

# SOMs und verwandte Verfahren

VU Selbst-Organisierende Systeme  
 SOMs und verwandte Verfahren  
 Andreas Rauber  
<http://www.ifs.tuwien.ac.at/~andi>

FACULTY OF INFORMATICS

## Outline

- Self-Organizing Map (SOM)
- Verwandte Verfahren
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

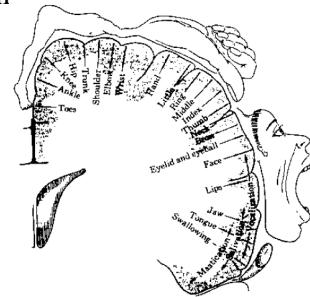
## Outline

- Self-Organizing Map (SOM)
  - Physiologische Grundlagen
  - Architekturen
  - Trainingsprozess
  - Beispiele
- Verwandte Verfahren
- Visualisierungen der SOM
- Applikationen
- Übung

FACULTY OF INFORMATICS

## Self-Organizing Map

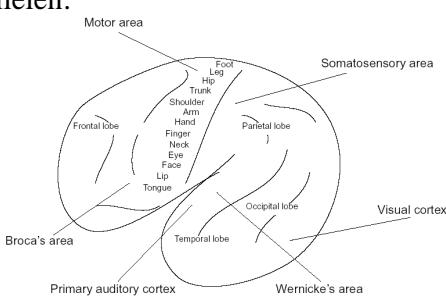
Physiologische  
 Parallelen:



FACULTY OF INFORMATICS

## Self-Organizing Map

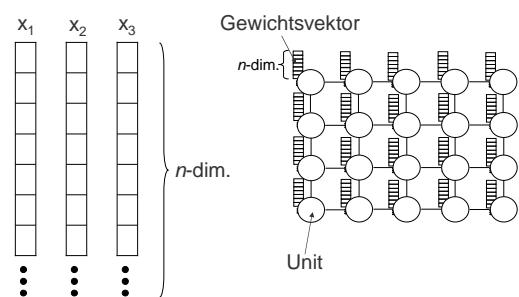
Physiologische  
 Parallelen:



FACULTY OF INFORMATICS

## Self-Organizing Map

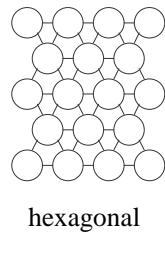
Architektur:



FACULTY OF INFORMATICS

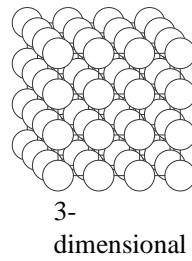
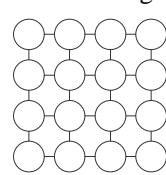
## Self-Organizing Map

Architekturen:



hexagonal

rechteckig

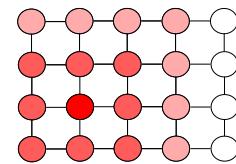
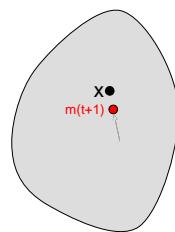


3-dimensional

FACULTY OF INFORMATICS

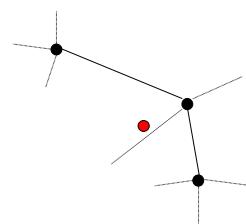
## Self-Organizing Map

Lernverfahren:



FACULTY OF INFORMATICS

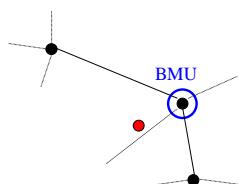
## Trainingsschritt Beispiel



Teil einer SOM:  
Prototyp-Vektoren  
(schwarz) im Gitter;  
Auswahl eines  
Datenvektors (rot)

FACULTY OF INFORMATICS

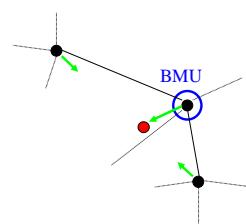
## Training Beispiel



Finden der Best  
Matching Unit (blau)

FACULTY OF INFORMATICS

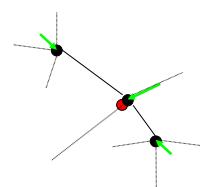
## Trainingsschritt Beispiel



Verschieben der  
Prototyp-Vektoren  
zum Datenvektor  
hin, abhängig vom  
Abstand zur BMU

FACULTY OF INFORMATICS

## Trainingsschritt Beispiel



Nach dem  
Trainingsschritt:  
Prototyp-Vektoren  
sind verschoben

FACULTY OF INFORMATICS

The figure consists of two side-by-side 3D scatter plots. Both plots have axes ranging from -5 to 5 on the x-axis, -4 to 4 on the y-axis, and -4 to 4 on the z-axis. The left plot, labeled 'untrainiert' (untrained), shows a dense, irregular cloud of points representing the initial state of the network weights. The right plot, labeled 'trainiert' (trained), shows a much more organized and structured distribution of points, forming a roughly rectangular prism-like shape, indicating that the training process has regularized the weight distribution.

**TU**  
VIENNA

# Self-Organizing Map

---

- Self-Organizing Map (SOM)

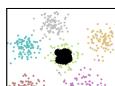
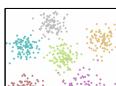


! TU VIENNA

# Self-Organizing Map

---

- Self-Organizing Map (SOM)



The figure consists of three square panels arranged horizontally. The left panel shows a scatter plot of data points colored by their cluster assignment. The middle panel shows a grid of neurons with lines connecting them to form a lattice. The right panel shows a trained SOM grid where neurons have been grouped into clusters of similar colors.

**Lernverfahren:**

- [1] Zufällige Auswahl eines Inputvektors
- [2] Berechnung der Aktivierung der Output Units.  
Aktivierung korreliert mit der Distanz zwischen  
Input und Gewichtsvektor
- [3] Auswahl der best-matching Unit ("Winner") –  
Unit mit größter Aktivierung, d.h. jene Unit mit der  
gerindesten Distanz zwischen Input- und Gewichtsvektor
- [4] Adaption der Gewichtsvektoren des "Winners" und  
der Units in seiner Nachbarschaft
- [5] Wiederholung der Schritte 1-4 bis ein definiertes  
Kriterium für das Ende des Lernens erreicht ist

## Self-Organizing Map

### Initialisierung:

- initialisierung der Gewichtvektoren
  - „zufällige“ Initialisierung
  - Initialisierung mit Beispieldaten
  - komplexere Methoden: PCA, ...

FACULTY OF INFORMATICS

## Self-Organizing Map

- Lernregel

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)]$$

FACULTY OF INFORMATICS

## Self-Organizing Map

### Lernrate:

- zu Beginn des Trainings hohe Lernrate, die sich im Laufe des Trainingsprozesses verringert
- konvergiert gegen 0 --> Konvergenz des Netzes

FACULTY OF INFORMATICS

## Self-Organizing Map

### Nachbarschaftsfunktion

$h_{ci}$ :

- zu Beginn des Trainings hoch, im Laufe des Trainingsprozesses verringern
- Gauss-Funktion, Linear, Box, ...

$$\lim_{t \rightarrow \infty} N_c(t) = \{c\} \quad \lim_{t \rightarrow \infty} h_{ci}(t) = 0 \quad (c \neq i)$$

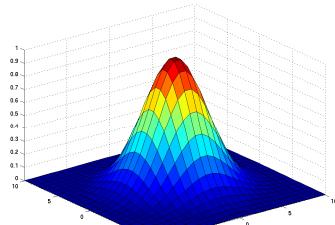
$$\text{z.B.: } h_{ci}(t) = \exp \left( -\frac{\|r_c - r_i\|^2}{2 \cdot \delta(t)^2} \right)$$

FACULTY OF INFORMATICS

## Self-Organizing Map

### Nachbarschaftsfunktion

:



INFORMATICS

## Self-Organizing Map

### Distanzmetriken:

- $L_{\infty}$  metrik

$$d(x_1, x_2) = \max_n (|x_{1n} - x_{2n}|)$$

- minkowski metrik ( $L_k$  norm)

$$d(x_1, x_2) = \sqrt[k]{\sum_n (x_{1n} - x_{2n})^k}$$

- euklidische metrik ( $L_2$  norm)

$$d(x_1, x_2) = \sqrt{\sum_n (x_{1n} - x_{2n})^2}$$

- city block metrik ( $L_1$  norm)

$$d(x_1, x_2) = \sum_n |x_{1n} - x_{2n}|$$

FACULTY OF INFORMATICS

## Self-Organizing Map

Trainingsende:

- fixe Anzahl von Iterationen
- keine weitere Änderung des Abbildungsfehlers ("quantization error")
- Schwellwert für den Abbildungsfehler wird erreicht
- manueller Abbruch

FACULTY OF INFORMATICS

## Self-Organizing Map

Abbildungsfehler (Quantisierungsfehler):

- eine Maßzahl für die Abbildungsqualität der Daten auf die Karte
- mittlere Distanz zwischen jedem Inputvektor und seiner „best-matching Unit“ (Winner)

$$qe = \frac{1}{|I|} \cdot \sum_{i \in I} \|x_i - m_{c(x_i)}\|, \quad c(x_i) = \arg \min_j (\|x_i - m_j\|)$$

$I \dots$  menge der units

FACULTY OF INFORMATICS

## Self-Organizing Map

Topographischer Fehler:

- eine Maßzahl für die Qualität der Topologietreue/-Erhaltung der Daten auf der Karte
- Prozentsatz jener Inputvektoren, deren „best-matching Unit“ und „second best-matching Unit“ nicht benachbart sind

FACULTY OF INFORMATICS

## Self-Organizing Map

Beispiel: Tiere

	dove	hen	duck	goose	owl	hawk	eagle	fox	dog	wolf	cat	tiger	lion	horse	zebra	cow
small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0
fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

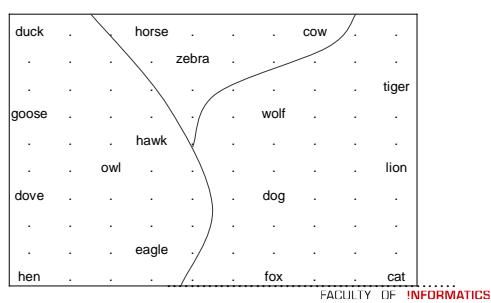
(ritter&kohonen, 1989)

FACULTY OF INFORMATICS

## Self-Organizing Map

Beispiel: Tiere

- 10x10 SOM

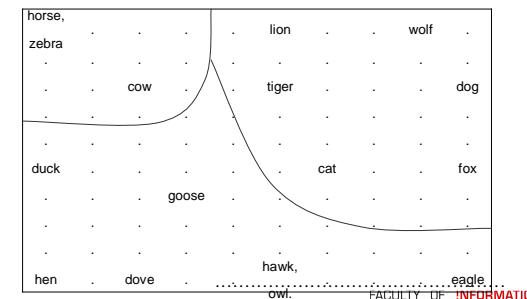


FACULTY OF INFORMATICS

## Self-Organizing Map

Beispiel: Tiere

- 10x10 SOM



FACULTY OF INFORMATICS

## Self-Organizing Map

Beispiel:

Bildkomprimierung



2000 cycles

FACULTY OF INFORMATICS

## Self-Organizing Map

Beispiel:

Bildkomprimierung



2700 cycles

FACULTY OF INFORMATICS

## Outline

- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
  - andere Varianten
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

## Incremental Grid Growing

### Prinzipien

- basiert auf der Self-Organizing Map
- 2-dimensionale Anordnung von Units
- während des Lernprozesses werden an jenen Stellen des Netzwerks Units hinzugefügt, wo der Inputraum noch nicht gut genug abgebildet werden kann
- Verbindungen zwischen Units die sehr unterschiedliche Inputmuster repräsentieren, können entfernt werden
- die Clustergrenzen von Inputmustern sollen besser erkennbar werden als bei der Self-Organizing Map

FACULTY OF INFORMATICS

## Incremental Grid Growing

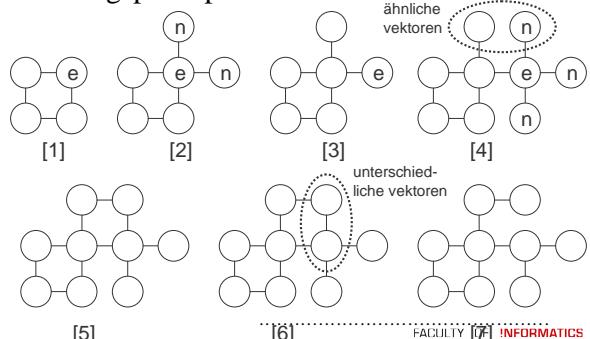
### Trainingsprinzipien:

- [1] Beginn mit 4 Units
- [2] Hinzufügen von neuen Units am Rand des bestehenden Netzes, wo der Fehler der Abbildung am höchsten ist
- [3] Untersuchen der Gewichtsvektoren von benachbarten Units, ob deren Verbindung gelöscht werden soll oder ob dazwischen eine neue Verbindung aufgebaut werden soll

FACULTY OF INFORMATICS

## Incremental Grid Growing

### Trainingsprinzipien:



FACULTY OF INFORMATICS

## Incremental Grid

### Growing

#### Trainingsprozess:

- [1] Auswahl eines Inputmusters  $x$
- [2] Berechnung der Aktivität der Output Units
- [3] Auswahl der Unit mit der höchsten Aktivität ("Winner")
- [4] Anpassung der Gewichtsvektoren (wie bei SOM)
  
- [5] Überprüfung des Abbildungsfehlers der einzelnen Units
- [6] Einfügen einer neuen Unit neben der bestehenden Unit mit dem größten Abbildungsfehler
- [7] Löschen und Hinzufügen von Verbindungen zwischen Units

FACULTY OF INFORMATICS

## Incremental Grid

### Growing

- Berechnung des Abbildungsfehlers

- Abbildungsfehler ist die verbleibende Distanz zwischen dem Inputmuster und dem Gewichtsvektor des Winners

$$E_i(t+1) = E_i(t) + \|x(t) - m_i(t)\|$$

$E_i(t)$  Abbildungsfehler der Unit  $i$  zum Zeitpunkt  $t$

$x(t)$  Inputmuster

$m_i(t)$  Gewichtsvektor der Unit  $i$

- Annahme: der Abbildungsfehler ist bei jenen Units groß, die eine große Anzahl unterschiedlicher Inputmuster repräsentieren.

- Einfügen neuer Units an diesen Stellen führt zu einer besseren Repräsentation des Inputraums

FACULTY OF INFORMATICS

## Incremental Grid

### Growing

- Einfügen neuer Units

- neue Units können dort eingefügt werden, wo eine bestehende Unit noch nicht alle möglichen Nachbarn besitzt, d.h. am Rand des bestehenden Netzes
- der Gewichtsvektor der neu eingefügten Unit wird mit dem Mittelwert der Gewichtsvektoren seiner Nachbarn initialisiert

FACULTY OF INFORMATICS

## Incremental Grid

### Growing

- Hinzufügen und Löschen von Verbindungen

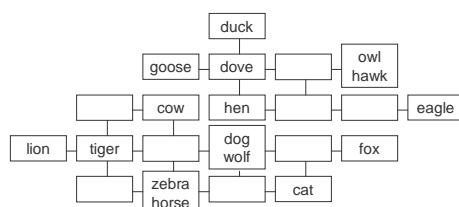
- nach dem Einfügen ist die neue Unit nur mit jener Unit verbunden, die den höchsten Abbildungsfehler aufgewiesen hat
- die Gewichtsvektoren zweier noch unverbundener jedoch benachbarter Units werden untersucht. Haben diese Gewichtsvektoren eine geringe Distanz, wird eine neue Verbindung zwischen diesen aufgebaut
- die Gewichtsvektoren von verbundenen Units werden untersucht. Haben diese eine große Distanz, wird die Verbindung entfernt

FACULTY OF INFORMATICS

## Incremental Grid

### Growing

- Beispiel: Tiere



FACULTY OF INFORMATICS

## Outline

- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
  - andere Varianten
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

## Growing Grid

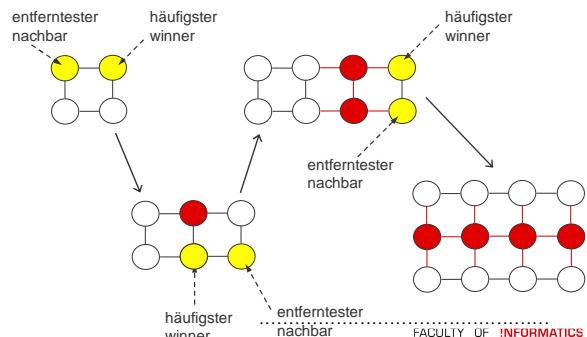
### Prinzipien:

- basiert auf der Self-Organizing Map
- 2-dimensionale (rechteckige) Anordnung von Units
- während des Lernprozesses werden an jenen Stellen des Netzwerkes ganze Reihen bzw. Spalten von Units hinzugefügt, wo der Raum noch nicht gut abgebildet werden kann

FACULTY OF INFORMATICS

## Growing Grid

### Prinzipien



FACULTY OF INFORMATICS

## Growing Grid

### Lernverfahren:

- [1] Auswahl eines Inputmusters  $x$
- [2] Berechnung der Aktivitäten der Units -> "Winner"
- [3] Anpassung der Gewichtsvektoren des "Winner" und seiner Nachbarn innerhalb einer fixen Umgebung
- [4] Erhöhen des "Winner-Zählers"
- [5] jeweils nach einer fixen Anzahl von Lernschritten werden neue Units zwischen dem häufigsten Winner und der Unit mit dem am weitest entfernten Gewichtsvektor eingefügt
- [6] Abbruch, wenn eine maximale Netzwerkgröße bzw. ein bestimmter Quantisierungsfehler erreicht ist

FACULTY OF INFORMATICS

## Growing Grid

### Anmerkungen

- Adaption der Gewichte wie bei der Self-Organizing Map
- statt des Winner-Zählers könnte auch der Quantisierungsfehler der einzelnen Units als Kriterium für die Expansion herangezogen werden

FACULTY OF INFORMATICS

## Outline

- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
  - andere Varianten
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

## Growing Cell Structures

### Prinzipien:

- basiert auf der Self-Organizing Map
- 2-dimensionale Anordnung von Units
- Units stehen untereinander in einer "Dreiecksbeziehung"
- während des Lernprozesses werden an jenen Stellen des Netzwerkes Units hinzugefügt, wo der Inputraum noch nicht gut genug abgebildet werden kann
- Units die einen zu geringen Beitrag zur Abbildung des Inputraums liefern werden entfernt
- Clustergrenzen von Inputmustern sollen durch ein Aufspalten des Netzes in mehrere unabhängige Teilnetze erkennbar werden

FACULTY OF INFORMATICS

## Growing Cell Structures

### Lernprozess:

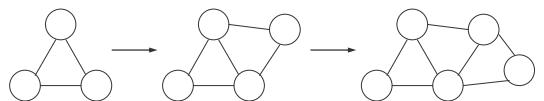
- [1] Trainingsbeginn mit 3 units
- [2] Hinzufügen von units als neue Nachbarn jener units, die häufig "winner" waren  
Idee: eine unit, die sehr häufig Winner ist, repräsentiert eine große Anzahl von Inputmustern, hohe Dichte im Datenraum
- [3] Löschen von Units die sehr selten bzw. nie "Winner" waren

FACULTY OF INFORMATICS

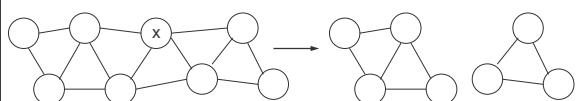
## Growing Cell Structures

### Prinzipien:

#### Einfügen von Units



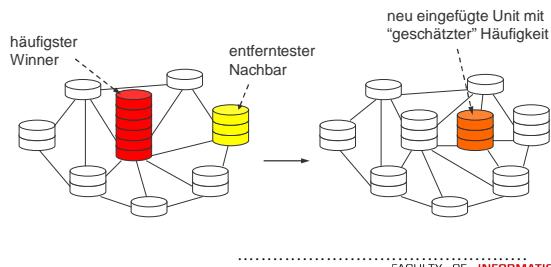
#### Löschen von Units



FACULTY OF INFORMATICS

## Growing Cell Structures

### Einfügen von Units

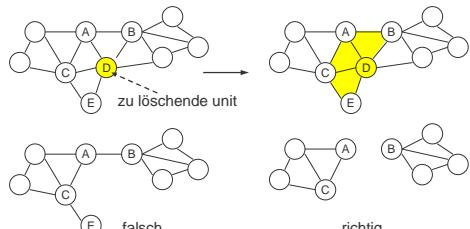


FACULTY OF INFORMATICS

## Growing Cell Structures

### Löschen von Units

- Löschen der Unit mit zu geringer Winner-häufigkeit und aller dreiecke dieser unit



FACULTY OF INFORMATICS

## Growing Cell Structures

### Lernverfahren:

- [1] Auswahl eines Inputmusters  $x$
- [2] Berechnung der Aktivität der Units und Auswahl des "Winners"
- [3] Anpassung der Gewichtsvektoren des "Winners" und seiner Nachbarn (Dreiecksstruktur)
- [4] Erhöhen des Winner-Zählers bzw. Reduktion der Zähler der nicht-winner
- [5] Jeweils nach einer fixen Anzahl von Lernschritten:  
Einfügen von neuen Units an jener Stelle, wo sich der häufigste "Winner" befindet. Löschen von jenen Units, die zu selten "Winner" waren

FACULTY OF INFORMATICS

## Growing Cell Structures

### Lernverfahren:

- Gewichtsvektor der neu eingefügten Unit wird mit dem Mittel der Gewichtsvektoren der benachbarten Units initialisiert

FACULTY OF INFORMATICS

## Outline

- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
  - andere Varianten
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

## Hierarchical Feature Maps

### Architektur:

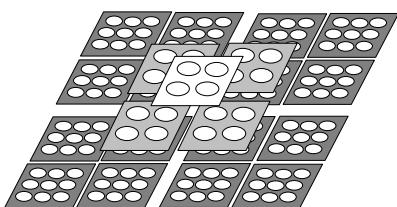
- basiert auf Self-Organizing Maps
- pyramidenförmige Anordnung von SOMs
- Idee: ähnliche Inputmuster sollen durch die selbe SOM repräsentiert werden.

Die Clusterstruktur soll dadurch besser darstellbar werden

FACULTY OF INFORMATICS

## Hierarchical Feature Maps

### Architektur:



FACULTY OF INFORMATICS

## Hierarchical Feature Maps

### Lernverfahren:

- [1] Auswahl eines Inputmusters  $x$
- [2] Präsentation dieses Musters an die Karte der obersten Schicht
- [3] Trainieren dieser Karte
- [4] wenn das Training beendet ist, werden die Karten der nächsten Schicht mit den jeweils zugeordneten Daten trainiert, bis die unterste Schicht trainiert ist

FACULTY OF INFORMATICS

## Hierarchical Feature Maps

### Lernverfahren -

#### Vorteile:

- schnellerer Lernprozess als bei einer normalen SOM vergleichbarer Größe.

#### Begründung:

- die Inputmuster können für das Training in den unteren Schichten der Hierarchie um jene Komponenten reduziert werden, die in allen von einer Unit repräsentierten Mustern gleich (oder zumindest sehr ähnlich) sind
- die Karten sind kleiner, d.h. die Winner-Suche und die Adaption der Gewichtsvektoren ist weniger rechenaufwendig

FACULTY OF INFORMATICS

## Hierarchical Feature Maps

### Lernverfahren - lernverfahren - cons Nachteile:

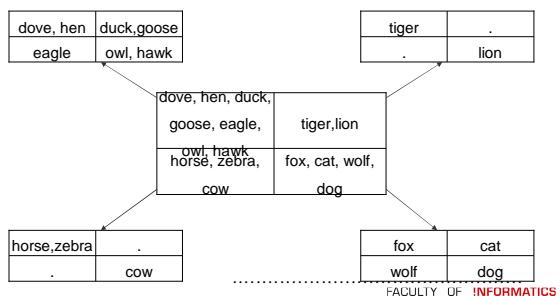
- Festlegung der Struktur vor dem Training (Größe und Anzahl der Karten, Tiefe der Hierarchie)
  - bei unbekannten Daten ist eine Festlegung der Netzwerkarchitektur a priori schwierig
  - das Netz zwingt der Datenrepräsentation seine Form auf
  - der umgekehrte Weg wäre wünschenswert

FACULTY OF INFORMATICS

## Hierarchical Feature Maps

### Beispiel:

- hierarchisches Clustering von Tieren



## Outline

- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
  - andere Varianten
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

## Growing Hierarchical SOM

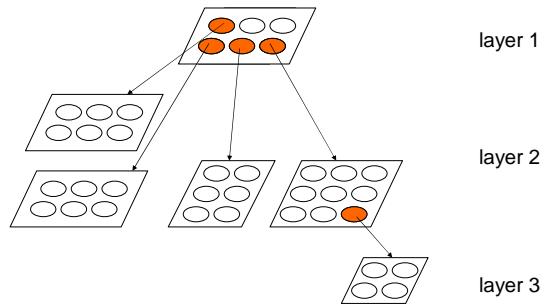
### Architektur:

- basiert auf Growing Grid und Hierarchical Feature Map
- Hierarchische Struktur von Growing Grids
- Karten wachsen und neue Hierarchiezweige entstehen dynamisch während des Trainingsprozesses dort, wo die Abbildung der Inputdaten noch nicht genau genug ist
- A. Rauber, D. Merkl, and M. Dittenbach: **The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data**  
In: IEEE Transactions on Neural Networks, Vol. 13, No 6, pp. 1331-1341, November 2002. IEEE.

FACULTY OF INFORMATICS

## Growing Hierarchical SOM

### Lernverfahren:



FACULTY OF INFORMATICS

## Growing Hierarchical SOM

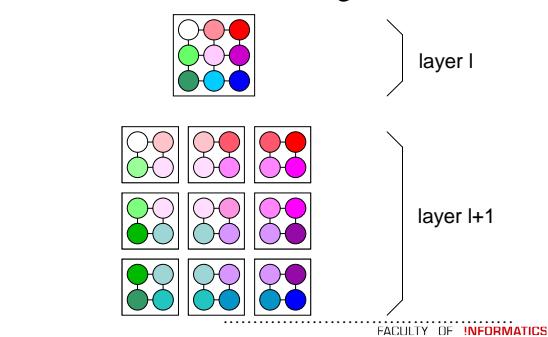
### Lernverfahren:

- zwei Parameter bestimmen den Trainingsverlauf maßgeblich
  - $\tau_1$  Granularität/Qualitätsgewinn der Datenrepräsentation pro Kartenebene (Interval [0,1])
    - beeinflusst die Größe der Karten und dadurch auch die Tiefe der Hierarchie
    - je kleiner  $\tau_1$ , desto genauer bilden die Karten die Inputdaten ab, d.h. desto größer werden die Karten
  - $\tau_2$  höchste Granularität der Datenrepräsentation (interval [0,1])
    - bestimmt die Tiefe der Hierarchie, d.h. die Anzahl der Schichten

FACULTY OF INFORMATICS

## Growing Hierarchical SOM

### Lernverfahren - Initialisierung:



FACULTY OF INFORMATICS

The image shows the TU Vienna logo in the top left corner, consisting of a red square with a white exclamation mark and the letters 'TU VIENNA'. To the right of the logo, the word 'Outline' is written in a large, black, serif font. Below the title, there is a horizontal dotted line.

- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
  - andere Varianten
- Visualisierungen der SOM
- Applikationen

.....

FACULTY OF !INFORMATICS



# Mnemonic SOM

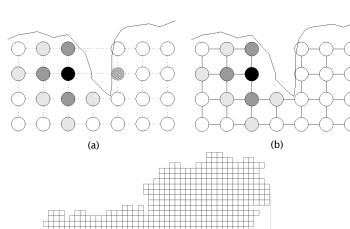
---

! TU VIENNA

# Mnemonic SOM

---

- Anpassung der Nachbarschafts-Berechnung



---

Figure 1 illustrates the Mnemonic Self-Organizing Map (SOM). Part (a) shows six clusters of blue dots labeled 1 through 6. Part (b) shows a 10x10 grid of neurons, each colored according to its assigned class. The legend indicates:

Class	Color
1	Red
2	Blue
3	Yellow
4	Green
5	Magenta
6	Cyan

The trained SOM grid in (b) has neurons labeled 1 through 6, where each label corresponds to one of the input classes from (a).

! TU VIENNA Mnemonic SOM

---

- Iris-Datenset

**(a)**

Setosa  
Versicolor  
Virginica

**(b)**

Setosa  
Versicolor  
Virginica

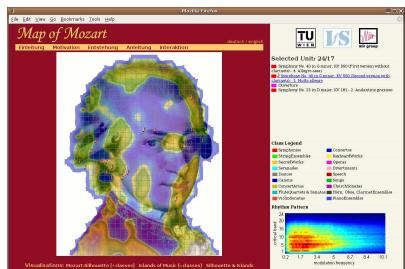
Class Legend:

Setosa	Red
Versicolor	Green
Virginica	Blue

The diagram illustrates the Mnemonic Self-Organizing Map (SOM). The top part shows a 2D grid of nodes representing various concepts like 'Christians', 'Soviet Union', 'Iraq', 'Syria', 'India', 'Pakistan', 'Africa', 'Asia', 'British politics', 'NATO', 'French politics', 'German politics', 'Intelligence Agencies', 'Vietnam', 'de Gaulle', and 'French politics'. A dotted line highlights a cluster of nodes. The bottom part shows a 3D visualization of the SOM's internal structure, where the nodes are arranged in a cube-like grid. The labels 'FACULTY OF INFORMATICS' are visible at the bottom right.

## Mnemonic SOM

- Map of Mozart



FACULTY OF INFORMATICS

## Outline

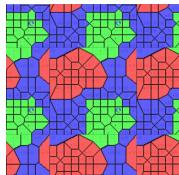
- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
  - andere Varianten
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

## Andere SOM Formen

### Toroide Topologie

- keine Border-effects
- Ultsch, A.: Maps for the Visualization of high-dimensional Data Spaces, In Proc. WSOM'03, Kyushu, Japan, pp. 225-230, (2003)

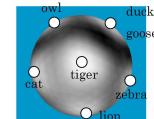


FACULTY OF INFORMATICS

## Andere SOM Formen

### Kugel-SOM

- Kugel anstelle von flacher Karte
- Ritter, H., **Self-Organizing Maps on non-euclidean Spaces**, Kohonen Maps, Oja, E. and Kaski, S. editors, Elsevier, 95-110, 1999
- Yingxin Wu, Masahiro Takatsuka. **Spherical self-organizing map using efficient indexed geodesic data structure** Source Neural Networks. 19(6):900-910. July 2006. Elsevier Science.

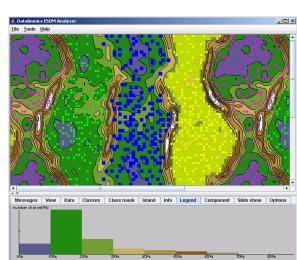


FACULTY OF INFORMATICS

## Andere SOM Formen

### Emergent SOM

- large SOM, more nodes than data points
- Ultsch, A.: Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series, In *Kohonen Maps*, (1999) , pp. 33-46
- databionic-esom.sourceforge.net



FACULTY OF INFORMATICS

## Andere SOM Formen

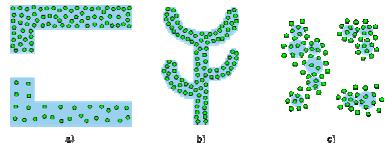
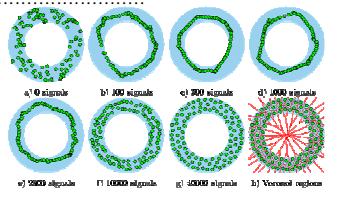
### Neural Gas

- keine Grid-Topologie im Ausgaberaum
- Quantisierung und Topologieerhaltung, aber keine Visualisierung
- T. M. Martinetz and K. J. Schulten. A ``neural-gas'' network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 397-402. North-Holland, Amsterdam, 1991.

FACULTY OF INFORMATICS

## Andere SOM Formen

### Neural Gas



<http://www.neuroinformatik.ruhr-uni-bochum.de/VDM/research/gsn/JavaPaper/node16.html>

## Andere SOM Formen

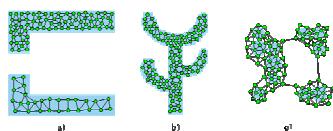
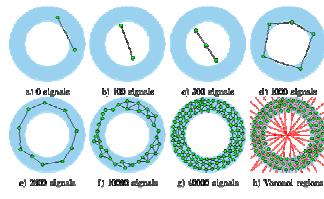
### Growing Neural Gas

- NG mit Wachstum (Growing Cell Structures)
- Quantisierung und Topologieerhaltung, aber keine Visualisierung
- B. Fritzke. Fast learning with incremental RBF networks. *Neural Processing Letters*, 1(1):-5, 1994.

FACULTY OF INFORMATICS

## Andere SOM Formen

### Growing Neural Gas



<http://www.neuroinformatik.ruhr-uni-bochum.de/VDM/research/gsn/JavaPaper/node169.htm>

## Outline

- SOM
- Verwandte Verfahren
  - Incremental Grid Growing
  - Growing Grid
  - Growing Cell Structures
  - Hierarchical Feature Maps
  - Growing Hierarchical SOM (GHSOM)
  - Mnemonic SOM
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS