Neuronale Netze
Fuzzy Logik

Prof. Dr. Hoffmann

SS 2003

Tobias Webelsiep
Daniel Webelsiep

Stand: 29. Juni 2003

Inhaltsverzeichnis

Fuzzy Logik

1 Einführung

1.1 Fuzzy Logik - Was ist das?

Unscharfe Logik Die klassische Logik kann in der Praxis nicht angewendet werden.

1.2 „Scharfe“ Logik - Klassische Aussagenlogik

Es gibt 5 Junktoren zur Verknüpfung von Aussagenvariablen:

a	b	$\neg a$	$a \wedge b$	$a \vee b$	$a \Rightarrow b$	$a \Leftrightarrow b$
f	f	w	f	f	w	w
f	w	w	f	w	w	f
w	f	f	f	w	f	f
w	w	f	w	w	w	w

Prioritäten: 1. $\neg$, 2. $\wedge$ $\vee$, 3. $\Rightarrow$, 4. $\Leftrightarrow$

Keinen Regeln bezüglich Transitivität $\wedge$ oder $\vee$, d.h. $a \wedge b \vee c$ ist unzulässig, da unklar ist, ob $(a \wedge b) \vee c$ oder $a \wedge (b \vee c)$ gemeint ist.

Logischer Wert von Ausdrücken wird mit Hilfe von Wahrheitsstufen ermittelt.

z.B. $(a \wedge b) \vee c$

a	b	c	$(a \wedge b)$	$(a \wedge b) \vee c$
f	f	f	f	f
f	f	w	f	w
f	w	w	f	w
w	f	f	f	f
w	f	w	f	w
w	w	f	w	w
w	w	w	w	w

Tautologie: Eine Aussage heißt allgemeingültig, wenn sich für alle Kombinationen der Eingangsgrößen das Resultat „wahr“ ergibt.

2 „Scharfe“ Mengenlehre - Klassische Mengenlehre

Eine Menge M (nach Cantor, 1845-1918) ist die Zusammenfassung von bestimmten wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens zu einem Ganzen.

„Scharfe“ Mengen (crisp sets) können auf folgende Arten definiert werden:

- $A = \{1, 3, 7, 8\}$
- $M = \{x|P(x)\}$ Menge aller x mit der Eigenschaft $P(x)$
  z.B. $G = \{x|x \text{ ist gerade}\}$
- Die Reihenfolge der Elemente einer Menge ist ohne Bedeutung
- Dasselbe Element darf nicht mehrfach vorkommen
- Zwei Mengen sind gleich, wenn sie die gleichen Element enthalten
- Eine Menge darf sich nicht selbst enthalten

Teilmenge (Untermenge) Wenn für jedes $x \in A$ gilt $x \in B$, dann ist A Teilmenge von B ($A \subset B$). Wenn $A \neq B$, ist A echte Teilmenge.

Leere Menge - Nullmenge $\emptyset$ enthält kein Element (*Leere Menge*) und ist Untermenge jeder beliebigen Menge M.

Potenzmenge Enthält M n Elemente, dann setzt sich die Potenzmenge $P(M)$ aus allen 2^n Untermengen von M zusammen.

Beispiel: $M = \{x, y, z\}$
P(M) = { {}, {x}, {y}, {z}, {x, y}, {x, z}, {y, z}, {x, y, z} }

2.1 Mengenoperationen - Mengen-Algebra

Vereinigung: $A \cup B = \{x|x \in A \vee x \in B\}$

Durchschnitt: $A \cap B = \{x|x \in A \wedge x \in B\}$

Differenz: $A \setminus B = \{x|x \in A \wedge x \notin B\}$

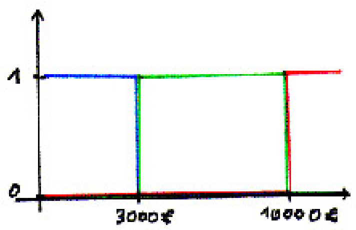
Symmetrische Differenz: $A \bigtriangleup B = (A \cup B)/(A \cap B)$
$= x|(x \in A \wedge x \notin B)(x \in B \wedge x \notin A)$

Produkt von Mengen: $A \times B$
$A = a_1, a_2, ..., a_n, B = b_1, b_2, ..., b_m$
$A \times B = \{(a_1, b_1), (a_1, b_2), ..., (a_1, b_m), (a_2, b_1), ..., (a_n, b_m)\}$
$A \times B = \{x|x = (a \in A, b \in B)\}$

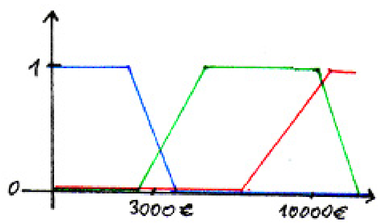
Komplement A: $\bar{A} = \{x|x \in G \wedge x \notin A\}$

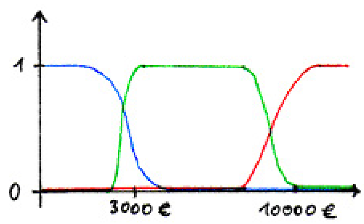
3 Fuzzy-Mengen - Unscharfe Mengen

Klassische Menge (Bsp. Verdienst)



Tatsächlich unscharfe Abgrenzung, fließende Übergänge





Ein Element einer unscharfen Menge kann nur teilweise zu dieser Menge gehören.

Bei „Scharfe Mengen" gehört ein Element entweder zu Menge A oder nicht. Dies kann durch eine sog. Zugehörigkeitsfunktion $\mu_A(x)$ folgendermaßen ausgedrückt werden:

$$\mu_A(x) = \begin{cases} \emptyset & \text{für } x \notin A \\ 1 & \text{für } x \in A \end{cases}$$

D.h. „*scharfe Mengen*" sind durch zweiwertige Zugehörigkeitsfunktionen gekennzeichnet.

Für „*unscharfe Mengen*" (Fuzzysets) gilt dagegen:

- $\mu_A(x)$ kann alle reellen Werte in [0, 1] annehmen
- $\mu_A(x)$ ist der Zugehörigkeitsgrad, der für jedes $x \in G$ angegeben wird
- Eine Fuzzy-Menge A über der Grundmenge G kann beschrieben werden als $A = \{(x, \mu_A(x)) | x \in G\}$
- Bei endlicher Zahl von Elementen x mit $\mu_A(x) > 0$ ist eine Darstellung durch geordnete Paare sinnvoll, z.B. $A = (x_1, \mu_A(x)_1), (x_2, \mu_A(x)_2), ..., (x_n, \mu_A(x)_n)$

Beispiel.: $A = \{gegen\ Jahresmitte\}$ über $G = \{Jan, Feb, März, ..., Dez\}$

Dann könnte A etwa folgendermaßen aussehen:
$A = \{(Apr, 0.1), (Mai, 0.5), (Juni, 1.0), (Juli, 1.0)\}$
(Sicht des Prof. bez. Abgabetermin)
$A = \{(Aug, 0.1), (Sept, 0.2), (Okt, 0.3), (Nov, 0.4), (Dez, 0.5)\}$
(Sicht des Studenten)

Fazit Festlegung oder Ermittlung von Zugehörigkeitsgraden ist nicht Gegenstand der Theorie der Fuzzysets. Zugehörigkeitsgrade sind unter Berücksichtigung des Anwendungsgebietes und der zu lösenden Fragestellung in der Regel von Experten festzulegen.

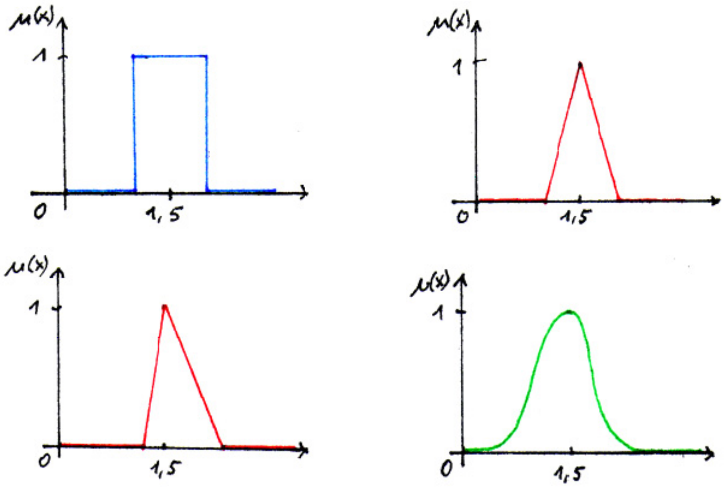
3.1 Wichtige Unterschiede zu scharfen Mengen

- Eine Fuzzy-Menge enthält stets alle Element der Grundmenge G (zur besser Übersicht werden Elemente mit = 0 nicht aufgeführt)
- Die leere Fuzzy-Menge hat daher genau so viele Elemente wie die Grundmenge G, wobei der Zugehörigkeitswert aller Elemente 0 ist (Die klassische Leere Menge enthält dagegen kein Element)

3.2 Fuzzy-Mengen über Grundmengen mit unendlich vielen Elementen

Beispiel.: $G = \{x | x \in \Re \wedge x \geq 0\}$

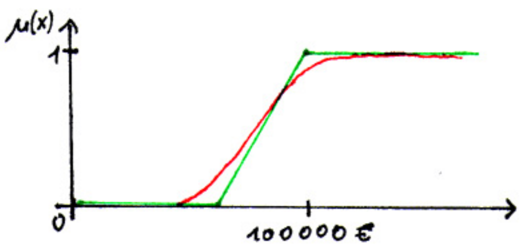
Es sein $A = \{x | x \approx 1,5\}$
A wird i. d. R. durch einen Graph der Zugehörigkeitsfunktion repräsentiert



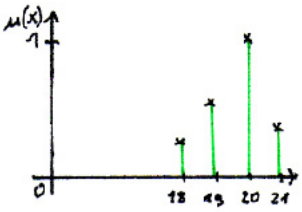
Singeltons

Beispiel: $G = \{x | x \in \aleph \wedge x \geq 0\}$

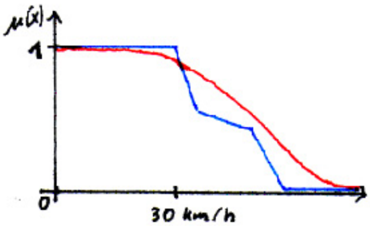
$G = \{x | x\ ist\ Jahreseinkommen\}$
$A(x) = \{x | x \in G \wedge x\ recht\ groß\}$



Obwohl Elemente von G eigentlich diskret sind, wird hier x wie eine stetige Größe behandelt.
(*Wegen des extrem geringen Abstands zwischen den einzelnen Singletons*)

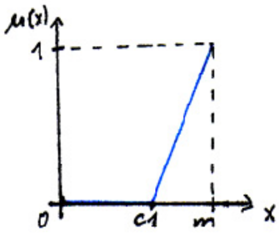


Beispiel: $G = \{x|x \text{ ist Geschwindigkeit}\}$
$A = \{x|x = \text{niedrige Geschwindigkeit}\}$



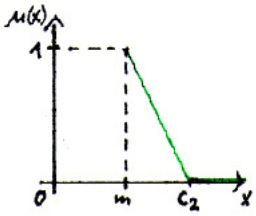
4 Zugehörigkeitsfunktionen: Mathematische Darstellung

4.1 Grundmuster



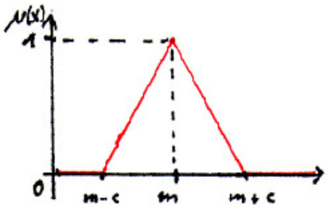
Linearer Anstieg:

$$\mu_A(x) = \begin{cases} 0 & \text{für } x < c_1 \\ \frac{x-c_1}{m-c_1} & \text{für } c_1 \leq x \leq m \end{cases}$$



Linearer Abfall:

$$\mu_A(x) = \begin{cases} \frac{x-c_2}{m-c_2} & \text{für } m \leq x \leq c_2 \\ 0 & \text{für } x > c_2 \end{cases}$$



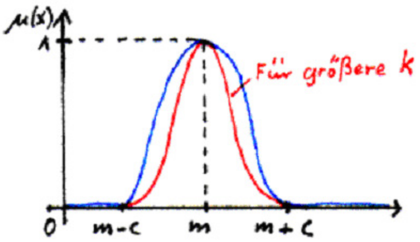
Symmetrisch, Triangulär:

$$\mu_A(x) = \begin{cases} 0 & \text{für } x < m-c \\ \frac{x-(m-c)}{c} & \text{für } m-c \leq x \leq m \\ \frac{m+c-x}{c} & \text{für } m \leq x \leq m+c \\ 0 & \text{für } x > m+c \end{cases}$$

oder

$$\mu_A(x) = \begin{cases} 0 & \text{für } |x-m| > c \\ 1 - \frac{x-m}{c} & \text{für } |x-m| \leq c \end{cases}$$

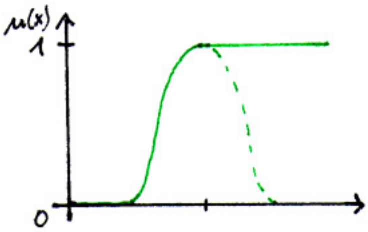
4.2 Grundmuster für glockenförmige Zugehörigkeitsfunktionen



Ansatz: $$\mu_A(x) = \begin{cases} 0 & \text{für } |x-m| \geq c \\ e^{-k\left[\frac{1}{(c-|x-m|)^2} - \frac{1}{c^2}\right]} & \text{für } |x-m| < c \end{cases}$$

Überprüfen der „kritischen Punkte“:

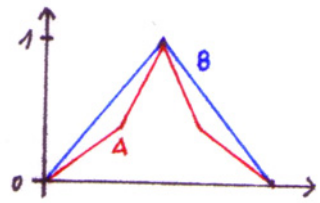
- $\mu_A(m) = e^{-k\left[\frac{1}{(c-|0|)^2} - \frac{1}{c^2}\right]} = e^{-k\left(\frac{1}{c^2} - \frac{1}{c^2}\right)} = e^0 = 1$
- $\mu_A(m+c) = e^{-k\left[\frac{1}{(c-c)^2} - \frac{1}{c^2}\right]} = e^{-\infty} = 0$
- $\mu_A(m-c) = e^{-k\left[\frac{1}{(c-c)^2} - \frac{1}{c^2}\right]} = e^{-\infty} = 0$



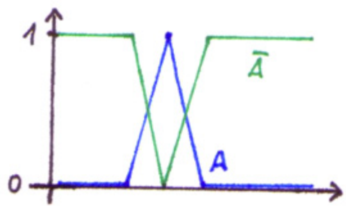
Fuzzy-Kurven können nun durch die einzelnen Grundmuster abschnittsweise definiert werden.

5 Weitere Definitionen

- **Betrag eine Fuzzy-Menge:** $|A| = \sum_{x \in G} \mu_A(x_i)$
- **Träger einer Fuzzy-Menge:** $S(A)$ oder *supp*(A) (support) ist die Menge aller Elemente aus A mit $\mu_A(x) > 0$, d.h. $S(A) = \{(x_i, \mu_A(x_i)) | \mu_A(x_i) > 0\}$
- **Toleranz:** $T(A)$ der $tol(A)$ ist die Menge der Elemente mit Zugehörigkeitsgrad 1 $T(A) = \{(x_i, \mu_A(x_i)) | \mu_A(x_i) = 1\}$
- **Alpha-Level-Menge** (α-Level-Menge): A ist die Menge der Elemente mit $Zugehrigkeitsgrad \geq \alpha$, d.h. $A_\alpha = \{(x_i, \mu_A(x_i)) | \mu_A(x_i) \geq \alpha\}$
- **Gleichheit von Fuzzy-Mengen:** $A = B \Leftrightarrow \mu_A(x) = \mu_B(x)\ \forall x \in G$ (identische Zugehörigkeitsgrade)
- **Untermengen:** A ist Untermenge von B, wenn $\forall x \in G, \mu_A(x) \leq \mu_B(x)$ ist, $A \subset B \Leftrightarrow \mu_A(x) \leq \mu_B(x)\ \forall x \in G$



- **Komplement von Fuzzy-Mengen:** Es sei $A = \{(x, \mu_A(x)) | x \in G\}$ Dann ist $\bar{A} = \{(x, \mu_{\bar{A}}(x)) | \mu_{\bar{A}}(x) = 1 - \mu_A(x) \in G\}$



Beispiel: $G = \{a, b, c, d, e\}$
$A = \{(a, 0.5), (c, 0.1), (d, 1.0)\}$
$\overline{A} = \{a, 0.5), (c, 0.9)\}$
$\overline{\overline{A}} = \{(a, 0.5), (c, 0.9)\} \neq A$

$\Rightarrow$ *Fehler*, da auch bei Fuzzy-Mengen $\overline{\overline{A}} = A$ gilt!

Richtig: $\overline{A} = \{(a, 0.5), (b, 1.0), (c, 0.9), (e, 1.0)\}$
$\overline{\overline{A}} = \{(a, 0.5), (c, 0.1), (d, 1.0)\}$

5.1 Modifizierer: Konzentration / Dilatation

Modifizierer sind Operatoren, die einen Wahrheitswert (Zugehörigkeit J/N) nicht grundsätzlich ändern, aber ihn in gewisser Weise beeinflussen.

Beispiel: Menge der guten Diplomanten

Durchschnittsnote	1.0	1.3	1.7	2.0	2.3	2.7	3.0	> 3.0
Zugehörigkeit	1.0	1.0	0.9	0.8	0.6	0.2	0.1	0

Begriffliche Verstärkung: Sehr gute Diplomanten (Menge A+)

Durchschnittsnote	1.0	1.3	1.7	2.0	2.3	2.7	3.0	> 3.0
Zugehörigkeit	1.0	1.0	0.8	0.5	0.2	0	0	0

Für alle Elemente gilt $\mu_{A+}(x) \leq \mu_A(x)$

$\Rightarrow$ $A+$ ist Untermenge von A ($A+ \subset A$), d.h. durch die begriffliche Verstärkung wird eine Teilmenge der ursprünglichen Menge gebildet.

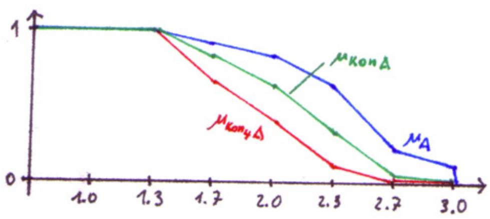
Begriffliche Verstärkung $\Rightarrow$ kleinere Zugehörigkeit
Diesen Effekt erreicht man auch durch die Bildung von $(\mu_A(x))^n$

$\Rightarrow$ Konzentrationsoperator $Kon_n A = \{(x, \mu_A(x)^n) | x \in G\}$

ohne Angabe von n wird üblicherweise n = 2 angenommen.
$Kon\ A = \{(x, \mu_A(x)^2) | x \in G\}$

Beispiel: Menge der guten Diplomanten

Durchschnittsnote	1.0	1.3	1.7	2.0	2.3	2.7	3.0	> 3.0
μ_A	1.0	1.0	0.9	0.8	0.6	0.2	0.1	0
$\mu_{Kon\ A}$	1.0	1.0	0.81	0.64	0.36	0.04	0.01	0
$\mu_{Kon_n A}$	1.0	1.0	0.656	0.41	0.1296	0.0016	0.0001	0



Die Umkehrung, d.h. begriffliche Abschwächung führt zu größeren Zugehörigkeiten. Zur Bildung einer Obermenge: $A \xrightarrow[Abschw\ddot{a}chung]{} A-$

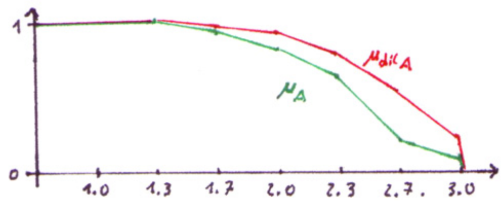
$\Rightarrow$ $A \subset A-$, d.h. $\mu_{A-}(x) \geq \mu_A(x)$

Die Abschwächung kann mit Hilfe des Dilatations-Operators nachgebildet werden, $dil_n A = \{x, \sqrt[n]{\mu_A(x)} | x \in G\}$

ohne Angabe von n wird 2 angenommen:
$dil\ A = \{x, \sqrt{\mu_A(x)} | x \in G\}$

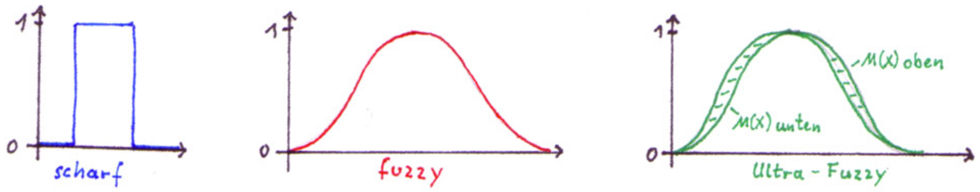
Beispiel: Menge der guten Diplomanten

Durchschnittsnote	1.0	1.3	1.7	2.0	2.3	2.7	3.0	> 3.0
μ_A	1.0	1.0	0.9	0.8	0.6	0.2	0.1	0
$\mu_{dil\ A}$	1.0	1.0	0.949	0.894	0.775	0.497	0.316	0



5.2 Ultra-Fuzzy

Fuzzy-Mengen, bei denen die Zugehörigkeitsgrade ebenfalls unscharfe Zahlen sind.



Wegen fehlender mathematischer Grundlagen in der Praxis nicht angewandt!

6 Rechenoperationen für Fuzzymengen

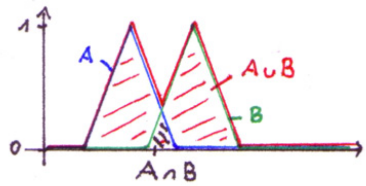
6.1 Vereinigung und Durchschnitt von Mengen

Es sei $A = \{(x, \mu_A(x) | x \in G\}$ und $B = \{(x, \mu_B(x) | x \in G\}$

Dann wird definiert:

$$A \cup B = \{(x, \mu_{A\cup B(x))} | x \in G\} \text{ mit } \mu_{A\cup B}(x) = max(\mu_A, \mu_B)$$
$$A \cap B = \{(x, \mu_{A\cap B(x))} | x \in G\} \text{ mit } \mu_{A\cap B}(x) = min(\mu_A, \mu_B)$$

Beispiel: $G = \{a, b, c, d, e, f\}$
$A = \{(a, 1.0), (b, 0.7), (d, 0.2), (f, 0.5)\}$
$B = \{(b, 0.5), (c, 1.0), (d, 0.9), (e, 0.6)\}$
$A \cup B = \{(a, 1.0), (b, 0.7), (c, 1.0), (d, 0.9), (e, 0.6), (f, 0.5)\}$
$A \cap B = \{(b, 0.5), (d, 0.2)\}$



6.2 Distributiv-Gesetze für Fuzzy-Mengen

$$A \cup (B \cup C) = (A \cup B) \cup C$$
$$A \cap (B \cap C) = (A \cap B) \cap C$$

Beweis für $A \cap (B \cap C) = (A \cap B) \cap C$ (für min- / max-Operatoren)
$\Leftrightarrow \ min(\mu_A, max(\mu_B, \mu_C)) = max(min(\mu_A, \mu_B), min(\mu_A, \mu_C))$

Beweis durch Betrachtung aller Fälle:

Fall	$max(\mu_B, \mu_C)$	$min(\mu_A, 2)$	$min(\mu_A, \mu_B)$	$min(\mu_A, \mu_C)$	$max(4, 5)$
$\mu_A \geq \mu_B \geq \mu_C$	μ_B	μ_B	μ_B	μ_C	μ_B
$\mu_A \geq \mu_C \geq \mu_B$	μ_C	μ_C	μ_B	μ_C	μ_C
$\mu_B \geq \mu_A \geq \mu_C$	μ_B	μ_A	μ_A	μ_C	μ_A
$\mu_B \geq \mu_C \geq \mu_A$	μ_B	μ_A	μ_A	μ_A	μ_A
$\mu_C \geq \mu_A \geq \mu_B$	μ_C	μ_A	μ_B	μ_A	μ_A
$\mu_C \geq \mu_B \geq \mu_A$	μ_C	μ_A	μ_A	μ_A	μ_A

Spalte 3 (3) und Spalte 6 (6) stimmen überein

6.3 Alternative Definitionen für $\cap$ und $\cup$ (t-Norm, s-Norm)

- min-, max-Operatoren sind eine Möglichkeit die Verknüpfungen $\cap$ und $\cup$ zu realisieren
- Allgemein benutzt man 2 Abbildungsfunktionen s und t: $[0, 1] \times [0, 1] \rightarrow [0, 1]$, aus denen sich die Zugehörigkeitsgrade von $\cap$ und $\cup$ ergeben nach $\mu_{A \cap B} = t(\mu_A, \mu_B)$ und $\mu_{A \cup B} = s(\mu_A, \mu_B)$
- t und s bezeichnet man als t-Norm und s-Norm, bzw. t-Conorm

6.3.1 Eigenschaften

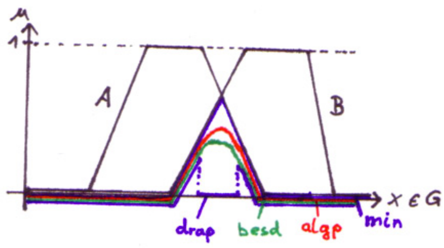
$t(x, y) = t(y, x)$; $s(x, y) = s(y, x)$	Kommutativität
$t(x, t(y, z)) = t(t(x, y), z)$; $s(x, s(y, z)) = s(s(x, y), z)$	Assoziativität
$t(x, 1) = 1$; $s(x, 1) = 1$ $t(x, 0) = 0$; $s(x, 0))x$	Spezielle Operationen mit 0 und 1

Für $x \leq w$ und $y \leq z$ folgt $t(x, y) \leq t(w, z)$ und $s(x, y) \leq s(w, z)$

Für alle t- und s-Normen gilt:
$min(x, y) \geq t(x, y)\ \forall t$
$max(x, y) \leq s(x, y)\ \forall s$

6.3.2 Alternative Definitionen (Auszug)

$A \cap B = \{(x, \mu_(A \cap B)|x \in G\} \hat{=}$ logisches UND

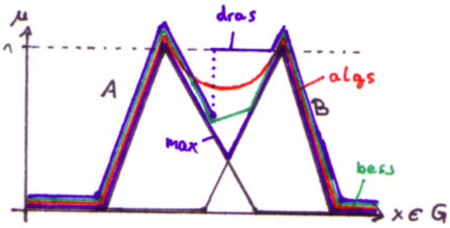


- min-Operator $\mu_{A \cap B} = min \mu_A, mu_B$
- algebraisches Produkt $\mu_{A \cap B} = algp(\mu_A, \mu_B) = \mu_A \cdot \mu_B$
- beschränkte Differenz $\mu_{A \cap B} = besd(\mu_A, \mu_B) = max(0, \mu_A + \mu_B - 1)$
- drastisches Produkt $\mu_{A \cap B} = drap(\mu_A, \mu_B) = \begin{cases} \mu_A & \text{für } \mu_B = 1 \\ \mu_B & \text{für } \mu_A = 1 \\ 0 & \text{sonst} \end{cases}$

Beispiel: Gesucht ist ein kostengünstiges (K) und farbenfrohes (F) Bild für die Geschäftsräume: Sei $G = \{B_1, B_2, B_3, B_4, B_5\}$

	μ_K	μ_F	min	algp	besd	drap
B1	0,8	0,8	0,8	0,64	0,6	0
B2	0,5	0,9	0,5	0,45	0,4	0
B3	0,77	0,94	0,77	0,7238	0,71	0
B4	0,73	0,99	0,73	0,7227	0,72	0
B5	1,0	0,2	0,2	0,2	0,2	0,2

$A \cup B = \{(x, \mu_(A \cup B) | x \in G\} \hat{=}$ logisches ODER



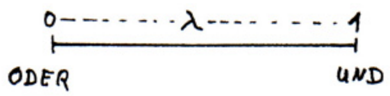
- max-Operator $\mu_{A \cup B} = max \mu_A, mu_B$
- algebraische Summe $= 1 - (1 - \mu_A) \cdot (1 - \mu_B)$ $\mu_{A \cup B} = algs(\mu_A, \mu_B) = \mu_A + \mu_B - \mu_A \cdot \mu_B$
- beschränkte Summe $\mu_{A \cup B} = bess(\mu_A, \mu_B) = min(1, \mu_A + \mu_B)$
- drastische Summe $\mu_{A \cap B} = dras(\mu_A, \mu_B) = \begin{cases} \mu_A & \text{für } \mu_B = 0 \\ \mu_B & \text{für } \mu_A = 0 \\ 1 & \text{sonst} \end{cases}$

6.4 Kompensatorische Operatoren

Weder UND- noch ODER-Verknüpfungen entsprechen häufig den Anforderungen eines speziellen Problems. Nachbildung der subjektiven Einstellung mittels *kompensatorischer Operatoren*: λ-Operator und γ-Operator

6.4.1 λ-Operator

$\mu_{A\lambda B} = \lambda \cdot \mu_{A \cap B} + (1 - \lambda)\mu_{A \cup B}$ für $0 \leq \lambda \leq 1$



$\lambda = 0 \Rightarrow$ ODER-Operator
$\lambda = 1 \Rightarrow$ UND-Operator

λ_k: $\mu_{A\lambda B} = \lambda \text{ min}(\mu_A, \mu_B) + (1 - \lambda) \text{ max}(\mu_A, \mu_B)$ *Komparator*
λ_a: $\mu_{A\lambda B} = \lambda \text{ algp}(\mu_A, \mu_B) + (1 - \lambda) \text{ algs}(\mu_A, \mu_B)$ *Algebraisch*
λ_b: $\mu_{A\lambda B} = \lambda \text{ besd}(\mu_A, \mu_B) + (1 - \lambda) \text{ bess}(\mu_A, \mu_B)$ *Beschränkt*
λ_d: $\mu_{A\lambda B} = \lambda \text{ drap}(\mu_A, \mu_B) + (1 - \lambda) \text{ dras}(\mu_A, \mu_B)$ *Drastisch*

Beispiel: Gesucht wird ein dunkles schnelles Auto

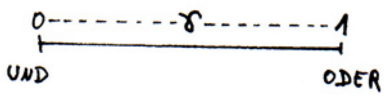
Nr	Farbe	W_{max}	N_{dunkel}	$N_{schnell}$
1	dunkelblau	170	0,9	0,4
2	rot	280	0,3	1,0
3	weiß	120	0,0	0,2

UND	*min*	*algp*	*besd*	*drap*
	0,4	0,36	0,3	0,0
	0,3	0,3	0,3	0,3
	0,0	0,0	0,0	0,0

$\lambda =$	1	0,75	0,5	0,25	0
	0,40	0,525	0,65	0,775	0,9
λ_k	0,30	0,475	0,65	0,825	1,0
	0	0,005	0,10	0,150	0,2
	0,36	0,505	0,65	0,795	0,94
λ_a	0,30	0,475	0,65	0,825	1,0
	0	0,005	0,10	0,150	0,2
	0,30	0,475	0,65	0,825	1,0
λ_b	0,30	0,475	0,65	0,825	1,0
	0	0,005	0,10	0,150	0,2
	0	0,250	0,50	0,750	1,0
λ_d	0,30	0,475	0,65	0,825	1,0
	0	0,005	0,10	0,150	0,2

6.4.2 γ-Operator

$\mu_{A\gamma B} = \mu_{A\cap B}^{1-\gamma} \cdot \mu_{A\cup B}^{\lambda}$ oder $\mu_{A\wedge B}^{1-\gamma} \cdot \mu_{A\vee B}^{\lambda}$ für $0 \leq \lambda \leq 1$



γ_k:	$\mu_{A\gamma B} =$	$\min(\mu_A, \mu_B)^{1-\gamma} \cdot \max(\mu_A, \mu_B)^{\gamma}$	*Komparator*
γ_a:	$\mu_{A\gamma B} =$	$\text{algp}(\mu_A, \mu_B)^{1-\gamma} \cdot \text{algs}(\mu_A, \mu_B)^{\gamma}$	*Algebraisch*
γ_b:	$\mu_{A\gamma B} =$	$\text{besd}(\mu_A, \mu_B)^{1-\gamma} \cdot \text{bess}(\mu_A, \mu_B)^{\gamma}$	*Beschränkt*
γ_d:	$\mu_{A\gamma B} =$	$\text{drap}(\mu_A, \mu_B)^{1-\gamma} \cdot \text{dras}(\mu_A, \mu_B)^{\gamma}$	*Drastisch*

Beispiel: Anwendung des γ-Operators:

$\gamma =$	1	0,75	0,5	0,25	0
	0,4	0,49	0,60	0,73	0,9
γ_k	0,3	0,41	0,55	0,74	1,0
	0	0	0	0	0,2
	0,36	0,458	0,582	0,739	0,9
γ_a	0,30	0,405	0,548	0,740	1,0
	0	0	0	0	0,2
	0,3	0,41	0,55	0,74	1,0
γ_b	0,3	0,41	0,55	0,74	1,0
	0	0	0	0	0,2
	0	0	0	0	1,0
γ_d	0,30	0,41	0,5	0,74	1,0
	0	0	0	0	0,2

Fazit:

- Qualitativ gleiche Ergebnisse bei der Anwendung von λ und γ
- γ-Operator wir häufiger verwendet

7 Fuzzy-Relationen

Gegeben: 2 klassische (scharfe) Mengen X und Y,
$X = \{x_1, x_2, ..., x_n\}$ $Y = \{y_1, y_2, ..., y_m\}$

Kartesisches Produkt

$$X \times Y = \{(x_1, y_1), (x_1, y_2), ..., (x_1, y_m), (x_2, y_1), ..., (x_n, y_1), ..., (x_n, y_m)\}$$

Klassische Relation

Aus *Kartesichem Produkt* werden nur die Paare aufgenommen, die in einer Relation zueinander stehen:

$$R_{xy} = \{(x, y)|x \in X, y \in Y, R_{xy}\}$$

Beispiel: $X = \{2, 3, 5\},\ Y = \{6, 9, 10, 11, 13\}$
$R_{xy} = X$ ist Teiler von Y

1. Schritt: Kartesisches Produkt
$X \times Y = \{(2, 6), (2, 9), ..., (2, 13), (3, 6), ..., (3, 13), (5, 6), ..., (5, 13)\}$
2. Schritt: Herausfiltern der Paare die R_{xy} erfüllen
$R = \{(2, 6), (2, 10), 3, 6), (3, 9), (5, 01)\}$

Fuzzy-Relationen

Es handelt sich um unscharfe Relationsvorschriften der Art:

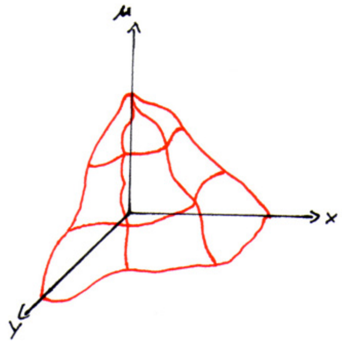
- x liegt in der Nähe von y
- x ist befreundet mit y
- kann gut mit y zusammenarbeiten

Fuzzy-Relation enthält alle Kombinationen des kartesischen Produkts. Die Relationsvorschrift wird von den Elementpaaren mit einem Grad $\mu(x, y) \in [0, 1]$ eingehalten (erfüllt). Diesen Wert bezeichnet man als *Possibilität*.

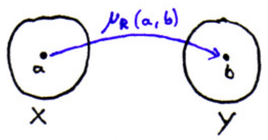
Notation: $R = \{((x, y), \mu_R(x, y))|x \in X, y \in Y\}$

7.1 Verdeutlichung von Relationen

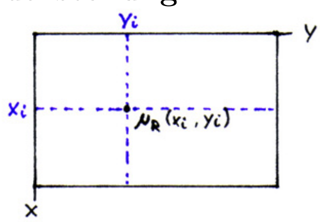
Posibilitätsfunktion über unendliche Fuzzy-Relationen



Pfeildiagramm



Matrixdarstellung



Beispiel: Einkommen in EUR in Gruppe X: $\{1350, 2800, 1620, 2060\}$
Einkommen in EUR in Gruppe Y: $\{2800, 2000, 6000, 3300, 4000\}$

Relation: x verdient nur halb so viel wie y: *Relationenmatrix*

	2800	2000	6000	3300	4000
1350	1,0	0,5	0,0	0,6	0,1
2800	0,0	0,0	0,8	0,0	0,1
1620	0,7	0,2	0,0	1,0	0,5
2060	0,3	0,0	0,0	0,5	0,9

7.2 Komposition von Fuzzy-Relationen (Verkettung)

Beispiel: (für scharfe Relationen)
$R_{1\,xy} : x < y, R_{2\,yz} : y < z, R_{3\,xz} : x < z$

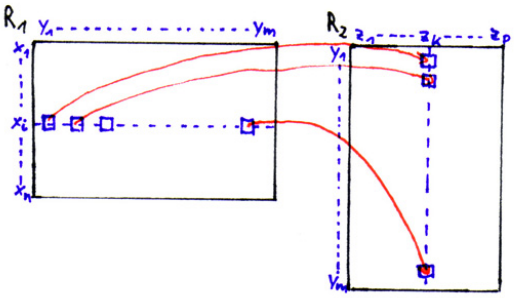
Dann gilt:
$(x < y) \wedge (y < z) \Rightarrow (x < z)$
$(x, y) \in R_1 \wedge (y, z) \in R_2 \Rightarrow (x, z) \in R_3$

R_3 ist Komposition von R_1 und R_2: $R_3 = R_2 * R_1$

- Reihenfolge ist in der Regel wichtig, d.h. $R_2 * R_1 \neq R_1 * R_2$
- Nicht alle Relationen lassen Kompositionen zu

Beispiel: $R_1 : x \neq y; R_2 : y \neq z$
$R_3 = R_2 * R_1$ macht keinen Sinn denn aus $x \neq y$ folgt nicht
$x \neq z$ (Bsp: $x = 3, y = 4, z = 3$)

Betrachtung der Verhältnisse beim Kompositionsschluss mit Fuzzy-Releationen:



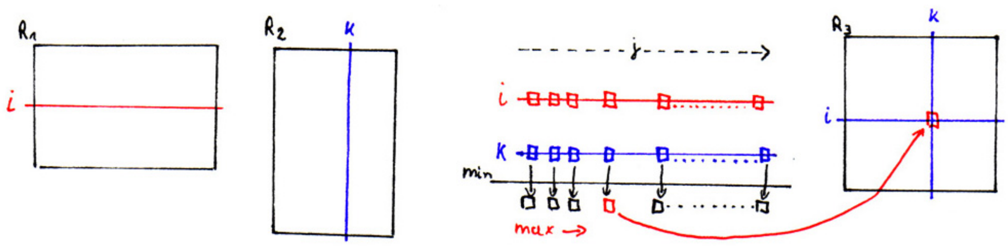
Beispiel: x verdient halb so viel wie y
y verdient doppelt so viel wie z
$\Rightarrow R_3 = R_2 * R_1$: x verdient so viel wie z!

$$\begin{aligned} R_3(x_i, z_k) = & [R_1(x_i, y_j) \wedge R_2(y_1, z_k)] \\ & \vee [R_1(x_i, y_2) \wedge R_2(y_2, z_k)] \\ & \vdots \\ & \vee [R_1(x_i, y_m) \wedge R_2(y_m, z_k))] \end{aligned}$$

- Für $\vee$ (ODER) benutzen wir max-Operator
- Für $\wedge$ (UND) benutzen wir min-Operator (oder algp)

$\Rightarrow$ max-min-Kompensation
$\mu_{R_2 * R_1}(x_i, z_k) = max(min(\mu_{R_1}(x_i, y_j), \mu_{R_2}(y_j, z_k)))$ für $1 \leq j \leq m$

Praktisches Vorgehen



Beispiel: R_1: verdient halb so viel wie y

	2800	2000	6000	3300	4000
1350	1,0	0,5	0,0	0,6	0,1
2800	0,0	0,0	0,8	0,0	0,1
1620	0,7	0,2	0,0	1,0	0,5
2060	0,3	0,0	0,0	0,5	0,9

R_2: y verdient doppelt so viel wie z

	1700	1000	2300
2800	0,6	0,1	0,0
2000	0,0	1,0	0,0
6000	0,0	0,0	0,8
3300	0,9	0,0	0,0
4000	0,6	0,0	0,1

R_3: x verdient genauso viel wie z

	1700	1000	2300
1350	0,6	0,5	0,1
2800	0,1	0,0	0,8
1620	0,9	0,2	0,1
2060	0,6	0,1	0,1

Nebenrechnung

$\mu_(x_1, z_1)$	1,0	0,5	0,0	0,6	0,1
	0,6	0,0	0,0	0,9	0,6
	0,6	0,0	0,0	0,6	0,1

$\mu_(x_1, z_2)$	1,0	0,5	0,0	0,6	0,1
	0,1	1,0	0,0	0,0	0,0
	0,1	0,5	0,0	0,0	0,0

$\mu_(x_1, z_3)$	1,0	0,5	0,0	0,6	0,1
	0,0	0,0	0,8	0,0	0,1
	0,0	0,0	0,0	0,0	0,1

Ergebnisse bei Verwendung von *algp*

	1700	1000	2300
1350	0,60	0,50	0,01
2800	0,06	0	0,64
1620	0,90	0,20	0,05
2060	0,54	0,03	0,09

8 Unscharfes Schließen / Fuzzy Reasoning

Ausgangspunkt: Zusammengesetzte logische Ausdrücke

Der Wahrheitswert $\mu(Aussage)$ $(0 \leq \mu \leq 1)$ muss sich aus der den Wahrheitswerten der atomaren Aussagen ergeben.

Bisher bekannt: $\vee$ ODER, $\wedge$ UND, $\neg$ NEGATION

$\mu(\neg a) = 1 - \mu(a)$

$\mu(a \wedge b) = min(\mu(a), \mu(b))$ *Es sind hier auch andere*

$\mu(a \vee b) = max(\mu(a), \mu(b))$ *t- und s-Normen möglich*

Problem: $a \Rightarrow b$?

Ansatz: $a \Rightarrow b \quad \Leftrightarrow \quad (\neg a \vee b) \Rightarrow \mu(a \Rightarrow b) = \mu(\neg a \vee b) = max(1 - \mu(a), \mu(b))$

$(a \vee b) \quad \Leftrightarrow (\neg a \Rightarrow b)$, d.h. $\mu(a \vee b) = \mu(\neg a \Rightarrow b) = max(1 - (1 - \mu(a), \mu(b))$
$= max(\mu(a), \mu(b))$

$$\begin{aligned}(a \vee b) \quad &\Leftrightarrow (a \vee b) \wedge (a \vee \neg a)\\ &\Leftrightarrow a \vee (b \wedge \neg a)\\ &\Leftrightarrow a \vee \neg(b \Rightarrow a)\\ &\Leftrightarrow \neg a \Rightarrow \neg(b \Rightarrow a)\end{aligned}$$

dann gilt (mit Wahrheitswerten arbeiten):

$$\begin{aligned}\mu(a \vee b) &= max(\mu(a), \mu(\neg(b \Rightarrow a))\\ &= max(\mu(a), 1 - \mu(b \Rightarrow a))\\ &= max(\mu(a), 1 - max(1 - \mu(b), \mu(a)))\end{aligned}$$

$1 - \mu(b) \geq \mu(a) : \mu(a \vee b) = max(\mu(a), \mu(b))$
$1 - \mu(b) < \mu(a) : \mu(a \vee b) = max(\mu(a), 1 - \mu(a))$ † Wiederspruch

D.h.: Die Definition $\mu(a \Rightarrow b) = max(1 - \mu(a), \mu(b))$ ist untauglich.

Konsequenz: Formel für $\mu(a \Rightarrow b)$ ist logisch nicht zu begründen.

Häufig verwendet wird: $\mu(a \Rightarrow b) = min(\mu(a), \mu(b))$ (Mamdani)

8.1 Beispiel für unscharfes Schließen

Wissensbasis:

1. Dass es sich um einen Lieferwagen mit Schiebedach und GPS handelt, ist äußerst unwahrscheinlich.
(Wahrheitswert $\alpha << 1$)

2. Dass der Wagen über 10 Jahre alt ist oder GPS hat, ist wahrscheinlich
(Wahrheitswert $\beta > 0.5$)

3. Sehr wahrscheinlich ist es ein Lieferwagen oder er ist höchstens 10 Jahre alt (Wahrheitswert $\gamma > \beta$)
4. Mit an Sicherheitgrenzender Wahrscheinlichkeit ist es ein Lieferwagen (Wahrheitswert $= 0,98$)

Atomare Ausdrücke:

a_0: Es ist ein Lieferwagen

a_1: Alter > 10 Jahre

a_2: Wagen hat Schiebedach

a_3: Wagen hat GPS

$\Rightarrow$ Wissensbasis:

1. $a_0 \wedge a_2 \wedge a_3,\ \alpha$
2. $a_1 \vee a_3,\ \beta$
3. $\neg a_0 \vee \neg a_1,\ \gamma$
4. $a_0,\ 0,98 \mid \alpha < \beta < \gamma$

8.2 Possibilistische Resolution

Annahme: $\alpha = 0,1$; $\beta = 0,8$; $\gamma = 0,9$

$$\begin{aligned}
\neg a_0 \vee \neg a_1 \quad & \Rightarrow \ \mu(\neg a_0 \vee \neg a_1) = max(1-\mu(a_0), 1-\mu(a_1)) = 0,9 \\
& \Rightarrow \ max(0,02, 1-\mu(a_1)) = 0,9 \Rightarrow 1-\mu(a_1) = 0,9 \\
& \Rightarrow \mu(a_1) = 0,1
\end{aligned}$$

$$\begin{aligned}
\mu(a_1 \vee a_3) \quad & = max(\mu(a_1), \mu(a_3)) = 0,8 \Rightarrow max(0,1; \mu(a_3)) = 0,8 \\
& \Rightarrow \mu(a_3) = 0,8
\end{aligned}$$

$$\begin{aligned}
\mu(a_0 \wedge a_2 \wedge a_3) & = min(\mu(a_0), \mu(a_2), \mu(a_3)) = min(0.98, \mu(a_2), 0.8) = 0,1 \\
& \Rightarrow \mu(a_2) = 0,1
\end{aligned}$$

$\Rightarrow$ *taugt alles nix, gieb im besten Fall untaugliche Ergebnisse, im schlimmsten Fall Kollisionen (Wiedersprüche)*

9 Fuzzy-Control / Fuzzy-Regelung

Abarbeiten von WENN-DANN-Regeln, wie z.B.

„Wenn Geschwindigkeit hoch und Abstand gering, dann bremse stark“
„Wenn Badewasser kalt, dann Warmwasserhahn voll aufdrehen“

Allgemein: WENN Prämisse DANN Konklusion

Ziel: Aus der Verarbeitung solcher WENN-DANN-Regeln ein Ergebnis abzuleiten, welches das Verhalten ein Systems bestimmt.

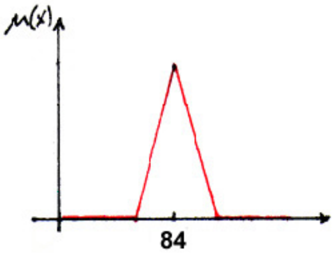
Beispiel: Notenfindung, d.h. Zuordnung von Punkten einer Klausur zu Noten

Regeln: WENN Pkt $\geq$ 96 DANN Note = 1 $\mu_{96} = 1$
WENN Pkt = 84 DANN Note = 2 $\mu_{84} = 1$
WENN Pkt = 72 DANN Note = 3 $\mu_{72} = 1$
WENN Pkt = 60 DANN Note = 4 $\mu_{60} = 1$
WENN Pkt $\leq$ 48 DANN Note = 5 $\mu_{48} = 1$

Konkrete Fragestellung: Welche Note gebe ich bei 93 Punkten

Problem:

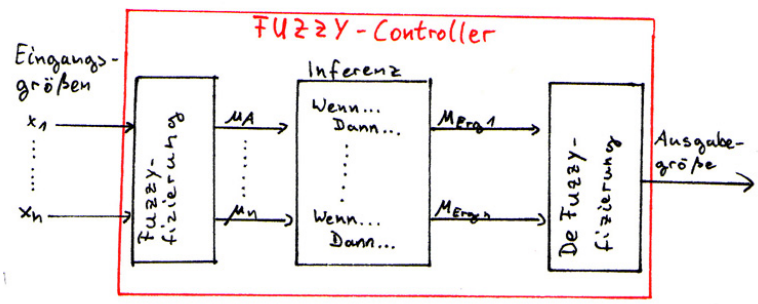
1. Festzulegen ist, wie aus konkretem Punktwert die Zugehörigkeiten $\mu_{96}, \mu_{84}, ..., \mu_{48}$ zu bestimmen sind: **Fuzzifizierung**



2. Wie wird aus $\mu_{96}, \mu_{84}, ..., \mu_{48}$ die Zugehörigkeit zu Noten bestimmt. $\Rightarrow \mu_1, \mu_2, ..., \mu_5$

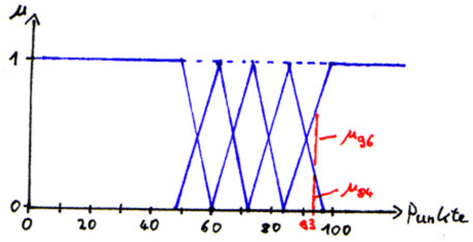
3. Das Ergebnis (nämlich die Zugehörigkeiten $\mu_1, ..., \mu_5$) muss in em eine konkrete Note umgesetzt werden: **Defuzzifizierung**

9.1 Ablauf



Details an konkretem Beispiel

1. Fuzzifizierung der Punktwerte:



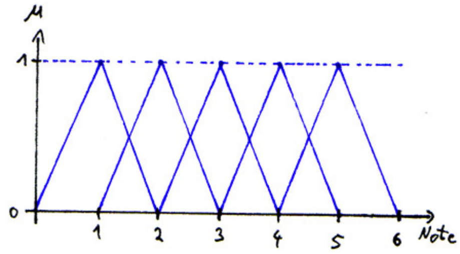
In der Regelungstechnik vorteilhaft (und meist benutzt): Überlappung der Zugehörigkeitsfunktionen in der hier gezeigten Art, d.h. Basis-Breite = 2 · Spitzenabstand.

2. Berechnung von Zugehörigkeitswerten für z.B. $p = 93$ Punkte
$\mu_{48} = \mu_{60} = \mu_{72} = 0$
$\mu_{84} = 0,25$; $\mu_{96} = 0,75$

3. Inferenz: Aberbeiten der Verarbeitungsregeln

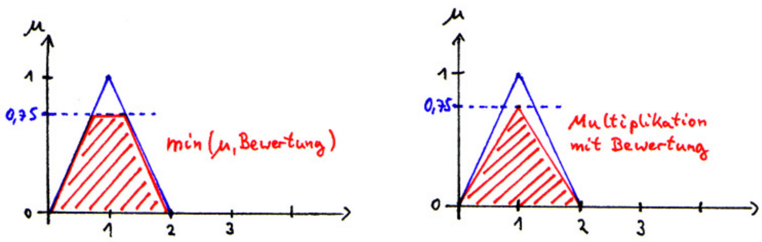
   - Wahrheitswerte der Prämissen werden ermittelt und diese auf den Verlauf der Zugehörigkeitsfunktionen der Konklusion angewandt.
   - Bewertete Zugehörigkeitfunktionen werden zu einer Ergebnisfläche zusammen gefasst.

   Konkret: Zugehörigkeitsfunktionen zu Noten erforderlich:

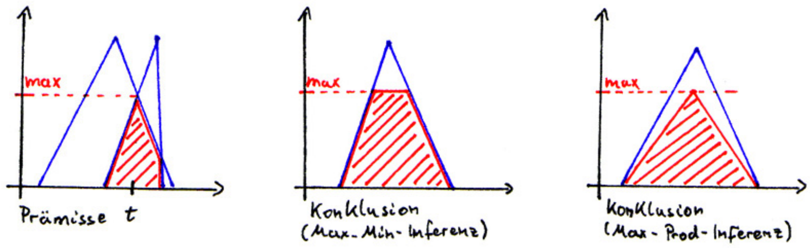


WENN PUNKTE ≥ 96 DANN NOTE = 1;
$\mu_{96} = 0,75 \rightarrow \mu_1$ mit $0,75$ bewerten

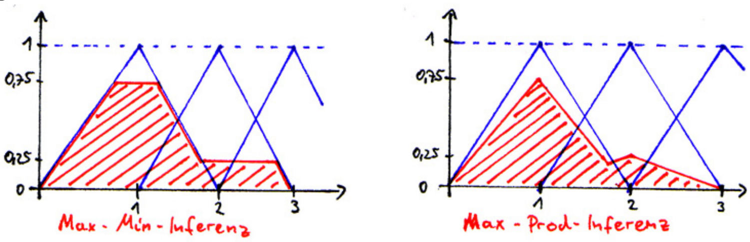
WENN PUNKTE ≥ 96 DANN NOTE = 1;
$\mu_{84} = 0,25 \rightarrow \mu_2$ mit $0,25$ bewerten



Allgemein:



Ergebnisfläche:



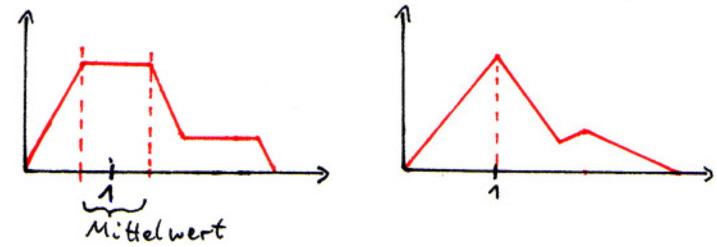
4. Defuzzifizierung

- Aus Gesamt-Ergebnisfläche muss jetzt *ein* einzelner Eregebniswert abgeleitet werden
- kann nach verschiedenen Methoden erfolgen

  (a) Mean of Maximum / Maximum-Mittelwert-Methode
  Es wird der Mittelwert aller Abszissenwert unter dem Maximum der Ergebnissfläche als Aussgabewert verwendet.

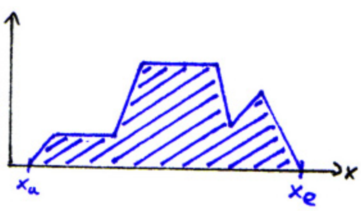
  Für unser Beispiel:

  max-min-Inferenz max-prod-Inferenz:



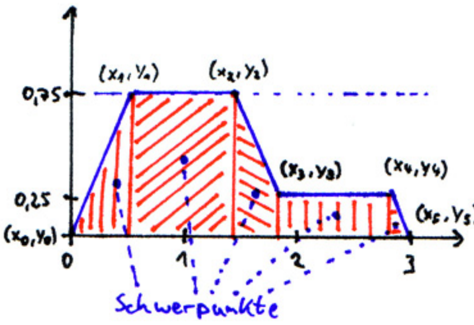
  Diese Methode ist relativ grob und in der Praxis wenig bewährt

  (b) Center of Gravity / Schwerpunkt-Methode
  Ermittelt wird die x-Koordinate des Schwerpunktes der Ergebnisfläche



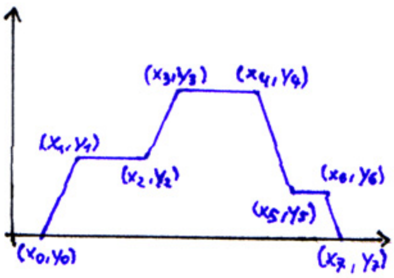
9.2 Schwerpunktberechnung bei linearen Zugehörigkeitsfunktionen

Für unser Beispiel

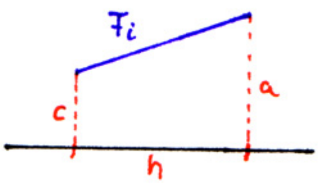


Schwerpunkte (x_{s_i}, y_{s_i}) der Teilflächen F_i: $x_s = \frac{\sum_{i=1}^{n} x_{s_i} \cdot F_i}{\sum_{i=1}^{n} F_i}$

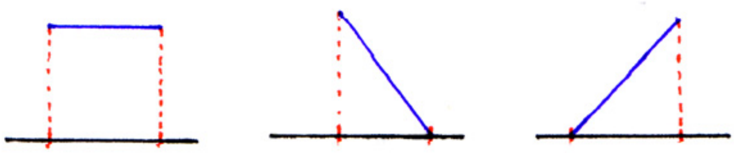
Berechnung von x_s aus vorgegebenem Streckenzug



Betrachte Ausschnitt:



Fläche: $F_i = h \cdot \frac{c+a}{2}$, d.h. $F_i = (x_i - x_{i-1}) \cdot \frac{y_i + y_{i-1}}{2}$

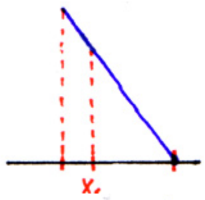


Schwerpunkt (x-Koordinate)

Trapezes: $x_{s_i} = x_{i-1} + \frac{x_i - x_{i-1}}{3} \cdot \frac{y_{i-1} + 2 \cdot y_i}{y_{i-1} + y_i}$

Rechteck: $x_{s_i} = x_{i-1} + \frac{x_i - x_{i-1}}{3} \cdot \frac{3 \cdot y_i}{2 + y_i} = x_{i-1} + \frac{x_i - x_{i-1}}{2}$

Dreieck:



$$x_{s_i} = x_{i-1} + \frac{x_i - x_{-1}}{3} \cdot \frac{y_i - 1}{y_i - 1}$$

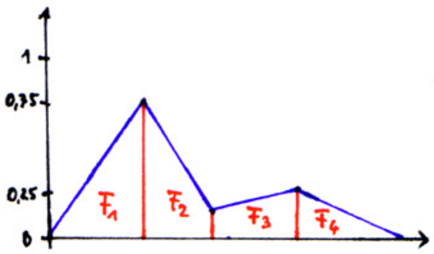
$$x_{s_i} = x_{i-1} + \frac{x_i - x_{-1}}{3} \cdot \frac{2 \cdot y_i}{y_i}$$

Konkret für obiges Beispiel (Notenfindung)

i	F_i		x_{s_i}	$x_{s_i} \cdot F_i$
1	$0,75 \cdot 0,75/2$	$= 0,28125$	$0,5$	$0,140625$
2	$0,5 \cdot 0,75$	$= 0,375$	1	$0,375$
3	$0,5 \cdot 0,5$	$= 0,25$	$1,458$	$0,364583$
4	$1 \cdot 0,25$	$= 0,25$	$2,25$	$0,5625$
5	$0,25 \cdot 0,25/2$	$= 0,03125$	$2,8333$	$0,088541$
	$\sum$	$= 1,18750$		$1,531249$

$\Rightarrow x_s = \frac{1,531249}{1,18750} = 1,3$ bei max-min-Inferenz

Max-Prod-Inferenz

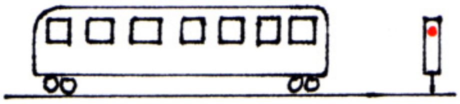


Berechnung:

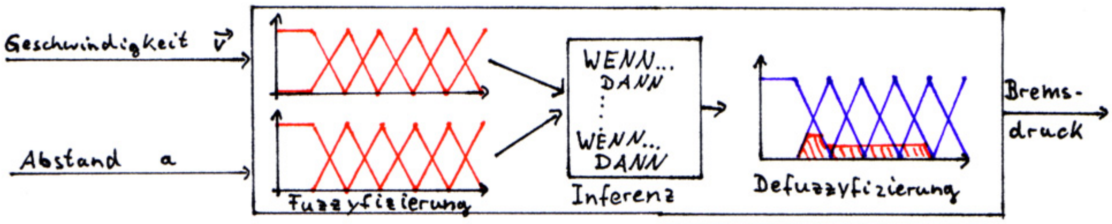
i	F_i		x_{s_i}	$x_{s_i} \cdot F_i$
1	$1 \cdot 0,75/2$	$= 0,28125$	$0,667$	$0,250013$
2	$0,75 \cdot (0,75 + 0,1875)/2$	$= 0,35156$	$1,3$	$0,457028$
3	$0,25 \cdot (0,1875 + 0,25)/2$	$= 0,05469$	$1,881$	$0,102866$
4	$1 \cdot 0,25/2$	$= 0,125$	$2,333$	$0,291666$
	$\sum$	$= 0,90625$		$1,101573$

$\Rightarrow x_s = \frac{1,101573}{0,90625} = 1,2$

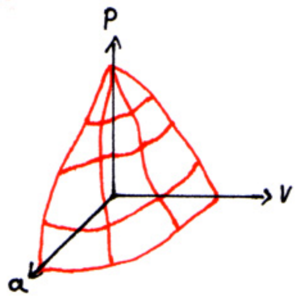
10 Beispiel: Regelung eines Bremsvorgangs



Prinzipieller Aufbau eines Reglers:



Klassische Methode: Regelfläche verwerten



Vorteil Fuzzy-Regelung: Regler kann im Prinzip von Laien erstellt werden.

10.1 Fuzzifizierung

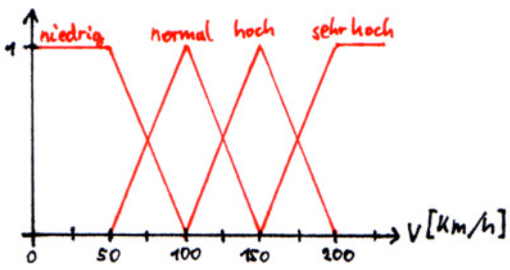
Regeln:

1. WENN v sehr hoch ODER a klein DANN bremse sehr stark
2. WENN v sehr hoch UND a groß DANN bremse schwach

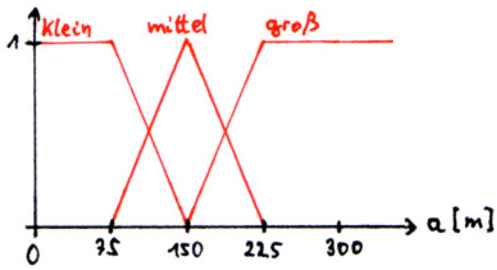
3. WENN v hoch UND a mittel DANN bremse stark
4. WENN v mittel UND a mittel DANN bremse mittel
5. WENN v niedrig UND a mittel DANN bremse schwach

6. WENN v sehr niedrig UND a klein DANN bremse schwach
7. WENN v sehr niedrig ODER a groß DANN bremse sehr schwach

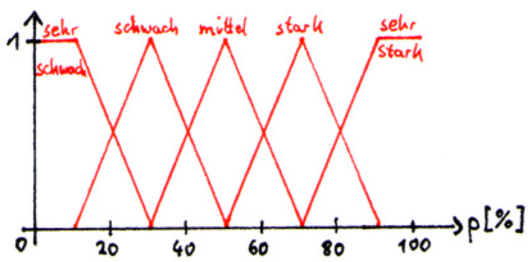
Geschwindigkeit:



Abstand:



Bremsdruck:



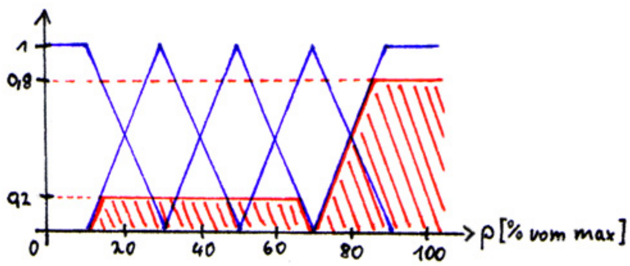
Frage: Wie soll gebremst werden?

Anwendung: Max-Min-Inferenz
Zahlenbeispiel: v = 90km/h; a = 90m;

Fuzzifizierung v: $\mu_{niedrig} = 0,75; \mu_{mittel} = 0,25$
Fuzzifizierung a: $\mu_{klein} = 0,8; \mu_{mittel} = 0,2$

Regeln abarbeiten:

1. $max(\mu_{sehrhoch}(v), \mu_{klein}(a) = 0,8 \Rightarrow \mu_{sehrstark}(p) = 0,8$
2. —
3. $min(\mu_{hoch}(v), \mu_{mittel}(a) = 0$
4. $min(\mu_{mittel}(v), \mu_{mittel}(a) = 02 \Rightarrow \mu_{mittel}(p) = 0,2$
5. $min(\mu_{niedrig}(v), \mu_{mittel}(a) = 02 \Rightarrow \mu_{schwach}(p) = 0,2$
6. —
7. —



$x_s = 69,6\%$

11 Kritische Würdigung

Vorteil:

- Erlaubt intuitiv leicht einsichtige Formulierungen einfacher Schlussfolgerungen in umgagssprachlicher (unschafer) Form

Nachteile:

- Erreicht schnell seine Grenzen bei mehrstufigen Schlussfolgerungsproblemen
- Die dahinter stehende methematische Theorie weist in vielen Punkten noch Schwachstellen auf bzw. ist letzlich nicht sauber begründbar
- Vielzahl „gleichwertiger“ Möglichkeiten, die nicht zielgerichtet eingesetzt werden können, sondern eher zum „Spielen“ verleiten
- Letztlich wird Ergebnis am Erfolg gemessen und nicht mathematisch begründet

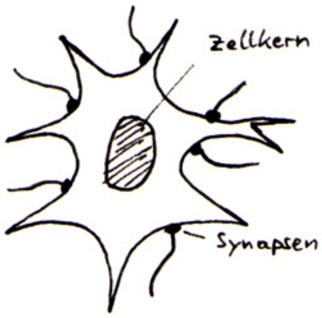
Neuronale Netze

12 Einführung

12.1 Neuronale Netze?

Versucht das Gehirn in Grundzügen nachzubauen. Wenn man versuchen würde das Gehirn nachzubilden, müsste man 10^{20} Neuronenrechnungen durchführen. Ein heutiger Rechner schafft 10^{10} Rechenoperationen. Dann bräuchte man 10^{10} Sekunden um einen Arbeitschritt im Gehirn nachzubilden.

12.2 Aufbau von Neuronen



Zellkern: Zusammenführung der Erregung

- übersteigt eine bestimmte Schwelle → Aktionsimpuls (Neuron feuert)
- übersteigt Schwelle nicht → Neuron feuert nicht, kein Aktionsimpuls

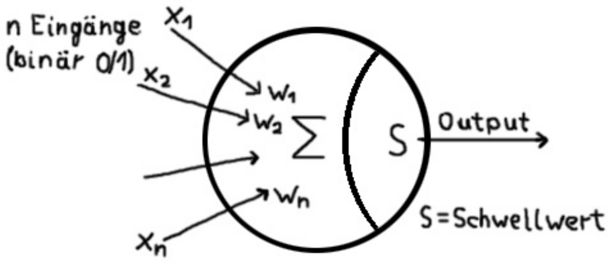
Synapsen:

- inhibitorische (wirkungs bremsend)
- exzitatorische (verstärkend)

Anzahl: ca. 10^{10} bis 10^{11} Nervenzellen

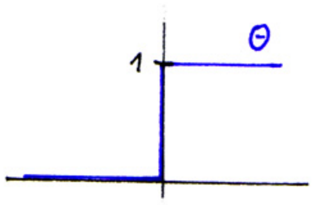
13 Künstliche Neuronen

Orientieren sich am Vobild des biologischen Neurons (Mc Culloch, Pitts (1943))



Eigenschaften:

- n-Eingänge (binär 0/1)
- $w_1, w_2, ..., w_n$ sind sybaptische Gewichte (Gewichtungsfaktoren)
- s = Schwellenwert
- $y = \Theta(\sum w_i \cdot x_i - s)$

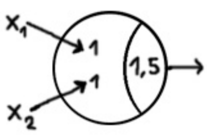


$$\Theta(x) = \begin{cases} 1 & \text{für } x \geq 0 \\ 0 & \text{sonst} \end{cases}$$

13.1 Einfache logische Funktionen mit McCulloch - Pitts - Neuronen

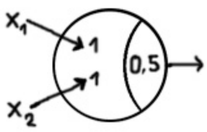
Zweiwertige Logik mit zwei Eingängen:

AND



x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

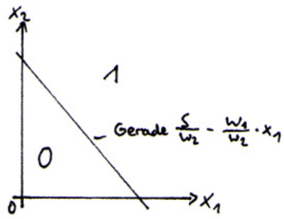
OR



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

1969 (Minsky und Papert): Nachweis, dass besimmte logische Funktionen von einem einzelnen Neuron nicht nachgebildet werden können, z.B.

XOR

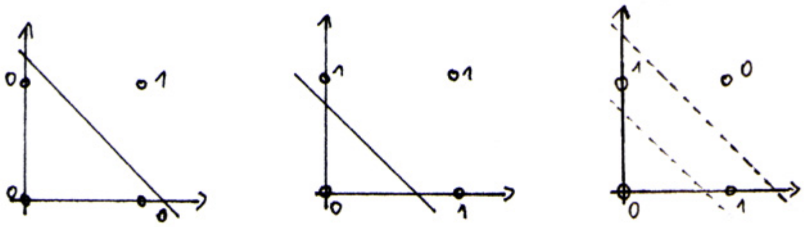


x_1	x_2	y
0	0	0
0	0	1
0	1	1
1	1	0

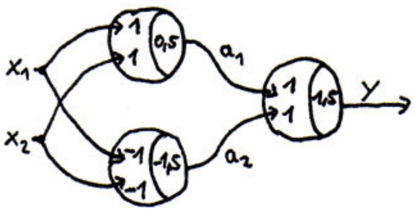
Begründung

Regel: $x_1 \cdot w_1 + x_2 \cdot w_2 \geq s$ (für Ouput 1)
$(w_2 > 0) \Rightarrow x_2 \geq \frac{s}{w_2} - \frac{w_1}{w_2} \cdot x_1$

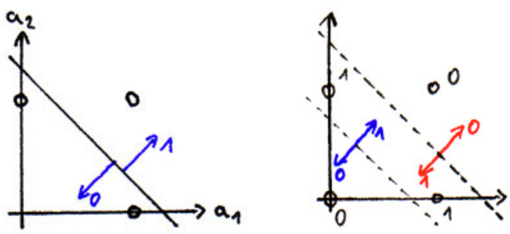
$x_1 \cdot w_1 + x_2 \cdot w_2 < s$ (für Ouput 0)
$\Rightarrow x_2 < \frac{s}{w_2} - \frac{w_1}{w_2} \cdot x_1$



XOR-Problem erfordert mehrschichtiges Netz mit 3 Neuronen:



Entscheidung kann jetzt anhand von zwei Geraden getroffen werden:



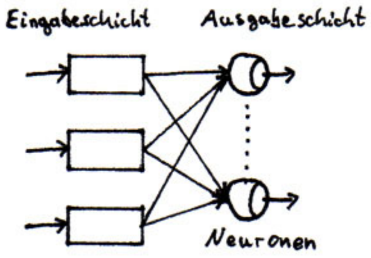
x_1	x_2	a_1	a_2	y
0	0	0	1	0
0	1	0	1	1
1	0	0	1	1
1	1	0	0	0

$\Rightarrow$

	w_1	w_2	s
1. Neuron	1	1	0,5
2. Neuron	-1	-1	-1,5
Ausgabe-Neuron	1	1	1,5

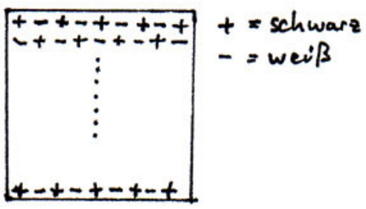
14 Perzeptron

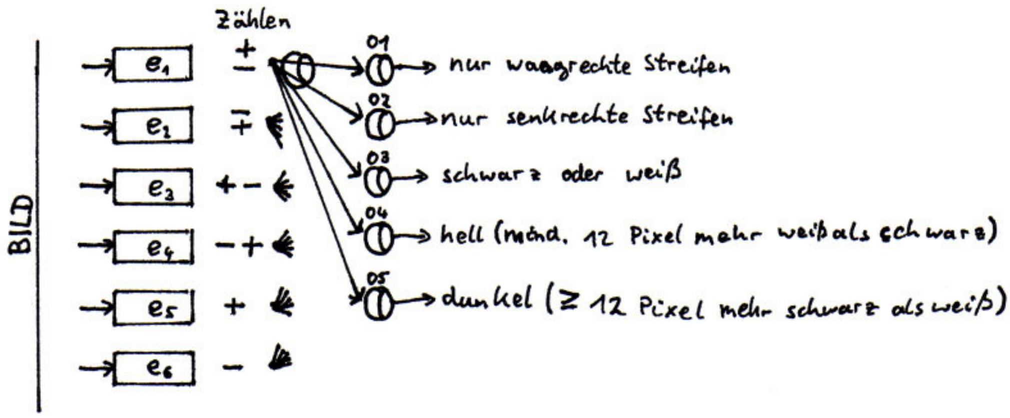
- 2-schichtiges Netz
- Eingabeschicht: Elemente, welche die Ausgabeschicht mit versorgen Informationen
- Ausgabeschicht: Neuronen, die von Eingabeschicht angesteuert werden und Output liefern



- Nur Vorwärtskopplung von Eingabe- zur Ausgabeschicht
- keine Rückkopplung und keine Querverweise zwischen Neuronen
- Entwickelt von: Rosenblatt, 1958

Beispiel: Beurteilung von Bildern mit weißen und schwarzen Pixeln





Hier: Input von Neuronen nicht mehr nur 0 und 1, sondern beliebig

Einstellung der Neuronen

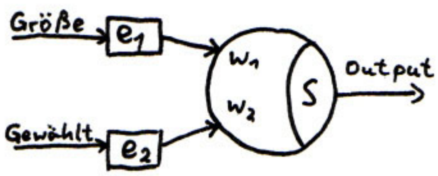
Neuronnummer	e_1	e_2	e_3	e_4	e_3	e_6	
	$\pm$	$\mp$	$+-$	$-+$	$+$	$-$	
	w_1	w_2	w_3	w_4	w_3	w_6	s
1	0	0	-1	-1	0	0	0
2	-1	-1	0	0	0	0	0
3	-1	-1	-1	-1	0	0	0
4	0	0	0	0	-1	1	12
5	0	0	0	0	1	-1	12

14.1 Anwendungsbeispiel von Perzeptron: Klassifikation nach 2 Merkmalen

Ausgangsdaten: Körpergröße und Gewicht von 5 Personen, die entweder American Football spielen oder Ballettänzer sind.

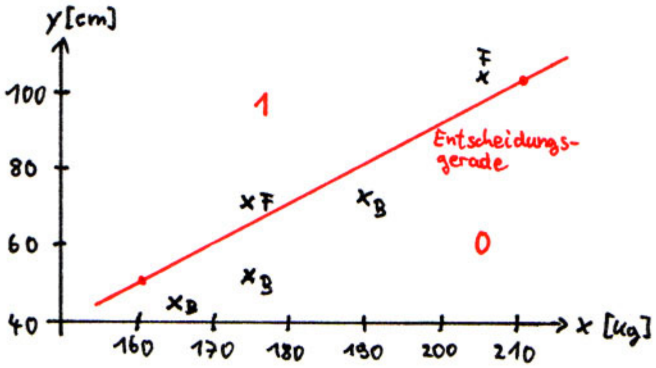
	F_1	F_2	B_1	B_2	B_3
Größe (cm)	205	175	165	175	190
gewicht (kg)	100	70	45	50	70

Gesucht ist Perzeptron folgenden Aufbaus:



$$\text{Output} = \begin{cases} 1 & \text{für } F \\ 0 & \text{für } B \end{cases}$$

Wie sind w_1, w_2 und s zu wählen?



Geradengleichung: $y - 50 = x - 160 \Rightarrow y = x - 110$

Output 1 bei: $y > x - 110$ oder $y - x + 110 \geq 0$

$\Rightarrow w_1 = -1, w_2 = 1, s = -110$

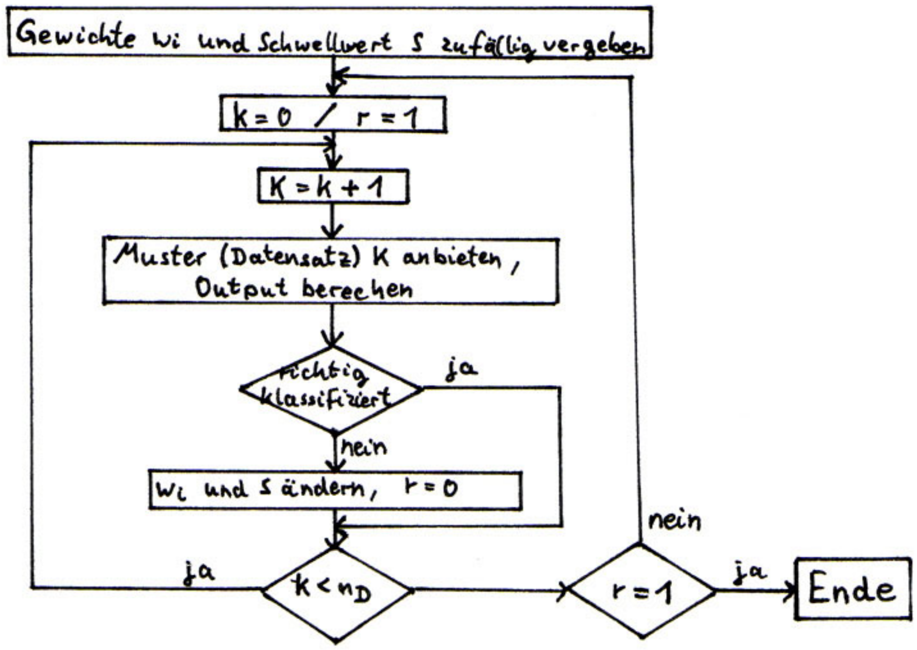
Verallgemeinerung:

- Bei 3 Eingängen wird durch eine Ebene separiert
- Bei 4 Eingängen ist das trennende Element ein 3D-Raum

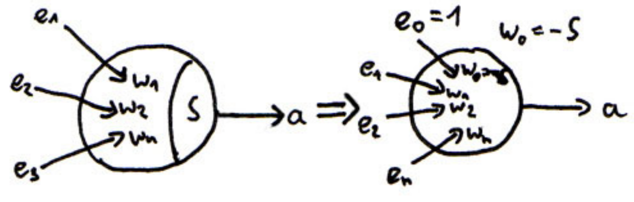
⇒ Einstellungen für Neuron lassen sich besser durch iteratives Lernverfahren bestimmen

15 Lernverfahren für Perzeptron

Überwachtes Lernen: Durch *Trail and Error* lernt das Perzeptron, die gestellte Aufgabe zu lösen.



Änderung im Neuronenmodell



Zu berechnen: $\sum_{i=1}^{n} e_i w_i - s \geq 0$ $\sum_{i=1}^{n} e_i w_i \geq 0$

15.1 Lernregel (Delta-Regel)

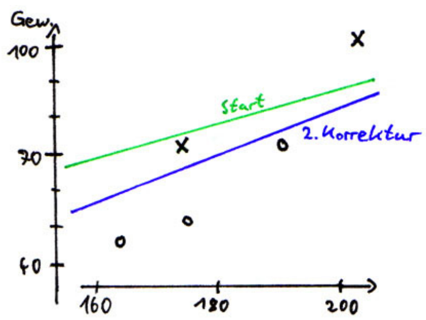
Es sein $\Delta = a_{soll} - a_{ist}$

Änderung der w_i dann nach folgender Formel: $w_i^{alt} + \alpha \cdot \Delta \cdot l_i$ $(i = 0, 1, ..., n)$ $\alpha =$ Lernrate

Beispiel:

	F_1	F_2	B_1	B_2	B_3
Größe (cm)	205	175	165	175	190
gewicht (kg)	100	70	45	50	70

e_1e_2	e_0w_0	e_1w_1	e_2w_2	$\sum$	a_{ist}	a_{soll}	$\alpha = 0,1; \Delta$
Gew.	800	-25	50				
205 100	800	-5125	5000	675	1	1	0
175 70	800	-4375	3500	-75	0	1	1
Neue Gew.	800 $+0,1 \cdot 1 \cdot 1$ $= 800,1$	-25 $+0,1 \cdot 1 \cdot 175$ $-7,5$	50 $+0,1 \cdot 1 \cdot 70$ 57				
165 45	$800,1$	$-1237,5$	2565	$2127,6$	1	1	-1
Neue Gew.	$800,1$ $-0,1 \cdot 1$ $= 800$	$-7,5$ $-0,1 \cdot 165$ -24	50 $-0,1 \cdot 45$ $52,5$				
175 50	800	-4200	2625	-775	0	0	0
190 70	800	-4560	3675	-85	0	0	0
205 100	800	-4920	5250	1130	1	1	0
175 70	800	-4200	3675	275	1	1	0
165 45	800	-3960	$2362,5$	$-797,5$	0	0	0



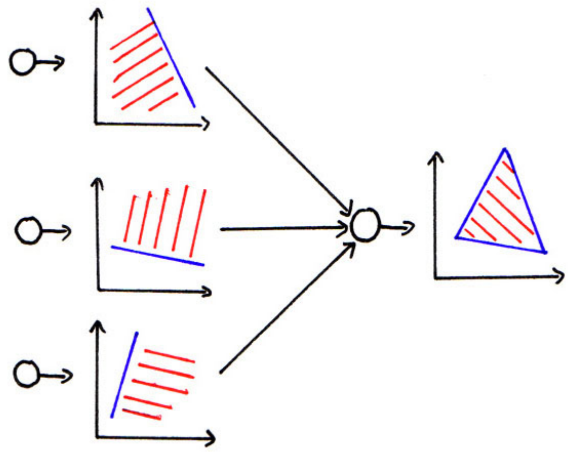
Entscheidungsgeraden:

Start $y = 0,5 \cdot x - 16$

1. Korrektur $y = \frac{7,5}{57} \cdot x + \frac{800,1}{57} = 0,132 \cdot x - 14,04$

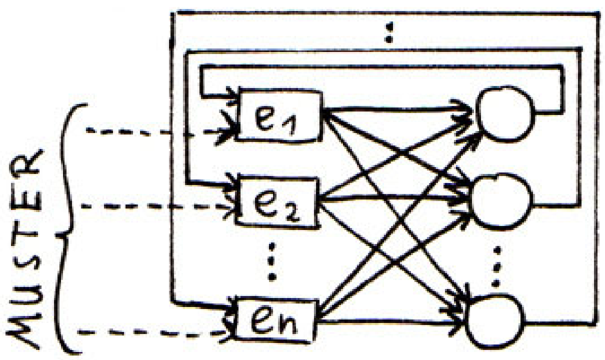
2. Korrektur $y = \frac{24}{52,5} \cdot x + \frac{800}{52,5} = 0,457 \cdot x - 15,24$

Die Delta-Regel versagt bei mehrstufigen Perzeptronen



16 Hopfield-Netze

Hopfield (1982), rückgekoppelte Netze



Arbeitsweise:

1. Phase: Anlegen eines Eingangsmusters
2. Phase: Ausgabe der euronen wird wieder als Eingabe angeboten
3. Phase: usw. wie 2. Phase

$\Leftarrow$ solange, bis Eingabe und Ausgabe übereinstimmen

$\Rightarrow$ **Auto-Assoziator**

Zweck: Erkennung von (unvollständigen) Mustern!

Aufgabenstellung

- Netz mit n Neuronen soll m verschiedene Muster erkennen
- Mustervektor $a^{(p)}(p = 1, ..., n)$ repräsentiert ein Schwarz-Weiß-Muster mit Codierung $1/-1$
- Neuronenoutput ist 1 oder -1, Schwellenwert $s = 0$

Lern-/ Konstruktionsregel:

1. Bestimme für jedes Muster die zugehörige Gewichtsmatrix nach
$$w_{ki}^{(p)} = \begin{cases} a_i^{(p)} \cdot a_k^{(p)} & \text{für } i \neq k \\ 0 & \text{für } i = k \end{cases}$$
2. Addiere alle so erzeugten Gewichtsmatrizen
3. Belege die Synapsengewichte mit den Werte der Summenmatrix

Beispiel: Ziffern in 3x5-Matrix

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

Muster für die Zahlen 1 und 9

$$\begin{array}{|ccc|}\hline - & - & + \\ - & + & + \\ + & - & + \\ - & - & + \\ - & - & + \\ \hline\end{array} \qquad \begin{array}{|ccc|}\hline + & + & + \\ + & - & + \\ + & + & + \\ - & - & + \\ + & + & + \\ \hline\end{array}$$

$-$ = weißes Pixel (-1)
$+$ = weißes Pixel (1)

1. Muster(1): $-1-1+1|-1+1+1|+1-1+1|-1-1+1|-1-1+1|$
2. Muster(9): $+1+1+1|+1-1+1|+1+1+1|-1-1+1|+1+1+1|$

Gewichtsmatrize für Muster (1)

0	+1	−1	+1	−1	−1	−1	+1	−1	+1	+1	−1	+1	+1	−1
	0	−1	+1	−1	−1	−1	+1	−1	+1	+1	−1	+1	+1	−1
		0	−1	+1	+1	+1	−1	+1	−1	−1	+1	−1	−1	+1
			0	−1	−1	−1	+1	−1	+1	+1	−1	+1	+1	−1
				0	+1	+1	−1	+1	−1	−1	+1	−1	−1	+1
					0	+1	−1	+1	−1	−1	+1	−1	−1	+1
						0	−1	+1	−1	−1	+1	−1	−1	+1
							0	−1	+1	+1	−1	+1	+1	−1
								0	−1	−1	+1	−1	−1	+1
									0	+1	−1	+1	+1	−1
										0	−1	+1	+1	−1
											0	−1	−1	+1
												0	+1	−1
													0	−1
														0

Gewichtsmatrize für Muster (9)

0	+1	+1	+1	−1	+1	+1	+1	+1	−1	−1	+1	+1	+1	+1
	0	+1	+1	−1	+1	+1	+1	+1	−1	−1	+1	+1	+1	+1
		0	+1	−1	+1	+1	+1	+1	−1	−1	+1	+1	+1	+1
			0	−1	+1	+1	+1	+1	−1	−1	+1	+1	+1	+1
				0	−1	−1	−1	−1	+1	+1	−1	−1	−1	−1
					0	+1	+1	+1	−1	−1	+1	+1	+1	+1
						0	+1	+1	−1	−1	+1	+1	+1	+1
							0	+1	−1	−1	+1	+1	+1	+1
								0	−1	−1	+1	+1	+1	+1
									0	+1	−1	−1	−1	−1
										0	−1	−1	−1	−1
											0	+1	+1	+1
												0	+1	+1
													0	+1
														0

$\Rightarrow$ Addition beider Matrizen

0	2	0	2	−2	0	0	2	0	0	0	0	2	2	0
2	0	0	2	−2	0	0	2	0	0	0	0	2	2	0
0	0	0	0	0	2	2	0	2	−2	−2	2	0	0	2
2	2	0	0	−2	0	0	2	0	0	0	0	2	2	0
−2	−2	0	−2	0	0	0	−2	0	0	0	0	−2	−2	0
0	0	2	0	0	0	2	0	2	−2	−2	2	0	0	2
0	0	2	0	0	2	0	0	2	−2	−2	2	0	0	2
2	2	0	2	−2	0	0	0	0	0	0	0	2	2	0
0	0	2	0	0	2	2	0	0	−2	−2	2	0	0	2
0	0	−2	0	0	−2	−2	0	−2	0	2	−2	0	0	−2
0	0	−2	0	0	−2	−2	0	−2	2	0	−2	0	0	−2
0	0	2	0	0	2	2	0	2	−2	−2	2	0	0	−2
2	2	0	2	−2	0	0	2	0	0	0	0	0	2	0
2	2	0	2	−2	0	0	2	0	0	0	0	2	0	0
0	0	2	0	0	2	2	0	2	−2	−2	2	0	0	0

Erkennungsphase:

1. Präge dem Netz ein (unvollständiges, verrauschtes) Musser auf: $e^{(0)}$
2. Berechne $e_i^{neu} = sgn(\sum_{j=1}^{n} w_{ij} e_i^{alt})$ mit $sgn(x) = \begin{cases} 1 & \text{für } x \geq 0 \\ -1 & \text{für } x < 0 \end{cases}$

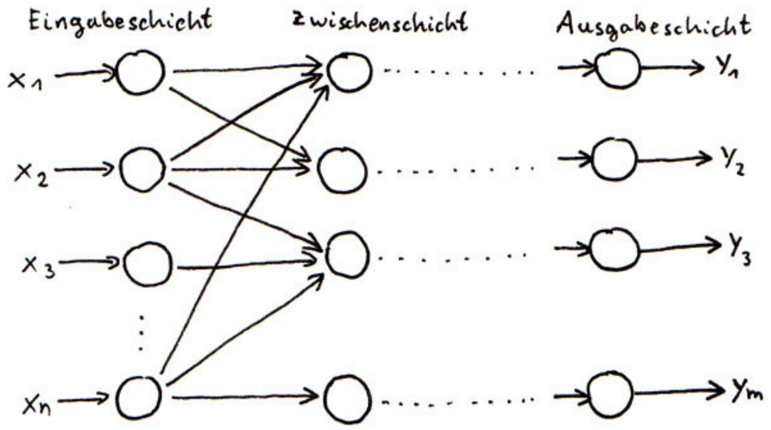
$$\begin{array}{rl} \text{Fehlermuster:} & -1+1+1|-1+1+1|+1+1+1|-1-1-1|-1+1+1| \\ \downarrow & +1+1+1|+1+1+1|+1+1+1|-1-1+1|+1+1+1| \\ \downarrow & +1+1+1|+1-1+1|+1+1+1|-1-1+1|+1+1+1| \end{array}$$

$$\boxed{\begin{array}{c} -++ \\ --+ \\ +++ \\ --- \\ -++ \end{array}} \rightarrow \boxed{\begin{array}{c} +++ \\ +++ \\ +++ \\ --+ \\ +++ \end{array}} \rightarrow \boxed{\begin{array}{c} +++ \\ +-+ \\ +++ \\ --+ \\ +++ \end{array}}$$

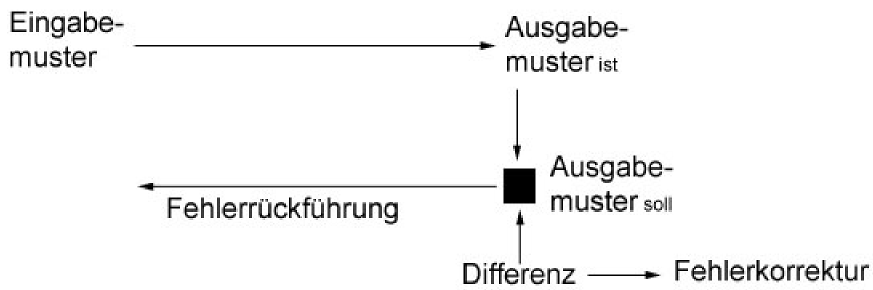
Anmerkungen:

- Anzahl der Neuronen muss hinreichend groß sein, um die Muster sicher erkennen zu können, d.h. es sollte gelten $m \leq 0,146 \cdot n$
- Schwierigkeiten bei translatierten (verschobenen) Mustern

17 Backprobagation-Netze

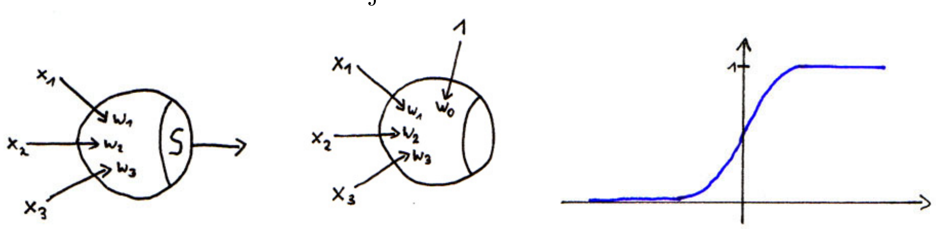


- Mehrschichtiges Neuronennetz
- Neuronen in einer Schich sind nicht untereinander verbunden
- Kein Neuron ist mit sich selbst verbunden
- Verbindungen bestehen ausschließlich von einer zur nachfolgenden Schicht



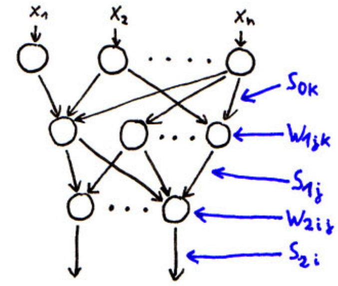
Für Backprobagaion erforderlich: Neuronen mit stetigem Antwortverhalten

bisher jetzt Antwortfunktion



Antwortbereich: $0 < Output < 1$

- „Feuern“ eines Neuron bei z. B. $Output > 0,8$
- „Ruhezustand“ z. B. bei $Output < 0,2$



s_{0_k}: Aktivität des k-ten Neurons in Schicht 0
s_{1_i}: Aktivität von Neuron j in Schicht 1
s_{2_j}: Aktivität von Neuron i in Schicht 2
$w_{1_{jk}}$: Gewicht des k-ten Eingangs von Neuron j in Schicht 1
$w_{2_{ij}}$: Gewicht des j-ten Eingangs von Neuron i in Schicht 2

Es gilt: $S_{1_j} = \sigma(\sum_{k=0}^{n} w_{1_{jk}} \cdot s_{0_k})$
$S_{2_j} = \sigma(\sum_{j=0}^{i} w_{2_{ij}} \cdot s_{1_j})$

Gegeben: Satz von p Eingabemustern $X^\nu(\nu = 1, ..., p)$
mit zugehörigen Ausgabemustern $Y^\nu(\nu = 1, ..., p)$

Das Netz ist so zu trainieren, dass der quadratische Fehler E minimiert wird, mit

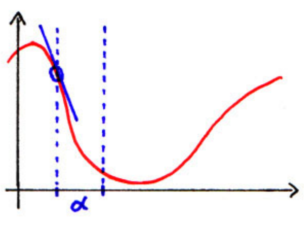
$$E = \frac{1}{2} \underbrace{\sum_{\nu=1}^{p} \underbrace{\sum_{i=1}^{m} (y_i^\nu - s_{2_i}(x^\nu))^2}_{Summe\ der\ Abweichung}}_{Summe\ ber\ alle\ Muster}$$

$$\Rightarrow E = \frac{1}{2} \sum_{\nu=1}^{p} \sum_{i=1}^{m} (y_i^\nu - \sigma(\sum_{j=1}^{l} w_{2_{ij}} \cdot s_{1_j}))^2, \text{ d.h. } E = f(w_{2_{ij}}, w_{1_{jk}}$$

Forderung: $\frac{\delta E}{\delta w_{ab}} = 0$ für alle $w_{ab} = \{w_{2_{ij}}, w_{1_{jk}}\}$

Lösung (näherungsweise): Durch ein Gradientenabstiegsverfahren

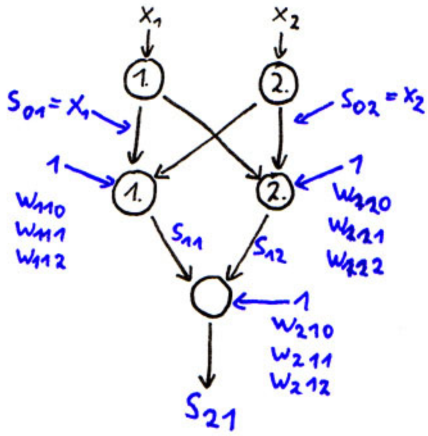
Für unser Problem ausgehend vom gerade erreichten Punkt, bewegt man sich für alle w_{ab} um $\Delta w_{ab} = -\alpha \frac{\delta E}{\delta w_{ab}}$ in w_{ab}-Richtung $(\alpha > 0, klein)$



Nach jeder Verarbeitung eines Musters werden folgende Korrekturen berechnet:

$$\Delta w_{2_{ij}} = \alpha \cdot \epsilon_i^\nu \cdot s_{1_j} \cdot s_{2_i} \cdot (1 - s_{2_i});\ \ \epsilon_i^\nu = y_{i_{soll}}^\nu - y_{i_{ist}}^\nu$$
$$\Delta w_{2_{ij}} = \alpha \cdot \sum_{i=1}^{m} \epsilon_i^\nu \cdot s_{0_k} \cdot s_{2_i} \cdot (1 - s_{2_i}) \cdot w_{2_{ij}} \cdot s_{1_j} \cdot (1 - s_{1_j})$$

17.1 Beispiel: BP-Netz zum Lernen des XOR-Problems



Es gilt:

$$s_{11} = \sigma(w_{110} + w_{111} \cdot x_1 + w_{112} \cdot x_2)$$
$$s_{12} = \sigma(w_{120} + w_{121} \cdot x_1 + w_{122} \cdot x_2)$$
$$s_{21} = \sigma(w_{210} + w_{211} \cdot x_1 + w_{212} \cdot x_2)$$

Korrekturformeln: $f = \alpha \cdot \epsilon_1^{\nu} \cdot s_{21} \cdot (1 - s_{21})$

$$\Delta w_{210} = f$$
$$\Delta w_{211} = f \cdot s_{11}$$
$$\Delta w_{212} = f \cdot s_{12}$$
$$\Delta w_{110} = f \cdot w_{211} \cdot s_{11} \cdot (1 - s_{11})$$
$$\Delta w_{111} = f \cdot x_1 \cdot w_{211} \cdot s_{11} \cdot (1 - s_{11})$$
$$\Delta w_{112} = f \cdot x_2 \cdot w_{211} \cdot s_{11} \cdot (1 - s_{11})$$
$$\Delta w_{120} = f \cdot w_{212} \cdot s_{12} \cdot (1 - s_{12})$$
$$\Delta w_{121} = f \cdot x_1 \cdot w_{212} \cdot s_{12} \cdot (1 - s_{12})$$
$$\Delta w_{122} = f \cdot x_2 \cdot w_{212} \cdot s_{12} \cdot (1 - s_{12})$$

Musterpool:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Startwerte:

$\alpha = 0.1$

$w_{110} = -2 \quad w_{111} = -3 \quad w_{112} = 3$

$w_{120} = -2 \quad w_{121} = 3 \quad w_{122} = -3$

$w_{210} = -3 \quad w_{211} = 5,94 \quad w_{212} = 6$

1. Muster(0,0):

$$s_{11} = \sigma(-2 + 0 + 0) = 0.1192$$
$$s_{12} = \sigma(-2 + 0 + 0) = 0.1192$$
$$s_{21} = \sigma(-3 + 5.94 \cdot 0.1192 + 6 \cdot 0.1192) = 0.17126 \quad \surd \text{ da } < 0.2$$

$\Rightarrow$ keine Korrektur

2. Muster(0,1): $s_{11} = \sigma(-2+0+3) = 0.73106$
$s_{12} = \sigma(-2+0-3) = 0.00669$
$s_{21} = \sigma(-3+5.94\cdot 0.73106 + 6\cdot 0.0069) = 0.79941 \quad f$ da < 0.8
$\Rightarrow$ Korrektur

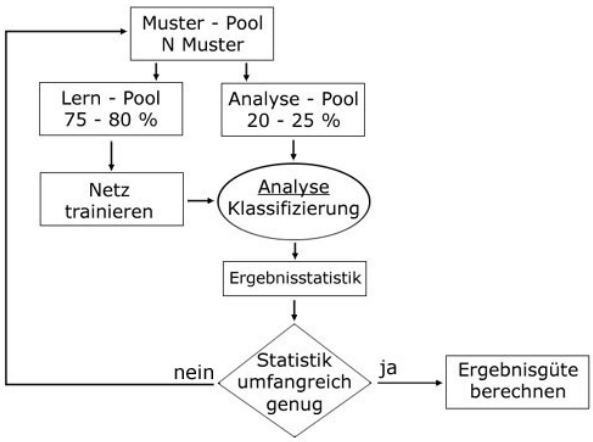
Korrektur: $f = 0.1\cdot(1-0.79941)\cdot 0.79941\cdot(1-0.79941) = 0.003217$

$\Delta w_{210} = 0.003217 \quad \Delta w_{211} = 0.002352 \quad \Delta w_{212} = 0.000022$
$\Delta w_{110} = 0.003757 \quad \Delta w_{111} = 0.0 \quad \Delta w_{112} = 0.003757$
$\Delta w_{120} = 0.0001283 \quad \Delta w_{121} = 0.0 \quad \Delta w_{122} = 0.00001283$

$w_{110} = -1.99624 \quad w_{111} = -3 \quad w_{112} = 3.00376$
$w_{120} = -1.98987 \quad w_{121} = 3 \quad w_{122} = -2.99987$
$w_{210} = -2.99678 \quad w_{211} = 5,94235 \quad w_{212} = 6.00002$

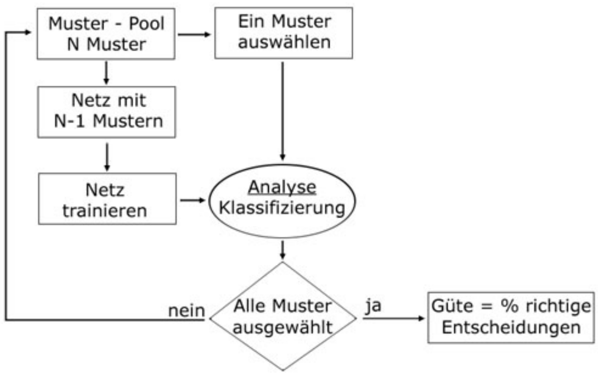
17.2 Beurteilung der Entscheidungsgüte von Backprobagation-Netzen

Splitting-Pool-Methode



Problem: Beurteilt wird nur das Netz, welches 20-25

Leaving-One-Out-Methode



Problem: Hoher Rechenaufwand

18 Kritische Würdigung

Vorteile:

- kein mathematisches Entscheidungsmodell erforderlich
- Fehlertoleranz (Hopfield-Netze)
- Fallbasiertes Lernen, kann mit Hilfe neuer Fälle noch verbessert werden
- Echtzeitfähgkeit bezüglich Entscheidungsfindung

Nachteile:

- Entscheidungsfindung ist nicht nachvollziehbar
- hoher Trainigsaufwand
- viele Parameter, an denen „herumgespielt“werden kann, ohne dass die Effekte leicht zu kontrollieren sind
- Konvergenz nicht immer gesichert
- Entscheidunsgüte häufig schlechter als erwartet!