TH BINGEN
University of Applied Sciences

Künstliche Neuronale Netze

Modellbildung mit neuronalen Netzen

Dieter Kilsch

Technische Hochschule Bingen, FB 2 • Technik, Informatik und Wirtschaft

APL Germany, Solingen

26. November 2018

Komfort durch autonomes Auto

2007: Vision

Das Auto lenkt,

der Mensch denkt,

aber nicht ans Fahren.

Komfort durch autonomes Auto

2007: Vision

Das Auto lenkt,

der Mensch denkt,

aber nicht ans Fahren.

Komfort durch autonomes Auto

heute: Vision ?

Das Auto lenkt,

der Mensch denkt,

aber nicht ans Fahren.

Komfort durch autonomes Auto

Erstes autonomes Auto im Straßenverkehr

TU Braunschweig, https://www.tu-braunschweig.de/presse/medien/presseinformationen?year=2010&pinr=133

Das Auto lenkt, der Mensch denkt, ...

Leonie 8.10.2010

Weltweit erstes automatisches Fahren im realen Stadtverkehr

Forschungsfahrzeug „Leonie“ fährt automatisch auf dem Braunschweiger Stadtring

Weltpremiere in Braunschweig: Erstmals fährt heute ein Fahrzeug automatisch im alltäglichen Stadtverkehr. Im Rahmen des Forschungsprojekts „Stadtpilot“ hat die Technische Universität Braunschweig in ihrem Kompetenzzentrum, dem niedersächsischen Forschungszentrum Fahrzeugtechnik, ein Forschungsfahrzeug entwickelt, dass automatisch eine vorgegebene Strecke im regulären Verkehr fährt.

(Künstliche) Intelligenz

Intelligenz

- Individuelle Intelligenz
- Intelligenz einer Gruppe (Schwarmintelligenz)
- Emotionale Intelligenz (Intelligenz der Gefühle)

Künstliche Intelligenz Alan Turing 1950: Turing Test

Eine Maschine hat künstliche Intelligenz, wenn ein Beobachter nicht unterscheiden kann, ob er es mit einem Menschen oder einer Maschine zu tun hat.

(Künstliche) Intelligenz

Intelligenz

- Individuelle Intelligenz
- Intelligenz einer Gruppe (Schwarmintelligenz)
- Emotionale Intelligenz (Intelligenz der Gefühle)

Künstliche Intelligenz Alan Turing 1950: Turing Test

Eine Maschine hat künstliche Intelligenz, wenn ein Beobachter nicht unterscheiden kann, ob er es mit einem Menschen oder einer Maschine zu tun hat.

(Künstliche) Intelligenz

Intelligenz

- Individuelle Intelligenz
- Intelligenz einer Gruppe (Schwarmintelligenz)
- Emotionale Intelligenz (Intelligenz der Gefühle)

Künstliche Intelligenz Alan Turing 1950: Turing Test

Eine Maschine hat künstliche Intelligenz, wenn ein Beobachter nicht unterscheiden kann, ob er es mit einem Menschen oder einer Maschine zu tun hat.

Forschungsziel

Intelligente Leistungen analysieren und berechenbar machen.

Vergleich Gehirn - Rechner

Gehirn und herkömmliche Rechner

Hohe Qualitäten bei der Erfüllung unterschiedlicher Aufgaben

Gehirn

- hohe Parallelität
- Fehlertoleranz
- Mustererkennung
- Verallgemeinern
- Selbstorganisation
- ca. 10^{11} Neuronen, abnehmend auf 10^7
- Jedes Neuron mit ca. 10^4 verbunden

Rechner

Vergleich Gehirn - Rechner

Gehirn und herkömmliche Rechner

Hohe Qualitäten bei der Erfüllung unterschiedlicher Aufgaben

Gehirn

- hohe Parallelität
- Fehlertoleranz
- Mustererkennung
- Verallgemeinern
- Selbstorganisation
- ca. 10^{11} Neuronen, abnehmend auf 10^7
- Jedes Neuron mit ca. 10^4 verbunden

Rechner

- Präzision
- Fehlerloses Speichern
- Schnelles Ausführen eines Algorithmus
- von Neumann Architektur
- kaum vernetzt

Vergleich Gehirn - Rechner

Gehirn und herkömmliche Rechner

Hohe Qualitäten bei der Erfüllung unterschiedlicher Aufgaben

Gehirn

- hohe Parallelität
- Fehlertoleranz
- Mustererkennung
- Verallgemeinern
- Selbstorganisation
- ca. 10^{11} Neuronen, abnehmend auf 10^7
- Jedes Neuron mit ca. 10^4 verbunden

Rechner

- Präzision
- Fehlerloses Speichern
- Schnelles Ausführen eines Algorithmus
- von Neumann Architektur
- kaum vernetzt

Problemlösungsprozess

Algorithmisch

- Aufgabenstellung intuitiv lösen
- In numerische Algorithmen umsetzen
- Als Programm realisieren
- Voraussetzungen streng einhalten

Expertensystem

Neurale Netze

Problemlösungsprozess

Algorithmisch

- Aufgabenstellung intuitiv lösen
- In numerische Algorithmen umsetzen
- Als Programm realisieren
- Voraussetzungen streng einhalten

Expertensystem

- Aufgabenstellung intuitiv lösen
- Regeln aufstellen
- Regeln anwenden
- auf ähnliche Probleme anwendbar

Neurale Netze

Problemlösungsprozess

Algorithmisch

- Aufgabenstellung intuitiv lösen
- In numerische Algorithmen umsetzen
- Als Programm realisieren
- Voraussetzungen streng einhalten

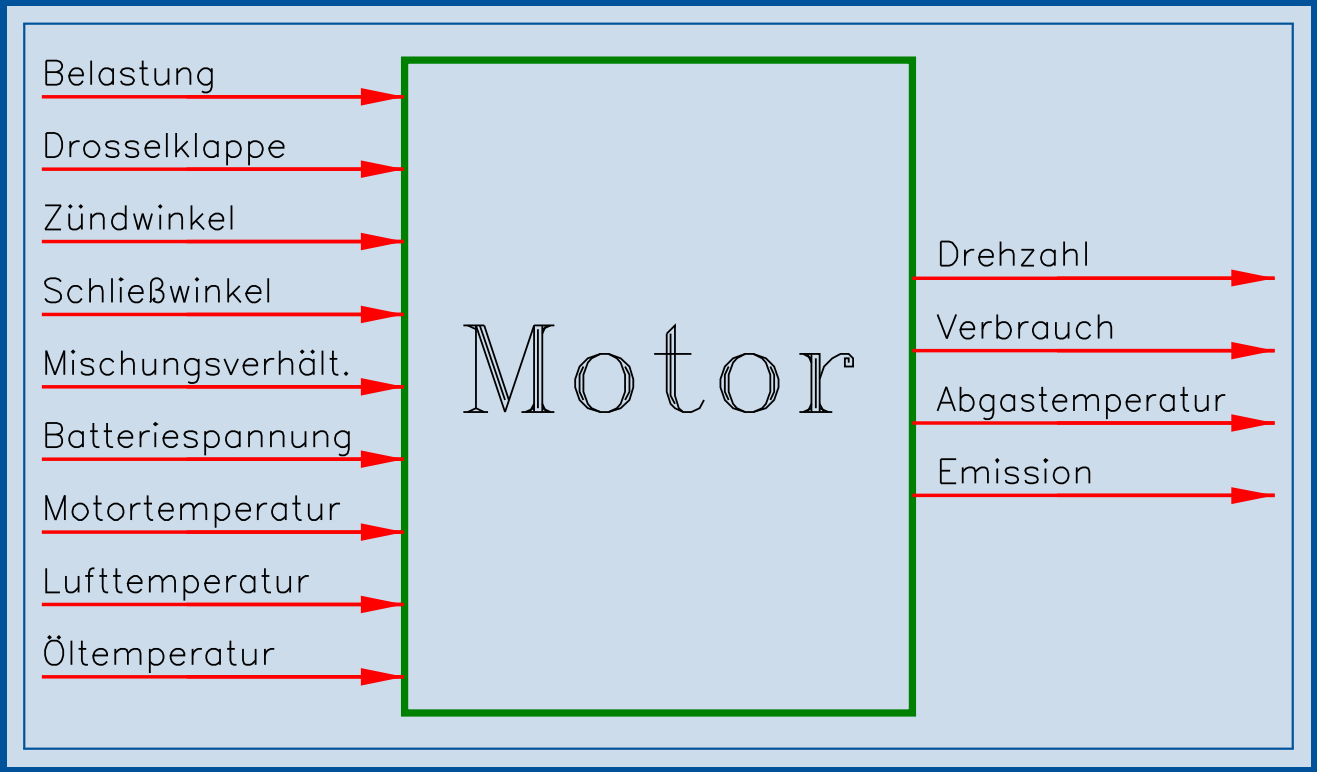
Expertensystem

- Aufgabenstellung intuitiv lösen
- Regeln aufstellen
- Regeln anwenden
- auf ähnliche Probleme anwendbar

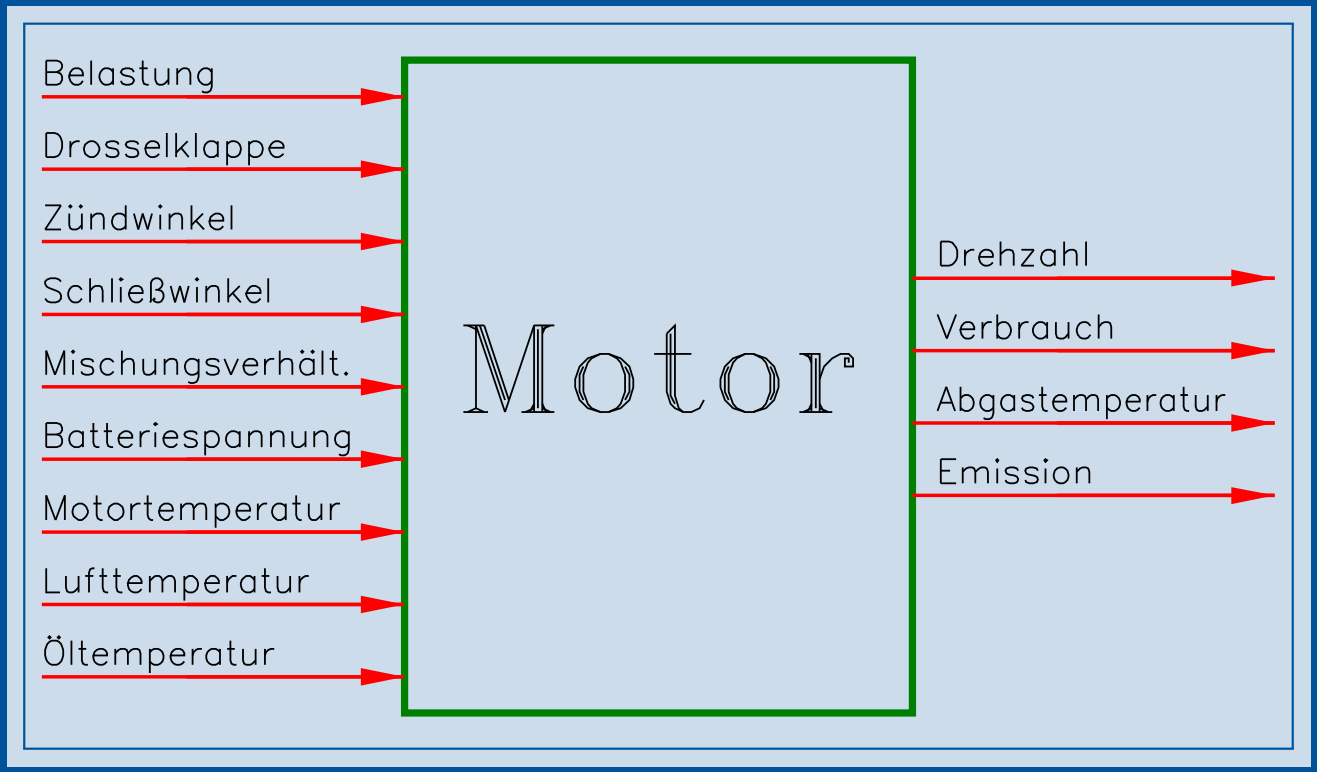
Neurale Netze

- Aufgabenstellung intuitiv lösen
- Vorgabe von Beispieldaten
- auf Basis der Lerndaten verallgemeinern
- auf ähnliche Probleme anwendbar

Physikalisches Wirkverhalten: Motor auf dem Prüfstand



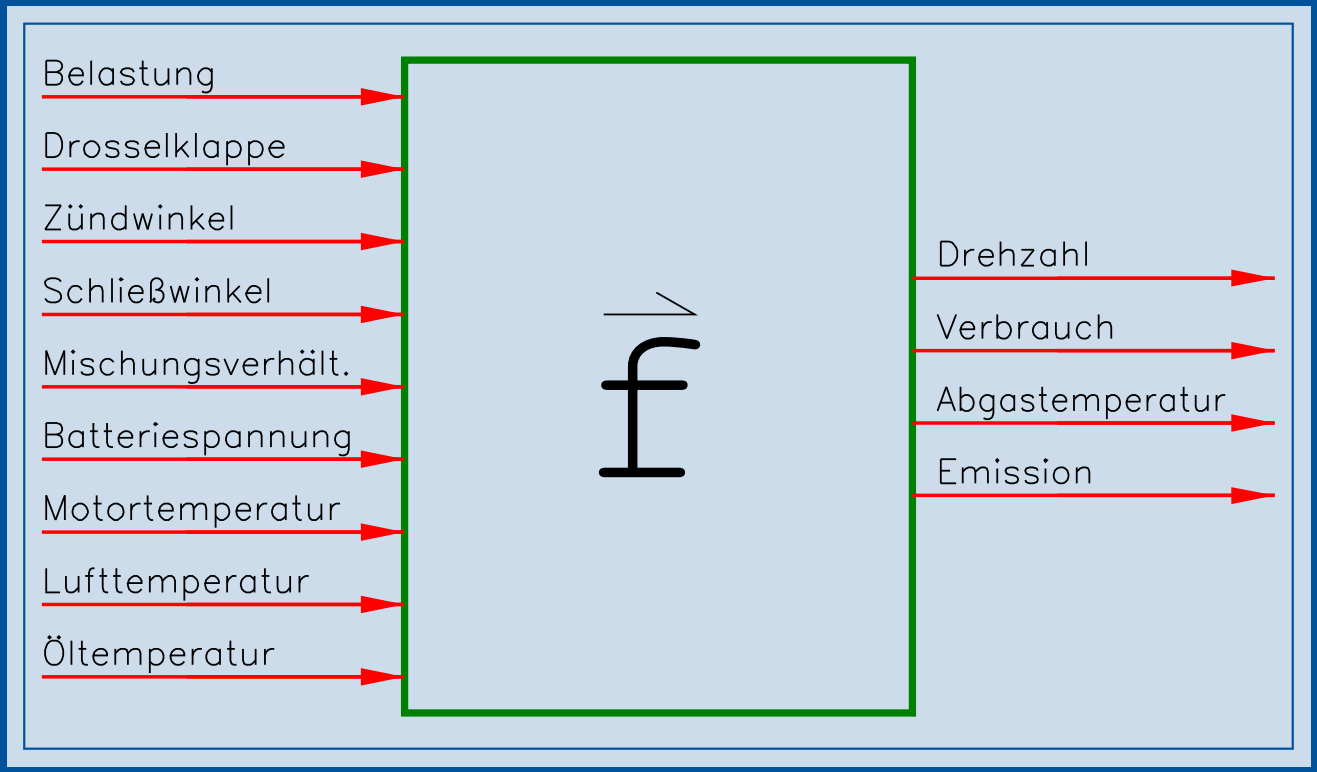
Physikalisches Wirkverhalten: Motor auf dem Prüfstand



Aufgabe und Ziel

- Optimierte stationäre Kennfelder ermitteln.
- Betriebsverhalten im Kaltbetrieb ermitteln
- Reduzierung der Prüfstandläufe

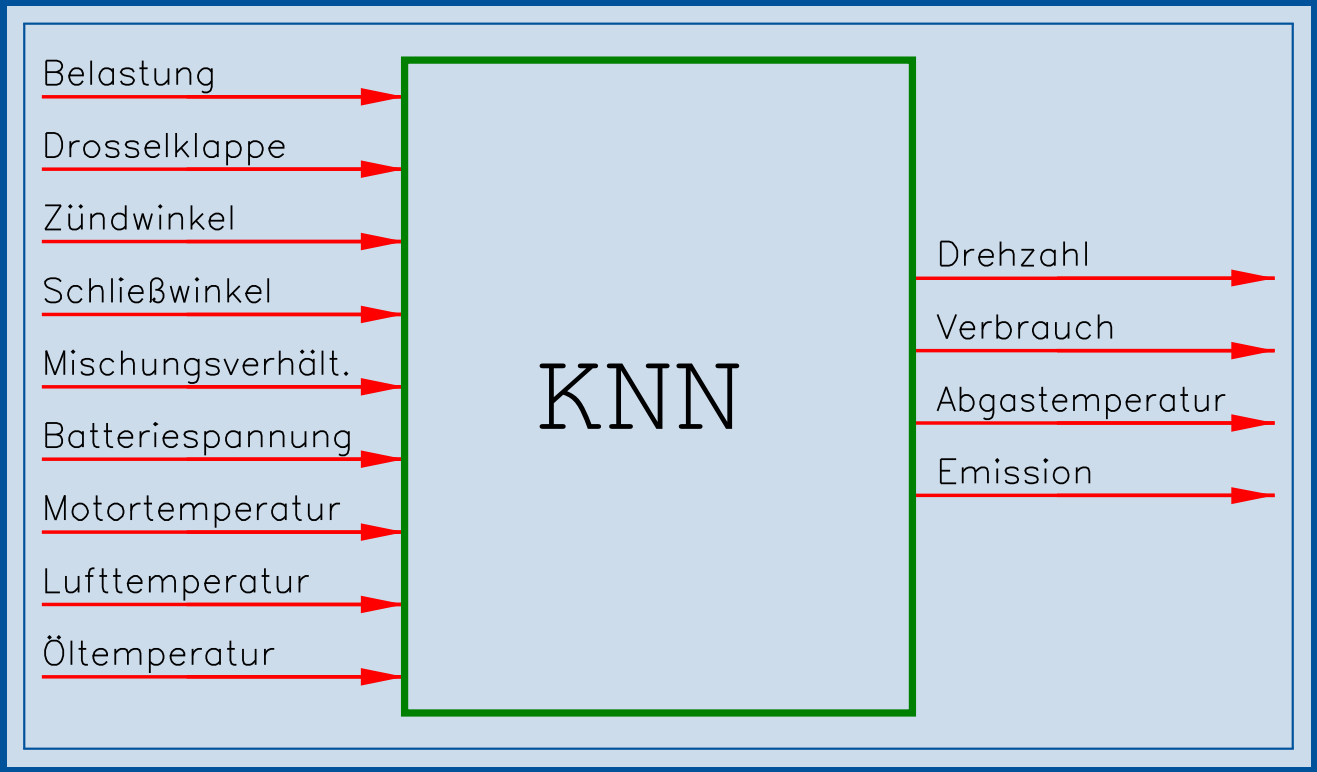
Modellbildung am Beispiel eines Motors



Modellbildung: mathematische Abstraktion

- Motor als Funktion auffassen
- Setzt funktionale Abhängigkeit voraus

Modellbildung am Beispiel eines Motors

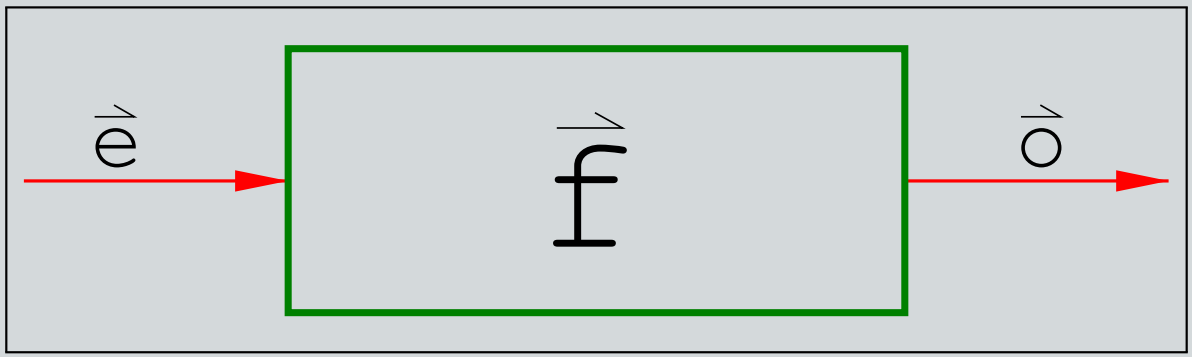


Modellbildung mit neuronalen Netzen

- Künstliches Neuronales Netz soll Motorverhalten lernen
- Setzt Beispieldaten zum Training voraus

Physikalisches Wirkverhalten: mathematische Abstraktion

Funktionale Abhängigkeit: **Alle Einflussparameter sind bekannt.**

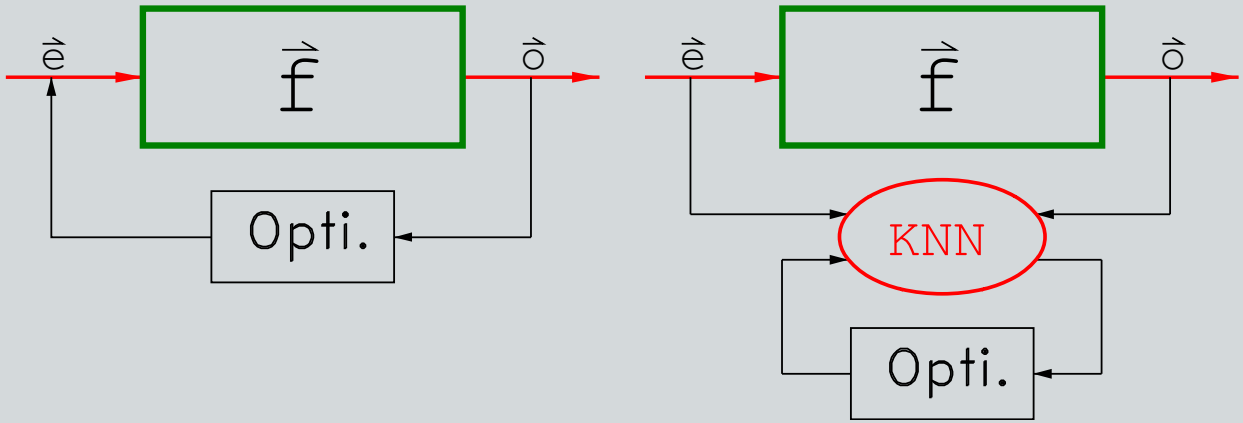


Der Ausgang $\vec{o}$ hängt funktional vom Eingang $\vec{e}$ ab. Dieser Zusammenhang wird durch eine Funktion f beschrieben:

$$\vec{o} = \vec{f}(\vec{e}) \; .$$

Anwendung: Reduktion der Prüfstandläufe

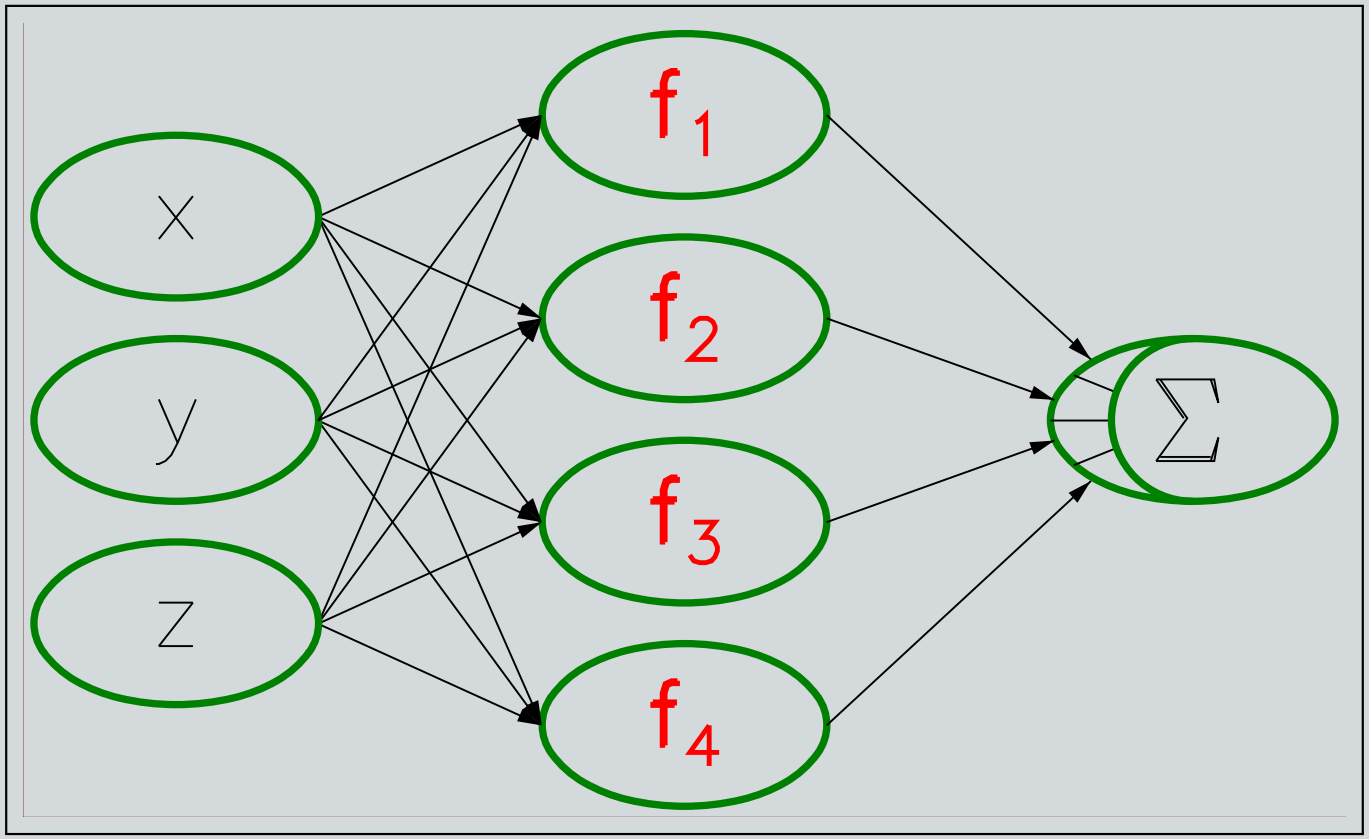
Optimieren der Motorkennlinien mit neuronalen Netzen



- Übernahme der Funktionsweise mehrerer Motoren in neuronales Netz: Training mit Daten von Prüfstandläufen
- Erweiterung des neuronalen Netzes bei neuen Motoren mit wenigen Prüfstandläufen
- Optimieren der Kennlinien mit dem neuronalen Netz

R. Stricker, BMW AG, 1996

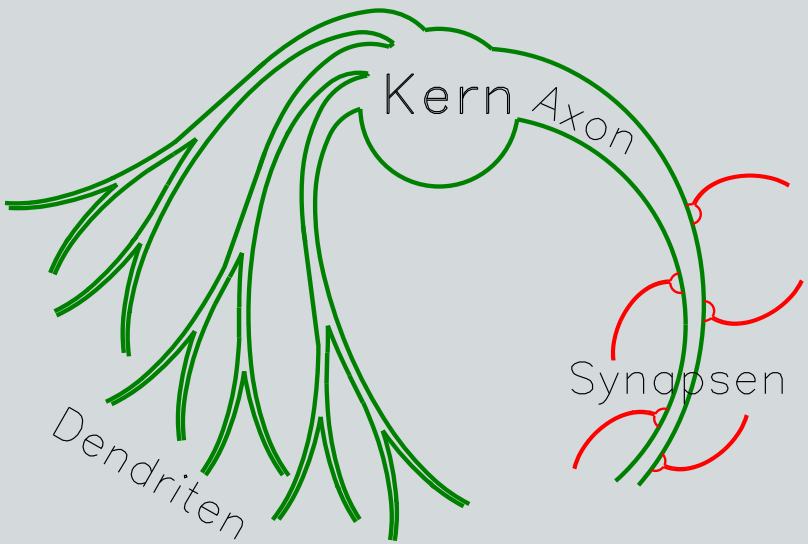
Lineare Ausgleichsrechnung als Vergleich



Lernen nur in der Ausgabeschicht

Das biologische Neuron

Arbeitsweise des Neurons

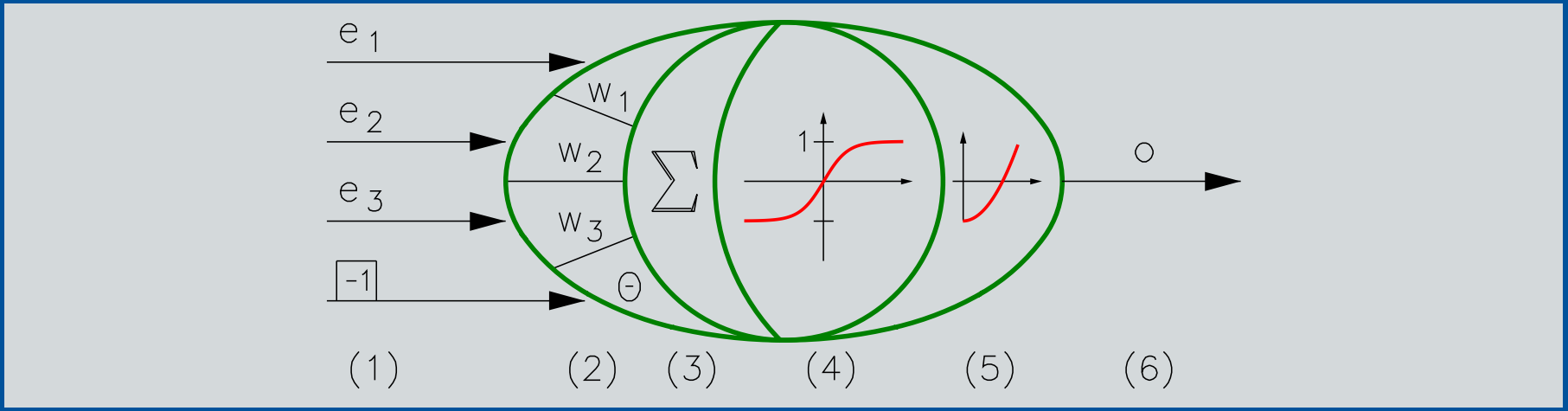


1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf.
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls.
5. Gesamtimpuls: Erregung des Neurons.
6. Schwellenwert erreicht: Neuron feuert.

Lernen:

Synapsen, Dendriten verstärken Verbindung.

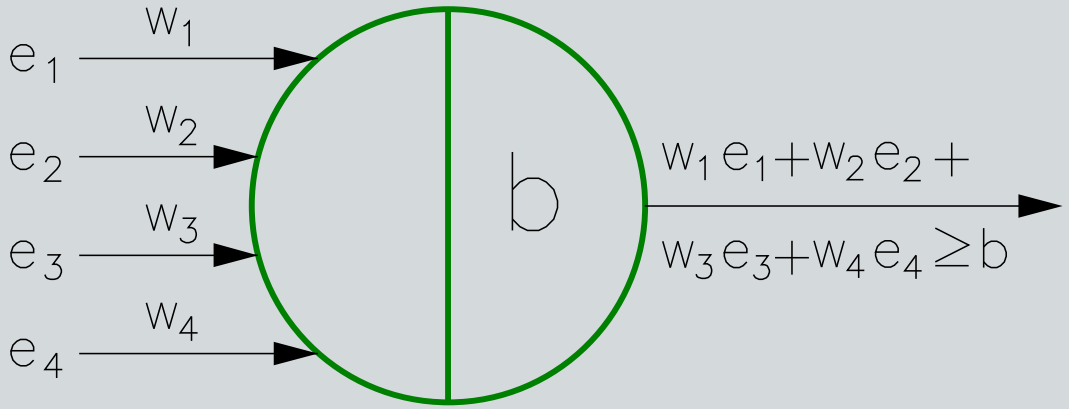
Das Rechnerneuron



1. Eingabe(vektor) $\vec{e} = (e_1, \ldots, e_n)$, -1 für Schwellenwert
2. Gewichte und Schwellenwert $\vec{w} = (w_1, \ldots, w_n)$ und θ
3. Nettowert (Propagierung) $net = \langle \vec{e}, \vec{w} \rangle - \theta = \sum_{i=1}^{n} e_i w_i - \theta$
4. Aktivierungsfunktion, Aktivität $a = a(\langle \vec{e}, \vec{w} \rangle - \theta)$
5. Ausgabefunktion
6. Ausgabe $o = o(a(\langle \vec{w}, \vec{e} \rangle - \theta))$

Gewichtetes Schwellenwertelement

Definition



$$0 \leq \left(\sum_{i=1}^{n} w_i e_i \right) - b = \langle \vec{w}, \vec{e} \rangle - b$$

Lineare Trennung

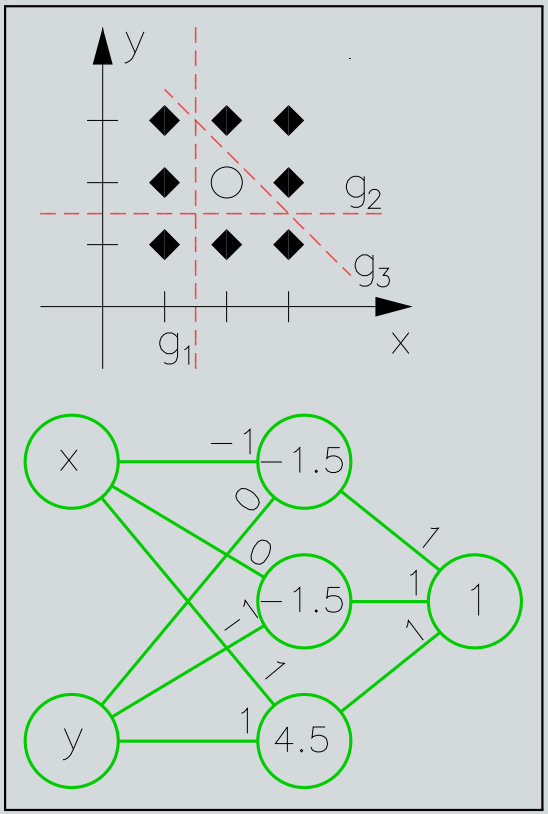
Verdeckte Neuronen trennen linear

$$\begin{pmatrix} x \\ y \end{pmatrix}\begin{pmatrix} -1 \\ 0 \end{pmatrix} \geq \begin{pmatrix} 1.5 \\ 0 \end{pmatrix}\begin{pmatrix} -1 \\ 0 \end{pmatrix} = -1.5$$

$$\begin{pmatrix} x \\ y \end{pmatrix}\begin{pmatrix} 0 \\ -1 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 1.5 \end{pmatrix}\begin{pmatrix} 0 \\ -1 \end{pmatrix} = -1.5$$

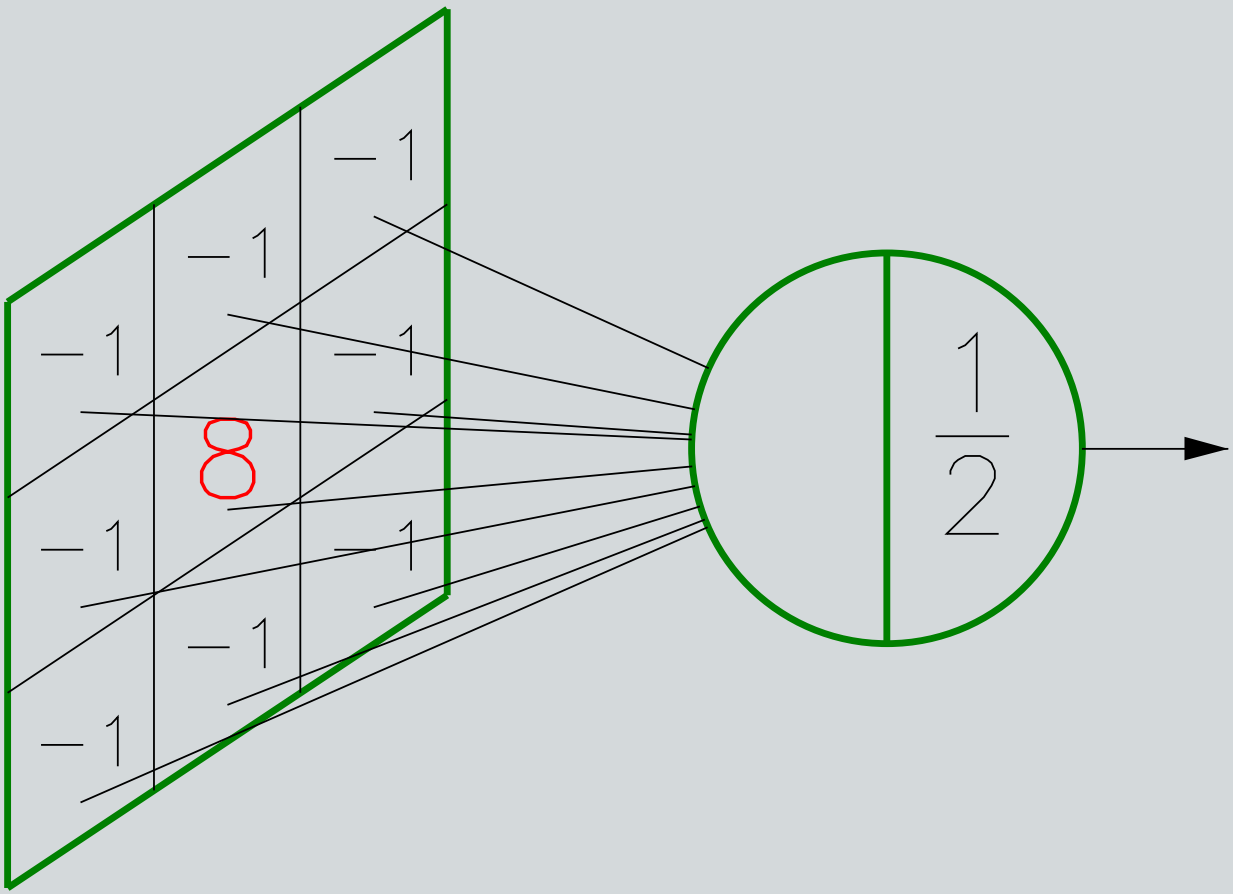
$$\begin{pmatrix} x \\ y \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \geq \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 4.5$$

Das Ausgabeneuron fasst diese Ergebnisse mit der logischen ODER-Funktion zusammen. Eine positive Netzantwort ($o = 1$) zeigt also die Zugehörigkeit zur äußeren Menge an, diese wird als positiv bezeichnet.



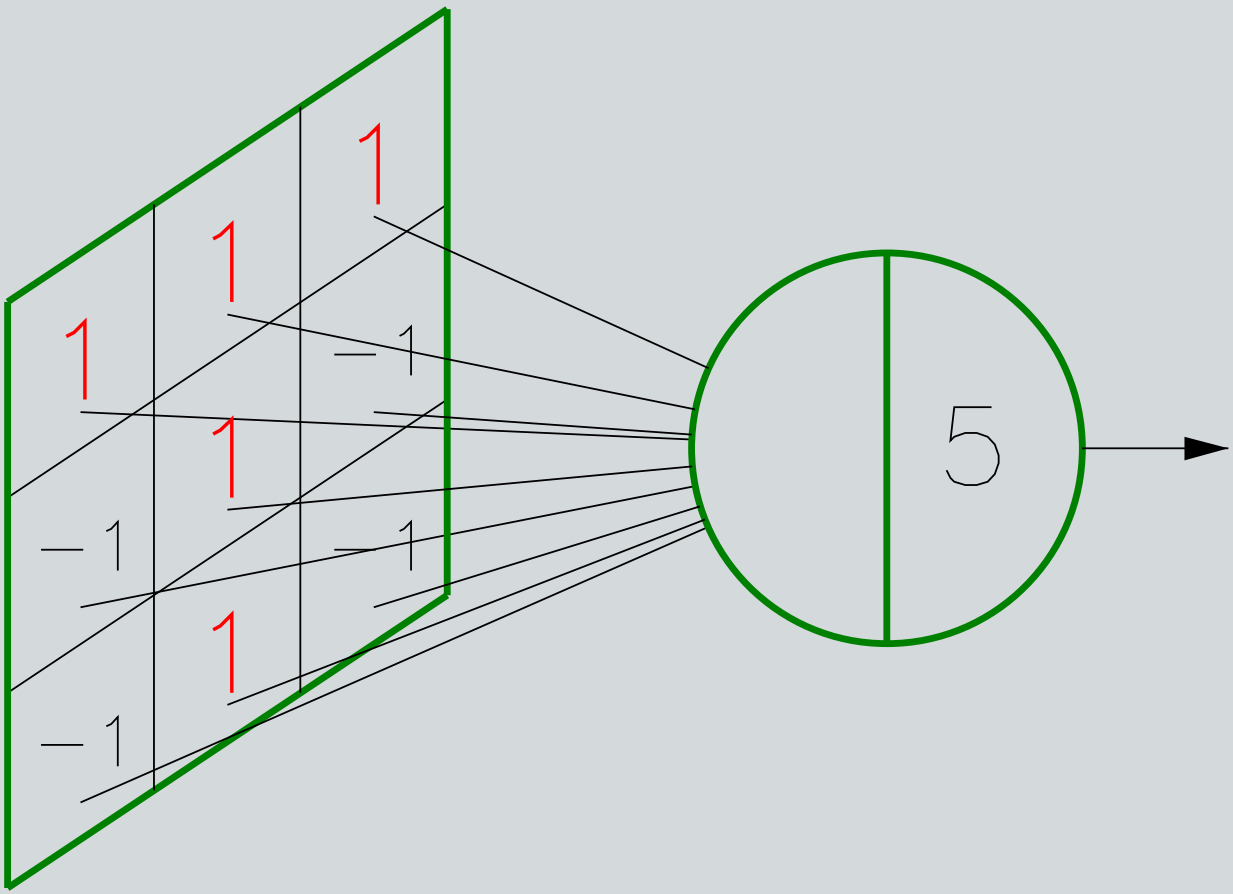
Rand- und Mustererkennung

Matrix of sensors to recognize boundaries



Rand- und Mustererkennung

Matrix of sensors to recognize symbols



Logische Funktionen, Pseudoinverse

Beispiel (UND ODER XOR)

$$Y = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix};$$

$$Z = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Logische Funktionen, Pseudoinverse

Beispiel (UND ODER XOR: $Z = 0.5 \leq W \cdot Y$)

$$Y = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix}; \qquad Z = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$YY^T = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & -2 \\ 1 & 2 & -2 \\ -2 & -2 & 4 \end{pmatrix}$$

$$W = ZY^T(YY^T)^{-1} = \begin{pmatrix} 0.5 & 0.5 & 0.25 \\ 0.5 & 0.5 & -0.25 \\ 0 & 0 & -0.5 \end{pmatrix}$$

Logische Funktionen, Pseudoinverse

Beispiel (UND ODER XOR: $Z = 0.5 \leq W \cdot Y$)

$$Y = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix}; \qquad Z = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$YY^T = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & -2 \\ 1 & 2 & -2 \\ -2 & -2 & 4 \end{pmatrix}$$

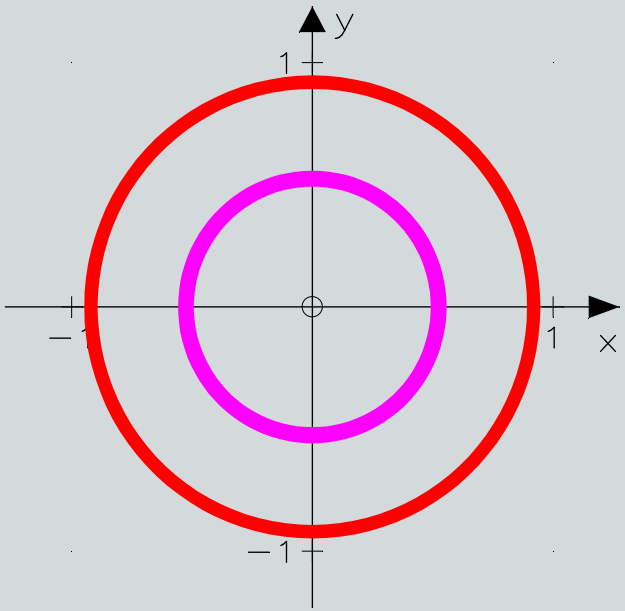
$$W = ZY^T(YY^T)^{-1} = \begin{pmatrix} 0.5 & 0.5 & 0.25 \\ 0.5 & 0.5 & -0.25 \\ 0 & 0 & -0.5 \end{pmatrix}$$

$$N = (\vec{n}_1, \ldots, \vec{n}_4) = W \cdot Y = \begin{pmatrix} -0.25 & 0.25 & 0.25 & 0.75 \\ 0.25 & 0.75 & 0.75 & 1.25 \\ 0.50 & 0.50 & 0.50 & 0.50 \end{pmatrix}$$

Support Vector Machnine

Nicht linear trennbar:

Zwei ineinander liegende Ringe können nicht linear getrennt werden.



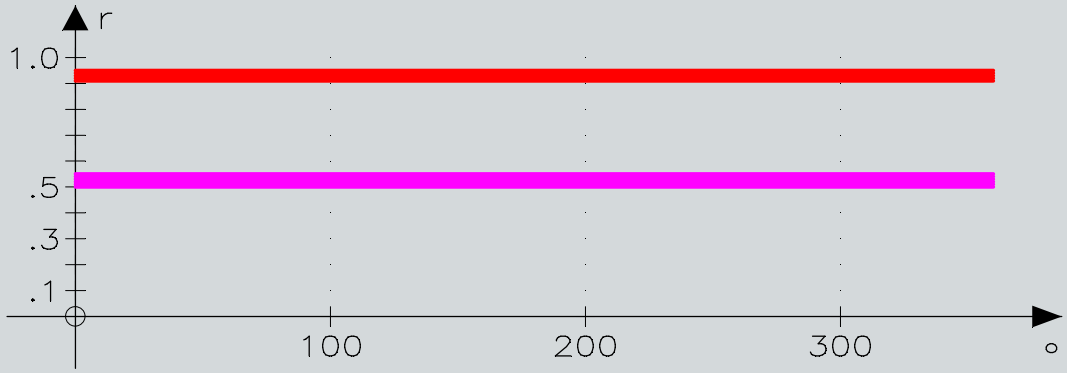
1. Darstellung in Polarkoordinaten
2. Höherdimensionale Darstellung, z. B. auf Gaußglocke

Support Vector Machnine

Nicht linear trennbar: Transformation als Ausweg

Zwei ineinander liegende Ringe können nicht linear getrennt werden.

1. Darstellung in Polarkoordinaten



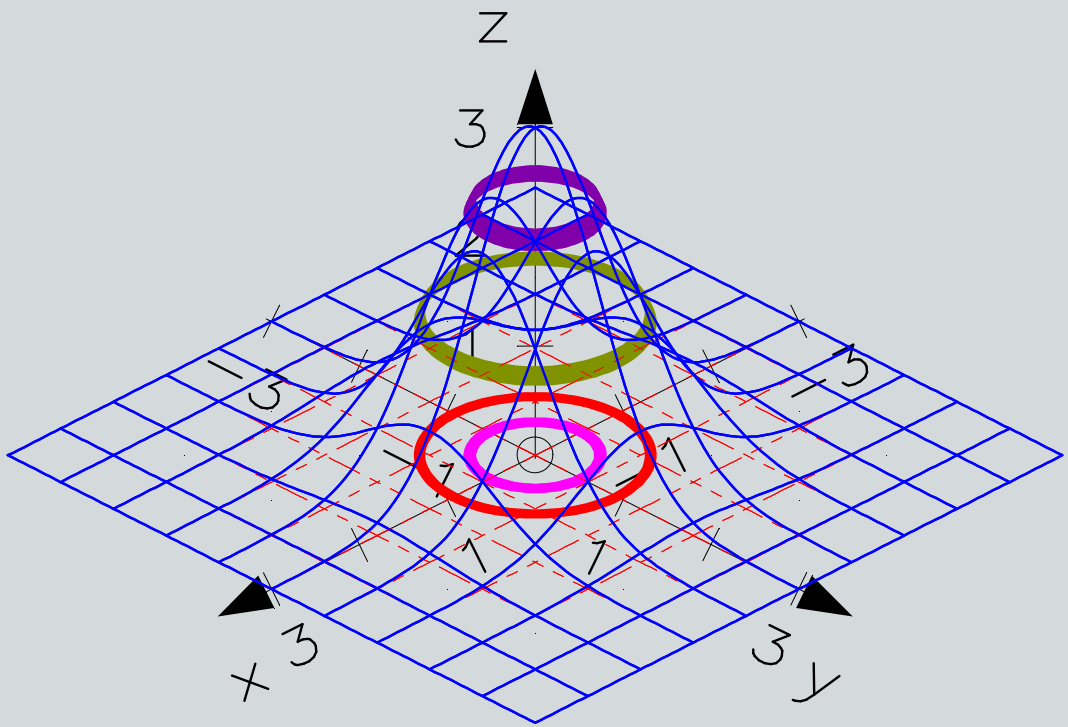
2. Höherdimensionale Darstellung, z. B. auf Gaußglocke

Support Vector Machnine

Nicht linear trennbar: Transformation als Ausweg

Zwei ineinander liegende Ringe können nicht linear getrennt werden.

1. Darstellung in Polarkoordinaten
2. Höherdimensionale Darstellung, z. B. auf Gaußglocke



Support Vector Machnine

Nicht linear trennbar: Transformation als Ausweg

Zwei ineinander liegende Ringe können nicht linear getrennt werden.

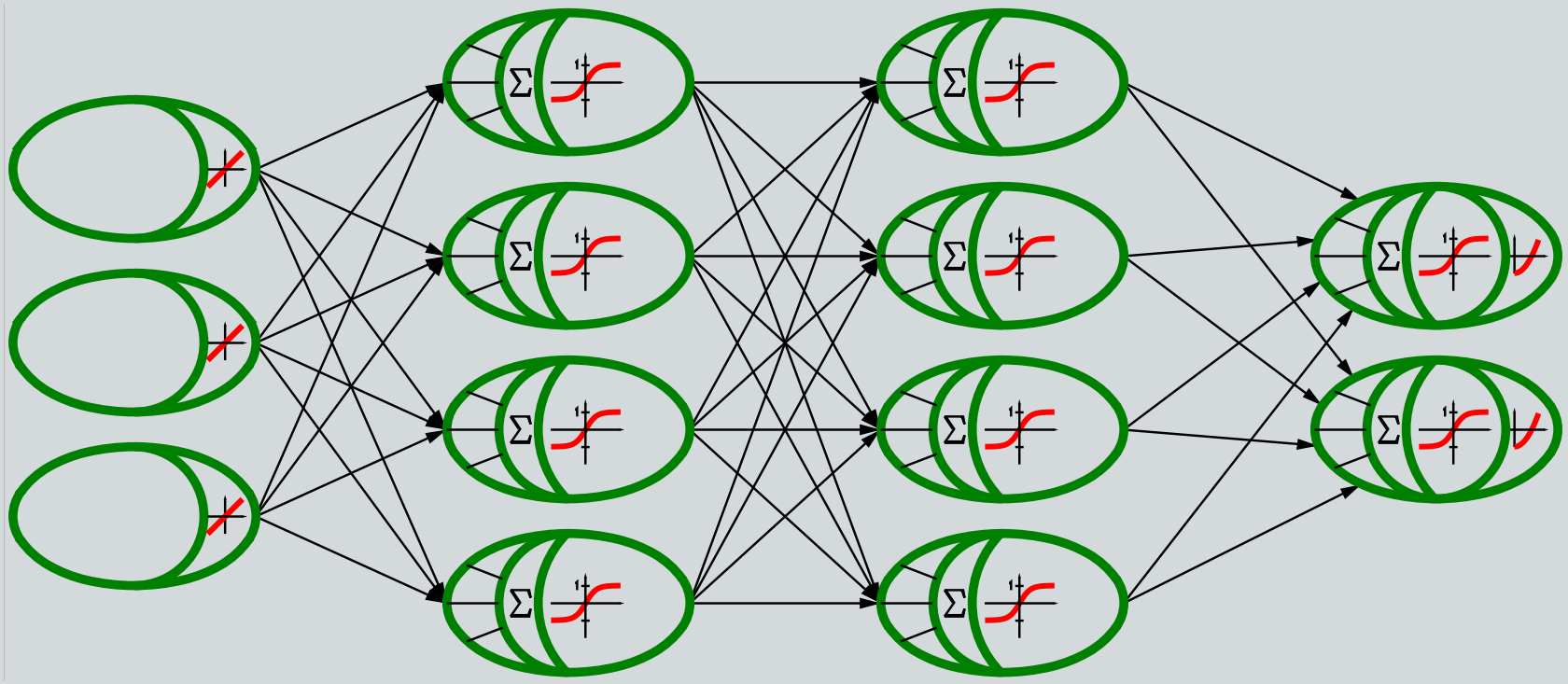
1. Darstellung in Polarkoordinaten
2. Höherdimensionale Darstellung, z. B. auf Gaußglocke

Support Vector Machines

Geeignete Transformationen suchen (lassen), die lineare Trennung ermöglichen.

Schichtenmodell

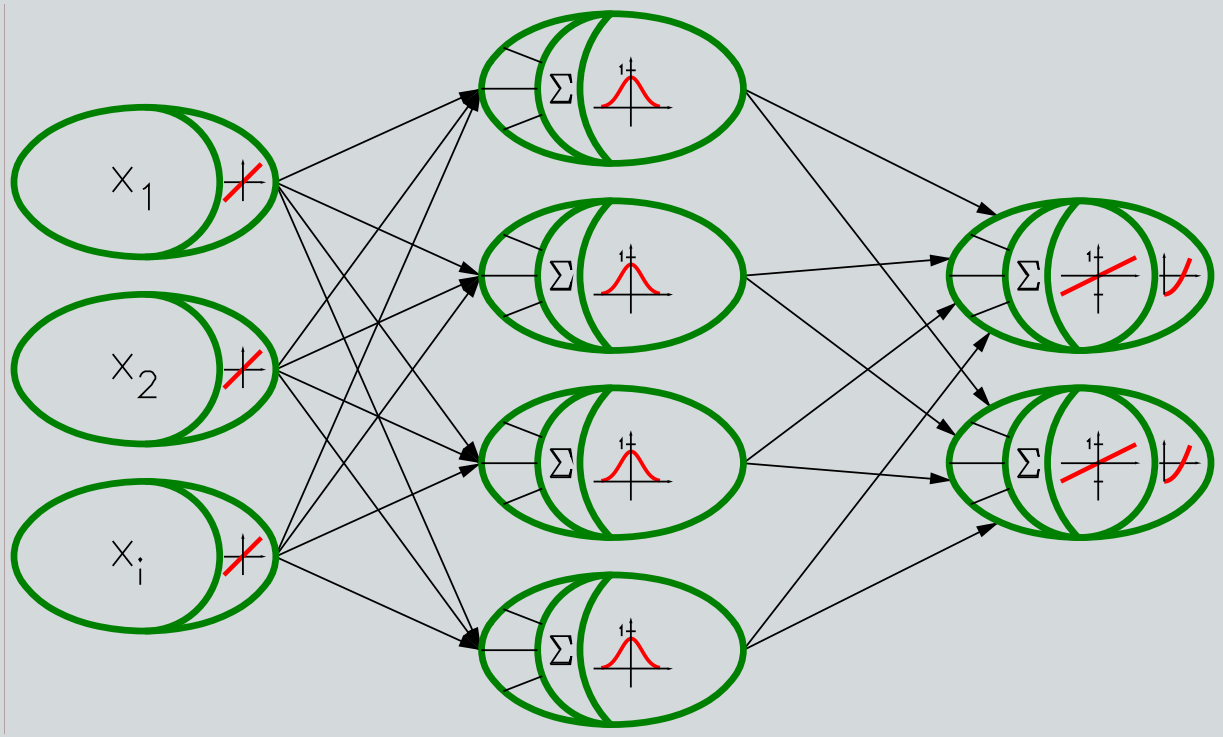
Schichtenmodell mit Topologie 3-4-4-2



Lernen: Gewichte und Schwellenwerte ändern bis Ausgabe stimmt.

Schichtenmodell

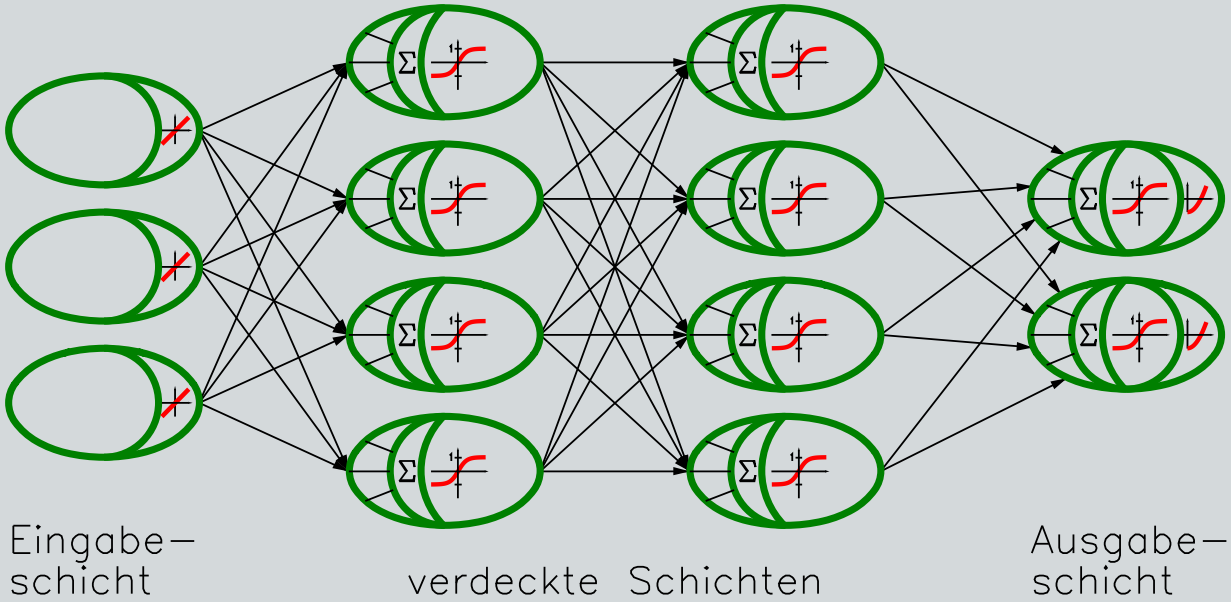
Schichtenmodell mit Topologie 3-4-4-2



Lernen: Gewichte und Schwellenwerte ändern bis Ausgabe stimmt.

Schichtenmodell

Schichtenmodell mit Topologie 3-4-4-2




```
r←s Bpforw ein;anzs;aus;is;net
anzs←↑⍴bpan◇aus←net←bpan⍴¨0◇aus[1]←⊂⍉ein⊥¨⊂⌽bpte          ⍝ Eing. trans.
is←1
DO4:→(anzs<is←is+1)/UNDO4          ⍝ Schleife über Schichten: Ausgaben
aus[is]←⊂1÷1+*-bpap×⊃net[is]←⊂(is⊃bpbi)+[1](r⊃bpgw)+.×(r←is-1)⊃aus◇→DO4
UNDO4:r←⍉(anzs⊃aus)⊥¨⊂⌽bpta
```

Topologie des Schichtenmodells

Satz (Kolmogoroff, 1957)

Jede vektorwertige Funktion $f : [0, 1]^n \to \mathbb{R}^m$ *kann durch ein dreischichtiges neuronales Netz abgebildet werden. Hierzu werden* n *Eingabeneuronen,* $2n + 1$ *verdeckte Neuronen und* m *Ausgabeneuronen benötigt. Die Aktivierungsfunktionen hängen von* f *und* n *ab.*

Bemerkung

1. *Der Beweis ist ein nicht-konstruktiver Existenzbeweis.*
2. *Er enthält keine Angaben über die Aktivierungsfunktion.*
3. *Der Satz hat keine praktische Bedeutung.*

Topologie des Schichtenmodells

Satz (Kolmogoroff, 1957)

Jede vektorwertige Funktion $f : [0, 1]^n \to \mathbb{R}^m$ *kann durch ein dreischichtiges neuronales Netz abgebildet werden. Hierzu werden* n *Eingabeneuronen,* $2n + 1$ *verdeckte Neuronen und* m *Ausgabeneuronen benötigt. Die Aktivierungsfunktionen hängen von* f *und* n *ab.*

Zusatz

Für die stetige Funktion $f : [-1, 1]^n \to [-1, 1]$ *gibt es Funktionen* g *und* g_i $(i = 1, \ldots, 2n + 1)$ *in einem Argument und Konstanten* λ_j $(j = 1, \ldots, n)$ *mit*

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{2n+1} g\left(\sum_{j=1}^{n} \lambda_j g_i(x_j)\right) .$$

Topologie des Schichtenmodells

Satz (Kolmogoroff, 1957)

Jede vektorwertige Funktion $f : [0, 1]^n \to \mathbb{R}^m$ *kann durch ein dreischichtiges neuronales Netz abgebildet werden. Hierzu werden* n *Eingabeneuronen,* $2n + 1$ *verdeckte Neuronen und* m *Ausgabeneuronen benötigt. Die Aktivierungsfunktionen hängen von* f *und* n *ab.*

Satz (Annäherung durch Netze)

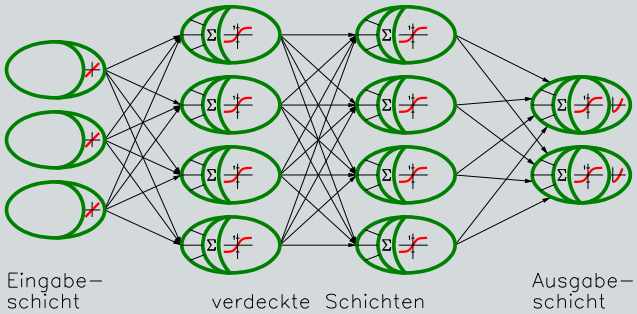
Jede Funktion kann durch Netze mit einer verdeckten Schicht angenähert werden.

Schichtenmodell

Eingabeschicht

- analoge Eingabe:
  Lineare Transformation auf [-1; 1]
- diskrete Eingabe:
  Je Wert ein Neuron, Eingabe -1, 1

Schichtenmodell

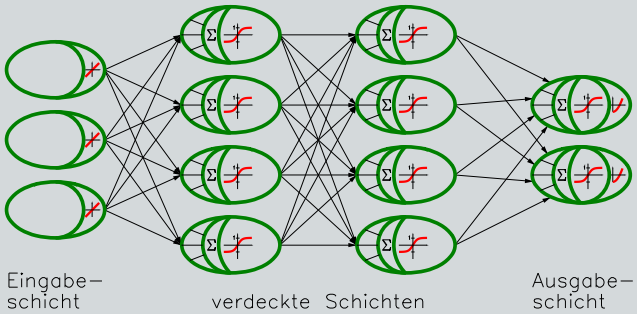


Schichtenmodell

Eingabeschicht

- analoge Eingabe:
  Lineare Transformation auf [-1; 1]
- diskrete Eingabe:
  Je Wert ein Neuron, Eingabe -1, 1

Schichtenmodell



Ausgabeschicht mit tangentiell sigmoider Aktivierungsfunktion

- Zielaktivitäten gleichverteilt im Intervall $[-0.6, 0.6]$!
- Umkehrung der Ausgabefunktion kann sein:

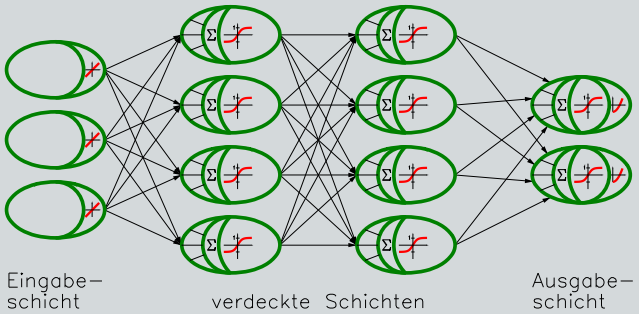
$$f(x) = \left\{ \begin{array}{ccc} [m,\ M] & \rightarrow & [-0.6, 0.6] \\ x & \mapsto & -0.6 + 1.2 \left(\frac{x-m}{M-m} \right)^s \ ; \quad s > 0 \end{array} \right\}$$

Schichtenmodell

Eingabeschicht

- analoge Eingabe:
  Lineare Transformation auf [-1; 1]
- diskrete Eingabe:
  Je Wert ein Neuron, Eingabe -1, 1

Schichtenmodell



Ausgabeschicht mit logarithmisch sigmoider Aktivierungsfunktion

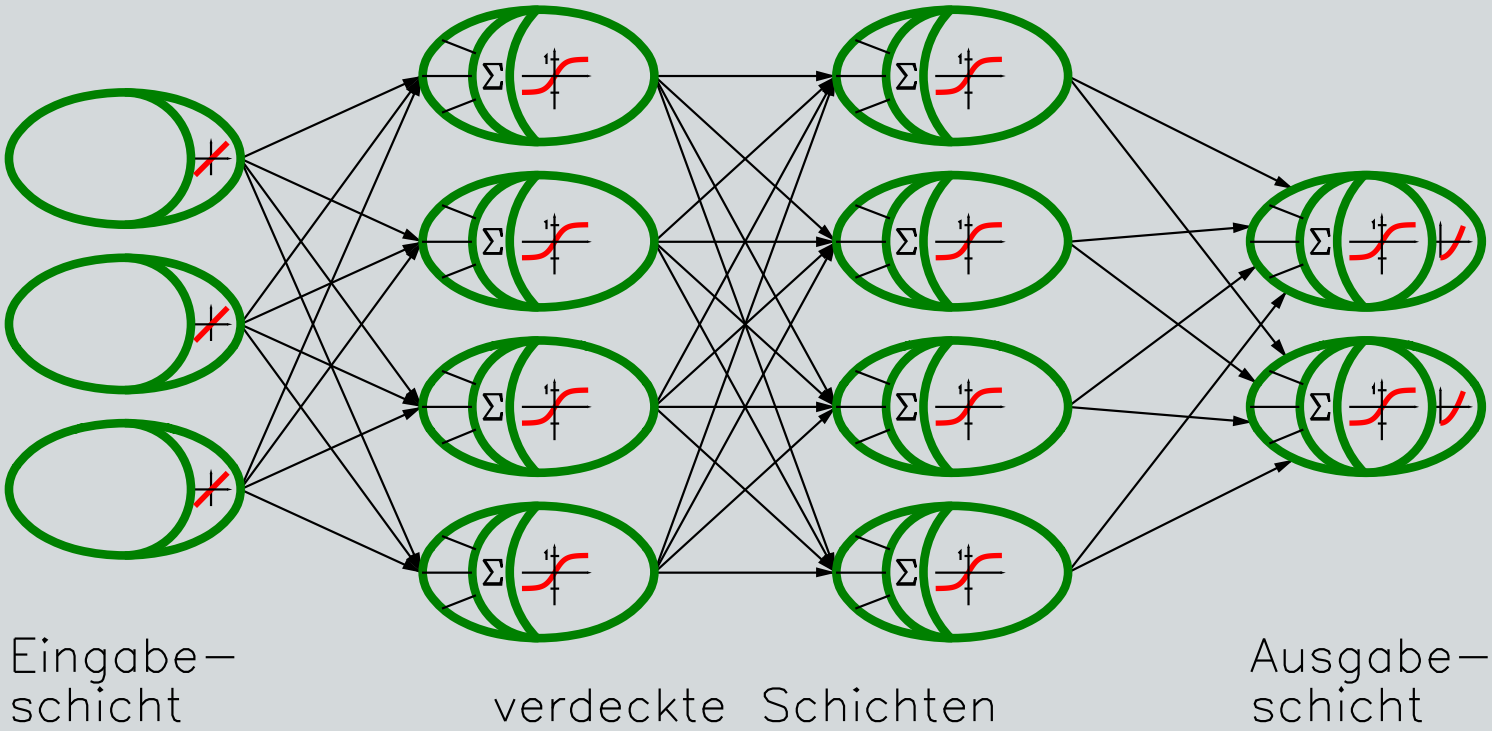
- Zielaktivitäten gleichverteilt im Intervall [0.2, 0.8]!
- Umkehrung der Ausgabefunktion kann sein:

$$f(x) = \left\{ \begin{array}{ccc} [m,\ M] & \rightarrow & [0.2,\ 0.8] \\ x & \mapsto & 0.2 + 0.6 \left(\frac{x-m}{M-m} \right)^s \ ; \quad s > 0 \end{array} \right\}$$

Lernen in vorwärtsgerichteten Schichtennetzen

Ziel

Gewichte so ändern, dass die Abweichungen in den Lerndaten klein werden.



Lernen in vorwärtsgerichteten Schichtennetzen

Ziel

Gewichte so ändern, dass die Abweichungen in den Lerndaten klein werden.

Berechnung

Fehler: $E(\vec{w}) = \frac{1}{2} \sum_{i=1}^{n} \|\vec{z}_i - \vec{o}_i(w)\|^2$

Gadient: $\overrightarrow{\text{grad}}_w E(\vec{w}) = \left(\frac{\partial E(\vec{w})}{\partial w_1}, \frac{\partial E(\vec{w})}{\partial w_2}, \ldots, \frac{\partial E(\vec{w})}{\partial w_3} \right)$

Lernen in vorwärtsgerichteten Schichtennetzen

Ziel

Gewichte so ändern, dass die Abweichungen in den Lerndaten klein werden.

Berechnung

Fehler: $E(\vec{w}) = \frac{1}{2} \sum_{i=1}^{n} \|\vec{z}_i - \vec{o}_i(w)\|^2$

Gadient: $\overrightarrow{\text{grad}}_w E(\vec{w}) = \left(\frac{\partial E(\vec{w})}{\partial w_1}, \frac{\partial E(\vec{w})}{\partial w_2}, \ldots, \frac{\partial E(\vec{w})}{\partial w_3} \right)$

Delta-Regel (Gradient descent)

$\Delta\vec{w}^{(t)} = -\sigma \overrightarrow{\text{grad}}_w E(\vec{w}); \qquad \vec{w}^{(t)} = \vec{w}^{(t-1)} + \Delta\vec{w}^{(t)} + \mu\Delta\vec{w}^{(t-1)}$

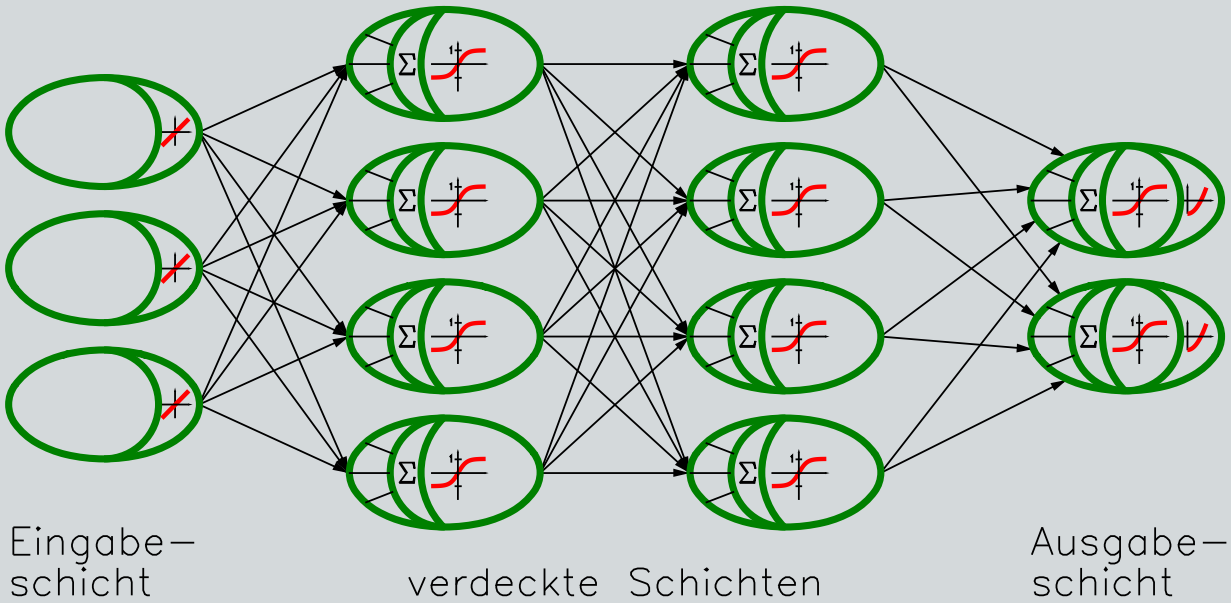
σ fallend, z.B. von 0.9 auf 0.1, μ steigend, z.B. $\mu = 1 - \sigma$.

Fehlerrückverfolgung

Fehler schrittweise zurück rechnen mit relativem Nettofehler δ_i:

$$\vec{\delta}_i := \frac{\partial E}{\partial \vec{n}_i} = \frac{\partial E}{\partial \vec{n}_{i+1}} \cdot \frac{\partial \vec{n}_{i+1}}{\partial \vec{o}_i} \cdot \frac{\partial \vec{o}_i}{\partial \vec{n}_i} = \vec{\delta}_{i+1} \cdot W_{i+1} \cdot A(\vec{n}_i)$$

$$\frac{\partial E}{\partial W_{i,rs}} = \frac{\partial E}{\partial \vec{n}_i} \cdot \frac{\partial \vec{n}_i}{\partial W_{i,rs}} = \vec{\delta}_i \cdot o_{i-1,s} \hat{e}_r = \delta_{i,r} o_{i-1,s}$$



Fehlerrückverfolgung

Fehler schrittweise zurück rechnen mit relativem Nettofehler δ_i:

$$\vec{\delta}_i := \frac{\partial E}{\partial \vec{n}_i} = \frac{\partial E}{\partial \vec{n}_{i+1}} \cdot \frac{\partial \vec{n}_{i+1}}{\partial \vec{o}_i} \cdot \frac{\partial \vec{o}_i}{\partial \vec{n}_i} = \vec{\delta}_{i+1} \cdot W_{i+1} \cdot A(\vec{n}_i)$$

$$\frac{\partial E}{\partial W_{i,rs}} = \frac{\partial E}{\partial \vec{n}_i} \cdot \frac{\partial \vec{n}_i}{\partial W_{i,rs}} = \vec{\delta}_i \cdot o_{i-1,s} \hat{e}_r = \delta_{i,r} o_{i-1,s}$$

```
r←ziel Bpback aus;anzs;dgwa;err;is;lr
(aus lr)←aus
err←bpan⍴¨0
dgwa←dgw                                          ⍝ dgw global
is←anzs←↑⍴bpan                                    ⍝ Anzahl Schichten
r←anzs⊃aus                                        ⍝ Fehler letzte Schicht
err[anzs]←⊂-2×bpap×(r×1-r)×ziel-r                 ⍝ Nettofehler
DO:→(1>is←is-1)/UNDO                              ⍝ B.P. über alle Sch.
dgw[is]←⊂(-(1+is)⊃err)∘.×r←is⊃aus                  ⍝ ΔGewicht je Schicht
⍎(is>1)/'err[is]←⊂bpap×(r×1-r)×(⍉is⊃bpgw)+.×⊃err[1+is]'   ⍝ Nettofehler
→DO
UNDO:bpgw←bpgw+lr×dgw+(1-lr)×dgwa                 ⍝ Gewichte ändern
bpbi←bpbi+(dbi←-lr×err)+(1-lr)×dbi                ⍝ Bias ändern
r←anzs⊃err                                        ⍝ Fehler in letzter Sch.
```

Lernen in vorwärtsgerichteten Schichtennetzen

Lernverfahren nach Levenberg-Marquardt

$$E(\vec{w}) = \frac{1}{2}\left\langle \vec{f}(\vec{w}), \vec{f}(\vec{w}) \right\rangle \qquad \text{mit} \qquad \vec{f}(\vec{w}) = \vec{z} - \vec{o}(\vec{w})$$

Lernen in vorwärtsgerichteten Schichtennetzen

Lernverfahren nach Levenberg-Marquardt

$$E(\vec{w}) = \frac{1}{2}\left\langle \vec{f}(\vec{w}), \vec{f}(\vec{w}) \right\rangle \quad \text{mit} \quad \vec{f}(\vec{w}) = \vec{z} - \vec{o}(\vec{w})$$

$$\vec{0} = E'(\vec{w}) = \overrightarrow{\operatorname{grad}} E(\vec{w}) = f'^{T}(\vec{w})\vec{f}(\vec{w})$$

Lernen in vorwärtsgerichteten Schichtennetzen

Lernverfahren nach Levenberg-Marquardt

$$
\begin{aligned}
E(\vec{w}) &= \frac{1}{2}\left\langle \vec{f}(\vec{w}), \vec{f}(\vec{w}) \right\rangle \qquad \text{mit} \qquad \vec{f}(\vec{w}) = \vec{z} - \vec{o}(\vec{w}) \\
\vec{0} &= E'(\vec{w}) = \overrightarrow{\operatorname{grad}} E(\vec{w}) = f'^{T}(\vec{w})\vec{f}(\vec{w}) \\
E''(\vec{w}) &= \left(f'^{T}(\vec{w})\vec{f}(\vec{w})\right)' = f''^{T}(\vec{w})\vec{f}(\vec{w}) + f'^{T}(\vec{w})f'(\vec{w}) \\
&= f'^{T}(\vec{w})f'(\vec{w}) \qquad \text{für} \qquad f''^{T}(\vec{w}) \qquad \text{klein!}
\end{aligned}
$$

Lernen in vorwärtsgerichteten Schichtennetzen

Lernverfahren nach Levenberg-Marquardt

$$
\begin{aligned}
E(\vec{w}) &= \frac{1}{2}\left\langle \vec{f}(\vec{w}), \vec{f}(\vec{w}) \right\rangle \qquad \text{mit} \qquad \vec{f}(\vec{w}) = \vec{z} - \vec{o}(\vec{w}) \\
\vec{0} &= E'(\vec{w}) = \overrightarrow{\operatorname{grad}} E(\vec{w}) = f'^{T}(\vec{w})\vec{f}(\vec{w}) \\
E''(\vec{w}) &= \left(f'^{T}(\vec{w})\vec{f}(\vec{w})\right)' = f''^{T}(\vec{w})\vec{f}(\vec{w}) + f'^{T}(\vec{w})f'(\vec{w}) \\
&\approx f'^{T}(\vec{w})f'(\vec{w}) \qquad \text{für} \qquad f''^{T}(\vec{w}) \qquad \text{klein!} \\
\vec{w}_{k+1} &= \vec{w}_k - E''(\vec{w})^{-1}E'(\vec{w}) \\
\Delta\vec{w} &= -\left(f'^{T}(\vec{w})f'(\vec{w})\right)^{-1} f'^{T}(\vec{w})\vec{f}(\vec{w})
\end{aligned}
$$

Lernen in vorwärtsgerichteten Schichtennetzen

Lernverfahren nach Levenberg-Marquardt

$$
\begin{aligned}
E(\vec{w}) &= \frac{1}{2}\left\langle \vec{f}(\vec{w}), \vec{f}(\vec{w}) \right\rangle \qquad \text{mit} \qquad \vec{f}(\vec{w}) = \vec{z} - \vec{o}(\vec{w}) \\
\vec{0} &= E'(\vec{w}) = \overrightarrow{\operatorname{grad}}\, E(\vec{w}) = f'^{T}(\vec{w})\vec{f}(\vec{w}) \\
E''(\vec{w}) &= f'^{T}(\vec{w})f'(\vec{w}) \qquad \text{für} \qquad f''^{T}(\vec{w}) \qquad \text{klein!} \\
\vec{w}_{k+1} &= \vec{w}_k - E''(\vec{w})^{-1}E'(\vec{w}) \\
\Delta\vec{w} &= -\left(f'^{T}(\vec{w})f'(\vec{w})\right)^{-1} f'^{T}(\vec{w})\vec{f}(\vec{w})
\end{aligned}
$$

Zu lösendes lineares Gleichungssystem:

$$f'^{T}(\vec{w})f'(\vec{w})\Delta\vec{w} = -f'^{T}(\vec{w})\vec{f}(\vec{w})$$

Beurteilung eines trainierten Netzes

Fehler in den Lerndaten

- maximaler Fehler
- durchschnittlicher Fehler
- Standardabweichung

Beurteilung eines trainierten Netzes

Fehler in den Lerndaten

- maximaler Fehler
- durchschnittlicher Fehler
- Standardabweichung

Fehler in den Kontrolldaten

- 20%-40% der Lerndaten
- maximaler und durchschnittlicher Fehler
- Standardabweichung

Beurteilung eines trainierten Netzes

Fehler in den Lerndaten

- maximaler Fehler
- durchschnittlicher Fehler
- Standardabweichung

Fehler in den Kontrolldaten

- 20%-40% der Lerndaten
- maximaler und durchschnittlicher Fehler
- Standardabweichung

Systemspezialist

- Beurteilung von Trendaussagen

Beurteilung eines trainierten Netzes

Fehler in den Lerndaten

- maximaler Fehler
- durchschnittlicher Fehler
- Standardabweichung

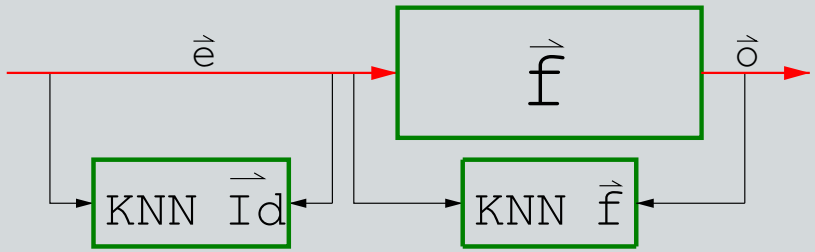
Systemspezialist

- Beurteilung von Trendaussagen

Fehler in den Kontrolldaten

- 20%-40% der Lerndaten
- maximaler und durchschnittlicher Fehler
- Standardabweichung

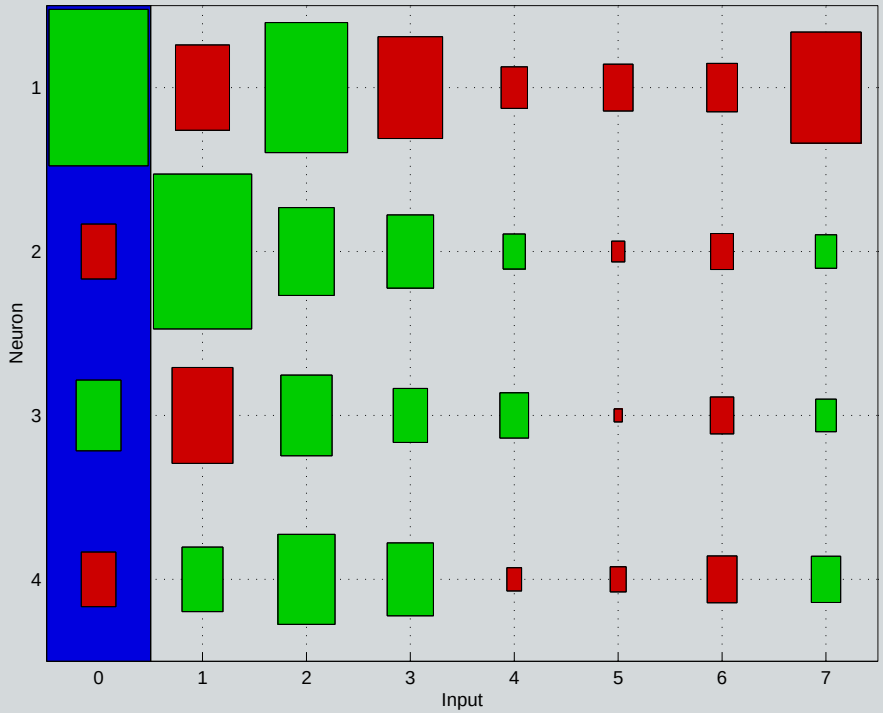
Autokorrelation



Verbessern des Lernens

Reduktion der Eingabeparameter

1. Gewichte der ersten Schicht

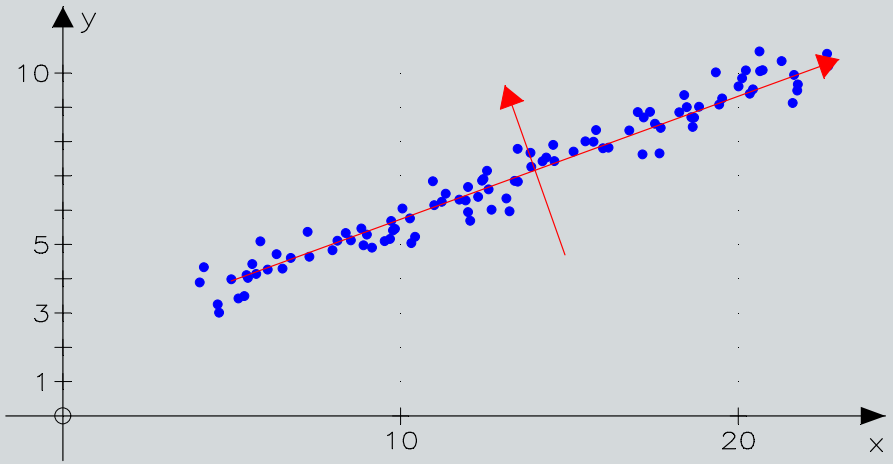


2. Hauptachsentransformation der Kovarianzmatrix
3. Dimensionsanalyse (physikalische Gleichungen)

Verbessern des Lernens

Reduktion der Eingabeparameter

1. Gewichte der ersten Schicht
2. Hauptachsentransformation der Kovarianzmatrix



3. Dimensionsanalyse (physikalische Gleichungen)

Verbessern des Lernens

Reduktion der Eingabeparameter

1. Gewichte der ersten Schicht
2. Hauptachsentransformation der Kovarianzmatrix
3. Dimensionsanalyse (physikalische Gleichungen)

$$u = \frac{5}{384}\frac{ql^4}{EI}$$

$$q[FL^{-1}],\ I[L],\ E[FL^{-2}],\ I[L^4] \quad \text{und} \quad u[L]$$

Verbessern des Lernens

Reduktion der Eingabeparameter

1. Gewichte der ersten Schicht
2. Hauptachsentransformation der Kovarianzmatrix
3. Dimensionsanalyse (physikalische Gleichungen)

$$u = \frac{5}{384}\frac{ql^4}{EI}$$

$$\pi_1 = \frac{q}{EI}; \qquad \pi_2 = \frac{l^4}{I}; \qquad \pi_3 = \frac{u}{l}$$

Verbessern des Lernens

Reduktion der Eingabeparameter

1. Gewichte der ersten Schicht
2. Hauptachsentransformation der Kovarianzmatrix
3. Dimensionsanalyse (physikalische Gleichungen)

$$u = \frac{5}{384}\frac{ql^4}{EI}$$

$$\pi_1 = \frac{q}{EI}; \qquad \pi_2 = \frac{l^4}{I}; \qquad \pi_3 = \frac{u}{I} = \frac{5}{384} \cdot \pi_1\pi_2$$

Verbessern des Lernens

Lernqualität Verbessern

1. Lernrate absteigend, Momentum angepasst

$$\begin{aligned} \Delta W_i^{(t)} &:= -\sigma \overrightarrow{\text{grad}}_{W_i} E = -\sigma(\vec{o}_{i-1} \cdot \vec{\delta}_i)^T \\ W_i^{(t)} &= W^{(t-1)} - \sigma \overrightarrow{\text{grad}}_{W_i} E + \mu \Delta W_i^{(t-1)} \end{aligned}$$

2. Lerndaten – Testdaten
3. Neuronen vorübergehend löschen (Drop out)

Verbessern des Lernens

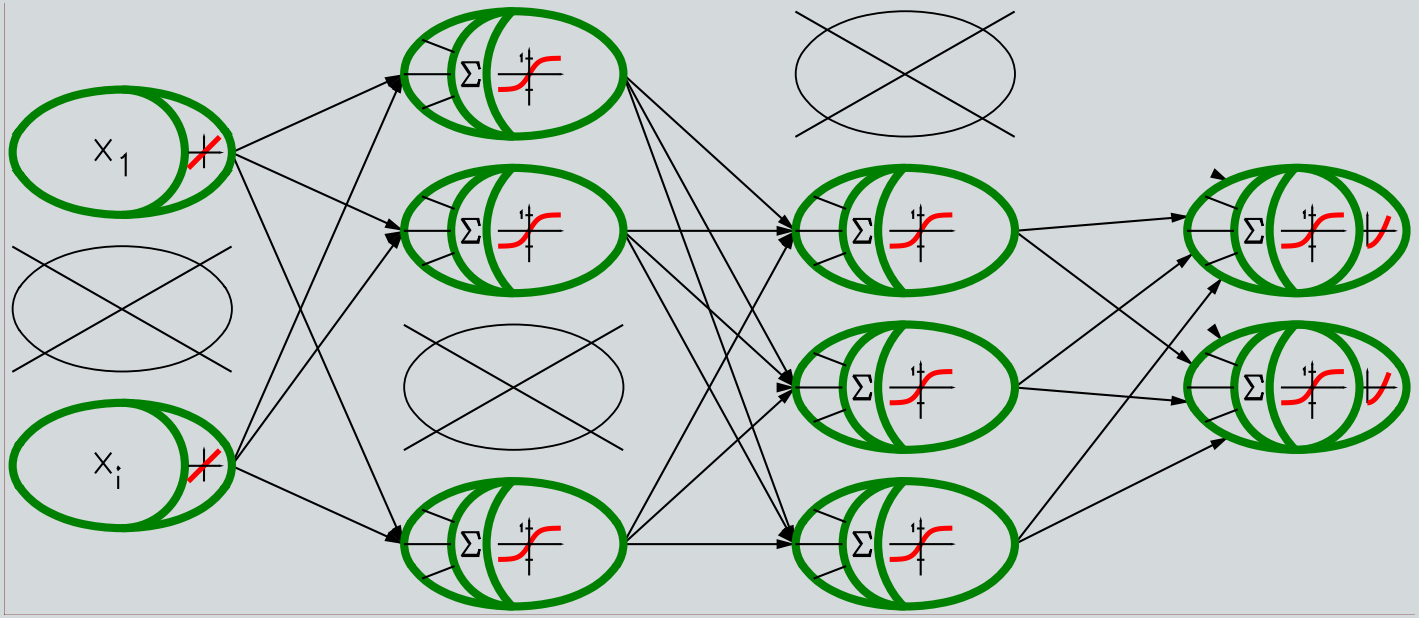
Lernqualität Verbessern

1. Lernrate absteigend, Momentum angepasst
2. Lerndaten – Testdaten : Lernen stoppen, wenn Fehler in den Testdaten größer werden:
   1. Bessere ausgleichende Wirkung
   2. Kein Übertrainieren (Overlearning)
3. Neuronen vorübergehend löschen (Drop out)

Verbessern des Lernens

Lernqualität Verbessern

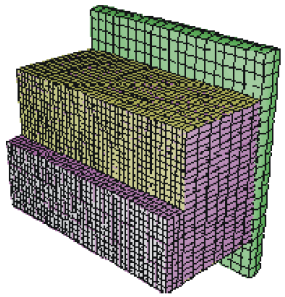
1. Lernrate absteigend, Momentum angepasst
2. Lerndaten – Testdaten
3. Neuronen vorübergehend löschen (Drop out)



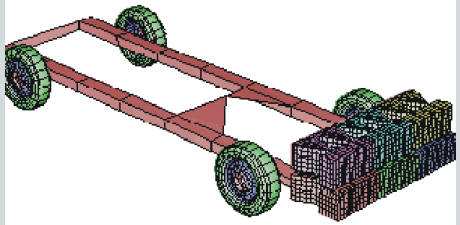
Vorhersage der Insassenbelastung

Industriesemester 1995 (mit M. Holzner, R. Stricker)

Barrieren



Frontalaufprall



Seitenaufprall

Frontalaufprall

Barrieren aus Schaum oder Alu-Waben
(Fa. Fritzmeier)

Vorhersage der Insassenbelastung

Industriesemester 1995 (mit M. Holzner, R. Stricker)

Untersuchung der Einsatzfähigkeit neuronaler Netze und Fuzzy Logic in der Crash-Berechnung

- Versuchsreihe Front-Crash 0°, 100% Überlappung, E36
- Ziel: Vorhersage der Insassenbelastung nach konstruktiven Änderungen
- Probleme:
  - Zusammenhänge „unbekannt“: keine Regeln
  - wenig Daten (ca. 90 Datensätze)

Vorhersage der Insassenbelastung

Industriesemester 1995 (mit M. Holzner, R. Stricker)

Aufgaben

- Festlegung geeigneter Eingabeparameter mit Datenbereichen:
  - Fahrzeugklassifikation: Typ (Türen), Zylinder, Getriebe, Lenksäule verstellbar?, Modelljahr
  - Airbag: Modell(jahr), Ausströmöffnung, Zündzeitpunkt, E-Modul, Volumen, Sprengstoffmasse
  - Versuchsdaten: Testort, -geschwindigkeit, Masse, Dummytyp
  - Versuchsergebnisse am Fahrzeug: Fahrzeugdeformation, Lenksäulenverschiebung
- Ausgabedaten:
- Netzbewertung:

Vorhersage der Insassenbelastung

Industriesemester 1995 (mit M. Holzner, R. Stricker)

Aufgaben

- Festlegung geeigneter Eingabeparameter mit Datenbereichen:
  - Fahrzeugklassifikation: Typ (Türen), Zylinder, Getriebe, Lenksäule verstellbar?, Modelljahr
  - Airbag: Modell(jahr), Ausströmöffnung, Zündzeitpunkt, E-Modul, Volumen, Sprengstoffmasse
  - Versuchsdaten: Testort, -geschwindigkeit, Masse, Dummytyp
  - Versuchsergebnisse am Fahrzeug: Fahrzeugdeformation, Lenksäulenverschiebung
- Ausgabedaten:
- Netzbewertung:

Vorhersage der Insassenbelastung

Industriesemester 1995 (mit M. Holzner, R. Stricker)

Aufgaben

- Festlegung geeigneter Eingabeparameter mit Datenbereichen:
  - Fahrzeugklassifikation: Typ (Türen), Zylinder, Getriebe, Lenksäule verstellbar?, Modelljahr
  - Airbag: Modell(jahr), Ausströmöffnung, Zündzeitpunkt, E-Modul, Volumen, Sprengstoffmasse
  - Versuchsdaten: Testort, -geschwindigkeit, Masse, Dummytyp
  - Versuchsergebnisse am Fahrzeug: Fahrzeugdeformation, Lenksäulenverschiebung
- Ausgabedaten:
  - Fahrerdummy, Beifahrerdummy: HIC, Kopf-, Brustbeschleunigung
- Netzbewertung:

Vorhersage der Insassenbelastung

Industriesemester 1995 (mit M. Holzner, R. Stricker)

Aufgaben

- Festlegung geeigneter Eingabeparameter mit Datenbereichen:
- Ausgabedaten:
  - Fahrerdummy, Beifahrerdummy: HIC, Kopf-, Brustbeschleunigung
- Netzbewertung:
  - 80% Lern- und 20% Testdaten:
    Vergleich der Standardabweichungen
  - Autokorrelation
  - Trendaussagen

Vorhersage der Insassenbelastung

Industriesemester 1995 (mit M. Holzner, R. Stricker)

Aufgaben

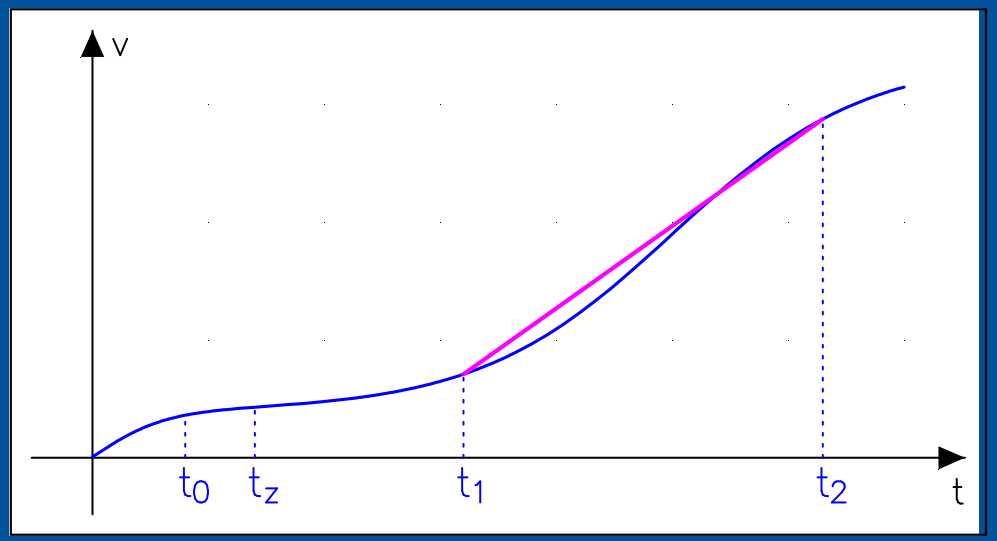
- Festlegung geeigneter Eingabeparameter mit Datenbereichen:
- Ausgabedaten:
  - Fahrerdummy, Beifahrerdummy: HIC, Kopf-, Brustbeschleunigung
- Netzbewertung:
  - 80% Lern- und 20% Testdaten:
    Vergleich der Standardabweichungen
  - Autokorrelation
  - Trendaussagen

Ergebnis

Die Insassenbelastung eines Versuchs wurde vorhergesagt.

Unfallschwere

mit A. Kuhn, J. Urbahn, BMW AG, 2000



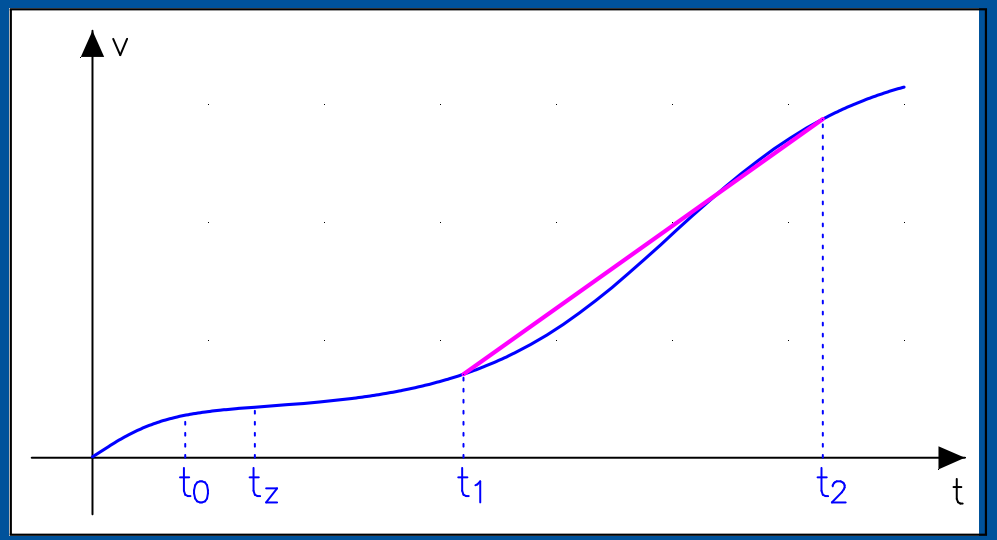
- t_0: Entscheidung
- t_Z: Zündzeitpunkt Airbag ($t_1 - t_Z \approx 30\,\mathrm{ms}$)
- t_1: Beginn kritischer Vorverlagerung
- t_2: Beschleunigung nimmt ab

Ziele

1. Schwere des Unfalls vorhersagen
2. Entscheidungshilfe für (An-)Steuerung der Rückhaltesysteme
3. Dadurch Insassen optimal schützen

Unfallschwere

mit A. Kuhn, J. Urbahn, BMW AG, 2000



- t_0: Entscheidung
- t_Z: Zündzeitpunkt Airbag ($t_1 - t_Z \approx 30\,\mathrm{ms}$)
- t_1: Beginn kritischer Vorverlagerung
- t_2: Beschleunigung nimmt ab

Ziele

1. Schwere des Unfalls vorhersagen
2. Entscheidungshilfe für (An-)Steuerung der Rückhaltesysteme
3. Dadurch Insassen optimal schützen

Projektziel

Unfallschwere-Parameter

1. (mittlere) Geschwindigkeit der Insassen (Zeitpunkt, Vorverlagerung)
2. mittlere Beschleunigung der Insassen

Grundlage

Daten aus Parametervariationen mit Monte-Carlo Methode:

1. Variation unfallrel. Parameter und Testart
2. FEM-Berechnungen mit PamCrash
3. 150 - 300 Datensätze für jede der 14 Modellreihen

Projektziel

Unfallschwere-Parameter

1. (mittlere) Geschwindigkeit der Insassen (Zeitpunkt, Vorverlagerung)
2. mittlere Beschleunigung der Insassen

Grundlage

Daten aus Parametervariationen mit Monte-Carlo Methode:

1. Variation unfallrel. Parameter und Testart
2. FEM-Berechnungen mit PamCrash
3. 150 - 300 Datensätze für jede der 14 Modellreihen

Projektziel

Unfallschwere-Parameter

1. (mittlere) Geschwindigkeit der Insassen (Zeitpunkt, Vorverlagerung)
2. mittlere Beschleunigung der Insassen

Grundlage

Daten aus Parametervariationen mit Monte-Carlo Methode:

1. Variation unfallrel. Parameter und Testart
2. FEM-Berechnungen mit PamCrash
3. 150 - 300 Datensätze für jede der 14 Modellreihen

Daten von einigen Crashtests

Projektziel

Unfallschwere-Parameter

1. (mittlere) Geschwindigkeit der Insassen (Zeitpunkt, Vorverlagerung)
2. mittlere Beschleunigung der Insassen

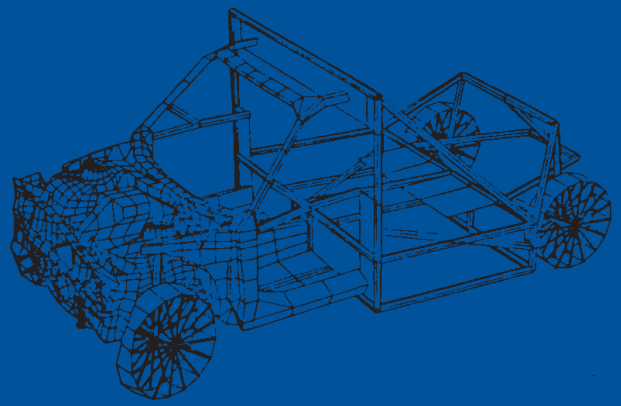
Grundlage

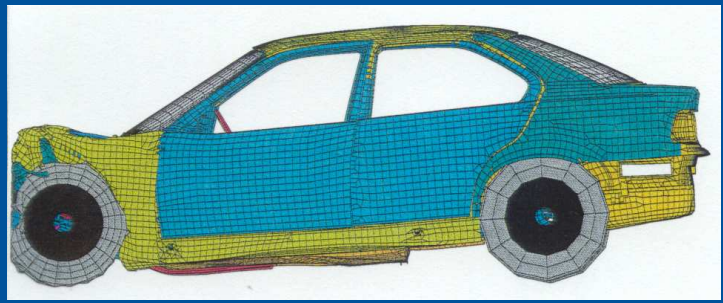
Daten aus Parametervariationen mit Monte-Carlo Methode:

1. Variation unfallrel. Parameter und Testart
2. FEM-Berechnungen mit PamCrash
3. 150 - 300 Datensätze für jede der 14 Modellreihen

Daten von einigen Crashtests

Ermöglicht durch

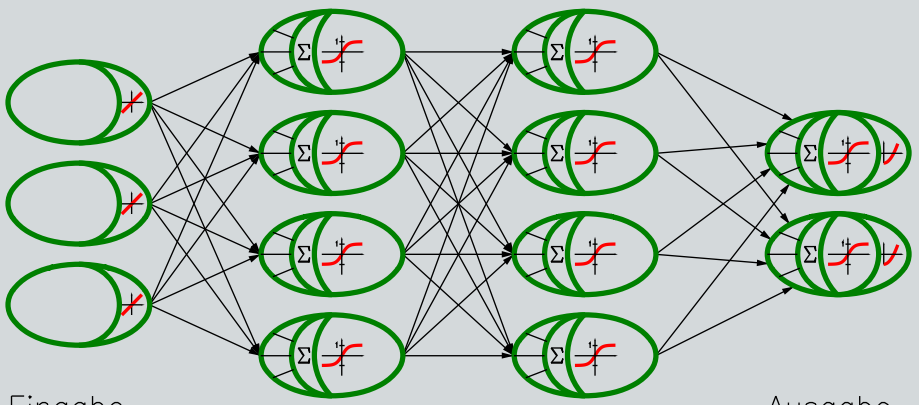




höhere Rechnerleistung!

Einsatz neuronaler Netze

3- oder 4-schichtige Netze



Eingaben

- Beschleunigungen, Geschwindigkeiten, Verschiebungen
- Maximale, gemittelte Werte

Ausgabe

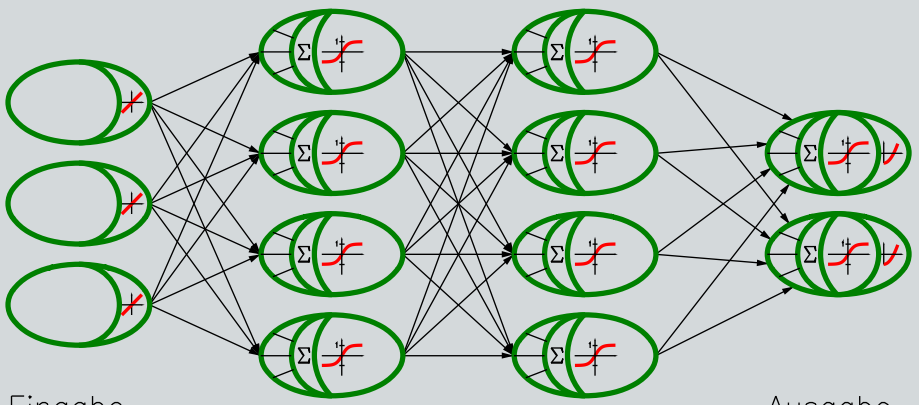
1. Geschwindigkeit
2. Gemittelte Beschleunigung (Wirkung auf Insassen)

Lernen:

- Aktivierungsfunktionen: tangential, parabelförmig
- Lernverfahren: Gradientenabstieg, Levenberg-Marquardt

Einsatz neuronaler Netze

3- oder 4-schichtige Netze



Eingaben

- Beschleunigungen, Geschwindigkeiten, Verschiebungen
- Maximale, gemittelte Werte

Ausgabe

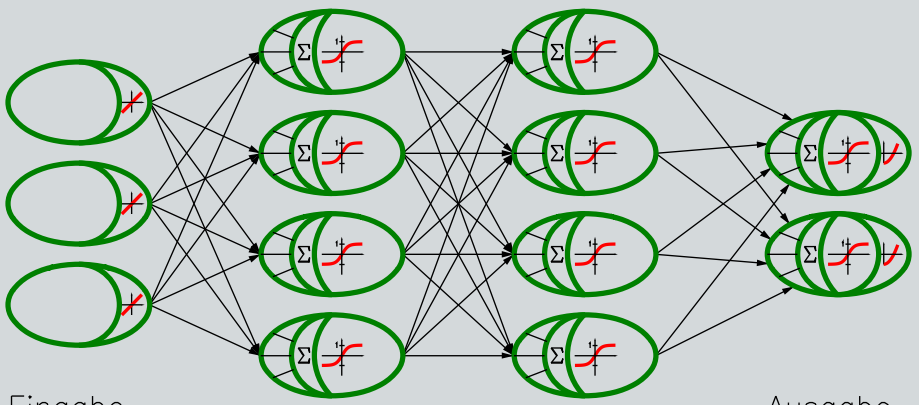
1. Geschwindigkeit
2. Gemittelte Beschleunigung (Wirkung auf Insassen)

Lernen:

- Aktivierungsfunktionen: tangential, parabelförmig
- Lernverfahren: Gradientenabstieg, Levenberg-Marquardt

Einsatz neuronaler Netze

3- oder 4-schichtige Netze



Eingaben

- Beschleunigungen, Geschwindigkeiten, Verschiebungen
- Maximale, gemittelte Werte

Ausgabe

1. Geschwindigkeit
2. Gemittelte Beschleunigung (Wirkung auf Insassen)

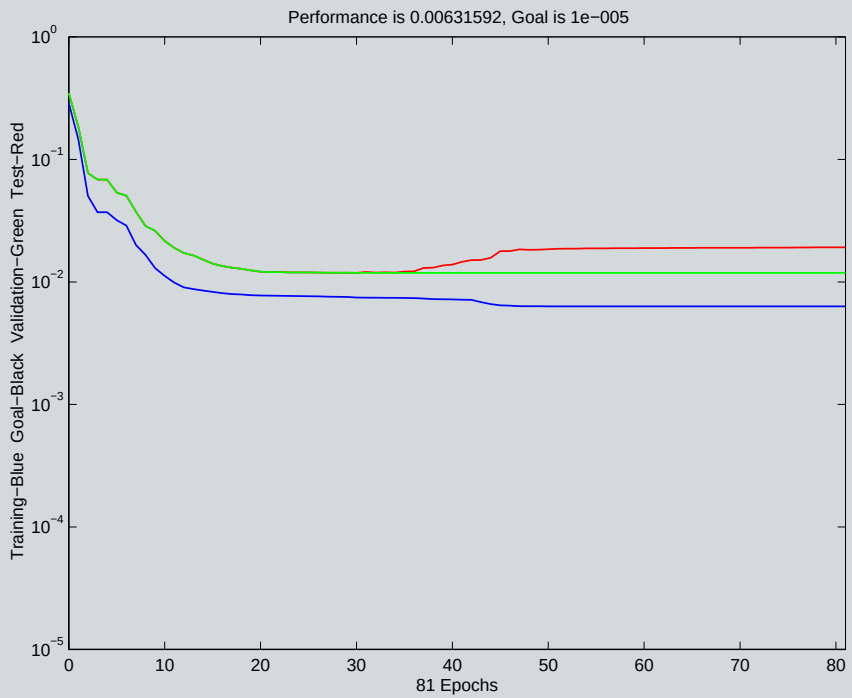
Lernen:

- Aktivierungsfunktionen: tangential, parabelförmig
- Lernverfahren: Gradientenabstieg, Levenberg-Marquardt

Training der Netze: Lernstrategie

- zufällige Wahl von 60% Lerndaten, 40% Testdaten
- Abbruch bei wachsendem Fehler in den Testdaten

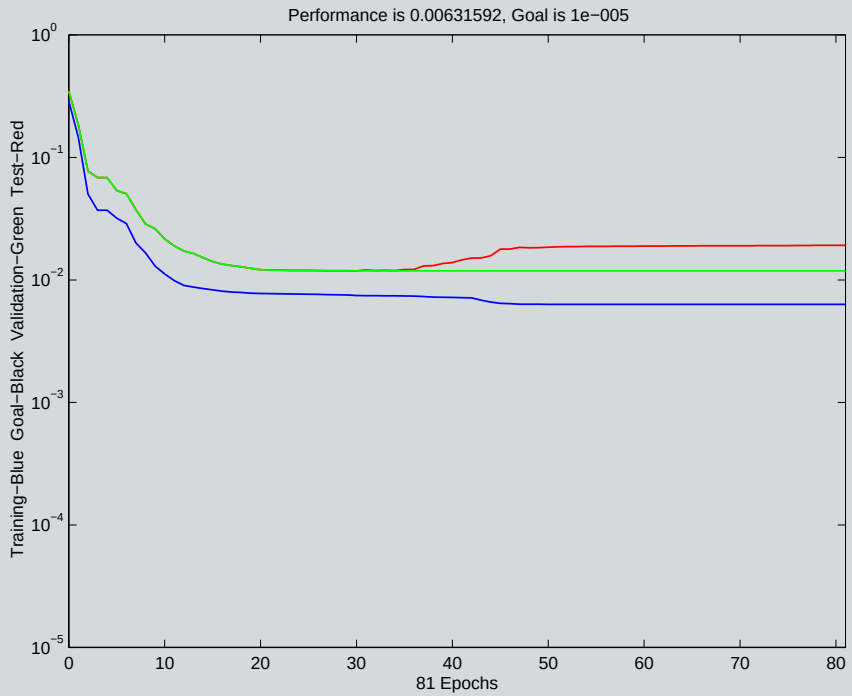
Mittlerer Fehler beim Lernen



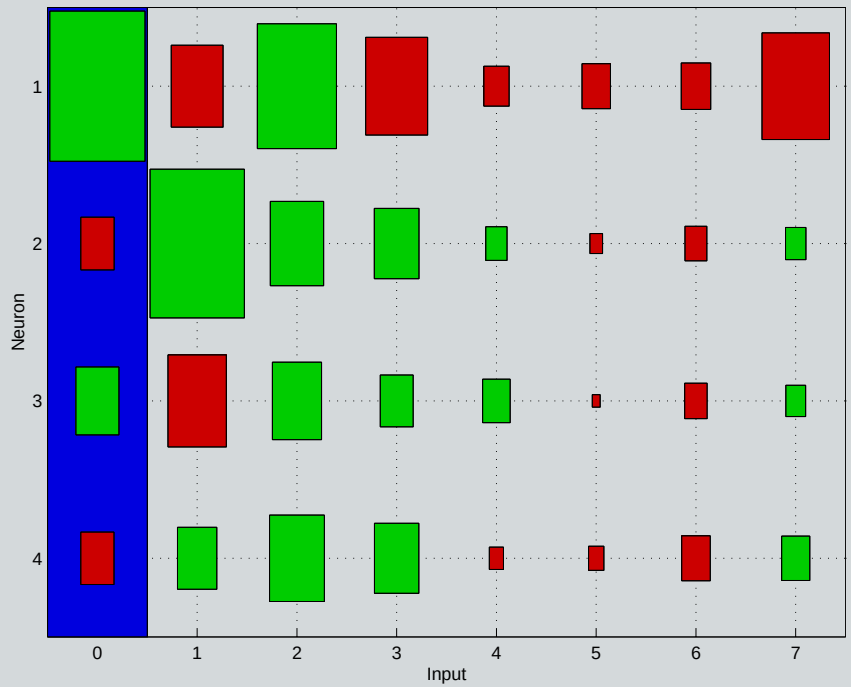
Training der Netze: Relevanz der Eingabe

- zufällige Wahl von 60% Lerndaten, 40% Testdaten
- Abbruch bei wachsendem Fehler in den Testdaten

Mittlerer Fehler beim Lernen

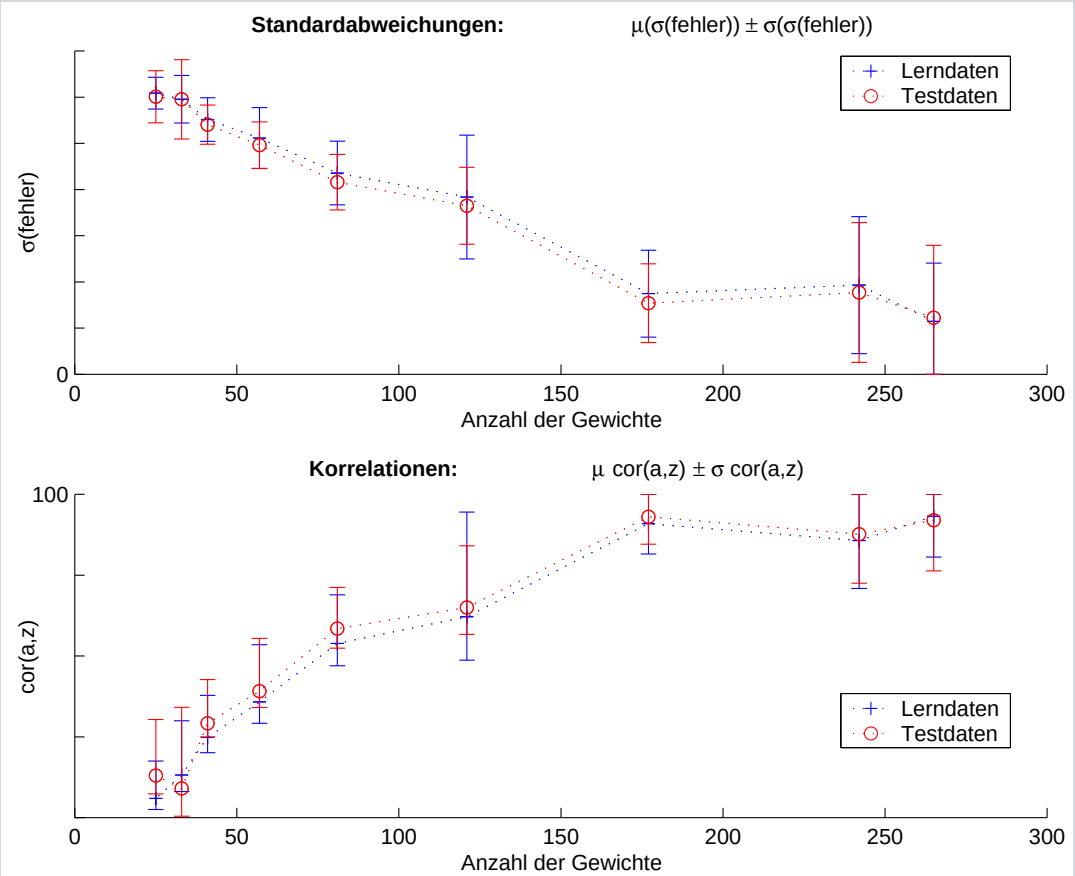


Gewichte in ersten Schicht



Optimierung

Modellstatistik über Neuronenzahl (1 - 2 verdeckten Schichten)



Grafiken:

- $\sigma(\sigma(o_i - z_i))$
- Korrelationen

Erwartung:

- $\sigma(\sigma)$ wird kleiner bis zu Sättigung.
- Lerndaten nur leicht besser als Testdaten.

Ergebnisse

Datenmodelle

1. Die Simulationen liefern eine gute Datenbasis.
2. Gute Topologien: z.B.: 4-15-8-1, 4-33-1
3. Gute Größe für Unfallschwere: mittlere Beschleunigung
4. Gute Eingangsparameter:
   gemittelte Beschleunigungen und Geschwindigkeiten

Methode σ^2 erlaubt:

- Netzgröße nach Qualitätsanforderungen auszuwählen.
- Aussagen über die Qualität der Daten.

Ergebnisse

Datenmodelle

1. Die Simulationen liefern eine gute Datenbasis.
2. Gute Topologien: z.B.: 4-15-8-1, 4-33-1
3. Gute Größe für Unfallschwere: mittlere Beschleunigung
4. Gute Eingangsparameter:
   gemittelte Beschleunigungen und Geschwindigkeiten

Methode σ^2 erlaubt:

- Netzgröße nach Qualitätsanforderungen auszuwählen.
- Aussagen über die Qualität der Daten.

Aktive Torsionstilgung

mit Ch. Hornung, G. Pflanz, BMW AG, 2005

Cabrio-Problem: Verlust an Torsionssteifigkeit M_x/d_y

Limousine: 100 %

Cabrio 7.3%

Entstehung der störenden Vibration

Anregung

Die Fahrzeuganregung wird durch Radresonanz dominiert,

$\Rightarrow$ Anregung wird über Koppelpunkte (Achse / Federbeindom) zu Komfortpunkten übertragen,

$\Rightarrow$ Vibrationen, die vom Insassen an den Komfortpunkten wahrgenommen werden.

Entstehung der störenden Vibration

Anregung

Die Fahrzeuganregung wird durch Radresonanz dominiert,

$\Rightarrow$ Anregung wird über Koppelpunkte (Achse / Federbeindom) zu Komfortpunkten übertragen,

$\Rightarrow$ Vibrationen, die vom Insassen an den Komfortpunkten wahrgenommen werden.

Entstehung der störenden Vibration

Anregung

Die Fahrzeuganregung wird durch Radresonanz dominiert,

$\Rightarrow$ Anregung wird über Koppelpunkte (Achse / Federbeindom) zu Komfortpunkten übertragen,

$\Rightarrow$ Vibrationen, die vom Insassen an den Komfortpunkten wahrgenommen werden.

Aktive Dämpfung: Aktuatoren erzeugen Gegenbewegung

Sensoren und Aktuatoren

- Sensoren registrieren Störung
- Aktuatoren erzeugen Gegenbewegung

$\Rightarrow$ Keine Bewegung am Windlauf

Aktive Dämpfung: Aktuatoren erzeugen Gegenbewegung

Sensoren und Aktuatoren

- Sensoren registrieren Störung
- Aktuatoren erzeugen Gegenbewegung

$\Rightarrow$ Keine Bewegung am Windlauf

Aktive Dämpfung: Aktuatoren erzeugen Gegenbewegung

Sensoren und Aktuatoren

- Sensoren registrieren Störung
- Aktuatoren erzeugen Gegenbewegung

$\Rightarrow$ Keine Bewegung am Windlauf

Training

Modelle

- Eine / alle Geschwindigkeiten
- Zeitreihen bis 500 ms
- Kombinationen der Beschleunigungssignale

Training der neuronalen Netze

- Mindestens 40% Testdaten
- Grad.-abstieg, Levenberg-Marquardt
- Abbruch: Fehler in Testdaten wachsen

Training

Modelle

- Eine / alle Geschwindigkeiten
- Zeitreihen bis 500 ms
- Kombinationen der Beschleunigungssignale

Training der neuronalen Netze

- Mindestens 40% Testdaten
- Grad.-abstieg, Levenberg-Marquardt
- Abbruch: Fehler in Testdaten wachsen

Training und Ergebnisse

Modelle

- Eine / alle Geschwindigkeiten
- Zeitreihen bis 500 ms
- Kombinationen der Beschleunigungssignale

Gute Ergebnisse

- Zeitreihen ca. 200 ms ,
- bei 2 und 4 Eingängen,
- mit beiden Lernverfahren
- schon bei kleinen Netzen $\Rightarrow$ stark lineares Verhalten der Karosserie und des Aktors

Training der neuronalen Netze

- Mindestens 40% Testdaten
- Grad.-abstieg, Levenberg-Marquardt
- Abbruch: Fehler in Testdaten wachsen

Validierung

Validierung

- Einbau in Simulink-Modell (MatLab bietet Export-Hilfe)
- Neuronale Netze liefert im Vergleich zu linearem Regler leicht bessere Komfortbewertung

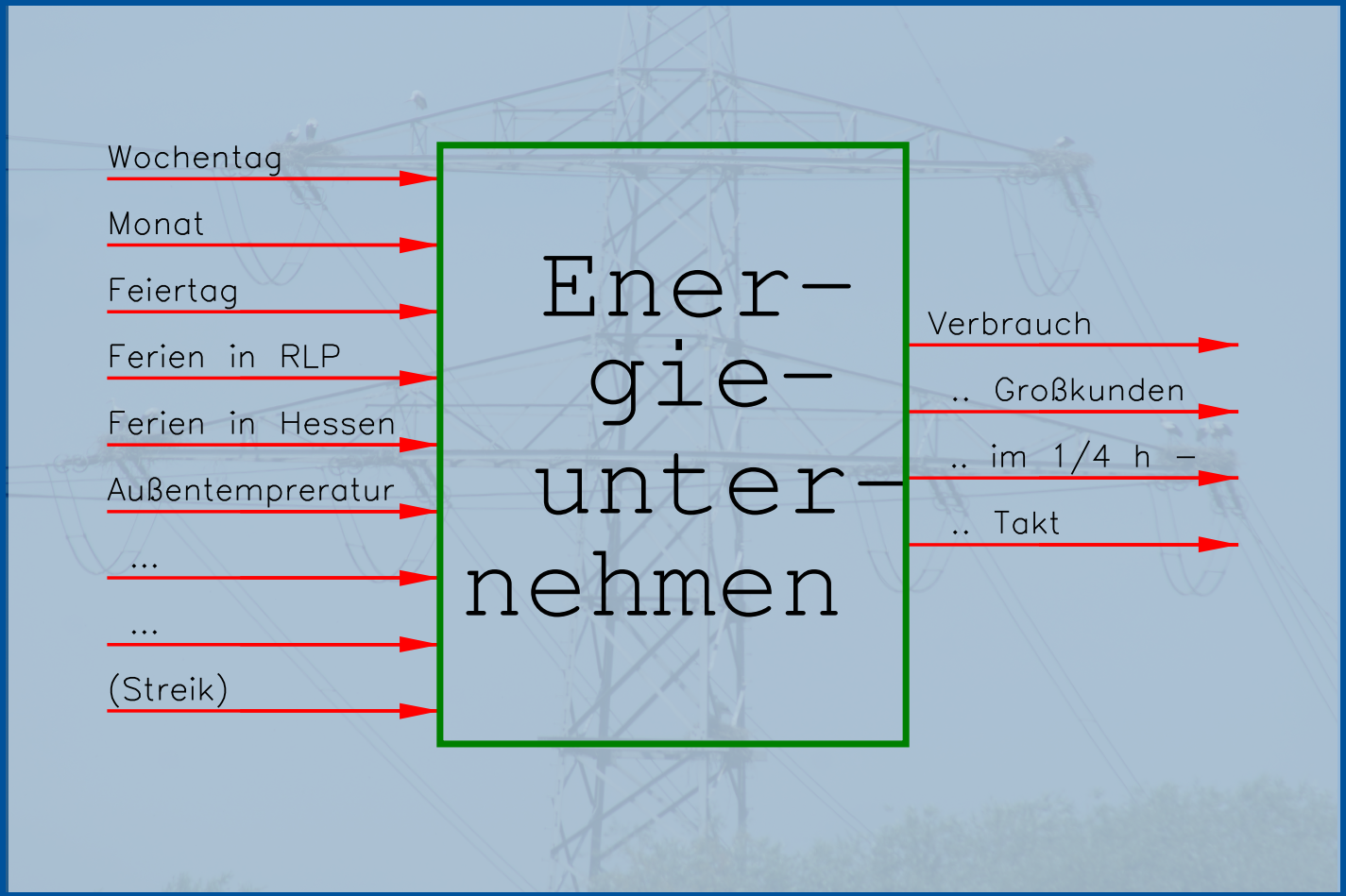
Validierung

Validierung

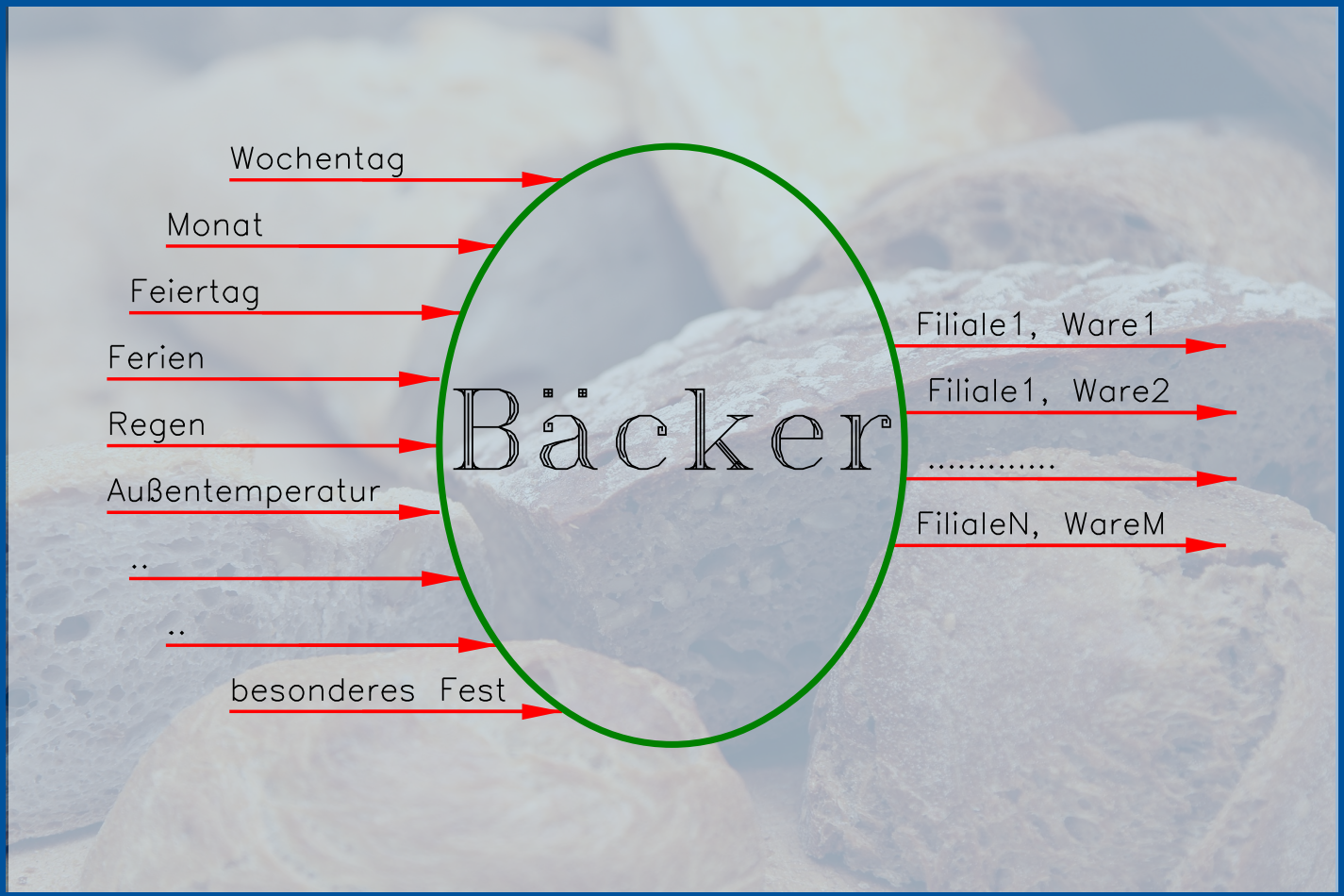
- Einbau in Simulink-Modell (MatLab bietet Export-Hilfe)
- Neuronale Netze liefert im Vergleich zu linearem Regler leicht bessere Komfortbewertung

Energieverbrauch Großkunden

EWR, DA Th. Müller, 2005

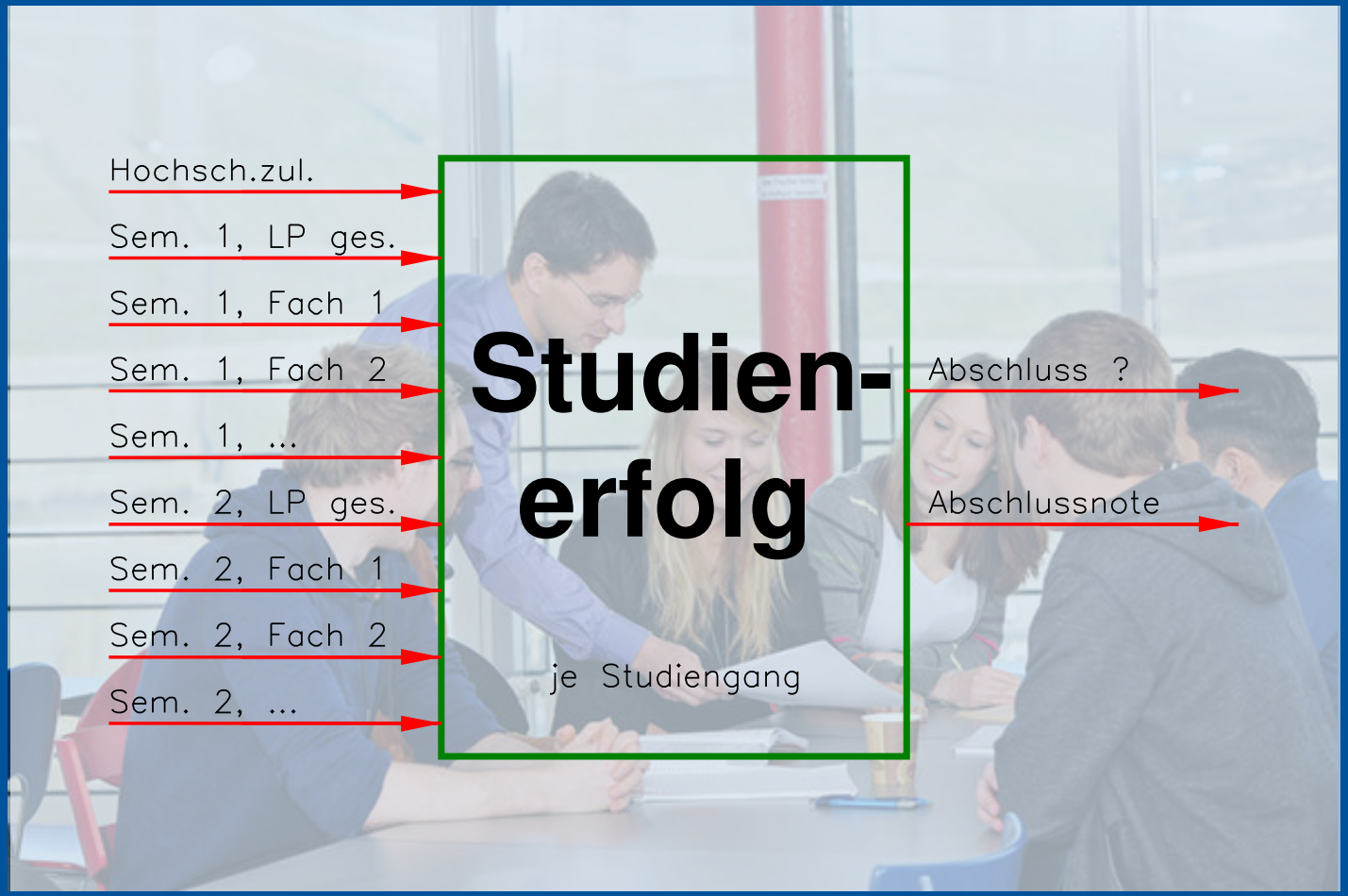


Vorhersage Verkauf Brotwaren



Vorhersage Verkauf Brotwaren

Vorhersage Studienerfolg



Beispiele

Insolvenzerkennung

Auf der Grundlage von Jahresbilanzen werden Muster ermittelt, die auf Insolvenzverdacht schließen lassen.

Beispiele

Fahrzeug, Robotersteuerung

Neuronales Netz steuert Fahrzeug oder Roboter korrekt.
Es lernt durch „Probefahren“..

Insolvenzerkennung

Auf der Grundlage von Jahresbilanzen werden Muster ermittelt, die auf Insolvenzverdacht schließen lassen.

Beispiele

Fahrzeug, Robotersteuerung

Neuronales Netz steuert Fahrzeug oder Roboter korrekt.
Es lernt durch „Probefahren“..

Insolvenzerkennung

Auf der Grundlage von Jahresbilanzen werden Muster ermittelt, die auf Insolvenzverdacht schließen lassen.

Aktienkursvorhersage

Vorhersage auf der Grundlage des bisherigen Kursverlaufs und betriebswirtschaftlicher Daten.

Beispiele

Insolvenzerkennung

Auf der Grundlage von Jahresbilanzen werden Muster ermittelt, die auf Insolvenzverdacht schließen lassen.

Potentielle Kündiger

Auf der Grundlage von Verbrauchs- und Kundendaten werden potentielle Kündiger ermittelt und mit Sonderkonditionen gehalten.

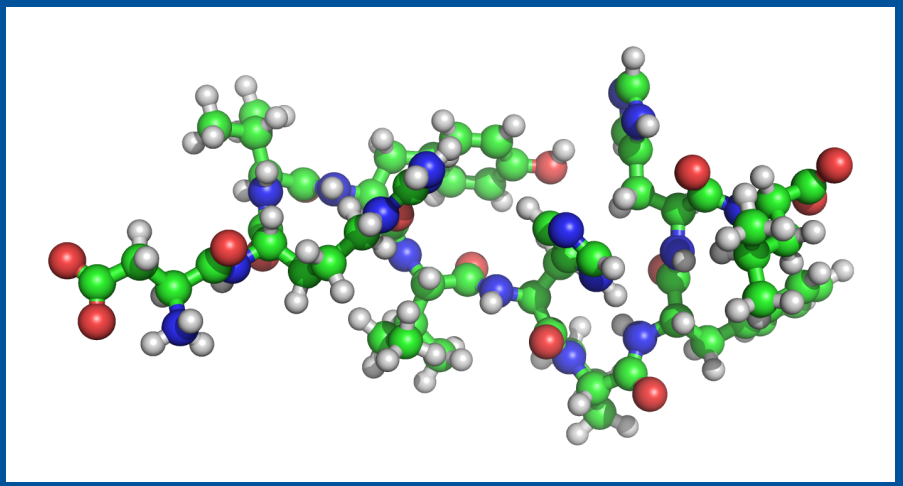
Aktienkursvorhersage

Vorhersage auf der Grundlage des bisherigen Kursverlaufs und betriebswirtschaftlicher Daten.

Beispiele

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.



Insolvenzerkennung

Auf der Grundlage von Jahresbilanzen werden Muster ermittelt, die auf Insolvenzverdacht schließen lassen.

Aktienkursvorhersage

Vorhersage auf der Grundlage des bisherigen Kursverlaufs und betriebswirtschaftlicher Daten.

Beispiele

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.

Herkunft Olivenöl

Die Konzentrationen von neun Säuren bestimmen die Herkunft (Region) italienischer Olivenölsorten.

Potentielle Kündiger

Auf der Grundlage von Verbrauchs- und Kundendaten werden potentielle Kündiger ermittelt und mit Sonderkonditionen gehalten.

Aktienkursvorhersage

Vorhersage auf der Grundlage des bisherigen Kursverlaufs und betriebswirtschaftlicher Daten.

Beispiele

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.

Herkunft Olivenöl

Die Konzentrationen von neun Säuren bestimmen die Herkunft (Region) italienischer Olivenölsorten.

Olfaktometer

Mikroquarzanlage mit 6 verschiedenen piezoelektrischen Sensoren. Neuronales Netz lernt, Duftstoffe zu erkennen.

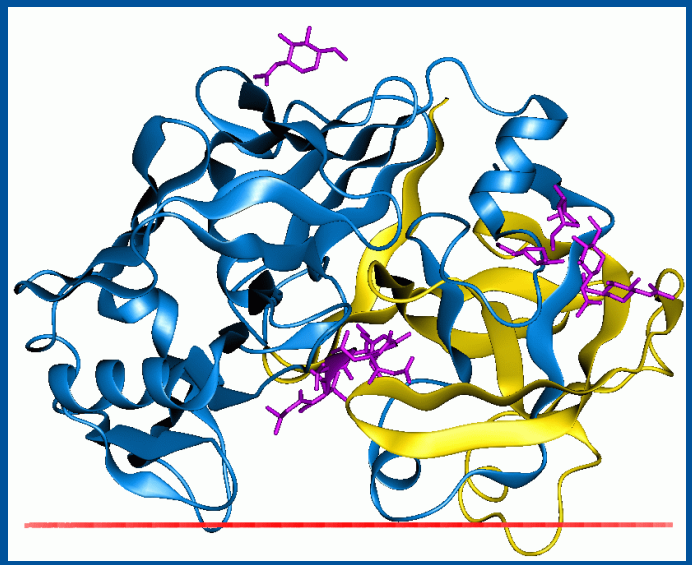
Aktienkursvorhersage

Vorhersage auf der Grundlage des bisherigen Kursverlaufs und betriebswirtschaftlicher Daten.

Beispiele

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.



Olfaktometer

Mikroquarzanlage mit 6 verschiedenen piezoelektrischen Sensoren. Neuronales Netz lernt, Duftstoffe zu erkennen.

Proteinstruktur

Aus der Primärstruktur der Proteine (Aminosäurenfolge) wird auf ihre (räumlich geometrische) Sekundärstruktur geschlossen.

Beispiele

Bremsmoment

Bestimmen des Bremsmoments aus hydraulischem Druck und Geschwindigkeit.

Olfaktometer

Mikroquarzanlage mit 6 verschiedenen piezoelektrischen Sensoren. Neuronales Netz lernt, Duftstoffe zu erkennen.

Proteinstruktur

Aus der Primärstruktur der Proteine (Aminosäurenfolge) wird auf ihre (räumlich geometrische) Sekundärstruktur geschlossen.

Beispiele

Bremsmoment

Bestimmen des Bremsmoments aus hydraulischem Druck und Geschwindigkeit.

Olfaktometer

Mikroquarzanlage mit 6 verschiedenen piezoelektrischen Sensoren. Neuronales Netz lernt, Duftstoffe zu erkennen.

Neuronales Stethoskop

Ein neuronales Netz interpretiert die Geräusche, die durch ein Stethoskop aufgenommen werden und stellt eine Diagnose bei Herzfehlern auf.

Sensorische Karten

Das Gehirn

- organisiert sich selbst,
- bildet Sinnesoberfläche auf Gehirnrindenfelder ab,
- baut Landkarte sensorischer Bezirke auf.

Übertragung auf eine Netzarchitektur mit dem Ziel

Eingabe bewirkt Gewichtsänderung, so dass die selbstorganisierende Karte die Topologie der Aufgabenstellung erkennt und abbildet.

Struktur

- Eingabeschicht
- topologische Ausgabeschicht:
  Jedem Neuron ist eine feste Position zugeordnet.

Sensorische Karten

Das Gehirn

- organisiert sich selbst,
- bildet Sinnesoberfläche auf Gehirnrindenfelder ab,
- baut Landkarte sensorischer Bezirke auf.

Übertragung auf eine Netzarchitektur mit dem Ziel

Eingabe bewirkt Gewichtsänderung, so dass die selbstorganisierende Karte die Topologie der Aufgabenstellung erkennt und abbildet.

Struktur

- Eingabeschicht
- topologische Ausgabeschicht:
  Jedem Neuron ist eine feste Position zugeordnet.

Sensorische Karten

Das Gehirn

- organisiert sich selbst,
- bildet Sinnesoberfläche auf Gehirnrindenfelder ab,
- baut Landkarte sensorischer Bezirke auf.

Übertragung auf eine Netzarchitektur mit dem Ziel

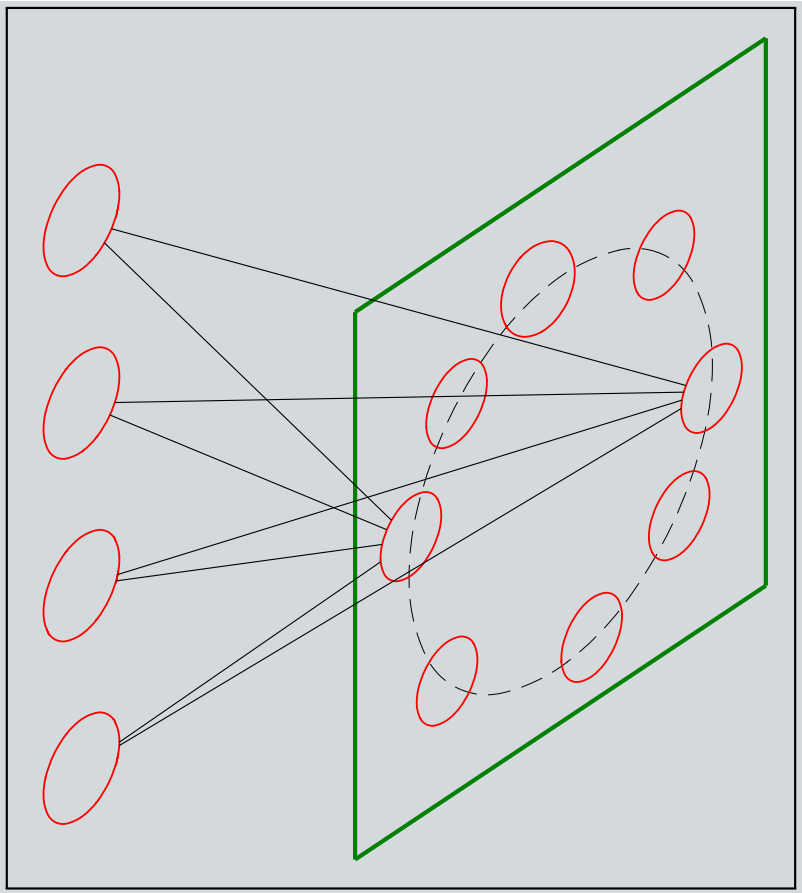
Eingabe bewirkt Gewichtsänderung, so dass die selbstorganisierende Karte die Topologie der Aufgabenstellung erkennt und abbildet.

Struktur

- Eingabeschicht
- topologische Ausgabeschicht:
  Jedem Neuron ist eine feste Position zugeordnet.

Sensorische Karten

Topologie



Lernverfahren

1. „Alles dem Sieger“
2. Neuronen im Erregungskreis ändern Gewichte, abnehmender Kreis

Sensorische Karten: Lernstrategie

Lernalgorithmus

1. Erregungsradius abnehmend:
$$\sigma(t) := \sigma_{\text{min}}^{t/t_{\text{max}}} \cdot \sigma_{\text{max}}^{1-t/t_{\text{max}}}$$

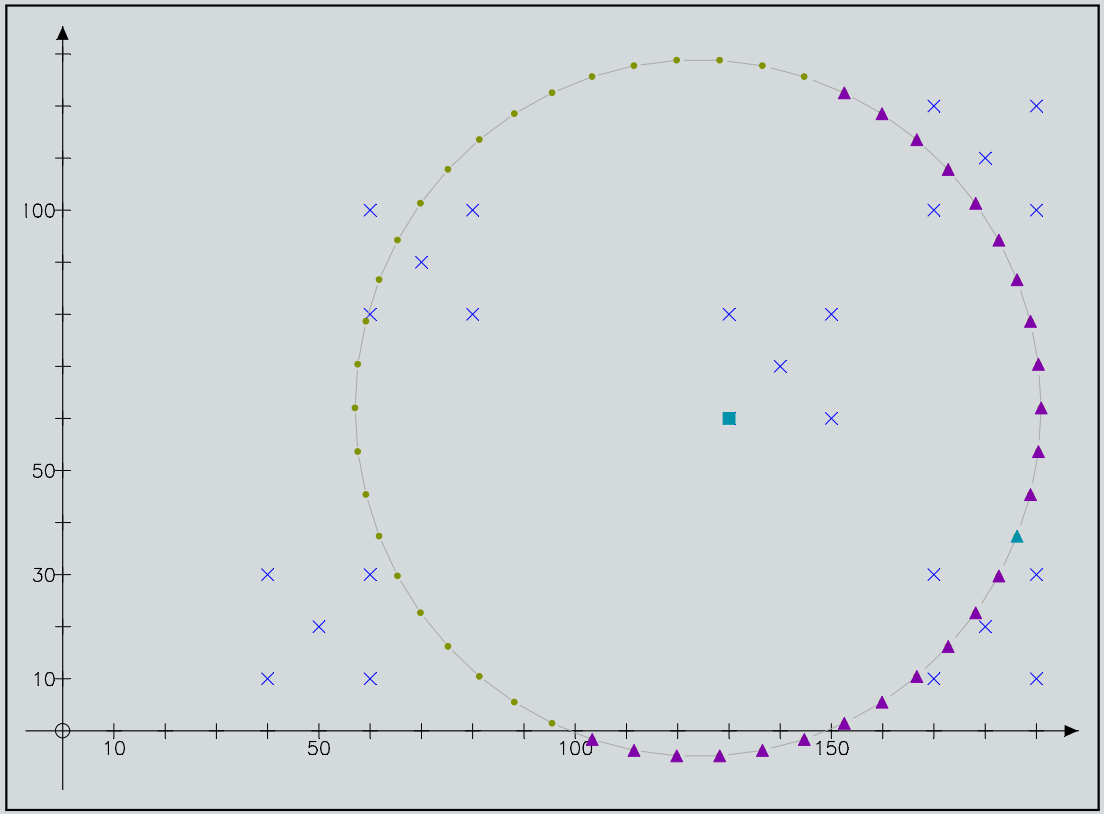
2. Lernrate abnehmend:
$$\varepsilon(t) := \varepsilon_{\text{min}}^{t/t_{\text{max}}} \cdot \varepsilon_{\text{max}}^{1-t/t_{\text{max}}}$$

3. Gewichtsänderungen Neuron i nahe Gewinner j ($d(i,j) \leq \sigma(t)$):
$$w_{ik}(t+1) = w_{ik}(t) + \varepsilon(t) \cdot \mathrm{e}^{-d(i,j)^2/2\sigma(t)^2} \cdot (e_k - \vec{w}_{ik}(t))$$

4. Am Ende der Lernphase:
Jede Eingabe wird mit dem Zentrum der Erregung identifiziert.

Sensorische Karten

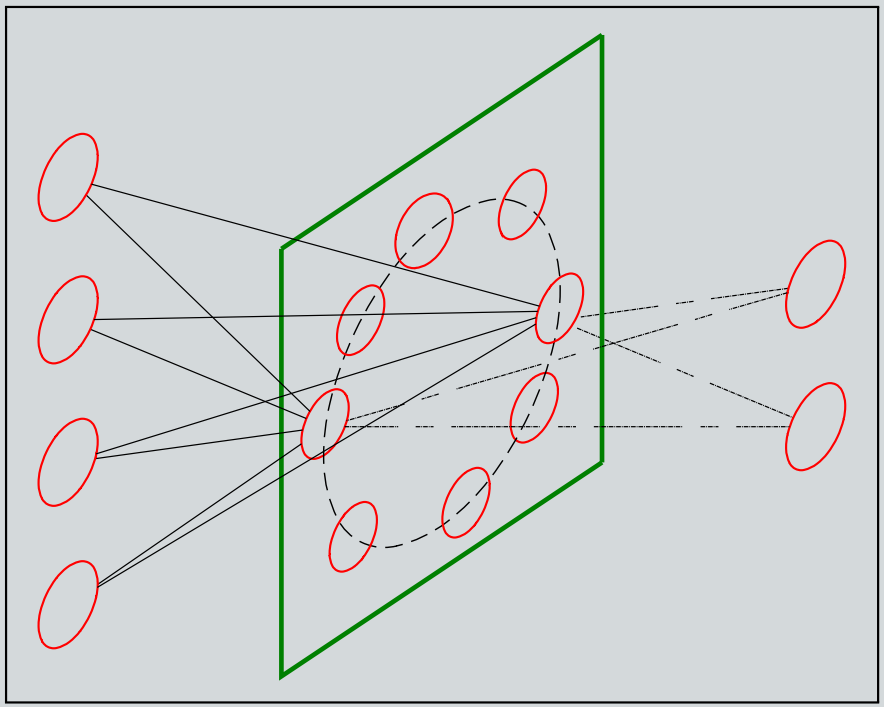
Initialisierung



Schach6, Platine2

Motorische Karte

Topologie

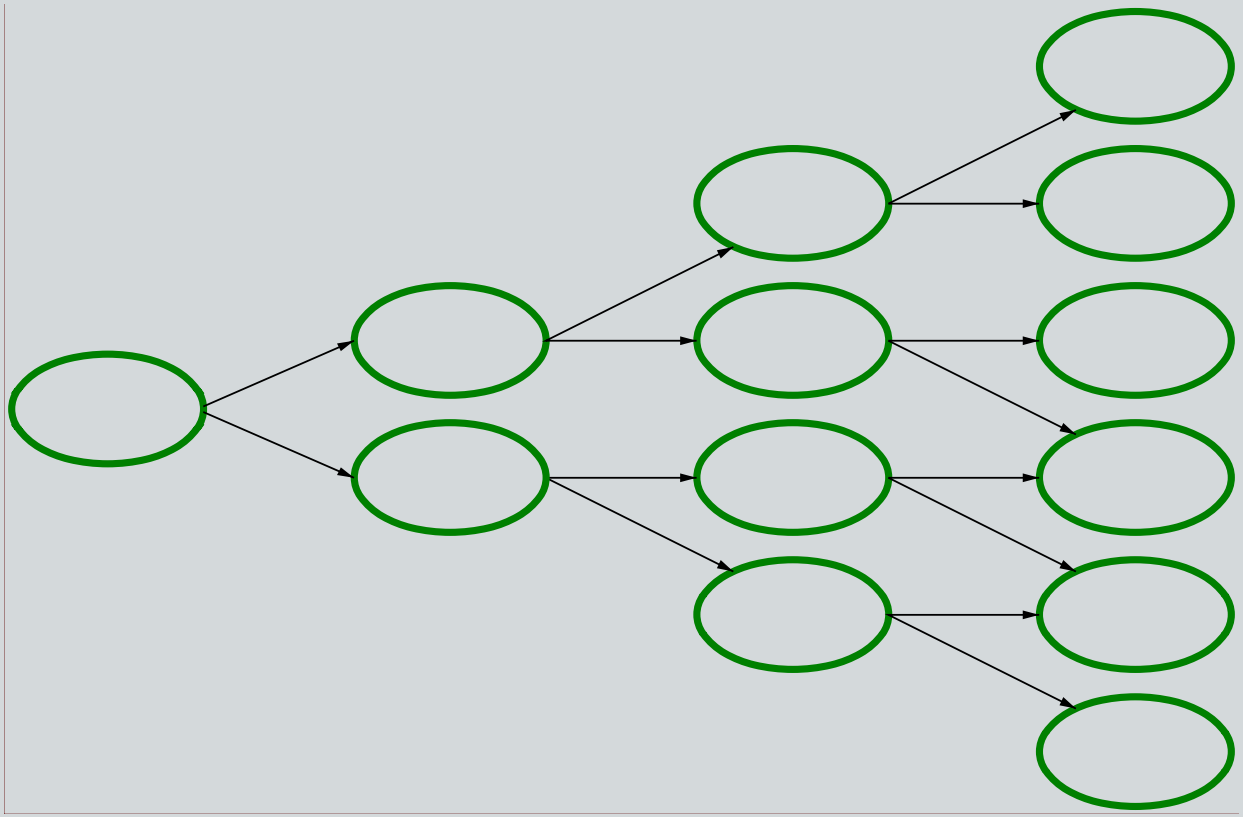


Lernverfahren

1. „Alles dem Sieger“
2. Neuronen im Erregungskreis ändern Gewichte, abnehmender Kreis

Hierarchische Klassifizierung

Mustererkennung verfeinernd



Big Data und Deep Learning

Anfallende Datenmengen sind sehr groß

- Bilder, Ton, Ultraschall, Laser-, Radarsignale
- Filme zu Verkehrsszenen (reell und Simulation)
- Schriftzeichen, Handschriften
- Aber Unterscheidung der Lern- und Produktivphase

... und das Lernen tief

- Neuronale Netze mit Millionen von Neuronen und Milliarden Verbindungen
- GPUs (graphical processing unit)

Big Data und Deep Learning

Anfallende Datenmengen sind sehr groß

- Bilder, Ton, Ultraschall, Laser-, Radarsignale
- Filme zu Verkehrsszenen (reell und Simulation)
- Schriftzeichen, Handschriften
- Aber Unterscheidung der Lern- und Produktivphase

... und das Lernen tief

- Neuronale Netze mit Millionen von Neuronen und Milliarden Verbindungen
- GPUs (graphical processing unit)

Big Data und Deep Learning

Anfallende Datenmengen sind sehr groß

- Bilder, Ton, Ultraschall, Laser-, Radarsignale
- Filme zu Verkehrsszenen (reell und Simulation)
- Schriftzeichen, Handschriften
- Aber Unterscheidung der Lern- und Produktivphase

... und das Lernen tief

- Neuronale Netze mit Millionen von Neuronen und Milliarden Verbindungen
- GPUs (graphical processing unit)

Big Data und Deep Learning

Anfallende Datenmengen sind sehr groß

- Bilder, Ton, Ultraschall, Laser-, Radarsignale
- Filme zu Verkehrsszenen (reell und Simulation)
- Schriftzeichen, Handschriften
- Aber Unterscheidung der Lern- und Produktivphase

... und das Lernen tief

- Neuronale Netze mit Millionen von Neuronen und Milliarden Verbindungen
- GPUs (graphical processing unit)

Big Data und Deep Learning

Anfallende Datenmengen sind sehr groß

- Bilder, Ton, Ultraschall, Laser-, Radarsignale
- Filme zu Verkehrsszenen (reell und Simulation)
- Schriftzeichen, Handschriften
- Aber Unterscheidung der Lern- und Produktivphase

... und das Lernen tief

- Neuronale Netze mit Millionen von Neuronen und Milliarden Verbindungen
- GPUs (graphical processing unit)

Big Data und Deep Learning

Anfallende Datenmengen sind sehr groß

- Bilder, Ton, Ultraschall, Laser-, Radarsignale
- Filme zu Verkehrsszenen (reell und Simulation)
- Schriftzeichen, Handschriften
- Aber Unterscheidung der Lern- und Produktivphase

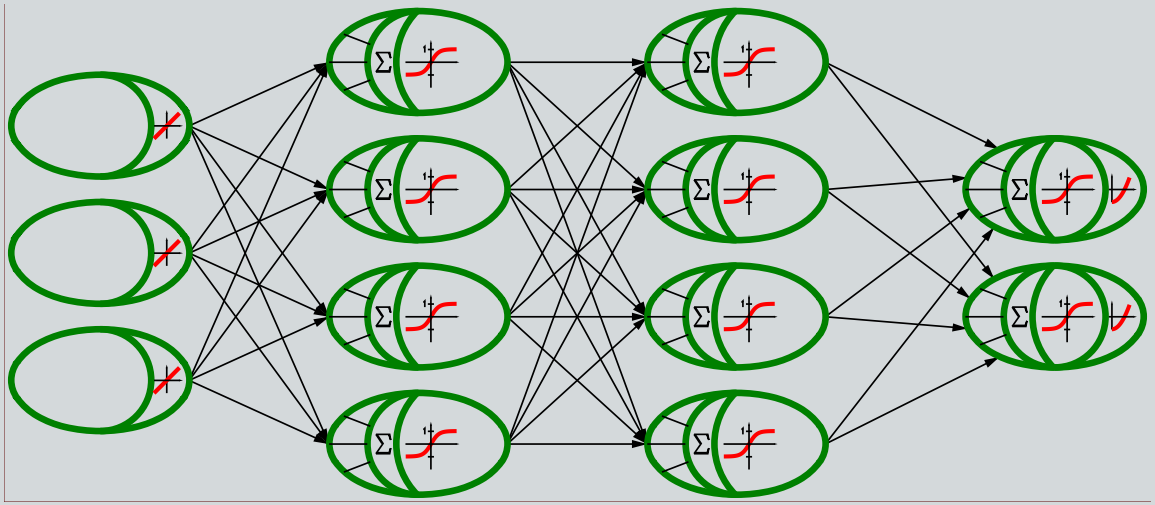
... und das Lernen tief

- Neuronale Netze mit Millionen von Neuronen und Milliarden Verbindungen
- GPUs (graphical processing unit)

Big Data und Deep Learning

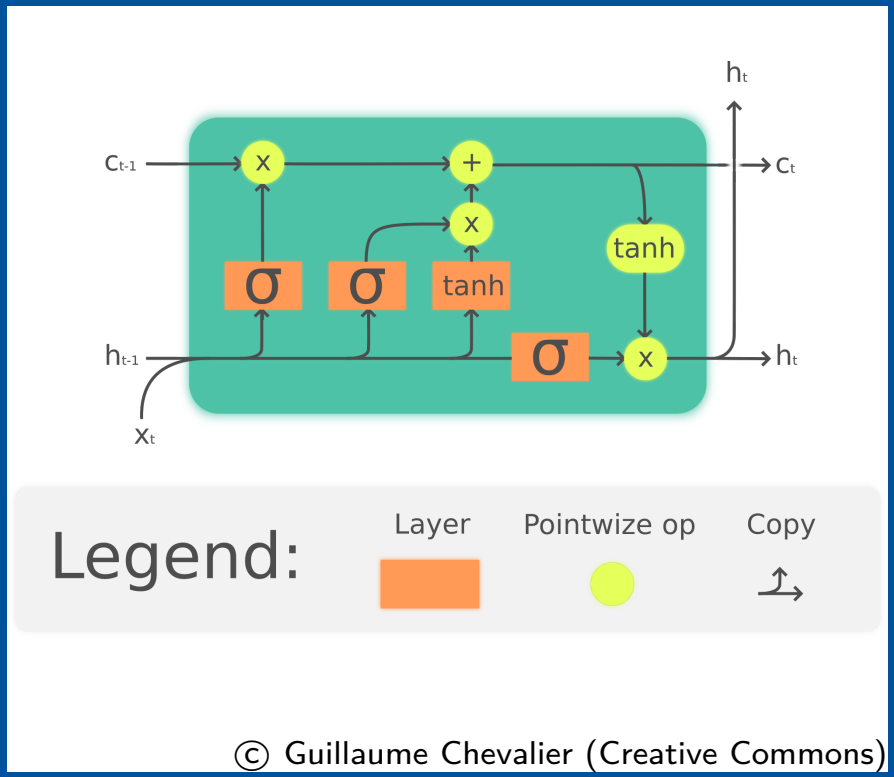
... und das Lernen tief

- Neuronale Netze mit Millionen von Neuronen und Milliarden Verbindungen
- GPUs (graphical processing unit)



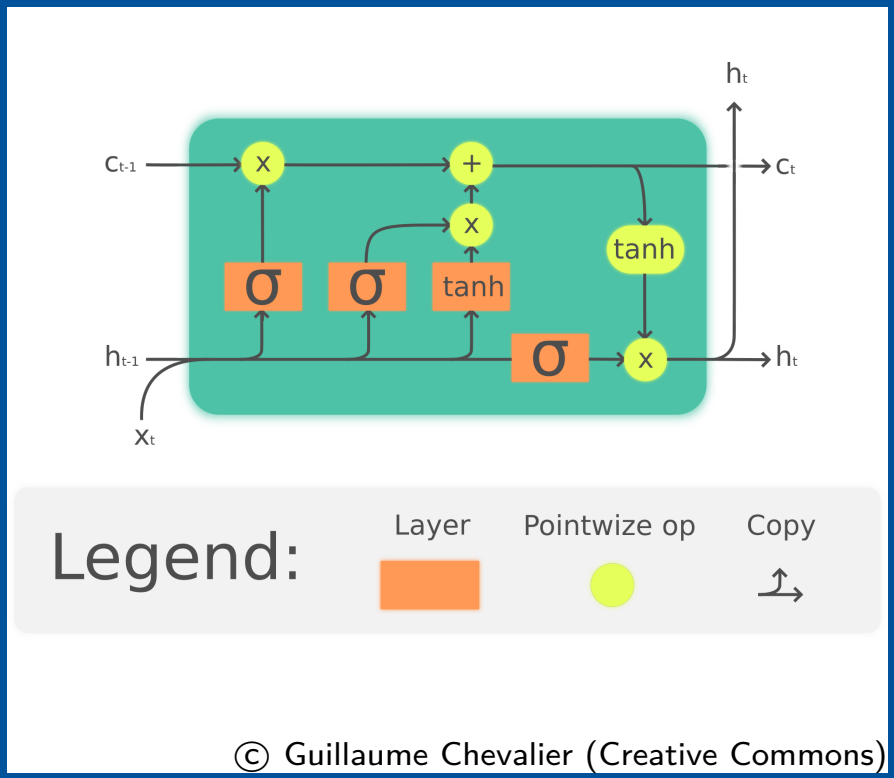
Hierarchisches Schichtenmodell

Langes Kurzzeitgedächtnis (Recurrent Network)



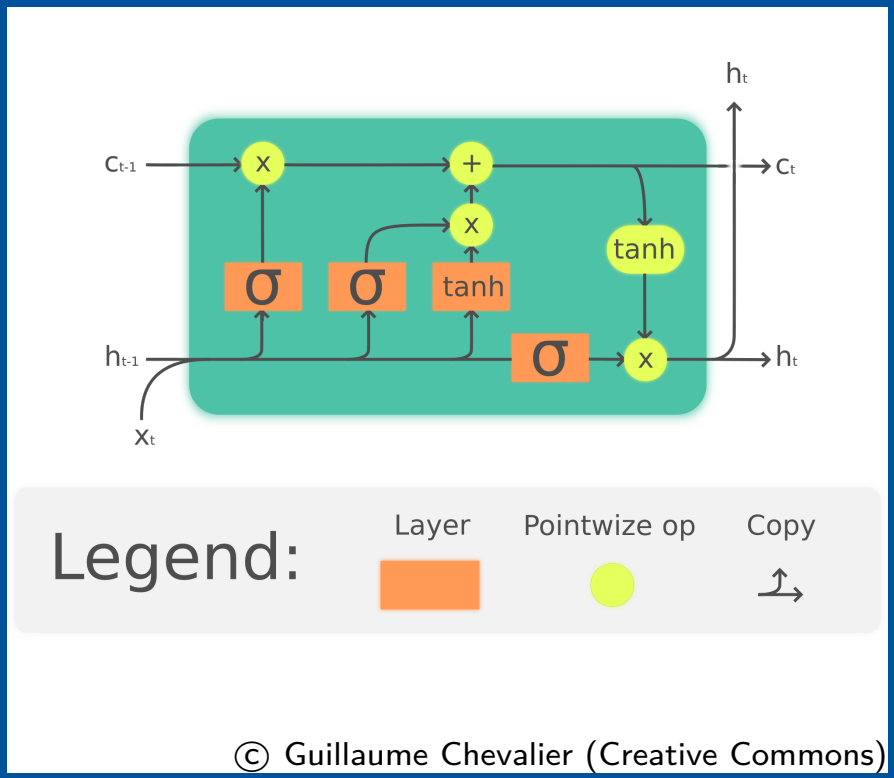
1. Zellenzustand und aktuelle Ausgabe werden gespeichert.
2. Zellenzustand kann vergessen und geändert werden.
3. Zellenzustand beeinflusst Ergebnis.

Langes Kurzzeitgedächtnis (Recurrent Network)



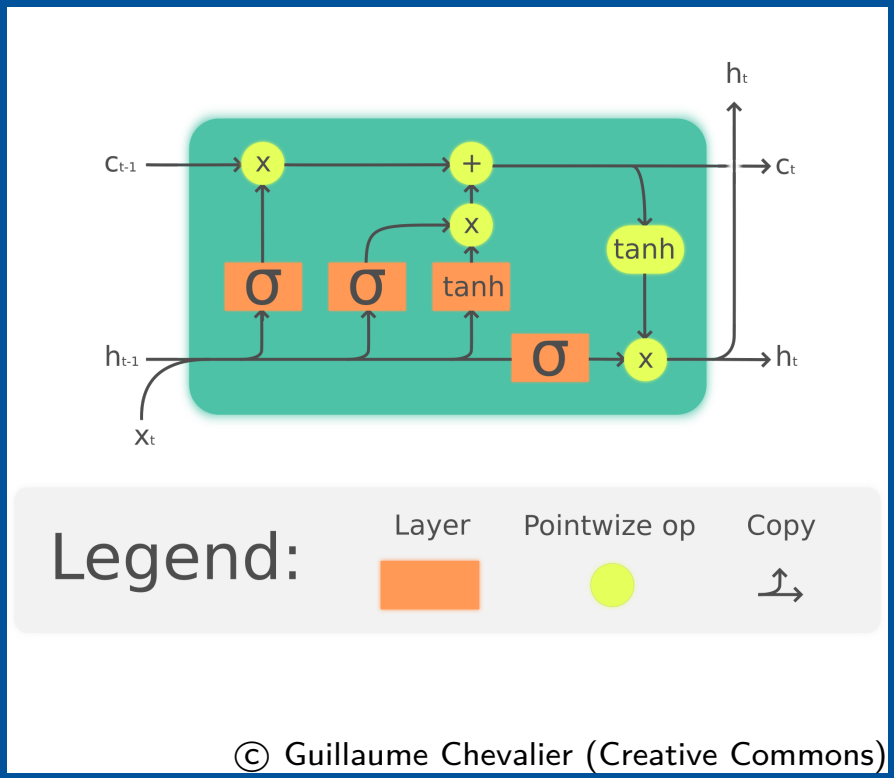
1. Zellenzustand und aktuelle Ausgabe werden gespeichert.
2. Zellenzustand kann vergessen und geändert werden.
3. Zellenzustand beeinflusst Ergebnis.

Langes Kurzzeitgedächtnis (Recurrent Network)



1. Zellenzustand und aktuelle Ausgabe werden gespeichert.
2. Zellenzustand kann vergessen und geändert werden.
3. Zellenzustand beeinflusst Ergebnis.

Langes Kurzzeitgedächtnis (Recurrent Network)



1. Zellenzustand und aktuelle Ausgabe werden gespeichert.
2. Zellenzustand kann vergessen und geändert werden.
3. Zellenzustand beeinflusst Ergebnis.

Dies ermöglicht ein zeitliches Gedächtnis.

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

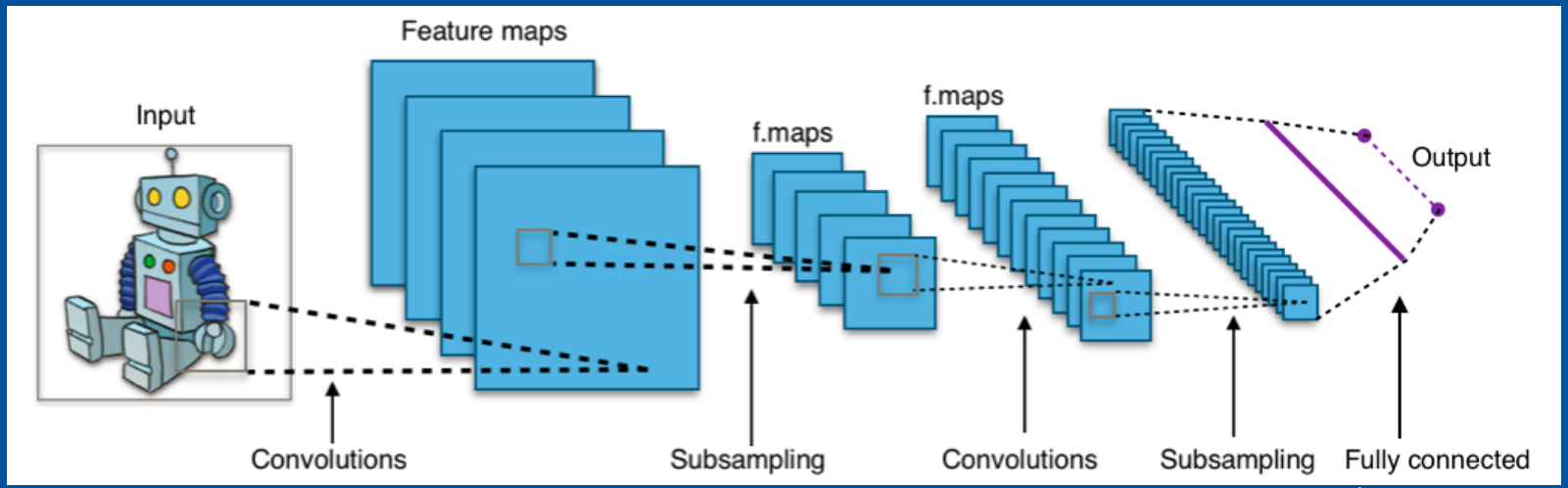
Erfolge vollständig vernetzter LSTM

1. ab 2003: Sprach- und Schriftzeichenerkennung
2. 2007: Schlüsselworterkennung
3. Gewonnene Wettbewerbe:
   - 2009: Erkennung französischer Handschrift
   - 2014: Erkennung arabischer Handschrift
   - ...
4. farsi, chinesisch
5. Schlüsselworterkennung
6. Spracherkennung mit LSTM: neueste Version google voice
7. Beschreibung von Videosequenzen

Langes Kurzzeitgedächtnis (Recurrent Network)

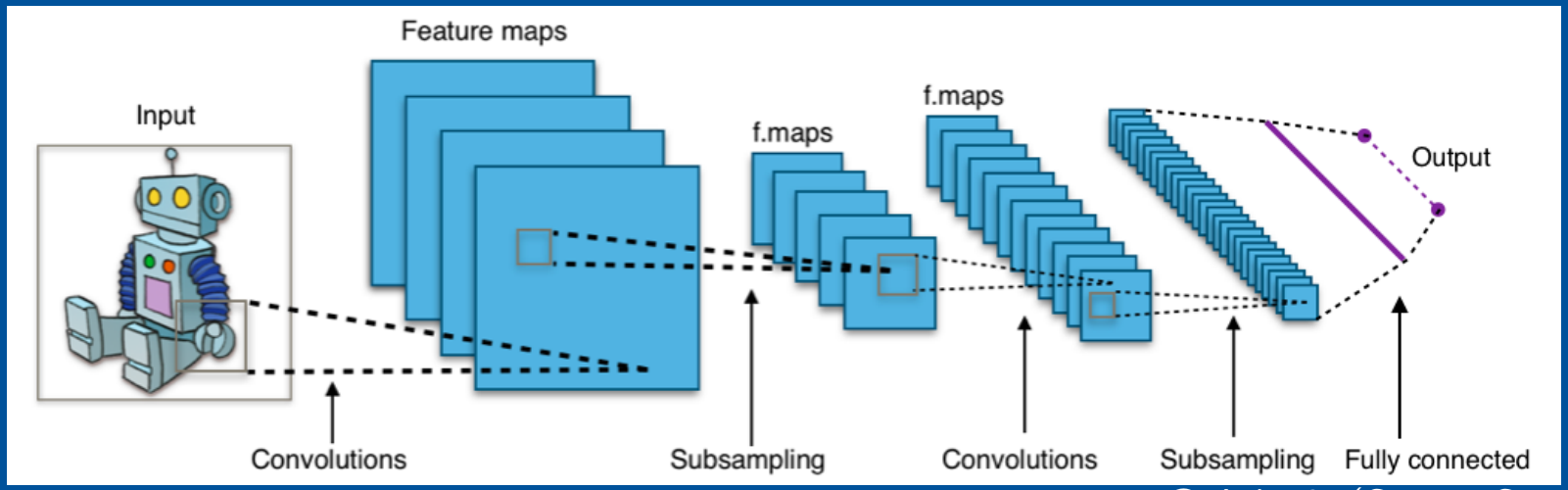
Google, 2014: Zu Bildern beschreibenden Text hinzufügen: Bild in Vinyals, Toshev u. a., “Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge”, S. 659 ..

Hierarchische neur. Netze (Convolutional NN)



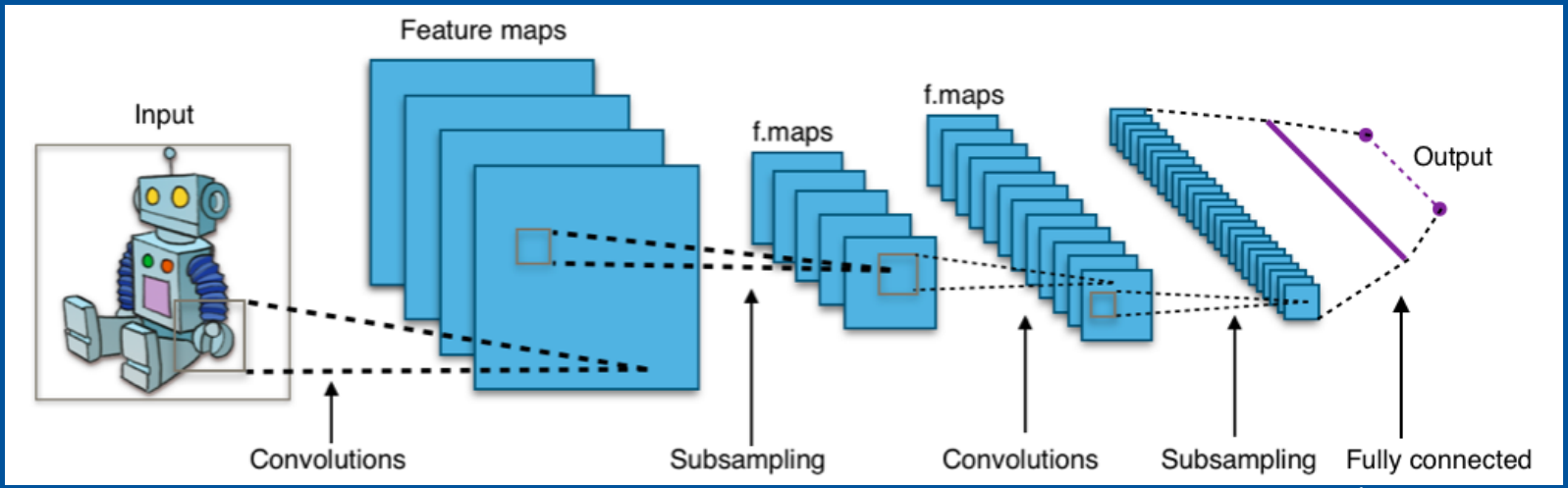
1. Teile des Gesamtbilds analysieren: Kanten erkennen
2. Ergebnisse bewerten: wahrscheinlichstes Merkmal übernehmen
3. Teile der Merkmale werden analysieren: Muster erkennen
4. Ergebnisse werden bewerten: wahrscheinlichstes Muster übernehmen
5. Geschehen erkennen (vollständiges Netz)

Hierarchische neur. Netze (Convolutional NN)



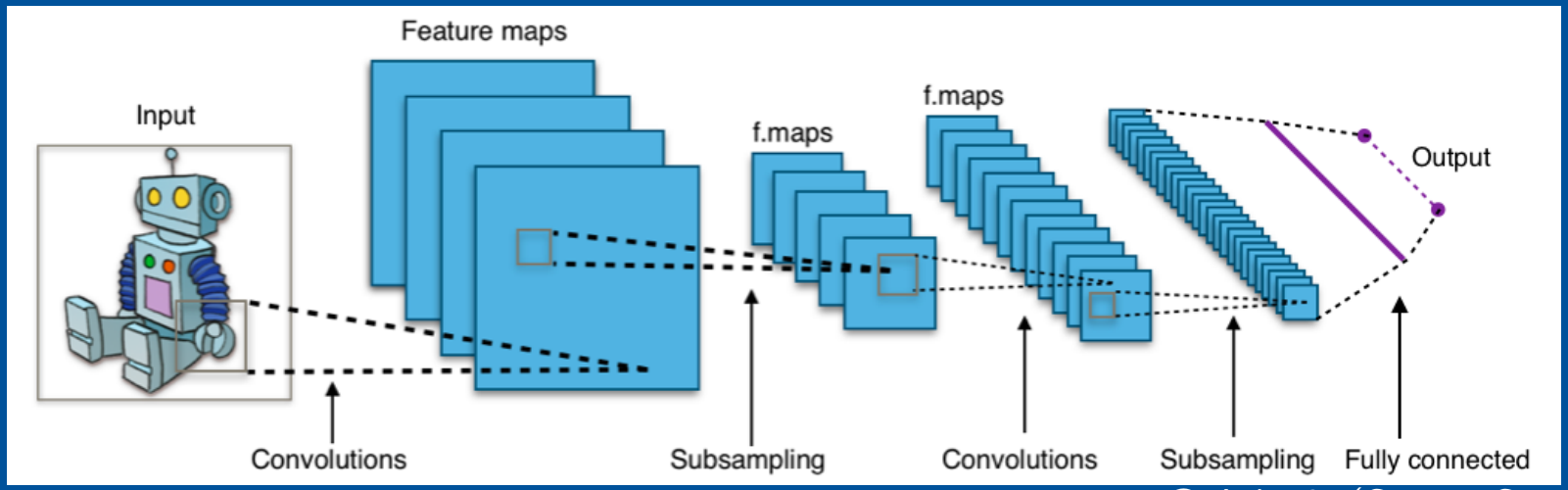
1. Teile des Gesamtbilds analysieren: Kanten erkennen
2. Ergebnisse bewerten: wahrscheinlichstes Merkmal übernehmen
3. Teile der Merkmale werden analysieren: Muster erkennen
4. Ergebnisse werden bewerten: wahrscheinlichstes Muster übernehmen
5. Geschehen erkennen (vollständiges Netz)

Hierarchische neur. Netze (Convolutional NN)



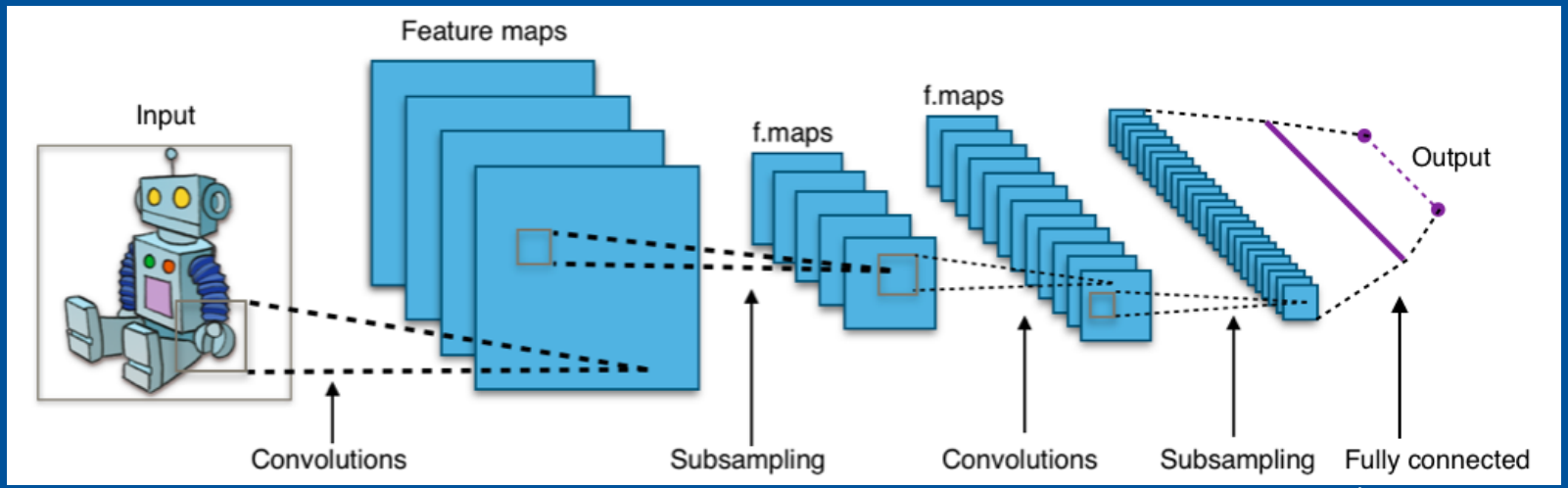
1. Teile des Gesamtbilds analysieren: Kanten erkennen
2. Ergebnisse bewerten: wahrscheinlichstes Merkmal übernehmen
3. Teile der Merkmale werden analysieren: Muster erkennen
4. Ergebnisse werden bewerten: wahrscheinlichstes Muster übernehmen
5. Geschehen erkennen (vollständiges Netz)

Hierarchische neur. Netze (Convolutional NN)



1. Teile des Gesamtbilds analysieren: Kanten erkennen
2. Ergebnisse bewerten: wahrscheinlichstes Merkmal übernehmen
3. Teile der Merkmale werden analysieren: Muster erkennen
4. Ergebnisse werden bewerten: wahrscheinlichstes Muster übernehmen
5. Geschehen erkennen (vollständiges Netz)

Hierarchische neur. Netze (Convolutional NN)



1. Teile des Gesamtbilds analysieren: Kanten erkennen
2. Ergebnisse bewerten: wahrscheinlichstes Merkmal übernehmen
3. Teile der Merkmale werden analysieren: Muster erkennen
4. Ergebnisse werden bewerten: wahrscheinlichstes Muster übernehmen
5. Geschehen erkennen (vollständiges Netz)

Big Data and Deep Learning

Sehr große CNN (convolutional neural network)

- Teile des Netzes Erkennen typische Merkmale in parallelen Sektionen
- Merkmale werden zusammengefasst und mit diesen weiter gelernt

Bild in ..

Big Data and Deep Learning

Sehr große CNN (convolutional neural network)

- Teile des Netzes Erkennen typische Merkmale in parallelen Sektionen
- Merkmale werden zusammengefasst und mit diesen weiter gelernt

Bild in ..

Big Data and Deep Learning

Sehr große CNN (convolutional neural network)

- Teile des Netzes Erkennen typische Merkmale in parallelen Sektionen
- Merkmale werden zusammengefasst und mit diesen weiter gelernt

Erfolge der CNN-Technik

- 2009: Deutliche Verbesserung der Spracherkennung (Yann LeCun, Geoffrey Hinton, George Dahl; Toronto)
- 2012: ImageNet Wettbewerb: Nach dem Training mit einer Million Bildern den Inhalt in neuen erkennen: Senken der Fehlerrate: 25% ↓ 15%
- 2013: Alle ImageNet-Wettbewerber benutzten diese Methode.
- 2013: Merck: Wer schlägt unser Programm zur Vorhersage der Wirkung von 30 000 kleinen Molekülen auf 15 Zielmoleküle.
  Gewinner: George Dahl mit CNN, um 15% besser.
- seither neue Anwendungsbereiche:
  Sprache verstehen, übersetzen; Wettervorhersage

Big Data and Deep Learning

Sehr große CNN (convolutional neural network)

- Teile des Netzes Erkennen typische Merkmale in parallelen Sektionen
- Merkmale werden zusammengefasst und mit diesen weiter gelernt

Erfolge der CNN-Technik

- 2009: Deutliche Verbesserung der Spracherkennung (Yann LeCun, Geoffrey Hinton, George Dahl; Toronto)
- 2012: ImageNet Wettbewerb: Nach dem Training mit einer Million Bildern den Inhalt in neuen erkennen: Senken der Fehlerrate: 25% ↓ 15%
- 2013: Alle ImageNet-Wettbewerber benutzten diese Methode.
- 2013: Merck: Wer schlägt unser Programm zur Vorhersage der Wirkung von 30 000 kleinen Molekülen auf 15 Zielmoleküle.
  Gewinner: George Dahl mit CNN, um 15% besser.
- seither neue Anwendungsbereiche:
  Sprache verstehen, übersetzen; Wettervorhersage

Big Data and Deep Learning

Sehr große CNN (convolutional neural network)

- Teile des Netzes Erkennen typische Merkmale in parallelen Sektionen
- Merkmale werden zusammengefasst und mit diesen weiter gelernt

Erfolge der CNN-Technik

- 2009: Deutliche Verbesserung der Spracherkennung (Yann LeCun, Geoffrey Hinton, George Dahl; Toronto)
- 2012: ImageNet Wettbewerb: Nach dem Training mit einer Million Bildern den Inhalt in neuen erkennen: Senken der Fehlerrate: 25% ↓ 15%
- 2013: Alle ImageNet-Wettbewerber benutzten diese Methode.
- 2013: Merck: Wer schlägt unser Programm zur Vorhersage der Wirkung von 30 000 kleinen Molekülen auf 15 Zielmoleküle.
  Gewinner: George Dahl mit CNN, um 15% besser.
- seither neue Anwendungsbereiche:
  Sprache verstehen, übersetzen; Wettervorhersage

Big Data and Deep Learning

Sehr große CNN (convolutional neural network)

- Teile des Netzes Erkennen typische Merkmale in parallelen Sektionen
- Merkmale werden zusammengefasst und mit diesen weiter gelernt

Erfolge der CNN-Technik

- 2009: Deutliche Verbesserung der Spracherkennung (Yann LeCun, Geoffrey Hinton, George Dahl; Toronto)
- 2012: ImageNet Wettbewerb: Nach dem Training mit einer Million Bildern den Inhalt in neuen erkennen: Senken der Fehlerrate: 25% ↓ 15%
- 2013: Alle ImageNet-Wettbewerber benutzten diese Methode.
- 2013: Merck: Wer schlägt unser Programm zur Vorhersage der Wirkung von 30 000 kleinen Molekülen auf 15 Zielmoleküle.
  Gewinner: George Dahl mit CNN, um 15% besser.
- seither neue Anwendungsbereiche:
  Sprache verstehen, übersetzen; Wettervorhersage

Big Data and Deep Learning

Sehr große CNN (convolutional neural network)

- Teile des Netzes Erkennen typische Merkmale in parallelen Sektionen
- Merkmale werden zusammengefasst und mit diesen weiter gelernt

Erfolge der CNN-Technik

- 2009: Deutliche Verbesserung der Spracherkennung (Yann LeCun, Geoffrey Hinton, George Dahl; Toronto)
- 2012: ImageNet Wettbewerb: Nach dem Training mit einer Million Bildern den Inhalt in neuen erkennen: Senken der Fehlerrate: 25% ↓ 15%
- 2013: Alle ImageNet-Wettbewerber benutzten diese Methode.
- 2013: Merck: Wer schlägt unser Programm zur Vorhersage der Wirkung von 30 000 kleinen Molekülen auf 15 Zielmoleküle.
  Gewinner: George Dahl mit CNN, um 15% besser.
- seither neue Anwendungsbereiche:
  Sprache verstehen, übersetzen; Wettervorhersage

Uber-Unfall

Autonomes Uber-Auto übersieht Fahrradfahrerin 18.3.2018

1. Fahrradfahrerin überquert Straße bei Dunkelheit.
2. Das Auto reagiert nicht.
3. Der Sicherheitsfahrer erkennt die Situation zu spät.
4. Die Fahrradfahrerin stirbt an den Verletzungen im Krankenhaus.

Uber-Unfall

Autonomes Uber-Auto übersieht Fahrradfahrerin 18.3.2018

1. Fahrradfahrerin überquert Straße bei Dunkelheit.
2. Das Auto reagiert nicht.
3. Der Sicherheitsfahrer erkennt die Situation zu spät.
4. Die Fahrradfahrerin stirbt an den Verletzungen im Krankenhaus.

Uber-Unfall

Autonomes Uber-Auto übersieht Fahrradfahrerin 18.3.2018

1. Fahrradfahrerin überquert Straße bei Dunkelheit.
2. Das Auto reagiert nicht.
3. Der Sicherheitsfahrer erkennt die Situation zu spät.
4. Die Fahrradfahrerin stirbt an den Verletzungen im Krankenhaus.

Uber-Unfall

Autonomes Uber-Auto übersieht Fahrradfahrerin 18.3.2018

1. Fahrradfahrerin überquert Straße bei Dunkelheit.
2. Das Auto reagiert nicht.
3. Der Sicherheitsfahrer erkennt die Situation zu spät.
4. Die Fahrradfahrerin stirbt an den Verletzungen im Krankenhaus.

Uber-Unfall

Autonomes Uber-Auto übersieht Fahrradfahrerin 18.3.2018

Offizielle Reaktionen

- Uber zahlt Entschädigung an Opfer-Familie.
- Der Gouverneur von Arizona entzieht Uber die Lizenz für alle autonomen Fahrzeuge.
- Grafik-Spezialist Nvidia stellte die Fahrten mit seiner Testflotte selbstfahrender Autos ein.

Uber-Unfall

Autonomes Uber-Auto übersieht Fahrradfahrerin 18.3.2018

Offizielle Reaktionen

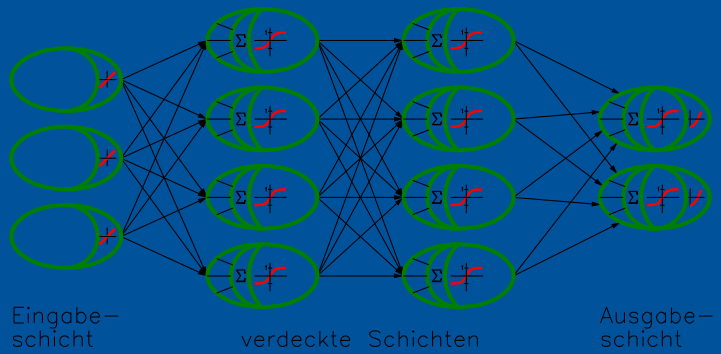
The Information 7.5.2018

Uber has determined that the likely cause ... was a problem with the software that decides how the car should react to objects it detects, according to two people briefed about the matter.
The car's sensors detected the pedestrian, who was crossing the street with a bicycle, but Uber's software decided it didn't need to react right away. That's a result of how the software was tuned. Like other autonomous vehicle systems, Uber's software has the ability to ignore "false positives", or objects in its path that wouldn't actually be a problem for the vehicle, such as a plastic bag floating over a road. In this case, Uber executives believe the company's system was tuned so that it reacted less to such objects. But the tuning went too far, and the car didn't react fast enough, one of these people said.

Zusammenfassung

Neuronale Netze eignen sich sehr gut

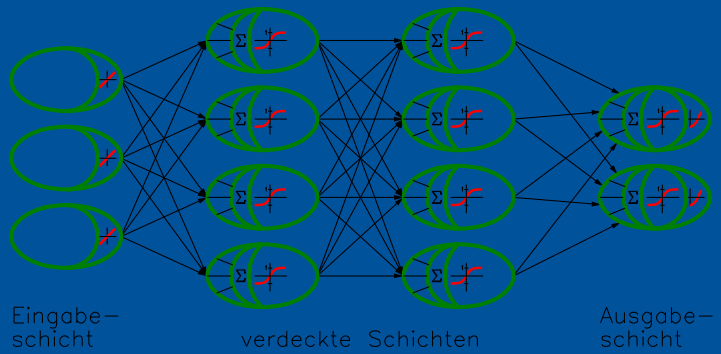
- zur Aufnahme von Systemfähigkeiten (KnowHow),
- zur Abbildung funktionaler Zusammenhänge



Zusammenfassung

Neuronale Netze eignen sich sehr gut

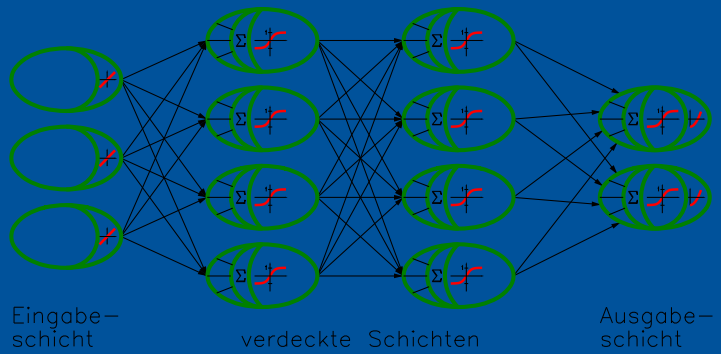
- zur Aufnahme von Systemfähigkeiten (KnowHow),
- zur Abbildung funktionaler Zusammenhänge



Zusammenfassung

Neuronale Netze eignen sich sehr gut

- zur Aufnahme von Systemfähigkeiten (KnowHow),
- zur Abbildung funktionaler Zusammenhänge

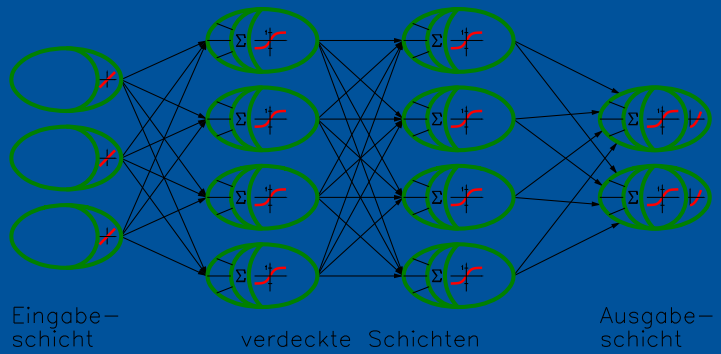


Zusammenfassung

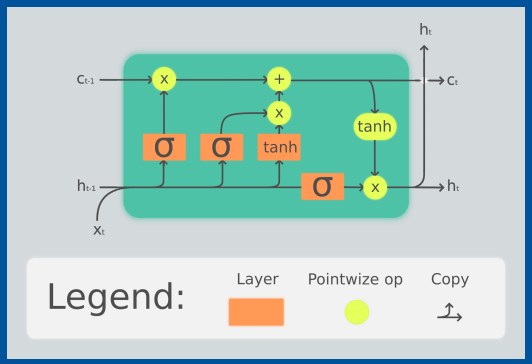
Neuronale Netze eignen sich sehr gut

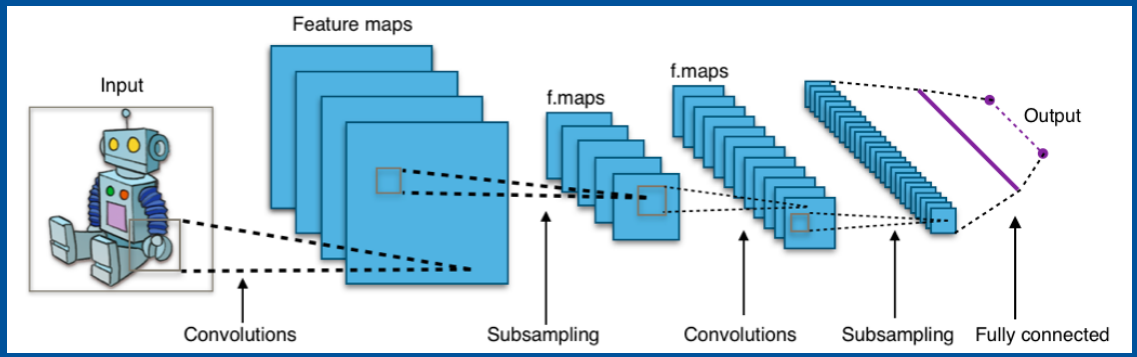
- zur Aufnahme von Systemfähigkeiten (KnowHow),
- zur Abbildung funktionaler Zusammenhänge

durch eine ausgleichende Interpolation zwischen den Stützstellen.



Zusammenfassung

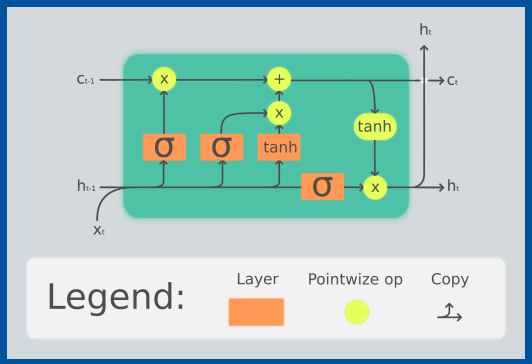


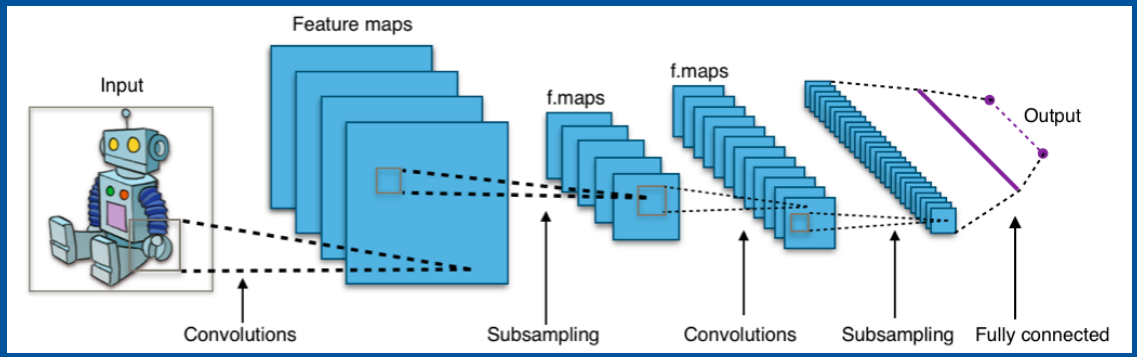


Große neuronale Netze erkennen

1. Handschrift,
2. Sprache,
3. Situationen and Handlung,
4. ...

Zusammenfassung

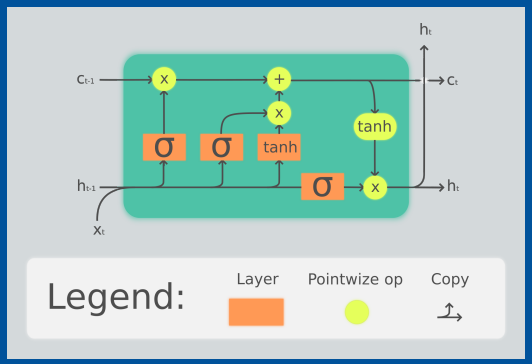


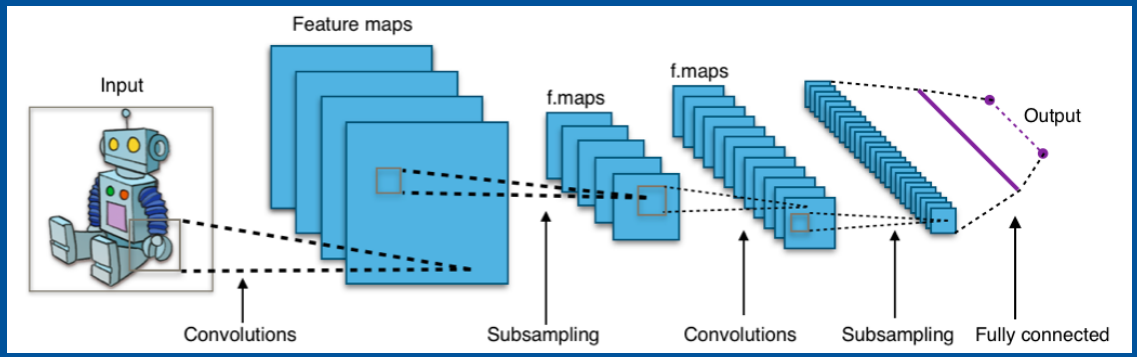


Große neuronale Netze erkennen

1. Handschrift,
2. Sprache,
3. Situationen and Handlung,
4. ...

Zusammenfassung

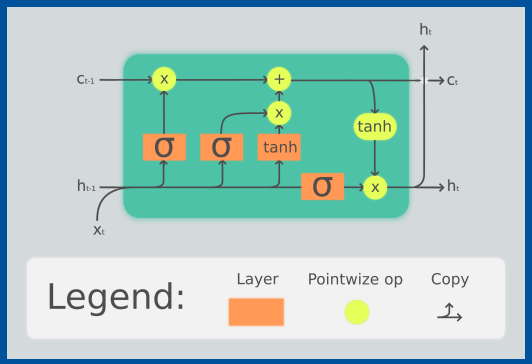


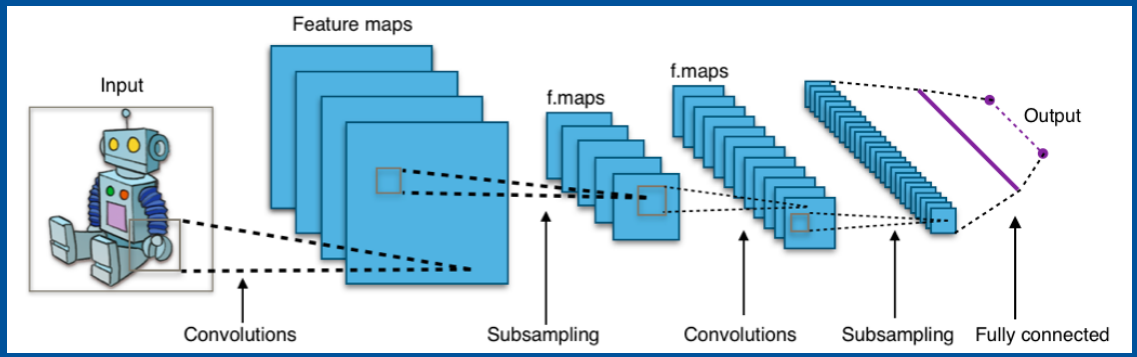


Große neuronale Netze erkennen

1. Handschrift,
2. Sprache,
3. Situationen and Handlung,
4. ...

Zusammenfassung

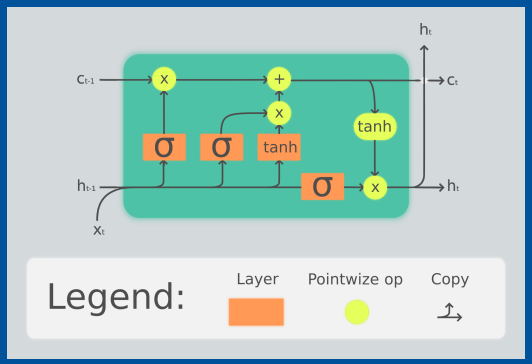


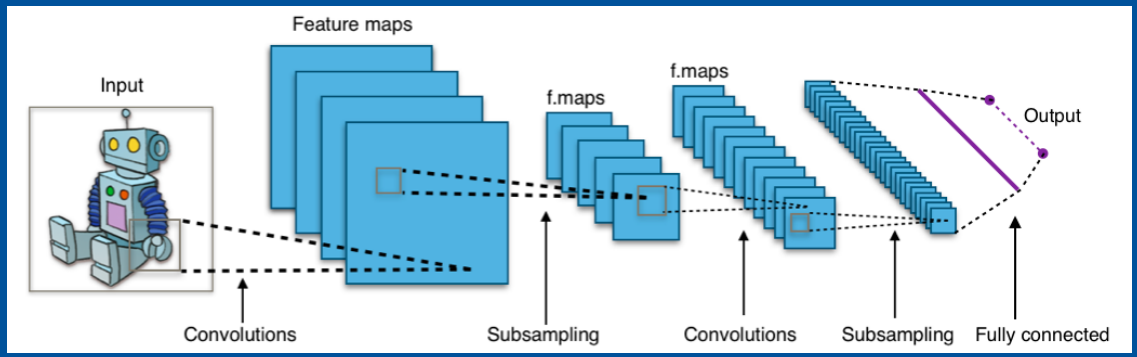


Große neuronale Netze erkennen

1. Handschrift,
2. Sprache,
3. Situationen and Handlung,
4. ...

Zusammenfassung

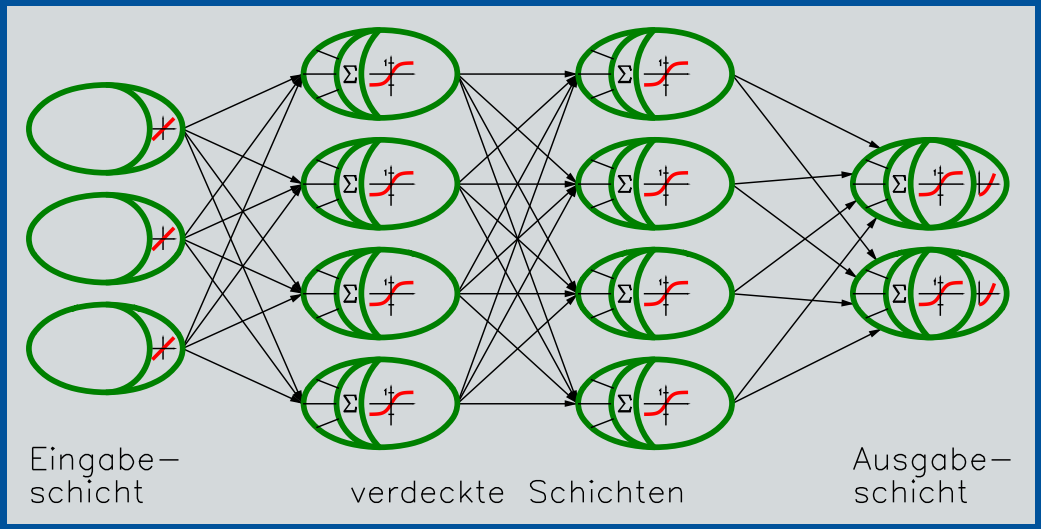




Große neuronale Netze erkennen

1. Handschrift,
2. Sprache,
3. Situationen and Handlung,
4. ...

Sie können Menschen in vielen Bereichen unterstützen!



Vielen Dank für Ihre Aufmerksamkeit!

Literatur I

Fay, R. „Hierachische neuronale Netze“. Masterarbeit. Universität Ulm, 2002. www.informatik.uni-ulm/../papers/masterthesis_fay_2002.pdf (besucht 12. 09. 2018).

Grünter, T. Deep Learning: Im „Kopf“ von künstlichen neuronalen Netzen. 2016. www.spektrum.de/video/im-kopf-von-kuenstlichen-neuronalen-netzen/1586616 (besucht 12. 09. 2018).

Helmig, C. HERE HD Live Map für vernetzte Smart Cars. HERE: www.here.com; www.mobilegeeks.de/?? 1. Sep. 2016. https://www.youtube.com/watch?v=R81_wg6gswA (besucht 11. 10. 2017).

JAAI. Convolutional Neural Networks – Aufbau, Funktion und Anwendungsgebiete. 27. Aug. 2015. jaai.de/convolutional-neural-networks-cnn-aufbau-funktion-und-anwendungsgebiete-1691/ (besucht 12. 09. 2018).

Jones, N. “The Learning Machines”. In: Nature 505 (Jan. 9, 2014). www.nvidia.com/content/tesla/pdf/machine-learning/nature-learning-machines.pdf (besucht 10/17/2017).

Khoi Nguyen, N. AI detectives are cracking open the black box of deep learning. July 6, 2017. https://www.spektrum.de/video/im-kopf-von-kuenstlichen-neuronalen-netzen/1586616 (besucht 09/12/2018).

Literatur II

LeCunn, Y., Y. Bengio, and G. Hinton. "Deep Learning". In: Nature 521 (May 28, 2015). DOI: 10.1038/nature14539.
www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf (besucht 10/17/2017).

Olah, C. Conv Nets: A Modular Perspective. July 8, 2014.
colah.github.io/posts/2014-07-Conv-Nets-Modular/ (besucht 09/12/2018).

– .Understanding LSTM Networks. Aug. 27, 2015.
colah.github.io/posts/2015-08-Understanding-LSTMs/ (besucht 09/12/2018).

Schmidhuber, J. Künstliche Intelligenz wird alles ändern. 2016.
www.youtube.com/watch?v=rafhHIQgd2A&t=690s (besucht 12. 09. 2018).

Vinyals, O., A. Toshev, et al. "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge". In: IEEE Transactions on Pattern Analysis and Machine Intelligence 39.4 (2017), pp. 652–663.
www.computer.org/csdl/trans/tp/2017/04/07505636.pdf.

von Hugo, C. Autonomes Fahren – Mehr als nur ein Hype. (IAA 2017 Future Talk). 18. Sep. 2017. www.youtube.com/watch?time_continue=62&v=8GzE0toDc24 (besucht 11. 10. 2017).

Zur Vertiefung

Filme

1. Schmidhuber: Künstliche Intelligenz wird alles ändern
2. von Hugo: Autonomes Fahren – Mehr als nur ein Hype. (IAA 2017 Future Talk)
3. Helmig: HERE HD Live Map für vernetzte Smart Cars
4. Khoi Nguyen: AI detectives are cracking open the black box of deep learning