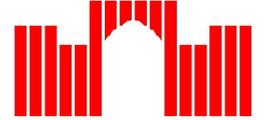
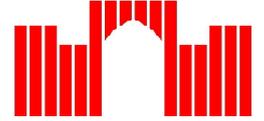


Klassifizieren und Visualisieren von Daten mit Selbstorganisierenden Karten

Diplomkolloquium
Sven Schröder



- Aufgabenstellung und Motivation
- Biologisches Vorbild
- Modell Kohonen-Netz
- Lernalgorithmus
- Visualisierung
- SOMARFF
- Zusammenfassung

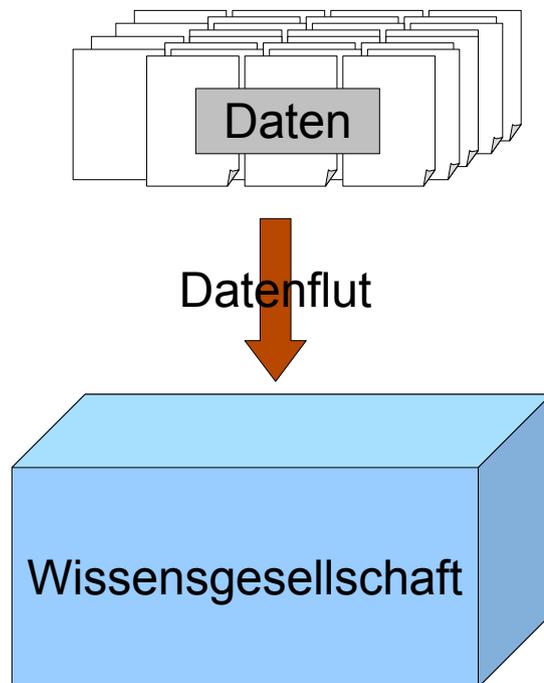


Aufgabenstellung

- Theorie Selbstorganisierende Karten
- Untersuchen von Visualisierungsmöglichkeiten
- Prototypische Umsetzung in Programmsystem



Motivation



- Ständige Datenerfassung
- Größter Teil der Daten wird nicht analysiert und verarbeitet
- Gründe:
 - Zeitdruck
 - Datenkomplexität
 - Datenmenge

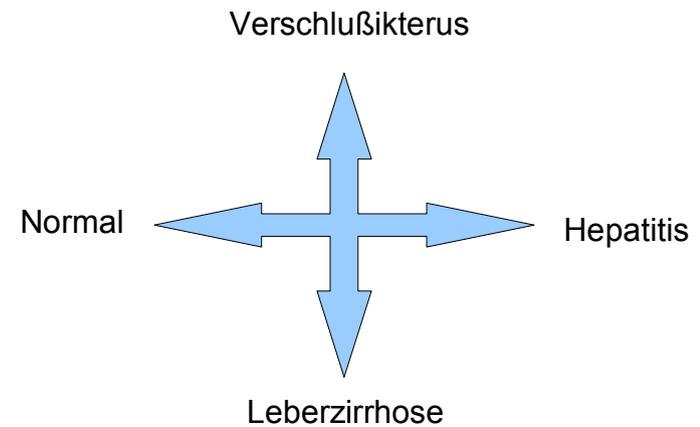
Bedarf an rechnergestützten Verfahren!

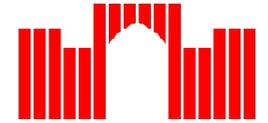


Beispiel

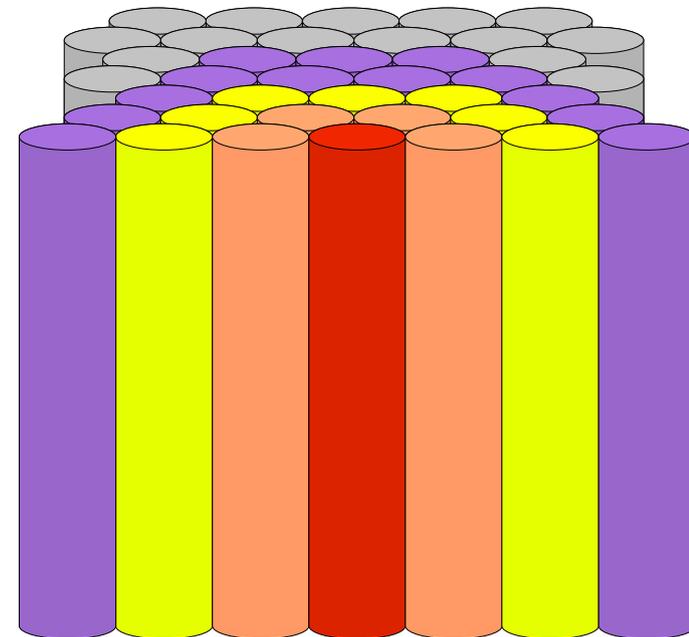
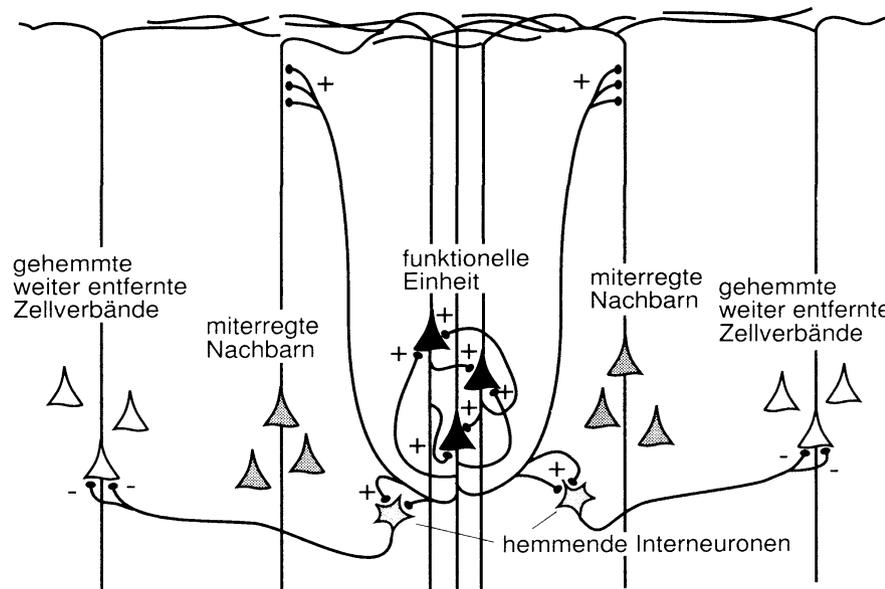
Patient-Nr.	AP (U/l)	FE (mg/l)
1	4,0	1,0
2	3,0	1,7
3	2,6	1,8
4	1,5	0,7
5	2,5	2,2
6	1,1	1,0
7	2,8	3,1
8	1,7	3,2
9	0,8	0,5
10	2,1	3,0
11	2,0	0,7
12	1,2	0,5
13	4,5	0,7
14	2,5	3,0
15	3,5	0,7
16	2,2	3,2
17	2,1	3,5
18	2,1	2,0
19	3,5	1,2
20	3,2	1,2

- Krankenhauslabor
- 20 Patienten
- Alkalische Phosphate
- Eisengehalt





Biologisches Vorbild



- Neuronen in funktioneller Einheit (Säule) erregen sich stark
- Benachbarte Säulen werden in Abhängigkeit der Entfernung miterregt
- Entfernte Säulen werden über Interneurone gehemmt

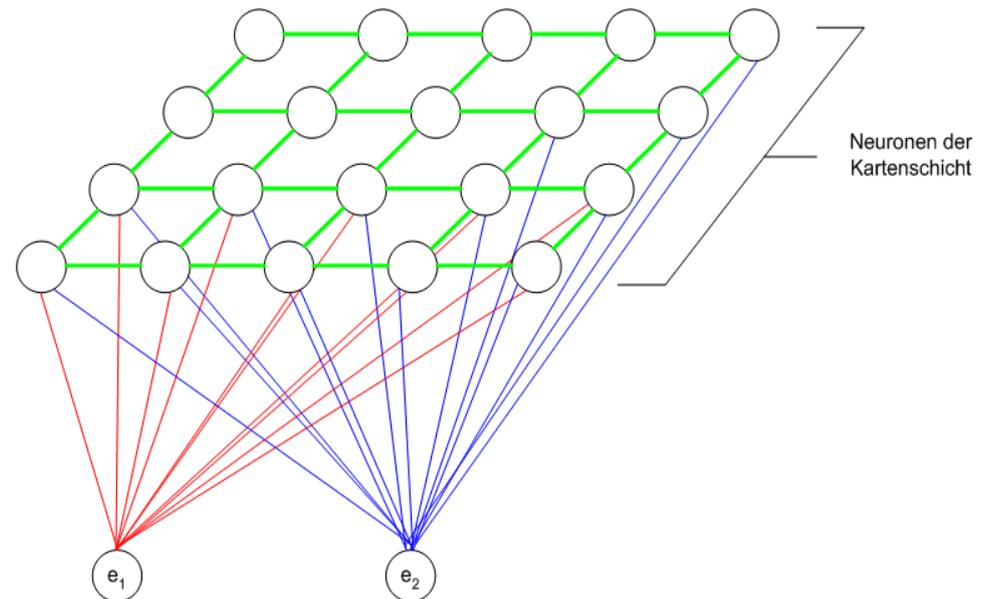
Laterale Inhibition



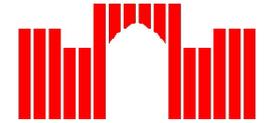
Modell Kohonen-Netz



Teuvo Kohonen(Bild von: www.cis.hut.fi)



- Erfinder ist Teuvo Kohonen (Finnland)
- Dem Aufbau des Kortex nachempfunden
- Mehrdimensionale Daten werden auf 1-, 2- oder dimensionalen Raum abgebildet
- Wettbewerbslernen
- Unüberwacht
- Auch SOM genannt



Modell Kohonen-Netz

Kartenschicht:

- Sollte möglichst viele Neuronen enthalten
- In der Regel 2-dimensional, aber auch 1 und 3 dimensional möglich
- Quadratische, hexagonale und andere Strukturen möglich
- Topologie kann planar oder auch toroid sein
- Lernalgorithmus berücksichtigt keine Rückkopplungen zwischen den Neuronen

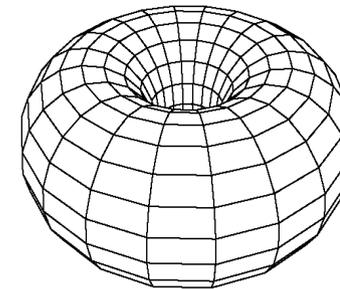
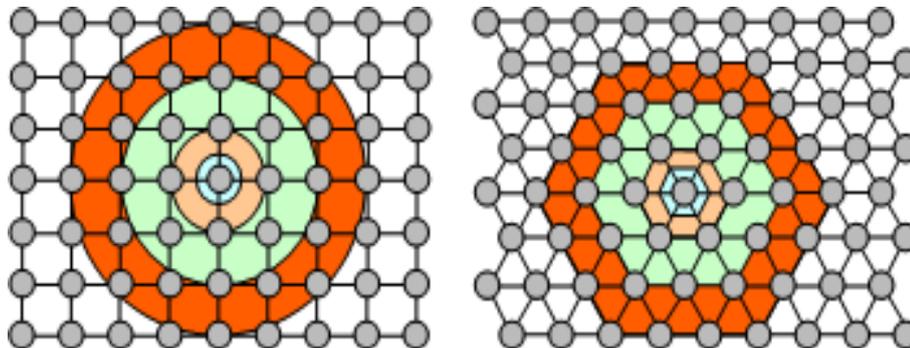


Abbildung aus [Ult06]

Eingabeschicht:

- Stellt Schnittstelle zur „Außenwelt“ dar
- Anzahl der Attribute entscheidet über die Menge der Eingabeneuronen
- Eingabeneuronen mit Kartenneuronen vollvernetzt und gewichtet



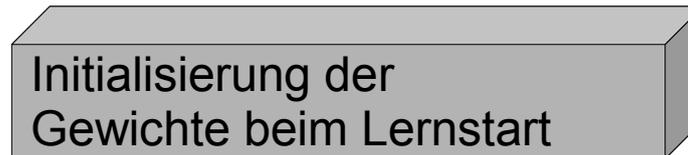
Lernalgorithmus

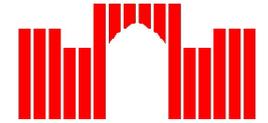
Prinzip: Wettbewerbslernen



Kartenneuronen treten untereinander in Konkurrenz!

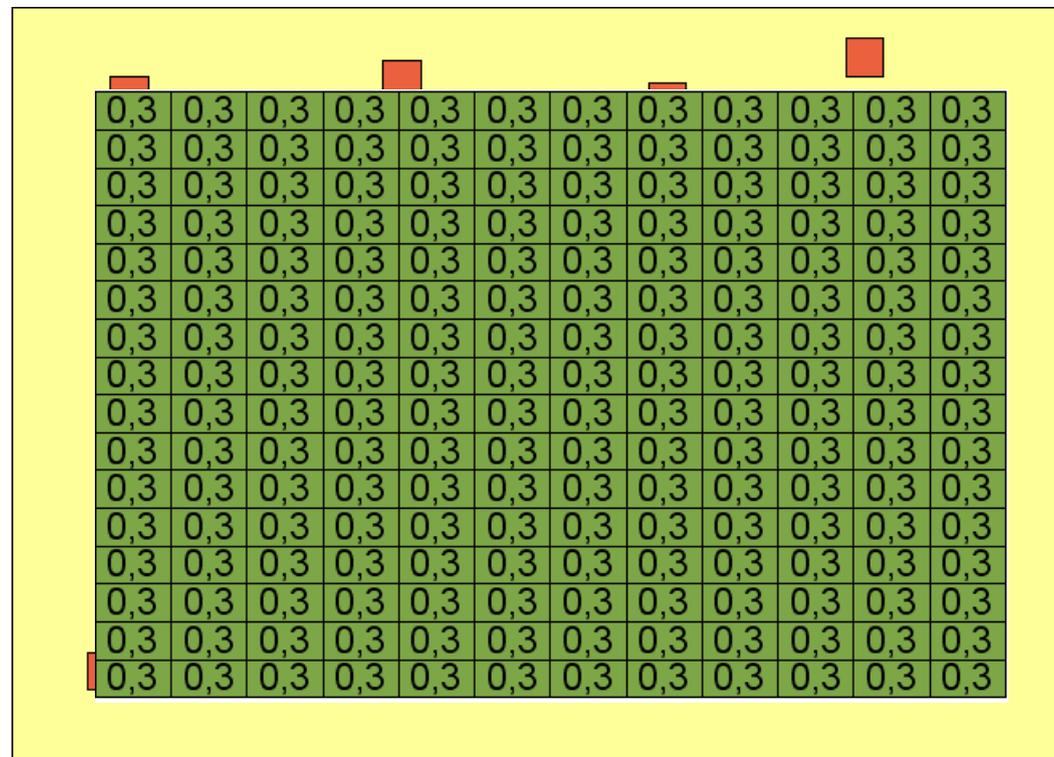
Lernvorgang:





Initialisierung

- Zufällige Verteilung von kleinen Gewichten
- Nach der zu Lösenden Aufgabe (z.B. beim Problem des Handlungsreisenden)
- Einheitlicher fester Wert





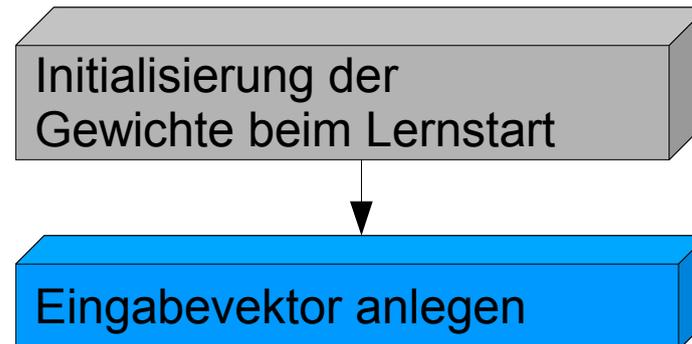
Lernalgorithmus

Prinzip: Wettbewerbslernen



Kartenneuronen treten untereinander in Konkurrenz!

Lernvorgang:





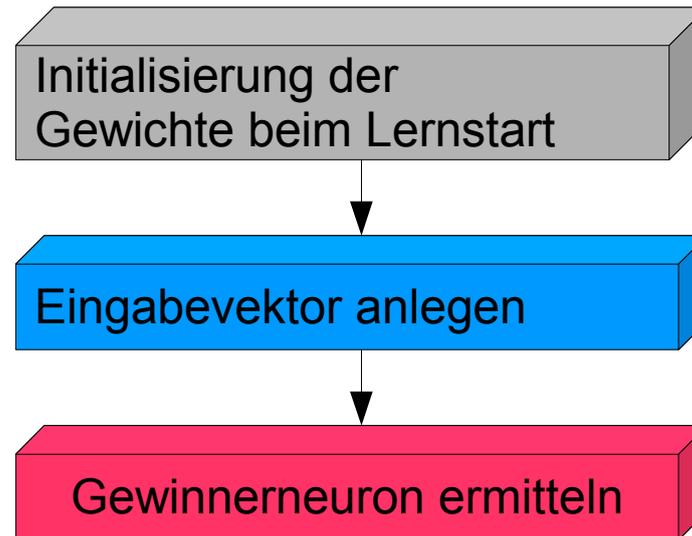
Lernalgorithmus

Prinzip: Wettbewerbslernen



Kartenneuronen treten untereinander in Konkurrenz!

Lernvorgang:

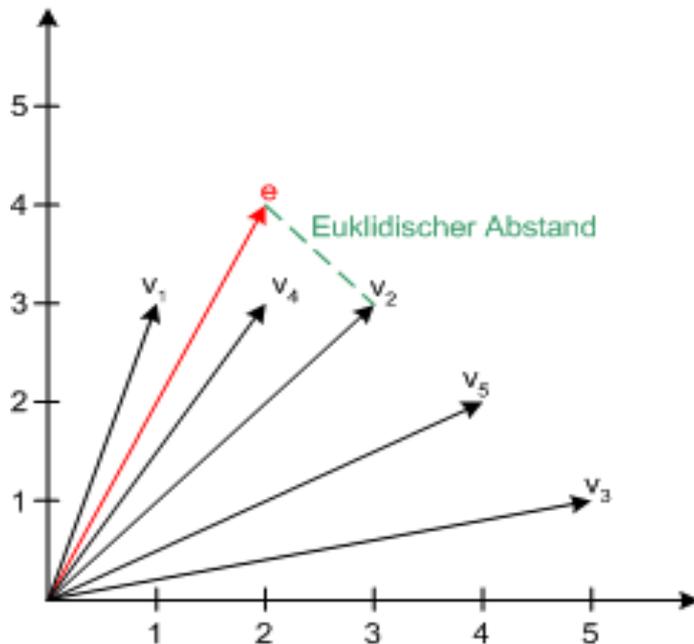




Ermittlung des Siegerneurons

Siegerneuron = Neuron, dessen Gewichtsvektor dem Eingabevektor am ähnlichsten ist!

- Verfahren:
- Ermittlung über die Euklidische Distanz (minimal)
 - Ermittlung über das Skalarprodukt (maximal)



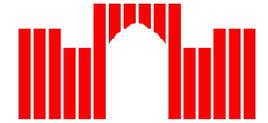
Euklidische Distanz

d = Eukl. Distanz
x = Eingabevektor
y = Gewichtsvektor
n = Anzahl Eingabeneuronen
i = Gewichtindex des Neurons

$$d(x, y) = \|x - y\|$$

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



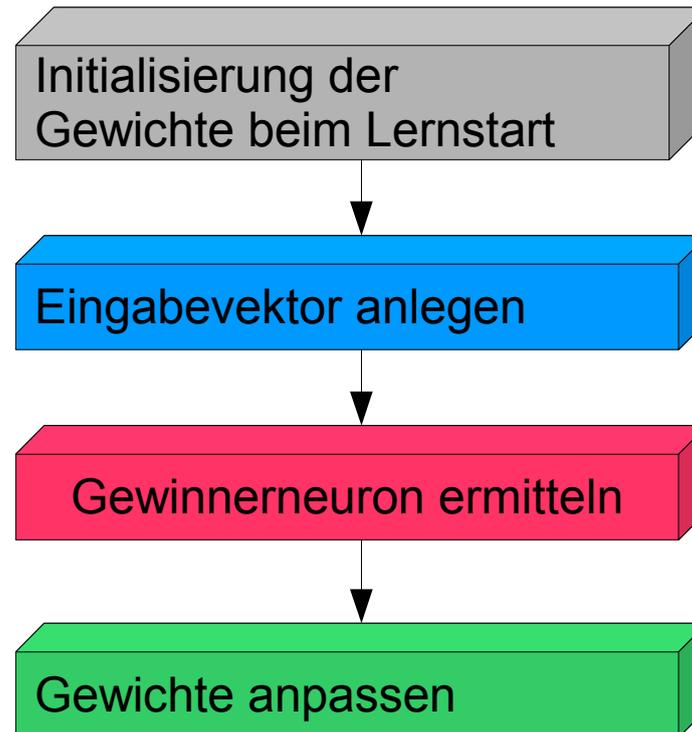
Lernalgorithmus

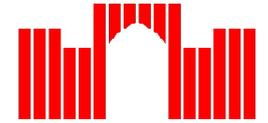
Prinzip: Wettbewerbslernen



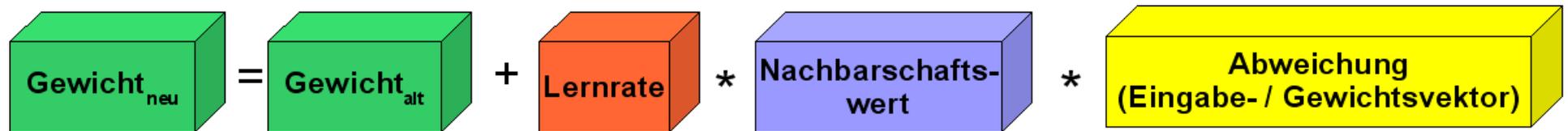
Kartenneuronen treten untereinander in Konkurrenz!

Lernvorgang:

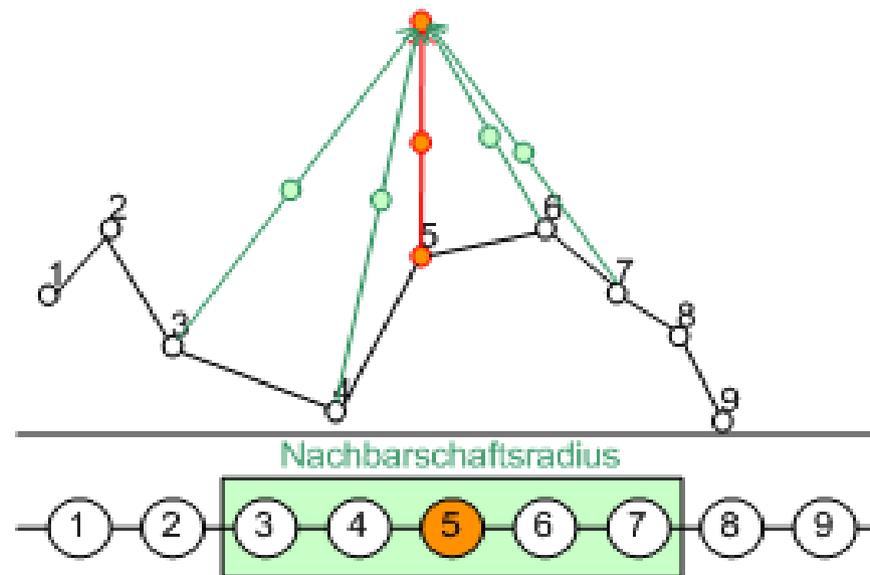




Anpassen der Gewichte



$$W_j(t+1) = W_j(t) + \eta \cdot h_{cj} [X(t) - W_j(t)]$$





Anpassen der Gewichte

Lernrate

- Verantwortlich für die Größe der Gewichtsanzpassung
- Dadurch Steuerung der Lerngeschwindigkeit
- Wert zwischen 0 und 1

Lernrate = 1

Änderung sehr groß, Gewichte passen sich im Erregungsmaximum vollständig an den letzten Eingabevektor an, vorher Gelerntes wird „vergessen“

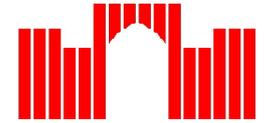
Lernrate = $0 < x < 1$

Bereits Gelerntes wird behalten, Neues fließt nur zu einem Bruchteil in die Anpassung mit ein. Zu kleine Werte verzögern den Lernfortschritt.

Lernrate = 0

Es findet keine Anpassung mehr statt.

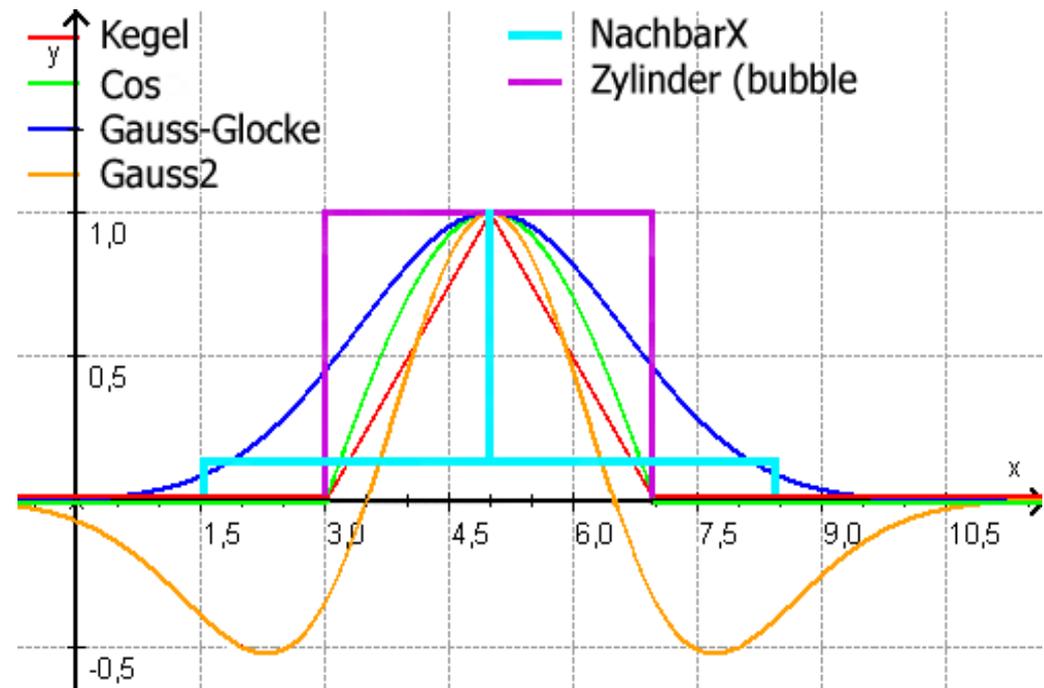
Wird im Laufe des Trainings über mathematische Funktionen reduziert.
In der Regel mit monoton fallender Funktion.



Anpassen der Gewichte

Nachbarfunktion

- Verantwortlich für Ausbreitungsbereich und Höhe der Gewichtsanzpassung
- Siegerneuron = 100%, also 1
- Nachbarneuronen werden durch mathematische Funktion miterregt (teilangepasst)
- Gängige Nachbarfunktionen:





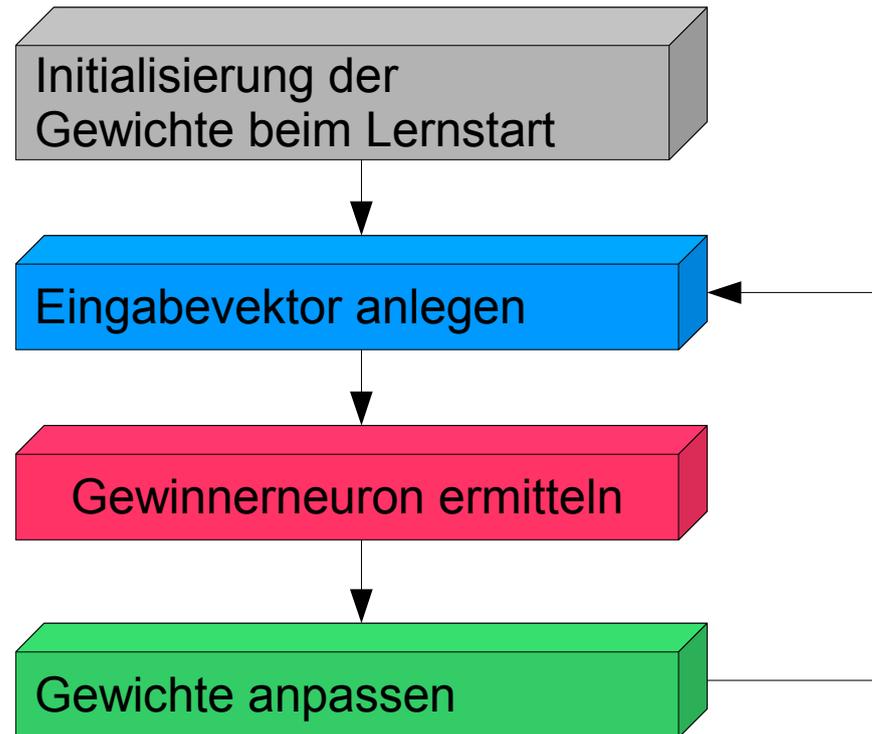
Lernalgorithmus

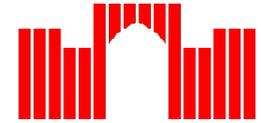
Prinzip: Wettbewerbslernen



Kartenneuronen treten untereinander in Konkurrenz!

Lernvorgang:





Visualisierung

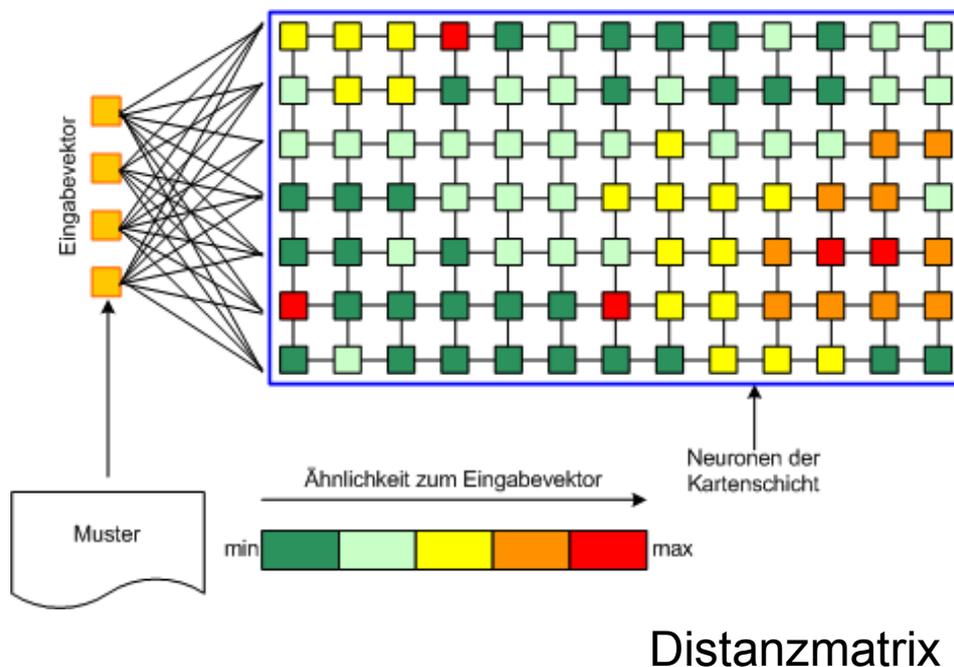
- Ziel: Komplexitätsreduktion
- Abbildung von Daten aus n-dimensionalem Datenraum in 2-dimensionalen Ausgaberaum
- Ermöglicht Zusammenhänge in den Daten besser und schneller zu erkennen
- Reduktion des Platzproblems
- SOM visualisiert die Daten nach den Prinzipien: **Ähnlichkeit** und **Häufigkeit**
- Unterscheidung in musterabhängige und musterunabhängige Visualisierungen

Musterabhängig	Musterunabhängig
Distanzmatrix	Gewichtsmatrix
Gewinnermatrix	Komponentenmatrix
	U-Matrix
	Intervertierte U-Matrix
	P-Matrix
	U*-Matrix
	Histogramm realtiver Gewinnerverteilung
	Verbindungs-Distanz-Matrix



Visualisierung

musterabhängig



Stellt die Aktivierungen der Neuronen bei Anlegen eines speziellen Musters dar.

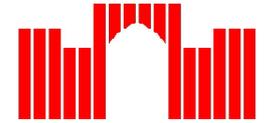
Dies kann durch Farb-, Helligkeits- und/oder 3D-Oberfläche erfolgen.

Netzeingabe:

$$net_j = net_e + net_o - BIAS$$

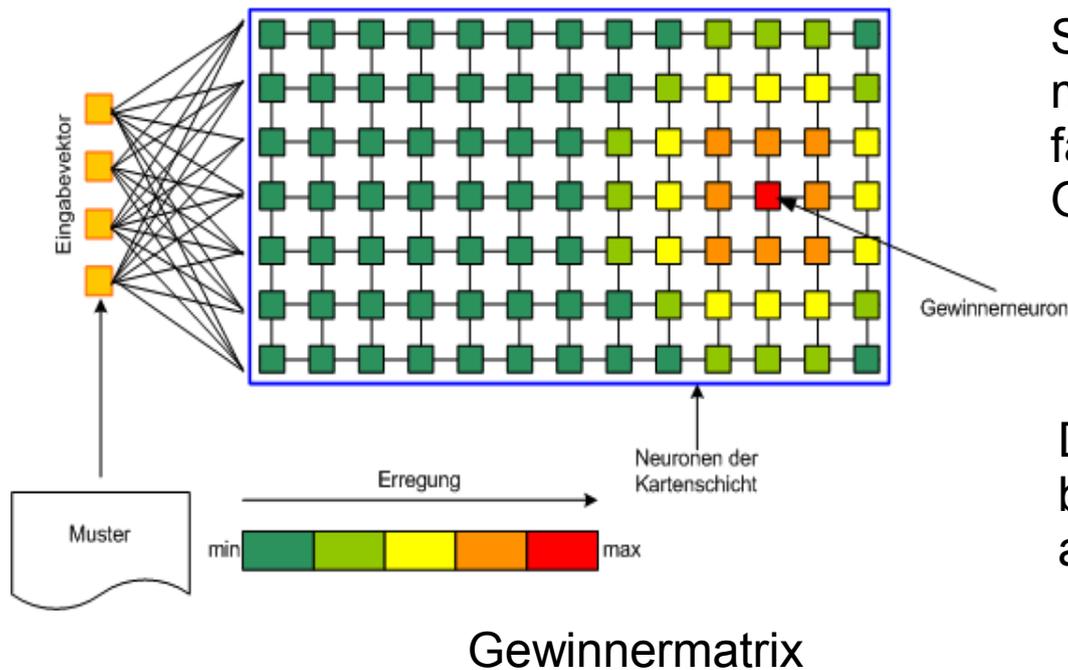
$$net_j = \sum_{i=0}^n e_i \cdot w_{ij} + \sum_{k=0}^n o_k \cdot w_{kj}$$

Aktivierung:
$$act_j = \frac{1}{1 + e^{-net_j}}$$



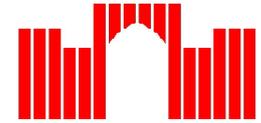
Visualisierung

musterabhängig



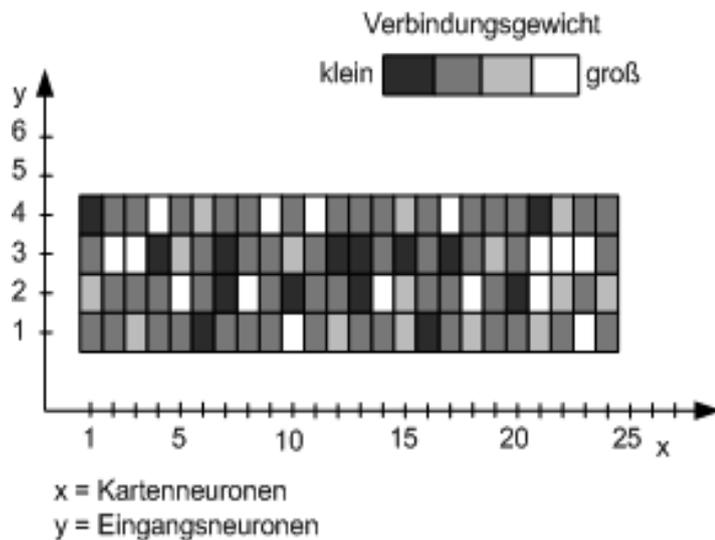
Stellt das Gewinnerneuron und die miterregten Nachbarneuronen durch farbliche Abstufung und/oder 3D Oberfläche dar.

Die Ausgabe ist stark von der benutzten Nachbarschaftsfunktion abhängig.



Visualisierung

musterunabhängig



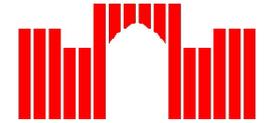
Gewichtsmatrix

Stellt ALLE Verbindungsgewichte zwischen den Kartenneuronen und den Eingabeneuronen dar.

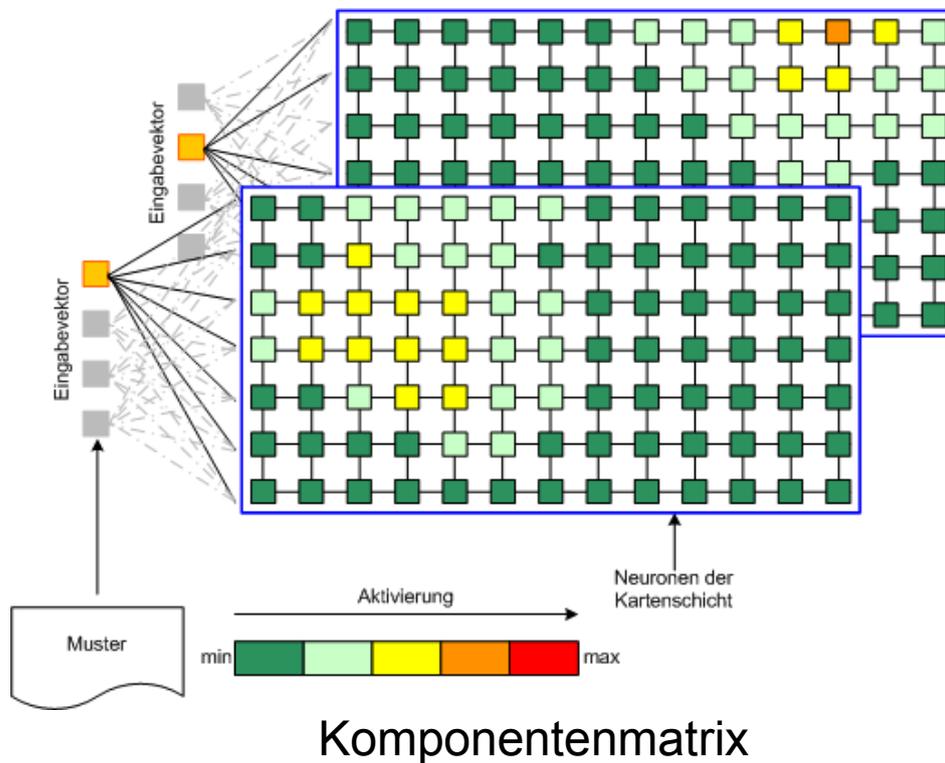
2 Achsen:

- Kartenneuronen
- Eingabeneuronen

Für Trainingsüberwachung nützlich, da sich Änderungen an den Gewichten leicht verfolgen lassen

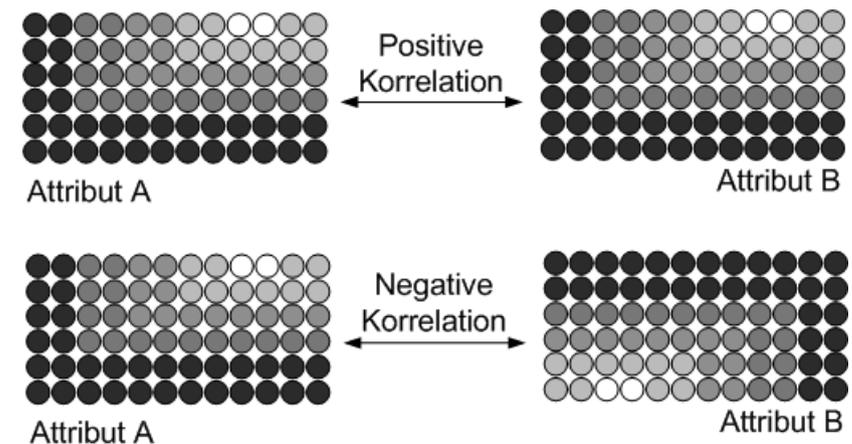


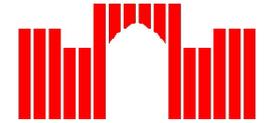
Visualisierung musterunabhängig



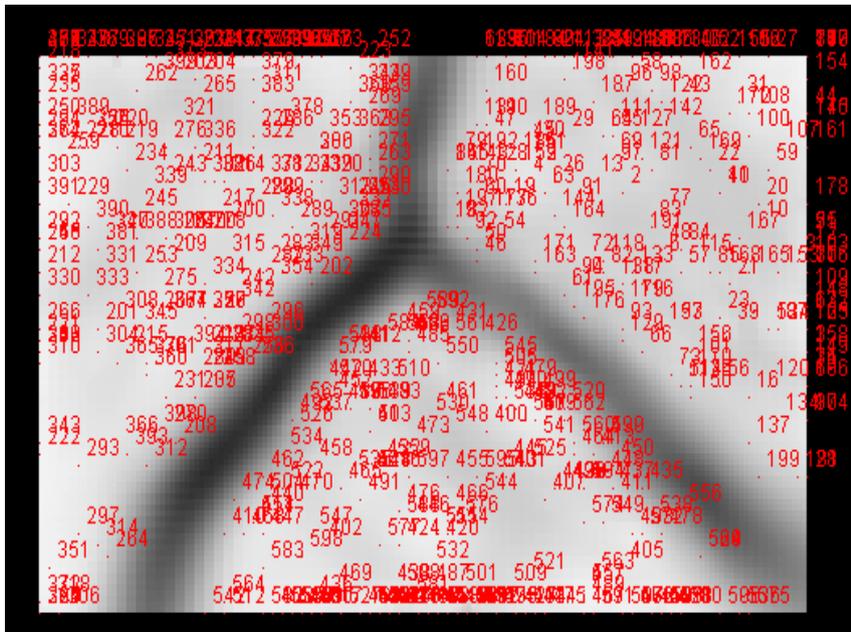
Ähneln der Gewichtsmatrix, jedoch werden nur die Gewichte für ein bestimmtes Eingabeneuron dargestellt.

Zum Auffinden von Korrelationen geeignet.

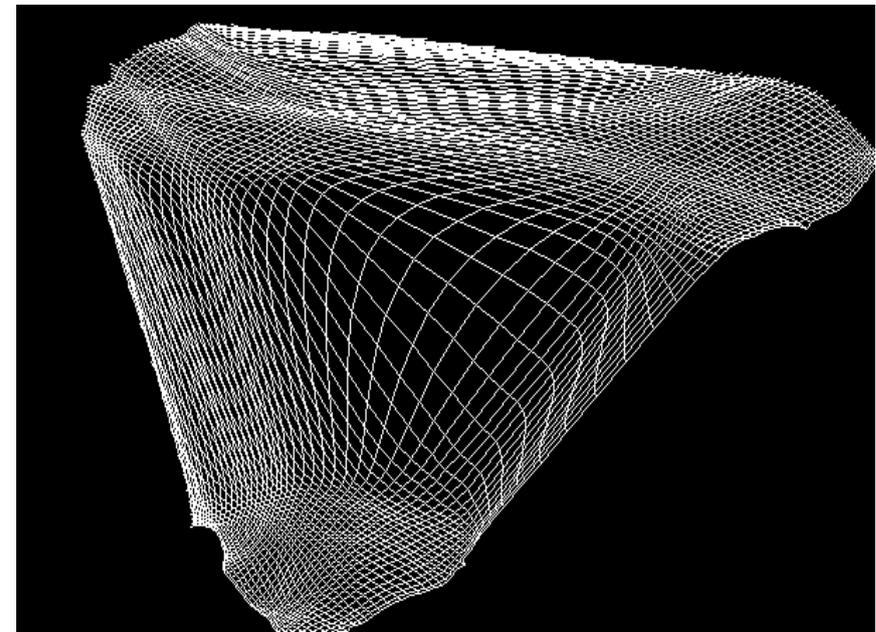




Visualisierung musterunabhängig



U-Matrix (aus SOMARFF)



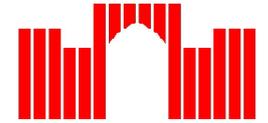
Gittermatrix (aus SOMARFF)

- Stellen die Entfernungen zwischen den Gewichtsvektoren der Kartenneuronen dar
- U-Matrix durch Farbe, Helligkeit oder 3D-Struktur
- Gittermatrix durch ein verzerrtes Netz
- Entfernung wird durch Euklidische Distanz berechnet



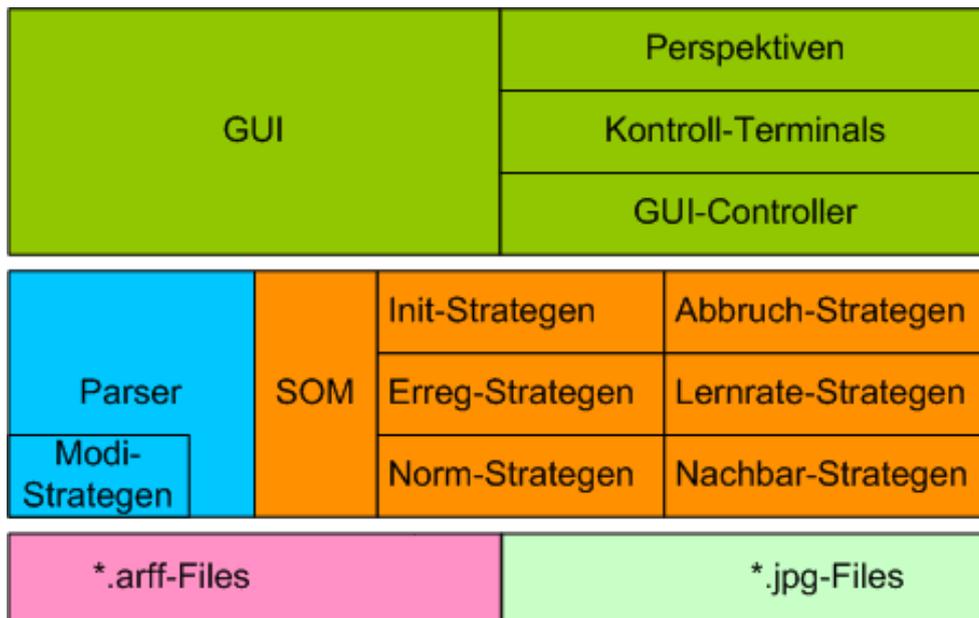
SOMARFF - Konzept

- Für Lehre Thema „Selbstorganisation“
- Soll einfach zu bedienen sein
- Datenquelle *.arff
- Datenanalyse durch ein Kohonen-Netz
- Einstellbare Parameter für Training, Netz-Topologie
- Mehrere Perspektiven
- Erweiterbarkeit, Wartbarkeit
- Soll ohne Installation auskommen

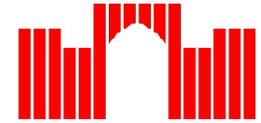


SOMARFF - Konzept

Architektur



- 3-Schichten-Architektur
- Entwurfsmuster:
 - ✓ Observer
 - ✓ Abstrakte Fabrik
 - ✓ Strategie
- Programmiersprache JAVA

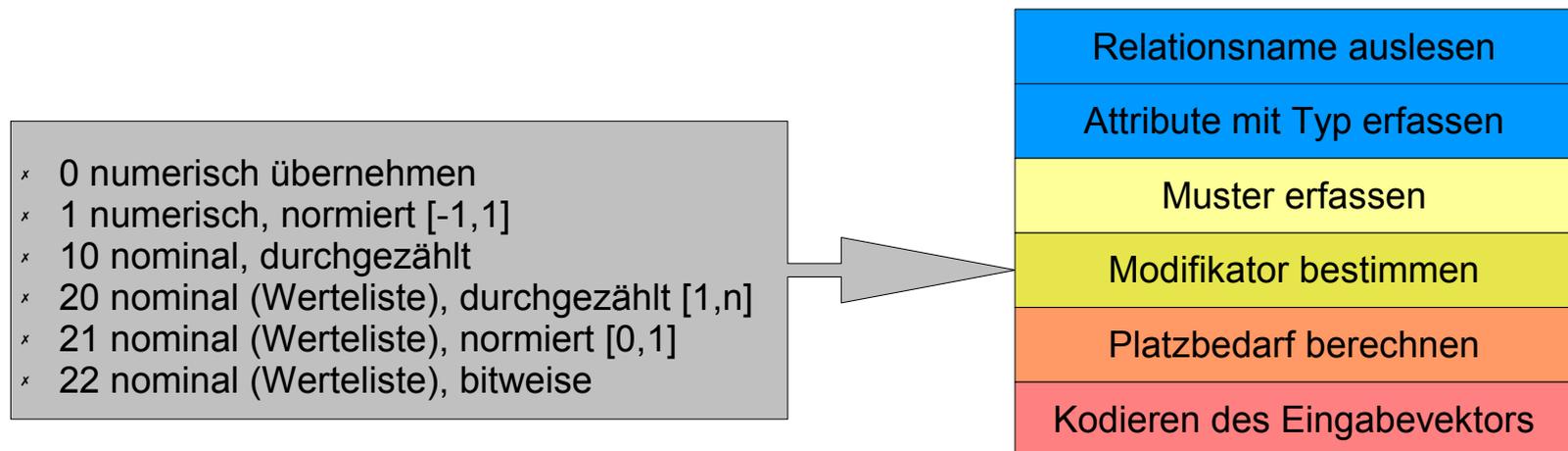


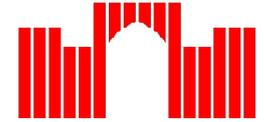
SOMARFF - Implementierung

Komponente: Parser



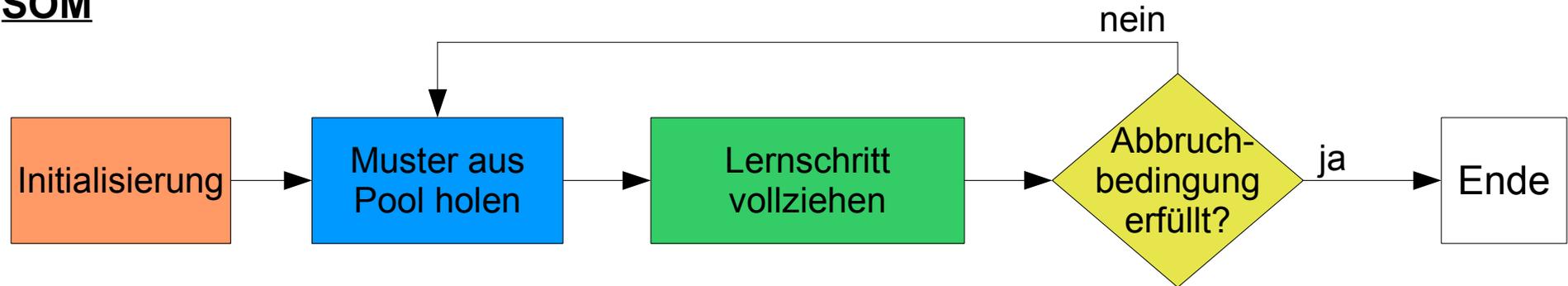
- Liest arff-Datei ein und erstellt entsprechende Muster für Netztraining
- Als Schnittstelle angelegt (durch Parser für andere Datenquellen austauschbar)
- Enthält einen Satz von Modifikatoren zur Anpassung an die Kodierregeln



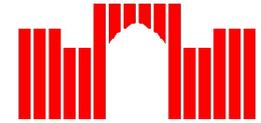


SOMARFF - Implementierung

SOM



- Verantwortlich für Netzinitialisierung und Netztraining
- Thread-basiert
- Benachrichtigt ihre Zuhörer (Sichten) bei Zustandsänderung
- Austauschbare Algorithmen für:
 - × Initialisierung
 - × Ermittlung Siegerneuron
 - × Reduktion der Lernrate
 - × Nachbarschaftsfunktion
 - × Abbruchbedingung
 - × Normalisierung



SOMARFF – Oberfläche 1 (Steuerung)

Parser-Einstellungen

numerische Werte: ordinale Werte: Wertauflistungen:

Wert übernehmen durchgezählt durchgezählt

Wert normieren durchgezählt und normiert

durchgezählt und normiert aufgeschlüsselt

Beschriftungsspalte:

ARFF-Terminal

Relation:body

Name	Geschlecht	Körpergröße	Gewicht
Anton	M	180	79
Berta	W	169	53
Charlie	M	160	65
David	M	175	71
Egon	M	179	82
Frieda	W	155	52
Gustav	M	189	102
Holger	M	185	100
Ina	W	182	75
Jörg	M	185	99

Lernparameter

Lernrate

Faktor

linear

exp

Nachbarfunktion

Ausdehnung

Reduktion

Abbruch Lernschritte

Fehler

Anzahl

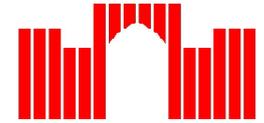
SOM-Parameter

Initialisierung

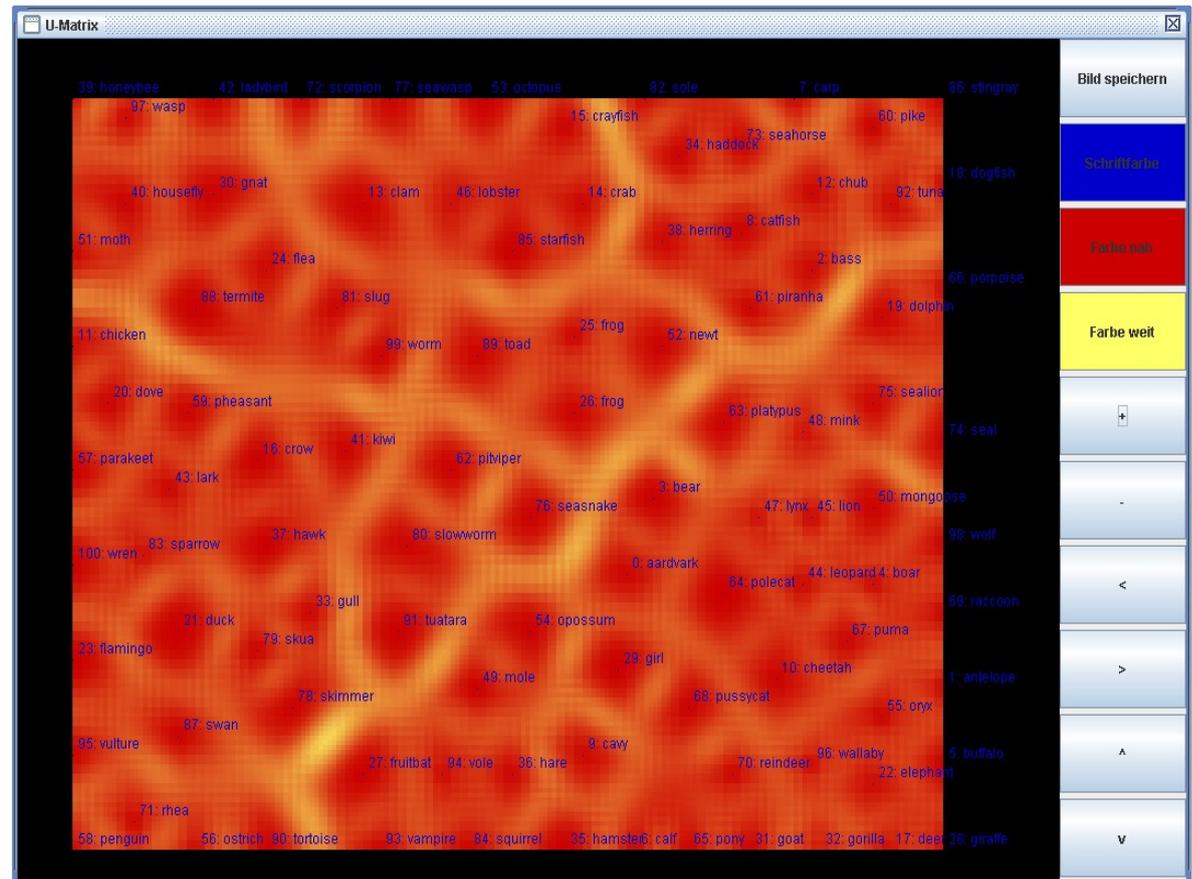
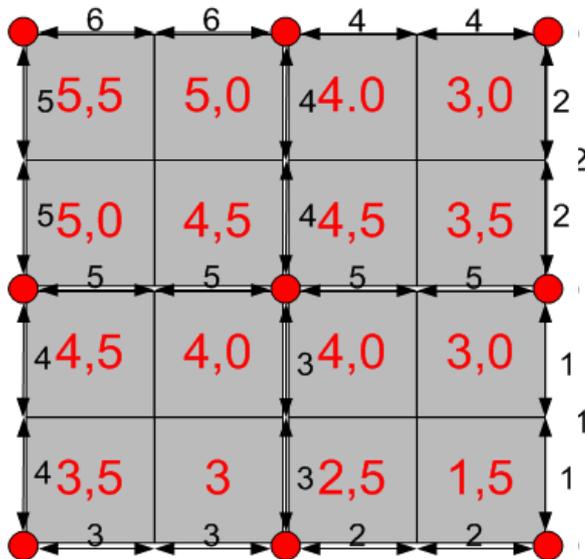
Wert

Spalten

Zeilen



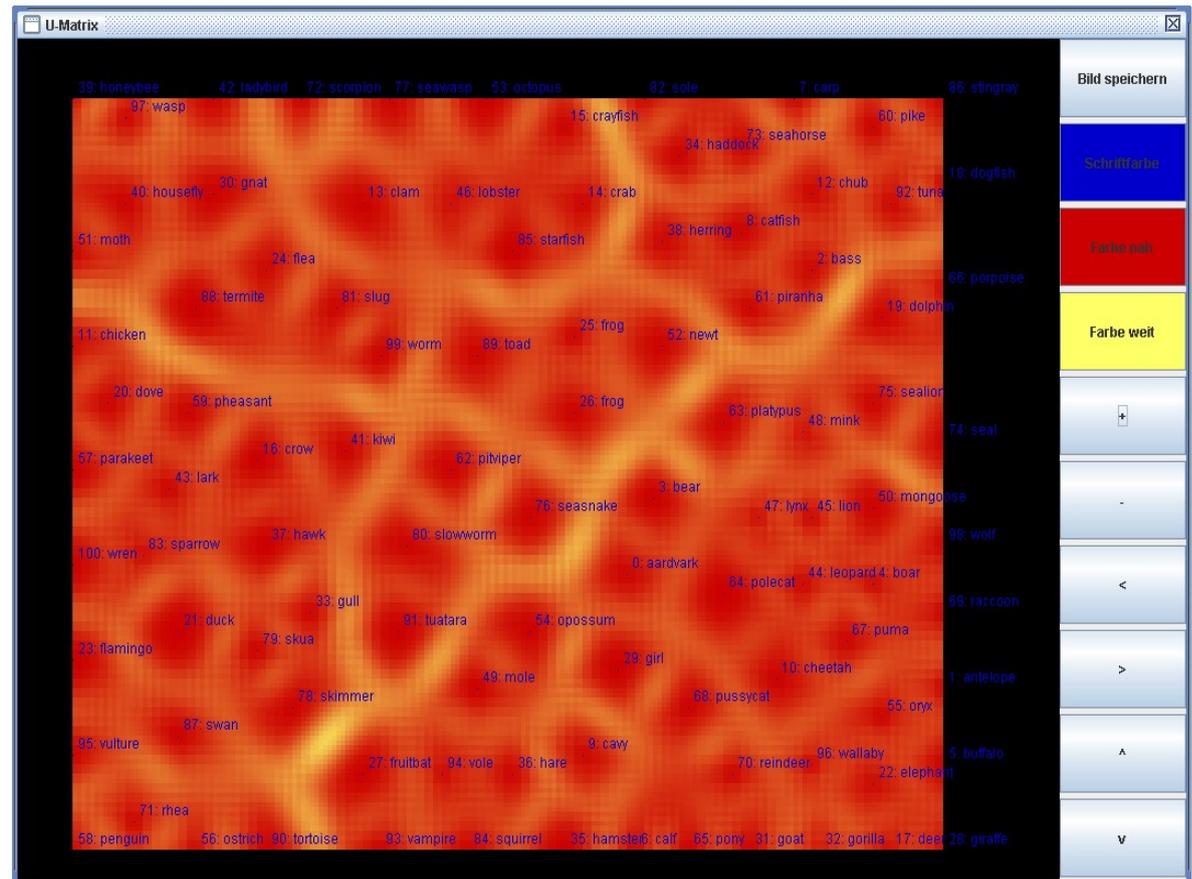
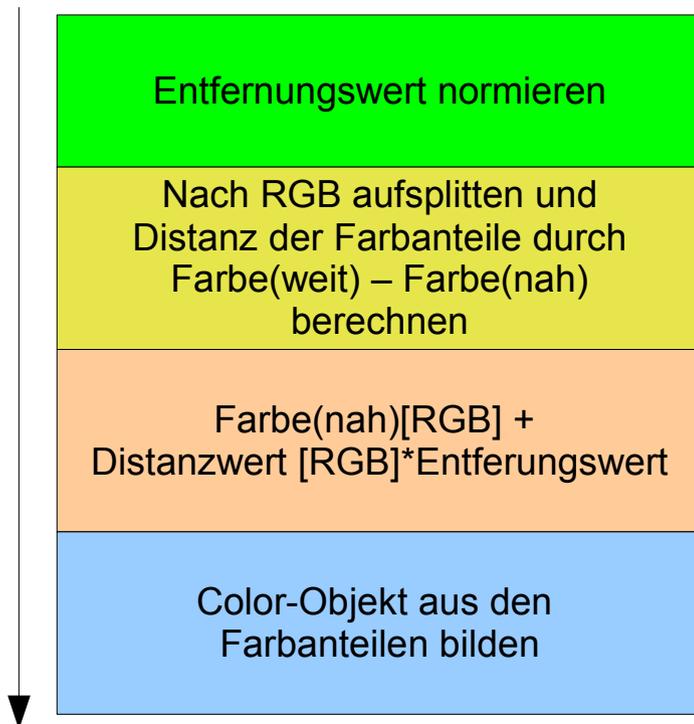
SOMARFF – Oberfläche 2 (Sichten: U-Matrix)





SOMARFF – Oberfläche 2 (Sichten: U-Matrix)

Farbwertberechnung





Zusammenfassung

Was wurde getan?

- Theorie Kohonen-Netz (biolog. Vorbild, Modell, Lernalgorithmus)
- Visualisierungsarten
- Umsetzung in SOMARFF

Ergebnisse

- Bestätigung der Theorie in der Praxis
- Auch auf kleineren Rechnern gut umsetzbar
- Umsetzung in SOMARFF
- Ausgabe kompliziert und rechenintensiv
- Langsamer Lernfortschritt wichtig

Ausblick

- trotz langem Training, hohes Potenzial
- z.B. für autonome Roboter, medizinische Implantate