Vorwärtsgerichtete neuronale Netze

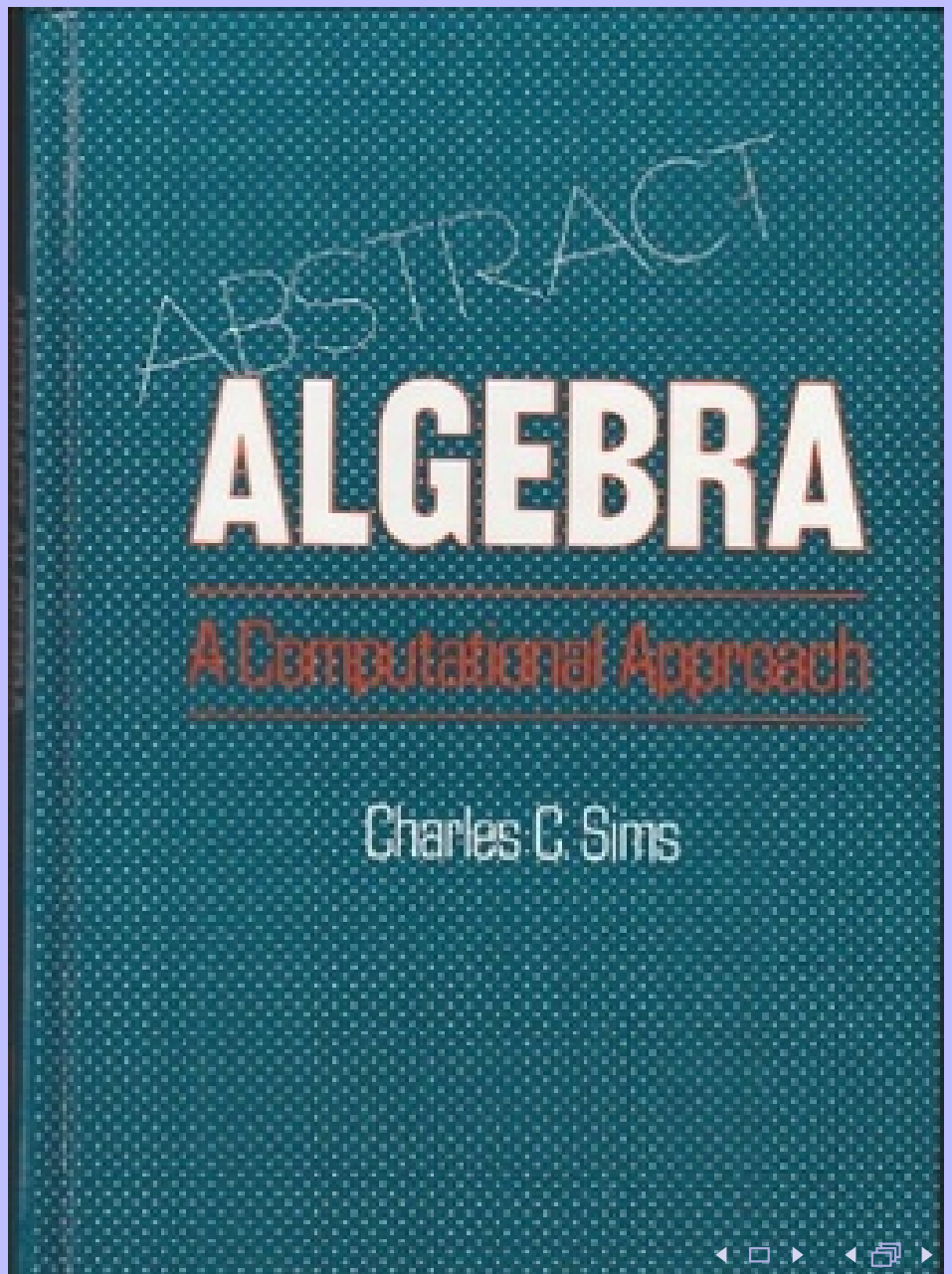
Modellbildung mit neuronalen Netzen

Dieter Kilsch

eh. Technische Hochschule Bingen, FB 2 • Technik, Informatik und Wirtschaft
GSE und APL-Germany
Böblingen

28. November 2016

Meine erste Begegnung mit APL: 1984



Meine erste APL-Konferenz: Ст. Петерсбург 1992

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 ?

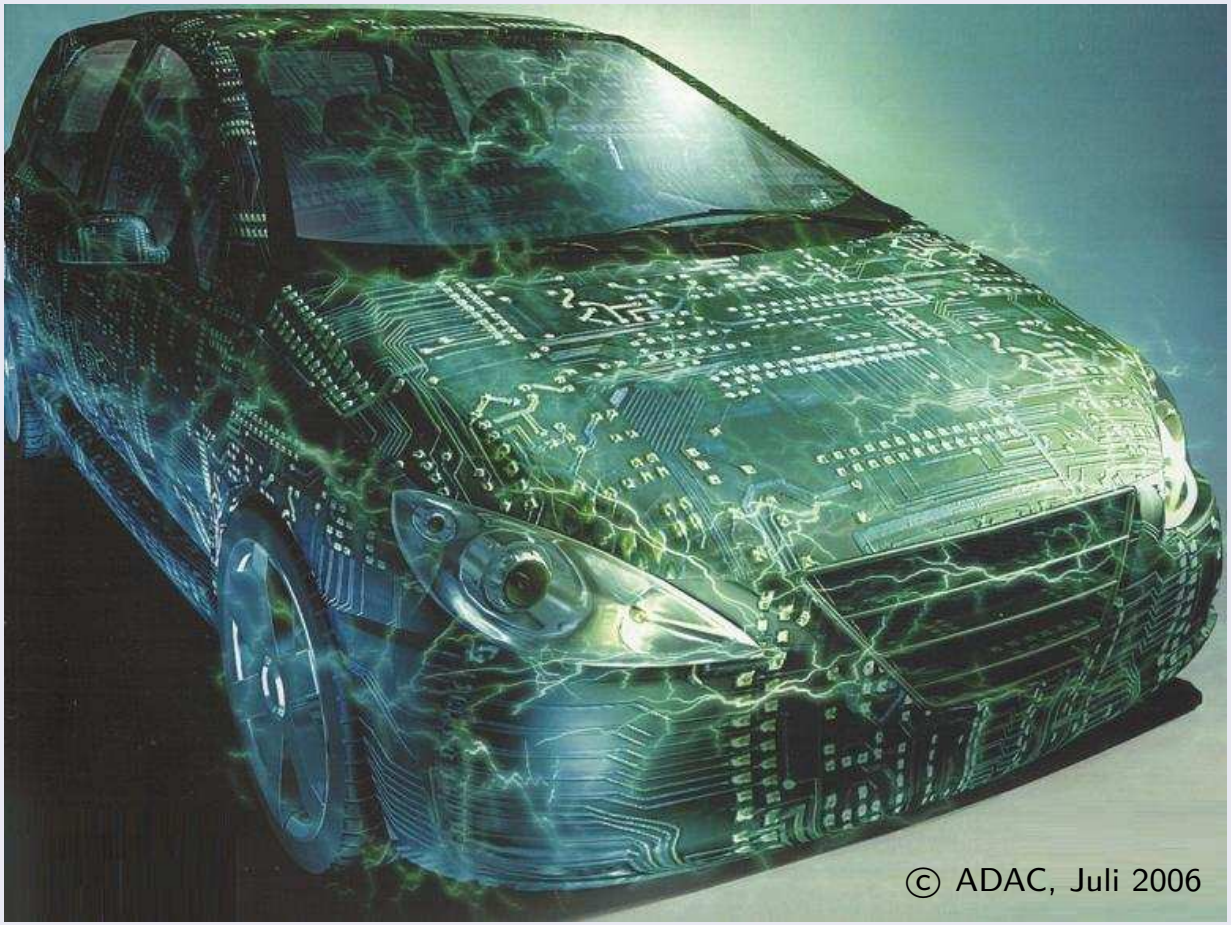
Das Auto lenkt,

der Mensch denkt,

aber nicht ans Fahren.

Vision: Voraussetzung

Vision



Das Auto „fühlt“

und „denkt“.

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.

Komfort durch autonomes Auto

Self-Driving Car Test: Steve Mahan

(youtube)

Autopiloten 4.10.2012

Das Wort Geisterfahrer bekommt in Kalifornien gerade eine ganz neue Bedeutung: Seit vergangener Woche dürfen auf den Highways selbststeuernde Autos cruisen. Wundern darf sich der Kalifornier also nicht, wenn er demnächst ein Auto ohne Fahrer neben sich hat. Denn der sitzt vermutlich auf der Rückbank und guckt Fernsehen, während der Autopilot lenkt.
Das neue System stammt von Google, dem es nun nicht mehr reicht, Straßen nur abzufilmen. 300 000 Meilen hätten die Gefährte schon unfallfrei zurückgelegt, teilte der Internetkonzern mit.

CHRISTINA KYRIASOGLOU © Die Zeit

Das Auto lenkt, der Mensch denkt, . . .

Komfort durch autonomes Auto

Freightliner Inspiration Truck Unveiled at Hoover Dam

LAS VEGAS, 5-5-2015, DTNA

First Licensed Autonomous Commercial Truck to Drive on U.S. Public Highway

In a spectacular evening ceremony at Hoover Dam, Daimler Trucks North America (DTNA) unveiled the Freightliner Inspiration Truck to several hundred international news media, trucking industry analysts and officials.

Komfort durch autonomes Auto

Freightliner Inspiration Truck Unveiled at Hoover Dam

LAS VEGAS, 5-5-2015, DTNA

First Licensed Autonomous Commercial Truck to Drive on U.S. Public Highway

In a spectacular evening ceremony at Hoover Dam, Daimler Trucks North America (DTNA) unveiled the Freightliner Inspiration Truck to several hundred international news media, trucking industry analysts and officials.

Daimler Inspiration Truck - So fährt der

Robo-Truck von Mercedes (youtube)

Komfort durch autonomes Auto

Freightliner Inspiration Truck Unveiled at Hoover Dam

LAS VEGAS, 5-5-2015, DTNA

First Licensed Autonomous Commercial Truck to Drive on U.S. Public Highway

In a spectacular evening ceremony at Hoover Dam, Daimler Trucks North America (DTNA) unveiled the Freightliner Inspiration Truck to several hundred international news media, trucking industry analysts and officials.

Daimler Inspiration Truck - So fährt der Robo-Truck von Mercedes (youtube)

The Freightliner Inspiration Truck is the first licensed autonomous commercial truck to operate on an open public highway in the United States. Developed by engineers at DTNA, it promises to unlock autonomous vehicle advancements that reduce accidents, improve fuel consumption, cut highway congestion, and safeguard the environment.

Komfort durch autonomes Auto

Freightliner Inspiration Truck Unveiled at Hoover Dam

LAS VEGAS, 5-5-2015, DTNA

First Licensed Autonomous Commercial Truck to Drive on U.S. Public Highway

In a spectacular evening ceremony at Hoover Dam, Daimler Trucks North America (DTNA) unveiled the Freightliner Inspiration Truck to several hundred international news media, trucking industry analysts and officials.

Daimler Inspiration Truck - So fährt der Robo-Truck von Mercedes (youtube)

The Freightliner Inspiration Truck is the first licensed autonomous commercial truck to operate on an open public highway in the United States. Developed by engineers at DTNA, it promises to unlock autonomous vehicle advancements that reduce accidents, improve fuel consumption, cut highway congestion, and safeguard the environment.

http://www.freightlinerinspiration.com/
http://www.freightlinerinspiration.com/newsroom/press/inspiration-truck-unveiled/
https://www.youtube.com/watch?v=mRkOGU3Gz9Y
https://www.youtube.com/watch?v=LL4dbq-n8Pg
https://www.youtube.com/watch?v=LJz4Ms_5AXE

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

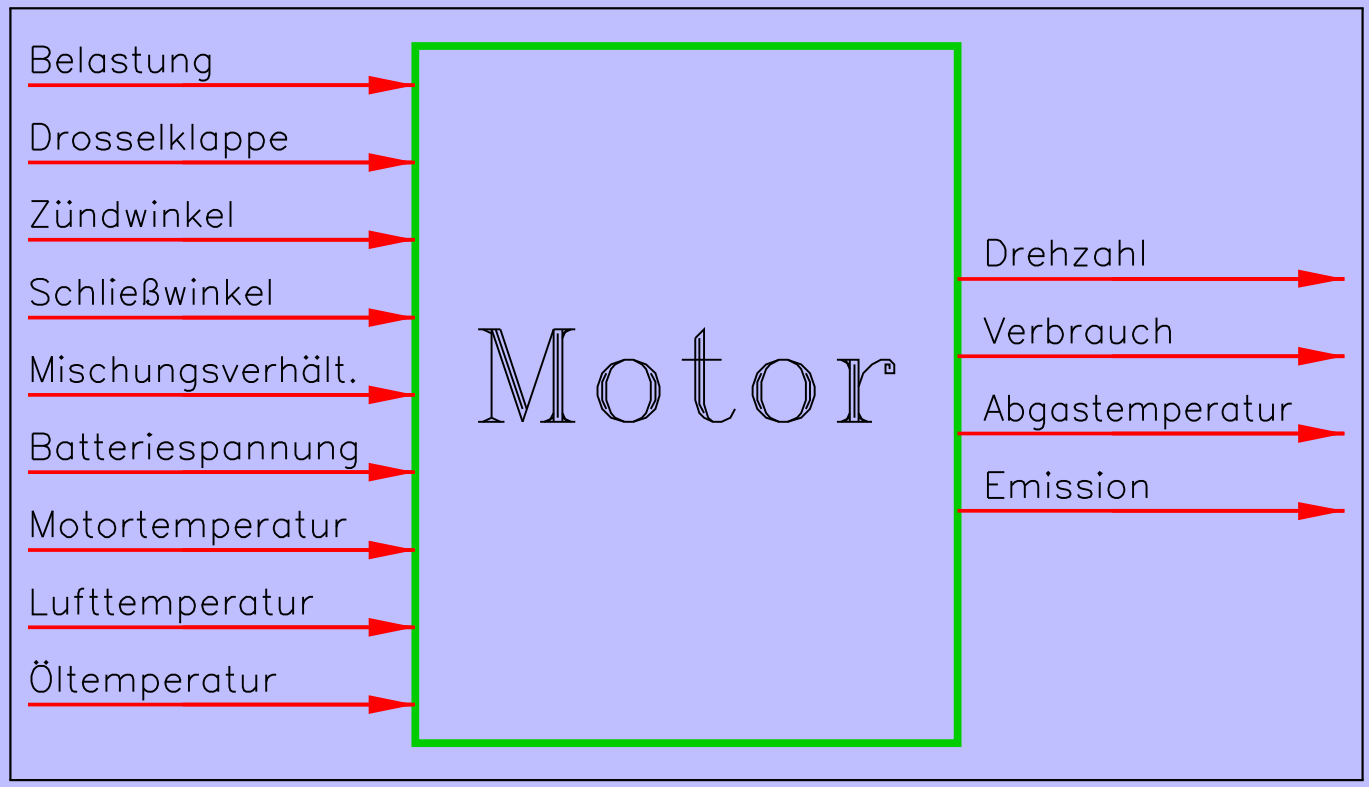
Expertensystem

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

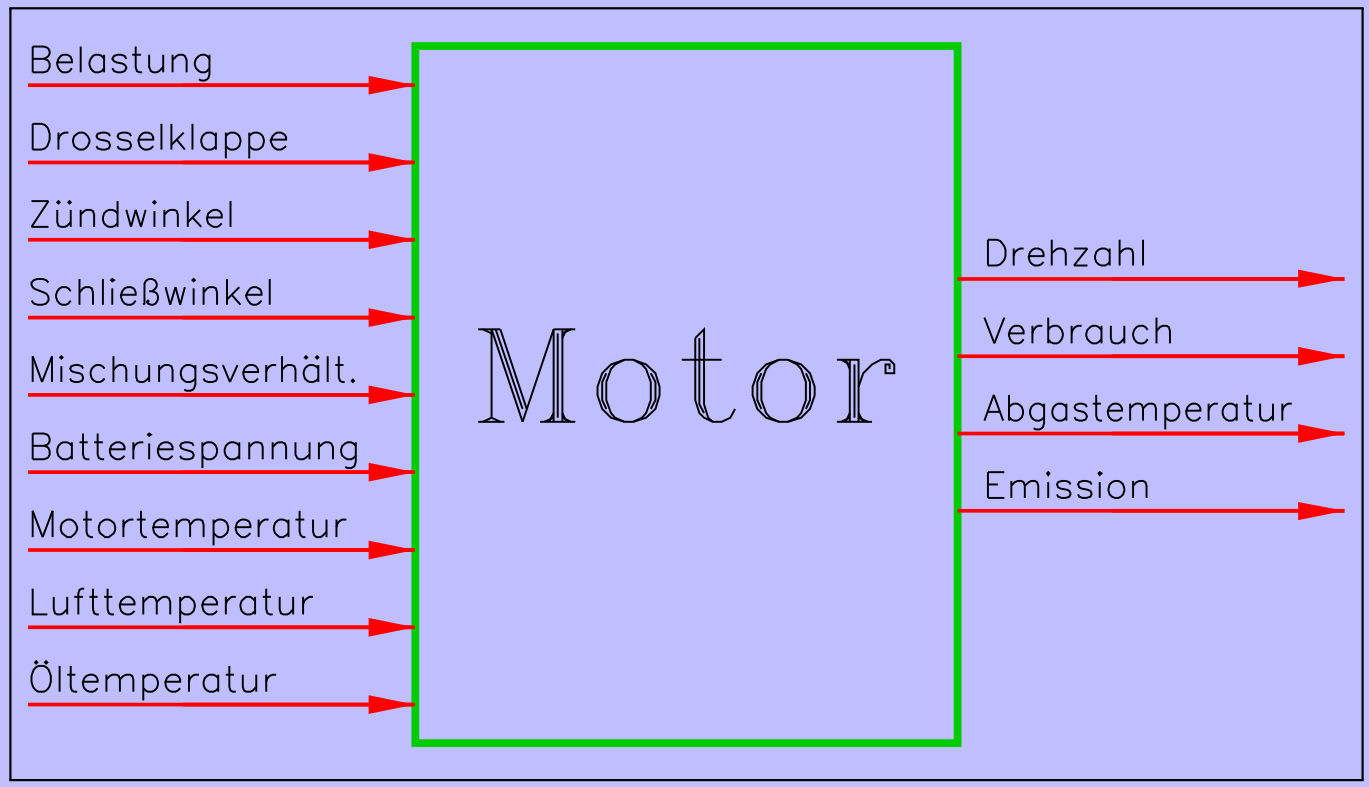
Neurale Netze

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

Physikalisches Wirkverhalten: Motor auf dem Prüfstand



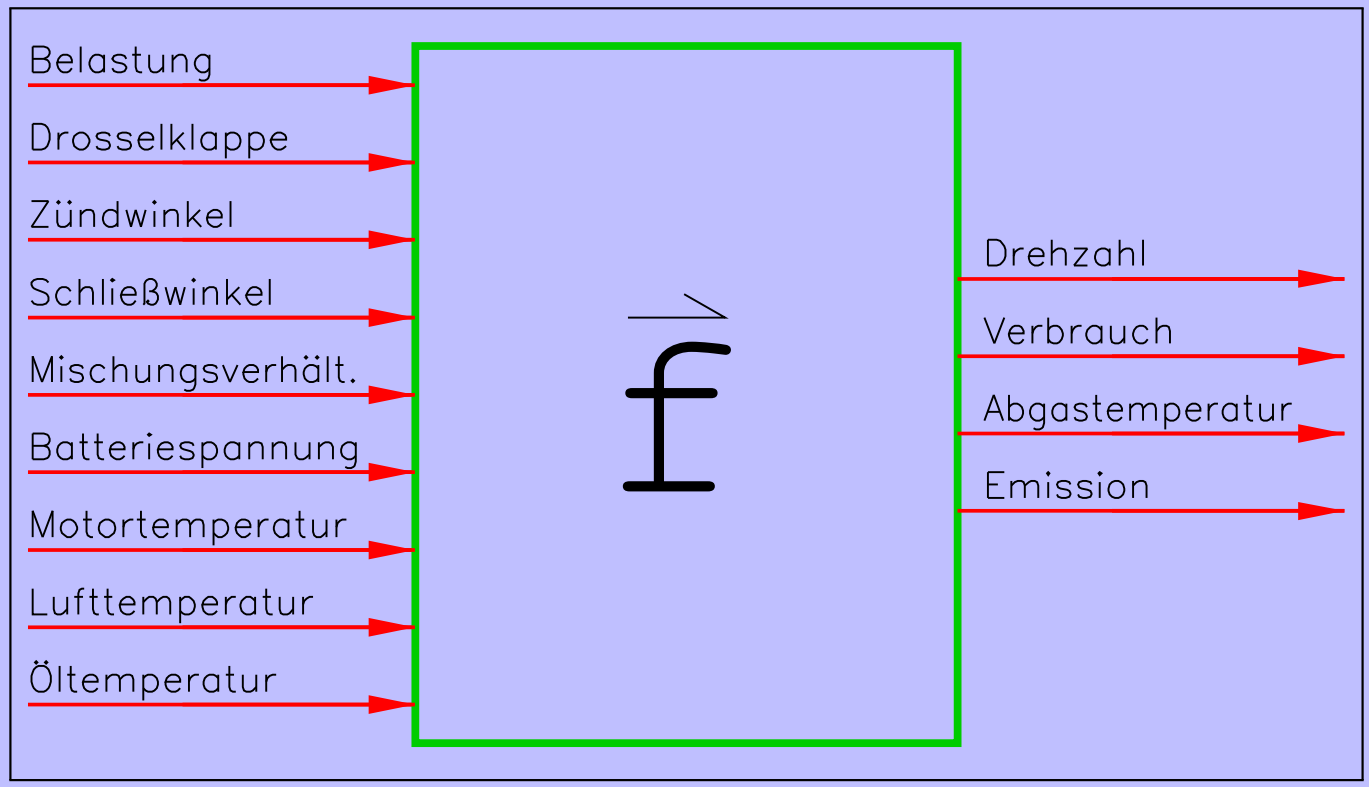
Physikalisches Wirkverhalten: Motor auf dem Prüfstand



Aufgabe und Ziel

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

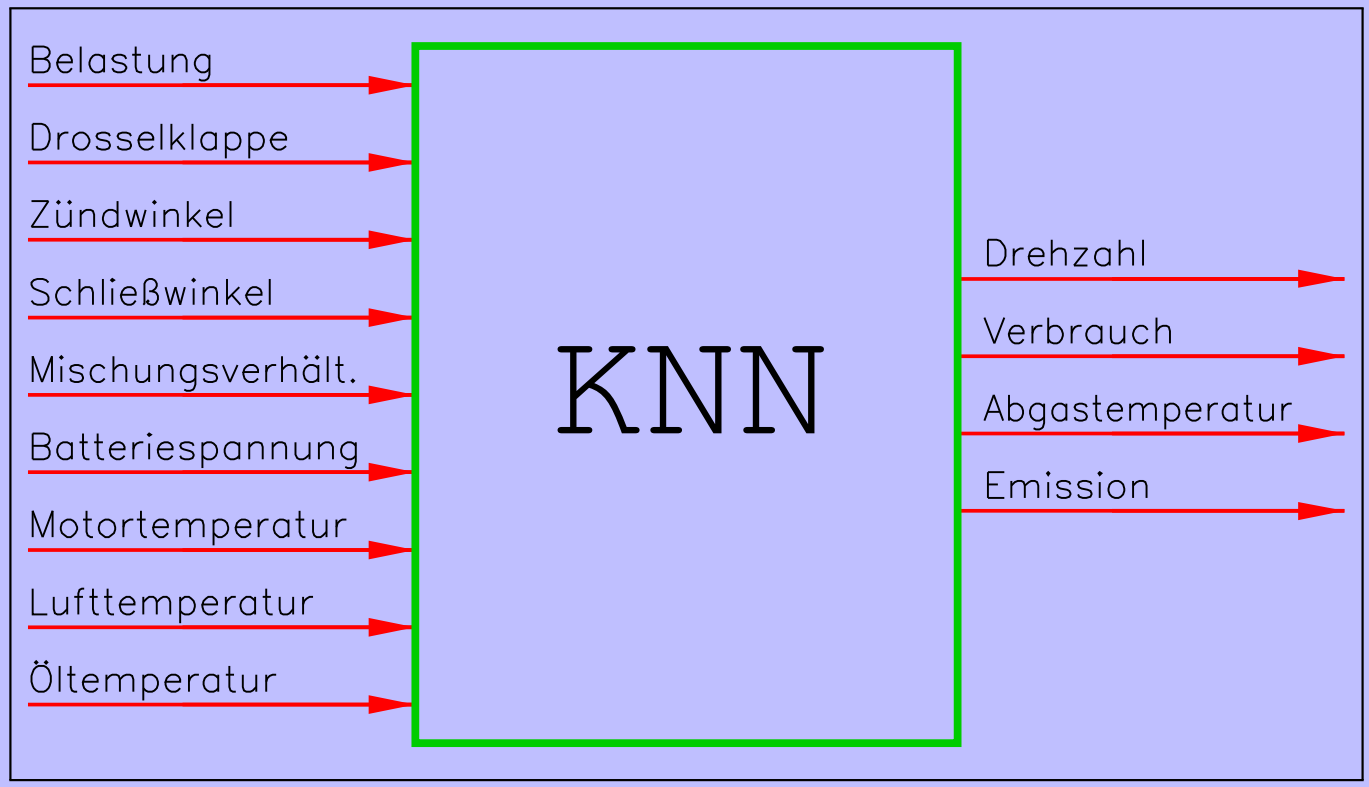
Modellbildung am Beispiel eines Motors



Modellbildung: mathematische Abstraktion

- Motor als Funktion auffassen
- Setzt funktionale Abhängigkeit voraus

Modellbildung am Beispiel eines Motors

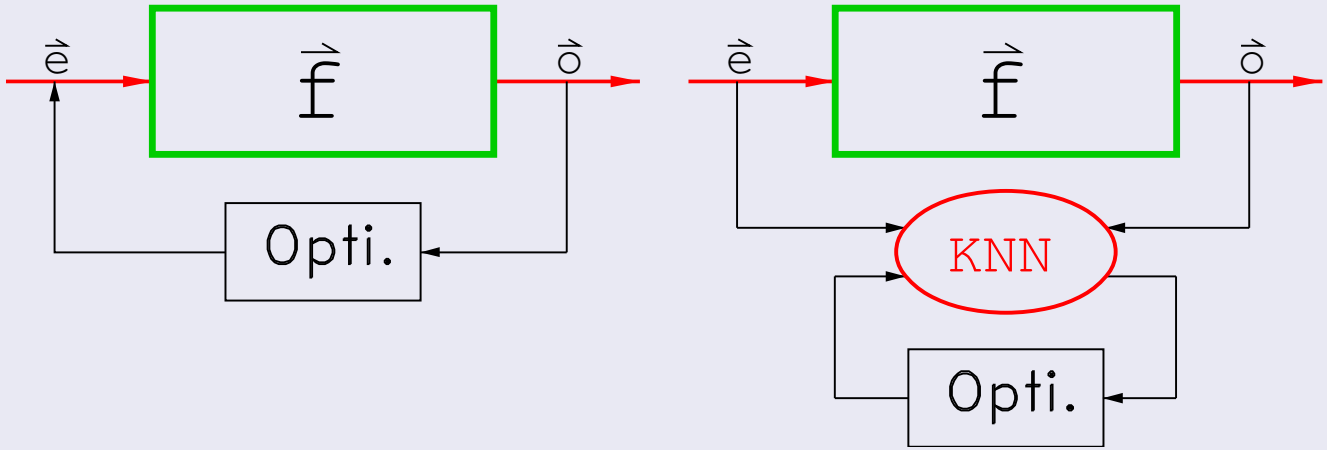


Modellbildung mit neuronalen Netzen

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

Anwendung: Reduktion der Prüfstandläufe

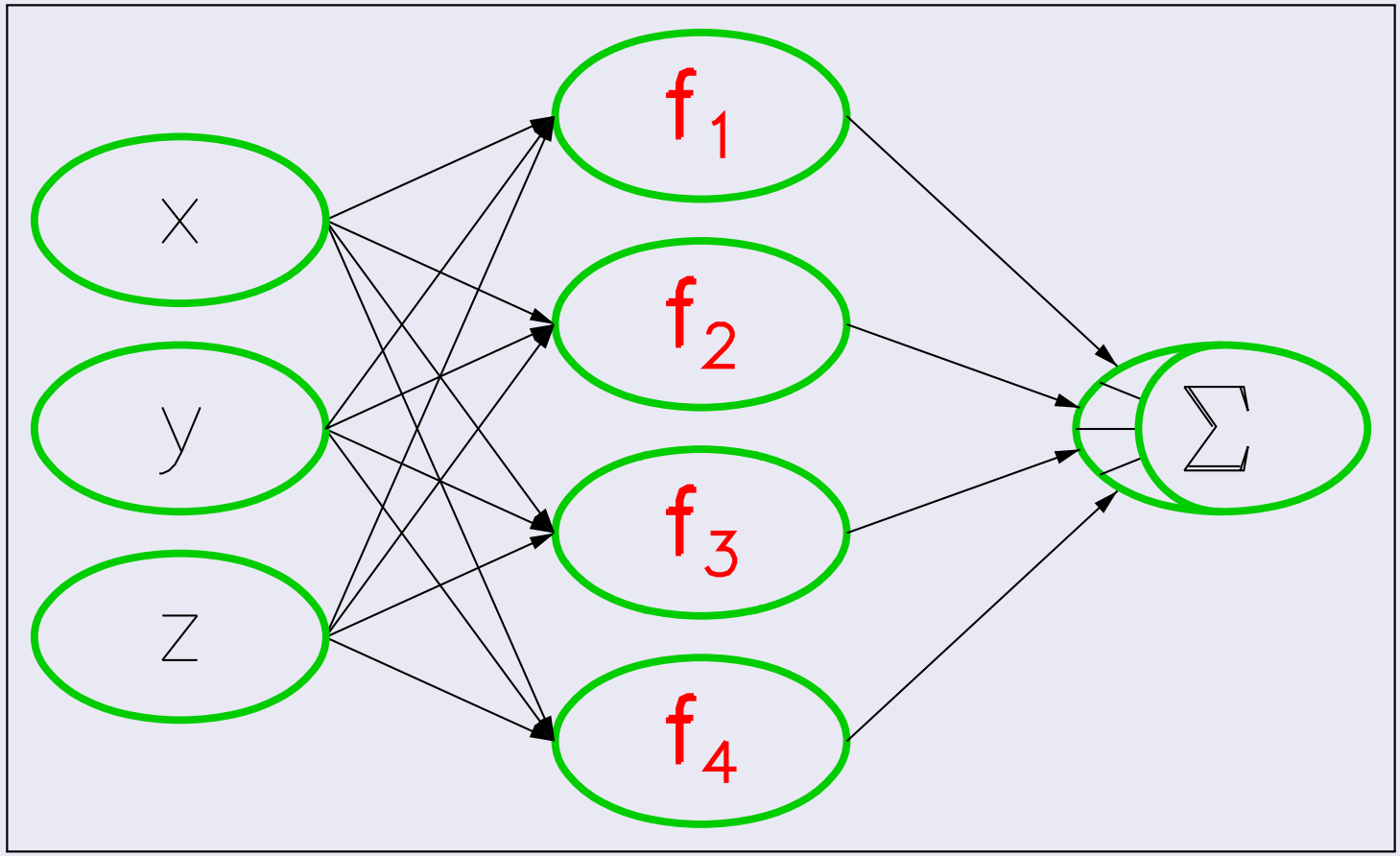
Optimieren der Motorkennlinien mit neuronalen Netzen



- 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

Lineare Ausgleichsrechnung: Beispiel

Ausgleichen mit Polynomen

$$f(x) = \sum_{k=0}^{n} a_k x^k$$

Lineare Ausgleichsrechnung: Beispiel

Ausgleichen mit Polynomen

$$f(x) = \sum_{k=0}^{n} a_k x^k$$

Ausgleichen mit trigonometrischen Polynomen

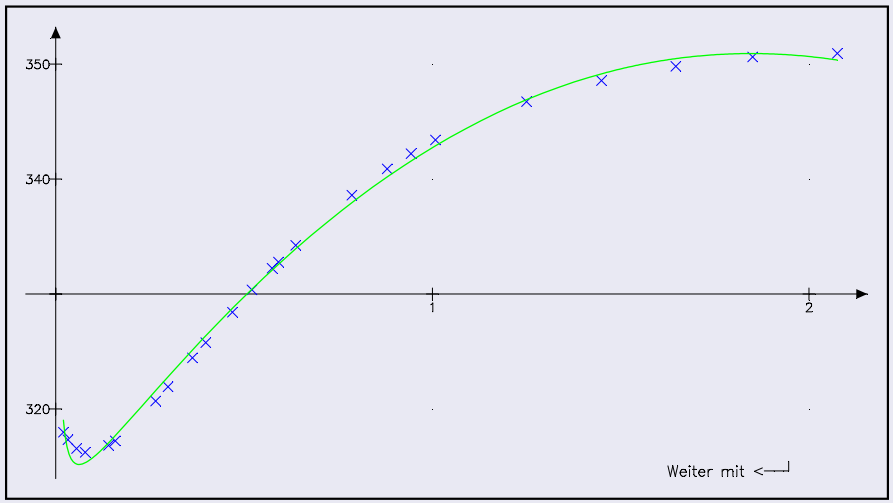
$$f(x) = a_0 + \sum_{k=1}^{n} a_k \cos(\omega k x) + a_k \sin(\omega k x)$$

Lineare Ausgleichsrechnung: Beispiel

Ausgleichen mit Polynomen

Ausgleichen mit trigonometrischen Polynomen

Dehnungs-Spannungs-Verlauf — BMW 1996



$$f(x) = \frac{a_{-1}}{x + .1} + a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

- liefert:

$a_{-1} = 2.2677;$

$a_0 = 297.9072;$

$a_1 = 71.4932;$

$a_2 = -33.7959;$

$a_3 = 5.5016.$

Vergleich Gehirn - Rechner

Gehirn und herkömmliche Rechner

erreichen 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

erreichen 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

erreichen hohe Qualitäten bei der Erfüllung unterschiedlicher Aufgaben:

Gehirn

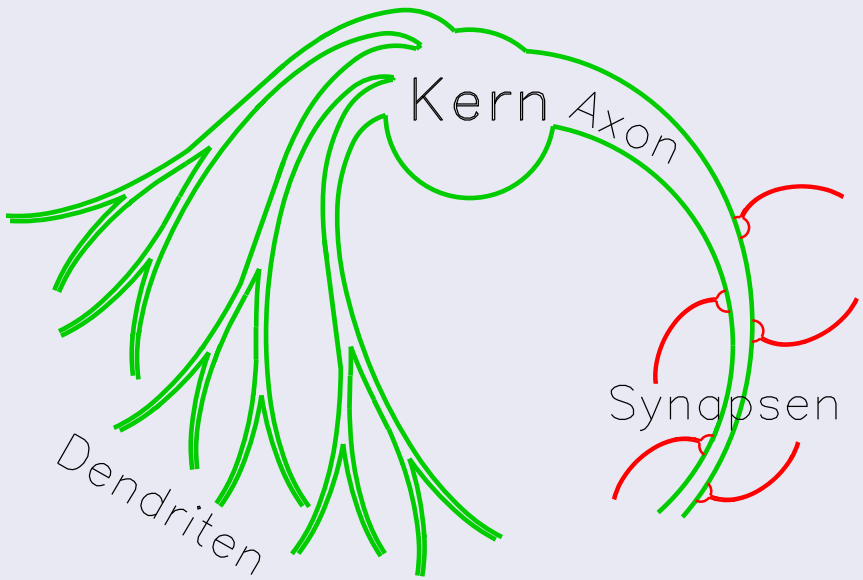
- hohe Parallelität
- Fehlertoleranz
- Mustererkennung
- Verallgemeinern
- Selbstorganisation
- ca. 10^{11} Neuronen, abnehmend auf 10^7
- Jedes Neuron mit ca. 10^4 verbunden

Rechner

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

Das biologische Neuron

Arbeitsweise des Neurons



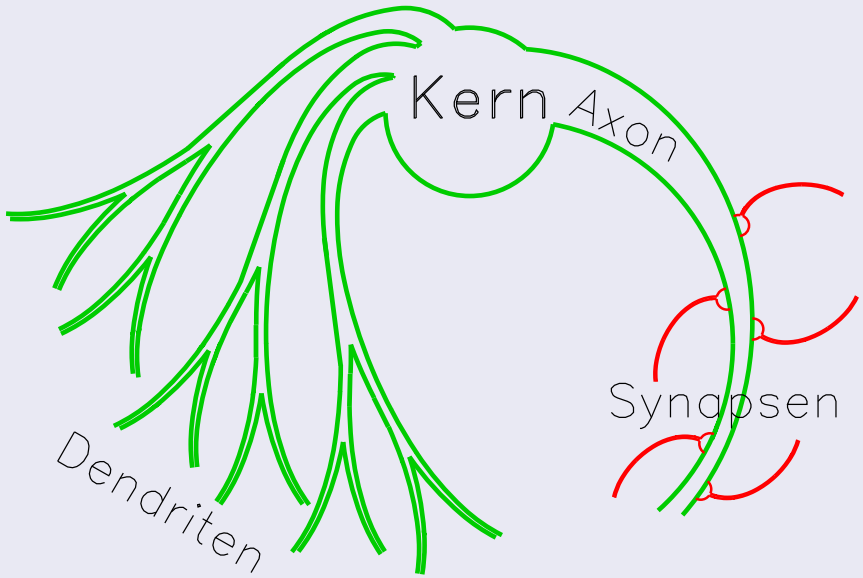
1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf. (Chemische Reaktion)
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls.
5. Gesamtimpuls: Erregung des Neurons.
6. Schwellenwert erreicht:

Lernen:

Synapsen, Dendriten verstärken Verbindung.

Das biologische Neuron

Arbeitsweise des Neurons



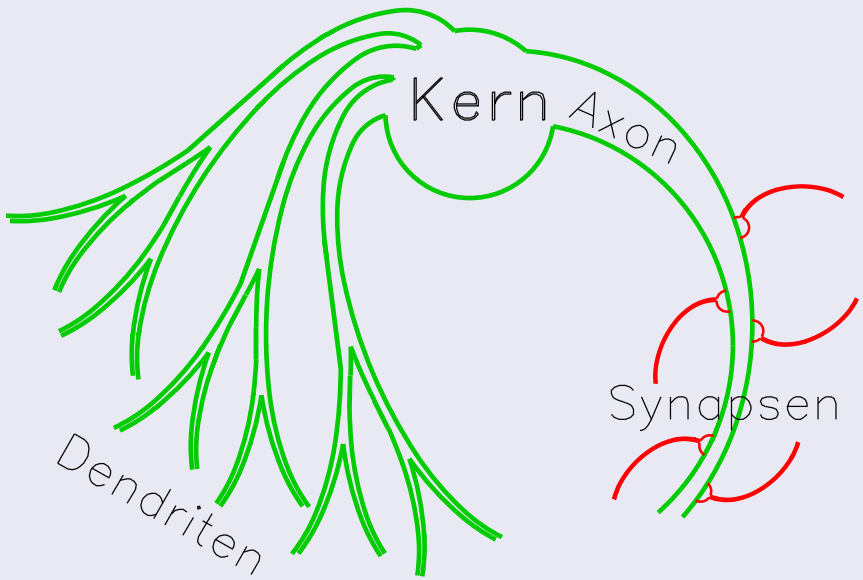
1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf. (Chemische Reaktion)
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls.
5. Gesamtimpuls: Erregung des Neurons.
6. Schwellenwert erreicht:

Lernen:

Synapsen, Dendriten verstärken Verbindung.

Das biologische Neuron

Arbeitsweise des Neurons



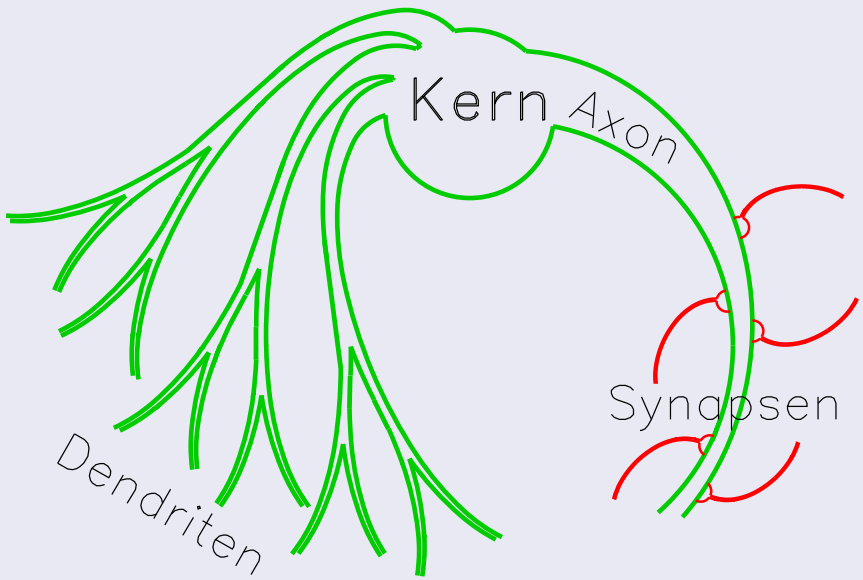
1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf. (Chemische Reaktion)
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls.
5. Gesamtimpuls: Erregung des Neurons.
6. Schwellenwert erreicht:

Lernen:

Synapsen, Dendriten verstärken Verbindung.

Das biologische Neuron

Arbeitsweise des Neurons



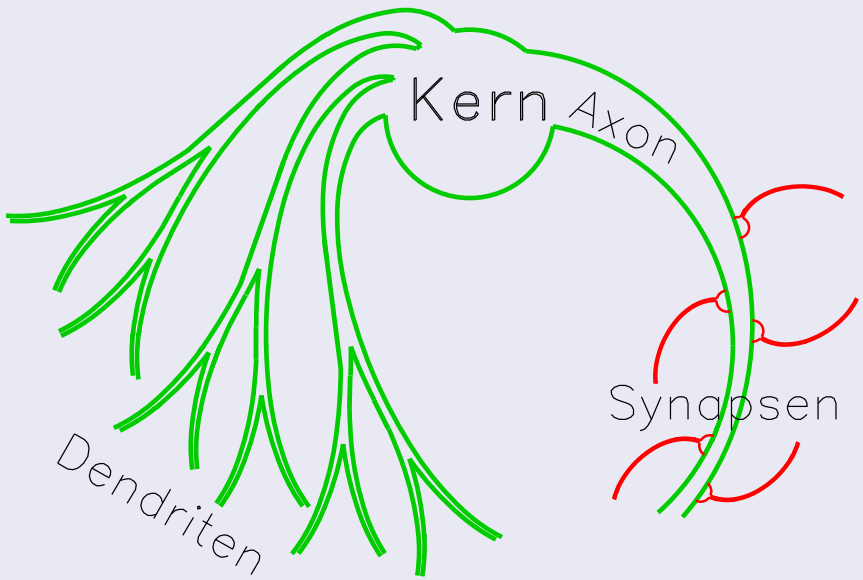
1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf.
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls. (Elektrischer Impuls)
5. Gesamtimpuls: Erregung des Neurons.
6. Schwellenwert erreicht:

Lernen:

Synapsen, Dendriten verstärken Verbindung.

Das biologische Neuron

Arbeitsweise des Neurons



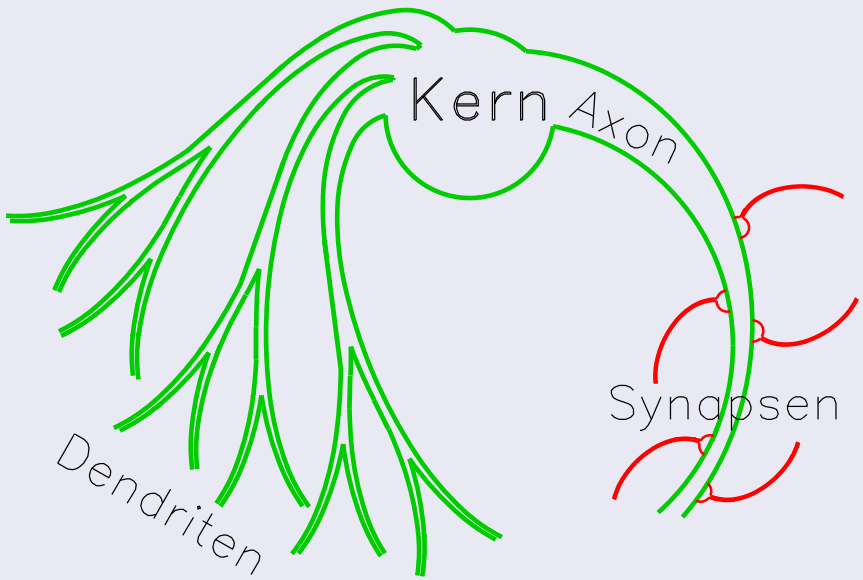
1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf.
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls.
   (Elektrischer Impuls)
5. Gesamtimpuls:
   Erregung des Neurons.
6. Schwellenwert erreicht:

Lernen:

Synapsen, Dendriten verstärken Verbindung.

Das biologische Neuron

Arbeitsweise des Neurons



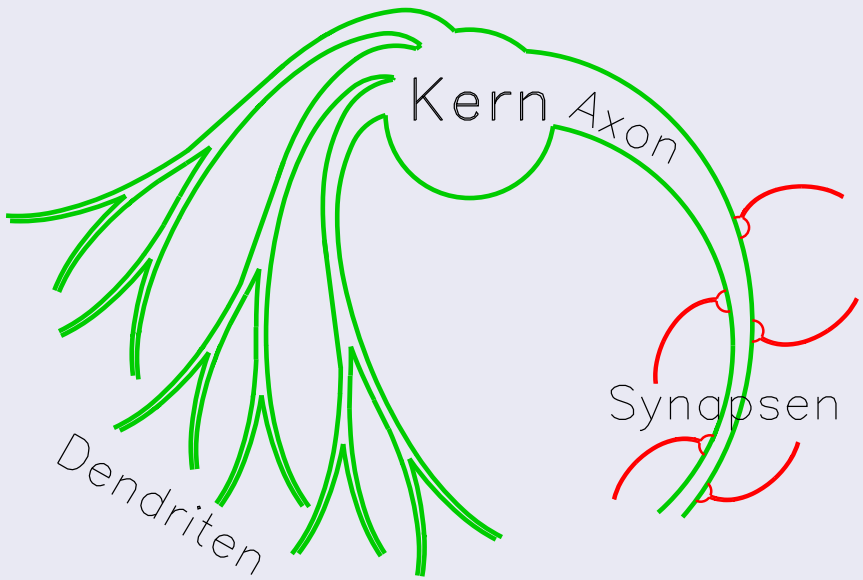
1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf.
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls.
5. Gesamtimpuls:
   Erregung des Neurons.
6. Schwellenwert erreicht:
   Neuron feuert.

Lernen:

Synapsen, Dendriten verstärken Verbindung.

Das biologische Neuron

Arbeitsweise des Neurons

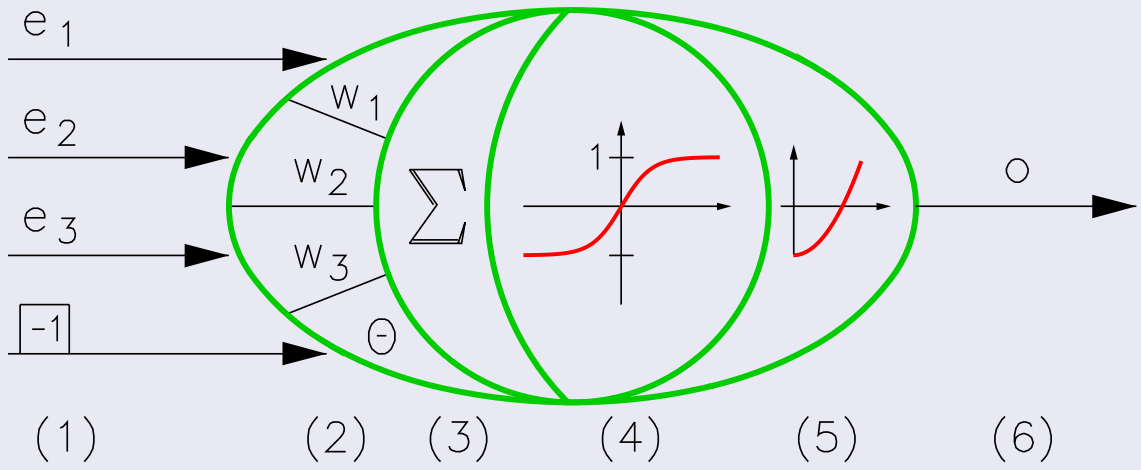


1. Impuls entlang eines Axons.
2. Synapsen nehmen Impuls auf.
3. Dendriten leiten ihn weiter.
4. Zellkern erhält Impuls.
5. Gesamtimpuls:
   Erregung des Neurons.
6. Schwellenwert erreicht:
   Neuron feuert.

Lernen:

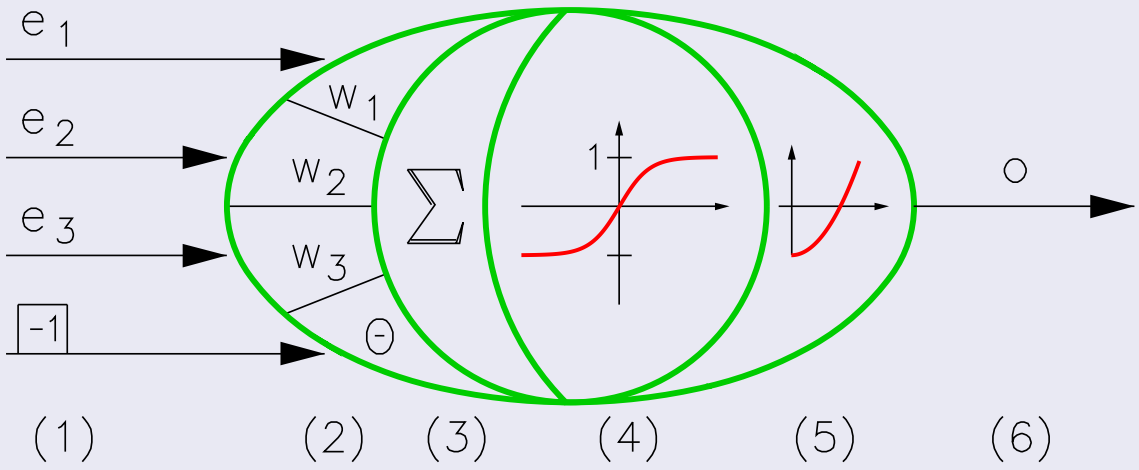
Synapsen, Dendriten verstärken Verbindung.

Das Rechnerneuron



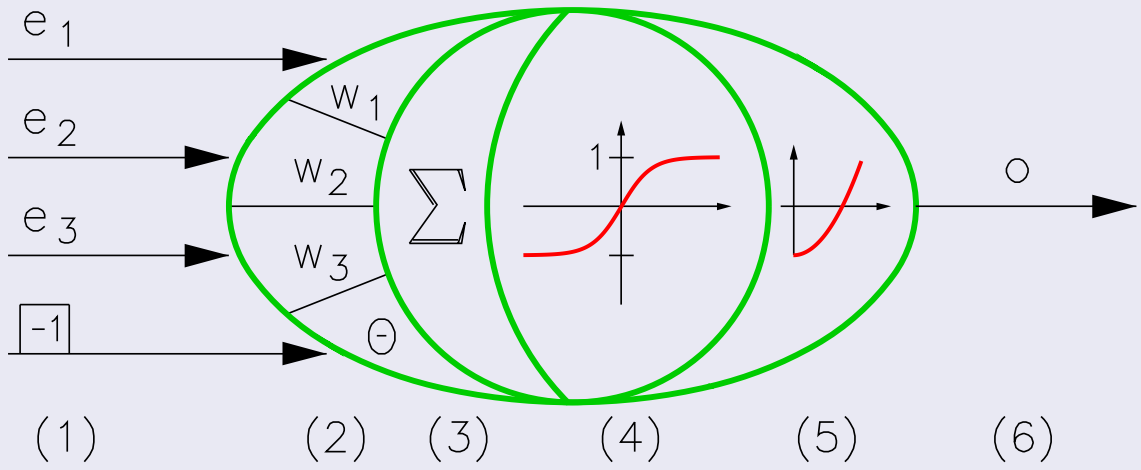
1. Eingabe(vektor) $\vec{e} = (e_1, \ldots, e_n)$, -1 für Schwellenwert
2. Gewichte und Schwellenwert $\vec{w} = (w_1, \ldots, w_n)$ und θ
3. Nettowert (Propagierung) $net = \langle \vec{e}, \vec{w} \rangle - \theta = \sum_{i=1}^{n} e_i w_i - \theta$
4. Aktivierungsfunktion, Aktivität $a = a(\langle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

Das Rechnerneuron



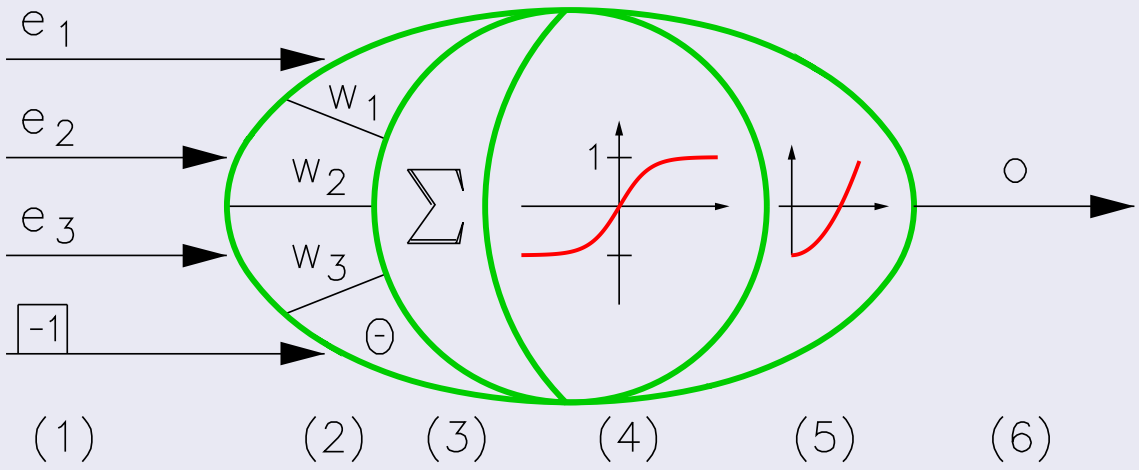
1. Eingabe(vektor) $\vec{e} = (e_1, \ldots, e_n)$, -1 für Schwellenwert
2. Gewichte und Schwellenwert $\vec{w} = (w_1, \ldots, w_n)$ und θ
3. Nettowert (Propagierung) $net = \langle \vec{e}, \vec{w} \rangle - \theta = \sum_{i=1}^{n} e_i w_i - \theta$
4. Aktivierungsfunktion, Aktivität $a = a(\langle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

Das Rechnerneuron



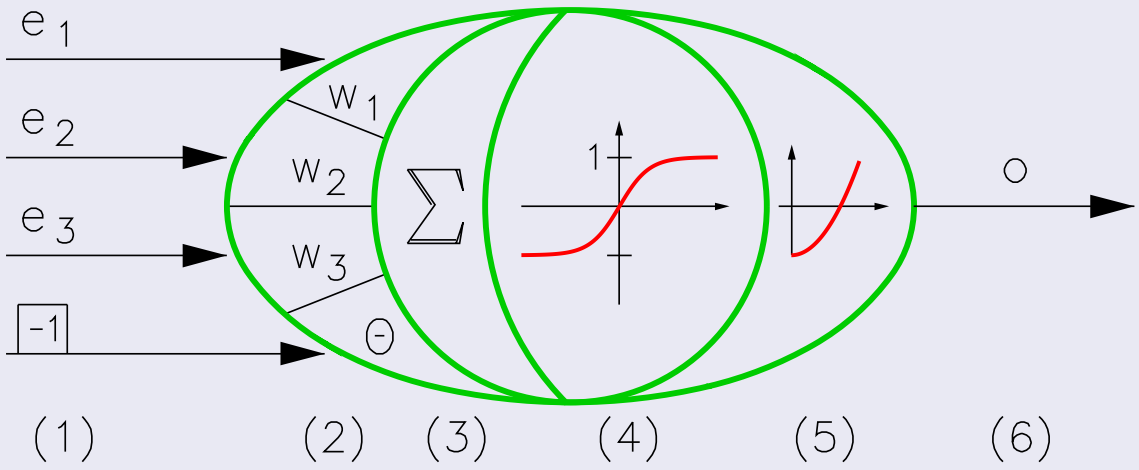
1. Eingabe(vektor) $\vec{e} = (e_1, \ldots, e_n)$, -1 für Schwellenwert
2. Gewichte und Schwellenwert $\vec{w} = (w_1, \ldots, w_n)$ und θ
3. Nettowert (Propagierung) $net = \langle \vec{e}, \vec{w} \rangle - \theta = \sum_{i=1}^{n} e_i w_i - \theta$
4. Aktivierungsfunktion, Aktivität $a = a(\langle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

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 $o = o(a(\langle \vec{w}, \vec{e} \rangle - \theta))$

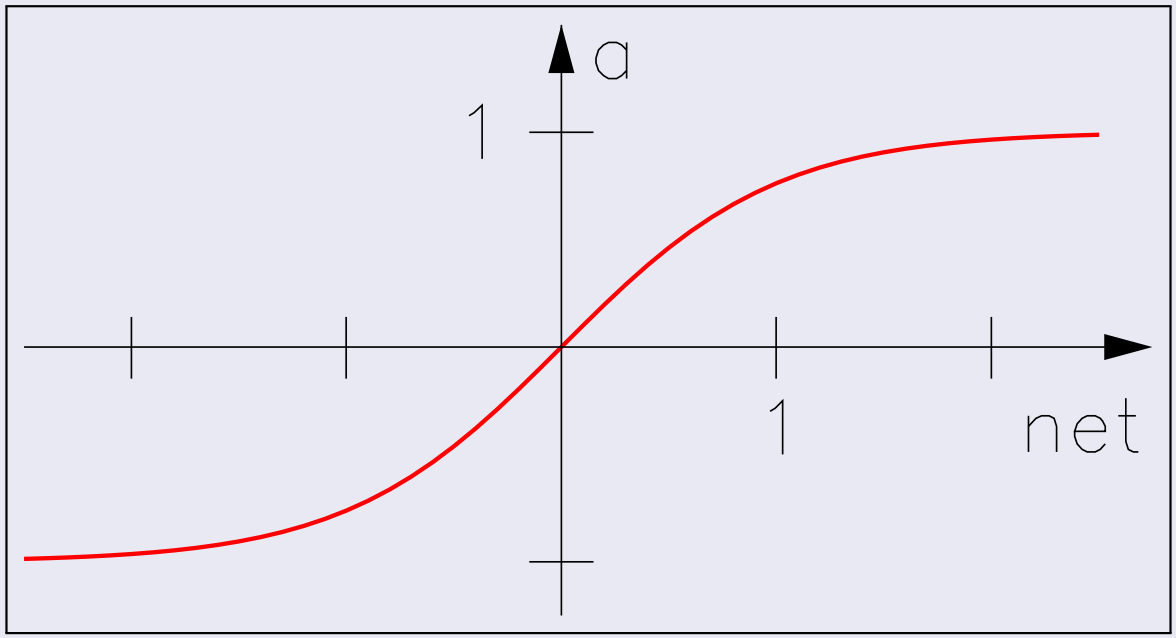
Das Rechnerneuron



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 $o = o(a(\langle \vec{w}, \vec{e} \rangle - \theta))$

Die sigmoide Aktivierungsfunktion

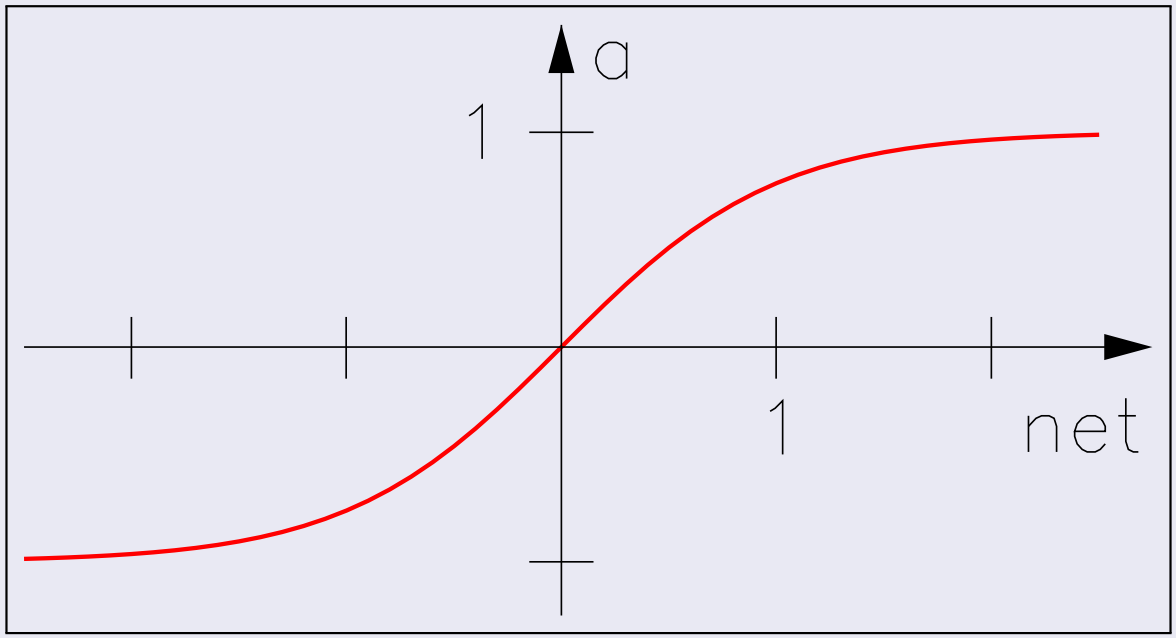
$a(x) = \tanh(gx) : \mathbb{R} \to (-1, 1)$



Ableitung: $a'(x) = g(-\tanh(gx)^2) = g(1 - a(x)^2)\,; \quad a'(0) = g$

Die sigmoide Aktivierungsfunktion

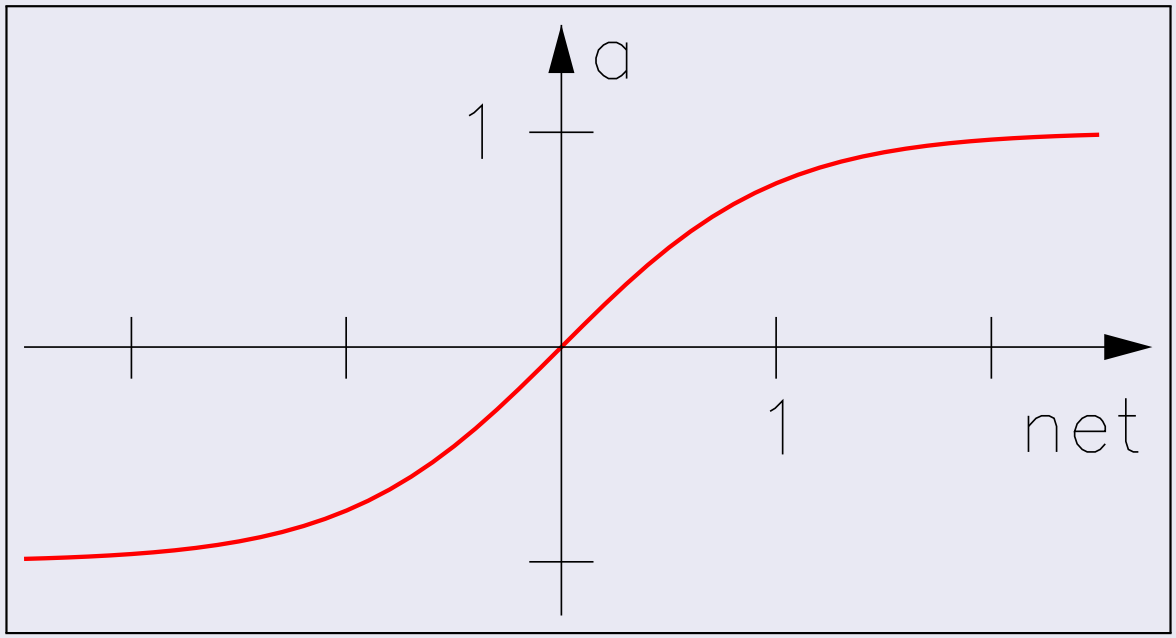
$a(x) = \tanh(gx) : \mathbb{R} \to (-1, 1)$



Ableitung: $a'(x) = g(-\tanh(gx)^2) = g(1 - a(x)^2)\,; \quad a'(0) = g$

Die sigmoide Aktivierungsfunktion

$a(x) = \tanh(gx) : \mathbb{R} \to (-1, 1)$



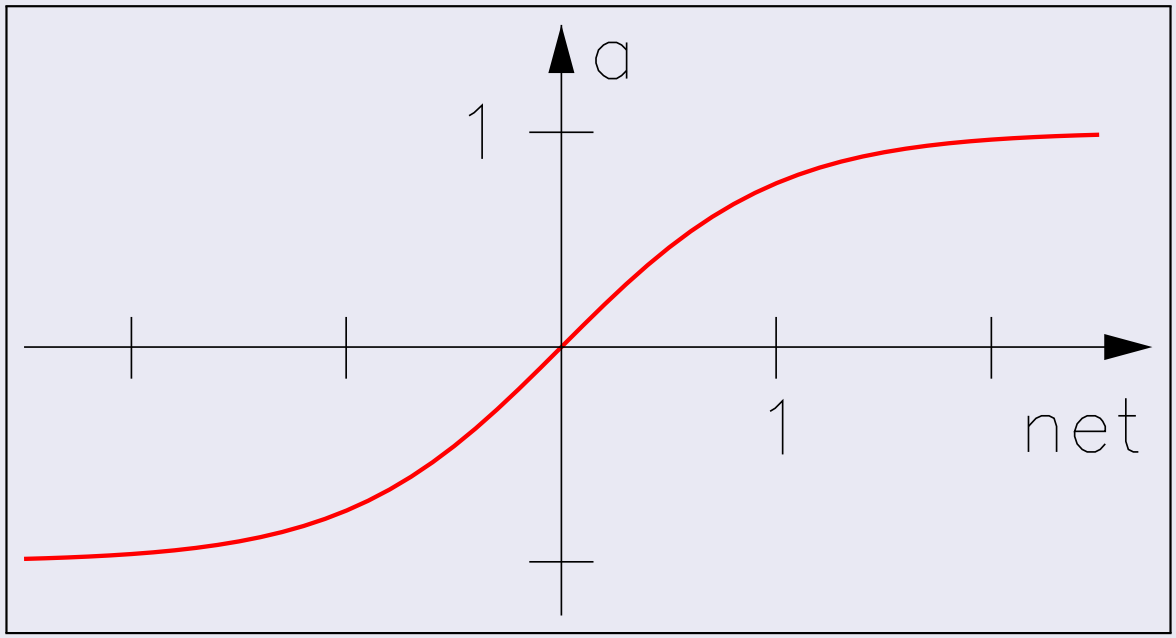
Ableitung: $a'(x) = g(-\tanh(gx)^2) = g(1 - a(x)^2)\,; \quad a'(0) = g$

Alternative Aktivierungsfunktionen:

1. $a(x) = \frac{1}{1+e^{-gx}} : \mathbb{R} \to (0, 1),\ a'(x) = g(1 - a(x)) \cdot a(x)\,; \quad a'(0) = \frac{g}{4}$
2. Parabelstücke für einfache Hardware-Implementierung (Carmen Stumm)

Die sigmoide Aktivierungsfunktion

$a(x) = \tanh(gx) : \mathbb{R} \to (-1, 1)$



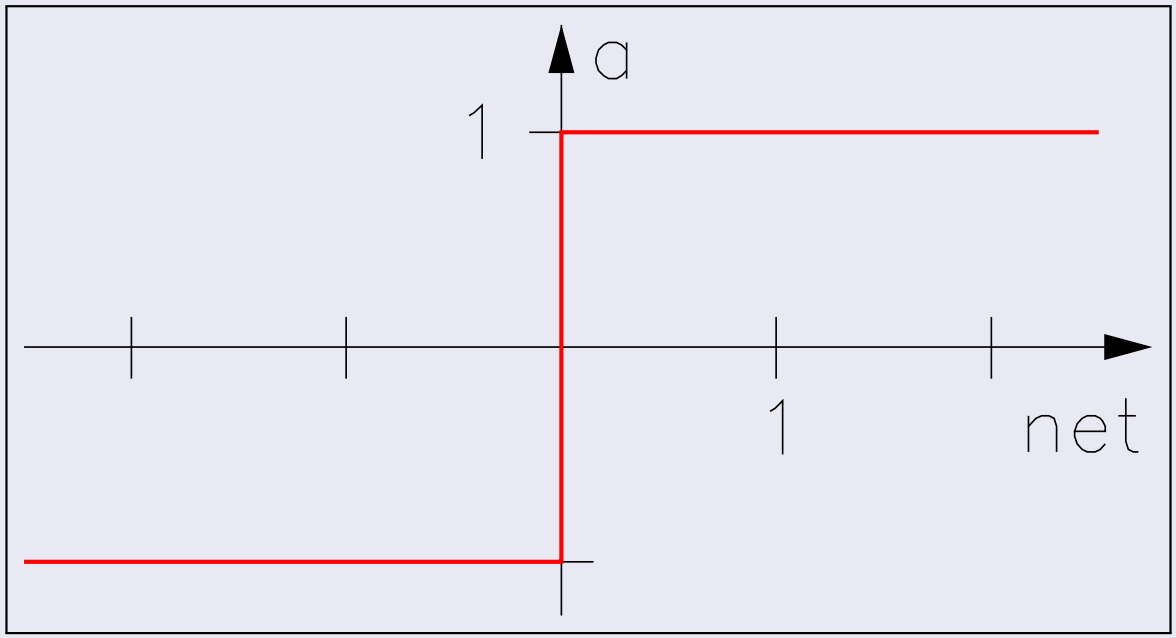
Ableitung: $a'(x) = g(-\tanh(gx)^2) = g(1 - a(x)^2)\,; \quad a'(0) = g$

Alternative Aktivierungsfunktionen:

1. $a(x) = \frac{1}{1+e^{-gx}} : \mathbb{R} \to (0, 1),\ a'(x) = g(1 - a(x)) \cdot a(x)\,; \quad a'(0) = \frac{g}{4}$
2. Parabelstücke für einfache Hardware-Implementierung (Carmen Stumm)

Bipolare und binäre Schwellenwertfunktion

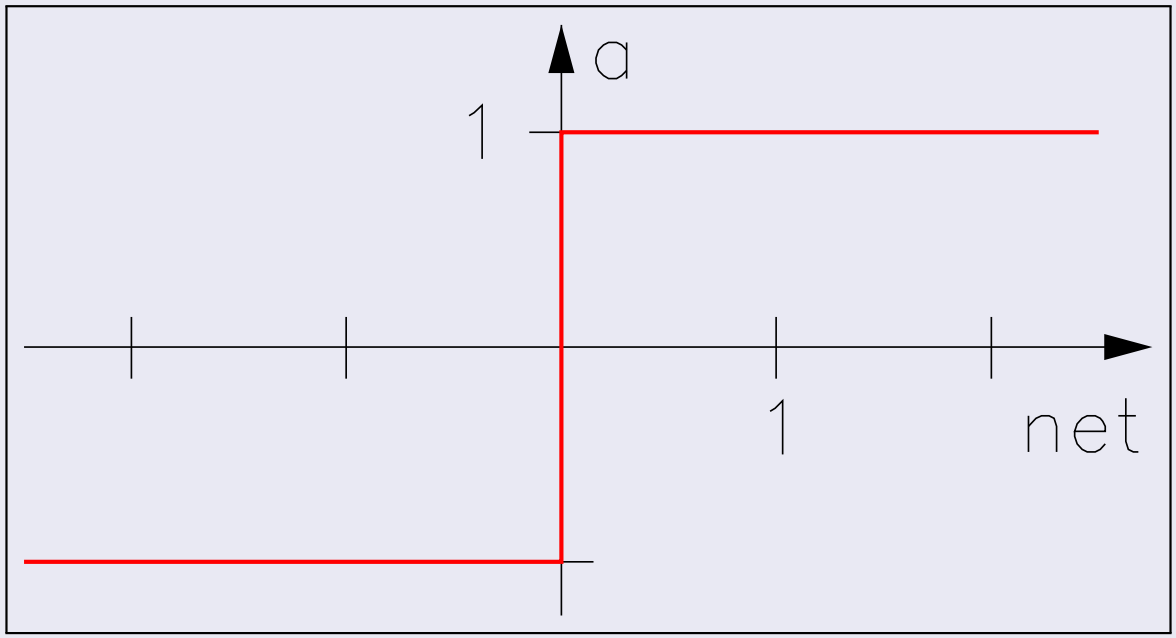
$a(x) = sign(x) : \mathbb{R} \rightarrow [-1, 1]$



Aktivität: $a(x) = \text{sign}(\langle \vec{e}, \vec{w} \rangle - \theta) = 2(\langle \vec{e}, \vec{w} \rangle - \theta \geq 0) - 1$

Bipolare und binäre Schwellenwertfunktion

$a(x) = sign(x) : \mathbb{R} \rightarrow [-1, 1]$



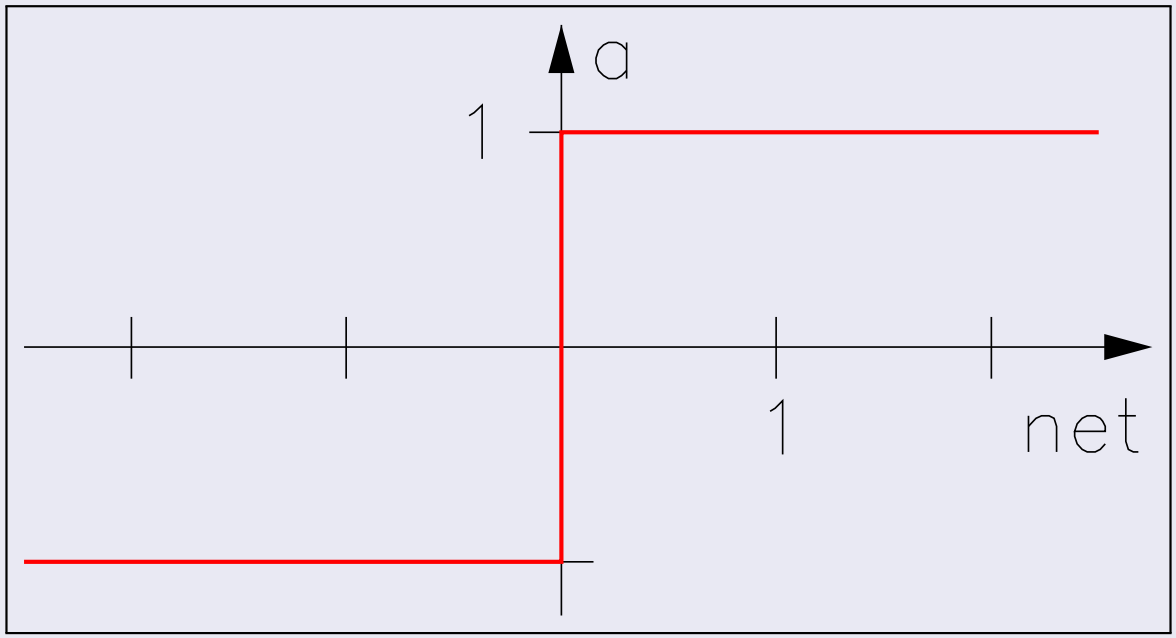
Aktivität: $a(x) = \text{sign}(\langle \vec{e}, \vec{w} \rangle - \theta) = 2(\langle \vec{e}, \vec{w} \rangle - \theta \geq 0) - 1$

Alternative Aktivierungsfunktionen:

$a(x) = (x \geq 0) = \frac{1+\text{sign}(x)}{2} = \langle x - 0 \rangle^0 : \mathbb{R} \rightarrow [0, 1]$

Bipolare und binäre Schwellenwertfunktion

$a(x) = sign(x) : \mathbb{R} \rightarrow [-1, 1]$



Aktivität: $a(x) = \text{sign}(\langle \vec{e}, \vec{w} \rangle - \theta) = 2(\langle \vec{e}, \vec{w} \rangle - \theta \geq 0) - 1$

Lineares Trennen

Schwellenwertelement trennt Eingabedaten linear, logisch verknüpfte Schwellenwertelemente legen Simplex fest.

Lineare Trennung

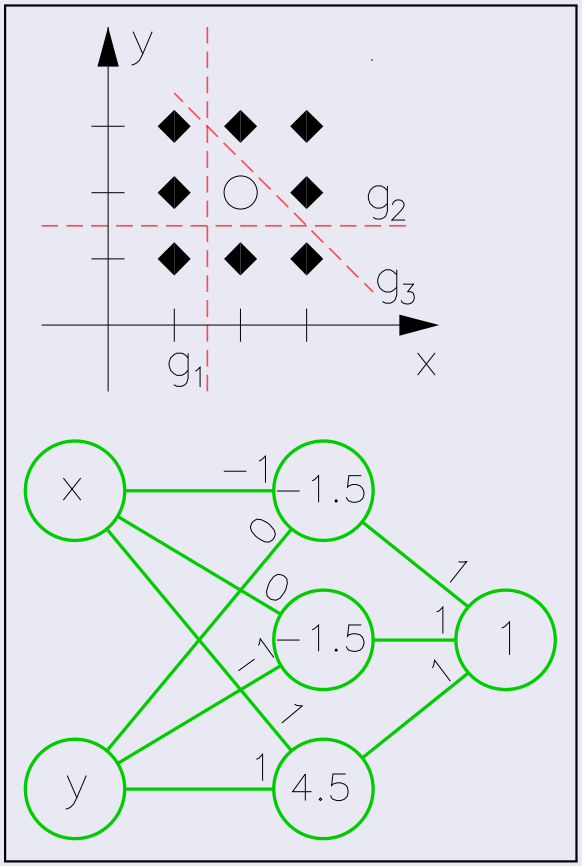
Verdeckte Neuronen trennen linear

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

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

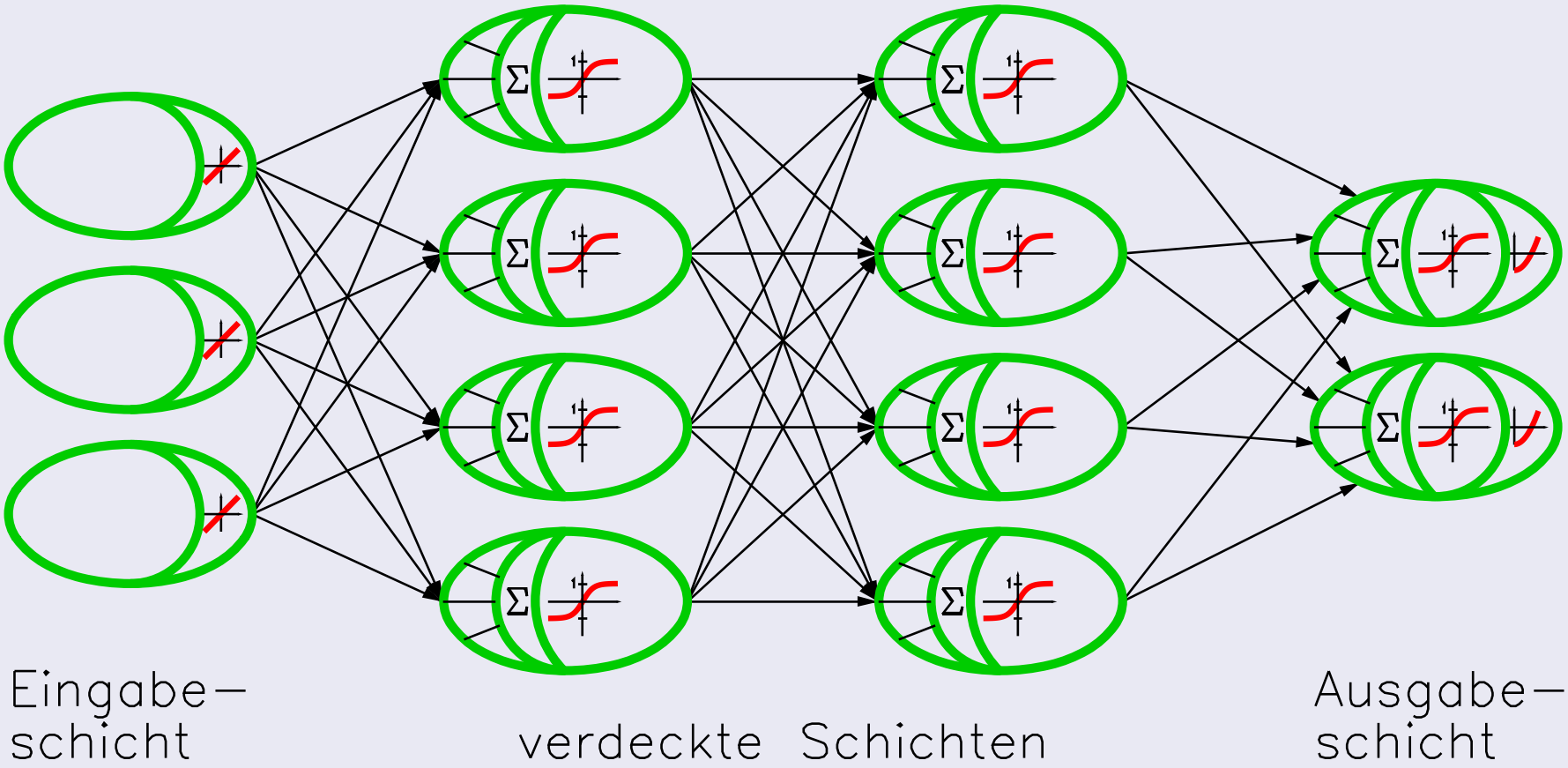
$$\begin{pmatrix} x \\ y \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \geq \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 4.5$$

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.



Schichtenmodell

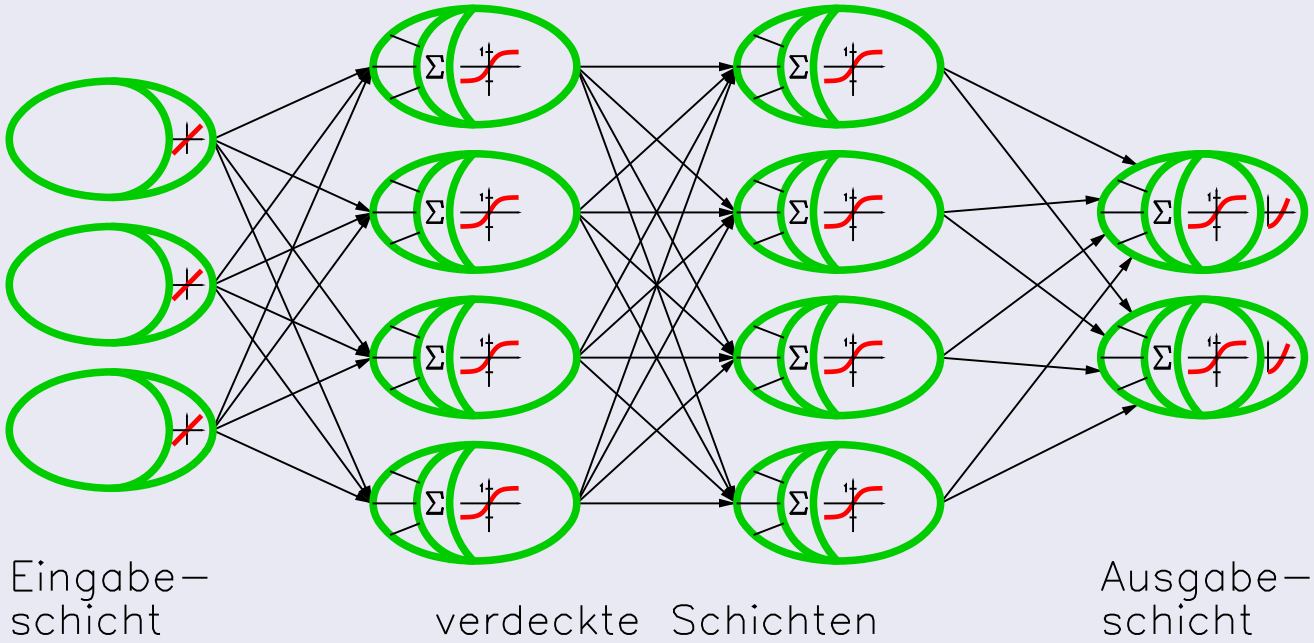
Schichtenmodell mit Topologie 3-4-4-2



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←↑ρbpan◊aus←net←bpanρ¨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)

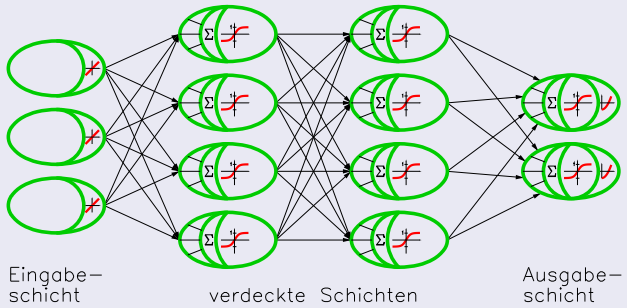
Jede Funktion kann durch Netze mit einer verdeckten Schicht angenähert werden.

Schichtenmodell

Eingabeschicht

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

Schichtenmodell

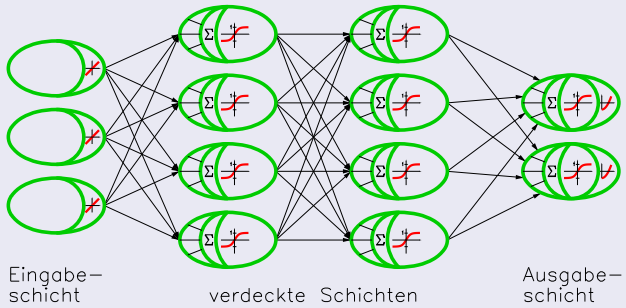


Schichtenmodell

Eingabeschicht

- analoge Eingabe:
  Lineare Transformation auf [-1; 1]
- diskrete Eingabe:
  Für jeden 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:

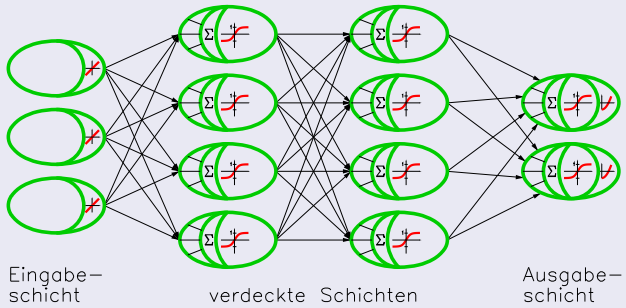
$$f(x) = \left\{ \begin{array}{ccc} [m,\ M] & \rightarrow & [-0.6, 0.6] \\ x & \mapsto & -0.6 + 1.2 \left(\frac{x-m}{M-m} \right)^s \ ; \quad s > 0 \end{array} \right\}$$

Schichtenmodell

Eingabeschicht

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

Schichtenmodell



Ausgabeschicht mit logarithmisch sigmoider Aktivierungsfunktion

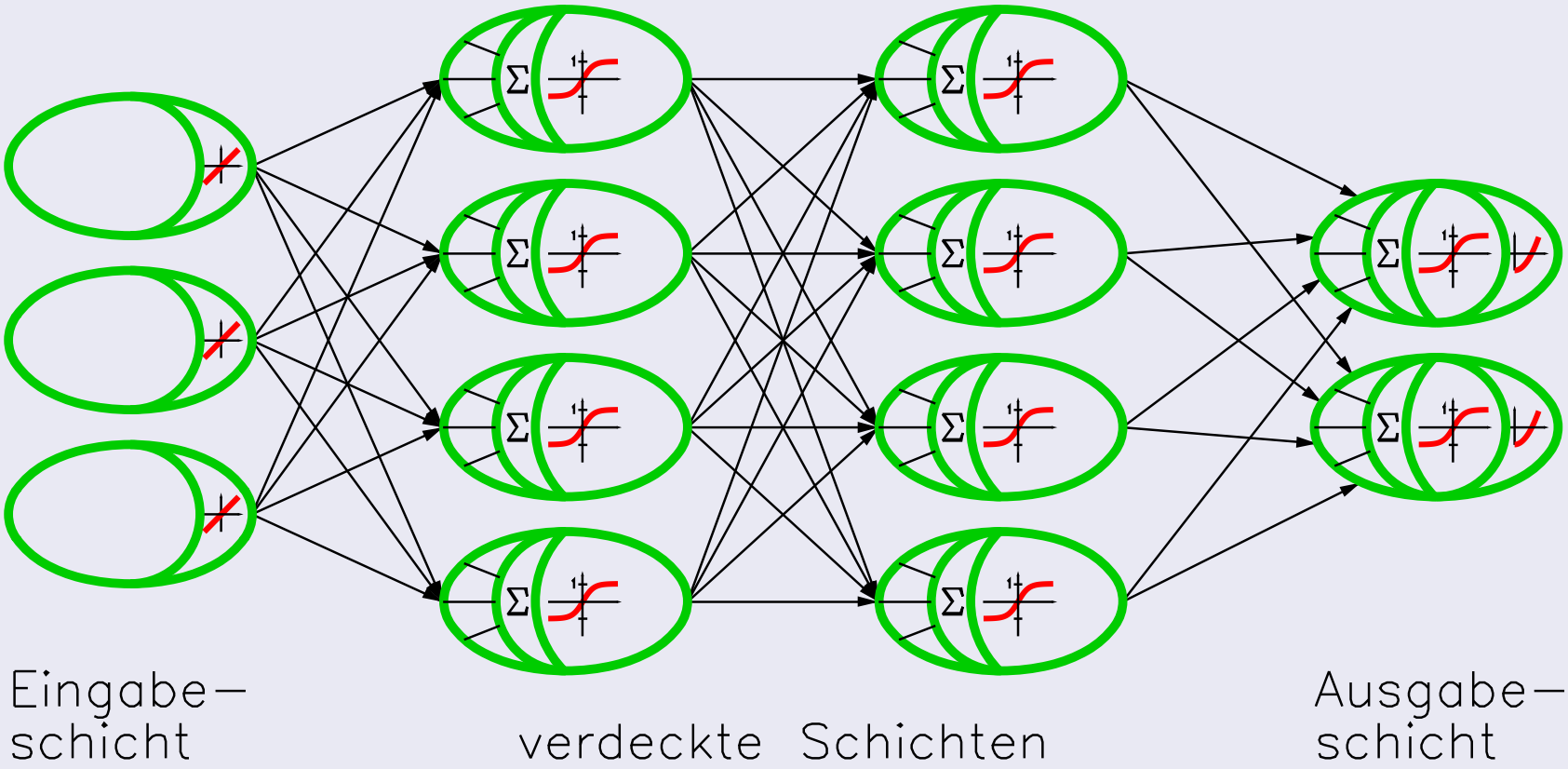
- Zielaktivitäten gleichverteilt im Intervall [0.2, 0.8]!
- Umkehrung der Ausgabefunktion kann sein:

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

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

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

Lernen in vorwärtsgerichteten Schichtennetzen

Ziel

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

Berechnung

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

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

Delta-Regel (Gradient descent)

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

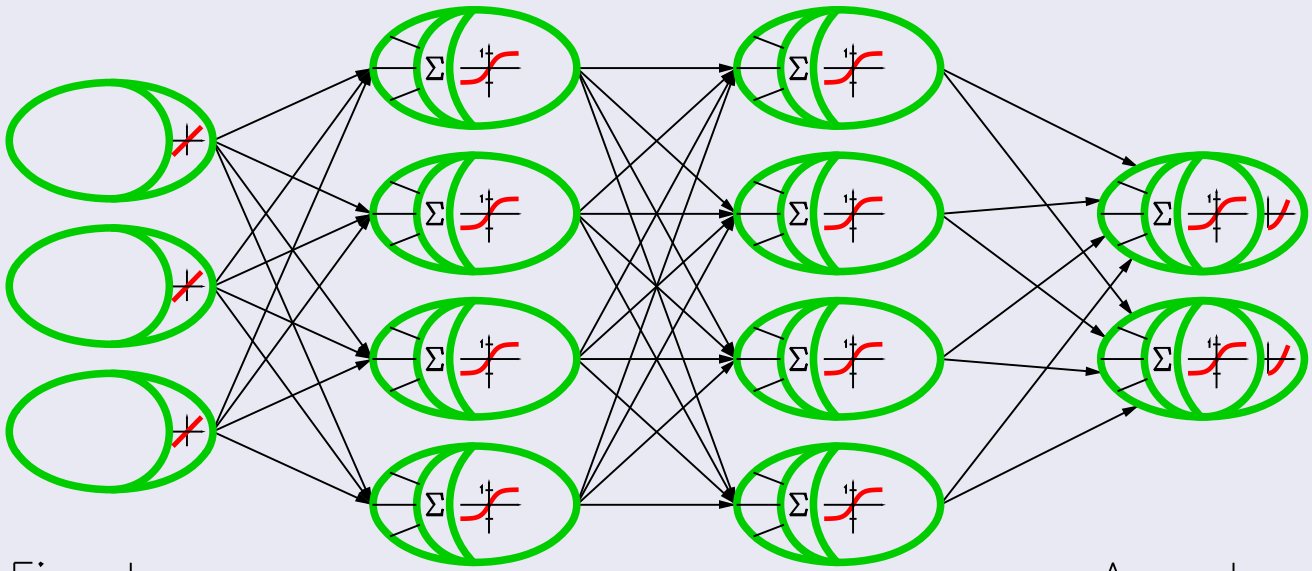
σ fallend, z.B. von 0.9 auf 0.1, μ steigend, z.B. $\mu = 1 - \sigma$.

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

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

Lernen in vorwärtsgerichteten Schichtennetzen

Lernverfahren nach Levenberg-Marquardt

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

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

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

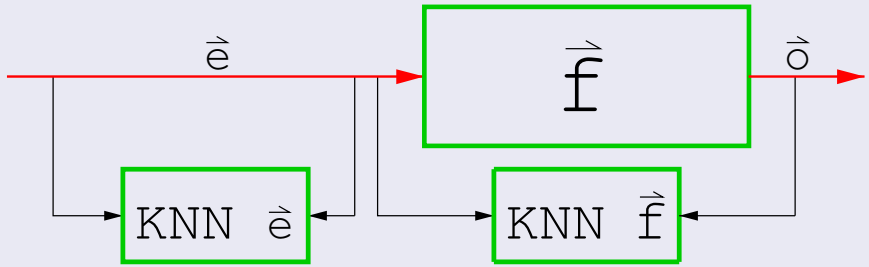
Systemspezialist

- Beurteilung von Trendaussagen

Fehler in den Kontrolldaten

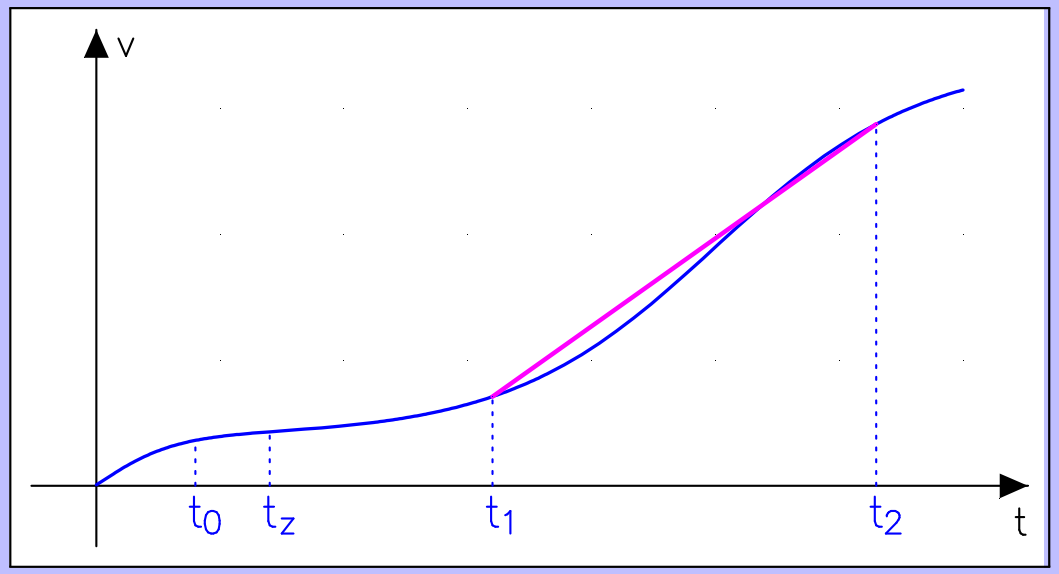
- 20%-40% der Lerndaten
- maximaler und durchschnittlicher Fehler
- Standardabweichung

Autokorrelation



Unfallschwere

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



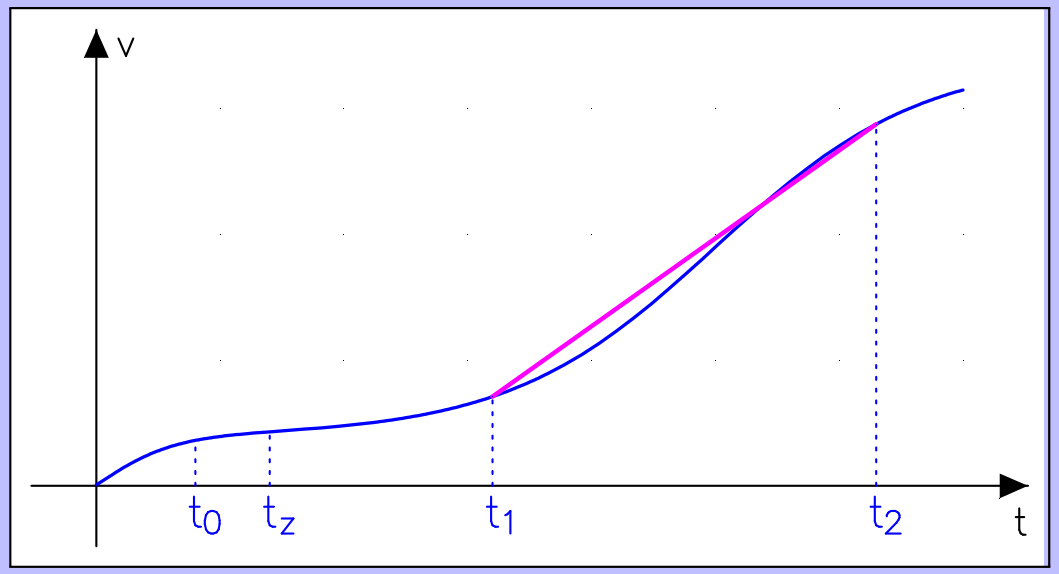
- t_0: Entscheidung
- t_Z: Zündzeitpunkt Airbag ($t_1 - t_Z \approx 30\,\mathrm{ms}$)
- t_1: Beginn kritischer Vorverlagerung
- t_2: Beschleunigung nimmt ab

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

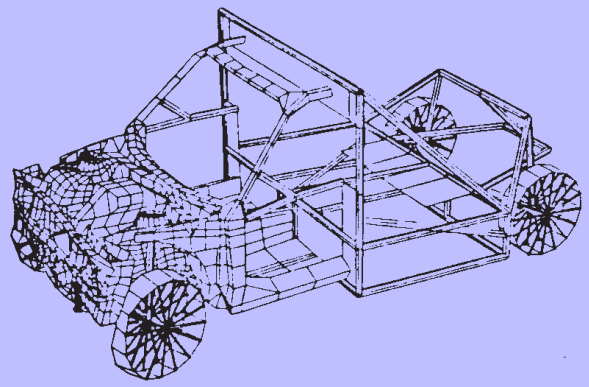
Grundlage

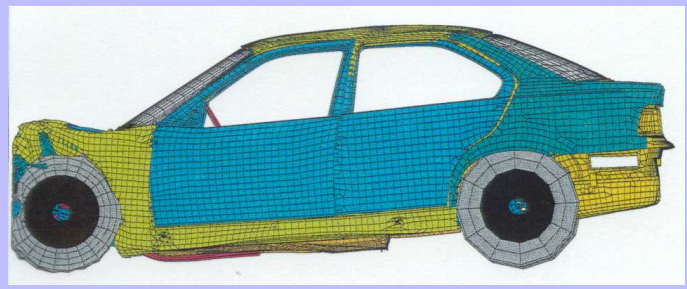
Daten aus Parametervariationen mit Monte-Carlo Methode:

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

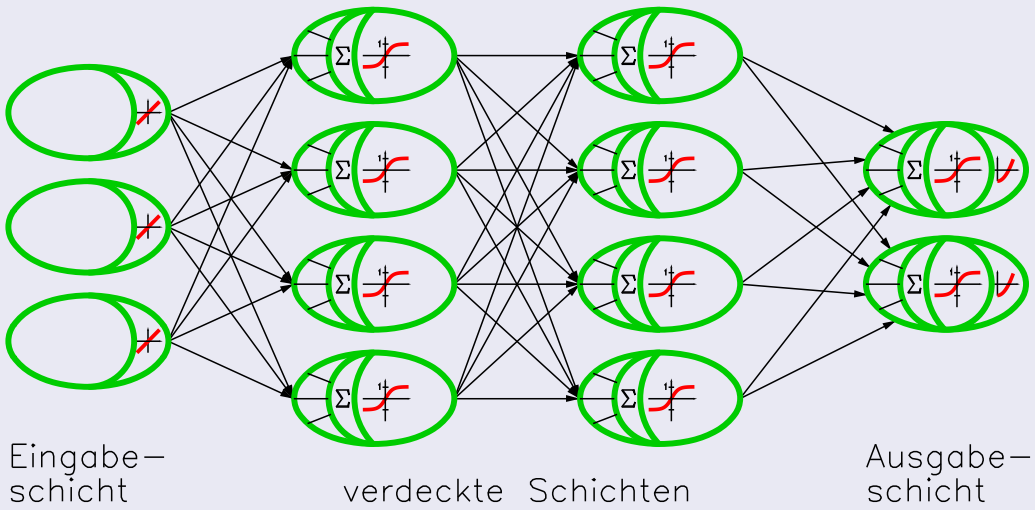




höhere Rechnerleistung!

Einsatz neuronaler Netze

3- oder 4-schichtige Netze



Eingaben

- Beschleunigungen, Geschwindigkeiten, Verschiebungen
- Maximale, gemittelte Werte

Ausgabe

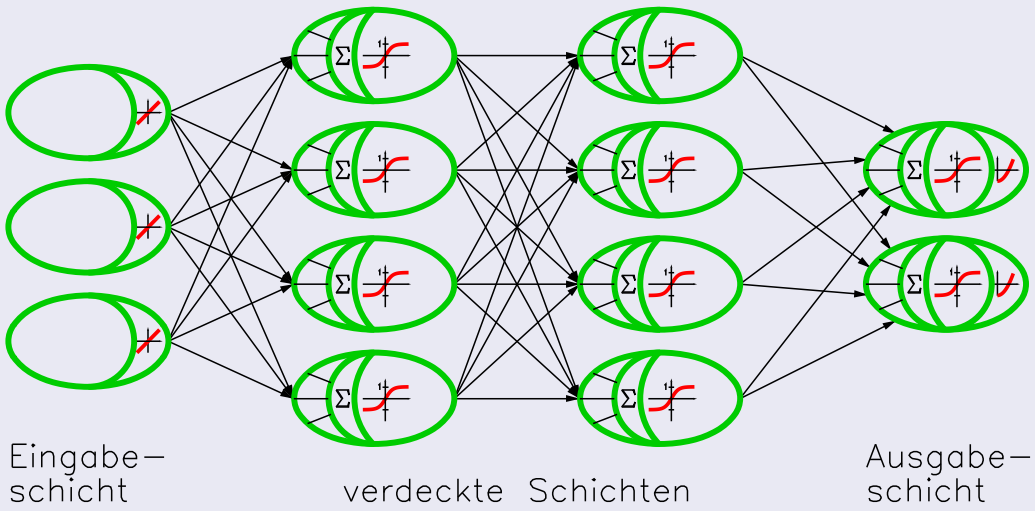
1. Geschwindigkeit
2. Gemittelte Beschleunigung (Wirkung auf Insassen)

Lernen:

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

Einsatz neuronaler Netze

3- oder 4-schichtige Netze



Eingaben

- Beschleunigungen, Geschwindigkeiten, Verschiebungen
- Maximale, gemittelte Werte

Ausgabe

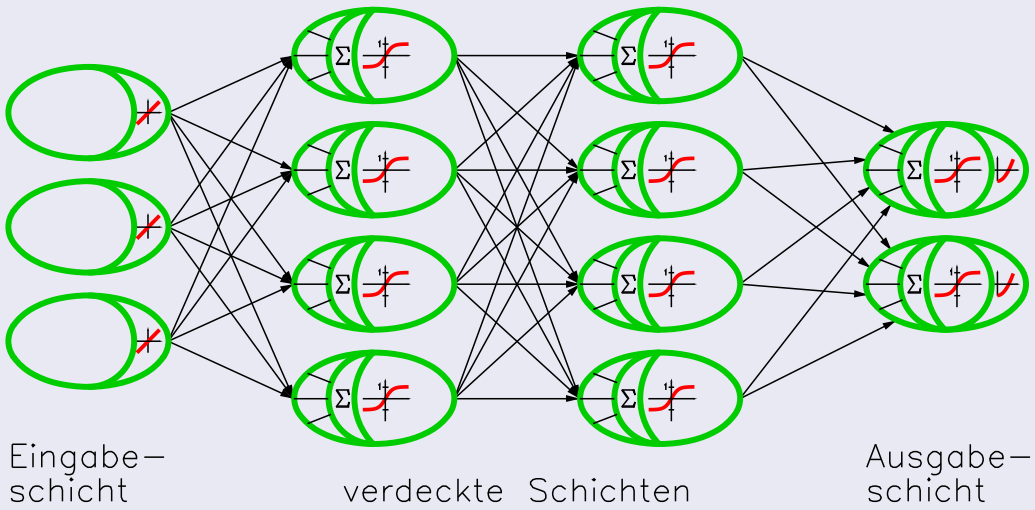
1. Geschwindigkeit
2. Gemittelte Beschleunigung (Wirkung auf Insassen)

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)

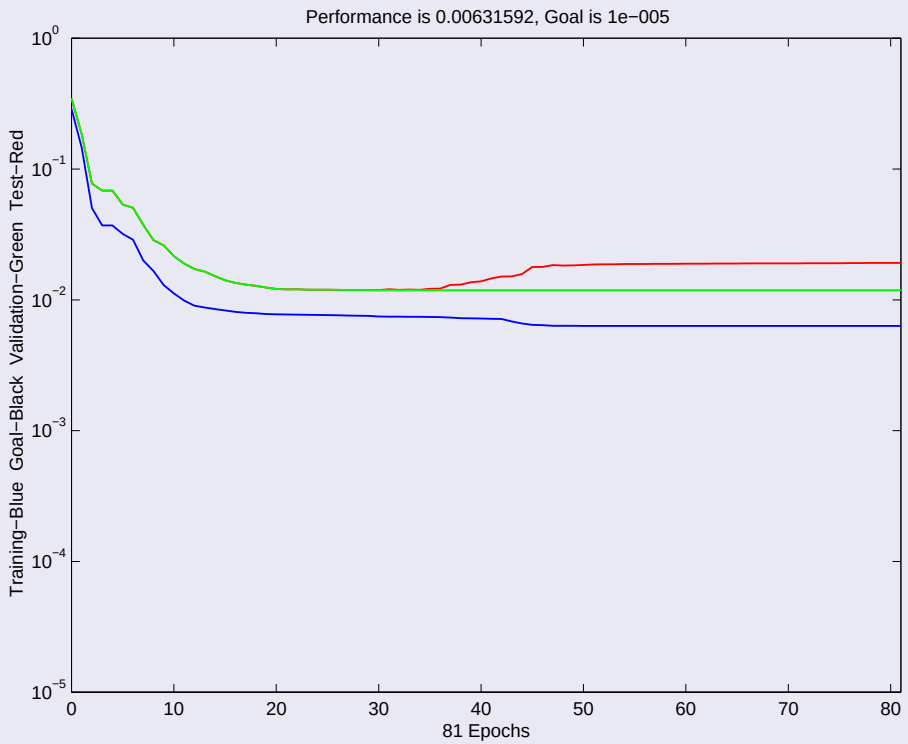
Lernen:

- 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

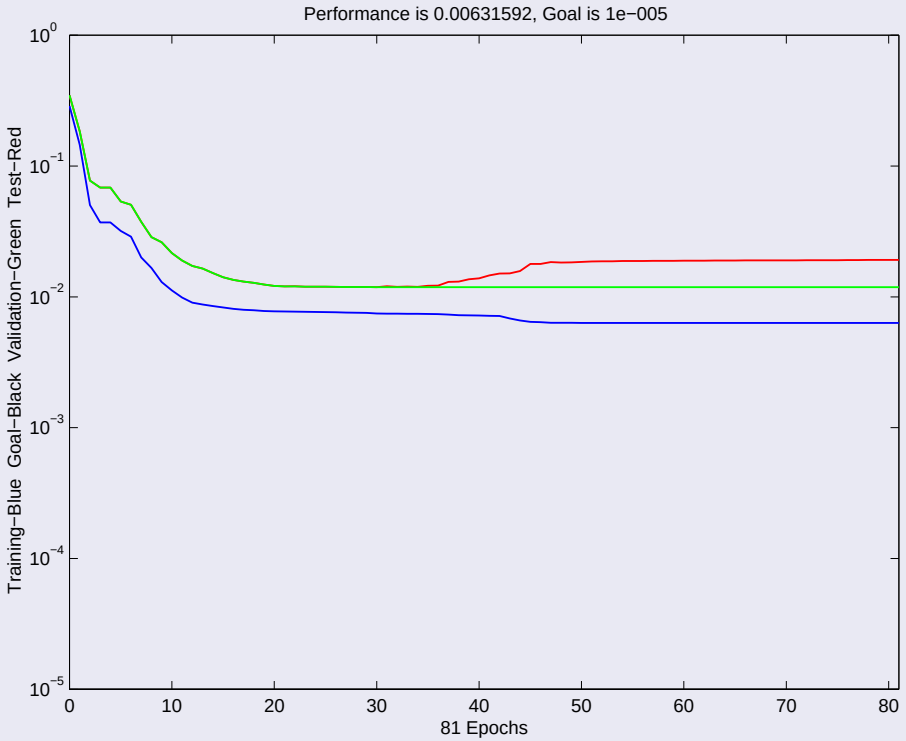
Mittlerer Fehler beim Lernen



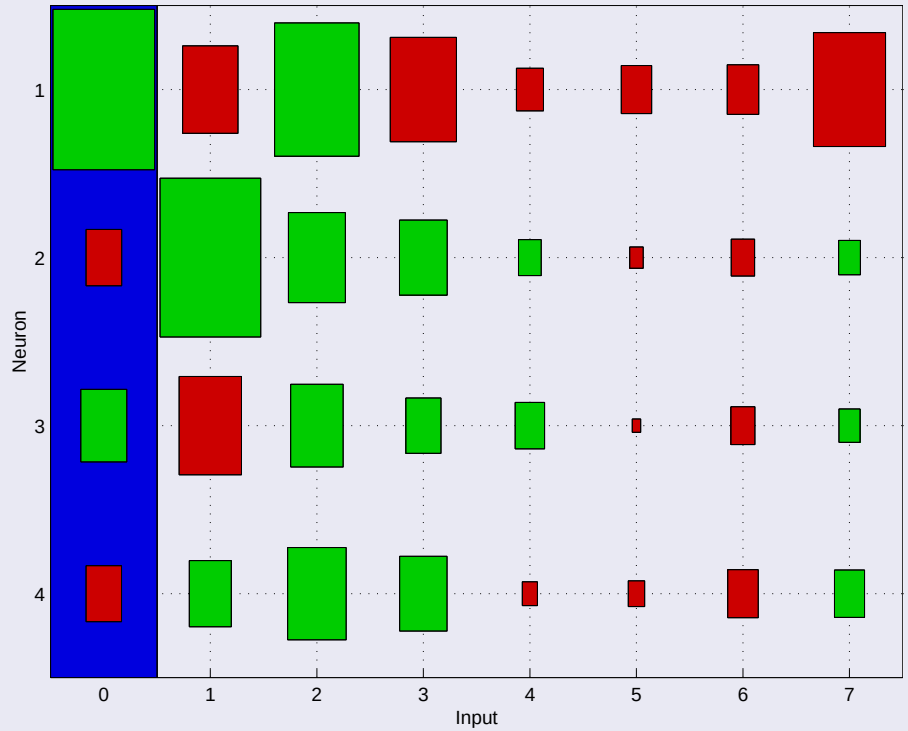
Training der Netze: Relevanz der Eingabe

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

Mittlerer Fehler beim Lernen

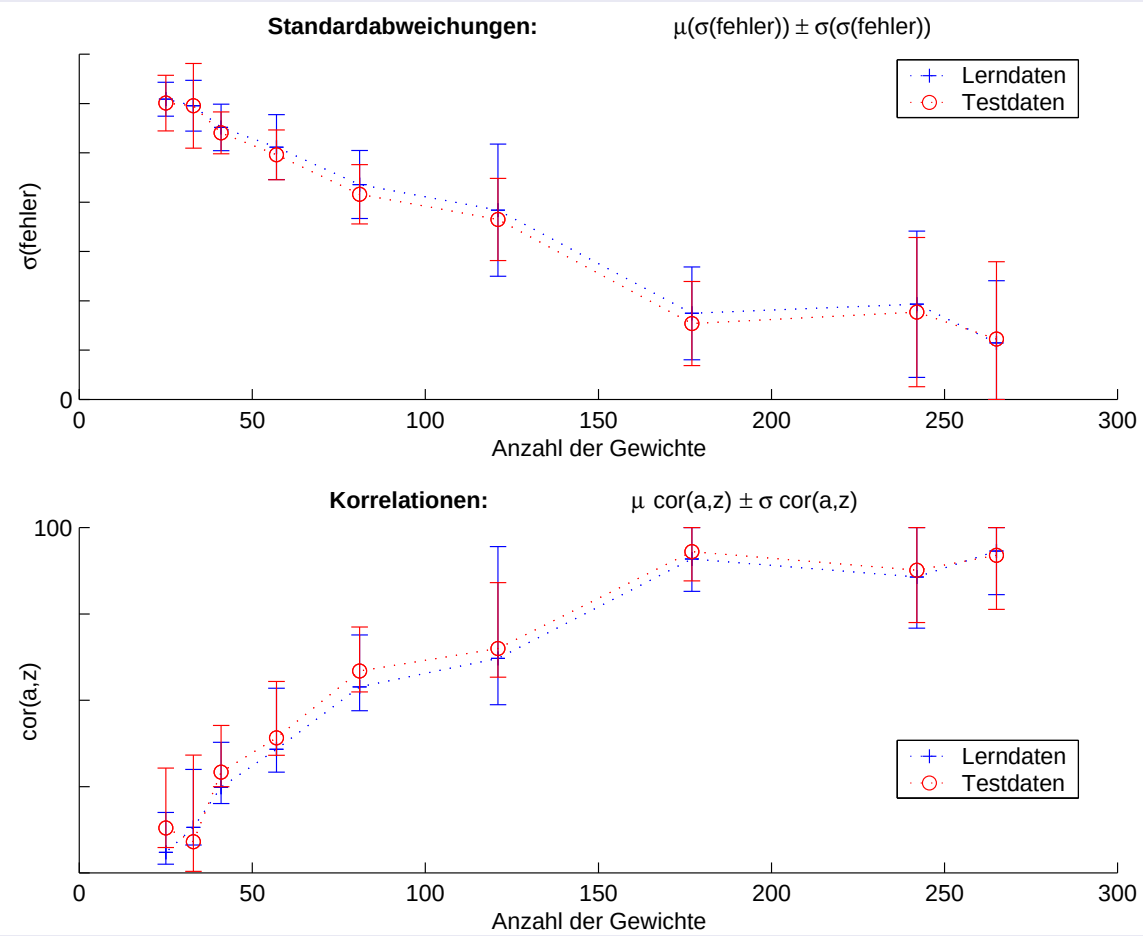


Gewichte in ersten Schicht



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,

⇒ Anregung wird über Koppelpunkte (Achse / Federbeindom) zu Komfortpunkten übertragen,

⇒ Vibrationen, die vom Insassen an den Komfortpunkten wahrgenommen werden.

Entstehung der störenden Vibration

Anregung

Die Fahrzeuganregung wird durch Radresonanz dominiert,

⇒ Anregung wird über Koppelpunkte (Achse / Federbeindom) zu Komfortpunkten übertragen,

⇒ 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

Training der neuronalen Netze

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

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

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

Mustererkennung

Insolvenzerkennung

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

Mustererkennung

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.

Mustererkennung

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.

Mustererkennung

Fahrzeug, Robotersteuerung

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

Potentielle Kündiger

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

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.

Mustererkennung

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.

Potentielle Kündiger

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

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.

Mustererkennung

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.

Potentielle Kündiger

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

Herkunft Olivenöl

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

Aktienkursvorhersage

Vorhersage auf der Grundlage des bisherigen Kursverlaufs und betriebswirtschaftlicher Daten.

Mustererkennung

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.

Olfaktometer

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

Herkunft Olivenöl

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

Aktienkursvorhersage

Vorhersage auf der Grundlage des bisherigen Kursverlaufs und betriebswirtschaftlicher Daten.

Mustererkennung

Reaktionsfähigkeit

Aus quantitativen Eigenschaften einer Bindung wird auf ihre Reaktionsfähigkeit geschlossen.

Olfaktometer

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

Herkunft Olivenöl

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

Proteinstruktur

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

Mustererkennung

Energiebedarf

Aus dem Stromverbrauch von Großabnehmern wird auf den Bedarf im Folgejahr geschlossen.

Olfaktometer

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

Herkunft Olivenöl

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

Proteinstruktur

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

Mustererkennung

Energiebedarf

Aus dem Stromverbrauch von Großabnehmern wird auf den Bedarf im Folgejahr geschlossen.

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.

Proteinstruktur

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

Mustererkennung

Energiebedarf

Aus dem Stromverbrauch von Großabnehmern wird auf den Bedarf im Folgejahr geschlossen.

Bremsmoment

Bestimmen des Bremsmoments aus hydraulischem Druck und Geschwindigkeit.

Neuronales Stethoskop

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

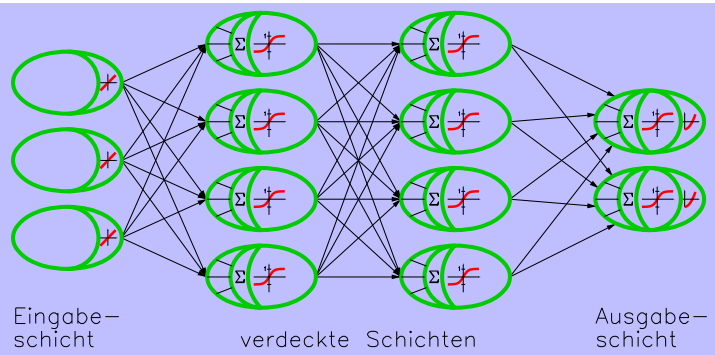
Proteinstruktur

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

Zusammenfassung

Neuronale Netze eignen sich sehr gut

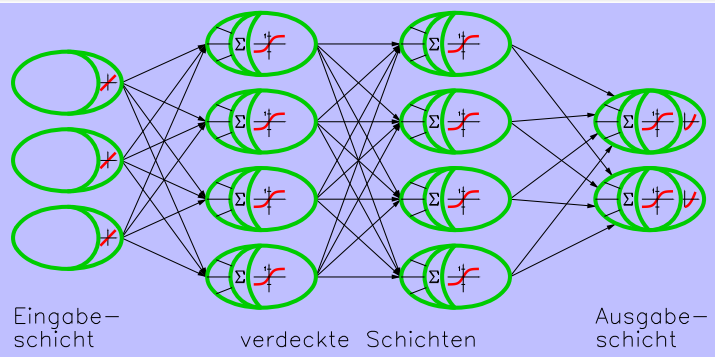
- zur Aufnahme von Systemfähigkeiten (KnowHow),
- zur Abbildung funktionaler Zusammenhänge



Zusammenfassung

Neuronale Netze eignen sich sehr gut

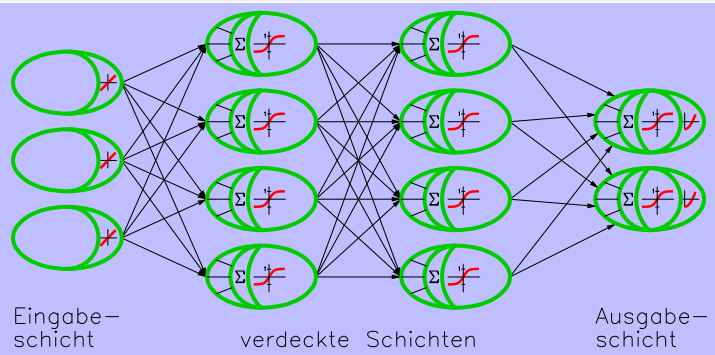
- zur Aufnahme von Systemfähigkeiten (KnowHow),
- zur Abbildung funktionaler Zusammenhänge



Zusammenfassung

Neuronale Netze eignen sich sehr gut

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

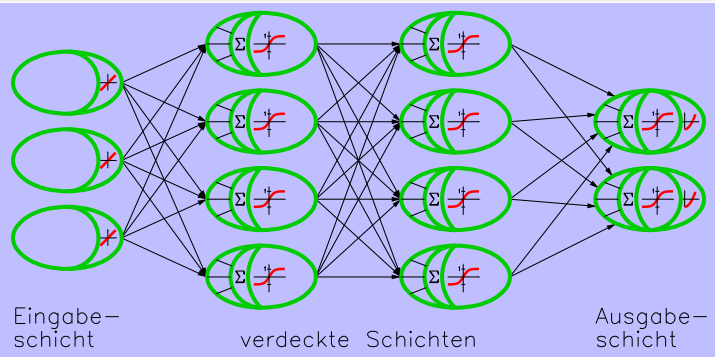


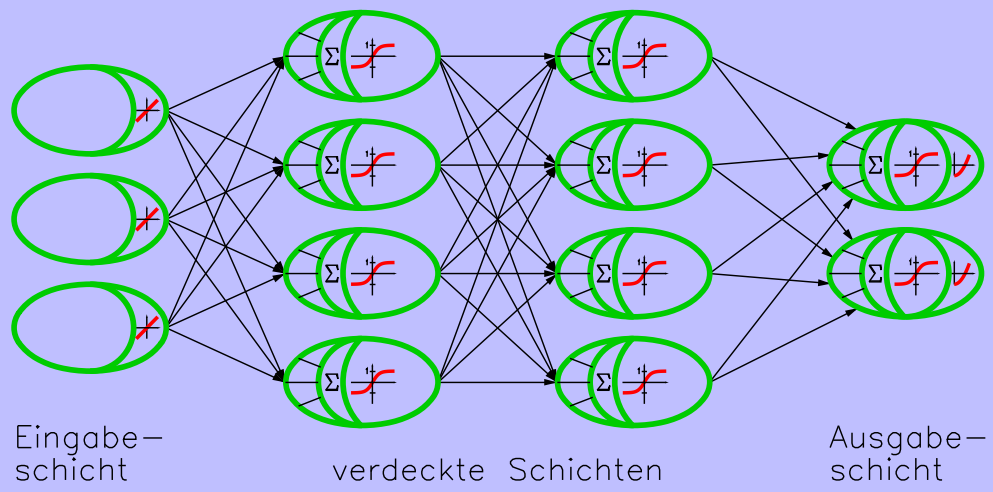
Zusammenfassung

Neuronale Netze eignen sich sehr gut

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

durch eine ausgleichende Interpolation zwischen den Stützstellen.





Vielen Dank für Ihre Aufmerksamkeit!