

Einführung in die Computerlinguistik

Neuronale Netze

WS 2014/2015
Vera Demberg

Neuronale Netze – Was ist das?

- Einer der **größten Fortschritte** in der **Sprachverarbeitung und Bildverarbeitung** der letzten Jahre:

NEURONALE NETZE

- Spracherkennungsfehlerrate soweit reduziert, dass freie Eingaben möglich sind.
- Gesichter / Katzen / Menschen können automatisch erkannt werden.

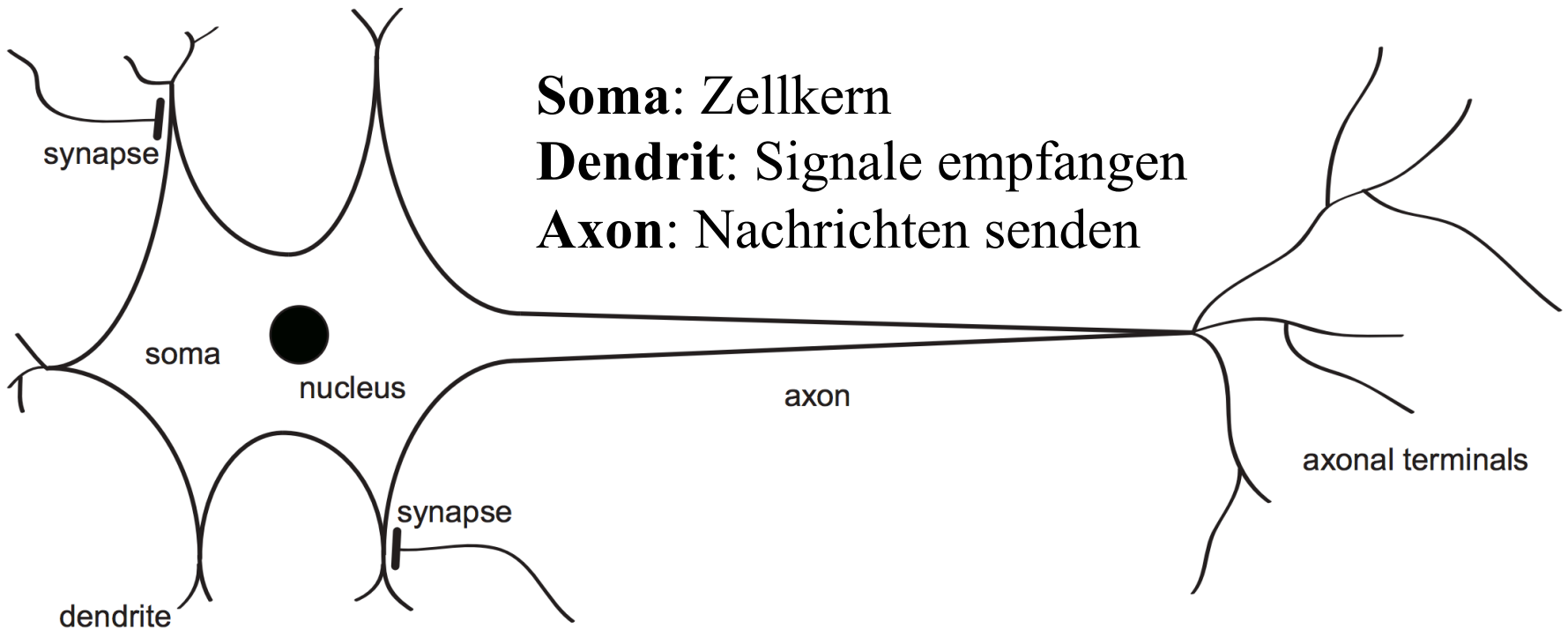
Neuronale Netze: Das Katzenerkennungsneuron

Input: Standbilder von Youtube videos

Ein Neuron hat sich auf die Erkennung von Katzen spezialisiert.



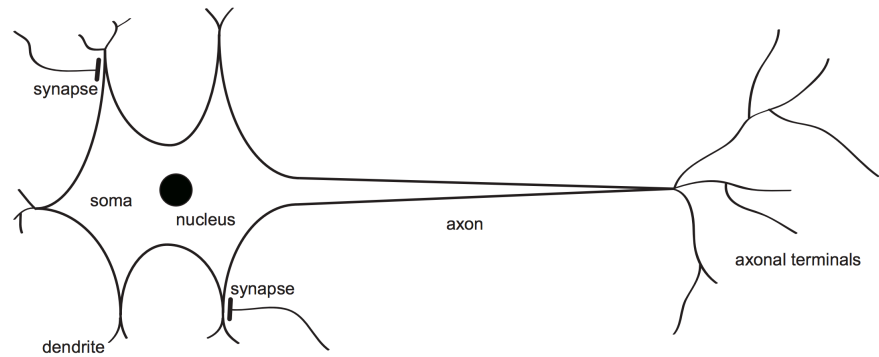
Neuronen im Gehirn



Soma: Zellkern
Dendrit: Signale empfangen
Axon: Nachrichten senden

schematisches Bild eines menschlichen Neurons im Gehirn

Neuronen im Gehirn

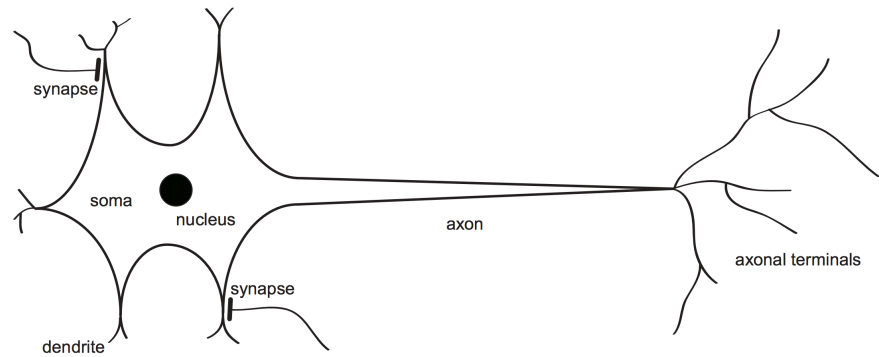


Wichtige relevante Fakten:

- Jedes Neuron hat viele **Dendriten** (Signale empfangen) und ein **Axon** (Signale senden) mit mehreren Terminalen.
- Das Axon eines Neurons **sendet ein Signal** an das Dendrit eines anderen Neurons.
- Axonterminal und Dendrit sind durch eine **Synapse** verbunden (kann Signal **verstärken** oder **dämpfen**)
- Starkes einkommendes Signal → Neuron sendet selbst ein Signal (elektrische Aktivität)

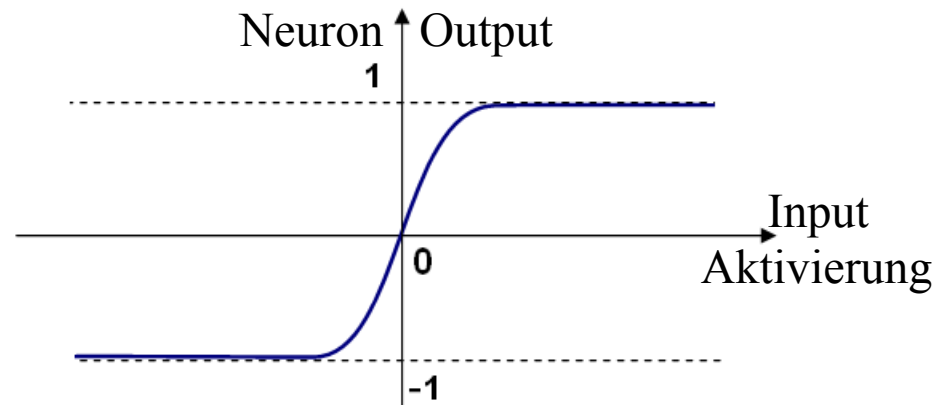
Neuronen im Gehirn

Wann feuert ein Neuron?

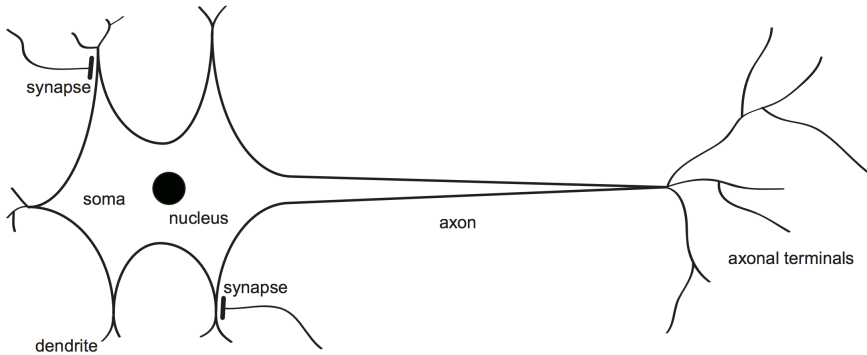


Aktivierung hängt ab von:

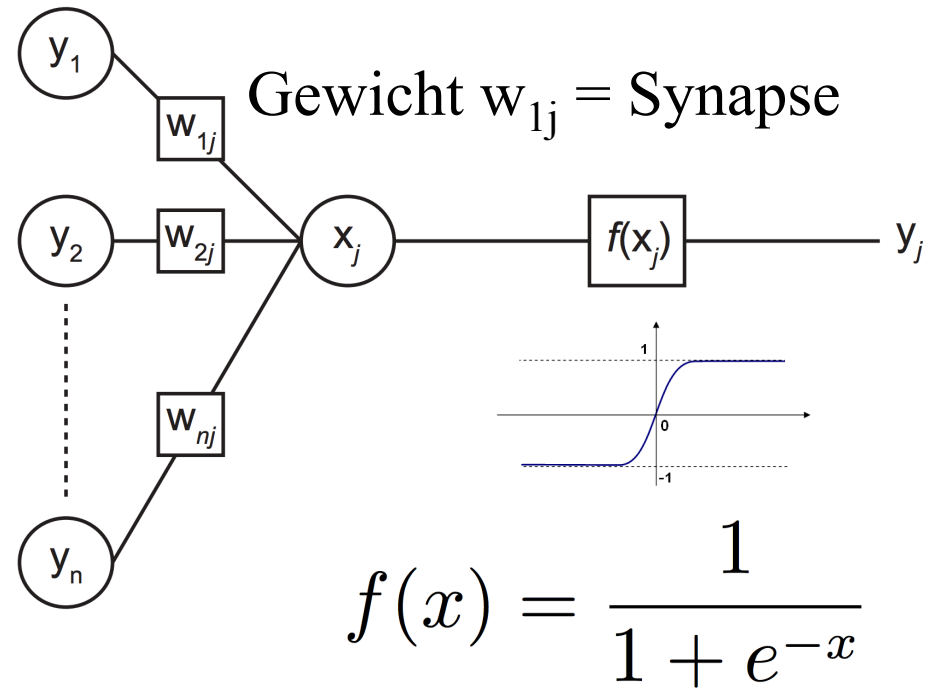
- Wieviele Signale gehen ein bei den Dendriten?
- Synapse: verstärkend oder dämpfend?
- Zufall: Aktivierungsfunktion ist nicht deterministisch; höhere Aktivierung erhöht Wahrscheinlichkeit, dass das Neuron feuert.



Ein künstliches Neuron

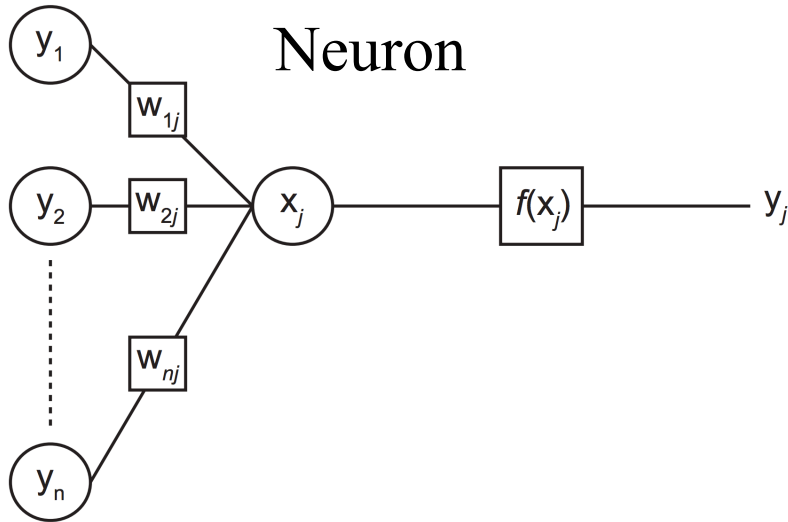


Lernen:
Wenn sich die Synapsen-
gewichte ändern.



künstliches Neuron aus einem
Neuronalen Netz

Neuronale Netze



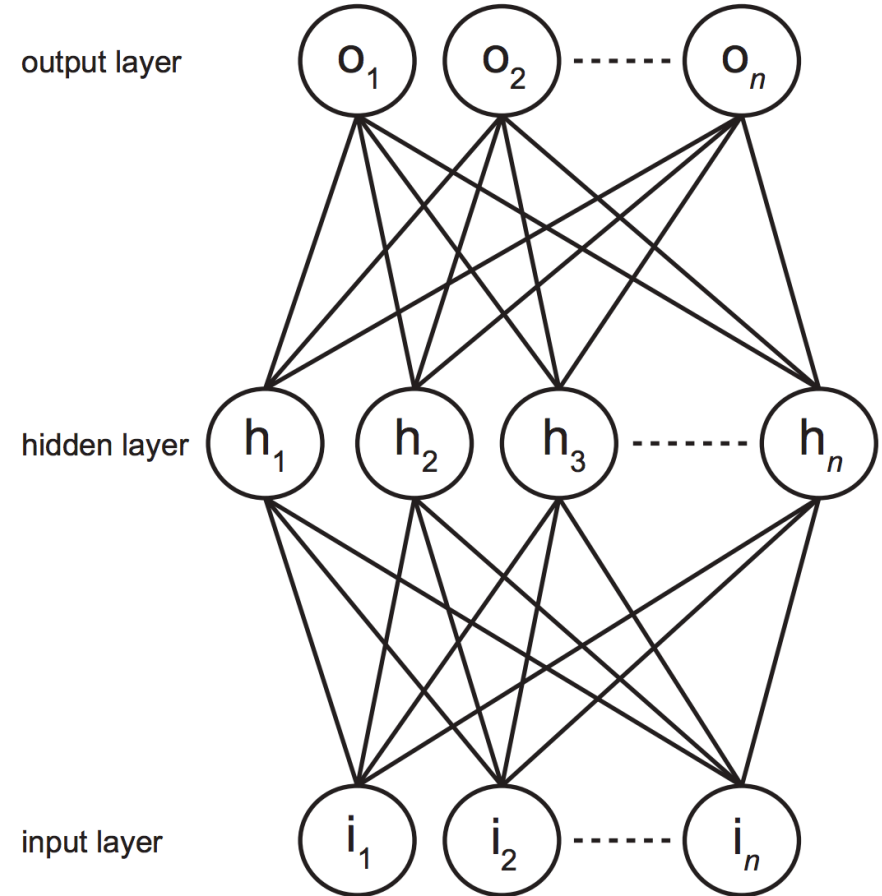
Input: Beobachtungen

- Bilder
- Laute

Output: Erkannte Muster

- Objekt
- Wort

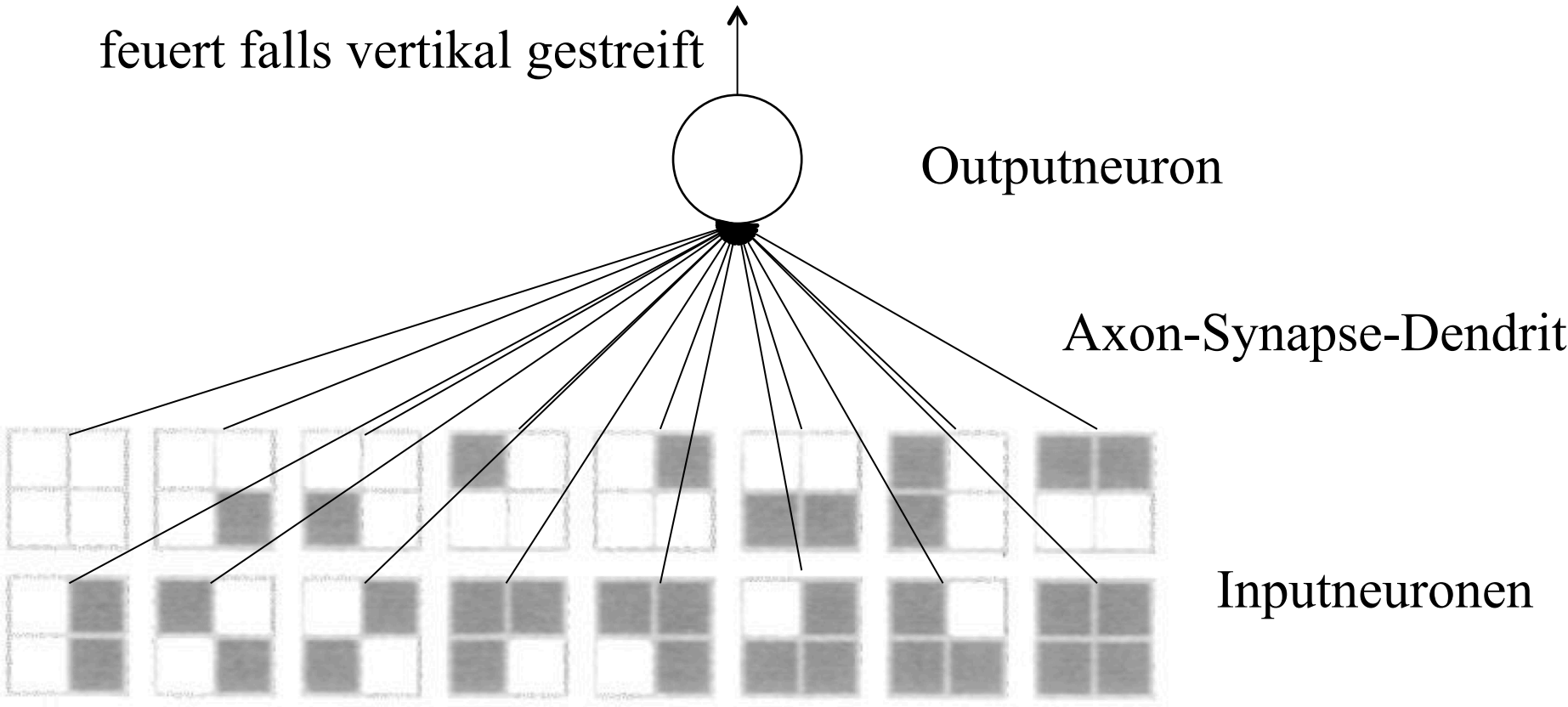
Neuronales Netz



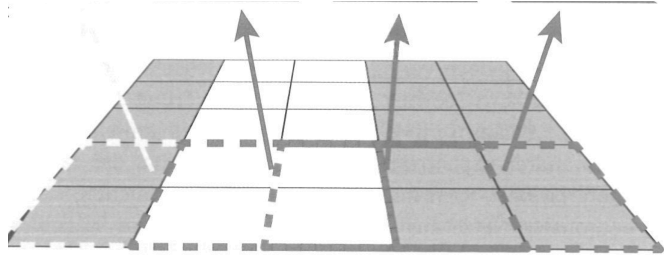
Perzeptron

Erstes Beispiel: **Streifendetektor** (vertikale Streifen)

feuert falls vertikal gestreift

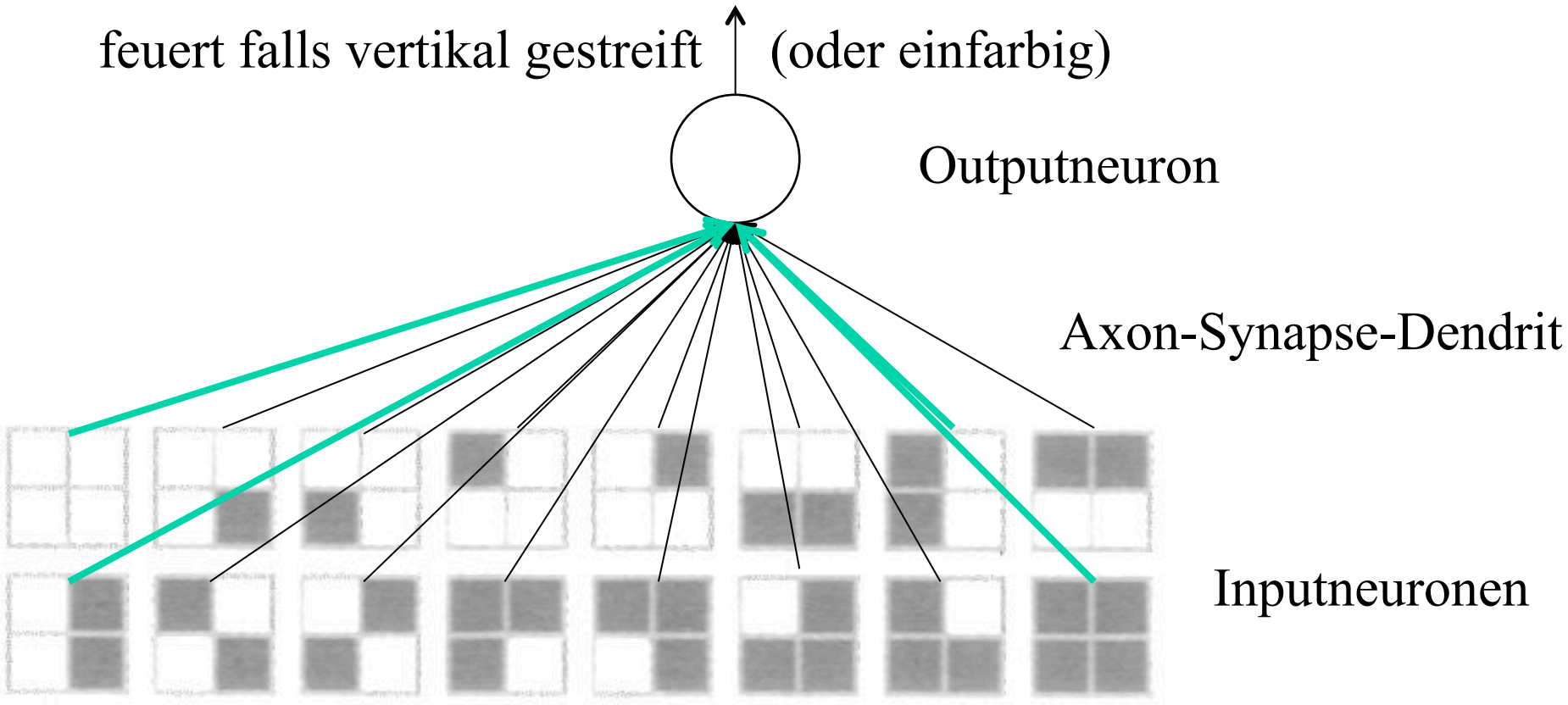


Perzeptron

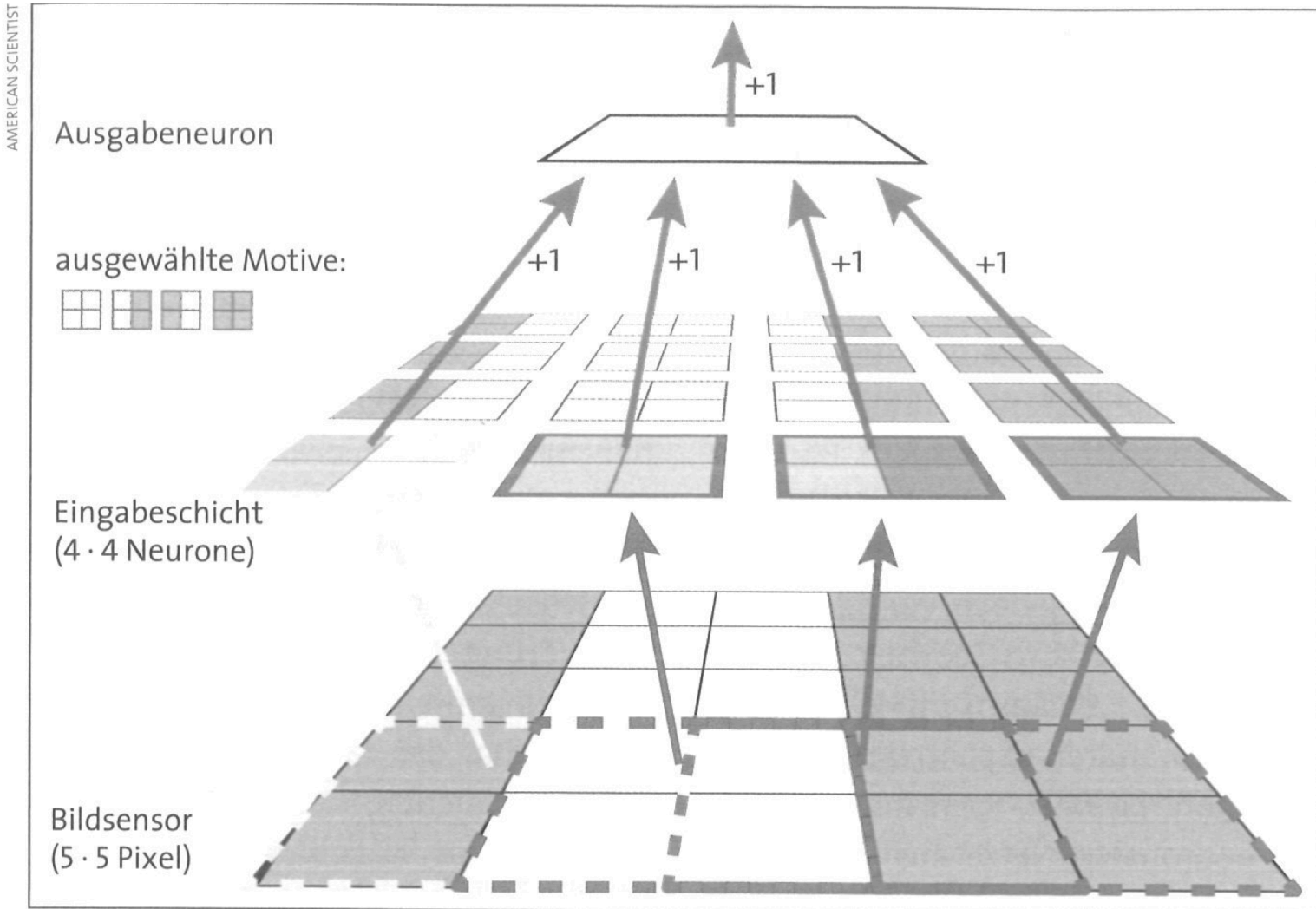


Erstes Beispiel: **Streifendetektor** (vertikale Streifen)

feuert falls vertikal gestreift (oder einfarbig)



Streifenperzeptron



Neuronale Netze sollen selbst lernen

- Streifenperzeptron: wir haben “programmiert” welche Inputneuronen wann feuern sollen, und die “Gewichte” selbst gesetzt (überall zu 1).
- Wir Menschen haben auch Neuronen in unserer primären Sehrinde, die auf Streifen reagieren.
- Aber Kindern muss man ja auch nicht erklären, wie sie einen Streifen sehen... kann das neuronale Netz das auch selbst lernen?

Lernende Neuronale Netze

Damit das Netz lernen kann, müssen wir

- ihm viele Bilder mit und ohne Streifen zeigen
- Feedback geben, auf welchen Bildern Streifen sind

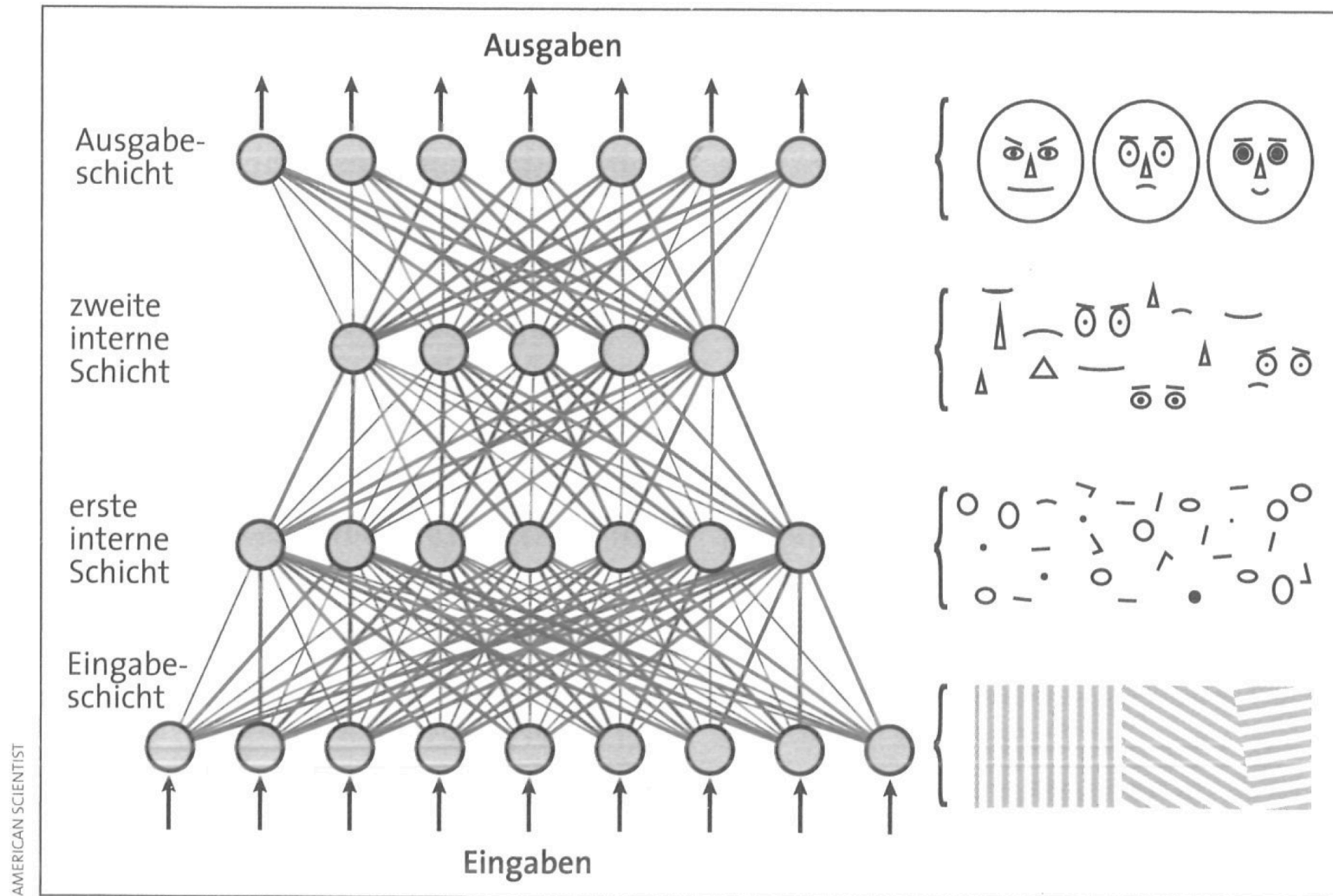
Richtige Antwort → das Netz muss nichts tun

Falsche Antwort → das Netz muss mind. ein Neuron in der Eingabeschicht anweisen, gegenteilige Antwort zu geben

Streifenperzeptron → Neuronales Netz

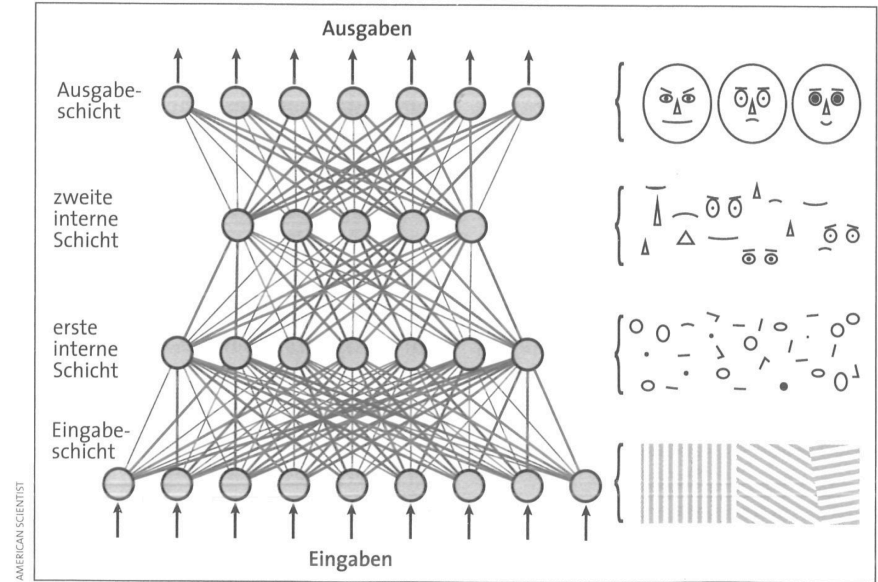
- Verbesserungsmöglichkeiten für einfaches Perzeptron:
 - Tiefere Struktur als input-output (eine oder mehr “hidden” layers)
 - Gewichtung (wie Synapsen) anstelle von Erkennung ob ein bestimmtes Neuron gefeuert hat oder nicht
 - Eingabeneuronen können viel mehr Input bekommen (z.B. gesamtes Bild geht an jedes Eingabeneuron)

Tiefe neuronale Netze

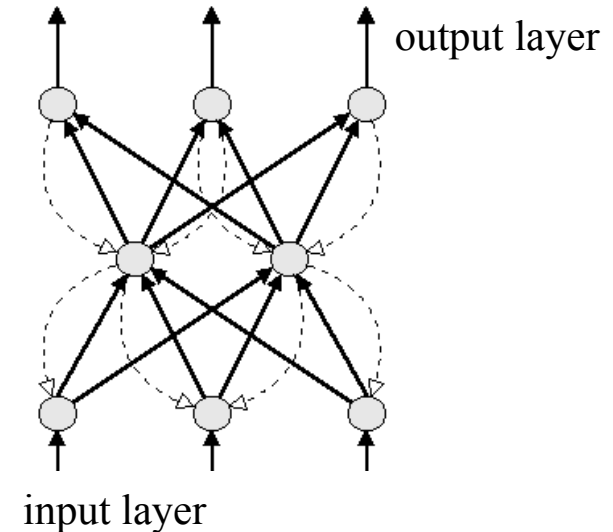


Lernen mit tiefen neuronalen Netzen

- Wie lernt man jetzt?
- In welcher Schicht liegt der Fehler, falls am Ende die Ausgabe falsch ist?



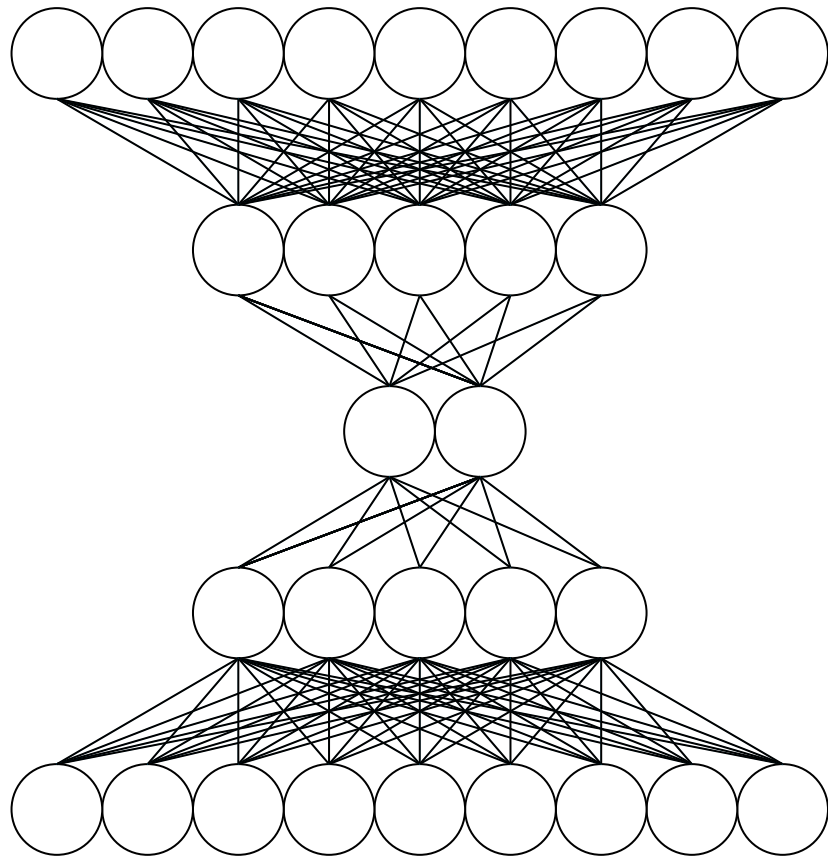
→ “Backpropagation”
jede Schicht meldet an die darunter weiter, falls ein Fehler gemacht wurde.



Lernen ohne Lehrer

- Wir müssen nicht mehr genau sagen, wie ein Objekt zu erkennen ist.
- Aber: immernoch müssen wir dem Netz mitteilen, was auf dem Gesamtbild zu sehen ist.
 - für Gesichtererkennung müssten wir immer mitteilen, ob auf einem bestimmten Foto ein Gesicht drauf ist.
- Es geht! Google brain: 1 Millarde Verbindungen, 9 Schichten, 10 Millionen Bilder, bestimmte Form von Architektur eines neuronalen Netzes: **Autoencoder**

Autoencoder: optimiert für gute Komprimierung



output layer (reconstruct input)

hidden layer

bottleneck hidden layer

hidden layer

input layer

Autoencoder lernt, Katzen zu erkennen ohne Trainingsdaten oder Labels

Katzenneuron



Bilder von denen der Autoencoder glaubt, dass es Katzen sind.



Figure 16. Top: most responsive stimuli on the test set for the cat neuron.

Autoencoders for object recognition

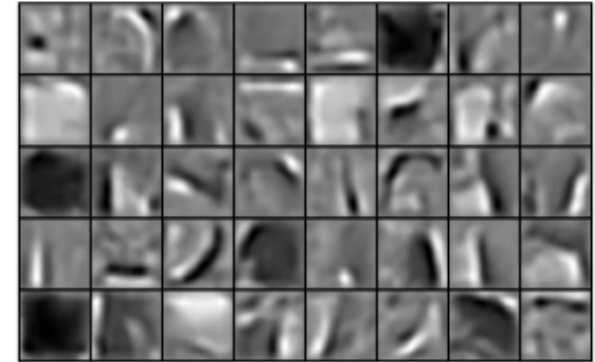
faces



cars



elephants



Neuronale Netze

Es gibt auch **Nachteile**:

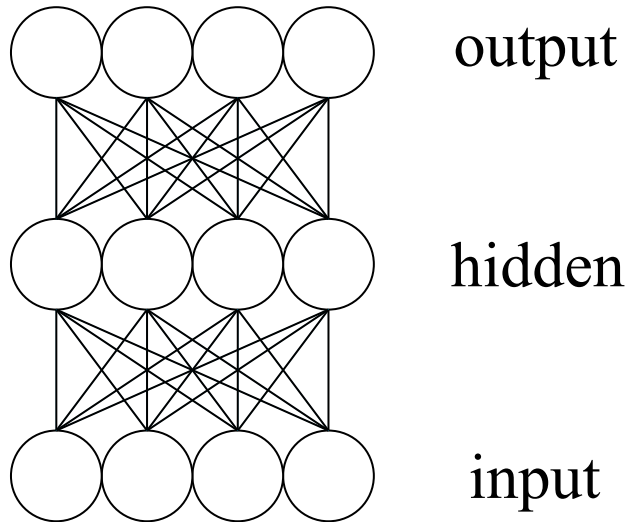
- Theoriedefizit!
 - viele Parameter – aber nur Heuristiken, wie wir sie setzen sollen.
 - Anzahl Schichten???
 - Anzahl Neuronen???
 - Anfangsgewichte???
 - Lernprotokoll???
- derzeit: ausprobieren!

Was hat das mit Coli zu tun???

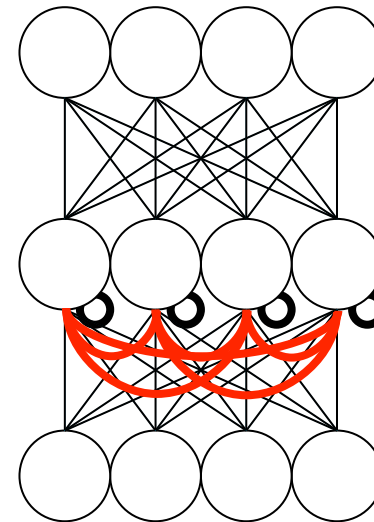
- Objekterkennung nur als anschauliches Beispiel
- erster großer Erfolg: **Spracherkennung** (2009)
 - Fehlerrate sank um 25%: absolut phänomenal.
- Verfahren ebenso erfolgreich angewendet auf
 - Language modelling
 - POS tagging, named entity recognition
 - Sprachsynthese
 - Document retrieval / Information retrieval
 - semantischer Ähnlichkeit, Paraphrasenerkennung
 - Maschinelle Übersetzung
 - Sentiment analysis

Tiefe neuronale Netze für NLP

Feedforward Neural Net

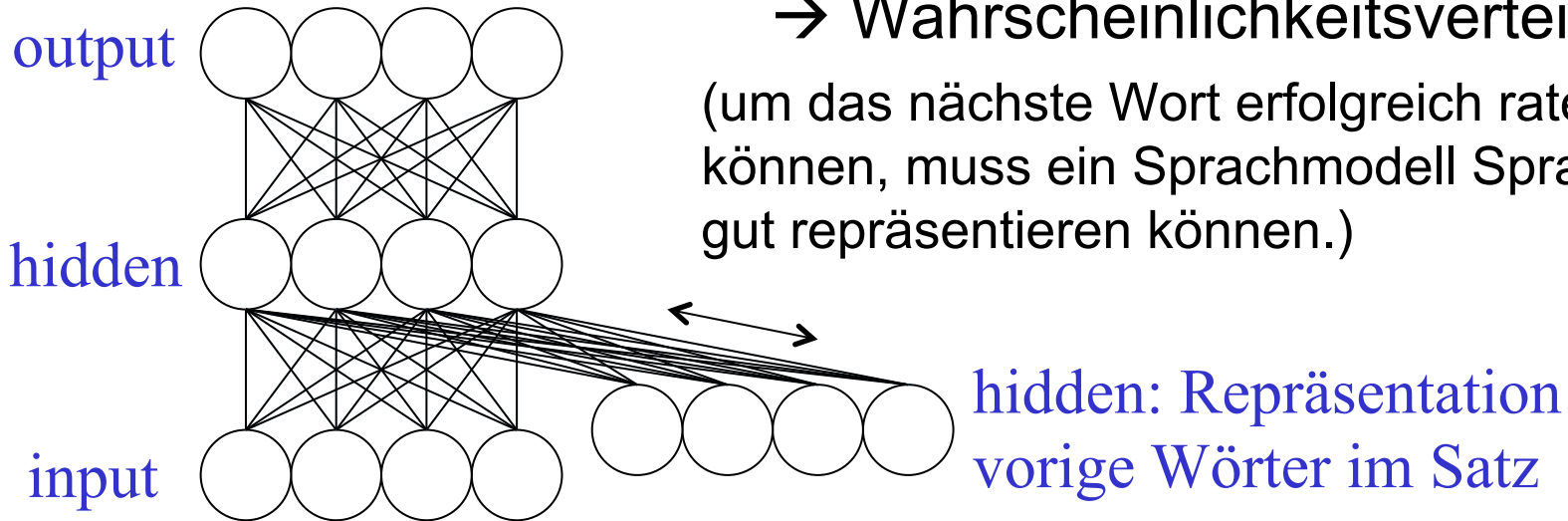


Recurrent Neural Net



Recurrent Neural Net als Sprachmodell

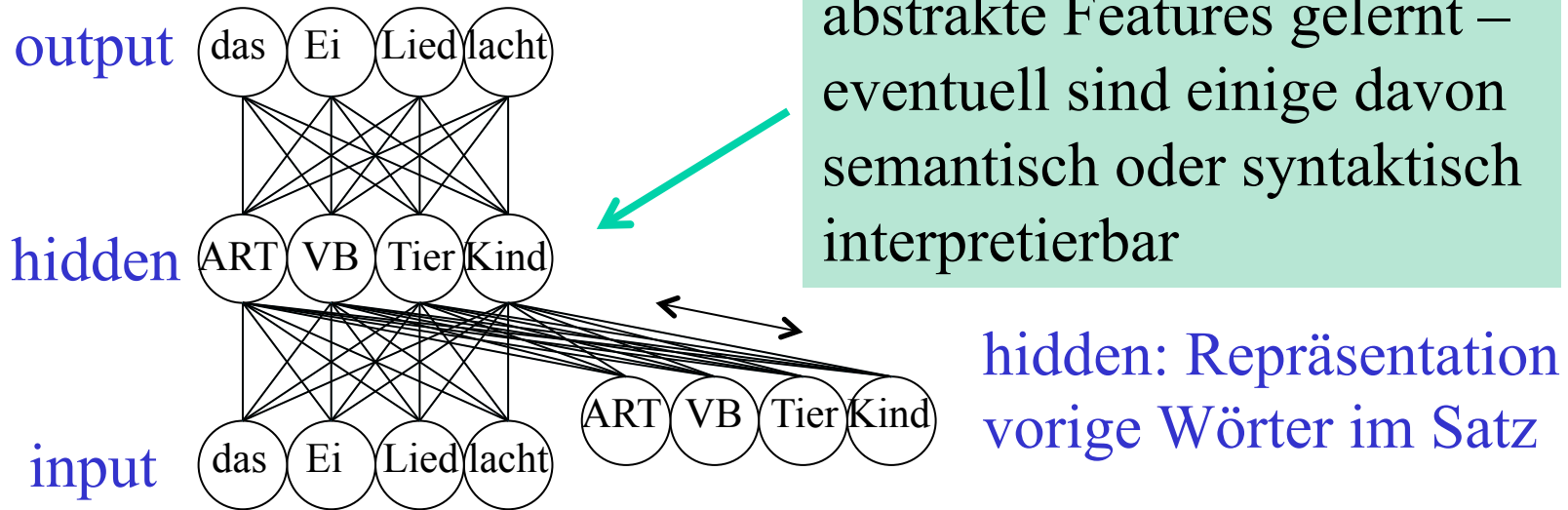
- input: ein Wort
- output: rate das nächste Wort
→ Wahrscheinlichkeitsverteilung
(um das nächste Wort erfolgreich raten zu können, muss ein Sprachmodell Sprache gut repräsentieren können.)



input: ein Neuron pro Wort im Vokabular
output: ein Neuron pro Wort im Vokabular
hidden: repräsentiert den bisherigen Satz

Recurrent Neural Net als Sprachmodell

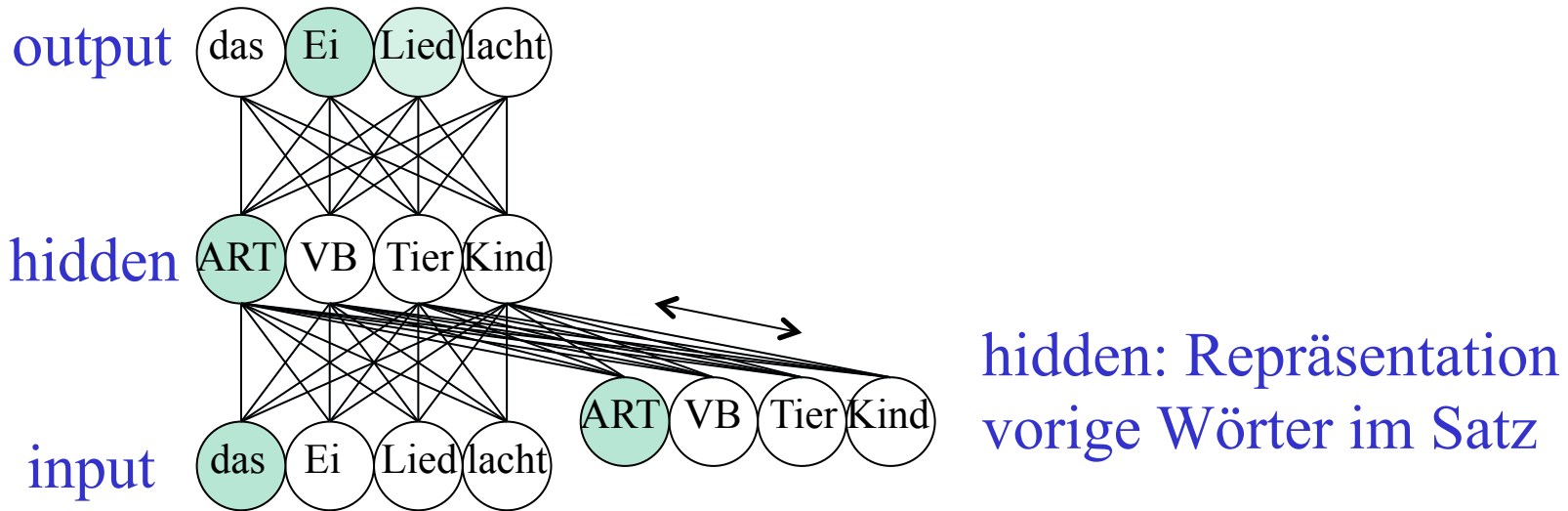
output: ein Neuron pro Wort im Vokabular



input: ein Neuron pro Wort im Vokabular

Recurrent Neural Net als Sprachmodell

output: ein Neuron pro Wort im Vokabular

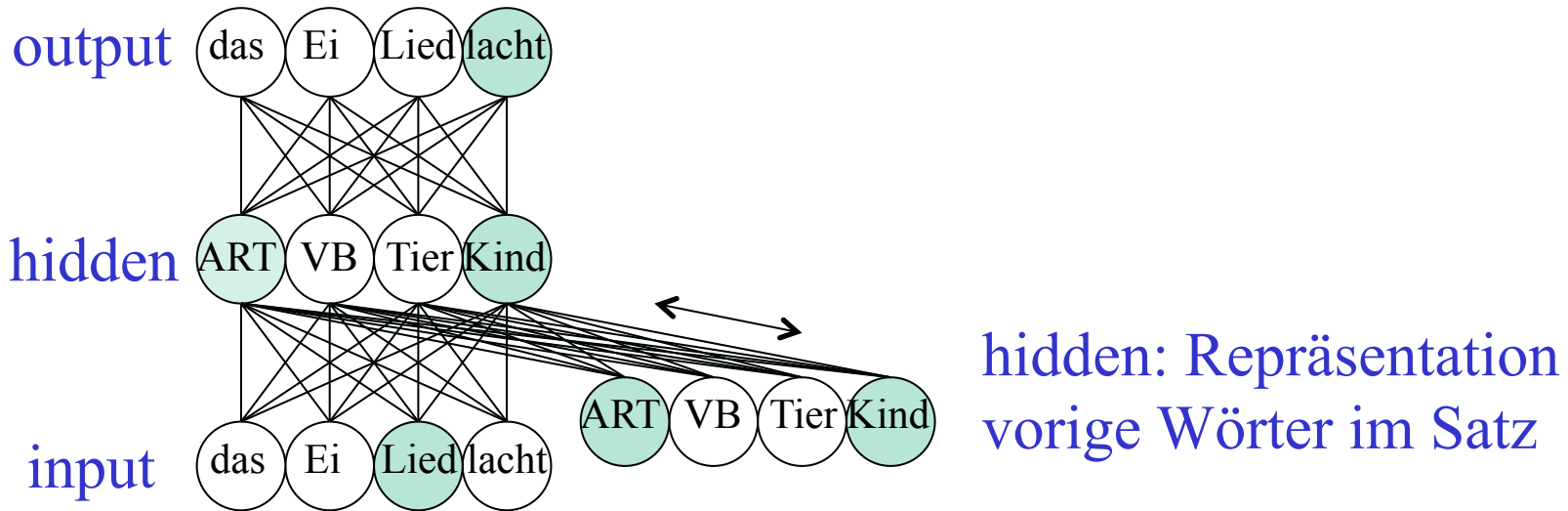


input: “das”

input: ein Neuron pro Wort im Vokabular

Recurrent Neural Net als Sprachmodell

output: ein Neuron pro Wort im Vokabular



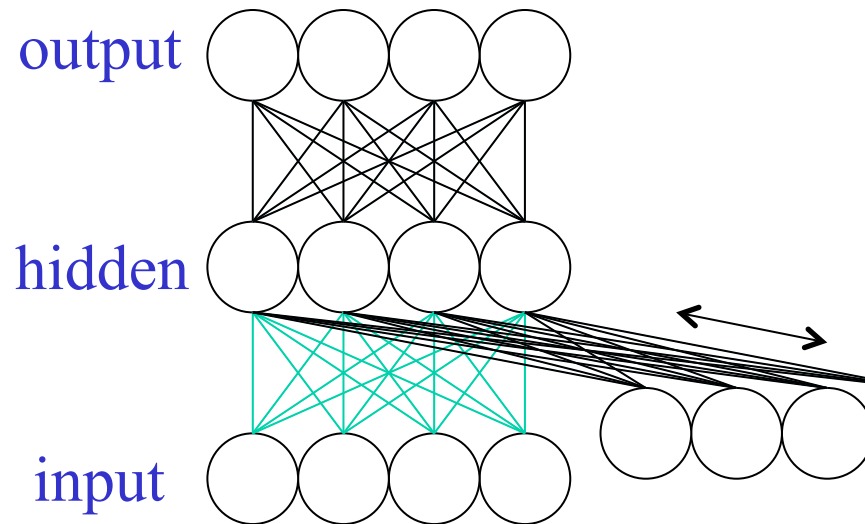
input: "Lied"

input: ein Neuron pro Wort im Vokabular

RNN Language Model

- mit dem RNN Sprachmodell kann man das nächste Wort besser vorhersagen als vorige Modelle.
- jedes Wort wird mit dem trainierten Modell als Vektor dargestellt.

Bank =	0.13	Tisch
	0.05	Schuh
	0.24	setzen
	0.00002	Lampe
	0.0005	Rad
	0.1211	Geld



(dies ist ein ähnlicher Effekt wie Kontextwörter
auszählen, siehe letzte VL)

...

(A237) ... Für diejenigen, denen *Komfort wichtig* ist, haben wir eine Bank mit
leicht schwingender Rückenlehne entwickelt. ...

(A295) ... Ich *suche* noch eine Bank für meinen *Garten* und *sondiere* deshalb
gerade *Angebote*. ...

(A303) ... Habe im *März* 2000 einen höheren *Betrag* bei einer Bank
angelegt. ...

(A452) ... Beim *Test Anlageberatung* der Banken *löste* kein *Institut* die
einfache Frage nach einer *sicheren Anlage* *wirklich gut*. ...

...

Vektorrepräsentation von Wörtern



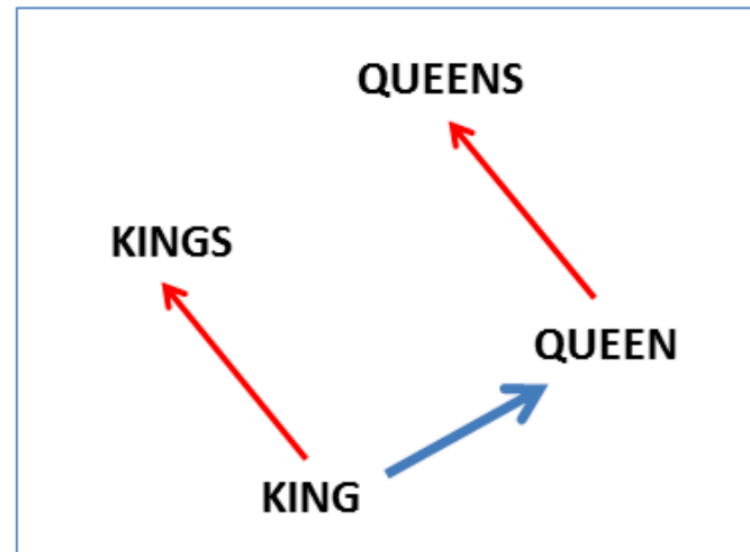
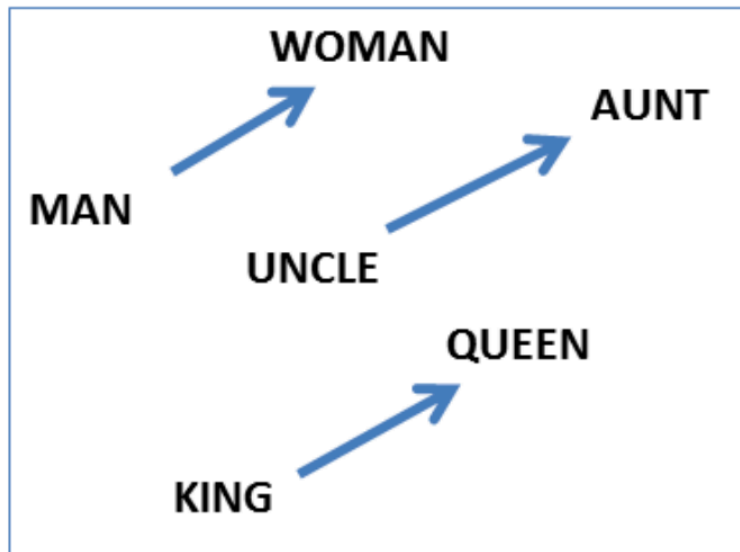
Syntaktische und Semantische Regularitäten in RNNs repräsentiert

queen – king + man = ?

woman

queens – queen + king = ?

kings

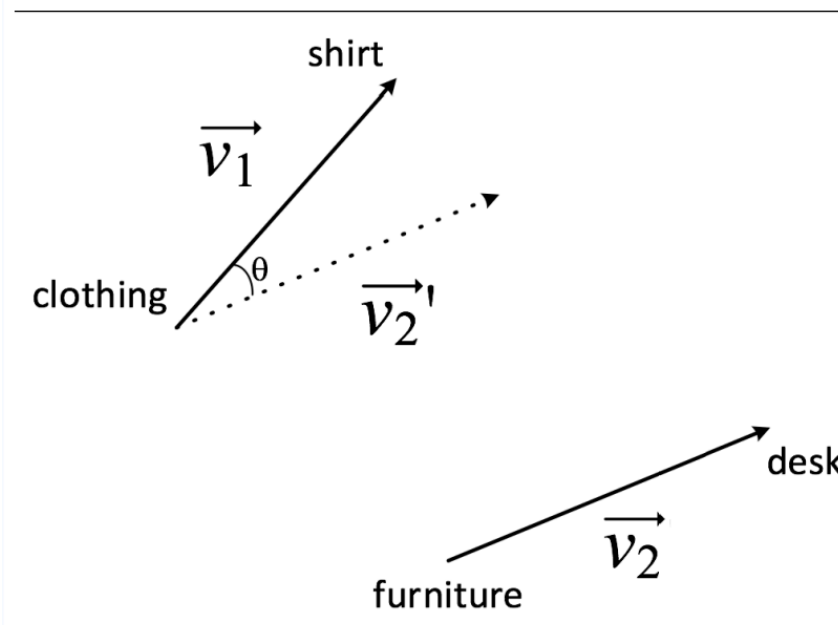


(Mikolov et al., 2013)

Syntaktische und Semantische Regularitäten in RNNs repräsentiert

Clothing is to shirt as furniture is to?

clothing – shirt + furniture = ? desk



(Mikolov et al., 2013)