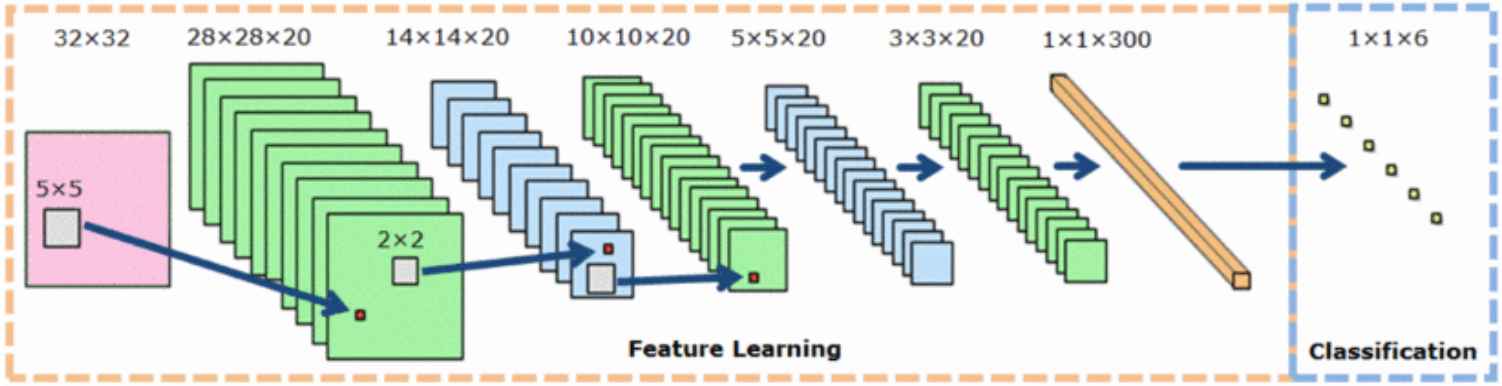
Bildanalyse und Visualisierung in Diagnostik und Therapie

Convolutional Neural Networks



M.Sc. Oskar Maier

Prof. Dr. rer. nat. habil. Heinz Handels

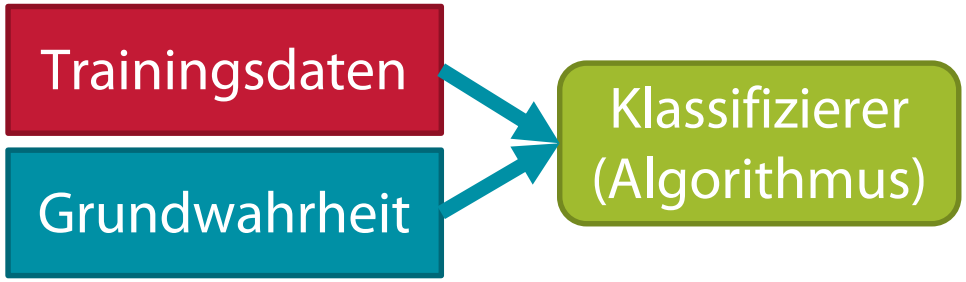
Institut für Medizinische Informatik

Universität zu Lübeck

Bildquelle: Bezák, Pavol, Yury Rafailovich Nikitin, and Pavol Božek. "Robotic Grasping System Using Convolutional Neural Networks." *American Journal of Mechanical Engineering* 2.7 (2014): 216-218.

Überwachtes Lernen eines Klassifizierers

Training



Testing

Anwendung

CNN? Deep Learning? Noch nie gehört!

CNN? Deep Learning? Noch nie gehört!

Bild nach: http://www.collection-panorama-art.com/de/l%C3%BCbeck

CNN? Deep Learning? Noch nie gehört!

DeepDream

Bild nach: http://www.collection-panorama-art.com/de/l%C3%BCbeck

CNN? Deep Learning? Noch nie gehört!

DeepDream

Bild nach: http://www.collection-panorama-art.com/de/l%C3%BCbeck

CNN? Deep Learning? Noch nie gehört!

DeepDream

Bild nach: http://www.collection-panorama-art.com/de/l%C3%BCbeck

CNN? Deep Learning? Noch nie gehört!

DeepDream

Bild nach: http://www.collection-panorama-art.com/de/l%C3%BCbeck

CNN? Deep Learning? Noch nie gehört!

Bildquelle: https://deepmind.com/alpha-go

CNN? Deep Learning? Noch nie gehört!

AlphaGo

Bildquelle: http://www.insight.co.kr/newsRead.php?ArtNo=54448

CNN? Deep Learning? Noch nie gehört!

Skype Translator

Bildquelle: http://www.dailymail.co.uk/sciencetech/article-2880429/How-Bien-gracias-Skype-Translator-beta-released-banish-language-barriers-world.html

Artificial Neural Networks

Artifical Neural Networks (ANN)

ANNs sind

- Multi-parameter Model
- Trainierbar durch Machine Learning Methoden

Anwendungsgebiete

- Klassifizierung
- Regression
- Prädiktion
- Kompression
- … und viele mehr

Inspiration aus der Neurobiologie

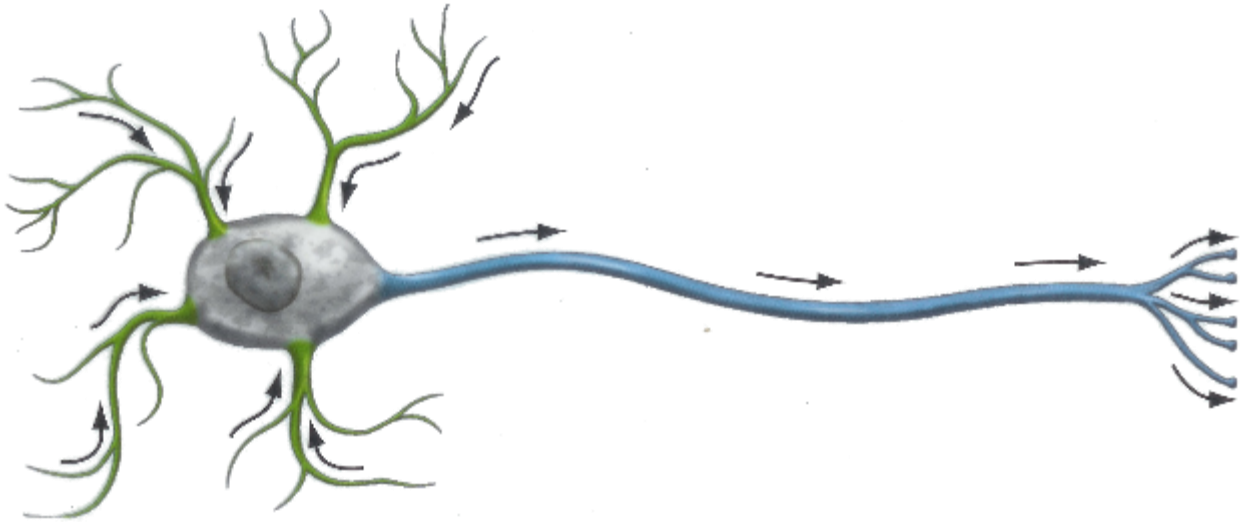
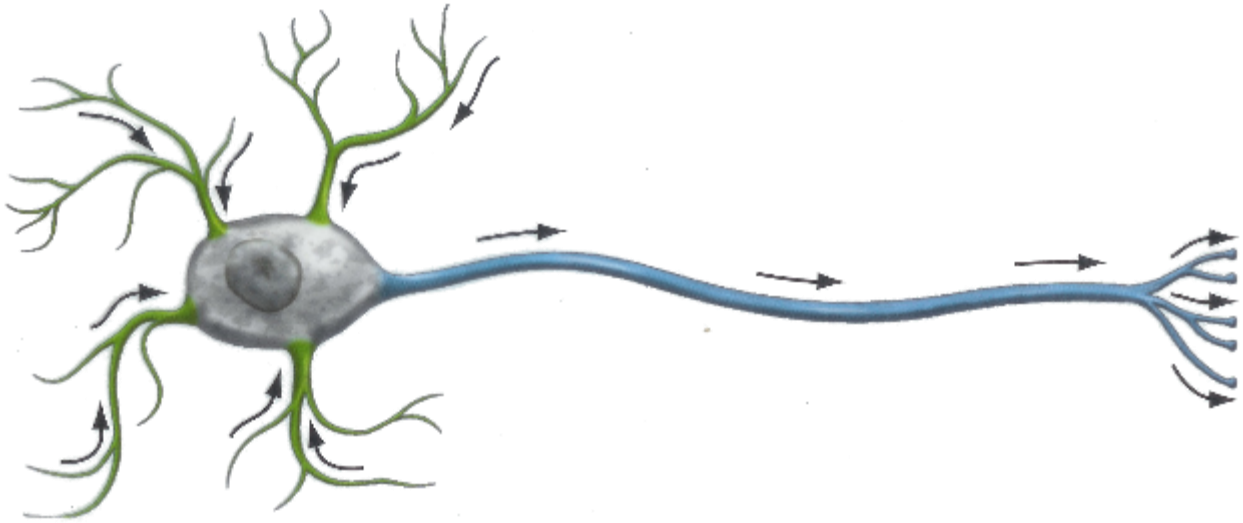


Bild nach: http://bio100.class.uic.edu/lectures/nervous.htm

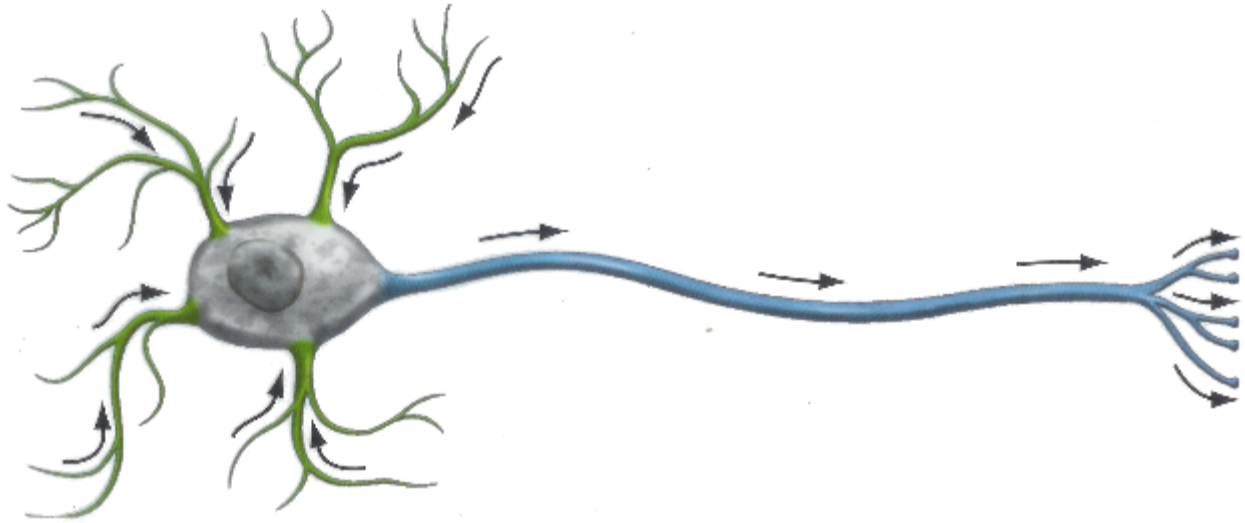
Inspiration aus der Neurobiologie



Dendrites → Zellkörper → Axon

Bild nach: http://bio100.class.uic.edu/lectures/nervous.htm

Inspiration aus der Neurobiologie

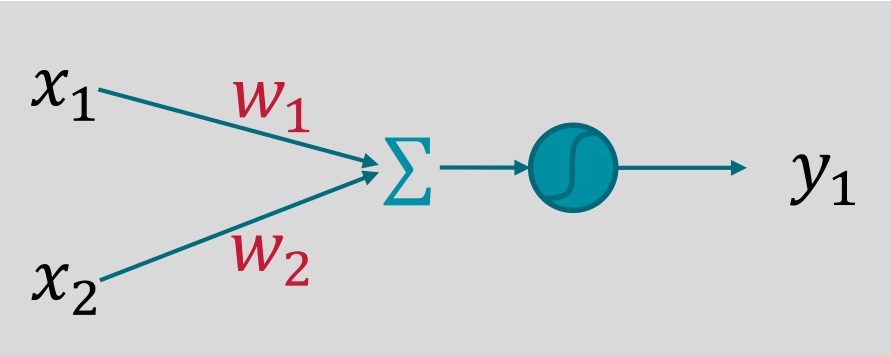


Dendrites → Zellkörper → Axon

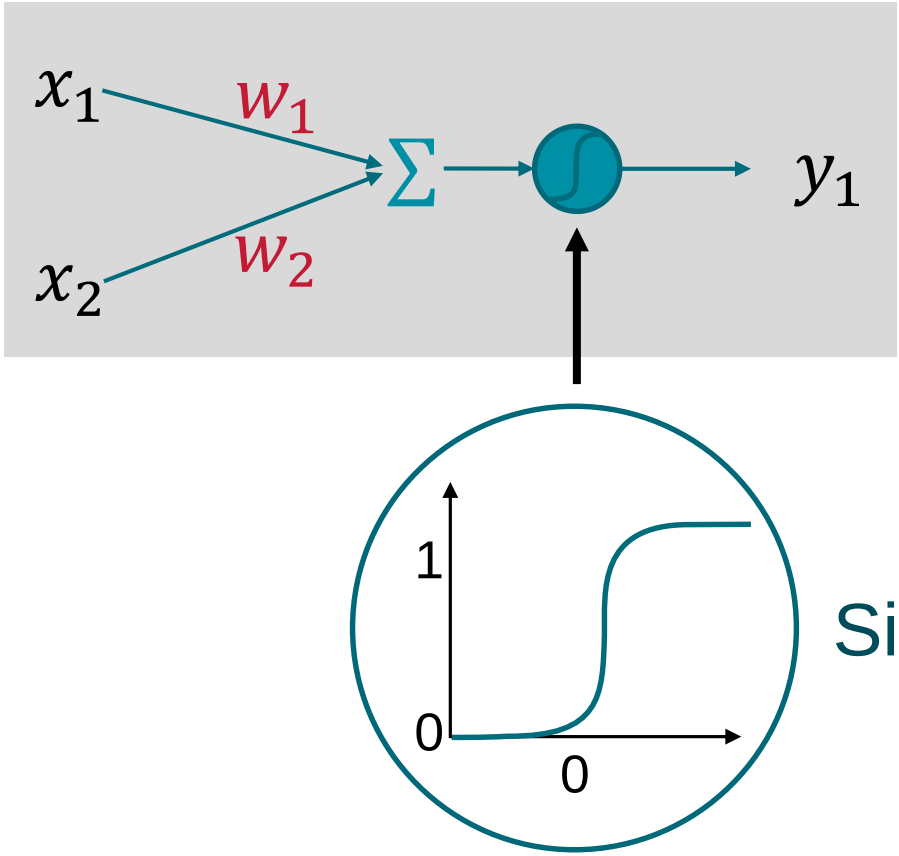
Bild nach: http://bio100.class.uic.edu/lectures/nervous.htm

Ein minimales Netwerk

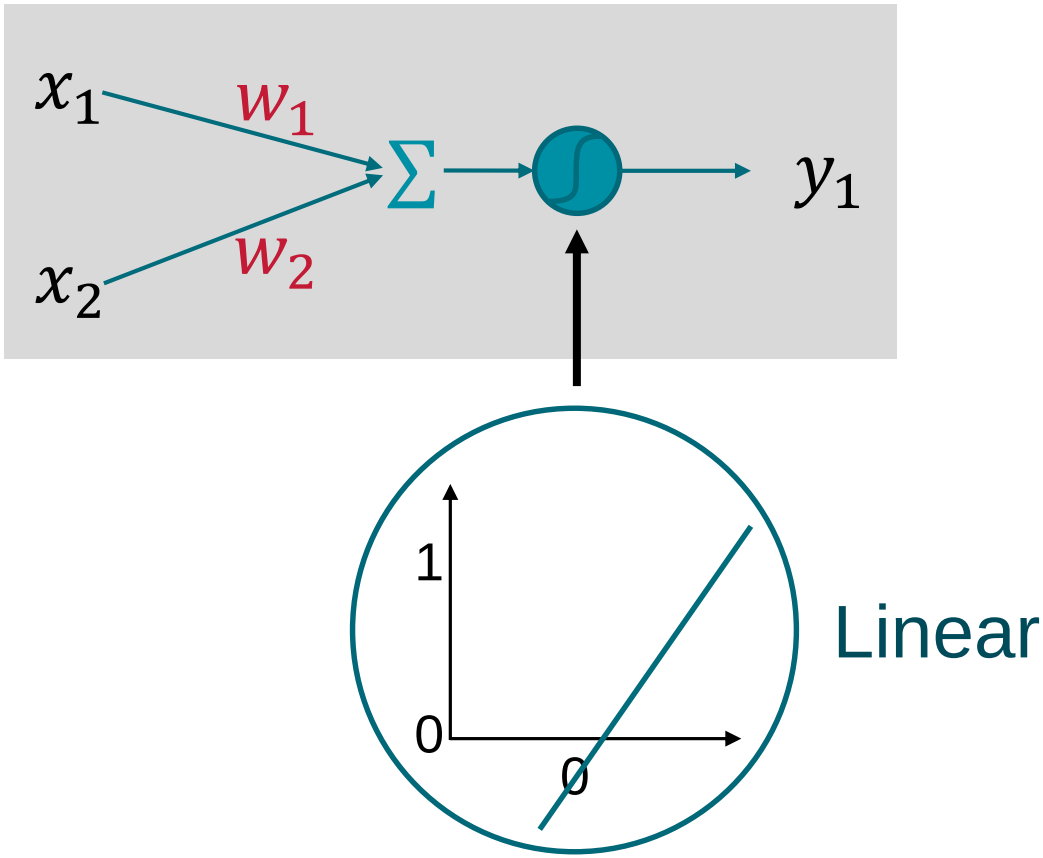
Ein minimales Netwerk



Ein minimales Netwerk: Activation function

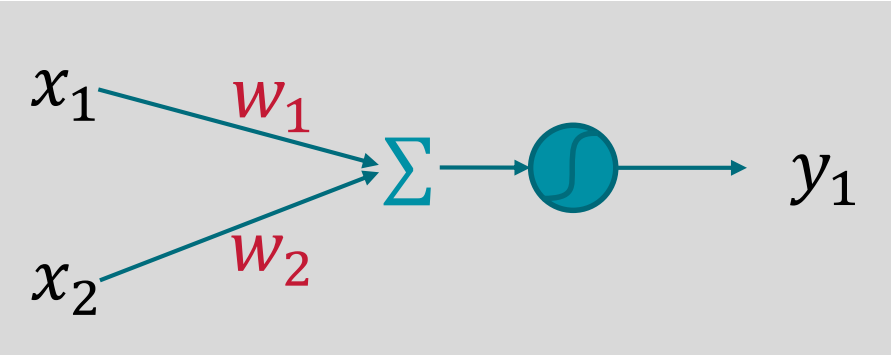


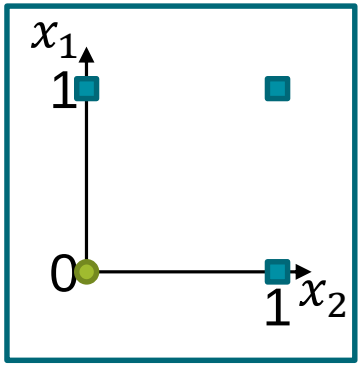
Ein minimales Netwerk: Activation function



Ein minimales Netwerk: OR Beispiel

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	1



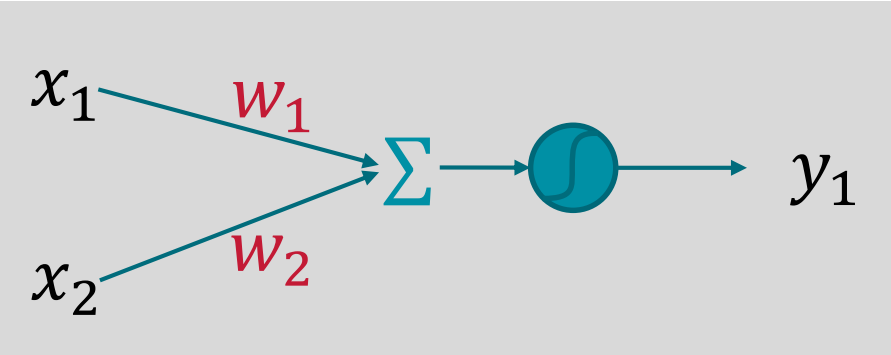


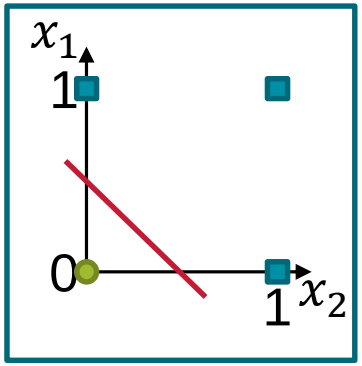
$$y_1 = \boldsymbol{x}^T \boldsymbol{w}$$

$$\hat{\boldsymbol{w}} = ?$$

Ein minimales Netwerk: OR Beispiel

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	1

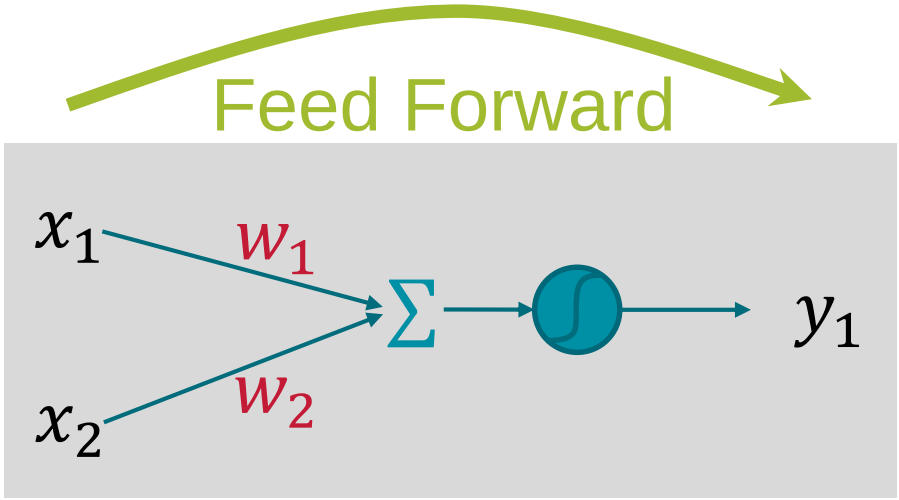




$$y_1 = \boldsymbol{x}^T \boldsymbol{w}$$

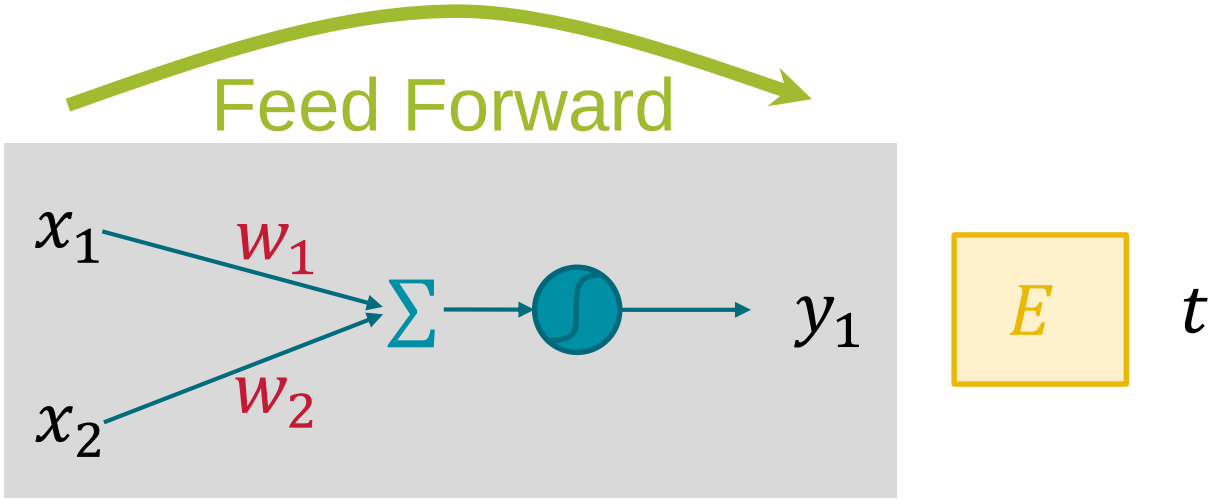
$$\hat{\boldsymbol{w}} = (1.0, 1.0)$$

How to train your network I



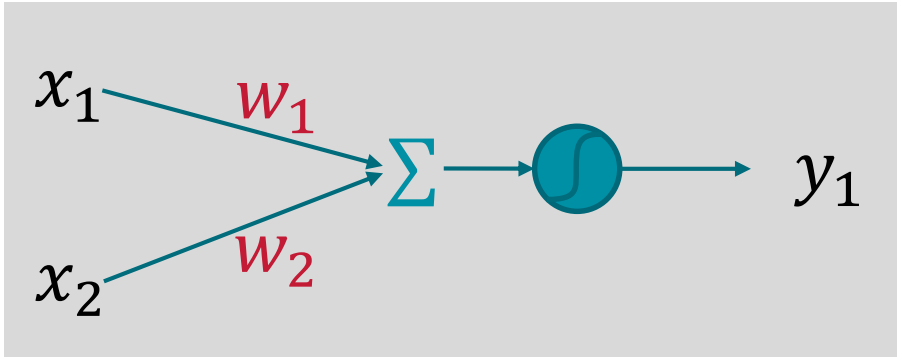
$$y_1 = \boldsymbol{x}^T \boldsymbol{w} = x_2 w_2 + x_1 w_1$$

How to train your network I



$$y_1 = \boldsymbol{x}^T \boldsymbol{w} = x_2 w_2 + x_1 w_1$$

How to train your network I



$$w_1' \leftarrow w_1 - \eta \frac{\partial E}{\partial w_1}$$

Gradientenabstiegsverfahren

$$E(w) = (y - t)^2$$

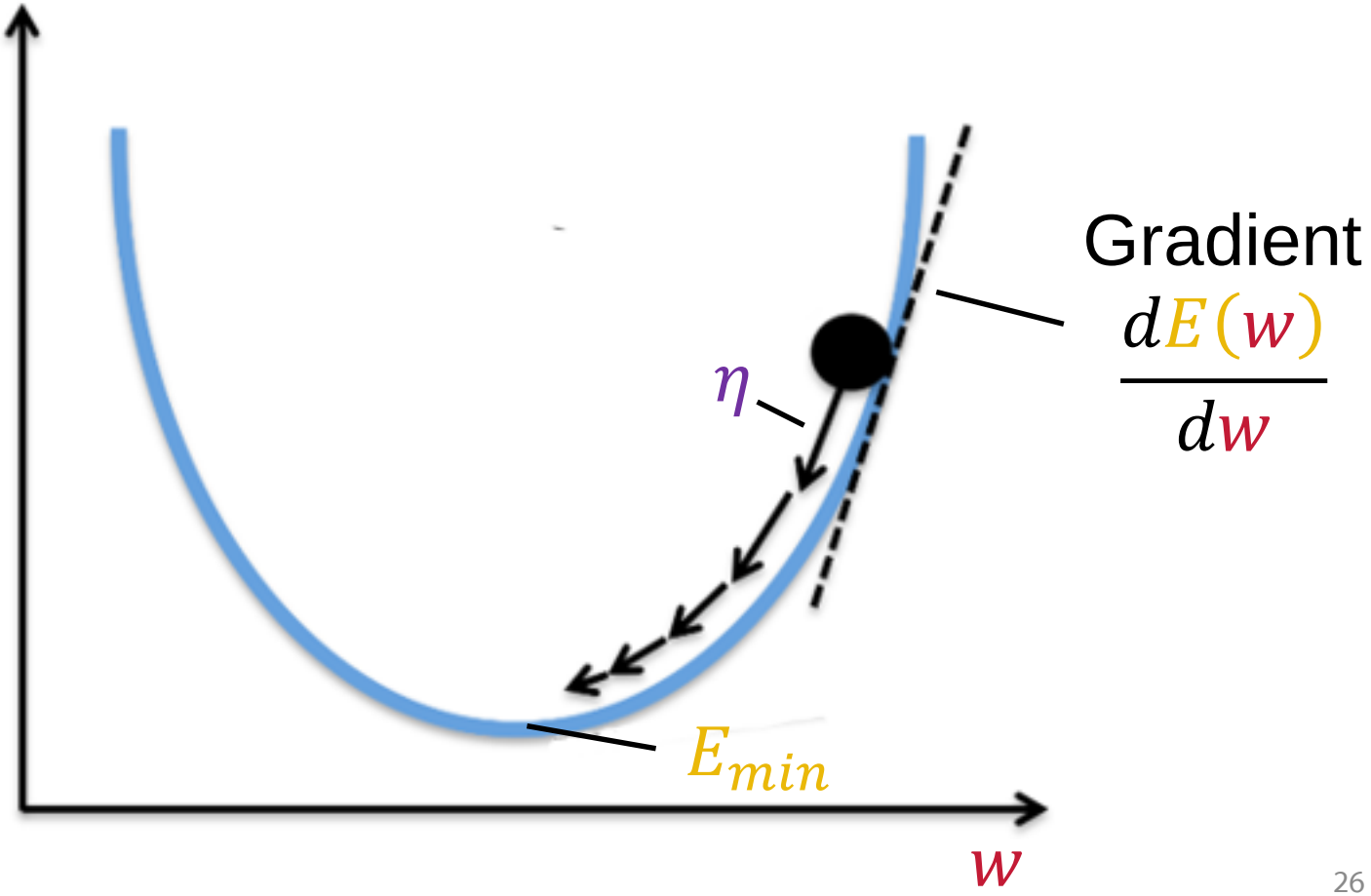
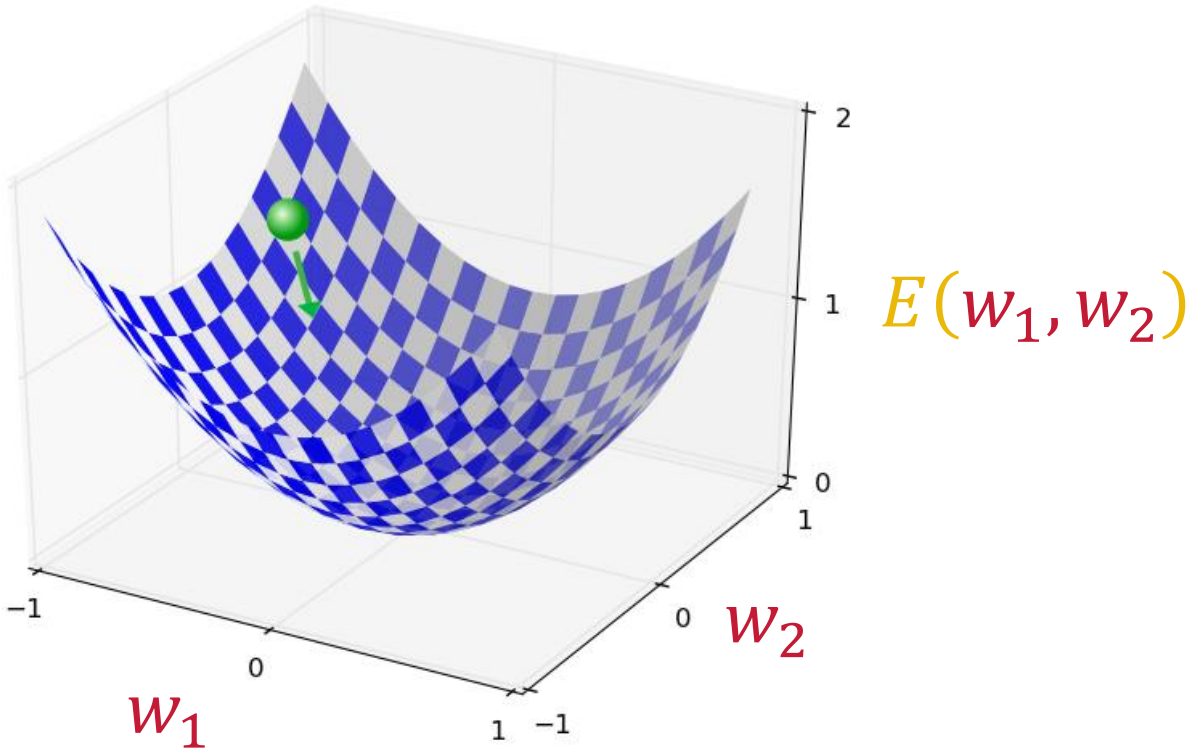


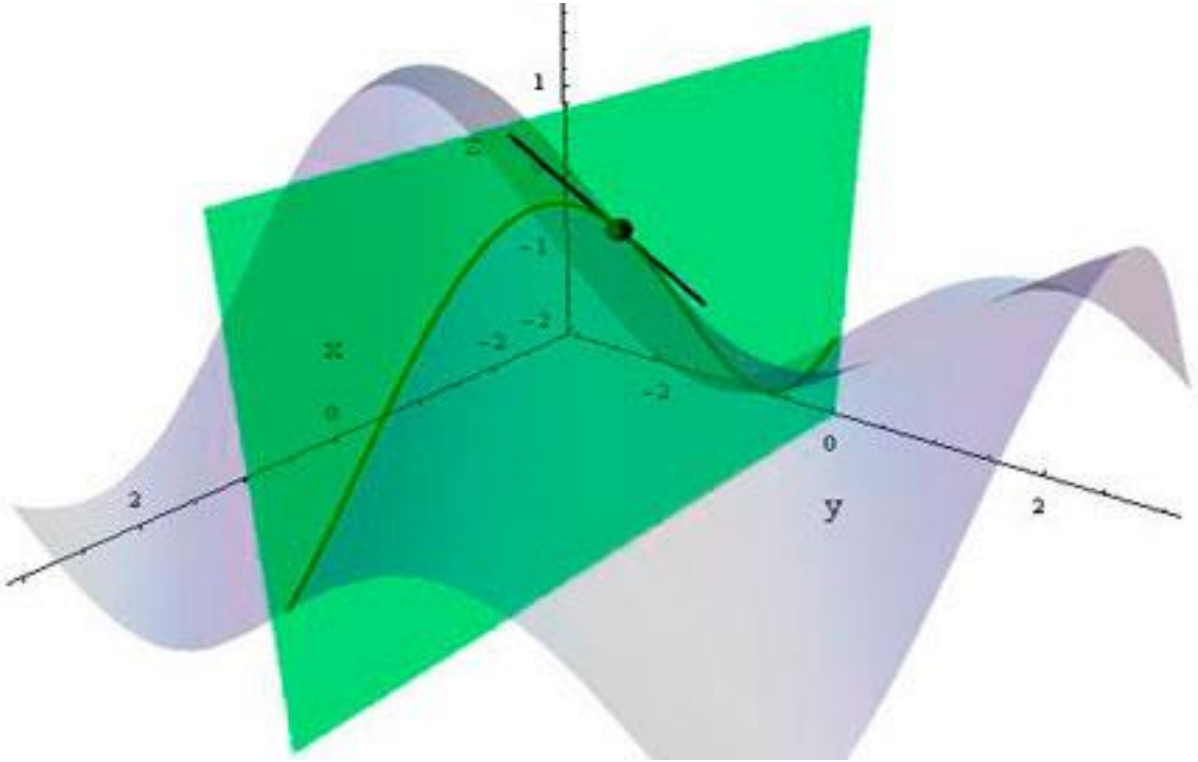
Bild nach: sebastianraschka.com

Gradientenabstiegsverfahren



Bildquelle: http://neuralnetworksanddeeplearning.com/chap1.html

Partielle Ableitung



Bildquelle: alamo.edu

How to train your network I

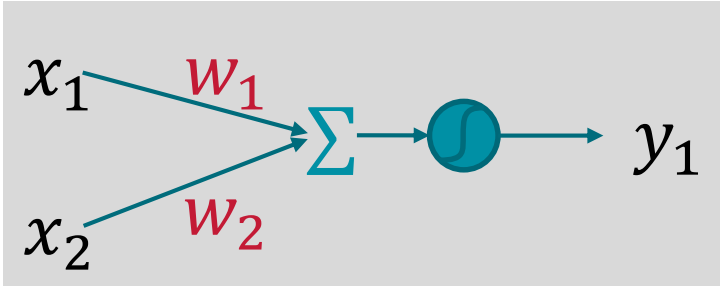
Zusammenfassung

- Mit Backpropagation trainieren wir jedes Gewicht gesondert
- Mit dem Gradientenabstiegsverfahren optimieren wir
- Die Fehlerfunktion bestimmt den Fehler
- Die Lernrate bestimmt die Lerngeschwindigkeit

$$w_1' \leftarrow w_1 - \eta \frac{\partial E}{\partial w_1}$$

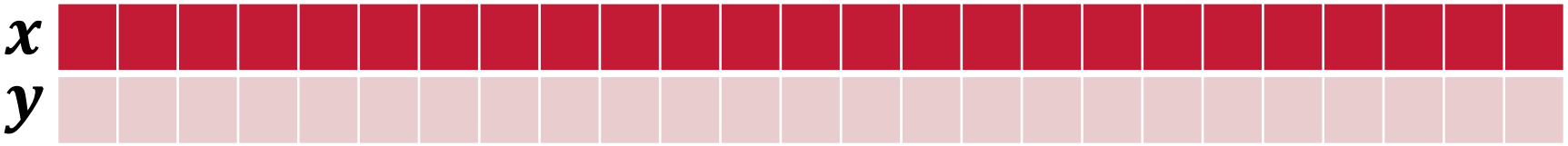
Schwierigkeiten

- Alle Elemente des Netwerkes müssen differenzierbar sein
- Fehlerfunktion bestimmen
- Lernrate bestimmen
- Overfitting

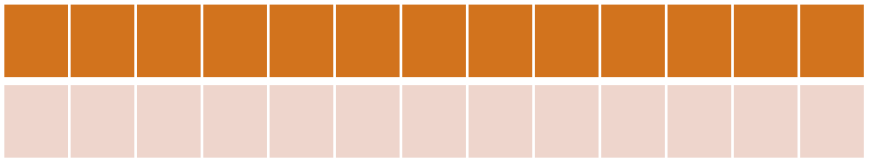


How to train your network II

Data

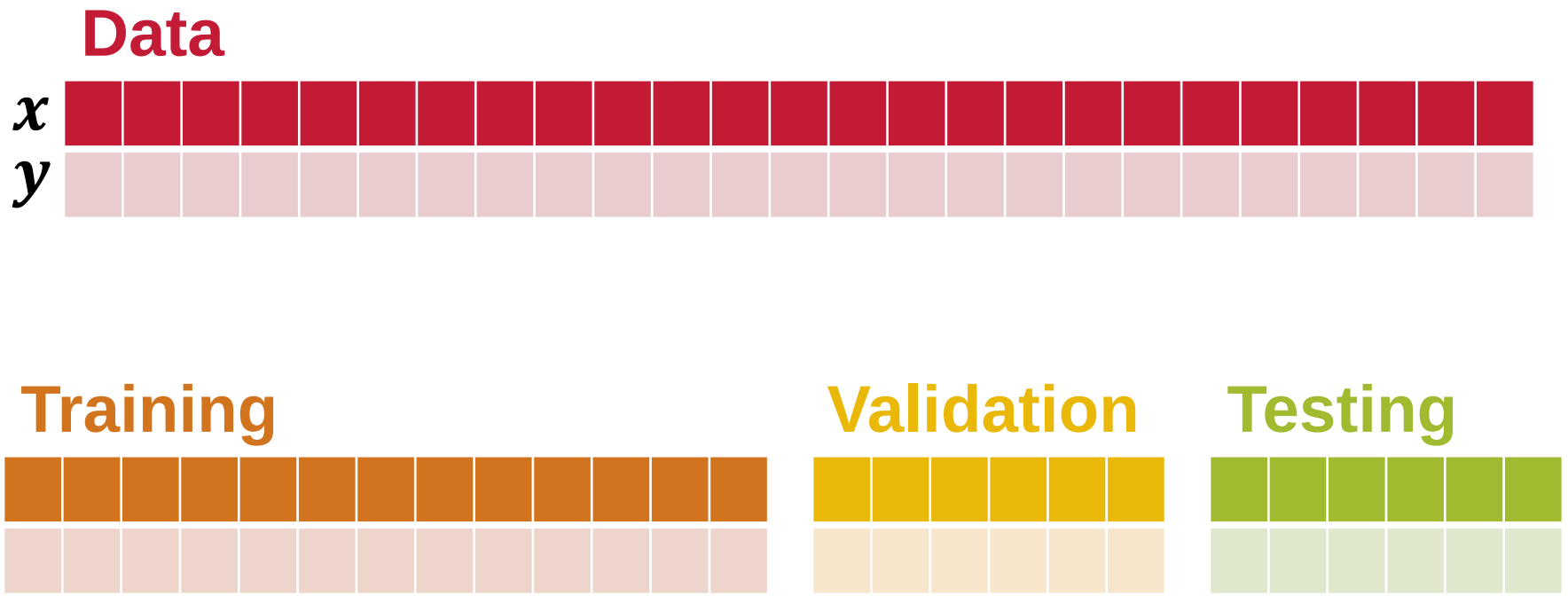


Training

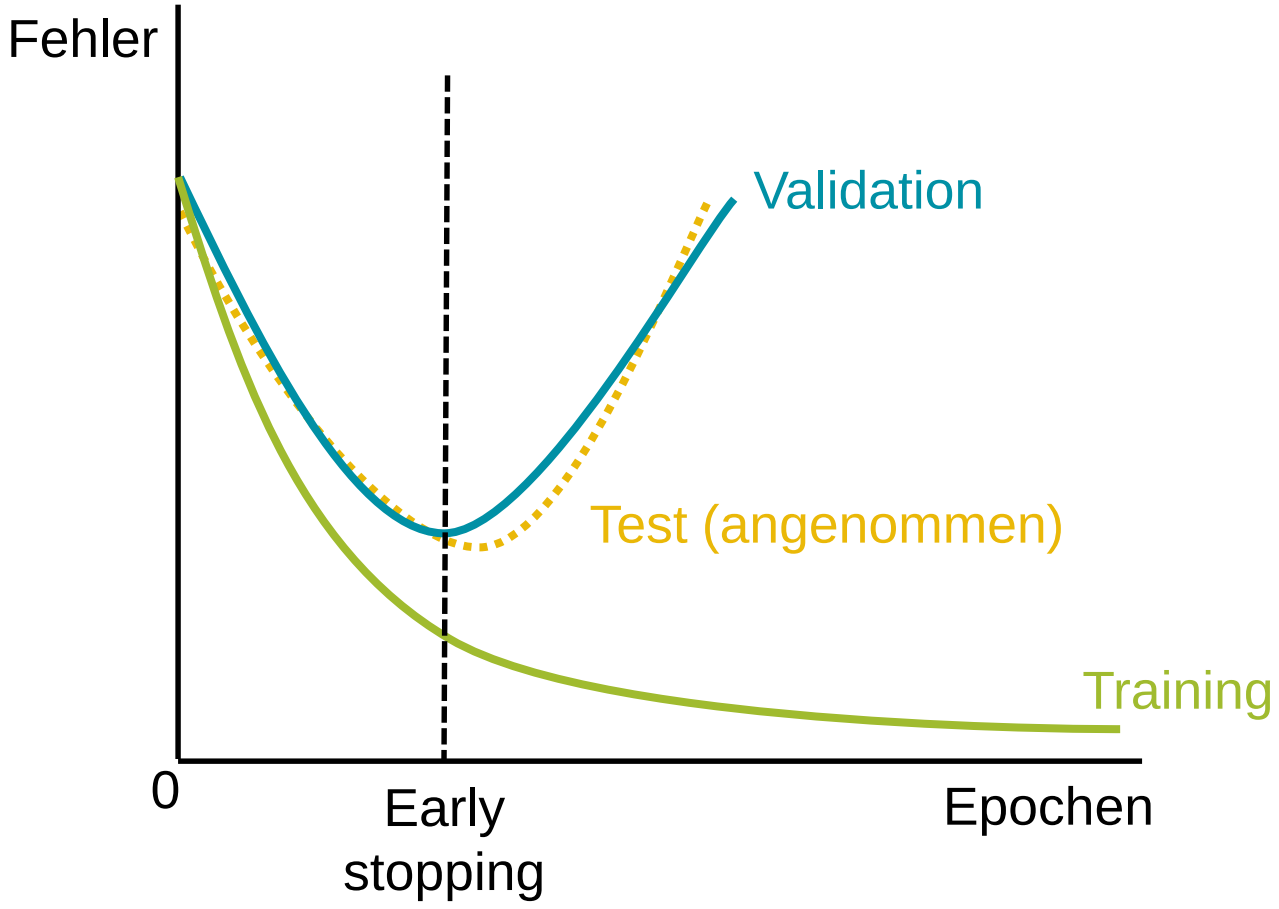


Testing

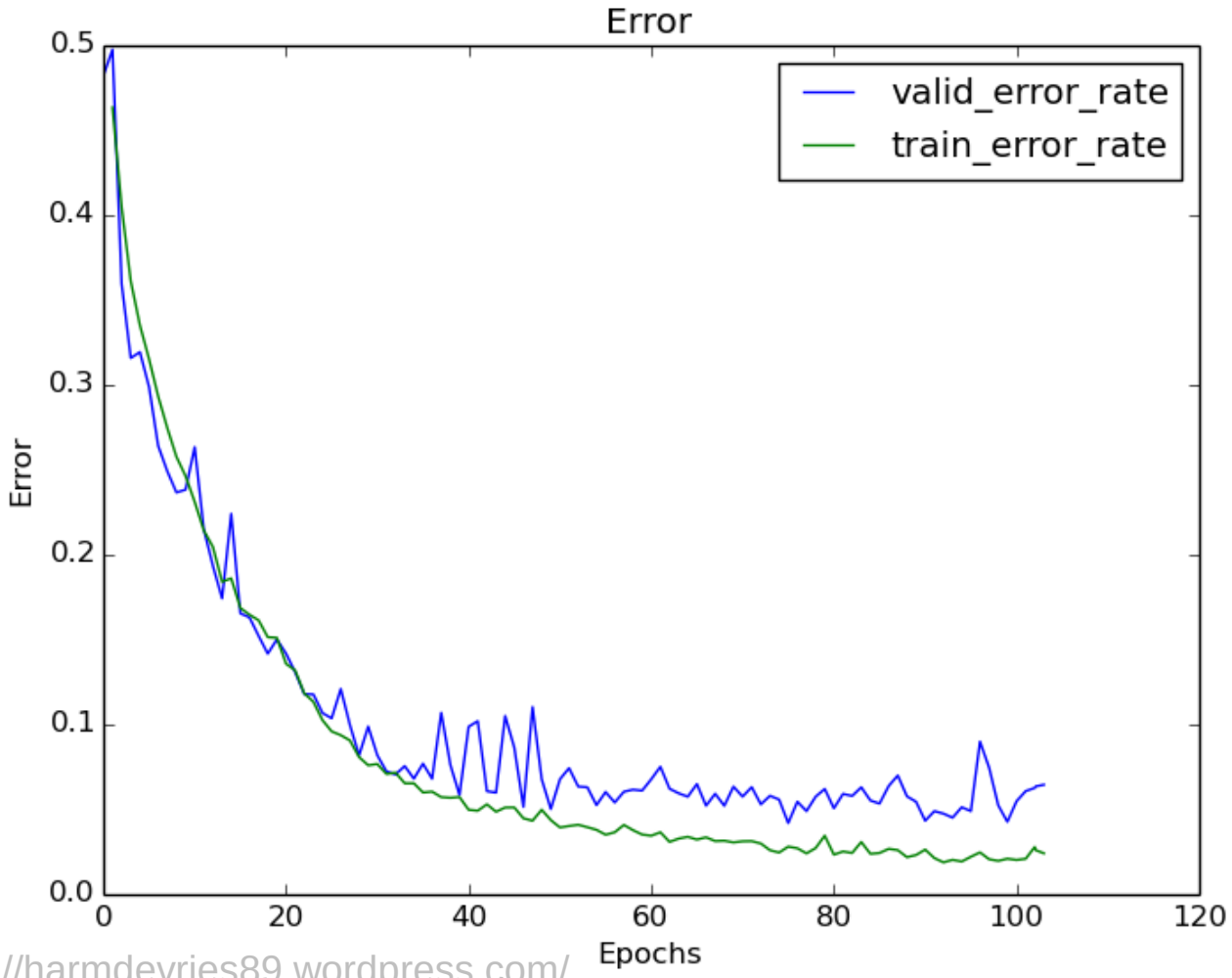
How to train your network II



Idealer Fall von Early Stopping



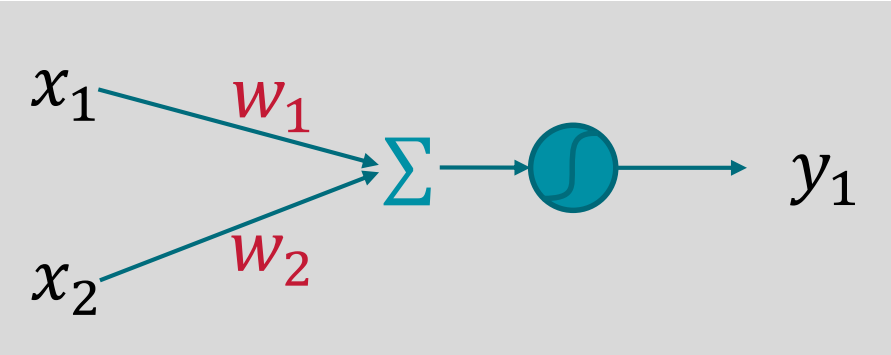
Praktischer Fall von Early Stopping

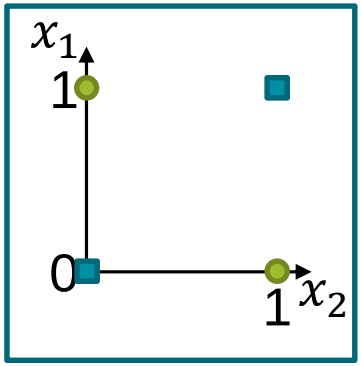


Bildquelle: https://harmdevries89.wordpress.com/

Ein minimales Netwerk: XOR Beispiel

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	0





$$y_1 = \boldsymbol{x}^T \boldsymbol{w}$$

$$\hat{\boldsymbol{w}} = ?$$

Grenzen des minimalen Netwerkes

Problem

- Ein einfaches Netwerk kann nur lineare Probleme lösen

Lösungsmöglichkeit

- Linearkombination von x_1 und x_2 als weiterer Input x_3

Aber

- Ist problemabhängig und häufig unbekannt

Wie kann das Netwerk diese lernen und somit nicht-lineare Problem zu lösen?

Grenzen des minimalen Netwerkes

Problem

- Ein einfaches Netwerk kann nur **lineare Probleme** lösen

Lösungsmöglichkeit

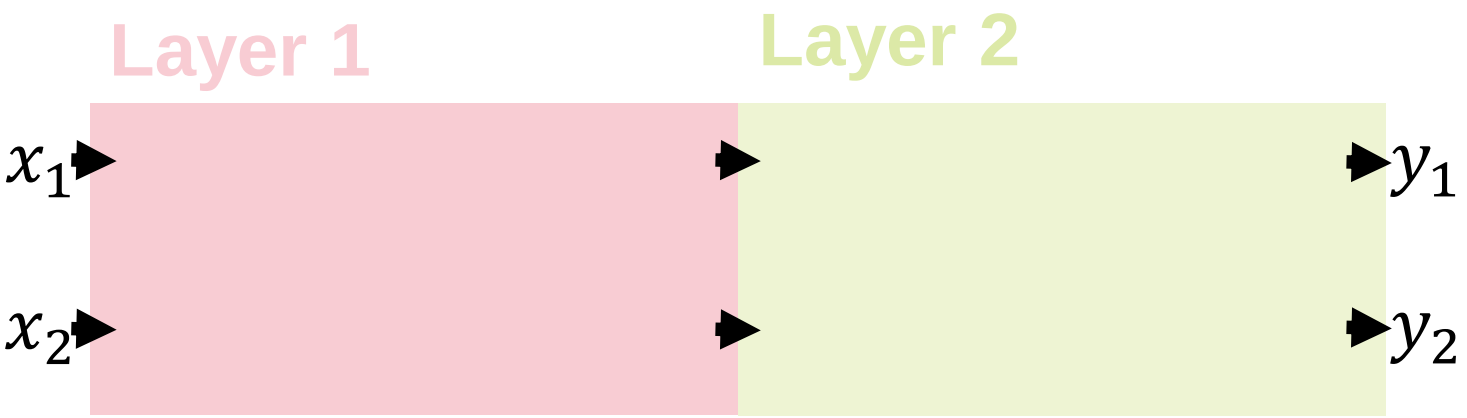
- **Linearkombination** von x_1 und x_2 als weiterer Input x_3

Aber

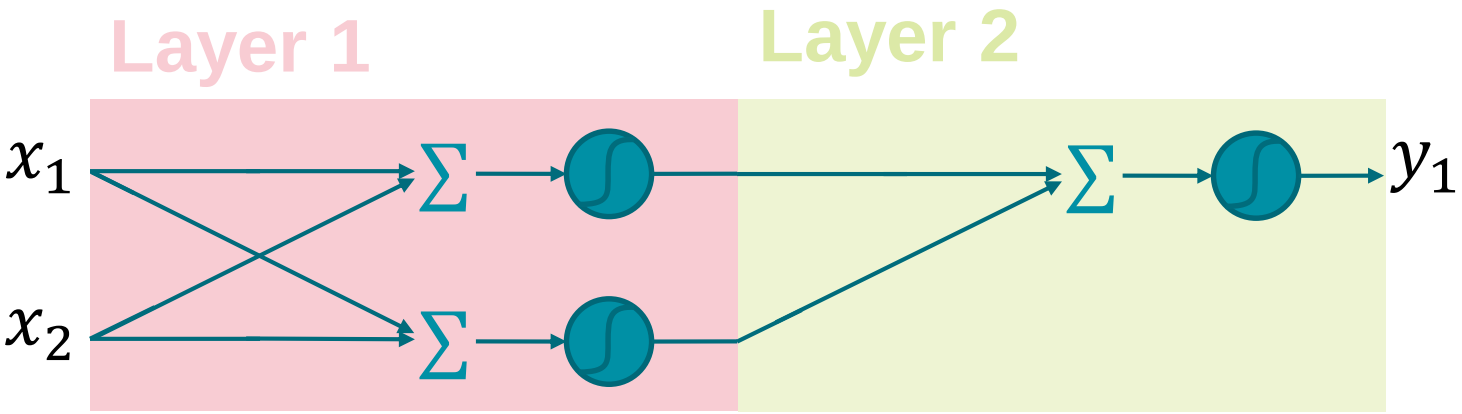
- Ist problemabhängig und häufig unbekannt

Wie kann das Netwerk diese lernen und somit nicht-lineare Problem zu lösen?

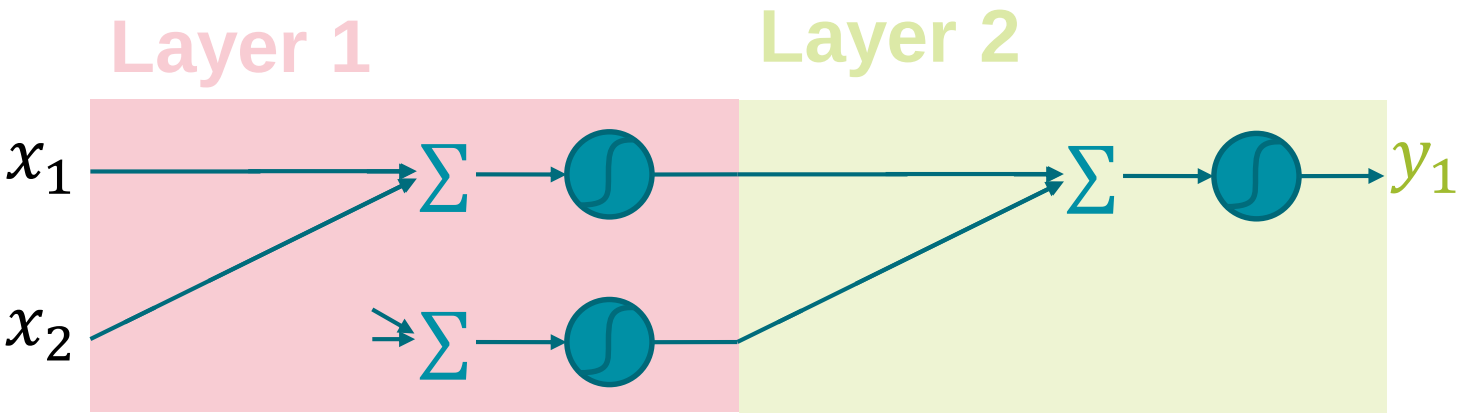
Ein einfaches multi-layer Netwerk



Ein einfaches multi-layer Netwerk

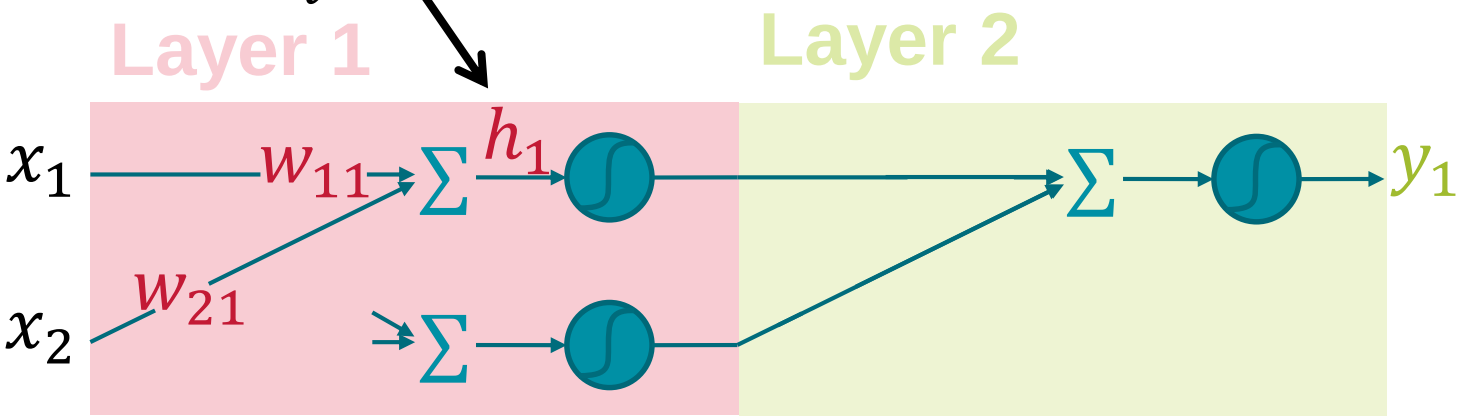


Feed Forward



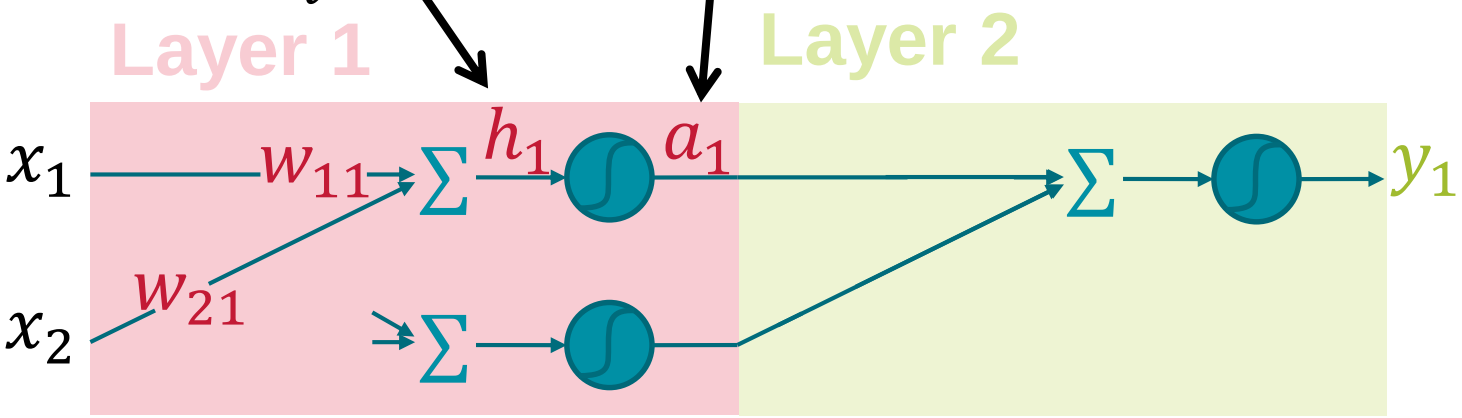
Feed Forward

$$h_1 = \sum_i w_{i1} x_i$$

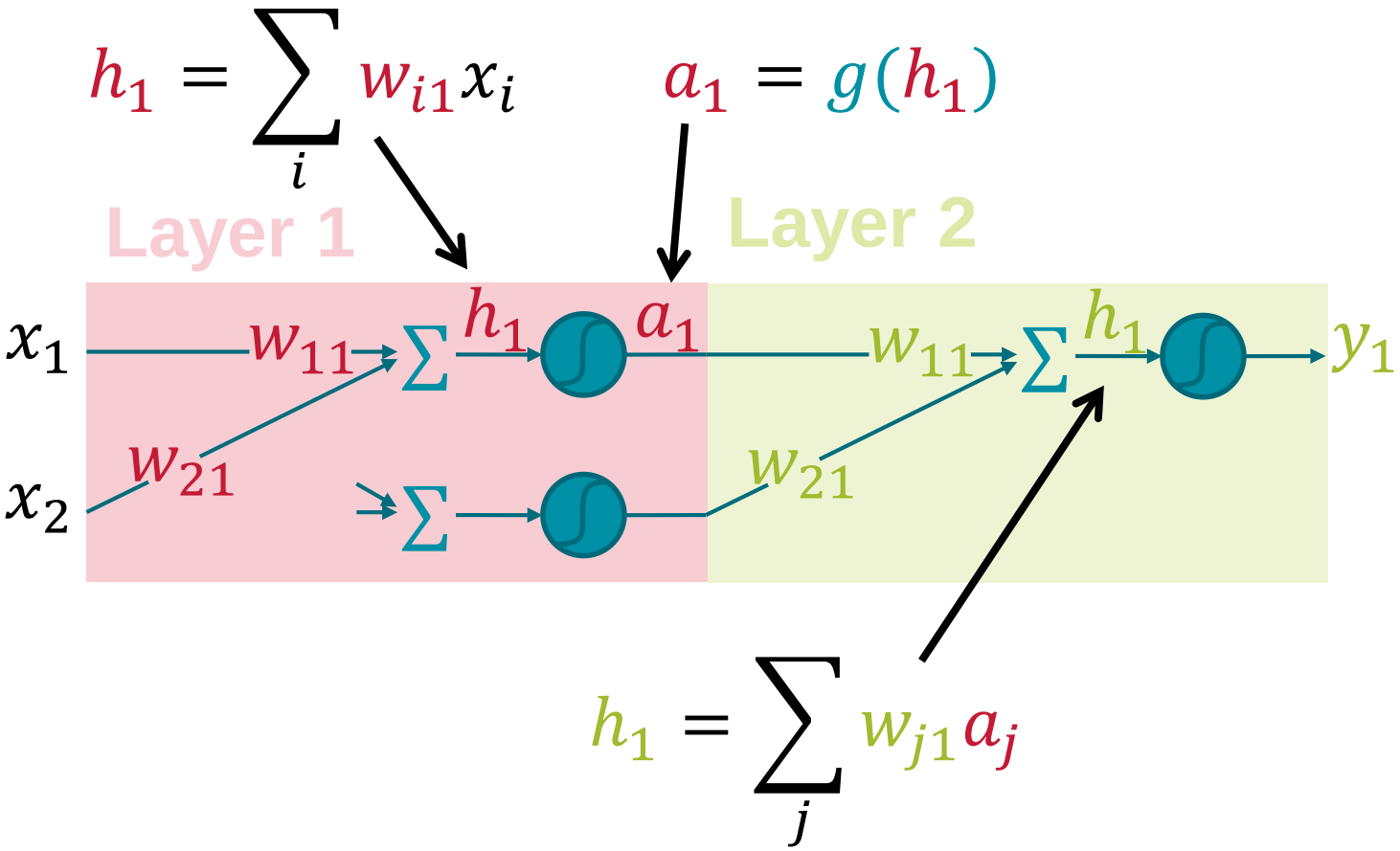


Feed Forward

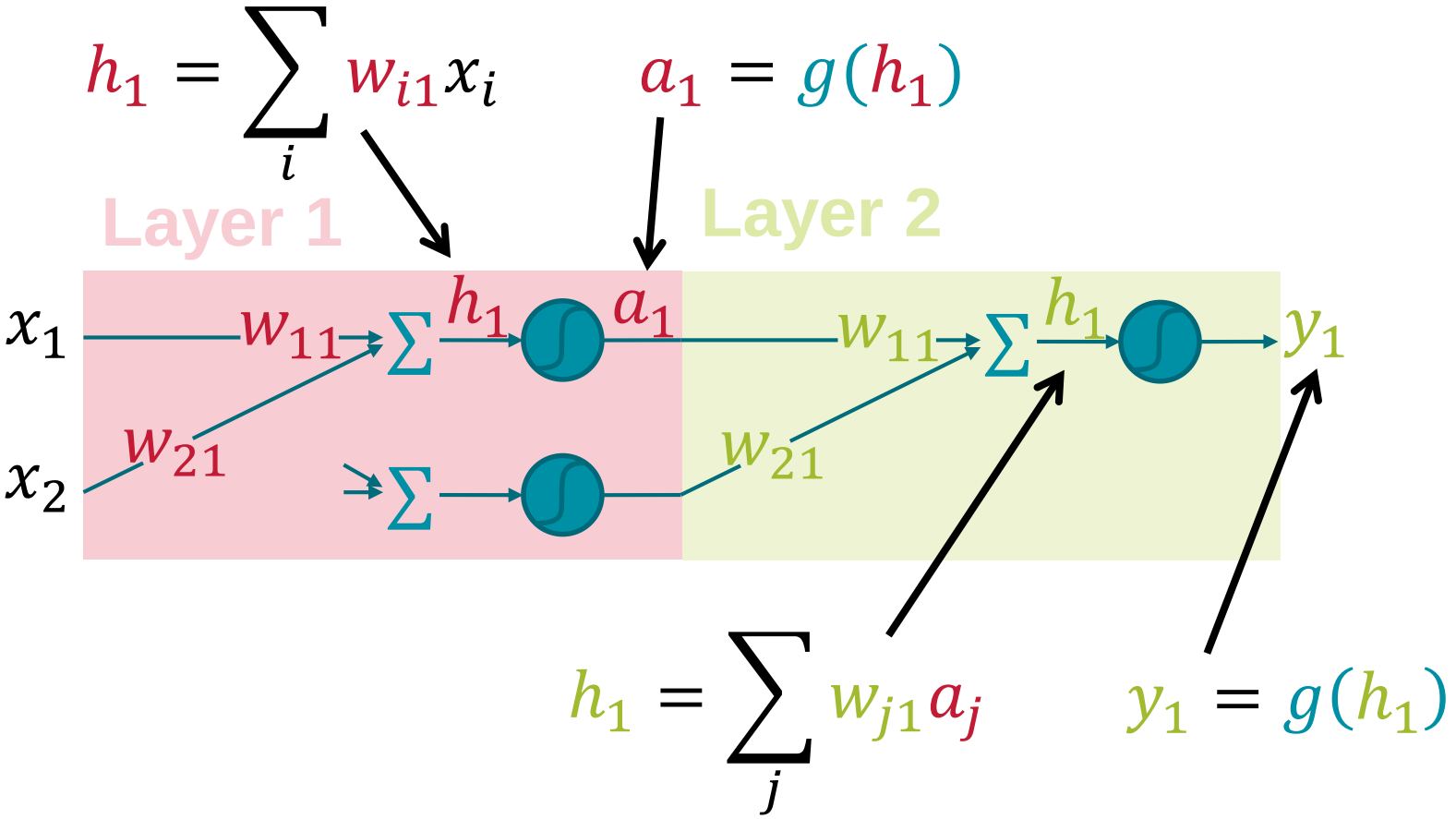
$$h_1 = \sum_i w_{i1} x_i \qquad a_1 = g(h_1)$$



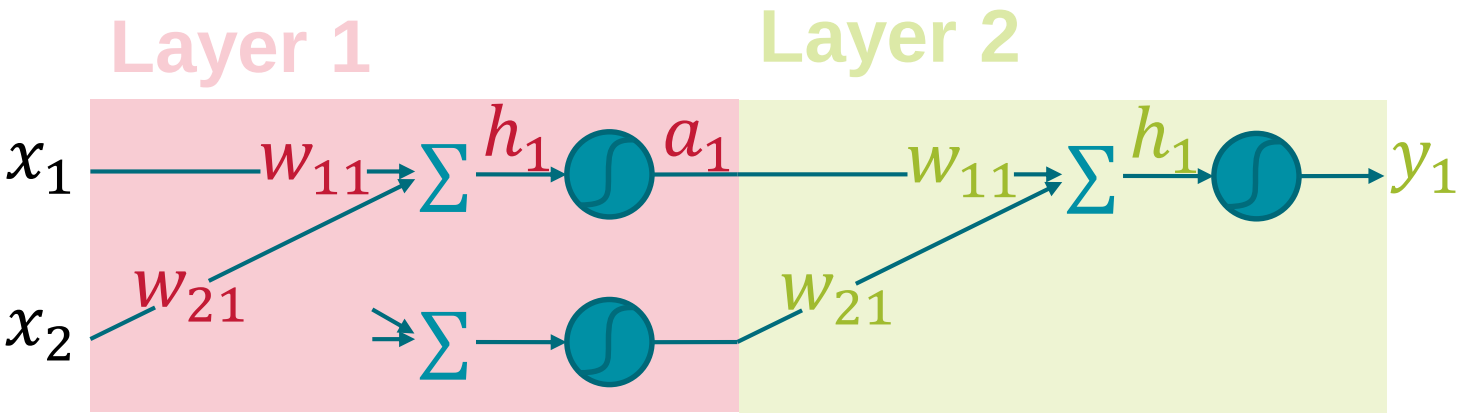
Feed Forward



Feed Forward

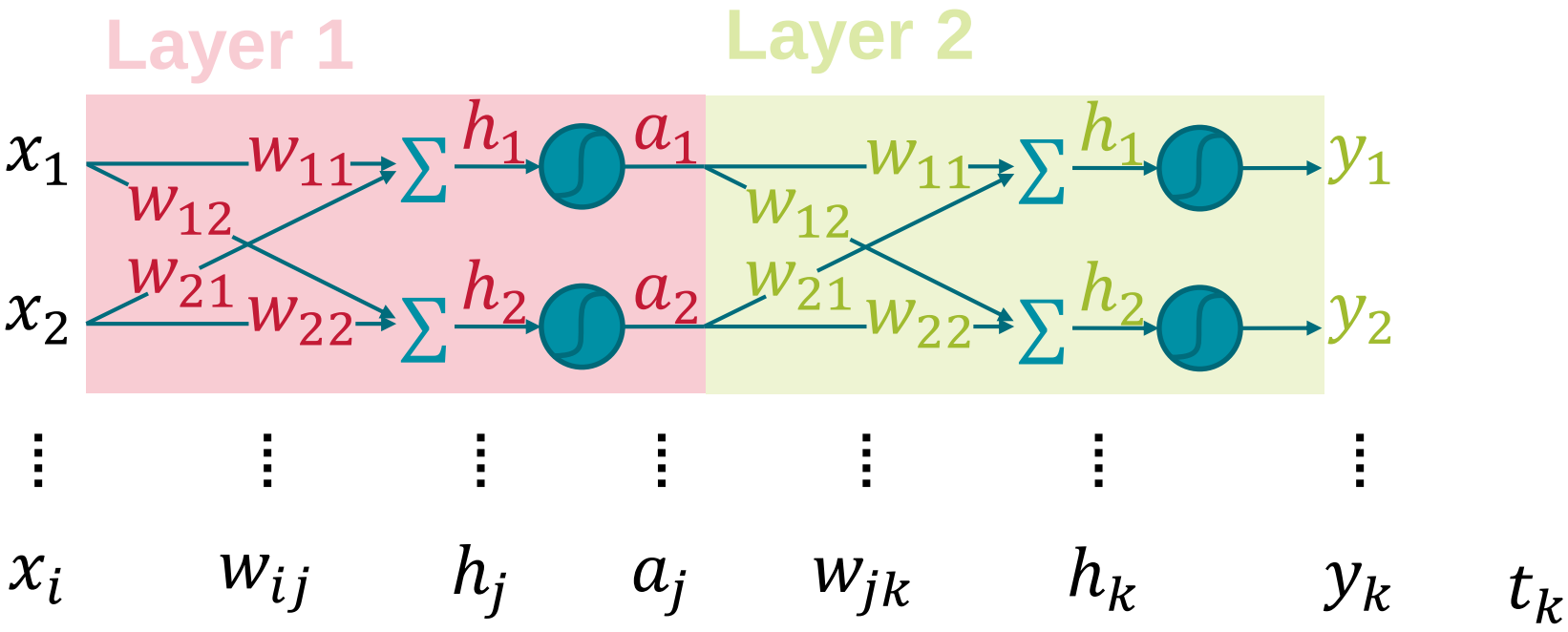


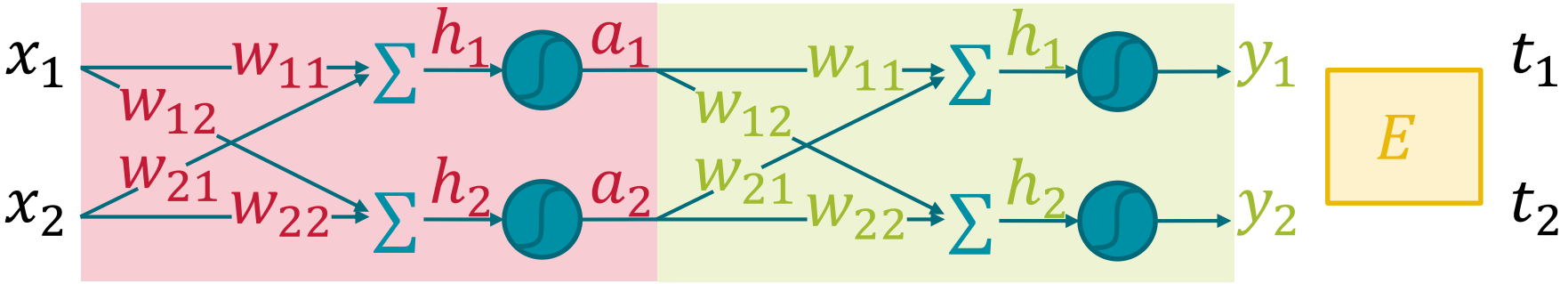
Feed Forward

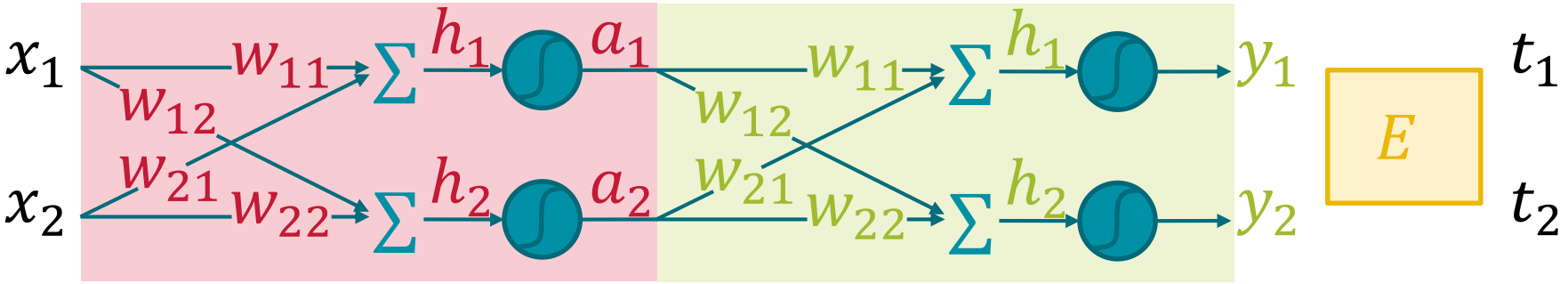


$$y_1 = g\left(\sum_j w_{j1}\, g\left(\sum_i w_{ij} x_i\right)\right)$$

Komplettansicht mit Variablen




x_1
x_2
w_{11}
w_{12}
w_{21}
w_{22}
h_1
h_2
a_1
a_2
y_1
y_2
E
t_1
t_2



$$E(\boldsymbol{w}, \boldsymbol{w}) = \frac{1}{2}\sum_k (t_k - y_k)^2$$

Partielle Ableitungen

Wir suchen

- Einfluss jedes Gewichtes auf den Fehler
- Richtung und Stärke der Steigung

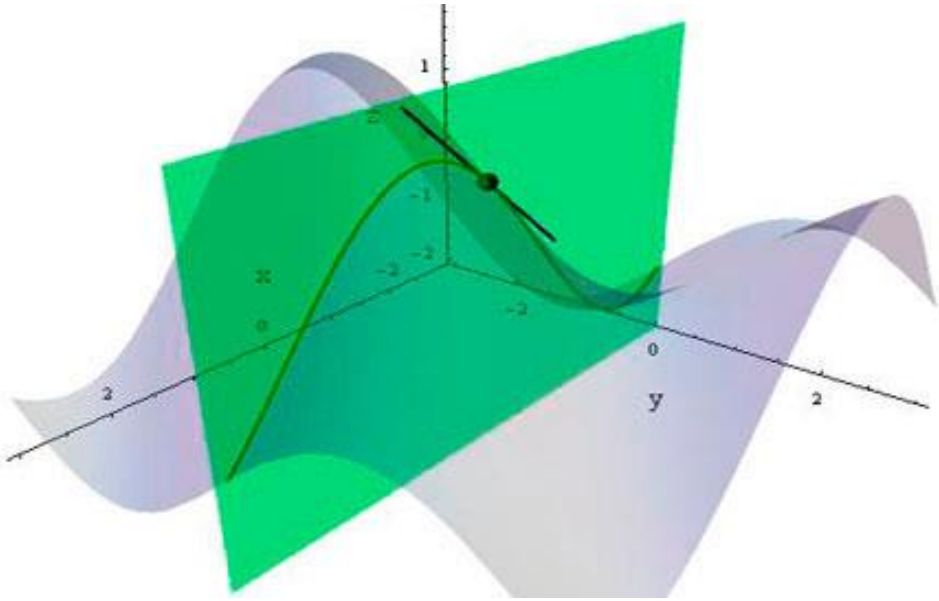
Vorgehen

- Partielle Ableitungen

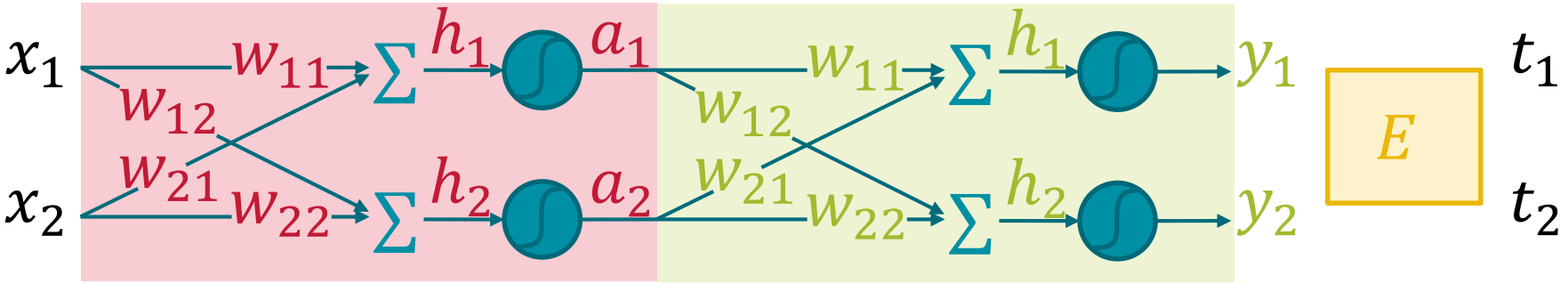
- Komplexer als für nur einen Layer
- **Aber**: Wichtige Eigenschaft

Kleine Auffrischung über Ableitungen

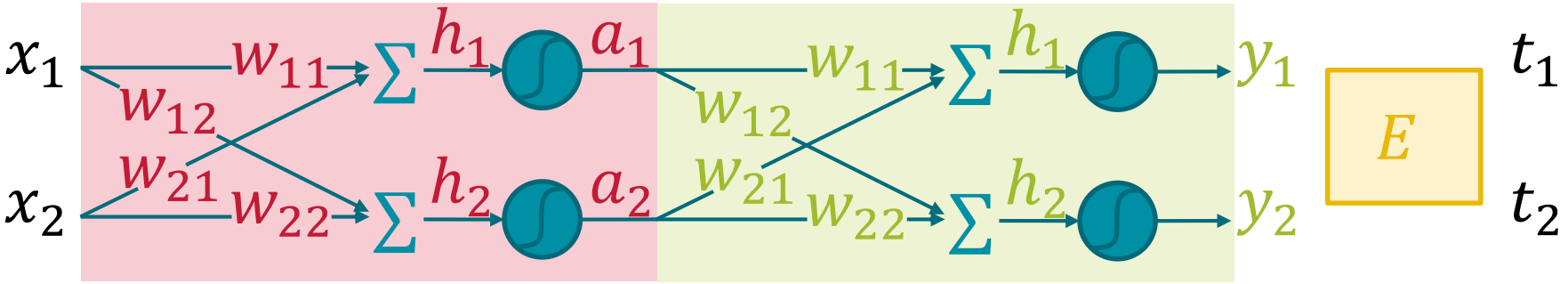
- $f(x) = \frac{1}{2}x^2 \rightarrow f'(x) = x$
- $\frac{df}{dx} = 0$, wenn f keine Funktion von x ist
- $\frac{df}{dx} = \frac{df}{dt}\frac{dt}{dx}$, die Kettenregel
- $\frac{\partial f}{\partial x}$, Partielle Ableitung



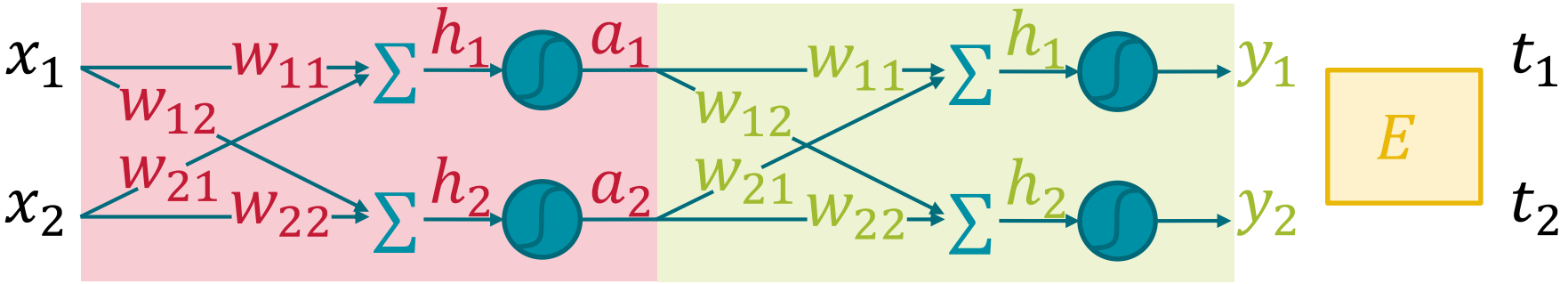
Bildquelle: alamo.edu



$$\frac{\partial E}{\partial w_{jk}}$$

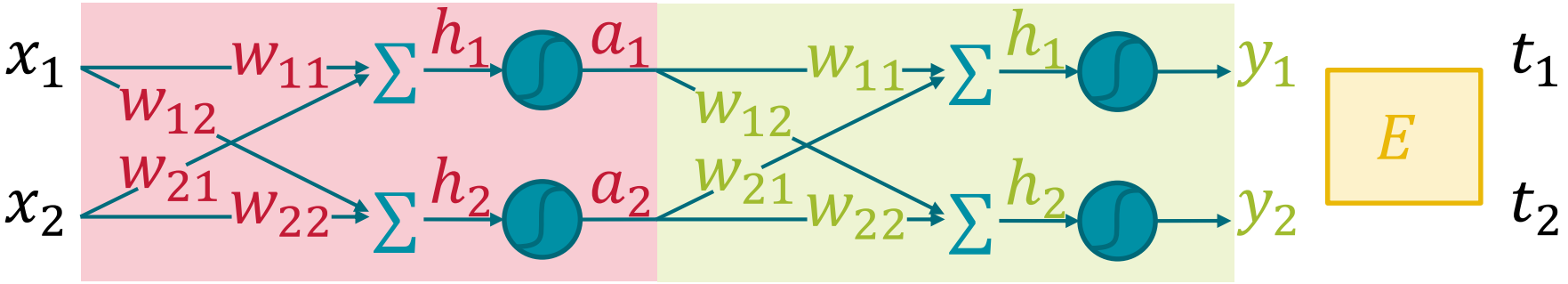


$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial h_k} \frac{\partial h_k}{\partial w_{jk}}$$



$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial h_k}\frac{\partial h_k}{\partial w_{jk}}$$

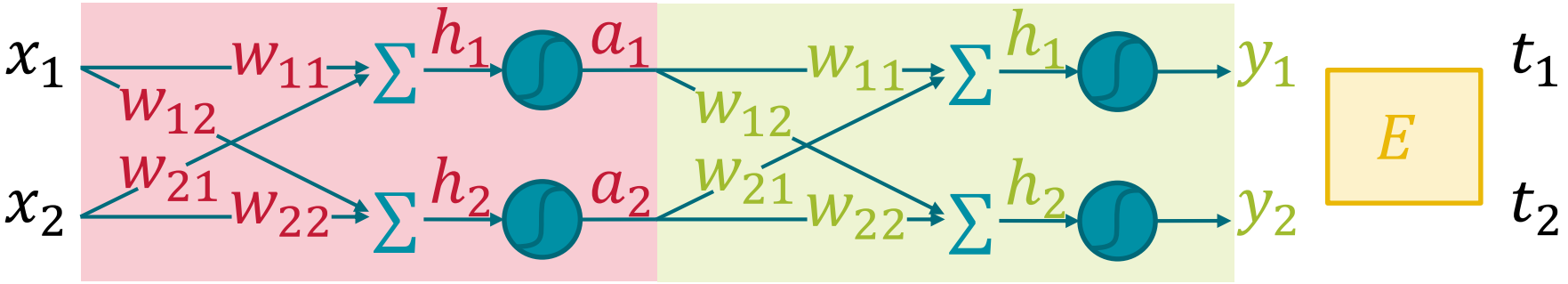
$$\frac{\partial h_k}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}}\sum a_l w_{lk}$$



$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial h_k}\frac{\partial h_k}{\partial w_{jk}}$$

$$\frac{\partial h_k}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}}\sum a_l w_{lk}$$

$$\frac{\partial}{\partial w_{jk}} a_1 w_{1k} + a_2 w_{2k} + \cdots$$

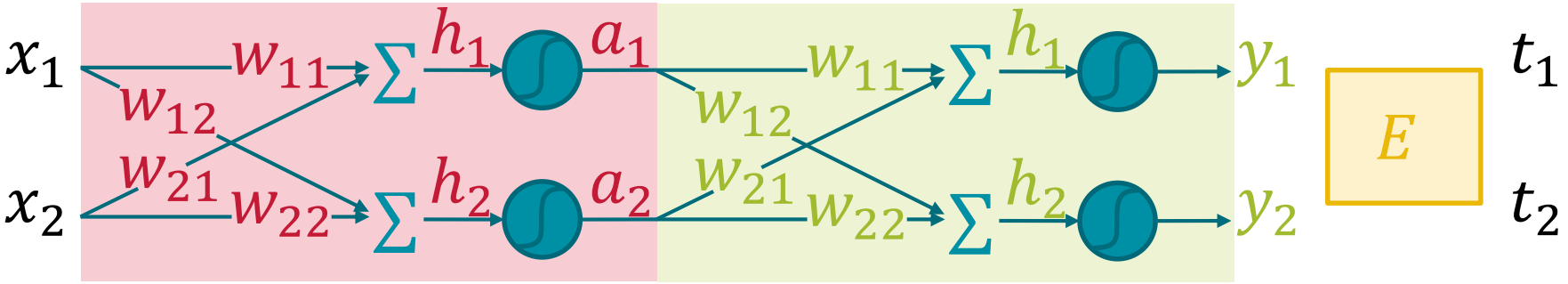


$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial h_k}\frac{\partial h_k}{\partial w_{jk}}$$

$$\frac{\partial h_k}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}}\sum a_l w_{lk}$$

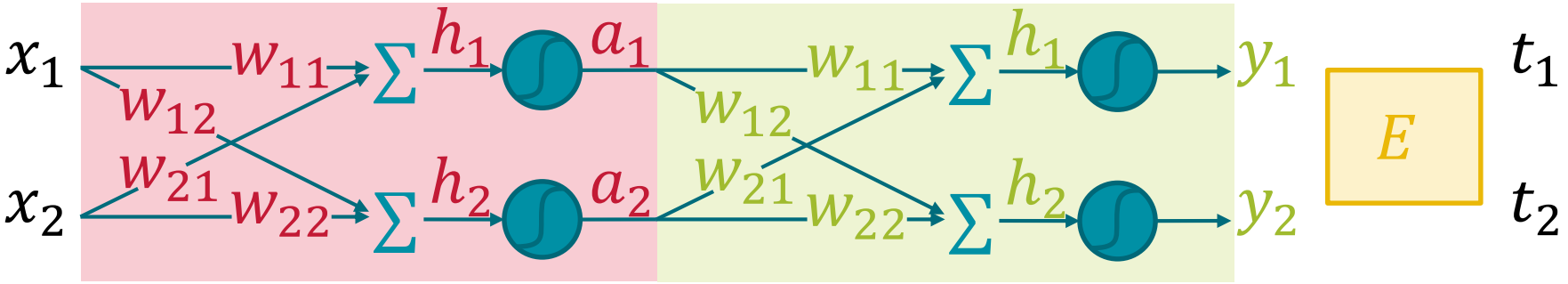
$$\frac{\partial}{\partial w_{jk}} a_1 w_{1k} + a_2 w_{2k} + \cdots$$

$$a_j$$

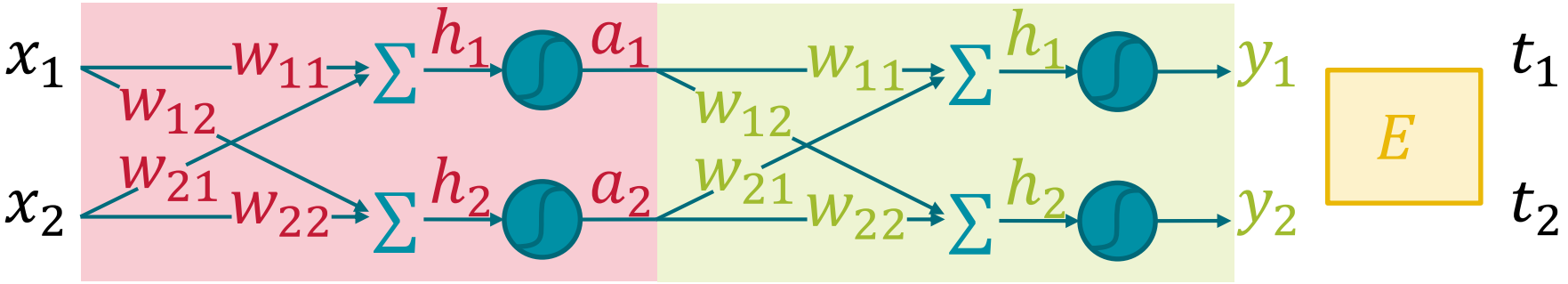


$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial h_k} a_j$$

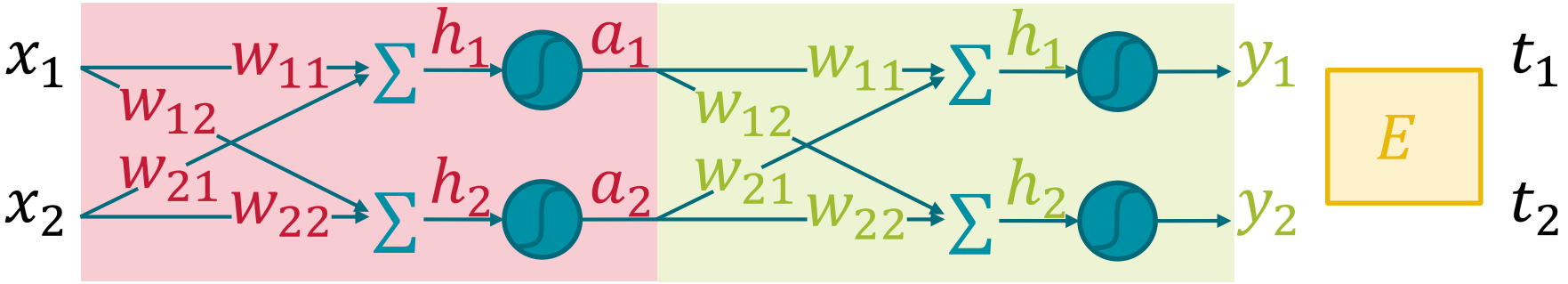
$$\frac{\partial h_k}{\partial w_{jk}} = a_j$$



$$\frac{\partial E}{\partial w_{jk}} = a_j \frac{\partial E}{\partial h_k}$$

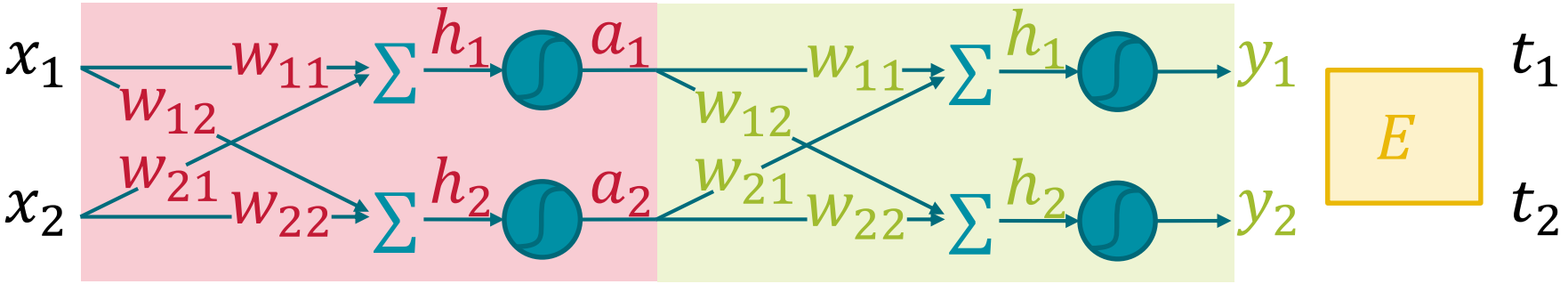


$$\frac{\partial E}{\partial w_{jk}} = a_j \frac{\partial E}{\partial h_k} = a_j \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial h_k}$$



$$\frac{\partial E}{\partial w_{jk}} = a_j \frac{\partial E}{\partial h_k} = a_j \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial h_k}$$

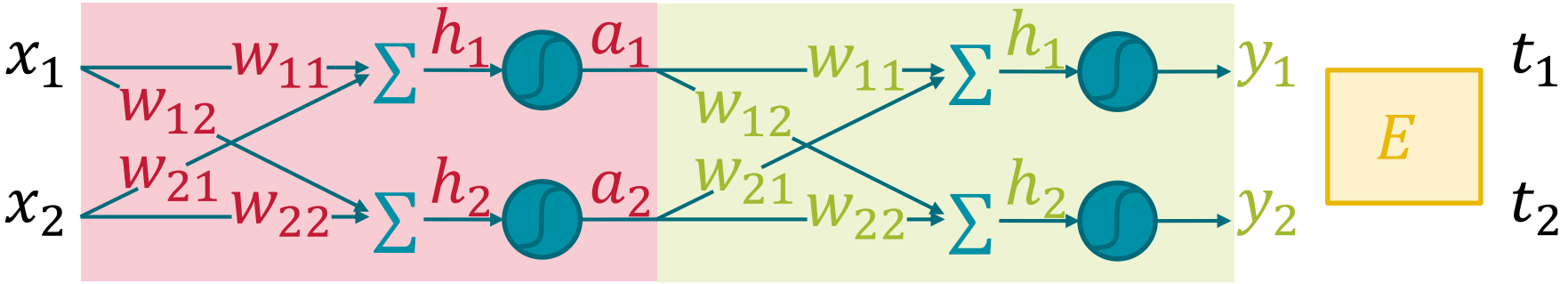
$$\frac{\partial E}{\partial y_k} = \frac{\partial}{\partial y_k} \frac{1}{2} \sum (t_k - y_k)^2$$



$$\frac{\partial E}{\partial w_{jk}} = a_j \frac{\partial E}{\partial h_k} = a_j \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial h_k}$$

$$\frac{\partial E}{\partial y_k} = \frac{\partial}{\partial y_k} \frac{1}{2} \sum (t_k - y_k)^2$$

$$\frac{\partial}{\partial y_k} \frac{1}{2} [(t_1 - y_1)^2 + (t_2 - y_2)^2 + \cdots]$$

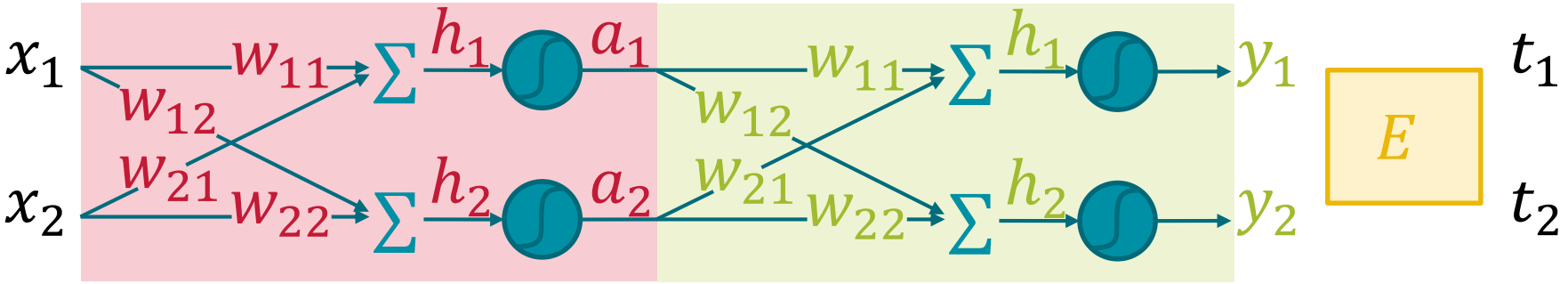


$$\frac{\partial E}{\partial w_{jk}} = a_j \frac{\partial E}{\partial h_k} = a_j \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial h_k}$$

$$\frac{\partial E}{\partial y_k} = \frac{\partial}{\partial y_k} \frac{1}{2} \sum (t_k - y_k)^2$$

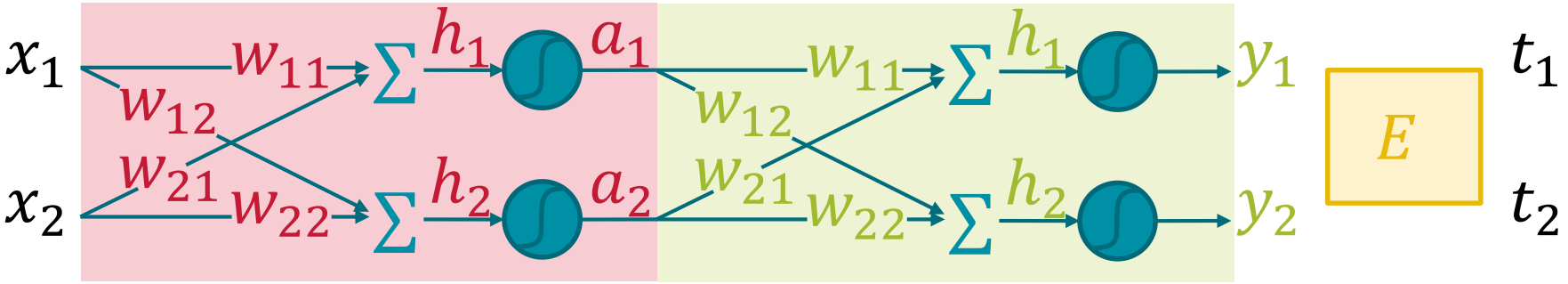
$$\frac{\partial}{\partial y_k} \frac{1}{2} [(t_1 - y_1)^2 + (t_2 - y_2)^2 + \cdots]$$

$$t_k - y_k$$

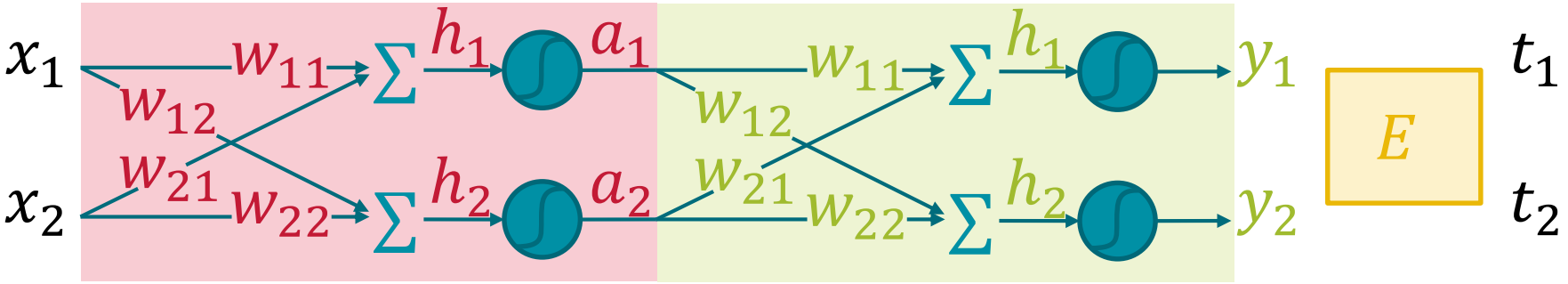


$$\frac{\partial E}{\partial w_{jk}} = a_j \frac{\partial E}{\partial h_k} = a_j (t_k - y_k) \frac{\partial y_k}{\partial h_k}$$

$$\frac{\partial E}{\partial y_k} = t_k - y_k$$

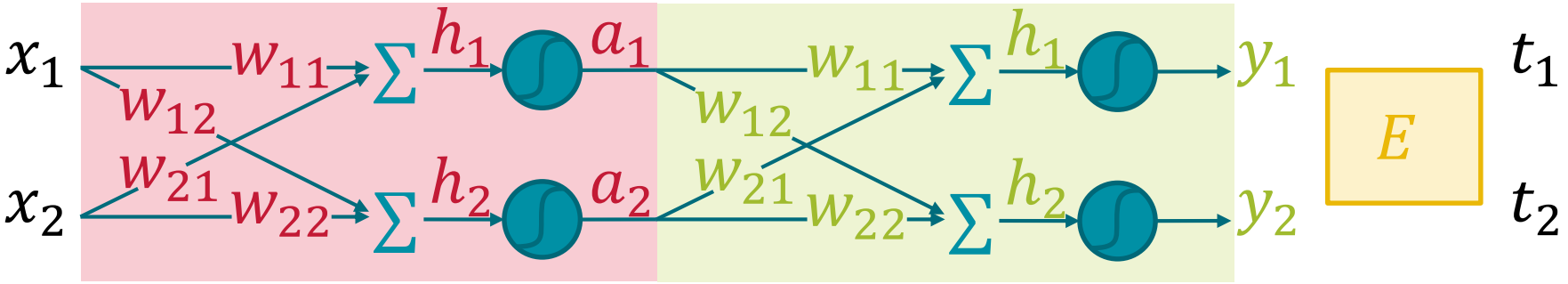


$$\frac{\partial E}{\partial w_{jk}} = a_j(t_k - y_k)\frac{\partial y_k}{\partial h_k}$$



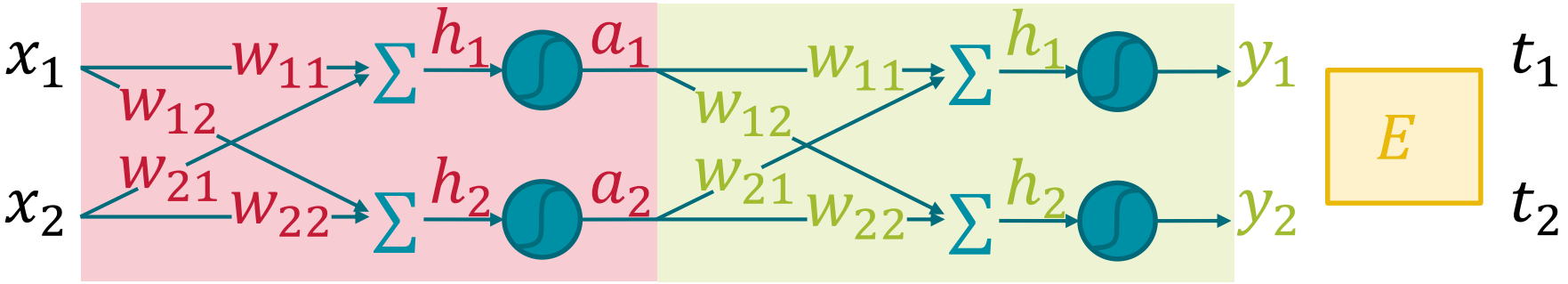
$$\frac{\partial E}{\partial w_{jk}} = a_j(t_k - y_k)\frac{\partial y_k}{\partial h_k}$$

$$\frac{\partial y_k}{\partial h_k} = g'(h_k)$$

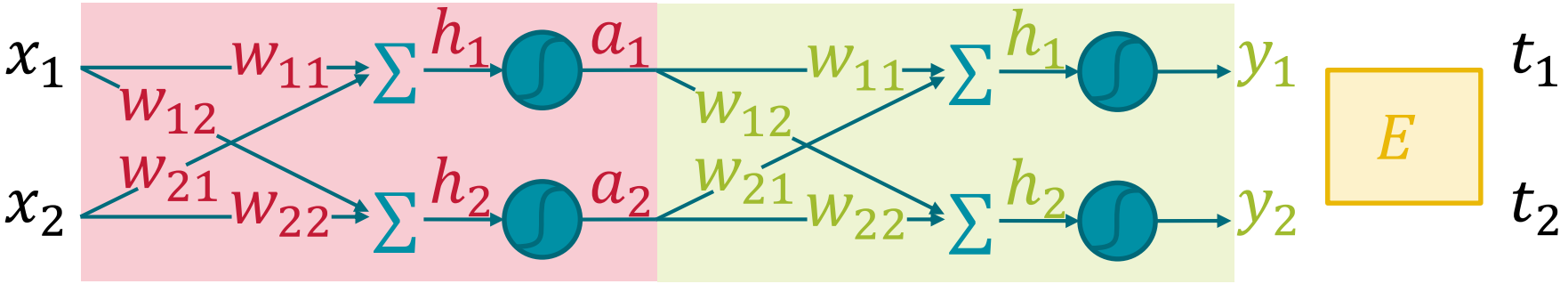


$$\frac{\partial E}{\partial w_{jk}} = a_j(t_k - y_k)g'(h_k)$$

$$\frac{\partial y_k}{\partial h_k} = g'(h_k)$$



$$\frac{\partial E}{\partial w_{jk}} = a_j(t_k - y_k)g'(h_k)$$

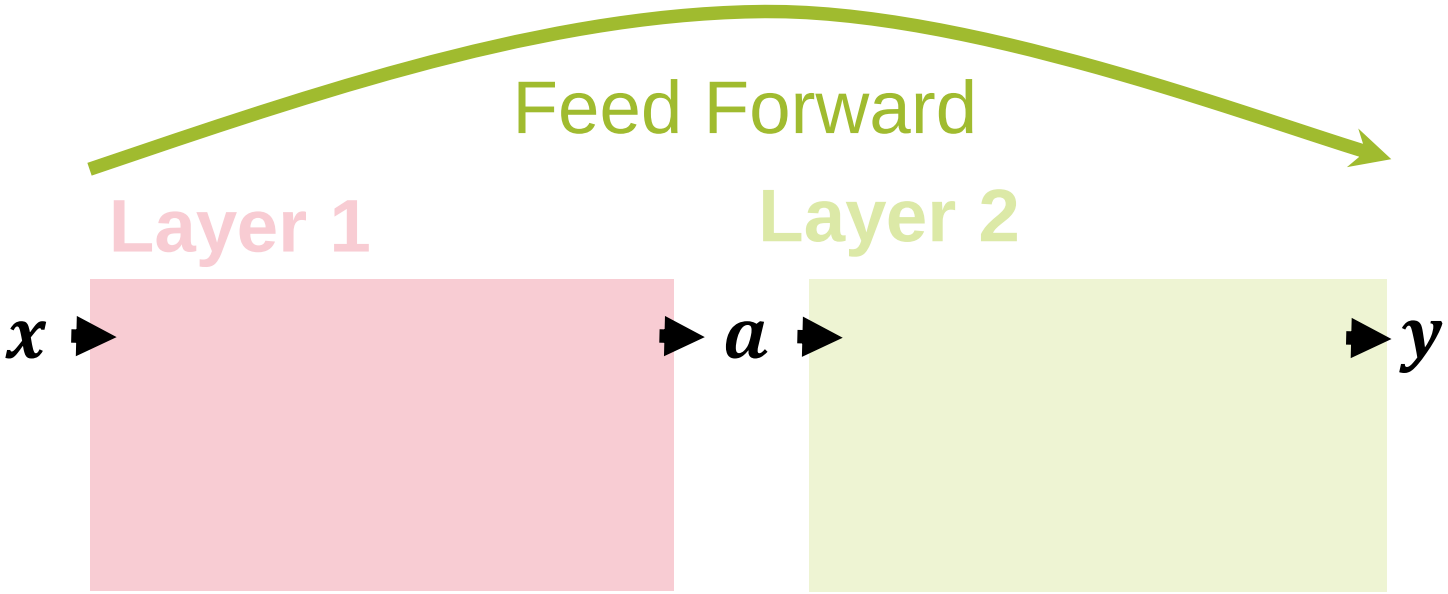


$$\frac{\partial E}{\partial w_{jk}} = a_j (t_k - y_k) g'(h_k)$$

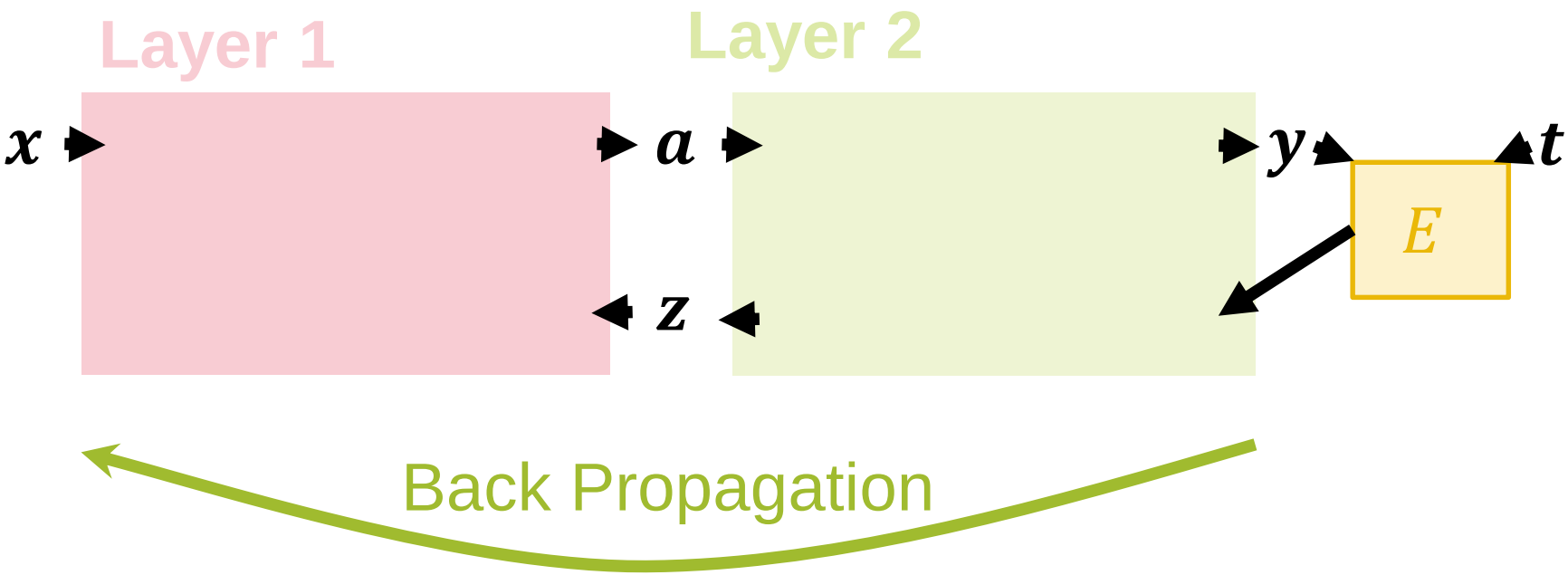
$$\frac{\partial E}{\partial w_{ij}} = x_i \left(\sum \delta_{L2} w_{jk} \right) g'(h_j)$$

δ_{L1}

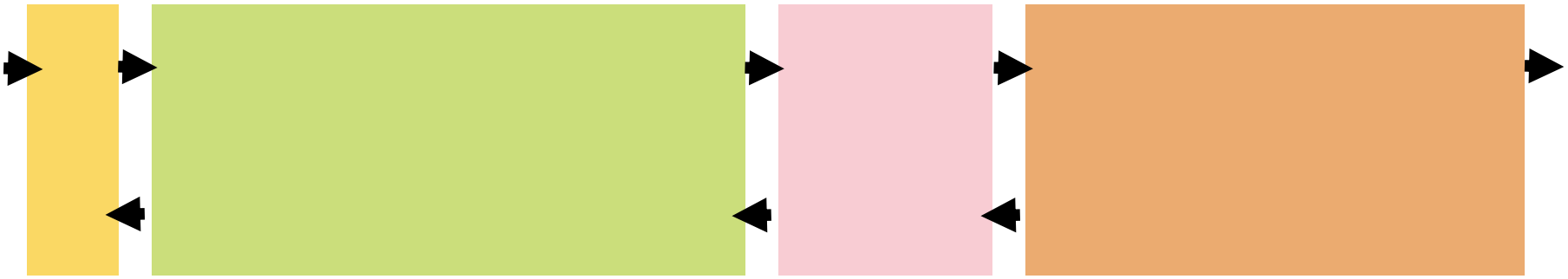
The Big Picture



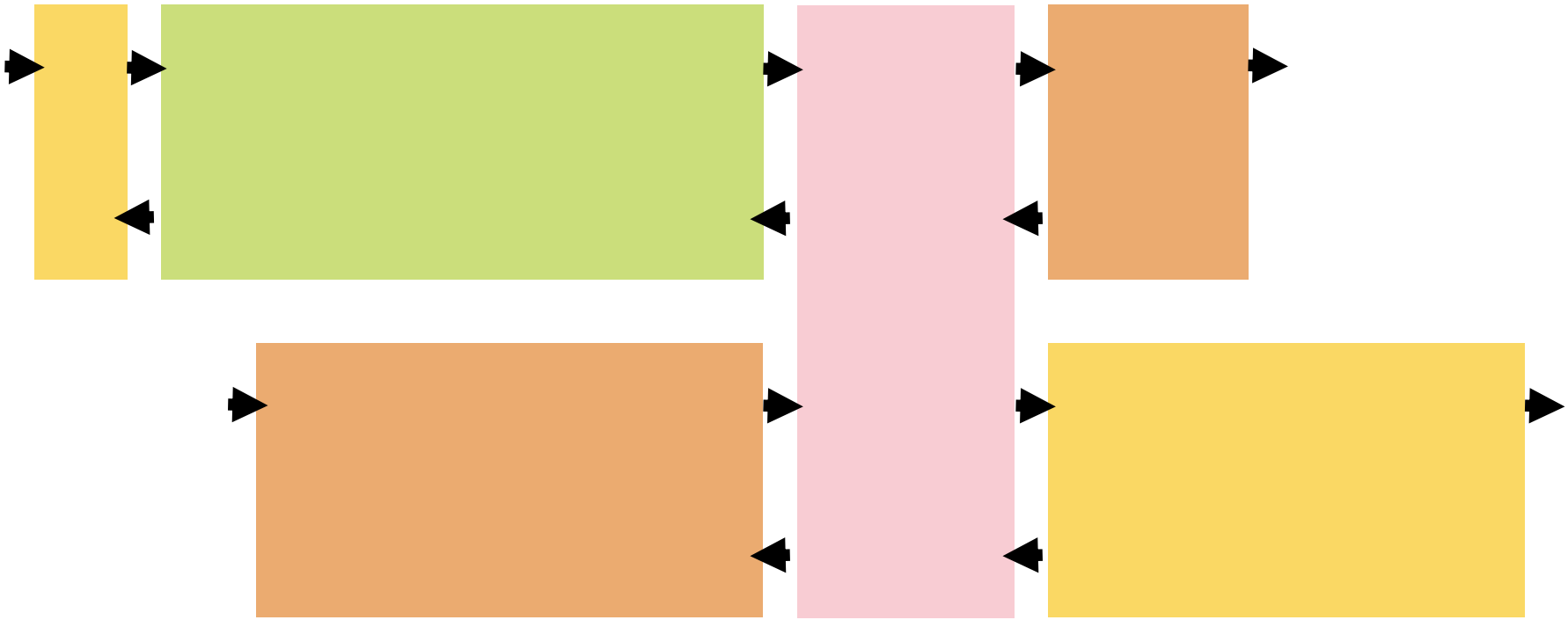
The Big Picture



Das Baukastenprinzip



Das Baukastenprinzip



Convolutional Neural Networks

Beispielsproblem Objekterkennung

Akkordeon

Hund

Hauskatze

ImageNet Datenbank

Problematik:

- Was bezeichnet einen Hund?
- Was unterscheidet Hund und Akkordeon?
- Was unterscheidet Hund und Hauskatze?

Bildquelle: www.image-net.org

Verteilte Repräsentation

Lokale Repräsentation

- Eine dedizierte Darstellung für jedes Konzept (z.B. Auto, Hund)

Verteilte Repräsentation

- Viele-zu-viele Beziehung zwischen kleinen Einheiten
- Die Konzepte aus ihren Einzelteilen zusammensetzen
- Einzelteile sind Sub-Konzepte
- Können geteilt werden, d.h. weniger Repräsentationen notwendig

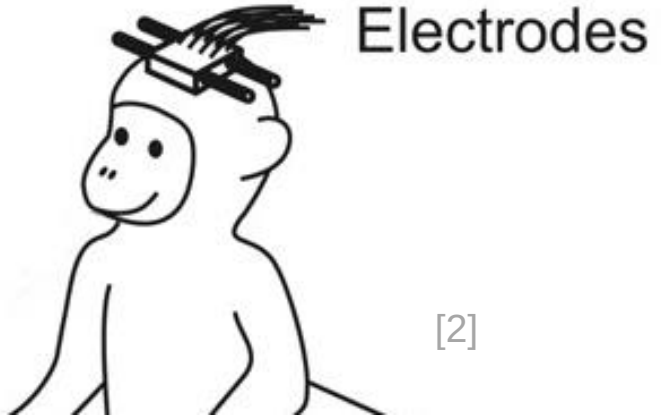
Beispiel: Zerlegung eines Textes

Paragraph	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur lacinia, purus sit amet malesuada accumsan, tellus odio feugiat velit, luctus dapibus odio leo at nisi. Etiam dapibus finibus nunc, sit amet consequat purus feugiat sed. Nunc vulputate dolor a odio egestas tristique. Proin vel efficitur orci, vitae imperdiet leo. Aenean ut nunc in lorem interdum lacinia. Sed condimentum leo scelerisque congue consequat. Aliquam erat volutpat. Phasellus maximus tempus turpis in pretium.
Satz	Nunc vulputate dolor a odio egestas tristique.
Wort	dolor
Buchstabe	o
Phoneme	/o/

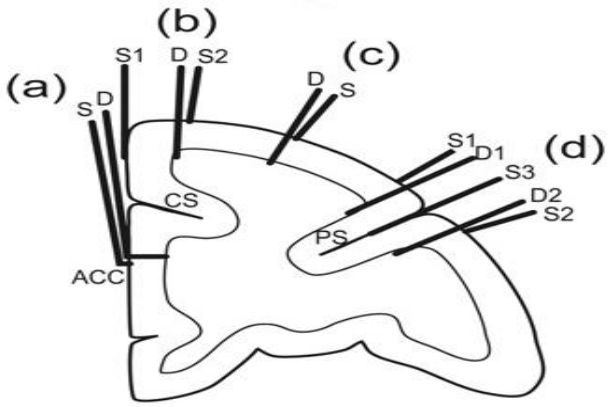
- Praktisch für z.B. Übersetzungen
- Aber für Bilder existieren keine solchen Konzepthierachien

Der Visual Cortex

- Forschung von Hubel and Wiesel (1968)[1] am Visuellen Cortex von Affen
- Abtastung der Neuronenaktivierung durch Elektroden
- Kontrollierte visuelle Stimulation



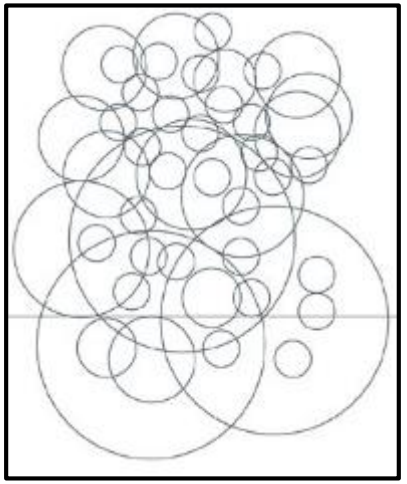
[2]



[2]

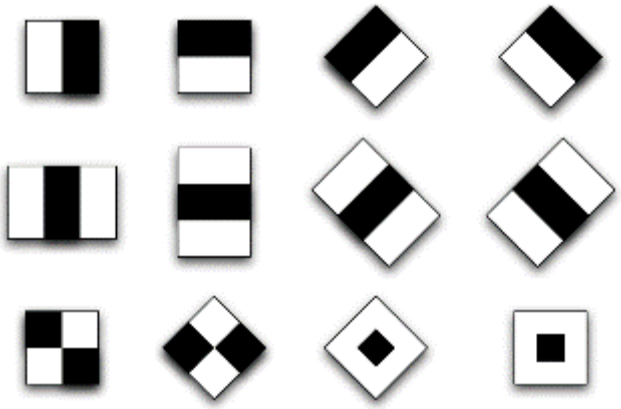
[1] Hubel and Wiesel (1968). Receptive fields and functional architecture of monkey striate cortex. Journal of Physiology
[2] Tsujimoto, Shimazu and Isomura (2006), “Direct Recording of Theta Oscillations in Primate Prefrontal and Anterior Cingulate Cortices”, Journal of Neurophysiology

Der Visual Cortex



[1]

- Zellstrukturen im Visuellen Cortex sind für Unterbereiche des Sichtfeldes zuständig, ihr sog. Receptive Field
- Viele, überlappende Strukturen decken das gesammte Sichtfeld ab
- Diese Strukturen führen Pattern Recognition durch
- Meisten einfache Kanten- und Blob-Detektoren
- Können wir aus dieser Erkenntnis lernen?



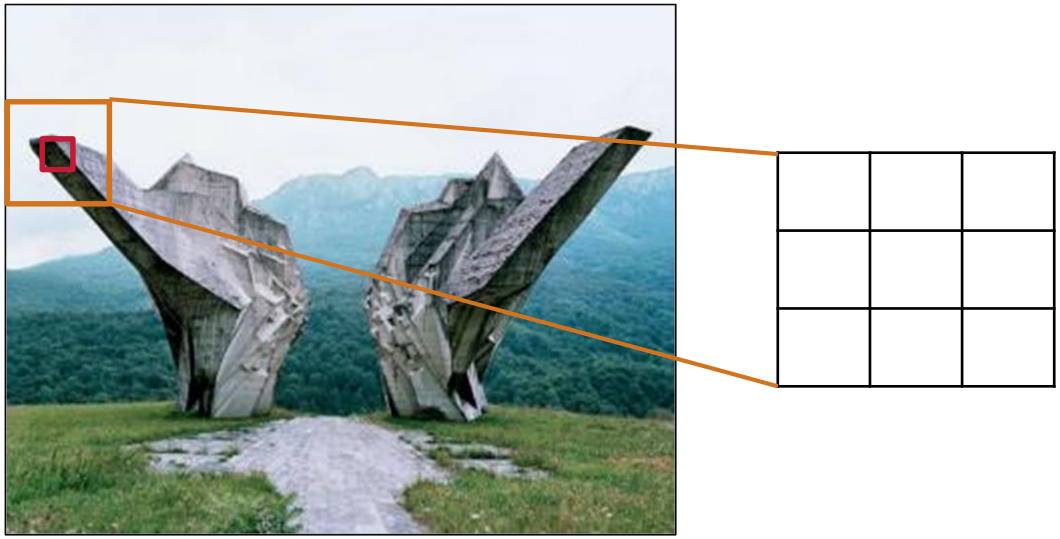
Bildquelle: http://fileadmin.cs.lth.se/graphics/theses/projects/facerecognition/

[1] Wandell and Smirnakis (2009) „Plasticity and stability of visual field maps in adult primary visual cortex", Nature Reviews Neuroscience

Convolution? Was ist das?

Bildquelle: http://cs.brown.edu/courses/cs129/results/proj3/rkuppig/

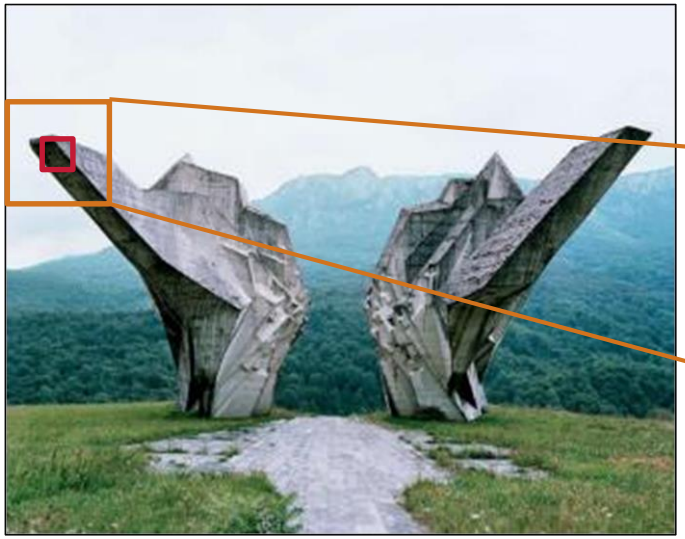
Convolution? Was ist das?



Bild

Bildquelle: http://cs.brown.edu/courses/cs129/results/proj3/rkuppig/

Convolution? Was ist das?



Bild

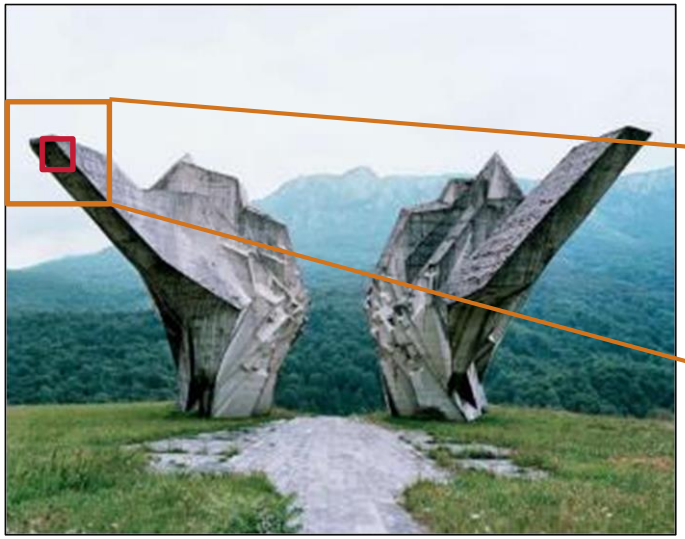
Sobel

-1	0	+1
-2	0	+2
-1	0	+1

Kernel

Bildquelle: http://cs.brown.edu/courses/cs129/results/proj3/rkuppig/

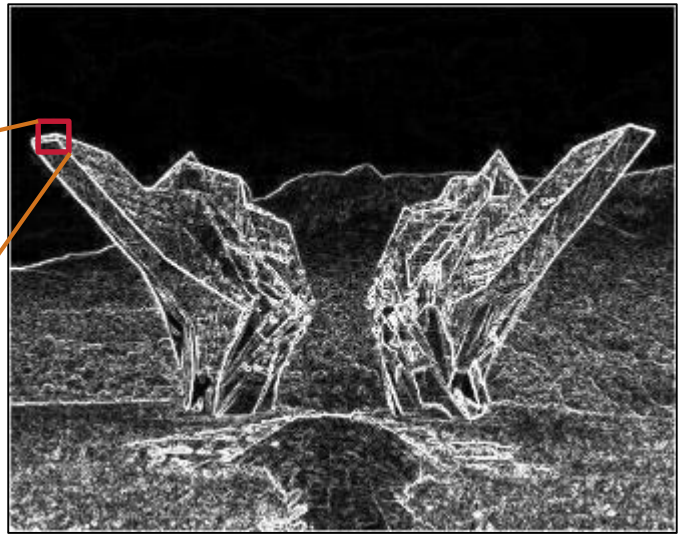
Convolution? Was ist das?



Sobel

-1	0	+1
-2	0	+2
-1	0	+1

Kernel

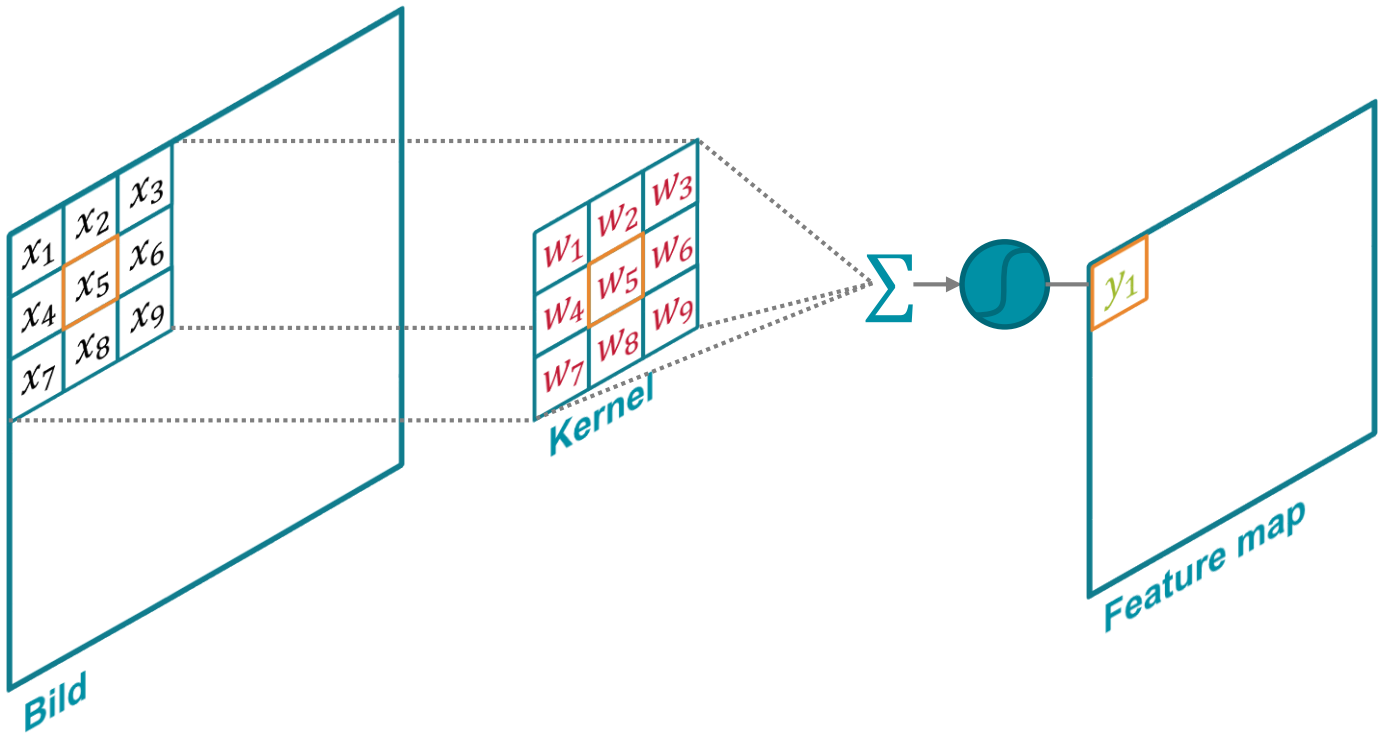


Bild

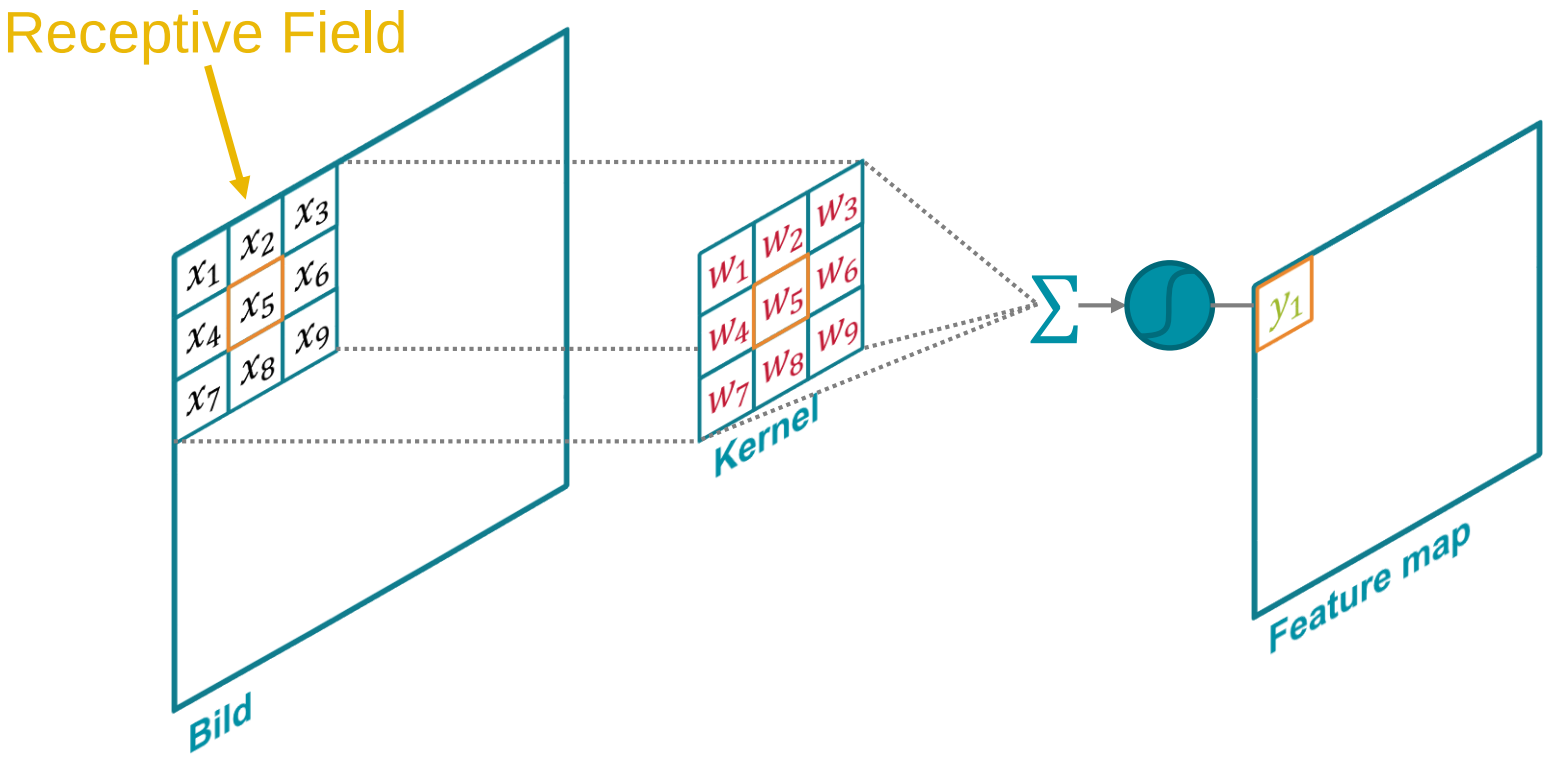
Feature map

Bildquelle: http://cs.brown.edu/courses/cs129/results/proj3/rkuppig/

Eine andere Sicht auf Convolution



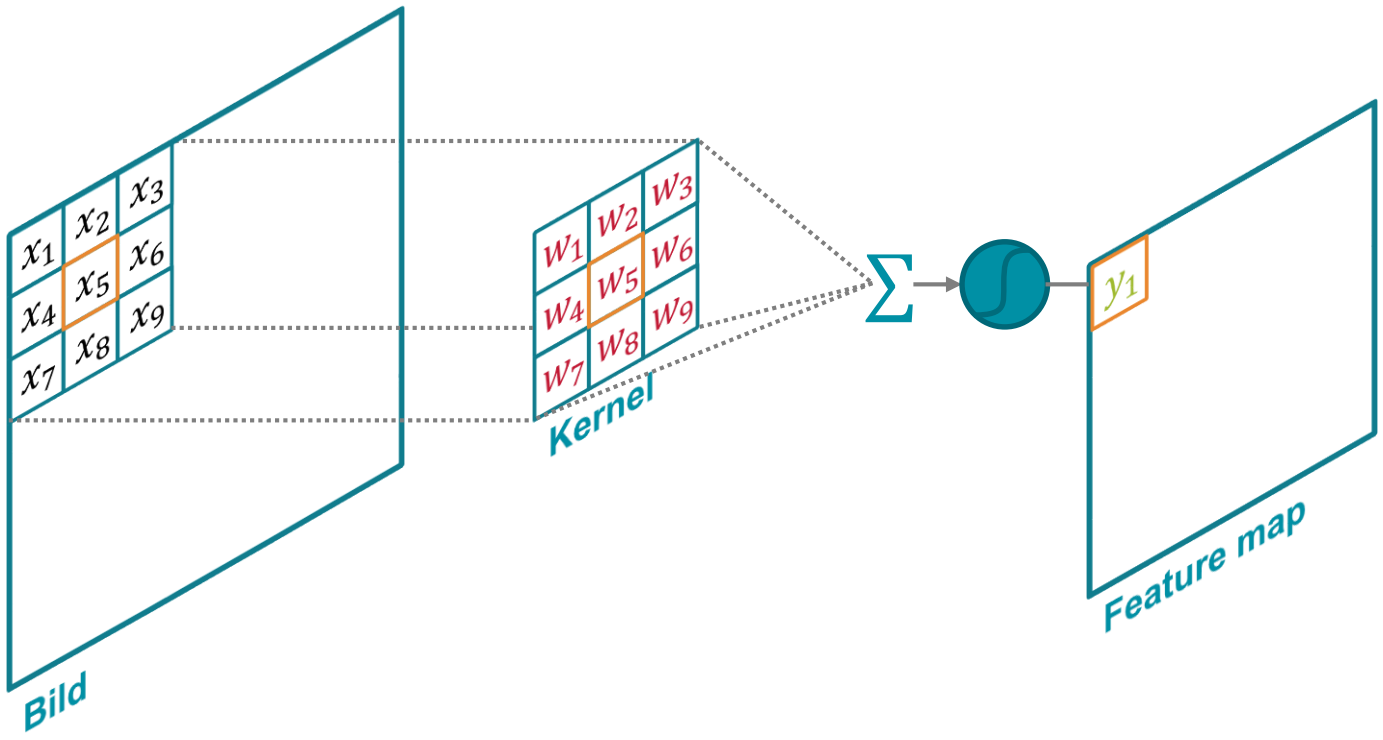
Eine andere Sicht auf Convolution



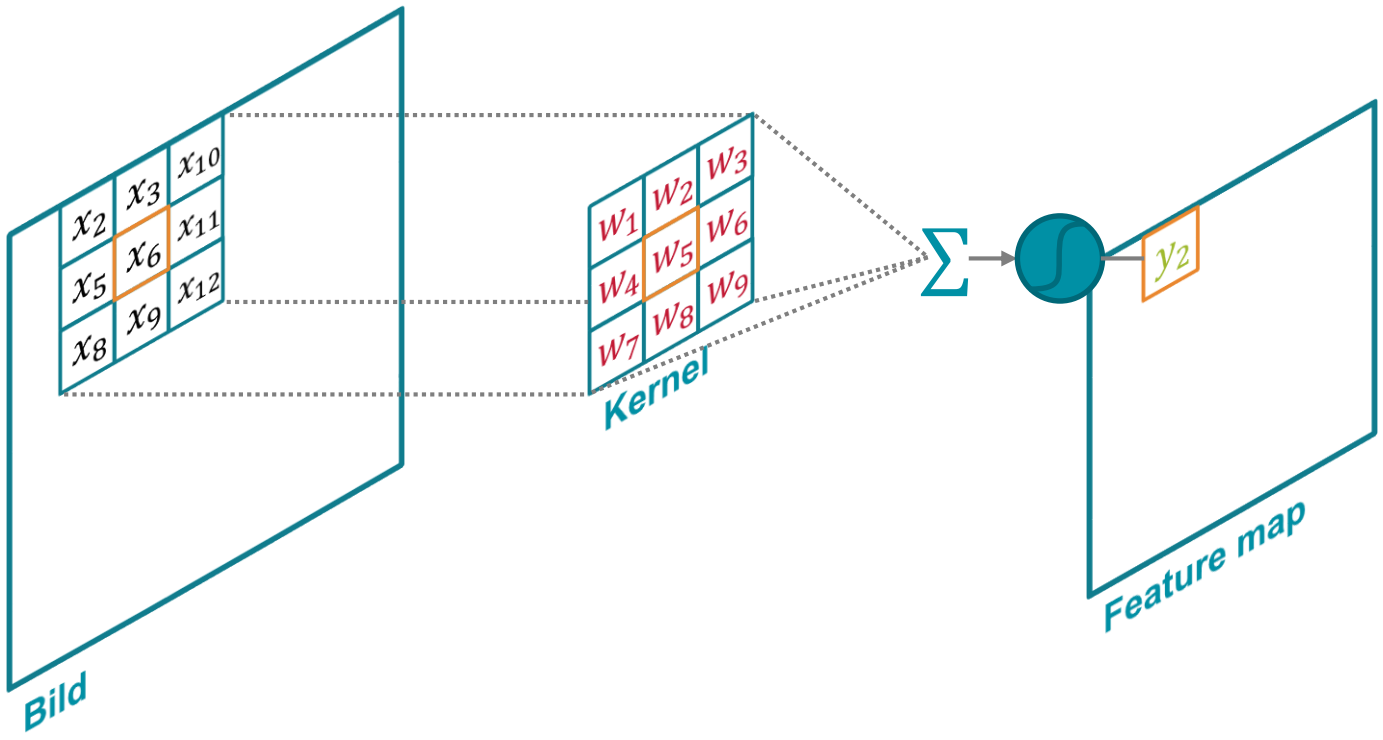
Zusammenfassung

- Convolution mit einem Kernel resultiert in eine Feature Map
- D.h. die Features, die dem Klassifizierer traditionellerweise übergeben werden
- Der Vorgang der Anwendung eines Kernels entspricht einem Neuron eines Neuronalen Netwerkes
- D.h. wir können den Prozess direkt in der Klassifizierer integrieren
- Und somit die besten Kernel für die angegangene Aufgabe automatisch lernen
- Dieser Lernprozess findet gemeinsam und gleichzeitig mit der Klassifikation statt
- **Fazit**: Out-of-the-Box Lösungen ohne Aufwand

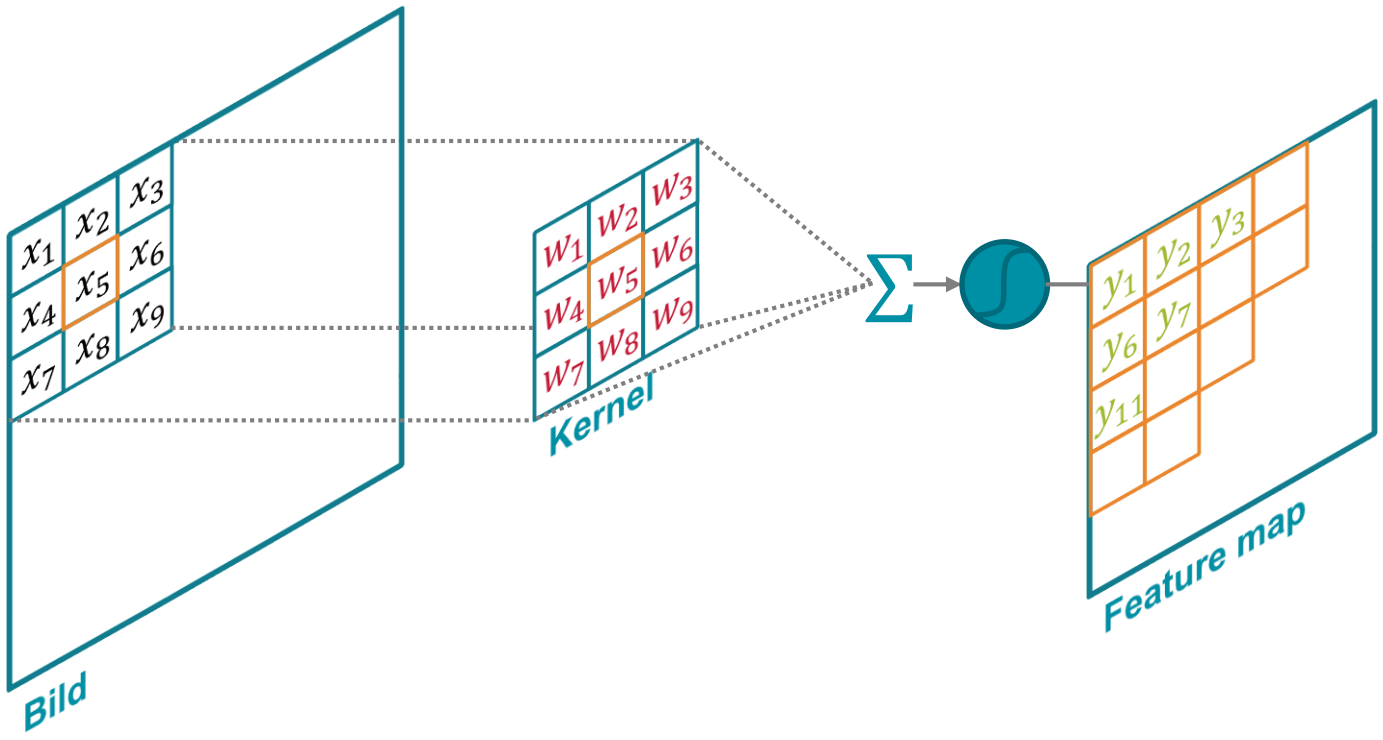
Shared Weights



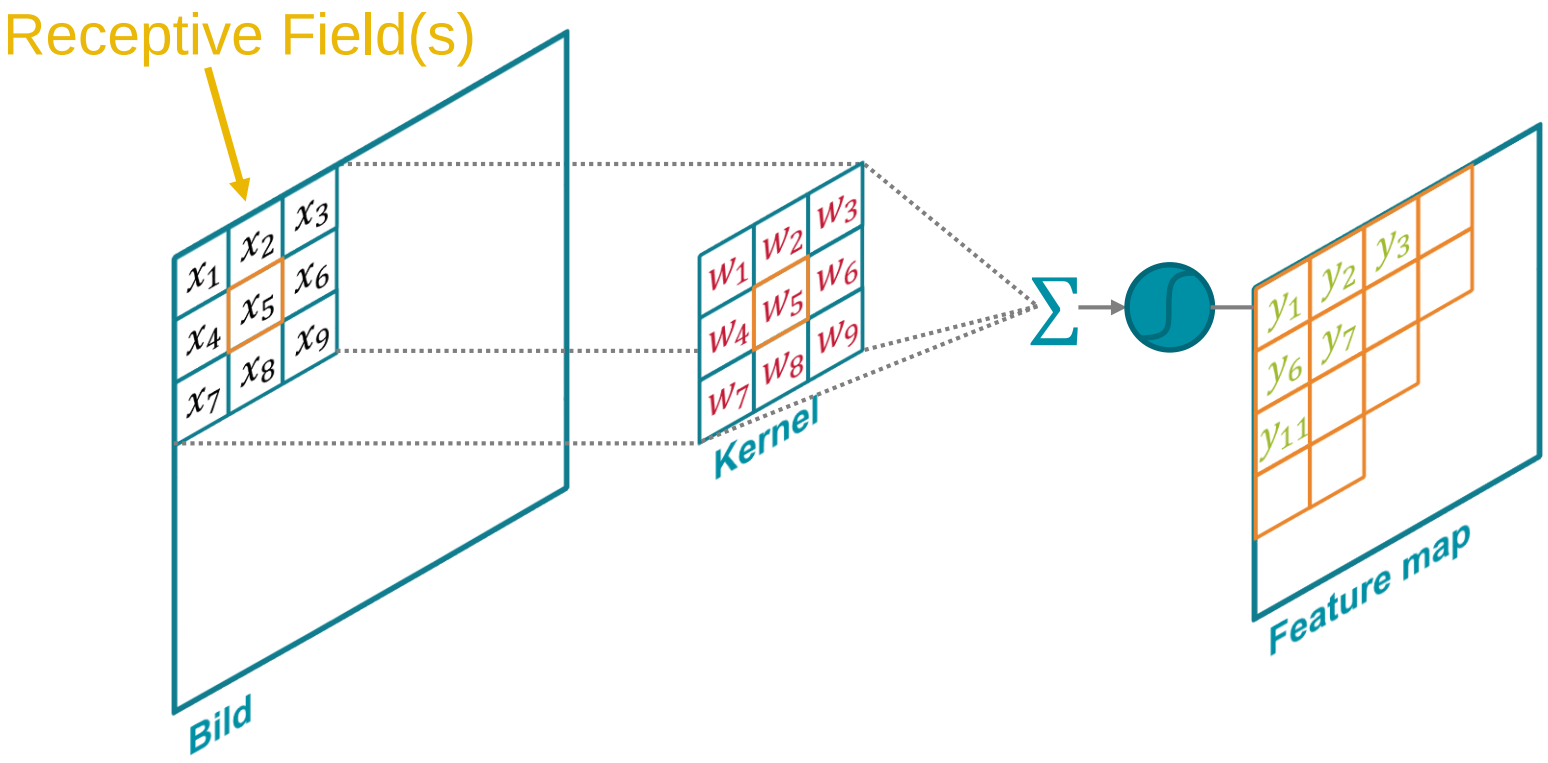
Shared Weights



Shared Weights



Shared Weights



Warum Convolutional Layer I

- Fully Connected Layer wäre ebenso möglich
- Ein solcher kann die Convolution imitieren und noch mehr
- Also warum nicht einen Fully Connected Layer?

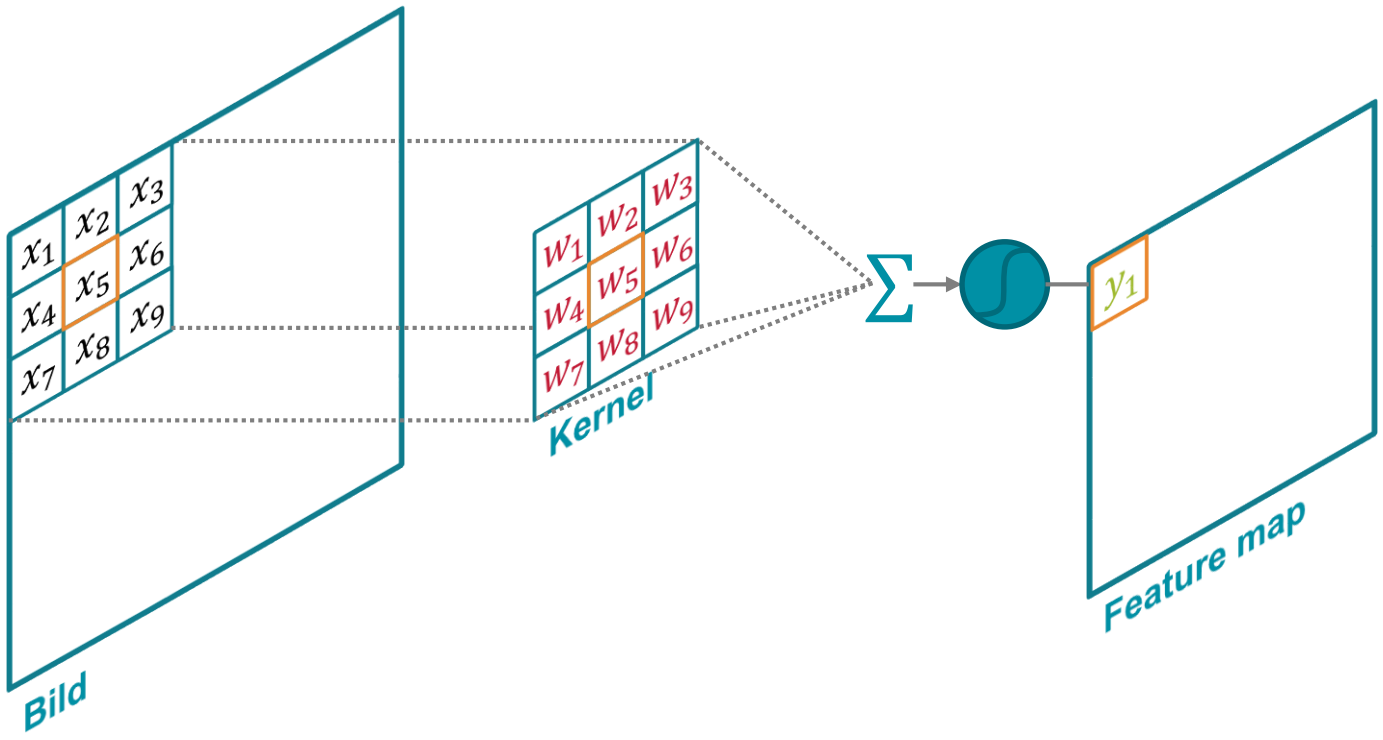
Vorteil 1: Regularisierung durch Translationsinvarianz

Vorteil 2: Weniger Parameter

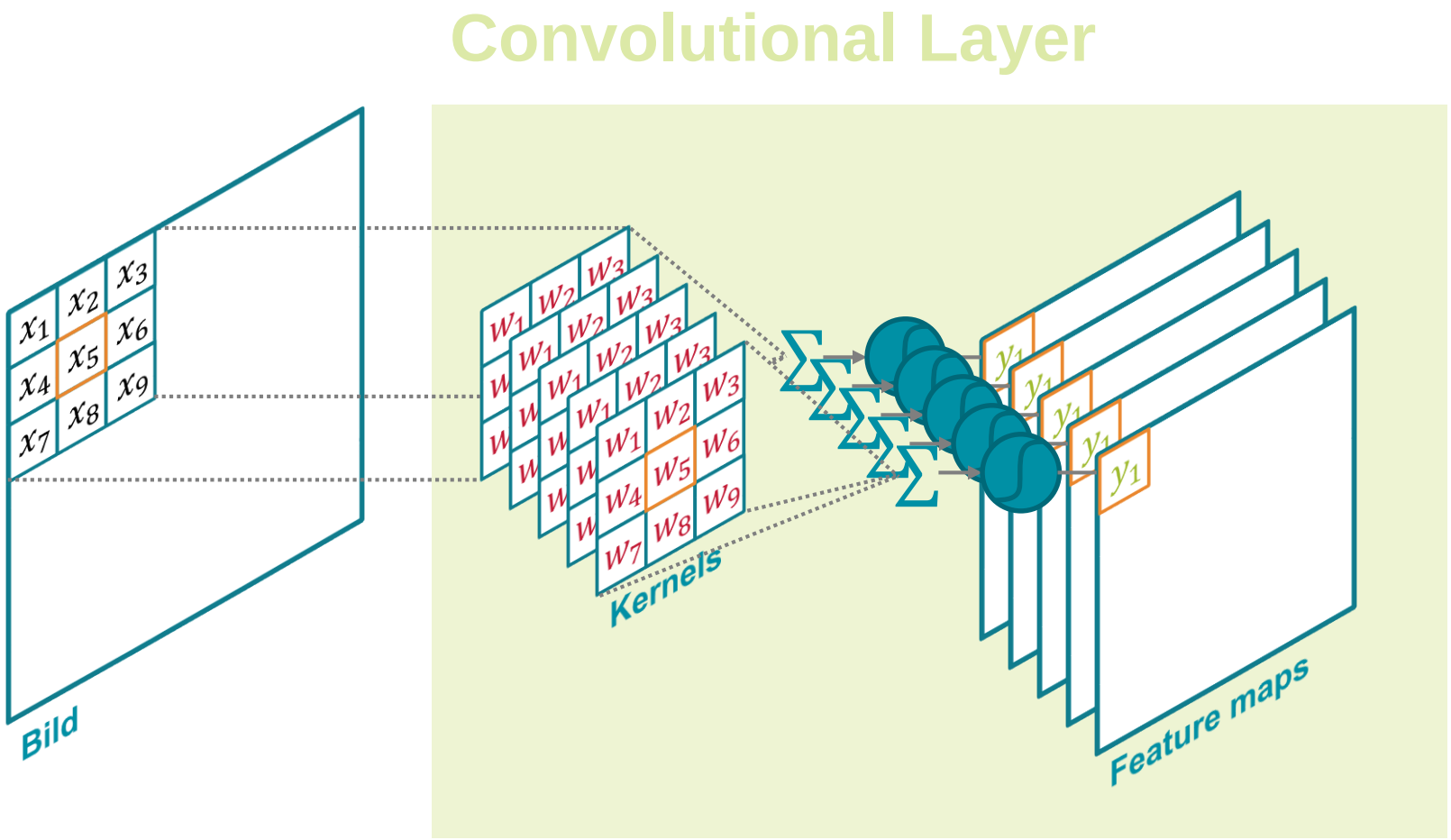
Bild: $m \times n$ Kernel: $k \times k$ $k \ll m, k \ll n$

	Fully Connected	Local Convolution	Shared Weights
#Neuronen	mn	mn	mn
#Gewichte	$(mn)^2$	$k^2(mn)$	k^2

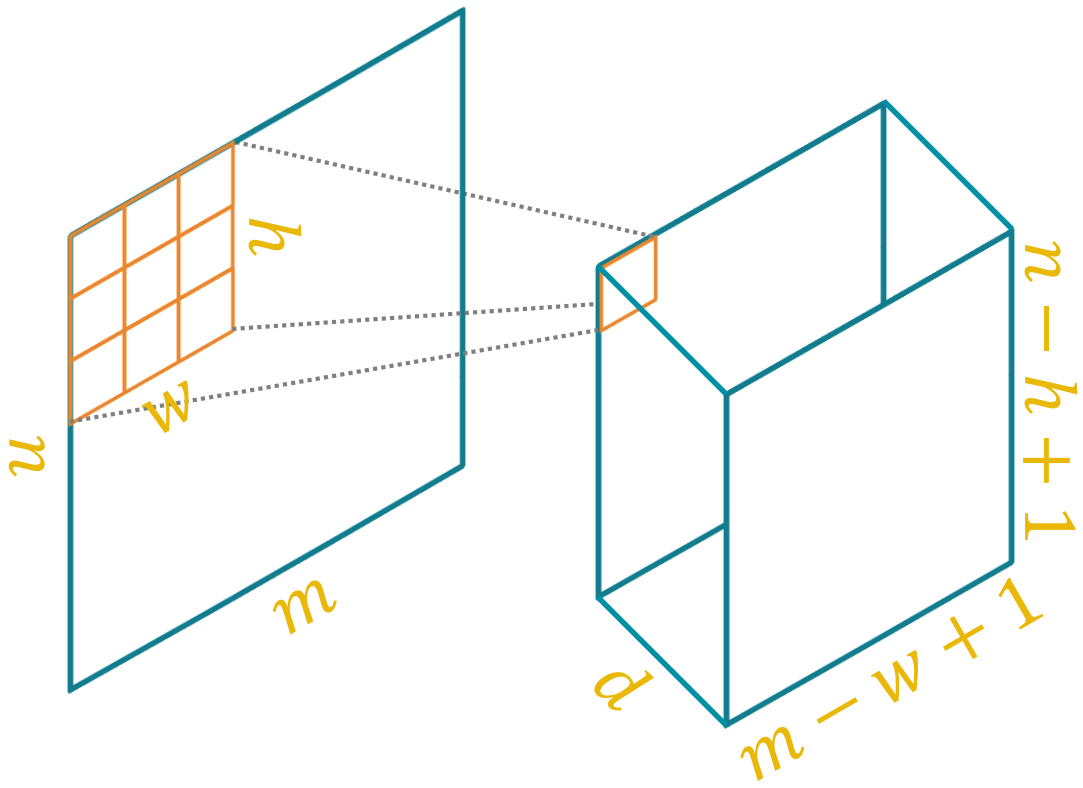
Wie sieht so ein Convolutional Layer aus?



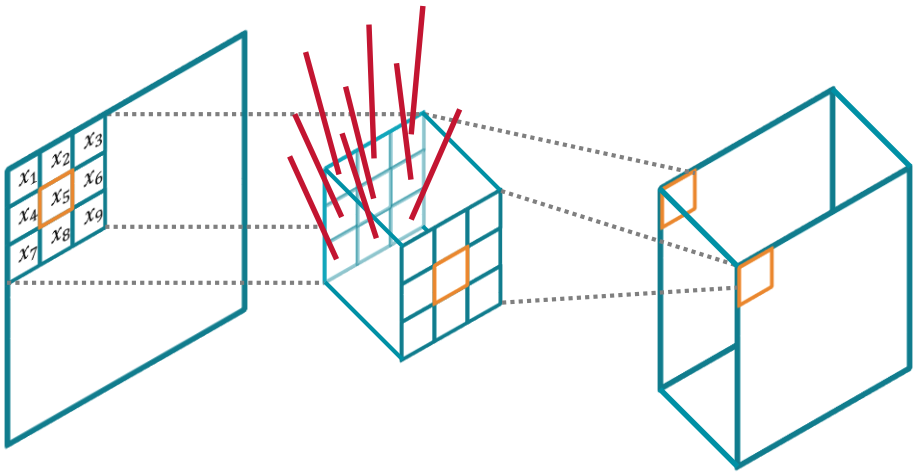
Wie sieht so ein Convolutional Layer aus?

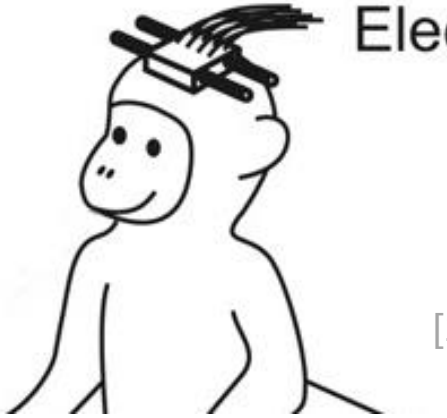


Wie sieht so ein Convolutional Layer aus?

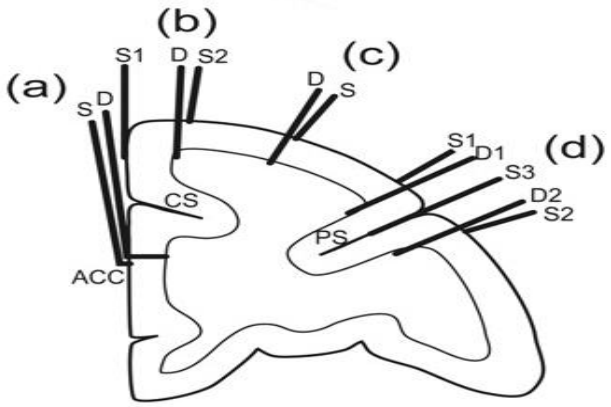


Visualisierung





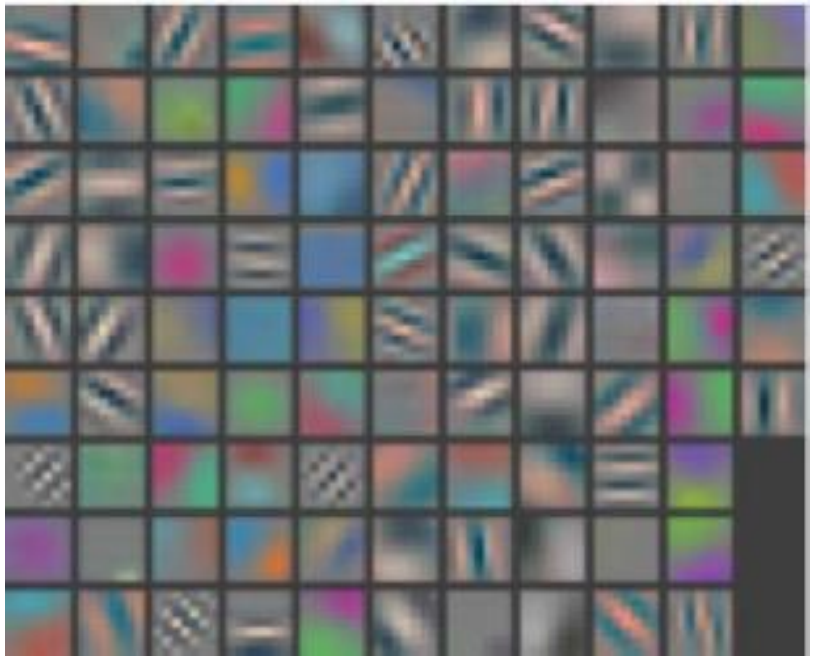
[1]

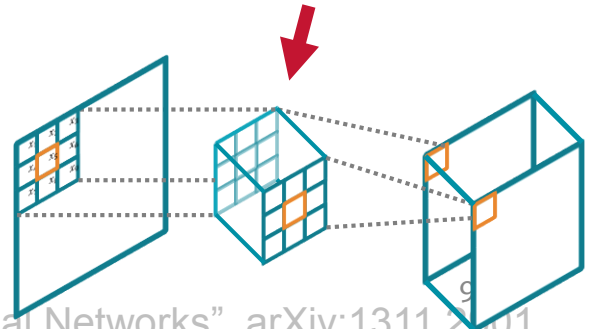


[1]

[1] Tsujimoto, Shimazu and Isomura (2006), “Direct Recording of Theta Oscillations in Primate Prefrontal and Anterior Cingulate Cortices”, Journal of Neurophysiology

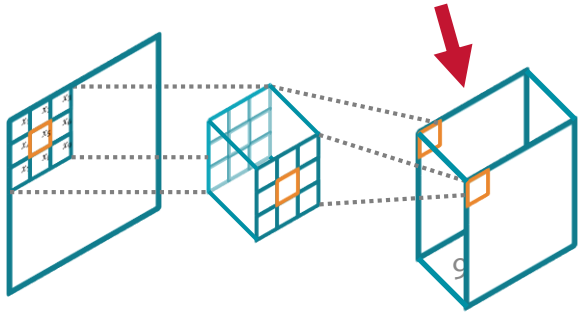
Visualisierung: First Layer Filter





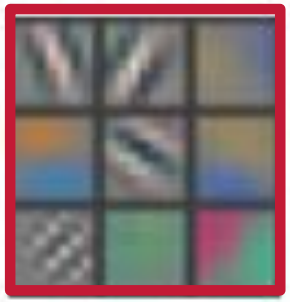
[1] Zeiler und Fergus (2013), “Visualizing and Understanding Convolutional Networks”, arXiv:1311.2901

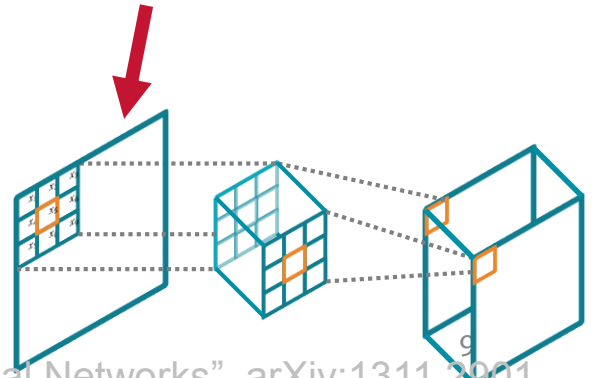
Visualisierung: First Layer Feature Maps



Bildquelle: cs231n.stanford.edu

Visualisierung: First Layer Aktivierungen





[1] Zeiler und Fergus (2013), “Visualizing and Understanding Convolutional Networks”, arXiv:1311.2901

Zusammenfassung

- Ein Convolutional Layer lernt htps. Kanten und Blob-Filter
- Dies entspricht dem Verhalten des Visuellen Cortex
- Diese Filter stellen die kleinsten Einheiten für die verteilte Repräsentation von Objekten in Bildern dar

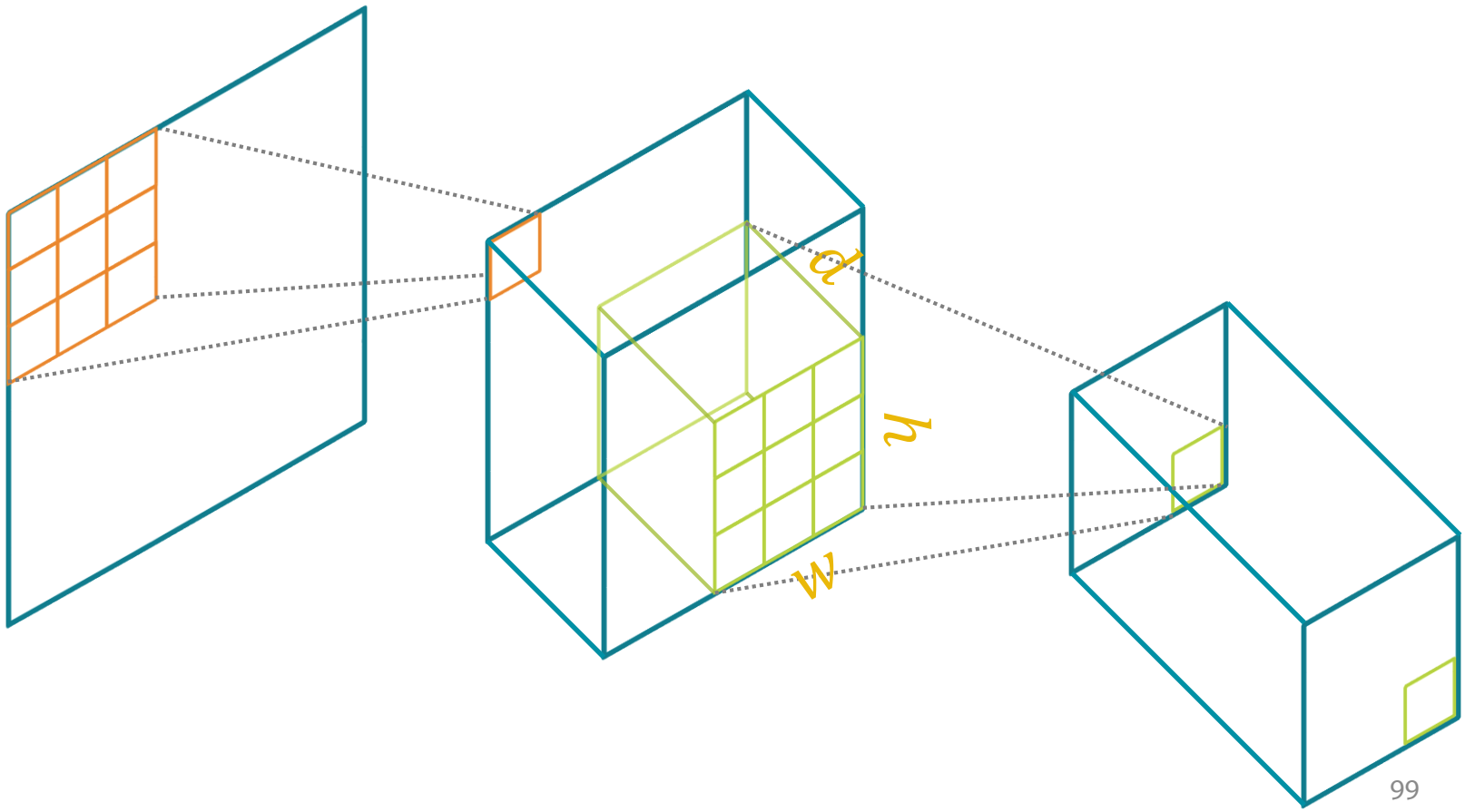
Aber

- Filter sind sehr ähnlich der seither manuell erstellten (nur etwas gezielter und weniger)
- Eine große Verbesserung kann nicht erwartet werden

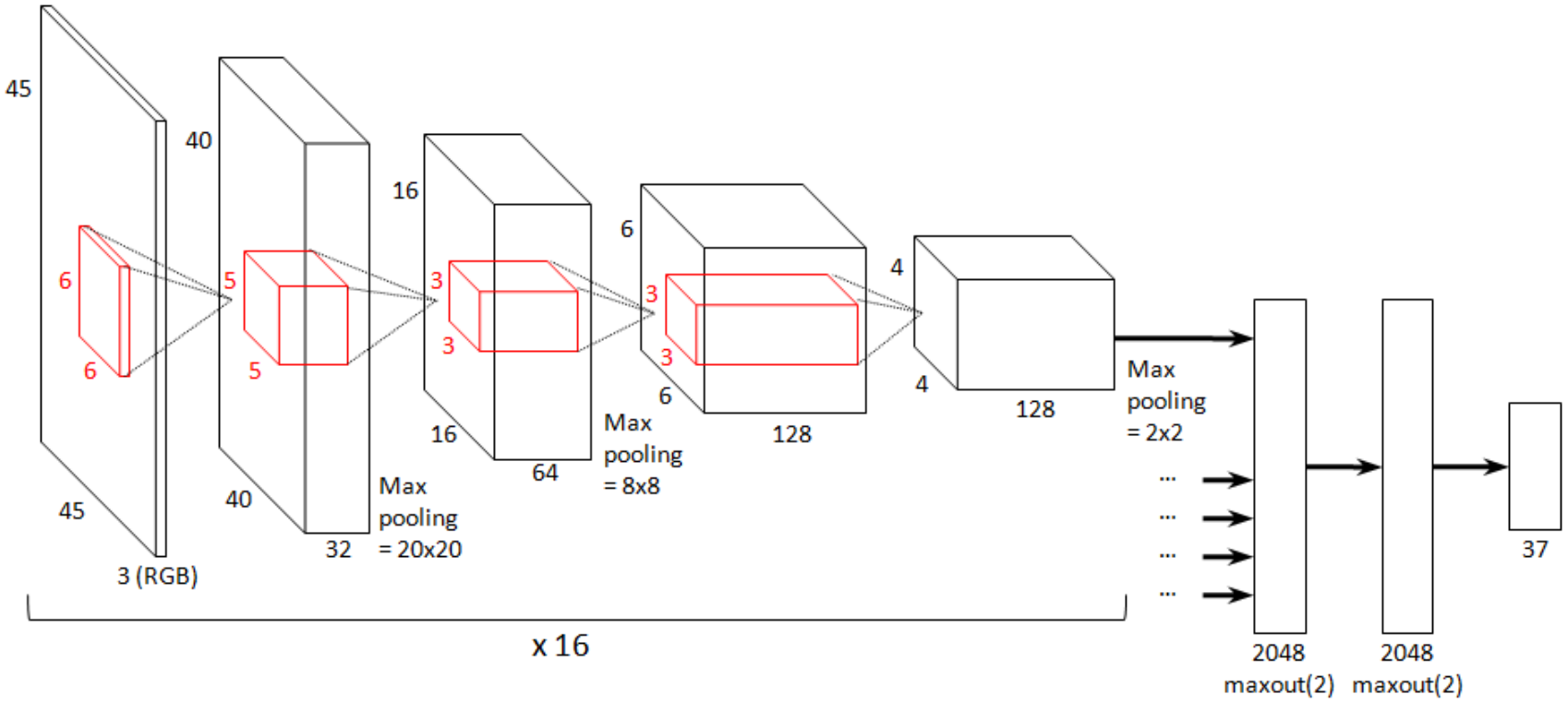
Idee

- Ein Layer kann nur lineare Problem lösen.
- Was passiert, wenn wir einen weiteren Conv Layer hinzufügen?

Wie sieht so ein Convolutional Layer aus?

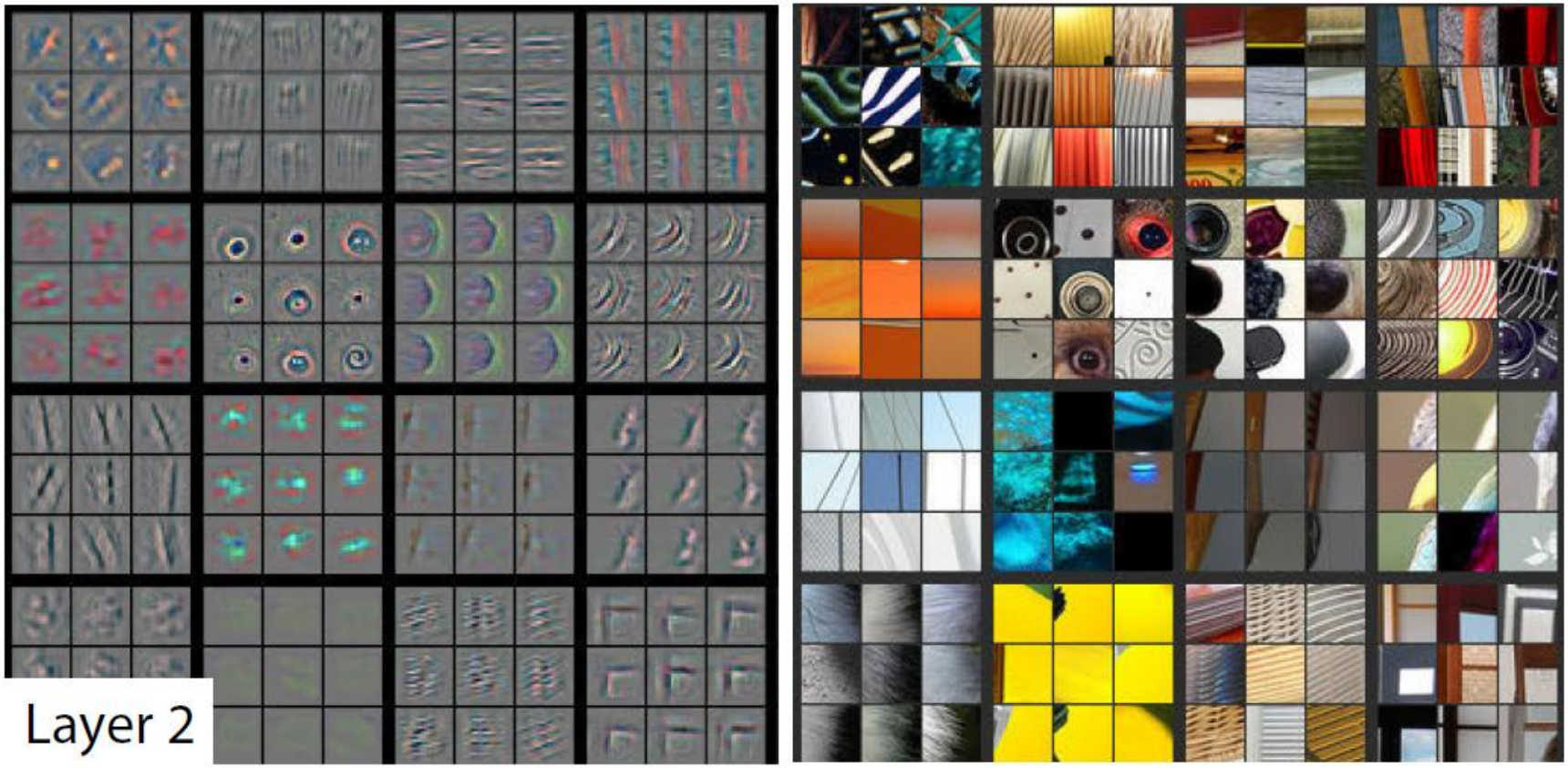


Eine typische CNN Architektur



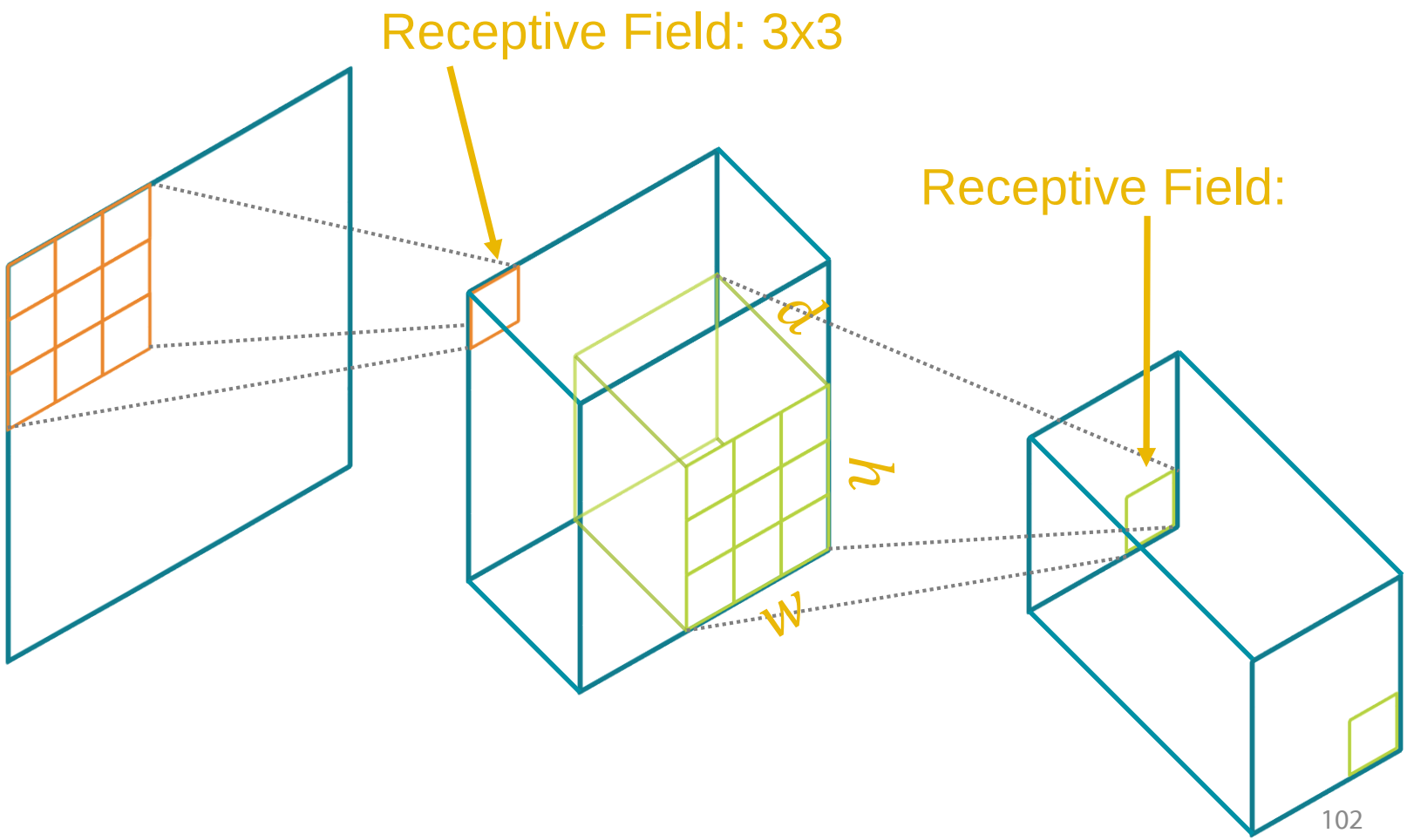
[1] Dielemann, Willett und Dambre, „Rotation-invariant convolutional neural networks for galaxy morphology prediction", arXiv:1503.07077

Visualisierung: Second Layer Filter und Aktivierungen

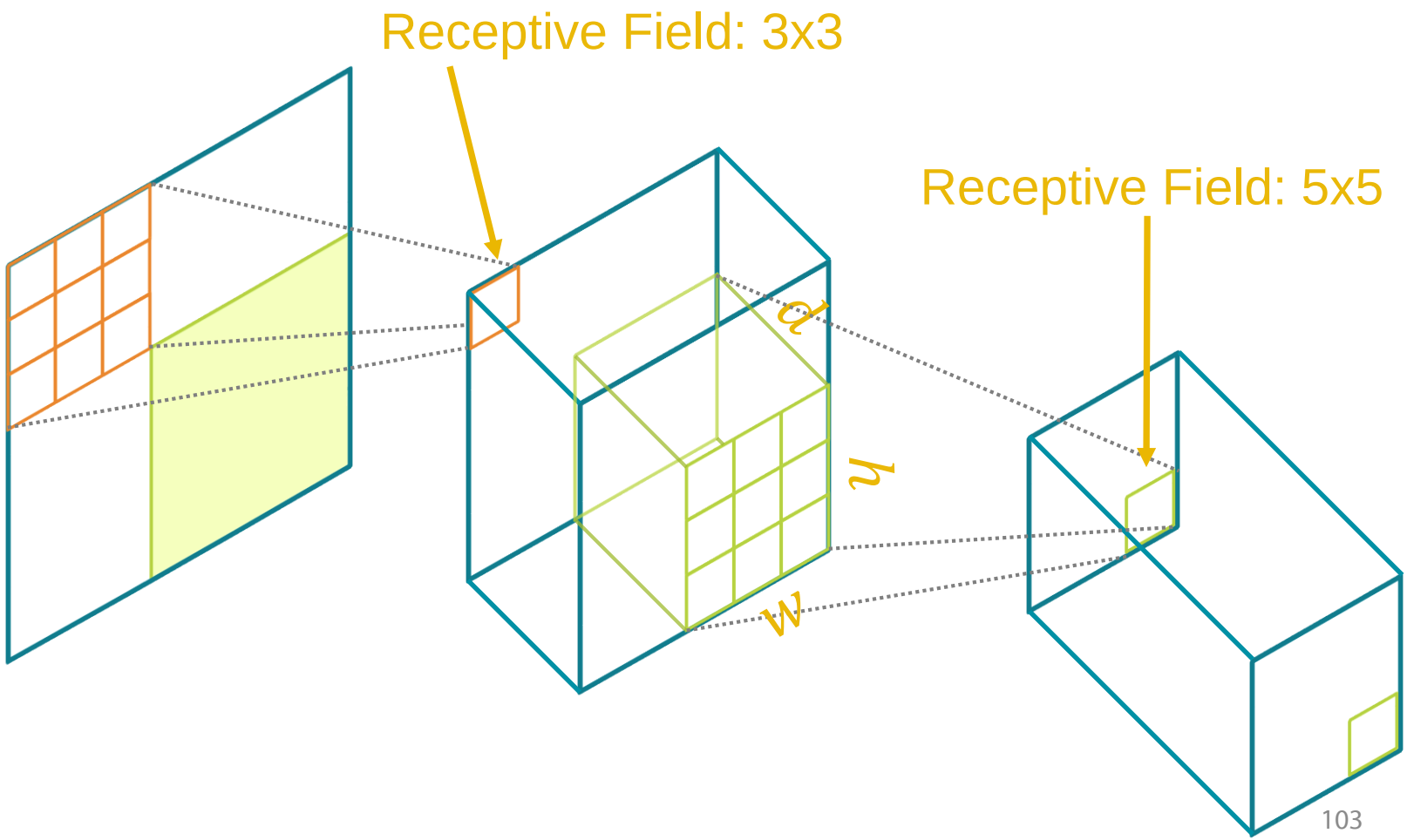


[1] Zeiler und Fergus (2013), “Visualizing and Understanding Convolutional Networks”, arXiv:1311.2901

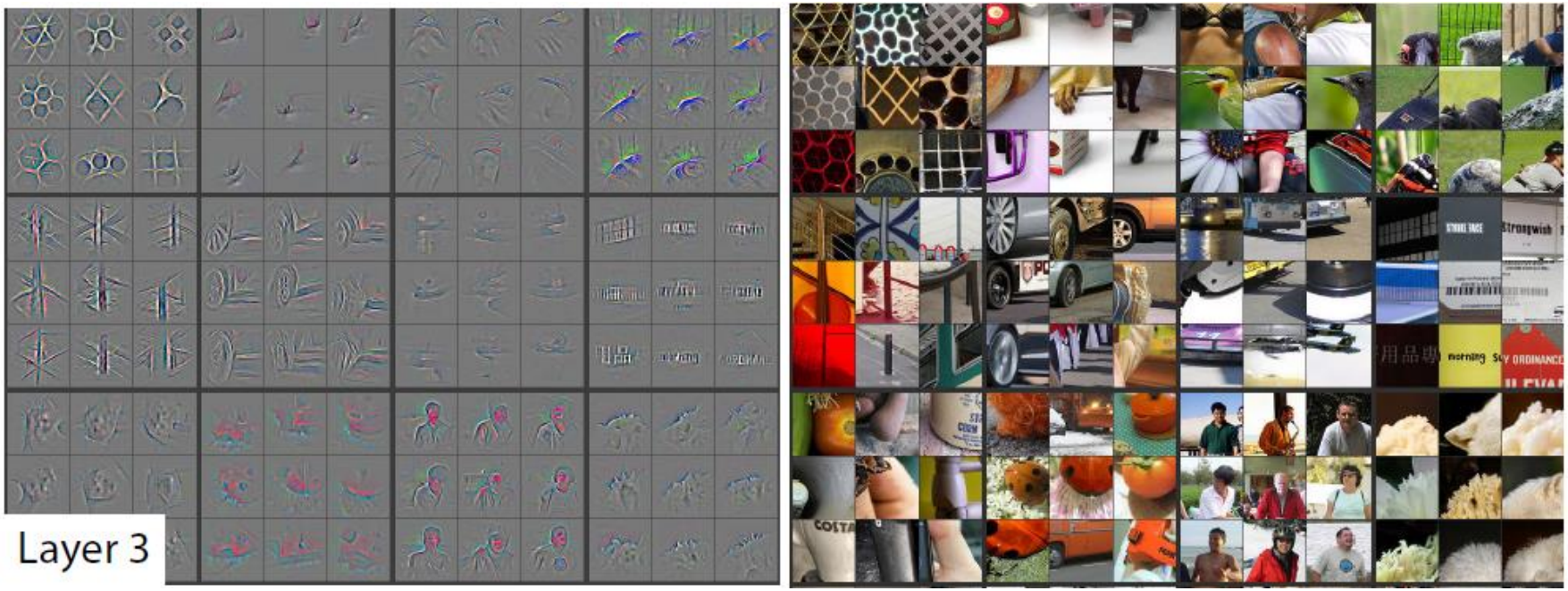
Receptive Field unterer Layer



Receptive Field unterer Layer

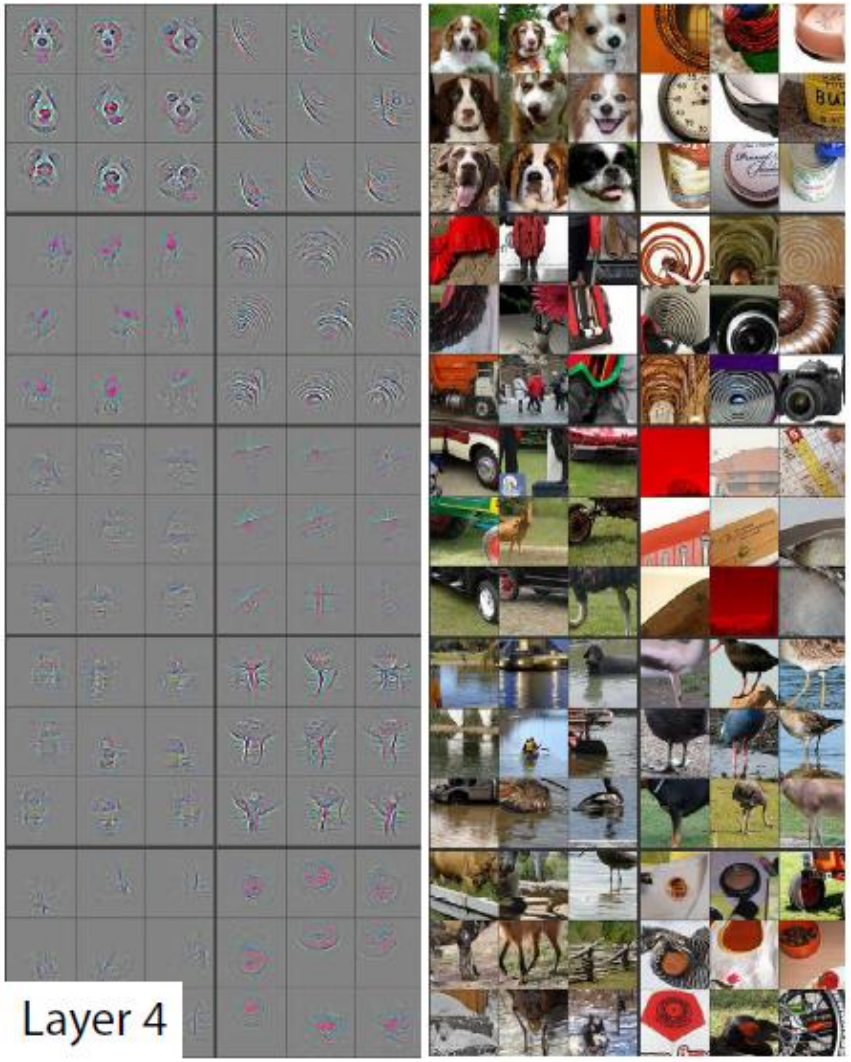


Visualisierung: Third Layer Filter und Aktivierungen



[1] Zeiler und Fergus (2013), “Visualizing and Understanding Convolutional Networks”, arXiv:1311.2901

Visualisierung: Fourth Layer Filter und Aktivierungen



[1] Zeiler und Fergus (2013), “Visualizing and Understanding Convolutional Networks”, arXiv:1311.2901

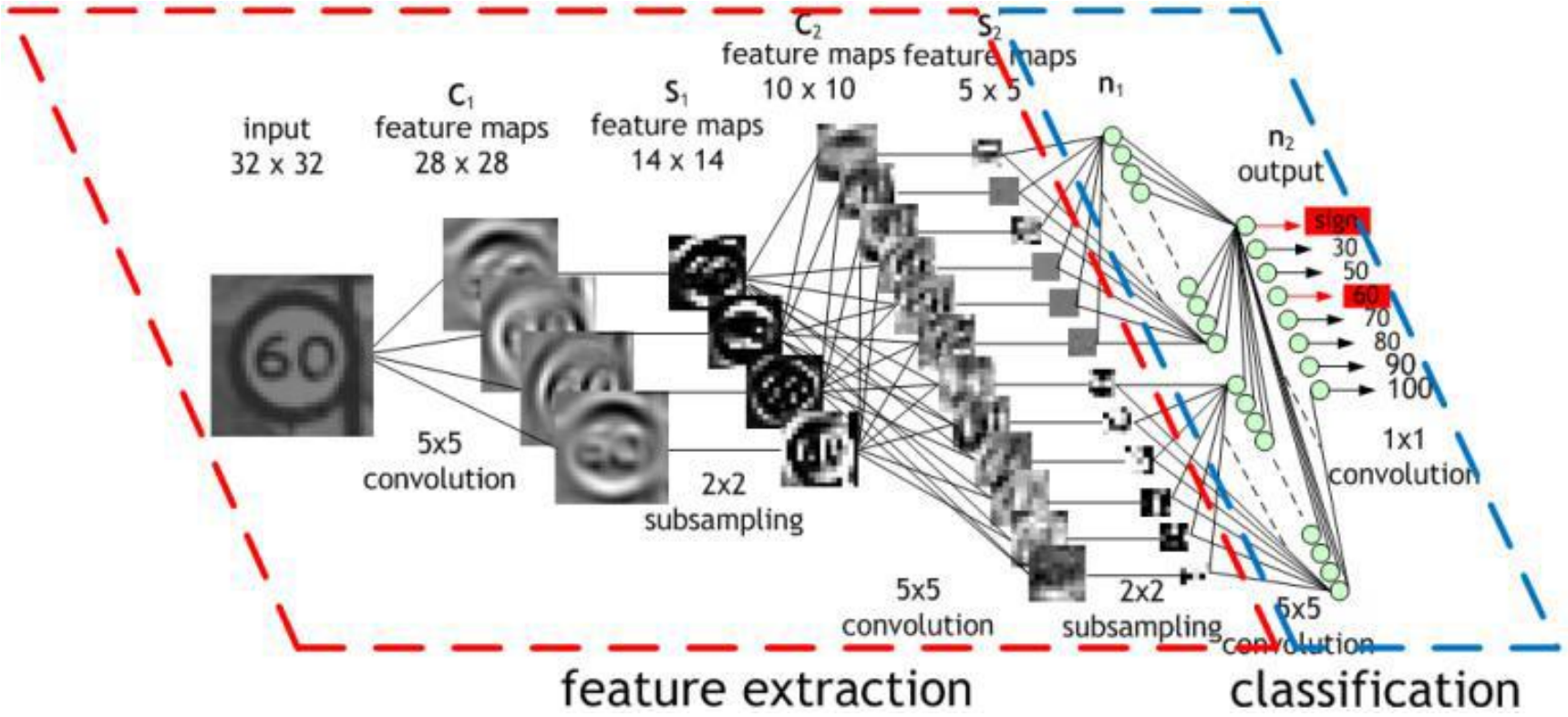
Zusammenfassung

- Auf tieferen Ebenen lernt ein Convolutional Layer weitere Bausteine zu erkennen
- Dies entspricht vermutlich ebenfalls dem Verhalten des Visuellen Cortex
- Je tiefer die Ebene, desto größer das Perceptive Field
- Und desto komplexer das Abgebildete Konzept

Vorteil

- Verschiedenste Bausteine werden erkannt
- Kombiniert über die abschließenden Fully Connected Layer können so sehr viele verschiedene Objekte mit einem verhältnismäßig einfachen Modell erkannt werden

Noch ein Beispiel



Bildquelle: Afxentis, Panagiotis, Alicia Sánchez Crespo, and Ying Zhang. "Neural Network Simulation."

Zusammenfassung CNNs

- Sind ANNs mit weniger Verbindungen und Shared Weights
- Sie folgen der Idee der Verteilten Repräsentation
- Modellieren so die Funktion des Visuellen Cortex
- Lernen die grafischen Bausteine, aus denen sich unsere Konzepte zusammensetzen

- Brauchen keine manuellen Features, sondern lernen die relevanten selbstständig gleichzeitig mit der Klassifikation
- Dadurch kann die Modellkomplexität vielfach reduziert werden

Zusammenfassung CNNs

- Sind Artificial Neural Networks
  - Back Propagation
  - Beliebige Architektur
  - Beliebe Verbindungspattern
  - Können Klassifizieren, Segmentieren, Prädiktieren, …
  - Supervised, Semi-Supervised, Un-Supervised
  - Feed-forward or recurrent

- Sind Deep Neural Networks

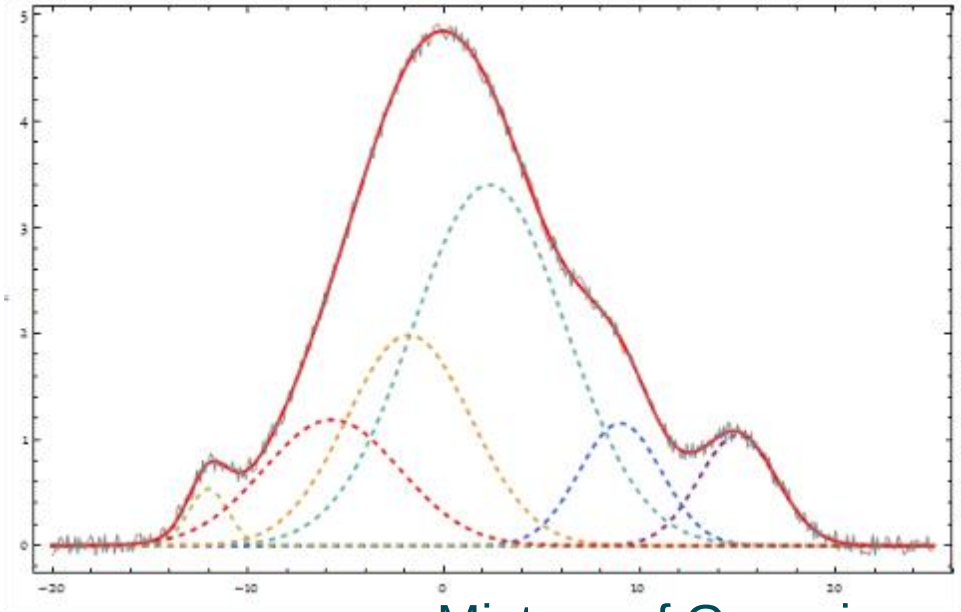
Deep Neural Networks

Theorem

Zwei Layer sind genug jede Funktion zu modelieren

Aber

In Praxis lernen solche Netzwerke nicht gut



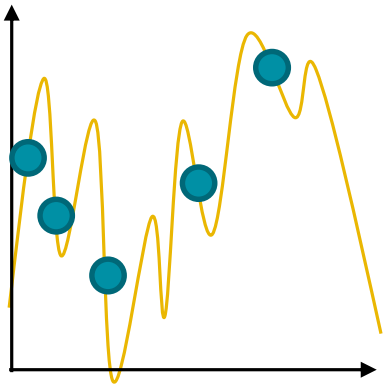
Mixture of Gaussians

Alternative: In die Tiefe gehen mit Deep Neural Networks (DNN)

Mehr Layer erlauben die Modellierung komplexester Funktionen mit (vergleichbar) weniger Parametern

Bildquelle: http://i.stack.imgur.com/tJXqK.png

Probleme tiefer Netwerke



1. Anzahl Parameter
2. Overfitting
3. Gradient vanishing/explosion
4. Input vanishing/explosion
5. Schwingungen in den unteren Layers

$w < 1$
$w > 1$

Viele kleine Veränderungen wurden vorgeschlagen, um diese Probleme zu überkommen.

Inwiefern CNNs diese Probleme lösen

Anzahl Parameter

- Mit k^2 statt $(mn)^2$ viel weniger als Fully Connected Netzwerke

Overfitting

- Local convolution
- Shared weights
- Max-pooling
- …alles Regularisierung

Weitere begünstigende Entwicklungen

Geschwindigkeit und Speicheranforderung

- GPU Implementierungen und bessere Grafikkarten

Gradient Vanishing

- GPU erlaubt viele Epochen und damit auch Training mit kleinen Gradienten in den oberen Layers

Overfitting

- Big Data

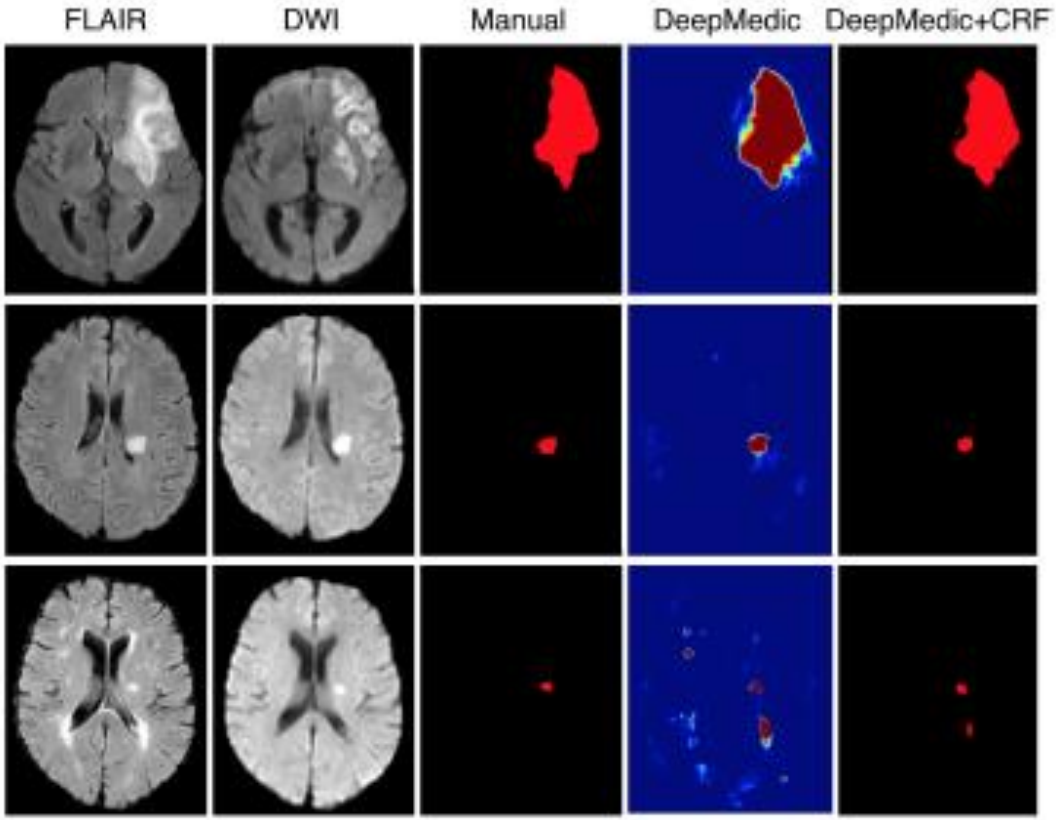
Weitere Entwicklungen

- Max-pooling
- Rectified Linear Units
- Weight Initialization
- Weight Decay
- Controlled Gradient Descent
- Drop Out (Simon Müller, Ilja Stechmann)
- Mini Batch
- Ensembles
- Weight Normalization (L1 or L2 norm)
- …and many more: Raum 5, 2. Stock, Geb. 64 in ca. 45 Minuten!

Anwendungsbeispiel

Kamnitsas et al. „Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation"

Medical Image Analysis 2016 (preprint: arXiv:1603.05959)

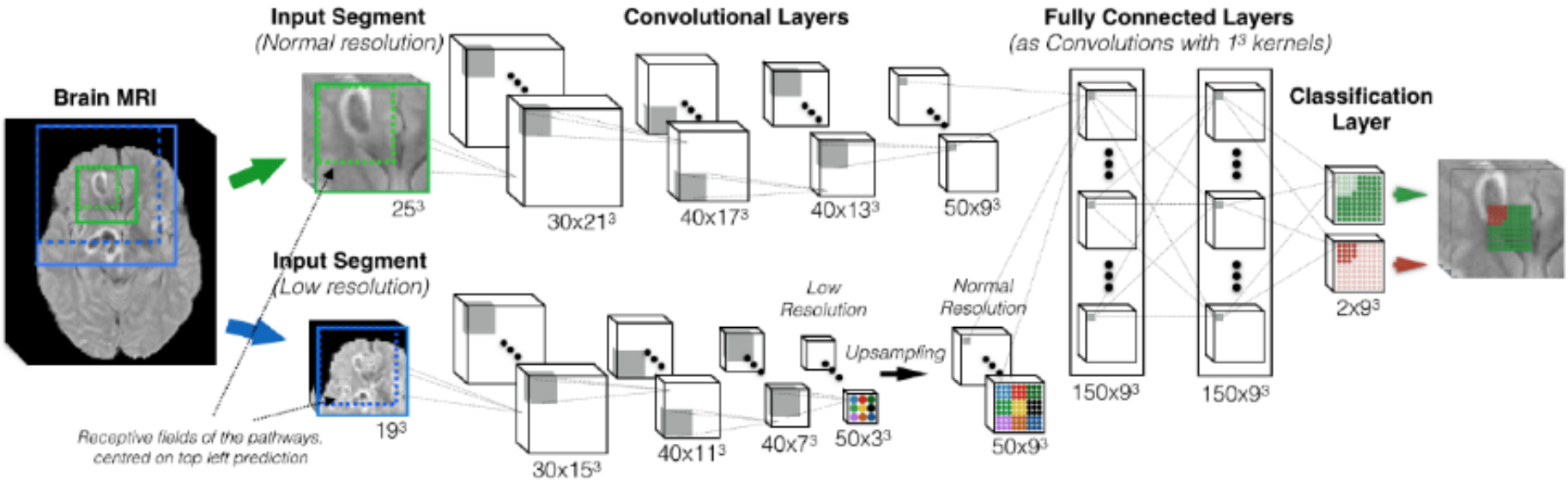


Beispielsanwendung auf ISLES 2015 Schlaganfalldaten

Anwendungsbeispiel

Kamnitsas et al. „Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation"

Medical Image Analysis 2016 (preprint: arXiv:1603.05959)



Das vorgeschlagenen Netzwerk mit Beispielsbild

Deep Learning: Die Lösung für alle Probleme

- Segmentierung, Übersetzung, Prädiktion, Träumen…
- Und das alles selbst-lernend mit out-of-the-box Tools!

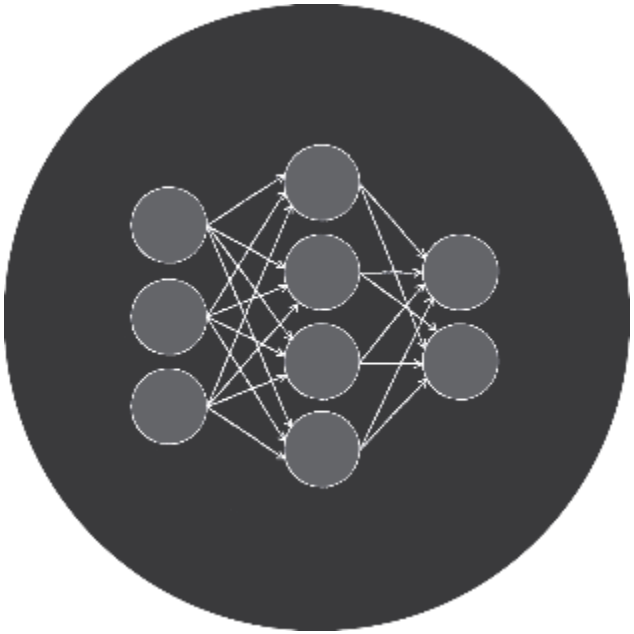
Wahr

1. Große Erfolge!
2. Potential!
3. Jahrelanger ANN Stillstand überwunden!

Aber

1. Enorme Flexibilität hat ihren Preis
2. Die guten Lösungen sind hochgradig Problemabhängig

Das elektronische Gehirn: Sind wird angekommen?



Single task
736 Watt[1]
1.6×10^{12} Weights[1]
152 layers[2]

Multi-task
20 Watt
1.5×10^{14} Weights
∞ Layers

[1]GeForce GTX 580 3-way SLI

[1] Trask, Gilmore and Russel (2015), „Modeling Order in Neural Word Embeddings at Scale”, arXiv:1506.02
[2] He, Zhang, Ren und Sun (2015), „Deep Residual Learning for Image Recognition”, arXiv:1512.03385

Zukunft

- Die Aufregung mag enden, aber wohl nicht so schnell
- Viele praktische Anwendungen und große Investitionen der Global Players (Google, Facebook, Amazon)
- Da kommt noch einiges auf uns zu
- Rechengeschwindigkeit wird zunehmen und weitere Anwendungsgebiete eröffnen

Weiterführende Literatur

Vorlesungen von Nando de Freitag @ Oxford

- http://www.cs.ox.ac.uk/teaching/courses/2014-2015/ml/
- https://www.youtube.com/playlist?list=PLE6Wd9FR--EfW8dtjAuPoTuPcqmOV53Fu

Vorlesungen zu Kurs CS231 @ Stanford

- http://cs231n.github.io
- https://www.youtube.com/playlist?list=PLrZmhn8sSgye6ijhLzlIXiU9GNaIwbF8B

Michael Nielsen: Neural Networks and Deep Learning

- http://neuralnetworksanddeeplearning.com

Andrew Ng: UFLDL Tutorial @ Stanford

- http://ufldl.stanford.edu/tutorial/

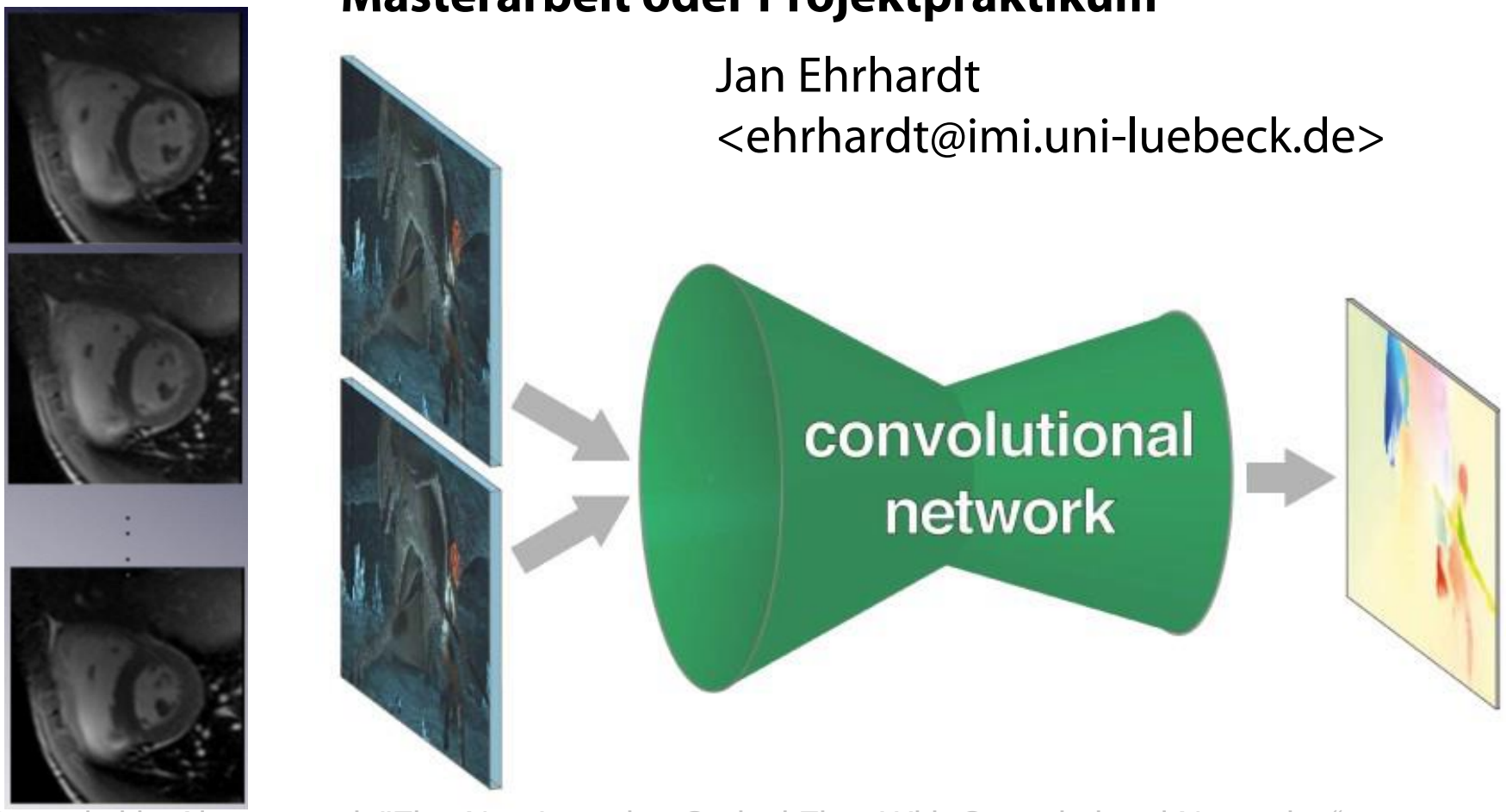
Machtvolle Deep Learning Frameworks

- http://caffe.berkeleyvision.org
- http://torch.ch

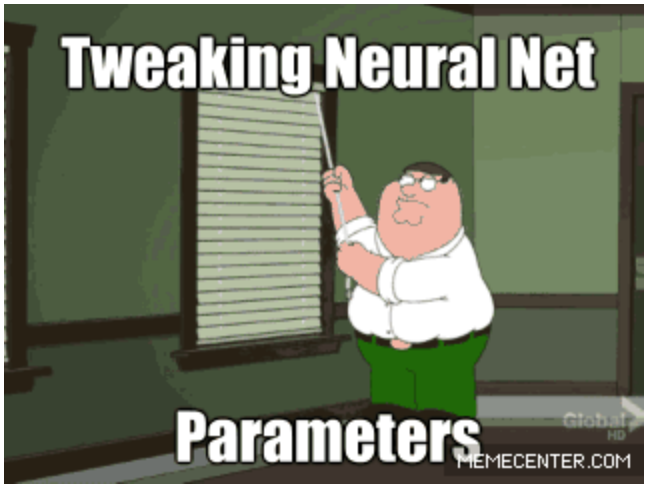
Bewegungsschätzung in medizinischen Bilddaten mit Deep Learning

Masterarbeit oder Projektpraktikum

Jan Ehrhardt
<ehrhardt@imi.uni-luebeck.de>



[1] Dosovitskiy, Alexey, et al. "FlowNet: Learning Optical Flow With Convolutional Networks.“ Proceedings of the IEEE International Conference on Computer Vision. 2015



Q&A
Ende

Bildquelle:http://bbabenko.tumblr.com/post/83319141207/convolutional-learnings-things-i-learned-by