

# Gliederung

---

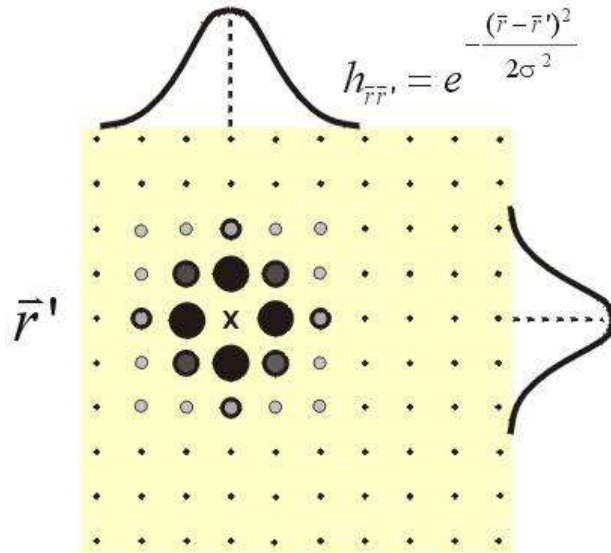
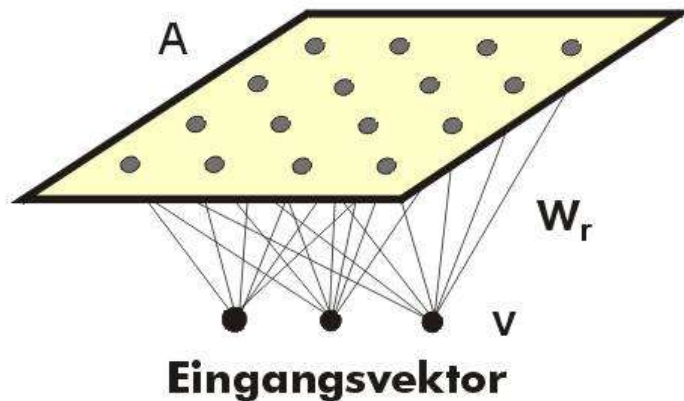
1. Einleitung
  1. Motivation
  2. Was ist *Data Mining*
  3. Was versteht man unter *Big Data*
  4. Erstes Anwendungsbeispiel von *Data Mining*
2. Notwendige Verarbeitungsschritte des *Data Mining*
  1. Einführung
  2. Grafische Datenanalyse
  3. Datenvorverarbeitung
  4. Datenaggregation und Merkmalsberechnung
  5. Behandlung der Datenstichprobe
3. Einige grundlegende Methoden
  1. Clusteranalyse
  2. Künstliche neuronale Netze / Deep Learning Verfahren
4. Einfache statistische Abhängigkeitsanalyse
5. Komplexere Methoden der Abhängigkeitsanalyse
  1. Diskriminanzanalyse
  2. Entscheidungsbaumverfahren / Random Forests
  - 3. Self Organising Map**
  4. Genetische Algorithmen
6. Methoden der datenbasierten Modellbildung am Beispiel der Klassifikation
  1. Grundlagen der Klassifikation
  2. Statistische Methoden
  3. Neuronale Methoden
7. Anwendungsbeispiele

# Neuronale Methoden zur Ursachenanalyse

---

- \* Perceptron
- \* Adaline
- \* Madaline
- \* Multi-Layer Perceptron
- \* Time-Delay Networks
- \* Elman- / Jordan Nets
- \* Associative memory
- \* Hopfield Network
- \* Haken Network
- \* Hamming Network
- \* Bidirectional Associative Memory
- \* Brain-State-in-a-Box
- \* Adaptive Resonance Theory 1,2,3
- \* ARTMAP
- \* Fuzzy ART
- \* Kohonen Self-Organizing Map SOM
- \* Counterpropagation
- \* Radial Basis Function Network
- \* Hyper Basis Function Network
- \* Boltzmann Machine
- \* Probabilistic Networks
- \* Neocognitron
- \* Learning Vectorquantisation
- \* Wavelet-Network
- \* Filter-Network
- \* General Regression Networks

# Self Organising Map (SOM)

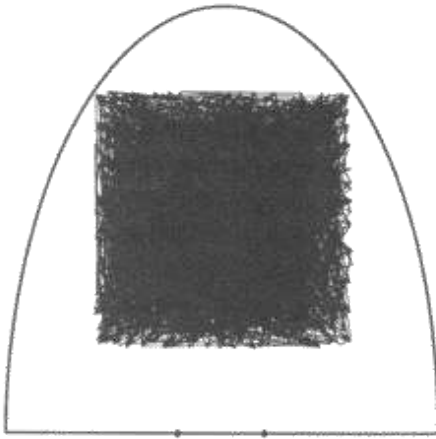


## Algorithmus:

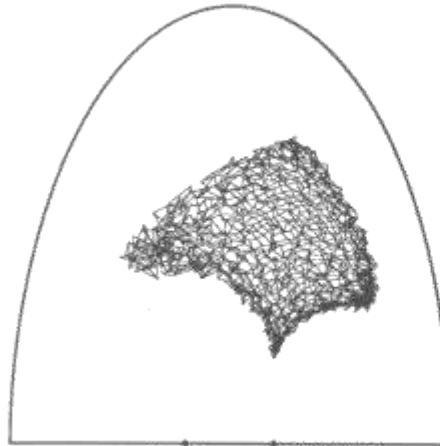
0. *Initialisierung*: Gewichte  $w_r$  und Parameter init.
1. *Stimuluswahl*: Wahl eines Eingangsvektors  $v$  aus der Datenstichprobe
2. *Response*: bestimme Erregungszentrum  $r'$  aus  $\|v - w_{r'}\| < \|v - w_r\|$  für alle  $r \in A$
3. *Adaption*: Lernschritt durch Veränderung der Gewichte gemäß  $w_r^{\text{neu}} = w_r^{\text{alt}} + \eta h_{rr'}(v - w_r^{\text{alt}})$
4. *Iteration*: fahre fort mit Schritt 1 bis Datenstichprobe abgearbeitet
5. *Dekrement*: Änderung der Lernrate und der Breite der Nachbarschaftsfunktion gemäß  $\eta = \eta_{\text{start}} \cdot 0.01^{0.8 \cdot k}$   $\sigma = \sigma_{\text{start}} \cdot 0.01^{0.8 \cdot k}$
6. *Stichprobeniteration*: fahre fort mit Schritt 1 bis Datenstichprobe N-mal abgearbeitet

# „selbstorganisierende Karte“

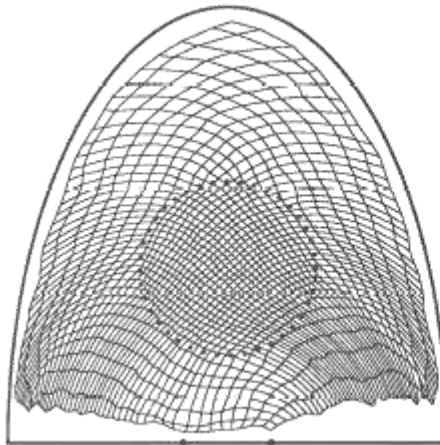
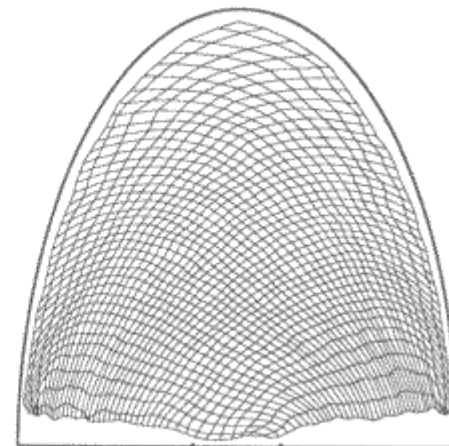
0 Lernschritte



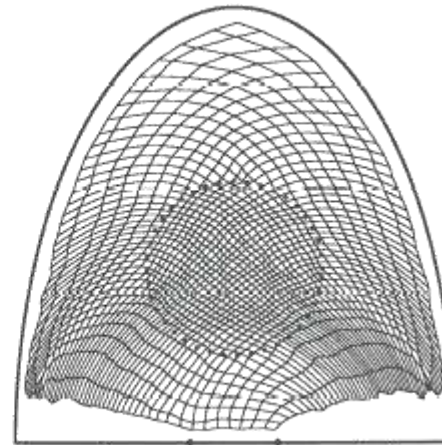
100 Lernschritte



40.000 Lernschritte



erhöhte Wahrscheinlichkeit



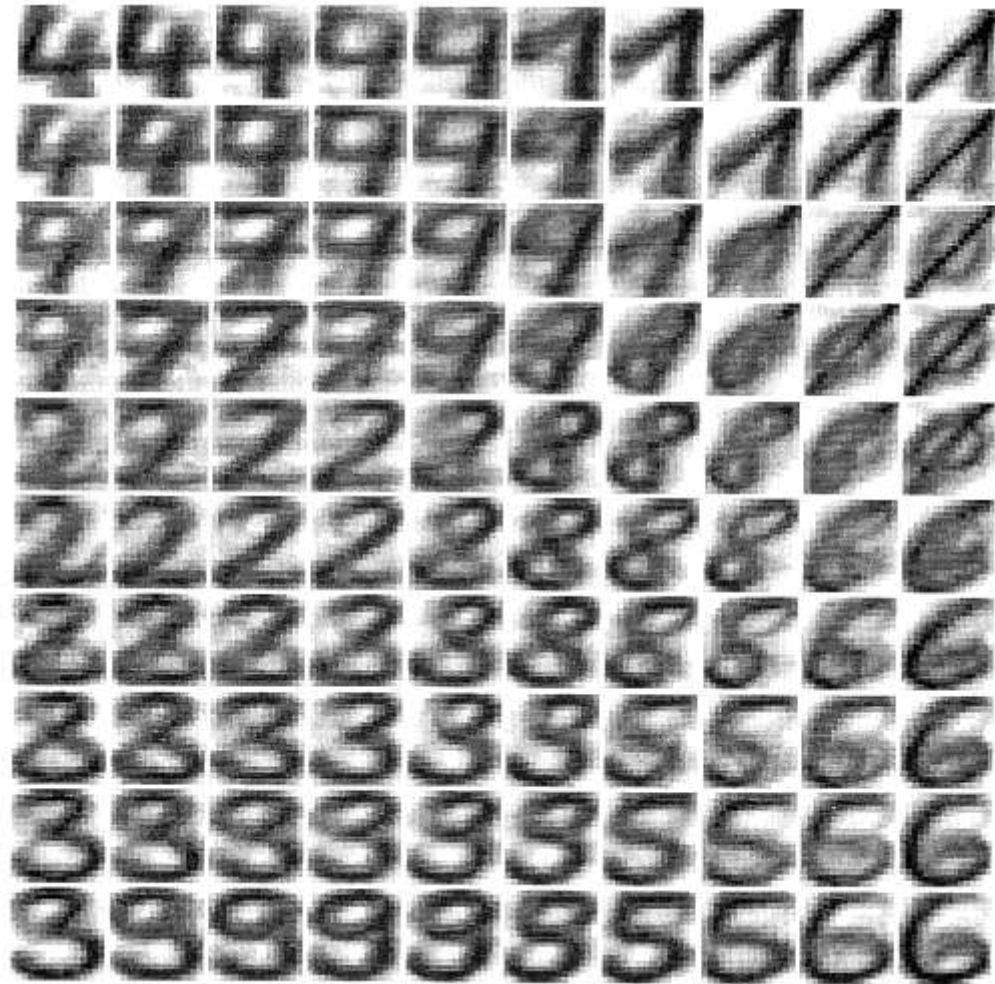
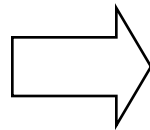
veränd. Lernschrittweite



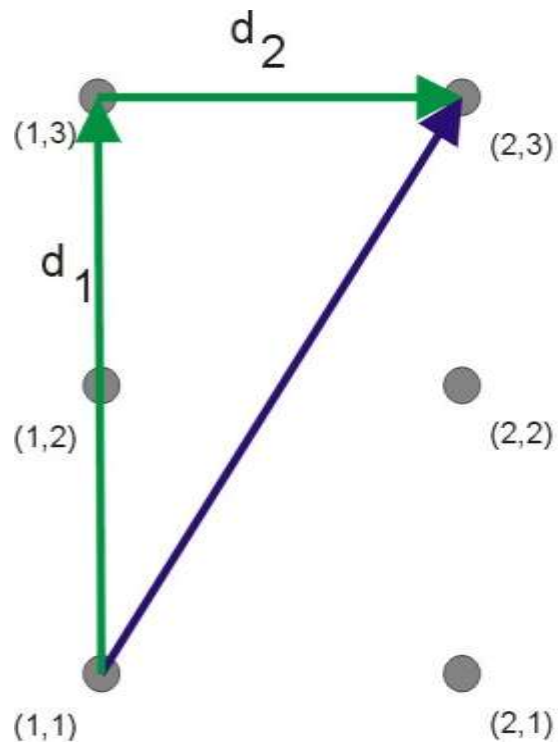
# Beispiel Schriftzeichenerkennung

100 Personen  
schreiben die  
Ziffern 0-9  
jeweils 20 mal

Datenstichprobe



# Distanzmaße



Vector distance:

$$D = \|\underline{x}_2 - \underline{x}_1\| = \sqrt{(\underline{x}_2 - \underline{x}_1)^2} = \sqrt{\left(\begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right)^2} = \sqrt{5}$$

Grid distance:

$$D = \max\left(\begin{pmatrix} 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) = \max\begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2$$

Manhattan distance:

$$L = d_1 + d_2 = 2 + 1 = 3$$

# Mathematische Eigenschaften der SOM

## Vektorquantisierung / Datenkompression

gegeben:  $\vec{v}(t)$ ,  $t = 1, 2, 3, \dots$

gesucht:  $W$ , "Kodebuch"

$$E(W) = \int \|\vec{v} - w_{s(\vec{v})}\|^2 P(\vec{v}) d\vec{v}$$

$$\bar{w}_r(t+1) = \bar{w}_{r(t)} - \frac{\varepsilon}{2} \frac{\partial E}{\partial \bar{w}_r}$$

$$= \bar{w}_r(t) + \varepsilon \int_{s(\vec{v})=r} (\vec{v} - \bar{w}_r(t)) P(\vec{v}) d\vec{v}$$

$$\bar{w}_{s(\vec{v})}(t^* + 1) = \bar{w}_{s(\vec{v})}(t^*) + \varepsilon (\vec{v} - \bar{w}_{s(\vec{v})}(t^*))$$

$$\text{SOM: } \bar{w}_r^{\text{neu}} = \bar{w}_r^{\text{alt}} + \eta \cdot h_{rr'} (\vec{v} - \bar{w}_r^{\text{alt}})$$

=>: \* SOM ist eine Verallgemeinerung eines Vektorquantisierungsverfahrens

\*  $\bar{w}_r$  entsprechen den Referenzvektoren

## Faktor- oder Varianzanalyse

gegeben:  $\vec{v}^{(1)}, \vec{v}^{(2)}, \vec{v}^{(3)}, \dots$  für  $\vec{v} = (v_1, v_2, v_3, \dots, v_L)^T$ ,  $v_j$  in der Regel korreliert

$v_i = f_i(r_1, r_2, \dots, r_D)$ ,  $i = 1 \dots L$ ,  $r_i$  ermöglichen sparsamere Beschreibung des Vorgangs

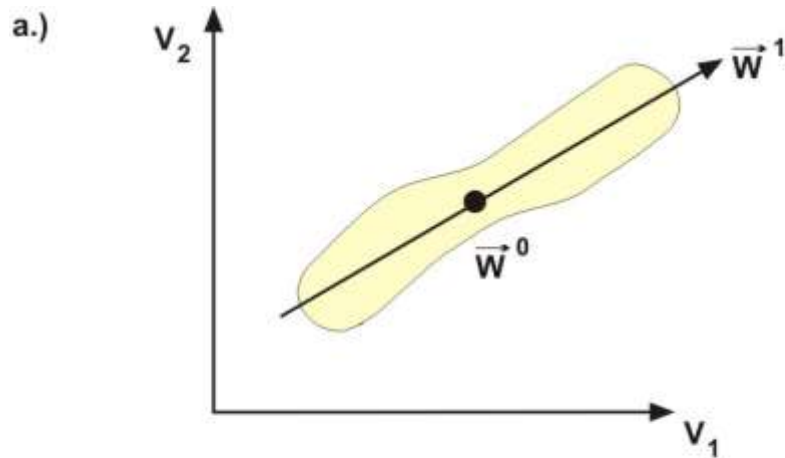
lin. Zus. zw.  $r_i$  und  $v_j$ :  $r_i^* = \bar{w}^i \cdot \vec{v}$ ,  $i = 1, 2, \dots, D$ , geeignete Wahl der  $\bar{w}^i$  z.B. die zu den  $D$  größten Eigenwerten gehör. Eigenvektoren

nichtlin. Zus. zw.  $r_i$  und  $u_i$ : Übergang von Hauptachsen zu gekr. Hauptflächen und von Eigenräumen zu Hauptmannigfaltigkeiten, die durch die  $\bar{w}^i$  der SOM aufgespannt wird !

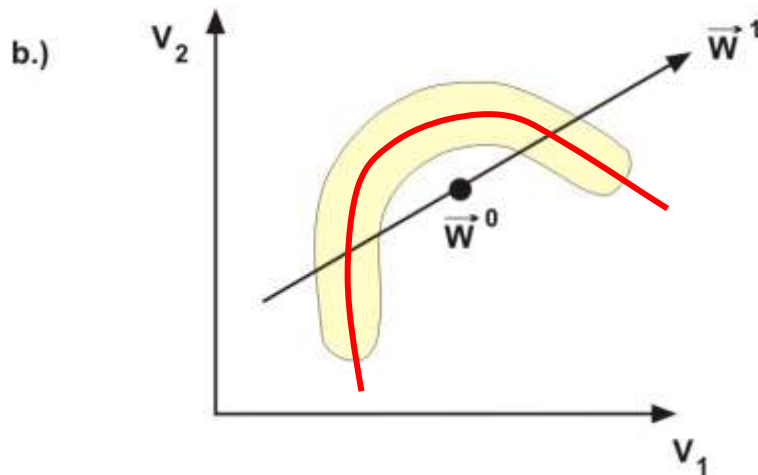
=>: \* Näherungsverfahren zur Berechnung von Hauptmannigfaltigkeiten !

\* dimensionsreduzierte Beschreibung nichtlinearer Datenverteilungen !

# Mathematische Eigenschaften der SOM



PCA : linear main axes

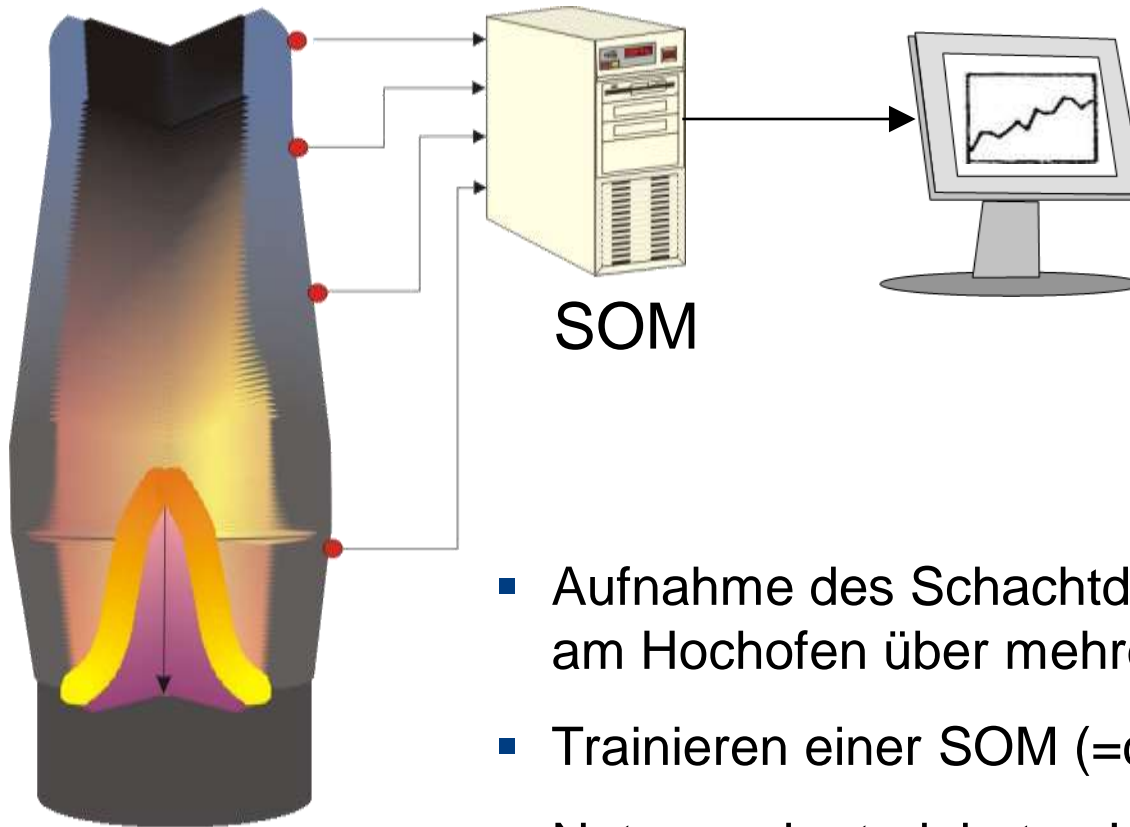


SOM: non-linear main curves / main surfaces

$\vec{W}$  = eigenvector of  $C$



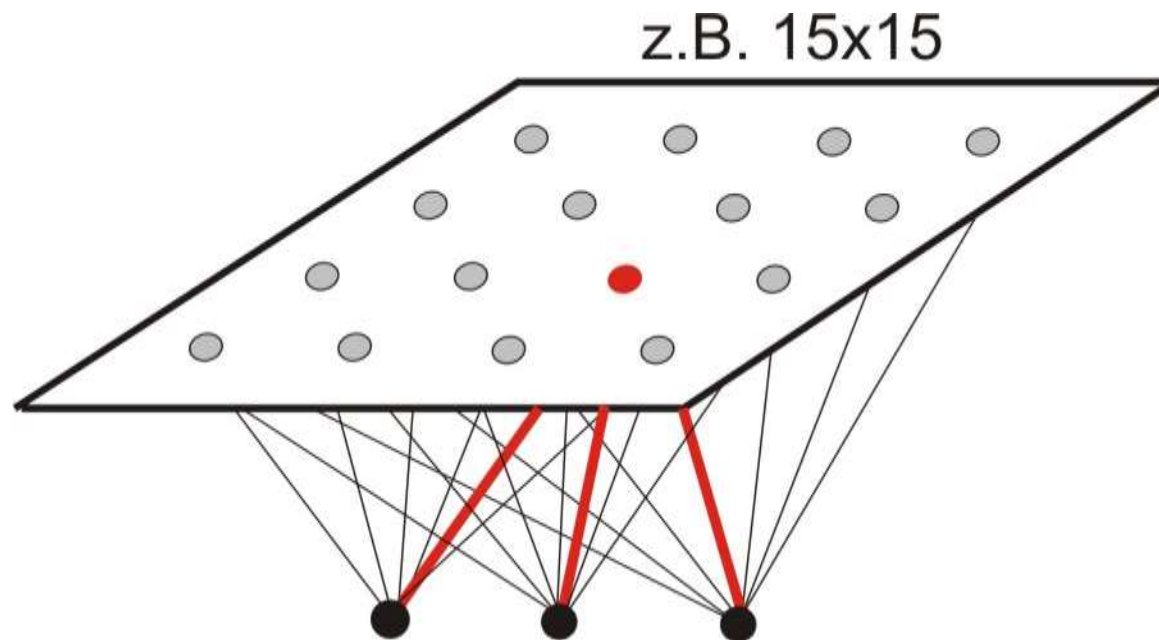
# Überwachung des Hochofenprozesses



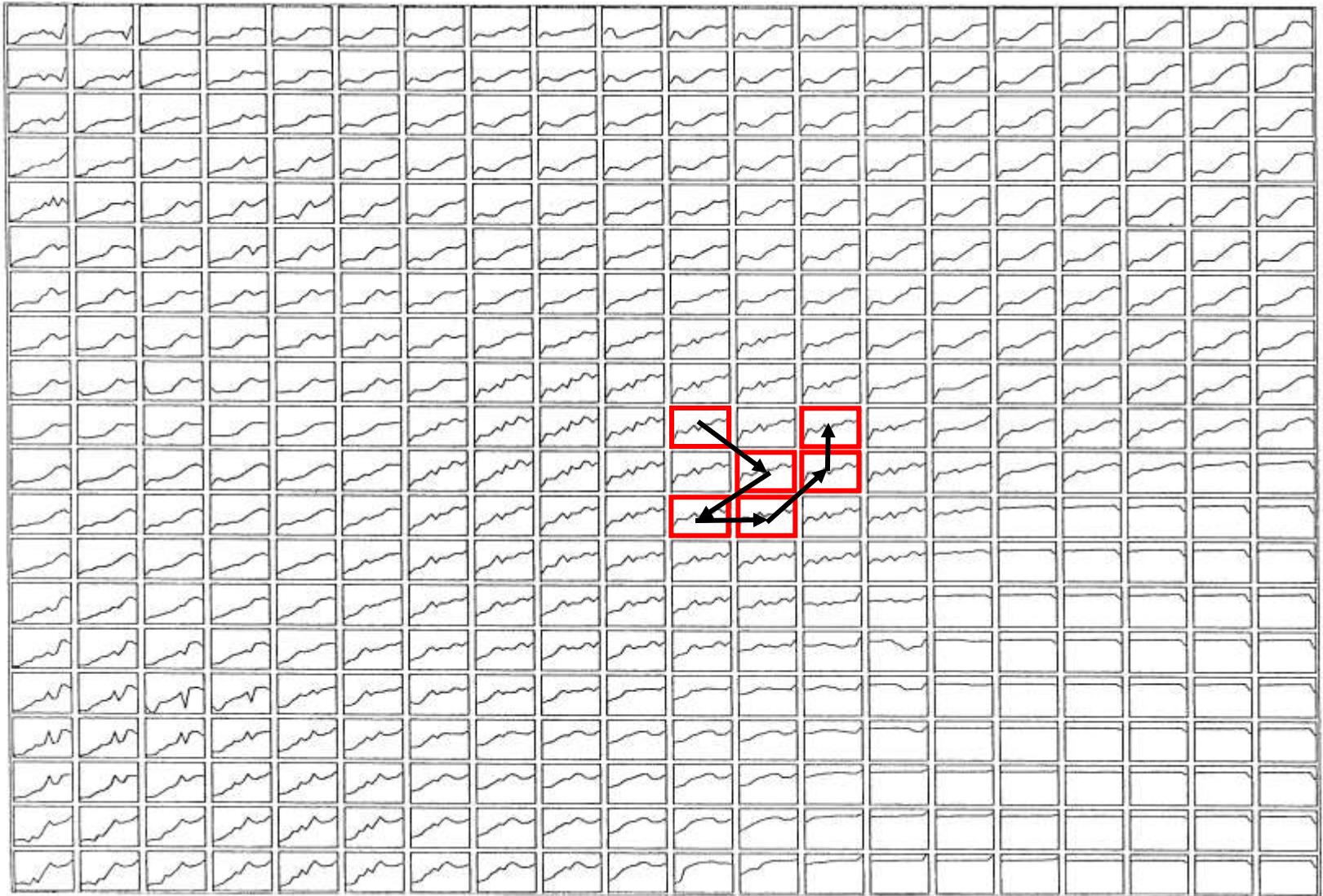
- Aufnahme des Schachtdrucks (über der Höhe) am Hochofen über mehrere Monate Produktion
- Trainieren einer SOM (=clustern der Signale)
- Nutzung der trainierten Karte zur online Überwachung des Schachtdrucks

Anwendung aus dem Jahr 1993

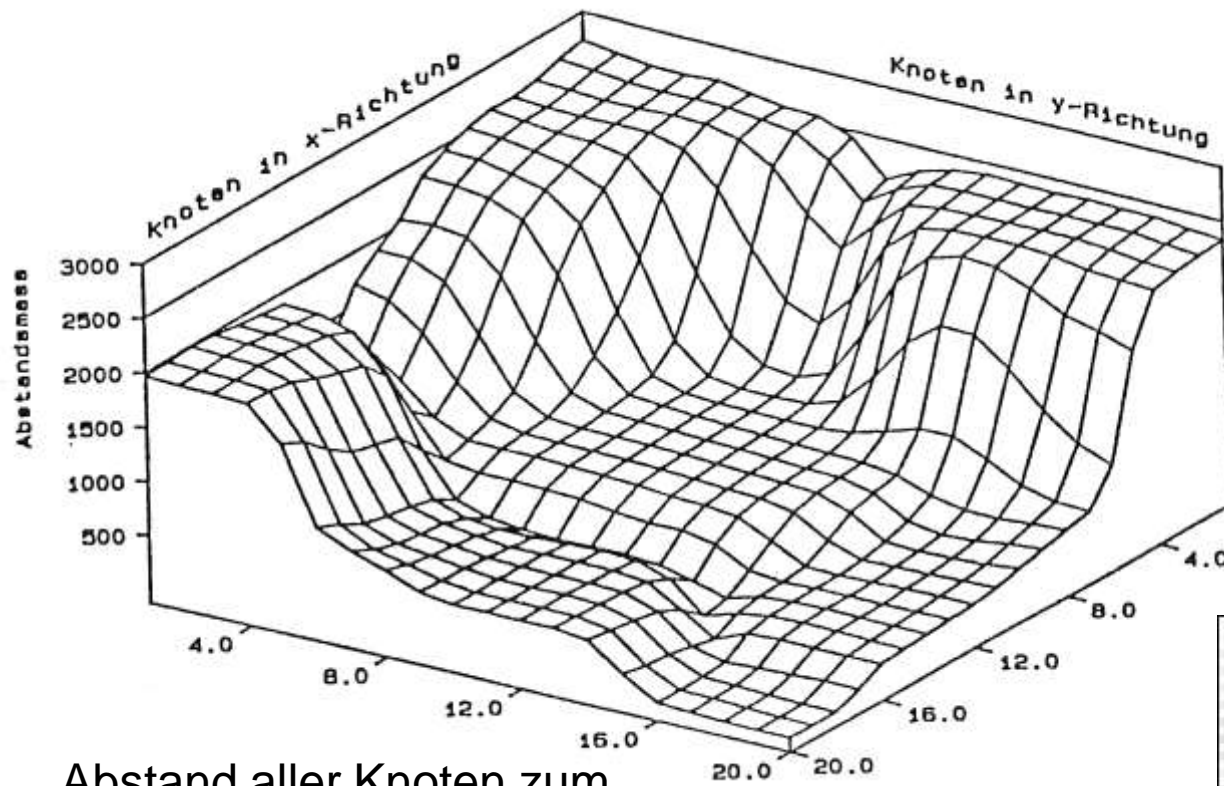
# Grafische Darstellung der Gewichte



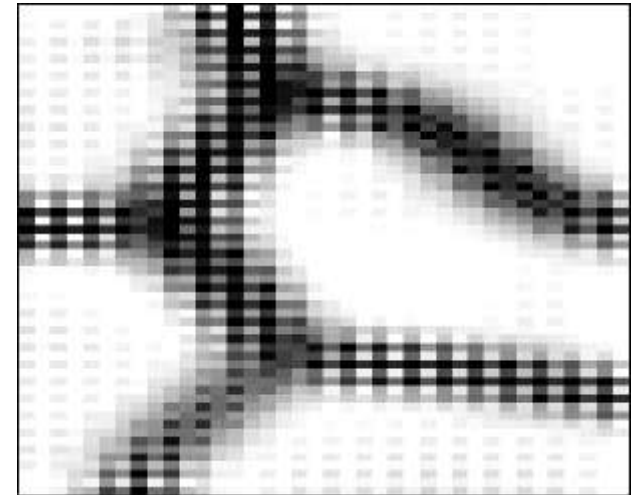
# Beispiel Schachtdruck Hochofen



# Andere Darstellung der Gewichtsmatrix

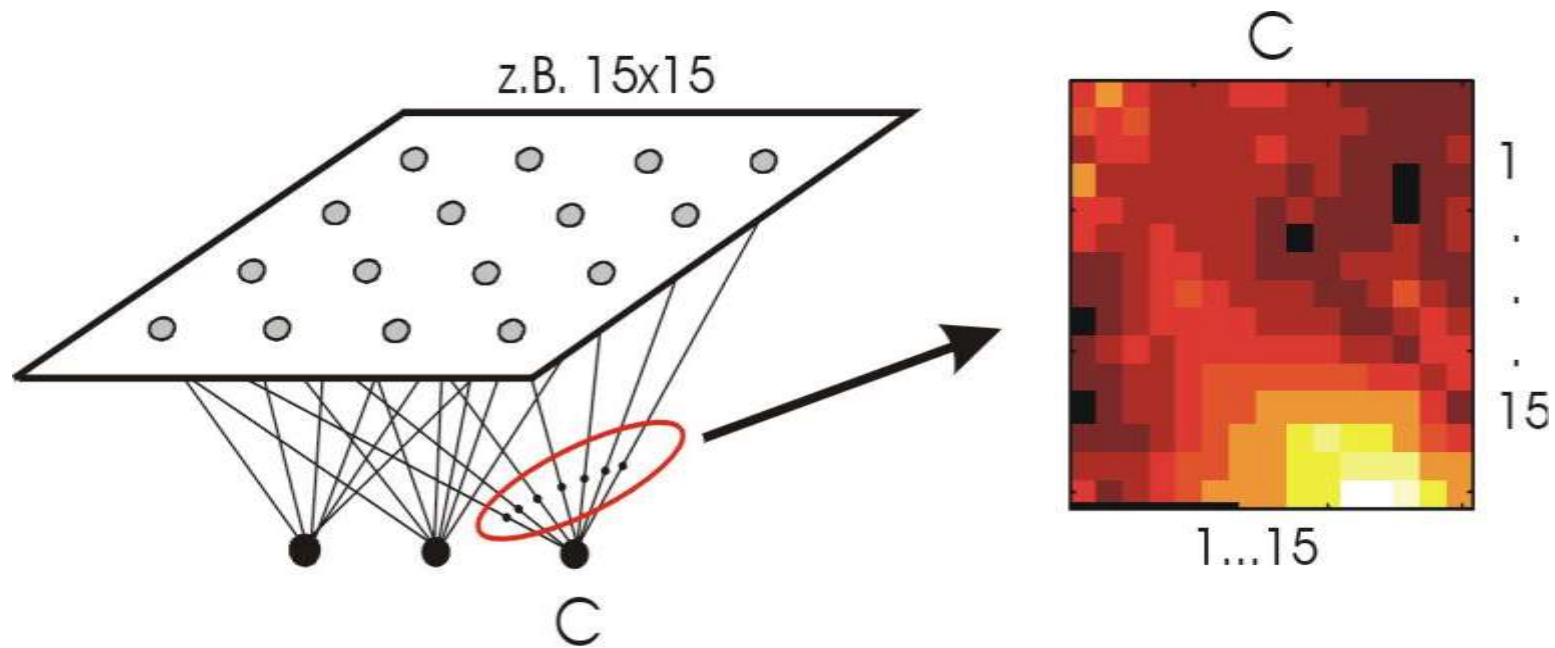


Abstand aller Knoten zum  
„mittleren“ Gewichtsvektor



U-Matrix

## „Component plane“ der SOM

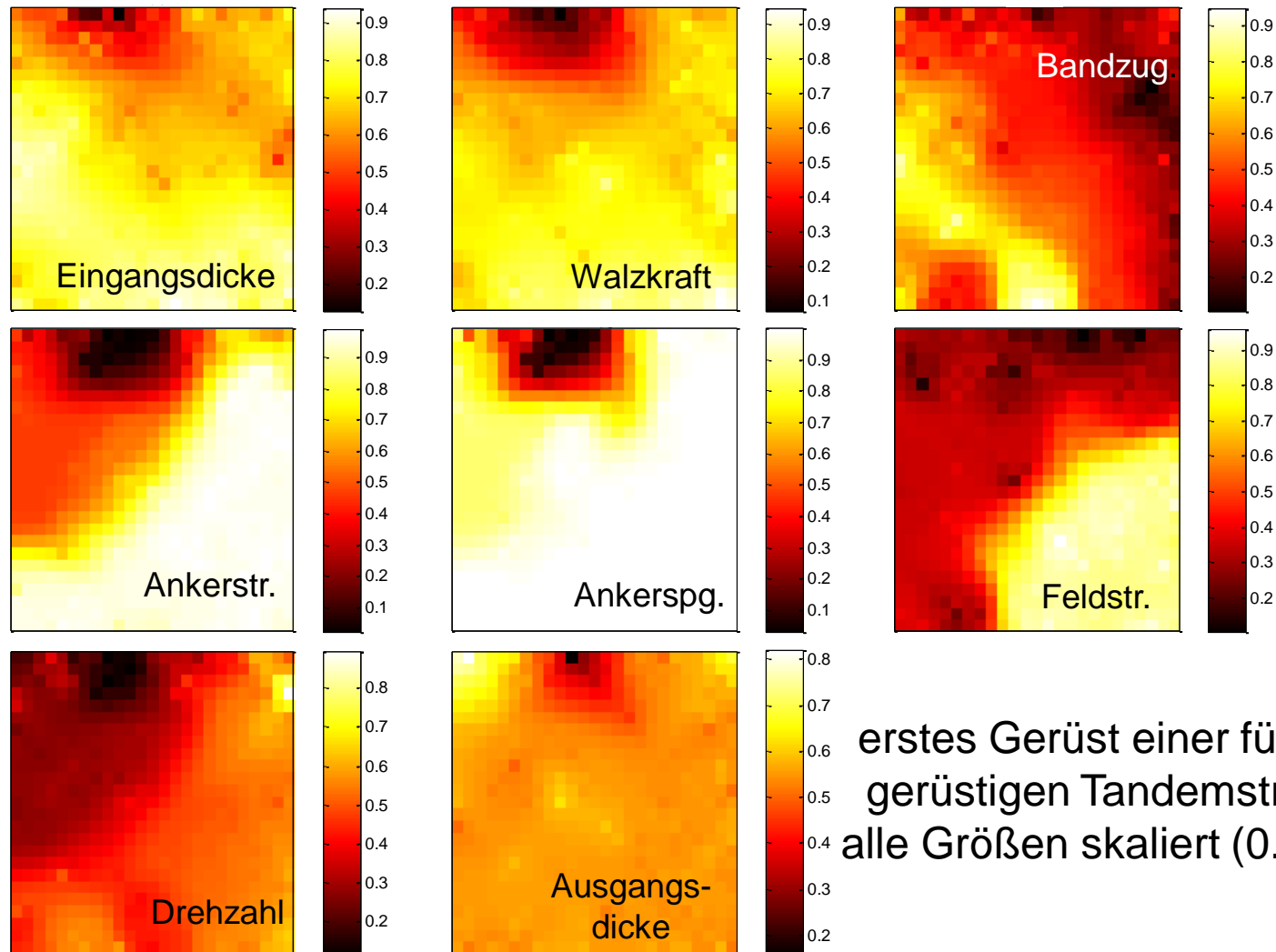


Die Gewichte zwischen einer Einflussgröße und allen Knoten der Ausgangsschicht werden in einer zweidimensionalen Karten durch Falschfarben dargestellt !

Die component planes aller Variablen sind nicht-linear miteinander verknüpft (da sie gemeinsam trainiert wurden)

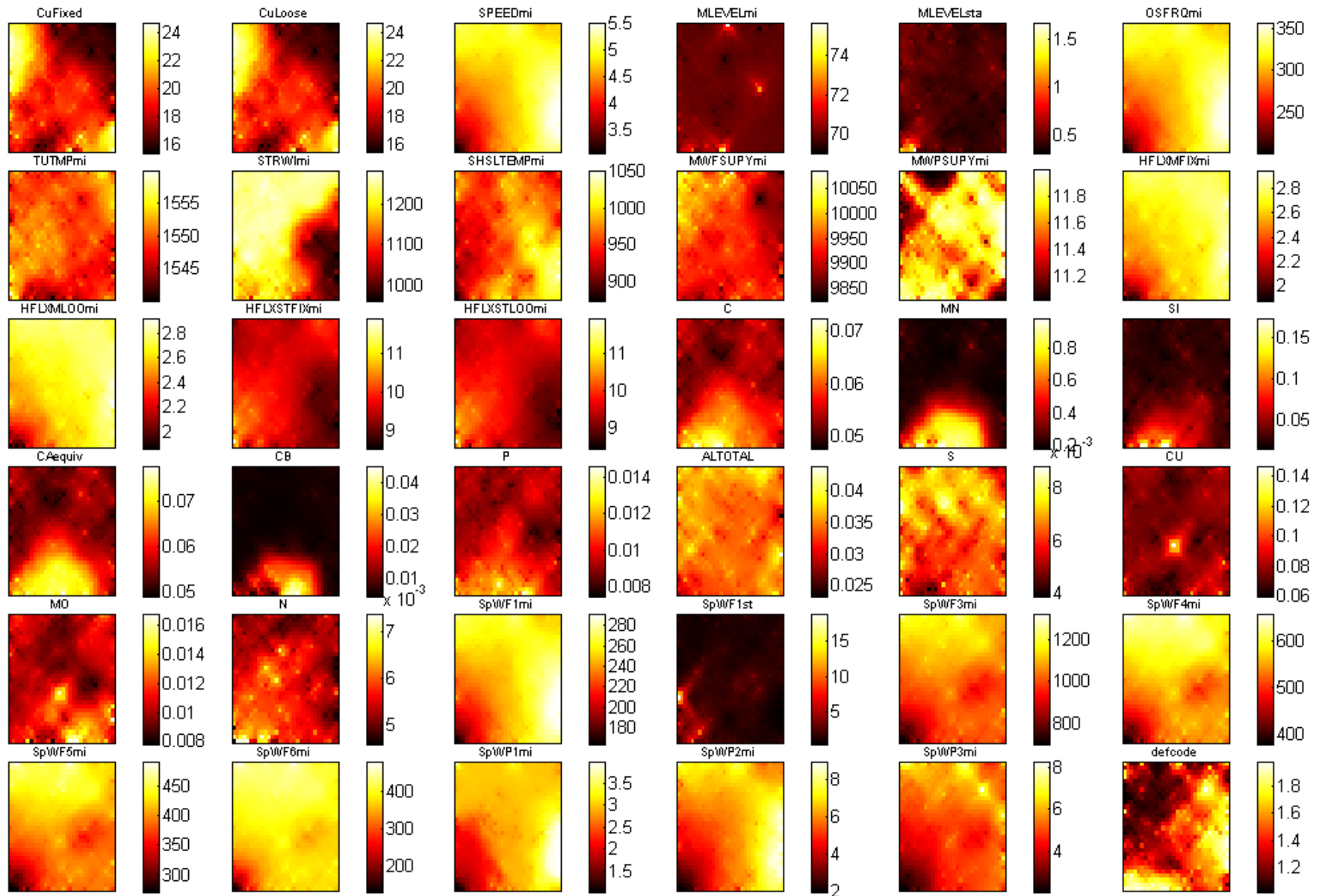


# Beispiel einer „Component plane“



03-Nov-2009 12:28:22

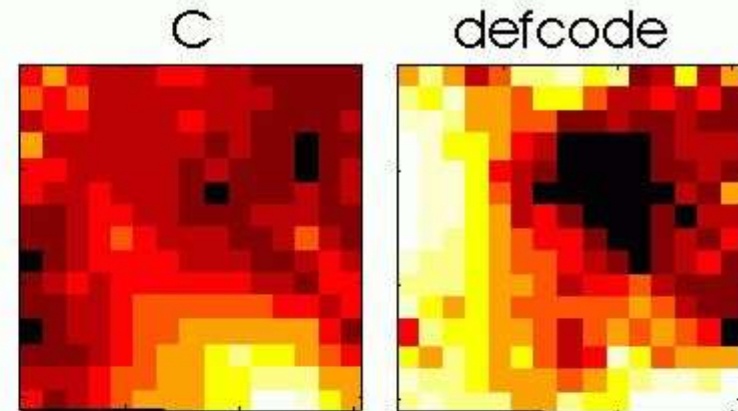
# Beispiel einer „Component plane“



# Auswertung der „Component plane“

1. visuell
2. global mittels LKK
3. lokal mittels LKK

$$r = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}}$$



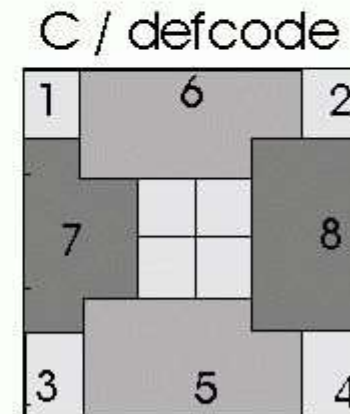
CompPlane priority - Microsoft Internet Explorer

Adresse [C:\TEMP\CompPriority.htm](http://C:\TEMP\CompPriority.htm)

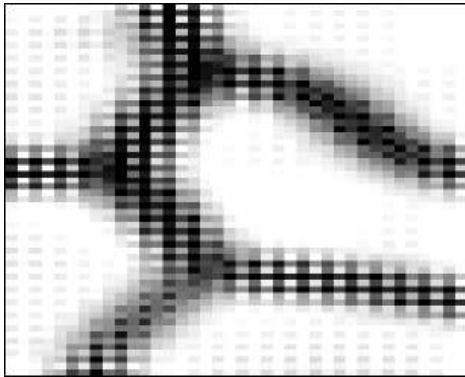
**Result of SOM component plane**

	Corr.Coeff	ranking
m_VG	0.59	1.00
std_MW_STOPFENSTELLUNG	0.54	2.00
FE_MO	0.51	3.00
std_MW_BADSP_IST	0.50	4.00
KO_AR_ZUGABE_GESAMT	0.48	5.00
std_MW_VG	0.46	6.00

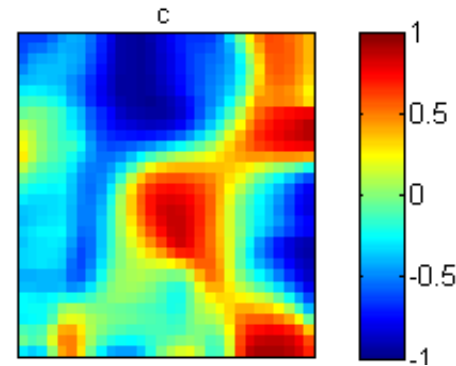
Fertig      Abbrückplatz



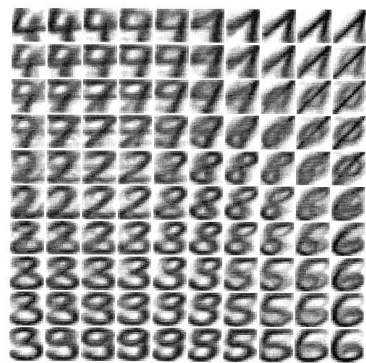
# Grafische Darstellungen der SOM



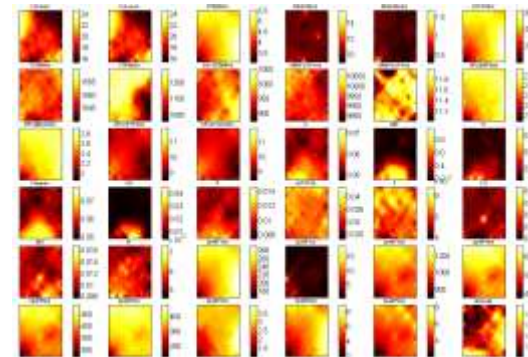
U-Matrix



Correlation Map



Gewichtsvektor



Component Plane

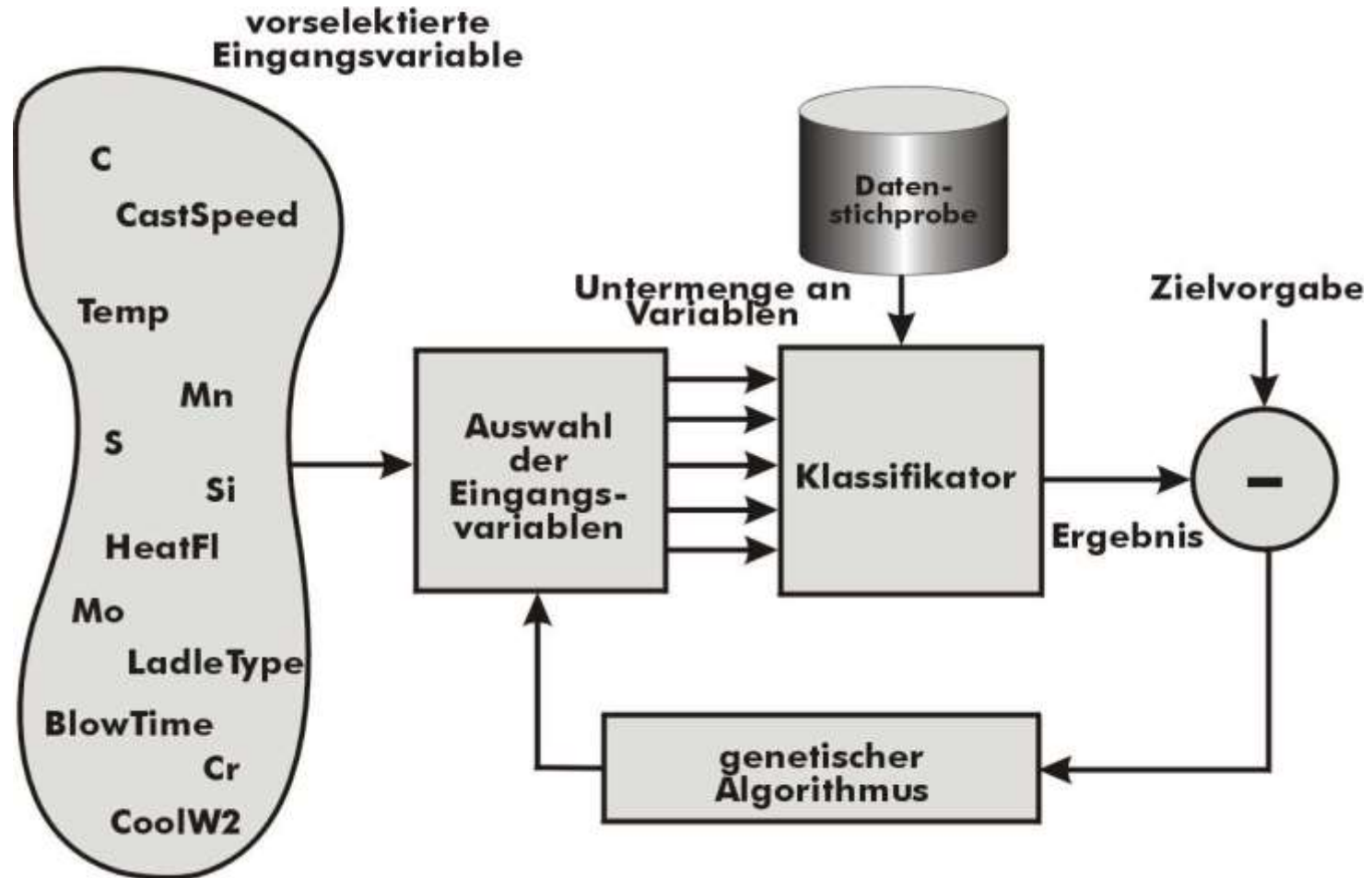
# Gliederung

---

1. Einleitung
  1. Motivation
  2. Was ist *Data Mining*
  3. Was versteht man unter *Big Data*
  4. Erstes Anwendungsbeispiel von *Data Mining*
2. Notwendige Verarbeitungsschritte des *Data Mining*
  1. Einführung
  2. Grafische Datenanalyse
  3. Datenvorverarbeitung
  4. Datenaggregation und Merkmalsberechnung
  5. Behandlung der Datenstichprobe
3. Einige grundlegende Methoden
  1. Clusteranalyse
  2. Künstliche neuronale Netze / Deep Learning Verfahren
4. Einfache statistische Abhängigkeitsanalyse
5. Komplexere Methoden der Abhängigkeitsanalyse
  1. Diskriminanzanalyse
  2. Entscheidungsbaumverfahren / Random Forests
  3. Self Organising Map
  - 4. Genetische Algorithmen**
6. Methoden der datenbasierten Modellbildung am Beispiel der Klassifikation
  1. Grundlagen der Klassifikation
  2. Statistische Methoden
  3. Neuronale Methoden
7. Anwendungsbeispiele

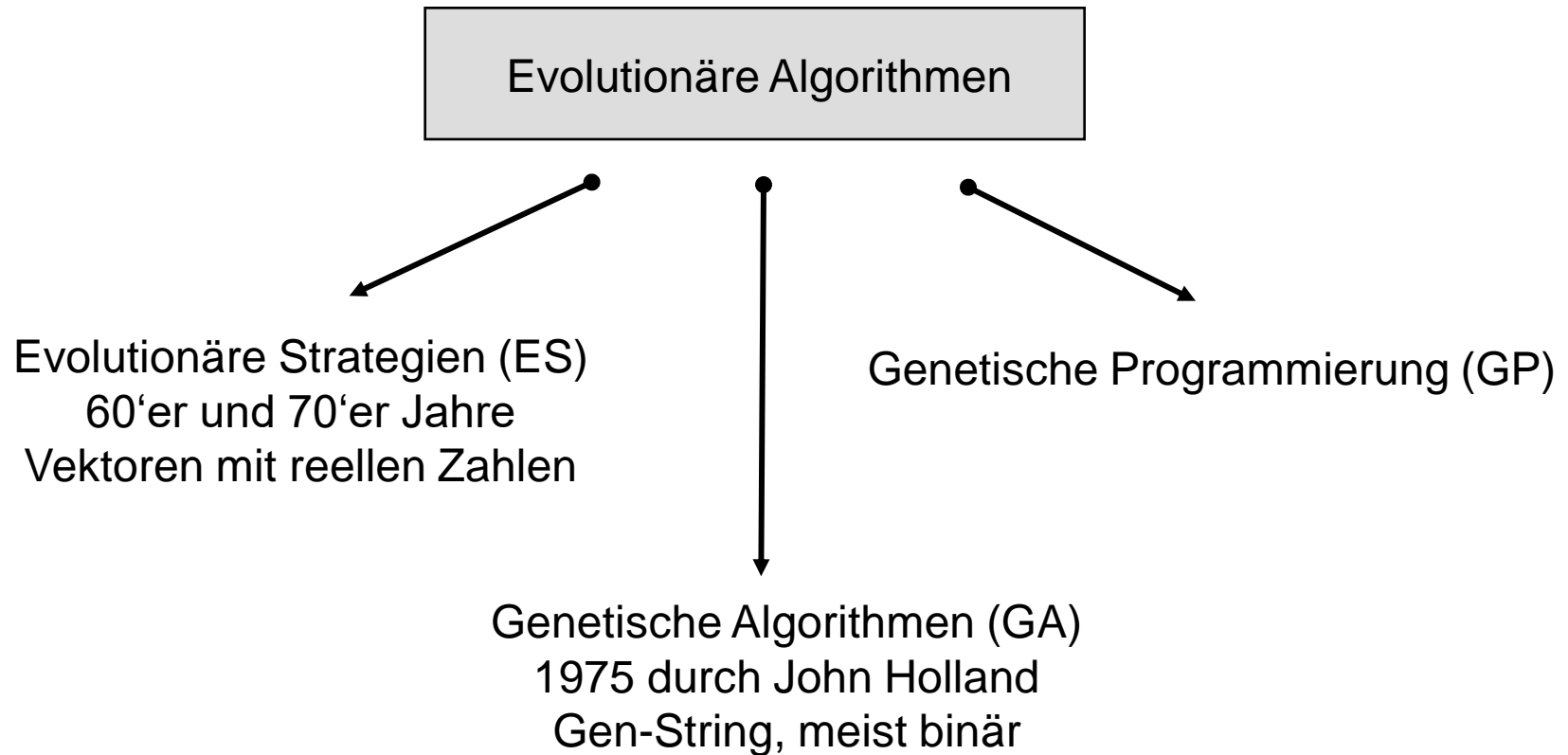


# „evolutionäre“ Variablenauswahl



# Evolutionäre Algorithmen

---

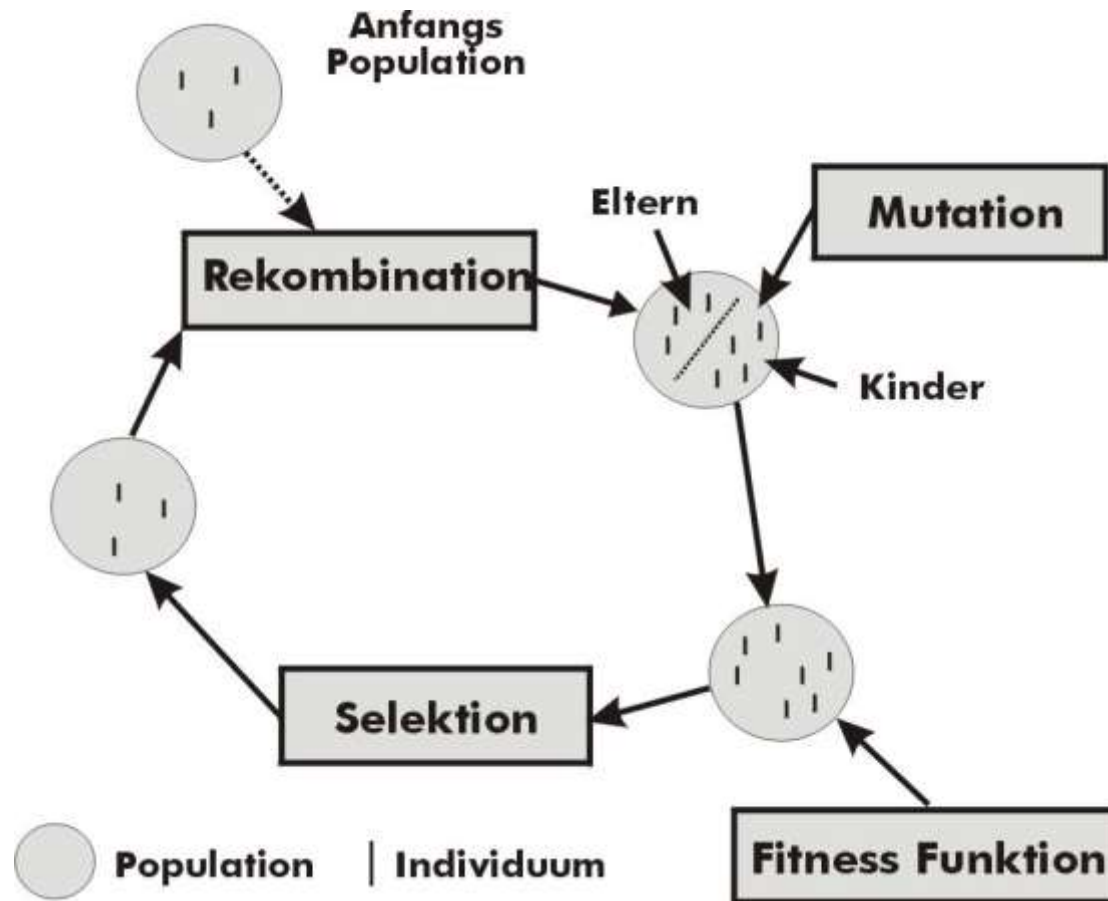


# Abhängigkeitsanalyse mit genetischen Algorithmen

---

- Situation:* es liegt eine klassifizierte Datenstichprobe mit  $M$  Eingangsvariablen vor
- Aufgabe:* es wird eine Kombination von  $N$  der vorliegenden  $M$  Eingangsvariablen gesucht, mit der die Abhängigkeit zwischen Ein- und Ausgang möglichst gut beschrieben werden kann.
- Voraussetzung:* es wird ein Klassifikationsverfahren festgelegt (z.B. Bayes-Klassifikator)
- Problem:* es muss ein hochdimensionales Optimierungsproblem gelöst werden, wobei keinerlei funktionalen Abhängigkeiten bekannt sind
- Lösungsansatz:* Optimierung mittels eines genetischen Algorithmus, die Kombinationen aus  $N$  Eingangsvariablen werden durch ein Genom beschrieben

# Prinzip des genetischen Algorithmus



# Bausteine eines genetischen Algorithmus

---

- Rekombination:* aus den Genen von zwei Eltern werden jeweils die Hälfte entnommen und zu den Genen eines Kindes zusammengesetzt. Die Auswahl welche Gene vom Vater und welche von der Mutter genommen werden erfolgt z.B. zufällig.
- Mutation:* Ein gewisser Teil der Gene wird zufällig geändert, d.h. bei binärer Beschreibung eines Genoms wird 0 in 1 verwandelt und umgekehrt.
- Selektion:* Basierend auf einer geeigneten Gütefunktion werden die besten Individuen in die nächste Generation übernommen.
- Reproduktion:* Ein Teil der bestehenden Population wird unverändert in die neue Population (Generation) übernommen.



# Skalare Fitness für einen Klassifikator

$$E = \sum_{i=1}^M \varepsilon_i \cdot \omega_i \quad \text{with} \quad \sum_{i=1}^M \omega_i = 1, \text{ e.g. } \omega_i = \frac{n_{c_i}}{n}$$

$$\text{Scalar fitness related to one class: } \varepsilon_i = \alpha_i \cdot \frac{m_{c_i}}{n_{c_i}} - \beta_i \cdot \sum_{j=1}^{M-1} \frac{m_{c_i}^*}{n_{c_i}}$$

$$\alpha_i, \beta_i \in [0,1] \quad \text{with} \quad \alpha_i + \beta_i = 1$$

$M$ : number of classes

$\omega_i$ : weight values

$m_{c_i}$ : right assigned to class  $i$ ,

$m_{c_i}^*$ : wrong assigned to class  $i$

$n_{c_i}$ : number of elements in class  $i$ ,

$n$ : total number of data

# Genetischer Algorithmus, Realisierung

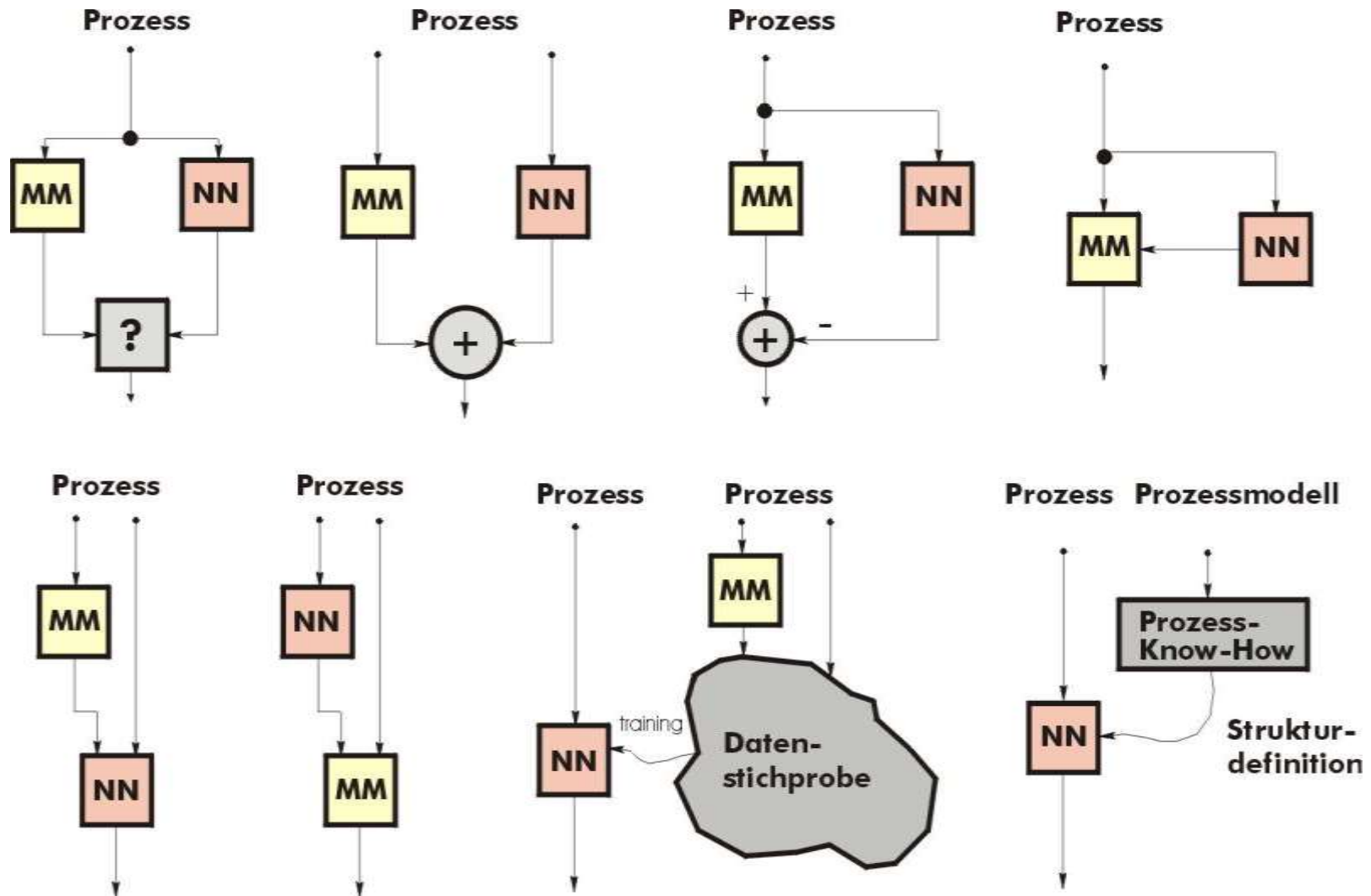
```
while(time<limit)
    % loop until the child generation is complete
    for i=1:genetic.PopulationSize
        % control of next genetic operation
        if zufall(i)<0.8 % 80% probability for recombination
            geneticOperation='recombine';
        elseif zufall(i)>0.8 & zufall(i)<0.85 % 5% probability for mutation
            geneticOperation='mutate';
        elseif zufall(i)>0.85 % 15% probability for reproduction
            geneticOperation='reproduce';
        end;
        % differentiate between several kinds of operations
        switch geneticOperation,
            case 'reproduce',
                dataspec=reproduceIndividuum(P00L1,data_meta);
            case 'recombine',
                % father
                VaterDatenSpez.nInputs=P00L1.nVar(index(anzinpool-indexrand(1)+1));
                % mother
                MutterDatenSpez.nInputs=P00L1.nVar(index(anzinpool-indexrand(2)+1));
                dataspec=Rekombiniere(VaterDatenSpez,MutterDatenSpez,genetic);
            case 'mutate',
                dataspec=Mutate(dataspec,data_meta);
            otherwise,
                end
        % here the classifier calculates the new model
        model=runClassifier(dataspec,genetic);
        % save the data of the individuum
        helpSave.DataSpec=dataspec;
        % actualise the pool of individuals
        P00L2=AbgleichPool(genetic,helpSave,P00L2,data_meta);
    end;
    % switch to the next generation
    genetic.anzGeneration = genetic.anzGeneration + 1;
    [fitness,index]=max(P00L1.F);
end % while
```

# Gliederung

---

1. Einleitung
  1. Motivation
  2. Was ist *Data Mining*
  3. Was versteht man unter *Big Data*
  4. Erstes Anwendungsbeispiel von *Data Mining*
2. Notwendige Verarbeitungsschritte des *Data Mining*
  1. Einführung
  2. Grafische Datenanalyse
  3. Datenvorverarbeitung
  4. Datenaggregation und Merkmalsberechnung
  5. Behandlung der Datenstichprobe
3. Einige grundlegende Methoden
  1. Clusteranalyse
  2. Künstliche neuronale Netze / Deep Learning Verfahren
4. Einfache statistische Abhängigkeitsanalyse
5. Komplexere Methoden der Abhängigkeitsanalyse
  1. Diskriminanzanalyse
  2. Entscheidungsbaumverfahren / Random Forests
  3. Self Organising Map
  4. Genetische Algorithmen
- 6. Methoden der datenbasierten Modellbildung am Beispiel der Klassifikation**
  1. Grundlagen der Klassifikation
  2. Statistische Methoden
  3. Neuronale Methoden
7. Anwendungsbeispiele

# Kombination von datenbasierten+analytischen Modellen



MM: Mathematisches Modell

NN: künstliches neuronales Netz

# Klassifikation und Regression

---

Alle innerhalb eines *Data Mining* Prozesses anfallenden Modellierungsaufgaben lassen sich auf die Gebiete **Klassifikation** und **Regression** abbilden:

*Klassifikation:* beliebige Eingangsgrößen, als Ausgangsgrößen werden Kategorien oder Klassen (diskrete Werte zwischen 1 und N, N ist typischerweise  $\leq 20$ , sowohl nominale als auch ordinale Variable) verwendet.

*Regression:* Ein- und Ausgangsgrößen sind in der Regel kontinuierliche Werte in einem festliegenden Werteintervall

Für beide Fälle muss eine geeignete (im statistischen Sinne vollständige) Datenstichprobe vorhanden sein.

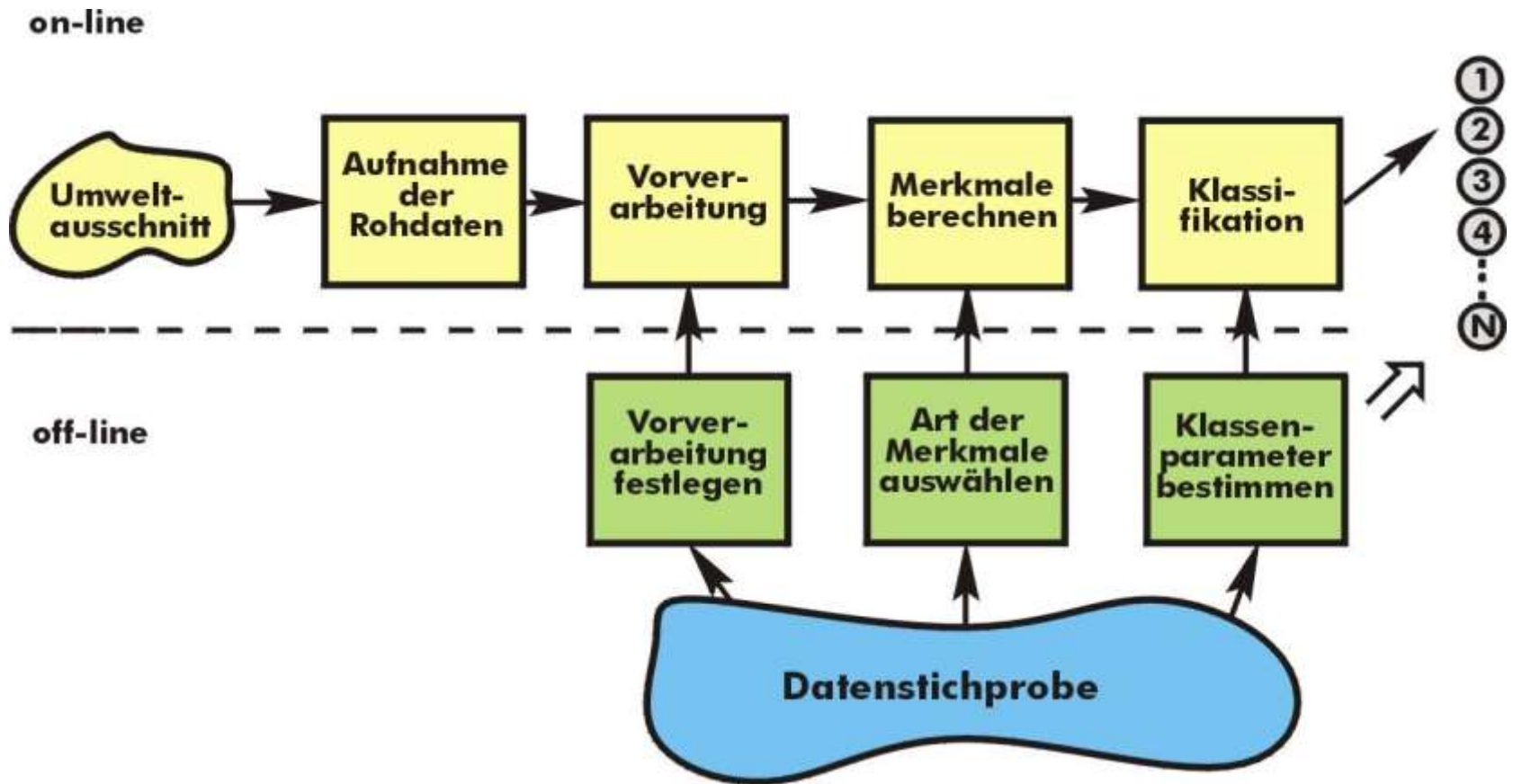
Grundprinzipien können teilweise in beiden Bereichen zum Einsatz kommen (z.B. multiple lineare Regression und Diskriminanzanalyse oder neuronale Methoden).

Das Einsatzgebiet dieser Modelle reicht von der Simulation über die Qualitätsprognose bis hin zum Einsatz in on-line Prozessregelungen.

Häufig werden solche „datenbasierten“ Modelle (auch: „datengetrieben“) in Kombination mit analytischen Modellen eingesetzt.



# Klassifikationssystem



# Schritte der Klassifikation

---

1. Aufbau der Klassenstruktur, auch Clusteranalyse genannt
2. Zuordnung eines aktuellen Eingangsmusters zu einer bereits existierenden Klasse
3. Anpassung der Klassenstruktur durch Eröffnung einer neuen Klasse bzw. Löschen / Zusammenfassen von Klassen

# Lernverfahren

---

1. „supervised learning“, überwachtes Lernen: zu jedem Eingangsdatensatz gibt es einen zugeordneten Ausgangswert (einen „Lehrer“)
2. „unsupervised learning“, unüberwachtes Lernen: es gibt nur Eingangsvektoren, keine Ausgangsinformationen
3. „reinforced learning“: Verstärkungslernen, der Lehrer sagt nach jedem Iterationsschritt nur: „Du wirst besser“ oder „Du wirst schlechter“

# Abstandsmessende Verfahren

Der „Abstand“ zwischen den Datenvektoren wird hier als Maß für die Ähnlichkeit der zugehörigen Eingangsinformationen herangezogen. Zur Berechnung dieses „Abstandes“ gibt es unterschiedliche Möglichkeiten, z.B.:

**Euklidischer Abstand**

$$D_{jk} = \|\vec{v}_j - \vec{v}_k\| = \sqrt{\sum_{i=1}^n (\vec{v}_{ji} - \vec{v}_{ki})^2}$$

$\vec{v}_i, \vec{v}_j$  : Merkmalvektor der Dimension n

**L<sub>r</sub>-Distanz**

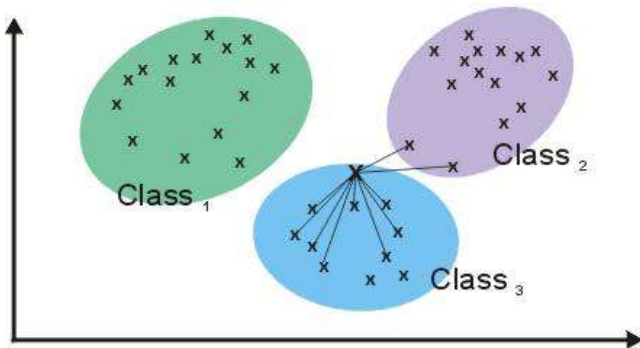
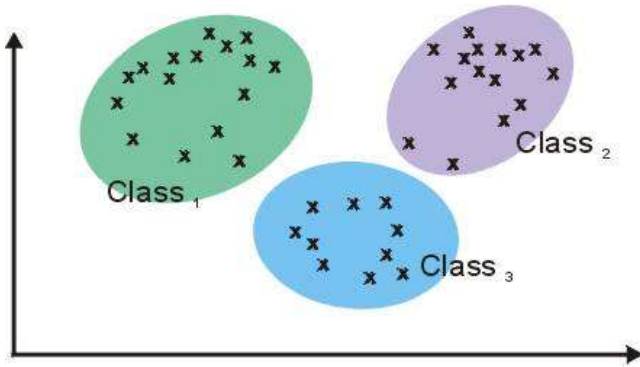
$$D_{jk}^r = \sqrt[r]{\sum_{i=1}^n |v_{ji} - v_{ki}|^r}$$

**Mahalanobis-Abstand**

$$D_{jk}^2 = (\vec{v}_j - \vec{v}_k)^T R^{-1} (\vec{v}_j - \vec{v}_k)$$

R: Kovarianz-Matrix von  $v_1$  und  $v_2$

# Nächster-Nachbar-Klassifikator



(1) Setting parameters

$N_{nn}$  : number of nearest neighbours  
taking into account

$P$  : limit value (percent) for assignment

(2) Calculate all distances  $d_{ik} := ||x_i - x_k||$

(3) Take the  $N_{nn}$  nearest neighbours of vector  $x_j$

(4) If more than  $P$  % of these neighbour  
vectors belong to class  $c_k$

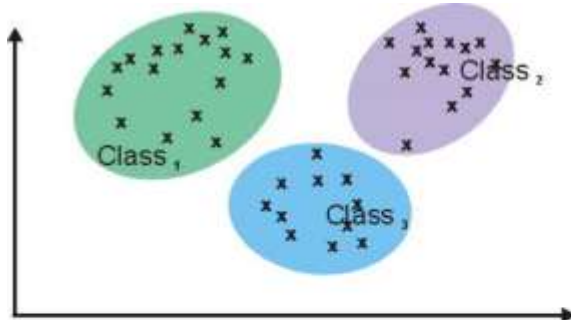
Then

$x_j$  is assigned to class  $c_k$

Else

$x_j$  is not assigned to any class

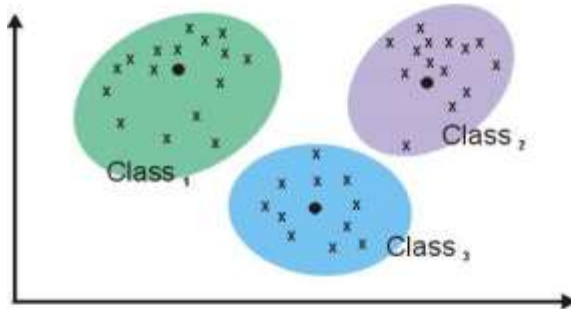
# Nächster-Prototyp-Klassifikator



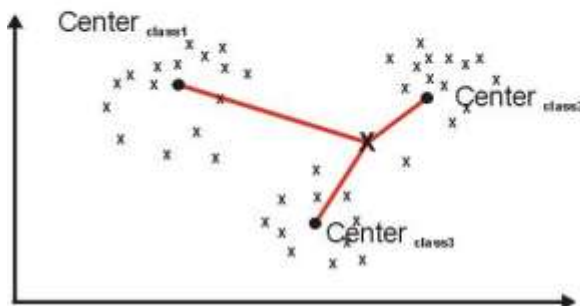
- (1) Calculate the centers of all classes by

$$c_{Cj} = 1/N_{Cj} * \sum x_{Cj}$$

(  $x_{Cj}$  : elements of class  $cj$  )



- (2) Calculate all distances  $d_{ij} := ||x_i - c_{Cj}||$



- (3) If distance  $d_{lk} = ||x_l - c_{Ck}|| = \min( d_{ij} )$  Then  
 $x_l$  is assigned to class  $c_k$

# Bayes-Klassifikator

(1) Calculate the following values :

probability of occurrence of class i

$$P(c_i) = \frac{n_{ci}}{n_{total}}$$

probability density function

$$p(\underline{x} | c_i) = \frac{1}{\sqrt{(2\pi)^n * |\underline{C}_i|}} * e^{\left(\frac{-1}{2} * (\underline{x} - m_i)^T * \underline{C}_i^{-1} * (\underline{x} - m_i)\right)}$$

$m_i$  : mean value of all  $\underline{x}$  of class  $c_i$

$\underline{C}_i$  : covariance matrix of all  $\underline{x}$  of class  $c_i$

(2) For new  $\underline{x}$  the following value is calculated

$$p(c_i | \underline{x}) = \frac{p(\underline{x} | c_i) * P(c_i)}{\sum_r p(\underline{x} | c_r) * P(c_r)}$$

$r$  : 1 ....no. of classes

(3) Assign  $\underline{x}$  to the class with the lowest  $r_j(\underline{x})$

$$r_j(\underline{x}) = \sum_{i=1}^M L_{ij} * p(c_i | \underline{x})$$



# Bayes-Klassifikator, Verlustfunktion

---

Alle Klassen sind gleichwertig:

$$L_{ij} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Die Klassen haben eine Ordnung:

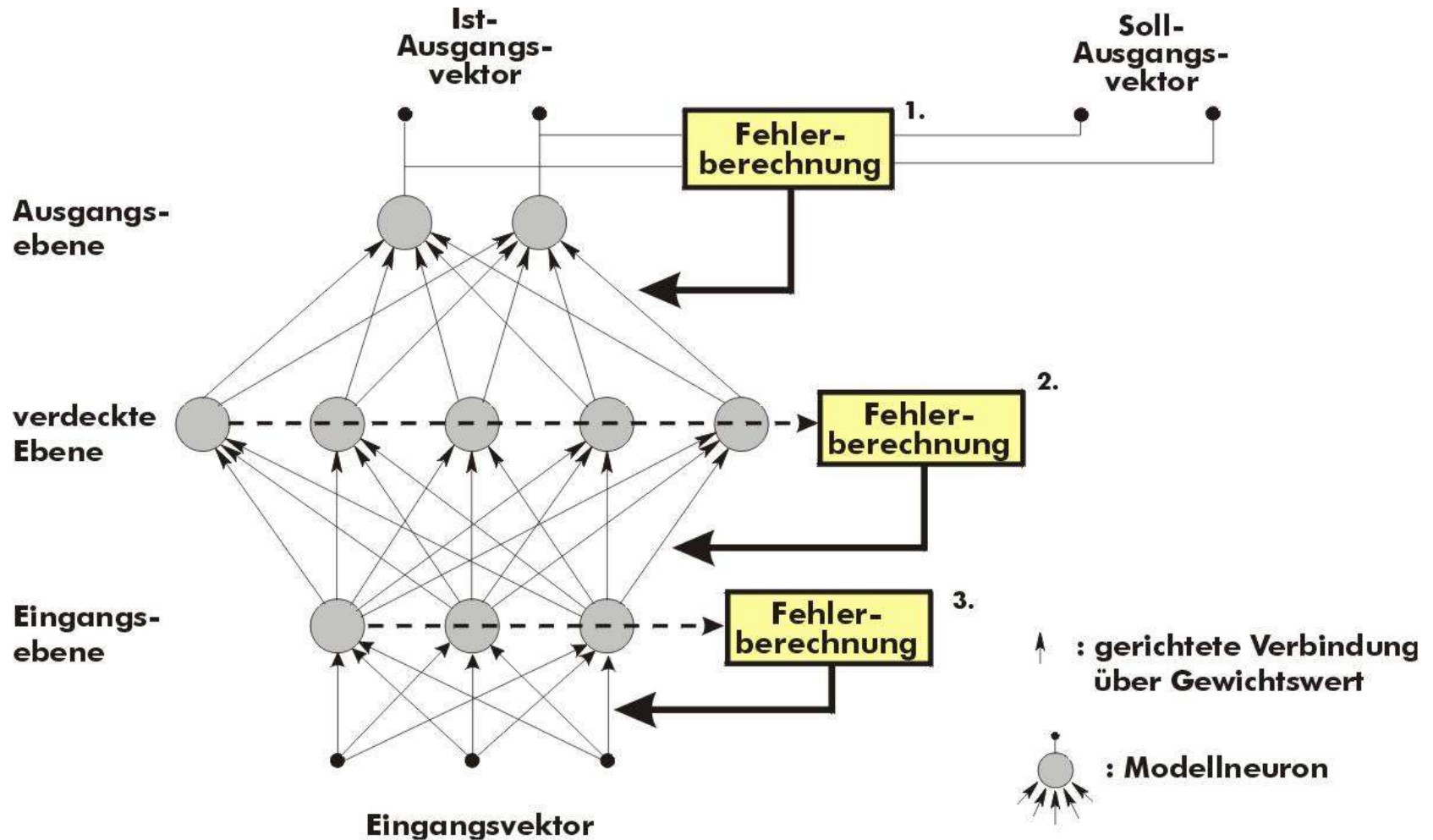
$$L_{ij} = \begin{bmatrix} 0 & 0,33 & 0,66 & 1 \\ 0,33 & 0 & 0,33 & 0,66 \\ 0,66 & 0,33 & 0 & 0,33 \\ 1 & 0,66 & 0,33 & 0 \end{bmatrix}$$

# Neuronale Methoden zur Klassifikation

---

- \* Perceptron
- \* Adaline
- \* Madaline
- \* **Multi-Layer Perceptron**
- \* Time-Delay Networks
- \* Elman- / Jordan Nets
- \* Associative memory
- \* Hopfield Network
- \* Haken Network
- \* Hamming Network
- \* Bidirectional Associative Memory
- \* Brain-State-in-a-Box
- ~~\* Adaptive Resonance Theory 1,2,3~~
- \* ARTMAP
- \* Fuzzy ART
- ~~\* Kohonen Self-Organizing Map SOM~~
- \* Counterpropagation
- \* **Radial Basis Function Network**
- \* Hyper Basis Function Network
- \* Boltzmann Machine
- \* Probabilistic Networks
- \* Neocognitron
- ~~\* Learning Vectorquantisation~~
- \* Wavelet-Network
- ~~\* Filter-Network~~
- ~~\* General Regression Networks~~

# Erinnerung: Multi-Layer-Perceptron-Netzwerk

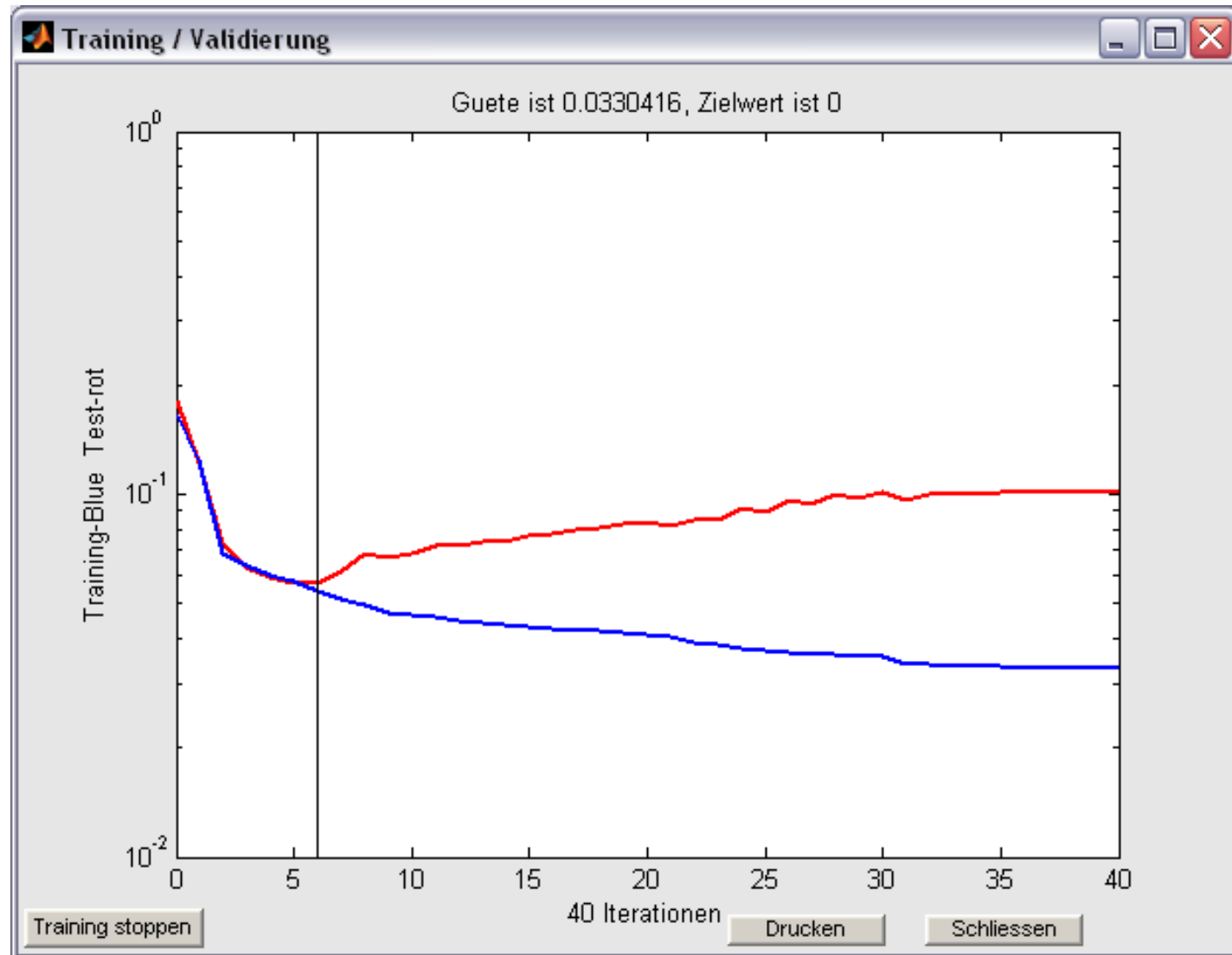


# Nutzung MLP als Klassifikator

---

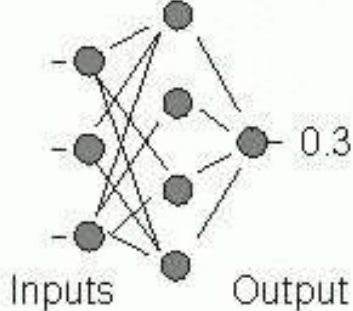
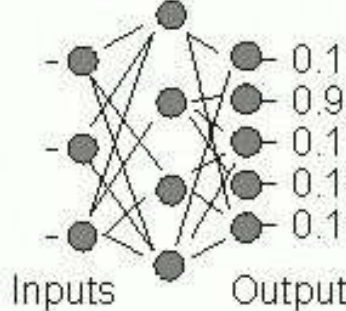
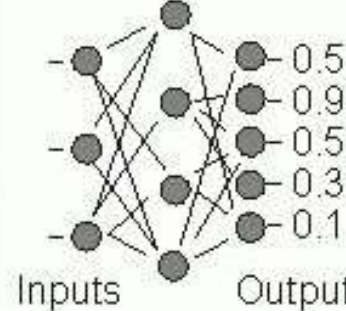
- Die Anzahl der Knoten im Netz und die Anzahl der Datensätze in der Trainingsstichprobe sollten in einem „vernünftigen“ Verhältnis stehen (Faustregel: mindestens 2x mehr Datensätze als freie Parameter)
- Zur Vermeidung von „Overfitting“ kann zusätzlich die Technik des „early stopping“ angewendet werden
- Die Datensätze aus der Stichprobe sollten dem Netz in einer bezüglich der Sollklassen abwechselnden Reihenfolge angeboten werden
- Die Codierung der Ausgangsvariable ist wesentlich, hängt von der Anzahl der Klassen ab und davon, ob die Klassen eine Ordnung haben
- Ein MLP als Klassifikator liefert zunächst kein eindeutiges Ergebnis, auf den Ausgang des Netzes muss noch eine „Entscheidungsregel“ angewendet werden.

# „Early stopping“



# Kodierung von Ausgangsvariablen

... Z.B. für MLP und RBF :

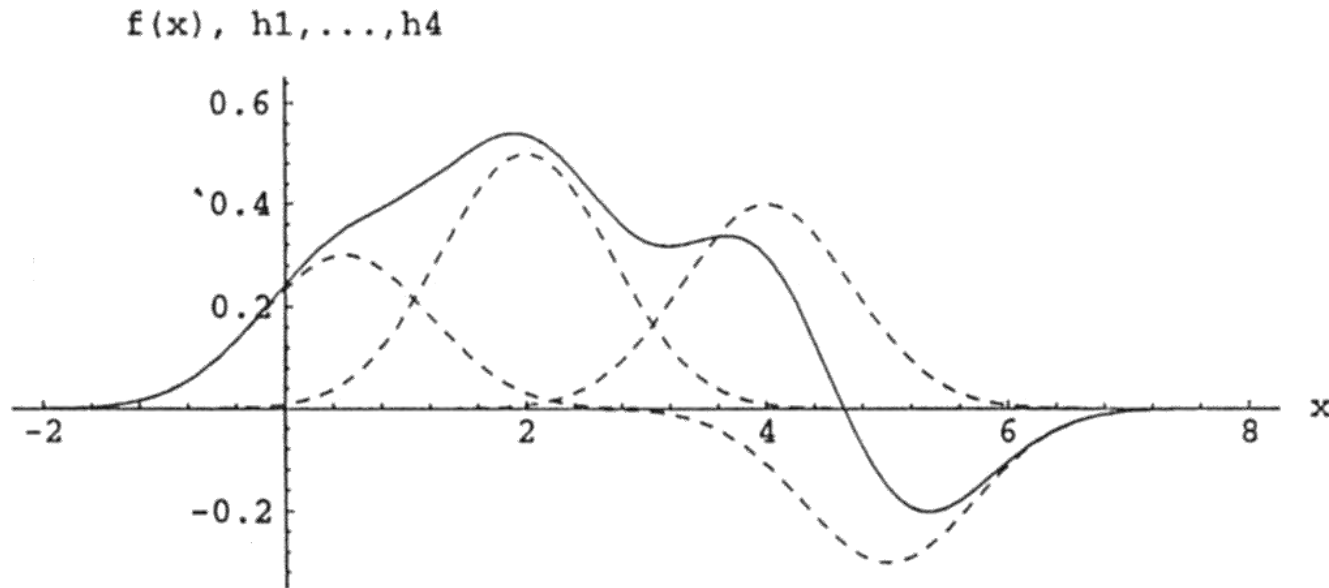
name	single coding	multiple coding I	multiple coding II
no. of nodes	one output node	N output nodes	N output nodes
values of nodes	N values, equidistant in interval [0.1 0.9] (N = no. of classes)	0.1 or 0.9	between 0.1 and 0.9, values represent similarities of classes
example	<p>target : class 2 of 5</p> 	<p>target : class 2 of 5</p> 	<p>target : class 2 of 5</p> 

# Entscheidungsregeln

	One output node (the number of classes is 'distributed' into a range between 0.1 and 0.9)	Several output nodes (Each class is represented by one output node, its target value = 0.9, for all other cases = 0.1)
<b>Winner takes all</b>	<p>There exist sharp ranges for the classes</p> <p>e.g. : 3 classes =&gt; 0.1 / 0.5 / 0.9 ranges =&gt; ... 0.3...0.7... target : class 2 of 3 output : 0.52 =&gt; class 2</p>	<p>the node with highest value will set the class</p> <p>e.g. ( 0.23   0.56   0.45 ) =&gt; class 2</p>
<b>40-20-40</b>	<p>Between the values of two classes exist three ranges in a proportion of 40-20-40</p> <p>e.g. : 3 classes =&gt; 0.1/0.5/0.9 ranges 0.1..0.26..0.34..0.5           0.5..0.66..0.74..0.9 i.e. Out &lt; 0.26 =&gt; class 1       &gt; 0.34 and &lt; 0.66 =&gt; class 2                   &gt; 0.74 =&gt; class 3 in all other cases =&gt; none</p>	<p>The output range of each node is divided into 3 ranges of a proportion of 40-20-40</p> <p>output range : 0.1 ... 0.9 decision ranges : (40%): 0.1 ... 0.42 : not to this class (20%): 0.42..0.58 : don't know (40%): 0.58..0.90 : assigned</p> <p>e.g. 0.23   0.66   0.45         not class 1         class 2         don't know</p>



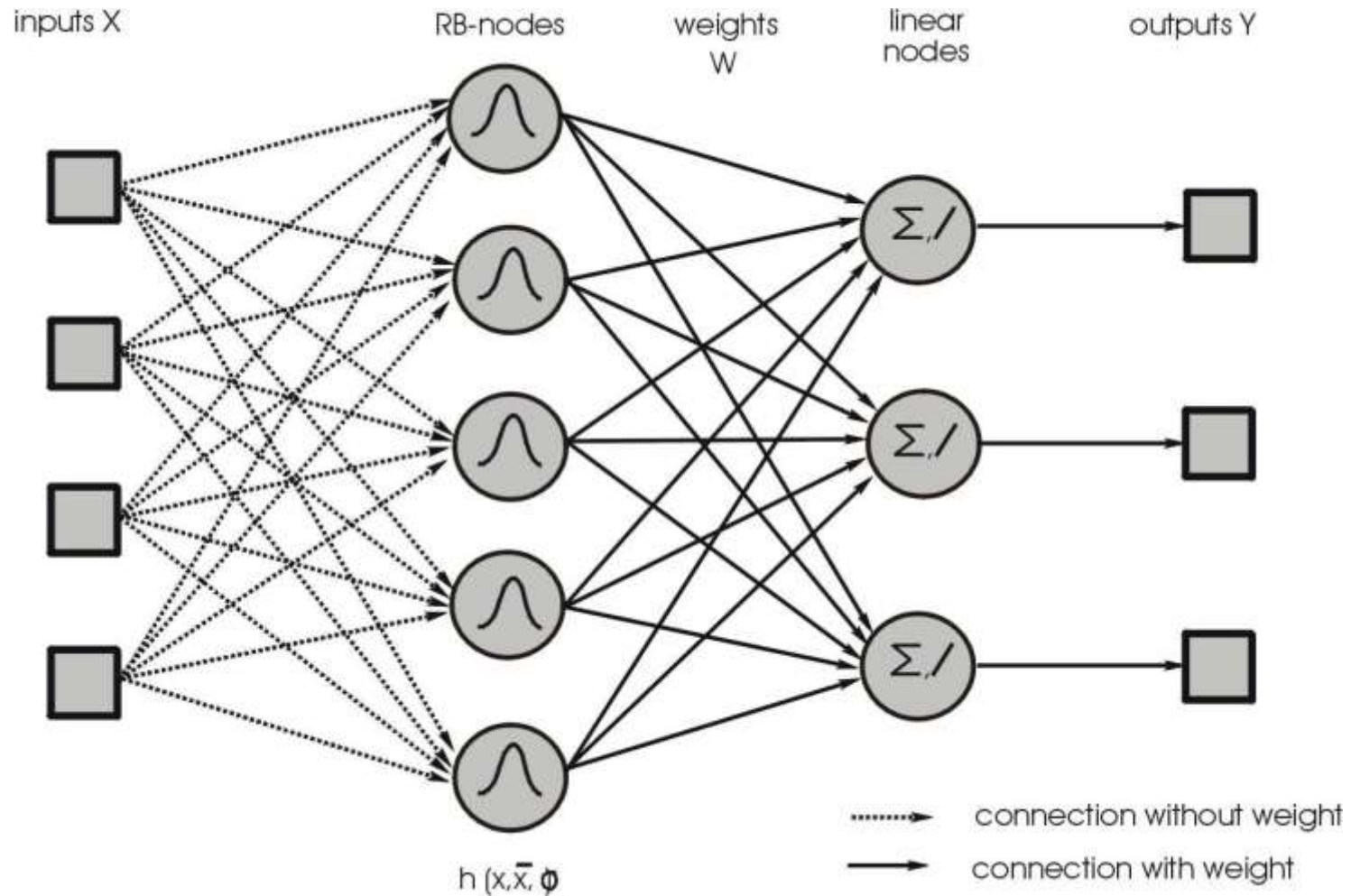
# Prinzip der radialen Basisfunktionen-Netze



**Abb. 20.2:** Überlagerung von vier Zentrumsfunktionen zu einer Summenfunktion. Hier wird die Funktion  $f(x) = 0.3 h(\|x-0.5\|) + 0.5 h(\|x-2\|) + 0.4 h(\|x-4\|) - 0.3 h(\|x-5\|)$  aus der Summe der einzelnen Gaußfunktionen  $h$  dargestellt. Für die vier Zentren  $x_1 = 0.5, x_2 = 2, x_3 = 4$  und  $x_4 = 5$  wurden die Gewichtungsfaktoren  $c_1 = 0.3, c_2 = 0.5, c_3 = 0.4, c_4 = -0.3$  verwendet.

Aus: A. Zell, Simulation Neuronaler Netze, Addison-Wesley, Bonn 1994, S. 227

# RBF-Netze (radiale Basisfunktionen Netze)



# Ergebnisdarstellung I

**Classification Results training - Microsoft Internet Explorer**

Datei Bearbeiten Ansicht Favoriten Extras ?

← Zurück → Suchen Favoriten Verlauf

Adresse C:\Dokumente und Einstellungen\demo\Lokale Einstellungen\Temp\ClResultstraining.htm Wechseln zu Links »

Time: 05-Jan-2004 15:02:10  
Classification Method: MLP 46 - 10 - 0 - 5 Levenberg/Marquardt, 10 iterations  
Decision rule: Winner takes all  
Scalar Fitness: -0.1600

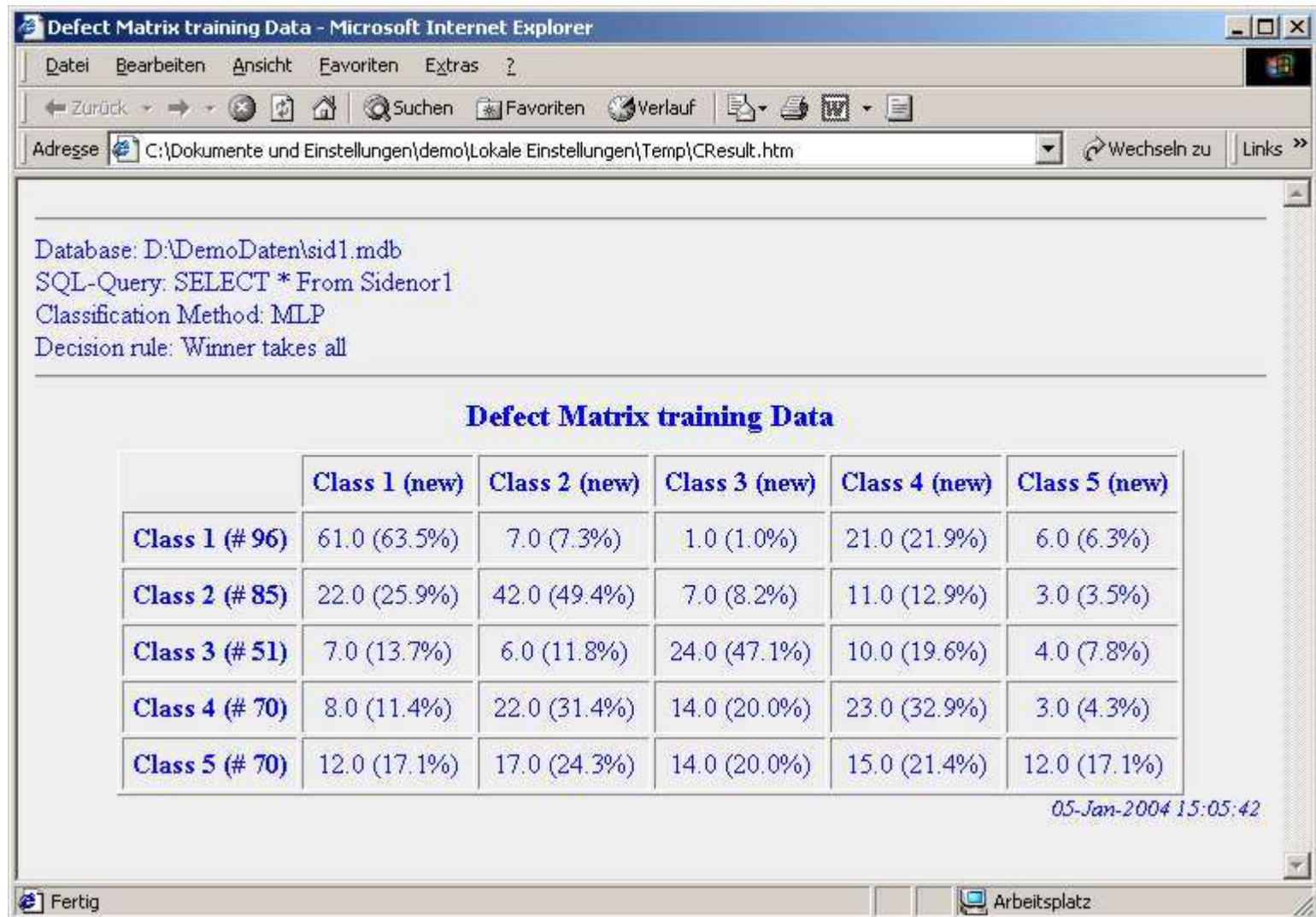
---

**Classification Results training**

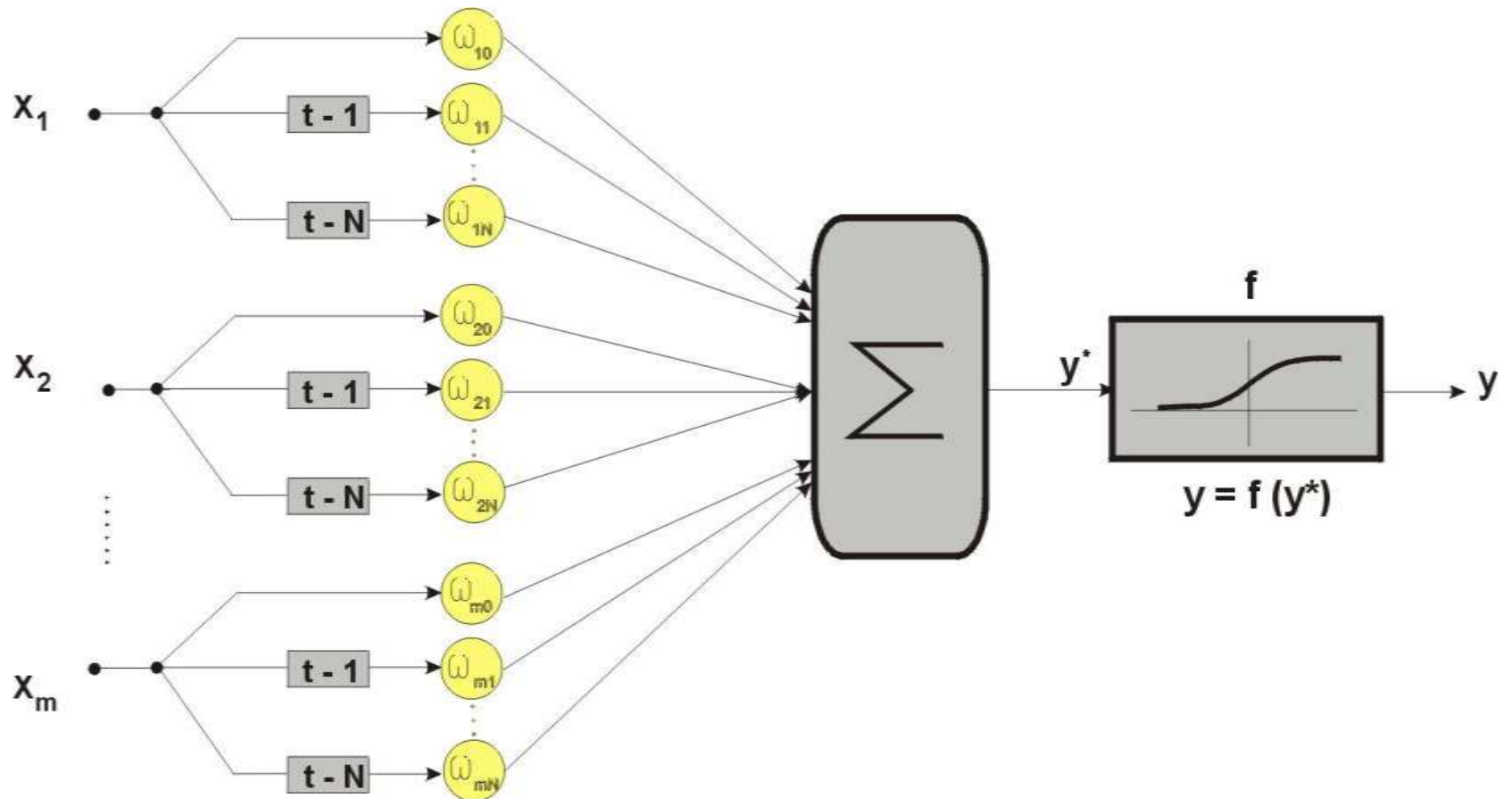
	Number of Data-Sets	right assigned	wrong assigned	fuzzy right assigned	fuzzy wrong assigned	no assignment possible
1	96 (100%)	61 (64%)	35 (36%)	0 (0%)	0 (0%)	0 (0%)
2	85 (100%)	42 (49%)	43 (51%)	0 (0%)	0 (0%)	0 (0%)
3	51 (100%)	24 (47%)	27 (53%)	0 (0%)	0 (0%)	0 (0%)
4	70 (100%)	23 (33%)	47 (67%)	0 (0%)	0 (0%)	0 (0%)
5	70 (100%)	12 (17%)	58 (83%)	0 (0%)	0 (0%)	0 (0%)
All Classes	372 (100%)	162 (44%)	210 (56%)	0 (0%)	0 (0%)	0 (0%)

Fertig Arbeitsplatz

# Ergebnisdarstellung II

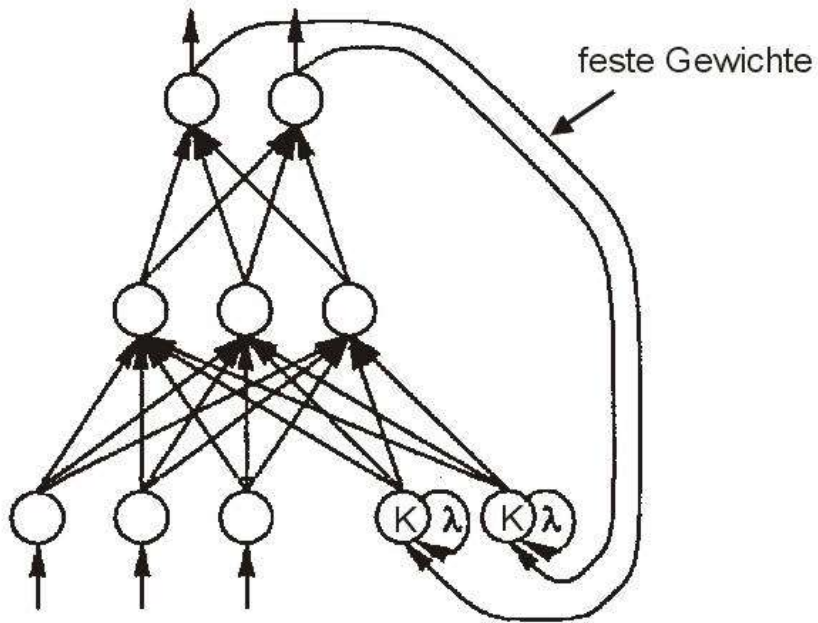


# Time Delay Netze



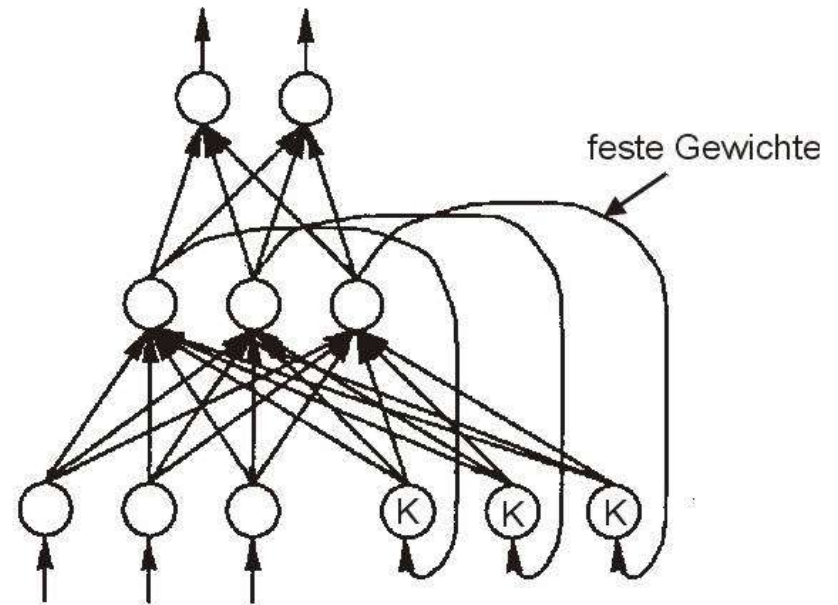


# Rekurrente Netze



**Jordan - Netz**

Quelle: A. Zell, Simulation Neuronaler Netze



**Elman - Netz**

(K) : Kontextzellen

( $\lambda$ ) [0...1], Rückkopplungsfaktor

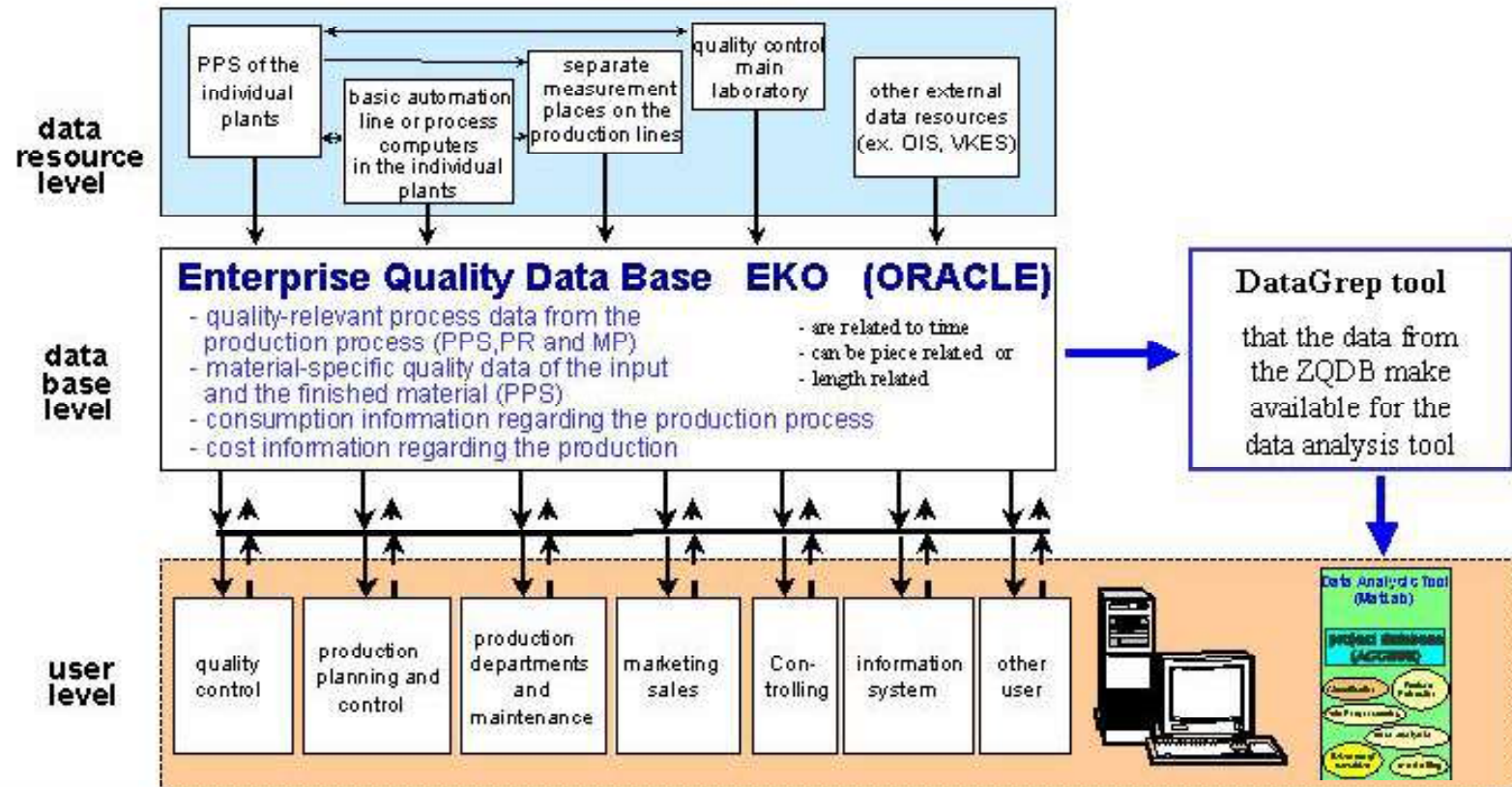
# Gliederung

---

1. Einleitung
  1. Motivation
  2. Was ist *Data Mining*
  3. Was versteht man unter *Big Data*
  4. Erstes Anwendungsbeispiel von *Data Mining*
2. Notwendige Verarbeitungsschritte des *Data Mining*
  1. Einführung
  2. Grafische Datenanalyse
  3. Datenvorverarbeitung
  4. Datenaggregation und Merkmalsberechnung
  5. Behandlung der Datenstichprobe
3. Einige grundlegende Methoden
  1. Clusteranalyse
  2. Künstliche neuronale Netze / Deep Learning Verfahren
4. Einfache statistische Abhängigkeitsanalyse
5. Komplexere Methoden der Abhängigkeitsanalyse
  1. Diskriminanzanalyse
  2. Entscheidungsbaumverfahren / Random Forests
  3. Self Organising Map
  4. Genetische Algorithmen
6. Methoden der datenbasierten Modellbildung am Beispiel der Klassifikation
  1. Grundlagen der Klassifikation
  2. Statistische Methoden
  3. Neuronale Methoden
- 7. Anwendungsbeispiele**



# System bei EKO Stahl



- - -> data base query  
 ———> result report

PPS - production planning and control system (operational data registration)  
 PR - process- and equipment computer; MP - separate measurement places  
 OIS - surface inspection system for hot rolled strip (by Parsitec)  
 VKES - data system for technological parameter of hot rolling mill

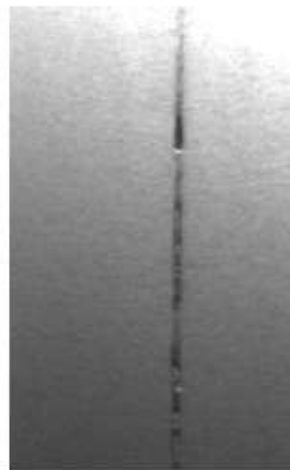
# Anwendungsbeispiel „Schalen“



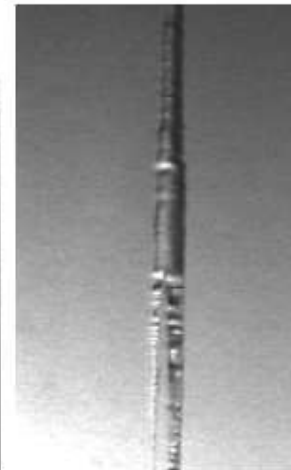
open edge shell



closed edge shell



middle shell of groove



partial open middle shell



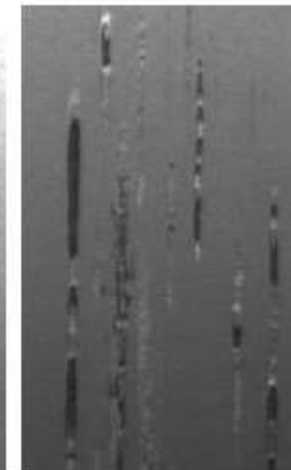
partial detached  
open middle shell



closed heavy  
middle shell

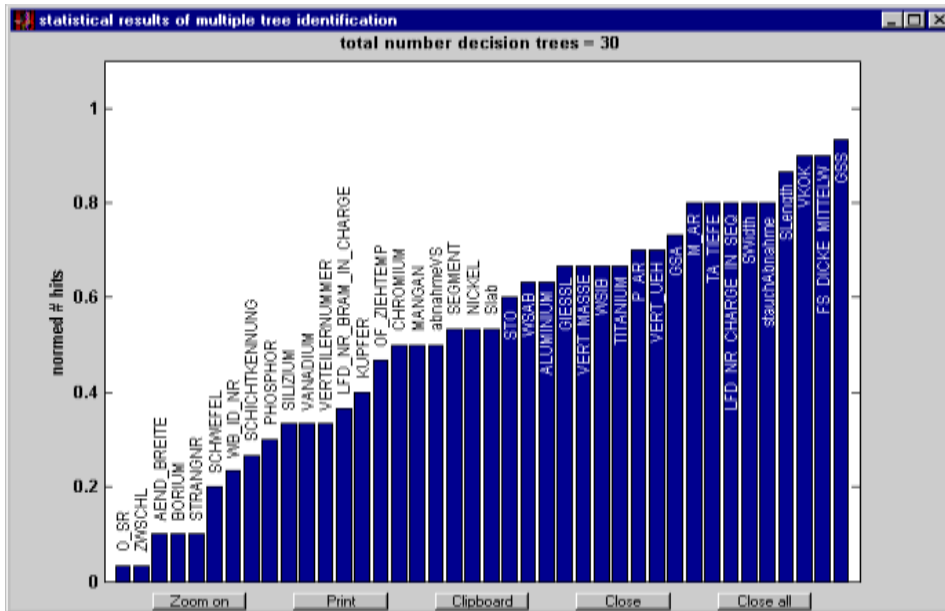


partial open  
very heavy



shell accumulation,  
heavy middle shell

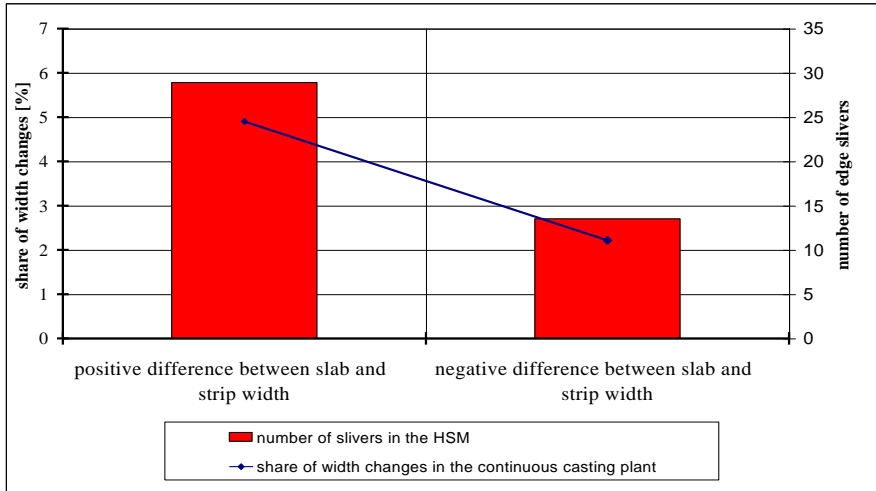
# Ergebnisse



- \* Mould level deviations
- \* Argon pressure and quantity
- \* Immersion depth of the nozzle during the sequence
- \* Tundish weight
- \* Sheet thickness in the finishing mill
- \* Modification of the casting width
- \* Edging draft

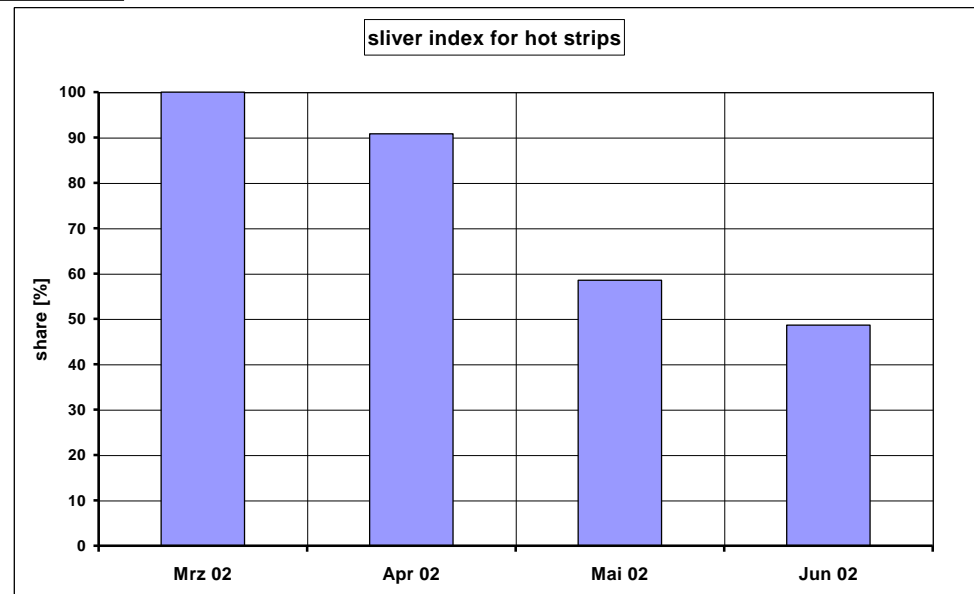
5 of 29 variables selected			
	Wilks lambda	Partial Lambda	F-Statistic
GSS	0.969	0.986	22.914
GISSL	0.963	0.991	14.121
M_AR	0.963	0.992	12.960
FS_DICKE_MITTELW	0.962	0.993	11.099
SCHICHTKENNUNG	0.961	0.994	10.481

# Nutzung der Ergebnisse

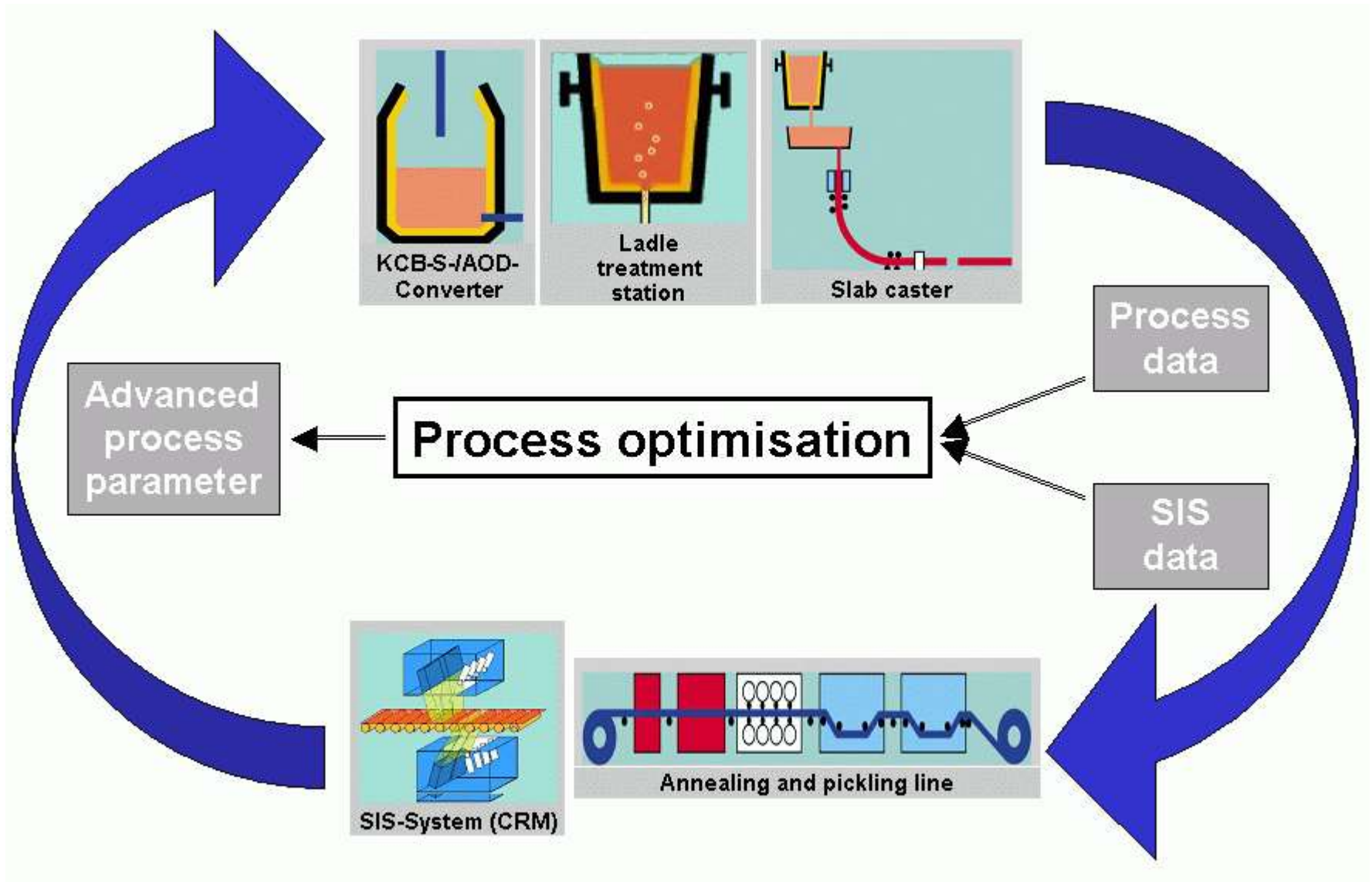


Influence of change of casting width on number of shells (electrical grades)

Reduction of argon quantity, pressure and immersion depth of nozzle on the number of shells on IF grades



# Anwendungsbeispiel „offene/geschlossene Zeilen“



# Ergebnisse

Numerical data correlation of  
process and inspection data  
for steel grade ASTM type 304



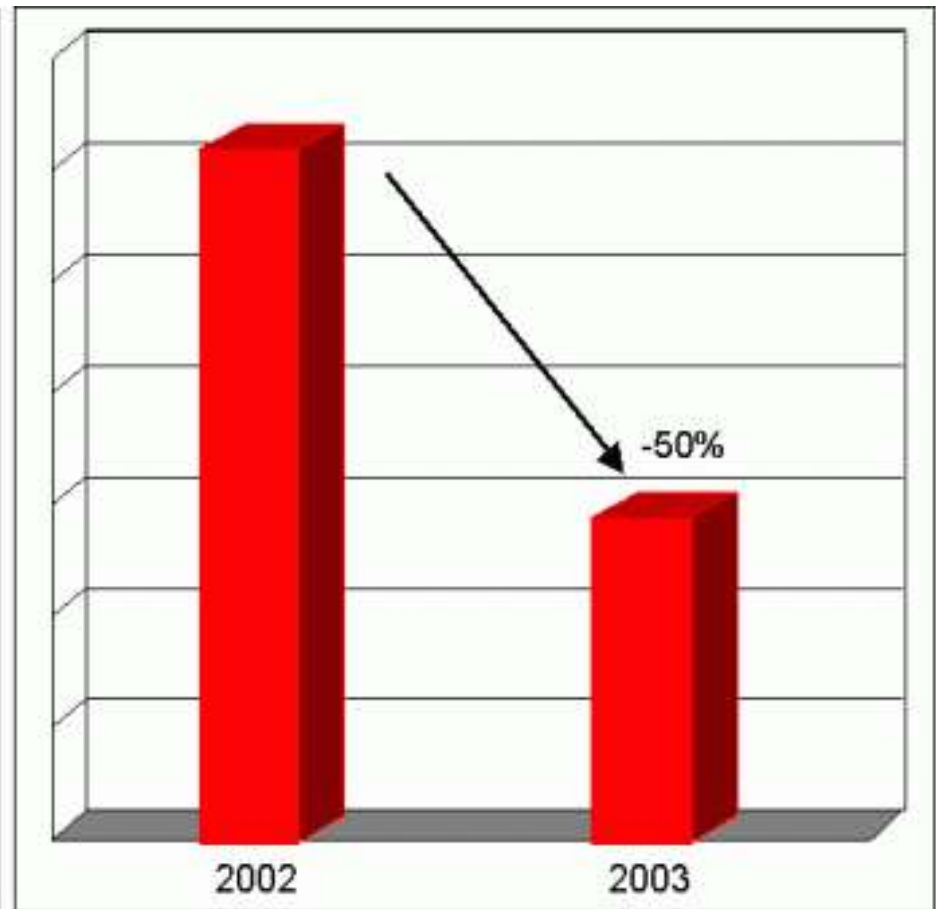
Determination of process  
parameters with influence on the  
formation of inclusion lines



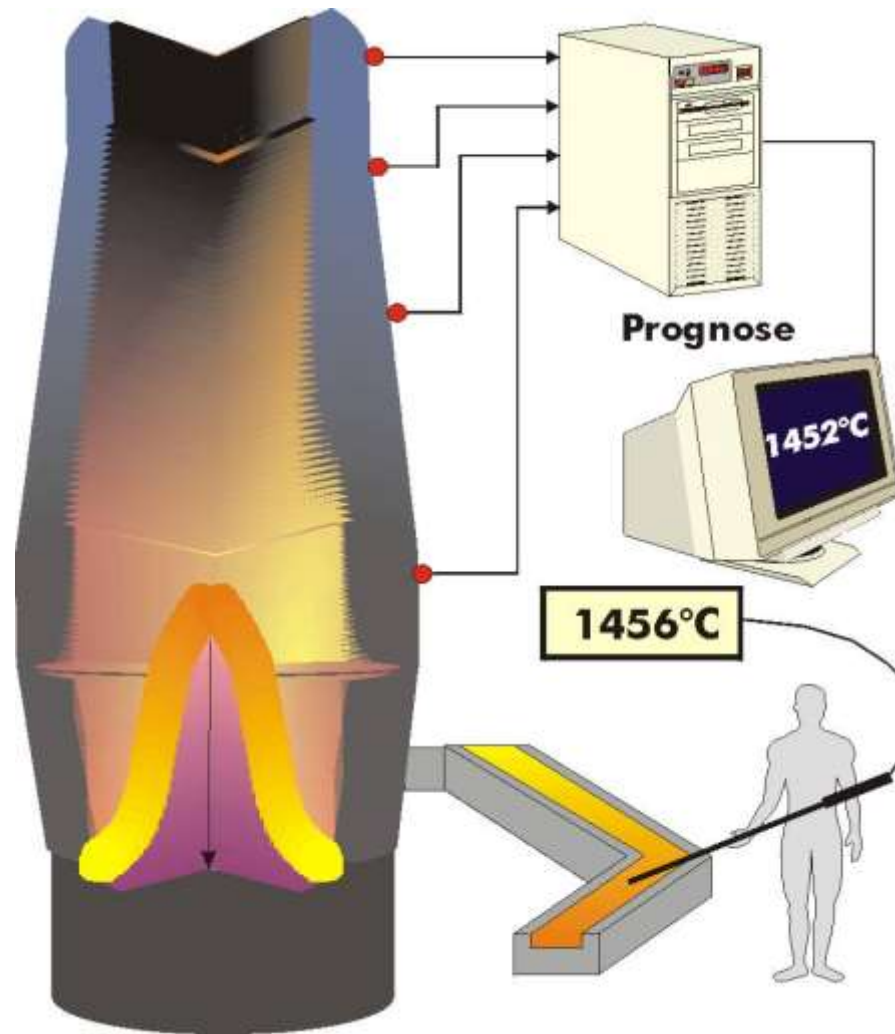
Calculation of advanced  
process parameters



**Reduction of inclusion line  
defects by  $\approx 50\%$**

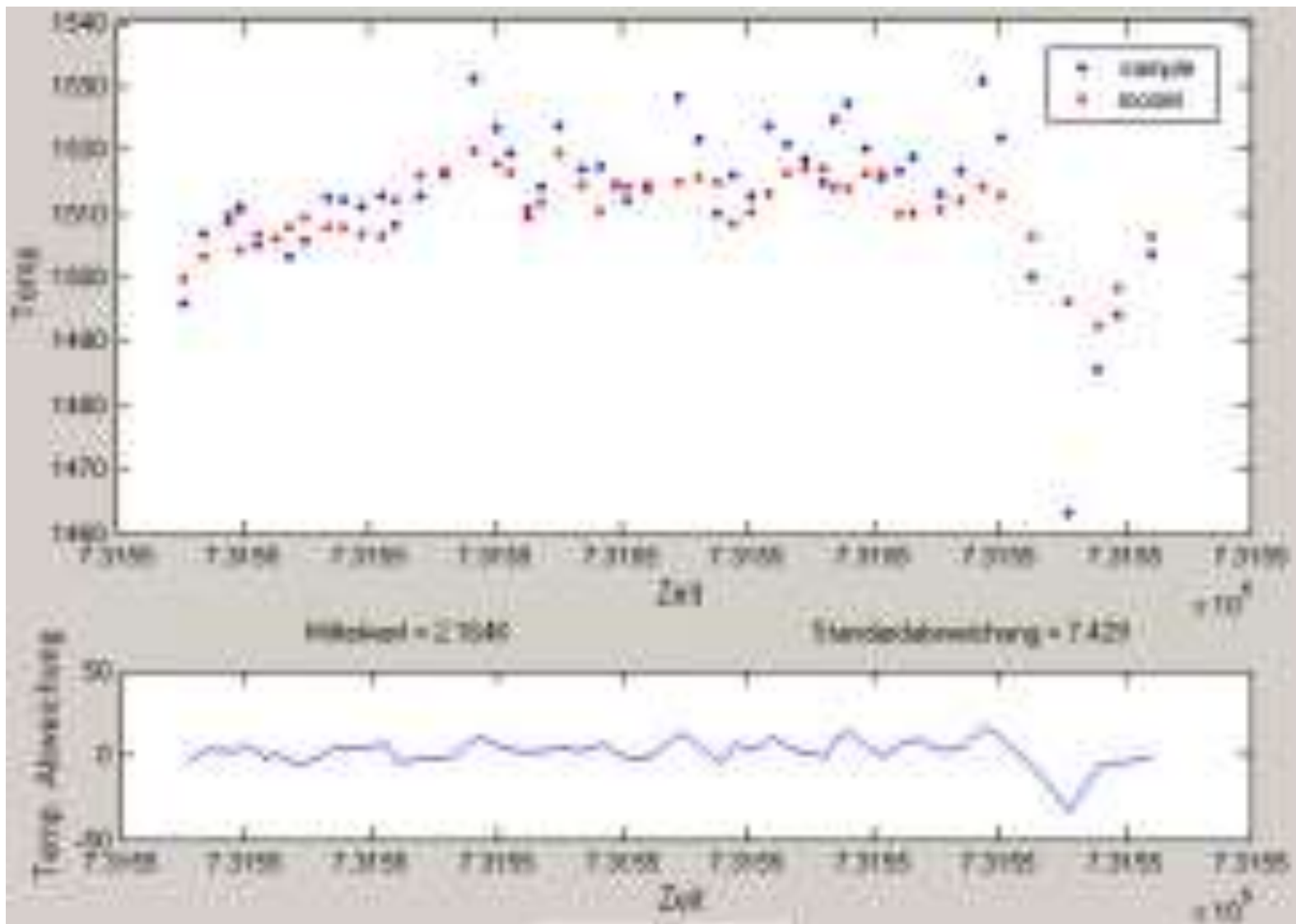


# Anwendungsbeispiel „Prognose Abstichtemperatur“

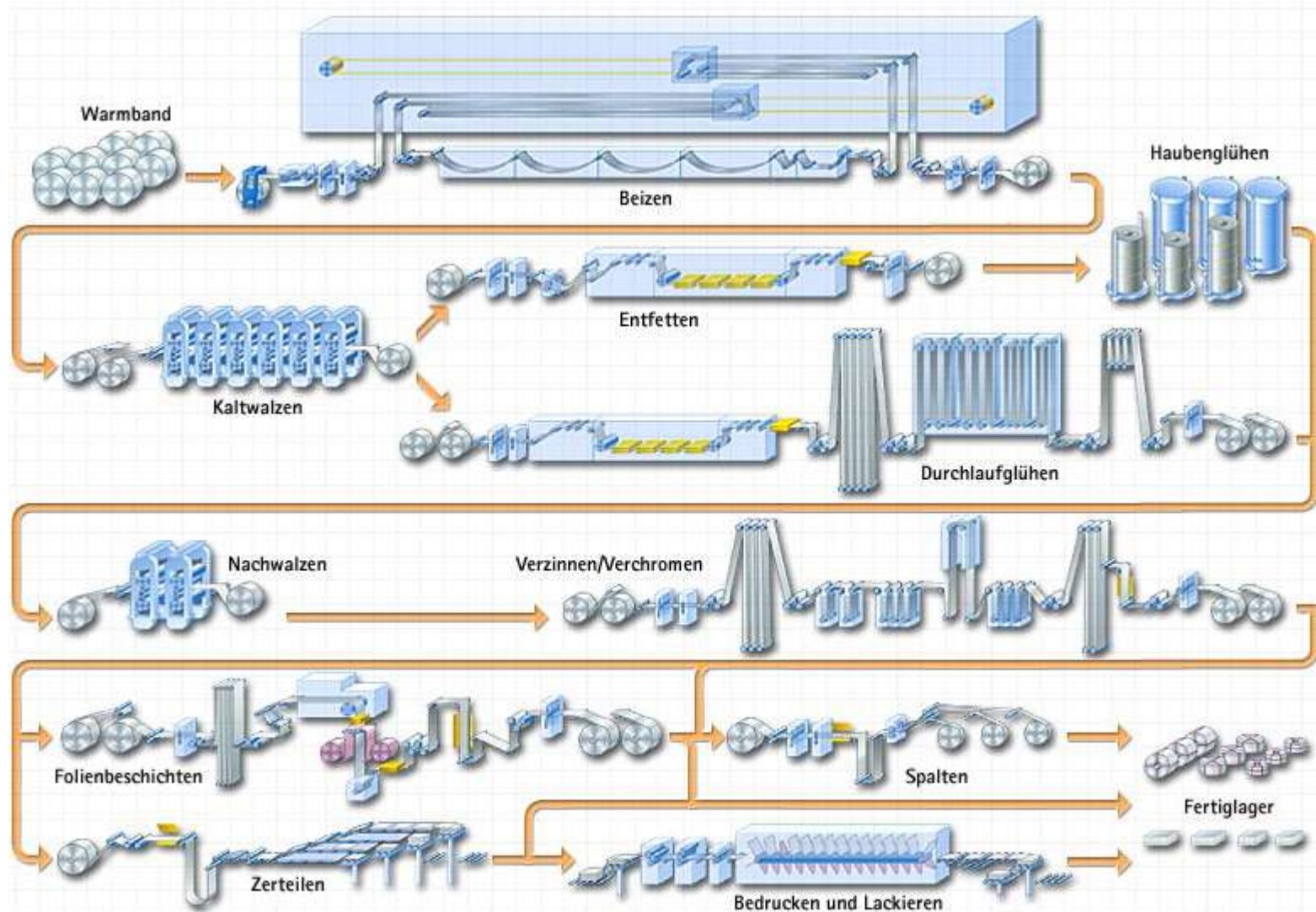




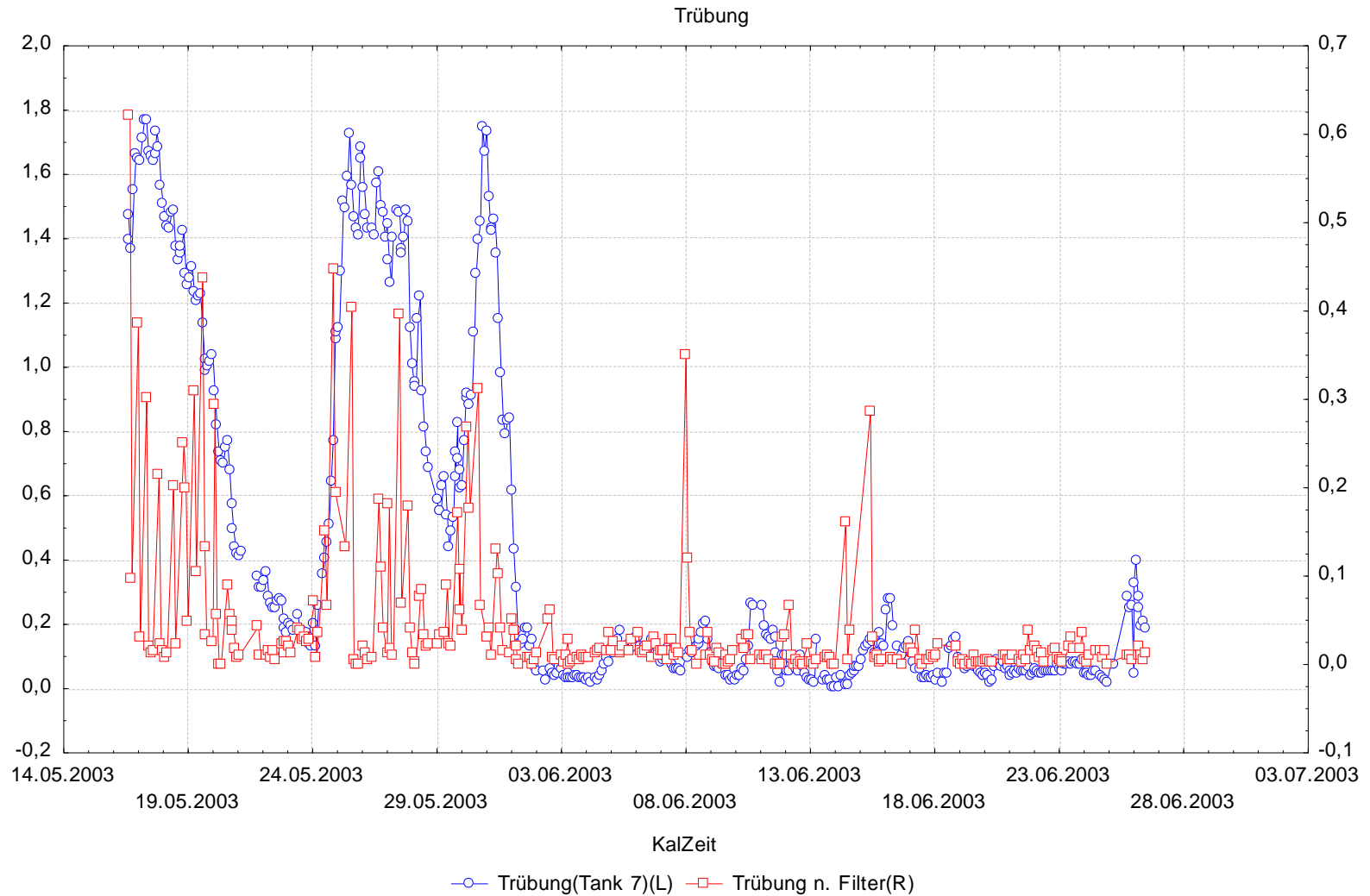
# Ergebnisse



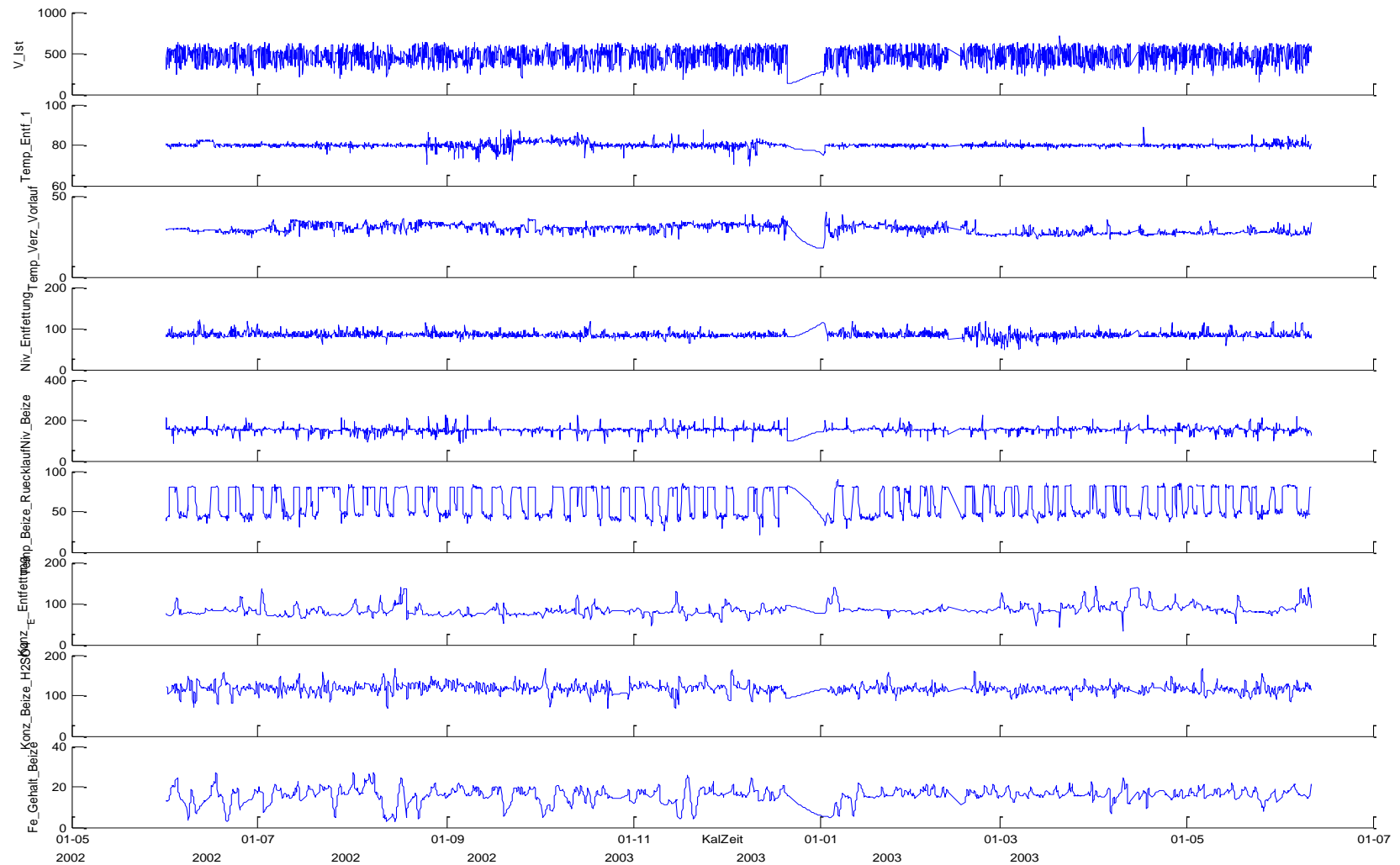
# Anwendungsbeispiel „Trübung“



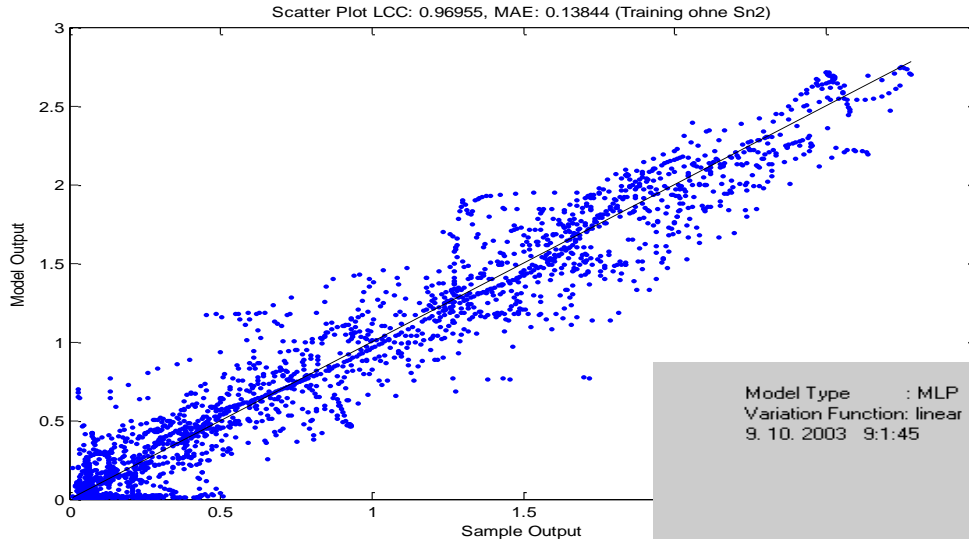
# Verlauf der Trübung



# Verlauf der Einflussgrößen

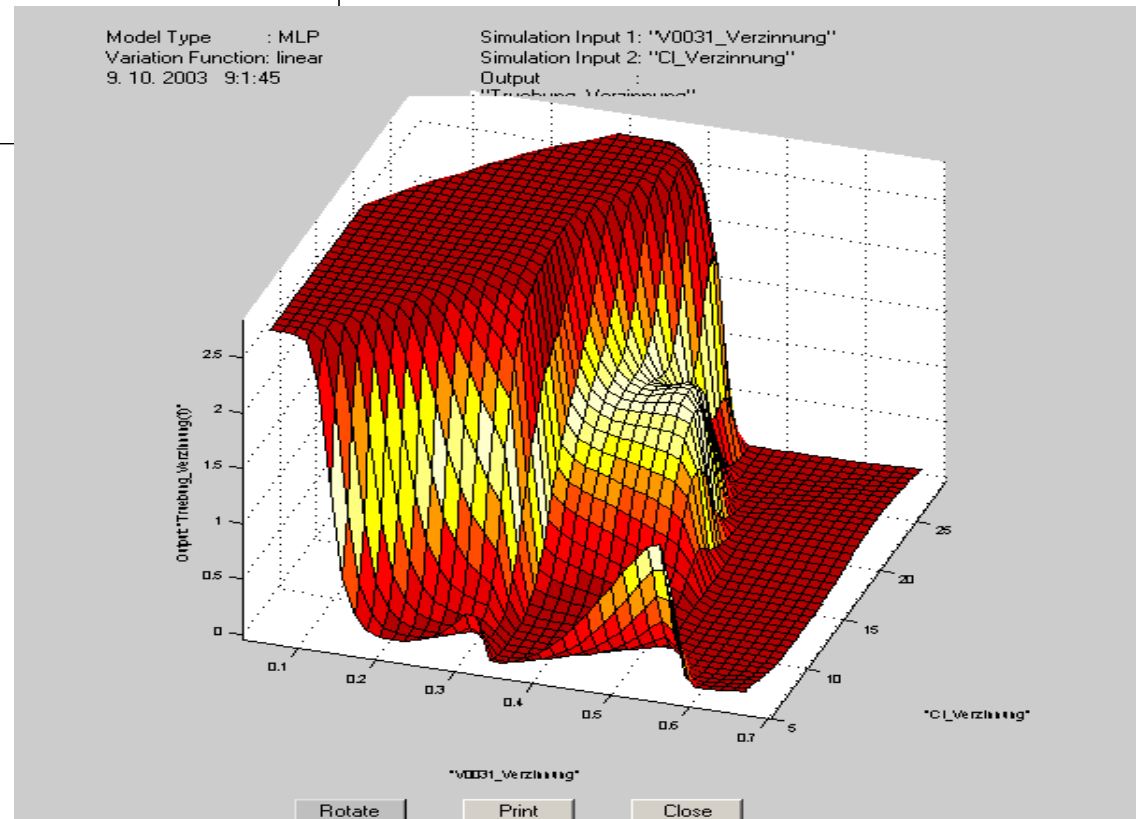


# Ergebnisse



- Bestimmtheitsmaß
  - 94 % für das Training
  - 92 % für das Testen
  - 93 % für die Validierung

- Topologie des neuronalen Netzes
  - 7 Eingangsknoten
  - 1 „hidden layer“ mit 10 Knoten
  - 1 Ausgangsknoten
- Trainingsmenge 80 %
- Test und Validierung je 10 %



## Fast zum Schluss ...

---

Was haben wir alles nicht besprochen ...

- Clusteranalyseverfahren (ISODATA-Algor., Fuzzy-C-Mean, Sub-Clustering, ...)
- Klassifikationsmethoden (Bayes-Netze, LVQ-Verfahren, SVM, Neuro-Fuzzy-Methoden, ... )
- Regressionsmethoden (AR, ARX, ARMAX, Wavelet-Netze, CMAC-Netze, Neuro-Fuzzy-Methoden, ...)
- ....

Statistikpakete, Data Mining Pakete (meist spezialisiert auf den Bereich des Finanzwesens, etc.), Methodenpakete (Fuzzy oder NN oder ...)

Beispiele:

- Statistica
- Cognos
- SAS Enterprise Miner
- Cornerstone
- DataEngine
- Clementine
- Delta Miner
- Darwin
- WEKA (Java-Bibliothek)
- Rapid Miner



# System im Praktikum: *DataDiagnose*

The screenshot displays the DataDiagnose software interface. The main window is titled "DataDiagnose" and features a menu bar with options: Datei, Ansicht, Stichprobe, Vorbereitung, Analyse, Modellierung, ControlPerf, Einstellung, and Hilfe. Below the menu bar is a toolbar with various icons for file operations, analysis, and visualization.

The interface is divided into several sections:

- Workbook:** Located on the left, it contains a tree view under the "Analyse" folder. The tree structure includes:
  - Grafisch
    - Y(index) (02/15/05 14:19:)
    - Linienplot (02/15/05 14:19:)
    - Scatter (02/15/05 14:19:)
  - Statistisch
    - Korrelation (02/15/05 14:20:)
    - Histogramm (02/15/05 14:20:)
  - Beschreibende Statis (02/15/05 14:20:)
  - DataMining
    - Diskriminanzanalyse
      - (02/15/05 14:21:)
      - (02/15/05 14:21:)
    - Entscheidungsbaum (02/15/05 14:21:)
    - Multiple Bäume (02/15/05 14:22:)

- Datengruppe Variablenwahl Parameter:** The central area for variable selection. It contains two lists:
- verfügbare Variablen:** A list of variables including grade, MoldNo\_1 through MoldNo\_6, Molduse1 through Molduse6, CastRate, MidLev\_ave, MidLev\_std, MidLev\_osc, Speed\_ave, no\_in\_sec, wait\_time, E\_IntP1\_ave, E\_IntP1\_std, Mo, Sn, Ti, P, S, Ni, Ca, and waterflow.
- ausgewählte Variablen:** A list of selected variables including TundTemp, SuperHeat, C, Mn, Si, Speed\_std (highlighted), WTmpDfave, WTmpDfst, WTmpDfosc, StPos\_ave, StPos\_std, StPos\_osc, CoolW1ave, CoolW1std, CoolW2ave, CoolW2std, CoolW3ave, CoolW3std, Al, and Cu.Between the lists are three buttons: a right arrow (>), a left arrow (<), and a double arrow (<>).
- Ausgangs-/ Zielgröße:** A text field containing the value "QP".

At the bottom of the interface, there is a status bar with the text "Auswahl Datenbank:" and a message box that reads "Datenbankfile wurde ausgewählt: D:\DemoDaten\sid2.mdb".

# Das war's ....

---

Do, 26.1.2017: Übungen am Computer anhand einiger realer Daten

# Untersucher Prozess

