

Anwendung künstlicher Neuronaler Netze in der Optimierung

Mike Steglich

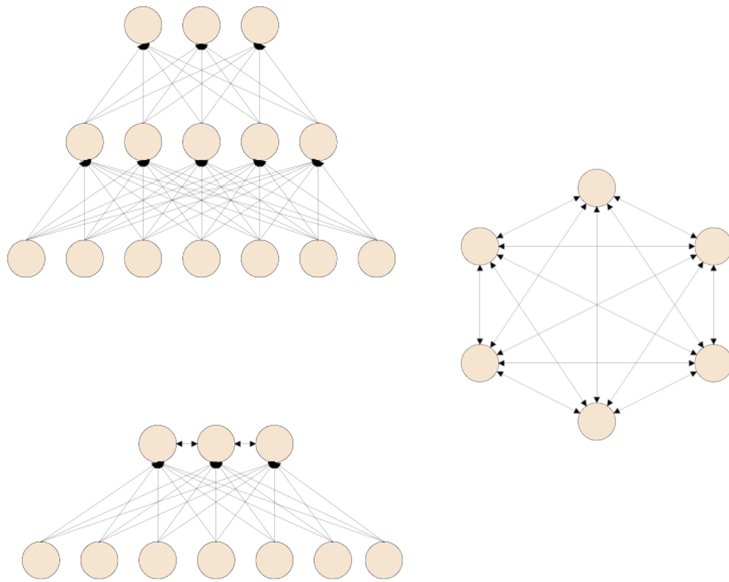
Technische Hochschule Wildau

Künstliche Intelligenz -> verständlich

06. Juli 2020

Zwei gegensätzliche Welten?

Künstliche Neuronale Netze (KNN)



Künstliche Neuronale Netze sind parallel verbundene Netzwerke aus einfachen Elementen in hierarchischer Anordnung, die mit der Welt in der selben Art wie biologische Nervensysteme interagieren sollen.

(Kohonen 1982)

Abbildung eines Inputvektors e auf einen Outputvektor o

Optimierung

$z = F(\mathbf{x}) \rightarrow \max! \text{ (oder min!)}$
u. d. N.

$$g_i(\mathbf{x}) \begin{cases} \geq \\ = \\ \leq \end{cases} 0; i \in \{1, 2, \dots, I\}$$

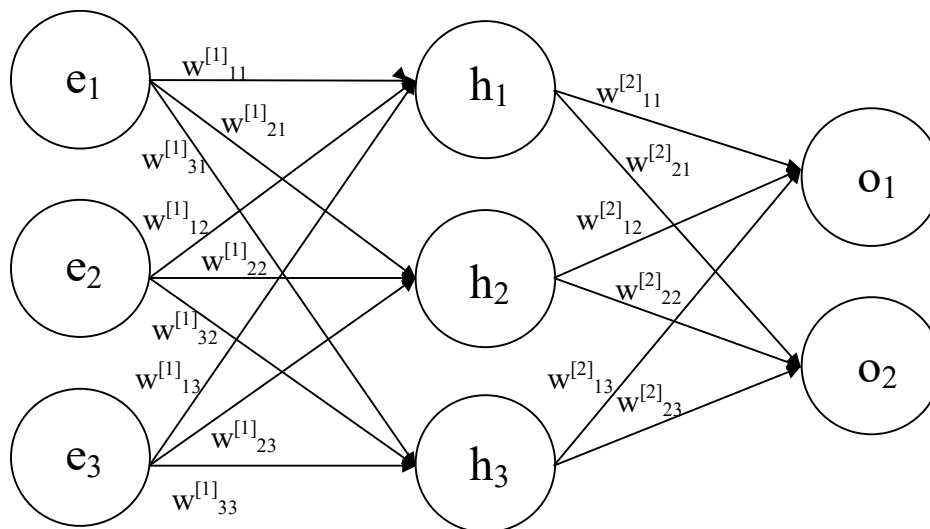
$$\mathbf{x} \geq 0$$

?

Bestimmung des Lösungselementes mit dem besten Zielfunktionswert aus einer Menge zulässiger Lösungselemente.

Künstliches Neuronales Netz (KNN)

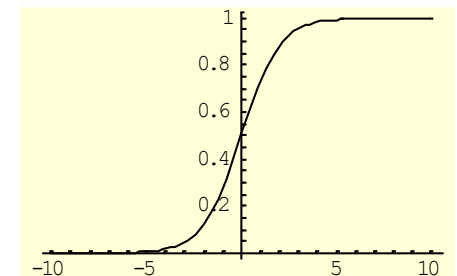
- KNN sind als **gerichteten Graphen** darstellbar, die sich durch eine Menge von **Knoten** (Neuronen) und eine Menge von bewerteten **Kanten** beschreiben lassen.
- Die über die gewichteten Kanten übertragenen Argumente werden in den Knoten (als eigentliche Berechnungseinheiten eines künstlichen neuronalen Netzes) über bestimmte Berechnungsvorschriften zu einem Outputwert moduliert. (Zell 1997)



(Steglich 2001, S. 215)

$$h = s(W^{[1]} \cdot e)$$

$$o = s(W^{[2]} \cdot h)$$



Künstliches Neuronales Netz (KNN)

- Die Aufgabe eines KNN besteht darin, die Abbildung vom Inputvektor e auf den Outputvektor o zu erlernen und zu repräsentieren.



- Ein KNN arbeitet in zwei Modi:
 - einer Lernphase und
 - einer Ausführungsphase (Recallphase).
- In der Lernphase werden die Kantengewichte zur Abbildung $F: R^L \rightarrow R^J$ in der Lernphase durch einen Lernalgorithmus festgelegt.
 - Überwachtes Lernen erfolgt anhand von Beispielen mit Eingabe- und zugehörigen Ausgabewerten.
 - Unüberwachtes Lernen erfolgt anhand von Beispielen einzig mit Eingabewerten.

(Aggarwal 2018, S. 4ff, Rojas 1996, S. 29)

Zwei miteinander verbundene Welten

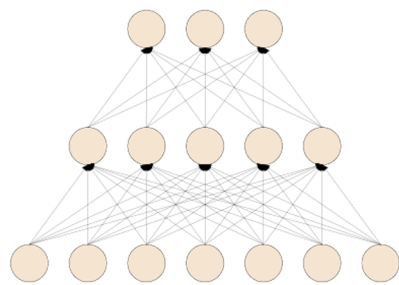
KNN als Pre- bzw. Post-Processing der Daten für eine Optimierungsanwendung

Optimierung als Bestandteil des KNN-Lernalgorithmus

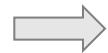
KNN als Bestandteil eines Optimierungsalgorithmus

Zwei miteinander verbundene Welten

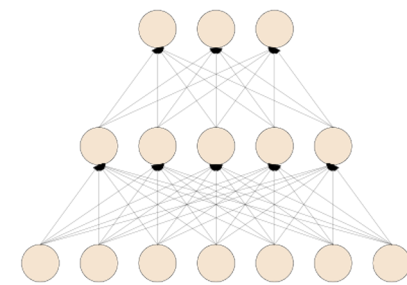
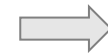
- KNN als Pre- bzw. Post-Processing der Daten für eine Optimierungsanwendung



Data pre-processing
zur Bestimmung von
Problem Daten



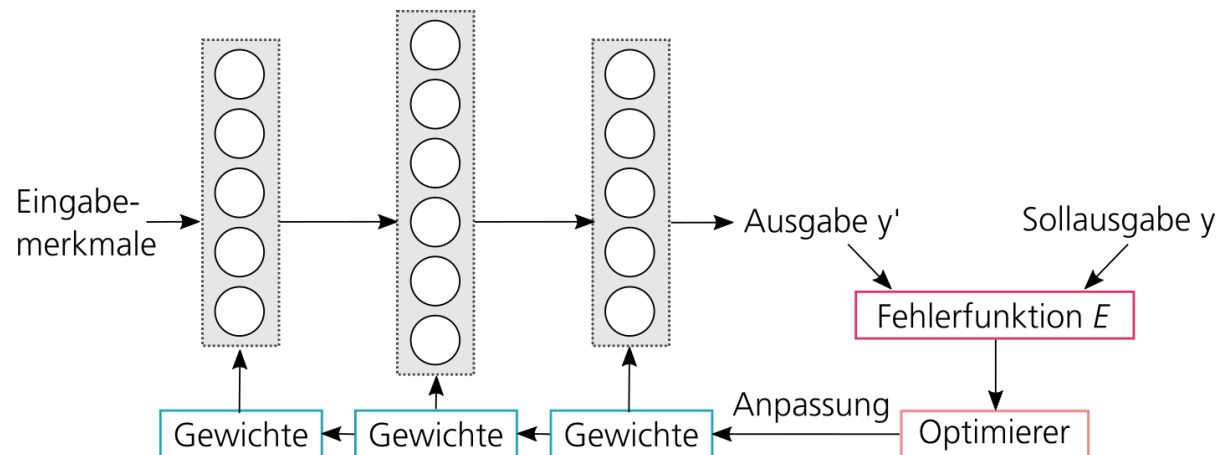
$$\begin{aligned} z = F(\mathbf{x}) &\rightarrow \max! \text{ (oder min!)} \\ &\text{u. d. N.} \\ g_i(\mathbf{x}) &\begin{cases} \geq \\ = \\ \leq \end{cases} 0; i \in \{1, 2, \dots, I\} \\ \mathbf{x} &\geq 0 \end{aligned}$$



Data post-processing
zur Analyse der
Lösung

Zwei miteinander verbundene Welten

- Optimierung als Bestandteil des KNN-Lernalgorithmus



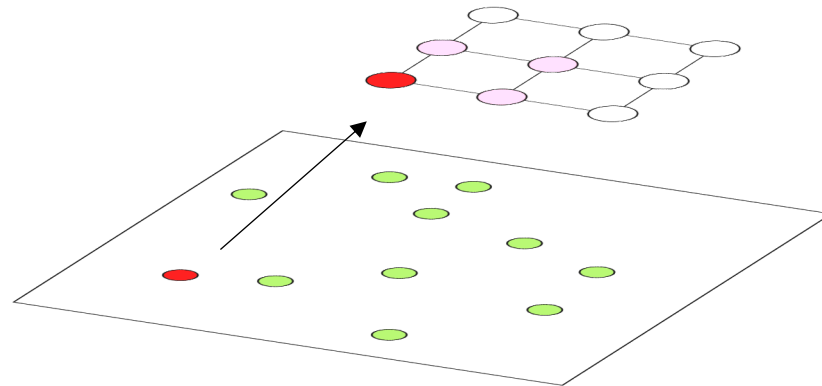
(Schaaf 2020)

Zwei miteinander verbundene Welten

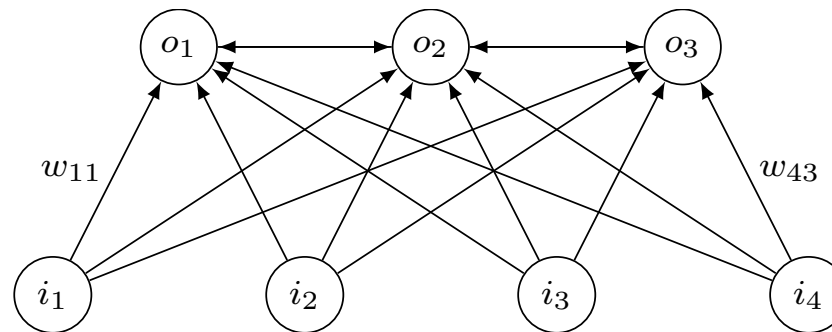
- KNN als Bestandteil eines Optimierungsalgorithmus (i.d.R. Heuristik)
 - KNN als eigenständiger Optimierungsalgorithmus
 - KNN als Bestandteil eines Optimierungsalgorithmus

Self-organising maps (SOM)

- A SOM is able to find a topographical projection $f : \Theta \rightarrow \Omega$, where Θ describes L input patterns of attributes and Ω a usually one- or two-dimensional representation of the input patterns (Kohonen 1982).



- A SOM can be described as a feed-forward network consisting of two layers of units. The I input units are completely connected with the O output units in the second layer. The weights on the edges between the input and output units can be formulated as a Matrix $\mathbf{w} = \{w_{mn} \in \mathbb{R} \mid m \in \{1, 2, \dots, I\}, n \in \{1, 2, \dots, O\}\}$ (Kohonen 1982).



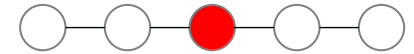
(Steglich 2019)

Self-organising maps (SOM)

- The projection $f: \Theta \rightarrow \Omega$ has to be found during the so-called learning phase, which means that the following steps have to be carried out several times (Kohonen, 2001, p. 106).

1. Initialisation: Initialise all weights by assigning random values.
2. Propagation: Choose randomly an input pattern $\theta_l \in \Theta = \{\theta_m | m \in \{1, 2, \dots, I\}\}; l \in \{1, 2, \dots, L\}$ and determine the winner unit η

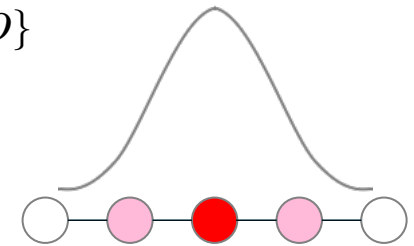
$$\eta = \arg \min_{n \in \{1, 2, \dots, O\}} \|\theta_l - \mathbf{w}_n\|^2$$



3. Updating weights for the winner and its neighbourhood :

$$\Delta w_{mn} = \alpha \cdot \delta(n, \eta) \cdot (\theta_m - w_{mn}) ; m \in \{1, 2, \dots, I\}, n \in \{1, 2, \dots, O\}$$

$$\delta(n, \eta) = \exp\left(-\frac{(n - \eta)^2}{2\sigma^2}\right) ; n \in \{1, 2, \dots, O\}$$

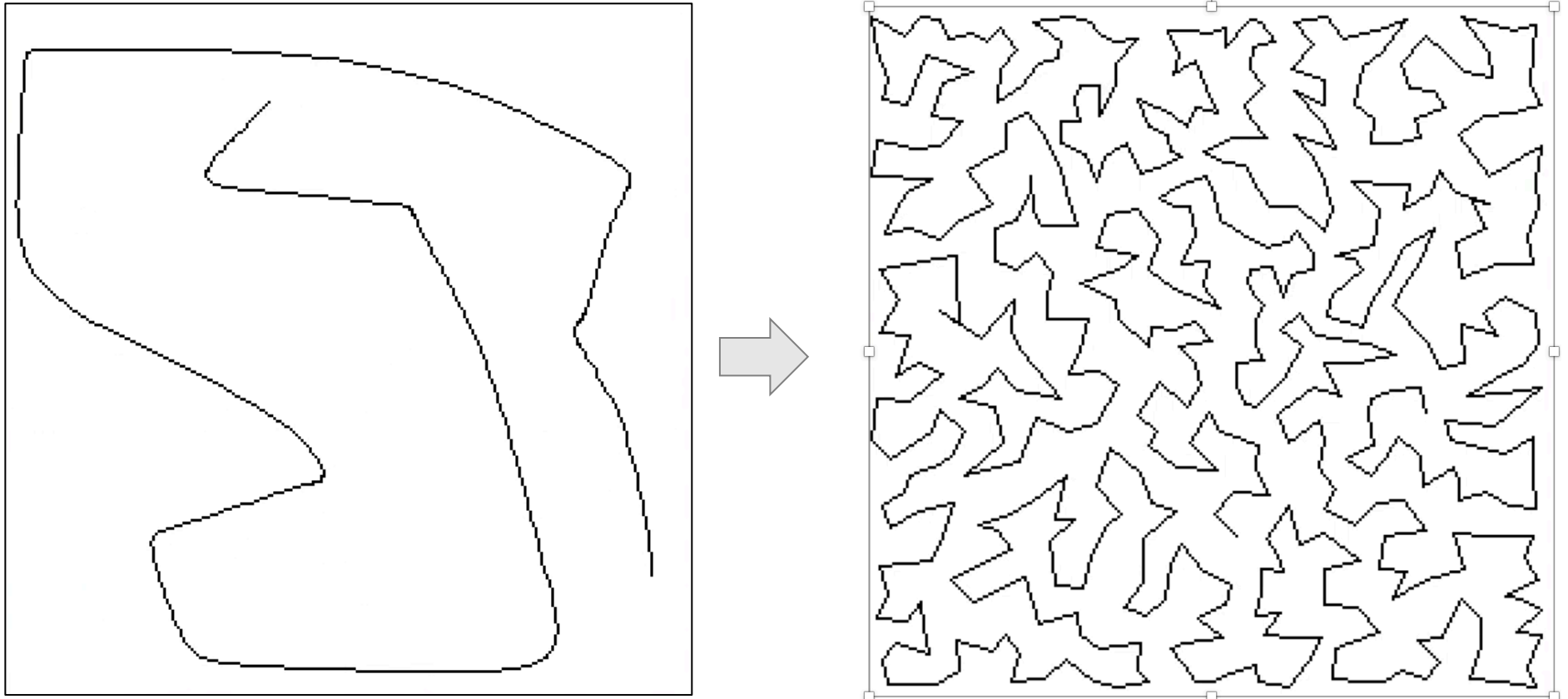


4. Stop, if a stopping rule is satisfied or else continue with step 2.

- Afterwards, for all patterns $\theta_l \in \Theta$ the corresponding outputs have to be determined by applying the winner-takes-all function:

$$\omega_l = \arg \min_{n \in \{1, 2, \dots, O\}} \|\theta_l - \mathbf{w}_n\|^2 ; l \in \{1, 2, \dots, L\}$$

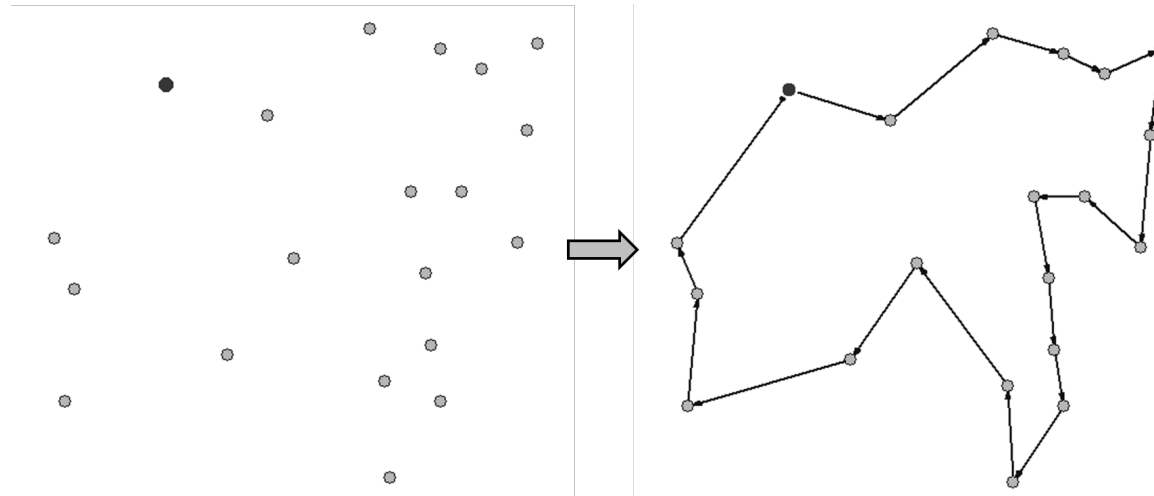
Self-organising maps (SOM)



(Eigenes Experiment)

Lösung des symmetrischen Rundreiseproblems mittels selbstorganisierter Karten

- Mit diesem Problem ist die kürzeste geschlossene Reiseroute über eine Anzahl von Orten zu finden. Die Distanzen zwischen den Orten sind bekannt. Jeder Ort muss mindestens bzw. möglichst genau einmal aufgesucht werden. Anschließend wird zum Ausgangsort zurückgekehrt (Steglich et al. 2016, S. 278 f).



$$\sum_{[i,j] \in A} c_{ij} \cdot x_{ij} \rightarrow \min!$$

s.t.

$$\sum_{[i,j] \in \delta(i)} x_{ij} = 2 \quad ; i \in N$$

$$\sum_{[i,j] \in \delta(S)} x_{ij} \leq |S| - 1 \quad ; S \subset N, |S| \geq 2$$

$$x_{ij} \in \{0,1\} \quad ; [i,j] \in A$$

Parameter:

N Menge der Knoten

A Menge der Kanten

$\delta(i)$ Menge der mit dem Knoten i inzidenten Kanten

$\delta(S)$ Menge der mit den in der Menge S enthaltenen Knoten verbundenen Kanten

c_{ij} Distanz der Kante $[i,j] \in A$

Variablen:

x_{ij} Kantenindikatorvariable $[i,j] \in A$

(Steglich et al. 2016, S. 281 f., Chen et al. 2010, S. 146 ff., Ghiani et al. 2013, S. 368 ff.)

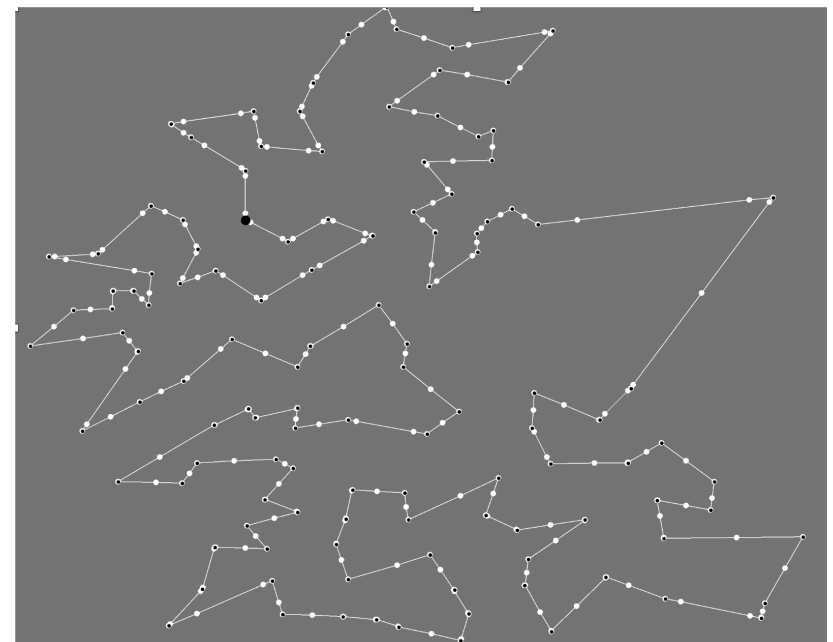
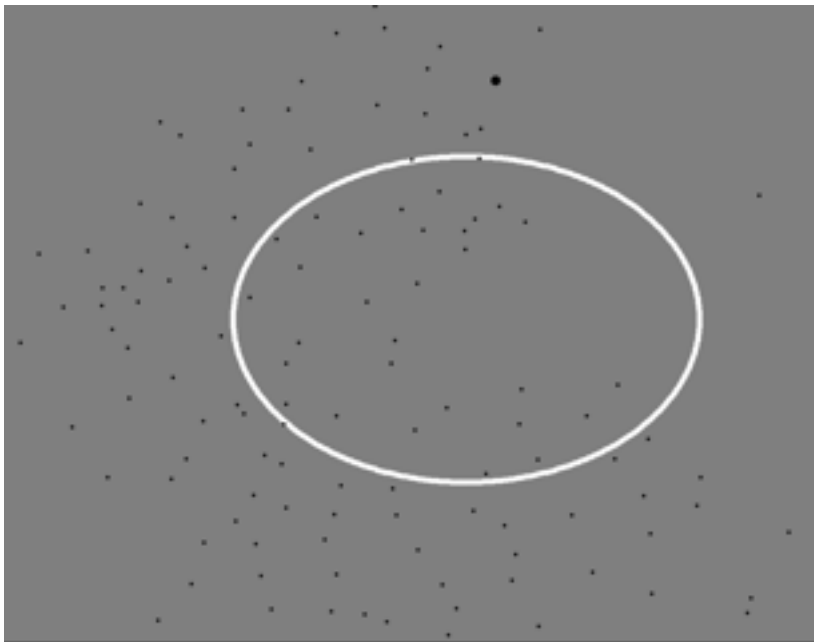
Lösung des symmetrischen Rundreiseproblems mittels selbstorganisierter Karten

- Die Anzahl möglicher Touren beträgt bei einem symmetrischen Rundreiseproblem $(n - 1)!/2$.
- Bei 10 Knoten existieren $\frac{9!}{2} = 181.440$ unterschiedliche Touren.
- Bei 26 Knoten existieren $\frac{25!}{2} = 7,76 \cdot 10^{24}$ Varianten, die bei einer Bearbeitungszeit von einer Millisekunde pro Variante $7,76 \cdot 10^{24}$ Millisekunden bei einer vollständigen Enumeration benötigen würden.
- Zum Vergleich: Das geschätzte Alter des Universums beträgt 13,7 Milliarden Jahre $\approx 4,32 \cdot 10^{24}$ Millisekunden (Vgl. Sperkel et al. (2003)).

(Vgl. Steglich et al. 2016, S. 285.)

Lösung des symmetrischen Rundreiseproblems mittels selbstorganisierter Karten

- Bei n vorliegenden Städten besteht die Ausgangsbeschriftung aus einer ein-dimensionalen Karte mit insgesamt n ringförmig angeordneten Knoten.
- Als Eingaben werden die Koordinaten der Städte verwendet. Durch den Lernprozess spezialisiert sich jeder Knoten der Ausgangsbeschriftung auf eine Stadt. Es entsteht eine Rundreise über alle Städte.



(Eigenes Experiment unter Verwendung der Software Sikone / Uni Halle)

Lösung des kontinuierlichen, unkapazitierten p-Median-Problems mittels SOM

$$\sum_{i \in S} \sum_{j \in D} x_{ij} \cdot d((u_i, v_i), (x_j, y_j)) \rightarrow \min!$$

u. d. N.

$$\sum_{i \in S} x_{ij} = 1 \quad ; j \in D$$
$$x_{ij} \in \{0,1\} ; i \in S, j \in D$$

Indexmengen:

S – Menge der Standorte, $|S| = p$

D – Menge der Bedarfsorte

Indizes:

i – Index der Standorte, $i \in S$

j – Index der Bedarfsorte, $j \in D$

Parameter:

p – maximale Anzahl von Standorten

(x_j, y_j) – Koordinaten des Bedarfsortes $j \in D$

Variablen:

(u_i, v_i) – Koordinaten des Standorts $i \in S$

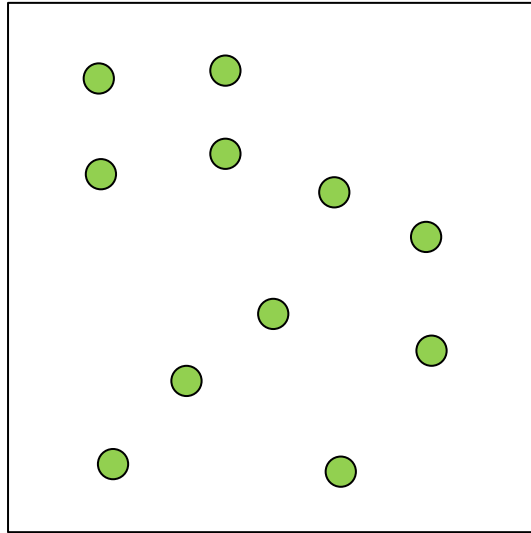
x_{ij} – Zuordnungsvariable des Bedarfsortes j zum Standort i

$d(\cdot)$ – Distanzfunktion zwischen einem Standort und einem Bedarfsort

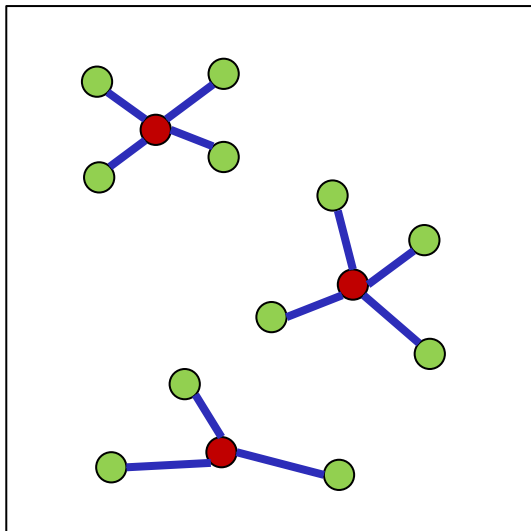
(Steglich et al. 2016, S. 402 ff., Winkels 2012, S. 416 f.)

Lösung des kontinuierlichen, unkapazitierten p-Median-Problems mittels SOM

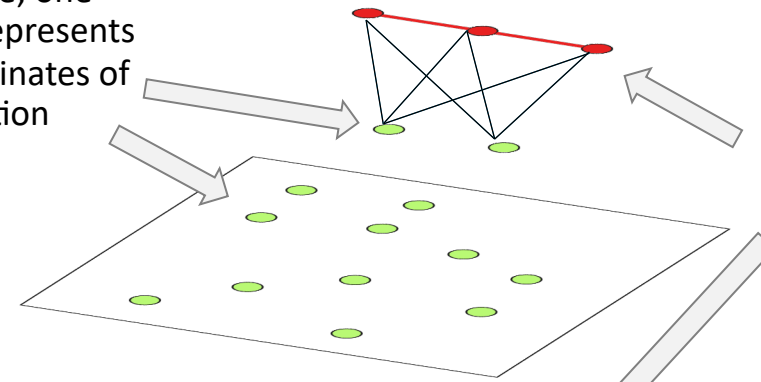
Map with the locations of the destinations



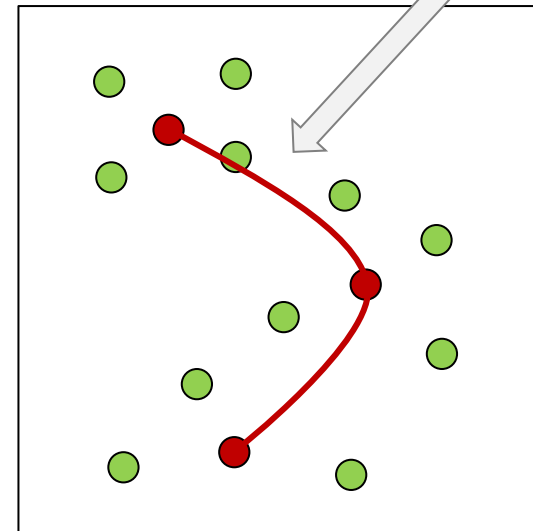
The winner unit per input (location of a demand node) represents the allocation of the destination to the source



One input unit per coordinate; one pattern represents the coordinates of a destination



There are p output units, which represent areas of the map after the learning phase. The weights of the output units are the coordinates of the sources.



(Steglich 2019, Lozano et al. 1998, Hsieh and Tien 2004, Aras et al. 2006)

Lösung des diskreten, kapazitierten p-Median-Problems mittels SomAla

- Es sind die Standorte für eine vorgegebene Anzahl p von logistischen Knoten in einem Logistiknetzwerk zu finden, so dass die Summe der gewichteten Distanzen der Bedarfsorte zum jeweils nächstgelegenen der gewählten Standorte minimal wird (Daskin and Maass, 2015):

$$\begin{aligned} & \sum_{(i,j) \in A} d_{ij} \cdot x_{ij} \rightarrow \min \\ & \text{s.t.} \\ & \sum_{i \in N} x_{ij} = 1 \quad ; j \in N \\ & \sum_{j \in N} q_j \cdot x_{ij} \leq Q \cdot y_i \quad ; i \in N \\ & \sum_{i \in N} y_i = p \\ & x_{ij} \in \{0, 1\} \quad ; (i, j) \in A \\ & y_i \in \{0, 1\} \quad ; i \in N \end{aligned}$$

Parameter:

N	Menge der Bedarfsorte (zugleich potentielle Standorte der Angebotsorte)
A	Menge der Kanten , $A = \{(i, j) \mid i \in N, j \in N\}$
p	Anzahl der Angebotsorte
q_j	Bedarf des Bedarfsortes $j \in N$
Q	Angebot der Angebotsorte
d_{ij}	Distanz zwischen den Knoten $i \in N$ und $j \in N$

Variablen:

y_i	Standortvariablen, $i \in N$
x_{ij}	Zuordnungsvariable für einen Bedarfsort $j \in N$ zum Angebotsort $i \in N$

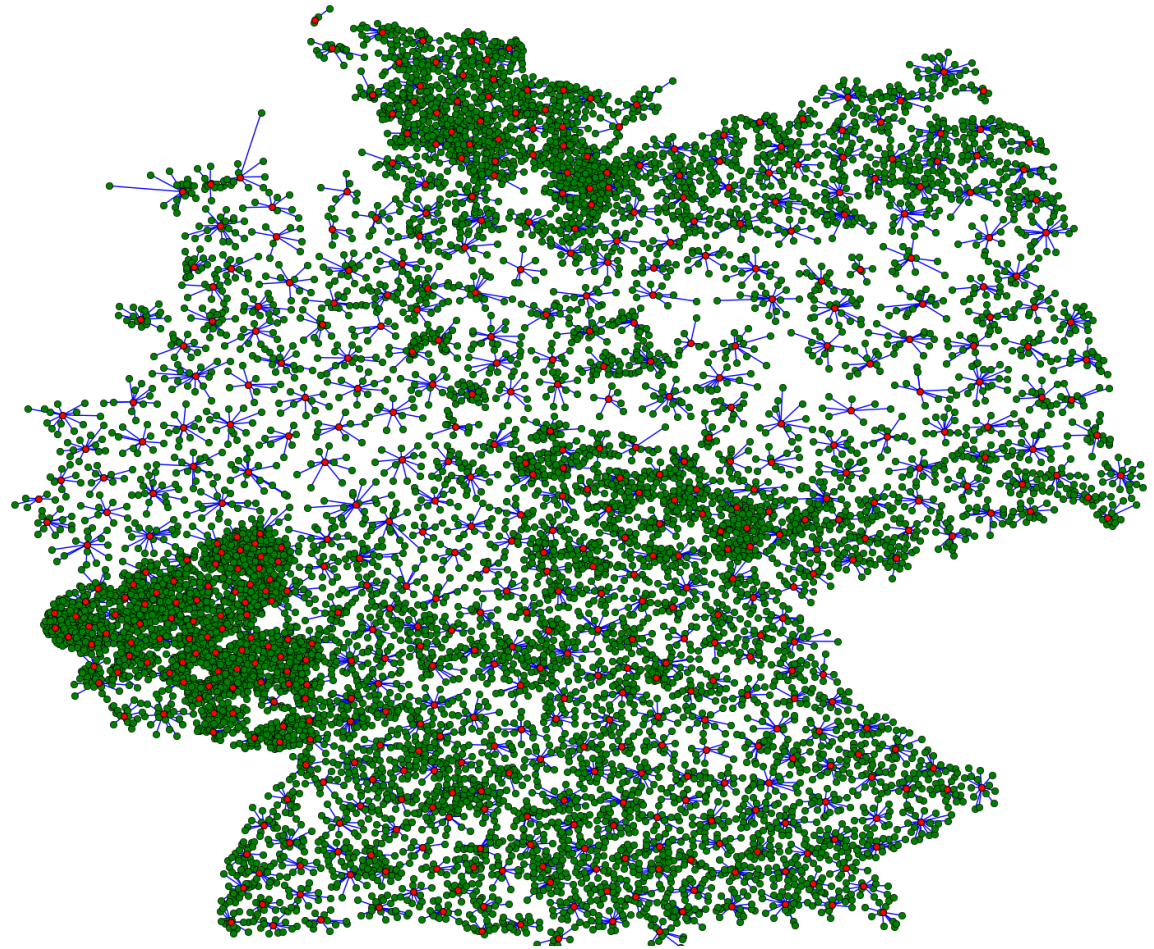
SomAla

- SomAla is a new hybrid heuristic for the capacitated p-median problem (CPMP) which combines a self-organising map (SOM), integer linear programming, an alternating location-allocation algorithm (ALA) and a partial neighbourhood optimisation.
 - 1. SOM-GAP-based heuristic to solve a continuous, capacitated p-median problem**
 - A SOM is used to solve a continuous, uncapacitated p-median problem.
 - This solution is the basis to solve a continuous, capacitated p-median problem using a generalised assignment problem (GAP).
 - 2. Capacitated alternating location-allocation heuristic**
 - The locations are modified by determining new medians for each allocation cluster.
 - The allocation of the destinations to the sources is based on a GAP which is solved by using a size reducing technique and variable relaxing-fixing heuristic.
 - 3. Partial neighbourhood optimisation heuristic**
 - In each step, a CPMP for a subregion, surrounding a selected median, is solved.
 - If the solution improves the objective function value of the entire problem, then the partial solution updates the entire solution.

(Steglich 2019)

g11056 instance

- The g11056 instance is the largest CPMP instance which has been ever solved and published.
- The demand nodes are the 11,056 German cities and communities published by the Statistisches Bundesamt (2017) with their geographical coordinates.
- Four instances with 111 up to 3317 sources
- These instances show that SomAla is able to find good feasible solutions in reasonable computational times (e.g. ~3,500 seconds for for g11056-3317) for very large instances.



(Steglich 2019)

References

Aggarwal, Charu C. 2018, *Neural Networks and Deep Learning*, Springer.

Chen, D.-S., R.G. Batson and Y. Dang (2010): *Applied Integer Programming: Modeling and Solution*, Wiley, Hoboken.

Daskin, M.S., Maass, K.L., 2015. The p-Median Problem. In Laporte, G., Nickel, S. and Saldanha da Gama, F. (eds), *Location Science*. Springer, Cham, Heidelberg, New York, Dordrecht, London, book section 2, pp. 21– 45.

Ghiani, G., G. Laporte and R. Musmanno (2013): *Introduction to Logistics Systems Management*, 2. Aufl., Wiley, Chichester.

Hsieh, K.H., Tien, F.C., 2004. Self-organizing feature maps for solving locationallocation problems with rectilinear distances. *Computers & Operations Research* 31, 10171031.

Kohonen, T., 1982. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics* 43, 59–69.

Kohonen, T., 2001. *Self-Organizing Maps* (3 edn.). Springer, Berlin et al.

Lozano, S., Guerrero, F., Onieva, L., Larraneta, J., 1998. Kohonen maps for solving a class of location-allocation problems. *European Journal of Operational Research* 108, 106–117.

Rojas, Raul 1993, *Theorie der neuronalen Netze*, Springer, Berlin Heidelberg.

References

- Schaaf, Nina 2020, Neuronale Netze: Ein Blick in die Black Box, <https://www.informatik-aktuell.de/betrieb/kuenstliche-intelligenz/neuronale-netze-ein-blick-in-die-black-box.html>, letzter Zugriff: 05. März 2020.
- Steglich, Mike 2001, Mike : Zielwertorientierte Auswertung von Kostenabweichungen, Wiesbaden 2001.
- Steglich, Mike 2019: A Hybrid Heuristic Based On Self-Organising Maps And Binary Linear Programming Techniques For The Capacitated P-Median Problem, in: Proceedings of the 33rd International ECMS Conference on Modelling and Simulation ECMS 2019, Caserta, Italy, p. 267-276.
- Steglich Mike, Dieter Feige und Peter Klaus: 2016, Logistik-Entscheidungen: Modellbasierte Entscheidungsunterstützung in der Logistik mit LogisticsLab, 2. aktualisierte und komplett überarbeitete Auflage, De Gruyter, Berlin und Boston.
- Statistisches Bundesamt, 2017. Statistisches Bundesamt, Gemeinden in Deutschland nach Fläche, Bevölkerung und Postleitzahl am 31.03.2017 (1. Quartal). <https://www.destatis.de/DE/ZahlenFakten/LaenderRegionen/Regionales/Gemeindeverzeichnis/Administrativ/Archiv/GVAuszugQ/AuszugGV1QAktuell.xlsx?blob=publicationFile>. Accessed: 20 September 2017.
- Winkels, H.-M. (2012): Modellbasiertes Logistikmanagement mit Excel: Lösungen von Problemen in der Logistik unter Verwendung der Tabellenkalkulation, DVV, Hamburg.
- Zell, A. 1997, Simulation neuronaler Netze, Oldenbourg.

Danke für Ihre Aufmerksamkeit