CNNs

January 31st, 2025

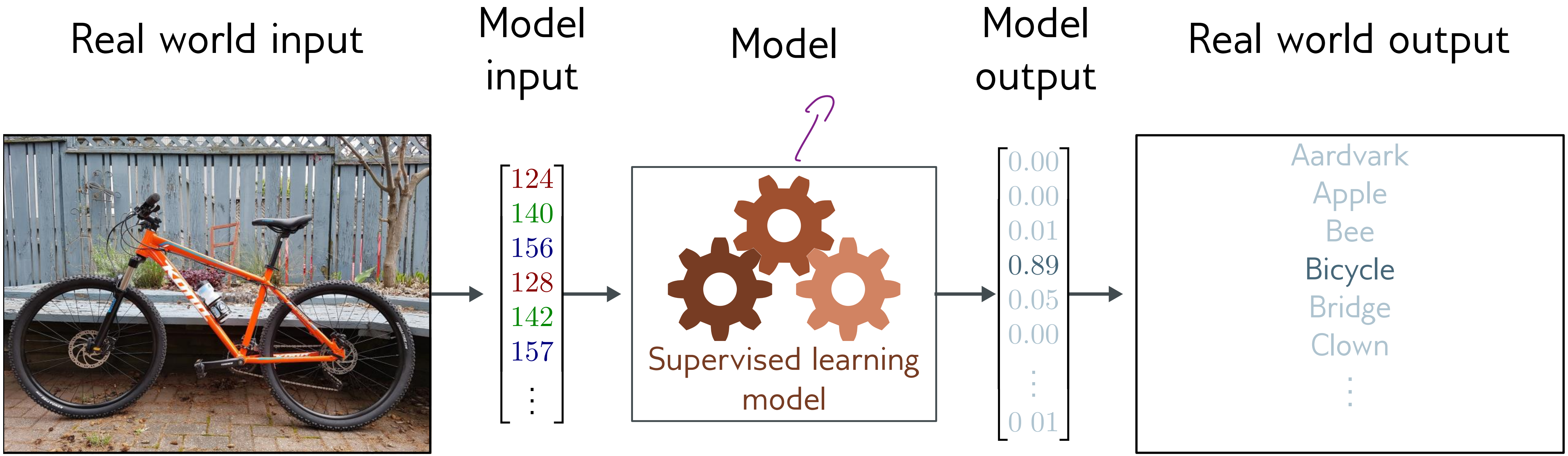
Deep Learning (CS60010)

**Slides adapted from http://udlbook.com*

Convolutional networks

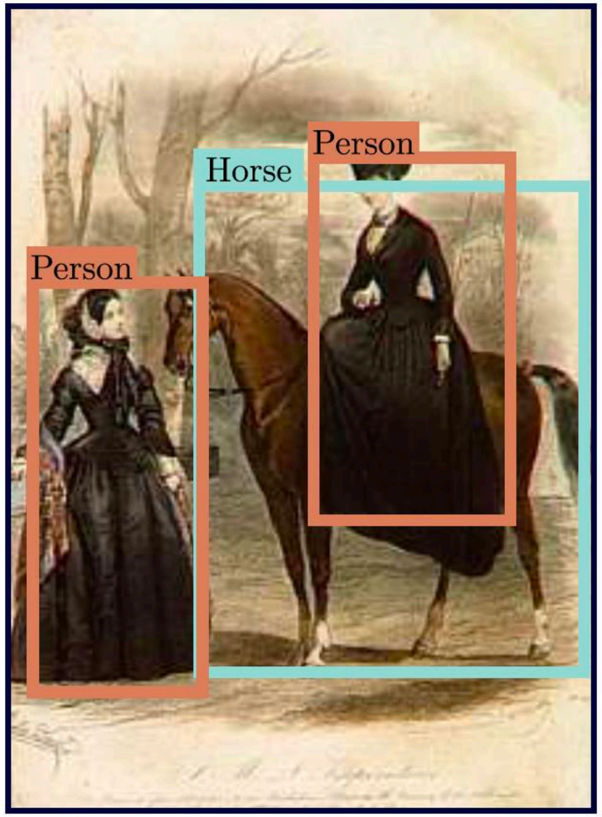
- Networks for images
- Invariance and equivariance
- 1D convolution
- Convolutional layers
- Channels
- Receptive fields
- Convolutional network for MNIST 1D

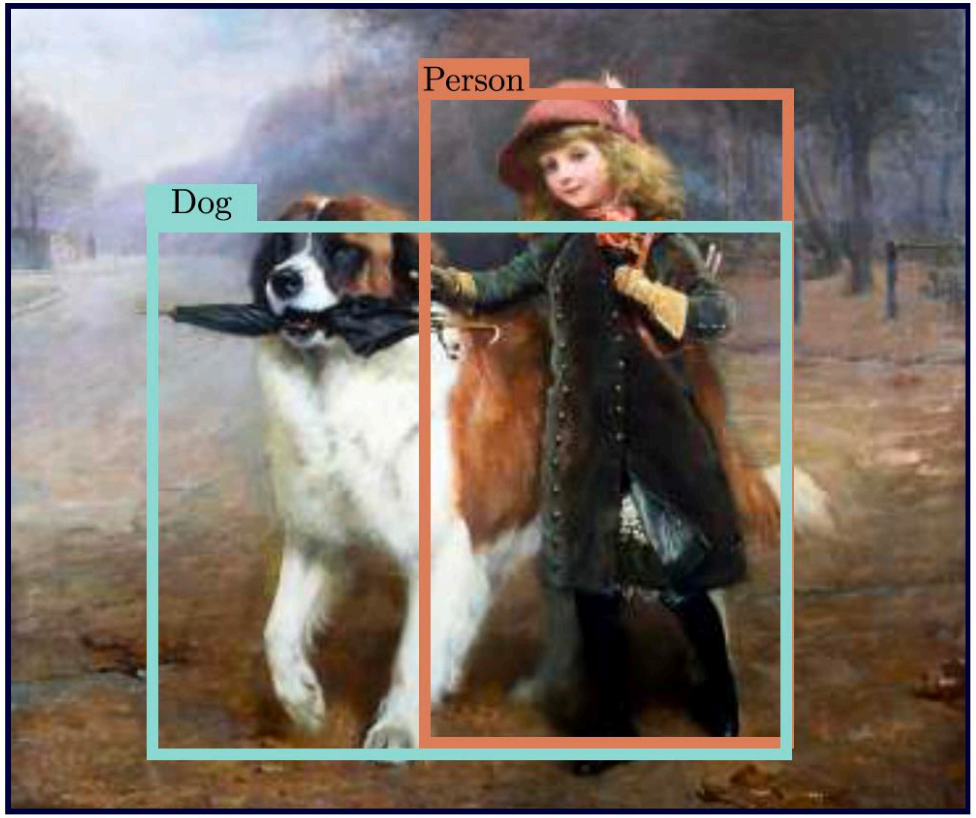
Image classification



- Multiclass classification problem (discrete classes, >2 possible classes)
- Convolutional network

Object detection





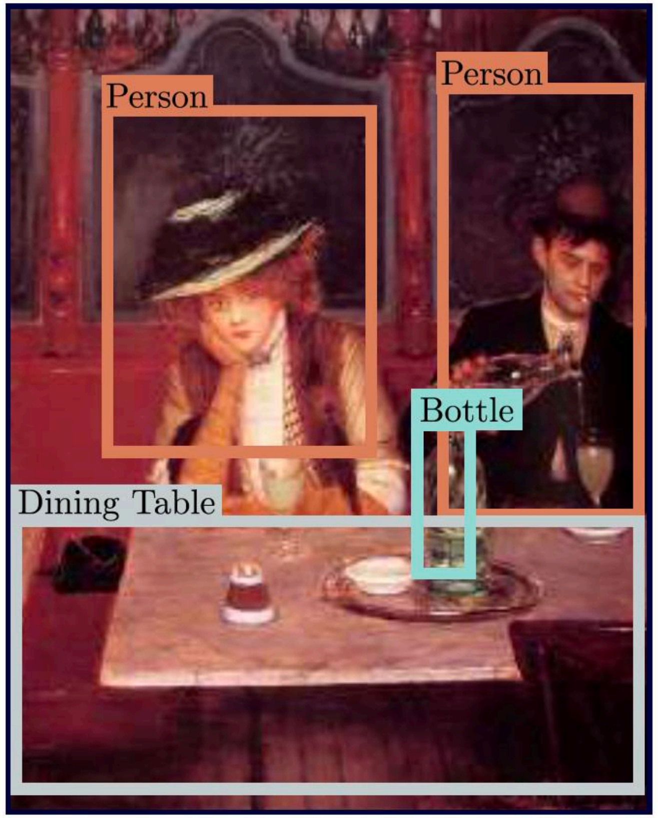
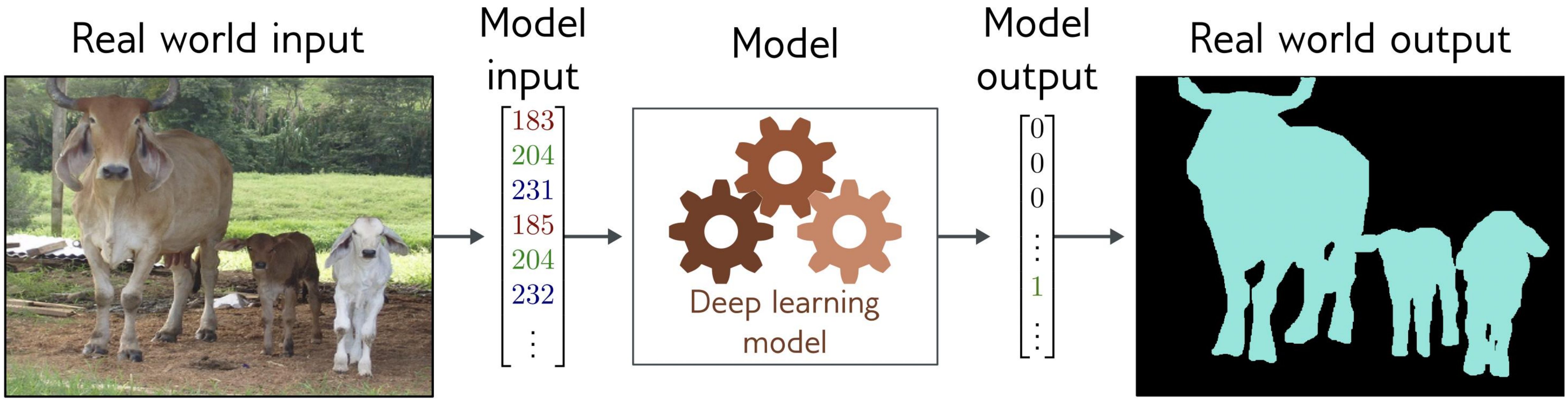


Image segmentation



- Multivariate binary classification problem (many outputs, two discrete classes)
- Convolutional encoder-decoder network

Networks for images

$\underbrace{224 \times 224}_{\text{pixels}}$ image $\times$ 3 channels

each value (0, 255)

- Problems with fully-connected networks

1. Size
   - 224x224 RGB image = 150,528 dimensions
   - Hidden layers generally larger than inputs
   - One hidden layer = 150,520x150,528 weights -- 22 billion
2. Nearby pixels statistically related
   - But could permute pixels and relearn and get same results with FC
3. Should be stable under transformations
   - Don't want to re-learn appearance at different parts of image

Convolutional networks

- Parameters only look at local image patches
- Share parameters across image

Convolutional networks

- Networks for images
- Invariance and equivariance
- 1D convolution
- Convolutional layers
- Channels
- Receptive fields
- Convolutional network for MNIST 1D

Invariance

$\boxed{\text{shift}}$

- A function f[x] is invariant to a transformation t[] if:

$$\mathbf{f}[\mathbf{t}[\mathbf{x}]] = \mathbf{f}[\mathbf{x}]$$

i.e., the function output is the same even after the transformation is applied.

Invariance example

e.g., Image classification

- Image has been translated, but we want our classifier to give the same result

Equivariance

- A function f[x] is equivariant to a transformation t[] if:

$$\mathbf{f}\left[\mathbf{t}[\mathbf{x}]\right] = \mathbf{t}\left[\mathbf{f}[\mathbf{x}]\right]$$

i.e., the output is transformed in the same way as the input

Equivariance example

e.g., Image segmentation

- Image has been translated and we want segmentation to translate with it

Convolutional networks

- Networks for images
- Invariance and equivariance
- 1D convolution
- Convolutional layers
- Channels
- Receptive fields
- Convolutional network for MNIST 1D

Convolution* in 1D

$\omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}$

$z_i =$

- Input vector **x**:

$$\mathbf{x} = [x_1, x_2, \ldots x_i \ldots, x_I]$$

- Output is weighted sum of neighbors:

$$z_i = \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}$$

- Convolutional kernel or filter:

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$$

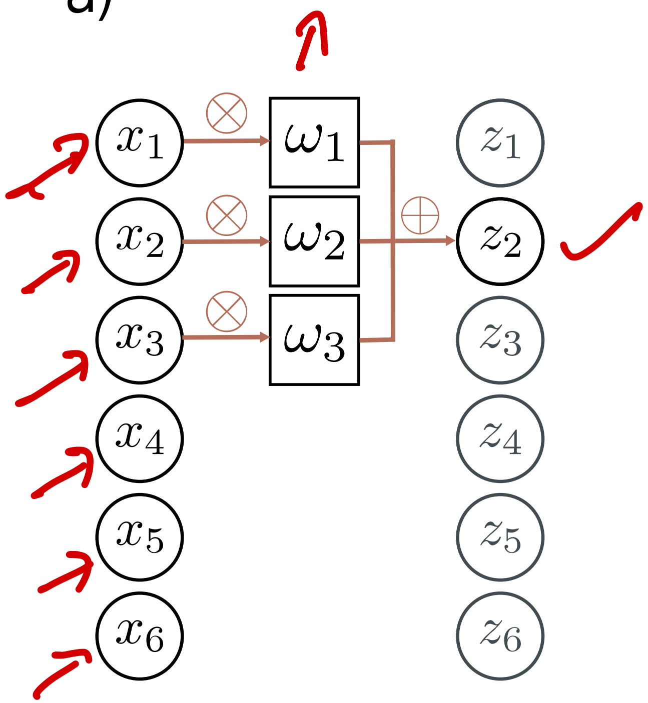
Kernel size = 3

* Not really technically convolution

Convolution with kernel size 3

odd size

a)



x_{-1}

x_0

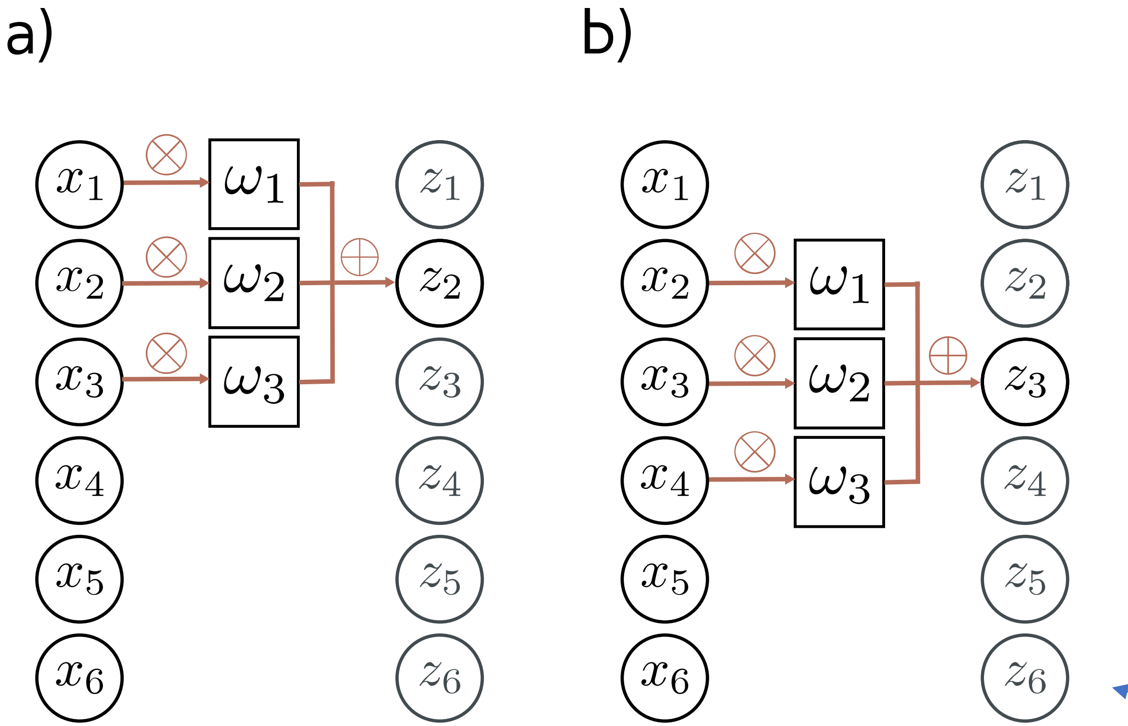
x_1

x_2 - - - $\boxed{w_1 \; w_2 \; w_3}$ z_2

x_3

x_4

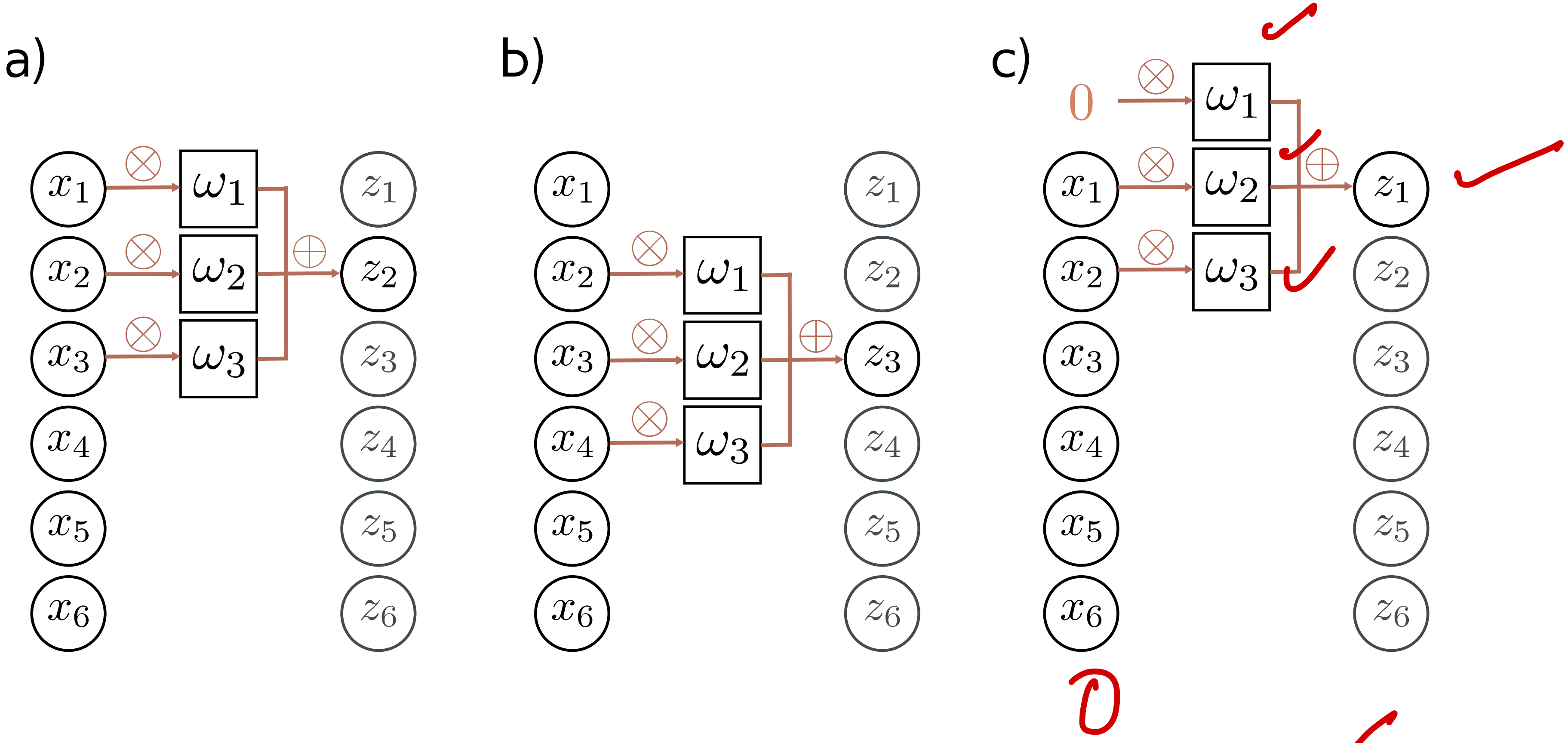
Convolution with kernel size 3



Equivariant to translation of input

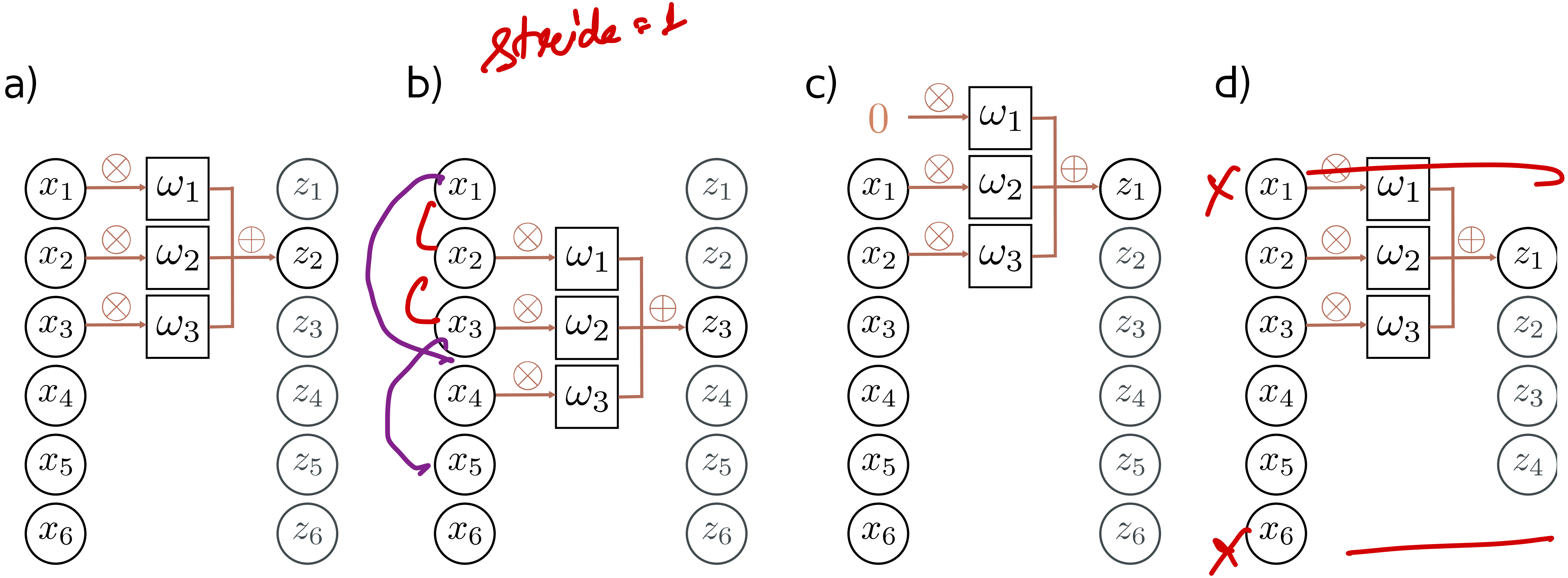
$$\mathbf{f}[\mathbf{t}[\mathbf{x}]] = \mathbf{t}[\mathbf{f}[\mathbf{x}]]$$

Zero padding



Treat positions that are beyond end of the input as zero.

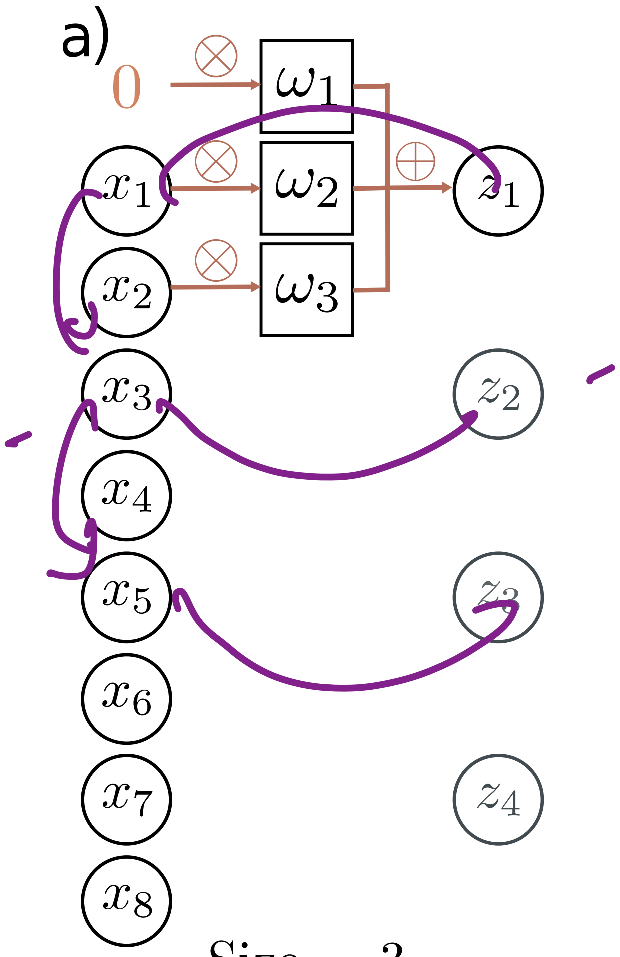
“Valid” convolutions



Only process positions where kernel falls in image (smaller output).

Stride, kernel size, and dilation

- Stride = shift by k positions for each output
  - Decreases size of output relative to input
- Kernel size = weight a different number of inputs for each output
  - Combine information from a larger area
  - But kernel size 5 uses 5 parameters
- Dilated convolutions = intersperse kernel values with zeros
  - Combine information from a larger area
  - Fewer parameters



Size $=3$

Stride $=2$

Dilation $=1$

Kernel size $=3$

Stride $=2$

dilation $=1$

dilation 2

w_1

0

w_2

0

w_3

dilation 3

w_1

0

0

w_2

0

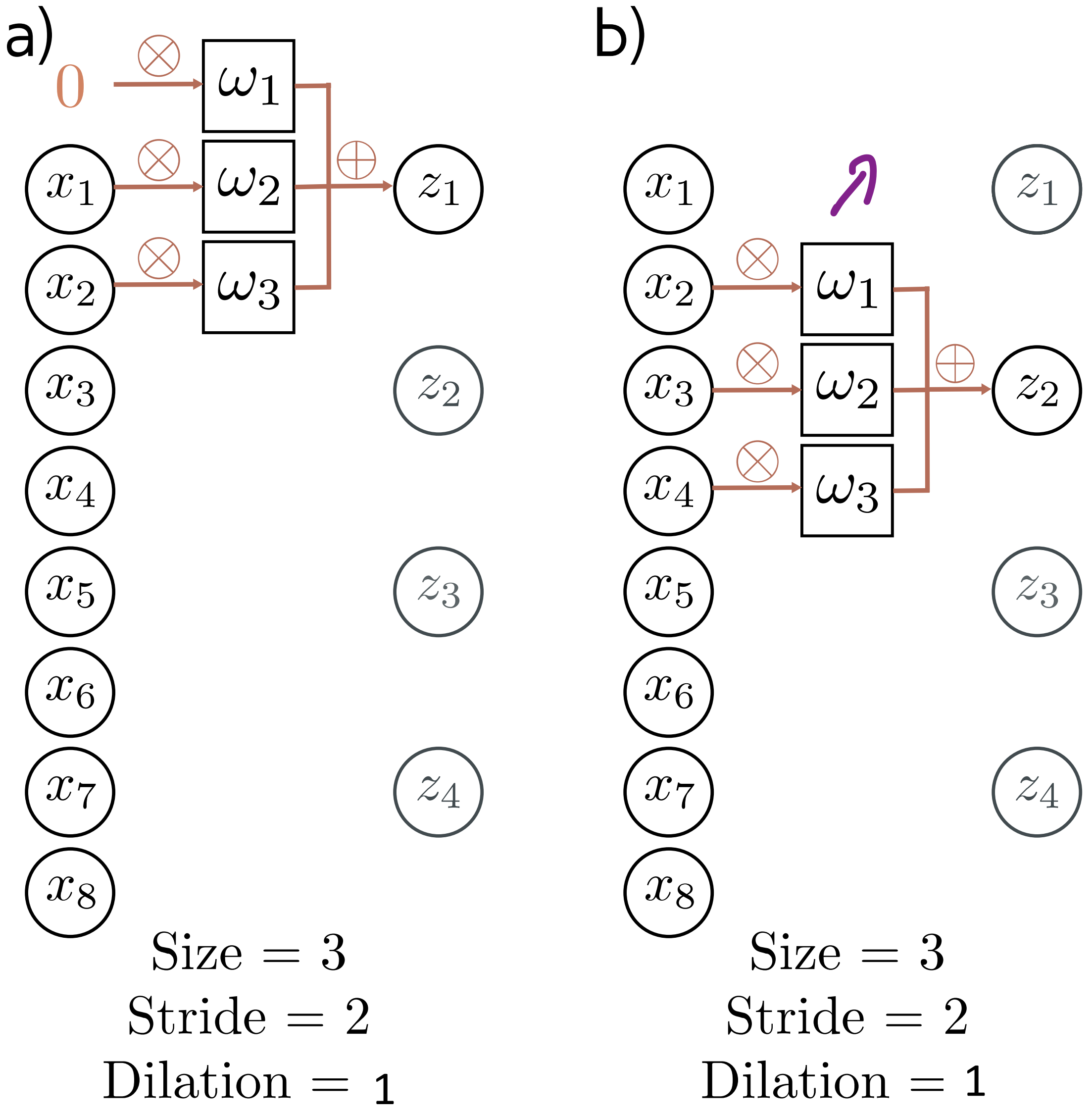
0

w_3

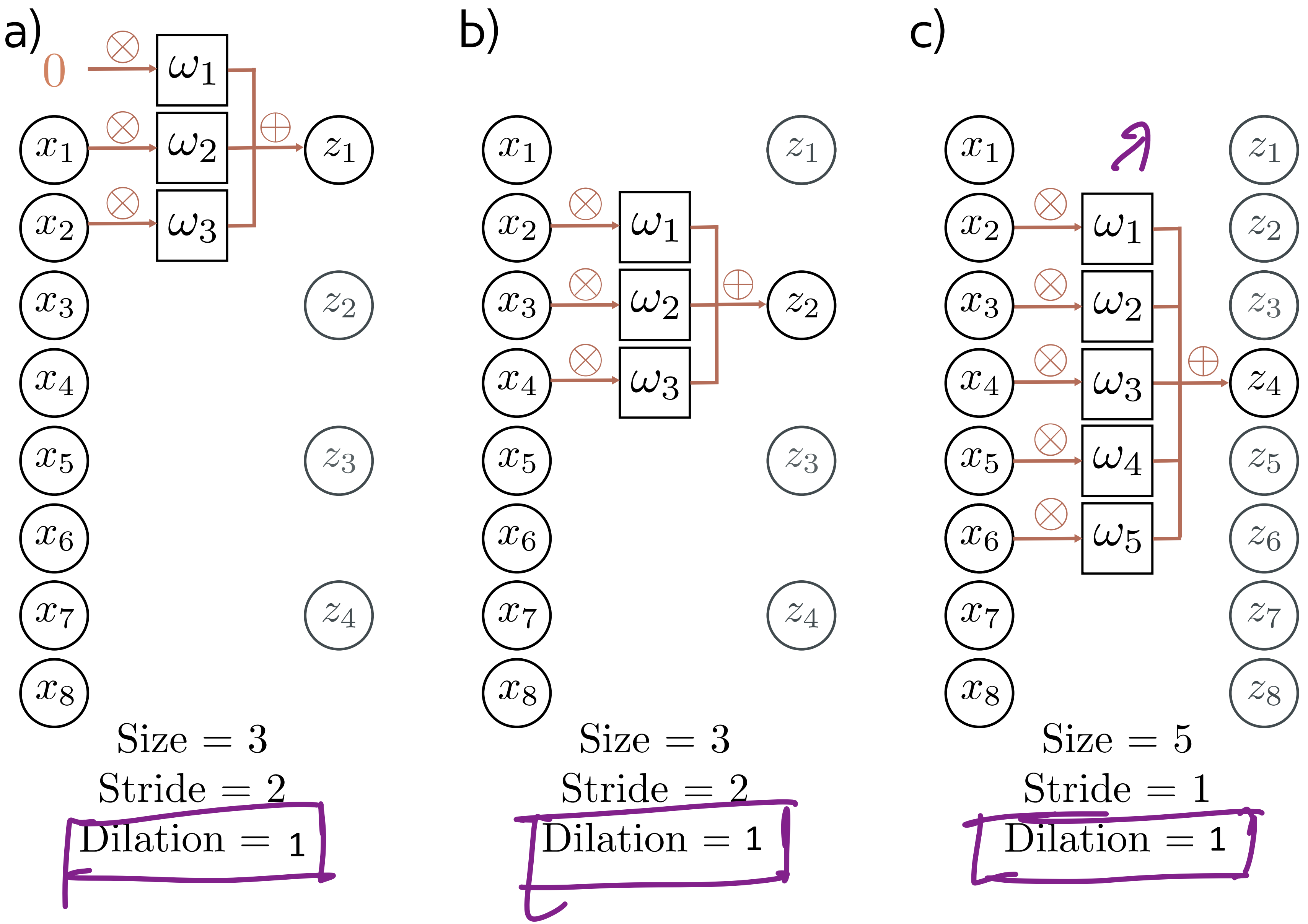
0

0

w_4

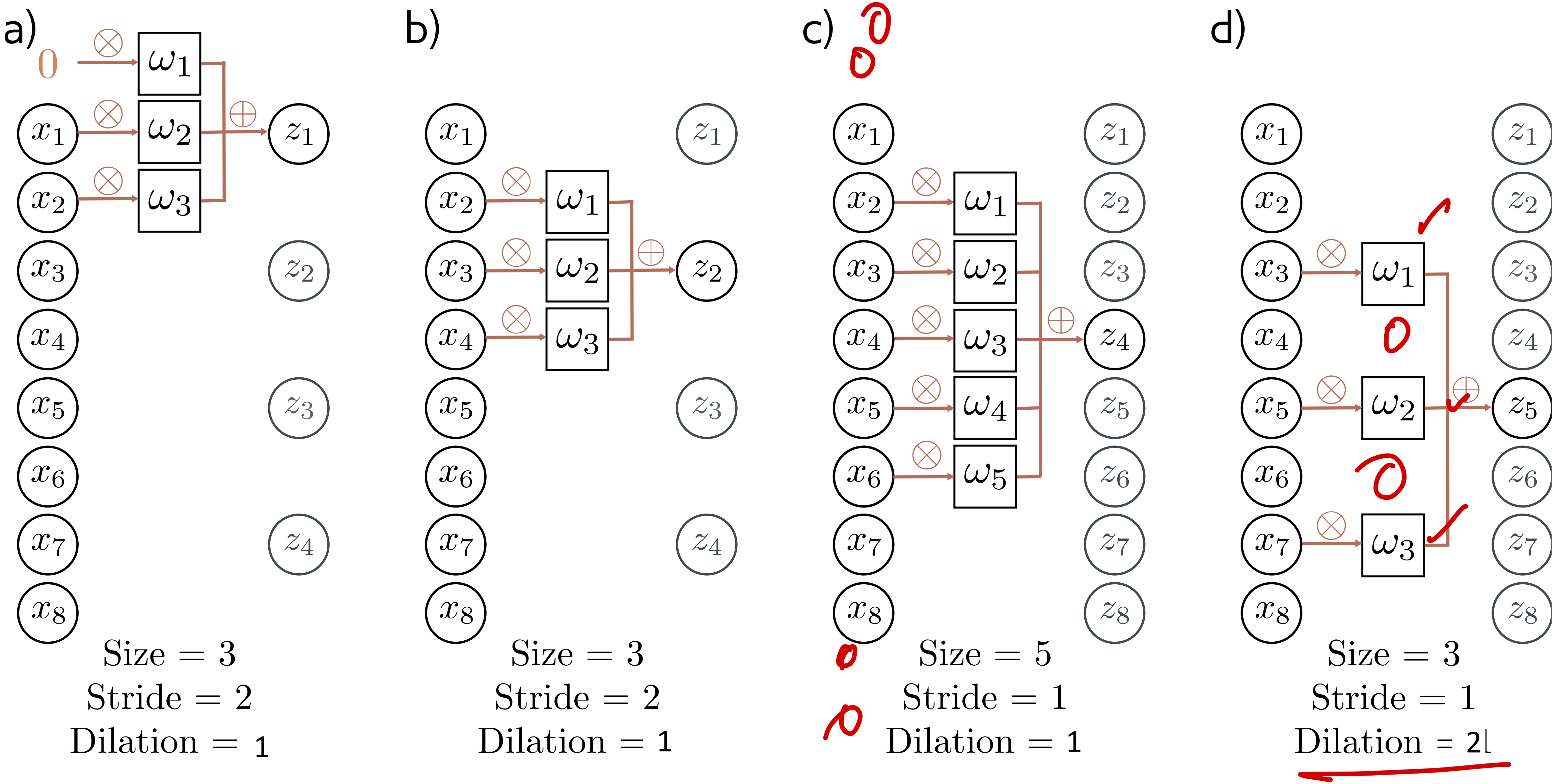


a)
0
ω_1
ω_2
ω_3
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
z_1
z_2
z_3
z_4
Size = 3
Stride = 2
Dilation = 1
b)
ω_1
ω_2
ω_3
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
z_1
z_2
z_3
z_4
Size = 3
Stride = 2
Dilation = 1



a)
0
ω_1
ω_2
ω_3
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
z_1 z_2 z_3 z_4
Size = 3
Stride = 2
Dilation = 1
b)
ω_1
ω_2
ω_3
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
z_1 z_2 z_3 z_4
Size = 3
Stride = 2
Dilation = 1
c)
ω_1
ω_2
ω_3
ω_4
ω_5
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8
Size = 5
Stride = 1
Dilation = 1

Zero-padding



Convolutional networks

- Networks for images
- Invariance and equivariance
- 1D convolution
- Convolutional layers
- Channels
- Receptive fields
- Convolutional network for MNIST 1D

x_1 – – x_n

FC. β

Convolutional layer

$$
\begin{aligned}
h_i &= \mathrm{a}\left[\beta + \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}\right] \\
&= \mathrm{a}\left[\beta + \sum_{j=1}^{3} \omega_j x_{i+j-2}\right]
\end{aligned}
$$

Special case of fully-connected network

Convolutional network:

$$h_i = \mathrm{a}\left[\beta + \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}\right]$$
$$= \mathrm{a}\left[\beta + \sum_{j=1}^{3} \omega_j x_{i+j-2}\right]$$

D

o/p

D

i/p

3 weights

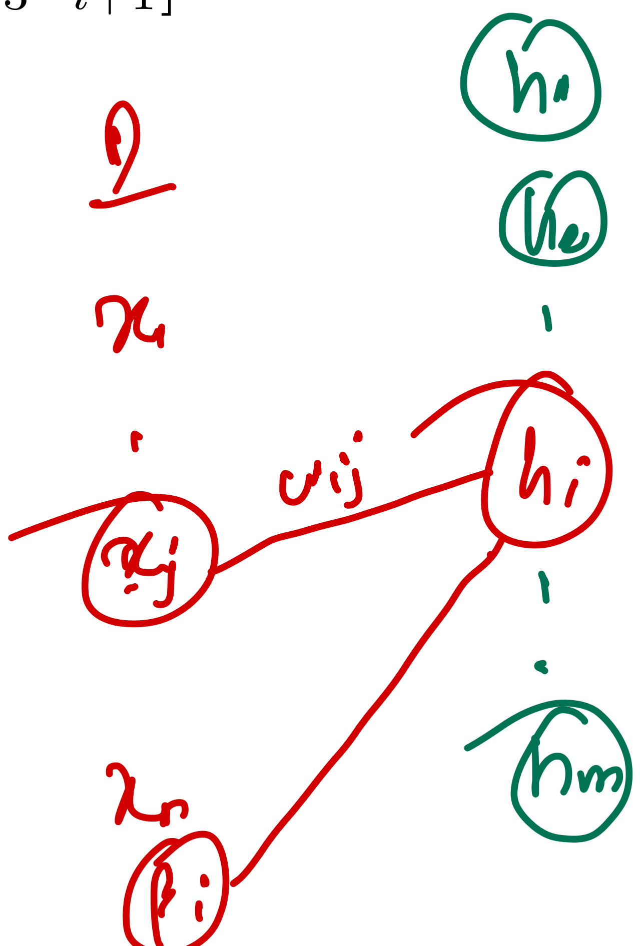
1 bias

Fully connected network:

D^2 weights

D biases

$$h_i = \mathrm{a}\left[\beta_i + \sum_{j=1}^{D} \omega_{ij} x_j\right]$$



Special case of fully-connected network

Convolutional network:

$$h_i = \mathrm{a}\left[\beta + \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}\right]$$
$$= \mathrm{a}\left[\beta + \sum_{j=1}^{3} \omega_j x_{i+j-2}\right]$$

3 weights, 1 bias

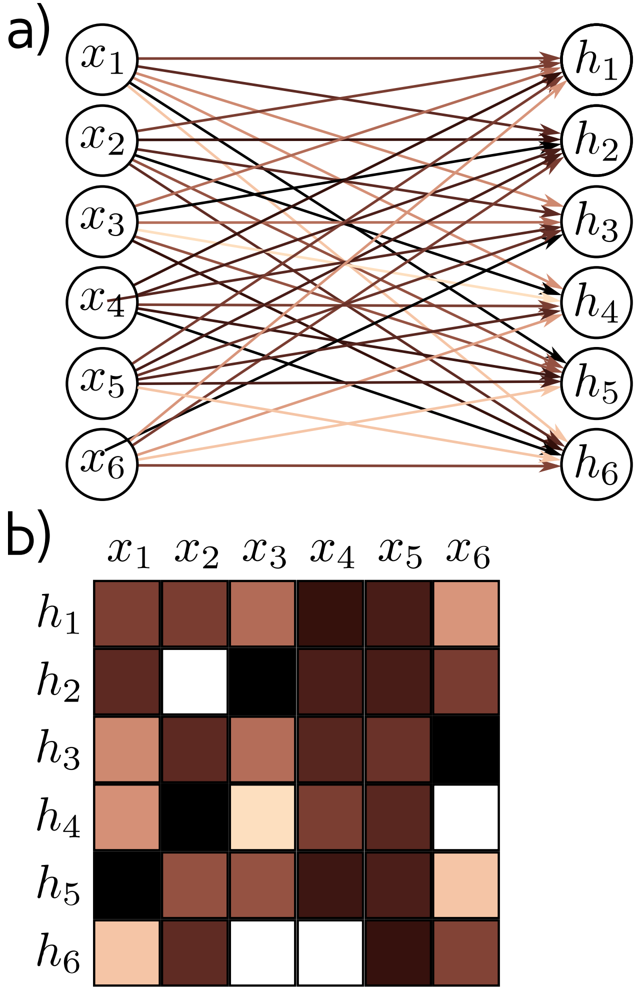
$\boxed{\text{inductive bias}}$

Fully connected network:

$$h_i = \mathrm{a}\left[\beta_i + \sum_{j=1}^{D} \omega_{ij} x_j\right]$$

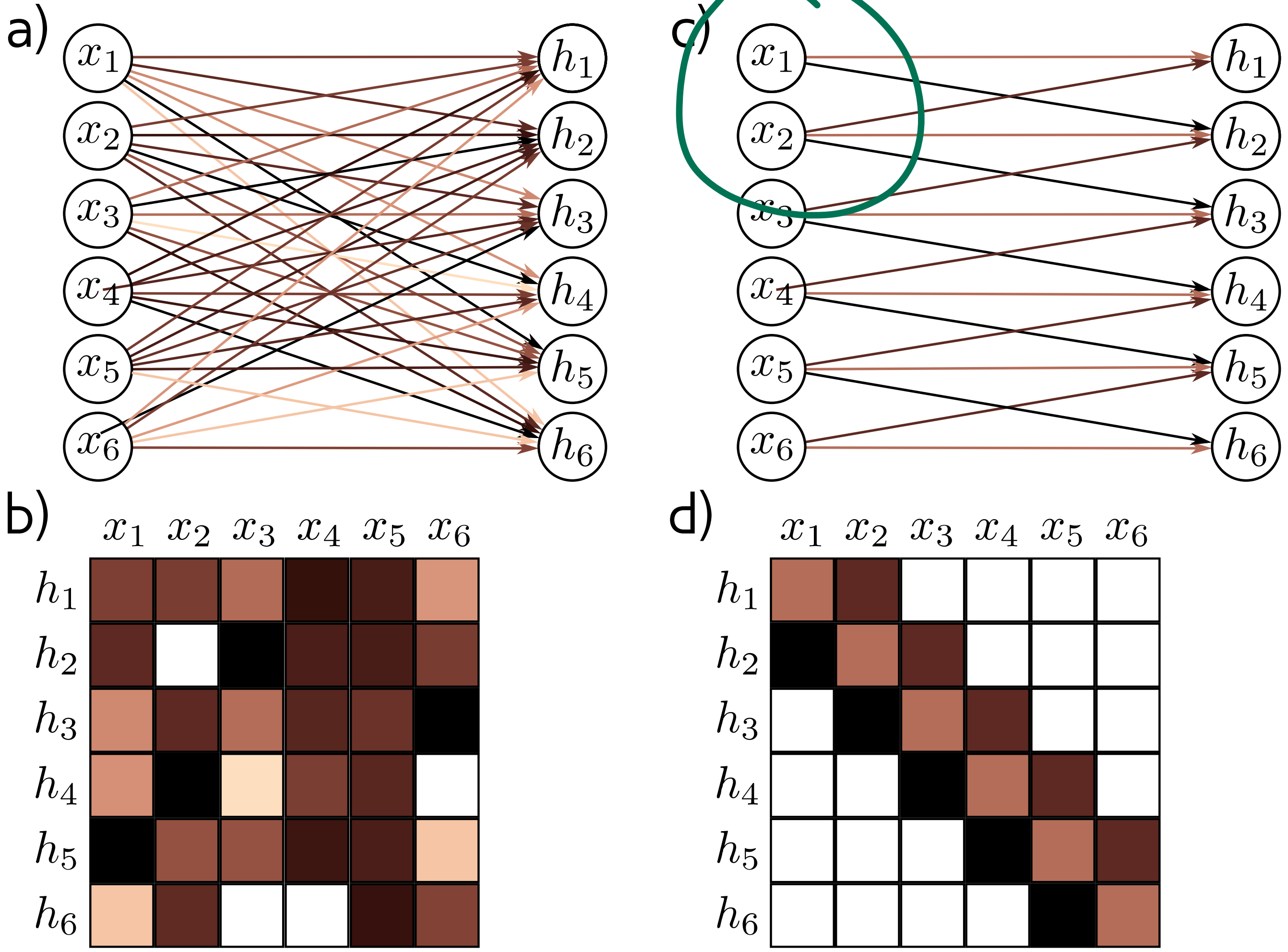
D^2 weights, D biases

Special case of fully-connected network



Fully connected network

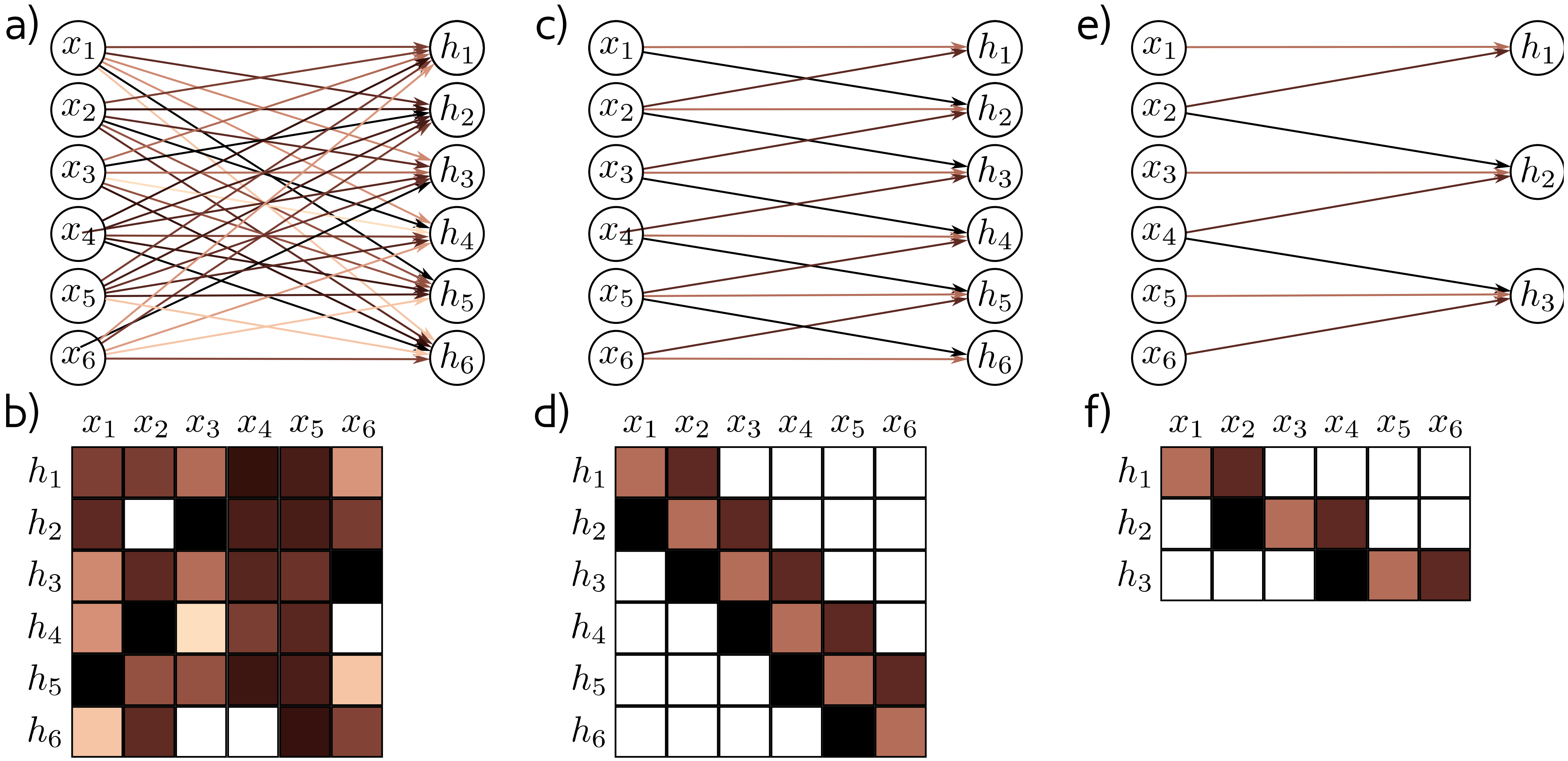
Special case of fully-connected network



Fully connected network

Convolution, kernel 3, stride 1, dilation 1

Special case of fully-connected network



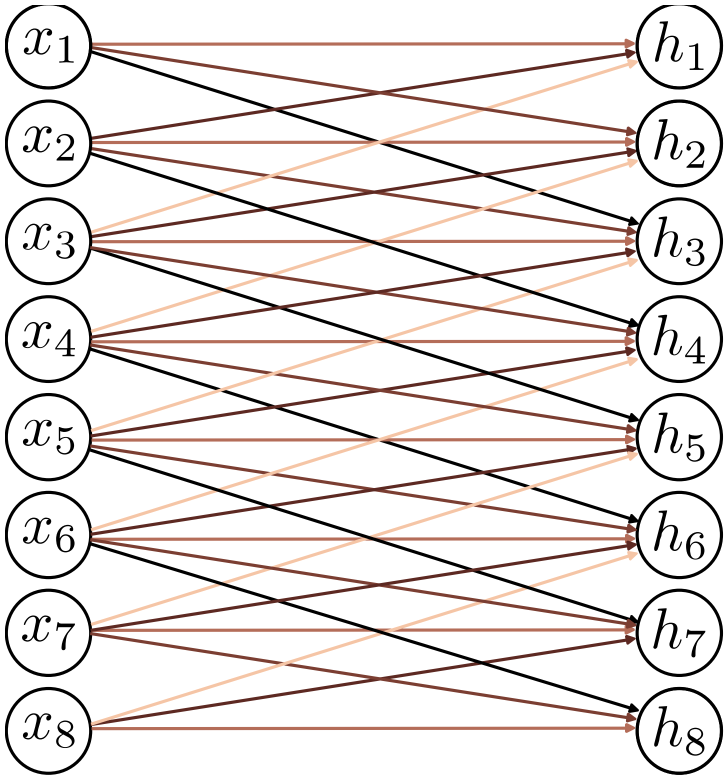
Fully connected network

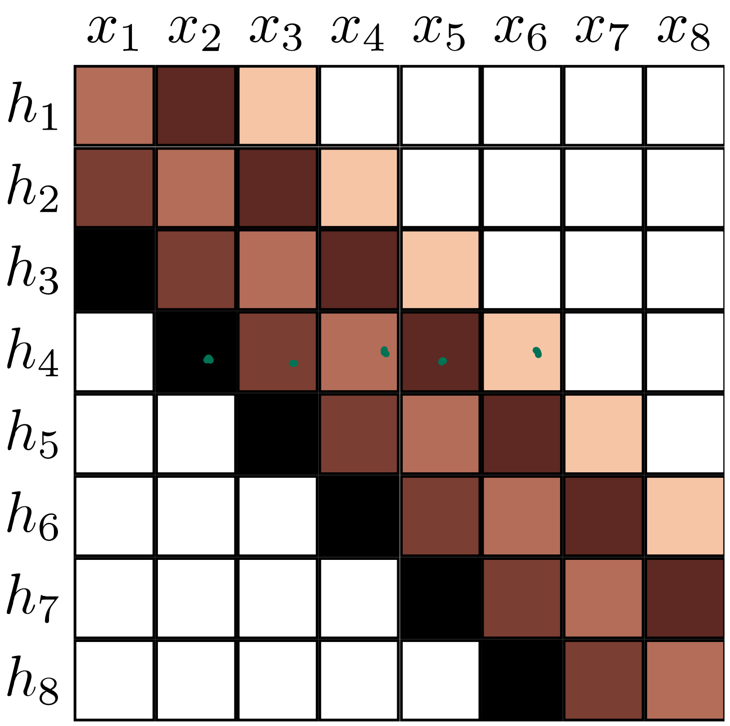
Convolution, size 3, stride 1, dilation 1, zero padding

Convolution, size 3, stride 2, dilation 1, zero padding

Question 1

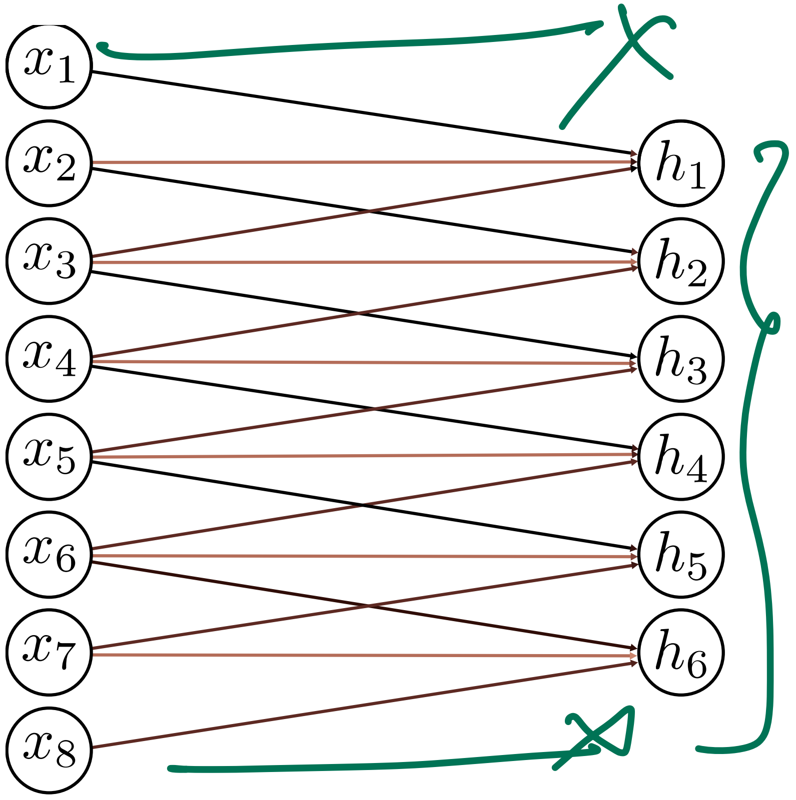
- Kernel size? 5
- Stride? 1
- Dilation? 1
- Zero padding / valid?
  ✓ ✗

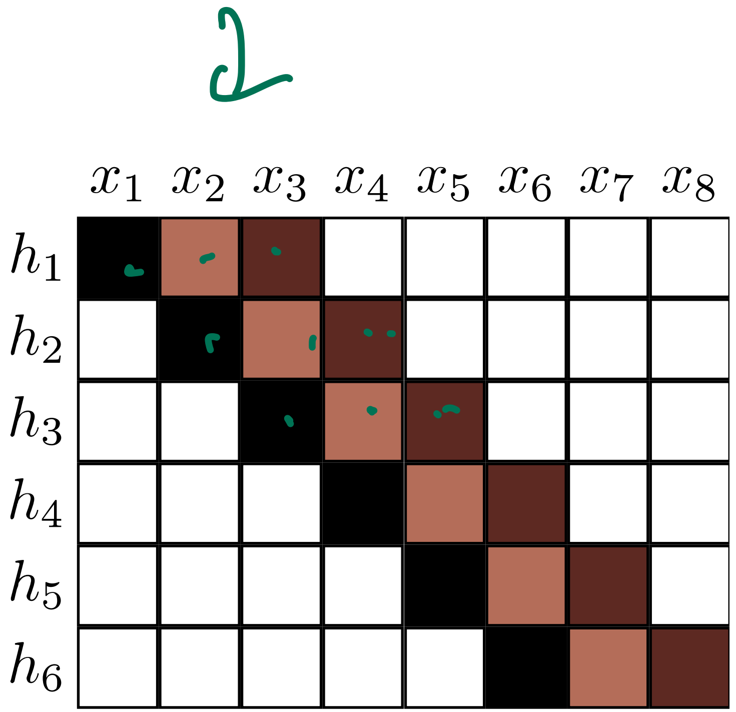




Question 2

- Kernel size? 3
- Stride? 1
- Dilation? 1
- Zero padding / valid?
  ✗ ✓

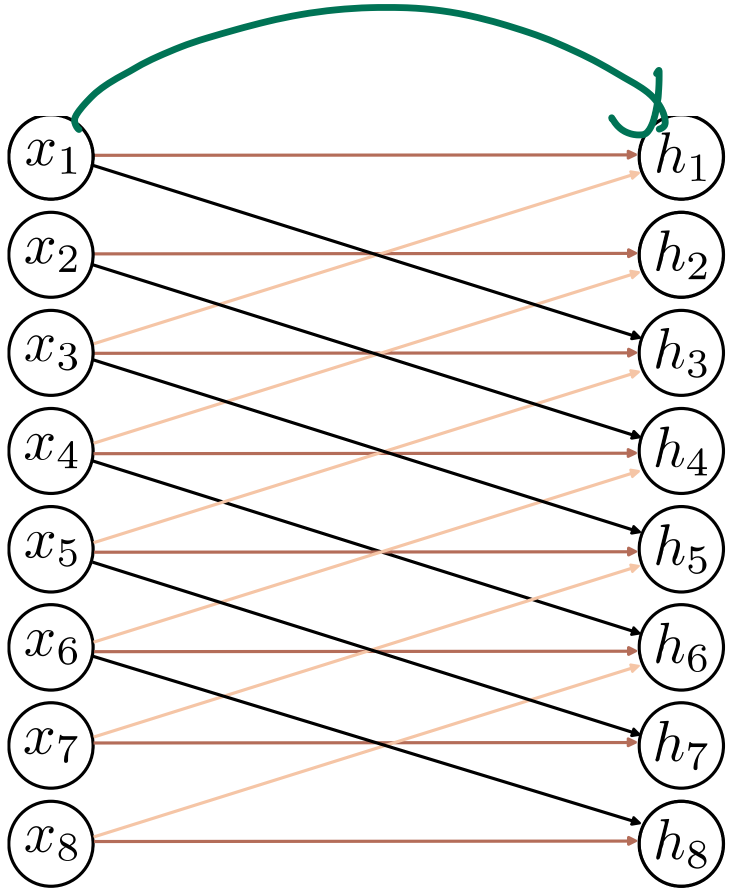


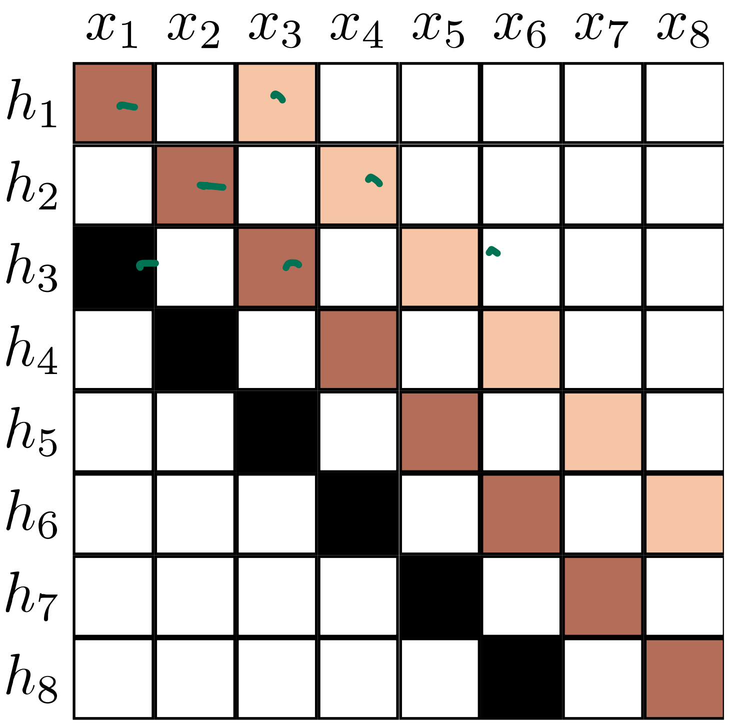


Question 3

- Kernel size? 3
- Stride? 1
- Dilation? 2
- Zero padding / valid?

✓ ✗





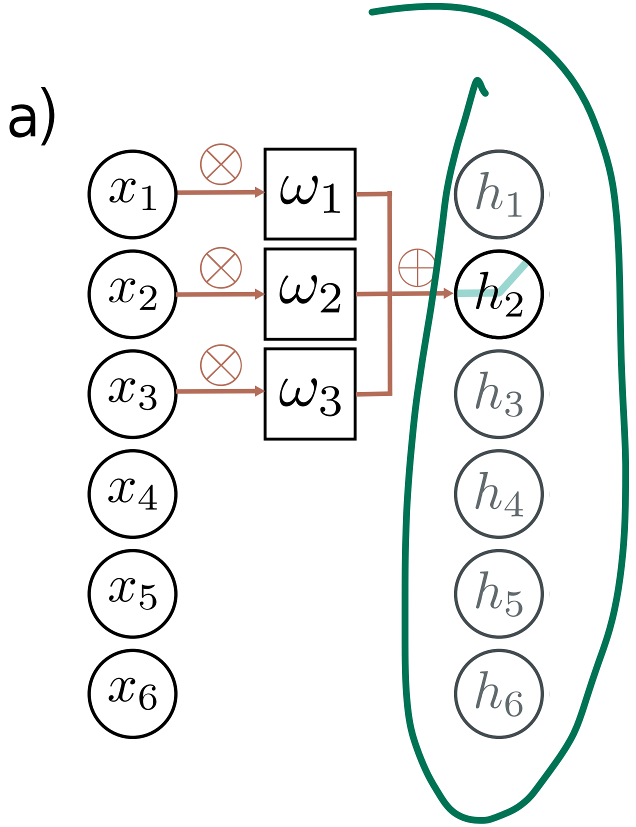
Convolutional networks

- Networks for images
- Invariance and equivariance
- 1D convolution
- Convolutional layers
- Channels
- Receptive fields
- Convolutional network for MNIST 1D

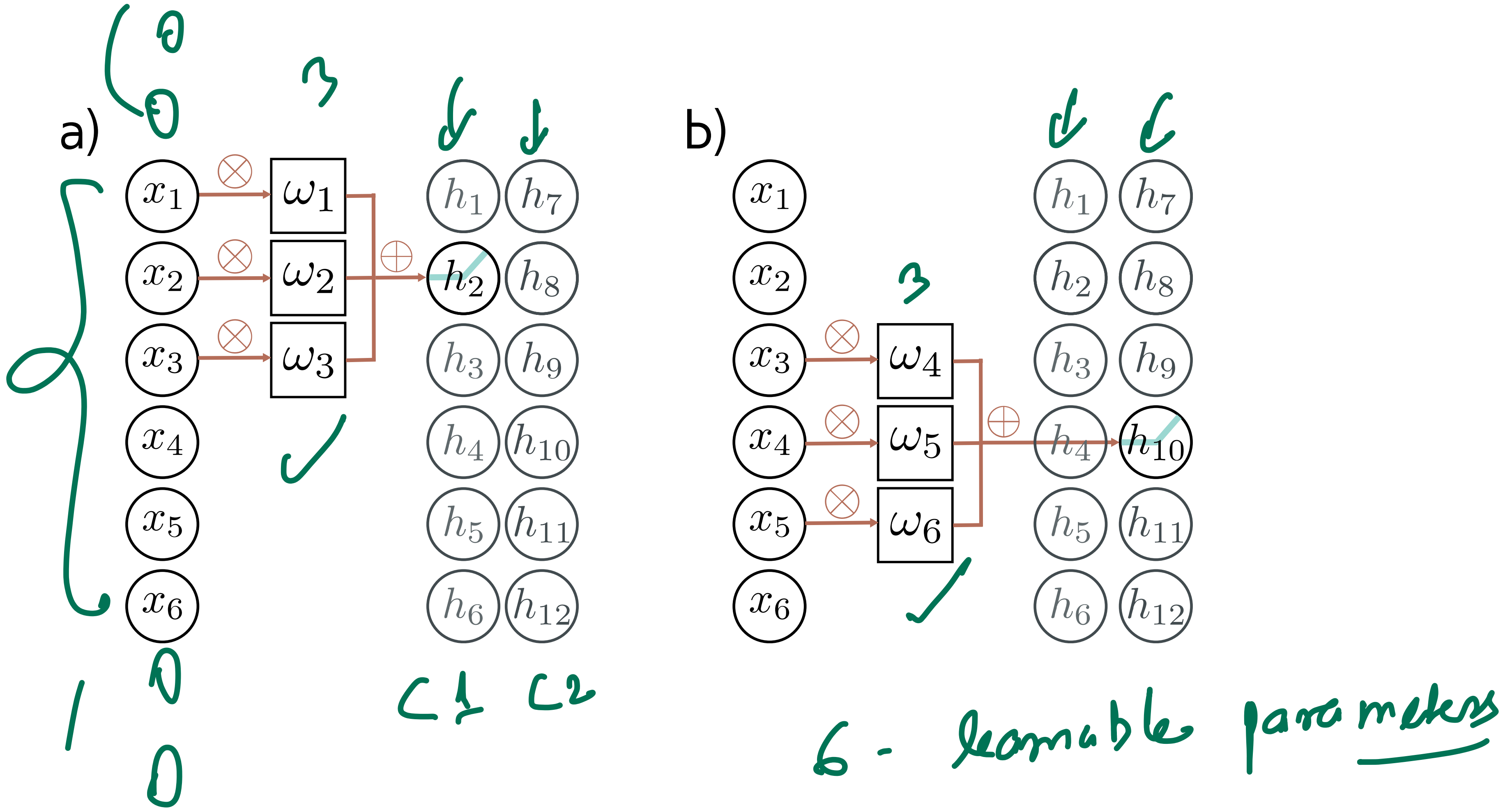
Channels

- The convolutional operation averages together the inputs
- Plus passes through ReLU function
- Has to lose information
- Solution:
  - apply several convolutions and stack them in channels
  - Sometimes also called feature maps

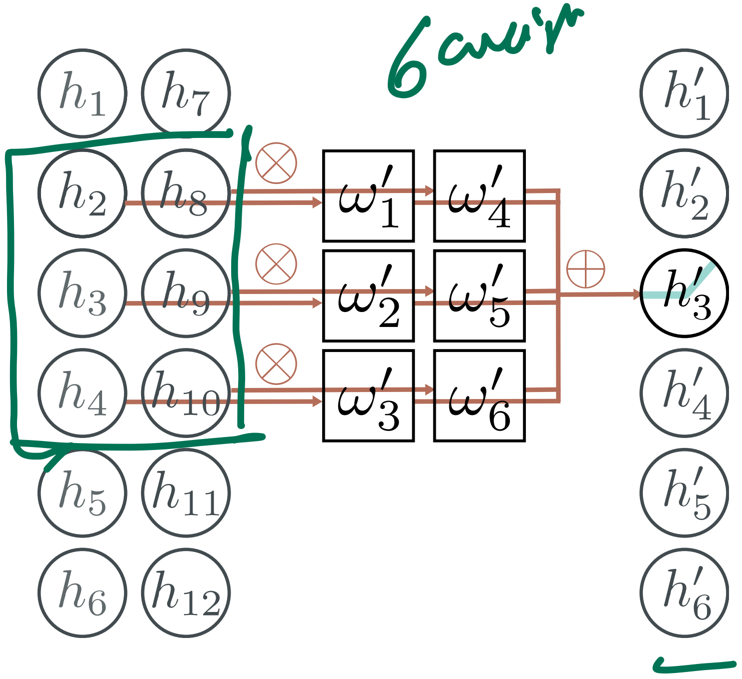
Two output channels, one input channel



Two output channels, one input channel



Two input channels, one output channel



How many parameters?

$C_i \times K$

- If there are C_i input channels and kernel size K

$$\boldsymbol{\Omega} \in \mathbb{R}^{C_i \times K} \qquad \boldsymbol{\beta} \in \mathbb{R}$$

- If there are C_i input channels and C_o output channels

$$\boldsymbol{\Omega} \in \mathbb{R}^{C_i \times C_o \times K} \qquad \boldsymbol{\beta} \in \mathbb{R}^{C_o}$$