



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# SELF-ORGANIZING PLAYER MODELING

Fachseminar "Machine Learning"

Letztes Update: 12. Januar 2016

David Lerch

Studienbereich Informatik  
Hochschule RheinMain

# MAPS UND



# EINLEITUNG

## EINE KLEINE TABELLE

	Population (millions mid-1990)	Area (thousands of square kilometers)	GNP per capita <sup>a</sup>		Average annual rate of inflation <sup>a</sup> (percent)		Life expectancy birth (years) 1990	Adult illiteracy (percent)	
			Dollars 1990	Average annual growth rate (percent) 1965-90	1965-80	1980-90		Female	Total 1990
<b>Low-income economies</b>	<b>3,058.3 t</b>	<b>37,780 t</b>	<b>350 w</b>	<b>2.9 w</b>	<b>8.0 w</b>	<b>9.6 w</b>	<b>62 w</b>	<b>52 w</b>	<b>40 w</b>
<b>China and India</b>	<b>1,983.2 t</b>	<b>12,849 t</b>	<b>360 w</b>	<b>3.7 w</b>	<b>3.2 w</b>	<b>6.8 w</b>	<b>65 w</b>	<b>50 w</b>	<b>37 w</b>
<b>Other low-income</b>	<b>1,075.1 t</b>	<b>24,931 t</b>	<b>320 w</b>	<b>1.7 w</b>	<b>17.3 w</b>	<b>15.1 w</b>	<b>55 w</b>	<b>56 w</b>	<b>45 w</b>
1 Mozambique	15.7	802	80	..	..	36.6	47	79	67
2 Tanzania <sup>b</sup>	24.5	945	110	-0.2	9.6	25.8	48	..	..
3 Ethiopia	51.2	1,222	120	-0.2	3.4	2.1	48	..	..
4 Somalia	7.8	638	120	-0.1	10.2	49.7	48	86	76
5 Nepal	18.9	141	170	0.5	7.8	9.1	52	87	74
6 Chad	5.7	1,284	190	-1.1	6.2	1.2	47	82	70
7 Bhutan	1.4	47	190	..	..	8.4	49	75	62
8 Lao PDR	4.1	237	200	..	..	..	49	..	..
9 Malawi	8.5	118	200	0.9	7.4	14.7	46	..	..
10 Bangladesh	106.7	144	210	0.7	15.9	9.6	52	78	65
11 Burundi	5.4	28	210	3.4	5.0	4.2	47	60	50
12 Zaire	37.3	2,345	220	-2.2	24.7	60.9	52	39	28
13 Uganda	16.3	236	220	-2.4	21.4	107.0	47	65	52
14 Madagascar	11.7	587	230	-1.9	7.7	17.1	51	27	20
15 Sierra Leone	4.1	72	240	0.0	7.9	56.1	42	89	79

Aus: World Development Report 1992, World Bank

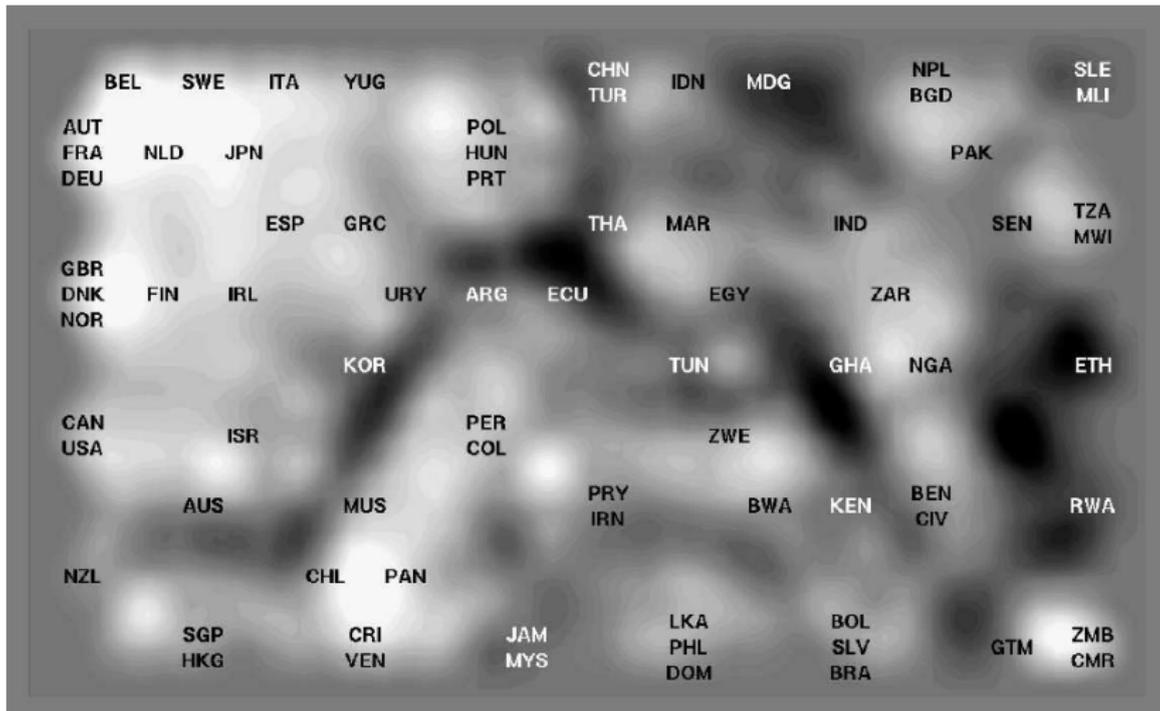
## EINE KLEINE TABELLE

	Population (millions) mid-1990	Area (thousands of square kilometers)	GNP per capita <sup>a</sup>		Average annual rate of inflation <sup>a</sup> (percent)		Life expectancy birth (years) 1990	Adult illiteracy (percent)	
			Dollars 1990	Average annual growth rate (percent) 1965-90	1965-80	1980-90		Female 1990	Total 1990
<b>Low-income economies</b>	<b>3,058.3 t</b>	<b>37,780 t</b>	<b>350 w</b>	<b>2.9 w</b>	<b>8.0 w</b>	<b>9.6 w</b>	<b>62 w</b>	<b>52 w</b>	<b>40 w</b>
<b>China and India</b>	<b>1,983.2 t</b>	<b>12,849 t</b>	<b>360 w</b>	<b>3.7 w</b>	<b>3.2 w</b>	<b>6.8 w</b>	<b>65 w</b>	<b>50 w</b>	<b>37 w</b>
<b>Other low-income</b>	<b>1,075.1 t</b>	<b>24,931 t</b>	<b>320 w</b>	<b>1.7 w</b>	<b>17.3 w</b>	<b>15.1 w</b>	<b>55 w</b>	<b>56 w</b>	<b>45 w</b>
1 Mozambique	15.7	802	80	..	..	36.6	47	79	67
2 Tanzania <sup>b</sup>	24.5	945	110	-0.2	9.6	25.8	48	..	..
3 Ethiopia	51.2	1,222	120	-0.2	3.4	2.1	48	..	..
4 Somalia	7.8	638	120	-0.1	10.2	49.7	48	86	76
5 Nepal	18.9	141	170	0.5	7.8	9.1	52	87	74
6 Chad	5.7	1,284	190	-1.1	6.2	1.2	47	82	70
7 Bhutan	1.4	47	190	..	..	8.4	49	75	62
8 Lao PDR	4.1	237	200	..	..	..	49	..	..
9 Malawi	8.5	118	200	0.9	7.4	14.7	46	..	..
10 Bangladesh	106.7	144	210	0.7	15.9	9.6	52	78	65
11 Burundi	5.4	28	210	3.4	5.0	4.2	47	60	50
12 Zaire	37.3	2,345	220	-2.2	24.7	60.9	52	39	28
13 Uganda	16.3	236	220	-2.4	21.4	107.0	47	65	52
14 Madagascar	11.7	587	230	-1.9	7.7	17.1	51	27	20
15 Sierra Leone	4.1	72	240	0.0	7.9	56.1	42	89	79

Aus: World Development Report 1992, World Bank

Wer möchte anfangen, zu interpretieren?

# UND NUN ETWAS ÜBERSICHTLICHER



Aus: Data exploration using Self-Organizing Maps, S. Kaski. Daten: World Development Report 1992, World Bank

# GRUNDLAGEN DER SELF-ORGANIZING MAP

# EINLEITUNG

- ▶ **Self-Organizing Maps (SOMs)**, benannt nach Erfinder Prof. Teuvo Kohonen



Teuvo Kohonen

# EINLEITUNG

- ▶ **Self-Organizing Maps (SOMs)**, benannt nach Erfinder Prof. Teuvo Kohonen
- ▶ Neuronale Netze mit **unüberwachtem Lernen**



Teuvo Kohonen

# EINLEITUNG

- ▶ **Self-Organizing Maps (SOMs)**, benannt nach Erfinder Prof. Teuvo Kohonen
- ▶ Neuronale Netze mit **unüberwachtem Lernen**
- ▶ Ziel: „Automatisch“ **hochdimensionale** Datensätze auf **niedrigdimensionale Karte** abbilden



Teuvo Kohonen

# EINLEITUNG

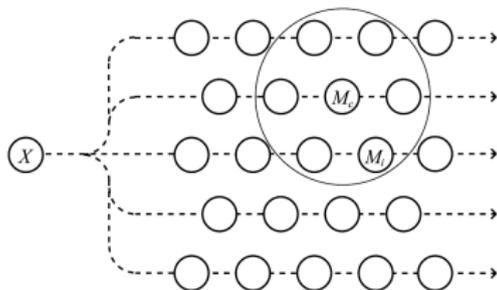
- ▶ **Self-Organizing Maps (SOMs)**, benannt nach Erfinder Prof. Teuvo Kohonen
- ▶ Neuronale Netze mit **unüberwachtem Lernen**
- ▶ Ziel: „Automatisch“ **hochdimensionale** Datensätze auf **niedrigdimensionale Karte** abbilden
- ▶ Einsatzgebiete:
  - ▶ Mustererkennung (Handschrift, Sprache, Gesichter)
  - ▶ Robotik
  - ▶ Medizinische Diagnostik
  - ▶ Technische Anlagensteuerung



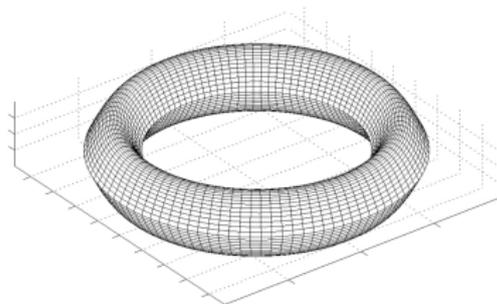
Teuvo Kohonen

# AUFBAU

- ▶ Anordnung von **Neuronen** in niedrigdimensionalem Raster (üblich: 2D)
- ▶ **Verschiedene Topologien** möglich (u.a. hexagonal, toroidal)



Aus: Self-Organizing Maps, T. Kohonen, 2001



Aus: Player Modeling using Self-Organization in Tomb Raider: Underworld, A. Drachen et al.

# INTERPRETATION DER NEURONEN

- ▶ Neuron haben Position in zwei Räumen:

# INTERPRETATION DER NEURONEN

- ▶ Neuron haben Position in zwei Räumen:
  - ▶ Im **niedrigdimensionalen Raum** der Ausgabewerte: Position auf der Karte

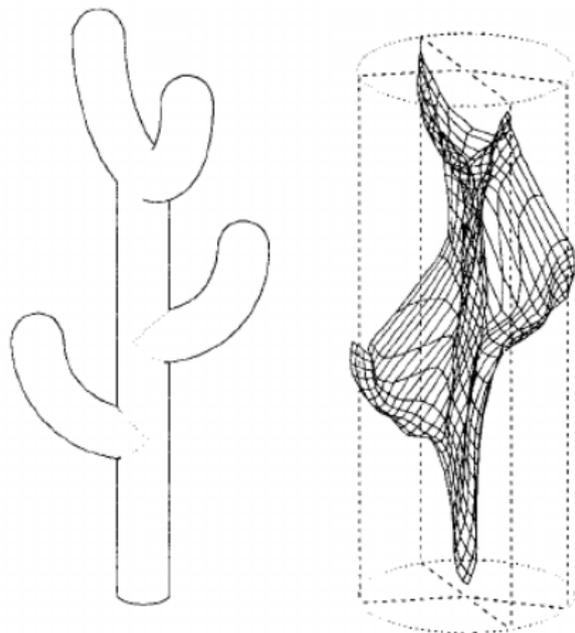
# INTERPRETATION DER NEURONEN

- ▶ Neuron haben Position in zwei Räumen:
  - ▶ Im **niedrigdimensionalen Raum** der Ausgabewerte: Position auf der Karte
  - ▶ Im **hochdimensionalen Raum** der Eingabewerte:  
**Gewichtungsvektor**

# INTERPRETATION DER NEURONEN

- ▶ Neuron haben Position in zwei Räumen:
  - ▶ Im **niedrigdimensionalen Raum** der Ausgabewerte: Position auf der Karte
  - ▶ Im **hochdimensionalen Raum** der Eingabewerte:  
**Gewichtungsvektor**
- ▶ Neuron beschreibt Abbildung eines Punktes aus hoch- in niedrigdimensionalen Raum

# INTERPRETATION DER NEURONEN - BEISPIEL



Aus: Self-Organizing Maps, T. Kohonen, 2001

- ▶ Abbildung 3D  $\rightarrow$  2D
- ▶ Uniformverteilte Werte in der Form eines Kaktus
- ▶ Links: Dichtefunktion der Daten (nur 1 oder 0)
- ▶ Rechts: Neuronenraster der SOM in Netzdarstellung

## ZWEI OPERATIONSMODI

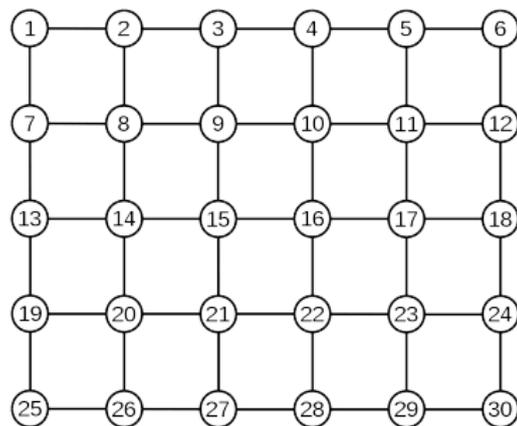
- ▶ Zwei Operationsmodi für SOMs:
  - ▶ **Klassifikation**: Eingabevektor wird auf Karte abgebildet
  - ▶ **Lernen**: Die Neuronen passen sich an Trainingswerte an

## ZWEI OPERATIONSMODI

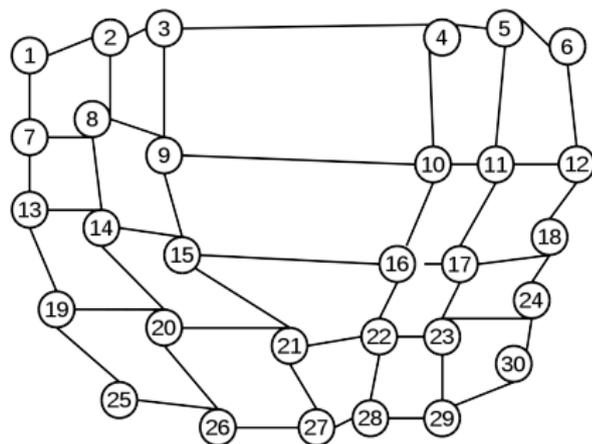
- ▶ Zwei Operationsmodi für SOMs:
  - ▶ **Klassifikation**: Eingabevektor wird auf Karte abgebildet
  - ▶ **Lernen**: Die Neuronen passen sich an Trainingswerte an
- ▶ Klassifikation macht nur auf angelernter SOM Sinn!

# KLASSIFIKATION

## KLASSIFIKATION - BEISPIEL

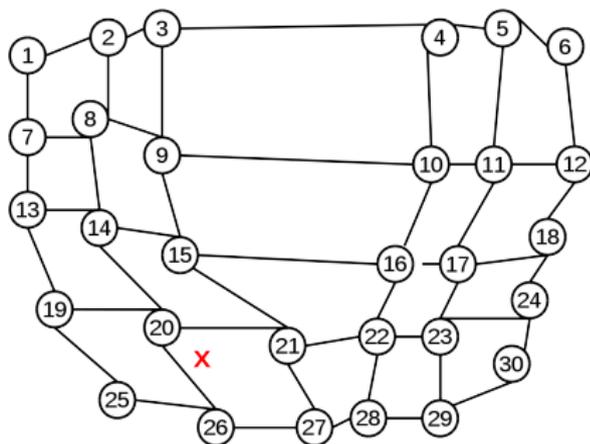
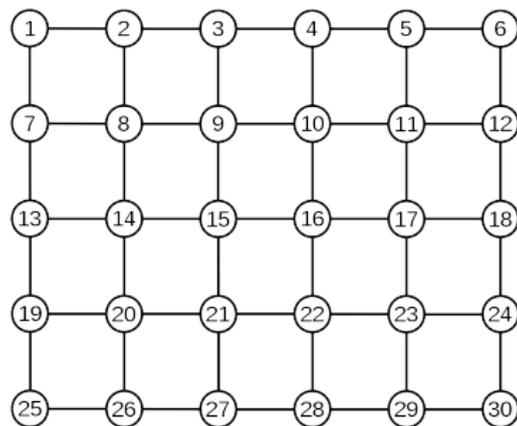


- ▶ Links: Neuronenraster (Karte) einer SOM mit  $6 \times 5$  Neuronen



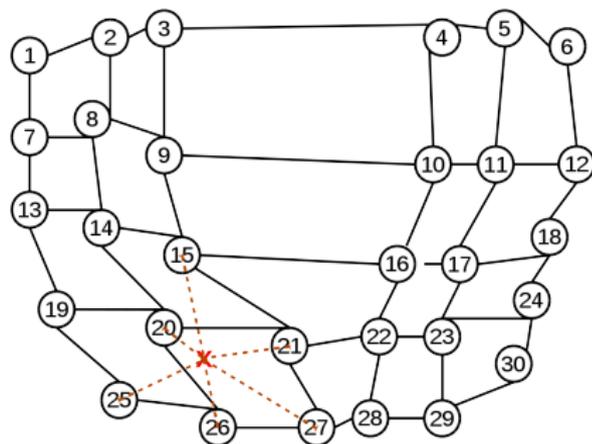
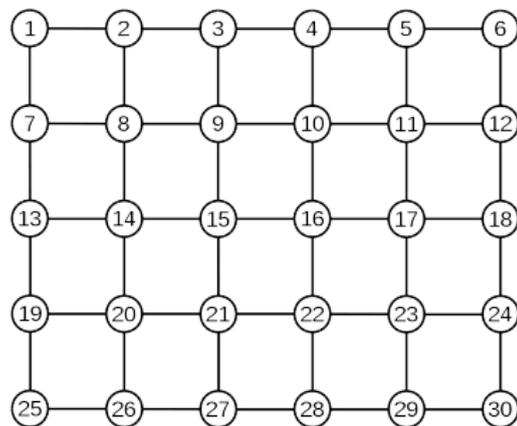
- ▶ Rechts: Neuronen im Eingaberaum

## KLASSIFIKATION - BEISPIEL



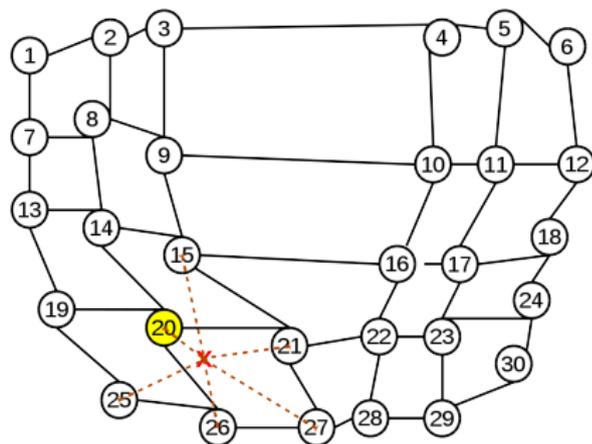
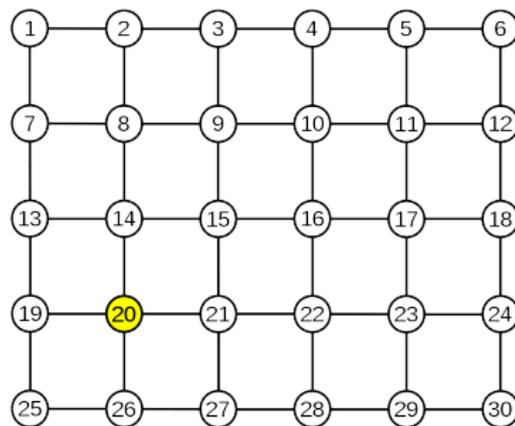
- Wir wollen einen Eingabevektor  $\vec{x}$  klassifizieren

# KLASSIFIKATION - BEISPIEL



- Jedes Neuron  $i$  berechnet Distanz  $d(\vec{x}, \vec{m}_i)$  vom Gewichtungsvektor  $\vec{m}_i$  zu  $\vec{x}$

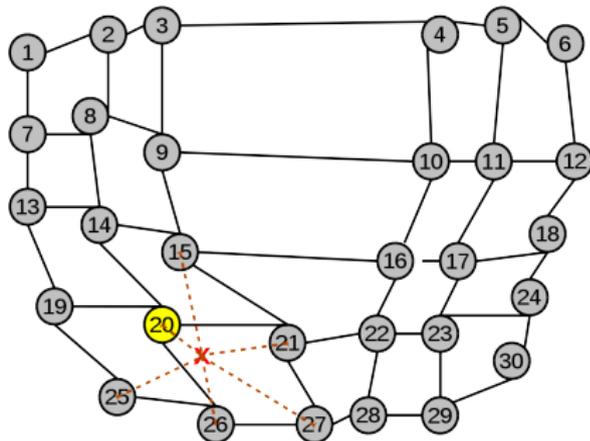
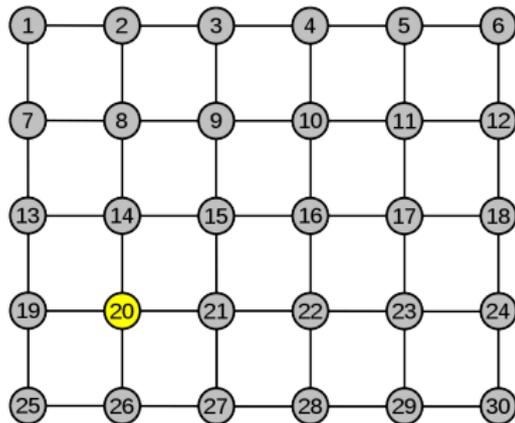
# KLASSIFIKATION - BEISPIEL



- Neuron mit geringster Distanz ist Gewinner:

$$c = \arg \min_i \{d(\vec{x}, \vec{m}_i)\}$$

# KLASSIFIKATION - BEISPIEL



- ▶ Alle anderen Neuronen deaktivieren sich
- ▶ Das Gewinnerneuron ist als einziges noch aktiv

LERNEN

# STEPWISE APPROXIMATION

- ▶ Ursprünglicher Lernalgorithmus für die SOM
- ▶ Mehrere **Lernschritte** ( $t = 0, 1, 2, \dots$ ) mit je einem Trainingsvektor

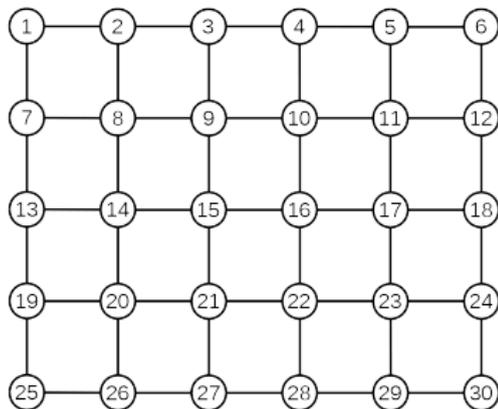
# STEPWISE APPROXIMATION

- ▶ Ursprünglicher Lernalgorithmus für die SOM
- ▶ Mehrere **Lernschritte** ( $t = 0, 1, 2, \dots$ ) mit je einem Trainingsvektor
- ▶ Schnellere, moderne Variante: Batch Learning

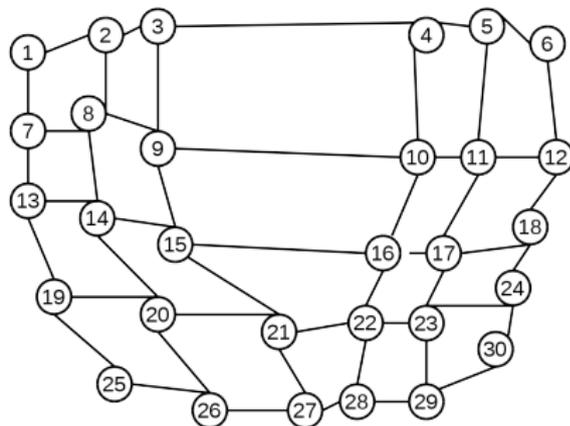
# STEPWISE APPROXIMATION

- ▶ Ursprünglicher Lernalgorithmus für die SOM
- ▶ Mehrere **Lernschritte** ( $t = 0, 1, 2, \dots$ ) mit je einem Trainingsvektor
- ▶ Schnellere, moderne Variante: Batch Learning
- ▶ Grundlage: **Competitive Learning**

## STEPWISE APPROXIMATION - BEISPIEL

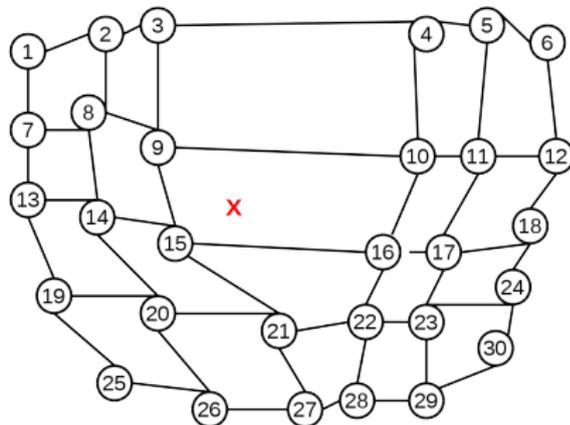
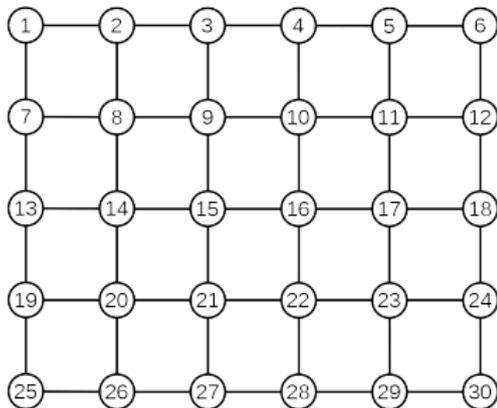


- ▶ Links: Neuronenraster (Karte) einer SOM mit  $6 \times 5$  Neuronen



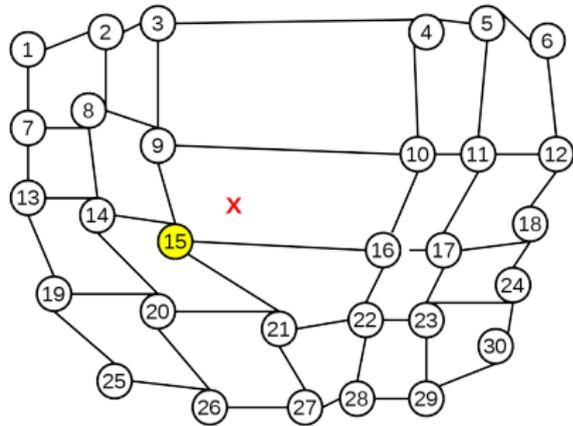
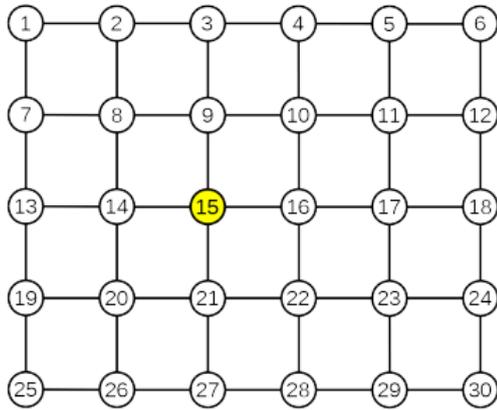
- ▶ Rechts: Neuronen im Eingaberaum

## STEPWISE APPROXIMATION - BEISPIEL



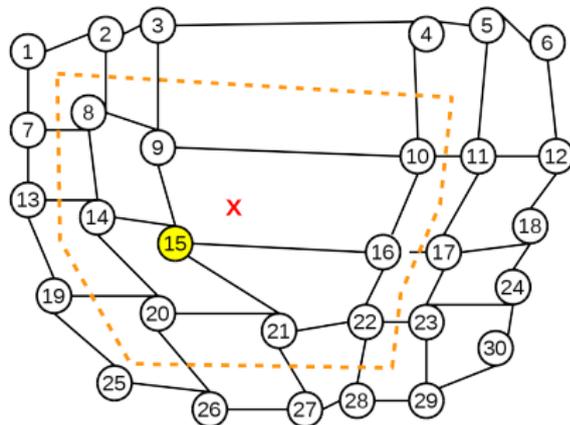
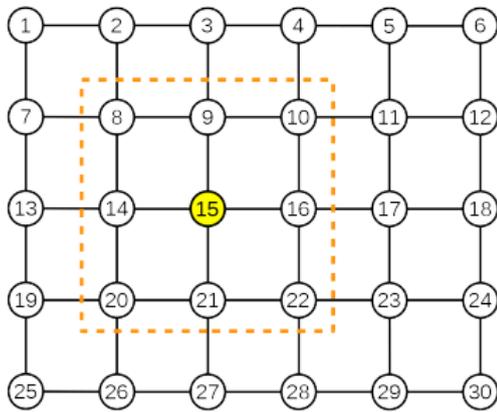
- Wir füttern der SOM einen Trainingsvektor  $\vec{x}(t = 0)$

## STEPWISE APPROXIMATION - BEISPIEL



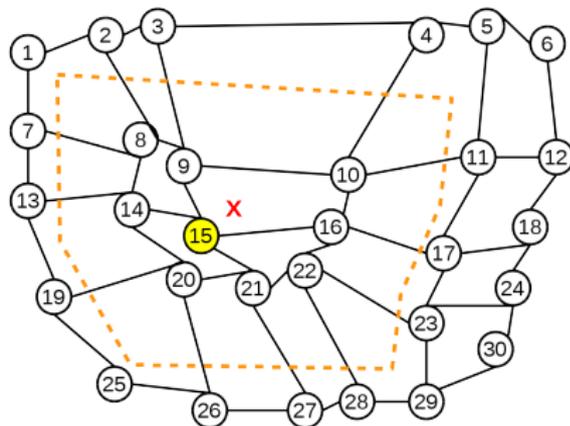
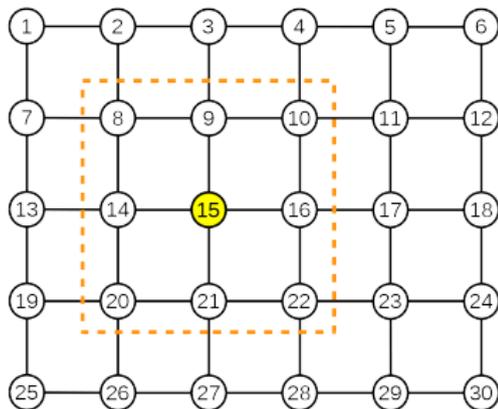
- Gewinnerneuron wird wie üblich ermittelt

## STEPWISE APPROXIMATION - BEISPIEL



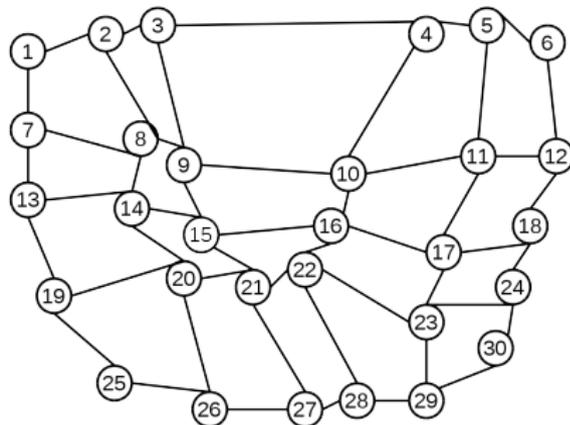
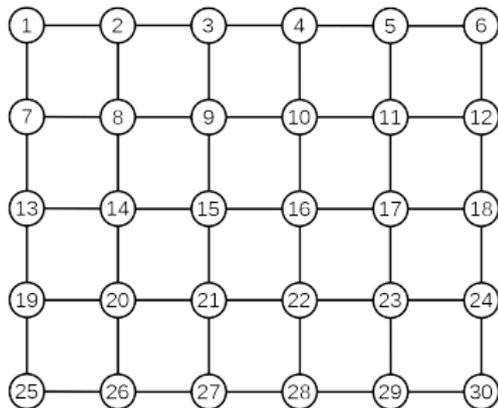
- Nachbarschaft des Gewinners wird betrachtet (orange)

## STEPWISE APPROXIMATION - BEISPIEL



- ▶ Trainingsvektor zieht Neuronen der Nachbarschaft zu sich
- ▶ Gewichtungsvektoren  $\vec{m}_i$  haben sich geändert, sonst nichts!

## STEPWISE APPROXIMATION - BEISPIEL



- ▶ Stabiler Zustand der Gewichtungsvektoren  $\vec{m}_i(t = 1)$

## STEPWISE APPROXIMATION - UPDATE-FORMEL

$$\vec{m}_j(t+1) = \vec{m}_j(t) + h_{ij}(t) \cdot \alpha(t) \cdot (\vec{x}(t) - \vec{m}_j(t))$$

- ▶  $\vec{m}_j(t)$  und  $\vec{m}_j(t+1)$ : Alter und neuer Gewichtungsvektor des Neurons  $j$

## STEPWISE APPROXIMATION - UPDATE-FORMEL

$$\vec{m}_j(t+1) = \vec{m}_j(t) + h_{ij}(t) \cdot \alpha(t) \cdot (\vec{x}(t) - \vec{m}_j(t))$$

- ▶  $\vec{m}_j(t)$  und  $\vec{m}_j(t+1)$ : Alter und neuer Gewichtungsvektor des Neurons  $j$
- ▶ **Lernrate  $\alpha(t)$** : Um anfangs stärker zu lernen als später

# STEPWISE APPROXIMATION - UPDATE-FORMEL

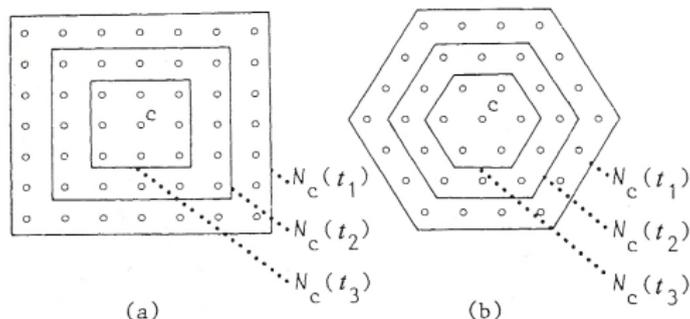
$$\vec{m}_j(t+1) = \vec{m}_j(t) + h_{ij}(t) \cdot \alpha(t) \cdot (\vec{x}(t) - \vec{m}_j(t))$$

- ▶  $\vec{m}_j(t)$  und  $\vec{m}_j(t+1)$ : Alter und neuer Gewichtungsvektor des Neurons  $j$
- ▶ **Lernrate  $\alpha(t)$** : Um anfangs stärker zu lernen als später
- ▶ **Nachbarschaftsfunktion  $h_{ij}(t)$** : Wie viele Nachbarn werden „mitgezogen“?

## STEPWISE APPROXIMATION - UPDATE-FORMEL

$$\vec{m}_j(t+1) = \vec{m}_j(t) + h_{ij}(t) \cdot \alpha(t) \cdot (\vec{x}(t) - \vec{m}_j(t))$$

- ▶  $\vec{m}_j(t)$  und  $\vec{m}_j(t+1)$ : Alter und neuer Gewichtungsvektor des Neurons  $j$
- ▶ **Lernrate**  $\alpha(t)$ : Um anfangs stärker zu lernen als später
- ▶ **Nachbarschaftsfunktion**  $h_{ij}(t)$ : Wie viele Nachbarn werden „mitgezogen“?



# LOKALE UND GLOBALE ORDNUNG

- ▶ Wichtige Eigenschaft der SOM: Sie **bewahrt Topologie** der Daten
- ▶ Zwei Arten von Ordnung werden beibehalten:

# LOKALE UND GLOBALE ORDNUNG

- ▶ Wichtige Eigenschaft der SOM: Sie **bewahrt Topologie** der Daten
- ▶ Zwei Arten von Ordnung werden beibehalten:
  - ▶ **Lokale Ordnung**: Ähnliche Daten stehen auch auf Karte nah beieinander

# LOKALE UND GLOBALE ORDNUNG

- ▶ Wichtige Eigenschaft der SOM: Sie **bewahrt Topologie** der Daten
- ▶ Zwei Arten von Ordnung werden beibehalten:
  - ▶ **Lokale Ordnung**: Ähnliche Daten stehen auch auf Karte nah beieinander
  - ▶ **Globale Ordnung**: Bereiche unterschiedlicher Daten liegen auch auf Karte weit auseinander

# LOKALE UND GLOBALE ORDNUNG

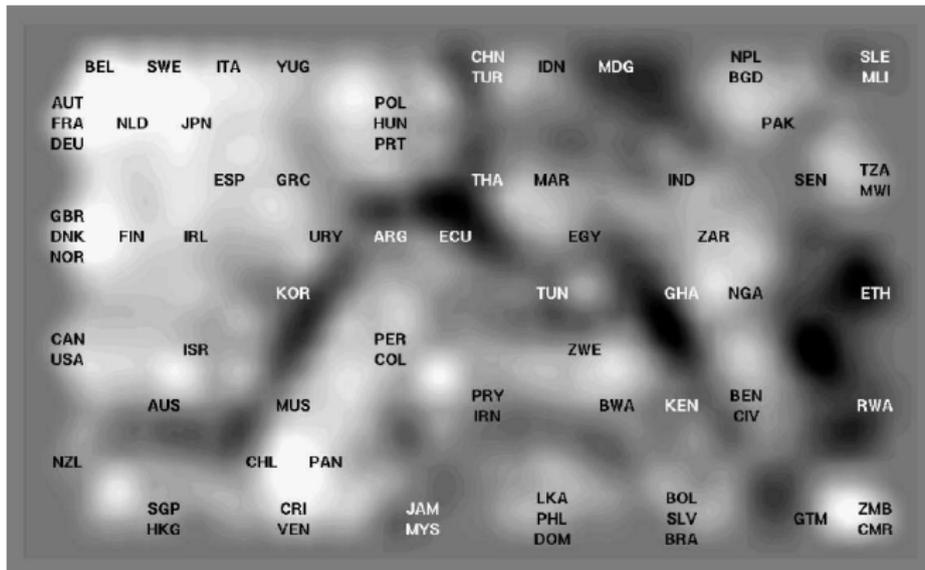
- ▶ Wichtige Eigenschaft der SOM: Sie **bewahrt Topologie** der Daten
- ▶ Zwei Arten von Ordnung werden beibehalten:
  - ▶ **Lokale Ordnung**: Ähnliche Daten stehen auch auf Karte nah beieinander
  - ▶ **Globale Ordnung**: Bereiche unterschiedlicher Daten liegen auch auf Karte weit auseinander
- ▶ Einfluss der Nachbarschaftsfunktion:
  - ▶ Zu breit: Zu viele unterschiedliche Neuronen werden mitgezogen → **keine globale Ordnung!**

# LOKALE UND GLOBALE ORDNUNG

- ▶ Wichtige Eigenschaft der SOM: Sie **bewahrt Topologie** der Daten
- ▶ Zwei Arten von Ordnung werden beibehalten:
  - ▶ **Lokale Ordnung**: Ähnliche Daten stehen auch auf Karte nah beieinander
  - ▶ **Globale Ordnung**: Bereiche unterschiedlicher Daten liegen auch auf Karte weit auseinander
- ▶ Einfluss der Nachbarschaftsfunktion:
  - ▶ Zu breit: Zu viele unterschiedliche Neuronen werden mitgezogen → **keine globale Ordnung!**
  - ▶ Zu schmal: Ähnliche Neuronen wachsen nicht zusammen → **keine lokale Ordnung!**

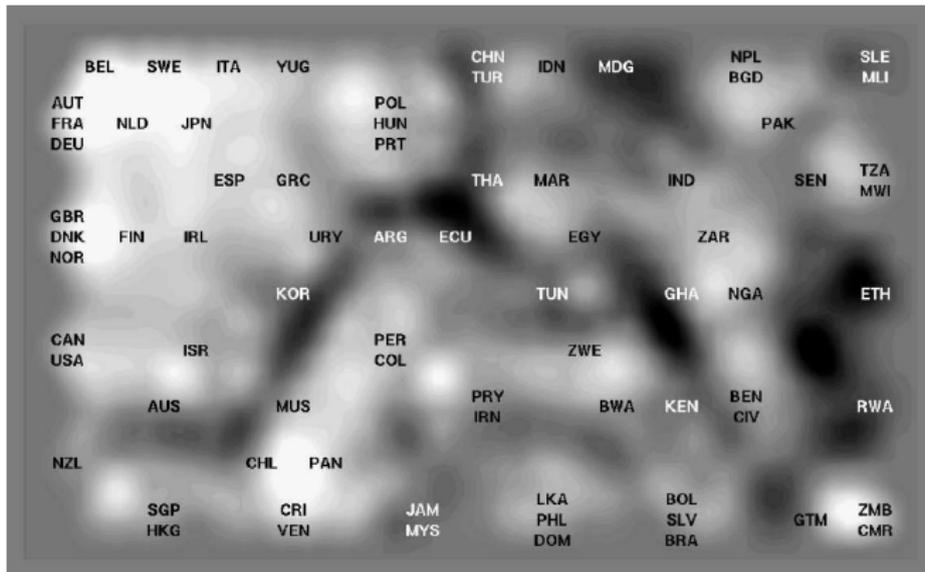
U-MATRIX

# U-MATRIX - BEISPIEL



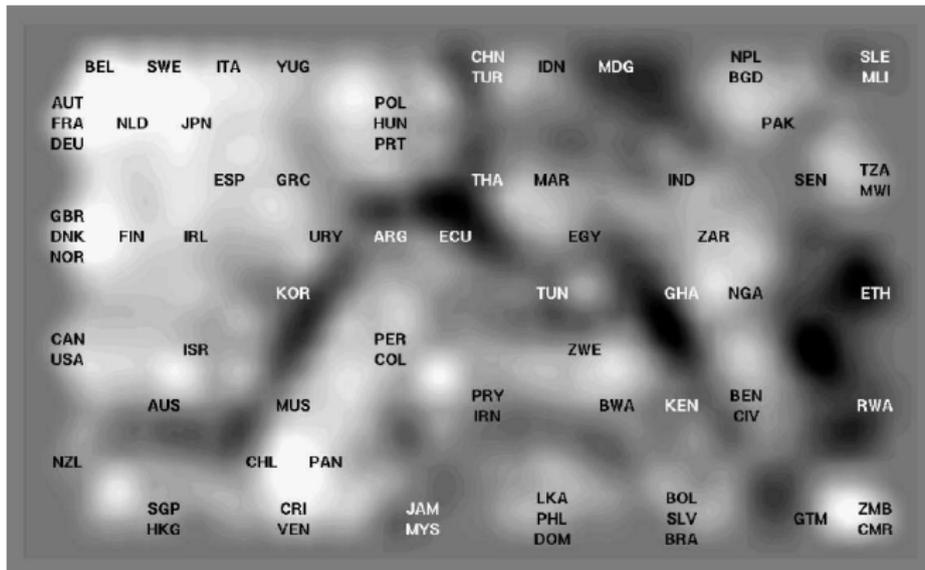
- U-Matrix der „Poverty SOM“,  $13 \times 9$  Neuronen, hexagonal

# U-MATRIX - BEISPIEL



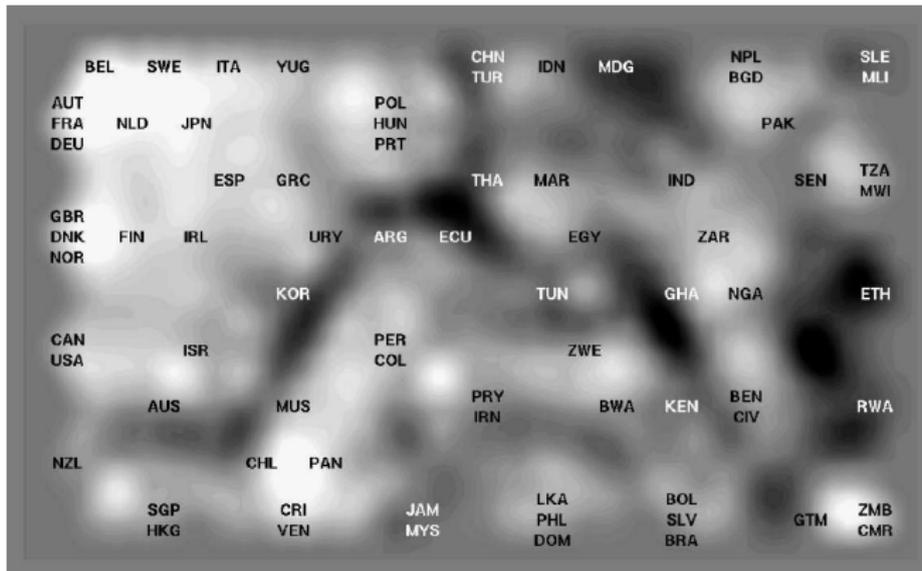
- Farblich kodiert: Mittlere Abweichung jedes Neurons zu seinen Nachbarn

# U-MATRIX - BEISPIEL



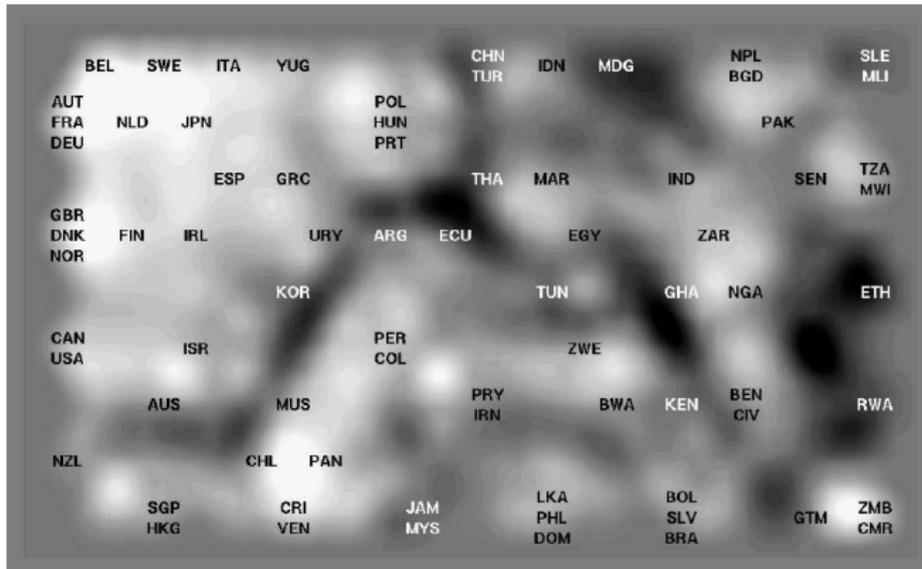
- Cluster („Täler“) und Clustergrenzen („Hügel“) erkennbar

# U-MATRIX - BEISPIEL



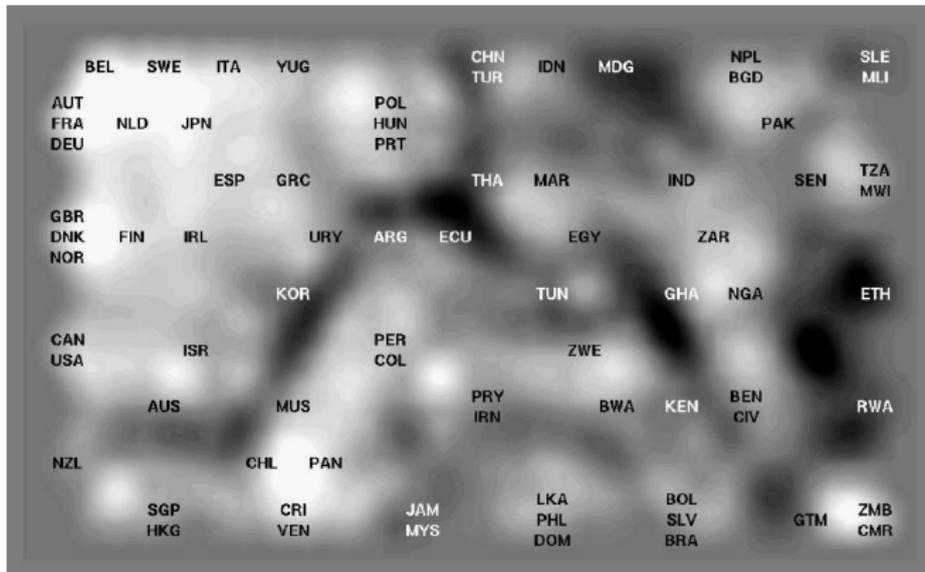
- Beobachtung: Kartenposition korreliert grob mit geographischer Position

# U-MATRIX - BEISPIEL



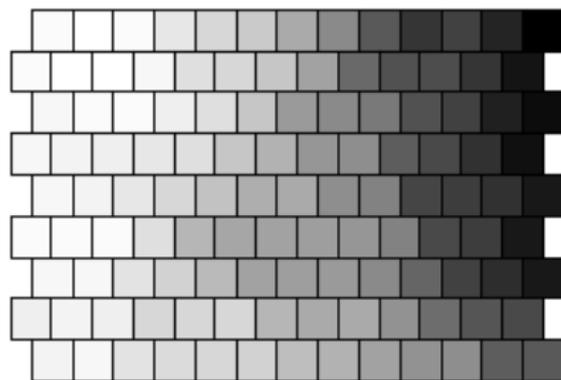
- ▶ Cluster oben links: OECD-Länder, rechts davon: Osteuropa

# U-MATRIX - BEISPIEL

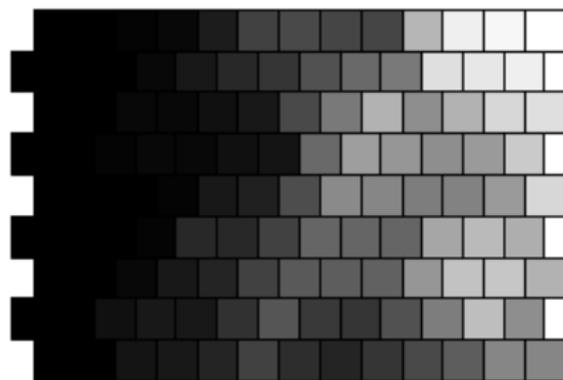


- ▶ Cluster unten: Südamerikanische Staaten, durch Gebirge getrennt

# KOMPONENTENEbenen



(a)



(b)

Aus: Data exploration using Self-Organizing Maps, S. Kaski

- ▶ Lebenserwartung ab Geburt (a)
- ▶ Analphabetenrate unter den Erwachsenen (b)

# PLAYER MODELING

# IDEE

- ▶ SOMs für Problemstellungen des **Game Designs!**

# IDEE

- ▶ SOMs für Problemstellungen des **Game Designs!**
- ▶ Ziel: Spielertypen in *Tomb Raider: Underworld* identifizieren
  - ▶ Ziel des Spiels: Rätsel in verschiedenen Levels lösen
  - ▶ Als *Lara Croft* Hindernisse überwinden

# IDEE

- ▶ SOMs für Problemstellungen des **Game Designs!**
- ▶ Ziel: Spielertypen in ***Tomb Raider: Underworld*** identifizieren
  - ▶ Ziel des Spiels: Rätsel in verschiedenen Levels lösen
  - ▶ Als *Lara Croft* Hindernisse überwinden
- ▶ Gesammelt: **Repräsentative Daten von 1365 Spielern**, die das Spiel von Anfang bis Ende durchspielten



Aus: Player Modeling using Self-Organization in Tomb Raider: Underworld, A. Drachen et al.

# GESAMMELTE DATEN

- ▶ **Insgesamt 6 Merkmale** wurden betrachtet
- ▶ **Gesamtanzahl an Toden**

# GESAMMELTE DATEN

- ▶ **Insgesamt 6 Merkmale** wurden betrachtet
- ▶ **Gesamtanzahl an Toden**
- ▶ Relativer Anteil verschiedener **Todesursachen**:

# GESAMMELTE DATEN

- ▶ **Insgesamt 6 Merkmale** wurden betrachtet
- ▶ **Gesamtanzahl an Toden**
- ▶ Relativer Anteil verschiedener **Todesursachen**:
  - ▶ Tod **durch gegnerischen NPC**
  - ▶ Tod **durch Umgebung** (Ertrinken, Verbrennen, ...)
  - ▶ Tod **durch Fallschaden** (bei gescheiterten Sprüngen)

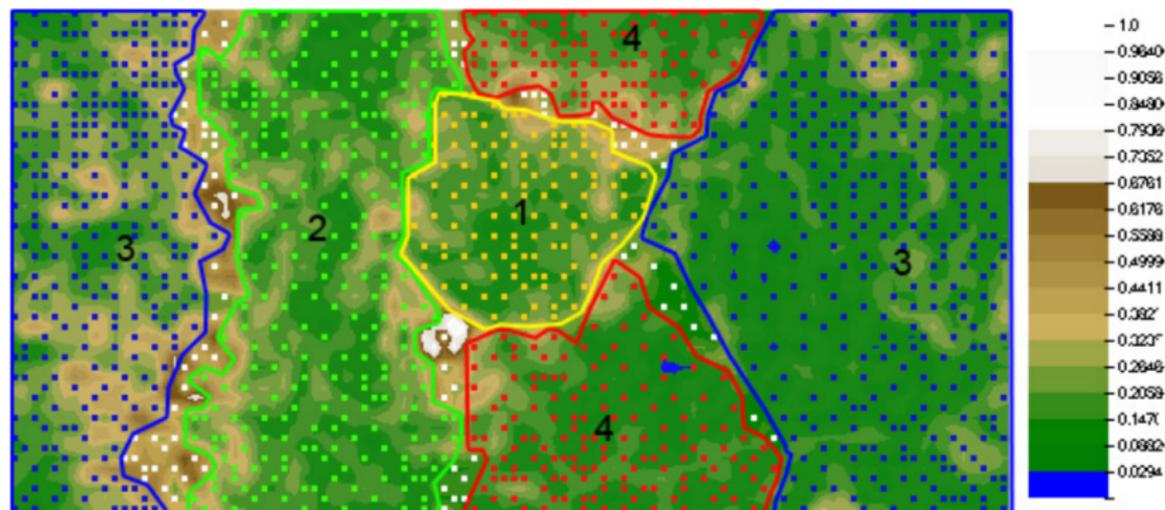
# GESAMMELTE DATEN

- ▶ **Insgesamt 6 Merkmale** wurden betrachtet
- ▶ **Gesamtanzahl an Toden**
- ▶ Relativer Anteil verschiedener **Todesursachen**:
  - ▶ Tod **durch gegnerischen NPC**
  - ▶ Tod **durch Umgebung** (Ertrinken, Verbrennen, ...)
  - ▶ Tod **durch Fallschaden** (bei gescheiterten Sprüngen)
- ▶ **Benötigte Spielzeit** zum Durchspielen

# GESAMMELTE DATEN

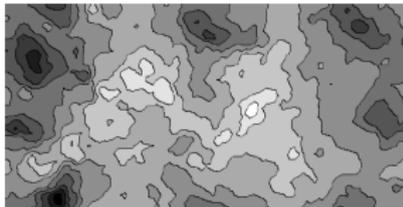
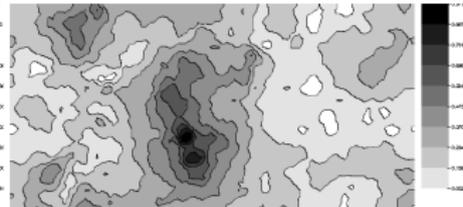
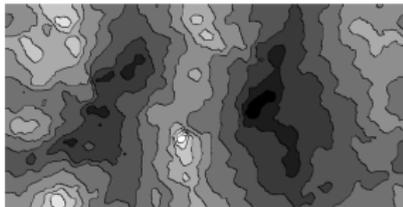
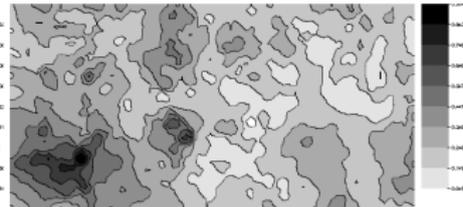
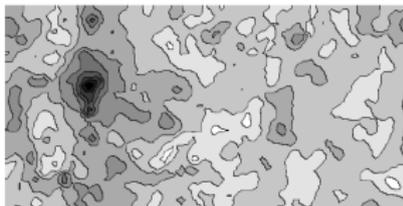
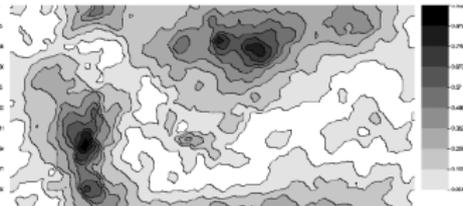
- ▶ **Insgesamt 6 Merkmale** wurden betrachtet
- ▶ **Gesamtanzahl an Toden**
- ▶ Relativer Anteil verschiedener **Todesursachen**:
  - ▶ Tod **durch gegnerischen NPC**
  - ▶ Tod **durch Umgebung** (Ertrinken, Verbrennen, ...)
  - ▶ Tod **durch Fallschaden** (bei gescheiterten Sprüngen)
- ▶ **Benötigte Spielzeit** zum Durchspielen
- ▶ Anzahl Aufrufe **Help-on-Demand**

## ERGEBNIS



Aus: Player Modeling using Self-Organization in Tomb Raider: Underworld, A. Drachen et al.

# ERGEBNIS - KOMPONENTENEbenen

(a) Cause of Death: Opponent,  $D_o$ (b) Cause of Death: Environment,  $D_e$ (c) Cause of Death: Falling,  $D_f$ (d) Number of Deaths,  $D$ (e) Completion Time,  $C$ (f) Help on Demand,  $H$ 

Aus: Player Modeling using Self-Organization in Tomb Raider: Underworld, A. Drachen et al.

# ERGEBNIS - VERSUCH DER KLASSIFIKATION

- ▶ Anhand der Cluster und der Komponentenebenen wurden **4 Spielertypen** charakterisiert

Tabelle: Spielertypen in Tomb Raider: Underworld

Titel	Tode	Ursache	HOD	Durchgespielt
Veteranen	wenige	Umgebung	eher wenig	schnell
Rätsellöser	viele	Fallschaden	kaum	langsam
Pazifisten	varriert	Gegner	wenig	schnell
Läufer	viele	Gegner/Umgebung	variiert	sehr schnell

## DISKUSSION & BEWERTUNG

- ▶ Gesammelt wurden Daten zu nur 6 Merkmalen
  - ▶ Daten **nicht hochdimensional** → SOMs sind Overkill!
  - ▶ Wieso nicht gleich mehr Daten gesammelt? Möglichkeit war vorhanden!

## DISKUSSION & BEWERTUNG

- ▶ Gesammelt wurden Daten zu nur 6 Merkmalen
  - ▶ Daten **nicht hochdimensional** → SOMs sind Overkill!
  - ▶ Wieso nicht gleich mehr Daten gesammelt? Möglichkeit war vorhanden!
- ▶ **Einfache und natürliche Visualisierung** → gute Diskussionsgrundlage für Game Designer!

## DISKUSSION & BEWERTUNG

- ▶ Gesammelt wurden Daten zu nur 6 Merkmalen
  - ▶ Daten **nicht hochdimensional** → SOMs sind Overkill!
  - ▶ Wieso nicht gleich mehr Daten gesammelt? Möglichkeit war vorhanden!
- ▶ **Einfache und natürliche Visualisierung** → gute Diskussionsgrundlage für Game Designer!
- ▶ SOM kann automatisch neu gebaut werden → **interessant für iterative Entwicklung!**

## DISKUSSION & BEWERTUNG

- ▶ Gesammelt wurden Daten zu nur 6 Merkmalen
  - ▶ Daten **nicht hochdimensional** → SOMs sind Overkill!
  - ▶ Wieso nicht gleich mehr Daten gesammelt? Möglichkeit war vorhanden!
- ▶ **Einfache und natürliche Visualisierung** → gute Diskussionsgrundlage für Game Designer!
- ▶ SOM kann automatisch neu gebaut werden → **interessant für iterative Entwicklung!**
- ▶ Durch **hervorragende Skalierbarkeit** auch komplexe Zusammenhänge erkennbar!

# Danke für die Aufmerksamkeit

(falls vorhanden)