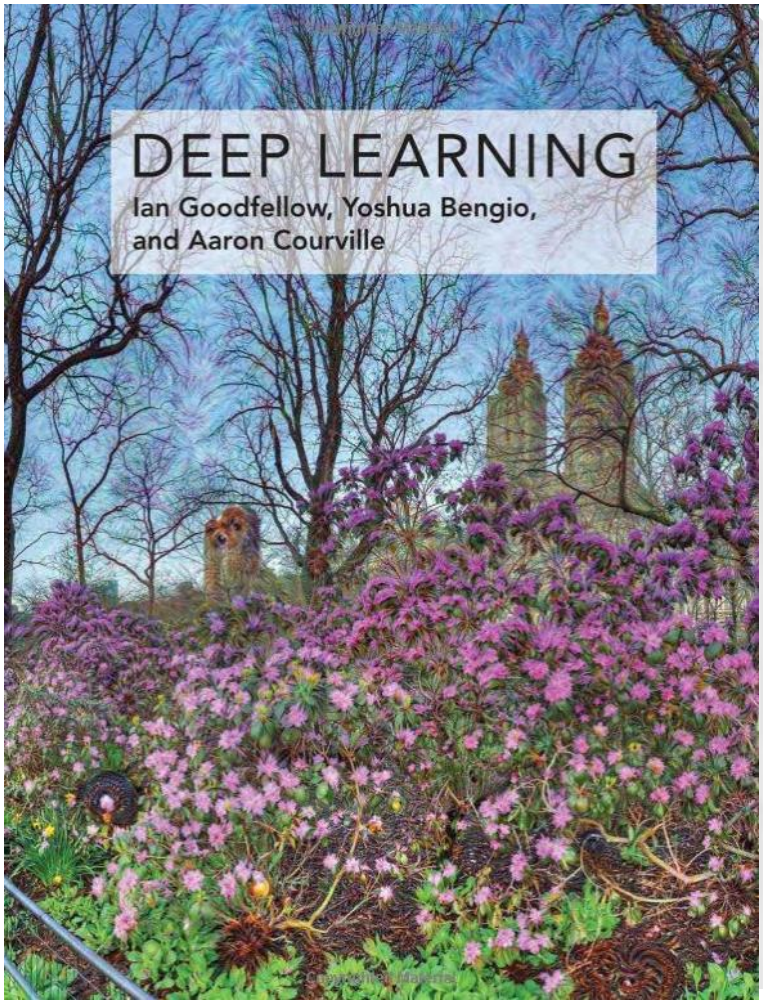
Ludwig Maximilians Universität München
Institut für Informatik
Forschungsgruppe Data Mining in der Medizin

Neuronale Netze

Christian Böhm

http://dmm.dbs.ifi.lmu.de/dbs

Lehrbuch zur Vorlesung



Lehrbuch zur Vorlesung:

Goodfellow, Bengio, Courville:

Deep Learning

Adaptive Computation and Machine Learning

MIT Press, 2016, 46,99 € (780 Seiten, gebunden)

Motivation

Bei vielen komplexen Aufgaben ist das menschliche Gehirn klassischen Algorithmen (zunächst) überlegen:

- Erkennung gesprochener Sprache oder Handschrift,
- Bild- und Gesichtserkennung,
- Spiele mit komplexen Entscheidungen,
- Fahrzeugsteuerung.

Neuronale Netze: Algorithmen bilden das natürliche Lernen nach.

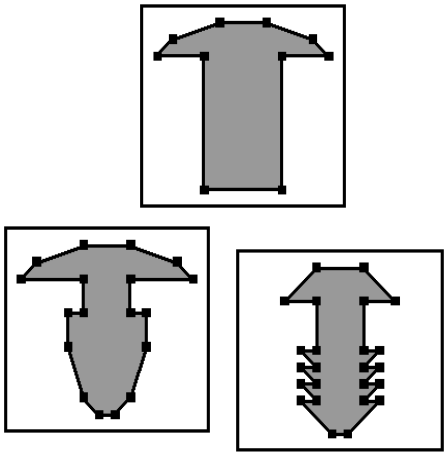
Historie:

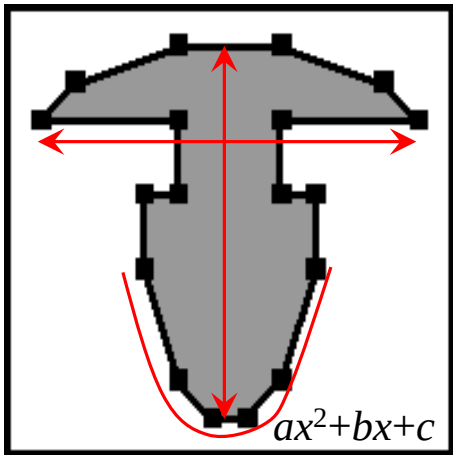
- Erste Ansätze schon 1943 (McCullon/Pitch-Neuron),
- 1970-1990 Verwendung einfacher neuronaler Netze,
- Seit ca. 2009: „*Deep Learning*“:
  - Wesentlich komplexere Netzwerke,
  - Hohe Rechenleistung zum Training erforderlich,
  - Deutlich verbesserte Erkennungsleistung.

Feature-Vektoren

- Viele Anwendungen arbeiten mit komplexen Objekten.
- Es ist die Aufgabe des Anwendungs-Experten, geeignete Merkmale („*Features*“) der Objekte zu definieren.

Beispiel: CAD-Zeichnungen kleiner Bauteile:

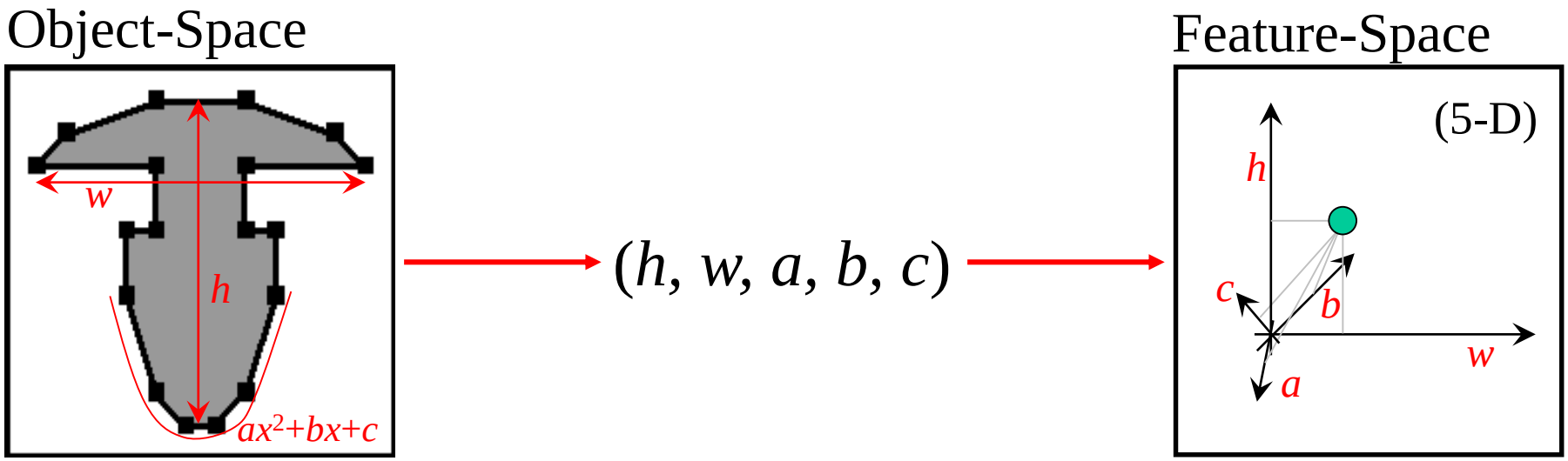




Mögliche Merkmale:

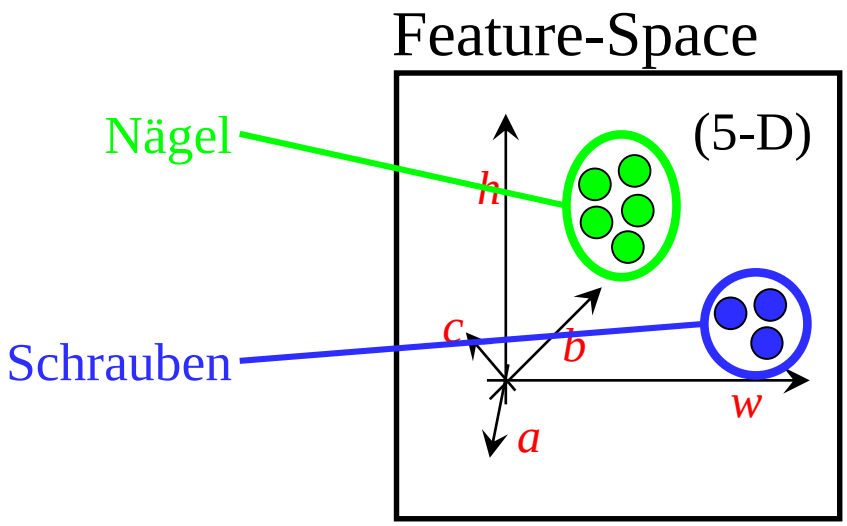
- Höhe h
- Breite w
- Form-Parameter (a,b,c)

Feature-Vektoren



- Die ausgewählten Features bilden einen *Feature-Vector*
- Der Feature-Space ist oft hochdimensional (im Beispiel 5-D)

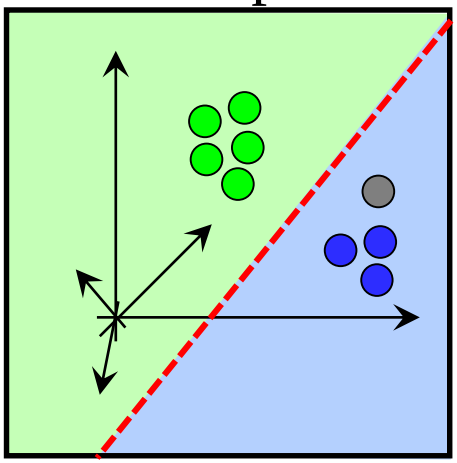
Feature-Vektoren



- Ähnliche Objekte haben ähnliche Eigenschaften.
- Die Distanz der Feature-Vektoren ist deshalb ein Maß für die Ähnlichkeit.
- Oft gehören ähnliche Objekte der gleichen Klasse an.

Feature-Vektoren

Feature-Space



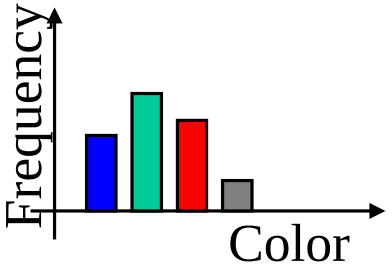
Definition *Klassifikation*:

Lerne von vorklassifizierten *Trainings-Daten* (●, ●)
die *Regeln*, um die Klasse neuer Objekte (●)
nur auf Basis der Features vorhersagen zu können.

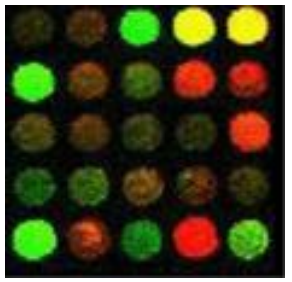
Beispiel: Support-Vector-Machine (lineare Trenn-Ebene ⁄)

Feature-Vektoren

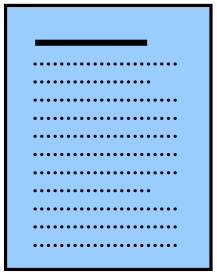
Bilddatenbanken:
Farb-Histogramme.



Gen-Datenbanken:
Expressionslevel.



Dokument-Datenbanken:
Term-Häufigkeiten.

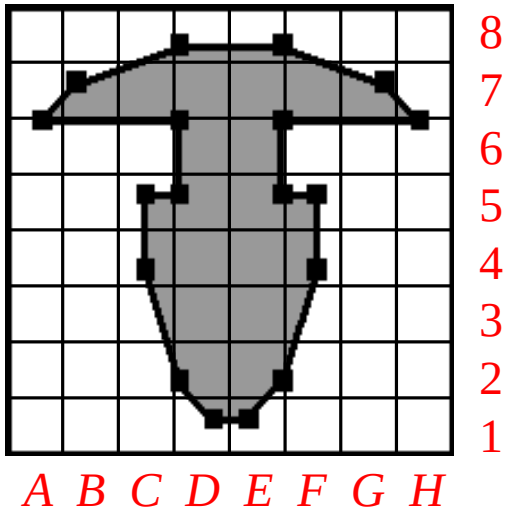


Data	25
Mining	15
Feature	12
Object	7
...	

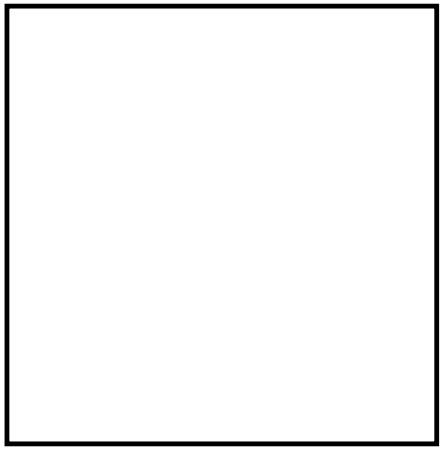
Der Feature-basierte Ansatz ermöglicht es, eine Vielzahl von Anwendungen einheitlich zu behandeln.

Feature-Vektoren

Objekt-Space

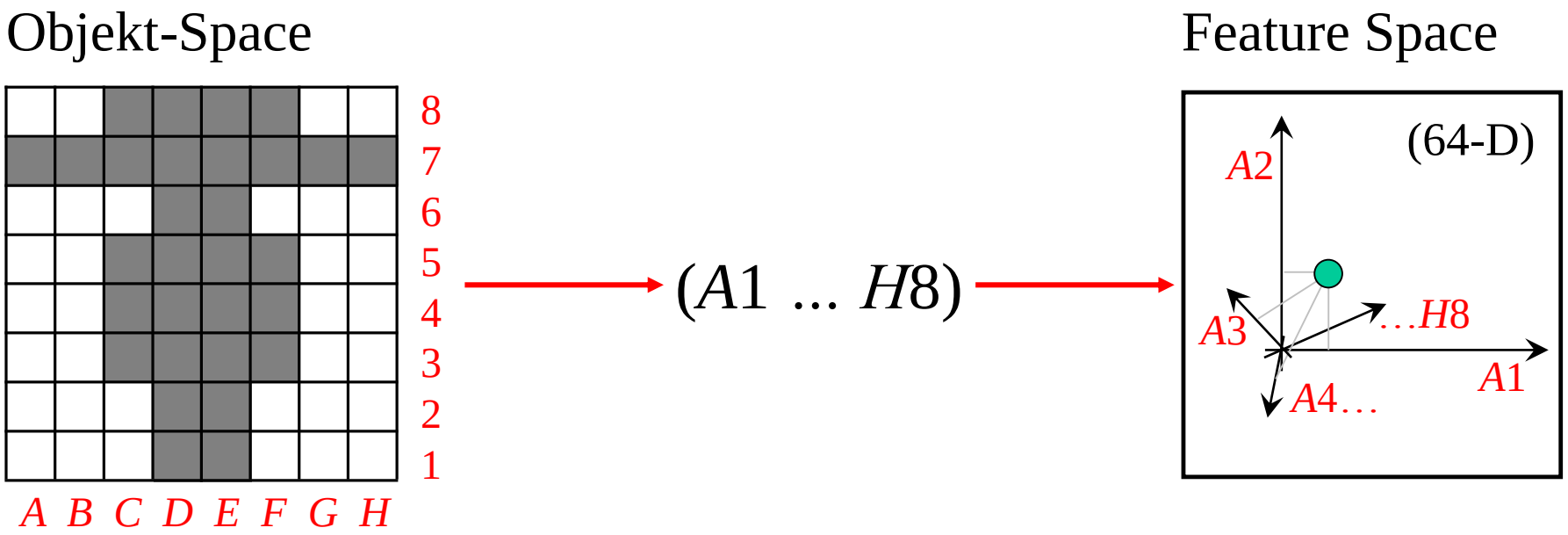


Feature Space



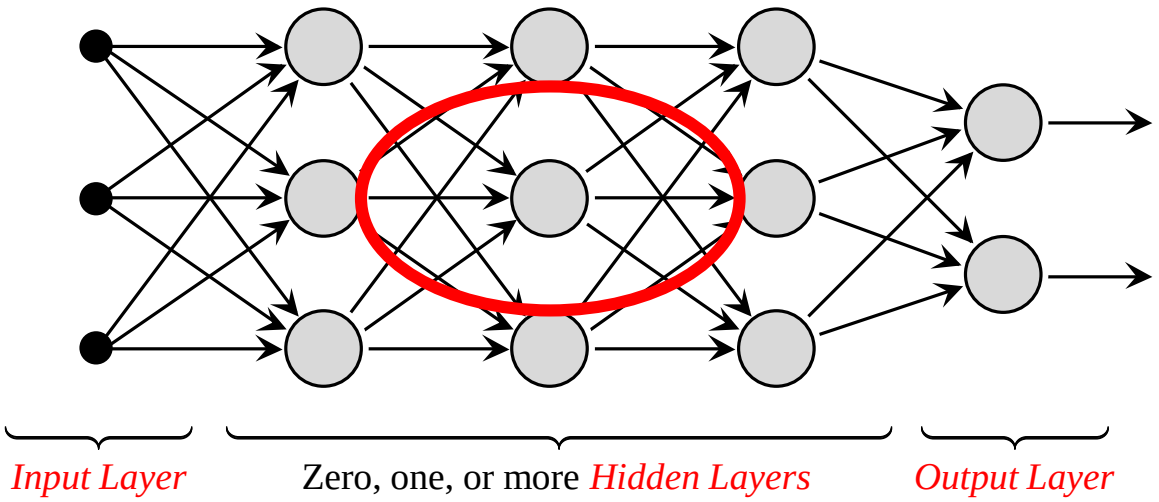
- Bei Anwendungen von tiefen neuronalen Netzen verwendet man häufig sehr einfache Features.
- Im Beispiel wird jedes Pixel (z.B. nach Reduktion oder Standardisierung der Auflösung) als Feature (64-D) verwendet.
- Komplexe Zusammenhänge soll das Lernverfahren selbst erkennen.

Feature-Vektoren



- Bei Anwendungen von tiefen neuronalen Netzen verwendet man häufig sehr einfache Features.
- Im Beispiel wird jedes Pixel (z.B. nach Reduktion oder Standardisierung der Auflösung) als Feature (64-D) verwendet.
- Komplexe Zusammenhänge soll das Lernverfahren selbst erkennen.

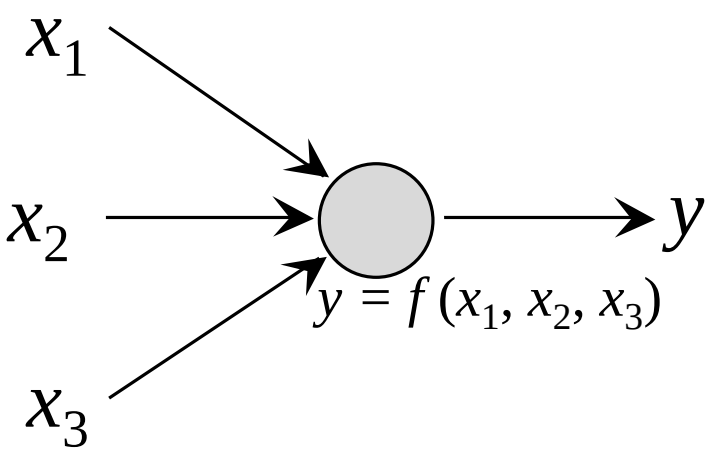
Artificial Neural Network (ANN)



Ein Artificial Neural Network (ANN, künstliches neuronales Netz) ist ein

- Netwerk (*gerichteter Graph*)
- von informationsverarbeitenden Einheiten (*künstliche Neuronen*),
- das meist für Aufgaben des maschinellen Lernens wie z.B. zur Klassifikation eingesetzt wird.

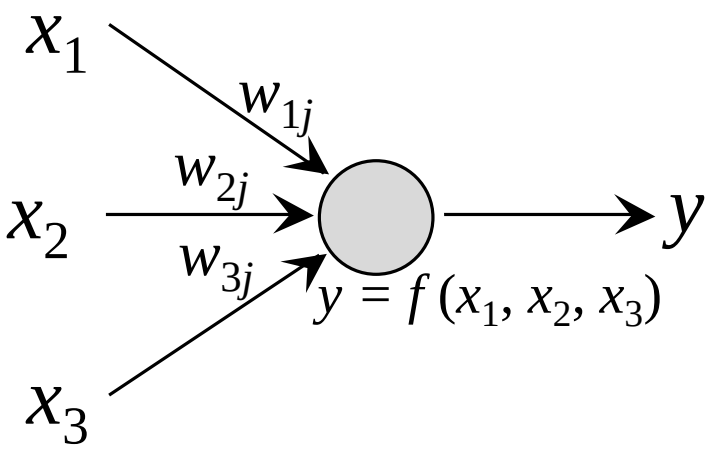
Das Künstliche Neuron



Das künstliche Neuron…

- Realisiert eine mathematische *Funktion* f ( ).
- Eingaben (x_1, x_2, x_3) werden über eingehende Kanten übergeben.
- Das Ergebnis (y) wird über ausgehende Kanten an andere künstliche Neuronen übermittelt.

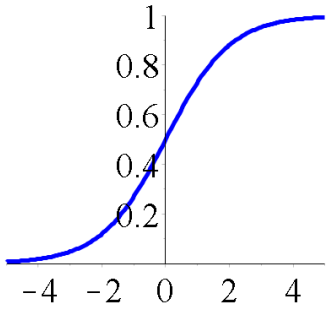
Das Künstliche Neuron



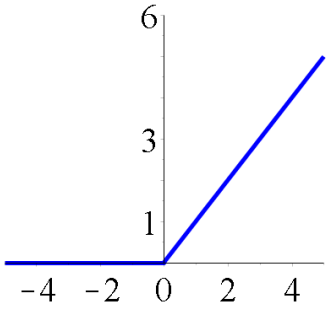
Die Funktion f ( ) ist oft zusammengesetzt aus…

- einer linearen *Übertragungsfunktion* der Eingaben x_i
  (z.B. einer gewichteten Summe $\Sigma_i\, w_{ij}\, x_i$)
- und danach einer nichtlinearen *Aktivierungsfunktion*, z.B.

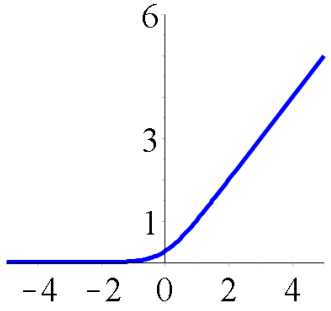
Sigmoid



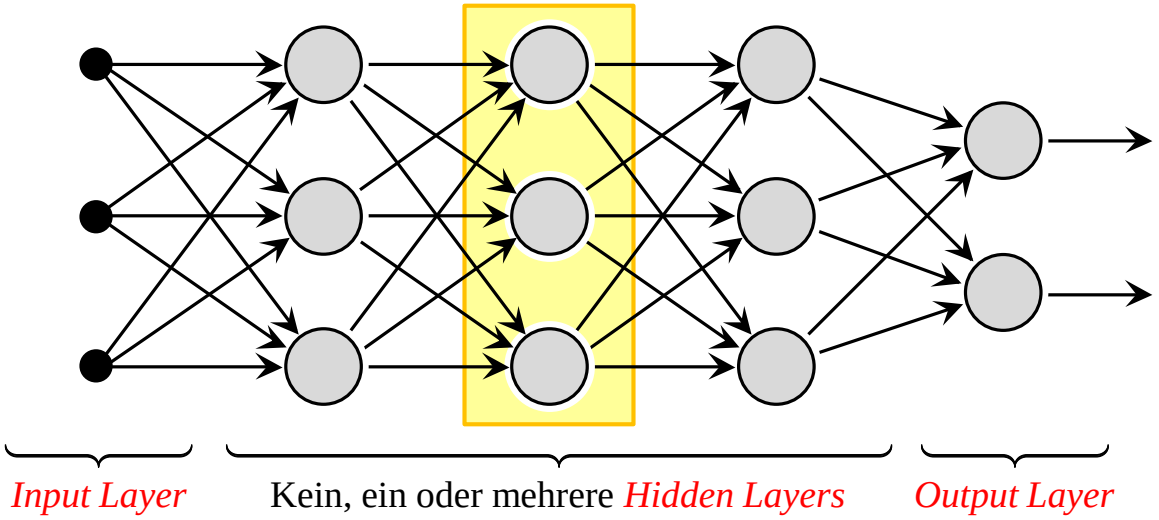
ReLU



Softplus

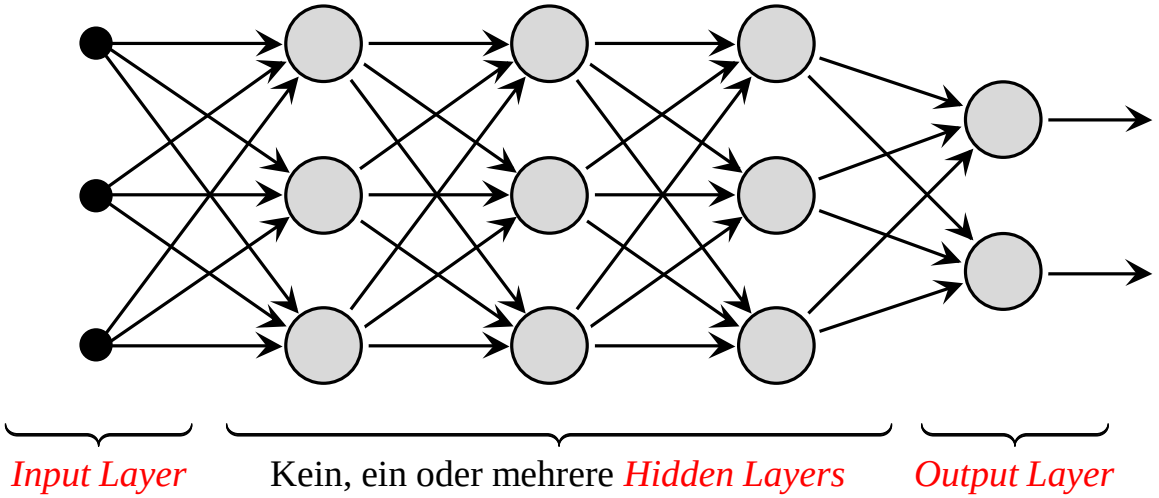


Layers eines ANN

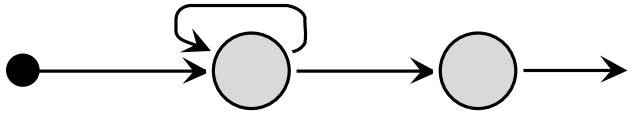


- In natürlichen neuronalen Netzen sind die Neuronen miteinander in beliebiger Ordnung verbunden.
- ANNs ordnen die Neuronen Lagen-weise (*Layer*) an:
  - genau ein Output-Layer
  - beliebig viele Hidden Layers
- Die Neuronen einer Ebene verwenden alle die gleiche Funktion, aber unterschiedliche Parameter (z.B. die Gewichte w_{ij})

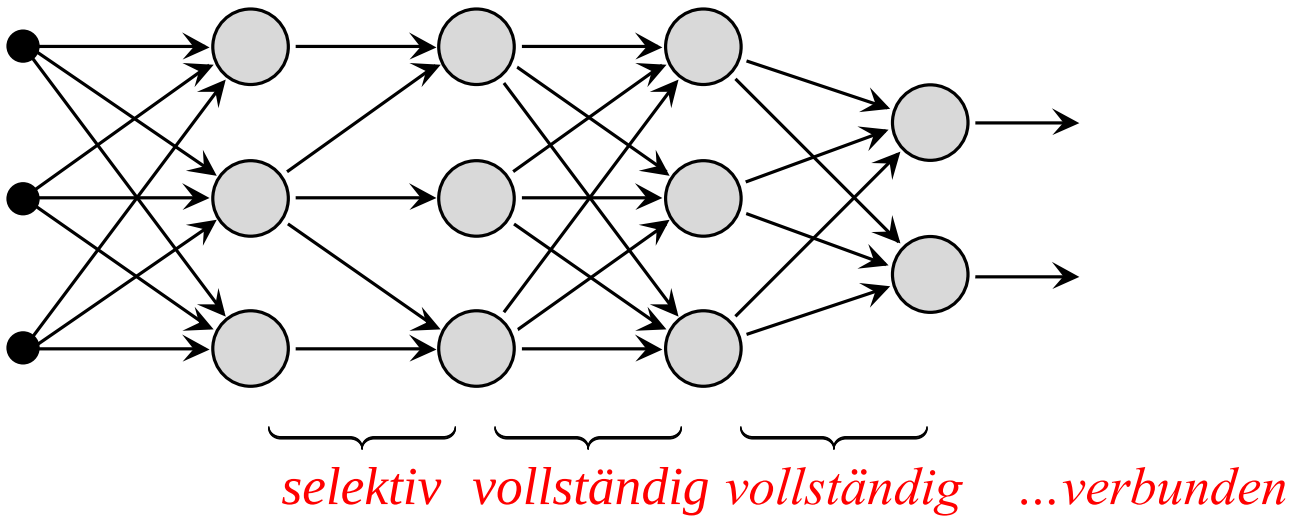
Feedforward vs. Rekurrente ANNs



- ANNs, die nur Vorwärts-Verbindungen haben (zwischen dem aktuellen Layer und dem jeweiligen Nachfolge-Layer, bei möglichem Überspringen von Layers) nennt man *Feedforward-NN.*
- ANNs mit rückwärtsgerichteten Kanten auf den gleichen oder vorige Layers werden *Rekurrente NNs* genannt.
  Sie ermöglichen eine Art Gedächtnis (häufig mit Zeitverzögerung)



Vollständig verbundene Layers



Layers können vollständig oder selektiv verbunden sein:

- Bei vollständiger Verbindung ist jedes Neuron eines Layers mit jedem Neuron des Nachfolgelayers verbunden.
- Das erlernte Gewicht kann aber 0 sein (wie „keine Verbindung“).
- Welche Verbindungsart man wählt, hängt von der Aufgabe ab, die der Layer zu erledigen hat:
  z.B. Konvolutionale NNs aus der Bildverarbeitung:
  - Mustererkennung auf lokaler Ebene (benachbarte Pixel)

Netz-Topologie

Alle Fragen der Netz-Topologie, wie z.B.

- Wie viele Hidden Layers?
- Wie viele Neuronen auf jedem Layer?
- Feedforward oder Rekurrent? Vollständig verbunden?
- Welche Übertragungs-/Aktivierungsfunktionen?

…usw. usw.

werden nicht automatisch gelernt, sondern vom Designer einer ANN-Anwendung festgelegt (Best Practices, Trial-and-Error).

➔ High-Level Interfaces wie z.B. Keras unterstützen das Design.

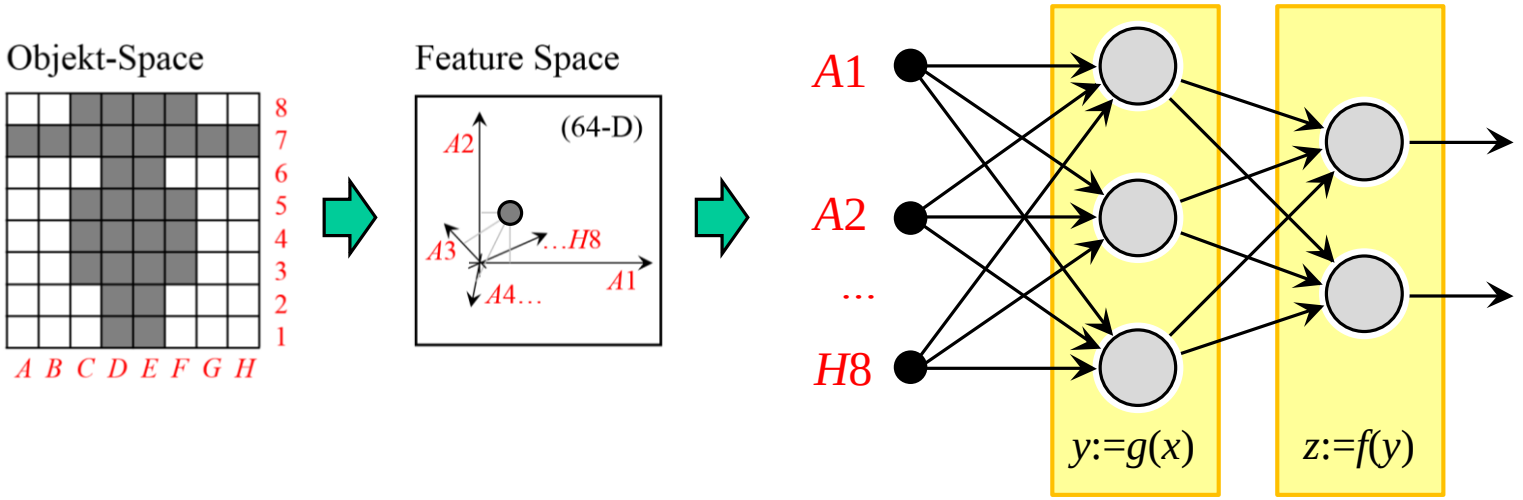
Automatisch gelernt werden bei ANNs die Funktions-Parameter (w_{ij}).

Bei natürlichen neuronalen Netzen ist dies ebenfalls Ergebnis von Lernprozessen (Evolution und Selbstorganisation).

Klassifikation mit einem ANN

Wir nehmen an, das ANN wäre bereits trainiert (siehe nächste Folie)

➔ Wie bestimmt man das Klassenlabel eines neuen Objekts?

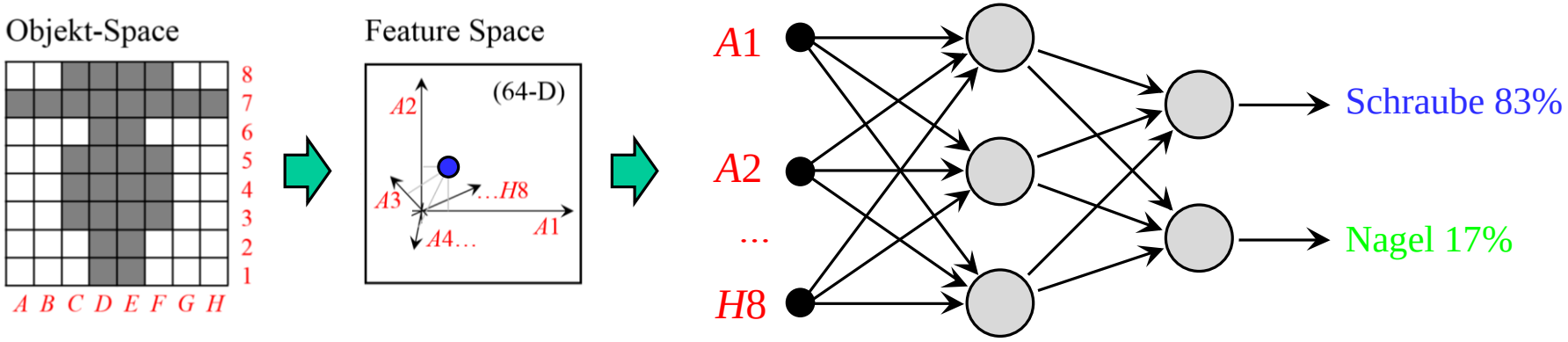


- Wir nehmen unser zu klassifizierendes Objekt.
- Wir wenden darauf die Feature-Transformation an.
- Wir wenden Layer für Layer die Funktionen der künstlichen Neuronen an (*Vorwärtspropagierung*).

Klassifikation mit einem ANN

Wir nehmen an, das ANN wäre bereits trainiert (siehe nächste Folie)

➔ Wie bestimmt man das Klassenlabel eines neuen Objekts?

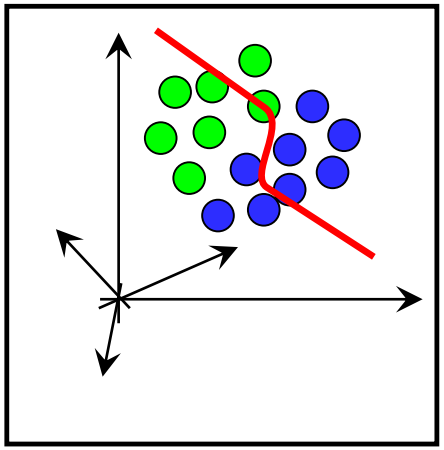


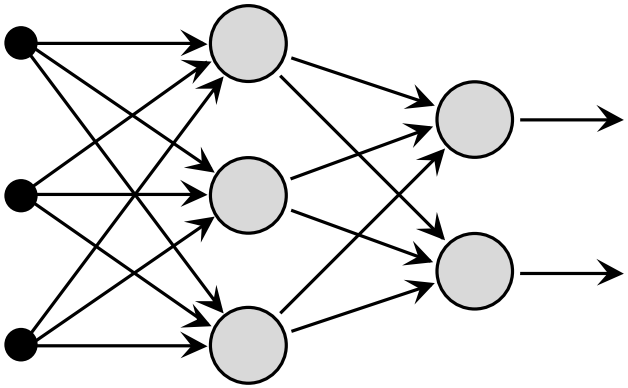
- Wir nehmen unser zu klassifizierendes Objekt.
- Wir wenden darauf die Feature-Transformation an.
- Wir wenden Layer für Layer die Funktionen der künstlichen Neuronen an (*Vorwärtspropagierung*).
- Das Endergebnis ist das gesuchte Klassenlabel.

ANN-Training durch Backpropagation

Ziel ist die optimale Bestimmung der Funtionsparameter (z.B. w_{ij}) für die Trainingsmenge.
Wir starten z.B. mit einer Zufalls-Initialisierung der w_{ij}.

- Schritt 1: Vorwärtspropagierung eines Objekts.

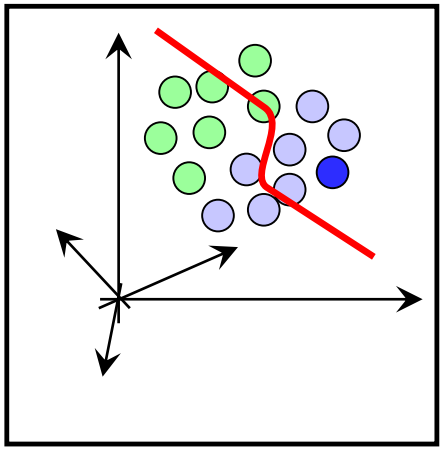


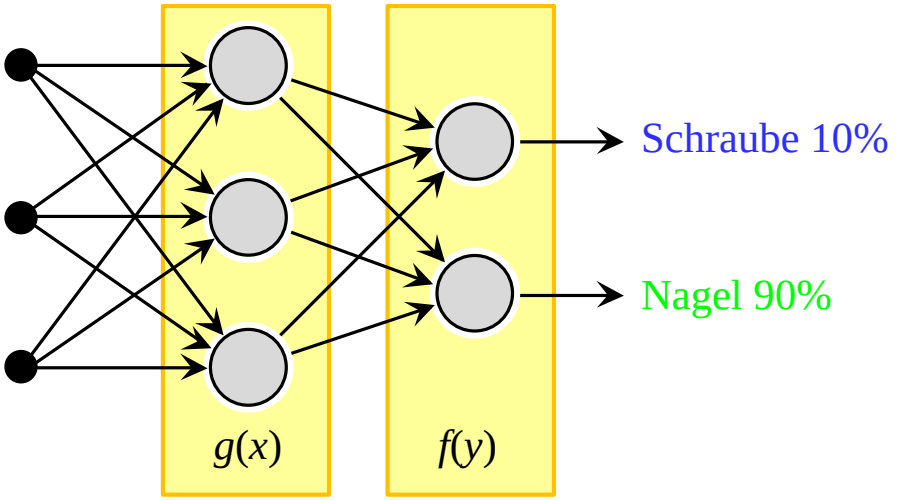


ANN-Training durch Backpropagation

Ziel ist die optimale Bestimmung der Funtionsparameter (z.B. w_{ij}) für die Trainingsmenge.
Wir starten z.B. mit einer Zufalls-Initialisierung der w_{ij}.

- Schritt 1: Vorwärtspropagierung eines Objekts.

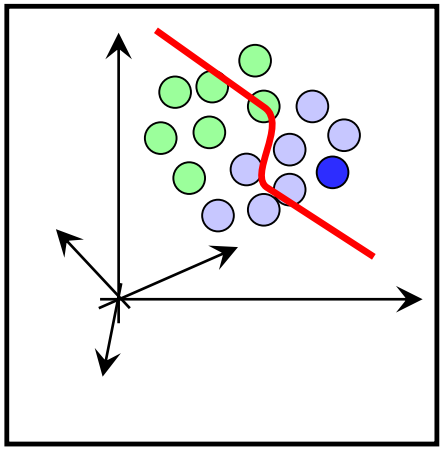


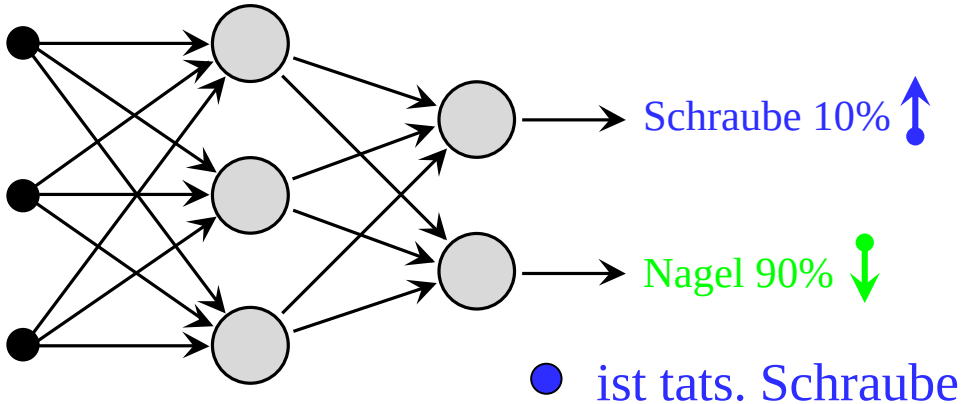


ANN-Training durch Backpropagation

Ziel ist die optimale Bestimmung der Funtionsparameter (z.B. w_{ij}) für die Trainingsmenge.
Wir starten z.B. mit einer Zufalls-Initialisierung der w_{ij}.

- Schritt 1: Vorwärtspropagierung eines Objekts.





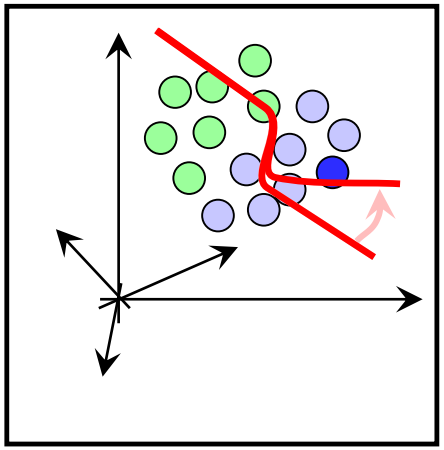
ist tats. Schraube!
(große Abweichung zur Vorhersage des ANN)

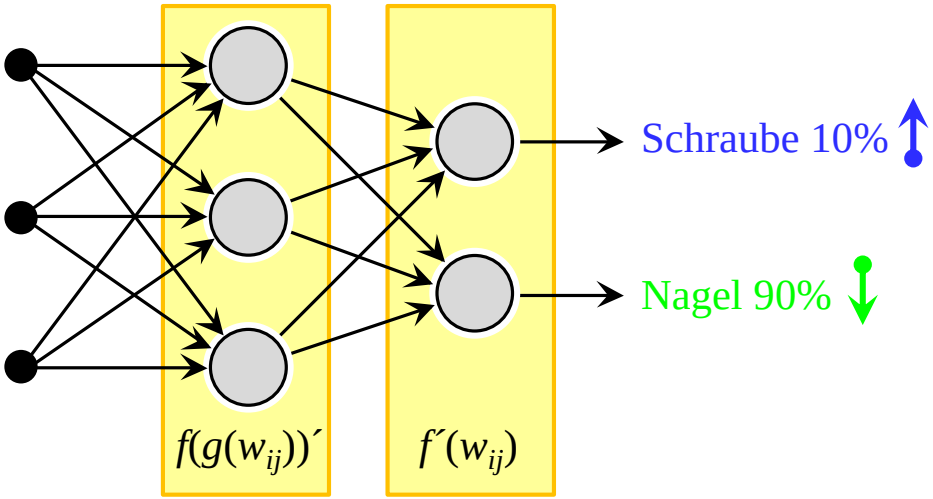
- Schritt 2: Bestimmung des Fehlers

ANN-Training durch Backpropagation

Ziel ist die optimale Bestimmung der Funtionsparameter (z.B. w_{ij}) für die Trainingsmenge.
Wir starten z.B. mit einer Zufalls-Initialisierung der w_{ij}.

- Schritt 1: Vorwärtspropagierung eines Objekts.

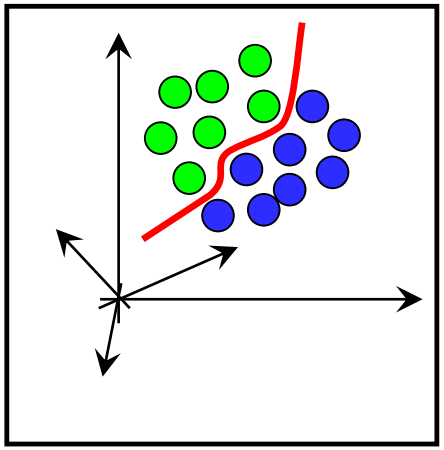


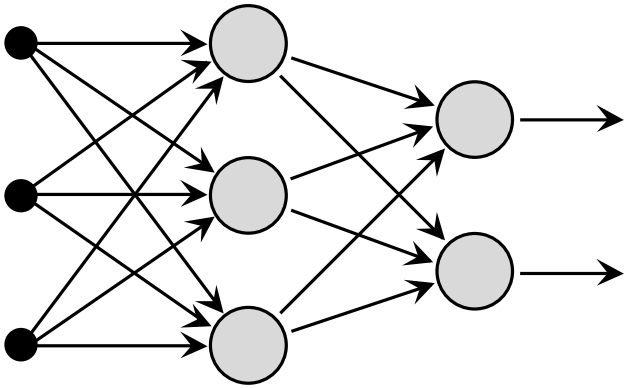


- Schritt 2: Bestimmung des Fehlers
- Schritt 3: Entgegen der Richtung der Vorwärtspropagierung: Anpassen der w_{ij}, so dass Fehler kleiner wird.

ANN-Training durch Backpropagation

Ziel ist die optimale Bestimmung der Funtionsparameter (z.B. w_{ij}) für die Trainingsmenge.





Die Schritte 1-3 werden so lange für alle Objekte wiederholt, bis sich der Fehler nicht mehr verbessert (*Konvergenz*).

Anwendung der Kettenregel

Bei der Rückpropagierung des Fehlers treten vielfach verschachtelte mathematische Funktionen auf. Zur Ableitung Kettenregel:

$$f(g(w_{ij}))' = f'(g(w_{ij})) \cdot g'(w_{ij}), \qquad \text{wobei } g'(w_{ij}) := \frac{\partial}{\partial w_{ij}} g(w_{ij})$$

In Wirklichkeit wesentlich mehr verschachtelte Funktionen:

- Jeder Layer definiert eine eigene Funktion *f*, *g* usw.
- Jede zusammengesetzt aus Übertragungs- und Aktivierungsfunkt.
- Am Ende (d.h. ganz außen) steht immer die Fehlerfunktion.

w_{ij} ist ein Vektor → Gradient statt eindimensionale Ableitung.

Stochastic Gradient Descent:

Gehe immer ein kleines Stück in Richtung der Fehler-Verringerung:

$$w_{ij} := \eta - \frac{\partial}{\partial w_{ij}} \text{error}(f(g(w_{ij}))); \qquad (\eta\text{: Lernrate}).$$

Fazit

- Tiefe Neuronale Netze (mit vielen Layers) haben zahlreiche Wettbewerbe gewonnen, z.B. zur Handschrifterkennung, Bilderkennung usw.
- Es gibt auch Nachteile:
  - Große Anzahl von Trainingsdaten ist nötig.
  - ANNs neigen zur Überanpassung (auswendig lernen der Traingsdaten).
- Aktuelles Projekt: DermaScreen
  - Studentische Existenzgründung (Dominik Seliger, Christian Ludwigs, Simon Schäfer).
  - Hautveränderungen (Muttermale, Leberflecke) mit Handy fotografiert.
  - Neuronales Netz (Server) klassifiziert auf malignes Melanom u.a.
  - ANNs haben in einer Studie Erkennungsraten vergleichbar mit Dermatologen (Nature Vol 542, Februar 2017).
  - EXIST-Stipendium für die Existenzgründung beantragt.