Hochschule Wismar
Fachhochschule für Technik, Wirtschaft und Gestaltung
Fachbereich Wirtschaft

Hochschule Wismar
University of Technology, Business and Design
Department of Business

Uwe Lämmel

Data-Mining
mittels künstlicher neuronaler Netze

Heft 07 / 2003

W D P

Wismarer Diskussionspapiere / Wismar Discussion Papers

Der Fachbereich Wirtschaft der Hochschule Wismar, Fachhochschule für Technik, Wirtschaft und Gestaltung bietet die Studiengänge Betriebswirtschaft, Management sozialer Dienstleistungen, Wirtschaftsinformatik und Wirtschaftsrecht an. Gegenstand der Ausbildung sind die verschiedenen Aspekte des Wirtschaftens in der Unternehmung, der modernen Verwaltungstätigkeit im sozialen Bereich, der Verbindung von angewandter Informatik und Wirtschaftswissenschaften sowie des Rechts im Bereich der Wirtschaft.

Nähere Informationen zu Studienangebot, Forschung und Ansprechpartnern finden Sie auf unserer Homepage im World Wide Web (WWW): http://www.wi.hs-wismar.de/.

Herausgeber: Prof. Dr. Jost W. Kramer
Fachbereich Wirtschaft
Hochschule Wismar
Fachhochschule für Technik, Wirtschaft und Gestaltung
Philipp-Müller-Straße
Postfach 12 10
D – 23966 Wismar
Telefon: ++49 / (0)3841 / 753 441
Fax: ++49 / (0)3841 / 753 131
e-mail: j.kramer@wi.hs-wismar.de

ISSN 1612-0884
ISBN 3-910102-31-X

JEL-Klassifikation C80, Z00

Inhaltsverzeichnis

Vorwort **4**

1. Wissensextraktion – Eine Einleitung **5**

2. Data-Mining **7**

2.1. Begriffe und Vorgehensweisen 7
2.2. Verfahren des Data-Mining 9
2.2.1. Entscheidungsbäume 10
2.2.2. Statistische Methoden 10
2.2.3. Maschinelles Lernen 11
2.2.4. Cluster-Analyse 11

3. Data-Mining mittels künstlicher neuronaler Netze **11**

3.1.Neuronale Netze 11
3.1.1. Das Künstliche Neuron 12
3.1.2. Aufbau, Arbeitsweise und Anwendung 13
3.2.Lernverfahren 14
3.2.1. Überwachtes Lernen: Backpropagation-Algorithmen 15
3.2.2. Unüberwachtes Lernen: Wettbewerbslernverfahren 16
3.3.Klassifikation 16
3.3.1. Backpropagation-Netze 17
3.3.2. Hopfield-Netz 19
3.4.Clusterung 20
3.4.1. Selbstorganisierende Karten 20
3.4.2. ART-Netze 23
3.4.3. Neuronale Gase 24

4. Anwendungen **25**

4.1.Analyse einer Umfrage zum Verkehrsverhalten Wismarer Studenten 25
4.2.Risiko-Analyse in einer Oberfinanzdirektion 26
4.3.Auswertung von Genexpressionsdaten 29

5. Werkzeuge für den Einsatz von Neuronalen Netzen **30**

6. Ausblick **32**

6.1.Offene Fragen 33
6.2.Data-Mining und Neuronale Netze in der Lehre 34

Literaturverzeichnis **35**

Autorenangaben **36**

Vorwort

Dieses Wismarer Diskussionspapier entspricht dem Bericht, den der Autor im Ergebnis des Forschungssemesters im Jahre 2002 verfasst hat. Der Bericht diente als Ausgangspunkt für die Arbeiten im Rahmen des vom Ministeriums für Bildung und Kultur des Landes Mecklenburg Vorpommern geförderten Projekts: Neuronale Netze für die Wissensextraktion aus Proteomdaten.

Aktuell werden die Arbeiten fortgesetzt im Rahmen einer Projekt-Vorlauf-Phase „Data-Mining Engineering“. Gleichzeitig wurde unter Beteiligung des Autors ein Antrag im Rahmen der Team-FH-Förderung gestellt, der insbesondere die Anwendung von Data-Mining Methoden im Bereich des Finanzwesens zum Inhalt hat.

Das Data-Mining ist nach wie vor eine aktuelle Thematik, da viele Techniken zwar bekannt, konkrete Vorgehensweisen, die nachweislich zu verwertbaren Resultaten in der Datenanalyse führen, jedoch allgemein nicht bekannt sind. Andererseits darf der betriebswirtschaftliche Nutzen, der aus einer Datenauswertung gezogen werden kann, nicht unterschätzt werden.

Die aktuellen Untersuchungen widmen sich insbesondere der effektiven Gestaltung des Analyse-Prozesses. Dabei liegt der Schwerpunkt auf der Beschreibung von Work-Flows, die dann zur Automatisierung des Prozesses herangezogen werden können. Im Ergebnis wird es möglich sein, die notwendigen Experimente im Zuge einer konkreten Data-Mining-Aufgabe semiautomatisch ablaufen zu lassen.

Wismar, August 2003 Uwe Lämmel

1. Wissensextraktion – eine Einleitung

Die Fortschritte auf dem Gebiet der Informations- und Kommunikationstechnologien haben die Möglichkeit eröffnet Daten massenhaft zu erfassen und zu speichern. Derartige Daten sind nutzlos, solange die diesen Daten innewohnenden Informationen nicht herausgefiltert werden (können). Erst auf der Basis von Informationen, die Zusammenhänge zwischen den Daten offenbaren, lässt sich Wissen ableiten, und erst dieses Wissen ermöglicht die Rückkopplung auf die Gestaltung des ursprünglichen Prozesses, des Prozesses über den Daten erfasst wurden.

Es entstand das geflügelte Wort:

Wir ertrinken in Daten und hungern nach Wissen.

Wird im Zuge einer Werbekampagne erfasst, wer von den angeschriebenen potentiellen Kunden zu einem tatsächlichen Kunden wurde und in welchem Umfang dieser Waren erworben hat, so kann man durch eine Datenauswertung ermitteln, wie eine nächste Werbekampagne effektiver gestalten kann. Zwei mittlerweile klassische Beispiele, die in vielen Literaturquellen [Schmidt97] erwähnt werden, betreffen das Kaufverhalten:

- Die englische Lebensmittelkette "Safeway" fand mit Data-Mining heraus, dass ein bestimmter Käse, der nur an Platz 209 der Verkaufsrangliste lag, hauptsächlich von ihren besten, umsatzstärksten Kunden gekauft wurde. Hätte das Unternehmen diesen umsatzmäßig unwichtigen Käse aus dem Sortiment genommen, hätte es damit seine beste Kundschaft verärgert. [CNC97]
- Die Gemischtwarenkette "Walmart" in den USA wollte mehr über das Kaufverhalten ihrer Kunden wissen. Hierzu wurden die Daten über die Verkäufe, die mittels Barcodelesern erfasst wurden, mit einer Knowledge-Discovery-Methode analysiert. Ergebnis: Wer Grußkarten kauft, so fand man heraus, kauft auch Kosmetika (kein Ergebnis, das von einem Werbepsychologen vorausgesagt worden wäre!). Walmart räumte die Läden um und stellte Grußkarten neben Kosmetika. Der Erfolg: dreißig Prozent Umsatzwachstum bei beiden Produktgruppen. [Klu97]

Natürlich beschränkt sich die beschriebene Situation nicht nur auf Beispiele aus dem Kundenverhalten. Das Problem, aus Massendaten Erkenntnisse zu gewinnen, um bestimmte Prozesse effektiver gestalten zu können, steht in einer Vielzahl von Anwendungsbereichen.

In der Genomforschung werden in sogenannten Chip-Experimenten große Mengen von Daten gewonnen, die sogenannten Expressionsdaten. Aus der Auswertung dieser Daten versucht man zu ermitteln, welche Gene unter gewissen Bedingungen expressioniert sind. Damit lassen sich Rückschlüsse auf die Bedeutung dieser Gene vornehmen.

Auch Prognose-Probleme lassen sich unter dem Gesichtspunkt der Wissensextraktion betrachten. Aus den Daten des Energieverbrauchs einer gewissen Region sind Vorhersagen für die zukünftige Energiebereitstellung ableitbar. Wissen über die zu liefernde Menge an Elektroenergie ist gerade in dem liberalisierten Strommarkt von großer Bedeutung.

In jüngster Zeit werden die Aktivitäten zur Datenhaltung in Datenbanken, diese möglichst integriert in ein firmenweites Data Warehouse, die elektronische Dokumentenverwaltung sowie die Datenanalyse im Sinne des Data-Mining zusammengefasst unter dem Oberbegriff „Wissensmanagement".

Die vorliegende Arbeit ist dem Einsatz einer speziellen Technik, der künstlichen neuronalen Netze, im Data-Mining gewidmet. Es wird eine Einführung in die Technik der neuronalen Netze einerseits und in das Anwendungsgebiet des Data-Mining andererseits gegeben.

Die Vielzahl der in der Literatur in diesem Zusammenhang verwendeten Begriffe lässt es notwendig erscheinen, eine Begriffsklärung vorzunehmen. Es werden im folgenden insbesondere die Begriffe näher betrachtet, die für die Themenstellung des Data-Mining mittels künstlicher neuronaler Netze von besonderer Bedeutung sind.

Unter **Data-Mining (DM)** versteht man das Herausarbeiten von Abhängigkeiten innerhalb der Datenmenge. Das in den Daten implizit vorhandene Wissen wird dabei explizit gemacht. Das Wissen wird aus den Daten herausgeholt, somit extrahiert: **Wissensextraktion**. Während in der deutschsprachigen Literatur die Begriffe Data-Mining und **Knowledge Discovery in Databases (KDD)** als synonyme Begriffe behandelt werden, wird im englischsprachigen Raum das Data-Mining als eine konkrete Form des KDD betrachtet.

Somit beleuchten beide Begriffe unterschiedliche Seiten einer Medaille: Data-Mining stellt die Quelle, die Massendaten, in den Mittelpunkt, aus denen, wie im Bergbau, unter vielem Schutt das wertvolle Gut herausgefunden werden muss. KDD fokussiert das Ergebnis: Wissen ist zu extrahieren.

Hinter dem Ansatz der künstlichen neuronalen Netze verbirgt sich ein „Nachbau" der natürlichen neuronalen Netze. Das Zusammenschalten vieler einfacher Zellen erzielt eine erstaunliche Leistungsfähigkeit. Darüber hinaus können wir Menschen aus Beispielen lernen bzw. auch neue Zusammenhänge erkennen, die uns nicht vorher explizit mitgeteilt wurden.

Die künstlichen neuronalen Netze ermöglichen nun prinzipiell ebenso wie ihre biologischen Vorbilder das Lernen aus Beispielen bzw. auch das „Erkennen" von bis dato unbekannten Zusammenhängen. Diese Fähigkeiten sind gerade auf dem Gebiet des Data-Mining besonders gefragt. Sie stellen eine konkrete Form des Data-Mining im engeren Sinne dar. Mittels künstlicher neuronaler Netze können Zusammenhänge zwischen den Daten sichtbar gemacht werden.

Das Kapitel 2 gibt einen kurzen Überblick insbesondere über verschiedene Techniken des Data-Mining. Dabei steht im Mittelpunkt die Aussage, dass es sehr viele unterschiedliche Herangehensweisen und Lösungsansätze gibt unter denen künstliche neuronale Netze einen Ansatz darstellen. Im anschließenden Kapitel 4 wird dann der Einsatz neuronaler Netze für das Data-Mining diskutiert. Neben einer Einführung in das Konzept künstlicher neuronaler Netze werden die Formen neuronaler Netze ausgewählt, die für die Anwendung im Data-Mining geeignet sind.

Konkrete Anwendungsbeispiele werden im Kapitel 4 vorgestellt. Es handelt sich dabei um Arbeiten, die konkret am Fachbereich Wirtschaft der Hochschule Wismar durchgeführt wurden. Die Arbeiten und Experimente wurden unter Nutzung konkreter Software vorgenommen. Die verwendete Software wird im Kapitel 5 behandelt.

Der Einsatz neuronaler Netze im Data-Mining stellt auch keinen Königsweg dar. Die Ableitung von Wissen aus den Daten gestaltet sich schwierig. Eine Vielzahl von Experimenten ist nötig, um zu zufriedenstellenden Ergebnissen zu gelangen. Viele offene Fragen, offene Probleme sowie Wünsche für weitere Entwicklungen existieren. Das Kapitel 6 gibt einen Ausblick, welche Fragen in Zukunft weiter zu bearbeiten sind.

Die Arbeit kann zur Einarbeitung in die Thematik dienen und gibt über die Vielzahl von Literaturverweisen Anregungen zur vertiefenden Beschäftigung mit diesem sehr aktuellen Thema.

2. Data-Mining

„Die Welt wird immer komplexer, überschwemmt uns mit ihren Daten, und das Data-Mining ist unsere einzige Hoffnung, die ihnen zugrunde liegenden Muster zu erkennen.“[1]

2.1. Begriff und Vorgehensweise

Data-Mining oder Knowledge Discovery in Databases ist ein in den letzten Jahren stark in den Mittelpunkt des Interesses gerücktes Teilgebiet der Informatik. Nicht zuletzt sind die technischen Möglichkeiten zum Erfassen und Speichern von Daten eine der wesentlichen Ursachen für diese Schwerpunktsetzung. Natürlich gibt es dabei die Erkenntnis, dass die Daten alleine wenig nutzbringend sind, solange diese nicht ausgewertet werden können.

Daten-Auswertung im Sinne des Data-Mining meint die Suche nach neuen Zusammenhängen innerhalb der Daten, wobei natürlich insbesondere solche Zusammenhänge gesucht sind, die für die zukünftige Gestaltung von Prozes-

[1] Aus [WittenFrank2001], Seite 3.

sen, z. B. einer Werbeaktion, der Energiebereitstellung oder einer Krankheitsbekämpfung, von besonderem Interesse sind.

„Data-Mining ist die nichttriviale und automatische Suche nach Wissen in Massendaten“ [Lusti2002].

Andere Definitionen beziehen sich auf das Erkennen von Beziehungen zwischen den Daten. Diese Sichtweise ist etwas eng. Aus den Beziehungen zwischen den Daten kann möglicherweise Wissen abgeleitet werden, aber dies muss nicht immer der Fall sein.

Betrachtet man Data-Mining als ein Prozess im Sinne der obigen Definition von Lusti, so kann der komplette Weg vom Problem bis hin zum Erkenntnisgewinn in Etappen unterteilt werden kann (siehe Abbildung 1).

Abbildung 1: Etappen des Data-Mining (nach [Runkler2000])

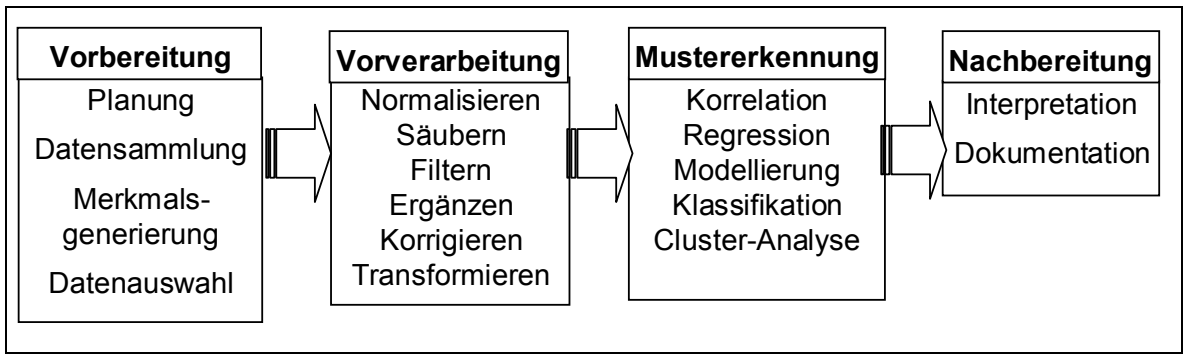
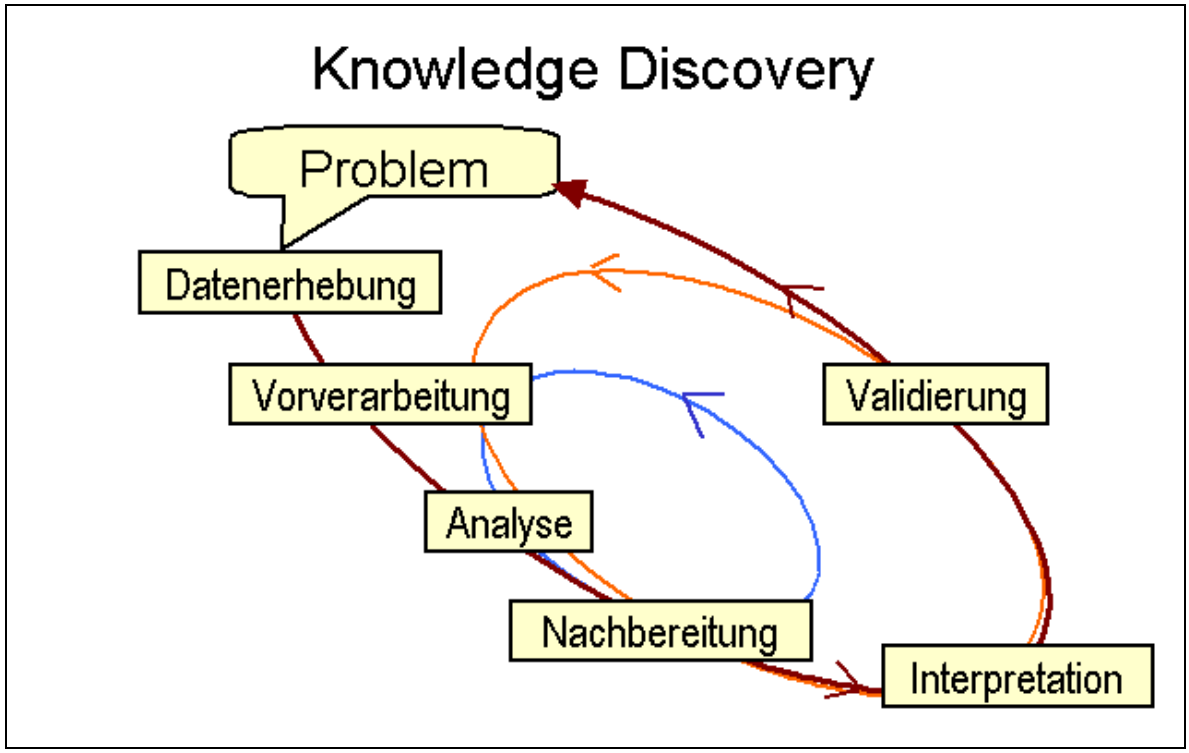


Abbildung 1 spiegelt dabei nur den sequentiellen Ablauf der vier Etappen wider, der in dieser Form nur nach vielen Experimenten durchlaufen werden kann. Ähnlich wie der Prozess der Software-Entwicklung ist auch das Extrahieren von Wissen aus einem Anwendungsfall ein eher zyklischer Prozess. Abbildung 2 verdeutlicht diese Sichtweise.

Abbildung 2: Zyklen im Prozess der Wissensextraktion



Der äußere, dunkelrote Kreis schließt direkt an die sequentielle Sichtweise an. Aus dem Problem heraus werden Daten erhoben, analysiert und das neu erworbene Wissen wirkt auf die Anwendung zurück. Die Ergebnisse werden validiert.

Der mittlere, gelbe und der innere, blaue Kreis verdeutlichen den iterativen Charakter der Wissensextraktion. Um überhaupt zu auswertbaren Ergebnissen zu gelangen, die eine Interpretation erlauben, ist ein sehr technischer, auf die Analysetechnik zugeschnittener Kreis (blau) zu durchlaufen. Dabei werden verschiedene Transformationen, Parameter des Analyse-Algorithmus sowie unterschiedliche Visualisierungen eingesetzt.

Kann Wissen aus den Analyse-Ergebnissen abgeleitet werden, so kann dieses Wissen sich letztendlich als doch nicht mit der Anwendung konform gehend erweisen. Damit ergibt sich ein weiterer Bearbeitungszyklus (gelb).

2.2. Verfahren des Data-Mining

Es gibt zahlreiche Verfahren, wobei die einzelnen Verfahren Konkretisierungen oder Verfeinerungen einiger weniger Ansätze darstellen. Da der Schwerpunkt der Arbeit im Einsatz neuronaler Netze für das Data-Mining liegt, werden an dieser Stelle nur einige häufig verwendete Ansätze vorgestellt. Eine gute Einführung anhand von Beispielen in mehrere Techniken (ohne neuronale Netze) gibt das Buch von Witten und Frank [WittenFrank2001] .

Misst man die Data-Mining-Verfahren an der Definition des Data-Mining [Lusti2002] so muss man feststellen, dass die Verfahren kein Wissen liefern, sondern sie liefern Beziehungen zwischen den Daten aus denen durch geeignete Interpretation Wissen abgeleitet werden kann. Die als Data-Mining-Verfahren bekannten Algorithmen sind in die Phase 3 Mustererkennung (siehe Abbildung 1) einzuordnen und können so nur eine, wenn auch nicht unwesentliche, Teilaufgabe im Data-Mining-Prozess lösen.

2.2.1. Entscheidungsbäume

Hierbei geht man von einer Beispielmenge aus, für die eine gewünschte Klassifikation bekannt ist. Jedes Merkmal der Datenmenge kann dazu benutzt werden alle Datensätze zu klassifizieren. Dabei werden so viele Klassen gebildet, wie dieses Merkmal unterschiedliche Ausprägungen (Werte) annimmt. Um eine möglichst gute Unterteilung zu erhalten, bestimmt man den Informationsgewinn, den eine Klasseneinteilung nach einem Merkmal erzielt. Das Merkmal mit dem höchsten Informationsgewinn wird dann als erster Knoten im Baum gesetzt. Es entstehen so viele Verzweigungen, wie unterschiedliche Werte für das Merkmal auftreten. In jedem Unterbaum wird nun analog weiter unterteilt. Enthält eine Verzweigung dann nur noch Datensätze, die einer Klasse zugehörig sind, so bricht das Verfahren in diesem Unterbaum ab, und andere Unterbäume sind analog weiter zu bearbeiten.

Die Blätter des Baumes geben dann die zugehörige Klasse für alle Datensätze an, die bei einem Weg von der Wurzel zu diesem Blatt führen. Aus einem Baum können leicht Regeln für die Klassifikation abgeleitet werden. Nicht immer ist eine vollständige Klassifikation möglich, da es durchaus gleiche Datensätze mit unterschiedlicher Zuordnung geben kann. Oft werden Entscheidungsbäume daher auch mit Wahrscheinlichkeitsangaben ergänzt.

2.2.2. Statistische Methoden

Statistische Methoden gehen von Häufigkeiten von Merkmalswerten aus und versuchen daraus eine Klasseneinteilung unter Angabe einer Wahrscheinlichkeitsverteilung abzuleiten.

Genau genommen benutzen viele andere Verfahren statistische Elemente ebenso. Als Vertreter der Gruppe der statistischen Verfahren sollen die Regressionsanalyse sowie das Naive-Bayes-Verfahren genannt werden. Letzteres geht von einer Unabhängigkeit der Merkmale aus, die häufig nicht gegeben ist. Trotzdem lasse sich brauchbare Ergebnisse erzielen. Hingewiesen sei auch auf den „Eine-Regel“-Ansatz: Dabei wird die Beispiel-Menge nach jedem Merkmal unter Nutzung der Häufigkeiten eines jeden Merkmalswertes aufgeteilt. Alle Aufteilungen werden verglichen und die mit dem kleinsten Fehler ist

dann das Ergebnis und liefert somit eine einzige Klassifikationsregel für das Problem.

2.2.3. Maschinelles Lernen

Man kann durchaus das Data-Mining als eine Form des maschinellen Lernens aus Beispielen betrachten. In der künstlichen Intelligenz wird unter maschinellem Lernen das Ableiten von Regeln aus Beispielen verstanden, häufig somit ein induktives Lernen. Derartige Ansätze wurden schon lange bevor das Data-Mining aktuell wurde behandelt, siehe [Görz1993].

Man versucht derartige Lernstrategien jetzt auch auf Probleme des Data-Mining anzuwenden.

2.2.4. Cluster-Analyse

Es gibt mehrere Ansätze eine Menge von Daten zu gruppieren, somit Cluster zu erzeugen. Ein besonderes Merkmal ist, dass hierbei von einer Datenmenge ausgegangen wird, für die (noch) keine Klassifikation existiert. Im Sinne der neuronalen Netze spricht man hier dann vom nicht überwachten Lernen, da keine Angaben über das gewünschte Ergebnis vorliegen. Die Verfahren müssen also eine Art Selbstorganisation der Datenmenge ermöglichen.

In die Gruppe der Cluster-Analyse-Verfahren ordnen sich einige neuronale Netze, wie die selbstorganisierenden Karten ein. Kapitel 3 widmet sich diesen Verfahren. Weitere Ansätze sind das k-means-Verfahren bzw. statistische Verfahren.

3. Data-Mining mittels Neuronaler Netze

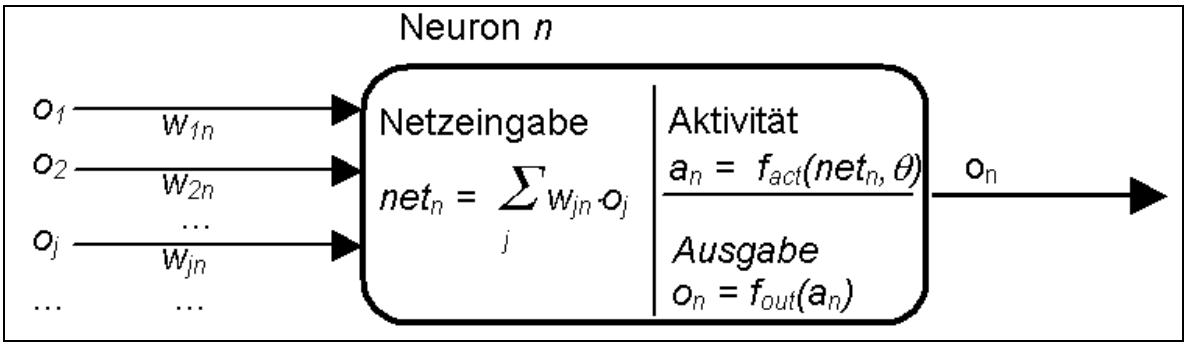
Wie im vorherigen Kapitel bereits ausgeführt kann man Herangehensweisen in Klassifikation bzw. Cluster-Analyse unterteilen. Nach einer Einführung in das Konzept der neuronalen Netze werden die verschiedenen Architekturen in diese beiden Formen unterteilt behandelt.

3.1. Neuronale Netze

Künstliche neuronale Netze stellen den Versuch dar, die Leistungsfähigkeit des menschlichen Gehirns durch Nachbau der Organisationsprinzipien auf einem Computer wenigstens in Ansätzen erreichen zu können. Neuronale Netze versuchen somit die Arbeitsweise eines Gehirns nachzubilden, um Aufgaben zu lösen, die sich sowohl einer algorithmischen als auch einer anderweitigen wissensbasierten Lösung entziehen. Ein (künstliches) **neuronales Netz** entsteht durch die Verknüpfung mehrere (vieler) simpler Einheiten (**Neuronen**), die über Verbindungen Signale austauschen. Ein neuronales Netz ist ein zu-

sammenhängender, gerichteter Graph, wobei zusätzlich zu den Kanten auch die Knoten (Neuronen oder Units genannt) mit Werten (Aktivität) versehen sind. [Lämmel2000]

Abbildung 3: Aufbau eines künstlichen Neurons



3.1.1. Das Künstliche Neuron

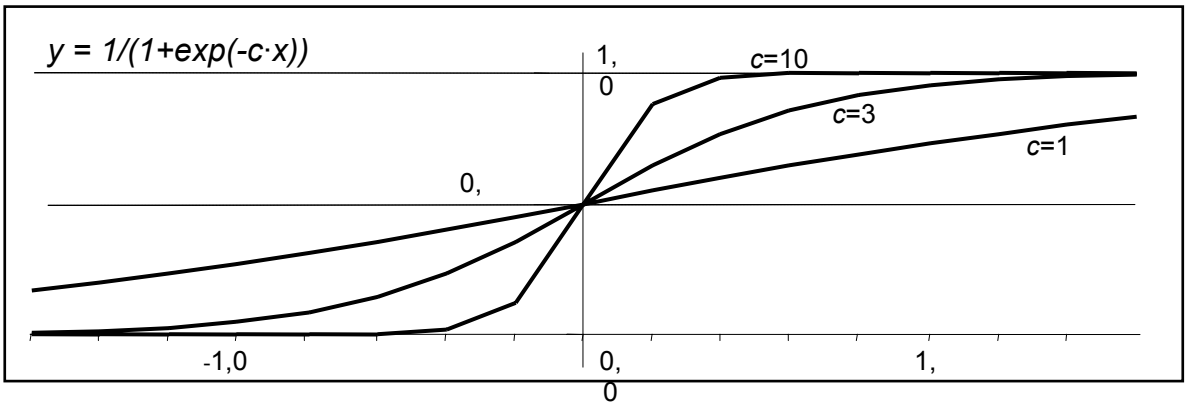
In einem künstlichen Neuron werden die Signale aus den Vorgänger-Neuronen gewichtet aufsummiert (net_i). Daraus wird unter Beachtung eines **Schwellwerts** θ_i eine **Aktivierung** a_i abgeleitet, die zumeist direkt als Ausgabe-Signal $o_i=a_i$ weitergeleitet wird.

Als **Aktivierungsfunktion** wird eine sigmoide Funktion verwendet. Meist ist es eine der folgenden Funktionen:

- Schwellwert-Funktion: $f_{Schwellwert}(x)=\begin{cases}1, & x \geq \theta \\ 0, & \text{sonst}\end{cases}$
- Identität $f(x) = x$
- Logistische Funktion: $f_{\text{logistic}}(x)=\frac{1}{1+e^{-c\cdot x}}$

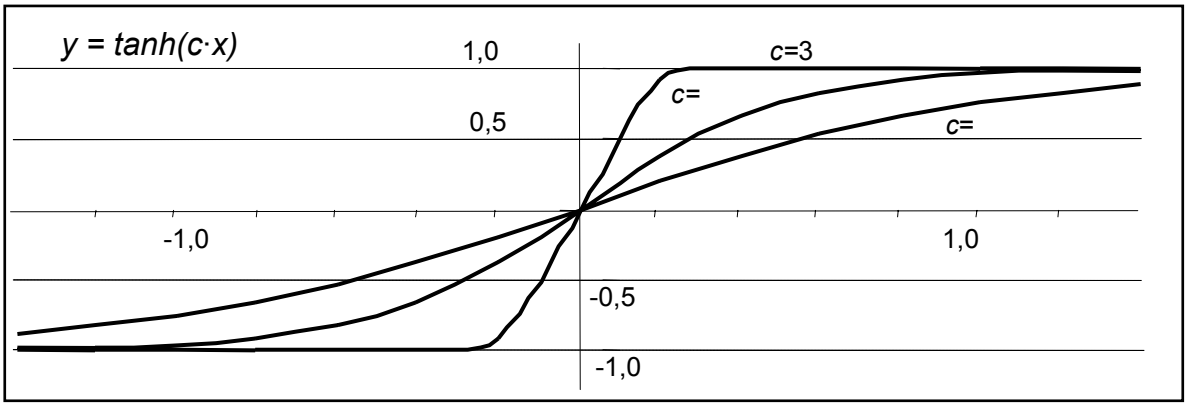
  Im Unterschied zur Schwellwert-Funktion ist die logistische Funktion stetig und somit differenzierbar. Ihr Wertebereich liegt zwischen 0 und 1. Mit Hilfe des Parameters c kann die Steilheit der Kurve gesteuert werden.

Abbildung 4: Funktionsverlauf der logistischen Funktion für verschiedene Parameter



- Tangens hyperbolicus
  Hierbei beträgt der Wertebereich [-1,1]. Dies führt häufig zu deutlicheren Unterscheidungen. Auch steuert ein Parameter c den sigmoiden Charakter der Funktion.

Abbildung 5 : Funktionsverlauf der Funktion tanh für verschiedene Parameter



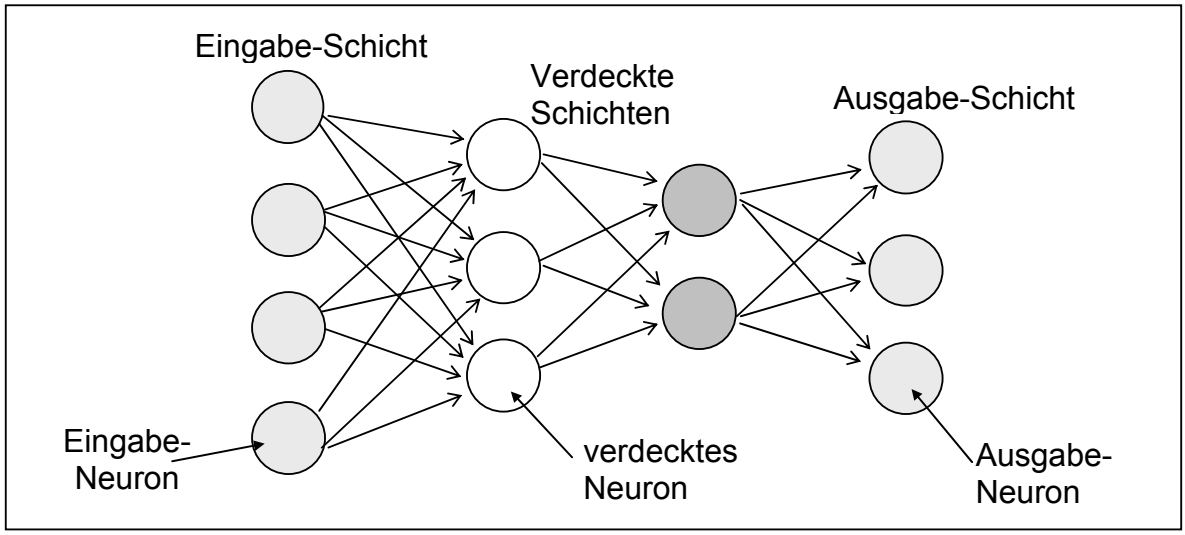
3.1.2. Aufbau, Arbeitsweise und Anwendung

Die Neuronen werden zu Netzen zusammengeknüpft. Die Verbindung zwischen zwei Neuronen ist durch ein (im allgemeinen veränderliches/ trainierbares) Verbindungsgewicht charakterisiert. Ein Netz kann aus mehreren Schichten von Neuronen bestehen: Eingabe-Schicht, Ausgabe-Schicht, verdeckte Schichten. Diese und andere Netzformen werden in den nachfolgenden Abschnitten näher erläutert.

Eine zu verarbeitende Eingabe, das Eingabe-Muster, wird dem Netz präsentiert. Die Aktivierungen der Neuronen der Eingabe-Schicht werden mit den

Werten des Eingabe-Musters belegt. Danach werden alle Aktivierungen der Neuronen neu berechnet. Die Aktivierungen der Ausgabe-Neuronen stellen dann das Ergebnis der Verarbeitung dar.

Abbildung 6: Schichtenaufbau eines vorwärts gerichteten neuronalen Netzes



Die Leistungsfähigkeit ergibt sich zum einen aus der Zahl der Neuronen im Netz. Dabei ist die Größe der Eingabe- sowie der Ausgabe-Schicht durch das zu verarbeitende Problem weitgehend vorbestimmt. Zum anderen sind es die Werte (Gewichte) der Verbindungen und die Schwellwerte in den Neuronen, die das Ergebnis bestimmen. Diese Werte, die Schwellwerte und Gewichte, werden durch geeignete Lernverfahren sukzessive verändert, bis ein gewünschtes Verhalten sich eingestellt hat.

Neuronale Netze können ein durch Training von Beispielen erworbenes Verhalten auf andere, bis dahin unbekannte Situationen anwenden (Generalisierungsfähigkeit). In einem derartigen Netz, in den Gewichten der Verbindungen und den Schwellwerten der Neuronen ist somit implizit Wissen über die Problemlösung gespeichert. Kann das Netz nur auf gelernte Muster und nicht wie gewünscht auf neue Situationen reagieren, ist es zu stark trainiert (Overfitting).

3.2. Lernverfahren

Das Lernen oder Trainieren eines neuronalen Netzes ist kein mystisches Verfahren, sondern beruht auf mathematischen Algorithmen, die eine schrittweise Anpassung der Werte, meist der Gewichte sowie der Schwellwerte, vornehmen. Man kann drei verschiedene Lernverfahren unterscheiden:

- Überwachtes Lernen (supervised learning)
  Anhand vorliegender Beispielergebnisse kann die Abweichung zwi-

schen der gewünschten Ausgabe des Netzes und der tatsächlichen Ausgabe berechnet werden. Dieser Fehler wird benutzt, um die Netzparameter zu verändern mit dem Ziel, den Fehler zu verringern.

- Bestätigendes Lernen (reinforcement learning)
  Lässt sich kein Fehler zwischen einer erwarteten und einer tatsächlichen Ausgabe bestimmen, sondern nur eine Aussage treffen, ob die Netz-Ausgabe richtig oder falsch ist, dann wird nur diese Information benutzt, die Netzparameter anzupassen.
- Nicht überwachtes Lernen (unsupervised learning)
  Liegen keine Beispiel-Ergebnisse vor, dann muss eine Selbstorganisation des Netzes erfolgen.

3.2.1. Überwachtes Lernen: Backpropagation-Algorithmen

Überwachtes Lernen kann nur durchgeführt werden, falls für eine Menge von Eingaben die zugehörigen Ausgaben bekannt sind. Diese Trainingsmenge kann dann zum Training des Netzes verwendet werden. Der bekannteste Algorithmus zum Lernen mehrschichtiger, vorwärts gerichteter neuronaler Netze ist der Backpropagation-Algorithmus. Dabei wird zuerst der Fehler δj in den Neuronen der Ausgabe-Schicht berechnet. Dies ist durch die vorliegenden gewünschten Ausgaben (teaching output tj) möglich. oj ist die tatsächliche Ausgabe des Neurons.

$$\delta_j = \begin{cases} o_j \cdot (1-o_j) \cdot (t_j - o_j), & falls\ j\ Ausgabe\text{-}Neuron \\ o_j \cdot (1-o_j) \cdot \sum_k \delta_k \cdot w_{jk}, & falls\ j\ inneres\ Neuron \end{cases} \qquad (1)$$

Dieses Fehlersignal δj wird dann zur Anpassung der Gewichte wij aller Verbindungen von einem Neuron i zu dem Neuron j benutzt:

$$w'_{ij} = w_{ij} + \eta \cdot o_j \cdot \delta_j$$

Der Faktor η ist der Lernfaktor, häufig $0.2 \leq \eta < 1.0$, der vom Anwender des Lernverfahrens gewählt werden kann.

Das Fehlersignal aller Neuronen in den inneren Schichten, kann nicht direkt berechnet werden, da für diese Neuronen keine Wunsch-Ausgabe bekannt ist. Wie die Formel (1) ausweist, werden hierzu die Fehler-Signale aller Neuronen k herangezogen, zu denen vom Neuron j aus eine Verbindung besteht.

Der Fehler kann somit von der Ausgabe-Schicht schichtweise zurück bis zur ersten inneren Schicht berechnet werden: Backpropagation of error.

Es gibt zahlreiche Vorschläge, dieses Lernverfahren zu beschleunigen. Siehe hierzu [Zell97] oder [LämmelCleve2001].

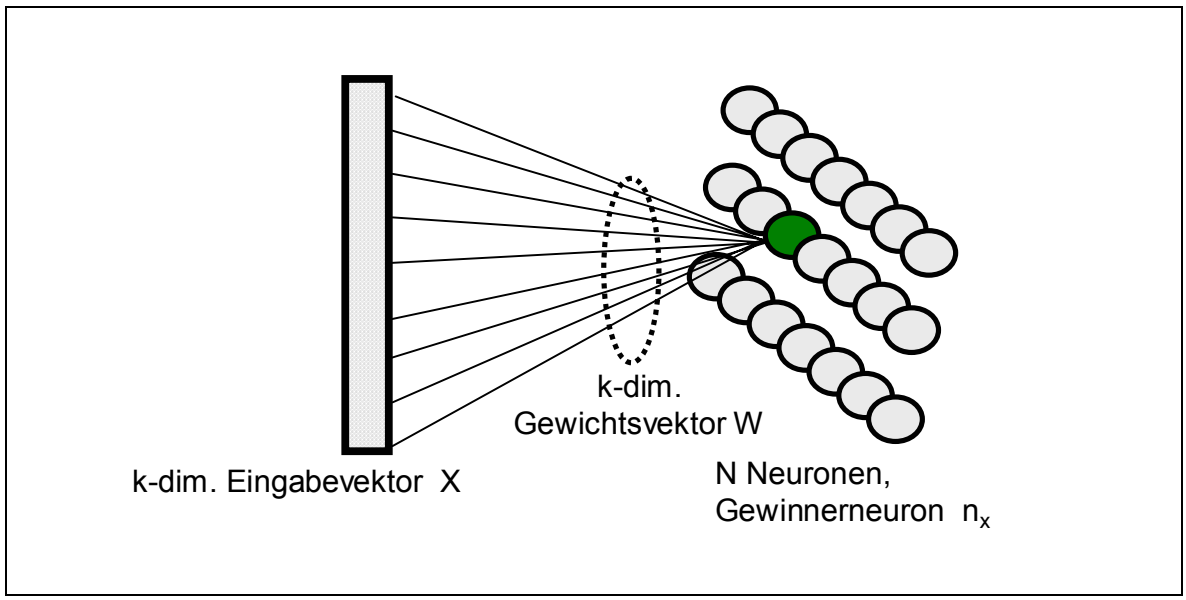
Das überwachte Lernen ist die effektivste Form des Lernens.

3.2.2. Unüberwachtes Lernen: Wettbewerbslernverfahren

Beim unüberwachten Lernen muss das Netz nur unter Ausnutzung der Eingabe-Werte organisiert werden. Häufig wird dabei ein Wettbewerbslernverfahren eingesetzt. Bei diesen Verfahren wird das Neuron des Netzes ermittelt, welches einer konkreten Eingabe in einem bestimmten Sinne am ähnlichsten ist. Dieses ist das Gewinner-Neuron. Die Neuronen treten somit in einem Wettbewerb miteinander.

Man kann eine Eingabe mit einem einzelnen Neuron tatsächlich vergleichen, da man davon ausgeht, dass jedes Eingabe-Neuron eine Verbindung zu jedem anderen Neuron des Netzes besitzt. Diese Verbindungen besitzen Gewichte. Die Gewichte der Verbindungen zu einem bestimmten Neuron stellen somit einen (reellwertigen) Vektor dar. Dieser hat dieselbe Dimension, wie die Eingabe: der Vektor, der durch alle Aktivierungen der Eingabe-Neuronen entsteht.

Abbildung 7: Eingabe- und Verarbeitungsschicht beim Wettbewerbslernen



Die Gewichte der Verbindungen zu diesem Gewinner-Neuron nx und eventuell je nach Verfahren auch zu Neuronen in der Nachbarschaft werden dann so verändert, dass die Ähnlichkeit verstärkt wird. Als Ähnlichkeitsmaß findet der euklidische Abstand oder auch das Skalarprodukt beider Vektoren Anwendung.

3.3. Klassifikation

Für eine Klassifikation muss eine gewünschte Klassen-Einteilung bekannt sein. Es gibt somit eine Menge von Beispieldaten, für die eine Klassen-

Einteilung gegeben ist. Diese Information wird ausgenutzt, um einem neuronalen Netz ein gewünschtes Verhalten anzutrainieren bzw. das Verhalten hinein zu codieren. Das so trainierte Netz kann dann für die Verarbeitung neuer, unbekannter Daten verwendet werden. Das ist dann der praktische Einsatz des Netzes. Es bieten sich somit Netze an, die mit einem überwachten Lernverfahren trainiert werden können. Im folgenden wird der zum einen die am häufigsten eingesetzte Netzart, das mehrschichtige vorwärts gerichtete neuronale Netz, trainiert mit dem Backpropagation-Algorithmus, vorgestellt.

Zum anderen wird ein etwas ungewöhnlicher Ansatz unter Nutzung von Hopfield-Netzen diskutiert.

3.3.1. Backpropagation-Netze

Backpropagation-Netze bieten sich unter der genannten Voraussetzung an, dass eine Menge von Beispieldaten zur Verfügung steht, die zum überwachten Lernen benutzt werden kann. Einschränkend muss hinzugefügt werden, dass in der Regel eine hohe Zahl von Beispielen verfügbar sein muss.

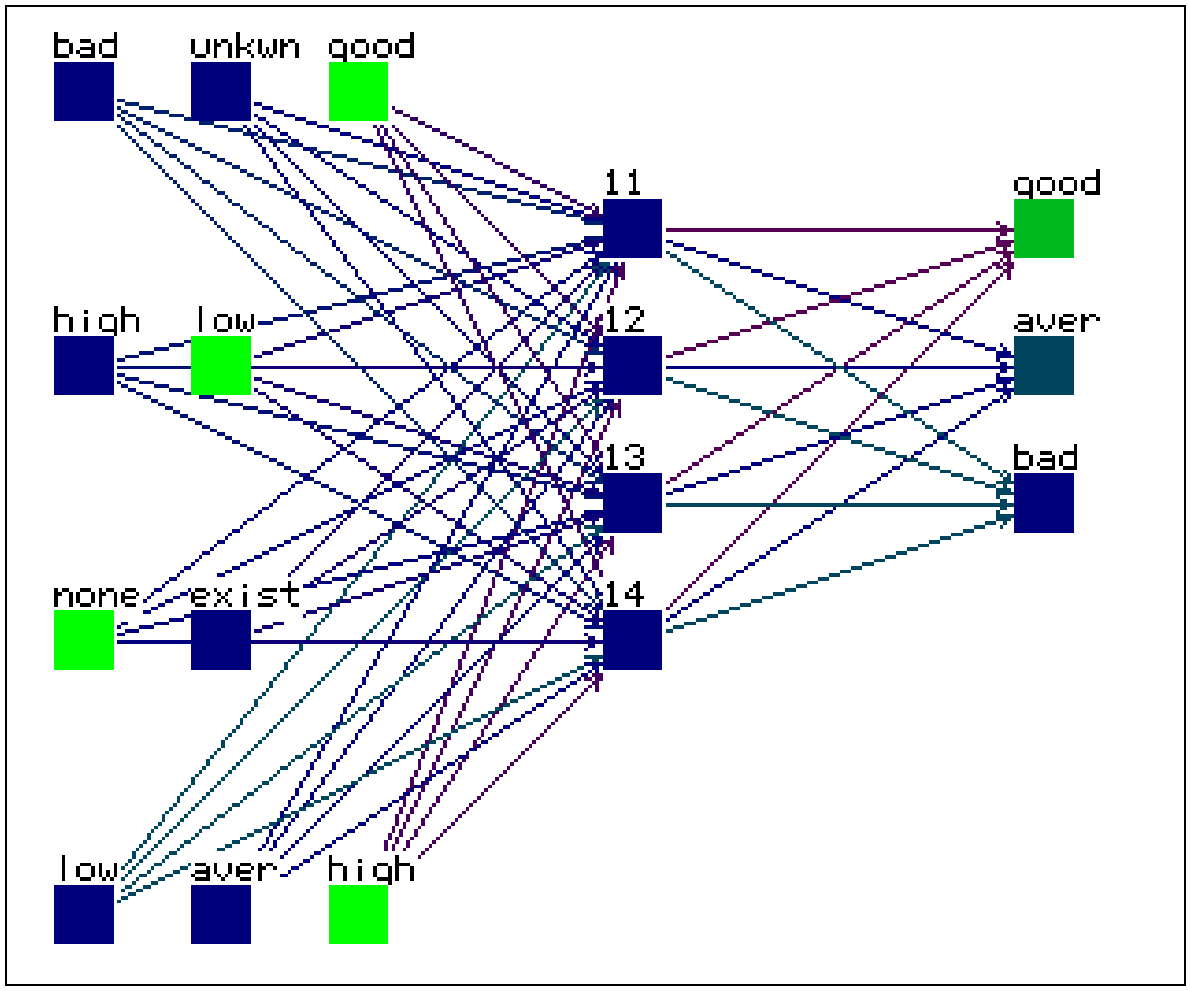
Ihr Aufbau ist dergestalt, dass die Neuronen in Schichten zusammengefasst sind und die Verbindungen eines Neurons nur zu Neuronen einer nachgeordneten Schicht führen. Es sind somit vorwärts gerichtete neuronale Netze, die mit einem Backpropagation-Lernverfahren trainiert werden.

Ein einfaches Beispiel geht auf [Lackes98] zurück. Eine ausführliche Darstellung und Diskussion verschiedener Merkmalscodierungen für das Beispiel sind [LämmelCleve2001] zu entnehmen. Stark vereinfacht wird anhand einiger weniger Daten eines Bankkunden

- Kreditgeschichte: bad, unknown, good
- Bisherige Schulden: low, high
- Sicherheiten: none, exists
- Einkommen: low, average, high

eine Klassifizierung vorgenommen: schlechter, normaler, guter Kunde.

Abbildung 8: Vorwärts gerichtetes neuronales Netz für die Klassifikation von Bankkunden



Die Menge der Beispieldaten wird unterteilt in eine **Trainingsmenge** und eine **Testmenge**. Erstere wird zum Lernen benutzt, letztere dient zur Ermittlung der Güte des trainierten Netzes. Die Schwierigkeit beim Trainieren eines solchen Netzes besteht darin, eine Balance zwischen der gewünschten Generalisierungsfähigkeit und einem möglichst kleinen Trainingsfehler zu erzielen. Auch hier sei auf [LämmelCleve2001] für weitere Betrachtungen verwiesen.

Nach dem Anlegen einer Eingabe wird die Aktivierung der Neuronen schichten weise neu berechnet. Unter Verwendung der logistischen Funktion erhält man folgende Formeln:

$$act_i = \frac{1}{1 + e^{-c \cdot (net_i - \theta_i)}} \qquad net_i = \sum_j w_{ji} \cdot o_j$$

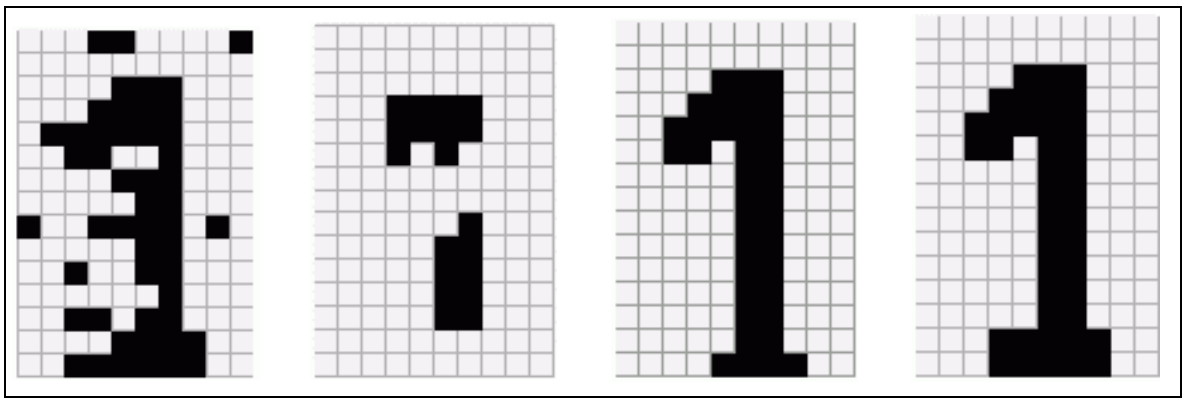
Die Aktivierung der Ausgabe-Neuronen stellt dann das Verarbeitungsergebnis dar.

3.3.2. Hopfield-Netz

Ein Hopfield-Netz wird nicht im eigentlichen Sinne trainiert, sondern die Gewichte werden aus Speichermustern heraus berechnet. Ein Hopfield-Netz kann dann dazu verwendet werden, Eingaben zu klassifizieren: Entweder ist die Eingabe eine (verrauschte) Form eines Speicher-Musters und wird als solche erkannt, oder die Eingabe ist keinem Speicher-Muster ähnlich.

Ein Hopfield-Netz besteht aus einer einzigen Schicht. Diese besteht aus einer Menge von Neuronen, wobei jedes Neuron eine Verbindung zu jedem anderen Neuron besitzt. Ein Hopfield-Netz wird meist zweidimensional dargestellt (siehe Abbildung).

Abbildung 9: Wiederherstellung eines verrauschten Musters in einem Hopfield-Netz



Die Abbildung zeigt die Wiedererkennung eines Musters. Das Hopfield-Netz besteht aus 10x15 Neuronen. Die Eingabe ist verrauscht, das System erkennt in drei Schritten das Muster. Das Muster der Zahl 1 ist eines von mehreren Mustern, welches dem Netz vorher eingeprägt wurde.

$$w_{ij} = w_{ji} = \begin{cases} \sum_p m_{pi} \cdot m_{pj} & , i \neq j \\ 0 & , i = j \end{cases}$$

Die Gewichte eines Hopfield-Netzes berechnen sich nach der obigen Formel. Dabei ist m_{pi} das i.-Pixel des Musters p. Schwarz wird hier als 1 und weiß als 0 gewertet. Häufig jedoch führt die Verwendung von (1,-1) zu besseren Resultaten.

Die neue Aktivierung eines Neurons wird mittels einer Schwellwert-Funktion berechnet:

$$act_i(t+1) = \begin{cases} 1, & falls\ net_i > \theta \\ 0, & falls\ net_i < \theta \\ a_i(t), & falls\ net_i = \theta \end{cases}$$

Ein derartiges Hopfield-Netz kann nun nicht nur zur Wiedererkennung normaler visueller Bilder oder Zeichen verwendet werden. Von Sosna (siehe [LämmelPrauseSosna2002] wird vorgeschlagen, ein derartiges Netz mit Unternehmensdaten zu konfrontieren. Die schrittweise „Wiedererkennung“ bildet die Eingabe in einer unterschiedlichen Zahl von Schritten auf ein Referenzbild ab. Siehe hierzu Abschnitt 4.3 .

3.4. Clusterung

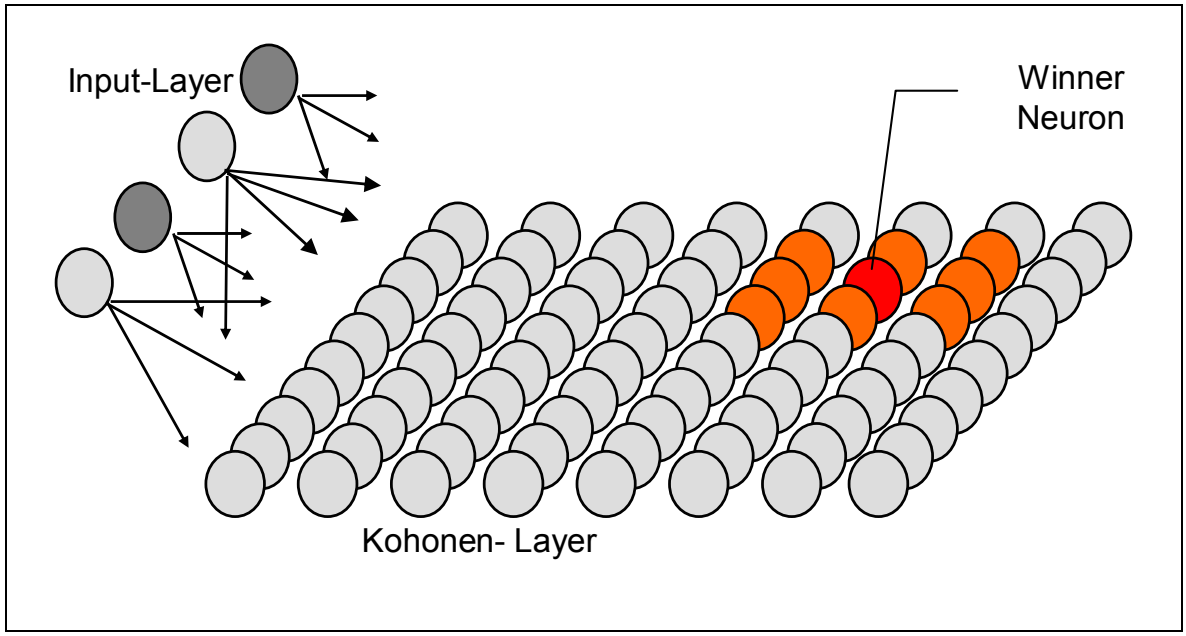
Von einer Clusterung oder Cluster-Analyse wird gesprochen, falls noch keine Klassen-Einteilung bekannt ist. Die Klassen-Einteilung wird erst durch das Data-Mining-Verfahren herausgearbeitet. Die Menge der Eingaben wird in Cluster gruppiert. Die Bedeutung dieser Gruppen muss dann durch geeignete Interpretation herausgefunden werden.

3.4.1. Selbstorganisierende Karten

Mittlerweile ist das Konzept der selbstorganisierenden Karte oder Kohonen-Karte in viele Data-Mining-Software-Produkte integriert. Die Fähigkeit aus einer Menge von Eingabedaten Cluster bilden zu können, sind für die Daten-Analyse sehr interessant.

Eine selbstorganisierende Karte besteht aus zwei Schichten, der Eingabe-Schicht sowie der Karten- oder Kohonen-Schicht. Dabei ist jedes Neuron der Eingabe-Schicht mit jedem Neuron der Karten-Schicht verbunden.

Abbildung 10: Aufbau einer selbstorganisierenden Karte



Das Gewinner-Neuron wird anhand des euklidischen Abstands zwischen eingehenden Gewichtsvektor und Eingabe-Vektor berechnet (vgl. Abschnitt 2.4.2). Eine Aktivierung der Neuronen im ursprünglichen Sinne gibt es nicht. Betrachtet man den euklidischen Abstand eines Neurons zur Eingabe als eine Art Aktivierung, so lässt sich ein Gewinner-Neuron auch aus der graphischen Darstellung des Netzes erkennen, z. B. durch eine entsprechende Farbe.

Die Beispiele werden dem Netz wiederholt aber in zufälliger Reihenfolge präsentiert. Nach der Bestimmung des Gewinner-Neurons z werden alle Gewichte w_{iz} der Verbindungen von der Eingabe-Schicht zum Neuron z angepasst. Darüber hinaus werden auch alle Neuronen in der Nachbarschaft, konkret deren Verbindungsgewichte, verändert:

$$w_{ij}^{/} = \begin{cases} w_{ij} + \eta \cdot h_{jz} \cdot (m_i - w_{ij}) & ,if \quad dist(j,z) \le r \\ w_{ij} & ,otherwise \end{cases}$$

Die Formel besagt, dass nur die Gewichte w_{ij} verändert werden, die zu einer Verbindung von einem Eingabe-Neuron i zu einem Karten-Neuron j, welches in der Umgebung von z liegt, ($dist(j,z) \le r$).

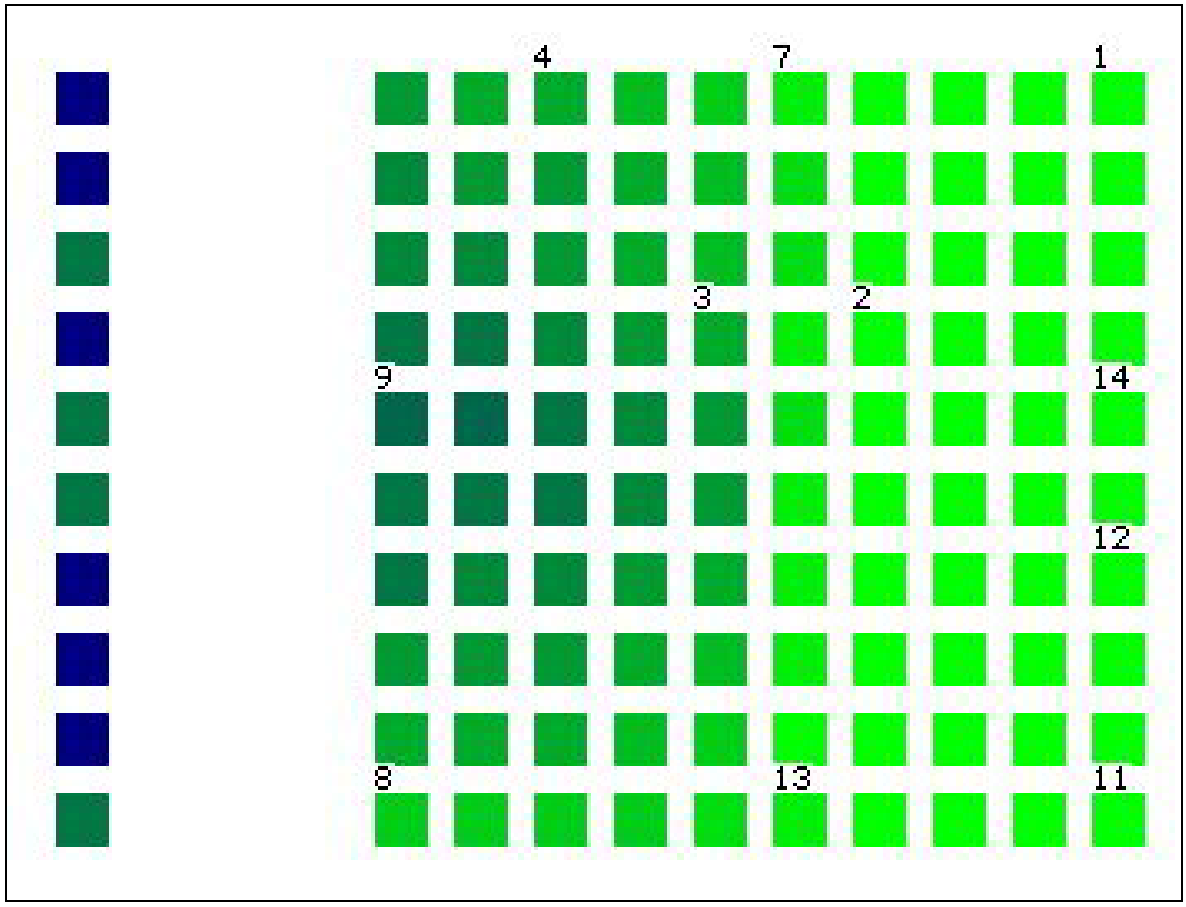
Die Änderung erfolgt auch in Abhängigkeit der Nähe von j zu z. Dies wird durch den Faktor hjz ausgedrückt, der sich aus der Gauß'schen Glockenkurve ergibt:

$$h_{jz} = e^{-\frac{dist(j,z)^2}{2 \cdot r^2}}$$

Der Lernfaktor η sowie der Radius r werden anfangs recht groß gewählt, damit die gesamte Karte in die Anpassung einbezogen wird. Im Laufe des Anpassungsprozesses werden diese Parameter verkleinert, so dass der Prozess der Cluster-Bildung zum Stillstand kommt.

Die selbstorganisierende Karte für das Bankkunden-Beispiel aus dem Abschnitt 2.5.1 zeigt Abbildung 4, wobei aus Gründen der Übersichtlichkeit auf die Darstellung der Verbindungen verzichtet wird.

Abbildung 11: Kohonenkarte für das Bankkunden-Beispiel



Die Zahlen an den Neuronen zeigen an, für welche Eingabe das Neuron das Gewinner-Neuron ist. Im konkreten Fall ist Muster 9 angelegt worden. Anhand der Färbung erkennt man das Erregungszentrum um das Neuron mit der Markierung 9 herum.

Die Schwierigkeiten bei der Anwendung einer selbstorganisierenden Karte liegen insbesondere auch in der Interpretation der Verteilung. Hierzu gibt dieser Ansatz wenig Hilfestellung. Anhand der Daten, die in ähnliche Kartenbereiche abgebildet wurden, sind dann Untersuchungen notwendig, die diese Ähnlichkeit der entsprechenden Datensätze inhaltlich begründen.

3.4.2. ART-Netze

ART, Adaptive Resonanz Theorie, stellt einen Versuch dar, Stabilität und Plasitizität eines Netzes zu erhöhen. Ein Netz bezeichnet man als stabil, wenn es einmal gelernte Muster durch das Lernen neuer Muster nicht wieder verlernt. Ein Backpropagation-Netz ist z. B. nicht stabil. Unter Plastizität wird die Fähigkeit verstanden, selbstständig neue Muster zu lernen.

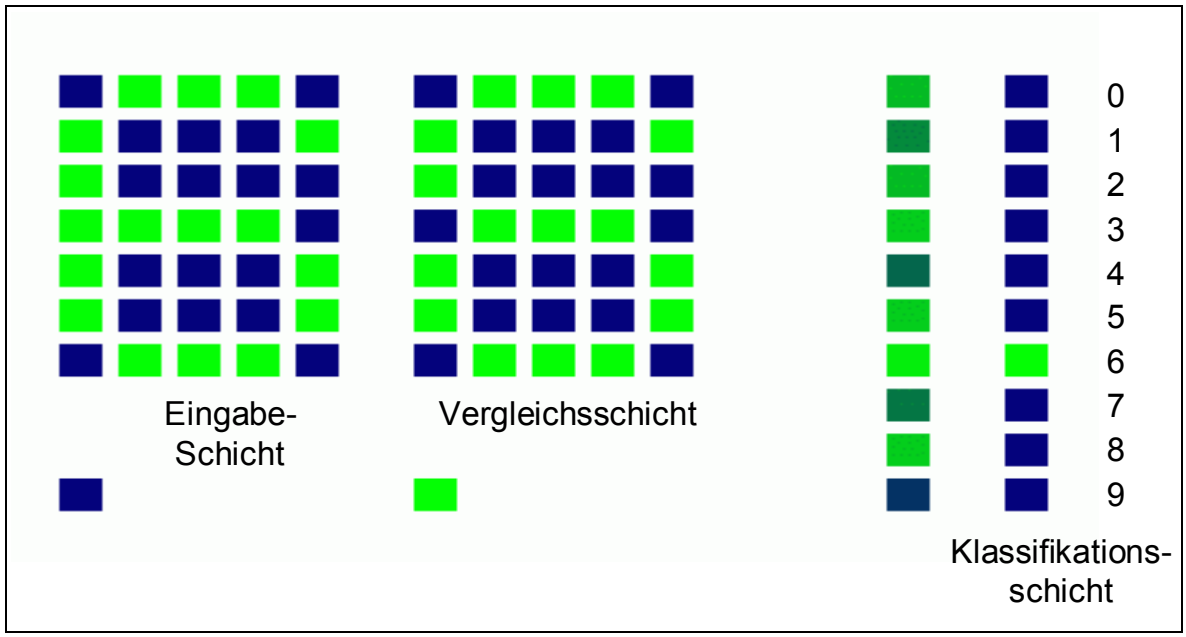
Das Prinzip eines ART-Netzes mit binärer Eingabe lässt sich leicht beschreiben, die Abläufe im einzelnen sind nicht durch wenige Formeln zu beschreiben. Für eine ausführlichere Darstellung sei auf [Zell97] verwiesen.

Ein ART-Netz besteht im Kern aus drei Neuronen-Schichten, einer Eingabe-, einer Vergleichs- und einer Klassifikationsschicht. Für eine Eingabe wird ein Gewinner-Neuron in der Klassifikationsschicht ermittelt. Dieses Neuron ist Vertreter einer Klasse von Mustern, für die ein Referenzmuster in der Vergleichsschicht abgelegt ist. Ist nun die Eingabe dem der Klasse zugeordneten Vergleichsmuster hinreichend ähnlich, wird die Eingabe als zu dieser Klasse zugehörig klassifiziert. Darüber hinaus wird das Referenzmuster etwas verändert, so dass es auch für das neue Muster noch besser als Vertreter gelten kann.

Ist die Eingabe dem Referenzmuster nicht ähnlich, wird ein neues Gewinner-Neuron in der Klassifikationsschicht bestimmt und analog verfahren. Ist die Eingabe zu keinem der bisherig gespeicherten Referenzmuster hinreichend ähnlich, so wird die Eingabe als Referenzmuster einer Klasse gespeichert und ein bisher nicht benutztes Neuron der Klassifikationsschicht als Vertreter der Klasse festgelegt.

Die gewünschte Ähnlichkeit kann vom Benutzer als Parameter festgelegt werden.

Abbildung 12: Ein ART-Netz für die Erkennung von Ziffern



Im Unterschied zu einer selbstorganisierenden Karte kann der Benutzer eines ART-Netzes die Klassifikation oder besser Clusterung besser steuern. Die maximale Zahl von Klassen kann durch die Größe der Klassifikationsschicht vorgegeben und die Gruppierung durch die Wahl des Ähnlichkeitsparameters direkt beeinflusst werden.

Da die Ähnlichkeit binärer Vektoren sich meist auf die Anzahl unterschiedlicher 0-1-Belegungen im Verhältnis zur Länge des Vektors bezieht, werden alle Eingabe-Positionen gleich stark berücksichtigt. Sind unterschiedliche Bedeutungen der Eingabepositionen zu berücksichtigen, so ist dies durch eine entsprechende Kodierung zu realisieren. Der Datenvorverarbeitung kommt somit eine besondere Bedeutung zu.

3.4.3. Neuronale Gase

Neuronale Gase sind mit den selbstorganisierenden Karten vergleichbar. Das Konzept geht auf Fritzke zurück, siehe [Fritzke98]. Neuronale Gase bestehen ebenso aus zwei Schichten, einer Eingabe- und einer Verarbeitungsschicht. Die Verarbeitungsschicht besteht dabei aus einer Menge von unvernetzten Neuronen. Die Anzahl der Neuronen kann fest oder variabel (wachsende neuronale Gase) sein.

Im Unterschied zu einer selbstorganisierenden Karte werden nicht die Neuronen in einer räumlichen Nachbarschaft zum Gewinner-Neuron mit verändert, sondern die Adaption betrifft das Gewinner-Neuron sowie die gemäß

Ordnungskriterium nächsten Neuronen, der „zweite Gewinner“, „dritte Gewinner“ usw.

Der Einsatz neuronaler Gase für die Klassifikation scheint recht vielversprechend zu sein, da damit auch Clusterungen erzeugt werden können, die sich nicht im Zweidimensionalen (wie bei einer selbstorganisierenden Karte) als zusammenhängendes Gebiet darstellen lassen.

4. Anwendungen

Die Anwendungen neuronaler Netze im Data-Mining sind in allen Bereichen des Data-Mining zu finden. In diesem Kapitel werden zum einen Anwendungsfälle aus der Literatur angeführt und zum anderen werden die Beispiele näher erläutert, die konkret untersucht wurden.

Bisher reduziert sich der Einsatz neuronaler Netze im Data-Mining auf die Anwendung selbstorganisierender Karten. Einen guten Überblick dazu liefert [Vesanto2000].

Schon klassisch zu nennende Anwendungsbeispiele betreffen die Analyse von Kundenverhalten. In einer Zusammenstellung von Wiedmann und Buckler [WiedmannBuckler2001] stehen diese Anwendungen im Mittelpunkt. Weitere Beispiele betreffen Zeitreihenanalyse und Einkommensschätzungen.

Im Rahmen der Arbeiten zum Projekt „Wissensextraktion aus Proteom-Daten mit Hilfe neuronaler Netze“ wurden biomedizinische Anwendungen ausgewertet. Eine Reihe von Arbeiten nutzen selbstorganisierende Karten für die Auswertung der bei Genchip-Experimenten gewonnenen Daten.

4.1. Analyse einer Umfrage zum Verkehrsverhalten Wismarer Studenten

Bereits vor ca. zwei Jahren wurde von Studenten des Studiengangs Sozialverwaltung / Management sozialer Dienstleistungen eine Umfrage zum Verkehrsverhalten der Wismarer Studenten durchgeführt, bisher jedoch noch nicht ausgewertet.

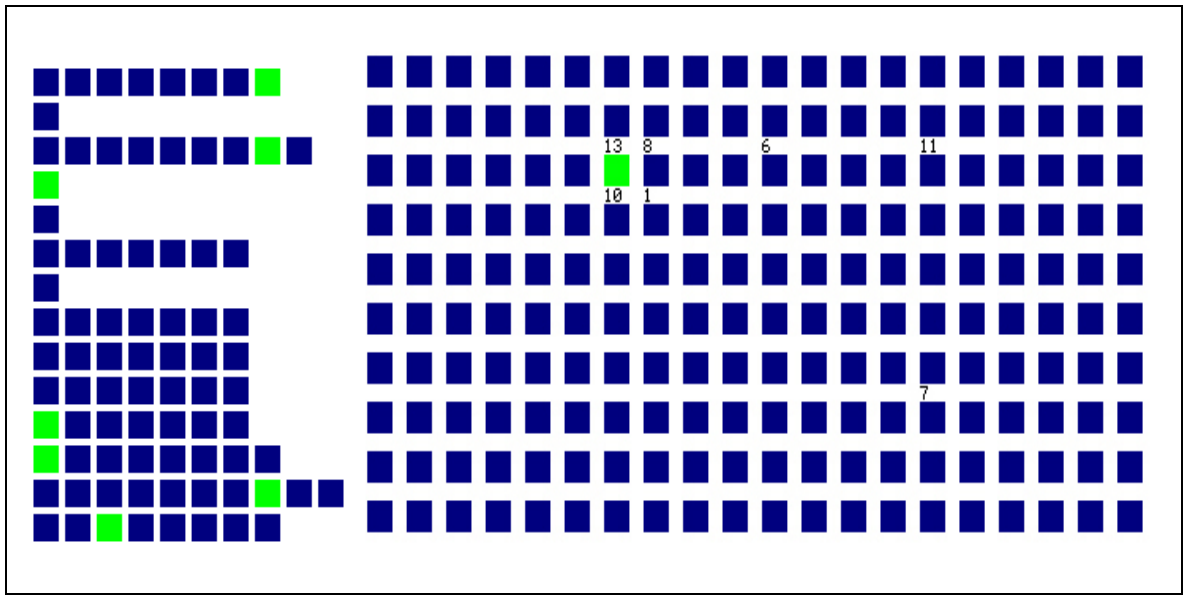
Ein typischer Data-Mining Fall: Es werden Daten erhoben, die Art der Auswertung ist zum Zeitpunkt der Datenerhebung noch unbekannt. Im Rahmen einer studentischen Arbeit wurde versucht, mittels neuronaler Netze zu Erkenntnissen zu gelangen. Auch hier zeigte sich, dass die Datenvorverarbeitung anfangs unterschätzt wurde und letztendlich einen großen Raum beanspruchte.

In der Umfrage wurden 16 Fragen gestellt. Die ersten Fragen beziehen sich auf das Alter, das Geschlecht, den Studiengang, das aktuelle Semester und die Anzahl der Urlaubssemester. Darauf folgt die Frage nach einem Hauptwohnsitz in Wismar. Wenn man in Wismar den Hauptwohnsitz hat, folgte die Frage, wie man von dort zur Hochschule kommt. Ist der Hauptwohnsitz nicht in Wismar, wird nach einem Zweitwohnsitz in Wismar gefragt. Ist der Zweit-

wohnsitz in Wismar, wird gefragt, wie häufig zum Hauptwohnsitz gefahren wird, wie man vom Hauptwohnsitz zur Hochschule kommt und wie man vom Hauptwohnsitz zum Zweitwohnsitz kommt.

Die letzten vier Fragen behandeln die Nutzung von Verkehrsmitteln bei Entfernung über 80 km, einen möglichen Umstieg auf alternative Verkehrsmittel sowie mögliche alternative Verkehrsmittel. Es stehen die Antworten von 211 befragten Studenten zur Verfügung.

Abbildung 13: Selbstorganisierende Karte mit Gewinner-Neuronen für die Umfrage-Daten



Mit den gegebenen Daten ist leider kein interpretierbares Ergebnis mittels einer selbstorganisierenden Karte unter Nutzung des SNNS (siehe Kapitel 5) erzielt worden [EwertWeiß2002]. Tests mit dem Intelligent Miner [IBM] zeigten Abhängigkeiten in den Daten auf. So sollte die gesamte Umfrage beispielsweise nach dem Haupt- bzw. Zweitwohnsitz getrennt werden. Damit sollten bessere Ergebnisse möglich sein. Und wieder ist dies eine Aufgabe der Datenvorverarbeitung. Eine bessere Gestaltung der Datenerhebung könnte natürlich von vornherein nicht notwendige Abhängigkeiten in den Fragen nach Möglichkeit ausschließen.

4.2. Risiko-Analyse in einer Oberfinanzdirektion

Diese Anwendung wird seit Anfang 2002 in Zusammenarbeit mit einer Oberfinanzdirektion entwickelt. Die Problemstellung und Ansätze zur Lösung sind in [LämmelPrauseSosna2002] dargelegt. In diesem Abschnitt werden einige durchgeführte Experimente diskutiert.

In Deutschland fallen regelmäßig Tausende von Umsatzsteuer-Voranmeldungen an. Diese können im monatlichen, vierteljährlichen oder jährlichen Rhythmus von den Firmen eingereicht werden. Eine Aufgabe der Finanzämter besteht auch darin, Steuerbetrug aufzudecken. Die geringe Zahl der Inspektoren erlaubt es nicht, jedes Unternehmen regelmäßig einer Steuerprüfung zu unterziehen.

Die Fragestellung lautet daher:

- Können aus den Daten der Umsatzsteuer-Voranmeldungen potentielle Unregelmäßigkeiten erkannt werden, damit die Inspektoren vordergründig die Unternehmen kontrollieren, in denen Steuerbetrug auftritt?

Damit sollen zwei Risiken minimiert werden:

- Ein Unternehmen wird kontrolliert, obwohl es steuerlich korrekt arbeitet.
- Ein Unternehmen wird nicht kontrolliert und der Steuerbetrug bleibt unentdeckt.

Der Umfang des Mehrwertsteuer-Betrugs hat in den vergangenen Jahren erheblich zugenommen und wird derzeit auf jährlich 12 Milliarden EUR geschätzt. „In einschlägigen Kreisen hat es sich herumgesprochen, dass es wesentlich einfacher und mit wesentlich weniger Risiko verbunden ist, sich beim Finanzamt durch Vorsteuerbetrug Geld zu beschaffen, als eine Bank zu überfallen.“ [Zeit15/2002]

So bedeutsam es ist, hier Fortschritte in der Aufklärung zu erzielen, so schwierig ist es, aus den reinen Formular-Werten einer Umsatzsteuer-Erklärung die notwendigen Informationen abzuleiten. Die Daten wurden von der Oberfinanzdirektion Rostock zur Verfügung gestellt, sie wurden anonymisiert und mittels einer Zufallsgröße verfremdet. Derzeit werden mehrere Ansätze verfolgt.

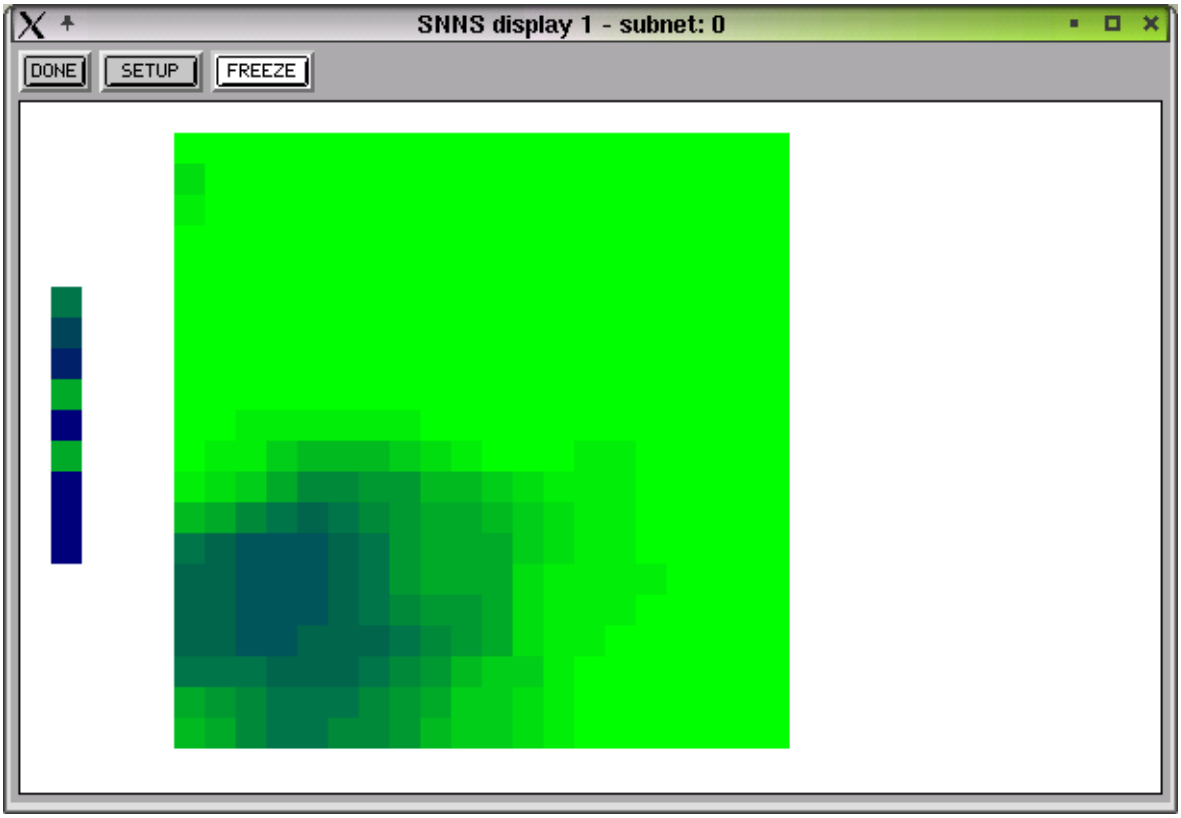
Die Clusterung der Daten unter Nutzung der Unsatzsteuer-Voranmeldungen. Dabei stehen relativ wenige Daten zur Verfügung. Neben der Gewerbekennziffer, der Rechtsform sowie einer Betriebsgröße (groß, mittel, klein, kleinst) sind dies die Angaben zu den Steuerarten. Es wurde untersucht, ob sich Assoziationen zwischen diesen Merkmalen herstellen lassen. Benutzt wurden selbstorganisierende Karten. Verschiedene Kodierungen und Normierungen sind notwendig. Zusätzliche Angaben, die Verhältnisse zwischen den Umsätzen der verschiedenen Steuerarten.

Die bisher erzielten Ergebnisse können noch nicht befriedigen. Es zeichnet sich aber ab, dass aus den Umsatzangaben alleine, kaum Hinweise auf Steuerbetrug ableitbar wären. Sinnvoll erscheint eine Verbindung zwischen Umsatzangaben und weiteren Informationen.

Es sollte das Wissen der Behörde bei der Auswahl der Betriebsprüfungen einbezogen werden. Aus einer Menge von Datensätzen, für die das Ergebnis

einer Betriebsprüfung bekannt ist, könnte man dann mit Hilfe von überwachten Lernverfahren ein vorwärts gerichtetes neuronales Netz trainieren.

Abbildung 14: Erregungszentrum für einen Datensatz der Umsatzsteuer-Voranmeldung



Ein etwas ungewöhnlicher Ansatz für ein Data-Mining-Problem wird von Sosna verfolgt, siehe [LämmelPrauseSosna2002]. Dabei verwischen die Grenzen zwischen Data-Mining und wissensbasierter Verarbeitung. Sogenannte weiche Indikatoren, wie z. B. Alter der Firma, Alter des Geschäftsführers, Sitz der Firma, usw., werden mit erfasst und in Form von Fuzzy-Werten in die Verarbeitung mit einbezogen. Kern ist ein einem Hopfield-Netz ähnliches Netz, dass dann zu einem Muster konvergiert, aus dem sich eine Aussage über das Risiko dieser Firma ableiten lässt.

In weiteren Experimenten werden beide Ansätze weiter verfolgt und auch auf andere Probleme der Steuerfahndung ausgedehnt.

4.3. Auswertung von Genexpressionsdaten[2]

Mit Hilfe der Genexpressionstechnik ist es seit einigen Jahren möglich, die Aktivitäten sämtlicher bekannter Proteine in einer Zelle gleichzeitig zu messen, indem das Expressionslevel der entsprechenden tRNA mittels eines Chips, auf dem an definierten Stellen die komplementären Oligonucleotide aufgebracht sind, gemessen wird. Pro Chip kann die Aktivität von ca. 10.000 Genen unter Verwendung von 24.000 verschiedenen Gen-Teilsequenzen gemessen werden. Zusammen mit der bekannten Nukleotid-Sequenz der Gene steht eine riesige Menge an Informationen zur Verfügung, deren Potential genutzt werden will.

Aufgrund des Umfangs und der Komplexität der bei der Genexpression anfallenden Daten ist eine manuelle Analyse nicht möglich. Allenfalls eine oberflächliche Betrachtung kann vorgenommen werden. Das informative Potential der Daten wird bei weitem nicht ausgeschöpft. Gerade unter Berücksichtigung der erheblichen Kosten, die für die Gewinnung dieser Daten anfallen, ist eine derart ineffiziente Nutzung eine Verschwendung von Ressourcen. Hier ist eine computerunterstütze Analyse der Daten notwendig, um die möglichen Ergebnisse in Relation zu den Kosten zu bringen [Khan2001], [Quakenbush2001].

Zum aktuellen Zeitpunkt verfügbare Software zur Analyse von Genexpressionsdaten (z. B. BioScout [LionBioscience] ist sowohl sehr teuer als auch für den Anwender unzureichend. Es werden in der Regel eine Auswahl von Standard-Cluster-Algorithmen und teilweise selbstorganisierende Karten als Vertreter der künstlichen neuronalen Netze angeboten. Jedoch ist hier ein Mangel an Transparenz zu beanstanden, der eine korrekte Interpretation der Ergebnisse fraglich macht. Die Komplexität der Daten erfordert zudem eine Berücksichtigung von mehr als nur zwei oder drei Merkmalen, wie es jedoch häufig der Fall ist. Es gibt kaum Unterstützung zur Klassifizierung mehrdimensionaler Daten.

Hier stellen die künstlichen Neuronalen Netze eine sinnvolle Ergänzung zu den klassischen Verfahren dar. Ihre Fähigkeit zur Abstraktion sowie die Verarbeitung hochdimensionaler Eingabevektoren machen sie für die Analyse von komplexen Daten besonders geeignet.

Anwendung finden können sowohl überwachte Lernverfahren zur Analyse von unterscheidungsrelevanten Merkmalen bei bekannter Klassifikation von Expressionsmustern wie dies z. B. bei Melanomen der Fall ist, als auch unüberwachte Lernverfahren, um bisher unbekannte Korrelationen zwischen Genen zu finden, sogenannte regulative Elemente.

---

[2] Die Ausführungen in diesem Abschnitt entstammen der Mitarbeit im Landesforschungsprojekt „Genomorientierte Biotechnologie“, konkret von Stefan Wissuwa, Projektmitarbeiter.

Anwendungsmöglichkeiten sind zum einen die Unterstützung der Diagnose von Krankheiten. Hier können Neuronale Netz eingesetzt werden, um z. B. den Unterschied zwischen Melanom-Zellen und normalen Hautzellen zu erlernen. Sie können anschließend zur Klassifizierung undiagnostizierter Zellen verwendet werden. Selbstorganisierende Karten können dazu dienen, bisher unbekannte Zusammenhänge zwischen verschiedenen Genen durch Cluster-Analyse zu finden, wie dies anhand von Mauslinien vorgenommen werden wird, um die regulativen Elemente bei Adipositas zu lokalisieren.

Ein Problem bei der Anwendung Neuronaler Netze stellt die Extraktion von Wissen darüber dar, aus welchem Grund ein Netz eine bestimmte Klassifikation vorgenommen hat. Hier existieren nur sehr wenige Ansätze, wie eine Wissensextraktion vorgenommen werden kann. Dementsprechend wichtig ist die Entwicklung einer entsprechenden Nutzerschnittstelle, um es dem Experten zu ermöglichen, mit Hilfe seines Wissens und Wissensbasen, wie etwa Genomdatenbanken, verwertbare Ergebnisse zu erhalten.

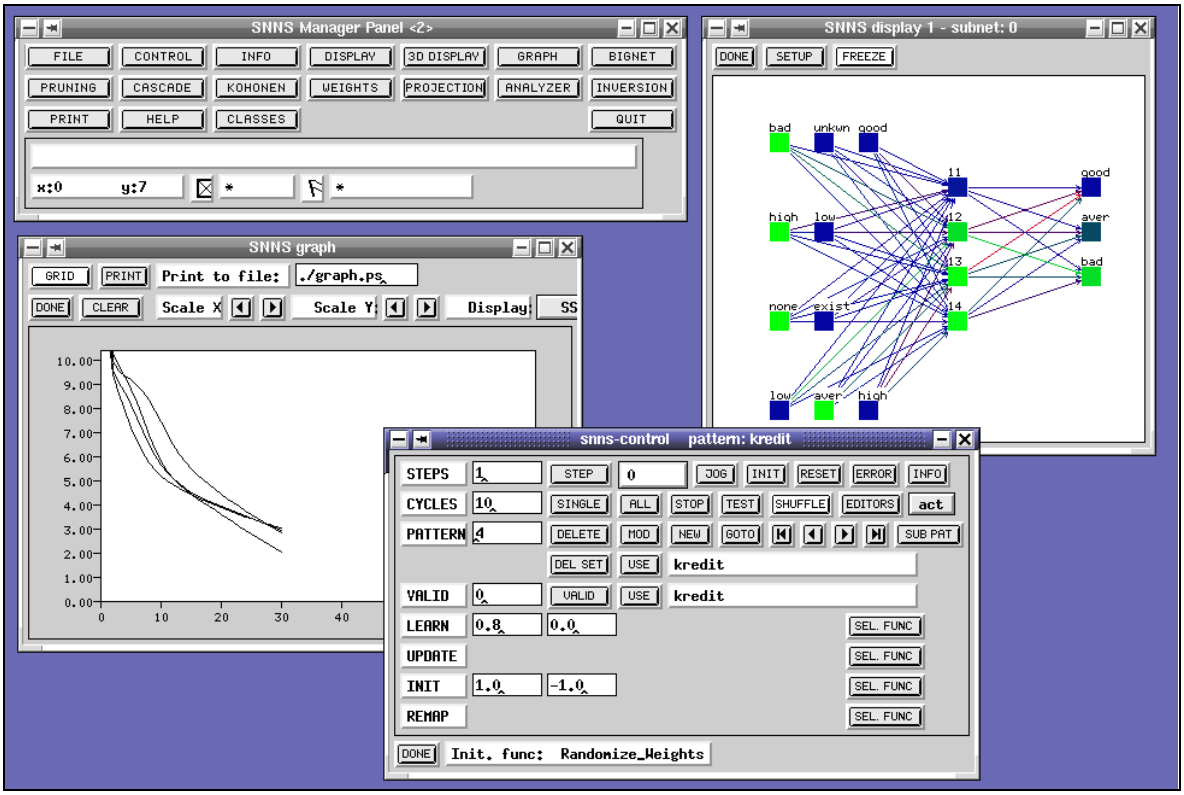
Einen weiteren Schwerpunkt der Untersuchung bildet die Datenvorverarbeitung, da die Kodierung der zu untersuchenden Daten in eine durch Neuronale Netze verwertbare Form einen großen Einfluss auf das Ergebnis hat. Auch hier existieren keine generellen Regeln für die Anwendung bestimmter Kodierungsformen, so dass diese erst durch die Erfahrungen im weiteren Projektverlauf – wenigstens für die Domäne der Genexpressionsanalyse - erlangt werden können.

5. Werkzeuge für den Einsatz von neuronalen Netzen

Eine Übersicht über verfügbare Software für die Entwicklung neuronaler Netze ist dem FAQ-Dokument der Diskussionsgruppe comp.ai.neuralnets zu entnehmen [FAQNN2002]. Die in dieser Arbeit vorgestellten Untersuchungen wurden bisher alle mit dem **Stuttgarter Neuronale Netze Simulator (SNNS)** durchgeführt. Dieser kann für nichtkommerzielle Zwecke kostenlos genutzt werden.

Ursprünglich für UNIX-Betriebssysteme entwickelt, kann der SNNS auch auf Windows-Rechnern eingesetzt werden.

Abbildung 15: Die SNNS-Benutzungsoberfläche

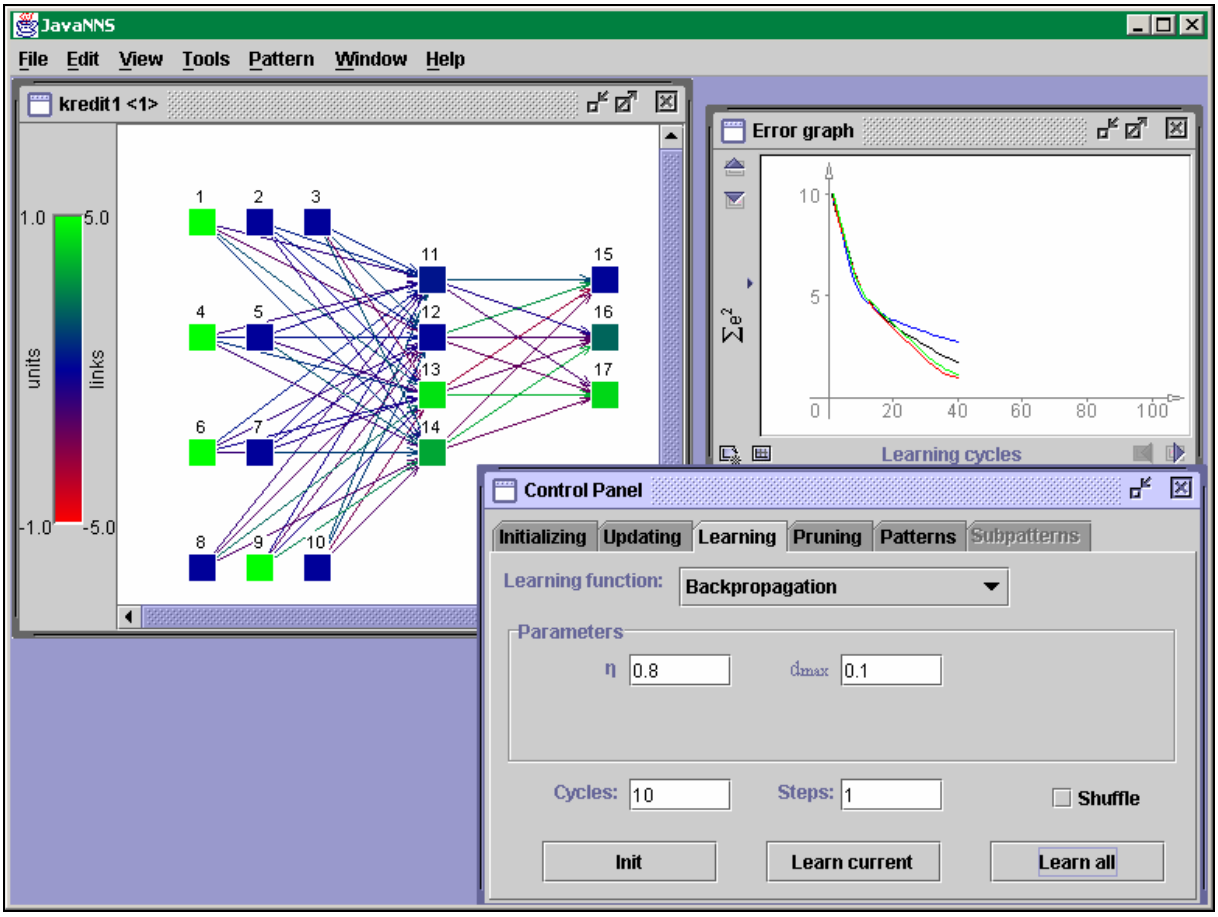


Mit dem SNNS können sehr viele Netzarchitekturen bearbeitet werden. Zudem steht eine Vielzahl unterschiedlicher Lernalgorithmen für die Experimente zur Verfügung.

Insbesondere für die Arbeit mit mehrschichtigen vorwärts verketteten neuronalen Netzen sowie selbstorganisierenden Karten eignet sich die Software **JavaNNS** in besonderer Weise. Unter Nutzung der SNNS-Bibliothek wurde ein neues graphisches Nutzer-Interface in Java entwickelt. Diese neue Benutzungsoberfläche orientiert sich am allgemein üblichen Fenster-Aufbau und ist somit wesentlich leichter zu bedienen als SNNS-Nutzer-Interface.

In Data-Mining-Software ist häufig auch die Möglichkeit der Anwendung einer selbstorganisierenden Karte mit enthalten. Stellvertretend seien hier genannt der Intelligent Miner [IBM], BioScout zur Auswertung von Genexpressionsdaten [LionBioscience] oder das System Weka ([WittenFrank2000])

Abbildung 16: Die JavaNNS Benutzungsoberfläche



6. Ausblick

Neuronale Netze stellen ein prinzipiell geeignetes Werkzeug für das Data-Mining dar. Um für den jeweiligen Anwendungsfall das geeignete Analyse-Verfahren auswählen zu können, sind derzeit viele Experimente notwendig. Erst dadurch lassen sich realistische Vergleiche vornehmen.

Wünschenswert wäre Wissen über die Gestaltung der Wissensextraktion und zwar über alle vier Etappen hinweg. Derartiges Wissen ist absolut notwendig, um aus dem Stadium einer experimentellen Herangehensweise zu einem ingenieurmäßigen Vorgehen zu gelangen. Unter einem ingenieurmäßigen Vorgehen im Bereich Data-Mining, also einem Data-Mining-Engineering, wird verstanden, dass für den jeweiligen Anwendungsfall die Techniken und Parameter der Vorverarbeitung, Mustererkennung und Nachverarbeitung von vornherein so gewählt werden können, das eine Wissensextraktion gesichert ist.

Um zu einen derartigen Vorgehen zu gelangen, sind noch viele Experimente durchzuführen, um daraus allgemeingültige Aussagen ableiten zu können. Für die effiziente Durchführung der Experimente sind geeignete Software-Umge-

bungen erforderlich, die die Techniken bereitstellen und Möglichkeiten der Kombination der Techniken und des Datenaustausches bieten.

Das so erarbeitete Wissen über die Wissensextraktion (Metawissen) kann dann in die Gestaltung von Anwendungsszenarien einfließen. Dies ist eine Zielvorstellung, die noch einen erheblichen Forschungsaufwand erfordert.

Interessant ist auch hier wieder der Ansatz, dass Verfahren, die „unscharf" arbeiten, häufig ein besseres Ergebnis erzielen. Auch im Data-Mining muss man damit leben, dass nicht alle Daten 100%ig richtig klassifiziert werden.

6.1. Offene Fragen

Aus den bisher durchgeführten Experimenten wird deutlich, dass die besonderen Schwierigkeiten im Data-Mining nicht in der Anwendung der Techniken in der Phase der Mustererkennung liegen. Ein mehrfaches an Aufwand erfordert die Vorverarbeitung, Codierung und Transformation, der Daten. Analoges gilt für die Nachbereitung: Visualisierung und Interpretation.

Die Probleme lassen sich in eine Frage komprimieren:

Wie lässt sich aus der Problemstellung erkennen, welches Verfahren mit welchen Parametern anzuwenden ist?

Aus dieser Fragestellung kann eine komplexe Aufgabenstellung abgeleitet werden: Die Antworten auf die oben gestellte Frage sind explizit als Wissen darzustellen. Damit würden unterschiedliche Techniken der Wissensrepräsentation, implizite Wissensdarstellung, z. B. in neuronalen Netzen, mit expliziter Wissensdarstellung, z. B. Regeln zum Einsatz von Data-Mining-Verfahren, verknüpft. Aufbauend auf dieses Wissen kann dann ein intelligentes Data-Mining-System entwickelt werden, dass dann den Benutzer entsprechend der Aufgabenstellung bei der Vorgehensweise unterstützt. Ein derartiges Data-Mining-Engineering-System könnte ähnlich wie auf dem Gebiet des Software-Engineering Hilfen bei der Entwicklung von Data-Mining-Lösungen anbieten. Das Konzept eines solchen Systems wurde in dem Projekt-Antrag „DaMEn-Data-Mining Engineering mit Anwendungen in Biotechnologie und Financial Engineering" [CleveLämmel2002] als Beitrag zum Antrag Mobile Landesforschungsschwerpunkt IuK im August 2002 dargestellt.

Speziell bezogen auf die Anwendung neuronaler Netze für das Data-Mining sind zukünftig auch andere Verfahren als das vorwärts gerichtete neuronale Netz für die Klassifikation bzw. die selbstorganisierende Karte für die Cluster-Analyse zu betrachten. Hier werden ART-Netze und neuronale Gase als einsetzbare Architekturen gesehen und sollten stärker untersucht werden. Auch die Eignung des Hopfield-Ansatz für eine Klassifikation im Sinne des Data-Mining ist zu klären. Diese Ergebnisse können dann unmittelbar in ein System oben geschilderter Bauart einfließen.

Auf dem Gebiet des Data-Minings sind noch viele, möglicherweise auch spektakuläre Ergebnisse zu erwarten. Die Notwendigkeit des Data-Minings wird weiter bestehen, einfach weil die Daten-Mengen nicht kleiner, sondern eher größer werden. Damit wird der Wunsch nach Analyse dieser Daten-Mengen eher noch zunehmen. Neuronale Netze als eine Gruppe von Verfahren werden in diesem Szenario eine nicht zu unterschätzende Rolle spielen, da die Fähigkeit des Lernens aus Beispielen bzw. die Selbstorganisation anhand von Beispielen Lösungsmöglichkeiten aufzeigen, wo andere Verfahren keine befriedigenden Lösungen erzielen.

6.2. Data-Mining und Neuronale Netze in der Lehre

In der Ausbildung zum Wirtschaftsinformatiker am Fachbereich Wirtschaft der Hochschule Wismar werden Verfahren des Data-Mining sowie neuronale Netze behandelt. Durch den Aufbau einer Vertiefungsrichtung Wissensbasierte Systeme / Wissensmanagement können diese Themen vertieft und im Zusammenhang mit anderen Themen der Wissensverarbeitung dargestellt werden.

Durch den experimentellen Charakter des Data-Mining ist es zwar einerseits nicht möglich fertige Konzepte zu vermitteln, aber andererseits eröffnet das Gebiet schier unerschöpfliche Möglichkeiten für praxisnahe Projekte. Auch auf diesem Wege lassen sich Erfahrungen sammeln, um (Teil-)Antworten auf die in Abschnitt 6.1 gestellte Frage zu finden.

Der Wissensextraktion mittels Data-Mining ist nur eine Form des Wissenserwerbs und damit auch integrierter Bestandteil eines Wissensmanagements, welches in jüngster Zeit stark für den firmenweiten praktischen Einsatz propagiert wird.

Die Vertiefungsrichtung Wissensbasierte Systeme/Wissensmanagement bietet moderne, praxisnahe Themen, deren Bedeutung in der Weiterentwicklung in Richtung Wissensgesellschaft zunehmen wird. Innerhalb dieser Vertiefung wird es eine Vielzahl von Projekt-Themen geben, die den praktischen Einsatz der erworbenen Kenntnisse auf dem Gebiet des Data.Minings sowie der neuronalen Netze erfordern.

Literaturverzeichnis

[Brown2000] **Brown**, Michael P.S. u. a.: Knowledge-based Analysis of Microarray Gene Expression Data by using Support Vector Machines. - PNAS vol. 97,No.1, Januar 4, 2000, 262-267.

[CleveLämmel2002] **Cleve**, Jürgen; **Lämmel**, Uwe: DaMEn-Data-Mining Engineering. – Projektantrag im Rahmen des LFS-Iuk, HS Wismar, FB Wirtschaft, August 2002.

[CNC 97] **Ch@nnelWeb News Center**, "Data-Mining: Goldadern im Informations-Chaos", Special - Ausgabe 20/97 vom 26.09.1997, Computer Reseller Verlag GmbH, nach [Schmidt98]

[CorstenMai1996] **Corsten**, Hans; **May**, Constantin (Hrsg): Neuronale Netze in der Betriebswirtschaft. – Wiesbaden: Gabler-Verlag, 1996.

[EwertWeiß2002] **Ewert**, Silvio; **Weiß**, Steffen: SV-Umfrage und SOM. Projekt wissensbasierte Systeme, HS Wismar, FB Wirtschaft, Januar 2002.

[FAQNN2002] ftp://ftp.sas.com/pub/neural/FAQ.html, Juli 2002

[Fritzke97] **Fritzke**, Bernd: Some Competetive Learning Methods. - Ruhr-Univ. Bochum, 1997.

[Fritzke98] **Fritzke**, Bernd: Vektorbasierte Neuronale Netze. - Shaker-Verlag, 1998.

[Füser95] **Füser**, Karsten: Neuronale Netze in der Finanzwirtschaft. – Wiesbaden: Gabler-Verlag, 1995.

[Görz93] **Görz**, G. (ed): Einführung in die Künstliche Intelligenz. - Addsison-Wesley-Verlag, 1993

[IBM] http://www-3.ibm.com/software/data/iminer/fordata/

[Kecman2001] **Kecman**, Vojislav: Learning and Softcomputing. - MIT-Press, 2001.

[Khan2001] **Khan**, J. u. a.: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. - Nature Medicine, vol 7, No 6, June 2001 673-679.

[Klu97] **Klüting** R., Artikel: Moderne Zeiten: "Kunde in Klarsichtfolie", Kölner Stadt-Anzeiger, 1997, nach [Schmidt98]

[Kohonen1997] **Kohonen**, Teuvo: Self-Organizing Maps. - Springer-Verlag, 1997.

[LämmelCleve2001] **Lämmel**, Uwe; **Cleve**, Jürgen: Lehr- und Übungsbuch künstliche Intelligenz. - Leipzig: Fachbuchverlag, 2001.

[LämmelCleve2002] **Lämmel**, Uwe; **Cleve**, Jürgen: Data-Mining Engineering mit Anwendungen in Biotechnologie und Financial Engineering.

Teilprojekt im Rahmen des Antrags: Mobiles, multimediales Content Management, Heuer A. u. a. Univ. Rostock FB Informatik, Univ. Rostock.

[LionBioscience] http://www.lionbioscience.com/solutions/products/bioscout

[Lusti2002] **Lusti**, Markus: Data Warehousing und Data-Mining. – Berlin: Springer Verlag, 2002.

[MaimonLast2001] **Maimon**, Oded; **Last**, Mark: Knowledge Discovery and Data-Mining. - Dordrecht, Boston, London: Kluwer Academic Publ., 2001.

[Quakenbush2001] **Quackenbush**, John: Computational Analysis of Microarray Data. -Nature Reviews, vol 2, June 2001, 418-427

[Runkler2000] **Runkler**, Thomas A.: Information Mining - Vieweg-Verlag, 2000.

[Schmidt98] **Schmidt**, Wolfgang: Eine Cluster-Strategie für Data-Mining Probleme in Netzstrukturen. – Diplomarbeit, Fernuniv. Hagen
http://www.stud.fernuni-hagen.de/q3283267/ , Juni 2002

[Vesanto2000] **Vesanto**, Juha: Using SOM in Data-Mining. - Phd Thesis, Helsinki Univ. of Technology, 2000.

[WiedmannBuckler2001] **Wiedmann**, Klaus-Peter; **Buckler**, Frank (Hrsg): Neuronale Netze im Marketing-Managment. Wiesbaden: Gabler-Verlag, 2001.

[WittenFrank2000] **Witten**, Ian H.; **Frank** Eibe: Data-Mining. - San Francisco: Morgan-Kaufmann Publ., 2000.

[Zeit15/2002] Das Finanzamt wird abkassiert. - Die Zeit Nr. 15, 4.4.2002, S.20.

[Zell1997] **Zell**, Andreas: Simulation Neuronaler Netze. - München: Oldenbourg, 1997.

Autorenangaben

Prof. Dr.-Ing. Uwe Lämmel
Grundlagen der Informatik / Künstliche Intelligenz
Hochschule Wismar, Fachbereich Wirtschaft
Philipp-Müller-Straße
Postfach 12 10
D – 23966 Wismar
☎ ++49 / (0)3841 / 753 617
Fax ++49 / (0)3841 / 753 131
✉ u.laemmel@wi.hs-wismar.de www.wi.hs-wismar.de/~laemmel/

WDP - Wismarer Diskussionspapiere / Wismar Discussion Papers

Heft 01/2003	Jost W. Kramer: Fortschrittsfähigkeit gefragt: Haben die Kreditgenossenschaften als Genossenschaften eine Zukunft?
Heft 02/2003	Julia Neumann-Szyszka: Einsatzmöglichkeiten der Balanced Scorecard in mittelständischen (Fertigungs-)Unternehmen
Heft 03/2003	Melanie Pippig: Möglichkeiten und Grenzen der Messung von Kundenzufriedenheit in einem Krankenhaus
Heft 04/2003	Jost W. Kramer: Entwicklung und Perspektiven der produktivgenossenschaftlichen Unternehmensform
Heft 05/2003	Jost W. Kramer: Produktivgenossenschaften als Instrument der Arbeitsmarktpolitik. Anmerkungen zum Berliner Förderungskonzept
Heft 06/2003	Herbert Neunteufel/Gottfried Rössel/Uwe Sassenberg: Das Marketingniveau in der Kunststoffbranche Westmecklenburgs
Heft 07/2003	Uwe Lämmel: Data-Mining mittels künstlicher neuronaler Netze