JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

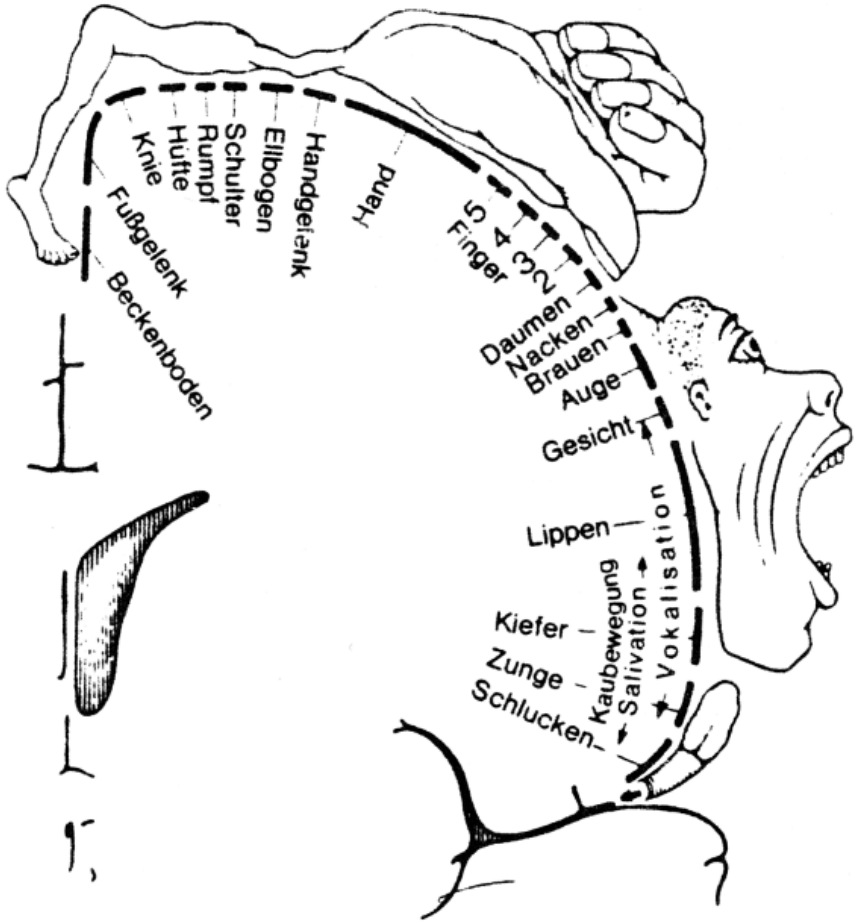
www.uni-frankfurt.de

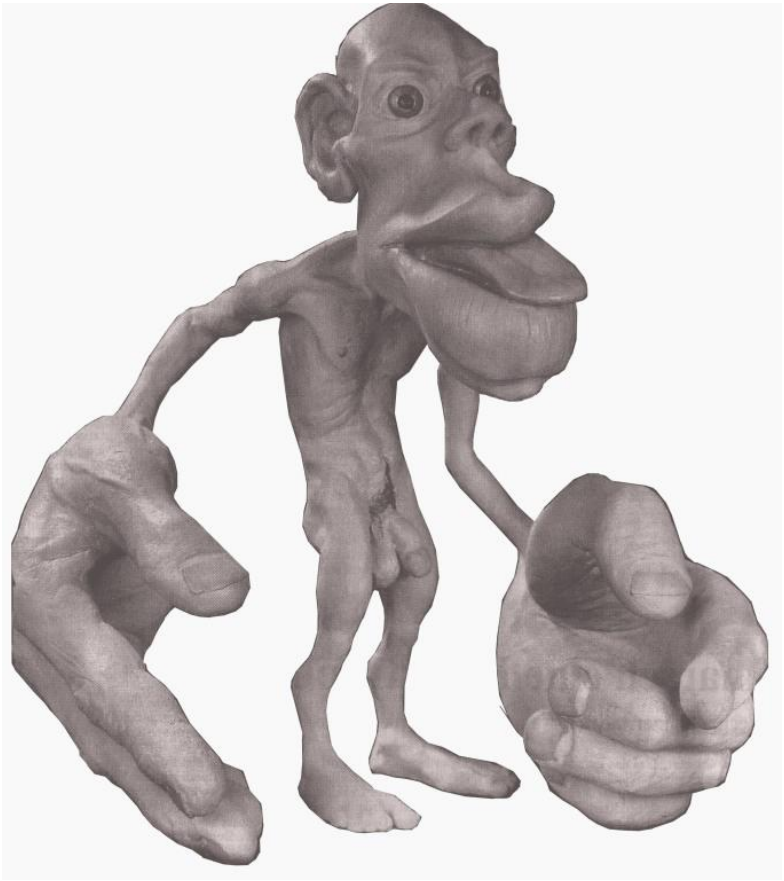
Merkmalskarten und Kohonenetze

Praktikum Adaptive Systeme

Somatotopie

- **Beobachtung: Abbilder, Karten**





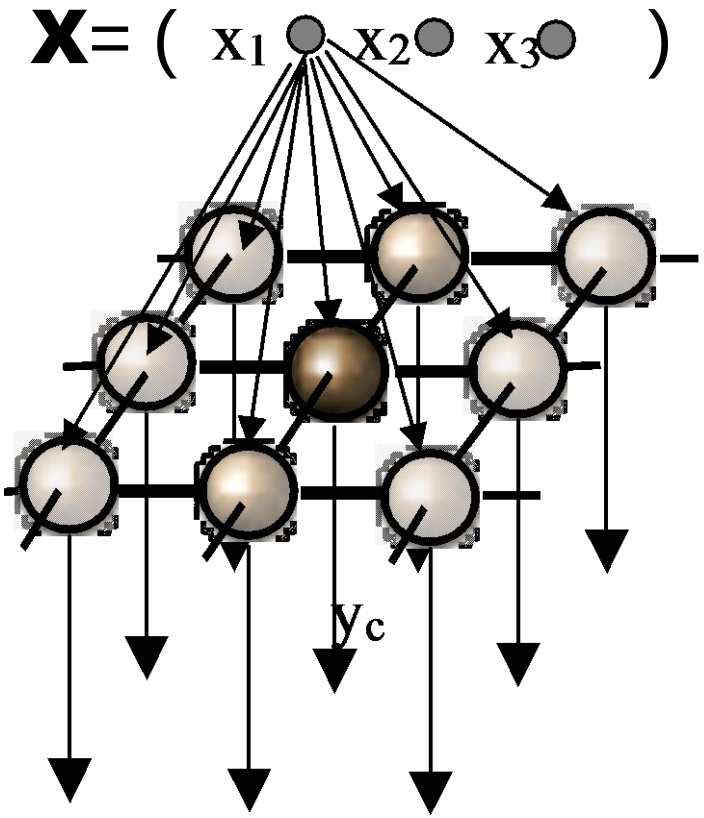
Kohonenetze-Topologie

Eingabedimension n = 3

Ausgabedimension d = 2

Def. „Nachbarschaft“

von Neuronen



Lernalgorithmus Kohonenkarten

JOHANN WOLFGANG GOETHE UNIVERSITÄT FRANKFURT AM MAIN

1. **Suche Neuron** (Gewichtsvektor **w** an Stelle **p**) **mit kleinstem Abstand zur Eingabe x**

$$\left|\mathbf{x}-\mathbf{w}_{\mathbf{p}}\right| = \min_{k}\left|\mathbf{x}-\mathbf{w}_{\mathbf{k}}\right|$$

Winner Take All

2. **Adaptiere die Gewichte**

$$\mathbf{w}_{\mathbf{p}}(t+1) = \mathbf{w}_{\mathbf{p}}(t) + \gamma(t)\,[\mathbf{x} - \mathbf{w}_{\mathbf{p}}(t)]$$

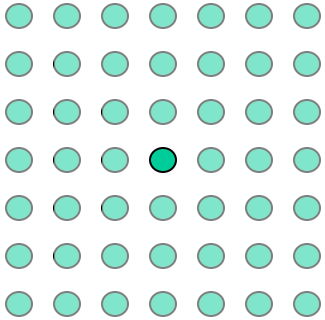
3. **Adaptiere auch die nächsten Nachbarn**

$$\mathbf{w}_{\mathbf{k}}(t+1) = \mathbf{w}_{\mathbf{k}}(t) + \gamma(t)\, h(\mathbf{p},\mathbf{k},t)\,[\mathbf{x} - \mathbf{w}_{\mathbf{k}}(t)]$$

z.B. mit $h(\mathbf{p},\mathbf{k},t) := \begin{cases} 1 & \text{wenn Neuron } \mathbf{k} \text{ aus der Nachbarschaft von } \mathbf{p} \text{ ist} \\ 0 & \text{sonst} \end{cases}$

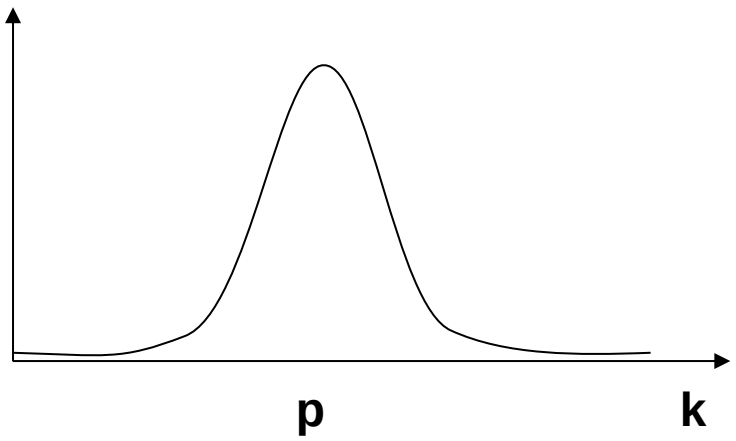
Nachbarschaftsfunktionen

- **Binäre Funktion: zeitabhängig**



- **Glockenfunktion**

$h(\mathbf{p},\mathbf{k},t) = \exp(\mathbf{p}-\mathbf{k},\sigma(t))$



Pseudocode für Kohonenkarten

TOPO: (* topologie-erhaltende Abbildung auf ein m×m Gitter *)

```
VAR x:   ARRAY [1..n] OF REAL; (* Muster*)
    w:   ARRAY[1..m,1..m] OF ARRAY [1..n] OF REAL; (* Gewichte*)
    p:   RECORD i,j : INTEGER END; (* selektiertes Neuron an Position p *)
 BEGIN
   σ := σmax                              (* initial: max. Nachbarschaft *)
   FOR t:=1 TO tmax DO
    Read(PatternFile,x)                          (* Muster erzeugen oder einlesen *)
    (* Neuron selektieren *)
    Min:= ABS(x-w[1,1])                          (* initialer Wert *)
    FOR i:=1 TO m DO                             (* In allen Spalten *)
       FOR j:= 1 TO m DO                         (* und Zeilen *)
           Abstand:= ABS(x-w[i,j]);    (* suche Minimum *)
           IF Abstand < Min
              THEN Min:=Abstand; p.i:=i; p.j:=j;
           END
       END
    END
    (* Gewichte adaptieren *)
    FOR i:= p.i-σ TO p.i+σ DO                    (* Evaluiere 2-dim. *)
      FOR j:= p.j-σ TO p.j+σ DO                  (* Nachbarschaft um c*)
       IF i>0 AND i<= m  AND j>0  AND  j<=m THEN  (* Vorsicht am Rand *)
              w[i,j] = w[i,j] + 1.0/FLOAT( t)*( x-w[i,j])
       END
      END
    END (*i*)
    GrafikAnzeige(t,w)                  (* Visualisierung der Iteration *)
    upDate(σ,t)                         (* Verkleinerung des Nachbarschaftsradius σ *)
   END (*t*)
```

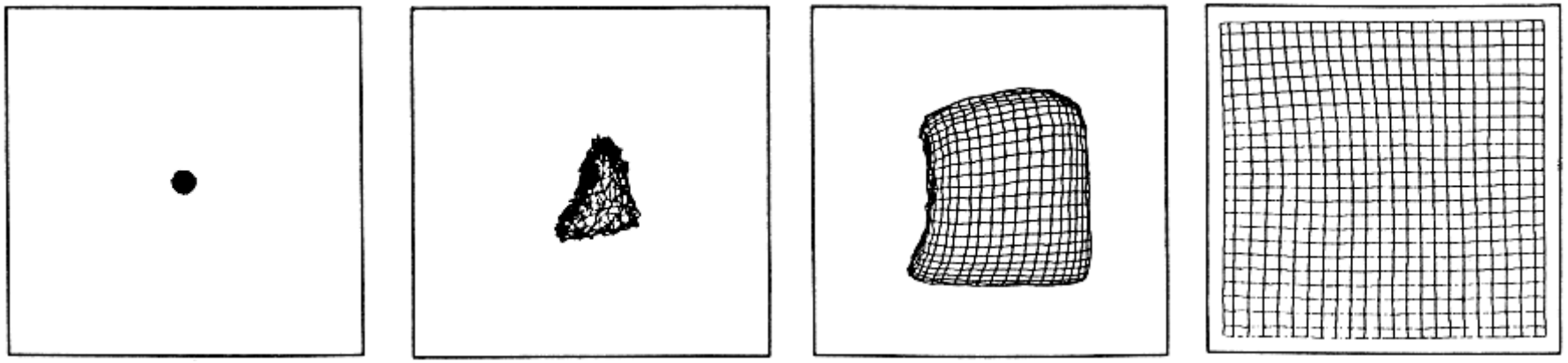
Topologie-Entwicklung

JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

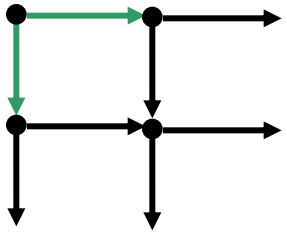
Eingabe: uniform-verteilte, 2-dim Zufallszahlen

Zeichnung: Verbindungsline vom Gewichtsvektorendpkt zum Nachbarn

t=0 **t=20** **t=100** **t=10000**



Zeichnung :



Neuronale Gase

- **Grundidee:** Nachbarschaftsverbindungen adaptiv formen

- **Graph des Netzes**:
  Ziel: Klassenprototypen sind Nachbarn $\Leftrightarrow$
  Adjazenzmatrix $A_{ij} = 1$

- **Bestimme Neuronenanzahl**, die dichter dran ist als Neuron *k*

$$r_k = | \{ i \text{ mit } |\mathbf{x}-\mathbf{w}_i| < |\mathbf{x}-\mathbf{w}_k| \} |$$

- **Lernregel** *k-te* Neuron

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + h(r_k,t)\,[\mathbf{x} - \mathbf{w}_k(t)]$$

Neuronale Gase

- **Grundidee:** Nachbarschaftsverbind. A_{ki} adaptiv formen

- **Mache Nachbarschaftsverbindung älter** *timer tick*
  Für alle direkten Nachbarn *i* von k : $T_{ki}(t+1) = T_{ki}(t) + 1$

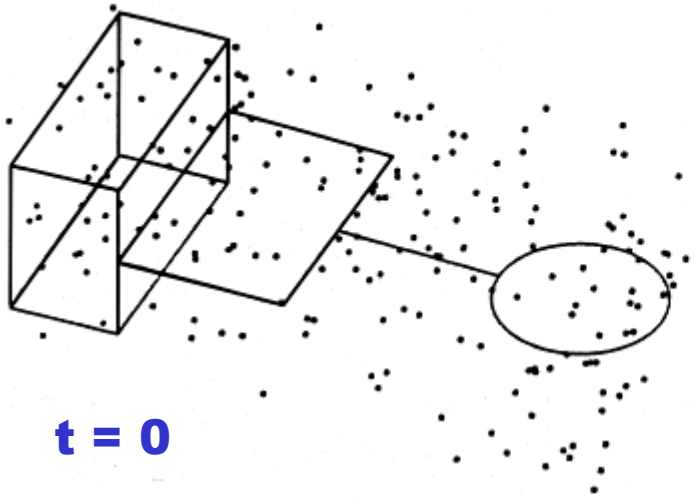
- **Bestätige als direkten Nachbarn**
  Wähle zweitnächstes Neuron *j* und setze $A_{kj} = 1, T_{kj} = 0$ *timer reset*

- **Lösche alte inaktive Nachbarn**
  Alle Neuronen mit $T_{kj} > T_{max}$ setze $A_{kj} = 0$

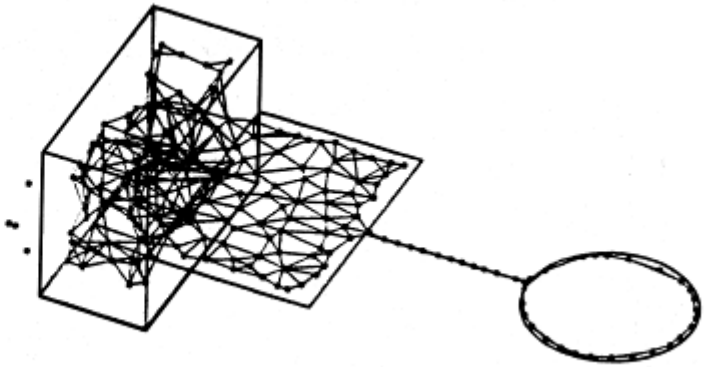
- **Aktualisiere Nachbarschaftssymmetrie**
  Setze für alle Neuronen $A_{ki} = A_{ik}$, $T_{ki} = T_{ik}$

Neuronale Gase

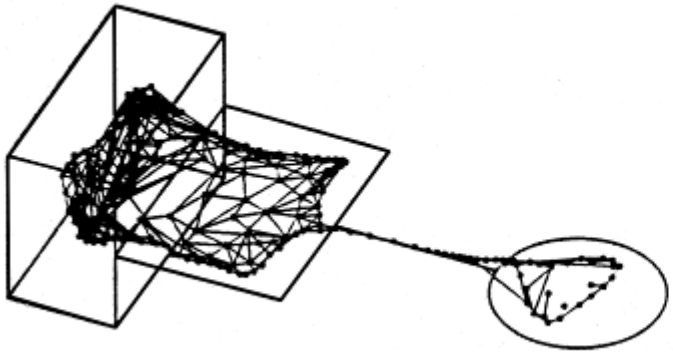
- Anpassung an interne Dimensionalität



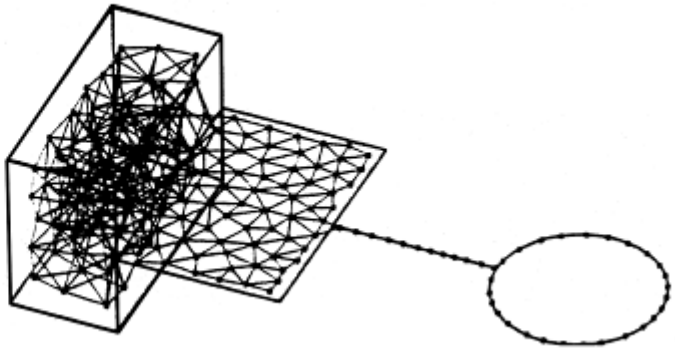
t = 0



t = 15.000



t = 5.000



t = 40.000

Fragen ?