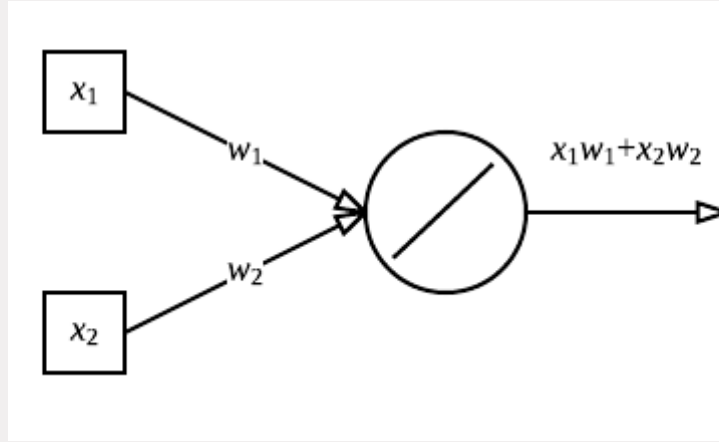# Convolutional Neural Networks (8DC00)
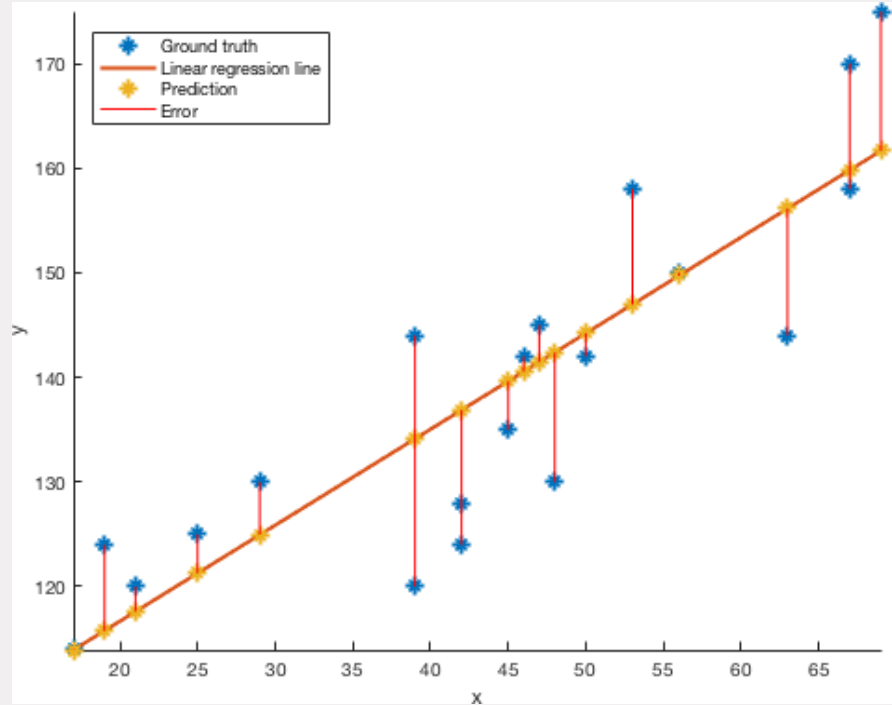
Cian Scannell (slides from Friso Heslinga)
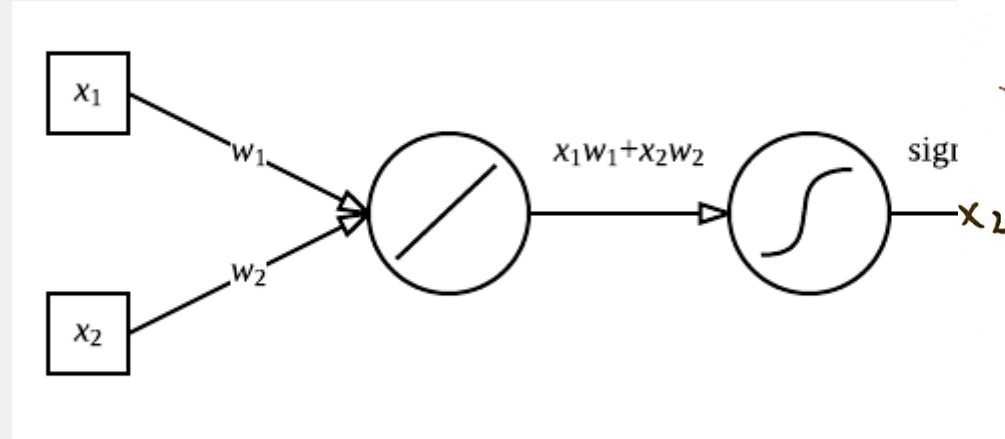
Previously – Linear regression

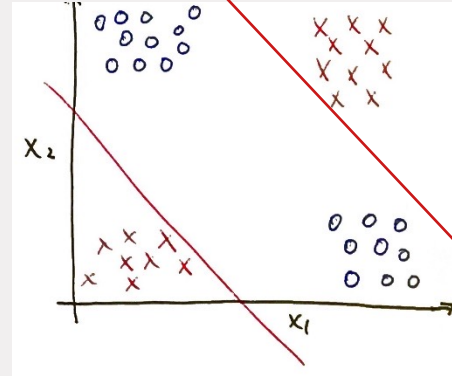$$\hat{y} = \boldsymbol{\theta}^{\mathsf{T}} \mathbf{x}$$
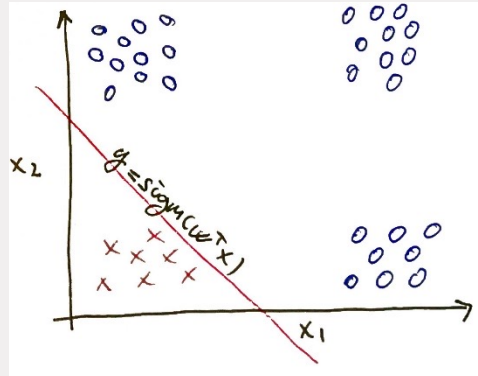
TU/e

Previously – Logistic regression

$$p(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\theta}^{\mathsf{T}} \mathbf{x})$$

TU/e

Previously – Neural networks

TU/e

Previously – Neural networks

input layer   **layer 1**   **layer 2**   **output layer**

TU/e
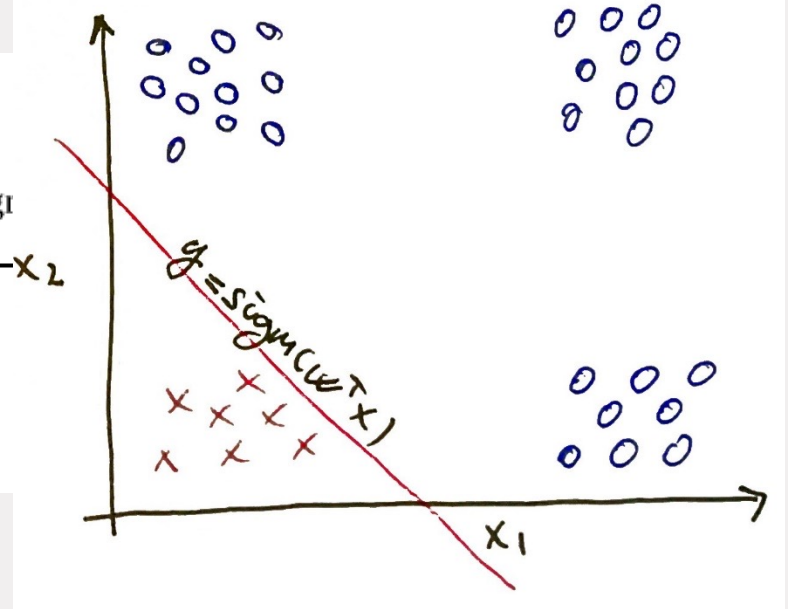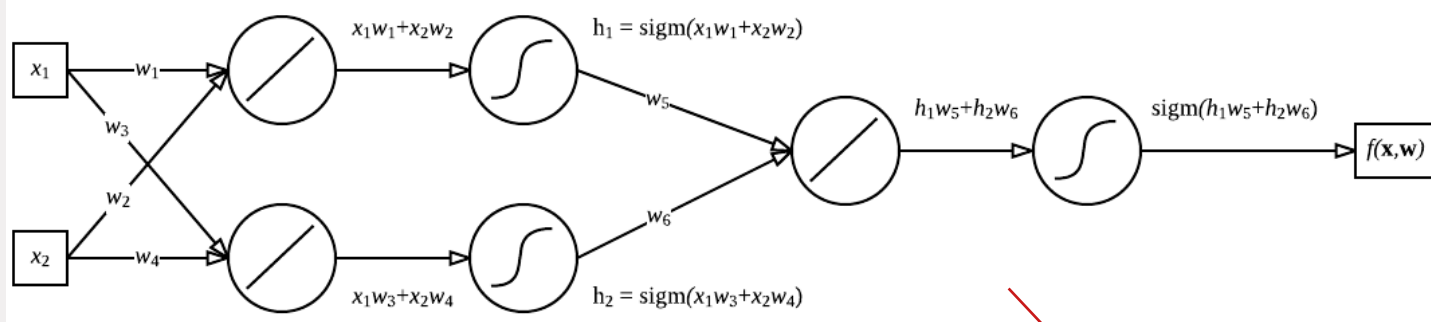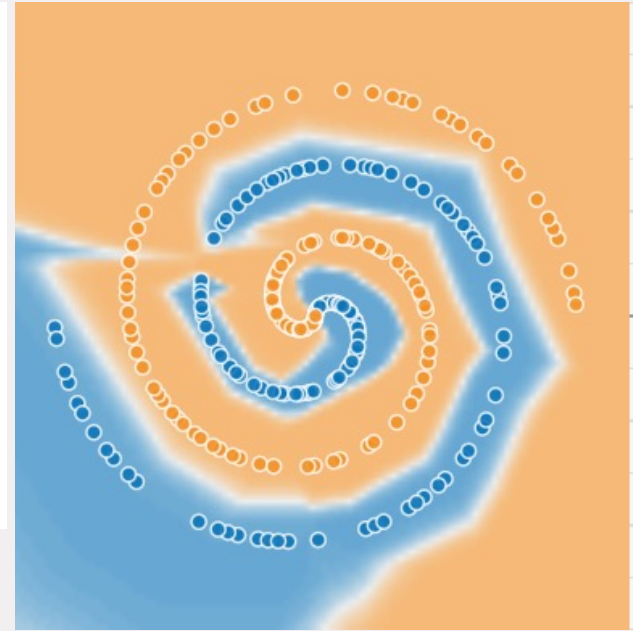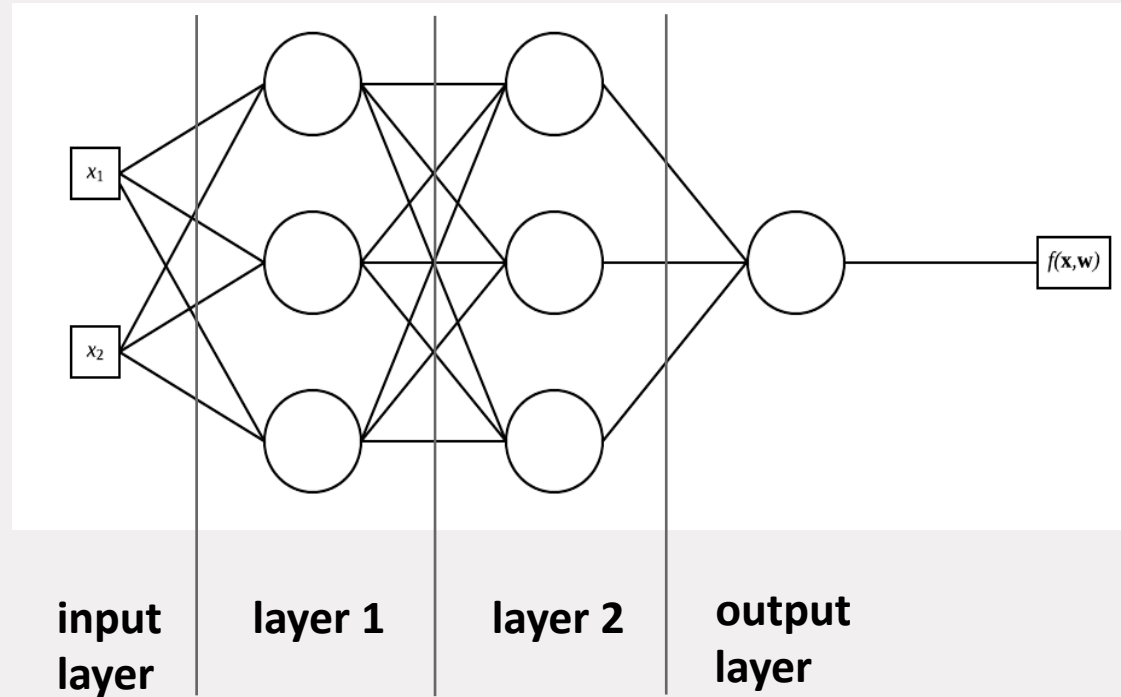
Previously – Machine learning
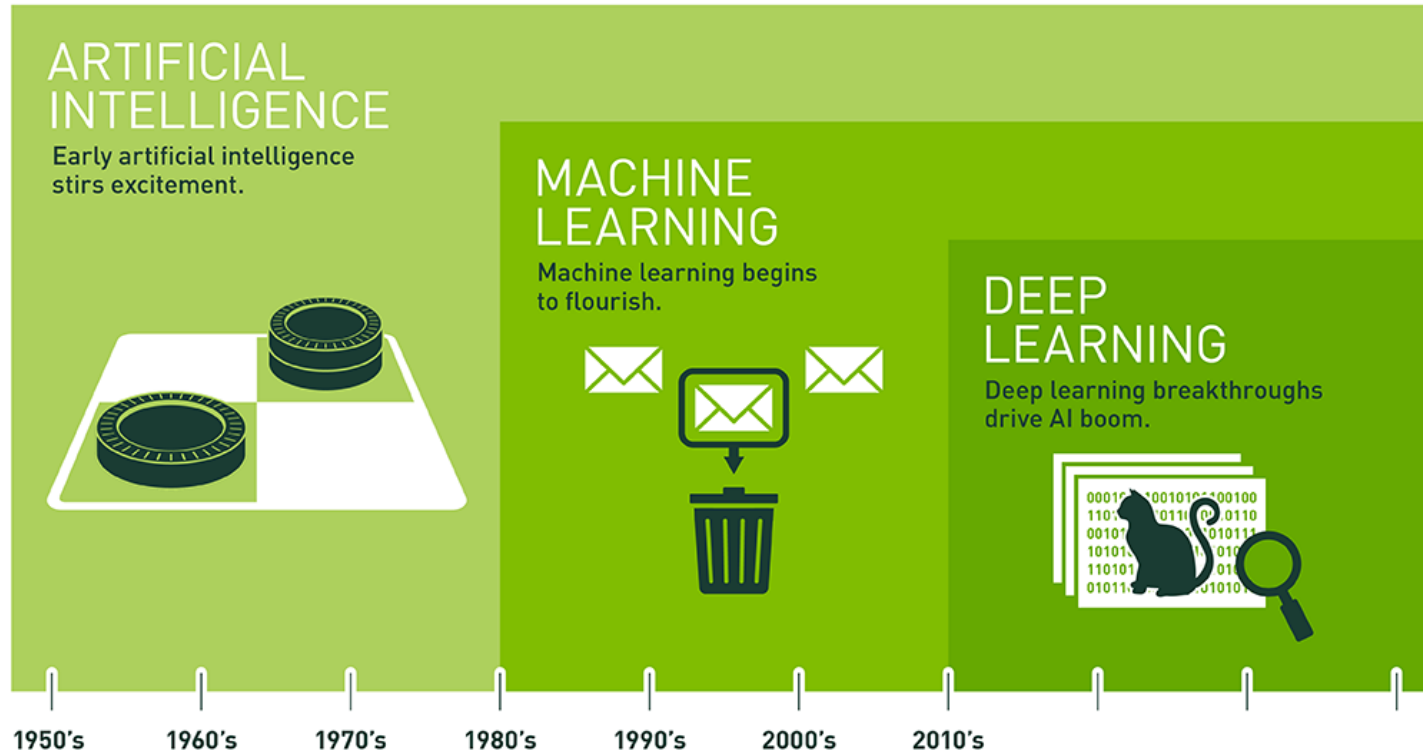
Today: Convolutional neural networks (CNN)

- Neural networks → Convolutional neural networks

Building blocks for deep learning models for image analysis:

- Convolutional layer

- Max-pooling layer

- Not needed for the project, but will be on the exam

TU/e

Learning outcomes

- Student can explain the concept of **convolutions** in a neural network

- Student can describe why we can use a **convolutional approach** for (medical) **images**

- Student can explain why convolutions enable development of **deep** (and large) **neural networks**

- Students can explain and apply the **max-pooling layer** in a convolutional neural network

- Students can motivate the choice for a **kernel size**

TU/e

Lecture outline

- Images as input to neural network

- Reducing # of weights

- 1D convolutions

- 2D convolutions

- Kernels

- Max-pooling

- Interactive example

8DC00 - Deep Learning Frameworks - Heslinga

TU/e

Images as input to neural networks

24 pixels

24 pixels

3 color channels
(red, green, blue)
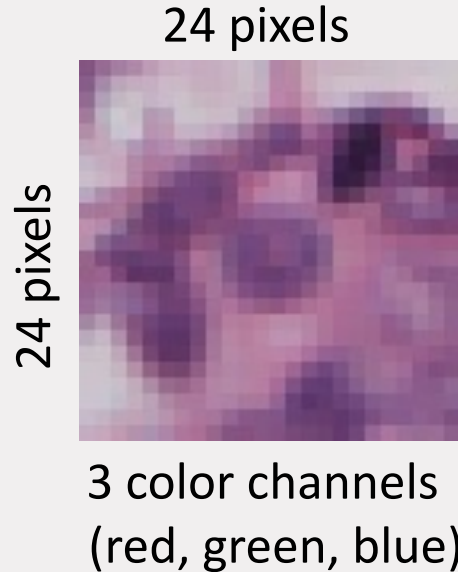
= 1728 features ($\mathbf{x}_i$ are 1728 dimensional vectors)

If we train linear regression with these inputs (such as in the first practical), we will have 1728 weights w and a bias b.

TU/e

Every pixel is an input

Every pixel from every color channel is multiplied by a weight

blue
green
red

$\theta^{T}x$

TU/e
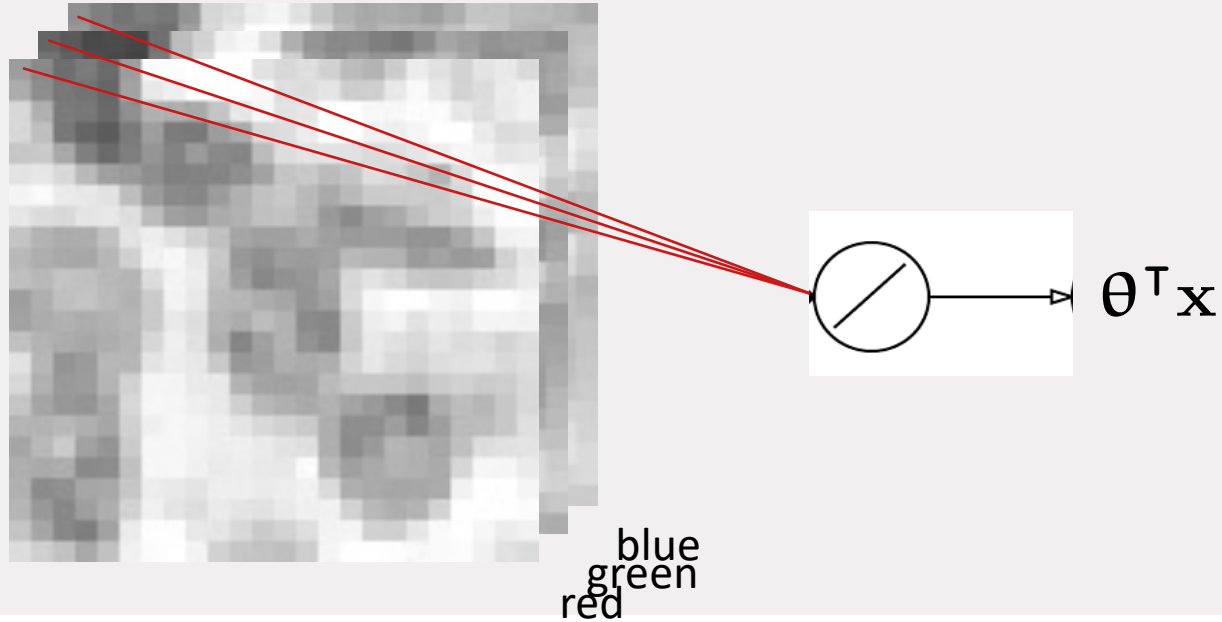
Post-training: visualizing what model has learned

Reshape vector of parameters into 24x24x3 image

24 pixels   24 pixels   24 pixels   24 pixels

24 pixels

3 color channels
(red, green, blue)

TU/e
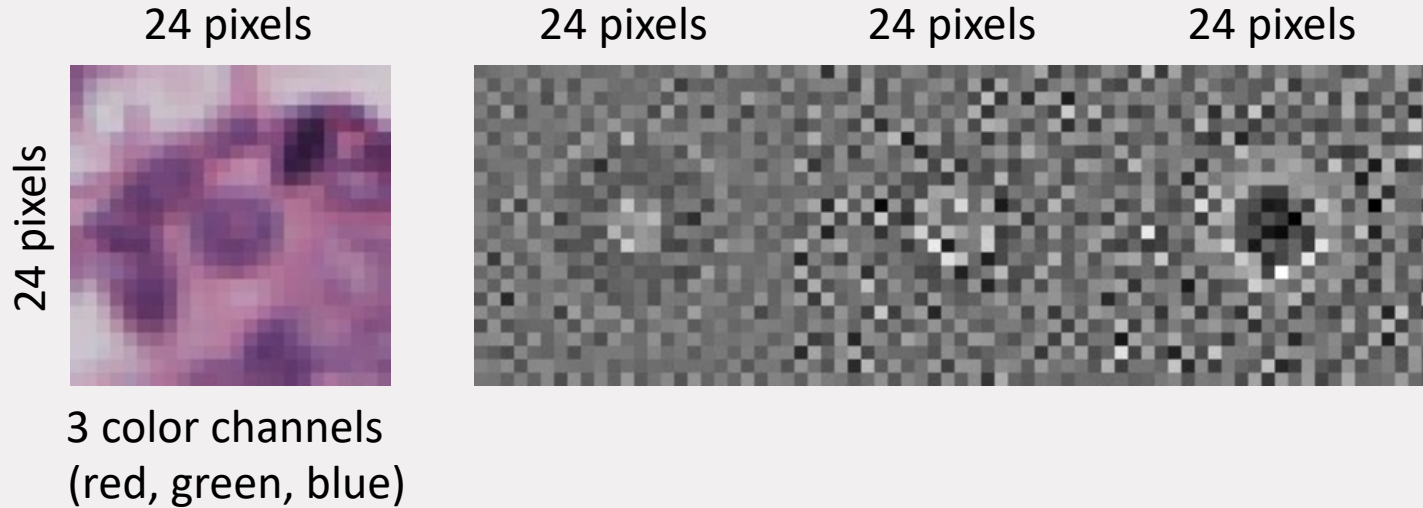
Post-training: visualizing what model has learned

Only 25% of training samples.. Looks noisy!

24 pixels

24 pixels

24 pixels

24 pixels

24 pixels

3 color channels
(red, green, blue)

8DC00 - Convolutional neural networks - Heslinga

TU/e
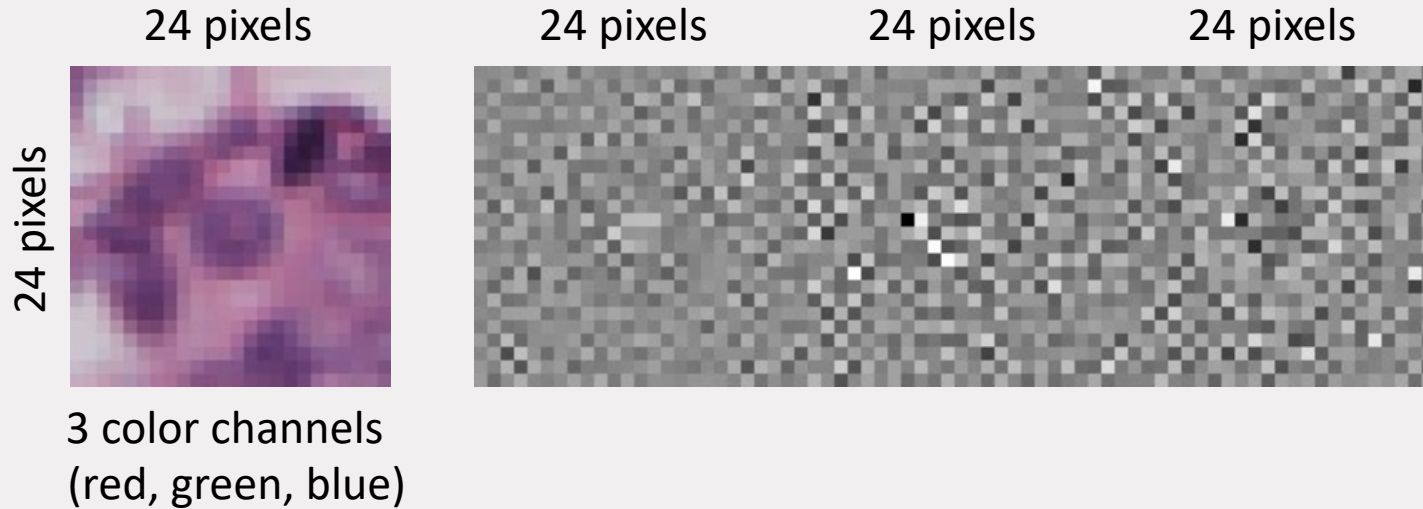
Post-training: visualizing what model has learned

You can think of it as "there is not enough training data to reliably estimate all model weights".

24 pixels 24 pixels 24 pixels 24 pixels

24 pixels

3 color channels
(red, green, blue)

TU/e

Many weights

input layer

hidden layer 1   hidden layer 2

output layer

width

TU/e
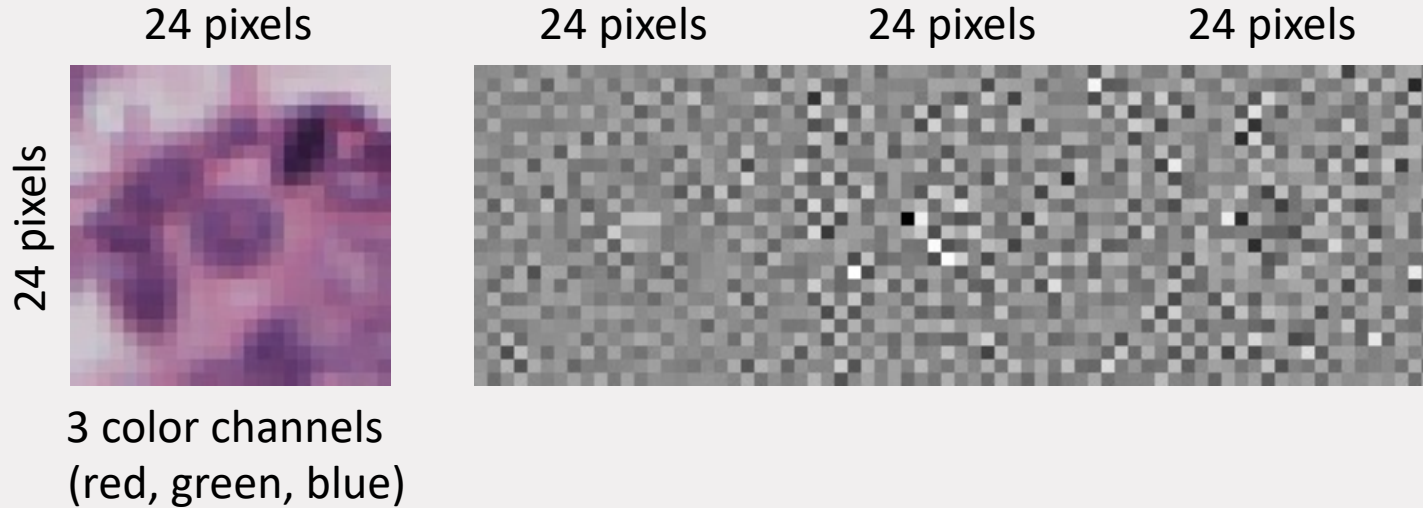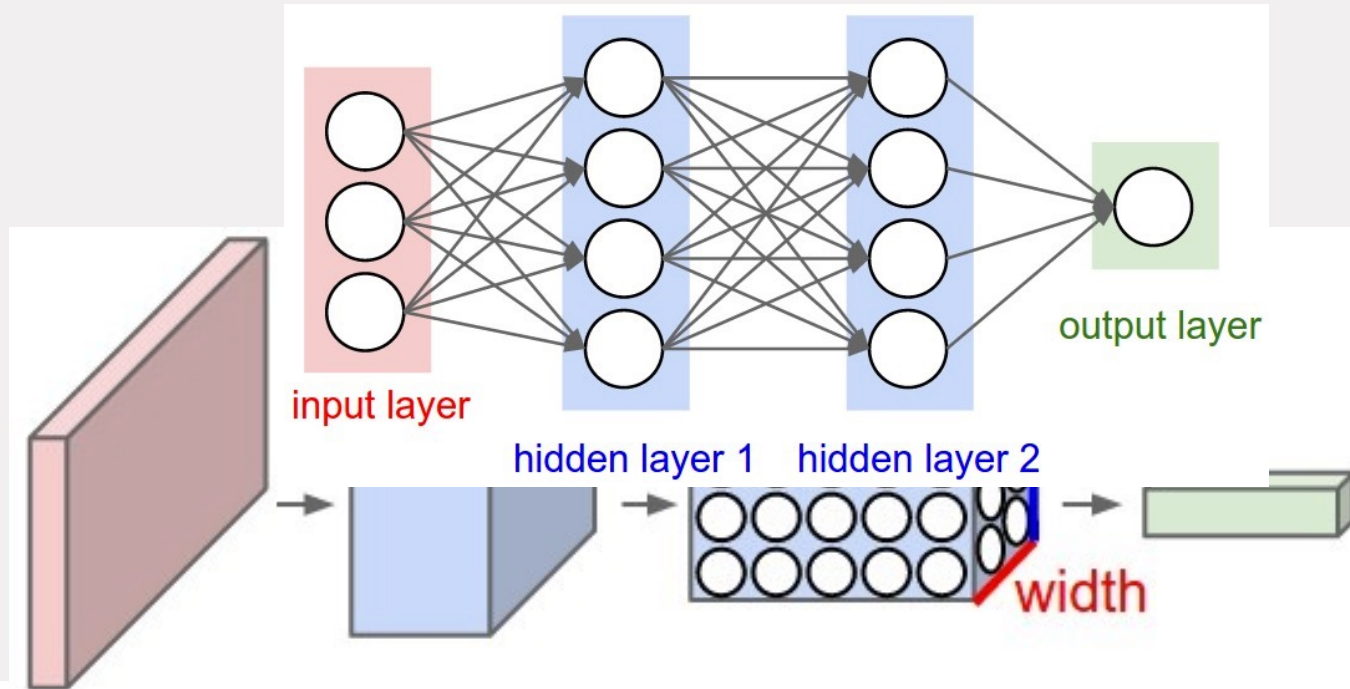
With large inputs such as images and deep networks, the number of weights "explodes". We need a way to reduce the number of weights, without sacrificing performance.

TU/e
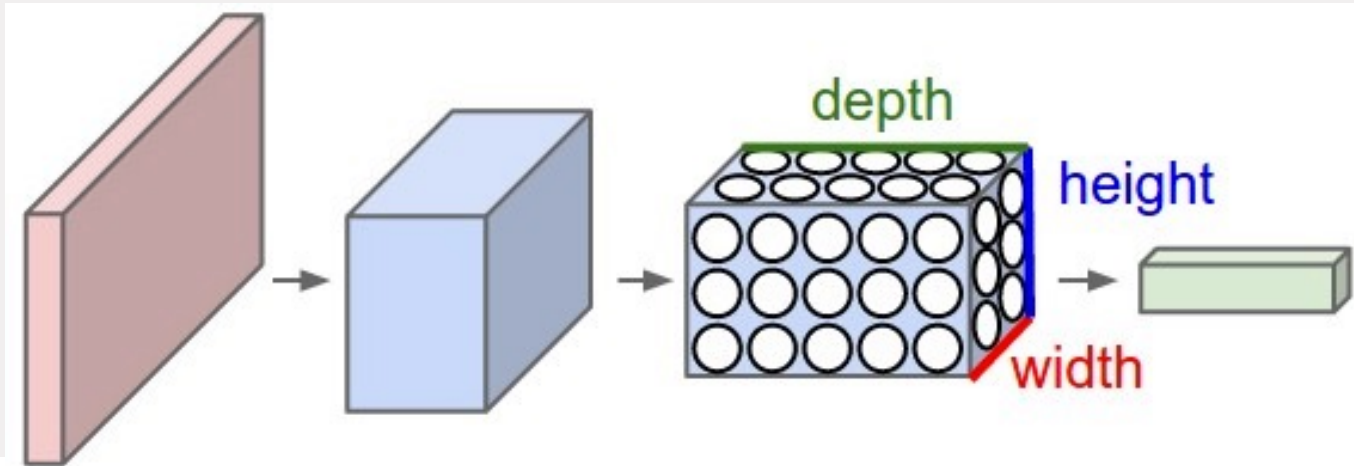
How many weights?
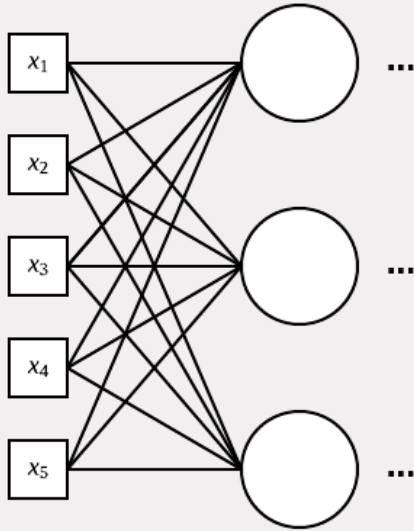
"regular" NN
15 weights
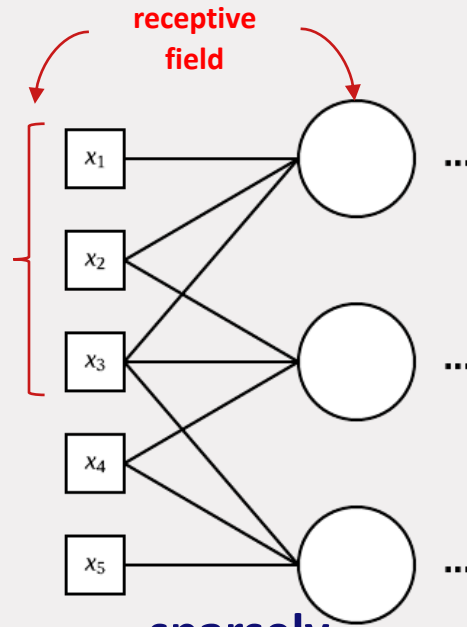
sparsely connected NN
9 weights

receptive field

shared weights
3 weights

8DC00 - Convolutional neural networks - Heslinga

TU/e

Shared weights

$= \text{sigm}(a_1)$

$= \text{sigm}(a_2)$

$= \text{sigm}(a_3)$
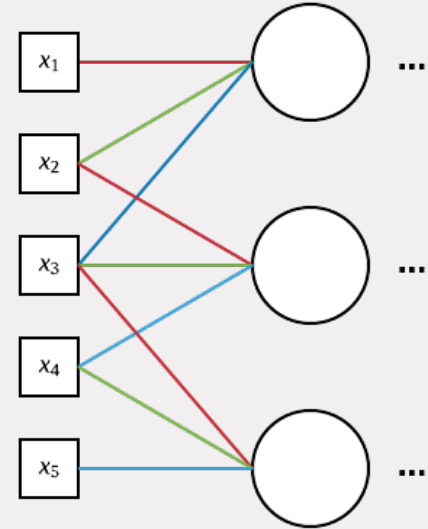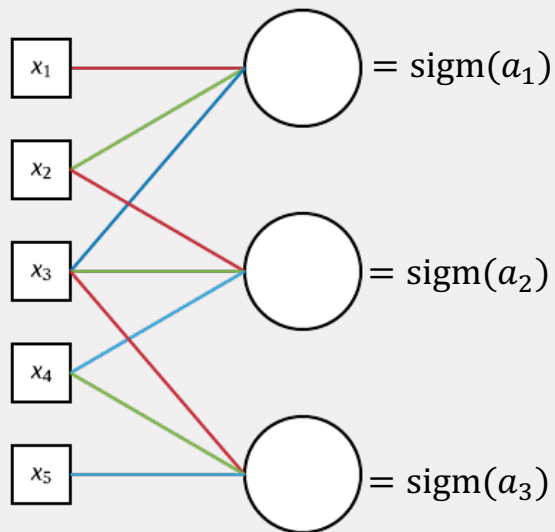
shared weights
3 weights

$$a_1 = x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$a_2 = x_2 w_1 + x_3 w_2 + x_4 w_3$$

$$a_3 = x_3 w_1 + x_4 w_2 + x_5 w_3$$

$$[a_1 \quad a_2 \quad a_3] =$$
$$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] * [w_1 \quad w_2 \quad w_3]$$

convolution, thus convolutional NN

TU/e

1-D Convolution

$$x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$x_2 w_1 + x_3 w_2 + x_4 w_3$$

$$x_3 w_1 + x_4 w_2 + x_5 w_3$$

8DC00 - Convolutional neural networks - Heslinga

TU/e

1-D Convolution

$x_1 w_1 + x_2 w_2 + x_3 w_3$

$x_2 w_1 + x_3 w_2 + x_4 w_3$

$x_3 w_1 + x_4 w_2 + x_5 w_3$
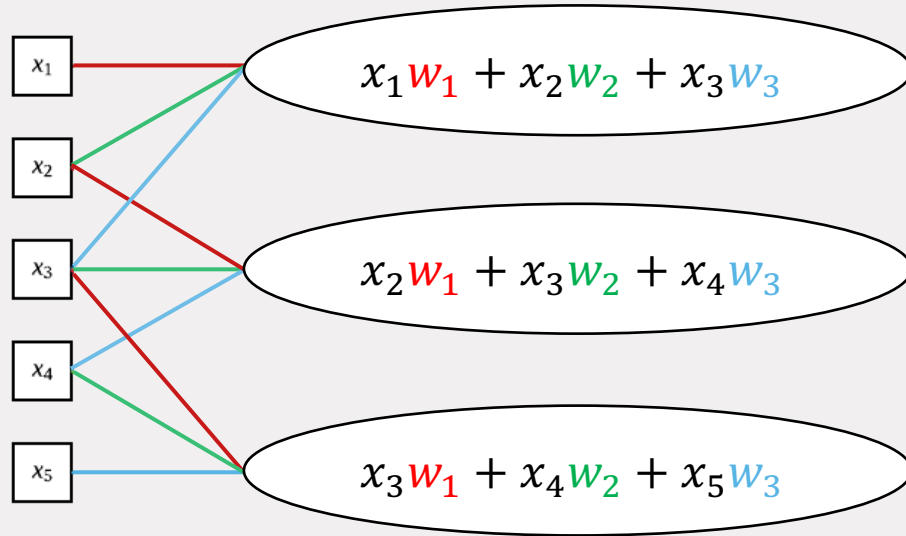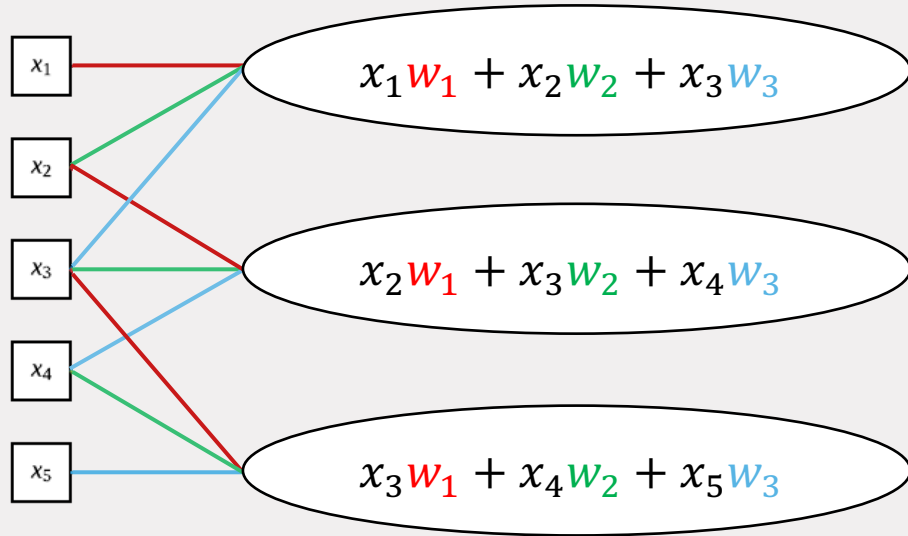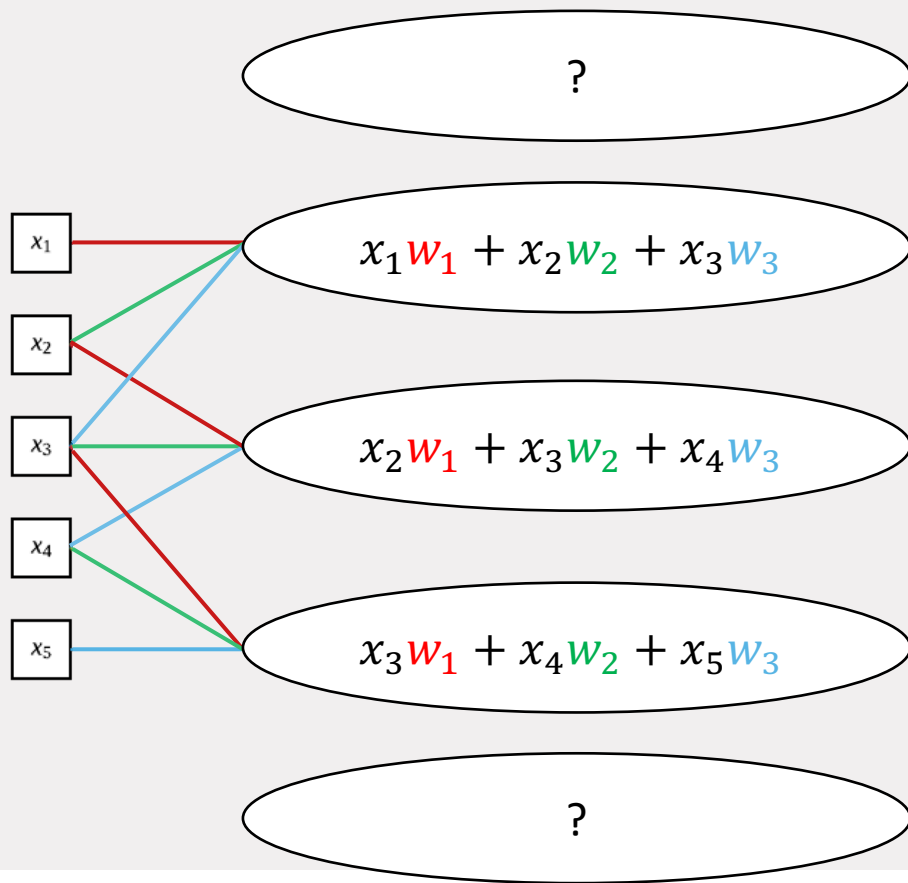
1-D Convolution

How can we keep the same number of features in hidden layer 1?

$$?$$

$$x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$x_2 w_1 + x_3 w_2 + x_4 w_3$$

$$x_3 w_1 + x_4 w_2 + x_5 w_3$$

$$?$$

x_1
x_2
x_3
x_4
x_5

TU/e

1-D Convolution

Zero-padding!

$$0w_1 + x_1w_2 + x_2w_3$$

$$x_1w_1 + x_2w_2 + x_3w_3$$

$$x_2w_1 + x_3w_2 + x_4w_3$$

$$x_3w_1 + x_4w_2 + x_5w_3$$

$$x_4w_1 + x_5w_2 + 0w_3$$

TU/e
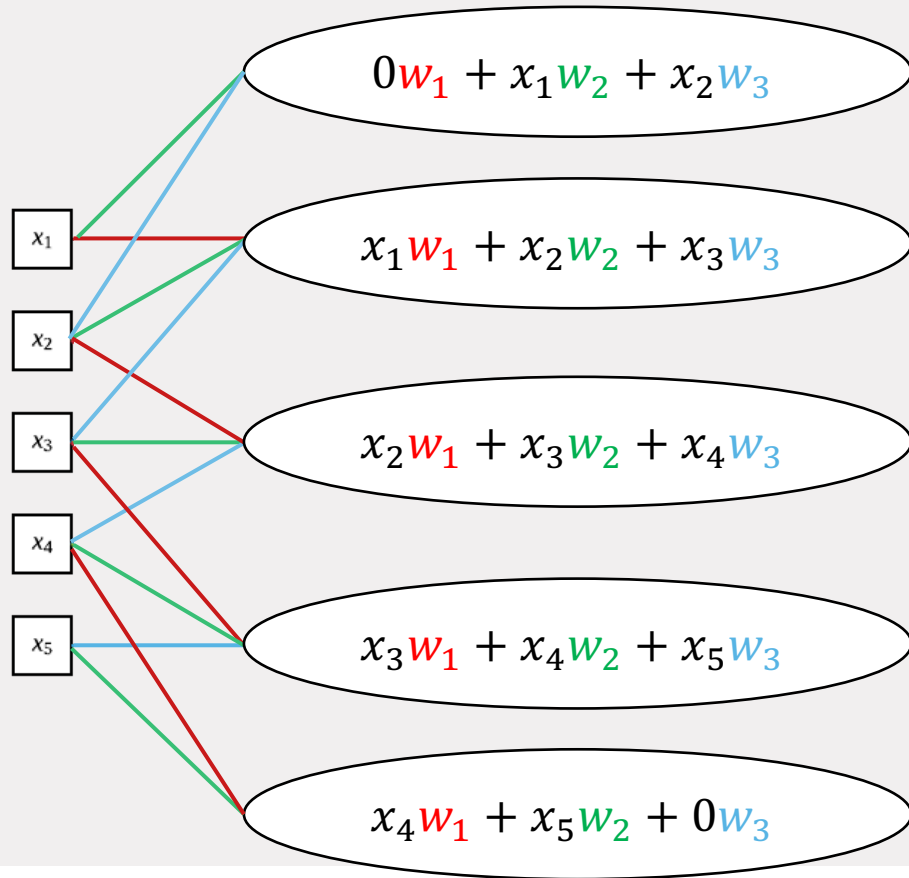
Properties of convolutional neural networks

- Sparse connectivity
- Weight sharing
- Parallel computations
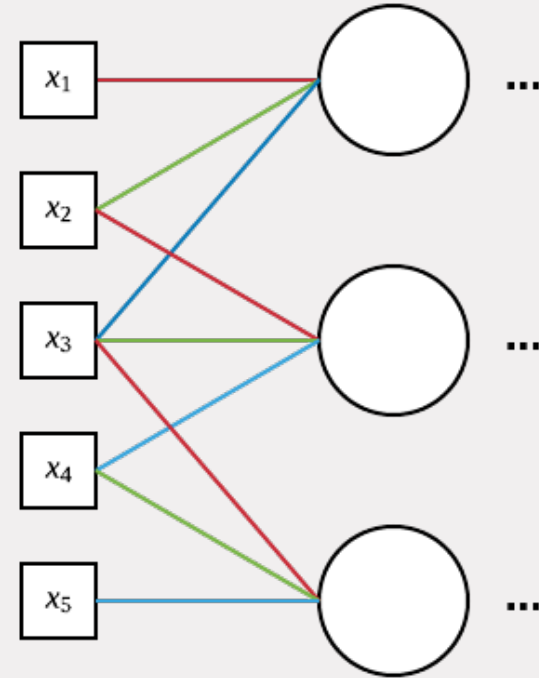
Why does this work?

- Multiple layers
- Hidden layers contain features calculated from previous layers

Neural network diagram: $x_1w_1+x_2w_2$, $h_1 = \text{sigm}(x_1w_1+x_2w_2)$, w_5, $h_1w_5+h_2w_6$, $\text{sigm}(h_1w_5+h_2w_6)$, $f(\mathbf{x},\mathbf{w})$, $x_1w_3+x_2w_4$, $h_2 = \text{sigm}(x_1w_3+x_2w_4)$, w_6
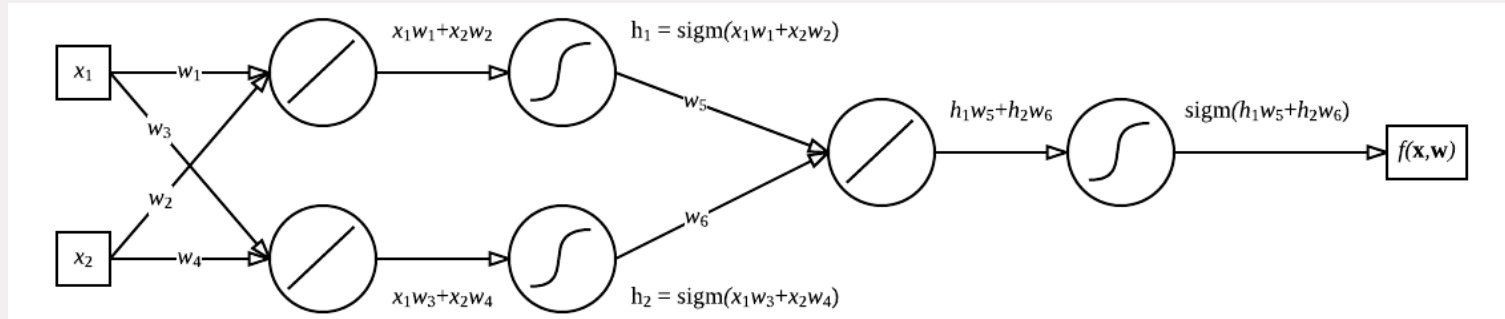
Why does this work?

- Different layers contain different transformation
    - Simple (e.g. edges, colors)
    - Complex (final layers)

One added benefit is that the learned transformations will be equivariant with translation (if the features/image is shifted up/down the features will still be detected).

TU/e
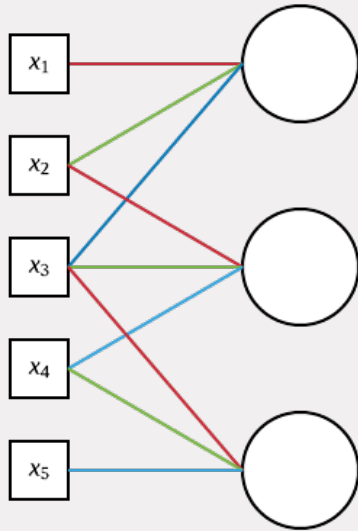
shared weights
3 weights
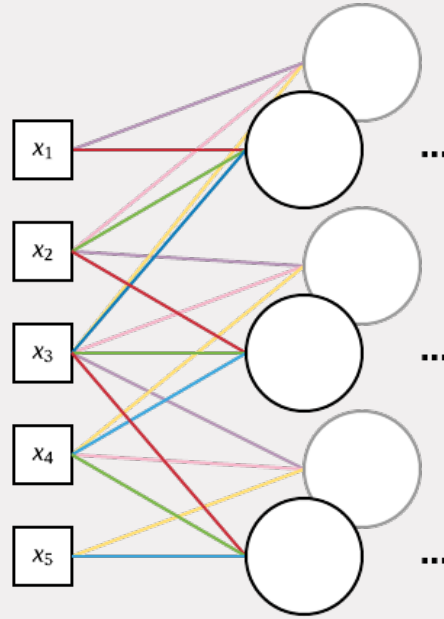
two sets **shared weights**
6 weights

We can add additional sets of weights that can learn additional interesting transformation of the input.

Note that the added neurons are not a new layer. They are part of layer 1.
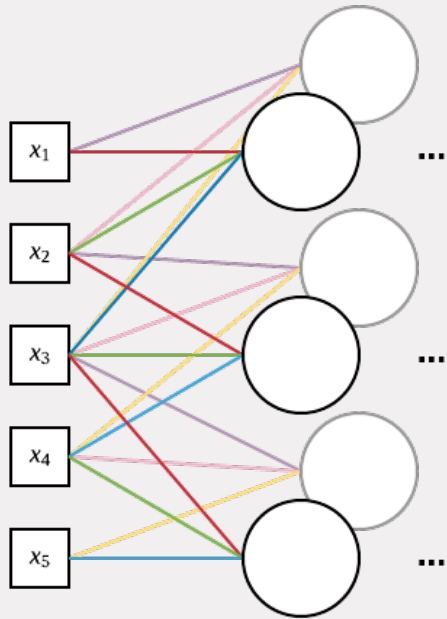
TU/e

$$[a_{1,1} \quad a_{1,2} \quad a_{1,3}] =$$
$$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] * [w_{1,1} \quad w_{1,2} \quad w_{1,3}]$$

$$[a_{2,1} \quad a_{2,2} \quad a_{2,3}] =$$
$$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] * [w_{2,1} \quad w_{2,2} \quad w_{2,3}]$$

two sets shared weights
6 weights

$[w_{1,1} \quad w_{1,2} \quad w_{1,3}]$, and $[w_{2,1} \quad w_{2,2} \quad w_{2,3}]$ are **convolution kernels**. They extract features. However, they are not hand-designed features – they were learned by the neural network.

TU/e

Convolutional neural networks are ideal for images

1D NN ⟷ **2D NN (images)**
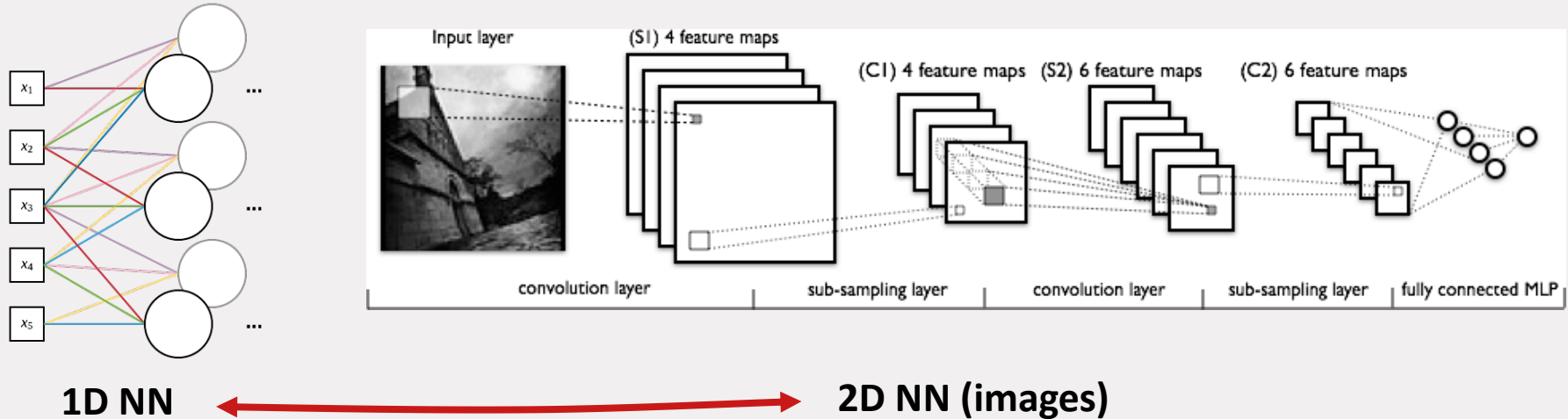
Because of the weight sharing, convolutional neural networks only work with structured data (such as images) as inputs.

$$I_1 \quad I_2 \quad I_3 \quad I_4 \quad I_5$$

$$I_6 \quad I_7 \quad I_8 \quad I_9 \quad I_{10}$$

$$I_{11} \quad I_{12} \quad I_{13} \quad I_{14} \quad I_{15} \qquad * \qquad W_1 \quad W_2 \quad W_3$$

$$I_{16} \quad I_{17} \quad I_{18} \quad I_{19} \quad I_{20} \qquad\qquad W_4 \quad W_5 \quad W_6 \qquad =$$

$$I_{21} \quad I_{22} \quad I_{23} \quad I_{24} \quad I_{25} \qquad\qquad W_7 \quad W_8 \quad W_9$$

$$\begin{array}{c} I_1 w_1 + I_2 w_2 + I_3 w_3 \\ + \\ I_6 w_4 + I_7 w_5 + I_8 w_6 \\ + \\ I_{11} w_7 + I_{12} w_8 + I_{13} w_9 \end{array}$$

TU/e

$$
\begin{array}{|c|c|c|c|c|}
\hline I_1 & I_2 & I_3 & I_4 & I_5 \\
\hline I_6 & I_7 & I_8 & I_9 & I_{10} \\
\hline I_{11} & I_{12} & I_{13} & I_{14} & I_{15} \\
\hline I_{16} & I_{17} & I_{18} & I_{19} & I_{20} \\
\hline I_{21} & I_{22} & I_{23} & I_{24} & I_{25} \\
\hline
\end{array}
\quad * \quad
\begin{array}{|c|c|c|}
\hline w_1 & w_2 & w_3 \\
\hline w_4 & w_5 & w_6 \\
\hline w_7 & w_8 & w_9 \\
\hline
\end{array}
\quad =
$$

$$
\begin{array}{|c|c|c|}
\hline
\begin{matrix} I_1 w_1 + I_2 w_2 + I_3 w_3 \\ + \\ I_6 w_4 + I_7 w_5 + I_8 w_6 \\ + \\ I_{11} w_7 + I_{12} w_8 + I_{13} w_9 \end{matrix}
&
\begin{matrix} I_2 w_1 + I_3 w_2 + I_4 w_3 \\ + \\ I_7 w_4 + I_8 w_5 + I_9 w_6 \\ + \\ I_{12} w_7 + I_{13} w_8 + I_{14} w_9 \end{matrix}
& \\
\hline
& & \\
\hline
& & \\
\hline
\end{array}
$$

TU/e

$$I_1 \quad I_2 \quad \boxed{I_3 \quad I_4 \quad I_5}$$
$$I_6 \quad I_7 \quad \boxed{I_8 \quad I_9 \quad I_{10}}$$
$$I_{11} \quad I_{12} \quad \boxed{I_{13} \quad I_{14} \quad I_{15}}$$
$$I_{16} \quad I_{17} \quad I_{18} \quad I_{19} \quad I_{20}$$
$$I_{21} \quad I_{22} \quad I_{23} \quad I_{24} \quad I_{25}$$

*

$$W_1 \quad W_2 \quad W_3$$
$$W_4 \quad W_5 \quad W_6$$
$$W_7 \quad W_8 \quad W_9$$

=

$I_1w_1+I_2w_2+I_3w_3$ $+$ $I_6w_4+I_7w_5+I_8w_6$ $+$ $I_{11}w_7+I_{12}w_8+I_{13}w_9$	$I_2w_1+I_3w_2+I_4w_3$ $+$ $I_7w_4+I_8w_5+I_9w_6$ $+$ $I_{12}w_7+I_{13}w_8+I_{14}w_9$	$I_3w_1+I_4w_2+I_5w_3$ $+$ $I_8w_4+I_9w_5+I_{10}w_6$ $+$ $I_{13}w_7+I_{14}w_8+I_{15}w_9$

TU/e

TU/e

Input matrix:

I_1	I_2	I_3	I_4	I_5
I_6	I_7	I_8	I_9	I_{10}
I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}

$*$

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

$=$

$I_1 w_1 + I_2 w_2 + I_3 w_3$ $+$ $I_6 w_4 + I_7 w_5 + I_8 w_6$ $+$ $I_{11} w_7 + I_{12} w_8 + I_{13} w_9$	$I_2 w_1 + I_3 w_2 + I_4 w_3$ $+$ $I_7 w_4 + I_8 w_5 + I_9 w_6$ $+$ $I_{12} w_7 + I_{13} w_8 + I_{14} w_9$	$I_3 w_1 + I_4 w_2 + I_5 w_3$ $+$ $I_8 w_4 + I_9 w_5 + I_{10} w_6$ $+$ $I_{13} w_7 + I_{14} w_8 + I_{15} w_9$
$I_6 w_1 + I_7 w_2 + I_8 w_3$ $+$ $I_{11} w_4 + I_{12} w_5 + I_{13} w_6$ $+$ $I_{16} w_7 + I_{17} w_8 + I_{18} w_9$	$I_7 w_1 + I_8 w_2 + I_9 w_3$ $+$ $I_{12} w_4 + I_{13} w_5 + I_{14} w_6$ $+$ $I_{17} w_7 + I_{18} w_8 + I_{19} w_9$	

TU/e

$$I_1 \quad I_2 \quad I_3 \quad I_4 \quad I_5$$
$$I_6 \quad I_7 \quad I_8 \quad I_9 \quad I_{10}$$
$$I_{11} \quad I_{12} \quad I_{13} \quad I_{14} \quad I_{15}$$
$$I_{16} \quad I_{17} \quad I_{18} \quad I_{19} \quad I_{20}$$
$$I_{21} \quad I_{22} \quad I_{23} \quad I_{24} \quad I_{25}$$

$*$

$$w_1 \quad w_2 \quad w_3$$
$$w_4 \quad w_5 \quad w_6$$
$$w_7 \quad w_8 \quad w_9$$

$=$

$I_1w_1+I_2w_2+I_3w_3$ $+$ $I_6w_4+I_7w_5+I_8w_6$ $+$ $I_{11}w_7+I_{12}w_8+I_{13}w_9$	$I_2w_1+I_3w_2+I_4w_3$ $+$ $I_7w_4+I_8w_5+I_9w_6$ $+$ $I_{12}w_7+I_{13}w_8+I_{14}w_9$	$I_3w_1+I_4w_2+I_5w_3$ $+$ $I_8w_4+I_9w_5+I_{10}w_6$ $+$ $I_{13}w_7+I_{14}w_8+I_{15}w_9$
$I_6w_1+I_7w_2+I_8w_3$ $+$ $I_{11}w_4+I_{12}w_5+I_{13}w_6$ $+$ $I_{16}w_7+I_{17}w_8+I_{18}w_9$	$I_7w_1+I_8w_2+I_9w_3$ $+$ $I_{12}w_4+I_{13}w_5+I_{14}w_6$ $+$ $I_{17}w_7+I_{18}w_8+I_{19}w_9$	$I_8w_1+I_9w_2+I_{10}w_3$ $+$ $I_{13}w_4+I_{14}w_5+I_{15}w_6$ $+$ $I_{18}w_7+I_{19}w_8+I_{20}w_9$

TU/e

I_1 I_2 I_3 I_4 I_5

I_6 I_7 I_8 I_9 I_{10}

I_{11} I_{12} I_{13} I_{14} I_{15}

I_{16} I_{17} I_{18} I_{19} I_{20}

I_{21} I_{22} I_{23} I_{24} I_{25}

∗

w_1 w_2 w_3

w_4 w_5 w_6

w_7 w_8 w_9

=

$I_1w_1+I_2w_2+I_3w_3$
$+$
$I_6w_4+I_7w_5+I_8w_6$
$+$
$I_{11}w_7+I_{12}w_8+I_{13}w_9$

$I_2w_1+I_3w_2+I_4w_3$
$+$
$I_7w_4+I_8w_5+I_9w_6$
$+$
$I_{12}w_7+I_{13}w_8+I_{14}w_9$

$I_3w_1+I_4w_2+I_5w_3$
$+$
$I_8w_4+I_9w_5+I_{10}w_6$
$+$
$I_{13}w_7+I_{14}w_8+I_{15}w_9$

$I_6w_1+I_7w_2+I_8w_3$
$+$
$I_{11}w_4+I_{12}w_5+I_{13}w_6$
$+$
$I_{16}w_7+I_{17}w_8+I_{18}w_9$

$I_7w_1+I_8w_2+I_9w_3$
$+$
$I_{12}w_4+I_{13}w_5+I_{14}w_6$
$+$
$I_{17}w_7+I_{18}w_8+I_{19}w_9$

$I_8w_1+I_9w_2+I_{10}w_3$
$+$
$I_{13}w_4+I_{14}w_5+I_{15}w_6$
$+$
$I_{18}w_7+I_{19}w_8+I_{20}w_9$

$I_{11}w_1+I_{12}w_2+I_{13}w_3$
$+$
$I_{16}w_4+I_{17}w_5+I_{18}w_6$
$+$
$I_{21}w_7+I_{22}w_8+I_{23}w_9$

TU/e

$$I_1 \quad I_2 \quad I_3 \quad I_4 \quad I_5$$

$$I_6 \quad I_7 \quad I_8 \quad I_9 \quad I_{10}$$

$$I_{11} \quad I_{12} \quad I_{13} \quad I_{14} \quad I_{15}$$

$$I_{16} \quad I_{17} \quad I_{18} \quad I_{19} \quad I_{20}$$

$$I_{21} \quad I_{22} \quad I_{23} \quad I_{24} \quad I_{25}$$

$$*$$

$$w_1 \quad w_2 \quad w_3$$

$$w_4 \quad w_5 \quad w_6$$

$$w_7 \quad w_8 \quad w_9$$

$$=$$

$I_1w_1+I_2w_2+I_3w_3$ $+$ $I_6w_4+I_7w_5+I_8w_6$ $+$ $I_{11}w_7+I_{12}w_8+I_{13}w_9$	$I_2w_1+I_3w_2+I_4w_3$ $+$ $I_7w_4+I_8w_5+I_9w_6$ $+$ $I_{12}w_7+I_{13}w_8+I_{14}w_9$	$I_3w_1+I_4w_2+I_5w_3$ $+$ $I_8w_4+I_9w_5+I_{10}w_6$ $+$ $I_{13}w_7+I_{14}w_8+I_{15}w_9$
$I_6w_1+I_7w_2+I_8w_3$ $+$ $I_{11}w_4+I_{12}w_5+I_{13}w_6$ $+$ $I_{16}w_7+I_{17}w_8+I_{18}w_9$	$I_7w_1+I_8w_2+I_9w_3$ $+$ $I_{12}w_4+I_{13}w_5+I_{14}w_6$ $+$ $I_{17}w_7+I_{18}w_8+I_{19}w_9$	$I_8w_1+I_9w_2+I_{10}w_3$ $+$ $I_{13}w_4+I_{14}w_5+I_{15}w_6$ $+$ $I_{18}w_7+I_{19}w_8+I_{20}w_9$
$I_{11}w_1+I_{12}w_2+I_{13}w_3$ $+$ $I_{16}w_4+I_{17}w_5+I_{18}w_6$ $+$ $I_{21}w_7+I_{22}w_8+I_{23}w_9$	$I_{12}w_1+I_{13}w_2+I_{14}w_3$ $+$ $I_{17}w_4+I_{18}w_5+I_{19}w_6$ $+$ $I_{22}w_7+I_{23}w_8+I_{24}w_9$	

TU/e

$$I_1 \quad I_2 \quad I_3 \quad I_4 \quad I_5$$
$$I_6 \quad I_7 \quad I_8 \quad I_9 \quad I_{10}$$
$$I_{11} \quad I_{12} \quad I_{13} \quad I_{14} \quad I_{15}$$
$$I_{16} \quad I_{17} \quad I_{18} \quad I_{19} \quad I_{20}$$
$$I_{21} \quad I_{22} \quad I_{23} \quad I_{24} \quad I_{25}$$

$$* \quad
\begin{matrix}
w_1 & w_2 & w_3 \\
w_4 & w_5 & w_6 \\
w_7 & w_8 & w_9
\end{matrix}
\quad + b =$$

$$I_1 w_1 + I_2 w_2 + I_3 w_3 + I_6 w_4 + I_7 w_5 + I_8 w_6 + I_{11} w_7 + I_{12} w_8 + I_{13} w_9$$
$$I_2 w_1 + I_3 w_2 + I_4 w_3 + I_7 w_4 + I_8 w_5 + I_9 w_6 + I_{12} w_7 + I_{13} w_8 + I_{14} w_9$$
$$I_3 w_1 + I_4 w_2 + I_5 w_3 + I_8 w_4 + I_9 w_5 + I_{10} w_6 + I_{13} w_7 + I_{14} w_8 + I_{15} w_9$$

$$I_6 w_1 + I_7 w_2 + I_8 w_3 + I_{11} w_4 + I_{12} w_5 + I_{13} w_6 + I_{16} w_7 + I_{17} w_8 + I_{18} w_9$$
$$I_7 w_1 + I_8 w_2 + I_9 w_3 + I_{12} w_4 + I_{13} w_5 + I_{14} w_6 + I_{17} w_7 + I_{18} w_8 + I_{19} w_9$$
$$I_8 w_1 + I_9 w_2 + I_{10} w_3 + I_{13} w_4 + I_{14} w_5 + I_{15} w_6 + I_{18} w_7 + I_{19} w_8 + I_{20} w_9$$

$$+ b$$

$$I_{11} w_1 + I_{12} w_2 + I_{13} w_3 + I_{16} w_4 + I_{17} w_5 + I_{18} w_6 + I_{21} w_7 + I_{22} w_8 + I_{23} w_9$$
$$I_{12} w_1 + I_{13} w_2 + I_{14} w_3 + I_{17} w_4 + I_{18} w_5 + I_{19} w_6 + I_{22} w_7 + I_{23} w_8 + I_{24} w_9$$
$$I_{13} w_1 + I_{14} w_2 + I_{15} w_3 + I_{18} w_4 + I_{19} w_5 + I_{20} w_6 + I_{23} w_7 + I_{24} w_8 + I_{25} w_9$$

TU/e

Convolutions + non-linearity

$= a_1$

$= a_2$

$= a_3$

x_1
x_2
x_3
x_4
x_5
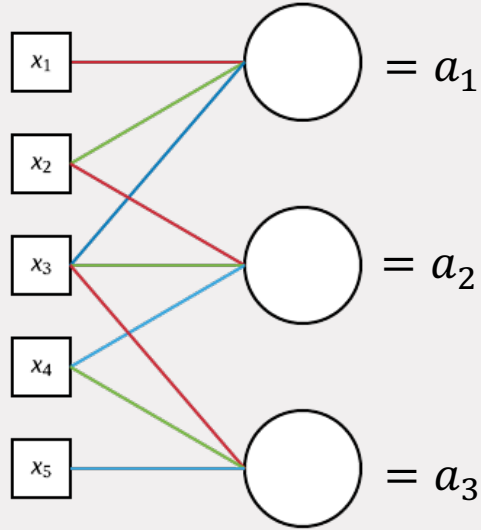
Convolutions + non-linearity (sigmoid)

$= sigm(a_1)$

$= sigm(a_2)$

$= sigm(a_3)$

TU/e
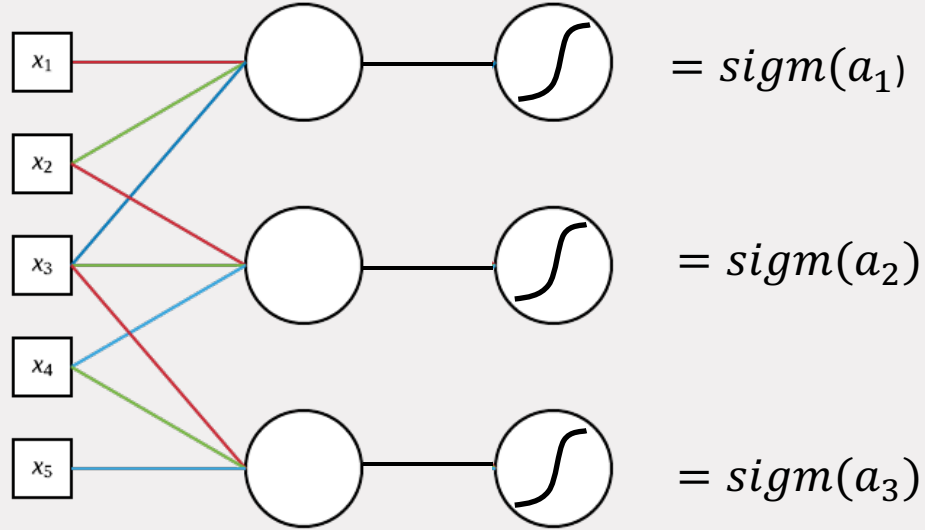
Convolutions + non-linearity (ReLU)

$= ReLU(a_1)$

$= ReLU(a_2)$

$= ReLU(a_3)$
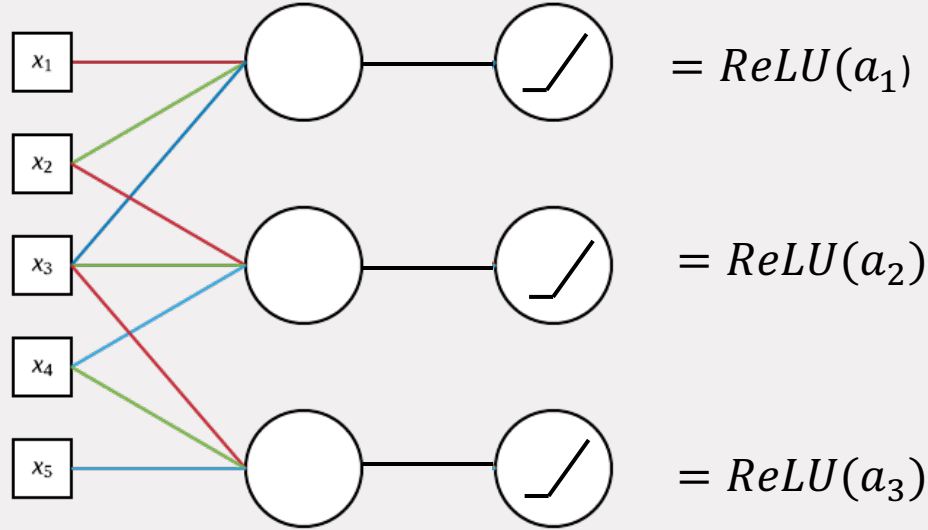
Convolutional kernels
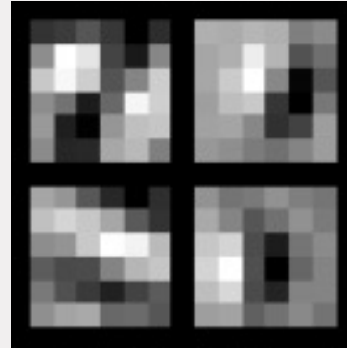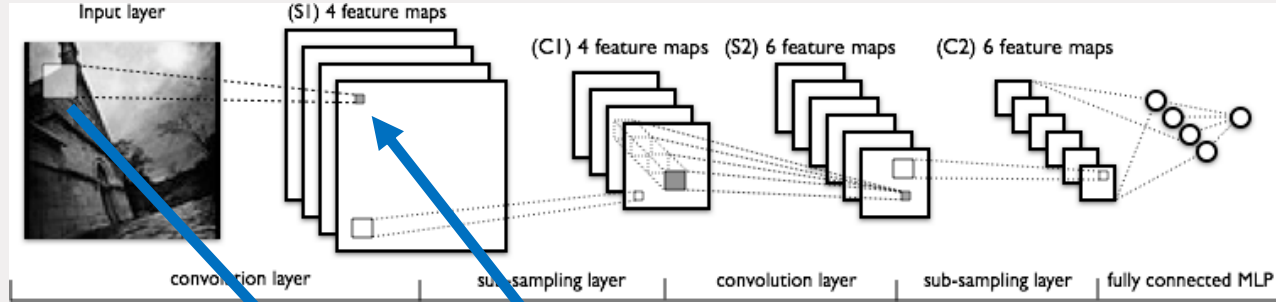
these are the weights (convolutional kernels) that produce the four feature maps

Kernel size
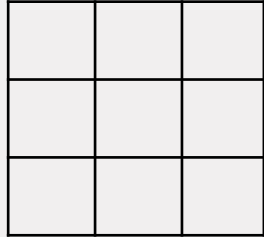
1 x 1   3 x 3   5 x 5
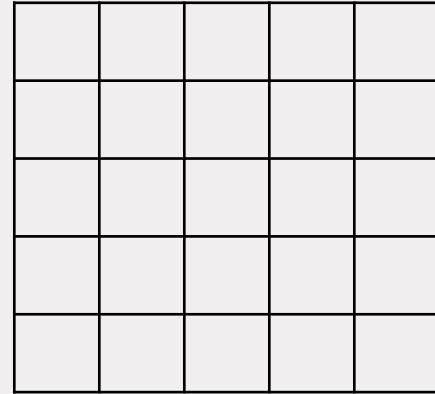
- More weights → more information
- More computations / memory
- Receptive field

TU/e
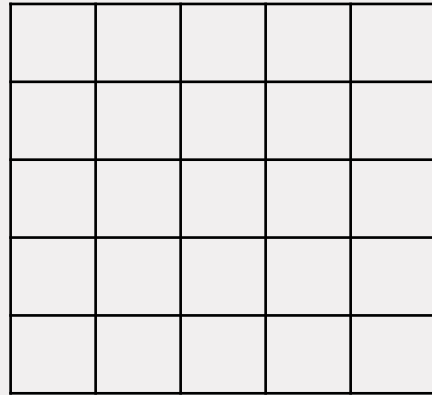
Receptive field

5 x 5

25 weights

Same receptive field

3 x 3

9 weights

2nd layer
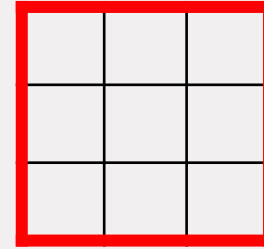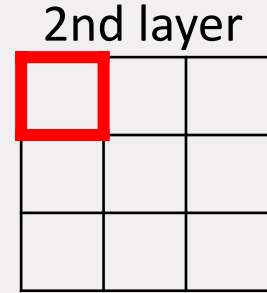
3 x 3

9 weights

= 18 weights

TU/e
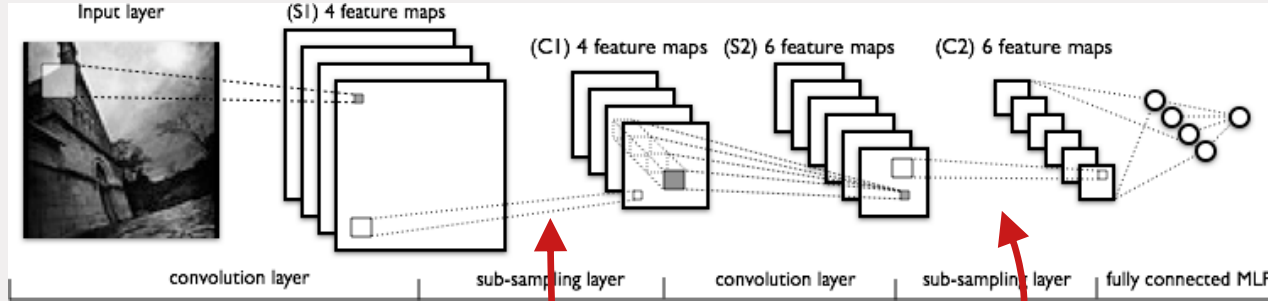
Max-pooling

max-pooling

Max-pooling

- Reduce size of feature space

- Maximum of features

- Typical kernel size = 2 x 2

TU/e

I_1 I_2 I_3 I_4 I_5

I_6 I_7 I_8 I_9 I_{10}

I_{11} I_{12} I_{13} I_{14} I_{15}

I_{16} I_{17} I_{18} I_{19} I_{20}

I_{21} I_{22} I_{23} I_{24} I_{25}

2×2 max pooling

$\max(I_1, I_2, I_6, I_7)$

TU/e

I_1 I_2 I_3 I_4 I_5

I_6 I_7 I_8 I_9 I_{10}

I_{11} I_{12} I_{13} I_{14} I_{15}

I_{16} I_{17} I_{18} I_{19} I_{20}

I_{21} I_{22} I_{23} I_{24} I_{25}

2×2 max pooling

$\max(I_1,I_2,I_6,I_7)$ $\max(I_3,I_4,I_8,I_9)$

TU/e

I_1 I_2 I_3 I_4 I_5
I_6 I_7 I_8 I_9 I_{10}
I_{11} I_{12} I_{13} I_{14} I_{15}
I_{16} I_{17} I_{18} I_{19} I_{20}
I_{21} I_{22} I_{23} I_{24} I_{25}

2×2 max pooling

$\max(I_1, I_2, I_6, I_7)$ $\max(I_3, I_4, I_8, I_9)$ $\max(I_5, I_{10})$
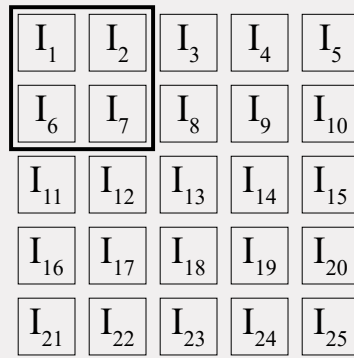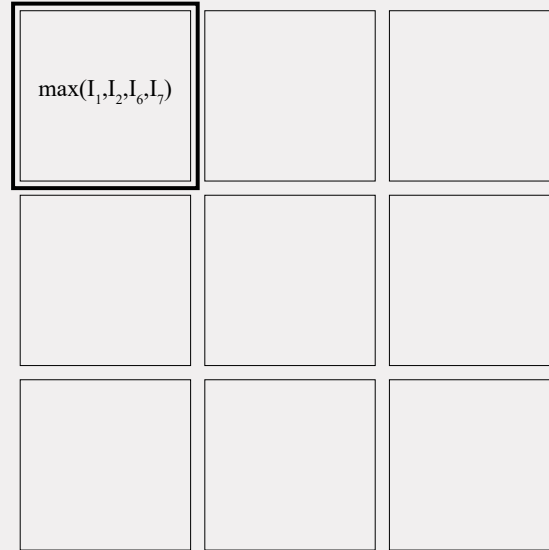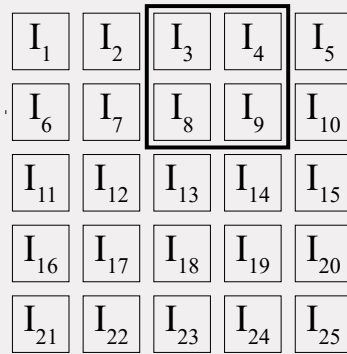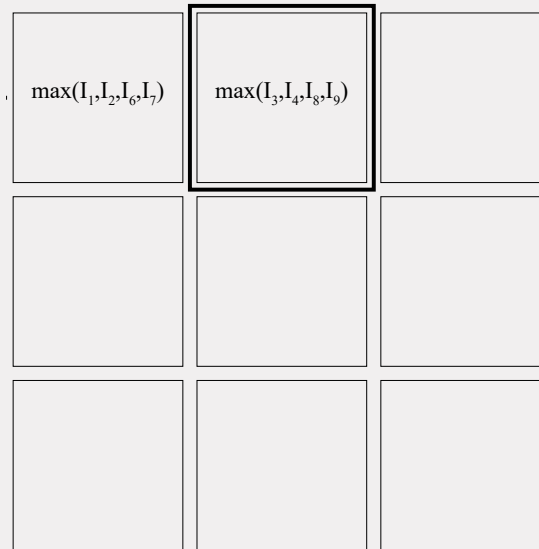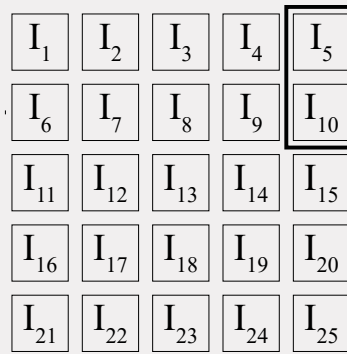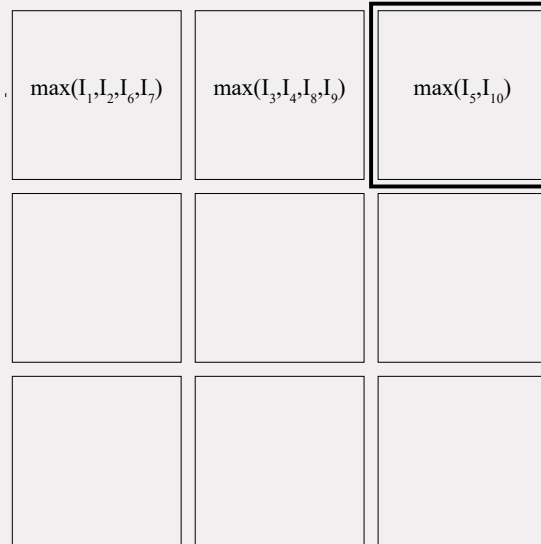
In this example:
no zero padding

TU/e

I_1	I_2	I_3	I_4	I_5
I_6	I_7	I_8	I_9	I_{10}
I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}

2×2 max pooling

$\max(I_1,I_2,I_6,I_7)$	$\max(I_3,I_4,I_8,I_9)$	$\max(I_5,I_{10})$
$\max(I_{11},I_{12},I_{16},I_{17})$		

TU/e

I_1	I_2	I_3	I_4	I_5
I_6	I_7	I_8	I_9	I_{10}
I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}

2×2 max pooling

$\max(I_1,I_2,I_6,I_7)$	$\max(I_3,I_4,I_8,I_9)$	$\max(I_5,I_{10})$
$\max(I_{11},I_{12},I_{16},I_{17})$	$\max(I_{13},I_{14},I_{18},I_{19})$	

TU/e

I_1 I_2 I_3 I_4 I_5

I_6 I_7 I_8 I_9 I_{10}

I_{11} I_{12} I_{13} I_{14} I_{15}

I_{16} I_{17} I_{18} I_{19} I_{20}

I_{21} I_{22} I_{23} I_{24} I_{25}

2×2 max pooling

$\max(I_1,I_2,I_6,I_7)$ $\max(I_3,I_4,I_8,I_9)$ $\max(I_5,I_{10})$

$\max(I_{11},I_{12},I_{16},I_{17})$ $\max(I_{13},I_{14},I_{18},I_{19})$ $\max(I_{15},I_{20})$

$\max(I_{21},I_{22})$

TU/e

I_1	I_2	I_3	I_4	I_5
I_6	I_7	I_8	I_9	I_{10}
I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}

2×2 max pooling

$\max(I_1,I_2,I_6,I_7)$	$\max(I_3,I_4,I_8,I_9)$	$\max(I_5,I_{10})$
$\max(I_{11},I_{12},I_{16},I_{17})$	$\max(I_{13},I_{14},I_{18},I_{19})$	$\max(I_{15},I_{20})$
$\max(I_{21},I_{22})$	$\max(I_{23},I_{24})$	

TU/e

I_1 I_2 I_3 I_4 I_5

I_6 I_7 I_8 I_9 I_{10}

I_{11} I_{12} I_{13} I_{14} I_{15}

I_{16} I_{17} I_{18} I_{19} I_{20}

I_{21} I_{22} I_{23} I_{24} I_{25}

2×2 max pooling

$\max(I_1,I_2,I_6,I_7)$	$\max(I_3,I_4,I_8,I_9)$	$\max(I_5,I_{10})$
$\max(I_{11},I_{12},I_{16},I_{17})$	$\max(I_{13},I_{14},I_{18},I_{19})$	$\max(I_{15},I_{20})$
$\max(I_{21},I_{22})$	$\max(I_{23},I_{24})$	I_{25}

TU/e
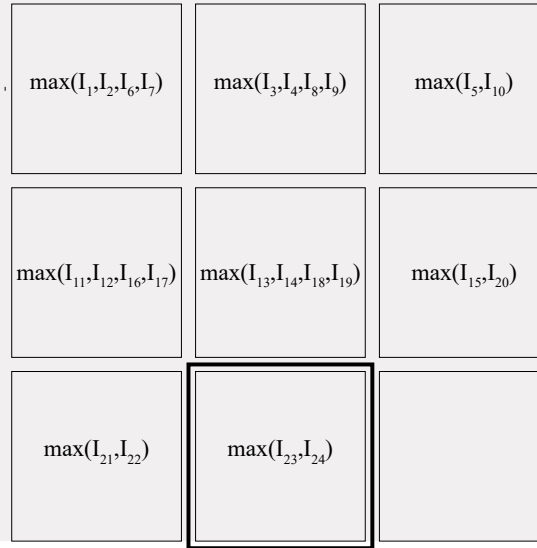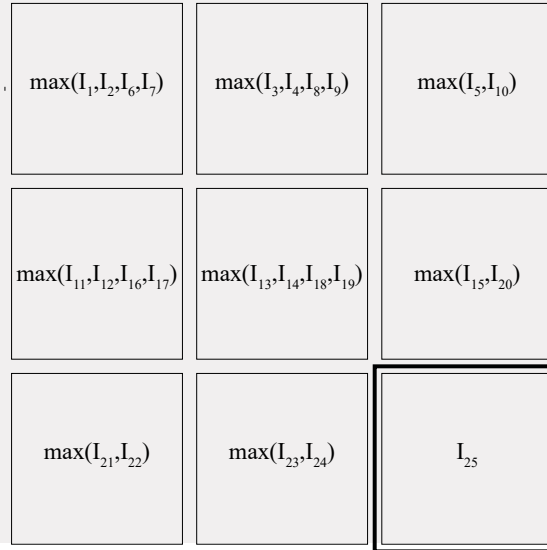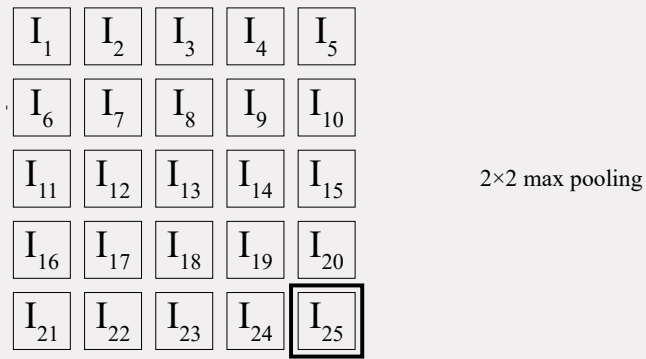
Benefits of max-pooling

- "Quickly" reduces the number size of the feature maps
- Introduces translation invariance (slightly translated version of the input image will result in the same output)

Alternative: Average Pooling

TU/e

Let try it out!

https://adamharley.com/nn_vis/cnn/2d.html

An Interactive Node-Link Visualization of Convolutional Neural Networks

Adam W. Harley[✉]

Department of Computer Science, Ryerson University,
Toronto, ON M5B 2K3, Canada
aharley@scs.ryerson.ca

Training of Convolutional Neural Networks

- Similar to training of 'fully connected' neural networks

- Choose some (random) initial values for network weights

- Optimize networks weights with respect to a loss function that describes difference between network output and label/annotation

- Update networks weights iteratively

Through a process called 'backpropagation'. A good explanation can be found here: https://www.youtube.com/watch?v=i94OvYb6noo (5:10 - 28:00, not exam material)

- Keep track of model performance on train & validation set

TU/e

Summary

- Concept of **convolutions** in a neural network

- Why can we use a **convolutional approach** for (medical) **images**

- Convolutions enable development of **deep** (and large) **neural networks**

- **Max-pooling layer** in a convolutional neural network

- **Kernel size**

- **Receptive field**

TU/e

8DC00 - Convolutional neural networks - Heslinga