)
- 1.4 mio images in 1,000 classes
- Models need to predict the top 5 most likely labels
  - Winner: lowest “top-5 error rate” – percentage of test images for which true label is not among top 5 most likely labels
- More information: https://www.image-net.org/challenges/LSVRC/index.php

Architectures over time



Source: Dieleman

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON


See you in class!

Sources

- Bhaskhar, 2021, Introduction to Deep Learning: https://cs229.stanford.edu/syllabus.html
- DeepLearning.AI, n.d.: deeplearning.ai
- Dieleman, 2020, Lecture 3: Convolutional Neural Networks: https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L3%20-%20UUCLxDeepMind%20DL2020.pdf
- Géron, 2019, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
- Goodfellow, Bengio, Courville, 2016, The Deep Learning Book: http://www.deeplearningbook.org
- Liang, 2016, Introduction to Deep Learning: https://www.cs.princeton.edu/courses/archive/spring16/cos495/
- Lin et al., 2014, Microsoft COCO: Common Objects in Context: https://arxiv.org/pdf/1405.0312.pdf