

KOHONEN NETZWERKE

BRUCKNER, HANGLER,
MOSER-SCHWAIGER & SPERL



Übersicht

- **Einführung**

Manuel Sperl

- **Algorithmus**

Stefan Hangler

- **Einsatzbereiche mit Beispiele**

Eva Bruckner

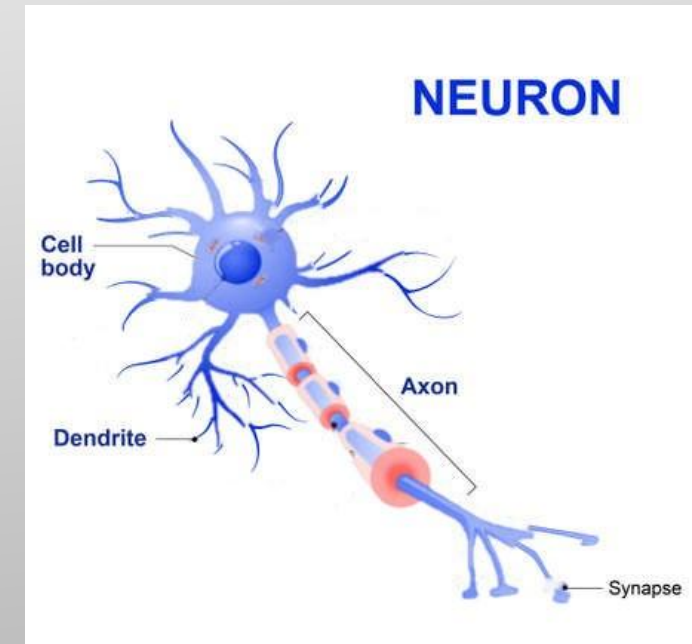
Marie Moser-Schwaiger



EINFÜHRUNG – Künstliche Neuronale Netze

Was sind Künstliche Neuronale Netze (KNN)

- Inspiration liegt in der Biologie und als Vorlage dient das menschliche Gehirn
- Neuronen bilden Netz
- Dendriten = Eingänge
- Axon = Ausgang
- Synapsen = Schnittstelle im Neuron



Aufbau eines Neurons

Was sind Künstliche Neuronale Netze (KNN)

- Kommunikation unter den Nervenzellen
- Netz an Nervenzellen
- Aufbau der Künstlichen Neuronalen Netze erfolgt in Schichten
- Eingabe-, Ausgabe-, und mögliche Zwischenschicht(en)
- Zwischenschicht, auch genannt verborgene Schicht, engl. “hidden layer”

Was sind Künstliche Neuronale Netze (KNN)

- verschiedene Lernstrategien
- unbeaufsichtigtes Lernen, engl. „unsupervised learning“
- keine Regeln vorab festgelegt
- nur anhand des Algorithmus



EINFÜHRUNG – Kohonennetze

Geschichte

- Teuvo Kohonen, geb. am 11. Juli 1934
- 1970er und 1980er Jahre - „selbständige assoziative Speicher“, woraus er den Kohonennetze-Algorithmus entwickelte
- viele Projekte und Arbeiten zu diesem Thema
- zwei Bücher (1995 und 2001) veröffentlicht



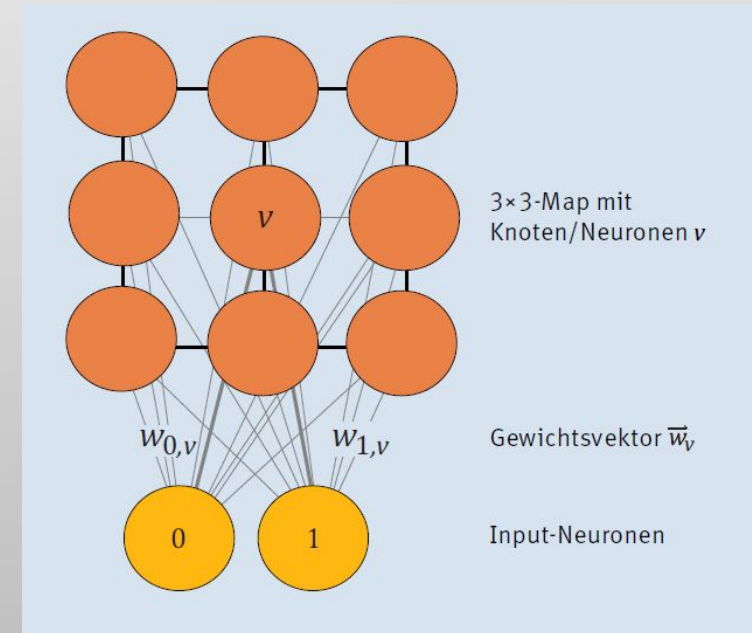
Teuvo Kohonen (1934 -)

Was sind Kohonennetze

- selbst-organisierende Karten, engl. “*self-organizing map*” = **SOM**
- große Verbreitung in einigen Anwendungen
- agieren ohne externen „Lehrer“
- Kohonennetze → „unsupervised learning“

Was sind Kohonennetze

- Aufgebaut in zwei Schichten
- Verknüpfung mittels Neuronen
- Neuron mit Knoten v verknüpft
- jeder dieser v Knoten mit Eingabeneuronen verbunden



SOM mit 2 Eingabe Neuronen und einer 3x3 Kohonen Schicht

Wesentliche Idee

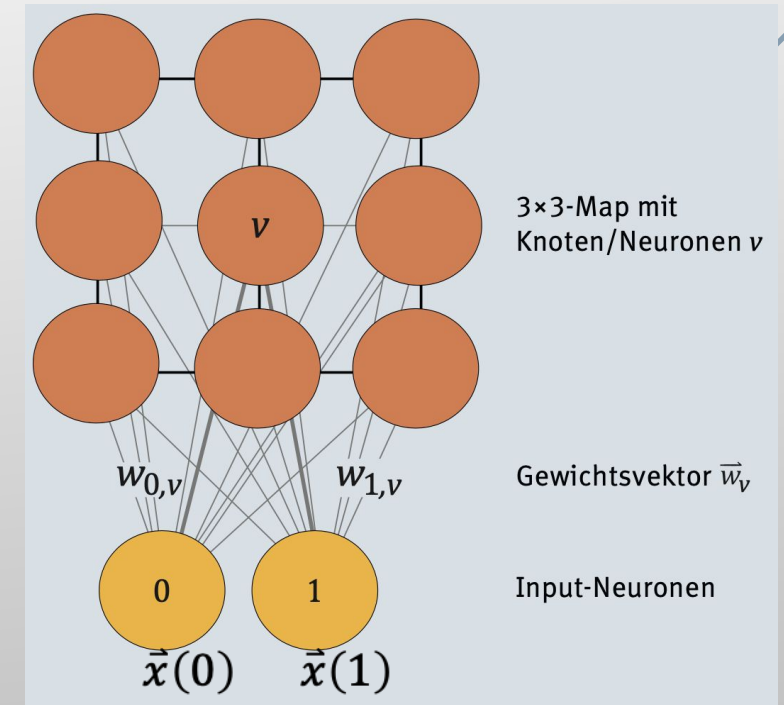
- betrachten einer unbekanntem Karte
- Legende gibt Informationen über diese Karte
- sammeln von Erfahrung und bei Bedarf abrufen dieser

ALGORITHMUS



Definitionen

- $\vec{w}_v(s)$ - Gewichtsvektor, Knoten v , Iterationsschritt s
- u – Index von der BMU (Best Matching Unit)
- $\vec{x}(s)$ - Input Vektor
- $\alpha(s)$ - Lernrate (nimmt im Laufe der Zeit ab)
- $\theta(u, v, s)$ - Nachbarschaftsfunktion (Werte im Intervall $[0,1]$)



Ablauf Algorithmus

1. Initialisierung der Gewichtsvektoren \vec{w}_v

zufällige Werte im Intervall $[0,1]$

2. Berechnungen - Wiederhole für I Iterationen

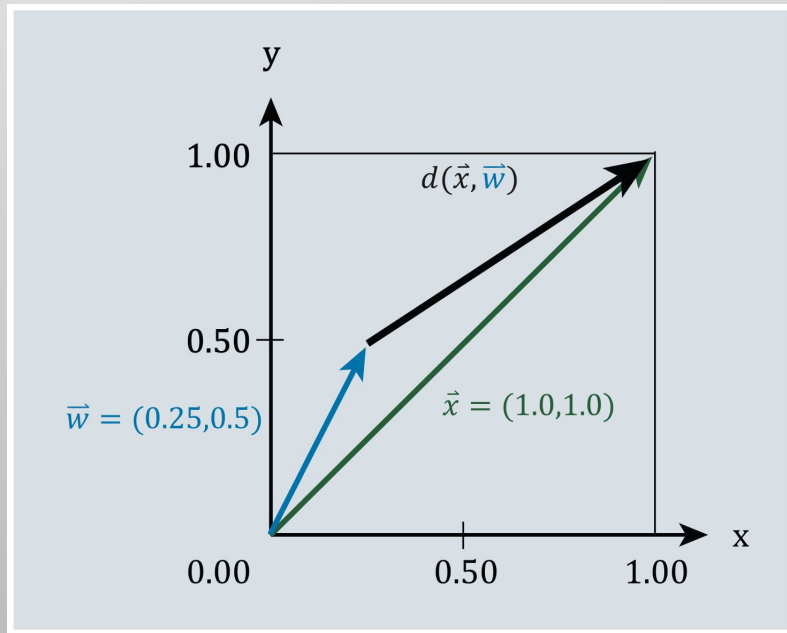
- a) Wähle Input-Vektor $\vec{x}(s)$
- b) Berechne Distanz zwischen $\vec{x}(s)$ und $\vec{w}(s)$ aller Knoten $\rightarrow d(\vec{x}(s), \vec{w}(s))$
- c) Knoten u mit kleinster Distanz wählen (BMU)
- d) $\vec{w}_u(s)$ und $\vec{w}(s)$ der Nachbarn werden Richtung $\vec{x}(s)$ angepasst

$$\vec{w}_v(s + 1) = \vec{w}_v(s) + \theta(u, v, s) * \alpha(s) * (\vec{x}(s) - \vec{w}(s))$$

Berechnung Distanz

Euklidischer Abstand:

$$d(\vec{x}, \vec{w}) = \sqrt{(x_1 - w_1)^2 + (x_2 - w_2)^2 + \dots} = \sqrt{\sum_i (x_i - w_i)^2}$$



$$\vec{w} = (0.25, 0.5) \quad \vec{x} = (1.0, 1.0)$$

$$= \sqrt{(1.0 - 0.25)^2 + (1.0 - 0.5)^2}$$

$$\cong 0.9$$

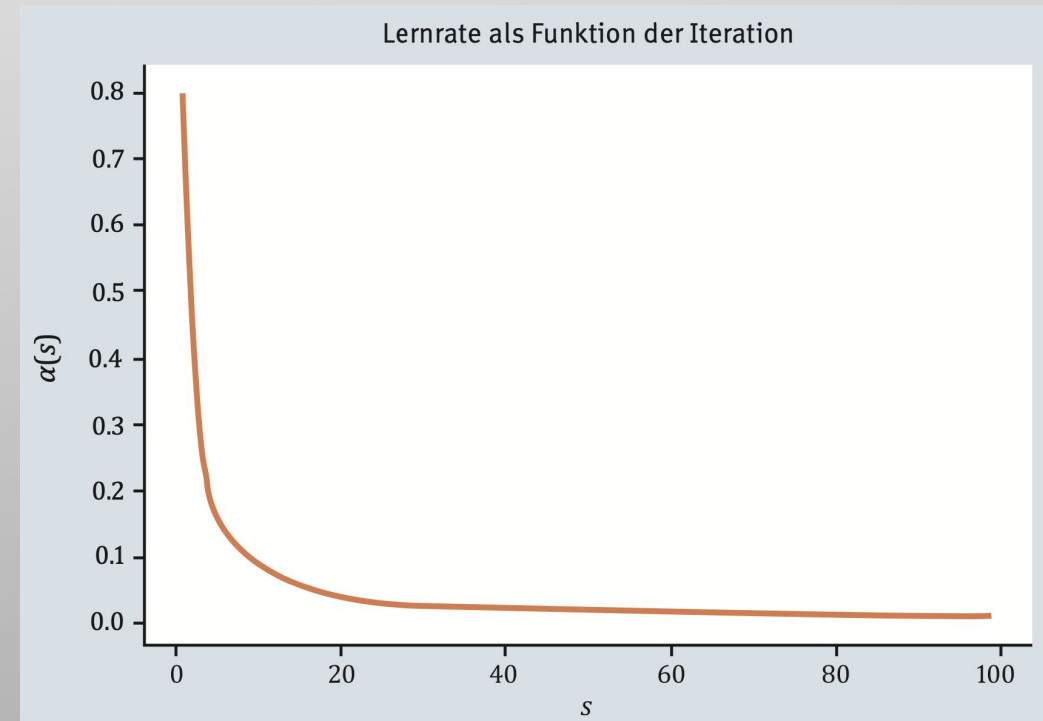
Lernrate α

- Beliebige Funktion
 - Anfangs starke Änderungen
 - Abfall der Lernrate mit zunehmender Iteration

$$\alpha(0) = 0.8$$

$$\alpha(s) = \alpha(0) * \frac{1}{s} \longrightarrow$$

$$\alpha(s, I) = \alpha(0) * \left(1 - \frac{s}{I}\right)$$



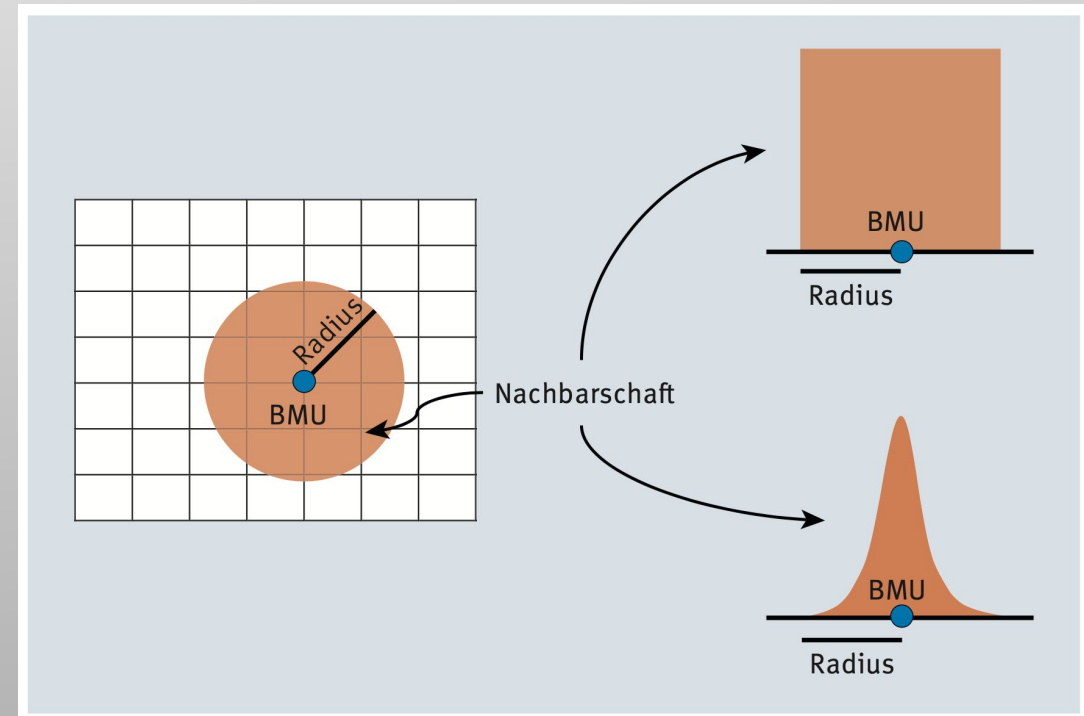
Nachbarschaftsfunktion θ

- Gleiche Eigenschaften wie Lernrate für $h_{u,v}(s)$ und $r(s, I)$

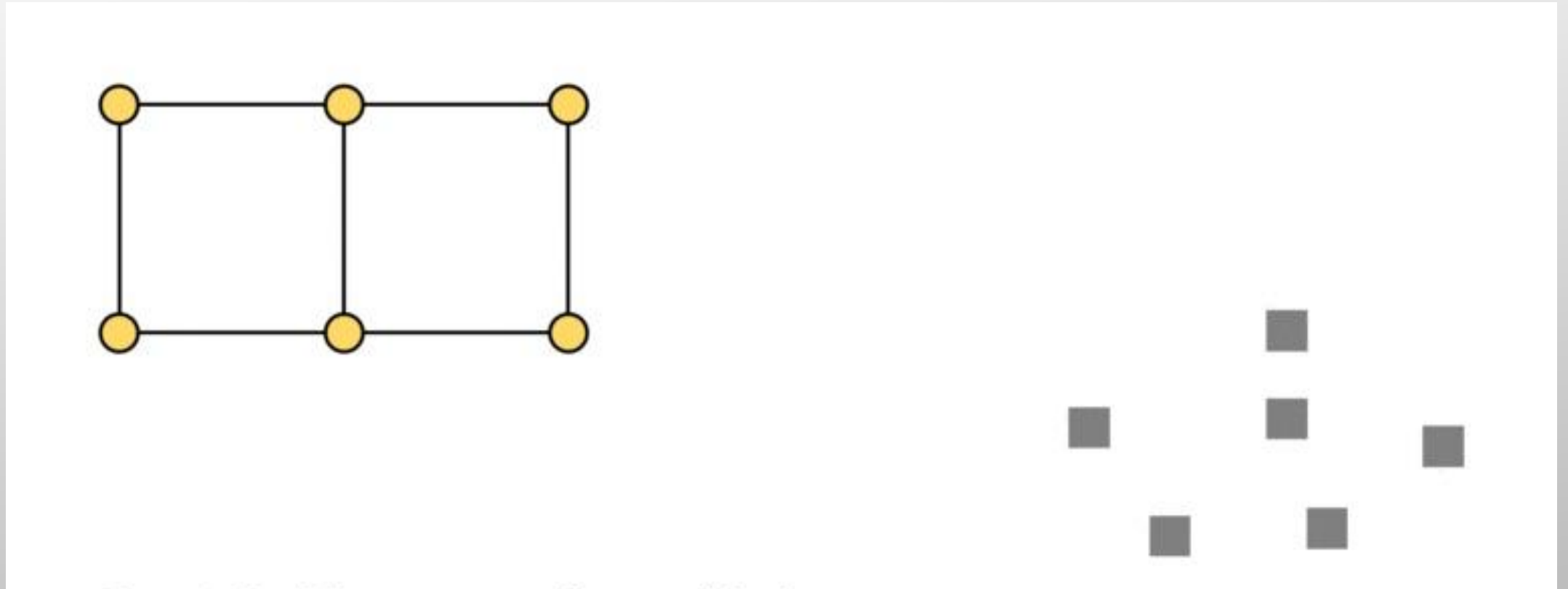
$$\theta(u, v, s) = \begin{cases} h_{u,v}(s), & \text{falls } d(u, v) \leq r(s, I) \\ 0, & \text{sonst} \end{cases}$$

$$r(s, I) = r(0) * e^{-\frac{s}{I}}$$

z.B.: $r(0) = 6$

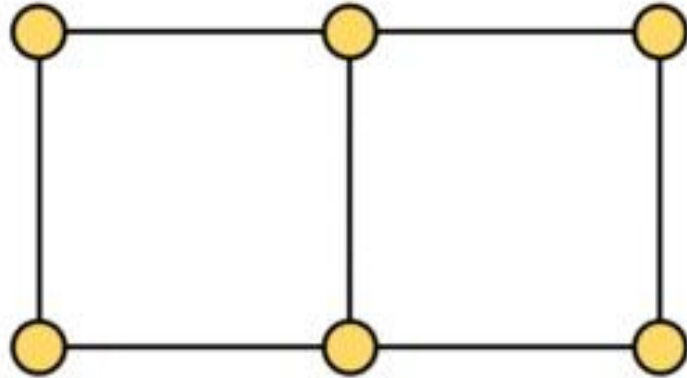


Animation Nachbarschaftsfunktion θ



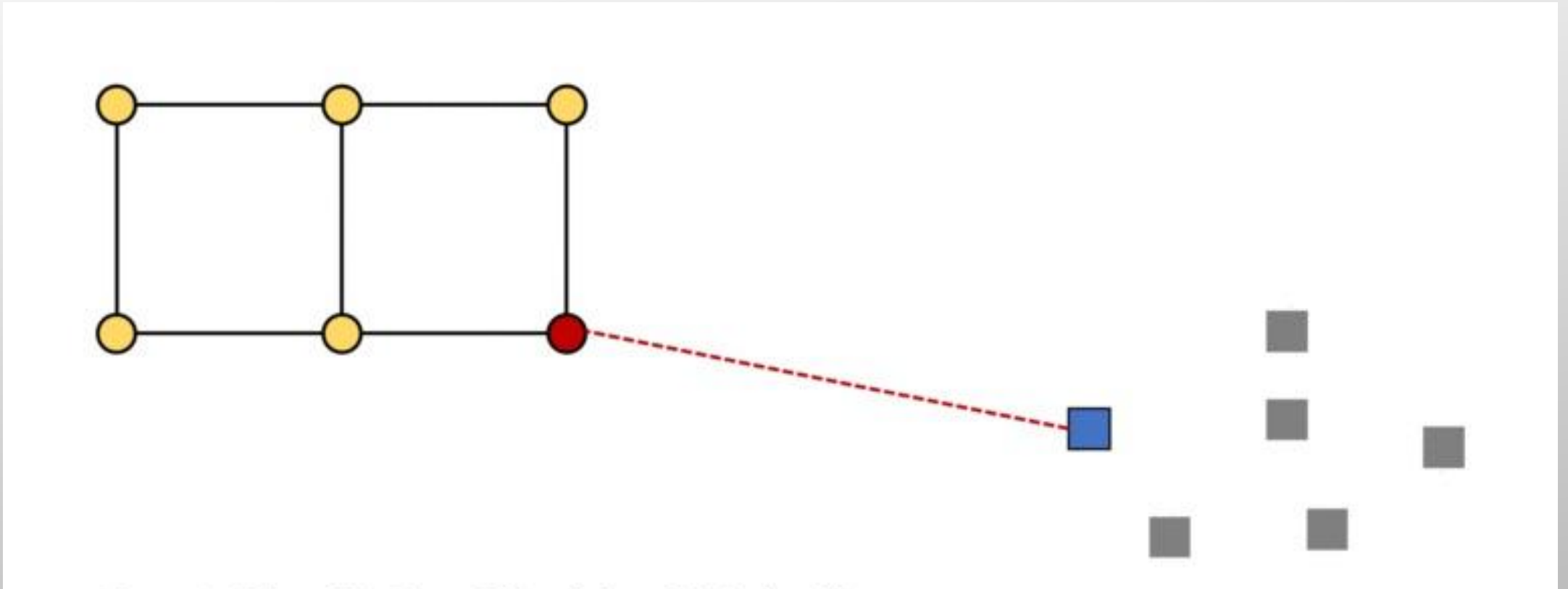
Schritt 0: alle Neuronen (gelb) sind positioniert im Netzwerk
und die grauen Quadrate sind Eingabeknoten (Inputdaten)

Animation Nachbarschaftsfunktion θ



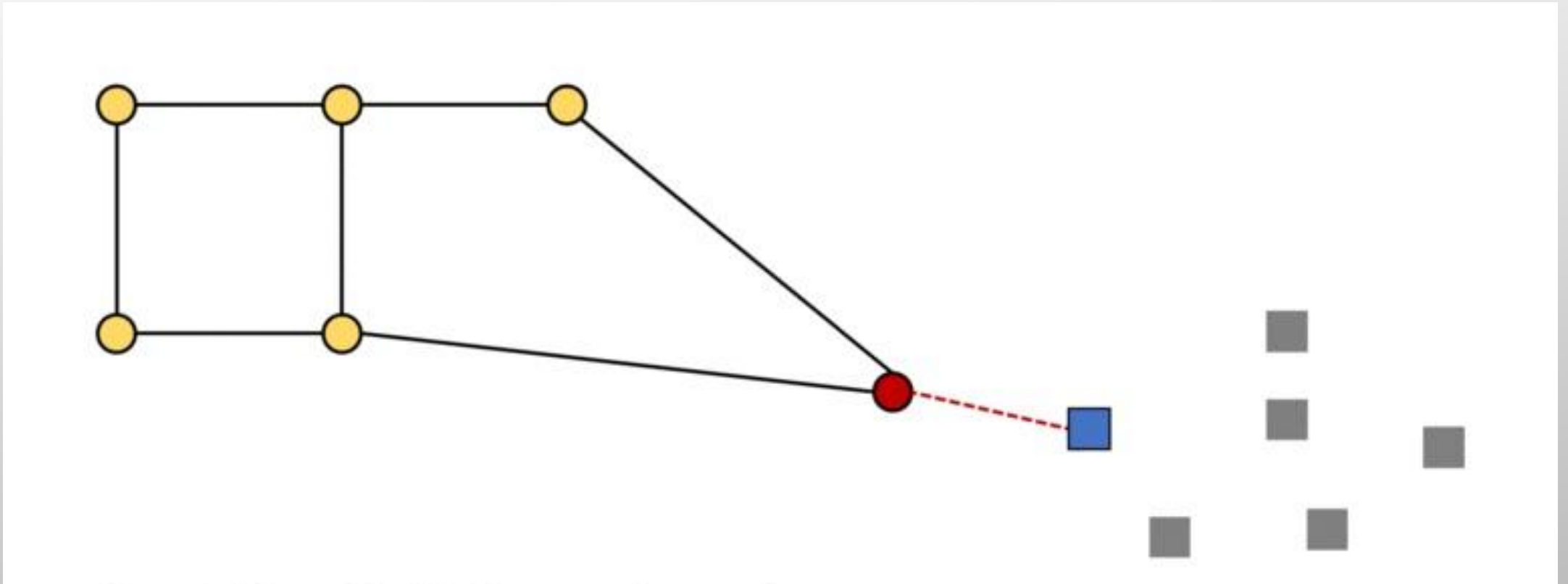
Schritt 1: einen Eingabeknoten auswählen (blau)

Animation Nachbarschaftsfunktion θ



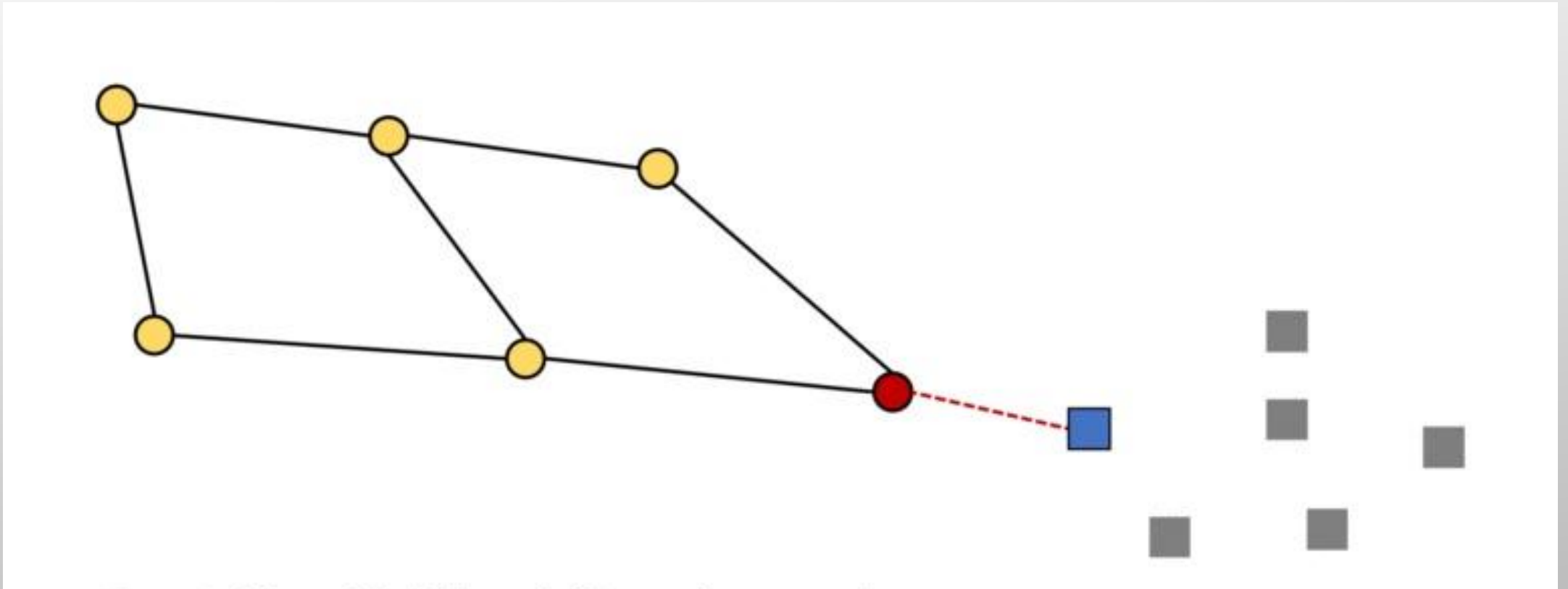
Schritt 2: identifizieren der Best Matching Unit (rot)

Animation Nachbarschaftsfunktion θ



Schritt 3: die BMU wandern nähert zum Eingabeknoten

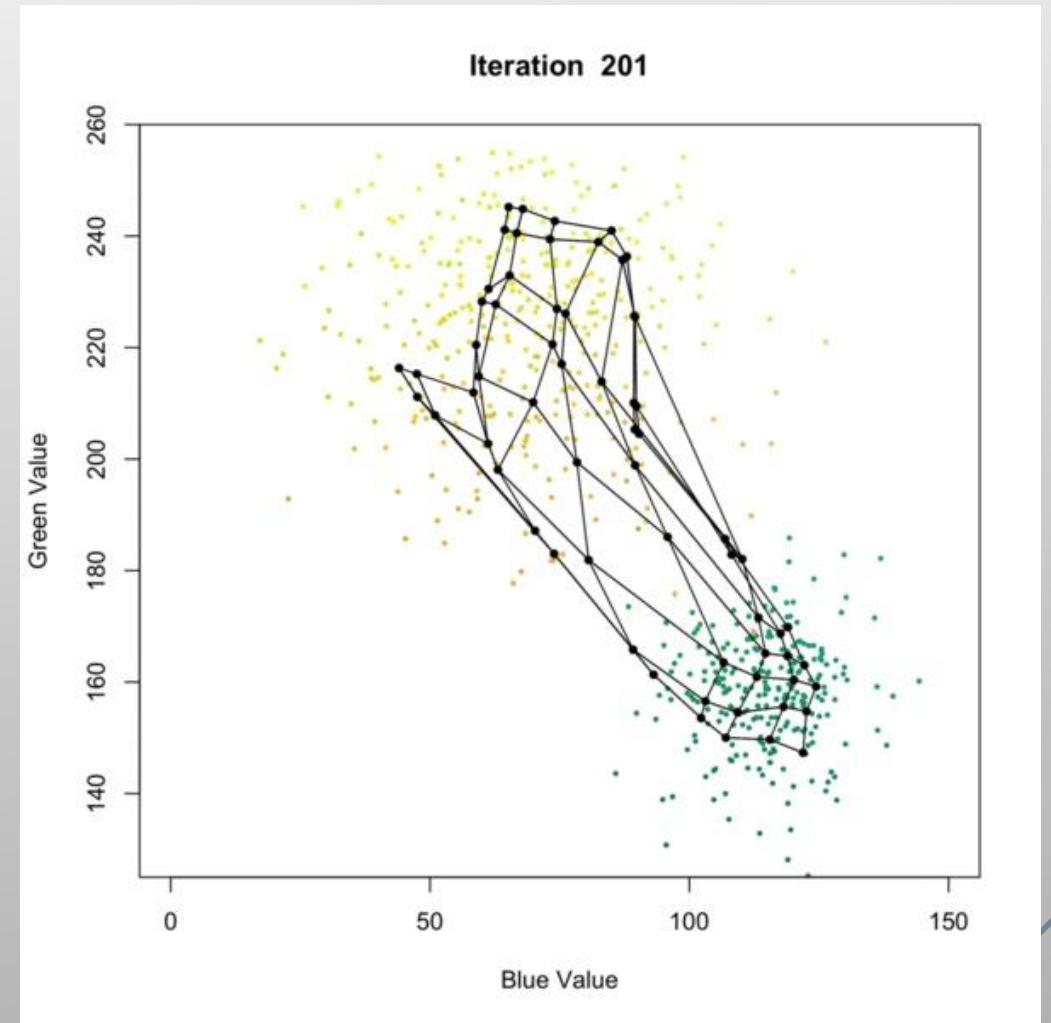
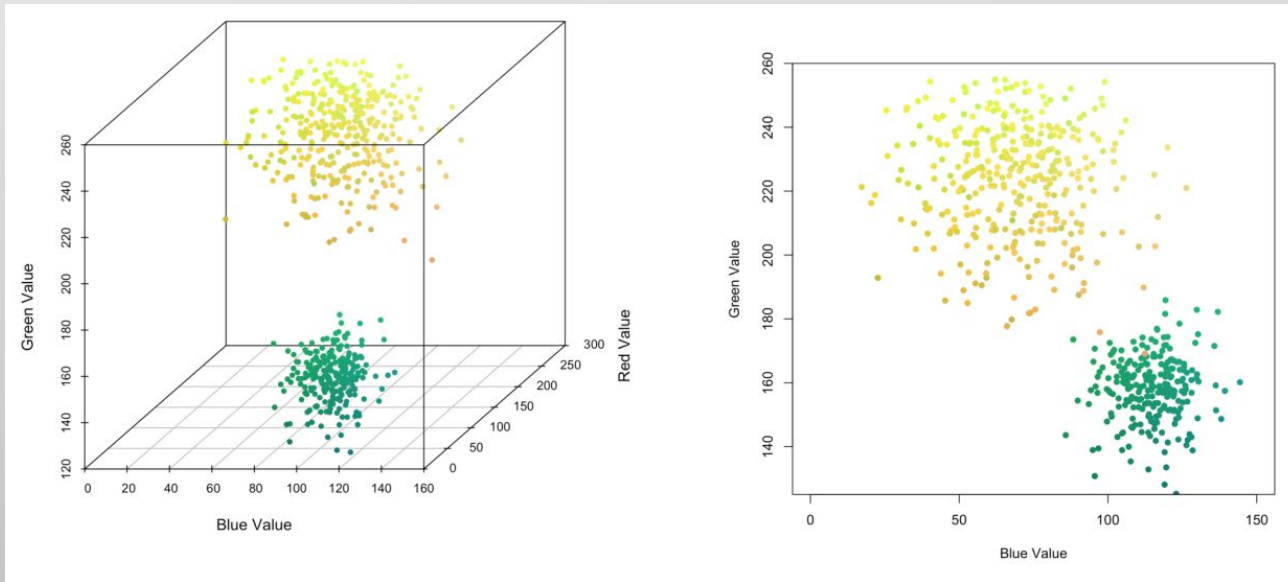
Animation Nachbarschaftsfunktion θ

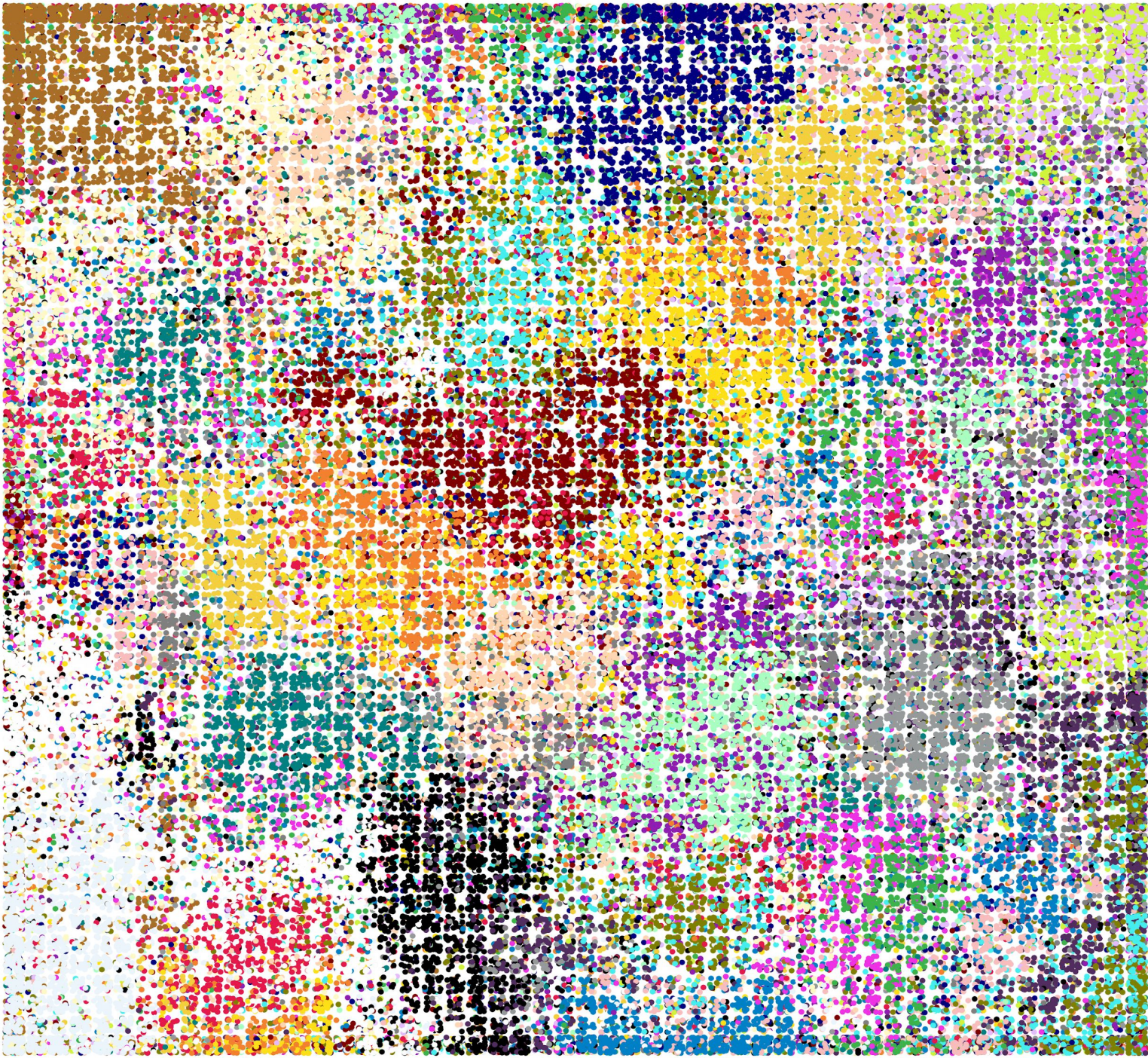


Schritt 4: Nachbarknoten der BMU und wieder deren Nachbarknoten wandern mit der BMU mit (abhängig von der Nachbarschaftsfunktion)

Animation Algorithmus

Zwei Farbcluster: gelb und grün





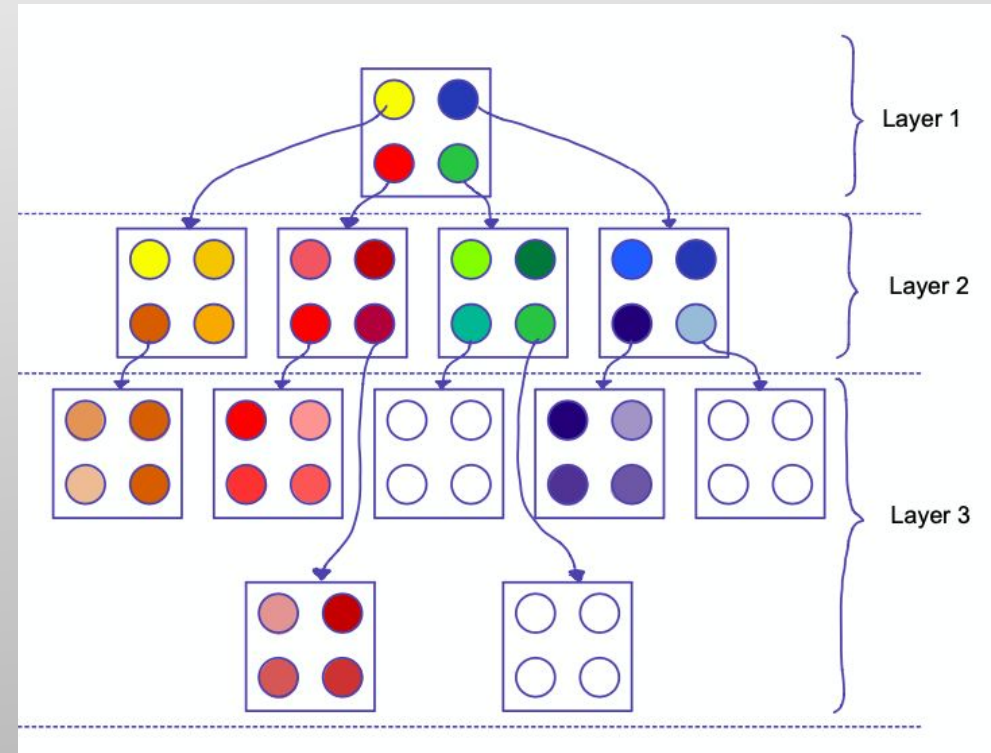
EINSATZBEREICHE

EINSATZBEREICHE

- Automatische Spracherkennung
- Organisation von großen Textarchiven
- Wolkenklassifizierung aus Satellitenbilder
- Klassifizierung verschiedener Input Daten
- Analyse elektrischer Signale im Gehirn

Hierarchische Strukturierung von Musikstücken

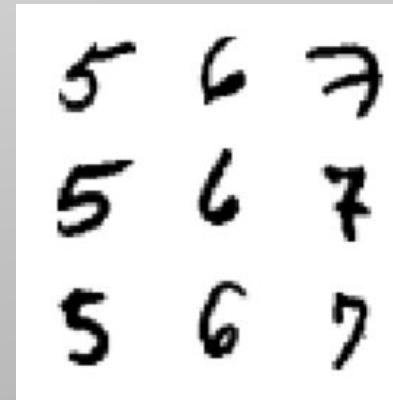
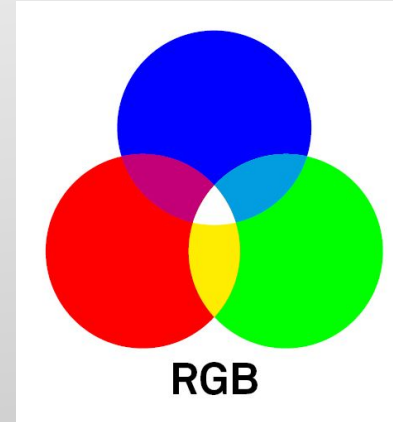
- mittels menschlichem Empfinden
- Hierarchische Wachsenden Selbstorganisierende Karten
- MusicMiner



MNIST Kohonen Netzwerk

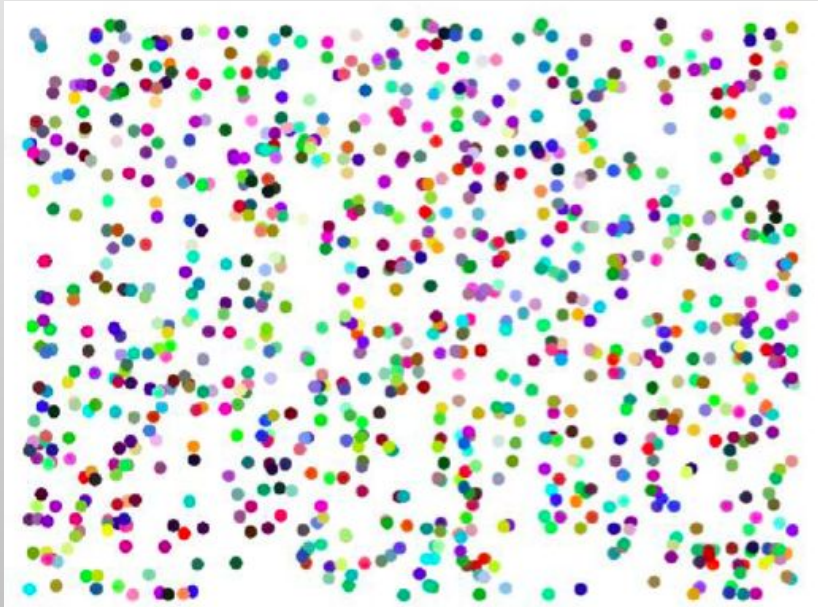
Trainingsdatensätze:

- RGB - Farbmodell
- Irisblüten Datensatz
- MNIST Datensatz

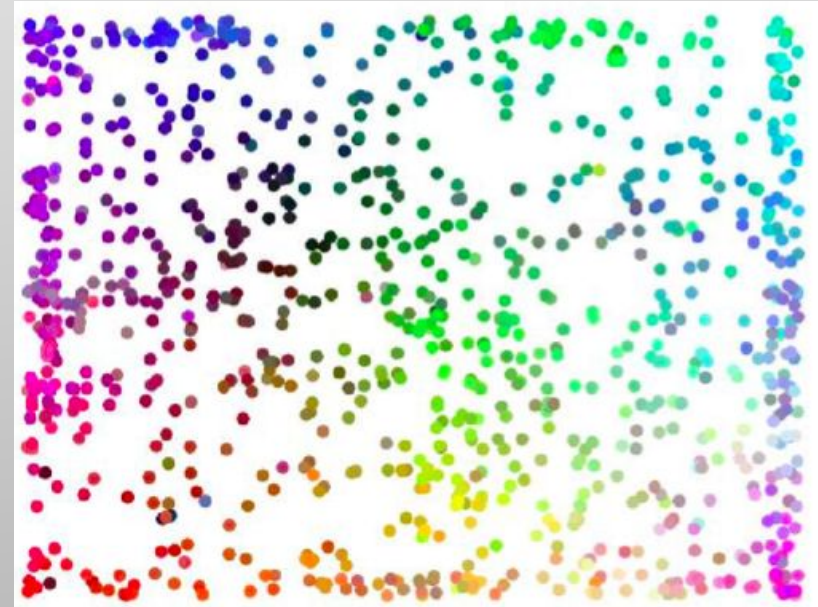


MNIST Kohonen Netzwerk

- Ungeordneter RGB Input



- Geordneter Output



MNIST Kohonen Netzwerk



- 88.000 Buchstaben

Vorteile und Nachteile

Vorteile:

- Algorithmus leicht verständlich und anwendbar
- auf jedem Computer ausführbar
- intuitive Ergebnisse
- biologisch plausibel

Nachteile:

- Überlaufen der Knoten
- Training / Lernphase aufwendig

Emotionserkennung beim Menschen

- Affective Computing/Emotional Artificial Intelligence
- Menschliche Emotionen sehr komplex
- Emotionsanalyse mittels maschinellen Lernverfahren durch:
 - Mimik oder Spracherkennung
 - Physiologische Signale

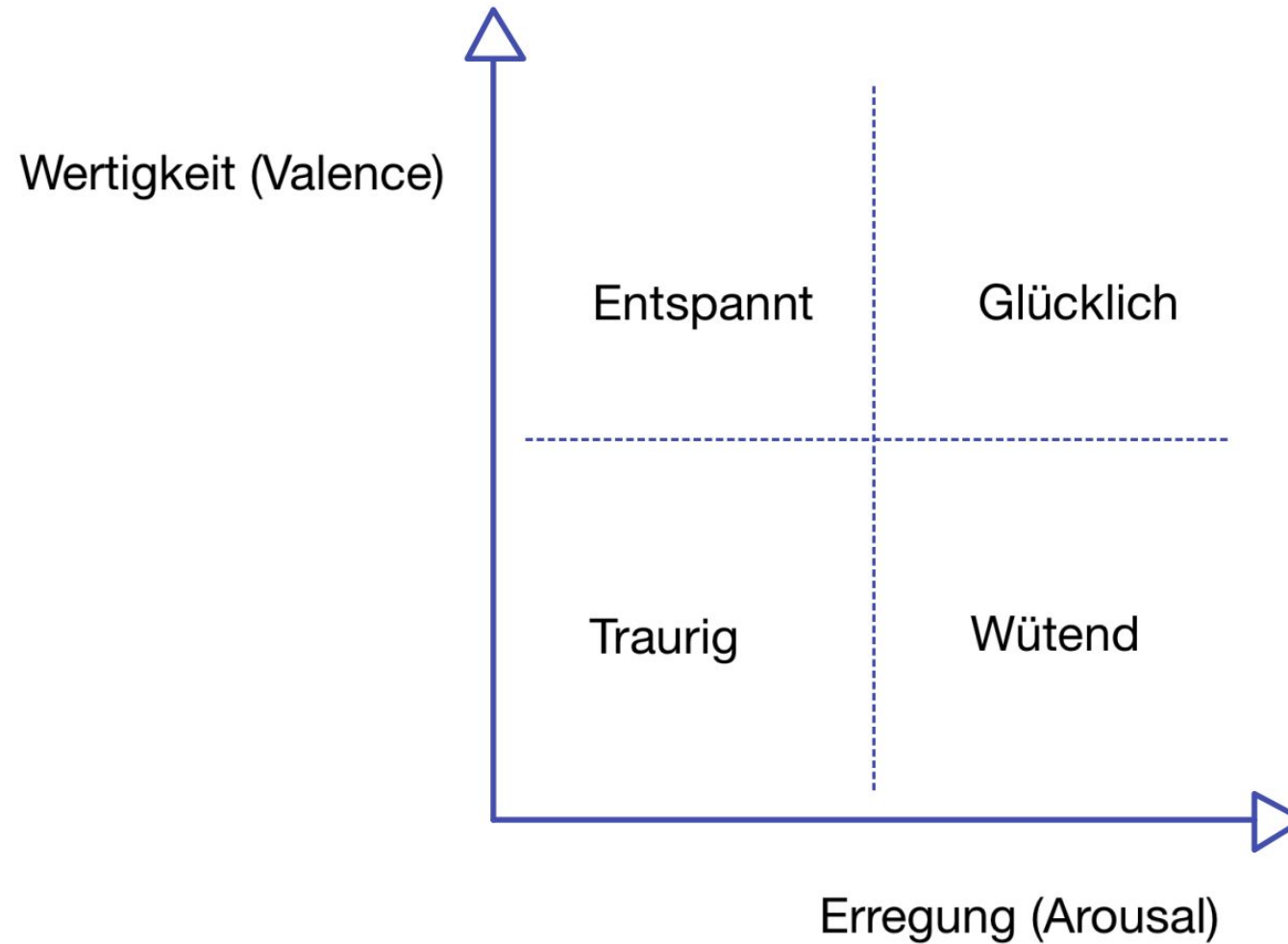
Studien über Emotionserkennung

- Vielzahl an Studien, unterscheiden sich in Wahl der maschinellen Lernverfahren
- Mit künstlichen neuronalen Netzen größtes Potenzial an Verbesserung
- Testungen mit modifizierten Back Propagation Neural Network (BPN) und Kohonen-Netzwerken
- EEG Signale des DEAP Datensatzes verwendet
 - <http://www.eecs.qmul.ac.uk/mmv/datasets/deap/>

Überblick der menschlichen Emotionen

- Bewusste Erfahrung, gekennzeichnet durch Aktivität des Gehirns
- Emotionserkennung mit Einführung von HCI an Bedeutung gewonnen
- Zwei Aspekte von Emotionen
 - 1) Wertigkeit (positiv bzw negativ)
 - 2) Erregung (Ausmaß der Emotion)

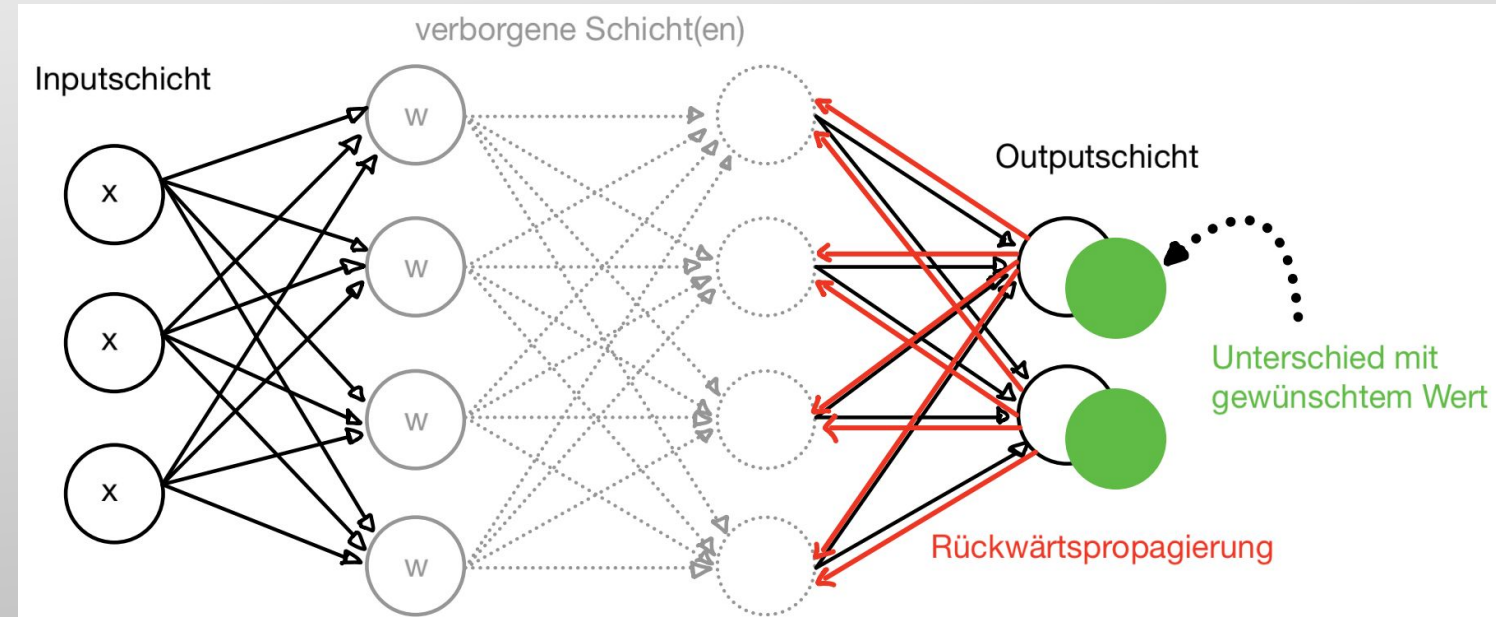
Überblick der menschlichen Emotionen



Wertigkeits-/Erregungszusammenhang

Konventionelle Künstliche Neuronale Netze

- Back propagation neural network (BPN)
- Überwachtes Lernverfahren, das zu Feedforward-Netzen gehört
- Fehlerkorrektur durch Rückwärtsschritte



Back Propagation Neural Network

Modifizierte Künstliche Neuronale Netze

- Circular back propagation neural network (CBPN)
 - Schichten sind kreisförmig angeordnet
 - Berechnungskomplexität verringert
- Deep Kohonen Neural Network (DKNN)
 - Abstract Layer zwischen Input- und Output Schicht
 - Exaktheit der Berechnung verbessert

Ergebnisse

- mithilfe von CBPN und DKNN durchschnittliche Genauigkeit von 95-98%
- Verbesserung von 3-4%
- Berechnungskomplexität verringert

Quellen

- Joachim Steinwender, R. S. (2020). *Neuronale Netze programmieren mit Python*. Bonn: Rheinwerk Computing.
- Algobeans.com. (November 2017). *ALGOBEANS Layman tutorials in machine learning*. Von <https://algobeans.com/2017/11/02/self-organizing-map/> abgerufen
- Hemanth, D., Anitha, J. and Son, L., 2018. Brain signal based human emotion analysis by circular back propagation and Deep Kohonen Neural Networks. *Computers & Electrical Engineering*, 68, pp.170-180.
<https://doi.org/10.1016/j.compeleceng.2018.04.006>
- Back Propagation Neural Network: Explained With Simple Example. <https://www.guru99.com/backpropagation-neural-network.html>
- Künstliche Intelligenz in 5 Minuten erklärt: <https://www.youtube.com/watch?v=3RsmRMqX2IY&t=19s>
- Teuvo Kohonen: https://de.wikipedia.org/wiki/Teuvo_Kohonen
- Künstliche Neuronale Netze - Aufbau & Funktionsweise: <https://jaai.de/kuenstliche-neuronale-netze-aufbau-funktion-291/>
- Aufbau und Funktionsweise von SOM: <https://tud.gucosa.de/api/gucosa%3A25430/attachment/ATT-0/>
- Neuronale Netze - Eine Einführung: <http://www.neuronalesnetz.de/kohonennetze1.html>
- ART-KOHONEN neural network:
https://www.sciencedirect.com/science/article/abs/pii/S0888327003000736?casa_token=W6t7EszglUQAAAAA:X_azUGJfVJMYvX9yZQiUs8Do5dz_PUabzIJPiZhlc79HoKoNYclQZ0lleSjg4JLDI0IIQ9L9vOQ
- Aufbau: <http://www.neuronalesnetz.de/kohonennetze2.html>
- Hierarchical Structuring of Music: https://www.researchgate.net/profile/Elias_Pampalk/publication/2542477_Using_Psycho-Acoustic_Models_and_Self-Organizing_Maps_To_Create_Hierarchical_Structuring_of_Music_by_Sound_Similarity/links/0c960519ea241319f9000000.pdf
- Artificial Neural Networks: Kohonen Self-Organising Map: <https://eklavyafcb.github.io/som.html>
- Machine Learning Repository: [http://archive.ics.uci.edu/ml",institution = "University of California, Irvine, School of Information and Computer Sciences](http://archive.ics.uci.edu/ml)

Quellen Abbildungen

- Aufbau eines Neurons:
https://de.123rf.com/photo_56921155_anatomie-eines-typischen-menschlichen-neuron-axon-synapse-dendriten-mitochondrium-myelinscheide-knot.html
- Teuvo Kohonen: <https://de.wikipedia.org/wiki/Datei:Teuvo-Kohonen-2.jpg>
- “SOM mit 2 Eingabe Neuronen und einer 3x3 Kohonen Schicht”:
Neuronale Netze programmieren mit Python..., Joachim Steinwendner, Roland Schwaiger, S. 333, Abbildung 12.7
- “Euklidischer Abstand” Vektorgrafik: Neuronale Netze programmieren mit Python..., S. 335, Abbildung 12.8
- “Lernrate” Graph: Neuronale Netze programmieren mit Python..., S. 335, Abbildung 12.9
- “Nachbarschaftsfunktion” Grafik: Neuronale Netze programmieren mit Python..., S. 337, Abbildung 12.10
- Animation Nachbarschaftsfunktion: <https://annalyzin.files.wordpress.com/2017/11/som-explainingif.gif?w=609&h=299>
- Animation Algorithmus: <https://annalyzin.files.wordpress.com/2017/11/som-iterateanimate.gif?w=620>
- Farb Cluster 3D Würfel: <https://annalyzin.files.wordpress.com/2017/11/som-rawplot2d3d.png?w=620>
- Somatosensorischer Cortex: https://anthrowiki.at/images/thumb/f/ff/Somatosensorischer_Cortex.png/250px-Somatosensorischer_Cortex.png
- Hemanth, D., Anitha, J. and Son, L., 2018. Brain signal based human emotion analysis by circular back propagation and Deep Kohonen Neural Networks. Computers & Electrical Engineering, 68, pp.170-180.
<https://doi.org/10.1016/j.compeleceng.2018.04.006>
- Trainierte Datensätze: <https://eklavyafcb.github.io/docs/KohonenThesis.pdf>
- RGB Farbmodell: <https://www.flyeralarm.com/blog/de/cmyk-vs-rgb-die-wichtigsten-unterschiede>
- EMNIST Datensatz: https://assets.digitalocean.com/articles/handwriting_tensorflow_python3/wBCHXd.png
- Irisblüte: <https://naturfotografen-forum.de/o382163-Irisblüte>
- ART-KOHONEN neural network:
https://www.sciencedirect.com/science/article/abs/pii/S0888327003000736?casa_token=W6t7EszglUQAAAAA:X_azUGJfVJMYvX9yZQiUs8Do5dzPUabzIJPiZhlc79HoKoNYclQZ0lleSjq4JLDI0IIQ9L9vOQ

A vertical graphic on the left side of the page featuring a stylized circuit board pattern with various lines and nodes in shades of blue and white.

Danke für Ihre Aufmerksamkeit

BRUCKNER, HANGLER,
MOSER-SCHWAIGER & SPERL

