

# Self Organizing Maps

Andreas Rauber

January 14, 2010

Institut für Softwaretechnik & Interaktive Systeme

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

Selbstorganisierende Systeme WS09/10

Folien übertragen<sup>1</sup> in Latex / Lyx. -  
modifiziert/gekürzt und möglicherweise mit Fehlern

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>SOM Basic</b>                         | <b>1</b> |
| 1.1      | Lernverfahren: . . . . .                 | 1        |
| 1.2      | Initialisierung der Gewichtsvektoren . . | 1        |
| 1.3      | Lernregel . . . . .                      | 2        |
| 1.4      | Distanzmetriken . . . . .                | 2        |
| 1.5      | Trainingsende . . . . .                  | 2        |
| 1.6      | Eigenschaften . . . . .                  | 2        |
| 1.7      | Abbildungsfehler - Quantisierungsfehler  | 2        |
| 1.8      | Topographischer Fehler . . . . .         | 2        |
| 1.9      | Vor-Nachteile . . . . .                  | 3        |
| 1.10     | Verwandte Verfahren . . . . .            | 3        |

---

<sup>1</sup>von Georg Regal

## 1 SOM Basic

### 1.1 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 geringsten 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

### 1.2 Initialisierung der Gewichtsvektoren

- zufällige Initialisierung:
  - längerer Trainingsprozess (mehr Iterationen)
  - in jedem Trainingsprozess andere (rotierte?) Karte
- Initialisierung mit Beispieldaten
  - Vordefinierte Orientierung der Karte
  - Risiko, Topologieverletzungen „vorzuprogrammieren“
  - evtl. basierend auf existierender SOM
- komplexere Methoden: PCA
  - Initialisierung entlang der ersten beiden Hauptkomponenten
  - aufwändig bei großen Datenmengen
  - beschleunigt Trainingsprozess -> direkt zu Finetuning-Phase

### 1.3 Lernregel

$$m_i(t+1) = m_i(t) + \alpha(t) * h(t) * [x(t) - m_i(t)] \quad (1)$$

- $m_i$  : Gewichtsvektor der Unit  $i$  (Codebook)
- $x(t)$ : Trainingsvektor zum Zeitpunkt  $t$
- $\alpha$ : lernrate
  - zu Beginn des Trainings hohe Lernrate, die sich im Laufe des Trainingsprozesses verringert
  - konvergiert gegen 0  $\rightarrow$  Konvergenz des Netzes
- $h_{ci}$ : Nachbarschaftsfunktion zum Zeitpunkt  $t$  zwischen Winner  $c$  und aktueller Unit  $i$ 
  - zu Beginn des Trainings hoch, im Laufe des Trainingsprozesses verringernd
  - Gauss-Funktion, Mexican Hat, Linear, Box, ...
  - Vorteile v. boxed neighborhood: performance
  - $\lim(t \rightarrow \infty) = N_c(t) = \{c\}$
  - $\lim(t \rightarrow \infty) = h_{ci}(t) = 0 \quad (c \neq i)$
  - z.B.  $h_{ci}(t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2 * \delta(t)^2}\right)$

### 1.4 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 Distanz ( $L_2$  norm)
  - $d(x_1, x_2) = \sqrt{\sum_n (x_{1n} - x_{2n})^2}$
- City block Metrik ( $L_1$  norm)
  - $d(x_1, x_2) = \sum_n |x_{1n} - x_{2n}|$

### 1.5 Trainingsende

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

### 1.6 Eigenschaften

- Vektor-Quantisierung
  - Clustering auf Units ähnlich k-means
  - je mehr Units, desto bessere Quantisierung
  - Beziehung zwischen "Clustern"
  - Achtung: 1 Knoten entspricht nicht einem Cluster!
  - Anzahl der Knoten nicht unmittelbar kritisch
- Topologie-erhaltende Abbildung (Fig. 1 p.3)
  - Projektion aus hoch-dim Raum in 2D
  - ähnlich PCA, Sammons Mapping
  - nicht-linear

### 1.7 Abbildungsfehler - Quantisierungsfehler

$$q_e = \frac{1}{|I|} * \sum_{i \in I} \|x_i - m_c(x_i)\|, c(x_i) = \arg \min_j (\|x_i - m_j\|) \quad (2)$$

*I ... menge der units*

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

### 1.8 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

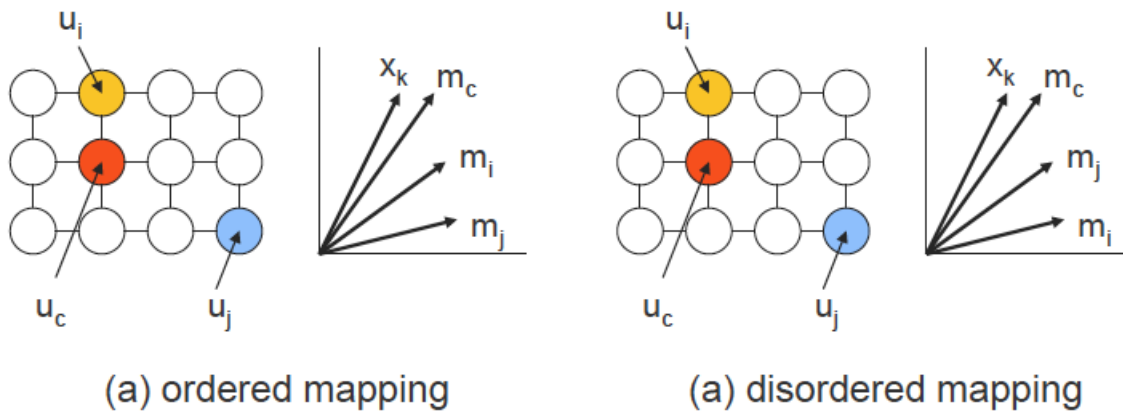


Figure 1: topologieerhaltende Abbildung

## 1.9 Vor-Nachteile

- Vorteile der Standard SOM
  - extrem stabil
    - verarbeitet sehr hochdimensionale Datenräume
    - intuitive Interpretation (Visualisierungen)
- Nachteile der Standard SOM
  - fixe Größe
  - keine hierarchischen Strukturen
  - statische Rechteckform
  - Clusterstruktur nur aus Visualisierungen ableitbar

## 1.10 Verwandte Verfahren

- Incremental Grid Growing
- Growing Grid
- Growing Cell Structures
- Hierarchical Feature Maps
- Growing Hierarchical SOM (GHSOM)
- Mnemonic SOM