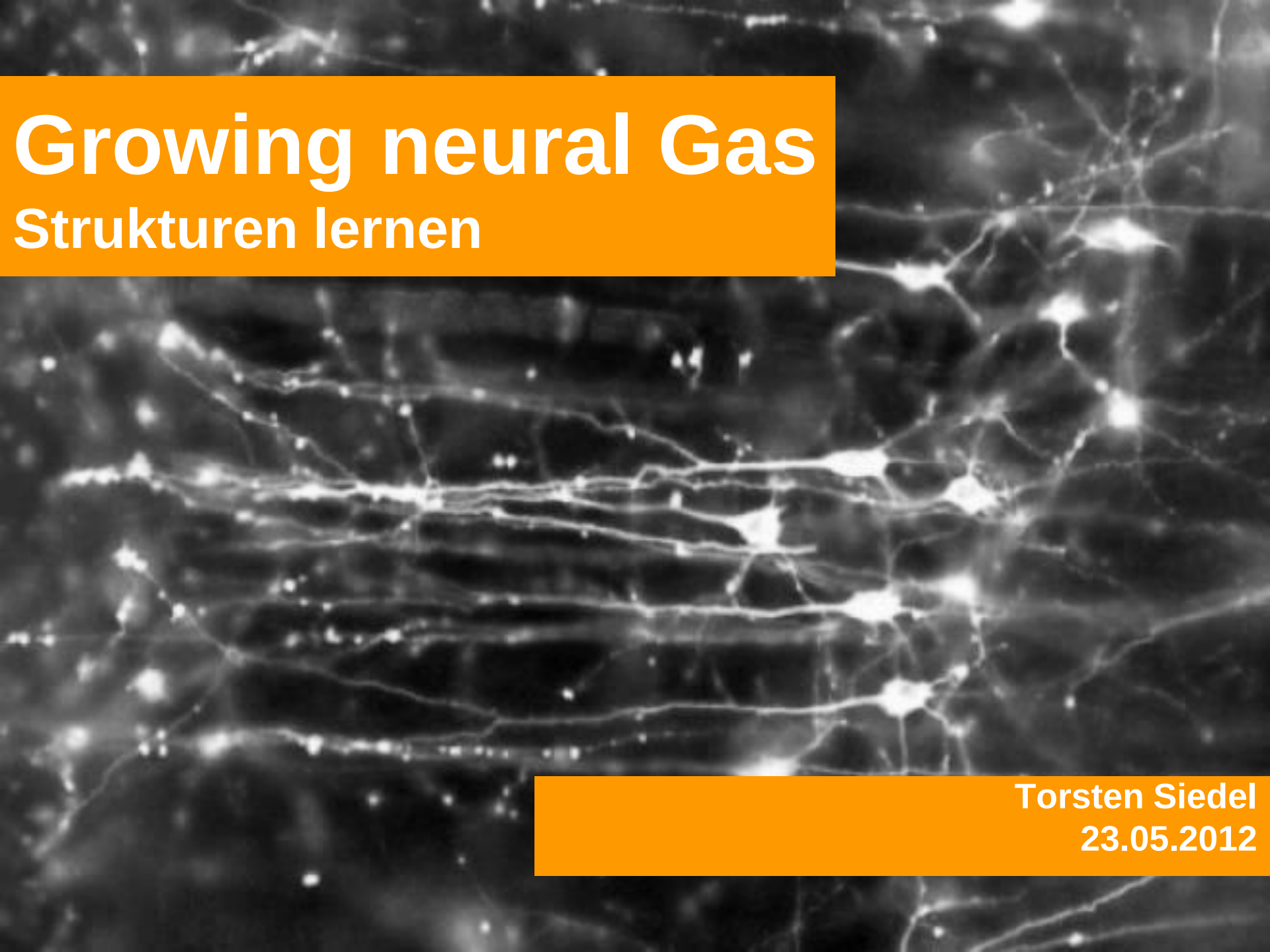


# Growing neural Gas

## Strukturen lernen

**Torsten Siedel**

**23.05.2012**

# Inhalt

1. Prozess der Selbstorganisation
2. Lernen - momentan oder statistisch?
3. Vektorbasierte Neuronale Netze
4. Klassifizierung der Lernverfahren
5. Online Lernen
6. Neuronales Gas
7. Wachsendes Neuronales Gas
8. JAVA-Tool (Demonstration)
9. Beispiel aus der Natur

# Prozess der Selbstorganisation

- Systemparameter sind von außen vorgegeben
- Genauer Endzustand des Systems vorher nicht bekannt
- Zustand entwickelt sich durch Selbstorganisation
- Einfluss von Einheiten meist nur durch „Nachbarn“
- Keine Zentrale Planung

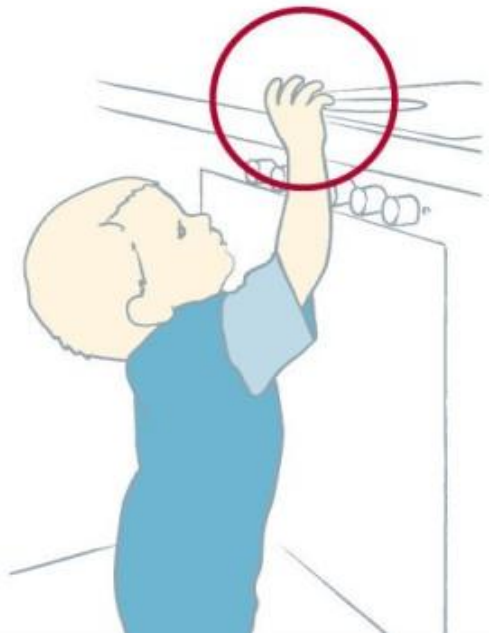
Beispiel: Ökonomisches Gefüge eines Staates

# Prozess der Selbstorganisation

Dynamische Selbstorganisation eines Vogelschwarms

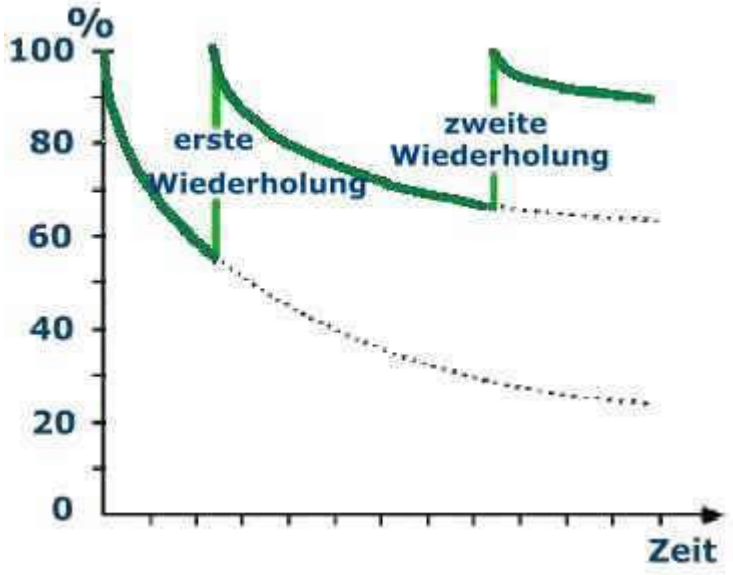
# Lernen - momentan oder statistisch?

## Momentanes Lernen



- Gesichtserkennung
- Gefahren

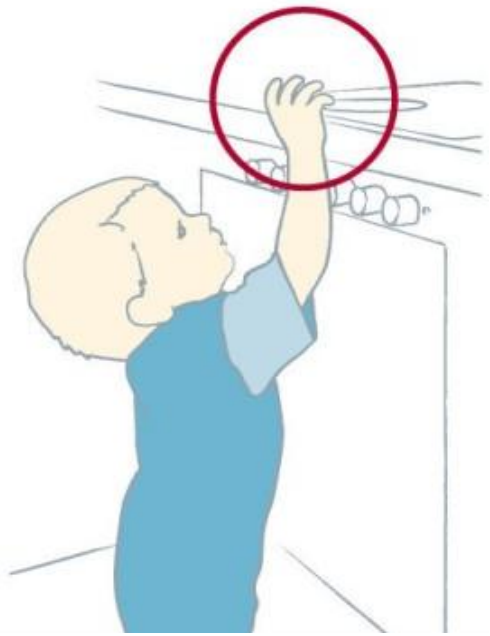
## Statistisches Lernen




- Komplexe Bewegungsabläufe
- Kombinatorische Aufgaben

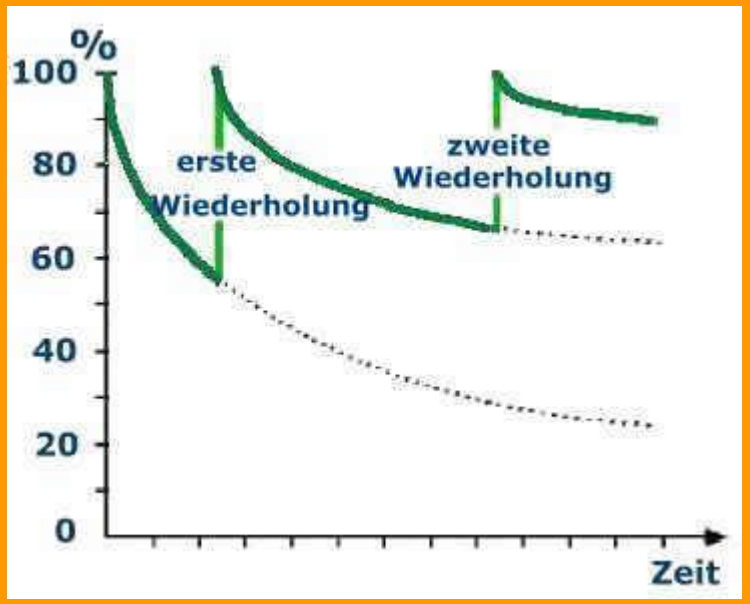
# Lernen - momentan oder statistisch?

## Momentanes Lernen



- Gesichtserkennung
- Gefahren

## Statistisches Lernen




- Komplexe Bewegungsabläufe
- Kombinatorische Aufgaben

# Vektorbasierte Neuronale Netze

**Struktur**

Der Netzzustand ist wesentlich durch eine Anzahl von n-dimensionalen Vektoren beschreiben, wobei n die Dimension des Eingaberaums ist. Diese Vektoren werden als Referenzvektoren bezeichnet.

Ein Netz besteht aus einer Menge $\mathcal{A} = \{c_1, \ldots, c_N\}$ von Neuronen.

Ein Neuron entspricht einer „Einheit“

Jeder Einheit c ist der Referenzvekto $\mathbf{w}_c \in \mathrm{R}^n$ zugeordnet.

# Vektorbasierte Neuronale Netze

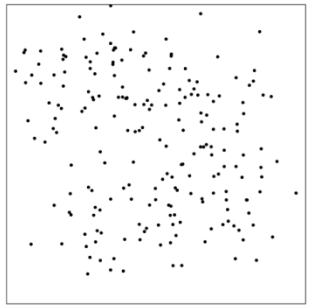
**Trainingsdaten**

Kontinuierliche Wahrscheinlichkeitsverteilung

$$p(\boldsymbol{\xi}),\ \boldsymbol{\xi} \in \mathrm{R}^n$$

Endliche Menge

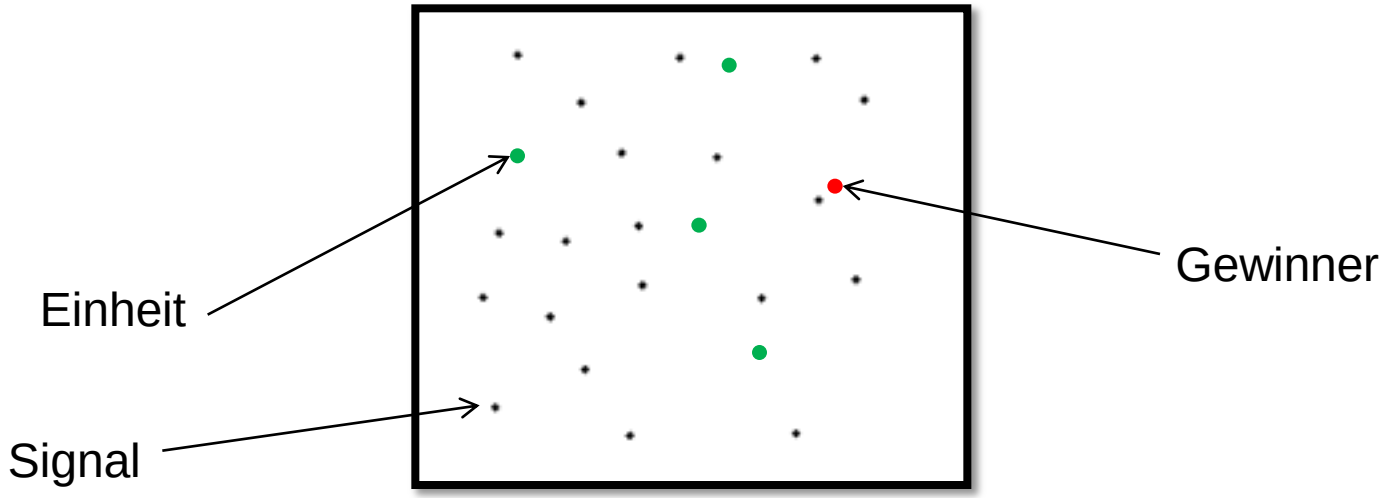
$$\mathcal{D} = \{\boldsymbol{\xi}^1, \ldots . \boldsymbol{\xi}^M\},\ \boldsymbol{\xi}^i \in \mathrm{R}^n.$$



**Gewinner**

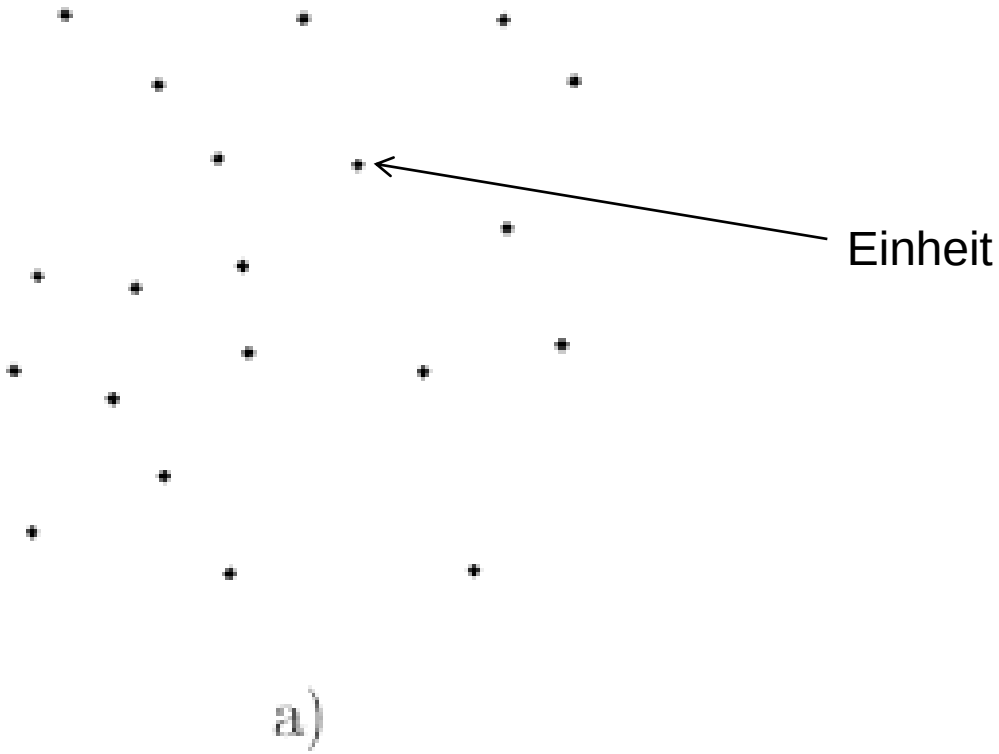
Der Gewinner ist die Einheit, die den *nächstgelegenen* Referenzvektor hat:

$$s(\boldsymbol{\xi}) = \arg\ \min_{c \in \mathcal{A}} \|\boldsymbol{\xi} - \mathbf{w}_c\|$$



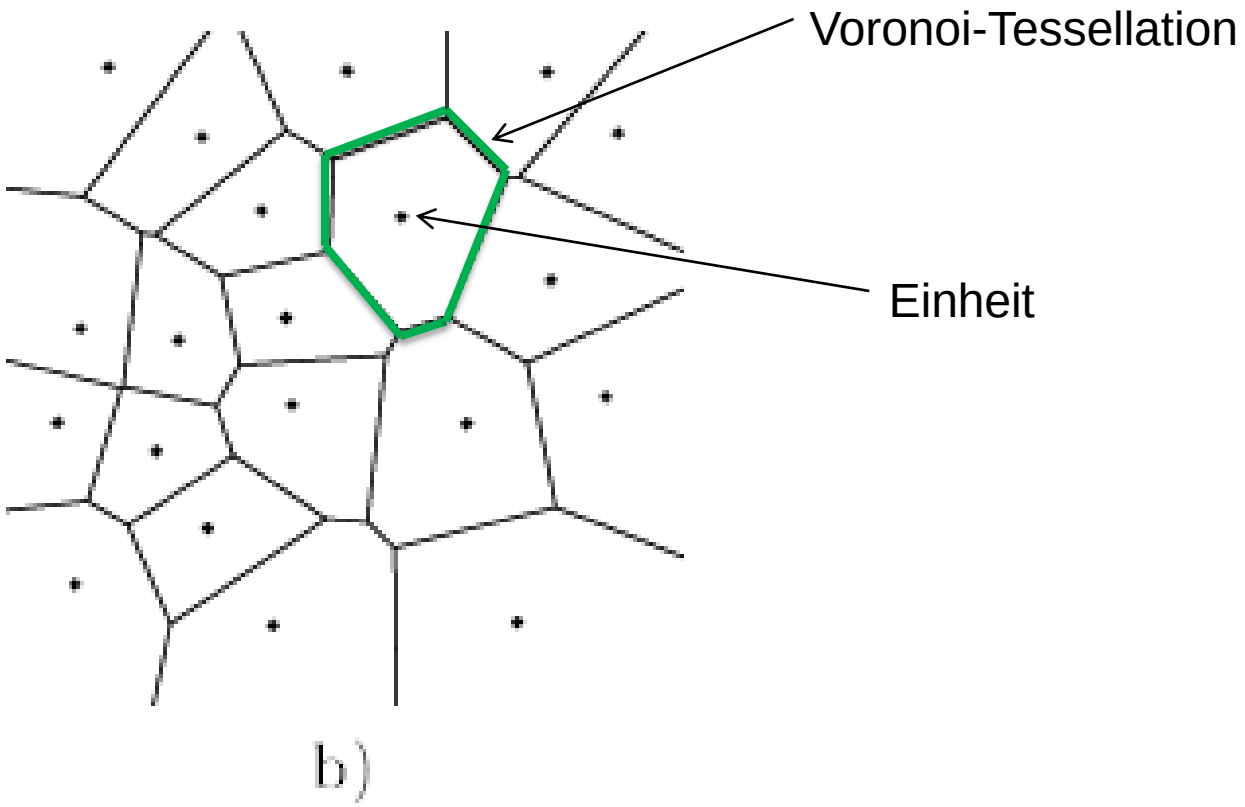
# Vektorbasierte Neuronale Netze

**Voronoi-Tessellation (b) und Delaunay-Triangulation (c)**



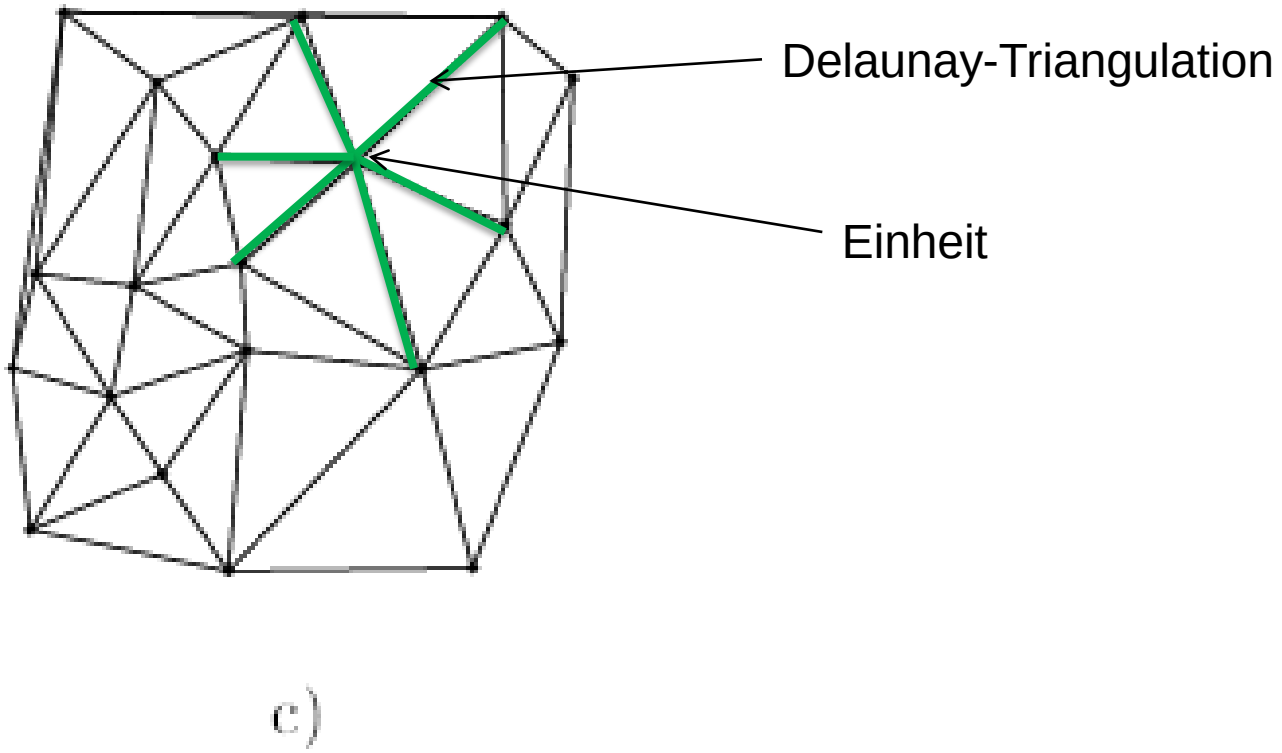
# Vektorbasierte Neuronale Netze

**Voronoi-Tessellation**



# Vektorbasierte Neuronale Netze

**Delaunay-Triangulation**



# Klassifizierung der Lernverfahren

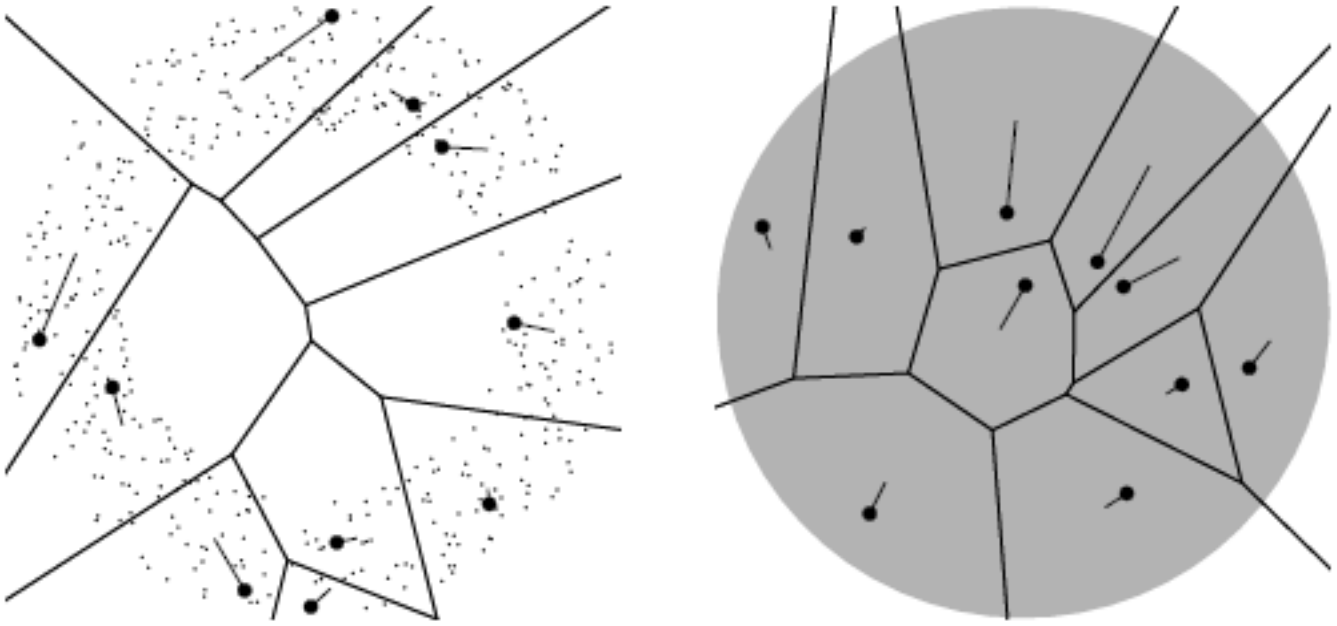
| | |
|---|---|
| Hartes Wettbewerbslernen für endliche Datenmengen | • LBG (Linde, Buzo, Gray)<br>• LBG-U (Fritzke) |
| Hartes Wettbewerbslernen für große und unendliche Datenmengen | • Hard Competitive Learning (standard)<br>• Competitive Hebbian Learning (Martinetz) |
| Weiches Wettbewerbslernen ohne feste Netzwerkdimension | • Neural Gas (Martinetz)<br>• Growing Neural Gas / GNG-U (Fritzke)<br>• Neural Gas with Competitive Hebbian Learning (Martinetz) |
| Weiches Wettbewerbslernen mit feste Netzwerkdimension | • Growing Grid (Fritzke)<br>• Self-Organizing Map (Kohonen) |

# Klassifizierung der Lernverfahren

| | |
|---|---|
| Hartes Wettbewerbslernen für endliche Datenmengen | • LBG (Linde, Buzo, Gray)<br>• LBG-U (Fritzke) |
| Hartes Wettbewerbslernen für große und unendliche Datenmengen | • Hard Competitive Learning (standard)<br>• Competitive Hebbian Learning (Martinetz) |
| Weiches Wettbewerbslernen ohne feste Netzwerkdimension | • Neural Gas (Martinetz)<br>• Growing Neural Gas (Fritzke)<br>• Neural Gas with Competitive Hebbian Learning (Martinetz) |
| Weiches Wettbewerbslernen mit feste Netzwerkdimension | • Growing Grid (Fritzke)<br>• Self-Organizing Map (Kohonen) |

# Online Lernen

**Lernen mittels Voronoi-Tessellation**



a) endliche Datenmenge

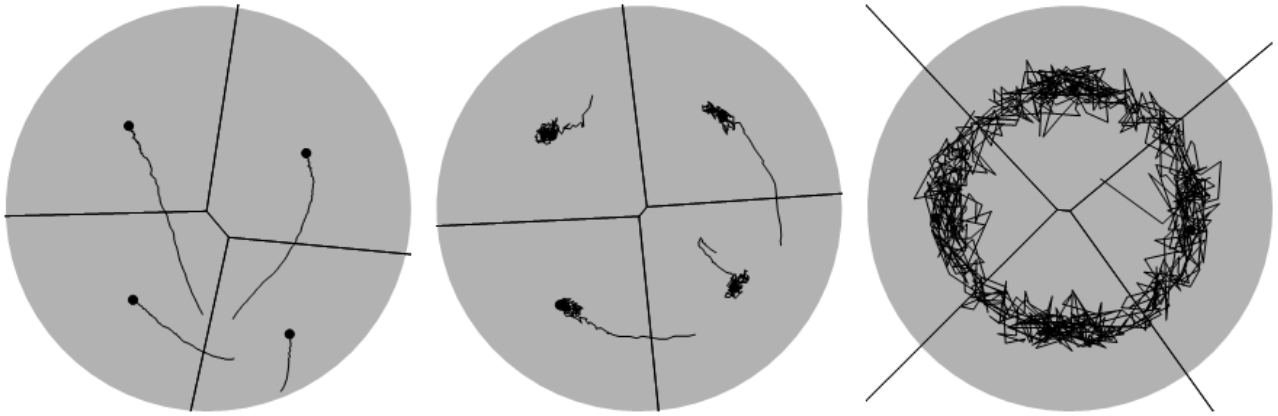
b) kontinuierliche W-Verteilung

# Online Lernen

**K-means-Lernen**

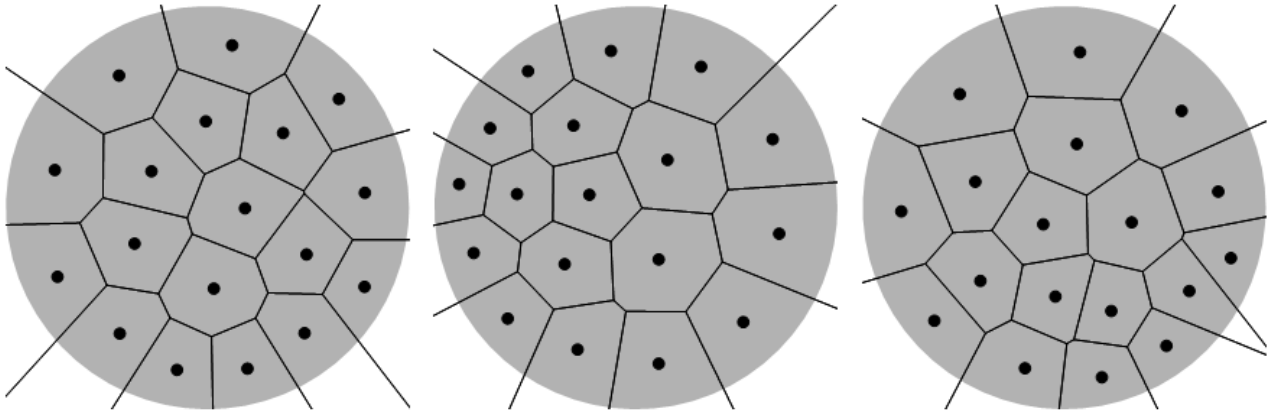
$\epsilon_0 =$ Lernrate

$\mathbf{w}_s := \mathbf{w}_s + \epsilon(t)(\boldsymbol{\xi} - \mathbf{w}_s)$



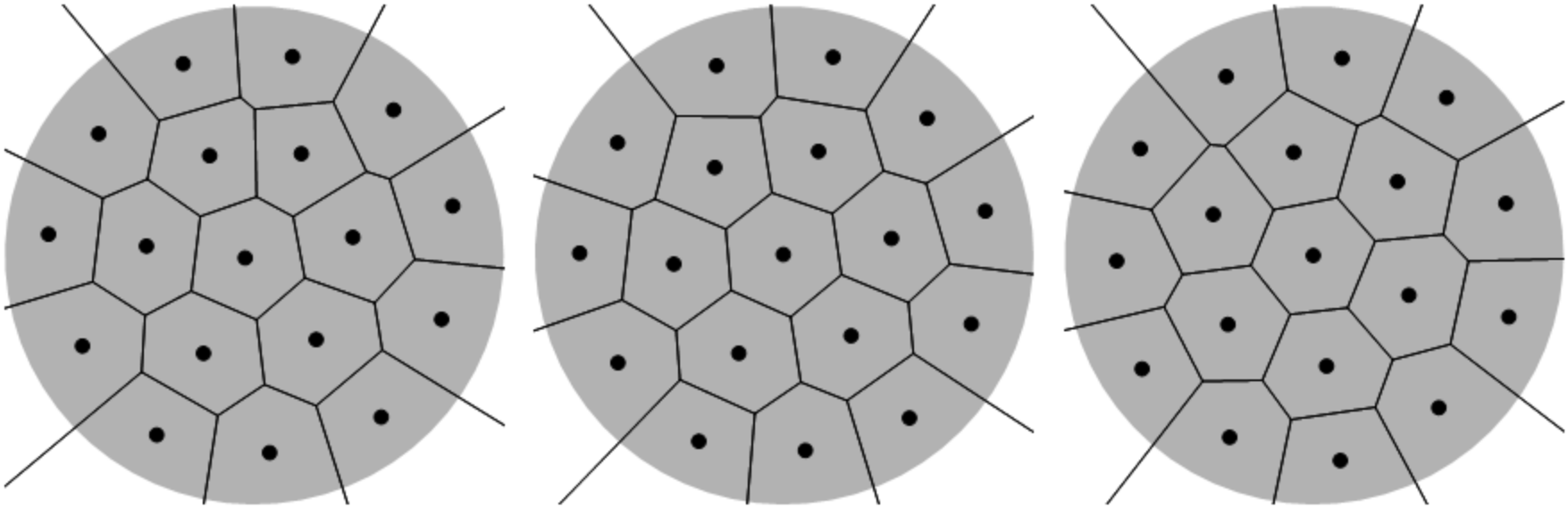
a) $\epsilon_0 = 0.001$ b) $\epsilon_0 = 0.01$ b) $\epsilon_0 = 0.1$

$\epsilon_0 =$ gleich



# Online Lernen

$\epsilon_0 =$ Exponentiell abnehmend



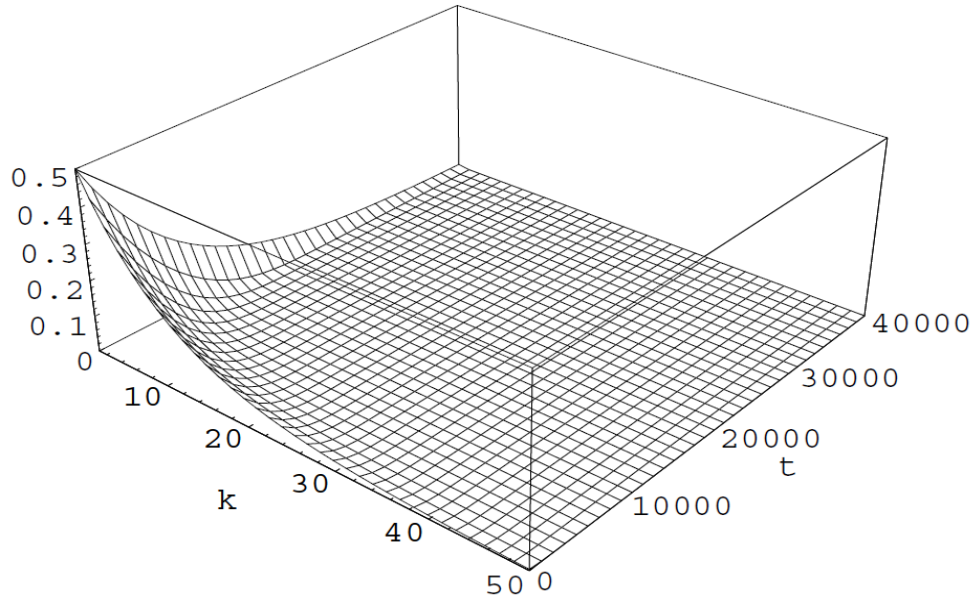
# Neuronales Gas

Lernrate $$\epsilon(t) := \epsilon_i(\epsilon_f/\epsilon_i)^{t/t_{\max}},$$

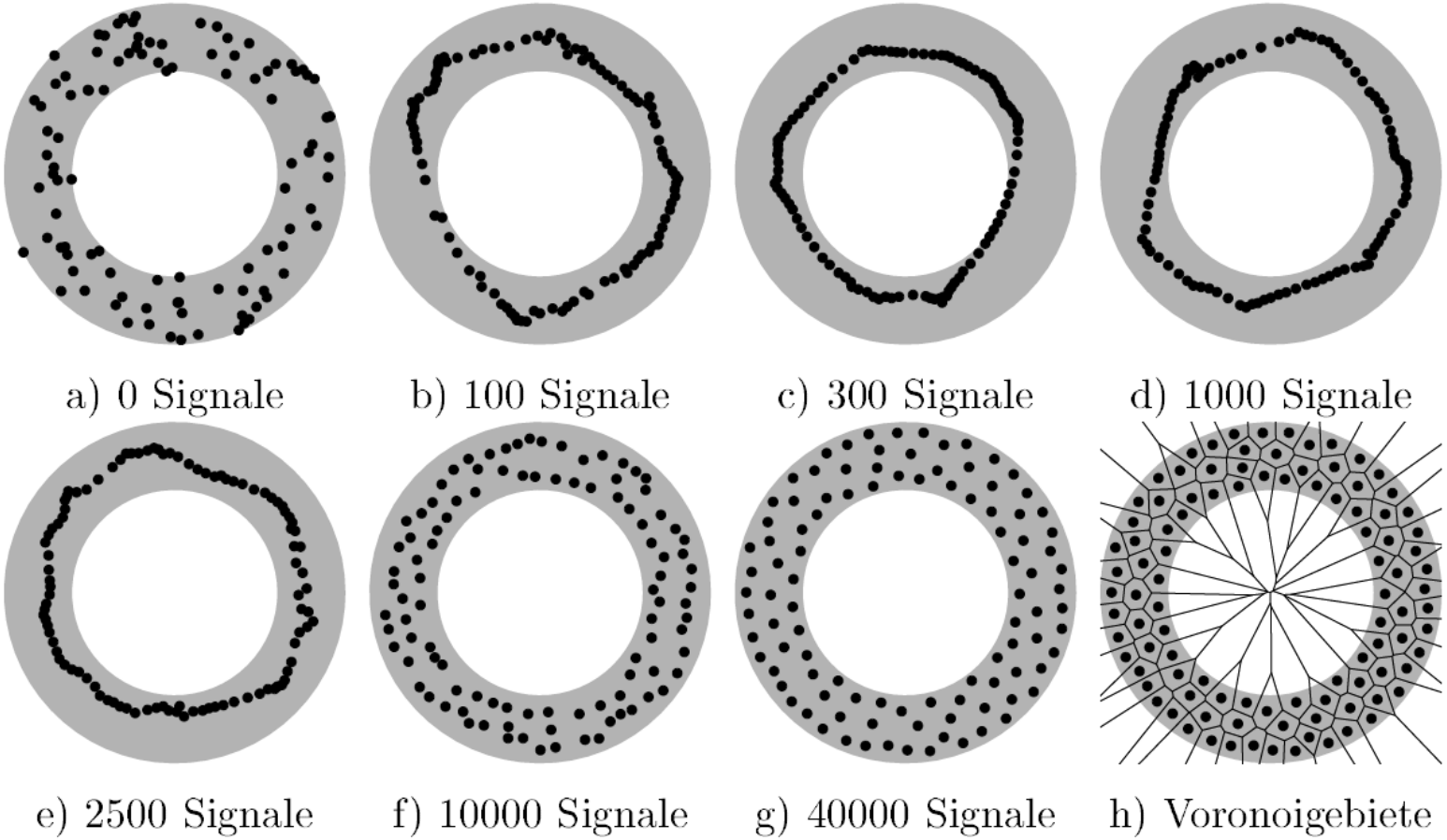
Lernrate $$\lambda(t) := \lambda_i(\lambda_f/\lambda_i)^{t/t_{\max}},$$

Nachbarschaftsreichweite $$h(t,k) := \exp((1-k)/\lambda(t))$$

Adaptionsstärke $$\Delta\mathbf{w}_{s_k} := \epsilon(t)\cdot h(t,k)\cdot(\boldsymbol{\xi}-\mathbf{w}_{s_k})$$



# Neuronales Gas

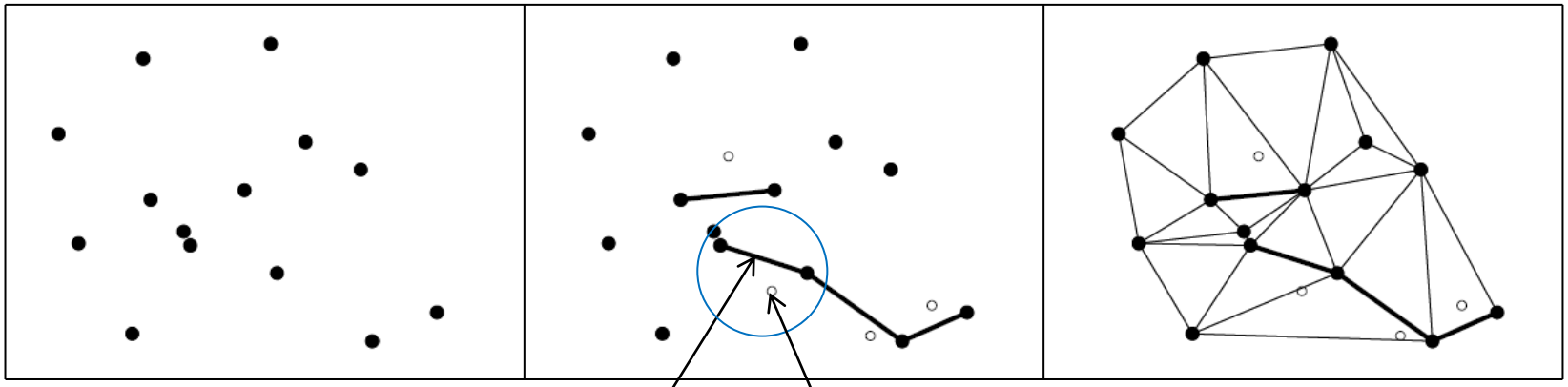


a) 0 Signale b) 100 Signale c) 300 Signale d) 1000 Signale

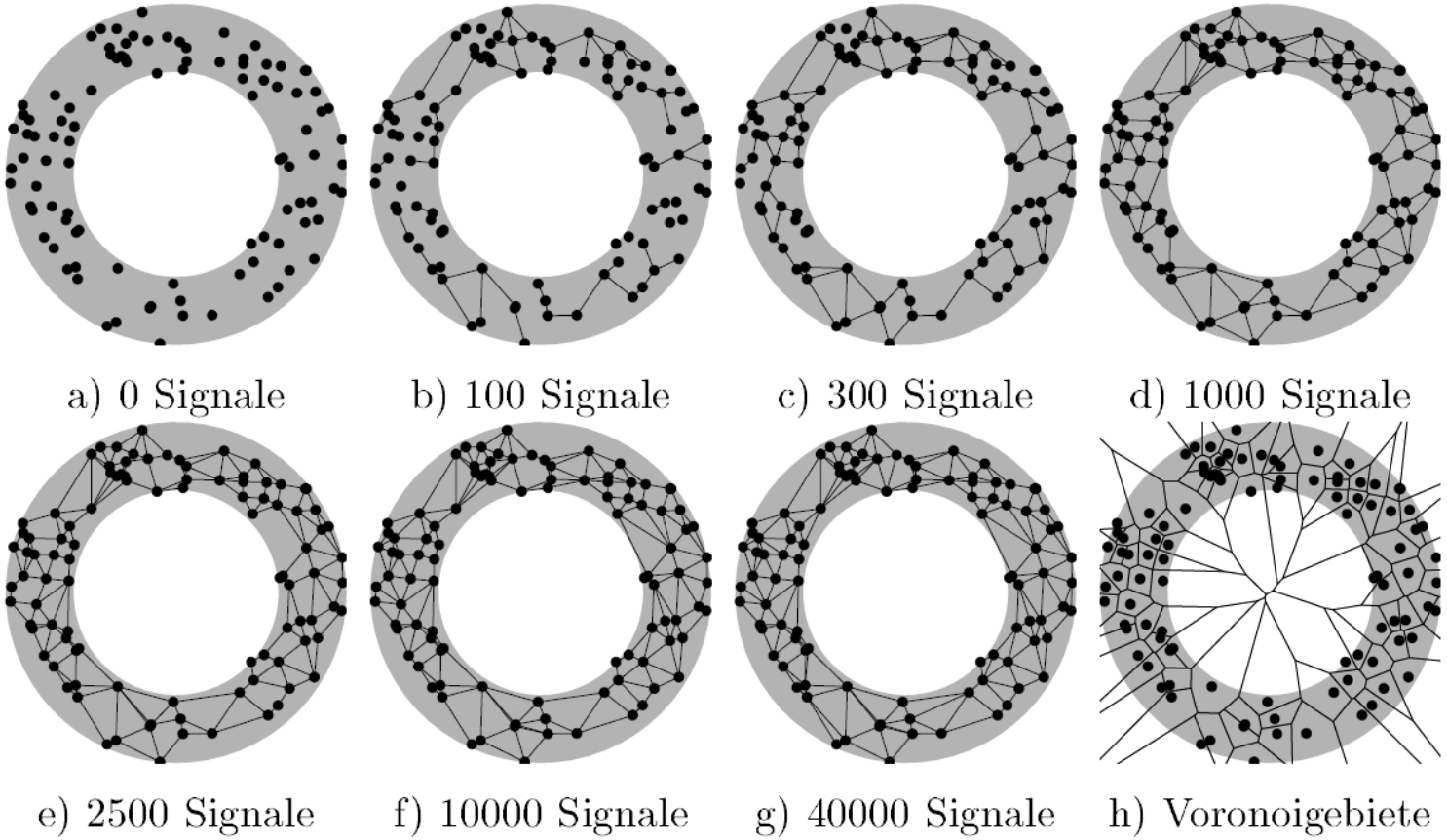
e) 2500 Signale f) 10000 Signale g) 40000 Signale h) Voronoigebiete

# Neuronales Gas

**Hebb’sches Wettbewerbslernen ->** Erzeugen einer Netztopologie



# Neuronales Gas

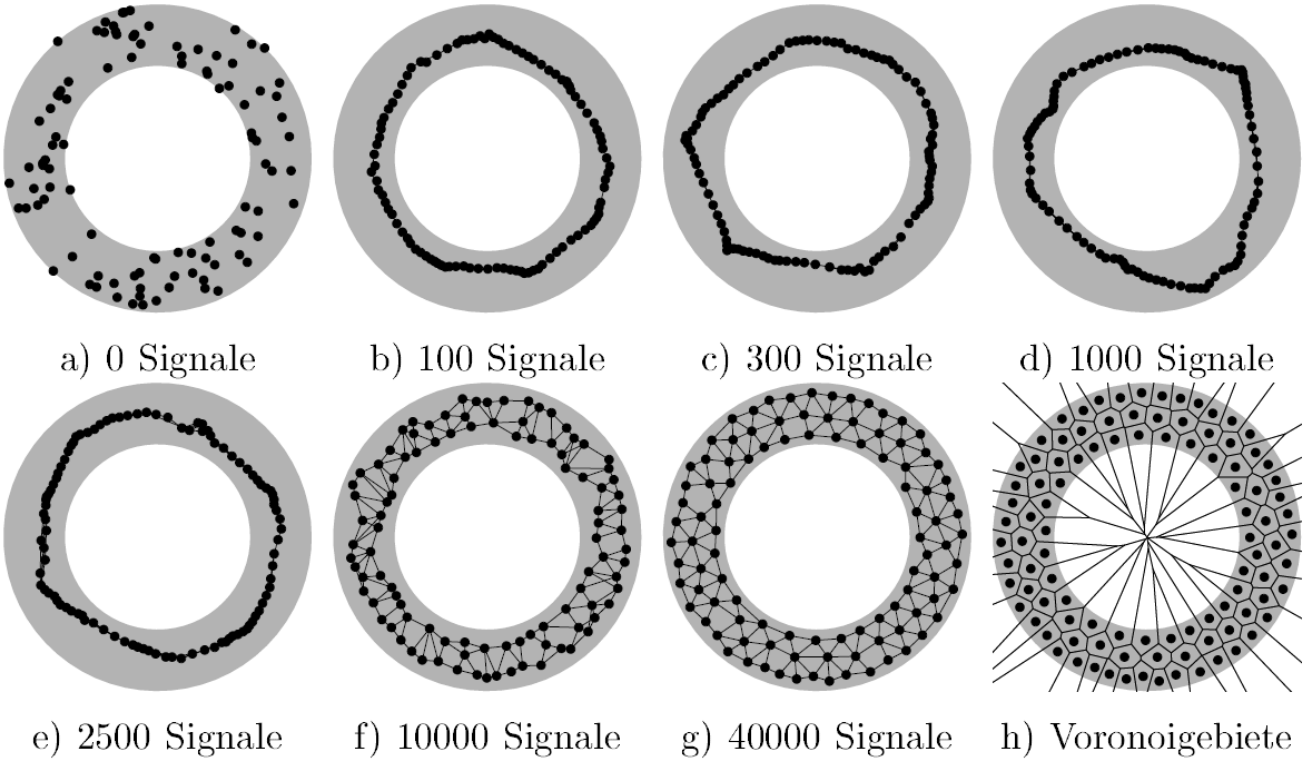


a) 0 Signale b) 100 Signale c) 300 Signale d) 1000 Signale

e) 2500 Signale f) 10000 Signale g) 40000 Signale h) Voronoigebiete

# Neuronales Gas

**Hebb'schen Wettbewerbslernen** + **Adaption der Referenzvektoren** (Neuronales Gas)



a) 0 Signale b) 100 Signale c) 300 Signale d) 1000 Signale

e) 2500 Signale f) 10000 Signale g) 40000 Signale h) Voronoigebiete

# Wachsendes Neuronales Gas

**Problem**

Üblicherweise *a priori - F*estlegen der Netzwerkgröße.

-> Viele Testdurchläufe notwendig um gewünschtes Netzwerk zu erzeugen

**Wachstum**

Start mit einem kleinem Netzwerk (z.B. nur zwei Einheiten)

Ende bei z.B. Erreichen eines mittleren Quantisierungsfehlers

# Wachsendes Neuronales Gas

**Problem**

Üblicherweise *a priori* - *F*estlegen der Netzwerkgröße.

-> Viele Testdurchläufe notwendig um gewünschtes Netzwerk zu erzeugen

**Wachstum**

Start mit einem kleinem Netzwerk (z.B. nur zwei Einheiten)

Ende bei z.B. Erreichen eines mittleren Quantisierungsfehlers

Fehlergröße soll reduziert werden.

$$\sum_{c \in \mathcal{A}} E_c \quad min!$$

Absenke des Fehlers durch Einfühgen einer neuen Einheit.

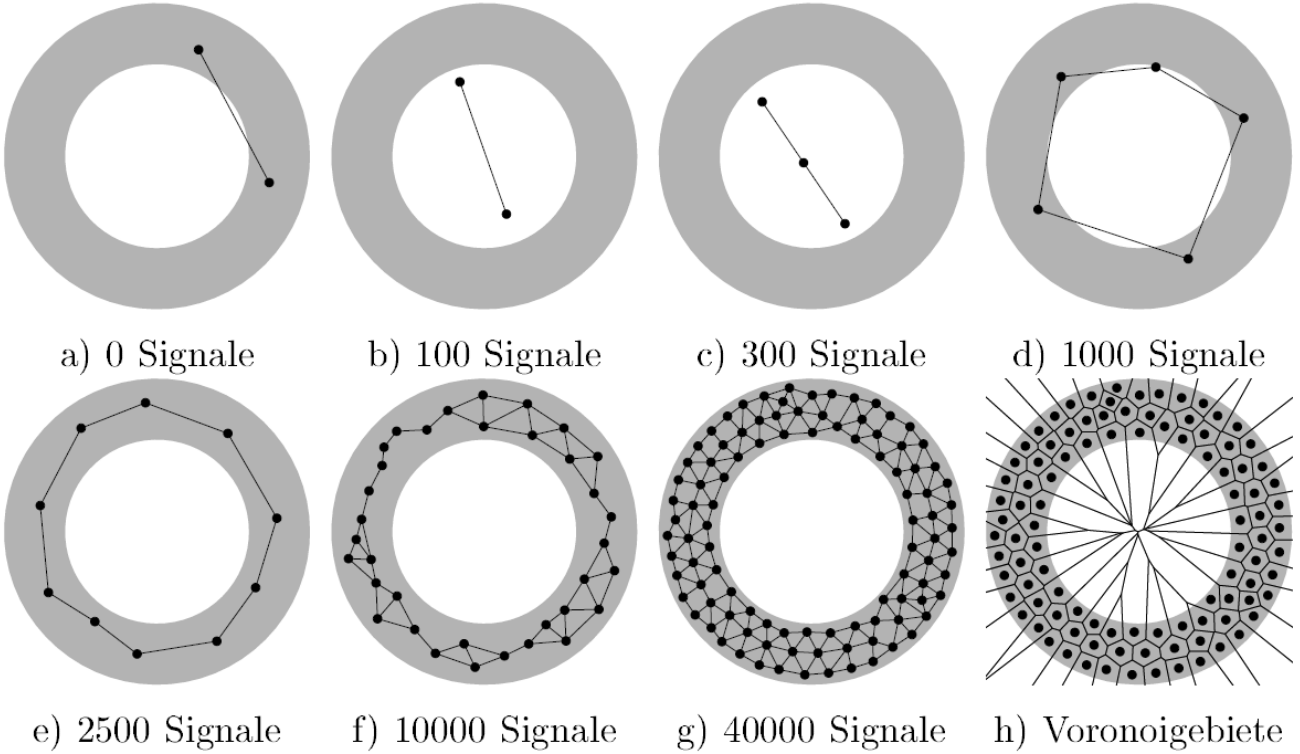
$$\sum_{c \in \mathcal{A}} E_c > \sum_{c \in \mathcal{A} \cup \{r\}} E_c$$

Stärkste Absenkung des Fehlers, durch Einfühgen bei Einheit mit dem größen Fehler.

$$q := \arg\max_{c \in \mathcal{A}} \mathrm{E}_c.$$

# Wachsendes Neuronales Gas

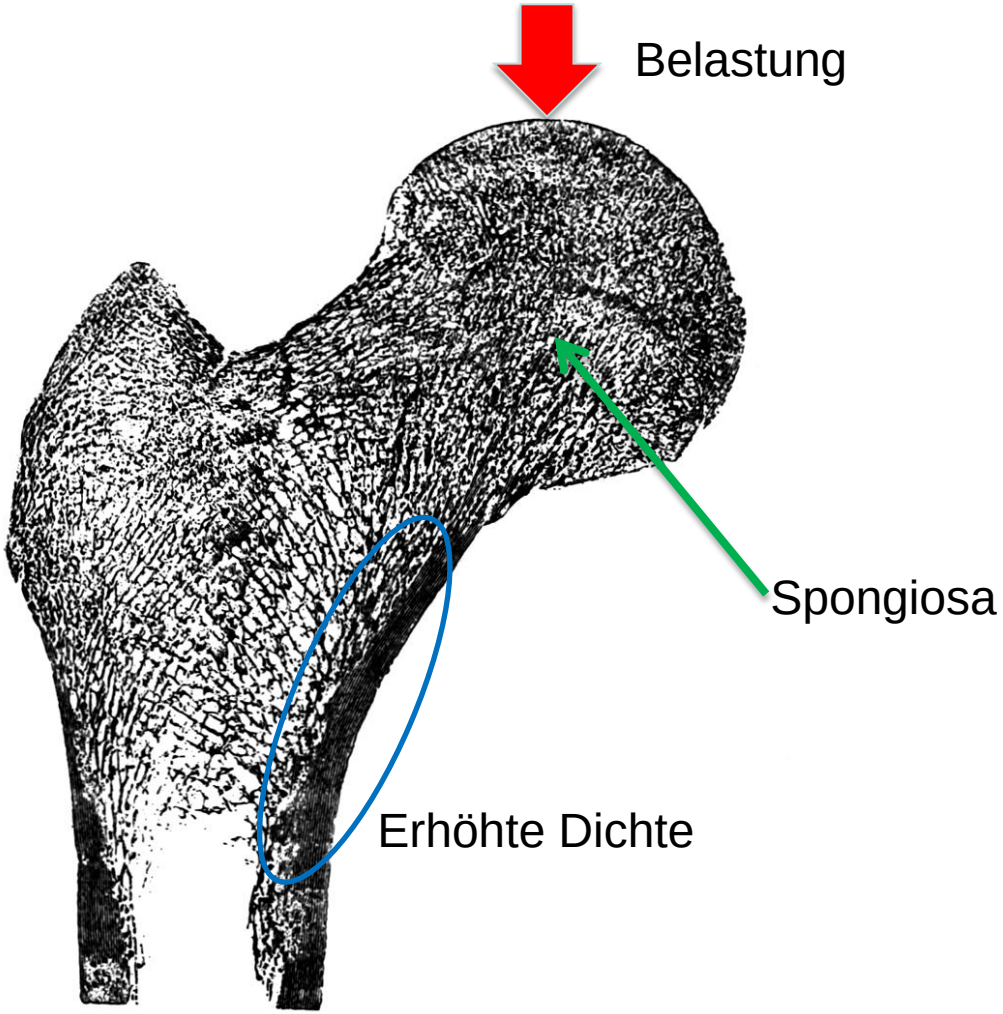
**Hebb'schen Wettbewerbslernen** + **Adaption der Referenzvektoren** (Neuronales Gas) + **Wachstum**



# Demonstration

JAVA-Tool

# Beispiel aus der Natur



# Beispiel aus der Natur

# Beispiel aus der Natur

# Beispiel aus der Natur

# Beispiel aus der Natur

# Beispiel aus der Natur

# Quellen

Fritzke, B.:
Vektorbasierte Neuronale Netze. Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Habilitation, 1998.

Fritzke, B.:
Wachsende Zellstrukturen - ein selbstorganisierendes neuronales Netzwerkmodell. Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Dissertation, 1992.

Java-Programm DemoGNG zur Simulation von Neuronalen Netzen:
http://sund.de/netze/applets/gng/full/GNG-U_0.html


Danke für Euer Zuhören
… nun zur Diskussion