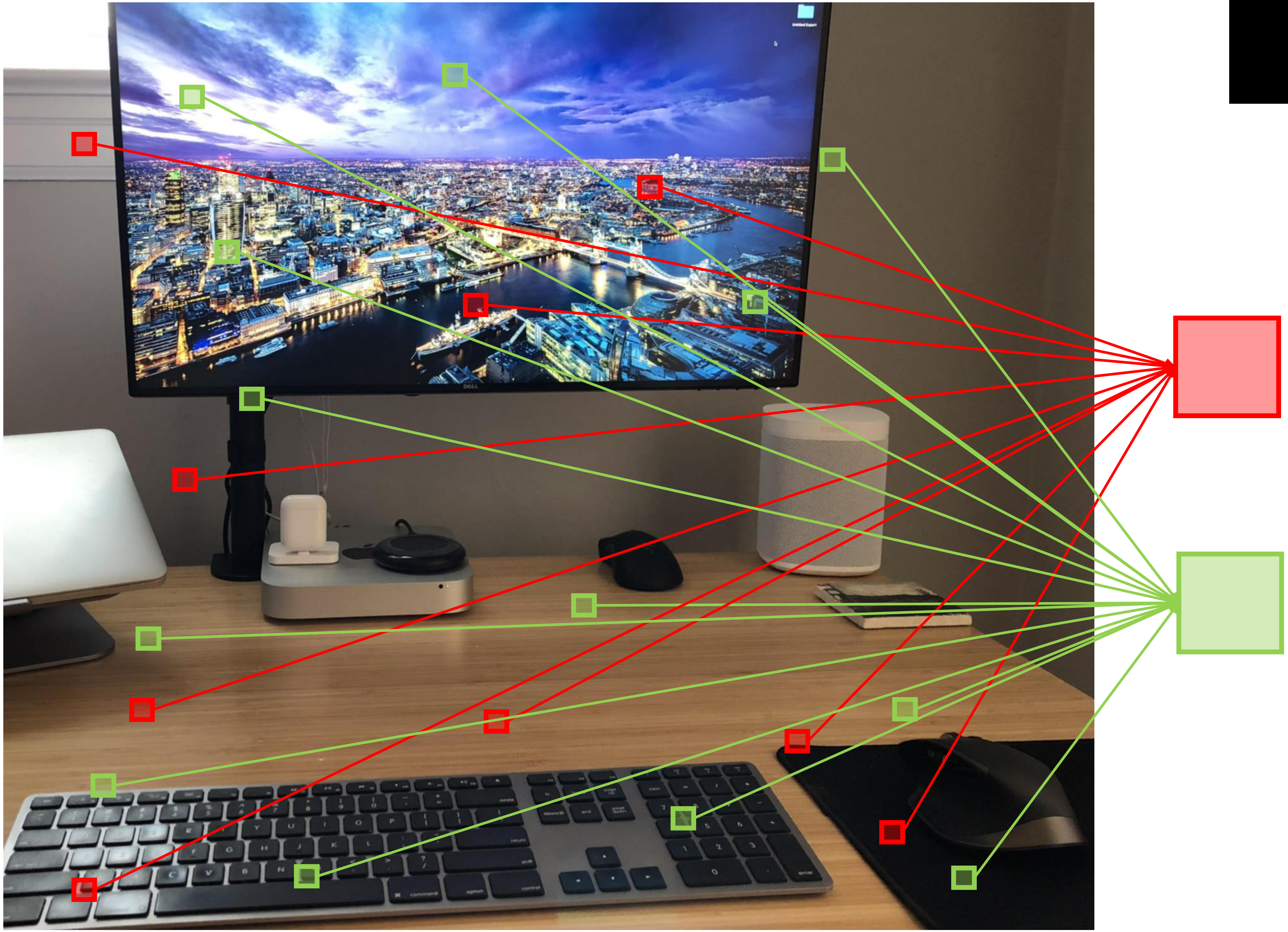
Lecture 3: Convolutional Neural Networks

Today:

- Expectations of visual recognition systems (15%)
- A Conv layer, stride, padding etc. (25%)
- Conv2D (20%)
- Transposed Conv (5%)
- Implementation and backprop (15%)
- What is encoded in feature maps? (10%)
- Max-pooling, convnet and conv variants (10%)

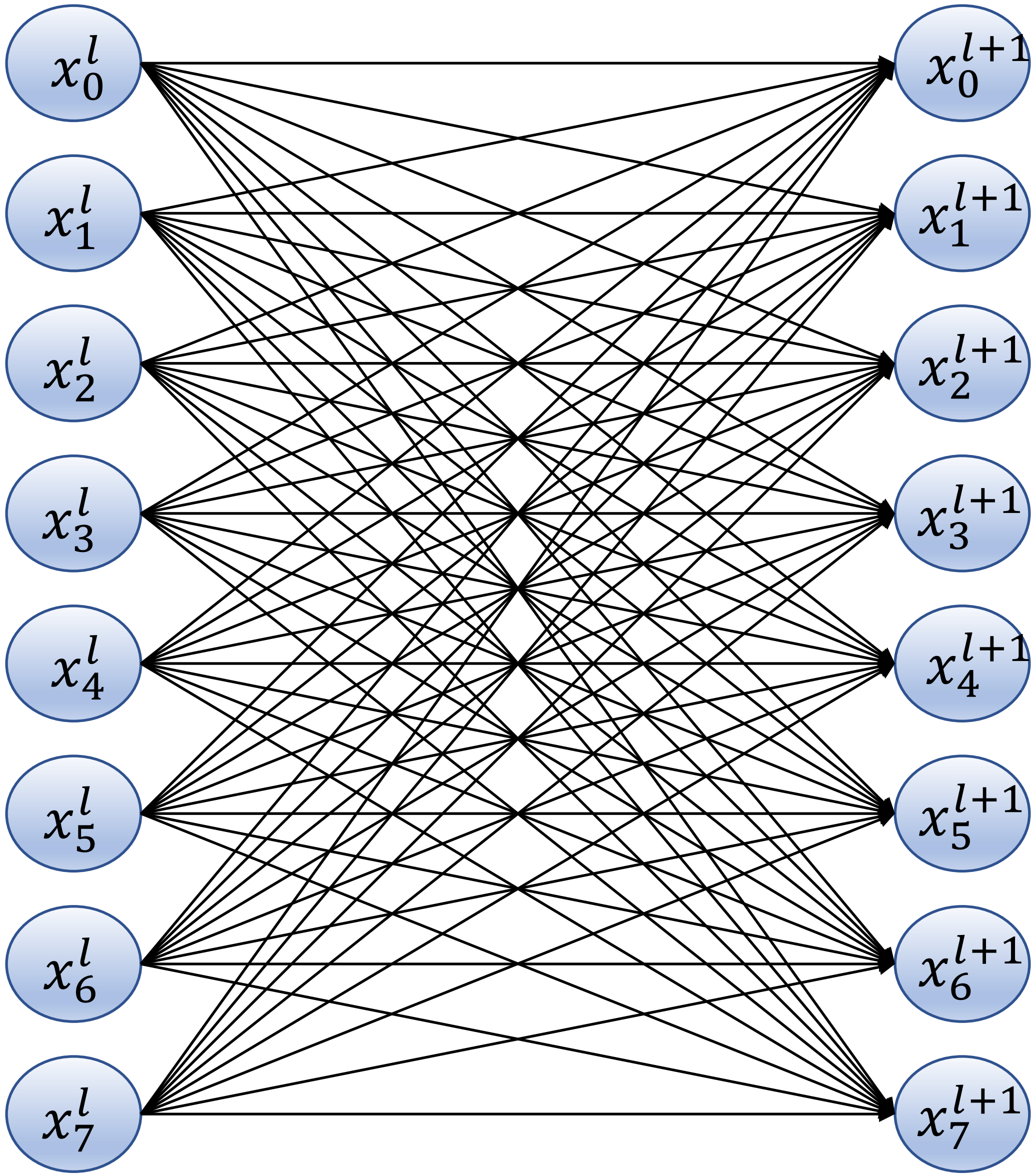



Monitor
Desk
vertical line
keyboard
Mouse
key
key
key key
key key key
key
key
key
horizontal line
key
key

Expectations of visual recognition network

1. Maintain 2D structure logic
2. Shift invariant (actually, equivariant)
3. Consider only local correlations
4. Hierarchically growing field of view
5. Hierarchically progressing complexity
6. Reasonable amount of params

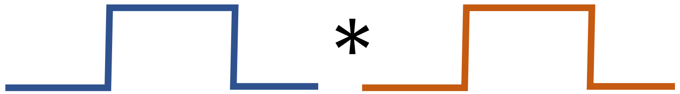
Fully connected layer

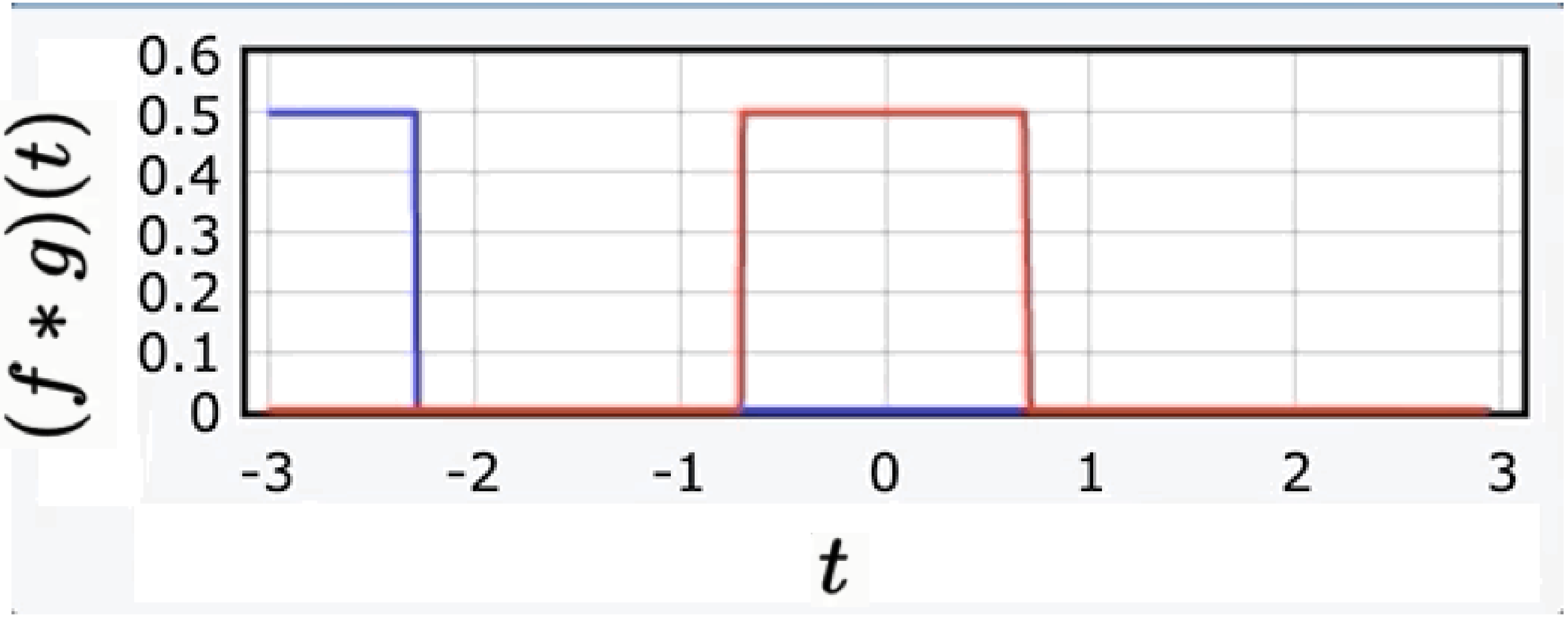


$$\begin{bmatrix} w_{00}^l & w_{01}^l & \cdots \\ w_{10}^l & w_{11}^l & \\ & \vdots & \ddots \end{bmatrix}$$

- ✗ 1. Maintain 2D structure logic
- ✗ 2. Shift invariant (actually, equivariant)
- ✗ 3. Consider only local correlations
- ✗ 4. Hierarchically growing field of view
- ✓ 5. Hierarchically progressing complexity
- ✗ 6. Reasonable amount of params

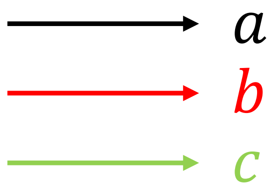
Convolution

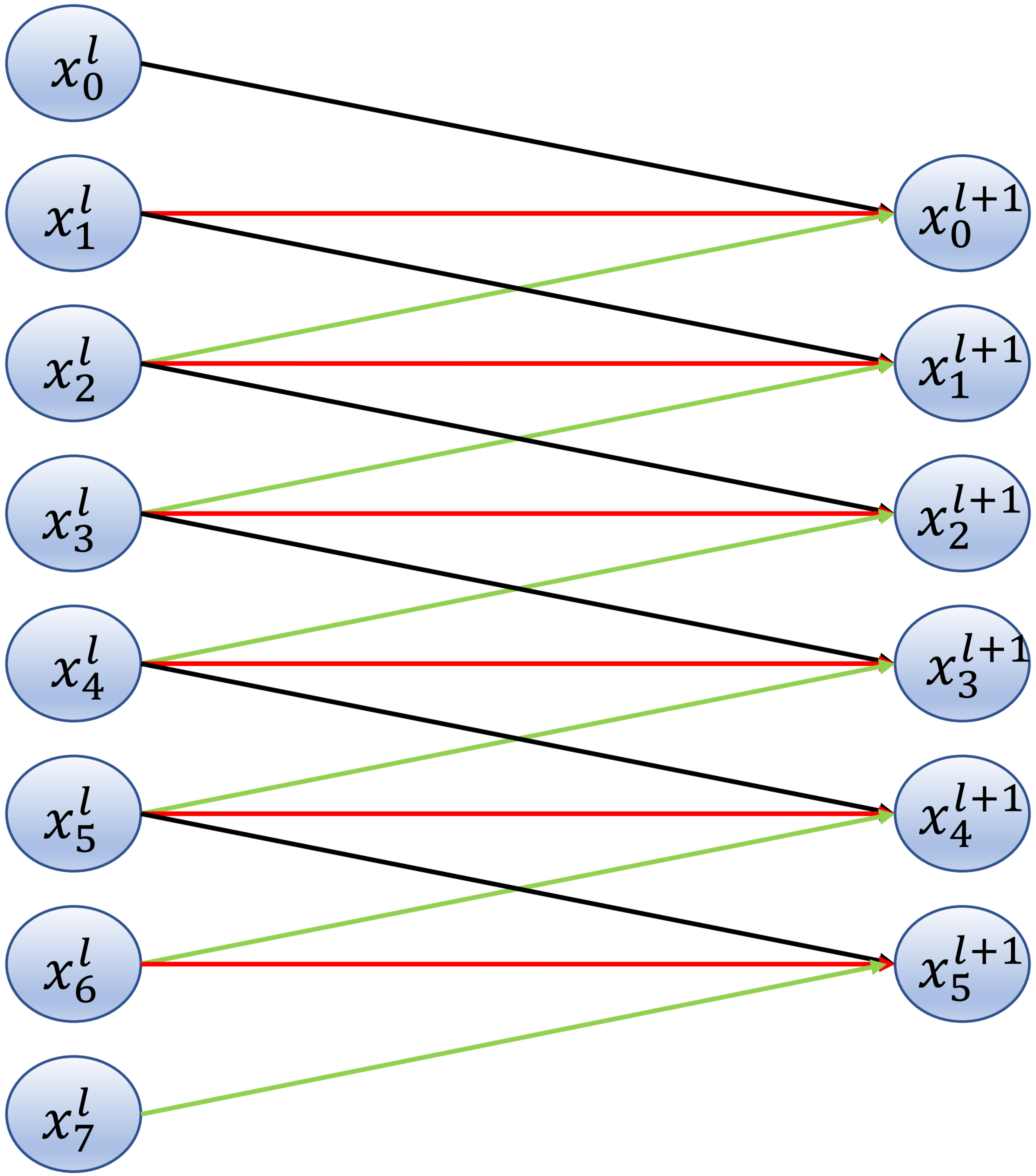




$$(f * g)(t) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

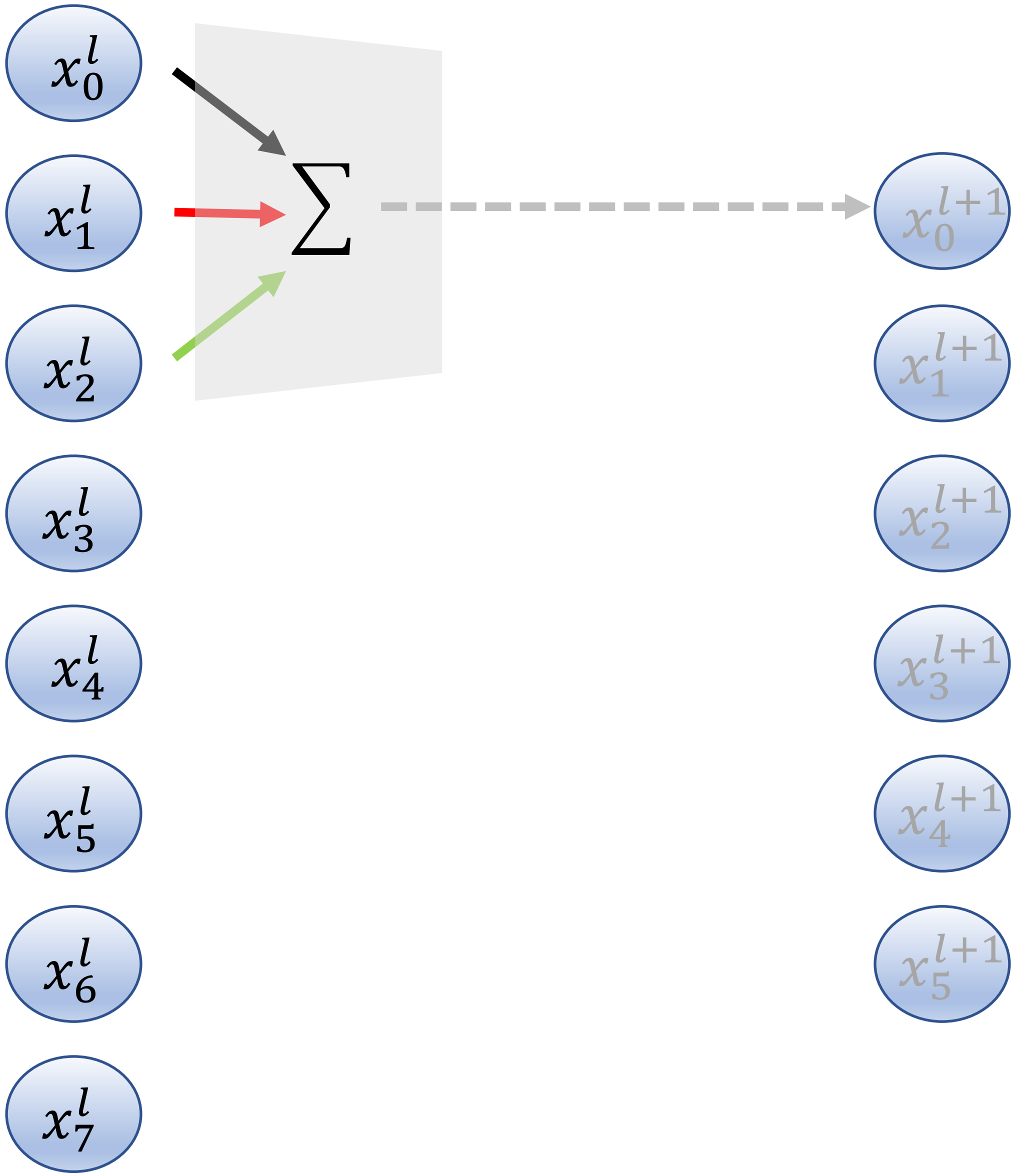
Convolution layer





Toeplitz matrix

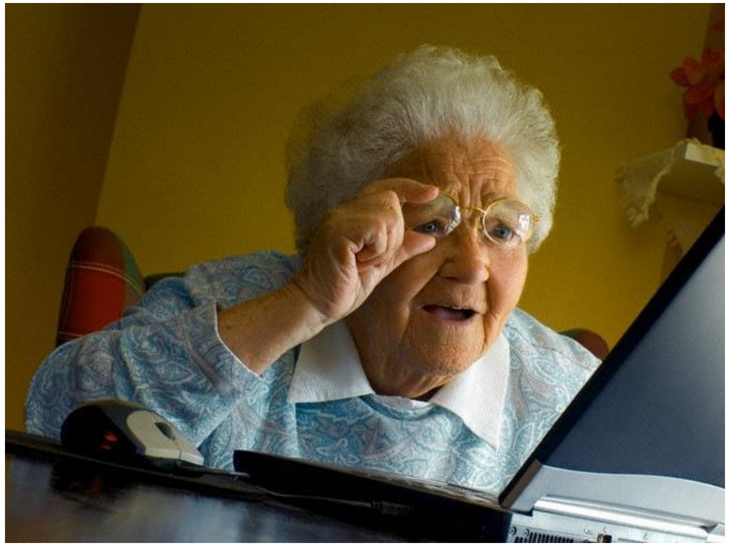
Convolution Filter



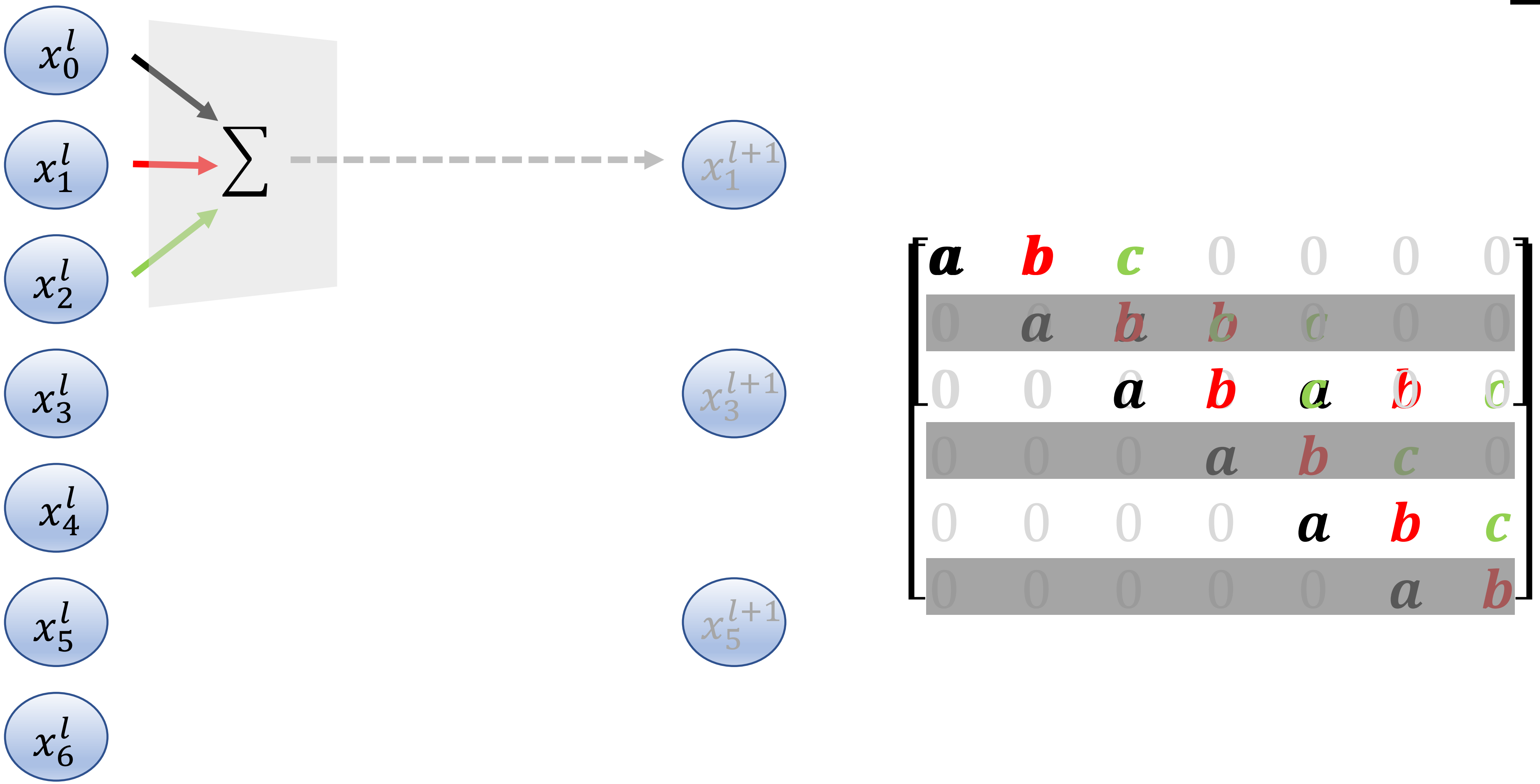
Q: Given input size and filter size, find output size.

Q: is this a convolution?

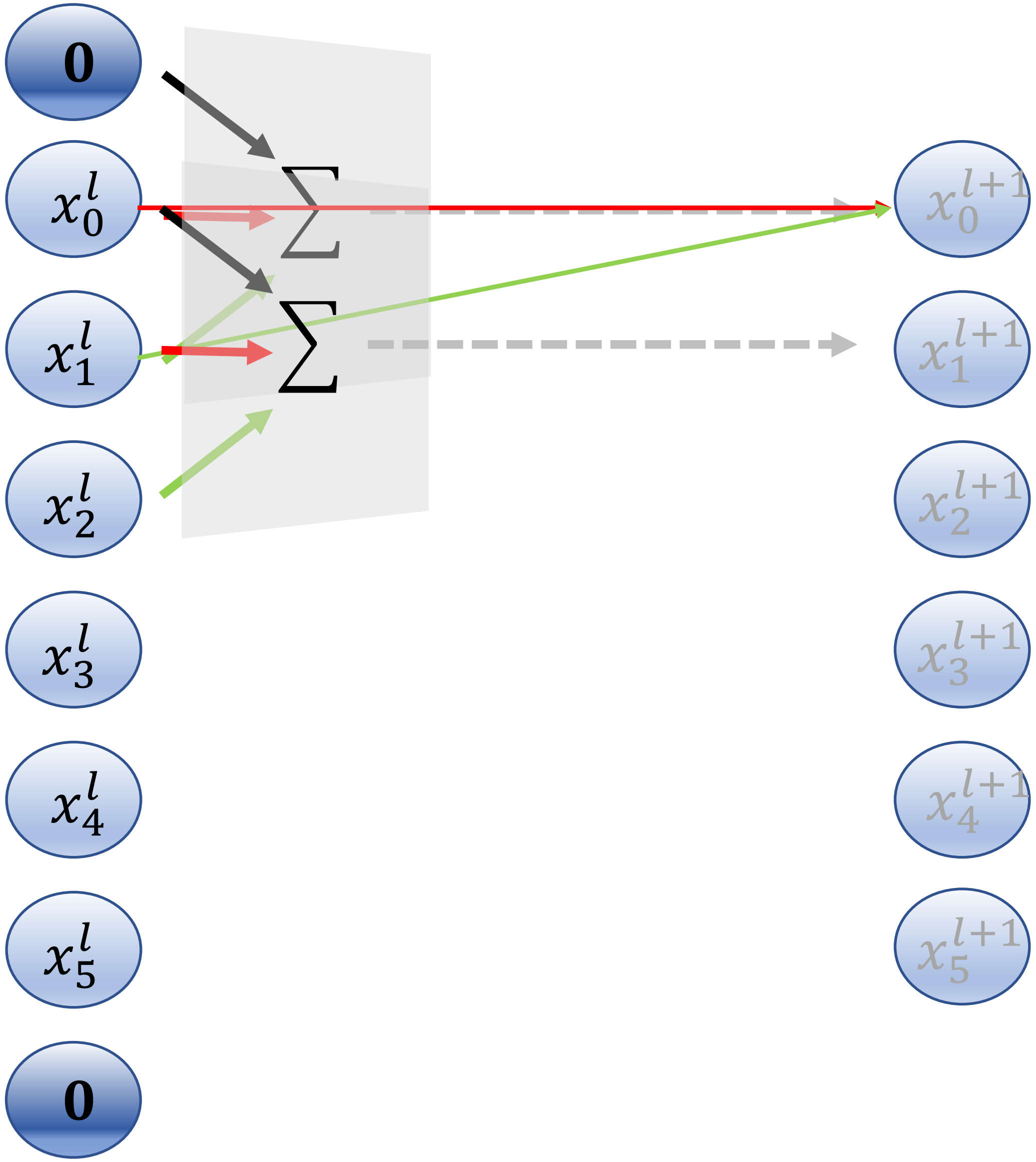
A: Yes, but with the flipped filter.
This is cross-correlation.



Stride

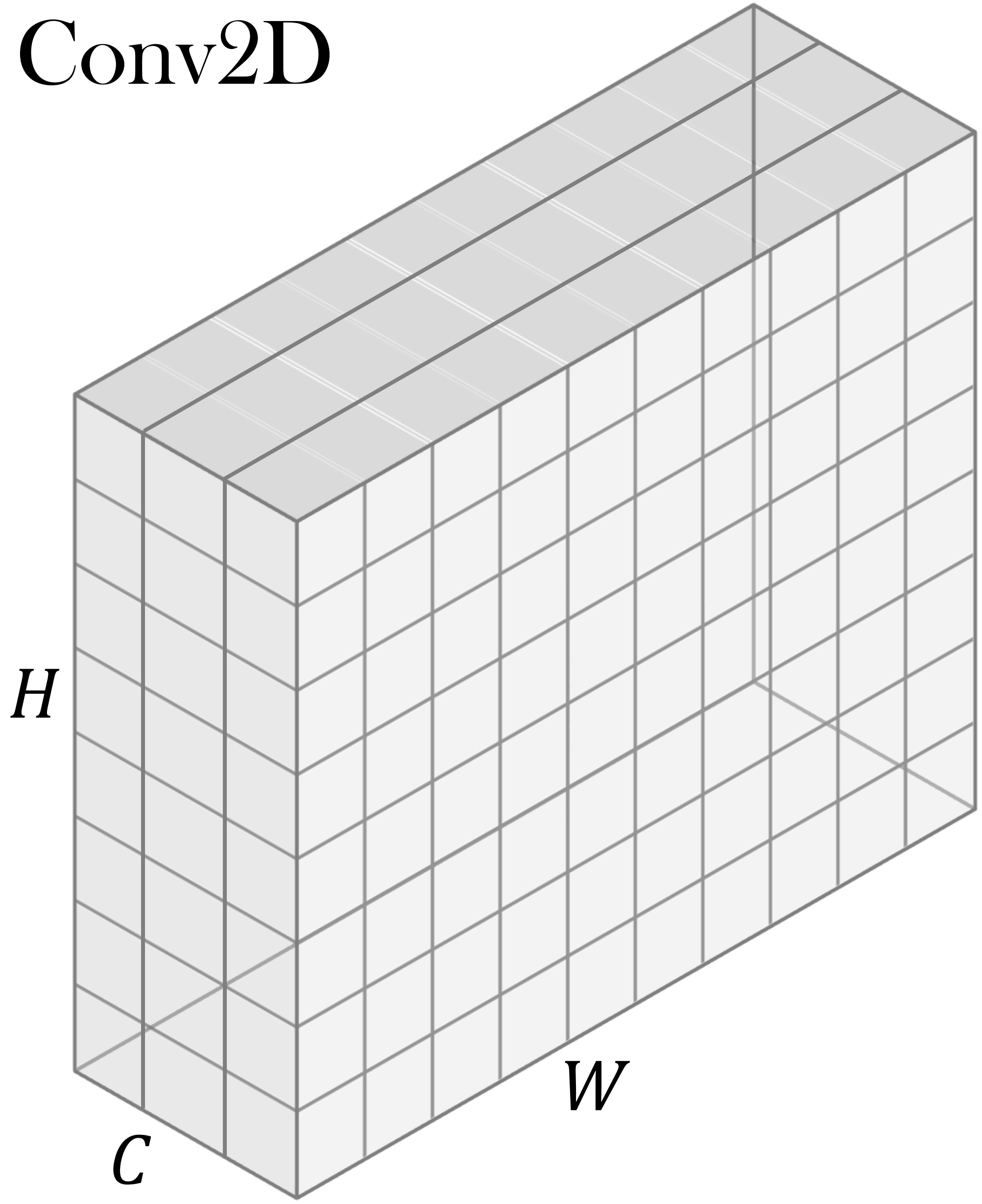


Padding

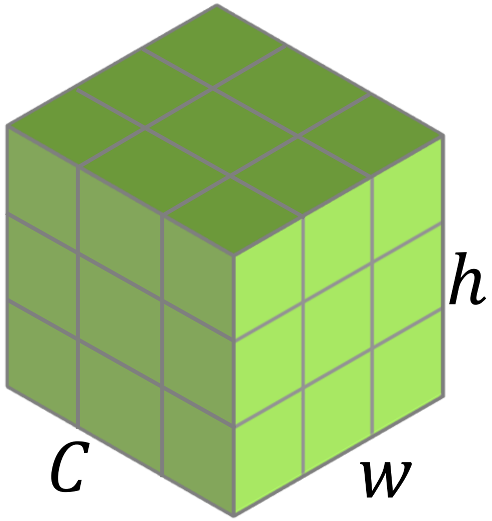


0
x_0^l
x_1^l
x_2^l
x_3^l
x_4^l
x_5^l
0
$\sum$
$\sum$
x_0^{l+1}
x_1^{l+1}
x_2^{l+1}
x_3^{l+1}
x_4^{l+1}
x_5^{l+1}

Conv2D

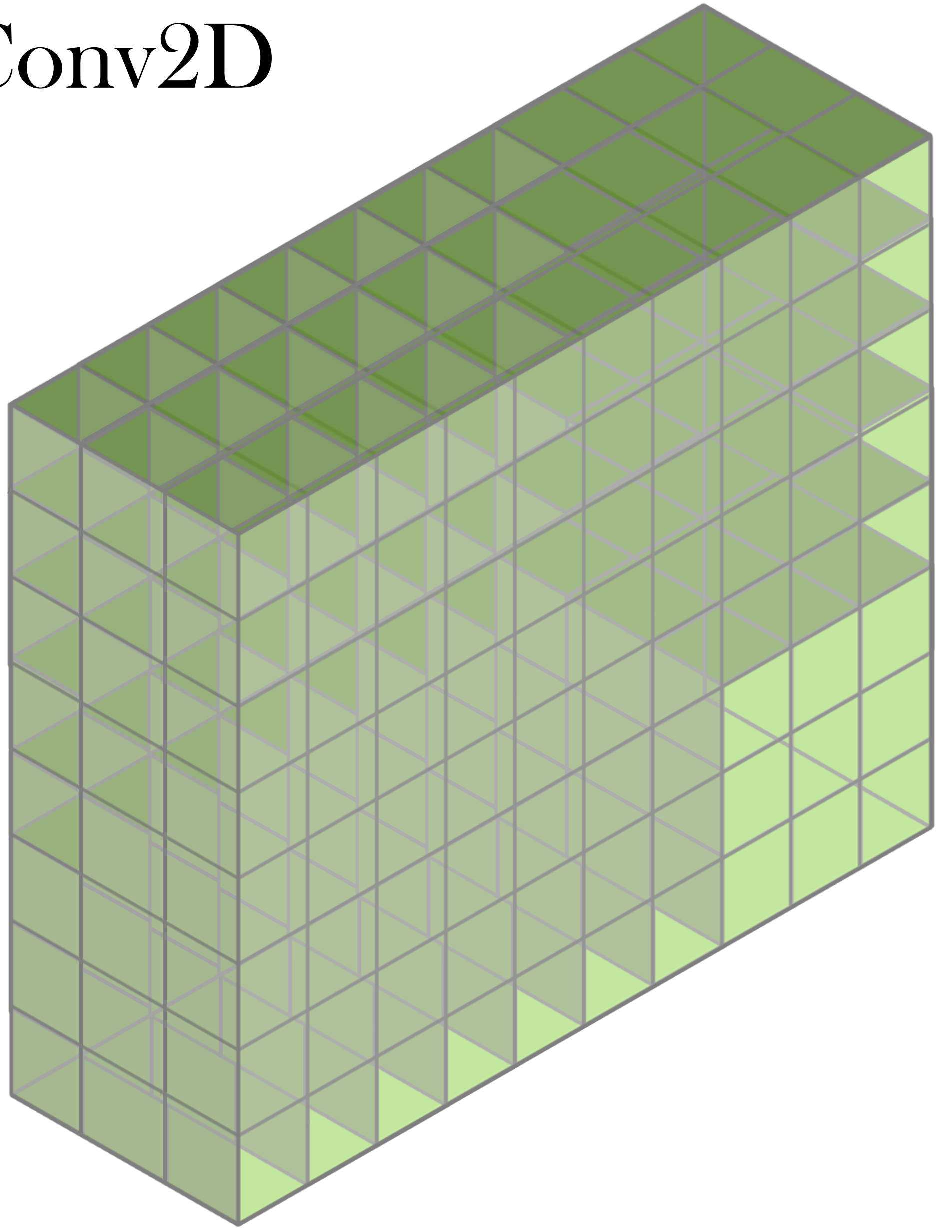


N, C, H, W

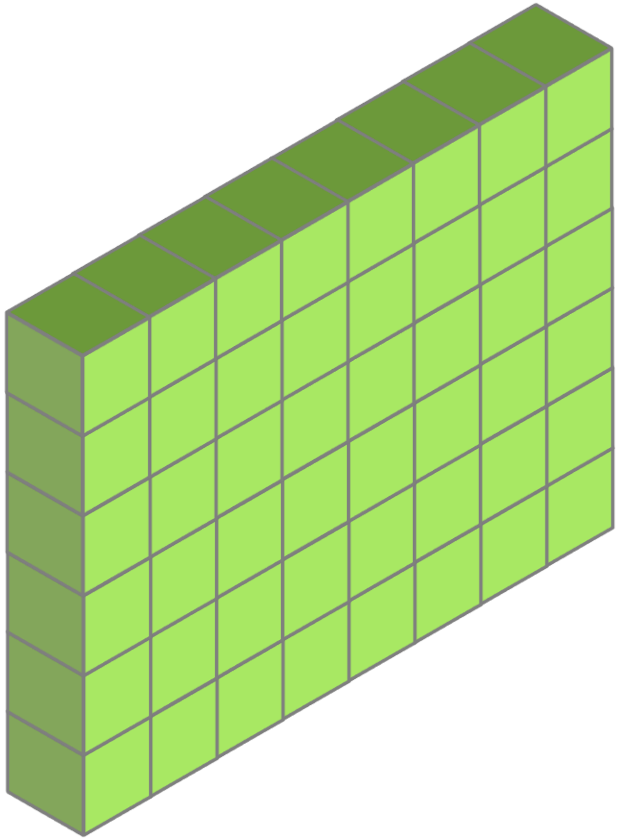


C, h, w

Conv2D

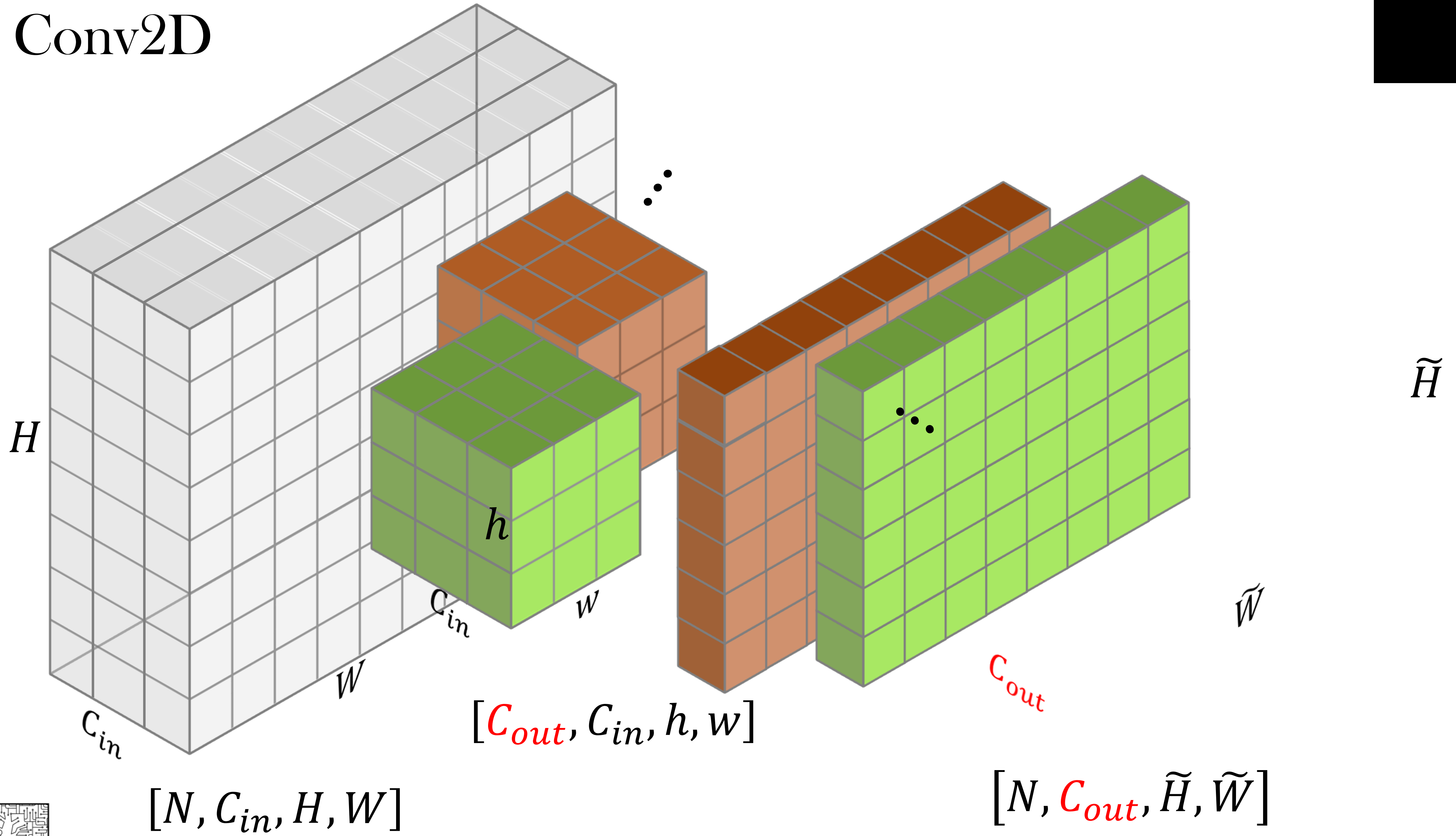


N, C, H, W



$\left[N, 1, \widetilde{H}, \widetilde{W}\right]$

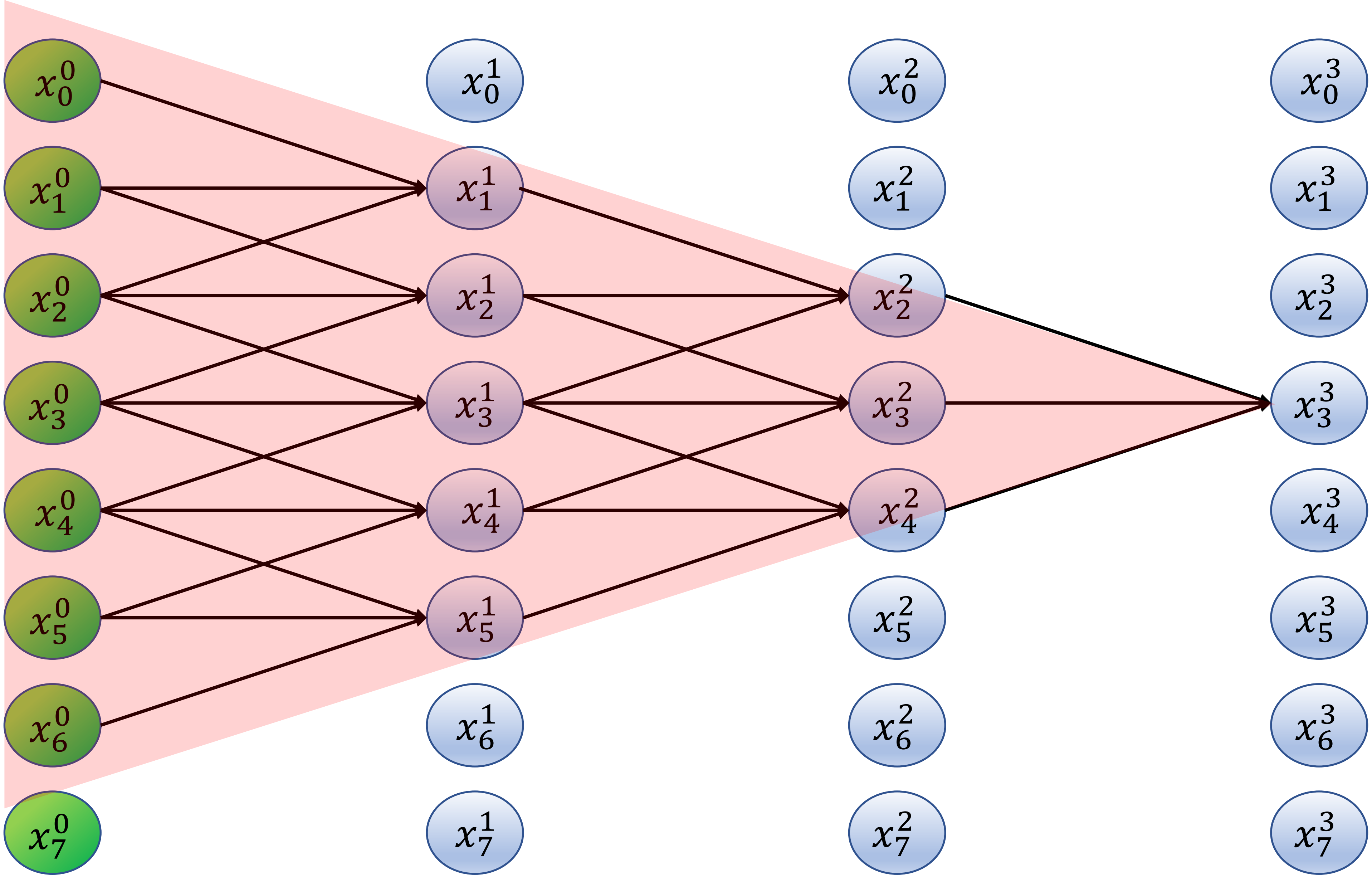
Conv2D



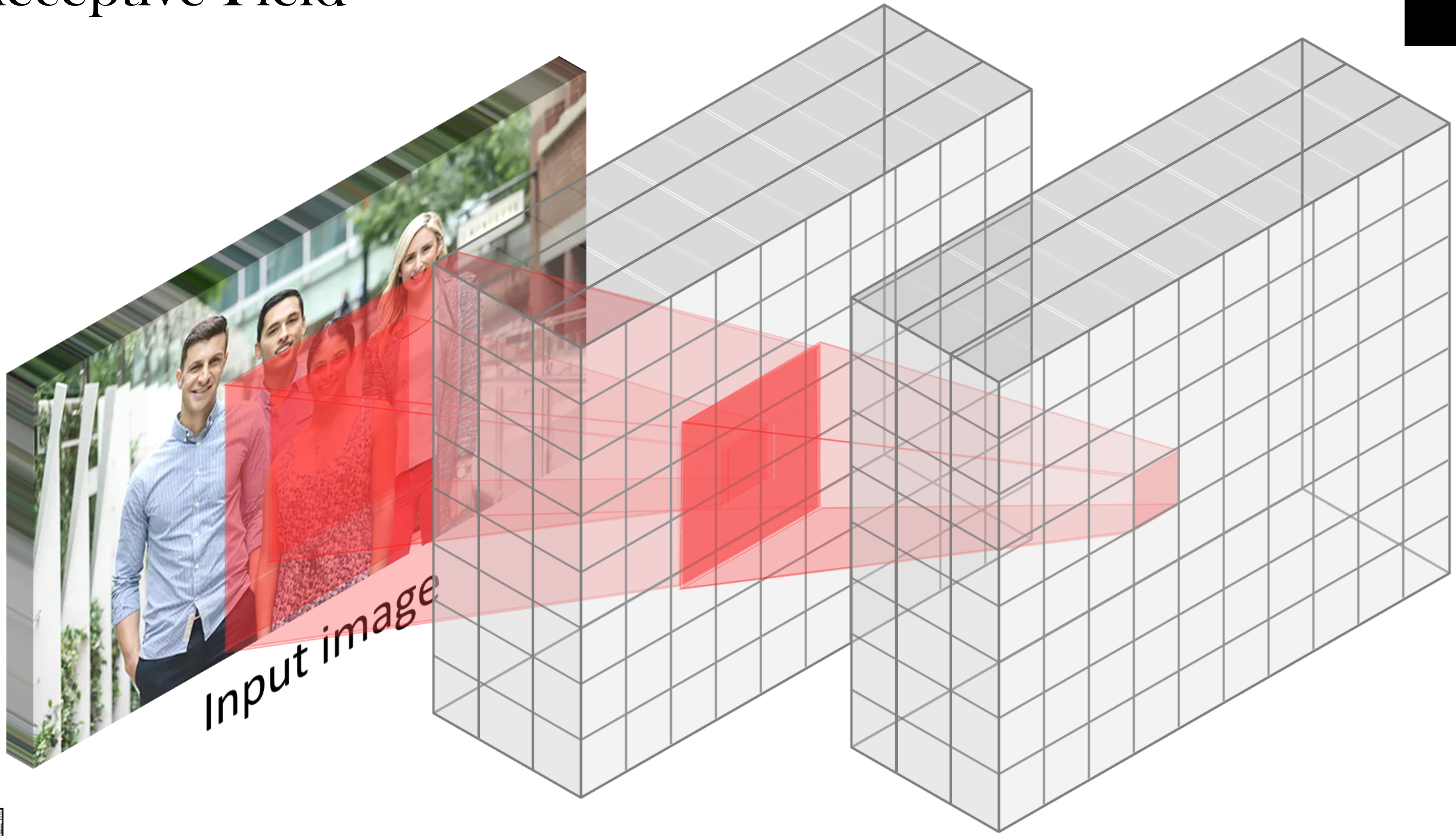
H
W
C_{in}
h
w
C_{in}
C_{out}
$\widetilde{H}$
$\widetilde{W}$
$[C_{out}, C_{in}, h, w]$
$[N, C_{in}, H, W]$
$[N, C_{out}, \widetilde{H}, \widetilde{W}]$

Receptive Field

$[\mathit{Filter\ size}: 3]$



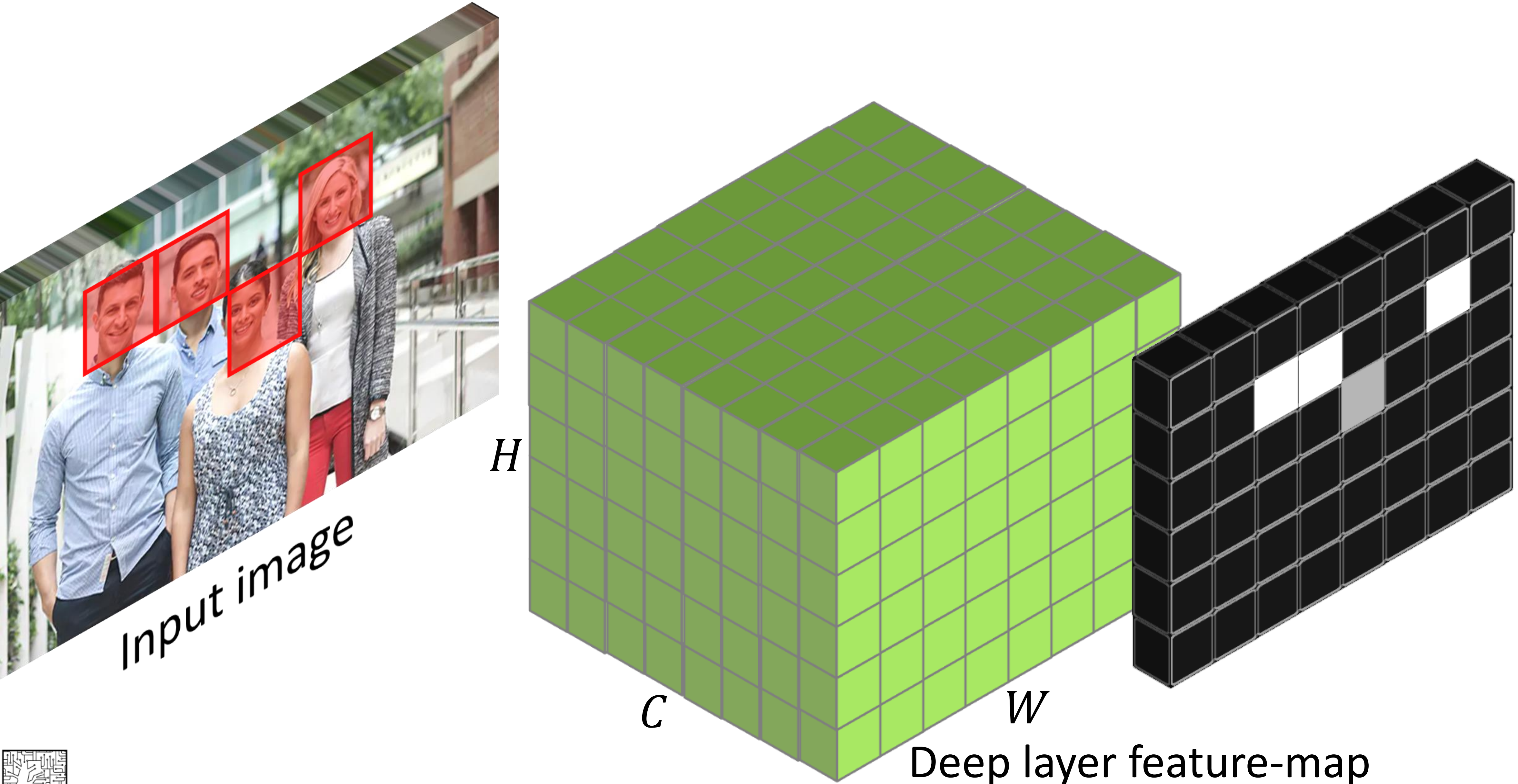
Receptive Field



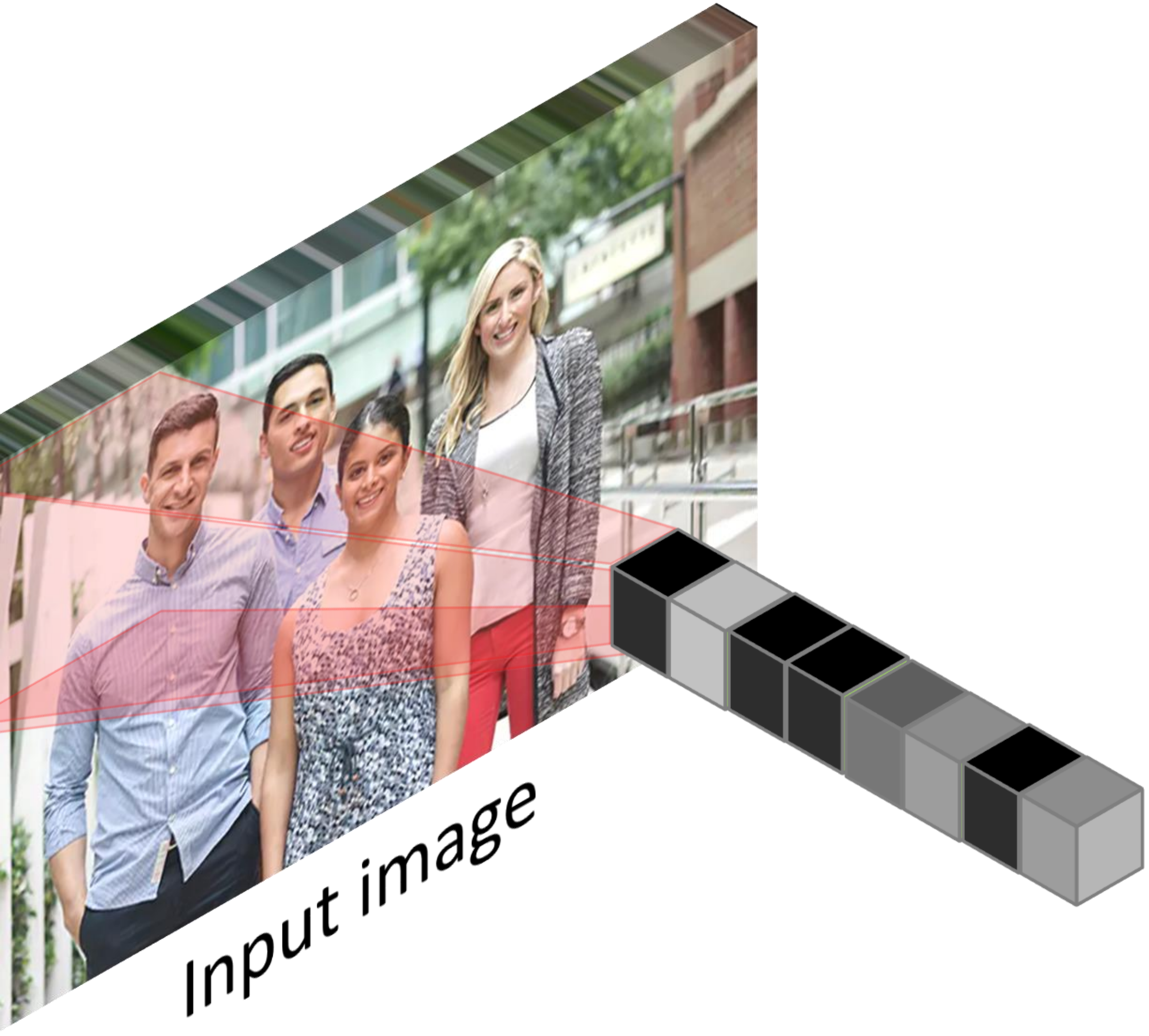
Convs rock!

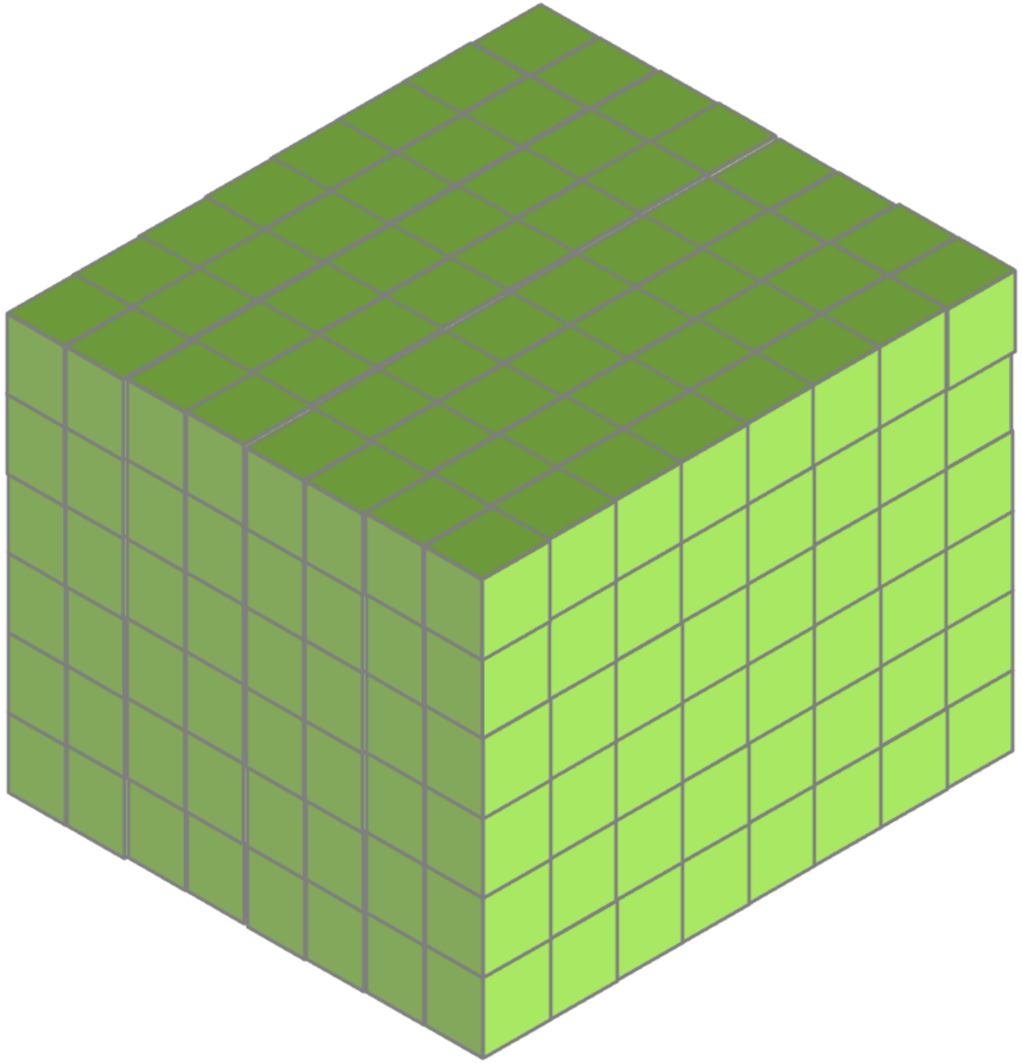
- ✓ 1. Maintain 2D structure logic
- ✓ 2. Shift invariant (actually, equivariant)
- ✓ 3. Consider only local correlations
- ✓ 4. Hierarchically growing field of view
- ✓ 5. Hierarchically progressing complexity
- ✓ 6. Reasonable amount of params

Two important intuitions about feature maps



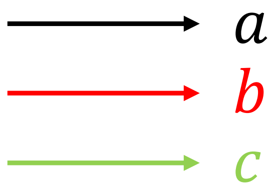
Two important intuitions about feature maps

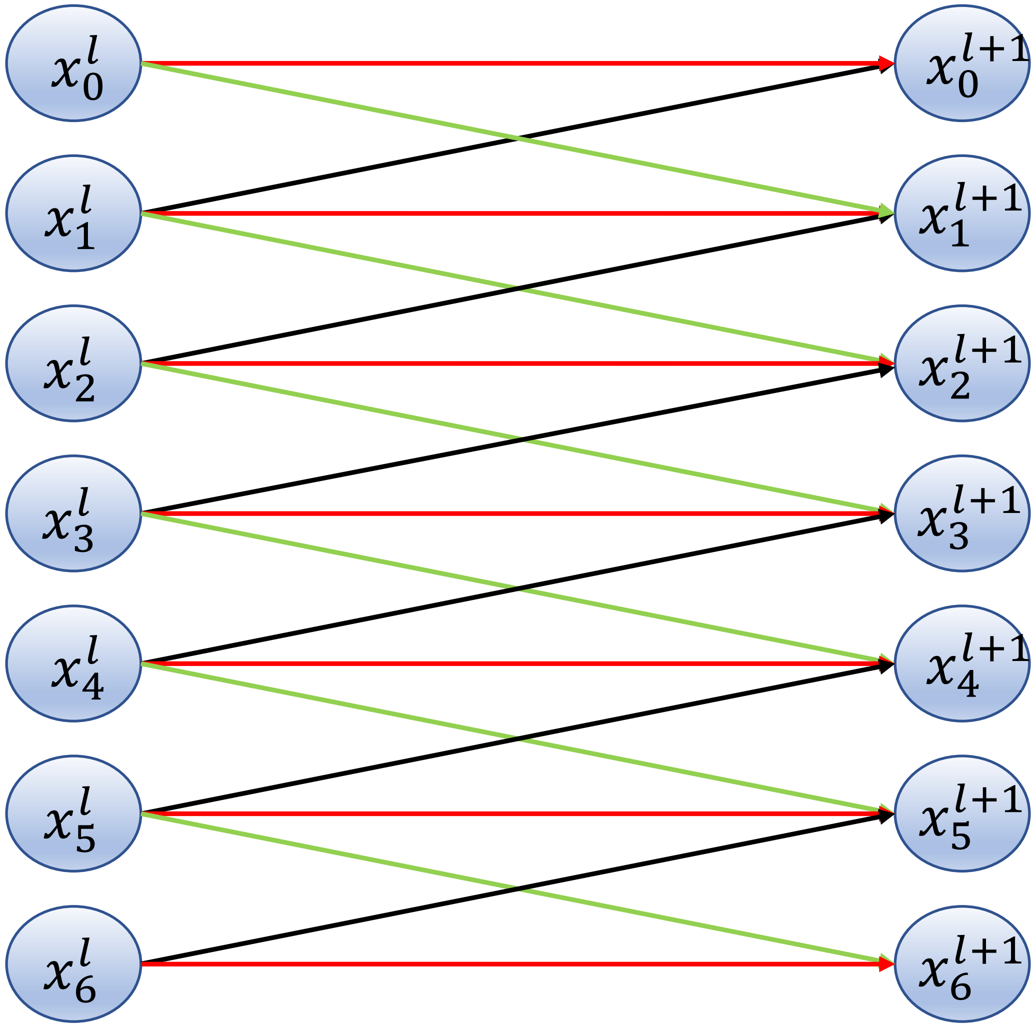




Deep layer feature-map

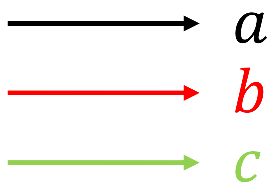
Transposed Convolution

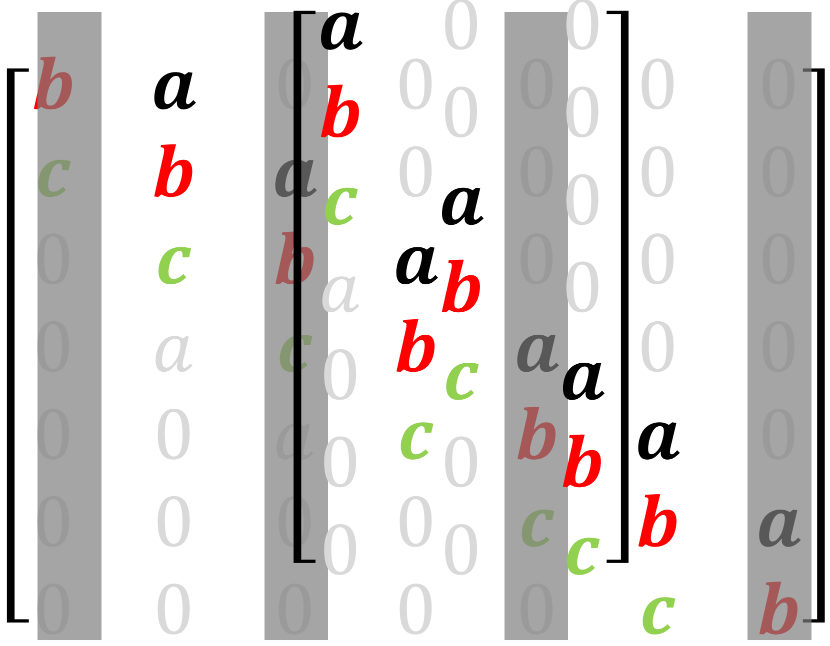




$$\begin{bmatrix} b & a & 0 & 0 & 0 & 0 & 0 \\ c & b & a & 0 & 0 & 0 & 0 \\ 0 & c & b & a & 0 & 0 & 0 \\ 0 & a & c & b & a & 0 & 0 \\ 0 & 0 & a & c & b & a & 0 \\ 0 & 0 & 0 & 0 & c & b & a \\ 0 & 0 & 0 & 0 & 0 & c & b \end{bmatrix}$$

Transposed Convolution with stride

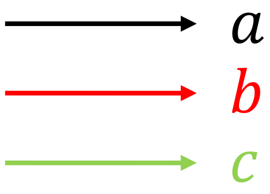


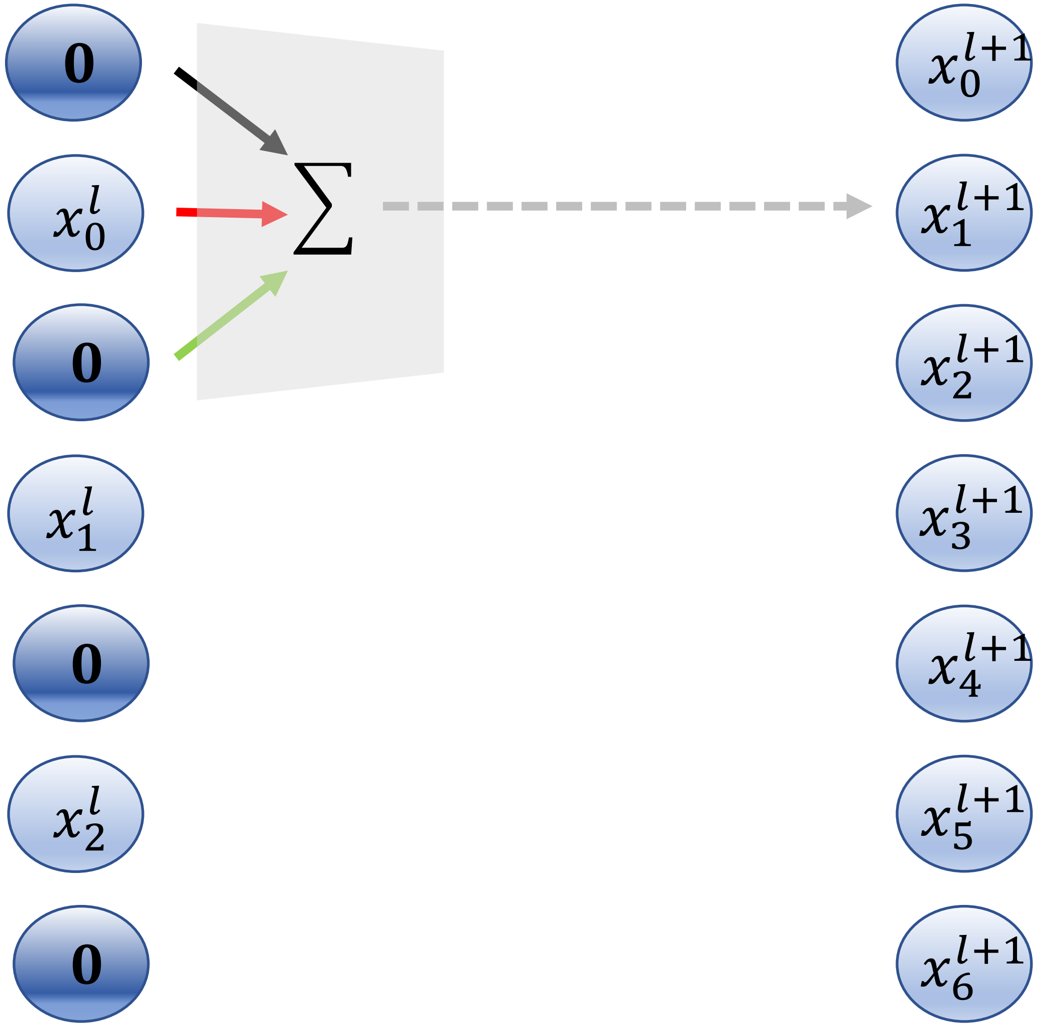


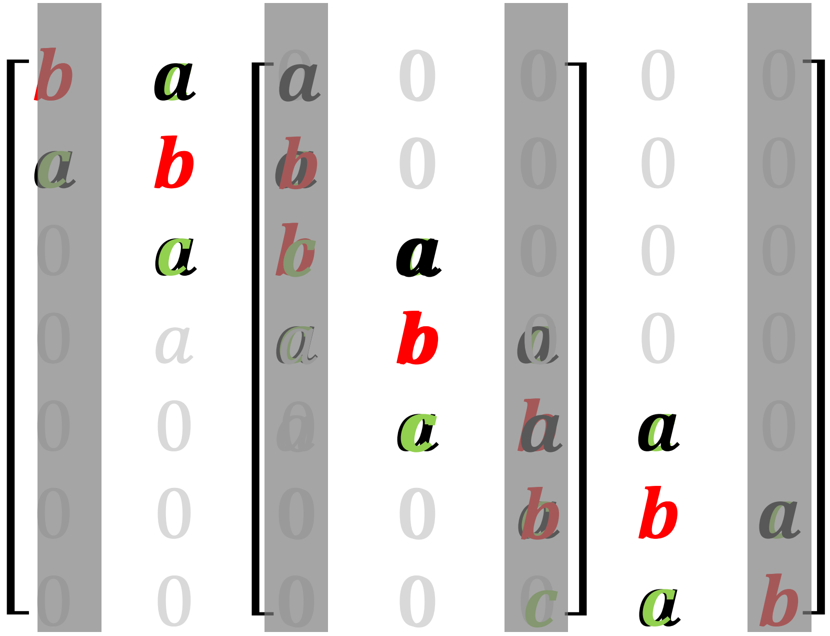
Recall- stride 2 conv:

$$\begin{bmatrix} a & b & c & 0 & 0 & 0 & 0 \\ 0 & 0 & a & b & c & 0 & 0 \\ 0 & 0 & 0 & 0 & a & b & c \end{bmatrix}$$

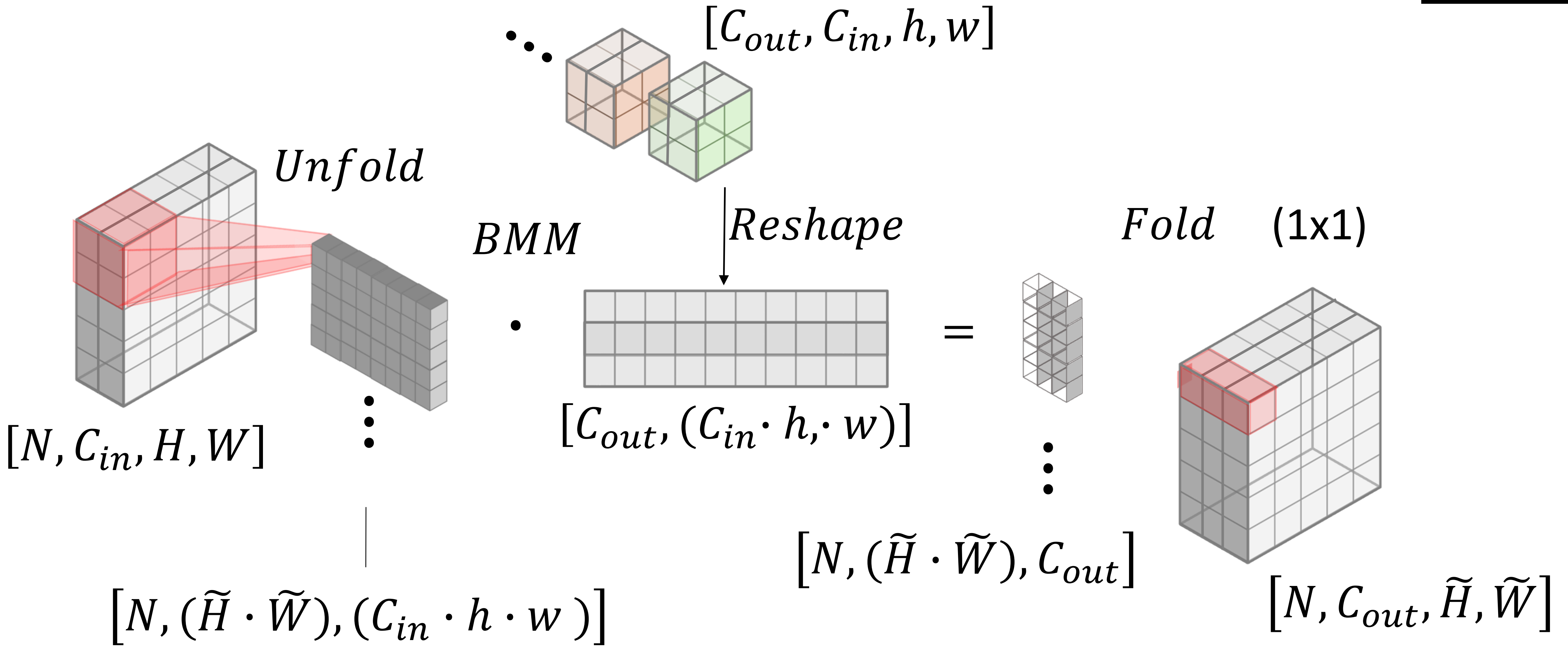
Transposed Convolution by dilation & flip







A note about the implementation of conv



Q: How do you backprop a Conv? (1D)

<u>For derivative of loss w.r.t all neurons</u>:
Transposed Conv with the same filter!

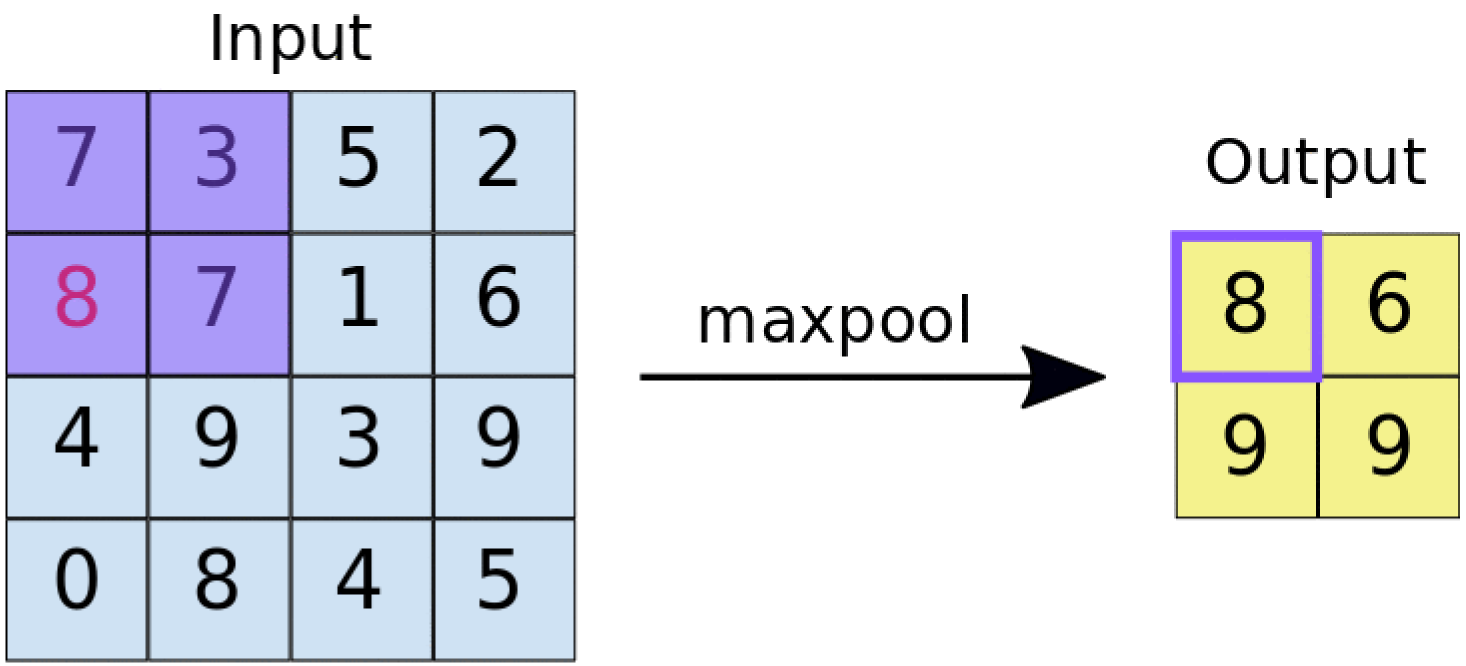
<u>For derivative of loss w.r.t filter weights</u>:
Multiply conv-output-grad with the neurons it “sees”, sum over all locations.

Q: How about Conv2D?

<u>For derivative of loss w.r.t all neurons</u>:
1. Transpose dims of filter c_{in} , c_{out} (as in FC)
2. Transposed Conv2D with the modified filter!

<u>For derivative of loss w.r.t filter weights</u>:
Outer-prod of conv-output-grad with the neurons it “sees”, sum over all locations.
(You need to unfold conv-input like in forward, and conv-out-grad with 1x1)

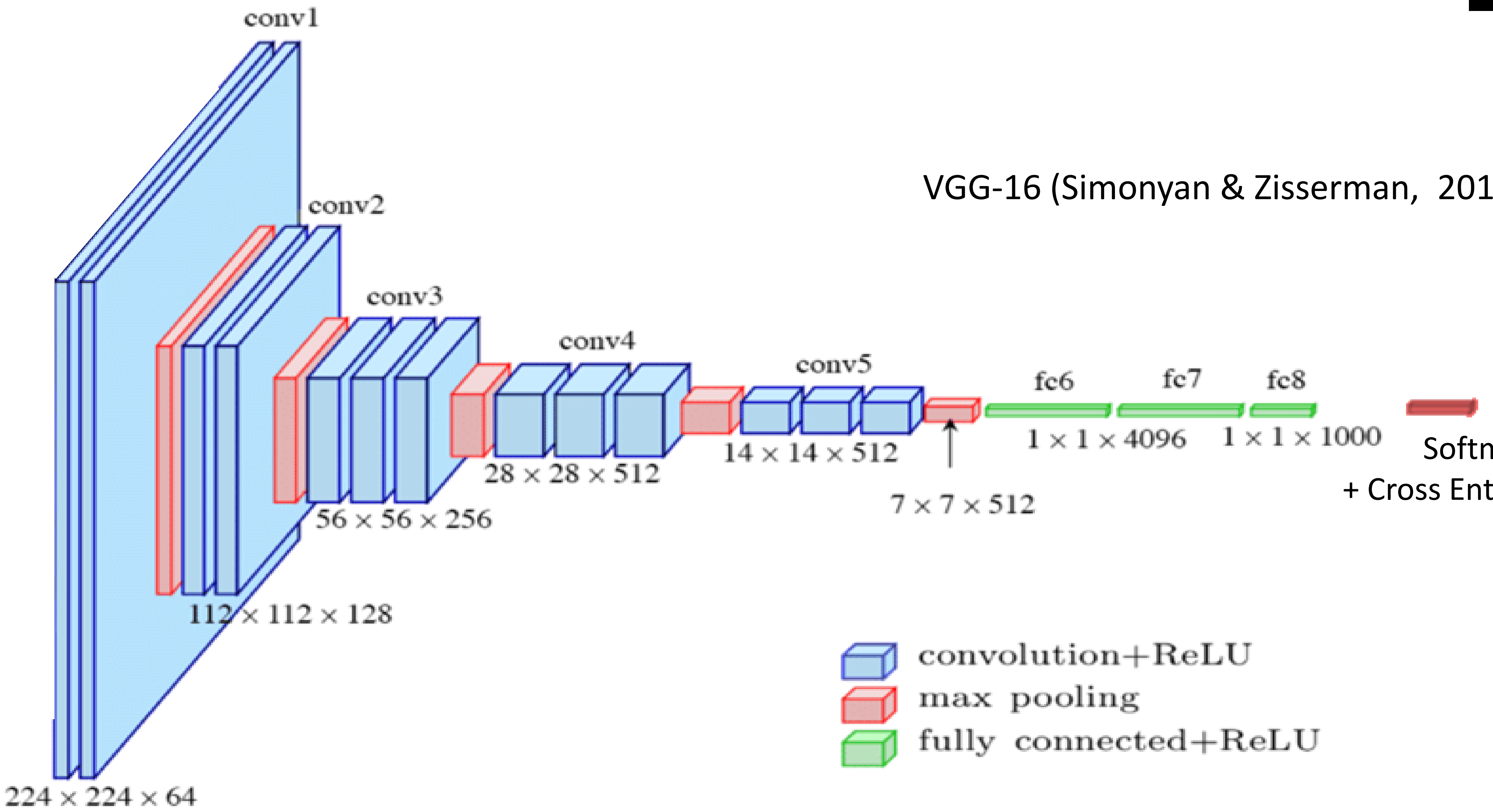
Max Pooling



- Usually stride=win-size, but not always.
- Each channel separately.

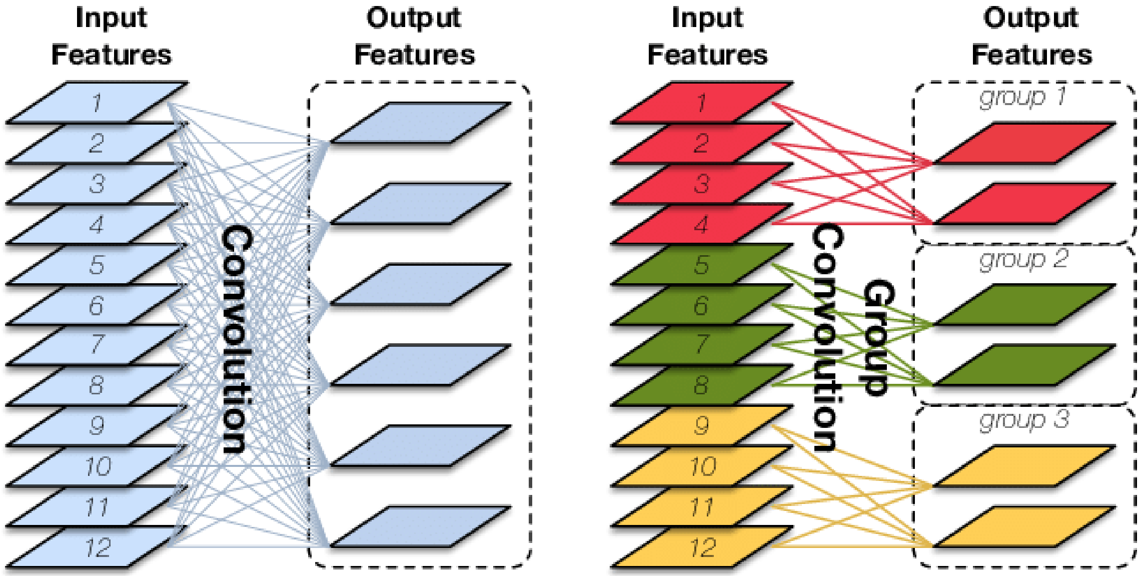
ConvNet Example

VGG-16 (Simonyan & Zisserman, 2014)

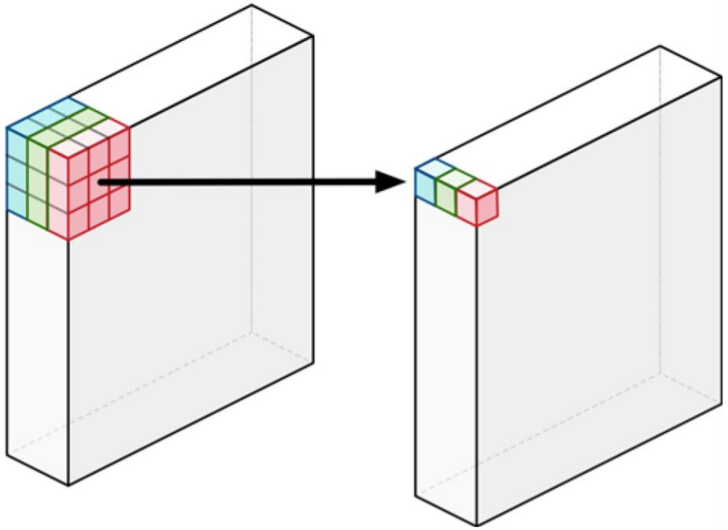


More special Convs!

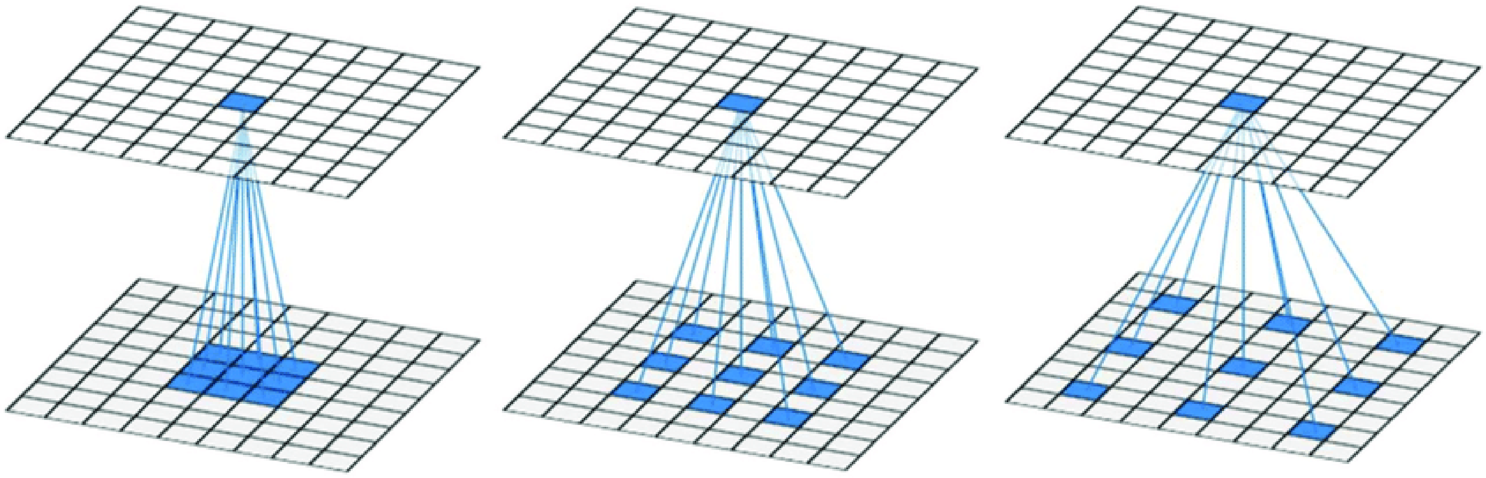
Group Conv (Krihzevsky 2012)



Depthwise Conv



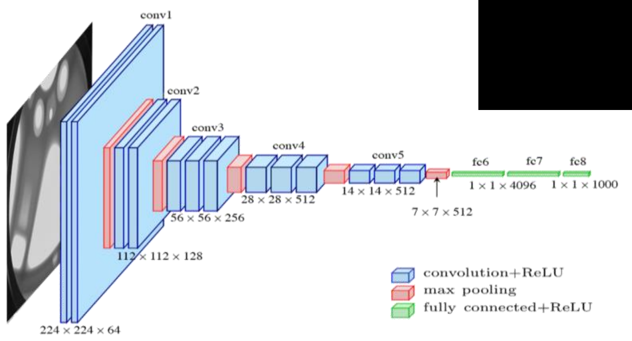
Dilated Conv (Yu&Koltun 2016)



dilation=1 dilation=2 dilation=3

This week's tutorial: CNN Architectures

Dror Moran



Next week's lecture:

Shai Bagon

Practical Training