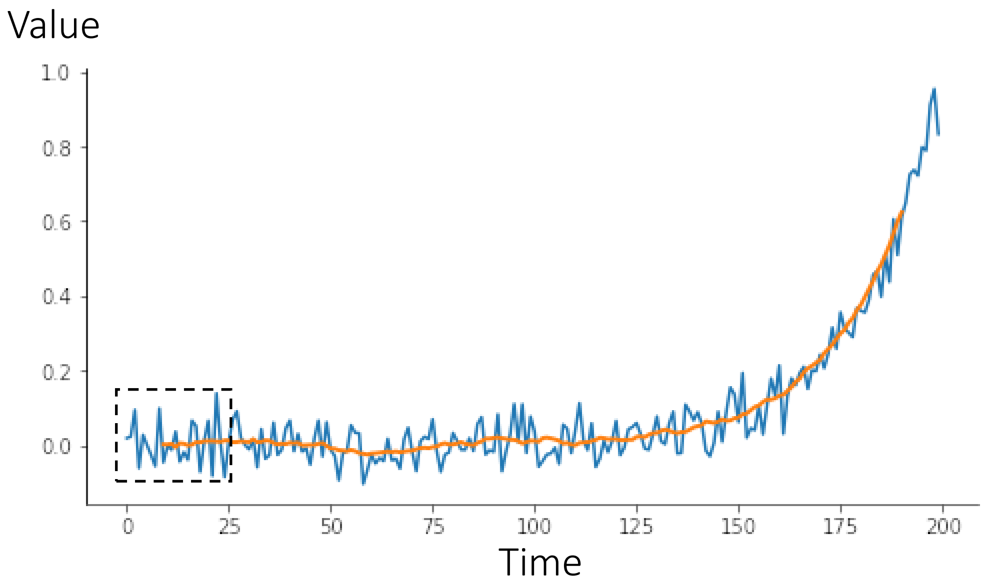
UNIVERSITY OF BIRMINGHAM

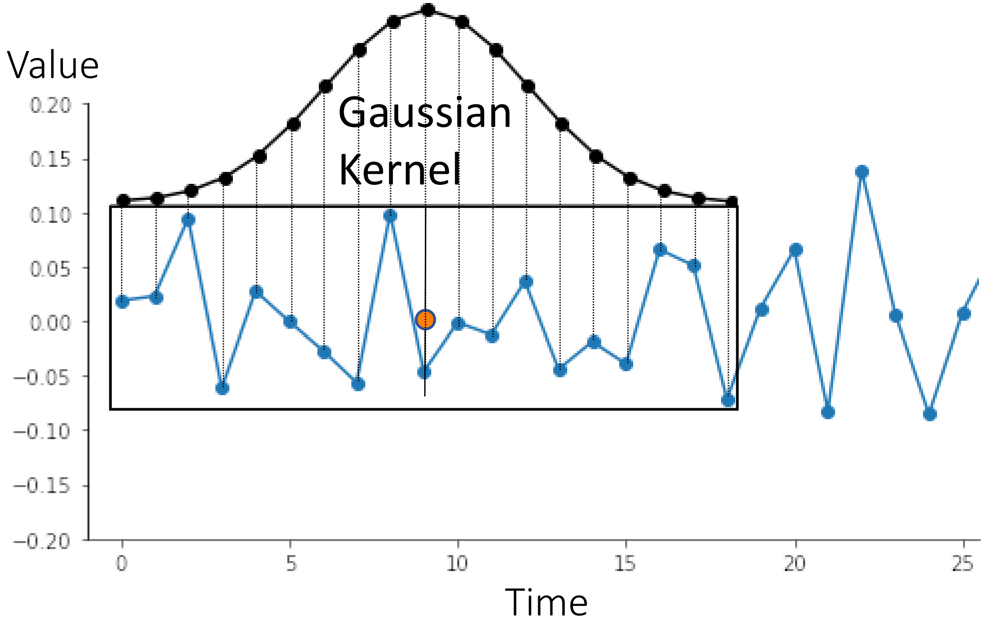
Neural Computation

Monday 30th of October

Convolutions

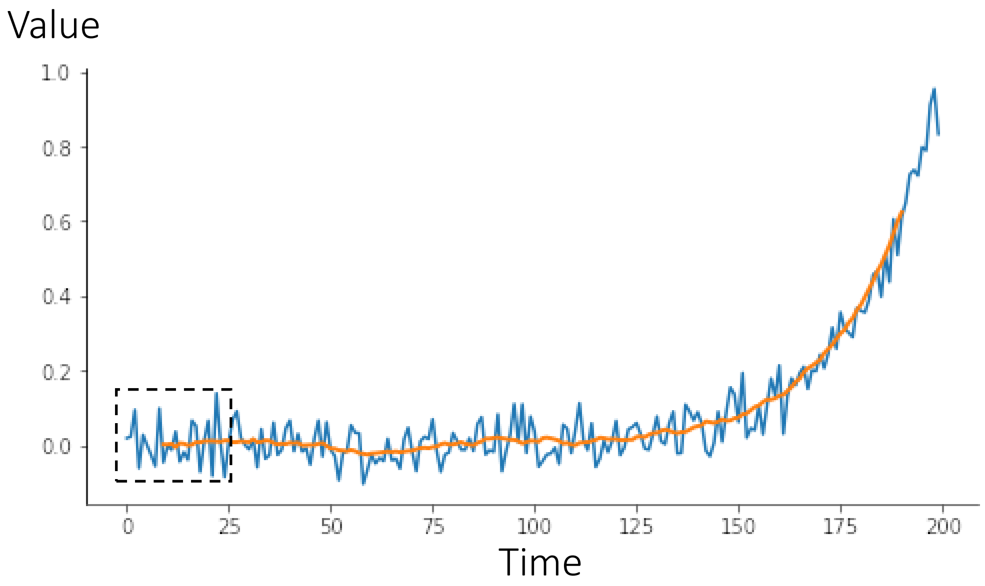
Weighted Average Smoothing is Convolution

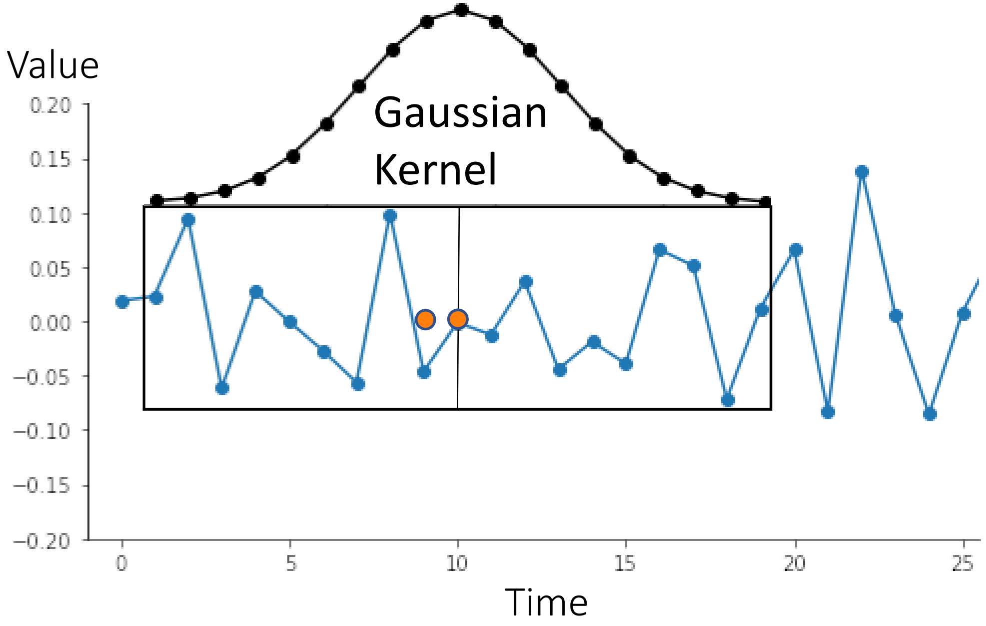




$$s(t) = \sum_{k=-m}^{m} x(t+k)\, g(k)$$

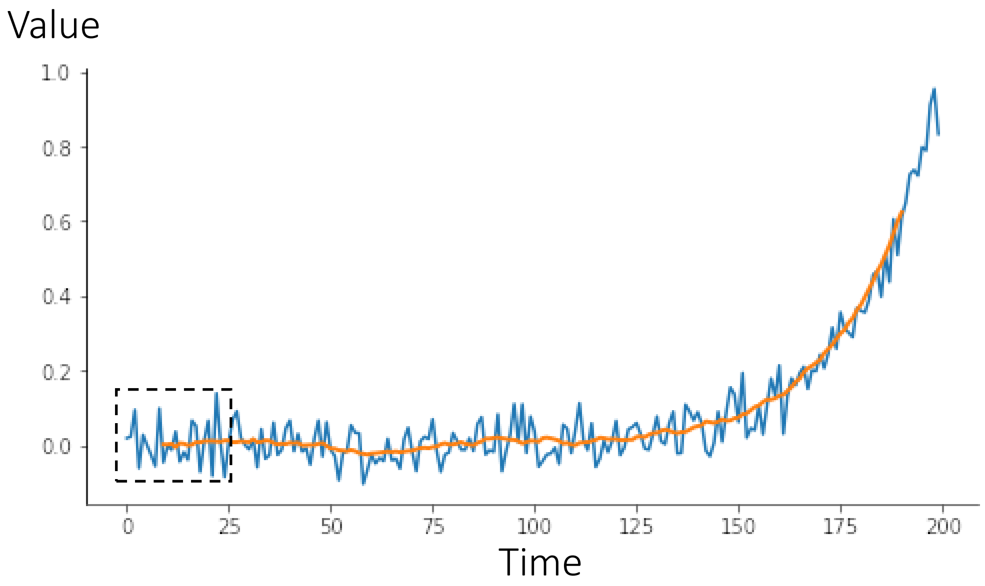
Weighted Average Smoothing is Convolution

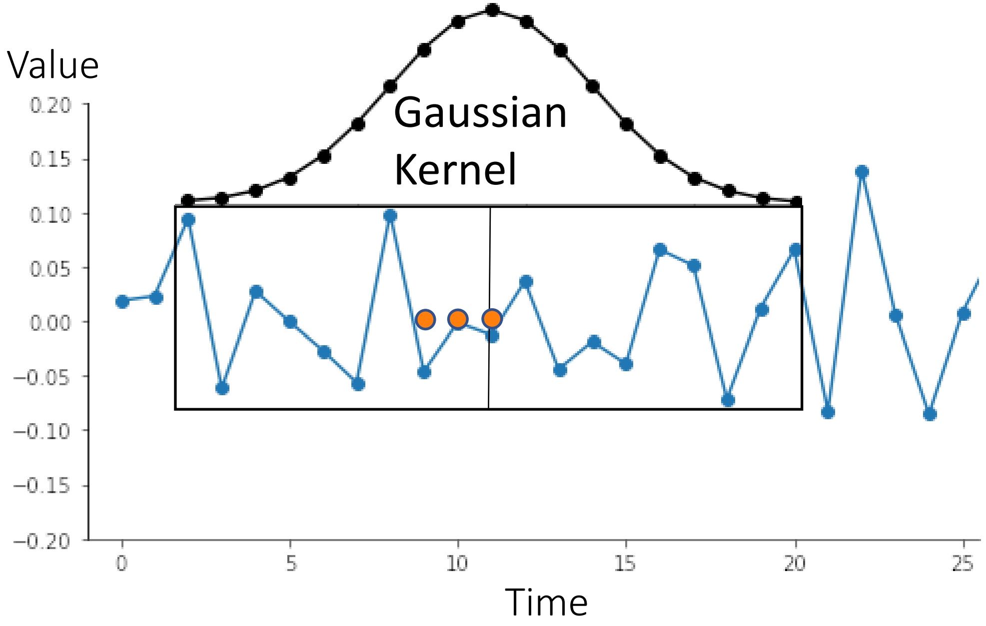




$$s(t) = \sum_{k=-m}^{m} x(t+k)\, g(k)$$

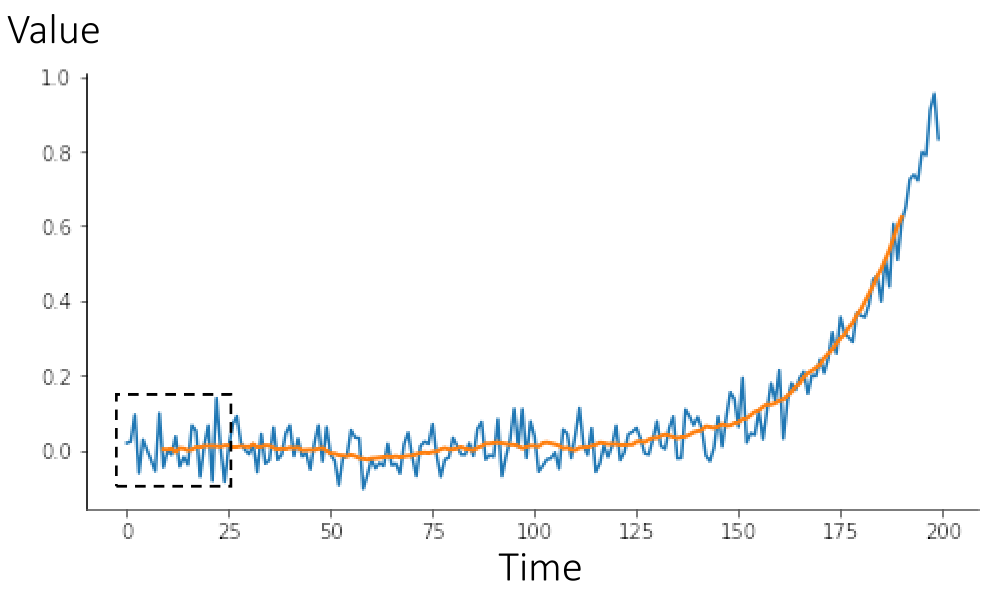
Weighted Average Smoothing is Convolution

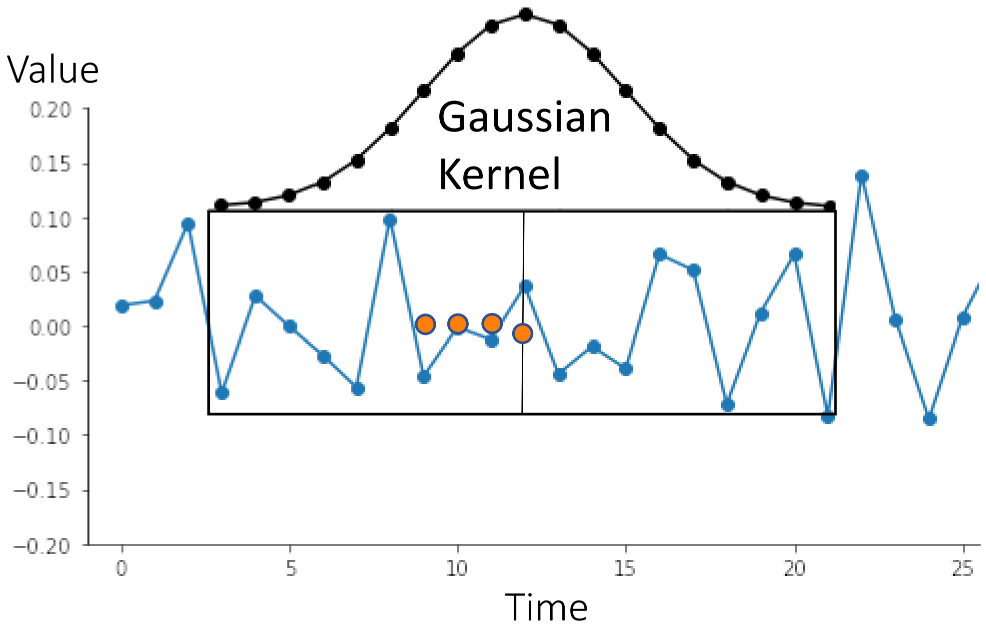




$$s(t) = \sum_{k=-m}^{m} x(t+k)\,g(k)$$

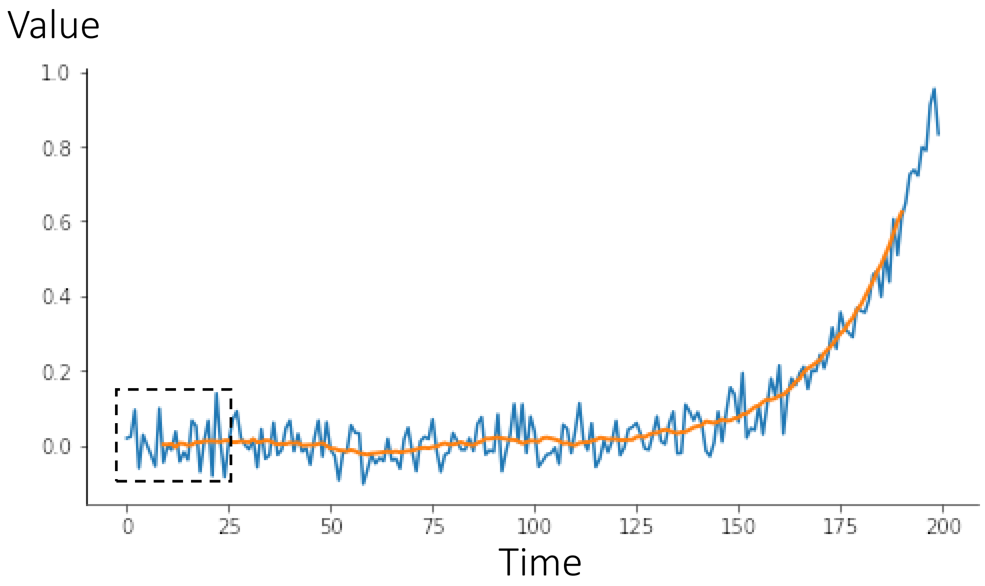
Weighted Average Smoothing is Convolution

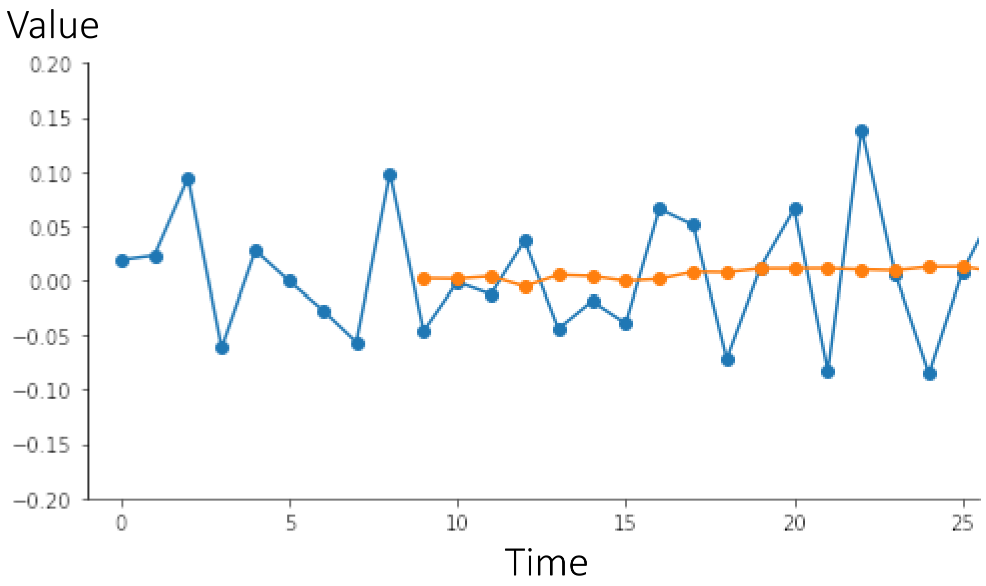




$$s(t) = \sum_{k=-m}^{m} x(t+k)\,g(k)$$

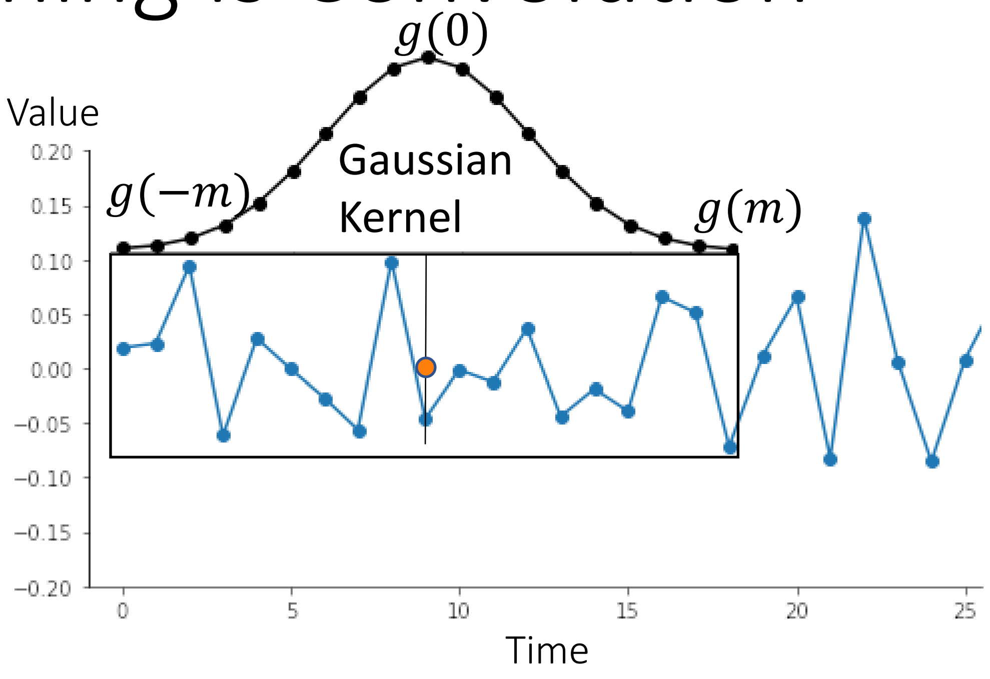
Weighted Average Smoothing is Convolution





$$s(t) = \sum_{k=-m}^{m} x(t+k)\, g(k)$$

Weighted Average Smoothing is Convolution



Definition of Convolution

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-k)g(k)dk$$

Discrete Version

$g(k) = 0$ outside $[-m, m]$

$$(f * g)(t) = \sum_{k=-\infty}^{\infty} f(t-k)\, g(k)$$

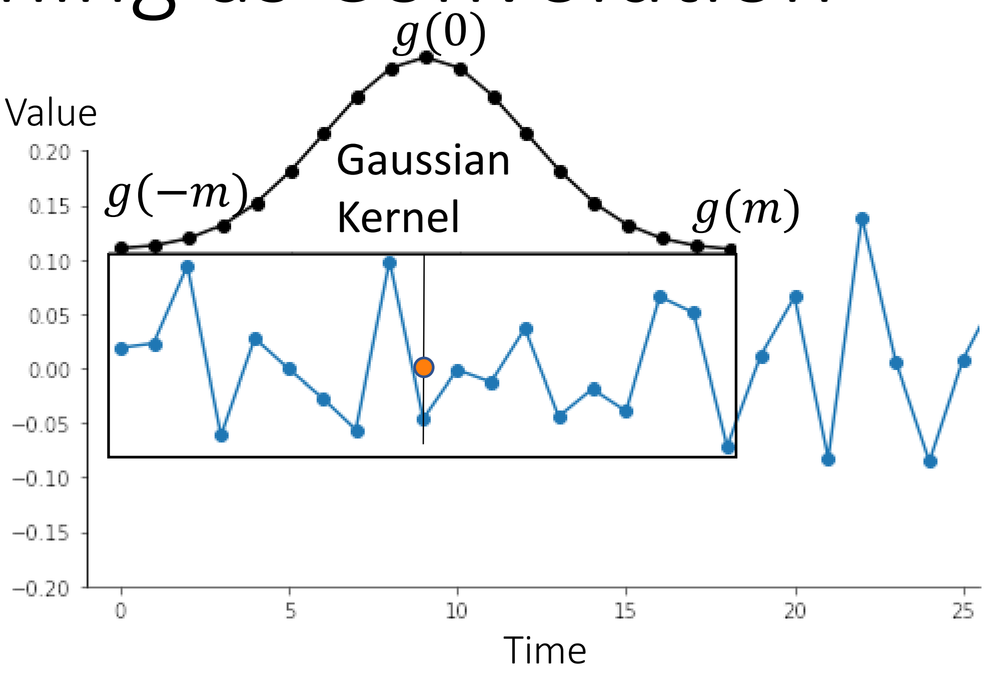
Flipped $g(k)$

$$s(t) = \sum_{k=-m}^{m} x(t+k)\, g(k)$$

Weighted Average Smoothing as Convolution

Definition of Convolution

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-k)g(k)dk$$



Discrete Version

$g(k) = 0$ outside $[-m, m]$

$$(f * g)(t) = \sum_{k=-\infty}^{\infty} f(t+k)\, g(-k)$$

Flipped $g(k)$

$$s(t) = \sum_{k=-m}^{m} x(t+k)\, g(k)$$

Weighted averaging is a convolution

Convolutions - Properties

Commutativity

$$f * g = g * f$$

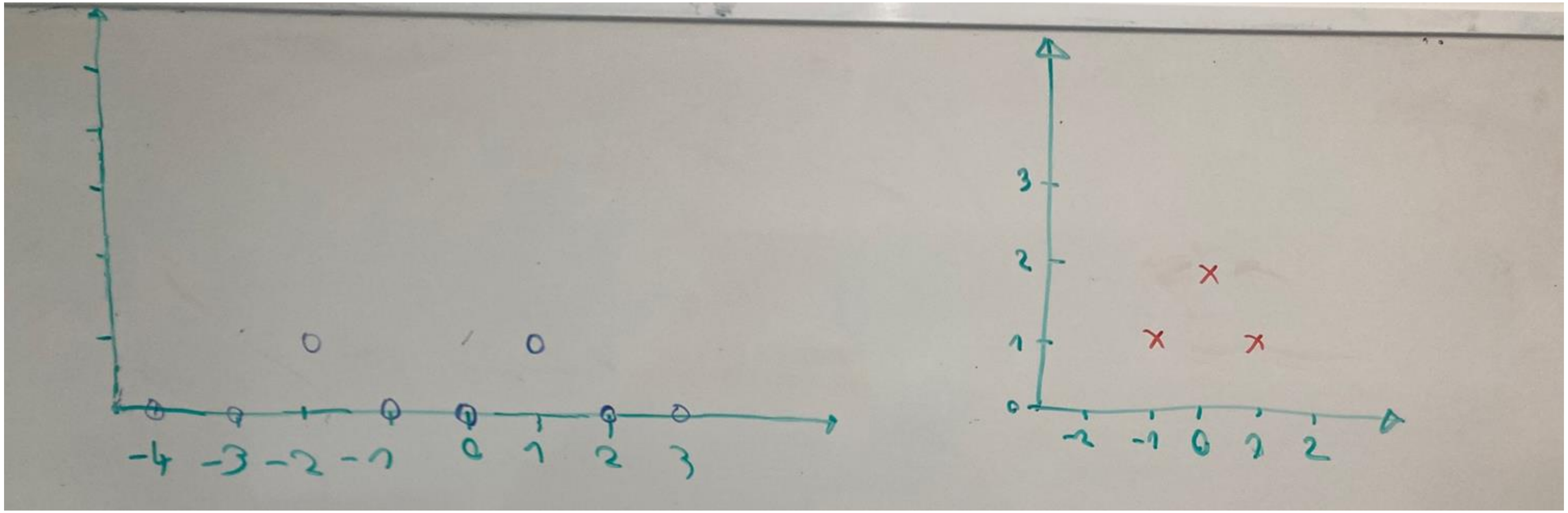
Associativity

$$(f * g) * h = f * (g * h)$$

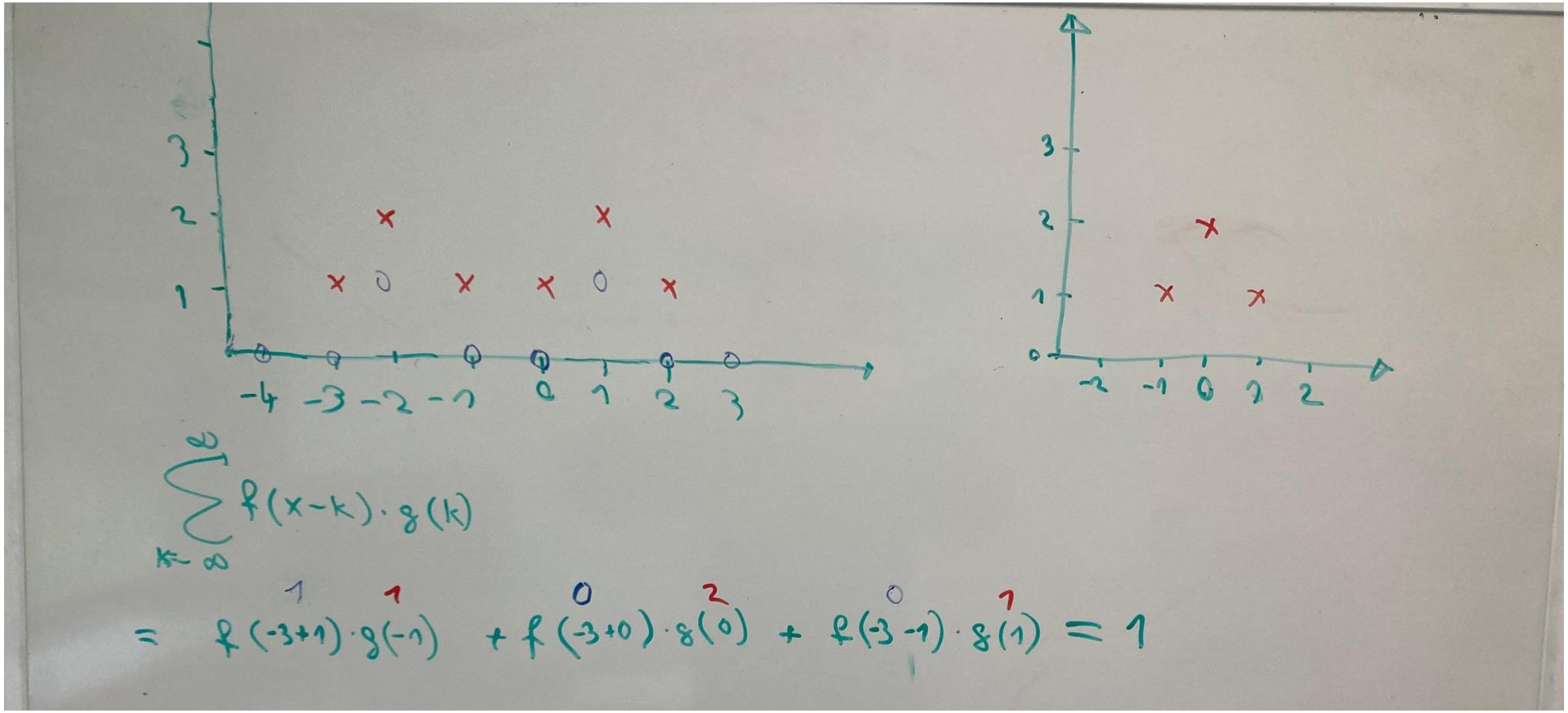
$$a(f * g) = (af) * g$$

Distributivity

$$(f + g) * h = (f * h) + (g * h)$$

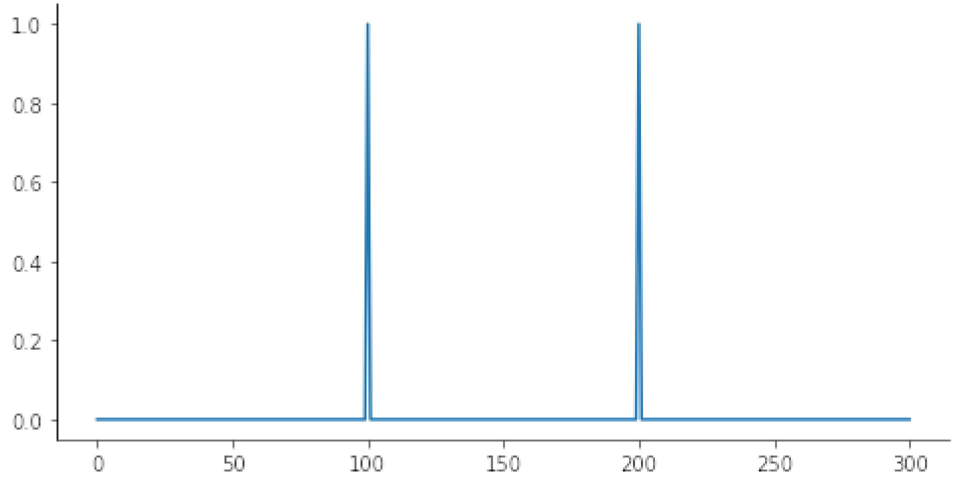


$$(f * g)(t) = \sum_{k=-\infty}^{\infty} f(t-k)\, g(k) = \sum_{k=-\infty}^{\infty} f(t+k)\, g(-k)$$

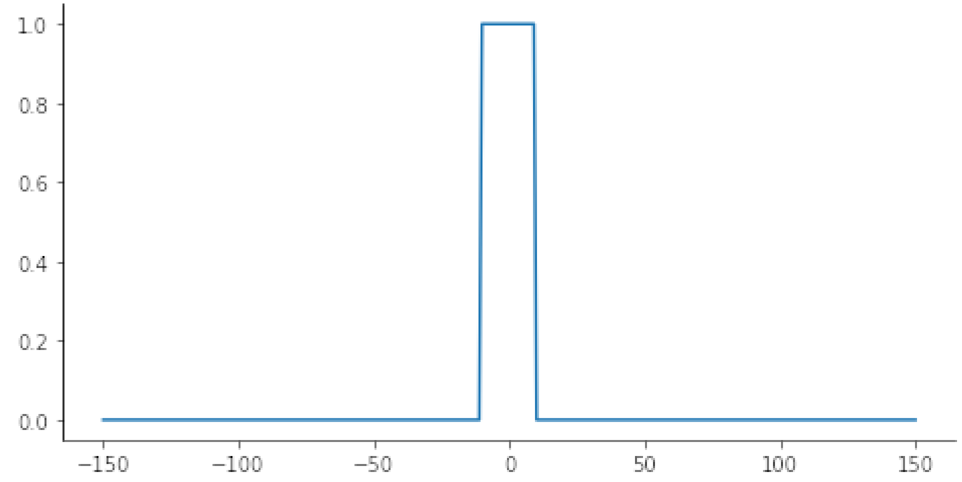


3
2
1
-4 -3 -2 -1 0 1 2 3
3
2
1
0
-2 -1 0 1 2
$\sum_{k=-\infty}^{\infty} f(x-k)\cdot g(k)$
1 1 0 2 0 1
$= f(-3+1)\cdot g(-1) + f(-3+0)\cdot g(0) + f(-3-1)\cdot g(1) = 1$

Convolutions - Examples

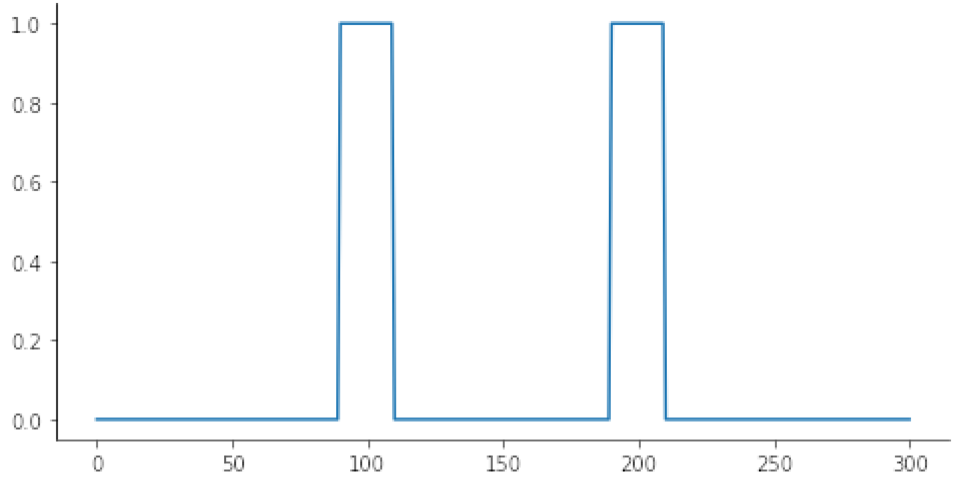


*

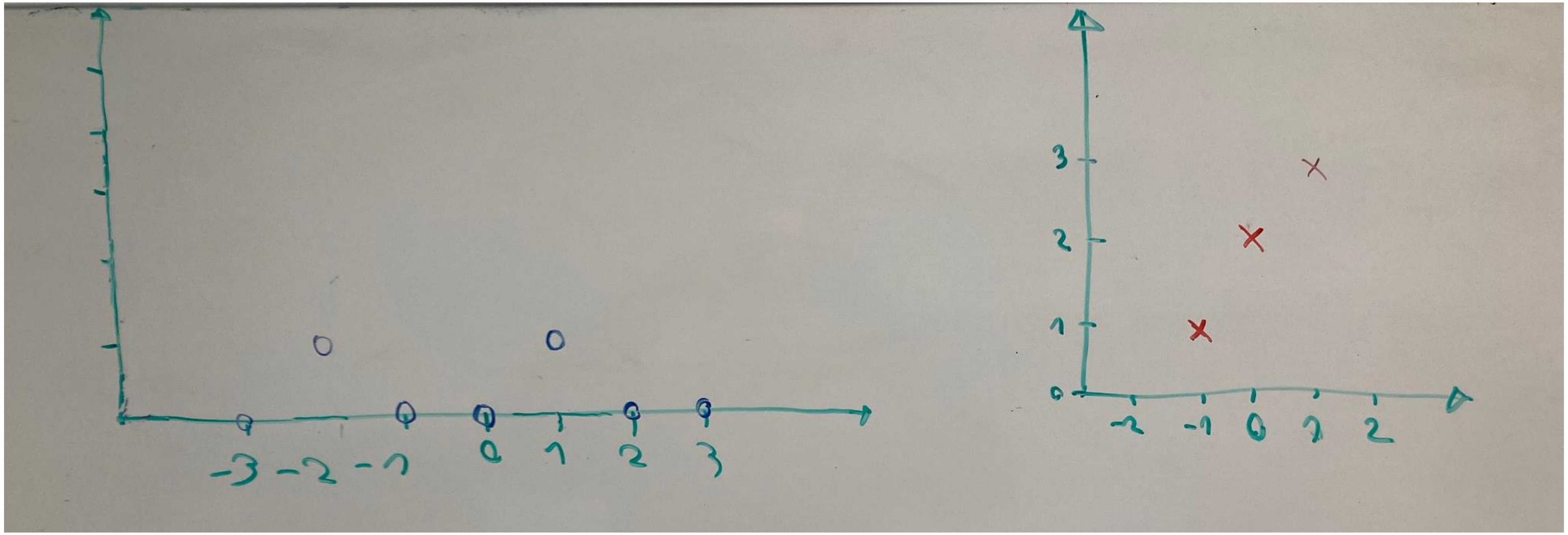


=

$$(f * g)(t) = \sum_{k=-\infty}^{\infty} f(t-k)\, g(k)$$

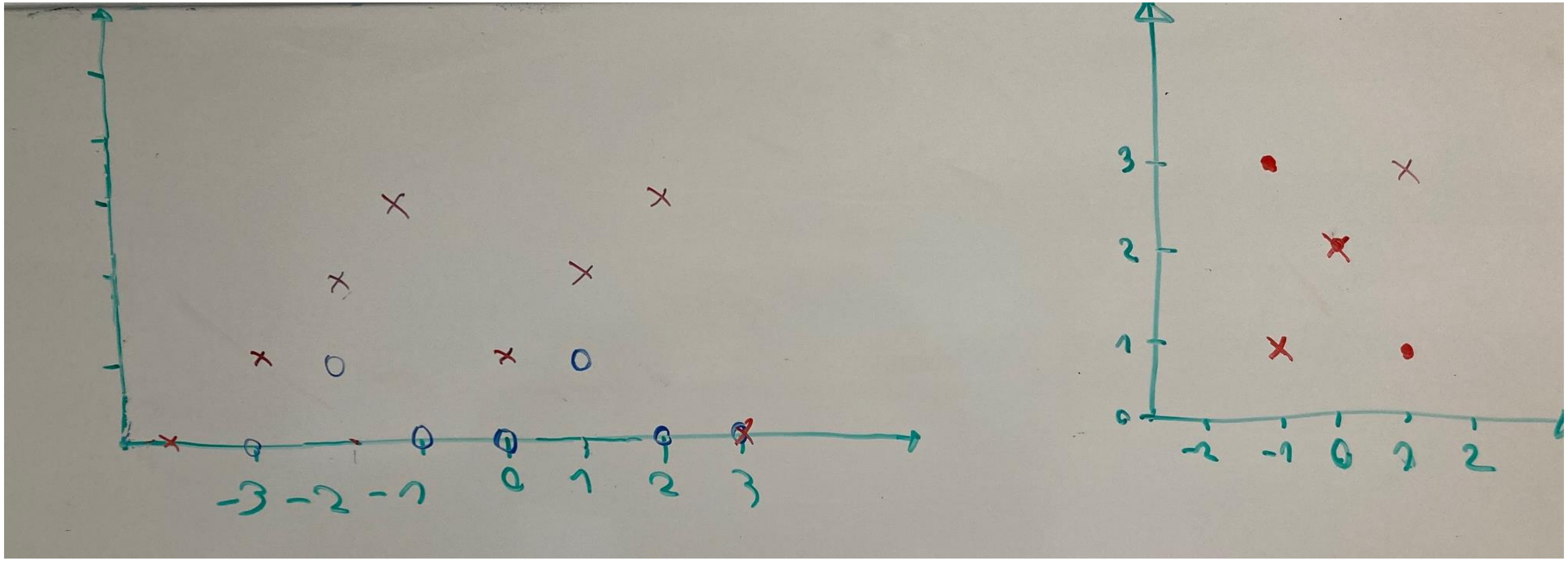


What is the Result of the Convolution?



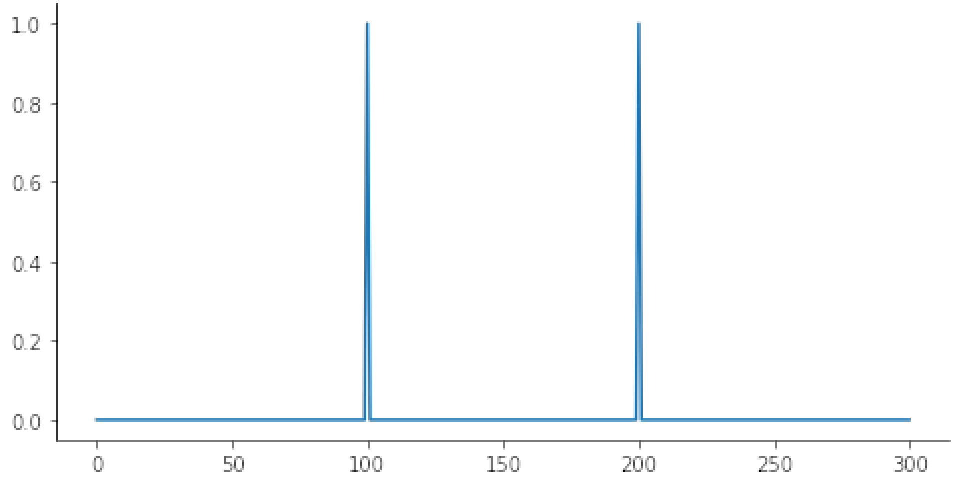
$$(f * g)(t) = \sum_{k=-\infty}^{\infty} f(t-k)\, g(k) = \sum_{k=-\infty}^{\infty} f(t+k)\, g(-k)$$

What is the Result of the Convolution?

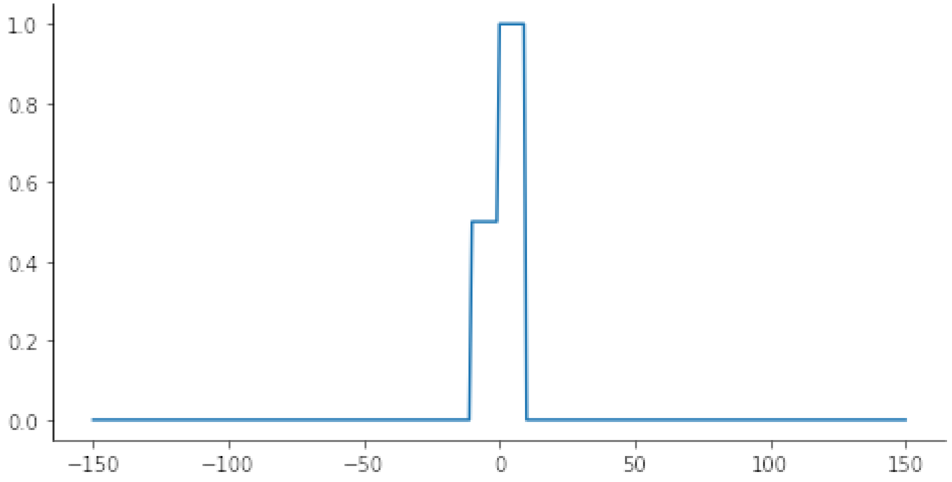


$$(f * g)(t) = \sum_{k=-\infty}^{\infty} f(t-k)\, g(k) = \sum_{k=-\infty}^{\infty} f(t+k)\, g(-k)$$

Convolutions - Examples

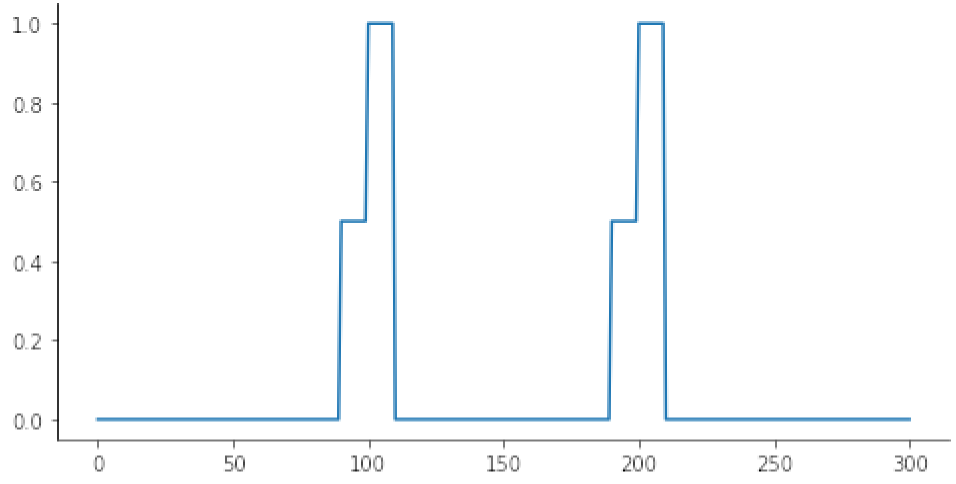


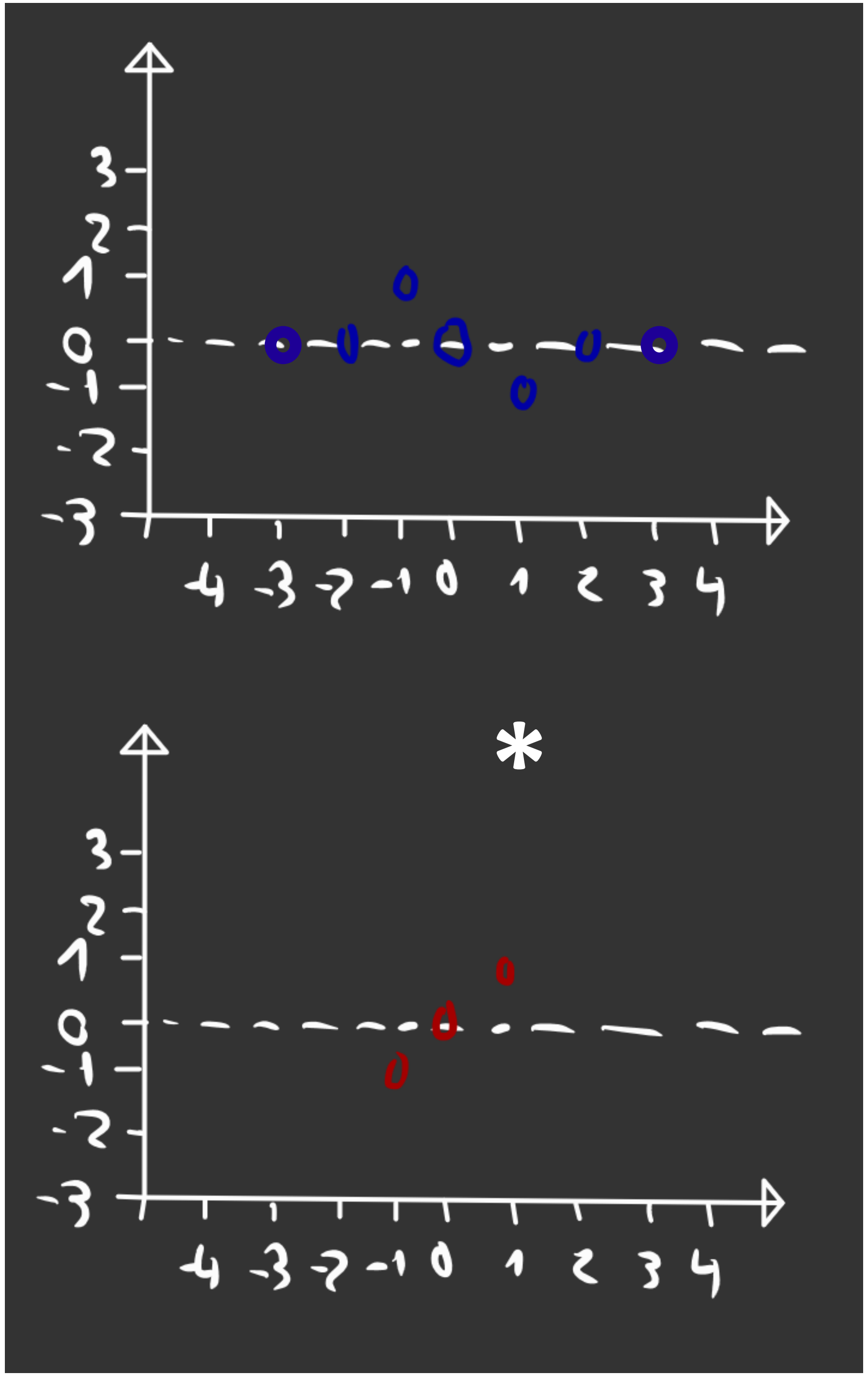
*



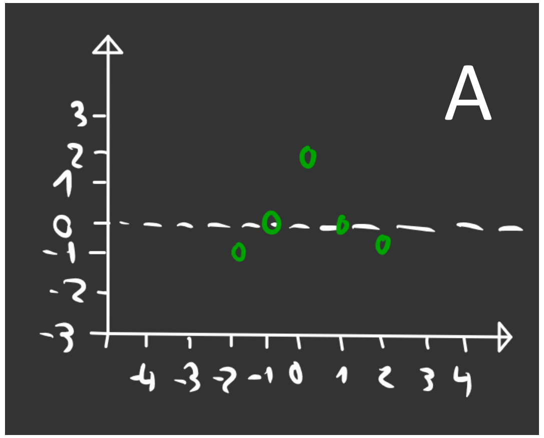
=

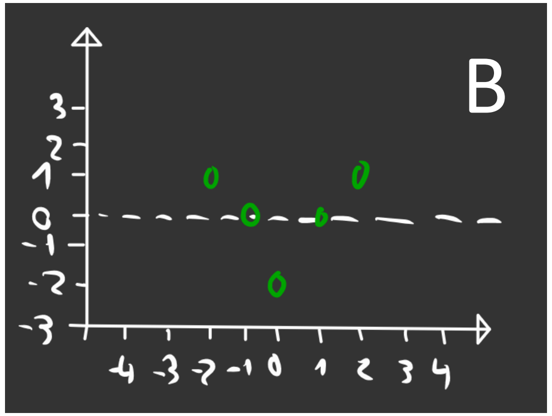
$$(f * g)(t) = \sum_{k=-\infty}^{\infty} f(t-k)\, g(k)$$

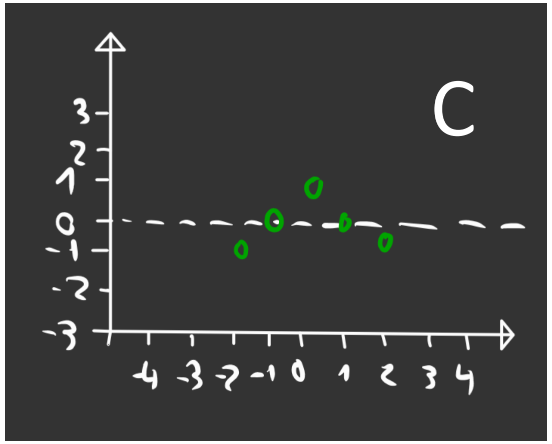


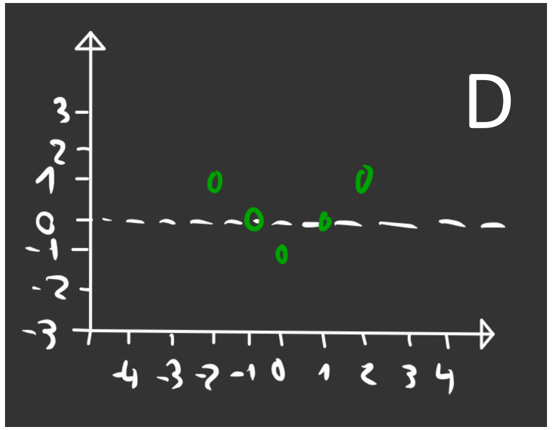

*

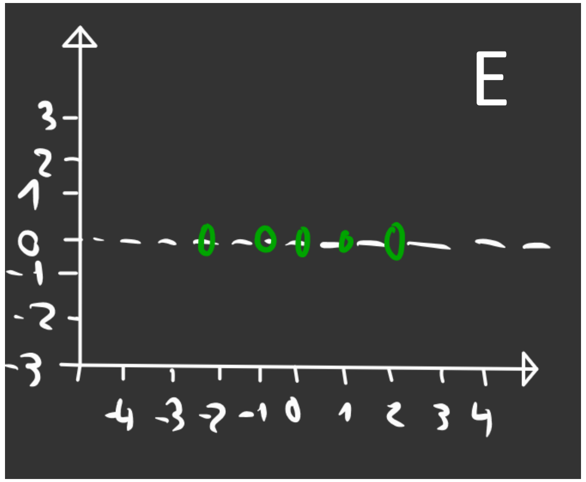
=

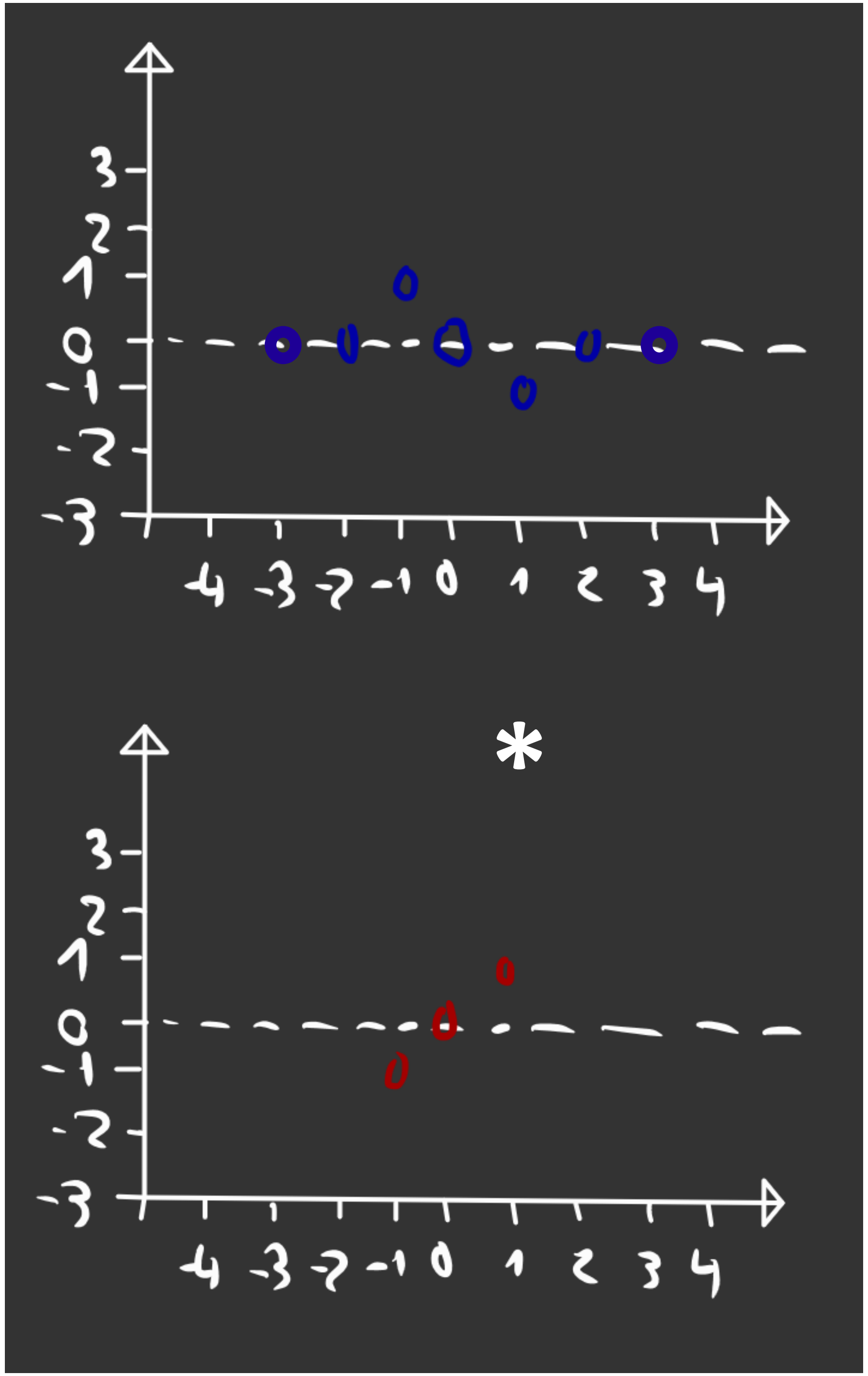

A


B

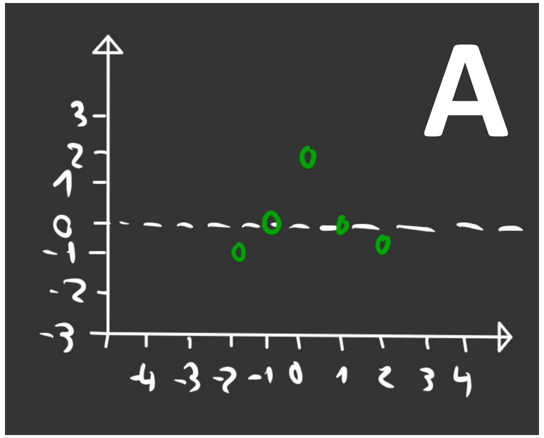

C

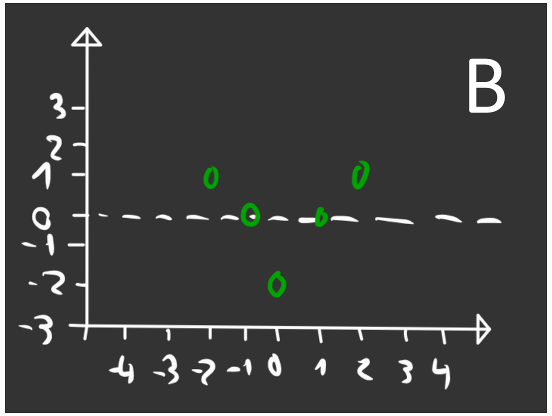

D

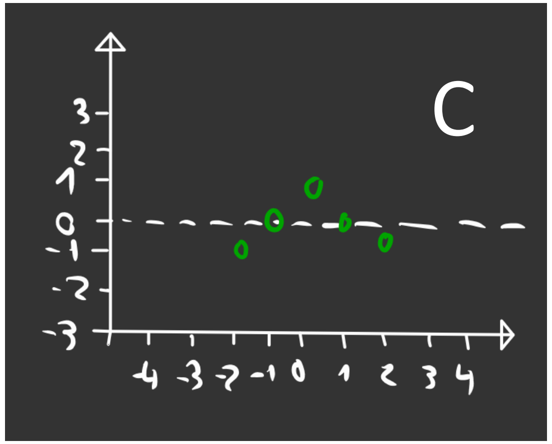

E

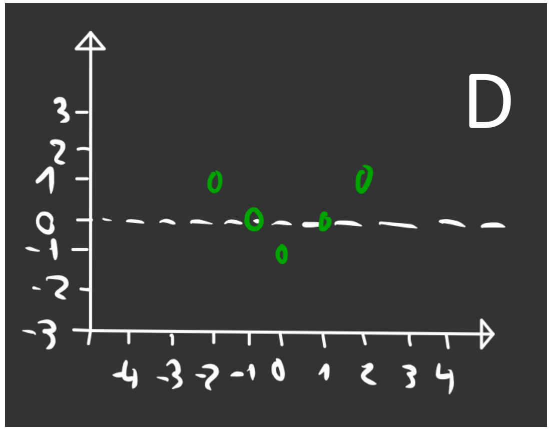

*

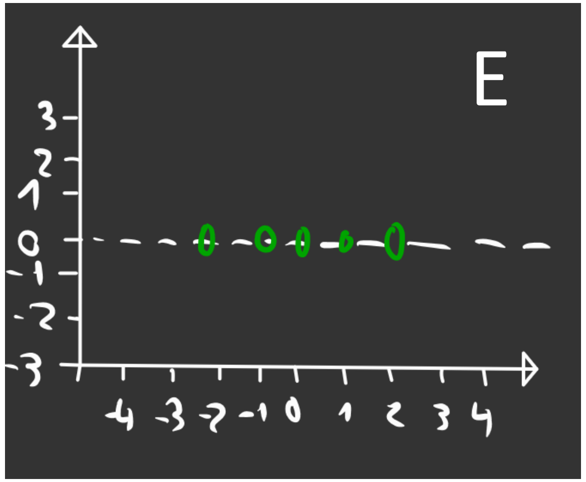
=


A


B


C

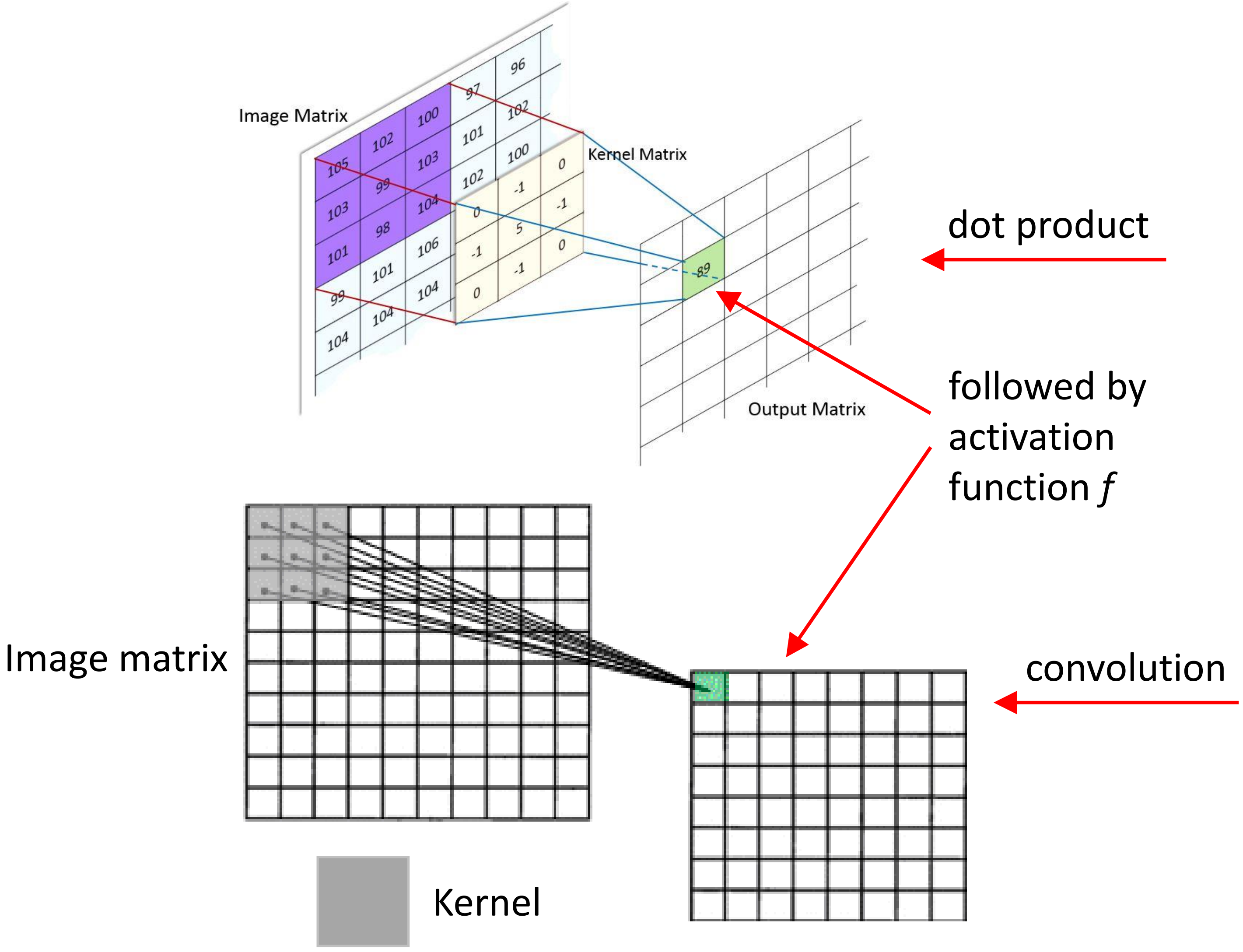

D


E

Convolution Reverb

Convolutional Neural Network

Depending on values, a kernel can cause a wide range of effects

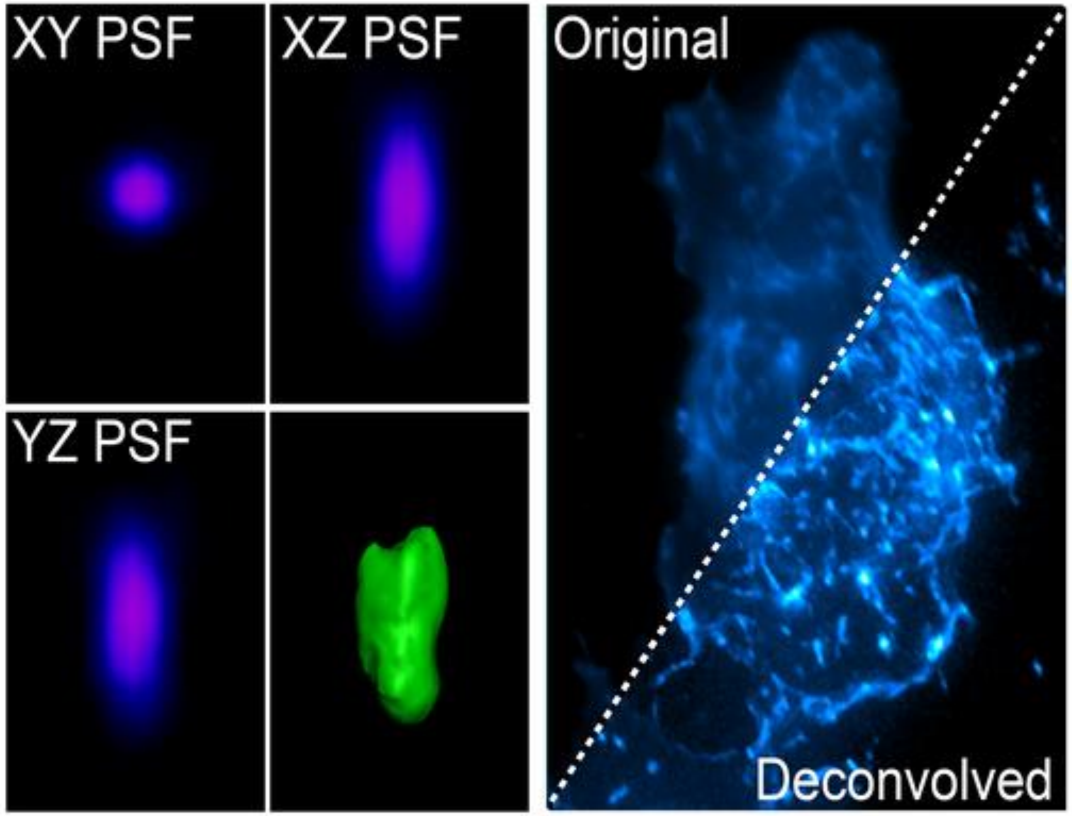


Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Deconvolution of Images

Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang, "Deblurring Text Images via L_0-Regularized Intensity and Gradient Prior",
IEEE Computer Society Conference on Computer Vision and Pattern Recognition (***CVPR***), 2014

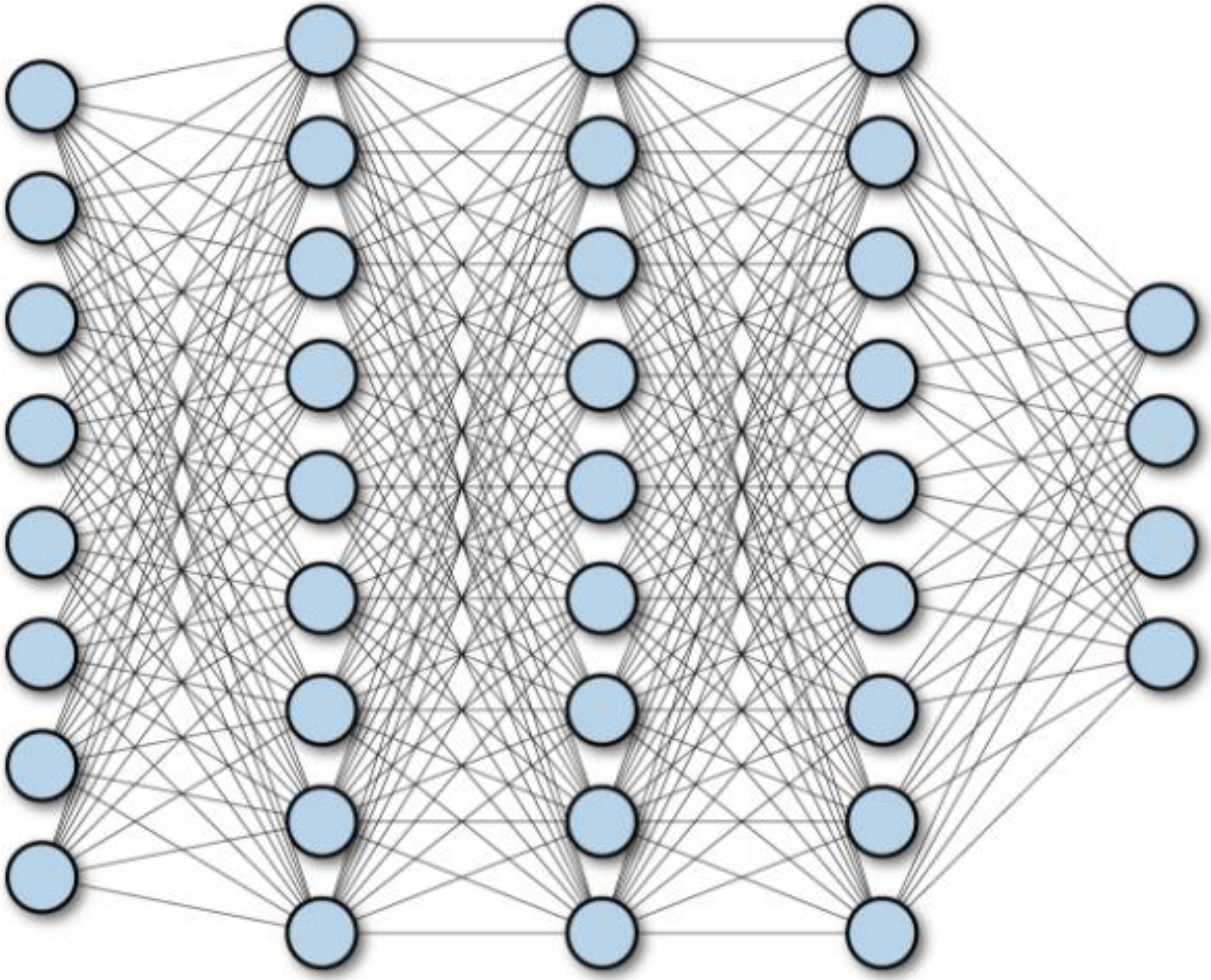
Deconvolution in Microscopy

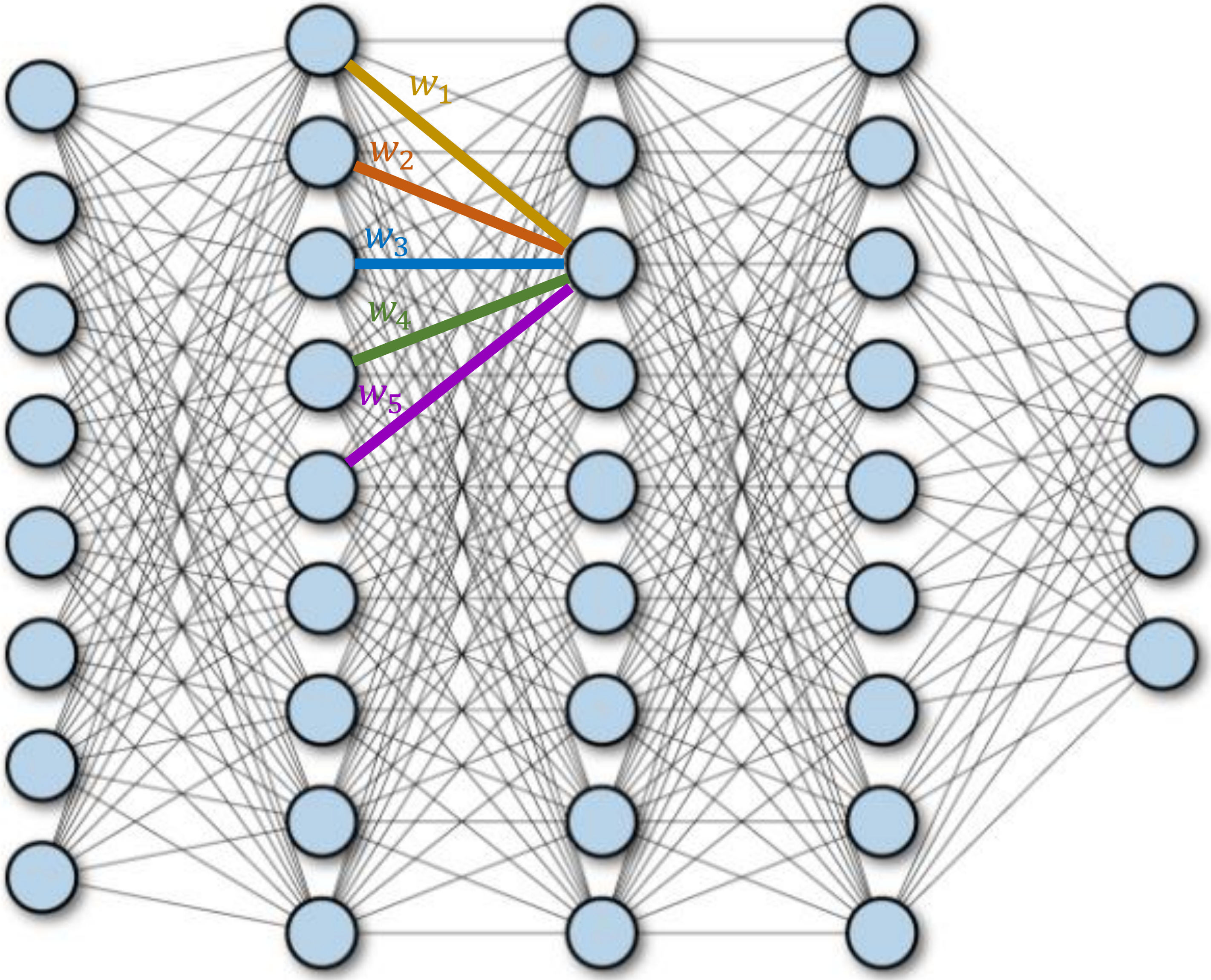


The University of Queensland Australia | Institute for Molecular Bioscience

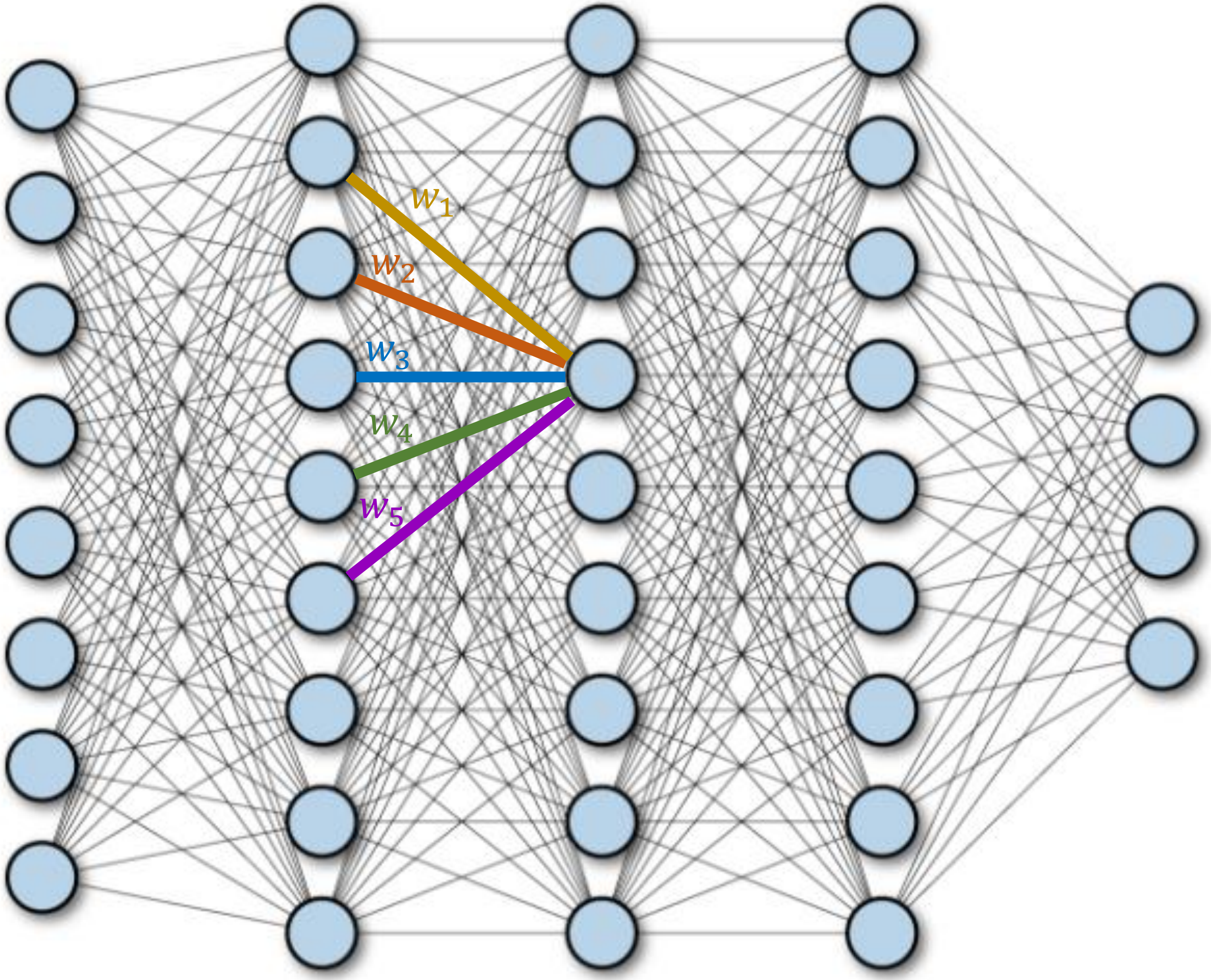
https://imb.uq.edu.au/facilities/microscopy/hardware-software/deconvolution-microscopes

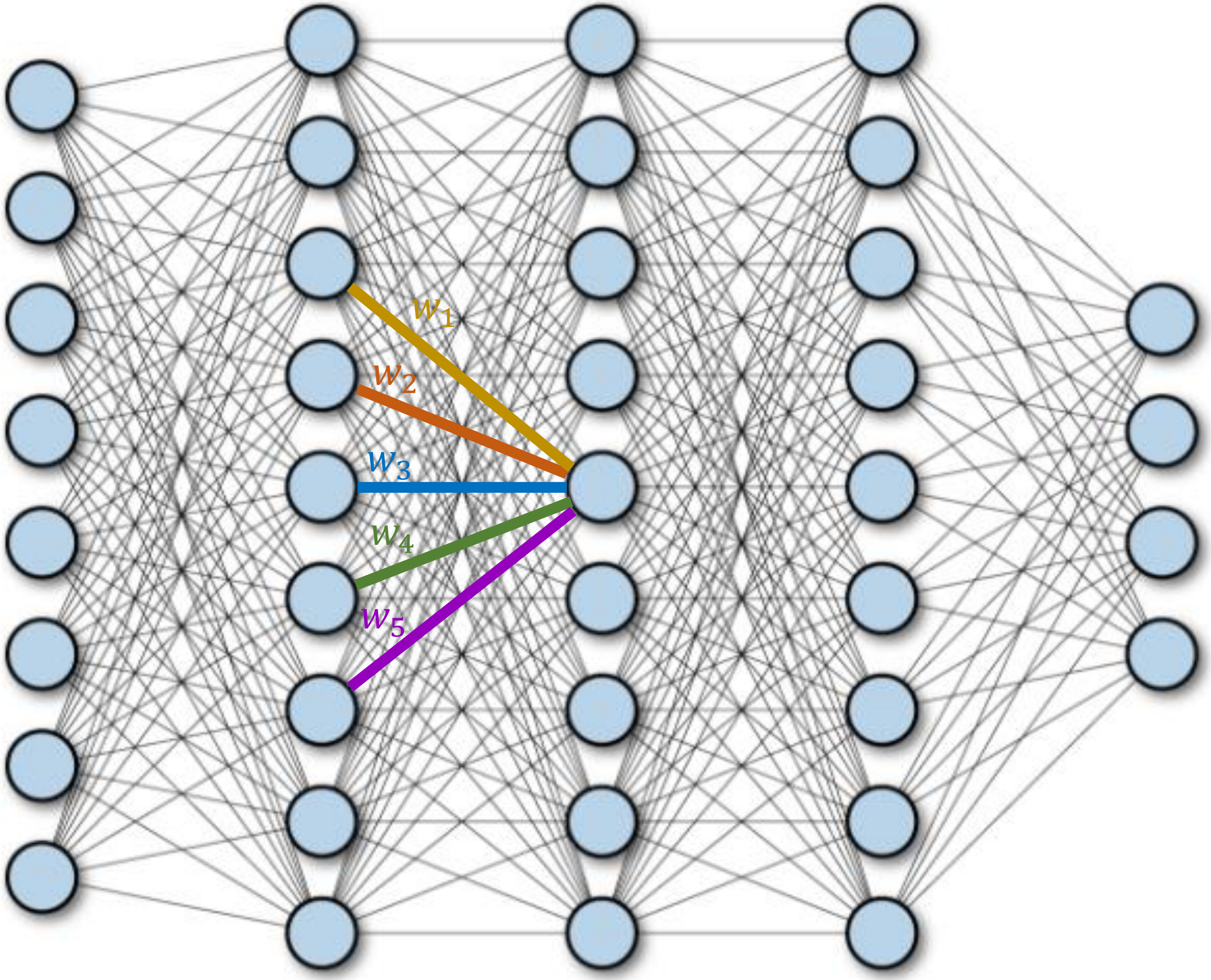
Convolutional Neural Networks

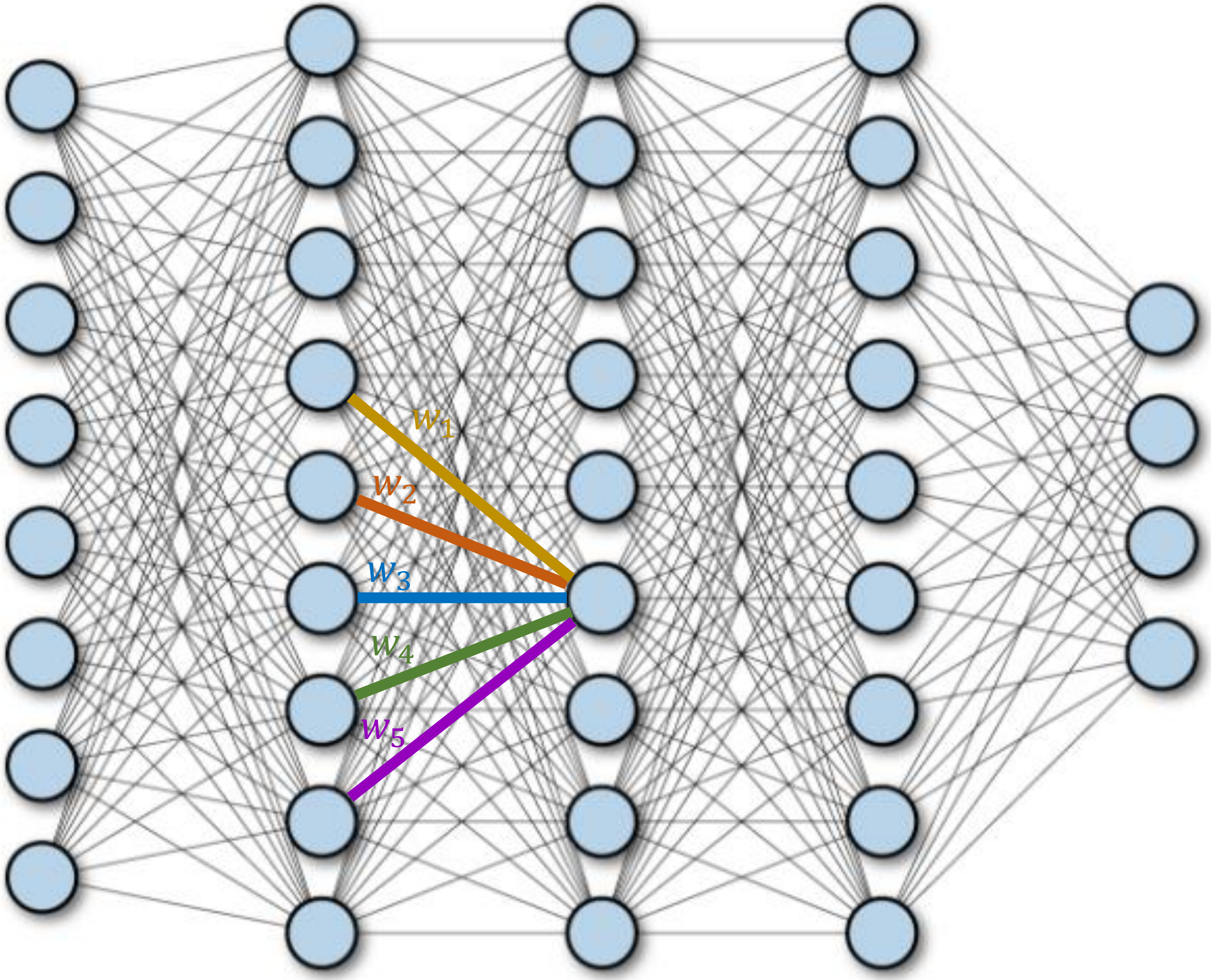


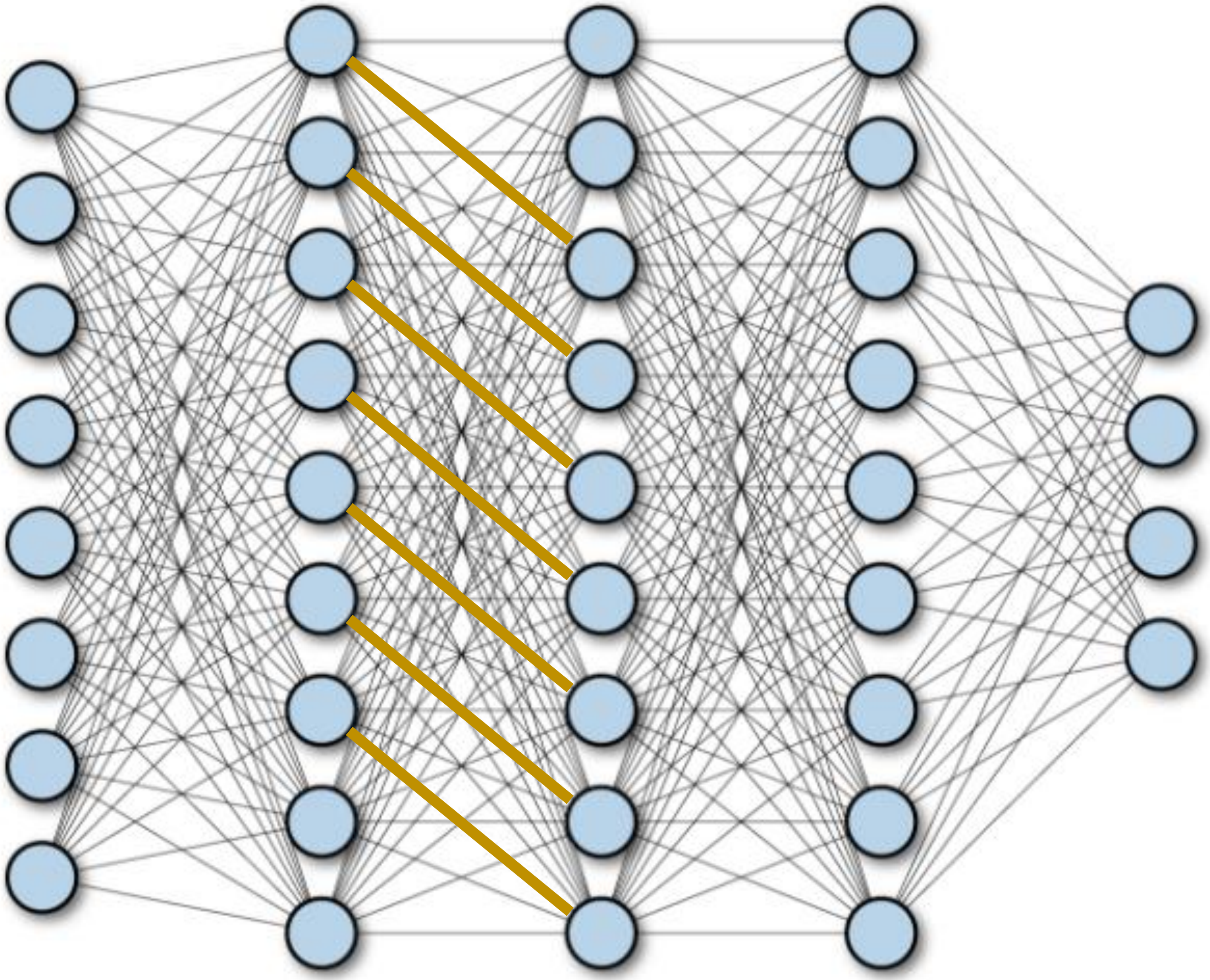


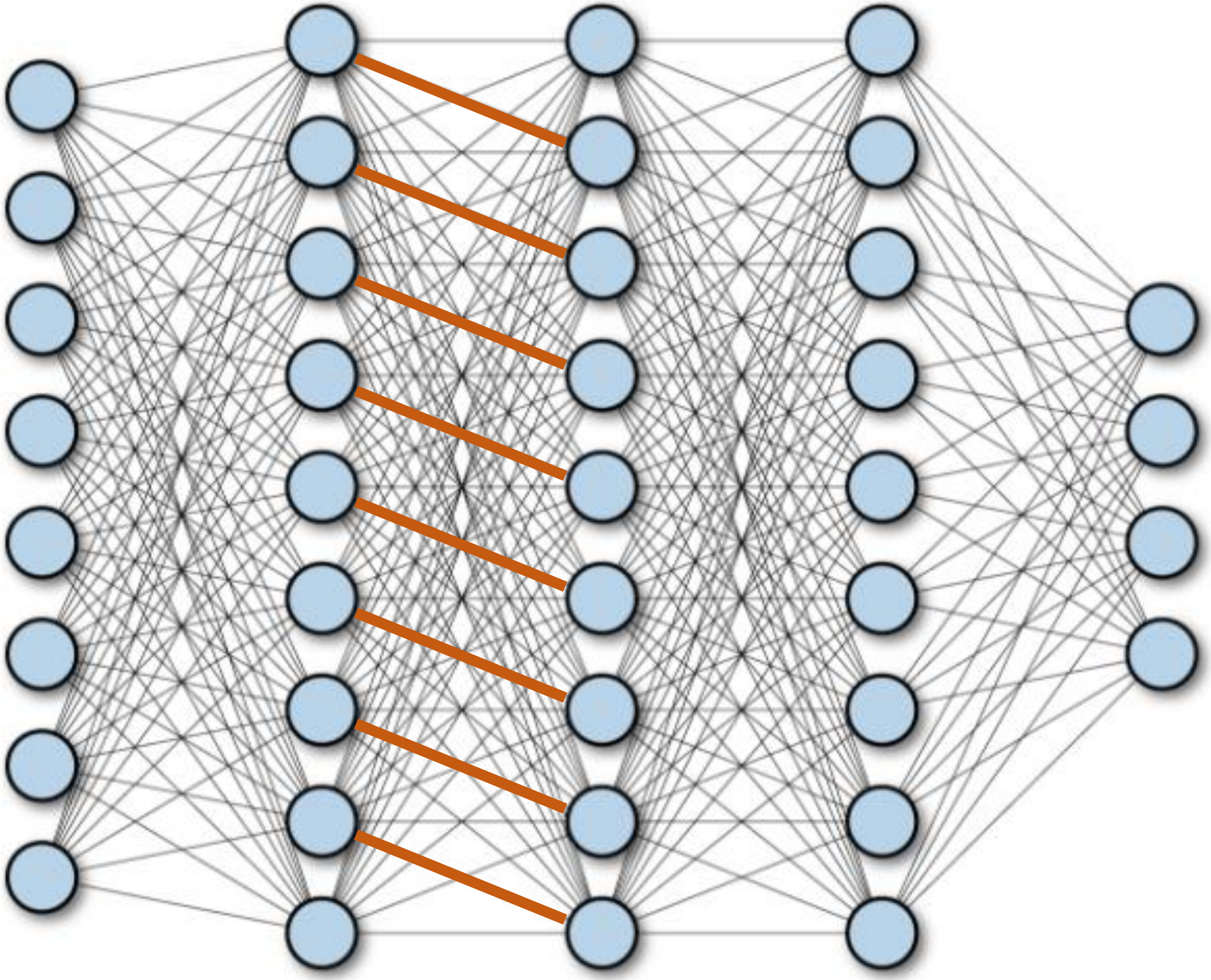
w_1
w_2
w_3
w_4
w_5

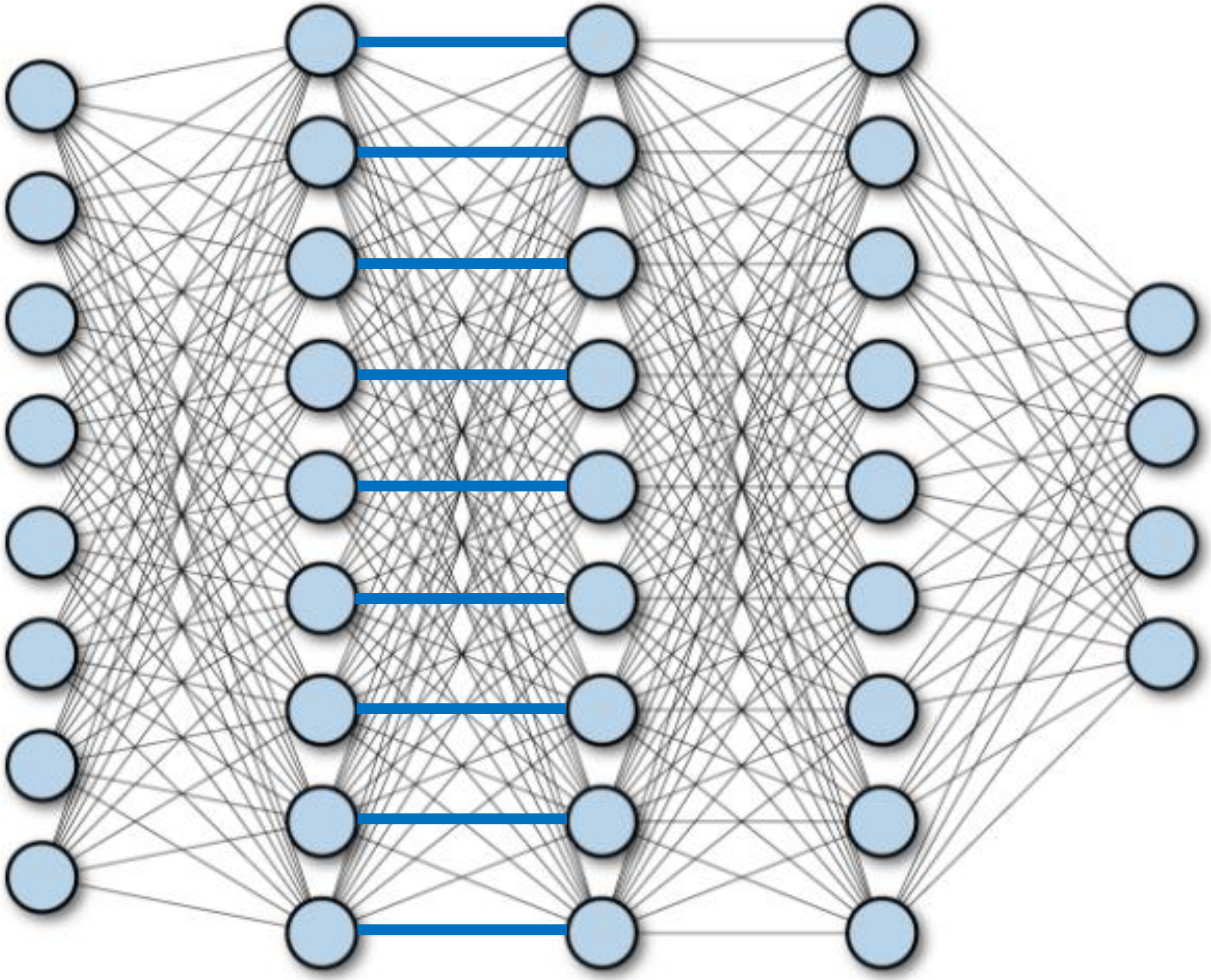

w_1
w_2
w_3
w_4
w_5

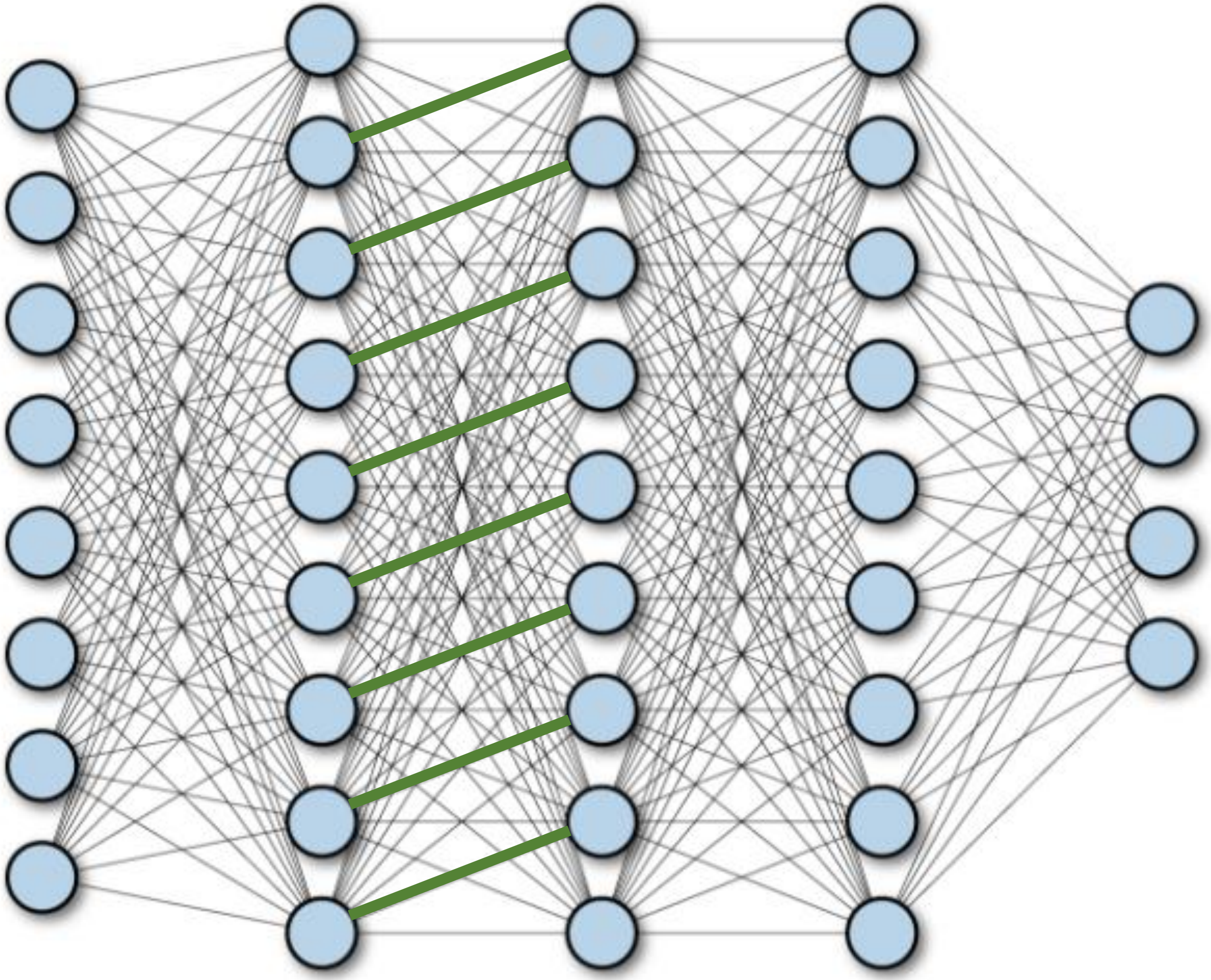

w_1
w_2
w_3
w_4
w_5

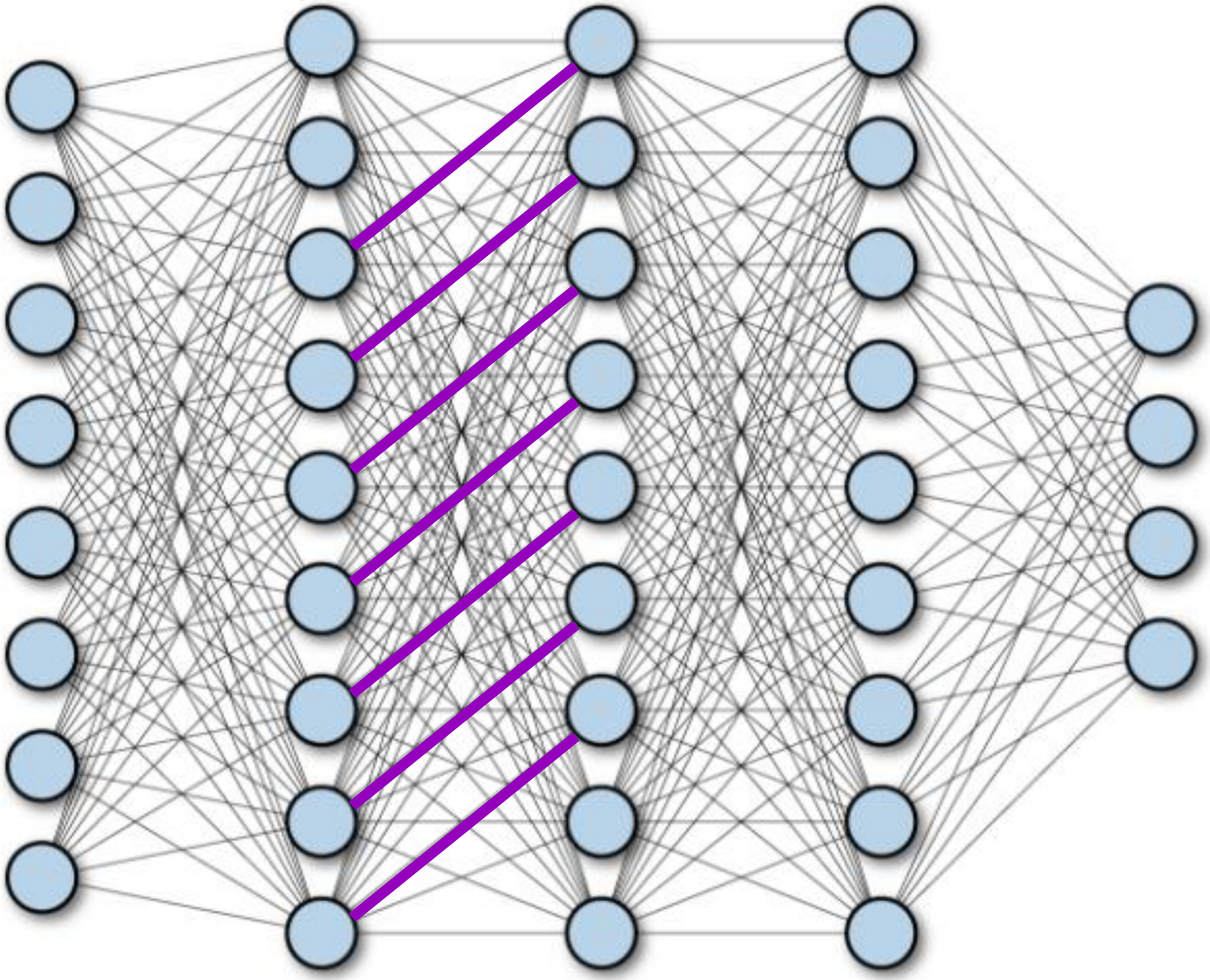

w_1
w_2
w_3
w_4
w_5



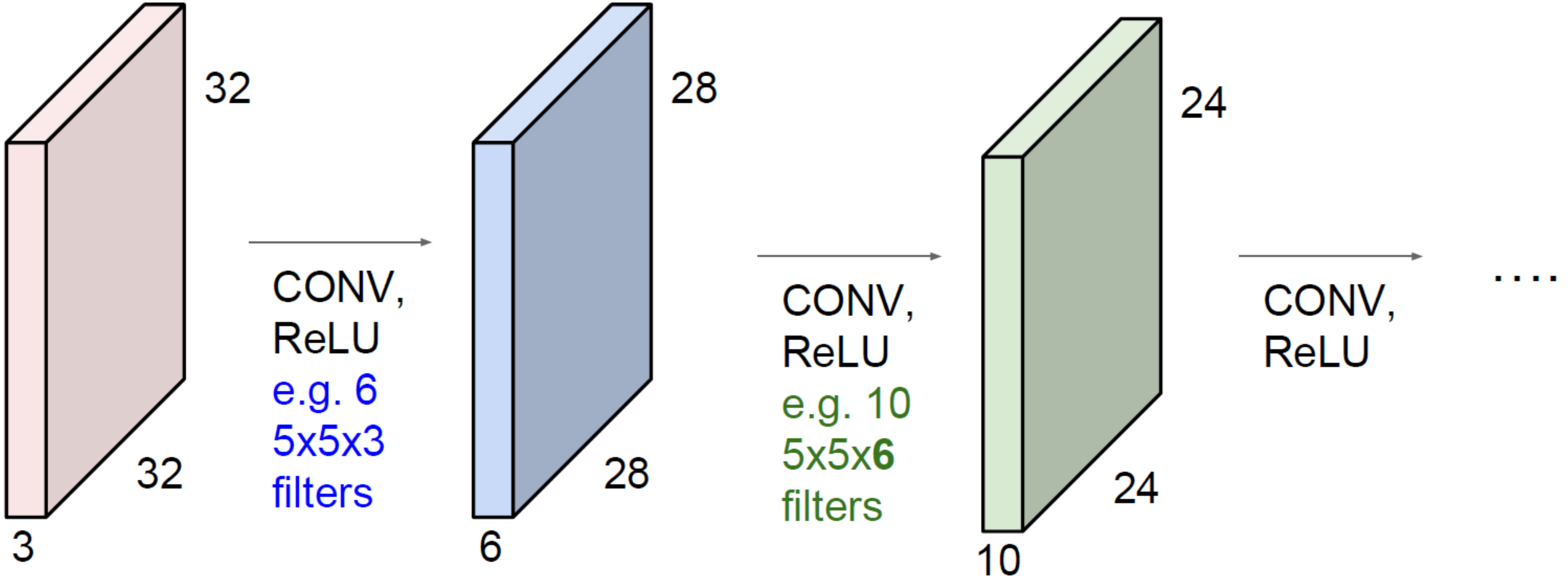






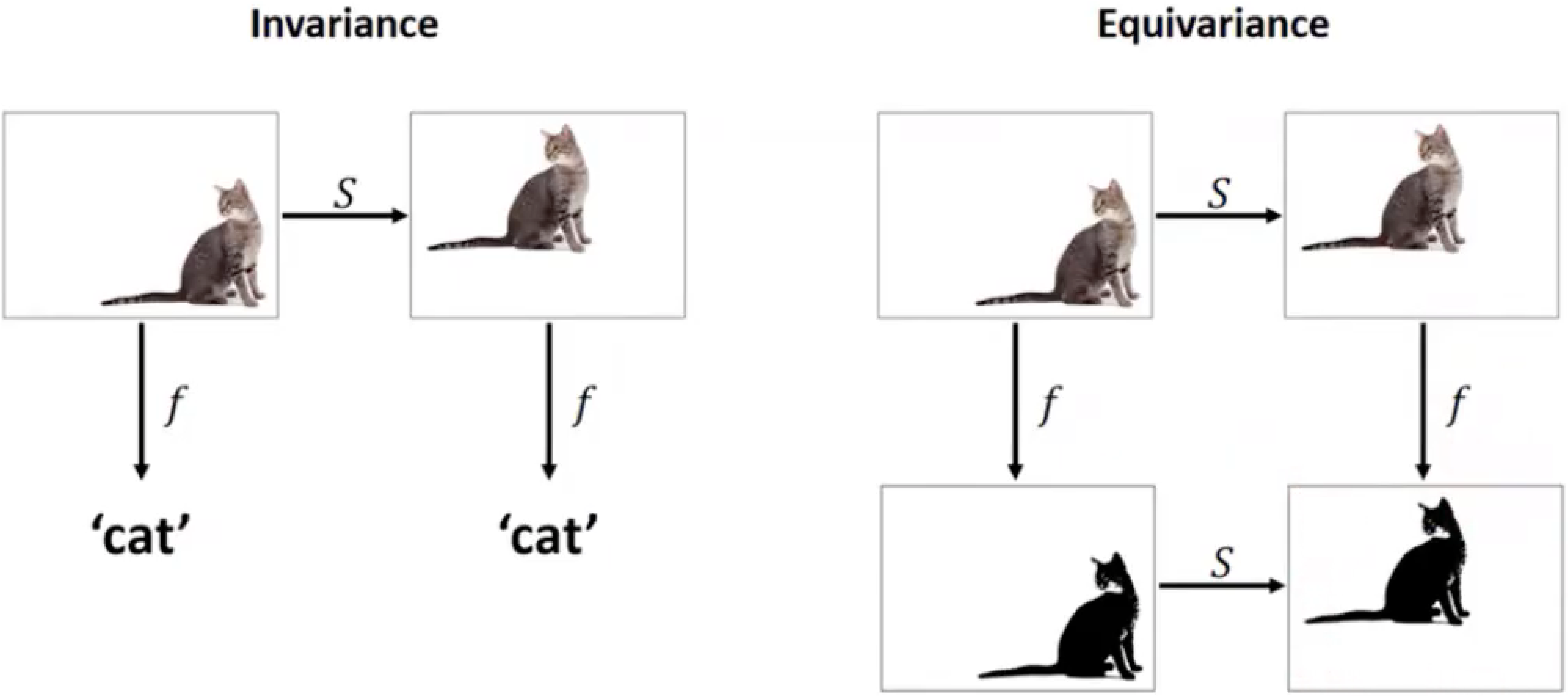


Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



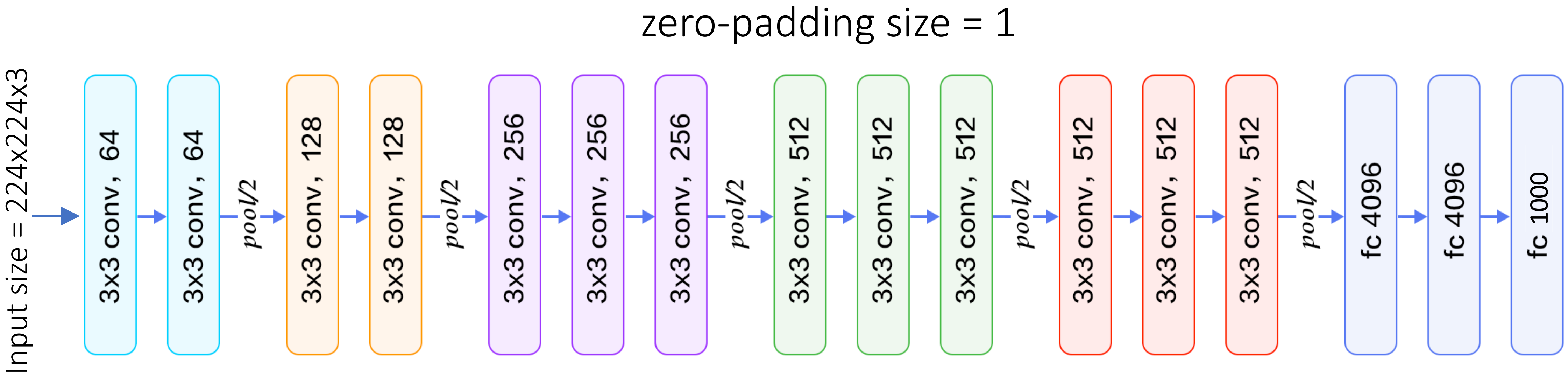
Associativity $(f * g) * h = f * (g * h)$

Invariance vs equivariance



Bernhard Kainz – Imperials deep learning course
http://wp.doc.ic.ac.uk/bkainz/teaching/70010-deep-learning/
https://www.youtube.com/watch?v=a4Quhf9NhMY&t=944s

Case study: VGG16 for ImageNet classification

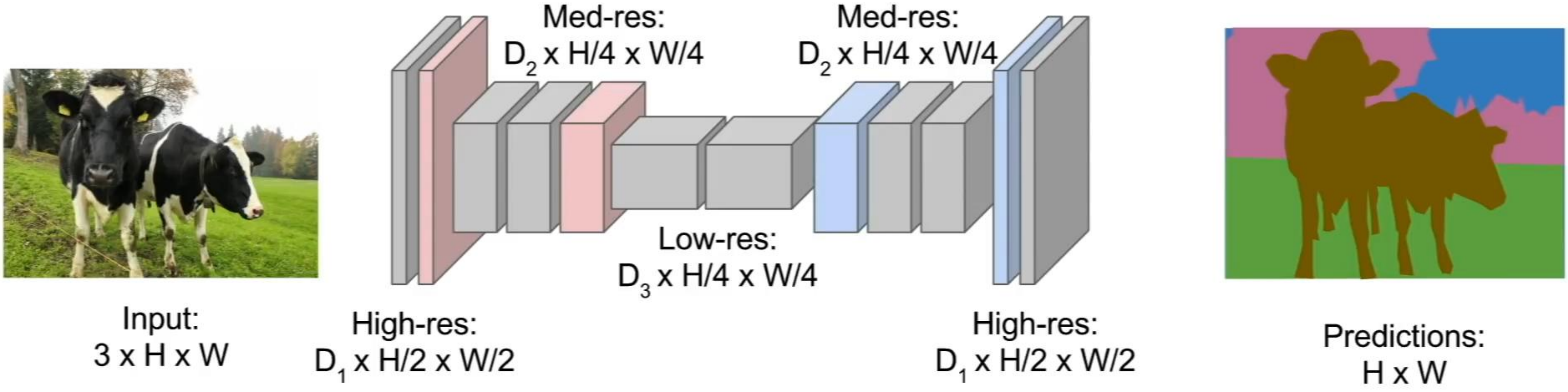


Invariant with respect to translation?

Equivariant with respect to translation?

Semantic Segmentation Idea: Fully Convolutional

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!



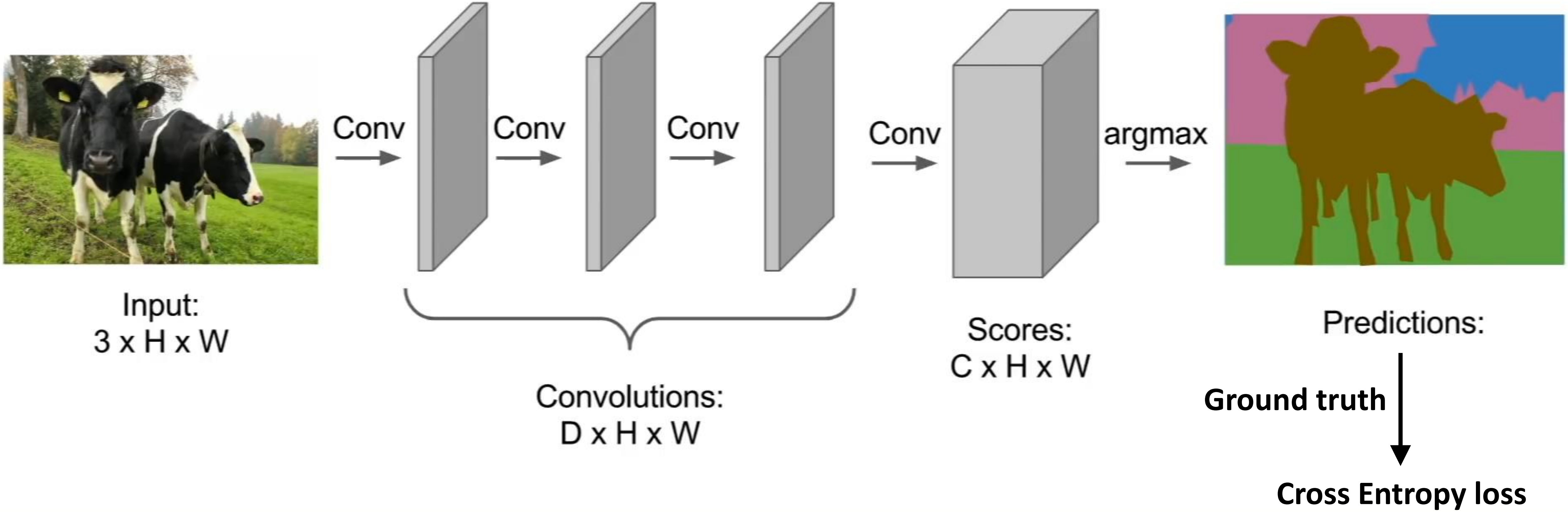
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Invariant with respect to translation?

Equivariant with respect to translation?

Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers
to make predictions for pixels all at once!



Equivariant with respect to translation!