



Adaptive Systeme

Unüberwachtes Lernen:

Adaptive Vektor Quantisierung und Kohonen Netze

Prof. Dr. rer. nat. Nikolaus Wulff



Überwachtes Lernen

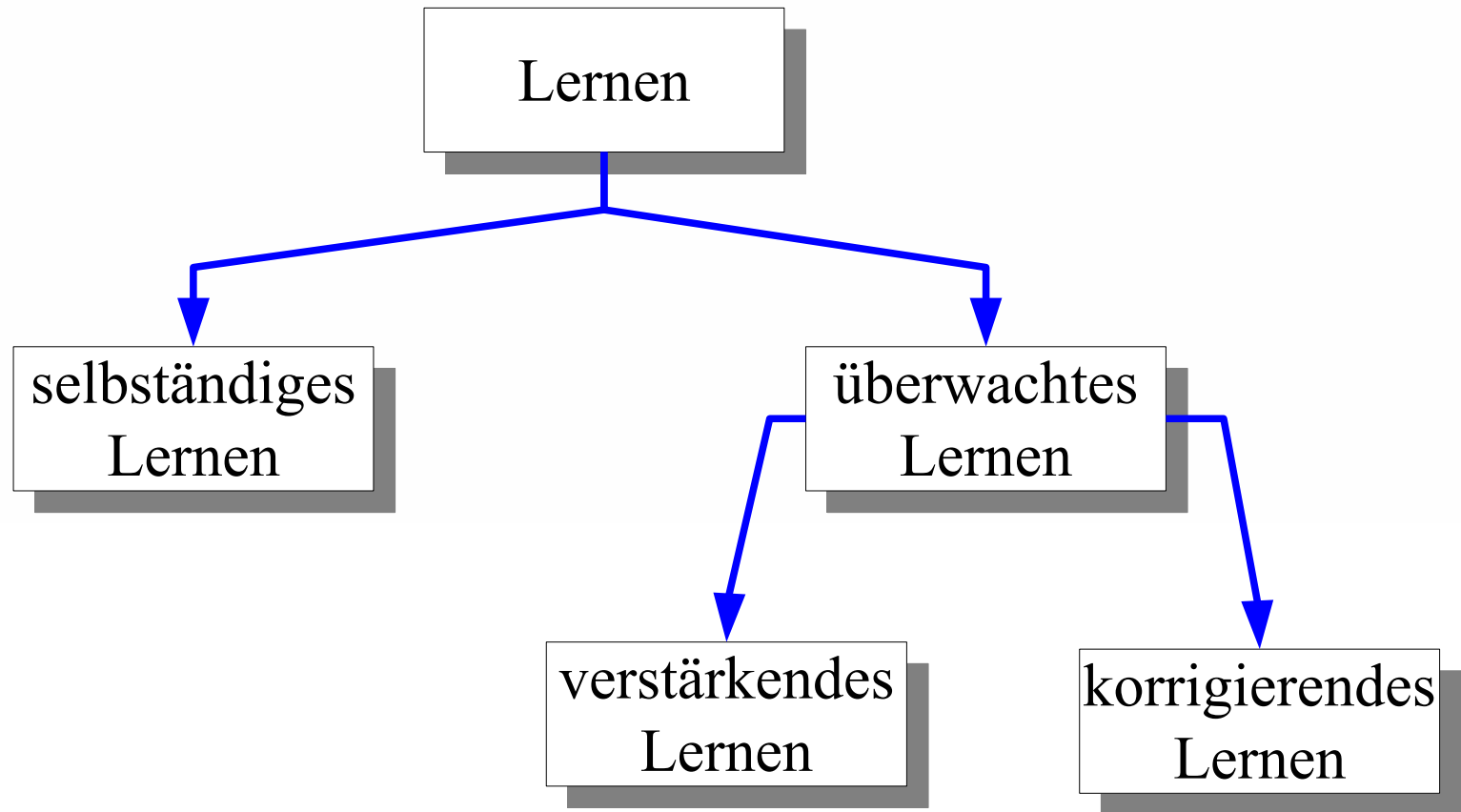
- Alle bis lang betrachteten Netzwerke benötigen einen Trainer, um den Lernerfolg zu überwachen.
- Dies ist eine erhebliche Einschränkung, die biologisch nicht plausibel ist. Ist doch die Natur, die als Vorlage für die Neuronalen Netzwerke dient, in der Lage ohne Lehrer Lernfähigkeit zu entwickeln.
- Auf jeden „kleinen Fehler“ mit einem Lernschritt zu reagieren, entspricht nicht dem Leitbild der KI.
- Offensichtlich fehlt ein entscheidender Baustein in der bisherigen Betrachtung oder einige Vereinfachungen bei der Modellbildung waren zu grob...



Prinzip der bisherigen Verfahren

1. Präsentation der Eingabemuster x .
2. Vorwärtspropagierung der Eingaben durch das Netz (oder dem linearen Filter), um die Ausgabevektoren y zu berechnen.
3. Vergleich von Ausgabevektor y mit der gewünschten Ausgabe d zur Berechnung des Fehlers e .
4. Fehlerkorrektur Δw nach der Hebbschen Regel oder einem Gradientenverfahren, wie z.B. Backward Propagation, durch Änderung der Gewichte.
5. Wiederholung ab 1. bis eine maximale Fehler-schwelle unterschritten wird, d.h. alle Muster erkannt werden.

Einordnung des Lernprozesses



- Überwachtes Lernen erfordert einen Trainer, der entweder „die Wahrheit“ kennt oder aber „belohnt“.



Bestärkendes Lernen

- Beim bestärkenden Lernen wird während des Trainings kein exakter Fehler angegeben, so dass es nicht möglich ist daraus eine Korrektur abzuleiten.
- Statt dessen zeigt der Trainers nur an ob die Antwort „richtig“ oder „falsch“ ist.
- Dies entspricht Rückkopplungsbedingungen der Natur, wo es auch nur Belohnung oder Strafe gibt.
- Der Nachteil dieser Lernverfahren ist die wesentlich längere Lernphase, da sie keinen korrigierenden Fehler verwenden können. Ob die Antwort richtig oder falsch ist, muss dennoch extern bekannt sein.



Unüberwachtes Lernen

- Beim unüberwachten Lernen besteht das Training aus Eingabemustern, ohne dass eine gewünschte Ausgabe bekannt ist. Es existiert kein Hinweis, ob das Netz die Muster richtig oder falsch klassifiziert hat.
- Ohne weitere Anhaltspunkte muss das Netz ähnliche Eingabevektoren identifizieren und auf Gruppen ähnlicher oder benachbarter Neuronen abbilden.
- Dies entspricht dem biologischen Vorbild der höheren Säugetiere, die auf ähnliche Stimuli mit der Ausbildung einer „topologierhaltenden Karte“ reagieren.

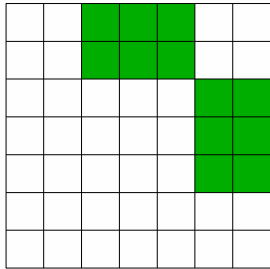


Netzwerktopologien

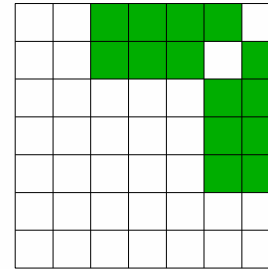
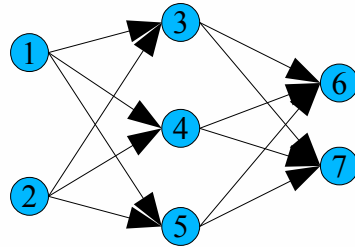
- Eine wesentliche Einschränkung der bisherigen Netzwerktopologien ist, dass es sich immer um Feedforward Propagation von einer Schicht zur Nächsten handelt. Die Natur leistet hier mehr:
- Es ist möglich Topologien zu entwickeln, in der Axonen einzelne Schichten mit einer Kurzschlußverbindung überspringen.
- Die Axonen können auf den eigenen Eingang oder auf die Eingänge der Neuronen der eigenen Schicht oder sogar der weiter vorne liegenden Schichten rückgekoppelt werden. Durch diese *Rückkoppelung* zeigen solche Netze ganz neue Eigenschaften.



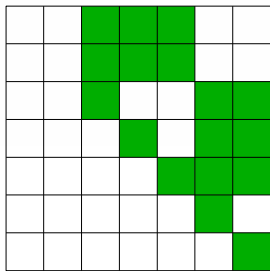
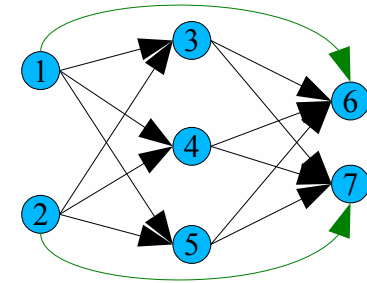
Topologie-Typen



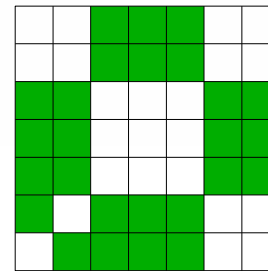
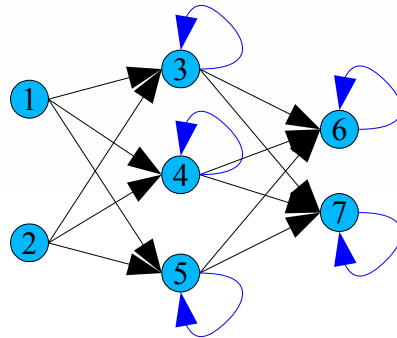
feedforward



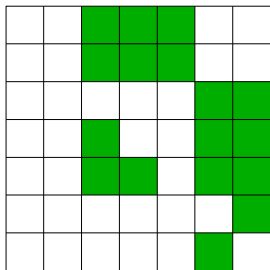
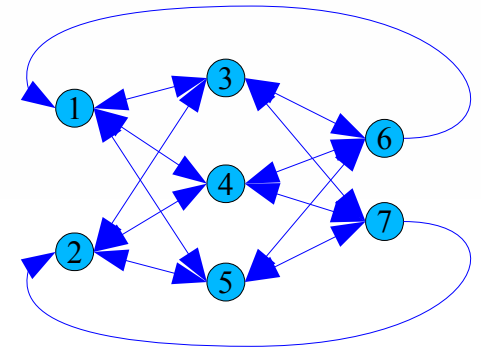
feedforward mit shortcut



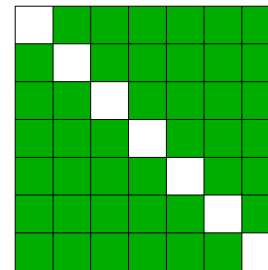
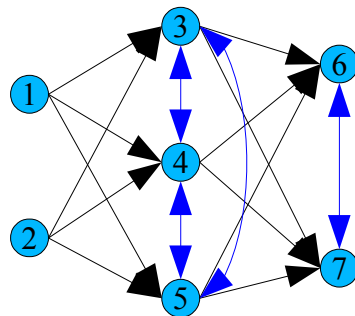
direkte Rückkoppelung



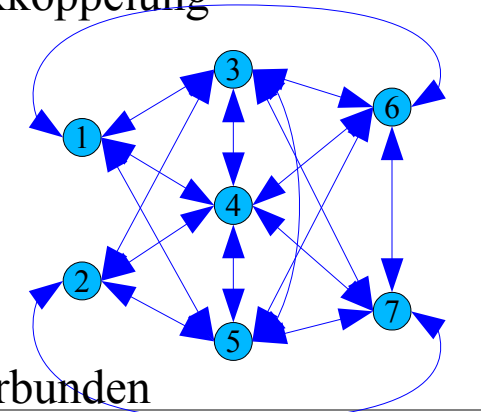
indirekte Rückkoppelung



laterale Rückkoppelung



vollständig verbunden





Topologische Karten

- In der Natur sind alle Netzwerktypen realisiert, häufig werden die Typen von Schicht zu Schicht gemischt.
- Besonders interessant für das selbständige Lernen sind die vollständig verbundenen Netzwerke, die Rückkopplungen zwischen allen Schichten und Neuronen ermöglichen. Einige können hemmend und andere verstärkend wirken.
- Solche Netze eignen sich zur Ausbildung einer topologischen Karte. Es bilden sich Regionen benachbarter Neuronen heraus, die auf ähnliche Reize gemeinsam ansprechen: Sprach-, Seh- Hörzentrum und Tastsinn um nur einige der Wichtigsten zu nennen.



Mustererkennung

- Topologische Karten sind „Mustererkenner“: Auf erlernte ähnliche Muster sprechen Neuronen einer Region mit ähnlichen Ausgaben an.
- Ein einfaches Verfahren zur Musterklassifizierung ist die adaptive Vektorquantisierung (LVQ), wie sie z.B. bei der Sprachcodierung, der JPEG Kompression oder Ähnlichem verwendet wird.
- Die mathematische Formulierung erfolgt meistens in einem anderen Kontext, als die bisherigen Formeln. LVQ lässt sich jedoch leicht mit dem mathematischen Formalismus der Neuronalen Netzen beschreiben und ist dann eine Vorstufe der Kohonen Netze.



Vektor Quantisierung

- Elemente v eines Vektorraums V sollen möglichst gut durch Chiffre c eines (endlichen) Codebuchs $C \subset V$ approximiert werden.
- Solche Verfahren werden z.B. zur Datenkompression bei der Bild- oder Tonübertragung eingesetzt:
 - Anstatt der Daten wird lediglich der Index im Codebuch übertragen, z.B. GIF 256 Farben Palette.
- Die Herausforderung ist daher für „die Daten“ ein geeignetes Codebuch C zu finden. Beispiele sind Codecs für MPEG, Wavelets etc.
 - Das Codebuch kann über die Zeit variieren...



Prinzip von VQ

- Das Eingangssignal x wird mit den Vektoren w_1, \dots, w_n des Codebooks C verglichen und durch den Besten w_c approximiert: $\vec{x} \in V \rightarrow c = f(\vec{x}) \in \{1, \dots, n\}$

$$\|\vec{x} - \vec{w}_c\| = \min_{\vec{w}_j \in C} \|\vec{x} - \vec{w}_j\|$$

- Sind x und die Codebook-Vektoren normiert, so läuft dies für einen Innenproduktraum V auf die Minimierung des Skalarprodukts in der L_2 -Norm hinaus:

$$\|\vec{x} - \vec{w}_j\|_2^2 = \vec{x}^T \cdot \vec{x} - 2 \vec{w}_j^T \cdot \vec{x} + \vec{w}_j^T \cdot \vec{w}_j = 2 - 2 \vec{w}_j^T \cdot \vec{x}$$

- Statt der L_2 ist eine andere Metrik $d(x, y)$ möglich.



Das LBG Verfahren

- Linde, Buzo & Gray beschrieben 1980 ein iteratives Verfahren zur Optimierung des Codebooks:

1. Starte mit dem initialen Codebook C^0 und bilde für alle x_k (eines Trainingsatzes) die Mengen

$$B_c = \left\{ \vec{x}_k \mid d(\vec{x}_k, \vec{w}_c) < d(\vec{x}_k, \vec{w}_j) \quad j \neq c \right\}$$

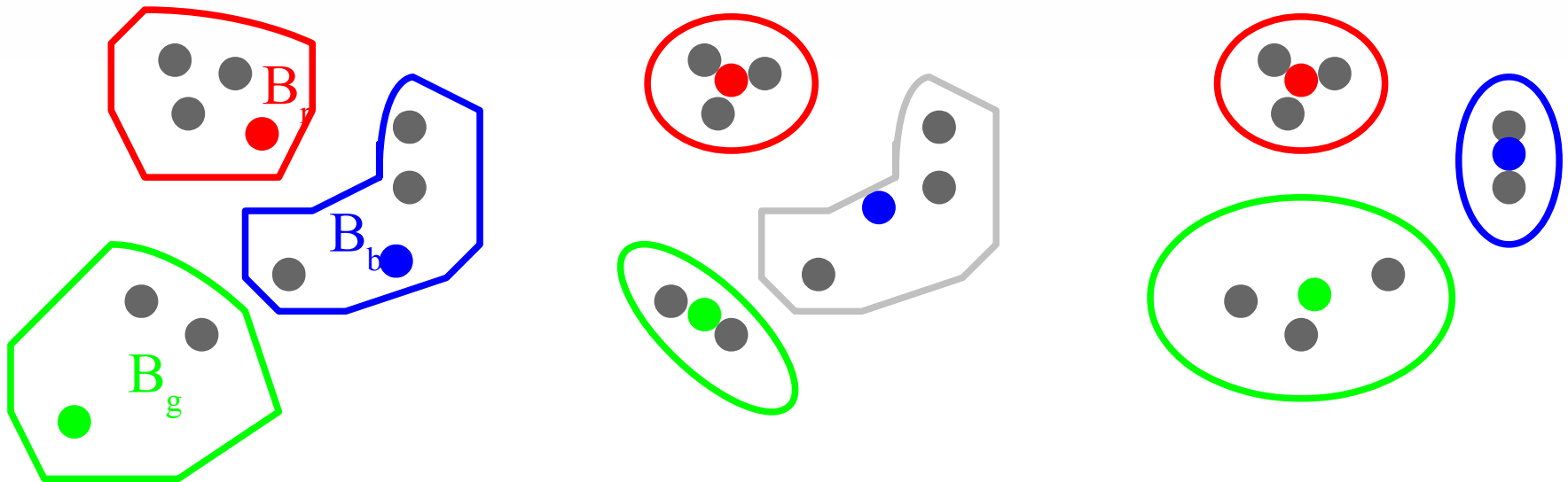
2. Berechne den neuen Schwerpunkt w_c der Menge B_c

$$\vec{w}_c = \frac{1}{|B_c|} \sum_{\vec{x}_k \in B_c} \vec{x}_k$$

3. Bilde das neue Codebook C^l . Fahre fort bei 1. bis C^m die Vektoren x_k ausreichend genau klassifiziert.

Das Prinzip von LVQ

- Bedingt durch die das neue Codebook werden einige x_k , je nach der gewählten Metrik, anders klassifiziert. Nach einigen Iterationen tritt keine neue Klassifizierung mehr ein und das Codebook stabilisiert sich.





NN als LVQ

- Ein Neuronales Netz, dem die Gewichtsvektoren nach dem LVG Verfahren antrainiert wurden funktioniert als Vektorquantisierer. Das Ausgabesignal c kommt von dem Neuron, das am stärksten angeregt wurde.
- Ein vereinfachtes Verfahren kann die Gewichte auch mit jedem Trainingsvektor x_k direkt ohne Mittelwertbildung berechnen: $\Delta \vec{w}_c = \mu (1 - \vec{w}_c^T \vec{x}_k)$
- Bemerkung: Es gibt beim LVQ Netz keinen Vergleich zwischen einem gewünschten Ausgabe- und dem vom Netz berechneten Vektor. Das Lernen geschieht ohne Fehlerkorrektur und kann allenfalls als verstärkendes Lernen bezeichnet werden: richtig klassifiziert?



Kohonennetze

- Kohonen schlug 1982 self-organizing maps (SOM) als erweitertes Konzept des selbständigen Lernen vor.
- Bei den selbstorganisierenden Karten handelt es sich um ein einschichtiges Neuronales Netz, bei dem alle Neuronen miteinander verbunden sind.
- Die Nachbarschaftbeziehung $j \leftrightarrow k$ wird durch $h(j,k)$ abgebildet, meist eine monoton fallende Funktion, die für entfernte Neuronen rasch abklingt.
- Das Axonsignal y_j beinhaltet eine Netzzrückkoppelung:

$$y_j = \sigma \left(\sum_m w_{jm} x_m + \sum_{k \neq j} h(j, k) y_k - \theta_j \right)$$



Neuronengitter

- Die Neuronen werden entweder als lineare Kette oder als zweidimensionales Gitter angeordnet.
 - Frequenzwahrnehmung als lineare Abbildung
 - Optische Reize als zweidimensionale Projektion
- Als Nachbarschaftsrelation existieren verschiedene Funktionen, z.B.:

$$h(j, k) = \begin{cases} 1 & \|\vec{n}_k - \vec{n}_j\|_1 = 1 \\ 0 & \text{sonst} \end{cases}$$

$$h(j, k) = \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left(-\frac{\|\vec{n}_k - \vec{n}_j\|^2}{2\sigma_t^2}\right)$$



Lernende SOM

- In der Lernphase werden der SOM Eingabevektoren x präsentiert und analog zur LVQ das Neuron c mit der größten Aktivierung y_c als auch in geringerem Maße die Neuronen in der Nachbarschaft trainiert:

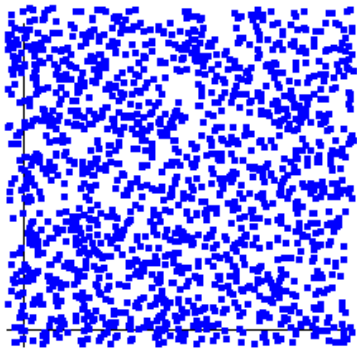
$$\Delta \vec{w}_k = \mu_t h_t(c, k) (\vec{x} - \vec{w}_k)$$

- Häufig wird sowohl die Lernrate μ_t als auch die Nachbarschaftsrelation σ_t mit der Zeit t – d. h. der Anzahl an Iterationen –, immer kleiner gemacht, so dass die Karte nach einer gewissen Zeit einfriert (=>simulated annealing).

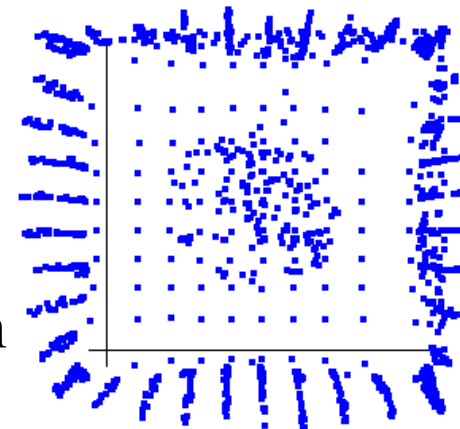
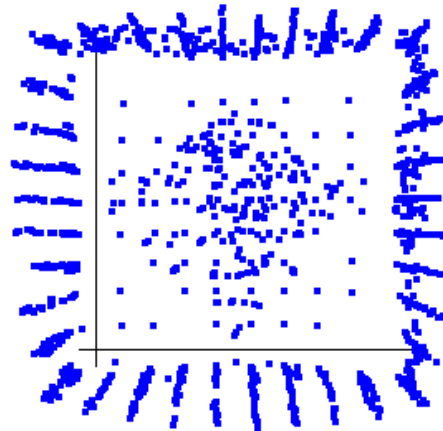


SOM Iteration

- Das Beispiel zeigt die Entwicklung der Gewichte w für einen Trainingssatz, mit zufällig entlang eines Gitters angeordneten Vektoren x .



Start mit Zufalls-
gewichten w .



Nach einigen Iterationen
nehmen die Gewichte w
die Gitterpositionen ein...



Topologische Karte

- Werden nächste Nachbarn graphisch verbunden, so erklärt sich der Name Karte von selber, insbesondere sind Defekte der Karte leicht zu erkennen....

