

Soft-Computing I

Zusammenfassung

Aless Lasaruk

5. November 2005

Zusammenfassung

Dieses Dokument enthält Fragen und Antworten zum Fach Soft-Computing I. Die Fragen sind durch systematisches Durchgehen des gleichnamigen Skriptums entstanden, decken aber den Stoff nicht vollständig. Man beachte auch, dass die Antworten nicht exakt formuliert sind. Das Ziel der Formulierung ist das Minimieren der schriftlichen Antwort und Maximieren der mündlichen.

Inhaltsverzeichnis

1	Einführung	2
2	Grundlagen	2
2.1	Biologische Grundlagen	2
2.2	Fuzzy-Systeme	3
3	Einlagige Perzeptren	3
3.1	Einfaches Neuronen und Netzmodell	3
3.2	Struktur von einlagigen Perzeptren (SLP)	4
3.3	Training von SLP	4
4	Mehrlagige Perzeptren	5
4.1	Erläuterungen zur Strukturdefinition	6
4.2	Training von MLP	6
4.3	Erläuterungen des Algorithmus	8
5	Anwendungen von MLP	8
5.1	Parameter-/Strukturoptimierung	9
5.2	Technische Anwendungen	10
6	First-Order-Lernalgorithmen	10
6.1	Einfache Modifikationen	10
6.2	Weitere Verfahren	12
7	Wettbewerbslernen	12
7.1	Einfaches Clusteringverfahren	12
7.2	Analyse des Verfahrens	13
8	Selbstorganisierende Karten	13
8.1	Struktur von SOM's	14
8.2	Training von SOM	14
9	Anwendungen von SOM	15
9.1	Klassifikation mit SOM	16
9.2	Bewertungskriterien	17

10 Fuzzy-Systeme	17
10.1 Einfache Operationen auf Fuzzy-Mengen	17
11 Anwendungen von Fuzzy-Systemen	19

1 Einführung

- **Welche wichtigen Aspekte sind für das Entstehen der Disziplin Soft-Computing ausschlaggebend?**
Ungenauigkeiten, Unwägbarkeiten und partiell wahre Informationen.
- **Welche typischen Anwendungsmöglichkeiten für Soft-Computing kennen Sie?**
Regelung bzw. *Steuerung*, Entscheidungsfindung (Klassifikation), Prozessüberwachung (Qualitätskontrolle) und Optimierung. Ein gutes vereinigendes Beispielgebiet ist RoboCUP.
- **Aus welchen Gebieten besteht die Disziplin Soft-Computing?**
Neuronale Netze, Fuzzy-Logik, Evolutionäre Algorithmen, Probabilistic Reasoning, Chaostheorie und Lerntheorie¹.
- **Welche Rolle spielen die biologischen Vorbilder?**
Heute nur geringe. Statt diese genau zu modellieren wird heute nach erfolgreicherem Wegen gesucht.
- **Wie wird Soft-Computing eingesetzt?**
Oft als Kombination aus unterschiedlichen Soft-Computing Techniken (Fusion, *Ensembles*) oder aber auch von Soft- und Hard-Computing (Differentialgleichungen).
- **Wann kommen die Soft-Computing Methoden zum Einsatz?**
Wenn das Problem für eine mathematische Modellierung zu komplex ist. Z.B. bei Echtzeitanwendungen (Zeitfaktor bestimmt die Komplexität).

2 Grundlagen

2.1 Biologische Grundlagen

- **Was ist ein neuronales Netz?**
Ein informationsverarbeitendes System aus vielen einfachen Funktionseinheiten (*Neuronen*), die sich Informationen über gerichtete Verbindungen zusenden.
- **Welche Eigenschaften zeichnen ein neuronales Netz aus?**
Einfache Neuronen und Verbindungen, Lernfähigkeit und ein massiv paralleler Aufbau.
- **Was ist der entscheidende Unterschied in der Funktionsweise des Gehirns und eines Rechners?**
In der massiv parallelen Verarbeitung. Alles hat aber Stärken und Schwächen.
- **Woraus besteht meistens ein biologisches Neuron?**
Zellkörper, Zellkern, Nervenfasern (*Axone* und *Dendriten*) und *Synapsen*.
- **Was sind Unterschiede zwischen einem Axon und Dendrit?**
Dendriten sind verzweigt und nehmen Eingangssignale auf. Axon ist meistens einmalig pro Zelle. Über Axone wird die Information weitergeleitet.
- **Welche Funktion erfüllen die Synapsen?**
Synapsen bieten (meistens) eine chemische Verbindung (*Neurotransmitter*) zwischen Zellen (excitatorisch(anregend)/inhibitorisch(hemmend)). Diese dient dem Lernen, weil durch Über/unter-Sättigung sich die Leitfähigkeit (*synaptische Plastizität*) verändert.

¹Die letzten vier Gebiete werden nicht behandelt. In evolutionären Algorithmen werde ich nicht geprüft. Daher entfallen die Bezüge in diesem Fragebogen zu diesen Themen

- **Wie funktioniert ein biologisches Neuron?**
Potenziale werden auf Dendriten aufsummiert. Übersteigt die Summe einen Schwellwert, so feuert das Neuron einen Impuls ab, welcher über den Axon über Synapsen an nachfolgenden weitergegeben wird.
- **Werden die Impulse bei Neuronen in beide Richtungen geleitet?**
Nein. Nur in eine, wie oben.
- **Wie ist die Information im Gehirn repräsentiert?**
Durch die synaptischen Gewichte, Verbindungsstruktur und durch die Ströme (Stärke und Frequenz). Ein einzelner Impuls trägt aber keine Information.
- **Wie heißt der jüngste und aktivste Teil der Hirnrinde?**
Neokortex. Der unterteilt sich in unterschiedliche Kortex-Arten (visuell, motorisch, usw.)
- **Welche Struktur weist der grobe Aufbau des Hirns?**
Eine Schichten/Regionen-Struktur.
- **Welche Eigenschaft erfüllen benachbarte Hirnregionen?**
Sie reagieren auf ähnliche Reize (*Topologieerhaltung*).
- **Sind die Zuordnungen der Felder zu den Funktionen statisch?**
Nein. Im Falle eines Ausfalls kann sich die Funktion verändern.
- **Wovon hängt die Größe eines Feldes ab?**
Von der Benutzungshäufigkeit.
- **Welche biologische Funktionsweisen findet man in den künstlichen neuronalen Netzen?**
Einfache Verarbeitungseinheiten, *Parallelität*, Integration der Aktivierungen, gerichtete Verbindungen, Information wird in Verbindungen gespeichert, erregende/hemmende Verbindungen, Modifikation der Verbindungen in Abhängigkeit der Häufigkeit der Benutzung, Schichtenmodelle, zeitliches Verhalten (9 Punkte).

2.2 Fuzzy-Systeme

- **Welcher Aspekt liegt den Fuzzy-Systemen zugrunde?**
Fähigkeit von Menschen mit unpräzisen Daten umzugehen.
- **Was ist der Unterschied zwischen klassischer Logik und *Fuzzy-Logik*?**
Klassische operiert mit zwei Werten. Fuzzy mit (oft sogar überabzählbar) vielen.
- **Wo findet Fuzzy-Logik Anwendung?**
Steuerung von Robotern z.B.

3 Einlagige Perzeptren

3.1 Einfaches Neuronen und Netzmodell

- **Was ist ein *Perzeptron*?**
Das ist eine Bezeichnung für ein Neuron mit additiv gewichteter Eingabe und Schwellwertaktivierungsfunktion.
- **Wie ist ein einfaches (boolesches) Neuron aufgebaut?**
Eingänge werden gewichtet aufaddiert (*Netzeingabefunktion*) und über eine Schwellfunktion (*Aktivierungsfunktion*) weitergeleitet.
- **Können Sie mit Hilfe eines Neurons die boolesche \vee -Funktion realisieren?**
Man setzt beide Gewichte auf 1 und den Schwellwert auf 0.5. Dann liefert die Schwellwertfunktion genau dann 1, wenn mindestens einer der Eingänge 1 ist.
- **Ist eine solche Gewichtungskonfiguration eindeutig?**
Nein. Im Allgemeinen nicht.

- **Kennen Sie ein anspruchsvolleres Beispiel für die Verwendung von einfachen Neuronen?**

Bildmasken Operatoren in der Bildverarbeitung. Zum Erkennen von senkrechten Streifen von links nach rechts kann man links negative, rechts positive und in der Mitte einer 3×3 Maske 0 in Spalten anordnen.

- **Was versteht man unter *Training* eines Netzes?**

Einstellen der Gewichte zum Erfüllen einer Aufgabe.

3.2 Struktur von einlagigen Perzeptren (SLP)

- **Wie viele Schichten hat ein *einlagiges Perzeptrennetz*?**

Zwei. Eine Eingabe und eine Ausgabeschicht.

- **Wie viele Neuronen befinden sich in der Ausgabeschicht?**

Nur einer.

- **Wo befinden sich die Gewichte?**

Auf Verbindungen zwischen der Eingabe und der Ausgabeschicht.

- **Ist ein SLP-Netz vollständig verbunden?**

Ja. Wenn man Kanten löschen möchte, so setzt man das entsprechende Gewicht auf 0.

- **Welche Aktivierungs- und Propagierungsfunktionen werden den Eingabeneuronen zugeordnet?**

Identische Abbildungen.

- **Welche Aktivierungsfunktion wird den Ausgabeneuronen zugeordnet?**

Eine Schwellwertfunktion. Ihrer Wertebereich ist $\{0, 1\}$.

- **Welche Propagierungsfunktion wird den Ausgabeneuronen zugeordnet?**

Eine Linearkombination der Ausgaben der Eingabeneurone.

- **Was sind die externen Ausgaben eines SLP?**

Die Ausgaben der Ausgabeschicht.

- **Was wird mit der Anzahl der Lagen gezählt?**

Die Anzahl der Gewichtsschichten.

3.3 Training von SLP

- **Was ist eine Lernaufgabe für SLP?**

Eine Menge von Eingabe/Ausgabemusterpaaren.

- **Was ist eine vollständige Lernaufgabe?**

Eine vollständige Lernaufgabe für SLP charakterisiert alle Eingabe/Ausgabewerte.

- **Was ist eine Epoche?**

Eine Epoche ist das einmaliges Präsentieren der kompletten Lernaufgabe.

- **Wie funktioniert der Lernalgorithmus bei SLP?**

Es wird eine Eingabe angelegt, tatsächliche Ausgabe berechnet und je nach Abweichung/Fehler (Wert) werden die Gewichte modifiziert.

- **Wie werden die Gewichte bei SLP initialisiert?**

Mit dem Wert 0.

- **Wie werden die Gewichte in einem Lernschritt verändert?**

Wenn erwünschte Ausgabe und tatsächliche Ausgabe gleich sind, gar nicht. Sonst wird das Gewicht um das $\pm\nu$ -Fache der Eingabe verändert, je nach Fehlertyp (Netz 0, erwünscht 1, dann +).

- **Wie werden die Schwellwerte in einem Lernschritt verändert?**

Die Schwellwerte werden im Falle eines Fehlers um $\pm\nu$ verändert in umgekehrter Zuordnung (Netz 0, erwünscht 1, dann -).

- **Wann gilt eine Lernaufgabe als erfüllt?**
Wenn für alle Muster keine Gewichtsänderungen durchgeführt wurden. Sonst nicht.
- **Wie oft werden alle Muster präsentiert?**
Hängt von der Aufgabenstellung und Funktion ab.
- **Wie wählt man ν ?**
Zwischen 0 und 1. Das Verfahren konvergiert in Abhängigkeit von ν . Es gibt keine optimalen Lernrate. Die ist von der Aufgabenstellung abhängig.
- **Kann man beweisen, dass das Trainingsverfahren für SLP eine Lösung findet, wenn eine existiert?**
Ja.
- **Wie kann man die Funktionsweise eines Perzeprons anschaulich darstellen?**
Ein Perzeptron separiert linear im Eingaberaum. Das kann man einsehen, indem man die Netzfunktion ausschreibt.
- **Kennen sie eine Funktion, die man nicht mit einem Perzeptron realisieren kann?**
Ja. Die XOR-Funktion. Die positiven und negativen Ausgaben sind nicht linear separierbar.
- **Wie kann man mit Hilfe mehrerer Perzeptren die XOR-Funktion realisieren?**
Durch Baumartiges hintereinanderschalten.
- **Gibt es dafür dann einen Lernalgorithmus?**
Nein. Die Netzfunktion ist zu kompliziert.
- **Welches Problem ergibt sich aus der Schwellwertfunktion?**
Man kann nicht bei geringfügiger Veränderung der Eingabe auf die Veränderungsrichtung des Ausgabe kommen, die zur Verbesserung der Ausgabe führt.
- **Wie kann man das Problem lösen?**
Man betrachtet stetig differenzierbare Funktionen.
- **Wie kann man den Schwellwert auch noch modellieren?**
In dem man den auf 0 ersetzt und stattdessen den negativen Wert $-\theta$ auf die Eingabe addiert. Dadurch kann man den Schwellwert als Gewicht eines eingehenden Neurons betrachten, dessen Aktivierung immer 1 ist.

4 Mehrlagige Perzeptren

- **Welches Verhalten weisen MLP's auf und warum?**
Statisches, weil keine Rückkopplungen vorhanden sind.
- **Wo verwendet man MLP's?**
Modellierung nichtlinearer kontinuierlicher Funktionen. Nichtlineare Klassifikation von Eingabedaten.
- **Welche Schichten hat ein MLP?**
Eingabeschicht, verdeckte Schicht und Ausgabeschicht.
- **Ist ein MLP vollständig verbunden?**
Nein, normal nicht. Nur Neuronen der Vorgängerschicht sind mit denen der Nachfolgerschicht verbunden. Es gibt aber auch Shortcuts. MLP ist aber schichtweise vollständig verbunden.
- **Wie sehen die Propagierungsfunktionen der Neuronen aus der verdeckten und der Ausgabeschicht?**
Linearkombination der Aktivierungen der Neuronen der Vorgängerschicht und Gewichte. Oft mit einem Bias.
- **Welche Aktivierungsfunktionen besitzen die verdeckte und Ausgabeschicht?**
Nichtlineare sigmoide Funktionen (ähnlich wie die Treppenfunktion vom Aussehen).

- **Wie sind die Eingabe und Ausgabe des Netzes beschaffen?**
Eingabe, wie bei SLP, Ausgabe sind die Werte der Aktivierungsfunktionen der Ausgabeschicht.
- **Welche Neuronen eines MLP kommunizieren mit der Umwelt des Netzes?**
Nur die Eingabe und die Ausgabeschicht.

4.1 Erläuterungen zur Strukturdefinition

- **Was dient als Informationsspeicher eines MLP?**
Die Gewichte der Verbindungen, wie im biologischen Vorbild.
- **Was bedeutet der Begriff feed-forward bei MLP?**
Bei MLP gibt es keine rückwärtswirkende Verbindungen. Der Informationsfluss verläuft von Eingabe zur Ausgabe.
- **Welche Aktivierungsfunktionen werden meist gewählt?**
Sigmoide, kontinuierliche, beschränkte, monoton steigende und stetig differenzierbare Funktionen.
- **Ist es üblich die Aktivierungsfunktion bei allen Neuronen gleich zu wählen?**
Ja. Ist aber prinzipiell nicht notwendig.
- **Welche unterschiedlichen Aktivierungsfunktionen kennen Sie?**
Logistische ($\frac{1}{1+e^{-x}}$), Tangens hyperbolicus ($\tanh(x)$), schnelle Aktivierungsfunktion ($\frac{x}{1+|x|}$).
- **Welche Eigenschaften besitzen obige Aktivierungsfunktionen?**
Ihre Ableitung kann in Termini der Funktion selbst dargestellt werden.
- **Wie verändert man die Wertebereiche der Eingabe und Ausgabe?**
Häufig werden diese skaliert auf ein Intervall $[c, d]$, so dass der größte Wert der Trainingsdaten auf d und der kleinste auf c abgebildet wird.
- **Wie notiert man bei MLP die Strukturdaten?**
Man schreibt die Anzahlen der Neuronen in den Schichten hintereinander getrennt durch einen Pfeil.

4.2 Training von MLP

- **Wird bei MLP das Training von der Anwendung getrennt?**
Nicht immer, obwohl es üblich ist. Man kann auch während der Anwendung die Gewichte trainieren.
- **Wann ist eine feste Lernaufgabe für ein MLP erfüllt (im Gegensatz zu SLP)?**
Wenn für jedes Eingabe/Ausgabe-Musterpaar die Antwort des Netzes auf die Ausgabe sehr gering von der erwünschten abweicht. Z.B. Schwellwert.
- **Wann ist eine freie Lernaufgabe erfüllt?**
Eine freie Lernaufgabe ist erfüllt, wenn das Netz die Gewichte so eingestellt hat, dass die Eingaben, die im Eingaberaum ähnliche Eingabe auf im Ausgaberaum ähnliche Ausgaben abgebildet werden.
- **Was muss notwendigerweise vorhanden sein, damit eine Bewertung eines Netzes zustande kommen kann?**
Ein geeignetes Fehlermaß.
- **Kennen Sie ein sinnvolles Fehlermaß für MLP?**
Ja. Z.B. $E(k) = \frac{1}{2} \langle t(k) - y(k) | t(k) - y(k) \rangle$, mit Wunschknoten $t(k)$ und Ausgabemuster $y(k)$. Der Gesamtfehler (Least-Square) ist dann $\sum_{i=1}^n E(k)$ (n ist die Anzahl der Schichten).
- **Wie ist der mittlere quadratische Fehler definiert?**
Man Teilt den Least-Square-Fehler durch die Anzahl der Muster und die Anzahl der Ausgabe-Neuronen. Ist aber ordnungstechnisch im Wesentlichen das selbe.

- **Welche Eigenschaft erfüllt das Least-Square Fehlermaß?**
Es bestraft wenige stärkere Abweichungen mehr als viele kleinere.
- **Was ist ein überwachter Lernalgorithmus?**
Ein Verfahren zur Erfüllung einer festen Lernaufgabe durch Modifikation der Netzgewichte.
- **Welche Lernmethoden gibt es bei überwachten Lernverfahren?**
Epochenweises Lernen und musterweises Lernen.
- **Was ist eine Fehlerfläche?**
Ein Fehlermaß induziert eine Funktion in Abhängigkeit von den Gewichten des Netzes auf einen Fehler. Der Graph ist die Fehlerfläche.
- **Warum ist beim Trainingsverfahren von MLP nur eine Umgebung der Fehlerfläche bekannt?**
Man bestimmt die Form der Fläche durch Ableitungen.
- **Welches Standardverfahren wird benutzt für MLP?**
Gradientenabstieg in der Fehlerfläche.
- **Konvergiert ein solches Lernverfahren immer?**
Nein. Aber nicht einmal bei Konvergenz kann man davon ausgehen, dass der Grenzwert ein globales Minimum ist.
- **Wie werden die Gewichte bei MLP am Start gesetzt?**
Auf zufällige kleine Werte (Startpunkt).
- **Wie wird beim Anlegen eines Musters verfahren?**
Man propagiert die Eingabe bis zur Ausgabeschicht und bestimmt die Abweichung (Fehler). Falls der Fehler groß genug ist berechne für alle Gewichte eine Änderung.
- **Wann bricht man das Training ab?**
Wenn der Epochenfehler kleiner als eine Schranke ist oder wenn eine feste Anzahl von Schritten durchgeführt wurde.
- **Welche grundsätzliche Lernziele kann man verfolgen?**
Approximation (gleiche Eingaben-kleiner Fehler) oder Generalisierung (andere Eingaben-kleiner Fehler).
- **Was ist der Unterschied zwischen Trainingsmustern und Testmustern?**
Beide Mengen sind üblicherweise disjunkt. Testmuster werden nicht beim Training präsentiert.
- **Kann man jede Funktion beliebig genau approximieren?**
Wenn man ausreichend viele Neuronen verwendet, dann schon.
- **Kann man Generalisierung genau so leicht erreichen, wie Approximation?**
Nein. Analog zur Polynominterpolation können zwischen den Stützstellen (Lernmuster) beliebig große Abweichungen auftreten oder die Anzahl der Neuronen kann zu schwach sein um die Funktion zu beschreiben.
- **Welchen Fehler begeht man beim Aufstellen von Generalisierungsaufgaben oft?**
Man wählt die Auswertungspunkte nahe an den Stützstellen. Z.B. in einem Intervall.
- **Genügt es nach einer Generalisierungsaufgabe einfach die Approximationsleistung des Netzes zu testen?**
Nein. Speziell für Werte weit von Stützstellen kann die Abweichung groß sein.
- **Was versteht man unter dem Begriff Backpropagation?**
Gradientenabstieg in der Fehlerfläche bei MLP.
- **Wie kann man die partielle Ableitung des Fehlers nach einem der Gewichte interpretieren?**
Richtungssteigung der Fehlerfläche in Richtung der Gewichtsachse. Sensitivität des Fehlers gegenüber einer Gewichtsänderung.

- **Wie kann man die Sensitivität noch darstellen?**
Mit der Kettenregel als Produkt der Ableitung des Fehlers nach der Netzeingabe für ein Neuron und der Ableitung des Netzeingabe nach dem gegebenen Gewicht. Den zweiten Faktor kann man direkt differenzieren und erhält die Aktivierung für das Muster der Vorgängerschicht.
- **Wie erhält man nun eine Formel für den ersten Term oben?**
Analog durch Kettenregel nur mit Netzeingabe der nächsten Schicht. Man erhält eine Abhängigkeit von dem Wert der Ableitung der Aktivierungsfunktion und der gewichteten Summe der Signalfunktionen der Oberschicht.
- **Wie kann man nun den Namen Backpropagation erklären?**
So, wie für die Berechnung der Ausgabe die Eingaben nach vorne propagiert werden, so werden für die Fehlerbestimmung die Differenzen nach hinten propagiert.
- **Wie bestimmt man also die Änderungen der Gewichtsfunktionen?**
Als Produkt einer Lernrate ν , der Signalfunktion δ für die gegebene Schicht und der Aktivierung der Vorgängerschicht für das gegebene Muster. Die Signalfunktion ist die negative Ableitung der Fehlerfunktion nach der Eingabe des gegebenen Neurons der gegebenen Schicht.

4.3 Erläuterungen des Algorithmus

- **Welche Nachteile hat Backpropagation?**
Die Wahl einer geeigneten Lernrate ist schwierig (unterschiedliche Minima), Wahl der Startpunktes ist schwierig (kein globales Minimum), Nichtintuitive Konvergenz (Oszillation, Abflachen). Weit von der Ausgabe liegende Gewichte werden deutlich langsamer trainiert (4 Punkte).
- **Ist das Systemverhalten eines MLP statisch oder dynamisch? Was bedeutet das?**
Statisch. Die Ausgabe hängt nur von der Eingabe zum gegebenen Zeitpunkt ab.
- **Ist MLP ein lineares System?**
Nein. Aufgrund der Aktivierungsfunktion.

5 Anwendungen von MLP

- **Was ist ein wichtiges Kriterium der Bewertung der Qualität von MLP?**
Eine repräsentative stochastische Analyse.
- **Nennen Sie mir ein paar praktische technische Anwendungen von MLP?**
Energieverbrauch in einem Gebäude, Klassifikation von Glassplittern.
- **Welche Störeinflüsse existieren bei der praktischen Anwendung von MLP?**
Ungenauere Daten (Störungen, Übertragungsfehler, digitalisierungsfehler), ungenaue Wunschwerte.
- **Welche Parameter können bei einem MLP angepasst werden?**
Skalierung der ein und Ausgabegrößen, Netzarchitektur, Wahl der Lernparameter (Lernrate, Epochenanzahl), Trainingsmuster, Bewertungskriterien.
- **Wie skaliert man quantitative Größen?**
Linear in einem Intervall $] -c, c[$, wobei die Steigung der Aktivierungsfunktion bei c deutlich größer 0 sein sollte. Auch kann man die Streuung als Grundlage wählen. Man kann auch Binärcode oder 1 aus n -Kodierung verwenden (3 Punkte).
- **Wann spricht man von Musterklassifikation?**
Wenn die Netzausgabe qualitativ ist. Sonst von Funktionenapproximation.
- **Wie kann man bei Klassifikationsaufgaben die Ausgabewerte des Netzes etwas zwischen 0 und 1 interpretieren?**
Als Wahrscheinlichkeiten.

- **Wie kann man bei Klassifikationsaufgaben die Zugehörigkeit zu einer Klasse ermitteln?**
Schwellwertverfahren. Möglicherweise sind “unsichere” Bereiche vorhanden.
- **Wie kann man die unsicheren Ergebnisse vermeiden?**
Z.B. Winner-takes-all mit einem Netzensemble.
- **Aus welchen Phasen besteht die Anwendung von neuronalen Netzen für technische Zwecke?**
Experimente am realen System (Daten), Bestimmung und Selektion von Merkmalen (Hard-Computing, Merkmale), Training und Anwendung des Netzes.
- **Wie würden sie Merkmale im Fall von Merkmalsexlektion bei Glassplittern nützen?**
Frequenzbereich aus Schallbereich und die Frequenzen als Eingaben. Aber auch Null-durchgänge in einem Zeitintervall oder die Abnahme der Intensität.
- **Welche Standardmethode zur Merkmalsextraktion würden Sie nützen?**
Sequentiell forward search (Nachschauen). Aber auch z.B. Koeffizienten von approximierenden Polynomen usw.
- **Was ist ein wesentlicher Nachteil bei Approximationsaussagen über MLP?**
Die Sätze nehmen beliebig viele Neuronen an.
- **Was ist ein Überanpassung des Netzes?**
Das Netz lernt die Muster “auswendig” kann aber nicht generalisieren.
- **Wie verhindert man Überanpassung?**
Durch ausreichende Anzahl von Neuronen, Regularisierung (Strafterm), ausreichend viele Trainingsmuster, Früher Trainingsabbruch, verrauschte Eingabedaten, Reduktion der Dimension des Eingaberaumes, Netzstrukturoptimierung, Ensembles (gleiche aufgabe) oder Mixture of Experts (verschiedene Aufgaben), Verbindungen mit identischen Gewichten (9 Punkte).
- **Kennen sie eine Schlagregel für die Anzahl von Neuronen?**
Etwa die Wurzel aus der Summe aus Eingabe und Ausgabeneuronen. Ist aber nur beschränkt anwendbar.
- **Kennen sie eine Regel für die Anzahl der Testmuster?**
Etwa der Quotient der Gesamtzahl der Freiheitsgrade und Fehler. Oder das doppelte der Anzahl der Eingabeneuronen.
- **Muss man bei höherer Anzahl von Merkmalen mehr Muster bereitstellen?**
Nicht unbedingt. Wenn die Muster bereits repräsentativ sind nicht.

5.1 Parameter-/Strukturoptimierung

- **Wie schwer ist das Problem der Strukturoptimierung?**
Nicht triviales kombinatorisches Optimierungsproblem (Nichtbeschränkter Lösungsraum, diskret, nichtdifferenzierbare Fehlerfunktion, sehr viele lokale Minima, ungenaue Ergebnisse)
- **Welche grundlegende Verfahren gibt es dazu?**
Destruktive (pruning, z.B. löschen von Neuronen mit kleinen Gewichten, penalty Term), konstruktive und hybride (evolutionär).
- **Wie nutzt man MLP mit dem Zweck der Generalisierung?**
Testen an unbekanntem Mustern. Man teilt die Daten in Lerndaten und Testdaten auf. Oft mit Kreuzvalidierung verbunden etwa mit verschiedenen Netzen oder Netzstrukturen.
- **Welche grundlegenden Merkmale untersucht man beim Bewerten der Netze?**
Generalisierungsfähigkeit, Reproduzierbarkeit, Robustheit.

- **Warum kann bei einem Fehlermaß auch das Vorzeichen interessant sein?**
Wenn man z.B. Verschleiß durch ein Fehlermaß ausdrückt, so ist auf zu geringen und zu großen Verschleiß verschieden zu reagieren.
- **Welche Angaben sind bei Beurteilung von Klassifikationsverfahren vernünftig?**
Statt der Anteils der korrekt klassifizieren Muster sollte man den Anteil der nicht korrekt klassifizierten Muster für jedes Merkmal angeben.
- **Wieso reicht nicht die Angabe des Anteils der korrekt klassifizierten Muster?**
Weil bei einer Klasse mit 1 zu 9 Verteilung der Muster (90% für Klasse 1) ist trivial eine 90% Erfolgsquote zu erzielen in dem man stets die Klasse 1 ausgibt.
- **Welche Fehlerarten gibt es für MLP?**
Der durchschnittliche Fehler, maximale Fehler, usw.
- **Wie kann man hohe Reproduzierbarkeit bzw. ein gutes Netzmodell erkennen?**
Aufgrund der unterschiedlichen Initialisierungsgewichten sollten die entstehenden Streuungen in den Antworten gering sein.
- **Reicht es aus mehreren Trainingsdurchgängen anhand eines Testsatzes das beste Netz zu wählen?**
Nein. Nur wenn die Streuungen gering sind.
- **Genügt beim Vergleich mehrerer Strukturen die Betrachtung der Fehlerdifferenzen?**
Nein. Es muss ein Signifikanztest durchgeführt werden. Man muss auf unterschiedlichen Wahrscheinlichkeitsniveaus überprüfen, ob der Fehler kleiner ist und die Streuungen klein genug sind.

5.2 Technische Anwendungen

- **Welche praktischen Anwendungen von MLP kennen Sie?**
NETtalk (Sprechnet), Neurogammon, Klassifikation vom Verschleiß von Werkzeugen, Energierverbrauchbestimmung im Gebäude, Glassplitter-Klassifikation.
- **Welche Merkmale und Ausgabedaten sind bei der Splitterklassifikation z.B. geeignet?**
Anteile der Elemente als Eingabe und Glastypen als Ausgabe.

6 First-Order-Lernalgorithmen

- **Wie kann man auf Backpropagationen gründende Verfahren beschreiben?**
Suchrichtung bestimmen, Linienabstieg durchführen, Fehler prüfen.
- **Welche Richtung wählt man bei der Backpropagation?**
Die des negativen Gradienten.
- **Was versteht man unter first-order Verfahren?**
Verfahren, die zur Suchrichtungsbestimmung Gradienten verwenden und keine tieferen Ableitungen.

6.1 Einfache Modifikationen

- **Wodurch unterscheidet sich Backpropagation mit Momentum von der üblichen?**
Man addiert zur Gewichtsänderung ein skalares Vielfaches der Änderung der letzten Epoche.
- **Welche Wirkung liefert das Momentum?**
Filterung hochfrequenter Änderungen und somit eine gewisse Trägheit. Auch eine bessere Konvergenz.

- **Wie kann man die Momentumpropagation verallgemeinern?**
Indem man zur Lernrate noch $(1 - \alpha)$ mit Momentum α multipliziert. Das nennt sich dann exponentielle Glättung.
- **Kann man andere Verfahren mit dem Momentum kombinieren?**
Ja.
- **Was ist die Regularisierung?**
Man addiert einen Term, der quadratisch hohe Gewichte als Fehler bewertet. Im Lernverfahren drückt sich das durch Subtraktion eines skalaren vielfachen des Gewichtsvektors von der Differenz (Dämpfung).
- **Wieso ist die Regularisierung begründet?**
Einerseits biologisch, da keine hohen Potenziale ausgetauscht werden und andererseits durch kleine Wertebereiche der Aktivierungsfunktionen.
- **Welche Wirkung hat Regularisierung?**
Eine höhere Generalisierungsfähigkeit.
- **Wie kann man die flat-spots eliminieren?**
Addition eines positiven Wertes an die Ableitung oder die Anpassung der Aktivierungsfunktion, sodass diese nicht unter einen bestimmten Wert sinken kann.
- **Welche allgemeinen Möglichkeiten gibt es die Lernrate anzupassen?**
Lokale (jedes Gewicht einzeln) und globale (für alle Gewichte).
- **Was zeichnet den Algorithmus der dynamischen Adaptation aus?**
Man berechnet den Fehler für zwei verschiedene Raten. Dabei verändert man die Rate entweder durch Multiplizieren mit einem Skalar $\nu > 1$ oder durch Division mit dem selben.
- **Welcher Vorteile hat dynamische Adaptation?**
Der zusätzliche Rechenaufwand besteht in einer zusätzlichen Auswertung der Fehlerfunktion durch Vorwärtspropagieren für jeden Updateschritt.
- **Wie funktioniert das Verfahren von Silva und Ameidá?**
Das Verfahren benutzt für jedes Gewicht eine eigene Rate. Ändert sich das Vorzeichen der partiellen Ableitung in zwei aufeinanderfolgenden Schritten, so wird die Lernrate verkleinert. Ist dies nicht der Fall, so wird die Lernrate vergrößert (multiplikativ).
- **Was passiert, wenn die partielle Ableitung das Vorzeichen ändert?**
Ein Minimum ist übersprungen worden.
- **Welches Problem hat der Algorithmus von Silva und Almeida?**
Die Lernraten können exponentiell wachsen oder sinken. Man kann es dadurch lösen, dass die Lernraten additiv verändert werden.
- **Worin bestehen die Forderungen der Resilient-Propagation?**
Die Lernraten für einzelnen Gewichte sollte sich dynamisch an die lokal sichtbare Form der Fehlerfunktion anpassen und nicht nur von der Höhe der partiellen Ableitungen abhängen. Das Verfahren RPROP berücksichtigt lediglich das Vorzeichen der Ableitungen.
- **Wie funktioniert RPROP?**
Es werden Gewichtsveränderungsraten mitgeführt. Wechselt die partielle Ableitung das Vorzeichen nicht, so wird die Rate multiplikativ μ^+ erhöht oder am Maximalwert geclippt. Wechselt die Ableitung das Vorzeichen, dann wird der Wert der Rate multiplikativ erniedrigt μ^- oder am Minimalwert geclippt. Zusätzlich wird die Richtung auf 0 gesetzt. Ist das Produkt der partiellen Ableitungen in beiden Schritten 0, so wird die Änderungsrate vom letzten Schritt übernommen.
- **Welche Probleme bringt es mit sich, wenn man die Faktoren μ^+ und μ^- bei RPROP in den Zusammenhang $\mu^+ = \frac{1}{\mu^-}$ bringt?**
Bei symmetrischen Umgebungen kann es zum Oszillieren kommen. Daher fordert man auch $\mu^+ < \frac{1}{\mu^-}$ damit die Zunahme langsamer ist als die Abnahme.
- **Welche Vorteile hat RPROP?**
Verbesserte Konvergenz, Schnellere Konvergenz (Strukturoptimierung, Online-Lernen), geringe Abhängigkeit der Konvergenzgeschwindigkeit von den Parametern des Lernalgorithmus (kleinere Änderungen führen nicht zu großen Veränderungen des Ergebnisses).

6.2 Weitere Verfahren

- **Auf welcher Vorstellung basiert QuickProp?**
Die Fehlerfunktion kann in der Nähe eines Minimums durch eine Parabel angenähert wird.
- **Wie bekommt man die Näherung?**
Man versucht die Gewichtsänderung durch den Quotienten der Differenzen der partiellen Ableitungen der aktuellen und der Nachfolgerepoche und der Vorgänger und der aktuellen Epoche zu berechnen. Unter der Annahme, dass die partielle Ableitung in der Nachfolgerepoche bei 0 liegt. Ergibt sich die Regel.
- **Welche Fälle sind problematisch für QuickProp?**
Ähnliche oder Gleiche Werte der Ableitung in Vorgänger und aktueller Epoche (großer Schritt oder Abbruch) und der Wert in der Vorgängerepoche ist betragsmäßig größer als der aktuelle und hat das gleiche Vorzeichen (Schritt in die Richtung des Maximums).
- **Wie kann man die Probleme beheben?**
Das erste durch ein Ratenbegrenzung und das zweite durch eine Fallunterscheidung und alternative Vorgehensweise.
- **Wie funktioniert QuickProp?**
Der Gradiententerm wird zunächst berechnet als Linearkombination aus Gewichtsänderung durch Backpropagation und ein skalares Vielfaches des Gewichtes (wobei dies der Gewichtsbeschränkung dienen soll, Strafterm). Wenn der Parabelterm betragsmäßig kleiner ist als ein γ ist, wird die Veränderungsrate mit dem Parabelterm abgeändert, sonst wird γ als Faktor genommen. Die Gewichtsänderung ist dann die Summe aus Gradiententerm und Parabelterm.
- **Wie kann man den Fall des Nullnenners abfangen?**
Indem man einen kleinen Wert zum Nenner addiert.
- **Wann funktioniert QuickProp gut?**
Wenn die Fehlerfunktion lokal durch eine Parabel dargestellt werden kann und wenn die Gewichtsänderung unabhängig von der Änderung anderer Gewichte durchgeführt werden kann.
- **Was ist QRPROP?**
Eine Kombination aus QuickProp und RPROP.
- **Was ist ein wesentlicher Unterschied von Backpropagation und Quick/RPROP?**
Unabhängige Veränderung der Gewichte voneinander.

7 Wettbewerbslernen

- **Welche Eigenschaften sollte ein Clustering-Verfahren erfüllen?**
Ähnlichkeit zwischen Elementen in einem Cluster und Verschiedenartigkeit in Verschiedenen Clustern.
- **Was ist der Unterschied zwischen Klassifikation und Clustering?**
Klassifikation versucht aus Beispielen mit Vorgabe die Klassenzugehörigkeit vorherzusagen und Clustering findet natürliche Cluster aus Beispielen ohne Vorgabe.
- **Was ist eine weitere Aufgabe eines Clustering-Verfahren?**
Einfache Beschreibung für Cluster durch z.B. einen Prototypen.

7.1 Einfaches Clusteringverfahren

- **Wie kann ein einfaches iteratives Verfahren zum Finden eines Prototypen aussehen?**
Initialisierung auf ein zufälliges Clusterelement, Auswählen eines weiteren Elements, Annähern des Prototyps an das Element.

- **Was muss man bei einem solchen Verfahren beachten?**
Maß, Ähnlicher-Werten, Konvergenz, Mehrere Prototypen für mehrere Cluster.
- **Welche Ähnlichkeitsmaße kann man verwenden?**
Skalarprodukt, falls die Merkmalsvektoren normiert sind, Euklidischer Abstand.
- **Was muss man machen, damit linear abhängige Vektoren durch Normierung auf den selben abgebildet werden?**
Man fügt eine 1-Komponente zu.
- **Wie kann man den Prototypen ähnlicher einem anderen Element machen?**
Z.B. durch Halbieren des Winkels beim Skalarprodukt und durch Halbieren des Abstandes beim Euklid.
- **Wie kann man Konvergenz erzwingen?**
Man senkt die Adaptationsrate Schritt für Schritt.
- **Wie kann man mehrere Prototypen finden?**
Initialisierung mehrerer Prototypen und Auswahl des ähnlichsten nach dem Winner-Takes-All Prinzip.
- **Wie funktioniert Wettbewerbslernen?**
Bestimme Ähnlichkeit des nächsten Musters zu allen Prototypen, Wähle den Gewinner-Prototyp und adaptieren den Gewinner.
- **Kann ein solches Verfahren Optimalität garantieren?**
Nein. Zufällige Wahl von Prototypen am Anfang ist bereits ein Grund dafür.

7.2 Analyse des Verfahrens

- **Welche Faktoren spielen beim Ergebnis eine Rolle?**
Absenken der Lernrate ist wichtig.
- **Funktioniert das Verfahren bei gleich initialisierten Prototypen?**
Ja. Im ersten Schritt sind die bereits nicht mehr gleichwertig.
- **Wie kann man das Verfahren mit neuronalen Netzen realisieren?**
Zwei Schichten, wobei eine die Eingaben und die andere die Prototypen beschreibt. Die Gewichte in der Ausgabeschicht eines jeden Neurons beschreiben den Prototypen.
- **Welches Prinzip spielt dabei eine wichtige Rolle?**
Topologieerhaltung. Benachbarte Cluster werden durch benachbarte Prototypen repräsentiert.
- **Wie wird dann ein solches Netz trainiert?**
Indem ein Sieger bestimmt wird und in seiner Umgebung alle Prototypen angepasst werden. Die Lernrate muss ebenfalls abgesenkt werden für Konvergenz.
- **Kennen Sie ein praktisches Beispiel für ein solches Netz?**
WEBSOM, Benachbarte Web-Artikel auf einer zweidimensionalen Karte.

8 Selbstorganisierende Karten

- **Welches Prinzip haben SOM's?**
Ähnliche Eingaben erzeugen Aktivierung benachbarter Neuronen.
- **Was kann man bezüglich der Dimensionen der Eingabe/Ausgabe-Räume aussagen?**
Ein hochdimensionaler Eingaberaum wird auf einen niedrigdimensionalen Raum abgebildet.
- **Welche Anwendungen von SOM's kennen Sie?**
Dimensionsreduktion komplexer Daten, Visualisierung von Ähnlichkeiten, Clustering, Funktionsmodellierung (Robotik), Organisations von Gehirnstrukturen.

8.1 Struktur von SOM's

- **Wie viele Schichten hat eine SOM?**
Zwei. Eingabeschicht und Wettbewerbsschicht.
- **Welche Eigenschaften besitzt ein Neuron in der Wettbewerbsschicht?**
Einen Koordinatenvektor und einen Gewichtsvektor.
- **Wie ist die Netzeingabefunktion für die Wettbewerbsschicht definiert?**
Euklidischer Abstand der Eingabe und des Gewichtsvektors.
- **Wie bestimmt man das Siegerneuron?**
Eines der Neurone mit dem minimalen Aktivierung.
- **Wie hoch ist die Dimension der Karte meistens?**
Meist nicht mehr als dreidimensional.
- **Wie werden Neuronen auf der Karte plziert?**
Normal in Gittern, welche meist quadratisch sind. Oder auch hexagonal. Die Anordnung muss nicht quadratisch sein, ist aber oft. Sehr selten auf Oberfläche einer dreidimensionalen Kugel.
- **Wie kann man noch die Propagierungsfunktion beschreiben?**
Über Skalarprodukt auf normierten Vektoren.

8.2 Training von SOM

- **Was ist eine freie Lernaufgabe?**
Eine freie Aufgabe aus einer Abfolge von Eingabedaten ist erfüllt, falls für ähnliche Eingaben ähnliche Ausgaben erfolgen.
- **Was ist eine Radiusfunktion bei SOM?**
Eine Funktion, die einen positiven Abstand eines Punktes vom Ursprung angibt.
- **Welche Abstände benutzt man da?**
 p -Normen, meistens die euklidische, manhattan oder Maximumsnorm.
- **Was ist eine Distanzfunktion bei SOM?**
Eine Funktion von \mathbb{R}^+ in sich selbst, die monoton fallend ist und für unendlich gegen 0 geht.
- **Welche Distanzfunktionen kennen Sie?**
Lineare, Exponentielle ae^{-tb} , Wurzdistanzfunktion $a\sqrt{\frac{1}{t}^b}$.
- **Was ist eine Nachbarschaftsfunktion?**
Eine Funktion $f(r, d) : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}$ mit $f(r, d) \leq f(0, d)$ und $\lim_{r \rightarrow \infty} f(r, d) = 0$.
- **Wie ist die Anwendung der Nachbarschaftsfunktion?**
Für eine feste Distanz d wird der Abstand r (also metrisch) eingesetzt. Daraus ergibt sich für unterschiedliche Abstandsdefinitionen ein Wert, der die Nachbarschaftsbedingung angibt.
- **Welche Nachbarschaftsfunktionen kennen Sie?**
Z.B. die Gauss-Funktion $f(r, d) = e^{-(r/d)^2}$.
- **Was ist eine Lernratenfunktion?**
Eine Lernratenfunktion ist eine gegen 0 konvergente Funktion $f(t) : \mathbb{R}^+ \rightarrow [0, 1]$.
- **Wie funktioniert das Training bei SOM?**
Initialisierung (kann auch mit gleichem Wert gehen), Lege eine Netzeingabe und bestimme das Siegerneuron, Bestimme für alle Neuronen den Grad der Nachbarschaft mit Hilfe der Radiusfunktion, Distanzfunktion und Nachbarschaftsfunktion, modifiziere die Gewichte um ein skalares Vielfaches des Unterschieds der Eingabe und des aktuellen Gewichtsvektors. Das Vielfache ist dabei der Nachbarschaftsgrad multipliziert mit einer Lernratenfunktionsauswertung.

- **Was ist der Quantisierungsfehler der SOM?**
Der Durchschnitt minimaler Gewichtsunterschiede.
- **Was kann man beobachten, wenn man Neuronen, die nicht in der Nachbarschaft liegen abstoßt?**
Eine Explosion der Karte ist möglich. Also divergentes Training.
- **Wie kann man das Lernen einer SOM modellieren?**
Als einen Markov-Prozess.
- **Welche Eigenschaften besitzt die erhaltene Karte?**
Bereiche mit mehr Mustern besitzen mehrere Ausgangsneuronen.
- **Was kann man beobachten, wenn man die Lernrate zu früh klein macht?**
Es kommt zu topologischen Defekten, weil sich die grobe Struktur eingestellt hat, bevor die Karte sich entfalten konnte.
- **Kennen Sie ein Beispiel einer Anwendung von SOM, in dem die Topologieerhaltung zur Geltung kommt?**
Peano-Kurve. Zufällige Verteilung von Punkten auf Gittern eines Quadrates auf eine eindimensionale Karte.
- **Welche weiteren Anwendungen kennen Sie?**
Adaptive Datenkompression, Vektorquantisierung, Hauptkomponentenanalyse.

9 Anwendungen von SOM

- **Was versteht man unter Kalibrierung eines SOM-Netzes?**
Interpretation einer trainierten Karte anhand bekannter Muster, deren Ähnlichkeitsbeziehungen bekannt sind.
- **Ist eine Skalierung der Eingabegrößen bei SOM erforderlich?**
Nein.
- **Warum gibt es wenig Hinweise auf die Wahl von Parametern in der Literatur?**
Weil SOM's sehr robust sind.
- **Welche Präsentationarten der Muster kennen Sie?**
Zyklisch mit fester Reihenfolge, zyklisch mit zufälliger Reihenfolge, mit zufälliger Reihenfolge.
- **Wie verfährt man, wenn wichtige Datenbereiche ausreichend repräsentiert werden müssen?**
Man präsentiert sie öfter, Man wählt eine hohe Lernrate oder Nachbarschaftsgrad.
- **Wie kann man die Repräsentation eines bestimmten Musters auf der Karte erzwingen?**
Indem man die Kartengewichte mit dem Muster initialisiert.
- **Wie kann man die Qualität von SOM bewerten?**
Z.B. durch Quantisierungsfehler. Man sollte aus vielen Netzen mit unterschiedlichen Parametern auswählen.
- **Wie kann man den Quantisierungsfehler für Generalisierung verwenden?**
Indem man unbekannte Muster präsentiert.
- **Kann man mit dem Quantisierungsfehler die Güte der Topologieerhaltung bewerten?**
Nein.

9.1 Klassifikation mit SOM

- **Welche Bedingung sollte erfüllt sein für Klassifikation mit SOM?**
Die Anzahl der Neuronen auf der Karte soll größer sein als die der Klassen.
- **Was ist eine freie Lernaufgabe mit Klassenzuordnung?**
Eine Partition der Aufgaben in Klassenteilmengen.
- **Was ist das Ziel eines Interpretationsverfahrens für eine Klassifikationsaufgabe mit Klassenzuordnung?**
Eine Zerlegung der Ausgabeneuronen in entsprechende Klassen.
- **Was ist ein Aktivierungsvektor?**
Durchschnittliche Aktivierung eines Neurons/Klasse von Neuronen für die Muster einer Klasse von Testmustern.
- **Wie kann man die Begriffe nächster Nachbar, entferntester Nachbar und durchschnittlicher Nachbar definieren?**
Minimaler/Maximaler/Durchschnittlicher Abstand von Gewichtsvektoren.
- **Wonach unterscheidet man Interpretationsalgorithmen?**
Ziel (Anzahl der Klassen), Zusatzwissen (Startpartition, Klassenzahl), Merkmale (Aktivierungs/Gewichtsvektoren), Aufbauschema der Cluster (Merkmale der Karte oder Schrittweise Verbesserung).
- **Was macht den CA-Algorithmus aus?**
Heuristisches Verfahren, freie Lernaufgabe mit Klassenzugehörigkeit, Aktivierungsvektoren.
- **Wie funktioniert CA-Algorithmus?**
Man ordnet zunächst jeder Klasse ein Neuron mit dem kleinsten Aktivierungsvektor für diese Klasse. Man bricht dabei ab, falls ein Neuron mehreren Klassen zugewiesen werden soll. Die anderen werden auf die restlichen Klassen entsprechend der Nähe der Aktivierungsvektoren zugeteilt. Falls dann für einen der Neurone eine bessere Aktivierung in einer anderen Klasse vorliegt wird nochmal getauscht.
- **Wieso muss man bei CA manchmal abbrechen?**
Ein Neuron darf auf keinen Fall ein Repräsentant mehrerer Klassen sein.
- **Ist das Terminieren der dritten Phase von CA garantiert?**
Nein. Man sollte rechtzeitig abbrechen.
- **Welche anderen Möglichkeiten zur Änderung bestehen?**
Mindestens ein Repräsentant pro Klasse, die letzten zwei Phasen abwechseln, jedes Neuron darf pro Durchgang wechseln.
- **Warum kann man das Verfahren als greedy bezeichnen?**
Weil das Ergebnis von der Reihenfolge der betrachteten Neuronen abhängt.
- **Welche Idee steckt hinter Single- und Complete-Linkage?**
Zuweisen jedes Neurons zu einer eigenen Klasse. Vermischen zweier Klassen solange die gewünschte Anzahl nicht vorhanden ist.
- **Was ist das Problem von hierarchischen Clustering-Verfahren?**
Schnelle grobe Lösung aber schlechtes Auffinden von globalen Optima, weil Entscheidungen am Anfang nicht revidiert werden können.
- **Braucht man bei hierarchischen Clustering-Verfahren eine Klassenzuordnung?**
Nein.
- **Welches Nähe-Kriterium kann bei Single-Linkage verwendet werden?**
Nächster Nachbar.
- **Welche Vorteile und Nachteile hat Single-Linkage?**
Ausreißer können erkannt werden. Entscheidungskriterium sind am nächsten liegende Neuronen. Daher Bildung von Ketten.

- **Welches Nähe-Kriterium kann bei Complete-Linkage verwendet werden?**
Entferntester Nachbar.
- **Welche Vorteile und Nachteile hat Complete-Linkage?**
Ausreißer können nicht mehr erkannt werden. Es gibt aber keine Bildung von Ketten.

9.2 Bewertungskriterien

- **Was muss man beim Bewerten von SOM beachten?**
Angabe von unkorrekt klassifizierten Mustern pro Klasse. Untersuchung von Testklassen zwischen nicht eindeutig klassifizierten Klassen. Mehrere Wiederholungen, Mehrere Netze bzw. Strukturen, Signifikanztest.
- **Welche Anwendungen von SOM kennen Sie?**
Intelligente Suchmaschine, Patente, Spracherkennung, Ultraschallfrequenzen bei einer Maus (Gehirnfunktionen), TSP, Robotersteuerung, Zeichenerkennung, Bildröhrenklassifikation.
- **Wie kann man SOM noch verbessern?**
Man beginnt mit einer Skalierung der Eingabedaten, die diese klein macht. Danach werden die Daten auf ihre eigentliche Größe skaliert. Oder Fairness-Bewusstsein verhindert häufige Sieger-Neuronen vom Weinen.

10 Fuzzy-Systeme

- **Was ist eine linguistische Regel bzw. Term?**
Regeln sind Aussagen und Terme sind vorkommende elementare unscharfe Werte.
- **Was ist eine Fuzzy-Menge?**
Eine Abbildung von einem Grundraum X in $[0, 1]$.
- **Was ist ein Singleton?**
Ein einzelnes Wertepaar (oft als Funktion betrachtet).
- **Kann man den Begriff der Fuzzy-Menge mit Wahrscheinlichkeit vergleichen?**
Nein bei Wahrscheinlichkeiten gehört das Objekt entweder zu Menge oder nicht. Bei Fuzzy-Mengen ist es anders.
- **Was ist Support und was ist Toleranz von Fuzzymengen?**
Support sind alle Eingaben mit Wert über 0. Toleranz sind alle Eingaben mit Wert 1.
- **Was ist eine Linguistische Variable?**
Eine Variable, die einen der Wert (Linguistische Term) annehmen kann.
- **Wie kann man Fuzzy-Mengen repräsentieren?**
Durch α -Schnitte. Man teilt die Menge in Bereiche mit Zugehörigkeitsgrad mehr als α . Man speichert eine Approximation von α -Schnitten für endliche Anzahl von α -Werten.
- **Was versteht man unter Fuzzifizierung?**
Bestimmen der Zugehörigkeitsgrade einer Eingabe bezüglich aller Terme.
- **Wie definiert man üblicherweise die Zugehörigkeitsgrade?**
Summe der Grade für einen scharfen Wert sollte 1 sein.

10.1 Einfache Operationen auf Fuzzy-Mengen

- **Welche Bedingungen muss eine t -Norm erfüllen?**
Existenz eines neutralen Elementes (1), Monotonie in beiden Argumenten, Kommutativität und Assoziativität.
- **Welche Bedingungen muss eine t -CoNorm erfüllen?**
Existenz eines neutralen Elementes (0), Monotonie in beiden Argumenten, Kommutativität und Assoziativität.

- **Wie kann man Negation definieren?**
Von der 1 die gegebene Fuzzymenge abziehen.
- **Welche Normen/Conormen kennen sie?**
Max/Min, Drastisches Produkt/Drastische Summe (Minimum, wenn von beiden, wenn eines davon 1 ist und sonst 0).
- **Was ist eine Fuzzy-Relation?**
Eine Fuzzymenge über eine Kreuzprodukt-Raum.
- **Warum ist die exakte Umsetzung der Implikation für Fuzzy-Mengen ungeeignet?**
Weil für 0-Werte die Implikation stets wahr ergibt. Das macht Probleme nachfolgenden Verknüpfungen.
- **Was kann man statt logischer Implikation benutzen?**
Mamdani-Implikation. Einfach das Minimum der Argumente (MIN/MAX-Inferenz). Oder das Produkt (MAX-PROD-Inferenz).
- **Was ist Fuzzy-Inferenz?**
Auswertung von Fuzzy-Regeln.
- **Wann ist eine Prämisse aktiv?**
Wenn ihr Erfüllungsgrad größer 0 ist.
- **Was entspricht einem System von Implikationen?**
Maximumbildung bzw. die Order-Relation der Implikationen.
- **Wie kann man den Inferenzvorgang bei Implikationen anschaulich erklären?**
Bei MAX-MIN-Inferenz entspricht der Bildung der Ergebnismenge für einen scharfen Wert das Abschneiden der Fuzzymenge der Folgerung ab der Höhe des Zugehörigkeitsgrades der Prämisse für den scharfen Wert.
- **Was ist das Ziel der Defuzzifizierung?**
Bilden eines scharfen Wertes nach der Inferenz für die Ausgabe.
- **Wie funktioniert Defuzzifizierung?**
Man kann z.B. Mittelwert der Maximawerte oder linken oder rechten Rand nehmen, Schwerpunkt oder Schwerpunkt mit SUM-MIN-Inferenz, Höhenmethode.
- **Welche Vorteile und Nachteile haben maximale Werte?**
Rechenaufwand ist gering, Unabhängigkeit vom Erfüllungsgrad der Regel mit maximalen Erfüllungsgrad. Bei mehreren Regeln mit einem Maximum sollte man Mittelung vorziehen.
- **Wie funktioniert die Schwerpunktmethode?**
Man nimmt den Eingabewert vom Schwerpunkt der Fuzzy-Menge, den man mit Hilfe der Integration errechnet.
- **Welche Vorteile und Nachteile hat die Schwerpunktmethode?**
Hoher Aufwand, Ergebnisse mit einem geringem Erfüllungsgrad möglich. Randerweiterung manchmal notwendig.
- **Worin besteht der Unterschied zwischen der MAX-Min und der SUM-MIN Inferenz?**
Man nimmt statt der Maximumfunktion eine lineare Funktion. Danach ist die Schwerpunktberechnung $\int y\nu(y)dy / \int \nu(y)dy$ leichter. Man kann den Schwerpunktewert aus den Schwerpunkten der Eingabeflächen berechnen.
- **Welche Vorteile und Nachteile hat die SUM-MIN-Inferenz?**
Es können Zugehörigkeitsgrad entstehen, die größer als 1 sind. Einfluss mehrere aktiver Regeln mit gleicher Folgerung wird verstärkt, Einfache Berechnung, Wenn man nur die Regel mit gleicher Folgerung berücksichtigt, die jeweils den höchsten Grad besitzt, so ist die Abweichung vom Schwerpunkt gering.

- **Wie funktioniert die Höhenmethode?**
Wenn man dreieckige Zugehörigkeitsfunktionen verwendet, so kann man einfach $\sum m_i H_i / \sum H_i$ verwendet, wobei m_i der Modalwert und H_i der entsprechende Erfüllungsgrad ist.
- **Welche Vorteile und Nachteile hat die Höhenmethode?**
Näherungsmethode an Schwerepunktmethode, Geringer Aufwand, Nur Dreiecksfunktionen.

11 Anwendungen von Fuzzy-Systemen

- **Welche Parameter muß man bei Fuzzy-Systemen einstellen?**
Zugehörigkeitsfunktionen, Operationen, Regelbasis, Inferenzmechanismus (Implikation und Überlagerung) und Defuzzifizierung.
- **Wie funktioniert ein Fuzzy-Regler?**
Es gibt eine Regelstrecke und ein Fuzzycontroller, der bei Abweichung vom Istwert und Sollwert die Strecke nachregelt mit Aktoren.
- **Welche Beispiele von Fuzzy-Systemen kennen Sie?**
Inverses Pendel, RoboCUP.
- **Welche Vorteile haben Fuzzy-Systeme gegenüber neuronalen Netzen?**
Gute Interpretationsfähigkeit, Einfache Implementierung, Kein Training.
- **Welche Nachteile haben Fuzzy-Systeme gegenüber neuronalen Netzen?**
Regelbasis muss verfügbar sein (Expertenwissen), Keine Lernfähigkeit, viele Parameter, Kein Erfolg bei möglichen Tuningversuchen.

Index

Aktivierungsfunktion, 3
Axon, 2

Dendrit, 2

einlagiges Perzeptrennetz, 4
Ensembles, 2

Fuzzy-Logik, 3

Neokortex, 3
Netzeingabefunktion, 3
Neuron, 2
Neurotransmitter, 2

Parallelität, 3
Perzeptron, 3

Regelung, 2

Steuerung, 2
Synapse, 2
synaptische Plastizität, 2

Topologieerhaltung, 3
Training, 4