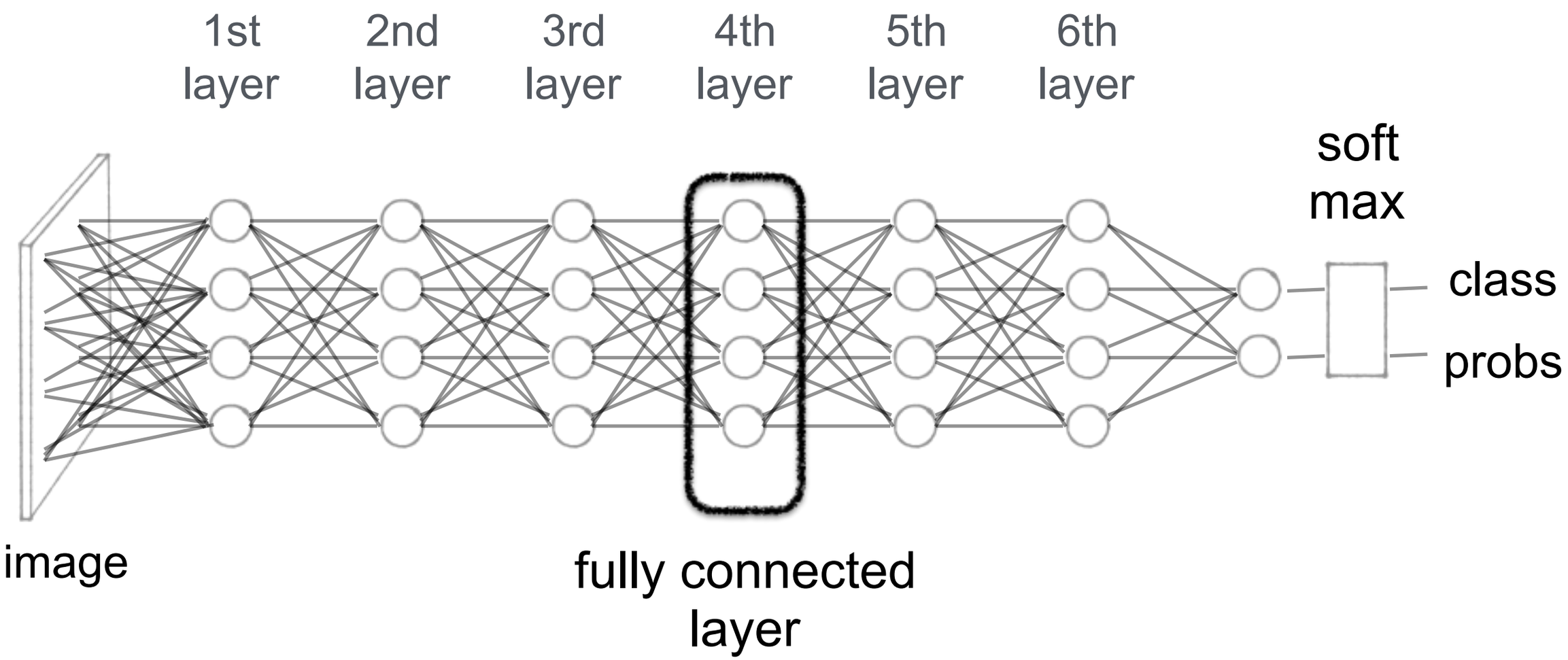
SSY097 - Image Analysis

Lecture 6 - (More) Convolutional Neural Networks

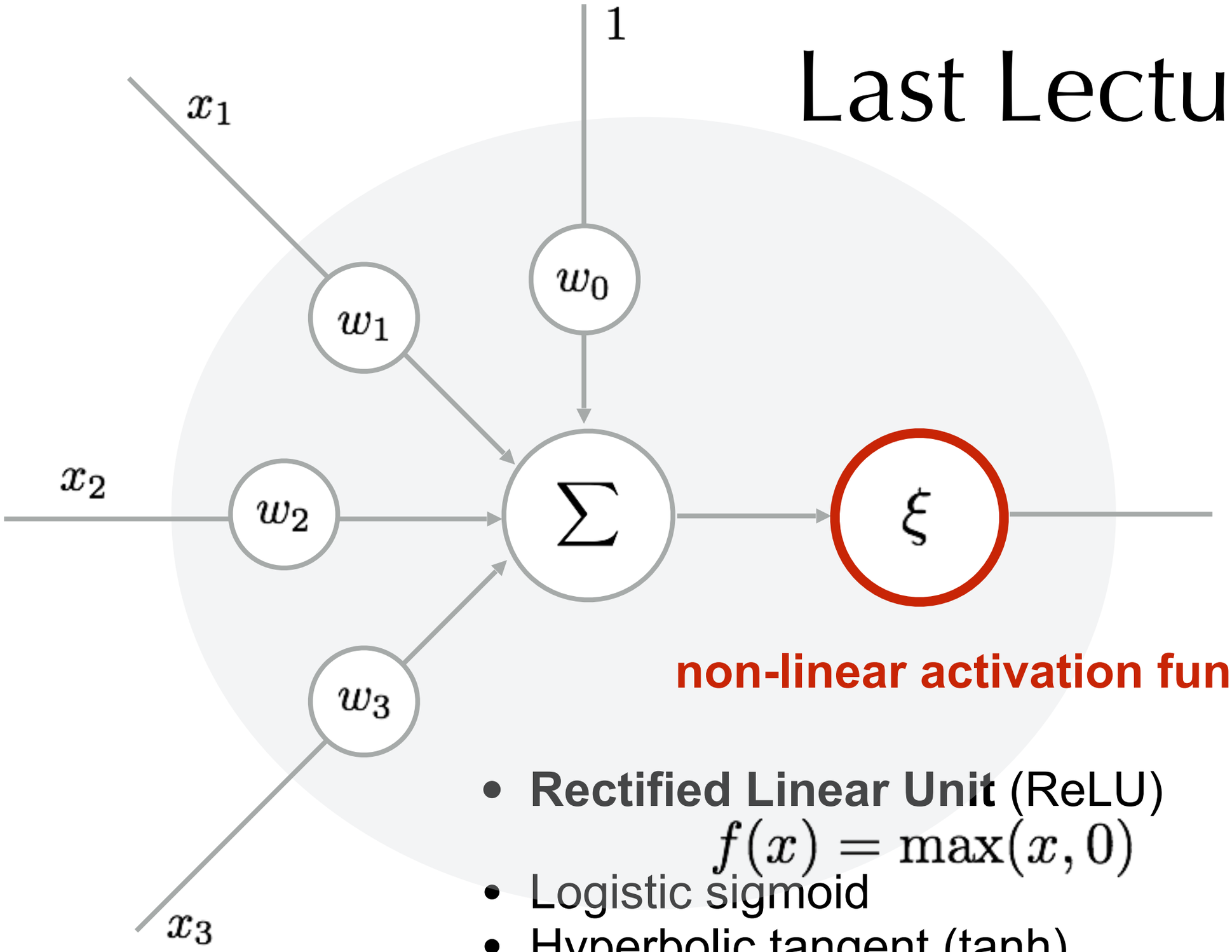
Torsten Sattler
(slides adapted from Olof Enqvist)

Last Lecture



Neural Networks

Last Lecture



non-linear activation function

- **Rectified Linear Unit** (ReLU)

$$f(x) = \max(x, 0)$$

- Logistic sigmoid
- Hyperbolic tangent (tanh)
- ...

A neuron

Last Lecture

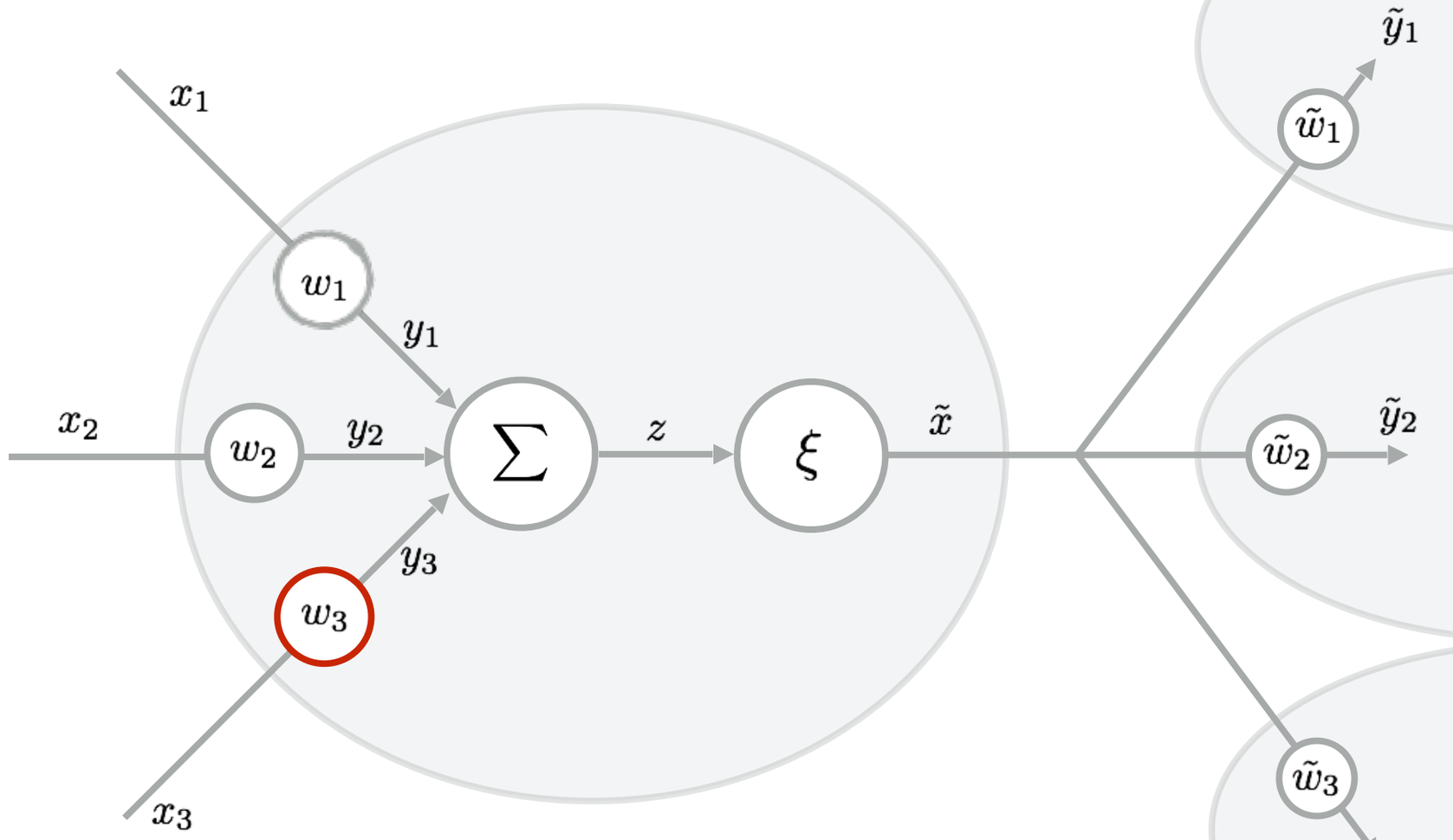
- Optimize all network parameters via gradient descent:

$$\theta^{(k+1)} = \theta^{(k)} - \mu \sum_i \nabla L_i(\theta) \approx \theta^{(k)} - \mu \nabla L_i(\theta)$$

- Compute derivative of partial loss for each parameter $\mathtt{w_k}$

Training a neural network

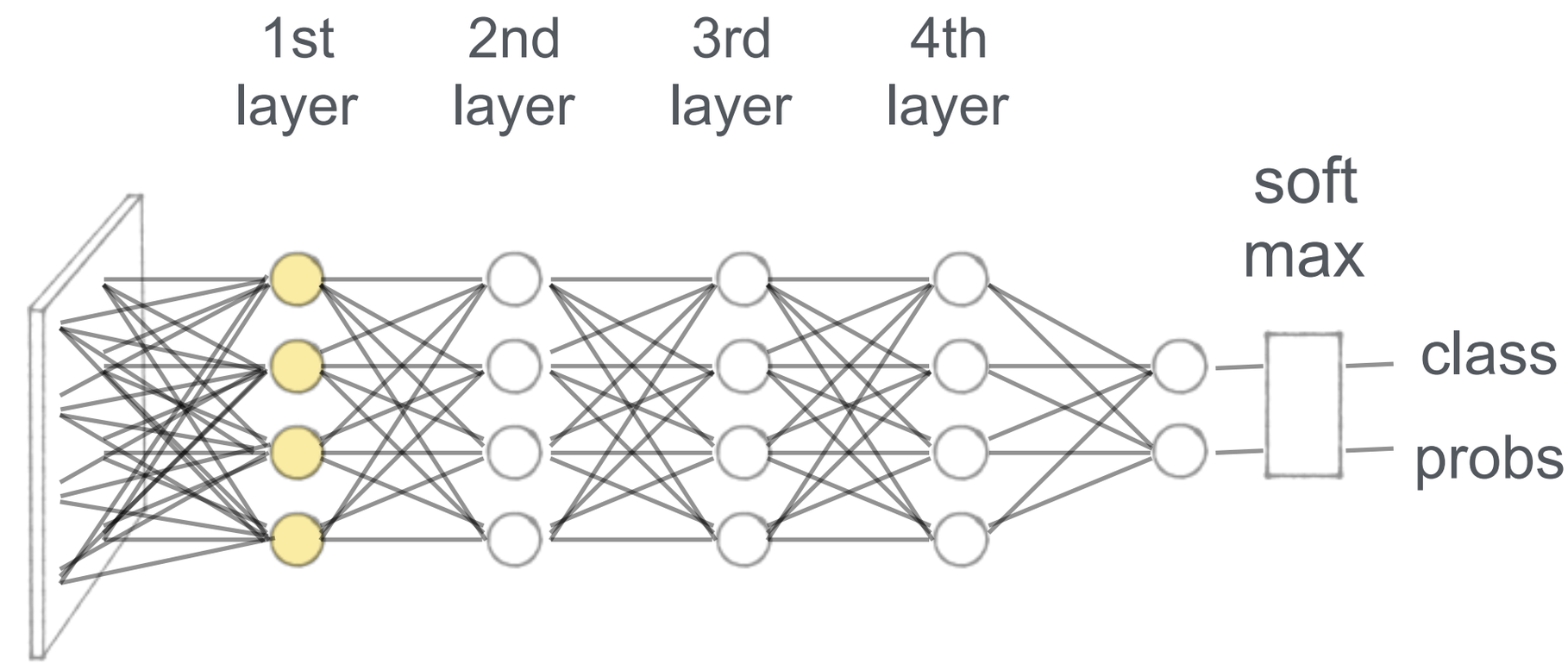
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation (backprop)

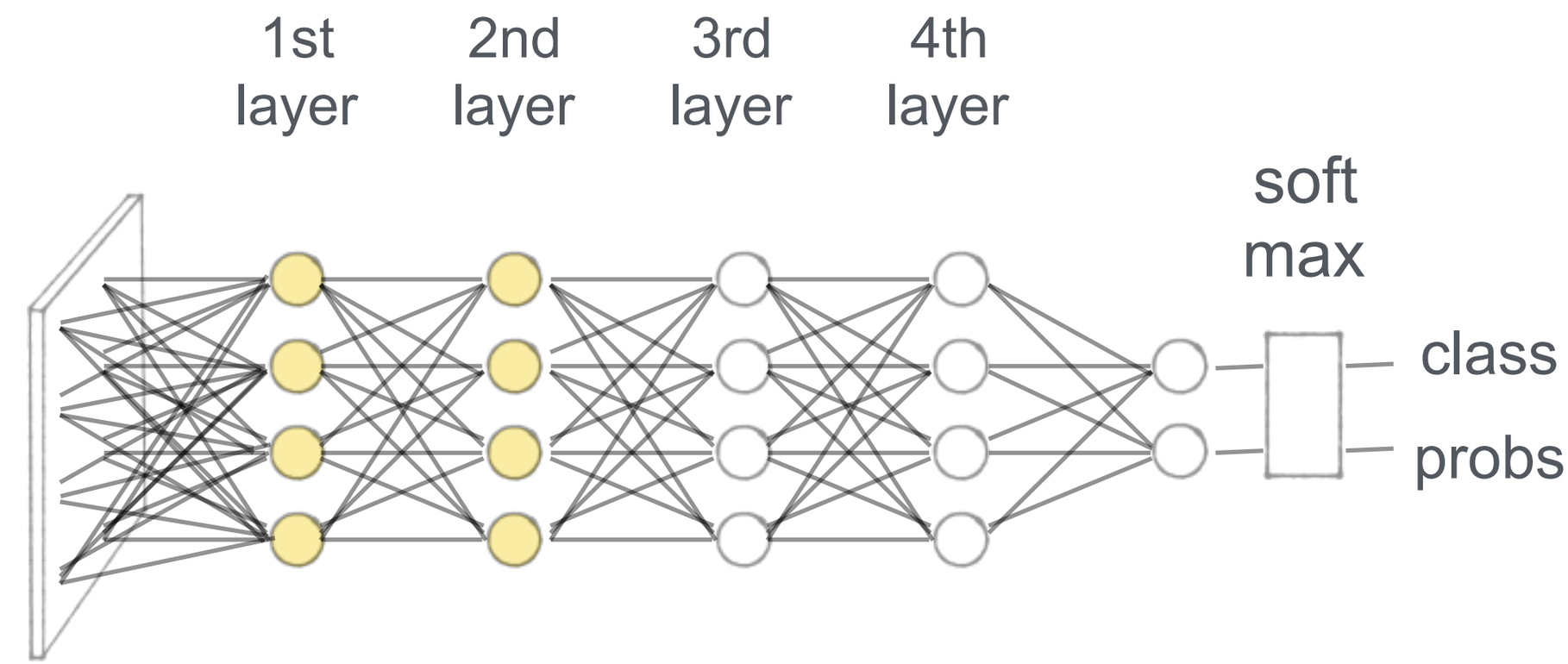
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

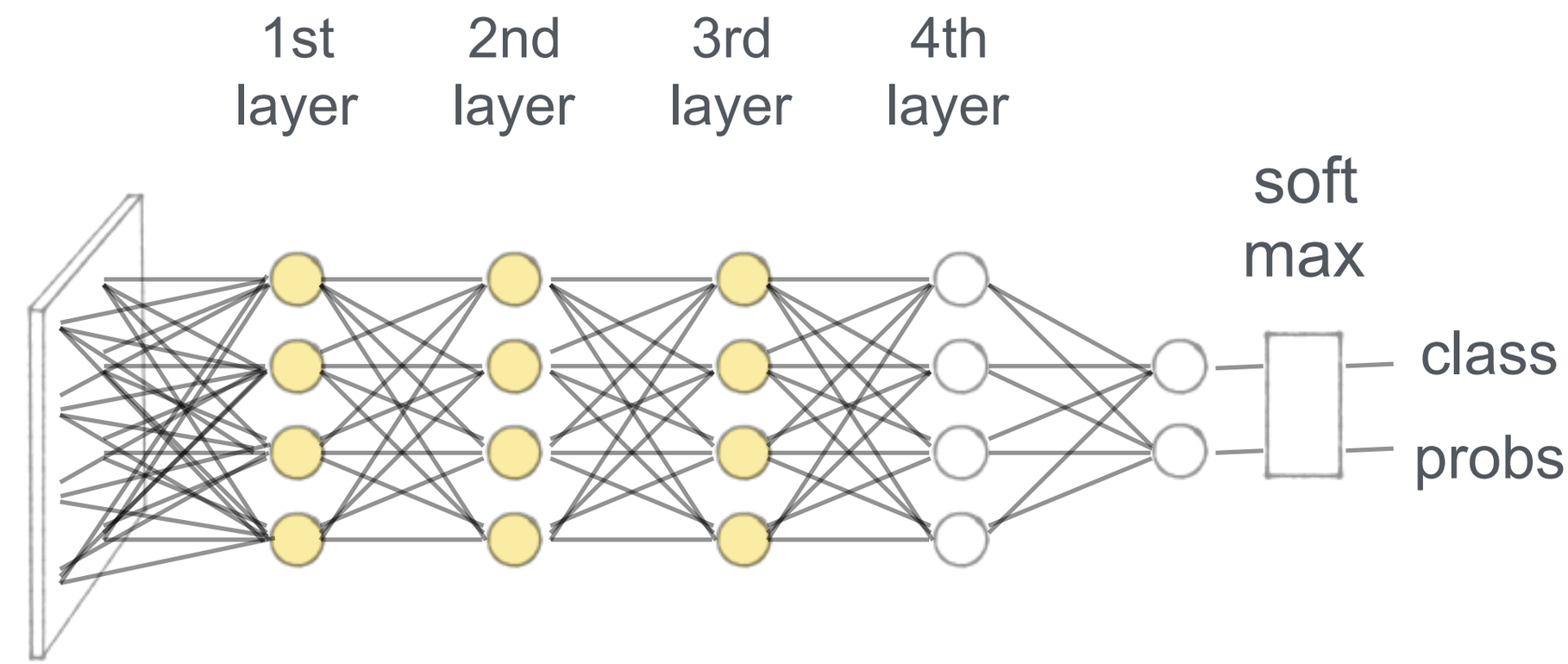
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

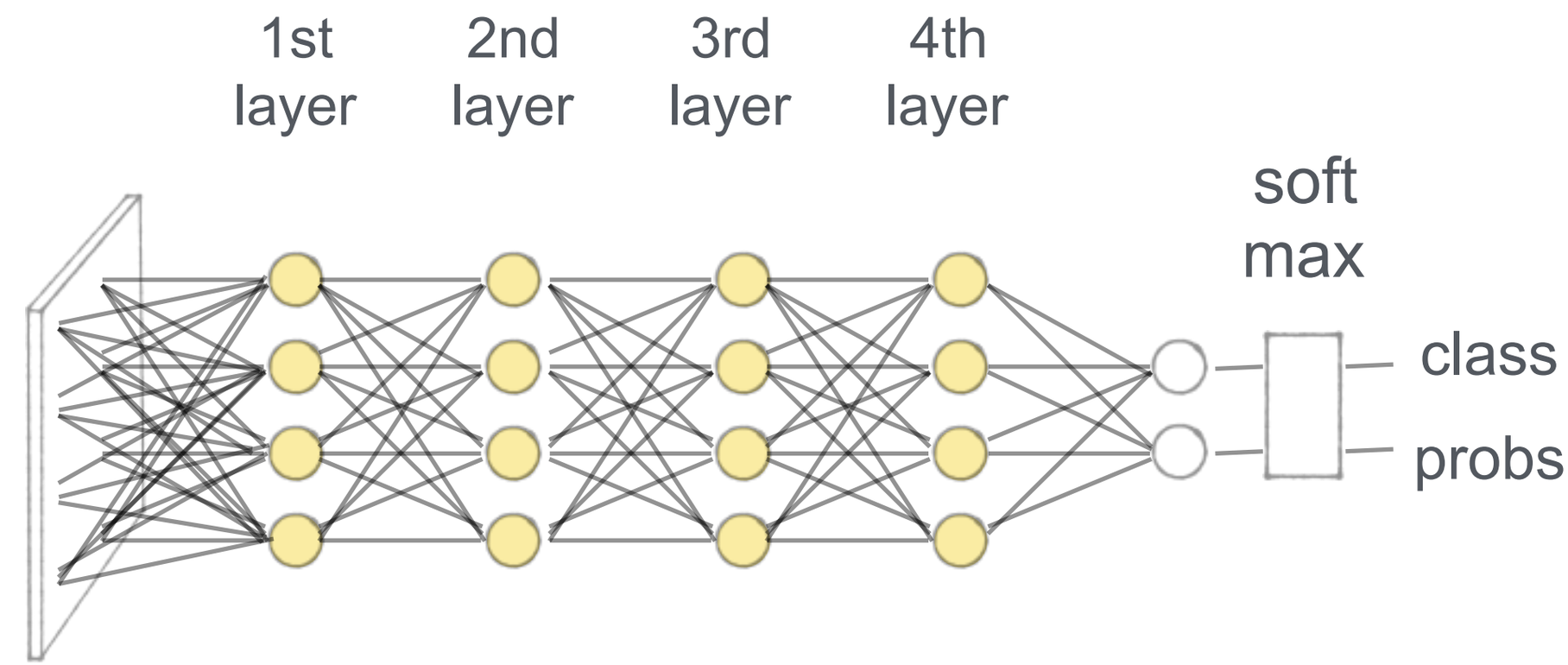
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

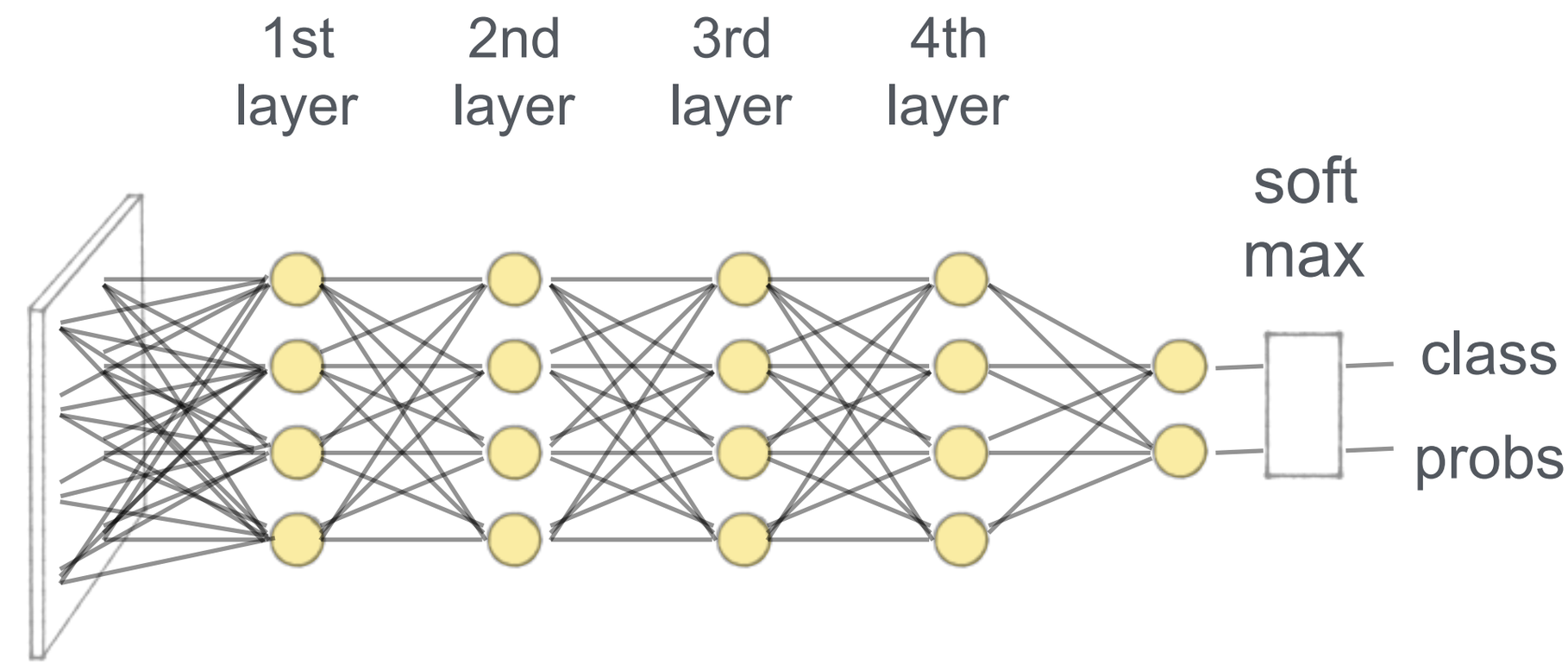
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

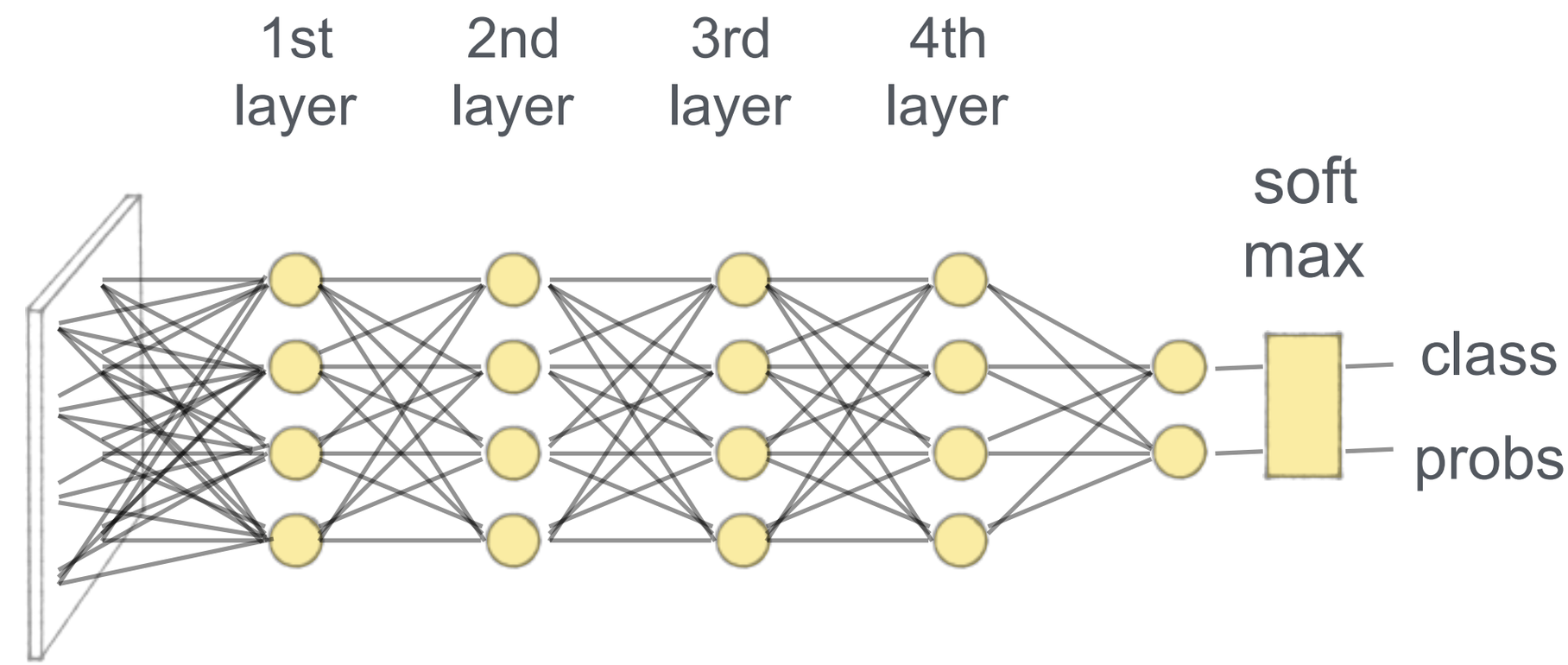
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

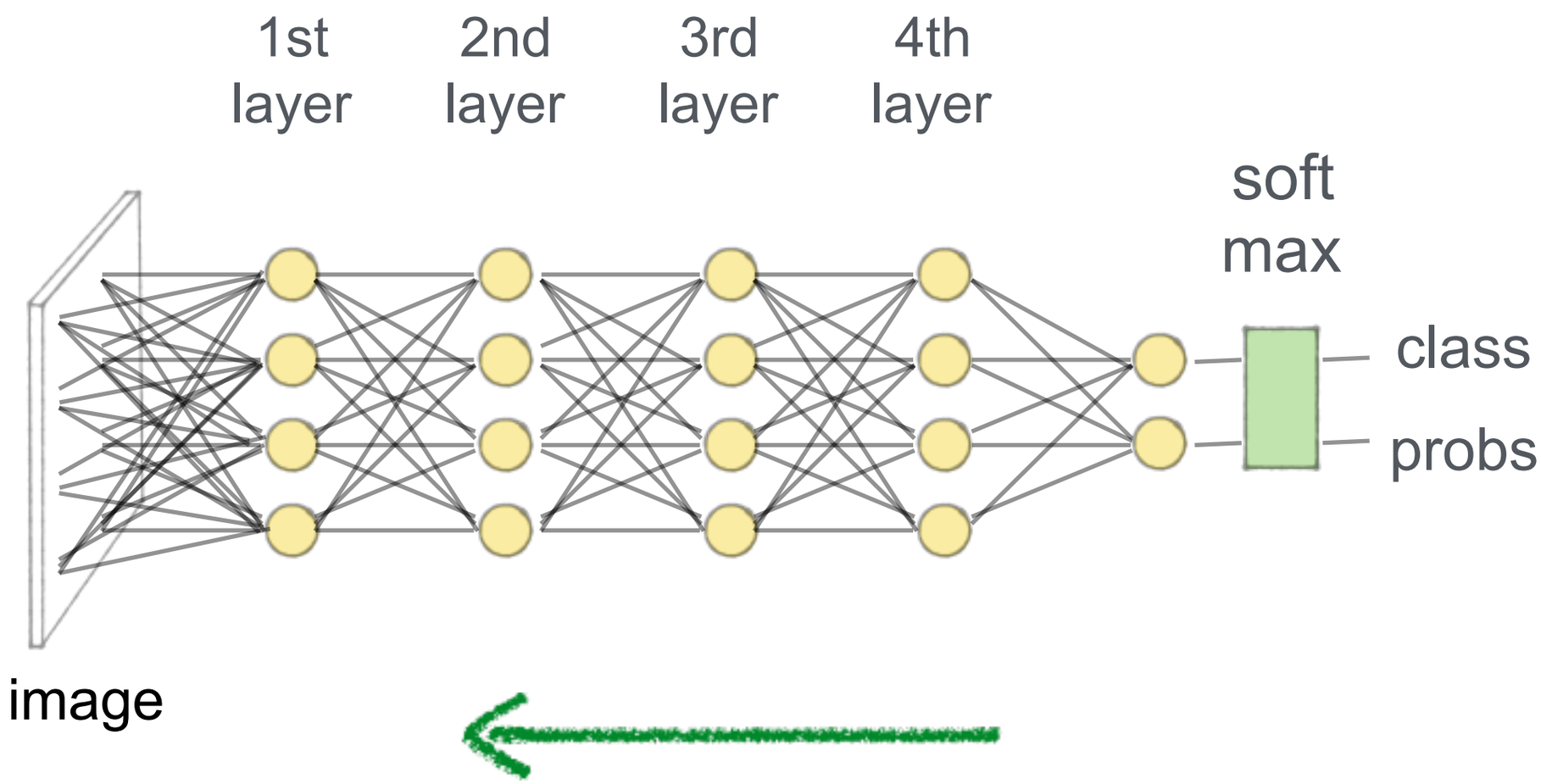
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

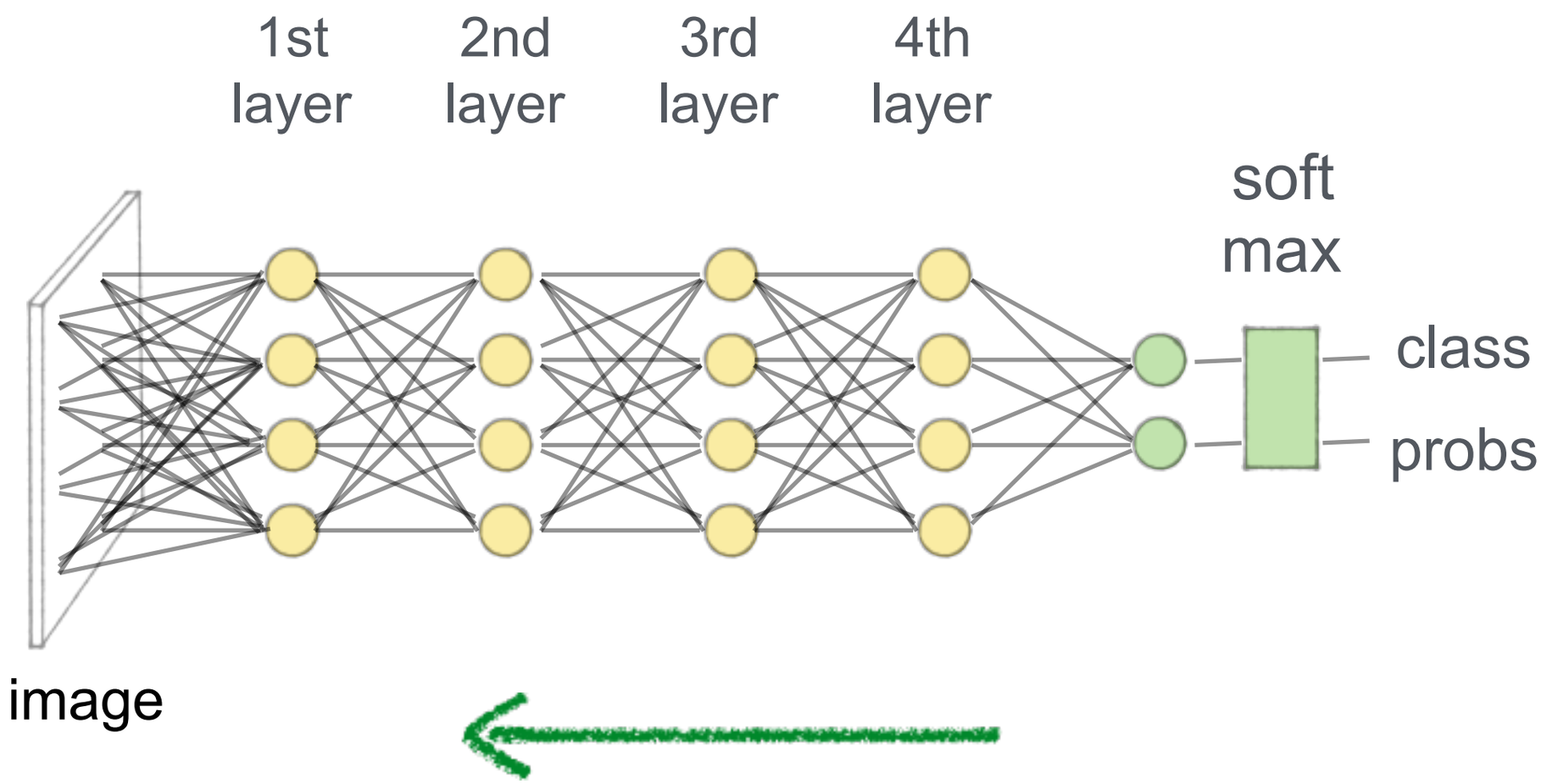
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

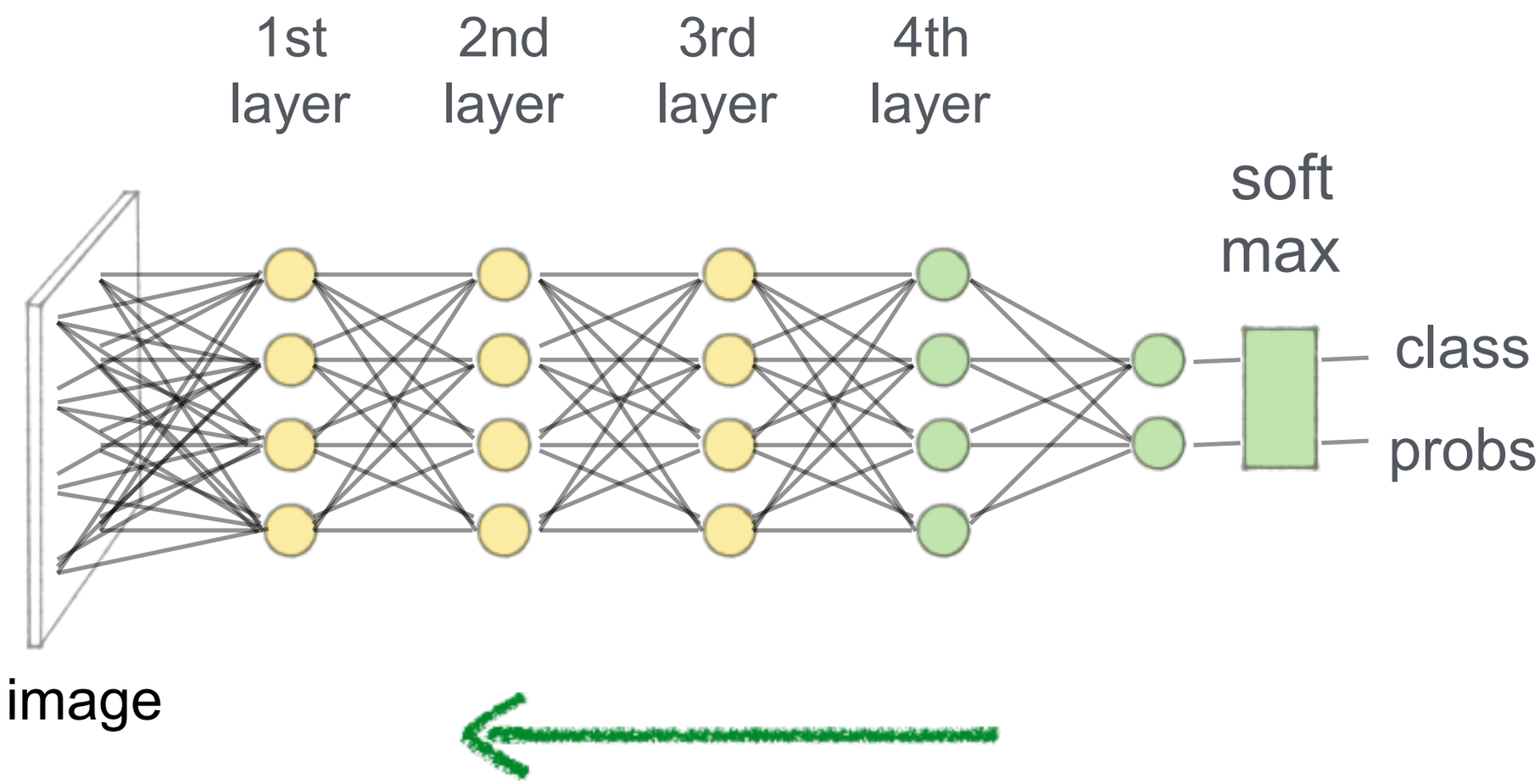
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

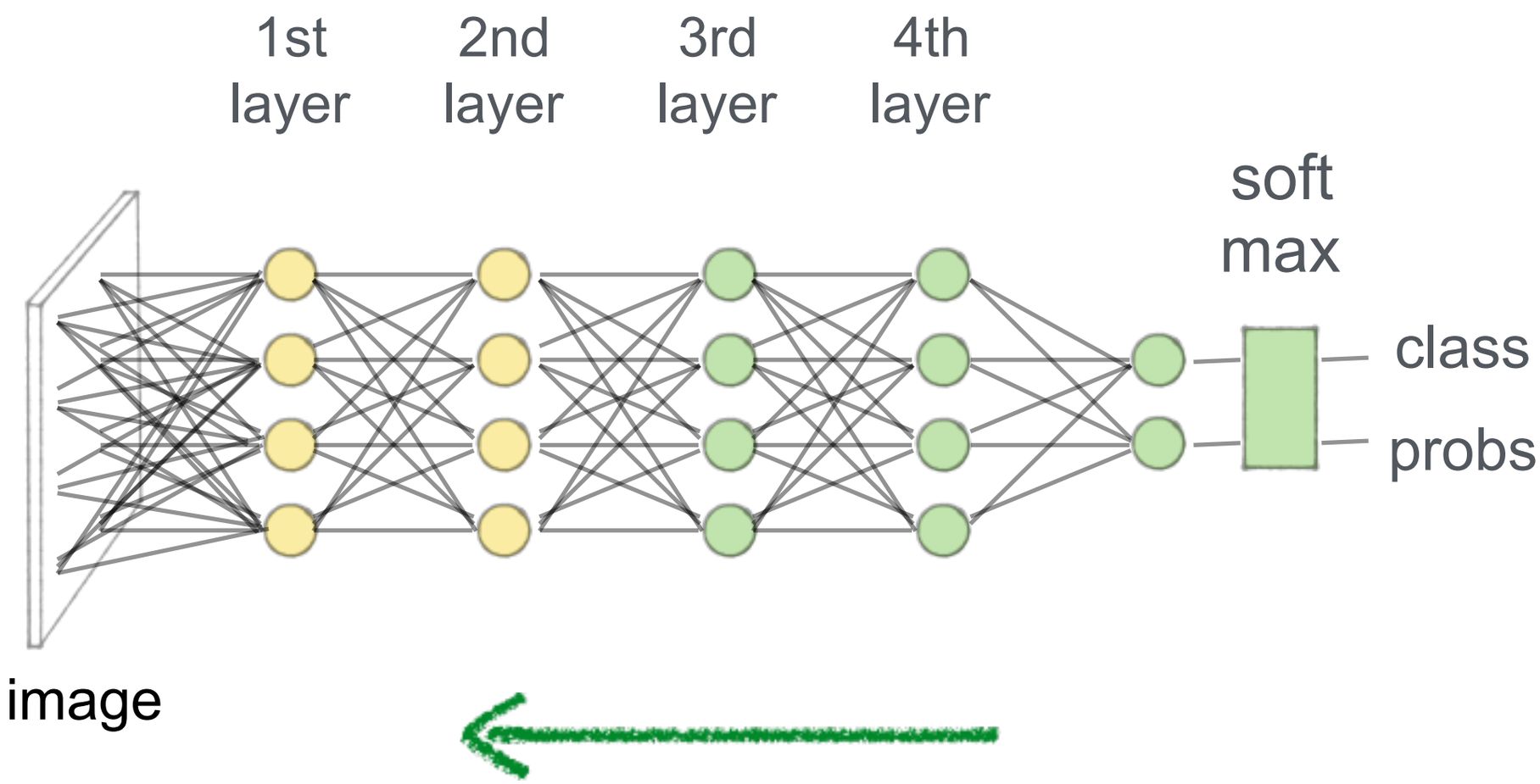
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

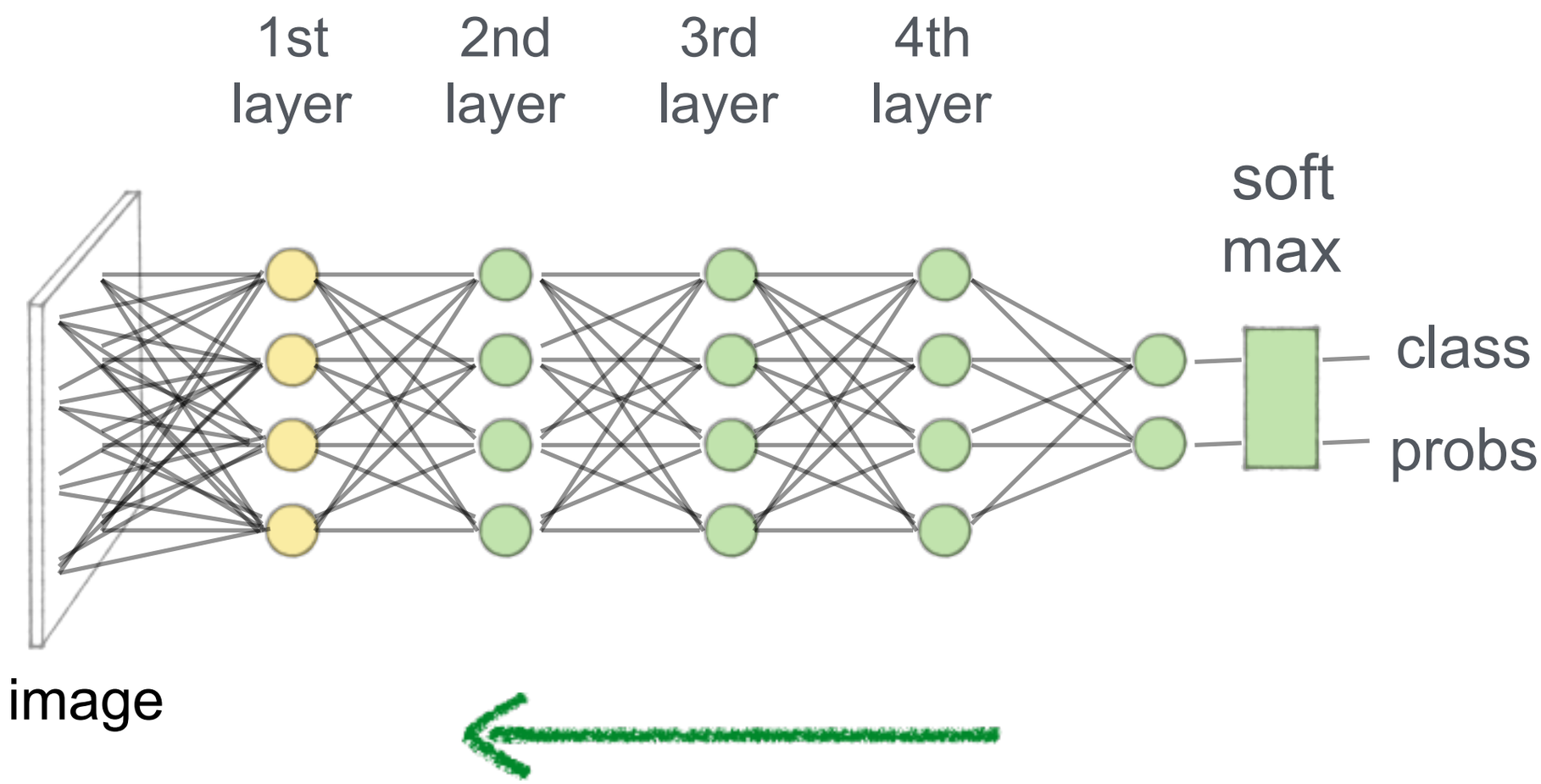
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

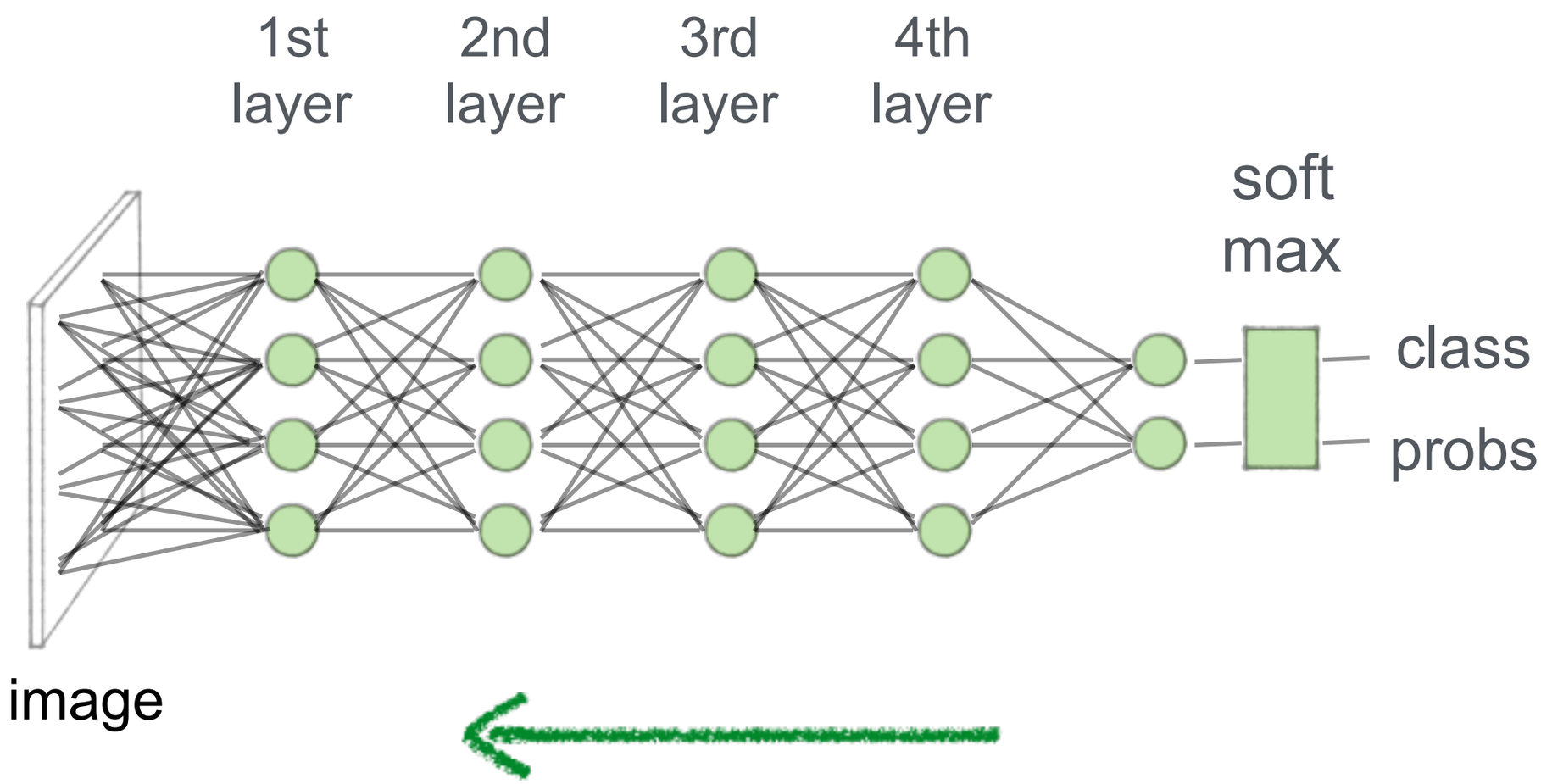
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Last Lecture



Backpropagation (backprop)

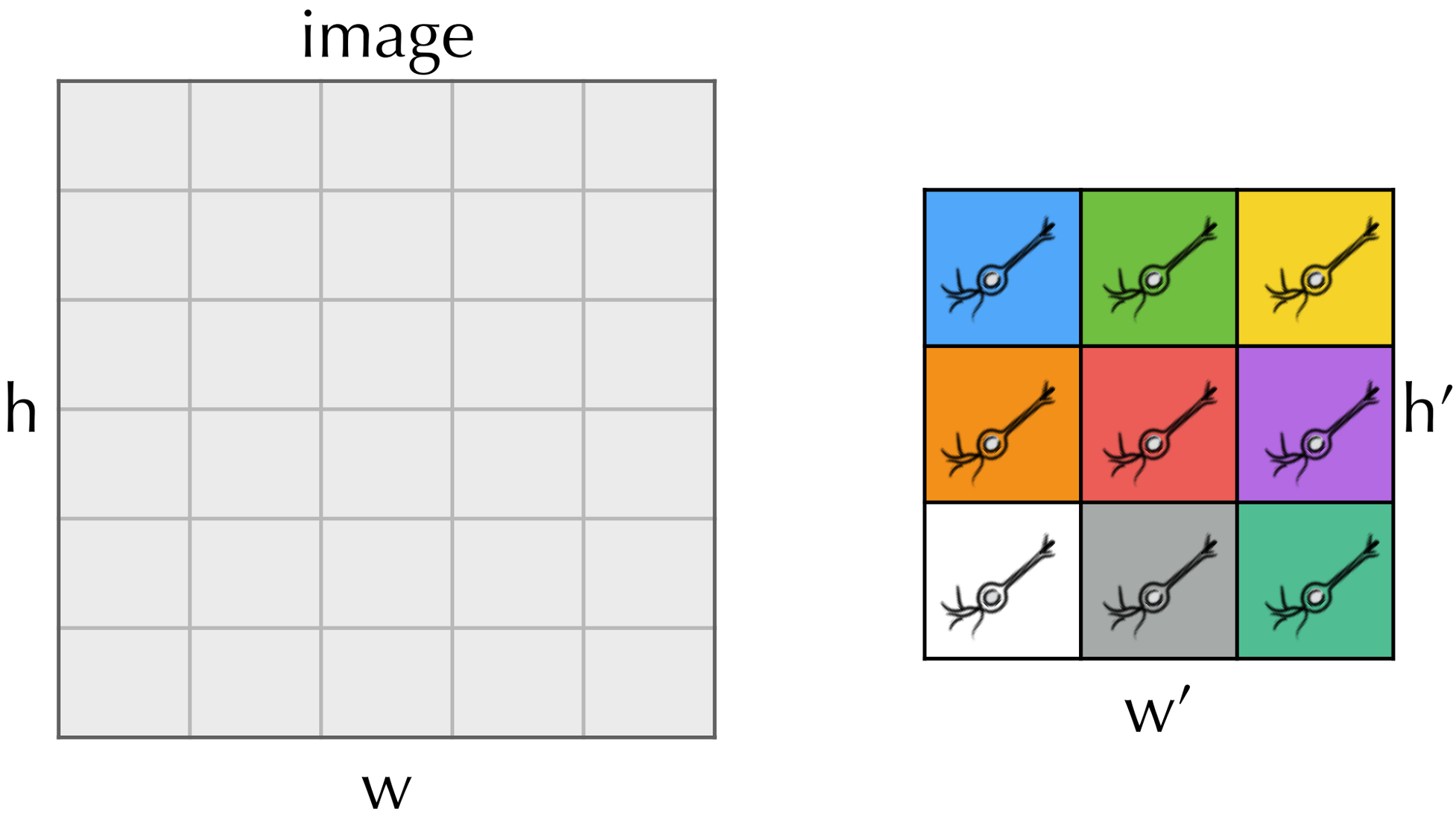
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Today

- Convolutional Neural Networks
- Fully Convolutional Neural Networks
- Convolutional Layers
- Overfitting, Part II

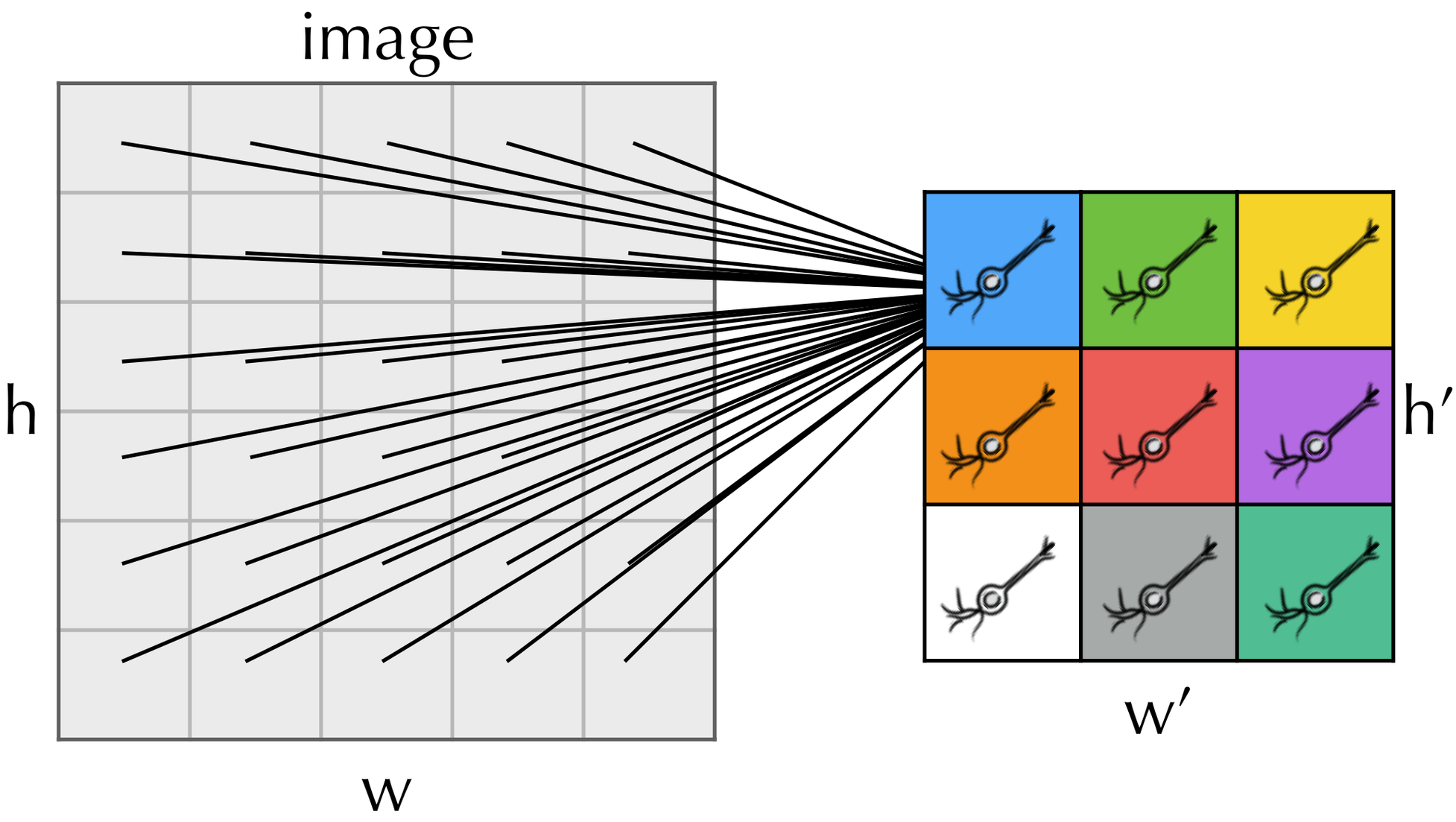
Convolutional Neural Networks

Fully Connected Layers



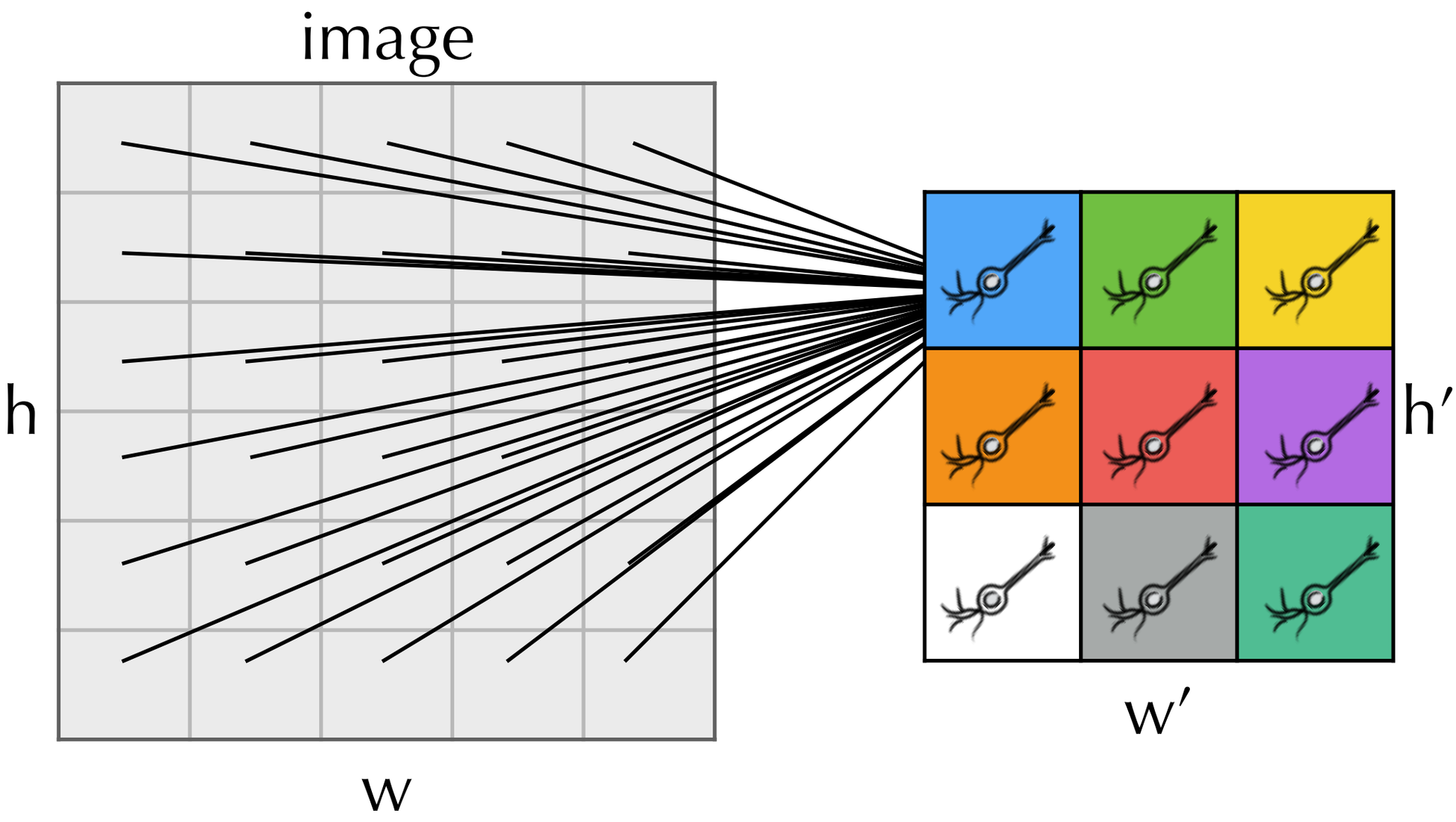
$h \cdot w \cdot h' \cdot w'$

Fully Connected Layers



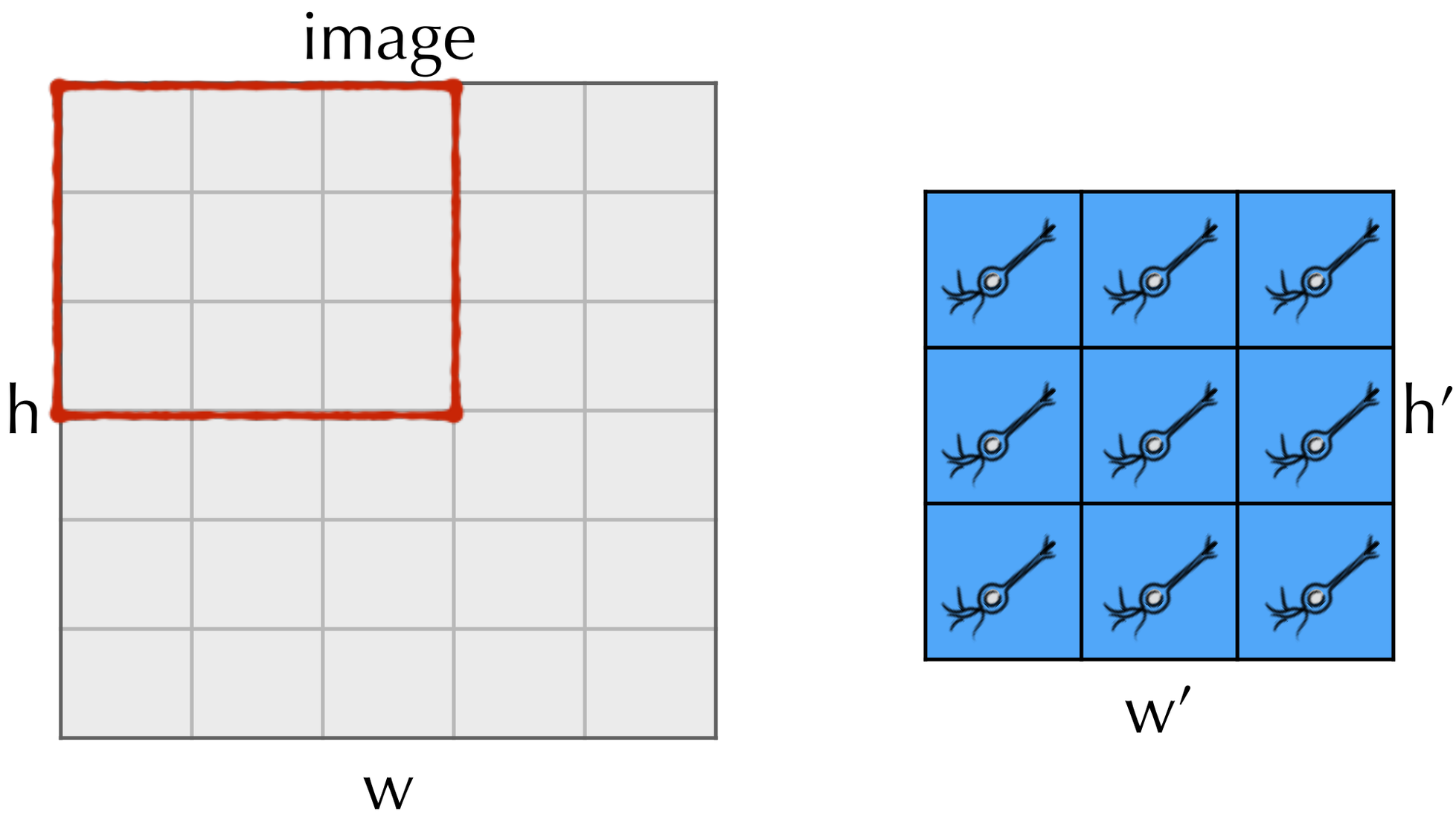
$h \cdot w \cdot h' \cdot w'$

Fully Connected Layers

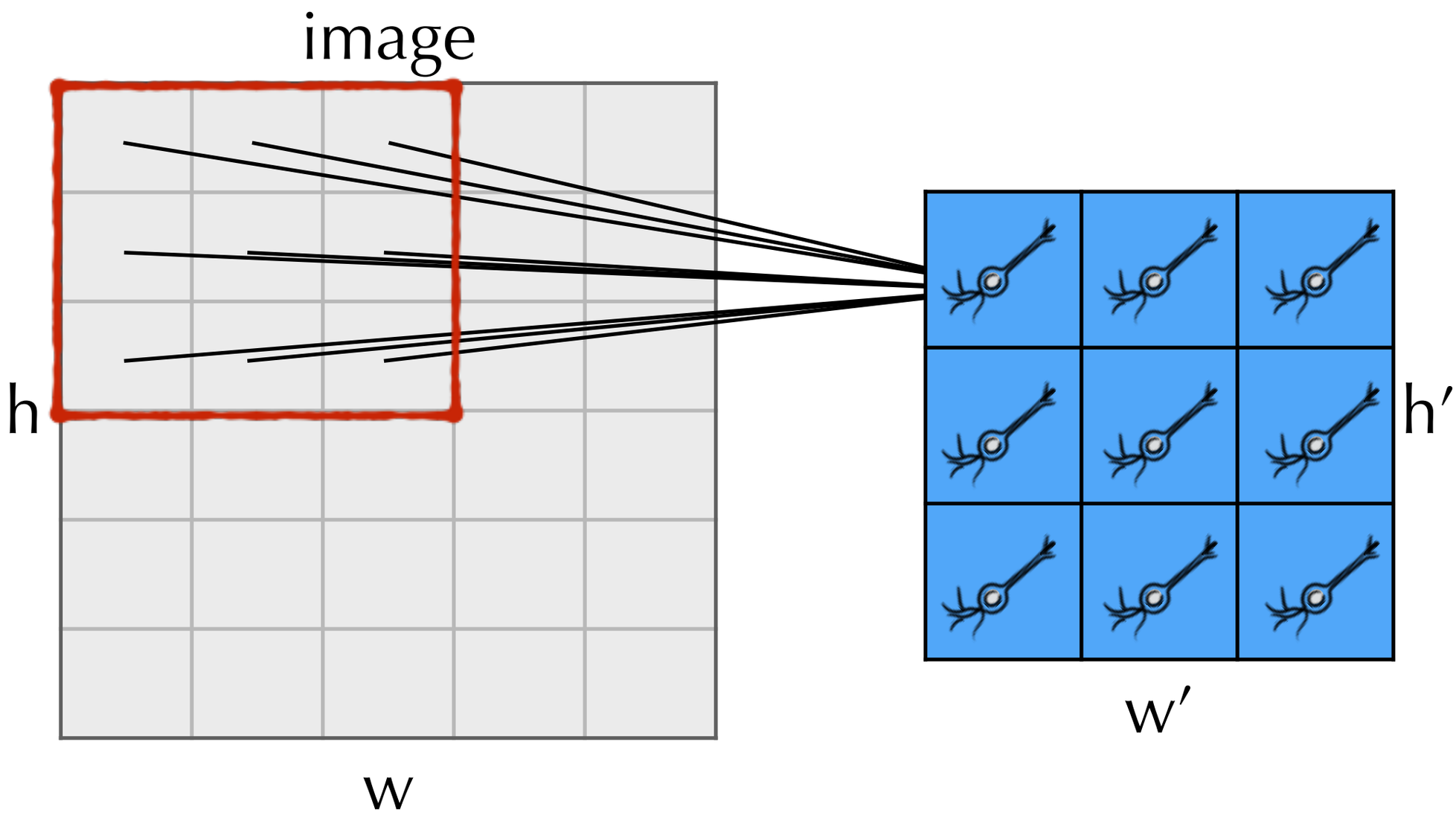


Number parameters: $h \cdot w \cdot h' \cdot w'$

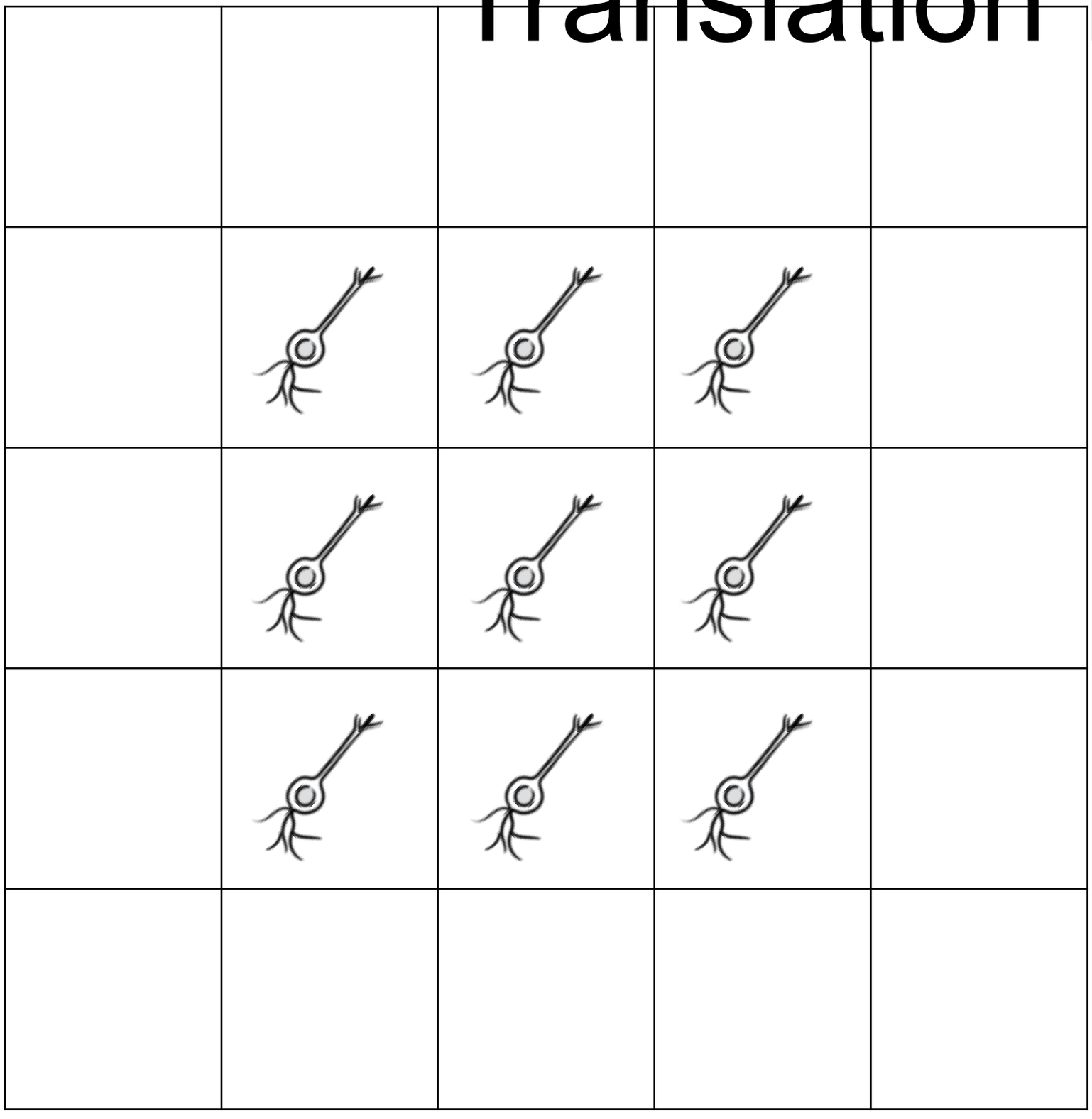
Convolutional Layers



Convolutional Layers

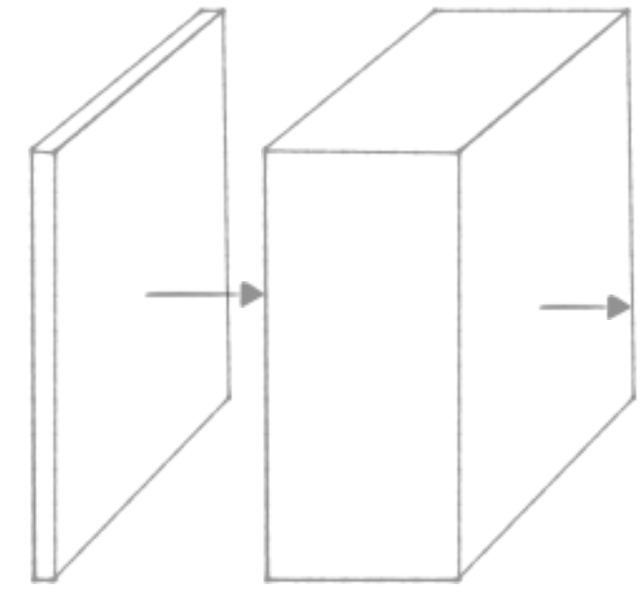


Translation Invariance



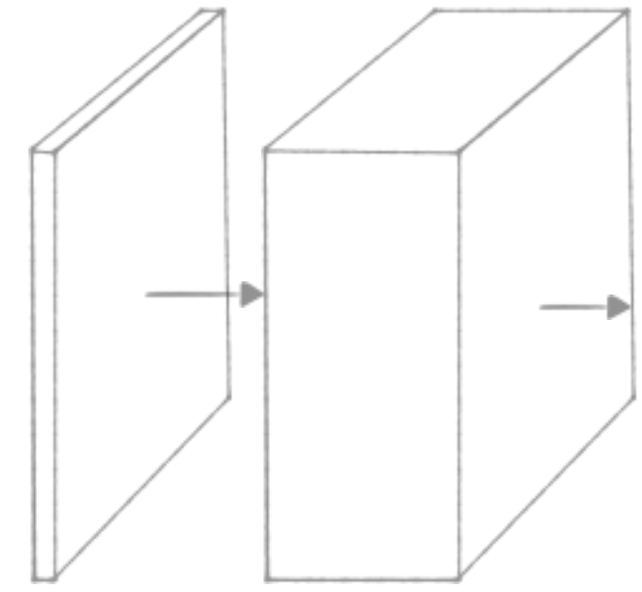
$$I \star \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Network Structure



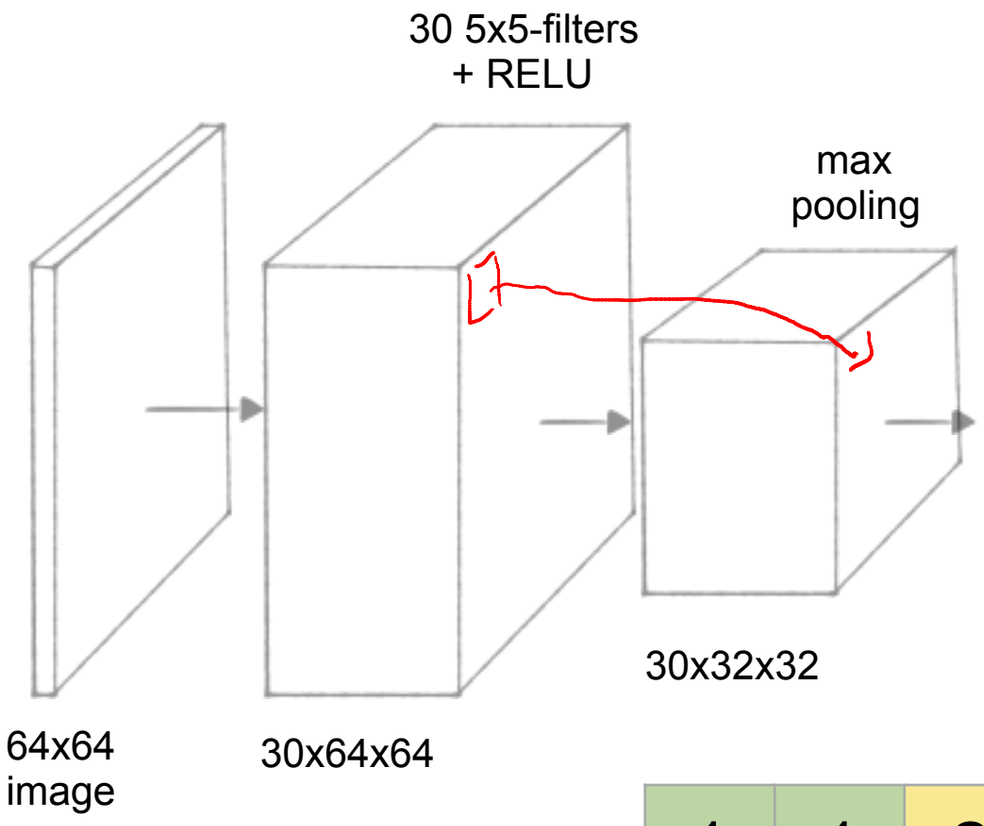
$$I \star w$$

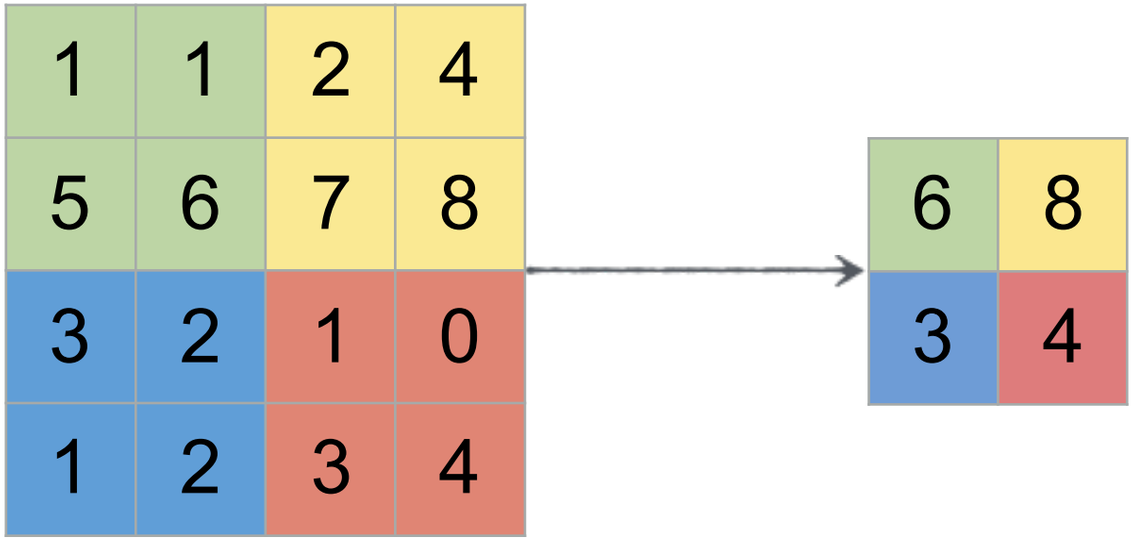
Network Structure



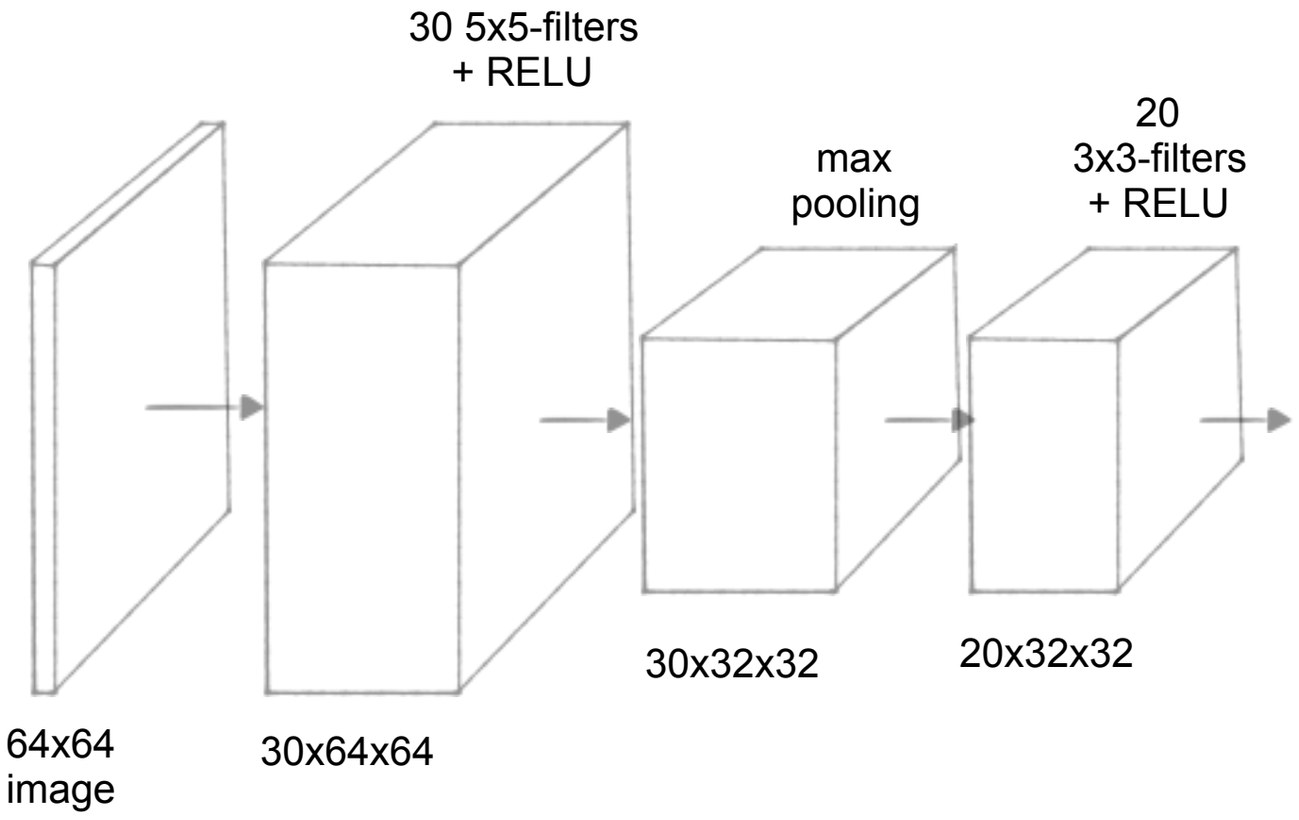
$$\max\{I \star w, 0\}$$

Network Structure

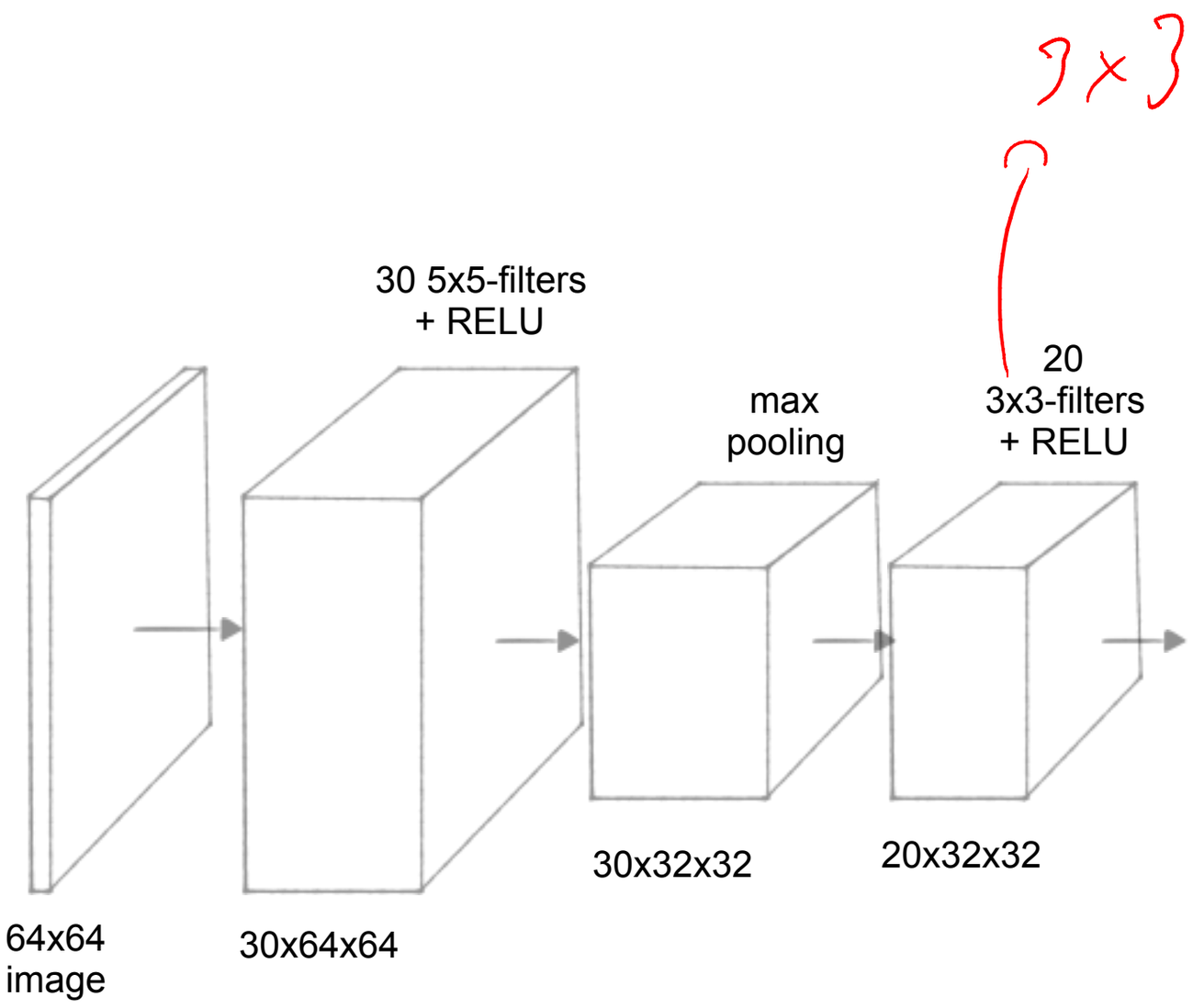


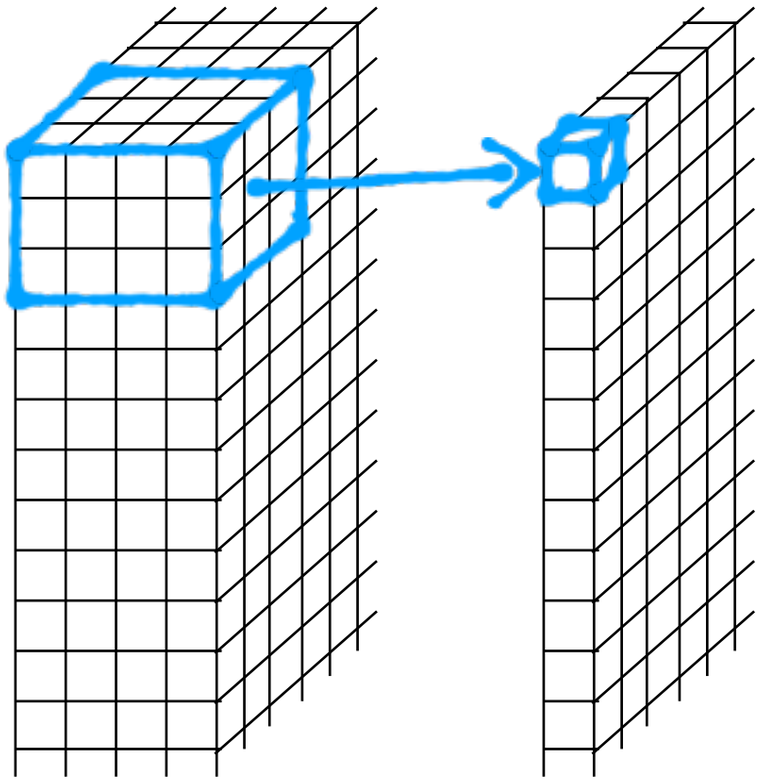


Network Structure



Network Structure

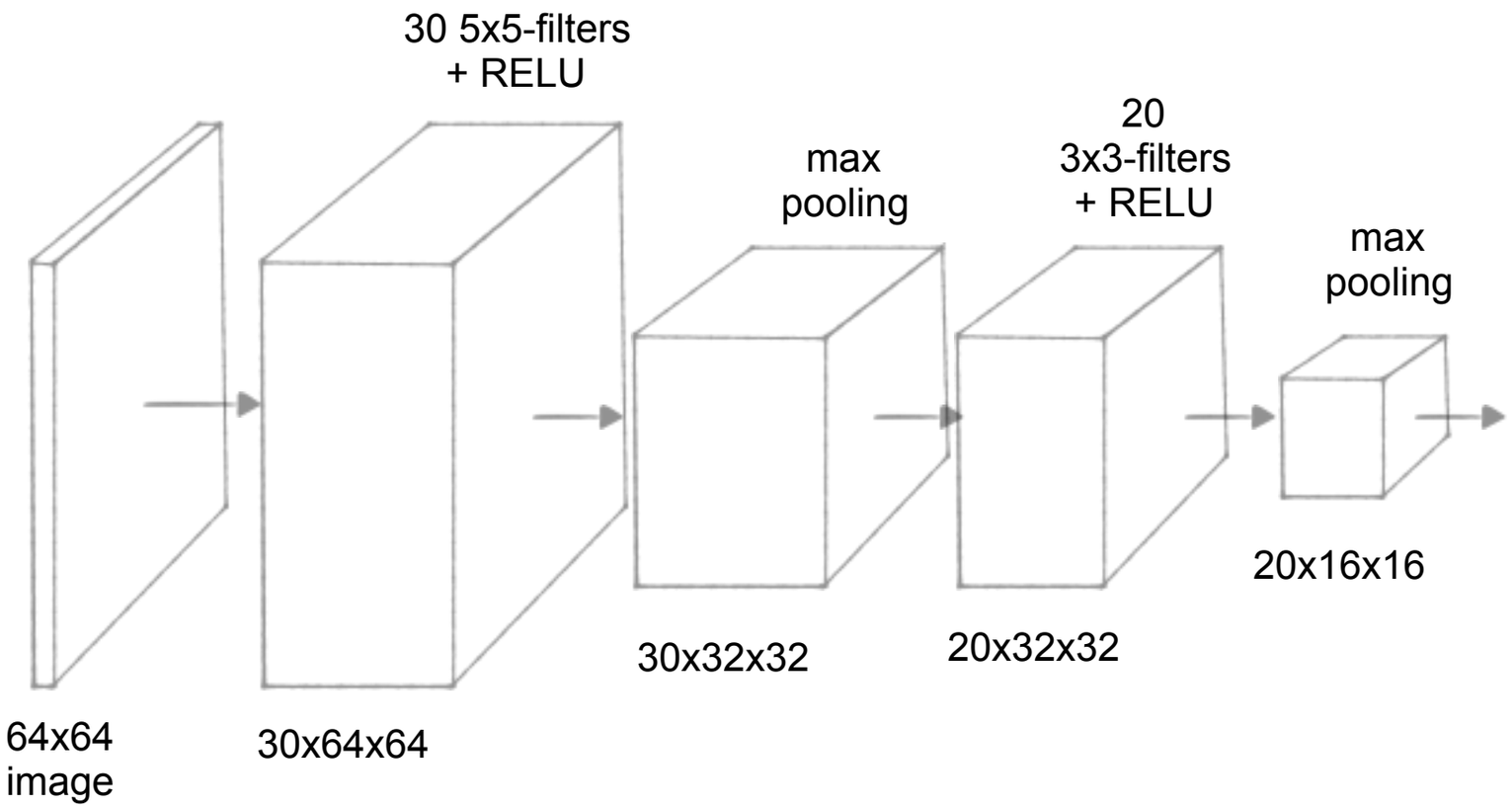




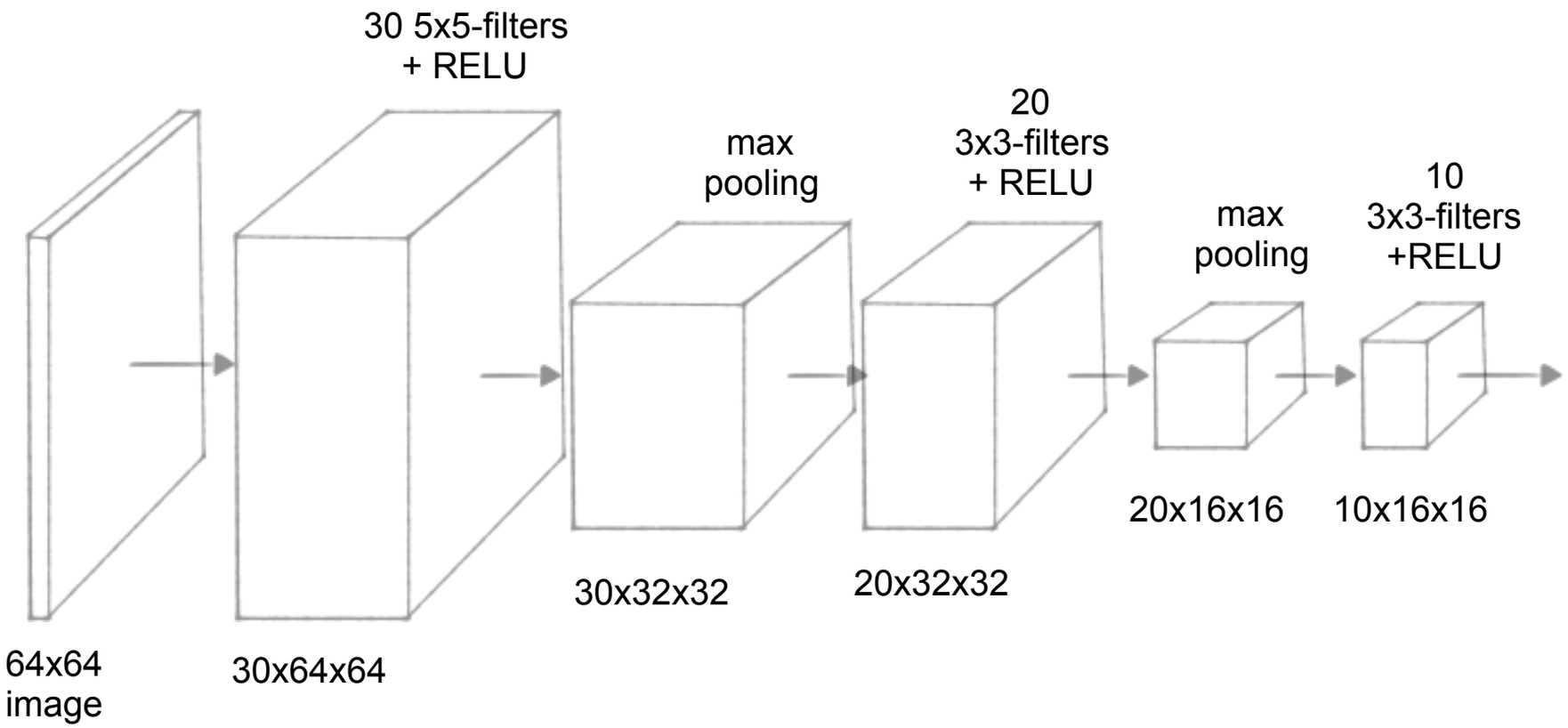
implemented as matrix multiplication:

$$\mathrm{vec}(\mathcal{J}) = F\mathrm{vec}(\mathcal{I})$$

Network Structure

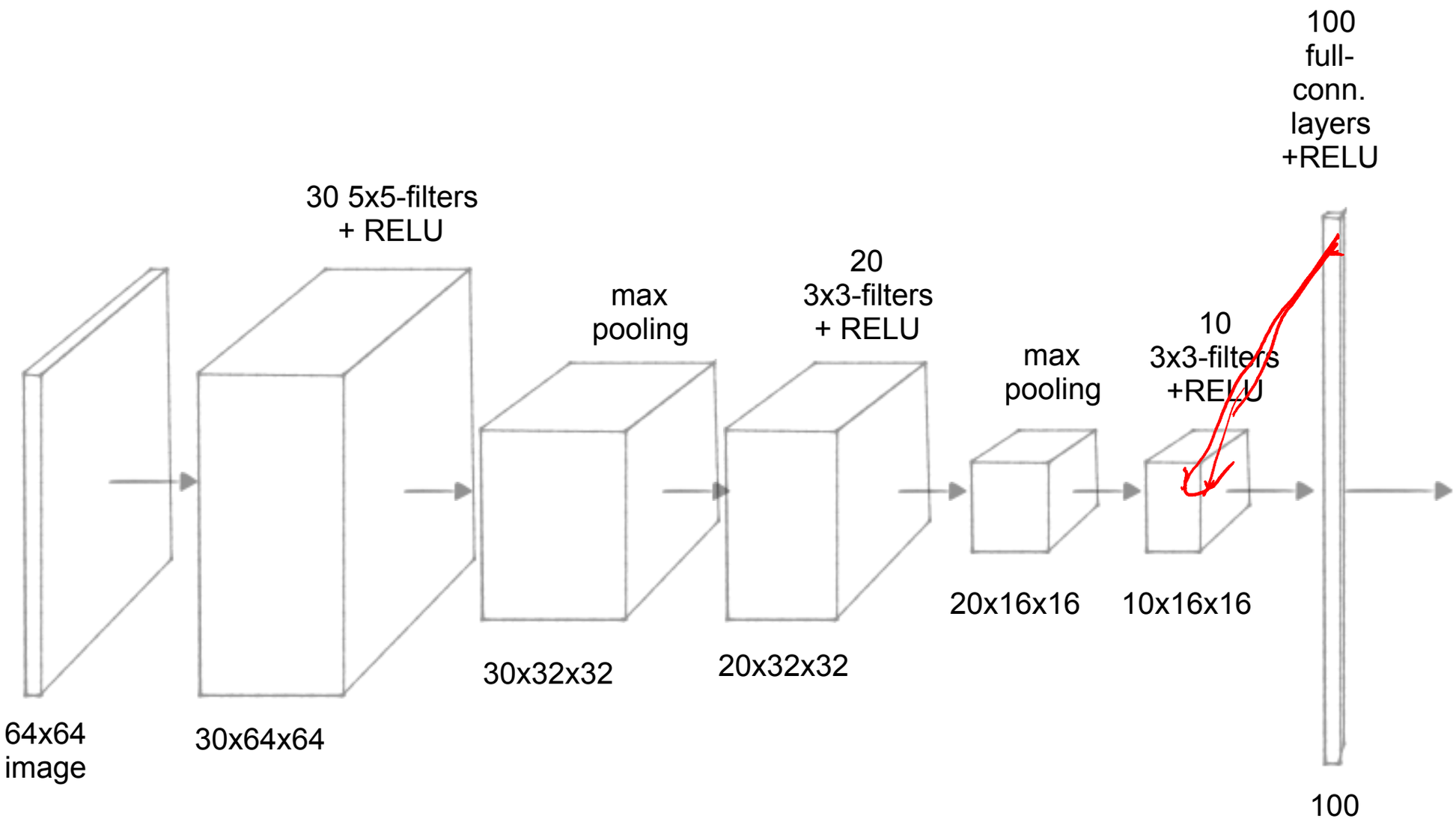


Network Structure



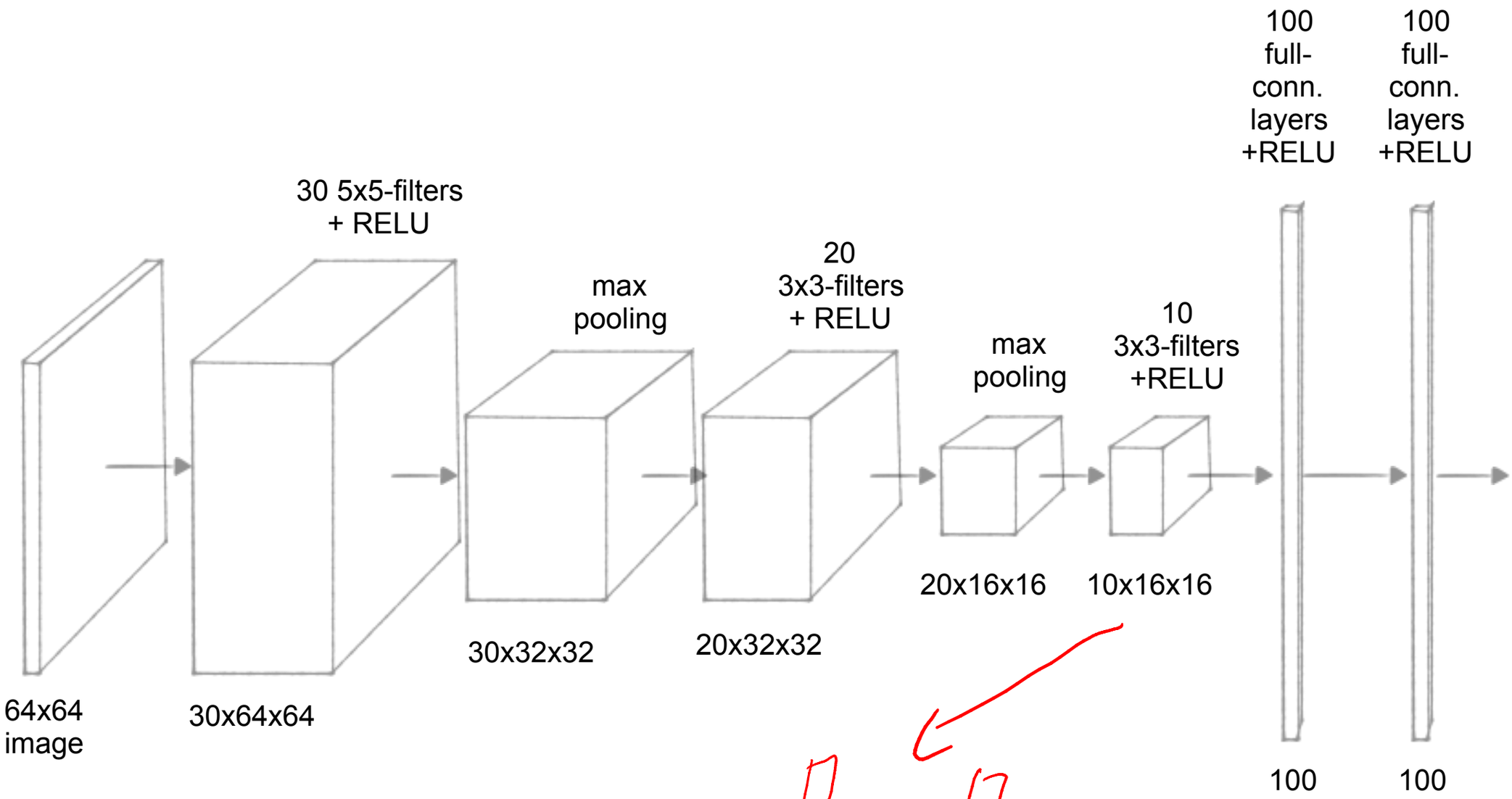
$$\max\{I \otimes f, 0\}$$

Network Structure

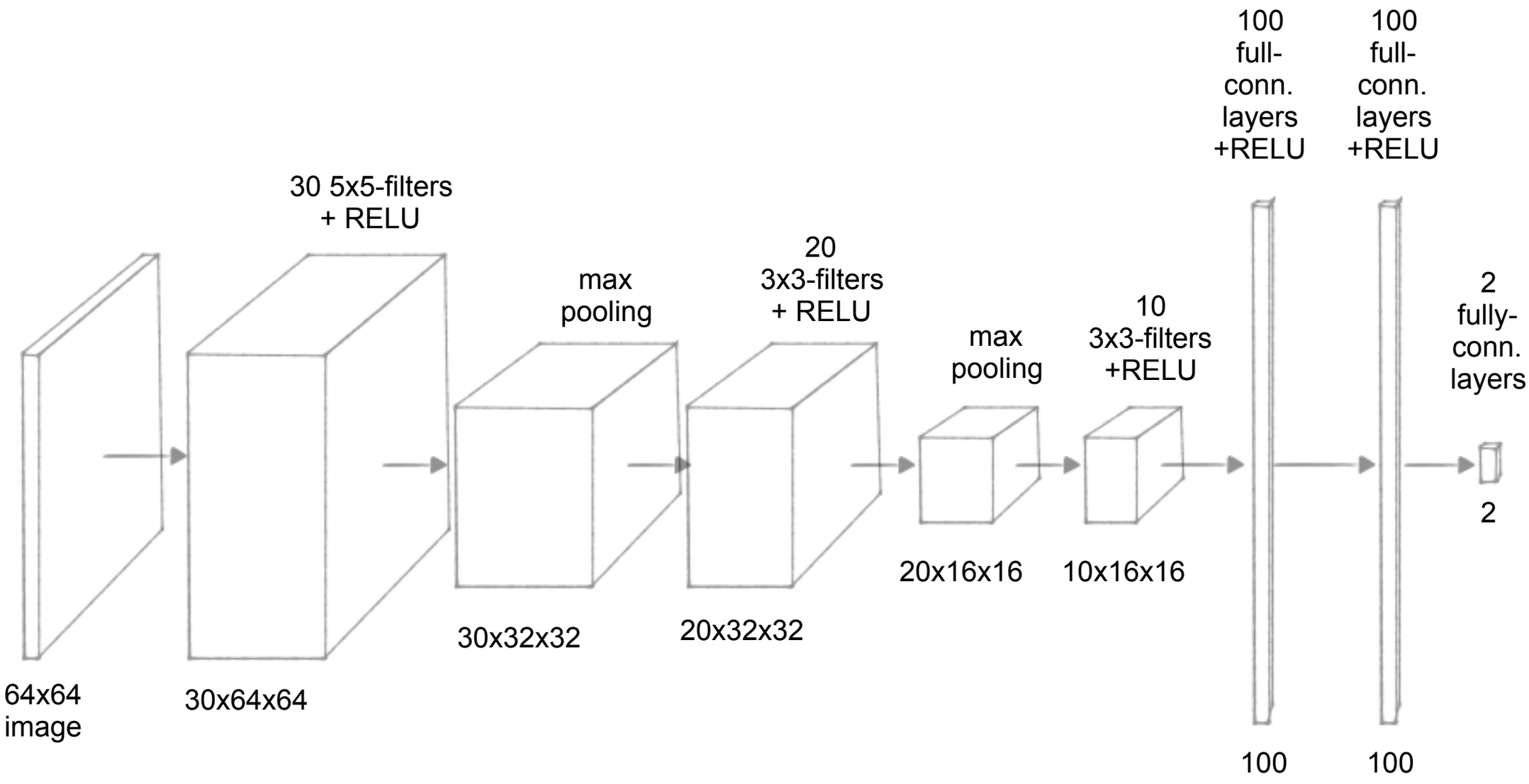


$$\sum_{i=1}^{10}\sum_{u=1}^{16}\sum_{v=1}^{16} w_{ijk}x_{ijk}$$

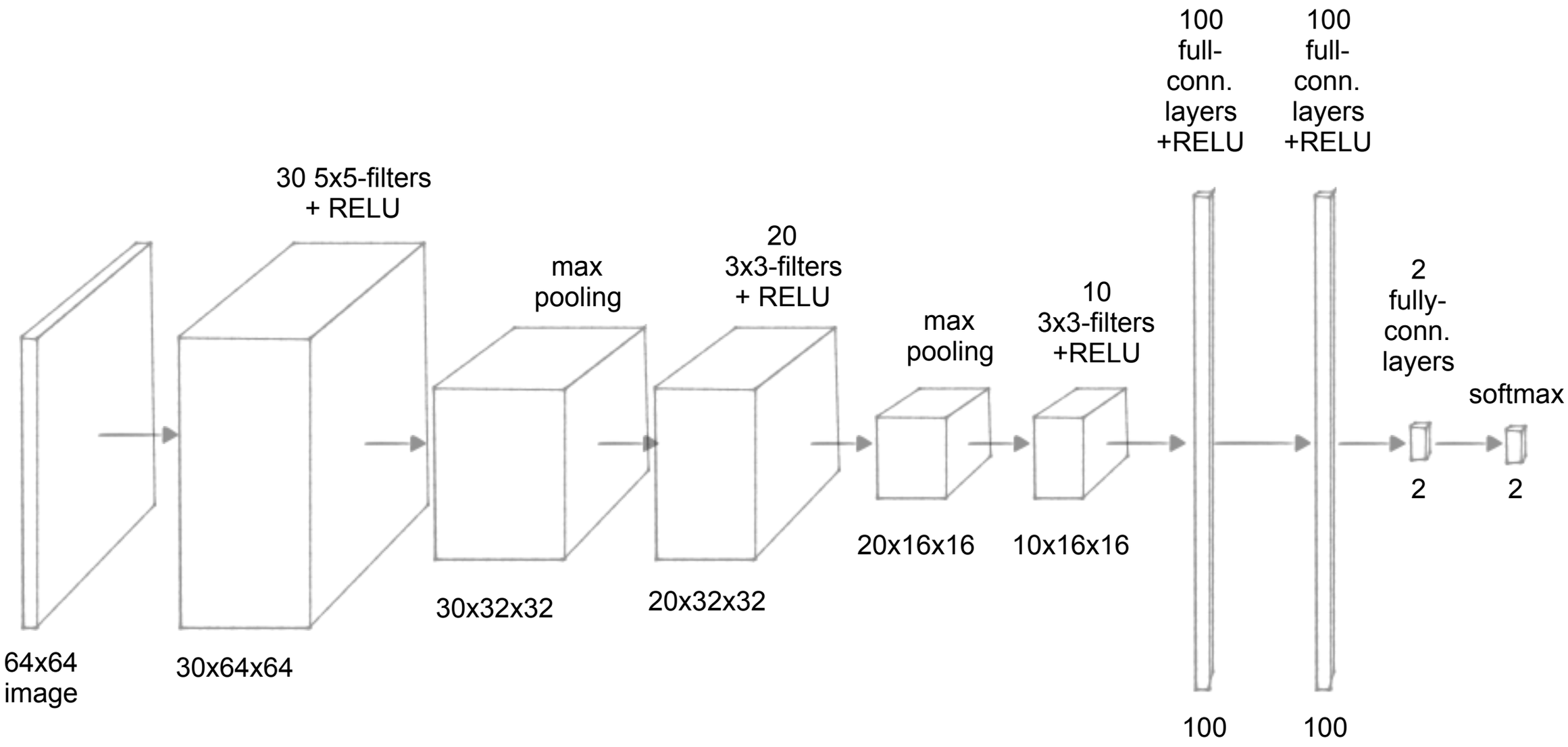
Network Structure



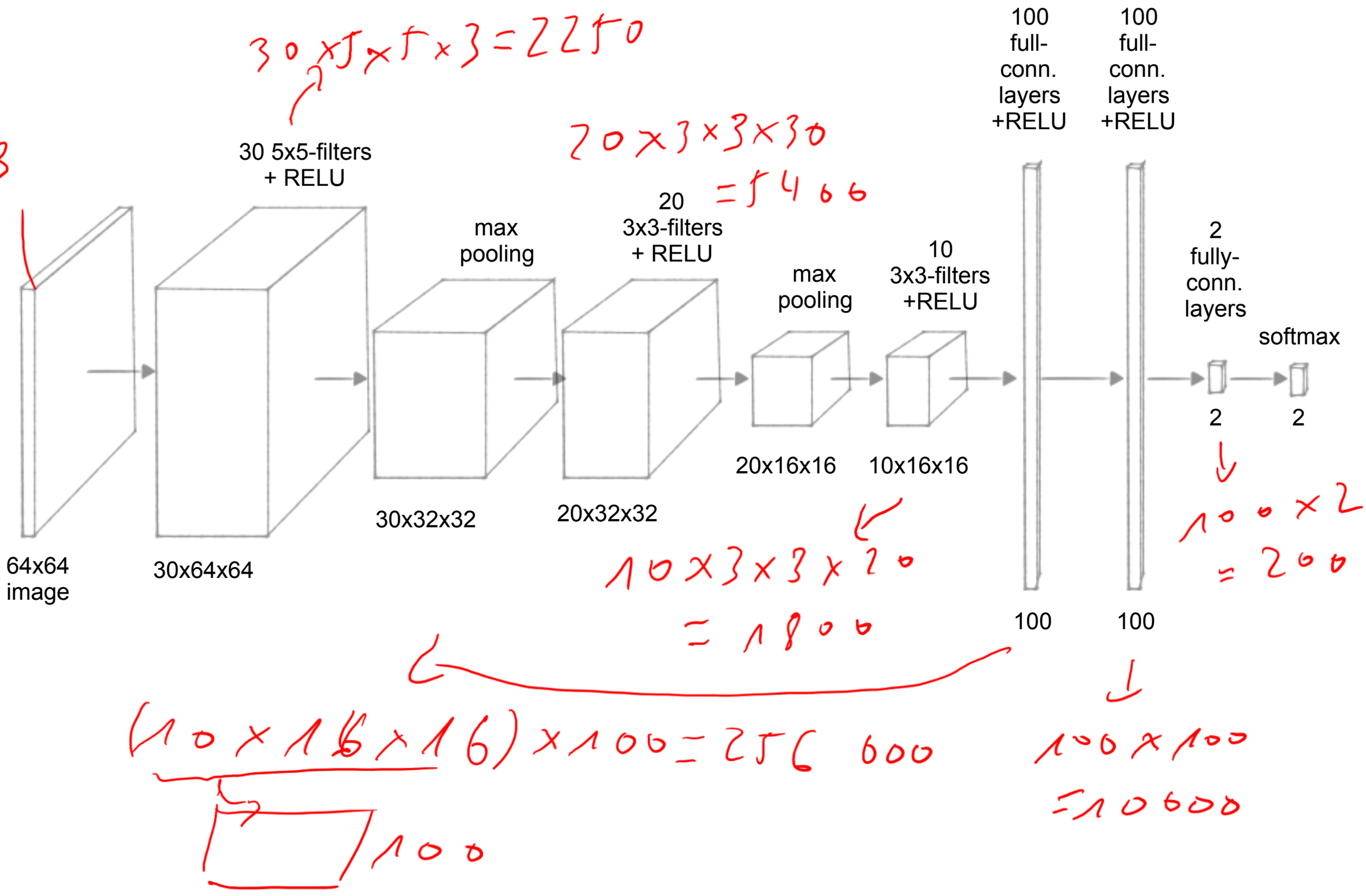
Network Structure



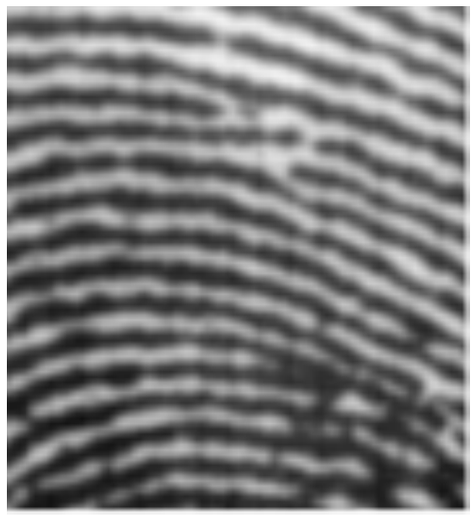
Network Structure

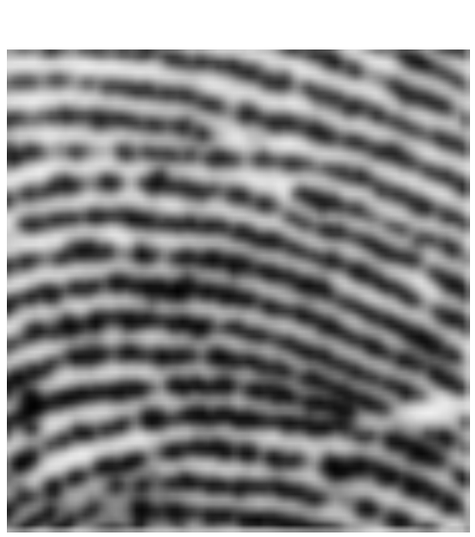


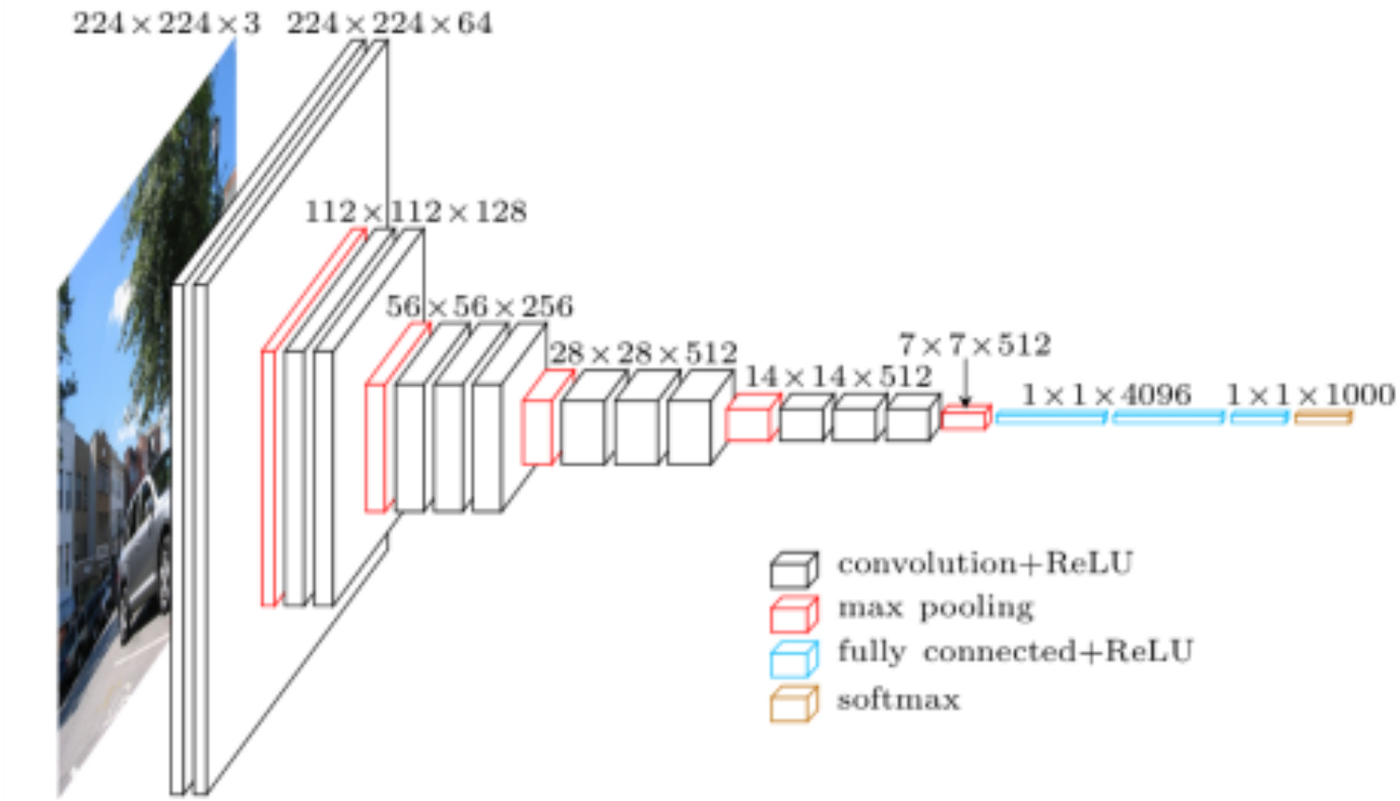
Number Parameters?



Live or Spoof?





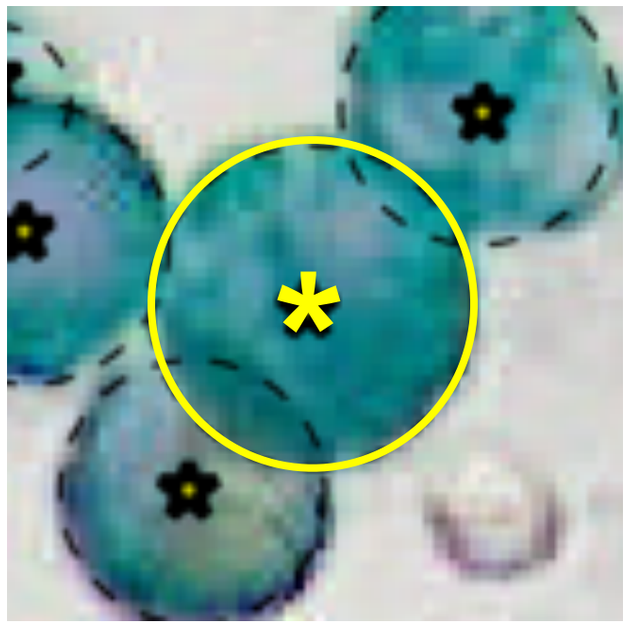


VGG-16

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

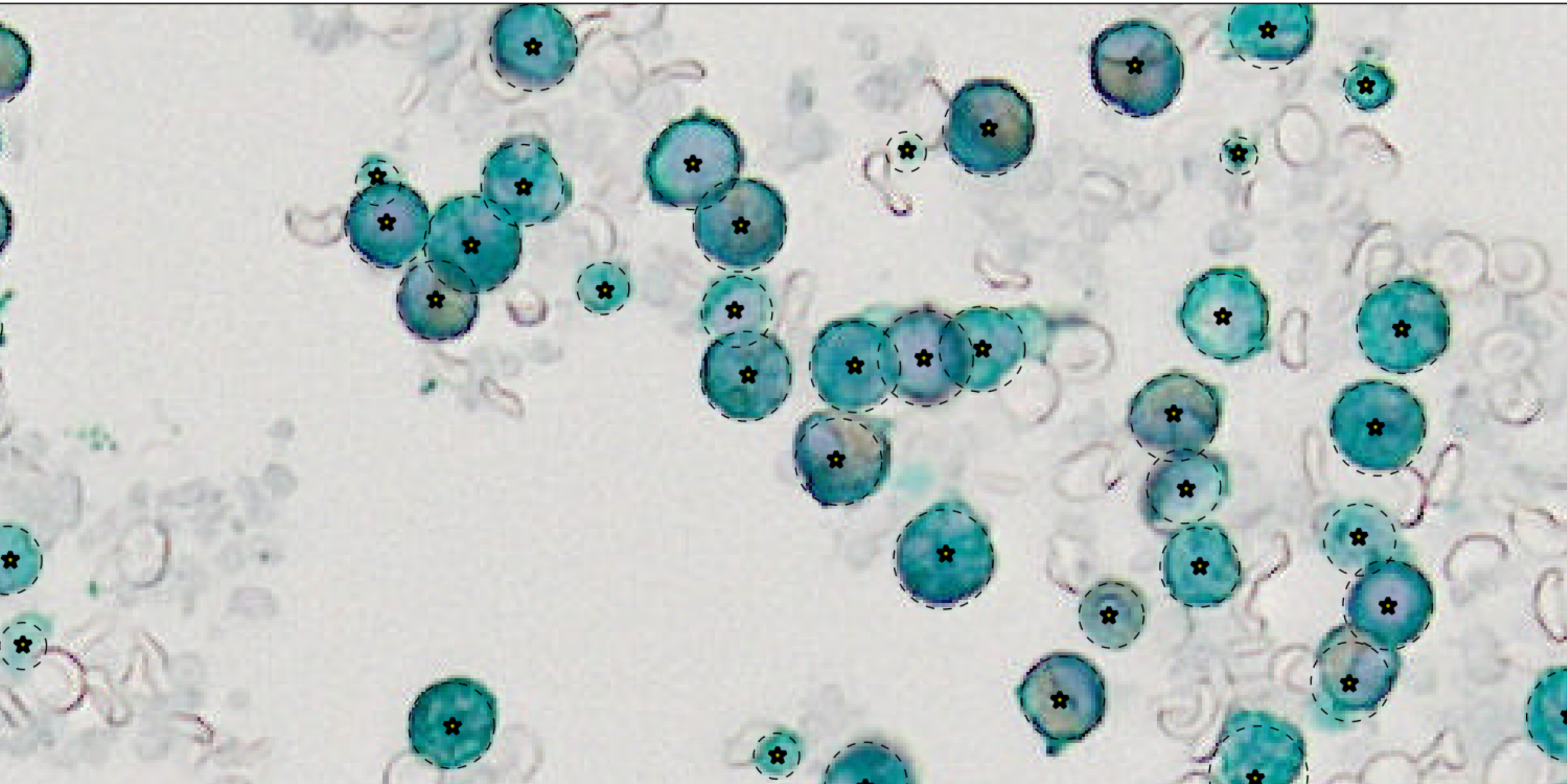
Centre and Radius

- No softmax
- Quadratic loss function

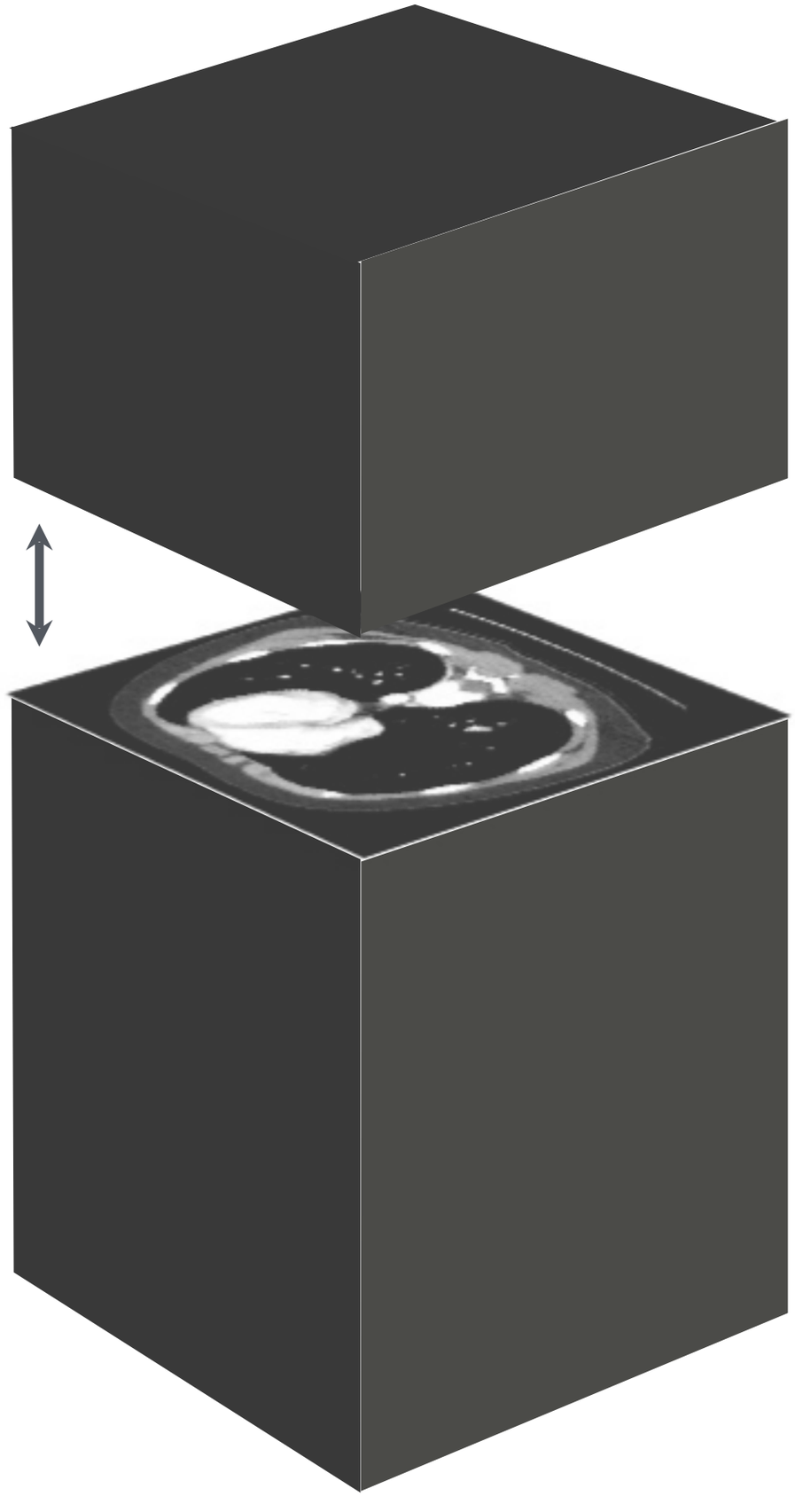


Centre and Radius

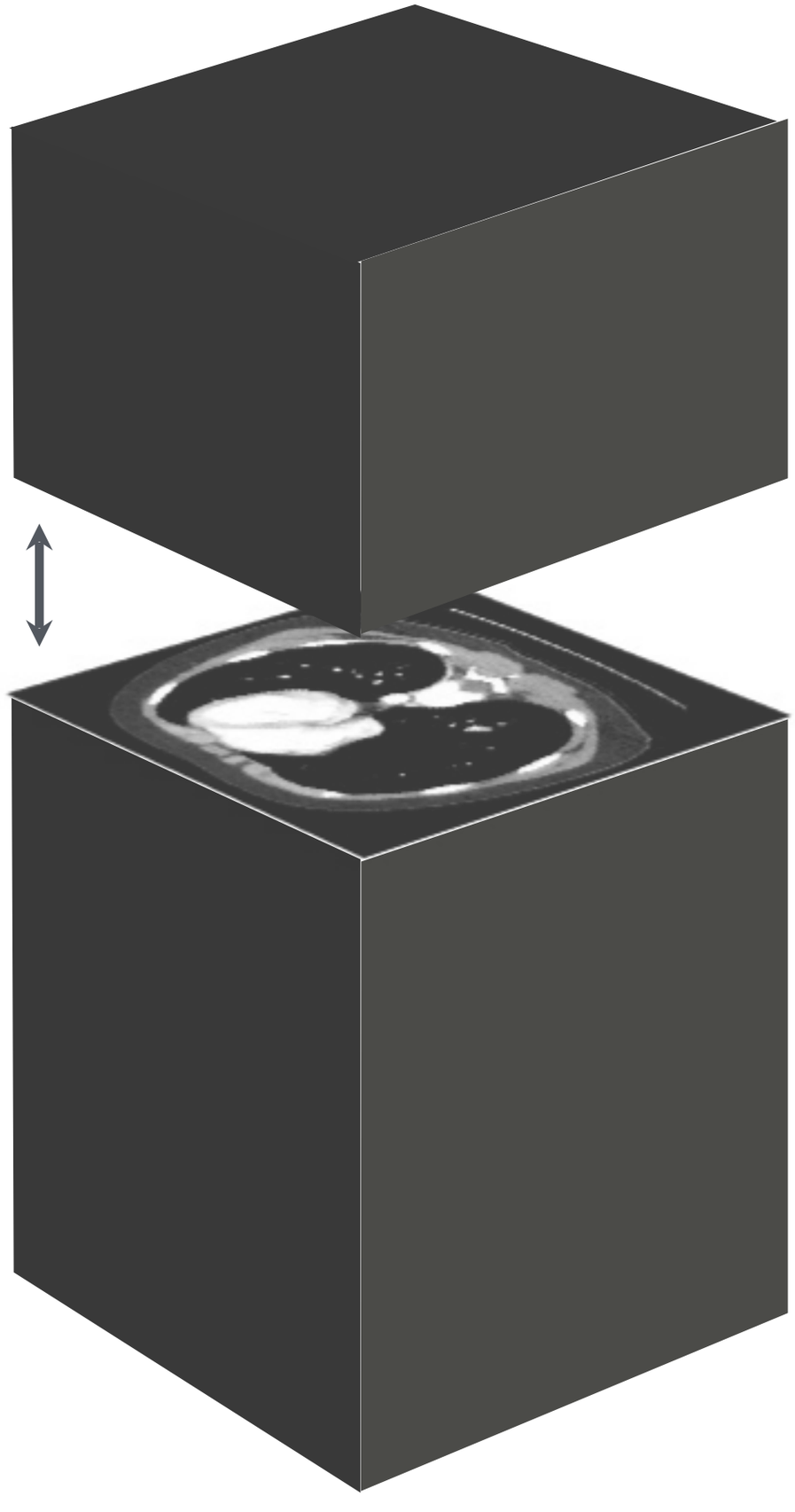
- No softmax
- Quadratic loss function



Segmentation

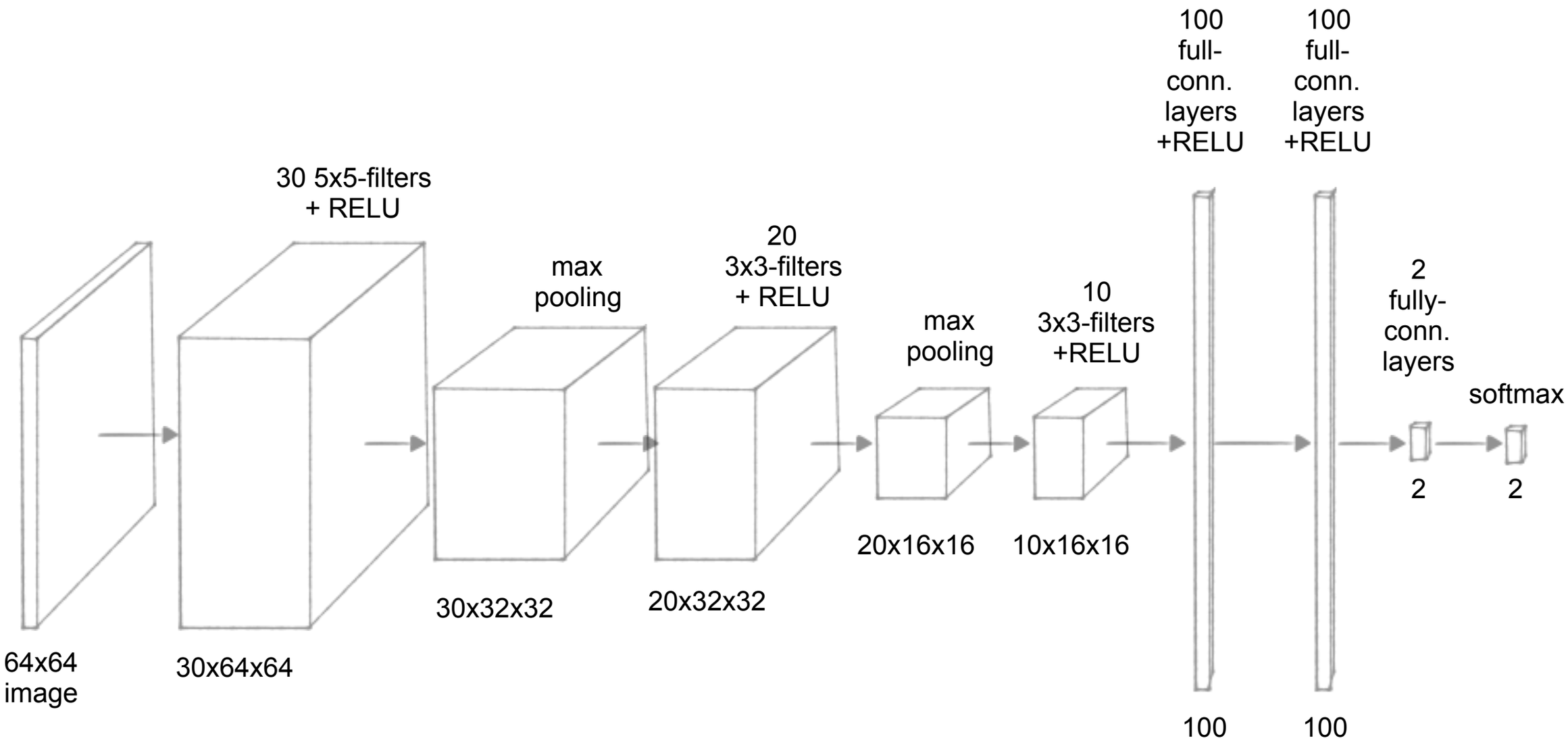


Segmentation



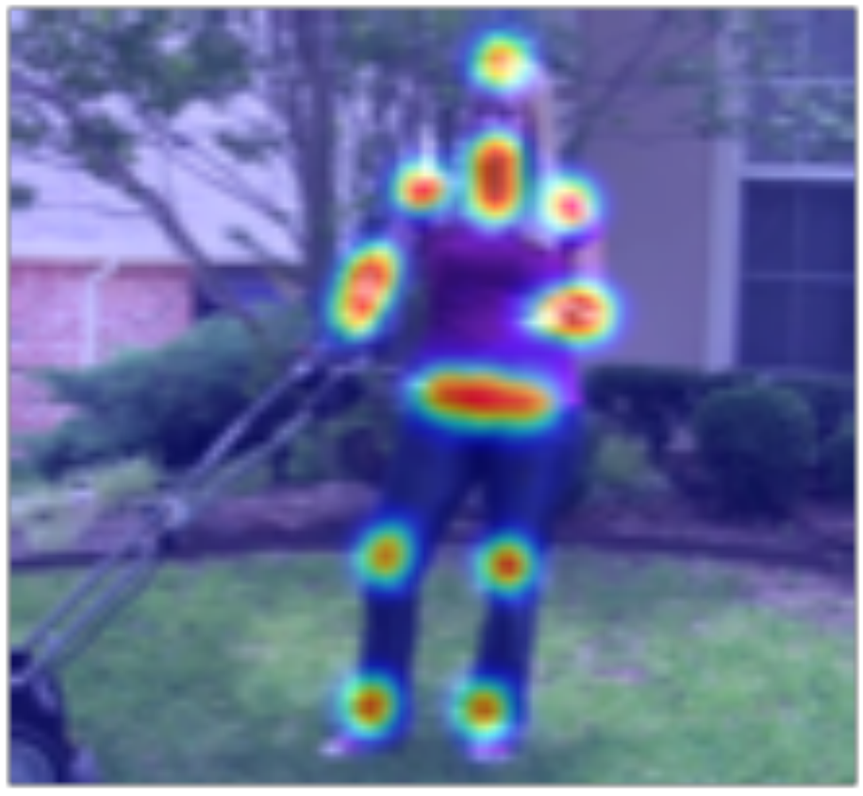
Fully Convolutional Neural Networks

Convolutional Neural Network for Classification

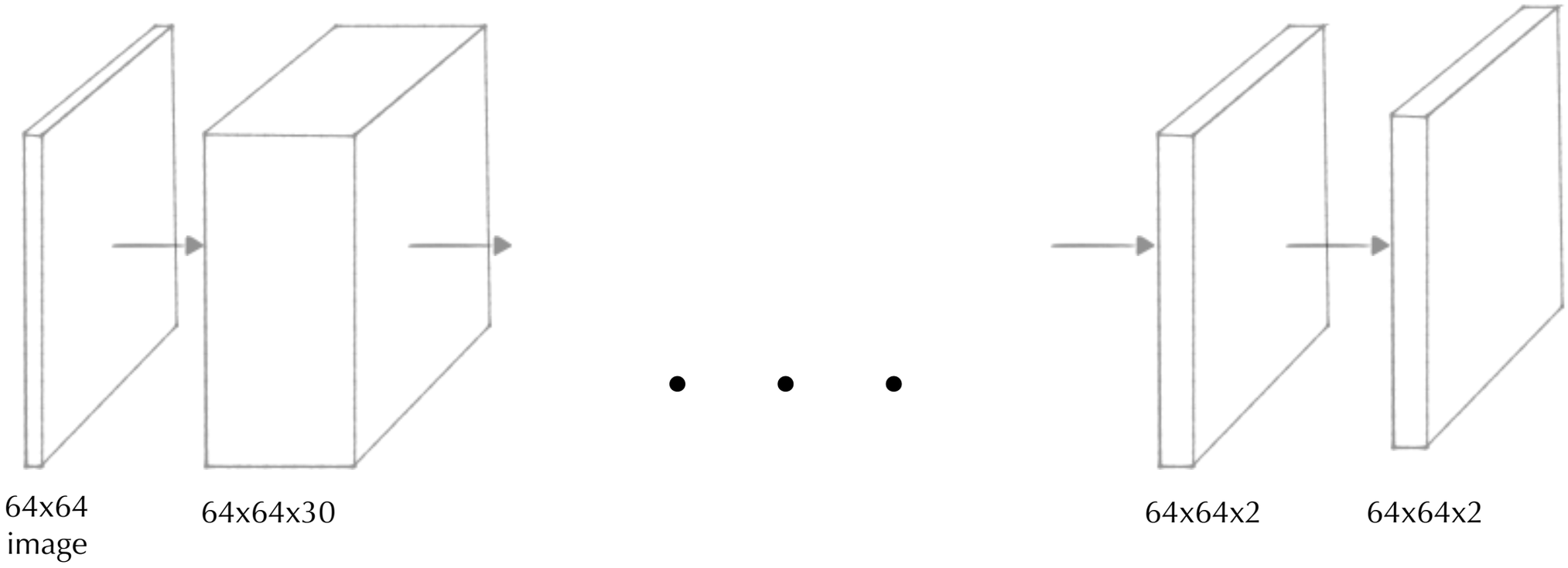


How to get a per-pixel decision?

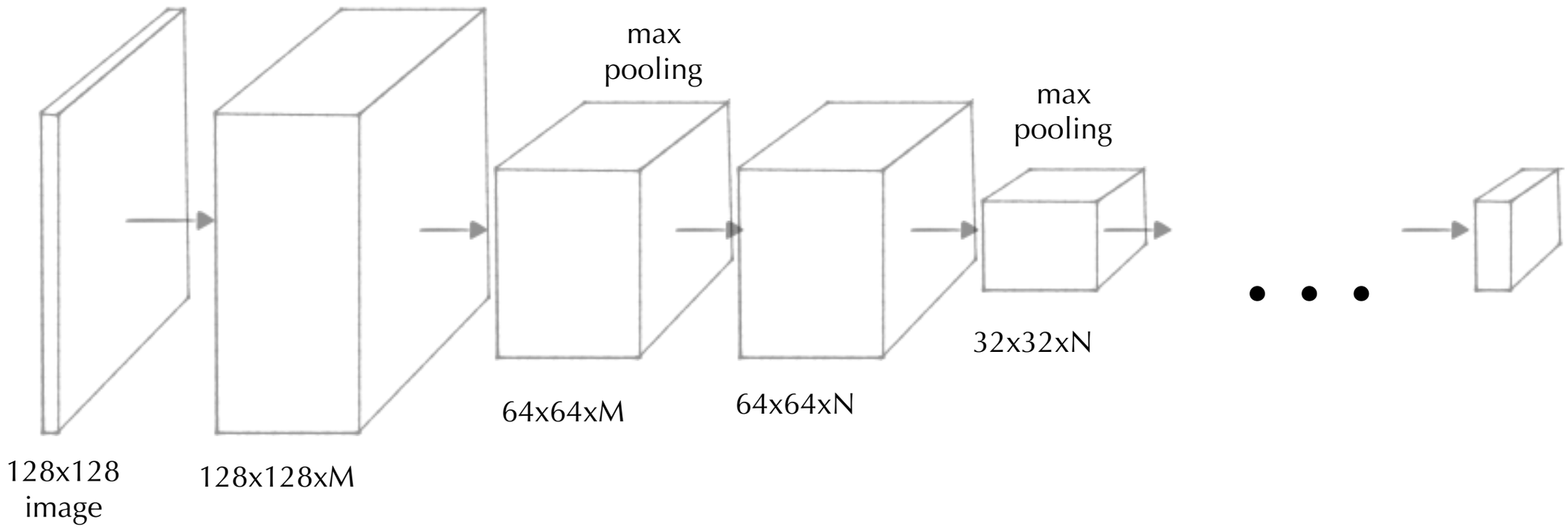
Human Pose Estimation



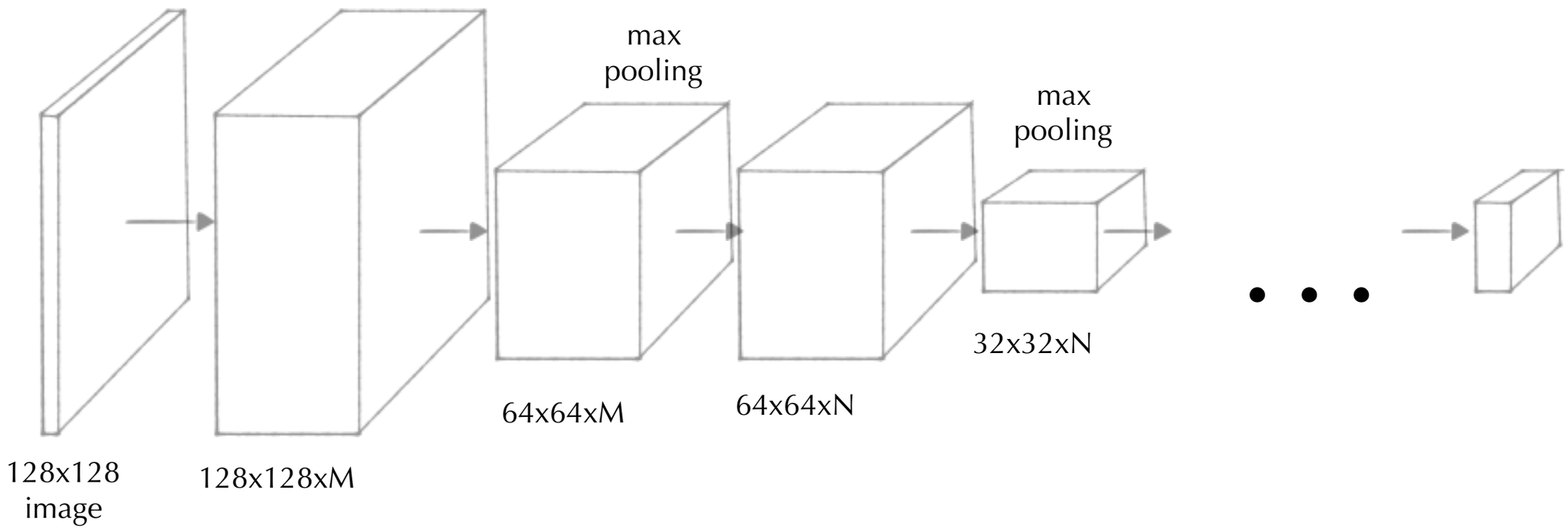
Fully-Convolutional Networks



Fully-Convolutional Network for Detection



Fully-Convolutional Network for Detection



Fully-Convolutional Network for Detection

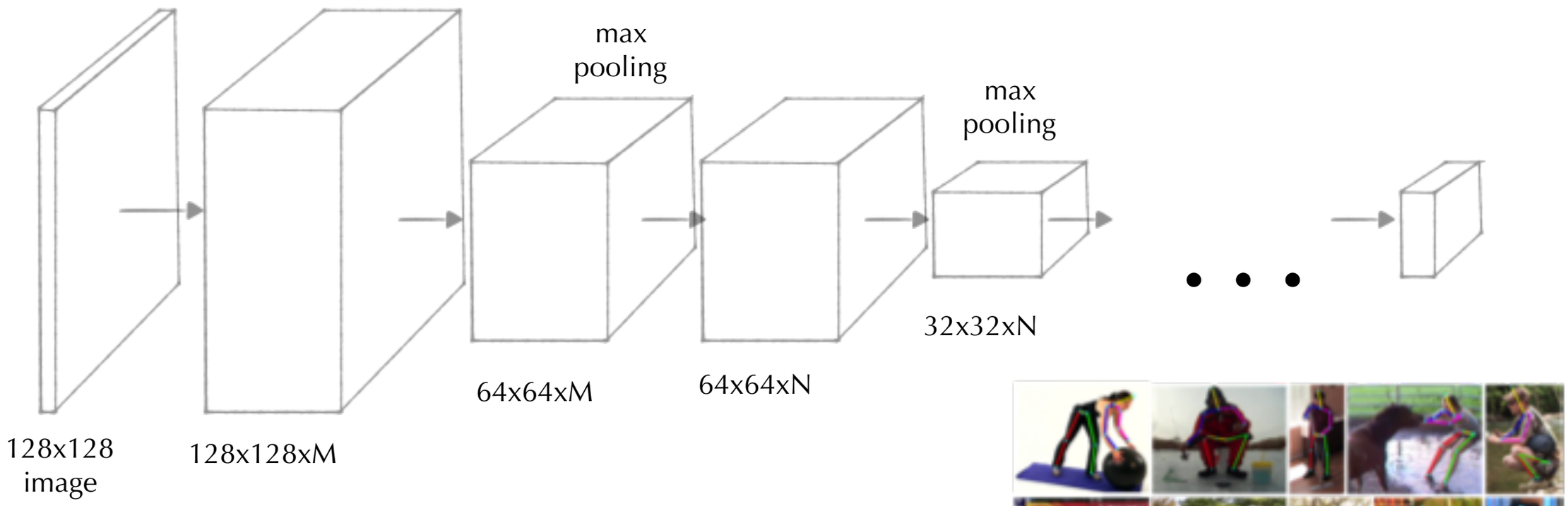
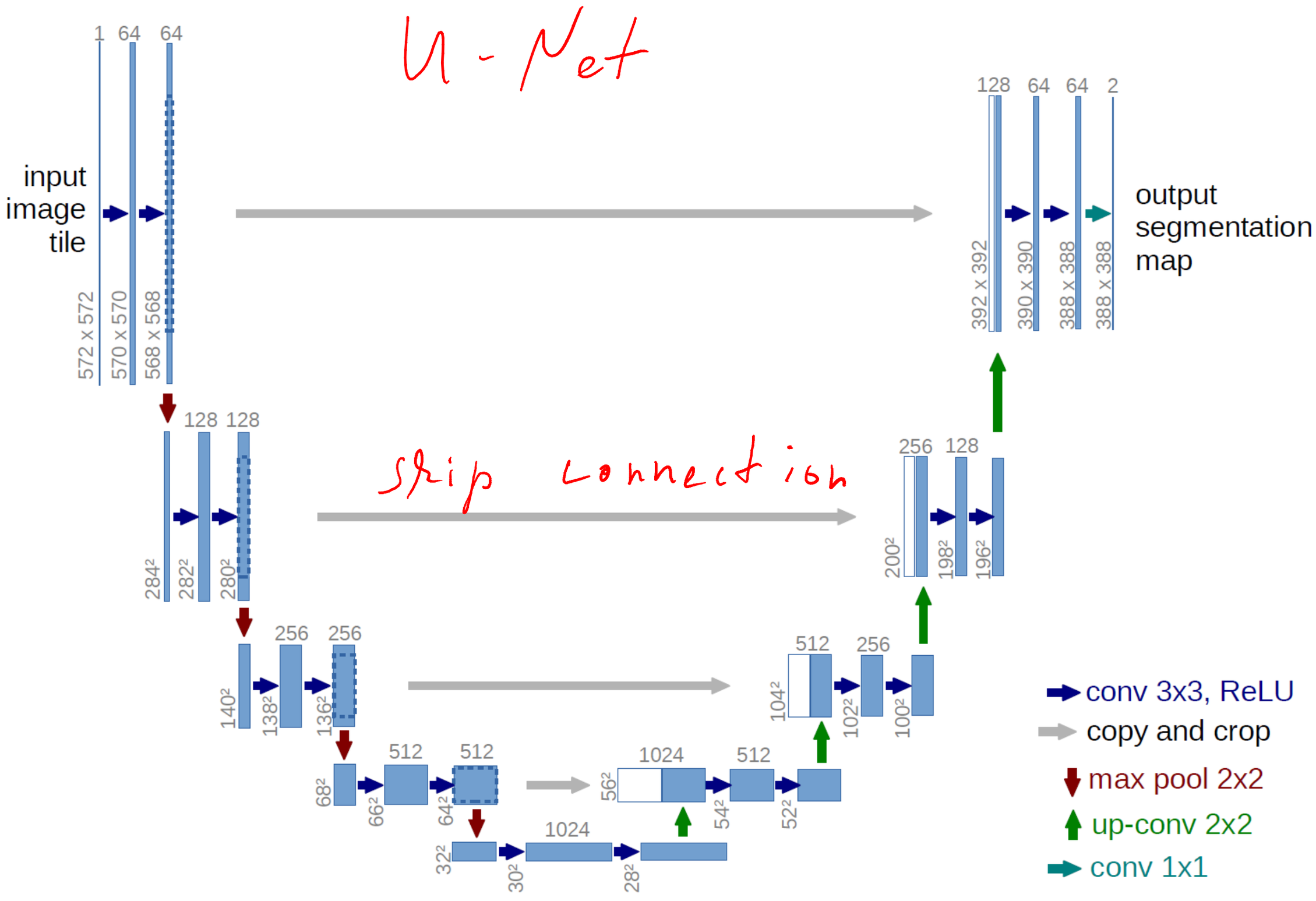


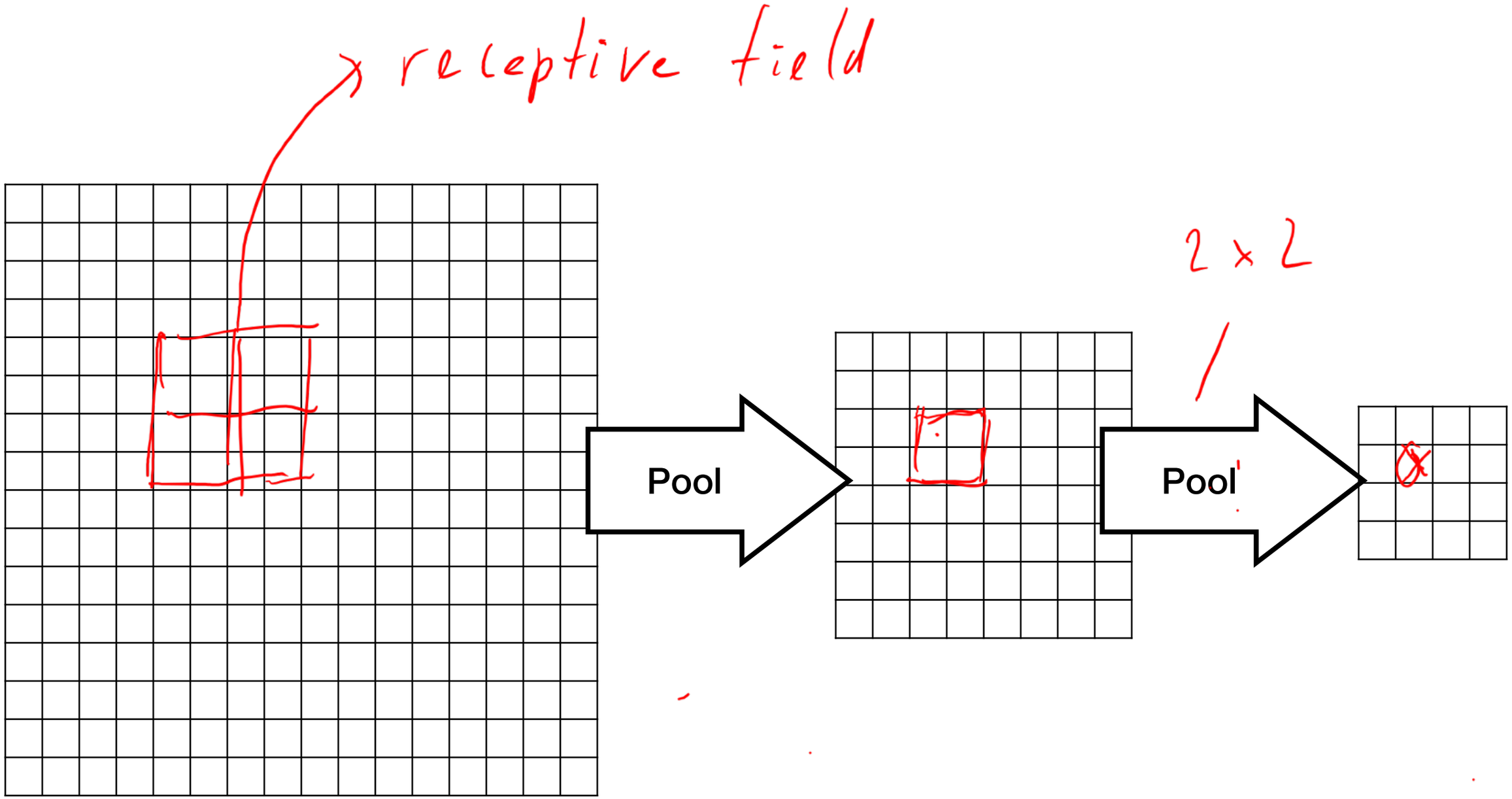
Image Segmentation

Decrease and Increase Resolution

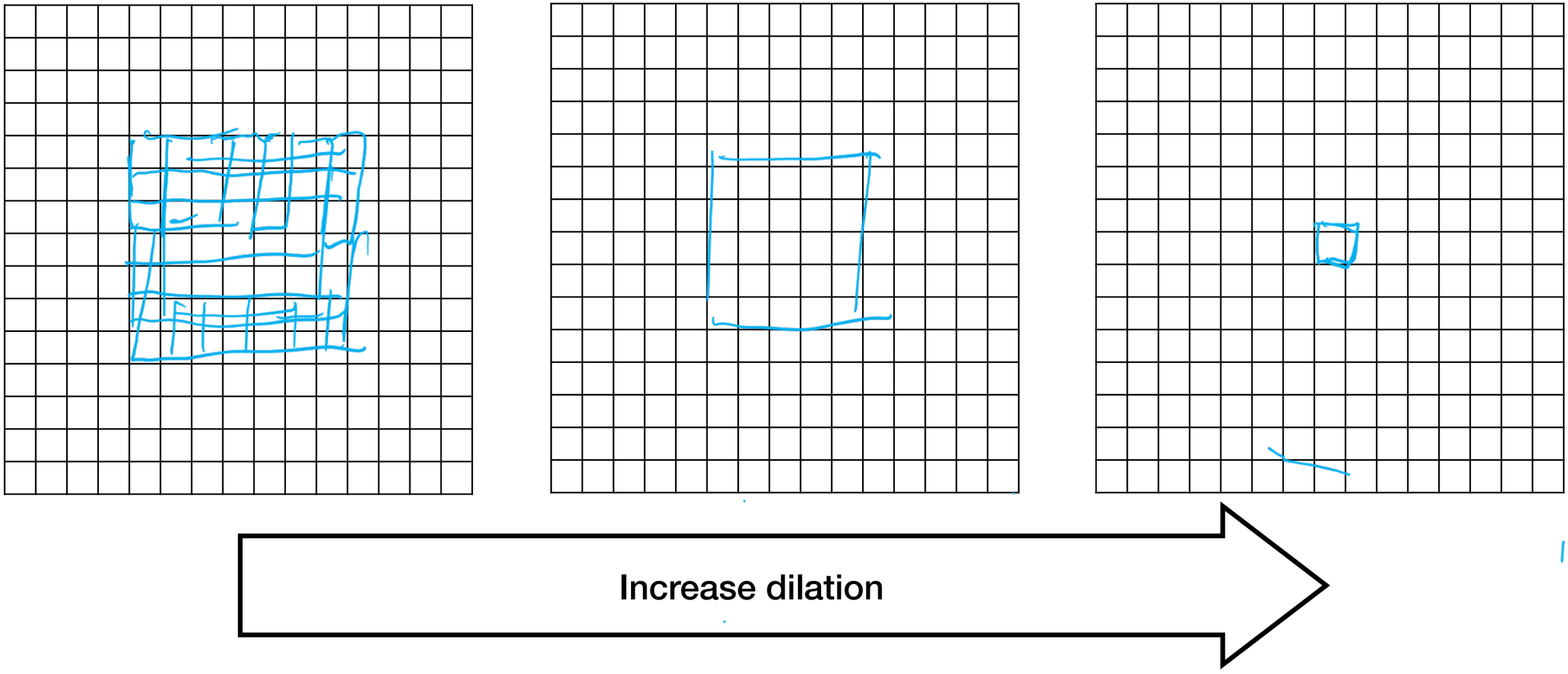


[Ronneberger et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015]

Effect of (Max-)Pooling

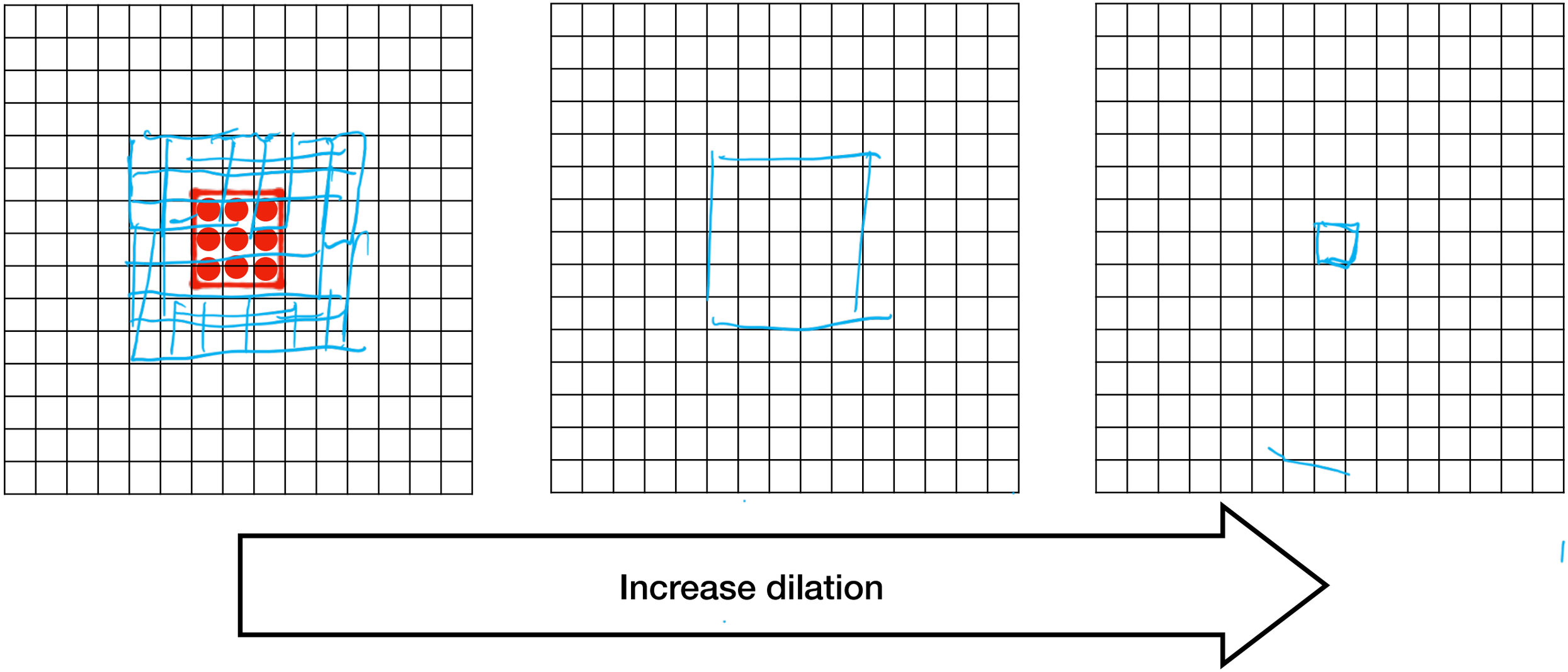


Dilated Convolutions



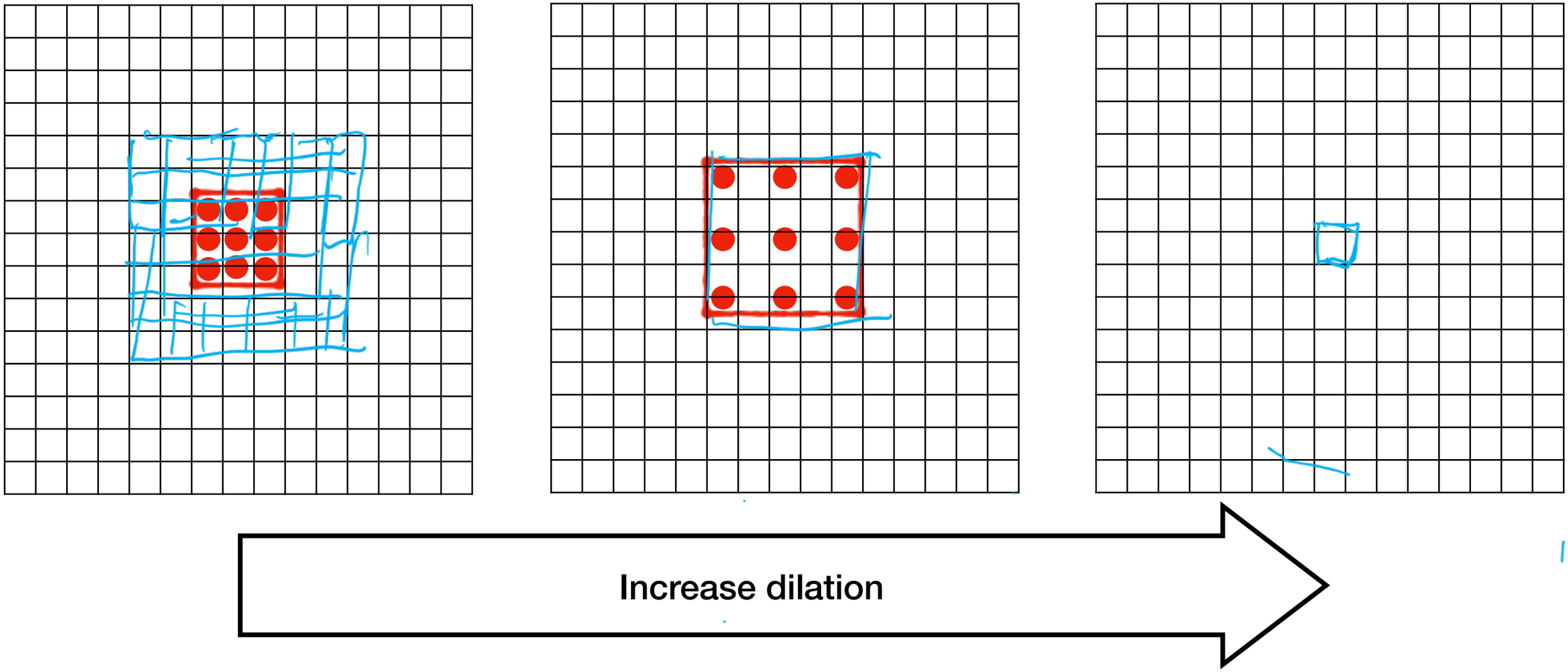
[Yu & Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, ICLR 2016]

Dilated Convolutions



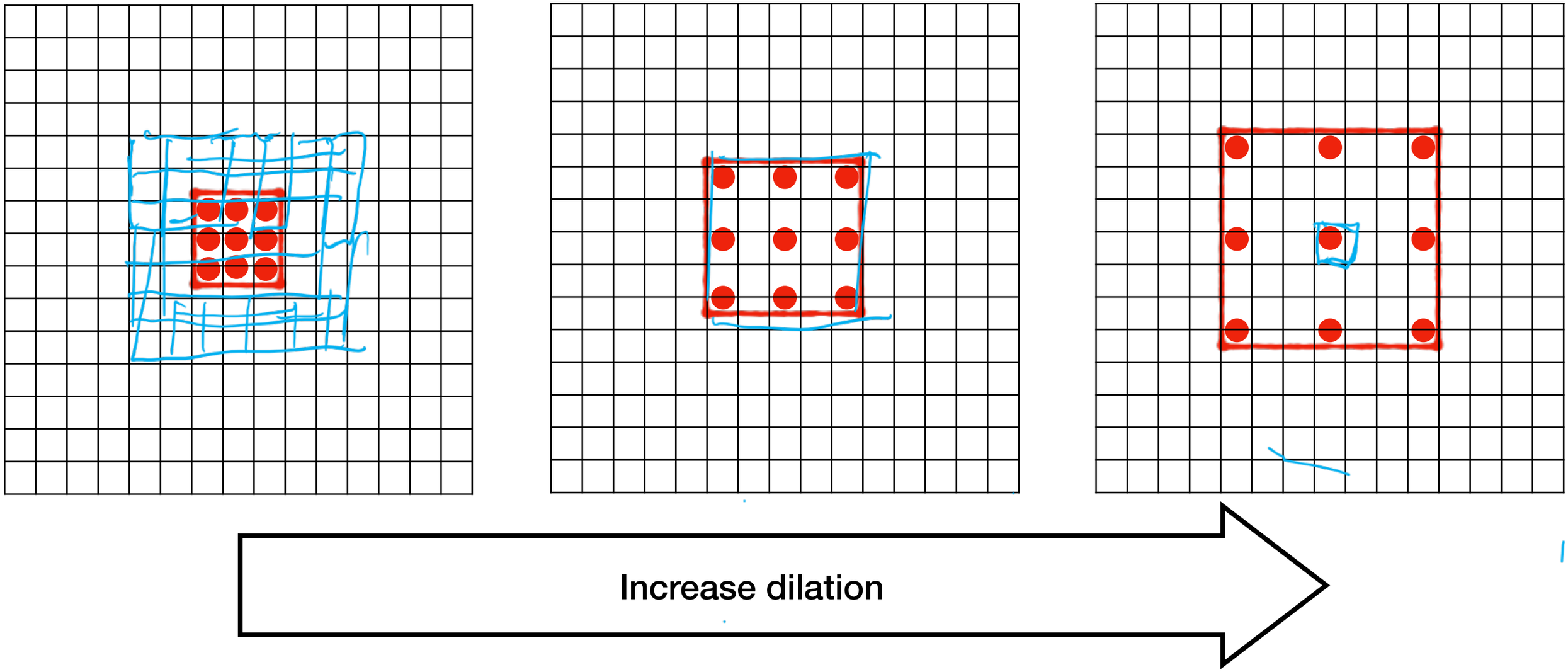
[Yu & Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, ICLR 2016]

Dilated Convolutions



[Yu & Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, ICLR 2016]

Dilated Convolutions



[Yu & Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, ICLR 2016]

Convolutional Layers

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0		

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

stride

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	0

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	0
2		

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	0
2	8	

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	0
2	8	1

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	0
2	8	1
0		

$I \star w$

Strided Filtering

$$w = \frac{1}{5}$$

0	1	0
1	1	1
0	1	0

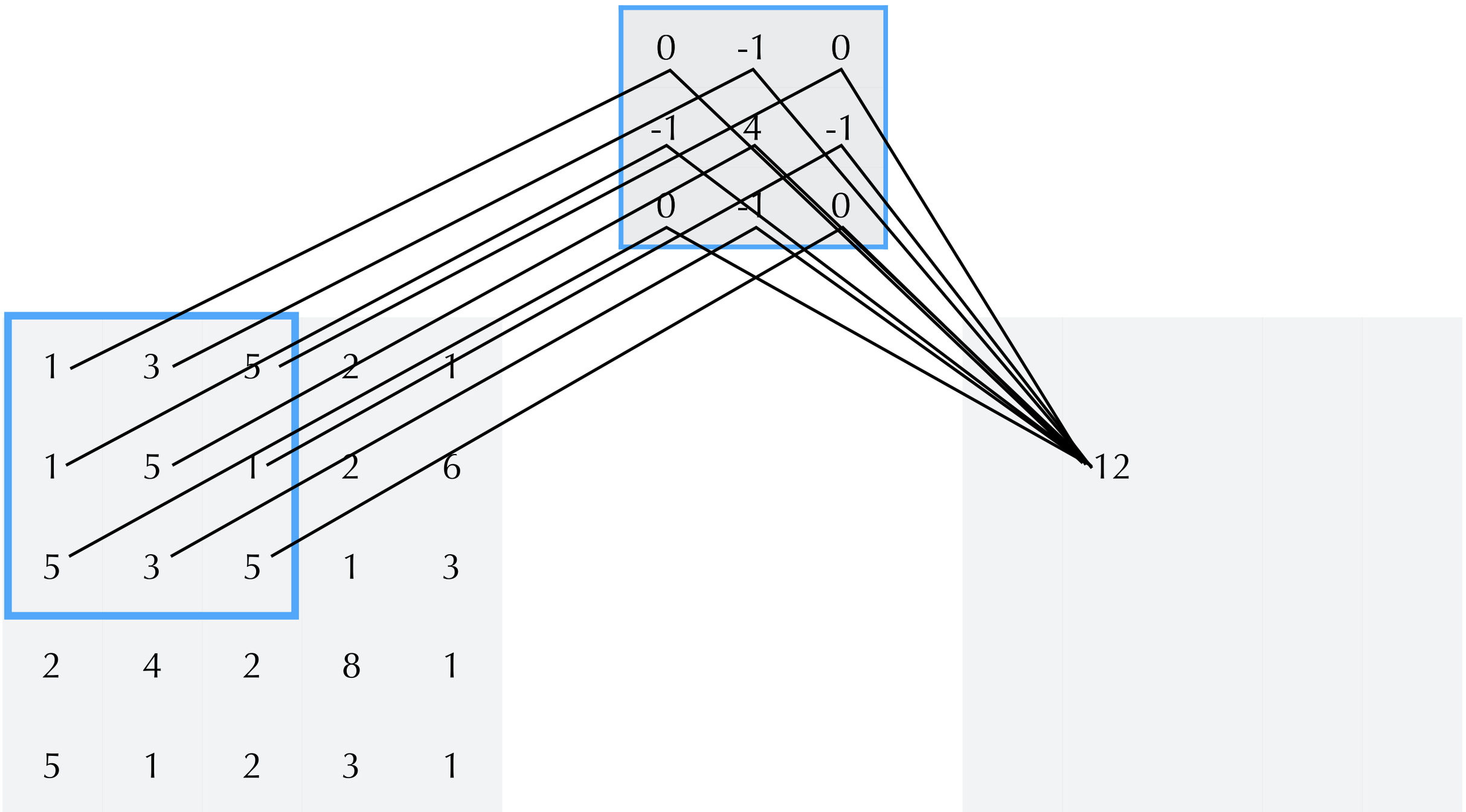
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	0
2	8	1
0	1	

$I \star w$

Different View



Different View

0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Different View

0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Different View

0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Different View

0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Different View

0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

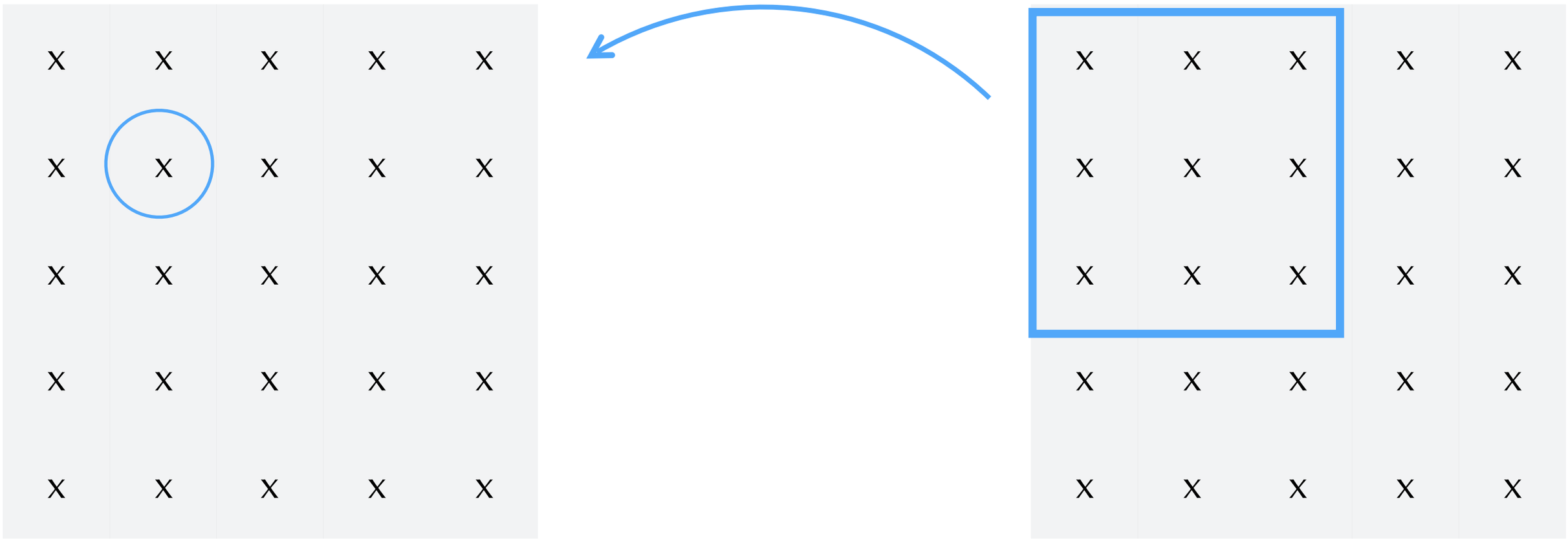
Different View

0	-1	0
-1	4	-1
0	-1	0

X	X	X	X	X
X	X	X	X	X
X	X	X	X	X
X	X	X	X	X
X	X	X	X	X
X	X	X	X	X

Different View

0	-1	0
-1	4	-1
0	-1	0



Transposed convolution

What's the point?

x	x	x
x	x	x
x	x	x

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Strided filtering

What's the point?

Transposed strided filtering

What's the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3			

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

What's the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

What's the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

What's the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

What's the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4
5				

	4	0	2	0	4
	0	0	0	0	0
	6	0	4	0	6
	0	0	0	0	0
	2	0	12	0	2

What's the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4
5	4			

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

What's the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4
5	4	3	4	5
6	5	4	5	6
4	6	8	6	4
2	7	12	7	2

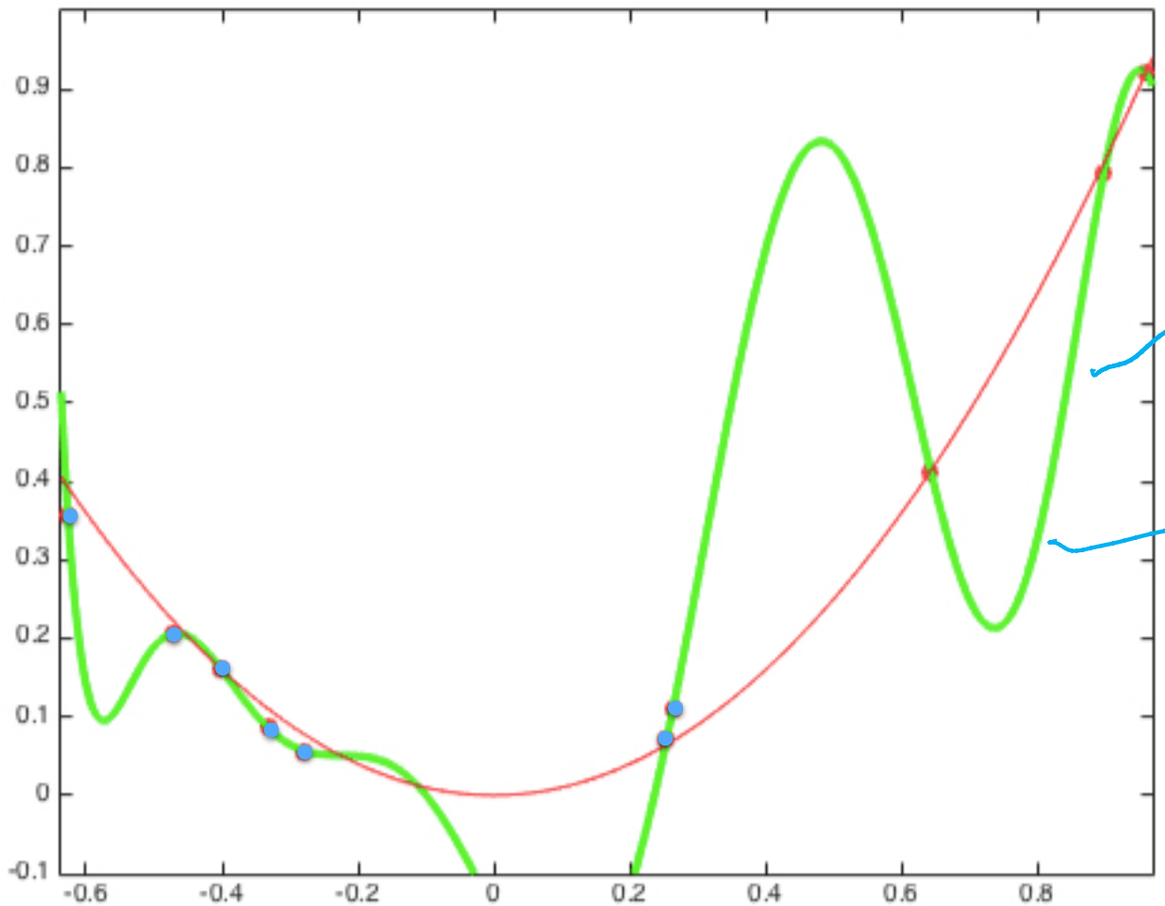
4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed convolution

Overfitting, Part II

Overfitting

10 data points - Fitting 10th degree polynomial



$$y = x^2$$

Overfitting

VGG-16 architecture

224

224

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

224

224

Conv 64 3x3

Conv 64 3x3

max pooling

conv 1

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

224

224

Conv 64 3x3

Conv 64 3x3

max pooling

Conv 128 3x3

Conv 128 3x3

max pooling

conv 1

conv 2

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

224

224

Conv 64 3x3 | Conv 64 3x3 | max pooling — **conv 1**

Conv 128 3x3 | Conv 128 3x3 | max pooling — **conv 2**

Conv 256 3x3 | Conv 256 3x3 | Conv 256 3x3 | max pooling — **conv 3**

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

224

224

Conv 64 3x3 | Conv 64 3x3 | max pooling — **conv 1**

Conv 128 3x3 | Conv 128 3x3 | max pooling — **conv 2**

Conv 256 3x3 | Conv 256 3x3 | Conv 256 3x3 | max pooling — **conv 3**

Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling — **conv 4**

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

224

224

Conv 64 3x3 | Conv 64 3x3 | max pooling — **conv 1**

Conv 128 3x3 | Conv 128 3x3 | max pooling — **conv 2**

Conv 256 3x3 | Conv 256 3x3 | Conv 256 3x3 | max pooling — **conv 3**

Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling — **conv 4**

Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling — **conv 5**

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

224

224

Conv 64 3x3 | Conv 64 3x3 | max pooling — **conv 1**

Conv 128 3x3 | Conv 128 3x3 | max pooling — **conv 2**

Conv 256 3x3 | Conv 256 3x3 | Conv 256 3x3 | max pooling — **conv 3**

Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling — **conv 4**

Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling — **conv 5**

FC 4096 | FC 4096 | FC 1000 | soft max

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

224

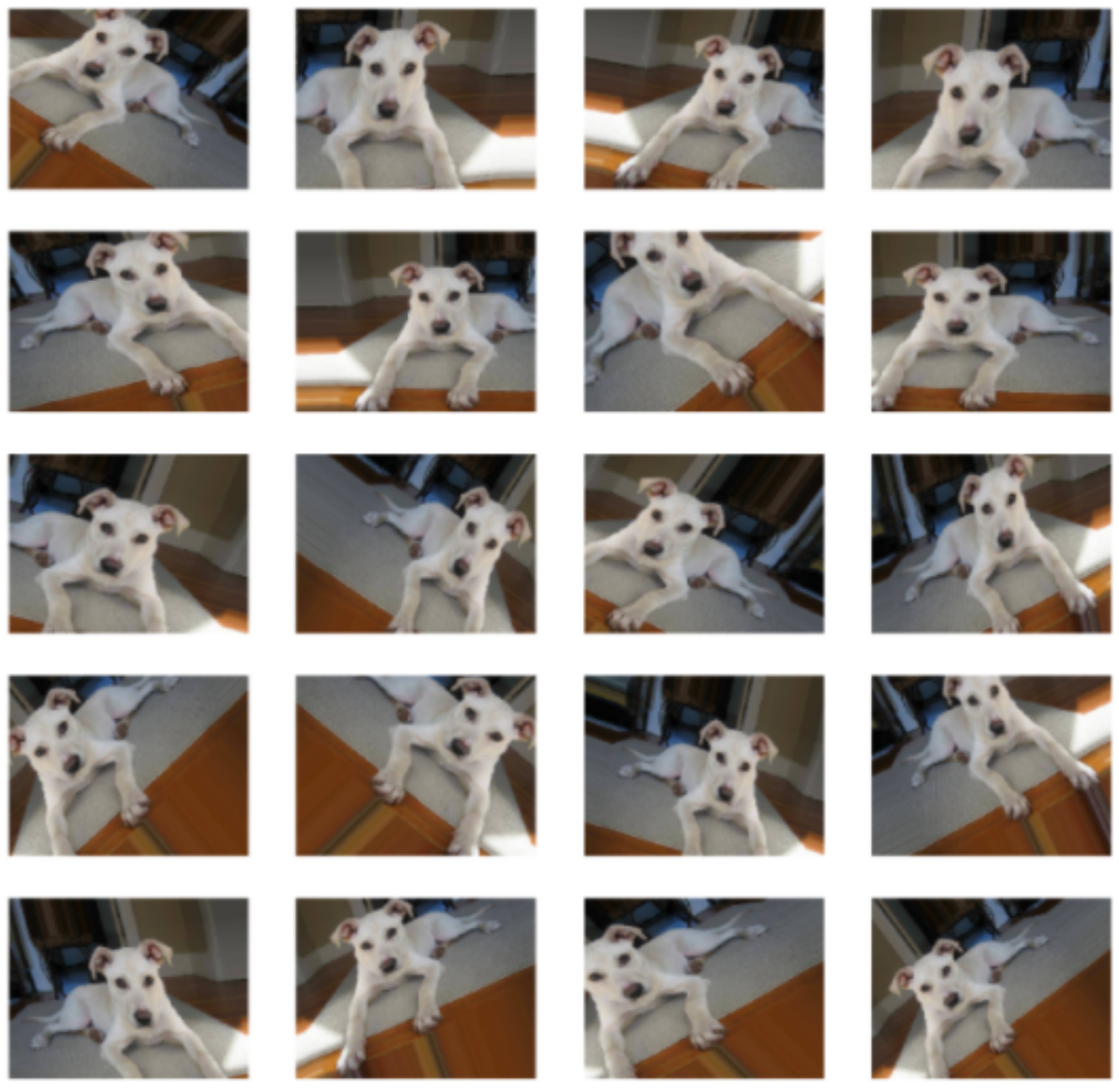
224

138M parameters!

Conv 64 3x3	Conv 64 3x3	max pooling		**conv 1**
Conv 128 3x3	Conv 128 3x3	max pooling		**conv 2**
Conv 256 3x3	Conv 256 3x3	Conv 256 3x3	max pooling	**conv 3**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 4**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 5**
FC 4096	FC 4096	FC 1000	soft max	

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Data Augmentation



Regularization

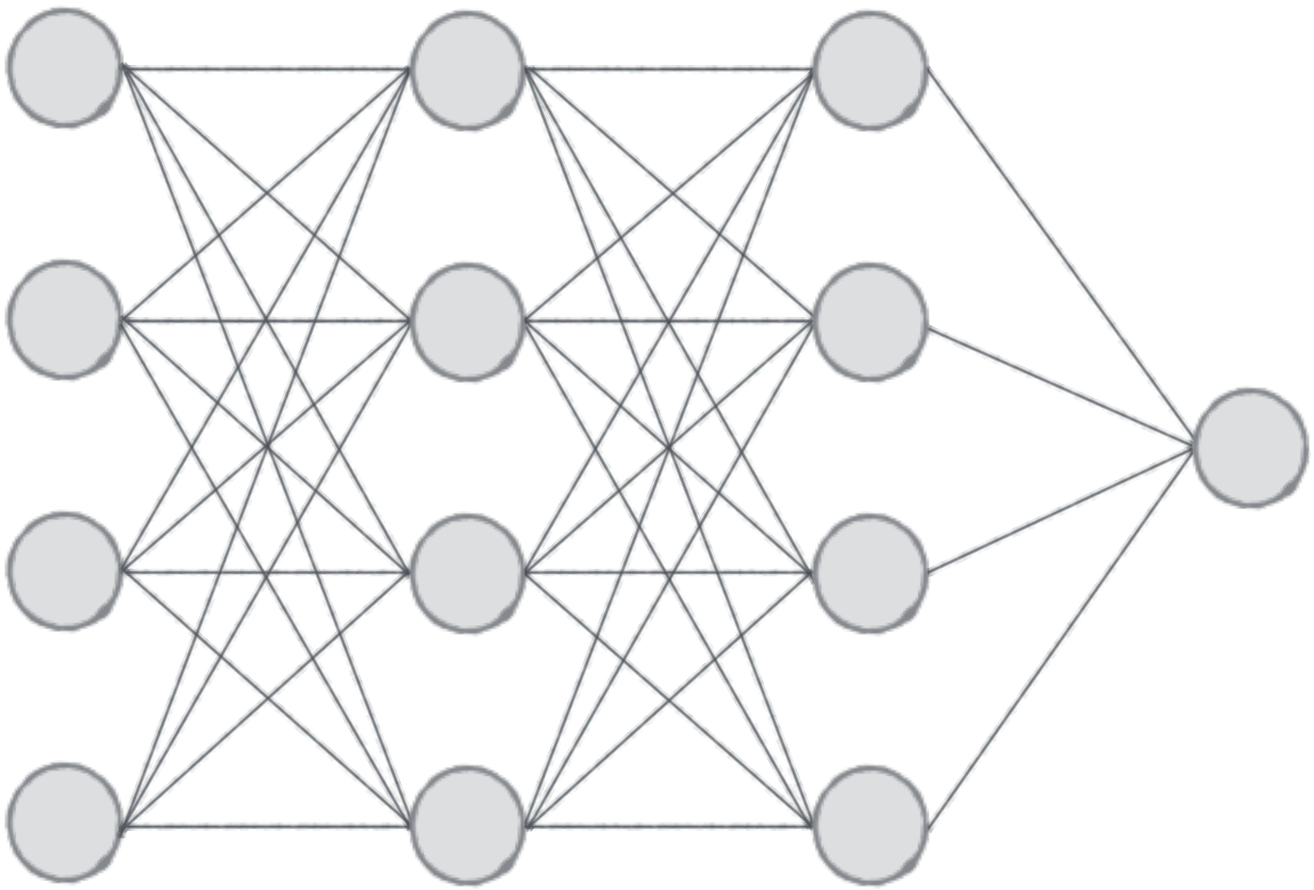
weighting term

$$\min_{w_1,\ldots,w_n} \left(L(w_1,\ldots,w_n) + c\sum_i w_i^2 \right)$$

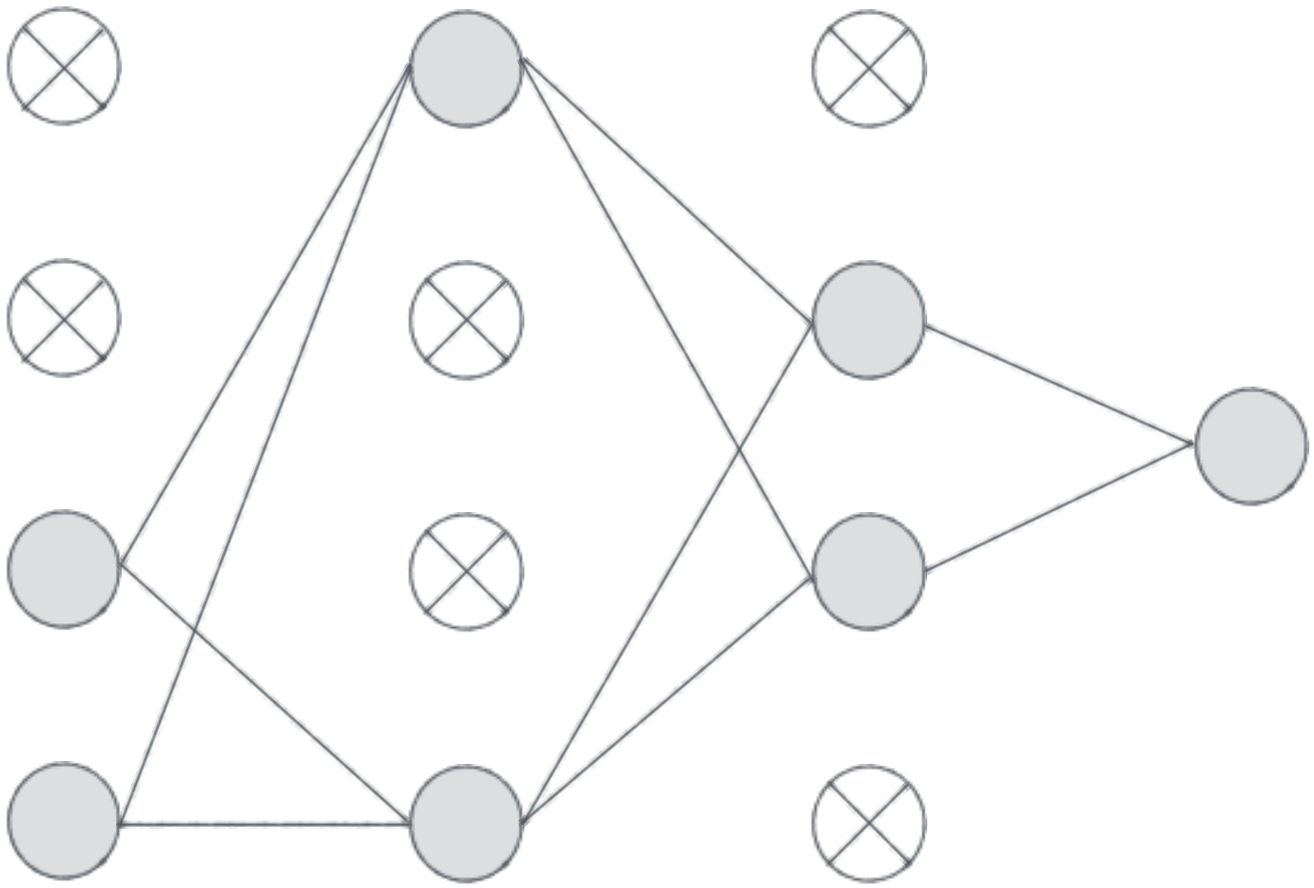
loss

regularization term

Dropout



Dropout



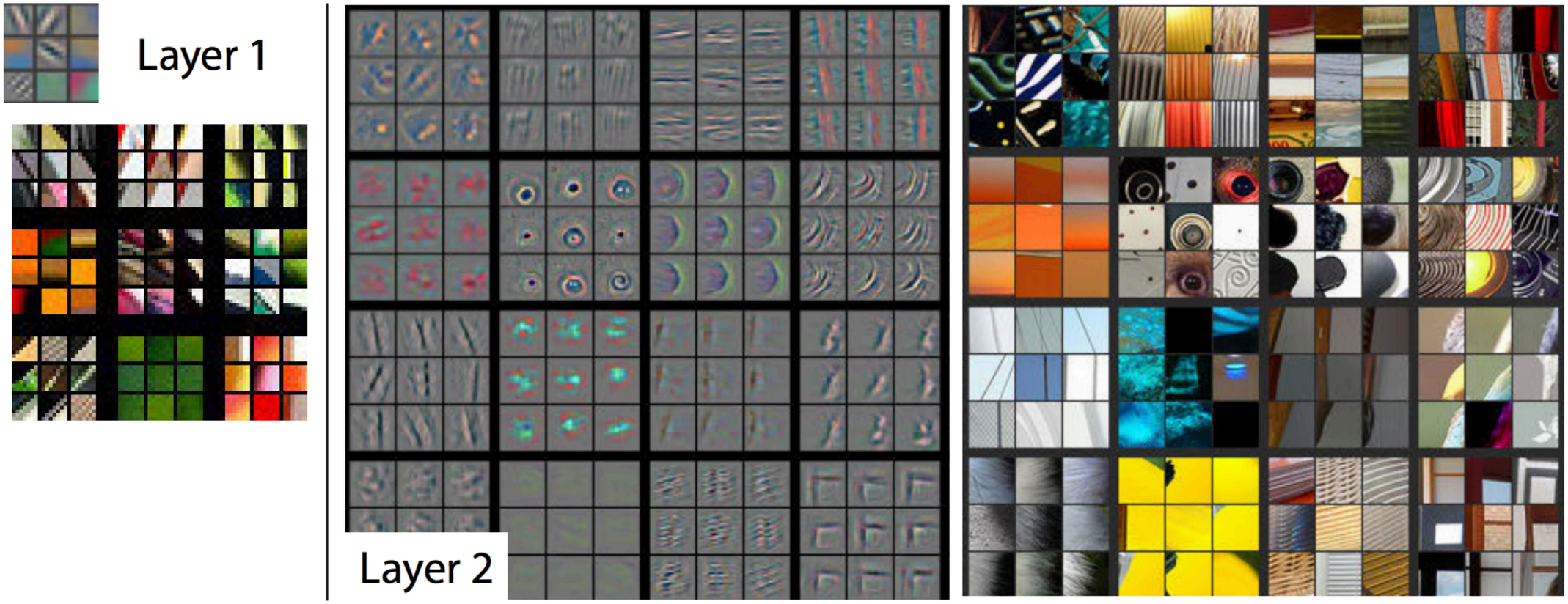
Randomly disable neurons for each minibatch

Transfer Learning

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*. [web]

14 million images of 1000 classes

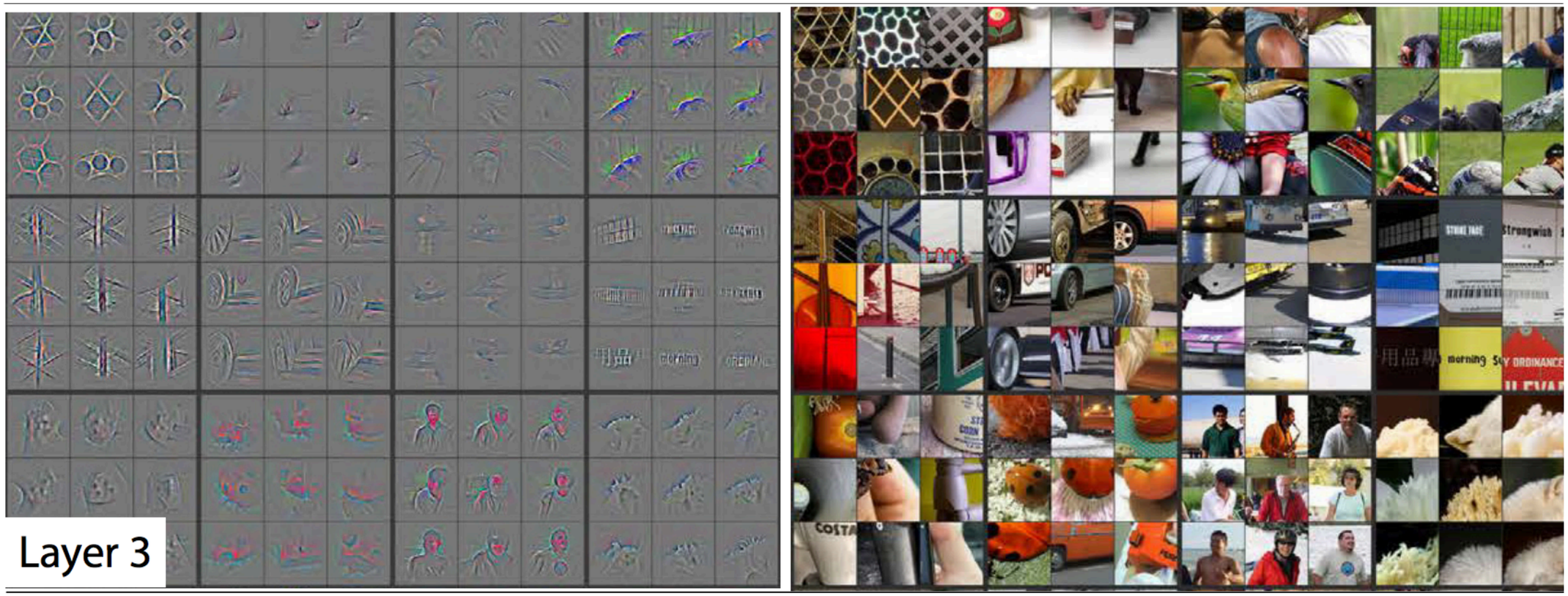
Transfer Learning



Low-level features: corners, edges, …

[Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014]

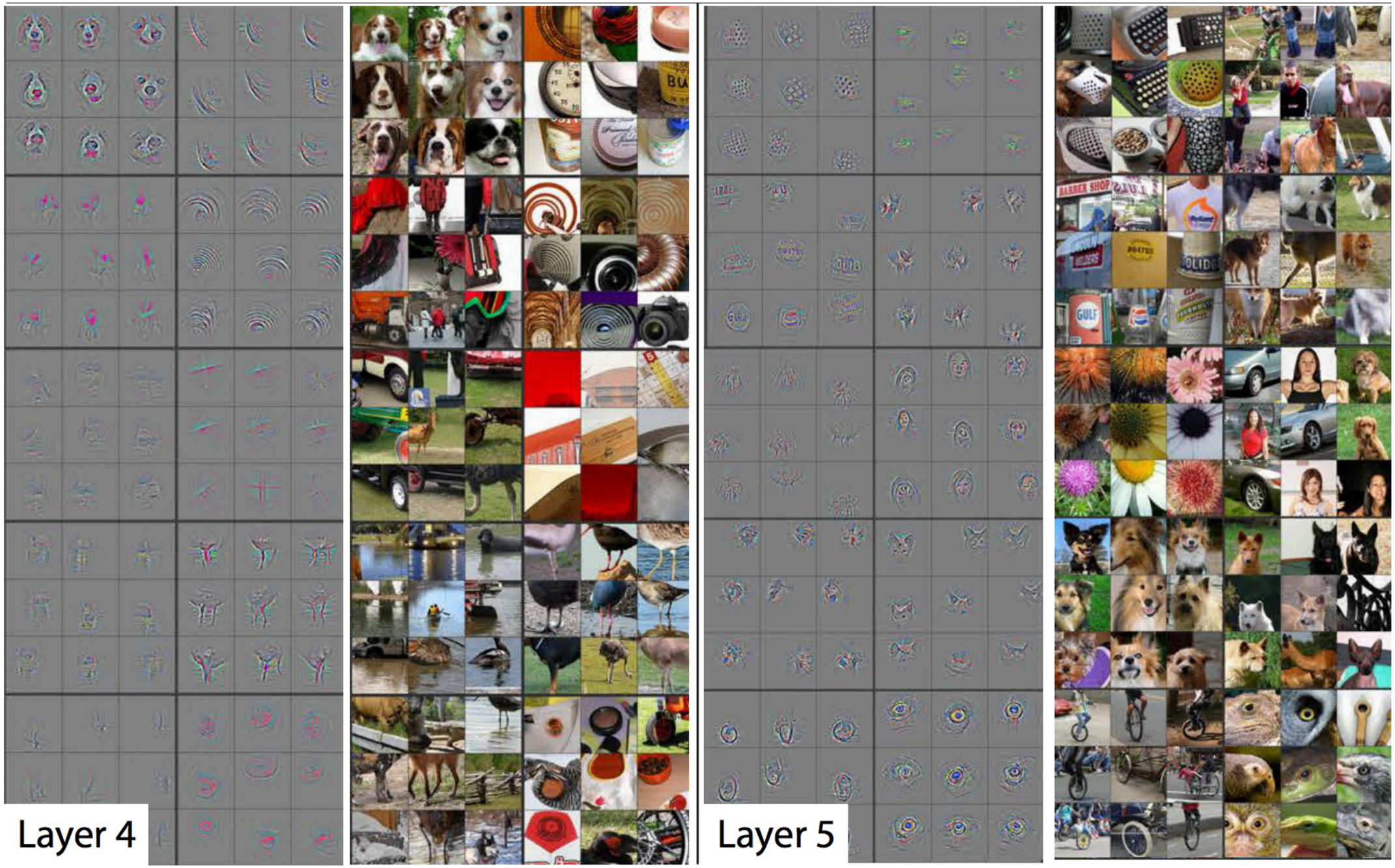
Transfer Learning



Mid-level features

[Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014]

Transfer Learning

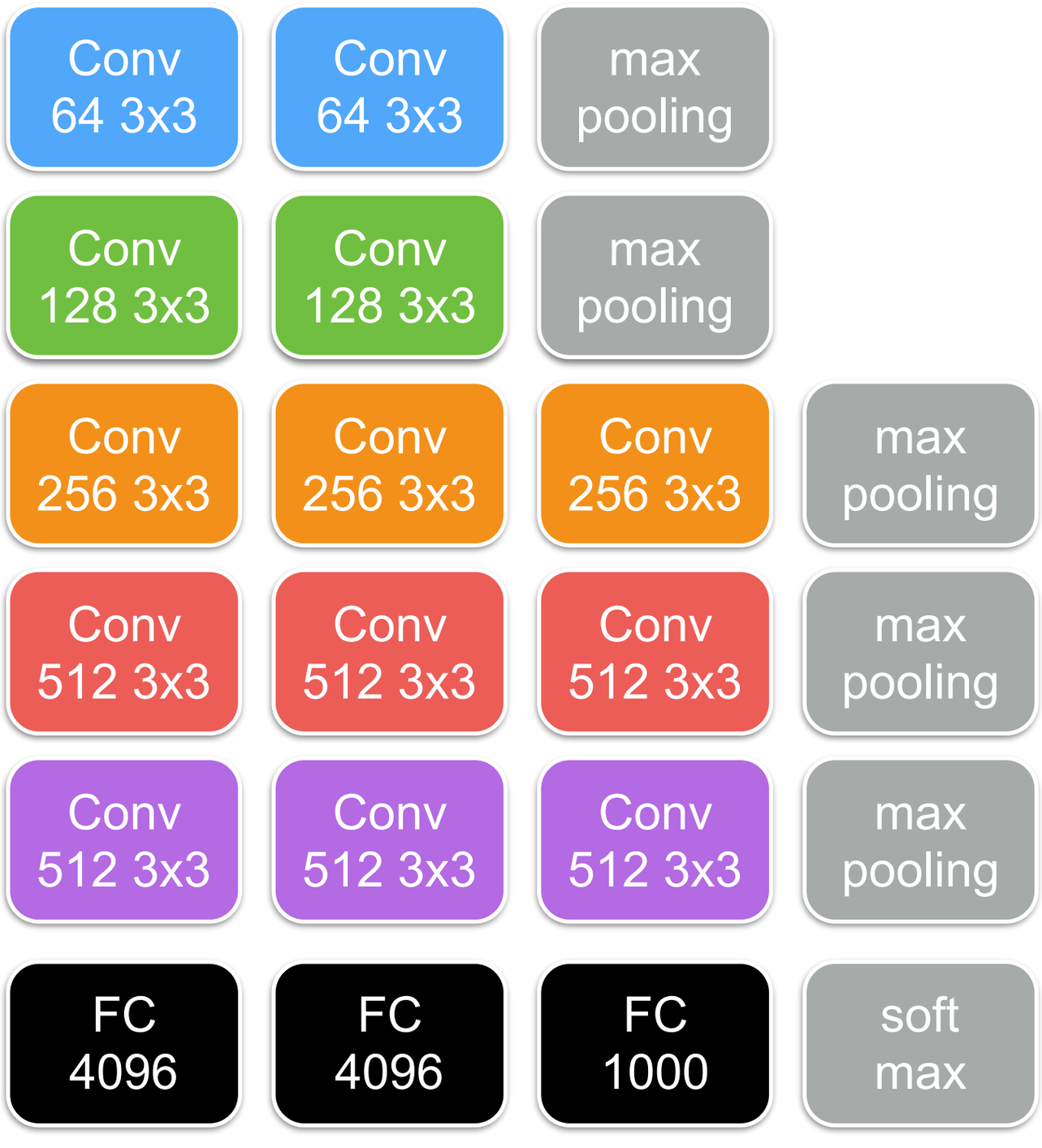


Higher-level features: object parts & objects

[Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014]

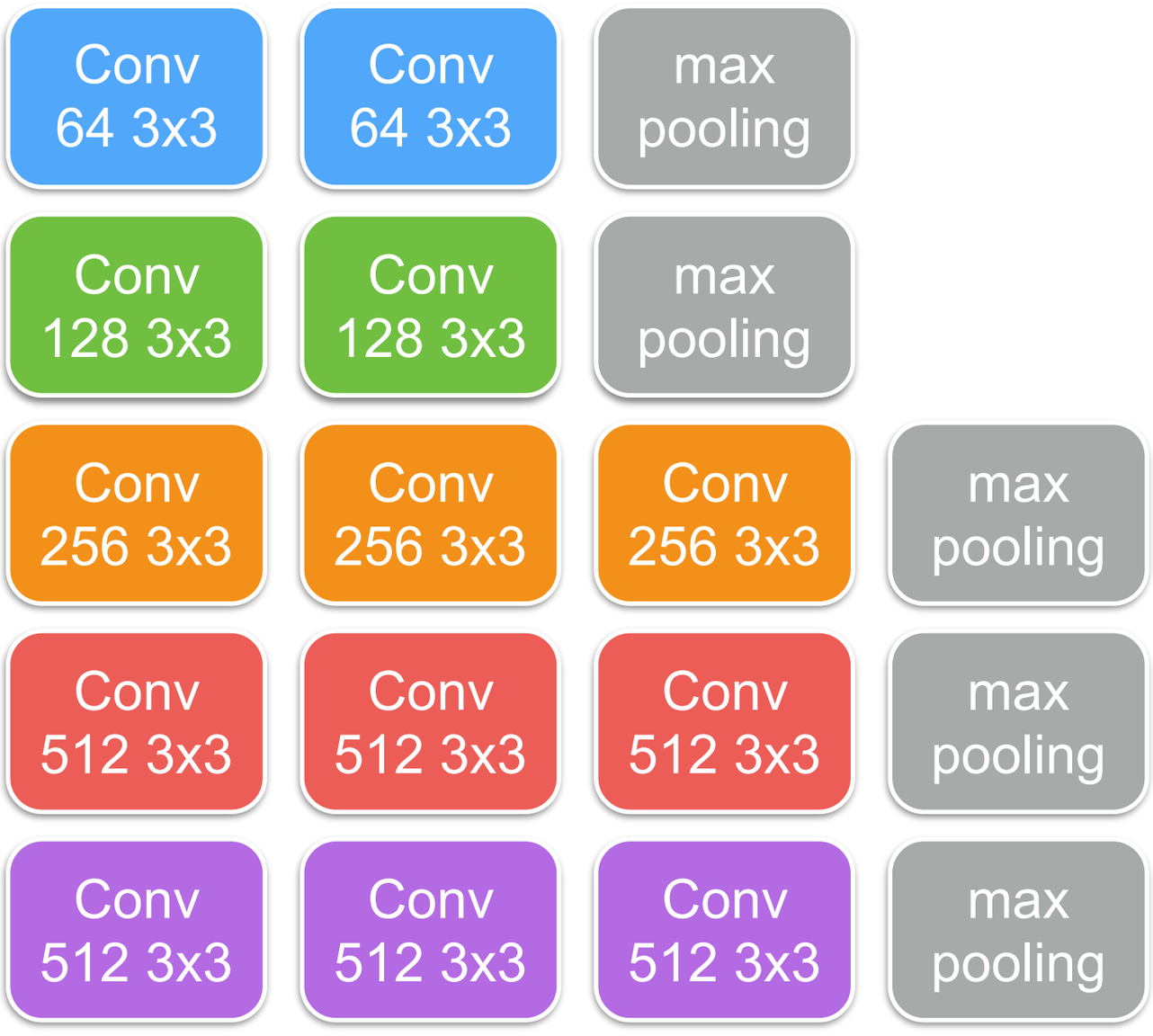
Transfer Learning

VGG-16 architecture



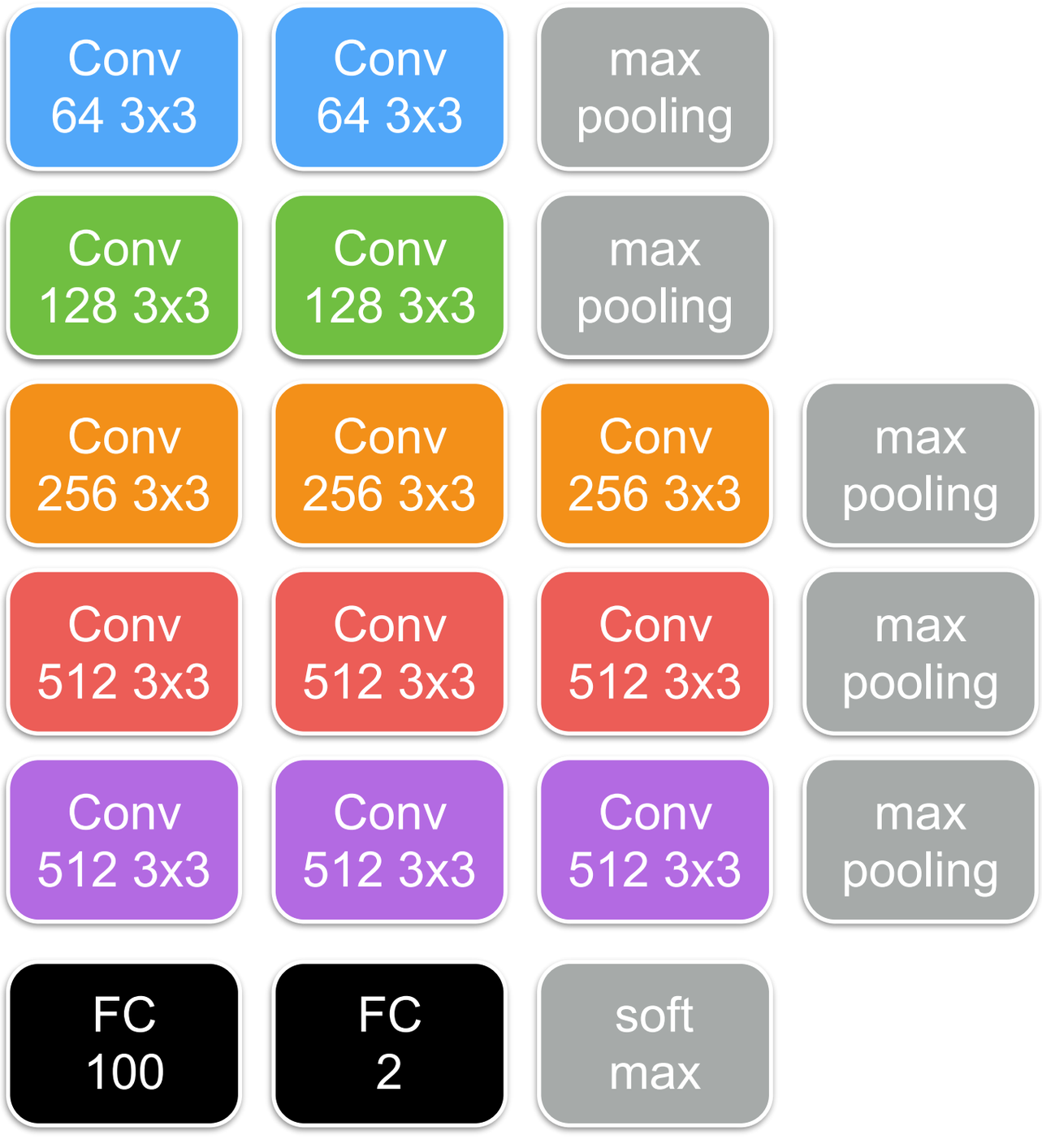
Transfer Learning

VGG-16 architecture



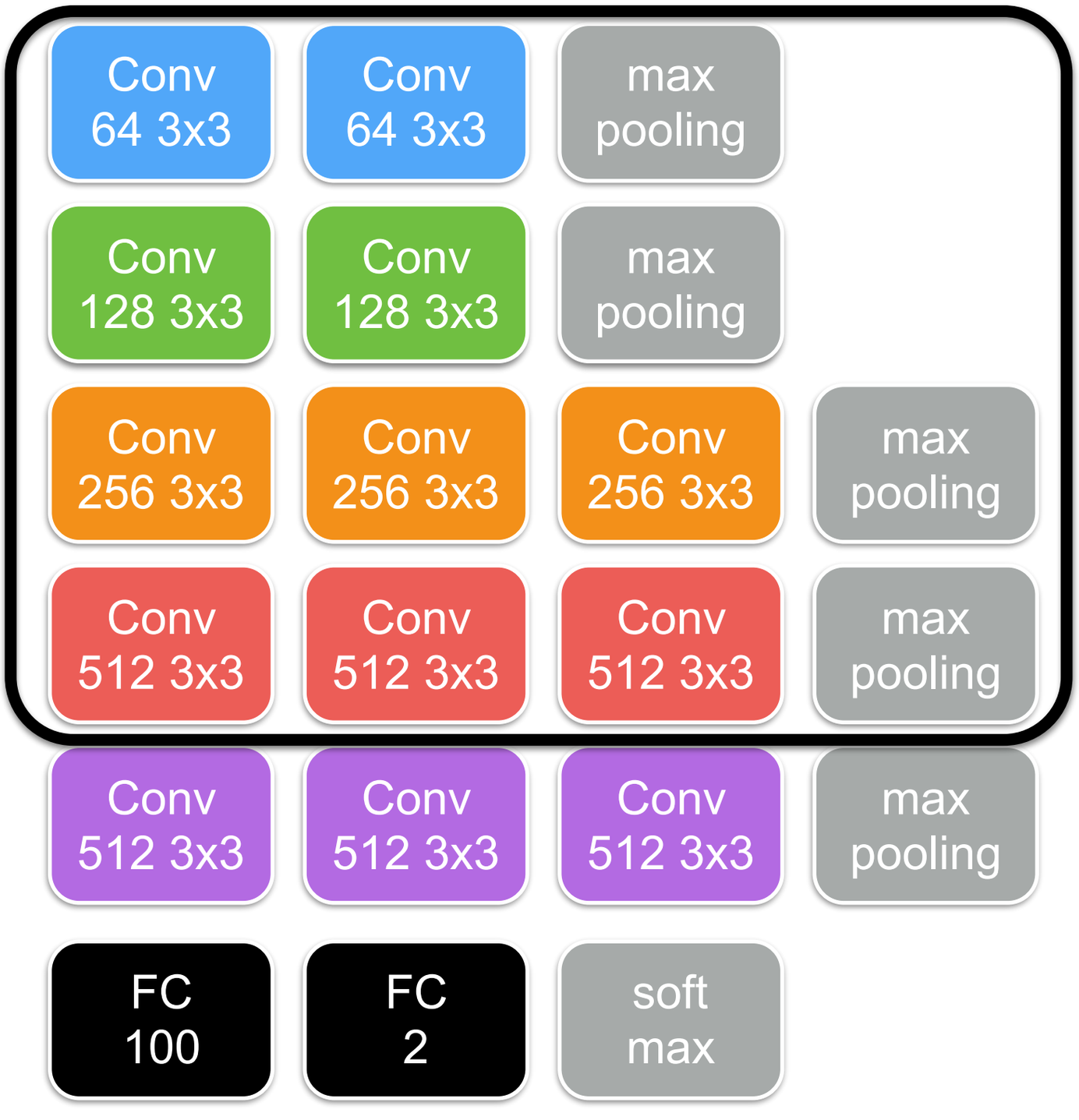
Transfer Learning

VGG-16 architecture



Transfer Learning

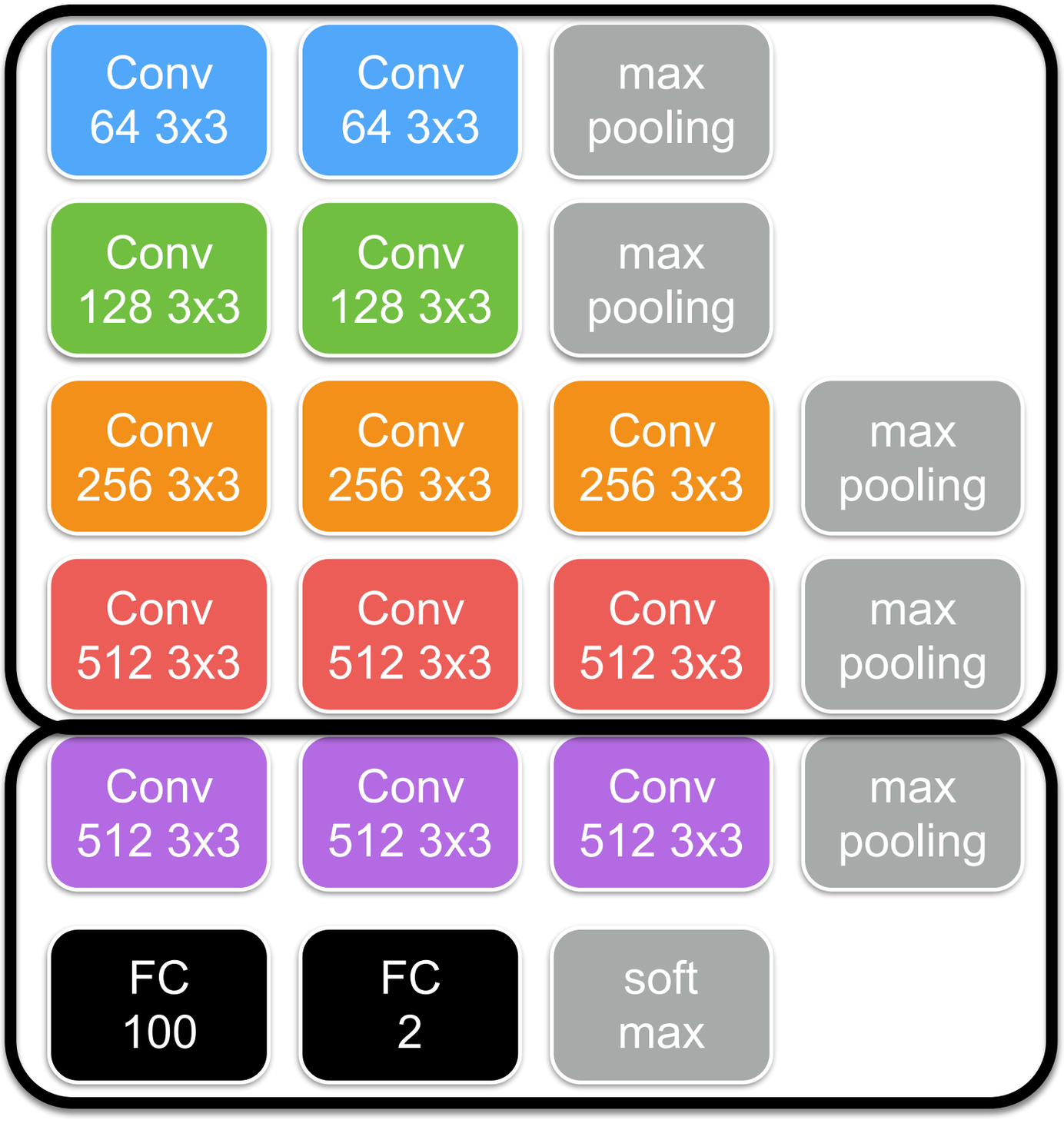
VGG-16 architecture



Fix weights

Transfer Learning

VGG-16 architecture



Fix weights

Train on limited data

Lessons Learned

- Main lessons from this lecture
  - Convolutional neural networks
  - Layers: Fully connected, convolutional, activation functions, max pooling, strides & dilated filters
  - Fully convolutional networks
  - Handling overfitting

Lessons Learned

- Main lessons from this lecture
  - Convolutional neural networks
  - Layers: Fully connected, convolutional, activation functions, max pooling, strides & dilated filters
  - Fully convolutional networks
  - Handling overfitting
- Next lecture: Robust Model Fitting