Applied Deep Learning

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

www.bayes.city.ac.uk

Learning objectives of today

Goals:

- Anomaly detection: understand what anomalies are, why we would want to detect them, and how we can do so
- CNNs: Understand how convolution works and how it can be implemented in TensorFlow

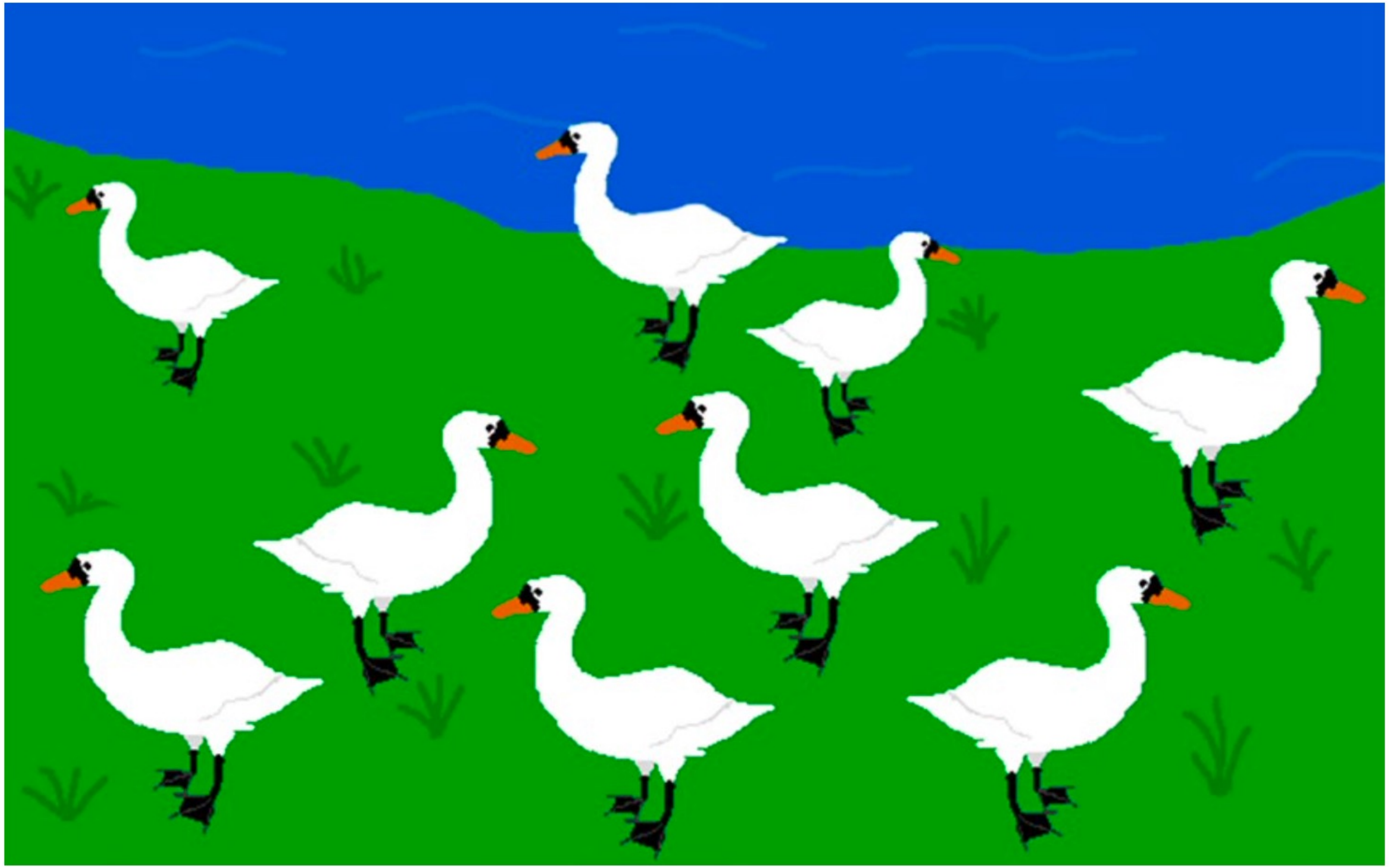
How will we do this?

- We will start by discussing anomaly detection and the different techniques used to perform it in different settings
- We will use this to lead over into the student presentations
- We then turn to computer vision, and the fundamental architectural innovation that makes it work: convolutional layers

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

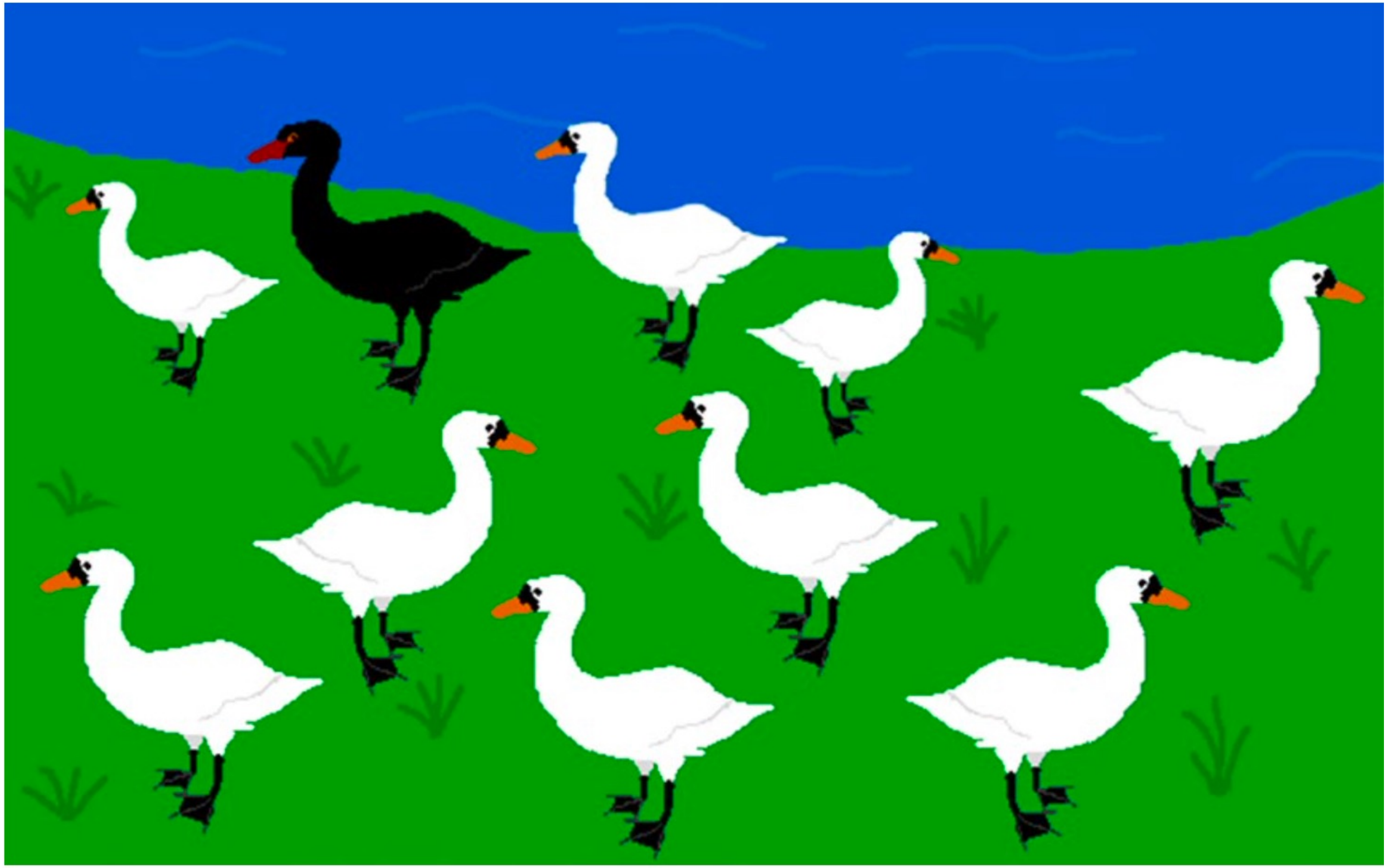
Anomaly detection

What color are swans?



Source: Alla

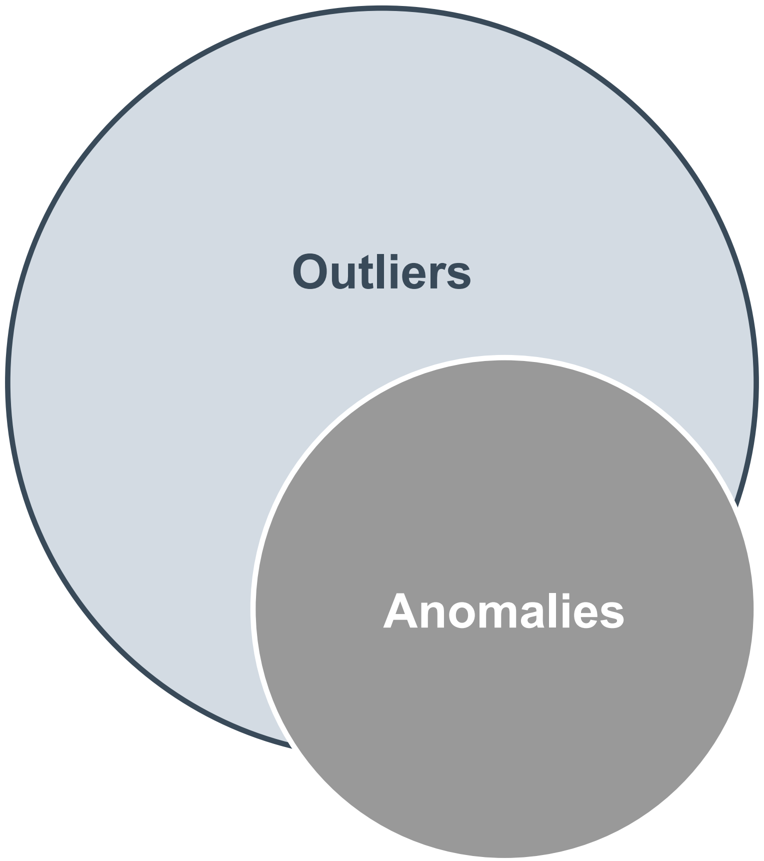
What color are swans?



Source: Alla

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Outliers and anomalies



Outliers:

- Data points that are distinctly different from other data points
- Can be caused by unavoidable random errors or by systematic errors relating to how data was sampled

Anomalies:

- Outliers or other values that are not expected to exist
- Can be context- or pattern-based:
  - Context: exceptionally high credit card spending on Black Friday versus near-simultaneous spending in New York and London
  - Pattern: high credit card spending every Saturday versus high spending on a day where spending is low in other weeks

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

What are possible anomalies and how would we detect them?

Consider the following situations:

- A machine produces thousands of screws per minute, every few days the type of screw is changed
- A software developer for a bank downloads a large number of entries from a customer database
- An intermediary supplies fair trade coffee beans

What is the expected outcome in each case?

What is an anomalous outcome?

What data do we observe?

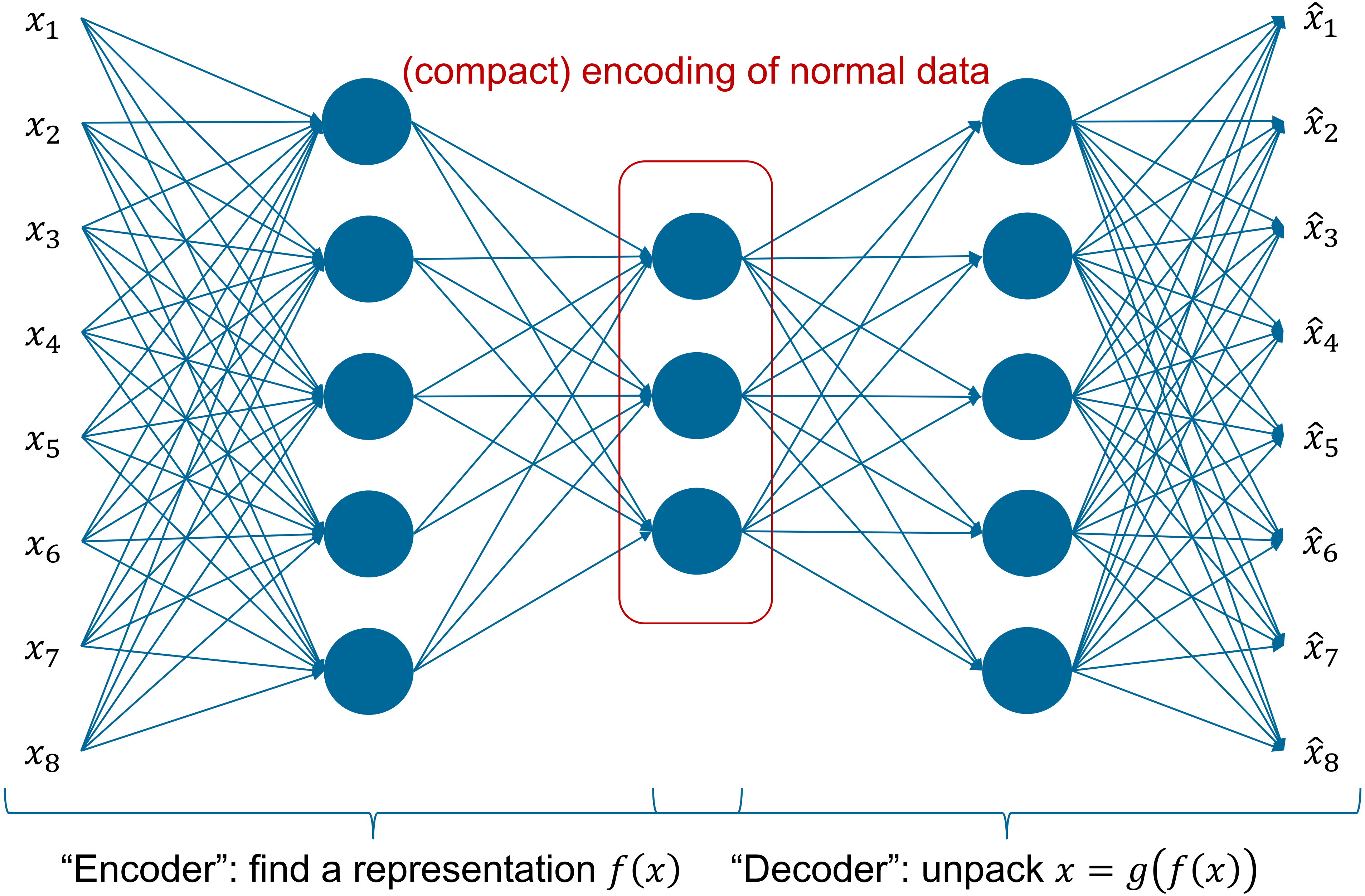
Detecting anomalies

Supervised anomaly detection:
- A fancy way of saying classification – learn to differentiate between two classes
- We can use the standard toolbox
- Upside: When feasible, usually the most failsafe method
- Downside: only works if we know how normal and anormal data looks like

Semi-supervised anomaly detection:
- Learn an efficient representation of normal data and then try to apply this to new data coming in
- We can use autoencoders and other tools
- Upside: we don't need to know how anormal data looks like
- Downside: still need to be sure that our normal data is actually normal

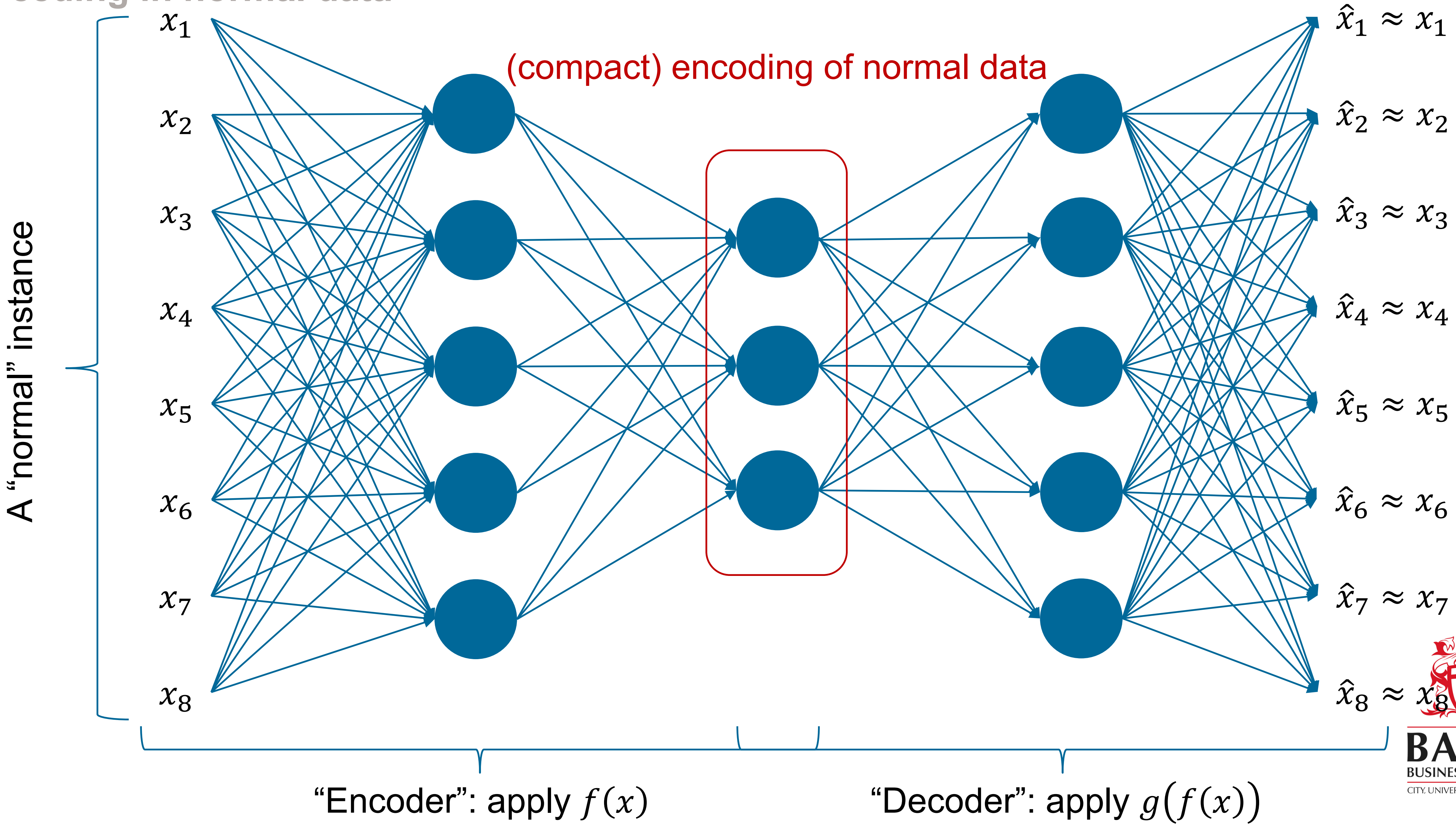
Unsupervised anomaly detection:
- Learn "how far" datapoints are from each other and recognize the ones that are far away from anything else
- We can use isolation forests and other tools
- Upside: we can work with any kind of data
- Downside: we don't have many guarantees

Anomaly detection with an autoencoder



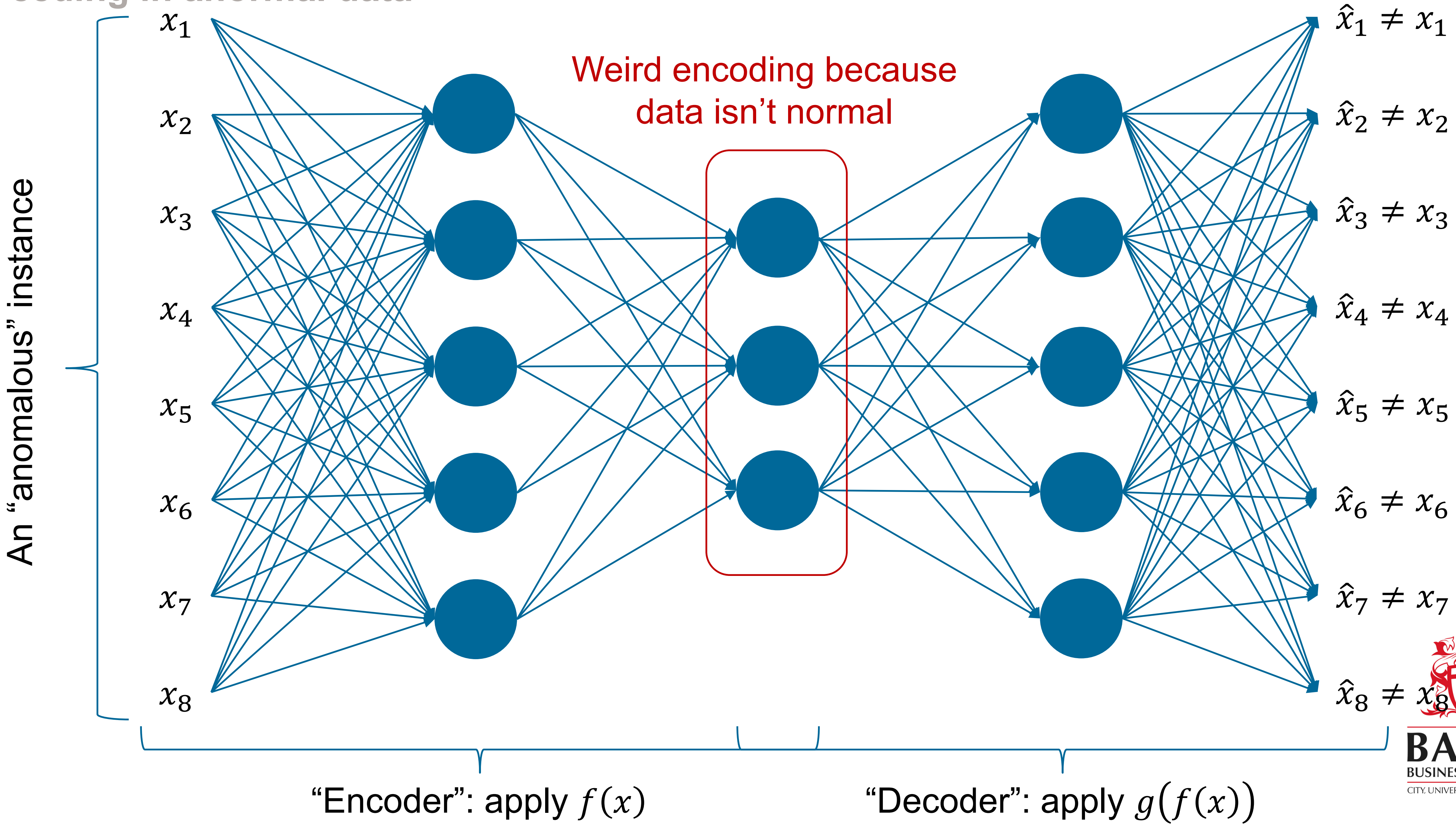
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Feeding in normal data



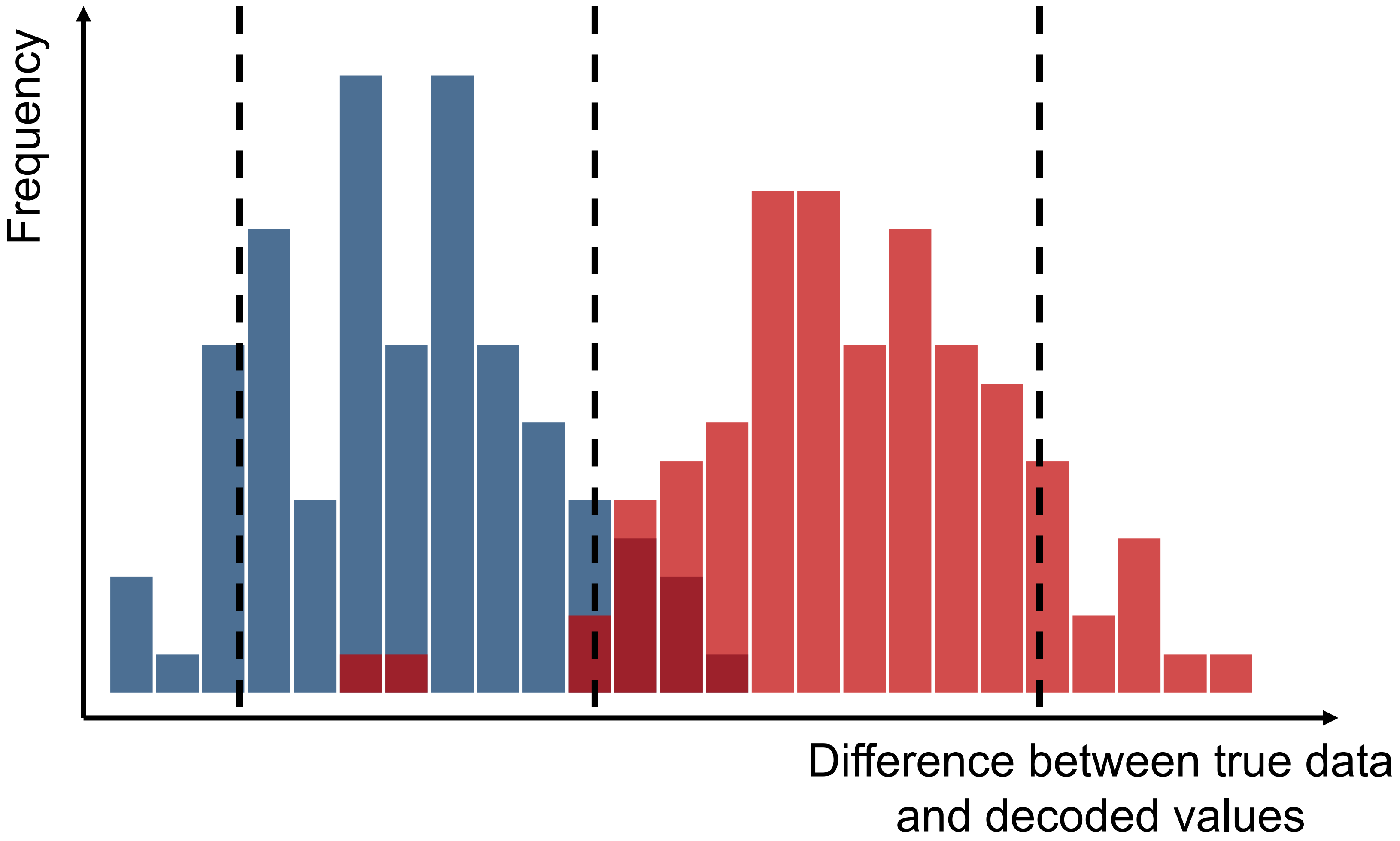
BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Feeding in anormal data



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

What we should be observing



Your turn – what did you find?

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Computer vision and convolutional layers

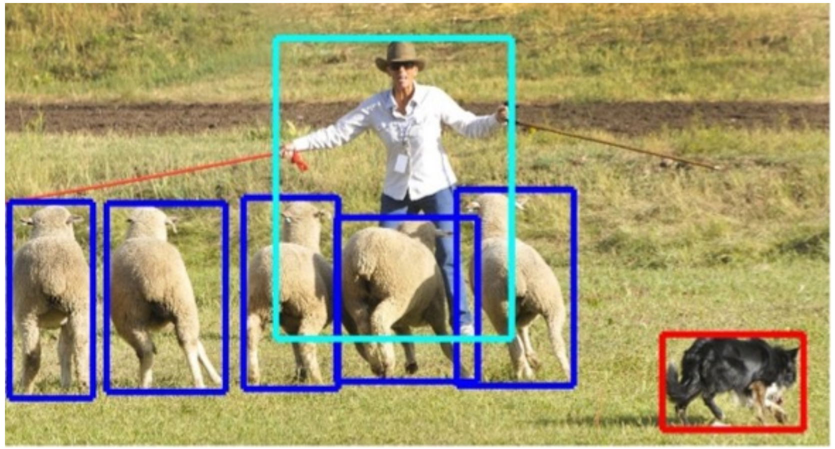
Importance of computer vision in business processes

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

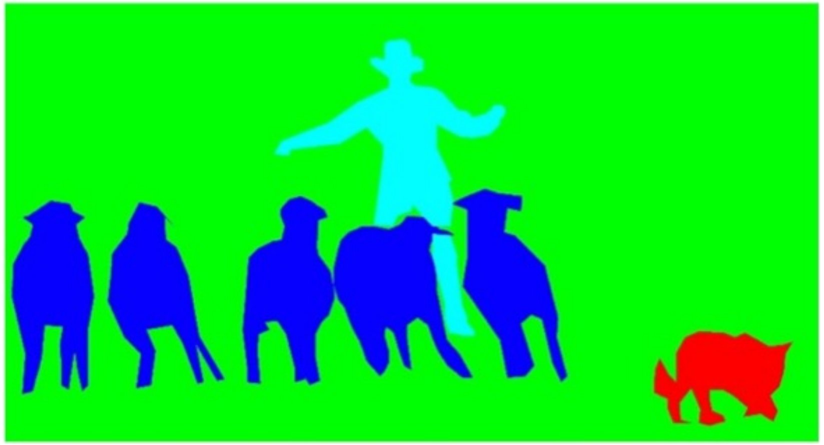
Typical computer vision problems

Image classification

Object detection



Semantic segmentation



Neural style transfer

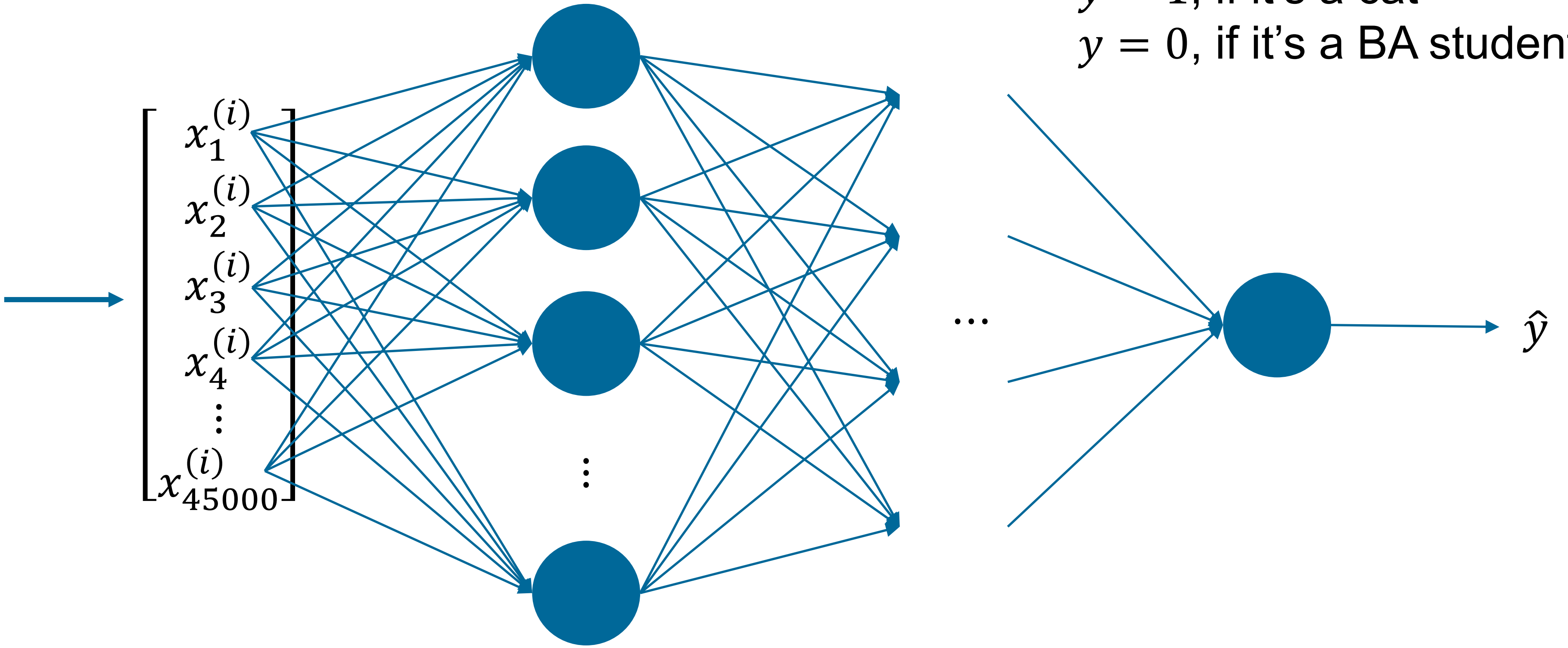
Source: Lin, reiinakano.com

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Challenges to deep learning using large images

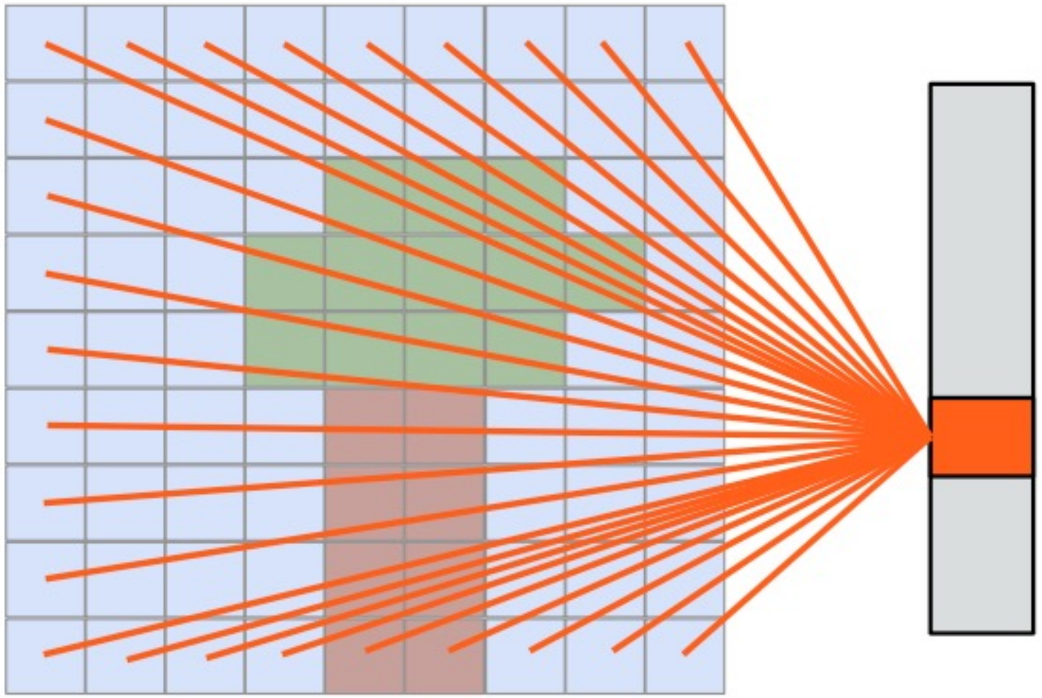
$y = 1$, if it's a cat
$y = 0$, if it's a BA student

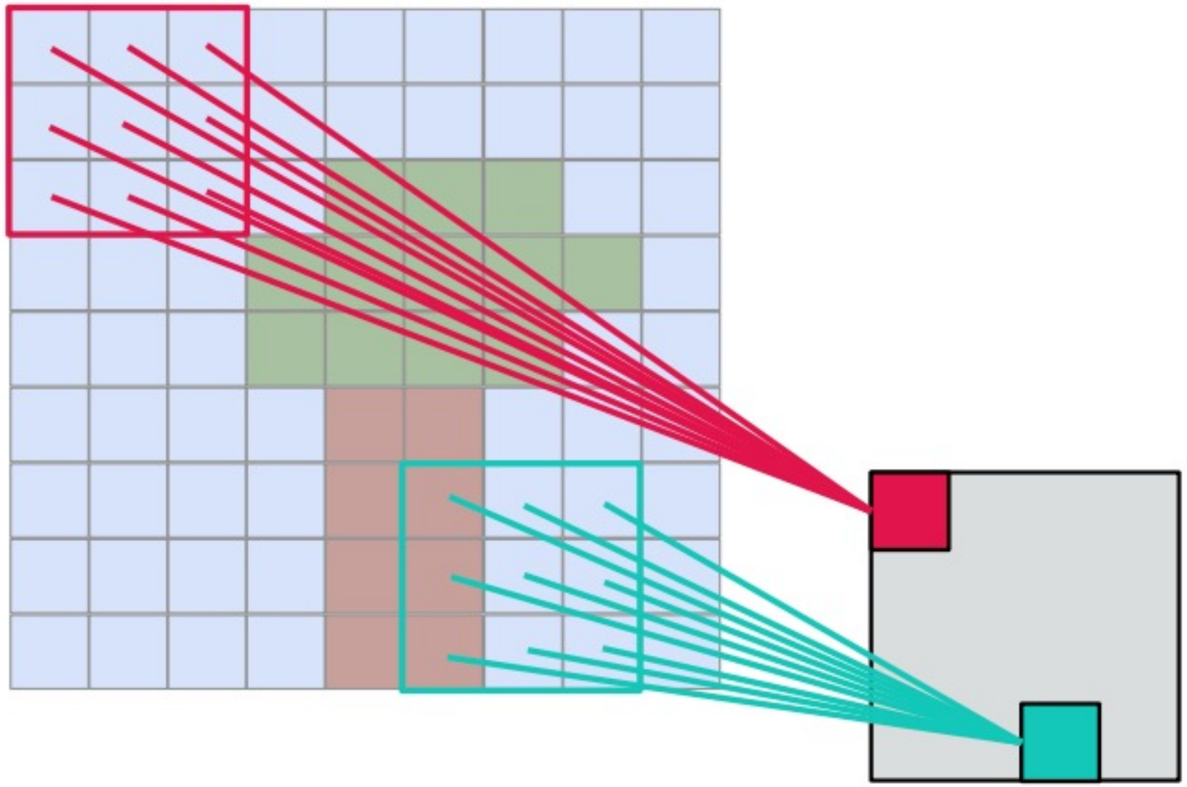
150x100x3



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

From fully connected to locally connected





Source: Dieleman

An image filter

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			

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

An image filter

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

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

An image filter

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	

An image filter

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

An image filter

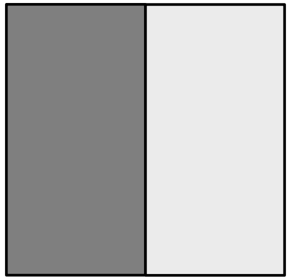
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

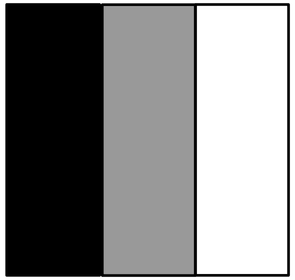
*

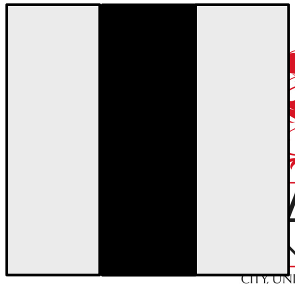
1	0	-1
1	0	-1
1	0	-1

=

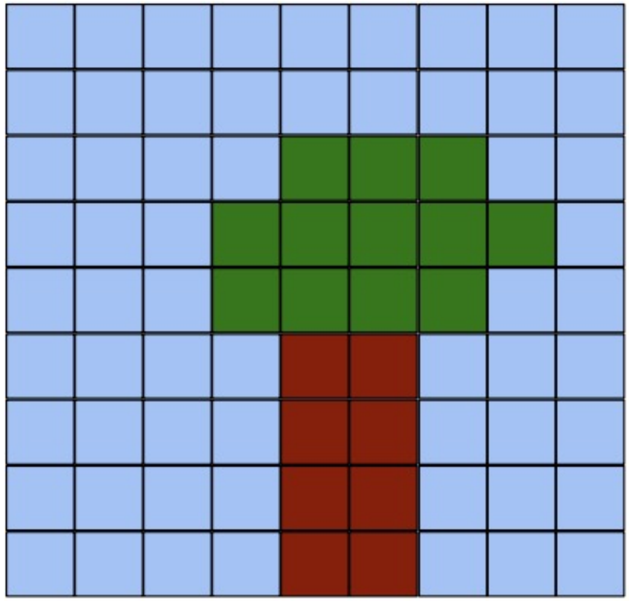
0	30	30	0
0	30	30	0
0	30	30	0



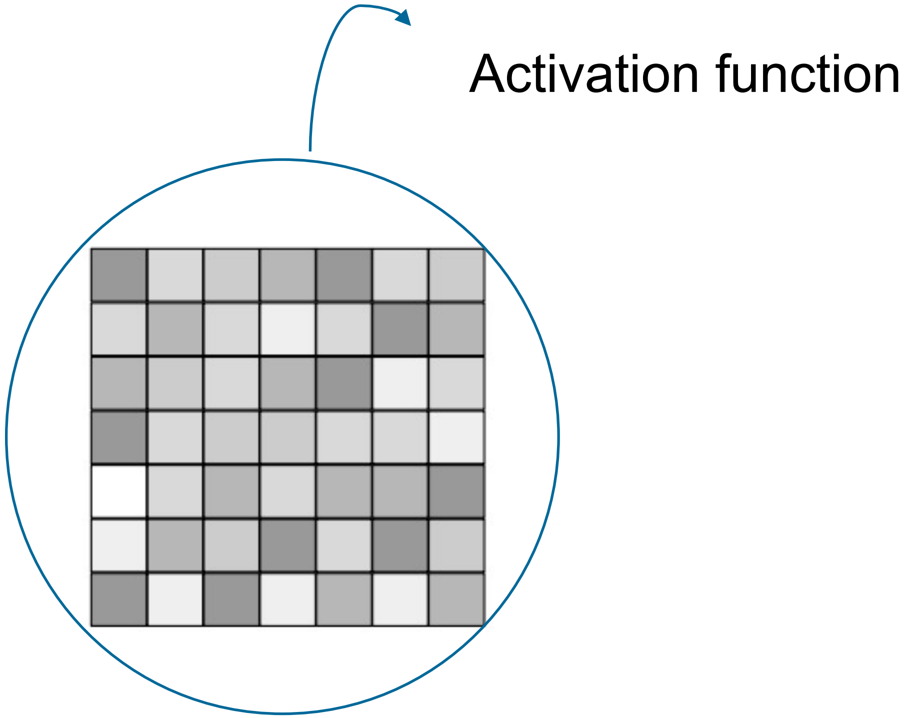




Convolutional layers: learning the filter

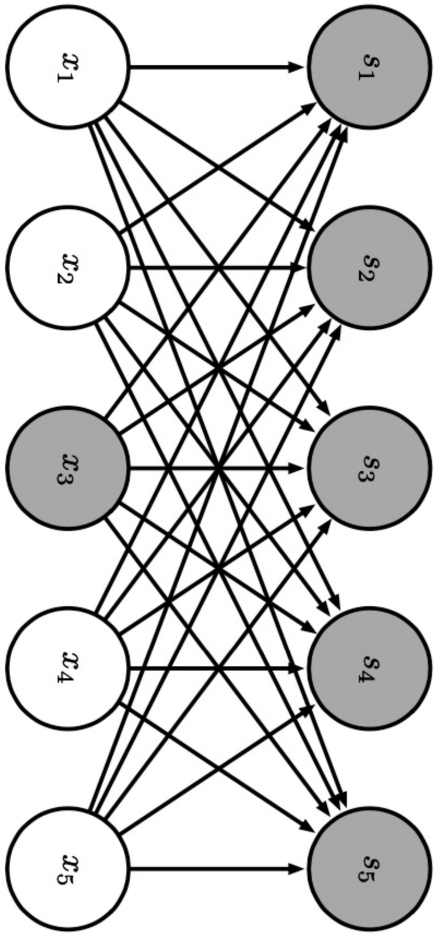


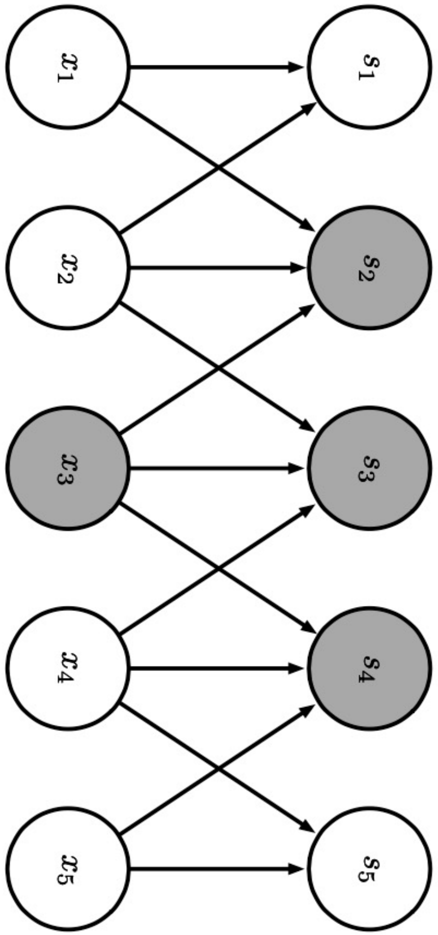
$$* \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \end{bmatrix} + b =$$



Source: Dieleman

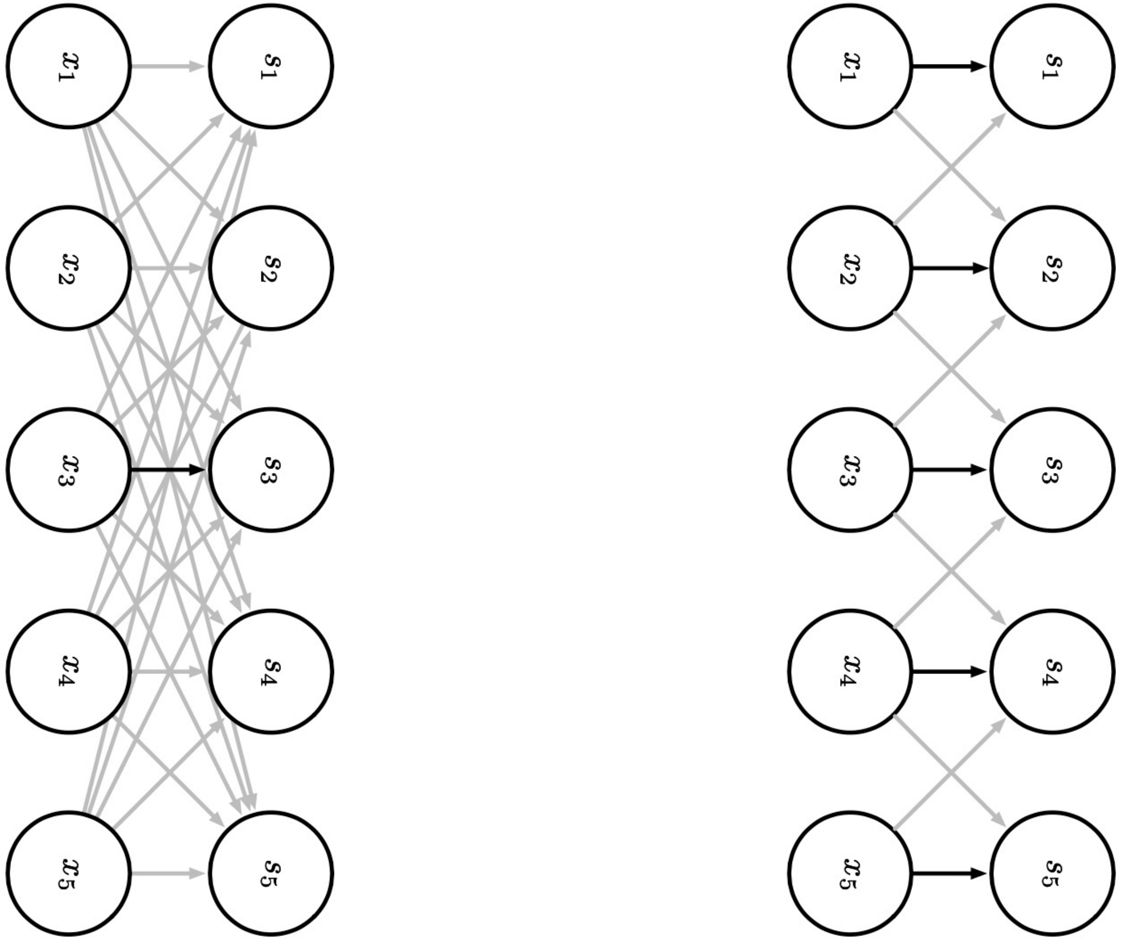
Less parameters required





Source: Goodfellow

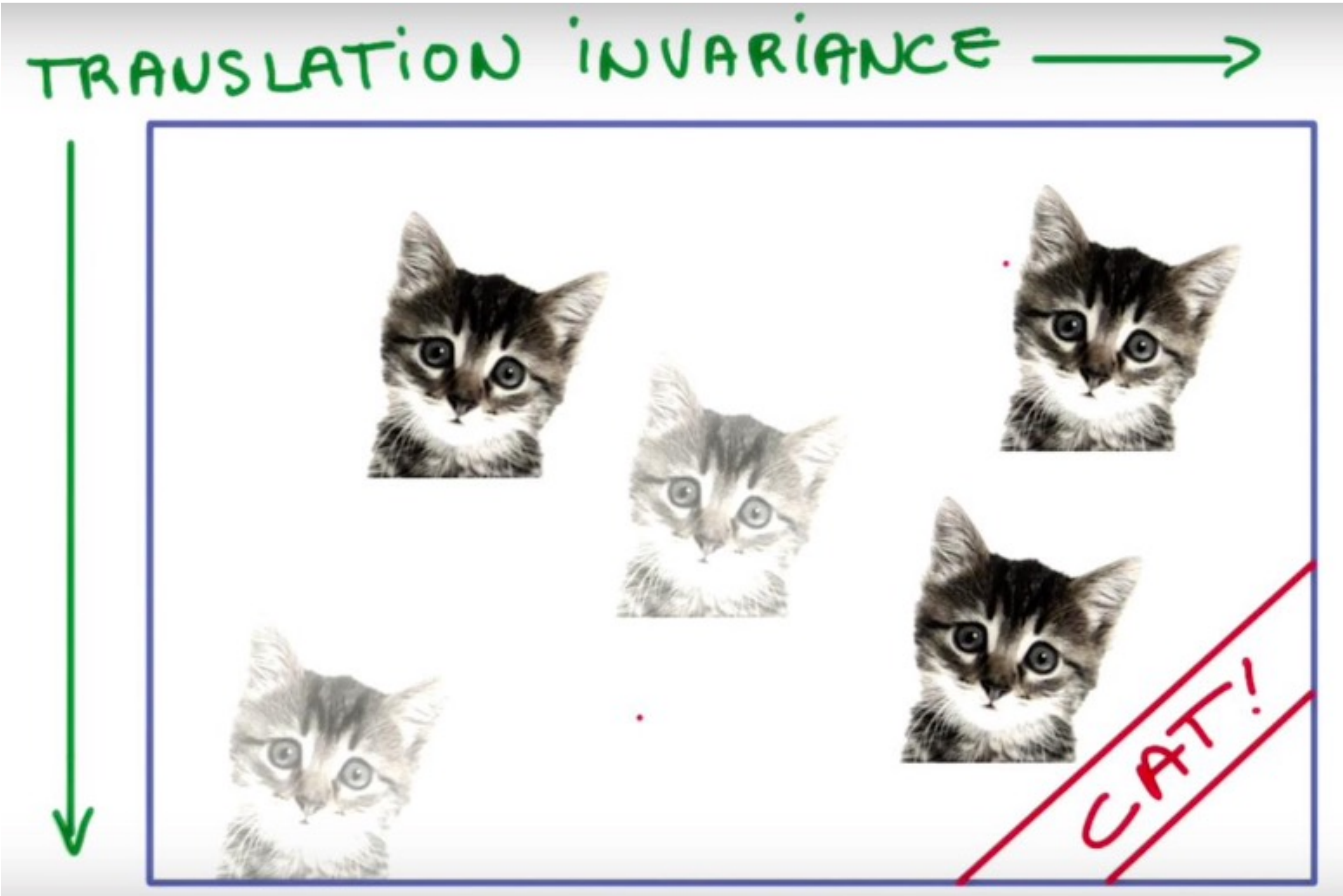
Parameters are shared



Source: Goodfellow

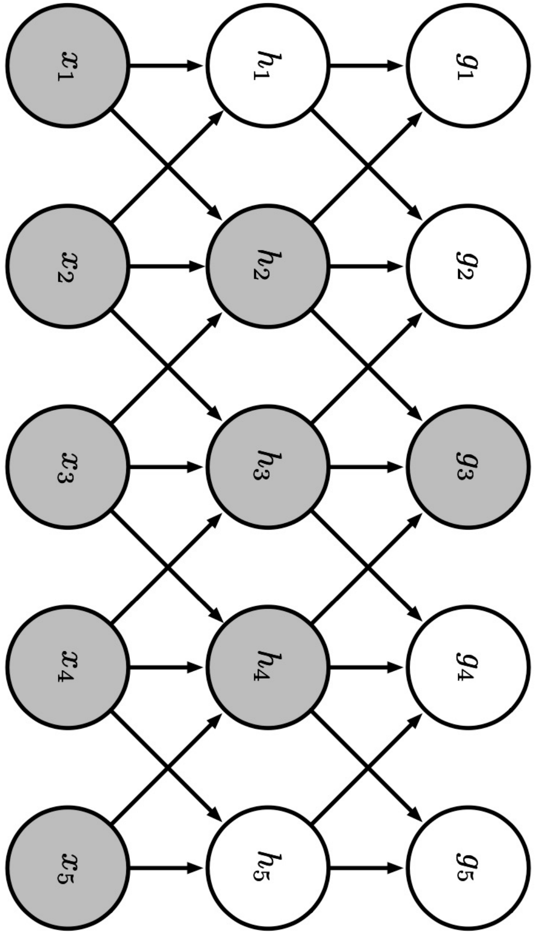
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Translation invariance



Source: Bhaskhar

Hierarchical setup



Source: Goodfellow

With convolution, the center matters more

$$\begin{bmatrix} 1 & 3 & 1 & 2 \\ 6 & 1 & 5 & 4 \\ 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 20 & 12 & 19 \\ 22 & 21 & 22 \\ 22 & 15 & 16 \end{bmatrix}$$

The convolution operator with padding

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

*

1	2
2	1

=

1				

The convolution operator with padding

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

*

1	2
2	1

=

1	4			

The convolution operator with padding

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

*

1	2
2	1

=

1	4	7		

The convolution operator with padding

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

*

1	2
2	1

=

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

The convolution operator with padding and stride

$$
\begin{bmatrix}
 & & & & & \\
 & 1 & 3 & 1 & 2 & \\
 & 6 & 1 & 5 & 4 & \\
 & 5 & 4 & 2 & 5 & \\
 & 3 & 3 & 1 & 2 & \\
 & & & & &
\end{bmatrix}
*
\begin{bmatrix}
1 & 2 \\
2 & 1
\end{bmatrix}
=
\begin{bmatrix}
1 & & \\
 & & \\
 & &
\end{bmatrix}
$$

The convolution operator with padding and stride

$$
\begin{array}{|c|c|c|c|c|c|}
\hline
 & & & & & \\
\hline
 & 1 & 3 & 1 & 2 & \\
\hline
 & 6 & 1 & 5 & 4 & \\
\hline
 & 5 & 4 & 2 & 5 & \\
\hline
 & 3 & 3 & 1 & 2 & \\
\hline
 & & & & & \\
\hline
\end{array}
\;*\;
\begin{array}{|c|c|}
\hline
1 & 2 \\
\hline
2 & 1 \\
\hline
\end{array}
\;=\;
\begin{array}{|c|c|c|}
\hline
1 & 7 & \\
\hline
 & & \\
\hline
 & & \\
\hline
\end{array}
$$

The convolution operator with padding and stride

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

*

1	2
2	1

=

1	7	4

The convolution operator with padding and stride

$$
\begin{bmatrix} & & & & & \\ & 1 & 3 & 1 & 2 & \\ & 6 & 1 & 5 & 4 & \\ & 5 & 4 & 2 & 5 & \\ & 3 & 3 & 1 & \boxed{2} & \\ & & & & & \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 7 & 4 \\ 17 & 21 & 14 \\ 6 & 5 & 2 \end{bmatrix}
$$

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

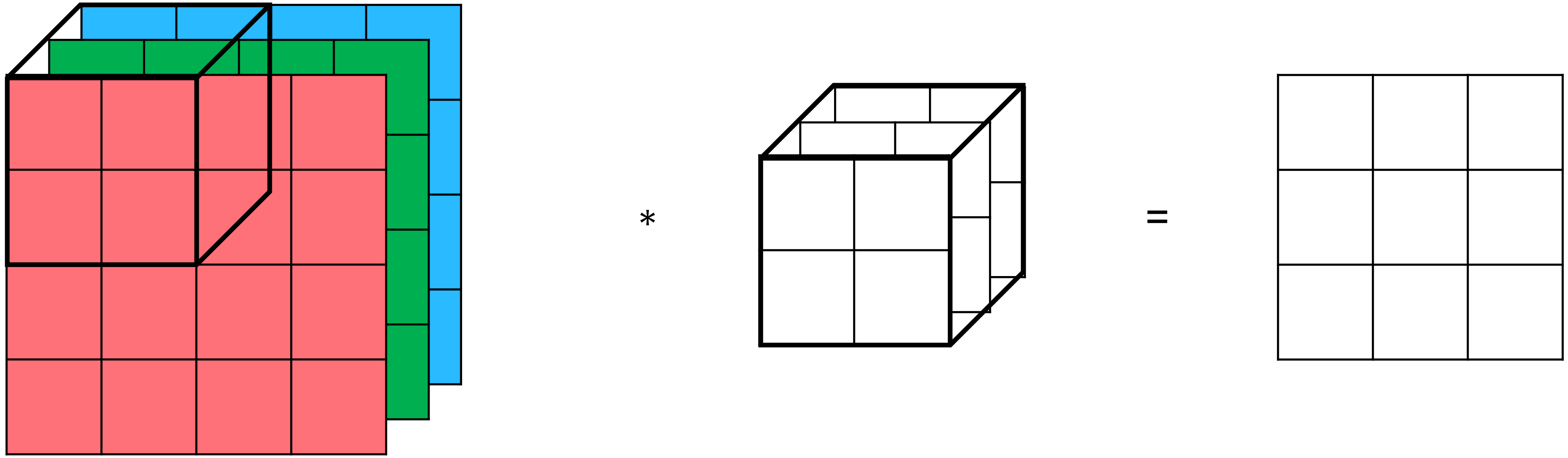
2x2 with padding 1x1 and stride 2x2
(f_H, f_W) with (p_H, p_W) and (s_H, s_W)

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

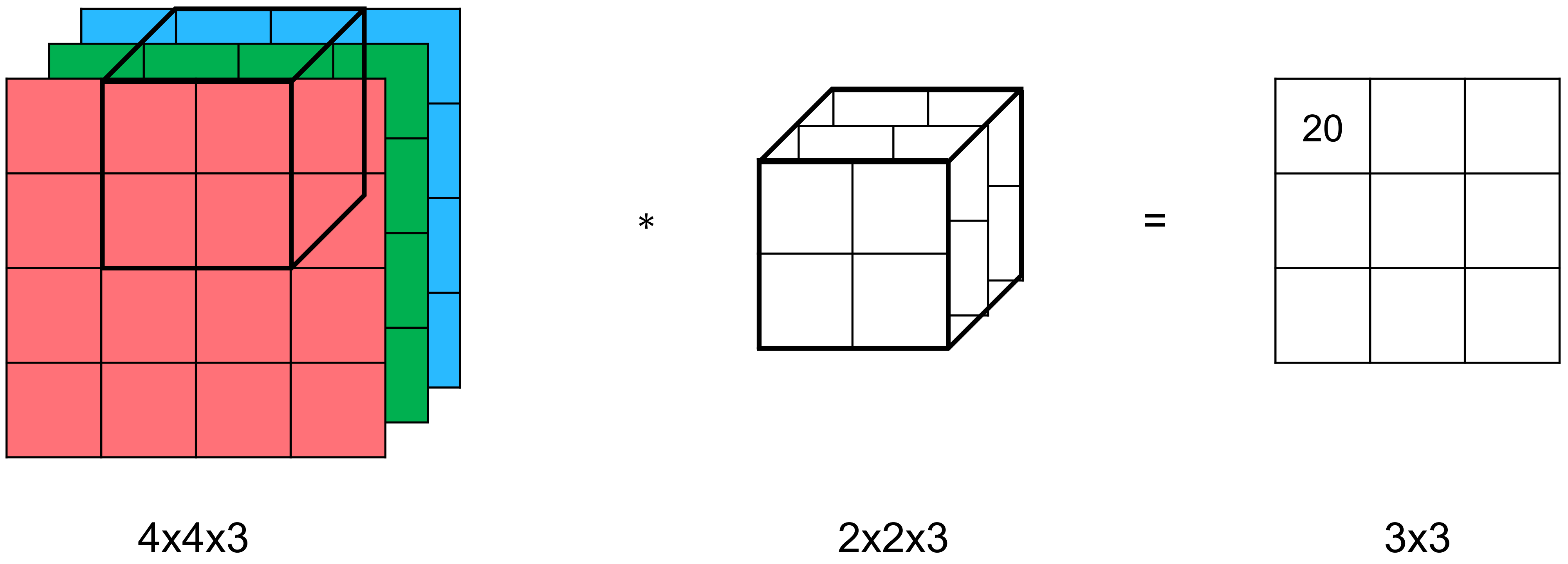
$$n_H^{[1]} = \left\lfloor \frac{n_H^{[0]} + 2p_H - f_H}{s_H} + 1 \right\rfloor \qquad n_W^{[1]} = \left\lfloor \frac{n_W^{[0]} + 2p_W - f_W}{s_W} + 1 \right\rfloor$$

The same, just in 3D

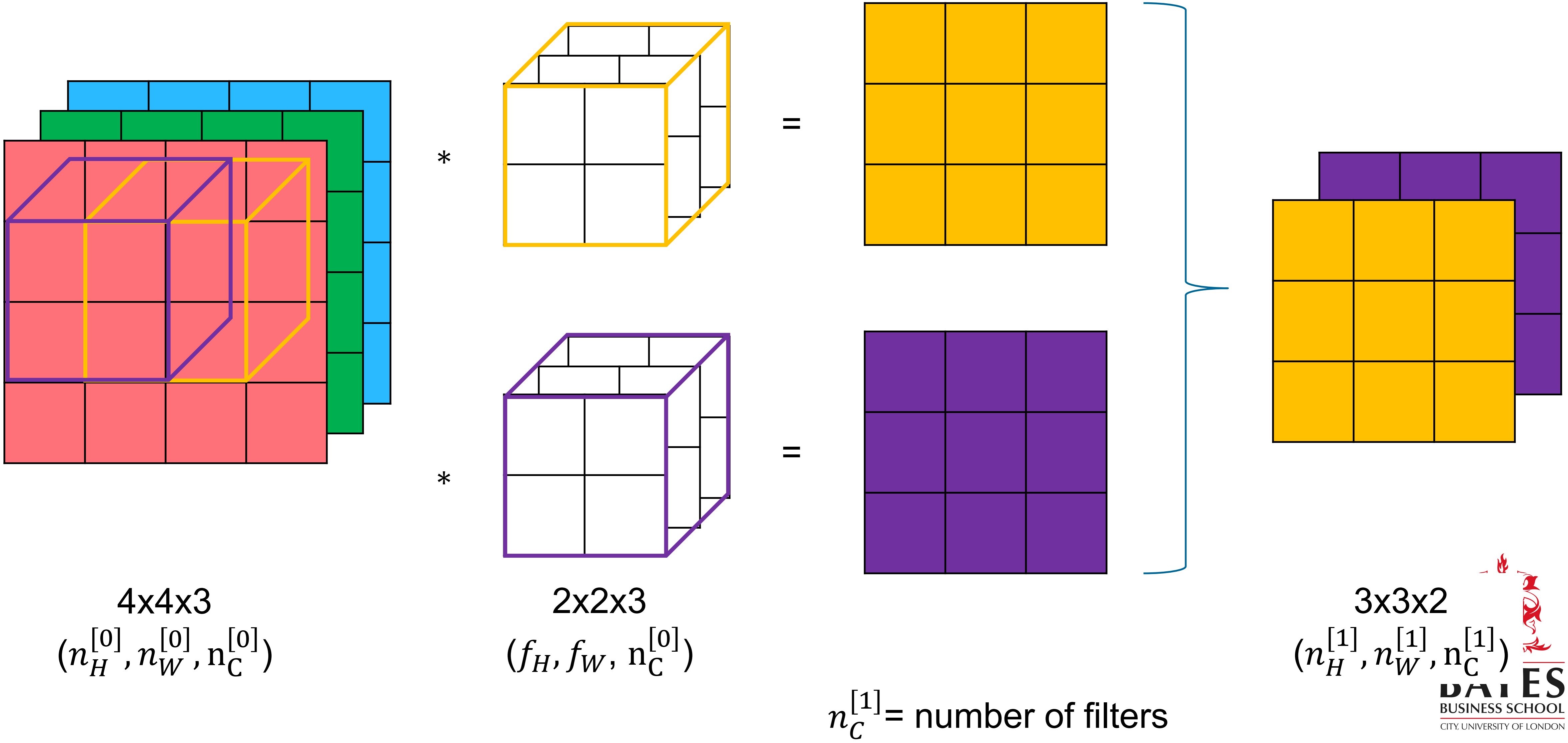
Convolution on a 3D array



Convolution on a 3D array



Multiple 3D convolutions

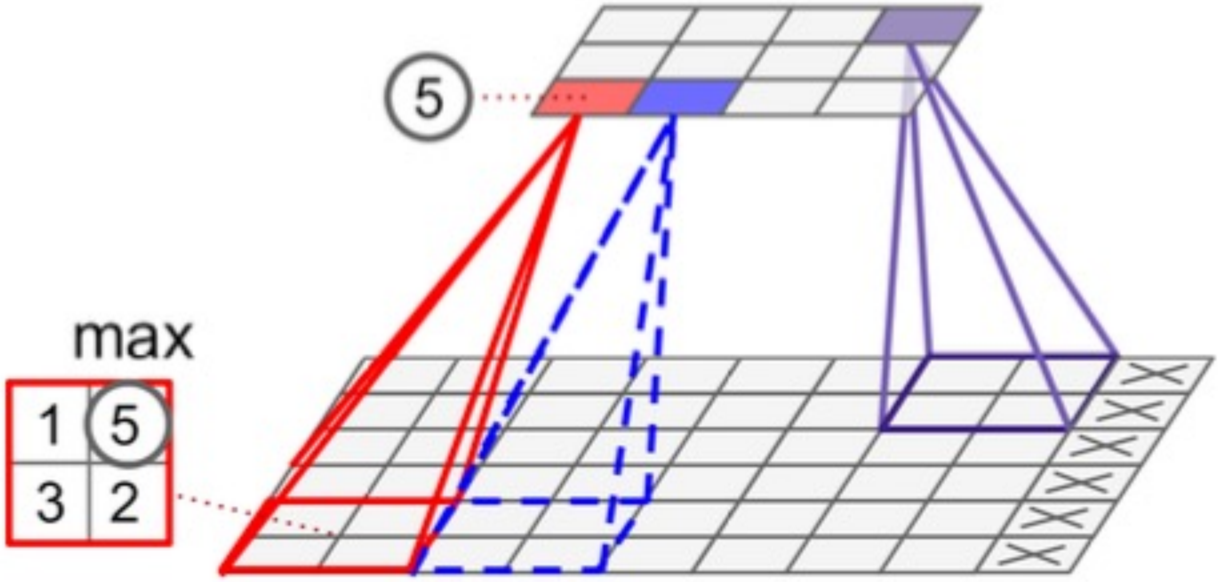


Additional architectural considerations

Objective of pooling layers

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

Max pooling



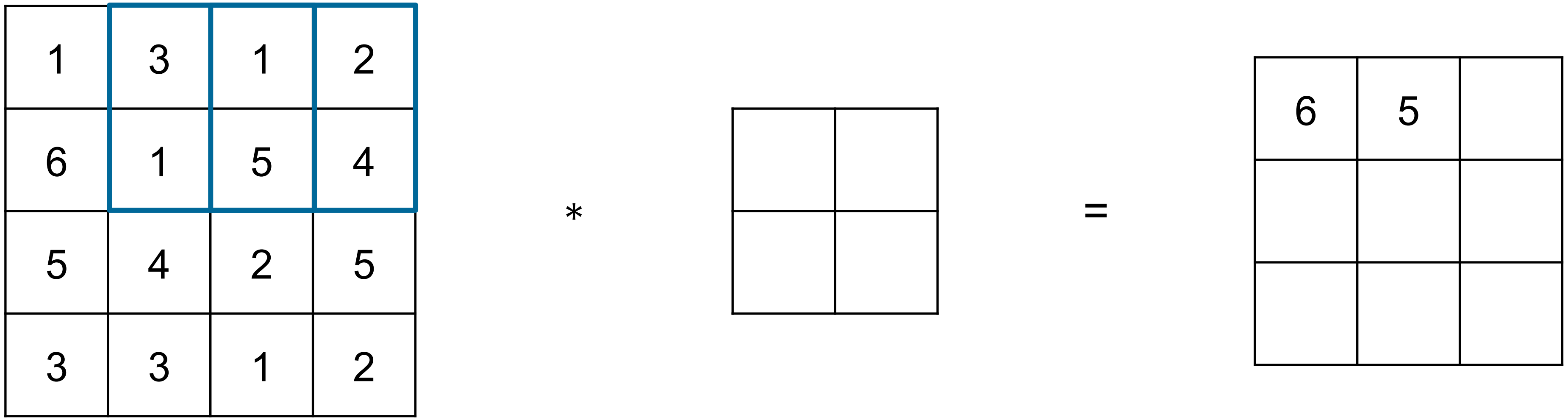
Source: Géron

Max pooling in practice

$$\begin{bmatrix} 1 & 3 & 1 & 2 \\ 6 & 1 & 5 & 4 \\ 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 2 \end{bmatrix} * \begin{bmatrix} & \\ & \end{bmatrix} = \begin{bmatrix} 6 & & \\ & & \\ & & \end{bmatrix}$$

Source: Géron

Max pooling in practice



Source: Géron

Max pooling in practice

$$\begin{bmatrix}1&3&1&2\\6&1&5&4\\5&4&2&5\\3&3&1&2\end{bmatrix} * \begin{bmatrix} & \\ & \end{bmatrix} = \begin{bmatrix}6&5&5\\ & & \\ & & \end{bmatrix}$$

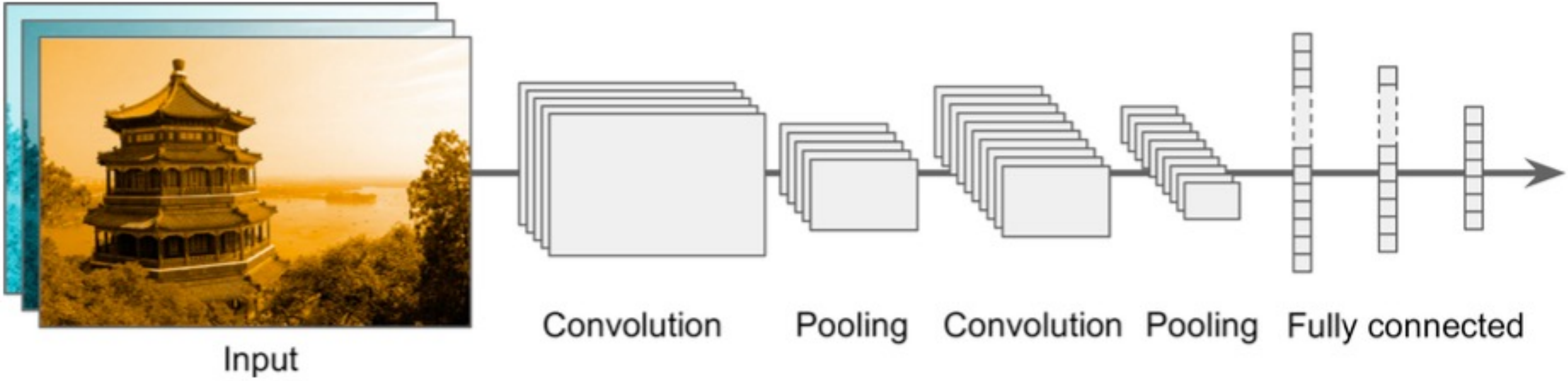
Source: Géron

Max pooling in practice

$$\begin{bmatrix} 1 & 3 & 1 & 2 \\ 6 & 1 & 5 & 4 \\ 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 2 \end{bmatrix} * \begin{bmatrix} \; & \; \\ \; & \; \end{bmatrix} = \begin{bmatrix} 6 & 5 & 5 \\ 6 & 5 & 5 \\ 5 & 4 & 5 \end{bmatrix}$$

Source: Géron

Typical architecture



Source: Géron

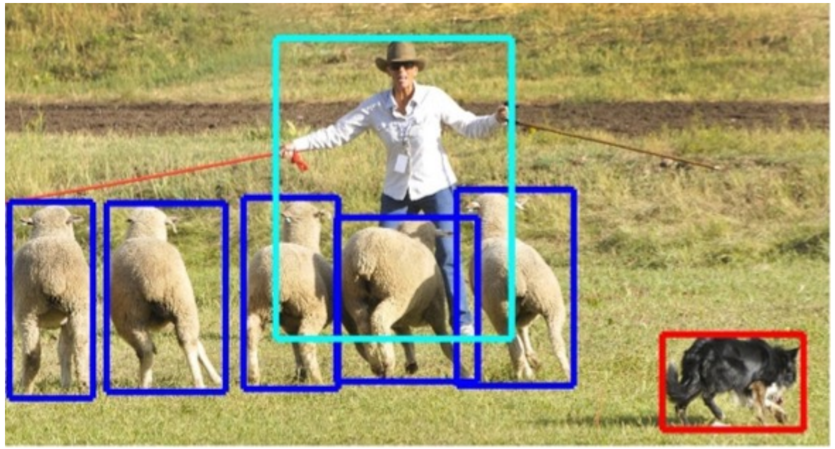
ConvNets in practice

Starting with object detection – non-sequential models

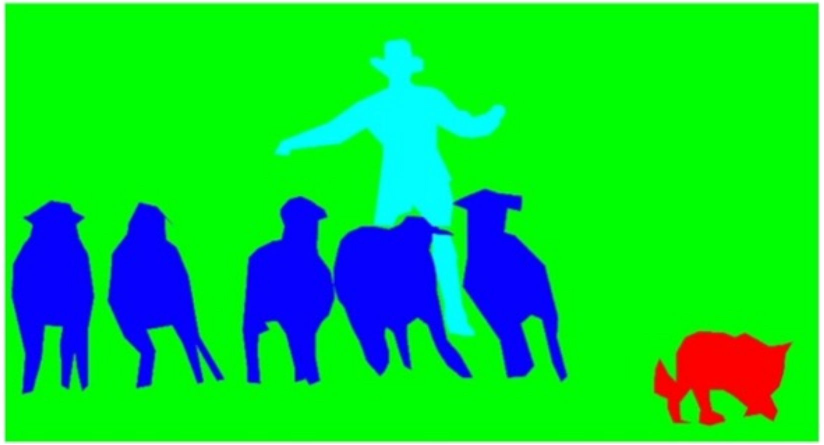
Typical computer vision problems

Image classification

Object detection



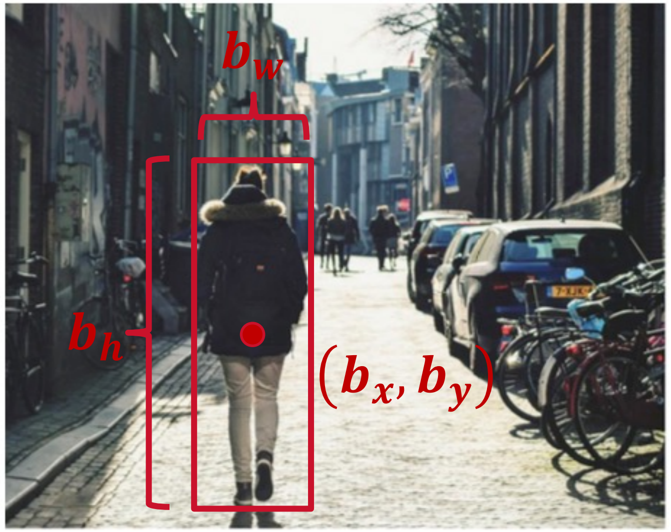
Semantic segmentation



Neural style transfer

Source: Lin, reiinakano.com

Before detection: classification + localization



A CNN

Softmax

$$\hat{y}_1 = \begin{cases} 1 - person \\ 2 - car \\ 3 - background \end{cases}$$

$$\hat{y}_2 = \begin{pmatrix} b_x \\ b_y \\ b_w \\ b_h \end{pmatrix}$$

Multi-Regression

Source: Géron

A look at creating non-sequential models

See you next week!

Sources

- Alla & Kalyan Adari, 2019, Beginning Anomaly Detection Using Python-Based Deep Learning, Apress
- 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