

Neuro-Fuzzy-Systeme

- Nachteil neuronaler Netze:
 - Ergebnisse sind schlecht interpretierbar (black box)
 - Vorwissen kann schlecht dargestellt werden
- Ausweg:
 - hybrides System, bei dem NN mit einem regelbasierten System gekoppelt sind
 - ein mögliches Verfahren: Neuro-Fuzzy-Systeme

Kurzeinführung in die Fuzzy-Theorie

- **Klassische Logik:** nur Wahrheitswerte *wahr* und *falsch*
- **Klassische Mengenlehre:** entweder *ist Element* oder *nicht*
- Zweiwertigkeit dieser Theorien: oft unangemessen
- Beispiel: **Sorites-Paradoxon** (griech. *sorites*: Haufen)
wahr: “Eine Milliarde Sandkörner sind ein Sandhaufen.”
wahr: “Wenn man von einem Sandhaufen ein Sandkorn entfernt, bleibt ein Sandhaufen übrig.”
- *wahr*: 999 999 999 “Sandkörner sind ein Sandhaufen.”
- mehrfache Wiederholung des gleichen Schlusses:
falsch: “1 Sandkorn ist ein Sandhaufen.”
- Frage: Bei welcher Anzahl Sandkörner ist Schluss nicht wahrheitsbewahrend?

Kurzeinführung in die Fuzzy-Theorie

- Offenbar: keine genau bestimmte Anzahl Sandkörner, bei der der Schluss auf nächstkleinere Anzahl falsch ist
- Problem: Begriffe der natürlichen Sprache (z.B. “Sandhaufen”, “kahlköpfig”, “warm”, “schnell”, “hoher Druck”, “leicht” etc.) sind **vage**
- beachte: vage Begriffe sind *unexakt*, aber nicht *unbrauchbar*
 - auch für vage Begriffe: Situationen/Objekte, auf die sie *sicher anwendbar* sind und solche, auf die sie *sicher nicht anwendbar* sind
 - dazwischen: **Penumbra** (lat. für *Halbschatten*) von Situationen, in denen es unklar ist, ob die Begriffe anwendbar sind, oder in denen sie nur mit Einschränkungen anwendbar sind (“kleiner Sandhaufen”).
 - Fuzzy-Theorie: mathematische Modellierung der Penumbra

- Erweiterung der klassischen Logik um Zwischenwerte zwischen *wahr* und *falsch*
- Wahrheitswert: jeder Wert aus $[0, 1]$, wobei $0 \equiv \textit{falsch}$ und $1 \equiv \textit{wahr}$

- **Erweiterung der logischen Operatoren**

Klassische Logik		Fuzzy-Logik		
Operation	Notation	Operation	Notation	Beispiel
Negation	$\neg a$	Fuzzy-Negation	$\sim a$	$1 - a$
Konjunktion	$a \wedge b$	t -Norm	$\top(a, b)$	$\min(a, b)$
Disjunktion	$a \vee b$	t -Konorm	$\perp(a, b)$	$\max(a, b)$

- **Prinzipien** der Erweiterung:

- für Extremwerte 0 und 1 sollen sich Operationen genauso verhalten wie ihre klassischen Vorbilder (Rand-/Eckbedingungen)
- für Zwischenwerte soll ihr Verhalten monoton sein
- Gesetze der klassischen Logik sollen (fast alle) erhalten werden

- klassische Mengenlehre basiert auf Begriff “*ist Element von*” (\in)
- alternativ: Zugehörigkeit zu Menge mit *Indikatorfunktion* beschreibbar: sei X eine Menge, dann heißt

$$I_M : X \rightarrow \{0, 1\}, \quad I_M(x) = \begin{cases} 1, & \text{falls } x \in X, \\ 0, & \text{sonst,} \end{cases}$$

Indikatorfunktion der Menge M bzgl. Grundmenge X

- in Fuzzy-Mengenlehre: ersetze Indikatorfunktion durch *Zugehörigkeitsfunktion*: sei X (klassische/scharfe) Menge, dann heißt

$$\mu_M : X \rightarrow [0, 1], \quad \mu_M(x) \equiv \text{Zugehörigkeitsgrad von } x \text{ zu } M,$$

Zugehörigkeitsfunktion (membership function) der **Fuzzy-Menge** M bzgl. der *Grundmenge* X

- Fuzzy-Menge: definiert über ihre Zugehörigkeitsfunktion

Formale Definition einer Fuzzy-Menge

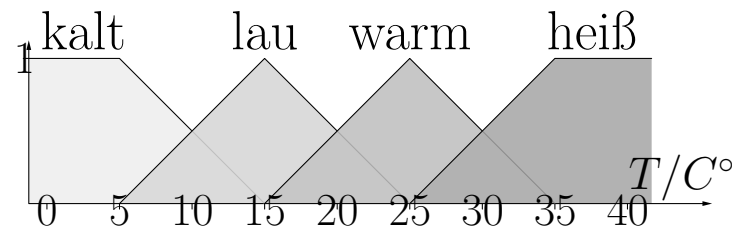
1. Eine Fuzzy-Menge $\mu \in X \neq \emptyset$ ist eine Funktion, die aus der Referenzmenge X in das Einheitsintervall abbildet, d.h. $\mu : X \rightarrow [0, 1]$.
2. $\mathcal{F}(X)$ stellt die Menge aller Fuzzy-Mengen von X dar, d.h. $\mathcal{F}(X) \stackrel{\text{def}}{=} \{\mu \mid \mu : X \rightarrow [0, 1]\}$.

Fuzzy-Partitionen und Linguistische Variablen

- um Wertebereich durch sprachliche (linguistische) Ausdrücke beschreiben zu können, wird er mithilfe von Fuzzy-Mengen fuzzy-partitioniert
- jeder Fuzzy-Menge der Partitionierung ist ein linguistischer Term zugeordnet
- übliche Bedingung: an jedem Punkt müssen sich Zugehörigkeitsgrade aller Fuzzy-Mengen zu 1 addieren

Beispiel: Fuzzy-Partitionierung für Temperaturen

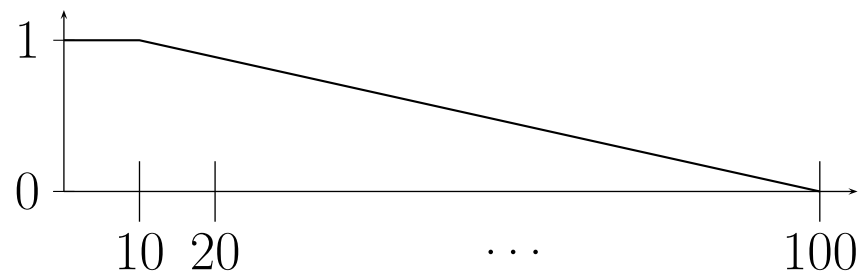
Linguistische Variable mit den Werten *kalt*, *lau*, *warm* und *heiß*.



Subjektive Definition einer Fuzzymenge – Beispiel

- X Menge der Magdeburger Einwohner im Alter zwischen 10 und 100 Jahren
- $Y = \{1, \dots, 100\}$
- $j(y)$ Anzahl der Einwohner die y alt sind, die sich als “jung” bezeichnen
- $n(y)$ Gesamtzahl der Einwohner im Alter y

$$\mu : Y \rightarrow [0, 1], \quad \mu(y) = \begin{cases} \frac{j(y)}{n(y)} & \text{if } y > 10 \\ 1 & \text{if } y \leq 10 \end{cases}$$



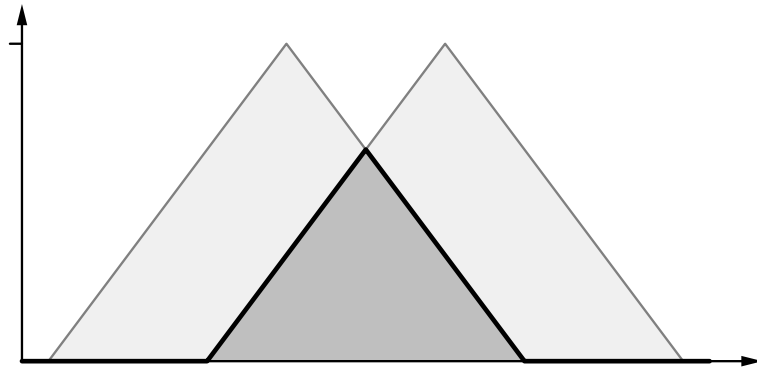
Operationen

- wie beim Übergang von klassischer Logik zur Fuzzy-Logik: hier auch Erweiterung der Operationen nötig
- **dieser Erweiterung:**
greife auf logische Definition der Operationen zurück
- elementweise Anwendung der logischen Operatoren
- (Fuzzy-)Mengen A und B über Grundmenge X

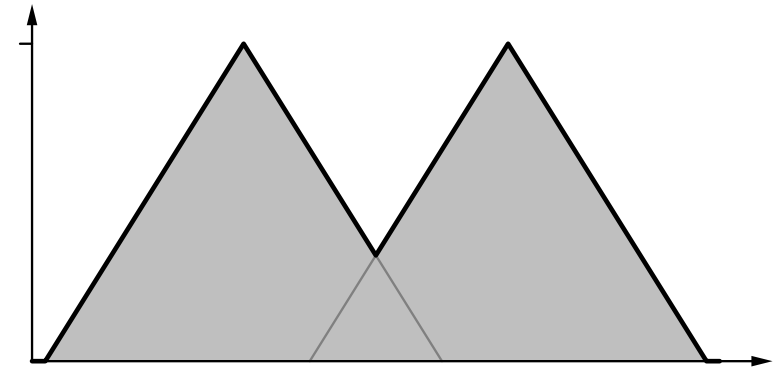
Komplement	klassisch	$\bar{A} = \{x \in X \mid x \notin A\}$
	fuzzy	$\forall x \in X: \mu_{\bar{A}}(x) = \sim\mu_A(x)$
Schnitt	klassisch	$A \cap B = \{x \in X \mid x \in A \wedge x \in B\}$
	fuzzy	$\forall x \in X: \mu_{A \cap B}(x) = \top(\mu_A(x), \mu_B(x))$
Vereinigung	klassisch	$A \cup B = \{x \in X \mid x \in A \vee x \in B\}$
	fuzzy	$\forall x \in X: \mu_{A \cup B}(x) = \perp(\mu_A(x), \mu_B(x))$

Fuzzy-Schnitt und Fuzzy-Vereinigung

Beispiele für Schnitt und Vereinigung

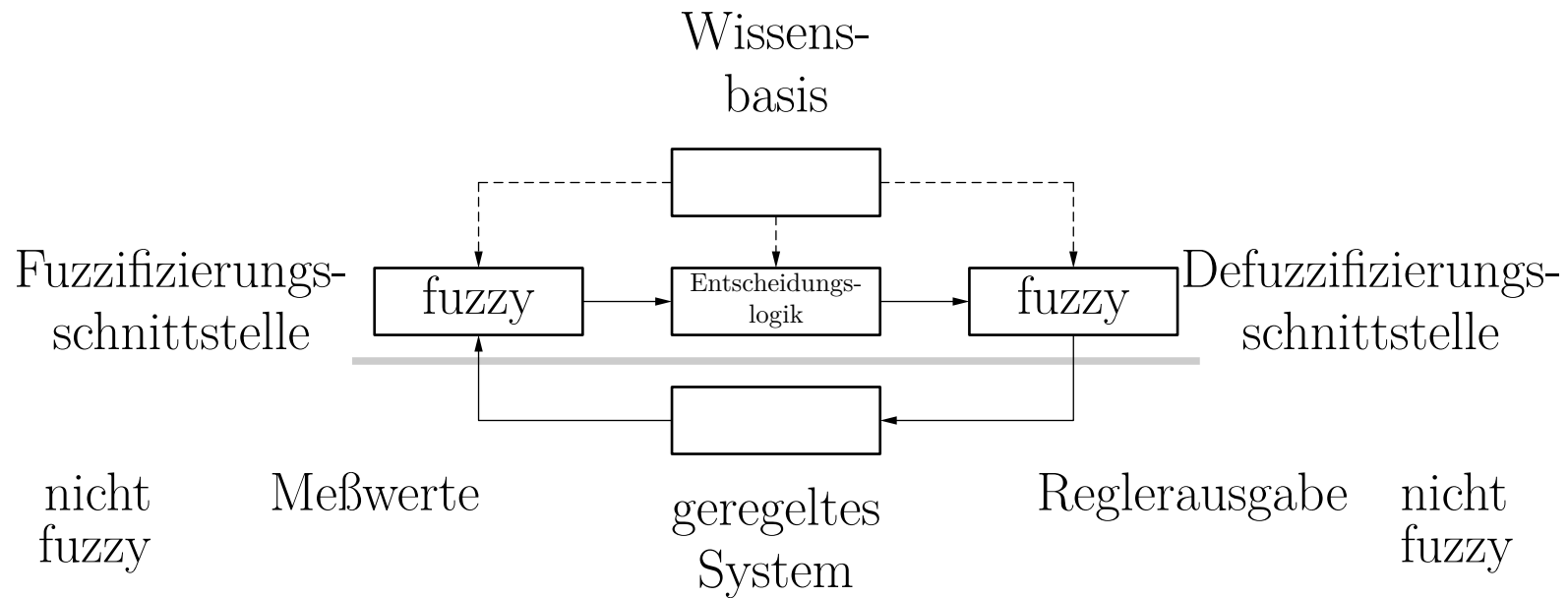


Fuzzy-Schnitt (min)



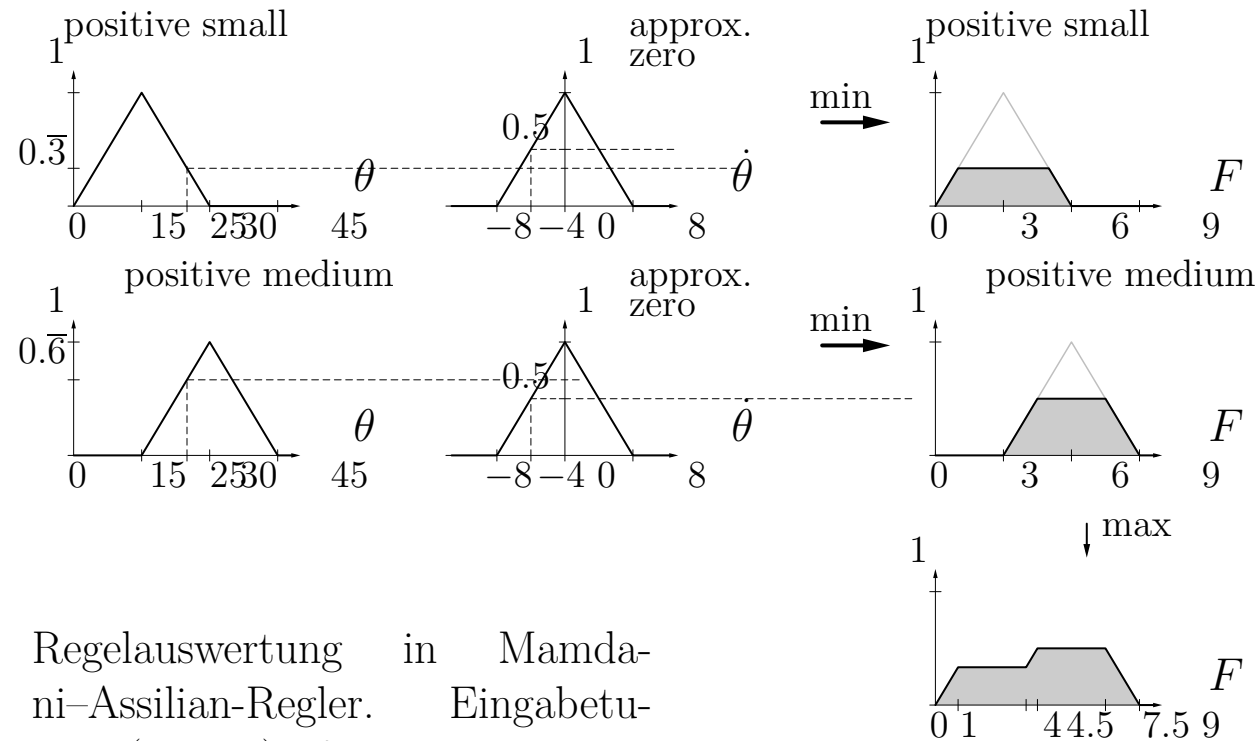
Fuzzy-Vereinigung (max)

Architektur eines Fuzzy-Reglers



- Wissensbasis enthält Fuzzy-Regeln für Steuerung und Fuzzy-Partitionen der Wertebereiche der Variablen
- Fuzzy-Regel: **if** X_1 **is** $A_{i_1}^{(1)}$ **and** ... **and** X_n **is** $A_{i_n}^{(n)}$ **then** Y **is** B .
 X_1, \dots, X_n sind Messgrößen und Y ist Stellgröße
 $A_{i_k}^{(k)}$, B : linguistische Terme (mit Fuzzy-Mengen assoziiert)

Fuzzy-Regelung nach Mamdani-Assilian



Regelauswertung in Mamdani-Assilian-Regler. Eingabetupel $(25, -4)$ führt zur rechts gezeigten unscharfen Ausgabe. Aus dieser Fuzzy-Menge wird entsprechender Ausgabewert durch Defuzzifizierung bestimmt, z.B. durch die Schwerpunktmethode (COG).

Der Aufbau eines Fuzzy-Systems benötigt:

- Vorwissen (Fuzzy-Regeln, Fuzzy-Mengen)
- Manuelle Anpassungen, die zeitaufwendig und fehlerträchtig sind

⇒ Unterstütze diesen Prozess durch Lernverfahren:

- Erlernen von Fuzzy-Regeln (Struktur-Lernen)
- Erlernen von Fuzzy-Mengen (Parameter-Lernen)

Ansatz mit künstlichen neuronalen Netzen kann genutzt werden

Fallstudie: Aktienkursvorhersage

Prognose der täglichen relativen Änderungen des DAX, aufbauend auf Zeitreihen von Börsen-Indizes im Zeitraum von 1986 bis 1997

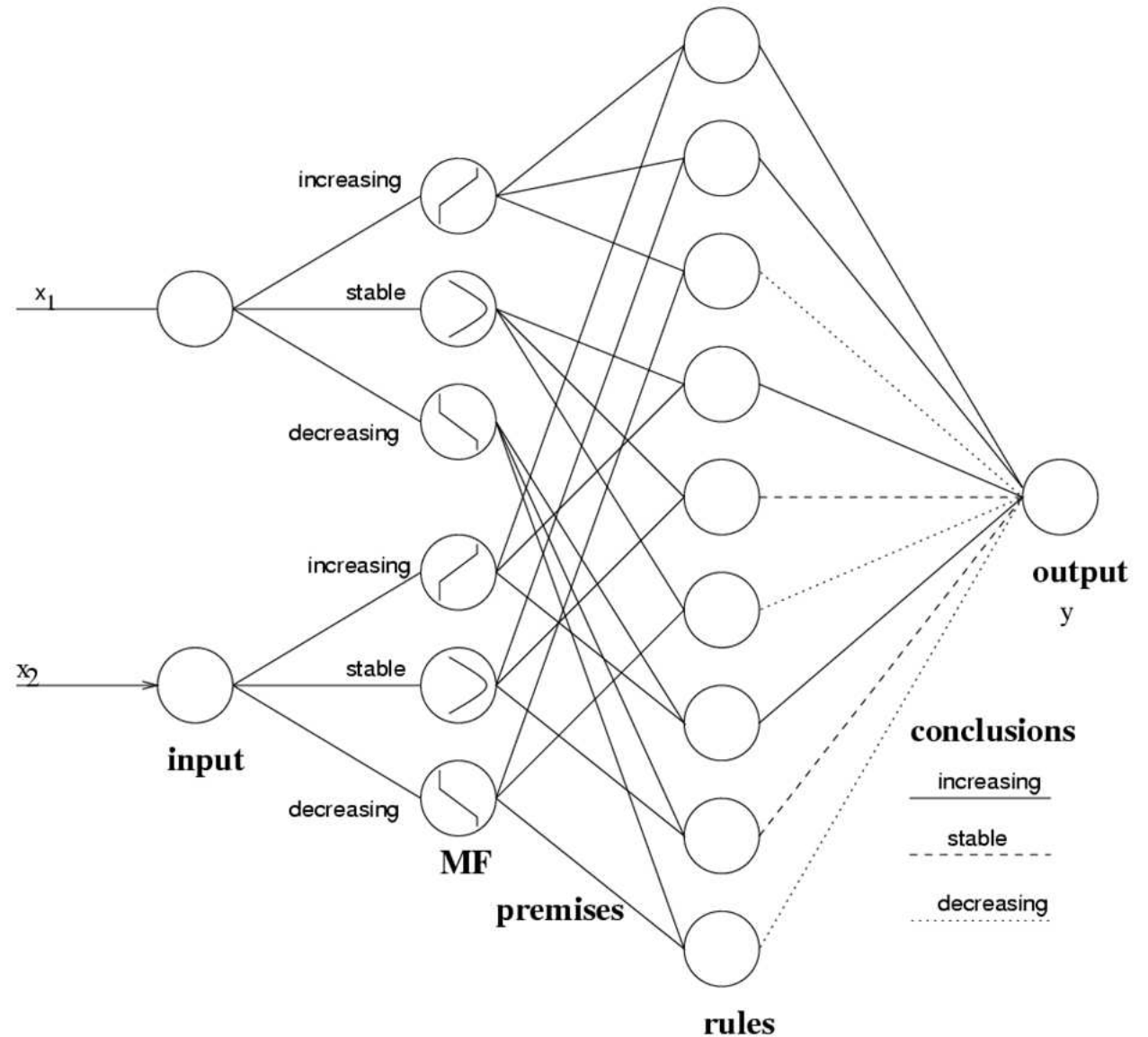
DAX	Composite-DAX
German 3 months interest rate	Return Germany
German Morgan-Stanley index	Dow Jones industrial index
DM / US-\$	US treasure bonds
gold price	Japanese Nikkei-Index
European Morgan-Stanley-Index	Price earning ratio

Fuzzy-Regeln im Finanzbereich

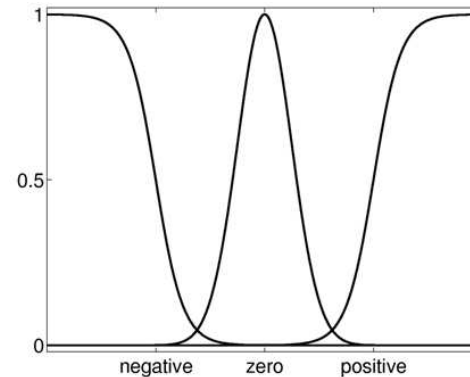
- trend rule
 - IF DAX = decreasing AND US-\$ = decreasing
 - THEN DAX prediction = decreasing
 - WITH high certainty
- turning point rule
 - IF DAX = decreasing AND US-\$ = increasing
 - THEN DAX prediction = increasing
 - WITH low certainty
- delay rule
 - IF DAX = stable AND US-\$ = decreasing
 - THEN DAX prognosis = decreasing
 - WITH very high certainty
- in general
 - IF x_1 is μ_1 AND x_2 is μ_2 AND ... AND x_n is μ_n
 - THEN $y = \eta$
 - WITH weight k

Neuro-Fuzzy-Architektur

Ausschnitt eines NF-Systems.



Von Regeln zu Neuronalen Netzen



1. Bewertung von Zugehörigkeitsgraden
2. Bewertung von Regeln (Regelaktivität)

$$\mu_l = \mathbb{R}^n \rightarrow [0, 1]^r, \quad \underline{x} \Rightarrow \prod_{j=1}^{D_l} \mu_{c,s}^{(j)}(x_i)$$

3. Akkumulation von Regeleingaben, Normalisierung

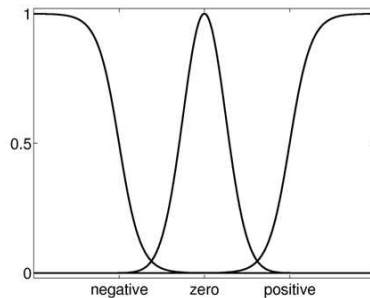
$$\text{NF} : \mathbb{R}^n \rightarrow \mathbb{R}, \quad \underline{x} \Rightarrow \sum_{l=1}^r w_l \frac{k_l \mu_l(\underline{x})}{\sum_{j=1}^r k_j \mu_j(\underline{x})}$$

Dimensionsreduktion des Gewichtsraums

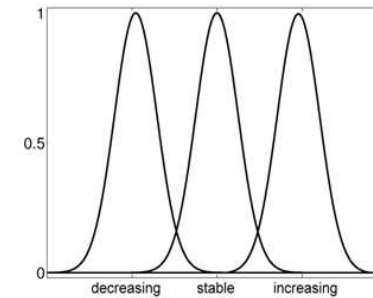
1. Zugehörigkeitsfunktionen verschiedener Eingaben teilen ihre Parameter untereinander, z.B.

$$\mu_{\text{DAX}}^{\text{stabil}} = \mu_{\text{C-DAX}}^{\text{stabil}}$$

2. Zugehörigkeitsfunktionen derselben Eingabevariable dürfen nicht einander passieren, sondern müssen ihre Originalreihenfolge beibehalten, d.h.



$$\mu_{\text{decreasing}} < \mu_{\text{stable}} < \mu_{\text{increasing}}$$



Vorteile:

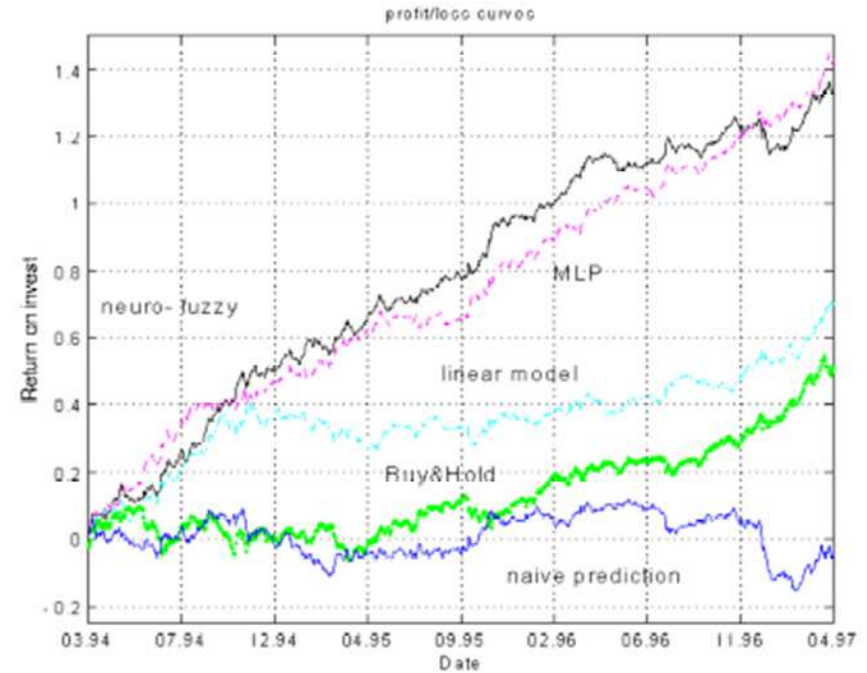
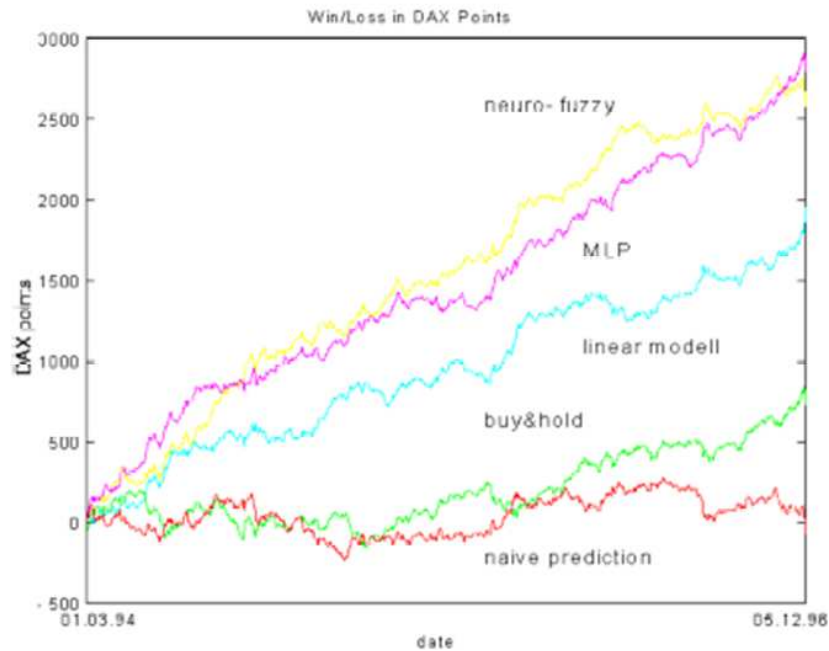
- Die optimierte Regelbasis ist immer noch interpretierbar.
- Die Anzahl freier Parameter wurde reduziert.

- Die Parameter der Fuzzy-Mengen,
- die Gewichte
- und die Regelwichtigkeiten

werden durch ein Backpropagation-Verfahren ermittelt. Es werden Pruning-Verfahren genutzt.

Gewinnkurven

- verschiedene Modelle
- Validierungsdaten: März 1994 bis April 1997



Fallstudie: medizinische Diagnose nach NEFCLASS-Einführung

- Ergebnisse von Patienten, die auf Brustkrebs getestet wurden (Wisconsin Breast Cancer Data)
- Entscheidungsunterstützung: liegt ein gutartiger oder bösartiger Fall vor?
- Ein Chirurg muß die Klassifikation auf ihre Plausibilität hin überprüfen können.
- Es wird nach einem einfachen und interpretierbaren Klassifikator gesucht

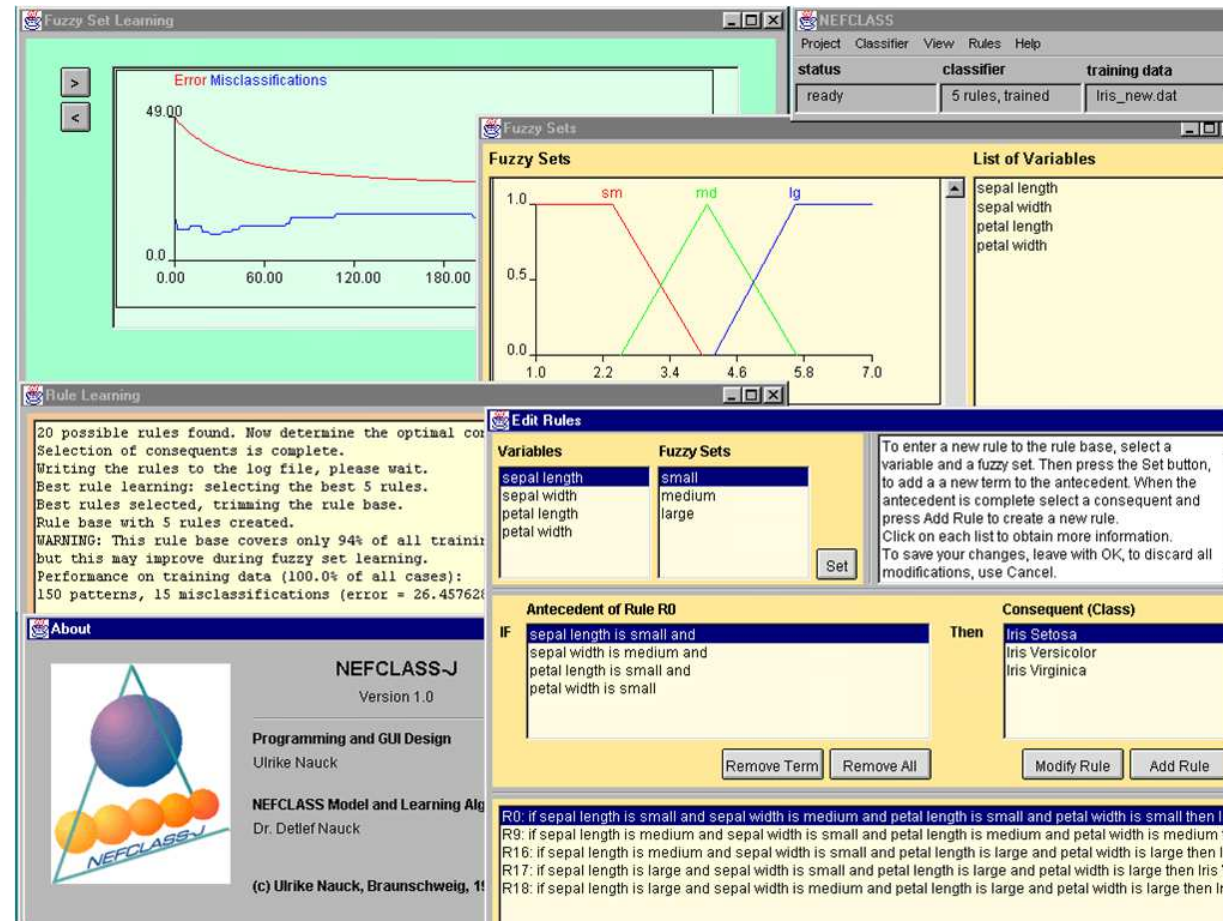
⇒ Wissensentdeckung

Fallstudie: WBC-Daten

- 699 Fälle (16 Fälle mit fehlenden Werten)
- 2 Klassen: gutartig(458), bösartig (241)
- 9 Attribute mit Werten in $\{1, \dots, 10\}$ (ordinale Skala, aber normalerweise numerisch interpretiert)
- Experiment: x_3 und x_6 werden als nominale Attribute interpretiert
- x_3 und x_6 werden oft als “wichtige” Attribute angesehen

Anwendung von NEFCLASS-J

- Werkzeug zur Entwicklung von NF-Klassifikatoren
- Java-Implementierung
- frei verfügbar zu Forschungszwecken
- Projekt in unserer Gruppe gestartet



<http://fuzzy.cs.ovgu.de/nefclass/nefclass-j/>

NEFCLASS: Neuro-Fuzzy-Klassifikator

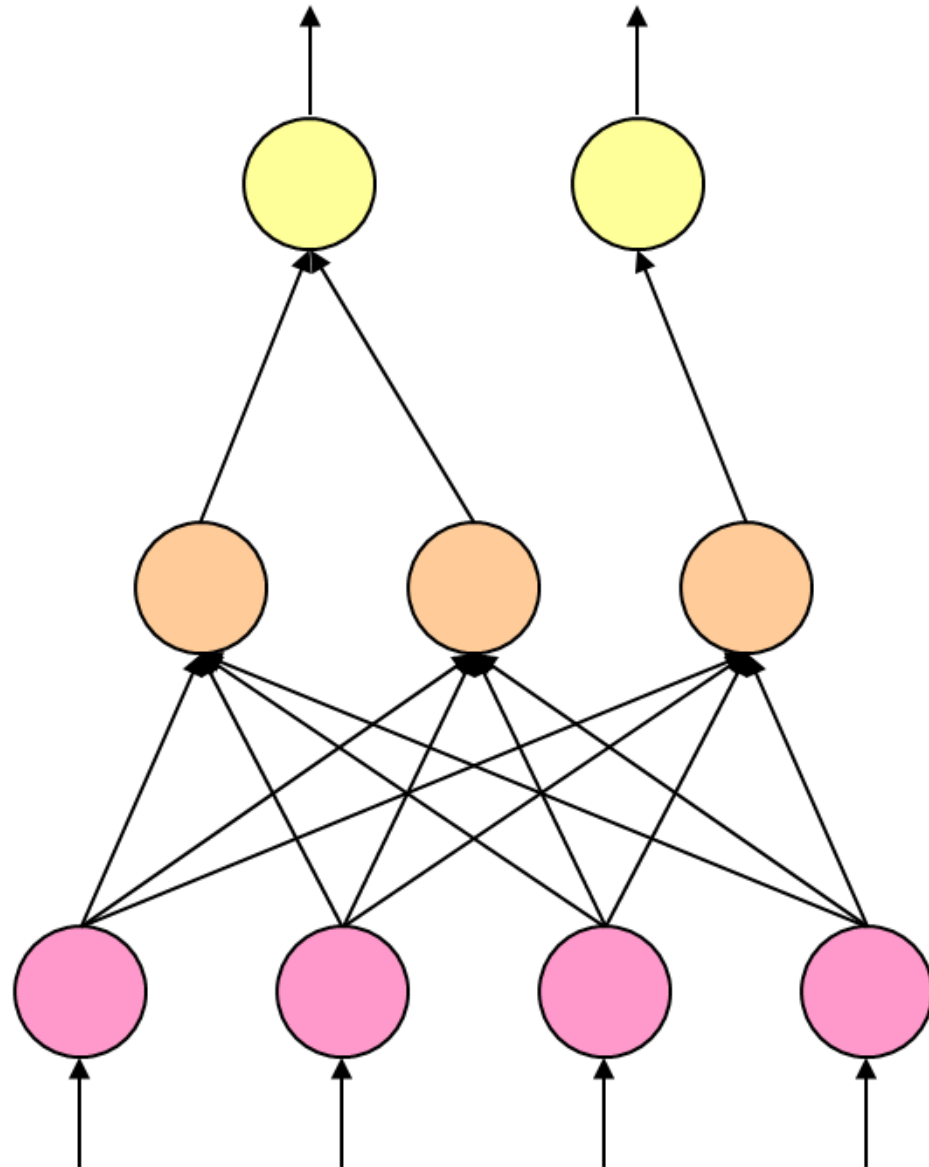
Ausgabevariablen

ungewichtete Verbindungen

Fuzzy-Regeln

Fuzzy-Mengen (Antezedens)

Eingabeattribute (Variablen)



NEFCLASS: Merkmale

- automatische Erstellung der Fuzzy-Regelbasis aus Daten
- Trainieren verschiedener Formen von Fuzzy-Mengen
- Verarbeiten von numerischen und symbolischen Attributen
- Behandlung von fehlenden Werten (kein Entfernen)
- automatische Beschneidungsstrategien
- Verschmelzen von Expertenwissen und Daten

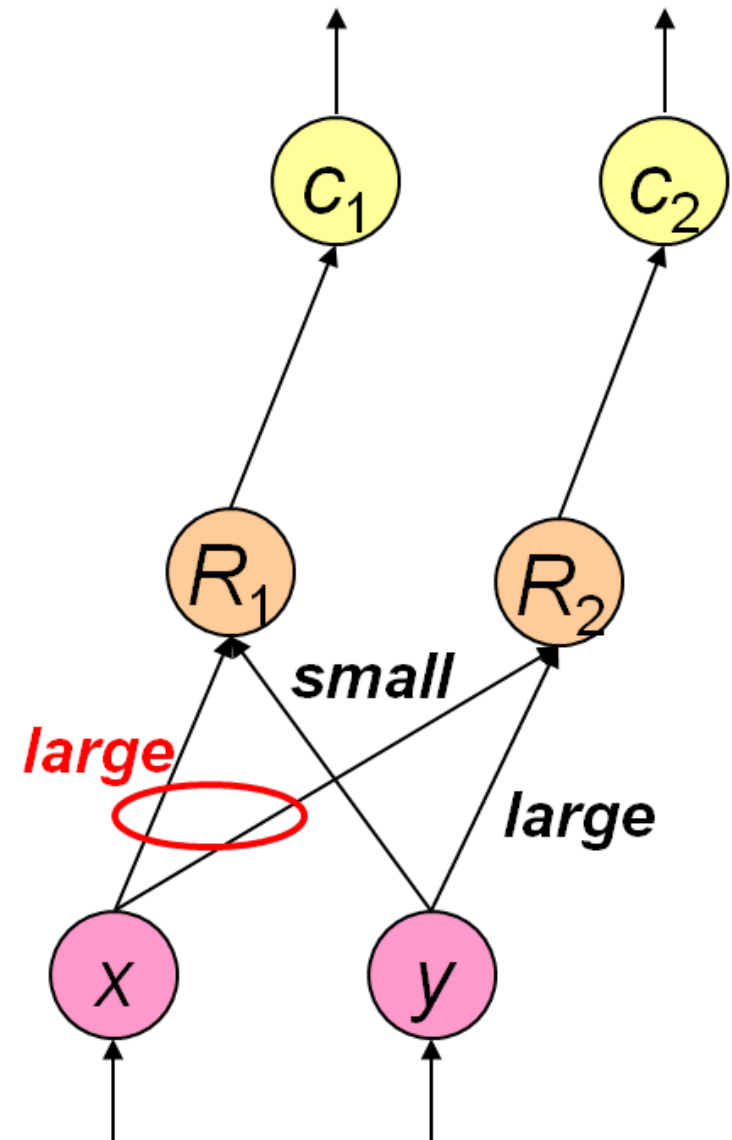
Darstellung von Fuzzy-Regeln

Beispiel: 2 Regeln

R_1 : if x is *large* and y is *small*, then class is c_1

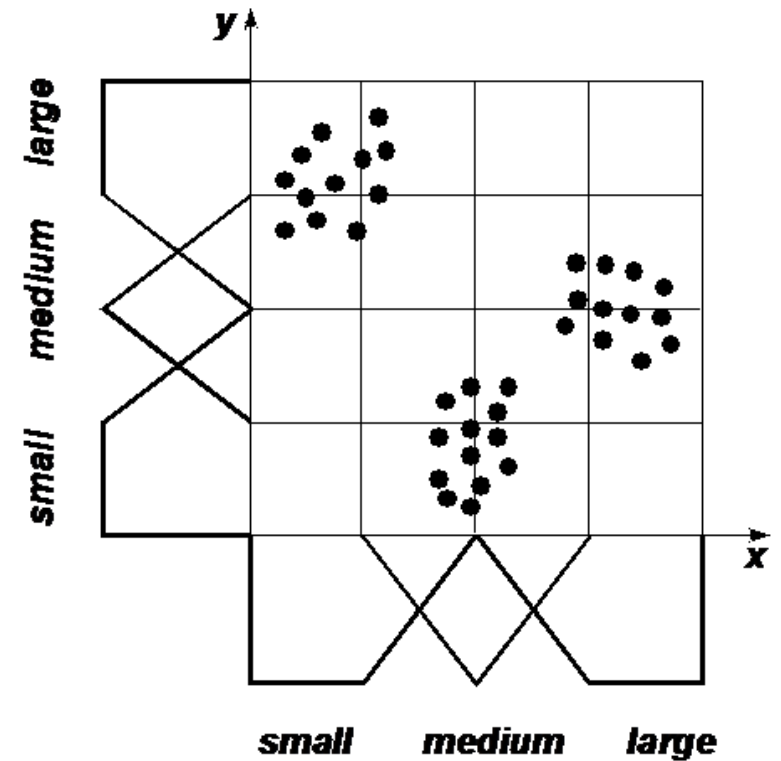
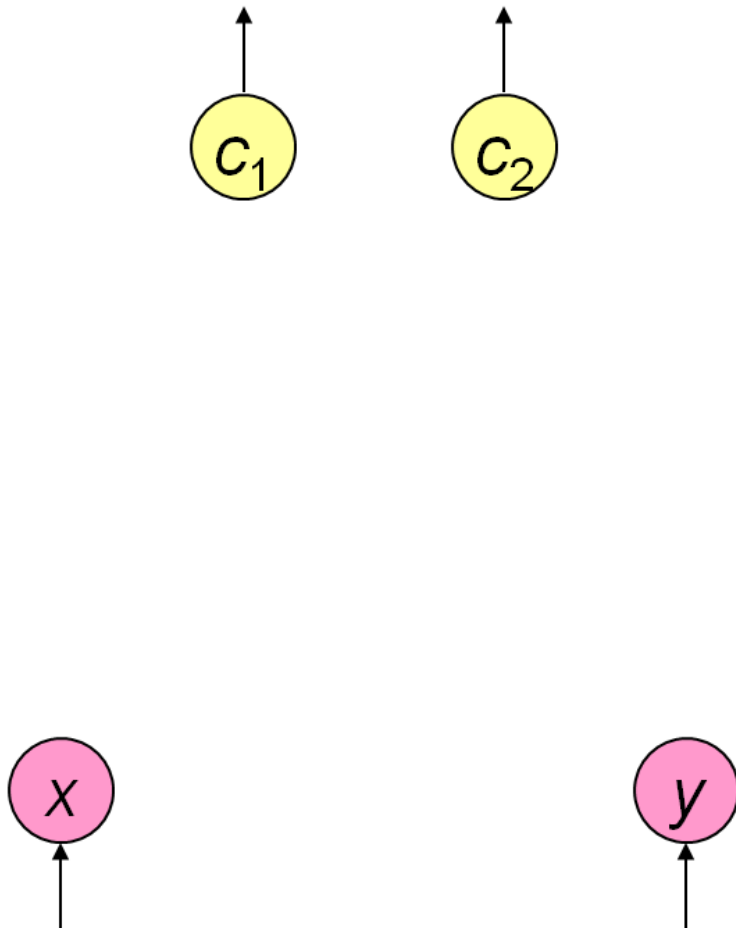
R_2 : if x is *large* and y is *large*, then class is c_2

- Verbindungen $x \rightarrow R_1$ und $x \rightarrow R_2$ sind verbu
- Fuzzymenge *large* teilt Gewicht auf
- d.h. *large* hat immer dieselbe Bedeutung in beiden Regeln



1. Trainieren: Initialisierung

Spezifiziere anfängliche Fuzzy-Partitionen für alle Eingabevariablen



1. Trainieren: Regelbasis

```
for each pattern  $p$  {  
    find antecedent  $A$  s.t.  $A(p)$  is maximal  
    if  $A \notin L$  {  
        add  $A$  to  $L$   
    }  
}  
for each antecedent  $A \in L$  {  
    find best consequent  $C$  for  $A$   
    create rule base candidate  $R = (A, C)$   
    determine performance of  $R$   
    add  $R$  to  $B$   
}  
return one rule base from  $B$ 
```

Fuzzy-Regel-Basen können auch aus Vorwissen, Fuzzy-Cluster-Analyse, Fuzzy-Entscheidungsbäumen, Evolutionären Algorithmen etc. gewonnen werden

Auswahl einer Regelbasis

Effizienz einer Regel:

$$P_r = \frac{1}{N} \sum_{i=1}^N N(-1)^c R_r(\mathbf{x}_i)$$

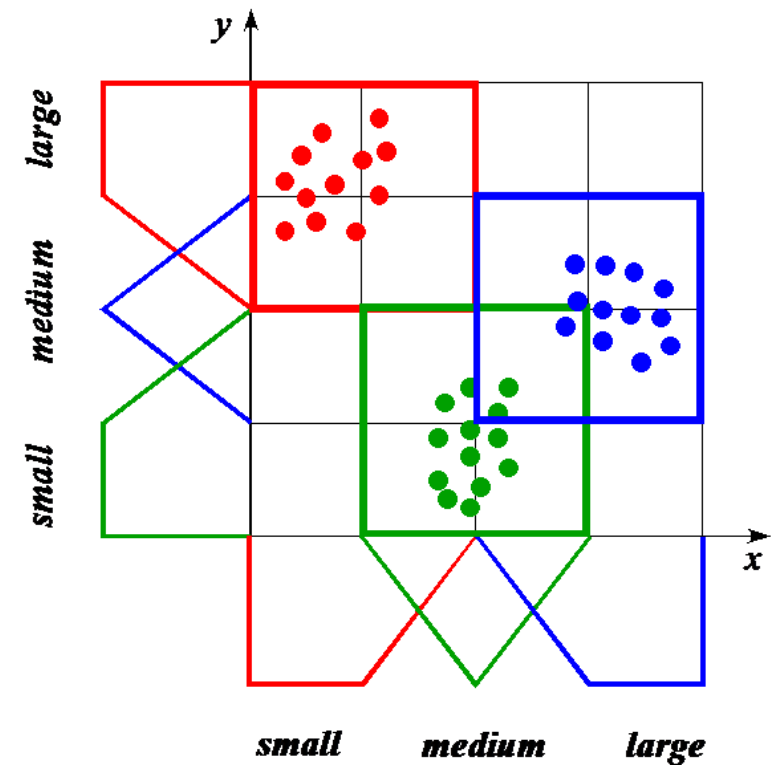
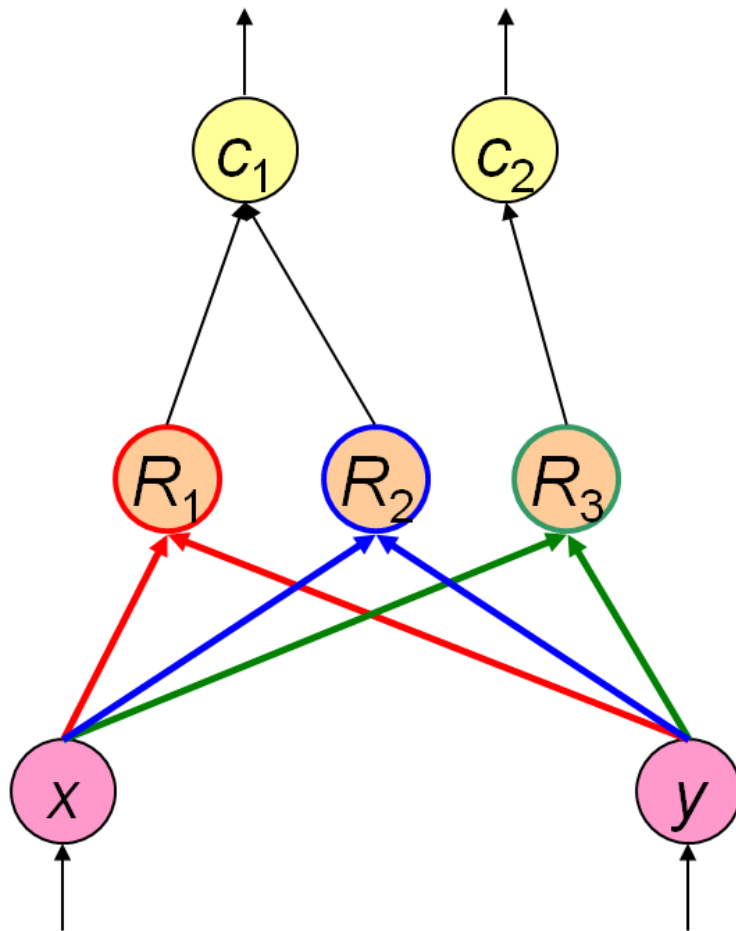
mit

$$c = \begin{cases} 0 & \text{falls } \text{class}(x_i) = \text{con}(R_r), \\ 1 & \text{sonst} \end{cases}$$

- sortiere Regeln nach Effizienz
- wähle entweder die besten r Regeln oder die besten r/m Regeln pro Klasse aus
- r ist entweder gegeben oder wird automatisch so bestimmt, dass alle Muster abgedeckt werden

Induktion der Regelbasis

NEFCLASS benutzt eine angepasste Wang-Mendel-Prozedur



Berechnung des Fehlersignals

Fuzzy-Fehler (j -te Ausgabe):

$$E_j = \text{sgn}(d)(1 - \gamma(d))$$

mit $d = t_j - o_j$ und

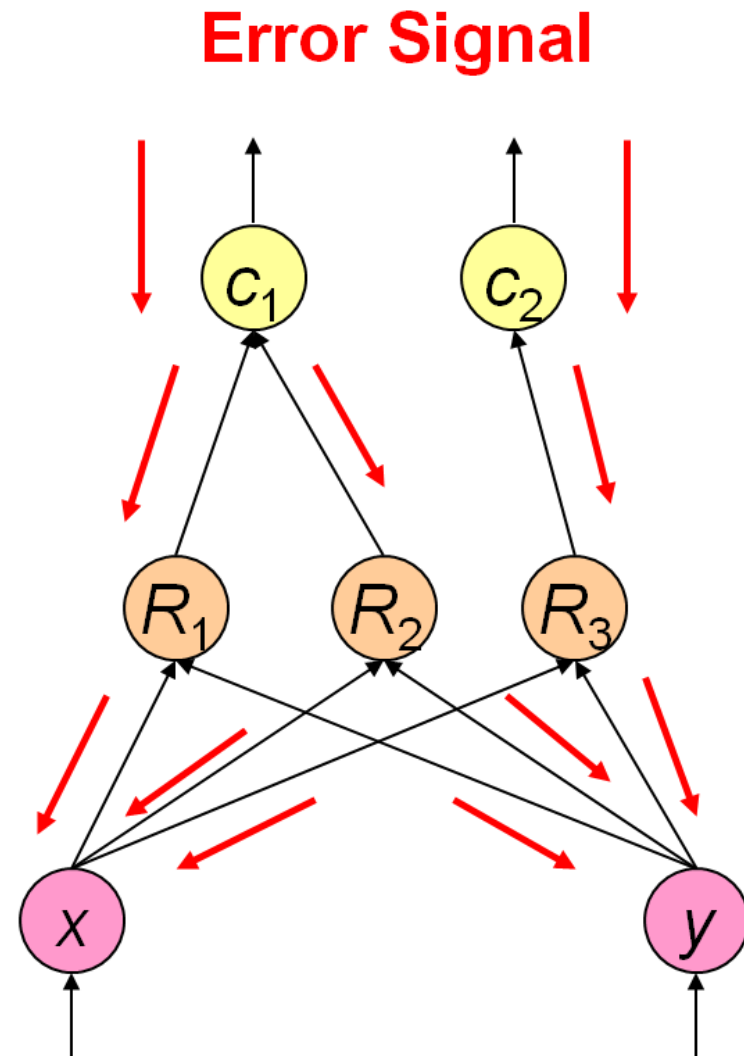
$$\gamma : \mathbb{R} \rightarrow [0, 1], \gamma(d) = \exp - \left(\frac{a \cdot d}{d_{\max}} \right)^2$$

(t : korrekte Ausgabe, o : aktuelle Ausgabe)

Regel-Fehler:

$$E_r = (\tau_r(1 - \tau_r) + \varepsilon) E_{\text{con}(R_r)}$$

mit $0 < \varepsilon \ll 1$



3. Trainingsschritt: Fuzzy-Mengen

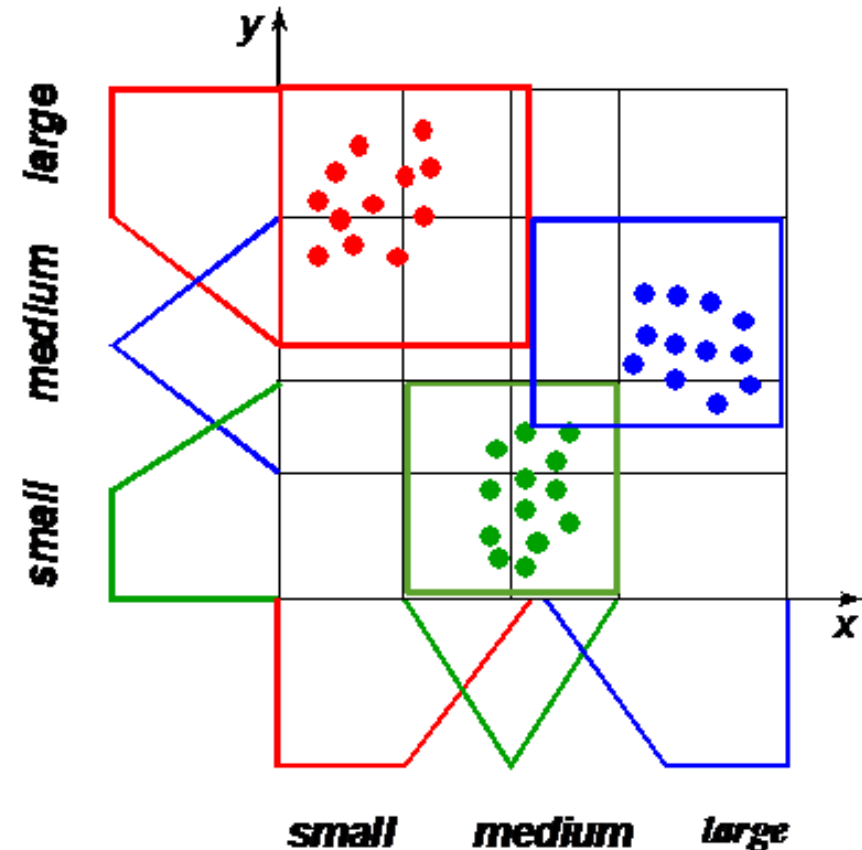
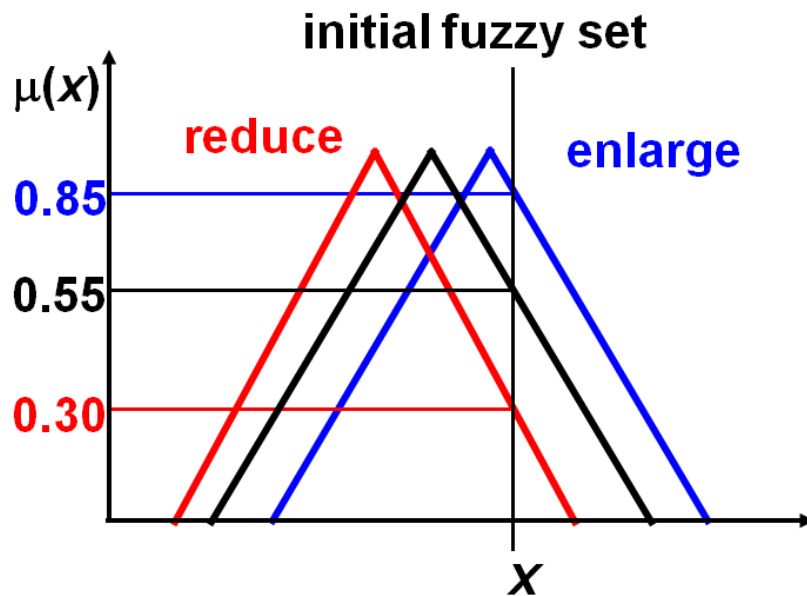
z.B. dreieckige Zugehörigkeitsfunktion

$$\mu_{a,b,c} : \mathbb{R} \rightarrow [0, 1], \quad \mu_{a,b,c}(x) = \begin{cases} \frac{x-a}{b-a} & \text{if } x \in [a, b), \\ \frac{c-x}{c-b} & \text{falls } x \in [b, c], \\ 0 & \text{sonst} \end{cases}$$

Parameteranpassungen für eine Antezedens-Fuzzymenge:

$$f = \begin{cases} \sigma \mu(x) & \text{falls } E < 0, \\ \sigma(1 - \mu(x)) & \text{sonst} \end{cases}$$
$$\Delta b = f \cdot E \cdot (c - a) \operatorname{sgn}(x - b)$$
$$\Delta a = -f \cdot E \cdot (b - a) + \Delta b$$
$$\Delta c = f \cdot E \cdot (c - b) + \Delta b$$

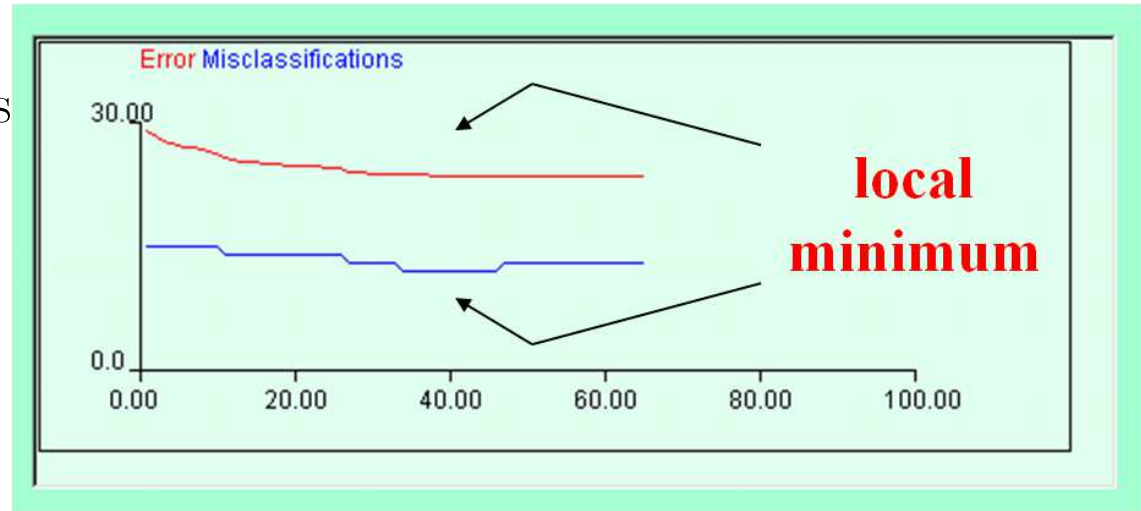
Trainieren von Fuzzy-Mengen



Heuristik: die Fuzzy-Menge wird **von x** weg (**auf x zu**) bewegt und ihr *support* wird **reduziert** (**vergrößert**) um den Zugehörigkeitsgrad von x zu **reduzieren** (**erhöhen**)

Trainieren von Fuzzy-Mengen

```
do {  
  for each pattern {  
    accumulate parameter updates  
    accumulate error  
  }  
  modify parameters  
} while change in error
```

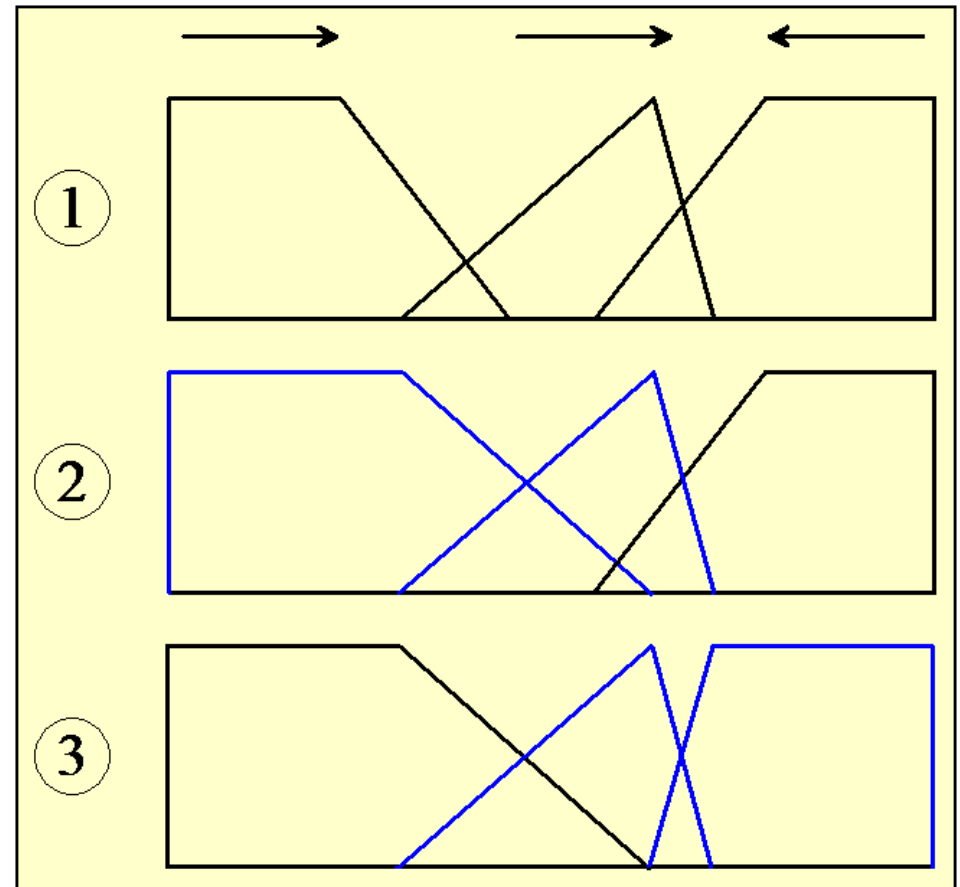


Varianten:

- Adaptive Lernrate
- Online-/Batch-Lernen
- Optimistisches Lernen (n Schritte in die Zukunft blickend)

Einschränkungen beim Trainieren von Fuzzy-Mengen

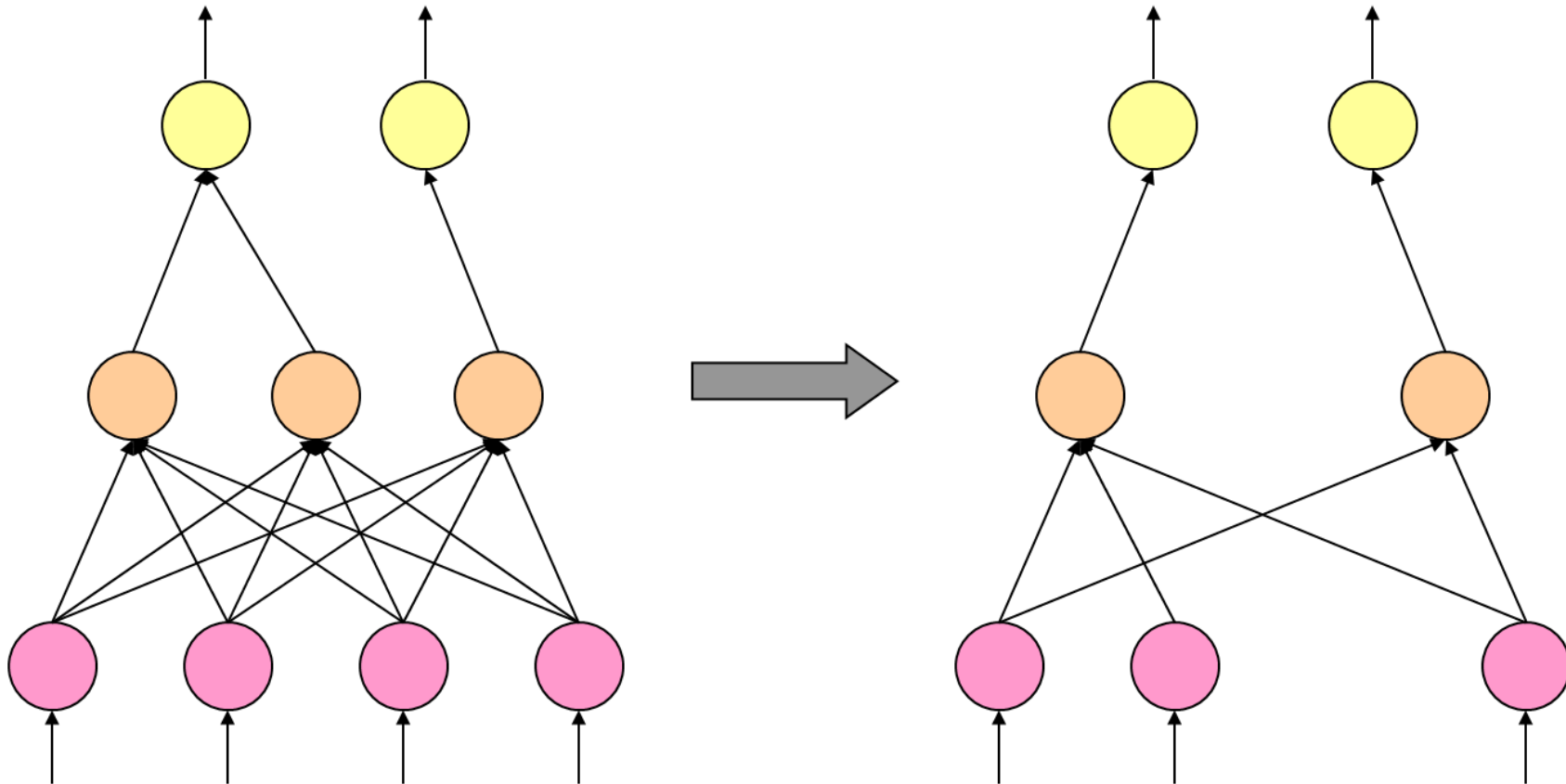
- gültige Parameterwerte
- nicht-leere Schnitte benachbarter Fuzzy-Mengen
- Beibehalten relativer Positionen
- Erhalt der Symmetrie
- Komplette Abdeckung (Zugehörigkeitsgrade für jedes Element summieren sich zu 1)



Correcting a partition after modifying the parameters

4. Trainingsschritt: Stutzen

Ziel: Entferne Variablen, Regeln und Fuzzy-Mengen, um die Interpretierbarkeit und Generalisierungsfähigkeit zu verbessern



Stutzen: Methoden

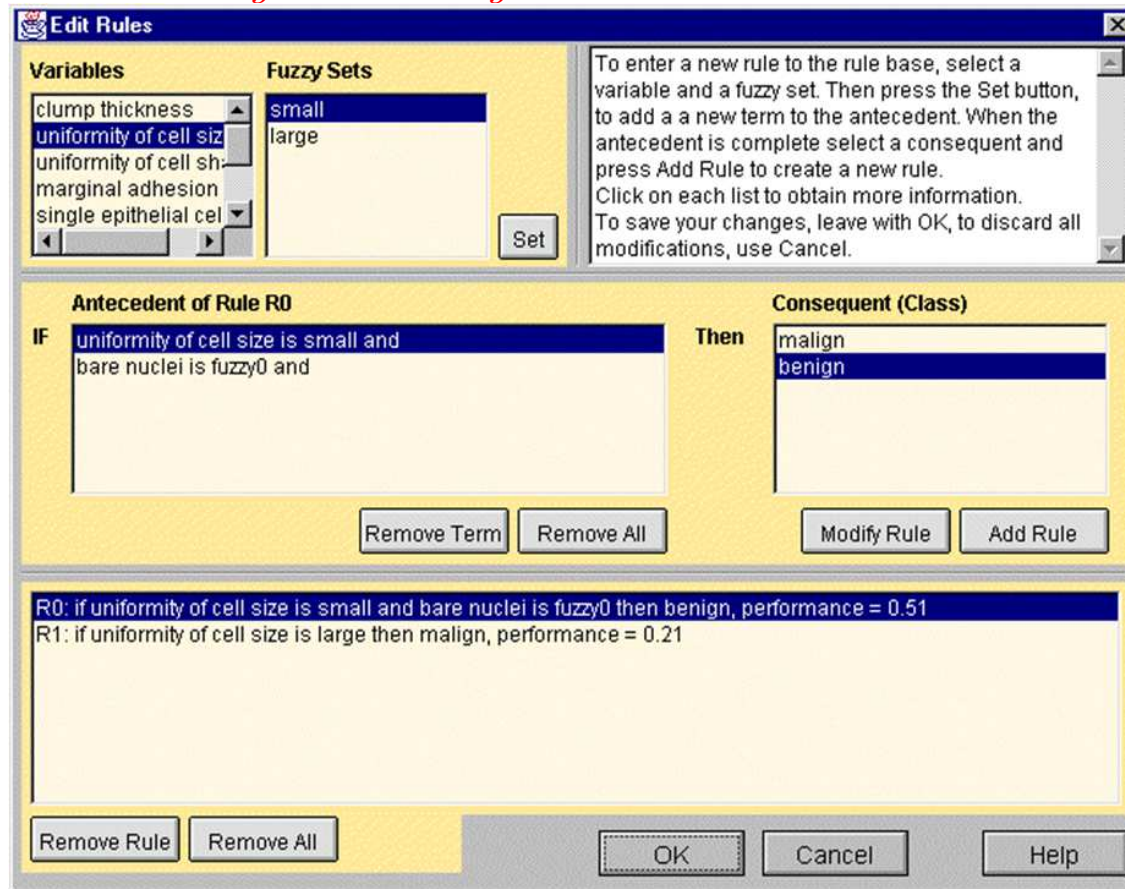
```
do {  
  select pruning method  
  do {  
    execute pruning step  
    train fuzzy sets  
    if no improvement {  
      undo step  
    }  
  } while there is improvement  
} while there is further method
```

1. Entferne Variablen (Korrelationen, Information Gain, etc.)
2. Entferne Regeln (Effizienz einer Regel)
3. Entferne Terme (Erfüllungsgrad einer Regel)
4. Entferne Fuzzy-Mengen

WBC- Ergebnisse: Fuzzy-Regeln

R_1 : if uniformity of cell size is *small* and bare nuclei is fuzzy0 then *benign*

R_2 : if uniformity of cell size is *large* then *malignant*



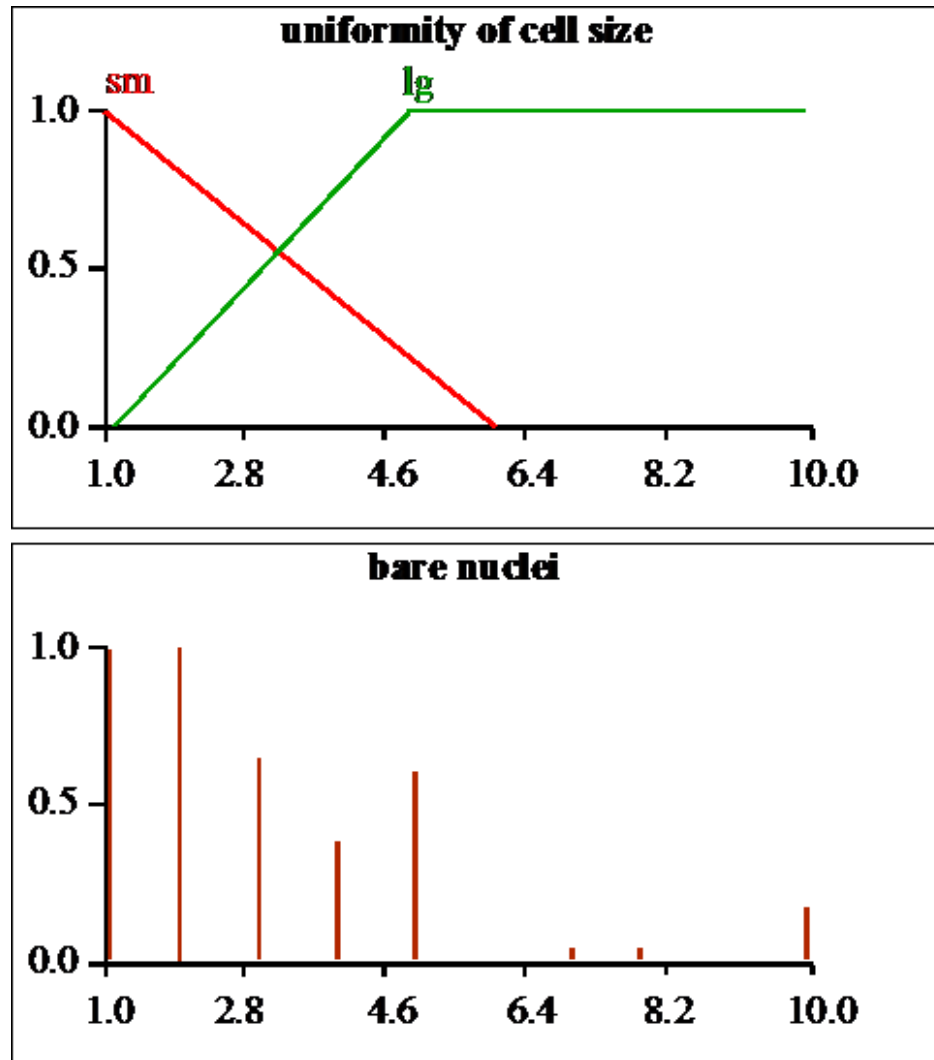
WBC-Ergebnisse: Klassifikation

	vorhergesagte Klasse		
	bösartig	gutartig	Σ
bösartig	228 (32.62%)	13 (1.86%)	241 (34.99%)
gutartig	15 (2.15%)	443 (63.38%)	458 (65.01%)
Σ	243 (34.76)	456 (65.24)	699 (100.00%)

Geschätzte Vorhersageleistung auf unbekanntem Daten (Kreuzvalidierung):

NEFCLASS-J:	95.42%	NEFCLASS-J (numerisch):	94.14%
Discriminant Analysis:	96.05%	Multilayer Perceptron:	94.82%
C 4.5:	95.10%	C 4.5 Rules:	95.40%

WBC-Ergebnisse: Fuzzy-Mengen



NEFCLASS-J

The screenshot displays the NEFCLASS-J software interface with several windows open:

- Fuzzy Set Learning:** A graph showing 'Error Misclassifications' over time. The error starts at 49.00 and decreases to approximately 10.00 over 180.00 iterations.
- NEFCLASS:** The main application window showing project status (ready), classifier (5 rules, trained), and training data (Iris_new.dat).
- Fuzzy Sets:** A graph showing three fuzzy sets: 'sm' (small), 'md' (medium), and 'lg' (large) for a variable ranging from 1.0 to 7.0.
- List of Variables:** A list of variables: sepal length, sepal width, petal length, and petal width.
- Rule Learning:** A text window displaying the results of rule learning, including the number of rules found (20), the selection of the best 5 rules, and performance metrics (94% coverage, 15 misclassifications).
- Edit Rules:** A window for editing rules, showing a list of variables and fuzzy sets, and a 'Set' button.
- About:** A window providing information about NEFCLASS-J, including the version (1.0), programming and GUI design by Ulrike Nauck, and the model and learning algorithm by Dr. Detlef Nauck.

The 'Edit Rules' window shows the following rule structure:

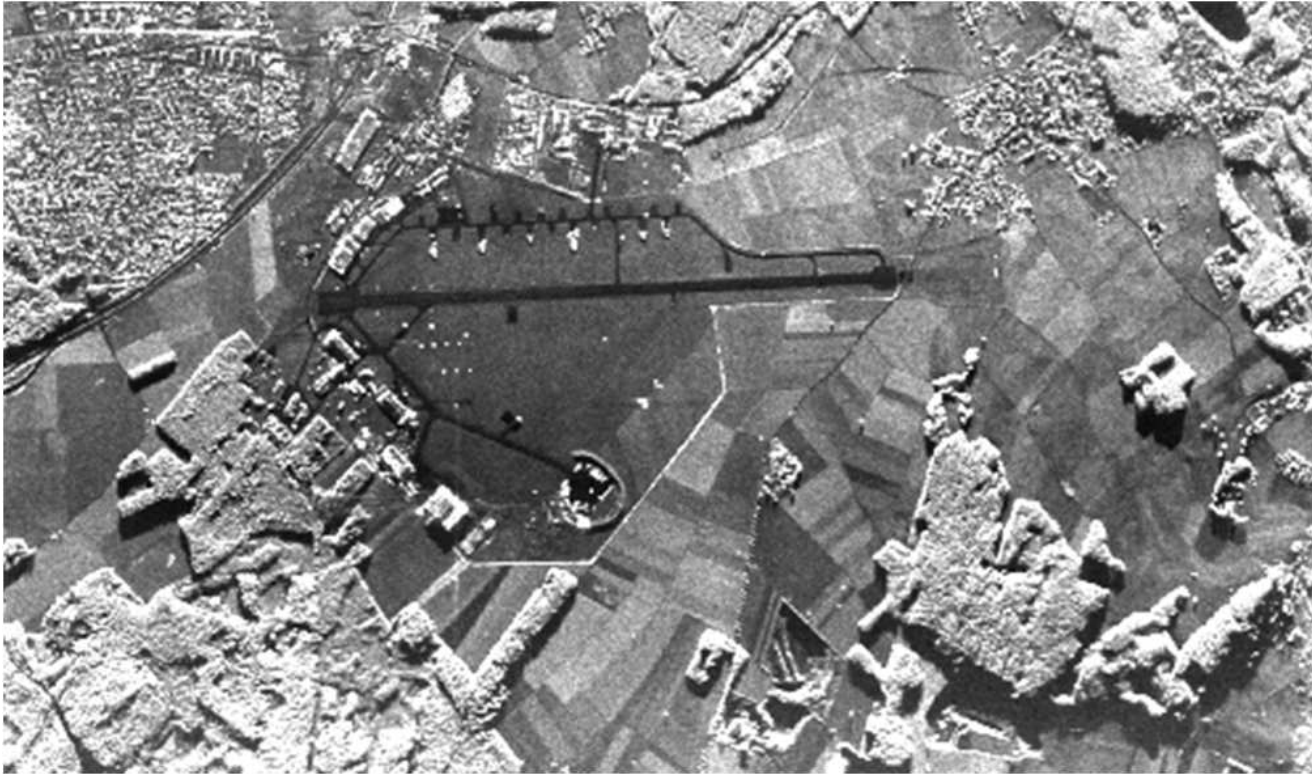
Antecedent of Rule R0	Consequent (Class)
IF sepal length is small and sepal width is medium and petal length is small and petal width is small	Then Iris Setosa Iris Versicolor Iris Virginica

Buttons: Remove Term, Remove All, Modify Rule, Add Rule

Legend for Rule R0:

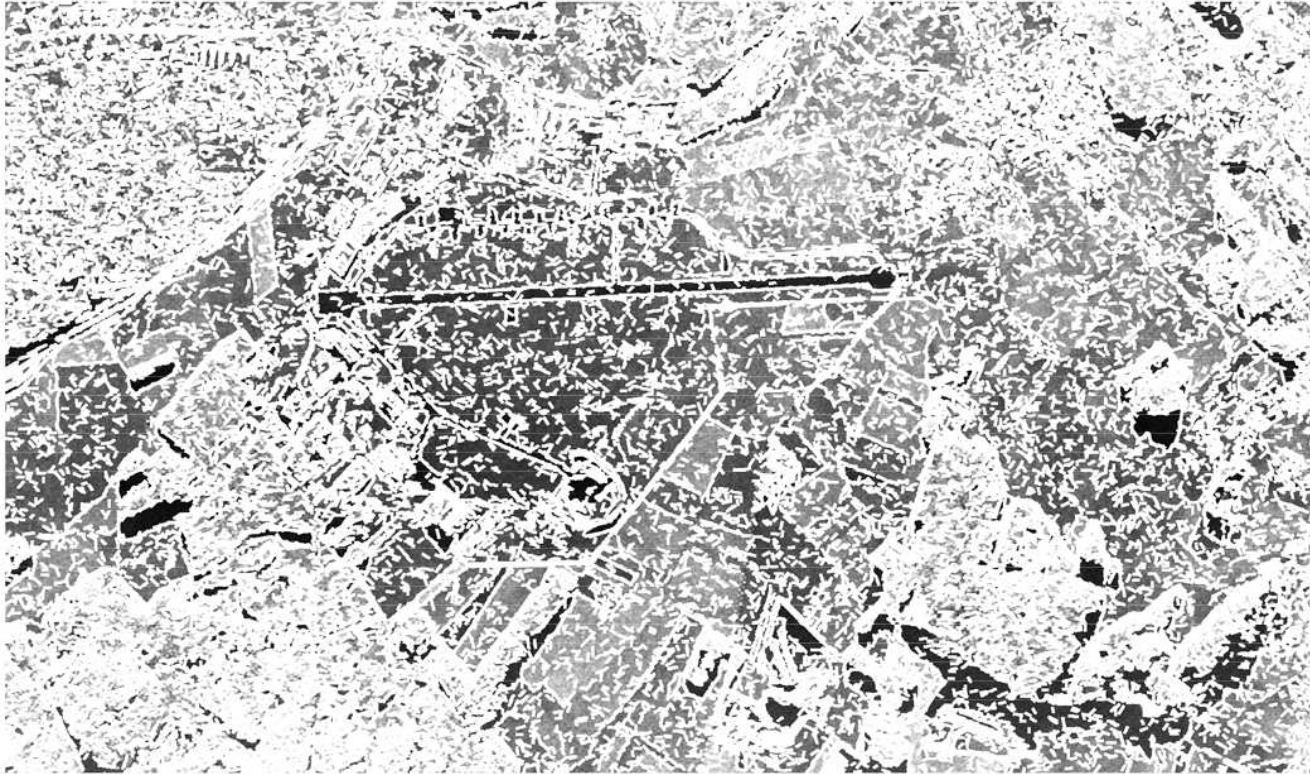
- R0: if sepal length is small and sepal width is medium and petal length is small and petal width is small then Iris
- R9: if sepal length is medium and sepal width is small and petal length is medium and petal width is medium then Iris
- R16: if sepal length is medium and sepal width is small and petal length is large and petal width is large then Iris
- R17: if sepal length is large and sepal width is small and petal length is large and petal width is large then Iris
- R18: if sepal length is large and sepal width is medium and petal length is large and petal width is large then Iris

Fallstudie: Linienerkennung



- Extraktion von Kantensegmenten (Burns' operator)
- weitere Schritte:
Kanten \rightarrow Linien \rightarrow lange Linien \rightarrow parallele Linien \rightarrow Landebahnen

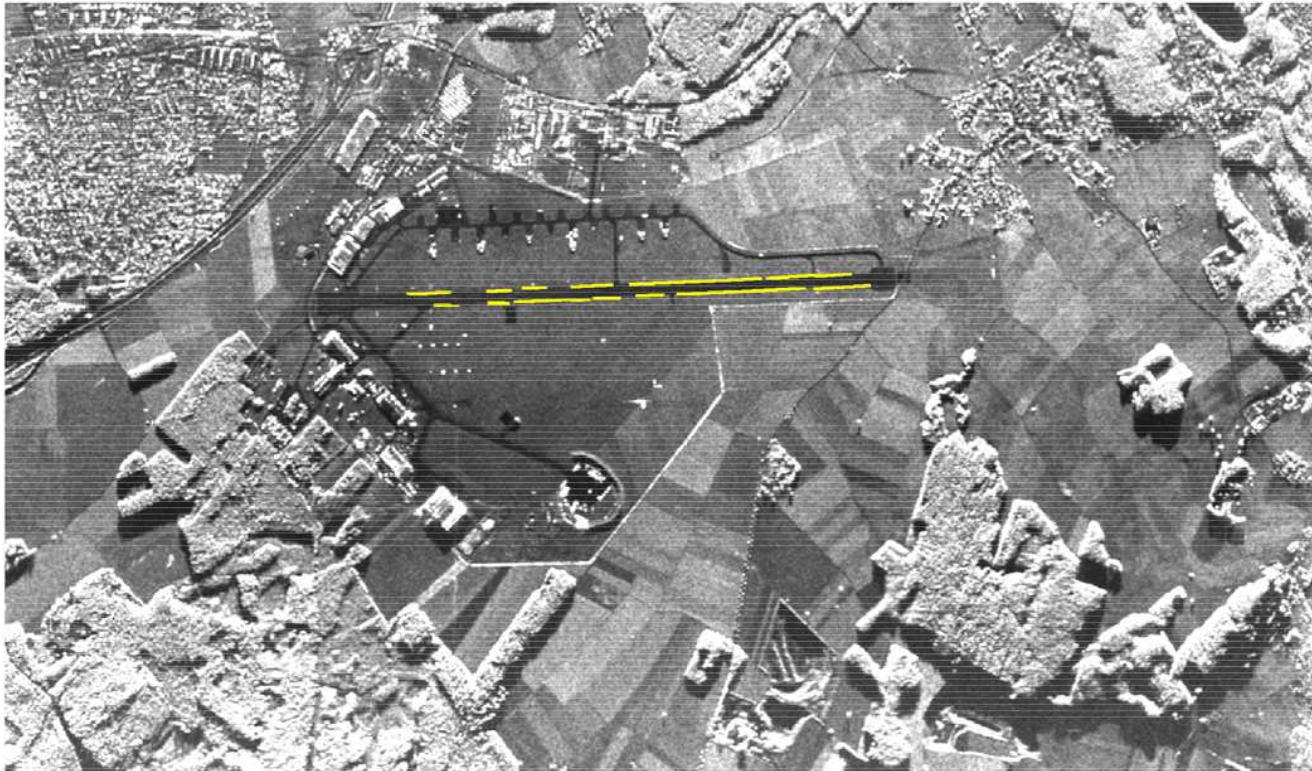
Fallstudie: Linienerkennung



Probleme:

- sehr viele Linien wegen verzerrter Bilder
- Lange Ausführungszeiten der Erstellungs-Schritte (bis Landebahnen)

Fallstudie: Linienerkennung

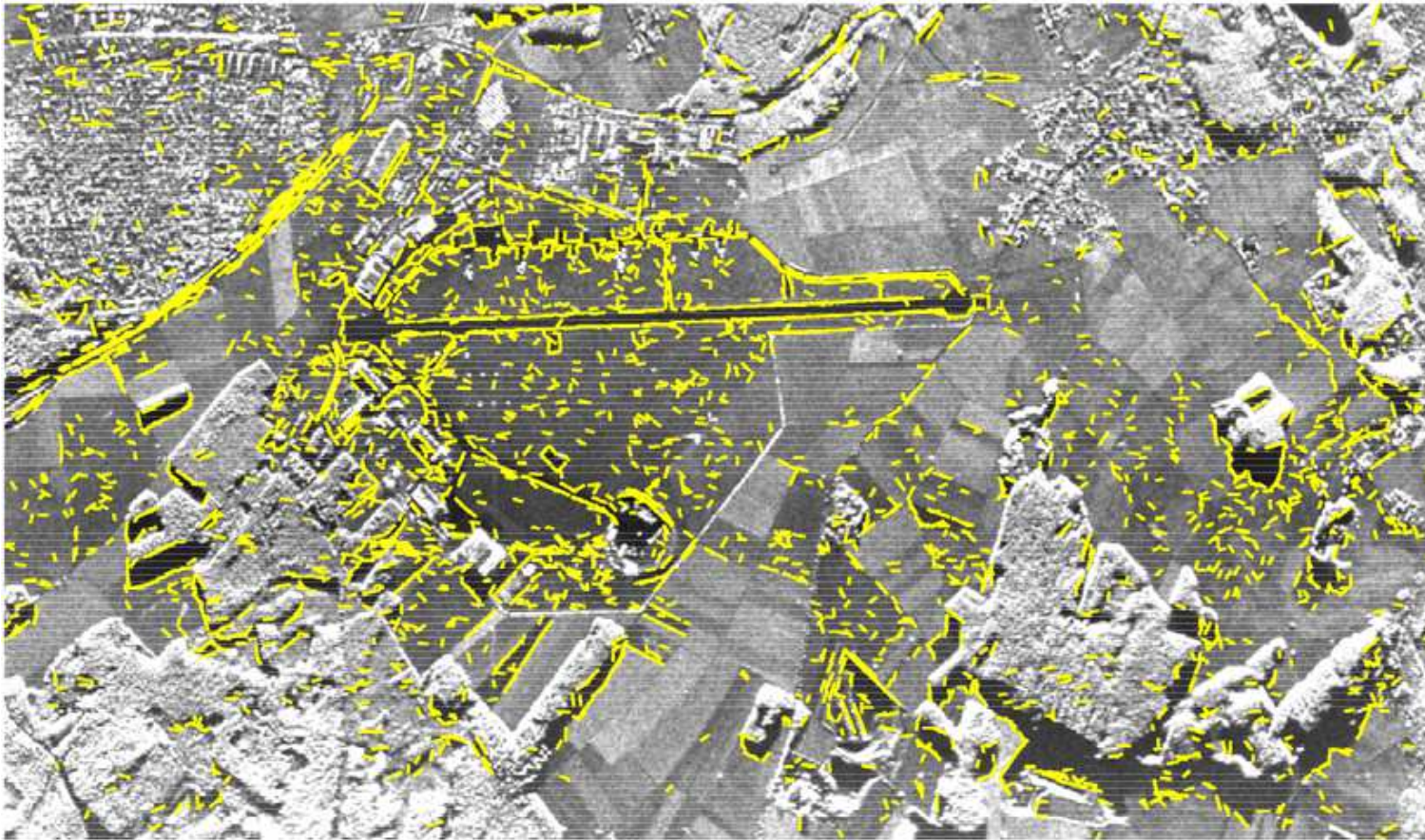


nur wenige Linien für Landebahnen genutzt

Ansatz:

- extrahiere Textureigenschaften der Linien
- identifiziere und verwirf überflüssige Linien

Fallstudie: Linienerkennung



mehrere Klassifikatoren: minimum distance, k-NN, C 4.5, NEFCLASS

Probleme: Klassen überlappen und sind extrem unbalanciert

obiges Ergebnis durch modifiziertes NEFCLASS erhalten:

- alle Landebahnlinien gefunden, Reduktion auf 8.7% der Kantensegmente

Zusammenfassung

- Neuro-Fuzzy-Systeme können nützlich zur Wissensentdeckung sein
- Interpretierbarkeit ermöglicht die Plausibilitätskontrolle und erhöht die Akzeptanz
- NFS nutzen Toleranzen aus, um zu beinahe optimalen Lösungen zu kommen
- NFS-Lernalgorithmen müssen mit Einschränkungen umgehen können, um die Semantik des ursprünglichen Modells nicht zu verletzen
- keine automatische Modellerstellung \Rightarrow Benutzer muß mit dem Werkzeug **umgehen**
- Einfache Lerntechniken unterstützen die explorative Datenanalyse.

Heutiges Verfahren

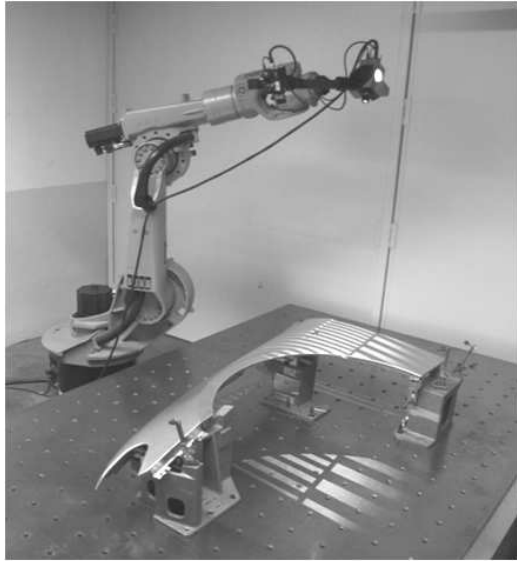
- Oberflächenkontrolle: manuell durchgeführt
- erfahrener Arbeiter bearbeitet Oberfläche mit Schleifstein
- Experten klassifizieren Abweichungen durch sprachliche Beschreibungen
- umständlich, subjektiv, fehleranfällig, zeitaufwendig



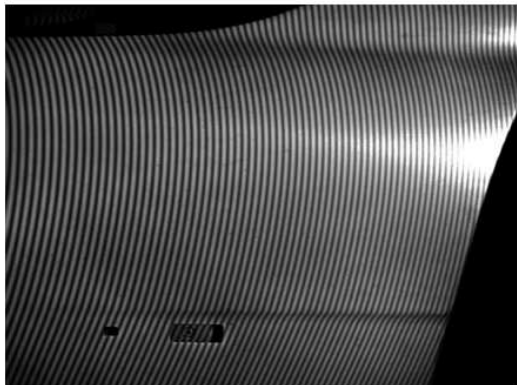
vorgeschlagener Ansatz:

- Digitalisierung der Oberfläche mit optischen Mess-Systemen
- Charakterisierung der Formabweichungen durch mathematische Eigenschaften (nahe der subjektiven Merkmale)

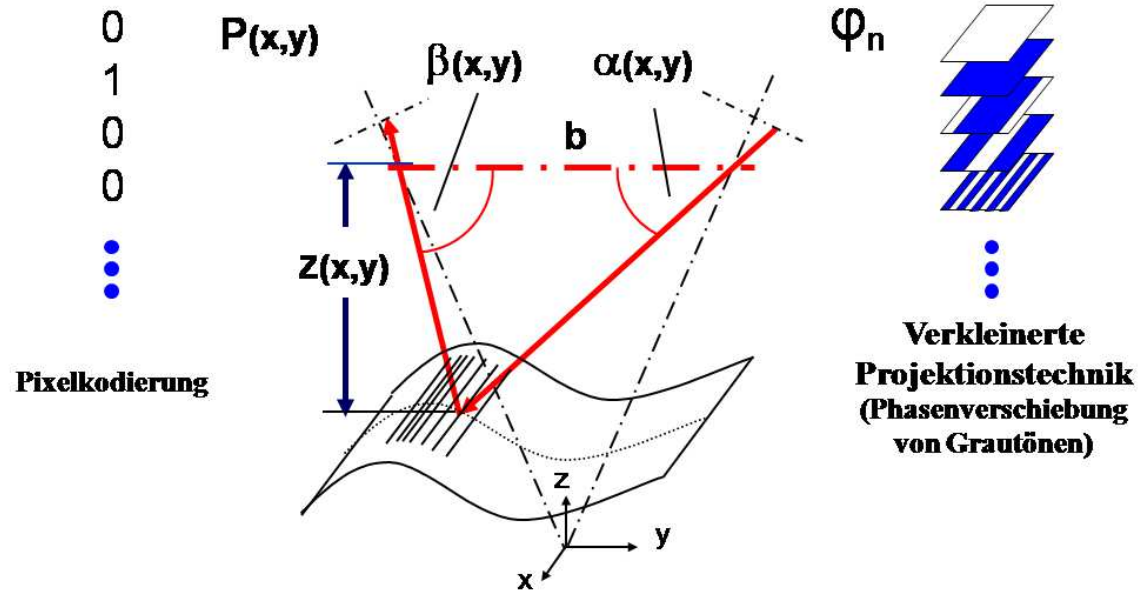
Topometrisches 3D Mess-System



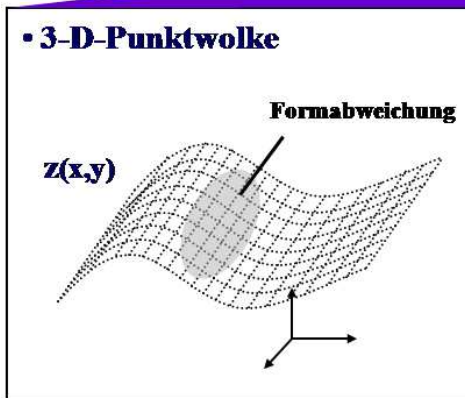
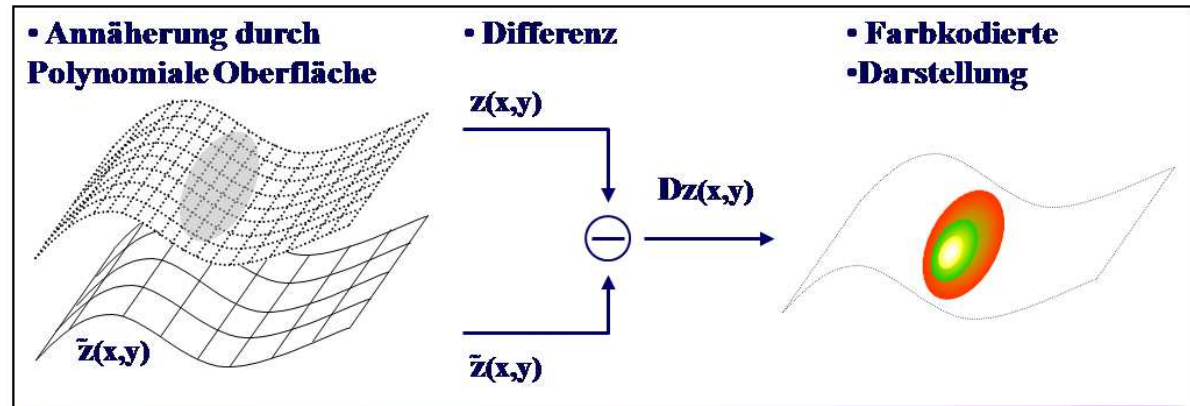
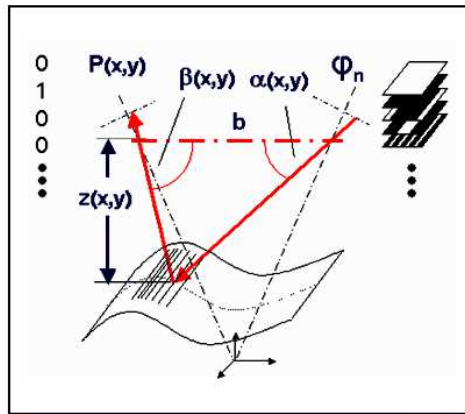
breuckmann 



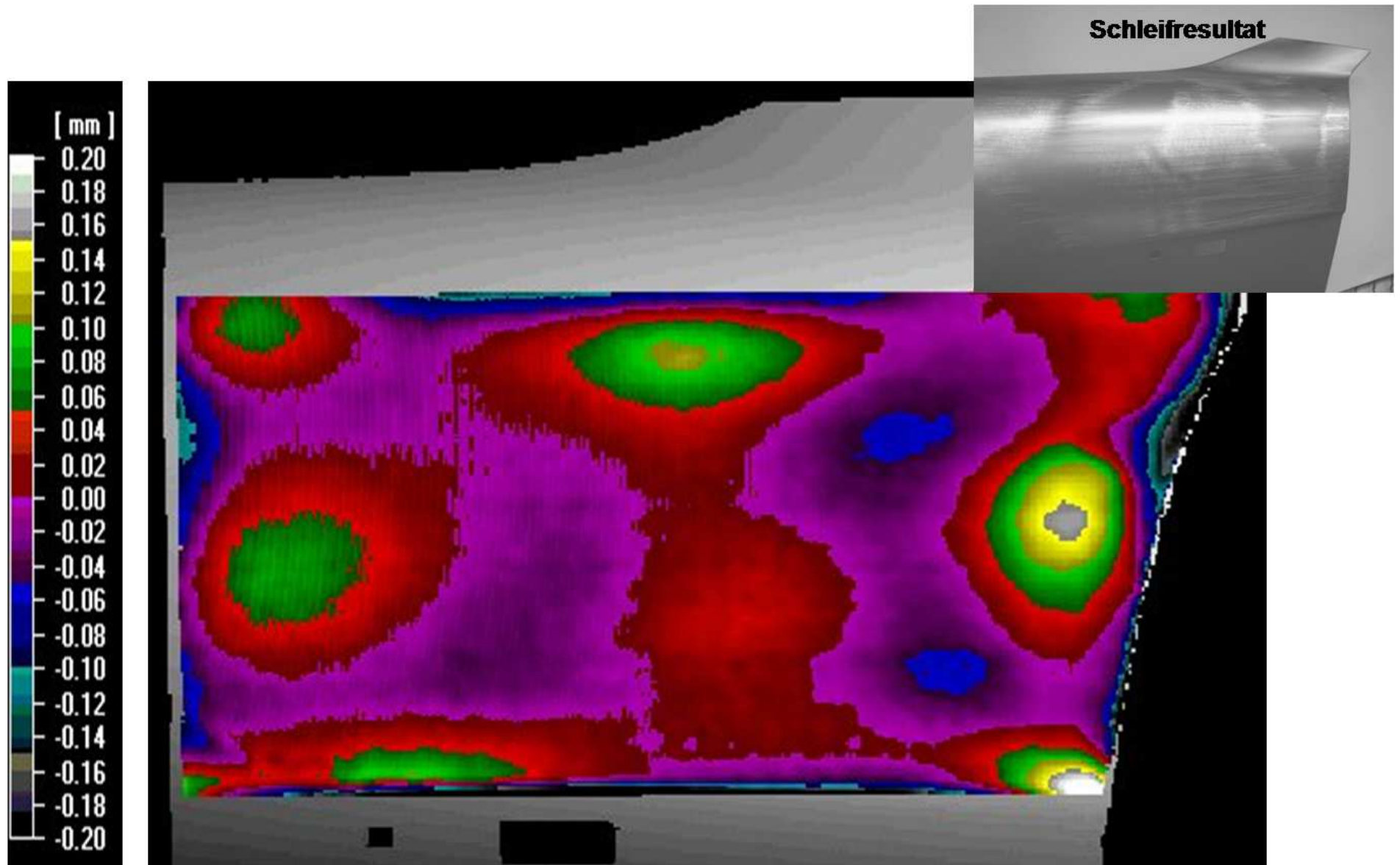
Triangulation und Gitterprojektion



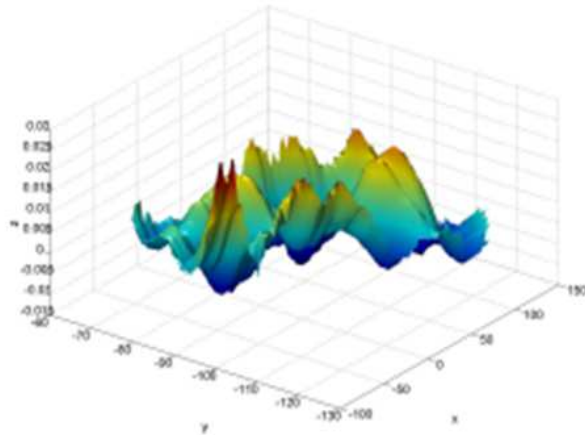
- hohe Punktdichte
- schnelle Datenansammlung
- genaue Messung
- kontakt- und harmlos



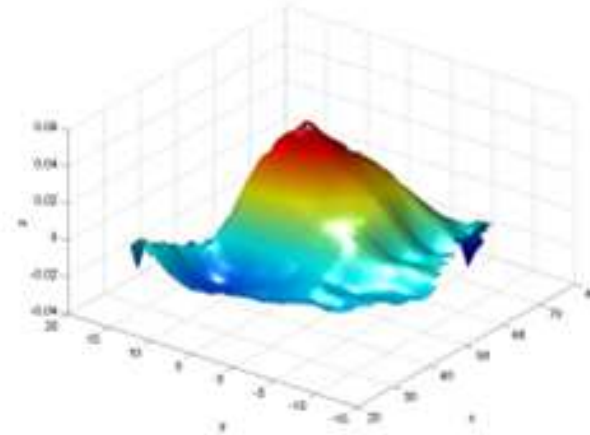
Farbkodierte Darstellung



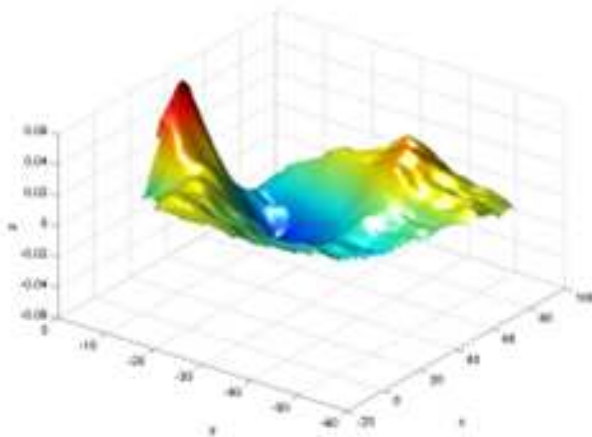
3D-Darstellung lokaler Oberflächendefekte



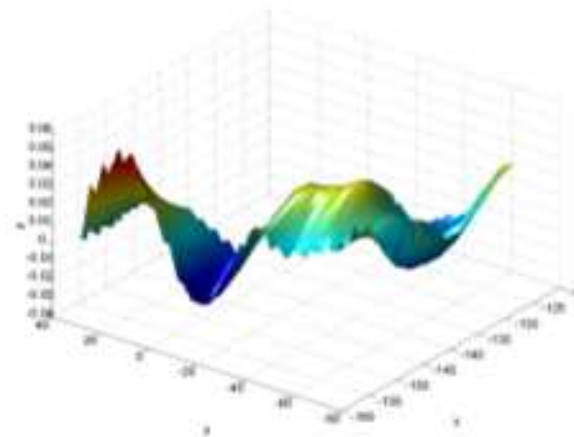
unebene Oberfläche
mehrere Einfallstellen in Serie/benachbart



Walzenmarkierung
lokale Glättung der Oberfläche



Einfallstelle
leichte flach basierte Senke einwärts



Wellplatte
mehrere schwerere Faltungen in Serie

Charakteristik der Daten

- 9 Meisterstücke mit insgesamt 99 Defekten analysiert
- für jeden Defekt, 42 Merkmale berechnet
- Typen sind eher unbalanciert
- seltene Klassen verworfen
- einige extrem korrelierte Merkmale verworfen (31 übrig)
- Rangfolge der 31 Merkmale nach Wichtigkeit
- geschichtete 4-fache Kreuzvalidierung fürs Experiment

Regelbasis für NEFCLASS:

- Rule base
 - Rule 1: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 1 AND max_extrema IS fun 1) THEN type IS press_mark
 - Rule 2: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND max_extrema IS fun 2) THEN type IS sink_mark
 - Rule 3: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
 - Rule 4: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
 - Rule 5: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND min_extrema IS fun 2) THEN type IS press_mark
 - Rule 6: IF (max_distance_to_cog IS fun 3 AND all_extrema IS fun 2 AND max_extrema IS fun 3) THEN type IS uneven_surface
 - Rule 7: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 3) THEN type IS uneven_surface

Klassifikationsgenauigkeit:

	NBC	DTree	NN	NEFCLASS	DC
Trainingsmenge	89.0%	94.7%	90%	81.6%	46.8%
Testmenge	75.6%	75.6%	85.5%	79.9%	46.8%