SSY097 - Image Analysis

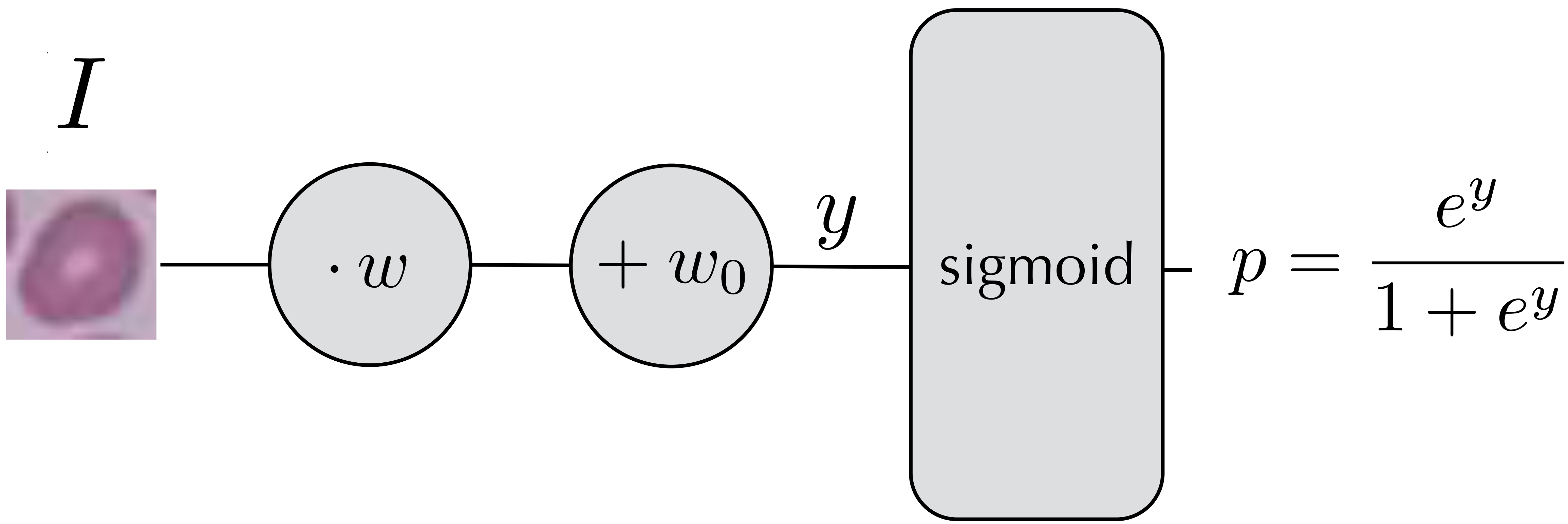
Lecture 5 - (Convolutional) Neural Networks

Torsten Sattler
(slides adapted from Olof Enqvist)

Last Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	**Learning a classifier**	Lab 2
Feb. 3	Convolutional neural networks	
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	

Last Lecture



Learning linear classifiers

Last Lecture

$$p_k = \frac{e^{y_k}}{\sum_{m=0}^{n} e^{y_m}}$$

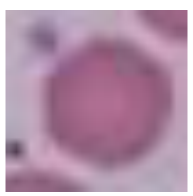
Softmax function for multi-class classification

Last Lecture

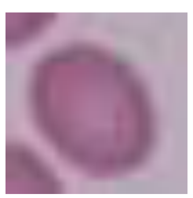
p_1

$\longrightarrow p_4$

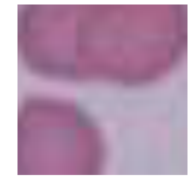
$\longrightarrow p_5$



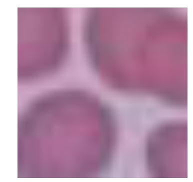
$\longrightarrow p_6$



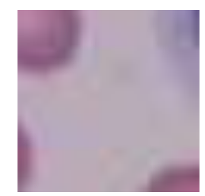
$\longrightarrow p_8$



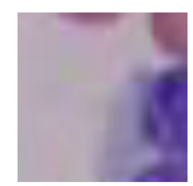
$\longrightarrow p_2$



$\longrightarrow p_3$



$\longrightarrow p_7$



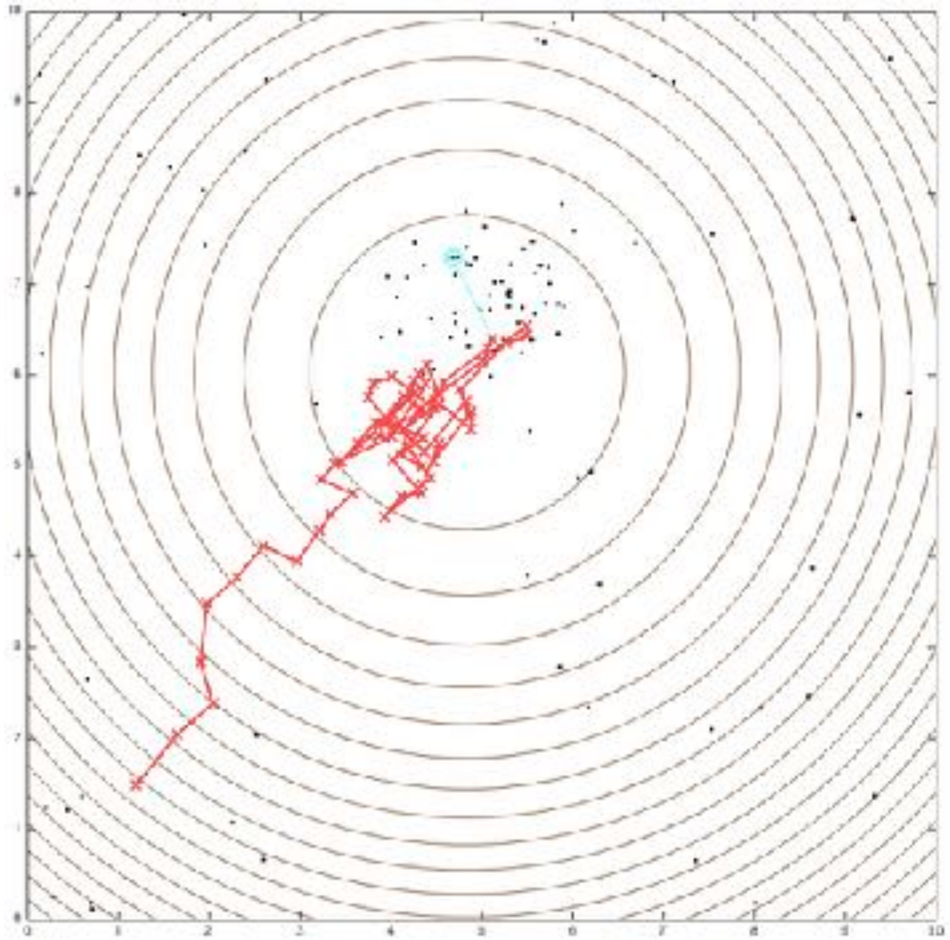
$\longrightarrow p_9$

$$-\Big(\sum_{\text{positive examples}} \ln p_i + \sum_{\text{negative examples}} \ln (1 - p_i) \Big)$$

Loss function: Negative Log-Likelihood

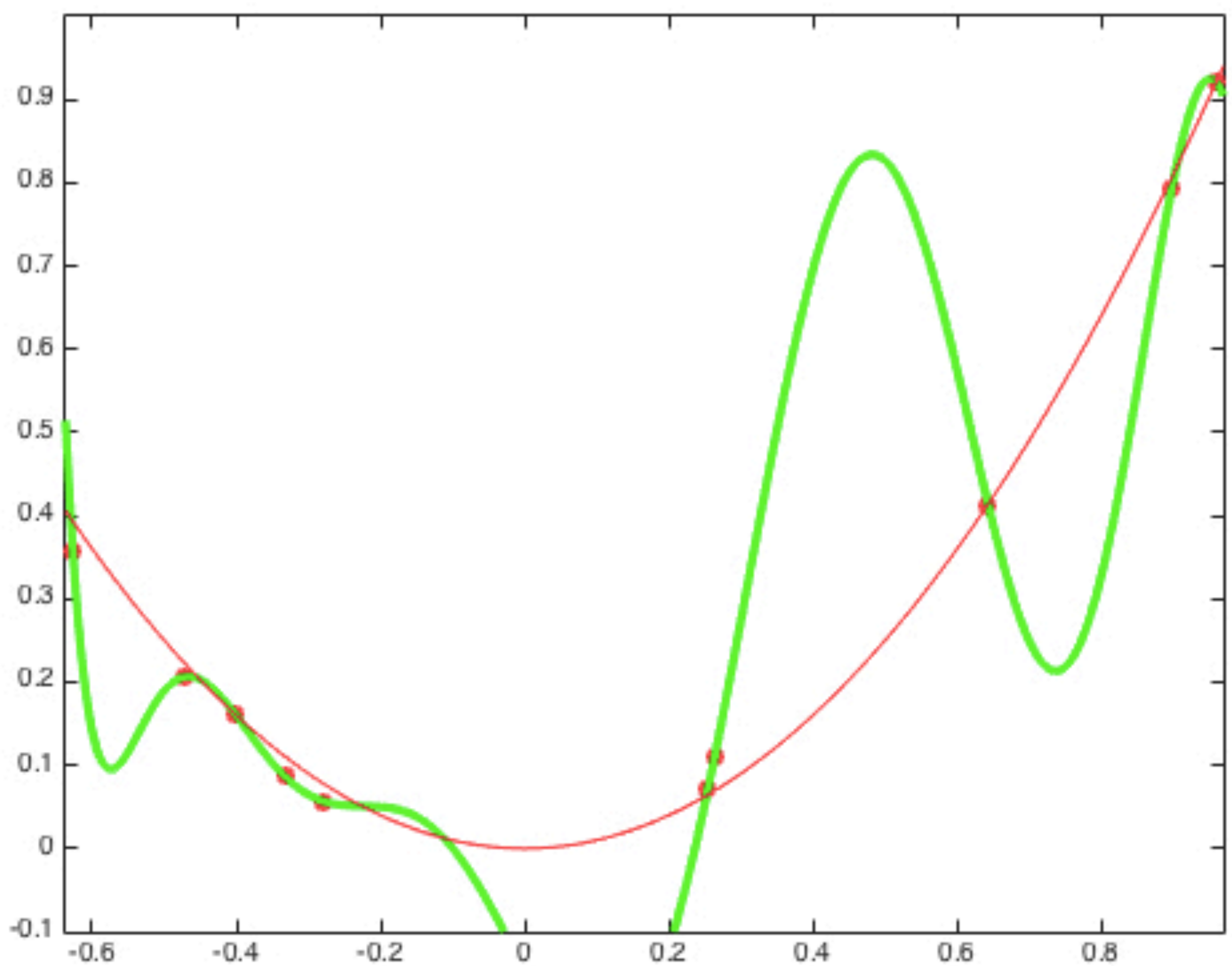
Last Lecture

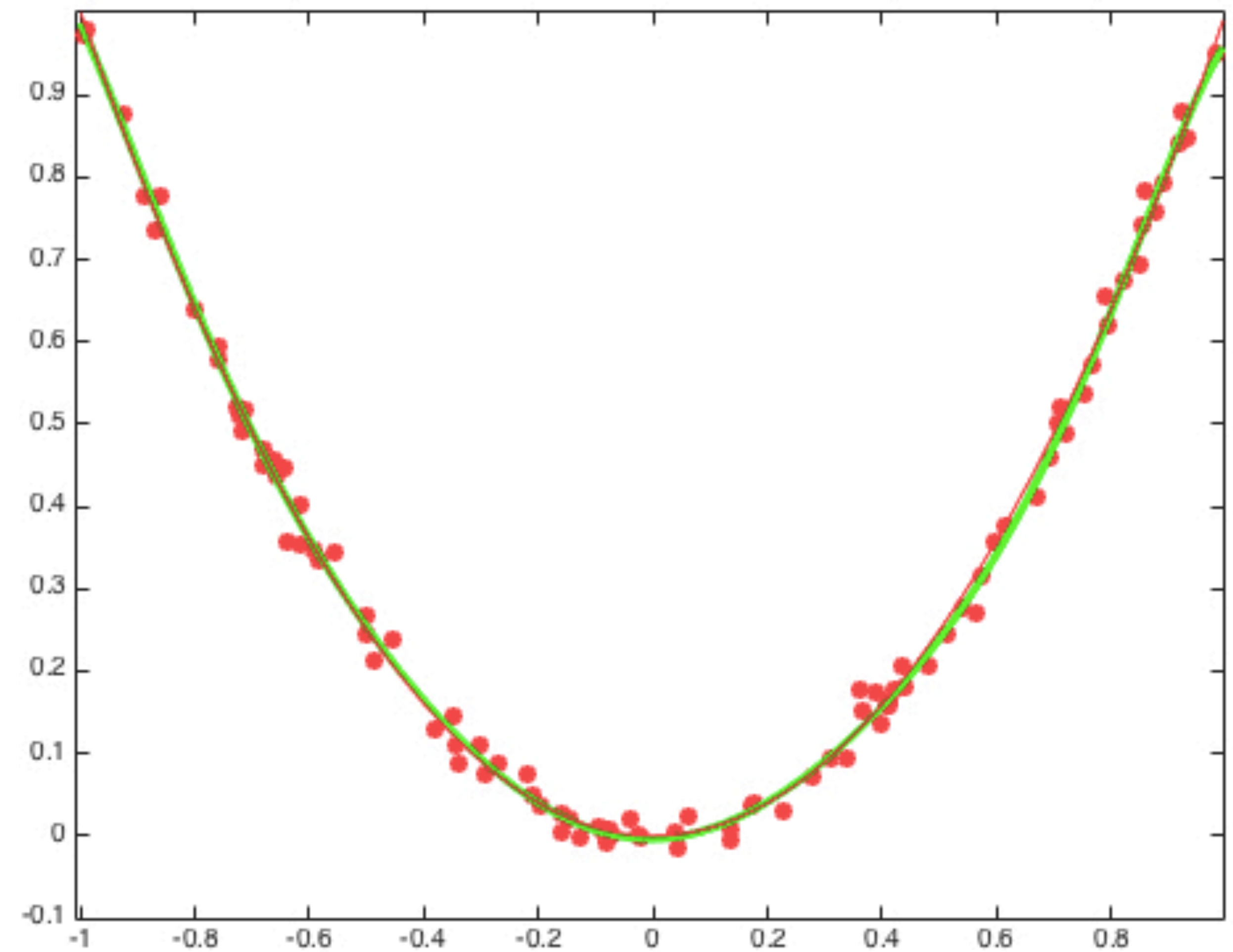
$$\theta^{(k+1)} = \theta^{(k)} - \mu \sum_i \nabla L_i(\theta) \approx \theta^{(k)} - \mu \nabla L_i(\theta)$$



Stochastic Gradient Descent (with Momentum)

Last Lecture

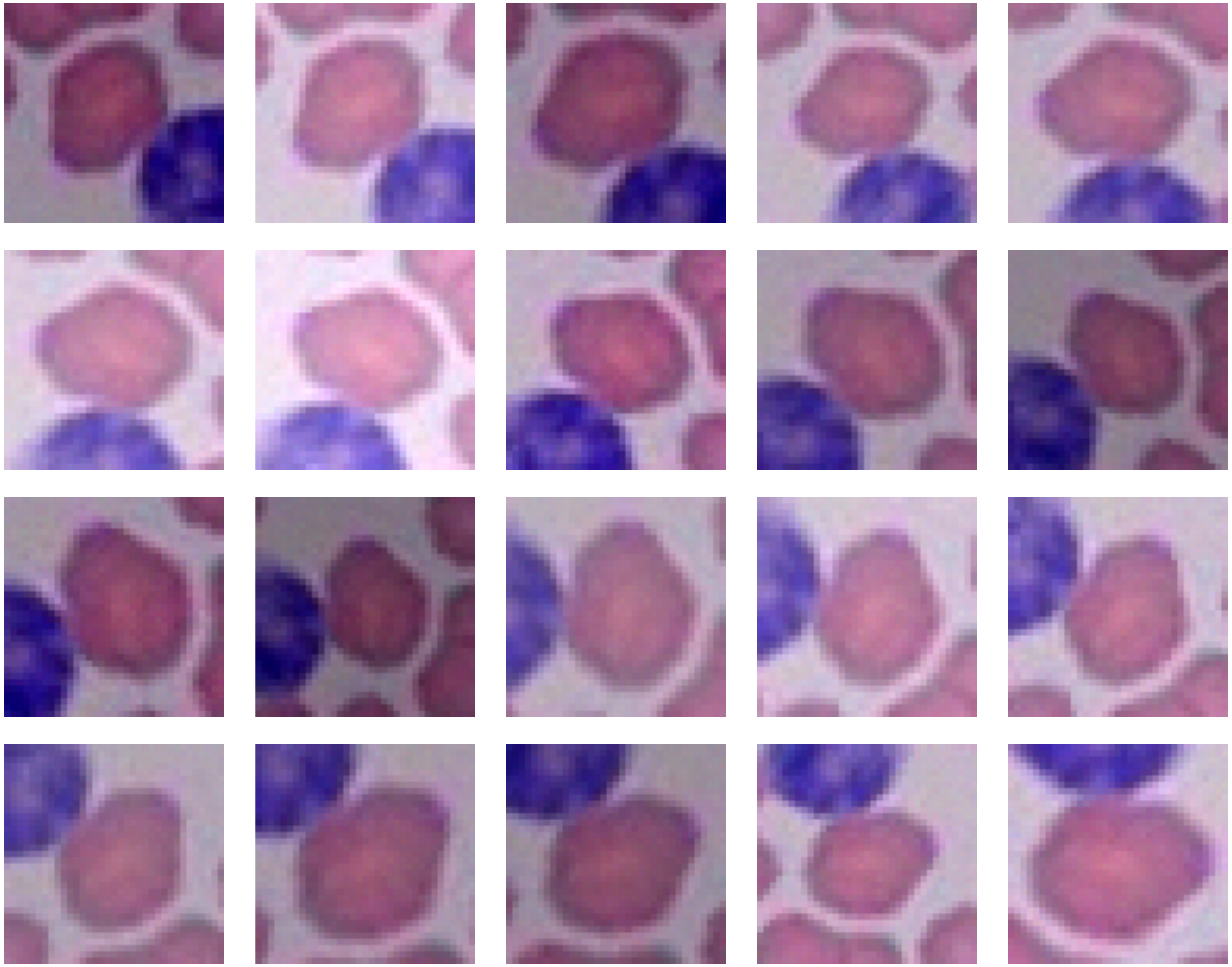




Overfitting

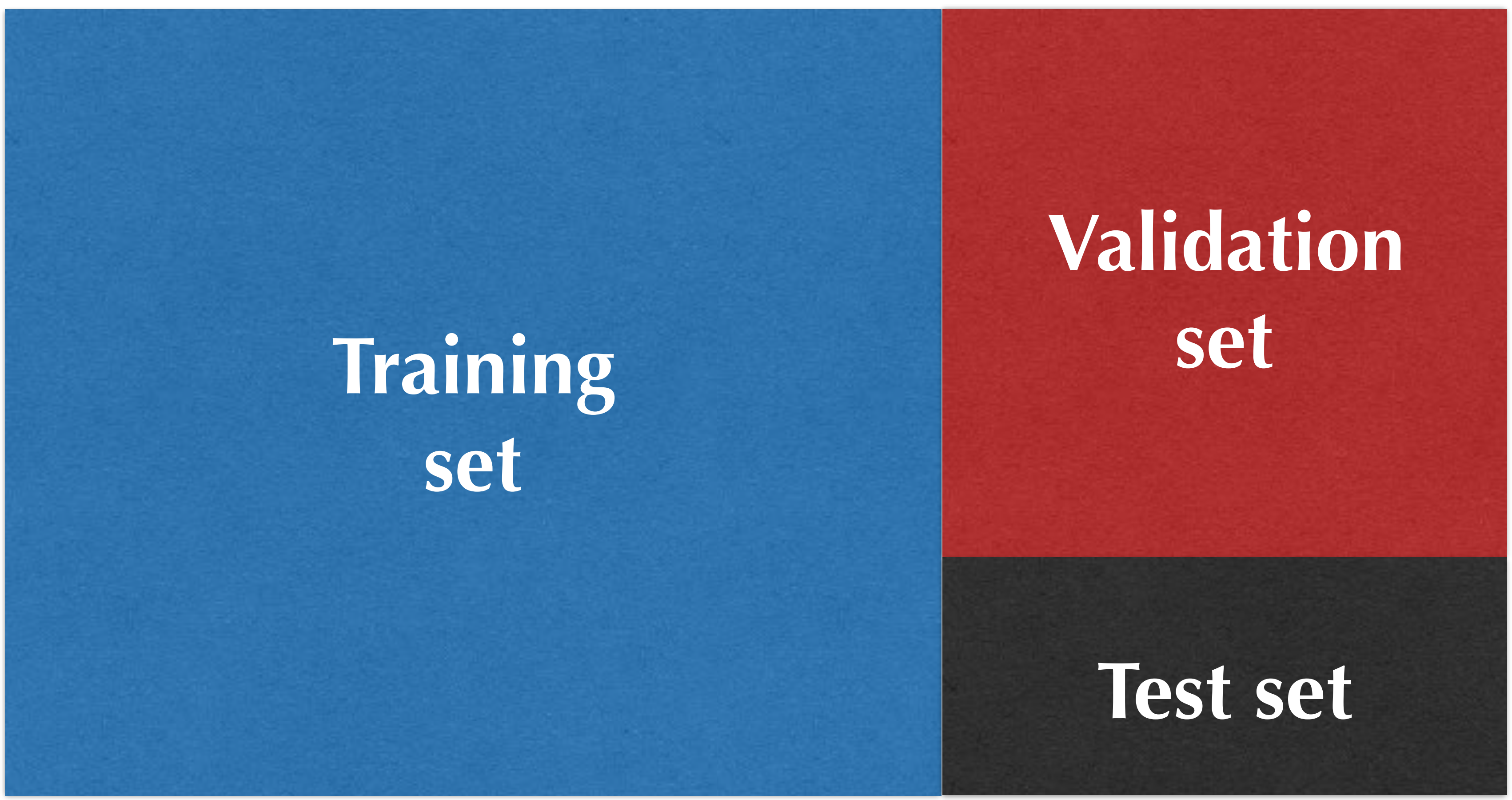
Last Lecture

- Rotate
- Scale
- Change brightness
- Add noise



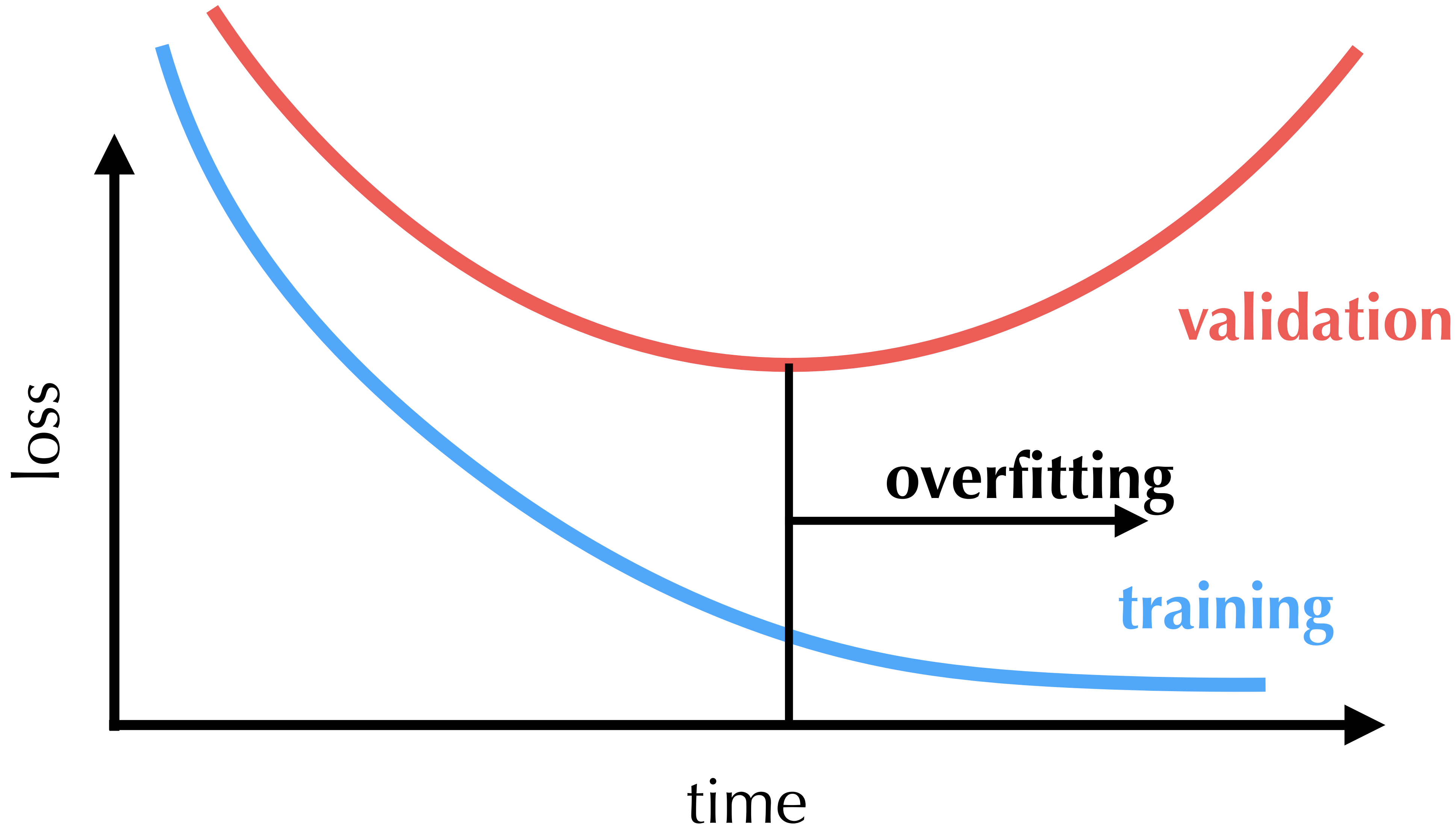
Data augmentation

Last Lecture



Training, validation, test sets

Last Lecture



Training, validation, test sets

Today

- Neural Networks
- Optimization: Backpropagation
- Convolutional Neural Networks
- Overfitting, Part II

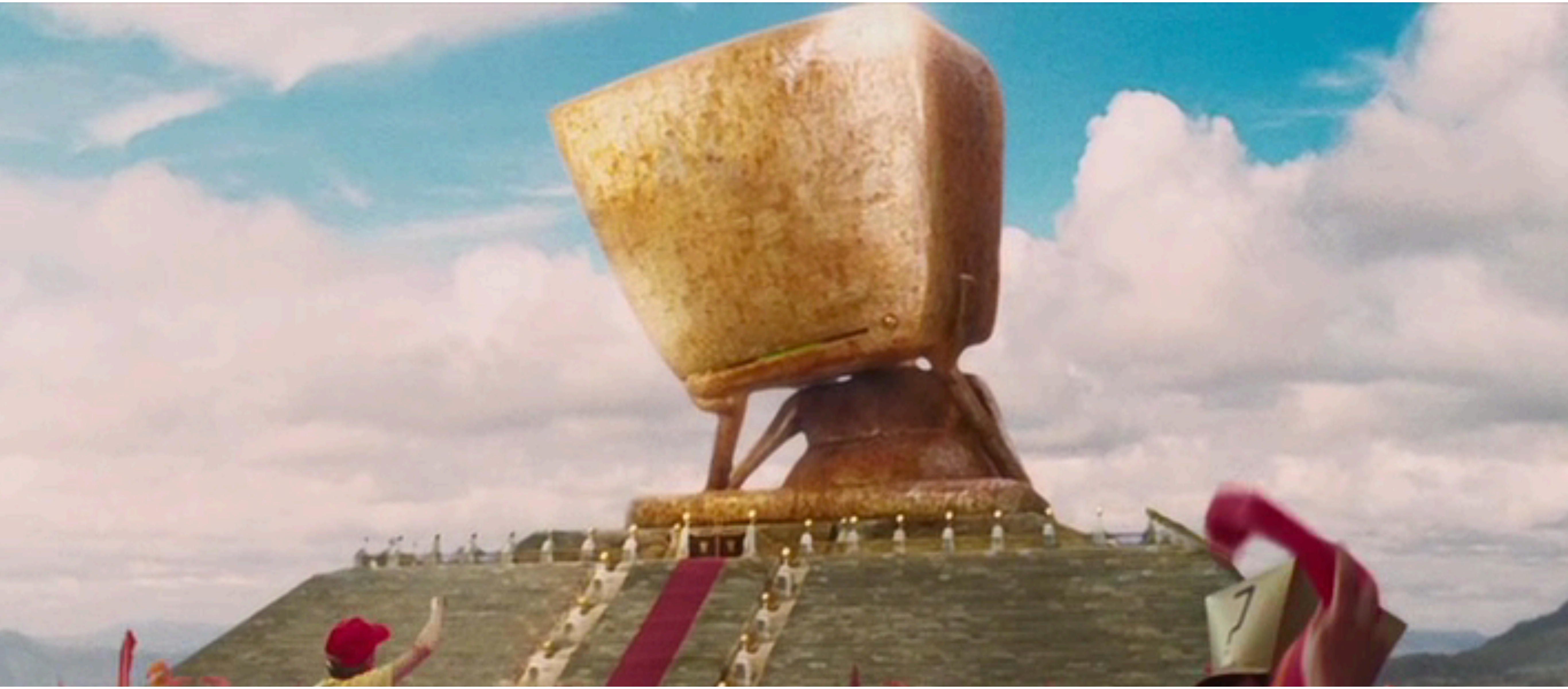
Neural Networks

Recognition Revisited

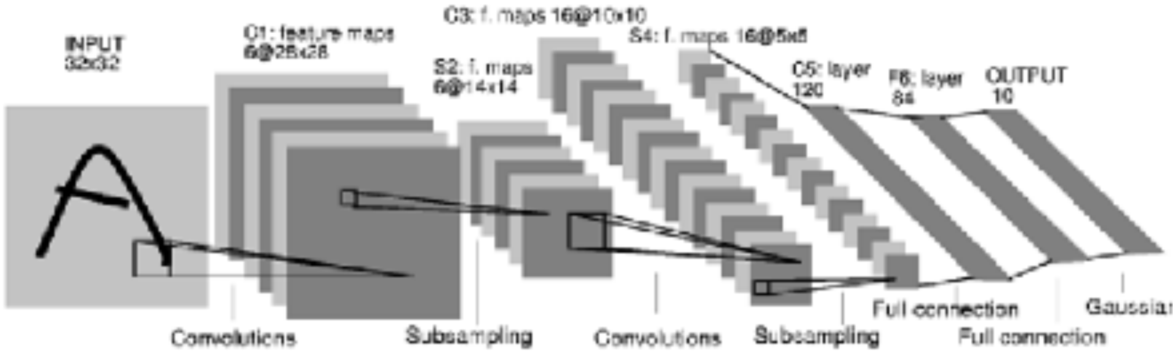
Is there a robin in this image?

Modern Solution: Deep Learning

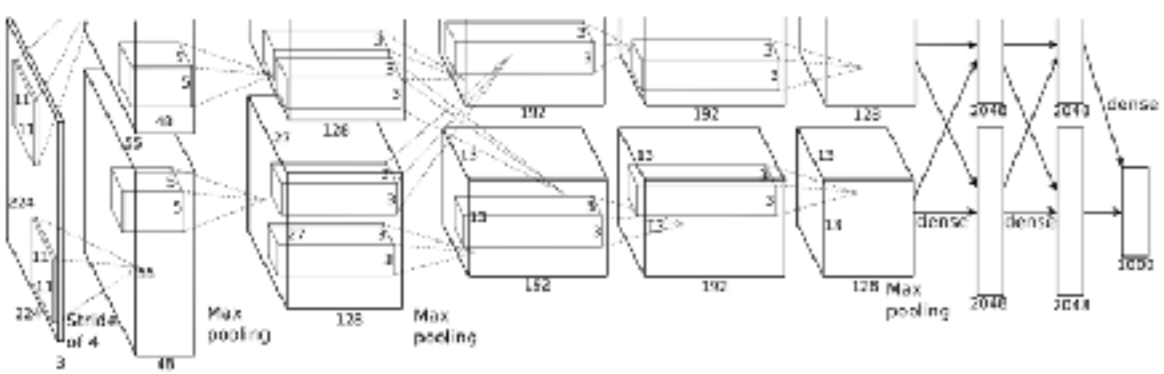
Modern Solution: Deep Learning



Deep Neural Networks

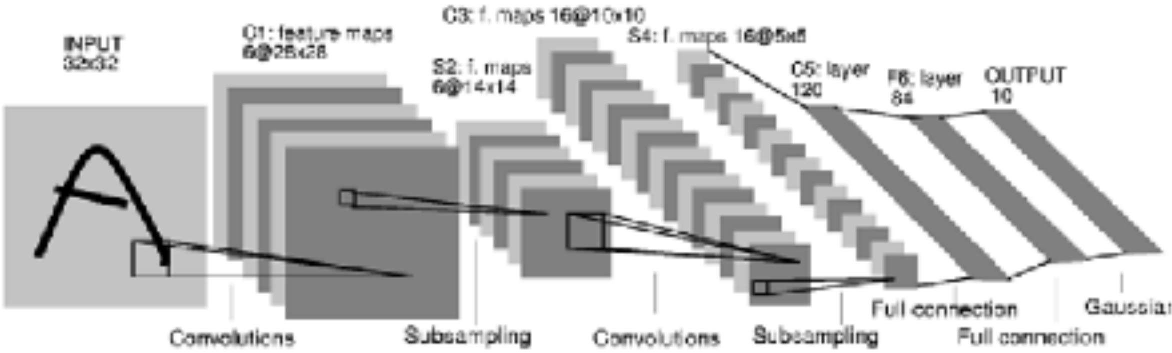


[LeCun et al. 1998]: 7 layers

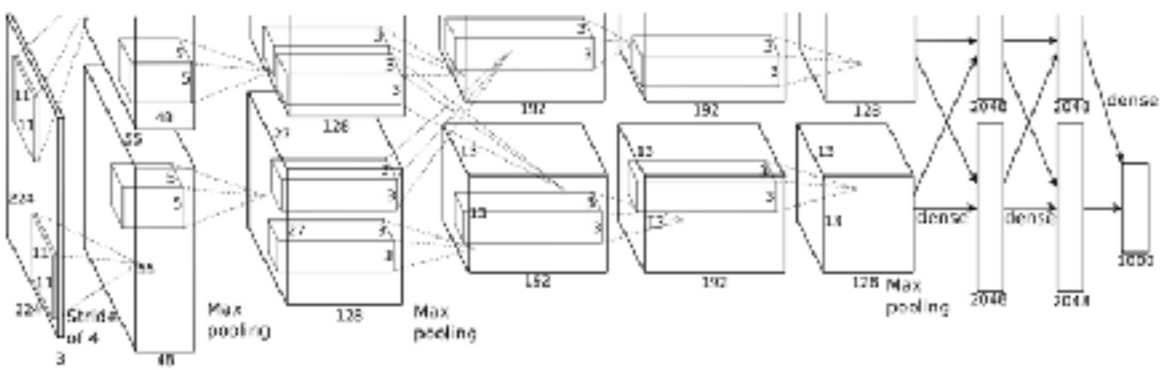


[Krizhevsky et al. 2012]: 8 layers

Deep Neural Networks



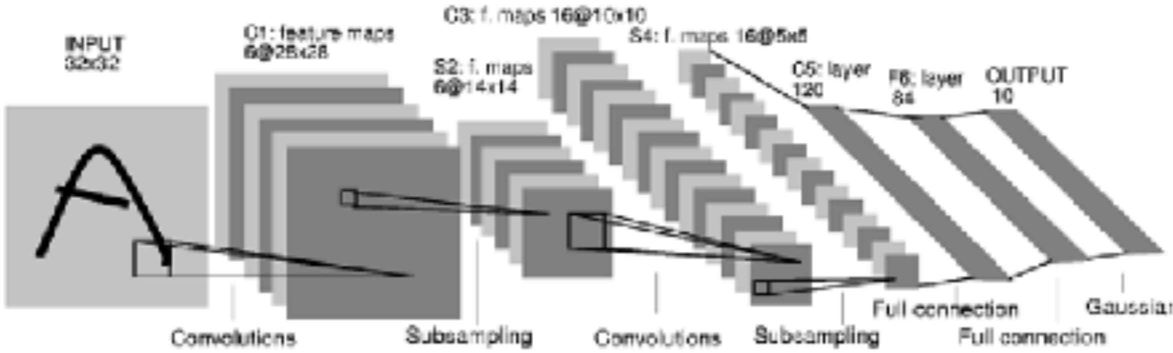
[LeCun et al. 1998]: 7 layers



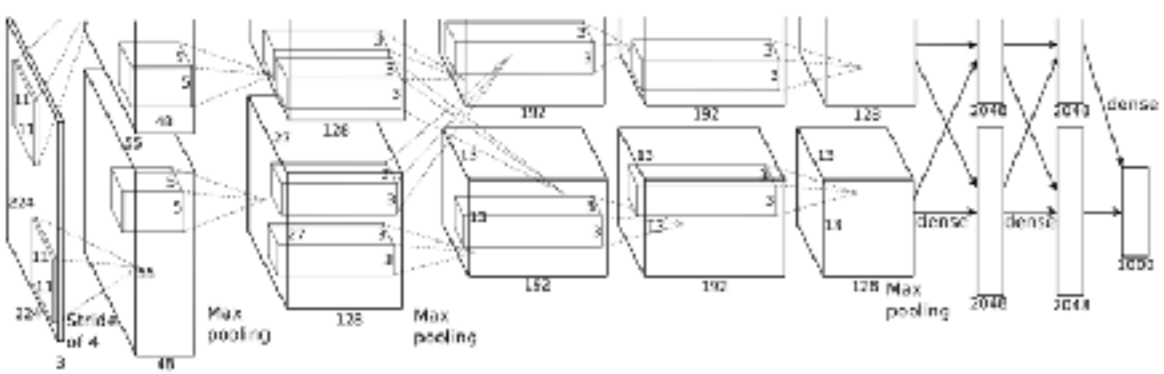
[Krizhevsky et al. 2012]: 8 layers

[Szegedy et al. 2014]:
22 layers

Deep Neural Networks

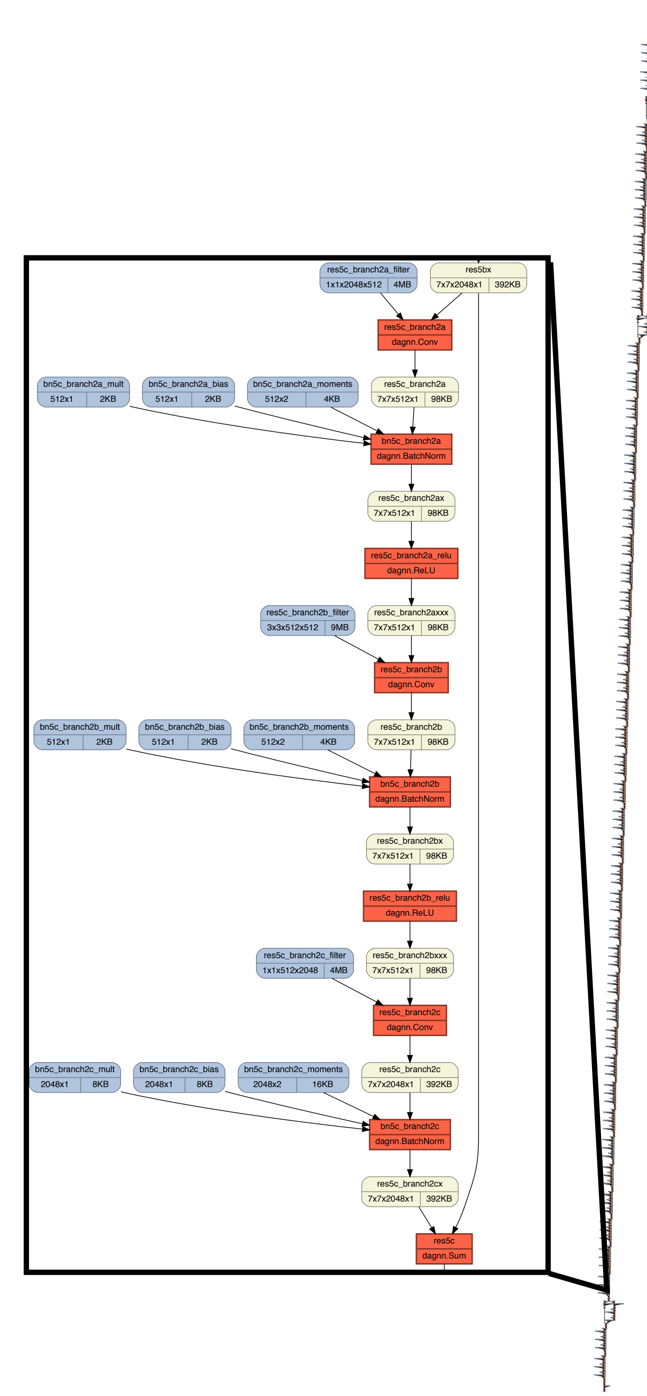


[LeCun et al. 1998]: 7 layers



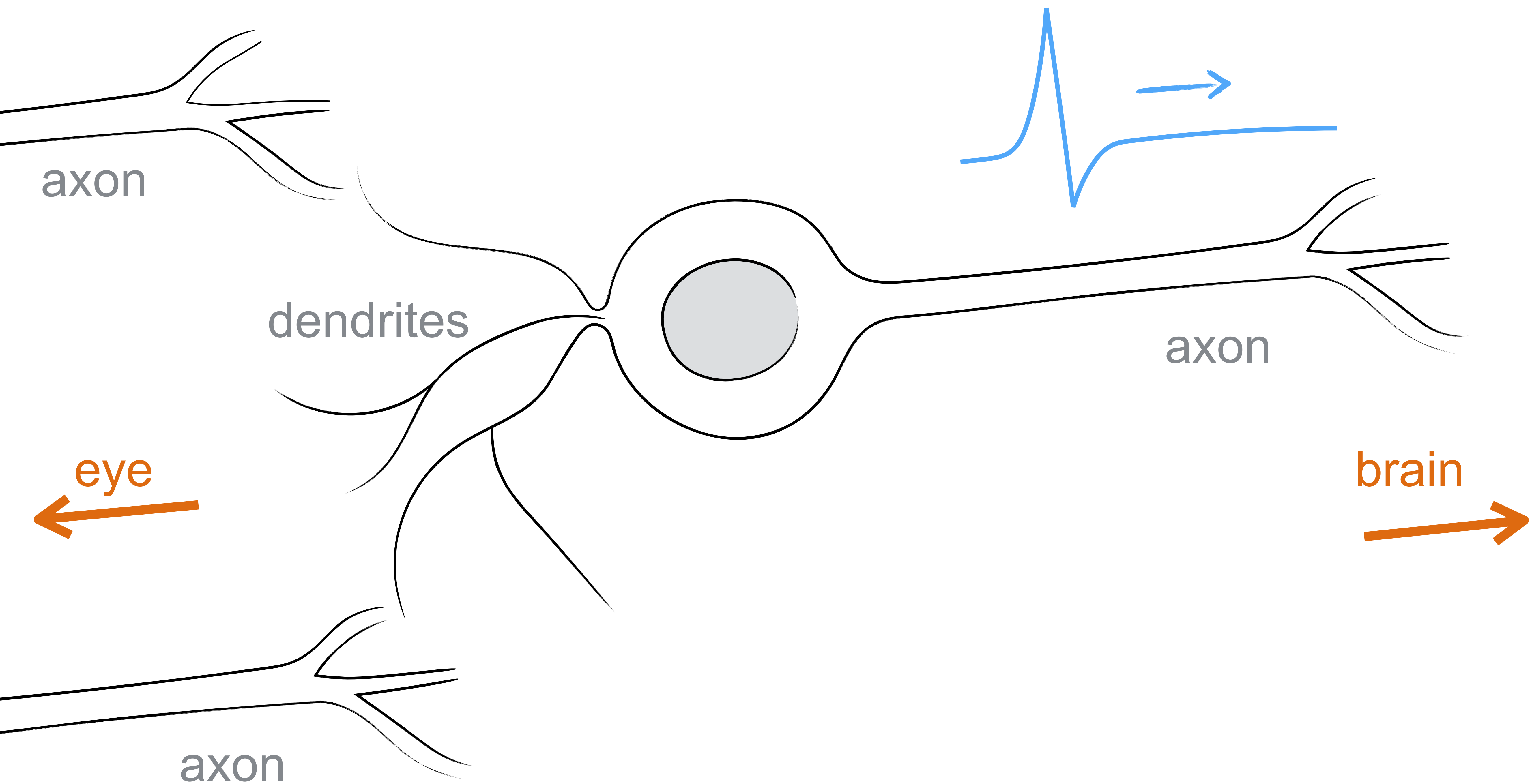
[Krizhevsky et al. 2012]: 8 layers

[Szegedy et al. 2014]: 22 layers

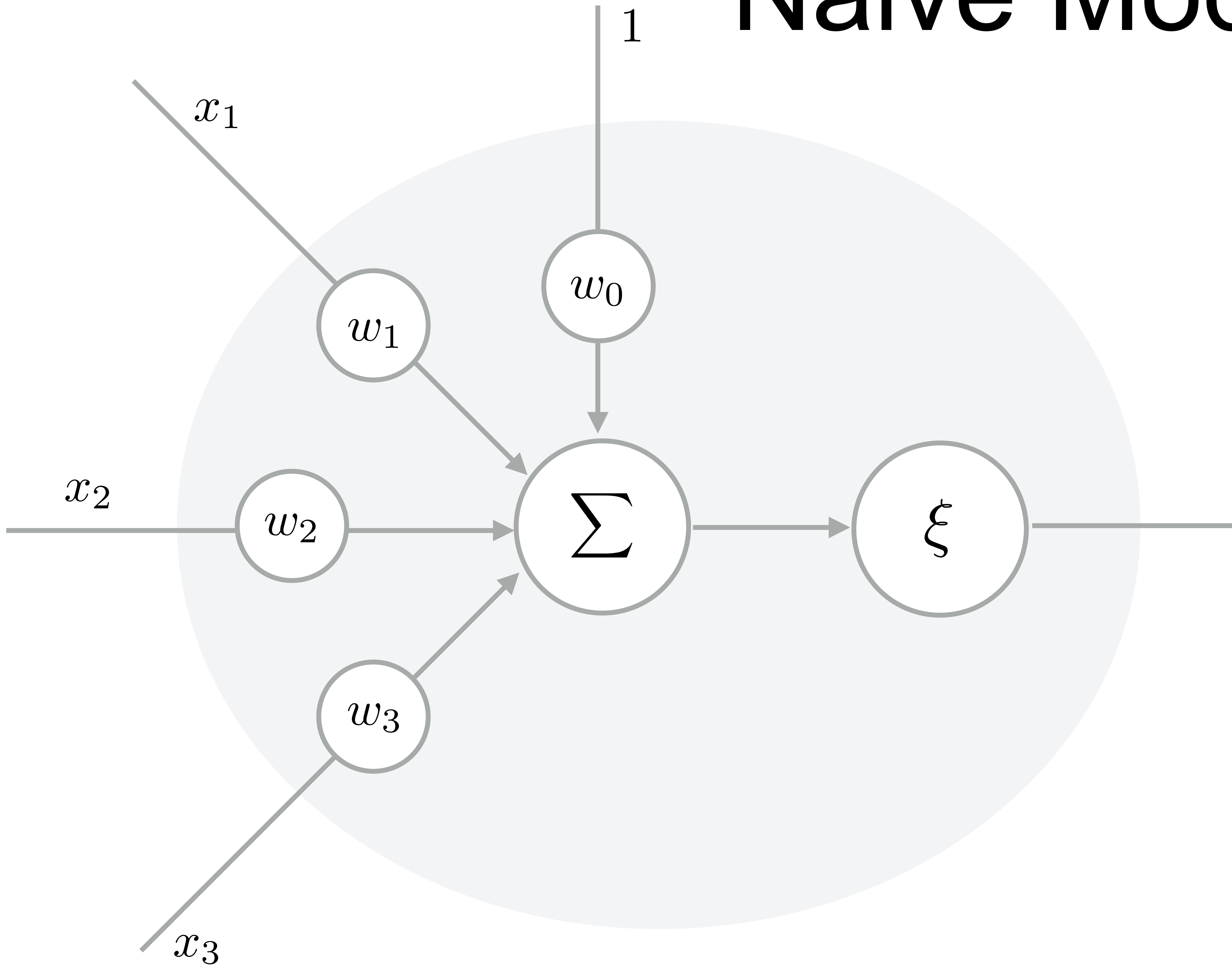


[He et al. 2015]: 152 layers

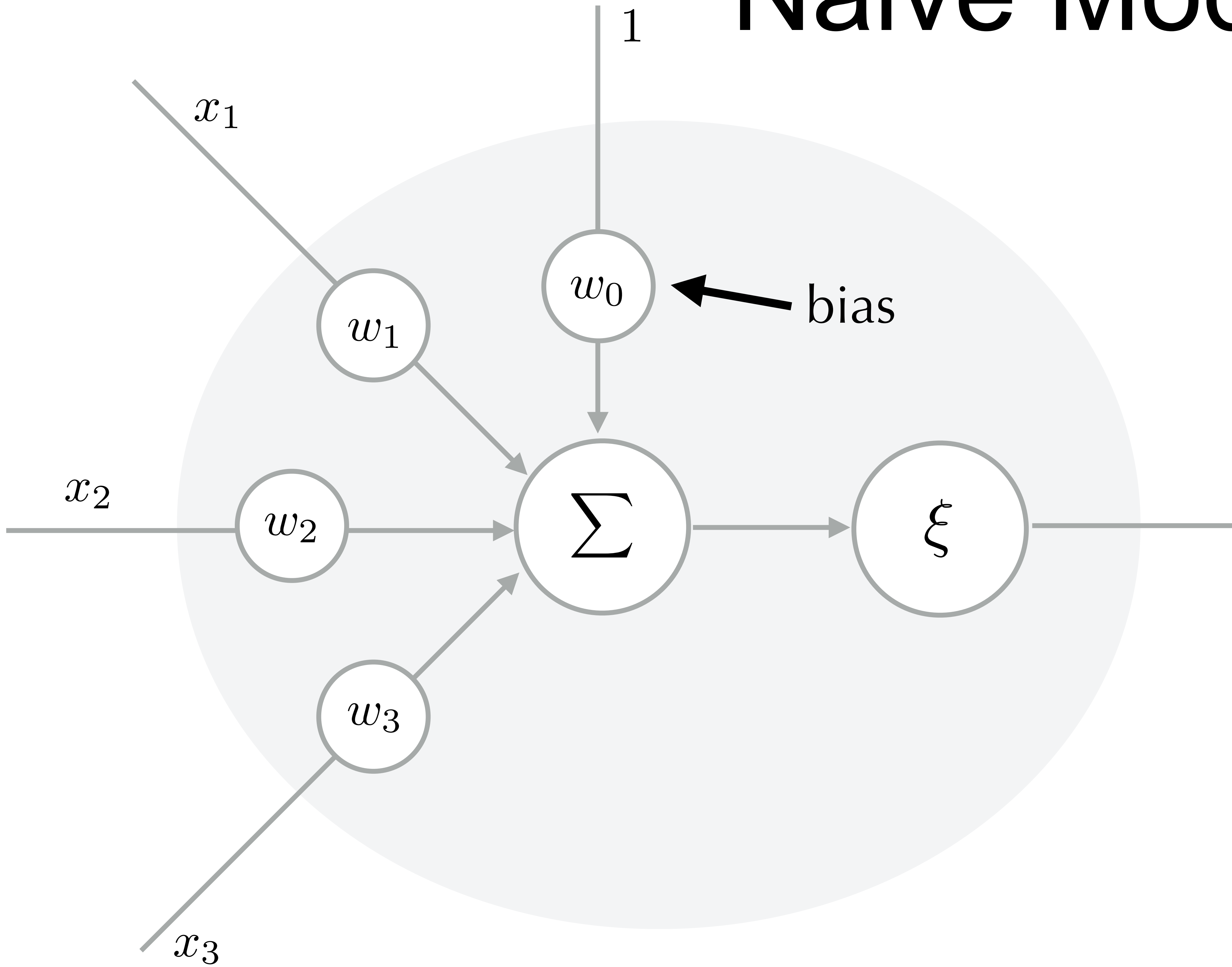
A Neuron



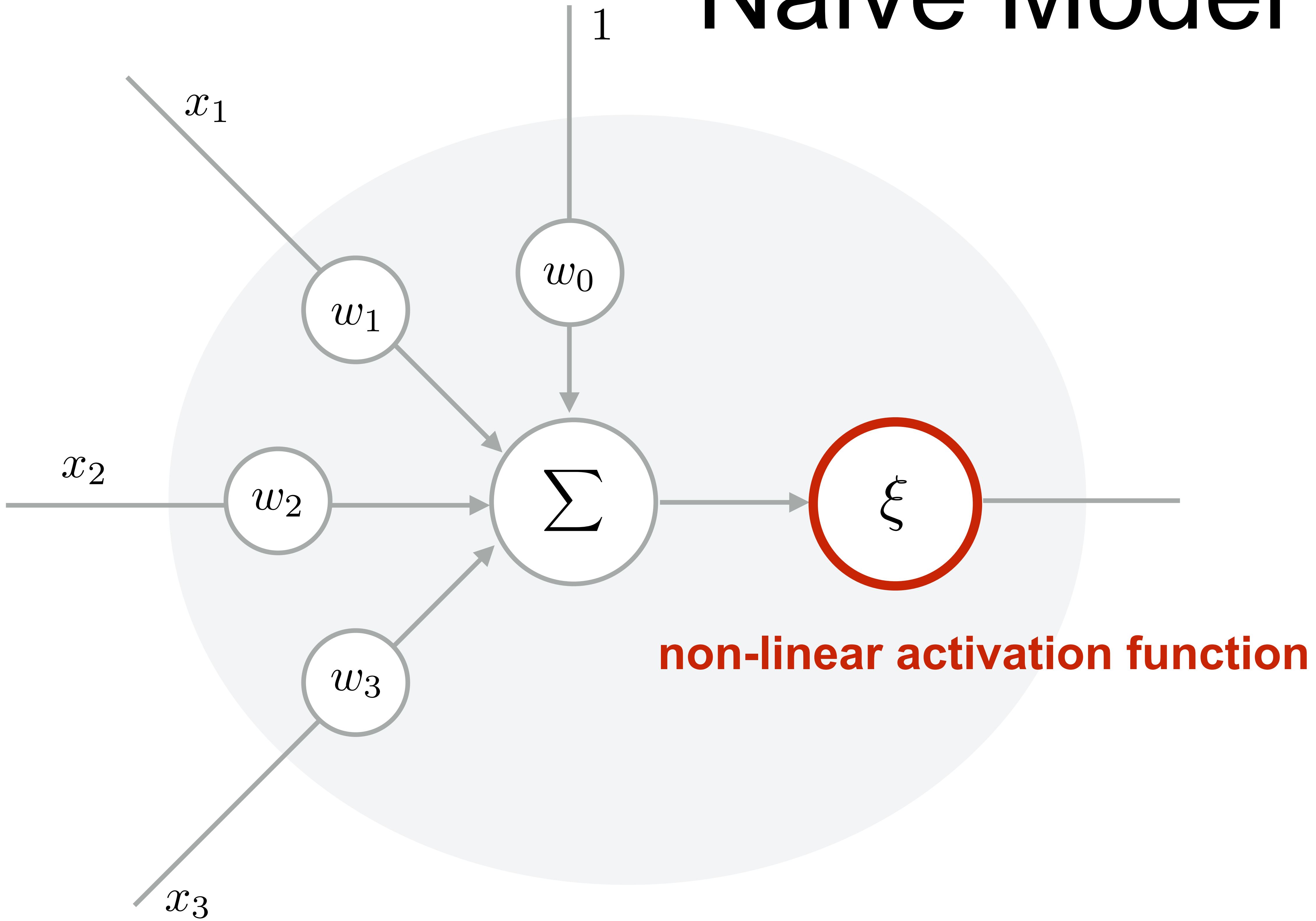
Naive Model



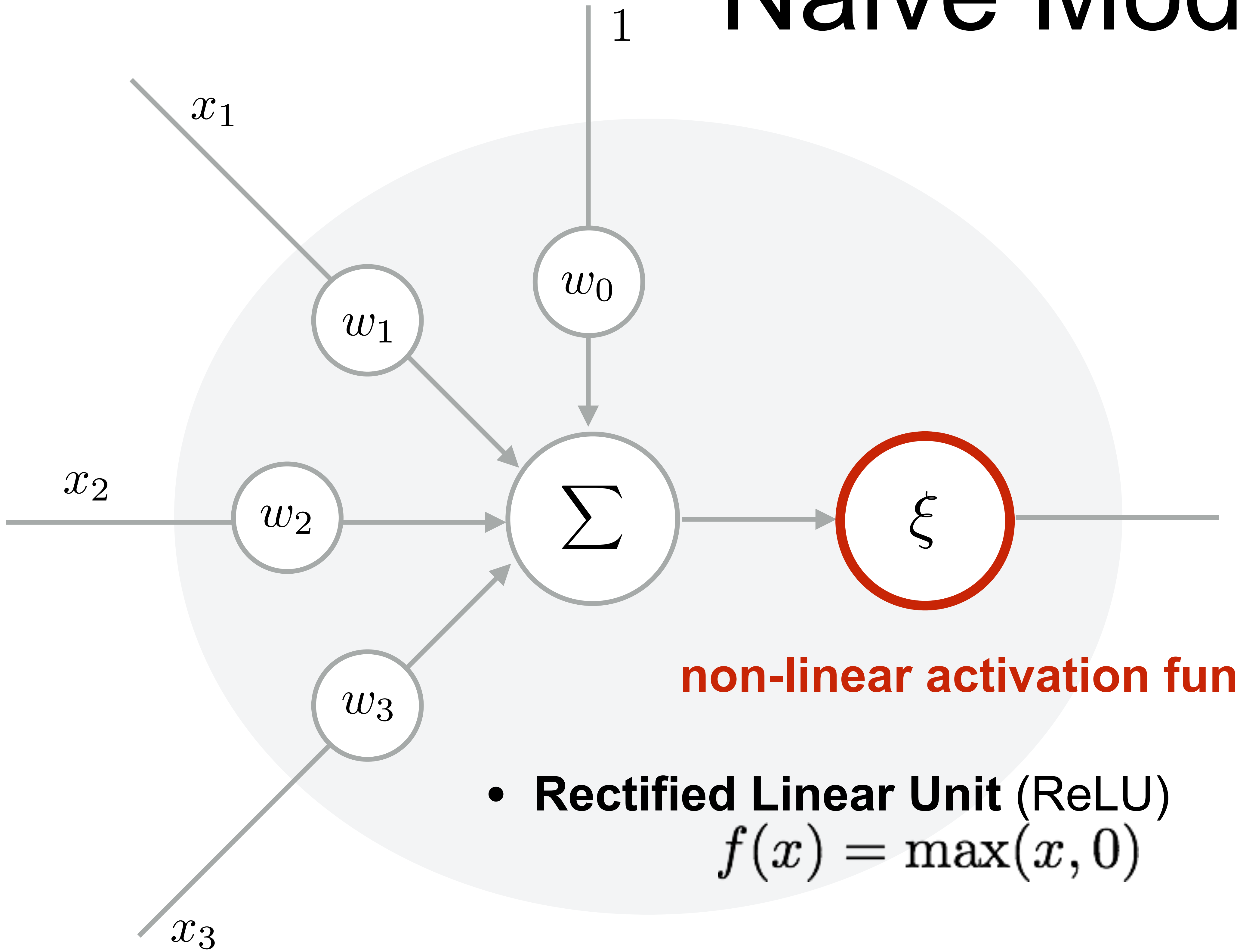
Naive Model



Naive Model



Naive Model

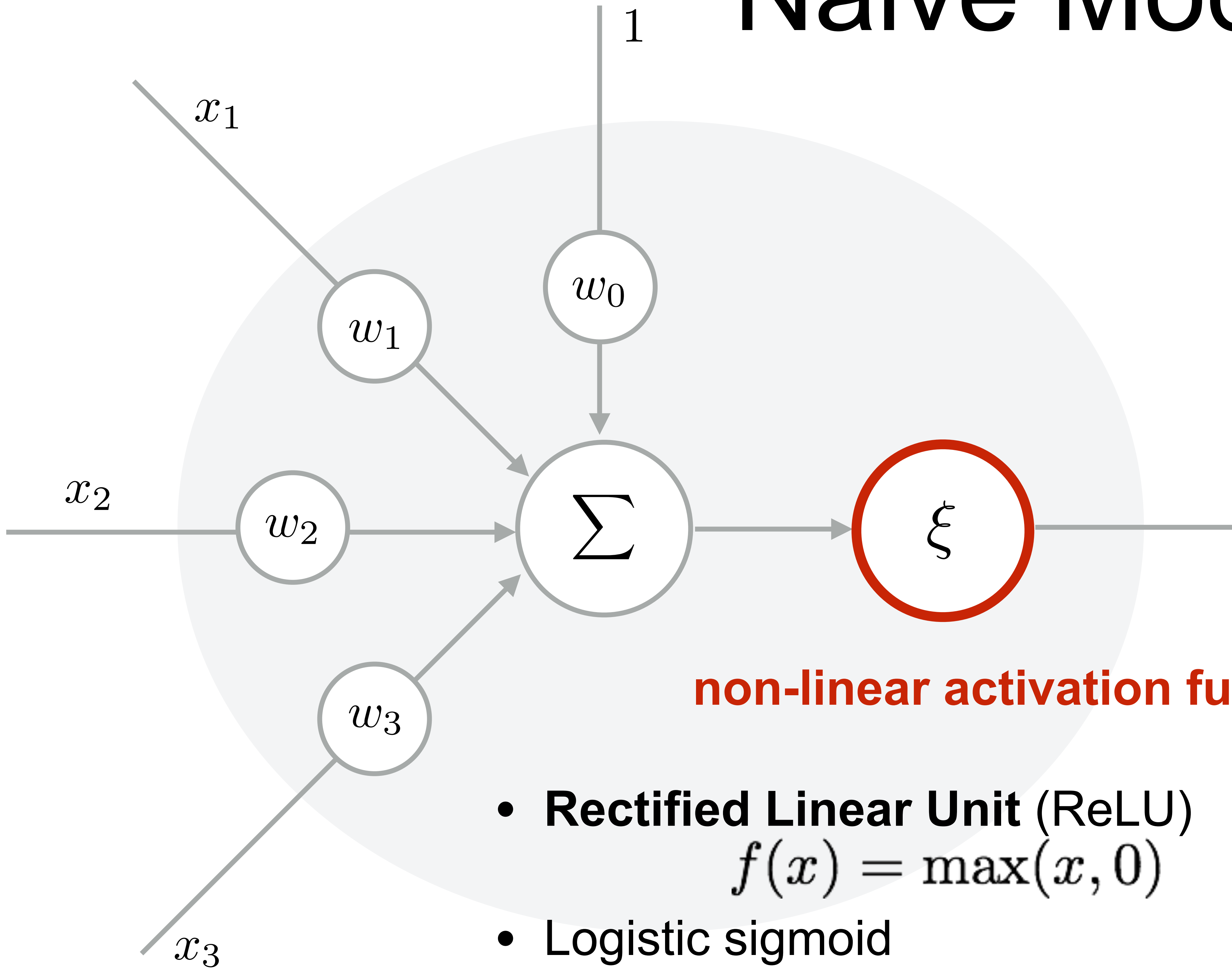


non-linear activation function

- **Rectified Linear Unit** (ReLU)

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

Naive Model

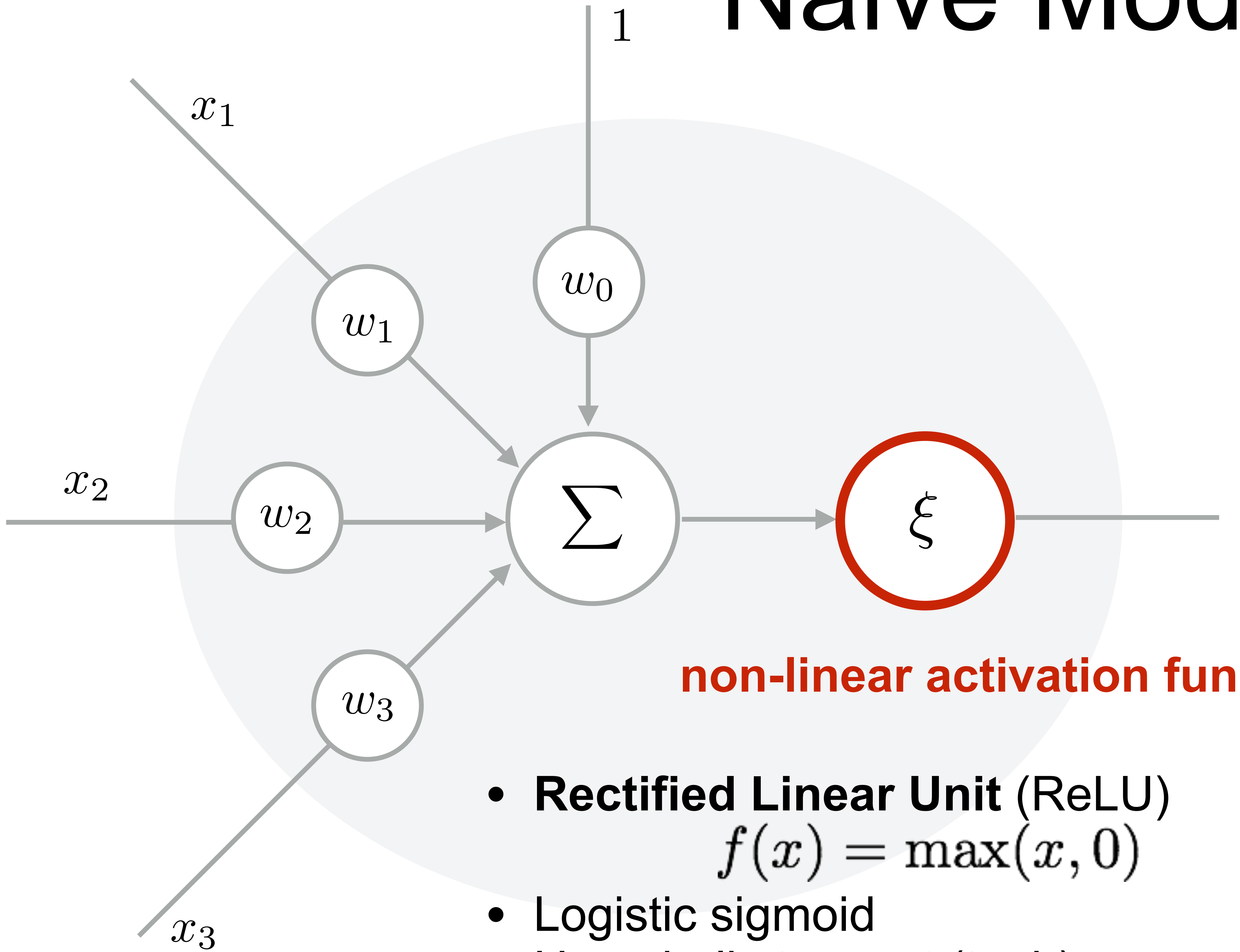


- **Rectified Linear Unit** (ReLU)

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

- Logistic sigmoid

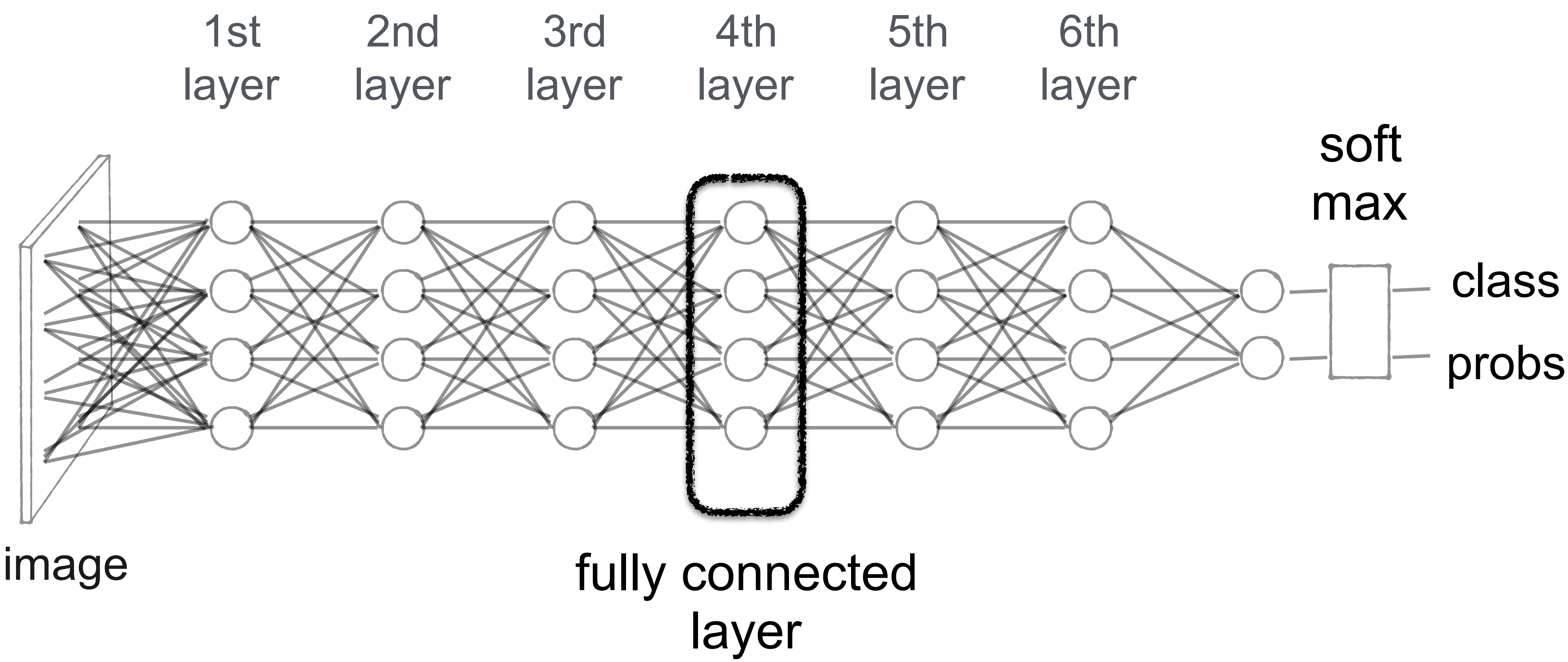
Naive Model



non-linear activation function

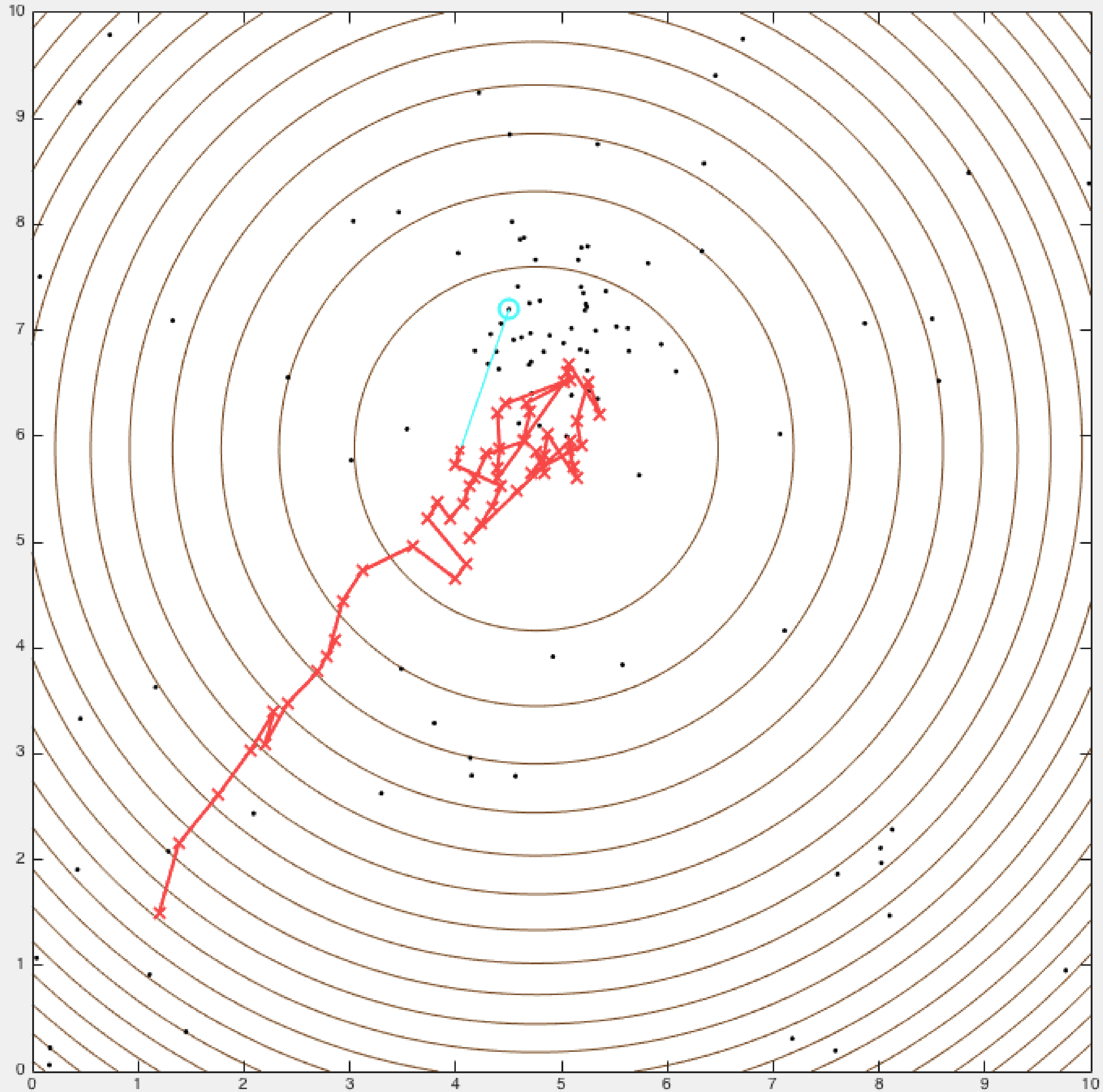
- **Rectified Linear Unit** (ReLU)
  $$f(x) = \max(x, 0)$$
- Logistic sigmoid
- Hyperbolic tangent (tanh)
- …

A Neural Network



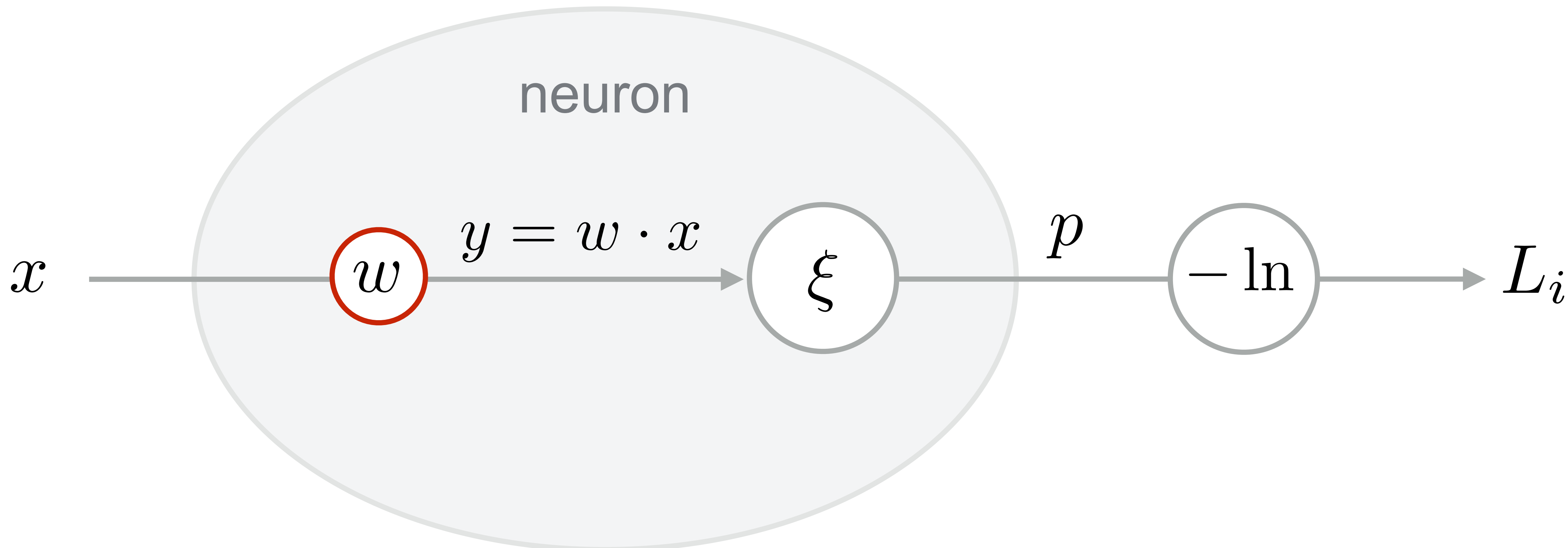
Optimization: Backpropagation
(backprop)

Stochastic Gradient Descent

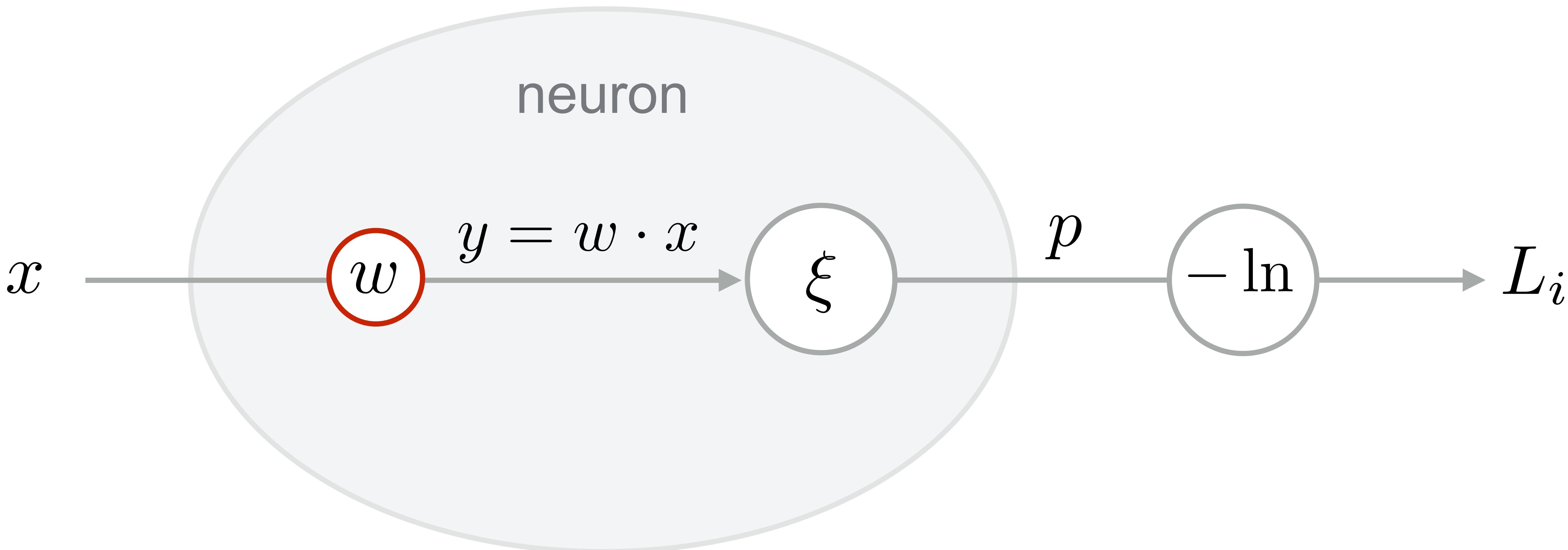


How do we differentiate?

The Chain Rule

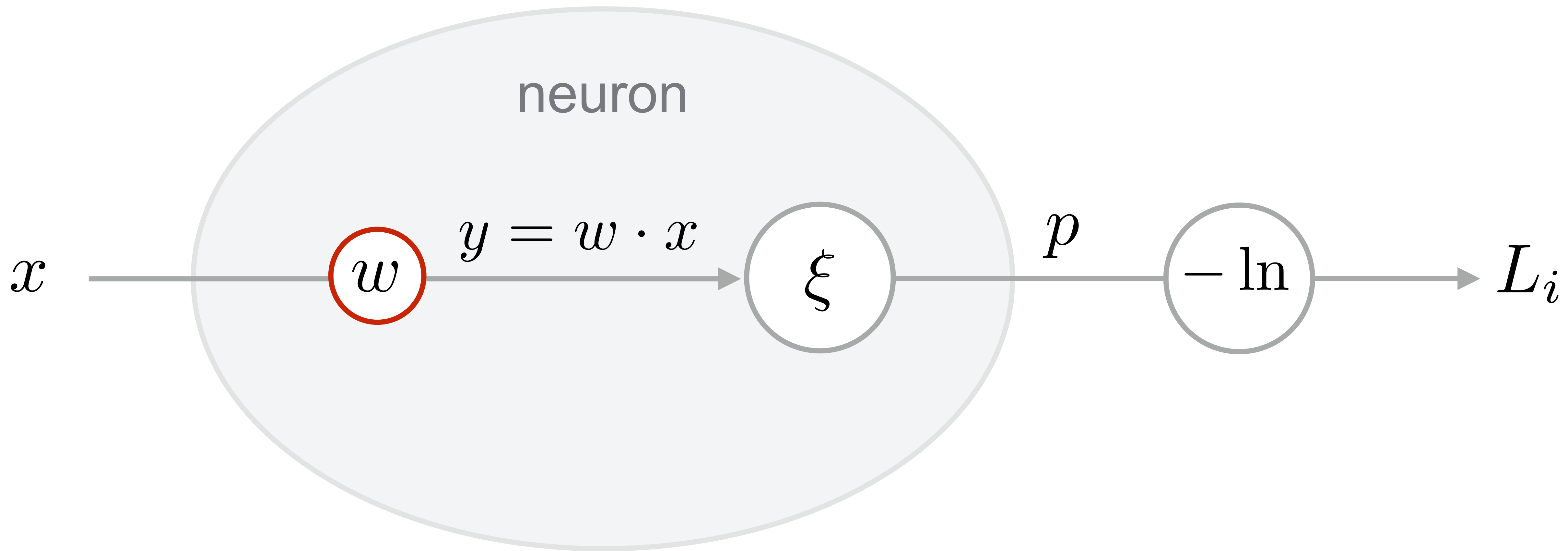


The Chain Rule



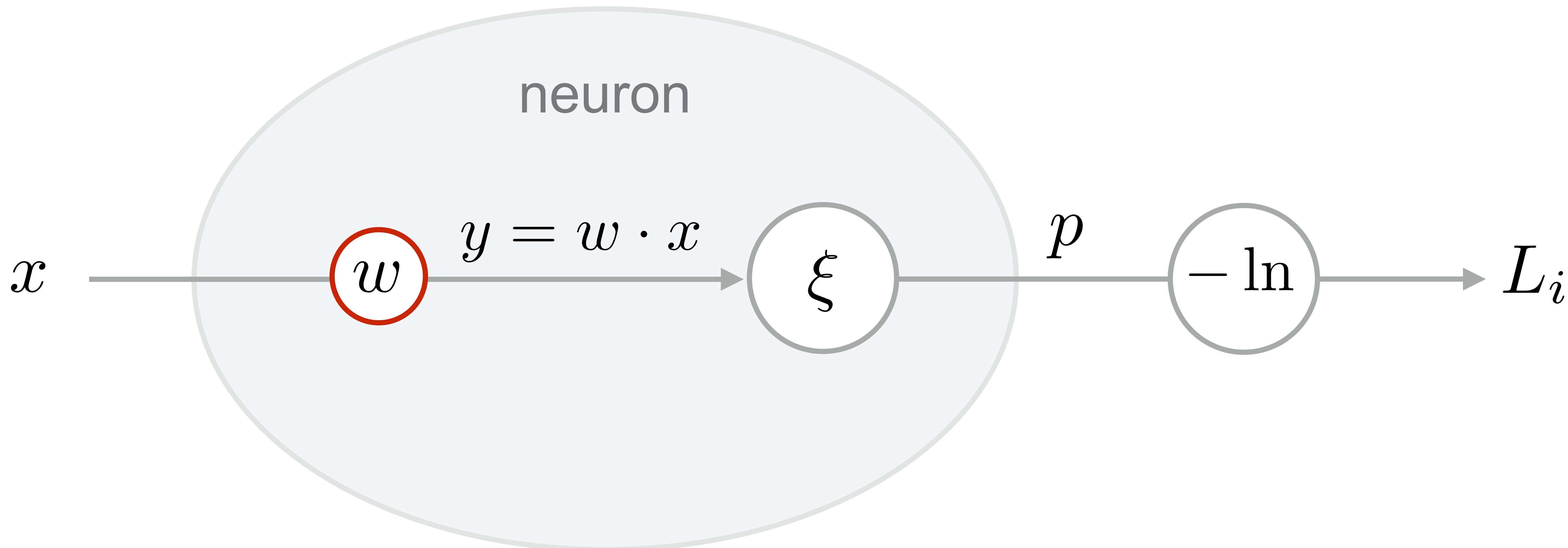
How to compute a gradient for parameter w?

The Chain Rule



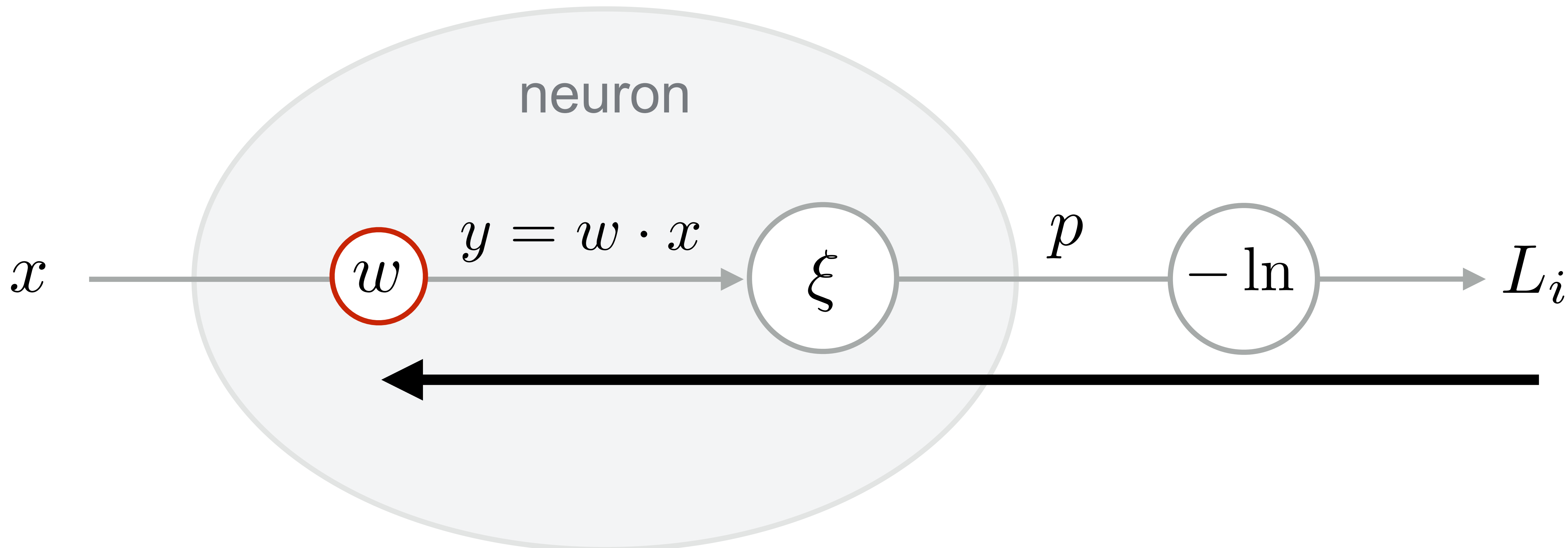
How to compute a gradient for parameter w?
Apply the Chain Rule!

The Chain Rule



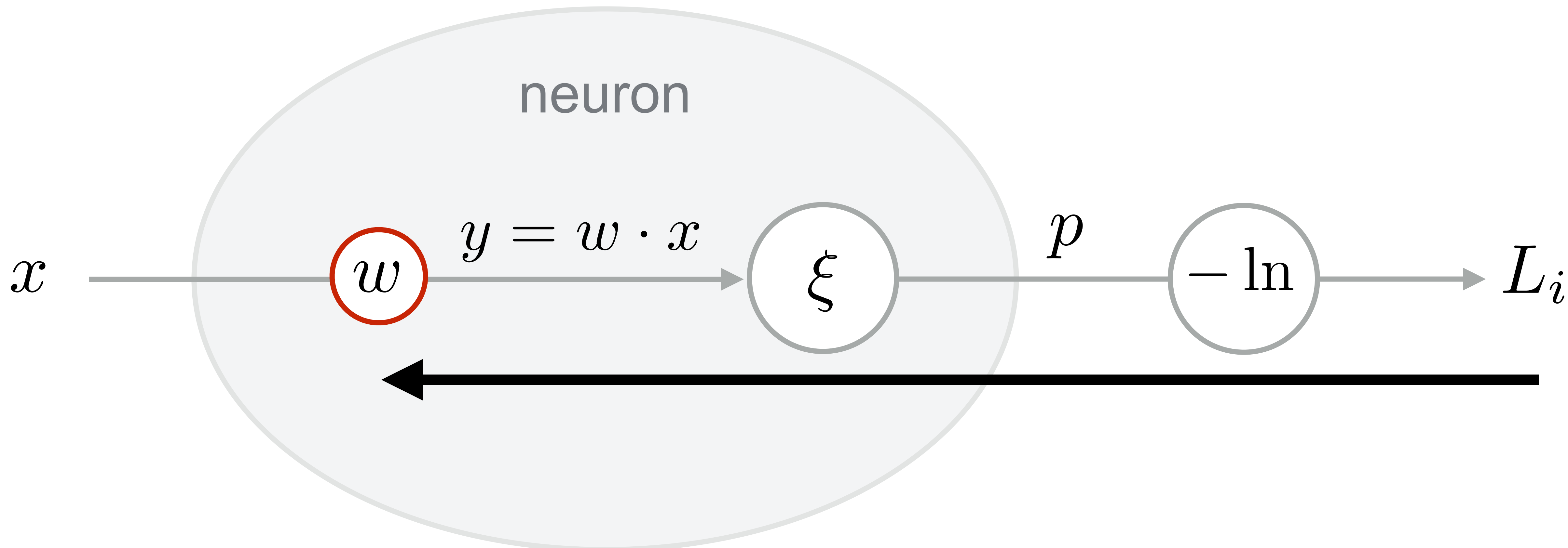
Gradient $\frac{\partial L_i}{\partial w}$

The Chain Rule



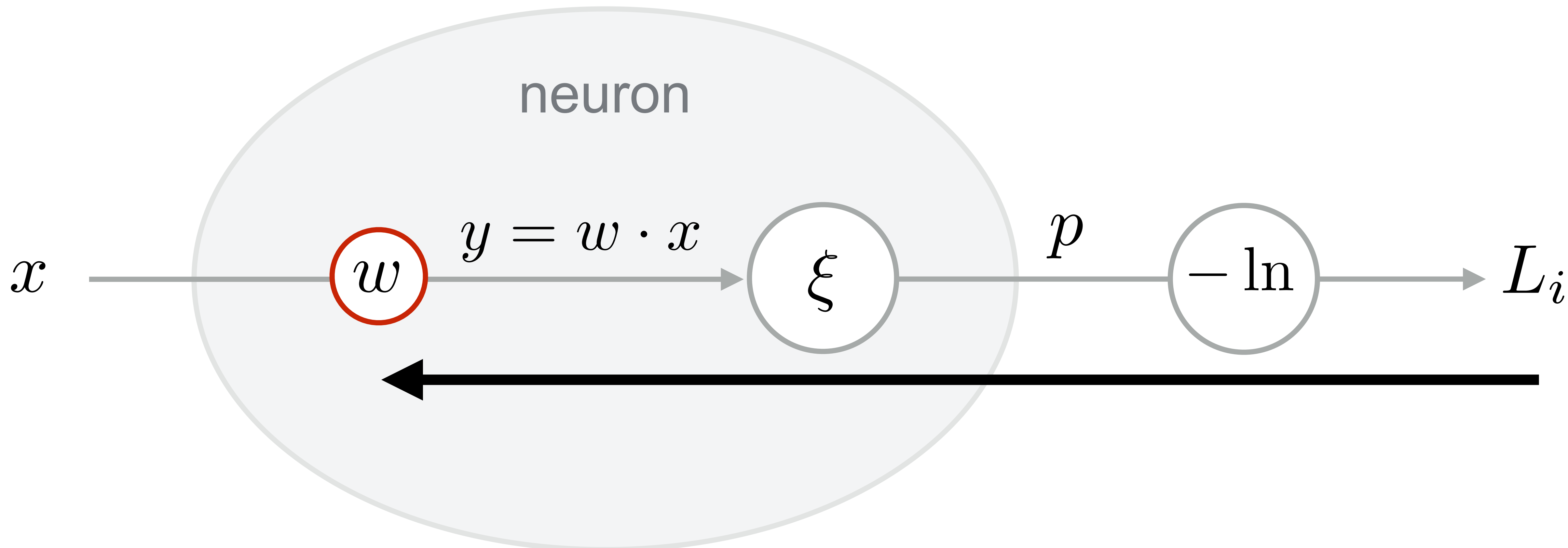
Gradient $\dfrac{\partial L_i}{\partial w}$

The Chain Rule



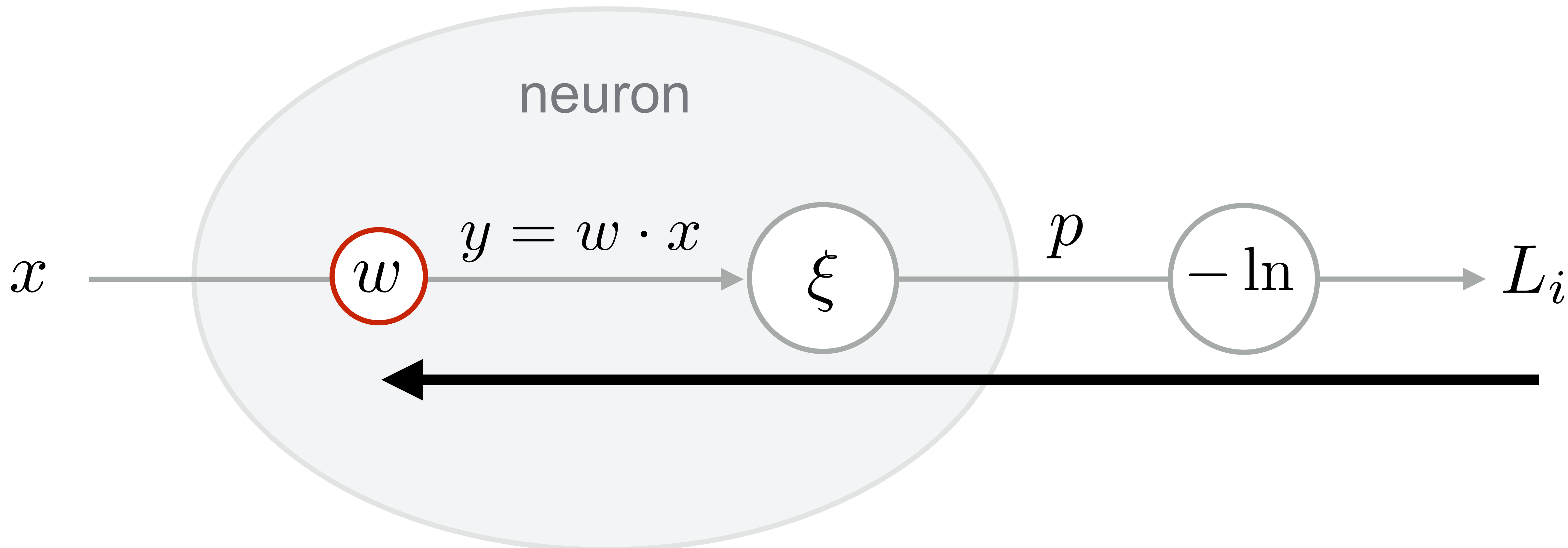
Gradient $\dfrac{\partial L_i}{\partial w} = \dfrac{\partial L_i}{\partial p}$

The Chain Rule



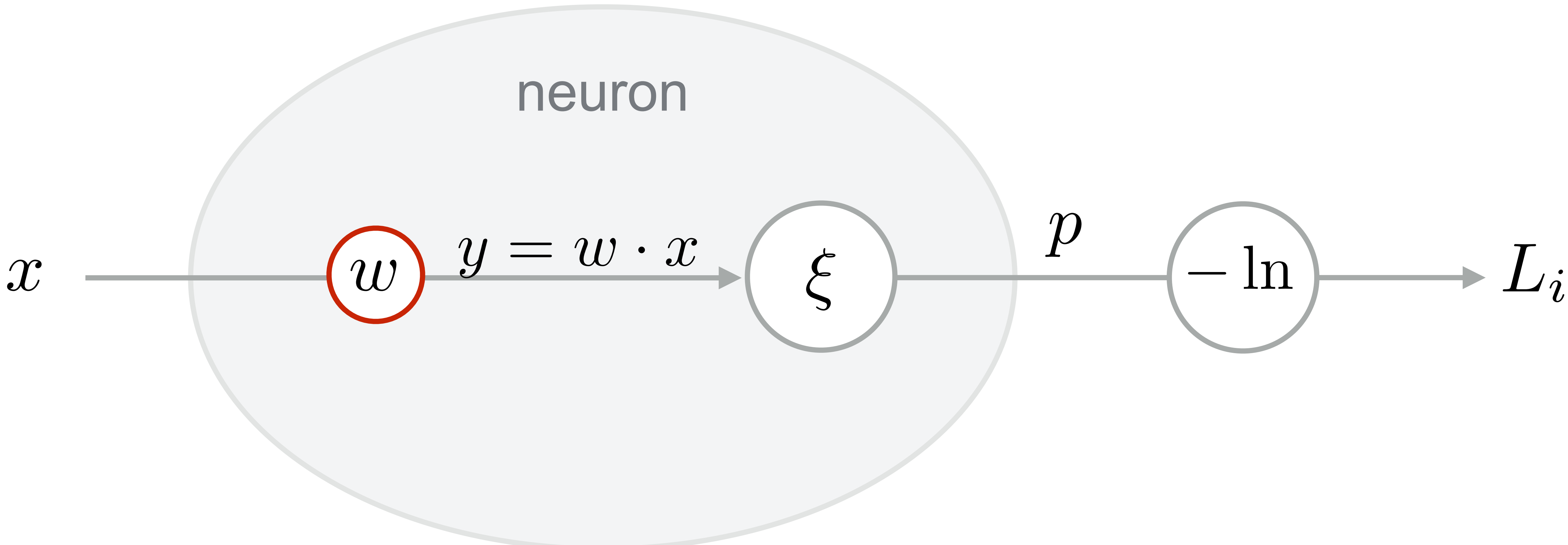
Gradient $\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p} \frac{\partial p}{\partial y}$

The Chain Rule

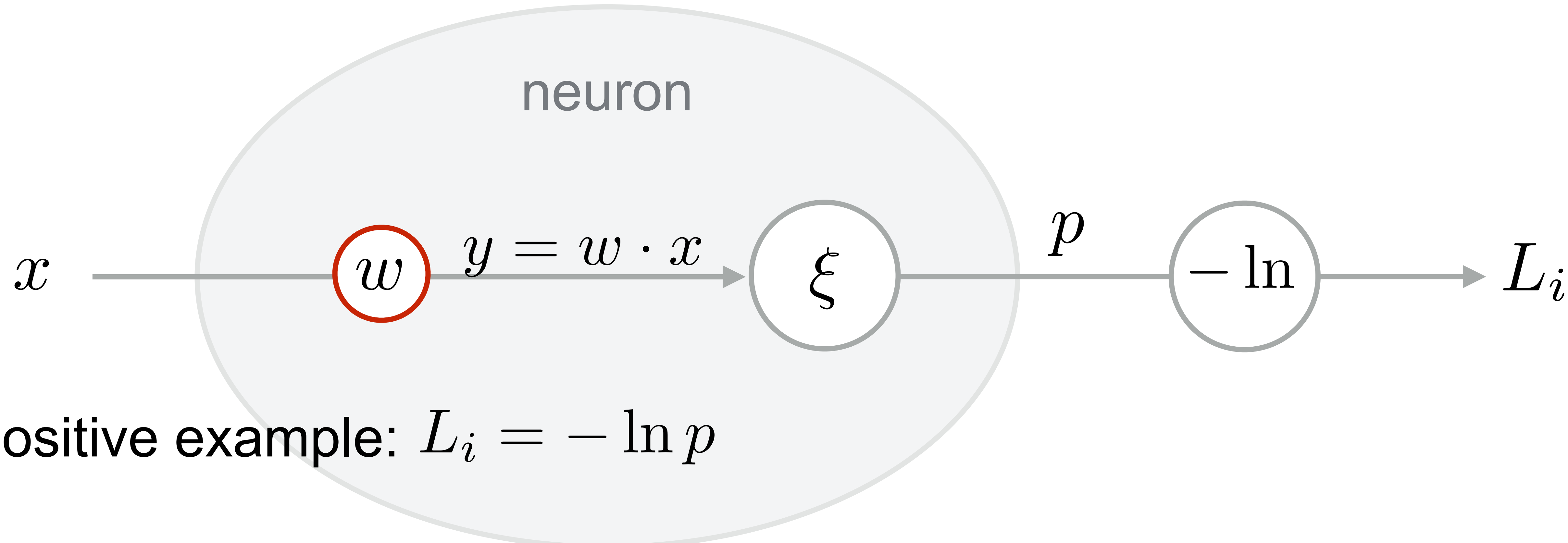


Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\frac{\partial p}{\partial y}\frac{\partial y}{\partial w}$$

The Chain Rule - Example

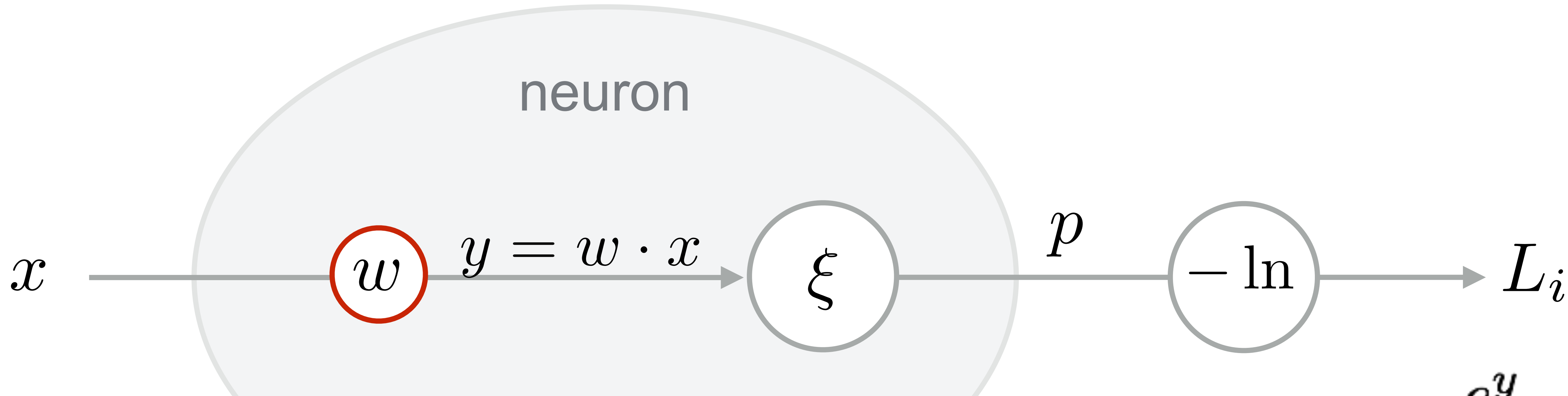


The Chain Rule - Example



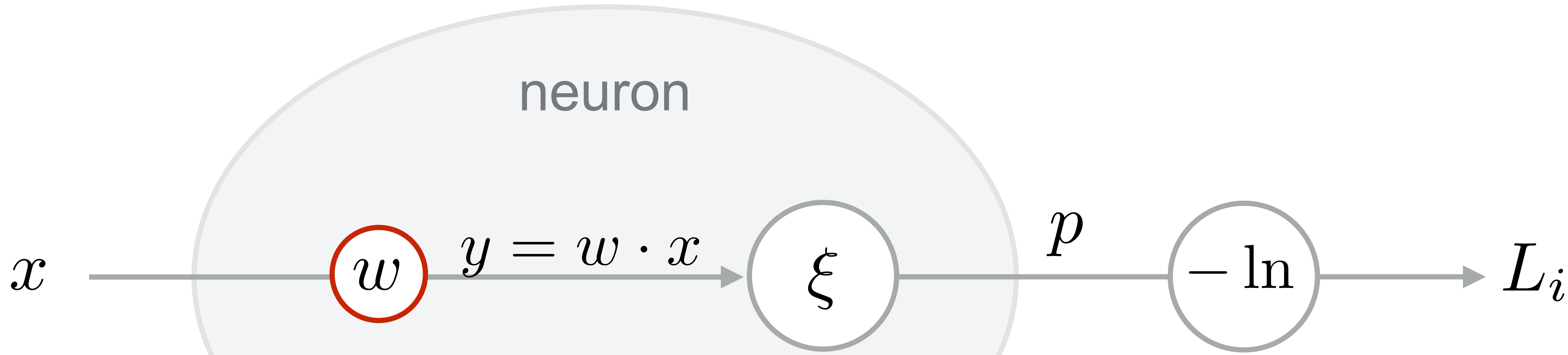
Positive example: $L_i = -\ln p$

The Chain Rule - Example



Positive example: $L_i = -\ln p$ logistic sigmoid $f(y) = \frac{e^y}{e^y + 1}$

The Chain Rule - Example

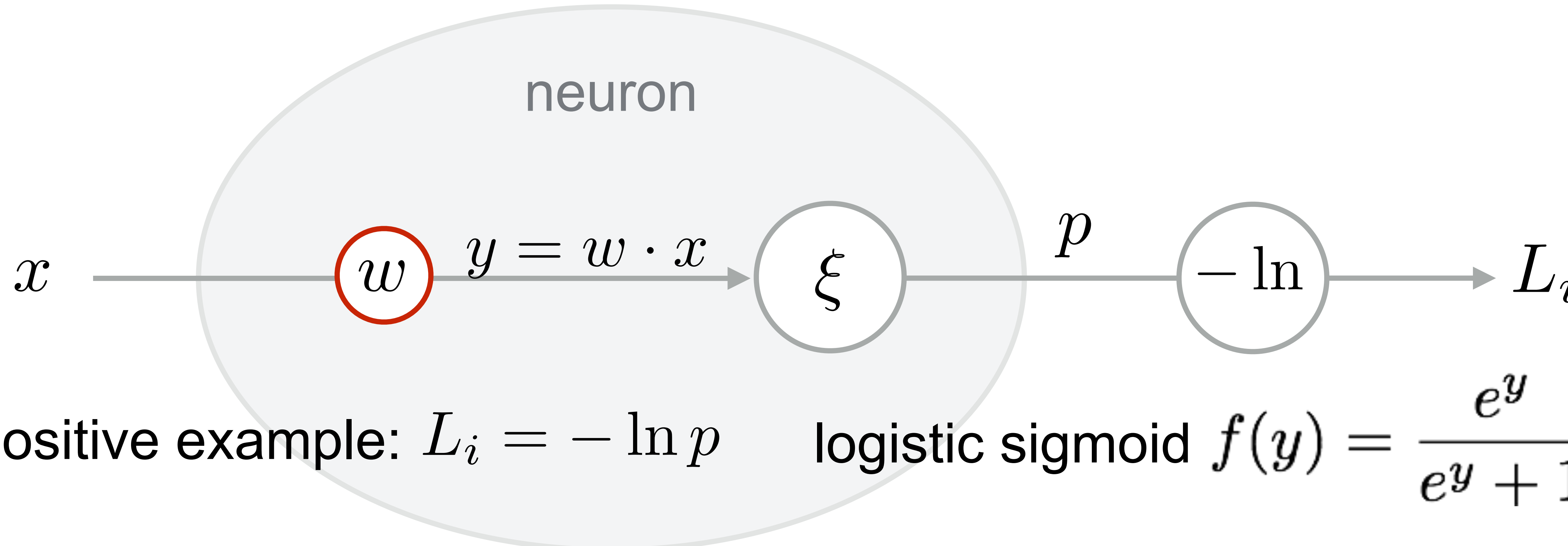


Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $\dfrac{\partial L_i}{\partial w} = \dfrac{\partial L_i}{\partial p} \dfrac{\partial p}{\partial y} \dfrac{\partial y}{\partial w}$

The Chain Rule - Example



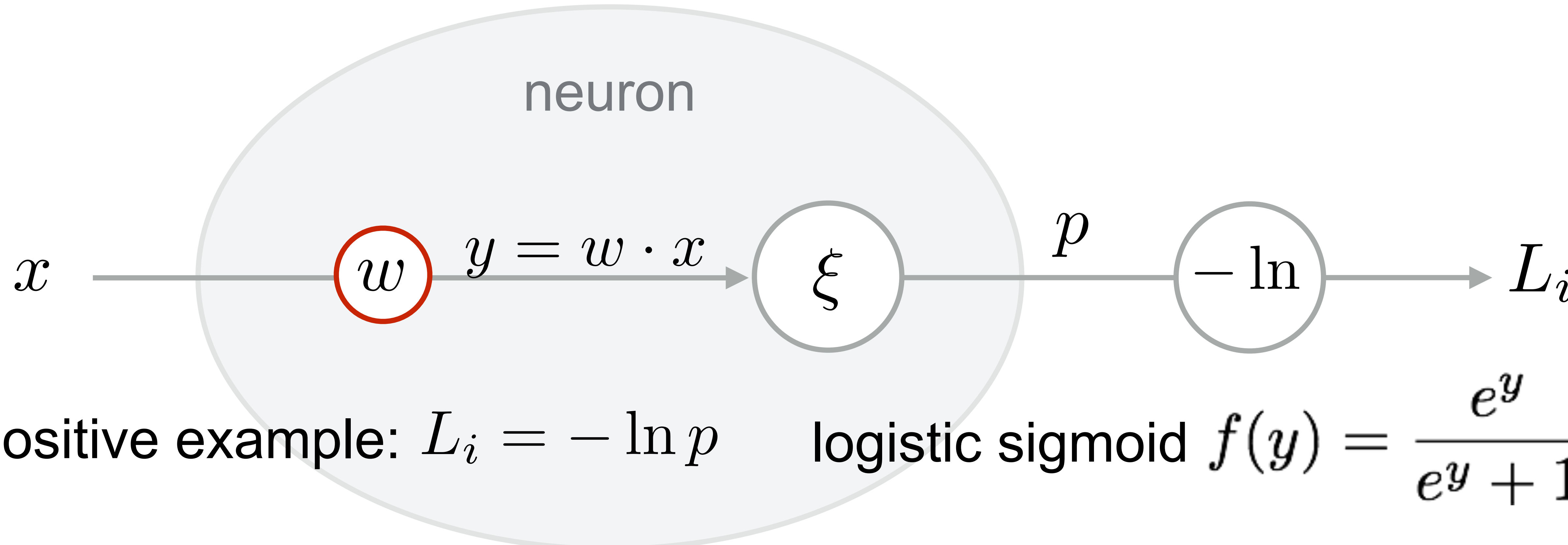
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \underbrace{\frac{\partial L_i}{\partial p}} \frac{\partial p}{\partial y} \frac{\partial y}{\partial w}$$

$$\frac{\partial L_i}{\partial p} = \frac{\partial}{\partial p}(-\ln p)$$

The Chain Rule - Example



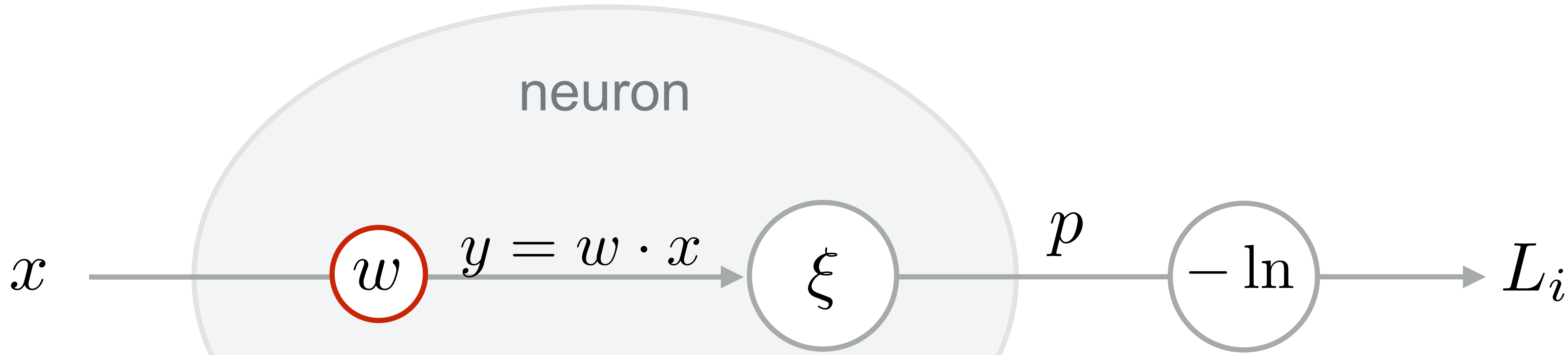
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \underbrace{\frac{\partial L_i}{\partial p}} \frac{\partial p}{\partial y} \frac{\partial y}{\partial w}$$

$$\frac{\partial L_i}{\partial p} = \frac{\partial}{\partial p}(-\ln p) = -\frac{1}{p}$$

The Chain Rule - Example



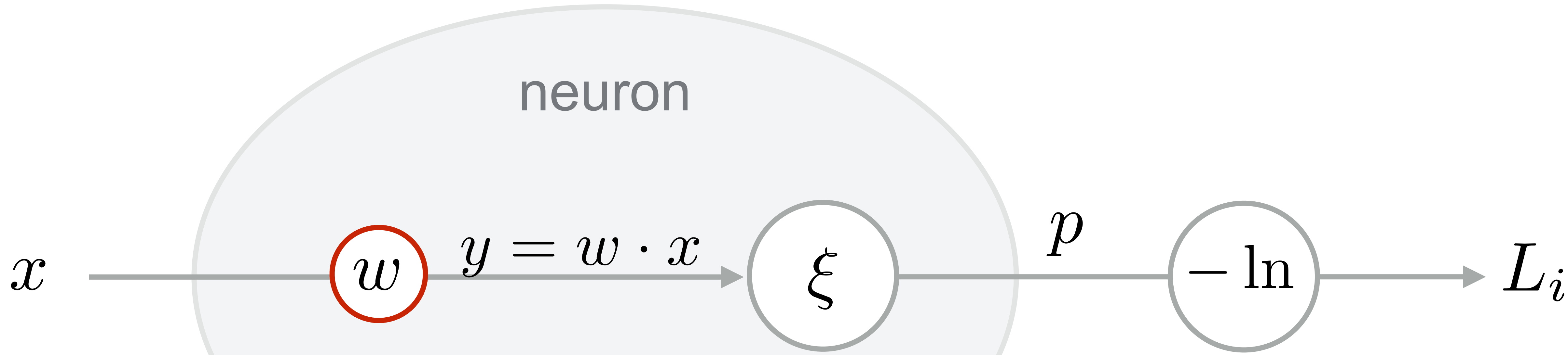
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \underbrace{\frac{\partial L_i}{\partial p}}\frac{\partial p}{\partial y}\frac{\partial y}{\partial w} = -\frac{1}{p}\frac{\partial p}{\partial y}\frac{\partial y}{\partial w}$$

$$\frac{\partial L_i}{\partial p} = \frac{\partial}{\partial p}(-\ln p) = -\frac{1}{p}$$

The Chain Rule - Example

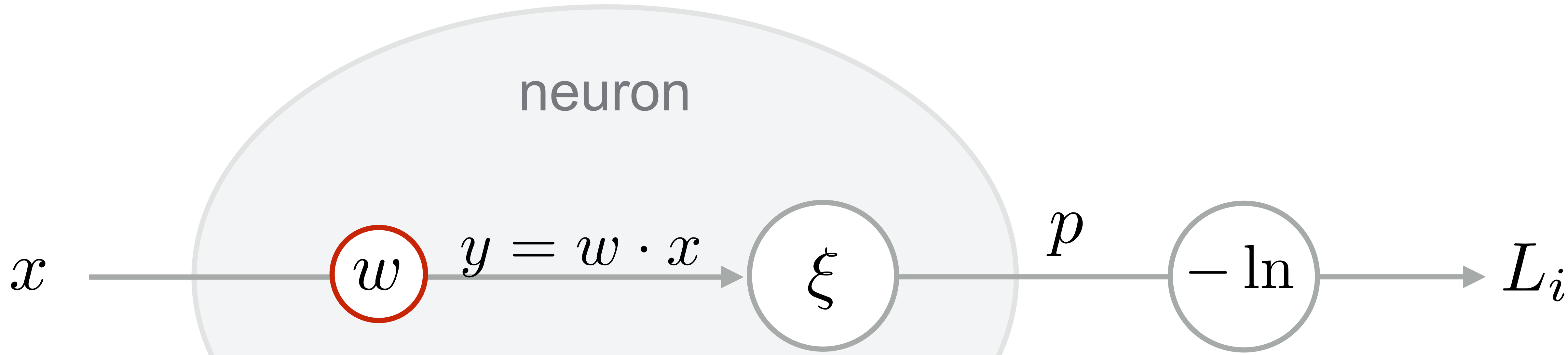


Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\frac{\partial p}{\partial y}\frac{\partial y}{\partial w} = -\frac{1}{p}\frac{\partial p}{\partial y}\frac{\partial y}{\partial w}$$

The Chain Rule - Example

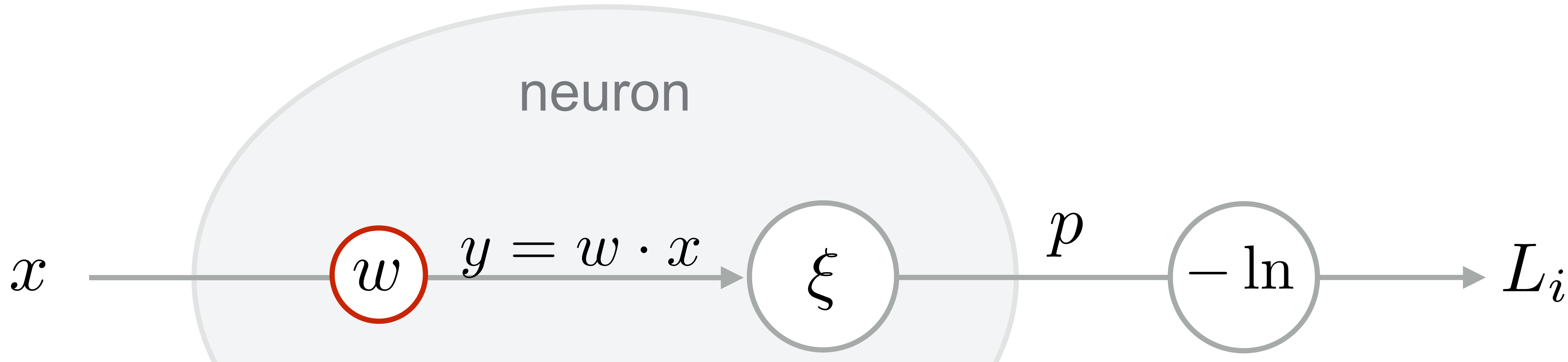


Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p} \underbrace{\frac{\partial p}{\partial y}} \frac{\partial y}{\partial w} = -\frac{1}{p} \frac{\partial p}{\partial y} \frac{\partial y}{\partial w}$$

The Chain Rule - Example



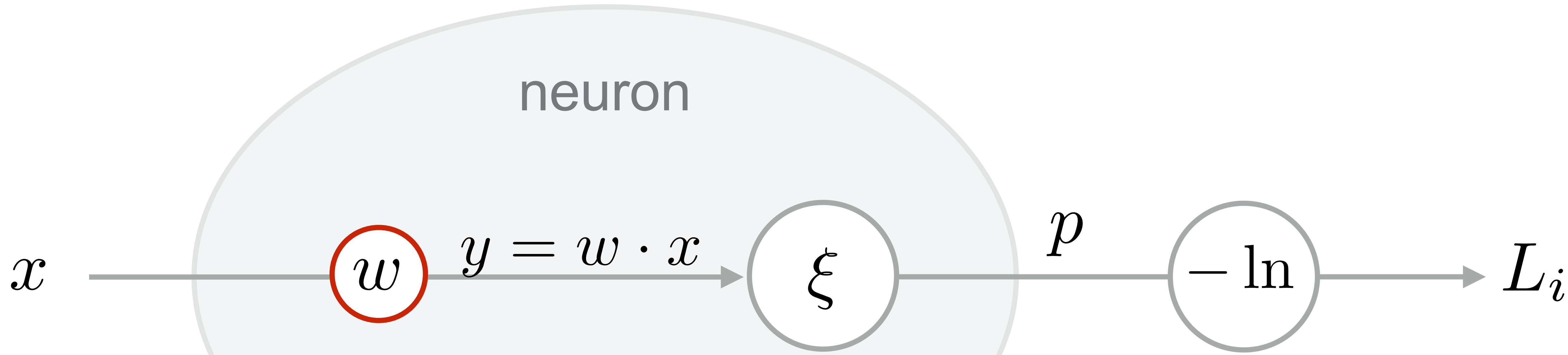
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p} \underbrace{\frac{\partial p}{\partial y}}_{} \frac{\partial y}{\partial w} = -\frac{1}{p} \frac{\partial p}{\partial y} \frac{\partial y}{\partial w}$$

$$\frac{\partial p}{\partial y}$$

The Chain Rule - Example

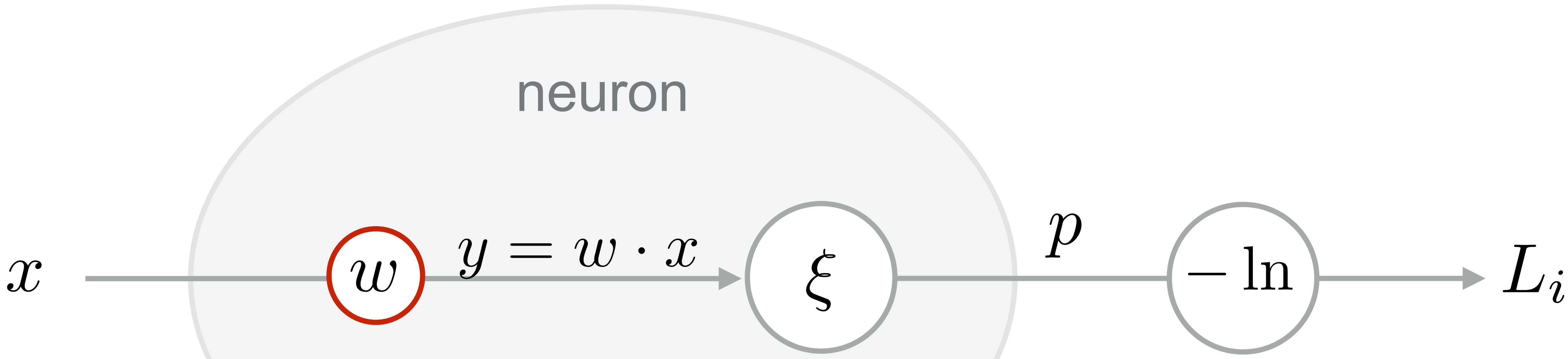


Positive example: $L_i = -\ln p$ logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p} \underbrace{\frac{\partial p}{\partial y}} \frac{\partial y}{\partial w} = -\frac{1}{p} \frac{\partial p}{\partial y} \frac{\partial y}{\partial w}$$

$$\frac{\partial p}{\partial y} = \frac{\partial}{\partial y} \frac{e^y}{e^y + 1}$$

The Chain Rule - Example



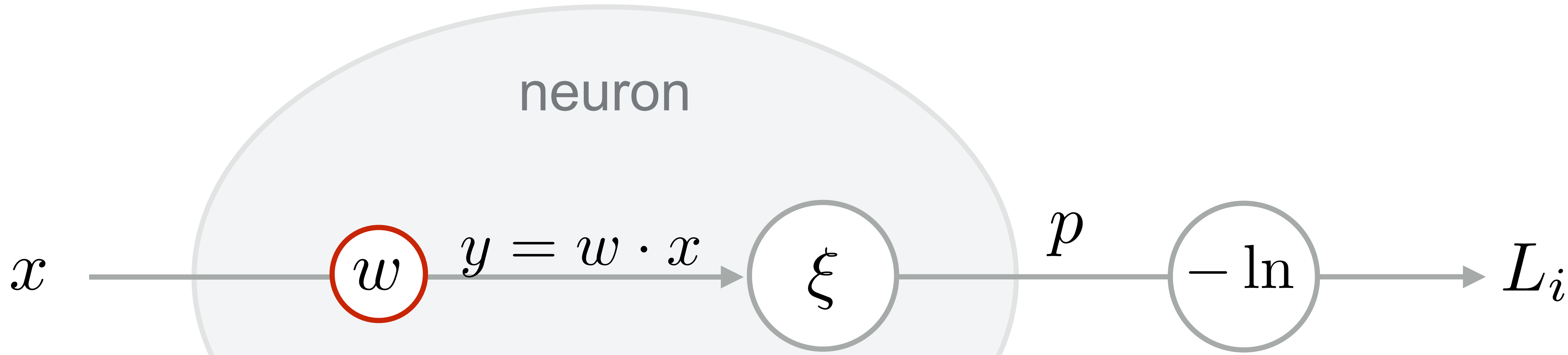
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p} \underbrace{\frac{\partial p}{\partial y}} \frac{\partial y}{\partial w} = -\frac{1}{p} \frac{\partial p}{\partial y} \frac{\partial y}{\partial w}$$

$$\frac{\partial p}{\partial y} = \frac{\partial}{\partial y} \frac{e^y}{e^y + 1} = \frac{e^y(1 + e^y) - (e^y)^2}{(e^y + 1)^2}$$

The Chain Rule - Example



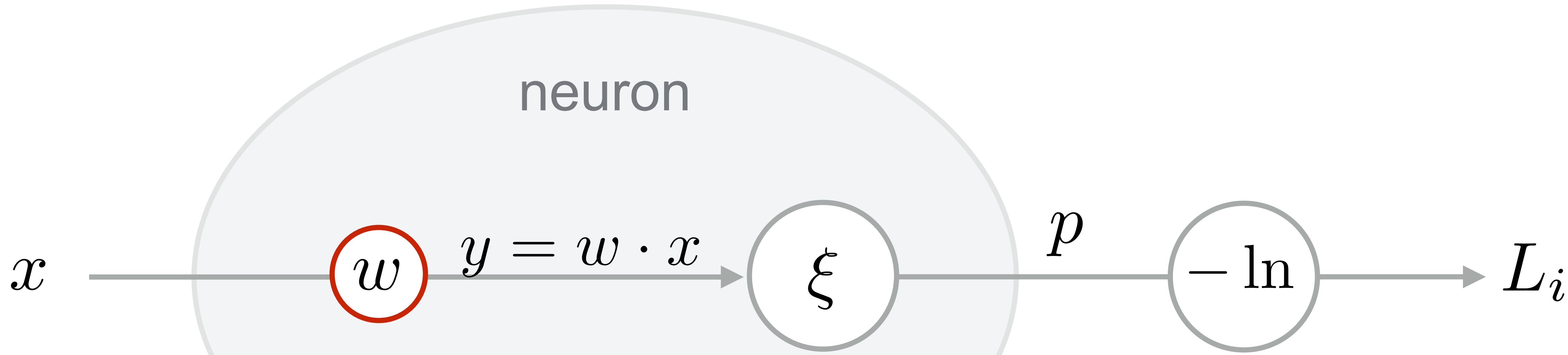
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\frac{\partial p}{\partial y}\frac{\partial y}{\partial w} = -\frac{1}{p}\frac{\partial p}{\partial y}\frac{\partial y}{\partial w}$$

$$\frac{\partial p}{\partial y} = \frac{\partial}{\partial y}\frac{e^y}{e^y + 1} = \frac{e^y(1 + e^y) - (e^y)^2}{(e^y + 1)^2} = \frac{e^y}{(e^y + 1)^2}$$

The Chain Rule - Example



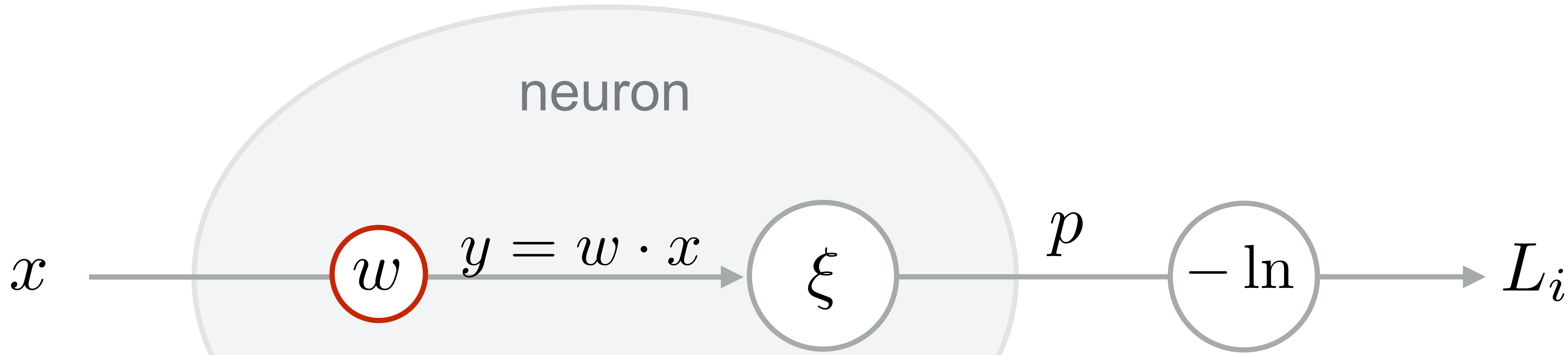
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $\dfrac{\partial L_i}{\partial w} = \dfrac{\partial L_i}{\partial p} \underbrace{\dfrac{\partial p}{\partial y}} \dfrac{\partial y}{\partial w} = -\dfrac{1}{p} \dfrac{\partial p}{\partial y} \dfrac{\partial y}{\partial w}$

$$\frac{\partial p}{\partial y} = \frac{\partial}{\partial y} \frac{e^y}{e^y + 1} = \frac{e^y(1 + e^y) - (e^y)^2}{(e^y + 1)^2} = \frac{e^y}{(e^y + 1)^2} = p(1 - p)$$

The Chain Rule - Example



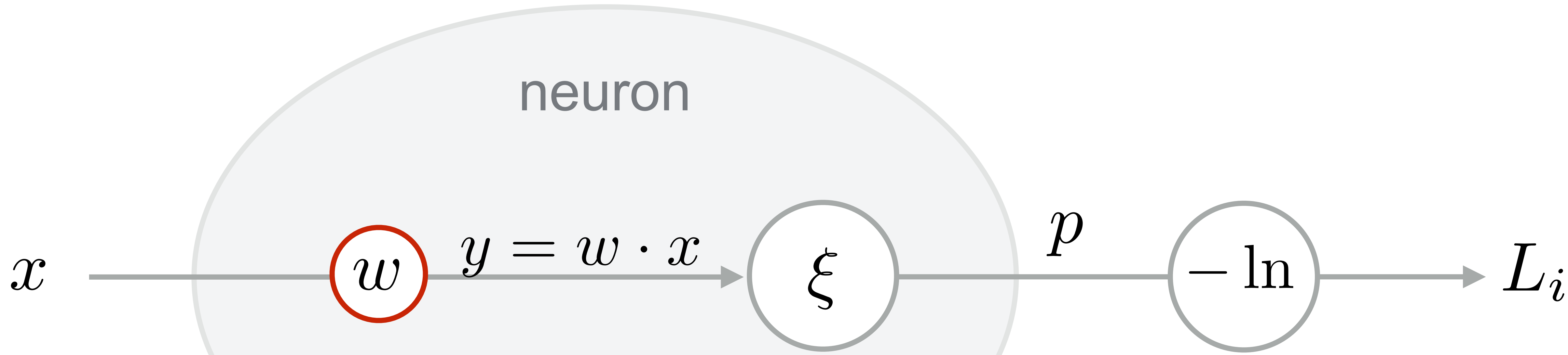
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\underbrace{\frac{\partial p}{\partial y}}\frac{\partial y}{\partial w} = -\frac{1}{p}p(1-p)\frac{\partial y}{\partial w}$$

$$\frac{\partial p}{\partial y} = \frac{\partial}{\partial y}\frac{e^y}{e^y + 1} = \frac{e^y(1+e^y) - (e^y)^2}{(e^y+1)^2} = \frac{e^y}{(e^y+1)^2} = p(1-p)$$

The Chain Rule - Example



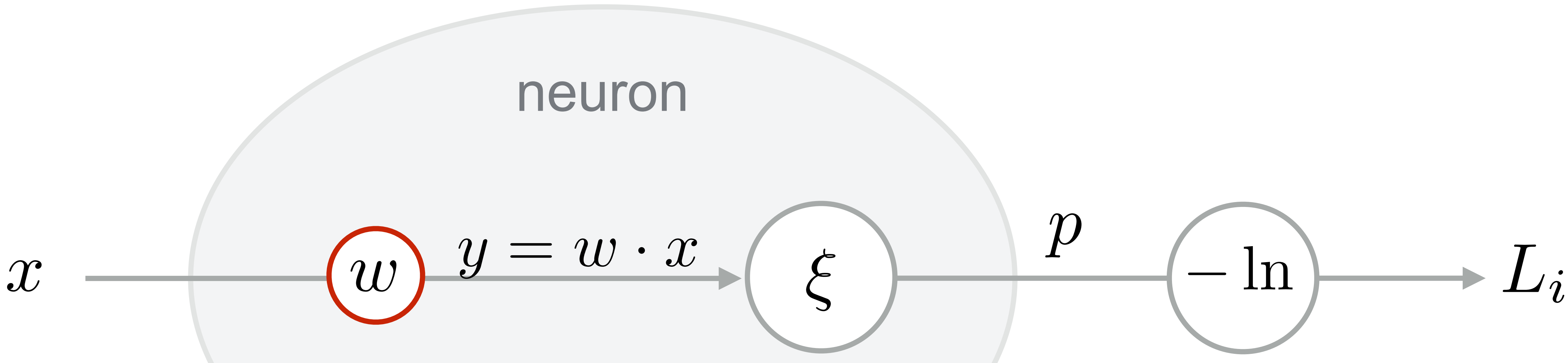
Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $\dfrac{\partial L_i}{\partial w} = \dfrac{\partial L_i}{\partial p} \underbrace{\dfrac{\partial p}{\partial y}} \dfrac{\partial y}{\partial w} = -\dfrac{1}{p} p(1-p) \dfrac{\partial y}{\partial w} = (p-1)\dfrac{\partial y}{\partial w}$

$$\frac{\partial p}{\partial y} = \frac{\partial}{\partial y} \frac{e^y}{e^y + 1} = \frac{e^y(1+e^y) - (e^y)^2}{(e^y+1)^2} = \frac{e^y}{(e^y+1)^2} = p(1-p)$$

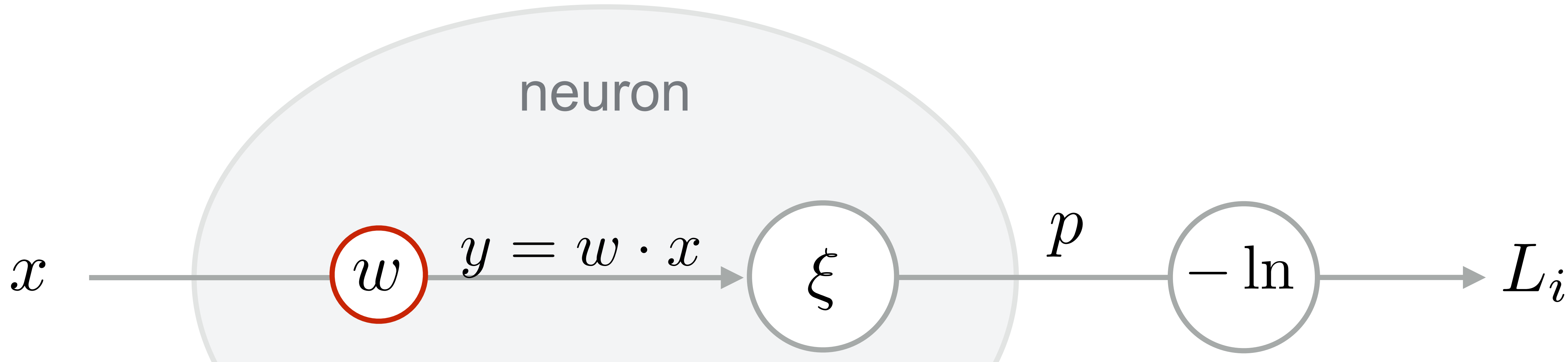
The Chain Rule - Example



Positive example: $L_i = -\ln p$ logistic sigmoid $f(y) = \frac{e^y}{e^y + 1}$

Gradient $\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p} \frac{\partial p}{\partial y} \frac{\partial y}{\partial w} = (p-1)\frac{\partial y}{\partial w}$

The Chain Rule - Example

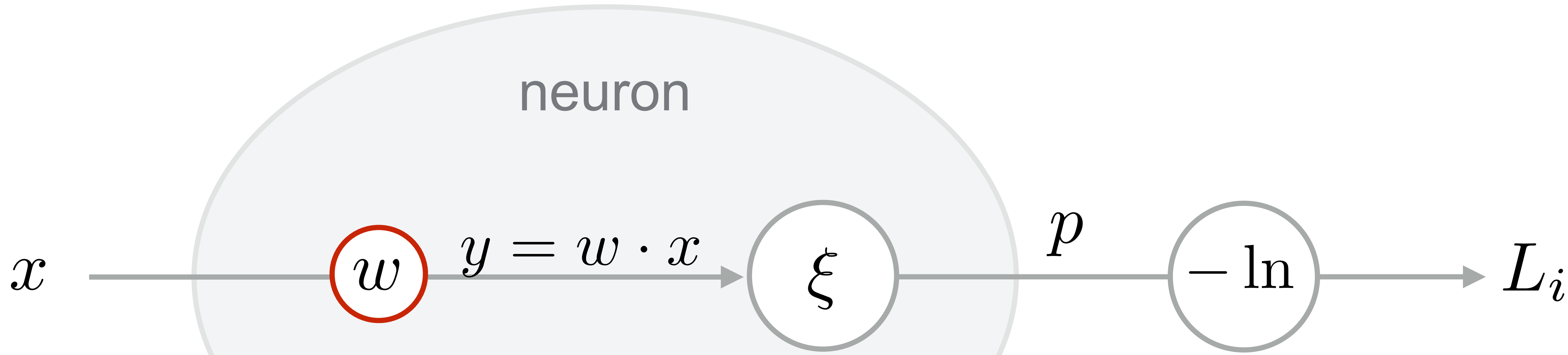


Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $\dfrac{\partial L_i}{\partial w} = \dfrac{\partial L_i}{\partial p} \dfrac{\partial p}{\partial y} \underbrace{\dfrac{\partial y}{\partial w}}_{} = (p-1) \dfrac{\partial y}{\partial w}$

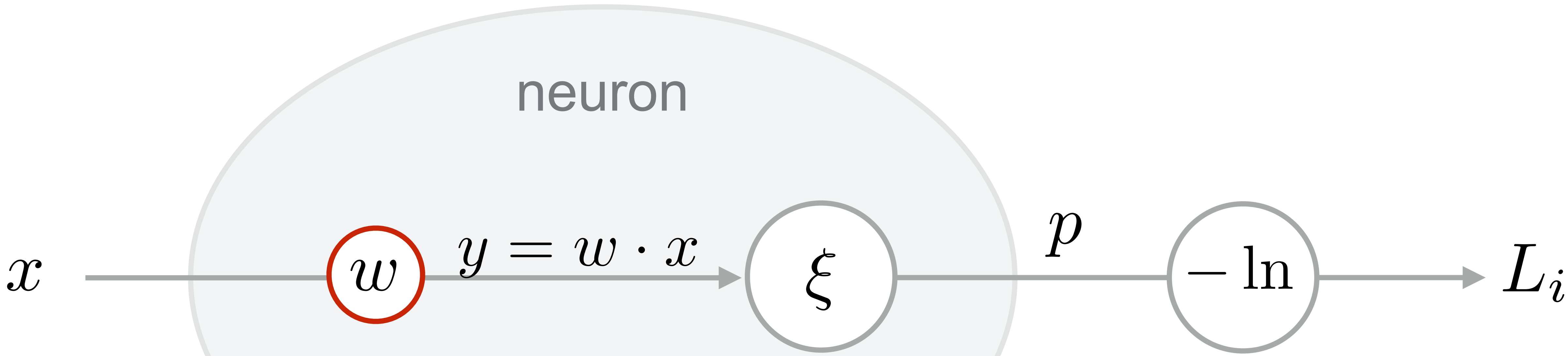
The Chain Rule - Example



Positive example: $L_i = -\ln p$ logistic sigmoid $f(y) = \frac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\frac{\partial p}{\partial y}\underbrace{\frac{\partial y}{\partial w}}_{\frac{\partial y}{\partial w}} = (p-1)\frac{\partial y}{\partial w}$$

The Chain Rule - Example

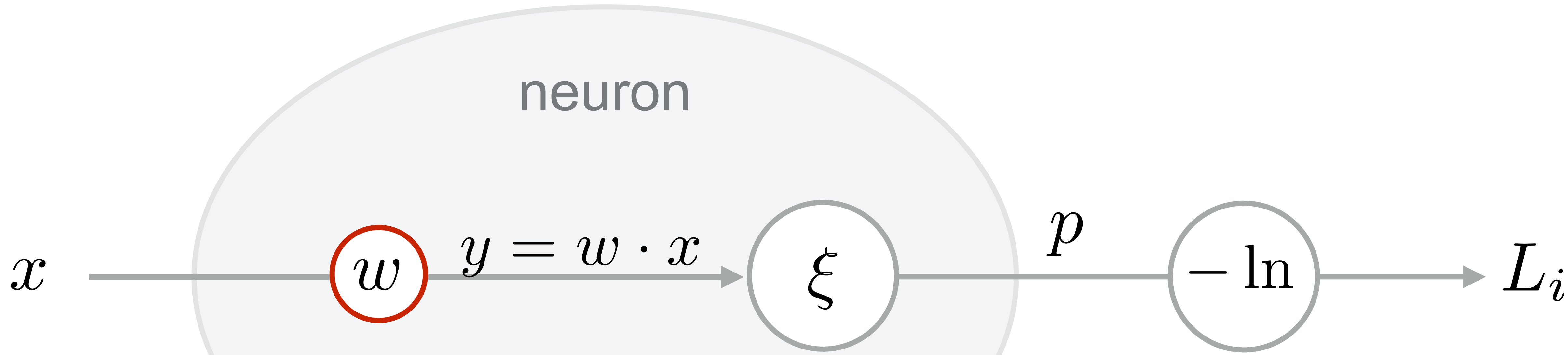


Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\frac{\partial p}{\partial y}\underbrace{\frac{\partial y}{\partial w}}_{\frac{\partial y}{\partial w} = \frac{\partial wx}{\partial w}} = (p-1)\frac{\partial y}{\partial w}$$

The Chain Rule - Example

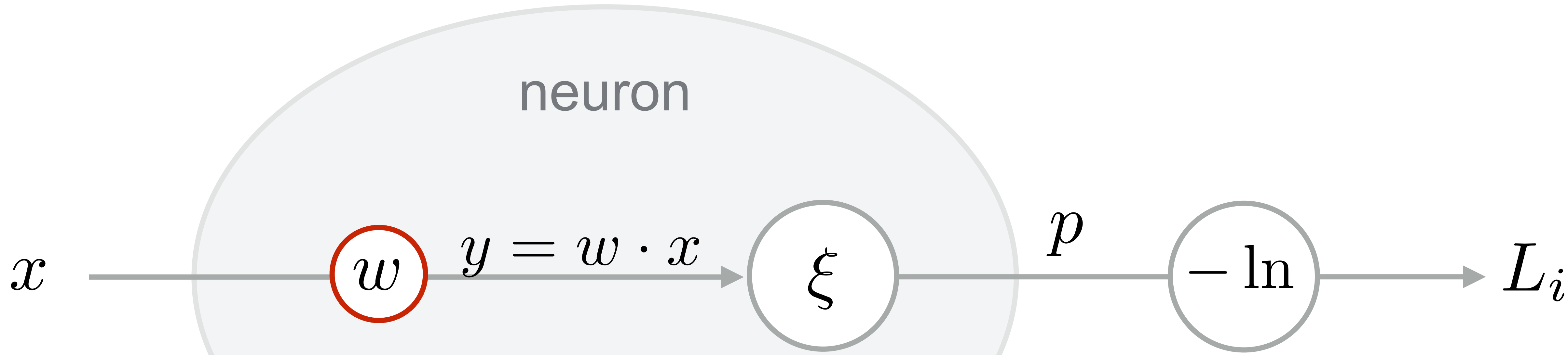


Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\frac{\partial p}{\partial y}\underbrace{\frac{\partial y}{\partial w}}_{\frac{\partial y}{\partial w} = \frac{\partial wx}{\partial w} = x} = (p-1)\frac{\partial y}{\partial w}$$

The Chain Rule - Example

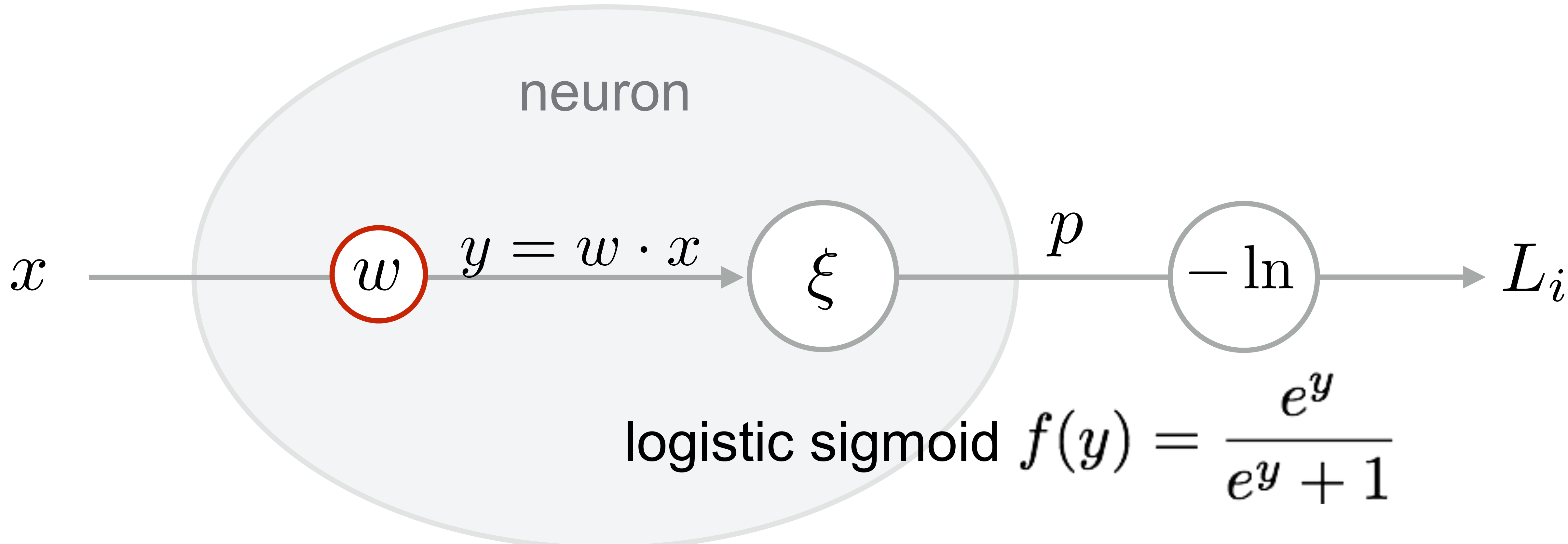


Positive example: $L_i = -\ln p$

logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient $$\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial p}\frac{\partial p}{\partial y}\underbrace{\frac{\partial y}{\partial w}}_{\frac{\partial y}{\partial w} = \frac{\partial wx}{\partial w} = x} = (p-1)\frac{\partial y}{\partial w} = (p-1)x$$

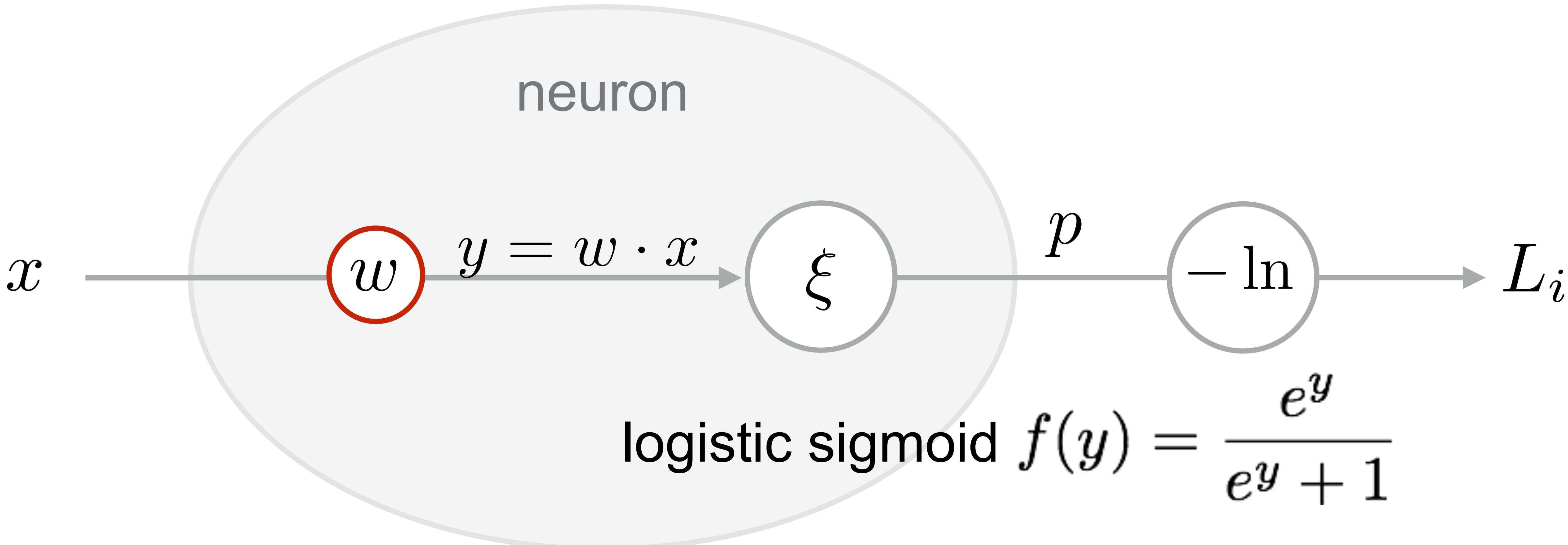
The Chain Rule - Example



logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

Gradient for positive example: $\dfrac{\partial L_i}{\partial w} = (p-1)x$

The Chain Rule - Example

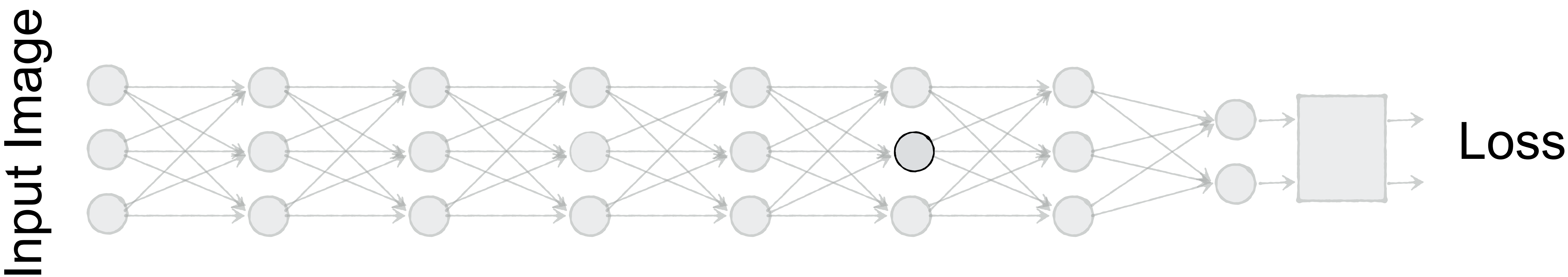


logistic sigmoid $f(y) = \dfrac{e^y}{e^y + 1}$

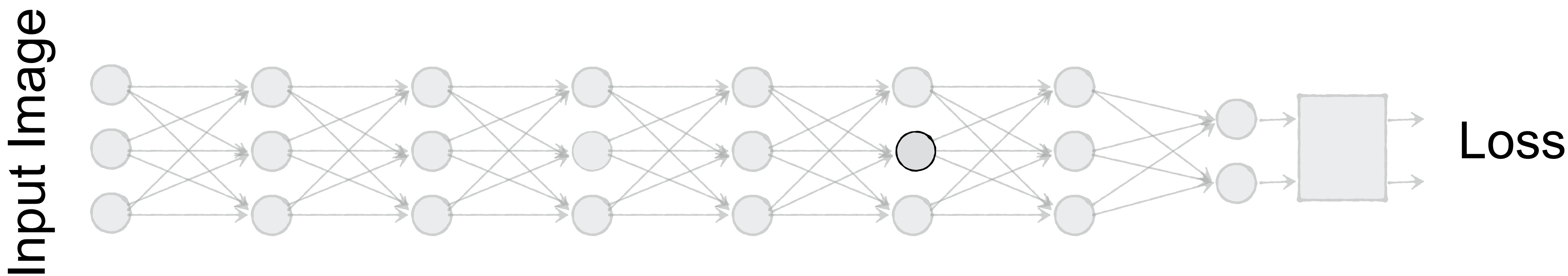
Gradient for positive example: $\dfrac{\partial L_i}{\partial w} = (p-1)x$

Gradient for negative example: $\dfrac{\partial L_i}{\partial w} = px$

Network Structure

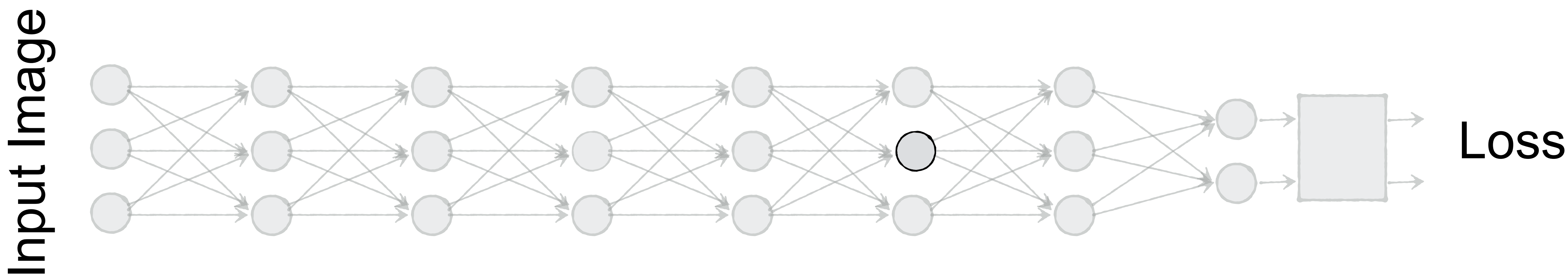


Network Structure



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

Network Structure

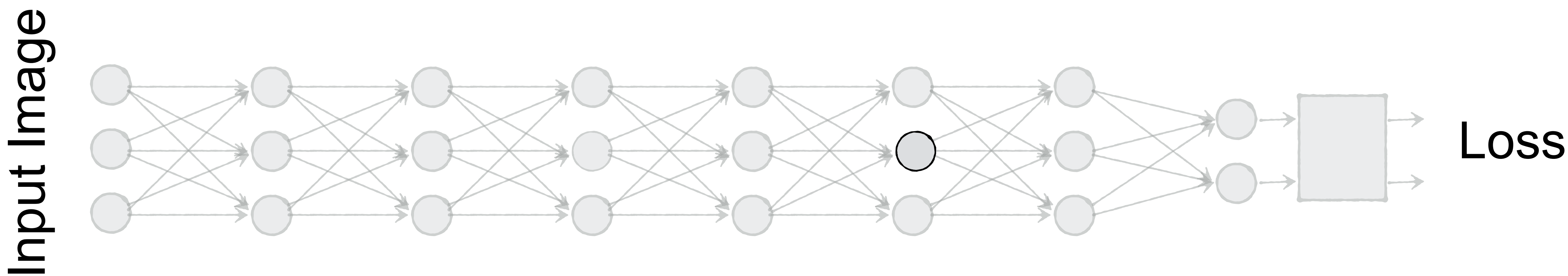


$$\theta^{(k+1)} = \theta^{(k)} - \mu \nabla L_i(\theta)$$

$$\begin{pmatrix} \vdots \\ w_k \\ \vdots \end{pmatrix}$$

parameters

Network Structure

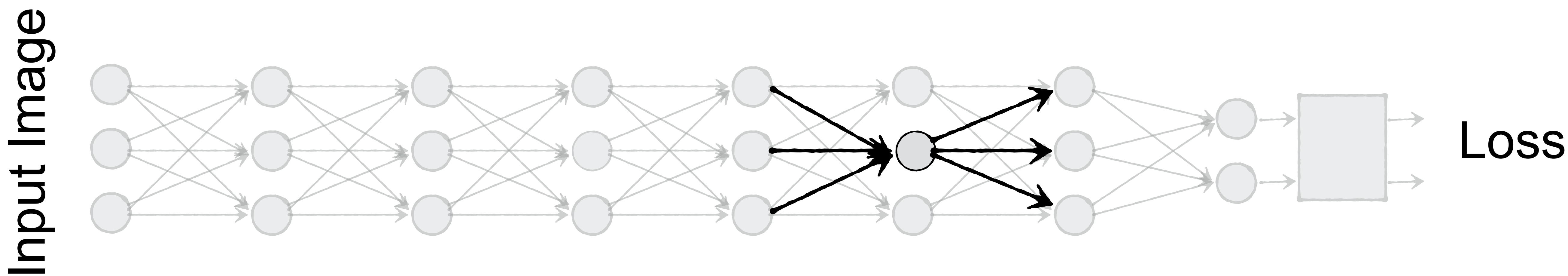


$$\theta^{(k+1)} = \theta^{(k)} - \mu \nabla L_i(\theta)$$

$$\begin{pmatrix} \vdots \\ w_k \\ \vdots \end{pmatrix} \qquad \begin{pmatrix} \vdots \\ \frac{\partial L_i}{\partial w_k} \\ \vdots \end{pmatrix}$$

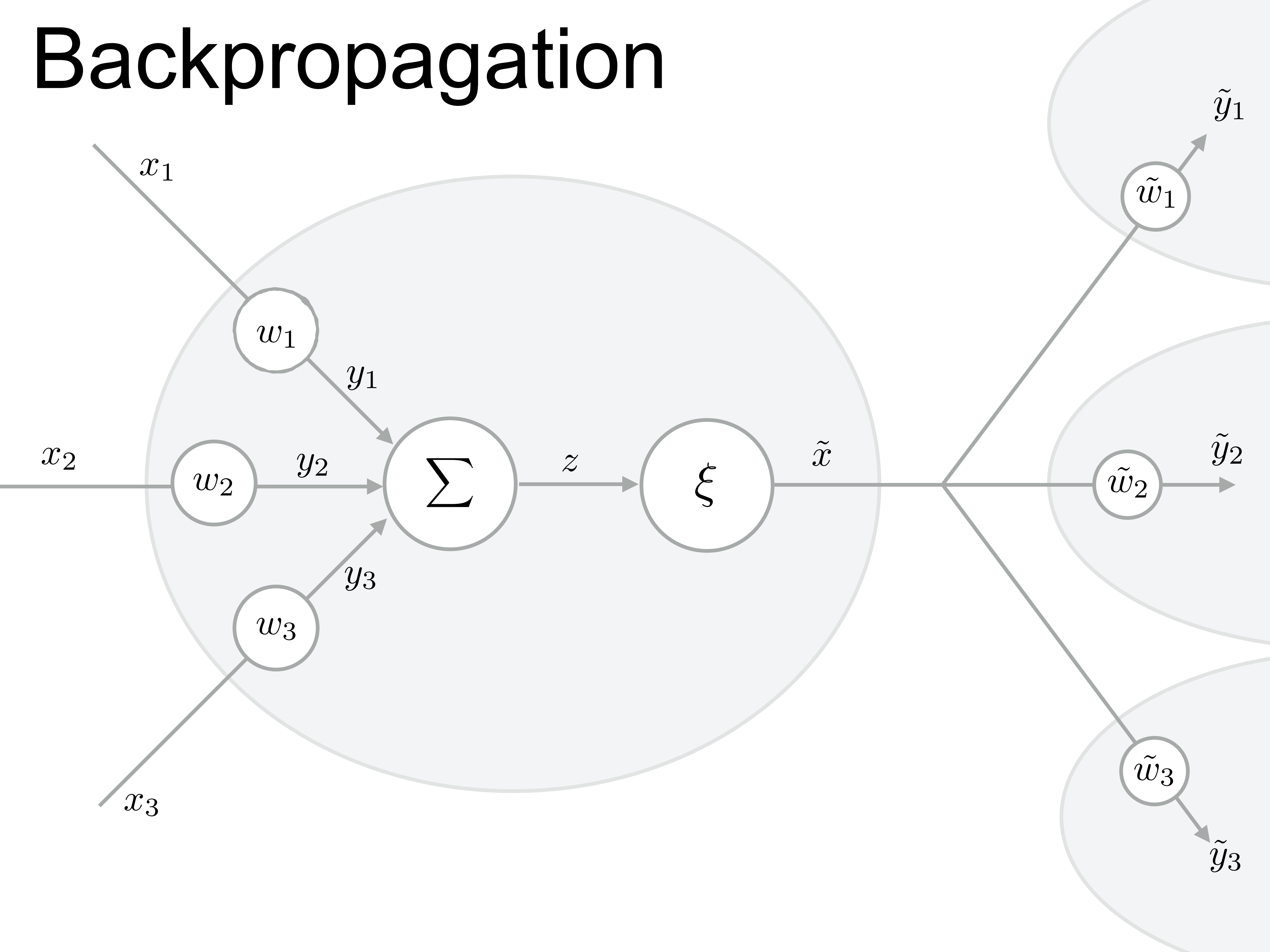
parameters gradient

Network Structure

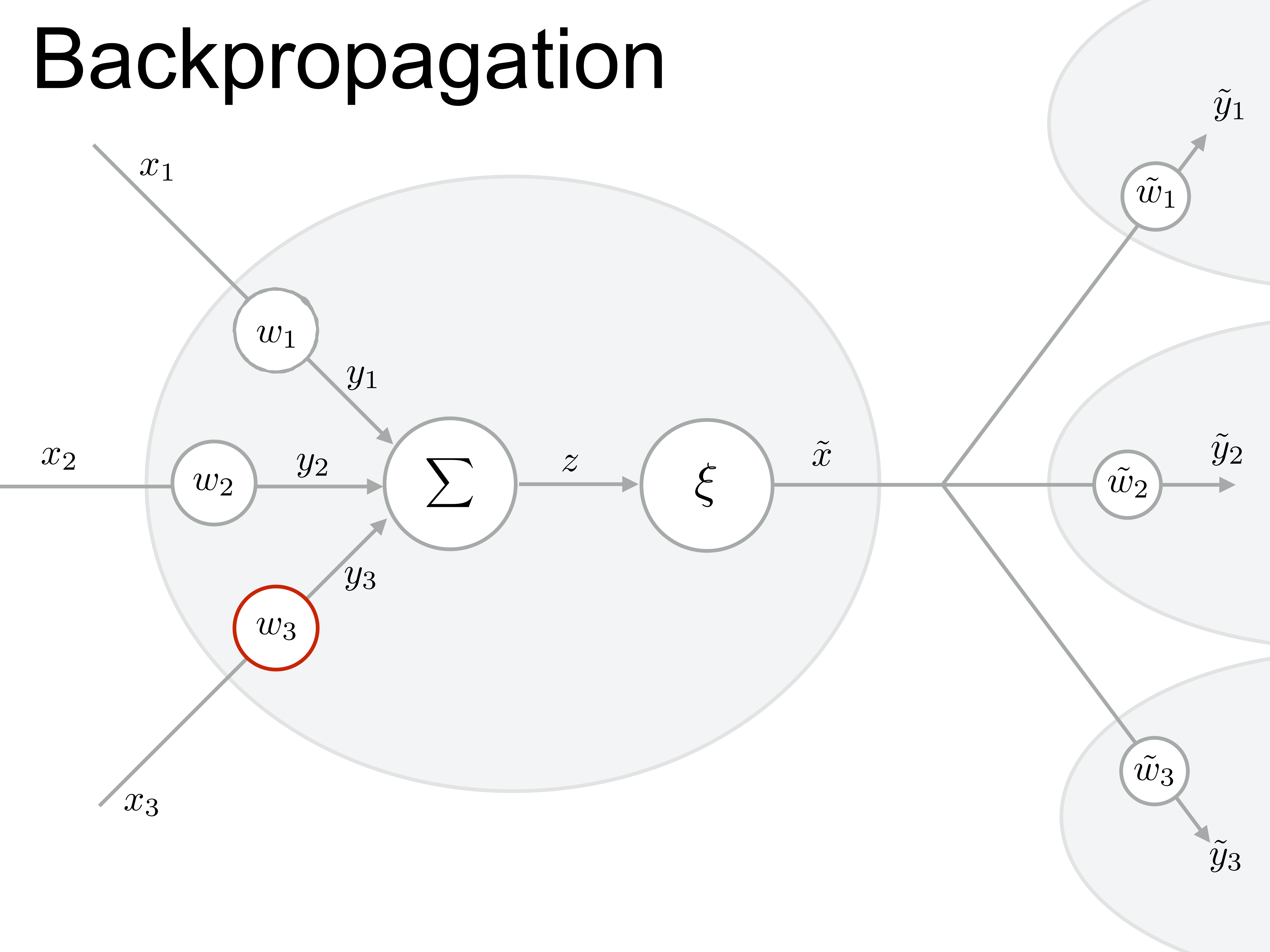


Common to have multiple inputs and outputs per neuron

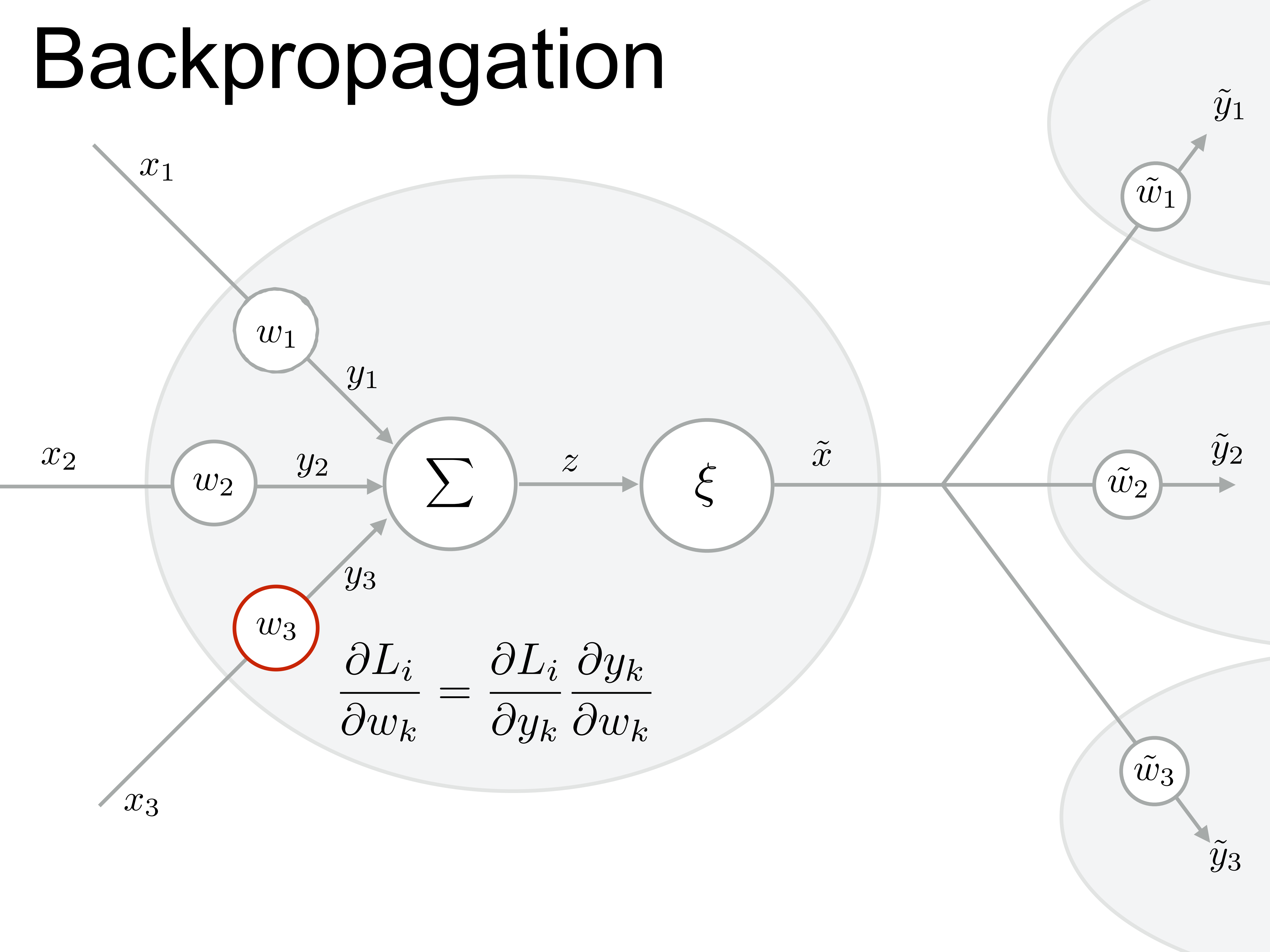
Backpropagation



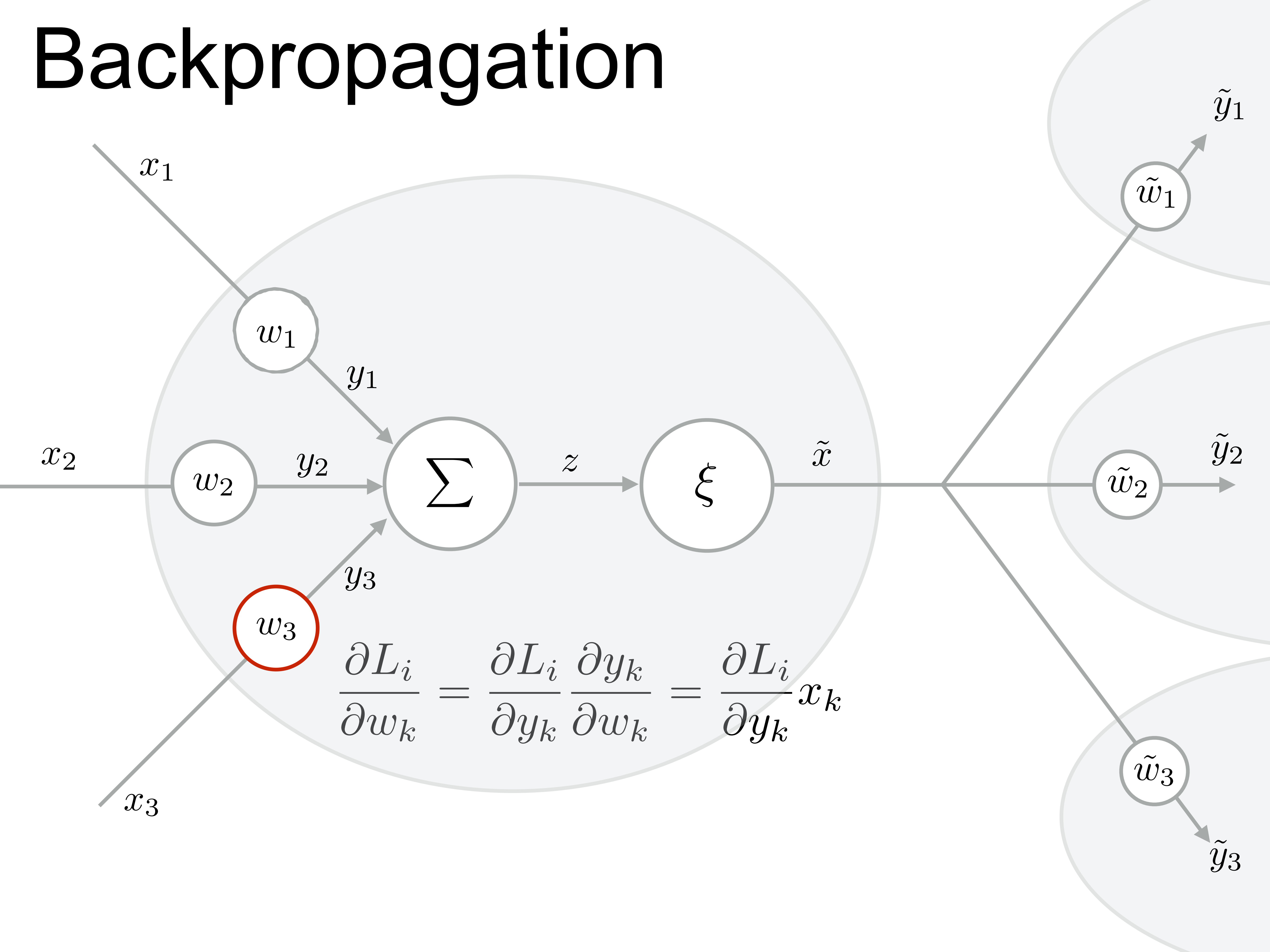
Backpropagation



Backpropagation

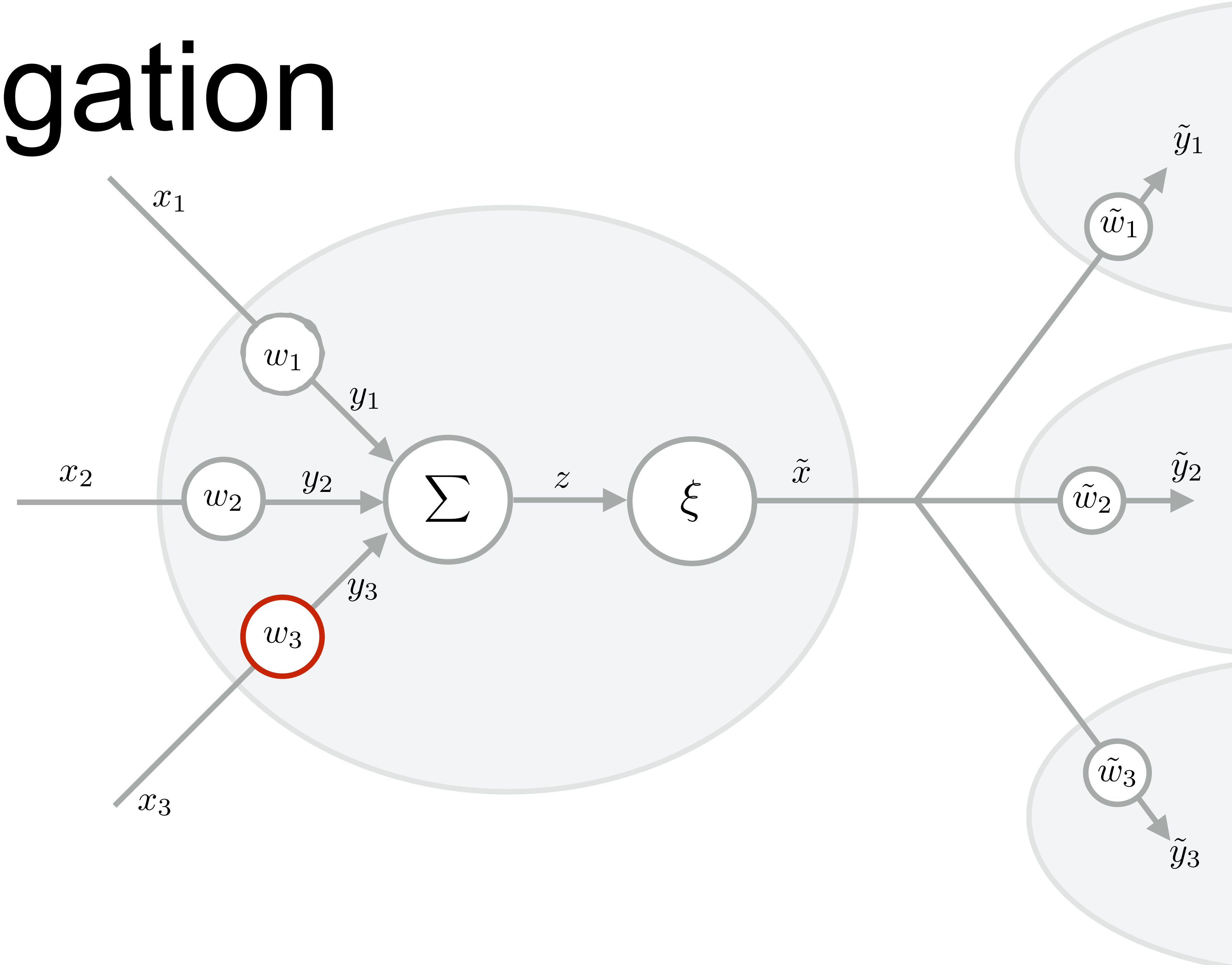


Backpropagation



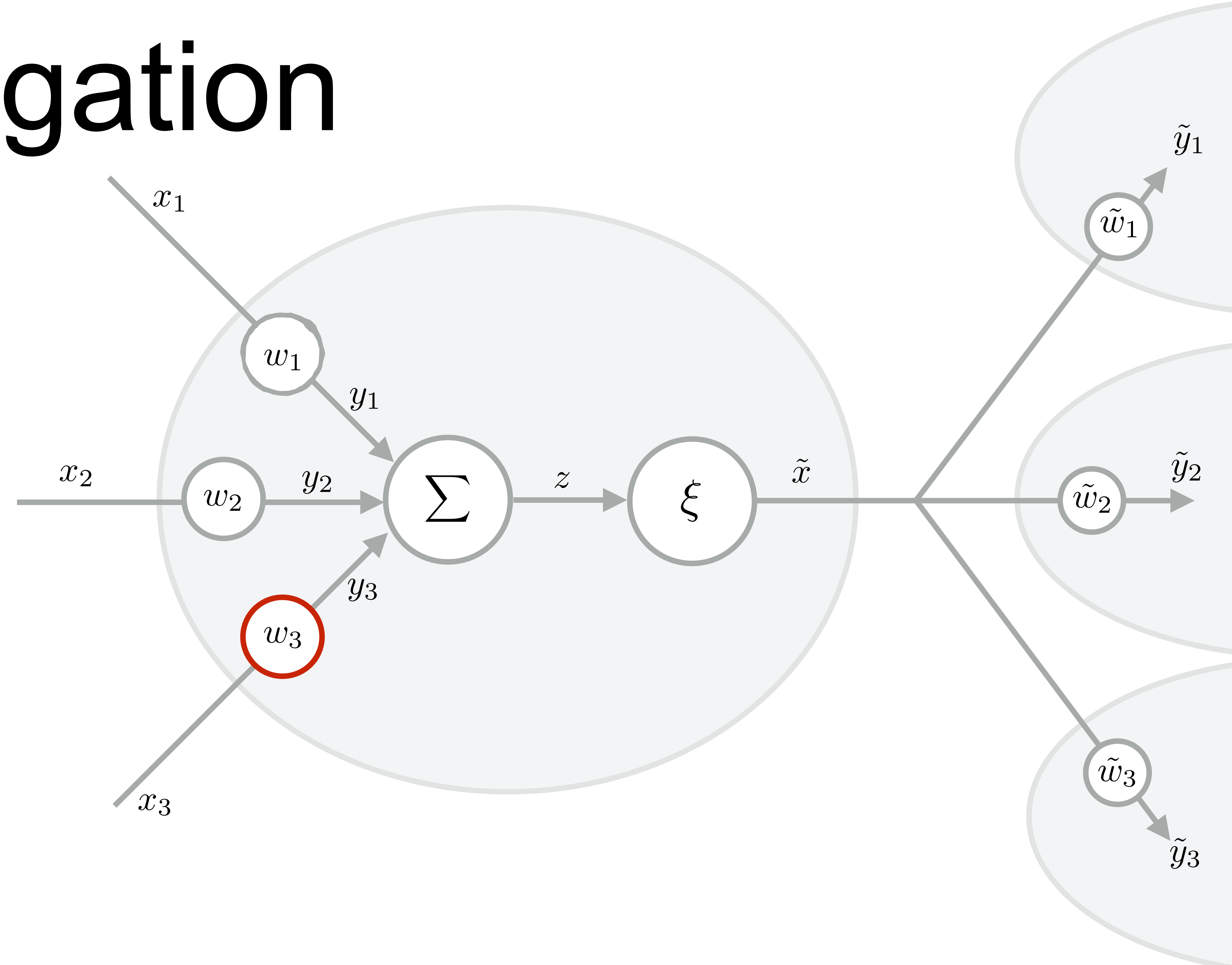
$$\frac{\partial L_i}{\partial w_k} = \frac{\partial L_i}{\partial y_k}\frac{\partial y_k}{\partial w_k} = \frac{\partial L_i}{\partial y_k} x_k$$

Backpropagation



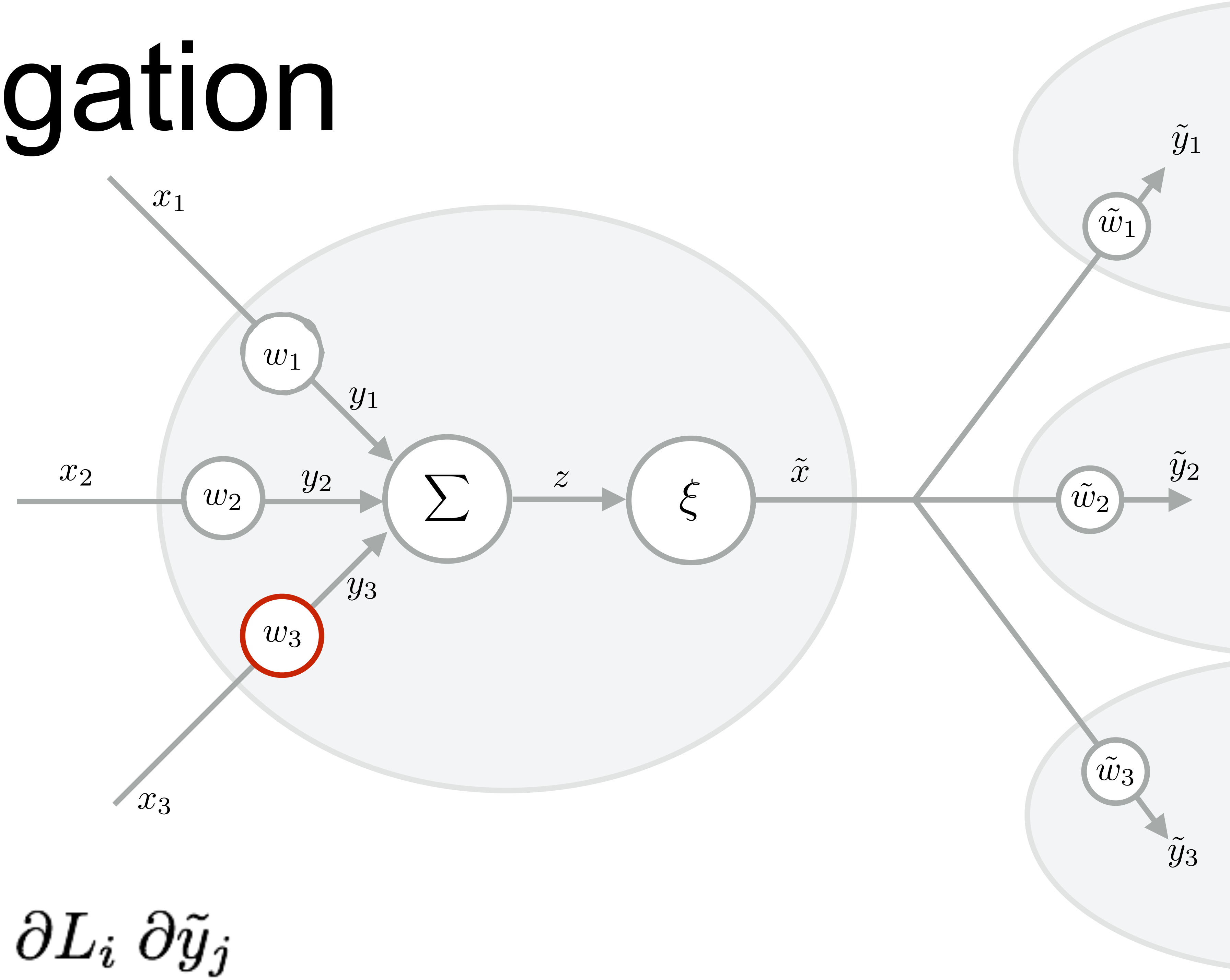
$$\frac{\partial L_i}{\partial w_k} = \frac{\partial L_i}{\partial y_k} x_k$$

Backpropagation



$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

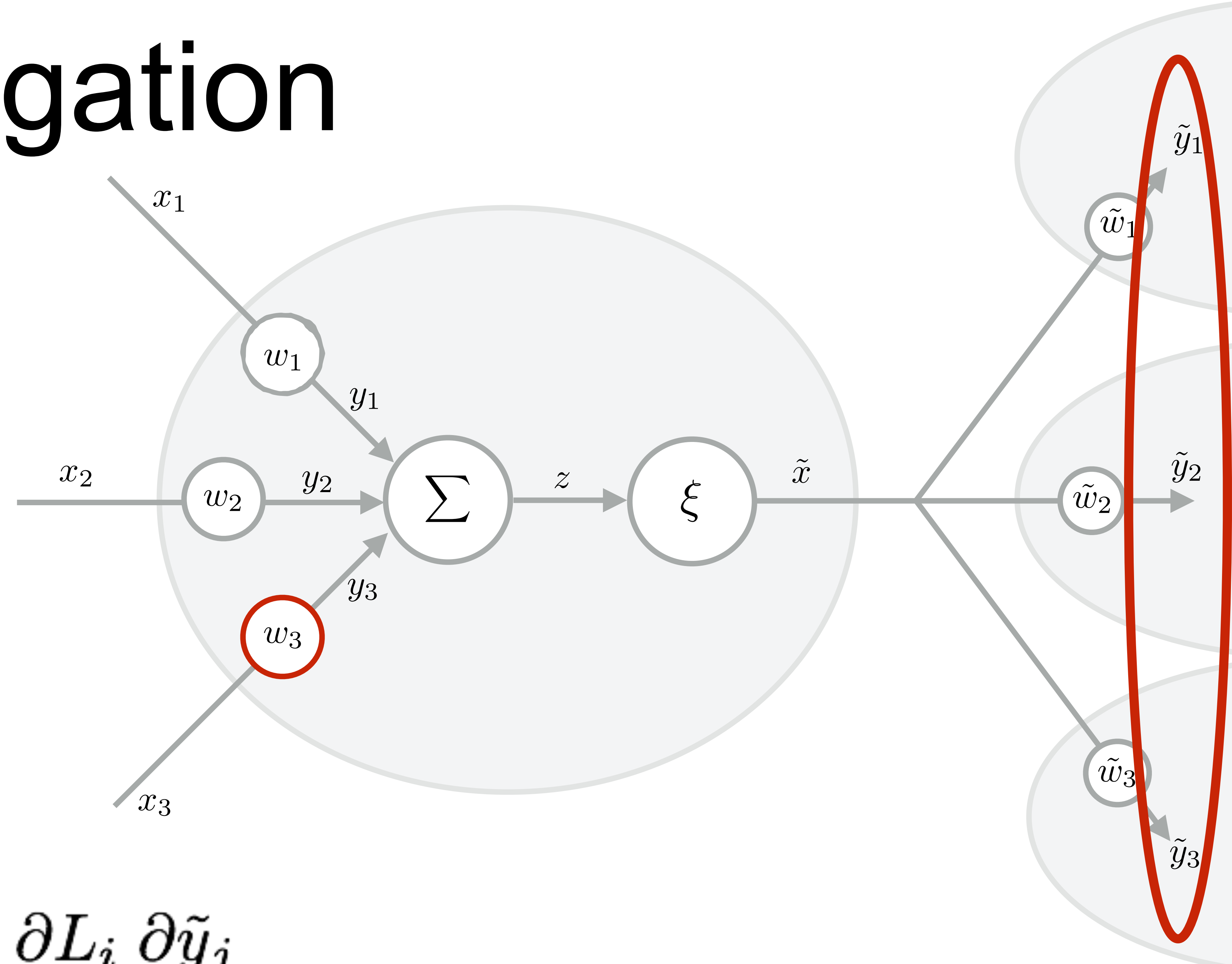
Backpropagation



$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \frac{\partial \tilde{y}_j}{\partial y_k}$$

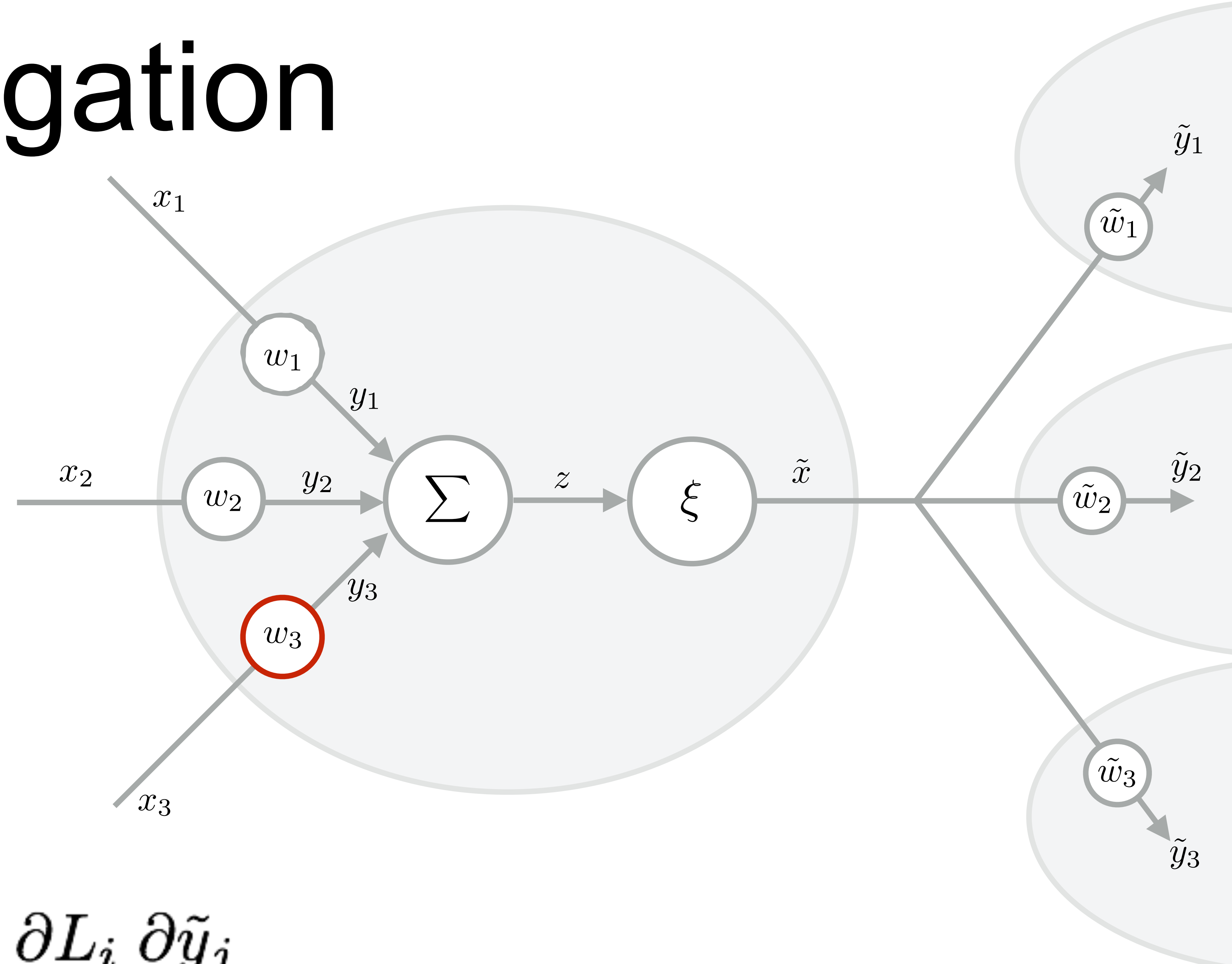
Backpropagation



$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \frac{\partial \tilde{y}_j}{\partial y_k}$$

Backpropagation

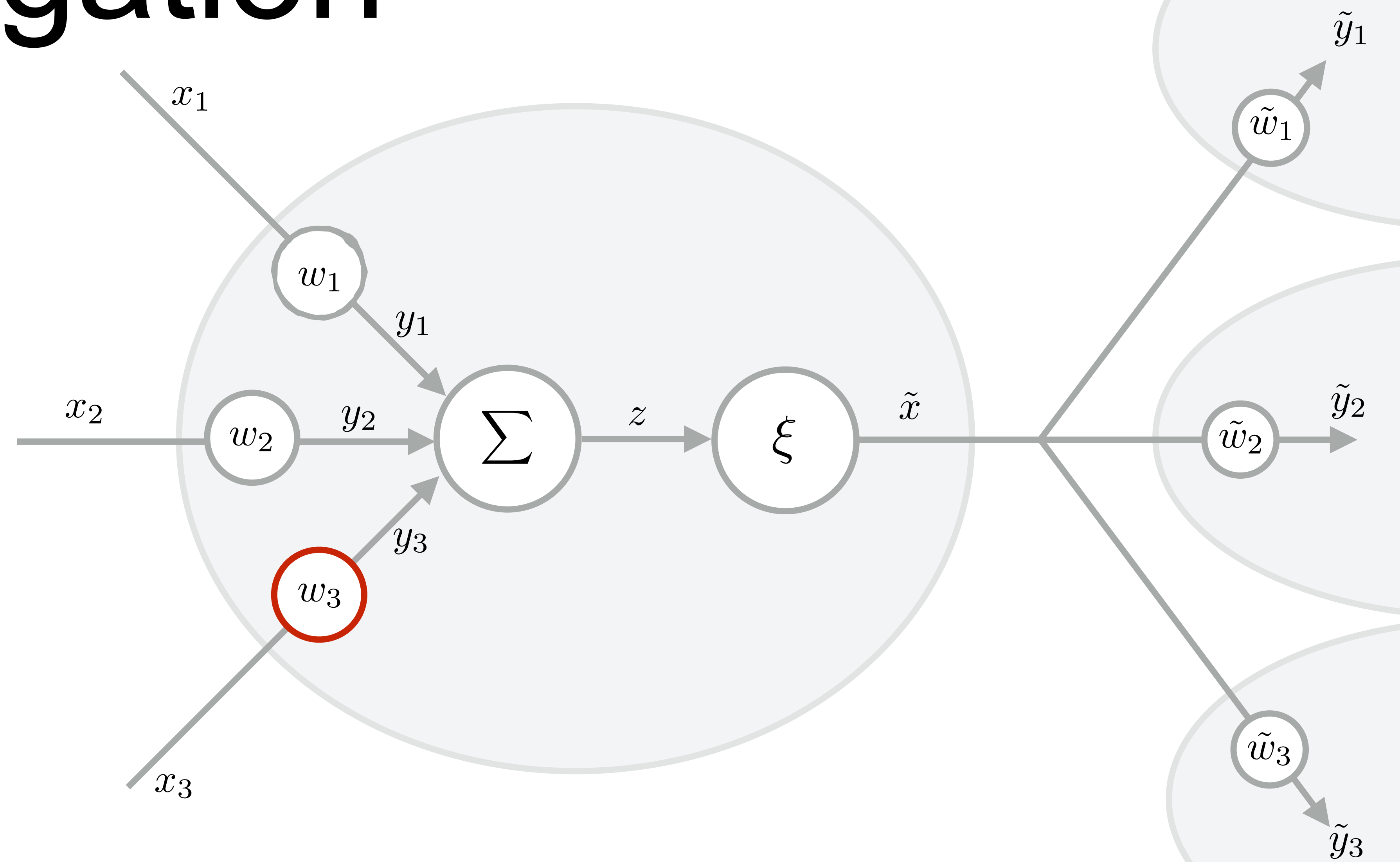


$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}}_{} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \underbrace{\frac{\partial \tilde{y}_j}{\partial y_k}}_{}$$

$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k}$$

Backpropagation



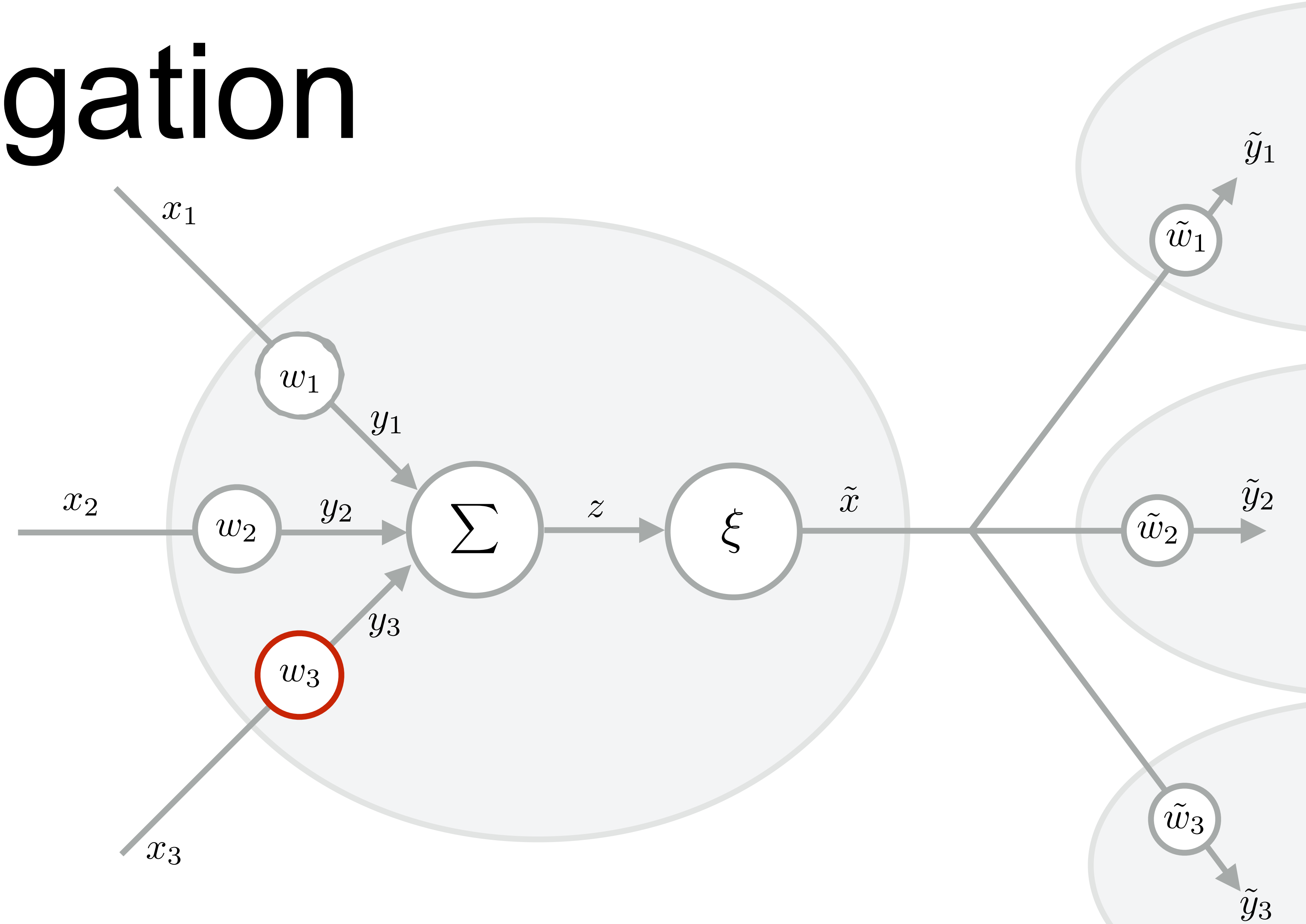
$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \underbrace{\frac{\partial \tilde{y}_j}{\partial y_k}}$$

$$\frac{\partial \tilde{y}_j}{\partial \tilde{x}} = \frac{\partial \tilde{w}_j \tilde{x}}{\partial \tilde{x}} = \tilde{w}_j$$

$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k}$$

Backpropagation



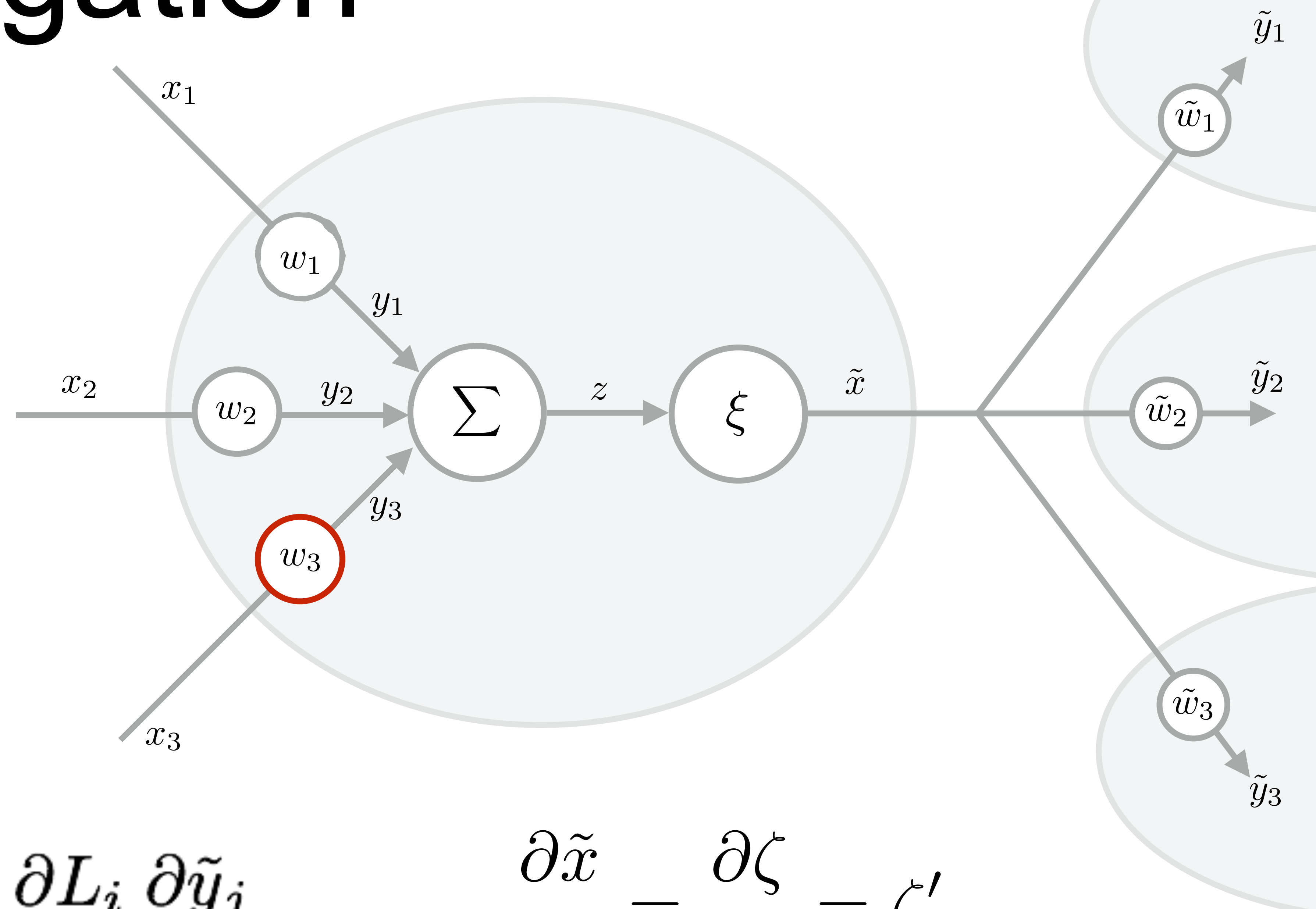
$$\frac{\partial L_i}{\partial w_k} = \frac{\partial L_i}{\partial y_k} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \frac{\partial \tilde{y}_j}{\partial y_k}$$

$$\frac{\partial \tilde{y}_j}{\partial \tilde{x}} = \frac{\partial \tilde{w}_j \tilde{x}}{\partial \tilde{x}} = \tilde{w}_j$$

$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k} = \tilde{w}_j$$

Backpropagation



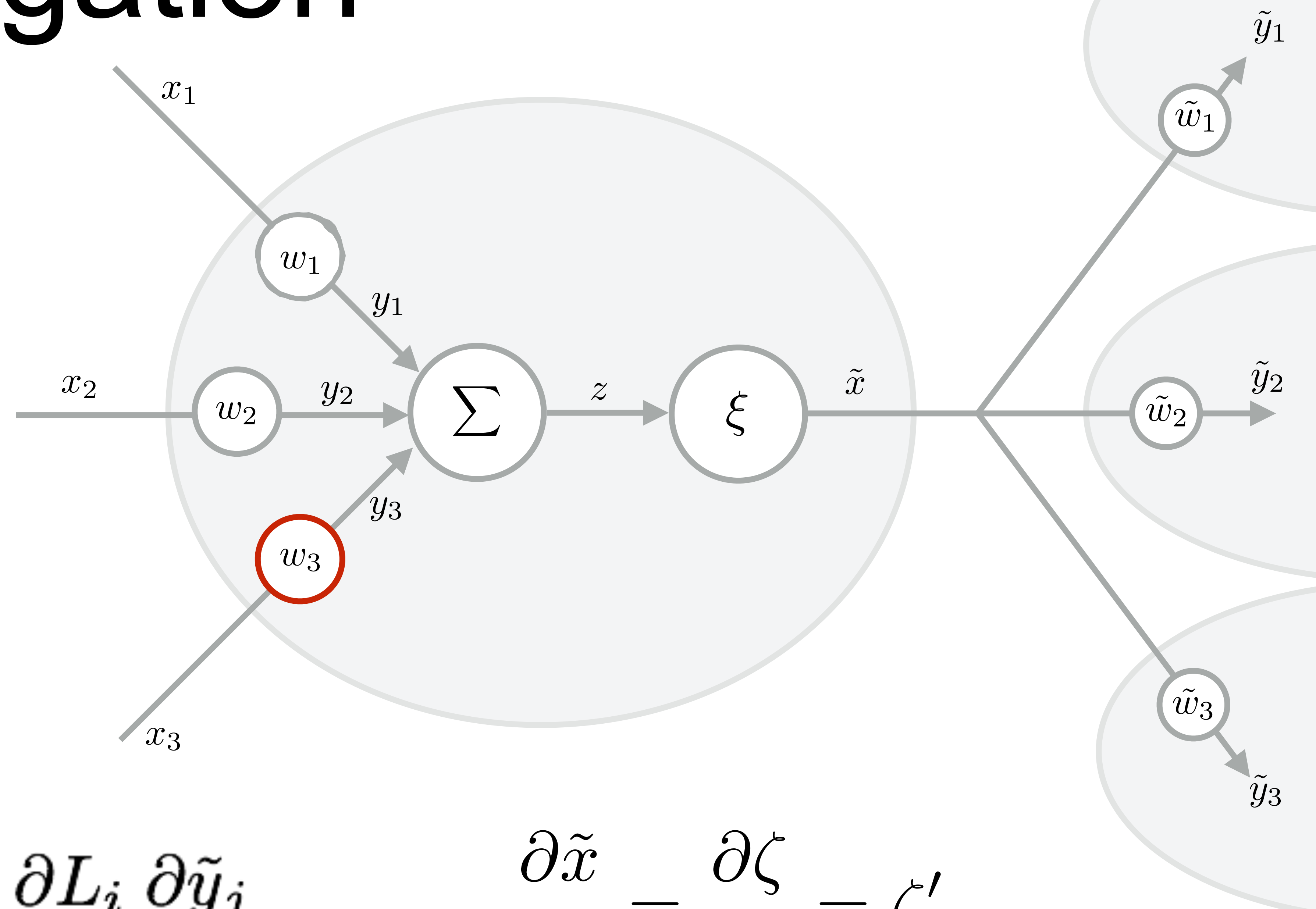
$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \underbrace{\frac{\partial \tilde{y}_j}{\partial y_k}}$$

$$\frac{\partial \tilde{x}}{\partial z} = \frac{\partial \zeta}{\partial z} = \zeta'$$

$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k} = \tilde{w}_j$$

Backpropagation



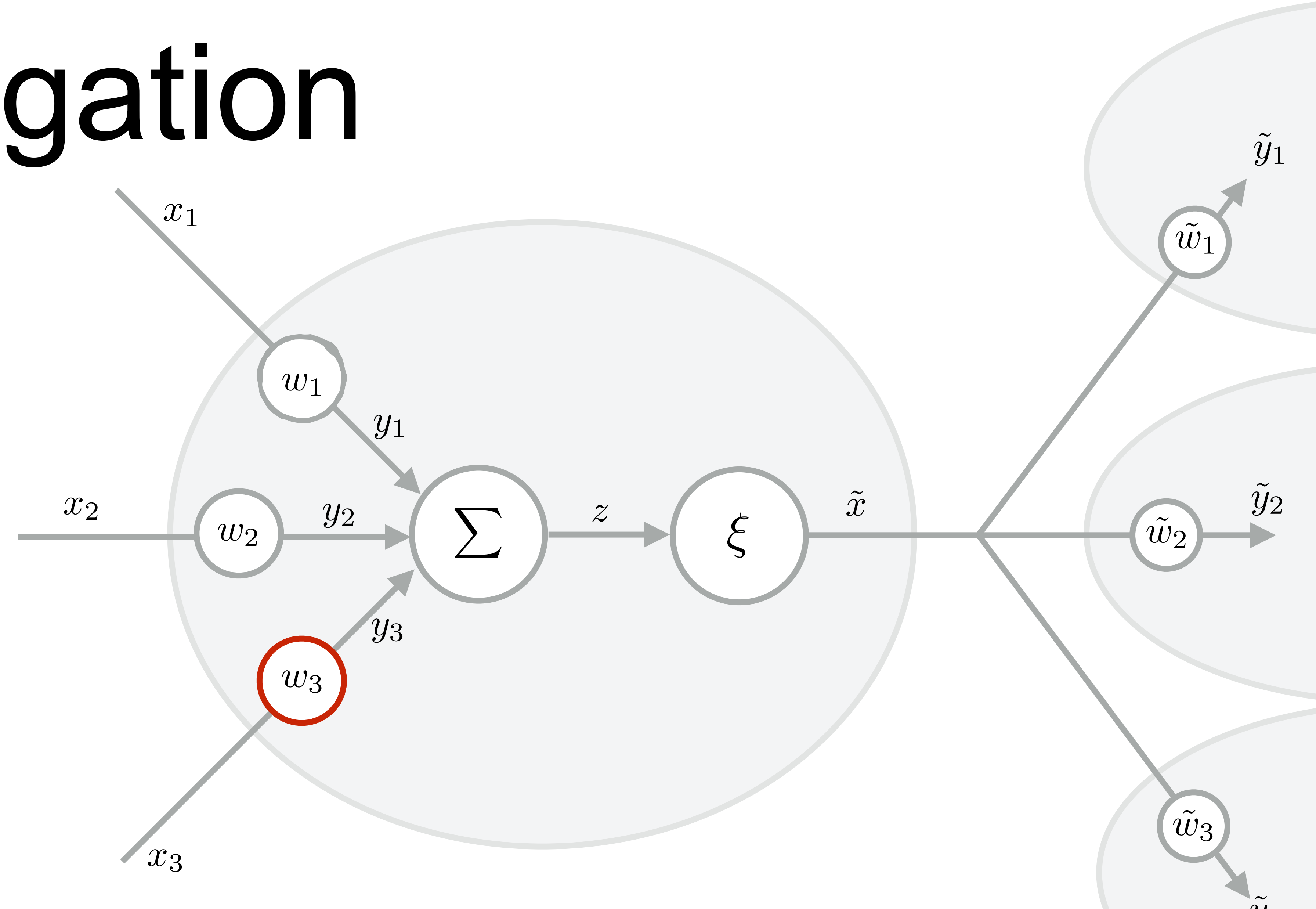
$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \underbrace{\frac{\partial \tilde{y}_j}{\partial y_k}}$$

$$\frac{\partial \tilde{x}}{\partial z} = \frac{\partial \zeta}{\partial z} = \zeta'$$

$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k} = \tilde{w}_j \xi'$$

Backpropagation



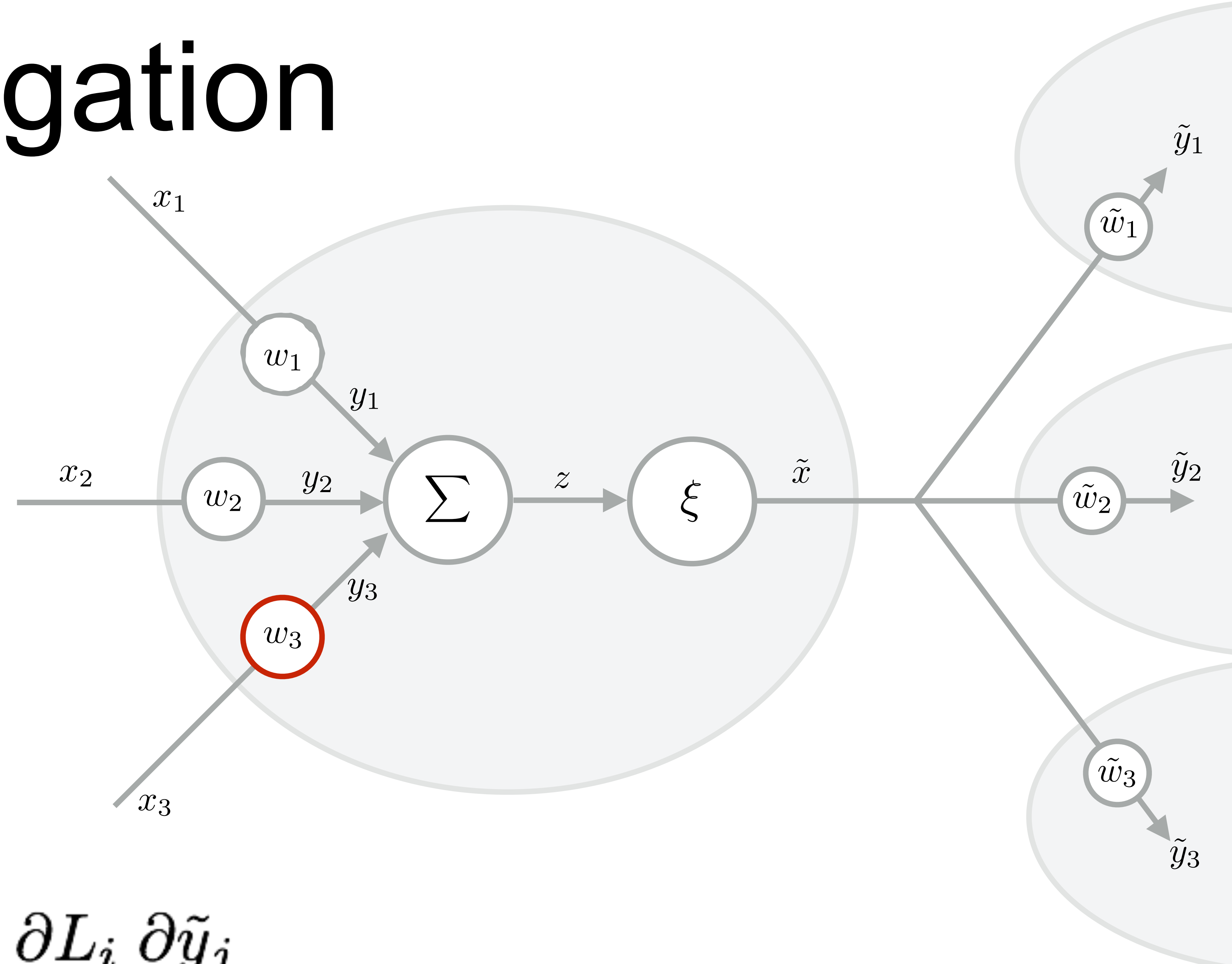
$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \underbrace{\frac{\partial \tilde{y}_j}{\partial y_k}}$$

$$\frac{\partial z}{\partial y_k} = \frac{\partial \sum_i y_i}{\partial y_k} = 1$$

$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k} = \tilde{w}_j \xi'$$

Backpropagation

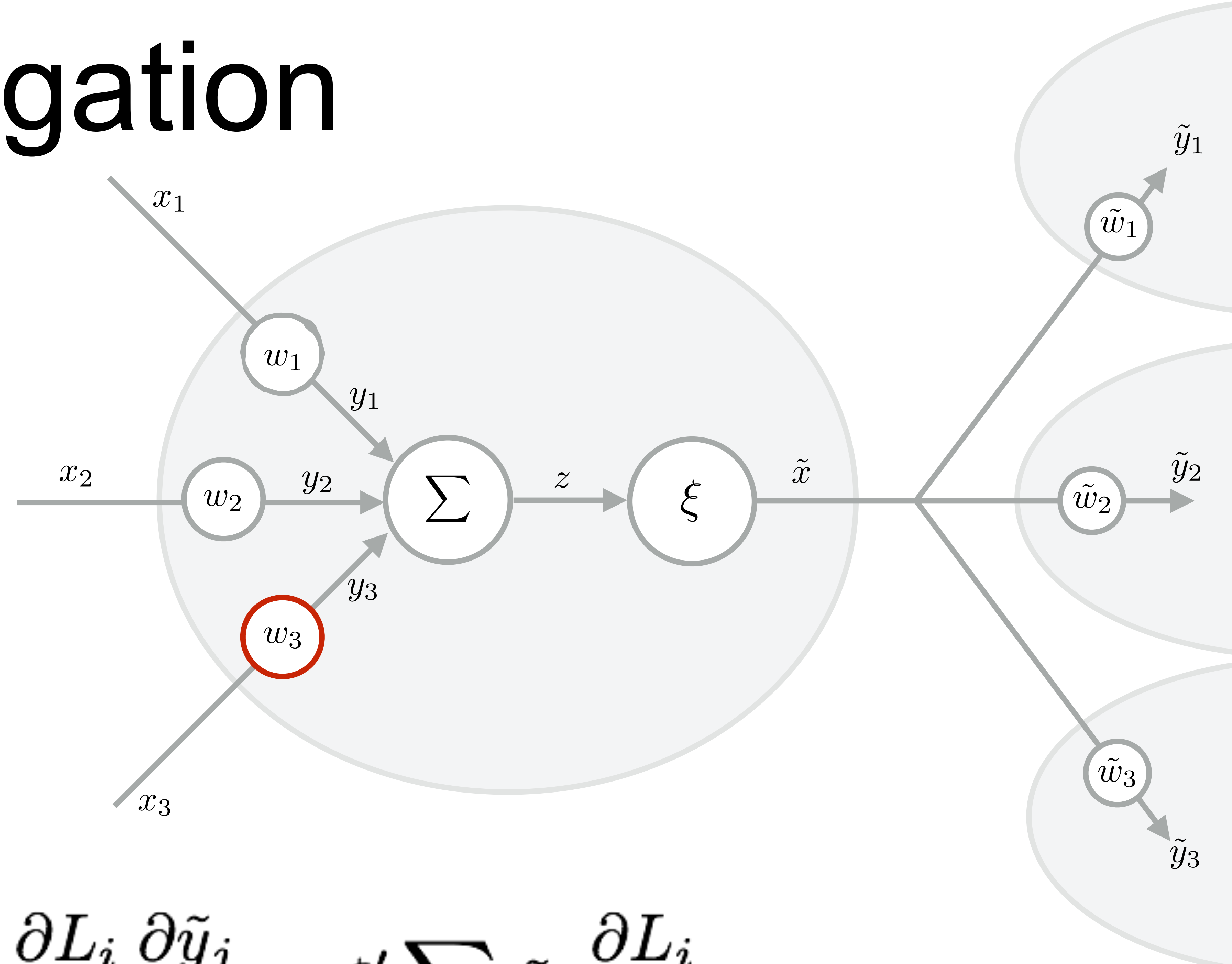


$$\frac{\partial L_i}{\partial w_k} = \underbrace{\frac{\partial L_i}{\partial y_k}} x_k$$

$$\frac{\partial L_i}{\partial y_k} = \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \underbrace{\frac{\partial \tilde{y}_j}{\partial y_k}}$$

$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k} = \tilde{w}_j \xi'$$

Backpropagation

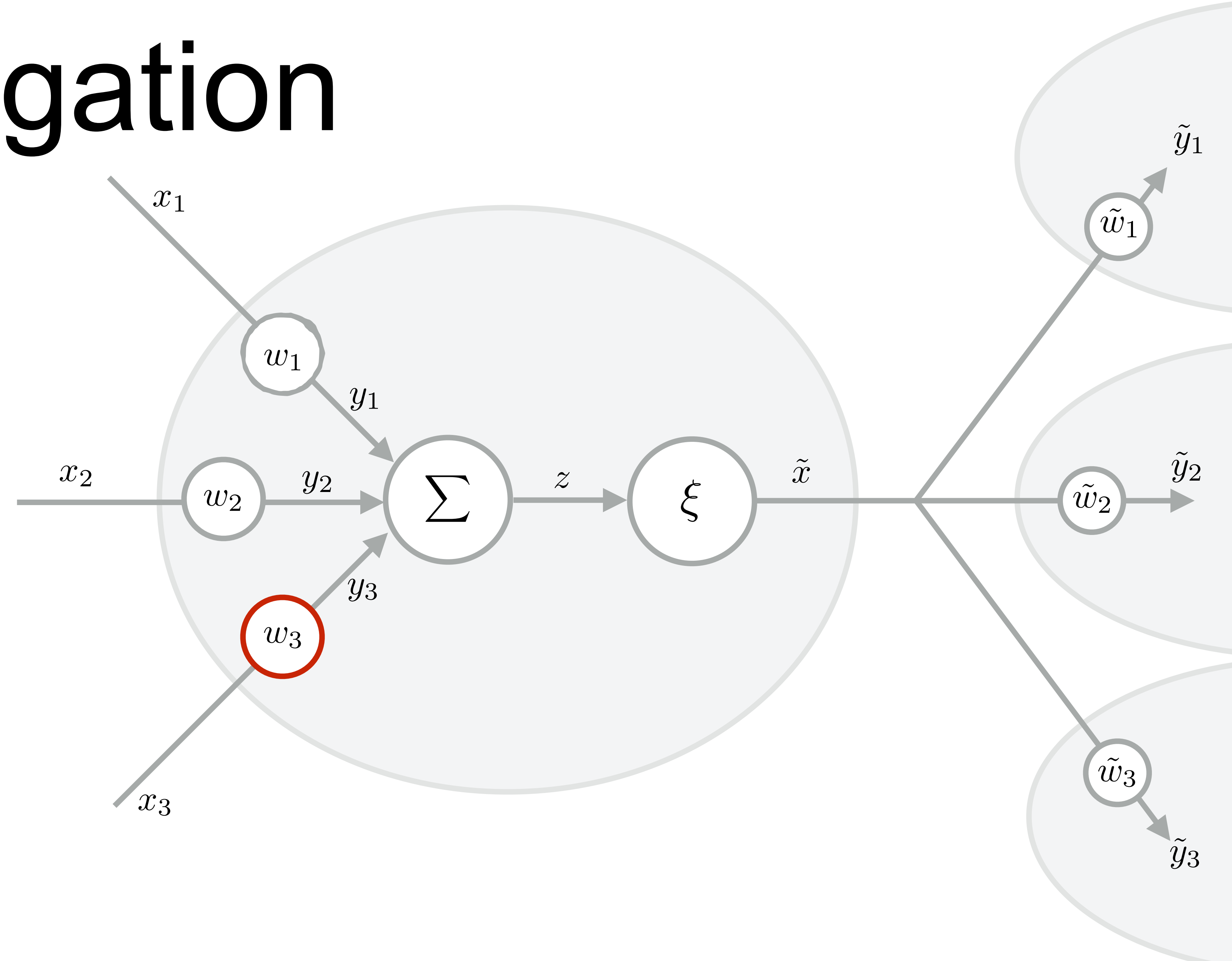


$$\frac{\partial L_i}{\partial w_k} = \frac{\partial L_i}{\partial y_k} x_k$$

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

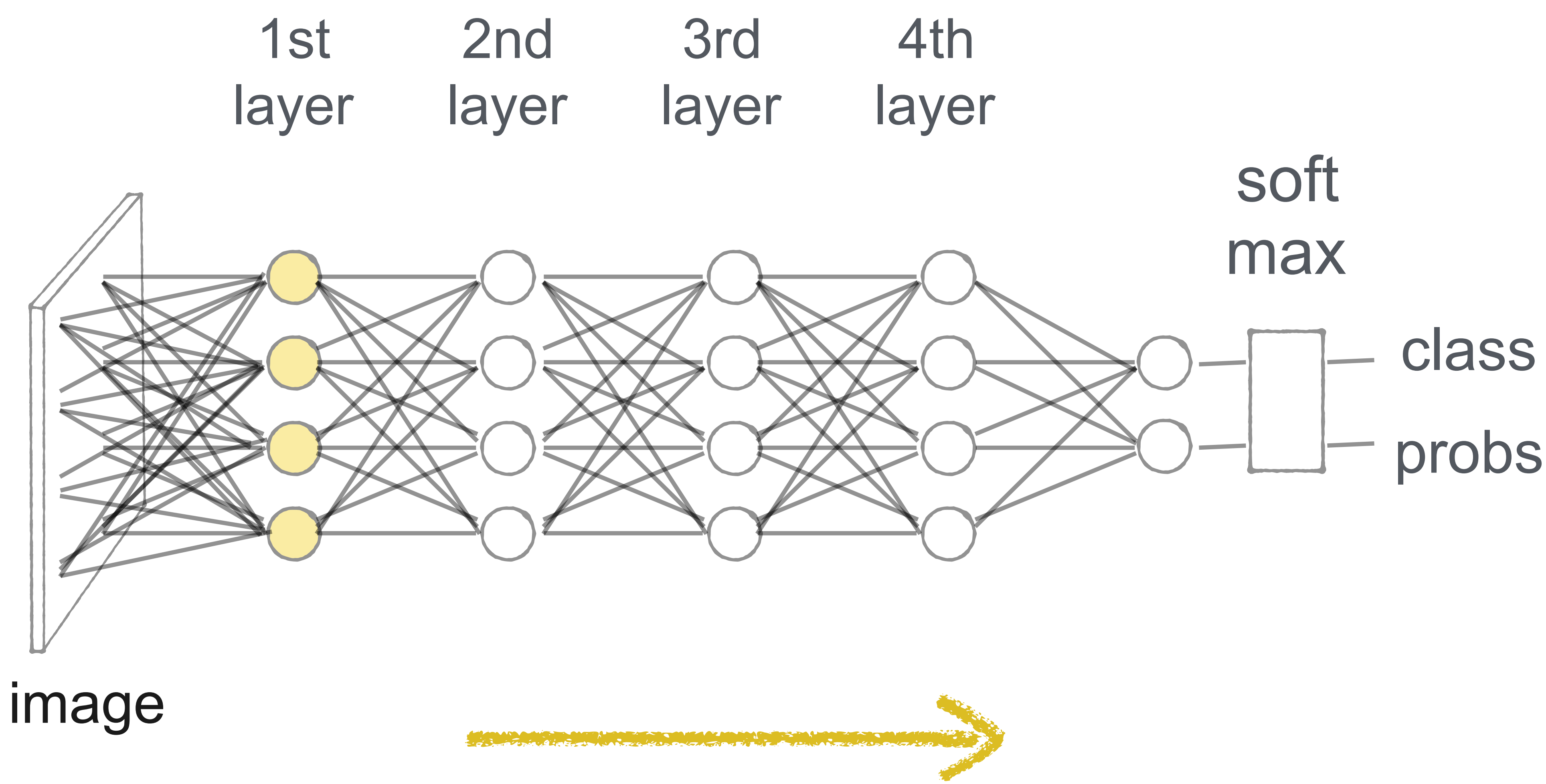
$$\frac{\partial \tilde{y}_j}{\partial y_k} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_k} = \tilde{w}_j \xi'$$

Backpropagation



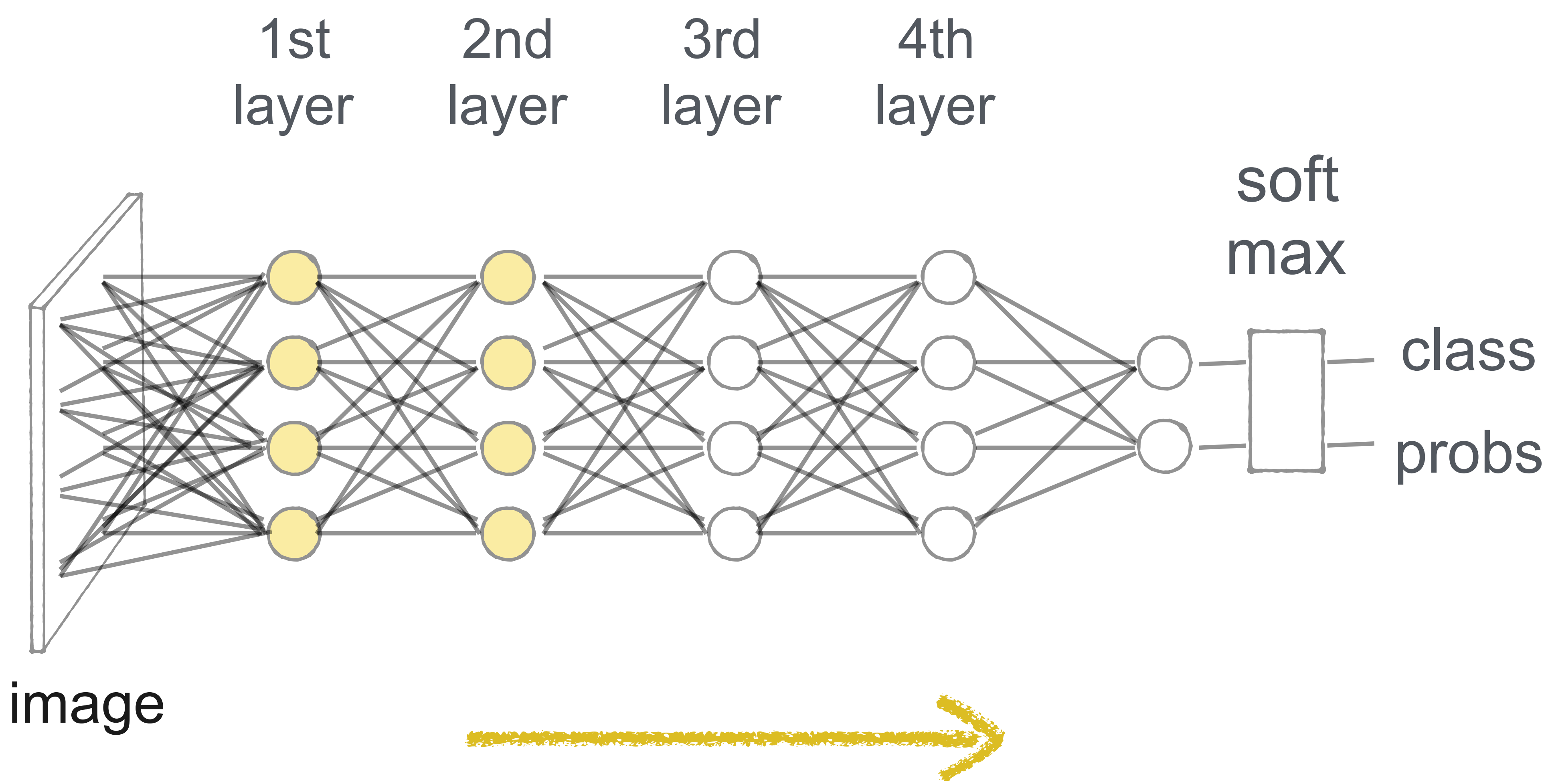
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Forward Pass



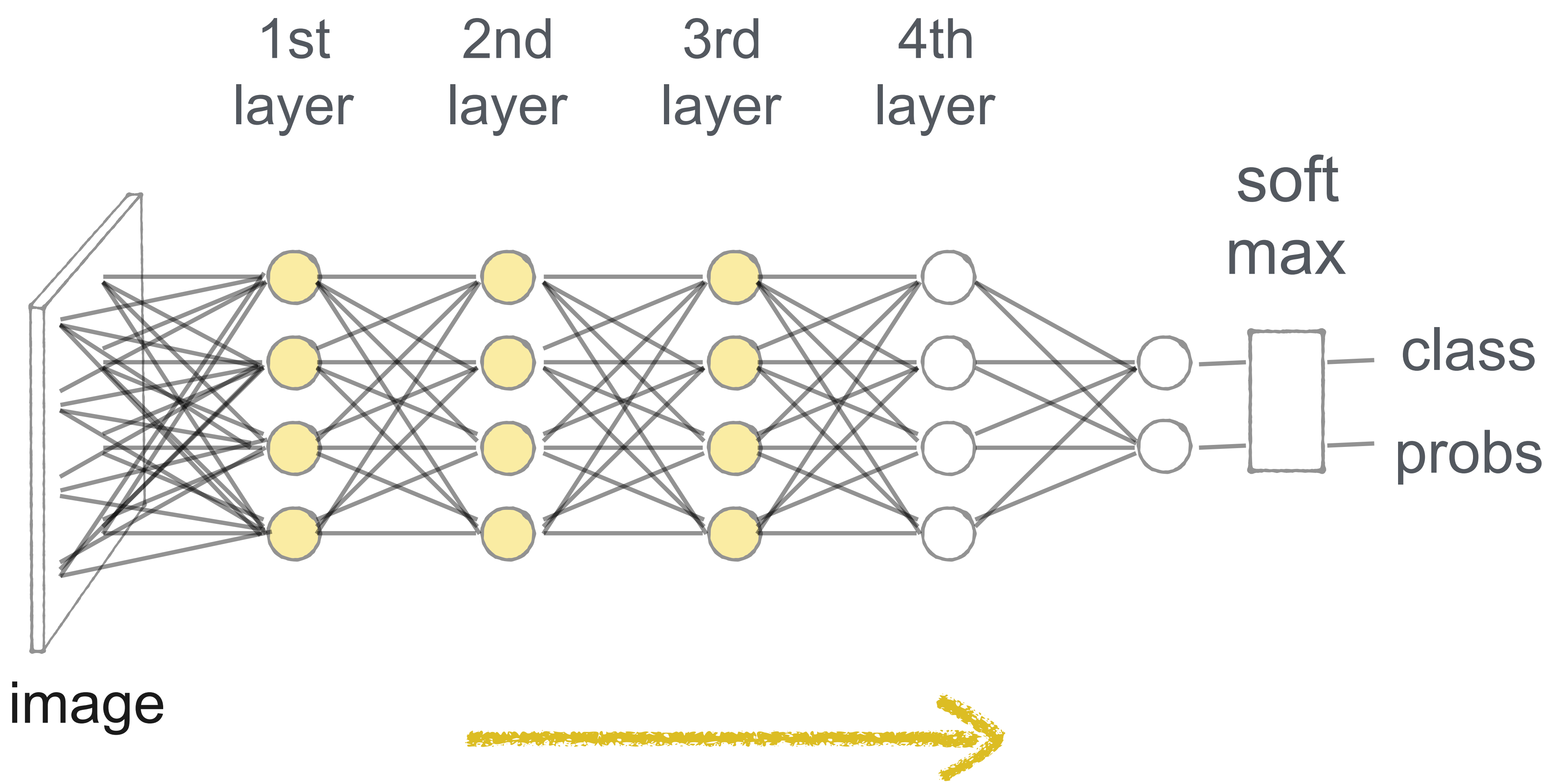
$$\frac{\partial L_i}{\partial w_k} = \underline{x_k} \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Forward Pass



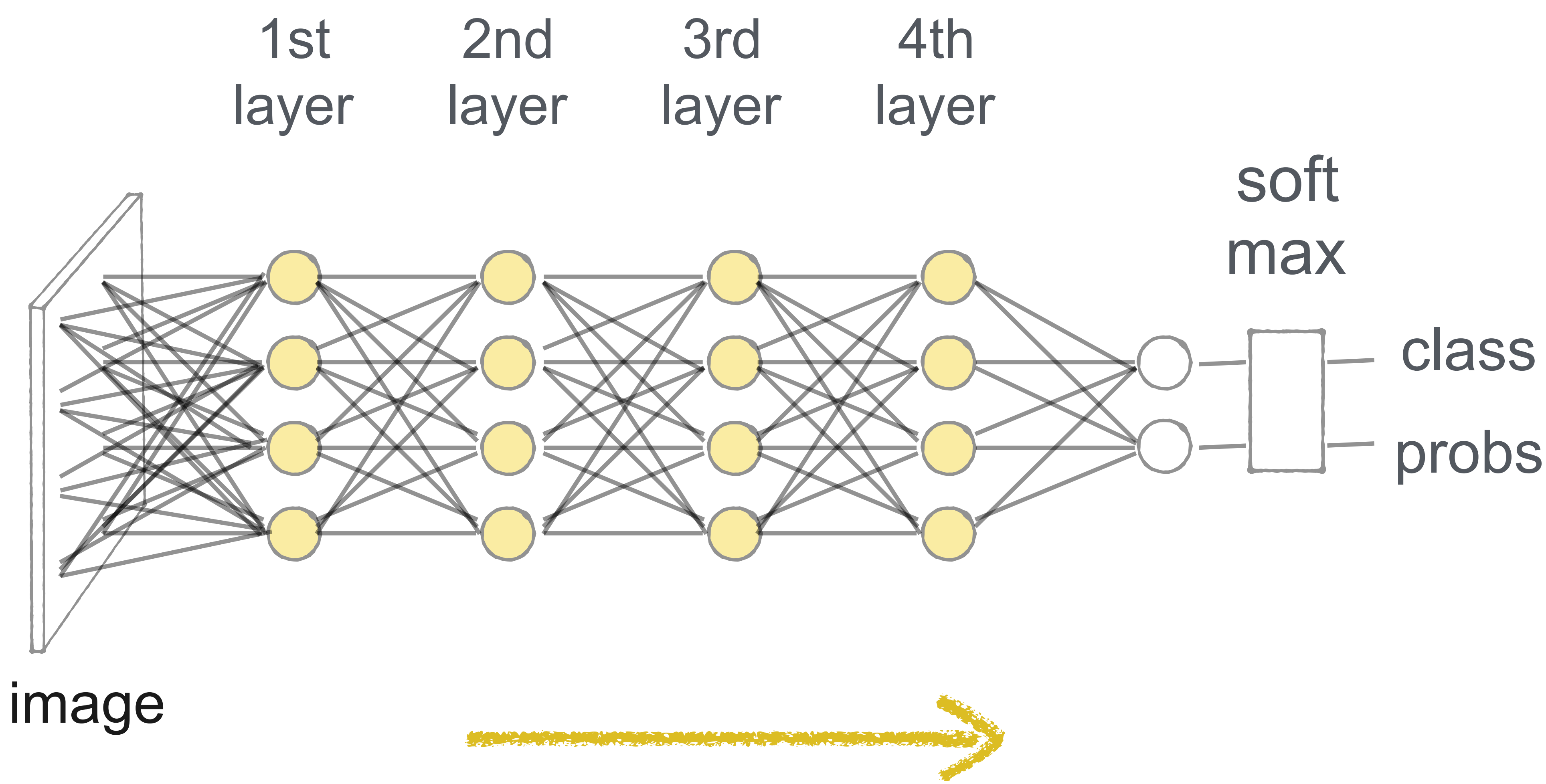
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Forward Pass



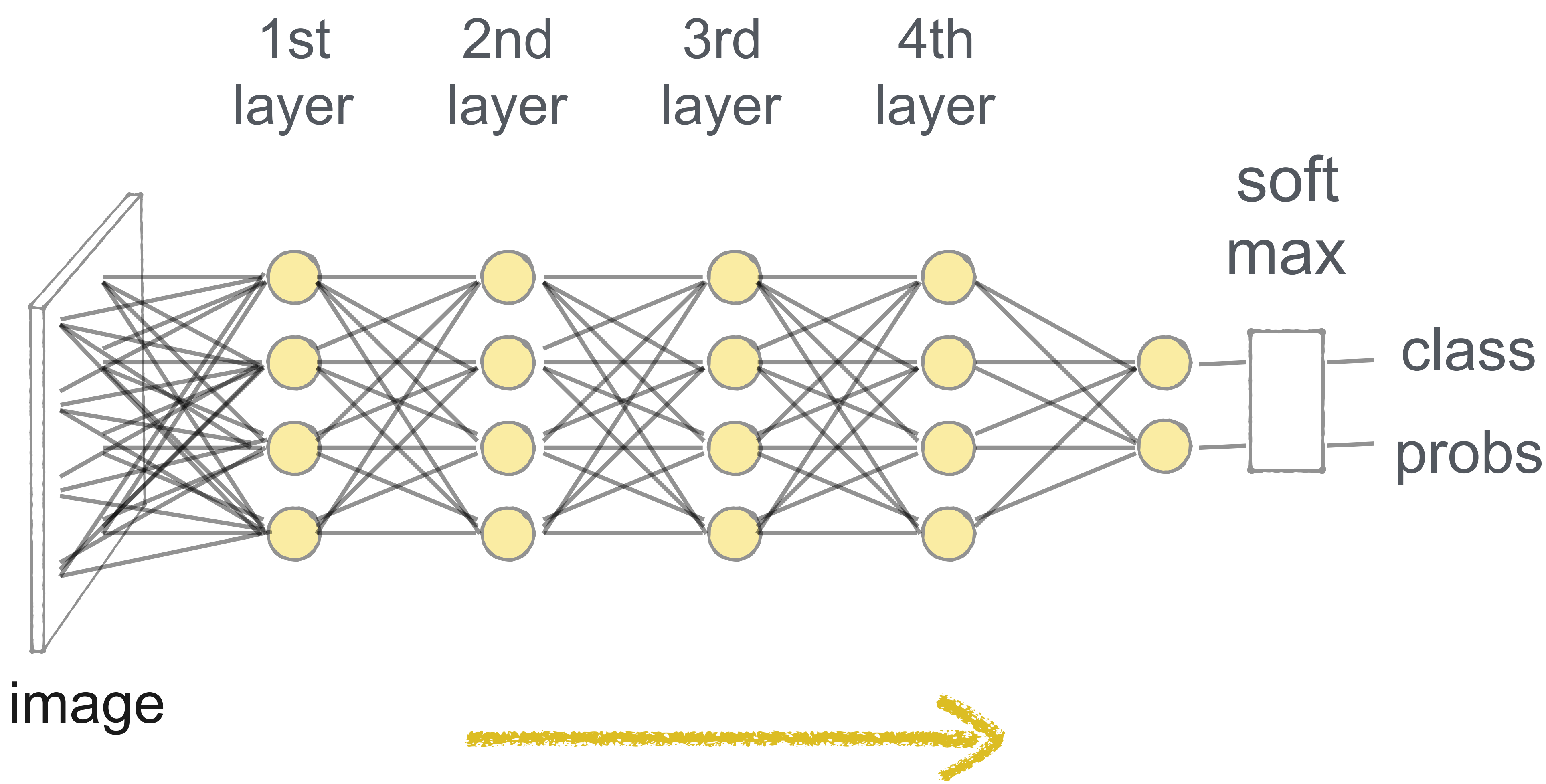
$$\frac{\partial L_i}{\partial w_k} = \underline{x_k} \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Forward Pass



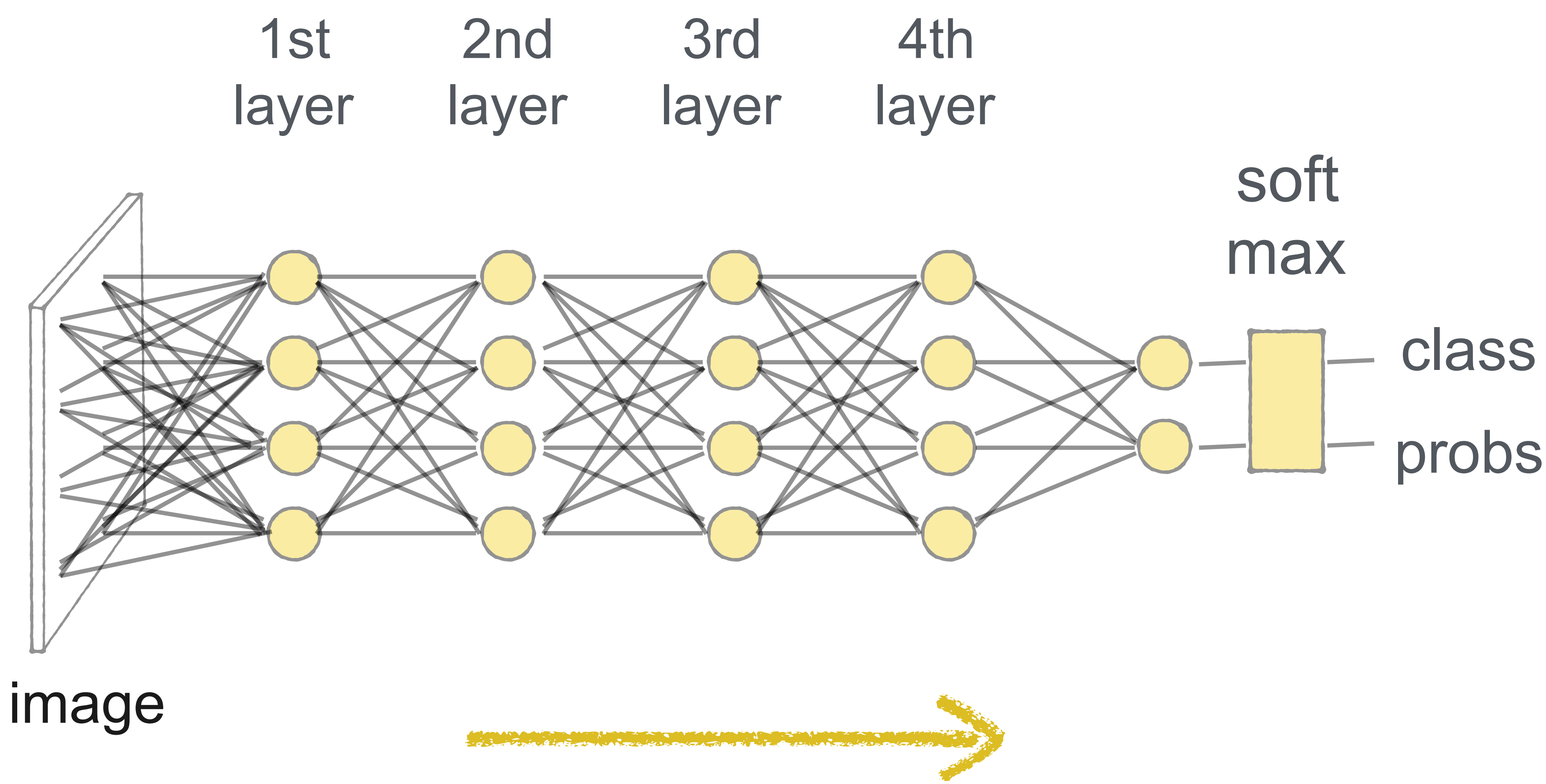
$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\,\xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Forward Pass



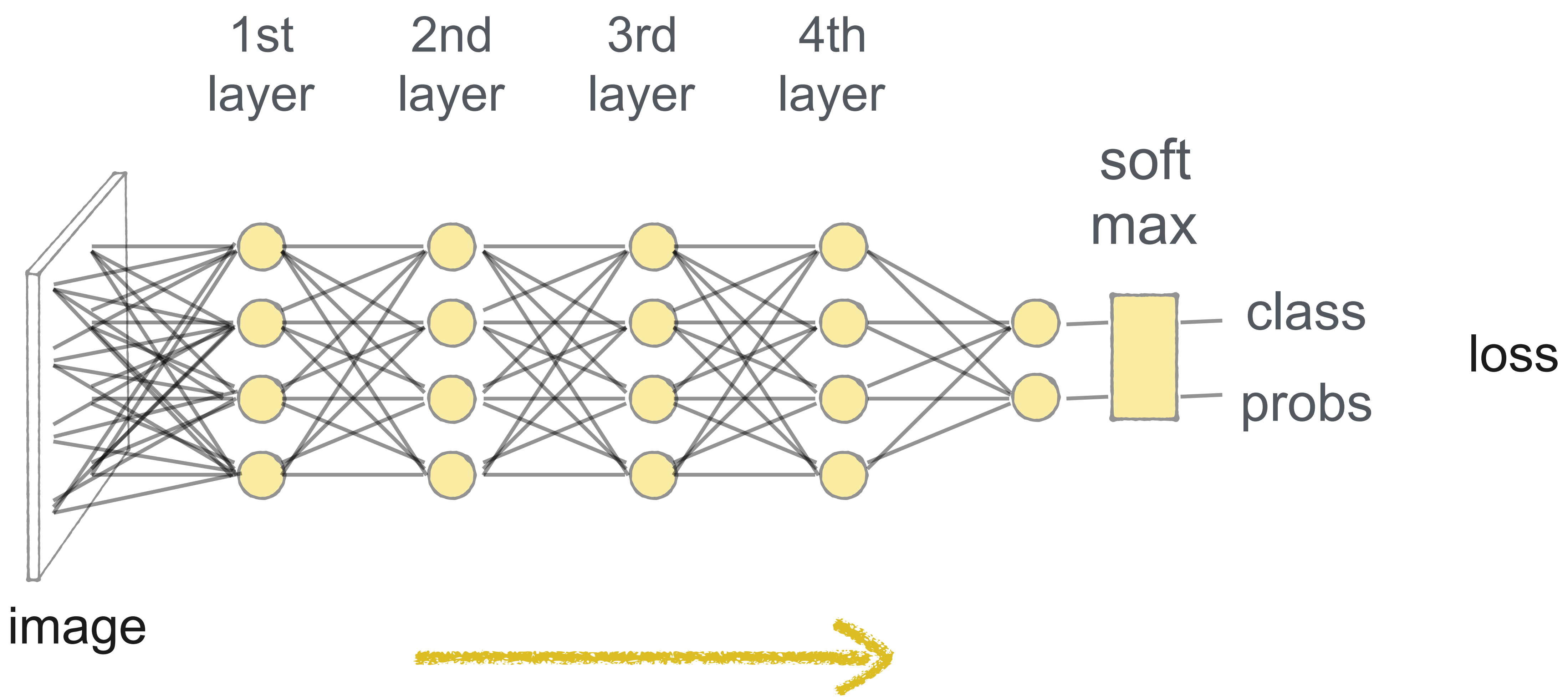
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Forward Pass



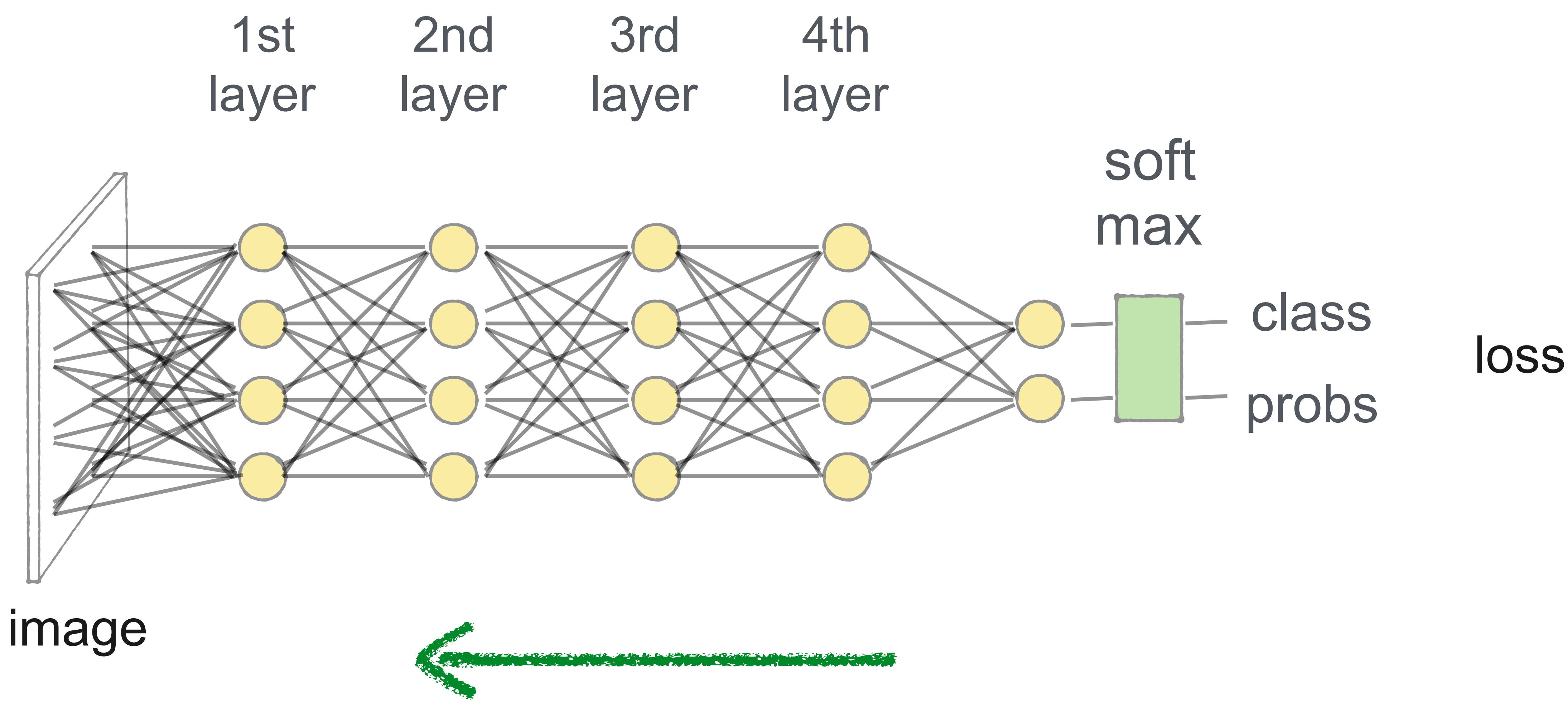
$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Forward Pass



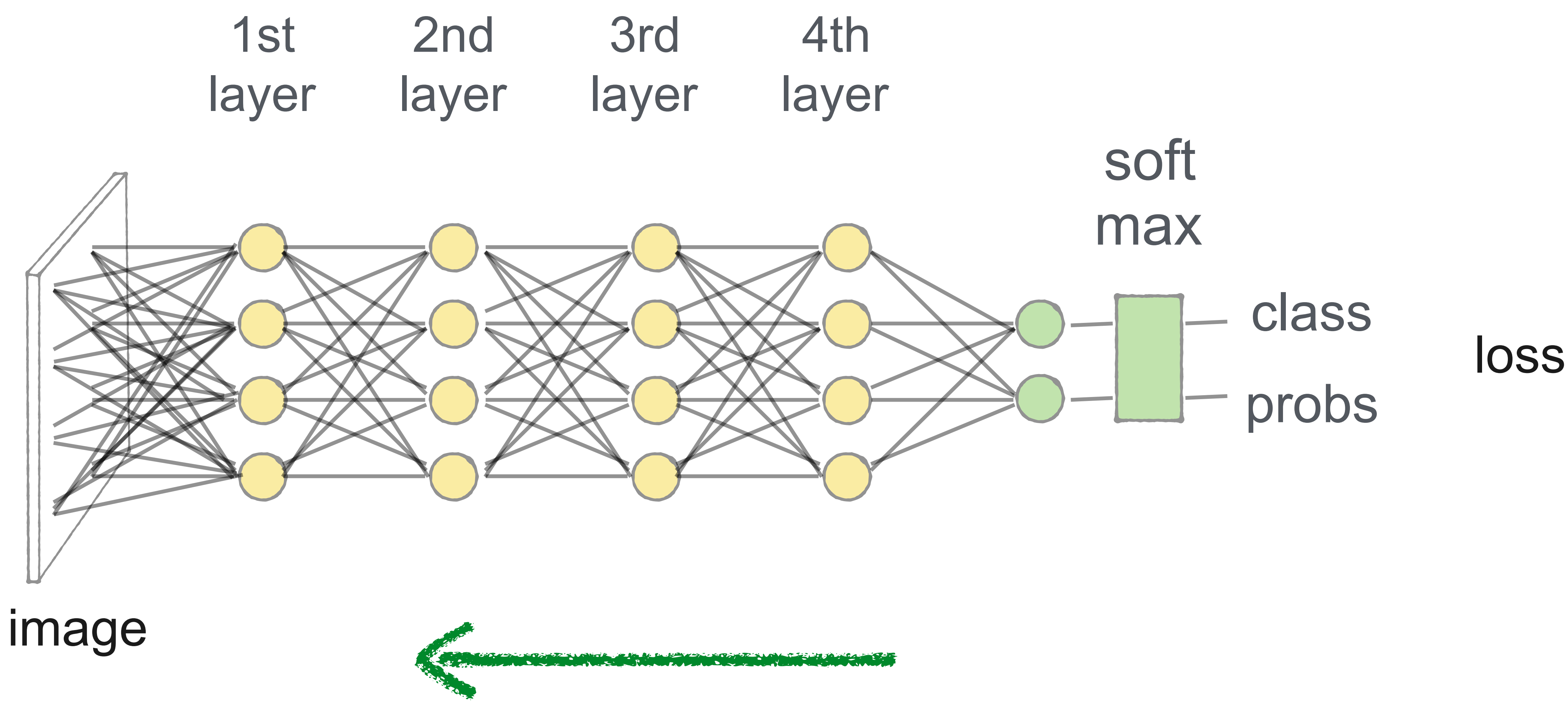
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation



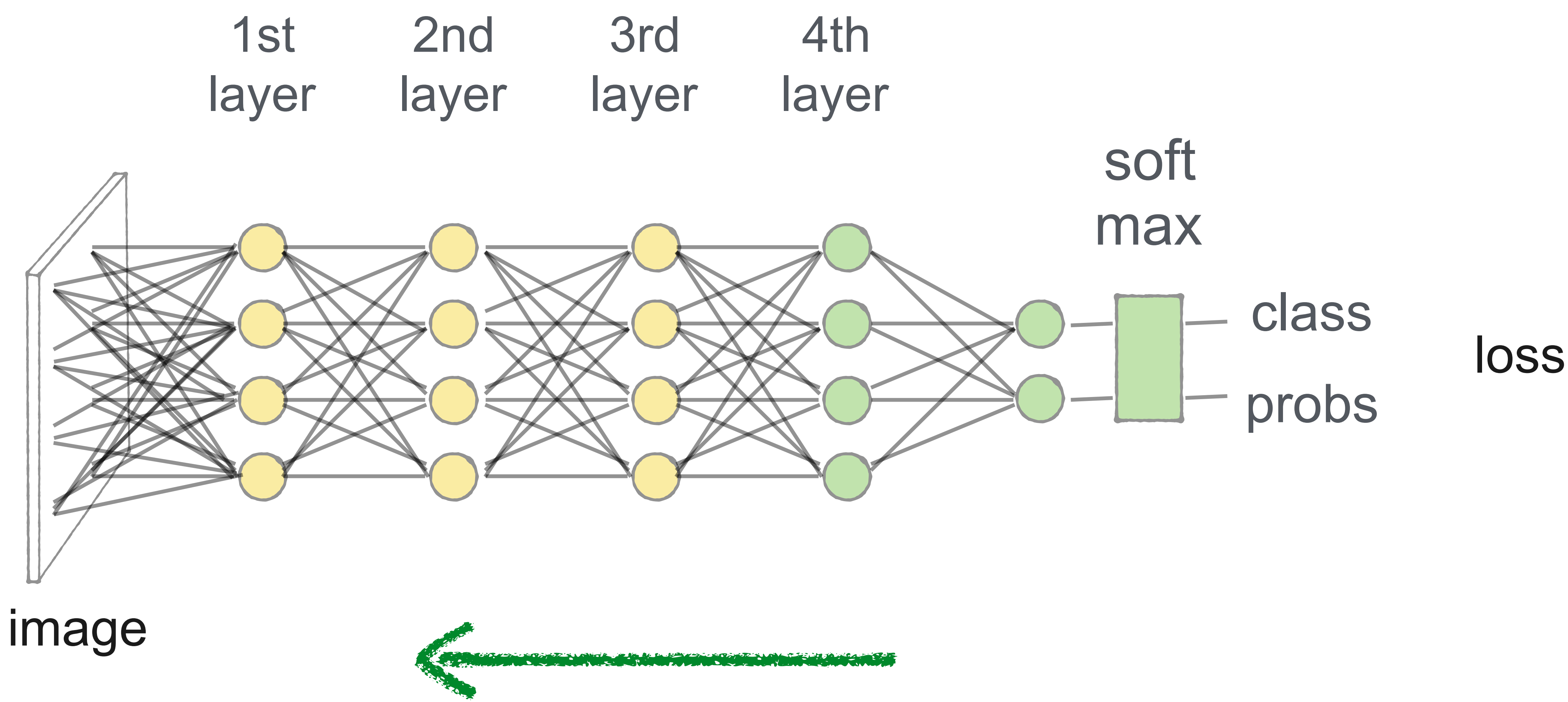
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation



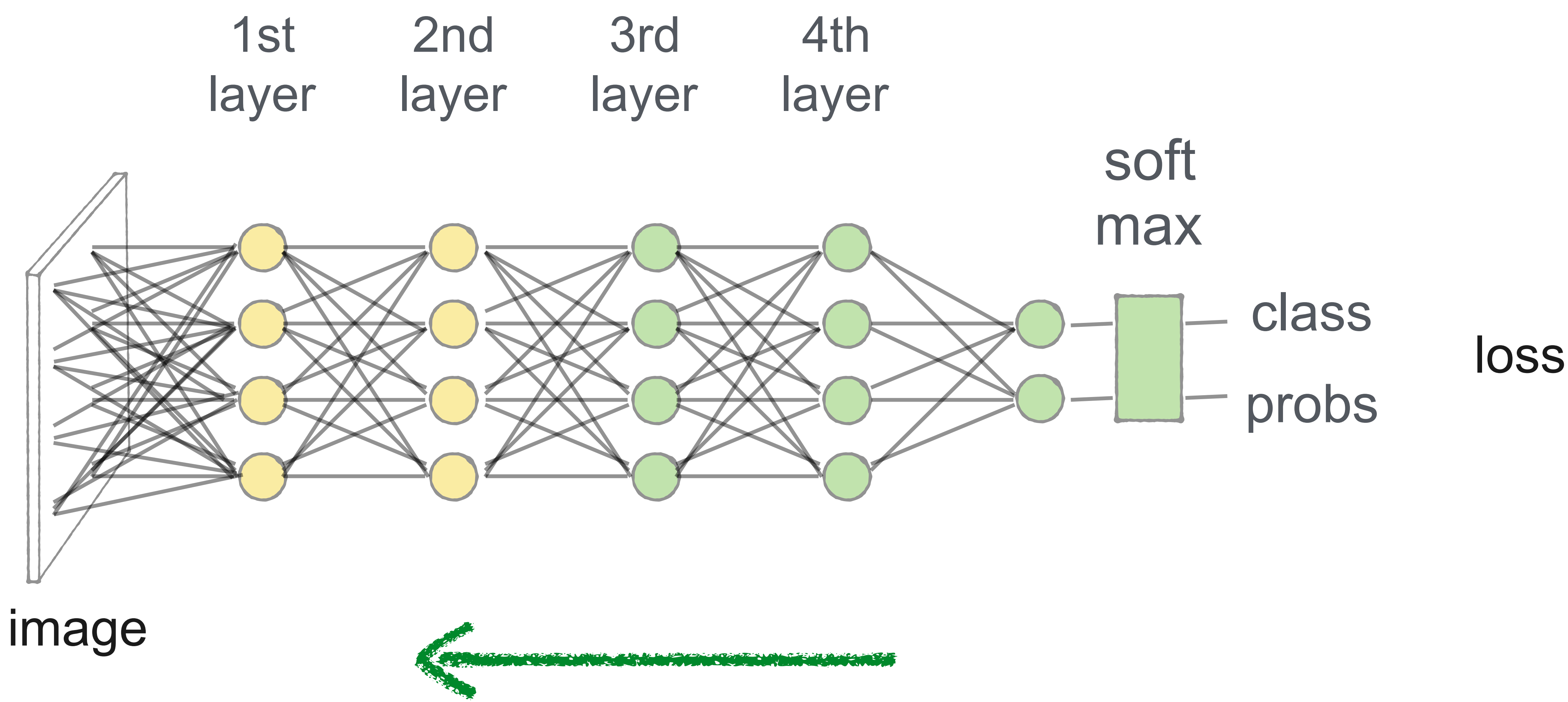
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation



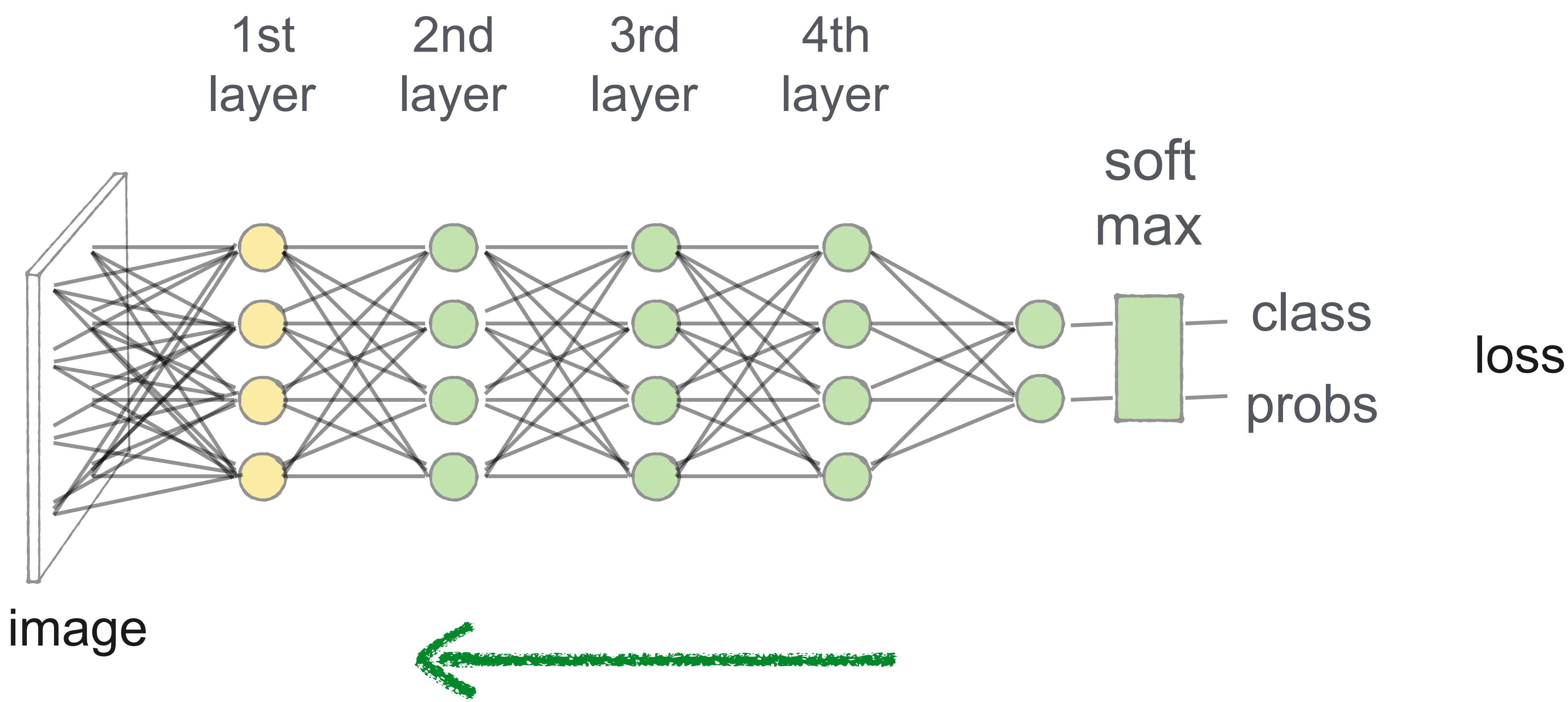
$$\frac{\partial L_i}{\partial w_k} = \underline{x_k} \underline{\xi'} \sum_j \tilde{w}_j \underline{\frac{\partial L_i}{\partial \tilde{y}_j}}$$

Backpropagation



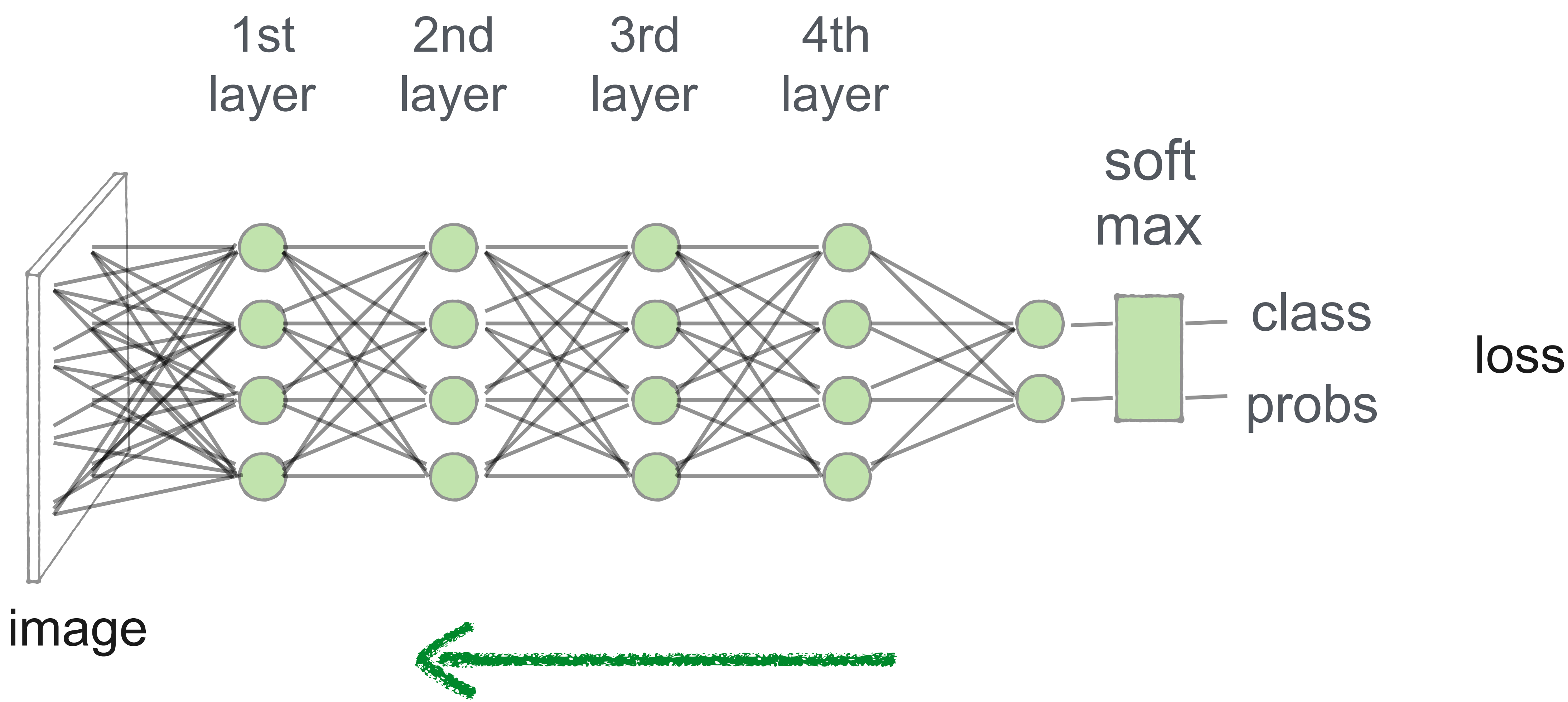
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation



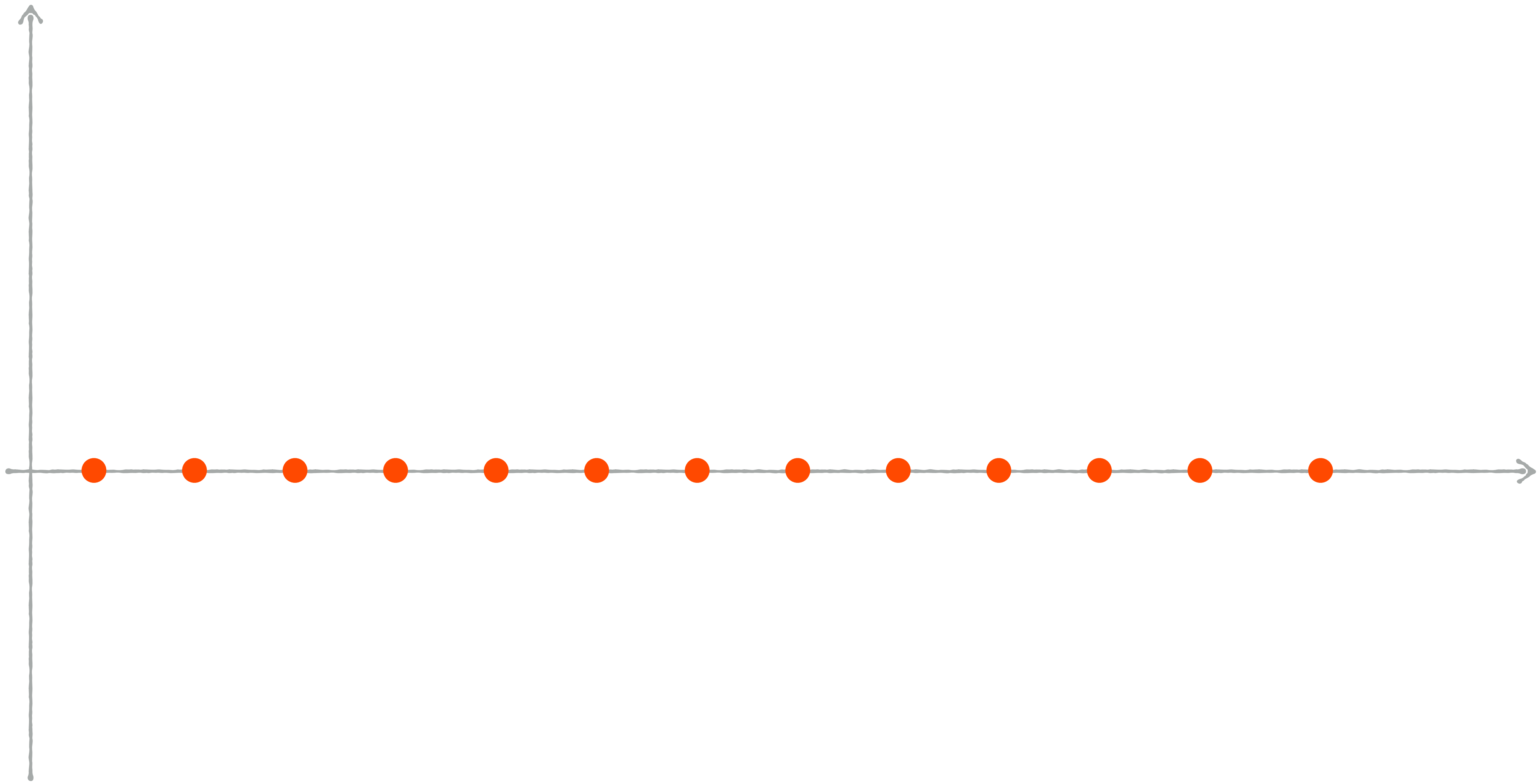
$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation

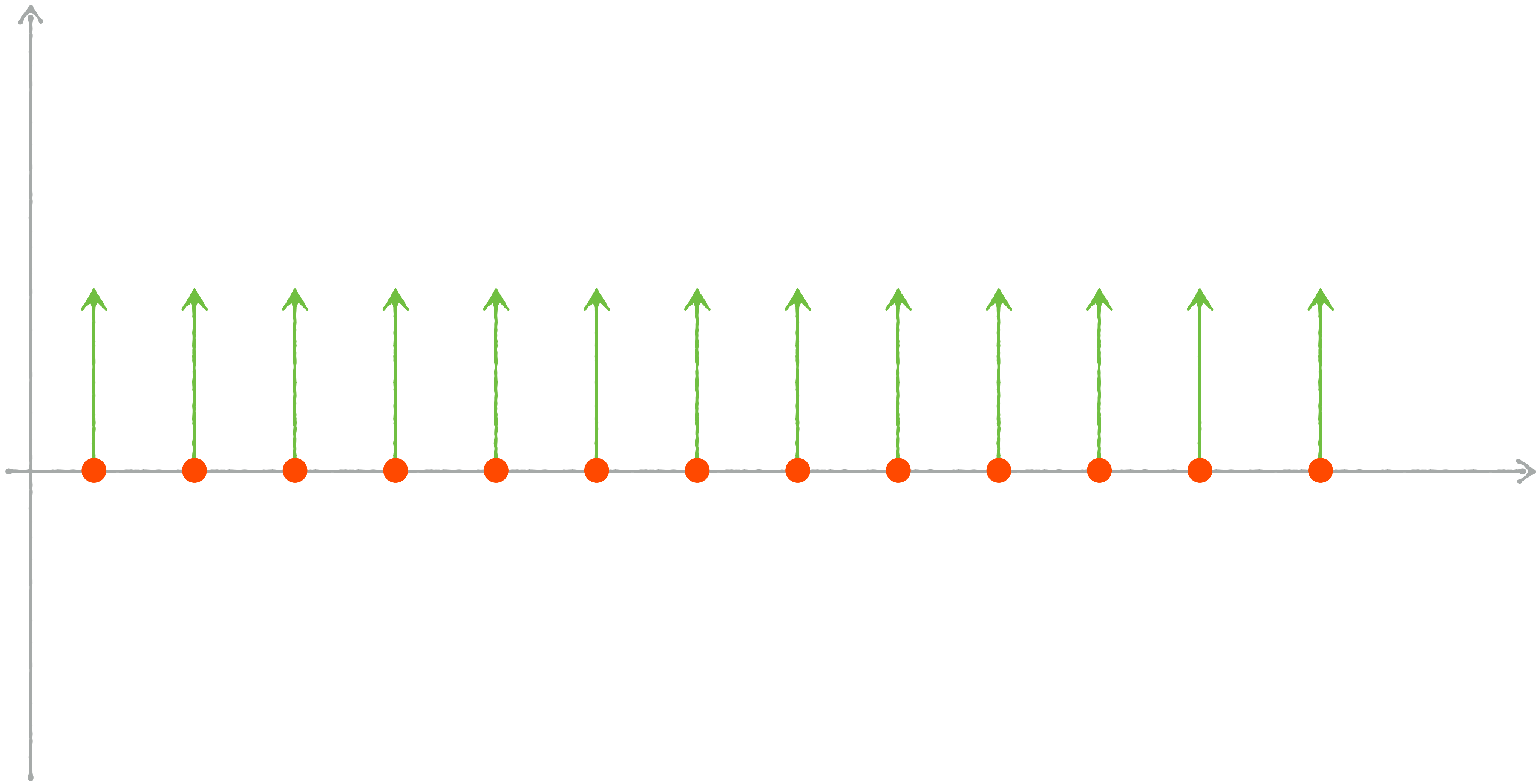


$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\,\underline{\xi'} \sum_j \tilde{w}_j \underline{\frac{\partial L_i}{\partial \tilde{y}_j}}$$

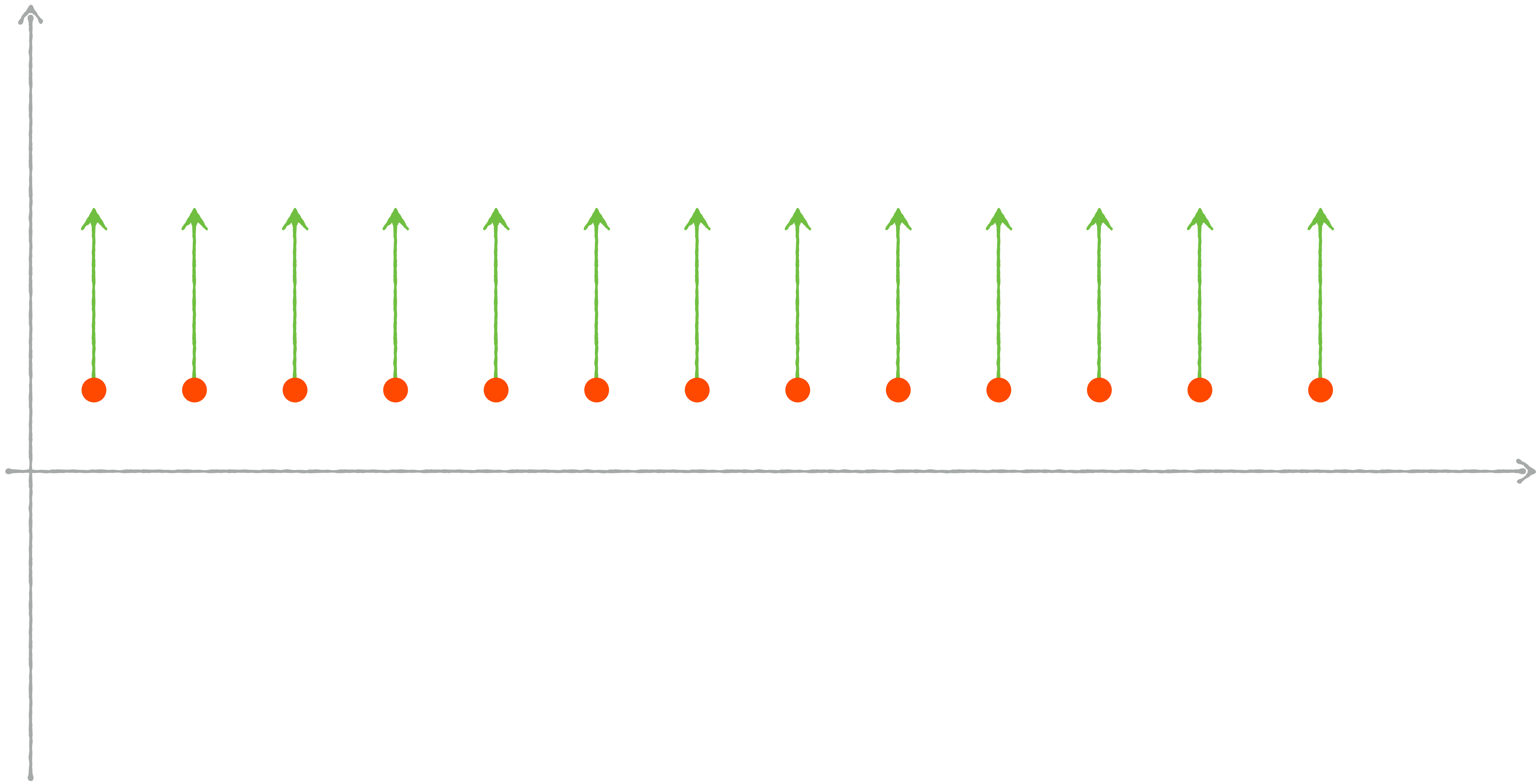
Weight Initialization



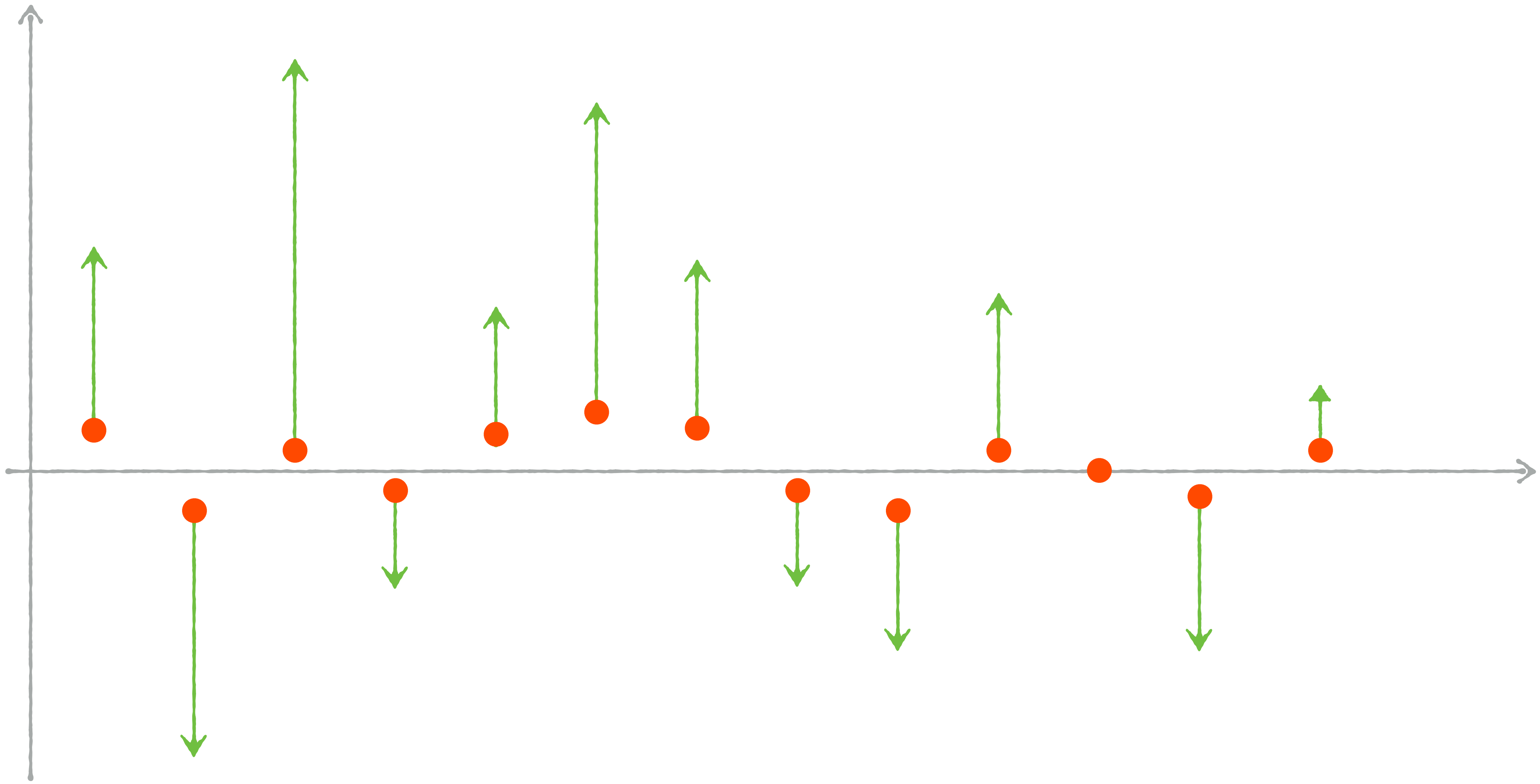
Weight Initialization



Weight Initialization

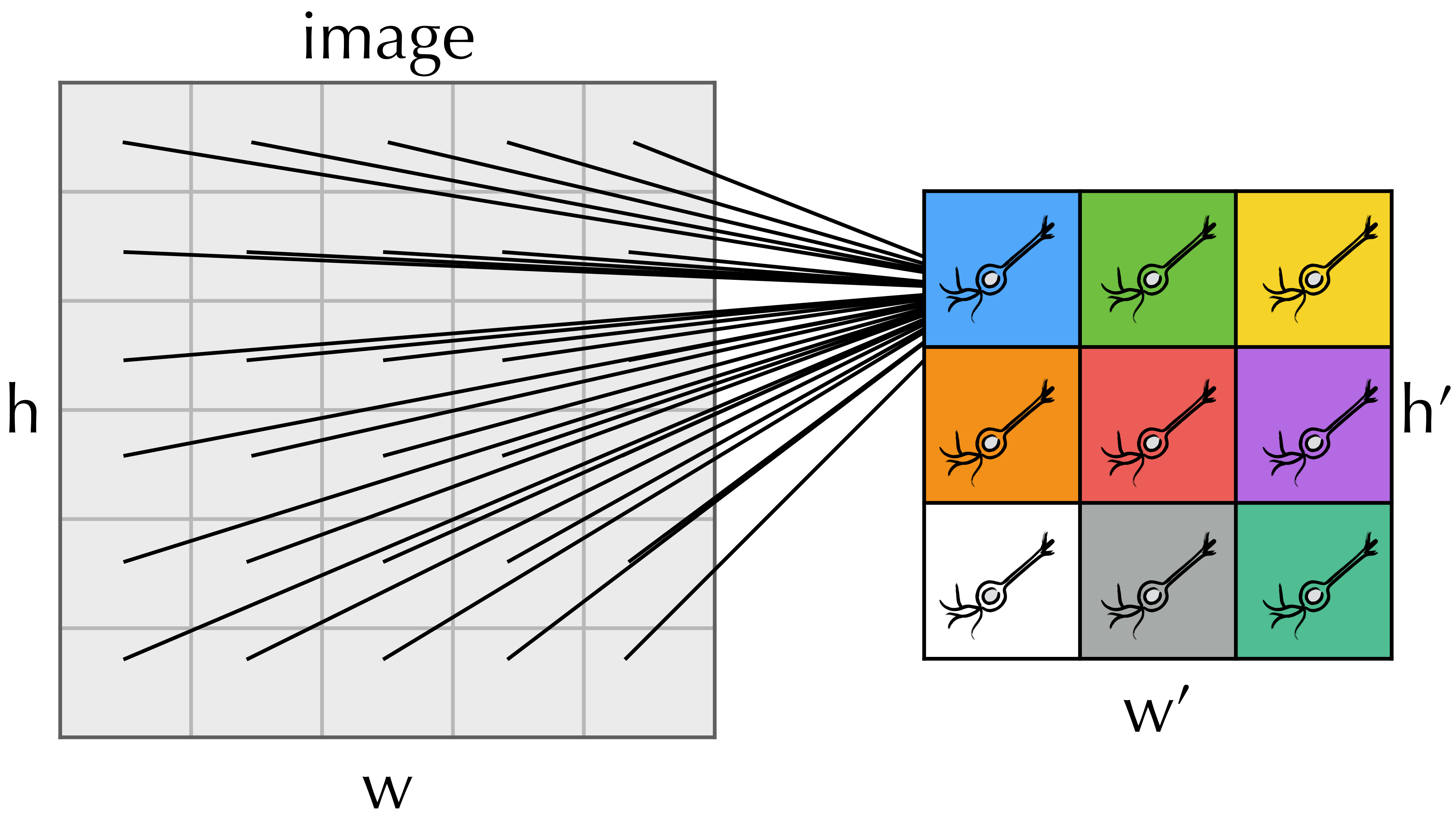


Weight Initialization

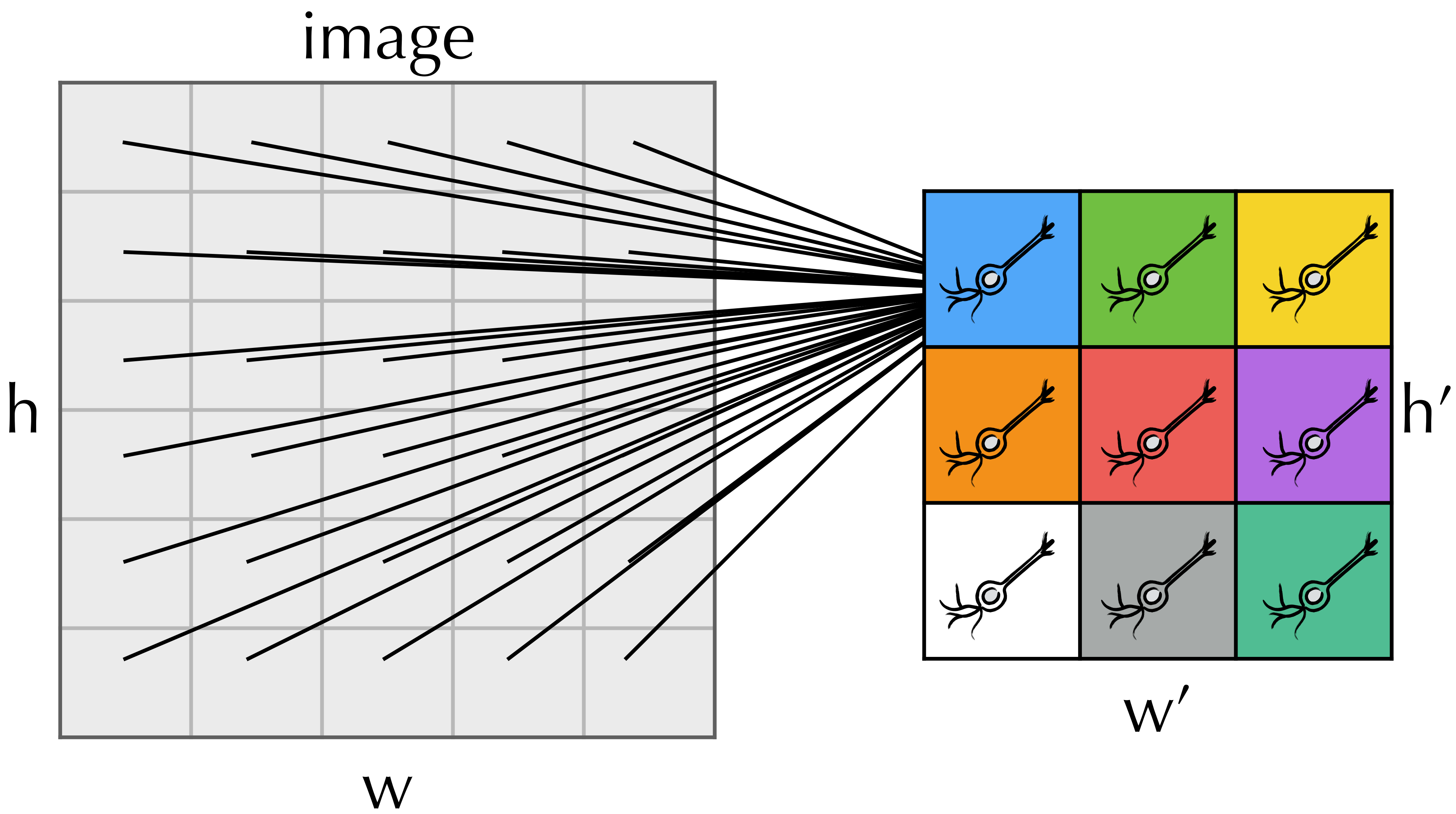


Convolutional Neural Networks

Fully Connected Layers

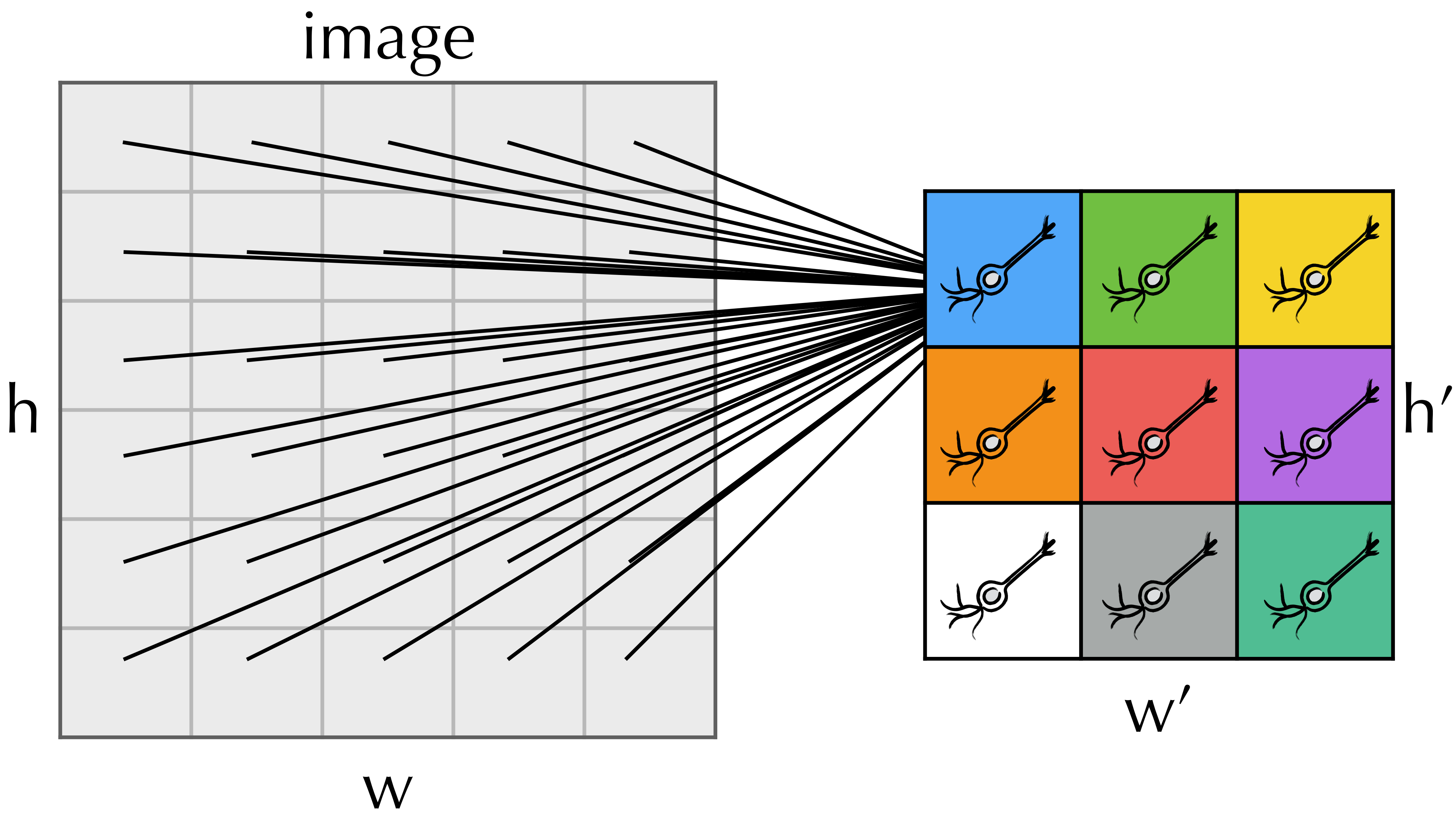


Fully Connected Layers



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

Fully Connected Layers

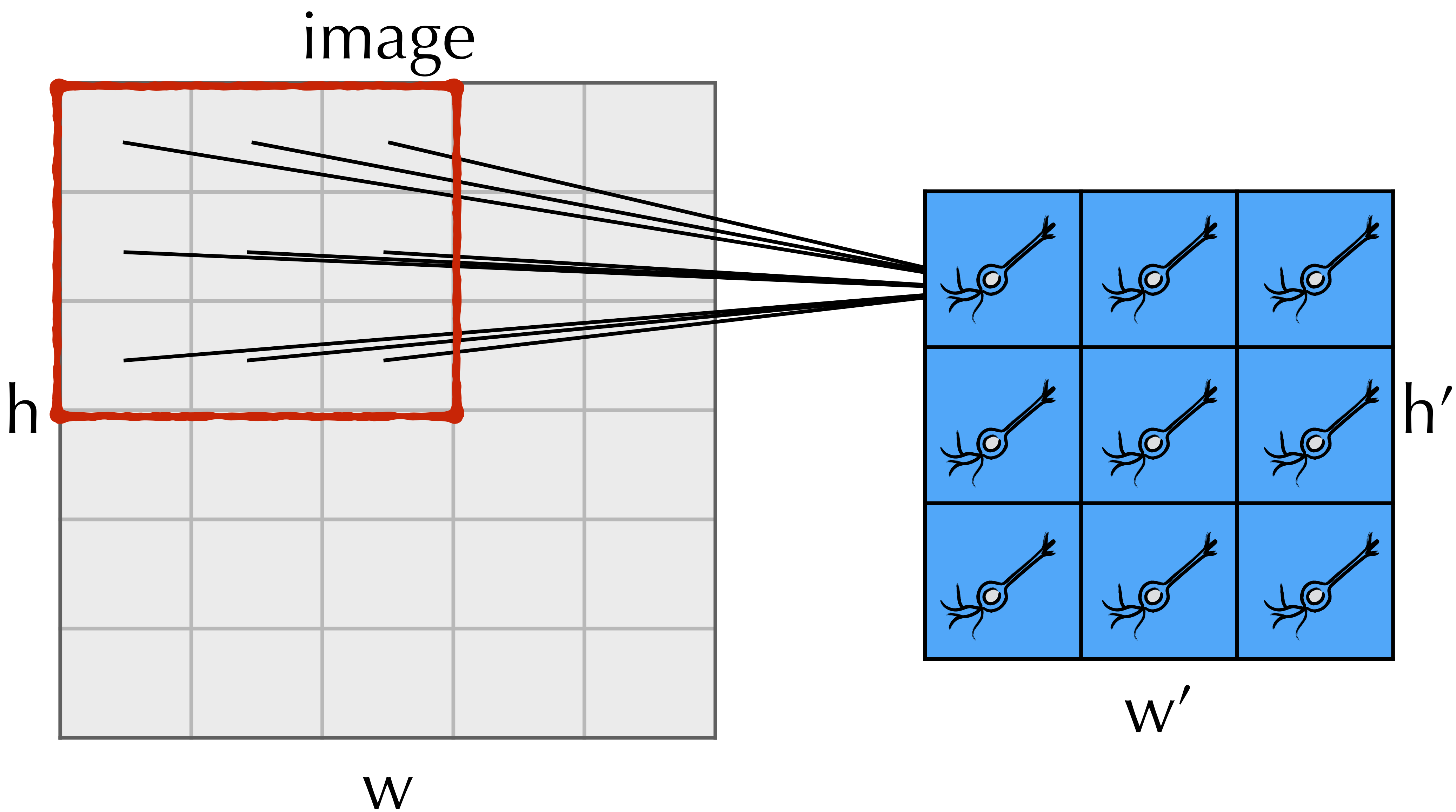


Number parameters: $w \cdot h \cdot w' \cdot h'$

Redundancy

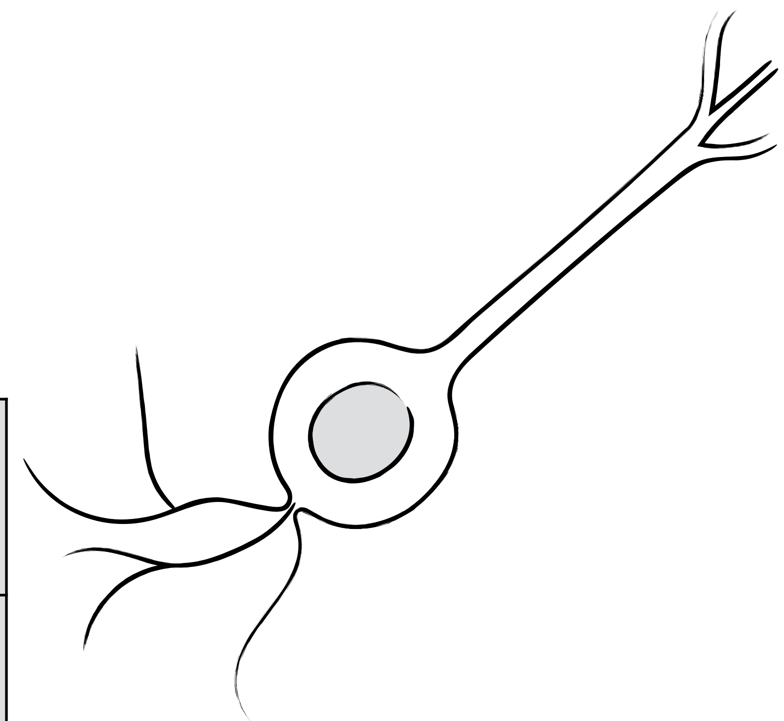
But no locality

Convolutional Layers



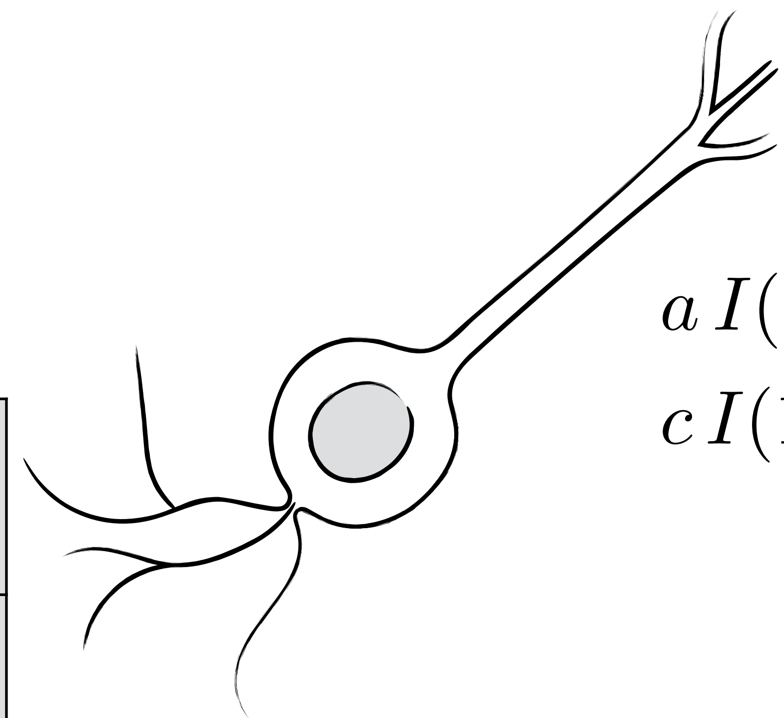
Translation Invariance

2	2	1	2	3
11	21	22	21	14
12	20	45	32	21
11	12	11	16	21
21	22	23	25	35



Translation Invariance

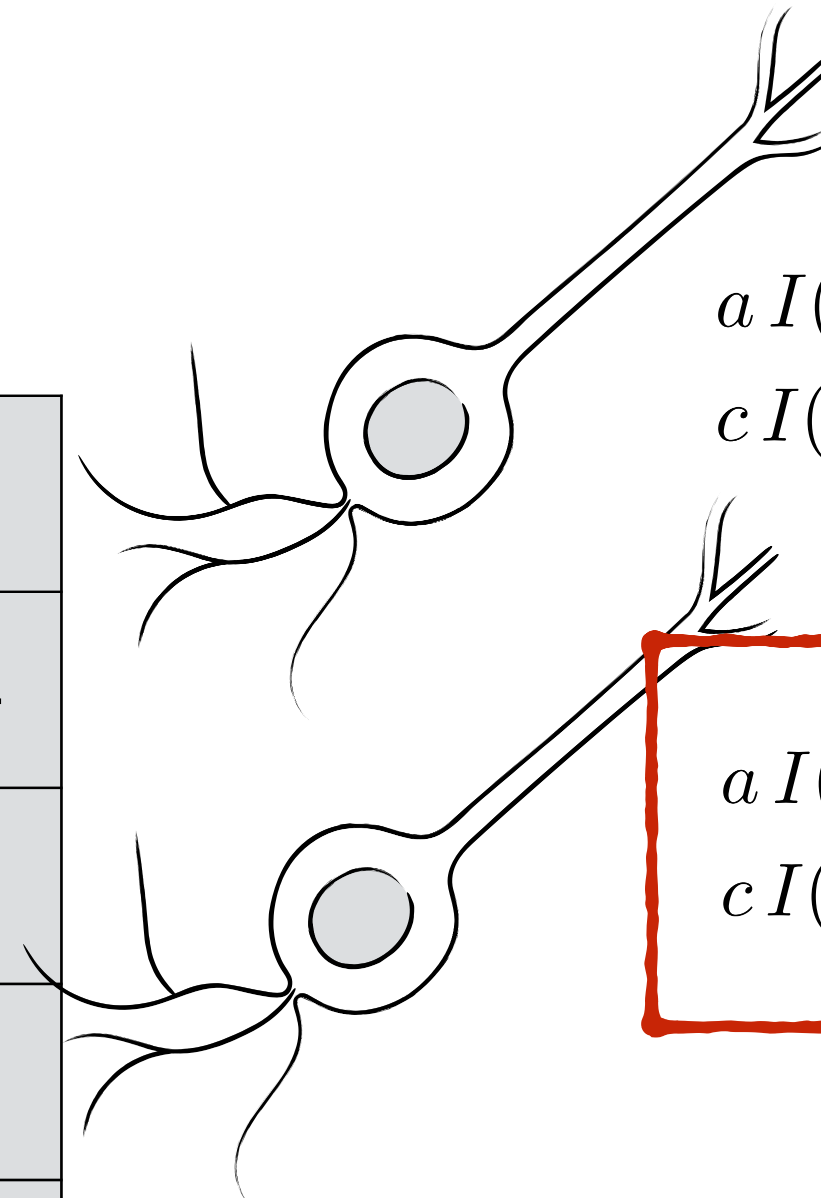
2	2	1	2	3
11	21	22	21	14
12	20	45	32	21
11	12	11	16	21
21	22	23	25	35



$$a\,I(1,1) + b\,I(2,1) +$$
$$c\,I(1,2) + d\,I(2,2)$$

Translation Invariance

2	2	1	2	3
11	21	22	21	14
12	20	45	32	21
11	12	11	16	21
21	22	23	25	35

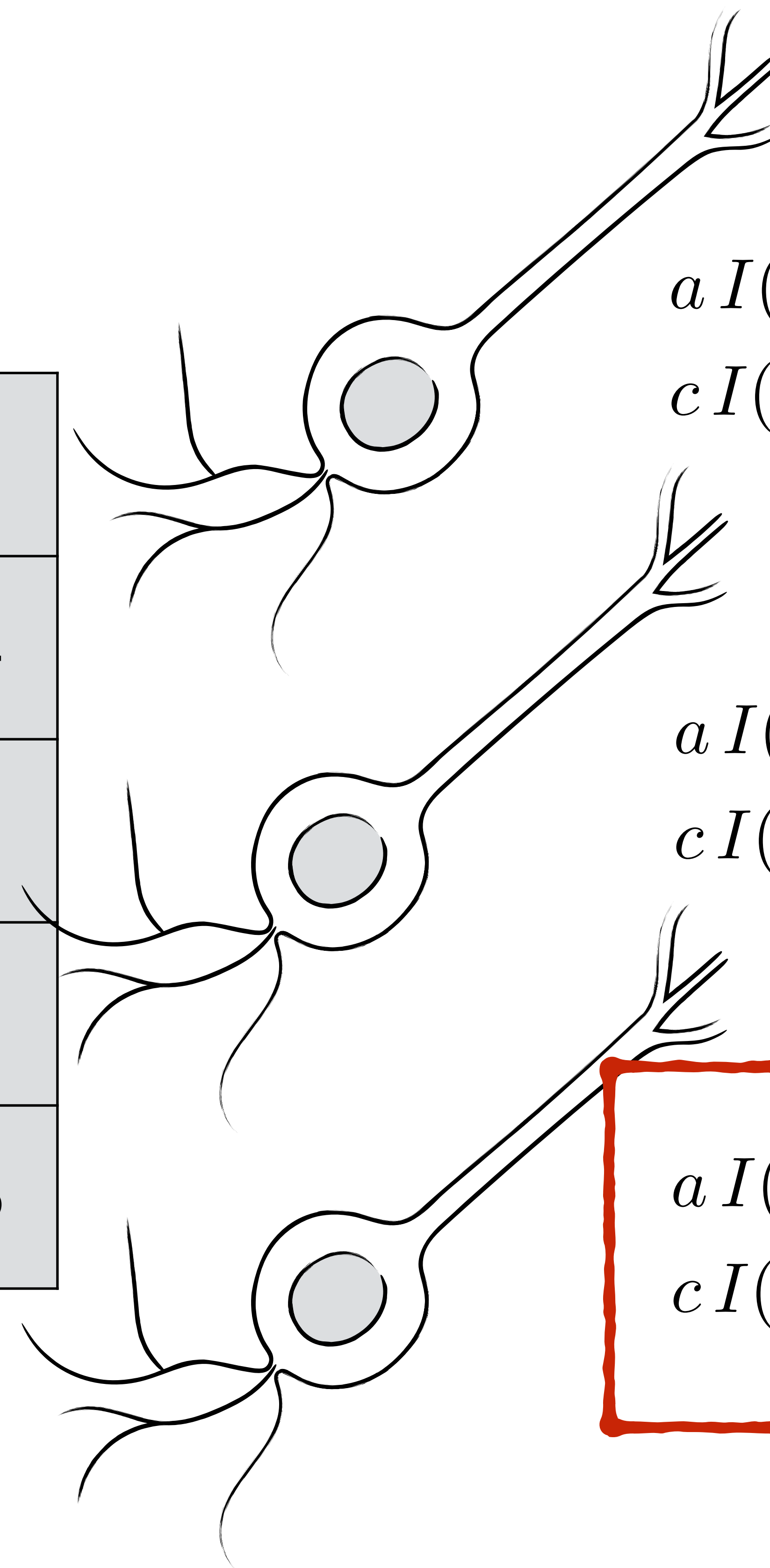


$$a\,I(1,1) + b\,I(2,1) + \\ c\,I(1,2) + d\,I(2,2)$$

$$\boxed{\begin{aligned} &a\,I(2,1) + b\,I(3,1) + \\ &c\,I(2,2) + d\,I(3,2) \end{aligned}}$$

Translation Invariance

2	2	1	2	3
11	21	22	21	14
12	20	45	32	21
11	12	11	16	21
21	22	23	25	35

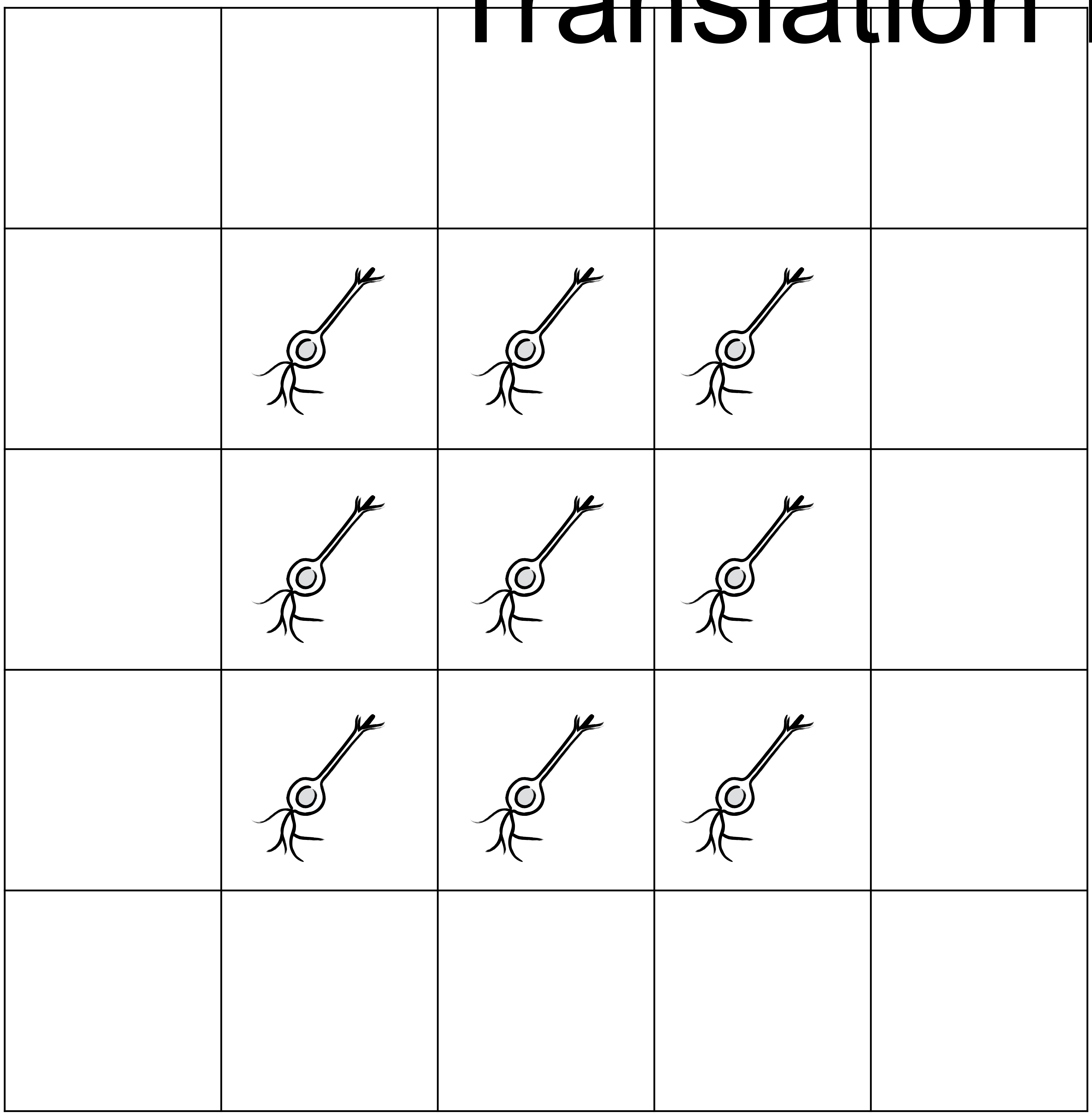


$$a\,I(1,1) + b\,I(2,1) + c\,I(1,2) + d\,I(2,2)$$

$$a\,I(2,1) + b\,I(3,1) + c\,I(2,2) + d\,I(3,2)$$

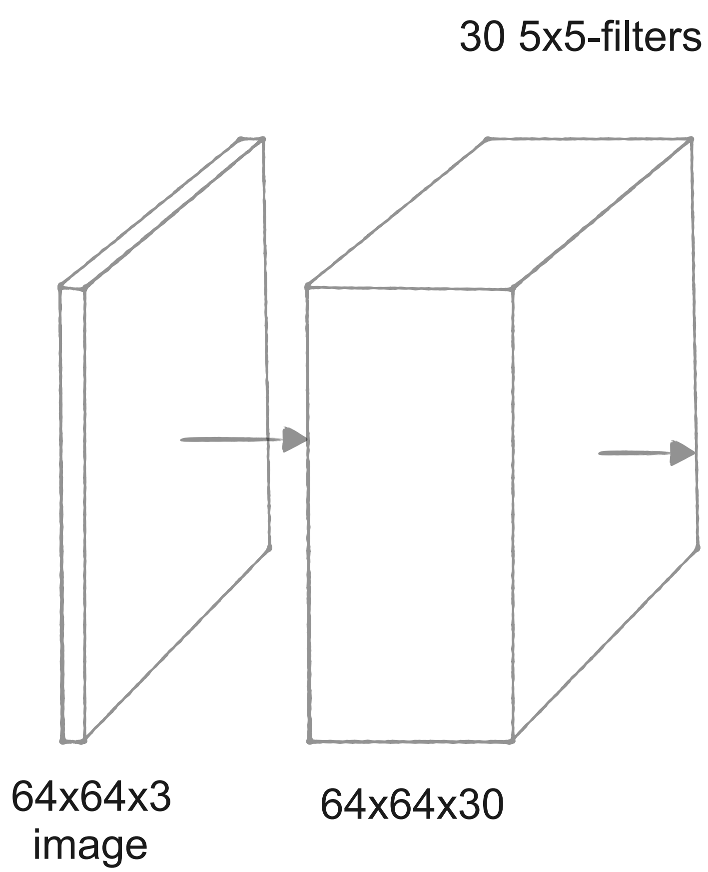
$$\boxed{a\,I(3,1) + b\,I(4,1) + c\,I(3,2) + d\,I(4,2)}$$

Translation Invariance



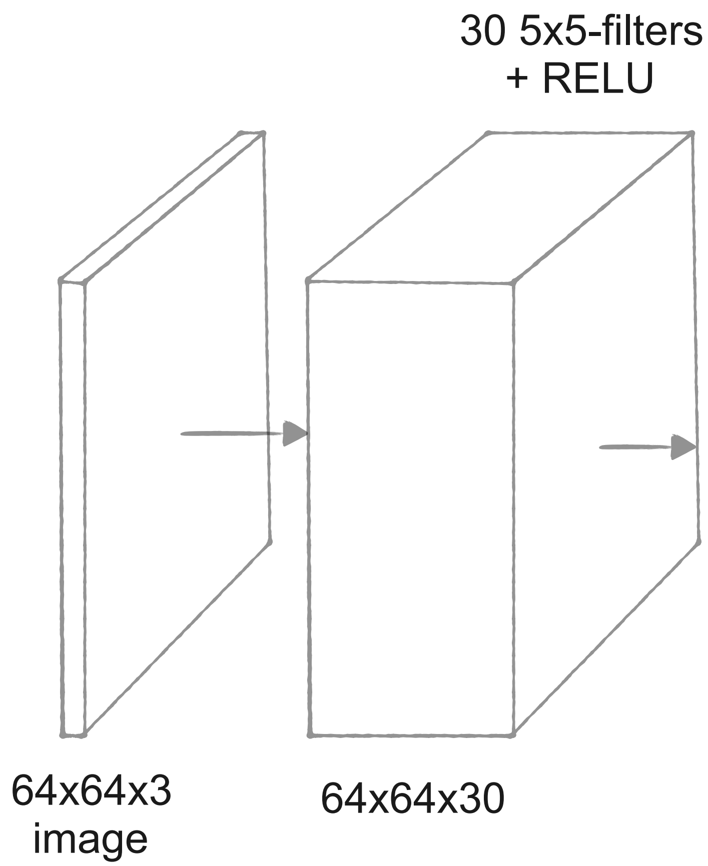
$$I \star \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Network Structure



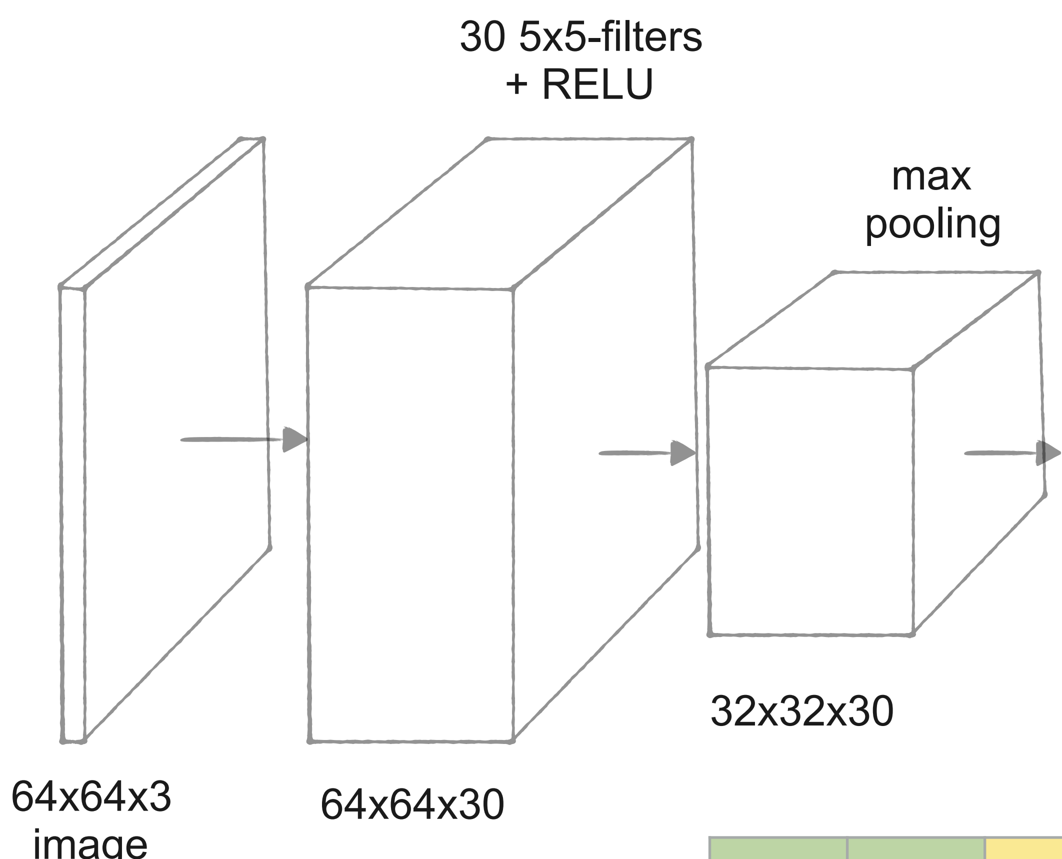
$$I \star w$$

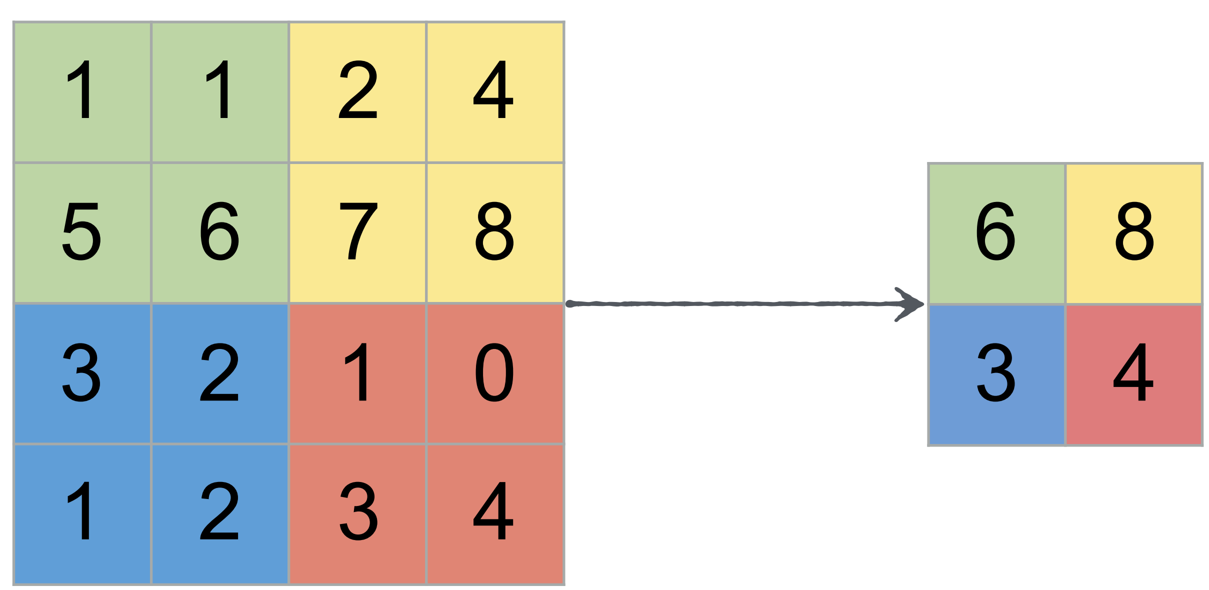
Network Structure



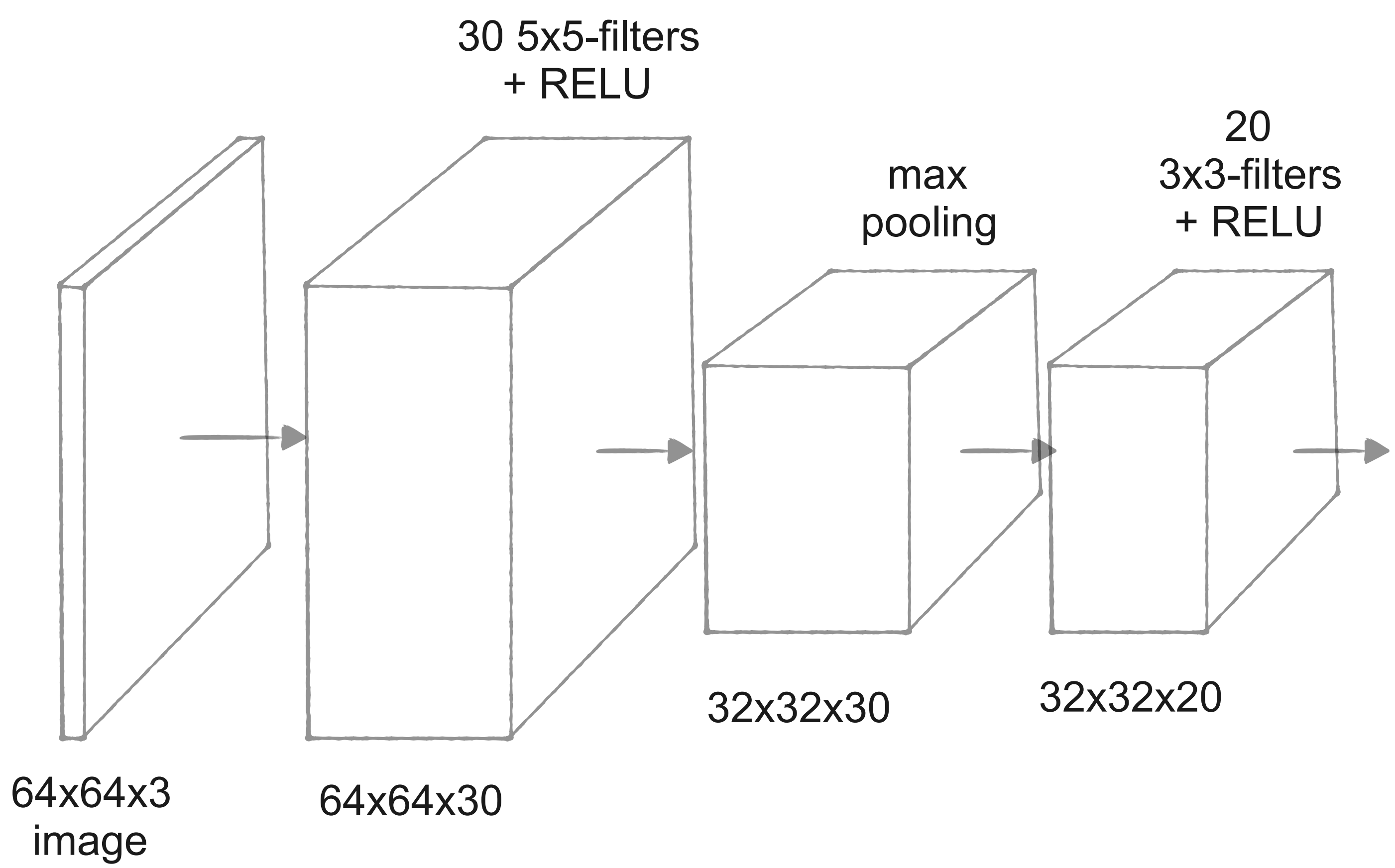
$$\max\{I \star w, 0\}$$

Network Structure

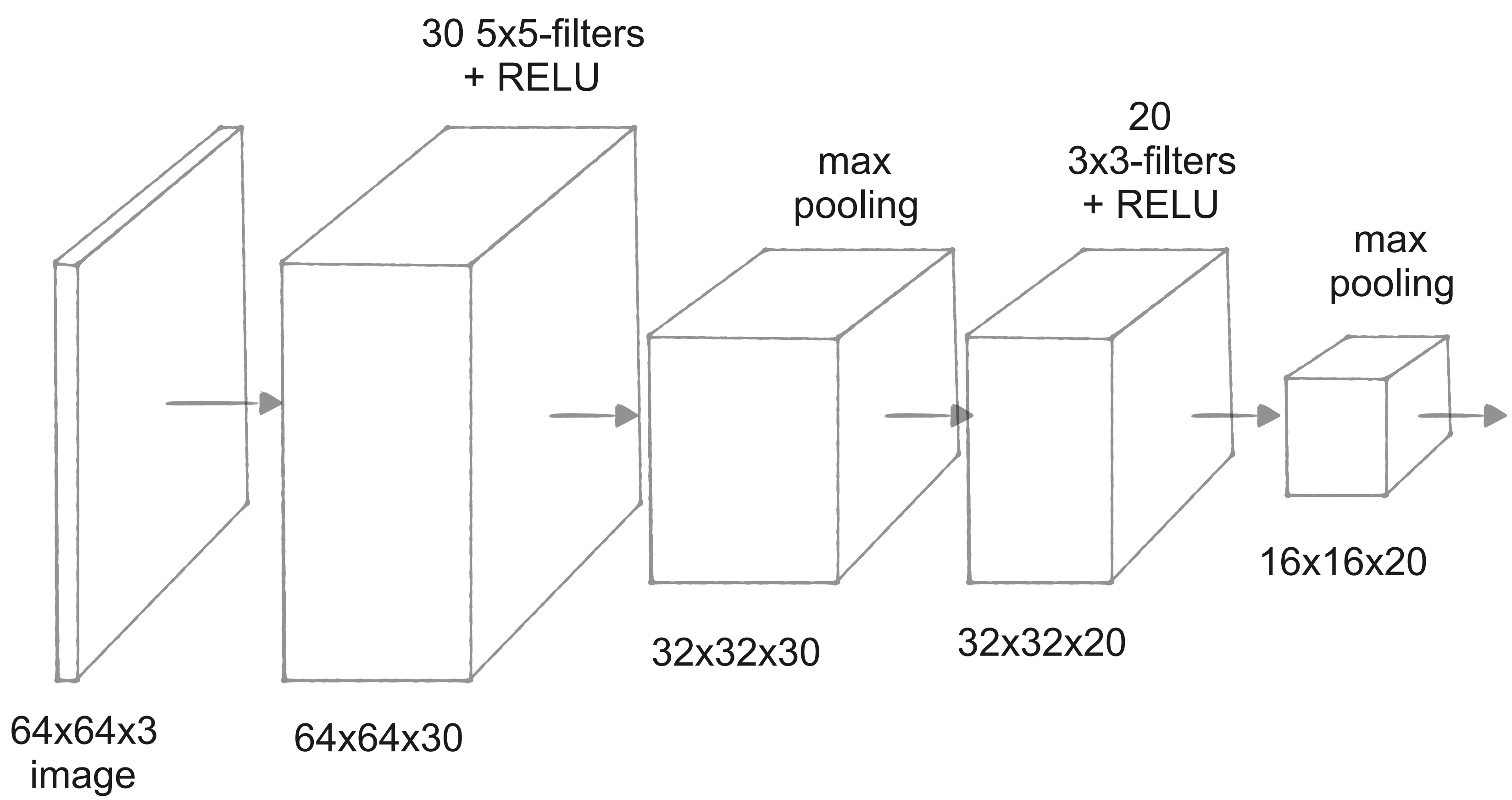




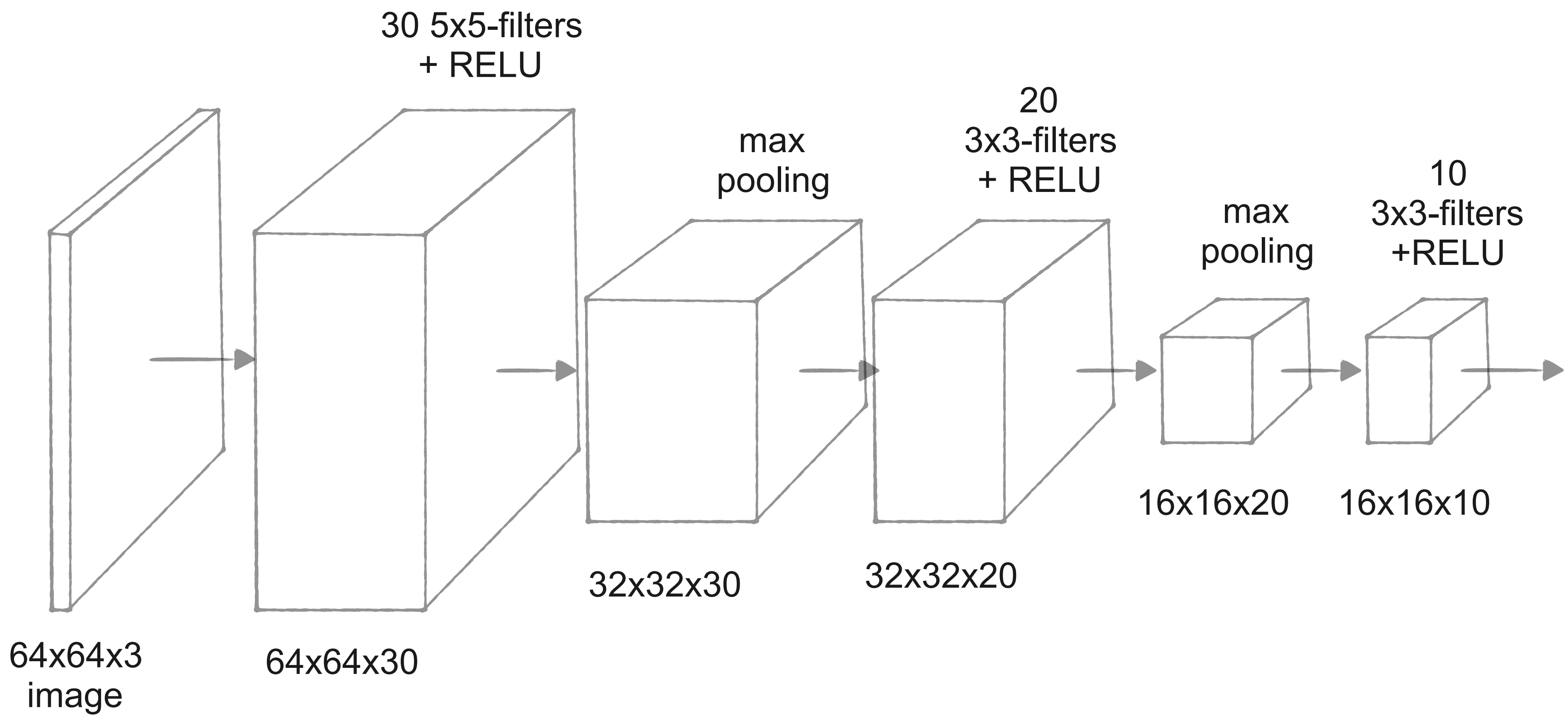
Network Structure



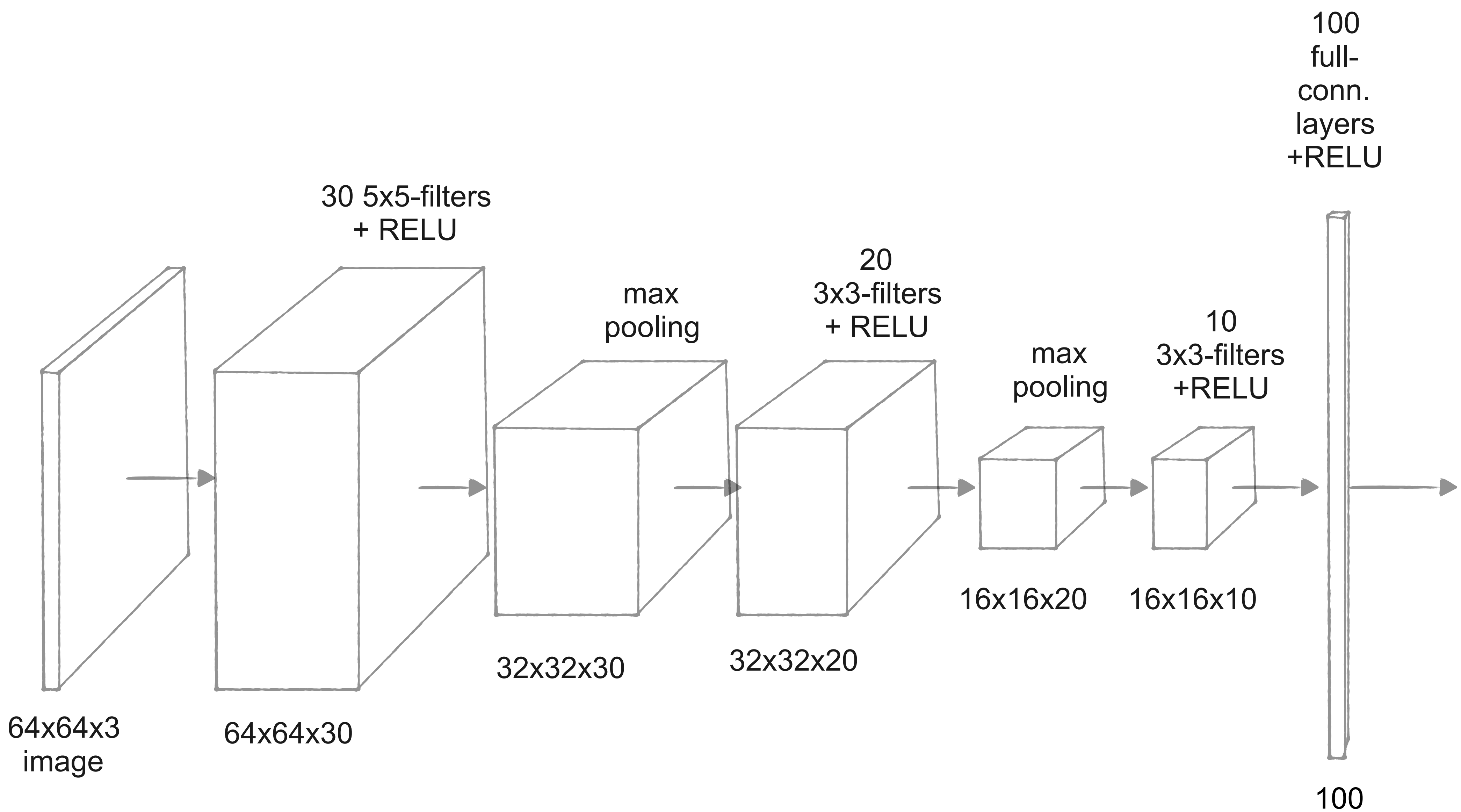
Network Structure



Network Structure

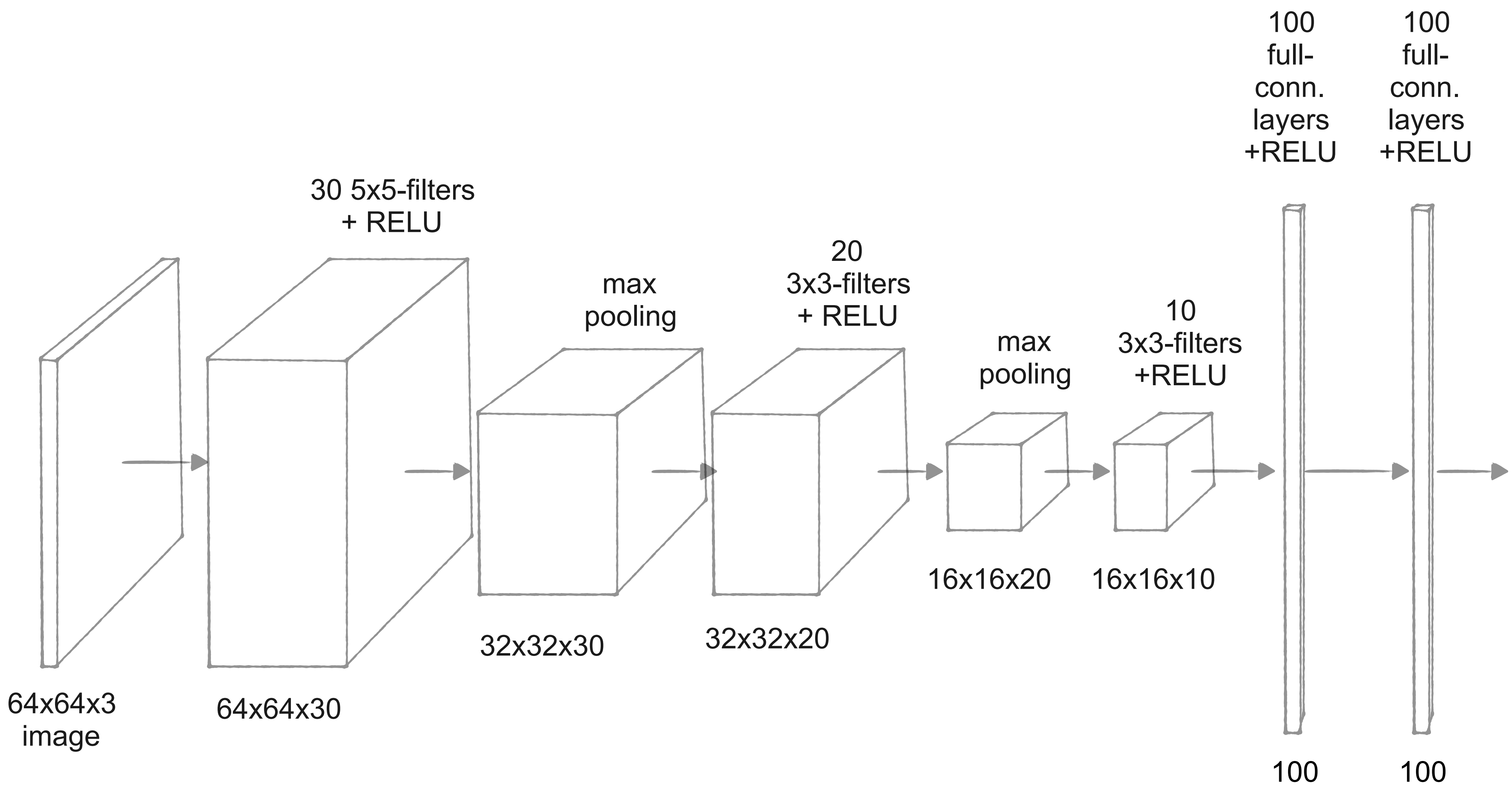


Network Structure

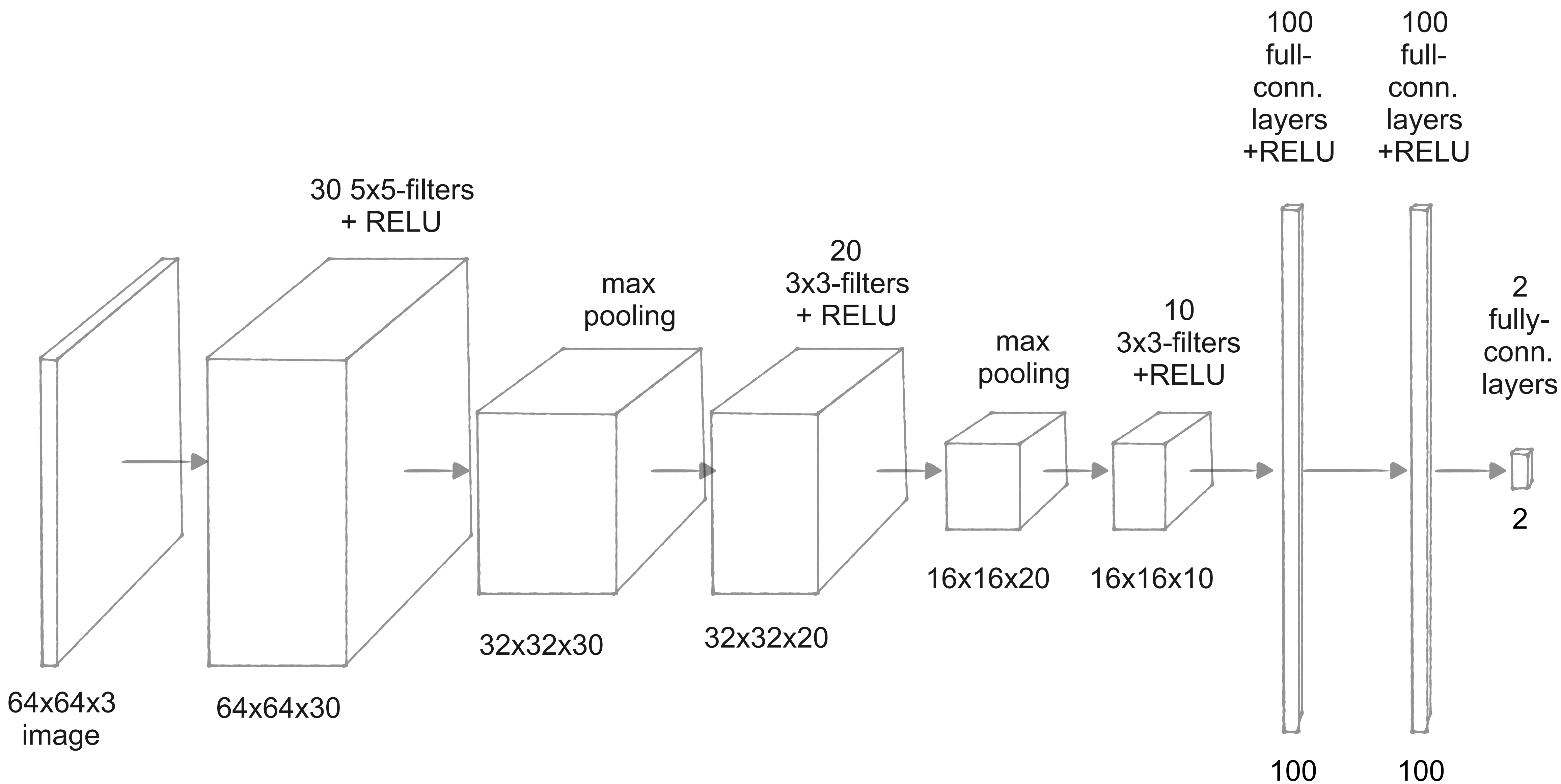


$$\sum_{u=1}^{16}\sum_{v=1}^{16}\sum_{i=1}^{10} w_{uvi}x_{uvi}$$

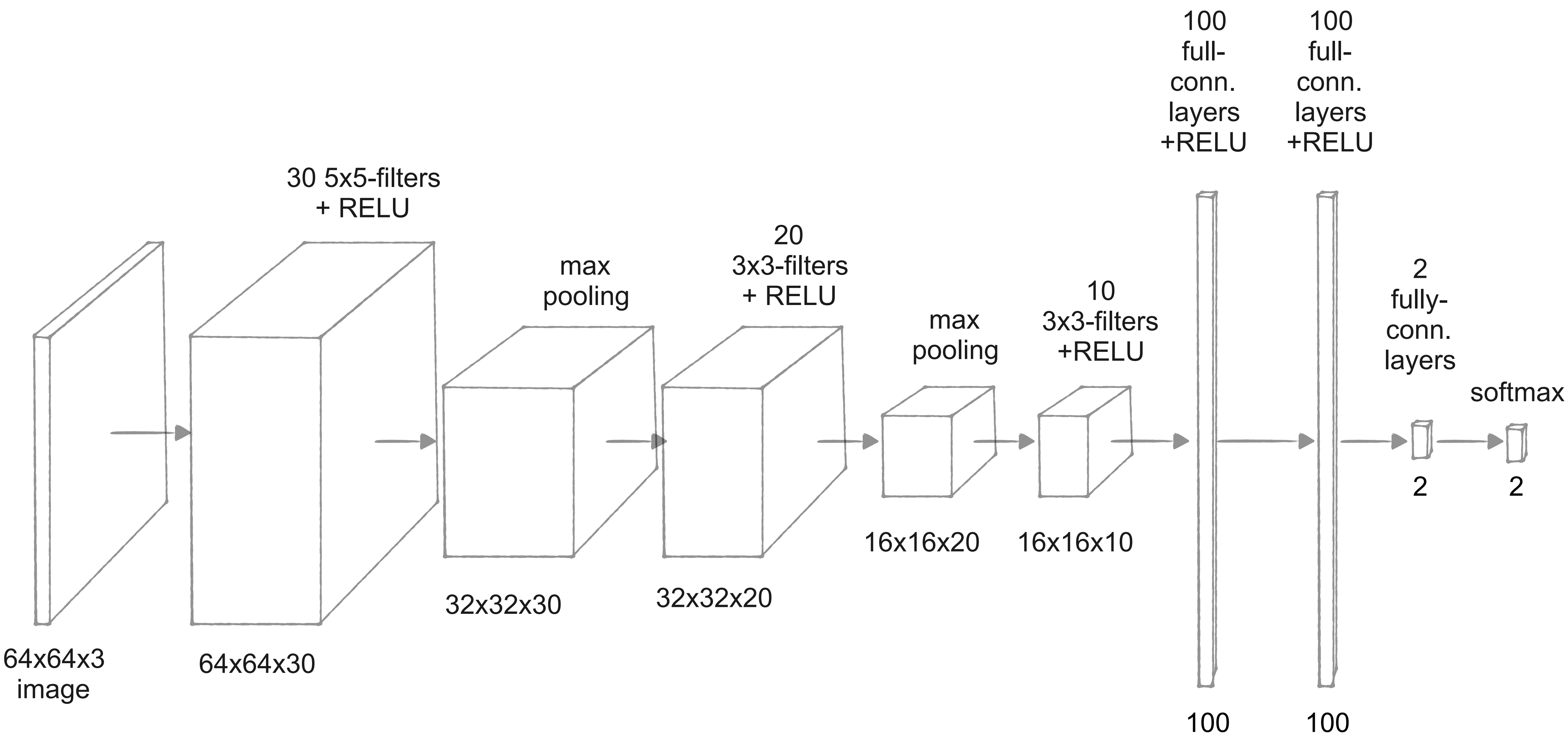
Network Structure



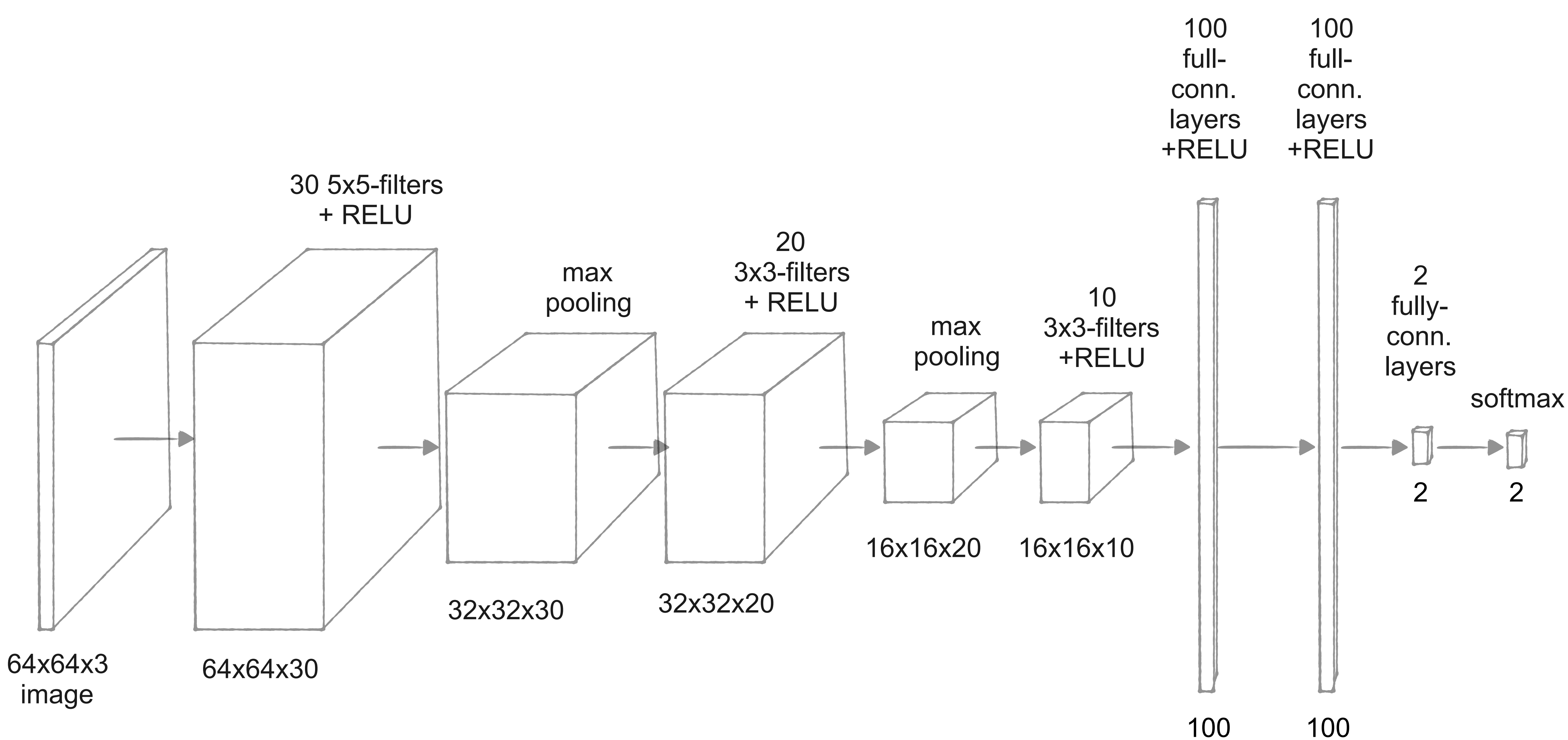
Network Structure



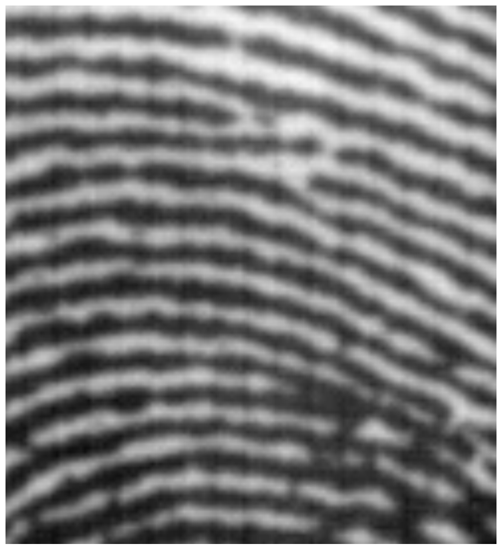
Network Structure

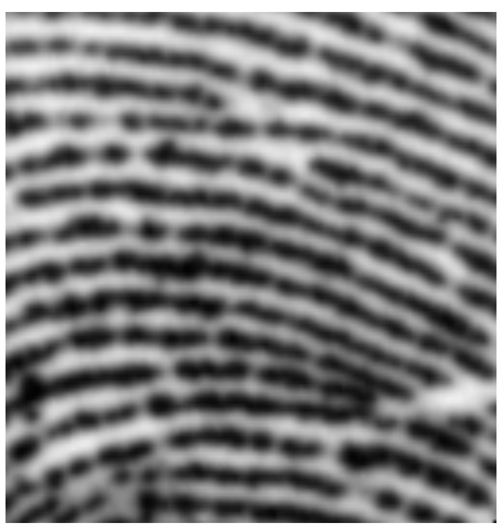


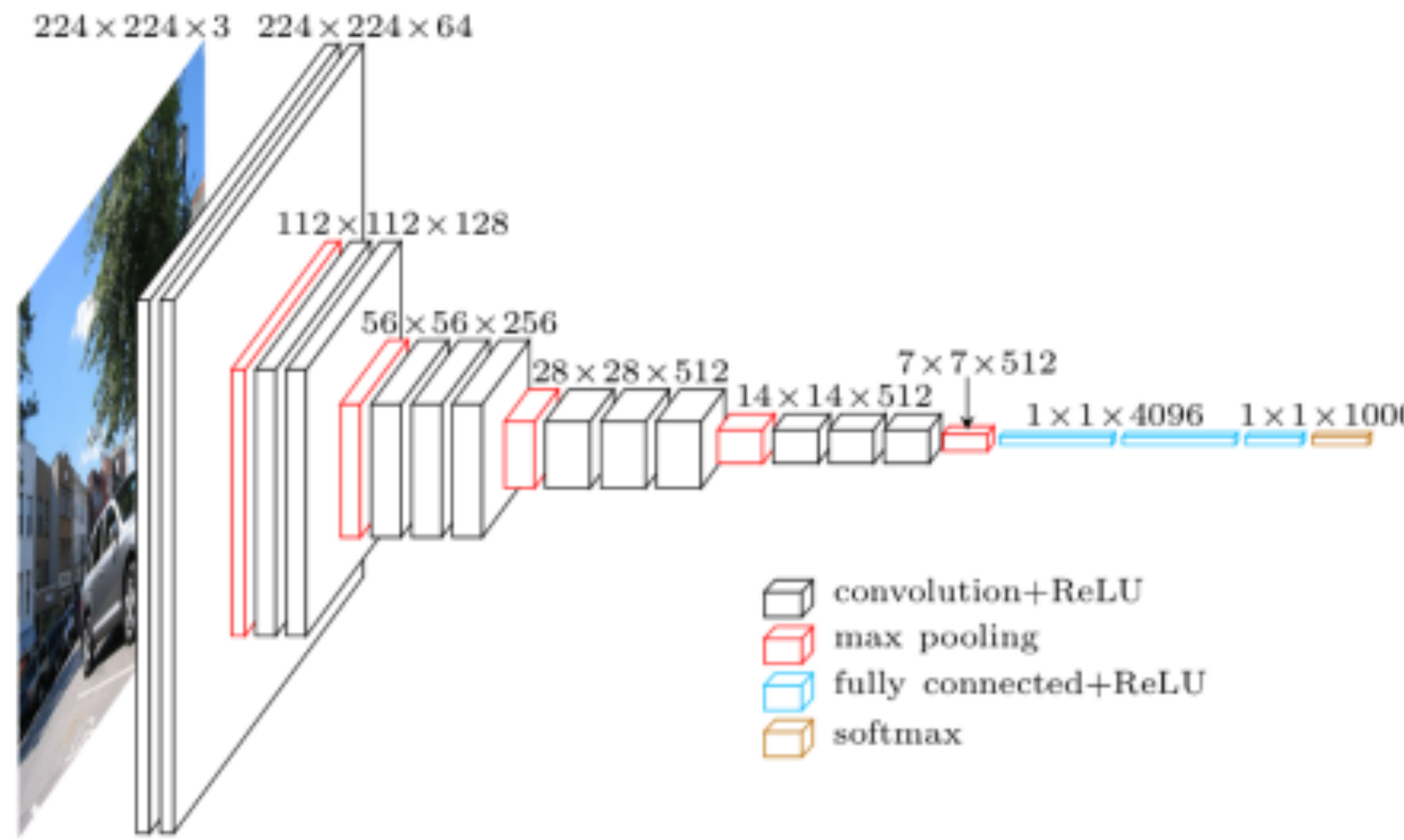
Number Parameters?



Live or Spoof?





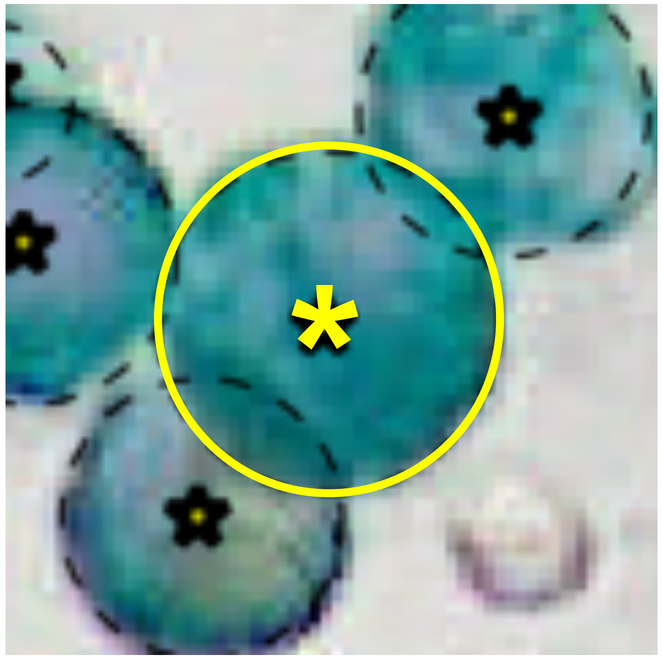


VGG-16

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

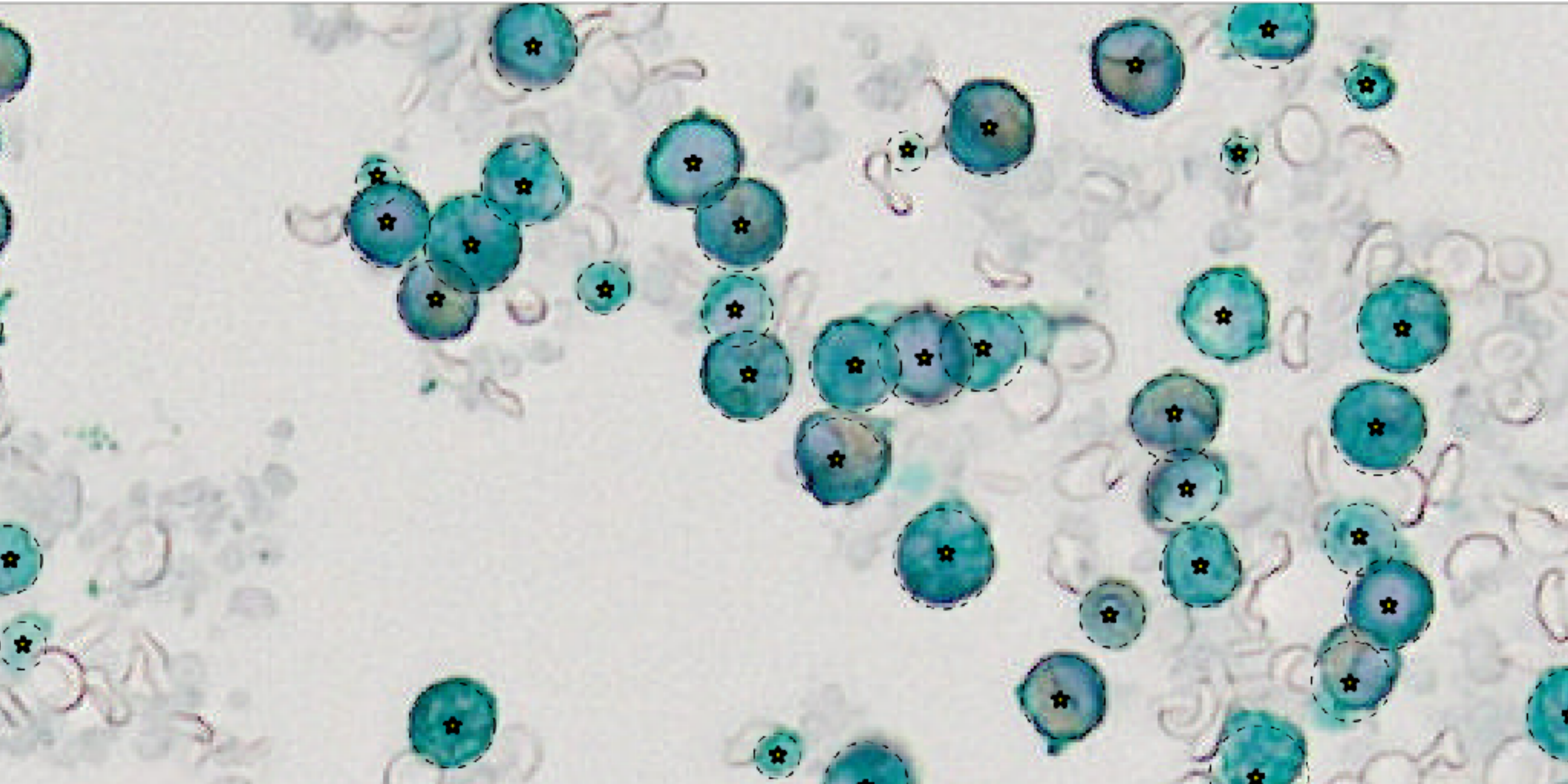
Centre and Radius

- No softmax
- Quadratic loss function

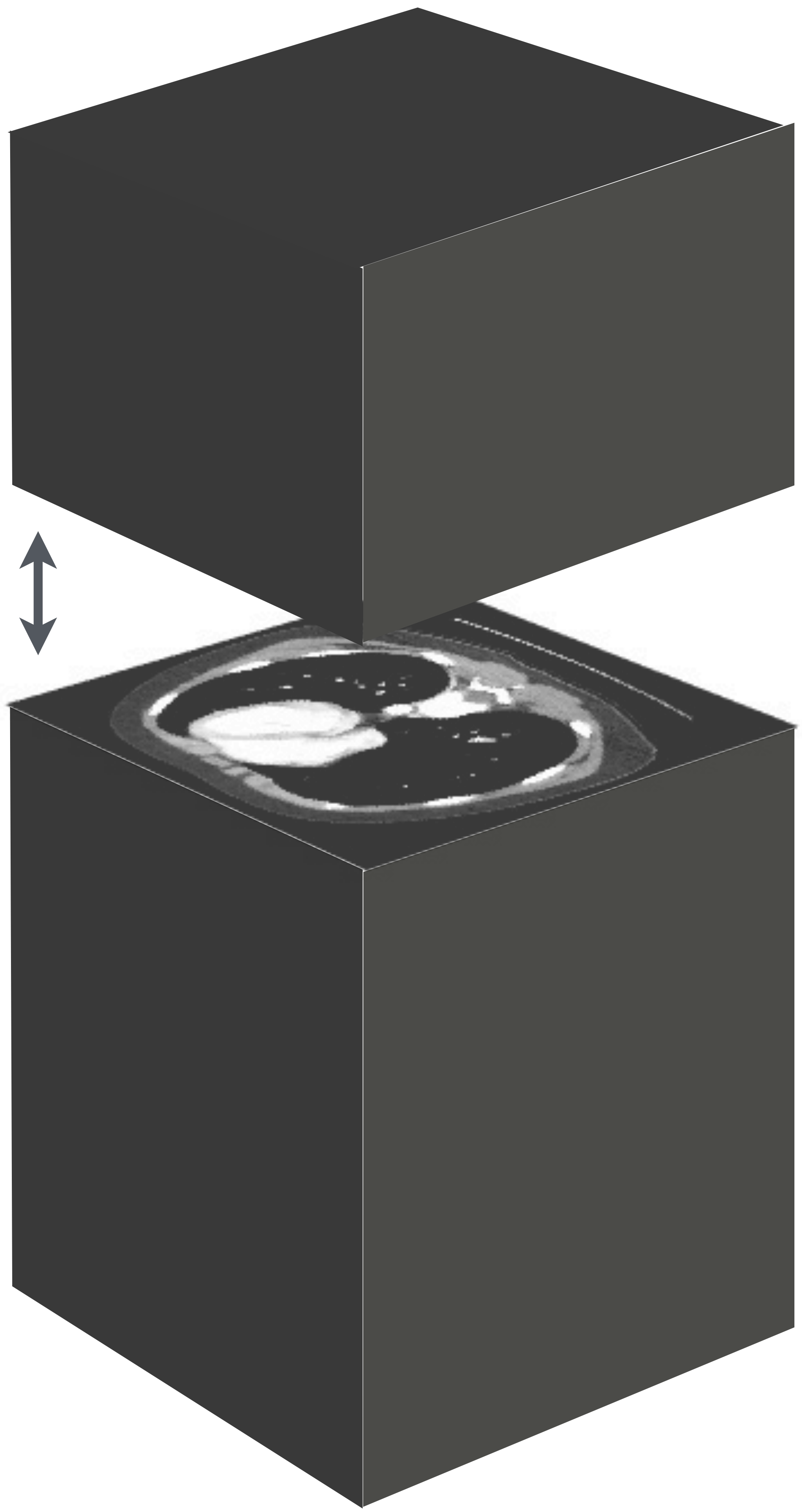


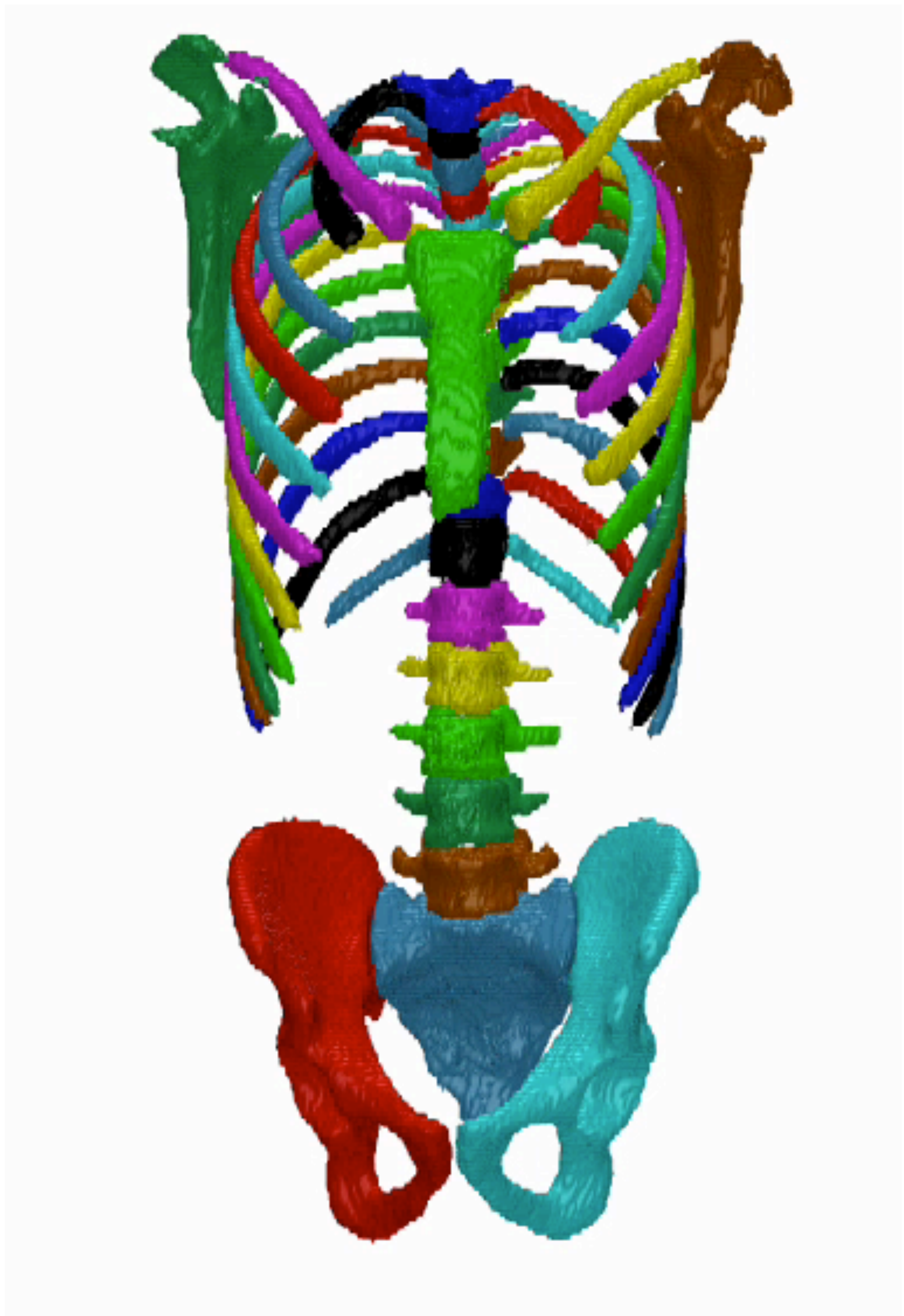
Centre and Radius

- No softmax
- Quadratic loss function

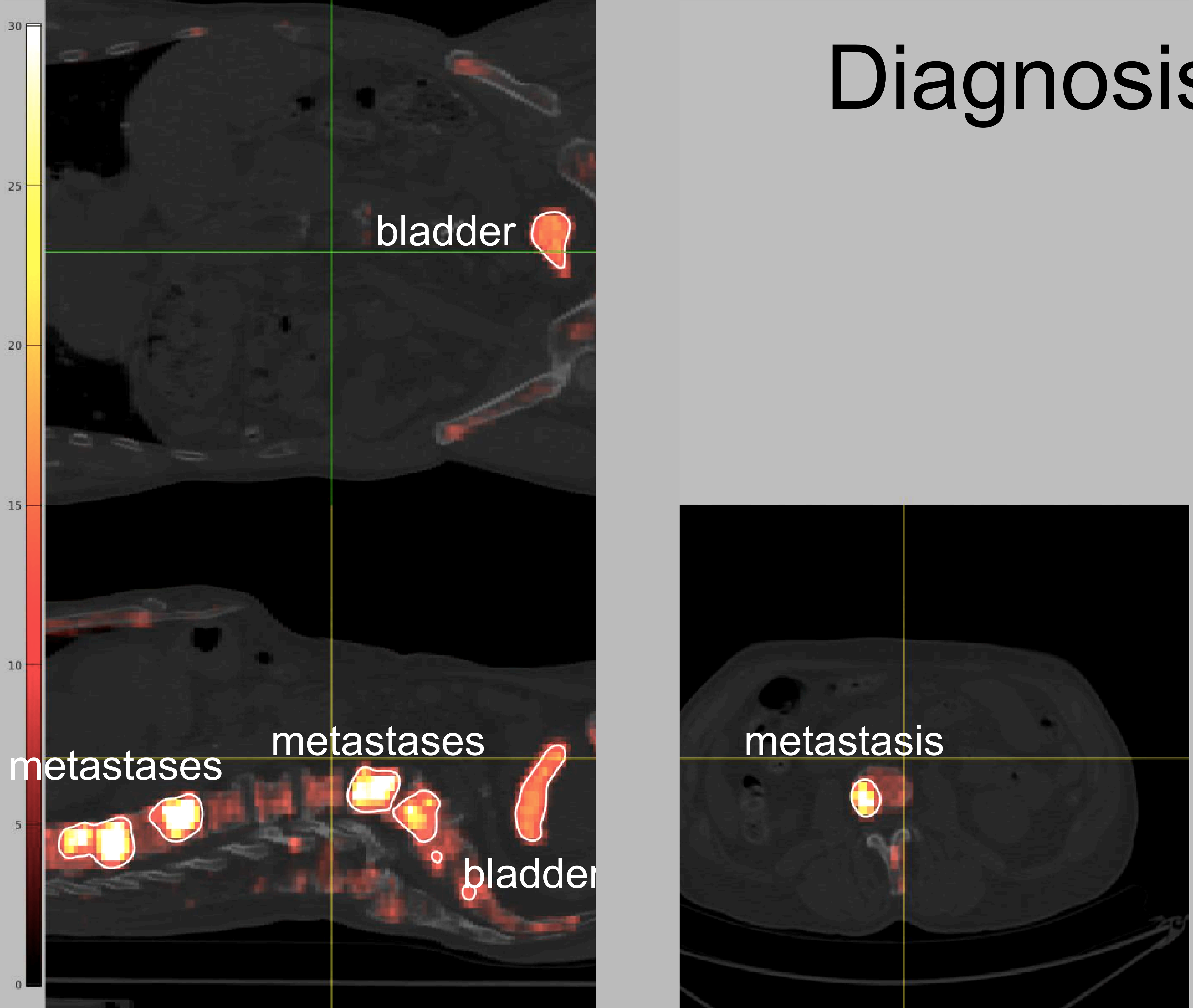


Segmentation





Diagnosis

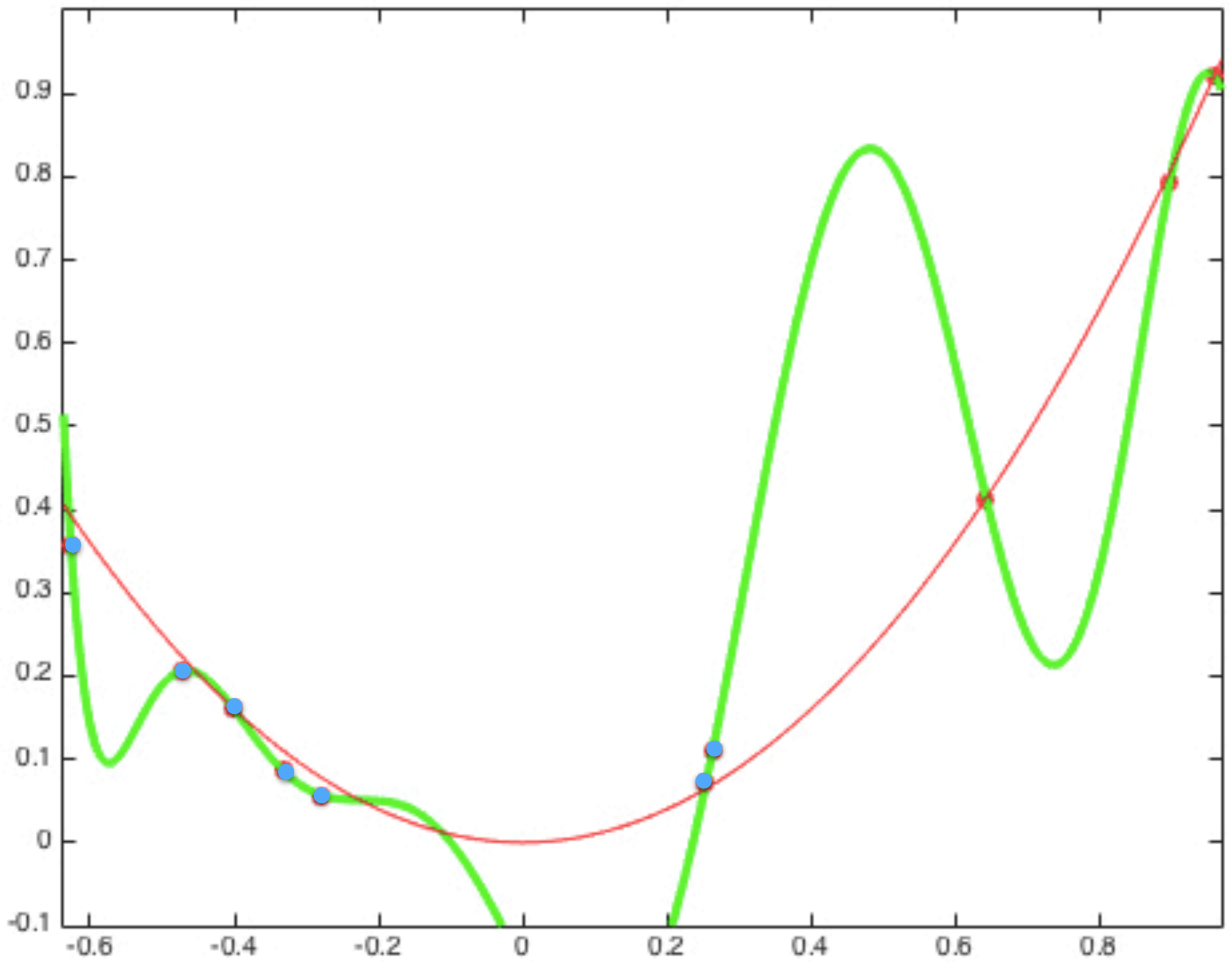


30
25
20
15
10
5
0
bladder
metastases
metastastases
bladder
metastasis

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 1**
Conv 128 3x3	Conv 128 3x3	max pooling	**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

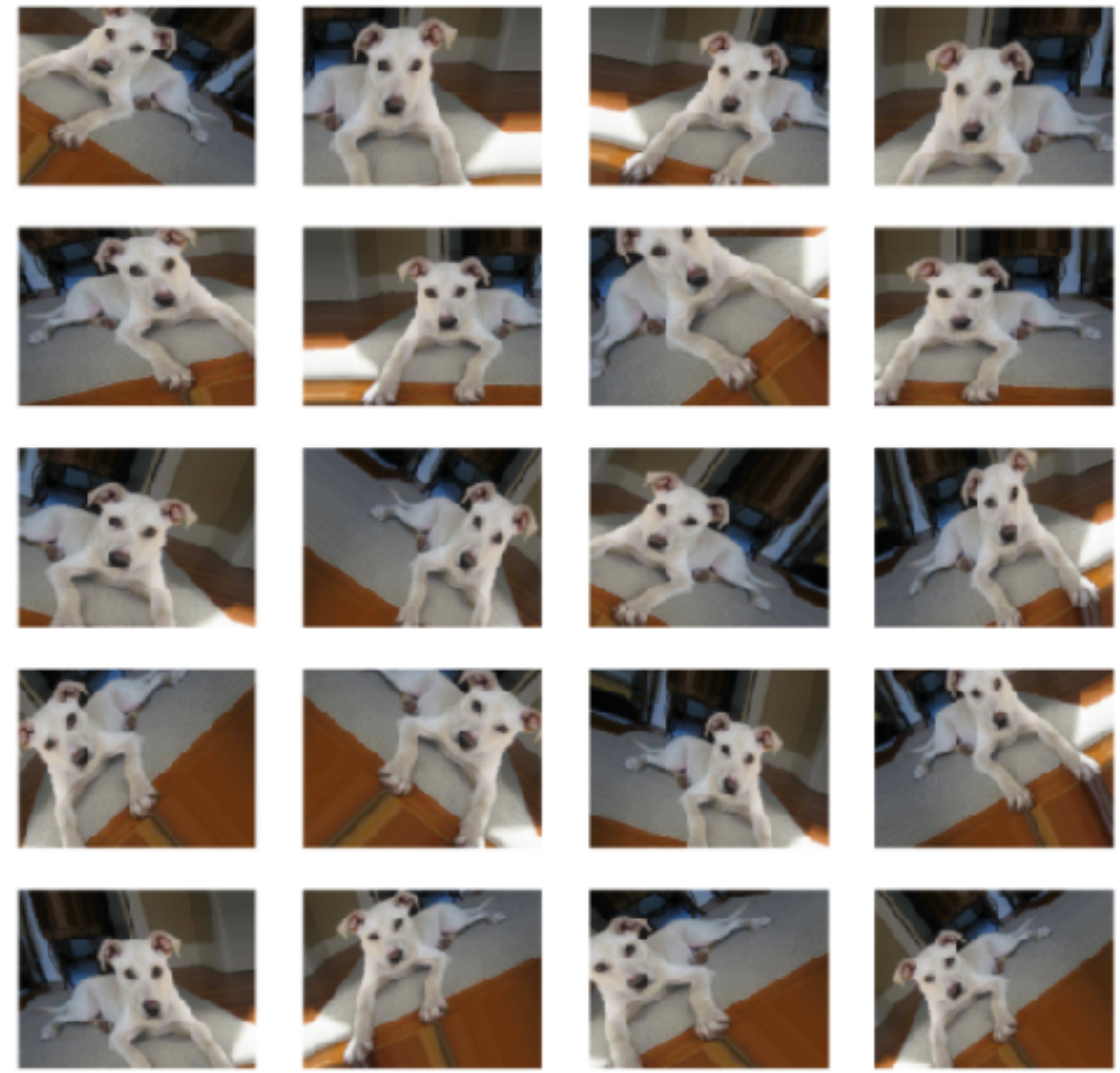
224

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	

138M parameters!

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

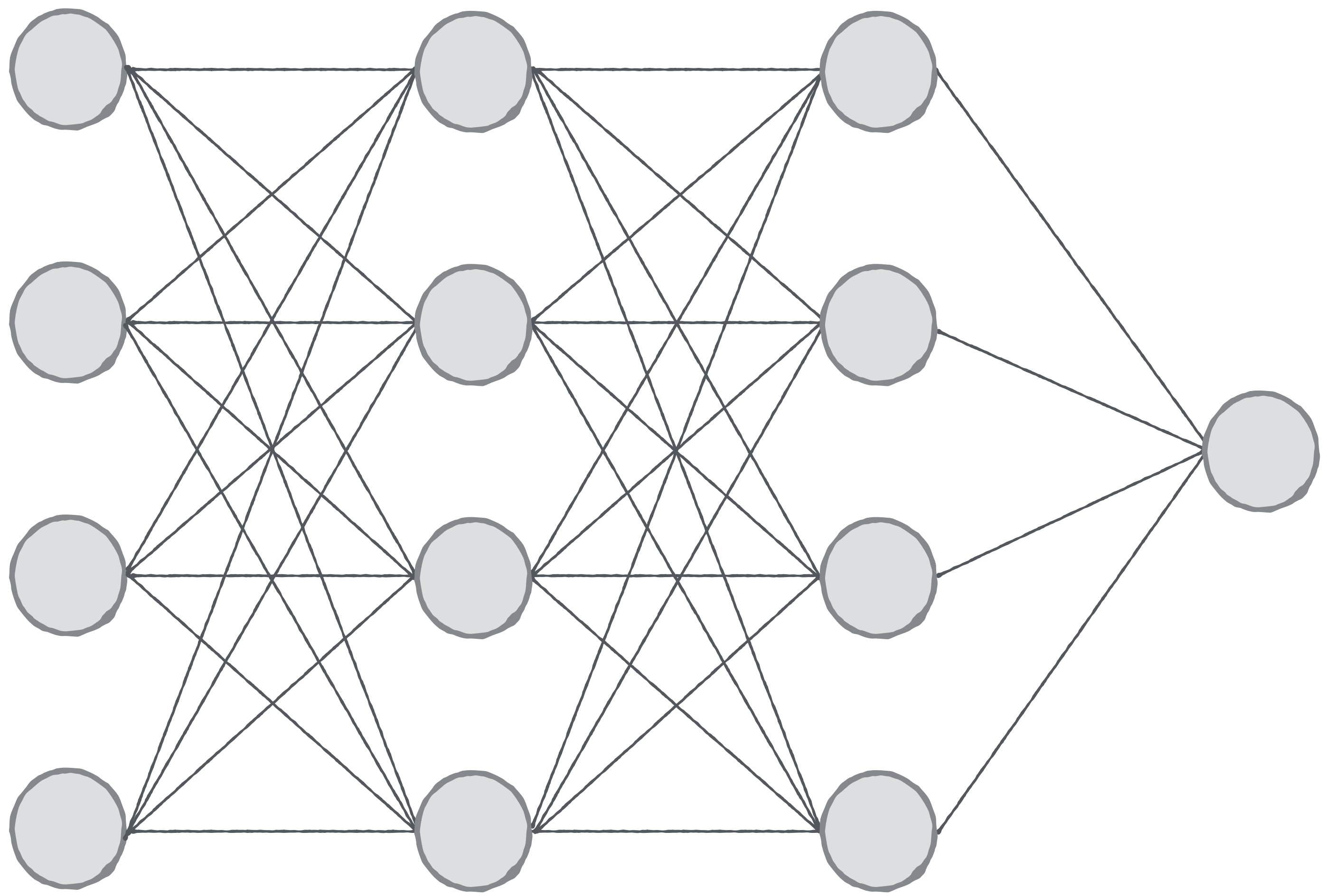
Data Augmentation



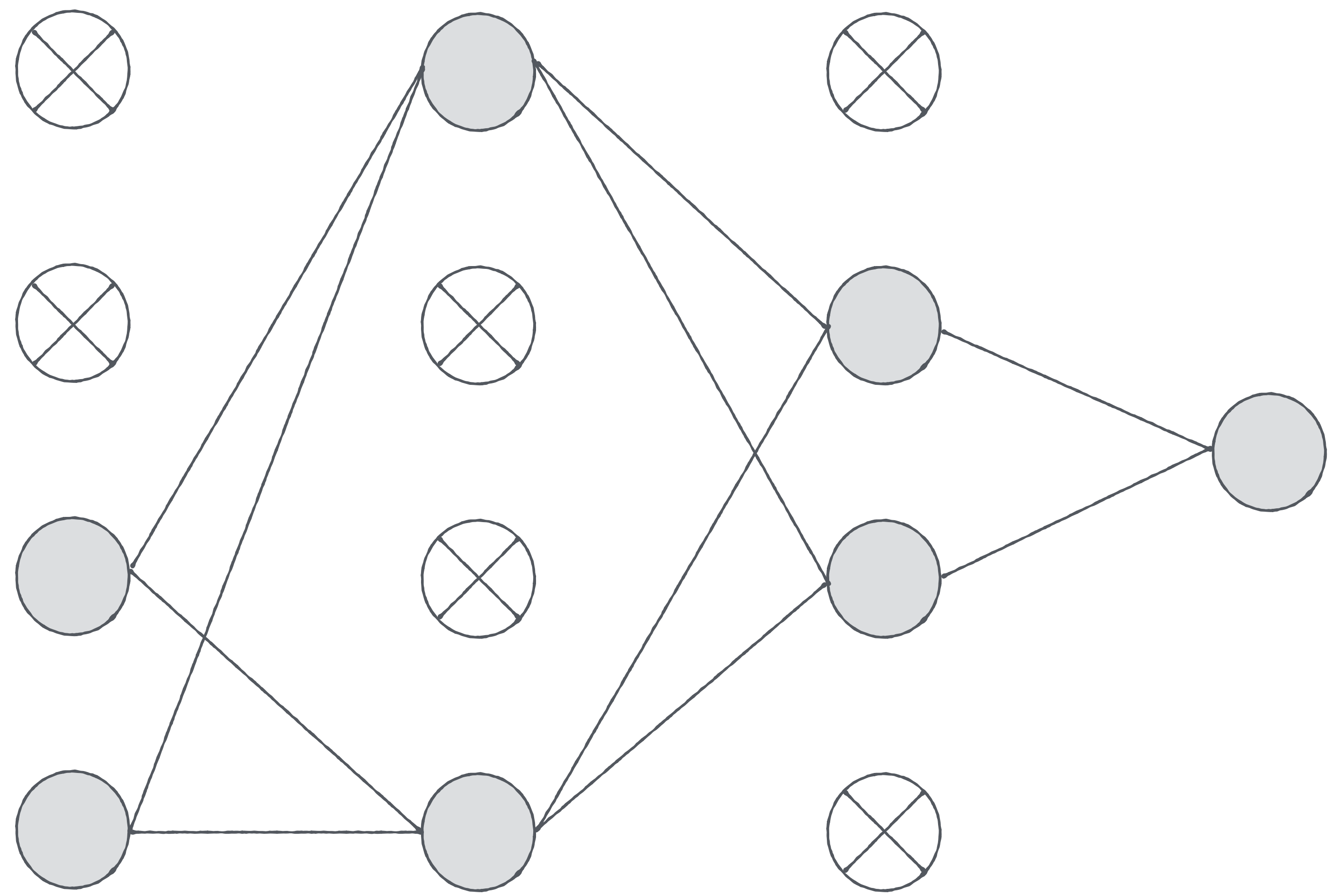
Regularization

$$L(w_1, \ldots, w_n) + c \sum_i w_i^2$$

Dropout



Dropout



Randomly disable neurons for each minibatch

Dropout

Transfer Learning

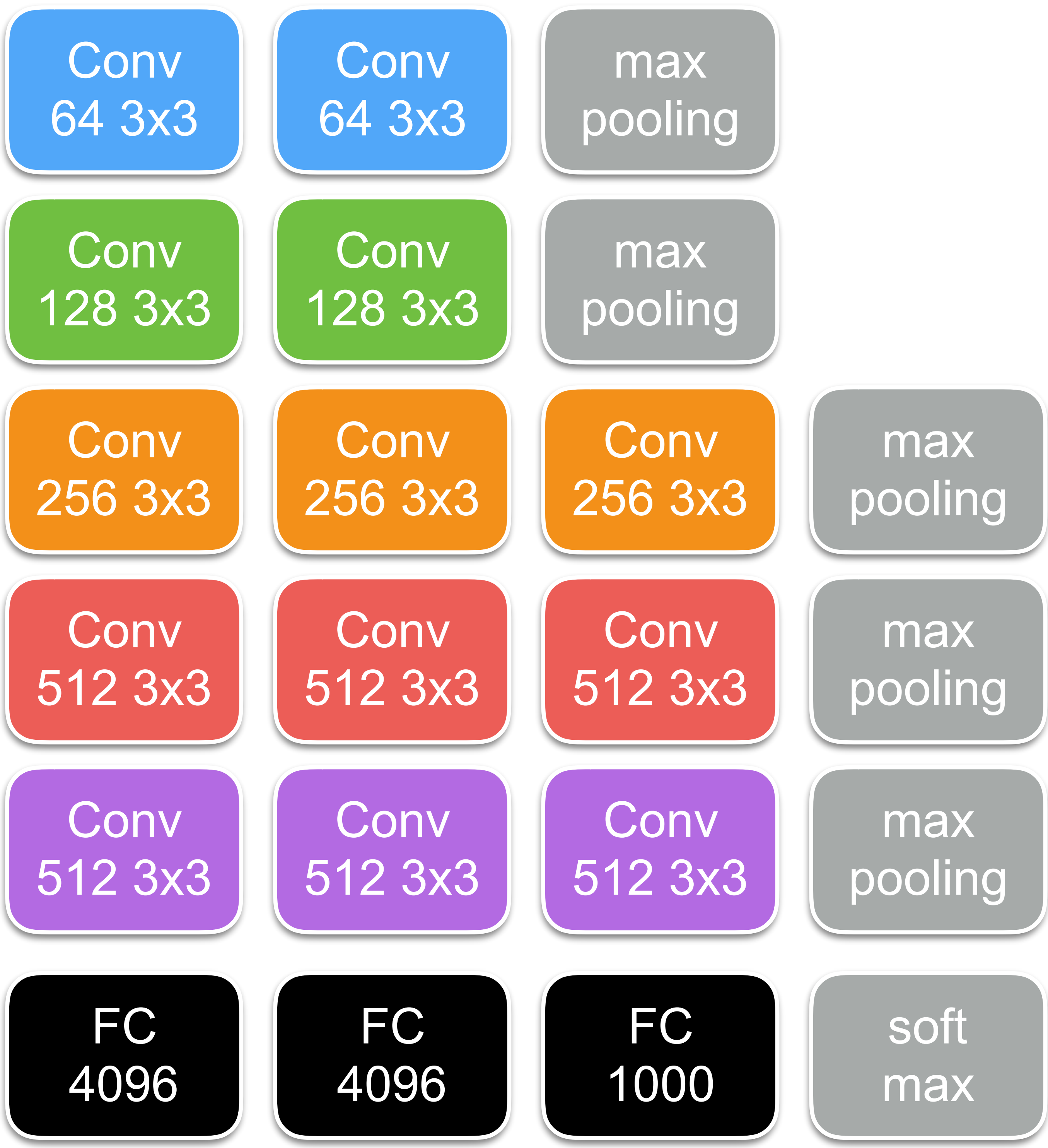
IMAGENET

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

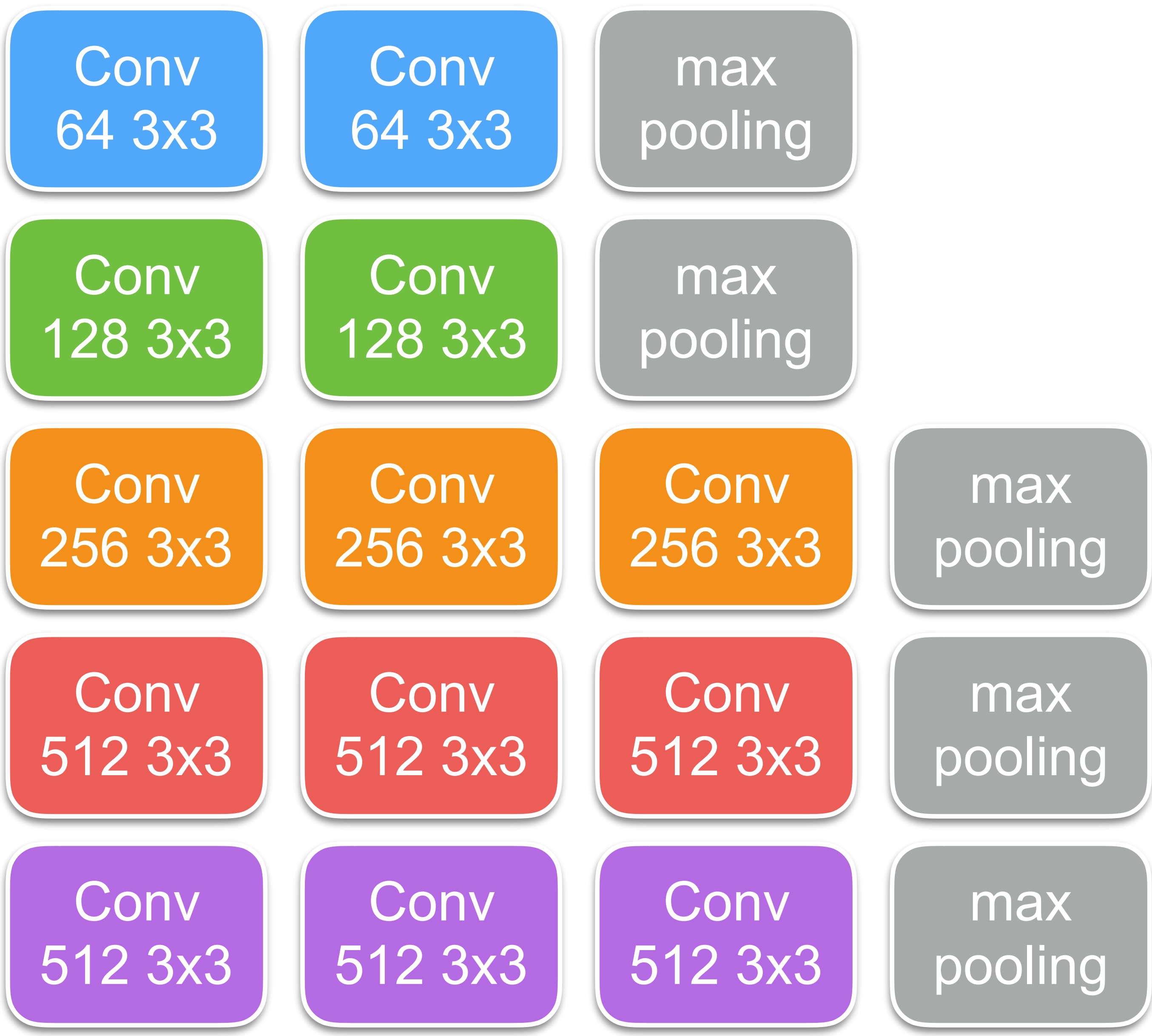
Transfer Learning

VGG-16 architecture



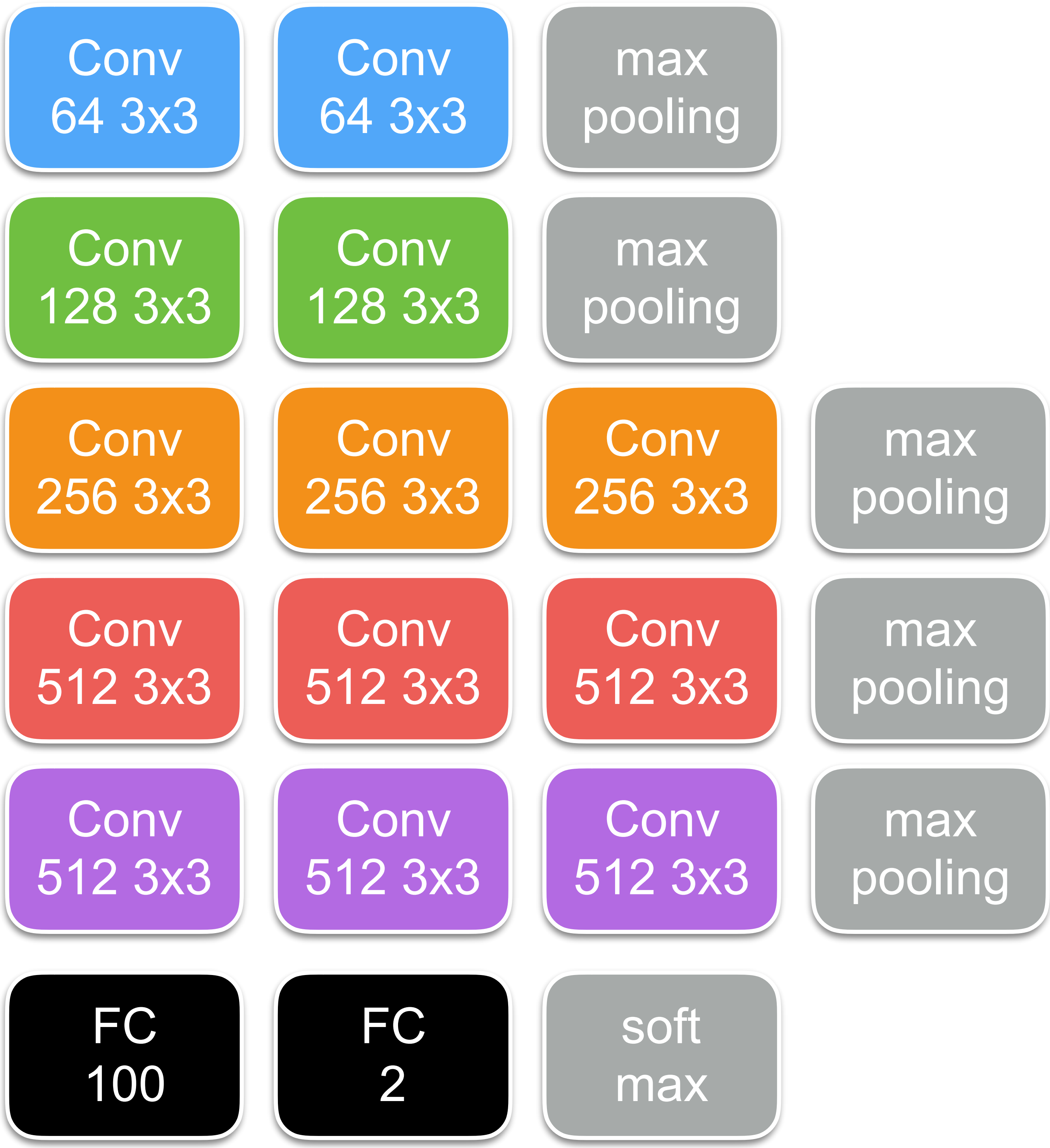
Transfer Learning

VGG-16 architecture



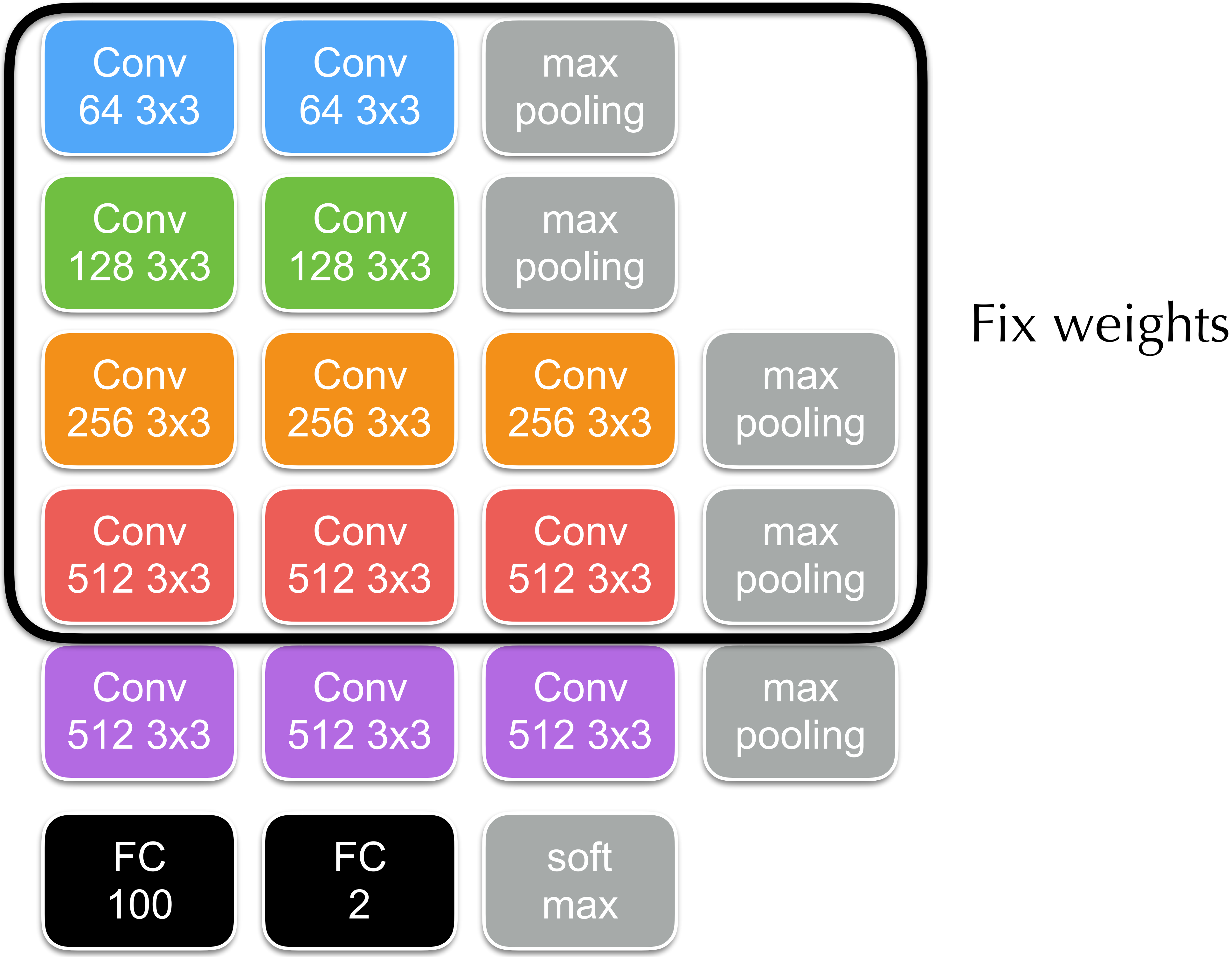
Transfer Learning

VGG-16 architecture



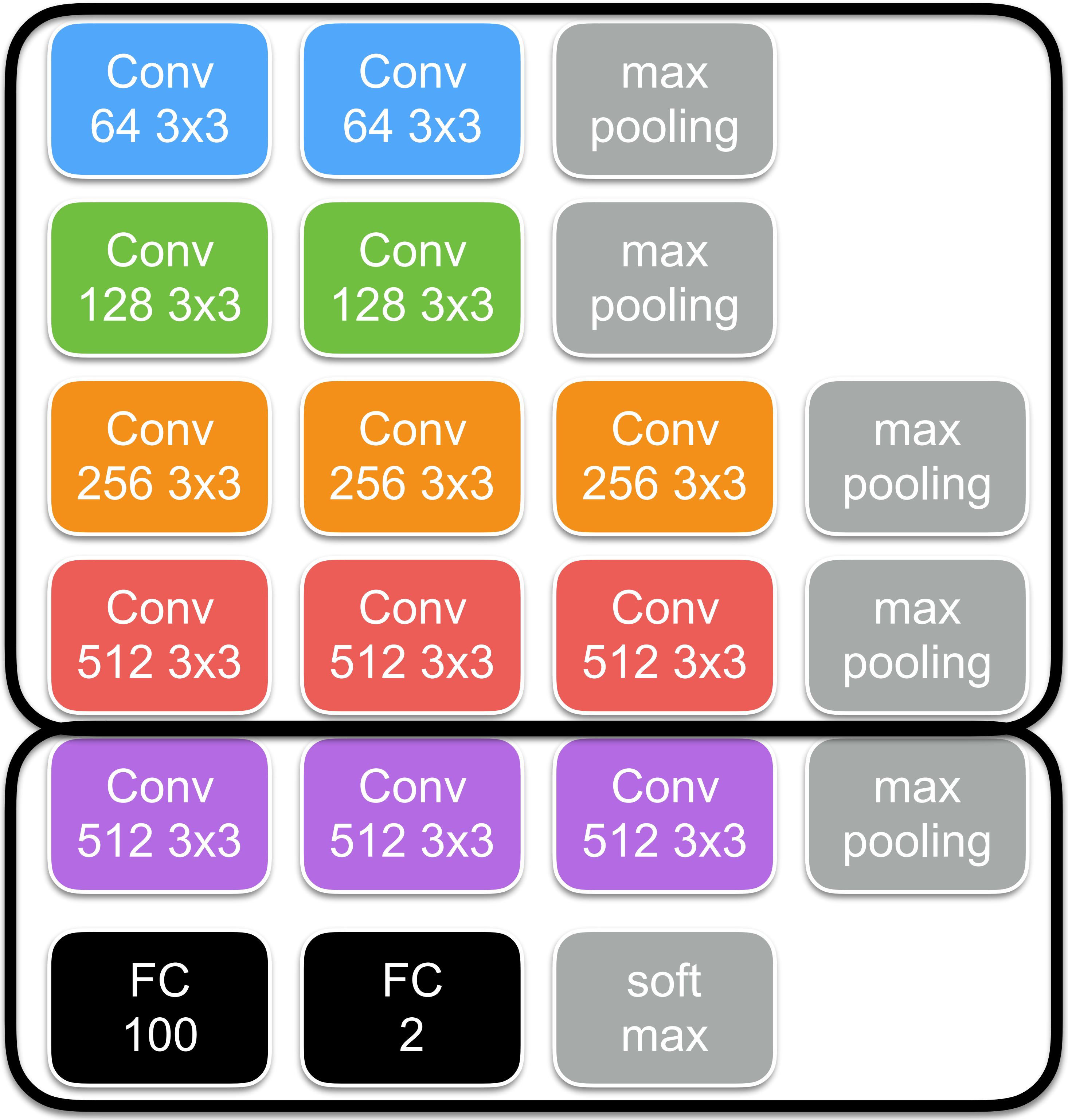
Transfer Learning

VGG-16 architecture



Transfer Learning

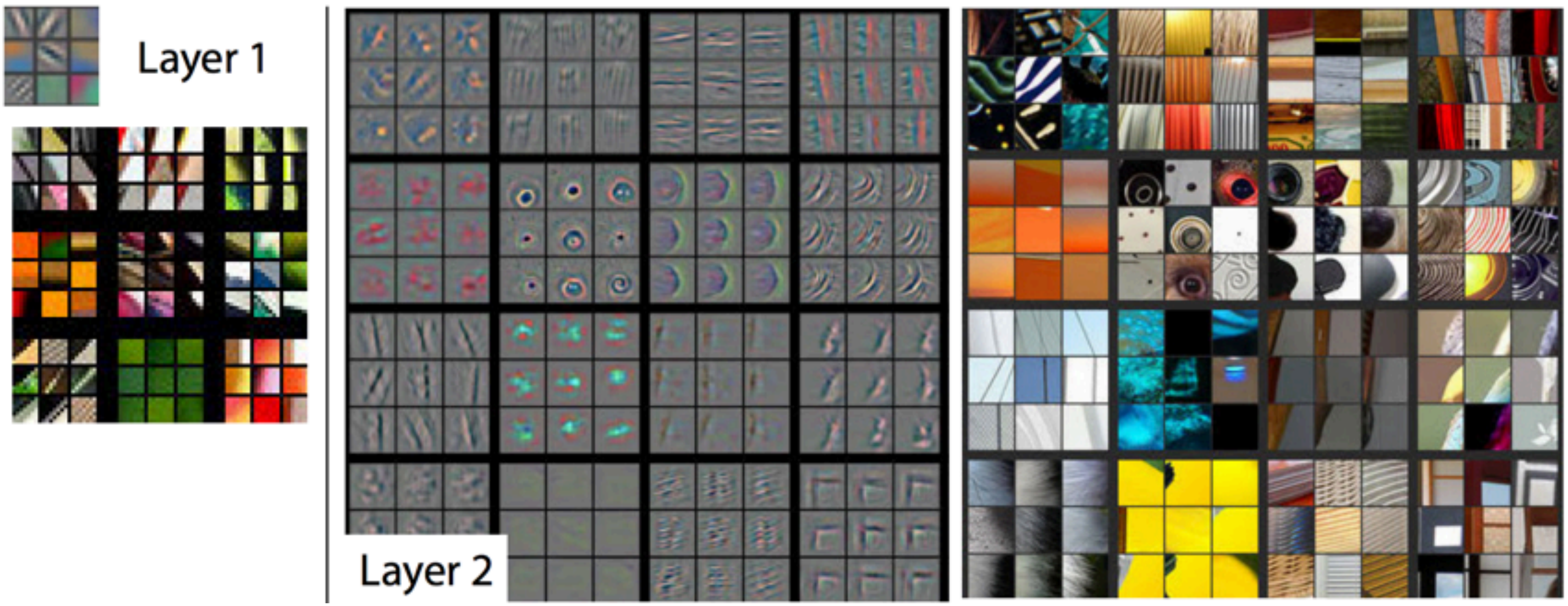
VGG-16 architecture



Fix weights

Train on limited data

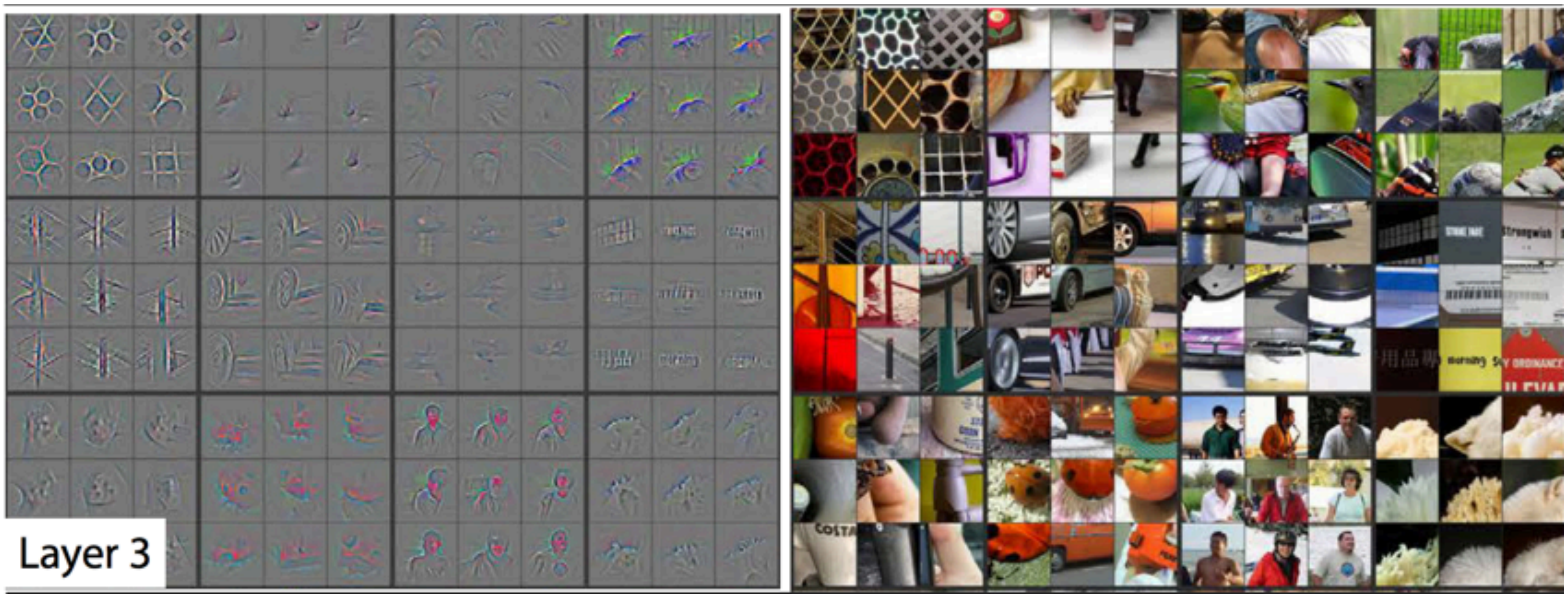
Transfer Learning



Low-level features: corners, edges, ...

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

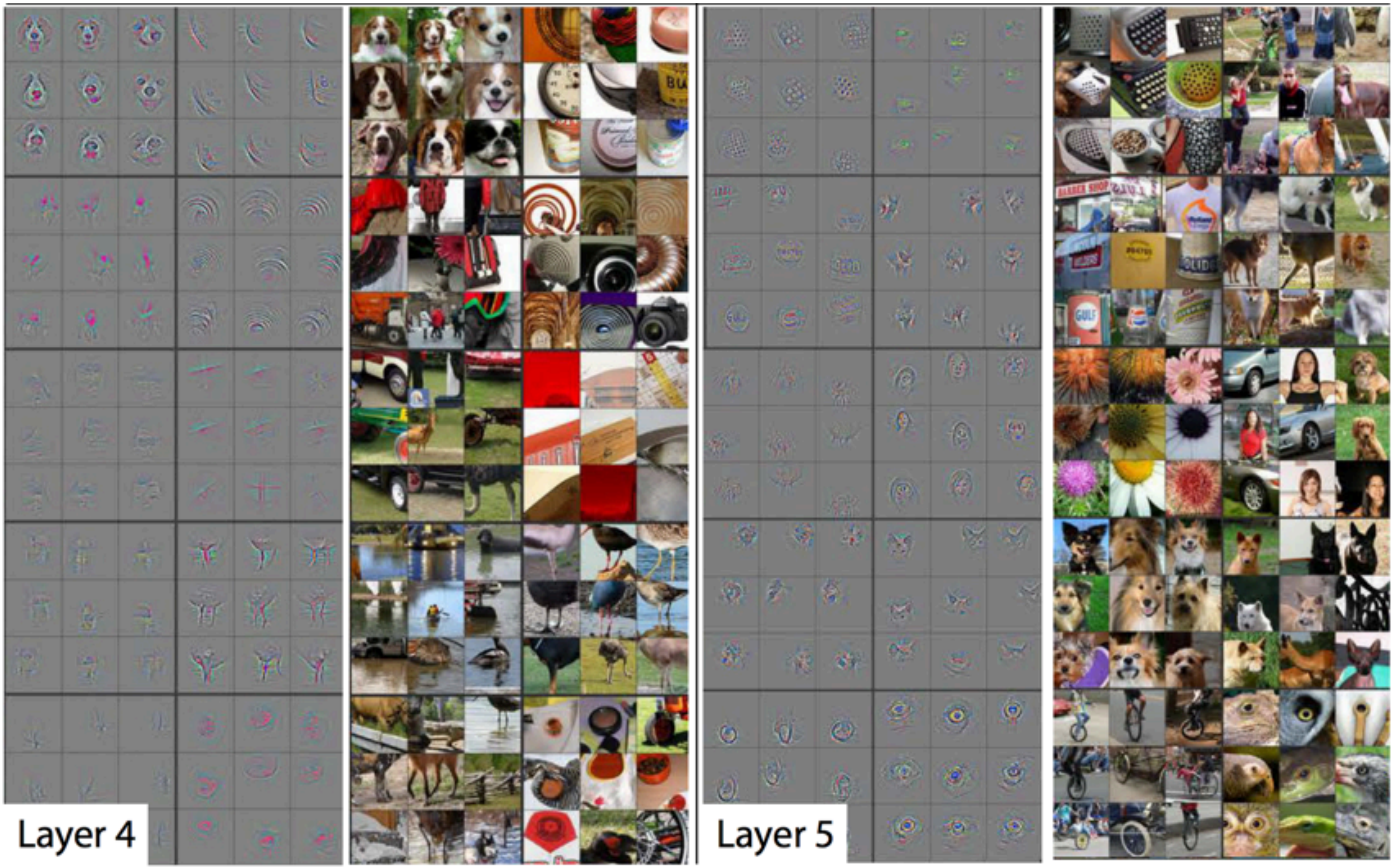
Transfer Learning



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]

Lessons Learned

- Main lessons from this lecture
  - Convolutional neural networks
  - Layers: Fully connected, convolutional, activation functions, max pooling
  - Backpropagation
  - Handling overfitting
- Next lecture: More convolutional Neural Networks

Next Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	**Convolutional neural networks**	
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	