

SOMs und verwandte Verfahren

VU Selbst-Organisierende Systeme
SOMs und verwandte Verfahren
Andreas Rauber

<http://www.ifs.tuwien.ac.at/~andi>

Outline

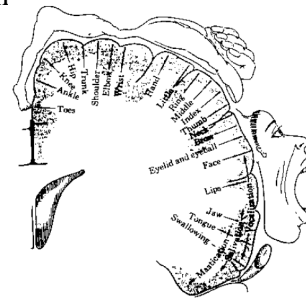
- Self-Organizing Map (SOM)
- Verwandte Verfahren
- Visualisierungen der SOM
- Applikationen

Outline

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

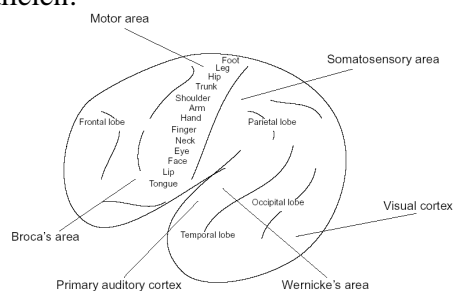
Self-Organizing Map

Physiologische
Parallelen:



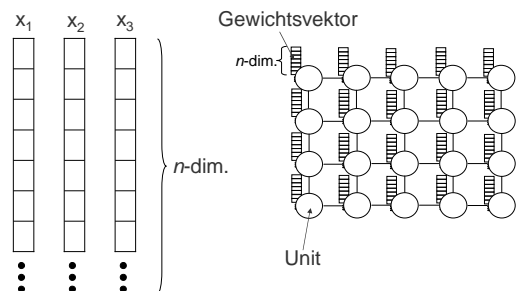
Self-Organizing Map

Physiologische
Parallelen:



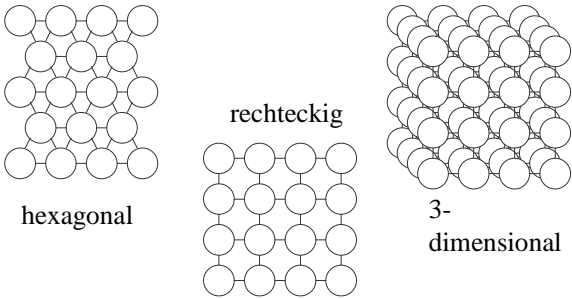
Self-Organizing Map

Architektur:



TU VIENNA Self-Organizing Map

Architekturen:

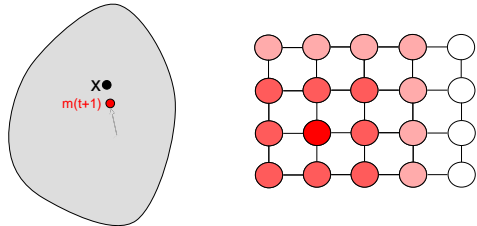


hexagonal rechteckig 3-dimensional

FACULTY OF INFORMATICS

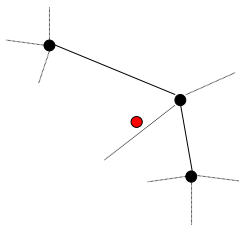
TU VIENNA Self-Organizing Map

Lernverfahren:



FACULTY OF INFORMATICS

TU VIENNA Trainingsschritt Beispiel

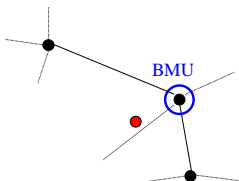


Teil einer SOM:
Prototyp-Vektoren
(schwarz) im Gitter;

Auswahl eines
Datenvektors (rot)

FACULTY OF INFORMATICS

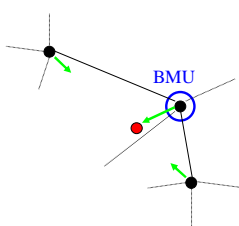
TU VIENNA Training Beispiel



Finden der Best
Matching Unit (blau)

FACULTY OF INFORMATICS

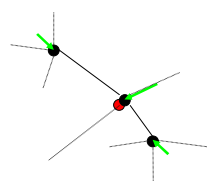
TU VIENNA Trainingsschritt Beispiel



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

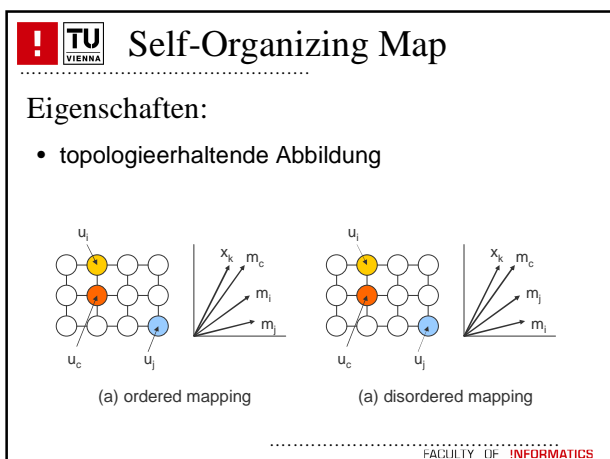
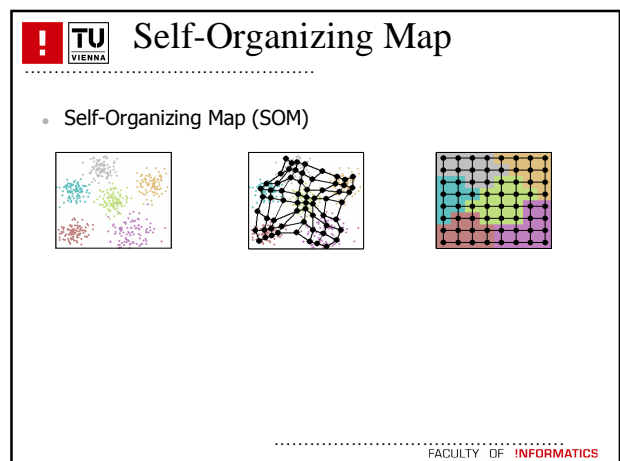
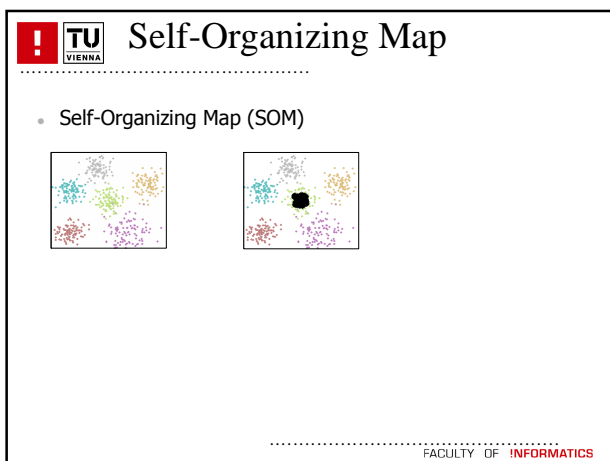
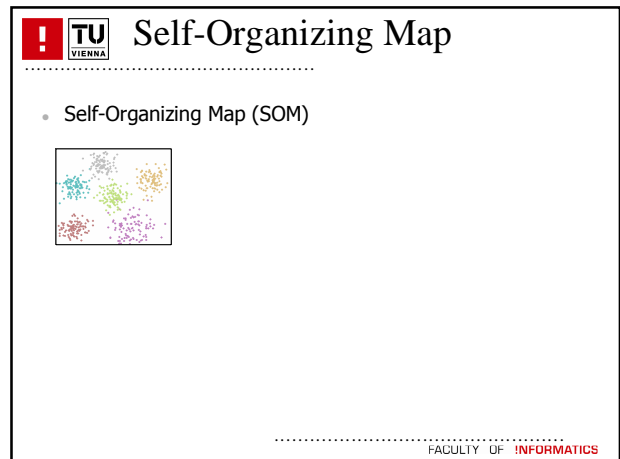
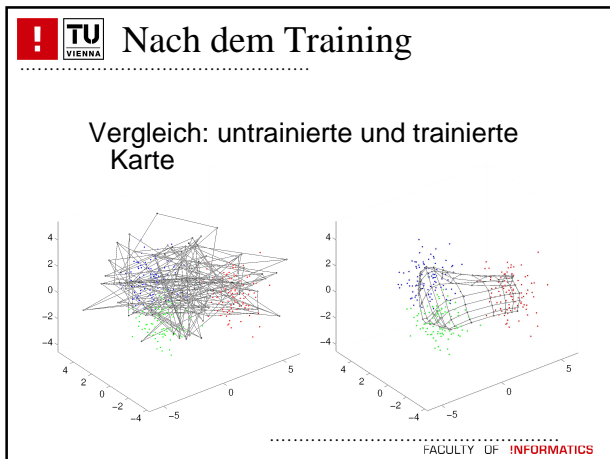
FACULTY OF INFORMATICS

TU VIENNA Trainingsschritt Beispiel



Nach dem
Trainingsschritt:
Prototyp-Vektoren
sind verschoben

FACULTY OF INFORMATICS



Initialisierung:

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

- Lernregel

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

Lernrate:

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

Nachbarschaftsfunktion

h_{ci} :

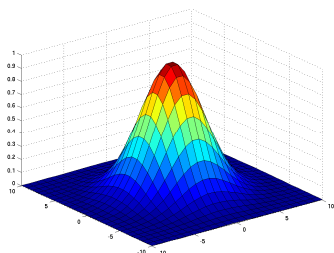
- zu Beginn des Trainings hoch, im Laufe des Trainingsprozesses verringernd
- Gauss-Funktion, Linear, Box, ...

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

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

Nachbarschaftsfunktion

:



Distanzmetriken:

- L_{inf} 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)

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

Trainingsende:

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

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

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

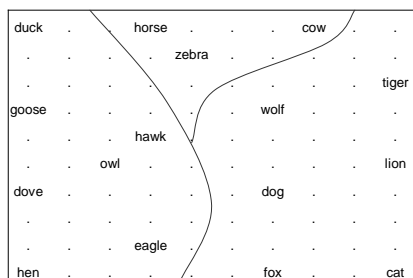
Beispiel: Tiere

	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	1	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	1	1	1	0	1	1	1	1	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)

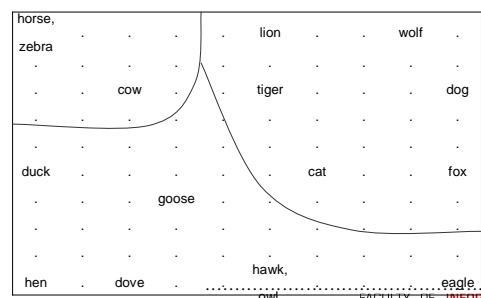
Beispiel: Tiere

- 10x10 SOM




Beispiel: Tiere

- 10x10 SOM



TU VIENNA Self-Organizing Map

Beispiel:
Bildkomprimierung




2000 cycles

FACULTY OF INFORMATICS

TU VIENNA Self-Organizing Map

Beispiel:
Bildkomprimierung



2700 cycles

FACULTY OF INFORMATICS

TU VIENNA 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

TU VIENNA 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

TU VIENNA Incremental Grid Growing

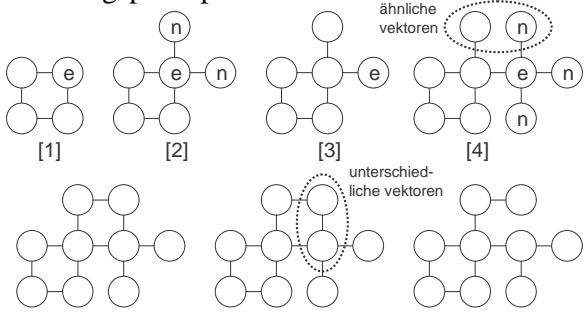
Trainingsprinzipien:

- [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

TU VIENNA Incremental Grid Growing

Trainingsprinzipien:



FACULTY OF INFORMATICS

! TU VIENNA 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

! TU VIENNA 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

! TU VIENNA 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

! TU VIENNA 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

! TU VIENNA Incremental Grid Growing

- Beispiel: Tiere

FACULTY OF INFORMATICS

! TU VIENNA Outline

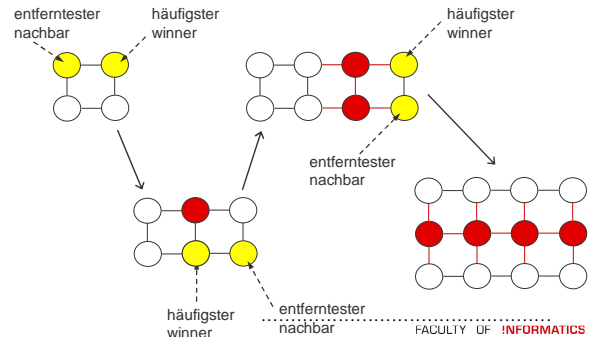
- SOM
- Verwandte Verfahren
 - Incremental Grid Growing
 - Growing Grid
 - Growing Cell Structures
 - Hierarchical Feature Maps
 - Growing Hierarchical SOM (GHSOM)
 - Mnemonic SOM
 - andere Varianten
- Visualisierungen der SOM
- Applikationen

FACULTY OF INFORMATICS

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

Prinzipien



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 Gewinner und der Unit mit dem am weitest entfernten Gewichtsvektor eingefügt
- [6] Abbruch, wenn eine maximale Netzwerkgröße bzw. ein bestimmter Quantisierungsfehler erreicht ist

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

- 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

Prinzipien:

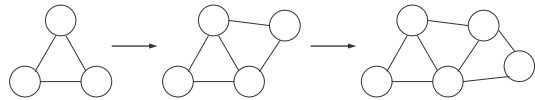
- basiert auf der Self-Organizing Map
- 2-dimensionale Anordnung von Units
- Units stehen untereinander in einer "Dreiecksbeziehung"
- während des Lernprozesses werden an jenen Stellen des Netzwerkes Units hinzugefügt, wo der Inputraum noch nicht gut genug abgebildet werden kann
- Units die einen zu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

Lernprozess:

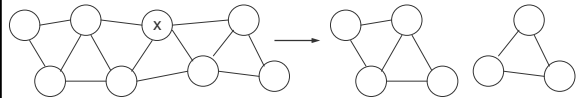
- [1] Trainingsbeginn mit 3 units
- [2] Hinzufügen von units als neue Nachbarn jener units, die häufig "winner" waren
Idee: eine unit, die sehr häufig Winner ist, repräsentiert eine große Anzahl von Inputmustern, hohe Dichte im Datenraum
- [3] Löschen von Units die sehr selten bzw. nie "Winner" waren

Prinzipien:

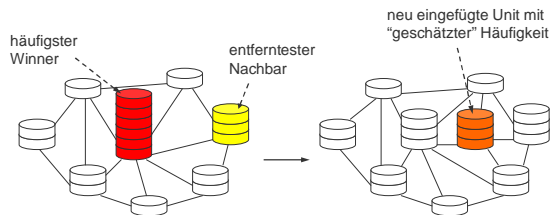
Einfügen von Units



Löschen von Units

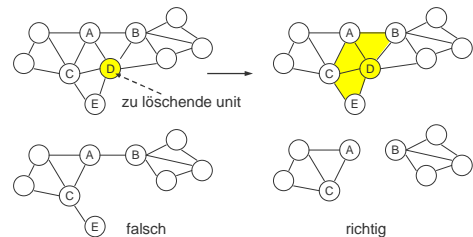


Einfügen von Units



Löschen von Units

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



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

Lernverfahren:

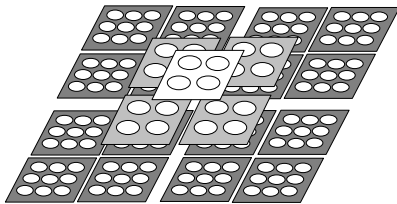
- Gewichtsvektor der neu eingefügten Unit wird mit dem Mittel der Gewichtsvektoren der benachbarten Units initialisiert

- 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

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

Architektur:



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 der Unit in der darüberliegenden Karte zugeordneten Daten trainiert, bis die unterste Schicht trainiert ist

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

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

Hierarchical Feature Maps

Beispiel:

- hierarchisches Clustering von Tieren

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 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
 - τ_1 Granularität/Qualitätsgewinn der Datenrepräsentation pro Kartenebene (Interval [0,1])
 - beeinflusst die Größe der Karten und dadurch auch die Tiefe der Hierarchie
 - je kleiner τ_1 , desto genauer bilden die Karten die Inputdaten ab, d.h. desto größer werden die Karten
 - τ_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

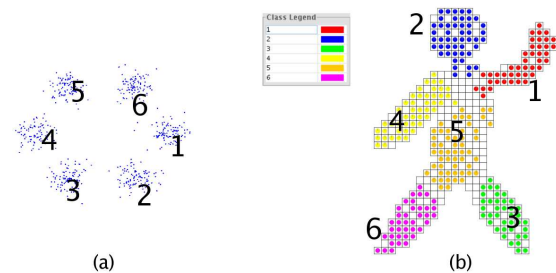
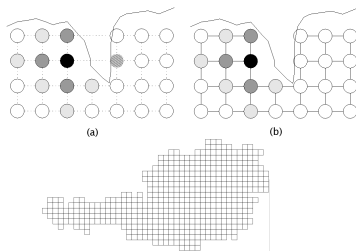
Lernverfahren - Initialisierung:

FACULTY OF INFORMATICS

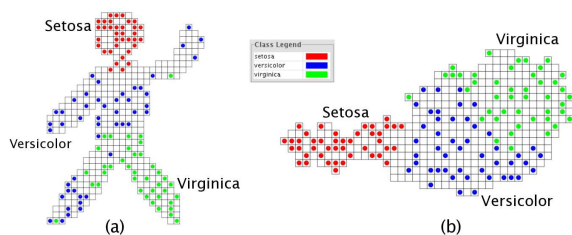
- SOM
- Verwandte Verfahren
 - Incremental Grid Growing
 - Growing Grid
 - Growing Cell Structures
 - Hierarchical Feature Maps
 - Growing Hierarchical SOM (GHSOM)
 - Mnemonic SOM
 - andere Varianten
- Visualisierungen der SOM
- Applikationen

- Nachteil konventioneller SOM: Bereiche schwer zu beschreiben
- oben/unten/Mitte, Ecken
- Schaffung einer SOM mit internen Strukturen
- z.B. Landkarten: intern Flüsse, Städte, Provinzen,...
- Achtung: Form darf nicht mit Daten korrelieren!

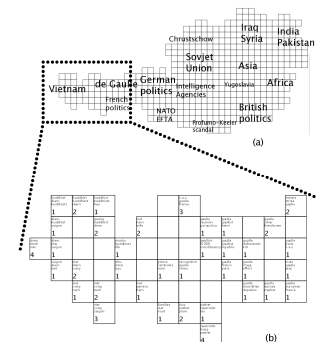
- Anpassung der Nachbarschafts-Berechnung



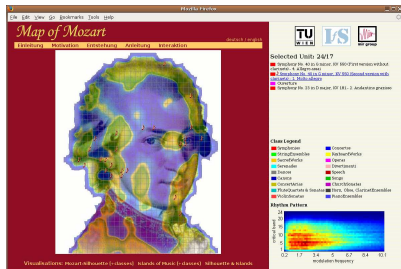
- Iris-Datenset



- Time Artikel



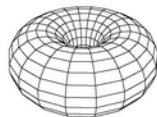
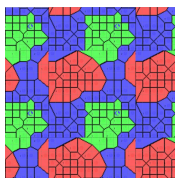
- Map of Mozart



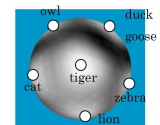
- SOM
- Verwandte Verfahren
 - Incremental Grid Growing
 - Growing Grid
 - Growing Cell Structures
 - Hierarchical Feature Maps
 - Growing Hierarchical SOM (GHSOM)
 - Mnemonic SOM
 - andere Varianten
- Visualisierungen der SOM
- Applikationen

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)



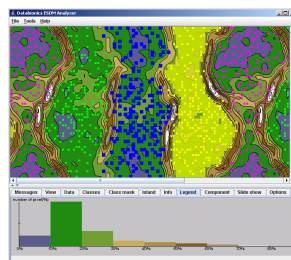
Kugel-SOM



- 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.

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



Neural Gas

- keine Grid-Topologie im Ausgaberaum
- Quantisierung und Topologierhaltung, 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.

TU VIENNA **Andere SOM Formen**

Neural Gas

a) 0 signals b) 100 signals c) 200 signals d) 4000 signals
e) 10000 signals f) 100000 signals g) 400000 signals h) Voronoi regions

a) b) c)

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

FACULTY OF INFORMATICS

TU VIENNA **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

TU VIENNA **Andere SOM Formen**

Growing Neural Gas

a) 0 signals b) 100 signals c) 200 signals d) 4000 signals
e) 10000 signals f) 100000 signals g) 400000 signals h) Voronoi regions

a) b) c)

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

FACULTY OF INFORMATICS

TU VIENNA **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