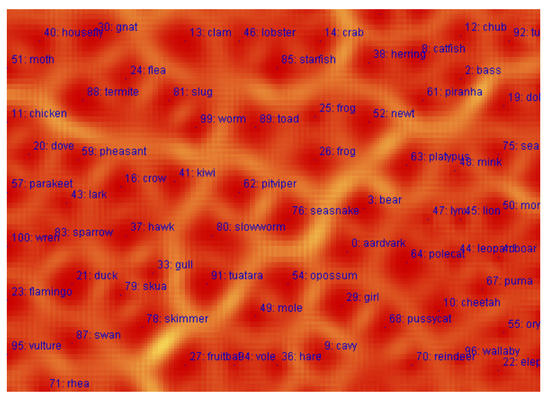
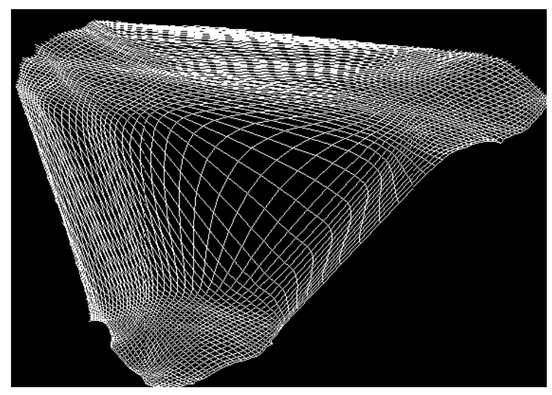
FACHHOCHSCHULE BRANDENBURG
FACHBEREICH INFORMATIK UND MEDIEN



Klassifizieren und Visualisieren von Daten mit Selbstorganisierenden Karten

Diplomarbeit, vorgelegt von Sven Schröder



Aufgabenstellung

Ziel der Arbeit ist die Untersuchung von Visualisierungsmöglichkeiten von Daten mit Hilfe von Selbstorganisierenden Karten. Hierbei sind sowohl musterbezogene als auch musterunabhängige Varianten zu betrachten. Die Verfahren sollen prototypisch in ein Programmsystem umgesetzt werden, welches ohne Installation über das Netz nutzbar ist. Der Lernprozess ist geeignet darzustellen. Hierzu sind Selbstorganisierende Karten im theoretischen Überblick zu beleuchten und für den Schwerpunkt Visualisierung trainierter Karten zu vertiefen. Anhand der im Verlauf der Diplomarbeit erzielten praktischen und theoretischen Erkenntnisse soll eine Bewertung und Systematisierung der Visualisierungsvarianten erstellt werden. Dabei sollen Aspekte wie Lesbarkeit, Informationsgehalt, Umsetzungskomplexität und andere Kriterien beurteilt werden.

Kohonen-Netze

sind ein bestimmter Typ Neuronaler Netze. Sie sind nach ihrem Erfinder Teuvo Kohonen benannt und entsprechen wahrscheinlich am ehesten dem Aufbau der Hirnrinde. In ihr wurden bereits Neuronale Karten mit ähnlichen Eigenschaften, wie sie Kohonen-Netzen eigen sind, nachgewiesen. Im biologischen Vorbild, wie auch in der programm-

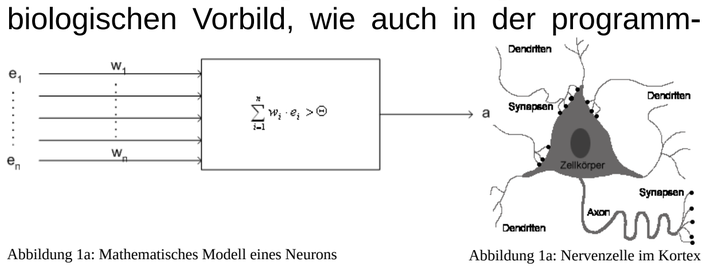


Abbildung 1a: Mathematisches Modell eines Neurons

Abbildung 1a: Nervenzelle im Kortex

technischen Umsetzung, findet das Prinzip der lateralen Hemmung Anwendung. Kohonen-Netze bestehen aus 2 Schichten, der Kartenschicht und der Eingabeschicht. Die Neuronen der Kartenschicht sind miteinander verbunden, wobei mehrere Strukturmuster möglich sind. Üblich sind quadratische und hexagonale Gitter. Die Form kann planar, wie auch toroid sein. In der einfachsten Form ist eine Kohonenkarte durch eine rechteckige quadratische Gittermatrix realisiert (Abbildung 2). Desweiteren ist jedes Kartenneuron mit jedem Neuron der Eingabeschicht verbunden. Jede Verbindung von einem Kartenneuron zu einem Eingabeneuron wird mit einem Wert gewichtet. An den Eingangsneuronen werden während des Trainings Eingabemuster angelegt. Die Muster bestehen aus Dezimalzahlen, welche aus den eingegebenen Daten gebildet werden.

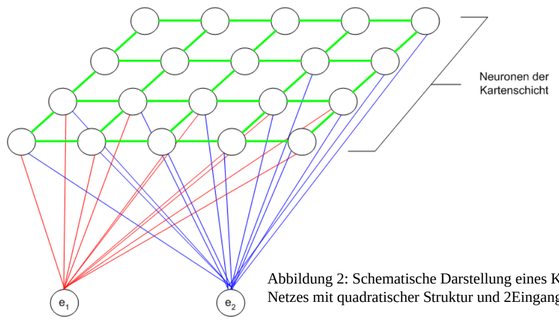


Abbildung 2: Schematische Darstellung eines Kohonen-Netzes mit quadratischer Struktur und 2Eingangsneuronen

Das Trainieren eines Kohonen-Netzes erfolgt in Lernschritten. Während eines Lernschrittes wird ein Muster an die Neuronen angelegt und das Siegerneuron ermittelt. Dies kann durch Berechnung des Euklidischen Abstands zwischen dem Gewichtsvektor eines Kartenneurons, welcher durch seine Verbindungen zu den Eingabeneuronen gebildet wird, und dem Mustervektor erfolgen. Das Kartenneuron mit dem kleinsten Abstand ist das Siegerneuron.

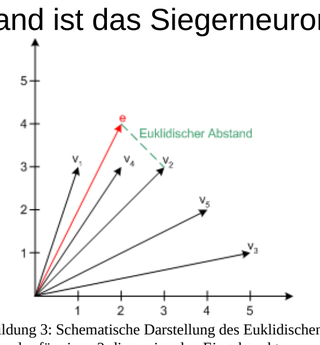


Abbildung 3: Schematische Darstellung des Euklidischen Abstandes für einen 2-dimensionalen Eingabevektor

Abbildung 3 zeigt ein Beispiel zum Euklidischen Abstand anhand eines 2-dimensionalen Eingabevektors. Den kleinsten Abstand vom Eingabevektor e hat der Gewichtsvektor für das Neuron V4. Die Bestimmung über das maximale Skalarprodukt ist auch möglich.

Die Verbindungsgewichte des Siegerneurons sowie seiner Umgebung werden anschließend über eine Lernrate an das Muster angepasst. Die Bestimmung der Nachbarschaft erfolgt über eine mathematische Funktion, z.B. Kegel. Die Abbildung 4 zeigt eine Reihe weiterer Nachbarfunktionen.

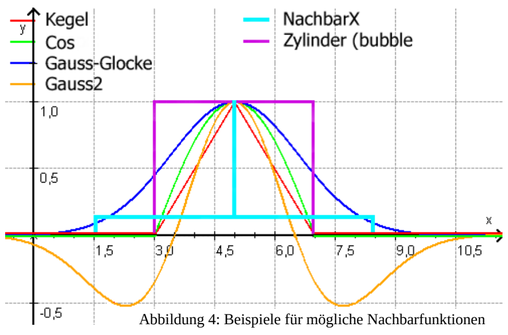


Abbildung 4: Beispiele für mögliche Nachbarfunktionen

Nach Abschluß des Trainings kann das Netz unbekannte Muster aufgrund des Wissens aus den Trainingsmustern einordnen.

Visualisierungsmöglichkeiten

Kohonen-Netze clustern die angelegten Muster in Bezug auf ihre Ähnlichkeit und Häufigkeit untereinander. Die Darstellung auch großer Datenmengen erfolgt in 2- und 3-dimensionalen Perspektiven. Dadurch ist es dem Betrachter möglich, die Zusammenhänge in den Daten, ähnlich dem Betrachten einer Landkarte, zu erfassen.

Es wird zwischen musterabhängigen und musterunabhängigen Darstellungen unterschieden, welche in der Diplomarbeit erfasst und bewertet wurden (Abbildung 5 und 6).

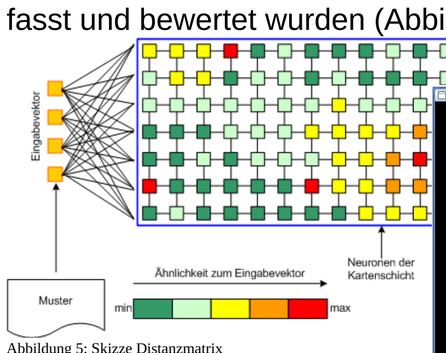


Abbildung 5: Skizze Distanzmatrix

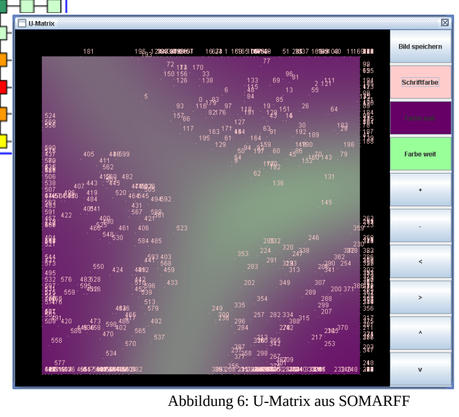


Abbildung 6: U-Matrix aus SOMARFF

Implementierung

Der praktische Teil der Arbeit umfasst die Erstellung der Anwendung SOMARFF, welche zur Clusterung von Daten aus arff-Dateien und deren Ausgabe in unterschiedlichen Perspektiven benutzt wird. Das Programm wurde in der Programmiersprache JAVA realisiert. Die Erstellung des Neuronalen Netzwerkes und des Parsers erfolgte ohne Nutzung von externen Bibliotheken oder Frameworks.

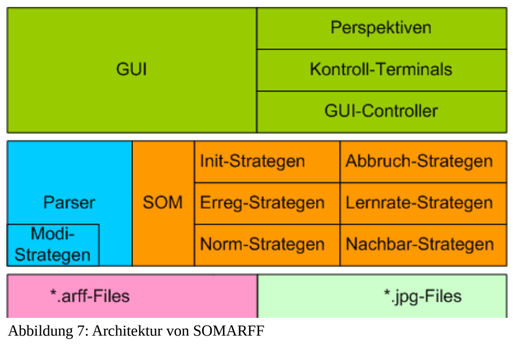


Abbildung 7: Architektur von SOMARFF

Die Implementierung wurde in 3 Module gegliedert. Das Modul GUI beschäftigt sich mit der Programmbedienung und der Visualiserung der Daten. Das Modul Parser kümmert sich um das Lesen und Übersetzen der Daten in Muster, welche an das Kohonen-Netz angelegt werden. Das Modul SOM initialisiert das Kohonen-Netz und steuert das Netztraining. Die Abbildung 7 verdeutlicht die Architektur.

Da das Programm um neue Funktionen erweiterbar und leicht wartbar sein sollte, erfolgte die Umsetzung in einer Mehrschichtenarchitektur. Weiterhin fanden die Entwurfsmuster "Abstrakte Fabrik", "Observer" und "Strategie" Verwendung.

Die Oberfläche wurde in Swing realisiert und verfügt über Panels zur Steuerung des Lernfortschritts, der Anpassung von Lernparametern und Parsereigenschaften, sowie zur Vorgabe der Dimension und Initialisierung des Kohonen-Netzes. Weiterhin wurden eine Reihe 2-dimensionaler Perspektiven umgesetzt. Abbildung 8 zeigt die Panels für die Konfiguration des Parsers, des Trainings und des Kohonen-Netzes. Abbildung 9 zeigt eine Auswahl aus einigen Perspektiven von SOMARFF während eines Trainings.

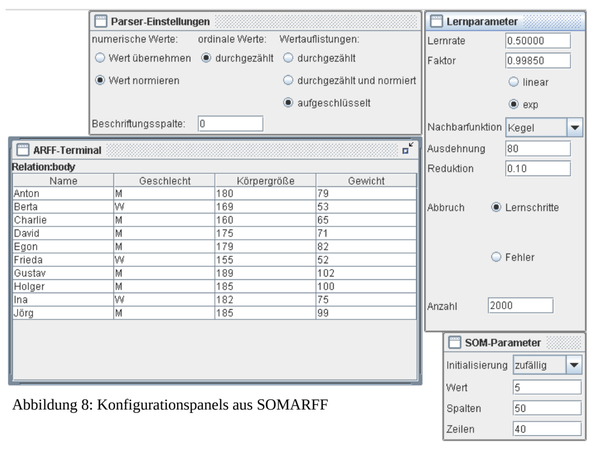


Abbildung 8: Konfigurationspanels aus SOMARFF

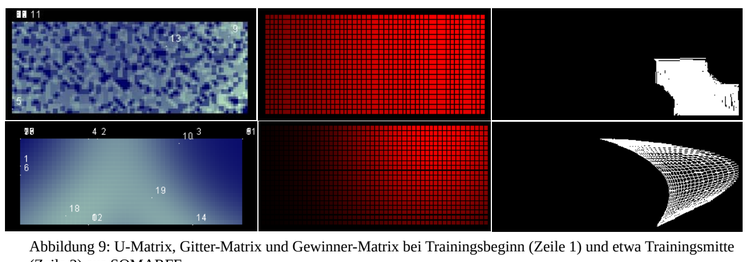


Abbildung 9: U-Matrix, Gitter-Matrix und Gewinner-Matrix bei Trainingsbeginn (Zeile 1) und etwa Trainingsmitte (Zeile 2) aus SOMARFF

Zusammenfassung

Die in der Erarbeitung des theoretischen Teils gewonnenen Erkenntnisse konnten in der praktischen Umsetzung durch Tests bestätigt und vertieft werden. Der Algorithmus ist auch auf kleineren Systemen gut umsetzbar.

Probleme gab es in der visuellen Ausgabe der Kartenergebnisse, da sie sehr rechenintensiv ist. Als besonders kritisch erwies sich die Erstellung der U-Matrix.

In den Tests konnte nachgewiesen werden, dass ein langsamer Lernfortschritt für die Funktionalität des Verfahrens von wesentlicher Bedeutung ist.

Wünschenswert, aber durch die begrenzte Zeit nicht umsetzbar, wäre eine 3-dimensionale Ausgabe über Java3D-API und eine Lade- und Speicherfunktion der SOM in XML-Dateien.

Trotz der langen Trainingsphasen sieht der Autor hohes Potential für diese Technologie. Dies gilt besonders für die Bereiche autonome Roboter und neuronale Implantate. Beispielsweise wäre es denkbar, eines Tages, defekte Hirnareale durch Module aus Kohonen-Netzen ersetzen zu können.

Betreuer: Dipl.-Inform. I. Boersch
Prof. Dr. Ing. J. Heinsohn

Abgabe: 03.03.2006